# Package 'fitdistcp'

April 23, 2025

Type Package

**Title** Distribution Fitting with Calibrating Priors for Commonly Used Distributions

Version 0.1.1

Maintainer Stephen Jewson < stephen.jewson@gmail.com>

**Imports** stats, mev, extraDistr, gnorm, fdrtool, pracma, rust, actuar, fExtremes

**Depends** R (>= 3.5.0)

**Description** Generates predictive distributions based on calibrating priors for various commonly used statistical models, including models with predictors. Routines for densities, probabilities, quantiles, random deviates and the parameter posterior are provided. The predictions are generated from the Bayesian prediction integral, with priors chosen to give good reliability (also known as calibration). For homogeneous models, the prior is set to the right Haar prior, giving predictions which are exactly reliable. As a result, in repeated testing, the frequencies of out-of-sample outcomes and the probabilities from the predictions agree. For other models, the prior is chosen to give good reliability. Where possible, the Bayesian prediction integral is solved exactly. Where exact solutions are not possible, the Bayesian prediction integral is solved using the Datta-Mukerjee-Ghosh-Sweeting (DMGS) asymptotic expansion. Optionally, the prediction integral can also be solved using posterior samples generated using Paul Northrop's ratio of uniforms sampling package ('rust'). Results are also generated based on maximum likelihood, for comparison purposes. Various model selection diagnostics and testing routines are included. Based on ``Reducing reliability bias in assessments of extreme weather risk using calibrating priors", Jewson, S., Sweeting, T. and Jewson, L. (2024); <doi:10.5194/ascmo-11-1-2025>.

License MIT + file LICENSE

BugReports https://github.com/stephenjewson/fitdistcp/issues

URL https://fitdistcp.info

**Encoding UTF-8** 

LazyData true
RoxygenNote 7.3.1
Suggests knitr, rmarkdown
NeedsCompilation no
$\textbf{Author} \ \ \text{Stephen Jewson [aut, cre] ()}$
Repository CRAN
<b>Date/Publication</b> 2025-04-23 09:40:02 UTC

dhoc_dmgs_cpmethod	26
nalytic_cpmethod	26
ayesian_dq_4terms_v1	27
alc_revert2ml	27
auchy_cp	28
auchy_f1f	35
auchy_f1fa	35
auchy_f2f	36
auchy_f2fa	36
auchy_fd	37
auchy_fdd	37
auchy_ldd	38
auchy_ldda	38
auchy_lddd	39
auchy_lddda	39
auchy_lmn	40
auchy_lmnp	40
auchy_logf	41
auchy_logfdd	42
auchy_logfddd	42
auchy_loglik	43
auchy_logscores	43
auchy_mu1f	44
auchy_mu2f	44
auchy_p1f	45
auchy_p1_cp	45
auchy_p1_f1f	53
auchy_p1_f1fa	54
auchy_p1_f2f	54
auchy_p1_f2fa	55
auchy_p1_fd	55
auchy_p1_fdd	56
auchy_p1_ldd	56
auchy_p1_ldda	57
auchy_p1_lddd	58
auchy n1 lddda	58

cauchy_p1_lmn	. 59
cauchy_p1_lmnp	. 60
cauchy_p1_logf	60
cauchy_p1_logfdd	61
cauchy_p1_logfddd	62
cauchy_p1_loglik	62
cauchy_p1_logscores	63
cauchy_p1_means	
cauchy_p1_mu1f	
cauchy_p1_mu2f	65
cauchy_p1_p1f	
cauchy_p1_p2f	
cauchy_p1_predictordata	
cauchy_p1_waic	
cauchy_p2f	
cauchy_waic	
crhpflat_dmgs_cpmethod	
1100gamma_example_data_v1	
1101invgamma_example_data_v1	
1102invgauss_example_data_v1	
1105burr_example_data_v1	
110exp_example_data_v1	
d110gev_example_data_v1	
111pareto_k2_example_data_v1	
1120gpd_k1_example_data_v1	
1150gev_p1_example_data_v1_t	
1150gev_p1_example_data_v1_x	
1151gev_p1_example_data_v1_t	
1151gev_p12_example_data_v1_x	
1151gev_p12_example_data_v1_x	
1152gev_p123_example_data_v1_t	73
132gev_p123_example_data_v1_x	
125unif_example_data_v1	
130norm_example_data_v1	
131norm_dmgs_example_data_v1	
132gnorm_k3_example_data_v1	
135Inorm_example_data_v1	
•	
d36lnorm_dmgs_example_data_v1	
140logis_example_data_v1	
d41lst_k3_example_data_v1	
d42cauchy_example_data_v1	
d50gumbel_example_data_v1	
d51frechet_k1_example_data_v1	
d52weibull_example_data_v1	
d53gev_k3_example_data_v1	
d55exp_p1_example_data_v1_t	
d55exp_p1_example_data_v1_x	
156pareto p1k2 example data v1 t	. 76

d56pareto_p1k2_example_data_v1_x	/6
$d60 norm\_p1\_example\_data\_v1\_t \dots \dots$	76
$d60 norm\_p1\_example\_data\_v1\_x  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  $	77
$d61 lnorm\_p1\_example\_data\_v1\_t  .  .  .  .  .  .  .  .  .  $	77
$d61 lnorm\_p1\_example\_data\_v1\_x \ \dots $	77
$d62log is\_p1\_example\_data\_v1\_t \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $	77
$d62log is \_p1\_example\_data\_v1\_x  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  $	77
$d63lst\_p1k3\_example\_data\_v1\_t \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $	78
$d63lst\_p1k3\_example\_data\_v1\_x  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  $	78
$d64 cauchy\_p1\_example\_data\_v1\_t \dots \dots$	78
d64cauchy_p1_example_data_v1_x	78
$d70 gumbel\_p1\_example\_data\_v1\_t  .  .  .  .  .  .  .  .  .  $	78
d70gumbel_p1_example_data_v1_x	79
d71frechet_p2k1_example_data_v1_t	79
d71frechet_p2k1_example_data_v1_x	79
d72weibull_p1_example_data_v1_t	79
d72weibull_p1_example_data_v1_x	79
d73weibull_p2_example_data_v1_t	80
d73weibull_p2_example_data_v1_x	
d74gev_p1k3_example_data_v1_t	
d74gev_p1k3_example_data_v1_x	
d80norm_p12_example_data_v1_t1	
d80norm_p12_example_data_v1_t2	
d80norm_p12_example_data_v1_x	81
d81lst_p12k3_example_data_v1_t1	81
d81lst_p12k3_example_data_v1_t2	81
d81lst_p12k3_example_data_v1_x	81
d82weibull_p12_example_data_v1_t1	82
d82weibull_p12_example_data_v1_t2	82
d82weibull_p12_example_data_v1_x	82
deauchysub	82
dcauchy_p1	83
dcauchy_p1sub	83
deriv_copyfdd	84
deriv_copyld2	85
deriv_copyldd	85
deriv_copylddd	
dexpsub	86
dexp_p1	87
dexp_plsub	87
dfrechetsub	88
dfrechet_p2k1	88
dfrechet_p2k1sub	89
dgammasub	90
dgevsub	90
dgev_k3sub	91
dgev_p1	92
dgev n12	92

dgev_p123												
dgev_p123sub												
lgev_p12sub	 	 	 	 	 							95
lgev_p1k3	 	 	 	 	 							96
gev_p1k3sub	 	 	 	 	 							97
lgev_p1sub												
lgnorm_k3sub												
gpdsub												
gumbelsub												
gumbel_p1												
gumbel_p1 · · · · · lgumbel_p1sub · · ·												
•												
halfnormsub												
invgammasub												
nvgausssub												
lnormsub												
lnorm_dmgssub .												
lnorm_p1	 	 	 	 	 							104
lnorm_p1sub	 	 	 	 	 							105
logis2sub	 	 	 	 	 							105
logis_p1												
logis_p1sub												
st_k3sub												
st p1k3												
lst_p1k3												
mgs												
•												
normsub												
norm_dmgssub												
norm_p1												
lnorm_p1sub												
lnorm_p1_formula												
lpareto_k2_sub												
lpareto_p1k2	 	 	 	 	 							113
lpareto_p1k2sub .	 	 		 	 							113
lunif_formula	 	 	 	 	 							114
weibullsub	 	 	 	 	 							114
lweibull_p2												
lweibull_p2sub												
exp_cp												
exp_f1f												
хр_111												
<b>.</b> —												
xp_f2f												
xp_f2fa												
xp_fd												
xp_fdd												
exp_l1111												
exp_ldd	 	 	 	 	 							125
exp_ldda	 	 	 	 	 							126
exp_lddd	 	 	 	 	 							126

exp_lddda	
exp_logf	127
exp_logfdd	128
exp_logfddd	
exp_logscores	129
$exp\_p1fa\ldots\ldots\ldots\ldots\ldots\ldots$	129
$exp\_p1\_cp\ldots\ldots\ldots\ldots\ldots\ldots$	
$exp\_p1\_f1f  \dots $	
$exp\_p1\_f1fa \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $	
exp_p1_f2f	
exp_p1_f2fa	
$exp\_p1\_fd \ \dots $	139
$exp\_p1\_fdd \ \dots \dots$	140
$exp\_p1\_ldd \ \dots \dots$	
exp_p1_ldda	141
$exp\_p1\_lddd  \dots $	141
$exp\_p1\_lddda \ \dots $	142
exp_p1_lmn	142
exp_p1_lmnp	143
exp_p1_logf	143
exp_p1_logfdd	144
exp_p1_logfddd	145
exp_p1_loglik	145
exp_p1_logscores	146
exp_p1_means	146
exp_p1_mu1f	147
exp_p1_mu1fa	148
exp_p1_mu2f	148
exp_p1_mu2fa	149
exp_p1_p1f	149
exp_p1_p1fa	150
exp_p1_p2f	150
exp_p1_p2fa	151
exp_p1_pd	151
exp_p1_pdd	152
exp_p1_predictordata	152
exp_p1_waic	153
exp_p2fa	154
exp_pd	
exp_pdd	
exp_waic	
fixgevrange	
fixgpdrange	
frechet_k1_cp	
frechet k1 f1f	
frechet_k1_f1fa	
frechet k1 f2f	
frechet_k1_f2fa	

$frechet\_k1\_fd \ \dots $	166
$frechet\_k1\_fdd  .  .  .  .  .  .  .  .  . $	
$frechet\_k1\_ldd  \dots \dots$	167
$frechet\_k1\_ldda \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $	168
$frechet\_k1\_lddd \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$	168
$frechet\_k1\_lddda \ \ldots \ $	169
frechet_k1_lmn	169
frechet_k1_lmnp	
frechet_k1_logf	
frechet_k1_logfdd	
frechet_k1_logfddd	
frechet_k1_mu1f	
frechet_k1_mu1fa	
frechet_k1_mu2f	
frechet_k1_mu2fa	
frechet_kl_plf	
frechet_k1_p1fa	
frechet_k1_p2f	
frechet_k1_p2fa	
frechet_k1_pd	
frechet_k1_pdd	
frechet_k1_waic	
frechet_loglik	
frechet_logscores	
frechet_means	
frechet_p2k1_cp	
frechet_p2k1_f1f	
frechet_p2k1_f1fa	
frechet_p2k1_f2f	
frechet_p2k1_f2fa	
$frechet\_p2k1\_fd \dots \dots$	
$frechet\_p2k1\_fdd \ \dots \dots$	
$frechet\_p2k1\_ldd \ \dots \dots$	192
$frechet\_p2k1\_ldda  .  .  .  .  .  .  .  .  .  $	193
$frechet\_p2k1\_lddd  \dots $	
frechet_p2k1_lddda	194
frechet_p2k1_lmn	195
frechet_p2k1_lmnp	195
frechet_p2k1_logf	196
frechet_p2k1_logfdd	197
frechet_p2k1_logfddd	
frechet_p2k1_loglik	
frechet_p2k1_logscores	
frechet_p2k1_means	
frechet_p2k1_mu1f	
frechet_p2k1_mu1fa	
frechet_p2k1_mu2f	
frechet_p2k1_mu2fa	
1100110t_p2R1_1110210	202

frechet_p2k1_p1f	203
	203
frechet_p2k1_p1fa	204
frechet_p2k1_p2f	204
frechet_p2k1_p2fa	205
frechet_p2k1_pd	206
frechet_p2k1_pdd	
frechet_p2k1_predictordata	
frechet_p2k1_waic	
gamma_cp	
gamma_flf	
gamma flfa	
gamma_f2f	
gamma_f2fa	
gamma_fd	
gamma_fdd	
gamma_gg	
gamma_gmn	
gamma_ldd	
gamma_ldda	
gamma_lddd	
gamma_lddda	
gamma_lmn	
gamma_lmnp	
gamma_logf	
gamma_logfdd	
gamma_logfddd	
gamma_loglik	
gamma_logscores	
gamma_means	
gamma_mu1f	
gamma_mu2f	
gamma_p1f	
gamma_p2f	
gamma_waic	
gev_checkmle	229
gev_cp	
gev_flf	239
gev_f1fa	240
gev_f2f	240
gev_f2fa	241
gev_fd	241
gev_fdd	242
gev_ggd_mev	
gev_ggid_mev	
gev_k12_ppm_minusloglik	
gev_k3_cp	
gev_k3_f1f	
gev k3 flfa	

gev_k3_f2f	4
gev_k3_f2fa	4
gev_k3_fd	5
gev k3 fdd	5
gev k3 ldd	6
gev_k3_ldda	
gev_k3_lddd	
gev_k3_lddda	
gev_k3_lmn	
gev_k3_lmnp	
gev_k3_logf	
gev_k3_logfdd	
gev_k3_logfddd	
gev_k3_loglik	
gev_k3_means	
gev_k3_mu1f	
gev_k3_mu1fa	
gev_k3_mu2f	
gev_k3_mu2fa	
gev_k3_pd	
gev_k3_pdd	
gev_k3_waic	
gev_ld12a	
gev_lda	
gev_ldd	
gev_ldda	
gev_lddd	
gev_lddda	
gev_lmn	
gev_lmnp	
gev_logf	
gev_logfd	
gev_logfdd	
gev_logfddd	
gev_loglik	
gev means	
gev mulf	
gev_mulfa	
gev_mu2f	
gev_mu2fa	
gev_p123_checkmle	
gev_p123_cp	
gev_p123_f1f	
gev_p123_f1fa	
gev_p123_f2f	
gev_p123_f2fa	
gev_p123_fd	
gev p123 fdd	

gev_p123_ldd	
gev_p123_ldda	
gev_p123_lddd	. 293
gev_p123_lddda	. 294
gev_p123_lmn	. 294
gev_p123_lmnp	
gev_p123_logf	
gev_p123_logfdd	
gev_p123_logfddd	
gev_p123_loglik	
gev_p123_means	
gev_p123_mu1f	
gev_p123_mu1fa	
gev_p123_mu2f	
gev_p123_mu2fa	
gev_p123_pd	
gev_p123_pdd	
gev_p123_predictordata	
gev_p123_setics	
gev_p123_waic	
gev_p1283_f1f	
gev_p12k3_f1fa	
gev_p12k3_f2f	
gev_p12k3_f2fa	
gev_p12k3_fd	
gev_p12k3_fdd	
gev_p12k3_ldd	
gev_p12k3_ldda	
gev_p12k3_lddd	
gev_p12k3_lddda	
C =1 =	
gev_p12k3_logfdd	
gev_p12k3_logfddd	
gev_p12k3_mu1fa	
gev_p12k3_mu2f	
gev_p12k3_mu2fa	
gev_p12k3_pd	
gev_p12k3_pdd	
gev_p12_checkmle	
gev_p12_cp	
gev_p12_f1f	
gev_p12_f1fa	
gev_p12_f2f	
gev_p12_f2fa	
gev_p12_fd	
gev_p12_fdd	
gev_p12_ggd	
gev_p12_ldd	. 335

gev_p12_ldda	335
$gev\_p12\_lddd \ \dots $	336
$gev\_p12\_lddda  .  .  .  .  .  .  .  .  .  $	
gev_p12_lmn	
gev_p12_lmnp	
$gev\_p12\_logf \ \dots $	339
gev_p12_logfdd	340
gev_p12_logfddd	340
gev_p12_loglik	341
gev_p12_means	342
$gev\_p12\_mu1f  \dots $	342
gev_p12_mu1fa	343
gev_p12_mu2f	344
gev_p12_mu2fa	344
gev_p12_pd	345
gev_p12_pdd	
gev_p12_predictordata	
gev_p12_setics	
gev_p12_waic	
gev_p1k3_cp	
gev_p1k3_f1f	
gev_p1k3_f1fa	
gev_p1k3_f2f	
gev_p1k3_f2fa	
gev_p1k3_fd	
gev_p1k3_fdd	
gev_p1k3_ldd	
gev_p1k3_ldda	
gev_p1k3_lddd	
gev_p1k3_lddda	
gev_p1k3_lmn	
gev_p1k3_lmnp	
gev p1k3 logf	
gev_p1k3_logfdd	
gev_p1k3_logfddd	
gev p1k3 loglik	
gev_p1k3_means	
gev_p1k3_mu1f	
gev_p1k3_mu1fa	
gev_p1k3_mu2f	
gev_p1k3_mu2fa	
gev_p1k3_pd	
gev_p1k3_pdd	
gev_p1k3_predictordata	
gev_p1k3_waic	
gev_p1_checkmle	
gev_p1_cneckinie	
gev_pr_cp	
EUV DI 111	

Contents Contents

gev_p1_f1fa
gev_p1_f2f
gev_p1_f2fa
gev_pl_fd
gev p1 fdd
gev_pl_ggd
gev_p1_ldd
gev p1 ldda
gev_p1_lddd
gev_p1_lddda
gev p1 lmn
gev_p1_lmnp
gev_pl_logf
gev_pl_logfdd
gev_pl_logfddd
gev_pl_loglik
gev_pl_means
gev_pl_mulf
gev_pl_mulfa
gev_p1_mu2f
gev_p1_mu2fa
gev_pl_pd
gev_pl_pdd
gev p1 predictordata
gev_pl_setics
gev_p1_waic
gev_pd
gev_pdd
gev_pwm_params
gev_setics
gev_waic
gnorm_k3_cp
gnorm_k3_f1f
gnorm_k3_f1fa
gnorm_k3_f2f
gnorm k3 f2fa
gnorm k3 fd
gnorm_k3_fdd
gnorm_k3_ldd
gnorm_k3_ldda
gnorm_k3_lddd
gnorm_k3_lddda
gnorm_k3_lmn
gnorm_k3_logf
gnorm_k3_logfdd
gnorm_k3_logfddd
gnorm_k3_loglik
gnorm_k3_logscores
gnorm_kJ_rogscores

gnorm_k3_mu1f .								
$gnorm\_k3\_mu2f  .$	 	 . 422						
gnorm_k3_p1f	 	 . 423						
gnorm_k3_p2f	 	 . 423						
gnorm_lmnp	 	 . 424						
gnorm_waic								
gpd_k13_f1f	 	 . 426						
gpd_k13_f1fa								
gpd_k13_f2f	 	 . 427						
gpd_k13_f2fa	 	 . 427						
gpd_k13_fd								
gpd_k13_fdd	 	 . 428						
gpd_k13_l11	 	 . 429						
gpd_k13_l111	 	 . 429						
gpd_k13_ldd								
gpd_k13_ldda								
gpd_k13_lddd								
gpd_k13_lddda								
gpd_k13_logfdd								
gpd_k13_logfddd .								
gpd_k13_mu1f								
gpd_k13_mu1fa								
gpd_k13_mu2f	 	 . 434						
gpd_k13_mu2fa	 	 . 434						
gpd_k13_p1f	 	 . 435						
gpd_k13_p2f	 	 . 435						
gpd_k13_pd	 	 . 436						
gpd_k13_pdd	 	 . 436						
gpd_k1_checkmle.	 	 . 437						
gpd_k1_cp	 	 . 437						
gpd_k1_f1f	 	 . 446						
gpd_k1_f1fa	 	 . 447						
gpd_k1_f2f	 	 . 447						
gpd_k1_f2fa	 	 . 448						
gpd_k1_fd	 	 . 448						
gpd_k1_fdd	 	 . 449						
gpd_k1_ggd_mev .	 	 . 449						
gpd_k1_ldd	 	 . 450						
gpd_k1_ldda	 	 . 450						
gpd_k1_lddd	 	 . 451						
gpd_k1_lddda	 	 . 451						
gpd_k1_lmn	 	 . 452						
gpd_k1_lmnp	 	 . 452						
gpd_k1_logf	 	 . 453						
gpd_k1_logfdd	 	 . 454						
gpd_k1_logfddd								
gpd_k1_loglik								
gpd_k1_means								

Contents Contents

gpd_k1_mu1f	
gpd_k1_mu1fa	
gpd_k1_mu2f	457
gpd_k1_mu2fa	458
gpd_k1_p1f	
gpd_k1_p2f	
gpd_k1_pd	
gpd_k1_pdd	460
gpd_k1_setics	460
gpd_k1_waic	461
gumbel_cp	462
gumbel_f1f	469
gumbel_f1fa	469
gumbel_f2f	470
gumbel_f2fa	470
gumbel_fd	471
gumbel_fdd	471
gumbel_ldd	472
gumbel_ldda	472
gumbel_lddd	473
gumbel_lddda	473
gumbel_lmn	474
gumbel_lmnp	474
gumbel_logf	475
gumbel_logfdd	476
gumbel_logfddd	476
gumbel_loglik	477
gumbel_logscores	477
gumbel_means	478
gumbel_mu1f	478
gumbel_mu1fa	479
gumbel_mu2f	479
gumbel_mu2fa	480
gumbel_p1f	
gumbel_p1fa	
gumbel_p1_cp	
gumbel_p1_f1f	
gumbel_p1_f1fa	
gumbel_p1_f2f	
gumbel_p1_f2fa	
gumbel_pl_fd	
gumbel_p1_fdd	
gumbel_pl_ldd	
gumbel_p1_ldda	
gumbel_p1_lddd	
gumbel_p1_lddda	
gumbel_p1_lmn	
C -1 -	496

gumbel_p1_logf	
gumbel_p1_logfdd	
gumbel_p1_logfddd	198
gumbel_p1_loglik	198
gumbel_p1_logscores	199
gumbel_p1_means	199
gumbel_p1_mu1f	500
gumbel_p1_mu1fa	501
gumbel_p1_mu2f	
gumbel_p1_mu2fa	502
gumbel_p1_p1f	502
gumbel pl plfa	
gumbel_p1_p2f	504
gumbel_p1_p2fa	504
gumbel_p1_pd	
gumbel_p1_pdd	
gumbel_p1_predictordata	
gumbel p1 waic	
gumbel p2f	
gumbel_p2fa	
gumbel_pd	
gumbel_pdd	
gumbel waic	
halfnorm_cp	
halfnorm f1f	
halfnorm f1fa	518
halfnorm f2f	518
halfnorm f2fa	
halfnorm fd	
halfnorm fdd	
halfnorm_gg	
halfnorm_gg11	
halfnorm 1111	
halfnorm_ldd	
halfnorm ldda	
halfnorm lddd	
halfnorm_lddda	
halfnorm logf	
halfnorm_logfdd	
halfnorm_logfddd	
halfnorm_loglik	
halfnorm_logscores	
halfnorm means	
halfnorm_mu1f	
halfnorm_mu2f	
halfnorm_p1f	
halfnorm_p2f	
halfnorm waic	

Contents Contents

invgamma_cp	 . 529
invgamma_f1f	 . 536
invgamma_f1fa	 . 537
invgamma_f2f	 . 538
invgamma_f2fa	 . 538
invgamma_fd	 . 539
invgamma_fdd	 . 539
invgamma_ldd	 . 540
invgamma_ldda	 . 540
invgamma_lddd	 . 541
invgamma_lddda	 . 541
invgamma_lmn	 . 542
invgamma_lmnp	
invgamma_logf	 . 543
invgamma_logfdd	 . 544
invgamma_logfddd	 . 544
invgamma_loglik	
invgamma_logscores	 . 545
invgamma_mu1f	 . 546
invgamma_mu2f	 . 546
invgamma_p1f	 . 547
invgamma_p2f	 . 547
invgamma_waic	 . 548
invgauss_cp	 . 549
invgauss_f1f	 . 556
invgauss_f1fa	 . 557
invgauss_f2f	 . 557
invgauss_f2fa	 . 558
invgauss_fd	 . 558
invgauss_fdd	 . 559
invgauss_ldd	 . 559
invgauss_ldda	 . 560
invgauss_lddd	 . 560
invgauss_lddda	 . 561
invgauss_lmn	 . 561
invgauss_lmnp	
invgauss_logf	 . 562
invgauss_logfdd	 . 563
invgauss_logfddd	 . 563
invgauss_loglik	 . 564
invgauss_logscores	 . 564
invgauss_means	 . 565
invgauss_mulf	 . 566
invgauss_mu2f	 . 566
invgauss_p1f	 . 567
invgauss_p2f	 . 567
invgauss_waic	 . 568
jpf2p	 . 569

jpf3p	
jpf4p	
lnorm_cp	
lnorm_dmgs_cp	
lnorm_dmgs_gg11	. 583
lnorm_dmgs_gg12	. 584
lnorm_dmgs_gg22	. 584
lnorm_dmgs_loglik	. 585
lnorm_dmgs_logscores	. 585
lnorm_dmgs_means	. 586
lnorm_dmgs_mu1f	. 586
lnorm_dmgs_mu2f	. 587
lnorm_dmgs_p1f	. 587
lnorm_dmgs_p2f	. 588
lnorm_dmgs_waic	. 588
lnorm_f1f	. 589
lnorm_f1fa	
lnorm f2f	
lnorm f2fa	. 591
lnorm_fd	. 591
lnorm_fdd	
lnorm_ldd	
lnorm ldda	
lnorm_lddd	. 593
lnorm_lddda	. 594
lnorm_lmn	. 594
lnorm_lmnp	. 595
lnorm_logf	. 595
lnorm_logfdd	. 596
lnorm_logfddd	. 596
lnorm_logscores	. 597
lnorm_mu1fa	. 597
lnorm_mu2fa	. 598
lnorm_p1fa	. 598
lnorm_p1_cp	. 599
lnorm_p1_f1f	
lnorm_p1_f1fa	. 607
lnorm_p1_f2f	. 607
lnorm_p1_f2fa	. 608
lnorm_p1_fd	. 608
lnorm_p1_fdd	. 609
lnorm_p1_ldd	. 609
lnorm_p1_ldda	. 610
lnorm_p1_lddd	. 611
lnorm_p1_lddda	. 611
lnorm_p1_lmn	
lnorm_p1_lmnp	. 613
lnorm_p1_logf	

Contents Contents

lnorm_p1_logfdd	
lnorm_p1_logfddd	
lnorm_p1_loglik	
lnorm_p1_logscores	616
lnorm_p1_mu1fa	616
lnorm_p1_mu2fa	617
lnorm_p1_p1fa	617
lnorm_p1_p2fa	618
lnorm_p1_pd	618
lnorm_p1_pdd	619
lnorm_p1_predictordata	
lnorm_p1_waic	620
lnorm_pdd	
lnorm_waic	
logis_cp	
logis_f1f	
logis_f1fa	
logis f2f	
logis_f2fa	
logis fd	
logis_fdd	
<b>c</b> –	
logis_ldd	
logis_ldda	
logis_lddd	
logis_lddda	
logis_lmn	
logis_lmnp	
logis_logf	
logis_logfdd	
logis_logfddd	
logis_loglik	
logis_logscores	
logis_mu1f	
logis_mu1fa	
logis_mu2f	640
logis_mu2fa	
logis_p1f	641
logis_p1fa	641
logis_p1_cp	
$logis\_p1\_f1f\ldots\ldots\ldots\ldots\ldots\ldots\ldots$	649
logis_p1_f1fa	650
logis_p1_f2f	651
logis_p1_f2fa	651
logis_p1_fd	652
logis_p1_fdd	652
logis pl ldd	

1 ' 1 111	
logis_p1_ldda	
logis_p1_lddd	
logis_p1_lddda	
logis_p1_lmn	
logis_p1_lmnp	
logis_p1_logf	
logis_p1_logfdd	
logis_p1_logfddd	
logis_p1_loglik	
logis_p1_logscores	
logis_p1_means	(
logis_p1_mu1f	
logis_p1_mu1fa	
logis_p1_mu2f	
logis_p1_mu2fa	
logis_pl_plf	
logis_p1_p1fa	
logis_p1_p2f	
logis_p1_p2fa	
logis_p1_pd	
logis_p1_pdd	
logis_p1_predictordata	
logis_p1_waic	
logis_p2f	
logis_p2fa	(
logis_pd	(
logis_pdd	(
logis_waic	
lst_k3_cp	
lst_k3_f1f	
lst_k3_f1fa	
lst k3 f2f	
lst_k3_f2fa	
lst_k3_fd	
lst_k3_fdd	
lst_k3_ldd	
lst_k3_ldda	
lst_k3_lddd	
lst_k3_lddda	
lst_k3_lmn	(
lst_k3_lmnp	
lst_k3_logf	(
lst_k3_logfdd	
lst_k3_logfddd	
lst_k3_loglik	
lst k3 logscores	
lst_k3_mu1f	
lst k3 mu2f	
10t KJ 111421	

lst_k3_p1f													
lst_k3_p2f													
lst_k3_waic		 	 		 								. 689
lst_p1k3_cp													
lst_p1k3_f1f		 	 		 								. 698
lst_p1k3_f1fa .		 	 		 								. 699
lst_p1k3_f2f		 	 		 								. 699
lst_p1k3_f2fa .		 	 		 								. 700
lst_p1k3_fd													
lst_p1k3_fdd													
lst_p1k3_ldd .													
lst_p1k3_ldda .													
st_p1k3_lddd .													
st_p1k3_lddda													
lst_p1k3_lmn .													
st_p1k3_lmnp													
lst_p1k3_mmp													
-ı – c													
lst_p1k3_logfdd													
st_p1k3_logfddd													
st_p1k3_loglik													
st_p1k3_logscore													
t_p1k3_mu1f													
st_p1k3_mu2f													
t_p1k3_p1f .													
st_p1k3_p2f		 	 		 								. 711
st_p1k3_predicto	rdata	 	 		 								. 712
st_p1k3_setics		 	 		 								. 713
st_p1k3_waic .		 	 		 								. 713
nakemuhat0		 	 		 								. 714
nakeq		 	 		 								. 715
maket0		 	 		 								. 715
naketa0													
nake cwaic													
nake_maic													
nake_se													
nake_waic													
nan	• • •												. 718 . 718
nan	• • •												
nan1f													
nan2f													
nandsub													
nanf													
nanldd													
nanlddd													
nanlnn		 	 		 								. 736
nanlnnn		 	 		 								. 737
manlogf		 	 		 								. 737
manloglik		 	 		 								. 737
manlogscores .													

manmeans	
manpredictor	38
manvector	39
manwaic	39
movexiawayfromzero	39
ms_flat_1tail	<del>1</del> 0
ms_flat_2tail	41
ms_predictors_1tail	12
ms_predictors_2tail	13
nopdfcdfmsg	14
norm_cp	<b>15</b>
norm_dmgs_cp	50
norm_dmgs_loglik	56
norm_dmgs_logscores	57
norm_dmgs_means	57
norm_dmgs_mu1f	58
norm_dmgs_mu2f	59
norm_dmgs_p1f	59
norm_dmgs_p2f	
norm_dmgs_waic	
norm_f1f	
norm_f1fa	52
norm_f2f	52
norm_f2fa	53
norm_fd	53
norm_fdd	54
norm_gg	54
norm_gmn	55
norm_ldd	55
norm_ldda	56
norm_lddd	56
norm_lddda	<b>57</b>
norm_lmn	<b>57</b>
norm_lmnp	58
norm_logf	59
norm_logfdd	59
norm_logfddd	<b>70</b>
norm_logscores	<b>70</b>
norm_ml_params	71
norm_mulfa	71
norm_mu2fa	72
norm_p1fa	72
norm_p1_cp	73
norm_p1_flf	30
norm_p1_f1fa	31
norm_p1_f2f	31
norm_p1_f2fa	32
norm_p1_fd	33

$norm\_p1\_fdd \ \dots \dots$	783
$norm\_p1\_ldd  \dots \dots$	784
$norm\_p1\_ldda \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$	785
$norm\_p1\_lddd\dots$	785
norm_p1_lddda	786
norm_p1_lmn	787
norm_p1_lmnp	787
norm_p1_logf	788
norm_p1_logfdd	789
norm_p1_logfddd	
norm_p1_loglik	
norm_p1_logscores	
norm_p1_mlparams	
norm_p1_mu1fa	
norm_p1_mu2fa	
norm_pl_p1fa	
norm_p1_p2fa	
norm_p1_pd	
norm_p1_pdd	
norm_p1_predictordata	
norm_p1_waic	
norm_p2fa	
norm_pd	
norm_pdd	
norm_unbiasedv_params	
norm_waic	
pareto k2 cp	
pareto k2 f1f	
pareto_k2_f1fa	
<u> </u>	
pareto_k2_f2f	
<u> </u>	
pareto_k2_fd	
pareto_k2_fdd	
pareto_k2_l111	
pareto_k2_ldd	
pareto_k2_ldda	
pareto_k2_lddd	
pareto_k2_lddda	
pareto_k2_logf	
pareto_k2_logfdd	
pareto_k2_logfddd	
pareto_k2_logscores	
pareto_k2_ml_params	
pareto_k2_mu1fa	
pareto_k2_mu2fa	
pareto_k2_p1fa	
$pareto\_k2\_p2fa  .  .  .  .  .  .  .  .  .  $	
pareto k2 pd	817

pareto_k2_pdd	817
pareto_k2_waic	818
pareto_p1k2_cp	818
pareto_p1k2_f1f	826
pareto_p1k2_f1fa	827
pareto_p1k2_f2f	827
pareto_p1k2_f2fa	828
pareto_p1k2_fd	828
pareto_p1k2_fdd	829
pareto_p1k2_ldd	829
pareto p1k2 ldda	
pareto_p1k2_lddd	831
pareto_p1k2_lddda	
pareto_p1k2_lmn	
pareto_p1k2_lmnp	
pareto_p1k2_logf	
pareto_p1k2_logfdd	
pareto_p1k2_logfddd	
pareto_p1k2_loglik	
pareto_p1k2_logscores	
pareto_p1k2_means	
pareto_p1k2_mu1f	
pareto_p1k2_mu1fa	
pareto_p1k2_mu2f	
pareto_p1k2_mu2fa	
pareto_p1k2_p1f	
pareto_p1k2_p1fa	
pareto_p1k2_p2f	
pareto_p1k2_p2fa	
pareto_p1k2_pd	
pareto_p1k2_pdd	
pareto_p1k2_predictordata	
pareto_p1k2_waic	
ocauchy_p1	
pexp_p1	
ofrechet p2k1	
pgev_p1	
- L	
ogev_p12	
ogev_p1k3	
ogumbel_p1	
plnorm_p1	
plogis_p1	
plst_p1k3	
onorm_p1	
onorm_p1_formula	
ppareto_p1k2	
ounif formula	852

pweibull_p2	852
qcauchy_p1	853
qexp_p1	853
qfrechet_p2k1	
qgamma_k1_ppm	
qgamma_ppm	
ggev_k12_ppm	
qgev_mpd_ppm	
qgev_n1	
qgev_p12	
qgev_p123	
qgev_p1k3	
qgev_p1_ppm	
qgev_ppm	
qgpd_k1_ppm	
qgumbel_p1	
qlnorm_p1	867
qlogis_p1	868
qlst_p1k3	868
qnorm_p1	869
qnorm_p1_formula	869
 qntt_ppm	870
qpareto_p1k2	
qunif_formula	
qweibull_p2	
reltest	
reltest2	
reltest2 cases	
reltest2_makeep	
reltest2_plot	
reltest2_predict	
reltest2_simulate	
reltest_makeep	
reltest_makemaxep	
reltest_predict	
reltest_simulate	
rgev_minmax	886
rgev_p123_minmax	
rgev_p12_minmax	887
rgev_p1_minmax	888
rgpd_k1_minmax	889
rhp_dmgs_cpmethod	889
rust_pumethod	
testppm_plot	
unif_cp	
weibull_cp	
weibull_f1f	
weibull flfa	
WVIUUII 111U	ハハ

weibull_f2f	. 905
weibull_f2fa	. 905
weibull_fd	. 906
weibull_fdd	. 906
weibull_ldd	. 907
weibull_ldda	. 907
weibull_lddd	. 908
weibull_lddda	. 908
weibull_lmn	. 909
weibull_lmnp	. 909
weibull_logf	. 910
weibull_logfdd	. 911
weibull_logfddd	. 911
weibull_loglik	. 912
weibull_logscores	. 912
weibull_means	
weibull_mu1f	
weibull_mu1fa	
weibull_mu2f	. 914
weibull_mu2fa	
weibull_p1f	. 915
weibull_p1fa	
weibull_p2f	
weibull_p2fa	. 917
weibull_p2_cp	. 917
weibull_p2_f1f	. 925
weibull_p2_f1fa	. 926
weibull_p2_f2f	. 926
weibull_p2_f2fa	. 927
weibull_p2_fd	. 927
weibull_p2_fdd	. 928
weibull_p2_ldd	. 928
weibull_p2_ldda	
weibull_p2_lddd	
weibull p2 lddda	
weibull_p2_lmn	
weibull_p2_lmnp	
weibull_p2_logf	
weibull_p2_logfdd	
weibull_p2_logfddd	
weibull_p2_loglik	
weibull_p2_logscores	
weibull_p2_means	
weibull_p2_mu1f	
weibull_p2_mu1fa	
weibull_p2_mu2f	
weibull_p2_mu2fa	
weibull p2 p1f	

26 analytic\_cpmethod

Index		946
	weibull_waic	945
	weibull_pdd	
	weibull_pd	944
	weibull_p2_waic	943
	weibull_p2_predictordata	942
	weibull_p2_pdd	941
	weibull_p2_pd	941
	weibull_p2_p2fa	940
	weibull_p2_p2f	940
	weibull_p2_p1fa	939

 ${\tt adhoc\_dmgs\_cpmethod}$ 

Generates a comment about the method

## Description

Generates a comment about the method

## Usage

```
adhoc_dmgs_cpmethod()
```

#### Value

String

analytic\_cpmethod

Generates a comment about the method

## Description

Generates a comment about the method

## Usage

```
analytic_cpmethod()
```

## Value

String

bayesian\_dq\_4terms\_v1 Evaluate DMGS equation 3.3

#### **Description**

Evaluate DMGS equation 3.3

#### Usage

```
bayesian_dq_4terms_v1(lddi, lddd, mu1, pidopi1, pidopi2, mu2, dim)
```

## Arguments

lddi inverse of second derivative of observed log-likelihood

1ddd third derivative of observed log-likelihood

mu1 DMGS mu1 vector

pidopi1 first part of the prior term pidopi2 second part of the prior term

mu2 DMGS mu2 matrix dim number of parameters

#### Value

Vector

calc\_revert2ml

determine revert2ml or not

## Description

determine revert2ml or not

#### Usage

```
calc_revert2ml(v5h, v6h, t3)
```

#### **Arguments**

v5h fifth parameter v6h sixth parameter

t3 a vector of predictors for the shape

#### Value

Logical

cauchy\_cp

Cauchy Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qcauchy_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    d1 = 0.01,
    fd2 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE,
    aderivs = TRUE
)
```

```
n,
 х,
 d1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
dcauchy_cp(
 х,
 y = x,
 d1 = 0.01,
 fd2 = 0.01,
  rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
pcauchy_cp(
 х,
 y = x,
 d1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
tcauchy_cp(n, x, d1 = 0.01, fd2 = 0.01, debug = FALSE)
```

a vector of training data values

# **Arguments** x

	· · · · · · · · · · · · · · · · · · ·
р	a vector of probabilities at which to generate predictive quantiles
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter $% \left( 1\right) =\left( 1\right) \left( 1\right) $
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter $$
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)

dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
y	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Cauchy distribution has probability density function

$$f(x; \mu, \sigma) = \frac{1}{\pi \sigma} \left( 1 + \left( \frac{x - \mu}{\sigma} \right)^2 \right)^{-1}$$

where x is the random variable and  $\mu, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2024) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),

- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
# # example 1
x=fitdistcp::d42cauchy_example_data_v1
p=c(1:9)/10
q=qlogis_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qcauchy_cp)",
main="Cauchy: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

cauchy\_f1f 35

	_	_
cauchy	£1	f

 $DMGS\ equation\ 3.3, f1\ term$ 

## Description

DMGS equation 3.3, f1 term

## Usage

```
cauchy_f1f(y, v1, d1, v2, fd2)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

## Value

Matrix

cauchy_f1fa	ca	uchv	f1f	à
-------------	----	------	-----	---

The first derivative of the density

## Description

The first derivative of the density

## Usage

```
cauchy_f1fa(x, v1, v2)
```

## Arguments

x a vector of training data valu
----------------------------------

v1 first parameter v2 second parameter

#### Value

Vector

36 cauchy\_f2fa

00110	h.,	£24
cauc	ΠV	$T \angle T$

 $DMGS\ equation\ 3.3, f2\ term$ 

## Description

DMGS equation 3.3, f2 term

## Usage

```
cauchy_f2f(y, v1, d1, v2, fd2)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

## Value

3d array

cauchy_	f2fa

The second derivative of the density

## Description

The second derivative of the density

## Usage

```
cauchy_f2fa(x, v1, v2)
```

## Arguments

v1 first parameter v2 second parameter

#### Value

Matrix

cauchy\_fd 37

cauchy_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
cauchy_fd(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Vector

cauchy_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
cauchy_fdd(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

38 cauchy\_ldda

	h.,	ı	ᅬ
caucl	IIV_	тu	u

Second derivative matrix of the normalized log-likelihood

### Description

Second derivative matrix of the normalized log-likelihood

### Usage

```
cauchy_ldd(x, v1, d1, v2, fd2)
```

### Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

eter

### Value

Square scalar matrix

The second derivative of the normalized log-likelihood

### Description

The second derivative of the normalized log-likelihood

### Usage

```
cauchy_ldda(x, v1, v2)
```

### Arguments

v1 first parameter v2 second parameter

### Value

Matrix

cauchy\_lddd 39

cauch	y_lddd	Third derivative tensor of the normalized log-likelihood

### Description

Third derivative tensor of the normalized log-likelihood

# Usage

```
cauchy_lddd(x, v1, d1, v2, fd2)
```

# Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

### Value

Cubic scalar array

cauchy_lddda	
--------------	--

### Description

The third derivative of the normalized log-likelihood

# Usage

```
cauchy_lddda(x, v1, v2)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter

### Value

3d array

40 cauchy\_lmnp

cauchy_lmn	One component of the second derivative of the normalized log-likelihood
------------	---

# Description

One component of the second derivative of the normalized log-likelihood

### Usage

```
cauchy_lmn(x, v1, d1, v2, fd2, mm, nn)
```

# Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

### Value

Scalar value

cauchy_lmnp	One component of the third derivative of the normalized log-likelihood

# Description

One component of the third derivative of the normalized log-likelihood

# Usage

```
cauchy_lmnp(x, v1, d1, v2, fd2, mm, nn, rr)
```

cauchy\_logf 41

# Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

cauchy_logf $Logf for RUST$
-----------------------------

# Description

Logf for RUST

### Usage

```
cauchy_logf(params, x)
```

# Arguments

params model parameters for calculating logf x a vector of training data values

### Value

Scalar value.

42 cauchy\_logfddd

cauchy_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
cauchy_logfdd(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Matrix

cauchy_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

### Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
cauchy_logfddd(x, v1, v2)
```

### **Arguments**

X	a vector of training	data values
X	a vector of training	data varue

v1 first parameter v2 second parameter

### Value

3d array

cauchy\_loglik 43

ood function	
--------------	--

# Description

log-likelihood function

# Usage

```
cauchy_loglik(vv, x)
```

### Arguments

vv parameters

x a vector of training data values

### Value

Scalar value.

cauchy_logscores	Log scores for MLE and RHP predictions calculated using leave-one- out
------------------	---

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

### Usage

```
cauchy_logscores(logscores, x, d1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

### Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
х	a vector of training data values
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

Two scalars

cauchy\_mu2f

cauchy_mu1f	DMGS equation 3.3, mu1 term
-------------	-----------------------------

# Description

DMGS equation 3.3, mu1 term

### Usage

```
cauchy_mu1f(alpha, v1, d1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Matrix

cauchy_mu2f DMGS equation 3.3, mu2 term
---

# Description

DMGS equation 3.3, mu2 term

### Usage

```
cauchy_mu2f(alpha, v1, d1, v2, fd2)
```

### **Arguments**

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

cauchy\_p1f 45

term

### Value

3d array

cauchy_p1f	DMGS equation 3.3, p1
caucity_pri	DMOS equation 5.5, pr

### Description

DMGS equation 3.3, p1 term

### Usage

```
cauchy_p1f(y, v1, d1, v2, fd2)
```

### **Arguments**

у	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Matrix

cauchy_p1_cp	Cauchy Distribution with a Predictor, Predictions Based on a Cali-
	brating Prior

### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

• q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.

- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qcauchy_p1_cp(
  Х,
  t,
  t0 = NA,
 n0 = NA,
 p = seq(0.1, 0.9, 0.1),
 d1 = 0.01,
  d2 = 0.01,
  fd3 = 0.01,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 predictordata = TRUE,
  centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
rcauchy_p1_cp(
  n,
  Х,
  t,
  t0 = NA,
 n0 = NA,
 d1 = 0.01,
  d2 = 0.01,
  fd3 = 0.01,
  rust = FALSE,
 mlcp = TRUE,
  debug = FALSE,
  aderivs = TRUE
```

```
47
cauchy_p1_cp
    )
    dcauchy_p1_cp(
      Х,
      t,
      t0 = NA,
      n0 = NA,
      y = x,
      d1 = 0.01,
      d2 = 0.01,
      fd3 = 0.01,
      rust = FALSE,
      nrust = 1000,
      centering = TRUE,
      debug = FALSE,
      aderivs = TRUE
    )
    pcauchy_p1_cp(
      Х,
      t,
      t0 = NA,
      n0 = NA,
      y = x,
      d1 = 0.01,
      d2 = 0.01,
      fd3 = 0.01,
      rust = FALSE,
      nrust = 1000,
      centering = TRUE,
      debug = FALSE,
      aderivs = TRUE
    )
    tcauchy_p1_cp(n, x, t, d1 = 0.01, d2 = 0.01, fd3 = 0.01, debug = FALSE)
Arguments
                     a vector of training data values
    Χ
                     a vector of predictors, such that length(t)=length(x)
    t
                      a single value of the predictor (specify either t0 or n0 but not both)
    t0
                      an index for the predictor (specify either t0 or n0 but not both)
    n0
                      a vector of probabilities at which to generate predictive quantiles
    р
    d1
                     if aderivs=FALSE, the delta used for numerical derivatives with respect to the
                      first parameter
```

if aderivs=FALSE, the delta used for numerical derivatives with respect to the

d2

second parameter

fd3	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the third parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- ullet adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

• ml\_params: maximum likelihood estimates for the parameters.

- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

### **Details of the Model**

The Cauchy distribution with a predictor has probability density function

$$f(x; a, b, \sigma) = \frac{1}{\pi \sigma} \left( 1 + \left( \frac{x - \mu(a, b)}{\sigma} \right)^2 \right)^{-1}$$

where x is the random variable,  $\mu = a + bt$  is the location parameter as a function of parameters a, b, and  $\sigma > 0$  is the scale parameter.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

#### If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

#### If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

### If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### Examples

```
#
# example 1
x=fitdistcp::d64cauchy_p1_example_data_v1_x
tt=fitdistcp::d64cauchy_p1_example_data_v1_t
p=c(1:9)/10
n0=10
```

cauchy\_p1\_f1f 53

```
q=qcauchy_p1_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qcauchy_p1_cp)",
main="Cauchy w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

cauchy\_p1\_f1f

DMGS equation 2.1, f1 term

### Description

DMGS equation 2.1, f1 term

### Usage

```
cauchy_p1_f1f(y, t0, v1, d1, v2, d2, v3, fd3)
```

### Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Matrix

54 cauchy\_p1\_f2f

cauchy	/ n1	f1	fa
Caucii	V - V		16

The first derivative of the density

### Description

The first derivative of the density

### Usage

```
cauchy_p1_f1fa(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

cauchy\_p1\_f2f

DMGS equation 2.1, f2 term

# Description

DMGS equation 2.1, f2 term

### Usage

```
cauchy_p1_f2f(y, t0, v1, d1, v2, d2, v3, fd3)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

cauchy\_p1\_f2fa 55

### Value

3d array

cauchy\_p1\_f2fa

The second derivative of the density

### **Description**

The second derivative of the density

### Usage

```
cauchy_p1_f2fa(x, t, v1, v2, v3)
```

### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

cauchy\_p1\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
cauchy_p1_fd(x, t, v1, v2, v3)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

56 cauchy\_p1\_ldd

### Value

Vector

cauchy_p1_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
cauchy_p1_fdd(x, t, v1, v2, v3)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

Matrix

cauc	hy_	p1.	_ldd
------	-----	-----	------

Second derivative matrix of the normalized log-likelihood

### Description

Second derivative matrix of the normalized log-likelihood

### Usage

```
cauchy_p1_ldd(x, t, v1, d1, v2, d2, v3, fd3)
```

cauchy\_p1\_ldda 57

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

Square scalar matrix

cauchy_p1_ldda	
----------------	--

# Description

The second derivative of the normalized log-likelihood

# Usage

```
cauchy_p1_ldda(x, t, v1, v2, v3)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

58 cauchy\_p1\_lddda

	- 4		
cauchy	/ ni	I٨	ุกก

Third derivative tensor of the normalized log-likelihood

### **Description**

Third derivative tensor of the normalized log-likelihood

### Usage

```
cauchy_p1_1ddd(x, t, v1, d1, v2, d2, v3, fd3)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Cubic scalar array

_		
cauchv	n1	ahhh [

The third derivative of the normalized log-likelihood

### Description

The third derivative of the normalized log-likelihood

# Usage

```
cauchy_p1_lddda(x, t, v1, v2, v3)
```

### **Arguments**

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

cauchy\_p1\_lmn 59

### Value

3d array

cauchy_p1_lmn	One component of the second derivative of the normalized log-likelihood
---------------	---

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
cauchy_p1_lmn(x, t, v1, d1, v2, d2, v3, fd3, mm, nn)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

### Value

Scalar value

60 cauchy\_p1\_logf

cauchy_p1_lmnp	One component of the second derivative of the normalized log-likelihood
----------------	---

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
cauchy_p1_lmnp(x, t, v1, d1, v2, d2, v3, fd3, mm, nn, rr)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

cauchy_p1_logf	Logf for RUST
----------------	---------------

# Description

Logf for RUST

# Usage

```
cauchy_p1_logf(params, x, t)
```

cauchy\_p1\_logfdd 61

# Arguments

params	model parameters for calculating logf
X	a vector of training data values
t	a vector or matrix of predictors

### Value

Scalar value.

cauchy_p1_logfdd Second derivative of the log density Created by Stephen Jewson us Deriv() by Andrew Clausen and Serguei Sokol
---

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
cauchy_p1_logfdd(x, t, v1, v2, v3)
```

### Arguments

х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

62 cauchy\_p1\_loglik

cauchy_p1_logfddd Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol	,
---	---

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
cauchy_p1_logfddd(x, t, v1, v2, v3)
```

### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter

v2 second parameter v3 third parameter

### Value

3d array

cauchy\_p1\_loglik

Cauchy-with-p1 observed log-likelihood function

### Description

Cauchy-with-p1 observed log-likelihood function

### Usage

```
cauchy_p1_loglik(vv, x, t)
```

### Arguments

VV	parameters

x a vector of training data valuest a vector or matrix of predictors

#### Value

Scalar value.

cauchy\_p1\_logscores 63

cauchy_p1_logscores	Log scores for MLE and RHP predictions calculated using leave-one- out
	out

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

### Usage

```
cauchy_p1_logscores(logscores, x, t, d1, d2, fd3, aderivs = TRUE)
```

### Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
X	a vector of training data values
t	a vector or matrix of predictors
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

Two scalars

auchy_p1_means Cauchy distribution: RHP mean
--

# Description

Cauchy distribution: RHP mean

# Usage

```
cauchy_p1_means(t0, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2)
```

cauchy\_p1\_mu1f

### Arguments

t0 a single value of the predictor (specify either t0 or n0 but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data
dim number of parameters

### Value

Two scalars

cauchy\_p1\_mu1f

DMGS equation 3.3, mu1 term

### Description

DMGS equation 3.3, mu1 term

### Usage

```
cauchy_p1_mu1f(alpha, t0, v1, d1, v2, d2, v3, fd3)
```

### **Arguments**

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Matrix

cauchy\_p1\_mu2f 65

cauchy_	n1	mu2f	
caucity_	_P ' .	_IIIU∠I	

DMGS equation 3.3, mu2 term

# Description

DMGS equation 3.3, mu2 term

# Usage

```
cauchy_p1_mu2f(alpha, t0, v1, d1, v2, d2, v3, fd3)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

3d array

cauchy_	ք1	p1f
caaciiy_	- 12	_P ' '

DMGS equation 2.1, p1 term

# Description

```
DMGS equation 2.1, p1 term
```

# Usage

```
cauchy_p1_p1f(y, t0, v1, d1, v2, d2, v3, fd3)
```

cauchy\_p1\_p2f

# Arguments

у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Matrix

DMGS equation 2.1, p2 term
----------------------------

# Description

DMGS equation 2.1, p2 term

# Usage

```
cauchy_p1_p2f(y, t0, v1, d1, v2, d2, v3, fd3)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

3d array

```
cauchy_p1_predictordata
```

Predicted Parameter and Generalized Residuals

### **Description**

Predicted Parameter and Generalized Residuals

### Usage

```
cauchy_p1_predictordata(predictordata, x, t, t0, params)
```

### Arguments

predictordata logical that indicates whether to calculate and return predictordata x a vector of training data values t a vector or matrix of predictors to a single value of the predictor (specify either to or no but not both) params model parameters for calculating logf

### Value

Two vectors

cauchy\_p1\_waic Waic

### Description

Waic

### Usage

```
cauchy_p1_waic(
  waicscores,
  x,
  t,
  v1hat,
  d1,
  v2hat,
  d2,
  v3hat,
  fd3,
  lddi,
```

68 cauchy\_p2f

```
lddd,
lambdad,
aderivs
)
```

### Arguments

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime) a vector of training data values Χ t a vector or matrix of predictors v1hat first parameter d1 the delta used in the numerical derivatives with respect to the parameter v2hat second parameter d2 the delta used in the numerical derivatives with respect to the parameter v3hat third parameter fd3 the fractional delta used in the numerical derivatives with respect to the parameter lddi inverse observed information matrix lddd third derivative of log-likelihood lambdad derivative of the log prior

logical for whether to use analytic derivatives (instead of numerical)

### Value

aderivs

Two numeric values.

cauchy_p2f DMGS equation 3.3, p2 term	
---------------------------------------	--

### **Description**

DMGS equation 3.3, p2 term

### Usage

```
cauchy_p2f(y, v1, d1, v2, fd2)
```

### Arguments

у	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

cauchy\_waic 69

### Value

3d array

# Description

Waic

### Usage

```
cauchy_waic(waicscores, x, v1hat, d1, v2hat, fd2, lddi, lddd, lambdad, aderivs)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
Χ	a vector of training data values
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

Two numeric values.

```
crhpflat_dmgs_cpmethod
```

Generates a comment about the method

### Description

Generates a comment about the method

### Usage

```
crhpflat_dmgs_cpmethod()
```

### Value

String

```
d100gamma_example_data_v1
```

This is data to be included in my package

### Description

This is data to be included in my package

```
d101invgamma_example_data_v1
```

This is data to be included in my package

# Description

This is data to be included in my package

```
d102invgauss_example_data_v1
```

This is data to be included in my package

### Description

This is data to be included in my package

d105burr\_example\_data\_v1

This is data to be included in my package

### **Description**

This is data to be included in my package

d10exp\_example\_data\_v1

This is data to be included in my package

### Description

This is data to be included in my package

d110gev\_example\_data\_v1

This is data to be included in my package

### Description

This is data to be included in my package

d11pareto\_k2\_example\_data\_v1

This is data to be included in my package

### Description

This is data to be included in my package

d120gpd\_k1\_example\_data\_v1

This is data to be included in my package

### **Description**

This is data to be included in my package

```
d150gev_p1_example_data_v1_t
```

This is data to be included in my package

### **Description**

This is data to be included in my package

```
d150gev_p1_example_data_v1_x
```

This is data to be included in my package

### Description

This is data to be included in my package

```
d151gev_p12_example_data_v1_t
```

This is data to be included in my package

### Description

This is data to be included in my package

```
d151gev_p12_example_data_v1_x
```

This is data to be included in my package

### Description

This is data to be included in my package

```
d152gev_p123_example_data_v1_t
```

This is data to be included in my package

### Description

This is data to be included in my package

 $d152 gev\_p123\_example\_data\_v1\_x$ 

This is data to be included in my package

### **Description**

This is data to be included in my package

d20halfnorm\_example\_data\_v1

This is data to be included in my package

# Description

This is data to be included in my package

d25unif\_example\_data\_v1

This is data to be included in my package

### **Description**

This is data to be included in my package

d30norm\_example\_data\_v1

This is data to be included in my package

## Description

This is data to be included in my package

d31norm\_dmgs\_example\_data\_v1

This is data to be included in my package

### **Description**

d32gnorm\_k3\_example\_data\_v1

This is data to be included in my package

## **Description**

This is data to be included in my package

d35lnorm\_example\_data\_v1

This is data to be included in my package

## Description

This is data to be included in my package

d36lnorm\_dmgs\_example\_data\_v1

This is data to be included in my package

## Description

This is data to be included in my package

d40logis\_example\_data\_v1

This is data to be included in my package

# Description

This is data to be included in my package

d41lst\_k3\_example\_data\_v1

This is data to be included in my package

### **Description**

d42cauchy\_example\_data\_v1

This is data to be included in my package

### **Description**

This is data to be included in my package

d50gumbel\_example\_data\_v1

This is data to be included in my package

## Description

This is data to be included in my package

 $d51frechet\_k1\_example\_data\_v1$ 

This is data to be included in my package

### **Description**

This is data to be included in my package

d52weibull\_example\_data\_v1

This is data to be included in my package

# Description

This is data to be included in my package

d53gev\_k3\_example\_data\_v1

This is data to be included in my package

### **Description**

```
d55exp_p1_example_data_v1_t
```

This is data to be included in my package

### **Description**

This is data to be included in my package

```
d55exp_p1_example_data_v1_x
```

This is data to be included in my package

## Description

This is data to be included in my package

```
d56pareto_p1k2_example_data_v1_t
```

This is data to be included in my package

## Description

This is data to be included in my package

```
d56pareto_p1k2_example_data_v1_x
```

This is data to be included in my package

# Description

This is data to be included in my package

```
d60norm_p1_example_data_v1_t
```

This is data to be included in my package

### **Description**

d60norm\_p1\_example\_data\_v1\_x

This is data to be included in my package

## **Description**

This is data to be included in my package

d61lnorm\_p1\_example\_data\_v1\_t

This is data to be included in my package

## Description

This is data to be included in my package

d61lnorm\_p1\_example\_data\_v1\_x

This is data to be included in my package

## Description

This is data to be included in my package

d62logis\_p1\_example\_data\_v1\_t

This is data to be included in my package

# Description

This is data to be included in my package

d62logis\_p1\_example\_data\_v1\_x

This is data to be included in my package

### **Description**

```
d63lst_p1k3_example_data_v1_t
```

This is data to be included in my package

### **Description**

This is data to be included in my package

```
d63lst_p1k3_example_data_v1_x
```

This is data to be included in my package

## Description

This is data to be included in my package

```
d64cauchy_p1_example_data_v1_t
```

This is data to be included in my package

## Description

This is data to be included in my package

```
d64cauchy_p1_example_data_v1_x
```

This is data to be included in my package

# Description

This is data to be included in my package

```
d70gumbel_p1_example_data_v1_t
```

This is data to be included in my package

### **Description**

d70gumbel\_p1\_example\_data\_v1\_x

This is data to be included in my package

## **Description**

This is data to be included in my package

d71frechet\_p2k1\_example\_data\_v1\_t

This is data to be included in my package

## Description

This is data to be included in my package

 $d71frechet\_p2k1\_example\_data\_v1\_x$ 

This is data to be included in my package

### **Description**

This is data to be included in my package

 $d72 weibull\_p1\_example\_data\_v1\_t$ 

This is data to be included in my package

## Description

This is data to be included in my package

 ${\sf d72weibull\_p1\_example\_data\_v1\_x}$ 

This is data to be included in my package

### **Description**

```
d73weibull_p2_example_data_v1_t
```

This is data to be included in my package

## Description

This is data to be included in my package

```
d73weibull_p2_example_data_v1_x
```

This is data to be included in my package

## Description

This is data to be included in my package

```
d74gev_p1k3_example_data_v1_t
```

This is data to be included in my package

## Description

This is data to be included in my package

```
d74gev_p1k3_example_data_v1_x
```

This is data to be included in my package

# Description

This is data to be included in my package

```
d80norm_p12_example_data_v1_t1
```

This is data to be included in my package

### **Description**

d80norm\_p12\_example\_data\_v1\_t2

This is data to be included in my package

## **Description**

This is data to be included in my package

d80norm\_p12\_example\_data\_v1\_x

This is data to be included in my package

# Description

This is data to be included in my package

d81lst\_p12k3\_example\_data\_v1\_t1

This is data to be included in my package

## Description

This is data to be included in my package

d81lst\_p12k3\_example\_data\_v1\_t2

This is data to be included in my package

# Description

This is data to be included in my package

d81lst\_p12k3\_example\_data\_v1\_x

This is data to be included in my package

### **Description**

82 dcauchysub

```
d82weibull_p12_example_data_v1_t1
```

This is data to be included in my package

## Description

This is data to be included in my package

```
d82weibull_p12_example_data_v1_t2
```

This is data to be included in my package

### **Description**

This is data to be included in my package

```
d82weibull_p12_example_data_v1_x
```

This is data to be included in my package

## Description

This is data to be included in my package

dcauchysub

Densities from MLE and RHP

### **Description**

Densities from MLE and RHP

### Usage

```
dcauchysub(x, y, d1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

X	a vector of training data values
у	a vector of values at which to calculate the density and distribution functions
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

dcauchy\_p1 83

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dcauchy_p1 Cauchy-with-p1 density function	
--	--

# Description

Cauchy-with-p1 density function

## Usage

```
dcauchy_p1(x, t0, ymn, slope, scale, log = FALSE)
```

# Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
scale	the scale parameter of the distribution
log	logical for the density evaluation

## Value

Vector

Densities from MLE and RHP	uchy_p1sub
Densines from MLE and RIII	uchy_proub

# Description

Densities from MLE and RHP

# Usage

```
dcauchy_p1sub(x, t, y, t0, d1, d2, fd3, aderivs = TRUE)
```

84 deriv\_copyfdd

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

deriv_copyfdd Extract the results from derivatives and put them into f2	
---	--

# Description

Extract the results from derivatives and put them into f2

# Usage

```
deriv_copyfdd(temp1, nx, dim)
```

## Arguments

temp1 output from derivative calculations

 $\begin{array}{ll} \text{nx} & \text{number of x values} \\ \\ \text{dim} & \text{number of parameters} \end{array}$ 

## Value

3d array

deriv\_copyld2 85

deriv	copyld2

Extract the results from derivatives and put them into ldd

## Description

Extract the results from derivatives and put them into ldd

## Usage

```
deriv_copyld2(temp1, nx, dim)
```

### **Arguments**

temp1 output from derivative calculations

nx number of x values
dim number of parameters

### Value

3d array

deriv\_copyldd

Extract the results from derivatives and put them into ldd

## Description

Extract the results from derivatives and put them into ldd

# Usage

```
deriv_copyldd(temp1, nx, dim)
```

## Arguments

temp1 output from derivative calculations

nx number of x values
dim number of parameters

## Value

Matrix

86 dexpsub

deriv\_copylddd

Extract the results from derivatives and put them into lddd

## Description

Extract the results from derivatives and put them into lddd

## Usage

```
deriv_copylddd(temp1, nx, dim)
```

## Arguments

temp1 output from derivative calculations

nx number of x values dim number of parameters

#### Value

3d array

dexpsub

Densities from MLE and RHP

## Description

Densities from MLE and RHP

## Usage

```
dexpsub(x, y, aderivs = TRUE)
```

### **Arguments**

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

aderivs logical for whether to use analytic derivatives (instead of numerical)

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dexp\_p1 87

# Description

Exponential-with-p1 density function

## Usage

```
dexp_p1(x, t0, ymn, slope, log = FALSE)
```

## Arguments

x a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

log logical for the density evaluation

## Value

Vector

dexp_p1sub	Densities from MLE and RHP

## **Description**

Densities from MLE and RHP

## Usage

```
dexp_p1sub(x, t, y, t0, d1, d2, aderivs = TRUE)
```

X	a vector of training data values
t	a vector or matrix of predictors
У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

88 dfrechet\_p2k1

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

tsub Densities from MLE and RH
--------------------------------

# Description

Densities from MLE and RHP

## Usage

```
dfrechetsub(x, y, kloc, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

## **Arguments**

X	a vector of training data values
у	a vector of values at which to calculate the density and distribution functions
kloc	the known location parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

# Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

Frechet_k1-with-p2 density function
-------------------------------------

# Description

Frechet\_k1-with-p2 density function

# Usage

```
dfrechet_p2k1(x, t0, ymn, slope, lambda, log = FALSE, kloc)
```

dfrechet\_p2k1sub

#### **Arguments**

x a vector of training data values	,
------------------------------------	---

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor lambda the lambda parameter of the distribution

log logical for the density evaluation kloc the known location parameter

### Value

Vector

## Description

Densities from MLE and RHP

### Usage

```
dfrechet_p2k1sub(x, t, y, t0, d1, d2, fd3, kloc, aderivs = TRUE)
```

### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

90 dgevsub

dgamması	ub
----------	----

Densities from MLE and RHP

# Description

Densities from MLE and RHP

## Usage

```
dgammasub(x, y, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

### **Arguments**

X	a vector of training data values
у	a vector of values at which to calculate the density and distribution functions
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgevsub

Densities for 5 predictions

## Description

Densities for 5 predictions

# Usage

```
dgevsub(
    x,
    y,
    ics,
    d1 = 0.01,
    fd2 = 0.01,
    d3 = 0.01,
    customprior,
    minxi,
    maxxi,
    extramodels = FALSE,
    aderivs = TRUE
)
```

dgev\_k3sub

# Arguments

Χ	a vector of training data values
у	a vector of values at which to calculate the density and distribution functions
ics	initial conditions for the maximum likelihood search
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
d3	the delta used in the numerical derivatives with respect to the parameter
customprior	a custom value for the slope of the log prior at the maxlik estimate
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi
extramodels	logical that indicates whether to add three additional prediction models
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgev_k3sub	Densities from MLE and RHP	

# Description

Densities from MLE and RHP

# Usage

```
dgev_k3sub(x, y, d1 = 0.01, fd2 = 0.01, kshape, aderivs = TRUE)
```

## Arguments

Х	a vector of training data values
у	a vector of values at which to calculate the density and distribution functions
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

92 dgev\_p12

dgev\_p1

GEVD-with-p1: Density function

# Description

GEVD-with-p1: Density function

## Usage

```
dgev_p1(x, t0, ymn, slope, sigma, xi, log = FALSE)
```

## Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either $t0$ or $n0$ but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma	the sigma parameter of the distribution
xi	the shape parameter of the distribution
log	logical for the density evaluation

## Value

Vector

dgev\_p12

GEVD-with-p1: Density function

# Description

GEVD-with-p1: Density function

# Usage

```
dgev_p12(x, t1, t2, ymn, slope, sigma1, sigma2, xi, log = FALSE)
```

dgev\_p123 93

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

sigma1 first coefficient for the sigma parameter of the distribution sigma2 second coefficient for the sigma parameter of the distribution

xi the shape parameter of the distribution log logical for the density evaluation

### Value

Vector

dgev\_p123 GEVD-with-p1: Density function

### **Description**

GEVD-with-p1: Density function

### Usage

```
dgev_p123(x, t1, t2, t3, ymn, slope, sigma1, sigma2, xi1, xi2, log = FALSE)
```

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

sigma1 first coefficient for the sigma parameter of the distribution
sigma2 second coefficient for the sigma parameter of the distribution
xi1 first coefficient for the shape parameter of the distribution
xi2 second coefficient for the shape parameter of the distribution

log logical for the density evaluation

#### Value

Vector

94 dgev\_p123sub

dgev\_p123sub

Densities for 5 predictions

## Description

Densities for 5 predictions

# Usage

```
dgev_p123sub(
 х,
  t1,
  t2,
  t3,
 у,
  t01,
  t02,
  t03,
  ics,
 d1 = 0.01,
 d2 = 0.01,
 d3 = 0.01,
  d4 = 0.01,
  d5 = 0.01,
 d6 = 0.01,
 extramodels,
 debug,
  aderivs = TRUE
)
```

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
у	a vector of values at which to calculate the density and distribution functions
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
ics	initial conditions for the maximum likelihood search
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
d3	the delta used in the numerical derivatives with respect to the parameter

dgev\_p12sub

d4	the delta used in the numerical derivatives with respect to the parameter
d5	the delta used in the numerical derivatives with respect to the parameter
d6	the delta used in the numerical derivatives with respect to the parameter
extramodels	logical that indicates whether to add three additional prediction models
debug	debug flag
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgev\_p12sub

Densities for 5 predictions

## Description

Densities for 5 predictions

## Usage

```
dgev_p12sub(
 Х,
  t1,
  t2,
 у,
  t01,
  t02,
  ics,
 d1 = 0.01,
 d2 = 0.01,
 d3 = 0.01,
 d4 = 0.01,
 d5 = 0.01,
 minxi,
 maxxi,
 debug,
 extramodels = FALSE,
 aderivs = TRUE
)
```

```
    x a vector of training data values
    t1 a vector of predictors for the mean
    t2 a vector of predictors for the sd
```

96 dgev\_p1k3

a vector of values at which to calculate the density and distribution functions

~	·
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
ics	initial conditions for the maximum likelihood search
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
d3	the delta used in the numerical derivatives with respect to the parameter
d4	the delta used in the numerical derivatives with respect to the parameter
d5	the delta used in the numerical derivatives with respect to the parameter
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi
debug	debug flag
extramodels	logical that indicates whether to add three additional prediction models
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

У

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgev_p1k3	GEV-with-known-shape-with-p1 density function	

# Description

GEV-with-known-shape-with-p1 density function

## Usage

```
dgev_p1k3(x, t0, ymn, slope, sigma, log = FALSE, kshape)
```

# Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma	the sigma parameter of the distribution
log	logical for the density evaluation
kshape	the known shape parameter

## Value

Vector

dgev\_p1k3sub

b Densities from MLE and RHP

# Description

Densities from MLE and RHP

# Usage

```
dgev_p1k3sub(x, t, y, t0, d1, d2, fd3, kshape, aderivs = TRUE)
```

# Arguments

x	a vector of training data values
t	a vector or matrix of predictors
у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

|--|

# Description

Densities for 5 predictions

98 dgev\_p1sub

# Usage

```
dgev_p1sub(
    x,
    t,
    y,
    t0,
    ics,
    d1 = 0.01,
    d2 = 0.01,
    fd3 = 0.01,
    d4 = 0.01,
    minxi,
    maxxi,
    extramodels = FALSE,
    aderivs = TRUE
)
```

# Arguments

х	a vector of training data values
t	a vector or matrix of predictors
у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
ics	initial conditions for the maximum likelihood search
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
d4	the delta used in the numerical derivatives with respect to the parameter
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi
extramodels	logical that indicates whether to add three additional prediction models
aderivs	logical for whether to use analytic derivatives (instead of numerical)

# Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgnorm\_k3sub

dgnorm	k3sub

Densities from MLE and RHP

## Description

Densities from MLE and RHP

## Usage

```
dgnorm_k3sub(x, y, d1 = 0.01, fd2 = 0.01, kbeta, aderivs = TRUE)
```

## Arguments

Χ	a vector of training data values
У	a vector of values at which to calculate the density and distribution functions
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter
kbeta	the known beta parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgpdsub

Densities for 5 predictions

## Description

Densities for 5 predictions

# Usage

```
dgpdsub(
    x,
    y,
    ics,
    fd1 = 0.01,
    d2 = 0.01,
    kloc = 0,
    dlogpi = 0,
    minxi,
    maxxi,
    extramodels = FALSE,
    aderivs = TRUE
)
```

100 dgumbelsub

# Arguments

X	a vector of training data values
У	a vector of values at which to calculate the density and distribution functions
ics	initial conditions for the maximum likelihood search
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter
dlogpi	gradient of the log prior
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi
extramodels	logical that indicates whether to add three additional prediction models
aderivs	logical for whether to use analytic derivatives (instead of numerical)

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

|--|

# Description

Densities from MLE and RHP

# Usage

```
dgumbelsub(x, y, d1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

# Arguments

X	a vector of training data values
У	a vector of values at which to calculate the density and distribution functions
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgumbel\_p1 101

dgumbel_p1	Gumbel-with-p1 density function

## Description

Gumbel-with-p1 density function

## Usage

```
dgumbel_p1(x, t0, ymn, slope, sigma, log = FALSE)
```

## Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma	the sigma parameter of the distribution
log	logical for the density evaluation

## Value

Vector

dgumbel_p1sub	Densities from MLE and RHP	

# Description

Densities from MLE and RHP

# Usage

```
dgumbel_p1sub(x, t, y, t0, d1, d2, fd3, aderivs = TRUE)
```

X	a vector of training data values
t	a vector or matrix of predictors
У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

102 dinvgammasub

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dhalfnormsub	Densi
unatinoniisub	Densi

Densities from MLE and RHP

## **Description**

Densities from MLE and RHP

#### Usage

```
dhalfnormsub(x, y, fd1 = 0.01, aderivs = TRUE)
```

## Arguments

x	a vector of training data values
у	a vector of values at which to calculate the density and distribution functions

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

aderivs logical for whether to use analytic derivatives (instead of numerical)

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

Densities from MLE and cp
---------------------------

## Description

Densities from MLE and cp

## Usage

```
dinvgammasub(x, y, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

Χ	a vector of training data values
У	a vector of values at which to calculate the density and distribution functions
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

dinvgausssub 103

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

es from MLE and RHP	Densities	dinvgausssub
---------------------	-----------	--------------

# Description

Densities from MLE and RHP

# Usage

```
dinvgausssub(x, y, prior, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

## Arguments

Х	a vector of training data values
у	a vector of values at which to calculate the density and distribution functions
prior	logical indicating which prior to use
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

# Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

|--|

## Description

Densities from MLE and RHP

# Usage

```
dlnormsub(x, y, aderivs = TRUE)
```

X	a vector of training data values
У	a vector of values at which to calculate the density and distribution functions
aderivs	logical for whether to use analytic derivatives (instead of numerical)

dlnorm\_p1

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dlnorm_dmgssub	Densities from MLE and RHF

## Description

Densities from MLE and RHP

## Usage

```
dlnorm\_dmgssub(x, y, d1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

## Arguments

X	a vector of training data values
У	a vector of values at which to calculate the density and distribution functions
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dlnorm_p1	Normal-with-p1 density function	

# Description

Normal-with-p1 density function

## Usage

```
dlnorm_p1(x, t0, ymn, slope, sigma, log = FALSE)
```

Х	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma	the sigma parameter of the distribution
log	logical for the density evaluation
_	•

dlnorm\_p1sub

## Value

Vector

dlnorm_p1sub	Densities from MLE and RHP	

## Description

Densities from MLE and RHP

## Usage

```
dlnorm_p1sub(x, t, y, t0, debug = FALSE, aderivs = TRUE)
```

## Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
debug	debug flag
aderivs	logical for whether to use analytic derivatives (instead of numerical)

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dlogis2sub	Densities from MLE and RHP	

## Description

Densities from MLE and RHP

# Usage

```
dlogis2sub(x, y, d1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

X	a vector of training data values
У	a vector of values at which to calculate the density and distribution functions
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

106 dlogis\_p1sub

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dlogis\_p1

Logistic-with-p1 density function

### **Description**

Logistic-with-p1 density function

## Usage

```
dlogis_p1(x, t0, ymn, slope, scale, log = FALSE)
```

### **Arguments**

Y	a vector of	of training	data values
Λ	a vector t	oi uanime	uata varues

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor
scale the scale parameter of the distribution
log logical for the density evaluation

## Value

Vector

dlogis\_p1sub

Densities from MLE and RHP

# Description

Densities from MLE and RHP

### Usage

```
dlogis_p1sub(x, t, y, t0, d1, d2, fd3, aderivs = TRUE)
```

dlst\_k3sub 107

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dist_k3sub Densines from MLE and KHF	dlst_k3sub	Densities from MLE and RHP	
--------------------------------------	------------	----------------------------	--

# Description

Densities from MLE and RHP

# Usage

```
dlst_k3sub(x, y, d1 = 0.01, fd2 = 0.01, kdf, aderivs = TRUE)
```

# Arguments

y yearton of volves at which to coloulate the density and distribution functions
y a vector of values at which to calculate the density and distribution functions
d1 the delta used in the numerical derivatives with respect to the parameter
the fractional delta used in the numerical derivatives with respect to the parameter
kdf the known degrees of freedom parameter
aderivs logical for whether to use analytic derivatives (instead of numerical)

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

108 dlst\_p1k3sub

41	st_	n1	レコ
uт	. S L_	_D I	ĸэ

LST-with-p1 density function

# Description

LST-with-p1 density function

# Usage

```
dlst_p1k3(x, t0, ymn, slope, sigma, log = FALSE, kdf)
```

# Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma	the sigma parameter of the distribution
log	logical for the density evaluation
kdf	the known degrees of freedom parameter

## Value

Vector

dlst	n1	k3sub
UISC.	_P '	KJSUD

Densities from MLE and RHP

# Description

Densities from MLE and RHP

# Usage

```
dlst_p1k3sub(x, t, y, t0, d1, d2, fd3, kdf, aderivs = TRUE)
```

dmgs 109

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)
	the known degrees of freedom parameter

# Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

|--|

# Description

Evaluate DMGS equation 3.3

## Usage

```
dmgs(lddi, lddd, mu1, pidopi, mu2, dim)
```

## Arguments

lddi	inverse of second derivative of observed log-likelihood
lddd	third derivative of observed log-likelihood
mu1	DMGS mu1 vector
pidopi	derivative of log prior
mu2	DMGS mu2 matrix
dim	number of parameters

### Value

Vector

110 dnorm\_dmgssub

d	ทด	rm	เรเ	ıh

Densities from MLE and RHP

# Description

Densities from MLE and RHP

## Usage

```
dnormsub(x, y, aderivs = TRUE)
```

## Arguments

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dnorm_dmgssub	Densities from MLE and RHP
---------------	----------------------------

### **Description**

Densities from MLE and RHP

# Usage

```
dnorm\_dmgssub(x, y, d1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

## Arguments

X	a vector of training data values
у	a vector of values at which to calculate the density and distribution functions
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

dnorm\_p1 111

dnorm_p1	Normal-with-p1 density function	

## Description

Normal-with-p1 density function

# Usage

```
dnorm_p1(x, t0, ymn, slope, sigma, log = FALSE)
```

### **Arguments**

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma	the sigma parameter of the distribution

sigma the sigma parameter of the distribution log logical for the density evaluation

## Value

Vector

|--|

## Description

Densities from MLE and RHP

## Usage

```
dnorm_p1sub(x, t, y, t0, aderivs = TRUE)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

dpareto\_k2\_sub

dnorm\_p1\_formula

Linear regression formula, densities

#### Description

Linear regression formula, densities

#### Usage

```
dnorm_p1_formula(y, ta, ta0, nx, muhat0, v3hat)
```

### **Arguments**

y a vector of values at which to calculate the density and distribution functions

ta predictor residuals

ta0 predictor residual at the point being predicted

nx length of training data

muhat at the point being predicted

v3hat third parameter

#### Value

Vector

dpareto\_k2\_sub

Densities from MLE and RHP

### **Description**

Densities from MLE and RHP

### Usage

```
dpareto_k2_sub(x, y, kscale, aderivs = TRUE)
```

# Arguments

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

kscale the known scale parameter

aderivs logical for whether to use analytic derivatives (instead of numerical)

## Value

dpareto\_p1k2

dpareto_r	1k2
-----------	-----

pareto\_k1-with-p2 density function

### **Description**

```
pareto_k1-with-p2 density function
```

### Usage

```
dpareto_p1k2(x, t0, ymn, slope, kscale, log = FALSE)
```

## Arguments

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

kscale the known scale parameter log logical for the density evaluation

#### Value

Vector

dpareto\_p1k2sub

Densities from MLE and RHP

### **Description**

Densities from MLE and RHP

## Usage

```
dpareto_p1k2sub(x, t, y, t0, d1, d2, kscale, aderivs = TRUE, debug = FALSE)
```

X	a vector of training data values
t	a vector or matrix of predictors
У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kscale	the known scale parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)
debug	debug flag

114 dweibullsub

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dunif\_formula

Predictive PDFs

## Description

Predictive PDFs

### Usage

```
dunif_formula(x, y)
```

### **Arguments**

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

#### Value

Two vectors

dweibullsub

Densities from MLE and RHP

### **Description**

Densities from MLE and RHP

#### Usage

```
dweibullsub(x, y, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

## Arguments

X	a vector of training data values
У	a vector of values at which to calculate the density and distribution functions
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

dweibull\_p2

dweibull_p2	Weibull-with-p1 density function	

## Description

Weibull-with-p1 density function

### Usage

```
dweibull_p2(x, t0, shape, ymn, slope, log = FALSE)
```

## Arguments

t0 a single value of the predictor (specify either t0 or n0 but not both)

shape the shape parameter of the distribution

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

log logical for the density evaluation

### Value

Vector

dweibull_p2sub	Densities from MLE and RHP	

### **Description**

Densities from MLE and RHP

### Usage

```
dweibull_p2sub(x, t, y, t0, fd1, d2, d3, aderivs = TRUE)
```

X	a vector of training data values
t	a vector or matrix of predictors
у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
d3	the delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

exp\_cp

Exponential Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qexp_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    fd1 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE,
    aderivs = TRUE
)
```

```
rexp_cp(n, x, rust = FALSE, mlcp = TRUE, debug = FALSE, aderivs = TRUE)

dexp_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE, aderivs = TRUE)

pexp_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE, aderivs = TRUE)

texp_cp(n, x, debug = FALSE)
```

### **Arguments**

X	a vector of training data values
p	a vector of probabilities at which to generate predictive quantiles
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

• predictedparameter: the estimated value for parameter, as a function of the predictor.

• adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

### **Details of the Model**

The exponential distribution has exceedance distribution function

$$S(x; \lambda) = \exp(-\lambda x)$$

where x > 0 is the random variable and  $\lambda > 0$  is the rate parameter.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\lambda) \propto \frac{1}{\lambda}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

## **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

· Cauchy (cauchy),

- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

122 exp\_f1fa

### **Examples**

```
#
# example 1
x=fitdistcp::d10exp_example_data_v1
p=c(1:9)/10
q=qexp_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qexp_cp)",
main="Exponential: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

exp\_f1f

DMGS equation 2.1, f1 term

### **Description**

DMGS equation 2.1, f1 term

#### **Usage**

```
exp_f1f(y, v1, fd1)
```

### Arguments

y a vector of values at which to calculate the density and distribution functions
v1 first parameter
fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Matrix

exp\_f1fa

The first derivative of the density

### **Description**

The first derivative of the density

#### Usage

```
exp_f1fa(x, v1)
```

exp\_f2f 123

# Arguments

x a vector of training data values

v1 first parameter

### Value

Vector

exp\_f2f

DMGS equation 2.1, f2 term

## Description

DMGS equation 2.1, f2 term

## Usage

## Arguments

y a vector of values at which to calculate the density and distribution functions

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

### Value

3d array

exp\_f2fa

The second derivative of the density

## Description

The second derivative of the density

# Usage

## Arguments

x a vector of training data values

v1 first parameter

124 exp\_fdd

### Value

Matrix

exp\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
exp_fd(x, v1)
```

### **Arguments**

x a vector of training data values

v1 first parameter

#### Value

Vector

exp\_fdd

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
exp_fdd(x, v1)
```

#### **Arguments**

x a vector of training data values

v1 first parameter

#### Value

Matrix

exp\_1111 125

exp\_1111

Third derivative of the normalized log-likelihood

## Description

Third derivative of the normalized log-likelihood

### Usage

```
exp_l111(x, v1, fd1)
```

## Arguments

x a vector of training data values

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

### Value

Scalar value

exp\_ldd

The second derivative of the normalized log-likelihood

## Description

The second derivative of the normalized log-likelihood

### Usage

```
exp_1dd(x, v1, fd1)
```

# **Arguments**

x a vector of training data values

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Square scalar matrix

126 exp\_lddd

exp\_ldda

The second derivative of the normalized log-likelihood

## Description

The second derivative of the normalized log-likelihood

## Usage

```
exp_ldda(x, v1)
```

## Arguments

x a vector of training data values

v1 first parameter

### Value

Matrix

exp\_lddd

Third derivative tensor of the log-likelihood

# Description

Third derivative tensor of the log-likelihood

## Usage

```
exp_1ddd(x, v1, fd1)
```

## **Arguments**

x a vector of training data values

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

### Value

Cubic scalar array

exp\_lddda 127

exp\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

## Usage

```
exp_lddda(x, v1)
```

## Arguments

x a vector of training data values

v1 first parameter

## Value

3d array

exp\_logf

 $Log f for \, RUST$ 

# Description

Logf for RUST

# Usage

```
exp_logf(params, x)
```

## Arguments

params model parameters for calculating logf x a vector of training data values

### Value

Scalar value.

128 exp\_logfddd

exp_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	-

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
exp_logfdd(x, v1)
```

### **Arguments**

x a vector of training data values

v1 first parameter

#### Value

Matrix

exp_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
exp_logfddd(x, v1)
```

### **Arguments**

x a vector of training data values

v1 first parameter

### Value

3d array

exp\_logscores 129

exp_logscores	Log scores for MLE and RHP predictions calculated using leave-one-
	out

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

# Usage

```
exp_logscores(logscores, x)
```

# Arguments

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

#### Value

Two scalars

exp\_p1fa

The first derivative of the cdf

## Description

The first derivative of the cdf

## Usage

```
exp_p1fa(x, v1)
```

# Arguments

x a vector of training data values

v1 first parameter

## Value

Vector

exp\_p1\_cp

Exponential Distribution with a Predictor, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qexp_p1_cp(
    x,
    t,
    t0 = NA,
    n0 = NA,
    p = seq(0.1, 0.9, 0.1),
    d1 = 0.01,
    d2 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    predictordata = TRUE,
```

```
centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
rexp_p1_cp(
 n,
 х,
  t,
  t0 = NA,
 n0 = NA,
 d1 = 0.01,
 d2 = 0.01,
  rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
dexp_p1_cp(
 х,
  t,
 t0 = NA,
 n0 = NA,
 y = x,
 d1 = 0.01,
 d2 = 0.01,
  rust = FALSE,
 nrust = 1000,
  centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
)
pexp_p1_cp(
 Х,
  t,
 t0 = NA,
 n0 = NA,
 y = x,
 d1 = 0.01,
 d2 = 0.01,
  rust = FALSE,
 nrust = 1000,
  centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
```

```
texp_p1_cp(n, x, t, d1 = 0.01, d2 = 0.01, debug = FALSE)
```

# Arguments

x	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
p	a vector of probabilities at which to generate predictive quantiles
d1	the fractional delta used in the numerical derivatives with respect to the location parameter
d2	the fractional delta used in the numerical derivatives with respect to the slope parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

## Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.

- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### Details of the Model

The exponential distribution with a predictor has exceedance distribution function

$$S(x; a, b) = \exp(-x\lambda(a, b))$$

where  $x \ge 0$  is the random variable and  $\lambda(a,b) = e^{-a-bt}$  is the rate parameter, modelled as a function of the parameters a,b and a predictor t.

The calibrating prior is given by the right Haar prior, which is

$$\pi(a,b) \propto 1$$

. as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

 $\exp_p 1_{cp}$ 

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

• Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),

exp\_p1\_f1f

• Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
#
# example 1
x=fitdistcp::d55exp_p1_example_data_v1_x
tt=fitdistcp::d55exp_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qexp_p1_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qexp_p1_cp)",
main="Exponential w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

exp\_p1\_f1f

DMGS equation 2.1, f1 term

### **Description**

DMGS equation 2.1, f1 term

# Usage

```
exp_p1_f1f(y, t0, v1, d1, v2, d2)
```

### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

#### Value

Matrix

 $exp_p1_f2f$ 

exp_	p1	f1	fa

The first derivative of the density

## Description

The first derivative of the density

## Usage

```
exp_p1_f1fa(x, t, v1, v2)
```

# Arguments

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

### Value

Vector

exp_	- 4	C 0 (
evn	nι	ナノナ
$C \wedge D_{-}$	_ 12 1	_ ' _ '

DMGS equation 2.1, f2 term

## Description

DMGS equation 2.1, f2 term

# Usage

## Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

# Value

3d array

exp\_p1\_f2fa 139

	1	Eat-
exp	DТ	f2fa

The second derivative of the density

### **Description**

The second derivative of the density

#### Usage

```
exp_p1_f2fa(x, t, v1, v2)
```

### **Arguments**

x a vector of training data values
t a vector or matrix of predictors
v1 first parameter

v1 first parameter v2 second parameter

### Value

Matrix

exp\_p1\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### **Description**

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
exp_p1_fd(x, t, v1, v2)
```

### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

#### Value

Vector

 $exp_p1_ldd$ 

exp_p1_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	() = 1

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
exp_p1_fdd(x, t, v1, v2)
```

### **Arguments**

x a vector of training data values
 t a vector or matrix of predictors
 v1 first parameter
 v2 second parameter

#### Value

Matrix

exp_p1_ldd	Second derivative matrix of the normalized log-likelihood

## Description

Second derivative matrix of the normalized log-likelihood

# Usage

```
exp_p1_1dd(x, t, v1, d1, v2, d2)
```

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

### Value

Square scalar matrix

exp\_p1\_ldda

The second derivative of the normalized log-likelihood

### **Description**

The second derivative of the normalized log-likelihood

### Usage

```
exp_p1_1dda(x, t, v1, v2)
```

### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

#### Value

Matrix

exp\_p1\_lddd

Third derivative tensor of the normalized log-likelihood

## Description

Third derivative tensor of the normalized log-likelihood

## Usage

```
exp_p1_1ddd(x, t, v1, d1, v2, d2)
```

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

142 exp\_p1\_lmn

## Value

Cubic scalar array

		-	 - 1		- 1	
ex	n	nı	 d	a	പ	2
-	$\sim$	$\mathbf{\nu}$	 u	u	u	u

The third derivative of the normalized log-likelihood

## Description

The third derivative of the normalized log-likelihood

### Usage

```
exp_p1_1ddda(x, t, v1, v2)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

### Value

3d array

One component of the second derivative of the normalized log-likelihood

## Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
exp_p1_lmn(x, t, v1, d1, v2, d2, mm, nn)
```

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

exp\_p1\_lmnp 143

## Value

Scalar value

ехі	p_p	1	1	mr	า	n
$c_{\lambda}$	$\nu$ _ $\nu$	, ı _	_		ı	μ

One component of the third derivative of the normalized log-likelihood

# Description

One component of the third derivative of the normalized log-likelihood

## Usage

```
exp_p1_lmnp(x, t, v1, d1, v2, d2, mm, nn, rr)
```

## **Arguments**

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

	-	-	_
exp_	nl	_lo	σt
$C \wedge D_{-}$	_10 .		5 1

Logf for RUST

# Description

Logf for RUST

# Usage

```
exp_p1_logf(params, x, t)
```

144 exp\_p1\_logfdd

# Arguments

params	model parameters for calculating log
X	a vector of training data values
t	a vector or matrix of predictors

### Value

Scalar value.

exp_p1_logfdd	Second derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
exp_p1_logfdd(x, t, v1, v2)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

## Value

Matrix

exp\_p1\_logfddd 145

exp_p1_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
exp_p1_logfddd(x, t, v1, v2)
```

### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

#### Value

3d array

exp\_p1\_loglik

observed log-likelihood function

## Description

observed log-likelihood function

## Usage

```
exp_p1_loglik(vv, x, t)
```

## **Arguments**

vv parameters

x a vector of training data valuest a vector or matrix of predictors

### Value

Scalar value.

146 *exp\_p1\_means* 

exp_p1_logscores Log scores for MLE and RHP predictions calculated using leave-one- out	eave-one-
--	-----------

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
exp_p1_logscores(logscores, x, t, d1, d2, aderivs)
```

## Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
x	a vector of training data values
t	a vector or matrix of predictors
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

Two scalars

	exp_p1_means	exp distribution: RHP means	
--	--------------	-----------------------------	--

# Description

```
exp distribution: RHP means
```

# Usage

```
exp_p1_means(means, t0, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2)
```

 $\exp_p 1_m u1f$ 

### **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

t0 a single value of the predictor (specify either t0 or n0 but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

### Value

Two scalars

exp\_p1\_mu1f DMGS equation 3.3, mu1 term

## Description

DMGS equation 3.3, mu1 term

### Usage

```
exp_p1_mu1f(alpha, t0, v1, d1, v2, d2)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

### Value

Matrix

148 exp\_p1\_mu2f

exp_p1	l mul	tа

Minus the first derivative of the cdf, at alpha

## Description

Minus the first derivative of the cdf, at alpha

## Usage

```
exp_p1_mu1fa(alpha, t, v1, v2)
```

### **Arguments**

alpha a vector of values of alpha (one minus probability)

t a vector or matrix of predictors

v1 first parameterv2 second parameter

### Value

Vector

exi	~ r	٠1	mıı	ำ	£
EXI	ソート	וע_	IIIU	ız	ı

DMGS equation 3.3, mu2 term

## Description

DMGS equation 3.3, mu2 term

### Usage

```
exp_p1_mu2f(alpha, t0, v1, d1, v2, d2)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

### Value

3d array

exp\_p1\_mu2fa 149

	- 4	~ ^
eyn	nΙ	_mu2fa

Minus the second derivative of the cdf, at alpha

## Description

Minus the second derivative of the cdf, at alpha

### Usage

```
exp_p1_mu2fa(alpha, t, v1, v2)
```

## Arguments

alpha a vector of values of alpha (one minus probability)

t a vector or matrix of predictors

v1 first parameter v2 second parameter

#### Value

Matrix

exn	n1	n1f	

DMGS equation 2.1, p1 term

### **Description**

DMGS equation 2.1, p1 term

## Usage

```
exp_p1_p1f(y, t0, v1, d1, v2, d2)
```

### **Arguments**

у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

### Value

Matrix

exp\_p1\_p2f

			4 0	
exp	n'	Ιn	11 t :	٦
CAP		' — P	, , , ,	4

The first derivative of the cdf

## Description

The first derivative of the cdf

## Usage

```
exp_p1_p1fa(x, t, v1, v2)
```

## Arguments

x a vector of training data values
t a vector or matrix of predictors

v1 first parameter v2 second parameter

### Value

Vector

	- 4	~ ~ ~
exp_	nι	n / t
$C \wedge D_{-}$	_ 12 1	_021

DMGS equation 2.1, p2 term

## Description

DMGS equation 2.1, p2 term

## Usage

```
exp_p1_p2f(y, t0, v1, d1, v2, d2)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

## Value

3d array

exp\_p1\_p2fa 151

	-	~ ~
ΔVN	nl	p2fa
$c_{ND}$	υı	DZ I a

The second derivative of the cdf

### **Description**

The second derivative of the cdf

### Usage

```
exp_p1_p2fa(x, t, v1, v2)
```

### **Arguments**

x a vector of training data values
 t a vector or matrix of predictors
 v1 first parameter

v2 second parameter

### Value

Matrix

exp\_p1\_pd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
exp_p1_pd(x, t, v1, v2)
```

## Arguments

x a vector of training data values t a vector or matrix of predictors v1 first parameter

v2 second parameter

#### Value

Vector

152 exp\_p1\_predictordata

exp_p1_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### **Description**

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
exp_p1_pdd(x, t, v1, v2)
```

### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

#### Value

Matrix

### **Description**

Predicted Parameter and Generalized Residuals

### Usage

```
exp_p1_predictordata(predictordata, x, t, t0, params)
```

## Arguments

predictordata logical that indicates whether to calculate and return predictordata

x a vector of training data values t a vector or matrix of predictors

to a single value of the predictor (specify either to or no but not both)

params model parameters for calculating logf

#### Value

Two vectors

exp\_p1\_waic 153

exp\_p1\_waic Waic

# Description

Waic

# Usage

```
exp_p1_waic(
  waicscores,
  x,
  t,
  v1hat,
  d1,
  v2hat,
  d2,
  lddi,
  lddd,
  lambdad,
  aderivs
)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
x	a vector of training data values
t	a vector or matrix of predictors
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

# Value

Two numeric values.

exp\_pd exp\_pd

exp\_p2fa

The second derivative of the cdf

# Description

The second derivative of the cdf

#### Usage

```
exp_p2fa(x, v1)
```

## Arguments

x a vector of training data values

v1 first parameter

### Value

Matrix

exp\_pd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
exp_pd(x, v1)
```

# Arguments

x a vector of training data values

v1 first parameter

## Value

Vector

exp\_pdd 155

exp_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
exp_pdd(x, v1)
```

## Arguments

x a vector of training data values

v1 first parameter

### Value

Matrix

exp_waic	Waicscores

## Description

Waicscores

### Usage

```
exp_waic(waicscores, x, v1hat, fd1, aderivs)
```

### **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values

v1hat first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

156 fixgpdrange

fixgevrange	Deal with situations in which the user wants d or p outside the GEV
	range

# Description

Deal with situations in which the user wants d or p outside the GEV range

## Usage

```
fixgevrange(y, v1, v2, v3)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Vector

fixgpdrange	Deal with situations in which the user wants d or p outside the GPD
	range

# Description

Deal with situations in which the user wants d or p outside the GPD range

# Usage

```
fixgpdrange(y, v1, v2, v3)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
v2	second parameter
v3	third parameter

## Value

Vector

frechet\_k1\_cp

Frechet Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qfrechet_k1_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    kloc = 0,
    fd1 = 0.01,
    fd2 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE,
    aderivs = TRUE
)
```

```
rfrechet_k1_cp(
 n,
 Х,
 kloc = 0,
 fd1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
dfrechet_k1_cp(
 Х,
 y = x,
 kloc = 0,
 fd1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
pfrechet_k1_cp(
 Х,
 y = x,
 kloc = 0,
 fd1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
tfrechet_k1_cp(n, x, kloc = 0, fd1 = 0.01, fd2 = 0.01, debug = FALSE)
```

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
kloc	the known location parameter
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter

means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- $\bullet\,$  cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Frechet distribution has distribution function

$$F(x; \sigma, \lambda) = \exp\left(-\left(\frac{x-\mu}{\sigma}\right)^{-\lambda}\right)$$

where  $x > \mu$  is the random variable,  $\sigma > 0, \lambda > 0$  are the parameters and we consider  $\mu$  to be known (hence the k1 in the name).

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma,\lambda) \propto \frac{1}{\sigma\lambda}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

• ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)

• cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
#
# example 1
x=fitdistcp::d51frechet_k1_example_data_v1
p=c(1:9)/10
q=qfrechet_k1_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
```

164 frechet\_k1\_f1fa

```
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qfrechet_k1_cp)",
main="Frechet: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

frechet\_k1\_f1f

DMGS equation 3.3, f1 term

## Description

DMGS equation 3.3, f1 term

## Usage

```
frechet_k1_f1f(y, v1, fd1, v2, fd2, kloc)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

#### Value

Matrix

frechet\_k1\_f1fa

The first derivative of the density

## Description

The first derivative of the density

### Usage

```
frechet_k1_f1fa(x, v1, v2, kloc)
```

frechet\_k1\_f2f 165

# Arguments

Χ	a vector of training	g data values
---	----------------------	---------------

v1 first parameter v2 second parameter

kloc the known location parameter

## Value

Vector

frechet\_k1\_f2f

DMGS equation 3.3, f2 term

# Description

DMGS equation 3.3, f2 term

## Usage

```
frechet_k1_f2f(y, v1, fd1, v2, fd2, kloc)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

### Value

3d array

166 frechet\_k1\_fd

frec	het	k1	f2fa

The second derivative of the density

### **Description**

The second derivative of the density

### Usage

```
frechet_k1_f2fa(x, v1, v2, kloc)
```

## Arguments

Y	a vector	of training	data values
Λ	a vector	or training	uata varues

v1 first parameter v2 second parameter

kloc the known location parameter

### Value

Matrix

frechet\_k1\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### **Description**

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
frechet_k1_fd(x, v1, v2, v3)
```

# Arguments

x a vector of training data va	iues
--------------------------------	------

v1 first parameterv2 second parameterv3 third parameter

#### Value

Vector

frechet\_k1\_fdd 167

frechet_k1_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
frechet_k1_fdd(x, v1, v2, v3)
```

## Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

frechet\_k1\_ldd Second derivative matrix of the normalized log-likelihood

## Description

Second derivative matrix of the normalized log-likelihood

# Usage

```
frechet_k1_ldd(x, v1, fd1, v2, fd2, kloc)
```

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

168 frechet\_k1\_lddd

### Value

Square scalar matrix

frechet\_k1\_ldda

The second derivative of the normalized log-likelihood

### **Description**

The second derivative of the normalized log-likelihood

### Usage

```
frechet_k1_ldda(x, v1, v2, kloc)
```

## Arguments

x a vector of training data value	X	a vector	of training	data value
-----------------------------------	---	----------	-------------	------------

v1 first parameter v2 second parameter

kloc the known location parameter

#### Value

Matrix

 $frechet_k1_lddd$ 

Third derivative tensor of the normalized log-likelihood

# Description

Third derivative tensor of the normalized log-likelihood

## Usage

```
frechet_k1_lddd(x, v1, fd1, v2, fd2, kloc)
```

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

frechet\_k1\_lddda 169

### Value

Cubic scalar array

frechet\_k1\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

#### Usage

```
frechet_k1_lddda(x, v1, v2, kloc)
```

## Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

kloc the known location parameter

### Value

3d array

## Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
frechet_k1_lmn(x, v1, fd1, v2, fd2, kloc, mm, nn)
```

170 frechet\_k1\_lmnp

# Arguments

x	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

### Value

Scalar value

frechet_k1_lmnp	omponent of the third derivative of the normalized log-likelihood
-----------------	---

# Description

One component of the third derivative of the normalized log-likelihood

# Usage

```
frechet_k1_lmnp(x, v1, fd1, v2, fd2, kloc, mm, nn, rr)
```

# Arguments

Χ	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

## Value

Scalar value

frechet\_k1\_logf

frechet\_k1\_logf

Logf for RUST

## Description

Logf for RUST

### Usage

```
frechet_k1_logf(params, x, kloc)
```

### **Arguments**

params model parameters for calculating logf x a vector of training data values kloc the known location parameter

#### Value

Scalar value.

frechet\_k1\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
frechet_k1_logfdd(x, v1, v2, v3)
```

## Arguments

x a vector of training data value	X	a vector of training	data values
-----------------------------------	---	----------------------	-------------

v1 first parameterv2 second parameterv3 third parameter

## Value

Matrix

frechet\_k1\_mu1f

_	native of the log density Created by Stephen Jewson using Andrew Clausen and Serguei Sokol
---	---

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
frechet_k1_logfddd(x, v1, v2, v3)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

## Value

3d array

# Description

DMGS equation 3.3, mu1 term

# Usage

```
frechet_k1_mu1f(alpha, v1, fd1, v2, fd2, kloc)
```

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

frechet\_k1\_mu1fa 173

## Value

Matrix

### **Description**

Minus the first derivative of the cdf, at alpha

### Usage

```
frechet_k1_mu1fa(alpha, v1, v2, kloc)
```

### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

kloc the known location parameter

# Value

Vector

# Description

DMGS equation 3.3, mu2 term

## Usage

```
frechet_k1_mu2f(alpha, v1, fd1, v2, fd2, kloc)
```

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

frechet\_k1\_p1f

### Value

3d array

frechet\_k1\_mu2fa

Minus the second derivative of the cdf, at alpha

### **Description**

Minus the second derivative of the cdf, at alpha

### Usage

```
frechet_k1_mu2fa(alpha, v1, v2, kloc)
```

### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

kloc the known location parameter

#### Value

Matrix

 $frechet_k1_p1f$ 

DMGS equation 3.3, p1 term

# Description

DMGS equation 3.3, p1 term

## Usage

```
frechet_k1_p1f(y, v1, fd1, v2, fd2, kloc)
```

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

frechet\_k1\_p1fa

## Value

Matrix

frechet\_k1\_p1fa

The first derivative of the cdf

## Description

The first derivative of the cdf

## Usage

```
frechet_k1_p1fa(x, v1, v2, kloc)
```

## Arguments

x a vector of training data value	X	a vector	of training	data value
-----------------------------------	---	----------	-------------	------------

v1 first parameter v2 second parameter

kloc the known location parameter

### Value

Vector

 $frechet_k1_p2f$ 

DMGS equation 3.3, p2 term

# Description

DMGS equation 3.3, p2 term

## Usage

```
frechet_k1_p2f(y, v1, fd1, v2, fd2, kloc)
```

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

### Value

3d array

frechet\_k1\_p2fa

The second derivative of the cdf

### **Description**

The second derivative of the cdf

## Usage

```
frechet_k1_p2fa(x, v1, v2, kloc)
```

### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

kloc the known location parameter

### Value

Matrix

frechet\_k1\_pd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
frechet_k1_pd(x, v1, v2, v3)
```

## Arguments

X	a vector of training data	values
---	---------------------------	--------

v1 first parameterv2 second parameterv3 third parameter

frechet\_k1\_pdd 177

## Value

Vector

frechet_k1_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
frechet_k1_pdd(x, v1, v2, v3)
```

## Arguments

x a vector of training data values

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

frechet\_k1\_waic
Waic

# Description

Waic

## Usage

```
frechet_k1_waic(
  waicscores,
  x,
  v1hat,
  fd1,
  v2hat,
  fd2,
  kloc,
```

178 frechet\_loglik

```
lddi,
lddd,
lambdad,
aderivs
```

#### **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values

v1hat first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

v2hat second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

kloc the known location parameter
lddi inverse observed information matrix
lddd third derivative of log-likelihood

lambdad derivative of the log prior

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

### Description

log-likelihood function

#### Usage

```
frechet_loglik(vv, x, kloc)
```

### Arguments

vv parameters

x a vector of training data valueskloc the known location parameter

#### Value

Scalar value.

frechet\_logscores 179

frechet_logscores	Log scores for MLE and RHP predictions calculated using leave-one-out

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
frechet_logscores(logscores, x, fd1 = 0.01, fd2 = 0.01, kloc, aderivs)
```

## Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
x	a vector of training data values
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

## Value

Two scalars

frechet_means	MLE and RHP predictive means	

# Description

MLE and RHP predictive means

# Usage

```
frechet_means(means, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2, kloc)
```

#### **Arguments**

logical that indicates whether to return analytical estimates for the distribution means

means (longer runtime)

ml\_params parameters

inverse observed information matrix lddi third derivative of log-likelihood 1ddd lambdad\_rhp derivative of the log RHP prior

nx length of training data number of parameters dim

kloc the known location parameter

#### Value

Two scalars

frechet\_p2k1\_cp

Frechet Distribution with Predictor, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qfrechet_p2k1_cp(
 Х,
  t,
  t0 = NA,
 n0 = NA,
 p = seq(0.1, 0.9, 0.1),
 d1 = 0.01,
  d2 = 0.01,
  fd3 = 0.01,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  kloc = 0,
  dmgs = TRUE,
  rust = FALSE,
  nrust = 1e+05,
 predictordata = TRUE,
  centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
)
rfrechet_p2k1_cp(
 n,
 Х,
  t,
  t0 = NA,
 n0 = NA,
 d1 = 0.01,
 d2 = 0.01,
  fd3 = 0.01,
 kloc = 0,
  rust = FALSE,
 mlcp = TRUE,
  centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
)
dfrechet_p2k1_cp(
 Х,
  t,
 t0 = NA,
 n0 = NA,
 y = x,
  d1 = 0.01,
 d2 = 0.01,
```

```
fd3 = 0.01,
  kloc = 0,
  rust = FALSE,
  nrust = 1000,
  centering = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
pfrechet_p2k1_cp(
 х,
 t,
  t0 = NA,
 n0 = NA,
 y = x,
 d1 = 0.01,
 d2 = 0.01,
  fd3 = 0.01,
 kloc = 0,
  rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
tfrechet_p2k1_cp(
 n,
 Х,
  t,
 d1 = 0.01,
 d2 = 0.01,
 fd3 = 0.01,
 kloc = 0,
 debug = FALSE
)
```

### Arguments

X	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
d2	if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter $% \left( 1\right) =\left( 1\right) \left( 1\right)$

fd3	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the third parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
kloc	the known location parameter
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### Details of the Model

The Frechet distribution with predictor has distribution function

$$F(x; a, b, \lambda) = \exp\left(-\left(\frac{x-\mu}{\sigma(a, b)}\right)^{-\lambda}\right)$$

where  $x > \mu$  is the random variable,  $\sigma = e^{a+bt}$  is the scale parameter, modelled as a function of parameters a, b and predictor t, and  $\lambda > 0$  is the shape parameter. We consider  $\mu$  to be known (hence the k1 in the name).

The calibrating prior is given by the right Haar prior, which is

$$\pi(a,b) \propto 1$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),

188 frechet\_p2k1\_f1f

• Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
#
# example 1
x=fitdistcp::d71frechet_p2k1_example_data_v1_x
tt=fitdistcp::d71frechet_p2k1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qfrechet_p2k1_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qfrechet_p2k1_cp)",
main="Frechet w/ p2: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

frechet\_p2k1\_f1f

DMGS equation 2.1, f1 term

### **Description**

DMGS equation 2.1, f1 term

### Usage

```
frechet_p2k1_f1f(y, t0, v1, d1, v2, d2, v3, fd3, kloc)
```

### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the param-
	eter
kloc	the known location parameter

frechet\_p2k1\_f1fa 189

### Value

Matrix

frechet\_p2k1\_f1fa

The first derivative of the density

### Description

The first derivative of the density

### Usage

```
frechet_p2k1_f1fa(x, t, v1, v2, v3, kloc)
```

### **Arguments**

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

kloc the known location parameter

### Value

Vector

 $frechet_p2k1_f2f$ 

DMGS equation 2.1, f2 term

### Description

DMGS equation 2.1, f2 term

```
frechet_p2k1_f2f(y, t0, v1, d1, v2, d2, v3, fd3, kloc)
```

190 frechet\_p2k1\_f2fa

### Arguments

у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

## Value

3d array

frechet\_p2k1\_f2fa

The second derivative of the density

# Description

The second derivative of the density

### Usage

```
frechet_p2k1_f2fa(x, t, v1, v2, v3, kloc)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
kloc	the known location parameter

# Value

Matrix

frechet\_p2k1\_fd 191

fr	echet_p2k1_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
frechet_p2k1_fd(x, t, v1, v2, v3, v4)
```

### Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

### Value

Vector

-, -	cond derivative of the density Created by Stephen Jewson using De- c() by Andrew Clausen and Serguei Sokol
------	---

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
frechet_p2k1_fdd(x, t, v1, v2, v3, v4)
```

192 frechet\_p2k1\_ldd

## Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

 $frechet_p2k1_ldd$ 

Second derivative matrix of the normalized log-likelihood

## Description

Second derivative matrix of the normalized log-likelihood

## Usage

```
frechet_p2k1_ldd(x, t, v1, d1, v2, d2, v3, fd3, kloc)
```

## **Arguments**

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

### Value

Square scalar matrix

frechet\_p2k1\_ldda 193

frechet_p2k1_l
----------------

The second derivative of the normalized log-likelihood

### Description

The second derivative of the normalized log-likelihood

### Usage

```
frechet_p2k1_ldda(x, t, v1, v2, v3, kloc)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
kloc	the known location parameter

#### Value

Matrix

frechet\_p2k1\_lddd

Third derivative tensor of the normalized log-likelihood

## Description

Third derivative tensor of the normalized log-likelihood

### Usage

```
frechet_p2k1_lddd(x, t, v1, d1, v2, d2, v3, fd3, kloc)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

194 frechet\_p2k1\_lddda

v3 third parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

kloc the known location parameter

### Value

Cubic scalar array

frechet\_p2k1\_lddda

The third derivative of the normalized log-likelihood

## Description

The third derivative of the normalized log-likelihood

### Usage

```
frechet_p2k1_lddda(x, t, v1, v2, v3, kloc)
```

### Arguments

x a vector of training data values

t a vector or matrix of predictors

v1 first parameter

v2 second parameter

v3 third parameter

kloc the known location parameter

### Value

3d array

frechet\_p2k1\_lmn 195

frechet_p2k1_lmn	One component of the second derivative of the normalized log-likelihood
------------------	---

## Description

One component of the second derivative of the normalized log-likelihood

### Usage

```
frechet_p2k1_lmn(x, t, v1, d1, v2, d2, v3, fd3, kloc, mm, nn)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

### Value

Scalar value

<pre>frechet_p2k1_lmnp</pre>	One component of the second derivative of the normalized log-
	likelihood

## Description

One component of the second derivative of the normalized log-likelihood

```
frechet_p2k1_lmnp(x, t, v1, d1, v2, d2, v3, fd3, kloc, mm, nn, rr)
```

frechet\_p2k1\_logf

# Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

# Description

Logf for RUST

### Usage

```
frechet_p2k1_logf(params, x, t, kloc)
```

# Arguments

params	model parameters for calculating logf
Х	a vector of training data values
t	a vector or matrix of predictors
kloc	the known location parameter

### Value

Scalar value.

frechet\_p2k1\_logfdd 197

frechet_p2k1_logfdd	Second derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
frechet_p2k1_logfdd(x, t, v1, v2, v3, v4)
```

### Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

frechet_p2k1_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
frechet_p2k1_logfddd(x, t, v1, v2, v3, v4)
```

frechet\_p2k1\_loglik

# Arguments

198

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

### Value

3d array

# Description

observed log-likelihood function

## Usage

```
frechet_p2k1_loglik(vv, x, t, kloc)
```

# Arguments

VV	parameters
Х	a vector of training data values
t	a vector or matrix of predictors
kloc	the known location parameter

### Value

Scalar value.

frechet\_p2k1\_logscores

Log scores for MLE and RHP predictions calculated using leave-one-out

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

### Usage

```
frechet_p2k1_logscores(logscores, x, t, d1, d2, fd3, kloc, aderivs = TRUE)
```

### Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
x	a vector of training data values
t	a vector or matrix of predictors
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

# Value

Two scalars

frechet\_p2k1\_means frechet\_k1 distribution: RHP mean

## Description

frechet\_k1 distribution: RHP mean

200 frechet\_p2k1\_mu1f

### Usage

```
frechet_p2k1_means(
  means,
  t0,
  ml_params,
  lddi,
  lddd,
  lambdad_rhp,
  nx,
  dim,
  kloc
)
```

## Arguments

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

t0 a single value of the predictor (specify either t0 or n0 but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

kloc the known location parameter

### Value

Two scalars

frechet\_p2k1\_mu1f
DMGS equation 3.3, mu1 term

### **Description**

DMGS equation 3.3, mu1 term

```
frechet_p2k1_mu1f(alpha, t0, v1, d1, v2, d2, v3, fd3, kloc)
```

frechet\_p2k1\_mu1fa 201

### Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

## Value

Matrix

# Description

Minus the first derivative of the cdf, at alpha

### Usage

```
frechet_p2k1_mu1fa(alpha, t, v1, v2, v3, kloc)
```

### Arguments

alpha	a vector of values of alpha (one minus probability)
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
kloc	the known location parameter

# Value

Vector

202 frechet\_p2k1\_mu2fa

# Description

DMGS equation 3.3, mu2 term

## Usage

```
frechet_p2k1_mu2f(alpha, t0, v1, d1, v2, d2, v3, fd3, kloc)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

### Value

3d array

frechet_p2k1_mu2fa	Minus the second derivative of the cdf, at alpha	

# Description

Minus the second derivative of the cdf, at alpha

```
frechet_p2k1_mu2fa(alpha, t, v1, v2, v3, kloc)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
kloc	the known location parameter

### Value

Matrix

# Description

DMGS equation 2.1, p1 term

### Usage

```
frechet_p2k1_p1f(y, t0, v1, d1, v2, d2, v3, fd3, kloc)
```

### Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

### Value

Matrix

### Description

The first derivative of the cdf

### Usage

```
frechet_p2k1_p1fa(x, t, v1, v2, v3, kloc)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

kloc the known location parameter

### Value

Vector

frechet\_p2k1\_p2f

DMGS equation 2.1, p2 term

## Description

DMGS equation 2.1, p2 term

### Usage

```
frechet_p2k1_p2f(y, t0, v1, d1, v2, d2, v3, fd3, kloc)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

frechet\_p2k1\_p2fa 205

v3 third parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

kloc the known location parameter

### Value

3d array

frechet\_p2k1\_p2fa

The second derivative of the cdf

### Description

The second derivative of the cdf

### Usage

```
frechet_p2k1_p2fa(x, t, v1, v2, v3, kloc)
```

### Arguments

x a vector of training data values

t a vector or matrix of predictors

v1 first parameter

v2 second parameter

v3 third parameter

kloc the known location parameter

### Value

Matrix

frechet_p2k1_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
frechet_p2k1_pd(x, t, v1, v2, v3, v4)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

### Value

Vector

frechet_p2k1_pdd Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol		,	
---	--	---	--

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
frechet_p2k1_pdd(x, t, v1, v2, v3, v4)
```

# Arguments

	a vector of training data values
	a vector or matrix of predictors
1	first parameter
2	second parameter
3	third parameter
4	fourth parameter
	1 2 3 4

### Value

Matrix

frechet\_p2k1\_predictordata

Predicted Parameter and Generalized Residuals

## Description

Predicted Parameter and Generalized Residuals

### Usage

```
frechet_p2k1_predictordata(predictordata, x, t, t0, params, kloc)
```

## Arguments

predictordata	logical that indicates whether to calculate and return predictordata
x	a vector of training data values
t	a vector or matrix of predictors
t0	a single value of the predictor (specify either $t0$ or $n0$ but not both)
params	model parameters for calculating logf
kloc	the known location parameter

### Value

Two vectors

208 frechet\_p2k1\_waic

frechet\_p2k1\_waic

Waic

## Description

Waic

## Usage

```
frechet_p2k1_waic(
 waicscores,
 Х,
 t,
 v1hat,
 d1,
 v2hat,
 d2,
 v3hat,
 fd3,
 kloc,
 lddi,
 lddd,
 lambdad,
  aderivs
)
```

### Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
t	a vector or matrix of predictors
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3hat	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

gamma\_cp

Gamma Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qgamma_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    fd1 = 0.01,
    fd2 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    prior = "type 1",
    debug = FALSE,
```

```
aderivs = TRUE
rgamma_cp(
 n,
 Х,
 fd1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
dgamma_cp(
 Х,
 y = x,
  fd1 = 0.01,
  fd2 = 0.01,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
pgamma_cp(
 Х,
 y = x,
 fd1 = 0.01,
 fd2 = 0.01,
  rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
tgamma_cp(n, x, fd1 = 0.01, fd2 = 0.01, debug = FALSE)
```

### **Arguments**

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)

waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
prior	logical indicating which prior to use
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Gamma distribution has probability density function

$$f(x; \alpha, \sigma) = \frac{1}{\sigma^{\alpha} \Gamma(\alpha)} x^{\alpha - 1} e^{-x/\sigma}$$

where  $x \ge 0$  is the random variable and  $\alpha > 0, \sigma > 0$  are the parameters.

The calibrating prior we use is

$$\pi(\alpha, \sigma) \propto \frac{1}{\alpha \sigma}$$

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

• ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible

• cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),

- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d100gamma_example_data_v1
p=c(1:9)/10
q=qgamma_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgamma_cp)",
main="Gamma: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

216 gamma\_f1fa

gamma f	1f
---------	----

DMGS equation 3.3, f1 term

### Description

DMGS equation 3.3, f1 term

### Usage

```
gamma_f1f(y, v1, fd1, v2, fd2)
```

### Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

eter

### Value

Matrix

~~~~	£1	۲,
gamma_	_	Ιd

The first derivative of the density

## Description

The first derivative of the density

### Usage

```
gamma_f1fa(x, v1, v2)
```

### Arguments

X	a vector of	training of	ata values
X	a vector of	training c	iata varues

v1 first parameterv2 second parameter

### Value

Vector

gamma\_f2f 217

gamma	f2f
gaillilla	1 4 1

DMGS equation 3.3, f2 term

### Description

DMGS equation 3.3, f2 term

### Usage

```
gamma_f2f(y, v1, fd1, v2, fd2)
```

### Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

eter

### Value

3d array

gamma	f2fa
gaiiiiia_	_ 1

The second derivative of the density

### Description

The second derivative of the density

### Usage

```
gamma_f2fa(x, v1, v2)
```

### Arguments

X	a vector of	training	data values
**			anta (mass

v1 first parameter v2 second parameter

#### Value

Matrix

218 gamma\_fdd

gamma_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

### Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gamma_fd(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

6.1.1	
gamma_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gamma_fdd(x, v1, v2)
```

### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

gamma\_gg 219

gamma_gg	Second derivative matrix of the expected log-likelihood	

### Description

Second derivative matrix of the expected log-likelihood

### Usage

```
gamma_gg(v1, fd1, v2, fd2)
```

### Arguments

v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Square scalar matrix

gamma_gmn	One component of the second derivative of the expected log-likelihood

### Description

One component of the second derivative of the expected log-likelihood

# Usage

```
gamma_gmn(alpha, v1, fd1, v2, fd2, mm, nn)
```

### Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

220 gamma\_ldda

#### Value

Scalar value

	_		
~~mm~	- 1.	do	1
gamma	- 1.1	uι	ı

Second derivative matrix of the normalized log-likelihood

#### Description

Second derivative matrix of the normalized log-likelihood

#### Usage

```
gamma_1dd(x, v1, fd1, v2, fd2)
```

#### **Arguments**

x a vector of training data value	X	a vector	of training	data value
-----------------------------------	---	----------	-------------	------------

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

fd2 the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Square scalar matrix

~~mm~	٦,	44-	
gamma_	_T(	dda	1

The second derivative of the normalized log-likelihood

#### Description

The second derivative of the normalized log-likelihood

#### Usage

```
gamma_ldda(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameterv2 second parameter

gamma\_lddd 221

### Value

Matrix

gamma_lddd Third derivative tensor of the normalized log-likelihood	d
---------------------------------------------------------------------	---

### Description

Third derivative tensor of the normalized log-likelihood

### Usage

```
gamma_lddd(x, v1, fd1, v2, fd2)
```

### Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Cubic scalar array

gamma_lddda	The third derivative of the normalized log-likelihood	
-------------	-------------------------------------------------------	--

### Description

The third derivative of the normalized log-likelihood

### Usage

```
gamma_lddda(x, v1, v2)
```

### Arguments

x a vector of training data valu	ies
----------------------------------	-----

v1 first parameter v2 second parameter 222 gamma\_lmnp

### Value

3d array

gamma_lmn One component of the second derivative of the normalized log- likelihood	gamma_lmn	One component of the second derivative of the normalized log-likelihood
---------------------------------------------------------------------------------------	-----------	-------------------------------------------------------------------------

### Description

One component of the second derivative of the normalized log-likelihood

### Usage

```
gamma_lmn(x, v1, fd1, v2, fd2, mm, nn)
```

### Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

#### Value

Scalar value

gamma_lmnp One component of the second derivative of the normalized log likelihood	3-
------------------------------------------------------------------------------------	----

### Description

One component of the second derivative of the normalized log-likelihood

### Usage

```
gamma_lmnp(x, v1, fd1, v2, fd2, mm, nn, rr)
```

gamma\_logf 223

# Arguments

x	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

$a\_logf$ $Logf for RUST$
---------------------------

# Description

Logf for RUST

### Usage

```
gamma_logf(params, x)
```

### Arguments

params model parameters for calculating logf x a vector of training data values

### Value

Scalar value.

224 gamma\_logfddd

Deriv() by Anarew Clausen and Serguet Sokot	gamma_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
---------------------------------------------	--------------	------------------------------------------------------------------------------------------------------------------

### Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gamma_logfdd(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Matrix

gamma_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

### Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gamma_logfddd(x, v1, v2)
```

### Arguments

X	a vector of training	data values
λ	a vector or training	uata vaiues

v1 first parameter v2 second parameter

#### Value

3d array

gamma\_loglik 225

gamma_loglik	gamma	loglik	
--------------	-------	--------	--

log-likelihood function

### Description

log-likelihood function

### Usage

```
gamma_loglik(vv, x)
```

### Arguments

vv parameters

x a vector of training data values

### Value

Scalar value.

gamma_logscores	Log scores for MLE and RHP predictions calculated using leave-one-
	out

### Description

Log scores for MLE and RHP predictions calculated using leave-one-out

### Usage

```
gamma_logscores(logscores, x, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

# Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
X	a vector of training data values
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

Two scalars

gamma\_mu1f

gamma_means	MLE and RHP predictive means
-------------	------------------------------

### Description

MLE and RHP predictive means

### Usage

```
gamma_means(means, ml_params, lddi, lddd, lambdad_cp, nx, dim = 2)
```

### Arguments

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrix
 lddd third derivative of log-likelihood
 lambdad\_cp derivative of the log prior
 length of training data
 dim number of parameters

#### Value

Two scalars

gamma_mu1f DMGS equation 3.3, mu1 term
----------------------------------------

### Description

DMGS equation 3.3, mu1 term

### Usage

```
gamma_mu1f(alpha, v1, fd1, v2, fd2)
```

### **Arguments**

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

gamma\_mu2f 227

### Value

Matrix

gamma\_mu2f

DMGS equation 3.3, mu2 term

### Description

DMGS equation 3.3, mu2 term

### Usage

```
gamma_mu2f(alpha, v1, fd1, v2, fd2)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

the fractional delta used in the numerical derivatives with respect to the parameter

### Value

3d array

gamma\_p1f

DMGS equation 3.3, p1 term

### Description

DMGS equation 3.3, p1 term

# Usage

```
gamma_p1f(y, v1, fd1, v2, fd2)
```

228 gamma\_p2f

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Matrix

gamma_p2f	DMGS equation 3.3, p2 term

# Description

DMGS equation 3.3, p2 term

# Usage

```
gamma_p2f(y, v1, fd1, v2, fd2)
```

# Arguments

у	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

3d array

gamma\_waic 229

### Description

Waic

### Usage

```
gamma_waic(waicscores, x, v1hat, fd1, v2hat, fd2, lddi, lddd, lambdad, aderivs)
```

### Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
x	a vector of training data values
v1hat	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior

logical for whether to use analytic derivatives (instead of numerical)

### Value

aderivs

Two numeric values.

|--|--|--|

### Description

Check MLE

### Usage

```
gev_checkmle(ml_params, minxi, maxxi)
```

#### **Arguments**

$ml_{-}$	params	parameters

minxi minimum value of shape parameter xi maxxi maximum value of shape parameter xi

#### Value

No return value (just a message to the screen).

gev_cp	Generalized Extreme Value Distribution, Predictions Based on a Cal-
	ibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

### Usage

```
qgev_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    ics = c(0, 0, 0),
    d1 = 0.01,
```

```
fd2 = 0.01,
  d3 = 0.01,
  fdalpha = 0.01,
 minxi = -1,
 maxxi = 999,
 means = FALSE,
 waicscores = FALSE,
 extramodels = FALSE,
  pdf = FALSE,
  customprior = 0,
  dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
  pwm = FALSE,
 debug = FALSE,
  aderivs = TRUE
)
rgev_cp(
 n,
 х,
 ics = c(0, 0, 0),
 d1 = 0.01,
 fd2 = 0.01,
 d3 = 0.01,
 minxi = -0.45,
 maxxi = 0.45,
 extramodels = FALSE,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
dgev_cp(
 х,
 y = x,
 ics = c(0, 0, 0),
 d1 = 0.01,
 fd2 = 0.01,
 d3 = 0.01,
 minxi = -0.45,
 maxxi = 0.45,
 extramodels = FALSE,
  rust = FALSE,
  nrust = 1000,
  debug = FALSE,
  aderivs = TRUE
```

```
)
pgev_cp(
 Х,
 y = x,
 ics = c(0, 0, 0),
 d1 = 0.01,
 fd2 = 0.01,
 d3 = 0.01,
 minxi = -0.45,
 \max xi = 0.45,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
tgev_cp(
 n,
 х,
 ics = c(0, 0, 0),
 d1 = 0.01,
 fd2 = 0.01,
 d3 = 0.01,
 extramodels = FALSE,
 debug = FALSE
)
```

### Arguments

х	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
ics	initial conditions for the maximum likelihood search
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter $% \left( 1\right) =\left( 1\right) \left( 1\right) $
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
d3	if aderivs=FALSE, the delta used for numerical derivatives with respect to the third parameter $% \left( 1\right) =\left( 1\right) \left( 1\right) $
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
minxi	the minimum allowed value of the shape parameter (decrease with caution)
maxxi	the maximum allowed value of the shape parameter (increase with caution)
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)

waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
extramodels	logical that indicates whether to run additional calculations and add three additional prediction models (longer runtime)
pdf	logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer runtime)
customprior	a custom value for the slope of the log prior at the maxlik estimate
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
pwm	logical for whether to include PWM results (longer runtime)
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- $\bullet$  predicted parameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.

- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The GEV distribution has distribution function

$$F(x; \mu, \sigma, \xi) = \exp(-t(x; \mu, \sigma, \xi))$$

where

$$t(x; \mu, \sigma, \xi) = \begin{cases} \left[1 + \xi \left(\frac{x - \mu}{\sigma}\right)\right]^{-1/\xi} & \text{if } \xi \neq 0\\ \exp\left(-\frac{x - \mu}{\sigma}\right) & \text{if } \xi = 0 \end{cases}$$

where x is the random variable and  $\mu, \sigma > 0, \xi$  are the parameters.

The calibrating prior we use is given by

$$\pi(\mu, \sigma, \xi) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

The code will stop with an error if the input data gives a maximum likelihood value for the shape parameter that lies outside the range (minxi, maxxi), since outside this range there may be numerical problems. Such values seldom occur in real observed data for maxima.

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Optional Return Values (EVT models only)**

q\*\*\*\* optionally returns the following, for EVT models only:

• cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

#### Optional Return Values (some EVT models only)

q\*\*\*\* optionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_quantiles: predictive quantiles calculated from Bayesian integration with a flat prior.
- rh\_ml\_quantiles: predictive quantiles calculated from Bayesian integration with the calibrating prior, and the maximmum likelihood estimate for the shape parameter.
- jp\_quantiles: predictive quantiles calculated from Bayesian integration with Jeffreys' prior.

r\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_deviates: predictive random deviates calculated using a Bayesian analysis with a flat prior.
- rh\_ml\_deviates: predictive random deviates calculated using a Bayesian analysis with the RHP-MLE prior.
- jp\_deviates: predictive random deviates calculated using a Bayesian analysis with the JP.

d\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_pdf: predictive density function from a Bayesian analysis with the flat prior.
- rh\_ml\_pdf: predictive density function from a Bayesian analysis with the RHP-MLE prior.
- jp\_pdf: predictive density function from a Bayesian analysis with the JP.

p\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_cdf: predictive distribution function from a Bayesian analysis with the flat prior.
- rh\_ml\_cdf: predictive distribution function from a Bayesian analysis with the RHP-MLE prior.
- jp\_cdf: predictive distribution function from a Bayesian analysis with the JP.

These additional predictive distributions are included for comparison with the calibrating prior model. They generally give less good reliability than the calibrating prior.

#### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2024) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

· Cauchy (cauchy),

- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

gev\_f1f 239

#### **Examples**

```
# # example 1
shape=-0.4
x=fitdistcp::d110gev_example_data_v1
p=c(1:9)/10
q=qgev_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgev_cp)",
main="GEVD: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue",lwd=2)
cat(" ml_params=",q$ml_params,"\n")
```

gev\_f1f

DMGS equation 3.3, f1 term

### Description

DMGS equation 3.3, f1 term

#### Usage

```
gev_f1f(y, v1, d1, v2, fd2, v3, d3)
```

#### **Arguments**

у	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

#### Value

Matrix

240 gev\_f2f

gev	f1	fa

The first derivative of the density

### Description

The first derivative of the density

### Usage

```
gev_f1fa(x, v1, v2, v3)
```

### Arguments

X	a vector of	training	data values
^	a vector or	uanning	data varues

v1 first parameterv2 second parameterv3 third parameter

#### Value

Vector

gev\_f2f

DMGS equation 3.3, f2 term

### Description

DMGS equation 3.3, f2 term

### Usage

### Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

gev\_f2fa 241

#### Value

3d array

gev\_f2fa

The second derivative of the density

#### Description

The second derivative of the density

# Usage

```
gev_f2fa(x, v1, v2, v3)
```

#### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

gev\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

### Arguments

x a vector of training data values

v1 first parameterv2 second parameterv3 third parameter

242 gev\_ggd\_mev

### Value

Vector

gev_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_fdd(x, v1, v2, v3)
```

### Arguments

x a vector of training data value	X	a vector	of training	data value
-----------------------------------	---	----------	-------------	------------

v1 first parameter

v2 second parameter

v3 third parameter

#### Value

Matrix

gev_ggd_mev	Derivative of expected information matrix, based on MEV routine
	gev.infomat

### Description

Derivative of expected information matrix, based on MEV routine gev.infomat

### Usage

```
gev_ggd_mev(v1, d1, v2, fd2, v3, d3)
```

gev\_ggid\_mev 243

### Arguments

v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

### Value

Square scalar matrix

gev_ggid_mev	Derivative of inverse expected information matrix, based on MEV routine gev.infomat

# Description

Derivative of inverse expected information matrix, based on MEV routine gev.infomat

### Usage

```
gev_ggid_mev(v1, d1, v2, fd2, v3, d3)
```

# Arguments

v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

### Value

Cubic scalar array

gev\_k12\_ppm\_minusloglik

Temporary dummy for one of the ppm models

#### **Description**

Temporary dummy for one of the ppm models

#### Usage

```
gev_k12_ppm_minusloglik(x)
```

#### **Arguments**

Х

a vector of training data values

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.

• cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

gev\_k3\_cp Generalized Extreme Value Distribution with Known Shape, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qgev_k3_cp(
  х,
 p = seq(0.1, 0.9, 0.1),
 d1 = 0.01,
  fd2 = 0.01,
  fdalpha = 0.01,
  kshape = 0,
 means = FALSE,
 waicscores = FALSE,
 pdf = FALSE,
  dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 debug = FALSE,
 aderivs = TRUE
)
rgev_k3_cp(
 n,
 х,
  d1 = 0.01,
  fd2 = 0.01,
  kshape = 0,
  rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
dgev_k3_cp(
 х,
 y = x,
 d1 = 0.01,
  fd2 = 0.01,
 kshape = 0,
  rust = FALSE,
  nrust = 1000,
 debug = FALSE,
  aderivs = TRUE
)
pgev_k3_cp(
 Х,
 y = x,
 d1 = 0.01,
  fd2 = 0.01,
 kshape = 0,
```

```
rust = FALSE,
nrust = 1000,
debug = FALSE,
aderivs = TRUE
)

tgev_k3_cp(n, x, d1 = 0.01, fd2 = 0.01, kshape = 0, debug = FALSE)
```

### Arguments

٠	•	
	x	a vector of training data values
	р	a vector of probabilities at which to generate predictive quantiles
	d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
	fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
	fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
	kshape	the known shape parameter
	means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
	waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
	pdf	logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer runtime)
	dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
	rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
	nrust	the number of posterior samples used in the RUST calculations
	debug	logical for turning on debug messages
	aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
	n	the number of random samples required
	mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
	у	a vector of values at which to calculate the density and distribution functions

### Value

q\*\*\*\* returns a list containing at least the following:

• ml\_params: maximum likelihood estimates for the parameters.

- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

#### For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

#### r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

#### d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

#### p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

#### t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The GEV distribution with known shape has distribution function

$$F(x; \mu, \sigma) = \exp(-t(x; \mu, \sigma))$$

where

$$t(x; \mu, \sigma) = \begin{cases} \left[1 + \xi \left(\frac{x - \mu}{\sigma}\right)\right]^{-1/\xi} & \text{if } \xi \neq 0\\ \exp\left(-\frac{x - \mu}{\sigma}\right) & \text{if } \xi = 0 \end{cases}$$

where x is the random variable,  $\mu, \sigma > 0$  are the parameters and  $\xi$  is known (hence the k3 in the name).

The calibrating prior we use is given by

$$\pi(\mu,\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\* optionally returns the following:

If rust=TRUE:

 ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Optional Return Values (EVT models only)**

q\*\*\*\* optionally returns the following, for EVT models only:

• cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),

- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
kshape=-0.4
x=fitdistcp::d53gev_k3_example_data_v1
p=c(1:9)/10
q=qgev_k3_cp(x,p,kshape=kshape,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from ggev_k3_cp)",
main="GEV: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
muhat=q$ml_params[1]
sghat=q$ml_params[2]
xi=kshape
qmax=ifelse(xi<0,muhat-sghat/xi,Inf)
cat(" ml_params=",q$ml_params,",")
cat(" qmax=",qmax,"\n")
```

gev\_k3\_f1f 253

gev_k3_f1	f
-----------	---

DMGS equation 3.3, f1 term

# Description

DMGS equation 3.3, f1 term

#### Usage

```
gev_k3_f1f(y, v1, d1, v2, fd2, kshape)
```

# Arguments

у	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter

#### Value

Matrix

gev	トコ	£1	£~
gev	K.3	ΤI	Τā

The first derivative of the density

# Description

The first derivative of the density

#### Usage

```
gev_k3_f1fa(x, v1, v2, kshape)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
kshape	the known shape parameter

#### Value

Vector

254 gev\_k3\_f2fa

gev_	k3	f2f
5C V _	ヘンニ	. 1 – 1

DMGS equation 3.3, f2 term

# Description

DMGS equation 3.3, f2 term

#### Usage

```
gev_k3_f2f(y, v1, d1, v2, fd2, kshape)
```

# Arguments

у	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter

#### Value

3d array

gev	ト3	f2f	ີລ

The second derivative of the density

# Description

The second derivative of the density

#### Usage

```
gev_k3_f2fa(x, v1, v2, kshape)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter

kshape the known shape parameter

#### Value

gev\_k3\_fd 255

gev_k3_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_k3_fd(x, v1, v2, v3)
```

### Arguments

x a vector of training data values	X	a vector of training	data values
------------------------------------	---	----------------------	-------------

v1 first parameterv2 second parameterv3 third parameter

# Value

Vector

gev_k3_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

#### Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_k3_fdd(x, v1, v2, v3)
```

#### Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

256 gev\_k3\_ldda

gev_	k3	1dd
5 C V _		_ T G G

Second derivative matrix of the normalized log-likelihood

#### Description

Second derivative matrix of the normalized log-likelihood

#### Usage

```
gev_k3_ldd(x, v1, d1, v2, fd2, kshape)
```

#### **Arguments**

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

eter

kshape the known shape parameter

#### Value

Square scalar matrix

gev k3 1	_   _   _
OPV K3 I	ana

The second derivative of the normalized log-likelihood

#### Description

The second derivative of the normalized log-likelihood

#### Usage

```
gev_k3_ldda(x, v1, v2, kshape)
```

### Arguments

x a vector of training data value	es
-----------------------------------	----

v1 first parameter v2 second parameter

kshape the known shape parameter

#### Value

gev\_k3\_lddd 257

gev_k3_lddd	Third derivative tensor of the normalized log-likelihood
_	·

# Description

Third derivative tensor of the normalized log-likelihood

# Usage

```
gev_k3_lddd(x, v1, d1, v2, fd2, kshape)
```

#### Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter

#### Value

Cubic scalar array

gev_k3_lddda	The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

### Usage

```
gev_k3_lddda(x, v1, v2, kshape)
```

# Arguments

Χ	a vector of training data values
v1	first parameter
v2	second parameter
kshape	the known shape parameter

#### Value

3d array

258 gev\_k3\_lmnp

gev_k3_lmn	One component of the second derivative of the normalized log-likelihood

# Description

One component of the second derivative of the normalized log-likelihood

#### Usage

```
gev_k3_lmn(x, v1, d1, v2, fd2, kshape, mm, nn)
```

# Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

#### Value

Scalar value

gev_k3_lmnp	One component of the third derivative of the normalized log-likelihood

# Description

One component of the third derivative of the normalized log-likelihood

# Usage

```
gev_k3_lmnp(x, v1, d1, v2, fd2, kshape, mm, nn, rr)
```

gev\_k3\_logf 259

#### **Arguments**

Χ	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
	•

v2 second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

kshape the known shape parameter

nn an index for which derivative to calculate
nn an index for which derivative to calculate
rr an index for which derivative to calculate

#### Value

Scalar value

gev_k3_logf	Logf for RUST

#### Description

Logf for RUST

#### Usage

```
gev_k3_logf(params, x, kshape)
```

#### Arguments

params model parameters for calculating logf
x a vector of training data values
kshape the known shape parameter

### Value

Scalar value.

260 gev\_k3\_logfddd

gev_k3_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_k3_logfdd(x, v1, v2, v3)
```

### Arguments

x a ve	tor of training data values
--------	-----------------------------

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

gev_k3_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

#### Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_k3_logfddd(x, v1, v2, v3)
```

#### Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

3d array

gev\_k3\_loglik 261

gev_k3_loglik log-likelinooa functio	log-likelihood function	gev_k3_loglik
--------------------------------------	-------------------------	---------------

#### **Description**

log-likelihood function

#### Usage

```
gev_k3_loglik(vv, x, kshape)
```

#### **Arguments**

vv parameters

x a vector of training data valueskshape the known shape parameter

#### Value

Scalar value.

gev_k3_means	MLE and RHP means	
--------------	-------------------	--

# Description

MLE and RHP means

### Usage

```
gev_k3_means(means, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2, kshape)
```

#### **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data
dim number of parameters
kshape the known shape parameter

gev\_k3\_mu1fa

#### Value

Two scalars

gev_k3_mu1f	DMGS equation 3.3, mu1 term
-------------	-----------------------------

# Description

DMGS equation 3.3, mu1 term

#### Usage

```
gev_k3_mu1f(alpha, v1, d1, v2, fd2, kshape)
```

### Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter

#### Value

Matrix

gev_k3_mu1fa	
--------------	--

# Description

Minus the first derivative of the cdf, at alpha

#### Usage

```
gev_k3_mu1fa(alpha, v1, v2, kshape)
```

# Arguments

alpha	a vector of v	alues of alpha (o	one minus probabili	ty)
-------	---------------	-------------------	---------------------	-----

v1 first parameter v2 second parameter

kshape the known shape parameter

gev\_k3\_mu2f 263

#### Value

Vector

gev_k3_mu2f	DMGS equation 3.3, mu2 term	

# Description

DMGS equation 3.3, mu2 term

#### Usage

```
gev_k3_mu2f(alpha, v1, d1, v2, fd2, kshape)
```

### Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter

#### Value

3d array

gev_k3_mu2fa <i>N</i>	linus the second derivative of the cdf, at alpha
-----------------------	--------------------------------------------------

# Description

Minus the second derivative of the cdf, at alpha

#### Usage

```
gev_k3_mu2fa(alpha, v1, v2, kshape)
```

#### Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

kshape the known shape parameter

264 *gev\_k3\_pdd* 

#### Value

Matrix

gev_k3_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_k3_pd(x, v1, v2, v3)
```

#### Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

gev_k3_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_k3_pdd(x, v1, v2, v3)
```

,	
Х	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

gev\_k3\_waic 265

#### Value

Matrix

gev\_k3\_waic Waic

# Description

Waic

#### Usage

```
gev_k3_waic(
   waicscores,
   x,
   v1hat,
   d1,
   v2hat,
   fd2,
   kshape,
   lddi,
   lddd,
   lambdad,
   aderivs
)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter
kshape	the known shape parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

266 gev\_lda

gev\_ld12a

The combined derivative of the normalized log-likelihood

#### Description

The combined derivative of the normalized log-likelihood

#### Usage

```
gev_ld12a(x, v1, v2, v3)
```

#### Arguments

x a vector of training data values

v1 first parameterv2 second parameterv3 third parameter

#### Value

3d array

gev\_lda

The first derivative of the normalized log-likelihood

#### Description

The first derivative of the normalized log-likelihood

### Usage

```
gev_lda(x, v1, v2, v3)
```

#### Arguments

Y	a vector	of training	data values
Λ	a vector	or training	uata varues

v1 first parameterv2 second parameterv3 third parameter

#### Value

Vector

gev\_ldd 267

gev_ldd Second derivative matrix of the normalized log-likelihood
-------------------------------------------------------------------

# Description

Second derivative matrix of the normalized log-likelihood

#### Usage

```
gev_ldd(x, v1, d1, v2, fd2, v3, d3)
```

# Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

### Value

Square scalar matrix

gev_ldda The second derivative of the normalized log-likelihood	gev_ldda	The second derivative of the normalized log-likelihood	
-----------------------------------------------------------------	----------	--------------------------------------------------------	--

# Description

The second derivative of the normalized log-likelihood

#### Usage

```
gev_ldda(x, v1, v2, v3)
```

x	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

268 gev\_lddda

#### Value

Matrix

gev\_lddd Third deriv

 $Third\ derivative\ tensor\ of\ the\ normalized\ log-likelihood$ 

# Description

Third derivative tensor of the normalized log-likelihood

#### Usage

```
gev_lddd(x, v1, d1, v2, fd2, v3, d3)
```

# Arguments

Χ	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

#### Value

Cubic scalar array

gev\_lddda

The third derivative of the normalized log-likelihood

### Description

The third derivative of the normalized log-likelihood

### Usage

```
gev_lddda(x, v1, v2, v3)
```

gev\_lmn 269

# Arguments

Χ	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

3d array

gev_lmn	One component of the second derivative of the normalized log-likelihood

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
gev_lmn(x, v1, d1, v2, fd2, v3, d3, mm, nn)
```

# Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

#### Value

Scalar value

270 gev\_logf

gev_lmnp One component of the second derivative of the normalized log likelihood	gev_lmnp	One component of the second derivative of the normalized log-likelihood
----------------------------------------------------------------------------------	----------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
gev_lmnp(x, v1, d1, v2, fd2, v3, d3, mm, nn, rr)
```

# Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

#### Value

Scalar value

gev_logf Logf	for RUST
---------------	----------

# Description

Logf for RUST

# Usage

```
gev_logf(params, x)
```

gev\_logfd 271

#### **Arguments**

params model parameters for calculating logf x a vector of training data values

#### Value

Scalar value.

gev\_logfd First derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Description

First derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_logfd(x, v1, v2, v3)
```

#### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameterv3 third parameter

### Value

Vector

gev_logfdd	Second derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

#### **Description**

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gev_logfdd(x, v1, v2, v3)
```

272 gev\_logfddd

# Arguments

X	a vector of training data values
v1	first parameter

v2 second parameter

v3 third parameter

#### Value

Matrix

gev_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gev_logfddd(x, v1, v2, v3)
```

#### Arguments

X	a vector of	of training	data value	S
---	-------------	-------------	------------	---

v1 first parameter

v2 second parameter

v3 third parameter

#### Value

3d array

gev\_loglik 273

gev\_loglik

log-likelihood function

#### **Description**

log-likelihood function

#### Usage

```
gev_loglik(vv, x)
```

#### **Arguments**

vv parameters x a vector of training data values

#### Value

Scalar value.

gev\_means

Analytical Expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

# Description

Analytical Expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

#### Usage

```
gev_means(
   means,
   ml_params,
   lddi,
   lddi_k3,
   lddd_k3,
   lambdad_flat,
   lambdad_rh_mle,
   lambdad_rp,
   lambdad_ip,
   lambdad_custom,
   nx,
   dim = 3
)
```

gev\_mu1f

#### Arguments

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

1ddi inverse observed information matrix

1ddi\_k3 inverse observed information matrix, fixed shape parameter

1ddd third derivative of log-likelihood

1ddd\_k3 third derivative of log-likelihood, fixed shape parameter

lambdad\_flat derivative of the log flat prior

lambdad\_rh\_mle derivative of the log CRHP-MLE prior

lambdad\_rh\_flat

derivative of the log CRHP-FLAT prior

lambdad\_jp derivative of the log JP prior

lambdad\_custom custom value of the derivative of the log prior

nx length of training data dim number of parameters

#### Value

Two scalars

gev_mu1f DM	GS equation 3.3, mu1 term
-------------	---------------------------

#### **Description**

DMGS equation 3.3, mu1 term

#### Usage

```
gev_mu1f(alpha, v1, d1, v2, fd2, v3, d3)
```

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

gev\_mu1fa 275

#### Value

Matrix

gev_mu1fa Minus the first der	rivative of the cdf, at alpha
-------------------------------	-------------------------------

# Description

Minus the first derivative of the cdf, at alpha

# Usage

```
gev_mu1fa(alpha, v1, v2, v3)
```

# Arguments

alpha	alpha a vector of values of alpha (one minus probabilit	
v1	first parameter	
v2	second parameter	
v3	third parameter	

#### Value

Vector

gev_mu2f	DMGS equation 3.3, mu2 term

# Description

DMGS equation 3.3, mu2 term

# Usage

```
gev_mu2f(alpha, v1, d1, v2, fd2, v3, d3)
```

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

276 gev\_p123\_checkmle

#### Value

3d array

gev\_mu2fa

Minus the second derivative of the cdf, at alpha

### Description

Minus the second derivative of the cdf, at alpha

### Usage

```
gev_mu2fa(alpha, v1, v2, v3)
```

### Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

gev\_p123\_checkmle

Check MLE

#### Description

Check MLE

#### Usage

```
gev_p123_checkmle(ml_params, minxi, maxxi, t1, t2, t3)
```

ml_params	parameters
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape

#### Value

No return value (just a message to the screen).

gev\_p123\_cp Generalized Extreme Value Distribution with Three Predictors, Predictions based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qgev_p123_cp(
    x,
    t1,
    t2,
    t3,
    t01 = NA,
    t02 = NA,
    t03 = NA,
    n01 = NA,
    n02 = NA,
    n03 = NA,
    p = seq(0.1, 0.9, 0.1),
```

```
ics = c(0, 0, 0, 0, 0, 0),
 d1 = 0.01,
 d2 = 0.01,
 d3 = 0.01,
 d4 = 0.01,
 d5 = 0.01,
 d6 = 0.01,
  fdalpha = 0.01,
 minxi = -0.45,
 maxxi = 0.45,
 means = FALSE,
 waicscores = FALSE,
  extramodels = FALSE,
 pdf = FALSE,
 dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
  centering = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
rgev_p123_cp(
 n,
 х,
  t1,
  t2,
  t3,
  t01 = NA,
  t02 = NA,
  t03 = NA,
 n01 = NA,
 n02 = NA,
 n03 = NA,
  ics = c(0, 0, 0, 0, 0, 0),
 d1 = 0.01,
 d2 = 0.01,
 d3 = 0.01,
 d4 = 0.01,
 d5 = 0.01,
 d6 = 0.01,
 minxi = -0.45,
 maxxi = 0.45,
 extramodels = FALSE,
  rust = FALSE,
 mlcp = TRUE,
  centering = TRUE,
  debug = FALSE,
```

```
aderivs = TRUE
dgev_p123_cp(
 Х,
  t1,
  t2,
  t3,
  t01 = NA,
  t02 = NA,
 t03 = NA,
 n01 = NA
 n02 = NA,
 n03 = NA,
 y = x,
  ics = c(0, 0, 0, 0, 0, 0),
 d1 = 0.01,
 d2 = 0.01,
 d3 = 0.01,
 d4 = 0.01,
 d5 = 0.01,
 d6 = 0.01,
 minxi = -0.45,
 maxxi = 0.45,
 extramodels = FALSE,
 rust = FALSE,
  nrust = 10,
  centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
pgev_p123_cp(
 Х,
  t1,
  t2,
  t3,
  t01 = NA,
  t02 = NA
  t03 = NA,
 n01 = NA,
 n02 = NA,
 n03 = NA,
 y = x,
 ics = c(0, 0, 0, 0, 0, 0),
 d1 = 0.01,
 d2 = 0.01,
 d3 = 0.01,
```

```
d4 = 0.01,
 d5 = 0.01,
 d6 = 0.01,
 minxi = -0.45,
 \max xi = 0.45,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 1000,
  centering = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
tgev_p123_cp(
 n,
 х,
  t1,
  t2,
  t3,
 ics = c(0, 0, 0, 0, 0, 0),
 d1 = 0.01,
 d2 = 0.01,
 d3 = 0.01,
 d4 = 0.01,
 d5 = 0.01,
 d6 = 0.01,
 extramodels = FALSE,
 debug = FALSE
)
```

Х	a vector of training data values
t1	a vector of predictors for the mean, such that $length(t1)=length(x)$
t2	a vector of predictors for the sd, such that length(t2)=length(x)
t3	a vector of predictors for the shape, such that length(t3)=length(x)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
n01	an index for the predictor (specify either t01 or n01 but not both)
n02	an index for the predictor (specify either t02 or n02 but not both)
n03	an index for the predictor (specify either t03 or n03 but not both)
p	a vector of probabilities at which to generate predictive quantiles
ics	initial conditions for the maximum likelihood search
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter

281 gev\_p123\_cp d2 if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter d3 if aderivs=FALSE, the delta used for numerical derivatives with respect to the third parameter d4 if aderivs=FALSE, the delta used for numerical derivatives with respect to the fourth parameter d5 if aderivs=FALSE, the delta used for numerical derivatives with respect to the fifth parameter d6 if aderivs=FALSE, the delta used for numerical derivatives with respect to the sixth parameter fdalpha if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles minxi the minimum allowed value of the shape parameter (decrease with caution) the maximum allowed value of the shape parameter (increase with caution) maxxi means logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime) waicscores logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime) extramodels logical that indicates whether to run additional calculations and add three additional prediction models (longer runtime) pdf logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer runlogical that indicates whether DMGS calculations should be run or not (longer dmgs run time)

rust logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)

nrust the number of posterior samples used in the RUST calculations centering logical that indicates whether the predictor should be centered

debug logical for turning on debug messages

aderivs (for code testing only) logical for whether to use analytic derivatives (instead of

numerical). By default almost all models now use analytical derivatives.

n the number of random samples required

mlcp logical that indicates whether maxlik and parameter uncertainty calculations

should be performed (turn off to speed up RUST)

y a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

• ml\_params: maximum likelihood estimates for the parameters.

- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The GEV distribution with three predictors has distribution function

$$F(x; a_1, b_1, a_2, b_2, a_3, b_3) = \exp(-t(x; \mu(a_1, b_1), \sigma(a_2, b_2), \xi(a_3, b_3)))$$

where

$$t(x; \mu(a_1, b_1), \sigma(a_2, b_2), \xi(a_3, b_3)) = \begin{cases} \left[1 + \xi(a_3, b_3) \left(\frac{x - \mu(a_1, b_1)}{\sigma(a_2, b_2)}\right)\right]^{-1/\xi(a_3, b_3)} & \text{if } \xi(a_3, b_3) \neq 0 \\ \exp\left(-\frac{x - \mu(a_1, b_1)}{\sigma(a_2, b_2)}\right) & \text{if } \xi(a_3, b_3) = 0 \end{cases}$$

where x is the random variable,  $\mu=a_1+b_1t_1$  is the location parameter, modelled as a function of parameters  $a_1,b_1$  and predictor  $t_1,\,\sigma=e^{a_2+b_2t_2}$  is the scale parameter, modelled as a function of parameters  $a_2,b_2$  and predictor  $t_2$ , and  $\xi=a_3+b_3t_3$  is the shape parameter, modelled as a function of parameters  $a_3,b_3$  and predictor  $t_3$ .

The calibrating prior we use is given by

$$\pi(a_1, b_1, a_2, b_2, a_3, b_3) \propto 1$$

as given in Jewson et al. (2025).

The code will stop with an error if the input data gives a maximum likelihood value for the shape parameter that lies outside the range (minxi, maxxi), since outside this range there may be numerical problems. Such values seldom occur in real observed data for maxima.

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Optional Return Values (EVT models only)**

q\*\*\*\* optionally returns the following, for EVT models only:

• cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

#### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2024) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),

- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
# example 1
x=fitdistcp::d152gev_p123_example_data_v1_x
tt=fitdistcp::d152gev_p123_example_data_v1_t
t1=tt[,1]
t2=tt[,2]
t3=tt[,3]
p=c(1:9)/10
n01=10
n02=10
n03=10
q=qgev_p123_cp(x=x,t1=t1,t2=t2,t3=t3,n01=n01,n02=n02,n03=n03,t01=NA,t02=NA,t03=NA,p=p,rust=FALSE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
```

gev\_p123\_f1f 287

```
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgev_p123_cp)",
main="GEVD w/ p123: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
cat(" ml_params=",q$ml_params,"\n")
```

gev\_p123\_f1f

DMGS equation 2.1, f1 term, fixed shape parameter DMGS equation 2.1, f1 term

#### Description

DMGS equation 2.1, f1 term, fixed shape parameter DMGS equation 2.1, f1 term

#### Usage

```
gev_p123_f1f(y, t01, t02, t03, v1, d1, v2, d2, v3, d3, v4, d4, v5, d5, v6, d6)
```

#### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter
v6	sixth parameter
d6	the delta used in the numerical derivatives with respect to the parameter

#### Value

288 gev\_p123\_f2f

gev	p123	_f1fa

The first derivative of the density

# Description

The first derivative of the density

#### Usage

```
gev_p123_f1fa(x, t01, t02, t03, v1, v2, v3, v4, v5, v6)
```

#### Arguments

X	a vector of training data values
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

#### Value

Vector

gev\_p123\_f2f

GEVD-with-p1: DMGS equation 1.2 f2 term

#### Description

GEVD-with-p1: DMGS equation 1.2 f2 term

# Usage

```
gev_p123_f2f(y, t01, t02, t03, v1, d1, v2, d2, v3, d3, v4, d4, v5, d5, v6, d6)
```

gev\_p123\_f2fa 289

## Arguments

У	a vector of values at which to calculate the density and distribution functions
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter
v6	sixth parameter
d6	the delta used in the numerical derivatives with respect to the parameter

## Value

3d array

The second derivative of the density
--------------------------------------

## Description

The second derivative of the density

## Usage

```
gev_p123_f2fa(x, t01, t02, t03, v1, v2, v3, v4, v5, v6)
```

Х	a vector of training data values
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
v1	first parameter
v2	second parameter

290 gev\_p123\_fd

v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

Matrix

gev_p123_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p123_fd(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

### **Arguments**

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

Vector

gev\_p123\_fdd 291

gev_p123_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### **Description**

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p123_fdd(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

Matrix

gev_p123_ldd	Second derivative matrix of the normalized log-likelihood	

## Description

Second derivative matrix of the normalized log-likelihood

```
gev_p123_ldd(x, t1, t2, t3, v1, d1, v2, d2, v3, d3, v4, d4, v5, d5, v6, d6)
```

292 gev\_p123\_ldda

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter
v6	sixth parameter
d6	the delta used in the numerical derivatives with respect to the parameter

## Value

Square scalar matrix

gev_p123_ldda	The second derivative of the normalized log-likelihood

## Description

The second derivative of the normalized log-likelihood

## Usage

```
gev_p123_ldda(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter

gev\_p123\_lddd 293

v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

Matrix

gev_p123_1ddd	Third derivative tensor of the normalized log-likelihood, with fixed shape parameter
	• •

## Description

Third derivative tensor of the normalized log-likelihood, with fixed shape parameter

### Usage

```
gev_p123_lddd(x, t1, t2, t3, v1, d1, v2, d2, v3, d3, v4, d4, v5, d5, v6, d6)
```

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter
v6	sixth parameter
d6	the delta used in the numerical derivatives with respect to the parameter

#### Value

Cubic scalar array

294 gev\_p123\_lmn

gev_p123_1ddda	f the normalized log-likelihood
----------------	---------------------------------

# Description

The third derivative of the normalized log-likelihood

## Usage

```
gev_p123_lddda(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

## Arguments

Χ	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

3d array

gev_p123_lmn	One component of the second derivative of the normalized log-likelihood
--------------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

gev\_p123\_lmn 295

# Usage

```
gev_p123_lmn(
  Х,
  t1,
  t2,
  t3,
  ν1,
  d1,
  v2,
  d2,
  ν3,
  d3,
  v4,
  d4,
  v5,
  d5,
  v6,
  d6,
  mm,
  nn
)
```

x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter
v6	sixth parameter
d6	the delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

296 gev\_p123\_lmnp

### Value

Scalar value

gev_p123_lmnp	One component of the second derivative of the normalized log-likelihood
---------------	-------------------------------------------------------------------------

### Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
gev_p123_lmnp(
  х,
  t1,
  t2,
  t3,
  v1,
  d1,
  v2,
  d2,
  ν3,
  d3,
  v4,
  d4,
  v5,
  d5,
  v6,
  d6,
  mm,
  nn,
  rr
)
```

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

gev\_p123\_logf 297

v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter
v6	sixth parameter
d6	the delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

## Description

Logf for RUST

# Usage

```
gev_p123_logf(params, x, t1, t2, t3)
```

# Arguments

params	model parameters for calculating logf
X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape

### Value

Scalar value.

298 gev\_p123\_logfddd

Deriv() by Andrew Clausen and Serguei Sokol	gev_p123_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
---------------------------------------------	-----------------	------------------------------------------------------------------------------------------------------------------

### Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_p123_logfdd(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

### Arguments

x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

Matrix

Deriv() by Andrew Clausen and Serguei Sokol	gev_p123_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
---------------------------------------------	------------------	-----------------------------------------------------------------------------------------------------------------

### Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p123_logfddd(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

gev\_p123\_loglik 299

### Arguments

х	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

3d array

# Description

observed log-likelihood function

## Usage

```
gev_p123_loglik(vv, x, t1, t2, t3)
```

## Arguments

VV	parameters
X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape

### Value

Scalar value.

300 gev\_p123\_mu1f

expectation of Biros equation 2.1	gev_p123_means	Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1
-----------------------------------	----------------	----------------------------------------------------------------------------------------------------

## Description

Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

## Usage

```
gev_p123_means(means, t01, t02, t03, ml_params, nx)
```

### Arguments

means	logical that indicates whether to return analytical estimates for the distribution means (longer runtime)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
ml_params	parameters
nx	length of training data

#### Value

Two scalars

```
gev_p123_mu1f GEVD-with-p1: DMGS equation 3.3 mu1 term
```

## Description

```
GEVD-with-p1: DMGS equation 3.3 mu1 term
```

```
gev_p123_mu1f(
    alpha,
    t01,
    t02,
    t03,
    v1,
    d1,
    v2,
```

gev\_p123\_mu1fa 301

```
d2,
v3,
d3,
v4,
d4,
v5,
d5,
v6,
d6
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter
v6	sixth parameter
d6	the delta used in the numerical derivatives with respect to the parameter

## Value

Matrix

gev_p123_mu1fa	Minus the first derivative of the cdf, at alpha	

# Description

Minus the first derivative of the cdf, at alpha

```
gev_p123_mu1fa(alpha, t01, t02, t03, v1, v2, v3, v4, v5, v6)
```

302 gev\_p123\_mu2f

# Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

## Value

Vector

gev\_p123\_mu2f

GEVD-with-p1: DMGS equation 3.3 mu2 term

# Description

GEVD-with-p1: DMGS equation 3.3 mu2 term

```
gev_p123_mu2f(
  alpha,
  t01,
  t02,
  t03,
  v1,
  d1,
  v2,
  d2,
  ν3,
  d3,
  v4,
  d4,
  ν5,
  d5,
  v6,
  d6
)
```

gev\_p123\_mu2fa 303

## Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter
v6	sixth parameter
d6	the delta used in the numerical derivatives with respect to the parameter

## Value

3d array

gev_p123_mu2fa	Minus the second derivative of the cdf, at alpha

## Description

Minus the second derivative of the cdf, at alpha

## Usage

```
gev_p123_mu2fa(alpha, t01, t02, t03, v1, v2, v3, v4, v5, v6)
```

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
v1	first parameter
v2	second parameter

304 gev\_p123\_pd

v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

Matrix

gev_p123_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by
	Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p123_pd(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

### Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

Vector

gev\_p123\_pdd 305

gev_p123_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_p123_pdd(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

### Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

#### Value

Matrix

```
gev_p123_predictordata
```

Predicted Parameter and Generalized Residuals

## Description

Predicted Parameter and Generalized Residuals

```
gev_p123_predictordata(x, t1, t2, t3, t01, t02, t03, params)
```

306 gev\_p123\_setics

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
params	model parameters for calculating logf

## Value

Two vectors

gev\_p123\_setics Set initial conditions

## Description

Set initial conditions

## Usage

```
gev_p123_setics(x, t1, t2, t3, ics)
```

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
ics	initial conditions for the maximum likelihood search

### Value

Vector

gev\_p123\_waic 307

gev\_p123\_waic

Waic

# Description

Waic

## Usage

```
gev_p123_waic(
  waicscores,
  Х,
  t1,
  t2,
  t3,
  v1h,
  d1,
  v2h,
  d2,
  v3h,
  d3,
  v4h,
  d4,
  v5h,
  d5,
  v6h,
  d6,
  lddi,
  lddd,
  lambdad,
  aderivs
)
```

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1h	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2h	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

308 gev\_p12k3\_f1f

v3h	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4h	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5h	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter
v6h	sixth parameter
d6	the delta used in the numerical derivatives with respect to the parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

## Value

Two numeric values.

gev_p12k3_f1f	DMGS equation 2.1, f1 term, fixed shape parameter
gev_p12k3_f1f	DMGS equation 2.1, f1 term, fixed shape parameter

# Description

DMGS equation 2.1, f1 term, fixed shape parameter

## Usage

```
gev_p12k3_f1f(y, t01, t02, v1, d1, v2, d2, v3, d3, v4, d4, v5)
```

У	a vector of values at which to calculate the density and distribution functions
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter

gev\_p12k3\_f1fa 309

### Value

Matrix

gev\_p12k3\_f1fa

The first derivative of the density

### Description

The first derivative of the density

### Usage

```
gev_p12k3_f1fa(x, t, v1, v2, v3, v4, kshape)
```

## Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

kshape the known shape parameter

#### Value

Vector

gev\_p12k3\_f2f

GEVD-with-p1: DMGS equation 1.2 f2 term

### Description

GEVD-with-p1: DMGS equation 1.2 f2 term

```
gev_p12k3_f2f(y, t01, t02, v1, d1, v2, d2, v3, d3, v4, d4, v5)
```

310 gev\_p12k3\_f2fa

### Arguments

У	a vector of values at which to calculate the density and distribution functions
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter

### Value

3d array

## Description

The second derivative of the density

## Usage

```
gev_p12k3_f2fa(x, t, v1, v2, v3, v4, kshape)
```

## Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
kshape	the known shape parameter

#### Value

Matrix

gev\_p12k3\_fd 311

gev_p12k3_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p12k3_fd(x, t1, t2, v1, v2, v3, v4, v5)
```

### Arguments

Х	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

#### Value

Vector

gev_p12k3_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p12k3_fdd(x, t1, t2, v1, v2, v3, v4, v5)
```

312 gev\_p12k3\_ldd

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Matrix

gev_p12k3_ldd	Second derivative matrix of the normalized log-likelihood, with fixed shape parameter
---------------	---------------------------------------------------------------------------------------

## Description

Second derivative matrix of the normalized log-likelihood, with fixed shape parameter

### Usage

```
gev_p12k3_ldd(x, t1, t2, v1, d1, v2, d2, v3, d3, v4, d4, v5)
```

## Arguments

Χ	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter

#### Value

Square scalar matrix

gev\_p12k3\_ldda 313

ge v	p12k3 <sub>.</sub>	Idda	

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

### Usage

```
gev_p12k3_ldda(x, t, v1, v2, v3, v4, kshape)
```

## Arguments

X	a vector of training data values			
t	a vector or matrix of predictors			
v1	first parameter			
v2	second parameter			
v3	third parameter			
v4	fourth parameter			
kshape	the known shape parameter			

#### Value

Matrix

gev_p12k3_lddd	Third derivative is	tensor of the	normalized	log-likelihood,	with fixed
	shape parameter				

### Description

Third derivative tensor of the normalized log-likelihood, with fixed shape parameter

```
gev_p12k3_lddd(x, t1, t2, v1, d1, v2, d2, v3, d3, v4, d4, v5)
```

314 gev\_p12k3\_lddda

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter

### Value

Cubic scalar array

gev_p12k3_lddda	The third derivative of the normalized log-likelihood
-----------------	-------------------------------------------------------

## Description

The third derivative of the normalized log-likelihood

## Usage

```
gev_p12k3_lddda(x, t, v1, v2, v3, v4, kshape)
```

## Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
kshape	the known shape parameter

### Value

3d array

gev\_p12k3\_logfdd 315

gev_p12k3_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
gev_p12k3_logfdd	

### Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p12k3_logfdd(x, t1, t2, v1, v2, v3, v4, v5)
```

### Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

#### Value

Matrix

gev_p12k3_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p12k3_logfddd(x, t1, t2, v1, v2, v3, v4, v5)
```

316 gev\_p12k3\_mu1f

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

3d array

 ${\it gev\_p12k3\_mu1f} \qquad \qquad {\it GEVD-with-p1:DMGS\ equation\ 3.3\ mu1\ term, fixed\ shape\ parameter}$ 

## Description

GEVD-with-p1: DMGS equation 3.3 mu1 term, fixed shape parameter

### Usage

```
gev_p12k3_mu1f(alpha, t01, t02, v1, d1, v2, d2, v3, d3, v4, d4, v5)
```

### Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter

#### Value

Matrix

gev\_p12k3\_mu1fa 317

σeν	n12k3	_mu1fa
gcv_	_レι∠トン	_IIIUIIa

Minus the first derivative of the cdf, at alpha

## Description

Minus the first derivative of the cdf, at alpha

### Usage

```
gev_p12k3_mu1fa(alpha, t, v1, v2, v3, v4, kshape)
```

#### **Arguments**

alpha	a vector of values of alpha (one minus probability)
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
kshape	the known shape parameter

### Value

Vector

gev\_p12k3\_mu2f

GEVD-with-p1: DMGS equation 3.3 mu2 term, fixed shape parameter

### Description

GEVD-with-p1: DMGS equation 3.3 mu2 term, fixed shape parameter

```
gev_p12k3_mu2f(alpha, t01, t02, v1, d1, v2, d2, v3, d3, v4, d4, v5)
```

318 gev\_p12k3\_mu2fa

#### **Arguments**

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter

### Value

3d array

gev_p12k3_mu2fa	Minus the second derivative of the cdf, at alpha
	· · · · · · · · · · · · · · · · · · ·

## Description

Minus the second derivative of the cdf, at alpha

## Usage

```
gev_p12k3_mu2fa(alpha, t, v1, v2, v3, v4, kshape)
```

### Arguments

alpha	a vector of values of alpha (one minus probability)
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
kshape	the known shape parameter

#### Value

Matrix

gev\_p12k3\_pd 319

gev_p12k3_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p12k3_pd(x, t1, t2, v1, v2, v3, v4, v5)
```

### Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

#### Value

Vector

gev_p12k3_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p12k3_pdd(x, t1, t2, v1, v2, v3, v4, v5)
```

320 gev\_p12\_checkmle

### Arguments

x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

#### Value

Matrix

## Description

Check MLE

### Usage

```
gev_p12_checkmle(ml_params, minxi, maxxi)
```

## Arguments

ml\_params parameters

minxi minimum value of shape parameter xi maxxi maximum value of shape parameter xi

### Value

No return value (just a message to the screen).

gev\_p12\_cp

Generalized Extreme Value Distribution with Two Predictors, Predictions based on a Calibrating Prior

#### Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qgev_p12_cp(
    x,
    t1,
    t2,
    t01 = NA,
    t02 = NA,
    n01 = NA,
    n02 = NA,
    p = seq(0.1, 0.9, 0.1),
    ics = c(0, 0, 0, 0, 0),
    d1 = 0.01,
    d2 = 0.01,
    d3 = 0.01,
    d4 = 0.01,
    d5 = 0.01,
```

```
fdalpha = 0.01,
 minxi = -0.45,
 \max xi = 0.45,
 means = FALSE,
 waicscores = FALSE,
  extramodels = FALSE,
 pdf = FALSE,
 dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 predictordata = TRUE,
  centering = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
rgev_p12_cp(
 n,
 х,
  t1,
  t2,
  t01 = NA,
  t02 = NA,
 n01 = NA,
 n02 = NA,
 ics = c(0, 0, 0, 0, 0),
 d1 = 0.01,
 d2 = 0.01,
 d3 = 0.01,
 d4 = 0.01,
 d5 = 0.01,
 minxi = -0.45,
 \max xi = 0.45,
 extramodels = FALSE,
  rust = FALSE,
 mlcp = TRUE,
  centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
dgev_p12_cp(
 Х,
  t1,
  t2,
  t01 = NA,
  t02 = NA,
  n01 = NA,
```

```
n02 = NA,
 y = x,
  ics = c(0, 0, 0, 0, 0),
 d1 = 0.01,
 d2 = 0.01,
 d3 = 0.01,
 d4 = 0.01,
 d5 = 0.01,
 minxi = -0.45,
 maxxi = 0.45,
 extramodels = FALSE,
  rust = FALSE,
 nrust = 10,
  centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
)
pgev_p12_cp(
 Х,
  t1,
 t2,
  t01 = NA,
  t02 = NA,
 n01 = NA,
 n02 = NA,
 y = x,
  ics = c(0, 0, 0, 0, 0),
 d1 = 0.01,
 d2 = 0.01,
 d3 = 0.01,
  d4 = 0.01,
  d5 = 0.01,
 minxi = -0.45,
 \max xi = 0.45,
  extramodels = FALSE,
  rust = FALSE,
  nrust = 1000,
  centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
)
tgev_p12_cp(
 n,
  Х,
  t1,
  t2,
```

```
ics = c(0, 0, 0, 0, 0),
d1 = 0.01,
d2 = 0.01,
d3 = 0.01,
d4 = 0.01,
d5 = 0.01,
extramodels = FALSE,
debug = FALSE
```

X	a vector of training data values
t1	a vector of predictors for the mean, such that $length(t1)=length(x)$
t2	a vector of predictors for the sd, such that $length(t2)=length(x)$
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
n01	an index for the predictor (specify either t01 or n01 but not both)
n02	an index for the predictor (specify either t02 or n02 but not both)
p	a vector of probabilities at which to generate predictive quantiles
ics	initial conditions for the maximum likelihood search
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
d2	if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter
d3	if aderivs=FALSE, the delta used for numerical derivatives with respect to the third parameter
d4	if aderivs=FALSE, the delta used for numerical derivatives with respect to the fourth parameter
d5	if aderivs=FALSE, the delta used for numerical derivatives with respect to the fifth parameter
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
minxi	the minimum allowed value of the shape parameter (decrease with caution)
maxxi	the maximum allowed value of the shape parameter (increase with caution)
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
extramodels	logical that indicates whether to run additional calculations and add three additional prediction models (longer runtime)

logical that indicates whether to run additional calculations and return density pdf functions evaluated at quantiles specified by the input probabilities (longer runlogical that indicates whether DMGS calculations should be run or not (longer dmgs run time) logical that indicates whether RUST-based posterior sampling calculations should rust be run or not (longer run time) nrust the number of posterior samples used in the RUST calculations logical that indicates whether predictordata should be calculated predictordata centering logical that indicates whether the predictor should be centered debug logical for turning on debug messages aderivs (for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives. the number of random samples required n mlcp logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)

#### Value

у

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.

a vector of values at which to calculate the density and distribution functions

- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The GEV distribution with two predictors has distribution function

$$F(x; a_1, b_1, a_2, b_2, \xi) = \exp(-t(x; \mu(a_1, b_1), \sigma(a_2, b_2), \xi))$$

where

$$t(x; \mu(a_1, b_1), \sigma(a_2, b_2), \xi) = \begin{cases} \left[ 1 + \xi \left( \frac{x - \mu(a_1, b_1)}{\sigma(a_2, b_2)} \right) \right]^{-1/\xi} & \text{if } \xi \neq 0 \\ \exp\left( -\frac{x - \mu(a_1, b_1)}{\sigma(a_2, b_2)} \right) & \text{if } \xi = 0 \end{cases}$$

where x is the random variable,  $\mu = a_1 + b_1 t_1$  is the location parameter, modelled as a function of parameters  $a_1, b_1$  and predictor  $t_1$ ,  $\sigma = e^{a_2 + b_2 t_2}$  is the scale parameter, modelled as a function of parameters  $a_2, b_2$  and predictor  $t_2$ , and  $\xi$  is the shape parameter.

The calibrating prior we use is given by

$$\pi(a_1, b_1, a_2, b_2, \xi) \propto 1$$

as given in Jewson et al. (2025).

The code will stop with an error if the input data gives a maximum likelihood value for the shape parameter that lies outside the range (minxi, maxxi), since outside this range there may be numerical problems. Such values seldom occur in real observed data for maxima.

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

#### If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

## Optional Return Values (EVT models only)

q\*\*\*\* optionally returns the following, for EVT models only:

• cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

#### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2024) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

· Cauchy (cauchy),

- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

330 gev\_p12\_f1f

#### **Examples**

```
# example 1
x=fitdistcp::d151gev_p12_example_data_v1_x
tt=fitdistcp::d151gev_p12_example_data_v1_t
t1=tt[,1]
t2=tt[,2]
p=c(1:9)/10
n01=10
n02=10
q=qgev_p12_cp(x=x,t1=t1,t2=t2,n01=n01,n02=n02,t01=NA,t02=NA,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgev_p12_cp)",
main="GEVD w/ p12: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue",lwd=2)
cat(" ml_params=",q$ml_params,"\n")
```

gev\_p12\_f1f

DMGS equation 2.1, f1 term

#### **Description**

DMGS equation 2.1, f1 term

#### Usage

```
gev_p12_f1f(y, t01, t02, v1, d1, v2, d2, v3, d3, v4, d4, v5, d5)
```

### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter

gev\_p12\_f1fa 331

### Value

Matrix

gev\_p12\_f1fa

The first derivative of the density

# Description

The first derivative of the density

### Usage

```
gev_p12_f1fa(x, t01, t02, v1, v2, v3, v4, v5)
```

## Arguments

X	a vector of training data values
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

#### Value

Vector

gev\_p12\_f2f

GEVD-with-p1: DMGS equation 1.2 f2 term

# Description

GEVD-with-p1: DMGS equation 1.2 f2 term

```
gev_p12_f2f(y, t01, t02, v1, d1, v2, d2, v3, d3, v4, d4, v5, d5)
```

332 gev\_p12\_f2fa

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter

# Value

3d array

# Description

The second derivative of the density

# Usage

```
gev_p12_f2fa(x, t01, t02, v1, v2, v3, v4, v5)
```

# Arguments

X	a vector of training data values
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Matrix

gev\_p12\_fd 333

gev_p12_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

### Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p12_fd(x, t1, t2, v1, v2, v3, v4, v5)
```

### Arguments

Χ	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

#### Value

Vector

gev_p12_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p12_fdd(x, t1, t2, v1, v2, v3, v4, v5)
```

334 gev\_p12\_ggd

# Arguments

x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Matrix

gev_p12_ggd	Derivative of information matrix, based on ldd
-------------	------------------------------------------------

# Description

Derivative of information matrix, based on ldd

# Usage

```
gev_p12_ggd(x, t1, t2, v1, d1, v2, d2, v3, d3, v4, d4, v5, d5)
```

# Arguments

Χ	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter

#### Value

Square scalar matrix

gev\_p12\_ldd 335

gev_p12_ldd Second derivative matrix of the normalized log-likelihood
-----------------------------------------------------------------------

# Description

Second derivative matrix of the normalized log-likelihood

### Usage

```
gev_p12_ldd(x, t1, t2, v1, d1, v2, d2, v3, d3, v4, d4, v5, d5)
```

### Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter

### Value

Square scalar matrix

gev_p12_	ldda	The second derivative of the normalized log-likelihood
gev_p12_	Iuua	The second derivative of the normalized log-tiketinood

### Description

The second derivative of the normalized log-likelihood

```
gev_p12_ldda(x, t1, t2, v1, v2, v3, v4, v5)
```

336 gev\_p12\_lddd

# Arguments

Х	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

# Value

#### Matrix

gev_p12_lddd	Third derivative tensor of the normalized log-likelihood, with fixed
	shape parameter

# Description

Third derivative tensor of the normalized log-likelihood, with fixed shape parameter

# Usage

```
gev_p12_lddd(x, t1, t2, v1, d1, v2, d2, v3, d3, v4, d4, v5, d5)
```

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter

### Value

Cubic scalar array

gev\_p12\_lddda 337

gev_	n 1	2 1	ا ا	145
gev_	וש	4_1	Luc	ıua

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

### Usage

```
gev_p12_lddda(x, t1, t2, v1, v2, v3, v4, v5)
```

### **Arguments**

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

#### Value

3d array

gev_p12_lmn	One component of the second derivative of the normalized log-
	likelihood

# Description

One component of the second derivative of the normalized log-likelihood

```
gev_p12_lmn(x, t1, t2, v1, d1, v2, d2, v3, d3, v4, d4, v5, d5, mm, nn)
```

338 gev\_p12\_lmnp

# Arguments

Х	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

### Value

Scalar value

gev_p12_lmnp	One component of the second derivative of the normalized log-
	likelihood

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
gev_p12_lmnp(x, t1, t2, v1, d1, v2, d2, v3, d3, v4, d4, v5, d5, mm, nn, rr)
```

# Arguments

Χ	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter

gev\_p12\_logf 339

d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

gev\_p12\_logf Logffor RUST

# Description

Logf for RUST

# Usage

```
gev_p12_logf(params, x, t1, t2)
```

# Arguments

params	model parameters for calculating logf
X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd

### Value

Scalar value.

340 gev\_p12\_logfddd

gev_p12_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p12_logfdd(x, t1, t2, v1, v2, v3, v4, v5)
```

### Arguments

Χ	a vector of training data values	
t1	a vector of predictors for the mean	
t2	a vector of predictors for the sd	
v1	first parameter	
v2	second parameter	
v3	third parameter	
v4	fourth parameter	
v5	fifth parameter	

#### Value

Matrix

gev_p12_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	3,11,15

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p12_logfddd(x, t1, t2, v1, v2, v3, v4, v5)
```

gev\_p12\_loglik 341

## Arguments

x	a vector of training data values	
t1	a vector of predictors for the mean	
t2	a vector of predictors for the sd	
v1	first parameter	
v2	second parameter	
v3	third parameter	
v4	fourth parameter	
v5	fifth parameter	

### Value

3d array

gev\_p12\_loglik

observed log-likelihood function

# Description

observed log-likelihood function

# Usage

### Arguments

VV	parameters
x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd

#### Value

Scalar value.

342 gev\_p12\_mu1f

gev_p12_means	Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

# Description

Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

### Usage

```
gev_p12_means(means, t01, t02, ml_params, nx)
```

# Arguments

means	logical that indicates whether to return analytical estimates for the distribution means (longer runtime)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
ml_params	parameters
nx	length of training data

#### Value

Two scalars

gev_p12_mu1f	GEVD-with-p1: DMGS equation 3.3 mu1 term

# Description

```
GEVD-with-p1: DMGS equation 3.3 mu1 term
```

```
gev_p12_mu1f(alpha, t01, t02, v1, d1, v2, d2, v3, d3, v4, d4, v5, d5)
```

gev\_p12\_mu1fa 343

# Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter

# Value

Matrix

gev_p12_mu1fa	Minus the first derivative of the cdf, at alpha

# Description

Minus the first derivative of the cdf, at alpha

# Usage

```
gev_p12_mu1fa(alpha, t01, t02, v1, v2, v3, v4, v5)
```

# **Arguments**

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Vector

344 gev\_p12\_mu2fa

gev_p12_mu2f GEVD-with-p1:	DMGS equation 3.3 mu2 term
----------------------------	----------------------------

# Description

GEVD-with-p1: DMGS equation 3.3 mu2 term

### Usage

```
gev_p12_mu2f(alpha, t01, t02, v1, d1, v2, d2, v3, d3, v4, d4, v5, d5)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
v5	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter

### Value

3d array

gev_p12_mu2fa	Minus the second derivative of the cdf, at alpha

# Description

Minus the second derivative of the cdf, at alpha

```
gev_p12_mu2fa(alpha, t01, t02, v1, v2, v3, v4, v5)
```

gev\_p12\_pd 345

# Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

# Value

Matrix

gev_p12_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gev_p12_pd(x, t1, t2, v1, v2, v3, v4, v5)
```

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

Vector

gev_p12_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	by March Causen and Berguet Bokot

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p12_pdd(x, t1, t2, v1, v2, v3, v4, v5)
```

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Matrix

```
gev_p12_predictordata Predicted Parameter and Generalized Residuals
```

# Description

Predicted Parameter and Generalized Residuals

```
gev_p12_predictordata(predictordata, x, t1, t2, t01, t02, params)
```

gev\_p12\_setics 347

# Arguments

predictordata	logical that indicates whether to calculate and return predictordata
x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
params	model parameters for calculating logf

### Value

Two vectors

gev_p12_setics	Set initial conditions	
----------------	------------------------	--

# Description

Set initial conditions

# Usage

```
gev_p12_setics(x, t1, t2, ics)
```

# Arguments

Χ	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
ics	initial conditions for the maximum likelihood search

#### Value

Vector

348 *gev\_p12\_waic* 

gev\_p12\_waic

Waic

# Description

Waic

# Usage

```
gev_p12_waic(
  waicscores,
  Х,
  t1,
  t2,
  v1hat,
  d1,
  v2hat,
  d2,
  v3hat,
  d3,
  v4hat,
  d4,
  v5hat,
  d5,
  lddi,
  lddd,
  lambdad,
  aderivs
)
```

### Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3hat	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
v4hat	fourth parameter

d4	the delta used in the numerical derivatives with respect to the parameter
v5hat	fifth parameter
d5	the delta used in the numerical derivatives with respect to the parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

gev_p1k3_cp GEV Distribution with Known Shape with a Predictor, Predictions Based on a Calibrating Prior
----------------------------------------------------------------------------------------------------------

#### Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qgev_p1k3_cp(
  х,
  t,
  t0 = NA,
 n0 = NA,
 p = seq(0.1, 0.9, 0.1),
 d1 = 0.01,
  d2 = 0.01,
  fd3 = 0.01,
  fdalpha = 0.01,
  kshape = 0,
 means = FALSE,
 waicscores = FALSE,
  pdf = FALSE,
  dmgs = TRUE,
  rust = FALSE,
  nrust = 1e+05,
 predictordata = TRUE,
  centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
)
rgev_p1k3_cp(
  n,
 х,
 t,
 t0 = NA,
 n0 = NA,
 d1 = 0.01,
  d2 = 0.01,
  fd3 = 0.01,
  kshape = 0,
  rust = FALSE,
 mlcp = TRUE,
  centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
)
dgev_p1k3_cp(
 х,
  t,
  t0 = NA,
 n0 = NA,
  y = x,
 d1 = 0.01,
```

```
d2 = 0.01,
  fd3 = 0.01,
  kshape = 0,
  rust = FALSE,
 nrust = 1000,
  centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
)
pgev_p1k3_cp(
 х,
  t,
  t0 = NA,
 n0 = NA,
 y = x,
 d1 = 0.01,
  d2 = 0.01,
  fd3 = 0.01,
  kshape = 0,
  rust = FALSE,
 nrust = 1000,
  centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
tgev_p1k3_cp(
  n,
 Х,
  t,
 d1 = 0.01,
 d2 = 0.01,
  fd3 = 0.01,
  kshape = 0,
 debug = FALSE
)
```

#### **Arguments**

d2	if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter
fd3	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the third parameter
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
kshape	the known shape parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
pdf	logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer runtime)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The GEV distribution with known shape with a predictor has distribution function

$$F(x; a, b, \sigma) = \exp\left(-t(x; \mu(a, b), \sigma)\right)$$

where

$$t(x; a, b, \sigma) = \begin{cases} \left[1 + \xi \left(\frac{x - \mu(a, b)}{\sigma}\right)\right]^{-1/\xi} & \text{if } \xi \neq 0\\ \exp\left(-\frac{x - \mu(a, b)}{\sigma}\right) & \text{if } \xi = 0 \end{cases}$$

where x is the random variable,  $\mu = a + bt$  is the location parameter,  $\sigma > 0$  is the shape parameter and  $\xi$  is known (hence the k3 in the name).

The calibrating prior we use is given by

$$\pi(\mu,\sigma)\propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),

gev\_p1k3\_f1f 357

• Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d150gev_p1_example_data_v1_x #use data for 150
tt=fitdistcp::d150gev_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qgev_p1k3_cp(x=x,t=tt,n0=n0,t0=NA,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgev_p1k3_cp)",
main="GEVD w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue",lwd=2)
cat(" ml_params=",q$ml_params,"\n")
```

gev\_p1k3\_f1f

DMGS equation 2.1, f1 term, fixed shape parameter

#### **Description**

```
DMGS equation 2.1, f1 term, fixed shape parameter DMGS equation 2.1, f1 term
```

#### Usage

```
gev_p1k3_f1f(y, t0, v1, d1, v2, d2, v3, fd3, kshape)
gev_p1k3_f1f(y, t0, v1, d1, v2, d2, v3, fd3, kshape)
```

#### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter

358 *gev\_p1k3\_f1fa* 

d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter

# Value

Matrix

gev_p1k3_f1fa	The first derivative of the density

# Description

The first derivative of the density

# Usage

```
gev_p1k3_f1fa(x, t, v1, v2, v3, kshape)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
kshape	the known shape parameter

# Value

Vector

gev\_p1k3\_f2f 359

gev_p	1k3	_f2f
-------	-----	------

GEVD-with-p1: DMGS equation 1.2 f2 term

### Description

GEVD-with-p1: DMGS equation 1.2 f2 term

DMGS equation 2.1, f2 term

### Usage

```
gev_p1k3_f2f(y, t0, v1, d1, v2, d2, v3, fd3, kshape)
gev_p1k3_f2f(y, t0, v1, d1, v2, d2, v3, fd3, kshape)
```

### Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter

### Value

3d array

gev\_p1k3\_f2fa

The second derivative of the density

## Description

The second derivative of the density

```
gev_p1k3_f2fa(x, t, v1, v2, v3, kshape)
```

360 gev\_p1k3\_fd

### Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
kshape	the known shape parameter

### Value

Matrix

gev_p1k3_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gev_p1k3_fd(x, t, v1, v2, v3, v4)
```

# Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

### Value

Vector

gev\_p1k3\_fdd 361

gev_p1k3_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p1k3_fdd(x, t, v1, v2, v3, v4)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

gev_p1k3_ldd	Second derivative matrix of the normalized log-likelihood, with fixed
	shape parameter

### **Description**

Second derivative matrix of the normalized log-likelihood, with fixed shape parameter Second derivative matrix of the normalized log-likelihood

```
gev_p1k3_ldd(x, t, v1, d1, v2, d2, v3, fd3, kshape)
gev_p1k3_ldd(x, t, v1, d1, v2, d2, v3, fd3, kshape)
```

362 gev\_p1k3\_ldda

## Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter

## Value

Square scalar matrix

## Description

The second derivative of the normalized log-likelihood

## Usage

```
gev_p1k3_ldda(x, t, v1, v2, v3, kshape)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
kshape	the known shape parameter

## Value

Matrix

gev\_p1k3\_lddd 363

gev_p1k3_lddd	Third derivative tensor of the normalized log-likelihood, with fixed shape parameter
---------------	--------------------------------------------------------------------------------------

## Description

Third derivative tensor of the normalized log-likelihood, with fixed shape parameter Third derivative tensor of the normalized log-likelihood

## Usage

```
gev_p1k3_lddd(x, t, v1, d1, v2, d2, v3, fd3, kshape)
gev_p1k3_lddd(x, t, v1, d1, v2, d2, v3, fd3, kshape)
```

### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter

### Value

Cubic scalar array

gev_p1k3_lddda	The third derivative of the normalized log-likelihood

## Description

The third derivative of the normalized log-likelihood

```
gev_p1k3_lddda(x, t, v1, v2, v3, kshape)
```

364 gev\_p1k3\_lmn

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
kshape	the known shape parameter

### Value

3d array

gev_p1k3_lmn	One component of the second derivative of the normalized log-likelihood
--------------	-------------------------------------------------------------------------

## Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
gev_p1k3_lmn(x, t, v1, d1, v2, d2, v3, fd3, kshape, mm, nn)
```

## Arguments

х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

### Value

Scalar value

gev\_p1k3\_lmnp 365

gev_p1k3_lmnp	One component of the second derivative of the normalized log-likelihood
---------------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
gev_p1k3_lmnp(x, t, v1, d1, v2, d2, v3, fd3, kshape, mm, nn, rr)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

## Value

Scalar value

$gev_p1k3_logf$ $Logf for RUST$
---------------------------------

## Description

Logf for RUST

```
gev_p1k3_logf(params, x, t, kshape)
```

366 gev\_p1k3\_logfdd

#### **Arguments**

params	model parameters for calculating logf
Х	a vector of training data values
t	a vector or matrix of predictors

kshape the known shape parameter

#### Value

Scalar value.

gev\_p1k3\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p1k3_logfdd(x, t, v1, v2, v3, v4)
```

#### **Arguments**

х	a vector of training data values
t	a vector or matrix of predictors
	_

v1 first parameter
v2 second parameter
v3 third parameter
v4 fourth parameter

## Value

Matrix

gev\_p1k3\_logfddd 367

gev_p1k3_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p1k3_logfddd(x, t, v1, v2, v3, v4)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

### Value

3d array

gev_p1k3_loglik	GEV-with-known-shape-with-pl	l observed log-likelihood function
8 · · = · · · · · = = - · 8 = - · ·	· · · · · · · · · · · · · · · · · ·	

## Description

GEV-with-known-shape-with-p1 observed log-likelihood function

## Usage

```
gev_p1k3_loglik(vv, x, t, kshape)
```

# Arguments

VV	parameters
X	a vector of training data values
t	a vector or matrix of predictors
kshape	the known shape parameter

368 gev\_p1k3\_mu1f

### Value

Scalar value.

gev_p1k3_means	Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

### Description

Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

#### Usage

```
gev_p1k3_means(means, t0, ml_params, kshape, nx)
```

### Arguments

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

t0 a single value of the predictor (specify either t0 or n0 but not both)

ml\_params parameters

kshape the known shape parameter nx length of training data

#### Value

Two scalars

gev_p1k3_mu1f	GEVD-with-p1: DMGS equation 3.3 mu1 term, fixed shape parameter
---------------	-----------------------------------------------------------------

## Description

```
GEVD-with-p1: DMGS equation 3.3 mu1 term, fixed shape parameter DMGS equation 3.3, mu1 term
```

```
gev_p1k3_mu1f(alpha, t0, v1, d1, v2, d2, v3, fd3, kshape)
gev_p1k3_mu1f(alpha, t0, v1, d1, v2, d2, v3, fd3, kshape)
```

gev\_p1k3\_mu1fa 369

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter

## Value

Matrix

gev_p1k3_mu1fa	Minus the first derivative of the cdf, at alpha

## Description

Minus the first derivative of the cdf, at alpha

## Usage

```
gev_p1k3_mu1fa(alpha, t, v1, v2, v3, kshape)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
kshape	the known shape parameter

## Value

Vector

370 gev\_p1k3\_mu2fa

gev\_p1k3\_mu2f

GEVD-with-p1: DMGS equation 3.3 mu2 term, fixed shape parameter

## Description

GEVD-with-p1: DMGS equation 3.3 mu2 term, fixed shape parameter DMGS equation 3.3, mu2 term

## Usage

```
gev_p1k3_mu2f(alpha, t0, v1, d1, v2, d2, v3, fd3, kshape)
gev_p1k3_mu2f(alpha, t0, v1, d1, v2, d2, v3, fd3, kshape)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter

### Value

3d array

gev\_p1k3\_mu2fa

Minus the second derivative of the cdf, at alpha

## Description

Minus the second derivative of the cdf, at alpha

```
gev_p1k3_mu2fa(alpha, t, v1, v2, v3, kshape)
```

gev\_p1k3\_pd 371

## Arguments

alpha a vector of values of alpha (one minus probability)

t a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

kshape the known shape parameter

### Value

Matrix

gev\_p1k3\_pd First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p1k3_pd(x, t, v1, v2, v3, v4)
```

# Arguments

x a vector of training data values
t a vector or matrix of predictors
v1 first parameter
v2 second parameter
v3 third parameter

v4 fourth parameter

### Value

Vector

gev_p1k3_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p1k3_pdd(x, t, v1, v2, v3, v4)
```

### **Arguments**

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

```
gev_p1k3_predictordata
```

Predicted Parameter and Generalized Residuals

## Description

Predicted Parameter and Generalized Residuals

### Usage

```
gev_p1k3_predictordata(predictordata, x, t, t0, params, kshape)
```

## Arguments

predictordata	logical that indicates whether to calculate and return predictordata
X	a vector of training data values
t	a vector or matrix of predictors
t0	a single value of the predictor (specify either t0 or n0 but not both)
params	model parameters for calculating logf
kshape	the known shape parameter

gev\_p1k3\_waic 373

# Value

Two vectors

gev\_p1k3\_waic Waic

## Description

Waic

## Usage

```
gev_p1k3_waic(
  waicscores,
  х,
  t,
  v1hat,
  d1,
  v2hat,
  d2,
  v3hat,
  fd3,
  kshape,
  lddi,
  lddd,
  lambdad,
  aderivs
)
```

## Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
t	a vector or matrix of predictors
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3hat	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kshape	the known shape parameter
lddi	inverse observed information matrix

lddd third derivative of log-likelihood

lambdad derivative of the log prior

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

gev_p1_checkmle	Check MLE
-----------------	-----------

### Description

Check MLE

### Usage

```
gev_p1_checkmle(ml_params, minxi, maxxi)
```

### Arguments

ml\_params parameters

minxi minimum value of shape parameter xi maxxi maximum value of shape parameter xi

#### Value

No return value (just a message to the screen).

gev_p1_cp	Generalized Extreme Value Distribution with a Predictor, Predictions
	Based on a Calibrating Prior

## Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.

- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qgev_p1_cp(
 Х,
  t,
  t0 = NA,
 n0 = NA,
 p = seq(0.1, 0.9, 0.1),
  ics = c(0, 0, 0, 0),
 d1 = 0.01,
  d2 = 0.01,
  fd3 = 0.01,
  d4 = 0.01,
  fdalpha = 0.01,
 minxi = -0.45,
 maxxi = 0.45,
 means = FALSE,
 waicscores = FALSE,
 extramodels = FALSE,
 pdf = FALSE,
  dmgs = TRUE,
  rust = FALSE,
  nrust = 1e+05,
 predictordata = TRUE,
 centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
rgev_p1_cp(
 n,
 Х,
  t,
  t0 = NA,
  n0 = NA,
  ics = c(0, 0, 0, 0),
```

```
d1 = 0.01,
 d2 = 0.01,
  fd3 = 0.01,
 d4 = 0.01,
 minxi = -0.45,
 \max xi = 0.45,
 extramodels = FALSE,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
dgev_p1_cp(
 Х,
 t,
 t0 = NA,
 n0 = NA,
 y = x,
 ics = c(0, 0, 0, 0),
 d1 = 0.01,
 d2 = 0.01,
 fd3 = 0.01,
 d4 = 0.01,
 minxi = -0.45,
 maxxi = 0.45,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 1000,
  centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
pgev_p1_cp(
 Х,
 t,
 t0 = NA,
 n0 = NA,
 y = x,
 ics = c(0, 0, 0, 0),
 d1 = 0.01,
 d2 = 0.01,
 fd3 = 0.01,
 d4 = 0.01,
 minxi = -0.45,
 \max xi = 0.45,
 extramodels = FALSE,
```

```
rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
tgev_p1_cp(
 n,
 Х,
 t,
 ics = c(0, 0, 0, 0),
 d1 = 0.01,
 d2 = 0.01,
 fd3 = 0.01,
 d4 = 0.01,
 extramodels = FALSE,
 debug = FALSE
)
```

## Arguments

х	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles
ics	initial conditions for the maximum likelihood search
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
d2	if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter
fd3	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the third parameter
d4	if aderivs=FALSE, the delta used for numerical derivatives with respect to the fourth parameter
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
minxi	the minimum allowed value of the shape parameter (decrease with caution)
maxxi	the maximum allowed value of the shape parameter (increase with caution)
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)

extramodels logical that indicates whether to run additional calculations and add three additional prediction models (longer runtime) pdf logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer runlogical that indicates whether DMGS calculations should be run or not (longer dmgs run time) logical that indicates whether RUST-based posterior sampling calculations should rust be run or not (longer run time) the number of posterior samples used in the RUST calculations nrust logical that indicates whether predictordata should be calculated predictordata centering logical that indicates whether the predictor should be centered debug logical for turning on debug messages aderivs (for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives. n the number of random samples required mlcp logical that indicates whether maxlik and parameter uncertainty calculations

#### Value

У

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.

a vector of values at which to calculate the density and distribution functions

- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

should be performed (turn off to speed up RUST)

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- ullet adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The GEV distribution with a predictor has distribution function

$$F(x; a, b, \sigma, \xi) = \exp\left(-t(x; \mu(a, b), \sigma, \xi)\right)$$

where

$$t(x; \mu(a, b), \sigma, \xi) = \begin{cases} \left[1 + \xi \left(\frac{x - \mu(a, b)}{\sigma}\right)\right]^{-1/\xi} & \text{if } \xi \neq 0\\ \exp\left(-\frac{x - \mu(a, b)}{\sigma}\right) & \text{if } \xi = 0 \end{cases}$$

where x is the random variable,  $\mu = a + bt$  is the location parameter, modelled as a function of parameters a, b and predictor t, and  $\sigma > 0$ ,  $\xi$  are the scale and shape parameters.

The calibrating prior we use is given by

$$\pi(a,b,\sigma,\xi) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

The code will stop with an error if the input data gives a maximum likelihood value for the shape parameter that lies outside the range (minxi, maxxi), since outside this range there may be numerical problems. Such values seldom occur in real observed data for maxima.

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Optional Return Values (EVT models only)**

q\*\*\*\* optionally returns the following, for EVT models only:

• cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

### Optional Return Values (some EVT models only)

q\*\*\*\* optionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_quantiles: predictive quantiles calculated from Bayesian integration with a flat prior.
- rh\_ml\_quantiles: predictive quantiles calculated from Bayesian integration with the calibrating prior, and the maximmum likelihood estimate for the shape parameter.
- jp\_quantiles: predictive quantiles calculated from Bayesian integration with Jeffreys' prior.

r\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_deviates: predictive random deviates calculated using a Bayesian analysis with a flat prior.
- rh\_ml\_deviates: predictive random deviates calculated using a Bayesian analysis with the RHP-MLE prior.
- jp\_deviates: predictive random deviates calculated using a Bayesian analysis with the JP.

d\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_pdf: predictive density function from a Bayesian analysis with the flat prior.
- rh\_ml\_pdf: predictive density function from a Bayesian analysis with the RHP-MLE prior.
- jp\_pdf: predictive density function from a Bayesian analysis with the JP.

p\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_cdf: predictive distribution function from a Bayesian analysis with the flat prior.
- rh\_ml\_cdf: predictive distribution function from a Bayesian analysis with the RHP-MLE prior.
- jp\_cdf: predictive distribution function from a Bayesian analysis with the JP.

These additional predictive distributions are included for comparison with the calibrating prior model. They generally give less good reliability than the calibrating prior.

#### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2024) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

· Cauchy (cauchy),

- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

384 gev\_p1\_f1f

### **Examples**

```
# example 1
x=fitdistcp::d150gev_p1_example_data_v1_x
tt=fitdistcp::d150gev_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qgev_p1_cp(x=x,t=tt,n0=n0,t0=NA,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgev_p1_cp)",
main="GEVD w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue",lwd=2)
cat(" ml_params=",q$ml_params,"\n")
```

gev\_p1\_f1f

DMGS equation 2.1, f1 term

### **Description**

DMGS equation 2.1, f1 term

#### Usage

```
gev_p1_f1f(y, t0, v1, d1, v2, d2, v3, fd3, v4, d4)
```

### Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter

#### Value

Matrix

gev\_p1\_f1fa 385

gev	n1	f1	fa

The first derivative of the density

## Description

The first derivative of the density

## Usage

```
gev_p1_f1fa(x, t, v1, v2, v3, v4)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

### Value

Vector

gev\_p1\_f2f

GEVD-with-p1: DMGS equation 1.2 f2 term

## Description

```
GEVD-with-p1: DMGS equation 1.2 f2 term
```

## Usage

```
gev_p1_f2f(y, t0, v1, d1, v2, d2, v3, fd3, v4, d4)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

386 gev\_p1\_f2fa

v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter

# Value

3d array

|--|

# Description

The second derivative of the density

# Usage

```
gev_p1_f2fa(x, t, v1, v2, v3, v4)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

## Value

Matrix

gev\_p1\_fd 387

gev_p1_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	Trr() by Tanaren Clausen and Serguet Sener

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p1_fd(x, t, v1, v2, v3, v4)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

### Value

Vector

gev_p1_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	Tiv() by Andrew Cidusen and Serguet Sokot

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p1_fdd(x, t, v1, v2, v3, v4)
```

388 gev\_p1\_ggd

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

## Value

Matrix

gev_p1_ggd	Derivative of information matrix, based on ldd

# Description

Derivative of information matrix, based on ldd

## Usage

```
gev_p1_ggd(x, t, v1, d1, v2, d2, v3, fd3, v4, d4)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter

## Value

Square scalar matrix

gev\_p1\_ldd 389

gev_p1_ldd	Second derivative matrix of the normalized log-likelihood
gev_pi_iuu	Secona derivative mairix of the normalized log-liketinood

## Description

Second derivative matrix of the normalized log-likelihood

## Usage

```
gev_p1_ldd(x, t, v1, d1, v2, d2, v3, fd3, v4, d4)
```

## Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter

### Value

Square scalar matrix

gev_p1_ldda	The second derivative of the normalized log-likelihood

## Description

The second derivative of the normalized log-likelihood

```
gev_p1_ldda(x, t, v1, v2, v3, v4)
```

390 gev\_p1\_lddd

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

## Value

Matrix

shape parameter	gev_p1_lddd	Third derivative tensor of the normalized log-likelihood, with fixed shape parameter
-----------------	-------------	--------------------------------------------------------------------------------------

# Description

Third derivative tensor of the normalized log-likelihood, with fixed shape parameter

## Usage

```
gev_p1_lddd(x, t, v1, d1, v2, d2, v3, fd3, v4, d4)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter

## Value

Cubic scalar array

gev\_p1\_lddda 391

σων	n1	lddda
gev	υı	Tuuua

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

## Usage

```
gev_p1_lddda(x, t, v1, v2, v3, v4)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

### Value

3d array

gev_p1_lmn	One component of the second derivative of the normalized log-	
	likelihood	

## Description

One component of the second derivative of the normalized log-likelihood

```
gev_p1_lmn(x, t, v1, d1, v2, d2, v3, fd3, v4, d4, mm, nn)
```

392 gev\_p1\_lmnp

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the param-
	eter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

## Value

Scalar value

gev_p1_lmnp	One component of the second derivative of the normalized log- likelihood
	***************************************

## Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
gev_p1_lmnp(x, t, v1, d1, v2, d2, v3, fd3, v4, d4, mm, nn, rr)
```

## **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

gev\_p1\_logf 393

V4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate

nn an index for which derivative to calculate rr an index for which derivative to calculate

### Value

Scalar value

$gev_p1_logf$ $Logf for RUST$
-------------------------------

## Description

Logf for RUST

### Usage

```
gev_p1_logf(params, x, t)
```

## Arguments

params	model parameters for calculating logi
X	a vector of training data values
t	a vector or matrix of predictors

### Value

Scalar value.

gev_p1_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p1_logfdd(x, t, v1, v2, v3, v4)
```

394 gev\_p1\_logfddd

# Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

## Value

Matrix

gev_p1_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p1_logfddd(x, t, v1, v2, v3, v4)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

3d array

gev\_p1\_loglik 395

	1	- 7		л.	٠	i.
gev_	рι	_1	.og	Ί	1	K

observed log-likelihood function

### **Description**

observed log-likelihood function

### Usage

```
gev_p1_loglik(vv, x, t)
```

### **Arguments**

VV	parameters

x a vector of training data valuest a vector or matrix of predictors

#### Value

Scalar value.

gev\_p1\_means

Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

## Description

Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

```
gev_p1_means(
  means,
  t0,
  ml_params,
  lddi,
  lddi_k4,
  lddd,
  lddd_k4,
  lambdad_flat,
  lambdad_rh_mle,
  lambdad_rh_flat,
  lambdad_jp,
  nx,
  dim = 4
)
```

396 gev\_p1\_mu1f

### **Arguments**

means logical that indicates whether to return analytical estimates for the distribution means (longer runtime)

to a single value of the predictor (specify either to or no but not both)

ml\_params parameters

1ddi inverse observed information matrix

1ddi\_k4 inverse observed information matrix, fixed shape parameter

1ddd third derivative of log-likelihood

1ddd\_k4 third derivative of log-likelihood, fixed shape parameter

lambdad\_flat derivative of the log flat prior

lambdad\_rh\_mle derivative of the log CRHP-MLE prior

lambdad\_rh\_flat

derivative of the log CRHP-FLAT prior

lambdad\_jp derivative of the log JP prior
nx length of training data
dim number of parameters

#### Value

Two scalars

gev_p1_mu1f GEVD-with-p1: DMGS equation 3.3 mu1 term	
------------------------------------------------------	--

## Description

GEVD-with-p1: DMGS equation 3.3 mu1 term

### Usage

```
gev_p1_mu1f(alpha, t0, v1, d1, v2, d2, v3, fd3, v4, d4)
```

#### **Arguments**

aipna	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
V <del>1</del>	Tourtii parametei
d4	the delta used in the numerical derivatives with respect to the parameter

gev\_p1\_mu1fa 397

## Value

Matrix

gev\_p1\_mu1fa

Minus the first derivative of the cdf, at alpha

#### **Description**

Minus the first derivative of the cdf, at alpha

## Usage

```
gev_p1_mu1fa(alpha, t, v1, v2, v3, v4)
```

## Arguments

alpha a vector of values of alpha (one minus probability)
t a vector or matrix of predictors
v1 first parameter
v2 second parameter
v3 third parameter

fourth parameter

#### Value

Vector

ν4

gev\_p1\_mu2f

GEVD-with-p1: DMGS equation 3.3 mu2 term

## Description

GEVD-with-p1: DMGS equation 3.3 mu2 term

```
gev_p1_mu2f(alpha, t0, v1, d1, v2, d2, v3, fd3, v4, d4)
```

398 gev\_p1\_mu2fa

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
v4	fourth parameter
d4	the delta used in the numerical derivatives with respect to the parameter

#### Value

3d array

gev_p1_mu2fa	Minus the second derivative of the cdf, at alpha
--------------	--------------------------------------------------

## Description

Minus the second derivative of the cdf, at alpha

## Usage

```
gev_p1_mu2fa(alpha, t, v1, v2, v3, v4)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)	
t	a vector or matrix of predictors	
v1	first parameter	
v2	second parameter	
v3	third parameter	
v4	fourth parameter	

#### Value

Matrix

gev\_p1\_pd 399

gev_p1_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p1_pd(x, t, v1, v2, v3, v4)
```

## Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Vector

gev_p1_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p1_pdd(x, t, v1, v2, v3, v4)
```

400 gev\_p1\_predictordata

## Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

gev\_p1\_predictordata Predicted Parameter and Generalized Residuals

## Description

Predicted Parameter and Generalized Residuals

## Usage

```
gev_p1_predictordata(predictordata, x, t, t0, params)
```

## Arguments

predictordata	logical that indicates whether to calculate and return predictordata	
x	a vector of training data values	
t	a vector or matrix of predictors	
t0	a single value of the predictor (specify either t0 or n0 but not both)	
params	model parameters for calculating logf	

#### Value

Two vectors

gev\_p1\_setics 401

gev\_p1\_setics

Set initial conditions

## Description

Set initial conditions

## Usage

```
gev_p1_setics(x, t, ics)
```

## Arguments

x a vector of training data valuest a vector or matrix of predictors

ics initial conditions for the maximum likelihood search

## Value

Vector

gev\_p1\_waic

Waic

## Description

Waic

```
gev_p1_waic(
 waicscores,
 х,
  t,
 v1hat,
 d1,
 v2hat,
  d2,
  v3hat,
  fd3,
  v4hat,
  d4,
  lddi,
  lddd,
 lambdad,
  aderivs
)
```

402 gev\_pd

## Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)	
Χ	a vector of training data values	
t	a vector or matrix of predictors	
v1hat	first parameter	
d1	the delta used in the numerical derivatives with respect to the parameter	
v2hat	second parameter	
d2	the delta used in the numerical derivatives with respect to the parameter	
v3hat	third parameter	
fd3	the fractional delta used in the numerical derivatives with respect to the parameter	
v4hat	fourth parameter	
d4	the delta used in the numerical derivatives with respect to the parameter	
lddi	inverse observed information matrix	
lddd	third derivative of log-likelihood	
lambdad	derivative of the log prior	
aderivs	logical for whether to use analytic derivatives (instead of numerical)	

#### Value

Two numeric values.

gev_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	Andrew Clausen and Serguet Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_pd(x, v1, v2, v3)
```

## Arguments

X	a vector of training data values	
v1	first parameter	
v2	second parameter	
v3	third parameter	

gev\_pdd 403

#### Value

Vector

gev\_pdd

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_pdd(x, v1, v2, v3)
```

#### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameterv3 third parameter

## Value

Matrix

gev\_pwm\_params

PWM parameter estimation

## Description

PWM parameter estimation

#### Usage

```
gev_pwm_params(x)
```

### **Arguments**

x a vector of training data values

#### Value

Vector

404 gev\_waic

gev\_setics

Set initial conditions

## Description

Set initial conditions

## Usage

```
gev_setics(x, ics)
```

## Arguments

x a vector of training data values

ics initial conditions for the maximum likelihood search

## Value

Vector

gev\_waic

Waic

## Description

Waic

```
gev_waic(
  waicscores,
  x,
  v1hat,
  d1,
  v2hat,
  fd2,
  v3hat,
  d3,
  lddi,
  lddd,
  lambdad,
  aderivs
)
```

#### **Arguments**

waicsc	logical that indicates whether to return estimate (longer runtime)	es for the waic1 and waic2 scores
X	a vector of training data values	
v1hat	first parameter	
d1	the delta used in the numerical derivatives wit	h respect to the parameter
v2hat	second parameter	
fd2	the fractional delta used in the numerical deri eter	vatives with respect to the param-
v3hat	third parameter	
d3	the delta used in the numerical derivatives wit	h respect to the parameter
lddi	inverse observed information matrix	
lddd	third derivative of log-likelihood	
lambda	derivative of the log prior	
aderiv	logical for whether to use analytic derivatives	(instead of numerical)

#### Value

Two numeric values.

gnorm_k3_cp	Generalized Normal Distribution Predictions Based on a Calibrating Prior
	Prior

## Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qgnorm_k3_cp(
 х,
 p = seq(0.1, 0.9, 0.1),
 kbeta = 4,
 d1 = 0.01,
  fd2 = 0.01,
 means = FALSE,
 waicscores = FALSE,
 logscores = FALSE,
 dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 debug = FALSE,
  aderivs = TRUE
rgnorm_k3_cp(
 n,
 Χ,
 d1 = 0.01,
 fd2 = 0.01,
 kbeta = 4,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
dgnorm_k3_cp(
 х,
 y = x,
 d1 = 0.01,
  fd2 = 0.01,
 kbeta = 4,
  rust = FALSE,
 nrust = 1000,
 debug = FALSE,
  aderivs = TRUE
)
```

```
pgnorm_k3_cp(
 Х,
 y = x,
 d1 = 0.01,
 fd2 = 0.01,
 kbeta = 4,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
tgnorm_k3_cp(n, x, d1 = 0.01, fd2 = 0.01, kbeta = 4, debug = FALSE)
```

## Arguments

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
kbeta	the known beta parameter
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The generalized normal distribution has probability density function

$$f(x; \mu, \alpha) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|x-\mu|/\alpha)^{\beta}}$$

where x is the random variable,  $\mu, \alpha > 0$  are the parameters and we consider  $\beta$  to be known (hence the k3 in the name).

The calibrating prior is given by the right Haar prior, which is

$$\pi(\alpha) \propto \frac{1}{\alpha}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),

412 gnorm\_k3\_f1f

- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d32gnorm_k3_example_data_v1
p=c(1:9)/10
q=qgnorm_k3_cp(x,p,kbeta=4,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgnorm_k3_cp)",
main="gnorm: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

gnorm\_k3\_f1f

DMGS equation 3.3, f1 term

#### **Description**

DMGS equation 3.3, f1 term

```
gnorm_k3_f1f(y, v1, d1, v2, fd2, kbeta)
```

gnorm\_k3\_f1fa 413

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter

## Value

Matrix

# Description

The first derivative of the density

## Usage

```
gnorm_k3_f1fa(x, v1, v2, kbeta)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

kbeta the known beta parameter

#### Value

Vector

414 gnorm\_k3\_f2fa

gnorm_	k3	f2f
giioi iii_	_にっ_	_   _

DMGS equation 3.3, f2 term

## Description

DMGS equation 3.3, f2 term

## Usage

```
gnorm_k3_f2f(y, v1, d1, v2, fd2, kbeta)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

kbeta the known beta parameter

#### Value

3d array

gnorm k3	f2fa
----------	------

The second derivative of the density

## Description

The second derivative of the density

## Usage

```
gnorm_k3_f2fa(x, v1, v2, kbeta)
```

## Arguments

		C .		1 .	1
X	a vector	of fra	ınıng	data	values

v1 first parameter v2 second parameter

the known beta parameter kbeta

gnorm\_k3\_fd 415

#### Value

Matrix

Matrix

gnorm\_k3\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gnorm_k3_fd(x, v1, v2, v3)
```

## **Arguments**

a vector of training data values Χ

v1 first parameter second parameter v2 v3 third parameter

Vector

Value

 $gnorm_k3_fdd$ 

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### **Description**

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gnorm_k3_fdd(x, v1, v2, v3)
```

gnorm\_k3\_ldd

## Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

## Value

Matrix

gnorm_k3_ldd	Second derivative matrix of the normalized log-likelihood

# Description

Second derivative matrix of the normalized log-likelihood

## Usage

```
gnorm_k3_ldd(x, v1, d1, v2, fd2, kbeta)
```

# Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter

## Value

Square scalar matrix

gnorm\_k3\_ldda 417

gnorm	1/2	144	_
RHOLIII	ĸэ	Tuu	d

The second derivative of the normalized log-likelihood

#### **Description**

The second derivative of the normalized log-likelihood

#### Usage

```
gnorm_k3_ldda(x, v1, v2, kbeta)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

kbeta the known beta parameter

#### Value

Matrix

gnorm	k3	1 dc	Ы

Third derivative tensor of the normalized log-likelihood

#### **Description**

Third derivative tensor of the normalized log-likelihood

### Usage

```
gnorm_k3_lddd(x, v1, d1, v2, fd2, kbeta)
```

## Arguments

X	a vector of training data values	
v1	first parameter	

d1 the delta used in the numerical derivatives with respect to the parameter

v2 second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

kbeta the known beta parameter

#### Value

Cubic scalar array

gnorm\_k3\_lmn

		_		
gnorm	k3	-10	dd	lda

The third derivative of the normalized log-likelihood

#### **Description**

The third derivative of the normalized log-likelihood

#### Usage

```
gnorm_k3_lddda(x, v1, v2, kbeta)
```

## Arguments

x a vector of training data values

v1 first parameterv2 second parameter

kbeta the known beta parameter

#### Value

3d array

gnorm_k3_lmn	One component of the second derivative of the normalized log-
	likelihood

## Description

One component of the second derivative of the normalized log-likelihood

#### Usage

```
gnorm_k3_lmn(x, v1, d1, v2, fd2, kbeta, mm, nn)
```

## Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

gnorm\_k3\_logf 419

#### Value

Scalar value

gnorm\_k3\_logf

Logf for RUST

#### **Description**

Logf for RUST

## Usage

```
gnorm_k3_logf(params, x, kbeta)
```

#### **Arguments**

params model parameters for calculating logf x a vector of training data values kbeta the known beta parameter

#### Value

Scalar value.

gnorm\_k3\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gnorm_k3_logfdd(x, v1, v2, v3)
```

### Arguments

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

420 gnorm\_k3\_loglik

gnorm_k3_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gnorm_k3_logfddd(x, v1, v2, v3)
```

## Arguments

values
V

v1 first parameterv2 second parameterv3 third parameter

#### Value

3d array

## Description

log-likelihood function

## Usage

```
gnorm_k3_loglik(vv, x, kbeta)
```

## **Arguments**

vv parameters

x a vector of training data values kbeta the known beta parameter

### Value

Scalar value.

gnorm\_k3\_logscores 421

gnorm_k3_logscores	Log scores for MLE and RHP predictions calculated using leave-one-out

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
gnorm_k3_logscores(logscores, x, d1 = 0.01, fd2 = 0.01, kbeta, aderivs)
```

## Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
x	a vector of training data values
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

## Description

DMGS equation 3.3, mu1 term

```
gnorm_k3_mu1f(alpha, v1, d1, v2, fd2, kbeta)
```

gnorm\_k3\_mu2f

## Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter

## Value

Matrix

gnorm_k3_mu2f	DMGS equation 3.3, mu2 term	

# Description

DMGS equation 3.3, mu2 term

## Usage

```
gnorm_k3_mu2f(alpha, v1, d1, v2, fd2, kbeta)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter

## Value

3d array

gnorm\_k3\_p1f 423

gnorm_k3_p1f DMGS equation 3.3, p1 term	
-----------------------------------------	--

## Description

DMGS equation 3.3, p1 term

## Usage

```
gnorm_k3_p1f(y, v1, d1, v2, fd2, kbeta)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter

## Value

Matrix

gnorm_k3_p2f DMGS equation 3.3, p2 term
-----------------------------------------

## Description

DMGS equation 3.3, p2 term

## Usage

```
gnorm_k3_p2f(y, v1, d1, v2, fd2, kbeta)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter

424 gnorm\_lmnp

## Value

3d array

gnorm_lmnp	One component of the second derivative of the normalized log-likelihood
------------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
gnorm_lmnp(x, v1, d1, v2, fd2, kbeta, mm, nn, rr)
```

## Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

## Value

Scalar value

gnorm\_waic 425

gnorm\_waic

Waic for RUST

# Description

Waic for RUST

## Usage

```
gnorm_waic(
  waicscores,
  x,
  v1hat,
  d1,
  v2hat,
  fd2,
  kbeta,
  lddi,
  lddd,
  lambdad,
  aderivs
)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
Х	a vector of training data values
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

426 gpd\_k13\_f1fa

gpd_	k1	3	f1	f
SPU_	_ [ ]	J	_ ! !	

DMGS equation 3.3, f1 term

## Description

DMGS equation 3.3, f1 term

#### Usage

```
gpd_k13_f1f(y, v1, fd1, v2, kloc)
```

## Arguments

y a vector of values at which to calculate the density and distribution functions

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

kloc the known location parameter

#### Value

Matrix

gpd\_k13\_f1fa

The first derivative of the density

## Description

The first derivative of the density

#### Usage

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

kloc the known location parameter

#### Value

Vector

gpd\_k13\_f2f 427

gpd_	k1	3	f2f
SPU_	_r\ ı	J	_   _

DMGS equation 3.3, f2 term

## Description

DMGS equation 3.3, f2 term

#### Usage

```
gpd_k13_f2f(y, v1, fd1, v2, kloc)
```

## Arguments

y a vector of values at which to calculate the density and distribution functions

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

kloc the known location parameter

#### Value

3d array

gpd\_k13\_f2fa

The second derivative of the density

## Description

The second derivative of the density

#### Usage

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

kloc the known location parameter

#### Value

Matrix

428 gpd\_k13\_fdd

gpd_k13_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	riv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gpd_k13_fd(x, v1, v2, v3)
```

## **Arguments**

Χ	a vector of training data values
v1	first parameter

v2 second parameter v3 third parameter

## Value

Vector

gpd_k13_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gpd_k13_fdd(x, v1, v2, v3)
```

## Arguments

,	
X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Matrix

gpd\_k13\_111 429

gpd_k13_l11 One component of the second derivative of the normalized log- likelihood
-----------------------------------------------------------------------------------------

## Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
gpd_k13_l11(x, v1, fd1, v2, kloc)
```

## Arguments

x	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
kloc	the known location parameter

## Value

Scalar value

gpd_k13_l111	One component of the third derivative of the normalized log-likelihood

## Description

One component of the third derivative of the normalized log-likelihood

## Usage

```
gpd_k13_l111(x, v1, fd1, v2, kloc)
```

## Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
kloc	the known location parameter

430 gpd\_k13\_ldda

#### Value

Scalar value

gpd_k13_ldd	Second derivative matrix of the normalized log-likelihood, with fixed shape
-------------	-----------------------------------------------------------------------------

## Description

Second derivative matrix of the normalized log-likelihood, with fixed shape

## Usage

```
gpd_k13_ldd(x, v1, fd1, v2, kloc)
```

## Arguments

Χ	a vector of training data values

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

kloc the known location parameter

## Value

Square scalar matrix

gpd_k13_ldda	
--------------	--

#### **Description**

The second derivative of the normalized log-likelihood

### Usage

```
gpd_k13_ldda(x, v1, v2, kloc)
```

## Arguments

x a vector of training data value	X	a vector	of training	data value
-----------------------------------	---	----------	-------------	------------

v1 first parameter v2 second parameter

kloc the known location parameter

gpd\_k13\_lddd 431

#### Value

Matrix

gpd\_k13\_lddd

Third derivative tensor of the normalized log-likelihood

#### **Description**

Third derivative tensor of the normalized log-likelihood

#### Usage

```
gpd_k13_lddd(x, v1, fd1, v2, kloc)
```

#### **Arguments**

x a vector of training data values

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

kloc the known location parameter

#### Value

Cubic scalar array

gpd\_k13\_lddda

The third derivative of the normalized log-likelihood

### **Description**

The third derivative of the normalized log-likelihood

#### Usage

```
gpd_k13_lddda(x, v1, v2, kloc)
```

### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

kloc the known location parameter

432 *gpd\_k13\_logfddd* 

#### Value

3d array

gpd_k13_logfdd	Second derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gpd_k13_logfdd(x, v1, v2, v3)
```

## Arguments

,	
Х	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Matrix

gpd_k13_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
-----------------	-----------------------------------------------------------------------------------------------------------------

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gpd_k13_logfddd(x, v1, v2, v3)
```

## Arguments

,	
Х	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

gpd\_k13\_mu1f 433

#### Value

3d array

gpd\_k13\_mu1f

DMGS equation 3.3, mul term

## Description

DMGS equation 3.3, mu1 term

#### Usage

```
gpd_k13_mu1f(alpha, v1, fd1, v2, kloc)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

kloc the known location parameter

#### Value

Matrix

gpd\_k13\_mu1fa

Minus the first derivative of the cdf, at alpha

#### **Description**

Minus the first derivative of the cdf, at alpha

#### Usage

```
gpd_k13_mu1fa(alpha, v1, v2, kloc)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

kloc the known location parameter

gpd\_k13\_mu2fa

#### Value

Vector

gpd\_k13\_mu2f

DMGS equation 3.3, mu2 term

## Description

DMGS equation 3.3, mu2 term

#### Usage

```
gpd_k13_mu2f(alpha, v1, fd1, v2, kloc)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

kloc the known location parameter

#### Value

3d array

gpd\_k13\_mu2fa

Minus the second derivative of the cdf, at alpha

#### **Description**

Minus the second derivative of the cdf, at alpha

#### Usage

```
gpd_k13_mu2fa(alpha, v1, v2, kloc)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

kloc the known location parameter

gpd\_k13\_p1f 435

#### Value

Matrix

gpd\_k13\_p1f

DMGS equation 3.3, p1 term

## Description

DMGS equation 3.3, p1 term

#### Usage

```
gpd_k13_p1f(y, v1, fd1, v2, kloc)
```

#### **Arguments**

y a vector of values at which to calculate the density and distribution functions

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

kloc the known location parameter

#### Value

Matrix

gpd\_k13\_p2f

DMGS equation 3.3, p2 term

#### **Description**

DMGS equation 3.3, p2 term

#### Usage

## Arguments

y a vector of values at which to calculate the density and distribution functions
-----------------------------------------------------------------------------------

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

kloc the known location parameter

436 gpd\_k13\_pdd

#### Value

3d array

gpd_k13_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gpd_k13_pd(x, v1, v2, v3)
```

## Arguments

v3

X	a vector of training data values
v1	first parameter
v2	second parameter

third parameter

#### Value

Vector

gpd_k13_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gpd_k13_pdd(x, v1, v2, v3)
```

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

gpd\_k1\_checkmle 437

#### Value

Matrix

gpd\_k1\_checkmle

Check MLE

#### **Description**

Check MLE

#### Usage

```
gpd_k1_checkmle(ml_params, kloc, minxi, maxxi)
```

#### **Arguments**

ml\_params parameters

kloc the known location parameter

minxi minimum value of shape parameter xi maxxi maximum value of shape parameter xi

#### Value

No return value (just a message to the screen).

gpd_k1_cp	Generalized Pareto Distribution with Known Location Parameter, Pre-
	dictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y

- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qgpd_k1_cp(
 Х,
 p = seq(0.1, 0.9, 0.1),
 kloc = 0,
  ics = c(0, 0),
  fd1 = 0.01,
 d2 = 0.01,
  fdalpha = 0.01,
  customprior = 0,
 minxi = -0.45,
 maxxi = 2,
 means = FALSE,
 waicscores = FALSE,
  extramodels = FALSE,
 pdf = FALSE,
  dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 debug = FALSE,
  aderivs = TRUE
)
rgpd_k1_cp(
 n,
 Х,
  kloc = 0,
  ics = c(0, 0),
  fd1 = 0.01,
  d2 = 0.01,
 minxi = -0.45,
 maxxi = 2,
  extramodels = FALSE,
  rust = FALSE,
 mlcp = TRUE,
  debug = FALSE,
  aderivs = TRUE
```

```
dgpd_k1_cp(
 Х,
 y = x,
 kloc = 0,
 ics = c(0, 0),
 fd1 = 0.01,
 d2 = 0.01,
 customprior = 0,
 minxi = -0.45,
 maxxi = 2,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
pgpd_k1_cp(
 х,
 y = x,
 kloc = 0,
 ics = c(0, 0),
 fd1 = 0.01,
 d2 = 0.01,
  customprior = 0,
 minxi = -0.45,
 \max xi = 2,
 extramodels = FALSE,
  rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
tgpd_k1_cp(
 n,
 Х,
 kloc = 0,
 ics = c(0, 0),
 fd1 = 0.01,
 d2 = 0.01,
 extramodels = FALSE,
 debug = FALSE
```

## Arguments

C	
X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
kloc	the known location parameter
ics	initial conditions for the maximum likelihood search
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter
d2	if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
customprior	a custom value for the slope of the log prior at the maxlik estimate
minxi	the minimum allowed value of the shape parameter (decrease with caution)
maxxi	the maximum allowed value of the shape parameter (increase with caution)
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
extramodels	logical that indicates whether to run additional calculations and add three additional prediction models (longer runtime)
pdf	logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer runtime)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

## **Details**

The GP distribution has exceedcance distribution function

$$S(x; \mu, \sigma, \xi) = \begin{cases} \left[1 + \xi \left(\frac{x - \mu}{\sigma}\right)\right]^{-1/\xi} & \text{if } \xi \neq 0\\ \exp\left(-\frac{x - \mu}{\sigma}\right) & \text{if } \xi = 0 \end{cases}$$

where x is the random variable and  $\mu, \sigma > 0, \xi$  are the parameters.

The calibrating prior we use is given by

$$\pi(\mu, \sigma, \xi) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

The code will stop with an error if the input data gives a maximum likelihood value for the shape parameter that lies outside the range (minxi,maxxi), since outside this range there may be numerical problems. Such values seldom occur in real observed data for maxima.

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

• ml\_params: maximum likelihood estimates for the parameters.

- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

 $gpd_kl_cp$  443

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### Optional Return Values (EVT models only)

q\*\*\*\* optionally returns the following, for EVT models only:

cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

#### Optional Return Values (some EVT models only)

q\*\*\*\* optionally returns the following, for some EVT models only:

If extramodels=TRUE:

- · flat\_quantiles: predictive quantiles calculated from Bayesian integration with a flat prior.
- rh\_ml\_quantiles: predictive quantiles calculated from Bayesian integration with the calibrating prior, and the maximmum likelihood estimate for the shape parameter.
- jp\_quantiles: predictive quantiles calculated from Bayesian integration with Jeffreys' prior.

r\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_deviates: predictive random deviates calculated using a Bayesian analysis with a flat prior.
- rh\_ml\_deviates: predictive random deviates calculated using a Bayesian analysis with the RHP-MLE prior.
- jp\_deviates: predictive random deviates calculated using a Bayesian analysis with the JP.

d\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_pdf: predictive density function from a Bayesian analysis with the flat prior.
- rh\_ml\_pdf: predictive density function from a Bayesian analysis with the RHP-MLE prior.
- jp\_pdf: predictive density function from a Bayesian analysis with the JP.

p\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_cdf: predictive distribution function from a Bayesian analysis with the flat prior.
- rh\_ml\_cdf: predictive distribution function from a Bayesian analysis with the RHP-MLE prior.
- jp\_cdf: predictive distribution function from a Bayesian analysis with the JP.

These additional predictive distributions are included for comparison with the calibrating prior model. They generally give less good reliability than the calibrating prior.

#### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2024) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

· Cauchy (cauchy),

- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

446 gpd\_k1\_f1f

#### **Examples**

```
#
# example 1
x=fitdistcp::d120gpd_k1_example_data_v1
p=c(1:9)/10
q=qgpd_k1_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgpd_k1_cp)",
main="GPD: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue",lwd=2)
cat(" ml_params=",q$ml_params,"\n")
```

gpd\_k1\_f1f

DMGS equation 3.3, f1 term

#### **Description**

DMGS equation 3.3, f1 term

## Usage

```
gpd_k1_f1f(y, v1, fd1, v2, d2, kloc)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

#### Value

Matrix

gpd\_k1\_f1fa 447

	1.1	£1	c-
gpd_	_KI.	_T I	та

The first derivative of the density

## Description

The first derivative of the density

## Usage

```
gpd_k1_f1fa(x, v1, v2, kloc)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

kloc the known location parameter

#### Value

Vector

gpd\_k1\_f2f

DMGS equation 3.3, f2 term

## Description

DMGS equation 3.3, f2 term

## Usage

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

#### Value

3d array

gpd\_k1\_fd

and	k1	_f2fa
gpu_	_ N I _	_ 1

The second derivative of the density

#### **Description**

The second derivative of the density

#### Usage

```
gpd_k1_f2fa(x, v1, v2, kloc)
```

## Arguments

X	a vector of training data values
---	----------------------------------

v1 first parameterv2 second parameter

kloc the known location parameter

#### Value

Matrix

gpc	l_k^	1_fd
-----	------	------

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### **Description**

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gpd_k1_fd(x, v1, v2, v3)
```

## Arguments

X	a v	ector	ot	training	data	values
---	-----	-------	----	----------	------	--------

v1 first parameterv2 second parameterv3 third parameter

#### Value

Vector

gpd\_k1\_fdd 449

gpd_k1_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gpd_k1_fdd(x, v1, v2, v3)
```

## Arguments

Χ	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Matrix

gpd_k1_ggd_mev	Derivative of expected information matrix, based on MEV routine gpd.infomat
----------------	-----------------------------------------------------------------------------

# Description

Derivative of expected information matrix, based on MEV routine gpd.infomat

## Usage

```
gpd_k1_ggd_mev(v1, fd1, v2, d2, kloc)
```

v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

450 gpd\_k1\_ldda

#### Value

Cubic scalar array

gpd_	k1	1dd
SPU-	_ ' \ ' -	

Second derivative matrix of the normalized log-likelihood

## Description

Second derivative matrix of the normalized log-likelihood

#### Usage

```
gpd_k1_ldd(x, v1, fd1, v2, d2, kloc)
```

## Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

#### Value

Square scalar matrix

gpd_k1_ldda	The second derivative of the normalized log-likelihood	
-------------	--------------------------------------------------------	--

## Description

The second derivative of the normalized log-likelihood

## Usage

```
gpd_k1_ldda(x, v1, v2, kloc)
```

X	a vector of training data values
v1	first parameter
v2	second parameter
kloc	the known location parameter

gpd\_k1\_lddd 451

## Value

Matrix

gpd_k1_lddd 7	Third derivative tensor of the normalized log-likelihood
---------------	----------------------------------------------------------

## Description

Third derivative tensor of the normalized log-likelihood

## Usage

```
gpd_k1_1ddd(x, v1, fd1, v2, d2, kloc)
```

## Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

#### Value

Cubic scalar array

gpd_k1_lddda	The third derivative of the normalized log-likelihood
--------------	-------------------------------------------------------

## Description

The third derivative of the normalized log-likelihood

## Usage

```
gpd_k1_lddda(x, v1, v2, kloc)
```

X	a vector of training data values		
v1	first parameter		
v2	second parameter		
kloc	the known location parameter		

452 gpd\_k1\_lmnp

## Value

3d array

gpd_k1_lmn	One component of the second derivative of the normalized log-likelihood
------------	-------------------------------------------------------------------------

## Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
gpd_k1_lmn(x, v1, fd1, v2, d2, kloc, mm, nn)
```

## Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

## Value

Scalar value

gpd_k1_lmnp	One component of the second derivative of the normalized log-likelihood
-------------	-------------------------------------------------------------------------

## Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
gpd_k1_lmnp(x, v1, fd1, v2, d2, kloc, mm, nn, rr)
```

gpd\_k1\_logf 453

# Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

## Value

Scalar value

gpd_k1_logf	Logf for RUST

# Description

Logf for RUST

## Usage

```
gpd_k1_logf(params, x, kloc)
```

## Arguments

params model parameters for calculating logf
x a vector of training data values
kloc the known location parameter

## Value

Scalar value.

454 gpd\_k1\_logfddd

gpd_k1_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gpd_k1_logfdd(x, v1, v2, v3)
```

## Arguments

x a ve	or of training	data values
--------	----------------	-------------

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

gpd_k1_logfddd	Third derivative of the log density Created by Stephen Jewson using	
	Deriv() by Andrew Clausen and Serguei Sokol	

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gpd_k1_logfddd(x, v1, v2, v3)
```

## Arguments

Χ	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

3d array

*gpd\_k1\_loglik* 455

		_	
gpd_	<b>∠</b> 1	Ing	lib
SPU_	_ I _ I _	_+05	TTV

log-likelihood function

#### **Description**

log-likelihood function

#### Usage

```
gpd_k1_loglik(vv, x, kloc)
```

#### **Arguments**

vv parameters

x a vector of training data valueskloc the known location parameter

#### Value

Scalar value.

gpd\_k1\_means

Analytical Expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

## Description

Analytical Expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

## Usage

```
gpd_k1_means(
  means,
  ml_params,
  lddi,
  lddi_k2,
  lddd,
  lddd_k2,
  lambdad_flat,
  lambdad_rh_mle,
  lambdad_rh_flat,
  lambdad_jp,
  nx,
  dim = 2,
  kloc = 0
)
```

456 gpd\_k1\_mu1f

#### **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrix

1ddi\_k2 inverse observed information matrix, fixed shape parameter

1ddd third derivative of log-likelihood

1ddd\_k2 third derivative of log-likelihood, fixed shape parameter

lambdad\_flat derivative of the log flat prior

lambdad\_rh\_mle derivative of the log CRHP-MLE prior

lambdad\_rh\_flat

derivative of the log CRHP-FLAT prior

lambdad\_jp derivative of the log JP priornx length of training datadim number of parameters

kloc the known location parameter

#### Value

Two scalars

gpd_k1_mu1f	DMGS equation 3.3, mu1 term
-------------	-----------------------------

#### **Description**

DMGS equation 3.3, mu1 term

## Usage

```
gpd_k1_mu1f(alpha, v1, fd1, v2, d2, kloc)
```

## Arguments

alpł	าล	a vector of	i val	lues of	`al	pha (	one	minus	probab	oility	)

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

d2 the delta used in the numerical derivatives with respect to the parameter

kloc the known location parameter

#### Value

Matrix

gpd\_k1\_mu1fa 457

and	L 1	_mu1fa	
gpu_	KI.	_IIIU I I a	

Minus the first derivative of the cdf, at alpha

## Description

Minus the first derivative of the cdf, at alpha

## Usage

```
gpd_k1_mu1fa(alpha, v1, v2, kloc)
```

## **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameterv2 second parameter

kloc the known location parameter

#### Value

Vector

1 m.i.2.f	
	1 mu2f

DMGS equation 3.3, mu2 term

## Description

DMGS equation 3.3, mu2 term

## Usage

```
gpd_k1_mu2f(alpha, v1, fd1, v2, d2, kloc)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

d2 the delta used in the numerical derivatives with respect to the parameter

kloc the known location parameter

#### Value

3d array

458 gpd\_k1\_p1f

and	k1	_mu2fa
SPU_	. I . I	_11114214

Minus the second derivative of the cdf, at alpha

## Description

Minus the second derivative of the cdf, at alpha

#### Usage

```
gpd_k1_mu2fa(alpha, v1, v2, kloc)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

kloc the known location parameter

## Value

Matrix

			_
bag	レ1	n1	+
guu	NΙ	υı	- 1

DMGS equation 3.3, p1 term

# Description

DMGS equation 3.3, p1 term

#### Usage

```
gpd_k1_p1f(y, v1, fd1, v2, d2, kloc)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter

#### Value

Matrix

gpd\_k1\_p2f 459

## Description

DMGS equation 3.3, p2 term

## Usage

```
gpd_k1_p2f(y, v1, fd1, v2, d2, kloc)
```

## Arguments

y a vector of values at which to calculate the density and distribution fur	Ctions
v1 first parameter	
fd1 the fractional delta used in the numerical derivatives with respect to the eter	e param-
v2 second parameter	
d2 the delta used in the numerical derivatives with respect to the parameter	er
kloc the known location parameter	

## Value

3d array

gpd_k1_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gpd_k1_pd(x, v1, v2, v3)
```

Χ	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

gpd\_k1\_setics

## Value

Vector

gpd_k1_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gpd_k1_pdd(x, v1, v2, v3)
```

## Arguments

X	a vector of training data values
---	----------------------------------

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

gpd_k1_setics	Set initial conditions	
---------------	------------------------	--

## Description

Set initial conditions

## Usage

```
gpd_k1_setics(x, ics)
```

## Arguments

x a vector of training data values

ics initial conditions for the maximum likelihood search

#### Value

Vector

gpd\_k1\_waic 461

gpd_k1_waic	Waic

# Description

Waic

# Usage

```
gpd_k1_waic(
   waicscores,
   x,
   v1hat,
   fd1,
   v2hat,
   d2,
   kloc,
   lddi,
   lddd,
   lambdad,
   aderivs
)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
x	a vector of training data values
v1hat	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kloc	the known location parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

gumbel\_cp

Gumbel Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qgumbel_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    d1 = 0.01,
    fd2 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE,
    aderivs = TRUE
)
```

```
n,
 х,
 d1 = 0.01,
  fd2 = 0.01,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
dgumbel_cp(
 х,
 y = x,
 d1 = 0.01,
 fd2 = 0.01,
  rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
pgumbel_cp(
 х,
 y = x,
 d1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
tgumbel_cp(n, x, d1 = 0.01, fd2 = 0.01, debug = FALSE)
```

a vector of training data values

# **Arguments** x

р	a vector of probabilities at which to generate predictive quantiles
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)

dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Gumbel distribution has distribution function

$$F(x; \mu, \sigma) = \exp\left(-\exp\left(-\frac{x-\mu}{\sigma}\right)\right)$$

where x is the random variable and  $\mu, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

• ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible

• cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

 ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

 ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),

- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d50gumbel_example_data_v1
p=c(1:9)/10
q=qgumbel_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgumbel_cp)",
main="Gumbel: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

gumbel\_f1f 469

gum	hel	f1f

 $DMGS\ equation\ 3.3, f1\ term$ 

### Description

DMGS equation 3.3, f1 term

## Usage

```
gumbel_f1f(y, v1, d1, v2, fd2)
```

### Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

### Value

Matrix

gumbel_f1fa	gumbe	e1	f1	fa
-------------	-------	----	----	----

The first derivative of the density

## Description

The first derivative of the density

## Usage

```
gumbel_f1fa(x, v1, v2)
```

### **Arguments**

x a vector of training dat	a values
----------------------------	----------

v1 first parameter v2 second parameter

### Value

Vector

470 gumbel\_f2fa

gum	hel	f2f

DMGS equation 3.3, f2 term

### Description

DMGS equation 3.3, f2 term

## Usage

```
gumbel_f2f(y, v1, d1, v2, fd2)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

### Value

3d array

The second derivative of the density

## Description

The second derivative of the density

### Usage

```
gumbel_f2fa(x, v1, v2)
```

### **Arguments**

v1 first parameter v2 second parameter

#### Value

gumbel\_fd 471

gumbel_fd	First derivative of the density Created by Stephen Jewson using De- riv() by Andrew Clausen and Serguei Sokol
	3

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gumbel_fd(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

<pre>gumbel_fdd</pre>	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gumbel_fdd(x, v1, v2)
```

## Arguments

S
5

v1 first parameter v2 second parameter

#### Value

gumbel\_ldda

gum	h ~ 1	ldd
gum	neı	- 1 aa

Second derivative matrix of the normalized log-likelihood

### Description

Second derivative matrix of the normalized log-likelihood

#### Usage

```
gumbel_ldd(x, v1, d1, v2, fd2)
```

### Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

eter

### Value

Square scalar matrix

grimbe]	Lpp [

The second derivative of the normalized log-likelihood

### Description

The second derivative of the normalized log-likelihood

### Usage

```
gumbel_ldda(x, v1, v2)
```

### Arguments

X	a vector of training data values

v1 first parameter v2 second parameter

#### Value

gumbel\_lddd 473

gumbel_lddd	Third derivative tensor of the normalized log-likelihood
8	

### Description

Third derivative tensor of the normalized log-likelihood

## Usage

```
gumbel_1ddd(x, v1, d1, v2, fd2)
```

## Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

#### Value

Cubic scalar array

gumbel_lddda
--------------

### Description

The third derivative of the normalized log-likelihood

## Usage

```
gumbel_lddda(x, v1, v2)
```

## Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter

### Value

3d array

474 gumbel\_lmnp

gumbel_lmn	One component of the second derivative of the normalized log-likelihood

## Description

One component of the second derivative of the normalized log-likelihood

### Usage

```
gumbel_lmn(x, v1, d1, v2, fd2, mm, nn)
```

## Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

#### Value

Scalar value

<pre>gumbel_lmnp</pre>	One component of the third derivative of the normalized log-likelihood

## Description

One component of the third derivative of the normalized log-likelihood

## Usage

```
gumbel_lmnp(x, v1, d1, v2, fd2, mm, nn, rr)
```

gumbel\_logf 475

# Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

	Logf for RUST	<pre>gumbel_logf</pre>
--	---------------	------------------------

# Description

Logf for RUST

### Usage

```
gumbel_logf(params, x)
```

## Arguments

params model parameters for calculating logf
x a vector of training data values

### Value

Scalar value.

476 gumbel\_logfddd

gumbel_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gumbel_logfdd(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Matrix

gumbel_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

### Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gumbel_logfddd(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

3d array

gumbel\_loglik 477

	<pre>gumbel_loglik</pre>	log-likelihood function	
--	--------------------------	-------------------------	--

## Description

log-likelihood function

## Usage

```
gumbel_loglik(vv, x)
```

### Arguments

vv parameters

x a vector of training data values

#### Value

Scalar value.

<pre>gumbel_logscores</pre>	Log scores for MLE and RHP predictions calculated using leave-one-
	out

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

### Usage

```
gumbel_logscores(logscores, x, d1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

## Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
X	a vector of training data values
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

gumbel\_mu1f

gumbel_means	MLE and RHP predictive means	

### Description

MLE and RHP predictive means

### Usage

```
gumbel_means(means, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2)
```

### Arguments

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

#### Value

Two scalars

gumbel_mu1f DMGS equation 3.3, mu1 term	
-----------------------------------------	--

### Description

DMGS equation 3.3, mu1 term

### Usage

```
gumbel_mu1f(alpha, v1, d1, v2, fd2)
```

#### **Arguments**

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

gumbel\_mu1fa 479

#### Value

Matrix

gumbel\_mu1fa

Minus the first derivative of the cdf, at alpha

#### **Description**

Minus the first derivative of the cdf, at alpha

#### Usage

```
gumbel_mu1fa(alpha, v1, v2)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

### Value

Vector

gumbel\_mu2f

DMGS equation 3.3, mu2 term

#### **Description**

DMGS equation 3.3, mu2 term

### Usage

```
gumbel_mu2f(alpha, v1, d1, v2, fd2)
```

#### **Arguments**

alpha	a vector of values	of alpha (one	minus probability	1)

v1 first parameter

d1 the delta used in the numerical derivatives with respect to the parameter

v2 second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

3d array

480 gumbel\_p1f

gumhe l	. mu2fa

Minus the second derivative of the cdf, at alpha

### Description

Minus the second derivative of the cdf, at alpha

#### Usage

```
gumbel_mu2fa(alpha, v1, v2)
```

### Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

#### Value

Matrix

gum	ha I	n	I +
guill	$\sigma c_1$		

DMGS equation 3.3, p1 term

## Description

DMGS equation 3.3, p1 term

#### Usage

```
gumbel_p1f(y, v1, d1, v2, fd2)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

#### Value

gumbel\_p1fa 481

	- 4	_
gumbel_	n I	tа
5 a	м.	

The first derivative of the cdf

### Description

The first derivative of the cdf

#### Usage

```
gumbel_p1fa(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter

v2 second parameter

#### Value

Vector

I 7	1	
gumbe]	l bi	CD

Gumbel Distribution with a Predictor, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qgumbel_p1_cp(
 х,
  t,
  t0 = NA,
 n0 = NA,
 p = seq(0.1, 0.9, 0.1),
 d1 = 0.01,
 d2 = 0.01,
  fd3 = 0.01,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  dmgs = TRUE,
  rust = FALSE,
  nrust = 1e+05,
 predictordata = TRUE,
  centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
rgumbel_p1_cp(
 n,
 х,
  t,
  t0 = NA,
 n0 = NA,
 d1 = 0.01,
 d2 = 0.01,
  fd3 = 0.01,
  rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
dgumbel_p1_cp(
  х,
  t,
```

```
t0 = NA,
 n0 = NA,
 y = x,
 d1 = 0.01,
 d2 = 0.01,
 fd3 = 0.01,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
pgumbel_p1_cp(
 Х,
 t,
 t0 = NA,
 n0 = NA,
 y = x,
 d1 = 0.01,
 d2 = 0.01,
 fd3 = 0.01,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
tgumbel_p1_cp(n, x, t, d1 = 0.01, d2 = 0.01, fd3 = 0.01, debug = FALSE)
```

#### **Arguments**

x	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
p	a vector of probabilities at which to generate predictive quantiles
d1	the fractional delta used in the numerical derivatives with respect to the location parameter
d2	the fractional delta used in the numerical derivatives with respect to the slope parameter
fd3	the fractional delta used in the numerical derivatives with respect to the scale parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)

waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- $\bullet$  ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Gumbel distribution with a predictor has distribution function

$$F(x; a, b, \sigma) = \exp\left(-\exp\left(-\frac{x - \mu(a, b)}{\sigma}\right)\right)$$

where x is the random variable,  $\mu = a + bt$  is the shape parameter as a function of parameters a, b and predictor t, and  $\sigma > 0$  is the scale parameter.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

• ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)

 cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### Examples

```
#
# example 1
x=fitdistcp::d70gumbel_p1_example_data_v1_x
tt=fitdistcp::d70gumbel_p1_example_data_v1_t
p=c(1:9)/10
n0=10
```

gumbel\_p1\_f1f 489

```
q=qgumbel_p1_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgumbel_p1_cp)",
main="Gumbel w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

gumbel\_p1\_f1f

DMGS equation 2.1, f1 term

### Description

DMGS equation 2.1, f1 term

#### Usage

```
gumbel_p1_f1f(y, t0, v1, d1, v2, d2, v3, fd3)
```

### Arguments

у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

490 gumbel\_p1\_f2f

gumbel	p1	f1	fa
Sumber	_0.	_ ' '	ı u

The first derivative of the density

#### **Description**

The first derivative of the density

### Usage

```
gumbel_p1_f1fa(x, t, v1, v2, v3)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

gumbel	n1	f2f

DMGS equation 2.1, f2 term

## Description

DMGS equation 2.1, f2 term

### Usage

```
gumbel_p1_f2f(y, t0, v1, d1, v2, d2, v3, fd3)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the param-
	eter

gumbel\_p1\_f2fa 491

### Value

3d array

gumbel\_p1\_f2fa

The second derivative of the density

### Description

The second derivative of the density

#### Usage

```
gumbel_p1_f2fa(x, t, v1, v2, v3)
```

#### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Matrix

gumbel_p1_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

### Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gumbel_p1_fd(x, t, v1, v2, v3)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

492 gumbel\_p1\_ldd

### Value

Vector

gumbel_p1_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
guinet_b1_1 dd	

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gumbel_p1_fdd(x, t, v1, v2, v3)
```

### Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

third parameter

### Value

Matrix

v3

gumbel_p1_ldd	Second derivative matrix of the normalized log-likelihood

## Description

Second derivative matrix of the normalized log-likelihood

## Usage

```
gumbel_p1_ldd(x, t, v1, d1, v2, d2, v3, fd3)
```

gumbel\_p1\_ldda 493

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

## Value

Square scalar matrix

gumbel_p1_ldda	
----------------	--

# Description

The second derivative of the normalized log-likelihood

# Usage

```
gumbel_p1_ldda(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

494 gumbel\_p1\_lddda

gumbel_	n 1	1 444
guilibet_	_D I _	_tuuu

Third derivative tensor of the normalized log-likelihood

#### **Description**

Third derivative tensor of the normalized log-likelihood

#### Usage

```
gumbel_p1_lddd(x, t, v1, d1, v2, d2, v3, fd3)
```

### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Cubic scalar array

<pre>gumbel_p1_lddda</pre>	The third derivative of the normalized log-likelihood
guiliber_pr_ruuua	The inita derivative of the hormalized log-likelihood

### Description

The third derivative of the normalized log-likelihood

## Usage

```
gumbel_p1_lddda(x, t, v1, v2, v3)
```

### **Arguments**

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

gumbel\_p1\_lmn 495

### Value

3d array

likelihood	gumbel_p1_lmn	One component of the second derivative of the normalized log-likelihood
------------	---------------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
gumbel_p1_lmn(x, t, v1, d1, v2, d2, v3, fd3, mm, nn)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

### Value

Scalar value

496 gumbel\_p1\_logf

gumbel_p1_lmnp	One component of the second derivative of the normalized log-likelihood
----------------	-------------------------------------------------------------------------

## Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
gumbel_p1_lmnp(x, t, v1, d1, v2, d2, v3, fd3, mm, nn, rr)
```

## Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

#### Value

Scalar value

	Logf for RUST	<pre>gumbel_p1_logf</pre>
--	---------------	---------------------------

## Description

Logf for RUST

## Usage

```
gumbel_p1_logf(params, x, t)
```

gumbel\_p1\_logfdd 497

## Arguments

params	model parameters for calculating logf
x	a vector of training data values

t a vector or matrix of predictors

#### Value

Scalar value.

gumbel_p1_logfdd	Second derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gumbel_p1_logfdd(x, t, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

498 gumbel\_p1\_loglik

<pre>gumbel_p1_logfddd</pre>	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gumbel_p1_logfddd(x, t, v1, v2, v3)
```

### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

#### Value

3d array

gumbel\_p1\_loglik ol

observed log-likelihood function

### Description

observed log-likelihood function

### Usage

```
gumbel_p1_loglik(vv, x, t)
```

### Arguments

VV	parameters

x a vector of training data valuest a vector or matrix of predictors

#### Value

Scalar value.

gumbel\_p1\_logscores 499

gumbel_p1_logscores	Log scores for MLE and RHP predictions calculated using leave-one- out
	out

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

### Usage

```
gumbel_p1_logscores(logscores, x, t, d1, d2, fd3, aderivs = TRUE)
```

## Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
x	a vector of training data values
t	a vector or matrix of predictors
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

|--|

## Description

Gumbel distribution: RHP mean

## Usage

```
gumbel_p1_means(means, t0, ml_params, lddi, lddd, lambdad_rhp, nx, dim)
```

500 gumbel\_p1\_mu1f

#### **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

t0 a single value of the predictor (specify either t0 or n0 but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

#### Value

Two scalars

gumbel\_p1\_mu1f DMGS equation 3.3, mu1 term

### Description

DMGS equation 3.3, mu1 term

### Usage

```
gumbel_p1_mu1f(alpha, t0, v1, d1, v2, d2, v3, fd3)
```

#### **Arguments**

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

gumbel\_p1\_mu1fa 501

gumbel	n1	mulfa	
guillber	υı	IIIUIIa	

Minus the first derivative of the cdf, at alpha

#### **Description**

Minus the first derivative of the cdf, at alpha

### Usage

```
gumbel_p1_mu1fa(alpha, t, v1, v2, v3)
```

### **Arguments**

alpha	a vector of values of alpha (one minus probability)
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

gumbel	1	2 £
gumbei	DТ	MU1/T

DMGS equation 3.3, mu2 term

## Description

DMGS equation 3.3, mu2 term

### Usage

```
gumbel_p1_mu2f(alpha, t0, v1, d1, v2, d2, v3, fd3)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

502 gumbel\_p1\_p1f

#### Value

3d array

 ${\tt gumbel\_p1\_mu2fa}$ 

Minus the second derivative of the cdf, at alpha

### Description

Minus the second derivative of the cdf, at alpha

## Usage

```
gumbel_p1_mu2fa(alpha, t, v1, v2, v3)
```

### Arguments

alpha a vector of values of alpha (one minus probability)

t a vector or matrix of predictors

v1 first parameter v2 second parameter

v3 third parameter

#### Value

Matrix

gumbel\_p1\_p1f

DMGS equation 2.1, p1 term

### Description

DMGS equation 2.1, p1 term

### Usage

```
gumbel_p1_p1f(y, t0, v1, d1, v2, d2, v3, fd3)
```

gumbel\_p1\_p1fa 503

## Arguments

7	/	a vector of values at which to calculate the density and distribution functions
1	:0	a single value of the predictor (specify either t0 or n0 but not both)
١	/1	first parameter
(	d1	the delta used in the numerical derivatives with respect to the parameter
١	/2	second parameter
(	12	the delta used in the numerical derivatives with respect to the parameter
١	/3	third parameter
İ	fd3	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Matrix

# Description

The first derivative of the cdf

# Usage

```
gumbel_p1_p1fa(x, t, v1, v2, v3)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Vector

504 gumbel\_p1\_p2fa

gumbel_	n1	n2f
guilloci.	_P'	_

DMGS equation 2.1, p2 term

## Description

DMGS equation 2.1, p2 term

### Usage

```
gumbel_p1_p2f(y, t0, v1, d1, v2, d2, v3, fd3)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

3d array

gumbel	ք1	p2fa

The second derivative of the cdf

### Description

The second derivative of the cdf

## Usage

```
gumbel_p1_p2fa(x, t, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

gumbel\_p1\_pd 505

## Value

Matrix

gumbel_p1_pd First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol	у
----------------------------------------------------------------------------------------------------------------------	---

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gumbel_p1_pd(x, t, v1, v2, v3)
```

## **Arguments**

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Vector

gumbel_p1_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gumbel_p1_pdd(x, t, v1, v2, v3)
```

## Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

third parameter

### Value

Matrix

v3

```
{\tt gumbel\_p1\_predictordata}
```

Predicted Parameter and Generalized Residuals

## Description

Predicted Parameter and Generalized Residuals

## Usage

```
gumbel_p1_predictordata(predictordata, x, t, t0, params)
```

# Arguments

predictordata	logical that indicates whether to calculate and return predictordata
x	a vector of training data values
t	a vector or matrix of predictors
t0	a single value of the predictor (specify either t0 or n0 but not both)
params	model parameters for calculating logf

### Value

Two vectors

gumbel\_p1\_waic 507

gumbel\_p1\_waic

Waic

# Description

Waic

# Usage

```
gumbel_p1_waic(
  waicscores,
  x,
  t,
  v1hat,
  d1,
  v2hat,
  d2,
  v3hat,
  fd3,
  lddi,
  lddd,
  lambdad,
  aderivs
)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
t	a vector or matrix of predictors
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3hat	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

508 gumbel\_p2fa

## Value

Two numeric values.

 $gumbel_p2f$ 

DMGS equation 3.3, p2 term

## Description

DMGS equation 3.3, p2 term

## Usage

```
gumbel_p2f(y, v1, d1, v2, fd2)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

# Value

3d array

gumbel\_p2fa

The second derivative of the cdf

# Description

The second derivative of the cdf

## Usage

```
gumbel_p2fa(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

gumbel\_pd 509

Andrew Clausen and Serguei Sokol	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gumbel_pd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Vector

gumbel_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gumbel_pdd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

## Description

Waic

### Usage

```
gumbel_waic(waicscores, x, v1hat, d1, v2hat, fd2, lddi, lddd, lambdad, aderivs)
```

### Arguments

waicsco	res logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior

logical for whether to use analytic derivatives (instead of numerical)

#### Value

aderivs

Two numeric values.

halfnorm_cp	Half-Normal Distribution Predictions Based on a Calibrating Prior

## Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.

- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qhalfnorm_cp(
 p = seq(0.1, 0.9, 0.1),
  fd1 = 0.01,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 debug = FALSE,
  aderivs = TRUE
)
rhalfnorm_cp(
 n,
 х,
  fd1 = 0.01,
  rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
dhalfnorm_cp(
  Х,
  y = x,
  fd1 = 0.01,
  rust = FALSE,
  nrust = 1000,
  debug = FALSE,
```

```
aderivs = TRUE
)

phalfnorm_cp(
    x,
    y = x,
    fd1 = 0.01,
    rust = FALSE,
    nrust = 1000,
    debug = FALSE,
    aderivs = TRUE
)

thalfnorm_cp(n, x, fd1 = 0.01, debug = FALSE)
```

## **Arguments**

X	a vector of training data values
p	a vector of probabilities at which to generate predictive quantiles
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

## Value

q\*\*\*\* returns a list containing at least the following:

• ml\_params: maximum likelihood estimates for the parameters.

- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The half-normal distribution has probability density function

$$f(x;\theta) = \frac{2\theta}{\pi} e^{-\theta^2 x^2/\pi}$$

where  $x \ge 0$  is the random variable and  $\theta > 0$  is the parameter.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\theta) \propto \frac{1}{\theta}$$

as given in Jewson et al. (2025). Some other authors may parametrize the half-normal differently.

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),

halfnorm\_f1f 517

- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
#
# example 1
x=fitdistcp::d20halfnorm_example_data_v1
p=c(1:9)/10
q=qhalfnorm_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles)
xmax=max(q$ml_quantiles,q$cp_quantiles)
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qhalfnorm_cp)",
main="Halfnorm: quantile estimates")
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

halfnorm\_f1f

DMGS equation 2.1, f1 term

# Description

```
DMGS equation 2.1, f1 term
```

### Usage

```
halfnorm_f1f(y, v1, fd1)
```

### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the param-
	eter

518 halfnorm\_f2f

## Value

Matrix

halfnorm\_f1fa

The first derivative of the density

## Description

The first derivative of the density

### Usage

```
halfnorm_f1fa(x, v1)
```

### **Arguments**

x a vector of training data values

v1 first parameter

### Value

Vector

halfnorm\_f2f

DMGS equation 2.1, f2 term

# Description

DMGS equation 2.1, f2 term

## Usage

```
halfnorm_f2f(y, v1, fd1)
```

# Arguments

y a vector of values at which to calculate the density and distribution functions

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

## Value

3d array

halfnorm\_f2fa 519

halfnorm_	f2fa
-----------	------

The second derivative of the density

## Description

The second derivative of the density

#### Usage

```
halfnorm_f2fa(x, v1)
```

### Arguments

x a vector of training data values

v1 first parameter

### Value

Matrix

halfnorm\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
halfnorm_fd(x, v1)
```

# Arguments

x a vector of training data values

v1 first parameter

### Value

Vector

520 halfnorm\_gg

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
halfnorm_fdd(x, v1)
```

### Arguments

x a vector of training data values

v1 first parameter

### Value

Matrix

ha	1 fr	nor	m	gg

Expected information matrix

# Description

Expected information matrix

## Usage

```
halfnorm_gg(v1, fd1)
```

### **Arguments**

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

## Value

Square scalar matrix

halfnorm\_gg11 521

	_	
hal	fnorm	gg]]

Second derivative of the expected log-likelihood

## Description

Second derivative of the expected log-likelihood

### Usage

```
halfnorm_gg11(alpha, v1, fd1)
```

## Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Scalar value

halfnorm\_l111

Third derivative of the normalized log-likelihood

## Description

Third derivative of the normalized log-likelihood

### Usage

```
halfnorm_l111(x, v1, fd1)
```

### **Arguments**

x a vector of training data values

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

### Value

Scalar value

522 halfnorm\_ldda

halfnorm_lo	dd
-------------	----

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

### Usage

```
halfnorm_ldd(x, v1, fd1)
```

#### **Arguments**

x a vector of training data values

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

## Value

Square scalar matrix

halfnorm\_ldda

The second derivative of the normalized log-likelihood

### **Description**

The second derivative of the normalized log-likelihood

## Usage

```
halfnorm_ldda(x, v1)
```

### **Arguments**

x a vector of training data values

v1 first parameter

### Value

halfnorm\_lddd 523

	C	-	
naı	fnorm	10	เกตเ

Third derivative tensor of the log-likelihood

# Description

Third derivative tensor of the log-likelihood

## Usage

```
halfnorm_lddd(x, v1, fd1)
```

## Arguments

x a vector of training data values

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

## Value

Cubic scalar array

halfnorm\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

### Usage

```
halfnorm_lddda(x, v1)
```

### **Arguments**

x a vector of training data values

v1 first parameter

### Value

3d array

524 halfnorm\_logfdd

halfnorm\_logf

Logf for RUST

## Description

Logf for RUST

#### Usage

```
halfnorm_logf(params, x)
```

## Arguments

params model parameters for calculating logf

x a vector of training data values

### Value

Scalar value.

halfnorm\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
halfnorm_logfdd(x, v1)
```

# Arguments

x a vector of training data values

v1 first parameter

### Value

halfnorm\_logfddd 525

halfnorm_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
halfnorm_logfddd(x, v1)
```

## Arguments

x a vector of training data values

v1 first parameter

## Value

3d array

 $halfnorm\_loglik$ 

Log-likelihood function

# Description

Log-likelihood function

## Usage

```
halfnorm_loglik(vv, x)
```

# Arguments

vv parameters

x a vector of training data values

## Value

Scalar value.

526 halfnorm\_means

halfnorm_logscores	Log scores for MLE and RHP predictions calculated using leave-one-
	out

#### Description

Log scores for MLE and RHP predictions calculated using leave-one-out

### Usage

```
halfnorm_logscores(logscores, x, fd1 = 0.01, aderivs = TRUE)
```

### **Arguments**

logiscores logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

the fractional delta used in the numerical derivatives with respect to the param-

eter

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

halfnorm\_means MLE and RHP predictive means RHP mean based on the expectation

of DMGS equation 2.1

### **Description**

MLE and RHP predictive means RHP mean based on the expectation of DMGS equation 2.1

# Usage

```
halfnorm_means(means, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 1)
```

#### **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

1ddiinverse observed information matrix1dddthird derivative of log-likelihood1ambdad\_rhpderivative of the log RHP prior

nx length of training data dim number of parameters halfnorm\_mu1f 527

### Value

Two scalars

halfnorm\_mu1f

DMGS equation 3.3, mul term

## Description

DMGS equation 3.3, mu1 term

### Usage

```
halfnorm_mu1f(alpha, v1, fd1)
```

# Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Matrix

halfnorm\_mu2f

DMGS equation 3.3, mu2 term

## Description

DMGS equation 3.3, mu2 term

# Usage

```
halfnorm_mu2f(alpha, v1, fd1)
```

### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

### Value

3d array

528 halfnorm\_p2f

halfnorm\_p1f

DMGS equation 2.1, p1 term

### **Description**

DMGS equation 2.1, p1 term

### Usage

```
halfnorm_p1f(y, v1, fd1)
```

## Arguments

y a vector of values at which to calculate the density and distribution functions

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

### Value

Matrix

halfnorm\_p2f

DMGS equation 2.1, p2 term

### **Description**

DMGS equation 2.1, p2 term

### Usage

```
halfnorm_p2f(y, v1, fd1)
```

### **Arguments**

y a vector of values at which to calculate the density and distribution functions

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

3d array

halfnorm\_waic 529

### Description

Waic

### Usage

```
halfnorm_waic(waicscores, x, v1hat, fd1, lddi, lddd, lambdad, aderivs)
```

### **Arguments**

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
v1hat	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

invgamma_cp	Inverse Gamma L Prior	Distribution, Prediction	s Based on a Calibrating

# Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.

- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qinvgamma_cp(
  Х,
 p = seq(0.1, 0.9, 0.1),
 fd1 = 0.01,
  fd2 = 0.01,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 prior = "type 1",
 debug = FALSE,
  aderivs = TRUE
)
rinvgamma_cp(
 n,
  fd1 = 0.01,
  fd2 = 0.01,
  rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
dinvgamma_cp(
 х,
 y = x,
  fd1 = 0.01,
  fd2 = 0.01,
  rust = FALSE,
 nrust = 1000,
```

```
debug = FALSE,
   aderivs = TRUE
)

pinvgamma_cp(
    x,
    y = x,
    fd1 = 0.01,
    fd2 = 0.01,
    rust = FALSE,
    nrust = 1000,
    debug = FALSE,
    aderivs = TRUE
)

tinvgamma_cp(n, x, fd1 = 0.01, fd2 = 0.01, debug = FALSE)
```

### **Arguments**

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
prior	logical indicating which prior to use
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Inverse Gamma distribution has probability density function

$$f(x; \alpha, \sigma) = \frac{1}{x\Gamma(\alpha)} \left(\frac{\sigma}{x}\right)^{\alpha} e^{-\sigma/x}$$

where  $x \ge 0$  is the random variable and  $\alpha > 0, \sigma > 0$  are the parameters.

The calibrating prior we use is

$$\pi(\alpha, \sigma) \propto \frac{1}{\alpha \sigma}$$

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),

536 invgamma\_f1f

- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
#
# example 1
x=fitdistcp::d101invgamma_example_data_v1
p=c(1:9)/10
q=qinvgamma_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qinvgamma_cp)",
main="Invgamma: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

invgamma\_f1f

DMGS equation 3.3, f1 term

### **Description**

DMGS equation 3.3, f1 term

#### Usage

```
invgamma_f1f(y, v1, fd1, v2, fd2)
```

invgamma\_f1fa 537

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

## Value

Matrix

# Description

The first derivative of the density

# Usage

```
invgamma_f1fa(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

### Value

Vector

538 invgamma\_f2fa

i	nve	ram	ma	_f2f

DMGS equation 3.3, f2 term

## Description

DMGS equation 3.3, f2 term

## Usage

```
invgamma_f2f(y, v1, fd1, v2, fd2)
```

## Arguments

у	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

#### eter

## Value

3d array

i	nvgamma.	f2fa
	IIV gallilla	_ 1

The second derivative of the density

# Description

The second derivative of the density

## Usage

```
invgamma_f2fa(x, v1, v2)
```

## Arguments

X	a vector of	r training data values	

v1 first parameter v2 second parameter

### Value

invgamma\_fd 539

invgamma_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	0

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
invgamma_fd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Vector

invgamma_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
invgamma_fdd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

540 invgamma\_ldda

101/	gamma	1 서서
TIIV	gaiiiiiia	Tuu

Second derivative matrix of the normalized log-likelihood

#### **Description**

Second derivative matrix of the normalized log-likelihood

#### Usage

```
invgamma_ldd(x, v1, fd1, v2, fd2)
```

## Arguments

X	a vector of training data values
---	----------------------------------

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

### Value

Square scalar matrix

invgamma\_ldda

The second derivative of the normalized log-likelihood

## Description

The second derivative of the normalized log-likelihood

### Usage

```
invgamma_ldda(x, v1, v2)
```

### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

invgamma\_lddd 541

invgamma_	Iddd
TIIVEAIIIIIA	Tuuu

Third derivative tensor of the normalized log-likelihood

#### **Description**

Third derivative tensor of the normalized log-likelihood

#### Usage

```
invgamma_lddd(x, v1, fd1, v2, fd2)
```

#### **Arguments**

Х	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
^	

v2 second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Cubic scalar array

i	nvgamma	Lddda

The third derivative of the normalized log-likelihood

## Description

The third derivative of the normalized log-likelihood

#### Usage

```
invgamma_lddda(x, v1, v2)
```

# Arguments

X	a vector of	f training	data val	ues
---	-------------	------------	----------	-----

v1 first parameter v2 second parameter

#### Value

3d array

542 invgamma\_lmnp

invgamma_lmn	component ihood	of	the	second	derivative	of	the	normalized	log-	

# Description

One component of the second derivative of the normalized log-likelihood

### Usage

```
invgamma_lmn(x, v1, fd1, v2, fd2, mm, nn)
```

# Arguments

х	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

# Value

Scalar value

invgamma_lmnp One component of the second derivative of the normalize likelihood	d log-
----------------------------------------------------------------------------------	--------

# Description

One component of the second derivative of the normalized log-likelihood

```
invgamma_lmnp(x, v1, fd1, v2, fd2, mm, nn, rr)
```

invgamma\_logf 543

### Arguments

x	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

$_{ m logf}$ $_{ m LogfforRUST}$
----------------------------------

# Description

Logf for RUST

# Usage

```
invgamma_logf(params, x)
```

# Arguments

params model parameters for calculating logf x a vector of training data values

# Value

Scalar value.

544 invgamma\_logfddd

invgamma_logfdd Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol		
----------------------------------------------------------------------------------------------------------------------------------	--	--

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
invgamma_logfdd(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Matrix

invgamma_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

### Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
invgamma_logfddd(x, v1, v2)
```

### **Arguments**

X	a vector of training	data values
X	a vector of training	data varue

v1 first parameter v2 second parameter

#### Value

3d array

invgamma\_loglik 545

#### **Description**

log-likelihood function

### Usage

```
invgamma_loglik(vv, x)
```

### Arguments

vv parameters

x a vector of training data values

#### Value

Scalar value.

invgamma\_logscores

Log scores for MLE and cp predictions calculated using leave-one-out

### Description

Log scores for MLE and cp predictions calculated using leave-one-out

#### Usage

```
invgamma_logscores(logscores, x, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

### Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
X	a vector of training data values
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

eter

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

546 invgamma\_mu2f

invgamma_mu1f	DMGS equation 3.3, mu1 term	

# Description

DMGS equation 3.3, mu1 term

# Usage

```
invgamma_mu1f(alpha, v1, fd1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Matrix

# Description

DMGS equation 3.3, mu2 term

# Usage

```
invgamma_mu2f(alpha, v1, fd1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

invgamma\_p1f 547

### Value

3d array

invgamma\_p1f

DMGS equation 3.3, p1 term

# Description

DMGS equation 3.3, p1 term

# Usage

```
invgamma_p1f(y, v1, fd1, v2, fd2)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

Matrix

invgamma\_p2f

DMGS equation 3.3, p2 term

# Description

DMGS equation 3.3, p2 term

```
invgamma_p2f(y, v1, fd1, v2, fd2)
```

548 invgamma\_waic

#### **Arguments**

a vector of values at which to calculate the density and distribution functions у v1 first parameter the fractional delta used in the numerical derivatives with respect to the paramfd1 v2 second parameter fd2 the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

3d array

invgamma\_waic

Waic

# **Description**

Waic

#### Usage

```
invgamma_waic(
 waicscores,
 х,
 v1hat,
  fd1,
  v2hat,
  fd2,
  lddi,
  lddd,
 lambdad,
  aderivs
)
```

#### **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

a vector of training data values Х

v1hat first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

v2hat second parameter

fd2	the fractional delta used in the numerical derivatives with respect to the parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

invgauss_cp	Inverse Gauss Distribution, Predictions Based on a Calibrating Prior

#### Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qinvgauss_cp(
  х,
  p = seq(0.1, 0.9, 0.1),
  fd1 = 0.01,
  fd2 = 0.01,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  dmgs = TRUE,
  rust = FALSE,
  nrust = 1e+05,
  prior = "type 1",
 debug = FALSE,
  aderivs = TRUE
)
rinvgauss_cp(
  n,
 Х,
  fd1 = 0.01,
  fd2 = 0.01,
  rust = FALSE,
  prior = "type 1",
 mlcp = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
dinvgauss_cp(
  х,
 y = x,
 fd1 = 0.01,
  fd2 = 0.01,
  rust = FALSE,
 nrust = 1000,
 prior = "type 1",
 debug = FALSE,
  aderivs = TRUE
)
pinvgauss_cp(
 х,
  y = x,
  fd1 = 0.01,
  fd2 = 0.01,
  rust = FALSE,
  nrust = 1000,
```

```
prior = "type 1",
  debug = FALSE,
  aderivs = TRUE
)

tinvgauss_cp(n, x, fd1 = 0.01, fd2 = 0.01, prior = "type 1", debug = FALSE)
```

### Arguments

x	a vector of training data values
p	a vector of probabilities at which to generate predictive quantiles
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
prior	logical indicating which prior to use
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.

- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Inverse Gaussian distribution has probability density function

$$f(x; \mu, \phi) = \left(\frac{1}{2\pi\phi x^3}\right)^{1/2} \exp\left(-\frac{(x-\mu)^2}{2\mu^2\phi x}\right)$$

where  $x \ge 0$  is the random variable and  $\mu > 0, \phi > 0$  are the parameters.

The calibrating prior we use by default is

$$\pi(\alpha, \sigma) \propto \frac{1}{\phi}$$

The prior

$$\pi(\alpha, \sigma) \propto \frac{1}{\mu \phi}$$

is also available as an option with prior="type 2".

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

• Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),

556 invgauss\_f1f

• Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
debug=FALSE
# example 1 can go wrong for small sample sizes, so I've increased to 50
#
# example 1
if(debug)cat("example 1\n")
x=fitdistcp::d102invgauss_example_data_v1
if(debug)cat("x=",x,"\n")
p=c(1:9)/10
q=qinvgauss_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qinvgauss_cp)",
main="Invgauss: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

invgauss\_f1f

DMGS equation 3.3, f1 term

# Description

DMGS equation 3.3, f1 term

#### Usage

```
invgauss_f1f(y, v1, fd1, v2, fd2)
```

#### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

invgauss\_f1fa 557

### Value

Matrix

invgauss\_f1fa

The first derivative of the density

### Description

The first derivative of the density

### Usage

```
invgauss_f1fa(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameterv2 second parameter

#### Value

Vector

invgauss\_f2f

DMGS equation 3.3, f2 term

## Description

DMGS equation 3.3, f2 term

### Usage

```
invgauss_f2f(y, v1, fd1, v2, fd2)
```

# Arguments

у	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

558 invgauss\_fd

#### Value

3d array

invgauss\_f2fa

The second derivative of the density

### Description

The second derivative of the density

#### Usage

```
invgauss_f2fa(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

invgauss\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
invgauss_fd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

invgauss\_fdd 559

invgauss_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
--------------	--------------------------------------------------------------------------------------------------------------

### Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
invgauss_fdd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

invgauss_ldd Second derivative matrix of the normalized log-likelihood	invgauss_ldd	Second derivative matrix of the normalized log-likelihood	
------------------------------------------------------------------------	--------------	-----------------------------------------------------------	--

### Description

Second derivative matrix of the normalized log-likelihood

### Usage

```
invgauss_ldd(x, v1, fd1, v2, fd2)
```

# Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Square scalar matrix

560 invgauss\_lddd

i	nvg	าลเม	SS	1	dd	a
_	1 I V 🗲	,uu	22		uu	u

The second derivative of the normalized log-likelihood

#### **Description**

The second derivative of the normalized log-likelihood

#### Usage

```
invgauss_ldda(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

_	_	
invgauss	٦,	146
1110890122		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

Third derivative tensor of the normalized log-likelihood

### Description

Third derivative tensor of the normalized log-likelihood

### Usage

```
invgauss_lddd(x, v1, fd1, v2, fd2)
```

#### **Arguments**

X	a vector	of training	data values
٨	a vector	or nanning	uata values

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Cubic scalar array

invgauss\_lddda 561

invgauss	1 ddda
TIIVEAUSS	_Tuuua

The third derivative of the normalized log-likelihood

### Description

The third derivative of the normalized log-likelihood

# Usage

```
invgauss_lddda(x, v1, v2)
```

### Arguments

Y	a vector of	ftraining	data values
^	a vector or	uanning	uata varues

v1 first parameter v2 second parameter

#### Value

3d array

invgauss_lmn	One component of the second derivative of the normalized log-
	likelihood

### Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
invgauss_lmn(x, v1, fd1, v2, fd2, mm, nn)
```

### Arguments

Х	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

562 invgauss\_logf

### Value

Scalar value

|--|

# Description

One component of the second derivative of the normalized log-likelihood

### Usage

```
invgauss_lmnp(x, v1, fd1, v2, fd2, mm, nn, rr)
```

# Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

invgauss_logf	Logf for RUST	

# Description

Logf for RUST

```
invgauss_logf(params, x, prior)
```

invgauss\_logfdd 563

#### **Arguments**

params model parameters for calculating logf

x a vector of training data values

prior logical indicating which prior to use

#### Value

Scalar value.

invgauss\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
invgauss_logfdd(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

#### Value

Matrix

invgauss\_logfddd

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### **Description**

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
invgauss_logfddd(x, v1, v2)
```

564 invgauss\_logscores

#### **Arguments**

x a vector of training data values

v1 first parameter

v2 second parameter

#### Value

3d array

invgauss\_loglik

log-likelihood function

# Description

log-likelihood function

### Usage

```
invgauss_loglik(vv, x)
```

#### **Arguments**

vv parameters

x a vector of training data values

#### Value

Scalar value.

invgauss\_logscores

Log scores for MLE and RHP predictions calculated using leave-one-out

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

```
invgauss_logscores(logscores, x, prior, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

invgauss\_means 565

#### **Arguments**

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

prior logical indicating which prior to use

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

the fractional delta used in the numerical derivatives with respect to the param-

eter

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

invgauss\_means *MLH* 

MLE and RHP predictive means

### **Description**

MLE and RHP predictive means

#### Usage

```
invgauss_means(means, ml_params, lddi, lddd, lambdad_cp, nx, dim = 2)
```

#### Arguments

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihood

lambdad\_cp derivative of the log prior
nx length of training data
dim number of parameters

#### Value

Two scalars

566 invgauss\_mu2f

invgauss_mu1f	DMGS equation 3.3, mu1 term	
---------------	-----------------------------	--

# Description

DMGS equation 3.3, mu1 term

# Usage

```
invgauss_mu1f(alpha, v1, fd1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Matrix

invgauss_mu2f	DMGS equation 3.3, mu2 term	

# Description

DMGS equation 3.3, mu2 term

# Usage

```
invgauss_mu2f(alpha, v1, fd1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

invgauss\_p1f 567

### Value

3d array

invgauss\_p1f

DMGS equation 3.3, p1 term

# Description

DMGS equation 3.3, p1 term

# Usage

```
invgauss_p1f(y, v1, fd1, v2, fd2)
```

# Arguments

у	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

Matrix

invgauss\_p2f

DMGS equation 3.3, p2 term

# Description

DMGS equation 3.3, p2 term

```
invgauss_p2f(y, v1, fd1, v2, fd2)
```

568 invgauss\_waic

#### **Arguments**

a vector of values at which to calculate the density and distribution functions У v1 first parameter the fractional delta used in the numerical derivatives with respect to the paramfd1 v2 second parameter fd2 the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

3d array

invgauss\_waic

Waic

# **Description**

Waic

#### Usage

```
invgauss_waic(
 waicscores,
 х,
 v1hat,
  fd1,
  v2hat,
  fd2,
  lddi,
  lddd,
 lambdad,
  aderivs
)
```

#### **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

a vector of training data values Х

v1hat first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

v2hat second parameter *jpf2p* 569

the fractional delta used in the numerical derivatives with respect to the param-

eter

lddi inverse observed information matrixlddd third derivative of log-likelihood

lambdad derivative of the log prior

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

jpf2p Jeffreys' Prior with two parameters

# Description

Jeffreys' Prior with two parameters

#### Usage

```
jpf2p(ggd, detg, ggi)
```

### Arguments

ggd gradient of the expected information matrix
detg determinant of the expected information matrix
ggi inverse of the expected information matrix

### Value

Vector of 2 values

jpf3p Jeffreys' Prior with three parameters

## Description

Jeffreys' Prior with three parameters

```
jpf3p(ggd, detg, ggi)
```

lnorm\_cp

#### **Arguments**

ggd gradient of the expected information matrix
detg determinant of the expected information matrix
ggi inverse of the expected information matrix

#### Value

Vector of 3 values

jpf4p Jeffreys' Prior with four parameters

### Description

Jeffreys' Prior with four parameters

#### Usage

```
jpf4p(ggd, detg, ggi)
```

#### **Arguments**

ggd gradient of the expected information matrix
detg determinant of the expected information matrix
ggi inverse of the expected information matrix

#### Value

Vector of 4 values

Inorm\_cp Log-normal Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

Inorm\_cp 571

q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.

- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qlnorm_cp(
  х,
  p = seq(0.1, 0.9, 0.1),
 d1 = 0.01,
  fd2 = 0.01,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  rust = FALSE,
  nrust = 1e+05,
  debug = FALSE,
  aderivs = TRUE
)
rlnorm_cp(n, x, rust = FALSE, mlcp = TRUE, debug = FALSE, aderivs = TRUE)
dlnorm_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE, aderivs = TRUE)
plnorm_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE, aderivs = TRUE)
tlnorm_cp(n, x, debug = FALSE)
```

#### **Arguments**

x a vector of training data values
 p a vector of probabilities at which to generate predictive quantiles
 d1 if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
 fd2 if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter

572 lnorm\_cp

means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

• ml\_params: maximum likelihood estimates for the parameters.

lnorm\_cp 573

- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The log normal distribution has probability density function

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma x} e^{-(\log(x) - \mu)^2/(2\sigma^2)}$$

where x is the random variable and  $\mu, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

574 Inorm\_cp

If means=TRUE:

• ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible

• cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

Inorm\_cp 575

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),

576 lnorm\_cp

- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d35lnorm_example_data_v1
p=c(1:9)/10
q=qlnorm_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qlnorm_cp)",
main="Log-normal: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

lnorm\_dmgs\_cp 577

lnorm\_dmgs\_cp

Log-normal Distribution Predictions Based on a Calibrating Prior, using DMGS (for testing only)

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qlnorm_dmgs_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    d1 = 0.01,
    fd2 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    debug = FALSE,
    aderivs = TRUE
)
```

578 lnorm\_dmgs\_cp

```
x,
d1 = 0.01,
fd2 = 0.01,
mlcp = TRUE,
debug = FALSE,
aderivs = TRUE
)

dlnorm_dmgs_cp(x, y = x, d1 = 0.01, fd2 = 0.01, debug = FALSE, aderivs = TRUE)

plnorm_dmgs_cp(x, y, d1 = 0.01, fd2 = 0.01, debug = FALSE, aderivs = TRUE)
```

## Arguments

x	a vector of training data values
p	a vector of probabilities at which to generate predictive quantiles
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.

Inorm\_dmgs\_cp 579

- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The log normal distribution has probability density function

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma x} e^{-(\log(x) - \mu)^2/(2\sigma^2)}$$

where x is the random variable and  $\mu, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

580 lnorm\_dmgs\_cp

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

Inorm\_dmgs\_cp 581

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/. 582 lnorm\_dmgs\_cp

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (1st\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),

lnorm\_dmgs\_gg11 583

• Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d35lnorm_example_data_v1
p=c(1:9)/10
q=qlnorm_dmgs_cp(x,p)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qlnorm_dmgs_cp)",
main="Log-normal_DMGS: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
```

lnorm\_dmgs\_gg11

One component of the second derivative of the expected log-likelihood

### **Description**

One component of the second derivative of the expected log-likelihood

#### Usage

```
lnorm_dmgs_gg11(alpha, v1, d1, v2, fd2)
```

### Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Scalar value

584 lnorm\_dmgs\_gg22

lnorm_dmgs_gg12	One component of the second derivative of the expected log-likelihood
11101 111_011163_6612	One component of the second derivative of the expected tog tiketinood

# Description

One component of the second derivative of the expected log-likelihood

## Usage

```
lnorm_dmgs_gg12(alpha, v1, d1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Scalar value

Inorm_dmgs_gg22 One component of the second derivative of the expected log-likelih	ıood
------------------------------------------------------------------------------------	------

# Description

One component of the second derivative of the expected log-likelihood

## Usage

```
lnorm_dmgs_gg22(alpha, v1, d1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

lnorm\_dmgs\_loglik 585

### Value

Scalar value

### **Description**

log-likelihood function

### Usage

```
lnorm_dmgs_loglik(vv, x)
```

### Arguments

vv parameters

x a vector of training data values

### Value

Scalar value.

lnorm\_dmgs\_logscores Log scores for MLE and RHP predictions calculated using leave-oneout

### **Description**

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
lnorm_dmgs_logscores(logscores, x, d1 = 0.01, fd2 = 0.01)
```

## Arguments

logscores	logical that indicates v	whether to return	leave-one-out estimates	estimates of the
-----------	--------------------------	-------------------	-------------------------	------------------

log-score (much longer runtime)

x a vector of training data values

d1 the delta used in the numerical derivatives with respect to the parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Two scalars

586 Inorm\_dmgs\_mu1f

lnorm\_dmgs\_means
MLE and RHP predictive means

**Description** 

MLE and RHP predictive means

#### Usage

```
lnorm_dmgs_means(means, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2)
```

### **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

### Value

Two scalars

lnorm\_dmgs\_mu1f
DMGS equation 3.3, mu1 term

#### **Description**

DMGS equation 3.3, mu1 term

#### Usage

```
lnorm_dmgs_mu1f(alpha, v1, d1, v2, fd2)
```

### **Arguments**

alpha	a vector of values of alpha (one minus probability)

v1 first parameter

d1 the delta used in the numerical derivatives with respect to the parameter

v2 second parameter

fd2 the fractional delta used in the numerical derivatives with respect to the param-

eter

lnorm\_dmgs\_mu2f 587

## Value

Matrix

lnorm\_dmgs\_mu2f
DMGS equation 3.3, mu2 term

# Description

DMGS equation 3.3, mu2 term

## Usage

```
lnorm_dmgs_mu2f(alpha, v1, d1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

#### Value

3d array

lnorm\_dmgs\_p1f DMGS equation 3.3, p1 term

# Description

DMGS equation 3.3, p1 term

# Usage

```
lnorm\_dmgs\_p1f(y, v1, d1, v2, fd2)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

588 lnorm\_dmgs\_waic

## Value

Matrix

lnorm\_dmgs\_p2f

DMGS equation 3.3, p2 term

# Description

```
DMGS equation 3.3, p2 term
```

# Usage

```
lnorm_dmgs_p2f(y, v1, d1, v2, fd2)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

#### •

## Value

3d array

lnorm\_dmgs\_waic

Waic

# Description

Waic

## Usage

```
lnorm_dmgs_waic(
  waicscores,
  x,
  v1hat,
  d1,
  v2hat,
  fd2,
  lddi,
```

lnorm\_f1f 589

```
lddd,
lambdad,
aderivs
)
```

### Arguments

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values

v1hat first parameter

d1 the delta used in the numerical derivatives with respect to the parameter

v2hat second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

1ddi inverse observed information matrix 1ddd third derivative of log-likelihood

lambdad derivative of the log prior

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

lnorm_f1f	DMGS equation 3.3, f1 term

### **Description**

DMGS equation 3.3, f1 term

## Usage

```
lnorm_f1f(y, v1, d1, v2, fd2)
```

### Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

### Value

Matrix

590 Inorm\_f2f

-	~ 4	_
lnorm_	+1	tа

The first derivative of the density

## Description

The first derivative of the density

## Usage

```
lnorm_f1fa(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Vector

]	.n	or	rm.	_f	2f

DMGS equation 3.3, f2 term

# Description

DMGS equation 3.3, f2 term

# Usage

```
lnorm_f2f(y, v1, d1, v2, fd2)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

## Value

3d array

lnorm\_f2fa 591

٦.		 £2.	c _
ΤI	101	f2:	Га

The second derivative of the density

## Description

The second derivative of the density

### Usage

```
lnorm_f2fa(x, v1, v2)
```

### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

### Value

Matrix

lnorm\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
lnorm_fd(x, v1, v2)
```

### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

## Value

Vector

592 Inorm\_ldd

lnorm_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
lnorm_fdd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameterv2 second parameter

## Value

Matrix

lnorm_ldd	Second derivative matrix of the lnormalized log-likelihood

## Description

Second derivative matrix of the lnormalized log-likelihood

## Usage

```
lnorm_ldd(x, v1, d1, v2, fd2)
```

# **Arguments**

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

### Value

Square scalar matrix

lnorm\_ldda 593

-	
lnorm	Idda
T1101 III_	_tuua

The second derivative of the normalized log-likelihood

## Description

The second derivative of the normalized log-likelihood

## Usage

```
lnorm_ldda(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Matrix

1	norm	1 446	

Third derivative tensor of the lnormalized log-likelihood

### **Description**

Third derivative tensor of the lnormalized log-likelihood

# Usage

```
lnorm_lddd(x, v1, d1, v2, fd2)
```

## Arguments

Χ	a vector of training	data values

v1 first parameter

d1 the delta used in the numerical derivatives with respect to the parameter

v2 second parameter

fd2 the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Cubic scalar array

lnorm\_lmn

Inorm_lddda	
-------------	--

## Description

The third derivative of the normalized log-likelihood

# Usage

```
lnorm_lddda(x, v1, v2)
```

# Arguments

X	a vector of training data values
v1	first parameter

v2 second parameter

### Value

3d array

lnorm_lmn	One component of the second derivative of the normalized log-
	likelihood

# Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
lnorm_lmn(x, v1, d1, v2, fd2, mm, nn)
```

# Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

### Value

Scalar value

lnorm\_lmnp 595

lnorm_lmnp	One component of the second derivative of the normalized log-likelihood
------------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
lnorm_lmnp(x, v1, d1, v2, fd2, mm, nn, rr)
```

# Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

<pre>lnorm_logf</pre>	Logf for RUST	
-----------------------	---------------	--

# Description

Logf for RUST

## Usage

```
lnorm_logf(params, x)
```

# Arguments

params	model parameters for calculating logf
X	a vector of training data values

### Value

Scalar value.

596 Inorm\_logfddd

lnorm_logfdd	Second derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
lnorm_logfdd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

## Value

Matrix

lnorm_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
lnorm_logfddd(x, v1, v2)
```

## **Arguments**

X	a vector of training	data values
X	a vector of training	data varue

v1 first parameter v2 second parameter

### Value

3d array

lnorm\_logscores 597

Inorm_logscores Log scores for MLE and RHP predictions calculated using leave-one out	?-
---------------------------------------------------------------------------------------	----

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

# Usage

```
lnorm_logscores(logscores, x)
```

## **Arguments**

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

### Value

Two scalars

lnorm_mu1fa	Minus the	first derivative	of the o	cdf, at alpha
		J	-,	,

## Description

Minus the first derivative of the cdf, at alpha

## Usage

```
lnorm_mu1fa(alpha, v1, v2)
```

# Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

#### Value

Vector

598 Inorm\_p1fa

lnorm\_mu2fa

Minus the second derivative of the cdf, at alpha

## Description

Minus the second derivative of the cdf, at alpha

## Usage

```
lnorm_mu2fa(alpha, v1, v2)
```

## **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

### Value

Matrix

lnorm\_p1fa

The first derivative of the cdf

# Description

The first derivative of the cdf

## Usage

```
lnorm_p1fa(x, v1, v2)
```

### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Vector

lnorm\_p1\_cp

Log-normal Distribution with a Predictor, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qlnorm_p1_cp(
    x,
    t,
    t0 = NA,
    n0 = NA,
    p = seq(0.1, 0.9, 0.1),
    d1 = 0.01,
    d2 = 0.01,
    fd3 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    rust = FALSE,
    nrust = 1e+05,
    centering = TRUE,
```

```
debug = FALSE,
     aderivs = TRUE
    )
    rlnorm_p1_cp(
     n,
     Х,
     t,
      t0 = NA,
     n0 = NA,
     rust = FALSE,
     mlcp = TRUE,
     debug = FALSE,
     aderivs = TRUE
    )
    dlnorm_p1_cp(
     Х,
     t,
     t0 = NA,
     n0 = NA,
     y = x,
     rust = FALSE,
     nrust = 1000,
     centering = TRUE,
     debug = FALSE,
     aderivs = TRUE
    )
   plnorm_p1_cp(
     х,
      t,
     t0 = NA,
     n0 = NA,
     y = x,
     rust = FALSE,
     nrust = 1000,
     centering = TRUE,
     debug = FALSE,
     aderivs = TRUE
    )
    tlnorm_p1_cp(n, x, t, debug = FALSE)
Arguments
                    a vector of training data values
   Χ
    t
                    a vector of predictors, such that length(t)=length(x)
```

t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
p	a vector of probabilities at which to generate predictive quantiles
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
d2	if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter
fd3	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the third parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- $\bullet \ \, {\tt ml\_quantiles:} \ \, {\tt quantiles:} \ \, {\tt quantiles} \ \, {\tt calculated} \ \, {\tt using} \ \, {\tt maximum} \ \, {\tt likelihood.}$
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

• predictedparameter: the estimated value for parameter, as a function of the predictor.

• adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The log normal distribution with a predictor has probability density function

$$f(x; a, b, \sigma) = \frac{1}{\sqrt{2\pi}x\sigma}e^{-(\log(x) - \mu(a, b))^2/(2\sigma^2)}$$

where x is the random variable,  $\mu = a + bt$  is the location parameter of the log of the random variable, modelled as a function of parameters a, b and predictor t, and  $\sigma > 0$  is the scale parameter of the log of the random variable.

The calibrating prior is given by the right Haar prior, which is

$$\pi(a,b,\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

## **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

· Cauchy (cauchy),

- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

lnorm\_p1\_f1f

### **Examples**

```
#
# example 1
x=fitdistcp::d61lnorm_p1_example_data_v1_x
tt=fitdistcp::d61lnorm_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qlnorm_p1_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qlnorm_p1_cp)",
main="Log-Normal w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

lnorm\_p1\_f1f

DMGS equation 2.1, f1 term

## Description

DMGS equation 2.1, f1 term

## Usage

```
lnorm_p1_f1f(y, t0, v1, d1, v2, d2, v3, fd3)
```

### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Matrix

lnorm\_p1\_f1fa 607

	lnorm.	n1	f1	fa
--	--------	----	----	----

The first derivative of the density

## **Description**

The first derivative of the density

## Usage

```
lnorm_p1_f1fa(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

٦	n	_	rn	_	ní	1 1	F٦.	f
- 1	n	Ю	rı	1	n	1 1	Γ/	Г

DMGS equation 2.1, f2 term

# Description

DMGS equation 2.1, f2 term

## Usage

```
lnorm_p1_f2f(y, t0, v1, d1, v2, d2, v3, fd3)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the param-
	eter

lnorm\_p1\_fd

### Value

3d array

lnorm\_p1\_f2fa

The second derivative of the density

### **Description**

The second derivative of the density

### Usage

```
lnorm_p1_f2fa(x, t, v1, v2, v3)
```

### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

lnorm\_p1\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### **Description**

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
lnorm_p1_fd(x, t, v1, v2, v3)
```

## Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

lnorm\_p1\_fdd 609

## Value

Vector

lnorm_p1_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
lnorm_p1_fdd(x, t, v1, v2, v3)
```

## **Arguments**

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

v3 third parameter

## Value

Matrix

lnorm_p1_ldd Second derivative matrix of the normalized log-likelihood	
------------------------------------------------------------------------	--

# Description

Second derivative matrix of the normalized log-likelihood

# Usage

```
lnorm_p1_ldd(x, t, v1, d1, v2, d2, v3, fd3)
```

lnorm\_p1\_ldda

# Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Square scalar matrix

lnorm_p1_ldda	The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

# Usage

```
lnorm_p1_ldda(x, t, v1, v2, v3)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

## Value

Matrix

lnorm\_p1\_lddd 611

lnorm_p1_lddd	Third derivative tensor of the normalized log-likelihood
---------------	----------------------------------------------------------

## Description

Third derivative tensor of the normalized log-likelihood

# Usage

```
lnorm_p1_lddd(x, t, v1, d1, v2, d2, v3, fd3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Cubic scalar array

lnorm_p1_lddda
----------------

## Description

The third derivative of the normalized log-likelihood

# Usage

```
lnorm_p1_lddda(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

lnorm\_p1\_lmn

## Value

3d array

<pre>lnorm_p1_lmn</pre>	1	norm_p1_lmn		log-
-------------------------	---	-------------	--	------

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
lnorm_p1_lmn(x, t, v1, d1, v2, d2, v3, fd3, mm, nn)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

## Value

Scalar value

lnorm\_p1\_lmnp 613

lnorm_p1_lmnp	One component of the second derivative of the normalized log-likelihood
---------------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
lnorm_p1_lmnp(x, t, v1, d1, v2, d2, v3, fd3, mm, nn, rr)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

#### Value

Scalar value

	Logf for RUST	<pre>lnorm_p1_logf</pre>
--	---------------	--------------------------

# Description

Logf for RUST

# Usage

```
lnorm_p1_logf(params, x, t)
```

lnorm\_p1\_logfdd

## Arguments

params	model parameters for calculating log	ζŤ

x a vector of training data valuest a vector or matrix of predictors

#### Value

Scalar value.

lnorm_p1_logfdd	Second derivative of the log density Created by Stephen Jewson using
	Deriy() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
lnorm_p1_logfdd(x, t, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

v3 third parameter

#### Value

lnorm\_p1\_logfddd 615

lnorm_p1_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	3

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
lnorm_p1_logfddd(x, t, v1, v2, v3)
```

## **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
	_

v2 second parameter v3 third parameter

#### Value

3d array

lnorm\_p1\_loglik

Log-normal-with-p1 observed log-likelihood function

## Description

Log-normal-with-p1 observed log-likelihood function

## Usage

```
lnorm_p1_loglik(vv, x, t)
```

# Arguments

vv pa	rameters

x a vector of training data valuest a vector or matrix of predictors

#### Value

Scalar value.

lnorm\_p1\_mu1fa

lnorm_p1_logscores	Log scores for MLE and RHP predictions calculated using leave-one-out
--------------------	-----------------------------------------------------------------------

#### **Description**

Log scores for MLE and RHP predictions calculated using leave-one-out

#### Usage

```
lnorm_p1_logscores(logscores, x, t)
```

#### Arguments

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data valuest a vector or matrix of predictors

#### Value

Two scalars

lnorm\_p1\_mu1fa

Minus the first derivative of the cdf, at alpha

## Description

Minus the first derivative of the cdf, at alpha

#### Usage

```
lnorm_p1_mu1fa(alpha, t, v1, v2, v3)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

t a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

#### Value

Vector

lnorm\_p1\_mu2fa 617

7	- 4	2.0
Inorm	nΙ	_mu2fa
T1101 III_	_ 🖊 1	a

Minus the second derivative of the cdf, at alpha

#### **Description**

Minus the second derivative of the cdf, at alpha

#### Usage

```
lnorm_p1_mu2fa(alpha, t, v1, v2, v3)
```

## Arguments

alpha a vector of values of alpha (one minus probability)

t a vector or matrix of predictors

v1 first parameter v2 second parameter v3 third parameter

#### Value

Matrix

lnorm\_p1\_p1fa

The first derivative of the cdf

## Description

The first derivative of the cdf

#### Usage

```
lnorm_p1_p1fa(x, t, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameter v2 second parameter v3 third parameter

#### Value

Vector

lnorm\_p1\_pd

lnorm	n1	n2fa
Inorm	DΙ	D∠⊤a

The second derivative of the cdf

## Description

The second derivative of the cdf

#### Usage

```
lnorm_p1_p2fa(x, t, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
4	C

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

1norm	n1	nd
THOLIII	υı	100

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
lnorm_p1_pd(x, t, v1, v2, v3)
```

#### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

lnorm\_p1\_pdd 619

## Value

Vector

lnorm_p1_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
lnorm_p1_pdd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

## Value

Matrix

```
lnorm_p1_predictordata
```

Predicted Parameter and Generalized Residuals

# Description

Predicted Parameter and Generalized Residuals

# Usage

```
lnorm_p1_predictordata(x, t, t0, params)
```

620 lnorm\_p1\_waic

# Arguments

a vector of training data values Х t a vector or matrix of predictors

a single value of the predictor (specify either t0 or n0 but not both) t0

model parameters for calculating logf params

## Value

Two vectors

lnorm\_p1\_waic Waic

## Description

Waic

## Usage

```
lnorm_p1_waic(
 waicscores,
 х,
 t,
 v1hat,
 d1,
 v2hat,
 d2,
 v3hat,
 fd3,
 aderivs = TRUE
)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
t	a vector or matrix of predictors
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3hat	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the param-
	eter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

lnorm\_p2fa 621

#### Value

Two numeric values.

lnorm\_p2fa

The second derivative of the cdf

#### **Description**

The second derivative of the cdf

## Usage

```
lnorm_p2fa(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

lnorm\_pd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
lnorm_pd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

lnorm\_waic

lnorm_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	by Andrew Cidusen and Serguei Sokoi

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
lnorm_pdd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

lnorm_waic	Waic for RUST

# Description

Waic for RUST

# Usage

```
lnorm_waic(waicscores, x, v1hat, d1, v2hat, fd2, aderivs)
```

## Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
x	a vector of training data values
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

logis\_cp

Logistic Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qlogis_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    d1 = 0.01,
    fd2 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE,
    aderivs = TRUE
```

```
)
rlogis_cp(
 n,
 Х,
 d1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
dlogis_cp(
 Х,
 y = x,
 d1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
plogis_cp(
 Х,
 y = x,
 d1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
  aderivs = TRUE
)
tlogis_cp(n, x, d1 = 0.01, fd2 = 0.01, debug = FALSE)
```

## Arguments

x	a vector of training data values
p	a vector of probabilities at which to generate predictive quantiles
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter $% \left( 1\right) =\left( 1\right) \left( 1\right) $
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter $$
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)

1- i - 1 that in diagter sub-other to many additional coloral stress and actions actions to

waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

• ml\_params: maximum likelihood estimates for the parameters.

- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The logistic distribution has distribution function

$$f(x; \mu, \sigma) = \frac{1}{1 + e^{-(x-\mu)/\sigma}}$$

where x is the random variable and  $\mu, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

• ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible

• cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2024) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),

- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d40logis_example_data_v1
p=c(1:9)/10
q=qlogis_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qlogis_cp)",
main="Logistic: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

logis\_f1fa

-		٠			•		_
- 1	og	ъ.	C		+ 1	1	+
_	V5		J	_			

DMGS equation 3.3, f1 term

## Description

DMGS equation 3.3, f1 term

# Usage

```
logis_f1f(y, v1, d1, v2, fd2)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

## Value

Matrix

logi	c	f1	fa

The first derivative of the density

# Description

The first derivative of the density

## Usage

# Arguments

			4	1.4.	. 1
X	a vector	OΙ	training	uata	varues

v1 first parameter v2 second parameter

#### Value

Vector

logis\_f2f 631

-		
10	gis.	f2f

 $DMGS\ equation\ 3.3, f2\ term$ 

## Description

DMGS equation 3.3, f2 term

# Usage

```
logis_f2f(y, v1, d1, v2, fd2)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

## Value

3d array

- ·	000
logis_	ナフナコ
TUEIS_	_ 1

The second derivative of the density

# Description

The second derivative of the density

## Usage

```
logis_f2fa(x, v1, v2)
```

# Arguments

X	a vector of trai	ınıng data values

v1 first parameter v2 second parameter

#### Value

logis\_fdd

logis_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
logis_fd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

Second derivative of the density Created by Stephen Jewson using De-
riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
logis_fdd(x, v1, v2)
```

# Arguments

x a vector of training da	ata values
---------------------------	------------

v1 first parameter v2 second parameter

#### Value

logis\_ldd 633

logis_ldd	Second derivative matrix of the normalized log-likelihood

## Description

Second derivative matrix of the normalized log-likelihood

# Usage

```
logis_ldd(x, v1, d1, v2, fd2)
```

# Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

Value

Square scalar matrix

logis_ldda	The second derivative of the normalized log-likelihood	

# Description

The second derivative of the normalized log-likelihood

# Usage

```
logis_ldda(x, v1, v2)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter

## Value

logis\_lddda

-						
П.	02	r ni	C	ld	М	
	∪⊭	. 1	0	LU	ıu	u

Third derivative tensor of the normalized log-likelihood

## Description

Third derivative tensor of the normalized log-likelihood

## Usage

```
logis_lddd(x, v1, d1, v2, fd2)
```

## Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

## Value

Cubic scalar array

logis	1ddda

The third derivative of the normalized log-likelihood

## Description

The third derivative of the normalized log-likelihood

## Usage

```
logis_lddda(x, v1, v2)
```

# Arguments

X	a vector of training data values
v1	first parameter

v2 second parameter

#### Value

3d array

logis\_lmn 635

logis_lmn	One component of the second derivative of the normalized log-likelihood

# Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
logis_lmn(x, v1, d1, v2, fd2, mm, nn)
```

# Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

#### Value

Scalar value

logis_lmnp	One component of the third derivative of the normalized log-likelihood

# Description

One component of the third derivative of the normalized log-likelihood

# Usage

```
logis_lmnp(x, v1, d1, v2, fd2, mm, nn, rr)
```

logis\_logf

# Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

## Value

Scalar value

|--|

# Description

Logf for RUST

## Usage

```
logis_logf(params, x)
```

# Arguments

params model parameters for calculating logf x a vector of training data values

## Value

Scalar value.

logis\_logfdd 637

logis_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
logis_logfdd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

## Value

Matrix

logis_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
logis_logfddd(x, v1, v2)
```

## **Arguments**

X	a vector of training data values
Λ	a vector of training data values

v1 first parameter v2 second parameter

#### Value

3d array

logis\_logscores

logis_loglik	log-likelihood function

# Description

log-likelihood function

# Usage

```
logis_loglik(vv, x)
```

## Arguments

vv parameters

x a vector of training data values

#### Value

Scalar value.

logis_logscores	Log scores for MLE and RHP predictions calculated using leave-one-
	out

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
logis_logscores(logscores, x, d1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

# Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
X	a vector of training data values
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

logis\_mu1f 639

	mu1f

DMGS equation 3.3, mu1 term

## Description

DMGS equation 3.3, mu1 term

## Usage

```
logis_mu1f(alpha, v1, d1, v2, fd2)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

## Value

Matrix

-			_
- 1	ogis	mu i l	tа
_	05 - 0		

Minus the first derivative of the cdf, at alpha

# Description

Minus the first derivative of the cdf, at alpha

## Usage

```
logis_mu1fa(alpha, v1, v2)
```

## Arguments

oility)	
	oility)

v1 first parameter v2 second parameter

#### Value

Vector

logis\_mu2fa

_			_
٦.	gis	mı	∙o£
10	215	IIII	1/1

DMGS equation 3.3, mu2 term

## Description

DMGS equation 3.3, mu2 term

## Usage

```
logis_mu2f(alpha, v1, d1, v2, fd2)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

eter

## Value

3d array

-		~ ~
- 14	റമാട	_mu2fa
	UB + U.	a

Minus the second derivative of the cdf, at alpha

## Description

Minus the second derivative of the cdf, at alpha

## Usage

```
logis_mu2fa(alpha, v1, v2)
```

## Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

#### Value

logis\_p1f 641

-				
10	r n	C	_p1	I +
ΤĆ	<i>'</i> 5 1	. J_	_レ	

DMGS equation 3.3, p1 term

## Description

DMGS equation 3.3, p1 term

# Usage

```
logis_p1f(y, v1, d1, v2, fd2)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

## Value

Matrix

logis_	p1fa
10510_	-p : : u

The first derivative of the cdf

# Description

The first derivative of the cdf

# Usage

```
logis_p1fa(x, v1, v2)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter

#### Value

Vector

logis\_p1\_cp

Logistic Distribution with a Predictor, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qlogis_p1_cp(
    x,
    t,
    t0 = NA,
    n0 = NA,
    p = seq(0.1, 0.9, 0.1),
    d1 = 0.01,
    d2 = 0.01,
    fd3 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    cmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
```

```
predictordata = TRUE,
  centering = TRUE,
  debug = FALSE,
 aderivs = TRUE
)
rlogis_p1_cp(
 n,
 х,
  t,
  t0 = NA,
 n0 = NA,
 d1 = 0.01,
 d2 = 0.01,
  fd3 = 0.01,
  rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
  aderivs = TRUE
dlogis_p1_cp(
 х,
  t,
 t0 = NA,
 n0 = NA,
 y = x,
 d1 = 0.01,
 d2 = 0.01,
  fd3 = 0.01,
  rust = FALSE,
 nrust = 1000,
  centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
)
plogis_p1_cp(
 Х,
  t,
  t0 = NA,
 n0 = NA,
 y = x,
 d1 = 0.01,
 d2 = 0.01,
  fd3 = 0.01,
  rust = FALSE,
  nrust = 1000,
```

```
centering = TRUE,
debug = FALSE,
aderivs = TRUE
)

tlogis_p1_cp(n, x, t, d1 = 0.01, d2 = 0.01, fd3 = 0.01, debug = FALSE)
```

# Arguments

x	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
d2	if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter
fd3	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the third parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The logistic distribution with a predictor has distribution function

$$f(x; a, b, \sigma) = \frac{1}{1 + e^{-(x - \mu(a, b))/\sigma}}$$

where x is the random variable,  $\mu = a + bt$  is the location paramter, and  $\sigma > 0$  is the scale parameter. The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

## **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),

logis\_p1\_f1f 649

- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d62logis_p1_example_data_v1_x
tt=fitdistcp::d62logis_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qlogis_p1_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qlogis_p1_cp)",
main="Logistic w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

logis\_p1\_f1f

DMGS equation 2.1, f1 term

#### **Description**

```
DMGS equation 2.1, f1 term
```

```
logis_p1_f1f(y, t0, v1, d1, v2, d2, v3, fd3)
```

logis\_p1\_f1fa

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Matrix

# Description

The first derivative of the density

# Usage

```
logis_p1_f1fa(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

logis\_p1\_f2f 651

10	gis	n1	_f2f
	5 I J.	_0 .	_   _

DMGS equation 2.1, f2 term

## Description

DMGS equation 2.1, f2 term

## Usage

```
logis_p1_f2f(y, t0, v1, d1, v2, d2, v3, fd3)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

3d array

logis	n1	f2fa

The second derivative of the density

#### Description

The second derivative of the density

## Usage

```
logis_p1_f2fa(x, t, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

logis\_p1\_fdd

#### Value

Matrix

logis_p1_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	riv() by Anarew Clausen and Serguei Sokoi

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
logis_p1_fd(x, t, v1, v2, v3)
```

#### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

logis_p1_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
logis_p1_fdd(x, t, v1, v2, v3)
```

logis\_p1\_ldd 653

# Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Matrix

logis_p1_ldd Second	d derivative matrix of the normalized log-likelihood
---------------------	------------------------------------------------------

## Description

Second derivative matrix of the normalized log-likelihood

## Usage

```
logis_p1_ldd(x, t, v1, d1, v2, d2, v3, fd3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Square scalar matrix

logis\_p1\_lddd

-			-	-		
- 1	ogi	s n	ıΤ	- 1 (	hh	12

The second derivative of the normalized log-likelihood

#### **Description**

The second derivative of the normalized log-likelihood

#### Usage

```
logis_p1_ldda(x, t, v1, v2, v3)
```

#### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Matrix

logis	p1	Lddd	t

 $Third\ derivative\ tensor\ of\ the\ normalized\ log-likelihood$ 

## Description

Third derivative tensor of the normalized log-likelihood

#### Usage

```
logis_p1_lddd(x, t, v1, d1, v2, d2, v3, fd3)
```

## Arguments

)	X	a vector of training data values
1	t	a vector or matrix of predictors
١	v1	first parameter
(	d1	the delta used in the numerical derivatives with respect to the parameter
١	v2	second parameter
(	d2	the delta used in the numerical derivatives with respect to the parameter
١	v3	third parameter
İ	fd3	the fractional delta used in the numerical derivatives with respect to the parameter

logis\_p1\_lddda 655

#### Value

Cubic scalar array

logis_	n1	1ddda
IUEIS_	_レ ၊ _	_±uuua

The third derivative of the normalized log-likelihood

## Description

The third derivative of the normalized log-likelihood

#### Usage

```
logis_p1_lddda(x, t, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

third parameter

# Value

v3

3d array

logis_p1_lmn	One component of the second derivative of the normalized log-
	likelihood

## Description

One component of the second derivative of the normalized log-likelihood

```
logis_p1_lmn(x, t, v1, d1, v2, d2, v3, fd3, mm, nn)
```

logis\_p1\_lmnp

#### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

## Value

Scalar value

logis_p1_lmnp	One component of the second derivative of the normalized log-likelihood

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
logis_p1_lmnp(x, t, v1, d1, v2, d2, v3, fd3, mm, nn, rr)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the param-
	eter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

logis\_p1\_logf 657

#### Value

Scalar value

logis\_p1\_logf

Logf for RUST

#### Description

Logf for RUST

#### Usage

```
logis_p1_logf(params, x, t)
```

#### **Arguments**

params model parameters for calculating logf
x a vector of training data values
t a vector or matrix of predictors

#### Value

Scalar value.

logis\_p1\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
logis_p1_logfdd(x, t, v1, v2, v3)
```

## Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter

v2 second parameter v3 third parameter logis\_p1\_loglik

#### Value

Matrix

#### Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
logis_p1_logfddd(x, t, v1, v2, v3)
```

#### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

#### Value

3d array

## Description

Logistic-with-p1 observed log-likelihood function

#### Usage

```
logis_p1_loglik(vv, x, t)
```

#### Arguments

parameters

x a vector of training data valuest a vector or matrix of predictors

logis\_p1\_logscores 659

#### Value

Scalar value.

logis_p1_logscores	Log scores for MLE and RHP predictions calculated using leave-one- out
--------------------	---------------------------------------------------------------------------

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
logis_p1_logscores(logscores, x, t, d1, d2, fd3, aderivs = TRUE)
```

#### **Arguments**

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
x	a vector of training data values
t	a vector or matrix of predictors
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

logis_p1_means	Logistic distribution: RHP mean

## Description

Logistic distribution: RHP mean

```
logis_p1_means(t0, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2)
```

logis\_p1\_mu1f

#### **Arguments**

t0 a single value of the predictor (specify either t0 or n0 but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

#### Value

Two scalars

#### Description

DMGS equation 3.3, mu1 term

#### Usage

```
logis_p1_mu1f(alpha, t0, v1, d1, v2, d2, v3, fd3)
```

#### **Arguments**

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

logis\_p1\_mu1fa 661

<b>-</b>	-	-	_
logis	n'l	mul	t a

Minus the first derivative of the cdf, at alpha

#### **Description**

Minus the first derivative of the cdf, at alpha

#### Usage

```
logis_p1_mu1fa(alpha, t, v1, v2, v3)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

-		_	
lΩg	1.5	n1	mu2f

DMGS equation 3.3, mu2 term

## Description

DMGS equation 3.3, mu2 term

#### Usage

```
logis_p1_mu2f(alpha, t0, v1, d1, v2, d2, v3, fd3)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
	CICI

logis\_p1\_p1f

#### Value

3d array

 $logis_p1_mu2fa$ 

Minus the second derivative of the cdf, at alpha

#### Description

Minus the second derivative of the cdf, at alpha

# Usage

```
logis_p1_mu2fa(alpha, t, v1, v2, v3)
```

#### Arguments

alpha a vector of values of alpha (one minus probability)

t a vector or matrix of predictors

v1 first parameter v2 second parameter

v3 third parameter

#### Value

Matrix

 $logis_p1_p1f$ 

DMGS equation 2.1, p1 term

#### Description

DMGS equation 2.1, p1 term

```
logis_p1_p1f(y, t0, v1, d1, v2, d2, v3, fd3)
```

logis\_p1\_p1fa 663

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Matrix

# Description

The first derivative of the cdf

# Usage

```
logis_p1_p1fa(x, t, v1, v2, v3)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

logis\_p1\_p2fa

-			-		~ "
To	g1	S	D	l_p	21

DMGS equation 2.1, p2 term

## Description

DMGS equation 2.1, p2 term

#### Usage

```
logis_p1_p2f(y, t0, v1, d1, v2, d2, v3, fd3)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
	CtCl

#### Value

3d array

logis_p1	l_p2fa
----------	--------

The second derivative of the cdf

#### Description

The second derivative of the cdf

## Usage

```
logis_p1_p2fa(x, t, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

logis\_p1\_pd 665

#### Value

Matrix

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
logis_p1_pd(x, t, v1, v2, v3)
```

#### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

logis_p1_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
logis_p1_pdd(x, t, v1, v2, v3)
```

# Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

third parameter

#### Value

Matrix

v3

```
logis_p1_predictordata
```

Predicted Parameter and Generalized Residuals

#### Description

Predicted Parameter and Generalized Residuals

#### Usage

```
logis_p1_predictordata(predictordata, x, t, t0, params)
```

## Arguments

predictordata	logical that indicates whether to calculate and return predictordata
x	a vector of training data values
t	a vector or matrix of predictors
t0	a single value of the predictor (specify either t0 or n0 but not both)
params	model parameters for calculating logf

#### Value

Two vectors

logis\_p1\_waic 667

logis\_p1\_waic Waic

# Description

Waic

# Usage

```
logis_p1_waic(
  waicscores,
  x,
  t,
  v1hat,
  d1,
  v2hat,
  d2,
  v3hat,
  fd3,
  lddi,
  lddd,
  lambdad,
  aderivs = TRUE
)
```

## Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
x	a vector of training data values
t	a vector or matrix of predictors
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3hat	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

logis\_p2fa

#### Value

Two numeric values.

logis\_p2f

DMGS equation 3.3, p2 term

#### Description

DMGS equation 3.3, p2 term

#### Usage

```
logis_p2f(y, v1, d1, v2, fd2)
```

#### Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

#### Value

3d array

logis\_p2fa

The second derivative of the cdf

#### Description

The second derivative of the cdf

#### Usage

```
logis_p2fa(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

logis\_pd 669

logis_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
logis_pd(x, v1, v2)
```

#### Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

logis_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
logis_pdd(x, v1, v2)
```

## Arguments

x a vector of training	ing data values
------------------------	-----------------

v1 first parameter v2 second parameter

#### Value

logis_waic	Waic		

#### **Description**

Waic

#### Usage

```
logis_waic(waicscores, x, v1hat, d1, v2hat, fd2, lddi, lddd, lambdad, aderivs)
```

#### Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

lst_k3_cp	t Distribution Predictions Based on a Calibrating Prior

## Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.

- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qlst_k3_cp(
 p = seq(0.1, 0.9, 0.1),
 kdf = 5,
  d1 = 0.01
  fd2 = 0.01,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 debug = FALSE,
  aderivs = TRUE
)
rlst_k3_cp(
 n,
 Х,
 d1 = 0.01,
  fd2 = 0.01,
  kdf = 5,
  rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
dlst_k3_cp(
  х,
 y = x,
```

```
d1 = 0.01,
  fd2 = 0.01,
 kdf = 5,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
plst_k3_cp(
 х,
 y = x,
 d1 = 0.01,
 fd2 = 0.01,
 kdf = 5,
  rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
tlst_k3_{cp}(n, x, d1 = 0.01, fd2 = 0.01, kdf = 5, debug = FALSE)
```

#### Arguments

X	a vector of training data values
p	a vector of probabilities at which to generate predictive quantiles
kdf	the known degrees of freedom parameter
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.

n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The t distribution (also known as the location-scale t distribution, hence the name lst), has probability density function

$$f(x; \mu, \sigma) = \frac{\Gamma((\nu+1)/2)}{\sqrt{\pi\nu}\sigma\Gamma(\nu/2)} \left(1 + \frac{(x-\mu)^2}{\sigma^2\nu}\right)^{(\nu+1)/2}$$

where x is the random variable,  $\mu, \sigma > 0$  are the parameters, and we consider the degrees of freedom  $\nu$  to be known (hence the k3 in the name).

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (1st\_p1k3),

lst\_k3\_f1f 677

- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### Examples

```
#
# example 1
x=fitdistcp::d41lst_k3_example_data_v1
p=c(1:9)/10
q=qlst_k3_cp(x,p,kdf=5,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qlst_k3_cp)",
main="t: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

lst\_k3\_f1f

DMGS equation 3.3, f1 term

#### **Description**

```
DMGS equation 3.3, f1 term
```

```
lst_k3_f1f(y, v1, d1, v2, fd2, kdf)
```

678 lst\_k3\_f1fa

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

#### Value

Matrix

# Description

The first derivative of the density

## Usage

```
lst_k3_f1fa(x, v1, v2, kdf)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
kdf	the known degrees of freedom parameter

#### Value

Vector

lst\_k3\_f2f 679

1st\_k3\_f2f

DMGS equation 3.3, f2 term

## Description

DMGS equation 3.3, f2 term

#### Usage

```
lst_k3_f2f(y, v1, d1, v2, fd2, kdf)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

#### Value

3d array

lst\_k3\_f2fa

The second derivative of the density

#### Description

The second derivative of the density

#### Usage

```
lst_k3_f2fa(x, v1, v2, kdf)
```

#### Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
kdf	the known degrees of freedom parameter

#### Value

680 lst\_k3\_fdd

lst_k3_fd First derivative of the density Created by Stephen Jewson using riv() by Andrew Clausen and Serguei Sokol	De-
---------------------------------------------------------------------------------------------------------------------	-----

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
lst_k3_fd(x, v1, v2, v3)
```

#### **Arguments**

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

## Value

Vector

lst_k3_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

#### Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
lst_k3_fdd(x, v1, v2, v3)
```

#### Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

lst\_k3\_ldd 681

-			
$lst_{-}$	kЗ	Idd	

Second derivative matrix of the normalized log-likelihood

## Description

Second derivative matrix of the normalized log-likelihood

#### Usage

```
lst_k3_ldd(x, v1, d1, v2, fd2, kdf)
```

#### Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

#### Value

Square scalar matrix

7 - 4	1.3	1dd:	
ICT	$\nu$	יחחו	ב

The second derivative of the normalized log-likelihood

#### Description

The second derivative of the normalized log-likelihood

# Usage

```
lst_k3_ldda(x, v1, v2, kdf)
```

#### Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
kdf	the known degrees of freedom parameter

#### Value

lst\_k3\_lddda

		7 1 1
let	$\nu$	_lddc
エンに_	_NJ_	$_{\perp}$ $_{\perp}$ $_{\sqcup}$ $_{\sqcup}$

Third derivative tensor of the normalized log-likelihood

## Description

Third derivative tensor of the normalized log-likelihood

#### Usage

```
lst_k3_lddd(x, v1, d1, v2, fd2, kdf)
```

#### Arguments

<	a vector of training data values
/1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
/2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
	k v1 d1 v2 fd2 kdf

#### Value

Cubic scalar array

ls	t ŀ	<u>ر3_</u>	10	dd	da
	·	<b>`</b> -	~		u

The third derivative of the normalized log-likelihood

## Description

The third derivative of the normalized log-likelihood

#### Usage

```
lst_k3_lddda(x, v1, v2, kdf)
```

#### Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter

kdf the known degrees of freedom parameter

#### Value

3d array

lst\_k3\_lmn 683

lst_k3_lmn		component ihood	of	the	second	derivative	of	the	normalized	log-
	likeli	ihood								

# Description

One component of the second derivative of the normalized log-likelihood

#### Usage

```
lst_k3_lmn(x, v1, d1, v2, fd2, kdf, mm, nn)
```

## Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

#### Value

Scalar value

lst_k3_lmnp	One component of the third derivative of the normalized log-likelihood

## Description

One component of the third derivative of the normalized log-likelihood

```
lst_k3_lmnp(x, v1, d1, v2, fd2, kdf, mm, nn, rr)
```

684 lst\_k3\_logf

# Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

#### Value

Scalar value

$lst_k3_logf$ $Logf for RUST$
-------------------------------

# Description

Logf for RUST

## Usage

```
lst_k3_logf(params, x, kdf)
```

## Arguments

params model parameters for calculating logf
x a vector of training data values

kdf the known degrees of freedom parameter

#### Value

Scalar value.

lst\_k3\_logfdd 685

lst_k3_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
lst_k3_logfdd(x, v1, v2, v3)
```

## Arguments

x a ve	tor of training data values
--------	-----------------------------

v1 first parameterv2 second parameterv3 third parameter

#### Value

#### Matrix

lst_k3_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
lst_k3_logfddd(x, v1, v2, v3)
```

## Arguments

Х	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

3d array

lst\_k3\_logscores

lst_k3_loglik	log-likelihood function
---------------	-------------------------

#### **Description**

log-likelihood function

## Usage

```
lst_k3_loglik(vv, x, kdf)
```

## Arguments

vv parameters

x a vector of training data values

kdf the known degrees of freedom parameter

## Value

Scalar value.

lst_k3_logscores	Log scores for MLE and RHP predictions calculated using leave-one-
	out

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
lst_k3_logscores(logscores, x, d1 = 0.01, fd2 = 0.01, kdf, aderivs = TRUE)
```

# Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the
	log-score (much longer runtime)
X	a vector of training data values
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter
kdf	the known degrees of freedom parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

lst\_k3\_mu1f 687

lst_k3_mu1f	DMGS equation 3.3, mu1 term
-------------	-----------------------------

# Description

DMGS equation 3.3, mu1 term

# Usage

```
lst_k3_mu1f(alpha, v1, d1, v2, fd2, kdf)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

## Value

Matrix

|--|

# Description

DMGS equation 3.3, mu2 term

# Usage

```
lst_k3_mu2f(alpha, v1, d1, v2, fd2, kdf)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

688 lst\_k3\_p2f

## Value

3d array

lst\_k3\_p1f

DMGS equation 3.3, p1 term

## Description

DMGS equation 3.3, p1 term

# Usage

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

#### Value

Matrix

 $lst_k3_p2f$ 

DMGS equation 3.3, p2 term

# Description

DMGS equation 3.3, p2 term

lst\_k3\_waic 689

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

## Value

3d array

# Description

Waic

# Usage

```
lst_k3_waic(
  waicscores,
  x,
  v1hat,
  d1,
  v2hat,
  fd2,
  kdf,
  lddi,
  lddd,
  lambdad,
  aderivs
)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter

fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

lst_p1k3_cp	t Distribution with a Predictor, Predictions Based on a Calibrating
130_p183_cp	Prior

## Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qlst_p1k3_cp(
 Х,
  t,
  t0 = NA,
 n0 = NA,
 p = seq(0.1, 0.9, 0.1),
 d1 = 0.01,
  d2 = 0.01,
  fd3 = 0.01,
  kdf = 10,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  dmgs = TRUE,
  rust = FALSE,
  nrust = 1e+05,
 predictordata = TRUE,
  centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
)
rlst_p1k3_cp(
 n,
 Х,
  t,
  t0 = NA,
 n0 = NA,
 d1 = 0.01,
 d2 = 0.01,
  fd3 = 0.01,
 kdf = 10,
  rust = FALSE,
 mlcp = TRUE,
  centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
)
dlst_p1k3_cp(
 Х,
  t,
 t0 = NA,
 n0 = NA,
 y = x,
  d1 = 0.01,
 d2 = 0.01,
```

```
fd3 = 0.01,
  kdf = 10,
  rust = FALSE,
  nrust = 1000,
  centering = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
plst_p1k3_cp(
 Х,
 t,
  t0 = NA,
 n0 = NA,
 y = x,
 d1 = 0.01,
 d2 = 0.01,
  fd3 = 0.01,
 kdf = 10,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
tlst_p1k3_cp(
 n,
 Х,
  t,
 d1 = 0.01,
 d2 = 0.01,
 fd3 = 0.01,
 kdf = 10,
 debug = FALSE
)
```

## Arguments

X	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
p	a vector of probabilities at which to generate predictive quantiles
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
d2	if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter

fd3	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the third parameter
kdf	the known degrees of freedom parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- $\bullet \ \, {\tt ml\_quantiles:} \ \, {\tt quantiles:} \ \, {\tt quantiles} \ \, {\tt calculated} \ \, {\tt using} \ \, {\tt maximum} \ \, {\tt likelihood.}$
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The t distribution with a predictor (also known as the location-scale t distribution with a predictor, hence the name lst), has probability density function

$$f(x; a, b, \sigma) = \frac{\Gamma((\nu + 1)/2)}{\sqrt{\pi \nu} \sigma \Gamma(\nu/2)} \left( 1 + \frac{(x - \mu(a, b))^2}{\sigma^2 \nu} \right)^{(\nu + 1)/2}$$

where x is the random variable,  $\mu = a + bt$  is the location parameter, and  $\sigma > 0$  is the scale parameter. We consider the degrees of freedom  $\nu$  to be known (hence the k3 in the name).

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),

• Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d63lst_p1k3_example_data_v1_x
tt=fitdistcp::d63lst_p1k3_example_data_v1_t
p=c(1:9)/10
n0=10
q=qlst_p1k3_cp(x,tt,n0=n0,p=p,kdf=5,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qlst_p1k3_cp)",
main="t w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

lst\_p1k3\_f1f

DMGS equation 2.1, f1 term

#### **Description**

DMGS equation 2.1, f1 term

#### Usage

```
lst_p1k3_f1f(y, t0, v1, d1, v2, d2, v3, fd3, kdf)
```

#### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the param-
	eter
kdf	the known degrees of freedom parameter

lst\_p1k3\_f1fa 699

## Value

Matrix

lst\_p1k3\_f1fa

The first derivative of the density

# Description

The first derivative of the density

## Usage

```
lst_p1k3_f1fa(x, t, v1, v2, v3, kdf)
```

# Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

v3 third parameter

kdf the known degrees of freedom parameter

## Value

Vector

 $lst_p1k3_f2f$ 

DMGS equation 2.1, f2 term

## Description

DMGS equation 2.1, f2 term

```
lst_p1k3_f2f(y, t0, v1, d1, v2, d2, v3, fd3, kdf)
```

700 lst\_p1k3\_f2fa

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

# Value

3d array

# Description

The second derivative of the density

## Usage

```
lst_p1k3_f2fa(x, t, v1, v2, v3, kdf)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
kdf	the known degrees of freedom parameter

# Value

Matrix

lst\_p1k3\_fd 701

lst_p1k3_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
lst_p1k3_fd(x, t, v1, v2, v3, v4)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Vector

lst_p1k3_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
lst_p1k3_fdd(x, t, v1, v2, v3, v4)
```

702 lst\_p1k3\_ldd

## Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

lst\_p1k3\_ldd Second derivative matrix of the normalized log-likelihood

# Description

Second derivative matrix of the normalized log-likelihood

# Usage

```
lst_p1k3_ldd(x, t, v1, d1, v2, d2, v3, fd3, kdf)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

#### Value

Square scalar matrix

lst\_p1k3\_ldda 703

lst	11.1	<b>1</b> .	I -I -
ICT	nıĸ	≺ ור	เกล

The second derivative of the normalized log-likelihood

## Description

The second derivative of the normalized log-likelihood

## Usage

```
lst_p1k3_ldda(x, t, v1, v2, v3, kdf)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

kdf the known degrees of freedom parameter

## Value

Matrix

lst_	n1	k٦	1	hhh
13 t_	יש	·\J_		uuu

Third derivative tensor of the normalized log-likelihood

# Description

Third derivative tensor of the normalized log-likelihood

#### Usage

```
lst_p1k3_lddd(x, t, v1, d1, v2, d2, v3, fd3, kdf)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

704 lst\_p1k3\_lddda

v3	third parameter
----	-----------------

the fractional delta used in the numerical derivatives with respect to the param-

eter

kdf the known degrees of freedom parameter

#### Value

Cubic scalar array

lst\_p1k3\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

## Usage

```
lst_p1k3_lddda(x, t, v1, v2, v3, kdf)
```

# Arguments

x	a vector of training data values

t a vector or matrix of predictors

v1 first parameter
v2 second parameter

v3 third parameter

kdf the known degrees of freedom parameter

## Value

3d array

lst\_p1k3\_lmn 705

lst_p1k3_lmn	lst_p1k3_lmn	One component of the second derivative of the normalized log-likelihood
--------------	--------------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
lst_p1k3_lmn(x, t, v1, d1, v2, d2, v3, fd3, kdf, mm, nn)
```

## Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

## Value

Scalar value

lst_p1k3_lmnp	One component of the second derivative of the normalized log-likelihood

# Description

One component of the second derivative of the normalized log-likelihood

```
lst_p1k3_lmnp(x, t, v1, d1, v2, d2, v3, fd3, kdf, mm, nn, rr)
```

706 lst\_p1k3\_logf

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

## Value

Scalar value

lst_p1k3_logf	Logf for RUST	
---------------	---------------	--

# Description

Logf for RUST

## Usage

```
lst_p1k3_logf(params, x, t, kdf)
```

# Arguments

params	model parameters for calculating logf	
x	a vector of training data values	
t	a vector or matrix of predictors	
kdf	the known degrees of freedom parameter	

#### Value

Scalar value.

lst\_p1k3\_logfdd 707

lst_p1k3_logfdd Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol	lst_p1k3_logfdd	
----------------------------------------------------------------------------------------------------------------------------------	-----------------	--

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
lst_p1k3_logfdd(x, t, v1, v2, v3, v4)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

lst_p1k3_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
lst_p1k3_logfddd(x, t, v1, v2, v3, v4)
```

708 lst\_p1k3\_loglik

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

## Value

3d array

lst\_p1k3\_loglik

LST-with-p1 observed log-likelihood function

# Description

LST-with-p1 observed log-likelihood function

# Usage

```
lst_p1k3_loglik(vv, x, t, kdf)
```

# Arguments

vv	parameters
Х	a vector of training data values
t	a vector or matrix of predictors
kdf	the known degrees of freedom parameter

#### Value

Scalar value.

lst\_p1k3\_logscores 709

_, _ 0	Log scores for MLE and RHP predictions calculated using leave-one- out
--------	---------------------------------------------------------------------------

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

# Usage

```
lst_p1k3_logscores(logscores, x, t, d1, d2, fd3, kdf, aderivs)
```

## Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
X	a vector of training data values
t	a vector or matrix of predictors
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

## Value

Two scalars

lst_p1k3_mu1f	DMGS equation 3.3, mu1 term	

# Description

DMGS equation 3.3, mu1 term

```
lst_p1k3_mu1f(alpha, t0, v1, d1, v2, d2, v3, fd3, kdf)
```

710 lst\_p1k3\_mu2f

# Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the param-
	eter
kdf	the known degrees of freedom parameter

## Value

Matrix

DMGS equation 3.3, mu2 term
-----------------------------

# Description

DMGS equation 3.3, mu2 term

# Usage

```
lst_p1k3_mu2f(alpha, t0, v1, d1, v2, d2, v3, fd3, kdf)
```

## **Arguments**

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the param-
	eter
kdf	the known degrees of freedom parameter

## Value

3d array

lst\_p1k3\_p1f 711

lst\_p1k3\_p1f

DMGS equation 2.1, p1 term

# Description

DMGS equation 2.1, p1 term

# Usage

```
lst_p1k3_p1f(y, t0, v1, d1, v2, d2, v3, fd3, kdf)
```

# Arguments

У	value of random variable
t0	value of predictor
v1	first parameter
d1	delta for numerical derivative
v2	second parameter
d2	delta for numerical derivative
v3	third parameter
fd3	fractional delta for numerical derivative
kdf	the known number of degrees of freedom

## Value

Matrix

 $lst_p1k3_p2f$ 

DMGS equation 2.1, p2 term

## Description

DMGS equation 2.1, p2 term

```
lst_p1k3_p2f(y, t0, v1, d1, v2, d2, v3, fd3, kdf)
```

712 lst\_p1k3\_predictordata

## Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

## Value

3d array

lst\_p1k3\_predictordata

Predicted Parameter and Generalized Residuals

# Description

Predicted Parameter and Generalized Residuals

# Usage

```
lst_p1k3_predictordata(predictordata, x, t, t0, params, kdf)
```

# **Arguments**

predictordata	logical that indicates whether to calculate and return predictordata
x	a vector of training data values
t	a vector or matrix of predictors
t0	a single value of the predictor (specify either t0 or n0 but not both)
params	model parameters for calculating logf
kdf	the known degrees of freedom parameter

#### Value

Two vectors

lst\_p1k3\_setics 713

lst\_p1k3\_setics

Set initial conditions

## Description

Set initial conditions

## Usage

```
lst_p1k3_setics(x, t, ics)
```

# Arguments

x a vector of training data values
t a vector or matrix of predictors

ics initial conditions for the maximum likelihood search

#### Value

Vector

lst\_p1k3\_waic

Waic

# Description

Waic

```
lst_p1k3_waic(
 waicscores,
  х,
  t,
  v1hat,
 d1,
  v2hat,
  d2,
  v3hat,
  fd3,
  kdf,
  lddi,
  lddd,
 lambdad,
  aderivs
)
```

714 makemuhat0

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
t	a vector or matrix of predictors
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3hat	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)
lddd lambdad	third derivative of log-likelihood derivative of the log prior

## Value

Two numeric values.

	makemuhat0 M
--	--------------

# Description

Make muhat0

# Usage

```
makemuhat0(t0, n0, t, mle_params)
```

# Arguments

t0	the value of the predictor vector at which to make the prediction (if n0 not specified)
n0	the position in the predictor vector at which to make the prediction (positive integer less than or equal to the length of $x$ ) (if t0 not specified)
t	predictor
mle_params	MLE params

#### Value

Scalar

makeq 715

makeq

Calculates quantiles from simulations by inverting the Hazen CDF

## Description

Calculates quantiles from simulations by inverting the Hazen CDF

## Usage

```
makeq(yy, pp)
```

## Arguments

yy vector of samples
pp vector of probabilities

#### Value

Vector

maket0

Determine t0

## Description

Determine t0

## Usage

```
maket0(t0, n0, t)
```

# Arguments

a single value of the predictor (specify either t0 or n0 but not both)
 an index for the predictor (specify either t0 or n0 but not both)
 a vector or matrix of predictors

#### Value

Scalar

716 make\_cwaic

maketa0	Make ta0	

## Description

Make ta0

#### Usage

```
maketa0(t0, n0, t)
```

## Arguments

the value of the predictor vector at which to make the prediction (if n0 not spec-

ified)

no the position in the predictor vector at which to make the prediction (positive

integer less than or equal to the length of x) (if t0 not specified)

t predictor

## Value

Scalar

ike WAIC
----------

## Description

Make WAIC

# Usage

```
make_cwaic(x, fhatx, lddi, lddd, f1f, lambdad, f2f, dim)
```

## Arguments

dim

X	the training data
fhatx	density of x at the maximum likelihood parameters
lddi	inverse of the second derivative log-likelihood matrix
lddd	the third derivative log-likelihood tensor
f1f	the f1 term from DMGS equation 2.1
lambdad	the slope of the log prior
f2f	the f2 term from DMGS equation 2.1

number of free parameters

make\_maic 717

## Value

Two scalars

make\_maic

Calculate MAIC

# Description

Calculate MAIC

## Usage

```
make_maic(ml_value, nparams)
```

## Arguments

ml\_value maximum of the likelihood nparams number of parameters

#### Value

Vector of 3 values Returns the two components of MAIC, and their sum

make\_se

Make Standard Errors from lddi

# Description

Make Standard Errors from Iddi

## Usage

```
make_se(nx, lddi)
```

## Arguments

nx length of training data

1ddi the inverse log-likelihood matrix

#### Value

Vector

718 man

m	ake_waic	Make WAIC	

#### **Description**

Make WAIC

#### Usage

```
make_waic(x, fhatx, lddi, lddd, f1f, lambdad, f2f, dim)
```

#### **Arguments**

X	the training data
fhatx	density of x at the maximum likelihood parameters
lddi	inverse of the second derivative log-likelihood matrix
lddd	the third derivative log-likelihood tensor
f1f	the f1 term from DMGS equation 2.1
lambdad	the slope of the log prior
f2f	the f2 term from DMGS equation 2.1
dim	number of free parameters

#### Value

Two scalars

man	A blank function I use for setting up the man page information
-----	----------------------------------------------------------------

## Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y

man 719

- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

## Usage

man( х, t, t1, t2, t3, t0, t01, t02, t03, t10, t20, n0, n01, n02, n03, n10, n20, р, n, у, ics, kloc, kscale, kshape, kdf, kbeta, d1, fd1, d2, fd2, d3, fd3, d4, fd4,

d5,

720 man

```
fd5,
  d6,
  fd6,
  fdalpha,
 minxi,
 maxxi,
 dlogpi,
 means,
 waicscores,
 logscores,
 extramodels,
 pdf,
  customprior,
 dmgs,
 mlcp,
 predictordata,
  centering,
  nonnegslopesonly,
  rnonnegslopesonly,
 prior,
 debug,
  rust,
 nrust,
  pwm,
 unbiasedv,
  aderivs
)
```

# Arguments

X	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t1	a vector of predictors for the mean, such that length(t1)=length(x)
t2	a vector of predictors for the sd, such that length(t2)=length(x)
t3	a vector of predictors for the shape, such that length(t3)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
t10	a single value of the predictor for the mean (specify either $t10$ or $n10$ but not both)
t20	a single value of the predictor for the sd (specify either t20 or n20 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
n01	an index for the predictor (specify either t01 or n01 but not both)
n02	an index for the predictor (specify either t02 or n02 but not both)

n03	an index for the predictor (specify either t03 or n03 but not both)
n10	an index for the predictor for the mean (specify either t10 or n10 but not both)
n20	an index for the predictor for the sd (specify either t20 or n20 but not both)
р	a vector of probabilities at which to generate predictive quantiles
n	the number of random samples required
у	a vector of values at which to calculate the density and distribution functions
ics	initial conditions for the maximum likelihood search
kloc	the known location parameter
kscale	the known scale parameter
kshape	the known shape parameter
kdf	the known degrees of freedom parameter
kbeta	the known beta parameter
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter $% \left( 1\right) =\left( 1\right) \left( 1\right) $
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter $$
d2	if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter $% \left( 1\right) =\left( 1\right) \left( 1\right)$
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter $$
d3	if aderivs=FALSE, the delta used for numerical derivatives with respect to the third parameter $% \left( 1\right) =\left( 1\right) \left( 1\right) $
fd3	if $aderivs=FALSE$ , the fractional delta used for numerical derivatives with respect to the third parameter
d4	if $aderivs=FALSE$ , the delta used for numerical derivatives with respect to the fourth parameter
fd4	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the fourth parameter
d5	if aderivs=FALSE, the delta used for numerical derivatives with respect to the fifth parameter $% \left( 1\right) =\left( 1\right) \left( 1\right) $
fd5	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the fourth parameter $\frac{1}{2}$
d6	if aderivs=FALSE, the delta used for numerical derivatives with respect to the sixth parameter $\frac{1}{2}$
fd6	if $aderivs=FALSE$ , the fractional delta used for numerical derivatives with respect to the fourth parameter
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
minxi	the minimum allowed value of the shape parameter (decrease with caution)
maxxi	the maximum allowed value of the shape parameter (increase with caution)

dlogpi	gradient of the log prior
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)
extramodels	logical that indicates whether to run additional calculations and add three additional prediction models (longer runtime)
pdf	logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer runtime)
customprior	a custom value for the slope of the log prior at the maxlik estimate
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
nonnegslopeson	
_	logical that indicates whether to disallow non-negative slopes
rnonnegslopeso	
	logical that indicates whether to disallow non-negative slopes
prior	logical indicating which prior to use
debug	logical for turning on debug messages
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
pwm	logical for whether to include PWM results (longer runtime)
unbiasedv	logical for whether to include unbiased variance results in norm
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.

### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.

- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

## **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

#### If logscores=TRUE:

• ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)

• cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

#### If rust=TRUE:

 ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

#### If rust=TRUE:

 ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Optional Return Values (EVT models only)**

q\*\*\*\* optionally returns the following, for EVT models only:

• cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

#### Optional Return Values (some EVT models only)

q\*\*\*\* optionally returns the following, for some EVT models only:

#### If extramodels=TRUE:

- flat\_quantiles: predictive quantiles calculated from Bayesian integration with a flat prior.
- rh\_ml\_quantiles: predictive quantiles calculated from Bayesian integration with the calibrating prior, and the maximmum likelihood estimate for the shape parameter.

jp\_quantiles: predictive quantiles calculated from Bayesian integration with Jeffreys' prior.

r\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_deviates: predictive random deviates calculated using a Bayesian analysis with a flat prior.
- rh\_ml\_deviates: predictive random deviates calculated using a Bayesian analysis with the RHP-MLE prior.
- jp\_deviates: predictive random deviates calculated using a Bayesian analysis with the JP.

d\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_pdf: predictive density function from a Bayesian analysis with the flat prior.
- rh\_ml\_pdf: predictive density function from a Bayesian analysis with the RHP-MLE prior.
- jp\_pdf: predictive density function from a Bayesian analysis with the JP.

p\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_cdf: predictive distribution function from a Bayesian analysis with the flat prior.
- rh\_ml\_cdf: predictive distribution function from a Bayesian analysis with the RHP-MLE prior.
- jp\_cdf: predictive distribution function from a Bayesian analysis with the JP.

These additional predictive distributions are included for comparison with the calibrating prior model. They generally give less good reliability than the calibrating prior.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2024) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

#### **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),

728 man2f

• Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

man1f

Return message for flf, plf, mulf

# Description

Return message for f1f, p1f, mu1f

### Usage

man1f()

### Value

Matrix

man2f

Return message for f2f, p2f, mu2f

# Description

Return message for f2f, p2f, mu2f

# Usage

man2f()

#### Value

3d array

mandsub 729

mandsub

Return message for dsub

# Description

Return message for dsub

# Usage

```
mandsub()
```

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

manf

Blank function I use for setting up the man page information for the functions

# Description

Blank function I use for setting up the man page information for the functions

# Usage

```
manf(
  dim,
  νv,
  ml_params,
  nx,
  nxx,
  Х,
  ХХ,
  t,
  t1,
  t2,
  t3,
  tt,
  tt1,
  tt2,
  tt3,
  tt2d,
  tt3d,
  t0,
  t01,
```

```
t02,
t03,
t10,
t20,
t30,
n0,
n10,
n20,
p,
n,
у,
ics,
ta,
ta0,
muhat0,
v1,
v1hat,
v1h,
d1,
fd1,
v2,
v2hat,
v2h,
d2,
fd2,
v3,
v3hat,
v3h,
d3,
fd3,
v4,
v4hat,
v4h,
d4,
fd4,
v5,
v5hat,
v5h,
d5,
v6,
v6hat,
v6h,
d6,
minxi,
maxxi,
ximin,
ximax,
fdalpha,
```

```
kscale,
kloc,
kshape,
kdf,
kbeta,
alpha,
ymn,
slope,
mu,
sigma,
sigma1,
sigma2,
scale,
shape,
хi,
xi1,
xi2,
lambda,
log,
mm,
nn,
rr,
lddi,
lddi_k2,
lddi_k3,
lddi_k4,
lddd,
lddd_k2,
lddd_k3,
lddd_k4,
lambdad,
lambdad_cp,
lambdad_rhp,
lambdad_flat,
lambdad_rh_mle,
lambdad_rh_flat,
lambdad_jp,
lambdad_custom,
means,
waicscores,
logscores,
extramodels,
pdf,
predictordata,
nonnegslopesonly,
rnonnegslopesonly,
customprior,
prior,
```

```
params,
yy,
pp,
dlogpi,
debug,
centering,
aderivs
```

# Arguments

dim	number of parameters
VV	parameters
ml_params	parameters
nx	length of training data
nxx	length of training data
x	a vector of training data values
xx	a vector of training data values
t	a vector or matrix of predictors
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
tt	a vector of predictors
tt1	a vector of predictors for the mean
tt2	a vector of predictors for the sd
tt3	a vector of predictors for the shape
tt2d	a matrix of predictors (nx by 2)
tt3d	a matrix of predictors (nx by 3)
t0	a single value of the predictor (specify either t0 or n0 but not both)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
t10	a single value of the predictor for the mean (specify either $t10$ or $n10$ but not both)
t20	a single value of the predictor for the sd (specify either t20 or n20 but not both)
t30	a single value of the predictor for the shape (specify either $t30$ or $n30$ but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
n10	an index for the predictor for the mean (specify either t10 or n10 but not both)
n20	an index for the predictor for the sd (specify either t10 or n10 but not both)

p a vector of probabilities at which to generate predictive quantiles

n number of random samples required

y a vector of values at which to calculate the density and distribution functions

ics initial conditions for the maximum likelihood search

ta predictor residuals

ta0 predictor residual at the point being predicted

muhat at the point being predicted

v1 first parameter v1hat first parameter v1h first parameter

d1 the delta used in the numerical derivatives with respect to the parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameterv2hat second parameterv2h second parameter

d2 the delta used in the numerical derivatives with respect to the parameter

fd2 the fractional delta used in the numerical derivatives with respect to the param-

eter

v3 third parameter v3hat third parameter v3h third parameter

d3 the delta used in the numerical derivatives with respect to the parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

v4 fourth parameterv4hat fourth parameterv4h fourth parameter

d4 the delta used in the numerical derivatives with respect to the parameter

fd4 the fractional delta used in the numerical derivatives with respect to the param-

eter

v5 fifth parameter v5hat fifth parameter v5h fifth parameter

d5 the delta used in the numerical derivatives with respect to the parameter

v6 sixth parameter v6hat sixth parameter v6h sixth parameter

d6 the delta used in the numerical derivatives with respect to the parameter

minxi minimum value of shape parameter xi
maxxi maximum value of shape parameter xi
ximin minimum value of shape parameter xi
ximax maximum value of shape parameter xi

fdalpha the fractional delta used in the numerical derivatives with respect to probability,

for calculating the pdf as a function of quantiles

kscale the known scale parameter
kloc the known location parameter
kshape the known shape parameter

kdf the known degrees of freedom parameter

kbeta the known beta parameter

alpha a vector of values of alpha (one minus probability)
ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor
mu the location parameter of the distribution
sigma the sigma parameter of the distribution

sigma1 first coefficient for the sigma parameter of the distribution sigma2 second coefficient for the sigma parameter of the distribution

scale the scale parameter of the distribution shape the shape parameter of the distribution the shape parameter of the distribution

xi1 first coefficient for the shape parameter of the distribution xi2 second coefficient for the shape parameter of the distribution

lambda the lambda parameter of the distribution

log logical for the density evaluation

nn an index for which derivative to calculate
nn an index for which derivative to calculate
rr an index for which derivative to calculate
lddi inverse observed information matrix

lddi\_k2 inverse observed information matrix, fixed shape parameter
 lddi\_k3 inverse observed information matrix, fixed shape parameter
 lddi\_k4 inverse observed information matrix, fixed shape parameter

1ddd third derivative of log-likelihood

lddd\_k2 third derivative of log-likelihood, fixed shape parameter
 lddd\_k3 third derivative of log-likelihood, fixed shape parameter
 lddd\_k4 third derivative of log-likelihood, fixed shape parameter

lambdad derivative of the log prior derivative of the log prior

lambdad\_rhp derivative of the log RHP prior lambdad\_flat derivative of the log flat prior

lambdad\_rh\_mle derivative of the log CRHP-MLE prior

lambdad\_rh\_flat

derivative of the log CRHP-FLAT prior

lambdad\_jp derivative of the log JP prior

lambdad\_custom custom value of the derivative of the log prior

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

extramodels logical that indicates whether to add three additional prediction models

pdf logical that indicates whether to return density functions evaluated at quantiles

specified by input probabilities

predictordata logical that indicates whether to calculate and return predictordata

nonnegslopesonly

logical that indicates whether to disallow non-negative slopes

rnonnegslopesonly

logical that indicates whether to disallow non-negative slopes

customprior a custom value for the slope of the log prior at the maxlik estimate

prior logical indicating which prior to use
params model parameters for calculating logf

yy vector of samples
pp vector of probabilities
dlogpi gradient of the log prior

debug debug flag

centering indicates whether the routine should center the data or not

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

No return value

736 manlnn

manldd

Return message for ldd

# Description

Return message for ldd

# Usage

manldd()

### Value

Square scalar matrix

manlddd

 $Return\ message\ for\ lddd$ 

# Description

Return message for lddd

# Usage

manlddd()

# Value

Cubic scalar array

manlnn

Return message for lnn

# Description

Return message for lnn

# Usage

manlnn()

#### Value

Scalar value

manlnnn 737

manlnnn

Return message for lnnn

# Description

Return message for lnnn

# Usage

manlnnn()

### Value

Scalar value

manlogf

Return message for Logf

# Description

Return message for Logf

# Usage

manlogf()

# Value

Scalar value.

 ${\tt manloglik}$ 

Return message for loglik

# Description

Return message for loglik

# Usage

manloglik()

#### Value

Scalar value.

738 manpredictor

manlogscores

Return message for logscores

# Description

Return message for logscores

# Usage

manlogscores()

### Value

Two scalars

manmeans

Return message for means

# Description

Return message for means

# Usage

manmeans()

# Value

Two scalars

 ${\tt manpredictor}$ 

Return message for predictor.

# Description

Return message for predictor.

# Usage

manpredictor()

#### Value

Two vectors

manyector 739

manvector

Return message for vector

# Description

Return message for vector

# Usage

manvector()

# Value

Vector

manwaic

Return message for WAIC

# Description

Return message for WAIC

# Usage

manwaic()

## Value

Two numeric values.

 ${\tt movexiaway from zero}$ 

Move xi away from zero a bit

# Description

Move xi away from zero a bit

## Usage

movexiawayfromzero(xi)

## **Arguments**

хi

хi

### Value

Scalar

740 ms\_flat\_1tail

ms\_flat\_1tail Illustration of Model Selection Among 10 One Tail Distributions from the fitdistcp Package

#### **Description**

Applies model selection using AIC, WAIC1, WAIC2 and leave-one-out logscore to the input data x, for 10 one tailed models in the fitdistcp package (although for the GPD, the logscore is NA for mathematical reasons).

The code is straightforward, and the point is to illustrate what is possible using the model selection outputs from the fitdistcp routines.

The input data may be automatically shifted so that the minimum value is positive.

For the Pareto, the data may be further shifted so that the minimum value is slightly greater than 1.

## Usage

```
ms_flat_1tail(x)
```

#### **Arguments**

x data vector

#### **Details**

The 10 models are: exp, pareto\_k2, halfnorm, lnorm, frechet\_k1, weibull, gamma, invgamma, invgauss and gpd\_k1.

## Value

Plots QQ plots to the screen, for each of the models, and returns a data frame containing

- MLE parameter values
- AIC scores (times -0.5), AIC weights
- WAIC1 scores, WAIC1 weights
- WAIC2 scores, WAIC2 weights
- logscores, logscore weights
- maximum likelihood and calibrating prior means
- maximum likelihood and calibrating prior standard deviations

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

ms\_flat\_2tail 741

#### **Examples**

```
# because it's too slow for CRAN
set.seed(1)
nx=50
x=rlnorm(nx)
print(ms_flat_1tail(x))
```

ms\_flat\_2tail

Illustration of Model Selection Among 18 Distributions from the fitdistcp Package

# Description

Applies model selection using AIC, WAIC1, WAIC2 and leave-one-out logscore to the input data x, for 7 two tailed models in the fitdistcp packages

The code is straightforward, and the point is to illustrate what is possible using the model selection outputs from the fitdistcp routines.

### Usage

```
ms_flat_2tail(x)
```

## **Arguments**

Х

data vector

#### **Details**

The 7 models are: norm, gnorm\_k3, gumbel, logis, lst\_k3, cauchy, gev

#### Value

Plots QQ plots to the screen, for each of the models, and returns a data frame containing

- AIC scores (times -0.5), AIC weights
- WAIC1 scores, WAIC1 weights
- WAIC2 scores, WAIC2 weights
- logscores, logscore weights
- maximum likelihood and calibrating prior means
- maximum likelihood and calibrating prior standard deviations

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

742 ms\_predictors\_1tail

#### **Examples**

```
# because it's too slow for CRAN
set.seed(1)
nx=50
x=rnorm(nx)
print(ms_flat_2tail(x))
```

ms\_predictors\_1tail

Model Selection Among 5 Distributions with predictors from the fitdistcp Package

### **Description**

Applies model selection using AIC, WAIC1, WAIC2 and leave-one-out logscore to the input data x, t, for 5 one tailed models with predictors in the fitdistcp package.

The code is straightforward, and the point is to illustrate what is possible using the model selection outputs from the fitdistcp routines.

The input data may be automatically shifted so that the minimum value is positive.

For the Pareto, the data is so that the minimum value is slightly greater than 1.

## Usage

```
ms_predictors_1tail(x, t)
```

#### **Arguments**

x data vector

t predictor vector

### **Details**

The 5 models are: exp\_p1, pareto\_p1k2, lnorm\_p1, frechet\_p2k1, weibull\_p2.

## Value

Plots QQ plots to the screen, for each of the 5 models, and returns a data frame containing

- AIC scores, AIC weights
- WAIC1 scores, WAIC1 weights
- WAIC2 scores, WAIC2 weights
- logscores and logscore weights

ms\_predictors\_2tail 743

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### **Examples**

```
# because it's too slow for CRAN
set.seed(2)
nx=100
predictor=c(1:nx)/nx
x=rlnorm(nx,meanlog=predictor,sdlog=0.1)
print(ms_predictors_1tail(x,predictor))
```

ms\_predictors\_2tail

Model Selection Among 6 Distributions with predictors from the fitdistcp Package

### **Description**

Applies model selection using AIC, WAIC1, WAIC2 and leave-one-out logscore to the input data x, t, for 6 two tail models with predictors in the fitdistcp packages (although for the GEV, the logscore is NA for mathematical reasons).

The code is straightforward, and the point is to illustrate what is possible using the model selection outputs from the fitdistcp routines.

GEVD is temperamental in that it doesn't work if the shape parameter is extreme.

#### Usage

```
ms_predictors_2tail(x, t)
```

### **Arguments**

```
x data vector
```

t predictor vector

#### **Details**

The 11 models are: norm\_p1, gumbel\_p1, logis\_p1, lst\_k3\_p1, cauchy\_p1 and gev\_p1.

744 nopdfcdfmsg

### Value

Plots QQ plots to the screen, for each of the 6 models, and returns a data frame containing

- AIC scores, AIC weights
- WAIC1 scores, WAIC1 weights
- WAIC2 scores, WAIC2 weights
- logscores and logscore weights

# Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

## **Examples**

```
# because it's too slow for CRAN
set.seed(2)
nx=100
predictor=c(1:nx)/nx
x=rnorm(nx,mean=predictor,sd=1)
print(ms_predictors_2tail(x,predictor))
```

nopdfcdfmsg

Message to explain why GEV and GPD d\*\*\* and p\*\*\* routines don't return DMGS pdfs and cdfs

# Description

Message to explain why GEV and GPD d\*\*\* and p\*\*\* routines don't return DMGS pdfs and cdfs

### Usage

```
nopdfcdfmsg(yy, pp)
```

## Arguments

yy vector of samples
pp vector of probabilities

#### Value

String

norm\_cp

Normal Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qnorm_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    d1 = 0.01,
    fd2 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    rust = FALSE,
    nrust = 1e+05,
    unbiasedv = FALSE,
    debug = FALSE,
    aderivs = TRUE
)
```

```
dnorm\_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE, aderivs = TRUE)
pnorm\_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE, aderivs = TRUE)
tnorm\_cp(n, x, debug = FALSE)
```

### **Arguments**

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
unbiasedv	logical for whether to include unbiased variance results in norm
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

## Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- $\bullet\,$  ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.

- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

### **Details of the Model**

The normal distribution has probability density function

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)}$$

where x is the random variable and  $\mu, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

# **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

· Cauchy (cauchy),

- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
# example 1
x=fitdistcp::d30norm_example_data_v1
p=c(1:9)/10
q=qnorm_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qnorm_cp)",
main="Normal: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

norm\_dmgs\_cp

Normal Distribution Predictions Based on a Calibrating Prior, using DMGS (for testing only)

# Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

## Usage

```
qnorm_dmgs_cp(
 х,
 p = seq(0.1, 0.9, 0.1),
 d1 = 0.01,
 fd2 = 0.01,
 means = FALSE,
 waicscores = FALSE,
 logscores = FALSE,
 dmgs = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
rnorm_dmgs_cp(
 n,
 х,
 d1 = 0.01,
 fd2 = 0.01,
 mlcp = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
dnorm\_dmgs\_cp(x, y = x, d1 = 0.01, fd2 = 0.01, debug = FALSE, aderivs = TRUE)
pnorm_dmgs_cp(x, y = x, d1 = 0.01, fd2 = 0.01, debug = FALSE, aderivs = TRUE)
```

### **Arguments**

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter $$
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
debug	logical for turning on debug messages

aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The normal distribution has probability density function

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)}$$

where x is the random variable and  $\mu, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (1st\_p1k3),

756 norm\_dmgs\_loglik

- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

## **Examples**

```
#
# example 1
x=fitdistcp::d30norm_example_data_v1
p=c(1:9)/10
q=qnorm_dmgs_cp(x,p)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qnorm_dmgs_cp)",
main="Normal_DMGS: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
```

norm\_dmgs\_loglik

log-likelihood function

## **Description**

log-likelihood function

#### Usage

```
norm_dmgs_loglik(vv, x)
```

norm\_dmgs\_logscores 757

### Arguments

vv parameters

x a vector of training data values

#### Value

Scalar value.

### **Description**

Log scores for MLE and RHP predictions calculated using leave-one-out

#### Usage

```
norm_dmgs_logscores(logscores, x, d1 = 0.01, fd2 = 0.01)
```

### **Arguments**

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

d1 the delta used in the numerical derivatives with respect to the parameter

fd2 the fractional delta used in the numerical derivatives with respect to the param-

eter

### Value

Two scalars

MLE and RHP predictive means

### **Description**

MLE and RHP predictive means

```
norm_dmgs_means(means, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2)
```

758 norm\_dmgs\_mu1f

### **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

#### Value

Two scalars

norm\_dmgs\_mu1f

DMGS equation 3.3, mul term

#### **Description**

DMGS equation 3.3, mu1 term

# Usage

```
norm_dmgs_mu1f(alpha, v1, d1, v2, fd2)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter

d1 the delta used in the numerical derivatives with respect to the parameter

v2 second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

### Value

Matrix

norm\_dmgs\_mu2f 759

norm_dmgs_mu2f	DMGS equation 3.3, mu2 term	
----------------	-----------------------------	--

# Description

DMGS equation 3.3, mu2 term

# Usage

```
norm_dmgs_mu2f(alpha, v1, d1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

3d array

norm_dmgs_p1f	DMGS equation 3.3, p1 term	

# Description

DMGS equation 3.3, p1 term

# Usage

```
norm_dmgs_p1f(y, v1, d1, v2, fd2)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

760 norm\_dmgs\_waic

# Value

Matrix

norm\_dmgs\_p2f

DMGS equation 3.3, p2 term

# Description

```
DMGS equation 3.3, p2 term
```

# Usage

```
norm_dmgs_p2f(y, v1, d1, v2, fd2)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

# Value

3d array

norm\_dmgs\_waic

Waic

# Description

Waic

```
norm_dmgs_waic(
  waicscores,
  x,
  v1hat,
  d1,
  v2hat,
  fd2,
  lddi,
```

norm\_f1f 761

```
lddd,
lambdad,
aderivs
)
```

### **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values

v1hat first parameter

d1 the delta used in the numerical derivatives with respect to the parameter

v2hat second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

lddi inverse observed information matrix lddd third derivative of log-likelihood

lambdad derivative of the log prior

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

norm_f1f	DMGS equation 3.3, f1 term	

#### **Description**

DMGS equation 3.3, f1 term

# Usage

```
norm_f1f(y, v1, d1, v2, fd2)
```

### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

#### Value

Matrix

762 norm\_f2f

norm\_f1fa

The first derivative of the density

# Description

The first derivative of the density

# Usage

```
norm_f1fa(x, v1, v2)
```

# **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

#### Value

Vector

norm\_f2f

DMGS equation 3.3, f2 term

# Description

DMGS equation 3.3, f2 term

# Usage

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

the fractional delta used in the numerical derivatives with respect to the parar

eter

#### Value

3d array

norm\_f2fa 763

norm	f2fa
HOLIN	1210

The second derivative of the density

# Description

The second derivative of the density

### Usage

```
norm_f2fa(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

norm\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
norm_fd(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

# Value

Vector

764 norm\_gg

norm_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
norm_fdd(x, v1, v2)
```

# Arguments

Χ	a vector of training data values
---	----------------------------------

v1 first parameter v2 second parameter

#### Value

Matrix

norm_gg	Second derivative matrix of the expected per-observation log-	
	likelihood	

# Description

Second derivative matrix of the expected per-observation log-likelihood

# Usage

```
norm_gg(nx, v1, d1, v2, fd2)
```

# Arguments

nx	length of training data
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

#### Value

Square scalar matrix

norm\_gmn 765

norm_gmn	One component of the second derivative of the expected log-likelihood

# Description

One component of the second derivative of the expected log-likelihood

# Usage

```
norm_gmn(alpha, v1, d1, v2, fd2, mm, nn)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

# Value

Scalar value

norm_ldd	Second derivative matrix of the normalized log-likelihood	

# Description

Second derivative matrix of the normalized log-likelihood

# Usage

```
norm_1dd(x, v1, d1, v2, fd2)
```

# Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

766 norm\_lddd

#### Value

Square scalar matrix

norm\_ldda

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

#### Usage

```
norm_ldda(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

# Value

Matrix

norm\_lddd

Third derivative tensor of the normalized log-likelihood

#### **Description**

Third derivative tensor of the normalized log-likelihood

#### Usage

```
norm_1ddd(x, v1, d1, v2, fd2)
```

#### **Arguments**

X	a vector of training	data values
X	a vector or training	uata varues

v1 first parameter

d1 the delta used in the numerical derivatives with respect to the parameter

v2 second parameter

fd2 the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Cubic scalar array

norm\_lddda 767

norm_lddda
------------

# Description

The third derivative of the normalized log-likelihood

# Usage

```
norm_lddda(x, v1, v2)
```

# Arguments

v1 first parameter v2 second parameter

#### Value

3d array

norm_lmn	One component of the second derivative of the normalized log-
	likelihood

# Description

One component of the second derivative of the normalized log-likelihood One component of the second derivative of the normalized log-likelihood

# Usage

```
norm_lmn(x, v1, d1, v2, fd2, mm, nn)
norm_lmn(x, v1, d1, v2, fd2, mm, nn)
```

# Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

768 norm\_lmnp

# Value

Scalar value

norm_1mnp	One component of the second derivative of the normalized log-likelihood
-----------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

One component of the third derivative of the normalized log-likelihood

# Usage

```
norm_lmnp(x, v1, d1, v2, fd2, mm, nn, rr)
norm_lmnp(x, v1, d1, v2, fd2, mm, nn, rr)
```

# Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

# Value

Scalar value

norm\_logf 769

norm_	I O ort
1101 111	TOKI

Logf for RUST

# Description

Logf for RUST

# Usage

```
norm_logf(params, x)
```

# Arguments

params model parameters for calculating logf

x a vector of training data values

#### Value

Scalar value.

norm\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
norm_logfdd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameterv2 second parameter

#### Value

Matrix

770 norm\_logscores

norm_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
norm_logfddd(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

# Value

3d array

norm_logscores	Log scores for MLE and RHP predictions calculated using leave-one-
	out

#### **Description**

Log scores for MLE and RHP predictions calculated using leave-one-out

#### Usage

```
norm_logscores(logscores, x)
```

# Arguments

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

#### Value

Two scalars

norm\_ml\_params 771

norm\_ml\_params

Maximum likelihood estimator

# Description

Maximum likelihood estimator

# Usage

```
norm_ml_params(x)
```

# Arguments

Х

a vector of training data values

#### Value

Scalar value.

norm\_mu1fa

Minus the first derivative of the cdf, at alpha

# Description

Minus the first derivative of the cdf, at alpha

#### Usage

```
norm_mu1fa(alpha, v1, v2)
```

# **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

#### Value

Vector

772 norm\_p1fa

norm\_mu2fa

Minus the second derivative of the cdf, at alpha

# Description

Minus the second derivative of the cdf, at alpha

# Usage

```
norm_mu2fa(alpha, v1, v2)
```

# Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

#### Value

Matrix

norm\_p1fa

The first derivative of the cdf

# Description

The first derivative of the cdf

# Usage

```
norm_p1fa(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Vector

norm\_p1\_cp

Normal Distribution with a Predictor, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qnorm_p1_cp(
    x,
    t,
    t0 = NA,
    n0 = NA,
    p = seq(0.1, 0.9, 0.1),
    d1 = 0.01,
    d2 = 0.01,
    fd3 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    rust = FALSE,
    nrust = 1e+05,
    centering = TRUE,
```

```
debug = FALSE,
     aderivs = TRUE
   )
   rnorm_p1_cp(
     n,
     Х,
     t,
     t0 = NA,
     n0 = NA,
     rust = FALSE,
     mlcp = TRUE,
     debug = FALSE,
     aderivs = TRUE
   )
   dnorm_p1_cp(
     Х,
     t,
     t0 = NA,
     n0 = NA,
     y = x,
     rust = FALSE,
     nrust = 1000,
     centering = TRUE,
     debug = FALSE,
     aderivs = TRUE
   )
   pnorm_p1_cp(
     х,
      t,
     t0 = NA,
     n0 = NA,
     y = x,
     rust = FALSE,
     nrust = 1000,
     centering = TRUE,
     debug = FALSE,
     aderivs = TRUE
   )
   tnorm_p1_cp(n, x, t, debug = FALSE)
Arguments
                   a vector of training data values
   Χ
```

a vector of predictors, such that length(t)=length(x)

t

an index for the predictor (specify either t0 or n0 but not both)  p a vector of probabilities at which to generate predictive quantiles  d1 if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter  d2 if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter  fd3 if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the third parameter  means logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)  waicscores logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)  logscores logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)  rust logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)  nrust the number of posterior samples used in the RUST calculations  centering logical that indicates whether the predictor should be centered debug logical for turning on debug messages  aderivs (for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.  n the number of random samples required	t0	a single value of the predictor (specify either t0 or n0 but not both)
if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter  if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter  if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the third parameter  means logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)  waicscores logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)  logscores logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)  rust logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)  nrust the number of posterior samples used in the RUST calculations  centering logical that indicates whether the predictor should be centered logical for turning on debug messages  aderivs (for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.	n0	an index for the predictor (specify either t0 or n0 but not both)
first parameter  if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter  fd3 if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the third parameter  means logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)  waicscores logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)  logscores logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)  rust logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)  nrust the number of posterior samples used in the RUST calculations  centering logical that indicates whether the predictor should be centered debug logical for turning on debug messages  aderivs (for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.	р	a vector of probabilities at which to generate predictive quantiles
second parameter  fd3 if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the third parameter  means logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)  waicscores logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)  logscores logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)  rust logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)  nrust the number of posterior samples used in the RUST calculations  centering logical that indicates whether the predictor should be centered debug logical for turning on debug messages  aderivs (for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.  n the number of random samples required	d1	<del>-</del>
spect to the third parameter  logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)  waicscores logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)  logscores logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)  rust logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)  nrust the number of posterior samples used in the RUST calculations  centering logical that indicates whether the predictor should be centered debug logical for turning on debug messages  aderivs (for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.  the number of random samples required	d2	
estimates for the distribution means (longer runtime)  logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)  logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)  rust logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)  nrust the number of posterior samples used in the RUST calculations  centering logical that indicates whether the predictor should be centered debug logical for turning on debug messages  aderivs (for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.  n the number of random samples required	fd3	
logscores logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)  rust logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)  nrust the number of posterior samples used in the RUST calculations  centering logical that indicates whether the predictor should be centered  debug logical for turning on debug messages  aderivs (for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.  n the number of random samples required	means	
one-out estimates of the log-score (much longer runtime, non-EVT models only)  rust logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)  nrust the number of posterior samples used in the RUST calculations  centering logical that indicates whether the predictor should be centered debug logical for turning on debug messages  aderivs (for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.  n the number of random samples required	waicscores	
be run or not (longer run time)  nrust the number of posterior samples used in the RUST calculations  centering logical that indicates whether the predictor should be centered  debug logical for turning on debug messages  aderivs (for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.  n the number of random samples required	logscores	· · ·
centering logical that indicates whether the predictor should be centered debug logical for turning on debug messages aderivs (for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.  n the number of random samples required	rust	
debug logical for turning on debug messages  aderivs (for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.  n the number of random samples required	nrust	the number of posterior samples used in the RUST calculations
aderivs (for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.  n the number of random samples required	centering	logical that indicates whether the predictor should be centered
numerical). By default almost all models now use analytical derivatives.  n the number of random samples required	debug	logical for turning on debug messages
• •	aderivs	
mlcp logical that indicates whether maxlik and parameter uncertainty calculations	n	the number of random samples required
should be performed (turn off to speed up RUST)	mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
y a vector of values at which to calculate the density and distribution functions	У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- $\bullet \ \, {\tt ml\_quantiles:} \ \, {\tt quantiles:} \ \, {\tt calculated} \ \, {\tt using} \ \, {\tt maximum} \ \, {\tt likelihood.}$
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

• predictedparameter: the estimated value for parameter, as a function of the predictor.

• adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The normal distribution with a predictor has probability density function

$$f(x; a, b, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu(a,b))^2/(2\sigma^2)}$$

where x is the random variable,  $\mu = a + bt$  is the location parameter, modelled as a function of parameters a, b and predictor t, and  $\sigma > 0$  is the scale parameter.

The calibrating prior is given by the right Haar prior, which is

$$\pi(a,b,\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

· Cauchy (cauchy),

- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

780 norm\_p1\_f1f

#### **Examples**

```
#
# example 1
x=fitdistcp::d60norm_p1_example_data_v1_x
tt=fitdistcp::d60norm_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qnorm_p1_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qnorm_p1_cp)",
main="Normal w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

norm\_p1\_f1f

DMGS equation 2.1, f1 term

# Description

DMGS equation 2.1, f1 term

#### Usage

```
norm_p1_f1f(y, t0, v1, d1, v2, d2, v3, fd3)
```

#### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Matrix

norm\_p1\_f1fa 781

norm_	р1	_f1	fa
-------	----	-----	----

The first derivative of the density

# Description

The first derivative of the density

The first derivative of the density

# Usage

```
norm_p1_f1fa(x, t, v1, v2, v3)
norm_p1_f1fa(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

# Value

Vector

```
norm_p1_f2f
```

DMGS equation 2.1, f2 term

# Description

DMGS equation 2.1, f2 term

```
norm_p1_f2f(y, t0, v1, d1, v2, d2, v3, fd3)
```

782 norm\_p1\_f2fa

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

3d array

norm\_p1\_f2fa

The second derivative of the density

# Description

The second derivative of the density

The second derivative of the density

# Usage

```
norm_p1_f2fa(x, t, v1, v2, v3)
norm_p1_f2fa(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

# Value

Matrix

norm\_p1\_fd 783

norm_p1_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
norm_p1_fd(x, t, v1, v2, v3)
norm_p1_fd(x, t, v1, v2, v3)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Vector

norm_p1_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
norm_p1_fdd(x, t, v1, v2, v3)
norm_p1_fdd(x, t, v1, v2, v3)
```

784 norm\_p1\_ldd

# Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

# Value

Matrix

norm_p1_ldd	Second derivative matrix of the normalized log-likelihood

# Description

Second derivative matrix of the normalized log-likelihood

# Usage

```
norm_p1_ldd(x, t, v1, d1, v2, d2, v3, fd3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Square scalar matrix

norm\_p1\_ldda 785

norm\_p1\_ldda

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

The second derivative of the normalized log-likelihood

# Usage

```
norm_p1_ldda(x, t, v1, v2, v3)
norm_p1_ldda(x, t, v1, v2, v3)
```

# Arguments

Χ	a vector of training data values

t a vector or matrix of predictors

v1 first parameter

v2 second parameter

v3 third parameter

#### Value

Matrix

norm\_p1\_lddd

Third derivative tensor of the normalized log-likelihood

# Description

Third derivative tensor of the normalized log-likelihood

```
norm_p1_lddd(x, t, v1, d1, v2, d2, v3, fd3)
```

786 norm\_p1\_lddda

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

Cubic scalar array

norm_p1_lddda	The third derivative of the normalized log-likelihood
---------------	-------------------------------------------------------

# Description

The third derivative of the normalized log-likelihood The third derivative of the normalized log-likelihood

# Usage

```
norm_p1_lddda(x, t, v1, v2, v3)
norm_p1_lddda(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

# Value

3d array

norm\_p1\_lmn 787

norm_p1_lmn One component of the second derivative of the normalized log- likelihood	norm_p1_lmn	One component of the second derivative of the normalized log-likelihood
-----------------------------------------------------------------------------------------	-------------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
norm_p1_lmn(x, t, v1, d1, v2, d2, v3, fd3, mm, nn)
```

### Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

#### Value

Scalar value

norm_p1_lmnp	One component of the second derivative of the normalized log-likelihood

# Description

One component of the second derivative of the normalized log-likelihood

```
norm_p1_lmnp(x, t, v1, d1, v2, d2, v3, fd3, mm, nn, rr)
```

788 norm\_p1\_logf

# Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

# Value

Scalar value

# Description

Logf for RUST

# Usage

```
norm_p1_logf(params, x, t)
```

# Arguments

params	model parameters for calculating logf		
X	a vector of training data values		
t	a vector or matrix of predictors		

# Value

Scalar value.

norm\_p1\_logfdd 789

	norm_p1_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
--	----------------	------------------------------------------------------------------------------------------------------------------

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
norm_p1_logfdd(x, t, v1, v2, v3)
norm_p1_logfdd(x, t, v1, v2, v3)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

Deriv() by Anarew Clausen and Serguei Sokol	norm_p1_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
---------------------------------------------	-----------------	-----------------------------------------------------------------------------------------------------------------

#### **Description**

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
norm_p1_logfddd(x, t, v1, v2, v3)
norm_p1_logfddd(x, t, v1, v2, v3)
```

790 norm\_p1\_loglik

# Arguments

х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

# Value

3d array

norm\_p1\_loglik

Normal-with-p1 observed log-likelihood function

# Description

Normal-with-p1 observed log-likelihood function

# Usage

```
norm_p1_loglik(vv, x, t)
```

# Arguments

VV	parameters
X	a vector of training data values
t	a vector or matrix of predictors

# Value

Scalar value.

norm\_p1\_logscores 791

norm_p1_logscores	Log scores for MLE and RHP predictions calculated using leave-one-
	out

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

#### Usage

```
norm_p1_logscores(logscores, x, t, aderivs = TRUE)
```

# Arguments

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data valuest a vector or matrix of predictors

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

# Description

Maximum likelihood estimator

#### Usage

```
norm_p1_mlparams(x, t)
```

### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

### Value

Vector

792 norm\_p1\_mu2fa

10 0 10 m	<u>ہ</u> 1	m 1	£ ~
norm	וט	IIIU I	Та

Minus the first derivative of the cdf, at alpha

# Description

Minus the first derivative of the cdf, at alpha Minus the first derivative of the cdf, at alpha

# Usage

```
norm_p1_mu1fa(alpha, t, v1, v2, v3)
norm_p1_mu1fa(alpha, t, v1, v2, v3)
```

# Arguments

alpha	a vector of	of values of	of alpha	(one minus	probability)

t a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

# Value

Vector

norm\_p1\_mu2fa

Minus the second derivative of the cdf, at alpha

# Description

Minus the second derivative of the cdf, at alpha Minus the second derivative of the cdf, at alpha

```
norm_p1_mu2fa(alpha, t, v1, v2, v3)
norm_p1_mu2fa(alpha, t, v1, v2, v3)
```

norm\_p1\_p1fa 793

# Arguments

alpha a vector of values of alpha (one minus probability)
t a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

## Value

Matrix

norm\_p1\_p1fa

The first derivative of the cdf

# Description

The first derivative of the cdf

The first derivative of the cdf

## Usage

```
norm_p1_p1fa(x, t, v1, v2, v3)
norm_p1_p1fa(x, t, v1, v2, v3)
```

# Arguments

x a vector of training data valuest a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

# Value

Vector

794 norm\_p1\_pd

norm	n1	n2fa

The second derivative of the cdf

# Description

The second derivative of the cdf

The second derivative of the cdf

# Usage

```
norm_p1_p2fa(x, t, v1, v2, v3)
norm_p1_p2fa(x, t, v1, v2, v3)
```

## **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

norm_	p1	pc
	-r·	_~~

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## **Description**

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
norm_p1_pd(x, t, v1, v2, v3)
norm_p1_pd(x, t, v1, v2, v3)
```

norm\_p1\_pdd 795

# Arguments

_	
X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Vector

norm_p1_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
norm_p1_pdd(x, t, v1, v2, v3)
norm_p1_pdd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

## Value

796 norm\_p1\_waic

 ${\tt norm\_p1\_predictordata} \ \ \textit{Predicted Parameter and Generalized Residuals}$ 

# Description

Predicted Parameter and Generalized Residuals

# Usage

```
norm_p1\_predictordata(x, t, t0, params)
```

## **Arguments**

x a vector of training data valuest a vector or matrix of predictors

t0 a single value of the predictor (specify either t0 or n0 but not both)

params model parameters for calculating logf

### Value

Two vectors

norm\_p1\_waic

Waic

# **Description**

Waic

```
norm_p1_waic(
  waicscores,
  X,
  t,
  v1hat,
  d1,
  v2hat,
  d2,
  v3hat,
  fd3,
  aderivs = TRUE
)
```

norm\_p2fa 797

# Arguments

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data valuest a vector or matrix of predictors

v1hat first parameter

d1 the delta used in the numerical derivatives with respect to the parameter

v2hat second parameter

d2 the delta used in the numerical derivatives with respect to the parameter

v3hat third parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

aderivs logical for whether to use analytic derivatives (instead of numerical)

### Value

Two numeric values.

# Description

The second derivative of the cdf

## Usage

```
norm_p2fa(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

798 norm\_pdd

norm_pd First derivative of the cdf Created by Stephen Jewson using Deriv()  Andrew Clausen and Serguei Sokol	norm_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
---------------------------------------------------------------------------------------------------------------	---------	---------------------------------------------------------------------------------------------------------

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
norm_pd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

# Value

Vector

norm_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
norm_pdd(x, v1, v2)
```

# Arguments

v1 first parameter v2 second parameter

## Value

# Description

Method of moments estimator

## Usage

```
norm_unbiasedv_params(x)
```

# Arguments

x a vector of training data values

# Value

Vector

|--|

# Description

Waic

# Usage

```
norm_waic(waicscores, x, v1hat, d1, v2hat, fd2, aderivs)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

## Value

Two numeric values.

pareto\_k2\_cp

Pareto Distribution Predictions Based on a Calibrating Prior

### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qpareto_k2_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    kscale = 1,
    fd1 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE,
    aderivs = TRUE
)

rpareto_k2_cp(
    n,
```

```
х,
 kscale = 1,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
dpareto_k2_cp(
 х,
 y = x,
 kscale = 1,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
  aderivs = TRUE
)
ppareto_k2_cp(
 х,
 y = x,
 kscale = 1,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
  aderivs = TRUE
)
tpareto_k2_cp(n, x, kscale = 1, debug = FALSE)
```

a vector of training data values

# **Arguments** Х

p	a vector of probabilities at which to generate predictive quantiles
kscale	the known scale parameter
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations

debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.

• cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Pareto distribution has various forms. The form we are using has exceedance distribution function

 $S(x; \alpha) = \left(\frac{\sigma}{x}\right)^{\alpha}$ 

where  $x \ge \sigma$  is the random variable and  $\alpha > 0, \sigma > 0$  are the shape and scale parameters. We consider the scale parameter  $\sigma$  to be known (hence the k2 in the name).

The calibrating prior is given by the right Haar prior, which is

$$\pi(\alpha) \propto \frac{1}{\alpha}$$

as given in Jewson et al. (2025). Some others authors may refer to the shape and scale parameters as the scale and location parameters, respectively.

### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

 ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

### **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

## References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),

pareto\_k2\_f1f

- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

## Examples

```
#
# example 1
x=fitdistcp::d11pareto_k2_example_data_v1
p=c(1:9)/10
q=qpareto_k2_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles)
xmax=max(q$ml_quantiles,q$cp_quantiles)
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qpareto_k2_cp)",
main="Pareto: quantile estimates")
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

pareto\_k2\_f1f

DMGS equation 2.1, f1 term

### **Description**

```
DMGS equation 2.1, f1 term
```

```
pareto_k2_f1f(y, v1, fd1, kscale)
```

pareto\_k2\_f1fa 807

## **Arguments**

y a vector of values at which to calculate the density and distribution functions

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

kscale the known scale parameter

### Value

Matrix

pareto\_k2\_f1fa

The first derivative of the density

# Description

The first derivative of the density

## Usage

```
pareto_k2_f1fa(x, v1, kscale)
```

# Arguments

x a vector of training data values

v1 first parameter

kscale the known scale parameter

#### Value

Vector

pareto\_k2\_f2f

DMGS equation 2.1, f2 term

# Description

DMGS equation 2.1, f2 term

```
pareto_k2_f2f(y, v1, fd1, kscale)
```

808 pareto\_k2\_fd

### **Arguments**

y a vector of values at which to calculate the density and distribution functions

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

kscale the known scale parameter

## Value

3d array

### **Description**

The second derivative of the density

### Usage

```
pareto_k2_f2fa(x, v1, kscale)
```

## **Arguments**

x a vector of training data values

v1 first parameter

kscale the known scale parameter

#### Value

Matrix

pareto\_k2\_fd First derivative of the density Created by Stephen Jewson using De-

riv() by Andrew Clausen and Serguei Sokol

## **Description**

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
pareto_k2_fd(x, v1, v2)
```

pareto\_k2\_fdd 809

## **Arguments**

x a	vector of	f training	data va	lues
-----	-----------	------------	---------	------

v1 first parameterv2 second parameter

#### Value

Vector

pareto\_k2\_fdd

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## **Description**

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
pareto_k2_fdd(x, v1, v2)
```

## **Arguments**

x a vector of training data values

v1 first parameter

v2 second parameter

#### Value

Matrix

pareto\_k2\_l111

 $Third\ derivative\ of\ the\ normalized\ log-likelihood$ 

## **Description**

Third derivative of the normalized log-likelihood

```
pareto_k2_l111(x, v1, fd1, kscale)
```

pareto\_k2\_ldd

# Arguments

x a vector of training data values

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

kscale the known scale parameter

## Value

Scalar value

pareto\_k2\_ldd

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

# Usage

```
pareto_k2_ldd(x, v1, fd1, kscale)
```

## **Arguments**

x a vector of training data values

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

kscale the known scale parameter

## Value

Square scalar matrix

pareto\_k2\_ldda 811

pareto	レコ	า ฝ	٦.
nareto	K /	10	กล

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

# Usage

```
pareto_k2_ldda(x, v1, kscale)
```

## **Arguments**

x a vector of training data values

v1 first parameter

kscale the known scale parameter

## Value

Matrix

nareto	レン	1 444

Third derivative tensor of the log-likelihood

# Description

Third derivative tensor of the log-likelihood

## Usage

```
pareto_k2_lddd(x, v1, fd1, kscale)
```

## **Arguments**

x a vector of training data values

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

kscale the known scale parameter

### Value

Cubic scalar array

812 pareto\_k2\_logf

pareto\_k2\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

## Usage

```
pareto_k2_lddda(x, v1, kscale)
```

## **Arguments**

x a vector of training data values

v1 first parameter

kscale the known scale parameter

## Value

3d array

pareto\_k2\_logf

Logf for RUST

# Description

Logf for RUST

# Usage

```
pareto_k2_logf(params, x, kscale)
```

## **Arguments**

params model parameters for calculating logf
x a vector of training data values
kscale the known scale parameter

## Value

Scalar value.

pareto\_k2\_logfdd 813

pareto_k2_logfdd Second derivative of the log density Created by Stephen Jewson usi Deriv() by Andrew Clausen and Serguei Sokol	pareto_k2_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
------------------------------------------------------------------------------------------------------------------------------------	------------------	------------------------------------------------------------------------------------------------------------------

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
pareto_k2_logfdd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

# Value

Matrix

pareto_k2_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
pareto_k2_logfddd(x, v1, v2)
```

# **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

## Value

3d array

## **Description**

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
pareto_k2_logscores(logscores, x, kscale)
```

## **Arguments**

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values kscale the known scale parameter

### Value

Two scalars

 ${\tt pareto\_k2\_ml\_params} \qquad \textit{Maximum likelihood estimator}$ 

# Description

Maximum likelihood estimator

## Usage

```
pareto_k2_ml_params(x, kscale)
```

# Arguments

x a vector of training data valueskscale the known scale parameter

### Value

Scalar value.

pareto\_k2\_mu1fa 815

pareto\_k2\_mu1fa

Minus the first derivative of the cdf, at alpha

# Description

Minus the first derivative of the cdf, at alpha

## Usage

```
pareto_k2_mu1fa(alpha, v1, kscale)
```

## **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter

kscale the known scale parameter

### Value

Vector

pareto\_k2\_mu2fa

Minus the second derivative of the cdf, at alpha

# Description

Minus the second derivative of the cdf, at alpha

# Usage

```
pareto_k2_mu2fa(alpha, v1, kscale)
```

## **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter

kscale the known scale parameter

## Value

pareto\_k2\_p2fa

pareto\_k2\_p1fa

The first derivative of the cdf

# Description

The first derivative of the cdf

## Usage

```
pareto_k2_p1fa(x, v1, kscale)
```

## **Arguments**

x a vector of training data values

v1 first parameter

kscale the known scale parameter

# Value

Vector

pareto\_k2\_p2fa

The second derivative of the cdf

# Description

The second derivative of the cdf

# Usage

```
pareto_k2_p2fa(x, v1, kscale)
```

## **Arguments**

x a vector of training data values

v1 first parameter

kscale the known scale parameter

## Value

pareto_k2_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
pareto_k2_pd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

## Value

Vector

pareto_k2_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
pa. 000paa	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
pareto_k2_pdd(x, v1, v2)
```

# **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

### Value

## **Description**

Waic

### Usage

```
pareto_k2_waic(waicscores, x, v1hat, fd1, kscale, aderivs)
```

## **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values

v1hat first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

kscale the known scale parameter

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

pareto_p1k2_cp	Pareto Distribution with a Predictor, Predictions Based on a Calibrating Prior

## **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.

- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qpareto_p1k2_cp(
 х,
  t,
  t0 = NA,
 n0 = NA,
 p = seq(0.1, 0.9, 0.1),
 d1 = 0.01,
 d2 = 0.01,
 kscale = 1,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 predictordata = TRUE,
  centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
rpareto_p1k2_cp(
 n,
  х,
  t,
  t0 = NA,
 n0 = NA,
 d1 = 0.01,
 d2 = 0.01,
 kscale = 1,
  rust = FALSE,
 mlcp = TRUE,
 centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
```

```
)
dpareto_p1k2_cp(
 Х,
 t,
 t0 = NA,
 n0 = NA,
 y = x,
 d1 = 0.01,
 d2 = 0.01,
 kscale = 1,
  rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
ppareto_p1k2_cp(
 Х,
 t,
 t0 = NA,
 n0 = NA,
 y = x,
 d1 = 0.01,
 d2 = 0.01,
 kscale = 1,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
tpareto_p1k2_cp(n, x, t, d1 = 0.01, d2 = 0.01, kscale = 1, debug = FALSE)
```

# **Arguments**

Х	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
d2	if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter

kscale	the known scale parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
aderivs	
auerivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	
	numerical). By default almost all models now use analytical derivatives.

### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- $\bullet$  cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.

- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Pareto distribution with a predictor has various forms. The form we are using has exceedance distribution function

$$S(x; a, b) = \left(\frac{\sigma}{x}\right)^{\alpha(a, b)}$$

where  $x \ge \sigma$  is the random variable,  $\alpha = \exp(-a - bt)$  is the shape parameter, modelled as a function of parameters a, b, and  $\sigma$  is the scale parameter. We consider the scale parameter  $\sigma$  to be known (hence the k2 in the name).

The calibrating prior is given by the right Haar prior, which is

$$\pi(a,b) \propto 1$$

as given in Jewson et al. (2025). Note that others authors have referred to the shape and scale parameters as the scale and location parameters, respectively.

## **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

#### If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUF:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

### If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

# Examples

```
#
# example 1
x=fitdistcp::d56pareto_p1k2_example_data_v1_x
tt=fitdistcp::d56pareto_p1k2_example_data_v1_t
p=c(1:9)/10
n0=10
```

pareto\_p1k2\_f1f

```
q=qpareto_p1k2_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qpareto_p1k2_cp)",
main="Pareto w/ p2: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

pareto\_p1k2\_f1f

DMGS equation 2.1, f1 term

## **Description**

DMGS equation 2.1, f1 term

# Usage

```
pareto_p1k2_f1f(y, t0, v1, d1, v2, d2, kscale)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kscale	the known scale parameter

### Value

pareto\_p1k2\_f1fa 827

Dai Clo Dinz i i a lite ii si delivalive di lite delisi	pareto_p1k2_f1fa	The first derivative of the densit	ν
---------------------------------------------------------	------------------	------------------------------------	---

# Description

The first derivative of the density

# Usage

```
pareto_p1k2_f1fa(x, t, v1, v2, kscale)
```

# Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors

v1 first parameter v2 second parameter

kscale the known scale parameter

# Value

Vector

pareto\_p1k2\_f2f DMGS equation 2.1, f2 term

# Description

DMGS equation 2.1, f2 term

## Usage

```
pareto_p1k2_f2f(y, t0, v1, d1, v2, d2, kscale)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kscale	the known scale parameter

828 pareto\_p1k2\_fd

## Value

3d array

pareto\_p1k2\_f2fa

The second derivative of the density

## **Description**

The second derivative of the density

### Usage

```
pareto_p1k2_f2fa(x, t, v1, v2, kscale)
```

## **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

kscale the known scale parameter

## Value

Matrix

pareto\_p1k2\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## **Description**

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
pareto_p1k2_fd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

pareto\_p1k2\_fdd 829

### Value

Vector

pareto_p1k2_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
pareto_p1k2_fdd(x, t, v1, v2, v3)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

third parameter

v3

Value

Matrix

### Description

Second derivative matrix of the normalized log-likelihood

### Usage

```
pareto_p1k2_ldd(x, t, v1, d1, v2, d2, kscale)
```

pareto\_p1k2\_ldda

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kscale	the known scale parameter

### Value

Square scalar matrix

### Description

The second derivative of the normalized log-likelihood

### Usage

```
pareto_p1k2_ldda(x, t, v1, v2, kscale)
```

### Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
kscale	the known scale parameter

#### Value

Matrix

pareto\_p1k2\_lddd 831

pareto_	n1k2	PPP [
paretu_	_レートと_	_±uuu

Third derivative tensor of the normalized log-likelihood

#### **Description**

Third derivative tensor of the normalized log-likelihood

#### Usage

```
pareto_p1k2_lddd(x, t, v1, d1, v2, d2, kscale)
```

#### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

kscale the known scale parameter

#### Value

Cubic scalar array

pareto\_p1k2\_lddda

The third derivative of the normalized log-likelihood

### Description

The third derivative of the normalized log-likelihood

#### Usage

```
pareto_p1k2_lddda(x, t, v1, v2, kscale)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameter v2 second parameter

kscale the known scale parameter

832 pareto\_p1k2\_lmnp

### Value

3d array

pareto_p1k2_lmn	One component of the second derivative of the normalized log-likelihood

### Description

One component of the second derivative of the normalized log-likelihood

#### Usage

```
pareto_p1k2_lmn(x, t, v1, d1, v2, d2, kscale, mm, nn)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kscale	the known scale parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

### Value

Scalar value

pareto_p1k2_lmnp	One component of the second derivative of the normalized log-
	likelihood

### Description

One component of the second derivative of the normalized log-likelihood

### Usage

```
pareto_p1k2_lmnp(x, t, v1, d1, v2, d2, kscale, mm, nn, rr)
```

pareto\_p1k2\_logf 833

#### **Arguments**

x	a vector of training data values
t	a vector or matrix of predictors

v1 first parameter

d1 the delta used in the numerical derivatives with respect to the parameter

v2 second parameter

d2 the delta used in the numerical derivatives with respect to the parameter

kscale the known scale parameter

nn an index for which derivative to calculate
rr an index for which derivative to calculate

#### Value

Scalar value

pareto\_p1k2\_logf

Logf for RUST

### Description

Logf for RUST

### Usage

```
pareto_p1k2_logf(params, x, t, kscale)
```

### Arguments

params model parameters for calculating logf
x a vector of training data values
t a vector or matrix of predictors
kscale the known scale parameter

#### Value

Scalar value.

pareto_p1k2_logfdd	Second derivative of the log density Created by Stephen Jewson using
. –, – 0	Deriv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
pareto_p1k2_logfdd(x, t, v1, v2, v3)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
_	•

v2 second parameter v3 third parameter

#### Value

Matrix

pareto_p1k2_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

### Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
pareto_p1k2_logfddd(x, t, v1, v2, v3)
```

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

pareto\_p1k2\_loglik 835

#### Value

3d array

pareto\_p1k2\_loglik

observed log-likelihood function

### Description

observed log-likelihood function

#### Usage

```
pareto_p1k2_loglik(vv, x, t, kscale)
```

### Arguments

VV	parameters

x a vector of training data valuest a vector or matrix of predictorskscale the known scale parameter

#### Value

Scalar value.

### Description

Log scores for MLE and RHP predictions calculated using leave-one-out

### Usage

```
pareto_p1k2_logscores(logscores, x, t, d1, d2, kscale, aderivs, debug)
```

pareto\_p1k2\_means

### Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
X	a vector of training data values
t	a vector or matrix of predictors
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kscale	the known scale parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)
debug	debug flag

### Value

Two scalars

pareto\_p1k2\_means pareto\_k1 distribution: RHP mean

## Description

pareto\_k1 distribution: RHP mean

### Usage

```
pareto_p1k2_means(
  means,
  t0,
  ml_params,
  lddi,
  lddd,
  lambdad_rhp,
  nx,
  dim = 2,
  kscale
)
```

### Arguments

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

t0 a single value of the predictor (specify either t0 or n0 but not both)

ml\_params parameters

1ddi inverse observed information matrix

pareto\_p1k2\_mu1f 837

1ddd third derivative of log-likelihood 1ambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

kscale the known scale parameter

#### Value

Two scalars

pareto\_p1k2\_mu1f

DMGS equation 3.3, mul term

### **Description**

DMGS equation 3.3, mu1 term

### Usage

```
pareto_p1k2_mu1f(alpha, t0, v1, d1, v2, d2, kscale)
```

### Arguments

alpha a vector of values of alpha (one minus probability)

to a single value of the predictor (specify either to or no but not both)

v1 first parameter

d1 the delta used in the numerical derivatives with respect to the parameter

v2 second parameter

d2 the delta used in the numerical derivatives with respect to the parameter

kscale the known scale parameter

#### Value

Matrix

838 pareto\_p1k2\_mu2f

nareto n1k2 mu1fa	Minus the first derivative of the cdf at alpha
pareto_p1k2_mu1fa	Minus the first derivative of the cdf, at alpha

### Description

Minus the first derivative of the cdf, at alpha

### Usage

```
pareto_p1k2_mu1fa(alpha, t, v1, v2, kscale)
```

### Arguments

alpha a vector of values of alpha (one minus probability)

t a vector or matrix of predictors

v1 first parameter v2 second parameter

kscale the known scale parameter

### Value

Vector

pareto_p1k2_mu2f	DMGS equation 3.3, mu2 term
------------------	-----------------------------

### Description

DMGS equation 3.3, mu2 term

### Usage

```
pareto_p1k2_mu2f(alpha, t0, v1, d1, v2, d2, kscale)
```

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kscale	the known scale parameter

pareto\_p1k2\_mu2fa 839

#### Value

3d array

pareto\_p1k2\_mu2fa

Minus the second derivative of the cdf, at alpha

#### **Description**

Minus the second derivative of the cdf, at alpha

### Usage

```
pareto_p1k2_mu2fa(alpha, t, v1, v2, kscale)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

t a vector or matrix of predictors

v1 first parameter v2 second parameter

kscale the known scale parameter

#### Value

Matrix

pareto\_p1k2\_p1f

DMGS equation 2.1, p1 term

### Description

```
DMGS equation 2.1, p1 term
```

### Usage

```
pareto_p1k2_p1f(y, t0, v1, d1, v2, d2, kscale)
```

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kscale	the known scale parameter

pareto\_p1k2\_p2f

#### Value

Matrix

pareto\_p1k2\_p1fa

The first derivative of the cdf

### Description

The first derivative of the cdf

### Usage

```
pareto_p1k2_p1fa(x, t, v1, v2, kscale)
```

#### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

kscale the known scale parameter

#### Value

Vector

pareto\_p1k2\_p2f

DMGS equation 2.1, p2 term

### Description

DMGS equation 2.1, p2 term

### Usage

```
pareto_p1k2_p2f(y, t0, v1, d1, v2, d2, kscale)
```

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
kscale	the known scale parameter

pareto\_p1k2\_p2fa 841

#### Value

3d array

pareto\_p1k2\_p2fa

The second derivative of the cdf

#### **Description**

The second derivative of the cdf

#### Usage

```
pareto_p1k2_p2fa(x, t, v1, v2, kscale)
```

#### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

kscale the known scale parameter

#### Value

Matrix

pareto\_p1k2\_pd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### **Description**

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
pareto_p1k2_pd(x, t, v1, v2, v3)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameter v2 second parameter v3 third parameter

### Value

Vector

	of the cdf Created by Stephen Jewson using Deriv() n and Serguei Sokol
--	---------------------------------------------------------------------------

### Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
pareto_p1k2_pdd(x, t, v1, v2, v3)
```

### Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

```
pareto_p1k2_predictordata

Predicted Parameter and Generalized Residuals
```

### Description

Predicted Parameter and Generalized Residuals

### Usage

```
pareto_p1k2_predictordata(predictordata, x, t, t0, params, kscale)
```

pareto\_p1k2\_waic 843

#### **Arguments**

predictordata logical that indicates whether to calculate and return predictordata

x a vector of training data valuest a vector or matrix of predictors

to a single value of the predictor (specify either to or no but not both)

params model parameters for calculating logf

kscale the known scale parameter

#### Value

Two vectors

pareto\_p1k2\_waic

Waic

### Description

Waic

### Usage

```
pareto_p1k2_waic(
  waicscores,
  x,
  t,
  v1hat,
  d1,
  v2hat,
  d2,
  kscale,
  lddi,
  lddd,
  lambdad
)
```

### Arguments

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data valuest a vector or matrix of predictors

v1hat first parameter

d1 the delta used in the numerical derivatives with respect to the parameter

v2hat second parameter

pcauchy\_p1

d2 the delta used in the numerical derivatives with respect to the parameter

kscale the known scale parameter

lddi inverse observed information matrixlddd third derivative of log-likelihood

lambdad derivative of the log prior

#### Value

Two numeric values.

pcauchy_p1	Cauchy-with-n1	distribution function
peaderry_pr	Canchy with pr	distribution junction

### Description

Cauchy-with-p1 distribution function

### Usage

```
pcauchy_p1(x, t0, ymn, slope, scale)
```

### **Arguments**

x a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor scale the scale parameter of the distribution

#### Value

pexp\_p1 845

		- 4
pex	n	ni
DCV	$\nu_{-}$	$\nu$ 1

Exponential-with-p1 distribution function

#### **Description**

Exponential-with-p1 distribution function

#### Usage

```
pexp_p1(x, t0, ymn, slope)
```

#### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

#### Value

Vector

pfrechet\_p2k1

Frechet\_k1-with-p2 distribution function

#### **Description**

Frechet\_k1-with-p2 distribution function

### Usage

```
pfrechet_p2k1(x, t0, ymn, slope, lambda, kloc)
```

#### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor lambda the lambda parameter of the distribution

kloc the known location parameter

#### Value

pgev_p1	GEVD-with-p1:	Distribution function

### Description

GEVD-with-p1: Distribution function

### Usage

```
pgev_p1(y, t0, ymn, slope, sigma, xi)
```

### Arguments

a vector of values at which to calculate the density and distribution functions
a single value of the predictor (specify either t0 or n0 but not both)
the location parameter of the function of the predictor
the slope of the function of the predictor
the sigma parameter of the distribution
the shape parameter of the distribution

#### Value

Vector

pgev_p12 GEVD-with-p1: Distribution function
----------------------------------------------

### Description

GEVD-with-p1: Distribution function

### Usage

```
pgev_p12(y, t1, t2, ymn, slope, sigma1, sigma2, xi)
```

У		a vector of values at which to calculate the density and distribution functions
t1		a vector of predictors for the mean
t2		a vector of predictors for the sd
ymn		the location parameter of the function of the predictor
slo	ре	the slope of the function of the predictor
sig	ma1	first coefficient for the sigma parameter of the distribution
sig	ma2	second coefficient for the sigma parameter of the distribution
хi		the shape parameter of the distribution

pgev\_p123 847

### Value

Vector

pgev\_p123

GEVD-with-p1: Distribution function

### Description

GEVD-with-p1: Distribution function

## Usage

```
pgev_p123(y, t1, t2, t3, ymn, slope, sigma1, sigma2, xi1, xi2)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma1	first coefficient for the sigma parameter of the distribution
sigma2	second coefficient for the sigma parameter of the distribution
xi1	first coefficient for the shape parameter of the distribution
xi2	second coefficient for the shape parameter of the distribution

### Value

pgumbel\_p1

n	ge	٠,	n	1	1/2
ν	ರ್ಷ	٧_	٠P	ı	ĸο

GEV-with-known-shape-with-p1 distribution function

#### **Description**

GEV-with-known-shape-with-p1 distribution function

#### Usage

```
pgev_p1k3(x, t0, ymn, slope, sigma, kshape)
```

#### Arguments

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

kshape the known shape parameter

#### Value

Vector

	ا مامس	11
pg	umbe:	T_D I

Gumbel-with-p1 distribution function

#### **Description**

Gumbel-with-p1 distribution function

#### Usage

```
pgumbel_p1(x, t0, ymn, slope, sigma)
```

#### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

### Value

plnorm\_p1 849

		-
nl	norm	ni
NΤ	.norm_	_レ1

Normal-with-p1 distribution function

#### **Description**

Normal-with-p1 distribution function

#### Usage

```
plnorm_p1(x, t0, ymn, slope, sigma)
```

#### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

#### Value

Vector

-		_
nΙ	ogis	n1

Logistic-with-p1 distribution function

#### **Description**

Logistic-with-p1 distribution function

#### Usage

```
plogis_p1(x, t0, ymn, slope, scale)
```

#### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor scale the scale parameter of the distribution

#### Value

850 pnorm\_p1

		- 4	1 ~
n	101	nı	$\nu$
$\nu$	lst.	_ レ י	NJ

LST-with-p1 distribution function

### Description

LST-with-p1 distribution function

#### Usage

```
plst_p1k3(x, t0, ymn, slope, sigma, kdf)
```

#### **Arguments**

Χ	a vector of training data values
---	----------------------------------

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution kdf the known degrees of freedom parameter

#### Value

Vector

	_
pnorm_	n1
prior iii_	$\mathbf{p}$

Normal-with-p1 distribution function

#### **Description**

Normal-with-p1 distribution function

#### Usage

```
pnorm_p1(x, t0, ymn, slope, sigma)
```

#### **Arguments**

X	a vector of training data values
^	a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

### Value

pnorm\_p1\_formula 851

pnorm_	n1	formu	ıla

Linear regression formula, densities

#### **Description**

Linear regression formula, densities

### Usage

```
pnorm_p1_formula(y, ta, ta0, nx, muhat0, v3hat)
```

#### **Arguments**

y a vector of values at which to calculate the density and distribution functions

ta predictor residuals

ta0 predictor residual at the point being predicted

nx length of training data

muhat at the point being predicted

v3hat third parameter

#### Value

Vector

ppareto\_p1k2

pareto\_k1-with-p2 distribution function

#### **Description**

```
pareto_k1-with-p2 distribution function
```

#### Usage

```
ppareto_p1k2(x, t0, ymn, slope, kscale)
```

#### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

kscale the known scale parameter

### Value

pweibull\_p2

punif\_formula

Predictive CDFs

### Description

Predictive CDFs

### Usage

```
punif_formula(x, y)
```

#### **Arguments**

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

#### Value

Two vectors

pweibull\_p2

Weibull-with-p1 distribution function

#### **Description**

Weibull-with-p1 distribution function

### Usage

```
pweibull_p2(x, t0, shape, ymn, slope)
```

#### **Arguments**

x a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

shape the shape parameter of the distribution

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

#### Value

qcauchy\_p1 853

		_
aca	auchy	n I

Cauchy-with-p1 quantile function

### **Description**

Cauchy-with-p1 quantile function

### Usage

```
qcauchy_p1(p, t0, ymn, slope, scale)
```

### Arguments

p a vector of probabilities at which to generate predictive quantiles t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor scale the scale parameter of the distribution

#### Value

Vector

qexp\_p1

-with-p1 quantile function

#### **Description**

-with-p1 quantile function

#### Usage

```
qexp_p1(p, t0, ymn, slope)
```

### Arguments

p a vector of probabilities at which to generate predictive quantiles
 t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

#### Value

854 qgamma\_k1\_ppm

qfrechet_	~ <b>⊃</b> 1∠1
arrechei	1)/K I

Frechet\_k1-with-p2 quantile function

### Description

Frechet\_k1-with-p2 quantile function

#### Usage

```
qfrechet_p2k1(p, t0, ymn, slope, lambda, kloc)
```

#### **Arguments**

p a vector of probabilities at which to generate predictive quantiles
 t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor lambda the lambda parameter of the distribution

kloc the known location parameter

#### Value

Vector

qgamma\_k1\_ppm

Temporary dummy for one of the cp models

### Description

Temporary dummy for one of the cp models

### Usage

```
qgamma_k1_ppm(x, p)
```

#### **Arguments**

x a vector of training data values

p a vector of probabilities at which to generate predictive quantiles

qgamma\_k1\_ppm 855

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

856 qgamma\_ppm

qgamma\_ppm

Temporary dummy for one of the ppm models

#### **Description**

Temporary dummy for one of the ppm models

#### Usage

```
qgamma_ppm(x, p)
```

### **Arguments**

- x a vector of training data values
- p a vector of probabilities at which to generate predictive quantiles

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.

qgev\_k12\_ppm 857

• cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

qgev\_k12\_ppm

Temporary dummy for one of the ppm models

#### Description

Temporary dummy for one of the ppm models

#### Usage

```
qgev_k12_ppm(x, p)
```

### Arguments

- x a vector of training data values
- p a vector of probabilities at which to generate predictive quantiles

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

858 qgev\_mpd\_ppm

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

qgev\_mpd\_ppm

Temporary dummy for one of the ppm models

#### **Description**

Temporary dummy for one of the ppm models

### Usage

```
qgev_mpd_ppm(x, p)
```

#### **Arguments**

x a vector of training data values

p a vector of probabilities at which to generate predictive quantiles

qgev\_mpd\_ppm 859

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

860 *qgev\_p12* 

qgev_p1	GEVD-with-p1: Quantile function
---------	---------------------------------

### Description

GEVD-with-p1: Quantile function

#### Usage

```
qgev_p1(p, t0, ymn, slope, sigma, xi)
```

### Arguments

p	a vector of probabilities at which to generate predictive quantiles
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma	the sigma parameter of the distribution

the shape parameter of the distribution

# хi

Vector

Value

qgev_p12	GEVD-with-p1: Quantile function
----------	---------------------------------

### Description

```
GEVD-with-p1: Quantile function
```

### Usage

```
qgev_p12(p, t1, t2, ymn, slope, sigma1, sigma2, xi)
```

p	a vector of probabilities at which to generate predictive quantiles
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma1	first coefficient for the sigma parameter of the distribution
sigma2	second coefficient for the sigma parameter of the distribution
xi	the shape parameter of the distribution

*qgev\_p123* 861

### Value

Vector

qgev\_p123

GEVD-with-p1: Quantile function

### Description

GEVD-with-p1: Quantile function

## Usage

```
qgev_p123(p, t1, t2, t3, ymn, slope, sigma1, sigma2, xi1, xi2)
```

### Arguments

р	a vector of probabilities at which to generate predictive quantiles
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma1	first coefficient for the sigma parameter of the distribution
sigma2	second coefficient for the sigma parameter of the distribution
xi1	first coefficient for the shape parameter of the distribution
xi2	second coefficient for the shape parameter of the distribution

### Value

862 qgev\_p1\_ppm

qgev_p1k3 GEV-with-known-shape-with-p1 quantile function	
----------------------------------------------------------	--

### Description

GEV-with-known-shape-with-p1 quantile function

### Usage

```
qgev_p1k3(p, t0, ymn, slope, sigma, kshape)
```

### Arguments

p	a vector of probabilities at which to generate predictive quantiles
t0	a single value of the predictor (specify either $t0$ or $n0$ but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

kshape the known shape parameter

#### Value

Vector

qgev_p1_ppm	Temporary dummy for one of the ppm models

### Description

Temporary dummy for one of the ppm models

### Usage

```
qgev_p1_pm(x, t, n0, p)
```

X	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles

qgev\_p1\_ppm 863

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

864 *qgev\_ppm* 

qgev\_ppm

Temporary dummy for one of the ppm models

#### **Description**

Temporary dummy for one of the ppm models

#### Usage

```
qgev_ppm(x, p)
```

### **Arguments**

- x a vector of training data values
- p a vector of probabilities at which to generate predictive quantiles

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.

qgpd\_k1\_ppm 865

• cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

qgpd\_k1\_ppm

Temporary dummy for one of the ppm models

#### Description

Temporary dummy for one of the ppm models

#### Usage

```
qgpd_k1_ppm(x, p)
```

## **Arguments**

- x a vector of training data values
- p a vector of probabilities at which to generate predictive quantiles

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

866 qgumbel\_p1

For models with predictors, q\*\*\* additionally returns:

• predictedparameter: the estimated value for parameter, as a function of the predictor.

• adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

qgumbel\_p1

Gumbel-with-p1 quantile function

#### **Description**

Gumbel-with-p1 quantile function

## Usage

```
qgumbel_p1(p, t0, ymn, slope, sigma)
```

qlnorm\_p1 867

## **Arguments**

p a vector of probabilities at which to generate predictive quantiles to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

#### Value

Vector

## **Description**

Normal-with-p1 quantile function

#### Usage

```
qlnorm_p1(p, t0, ymn, slope, sigma)
```

#### **Arguments**

p a vector of probabilities at which to generate predictive quantiles t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

#### Value

868 qlst\_p1k3

$\alpha$	.ogis	മി
u.	OET:	וט כ

Logistic-with-p1 quantile function

# Description

Logistic-with-p1 quantile function

## Usage

```
qlogis_p1(p, t0, ymn, slope, scale)
```

## Arguments

p	a vector of probabilities at which to generate predictive quantiles
t0	a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor scale the scale parameter of the distribution

#### Value

Vector

	_	
alst	ומ	ĸЗ

LST-with-p1 quantile function

## Description

LST-with-p1 quantile function

## Usage

```
qlst_p1k3(p, t0, ymn, slope, sigma, kdf)
```

## Arguments

p	a vector of probabilities at which to generate predictive quantiles
t0	a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution kdf the known degrees of freedom parameter

## Value

qnorm\_p1 869

qnorm_	n1
quion iii_	_P '

Normal-with-p1 quantile function

#### **Description**

Normal-with-p1 quantile function

## Usage

```
qnorm_p1(p, t0, ymn, slope, sigma)
```

#### Arguments

p a vector of probabilities at which to generate predictive quantiles
 t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

#### Value

Vector

qnorm\_p1\_formula

Linear regression formula, quantiles

## Description

Linear regression formula, quantiles

#### Usage

```
qnorm_p1_formula(alpha, ta, ta0, nx, muhat0, v3hat)
```

#### Arguments

alpha a vector of values of alpha (one minus probability)

ta predictor residuals

ta0 predictor residual at the point being predicted

nx length of training data

muhat at the point being predicted

v3hat third parameter

## Value

870 qntt\_ppm

qntt\_ppm

Temporary dummy for one of the ppm models

#### **Description**

Temporary dummy for one of the ppm models

#### Usage

```
qntt_ppm(x, p)
```

#### **Arguments**

- x a vector of training data values
- p a vector of probabilities at which to generate predictive quantiles

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.

qpareto\_p1k2 871

• cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

qpareto\_p1k2

pareto\_k1-with-p2 quantile function

## **Description**

pareto\_k1-with-p2 quantile function

#### Usage

```
qpareto_p1k2(p, t0, ymn, slope, kscale)
```

## **Arguments**

p	a vector of probabilities at which to generate predictive quantiles
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
kscale	the known scale parameter

#### Value

gweibull\_p2

qunif\_formula

Predictive Quantiles

## Description

**Predictive Quantiles** 

## Usage

```
qunif_formula(x, p)
```

#### **Arguments**

x a vector of training data values

p a vector of probabilities at which to generate predictive quantiles

#### Value

Two vectors

qweibull\_p2

Weibull-with-p1 quantile function

#### **Description**

Weibull-with-p1 quantile function

## Usage

```
qweibull_p2(p, t0, shape, ymn, slope)
```

#### **Arguments**

p a vector of probabilities at which to generate predictive quantiles
 t0 a single value of the predictor (specify either t0 or n0 but not both)

shape the shape parameter of the distribution

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

#### Value

reltest

Evaluation of Reliability for Models in the fitdistcp Package

#### **Description**

Uses simulations to evaluate the reliability of the predictive quantiles produced by the q\*\*\*\*\_cp routines in the fitdistcp package.

## Usage

```
reltest(
 model = "exp",
 ntrials = 1000,
 nrepeats = 3,
 nx = 20,
  params = c(1),
  alpha = seq(0.005, 0.995, 0.005),
 plotflag = TRUE,
  verbose = TRUE,
  dmgs = TRUE,
  debug = FALSE,
  aderivs = TRUE,
  unbiasedv = FALSE,
  pwm = FALSE,
 minxi = -10,
 maxxi = 10
)
```

#### **Arguments**

```
mode1
                  which distribution to test. Possibles values are "exp", "pareto_k1", "halfnorm",
                  "unif", "norm", "norm_dmgs", "gnorm_k3", "lnorm", "lnorm_dmgs", "logis",
                  "lst_k3", "cauchy", "gumbel", "frechet_k1", "weibull", "gev_k3", "exp_p1",
                  "pareto_p1k3", "norm_p1", "lnorm_p1", "logis_p1", "lst_p1k4", "cauchy_p1",
                  "gumbel_p1", "frechet_p2k1", "weibull_p2", "gev_p1k4", "norm_p12", "lst_p12k5",
                  "gamma", "invgamma", "invgauss", "gev", "gpd_k1", "gev_p1". "gev_p12".
                  "gev_p123".
ntrials
                  the number of trials to run. 5000 typically gives good results.
nrepeats
                  the number of entire repeats of the test to run, to check for convergence. 3 is a
                  good choice.
nx
                  the length of the training data to use.
                  values for the parameters for the specified distribution
params
alpha
                  the exceedance probability values at which to test
plotflag
                  logical to turn the plotting on and off
verbose
                  logical to turn loop counting on and off
```

dmgs logical to turn DMGS calculations on and off (to optimize speed for maxlik only

calculations)

debug logical for turning debug messages on and off

aderivs logical for whether to use analytic derivatives (instead of numerical)

unbiasedv logical for whether to use the unbiased variance instead of maxlik (for the nor-

mal)

pwm logical for whether to use PWM instead of maxlik (for the GEV)

minxi minimum value for EVT shape parameter maxxi maximum value for EVT shape parameter

#### **Details**

The maximum likelihood quantiles (plotted in blue) do not give good reliability. They typically underestimate the tails (see panel (f)).

For "exp", "pareto\_k1", "unif", "norm", "lnorm", "norm\_p1" and "lnorm\_p1", the calibrating prior quantiles are calculated using the right Haar prior and an exact solution for the Bayesian prediction integral. They will converge towards exact reliability with a large enough number of trials, for any sample size.

For "halfnorm", "norm\_dmgs", "lnorm\_dmgs", "gnorm\_k3", "logis", "lst\_k3", "cauchy", "gumbel", "frechet\_k1", "weibull", "gev\_k3", "exp\_p1", "pareto\_p1k3", "gumbel\_p1", "logis\_p1" and "lst\_p1k4" "cauchy\_p1", "gumbel\_p1", "frechet\_p2k1", "weibull\_p2", "gev\_p1k4", "norm\_p12", "lst\_p12k5" the calibrating prior quantiles are calculated using the right Haar prior, with the DMGS asymptotic solution for the Bayesian prediction integral. They will converge towards good reliability with a large enough number of trials, with the only deviation from exact reliability being due to the neglect of higher order terms in the asymptotic expansion. They will converge towards exact reliability with a large enough number of trials and a large enough sample size.

For "gamma", "invgamma", "invgauss", "gev", "gpd\_k1" and "gev\_p1", "gev\_p12", "gev\_p123", the calibrating prior quantiles are calculated using the "fitdistcp" recommended calibrating priors, with the DMGS asymptotic solution for the Bayesian prediction integral. The chosen priors give reasonably good reliability with a large enough number of trials, and for large sample sizes, but may give poor reliability for small sample sizes (e.g., n<20).

#### Value

A plot showing 9 different reliability checks, and a list containing various outputs, including the probabilities shown in the plot.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),

- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
set.seed(1)
# example 1
# -runs the default settings, which test reliability for the exponential distribution
reltest()
```

reltest2

Evaluation of Reliability for Certain Additional Models in the fitdistcp Package

#### **Description**

This routine is mainly for reproducing certain results in Jewson et al. (2025), and not of general interest.

It uses simulations to evaluate the reliability of the predictive quantiles produced by the qgev\_cp, ggpd\_cp and qgev\_p1\_cp routines in the fitdistcp package. For each model, results for 5 models are calculated. This is to illustrate that the calibrating prior predictions dominate the ml, flat, crhp\_ml and jp predictions, in terms of reliability.

#### Usage

```
reltest2(
  model = "gev",
  ntrials = 100,
  nrepeats = 3,
  nx = 50,
  params = c(0, 1, 0),
  alpha = seq(0.005, 0.995, 0.005),
  plotflag = TRUE,
  verbose = TRUE
)
```

#### **Arguments**

model which distribution to test. Possibles values are "gev", "gpd\_k1", "gev\_p1".

ntrials the number of trials to run. 5000 typically gives good results.

nrepeats the number of entire repeats of the test to run, to check for convergence. 3 is a

good choice.

nx the length of the training data.

params values for the parameters for the specified distribution

alpha the alpha values at which to test

plotflag logical to turn the plotting on and off verbose logical to turn loop counting on and off

#### **Details**

The maximum likelihood quantiles (plotted in blue) do not give good reliability. They typically underestimate the tails (see panel (f)).

The cp predictive quantiles generally give reasonably good reliability, especially for sample sizes of ~100. The other predictions generally give poor reliability.

#### Value

A plot showing 9 different reliability checks, and a list containing various outputs, including the probabilities shown in the plot.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

• Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),

- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

reltest2\_cases 879

## **Examples**

```
set.seed(1)
# example 1
# -runs the default settings, which test reliability for the GEV distribution
reltest2(nrepeats=1)
```

reltest2\_cases

Cases

#### **Description**

Cases

## Usage

```
reltest2_cases(model = "gev", nx = 50, params)
```

## Arguments

model which distribution to test. Possibles values are "gev", "gpd\_k1", "gev\_pred1".

nx length of training data params model parameters

#### Value

Two integers

reltest2\_makeep

Cases

## Description

Cases

## Usage

```
reltest2_makeep(model, pred1, tt0, params)
```

## **Arguments**

model which distribution to test. Possibles values are "gev", "gpd\_k1", "gev\_pred1".

pred1 quantile predictions tt0 value of predictor vector

params model parameters

880 reltest2\_plot

#### Value

Vector

reltest2\_plot

Plotting routine for reltest2

#### **Description**

Plots 9 diagnostics related to predictive probability matching.

## Usage

```
reltest2_plot(
  model,
  ntrials,
  nrepeats,
  nx,
  params,
  nmethods,
  alpha,
  freqexceeded,
  case
)
```

#### **Arguments**

model which distribution to test. Possibles values are "gev", "gpd", "gev\_p1".

ntrials the number of trials o run. 5000 typically gives good results.

nrepeats the number of entire repeats of the test to run, to check for convergence

nx the length of the training data.

params values for the parameters for the specified distribution

nmethods the number of methods being tested alpha the values of alpha being tested

freqexceeded the exceedance counts

there are 3 cases (must be set to case=1 except for my testing)

#### Value

Plots the results of reliability testing

reltest2\_predict 881

|--|

# Description

Make prediction from one model

## Usage

```
reltest2_predict(model = "gev", xx, tt, n0, pp, params, case, nmethods)
```

# Arguments

model	which distribution to test. Possibles values are "exp", "pareto_k1", "halfnorm", "norm", "lnorm", "gumbel", "frechet_k1", "weibull", "gev_k3", "logis", "lst_k3", "cauchy", "norm_p1", "lnorm_p1", "logis_p1", "lst_k3p1", "gumbel_p1", "norm_p12", "gev", "gpd", "gev_p1".
xx	training data
tt	predictor vector
n0	index for predictor vector
рр	probabilities to predict
params	model parameters
case	the case number: different models have different lists of methods
nmethods	the number of methods: different models have different numbers of methods

#### Value

Vector

reltest2\_simulate Random training data from one model

# Description

Random training data from one model

# Usage

```
reltest2_simulate(model = "gev", nx = 50, tt, params)
```

882 reltest\_makeep

## Arguments

model which distribution to test. Possibles values are "gev", "gpd\_k1", "gev\_pred1".

nx the length of the training data.

tt the predictor

params values for the parameters for the specified distribution

#### Value

Vector

reltest\_makeep Calculate EP from one model

# Description

Calculate EP from one model

## Usage

```
reltest_makeep(model, pred1, tt0, tt10, tt20, tt30, params)
```

## Arguments

model	which distribution to test. Possibles values are "exp", "pareto_k2", "halfnorm", "unif", "norm", "norm_dmgs", "gnorm_k3", "lnorm", "lnorm_dmgs", "logis", "lst_k3", "cauchy", "gumbel", "frechet_k1", "weibull", "gev_k3", "exp_p1", "logis",
	<pre>"pareto_p1k2", "norm_p1", "lnorm_p1", "logis_p1", "lst_p1k3", "cauchy_p1",    "gumbel_p1", "frechet_p2k1", "weibull_p2", "gev_p1k3", "norm_p12", "lst_p12k3",    "gamma", "invgamma", "invgauss", "gev", "gpd_k1", "gev_p1". "gev_p12".    "gev_p123".</pre>
pred1	quantile predictions
tt0	value of the predictor
tt10	value of predictor 1
tt20	value of predictor 2
tt30	value of predictor 3
params	the model parameters

## Value

reltest\_makemaxep 883

 $reltest\_makemaxep$ 

Calculate MaxEP from one model

# Description

Calculate MaxEP from one model

# Usage

```
reltest_makemaxep(model, ml_max, tt0, tt10, tt20, tt30, params)
```

# Arguments

model	which distribution to test. Possibles values are "gev", "gpd_k1", "gev_p1". "gev_p12". "gev_p123".
ml_max	predicted max value
tt0	value of the predictor
tt10	value of predictor 1
tt20	value of predictor 2
tt30	value of predictor 3
params	the model parameters

#### Value

Vector

reltest\_predict

Make prediction from one model

# Description

Make prediction from one model

## Usage

```
reltest_predict(
  model,
  xx,
  tt,
  tt1,
  tt2,
  tt3,
  n0,
  n10,
```

884 reltest\_predict

```
n20,
n30,
pp,
params,
dmgs = TRUE,
debug = FALSE,
aderivs = TRUE,
unbiasedv = FALSE,
pwm = FALSE,
minxi = -10,
maxxi = 10
```

#### **Arguments**

```
model
                  which distribution to test. Possibles values are "exp", "pareto_k2", "halfnorm",
                  "unif", "norm", "norm_dmgs", "gnorm_k3", "lnorm", "lnorm_dmgs", "logis",
                  "lst_k3", "cauchy", "gumbel", "frechet_k1", "weibull", "gev_k3", "exp_p1",
                  "pareto_p1k2", "norm_p1", "lnorm_p1", "logis_p1", "lst_p1k3", "cauchy_p1",
                  "gumbel_p1", "frechet_p2k1", "weibull_p2", "exp_p1k4", "norm_p12", "lst_p12k3",
                  "gamma", "invgamma", "invgauss", "gev", "gpd_k1", "gev_p1". "gev_p12".
                  "gev_p123".
                  training data
xx
                  predictor vector
tt
tt1
                  predictor vector 1
tt2
                  predictor vector 2
                  predictor vector 3
tt3
                  index for predictor vector
n0
n10
                  index for predictor vector 1
                  index for predictor vector 2
n20
n30
                  index for predictor vector 2
                  probabilites at which to make quantile predictions
pp
                  model parameters
params
                  flag for whether to run dmgs calculations or not
dmgs
                  flag for turning debug messages on
debug
                  a logical for whether to use analytic derivatives (instead of numerical)
aderivs
unbiasedv
                  a logical for whether to use the unbiased variance instead of maxlik (for the
                  a logical for whether to use PWM instead of maxlik (for the GEV)
pwm
                  minimum value for EVT shape parameter
minxi
maxxi
                  maximum value for EVT shape parameter
```

#### Value

Two vectors

reltest\_simulate 885

reltest\_simulate

Random training data from one model

#### **Description**

Random training data from one model

#### Usage

```
reltest_simulate(
  model = "exp",
  nx = 20,
  tt,
  tt1,
  tt2,
  tt3,
  params,
  minxi = -10,
  maxxi = -10
)
```

#### **Arguments**

```
which distribution to test. Possibles values are "exp", "pareto_k2", "halfnorm",
model
                  "unif", "norm", "norm_dmgs", "gnorm_k3", "lnorm", "lnorm_dmgs", "logis",
                  "lst_k3", "cauchy", "gumbel", "frechet_k1", "weibull", "gev_k3", "exp_p1",
                  "pareto_p1k2", "norm_p1", "lnorm_p1", "logis_p1", "lst_p1k3", "cauchy_p1",
                  "gumbel_p1", "frechet_p2k1", "weibull_p2", "gev_p1k3", "norm_p12", "lst_p12k3",
                  "gamma", "invgamma", "invgauss", "gev", "gpd_k1", "gev_p1". "gev_p12".
                  "gev_p123".
                  the length of the training data to use.
nx
tt
                  predictor vector
                  predictor vector 1
tt1
                  predictor vector 2
tt2
tt3
                  predictor vector 2
                  values for the parameters for the specified distribution
params
                  minimum value for EVT shape parameter
minxi
                  maximum value for EVT shape parameter
maxxi
```

## Value

886 rgev\_p123\_minmax

rgev\_minmax

rgev but with maxlik xi guaranteed within bounds

## Description

rgev but with maxlik xi guaranteed within bounds

# Usage

```
rgev_minmax(nx, mu, sigma, xi, minxi = -0.45, maxxi = 0.45)
```

## Arguments

nx	length of training data
mu	the location parameter of the distribution
sigma	the sigma parameter of the distribution
xi	the shape parameter of the distribution
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi

#### Value

Vector

rgev\_p123\_minmax

rgev for gev\_p123 but with maxlik xi within bounds

#### **Description**

rgev for gev\_p123 but with maxlik xi within bounds

## Usage

```
rgev_p123_minmax(
    nx,
    mu,
    sigma,
    xi,
    t1,
    t2,
    t3,
    minxi = -0.45,
    maxxi = 0.45,
    centering = TRUE
)
```

rgev\_p12\_minmax 887

## **Arguments**

nx	length of training data
mu	the location parameter of the distribution
sigma	the sigma parameter of the distribution
xi	the shape parameter of the distribution
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi
centering	indicates whether the routine should center the data or not

#### Value

Vector

rgev\_p12\_minmax

rgev for gev\_p12 but with maxlik xi within bounds

## Description

rgev for gev\_p12 but with maxlik xi within bounds

## Usage

```
rgev_p12_minmax(
    nx,
    mu,
    sigma,
    xi,
    t1,
    t2,
    minxi = -0.45,
    maxxi = 0.45,
    centering = TRUE
)
```

# Arguments

nx length of training data
mu the location parameter of the distribution
sigma the sigma parameter of the distribution
xi the shape parameter of the distribution

888 rgev\_p1\_minmax

t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi
centering	indicates whether the routine should center the data or not

#### Value

Vector

rgev\_p1\_minmax rgev for gev\_p1 but with maxlik xi within bounds

## Description

rgev for gev\_p1 but with maxlik xi within bounds

## Usage

```
rgev_p1_minmax(
    nx,
    mu,
    sigma,
    xi,
    tt,
    minxi = -0.45,
    maxxi = 0.45,
    centering = TRUE
)
```

# Arguments

length of training data nx the location parameter of the distribution mu the sigma parameter of the distribution sigma the shape parameter of the distribution хi tt a vector of predictors minxi minimum value of shape parameter xi maxxi maximum value of shape parameter xi indicates whether the routine should center the data or not centering

## Value

rgpd\_k1\_minmax 889

rond	k1	minmax	

rgpd for gpd\_k1 but with maxlik xi within bounds

## Description

rgpd for gpd\_k1 but with maxlik xi within bounds

## Usage

```
rgpd_k1_minmax(nx, kloc, sigma, xi, minxi = -0.45, maxxi = 0.45)
```

## Arguments

nx	length of training data
kloc	the known location parameter
sigma	the sigma parameter of the distribution
xi	the shape parameter of the distribution
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi

#### Value

Vector

rhn	dmgs	_cpmet	hod
- אוייי	_umg	_cpilic t	iiou

Generates a comment about the method

## Description

Generates a comment about the method

## Usage

```
rhp_dmgs_cpmethod()
```

#### Value

String

890 testppm\_plot

rust\_pumethod

Generates a comment about the method

#### **Description**

Generates a comment about the method

#### Usage

```
rust_pumethod()
```

#### Value

String

testppm\_plot

Plotting routine for testppm

#### **Description**

Plots 9 diagnostics related to predictive probability matching.

## Usage

```
testppm_plot(
  model,
  ntrials,
  nrepeats,
  nx,
  params,
  nmethods,
  alpha,
  freqexceeded
)
```

#### **Arguments**

model which distribution to test. Possibles values are

ntrials the number of trials to run. 5000 typically gives good results.

nrepeats the number of entire repeats of the test to run, to check for convergence

nx the length of the training data.

params values for the parameters for the specified distribution

nmethods the number of methods being tested alpha the values of alpha being tested

freqexceeded the exceedance counts

#### Value

Plots the results of reliability testing

unif\_cp

Uniform Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qunif_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    means = FALSE,
    debug = FALSE,
    aderivs = TRUE
)

runif_cp(n, x, mlcp = TRUE, debug = FALSE, aderivs = TRUE)

dunif_cp(x, y = x, debug = FALSE, aderivs = TRUE)

punif_cp(x, y = x, debug = FALSE, aderivs = TRUE)
```

#### **Arguments**

Х	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The uniform distribution has probability density function

$$f(x; min, max) = \frac{1}{max - min}$$

and zero otherwise, where  $min \le x \le max$  is the random variable and min, max are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\lambda) \propto \frac{1}{max - min}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\* optionally returns the following:

If rust=TRUE:

 ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

## **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),

- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d25unif_example_data_v1
cat("length(x)=",length(x),"\n")
p=c(1:9)/10
q=qunif_cp(x,p)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qunif_cp)",
main="unif: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
```

weibull\_cp

Weibull Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qweibull_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    fd1 = 0.01,
    fd2 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE,
    aderivs = TRUE
)
```

```
n,
  х,
  fd1 = 0.01,
  fd2 = 0.01,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
dweibull_cp(
 х,
 y = x,
 fd1 = 0.01,
 fd2 = 0.01,
  rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
pweibull_cp(
 х,
 y = x,
 fd1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
tweibull_cp(n, x, fd1 = 0.01, fd2 = 0.01, debug = FALSE)
```

a vector of training data values

# **Arguments** ×

	· · · · · · · · · · · · · · · · · · ·
р	a vector of probabilities at which to generate predictive quantiles
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter $% \left( 1\right) =\left( 1\right) \left( $
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter $$
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)

dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Weibull distribution has exceedance distribution function

$$S(x; k, \sigma) = \exp\left(-\left(\frac{x}{\sigma}\right)^k\right)$$

where  $x \ge 0$  is the random variable and  $k > 0, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(k,\sigma) \propto \frac{1}{k\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

• ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible

weibull\_cp 901

• cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

902 weibull\_cp

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),

weibull\_cp 903

- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d52weibull_example_data_v1
p=c(1:9)/10
q=qweibull_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qweibull_cp)",
main="Weibull: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

904 weibull\_f1fa

wei		

DMGS equation 3.3, f1 term

### Description

DMGS equation 3.3, f1 term

### Usage

```
weibull_f1f(y, v1, fd1, v2, fd2)
```

eter

### Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

# Value

Matrix

wei	h.,	11	£1	۲,
wer	υu	TT	_	Ιd

The first derivative of the density

## Description

The first derivative of the density

### Usage

```
weibull_f1fa(x, v1, v2)
```

### Arguments

x a vector of training	data values
------------------------	-------------

v1 first parameterv2 second parameter

#### Value

Vector

weibull\_f2f 905

		-	
wei	hiil	I I	+ 7+
WCI	DU.		1 4 1

DMGS equation 3.3, f2 term

### Description

DMGS equation 3.3, f2 term

### Usage

```
weibull_f2f(y, v1, fd1, v2, fd2)
```

### Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

## eter

### Value

3d array

	<b>L</b> .	.11	f2	٠.
wel	· D	ιтт		па

The second derivative of the density

## Description

The second derivative of the density

### Usage

```
weibull_f2fa(x, v1, v2)
```

## Arguments

v1 first parameterv2 second parameter

#### Value

906 weibull\_fdd

weibull_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
weibull_fd(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

weibull_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
weibull_fdd(x, v1, v2)
```

### **Arguments**

X	a vector of training data values	,
X	a vector of training data values	,

v1 first parameter v2 second parameter

#### Value

weibull\_ldd 907

weibull_ldd	Second derivative matrix of the normalized log-likelihood

### Description

Second derivative matrix of the normalized log-likelihood

### Usage

```
weibull_ldd(x, v1, fd1, v2, fd2)
```

### Arguments

x	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Square scalar matrix

weibull_ldda	The second derivative of the normalized log-likelihood

## Description

The second derivative of the normalized log-likelihood

### Usage

```
weibull_ldda(x, v1, v2)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter

#### Value

908 weibull\_lddda

			_	
wei	hu	11	1 ~	144
wei	1717			11111

Third derivative tensor of the normalized log-likelihood

#### **Description**

Third derivative tensor of the normalized log-likelihood

#### Usage

```
weibull_lddd(x, v1, fd1, v2, fd2)
```

### Arguments

Χ	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Cubic scalar array

woi	hu1	1 1	ahhh

The third derivative of the normalized log-likelihood

## Description

The third derivative of the normalized log-likelihood

#### Usage

```
weibull_lddda(x, v1, v2)
```

## Arguments

X	a vector of	f training	data values
---	-------------	------------	-------------

v1 first parameterv2 second parameter

#### Value

3d array

weibull\_lmn 909

weibull_lmn	One component of the second derivative of the normalized log-likelihood

## Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
weibull_lmn(x, v1, fd1, v2, fd2, mm, nn)
```

### Arguments

х	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

## Value

Scalar value

weibull_lmnp	One component of the third derivative of the normalized log-likelihood
•	1 0

## Description

One component of the third derivative of the normalized log-likelihood

## Usage

```
weibull_lmnp(x, v1, fd1, v2, fd2, mm, nn, rr)
```

910 weibull\_logf

# Arguments

х	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

weibull_logf	Logf for RUST		
--------------	---------------	--	--

# Description

Logf for RUST

## Usage

```
weibull_logf(params, x)
```

## Arguments

params model parameters for calculating logf x a vector of training data values

## Value

Scalar value.

weibull\_logfdd 911

Detro() by thatew Clausen and Serguet Sokol	weibull_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
---------------------------------------------	----------------	------------------------------------------------------------------------------------------------------------------

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
weibull_logfdd(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Matrix

weibull_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

### Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
weibull_logfddd(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

## Value

3d array

912 weibull\_logscores

weibull_loglik	log-likelihood function	
----------------	-------------------------	--

### Description

log-likelihood function

### Usage

```
weibull_loglik(vv, x)
```

### Arguments

vv parameters

x a vector of training data values

#### Value

Scalar value.

weibull_logscores Log scores for MLE and RHP predictions calculated using leave-one- out
---------------------------------------------------------------------------------------------

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

### Usage

```
weibull_logscores(logscores, x, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

## **Arguments**

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
X	a vector of training data values
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

weibull\_means 913

weibull_means	MLE and RHP predictive means

## Description

MLE and RHP predictive means

### Usage

```
weibull_means(means, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2)
```

### Arguments

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

#### Value

Two scalars

weibull_mu1f	DMGS equation 3.3, mul term	
--------------	-----------------------------	--

### Description

DMGS equation 3.3, mu1 term

#### Usage

```
weibull_mu1f(alpha, v1, fd1, v2, fd2)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

914 weibull\_mu2f

#### Value

Matrix

weibull\_mu1fa

Minus the first derivative of the cdf, at alpha

### Description

Minus the first derivative of the cdf, at alpha

### Usage

```
weibull_mu1fa(alpha, v1, v2)
```

### Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

#### Value

Vector

weibull\_mu2f

DMGS equation 3.3, mu2 term

### Description

DMGS equation 3.3, mu2 term

### Usage

```
weibull_mu2f(alpha, v1, fd1, v2, fd2)
```

## **Arguments**

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

eter

weibull\_mu2fa 915

### Value

3d array

weibull\_mu2fa

Minus the second derivative of the cdf, at alpha

### Description

Minus the second derivative of the cdf, at alpha

### Usage

```
weibull_mu2fa(alpha, v1, v2)
```

### Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

#### Value

Matrix

weibull\_p1f

DMGS equation 3.3, p1 term

### Description

DMGS equation 3.3, p1 term

### Usage

```
weibull_p1f(y, v1, fd1, v2, fd2)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

916 weibull\_p2f

### Value

Matrix

weibull\_p1fa

The first derivative of the cdf

### Description

The first derivative of the cdf

### Usage

```
weibull_p1fa(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

weibull\_p2f

DMGS equation 3.3, p2 term

### Description

DMGS equation 3.3, p2 term

### Usage

```
weibull_p2f(y, v1, fd1, v2, fd2)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

weibull\_p2fa 917

#### Value

3d array

weibull\_p2fa

The second derivative of the cdf

#### **Description**

The second derivative of the cdf

#### Usage

```
weibull_p2fa(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter

v2 second parameter

#### Value

Matrix

weibull\_p2\_cp

weibull Distribution with a Predictor on the Scale Parameter, Predictions Based on a Calibrating Prior

#### Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y

• t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qweibull_p2_cp(
 Х,
  t,
  t0 = NA,
 n0 = NA,
 p = seq(0.1, 0.9, 0.1),
  fd1 = 0.01,
 d2 = 0.01,
 d3 = 0.01,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 predictordata = TRUE,
  centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
rweibull_p2_cp(
 n,
 х,
  t,
  t0 = NA,
 n0 = NA,
  fd1 = 0.01,
  d2 = 0.01,
 d3 = 0.01,
  rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
  aderivs = TRUE
dweibull_p2_cp(
  Х,
```

```
t,
  t0 = NA,
 n0 = NA,
 y = x,
 fd1 = 0.01,
 d2 = 0.01,
 d3 = 0.01,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
pweibull_p2_cp(
 Х,
  t,
 t0 = NA,
 n0 = NA,
 y = x,
 fd1 = 0.01,
 d2 = 0.01,
 d3 = 0.01,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
tweibull_p2_cp(n, x, t, fd1 = 0.01, d2 = 0.01, d3 = 0.01, debug = FALSE)
```

### Arguments

x	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter
d2	if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter
d3	if aderivs=FALSE, the delta used for numerical derivatives with respect to the third parameter $% \left( 1\right) =\left( 1\right) \left( 1\right) $
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)

waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- $\bullet$  ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Weibull distribution with predictor on the scale parameter has exceedance distribution function

$$S(x; k, a, b) = \exp\left(-\left(\frac{x}{\sigma(a, b)}\right)^k\right)$$

where  $x \ge 0$  is the random variable, k > 0 is the shape parameter and  $\sigma = e^{a+bt}$  is the scale parameter, modelled as a function of parameters a, b and predictor t.

The calibrating prior is given by the right Haar prior, which is

$$\pi(k,\sigma) \propto \frac{1}{k}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

#### If logscores=TRUE:

• ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)

• cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\* optionally returns the following:

#### If rust=TRUE:

 ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2024): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with linear predictor on the mean (norm\_p1),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d73weibull_p2_example_data_v1_x
tt=fitdistcp::d73weibull_p2_example_data_v1_t
p=c(1:9)/10
n0=10
```

weibull\_p2\_f1f 925

```
q=qweibull_p2_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qweibull_p2_cp)",
main="Weibull w/ p2: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

weibull\_p2\_f1f

DMGS equation 2.1, f1 term

### Description

DMGS equation 2.1, f1 term

#### Usage

```
weibull_p2_f1f(y, t0, v1, fd1, v2, d2, v3, d3)
```

### Arguments

у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

#### Value

926 weibull\_p2\_f2f

weibull_p2_f1fa	`a	fa	a
-----------------	----	----	---

The first derivative of the density

### **Description**

The first derivative of the density

### Usage

```
weibull_p2_f1fa(x, t, v1, v2, v3)
```

## **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

weibull\_p2\_f2f

DMGS equation 2.1, f2 term

## Description

DMGS equation 2.1, f2 term

### Usage

```
weibull_p2_f2f(y, t0, v1, fd1, v2, d2, v3, d3)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

weibull\_p2\_f2fa 927

#### Value

3d array

weibull\_p2\_f2fa

The second derivative of the density

#### **Description**

The second derivative of the density

#### Usage

```
weibull_p2_f2fa(x, t, v1, v2, v3)
```

#### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Matrix

weibull\_p2\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
weibull_p2_fd(x, t, v1, v2, v3)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

928 weibull\_p2\_ldd

### Value

Vector

weibull_p2_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
weibull_p2_fdd(x, t, v1, v2, v3)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

third parameter

### Value

v3

Matrix

weibull_p2_ldd	Second derivative matrix of the normalized log-likelihood
----------------	-----------------------------------------------------------

## Description

Second derivative matrix of the normalized log-likelihood

## Usage

```
weibull_p2_ldd(x, t, v1, fd1, v2, d2, v3, d3)
```

weibull\_p2\_ldda 929

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

### Value

Square scalar matrix

weibull_p2_ldda	The second derivative of the normalized log-likelihood
-----------------	--------------------------------------------------------

# Description

The second derivative of the normalized log-likelihood

# Usage

```
weibull_p2_ldda(x, t, v1, v2, v3)
```

# Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

930 weibull\_p2\_lddda

WP.	i hui	11 1	n2	1ddd
wc.	LDU.	L T	$\nu L_{-}$	Tuuu

Third derivative tensor of the normalized log-likelihood

#### **Description**

Third derivative tensor of the normalized log-likelihood

#### Usage

```
weibull_p2_lddd(x, t, v1, fd1, v2, d2, v3, d3)
```

#### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

#### Value

Cubic scalar array

weibull\_p2\_lddda

The third derivative of the normalized log-likelihood

### Description

The third derivative of the normalized log-likelihood

## Usage

```
weibull_p2_lddda(x, t, v1, v2, v3)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

weibull\_p2\_lmn 931

### Value

3d array

weibull_p2_lmn		
----------------	--	--

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
weibull_p2_lmn(x, t, v1, fd1, v2, d2, v3, d3, mm, nn)
```

## Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

### Value

Scalar value

932 weibull\_p2\_logf

weibull_p2_lmnp	One component of the second derivative of the normalized log-likelihood
-----------------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
weibull_p2_lmnp(x, t, v1, fd1, v2, d2, v3, d3, mm, nn, rr)
```

## **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

#### Value

Scalar value

weibull_p2_logf	Logf for RUST		
-----------------	---------------	--	--

## Description

Logf for RUST

## Usage

```
weibull_p2_logf(params, x, t)
```

weibull\_p2\_logfdd 933

## Arguments

params	model parameters	for calcul	lating logf

x a vector of training data valuest a vector or matrix of predictors

#### Value

Scalar value.

weibull_p2_logfdd	Second derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
weibull_p2_logfdd(x, t, v1, v2, v3)
```

## Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

934 weibull\_p2\_loglik

weibull_p2_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
weibull_p2_logfddd(x, t, v1, v2, v3)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
_	C .

v1 first parameterv2 second parameterv3 third parameter

#### Value

3d array

weibull\_p2\_loglik

observed log-likelihood function

### Description

observed log-likelihood function

### Usage

```
weibull_p2_loglik(vv, x, t)
```

### Arguments

VV	parameters

x a vector of training data valuest a vector or matrix of predictors

#### Value

Scalar value.

weibull\_p2\_logscores 935

weibull_p2_logscores	Log scores for MLE and RHP predictions calculated using leave-one-
	out

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

### Usage

```
weibull_p2_logscores(logscores, x, t, fd1, d2, d3, aderivs)
```

## Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
x	a vector of training data values
t	a vector or matrix of predictors
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
d3	the delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

## Description

weibull distribution: RHP mean

## Usage

```
weibull_p2_means(means, t0, ml_params, lddi, lddd, lambdad_rhp, nx, dim)
```

936 weibull\_p2\_mu1f

#### **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

t0 a single value of the predictor (specify either t0 or n0 but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

#### Value

Two scalars

weibull\_p2\_mu1f DMGS equation 3.3, mu1 term

### Description

DMGS equation 3.3, mu1 term

## Usage

```
weibull_p2_mu1f(alpha, t0, v1, fd1, v2, d2, v3, d3)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

#### Value

weibull\_p2\_mu1fa 937

wei	bul	1	n2	mıı'	1fa
MCT	.vu1	. т	υZ	IIIU	па

Minus the first derivative of the cdf, at alpha

#### **Description**

Minus the first derivative of the cdf, at alpha

## Usage

```
weibull_p2_mu1fa(alpha, t, v1, v2, v3)
```

## **Arguments**

alpha	a vector of values of alpha (one minus probability)
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

weibull\_p2\_mu2f

DMGS equation 3.3, mu2 term

## Description

DMGS equation 3.3, mu2 term

# Usage

```
weibull_p2_mu2f(alpha, t0, v1, fd1, v2, d2, v3, d3)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

938 weibull\_p2\_p1f

#### Value

3d array

weibull\_p2\_mu2fa

Minus the second derivative of the cdf, at alpha

## Description

Minus the second derivative of the cdf, at alpha

#### Usage

```
weibull_p2_mu2fa(alpha, t, v1, v2, v3)
```

## Arguments

alpha a vector of values of alpha (one minus probability)

t a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

weibull\_p2\_p1f

DMGS equation 2.1, p1 term

## Description

DMGS equation 2.1, p1 term

## Usage

```
weibull_p2_p1f(y, t0, v1, fd1, v2, d2, v3, d3)
```

weibull\_p2\_p1fa 939

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

#### Value

Matrix

weibull\_p2\_p1fa

The first derivative of the cdf

# Description

The first derivative of the cdf

# Usage

```
weibull_p2_p1fa(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

## Value

Vector

940 weibull\_p2\_p2fa

wei	hu.	11	n2	n2f
wei	υu.	ιт_	$\nu_{-}$	PΖI

DMGS equation 2.1, p2 term

## **Description**

DMGS equation 2.1, p2 term

## Usage

```
weibull_p2_p2f(y, t0, v1, fd1, v2, d2, v3, d3)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
d3	the delta used in the numerical derivatives with respect to the parameter

#### Value

3d array

weibull	n2	n2fa

The second derivative of the cdf

## Description

The second derivative of the cdf

# Usage

```
weibull_p2_p2fa(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

weibull\_p2\_pd 941

## Value

Matrix

Anarew Ciausen and Serguei Sokol	weibull_p2_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
----------------------------------	---------------	---------------------------------------------------------------------------------------------------------

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
weibull_p2_pd(x, t, v1, v2, v3)
```

## Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

## Value

Vector

weibull_p2_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
weibull_p2_pdd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

v3 third parameter

#### Value

Matrix

```
weibull_p2_predictordata
```

Predicted Parameter and Generalized Residuals

## Description

Predicted Parameter and Generalized Residuals

## Usage

```
weibull_p2_predictordata(predictordata, x, t, t0, params)
```

## Arguments

predictordata	logical that indicates whether to calculate and return predictordata
x	a vector of training data values
t	a vector or matrix of predictors
t0	a single value of the predictor (specify either t0 or n0 but not both)
params	model parameters for calculating logf

#### Value

Two vectors

weibull\_p2\_waic 943

weibull\_p2\_waic

Waic

# Description

Waic

# Usage

```
weibull_p2_waic(
  waicscores,
  x,
  t,
  v1hat,
  fd1,
  v2hat,
  d2,
  v3hat,
  d3,
  lddi,
  lddd,
  lambdad,
  aderivs = TRUE
)
```

# Arguments

waics	scores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X		a vector of training data values
t		a vector or matrix of predictors
v1hat		first parameter
fd1		the fractional delta used in the numerical derivatives with respect to the parameter
v2hat		second parameter
d2		the delta used in the numerical derivatives with respect to the parameter
v3hat		third parameter
d3		the delta used in the numerical derivatives with respect to the parameter
lddi		inverse observed information matrix
lddd		third derivative of log-likelihood
lambo	lad	derivative of the log prior
aderi	.vs	logical for whether to use analytic derivatives (instead of numerical)

944 weibull\_pdd

#### Value

Two numeric values.

weibull_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by
weibuli_pu	Andrew Clausen and Serguei Sokol
	Anarew Ciausen ana Serguei Sokoi

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
weibull_pd(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

## Value

Vector

weibull_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

#### **Description**

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
weibull_pdd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

weibull\_waic 945

weibull_waic Waicf	for RUST	!
--------------------	----------	---

# Description

Waic for RUST

# Usage

```
weibull_waic(
   waicscores,
   x,
   v1hat,
   fd1,
   v2hat,
   fd2,
   lddi,
   lddd,
   lambdad,
   aderivs
)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
v1hat	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

## Value

Two numeric values.

# **Index**

adhoc_dmgs_cpmethod, 26	cauchy_p1_logfdd, 61
analytic_cpmethod, 26	cauchy_p1_logfddd, 62
	cauchy_p1_loglik,62
bayesian_dq_4terms_v1, 27	cauchy_p1_logscores, 63
1 1 25	cauchy_p1_means, 63
calc_revert2ml, 27	cauchy_p1_mu1f, 64
cauchy_cp, 28	cauchy_p1_mu2f, 65
cauchy_f1f, 35	cauchy_p1_p1f, 65
cauchy_f1fa, 35	cauchy_p1_p2f, 66
cauchy_f2f, 36	cauchy_p1_predictordata,67
cauchy_f2fa, 36	cauchy_p1_waic,67
cauchy_fd, 37	cauchy_p1f, 45
cauchy_fdd, 37	cauchy_p2f, 68
cauchy_ldd, 38	cauchy_waic, 69
cauchy_1dda, 38	$crhpflat\_dmgs\_cpmethod, 70$
cauchy_1ddd, 39	
cauchy_lddda, 39	d100gamma_example_data_v1,70
cauchy_1mn, 40	d101invgamma_example_data_v1,70
cauchy_lmnp, 40	d102invgauss_example_data_v1,70
cauchy_logf, 41	d105burr_example_data_v1,71
cauchy_logfdd, 42	d10exp_example_data_v1, 71
cauchy_logfddd, 42	d110gev_example_data_v1,71
cauchy_loglik, 43	d11pareto_k2_example_data_v1,71
cauchy_logscores, 43	d120gpd_k1_example_data_v1,71
cauchy_mu1f, 44	d150gev_p1_example_data_v1_t,72
cauchy_mu2f, 44	d150gev_p1_example_data_v1_x,72
cauchy_p1_cp, 45	d151gev_p12_example_data_v1_t,72
cauchy_p1_f1f, 53	d151gev_p12_example_data_v1_x, 72
cauchy_p1_f1fa, 54	d152gev_p123_example_data_v1_t, 72
cauchy_p1_f2f, 54	d152gev_p123_example_data_v1_x, 73
cauchy_p1_f2fa, 55	d20halfnorm_example_data_v1,73
cauchy_p1_fd, 55	d25unif_example_data_v1,73
cauchy_p1_fdd, 56	d30norm_example_data_v1,73
cauchy_p1_ldd, 56	d31norm_dmgs_example_data_v1,73
cauchy_p1_ldda, 57	d32gnorm_k3_example_data_v1,74
cauchy_p1_1ddd, 58	d35lnorm_example_data_v1,74
cauchy_p1_1ddda, 58	d36lnorm_dmgs_example_data_v1,74
cauchy_p1_1mn, 59	d40logis_example_data_v1,74
cauchy_p1_lmnp, 60	d41lst_k3_example_data_v1,74
cauchy_p1_logf, 60	d42cauchy_example_data_v1,75

d50gumbel_example_data_v1,75	dexp_p1_cp (exp_p1_cp), 130
d51frechet_k1_example_data_v1, 75	dexp_p1sub, 87
d52weibull_example_data_v1,75	dexpsub, 86
d53gev_k3_example_data_v1,75	dfrechet_k1_cp (frechet_k1_cp), 157
d55exp_p1_example_data_v1_t, 76	dfrechet_p2k1,88
d55exp_p1_example_data_v1_x, 76	dfrechet_p2k1_cp (frechet_p2k1_cp), 180
d56pareto_p1k2_example_data_v1_t, 76	dfrechet_p2k1sub, 89
d56pareto_p1k2_example_data_v1_x, 76	dfrechetsub, 88
d60norm_p1_example_data_v1_t, 76	dgamma_cp (gamma_cp), 209
d60norm_p1_example_data_v1_x, 77	dgammasub, 90
d61lnorm_p1_example_data_v1_t, 77	dgev_cp (gev_cp), 230
d61lnorm_p1_example_data_v1_x, 77	dgev_k3_cp (gev_k3_cp), 245
d62logis_p1_example_data_v1_t, 77	dgev_k3sub, 91
d62logis_p1_example_data_v1_x, 77	dgev_p1, 92
d63lst_p1k3_example_data_v1_t, 78	dgev_p12, 92
d63lst_p1k3_example_data_v1_x, 78	dgev_p123, 93
d64cauchy_p1_example_data_v1_t, 78	dgev_p123_cp (gev_p123_cp), 277
d64cauchy_p1_example_data_v1_x, 78	dgev_p123sub, 94
d70gumbel_p1_example_data_v1_t, 78	dgev_p12_cp (gev_p12_cp), 321
d70gumbel_p1_example_data_v1_x, 79	dgev_p12sub, 95
d71frechet_p2k1_example_data_v1_t, 79	dgev_p1_cp (gev_p1_cp), 374
d71frechet_p2k1_example_data_v1_x, 79	dgev_p1k3, 96
d72weibull_p1_example_data_v1_t, 79	dgev_p1k3_cp (gev_p1k3_cp), 349
d72weibull_p1_example_data_v1_x, 79	dgev_p1k3sub, 97
d73weibull_p2_example_data_v1_t, 80	dgev_p1sub, 97
d73weibull_p2_example_data_v1_x, 80	dgevsub, 90
d74gev_p1k3_example_data_v1_t, 80	dgnorm_k3_cp (gnorm_k3_cp), 405
d74gev_p1k3_example_data_v1_x, 80	dgnorm_k3sub, 99
d80norm_p12_example_data_v1_t1, 80	dgpd_k1_cp (gpd_k1_cp), 437
d80norm_p12_example_data_v1_t1, 80	dgpdsub, 99
d80norm_p12_example_data_v1_x, 81	dgumbel_cp (gumbel_cp), 462
d81lst_p12k3_example_data_v1_t1, 81	dgumbel_cp (gumbel_cp), 402
d811st_p12k3_example_data_v1_t1,81	dgumbel_p1, 101 dgumbel_p1_cp (gumbel_p1_cp), 481
d811st_p12k3_example_data_v1_tz, 81	dgumbel_p1sub, 101
d82weibull_p12_example_data_v1_t1, 82	dgumbelsub, 100
d82weibull_p12_example_data_v1_t1, 82	dhalfnorm_cp (halfnorm_cp), 510
d82weibull_p12_example_data_v1_tz, 82	dhalfnormsub, 102
·	
dcauchy_cp (cauchy_cp), 28	dinvgamma_cp (invgamma_cp), 529 dinvgammasub, 102
deauchy_p1, 83	
dcauchy_p1_cp (cauchy_p1_cp), 45	dinvgauss_cp (invgauss_cp), 549
dcauchy_p1sub, 83	dinvgausssub, 103
dcauchysub, 82	dlnorm_cp (lnorm_cp), 570
deriv_copyfdd, 84	dlnorm_dmgs_cp (lnorm_dmgs_cp), 577
deriv_copyld2, 85	dlnorm_dmgssub, 104
deriv_copyldd, 85	dlnorm_p1, 104
deriv_copylddd, 86	dlnorm_p1_cp (lnorm_p1_cp), 599
dexp_cp (exp_cp), 116	dlnorm_p1sub, 105
dexp_p1, 87	dlnormsub, 103

dlogis2sub, 105	exp_p1_f1f, 137
dlogis_cp (logis_cp), 623	exp_p1_f1fa, 138
dlogis_p1, 106	exp_p1_f2f, 138
dlogis_p1_cp (logis_p1_cp), 642	exp_p1_f2fa, 139
dlogis_p1sub, 106	exp_p1_fd, 139
dlst_k3_cp (lst_k3_cp), 670	exp_p1_fdd, 140
dlst_k3sub, 107	exp_p1_ldd, 140
dlst_p1k3, 108	exp_p1_ldda, 141
dlst_p1k3_cp (lst_p1k3_cp), 690	exp_p1_lddd, 141
dlst_p1k3sub, 108	exp_p1_lddda, 142
dmgs, 109	exp_p1_lmn, 142
dnorm_cp (norm_cp), 745	exp_p1_lmnp, 143
dnorm_dmgs_cp (norm_dmgs_cp), 750	exp_p1_logf, 143
dnorm_dmgssub, 110	exp_p1_logfdd, 144
dnorm_p1, 111	exp_p1_logfddd, 145
dnorm_p1_cp (norm_p1_cp), 773	exp_p1_loglik, 145
dnorm_p1_formula, 112	exp_p1_logscores, 146
dnorm_p1sub, 111	exp_p1_means, 146
dnormsub, 110	exp_p1_mu1f, 147
dpareto_k2_cp (pareto_k2_cp), 800	exp_p1_mu1fa, 148
dpareto_k2_sub, 112	exp_p1_mu2f, 148
dpareto_p1k2, 113	exp_p1_mu2fa, 149
dpareto_p1k2_cp (pareto_p1k2_cp), 818	exp_p1_ma2*a, 119
dpareto_p1k2_up, (pareto_p1k2_up), 010	exp_p1_p1fa, 150
dunif_cp (unif_cp), 891	exp_p1_p1fd, 150
dunif_formula, 114	exp_p1_p2fa, 151
dweibull_cp (weibull_cp), 897	exp_p1_p21a, 151 exp_p1_pd, 151
dweibull_p2, 115	exp_p1_pd, 151 exp_p1_pdd, 152
dweibull_p2_cp (weibull_p2_cp), 917	exp_p1_predictordata, 152
dweibull_p2sub, 115	exp_p1_waic, 153
dweibullsub, 114	exp_p1fa, 129
uwerburrsub, 114	exp_p11a, 129 exp_p2fa, 154
exp_cp, 116	exp_pd, 154
exp_f1f, 122	exp_pdd, 155
exp_f1fa, 122	
exp_f2f, 123	exp_waic, 155
exp_f2fa, 123	fixgevrange, 156
exp_fd, 124	fixgpdrange, 156
exp_fdd, 124	frechet_k1_cp, 157
exp_1111, 125	frechet_k1_f1f, 164
exp_1dd, 125	frechet_k1_f1fa, 164
exp_1dda, 126	frechet_k1_f2f, 165
exp_1ddd, 126	frechet_k1_f2fa, 166
exp_1ddda, 127	frechet_k1_fd, 166
exp_logf, 127	frechet_k1_fdd, 167
exp_logf, 127 exp_logfdd, 128	frechet_k1_ldd, 167
exp_logfddd, 128	frechet_k1_ldda, 168
exp_logscores, 129	frechet_k1_lddd, 168
exp_10gscores, 129 exp_p1_cp, 130	frechet_k1_lddda, 169
CAP_P 1_CP, 130	i i eche c_ki_tuuua, 109

frechet_k1_lmn, 169	frechet_p2k1_predictordata, 207
frechet_k1_lmnp, 170	frechet_p2k1_waic, 208
frechet_k1_logf, 171	
frechet_k1_logfdd, 171	gamma_cp, 209
frechet_k1_logfddd, 172	gamma_f1f, 216
frechet_k1_mu1f, 172	gamma_f1fa, 216
frechet_k1_mu1fa, 173	gamma_f2f, 217
frechet_k1_mu2f, 173	gamma_f2fa, 217
frechet_k1_mu2fa, 174	gamma_fd, 218
frechet_k1_p1f, 174	gamma_fdd, 218
frechet_k1_p1fa, 175	gamma_gg, 219
frechet_k1_p2f, 175	gamma_gmn, 219
frechet_k1_p2fa, 176	gamma_1dd, 220
frechet_k1_pd, 176	gamma_ldda, 220
frechet_k1_pdd, 177	gamma_lddd, 221
frechet_k1_waic, 177	gamma_lddda, 221
frechet_loglik, 178	gamma_lmn, 222
frechet_logscores, 179	gamma_lmnp, 222
frechet_means, 179	gamma_logf, 223
frechet_p2k1_cp, 180	gamma_logfdd, 224
frechet_p2k1_f1f, 188	gamma_logfddd, 224
frechet_p2k1_f1fa, 189	gamma_loglik, 225
frechet_p2k1_f2f, 189	gamma_logscores, 225
frechet_p2k1_f2fa, 190	gamma_means, 226
frechet_p2k1_fd, 191	gamma_mu1f, 226
frechet_p2k1_fdd, 191	gamma_mu2f, 227
frechet_p2k1_ldd, 192	gamma_p1f, 227
frechet_p2k1_ldda, 192	gamma_p2f, 228
frechet_p2k1_lddd, 193	gamma_waic, 229
frechet_p2k1_lddda, 194	gev_checkmle, 229
	gev_cp, 230
frechet_p2k1_lmn, 195	gev_f1f, 239
frechet_p2k1_lmnp, 195	gev_f1fa, 240
frechet_p2k1_logf, 196	gev_f2f, 240
frechet_p2k1_logfdd, 197	gev_f2fa, 241
frechet_p2k1_logfddd, 197	gev_fd, 241
frechet_p2k1_loglik, 198	gev_fdd, 242
frechet_p2k1_logscores, 199	gev_ggd_mev, 242
frechet_p2k1_means, 199	gev_ggid_mev, 243
frechet_p2k1_mu1f, 200	gev_k12_ppm_minusloglik, 244
frechet_p2k1_mu1fa, 201	gev_k3_cp, 245
frechet_p2k1_mu2f, 202	gev_k3_f1f, 253
frechet_p2k1_mu2fa, 202	gev_k3_f1fa, 253
frechet_p2k1_p1f, 203	gev_k3_f2f, 254
frechet_p2k1_p1fa, 204	gev_k3_f2fa, 254
frechet_p2k1_p2f, 204	gev_k3_fd, 255
frechet_p2k1_p2fa, 205	gev_k3_fdd, 255
frechet_p2k1_pd, 206	gev_k3_1dd, 256
frechet_p2k1_pdd, 206	gev_k3_1dda, 256

gev_k3_1ddd, 257	gev_p123_logf, 297
gev_k3_1ddda, 257	gev_p123_logfdd, 298
gev_k3_1mn, 258	gev_p123_logfddd, 298
gev_k3_1mnp, 258	gev_p123_loglik, 299
gev_k3_logf, 259	gev_p123_means, 300
gev_k3_logfdd, 260	gev_p123_mu1f, 300
gev_k3_logfddd, 260	gev_p123_mu1fa, 301
gev_k3_loglik, 261	gev_p123_mu2f, 302
gev_k3_means, 261	gev_p123_mu2fa, 303
gev_k3_mu1f, 262	gev_p123_pd, 304
gev_k3_mu1fa, 262	gev_p123_pdd, 305
gev_k3_mu2f, 263	gev_p123_predictordata, 305
gev_k3_mu2fa, 263	gev_p123_setics, 306
gev_k3_pd, 264	gev_p123_waic, 307
gev_k3_pdd, 264	gev_p12_checkmle, 320
gev_k3_waic, 265	gev_p12_cp, 321
gev_ld12a, 266	gev_p12_f1f, 330
gev_lda, 266	gev_p12_f1fa, 331
gev_ldd, 267	gev_p12_f1fa, 331
gev_1dda, 267 gev_1dda, 267	gev_p12_f2fa, 332
gev_1ddd, 268	gev_p12_fd, 333
gev_lddda, 268	gev_p12_fdd, 333
	gev_p12_ggd, 334
gev_lmn, 269	
gev_lmnp, 270	gev_p12_ldd, 335
gev_logf, 270	gev_p12_ldda, 335
gev_logfd, 271	gev_p12_lddd, 336
gev_logfdd, 271	gev_p12_lddda, 337
gev_logfddd, 272	gev_p12_lmn, 337
gev_loglik, 273	gev_p12_1mnp, 338
gev_means, 273	gev_p12_logf, 339
gev_mu1f, 274	gev_p12_logfdd, 340
gev_mu1fa, 275	gev_p12_logfddd, 340
gev_mu2f, 275	gev_p12_loglik, 341
gev_mu2fa, 276	gev_p12_means, 342
gev_p123_checkmle, 276	gev_p12_mu1f, 342
gev_p123_cp, 277	gev_p12_mu1fa, 343
gev_p123_f1f, 287	gev_p12_mu2f, 344
gev_p123_f1fa, 288	gev_p12_mu2fa, 344
gev_p123_f2f, 288	gev_p12_pd, 345
gev_p123_f2fa, 289	gev_p12_pdd, 346
gev_p123_fd, 290	<pre>gev_p12_predictordata, 346</pre>
gev_p123_fdd, 291	gev_p12_setics, 347
gev_p123_1dd, 291	gev_p12_waic, 348
gev_p123_ldda, 292	gev_p12k3_f1f, 308
gev_p123_1ddd, 293	gev_p12k3_f1fa, 309
gev_p123_lddda, 294	gev_p12k3_f2f, 309
gev_p123_lmn, 294	gev_p12k3_f2fa, 310
gev_p123_lmnp, 296	gev_p12k3_fd, 311

gev_p12k3_fdd, 311	gev_p1k3_fdd, 361
gev_p12k3_ldd, 312	gev_p1k3_ldd, 361
gev_p12k3_ldda, 313	gev_p1k3_ldda, 362
gev_p12k3_lddd, 313	gev_p1k3_lddd, 363
gev_p12k3_lddda, 314	gev_p1k3_1ddda, 363
gev_p12k3_logfdd, 315	gev_p1k3_1mn, 364
gev_p12k3_logfddd, 315	gev_p1k3_1mnp, 365
gev_p12k3_mu1f, 316	gev_p1k3_logf, 365
gev_p12k3_mu1fa, 317	gev_p1k3_logfdd, 366
gev_p12k3_mu2f, 317	gev_p1k3_logfddd, 367
gev_p12k3_mu2fa, 318	gev_p1k3_loglik, 367
gev_p12k3_pd, 319	gev_p1k3_means, 368
gev_p12k3_pdd, 319	gev_p1k3_mu1f, 368
gev_p1_checkmle, 374	gev_p1k3_mu1fa, 369
gev_p1_cp, 374	gev_p1k3_mu2f, 370
gev_p1_f1f, 384	gev_p1k3_mu2fa, 370
gev_p1_f1fa, 385	gev_p1k3_pd, 371
gev_p1_f2f, 385	gev_p1k3_pdd, 372
gev_p1_f2fa, 386	<pre>gev_p1k3_predictordata, 372</pre>
gev_p1_fd, 387	gev_p1k3_waic, 373
gev_p1_fdd, 387	gev_pd, 402
gev_p1_ggd, 388	gev_pdd, 403
gev_p1_1dd, 389	gev_pwm_params, 403
gev_p1_1dda, 389	gev_setics, 404
gev_p1_1ddd, 390	gev_waic, 404
gev_p1_1ddda, 391	gnorm_k3_cp, 405
gev_p1_1mn, 391	gnorm_k3_f1f, 412
${\tt gev\_p1\_lmnp, 392}$	gnorm_k3_f1fa, 413
gev_p1_logf, 393	gnorm_k3_f2f, 414
gev_p1_logfdd, 393	gnorm_k3_f2fa, 414
gev_p1_logfddd, 394	gnorm_k3_fd, 415
gev_p1_loglik, 395	gnorm_k3_fdd, 415
$gev_p1_means, 395$	gnorm_k3_ldd, 416
gev_p1_mu1f, 396	gnorm_k3_ldda, 417
gev_p1_mu1fa, 397	gnorm_k3_lddd, 417
gev_p1_mu2f, 397	gnorm_k3_lddda,418
gev_p1_mu2fa, 398	gnorm_k3_1mn, 418
gev_p1_pd, 399	gnorm_k3_logf, 419
gev_p1_pdd, 399	gnorm_k3_logfdd, 419
gev_p1_predictordata, 400	gnorm_k3_logfddd, 420
gev_p1_setics, 401	gnorm_k3_loglik, 420
${\tt gev\_p1\_waic}, {\tt 401}$	<pre>gnorm_k3_logscores, 421</pre>
gev_p1k3_cp, 349	gnorm_k3_mu1f, 421
gev_p1k3_f1f, 357	gnorm_k3_mu2f, 422
gev_p1k3_f1fa, 358	gnorm_k3_p1f, 423
gev_p1k3_f2f, 359	gnorm_k3_p2f, 423
gev_p1k3_f2fa, 359	gnorm_lmnp, 424
gev_p1k3_fd, 360	gnorm_waic, 425

gpd_k13_f1f, 426	gpd_k1_pd, 459
gpd_k13_f1fa, 426	gpd_k1_pdd, 460
gpd_k13_f2f, 427	gpd_k1_setics, 460
gpd_k13_f2fa, 427	gpd_k1_waic, 461
gpd_k13_fd, 428	gumbel_cp, 462
gpd_k13_fdd, 428	gumbel_f1f, 469
gpd_k13_l11, 429	gumbel_f1fa, 469
gpd_k13_l111, 429	gumbel_f2f, 470
gpd_k13_ldd, 430	gumbel_f2fa, 470
gpd_k13_1dda, 430	gumbel_fd, 471
gpd_k13_1ddd, 431	<pre>gumbel_fdd, 471</pre>
gpd_k13_1ddda, 431	gumbel_1dd, 472
gpd_k13_logfdd, 432	gumbel_ldda,472
gpd_k13_logfddd, 432	gumbel_lddd, 473
gpd_k13_mu1f, 433	<pre>gumbel_lddda, 473</pre>
gpd_k13_mu1fa, 433	gumbel_lmn, 474
gpd_k13_mu2f, 434	<pre>gumbel_lmnp, 474</pre>
gpd_k13_mu2fa, 434	<pre>gumbel_logf, 475</pre>
gpd_k13_p1f, 435	gumbel_logfdd, 476
gpd_k13_p2f, 435	gumbel_logfddd, 476
gpd_k13_pd, 436	gumbel_loglik, 477
gpd_k13_pdd, 436	gumbel_logscores, 477
gpd_k1_checkmle, 437	gumbel_means, 478
gpd_k1_cp, 437	gumbel_mu1f, 478
gpd_k1_f1f, 446	gumbel_mu1fa, 479
gpd_k1_f1fa, 447	gumbel_mu2f, 479
gpd_k1_f2f, 447	gumbel_mu2fa, 480
gpd_k1_f2fa, 448	gumbel_p1_cp, 481
gpd_k1_fd, 448	gumbel_p1_f1f, 489
gpd_k1_fdd, 449	gumbel_p1_f1fa, 490
gpd_k1_ggd_mev, 449	gumbel_p1_f2f, 490
gpd_k1_ldd, 450	gumbel_p1_f2fa, 491
gpd_k1_ldda, 450	gumbel_p1_fd, 491
gpd_k1_lddd, 451	gumbel_p1_fdd, 492
gpd_k1_lddda, 451	gumbel_p1_ldd, 492
gpd_k1_lmn, 452	gumbel_p1_ldda, 493
gpd_k1_lmnp, 452	gumbel_p1_lddd, 494
gpd_k1_logf, 453	gumbel_p1_lddda, 494
gpd_k1_logfdd, 454	gumbel_p1_lmn, 495
gpd_k1_logfddd, 454	gumbel_p1_lmnp, 496
gpd_k1_loglik, 455	gumbel_p1_logf, 496
gpd_k1_means, 455	gumbel_p1_logfdd, 497
gpd_k1_mu1f, 456	gumbel_p1_logfddd, 498
gpd_k1_mu1fa, 457	gumbel_p1_loglik, 498
	gumbel_p1_logscores, 499
gpd_k1_mu2f, 457	
gpd_k1_mu2fa, 458	gumbel_p1_means, 499
gpd_k1_p1f, 458	gumbel_p1_mu1f,500
gpd_k1_p2f, 459	$gumbel_p1_mu1fa, 501$

$gumbel_p1_mu2f, 501$	invgamma_fd, 539
$gumbel_p1_mu2fa, 502$	invgamma_fdd,539
$gumbel_p1_p1f, 502$	invgamma_ldd,540
<pre>gumbel_p1_p1fa, 503</pre>	invgamma_ldda,540
gumbel_p1_p2f, 504	invgamma_lddd, 541
$gumbel_p1_p2fa, 504$	invgamma_lddda,541
<pre>gumbel_p1_pd, 505</pre>	invgamma_lmn, 542
<pre>gumbel_p1_pdd, 505</pre>	invgamma_lmnp, 542
<pre>gumbel_p1_predictordata, 506</pre>	invgamma_logf, 543
<pre>gumbel_p1_waic, 507</pre>	invgamma_logfdd, 544
<pre>gumbel_p1f, 480</pre>	invgamma_logfddd, 544
<pre>gumbel_p1fa, 481</pre>	invgamma_loglik, 545
gumbel_p2f, 508	<pre>invgamma_logscores, 545</pre>
<pre>gumbel_p2fa, 508</pre>	invgamma_mu1f,546
<pre>gumbel_pd, 509</pre>	invgamma_mu2f, 546
<pre>gumbel_pdd, 509</pre>	invgamma_p1f,547
<pre>gumbel_waic, 510</pre>	invgamma_p2f, 547
	invgamma_waic,548
halfnorm_cp, 510	invgauss_cp, 549
halfnorm_f1f, 517	invgauss_f1f, 556
halfnorm_f1fa, 518	invgauss_f1fa,557
halfnorm_f2f, 518	invgauss_f2f, 557
halfnorm_f2fa, 519	invgauss_f2fa, 558
halfnorm_fd, 519	invgauss_fd, 558
halfnorm_fdd, 520	invgauss_fdd,559
halfnorm_gg, 520	invgauss_ldd, 559
halfnorm_gg11, 521	invgauss_ldda, 560
halfnorm_1111, 521	invgauss_lddd, 560
halfnorm_ldd, 522	invgauss_lddda, 561
halfnorm_ldda, 522	invgauss_lmn, 561
halfnorm_lddd, 523	invgauss_lmnp, 562
halfnorm_lddda, 523	invgauss_logf, 562
halfnorm_logf, 524	invgauss_logfdd, 563
halfnorm_logfdd, 524	invgauss_logfddd, 563
halfnorm_logfddd, 525	invgauss_loglik, 564
halfnorm_loglik, 525	invgauss_logscores, 564
halfnorm_logscores, 526	invgauss_means, 565
halfnorm_means, 526	invgauss_mu1f, 566
halfnorm_mu1f, 527	invgauss_mu2f, 566
halfnorm_mu2f, 527	invgauss_p1f, 567
halfnorm_p1f, 528	invgauss_p2f, 567
halfnorm_p2f, 528	invgauss_waic, 568
halfnorm_waic, 529	
	jpf2p, 569
invgamma_cp, 529	jpf3p, 569
invgamma_f1f, 536	jpf4p, 570
invgamma_f1fa, 537	
invgamma_f2f, 538	lnorm_cp, 570
invgamma_f2fa, 538	<pre>lnorm_dmgs_cp, 577</pre>

lnorm_dmgs_gg11,583	lnorm_p1_mu2fa, 617
lnorm_dmgs_gg12, 584	lnorm_p1_p1fa, 617
lnorm_dmgs_gg22, 584	lnorm_p1_p2fa, 618
<pre>lnorm_dmgs_loglik, 585</pre>	lnorm_p1_pd, 618
<pre>lnorm_dmgs_logscores, 585</pre>	lnorm_p1_pdd, 619
lnorm_dmgs_means, 586	<pre>lnorm_p1_predictordata, 619</pre>
lnorm_dmgs_mu1f, 586	lnorm_p1_waic, 620
lnorm_dmgs_mu2f, 587	lnorm_p1fa, 598
lnorm_dmgs_p1f, 587	lnorm_p2fa, 621
lnorm_dmgs_p2f, 588	lnorm_pd, 621
lnorm_dmgs_waic, 588	lnorm_pdd, 622
lnorm_f1f, 589	lnorm_waic, 622
lnorm_f1fa, 590	logis_cp, 623
lnorm_f2f, 590	logis_f1f, 630
lnorm_f2fa, 591	logis_f1fa, 630
lnorm_fd, 591	logis_f2f, 631
lnorm_fdd, 592	logis_f2fa, 631
lnorm_ldd, 592	logis_fd, 632
lnorm_ldda, 593	logis_fdd, 632
lnorm_lddd, 593	logis_1dd, 633
lnorm_lddda, 594	logis_ldda, 633
lnorm_lmn, 594	logis_lddd, 634
lnorm_lmnp, 595	logis_lddda, 634
lnorm_logf, 595	logis_lmn, 635
lnorm_logfdd, 596	logis_lmnp, 635
lnorm_logfddd, 596	logis_logf, 636
lnorm_logscores, 597	logis_logfdd, 637
lnorm_mu1fa, 597	logis_logfddd, 637
lnorm_mu2fa, 598	logis_loglik,638
lnorm_p1_cp, 599	logis_logscores, 638
lnorm_p1_f1f, 606	logis_mu1f, 639
lnorm_p1_f1fa, 607	logis_mu1fa,639
lnorm_p1_f2f, 607	logis_mu2f,640
lnorm_p1_f2fa, 608	logis_mu2fa,640
lnorm_p1_fd, 608	logis_p1_cp, 642
lnorm_p1_fdd, 609	logis_p1_f1f,649
lnorm_p1_ldd, 609	logis_p1_f1fa,650
lnorm_p1_ldda, 610	logis_p1_f2f,651
lnorm_p1_lddd, 611	logis_p1_f2fa, 651
lnorm_p1_lddda, 611	logis_p1_fd, 652
lnorm_p1_lmn, 612	logis_p1_fdd,652
lnorm_p1_lmnp, 613	logis_p1_ldd, 653
lnorm_p1_logf, 613	logis_p1_ldda,654
lnorm_p1_logfdd, 614	logis_p1_lddd,654
<pre>lnorm_p1_logfddd, 615</pre>	logis_p1_lddda,655
lnorm_p1_loglik, 615	logis_p1_lmn, 655
<pre>lnorm_p1_logscores, 616</pre>	logis_p1_lmnp,656
lnorm_p1_mu1fa, 616	logis_p1_logf,657

logis_p1_logfdd, 657	lst_p1k3_f1f, 698
logis_p1_logfddd,658	lst_p1k3_f1fa, 699
logis_p1_loglik,658	lst_p1k3_f2f, 699
logis_p1_logscores, 659	lst_p1k3_f2fa, 700
logis_p1_means, 659	lst_p1k3_fd, 701
logis_p1_mu1f, 660	lst_p1k3_fdd, 701
logis_p1_mu1fa, 661	lst_p1k3_ldd, 702
logis_p1_mu2f, 661	lst_p1k3_ldda, 703
logis_p1_mu2fa, 662	lst_p1k3_lddd, 703
logis_p1_p1f, 662	lst_p1k3_lddda, 704
logis_p1_p1fa, 663	lst_p1k3_lmn, 705
logis_p1_p2f, 664	lst_p1k3_lmnp, 705
logis_p1_p2fa, 664	lst_p1k3_logf, 706
logis_p1_pd, 665	lst_p1k3_logfdd, 707
logis_p1_pdd, 665	lst_p1k3_logfddd, 707
logis_p1_predictordata, 666	lst_p1k3_loglik, 708
logis_p1_waic, 667	lst_p1k3_logscores, 709
logis_p1f, 641	lst_p1k3_mu1f, 709
logis_p1fa, 641	lst_p1k3_mu2f, 710
logis_p17a, 041	lst_p1k3_p1f, 711
	lst_p1k3_p2f, 711
logis_p2fa, 668	lst_p1k3_predictordata, 712
logis_pd, 669	lst_p1k3_setics, 713
logis_pdd, 669	lst_p1k3_waic, 713
logis_waic, 670	_, _ ,
lst_k3_cp, 670	make_cwaic, 716
lst_k3_f1f, 677	make_maic, 717
lst_k3_f1fa, 678	
lst_k3_f1fa, 678 lst_k3_f2f, 679	make_maic, 717
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679	<pre>make_maic,717 make_se,717</pre>
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680	<pre>make_maic, 717 make_se, 717 make_waic, 718</pre>
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_fdd, 680	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_fdd, 680 lst_k3_ldd, 681	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714 makeq, 715 maket0, 715
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_fdd, 681 lst_k3_ldda, 681	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714 makeq, 715 maket0, 715 maketa0, 716
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_fdd, 680 lst_k3_ldd, 681 lst_k3_ldda, 681 lst_k3_lddd, 682	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714 makeq, 715 maket0, 715
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_fdd, 681 lst_k3_ldda, 681	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714 makeq, 715 maket0, 715 maketa0, 716 man, 718 man1f, 728
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_fdd, 680 lst_k3_ldd, 681 lst_k3_ldda, 681 lst_k3_lddd, 682	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714 makeq, 715 maket0, 715 maketa0, 716 man, 718
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_ldd, 681 lst_k3_ldda, 681 lst_k3_lddda, 682 lst_k3_lddda, 682	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714 makeq, 715 maket0, 715 maketa0, 716 man, 718 man1f, 728 man2f, 728 mandsub, 729
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_ldd, 681 lst_k3_ldda, 681 lst_k3_lddd, 682 lst_k3_lddd, 682 lst_k3_lmn, 683	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714 makeq, 715 maket0, 715 maketa0, 716 man, 718 man1f, 728 man2f, 728 mandsub, 729 manf, 729
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_ldd, 681 lst_k3_ldda, 681 lst_k3_lddda, 682 lst_k3_lddda, 682 lst_k3_lmn, 683 lst_k3_lmnp, 683	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714 makeq, 715 maket0, 715 maketa0, 716 man, 718 man1f, 728 man2f, 728 mandsub, 729 manf, 729 man1dd, 736
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_ldd, 681 lst_k3_ldda, 681 lst_k3_lddd, 682 lst_k3_lddda, 682 lst_k3_lmn, 683 lst_k3_lmnp, 683 lst_k3_logf, 684	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714 makeq, 715 maket0, 715 maketa0, 716 man, 718 man1f, 728 man2f, 728 mandsub, 729 manf, 729
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fdd, 680 lst_k3_ldd, 681 lst_k3_ldda, 681 lst_k3_lddd, 682 lst_k3_lddda, 682 lst_k3_lmn, 683 lst_k3_lmnp, 683 lst_k3_logfd, 685	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714 makeq, 715 maket0, 715 maketa0, 716 man, 718 man1f, 728 man2f, 728 mandsub, 729 manf, 729 manldd, 736 manlddd, 736 manlnn, 736
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_fdd, 680 lst_k3_ldd, 681 lst_k3_ldda, 681 lst_k3_lddda, 682 lst_k3_lddda, 682 lst_k3_lmn, 683 lst_k3_lmnp, 683 lst_k3_logf, 684 lst_k3_logfddd, 685 lst_k3_logfddd, 685	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714 makeq, 715 maket0, 715 maketa0, 716 man, 718 man1f, 728 man2f, 728 mandsub, 729 manf, 729 manldd, 736 manlddd, 736
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_fdd, 680 lst_k3_ldd, 681 lst_k3_ldda, 681 lst_k3_lddd, 682 lst_k3_lddda, 682 lst_k3_lmn, 683 lst_k3_lmnp, 683 lst_k3_logf, 684 lst_k3_logfdd, 685 lst_k3_loglik, 686	make_maic, 717 make_se, 717 make_se, 718 makemuhat0, 714 makeq, 715 maket0, 715 maket0, 716 man, 718 man1f, 728 man2f, 728 mandsub, 729 manf, 729 manldd, 736 manlddd, 736 manlnn, 736 manlnn, 737 manlogf, 737
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_fdd, 680 lst_k3_ldd, 681 lst_k3_ldda, 681 lst_k3_lddd, 682 lst_k3_lddda, 682 lst_k3_lmn, 683 lst_k3_lmnp, 683 lst_k3_logfdd, 685 lst_k3_logfddd, 685 lst_k3_logscores, 686	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714 makeq, 715 maket0, 715 maket0, 716 man, 718 man1f, 728 man2f, 728 mandsub, 729 manf, 729 manldd, 736 manlnn, 736 manlnn, 736
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_fdd, 680 lst_k3_ldd, 681 lst_k3_lddd, 682 lst_k3_lddd, 682 lst_k3_lmn, 683 lst_k3_lmnp, 683 lst_k3_logfdd, 685 lst_k3_logfddd, 685 lst_k3_logscores, 686 lst_k3_mu1f, 687	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714 makeq, 715 maket0, 715 maketa0, 716 man, 718 man1f, 728 man2f, 728 mandsub, 729 manf, 729 manldd, 736 manlnn, 736 manlnn, 736 manlnnn, 737 manlogf, 737 manloglik, 737
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_ldd, 681 lst_k3_ldda, 681 lst_k3_lddda, 682 lst_k3_lddda, 682 lst_k3_lmn, 683 lst_k3_lmnp, 683 lst_k3_logfdd, 685 lst_k3_logfddd, 685 lst_k3_loglik, 686 lst_k3_mu1f, 687 lst_k3_mu2f, 687	make_maic, 717 make_se, 717 make_waic, 718 makemuhat0, 714 makeq, 715 maket0, 715 maketa0, 716 man, 718 man1f, 728 man2f, 728 mandsub, 729 manf, 729 manldd, 736 manlnn, 736 manlnnn, 737 manlogf, 737 manlogscores, 738
lst_k3_f1fa, 678 lst_k3_f2f, 679 lst_k3_f2fa, 679 lst_k3_fd, 680 lst_k3_fdd, 680 lst_k3_ldd, 681 lst_k3_ldda, 681 lst_k3_lddda, 682 lst_k3_lddda, 682 lst_k3_lmn, 683 lst_k3_lmnp, 683 lst_k3_logf, 684 lst_k3_logfdd, 685 lst_k3_logfddd, 685 lst_k3_logscores, 686 lst_k3_mu1f, 687 lst_k3_mu2f, 687 lst_k3_p1f, 688	make_maic, 717 make_se, 717 make_se, 718 makemuhat0, 714 makeq, 715 maket0, 715 maketa0, 716 man, 718 man1f, 728 man2f, 728 mandsub, 729 manf, 729 manldd, 736 manlndd, 736 manlnnn, 736 manlnnn, 737 manlogf, 737 manlogscores, 738 manmeans, 738

movexiawayfromzero, 739	norm_p1_1mn, 787
ms_flat_1tail, 740	norm_p1_1mnp, 787
ms_flat_2tail, 741	norm_p1_logf, 788
ms_predictors_1tail, 742	norm_p1_logfdd,789
ms_predictors_2tail, 743	norm_p1_logfddd, 789
	norm_p1_loglik,790
nopdfcdfmsg, 744	norm_p1_logscores, 791
norm_cp, 745	norm_p1_mlparams, 791
norm_dmgs_cp, 750	norm_p1_mu1fa, <mark>792</mark>
norm_dmgs_loglik,756	norm_p1_mu2fa,792
norm_dmgs_logscores, 757	norm_p1_p1fa, 793
norm_dmgs_means, 757	norm_p1_p2fa, <del>794</del>
norm_dmgs_mu1f, 758	norm_p1_pd, 794
norm_dmgs_mu2f, 759	norm_p1_pdd, 795
norm_dmgs_p1f, 759	norm_p1_predictordata, 796
norm_dmgs_p2f, 760	norm_p1_waic, 796
norm_dmgs_waic, 760	norm_p1fa, 772
norm_f1f, 761	norm_p2fa, <del>797</del>
norm_f1fa, 762	norm_pd, 798
norm_f2f, 762	norm_pdd, 798
norm_f2fa, 763	norm_unbiasedv_params, 799
norm_fd, 763	norm_waic, 799
norm_fdd, 764	
norm_gg, 764	pareto_k2_cp, 800
norm_gmn, 765	pareto_k2_f1f, 806
norm_ldd, 765	pareto_k2_f1fa, 807
norm_ldda, 766	pareto_k2_f2f, 807
norm_lddd, 766	pareto_k2_f2fa, 808
norm_lddda, 767	pareto_k2_fd, 808
norm_lmn, 767	pareto_k2_fdd, 809
norm_lmnp, 768	pareto_k2_1111, 809
norm_logf, 769	pareto_k2_ldd, 810
norm_logfdd, 769	pareto_k2_ldda, 811
norm_logfddd, 770	pareto_k2_lddd, 811
norm_logscores, 770	pareto_k2_lddda, 812
norm_ml_params, 771	pareto_k2_logf, 812
norm_mu1fa, 771	pareto_k2_logfdd, 813
norm_mu2fa, 772	pareto_k2_logfddd, 813
norm_p1_cp, 773	pareto_k2_logscores, 814
norm_p1_f1f, 780	pareto_k2_ml_params, 814
norm_p1_f1fa, 781	pareto_k2_mu1fa,815
norm_p1_f2f, 781	pareto_k2_mu2fa, 815
norm_p1_f2fa, 782	pareto_k2_p1fa, <u>816</u>
norm_p1_fd, 783	pareto_k2_p2fa, <u>816</u>
norm_p1_fdd, 783	pareto_k2_pd, 817
norm_p1_1dd, 784	pareto_k2_pdd, 817
norm_p1_1dda, 785	pareto_k2_waic,818
norm_p1_1ddd, 785	pareto_p1k2_cp,818
norm_p1_lddda, 786	pareto_p1k2_f1f, <u>826</u>

pareto_p1k2_f1fa, 827	pgev_p1k3_cp (gev_p1k3_cp), 349
pareto_p1k2_f2f, 827	pgnorm_k3_cp (gnorm_k3_cp), 405
pareto_p1k2_f2fa, 828	pgpd_k1_cp (gpd_k1_cp), 437
pareto_p1k2_fd, 828	pgumbel_cp (gumbel_cp), 462
pareto_p1k2_fdd, 829	pgumbel_p1,848
pareto_p1k2_1dd, 829	pgumbel_p1_cp(gumbel_p1_cp), 481
pareto_p1k2_ldda, 830	phalfnorm_cp (halfnorm_cp), 510
pareto_p1k2_lddd, 831	pinvgamma_cp(invgamma_cp), 529
pareto_p1k2_lddda, 831	pinvgauss_cp (invgauss_cp), 549
pareto_p1k2_lmn, 832	plnorm_cp(lnorm_cp), 570
pareto_p1k2_lmnp, 832	<pre>plnorm_dmgs_cp (lnorm_dmgs_cp), 577</pre>
pareto_p1k2_logf, 833	plnorm_p1, 849
pareto_p1k2_logfdd, 834	plnorm_p1_cp (lnorm_p1_cp), 599
pareto_p1k2_logfddd, 834	plogis_cp(logis_cp), 623
pareto_p1k2_loglik, 835	plogis_p1,849
pareto_p1k2_logscores, 835	plogis_p1_cp(logis_p1_cp),642
pareto_p1k2_means, 836	plst_k3_cp (lst_k3_cp), 670
pareto_p1k2_mu1f, 837	plst_p1k3, 850
pareto_p1k2_mu1fa, 838	plst_p1k3_cp (lst_p1k3_cp), 690
pareto_p1k2_mu2f, 838	pnorm_cp (norm_cp), 745
pareto_p1k2_mu2fa, 839	<pre>pnorm_dmgs_cp (norm_dmgs_cp), 750</pre>
pareto_p1k2_p1f, 839	pnorm_p1, 850
pareto_p1k2_p1fa, 840	pnorm_p1_cp (norm_p1_cp), 773
pareto_p1k2_p2f, 840	pnorm_p1_formula, 851
pareto_p1k2_p2fa, 841	ppareto_k2_cp (pareto_k2_cp), 800
pareto_p1k2_pd, 841	ppareto_p1k2, 851
pareto_p1k2_pdd, 842	ppareto_p1k2_cp (pareto_p1k2_cp), 818
pareto_p1k2_predictordata, 842	punif_cp (unif_cp), 891
pareto_p1k2_waic, 843	punif_formula, 852
pcauchy_cp (cauchy_cp), 28	pweibull_cp (weibull_cp), 897
pcauchy_p1, 844	pweibull_p2, 852
pcauchy_p1_cp (cauchy_p1_cp), 45	<pre>pweibull_p2_cp (weibull_p2_cp), 917</pre>
pexp_cp (exp_cp), 116	qcauchy_cp (cauchy_cp), 28
pexp_p1, 845	qcauchy_cp (cauchy_cp), 28
pexp_p1_cp (exp_p1_cp), 130	qcauchy_p1_cp (cauchy_p1_cp), 45
pfrechet_k1_cp (frechet_k1_cp), 157	qexp_cp (exp_cp), 116
pfrechet_p2k1, 845	qexp_cp (exp_cp), 110 qexp_p1, 853
pfrechet_p2k1_cp (frechet_p2k1_cp), 180	qexp_p1, 055 qexp_p1_cp (exp_p1_cp), 130
pgamma_cp (gamma_cp), 209	qfrechet_k1_cp (frechet_k1_cp), 157
pgev_cp (gev_cp), 230	qfrechet_p2k1, 854
pgev_k3_cp (gev_k3_cp), 245	qfrechet_p2k1_cp (frechet_p2k1_cp), 180
pgev_p1, 846	qgamma_cp (gamma_cp), 209
pgev_p12, 846	qgamma_k1_ppm, 854
pgev_p123, 847	qgamma_ppm, 856
pgev_p123_cp (gev_p123_cp), 277	qgev_cp (gev_cp), 230
pgev_p12_cp (gev_p12_cp), 321	qgev_k12_ppm, 857
pgev_p1_cp (gev_p1_cp), 374	qgev_k3_cp (gev_k3_cp), 245
pgev_p1k3, 848	qgev_mpd_ppm, 858
10 -1	

qgev_p1, 860	reltest2_makeep, 879
qgev_p12, 860	reltest2_plot, 880
qgev_p123, 861	reltest2_predict,881
qgev_p123_cp (gev_p123_cp), 277	reltest2_simulate, 881
qgev_p12_cp (gev_p12_cp), 321	reltest_makeep, 882
qgev_p1_cp (gev_p1_cp), 374	reltest_makemaxep, 883
qgev_p1_ppm, 862	reltest_predict, 883
qgev_p1k3, 862	reltest_simulate, 885
qgev_p1k3_cp (gev_p1k3_cp), 349	rexp_cp (exp_cp), 116
qgev_ppm, 864	rexp_p1_cp (exp_p1_cp), 130
qgnorm_k3_cp (gnorm_k3_cp), 405	rfrechet_k1_cp (frechet_k1_cp), 157
qgpd_k1_cp (gpd_k1_cp), 437	rfrechet_p2k1_cp (frechet_p2k1_cp), 180
qgpd_k1_ppm, 865	rgamma_cp (gamma_cp), 209
qgumbel_cp (gumbel_cp), 462	rgev_cp (gev_cp), 230
qgumbel_p1,866	rgev_k3_cp (gev_k3_cp), 245
qgumbel_p1_cp (gumbel_p1_cp), 481	rgev_minmax, 886
qhalfnorm_cp (halfnorm_cp), 510	rgev_p123_cp (gev_p123_cp), 277
qinvgamma_cp (invgamma_cp), 529	rgev_p123_minmax, 886
qinvgauss_cp (invgauss_cp), 549	rgev_p12_cp (gev_p12_cp), 321
qlnorm_cp (lnorm_cp), 570	rgev_p12_minmax, 887
qlnorm_dmgs_cp(lnorm_dmgs_cp), 577	rgev_p1_cp (gev_p1_cp), 374
qlnorm_p1, 867	rgev_p1_minmax, 888
qlnorm_p1_cp (lnorm_p1_cp), 599	rgev_p1k3_cp (gev_p1k3_cp), 349
qlogis_cp(logis_cp), 623	rgnorm_k3_cp (gnorm_k3_cp), 405
qlogis_p1, 868	rgpd_k1_cp (gpd_k1_cp), 437
qlogis_p1_cp (logis_p1_cp), 642	rgpd_k1_minmax, 889
qlst_k3_cp (lst_k3_cp), 670	rgumbel_cp (gumbel_cp), 462
qlst_p1k3, 868	rgumbel_p1_cp (gumbel_p1_cp), 481
qlst_p1k3_cp (lst_p1k3_cp), 690	rhalfnorm_cp (halfnorm_cp), 510
qnorm_cp (norm_cp), 745	rhp_dmgs_cpmethod, 889
qnorm_dmgs_cp (norm_dmgs_cp), 750	rinvgamma_cp (invgamma_cp), 529
qnorm_p1, 869	rinvgauss_cp (invgauss_cp), 549
qnorm_p1_cp (norm_p1_cp), 773	rlnorm_cp (lnorm_cp), 570
qnorm_p1_formula, 869	rlnorm_dmgs_cp (lnorm_dmgs_cp), 577
qntt_ppm, 870	rlnorm_p1_cp (lnorm_p1_cp), 599
qpareto_k2_cp (pareto_k2_cp), 800	rlogis_cp (logis_cp), 623
qpareto_p1k2, 871	rlogis_p1_cp (logis_p1_cp), 642
qpareto_p1k2_cp (pareto_p1k2_cp), 818	rlst_k3_cp (lst_k3_cp), 670
qunif_cp (unif_cp), 891	rlst_p1k3_cp (lst_p1k3_cp), 690
qunif_formula, 872	rnorm_cp (norm_cp), 745
qweibull_cp (weibull_cp), 897	rnorm_dmgs_cp (norm_dmgs_cp), 750
qweibull_p2,872	rnorm_p1_cp (norm_p1_cp), 773
qweibull_p2_cp (weibull_p2_cp), 917	rpareto_k2_cp (pareto_k2_cp), 800
nearthy on (acroby on) 20	rpareto_p1k2_cp (pareto_p1k2_cp), 818
reauchy_cp (cauchy_cp), 28	runif_cp (unif_cp), 891
<pre>rcauchy_p1_cp (cauchy_p1_cp), 45 reltest, 873</pre>	rust_pumethod, 890
reltest, 875	rweibull_cp (weibull_cp), 897
	rweibull_cp (weibull_cp), 89/ rweibull_p2_cp (weibull_p2_cp), 917
reltest2_cases, 879	r weibuii_pz_cp (weibuii_pz_cp), 91/

tcauchy_cp (cauchy_cp), 28	weibull_logf, 910
tcauchy_p1_cp (cauchy_p1_cp), 45	weibull_logfdd,911
testppm_plot, 890	weibull_logfddd,911
texp_cp (exp_cp), 116	weibull_loglik,912
texp_p1_cp (exp_p1_cp), 130	weibull_logscores, 912
tfrechet_k1_cp (frechet_k1_cp), 157	weibull_means, 913
tfrechet_p2k1_cp (frechet_p2k1_cp), 180	weibull_mu1f, 913
tgamma_cp (gamma_cp), 209	weibull_mu1fa,914
tgev_cp (gev_cp), 230	weibull_mu2f,914
tgev_k3_cp (gev_k3_cp), 245	weibull_mu2fa, 915
tgev_p123_cp (gev_p123_cp), 277	weibull_p1f,915
tgev_p12_cp (gev_p12_cp), 321	weibull_p1fa, 916
tgev_p1_cp (gev_p1_cp), 374	weibull_p2_cp, 917
tgev_p1k3_cp (gev_p1k3_cp), 349	weibull_p2_f1f, 925
tgnorm_k3_cp (gnorm_k3_cp), 405	weibull_p2_f1fa, 926
tgpd_k1_cp (gpd_k1_cp), 437	weibull_p2_f2f, 926
tgumbel_cp (gumbel_cp), 462	weibull_p2_f2fa, 927
tgumbel_p1_cp (gumbel_p1_cp), 481	weibull_p2_fd, 927
thalfnorm_cp (halfnorm_cp), 510	weibull_p2_fdd, 928
tinvgamma_cp (invgamma_cp), 529	weibull_p2_ldd, 928
tinvgauss_cp (invgauss_cp), 549	weibull_p2_ldda, 929
tlnorm_cp (lnorm_cp), 570	weibull_p2_lddd, 930
tlnorm_p1_cp (lnorm_p1_cp), 599	weibull_p2_lddda, 930
tlogis_cp (logis_cp), 623	weibull_p2_lmn, 931
tlogis_p1_cp (logis_p1_cp), 642	weibull_p2_lmnp, 932
tlst_k3_cp (lst_k3_cp), 670	weibull_p2_logf, 932
tlst_p1k3_cp (lst_p1k3_cp), 690	weibull_p2_logfdd, 933
tnorm_cp (norm_cp), 745	weibull_p2_logfddd, 934
tnorm_p1_cp (norm_p1_cp), 773	weibull_p2_loglik, 934
tpareto_k2_cp (pareto_k2_cp), 800	weibull_p2_logscores, 935
tpareto_p1k2_cp (pareto_p1k2_cp), 818	weibull_p2_means, 935
tweibull_cp (weibull_cp), 897	weibull_p2_mu1f, 936
tweibull_p2_cp (weibull_p2_cp), 917	weibull_p2_mu1fa, 937
twelbull_pz_cp (welbull_pz_cp), 917	weibull_p2_mu2f, 937
unif_cp, 891	weibull_p2_mu2fa, 938
шт _ср, 071	weibull_p2_p1f, 938
weibull_cp, 897	weibull_p2_p1fa, 939
weibull_f1f, 904	
weibull_f1fa, 904	weibull_p2_p2f, 940
weibull_f2f, 905	weibull_p2_p2fa, 940 weibull_p2_pd, 941
weibull_f2fa, 905	
weibull_fd, 906	weibull_p2_pdd, 941
weibull_fdd, 906	weibull_p2_predictordata, 942
weibull_ldd, 907	weibull_p2_waic, 943
weibull_ldda, 907	weibull_p2f, 916
weibull_lddd, 908	weibull_p2fa, 917
weibull_lddda, 908	weibull_pd, 944
weibull_lmn, 909	weibull_pdd, 944
	weibull_waic, 945
weibull_lmnp, 909	