VAST

James T. Thorson

```r
library(tinyVAST)
library(fmesher)
set.seed(101)
```

tinyVAST is an R package for fitting vector autoregressive spatio-temporal (VAST) models. We here explore the capacity to specify the vector-autoregressive spatio-temporal component.

Spatio-temporal autoregressive model

We first explore the ability to specify a first-order autoregressive spatio-temporal process:

```r
# Simulate settings
theta_xy = 0.4
n_x = n_y = 10
n_t = 15
rho = 0.8
spatial_sd = 0.5

# Simulate GMRFs
R = exp(-theta_xy * abs(outer(1:n_x, 1:n_y, FUN="-")) )
d = mvtnorm::rmvnorm(n_t, sigma=spatial_sd^2*kronecker(R,R) )

# Project through time and add mean
for( t in seq_len(n_t) ){
  if(t>1) d[t,] = rho*d[t-1,] + d[t,]
}
#d = d + 0.5

# Shape into longform data-frame and add error
Data = data.frame( expand.grid(time=1:n_t, x=1:n_x, y=1:n_y), "var"="logn", z=exp(as.vector(d)))
Data$n = tweedie::rtweedie( n=nrow(Data), mu=Data$z, phi=0.5, power=1.5 )
mean(Data$n==0)
#> [1] 0.046

# make mesh
mesh = fm_mesh_2d( Data[,c('x','y')] )

# fit model
out = fit( sem = "logn -> logn, 1, rho",
           data = Data,
           formula = n ~ 0 + factor(time),
           spatial_graph = mesh,
```

```
          quiet = TRUE,
          family_link = c(1,0) )
#>   0:     2073.3611:  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000
#>   1:     1863.3871: -0.113813 -0.00198865 -0.00237104 -0.00163534 -0.00219757 -0.00493548 -0.0027809
#>   2:     1793.7525: -0.0814791 -0.00358532 -0.00435124 -0.00246101 -0.00331800 -0.0101719 -0.0039766
#>   3:     1755.7821: -0.0900362 -0.00575196 -0.00688912 -0.00263295 -0.00352460 -0.0170919 -0.0031083
#>   4:     1737.9714: -0.0762625 -0.00636682 -0.00783814 -0.00225541 -0.00277733 -0.0204192 -0.0014063
#>   5:     1729.4793: -0.0656585 -0.00778650 -0.0100780 -0.000860143 -0.000620949 -0.0278780 0.0035305
#>   6:     1727.9143: -0.0617113 -0.00827491 -0.0109572 -0.000175085 0.000588884 -0.0311455 0.00608982
#>   7:     1726.9417: -0.0612841 -0.00908388 -0.0124292 0.00107162 0.00267368 -0.0365291 0.0105268 -0.
#>   8:     1725.5892: -0.0601763 -0.0116729 -0.0172487 0.00552869 0.0105070 -0.0551780 0.0267589 -0.01
#>   9:     1723.7397: -0.0577084 -0.0145978 -0.0220312 0.0100644 0.0179209 -0.0732593 0.0422878 -0.015
#>  10:     1722.5938: -0.0333077 -0.0179773 -0.0260605 0.0144109 0.0231775 -0.0869391 0.0536038 -0.020
#>  11:     1721.7358: -0.0132876 -0.0212181 -0.0294696 0.0181897 0.0273172 -0.0978914 0.0619804 -0.026
#>  12:     1720.9722: -0.00214397 -0.0245708 -0.0317083 0.0206674 0.0287638 -0.102993 0.0619434 -0.041
#>  13:     1720.7991: 0.000352057 -0.0249559 -0.0322316 0.0210639 0.0294085 -0.104873 0.0629282 -0.042
#>  14:     1720.4747: 0.00311944 -0.0258865 -0.0330124 0.0214830 0.0297184 -0.106933 0.0615628 -0.0489
#>  15:     1720.2531: 0.00750308 -0.0269668 -0.0339877 0.0220303 0.0302277 -0.109863 0.0603233 -0.0557
#>  16:     1719.9192: 0.0155160 -0.0289930 -0.0359197 0.0226795 0.0303984 -0.116131 0.0551075 -0.07200
#>  17:     1719.3762: 0.0257196 -0.0311510 -0.0395415 0.0198321 0.0242499 -0.134130 0.0284519 -0.11344
#>  18:     1718.9935: 0.0352096 -0.0338808 -0.0447514 0.0140675 0.0133552 -0.164691 0.00542241 -0.1478
#>  19:     1718.8980: 0.0364787 -0.0342685 -0.0452875 0.0141355 0.0133479 -0.166205 0.00570269 -0.1488
#>  20:     1718.8227: 0.0403867 -0.0354448 -0.0469809 0.0114645 0.00853938 -0.170227 0.00139906 -0.153
#>  21:     1718.7280: 0.0423465 -0.0368725 -0.0493174 0.00757024 0.00224595 -0.175191 -0.00503610 -0.1
#>  22:     1718.6195: 0.0410277 -0.0395245 -0.0547515 -6.29451e-05 -0.00625689 -0.187415 -0.0151756 -0
#>  23:     1718.4542: 0.0587225 -0.0452497 -0.0667977 -0.0137996 -0.0206808 -0.213555 -0.0390978 -0.20
#>  24:     1718.4387: 0.0575309 -0.0454635 -0.0670464 -0.0137462 -0.0207290 -0.214371 -0.0390043 -0.20
#>  25:     1718.3985: 0.0574713 -0.0460922 -0.0678137 -0.0144454 -0.0223051 -0.215952 -0.0411122 -0.20
#>  26:     1718.3557: 0.0565374 -0.0474660 -0.0694907 -0.0159961 -0.0256582 -0.219609 -0.0455441 -0.21
#>  27:     1718.2836: 0.0573342 -0.0499465 -0.0728392 -0.0214015 -0.0344212 -0.227001 -0.0578151 -0.22
#>  28:     1718.1840: 0.0577220 -0.0507005 -0.0797803 -0.0371991 -0.0466712 -0.248912 -0.0763027 -0.24
#>  29:     1718.0683: 0.0620151 -0.0640436 -0.100102 -0.0434166 -0.0861373 -0.281450 -0.117066 -0.2730
#>  30:     1718.0345: 0.0622371 -0.0640631 -0.100050 -0.0437044 -0.0858448 -0.281748 -0.116858 -0.2735
#>  31:     1718.0080: 0.0622243 -0.0642106 -0.0997824 -0.0454051 -0.0840906 -0.283579 -0.115592 -0.276
#>  32:     1717.9646: 0.0632308 -0.0699573 -0.0988202 -0.0496147 -0.0869135 -0.281107 -0.120288 -0.282
#>  33:     1717.9095: 0.0651749 -0.0637969 -0.110497 -0.0532064 -0.0820910 -0.288906 -0.121670 -0.3045
#>  34:     1717.8392: 0.0671192 -0.0665845 -0.109121 -0.0546113 -0.0864631 -0.304234 -0.133371 -0.3125
#>  35:     1717.8347: 0.0670841 -0.0666504 -0.109008 -0.0548133 -0.0864533 -0.304341 -0.133378 -0.3128
#>  36:     1717.8314: 0.0671896 -0.0667717 -0.108789 -0.0552069 -0.0864377 -0.304544 -0.133402 -0.3135
#>  37:     1717.8260: 0.0671935 -0.0671239 -0.108245 -0.0560191 -0.0869100 -0.305057 -0.133946 -0.3150
#>  38:     1717.8146: 0.0672817 -0.0678775 -0.107214 -0.0575812 -0.0885730 -0.306164 -0.135868 -0.3182
#>  39:     1717.7846: 0.0677928 -0.0697625 -0.106783 -0.0641197 -0.0959688 -0.310644 -0.146640 -0.3325
#>  40:     1717.7608: 0.0694394 -0.0757866 -0.111809 -0.0711160 -0.102011 -0.322443 -0.159549 -0.34180
#>  41:     1717.7510: 0.0675212 -0.0712584 -0.118159 -0.0699104 -0.116558 -0.338783 -0.164038 -0.34716
#>  42:     1717.7410: 0.0696295 -0.0692434 -0.111686 -0.0871603 -0.122675 -0.351319 -0.172600 -0.35389
#>  43:     1717.7242: 0.0703921 -0.0722263 -0.121060 -0.0920141 -0.129446 -0.352798 -0.183456 -0.37165
#>  44:     1717.7187: 0.0708326 -0.0788799 -0.135589 -0.0920521 -0.126015 -0.352718 -0.191920 -0.38993
#>  45:     1717.7111: 0.0704229 -0.0960536 -0.135560 -0.0900449 -0.118497 -0.355459 -0.195275 -0.39051
#>  46:     1717.7038: 0.0748635 -0.0809841 -0.128889 -0.0951938 -0.127564 -0.349972 -0.191953 -0.39118
#>  47:     1717.7014: 0.0747587 -0.0809363 -0.128896 -0.0951169 -0.127548 -0.350413 -0.192018 -0.39143
#>  48:     1717.7010: 0.0743590 -0.0805616 -0.128674 -0.0950703 -0.127433 -0.351302 -0.192146 -0.39195
#>  49:     1717.6989: 0.0739897 -0.0800096 -0.128183 -0.0950033 -0.127368 -0.352408 -0.192387 -0.39252
#>  50:     1717.6974: 0.0732549 -0.0789980 -0.127144 -0.0946046 -0.127753 -0.354799 -0.193818 -0.39373
```

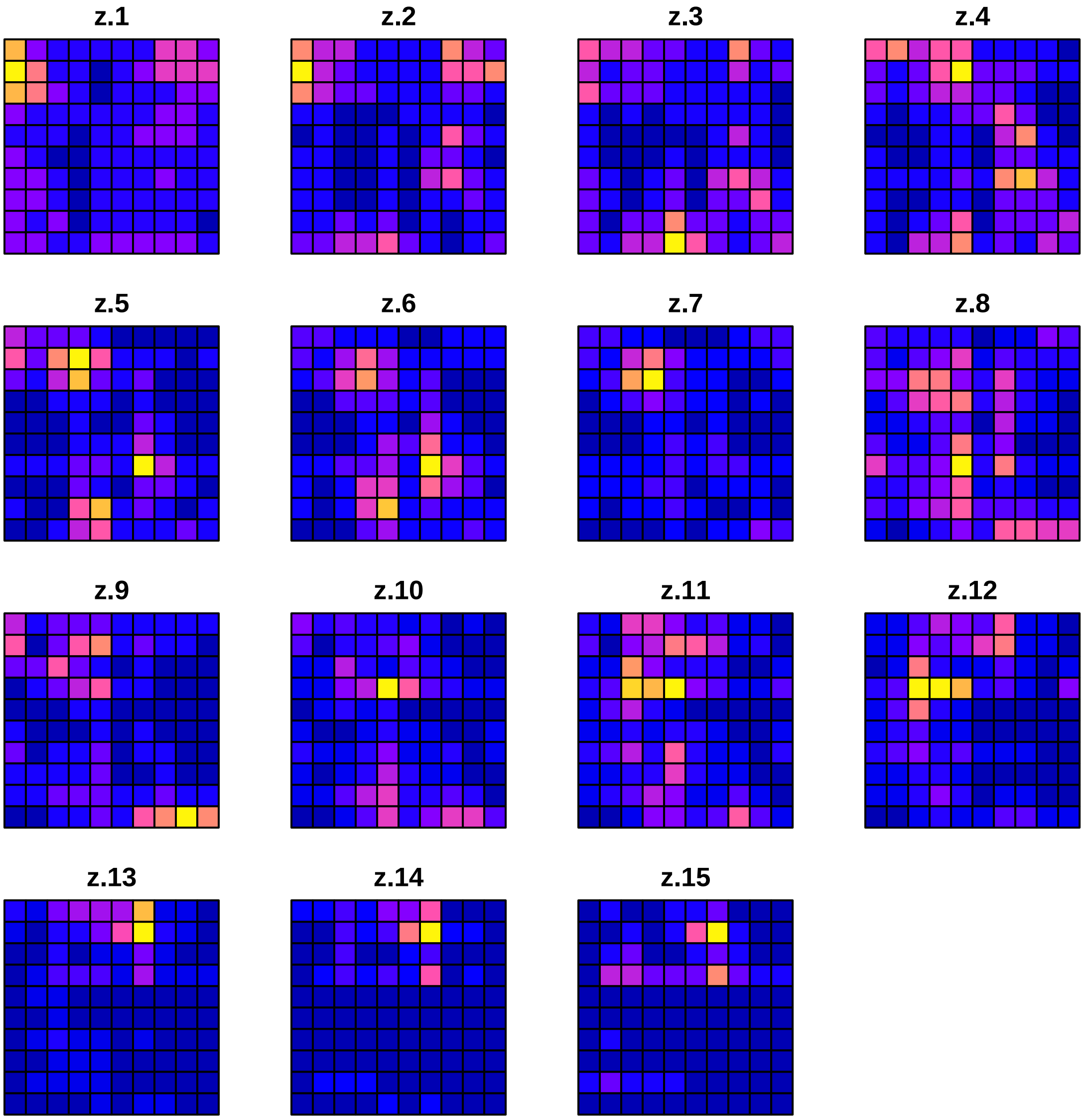
```
#>  51:     1717.6959: 0.0748149 -0.0787190 -0.126606 -0.0944231 -0.130784 -0.357081 -0.199624 -0.39560
#>  52:     1717.6936: 0.0735730 -0.0785016 -0.127773 -0.0975561 -0.132428 -0.361463 -0.200386 -0.39940
#>  53:     1717.6935: 0.0734954 -0.0785387 -0.127815 -0.0975146 -0.132449 -0.361554 -0.200414 -0.39944
#>  54:     1717.6934: 0.0734198 -0.0785848 -0.127866 -0.0974645 -0.132475 -0.361668 -0.200452 -0.39949
#>  55:     1717.6932: 0.0731436 -0.0786748 -0.128136 -0.0976506 -0.132647 -0.361848 -0.200653 -0.39952
#>  56:     1717.6929: 0.0727772 -0.0789164 -0.128739 -0.0982031 -0.133102 -0.362198 -0.201079 -0.39964
#>  57:     1717.6923: 0.0743471 -0.0809257 -0.130526 -0.0999243 -0.135409 -0.364087 -0.201897 -0.40167
#>  58:     1717.6920: 0.0734358 -0.0809400 -0.130761 -0.0996010 -0.135441 -0.364839 -0.202363 -0.40181
#>  59:     1717.6916: 0.0733511 -0.0810622 -0.131205 -0.100074 -0.135875 -0.365466 -0.203185 -0.402302
#>  60:     1717.6913: 0.0728640 -0.0814475 -0.131956 -0.100965 -0.136906 -0.366615 -0.204739 -0.403458
#>  61:     1717.6907: 0.0728927 -0.0816881 -0.131633 -0.100944 -0.137675 -0.367888 -0.205881 -0.404923
#>  62:     1717.6905: 0.0733247 -0.0808373 -0.132580 -0.101508 -0.136807 -0.368902 -0.207730 -0.405440
#>  63:     1717.6901: 0.0725516 -0.0814993 -0.132473 -0.101728 -0.138189 -0.370023 -0.208416 -0.406933
#>  64:     1717.6900: 0.0738512 -0.0821143 -0.132871 -0.101947 -0.140472 -0.373125 -0.209550 -0.408508
#>  65:     1717.6899: 0.0733622 -0.0822918 -0.133101 -0.102495 -0.141414 -0.373832 -0.211218 -0.410155
#>  66:     1717.6896: 0.0727349 -0.0826902 -0.133267 -0.103164 -0.142609 -0.374882 -0.212860 -0.411873
#>  67:     1717.6893: 0.0725131 -0.0831413 -0.134469 -0.104566 -0.143270 -0.375496 -0.213029 -0.411385
#>  68:     1717.6892: 0.0724159 -0.0825053 -0.133539 -0.103249 -0.142744 -0.375764 -0.213971 -0.413071
#>  69:     1717.6891: 0.0722461 -0.0828962 -0.134819 -0.104939 -0.144248 -0.376883 -0.215297 -0.414214
#>  70:     1717.6890: 0.0722595 -0.0830215 -0.134835 -0.105207 -0.144416 -0.377559 -0.215760 -0.414440
#>  71:     1717.6890: 0.0722578 -0.0831402 -0.134969 -0.105384 -0.144683 -0.378002 -0.216013 -0.414579
#>  72:     1717.6890: 0.0722600 -0.0831690 -0.135322 -0.105535 -0.144749 -0.377917 -0.216129 -0.414680
#>  73:     1717.6890: 0.0722920 -0.0831689 -0.135486 -0.105590 -0.144832 -0.378044 -0.216218 -0.414777
#>  74:     1717.6890: 0.0722801 -0.0832268 -0.135492 -0.105757 -0.144952 -0.378148 -0.216252 -0.414888
#>  75:     1717.6890: 0.0722961 -0.0832402 -0.135469 -0.105823 -0.144984 -0.378238 -0.216324 -0.414892
#>  76:     1717.6890: 0.0722854 -0.0832360 -0.135491 -0.105792 -0.144991 -0.378239 -0.216333 -0.414900
#>  77:     1717.6890: 0.0722854 -0.0832360 -0.135491 -0.105792 -0.144991 -0.378239 -0.216333 -0.414900
out
#> $call
#> fit(data = Data, formula = n ~ 0 + factor(time), sem = "logn -> logn, 1, rho",
#>     family_link = c(1, 0), spatial_graph = mesh, quiet = TRUE)
#>
#> $opt
#> $opt$par
#>   log_kappa     alpha_j     alpha_j     alpha_j     alpha_j     alpha_j     alpha_j     alpha_j     a
#>  0.07228542 -0.08323604 -0.13549104 -0.10579217 -0.14499113 -0.37823868 -0.21633306 -0.41489958 -0.6
#>     alpha_j     alpha_j      beta_z      beta_z   log_sigma   log_sigma
#>  0.16887044  0.30040122  0.81229112  0.40988915 -0.64868475  0.04394543
#>
#> $opt$objective
#> [1] 1717.689
#>
#> $opt$convergence
#> [1] 0
#>
#> $opt$iterations
#> [1] 77
#>
#> $opt$evaluations
#> function gradient
#>      107       77
#>
#> $opt$message
```

```
#> [1] "relative convergence (4)"
#>
#>
#> $sdrep
#> sdreport(.) result
#>               Estimate Std. Error
#> log_kappa  0.07228542 0.10755269
#> alpha_j   -0.08323604 0.15196456
#> alpha_j   -0.13549104 0.18670340
#> alpha_j   -0.10579217 0.20529851
#> alpha_j   -0.14499113 0.21780791
#> alpha_j   -0.37823868 0.22691800
#> alpha_j   -0.21633306 0.23026450
#> alpha_j   -0.41489958 0.23456777
#> alpha_j   -0.67168423 0.23833635
#> alpha_j   -0.49463135 0.23869331
#> alpha_j   -0.13968721 0.23733095
#> alpha_j    0.14836186 0.23640201
#> alpha_j   -0.21516693 0.23873590
#> alpha_j   -0.20120060 0.23979214
#> alpha_j    0.16887044 0.23708655
#> alpha_j    0.30040122 0.23660035
#> beta_z     0.81229112 0.03708631
#> beta_z     0.40988915 0.03291043
#> log_sigma -0.64868475 0.05422114
#> log_sigma  0.04394543 0.07275797
#> Maximum gradient component: 0.006323539
#>
#> $run_time
#> Time difference of 21.06885 secs
```

The estimated values for `beta_z` then correspond to the simulated value for `rho` and `spatial_sd`.

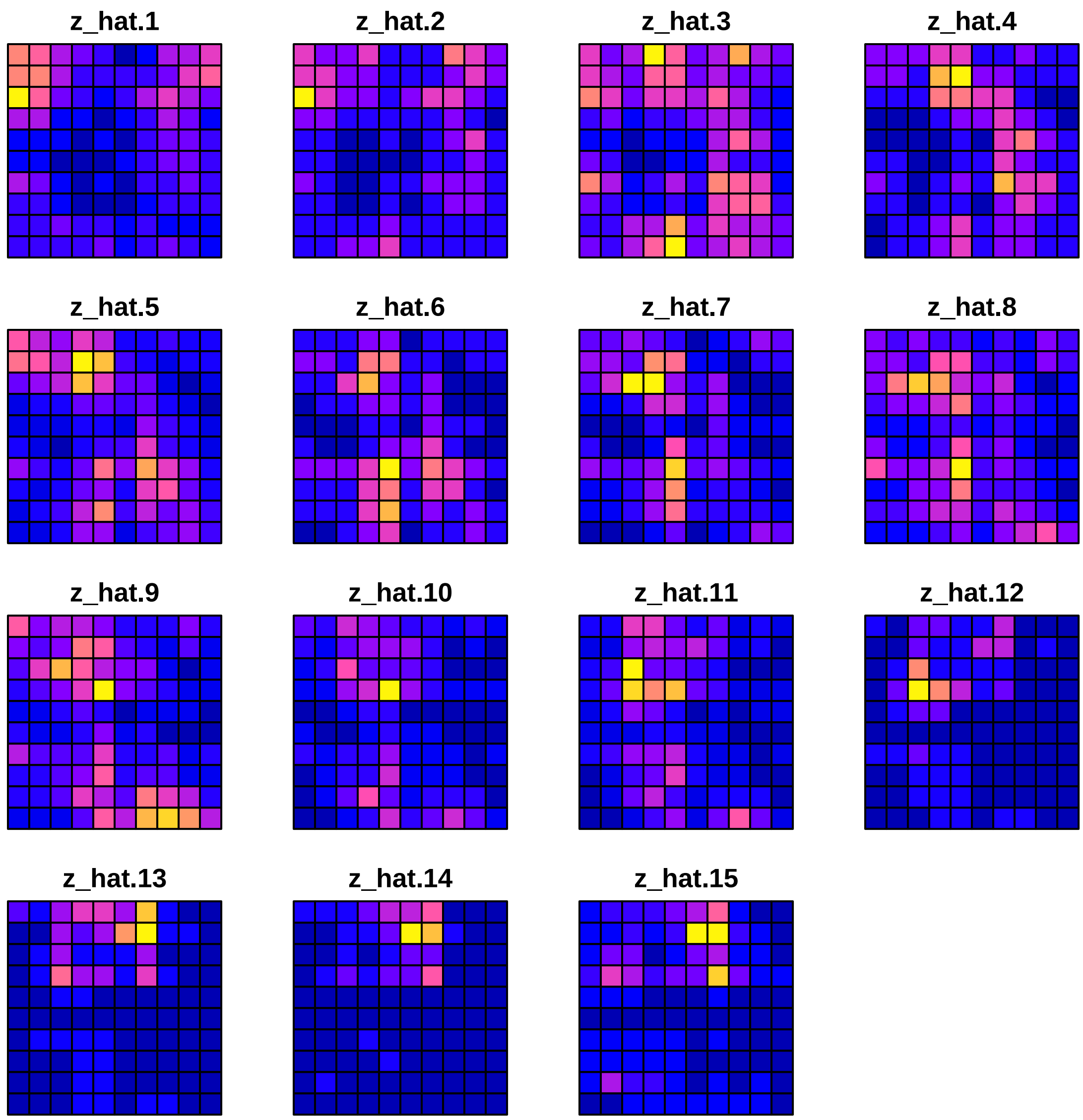
We can compare the true densities:

```r
library(sf)
#> Warning: package 'sf' was built under R version 4.3.1
data_wide = reshape( Data[,c('x','y','time','z')],
                     direction = "wide", idvar = c('x','y'), timevar = "time")
sf_data = st_as_sf( data_wide, coords=c("x","y"))
sf_grid = sf::st_make_grid( sf_data )
sf_plot = st_sf(sf_grid, st_drop_geometry(sf_data) )
plot(sf_plot, max.plot=n_t )
```



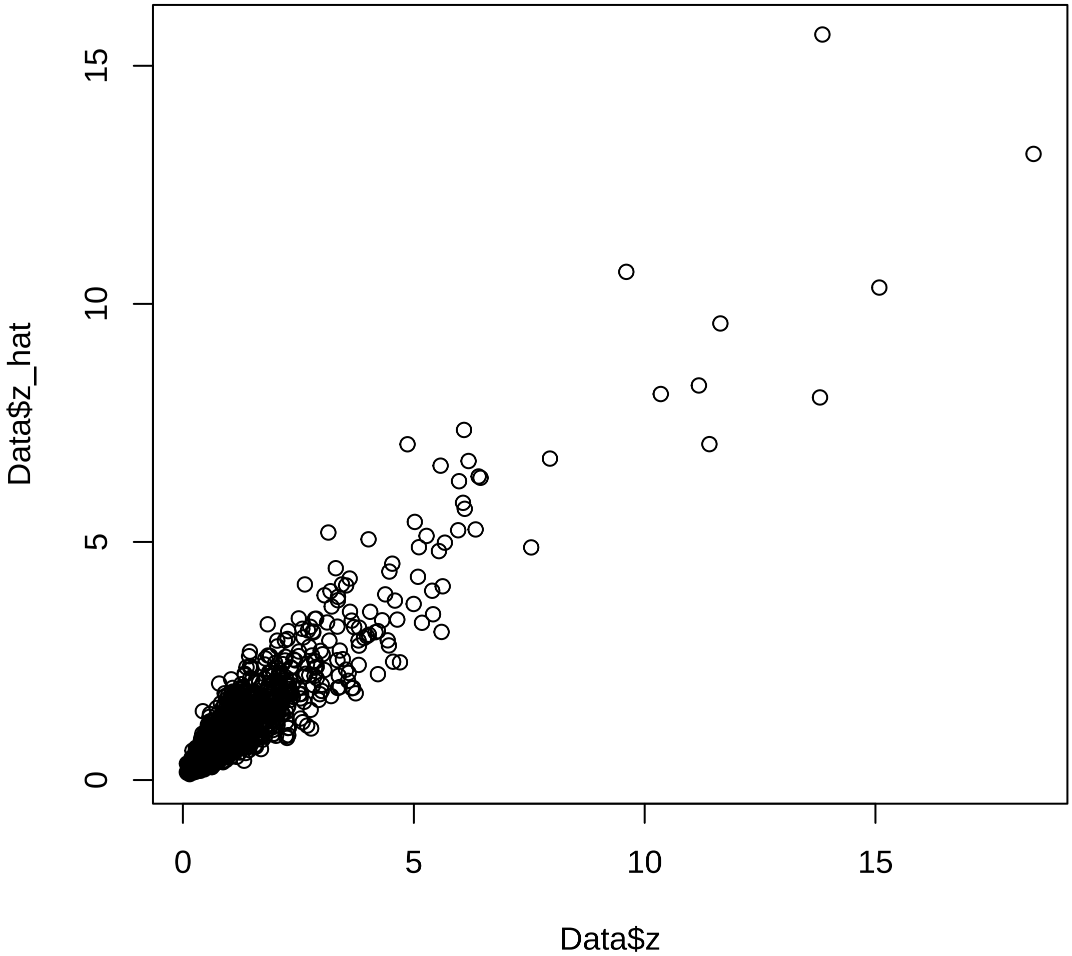
with the estimated densities:

```
Data$z_hat = predict(out)
data_wide = reshape( Data[,c('x','y','time','z_hat')],
                     direction = "wide", idvar = c('x','y'), timevar = "time")
sf_data = st_as_sf( data_wide, coords=c("x","y"))
sf_plot = st_sf(sf_grid, st_drop_geometry(sf_data) )
plot(sf_plot, max.plot=n_t )
```



where a scatterplot shows that they are highly correlated:

```
plot( x=Data$z, y=Data$z_hat )
```



We can then calculate the area-weighted total abundance and compare it with its true value:

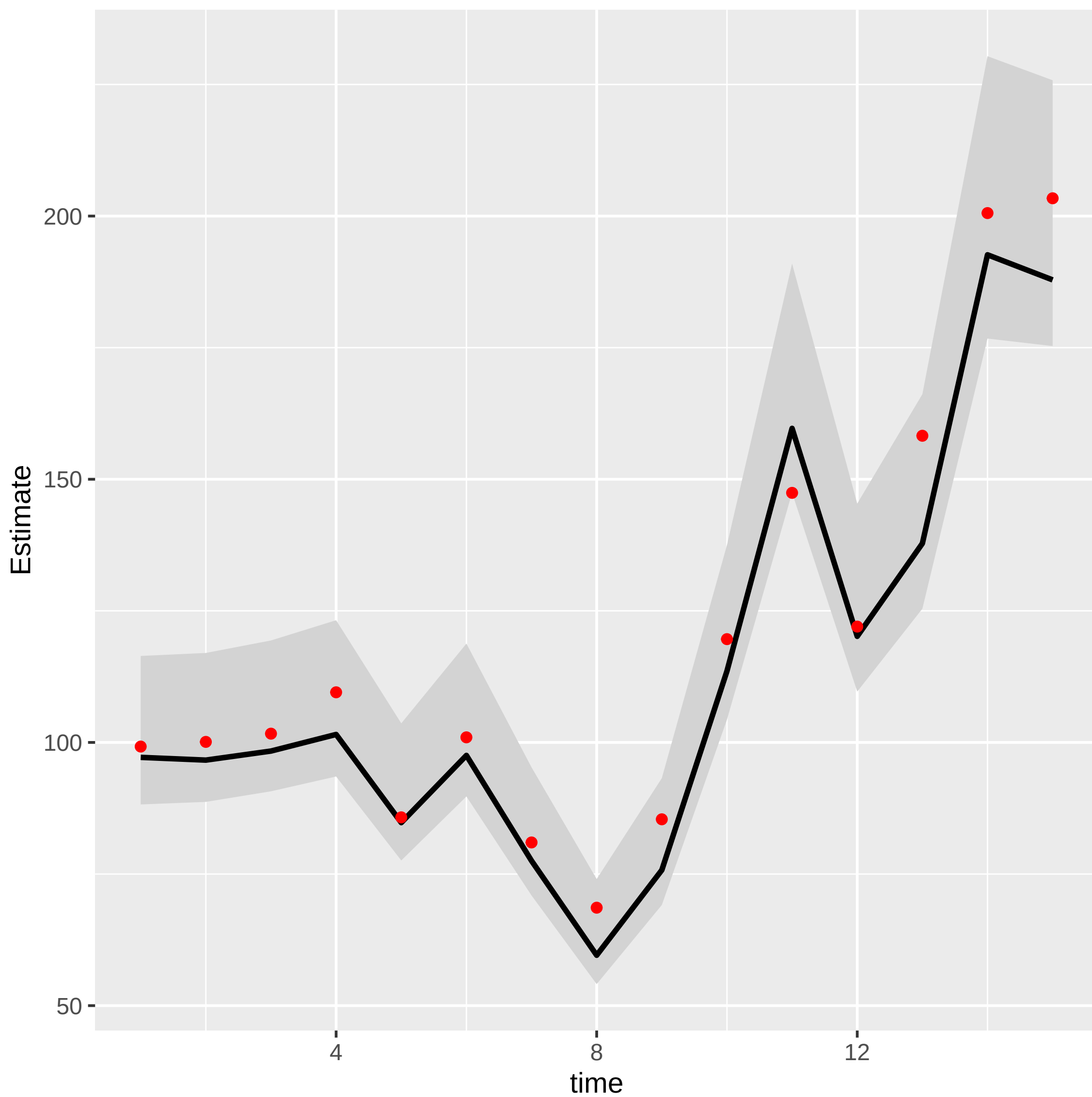
```
# Predicted sample-weighted total
(Est = sapply( seq_len(n_t), FUN=\(t) integrate_output(out, newdata=subset(Data,time==t)) ))
#>                               [,1]        [,2]        [,3]        [,4]       [,5]        [,6]       [,7]
#> Estimate              97.164903  96.643634  98.362457 101.517620 84.760587  97.538111  77.520820 59.56
#> Std. Error             7.194683   7.216494   7.309313   7.572241  6.643419   7.406226   6.207919  5.09
#> Est. (bias.correct) 102.324275 102.850496 105.043004 108.373177 90.604659 104.258111 83.102278 64.06
#> Std. (bias.correct)         NA          NA          NA          NA        NA          NA         NA
#>                             [,15]
#> Estimate             187.86973
#> Std. Error            12.88189
#> Est. (bias.correct) 200.54754
#> Std. (bias.correct)         NA

# True (latent) sample-weighted total
(True = tapply( Data$z, INDEX=Data$time, FUN=sum ))
```

```
#>         1         2         3         4         5         6         7         8         9        10
#>  99.21643 100.10603 101.66846 109.52622  85.76973 100.97116  80.99847  68.60738  85.39974 119.62380

#
Index = data.frame( time=seq_len(n_t), t(Est), True )
Index$low = Index[,'Est...bias.correct.'] - 1.96*Index[,'Std..Error']
Index$high = Index[,'Est...bias.correct.'] + 1.96*Index[,'Std..Error']

#
library(ggplot2)
ggplot(Index, aes(time, Estimate)) +
  geom_ribbon(aes(ymin = low,
                  ymax = high),    # shadowing cnf intervals
              fill = "lightgrey") +
  geom_line( color = "black",
            linewidth = 1) +
  geom_point( aes(time, True), color = "red" )
```



Next, we compare this against the current version of VAST

```
#library(VAST)
#settings = make_settings( purpose="index3",
#                          n_x = n_x*n_y,
#                          Region = "Other",
#                          bias.correct = FALSE,
#                          use_anisotropy = FALSE )
#settings$FieldConfig['Epsilon','Component_1'] = 0
#settings$FieldConfig['Omega',] = 0
#settings$RhoConfig['Epsilon2'] = 4
#settings$RhoConfig['Beta1'] = 3
#
#myVAST = fit_model( settings=settings,
#                 Lat_i = Data[,'y'],
#                 Lon_i = Data[,'x'],
#                 t_i = Data[,'time'],
```

```
#                   b_i = exp(Data[,'n']),
#                   a_i = rep(1,nrow(Data)),
#                   observations_LL = cbind(Lat=Data[,'y'],Lon=Data[,'x']),
#                   grid_dim_km = c(100,100),
#                   ObsModel = c(10,2),
#                   newtonsteps = 0,
#                   loopnum = 0,
#                   control = list(eval.max=100, iter.max=100, trace=0) )
#myVAST
```

Or with sdmTMB

```
#library(INLA)
library(sdmTMB)
#> Warning: package 'sdmTMB' was built under R version 4.3.1
mesh = make_mesh(Data, c("x","y"), n_knots=n_x*n_y )

start_time = Sys.time()
mysdmTMB = sdmTMB(
  formula = n ~ 0 + factor(time),
  data = Data,
  mesh = mesh,
  spatial = "off",
  spatiotemporal = "ar1",
  time = "time",
  family = tweedie()
)
Sys.time() - start_time
#> Time difference of 20.78361 secs
```

Bivariate spatio-temporal autoregressive model

We next highlight how to specify a bivariate spatio-temporal model with a cross-laggged (vector autoregressive) interaction.

```
# Simulate settings
theta_xy = 0.2
n_x = n_y = 10
n_t = 20
B = rbind( c( 0.5, -0.25),
           c(-0.1,  0.50) )

# Simulate GMRFs
R = exp(-theta_xy * abs(outer(1:n_x, 1:n_y, FUN="-")) )
d1 = mvtnorm::rmvnorm(n_t, sigma=0.2*kronecker(R,R) )
d2 = mvtnorm::rmvnorm(n_t, sigma=0.2*kronecker(R,R) )
d = abind::abind( d1, d2, along=3 )

# Project through time and add mean
for( t in seq_len(n_t) ){
  if(t>1) d[t,,] = t(B%*%t(d[t-1,,])) + d[t,,]
```

```
}

# Shape into longform data-frame and add error
Data = data.frame( expand.grid(time=1:n_t, x=1:n_x, y=1:n_y, "var"=c("d1","d2")), z=exp(as.vector(d)))
Data$n = tweedie::rtweedie( n=nrow(Data), mu=Data$z, phi=0.5, power=1.5 )

# make mesh
mesh = fm_mesh_2d( Data[,c('x','y')] )

# Define sem
sem = "
  d1 -> d1, 1, b11
  d2 -> d2, 1, b22
  d2 -> d1, 1, b21
  d1 -> d2, 1, b12
  d1 <-> d1, 0, var1
  d2 <-> d2, 0, var1
"

# fit model
out = fit( sem = sem,
           data = Data,
           formula = n ~ 0 + var,
           spatial_graph = mesh,
           quiet = TRUE,
           family_link = c(1,0) )
#>   0:     5463.7926:  0.00000  0.00000  0.00000 0.0100000 0.0100000 0.0100000 0.0100000  1.00000  0.0
#>   1:     4468.1932: -0.130049 -0.0140428 -0.00910783 0.0261043 0.0251760 -0.00569561 0.00650995 0.40
#>   2:     4439.4453: -0.215344 -0.0646324 -0.0272373 0.111636 0.105730 -0.0691738 -0.0176371 0.435290
#>   3:     4411.3539: -0.234522 -0.0681309 -0.0286478 0.125461 0.119819 -0.0793320 -0.0229344 0.351924
#>   4:     4402.7917: -0.287323 -0.0720996 -0.0261582 0.170130 0.165610 -0.111223 -0.0400816 0.417072
#>   5:     4392.4429: -0.318655 -0.0751171 -0.0278434 0.209144 0.207144 -0.137387 -0.0566640 0.329167
#>   6:     4383.5855: -0.356740 -0.0732864 -0.0266103 0.263726 0.266862 -0.172346 -0.0806822 0.366851
#>   7:     4375.1196: -0.388807 -0.0698254 -0.0268313 0.321847 0.336152 -0.200096 -0.107974 0.322807 -
#>   8:     4371.1175: -0.475038 -0.0824505 -0.0295182 0.447706 0.504266 -0.222124 -0.140869 0.329476 -
#>   9:     4368.5194: -0.661435 -0.0789254 -0.00412338 0.471662 0.499045 -0.161733 -0.0671517 0.280467
#>  10:     4367.1890: -0.661109 -0.0788828 -0.00422126 0.477171 0.504751 -0.167275 -0.0721269 0.319666
#>  11:     4365.8230: -0.665249 -0.0806453 -0.00366716 0.479741 0.499473 -0.174842 -0.0823816 0.299816
#>  12:     4365.6483: -0.658141 -0.0816606 -0.00374626 0.488194 0.504484 -0.191372 -0.103005 0.303441
#>  13:     4365.2276: -0.634996 -0.0841607 -0.00591765 0.487400 0.505857 -0.202898 -0.125838 0.294829
#>  14:     4365.1572: -0.635698 -0.0843502 -0.00594408 0.488872 0.507459 -0.203650 -0.125821 0.302587
#>  15:     4365.0972: -0.639386 -0.0858514 -0.00633171 0.490790 0.509413 -0.202391 -0.120276 0.297489
#>  16:     4365.0827: -0.644639 -0.0864918 -0.00561349 0.493743 0.513533 -0.201282 -0.116565 0.299976
#>  17:     4365.0687: -0.655565 -0.0825834 0.000628898 0.496471 0.523295 -0.196777 -0.121207 0.296355
#>  18:     4365.0420: -0.656097 -0.0830745 0.000229283 0.497354 0.523492 -0.197410 -0.120984 0.295059
#>  19:     4365.0355: -0.656351 -0.0833971 -2.64364e-05 0.498083 0.523787 -0.197911 -0.120886 0.296568
#>  20:     4365.0309: -0.656736 -0.0839570 -0.000470777 0.499022 0.524006 -0.198519 -0.120630 0.295518
#>  21:     4365.0246: -0.657500 -0.0855687 -0.00175862 0.501175 0.523465 -0.199839 -0.119754 0.296703
#>  22:     4365.0222: -0.660285 -0.0898865 -0.00475161 0.503733 0.520877 -0.200754 -0.117905 0.294498
#>  23:     4365.0154: -0.664498 -0.0908139 -0.00243753 0.503148 0.524323 -0.198582 -0.116556 0.295556
#>  24:     4365.0115: -0.663570 -0.0874999 0.000521568 0.507929 0.526248 -0.200735 -0.116124 0.294597
#>  25:     4365.0101: -0.663696 -0.0876799 0.000318875 0.507993 0.526498 -0.200811 -0.116268 0.295241
#>  26:     4365.0091: -0.663929 -0.0880226 -0.000232412 0.507831 0.526848 -0.200800 -0.116623 0.294553
```

```
#>  27:     4365.0083: -0.664233 -0.0883408 -0.000846481 0.507716 0.527248 -0.200823 -0.117094 0.295037
#>  28:     4365.0078: -0.665318 -0.0886451 -0.00209962 0.507376 0.527460 -0.200596 -0.118349 0.294527
#>  29:     4365.0072: -0.667042 -0.0883068 -0.00198054 0.507757 0.527061 -0.200206 -0.118226 0.294766
#>  30:     4365.0068: -0.666387 -0.0894045 -0.00193958 0.507706 0.528379 -0.200217 -0.117021 0.294530
#>  31:     4365.0067: -0.667264 -0.0905253 -0.00140119 0.508859 0.528084 -0.200717 -0.117789 0.294864
#>  32:     4365.0067: -0.667762 -0.0900651 -0.00162759 0.509235 0.527878 -0.200732 -0.117830 0.294187
#>  33:     4365.0063: -0.667757 -0.0898829 -0.00183551 0.509244 0.528078 -0.200727 -0.117706 0.294540
#>  34:     4365.0062: -0.667806 -0.0896632 -0.00207909 0.509097 0.528384 -0.200695 -0.117530 0.294386
#>  35:     4365.0061: -0.668308 -0.0901159 -0.00142377 0.509213 0.528526 -0.200248 -0.117691 0.294459
#>  36:     4365.0061: -0.668851 -0.0895239 -0.00192741 0.509118 0.528689 -0.200659 -0.117347 0.294400
#>  37:     4365.0061: -0.668833 -0.0896377 -0.00200130 0.509178 0.528809 -0.200655 -0.117345 0.294338
#>  38:     4365.0061: -0.668835 -0.0896474 -0.00211183 0.509297 0.528872 -0.200547 -0.117344 0.294373
#>  39:     4365.0060: -0.668847 -0.0896822 -0.00220804 0.509424 0.528937 -0.200396 -0.117315 0.294347
#>  40:     4365.0060: -0.668919 -0.0899979 -0.00197822 0.509246 0.529016 -0.200634 -0.117266 0.294339
#>  41:     4365.0060: -0.668919 -0.0900060 -0.00198427 0.509263 0.529033 -0.200629 -0.117264 0.294362
#>  42:     4365.0060: -0.668921 -0.0900144 -0.00198914 0.509279 0.529043 -0.200616 -0.117262 0.294336
#>  43:     4365.0060: -0.668935 -0.0900318 -0.00199027 0.509336 0.529060 -0.200576 -0.117262 0.294356
#>  44:     4365.0060: -0.668977 -0.0900679 -0.00196831 0.509436 0.529093 -0.200481 -0.117254 0.294331
#>  45:     4365.0060: -0.668964 -0.0901023 -0.00196244 0.509454 0.529128 -0.200475 -0.117275 0.294328
#>  46:     4365.0060: -0.668958 -0.0900958 -0.00200627 0.509449 0.529170 -0.200457 -0.117245 0.294338
#>  47:     4365.0060: -0.668998 -0.0901162 -0.00198859 0.509473 0.529176 -0.200449 -0.117195 0.294328
#>  48:     4365.0060: -0.669037 -0.0901292 -0.00198975 0.509508 0.529196 -0.200444 -0.117232 0.294324
#>  49:     4365.0060: -0.669045 -0.0901288 -0.00199467 0.509507 0.529214 -0.200433 -0.117231 0.294316
#>  50:     4365.0060: -0.669053 -0.0901263 -0.00199961 0.509508 0.529231 -0.200423 -0.117227 0.294323
#>  51:     4365.0060: -0.669048 -0.0901340 -0.00200694 0.509517 0.529222 -0.200421 -0.117216 0.294325
#>  52:     4365.0060: -0.669057 -0.0901284 -0.00200043 0.509529 0.529236 -0.200419 -0.117205 0.294319
out
#> $call
#> fit(data = Data, formula = n ~ 0 + var, sem = sem, family_link = c(1,
#>     0), spatial_graph = mesh, quiet = TRUE)
#>
#> $opt
#> $opt$par
#>    log_kappa      alpha_j      alpha_j       beta_z       beta_z       beta_z       beta_z       bet
#> -0.669057051 -0.090128408 -0.002000431  0.509529379  0.529236413 -0.200418867 -0.117205379  0.294319
#>
#> $opt$objective
#> [1] 4365.006
#>
#> $opt$convergence
#> [1] 0
#>
#> $opt$iterations
#> [1] 52
#>
#> $opt$evaluations
#> function gradient
#>       66       53
#>
#> $opt$message
#> [1] "relative convergence (4)"
#>
#>
```

```
#> $sdrep
#> sdreport(.) result
#>              Estimate Std. Error
#> log_kappa -0.669057051 0.09746707
#> alpha_j   -0.090128408 0.09771148
#> alpha_j   -0.002000431 0.09611298
#> beta_z     0.509529379 0.07886506
#> beta_z     0.529236413 0.07336727
#> beta_z    -0.200418867 0.08304602
#> beta_z    -0.117205379 0.07264403
#> beta_z     0.294319075 0.01800602
#> log_sigma -0.646266078 0.02660900
#> log_sigma  0.012846260 0.04964136
#> Maximum gradient component: 0.004488392
#>
#> $run_time
#> Time difference of 2.170671 mins
```

The values for `beta_z` again correspond to the specified value for interaction-matrix `B`

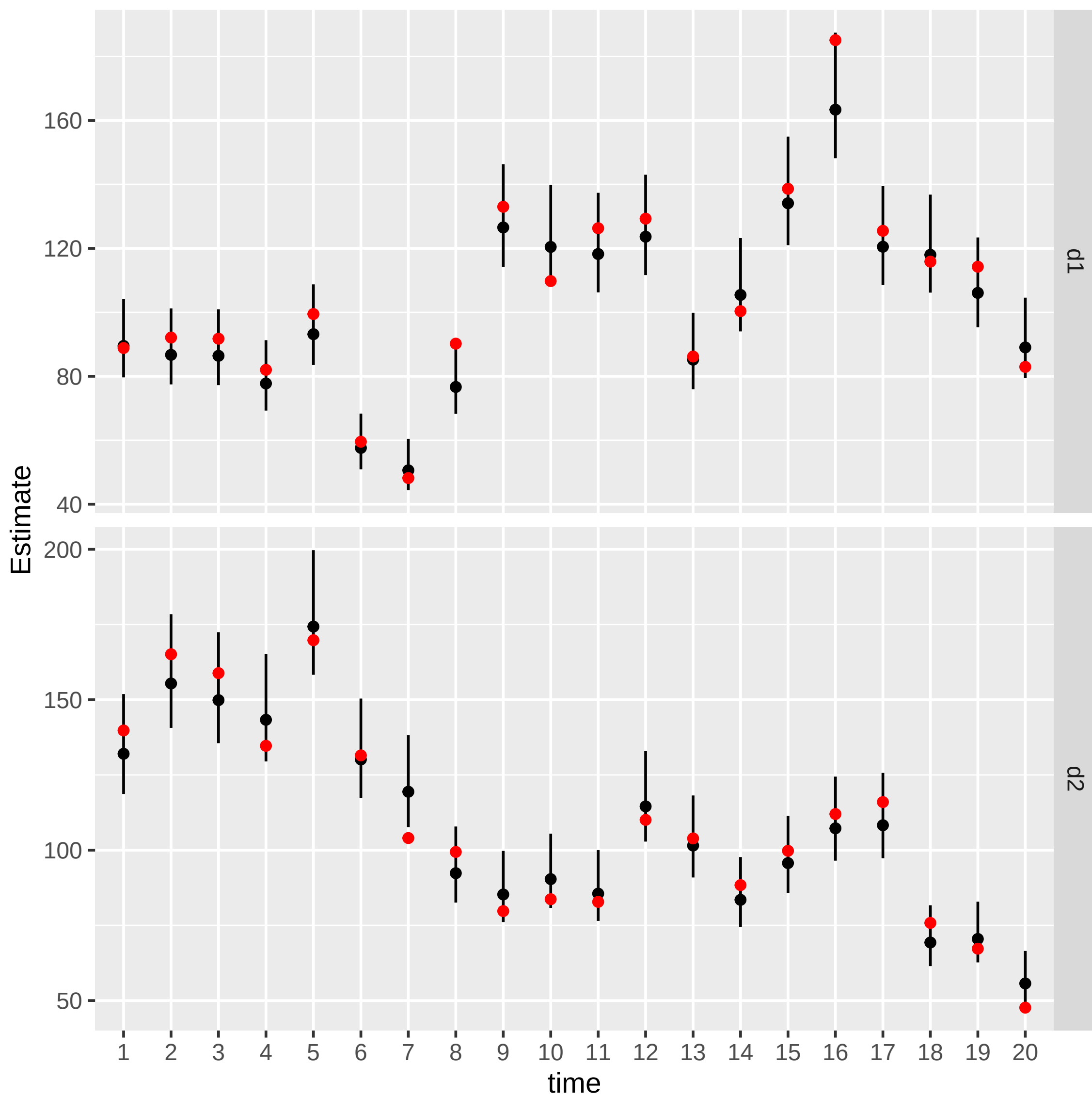
We can again calculate the area-weighted total abundance and compare it with its true value:

```
# Predicted sample-weighted total
Est1 = sapply( seq_len(n_t), FUN=\(t) integrate_output(out, newdata=subset(Data,time==t & var=="d1")) )
Est2 = sapply( seq_len(n_t), FUN=\(t) integrate_output(out, newdata=subset(Data,time==t & var=="d2")) )

# True (latent) sample-weighted total
True = tapply( Data$z, INDEX=list("time"=Data$time,"var"=Data$var), FUN=sum )

#
Index = data.frame( expand.grid(dimnames(True)), "True"=as.vector(True) )
Index = data.frame( Index, rbind(t(Est1), t(Est2)) )
Index$low = Index[,'Est...bias.correct.'] - 1.96*Index[,'Std..Error']
Index$high = Index[,'Est...bias.correct.'] + 1.96*Index[,'Std..Error']

#
library(ggplot2)
ggplot(Index, aes( time, Estimate )) +
  facet_grid( rows=vars(var), scales="free" ) +
  geom_segment(aes(y = low,
                   yend = high,
                   x = time,
                   xend = time) ) +
  geom_point( aes(x=time, y=Estimate), color = "black") +
  geom_point( aes(x=time, y=True), color = "red" )
```


Estimate
160
120
80
40
200
150
100
50
d1
d2
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
time