homework11

Homework 11

Yi Chen, YC3356

1. Review the following case study, focusing on the model:

```
# load the libraries
library(extraDistr)
library(ggplot2)
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(rstan)
```

```
## Loading required package: StanHeaders
```

```
## rstan (Version 2.18.2, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
##
## Attaching package: 'rstan'
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(bayesplot)
```

```
## This is bayesplot version 1.6.0
```

```
## - Online documentation and vignettes at mc-stan.org/bayesplot
```

```
## - bayesplot theme set to bayesplot::theme_default()
```

```
##    * Does _not_ affect other ggplot2 plots
```

```
##    * See ?bayesplot_theme_set for details on theme setting
```

```
library(loo)
```

```
## This is loo version 2.0.0.9000.
## **NOTE: As of version 2.0.0 loo defaults to 1 core but we recommend using as many as
possible. Use the 'cores' argument or set options(mc.cores = NUM_CORES) for an entire se
ssion. Visit mc-stan.org/loo/news for details on other changes.
```

```
##
## Attaching package: 'loo'
```

```
## The following object is masked from 'package:rstan':
##
##     loo
```

```
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
```

a. Simulate fake data and check that the model recovers the parameters. Feel free to simplify the model as necessary.

```
# first simulate the data which follow the generalized Pareto distribution
u <- 2
k <- 3
sigma <- 3

gpareto = function(n,u,sigma,k){
  y = c()
  for (i in 1:n){
      if (k != 0) {y = c(y,( u + (runif(n=1,min = 0,max = 1)^(-k) -1) * sigma / k))}
    else{y = c(y,(u - sigma*log(runif(n=1,min = 0,max = 1))))}
  }
  return(y)
}
n = 100
y_fake = gpareto(n=n,u=u,k=k,sigma=sigma)
yt<-append(seq(2,3,.01)*30,values = 10)
ds<-list(ymin=u, N=n, y=y_fake, Nt=length(yt), yt=yt)
```

```
writeLines(readLines("gpareto.stan"))
```

```
## Warning in readLines("gpareto.stan"): incomplete final line found on
## 'gpareto.stan'
```

```
## functions {
##   real gpareto_lpdf(vector y, real ymin, real k, real sigma) {
##     // generalised Pareto log pdf
##     int N = rows(y);
##     real inv_k = inv(k);
##     if (k<0 && max(y-ymin)/sigma > -inv_k)
##       reject("k<0 and max(y-ymin)/sigma > -1/k; found k, sigma =", k, sigma)
##     if (sigma<=0)
##       reject("sigma<=0; found sigma =", sigma)
##     if (fabs(k) > 1e-15)
##       return -(1+inv_k)*sum(log1p((y-ymin) * (k/sigma))) -N*log(sigma);
##     else
##       return -sum(y-ymin)/sigma -N*log(sigma); // limit k->0
##   }
##   real gpareto_cdf(vector y, real ymin, real k, real sigma) {
##     // generalised Pareto cdf
##     real inv_k = inv(k);
##     if (k<0 && max(y-ymin)/sigma > -inv_k)
##       reject("k<0 and max(y-ymin)/sigma > -1/k; found k, sigma =", k, sigma)
##     if (sigma<=0)
##       reject("sigma<=0; found sigma =", sigma)
##     if (fabs(k) > 1e-15)
##       return exp(sum(log1m_exp((-inv_k)*(log1p((y-ymin) * (k/sigma))))));
##     else
##       return exp(sum(log1m_exp(-(y-ymin)/sigma))); // limit k->0
##   }
##   real gpareto_lcdf(vector y, real ymin, real k, real sigma) {
##     // generalised Pareto log cdf
##     real inv_k = inv(k);
##     if (k<0 && max(y-ymin)/sigma > -inv_k)
##       reject("k<0 and max(y-ymin)/sigma > -1/k; found k, sigma =", k, sigma)
##     if (sigma<=0)
##       reject("sigma<=0; found sigma =", sigma)
##     if (fabs(k) > 1e-15)
##       return sum(log1m_exp((-inv_k)*(log1p((y-ymin) * (k/sigma)))));
##     else
##       return sum(log1m_exp(-(y-ymin)/sigma)); // limit k->0
##   }
##   real gpareto_lccdf(vector y, real ymin, real k, real sigma) {
##     // generalised Pareto log ccdf
##     real inv_k = inv(k);
##     if (k<0 && max(y-ymin)/sigma > -inv_k)
##       reject("k<0 and max(y-ymin)/sigma > -1/k; found k, sigma =", k, sigma)
##     if (sigma<=0)
##       reject("sigma<=0; found sigma =", sigma)
##     if (fabs(k) > 1e-15)
##       return (-inv_k)*sum(log1p((y-ymin) * (k/sigma)));
##     else
##       return -sum(y-ymin)/sigma; // limit k->0
##   }
##   real gpareto_rng(real ymin, real k, real sigma) {
##     // generalised Pareto rng
##     if (sigma<=0)
```

```
##       reject("sigma<=0; found sigma =", sigma)
##     if (fabs(k) > 1e-15)
##       return ymin + (uniform_rng(0,1)^-k -1) * sigma / k;
##     else
##       return ymin - sigma*log(uniform_rng(0,1)); // limit k->0
##   }
## }
## data {
##   real ymin;
##   int<lower=0> N;
##   vector<lower=ymin>[N] y;
##   int<lower=0> Nt;
##   vector<lower=ymin>[Nt] yt;
## }
## transformed data {
##   real ymax = max(y);
## }
## parameters {
##   real<lower=0> sigma;
##   real<lower=-sigma/(ymax-ymin)> k;
## }
## model {
##   y ~ gpareto(ymin, k, sigma);
## }
## generated quantities {
##   vector[N] log_lik;
##   vector[N] yrep;
##   vector[Nt] predccdf;
##   for (n in 1:N) {
##     log_lik[n] = gpareto_lpdf(rep_vector(y[n],1) | ymin, k, sigma);
##     yrep[n] = gpareto_rng(ymin, k, sigma);
##   }
##   for (nt in 1:Nt)
##     predccdf[nt] = exp(gpareto_lccdf(rep_vector(yt[nt],1) | ymin, k, sigma));
## }
```

```
fake_gpd <- stan_model('gpareto.stan')
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on '/
## Users/yi/Desktop/study/subjects/bayesian-data-analysis/homework/homework16/
## gpareto.stan'
```

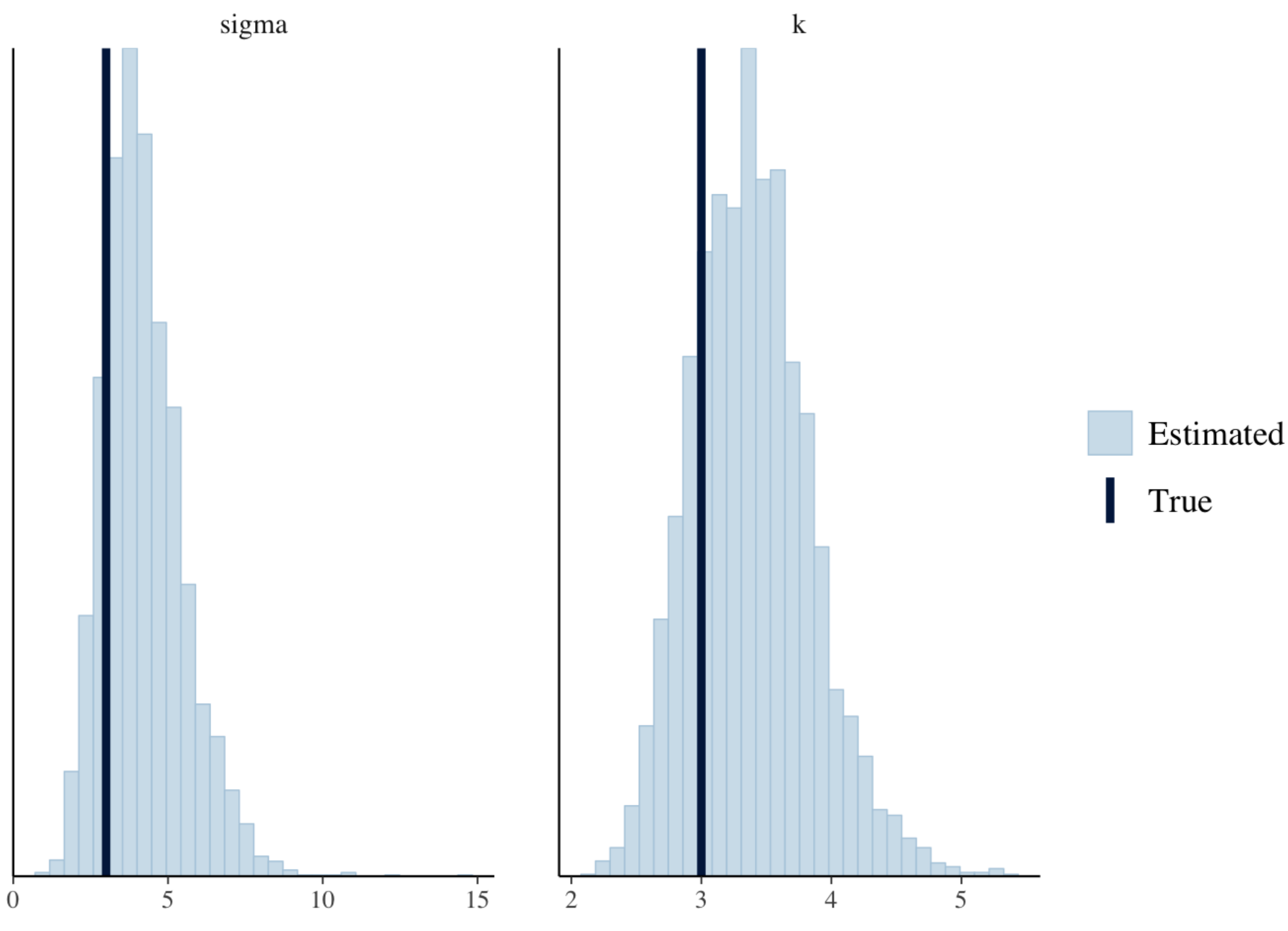
```
fake_fit <- sampling(fake_gpd, data=ds)

posterior_sigma_k <- as.matrix(fake_fit, pars = c('sigma','k'))
head(posterior_sigma_k)
```

```
##            parameters
## iterations    sigma        k
##       [1,] 4.974953 3.024038
##       [2,] 3.232378 3.879078
##       [3,] 2.770637 3.070047
##       [4,] 5.769755 3.625278
##       [5,] 3.560895 3.751890
##       [6,] 4.020748 3.443346
```

```
true_sigma_k <- c(sigma, k)
mcmc_recover_hist(posterior_sigma_k, true = true_sigma_k)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



As we can see the MCMC can give us a good fit wit the true variables.

b. In two or three sentences, discuss the strengths and weaknesses of the model. How might the model be expanded?

The tutorial is very clear and well designed. One improvement i want to have a try is to do the extreme value analysis beyond the generalized Pareto distribution (GPD).

There is a distribution called generalized extreme value (GEV) distribution which is a family of continuous probability distributions developed within extreme value theory. To see more in https://en.wikipedia.org/wiki/Generalized_extreme_value_distribution (https://en.wikipedia.org/wiki/Generalized_extreme_value_distribution).

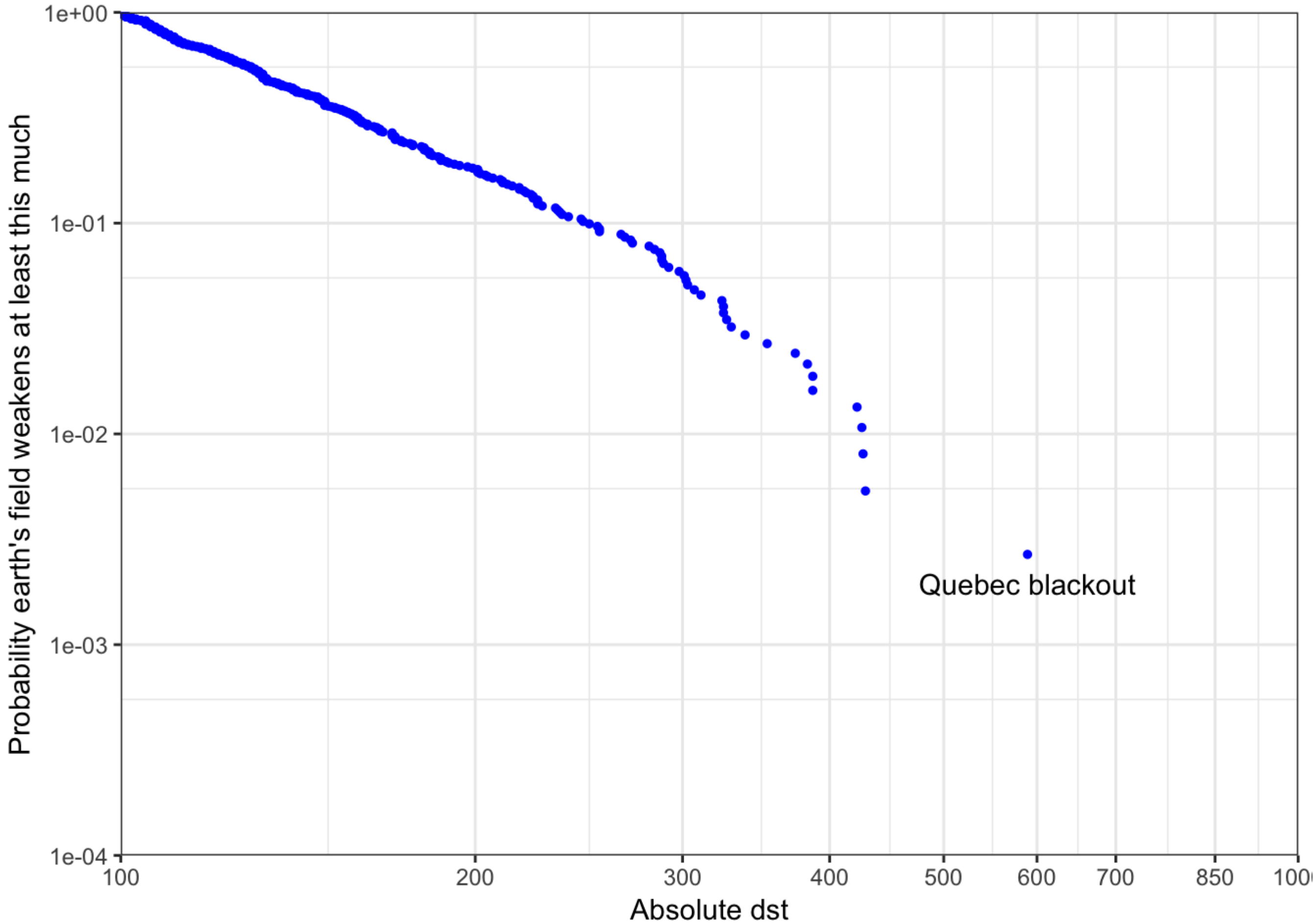
2.

a. Fit the model to the real data and perform model checking and/or validation (Chapters 6 and 7 of BDA). Data can be found at:

```
# file preview shows a header row
d <- read.csv("geomagnetic_tail_data.csv", header = FALSE)
# dst are the absolute magnitudes
colnames(d) <- "dst"
d <- d %>% mutate(dst = abs(dst)) %>% arrange(dst)
n <- dim(d)[1]
d$ccdf <- seq(n,1,-1)/n
head(d)
```

```
##   dst      ccdf
## 1 100 1.0000000
## 2 100 0.9973190
## 3 100 0.9946381
## 4 100 0.9919571
## 5 100 0.9892761
## 6 100 0.9865952
```

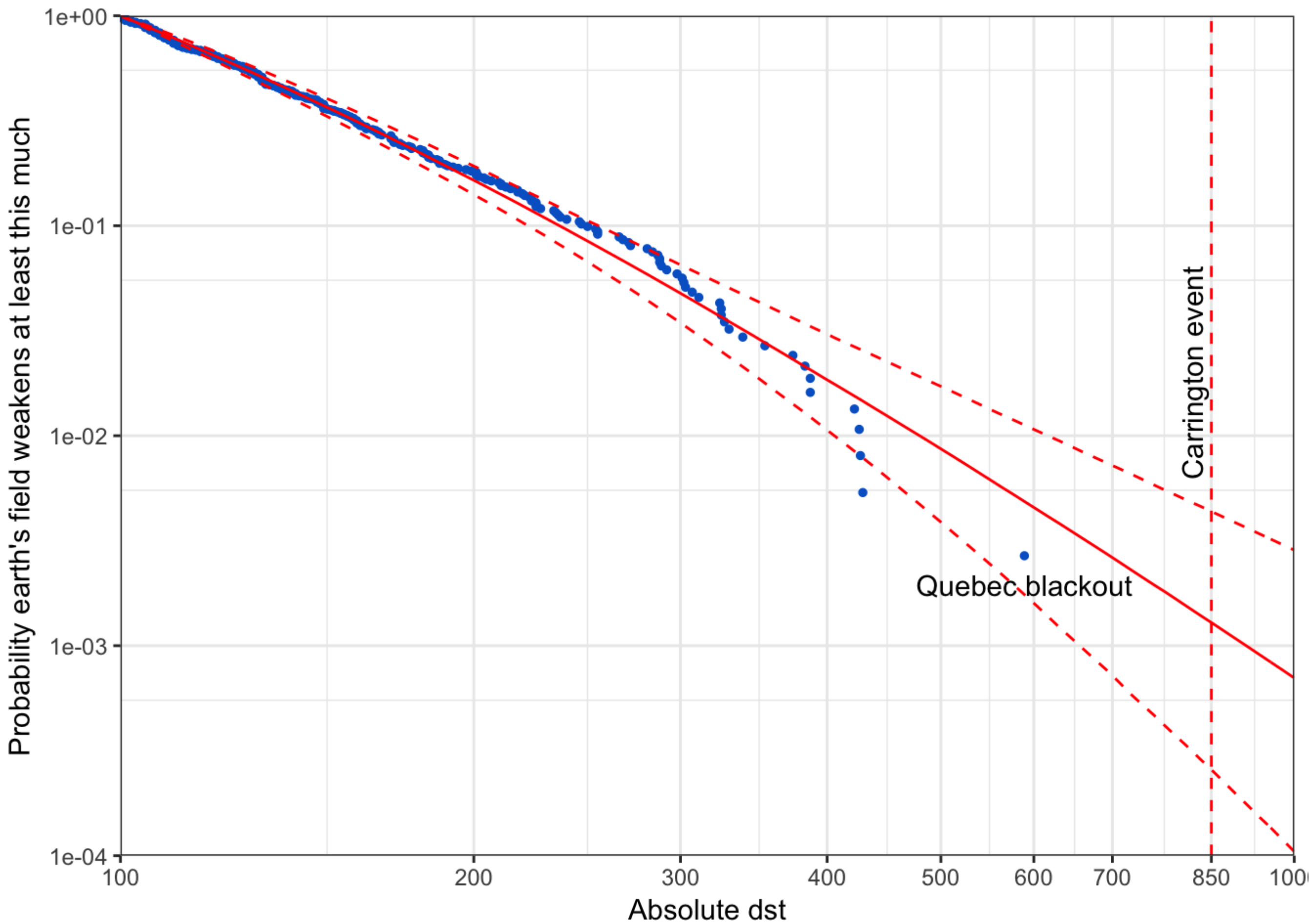
```
ggplot() +
  geom_point(aes(dst, ccdf), data = d, size = 1, colour = "blue") +
  coord_trans(x="log10", y="log10", limx=c(100,1000), limy=c(1e-4,1)) +
  scale_y_continuous(breaks=c(1e-5,1e-4,1e-3,1e-2,1e-1,1), limits=c(1e-4,1)) +
  scale_x_continuous(breaks=c(100,200,300,400,500,600,700,850,1000), limits=c(100,1000))
 +
  labs(y = "Probability earth's field weakens at least this much", x= "Absolute dst") +
  geom_text(aes(x = d$dst[n], y = d$ccdf[n]),
            label = "Quebec blackout", vjust="top", nudge_y=-0.0005) +
  guides(linetype = F) +
  theme_bw()
```




```
yt<-append(10^seq(2,3,.01),850)
ds<-list(ymin=100, N=n, y=d$dst, Nt=length(yt), yt=yt)
fit_gpd <- stan(file='gpareto.stan', data=ds, refresh=0,chains=4, seed=100)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on '/
## Users/yi/Desktop/study/subjects/bayesian-data-analysis/homework/homework16/
## gpareto.stan'
```

```
gpd_params <- rstan::extract(fit_gpd)
mu <- apply(t(gpd_params$predccdf), 1, quantile, c(0.05, 0.5, 0.95)) %>% t() %>% data.fr
ame(x = yt, .) %>% gather(pct, y, -x)
clrs <- color_scheme_get("brightblue")
ggplot() +
  geom_point(aes(dst, ccdf), data = d, size = 1, color = clrs[[5]]) +
  geom_line(aes(x=c(850,850),y=c(1e-4,1)),linetype="dashed",color="red") +
  geom_line(aes(x, y, linetype = pct), data = mu, color = 'red') +
  scale_linetype_manual(values = c(2,1,2)) +
  coord_trans(x="log10", y="log10", limx=c(100,1000), limy=c(1e-4,1)) +
  scale_y_continuous(breaks=c(1e-5,1e-4,1e-3,1e-2,1e-1,1), limits=c(1e-4,1)) +
  scale_x_continuous(breaks=c(100,200,300,400,500,600,700,850,1000), limits=c(100,1000))
 +
  geom_text(aes(x = d$dst[n], y = d$ccdf[n]), label = "Quebec blackout", vjust="top", nu
dge_y=-0.0005) +
  geom_text(aes(x = 820, y = 0.02), label = "Carrington event", angle=90) +
  labs(y = "Probability earth's field weakens at least this much", x= "Absolute dst") +
  guides(linetype = F) +
  theme_bw()
```




```
ppc1 <- ppc_dens_overlay(log(d$dst),log(gpd_params$yrep[1:50,])) + labs(x="log(absolute
 dst)")
ppc2 <- ppc_stat(log(d$dst), log(gpd_params$yrep), stat = "max") + labs(x="max(log(absol
ute dst))")
psis <- psislw(-gpd_params$log_lik)
```

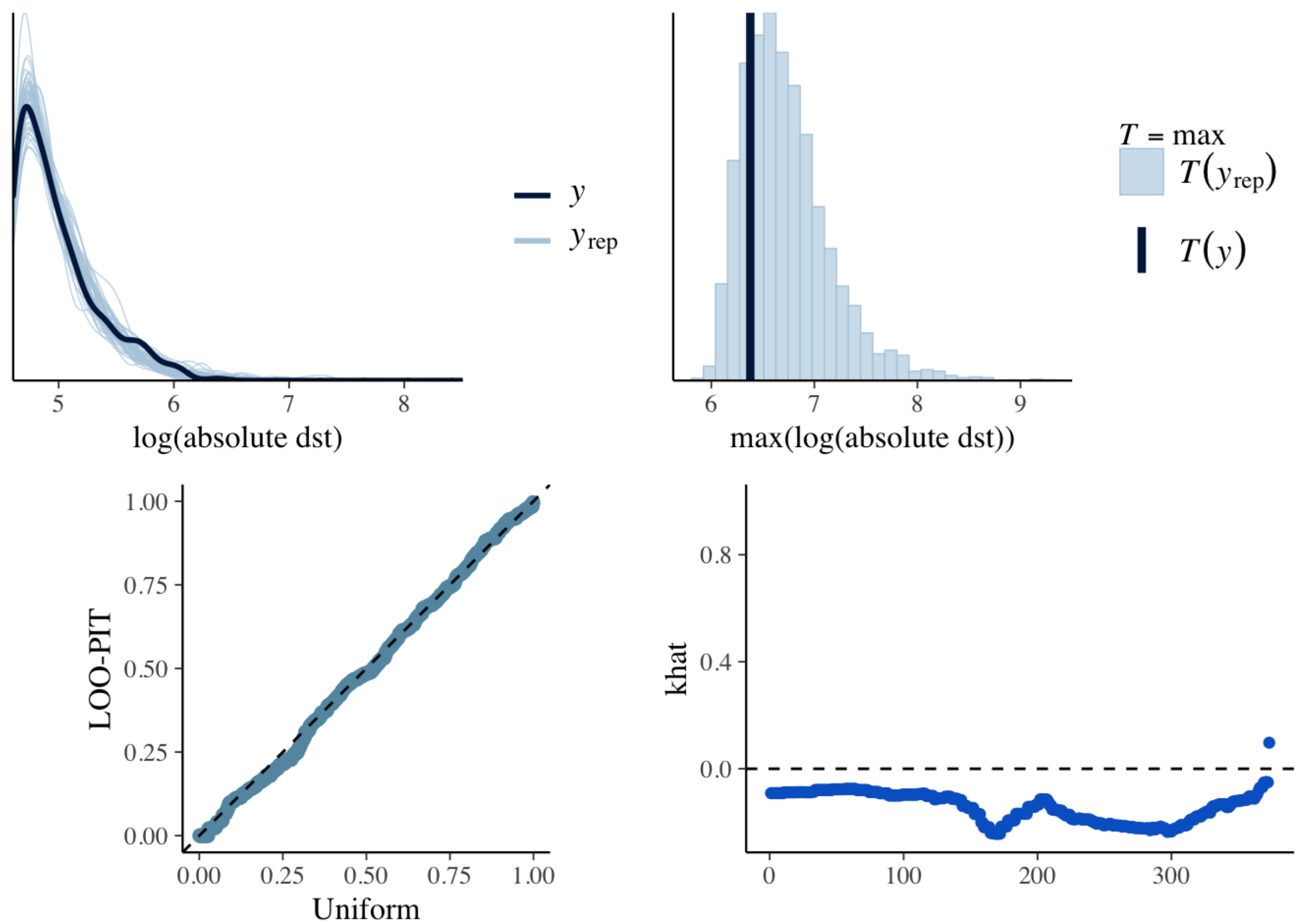
```
## Warning: 'psislw' is deprecated.
## Use 'psis' instead.
## See help("Deprecated")
```

```
clrs <- color_scheme_get("brightblue")
pkhats <- ggplot() + geom_point(aes(x=seq(1,n),y=psis$pareto_k), color=clrs[[5]]) + labs
(y="khat", x="") +
  geom_hline(yintercept=0, linetype="dashed") + ylim(-0.25,1) + theme_default()
ppc3 <- ppc_loo_pit(log(d$dst), log(gpd_params$yrep), lw=psis$lw_smooth)
```

```
## Warning: 'ppc_loo_pit' is deprecated.
## Use 'ppc_loo_pit_qq or ppc_loo_pit_overlay' instead.
## See help("Deprecated")
```

```
grid.arrange(ppc1,ppc2,ppc3,pkhats,ncol=2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



d. Expand the model as discussed in 1.b./class and interpret the results.

In this part, I will follow the tutorial and redo the whole things wit the generalized extreme value (GEV) distribution.

$$p(x|\mu, \sigma, \xi) = \frac{1}{\sigma} t(x)^{\xi+1} e^{-t(x)}$$

Where

$$t(x) = (1 + \xi(\frac{x-\mu}{\sigma})^{-\frac{-1}{\sigma}})I(\xi \neq 0) + e^{-\frac{(x-\mu)}{\sigma}} I(I(\xi = 0))$$

And:$x \in [\mu - \sigma/\xi, +\infty]$ for $\xi > 0$, $x \in [-\infty, +\infty]$ for $\xi = 0$, $x \in [-\infty, \mu - \sigma/\xi]$ for $\xi > 0$, and $\sigma > 0$

```
writeLines(readLines("gev.stan"))
```

```
## Warning in readLines("gev.stan"): incomplete final line found on 'gev.stan'
```

```
## functions {
##   real gev_lpdf(vector y, real mu, real xi, real sigma) {
##     // generalised gev log pdf
##     int N = rows(y);
##     real inv_xi = inv(xi);
##     real inv_sigma = inv(sigma);
##     vector[N] lpdf_y;
##
##     if (xi > 0 && min(y) < mu-sigma/xi){
##       reject("xi > 0 && min(y) < mu-sigma/xi, found mu, xi, sigma = ", mu, xi, sigm
a)}
##     if (xi < 0 && max(y) > mu-sigma/xi){
##       reject("xi < 0 && max(y) > mu-sigma/xi, found mu, xi, sigma = ", mu, xi, sigm
a)}
##     if (sigma<=0){
##       reject("sigma<=0; found sigma =", sigma)}
##
##     if (fabs(mu) > 1e-15){
##       for (i in 1:N){
##         lpdf_y[i] = (xi+1) * (-inv_xi) * log( 1+xi*((y[i]-mu)*inv_sigma)) -  log(sigm
a) -  (1+xi*((y[i]-mu)*inv_sigma)) ^ -inv_xi ;
##       }
##       return(sum(lpdf_y));}
##     else{
##       for (i in 1:N){
##         lpdf_y[i] = (xi+1)* inv_sigma * (y[i]-mu) -  log(sigma) - exp((y[i]-mu)*(-inv
_sigma));
##       }
##       return(sum(lpdf_y));}
##   }
##
##   real gev_rng(real mu, real xi, real sigma) {
##     // generalised Pareto rng
##     real inv_xi = inv(xi);
##     real inv_sigma = inv(sigma);
##
##     if (sigma<=0){
##       reject("sigma<=0; found sigma =", sigma)}
##
##     if (fabs(mu) > 1e-15){
##          return( mu + sigma / xi * ((-log(uniform_rng(0,1)))^(-xi) - 1));}
##     else{
##       return( mu - log(-sigma * log(uniform_rng(0,1))) );}
## }}
##
## data {
##   real xi;
##   int<lower=0> N;
##   vector[N] y;
## }
## parameters {
##   real<lower=0> sigma;
##   real mu;
```

```
## }
## model {
##   y ~ gev(xi, mu, sigma);
## }
## generated quantities {
##   vector[N] yrep;
##   for (n in 1:N) {
##     yrep[n] = gev_rng(xi, mu, sigma);
##   }
## }
```

```
xi = rnorm(1,0,10)
ds<-list(xi=xi, N=length(d$dst), y=d$dst)
fit_gev <- stan(file='gev.stan', data=ds)
```

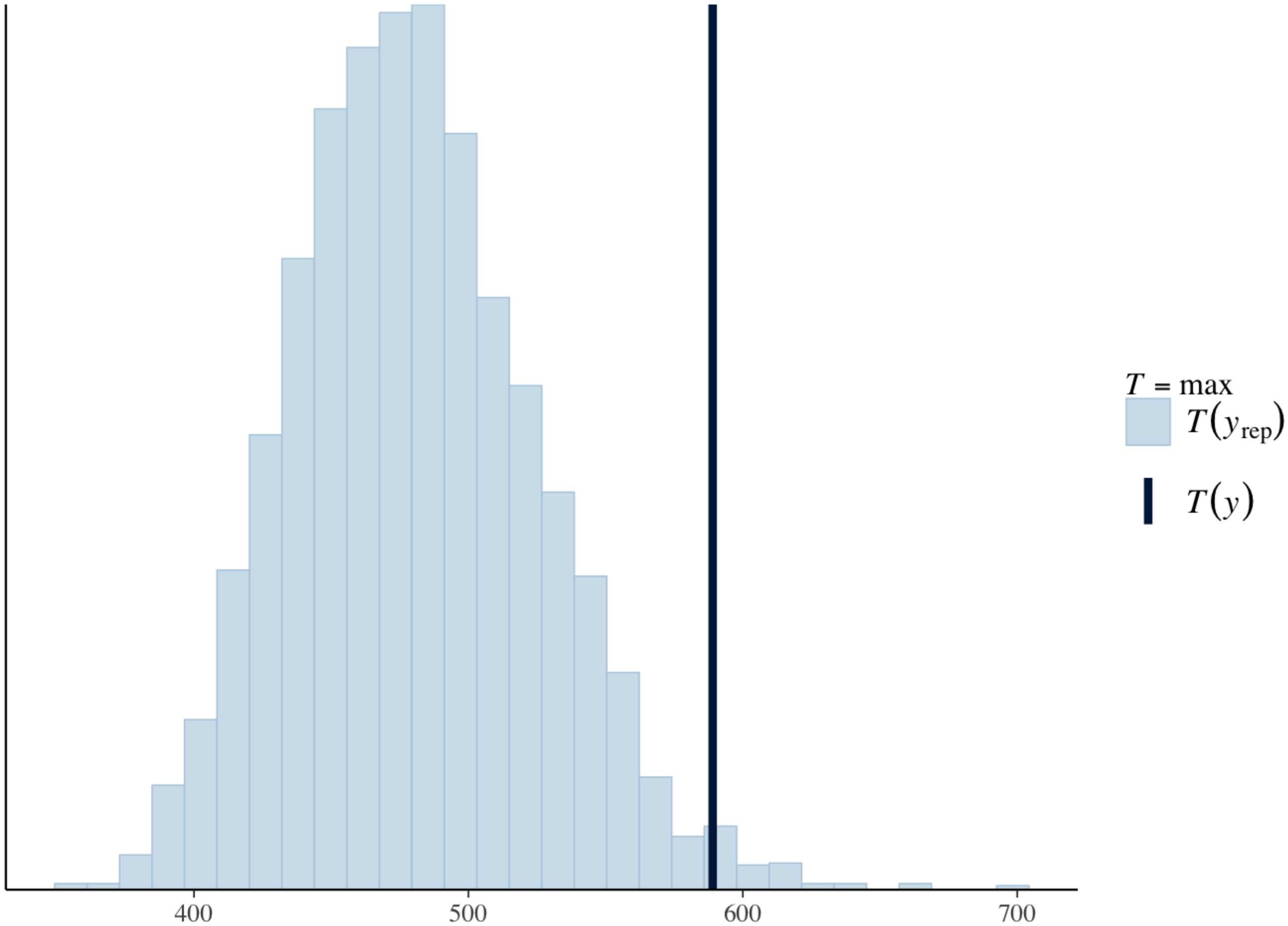
```
## Warning in readLines(file, warn = TRUE): incomplete final line found on '/
## Users/yi/Desktop/study/subjects/bayesian-data-analysis/homework/homework16/
## gev.stan'
```

```
## Warning: There were 204 divergent transitions after warmup. Increasing adapt_delta ab
ove 0.8 may help. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
gev_params <- rstan::extract(fit_gev)
y_rep <- as.matrix(fit_gev, pars = "yrep")
ppc_stat(y = ds$y, yrep = y_rep, stat = "max")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


400
500
600
700
T = max
T(y_rep)
T(y)