

Final

dan crowley

4/28/2020

```
library(ape)
library(phylofactor)

## Loading required package: magrittr
## Loading required package: data.table
## Loading required package: Matrix
library(tidyverse)

## -- Attaching packages -----
## v ggplot2 3.3.0      v purrr  0.3.3
## v tibble  3.0.0      v dplyr  0.8.5
## v tidyr   1.0.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.4.0

## -- Conflicts -----
## x dplyr::between()   masks data.table::between()
## x tidyr::expand()    masks Matrix::expand()
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x tidyr::pack()      masks Matrix::pack()
## x purrr::set_names() masks magrittr::set_names()
## x purrr::transpose() masks data.table::transpose()
## x tidyr::unpack()    masks Matrix::unpack()

sim = (seq(0.000001,100,10)) #must be the same as line 11
eps = (seq(0.000001,100,10))
error <- matrix(0, length(sim)*length(eps), 14)
index = 0

set.seed(Sys.time())
num= 50 #rpois(1,60)
tree <-rtree(num)
tree$tip.label <- as.character(1:num)
drop_tips = num/20 # determines ratio of test to training data

#determine max groups
max_groups <- 2*num-2

#choose a clade of the good length, we want to make sure its not monophyletic
```

```

clade1 = 0
clade2 = 0

while((length(clade1) < num / 3) | (length(clade2) < num / 3))
{
  #grab clades
  samp = sample(max_groups, 1)
  clade1 = getPhyloGroups(tree)[samp][1][[1]][[1]]
  clade2 = getPhyloGroups(tree)[samp][1][[1]][[2]]
}

#randomly generate data
BodySize <- rlnorm(num, sdlog = 1.6)
BodySize[clade1] <- rlnorm(length(clade1), sdlog = 1.6)*5

#create a data matrix, with the body size, tip labels, and a basis function of all 1s
BodySize = as.data.frame(cbind(BodySize, tree$tip.label, basis = 1, intercept = 1))
BodySize <- BodySize %>%
  dplyr::mutate(Species= as.character(V2)) %>%
  dplyr::select(-V2)

#now wegenerate the missing data.
#these are species we haven't observed data for
#in this simulation around 1/3 of the data are missing

#num_miss = 1 + rpois(1,nrow(train_BodySize)/2.5)
num_miss = nrow(BodySize)/3
#num_miss = rpois(1,10)
miss_tip = sample(tree$tip.label, num_miss)

#create missing body size variable
BodySize$BodySize_miss = BodySize$BodySize
BodySize$basis_miss = BodySize$basis
BodySize$intercept_miss = BodySize$intercept

#create a new variable, and label the boddy size and basis functions NA for these tips
BodySize[(BodySize$Species %in% miss_tip),]$BodySize_miss = NA
BodySize[(BodySize$Species %in% miss_tip),]$basis_miss = NA
BodySize[(BodySize$Species %in% miss_tip),]$intercept_miss = NA

BodySize$basis = as.numeric(BodySize$basis)
BodySize$BodySize = as.numeric(as.character(BodySize$BodySize))
BodySize$intercept = as.numeric(BodySize$intercept)

#grab a sample of N tips for the testing tree
test_tree <- ape::drop.tip(tree,tree$tip.label[!(tree$tip.label %in% sample(tree$tip.label, drop_tips))])
train_tree <- ape::drop.tip(tree,tree$tip.label[(tree$tip.label %in% test_tree$tip.label)])

#should be equal to 0
sum(test_tree$tip.label %in% train_tree$tip.label) == 0

## [1] TRUE

```

```

sum(train_tree$tip.label %in% test_tree$tip.label) == 0

## [1] TRUE
num - (length(train_tree$tip.label) + length(test_tree$tip.label)) == 0

## [1] TRUE
#now, split the dataset into the training and testing datasets
train_BodySize = BodySize %>%
  dplyr::filter(Species %in% train_tree$tip.label)

test_BodySize = BodySize %>%
  dplyr::filter(Species %in% test_tree$tip.label)

#should both be true:
sum(train_BodySize$Species %in% test_BodySize$Species) == 0

## [1] TRUE
sum(test_BodySize$Species %in% train_BodySize$Species) == 0

## [1] TRUE
train_BodySize$basis_miss = as.numeric(train_BodySize$basis_miss)
train_BodySize$BodySize_miss = as.numeric(as.character(train_BodySize$BodySize_miss))
train_BodySize$intercept_miss = as.numeric(train_BodySize$intercept_miss)

test_BodySize <- test_BodySize %>%
  dplyr::mutate(intercept = as.numeric(as.character(intercept))) %>%
  dplyr::mutate(BodySize = as.numeric(as.character(BodySize))) %>%
  dplyr::mutate(basis = as.numeric(as.character(basis)))

train_BodySize$basis_miss = as.numeric(as.character(train_BodySize$basis_miss))
train_BodySize$intercept_miss = as.numeric(as.character(train_BodySize$intercept_miss))
train_BodySize$train_BodySize_miss = as.numeric(as.character(train_BodySize$BodySize_miss))

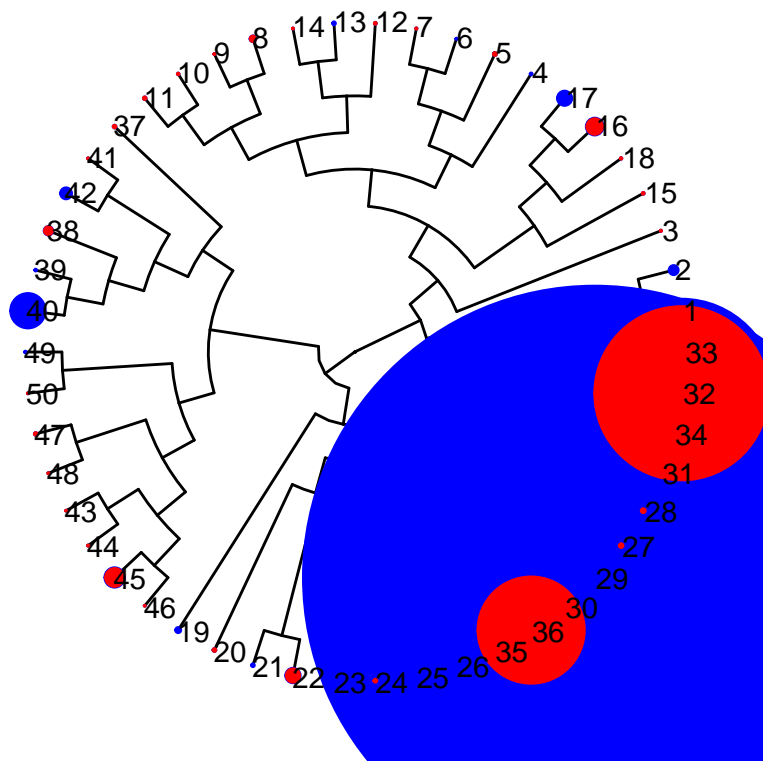
ggtree::ggtree(tree, branch.length = 'none', layout = 'circular') +
  ggtree::geom_tippoint(size=.25*as.numeric(BodySize$BodySize),col='blue') +
  ggtree::geom_tippoint(size=.23*as.numeric(as.character(BodySize$BodySize_miss)),col='red') +
  ggtree::geom_tiplab()

## Registered S3 method overwritten by 'treeio':
##   method      from
##   root.phylo ape

## Warning: `data_frame()` is deprecated as of tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.

## Warning: Removed 16 rows containing missing values (geom_point_ggtree).

```



```
for( x in sim)
{
  print(paste("X =", x))
  epsilon = x
  j=1
  for(j in eps)
  {

    #switch things up, keep the same tree, but grab different training and tesing everytime

    print(paste("j =", j))
    index = index + 1
    delta = 1 #delta is a control parameter for the bayesian model. as it gets bigger the estimates shr
    delta = delta*(j)
    #delta = delta*j
    #print(num)
    source('6th_attempt.R')

    #create the predictions based on mu_map
    #create a new variable, and label the boddy size and basis functions NA for these tips

    #we need to test the max estimate, and the mle estimate
    #when there are no missing data, they find the same split point
    #however, when the tips are missing they tend to get different
    #we should probably test them both on the actual split point..

    #first, identify the two groups pulled out by GPF in the training dataset

    #grp1 = train_tree$tip.label[gpf_results$groups[1][[1]][[1]]]
```

```

#grp2 = train_tree$tip.label[gpf_results$groups[1][[1]][[2]]]
min = which.min(results[,1])
min_2 = which.min(results[,7])

# results[i,1] <- sse_map
# results[i,2] <- beta_ridge[2]
# results[i,3] <- beta_ridge[1]
# results[i,4] <- N1
# results[i,5] <- N2
# results[i,6] <- theta_2
# results[i,7] <- SSE_train

error[index,1] <- results[min,1] #sse
error[index,2] <- results[min,2]
error[index,3] <- results[min,3]
error[index,4] <- results[min,4]
error[index,5] <- results[min,5]
error[index,6] <- results[min,6]
error[index,7] <- delta
error[index,8] <- epsilon
error[index,9] <- sse_gpf
error[index,10] <- results[min,7]
error[index,11] <- results[min_2,1]
error[index,12] <- results[min_2,7]
error[index,13] <- min
error[index,14] <- min_2

print(error[index,])
}
}

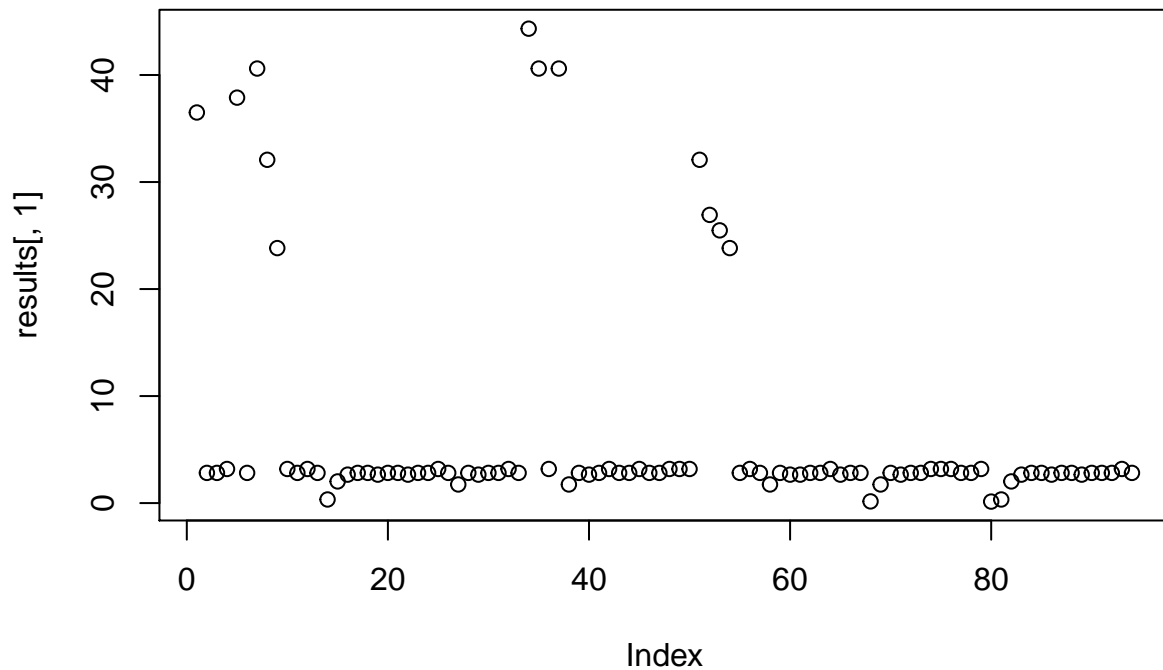
```

```
## [1] "X = 1e-06"
```

```
## [1] "j = 1e-06"
```

```
## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

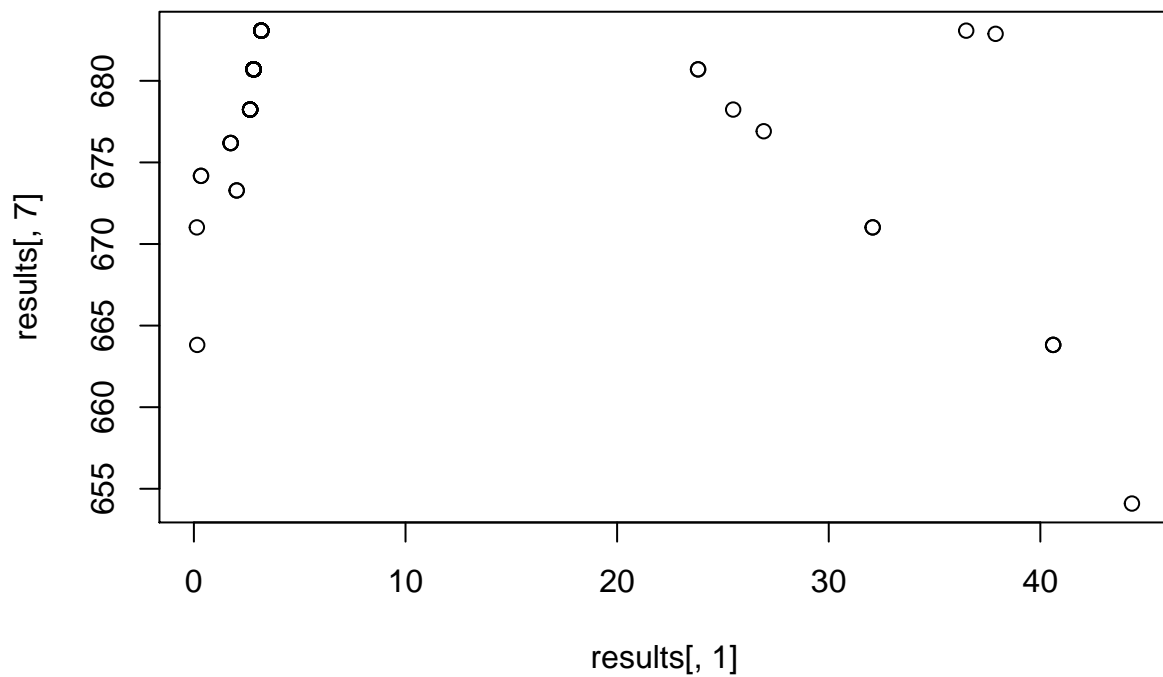
```
## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

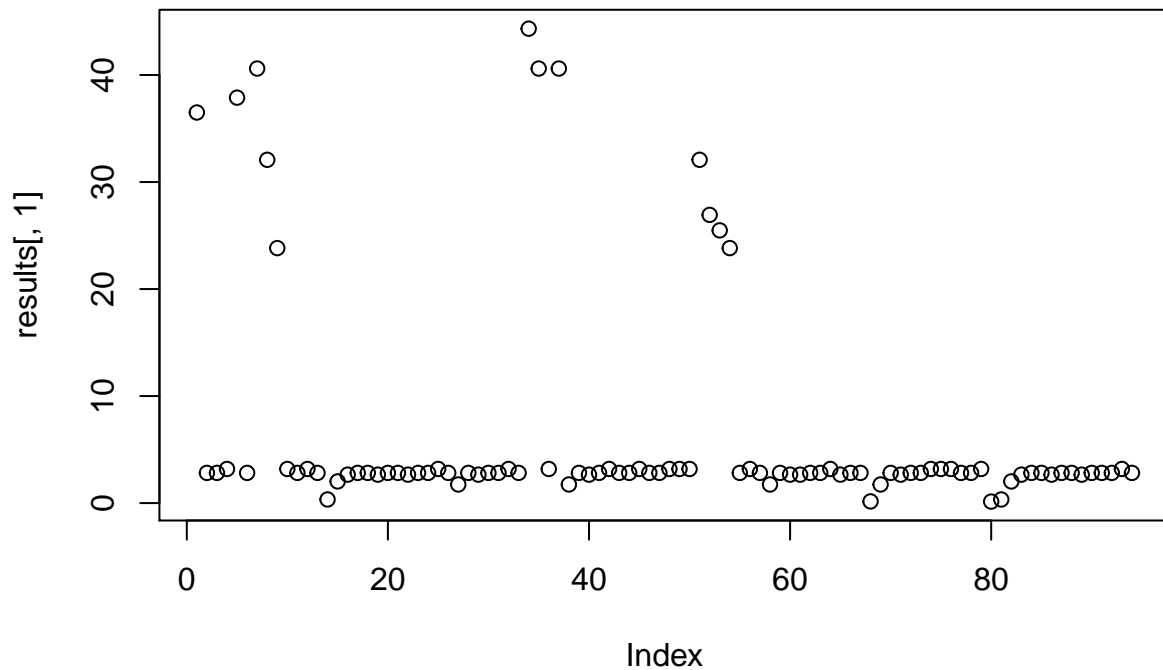


```
## 1 factor completed in 0.00642 minutes. Estimated time of completion: 2020-04-29 13:46:58 [1]
## [6] 9.999995e-07 1.000000e-06 1.000000e-06 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 10.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

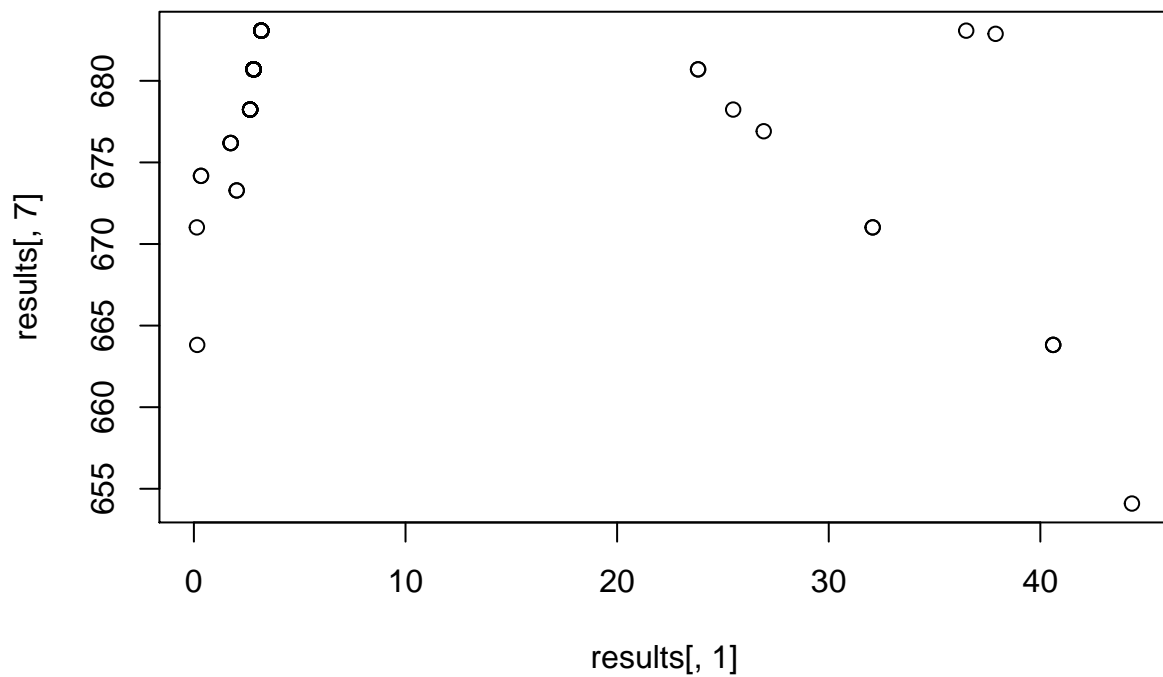


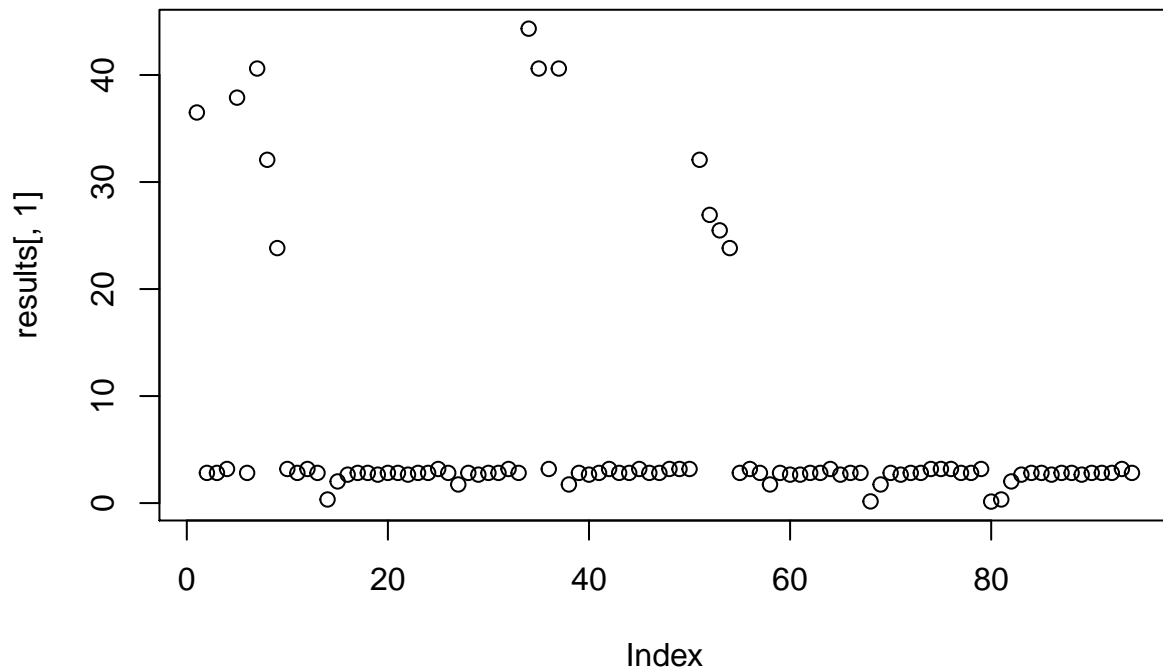


```
## 1 factor completed in 0.00437 minutes. Estimated time of completion: 2020-04-29 13:47:00 [1]
## [6] 4.216497e-09 1.000000e+01 1.000000e-06 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 20.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

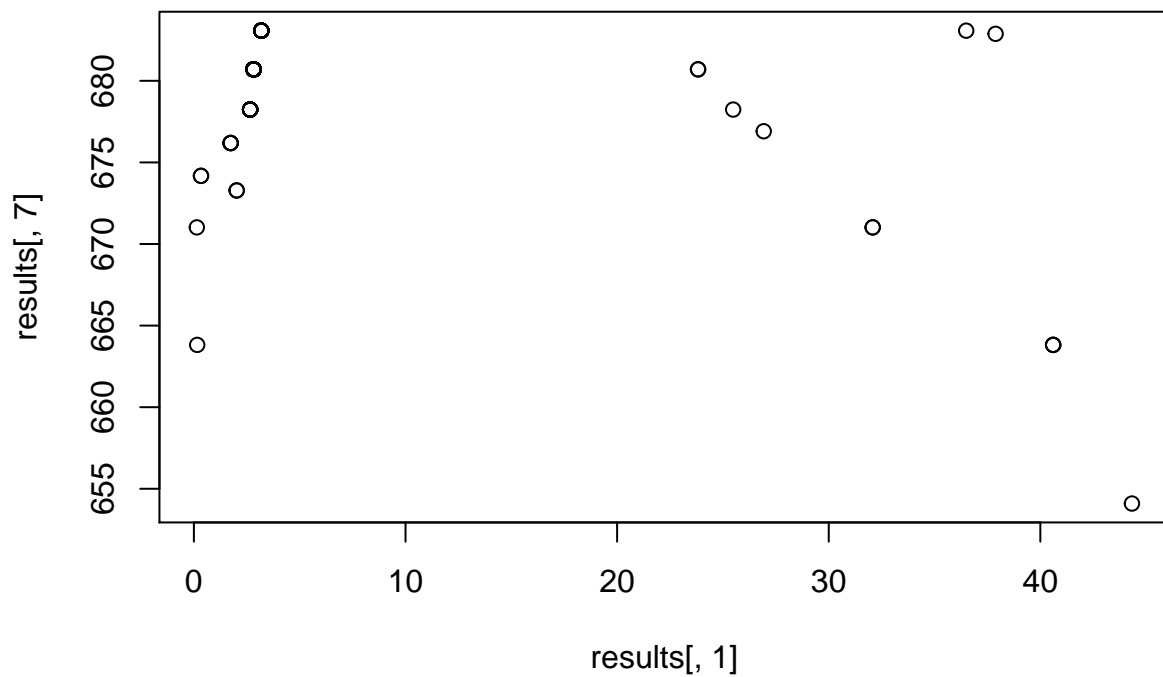


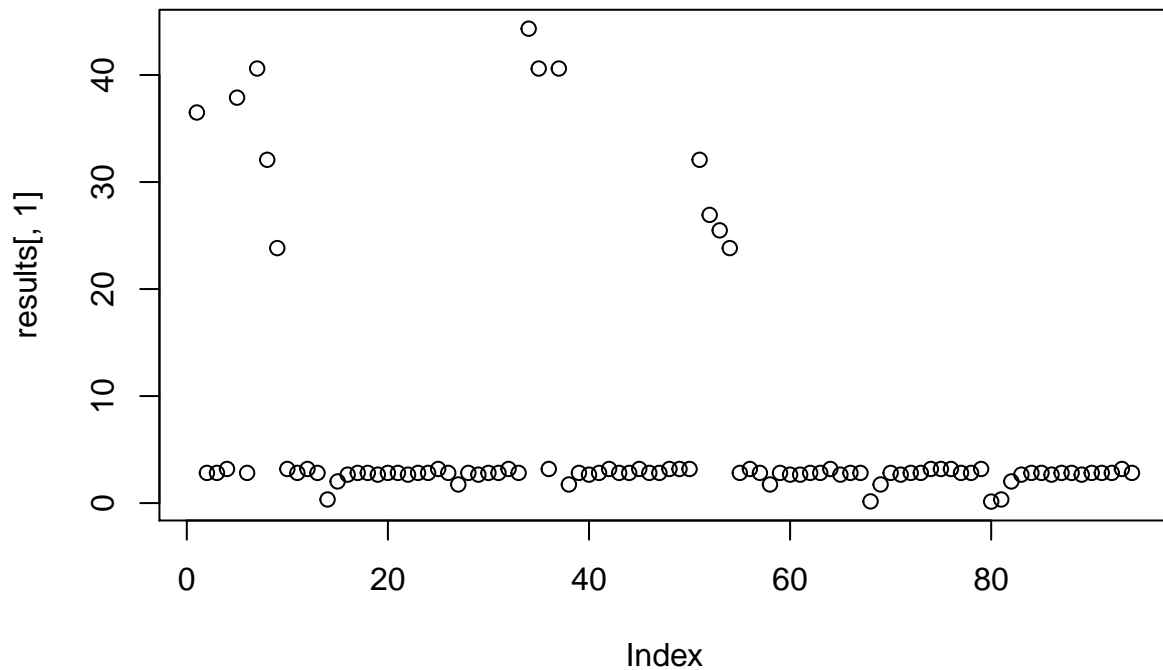


```
## 1 factor completed in 0.0042 minutes. Estimated time of completion: 2020-04-29 13:47:02 [1]
## [6] 1.777886e-11 2.000000e+01 1.000000e-06 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 30.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

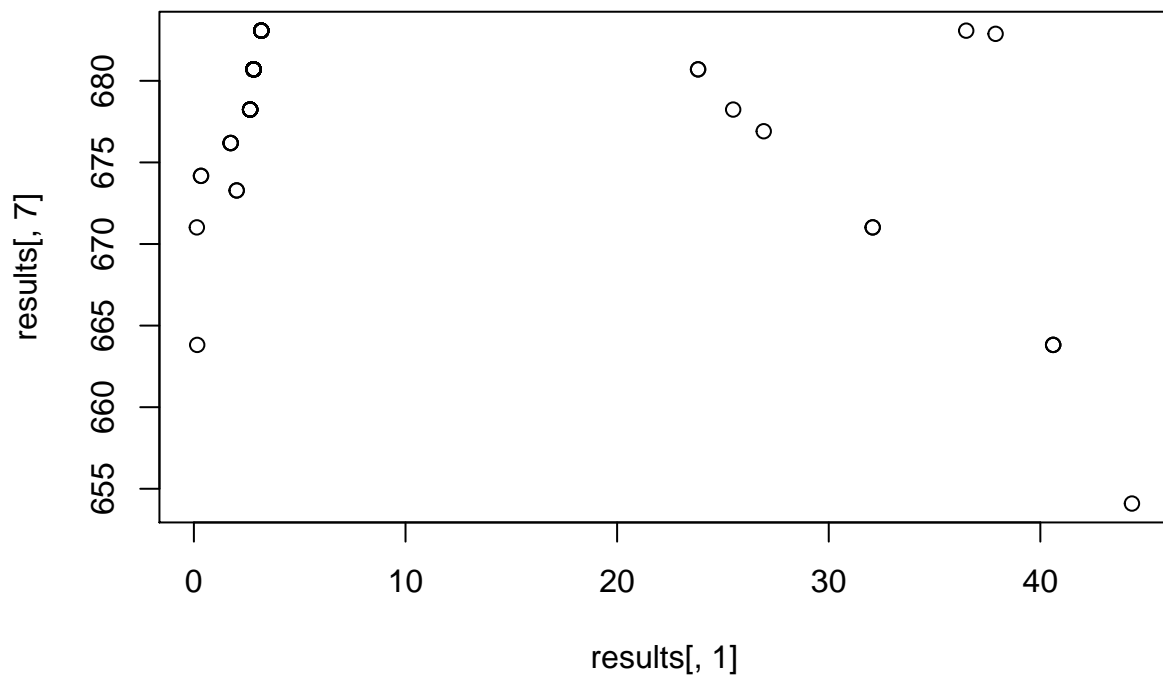


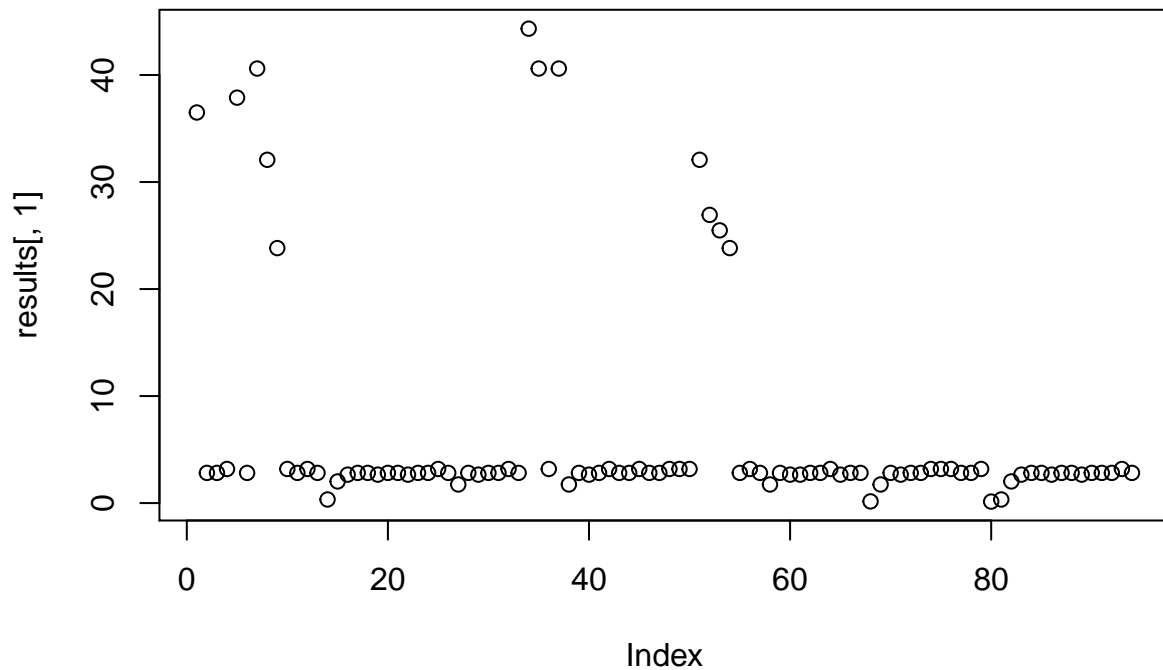


```
## 1 factor completed in 0.00417 minutes. Estimated time of completion: 2020-04-29 13:47:05 [1]
## [6] 7.496455e-14 3.000000e+01 1.000000e-06 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 40.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

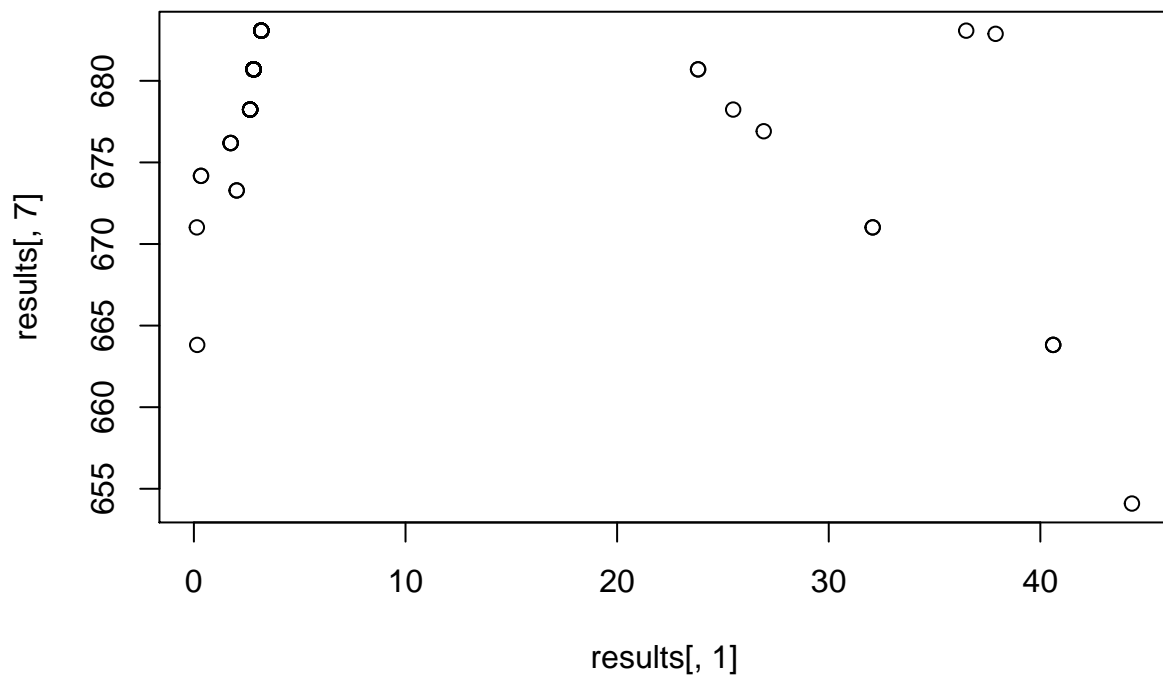


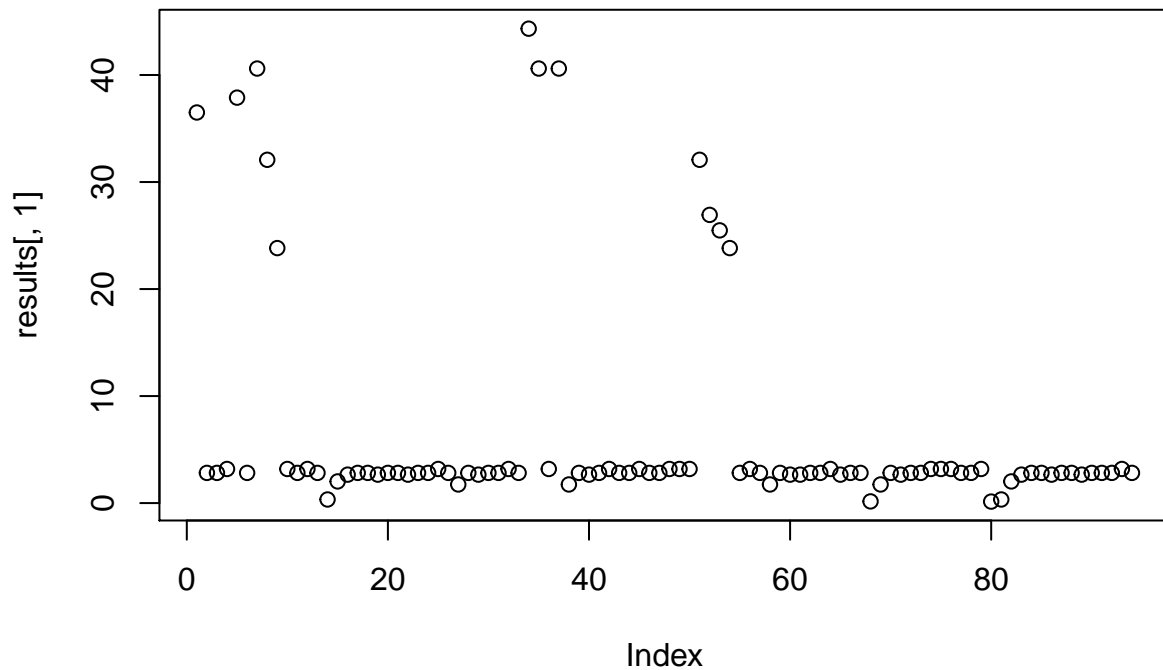


```
## 1 factor completed in 0.00441 minutes. Estimated time of completion: 2020-04-29 13:47:07 [1]
## [6] 3.160880e-16 4.000000e+01 1.000000e-06 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 50.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

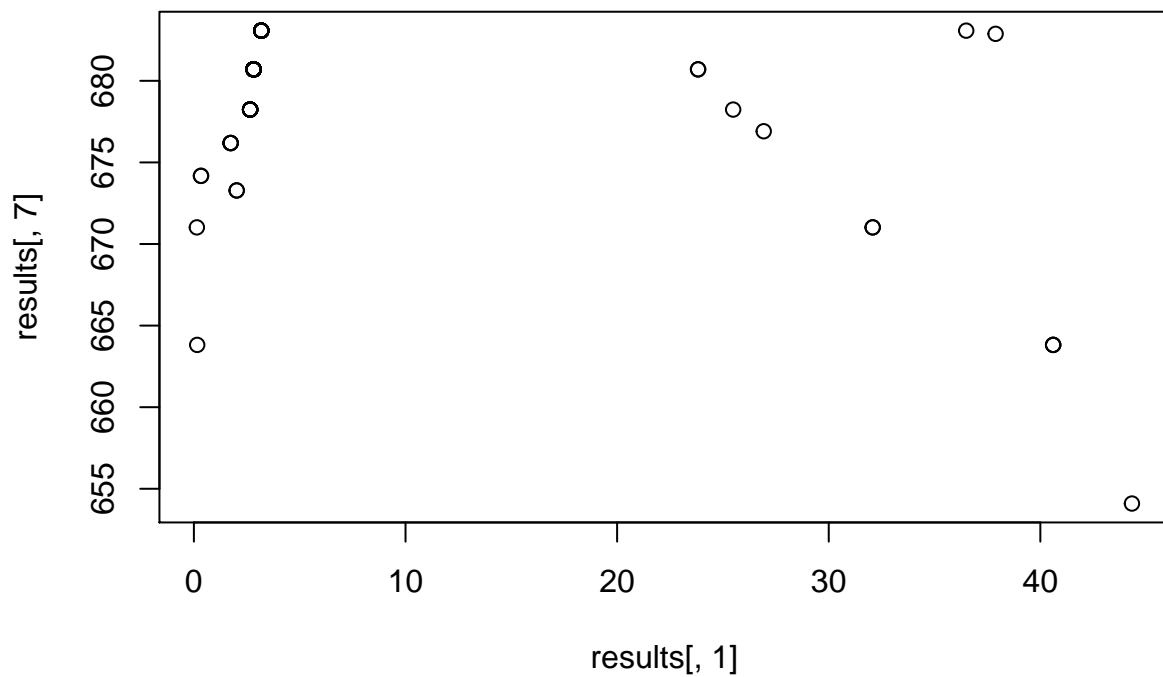


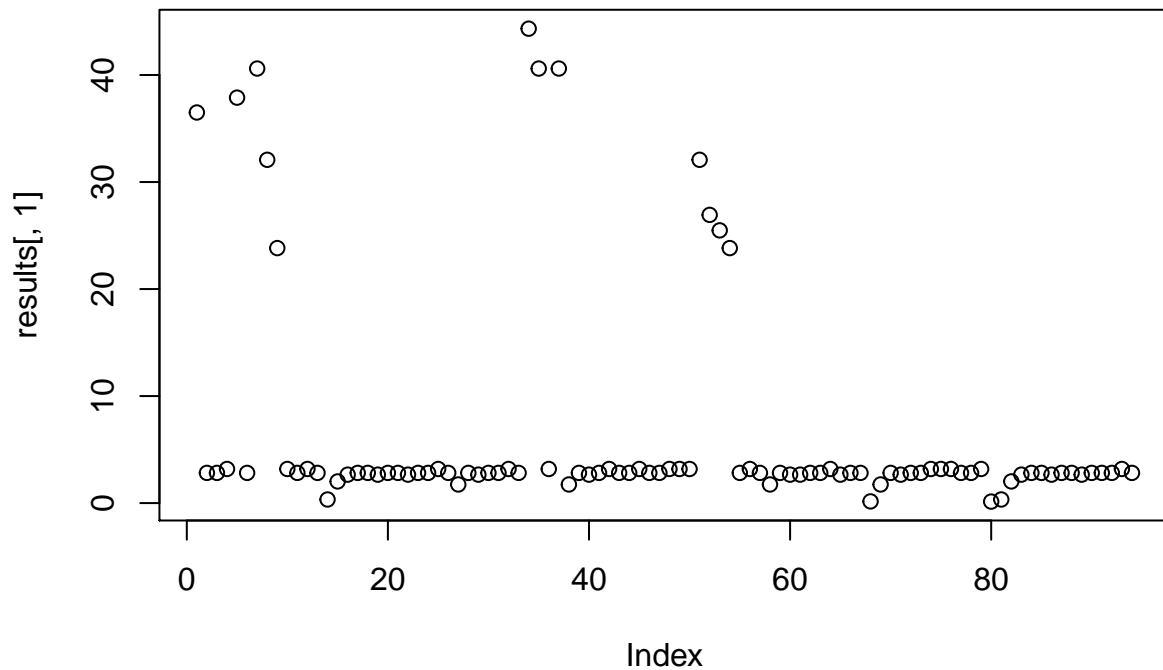


```
## 1 factor completed in 0.00419 minutes. Estimated time of completion: 2020-04-29 13:47:09 [1]
## [6] 1.332785e-18 5.000000e+01 1.000000e-06 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 60.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

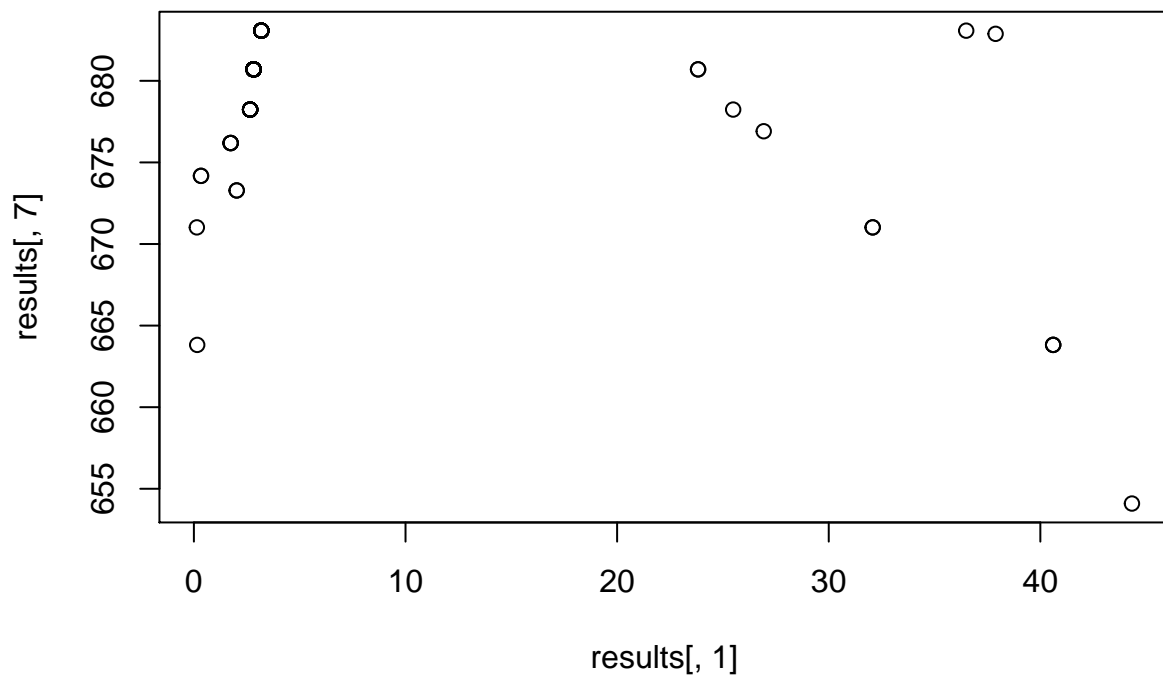


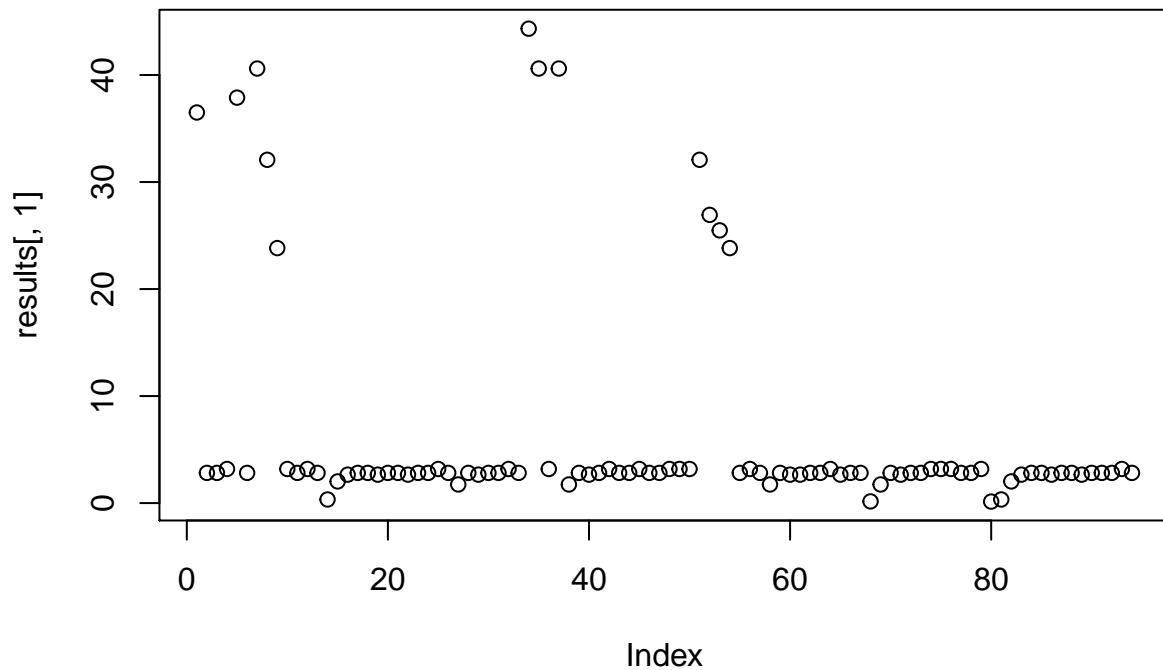


```
## 1 factor completed in 0.00426 minutes. Estimated time of completion: 2020-04-29 13:47:11 [1]
## [6] 5.619686e-21 6.000000e+01 1.000000e-06 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 70.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

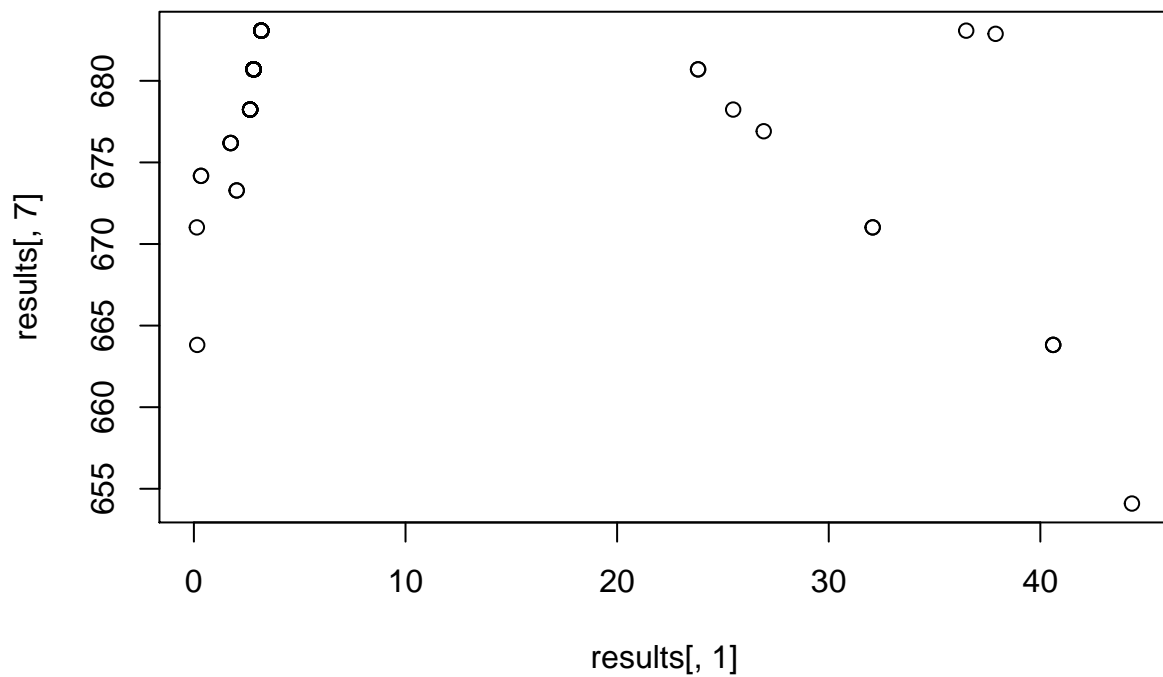


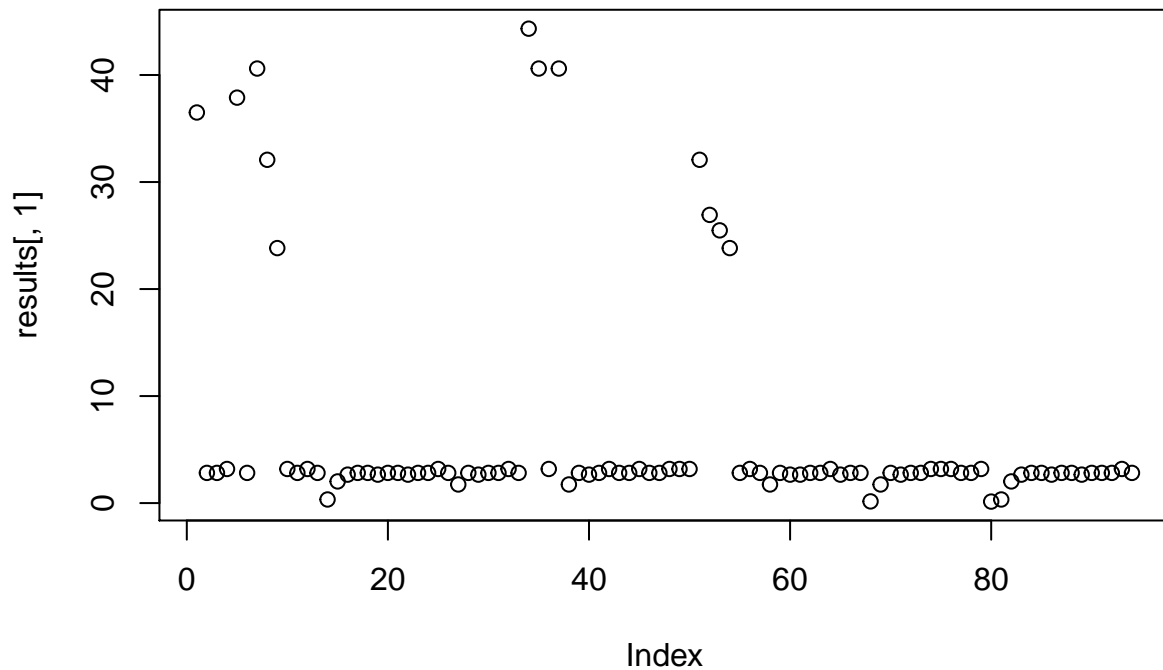


```
## 1 factor completed in 0.00429 minutes. Estimated time of completion: 2020-04-29 13:47:14 [1]
## [6] 2.369540e-23 7.000000e+01 1.000000e-06 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 80.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

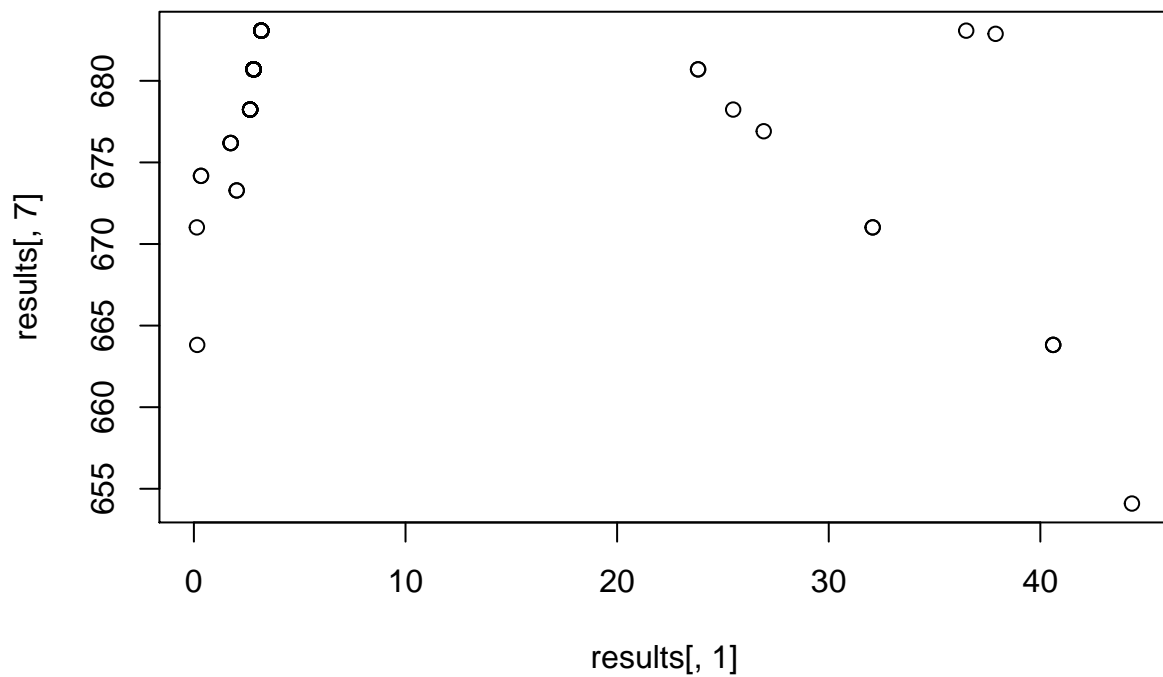


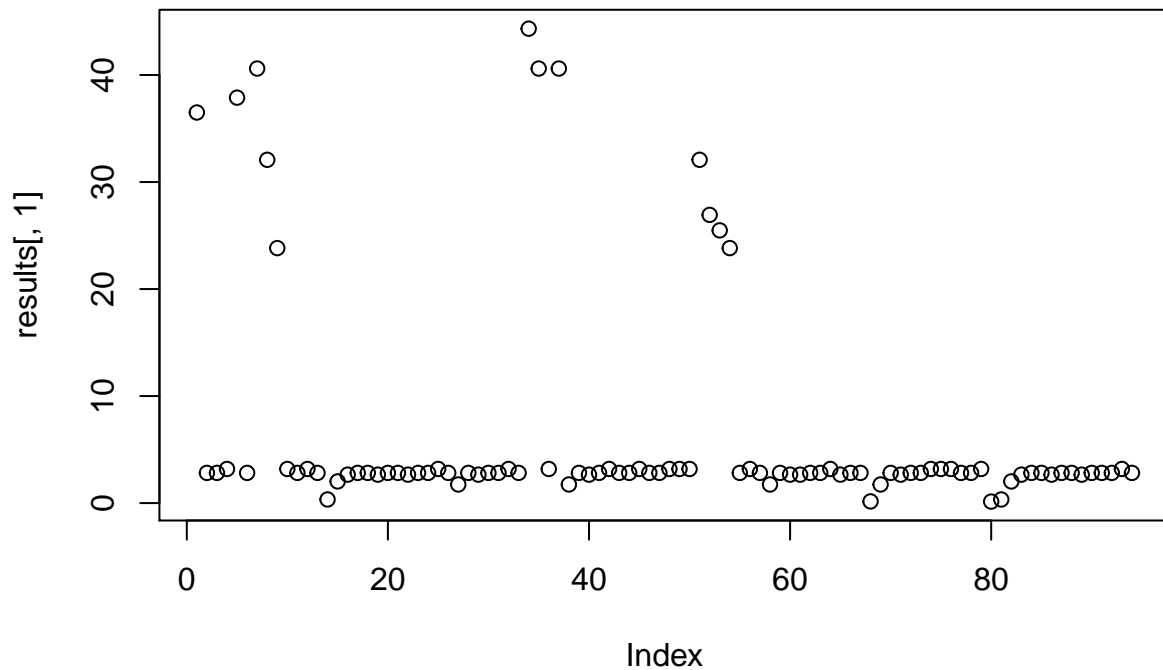


```
## 1 factor completed in 0.00402 minutes. Estimated time of completion: 2020-04-29 13:47:16 [1]
## [6] 9.991166e-26 8.000000e+01 1.000000e-06 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 90.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

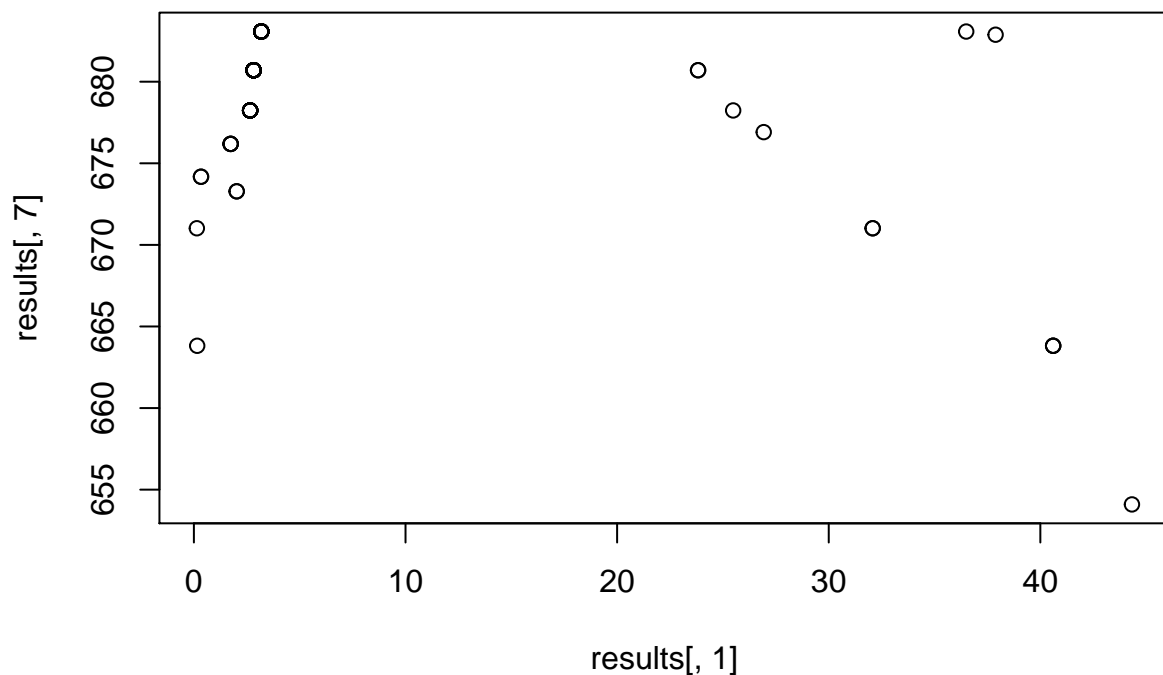


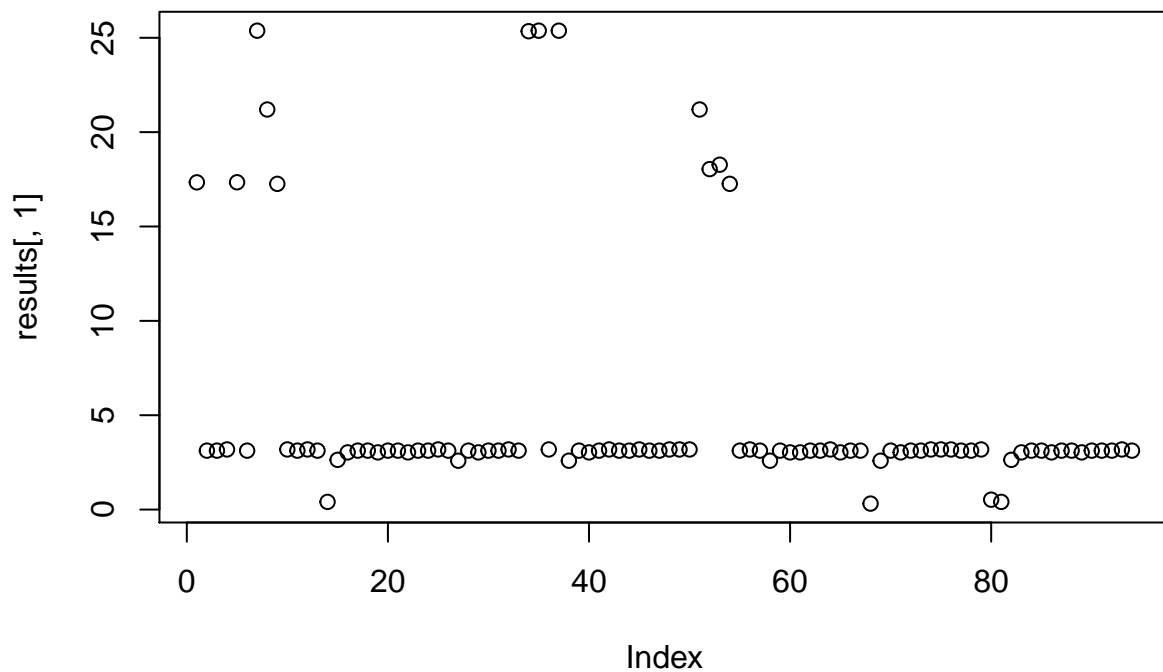


```
## 1 factor completed in 0.0041 minutes. Estimated time of completion: 2020-04-29 13:47:18 [1]
## [6] 4.212775e-28 9.000000e+01 1.000000e-06 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "X = 10.000001"
## [1] "j = 1e-06"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

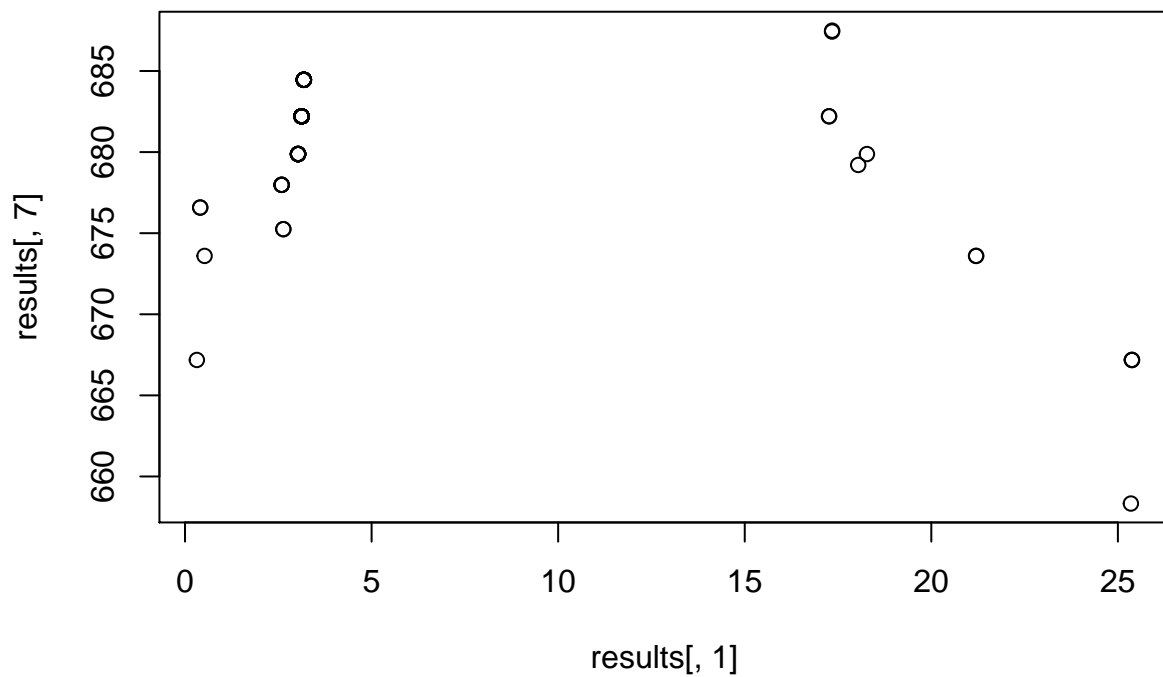


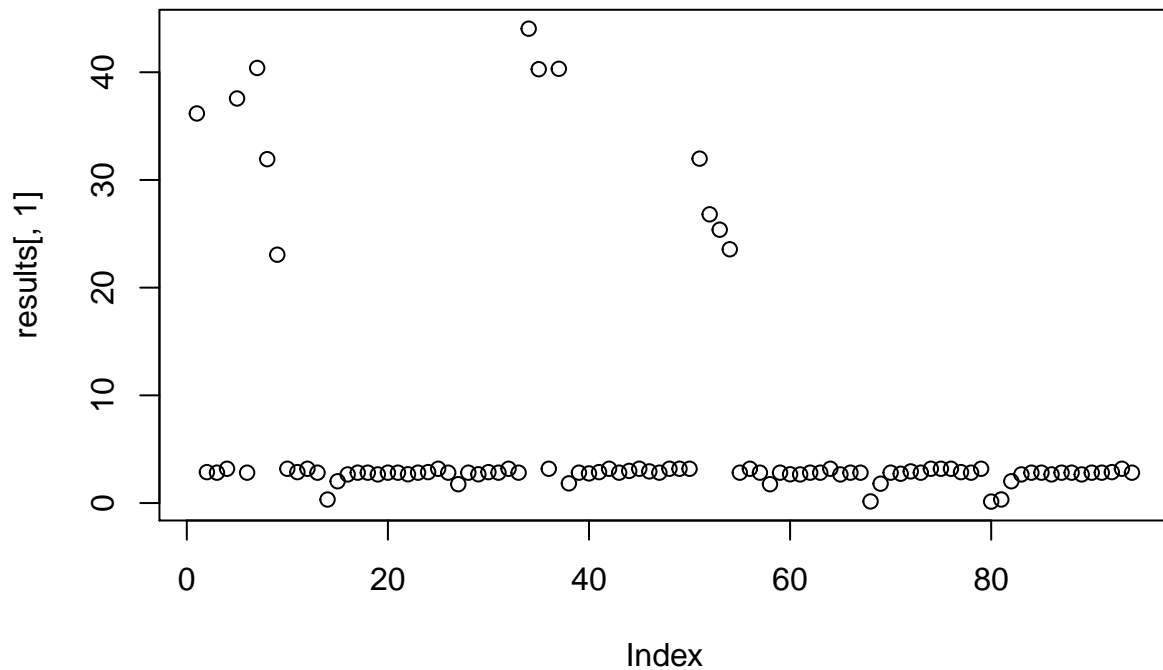


```
## 1 factor completed in 0.00392 minutes. Estimated time of completion: 2020-04-29 13:47:20 [1]
## [7] 0.0000010 10.0000010 7.8130568 667.1843727 25.3483148 658.3318744
## [13] 68.0000000 34.0000000
## [1] "j = 10.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

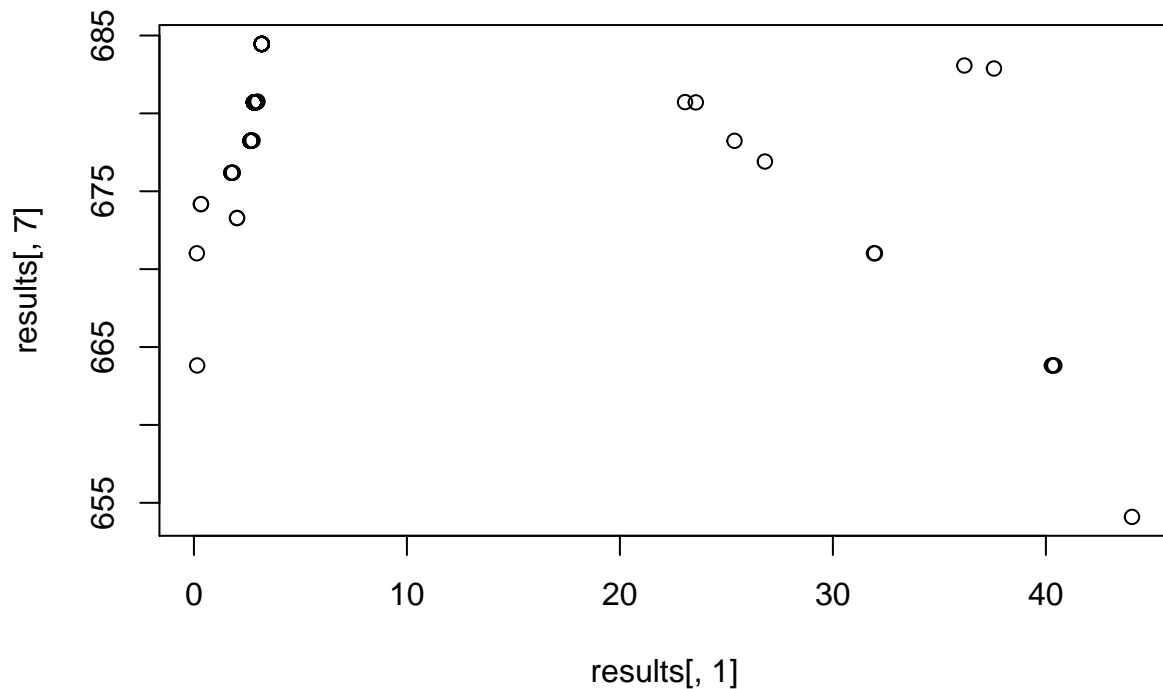


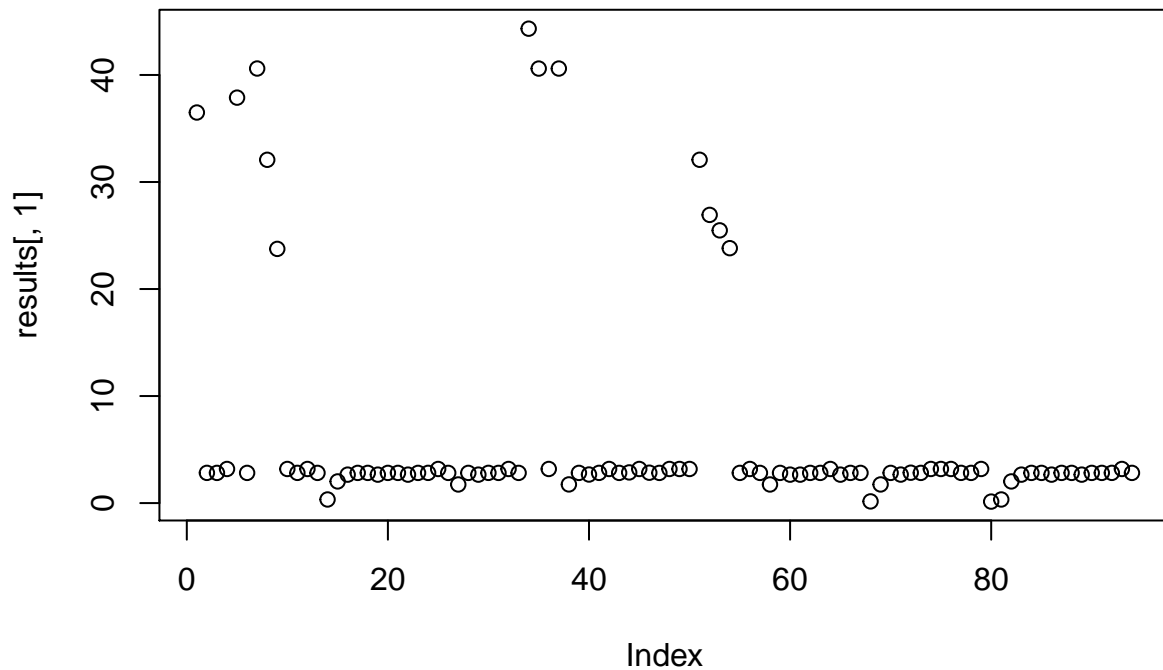


```
## 1 factor completed in 0.0052 minutes. Estimated time of completion: 2020-04-29 13:47:22 [1]
## [6] 0.04216498 10.00000100 10.00000100 7.81305685 671.01965731
## [11] 44.04399056 654.09641676 80.00000000 34.00000000
## [1] "j = 20.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

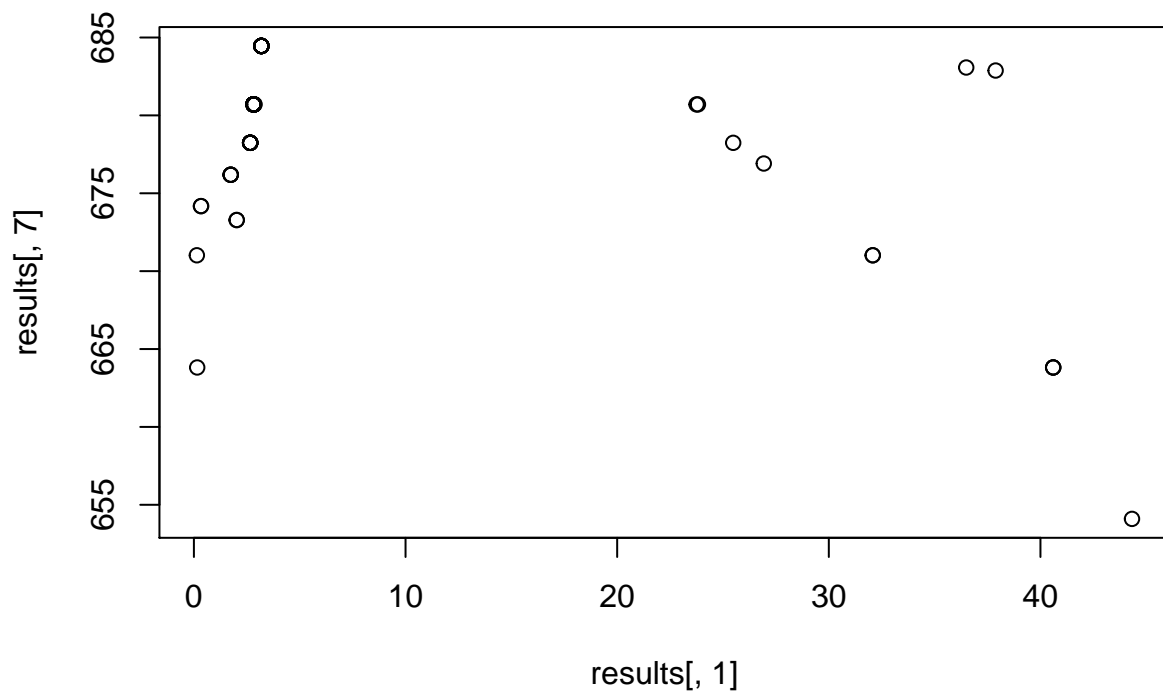


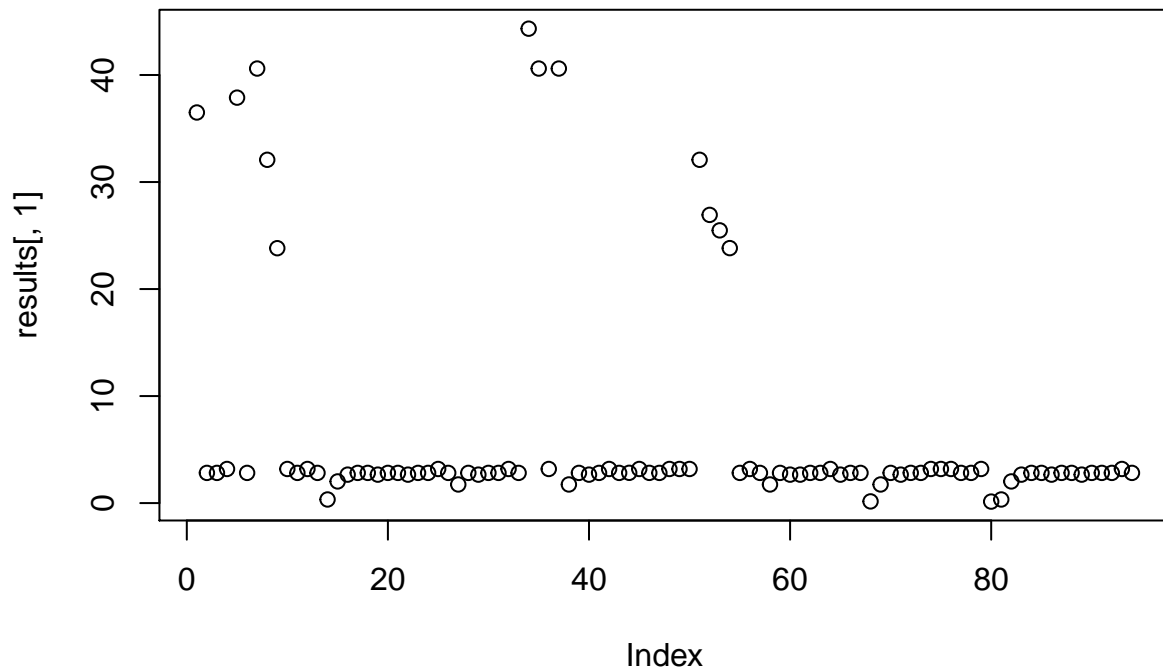


```
## 1 factor completed in 0.00409 minutes. Estimated time of completion: 2020-04-29 13:47:24 [1]
## [6] 1.777886e-04 2.000000e+01 1.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432578e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 30.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

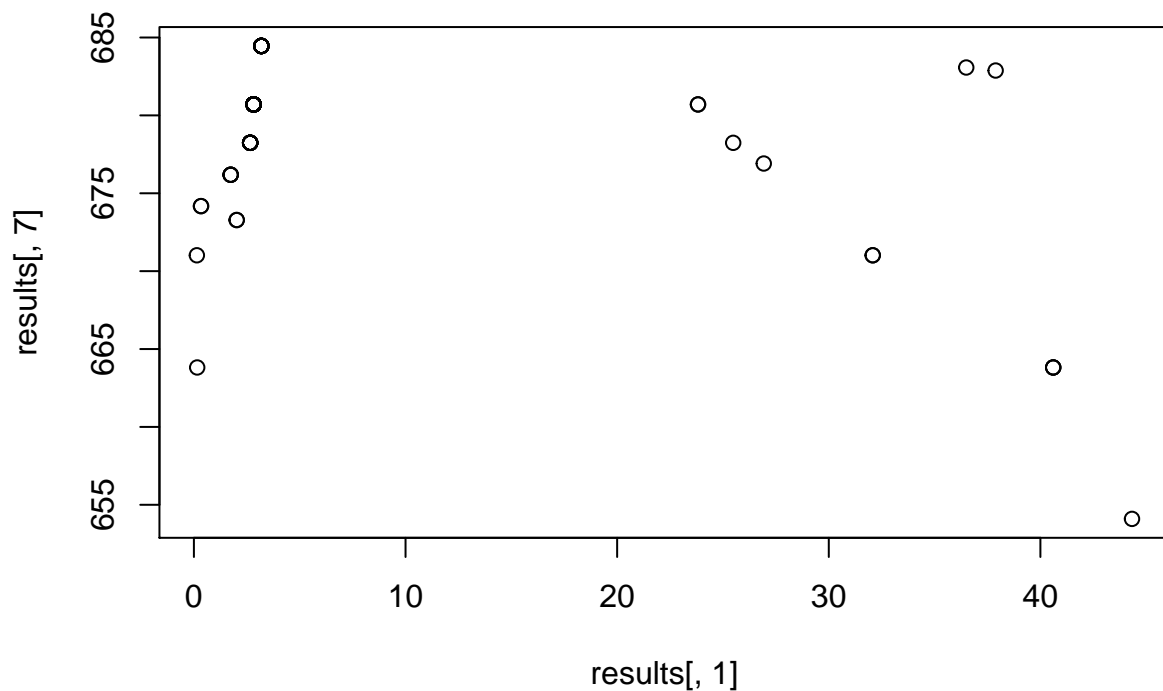


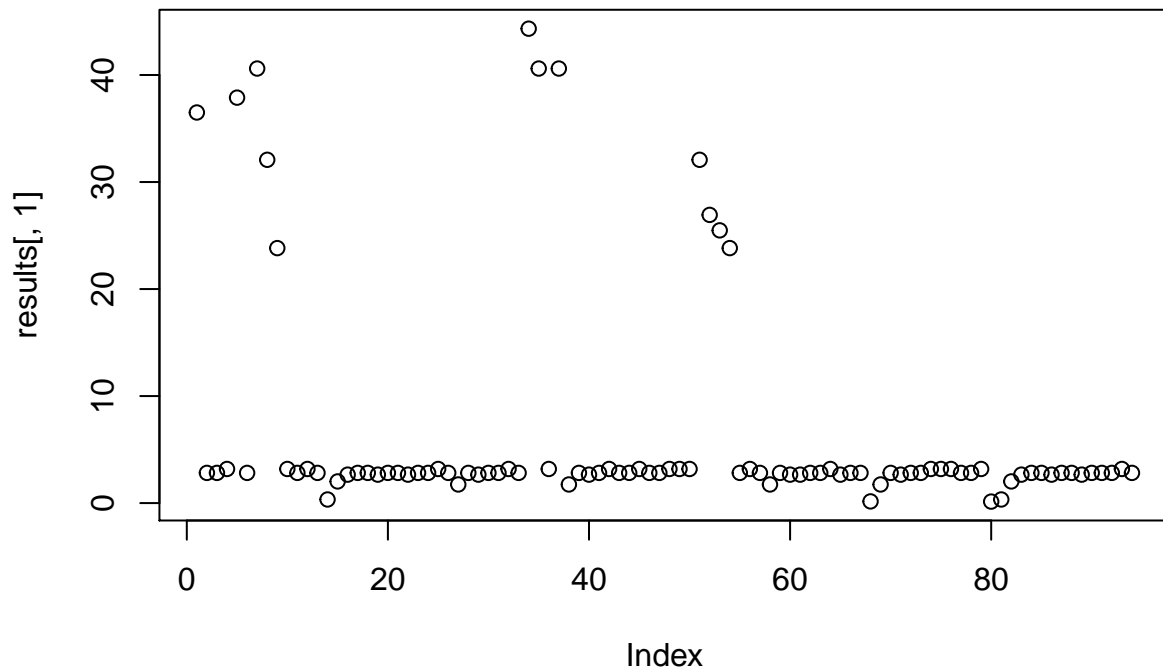


```
## 1 factor completed in 0.00384 minutes. Estimated time of completion: 2020-04-29 13:47:26 [1]
## [6] 7.496455e-07 3.000000e+01 1.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432876e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 40.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

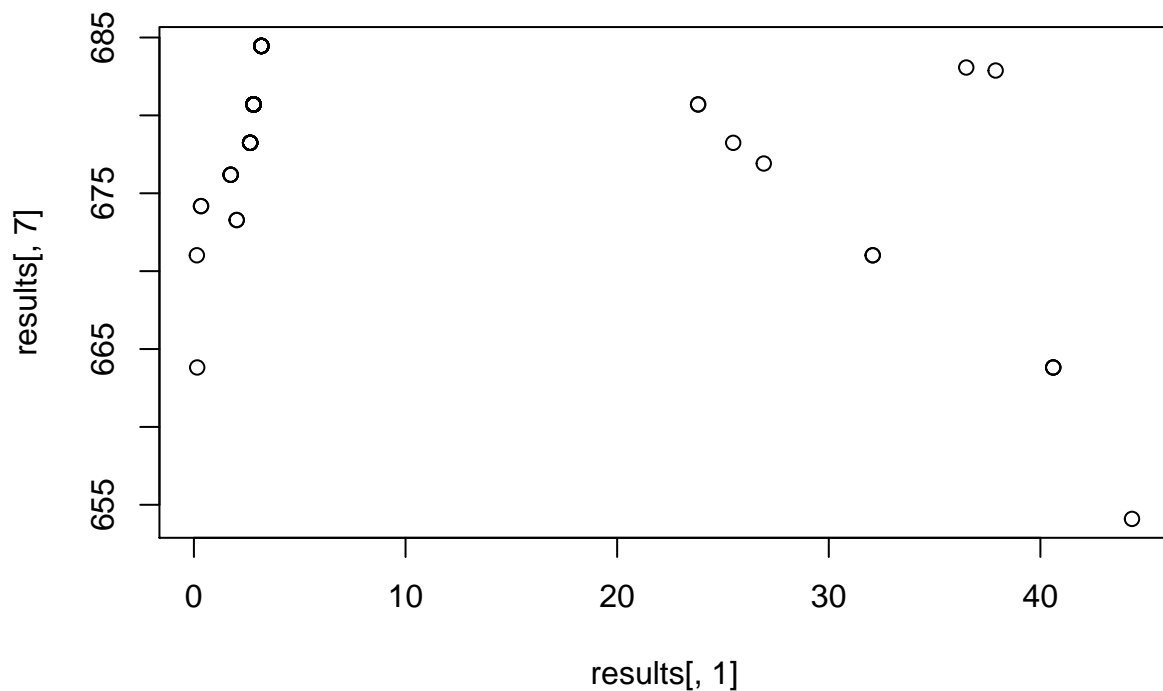


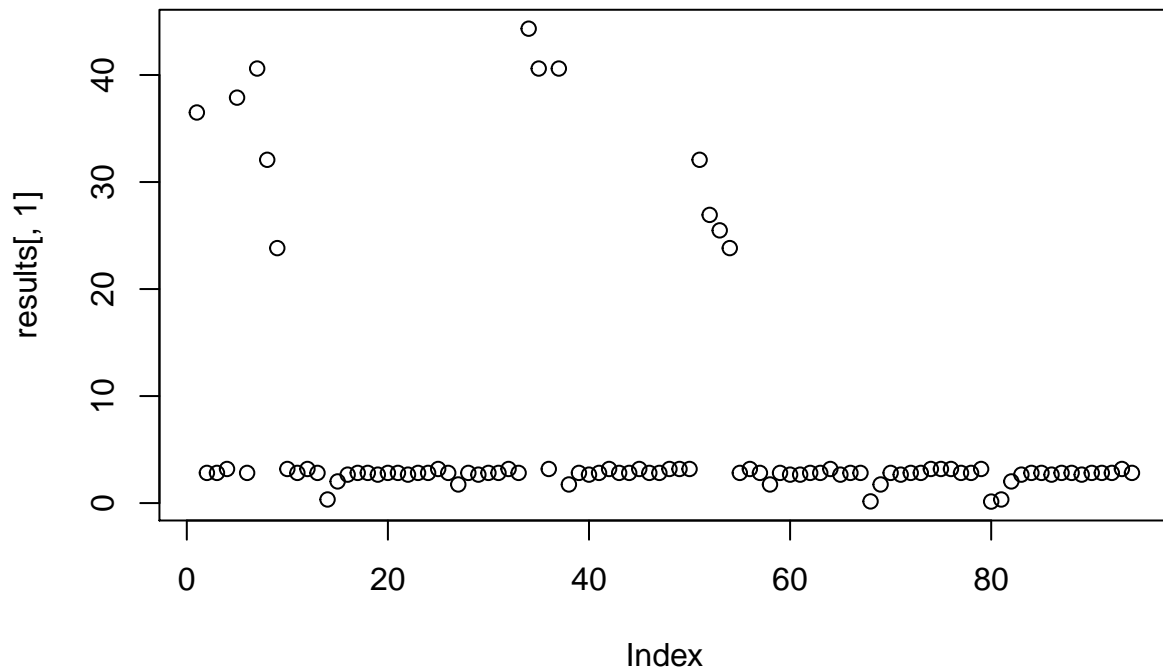


```
## 1 factor completed in 0.00394 minutes. Estimated time of completion: 2020-04-29 13:47:28 [1]
## [6] 3.160880e-09 4.000000e+01 1.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 50.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

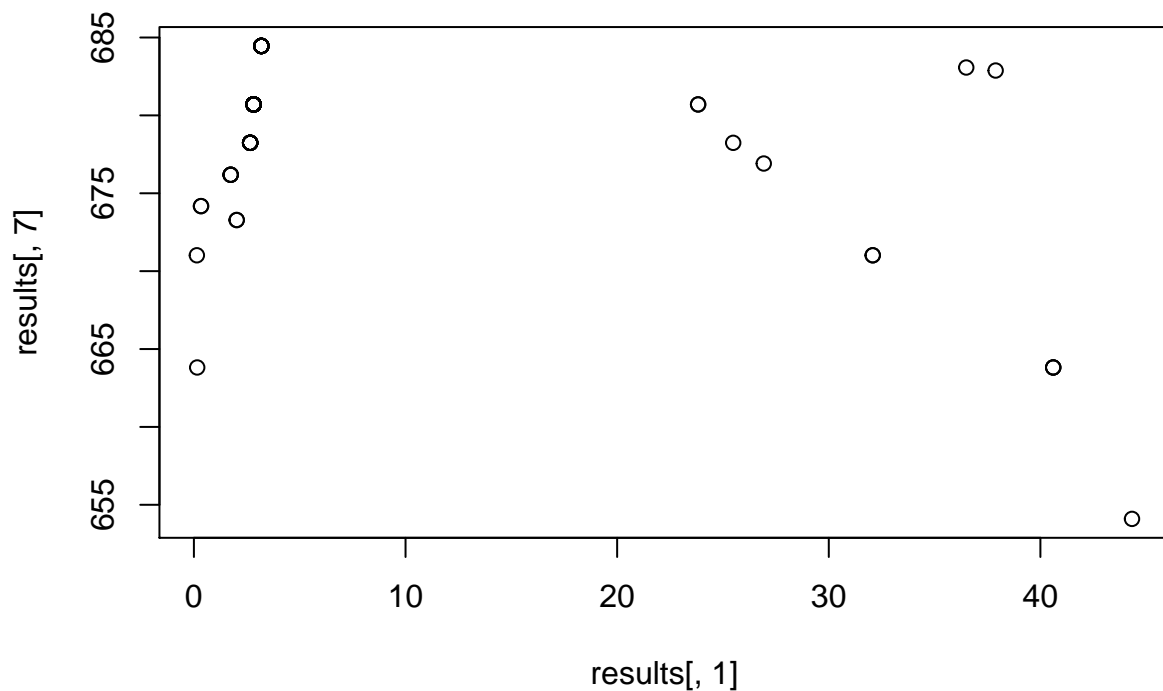


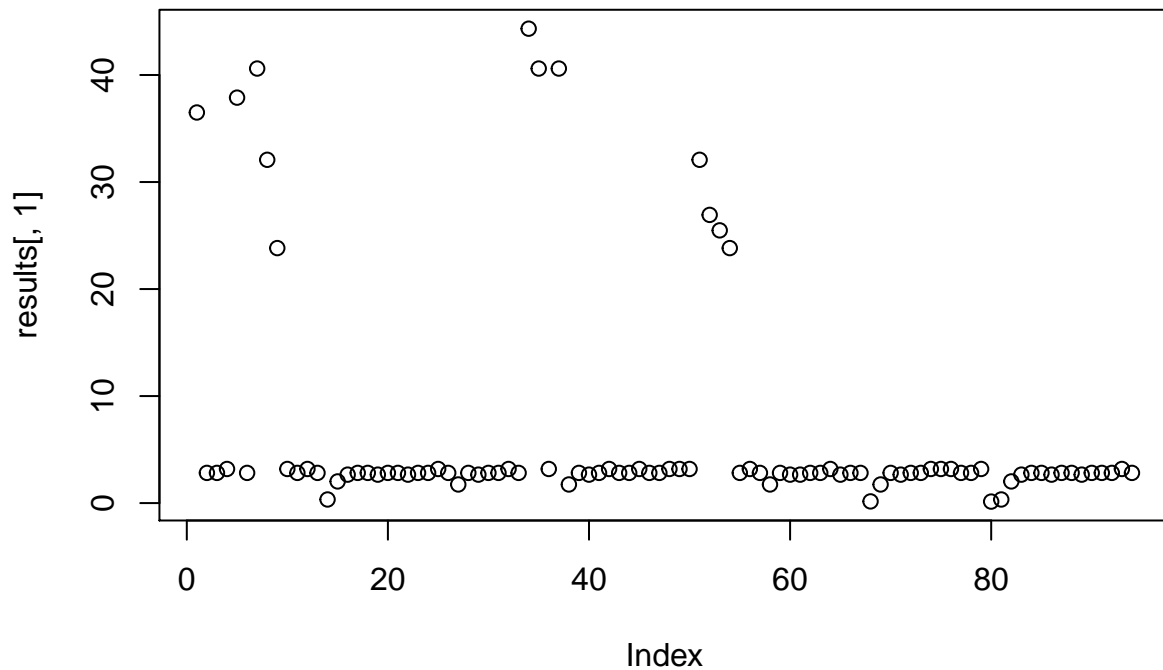


```
## 1 factor completed in 0.00414 minutes. Estimated time of completion: 2020-04-29 13:47:30 [1]
## [6] 1.332785e-11 5.000000e+01 1.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 60.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

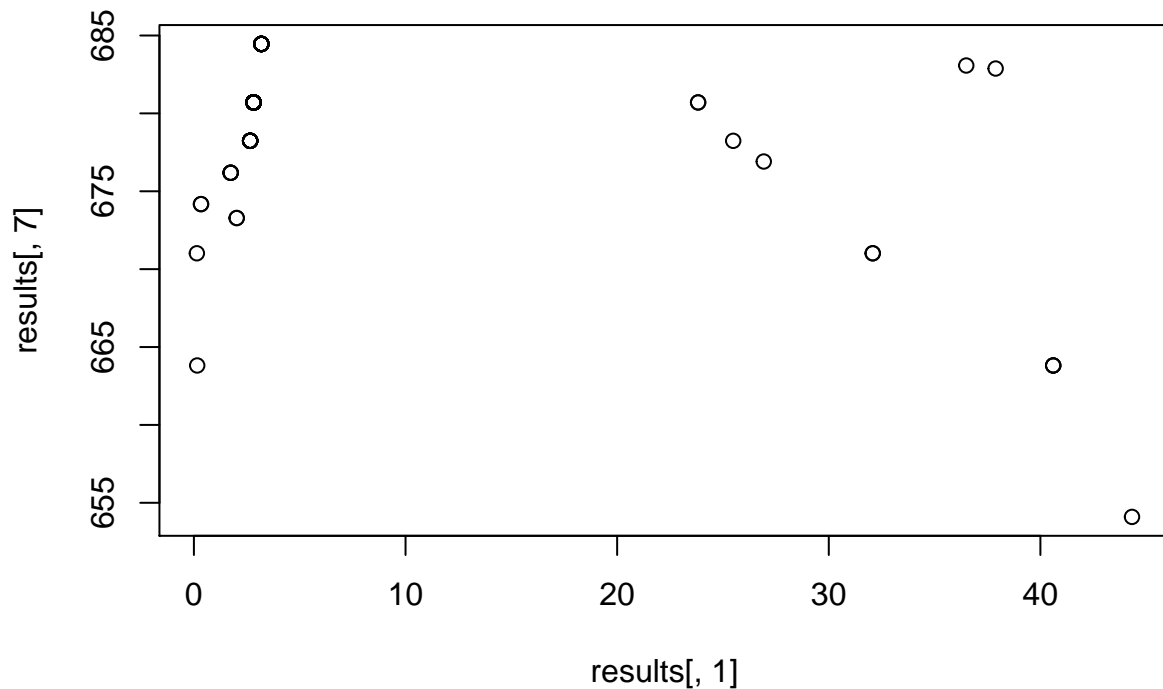


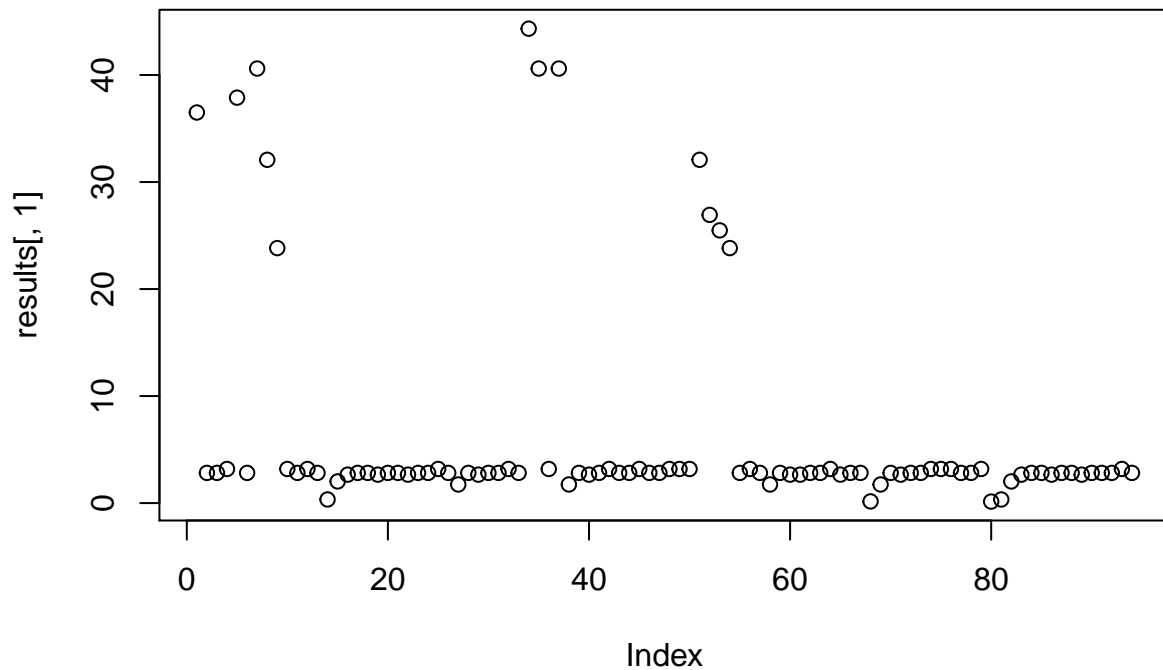


```
## 1 factor completed in 0.00404 minutes. Estimated time of completion: 2020-04-29 13:47:32 [1]
## [6] 5.619687e-14 6.000000e+01 1.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 70.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

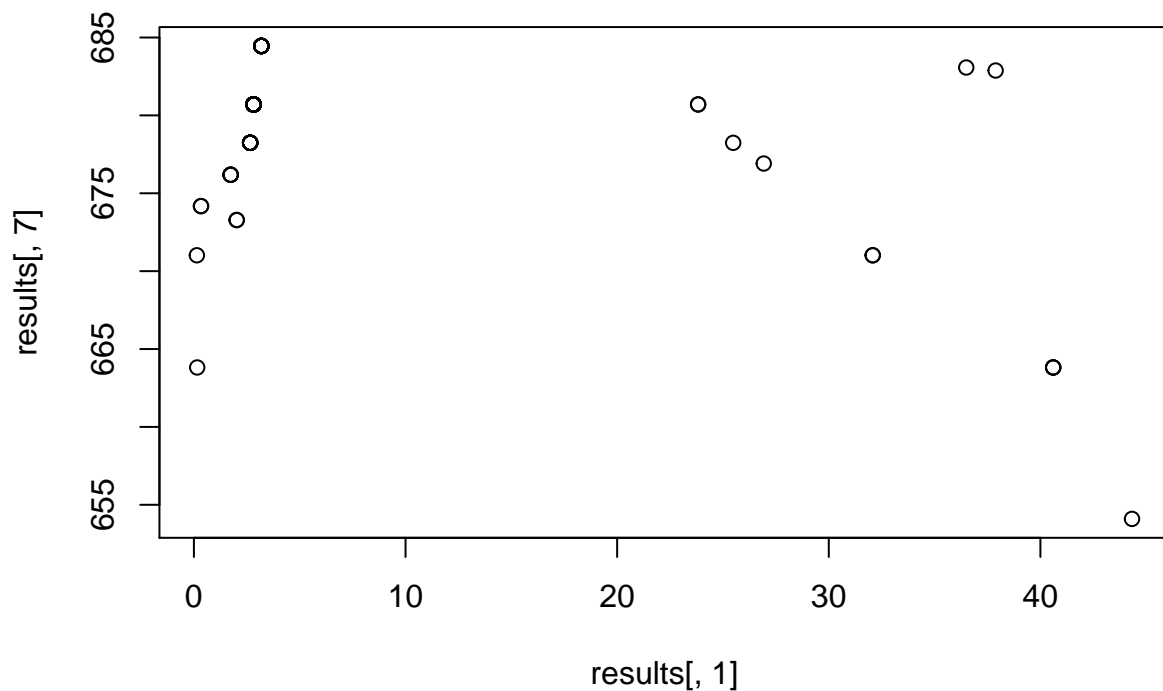


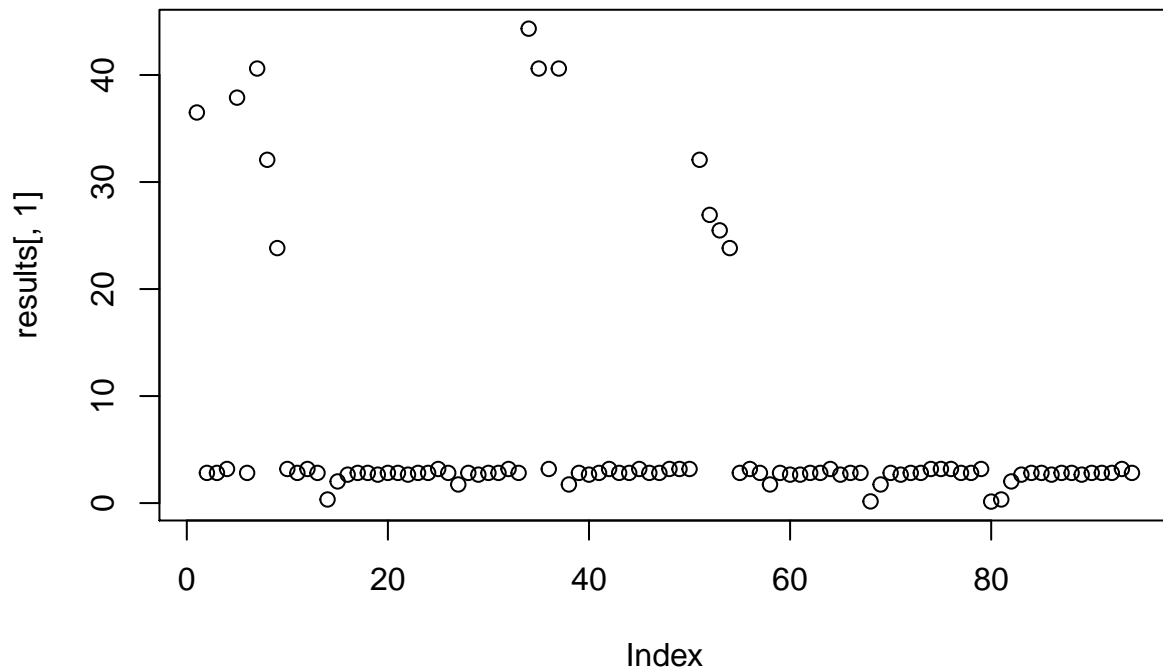


```
## 1 factor completed in 0.00689 minutes. Estimated time of completion: 2020-04-29 13:47:35 [1]
## [6] 2.369541e-16 7.000000e+01 1.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 80.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

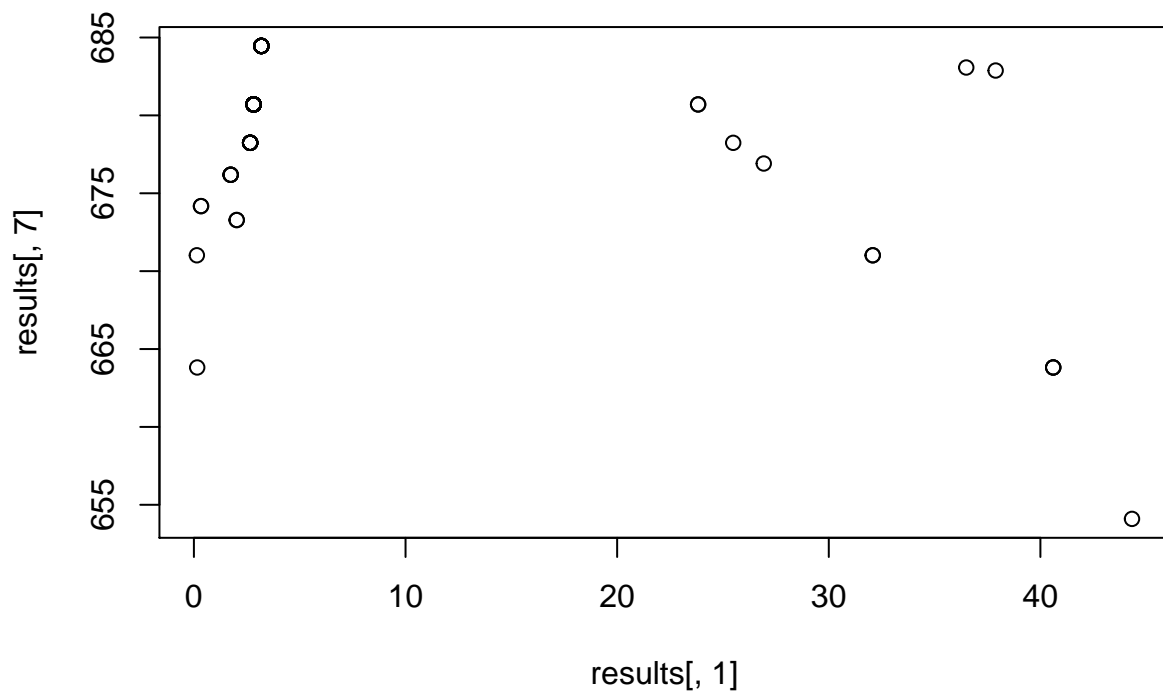


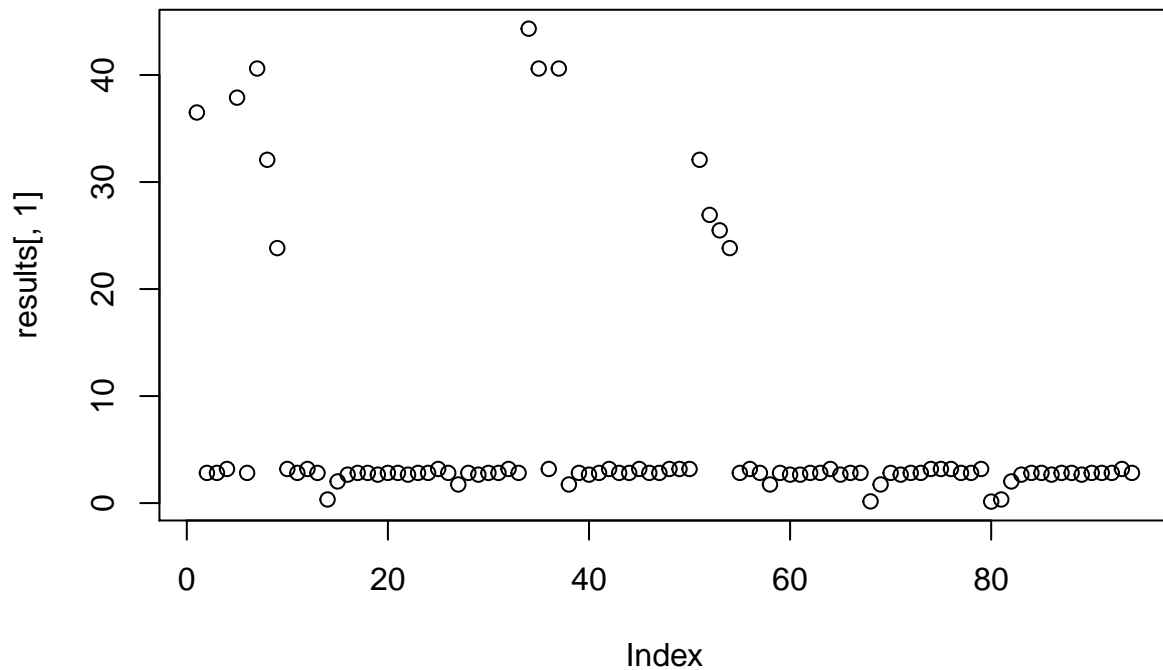


```
## 1 factor completed in 0.00438 minutes. Estimated time of completion: 2020-04-29 13:47:37 [1]
## [6] 9.991167e-19 8.000000e+01 1.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 90.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

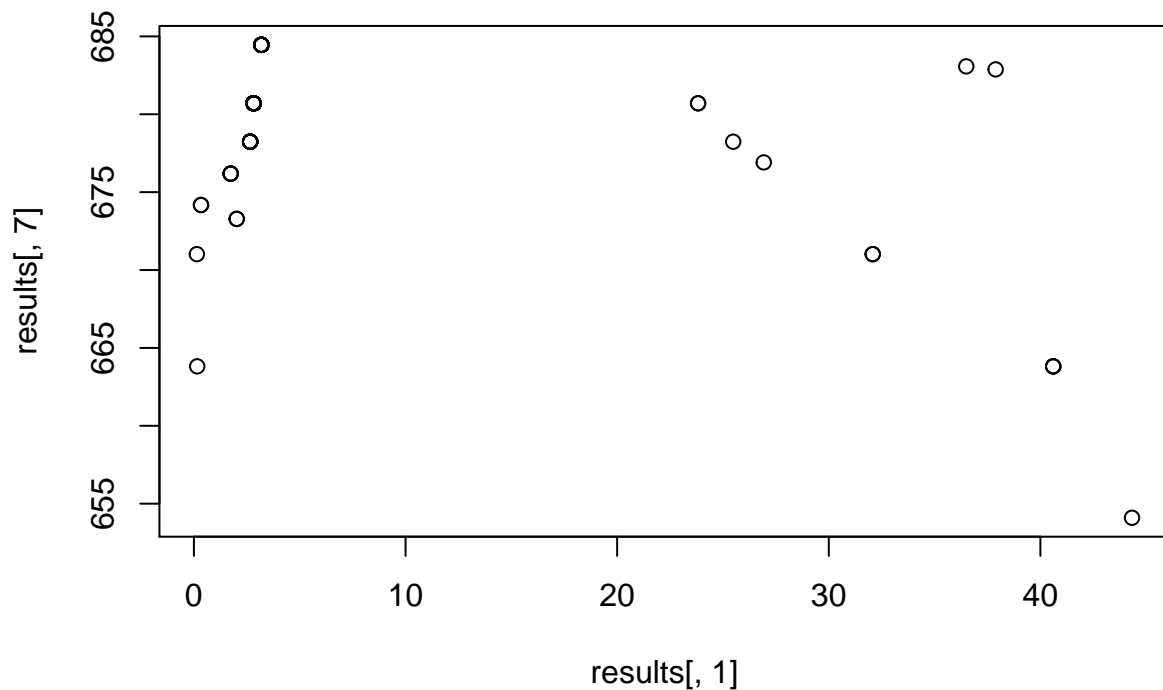


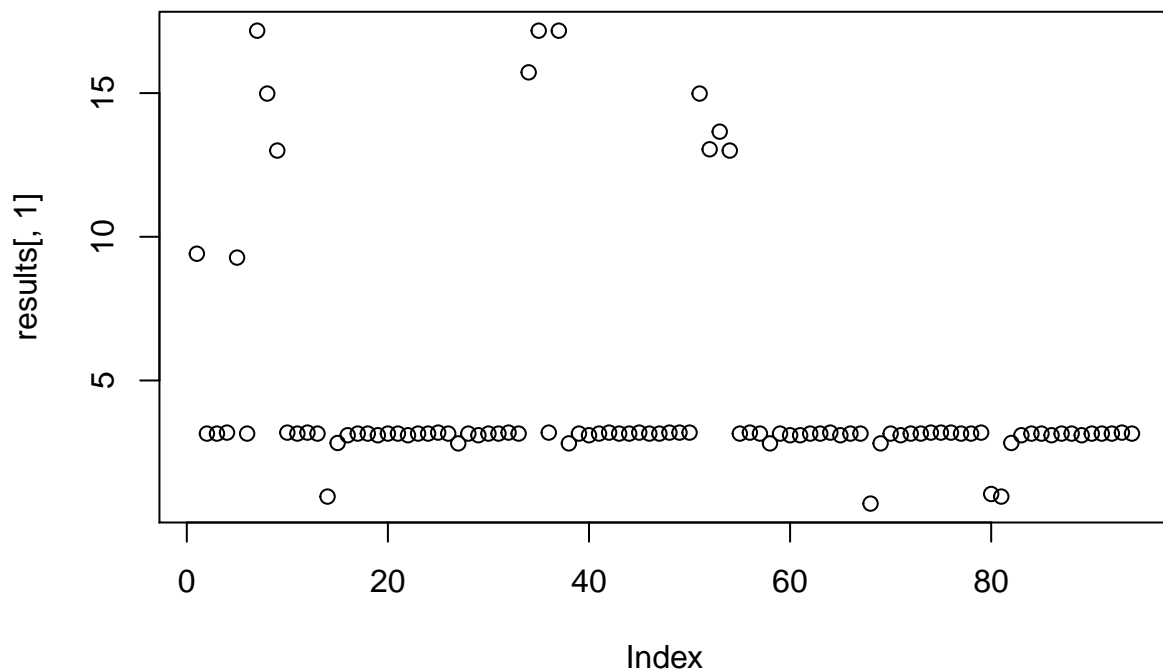


```
## 1 factor completed in 0.00389 minutes. Estimated time of completion: 2020-04-29 13:47:39 [1]
## [6] 4.212775e-21 9.000000e+01 1.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "X = 20.000001"
## [1] "j = 1e-06"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

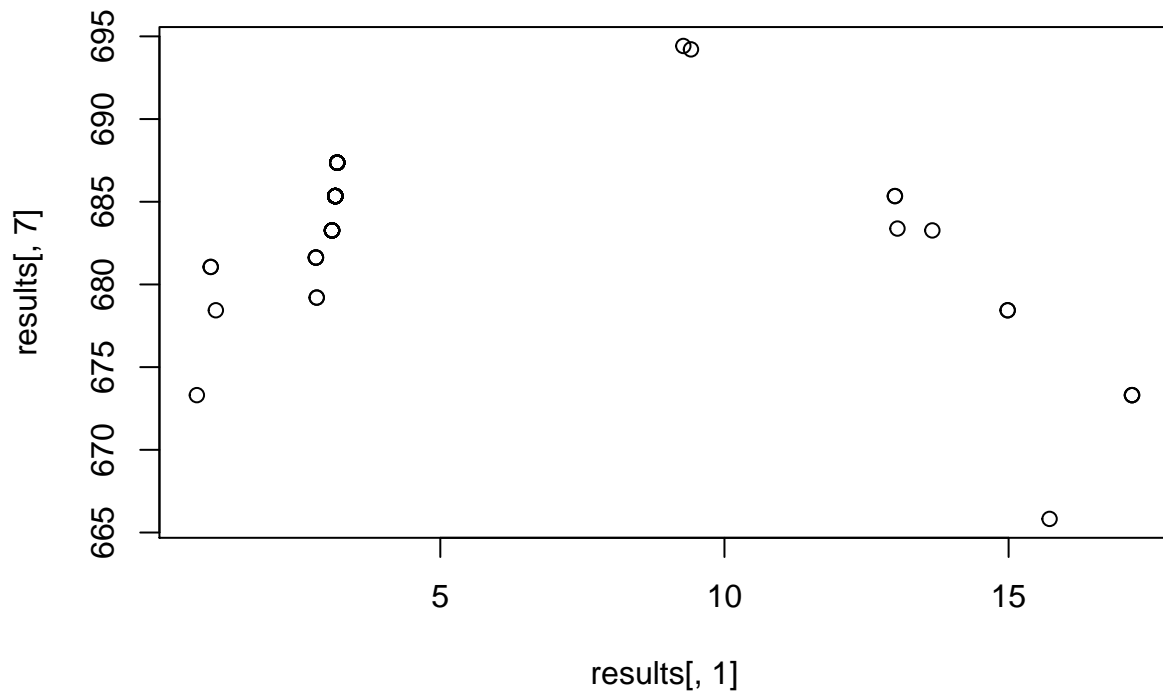


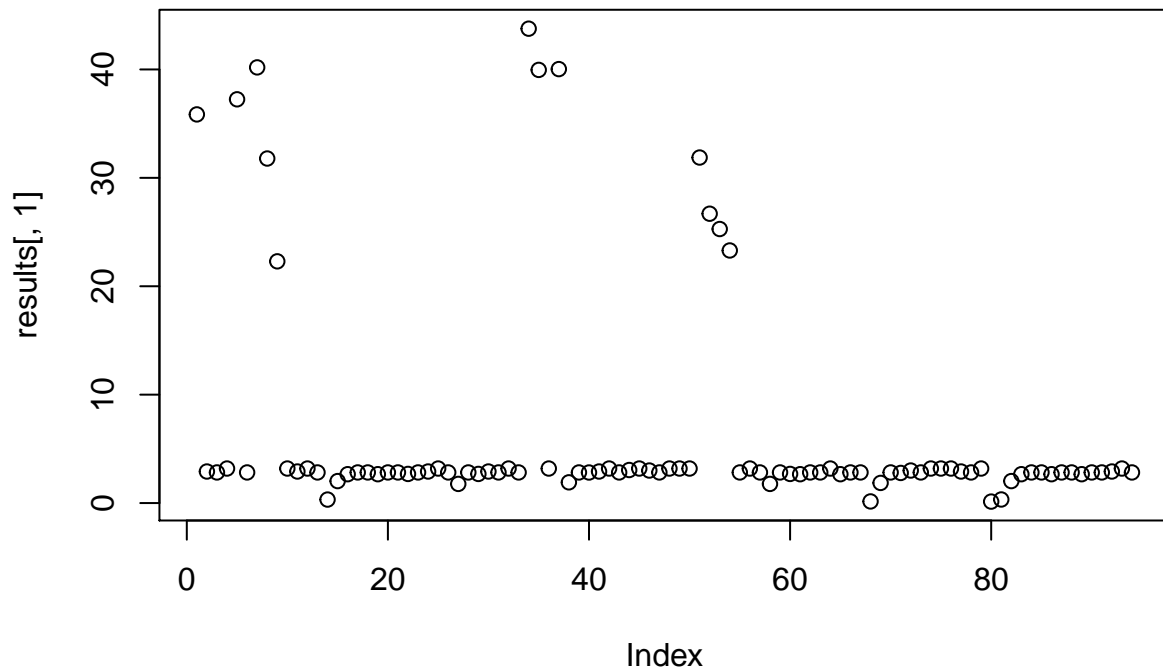


```
## 1 factor completed in 0.00403 minutes. Estimated time of completion: 2020-04-29 13:47:41 [1]
## [7] 0.0000010 20.0000010 7.8130568 673.3074576 15.7211121 665.8244292
## [13] 68.0000000 34.0000000
## [1] "j = 10.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

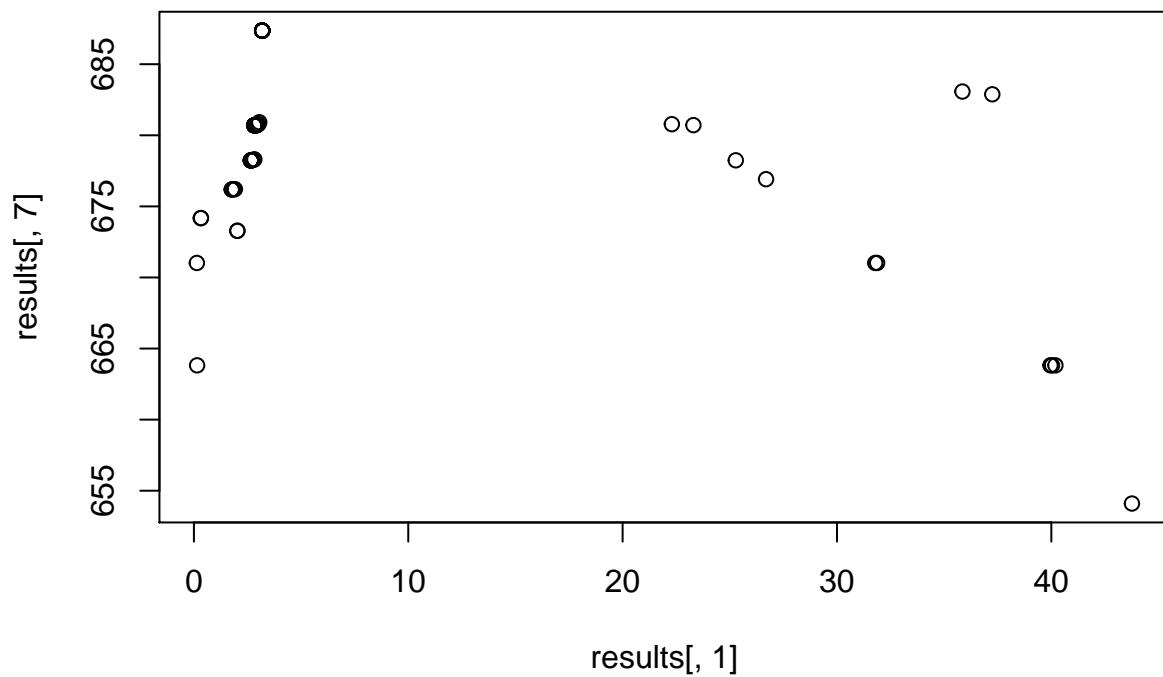


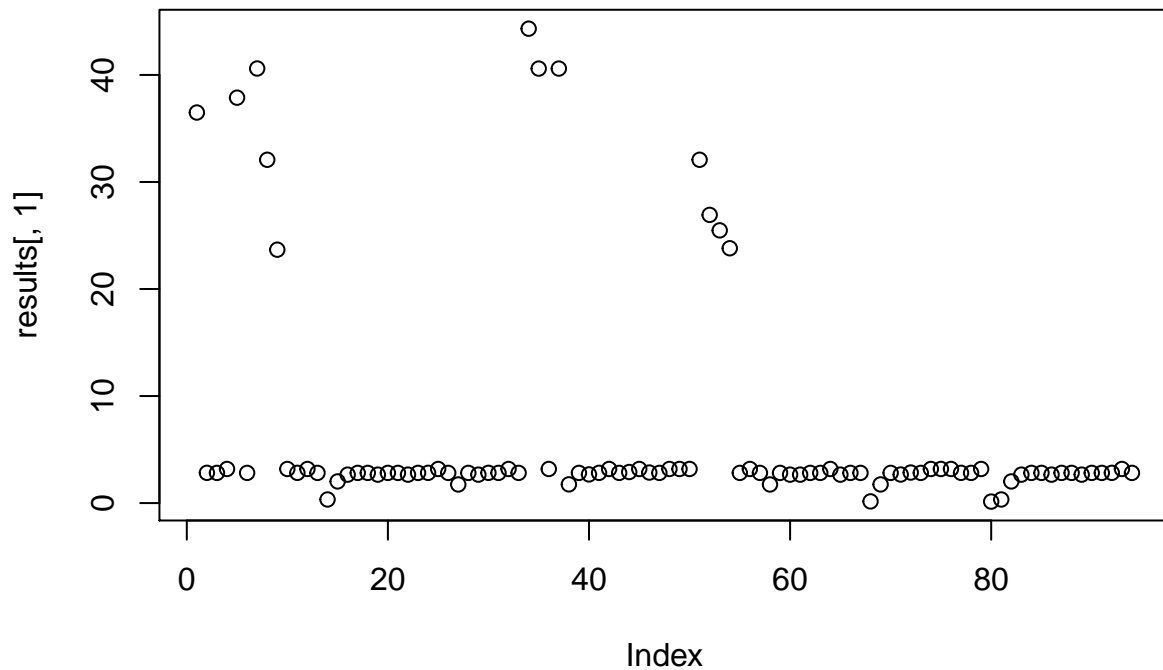


```
## 1 factor completed in 0.0039 minutes. Estimated time of completion: 2020-04-29 13:47:43 [1]
## [6] 0.08432995 10.00000100 20.00000100 7.81305685 671.01986817
## [11] 43.76163459 654.09860321 80.00000000 34.00000000
## [1] "j = 20.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

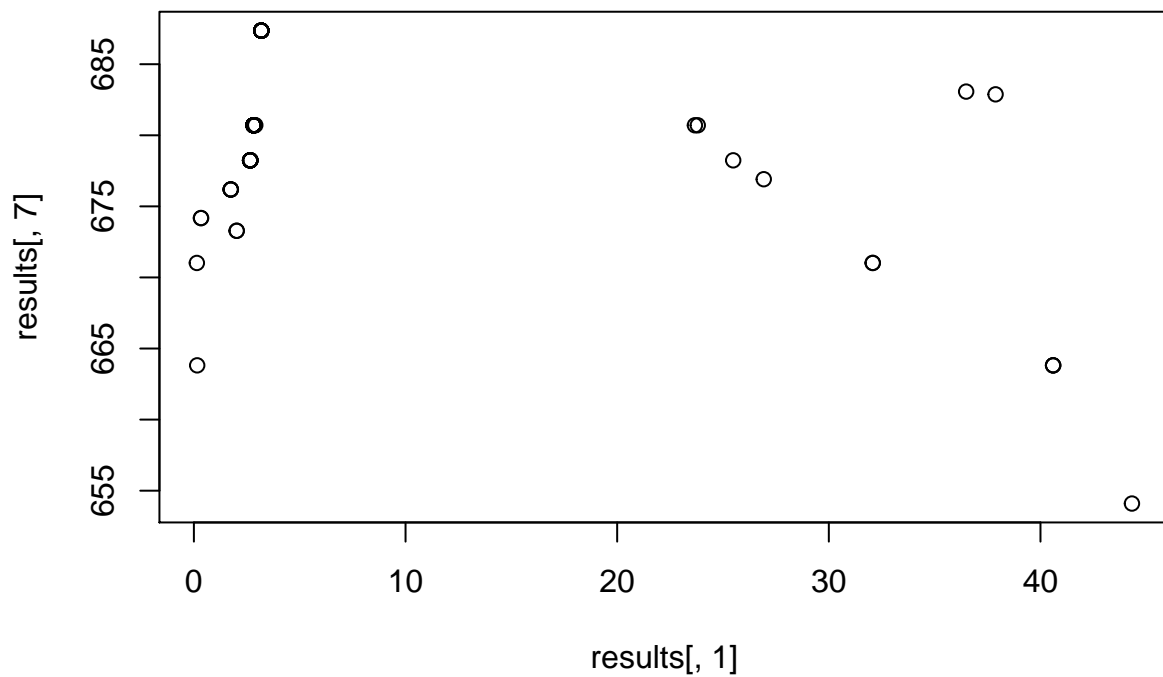


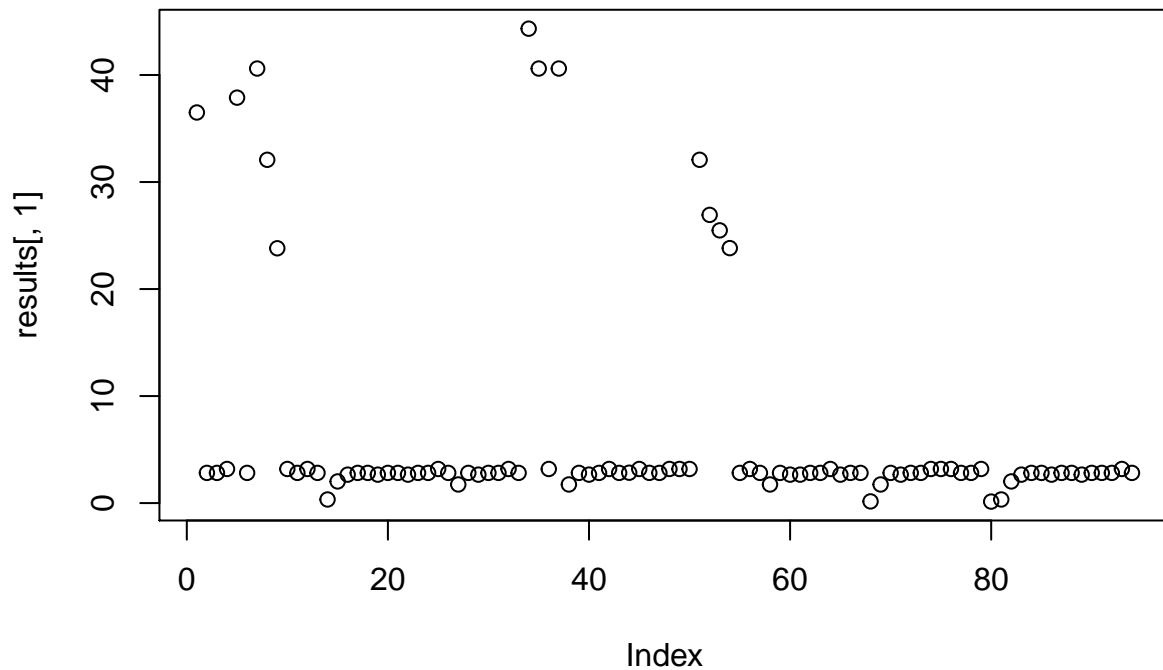


```
## 1 factor completed in 0.00392 minutes. Estimated time of completion: 2020-04-29 13:47:45 [1]
## [6] 3.555772e-04 2.000000e+01 2.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432277e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 30.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

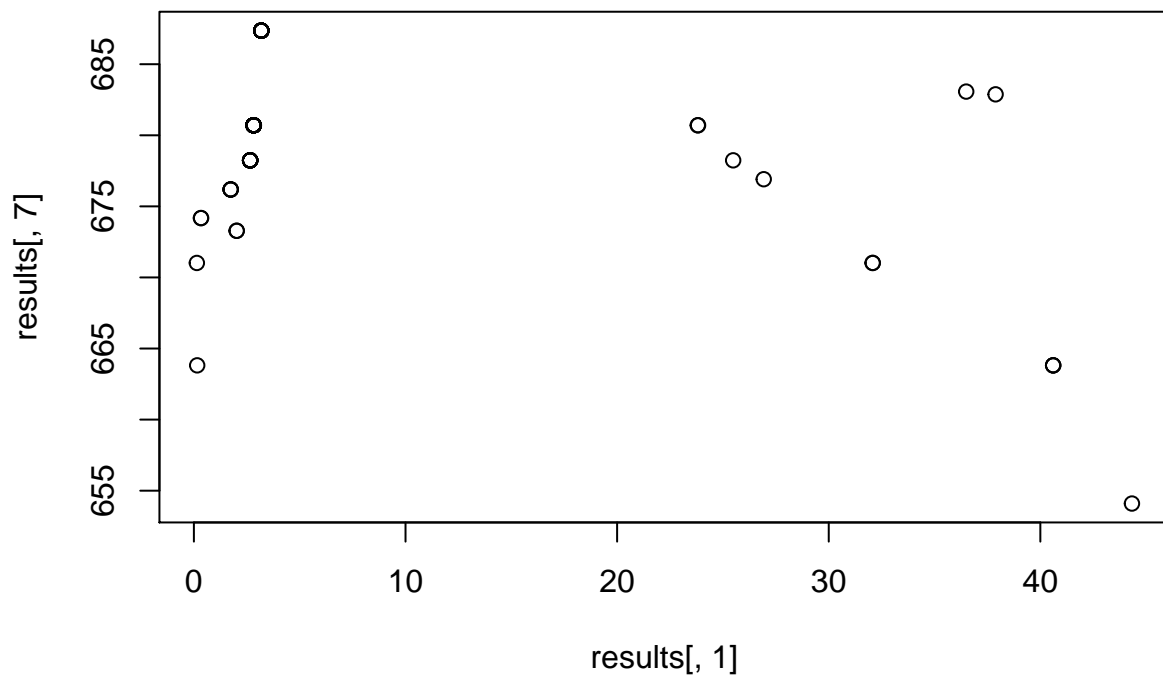


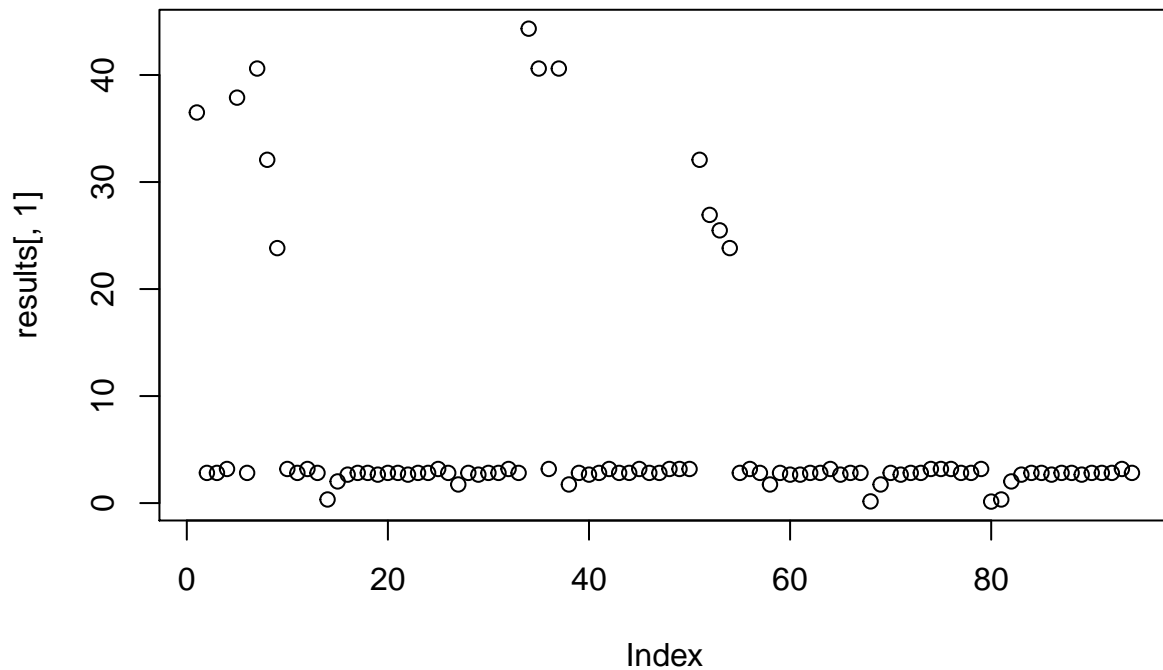


```
## 1 factor completed in 0.00462 minutes. Estimated time of completion: 2020-04-29 13:47:47 [1]
## [6] 1.499291e-06 3.000000e+01 2.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432873e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 40.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

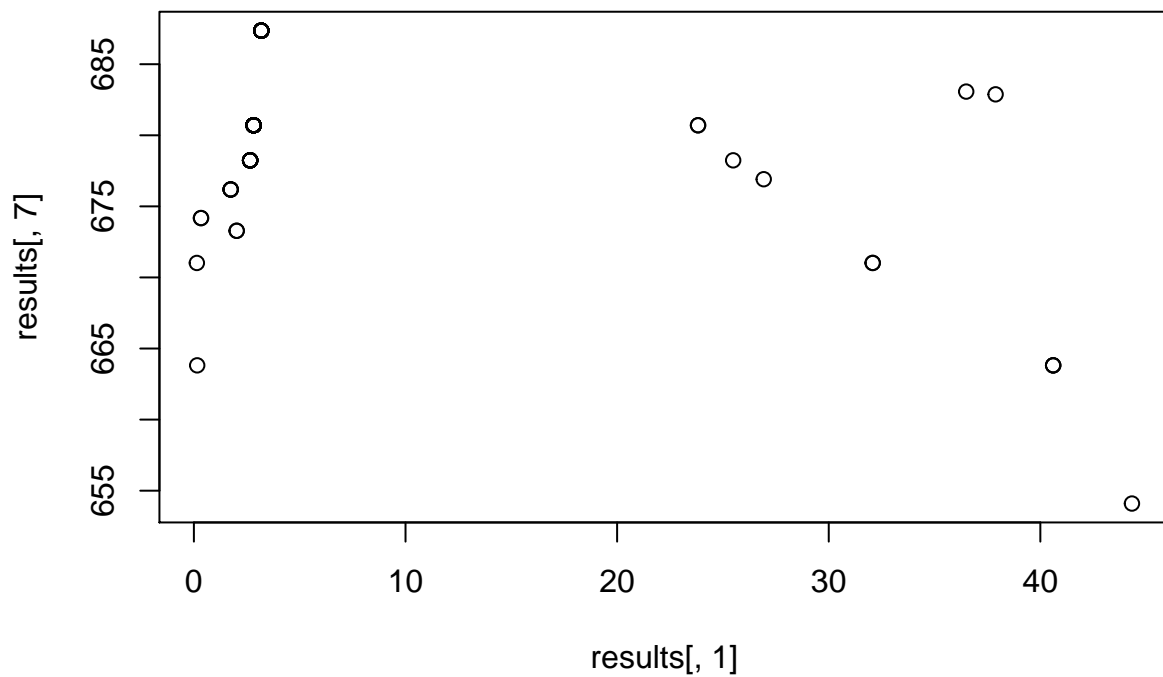


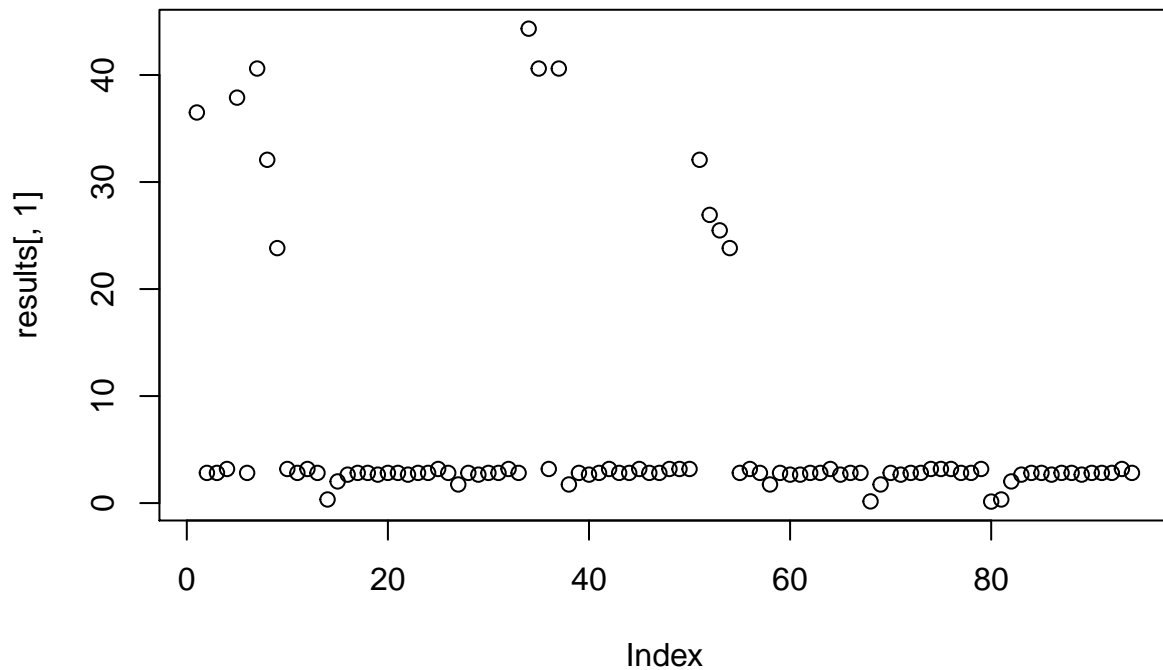


```
## 1 factor completed in 0.00405 minutes. Estimated time of completion: 2020-04-29 13:47:49 [1]
## [6] 6.321760e-09 4.000000e+01 2.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 50.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

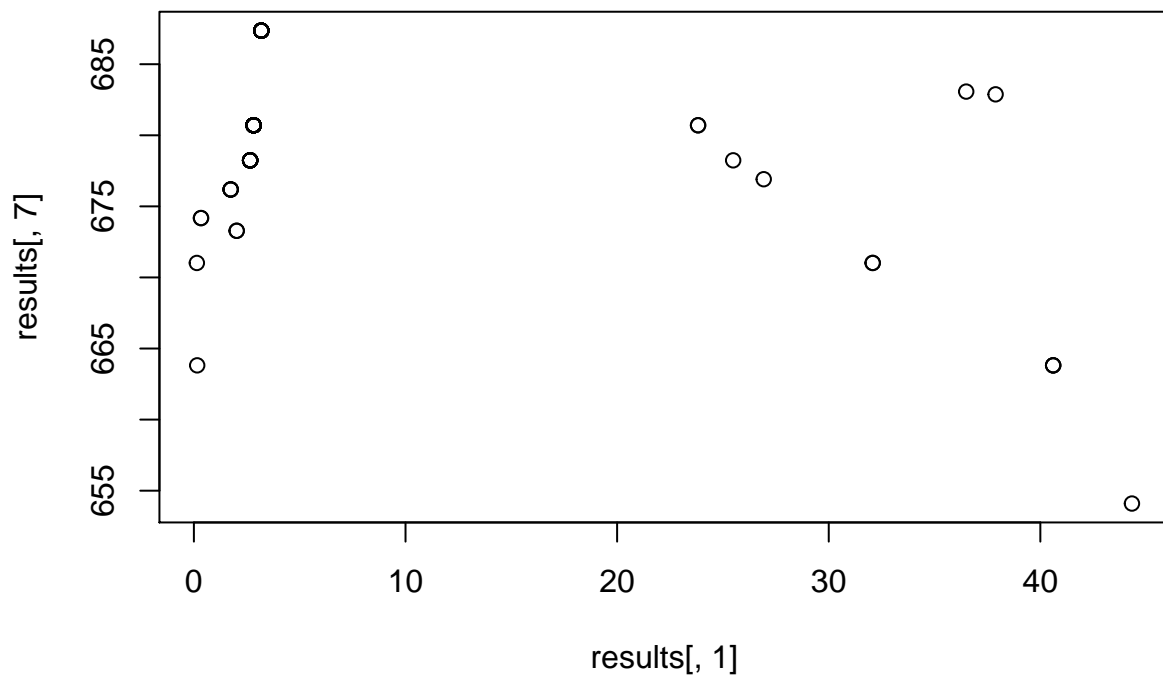


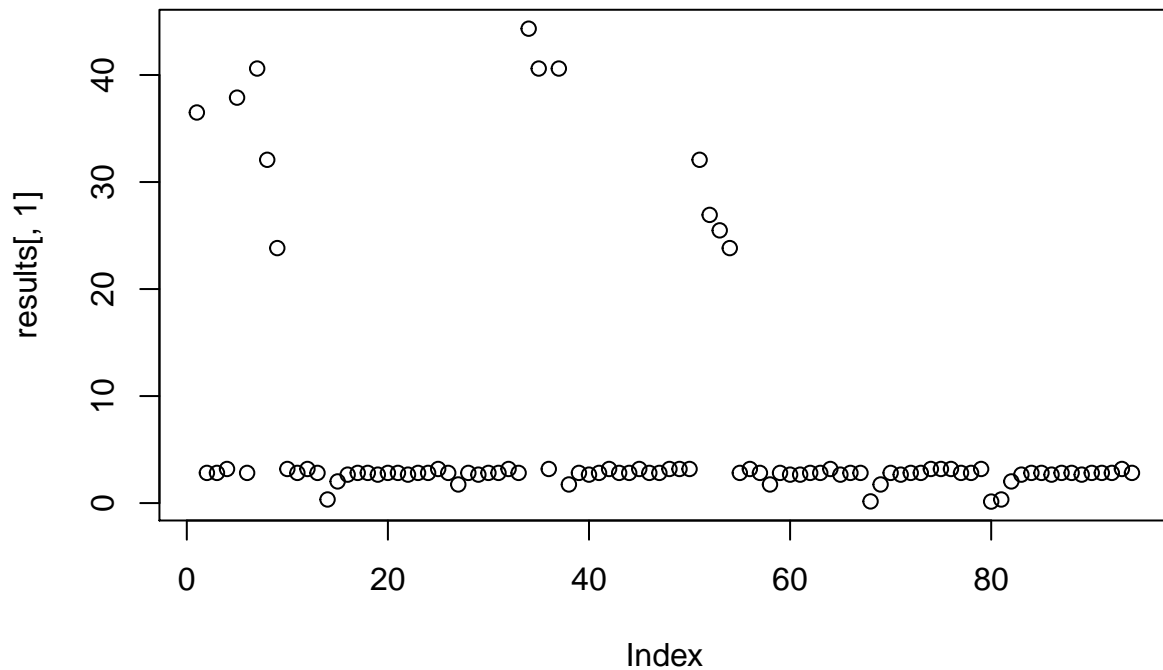


```
## 1 factor completed in 0.00405 minutes. Estimated time of completion: 2020-04-29 13:47:51 [1]
## [6] 2.665570e-11 5.000000e+01 2.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 60.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

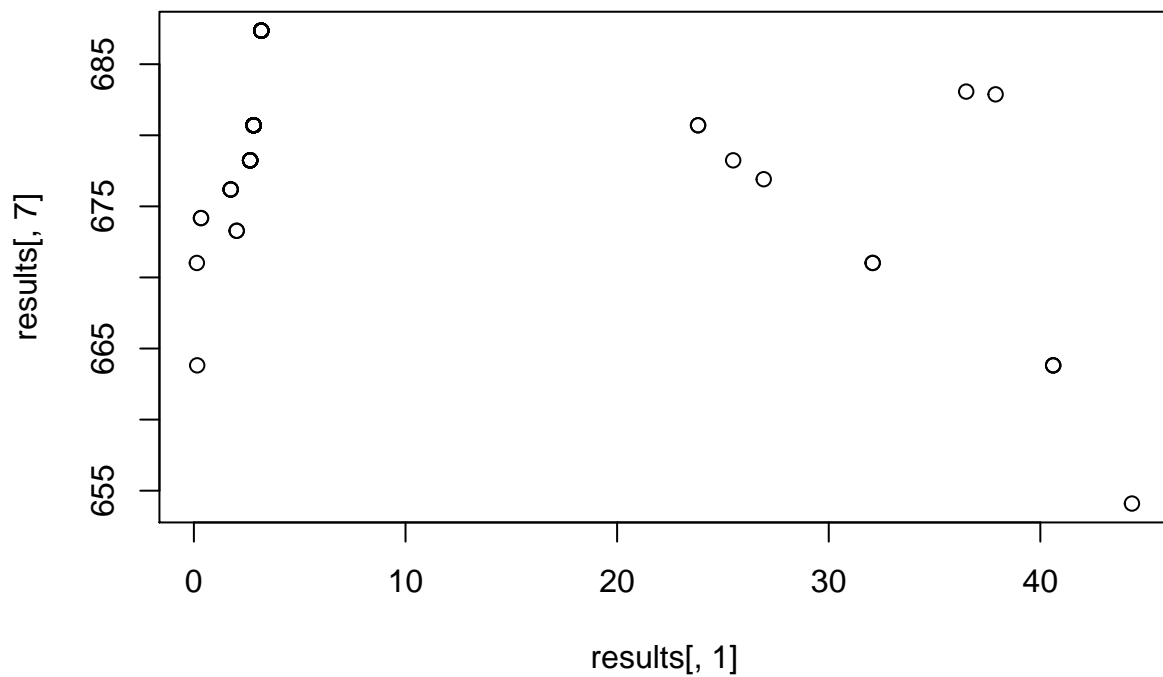


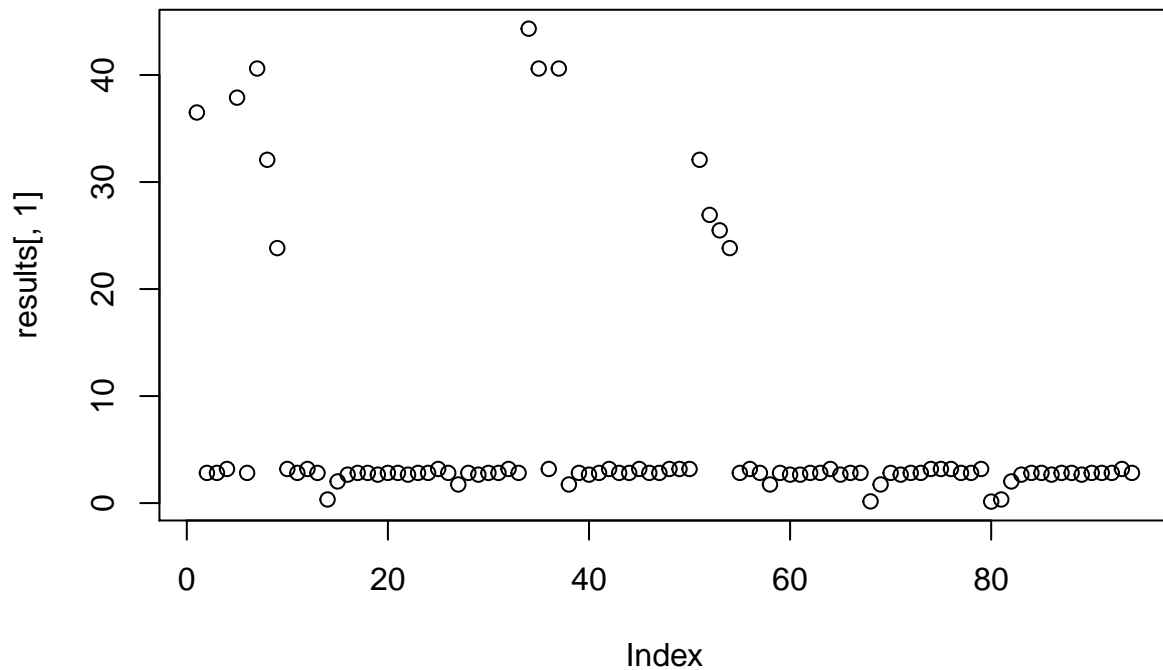


```
## 1 factor completed in 0.00382 minutes. Estimated time of completion: 2020-04-29 13:47:53 [1]
## [6] 1.123937e-13 6.000000e+01 2.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 70.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

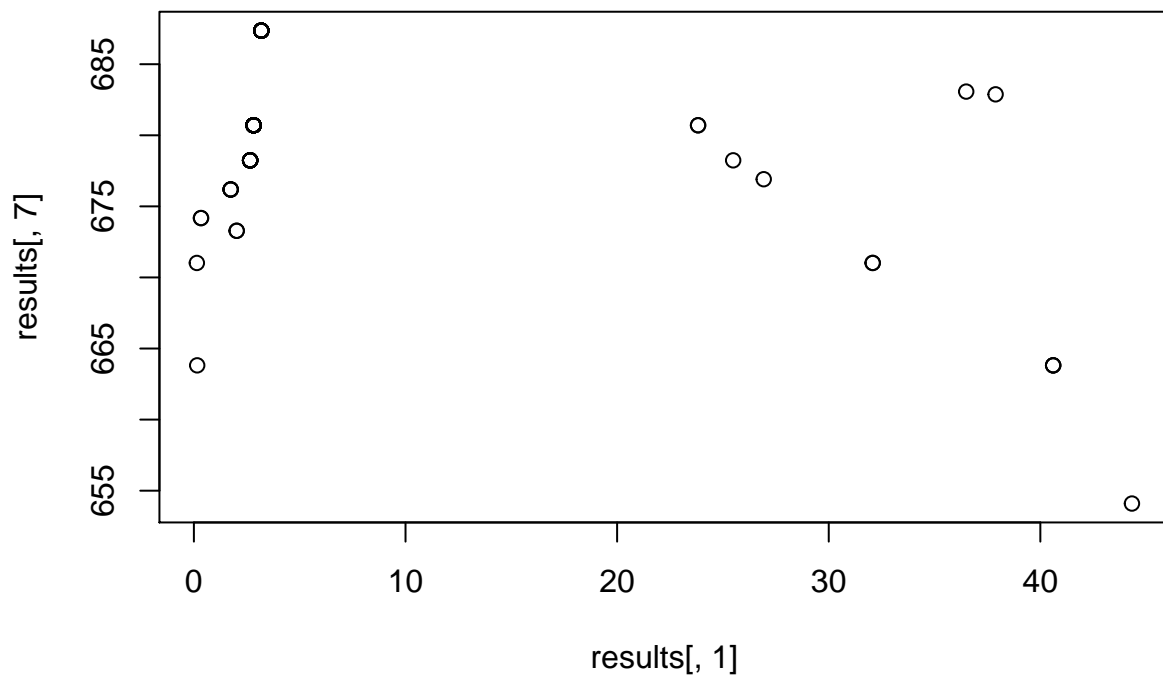


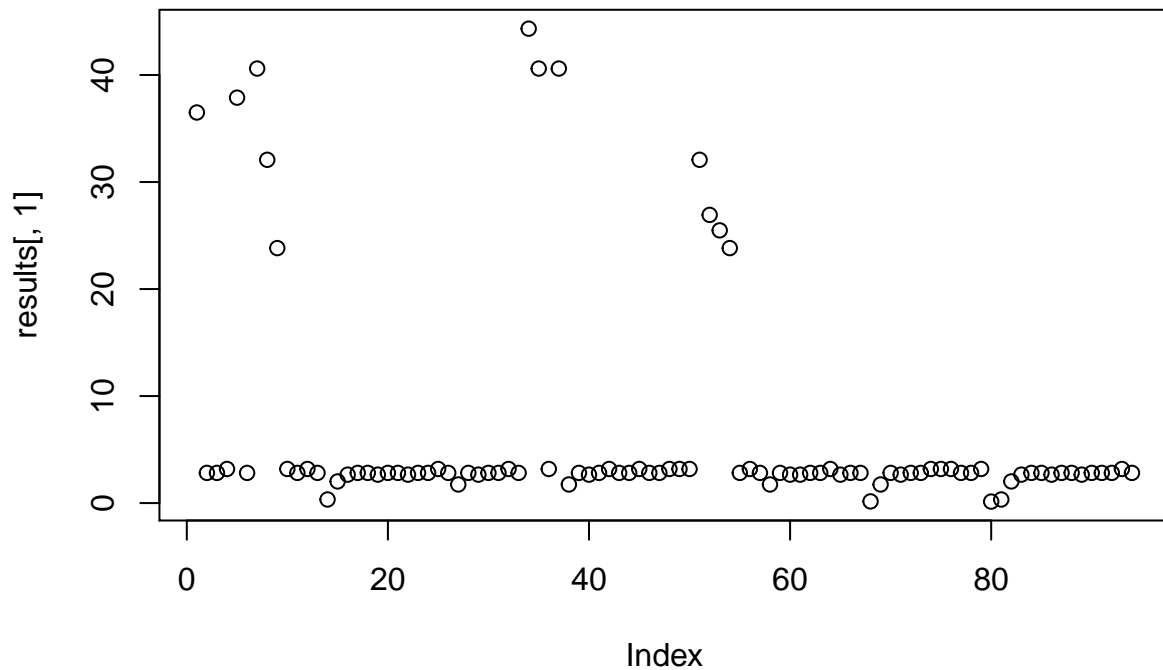


```
## 1 factor completed in 0.00393 minutes. Estimated time of completion: 2020-04-29 13:47:55 [1]
## [6] 4.739081e-16 7.000000e+01 2.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 80.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

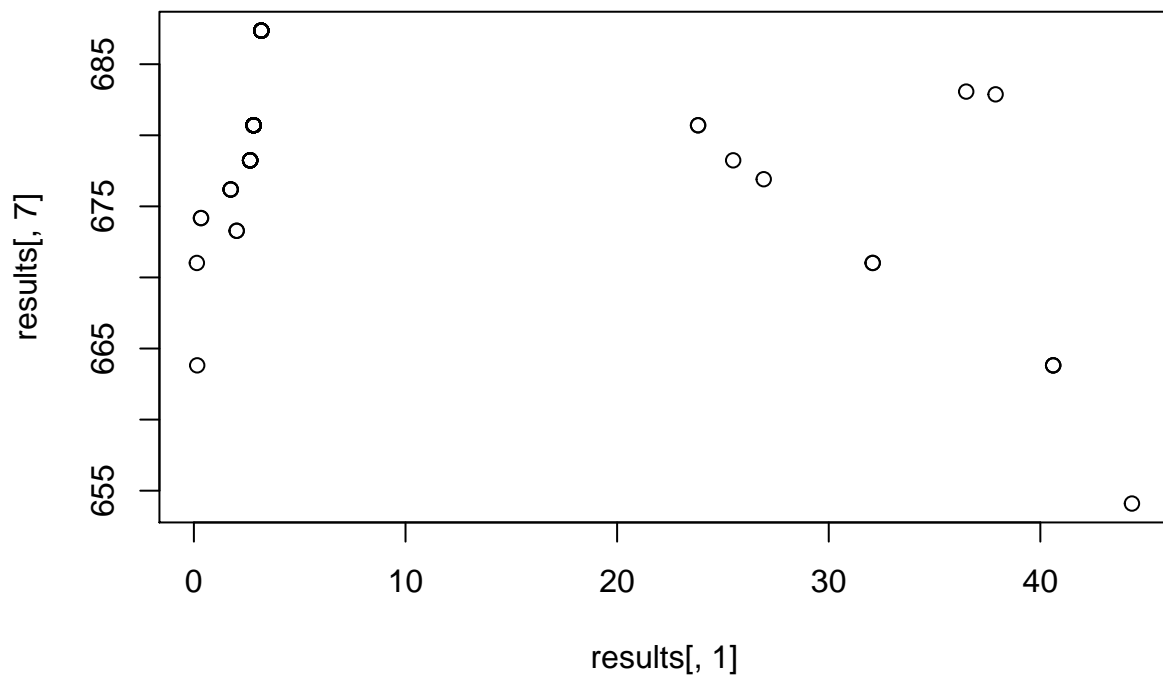


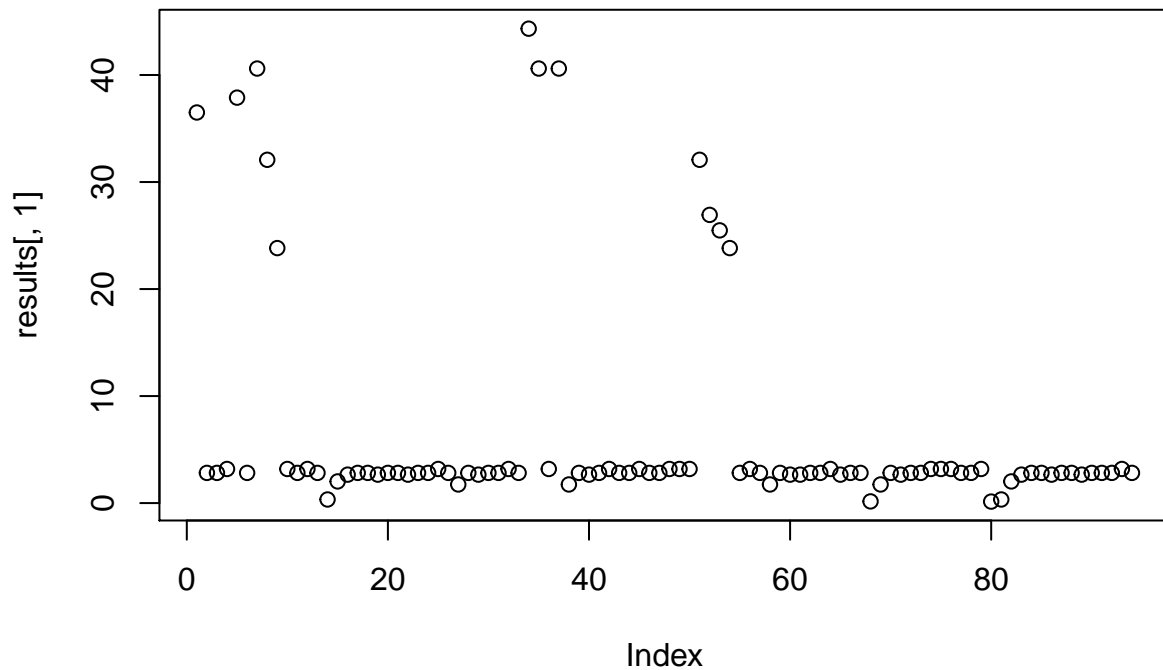


```
## 1 factor completed in 0.00391 minutes. Estimated time of completion: 2020-04-29 13:47:57 [1]
## [6] 1.998233e-18 8.000000e+01 2.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 90.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

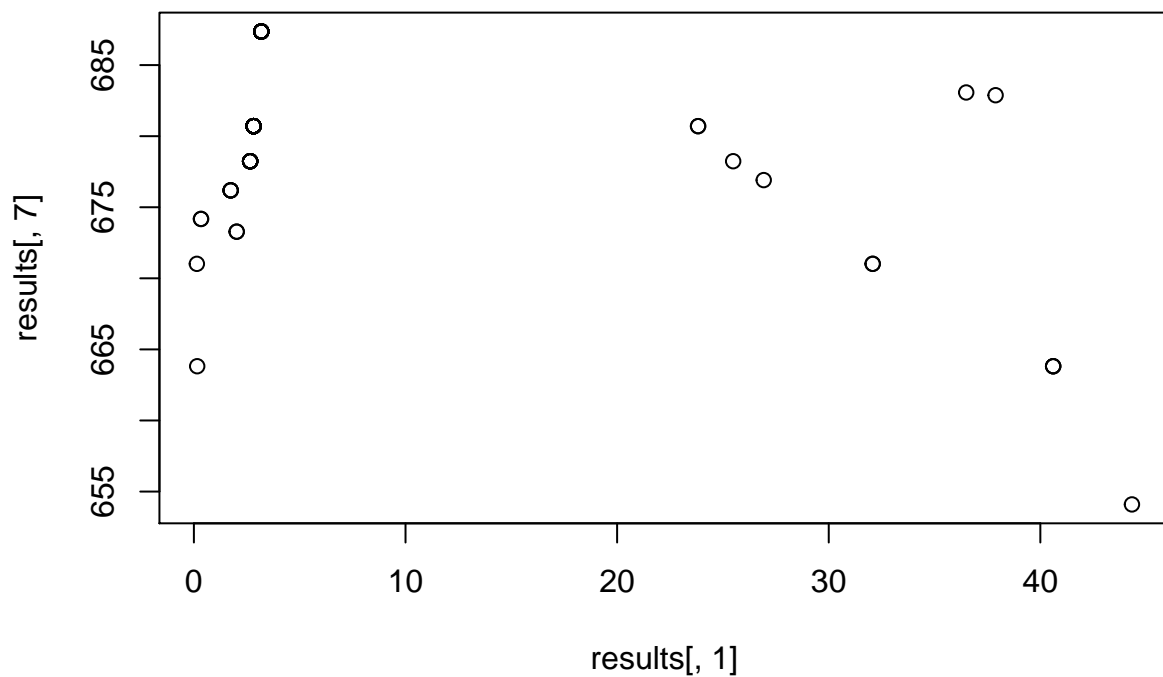


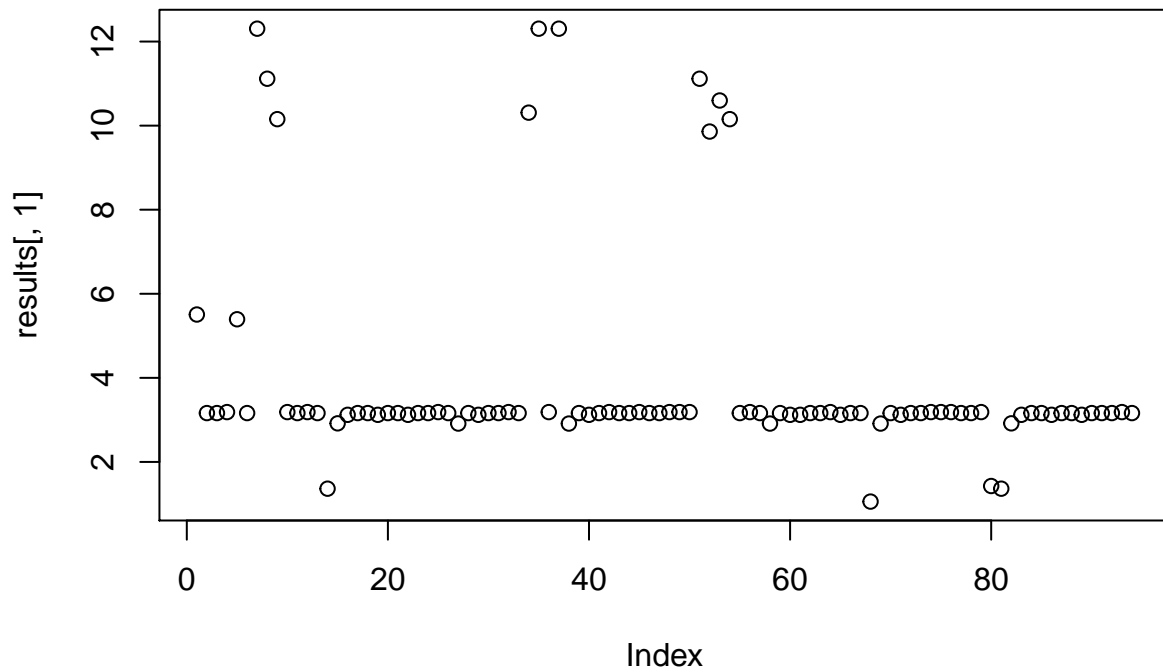


```
## 1 factor completed in 0.00608 minutes. Estimated time of completion: 2020-04-29 13:47:59 [1]
## [6] 8.425550e-21 9.000000e+01 2.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "X = 30.000001"
## [1] "j = 1e-06"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

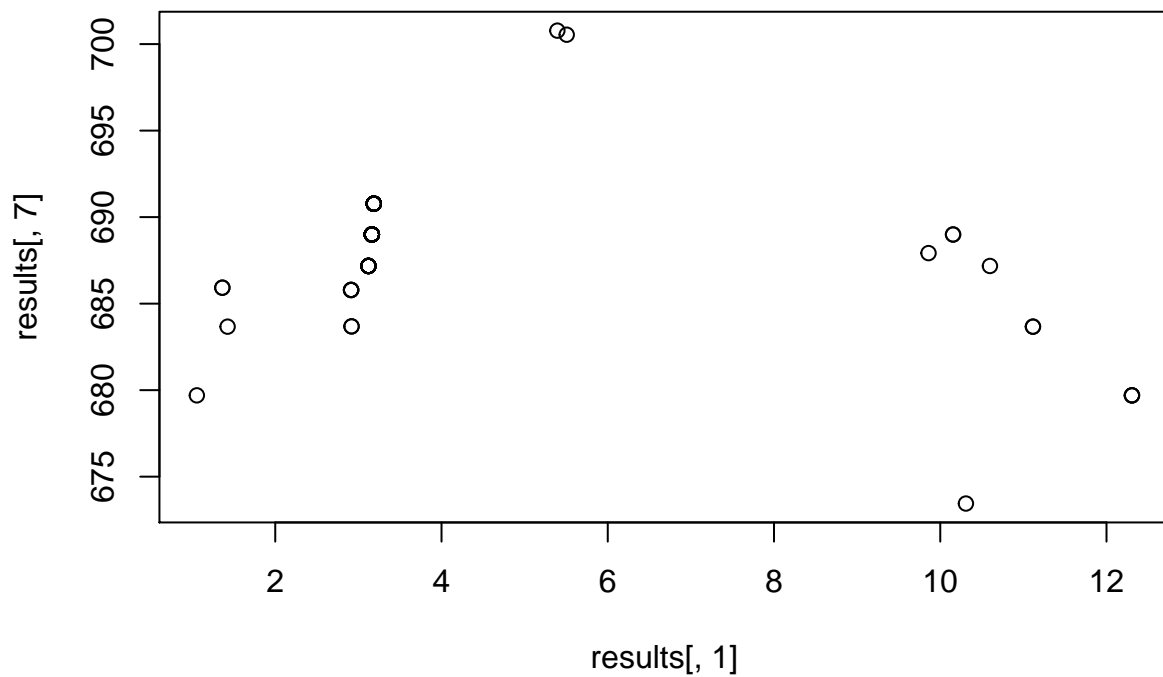


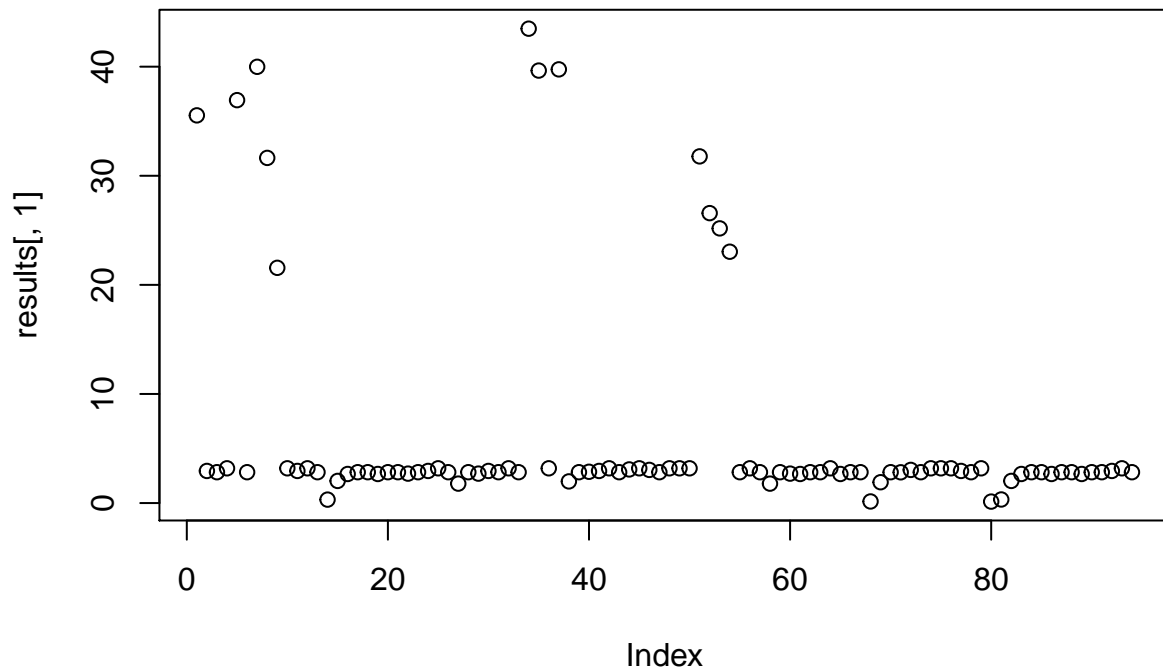


```
## 1 factor completed in 0.00803 minutes. Estimated time of completion: 2020-04-29 13:48:02 [1]
## [7] 0.0000010 30.0000010 7.8130568 679.6987543 10.3085190 673.4464585
## [13] 68.0000000 34.0000000
## [1] "j = 10.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

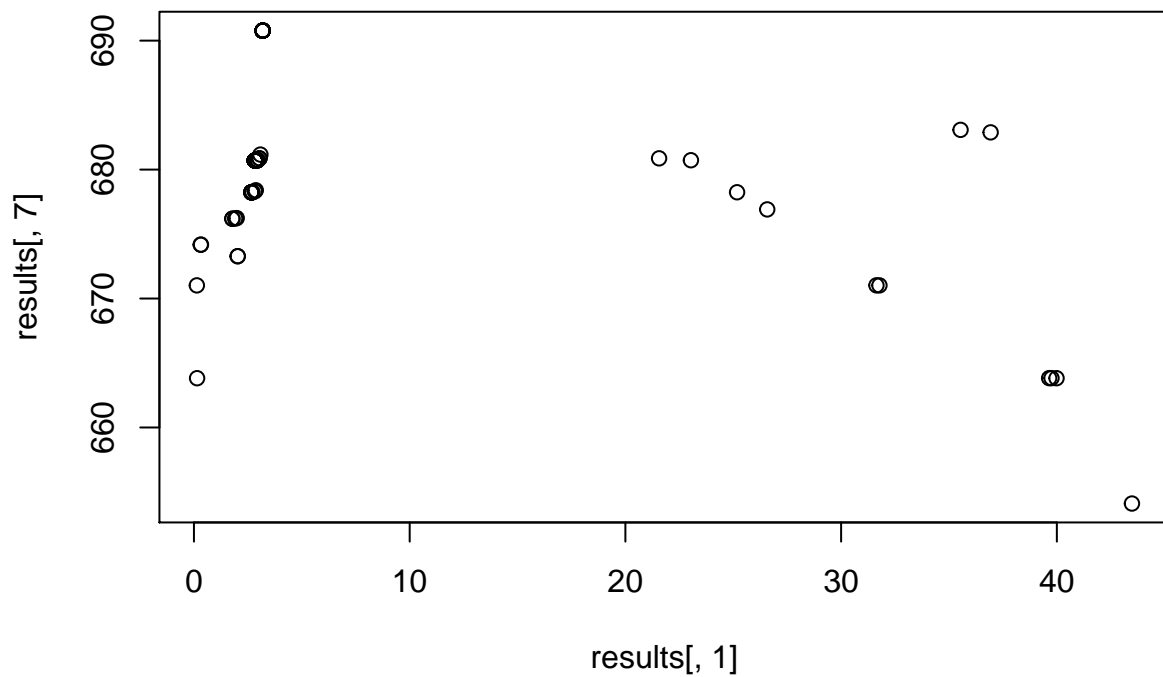


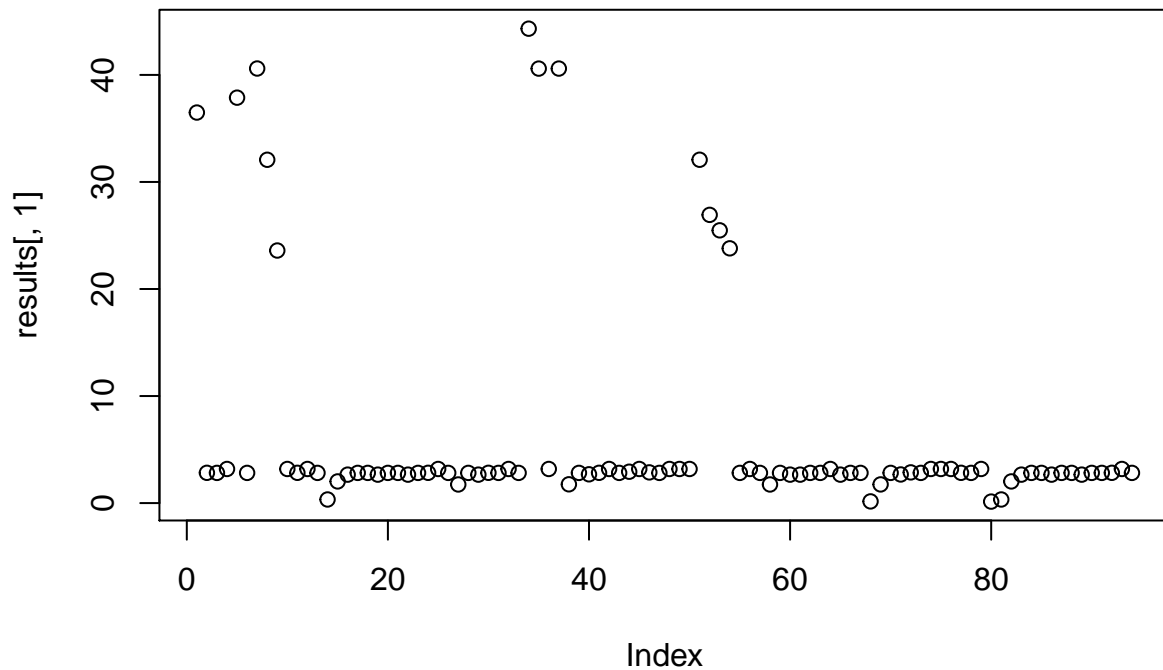


```
## 1 factor completed in 0.00422 minutes. Estimated time of completion: 2020-04-29 13:48:04 [1]
## [7] 10.0000010 30.0000010 7.8130568 671.0202185 43.4816961 654.1022189
## [13] 80.0000000 34.0000000
## [1] "j = 20.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

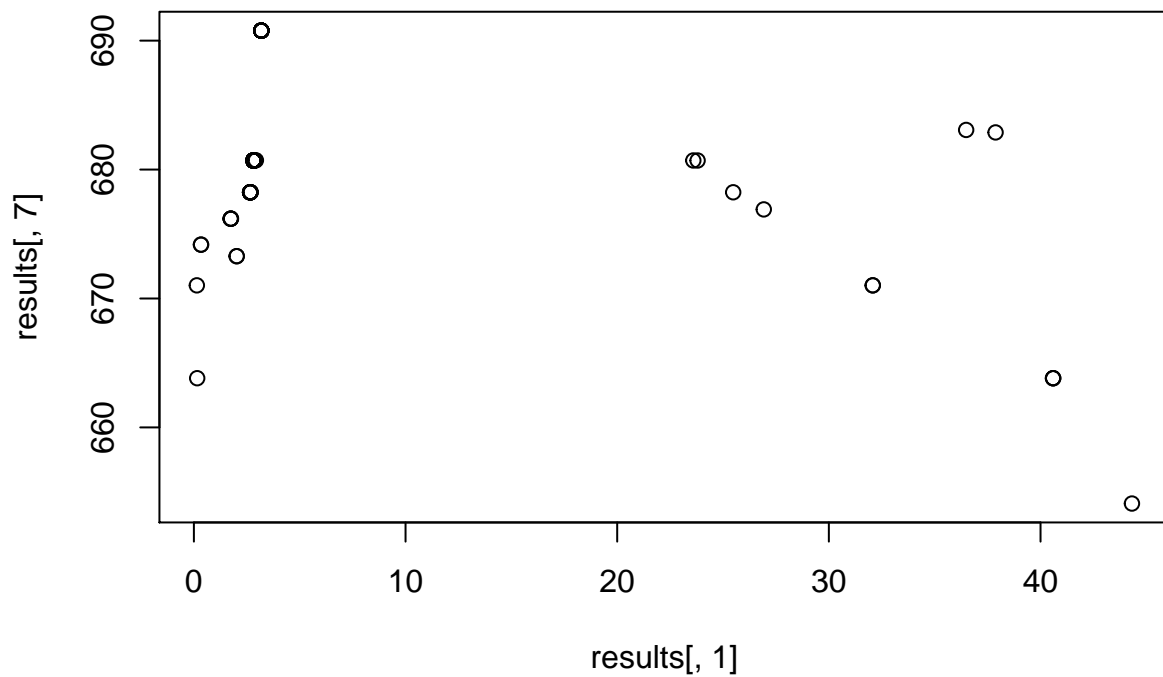


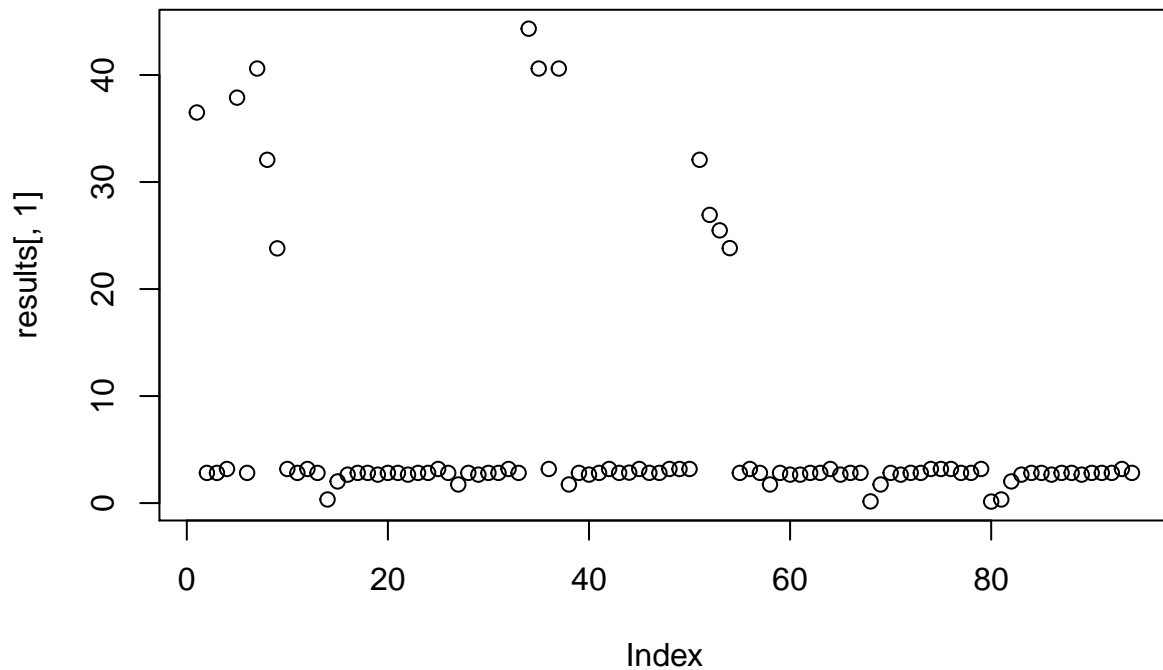


```
## 1 factor completed in 0.00394 minutes. Estimated time of completion: 2020-04-29 13:48:06 [1]
## [6] 5.333658e-04 2.000000e+01 3.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.431976e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 30.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

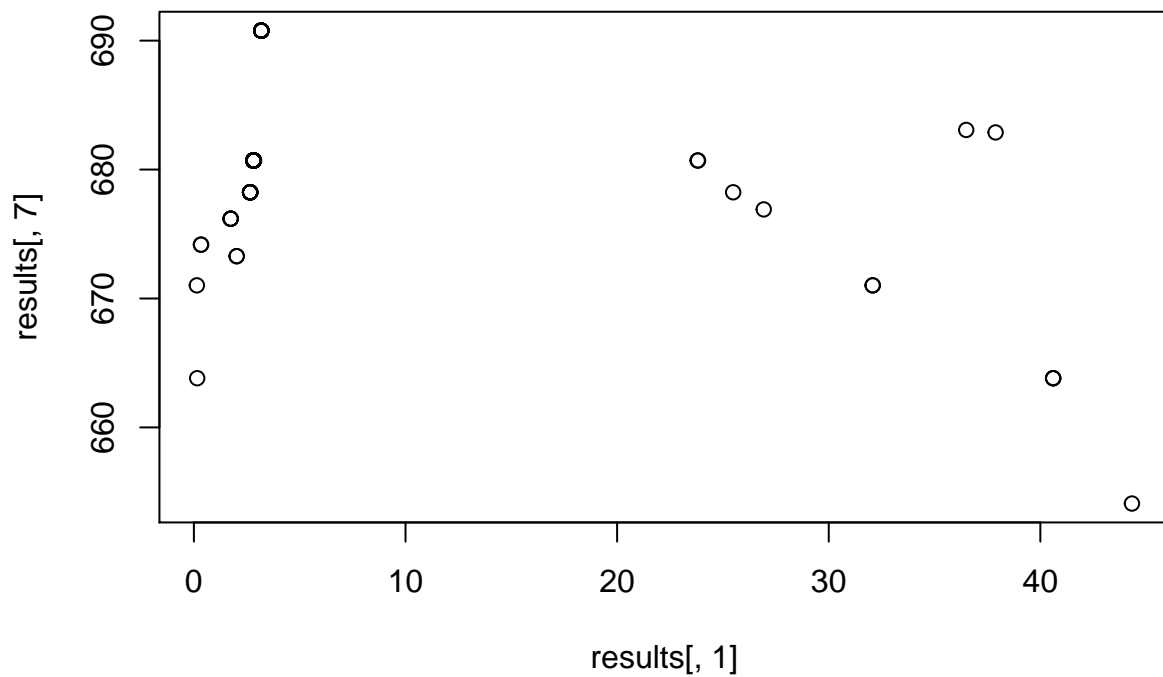


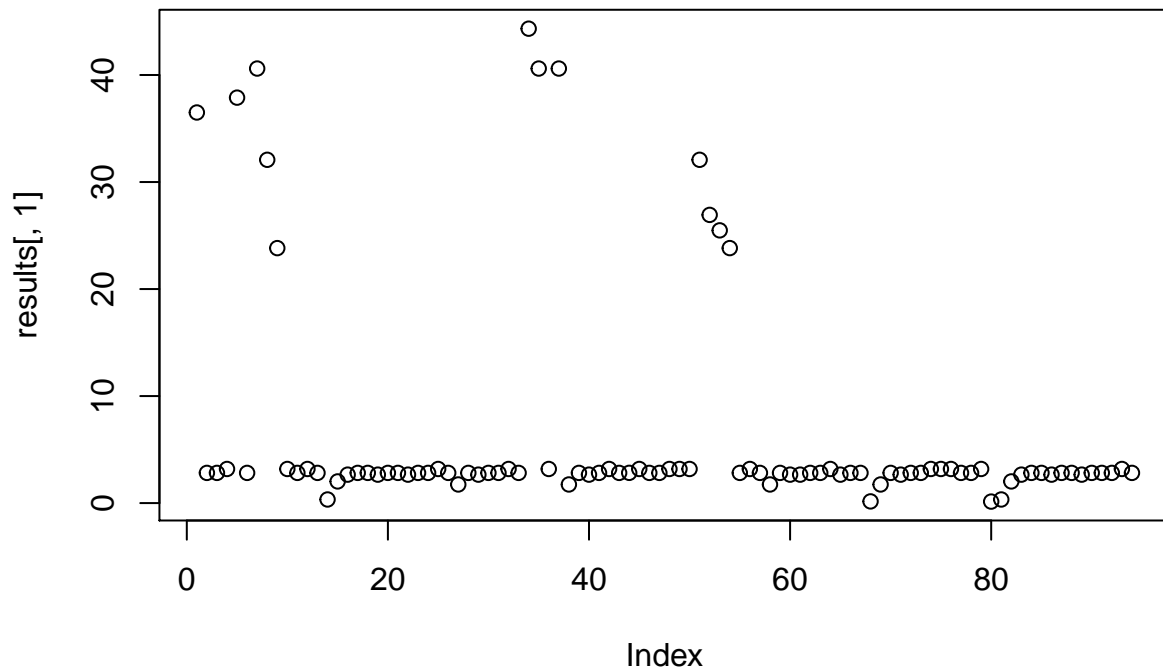


```
## 1 factor completed in 0.00395 minutes. Estimated time of completion: 2020-04-29 13:48:08 [1]
## [6] 2.248936e-06 3.000000e+01 3.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432870e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 40.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

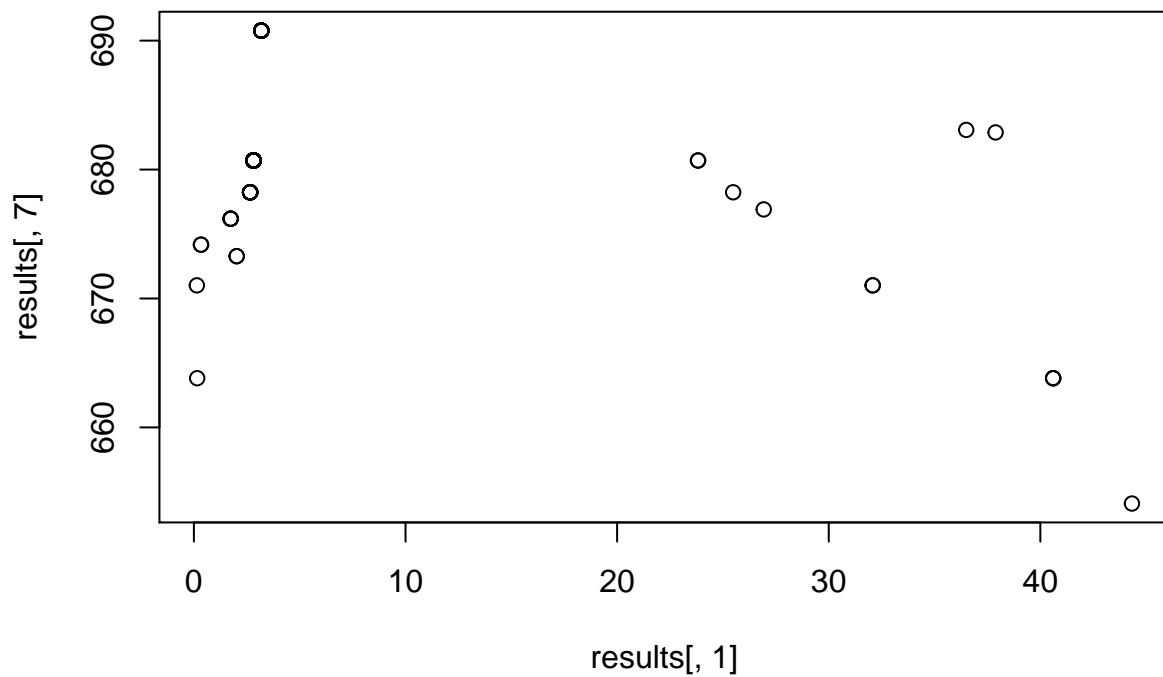


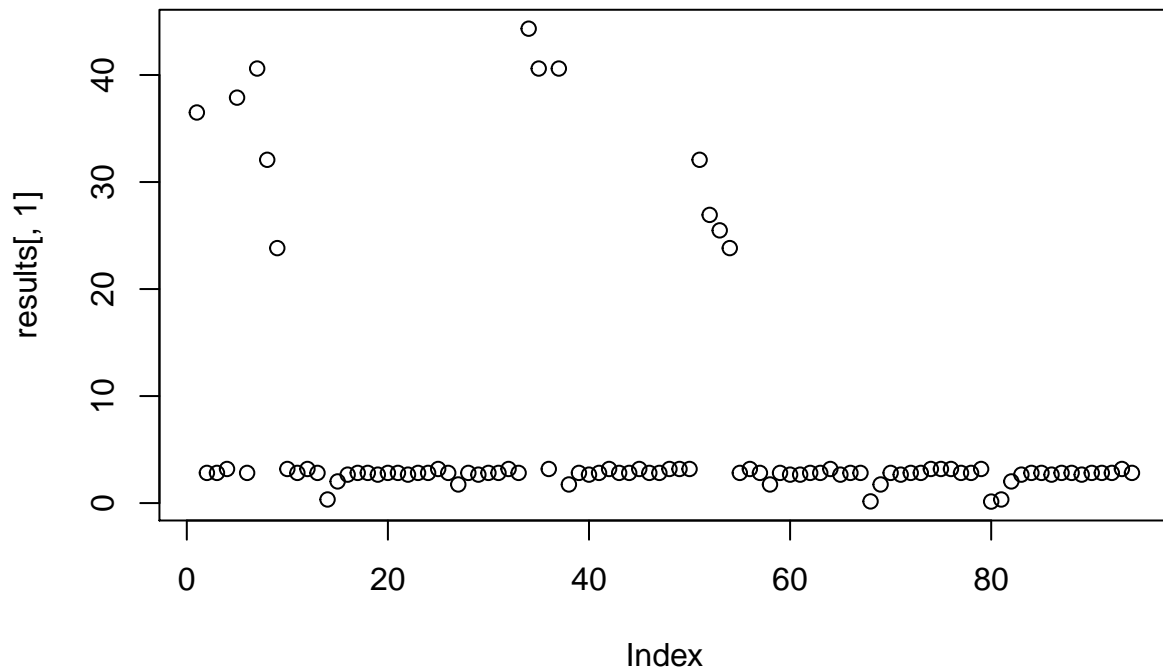


```
## 1 factor completed in 0.00412 minutes. Estimated time of completion: 2020-04-29 13:48:10 [1]
## [6] 9.482639e-09 4.000000e+01 3.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 50.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

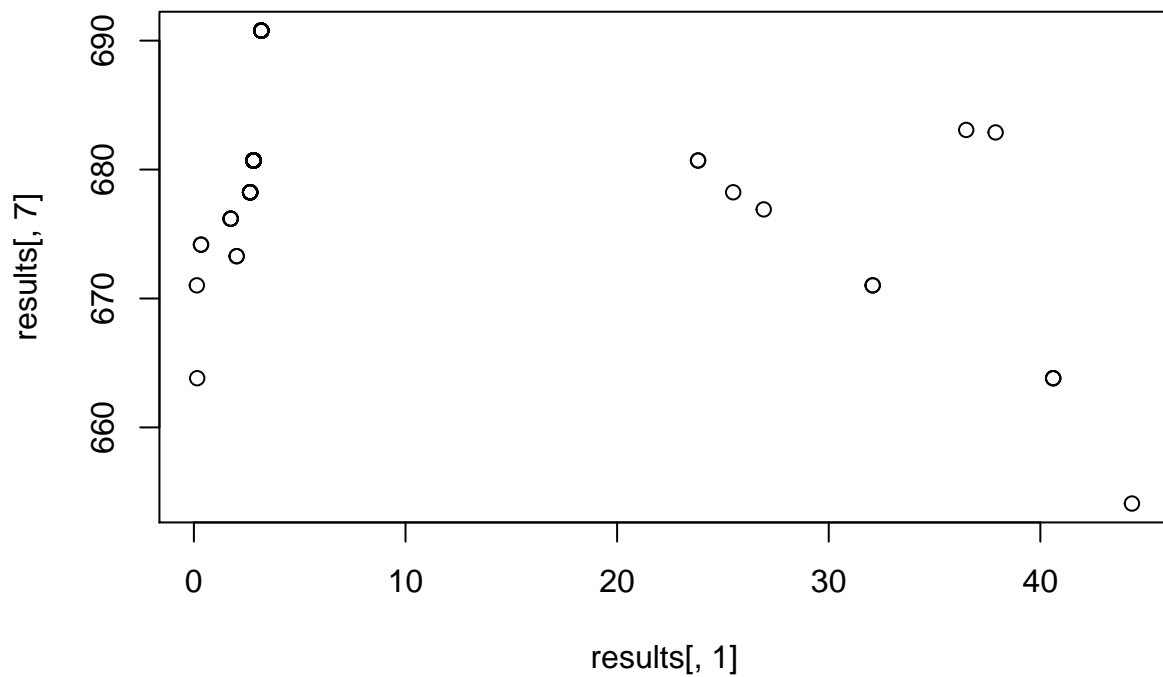


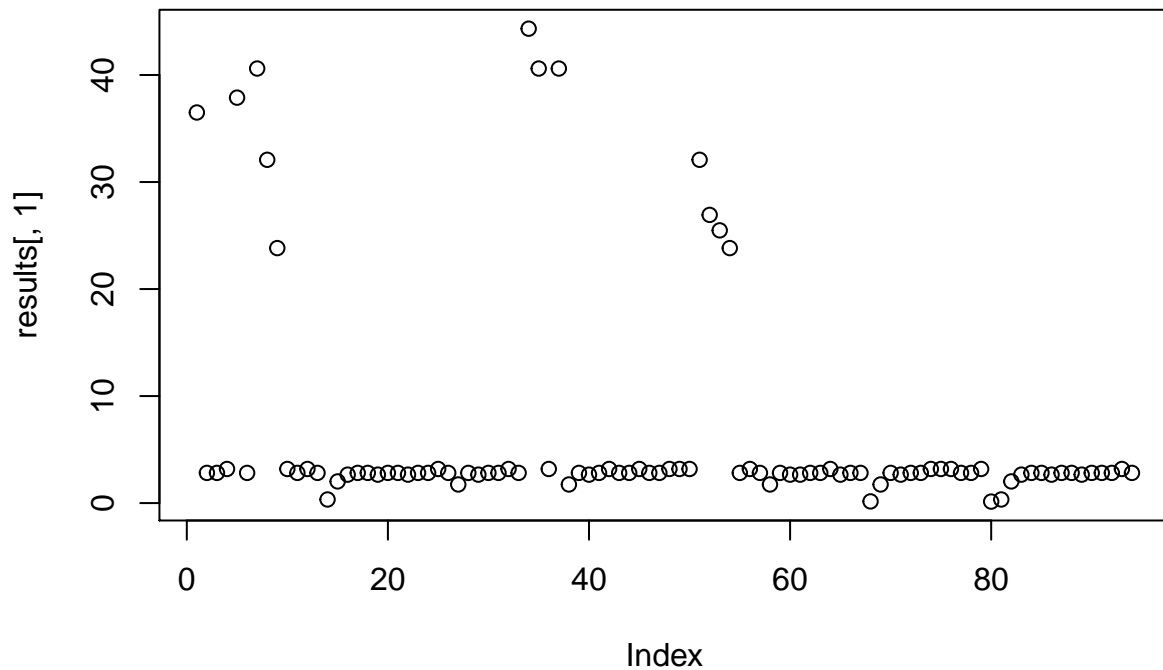


```
## 1 factor completed in 0.00397 minutes. Estimated time of completion: 2020-04-29 13:48:12 [1]
## [6] 3.998354e-11 5.000000e+01 3.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 60.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

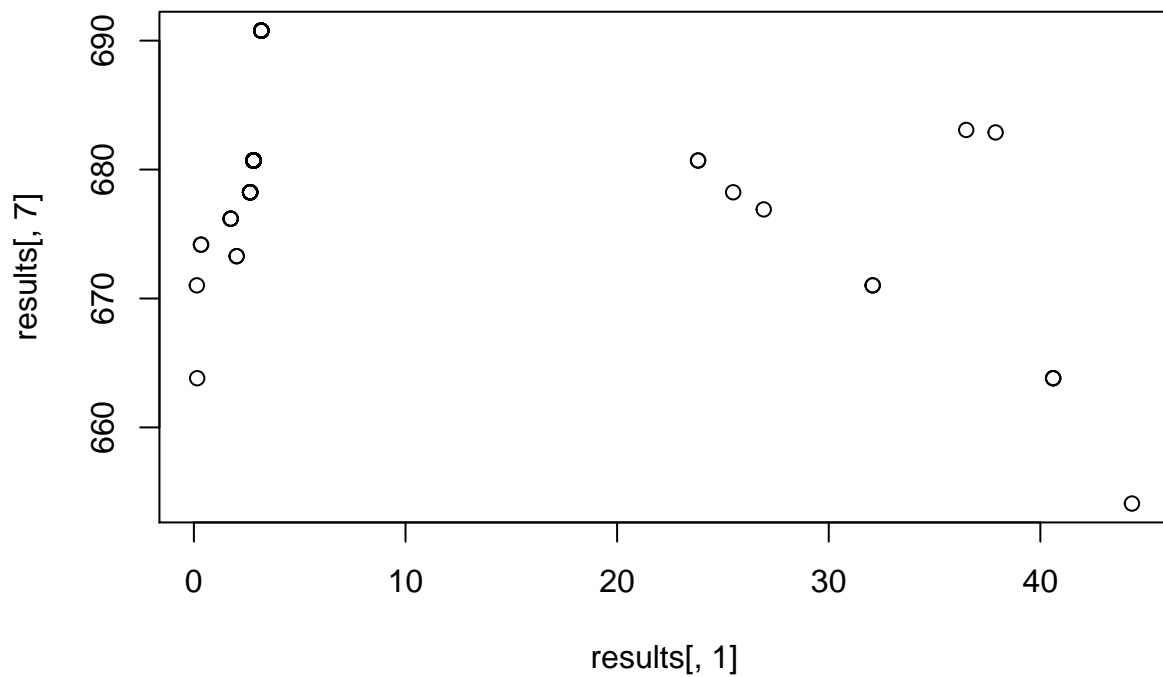


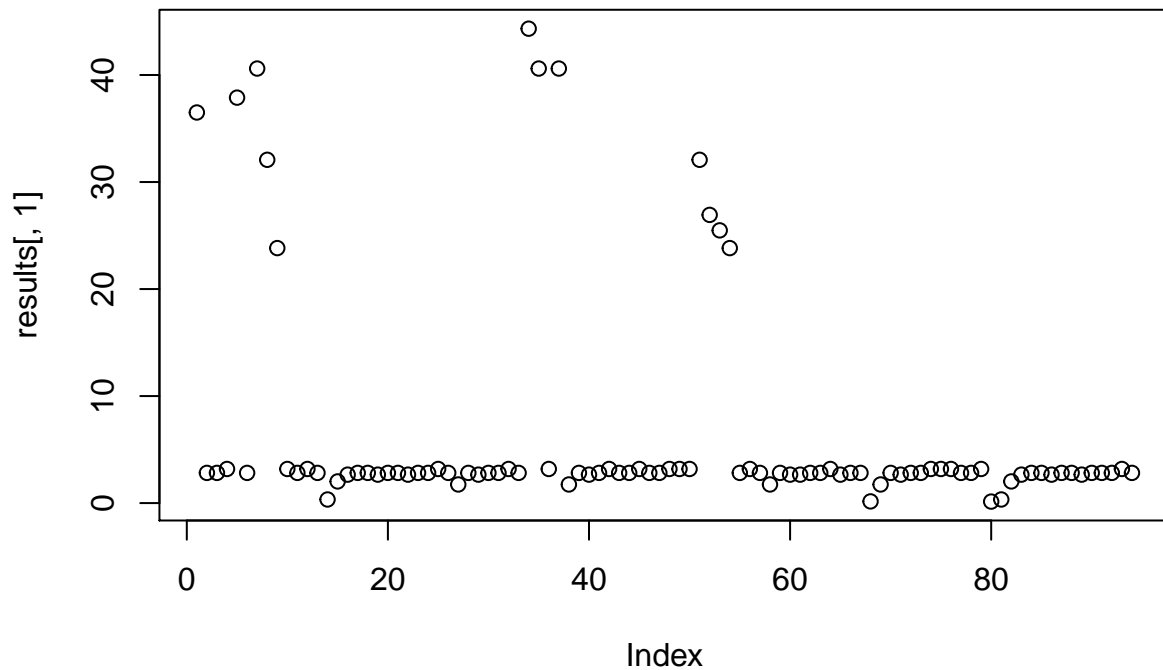


```
## 1 factor completed in 0.0043 minutes. Estimated time of completion: 2020-04-29 13:48:14 [1]
## [6] 1.685906e-13 6.000000e+01 3.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 70.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

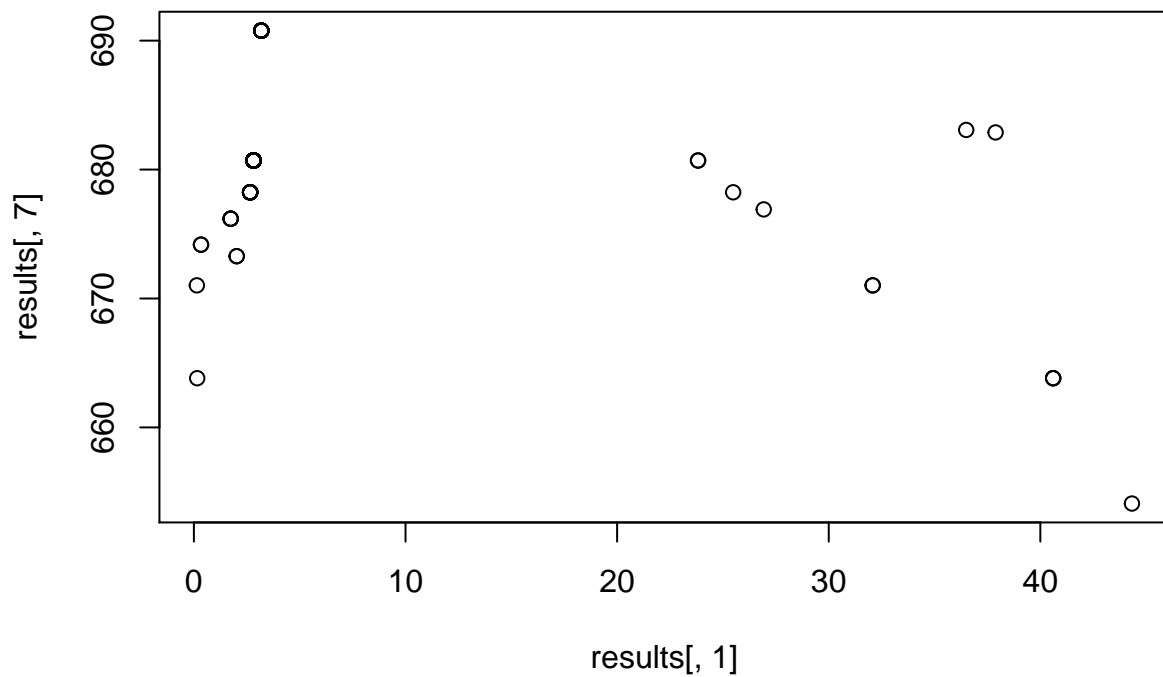


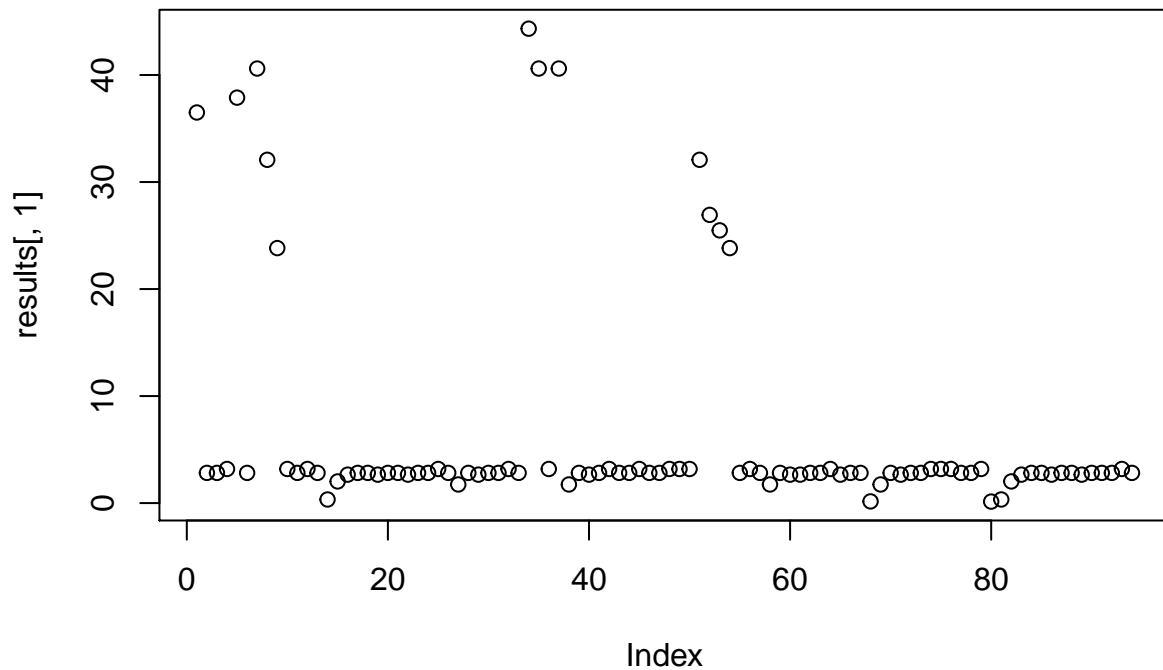


```
## 1 factor completed in 0.00412 minutes. Estimated time of completion: 2020-04-29 13:48:16 [1]
## [6] 7.108622e-16 7.000000e+01 3.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 80.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

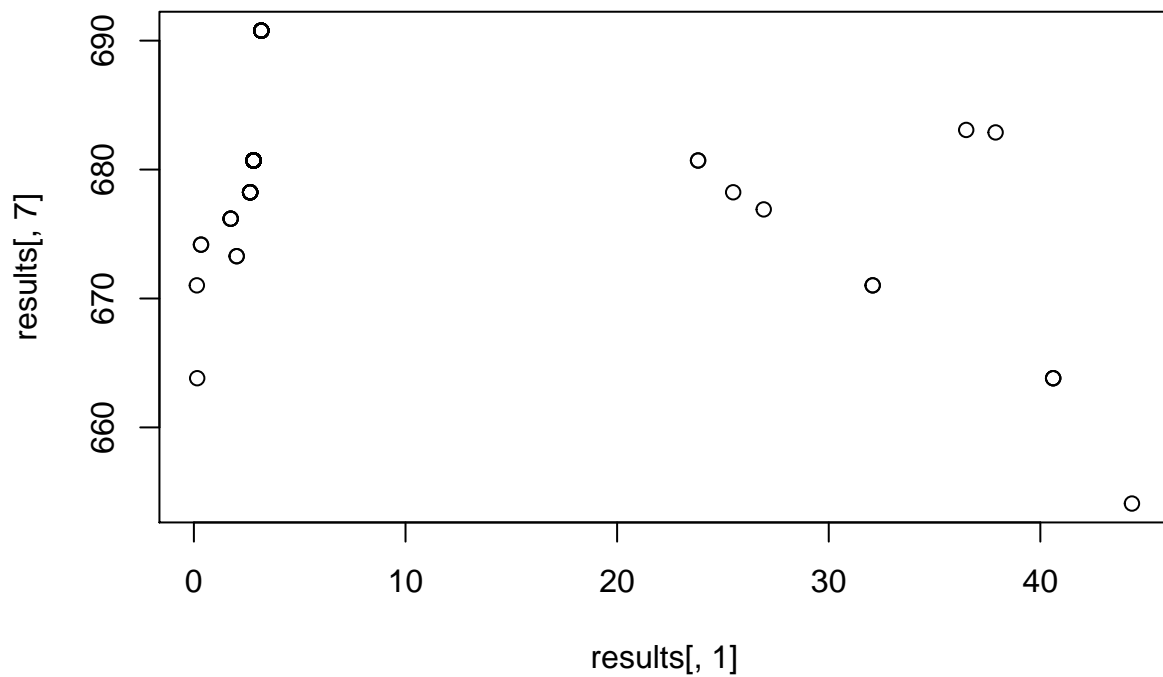


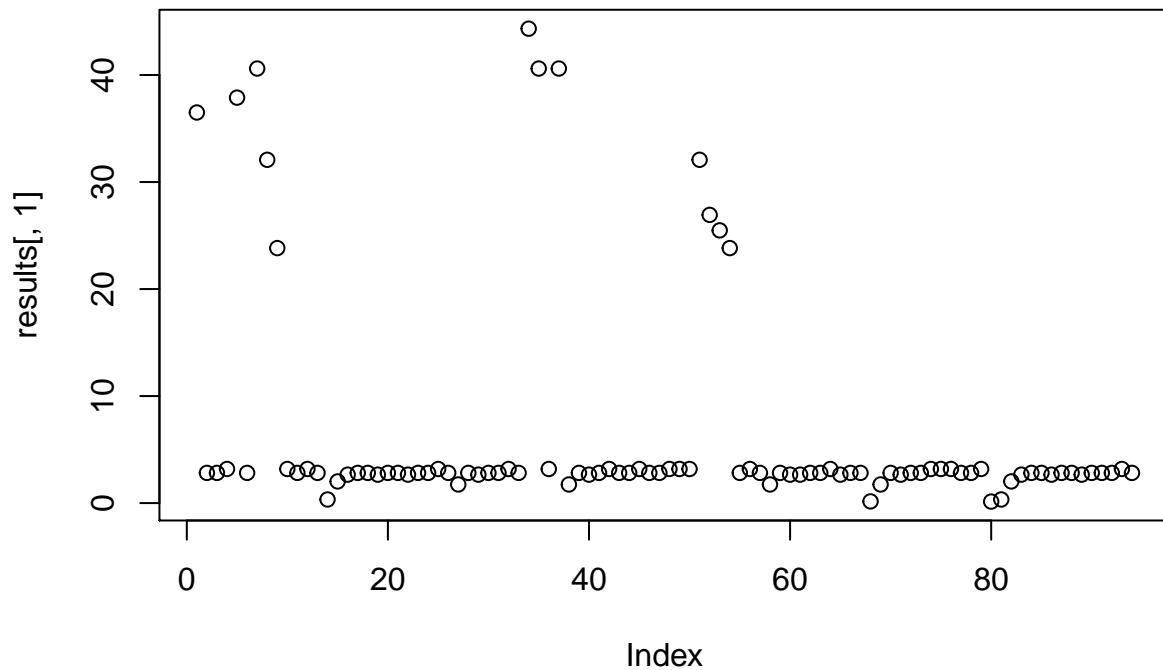


```
## 1 factor completed in 0.00396 minutes. Estimated time of completion: 2020-04-29 13:48:18 [1]
## [6] 2.997350e-18 8.000000e+01 3.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 90.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

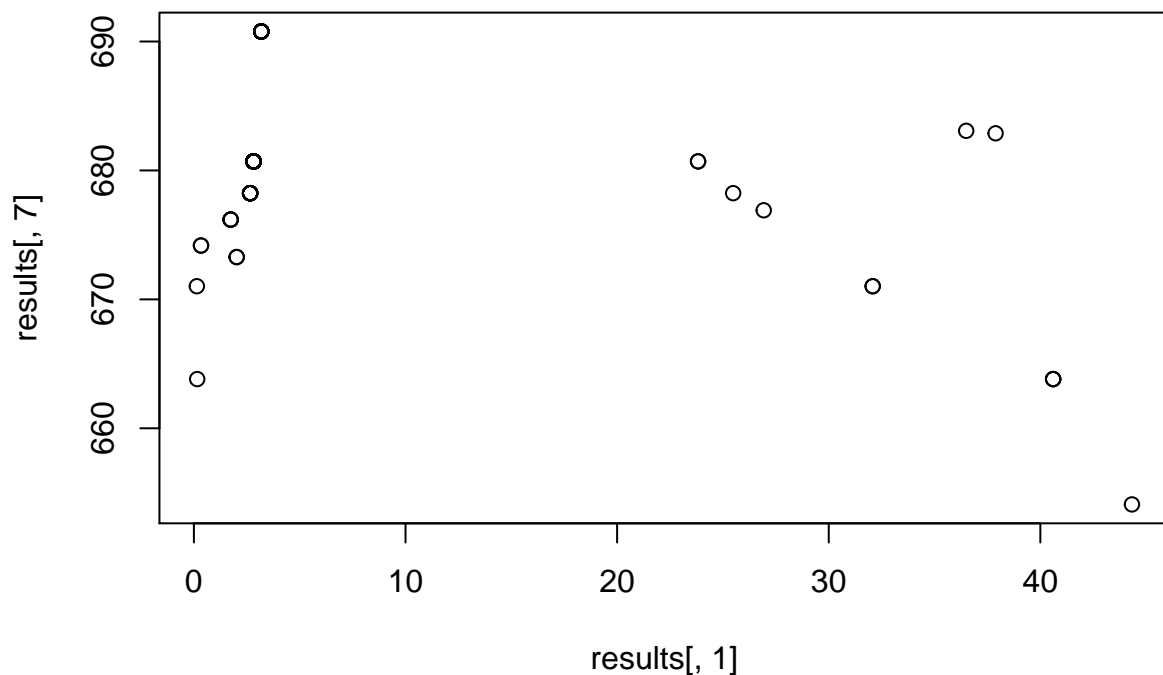


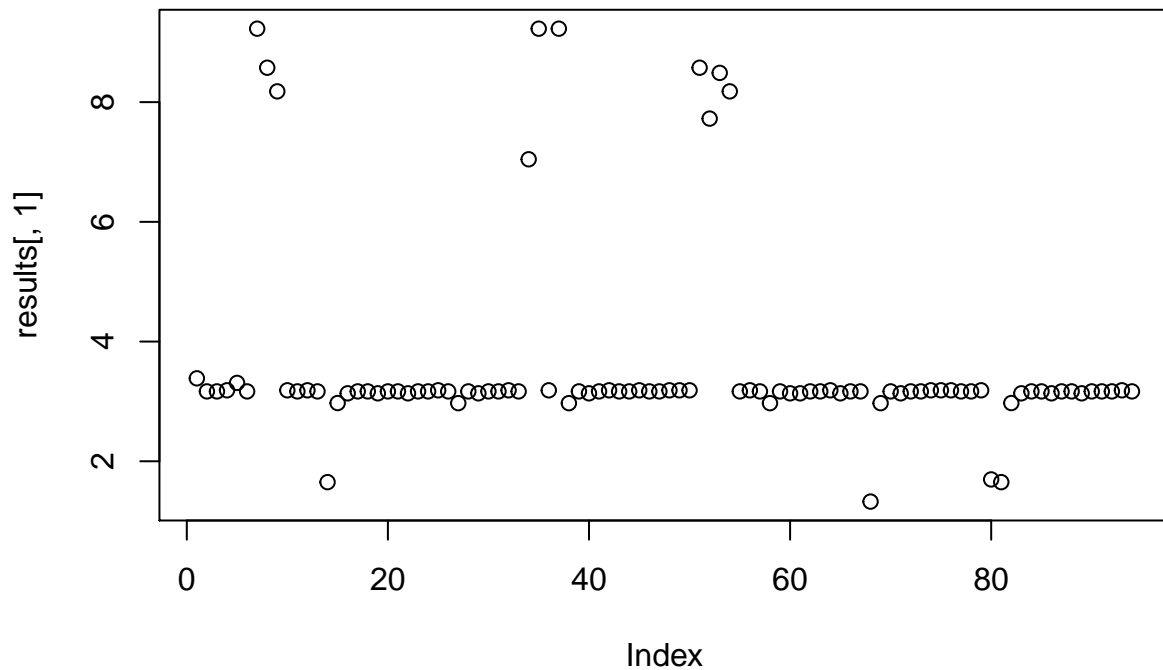


```
## 1 factor completed in 0.00405 minutes. Estimated time of completion: 2020-04-29 13:48:20 [1]
## [6] 1.263832e-20 9.000000e+01 3.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "X = 40.000001"
## [1] "j = 1e-06"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

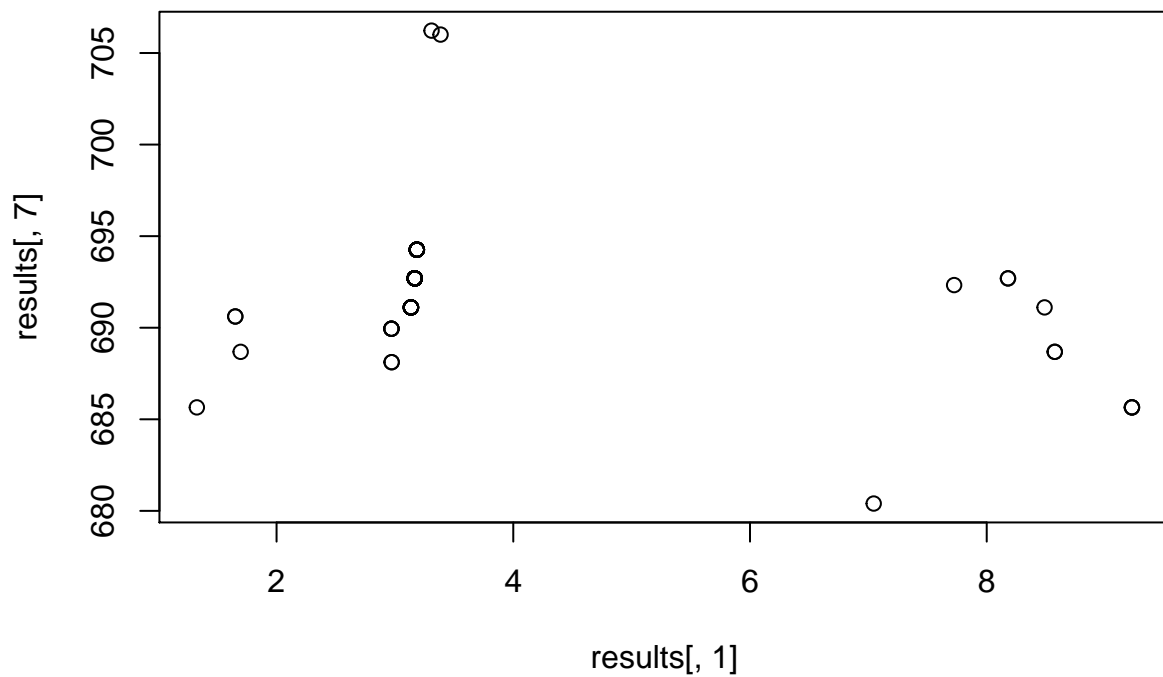


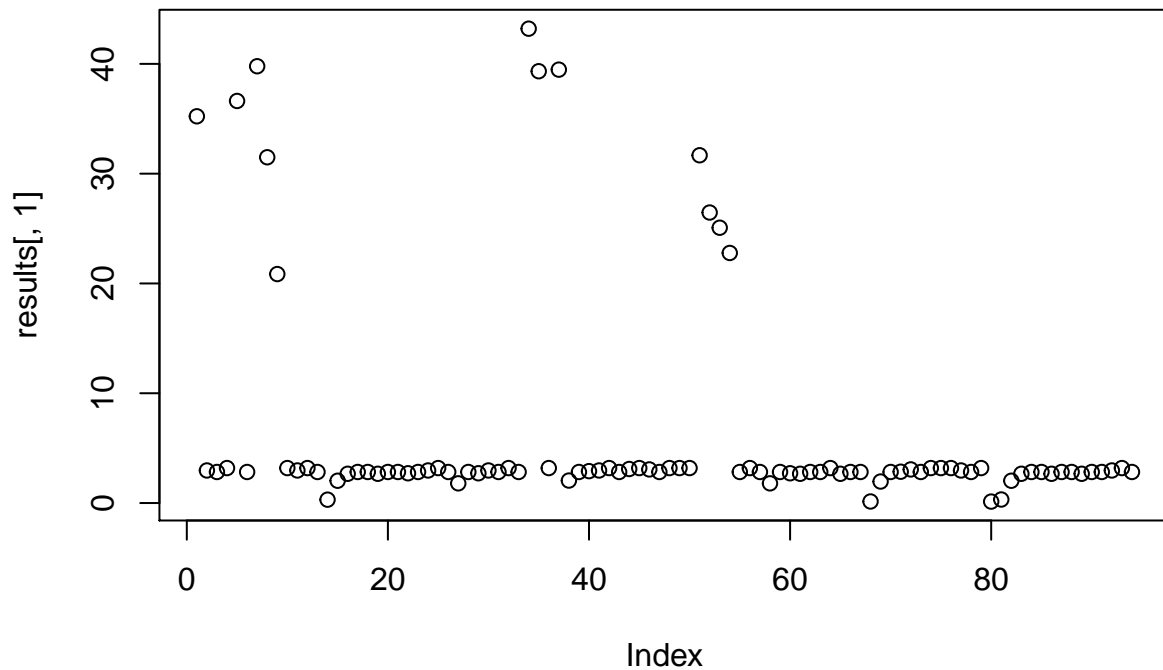


```
## 1 factor completed in 0.00405 minutes. Estimated time of completion: 2020-04-29 13:48:23 [1]
## [7] 0.0000010 40.0000010 7.8130568 685.6510533 7.0452648 680.4005931
## [13] 68.0000000 34.0000000
## [1] "j = 10.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

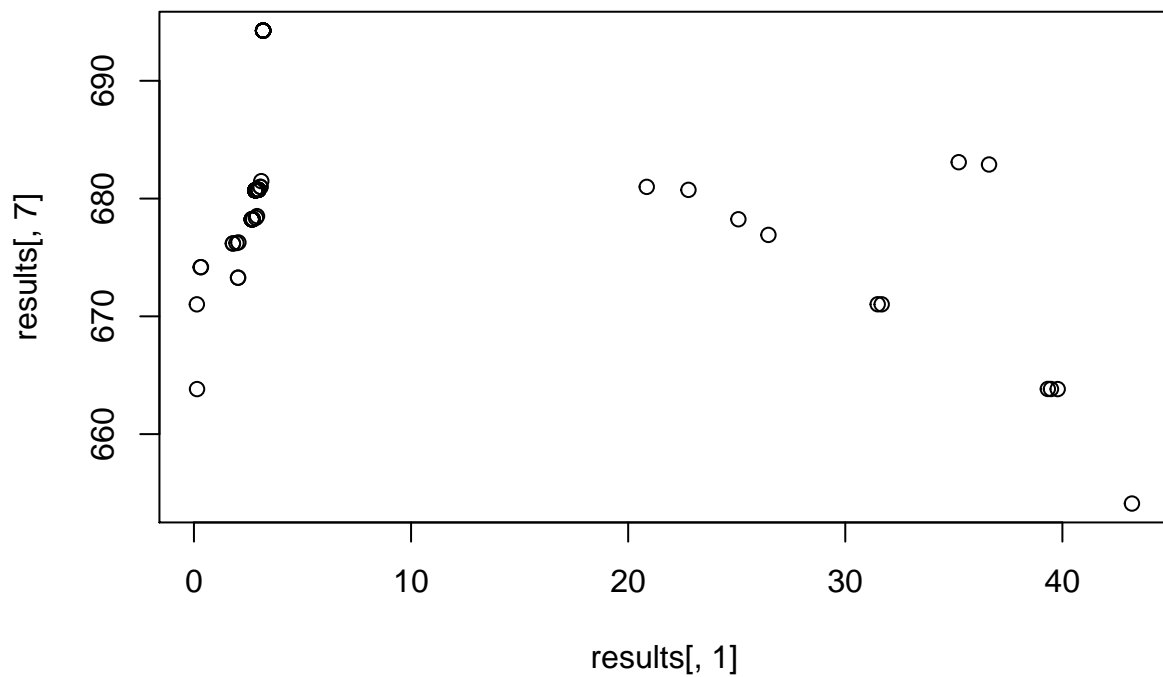


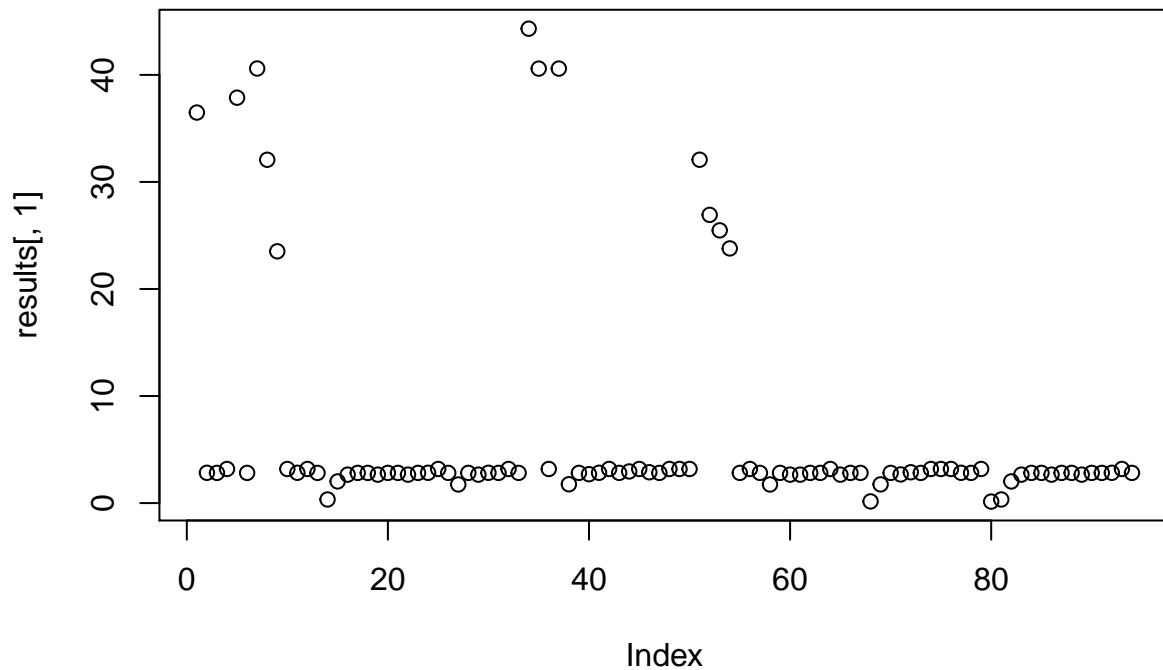


```
## 1 factor completed in 0.00429 minutes. Estimated time of completion: 2020-04-29 13:48:25 [1]
## [7] 10.0000010 40.0000010 7.8130568 671.0207072 43.2041485 654.1072410
## [13] 80.0000000 34.0000000
## [1] "j = 20.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

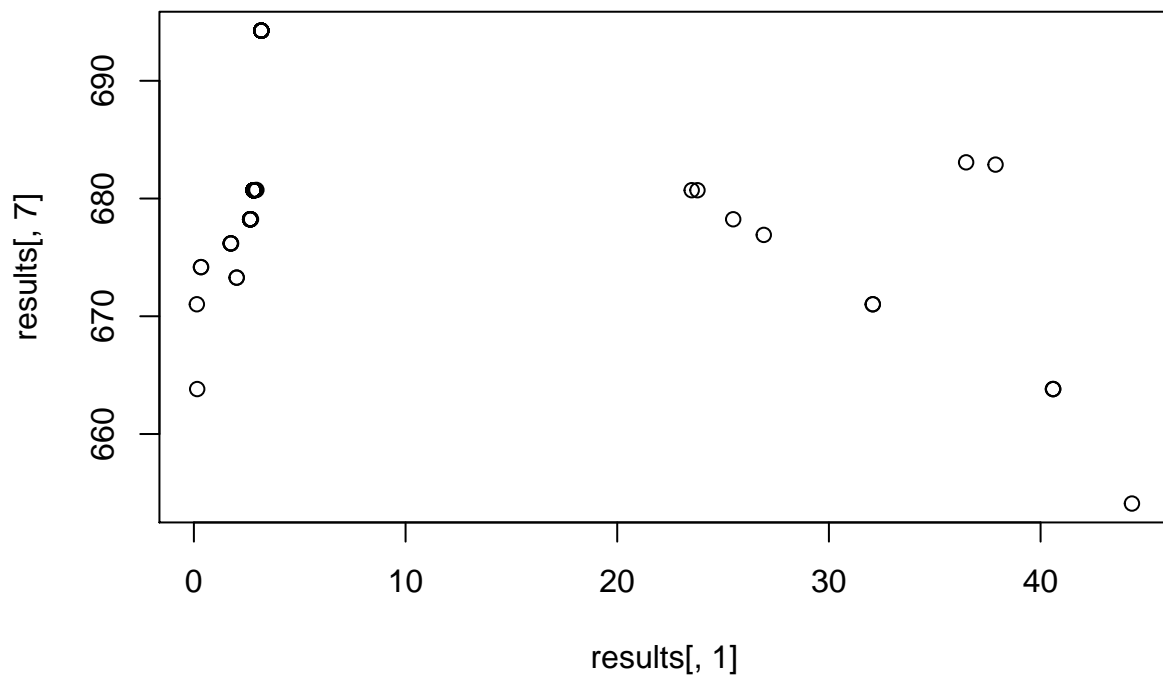


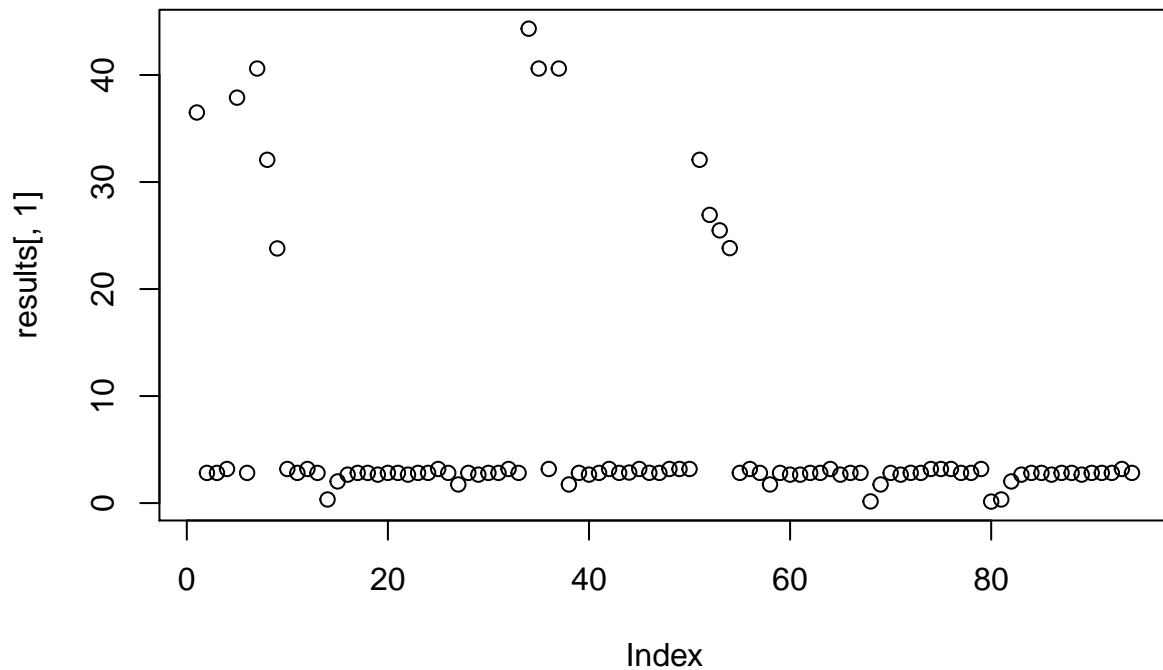


```
## 1 factor completed in 0.0042 minutes. Estimated time of completion: 2020-04-29 13:48:27 [1]
## [6] 7.111543e-04 2.000000e+01 4.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.431675e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 30.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

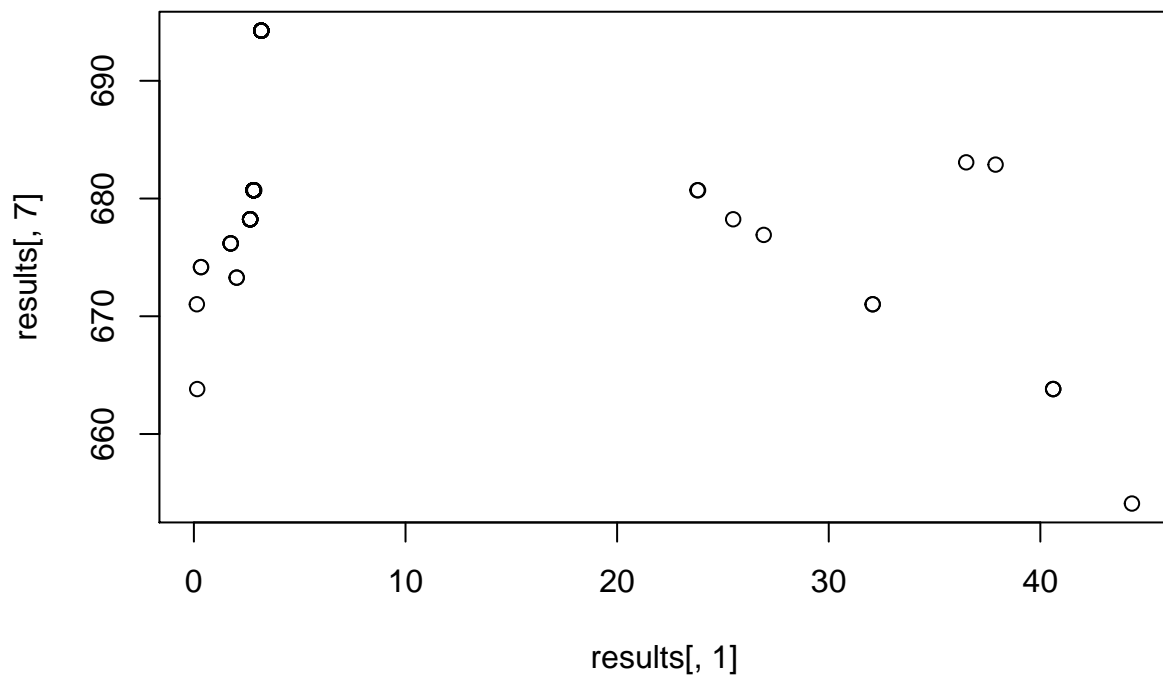


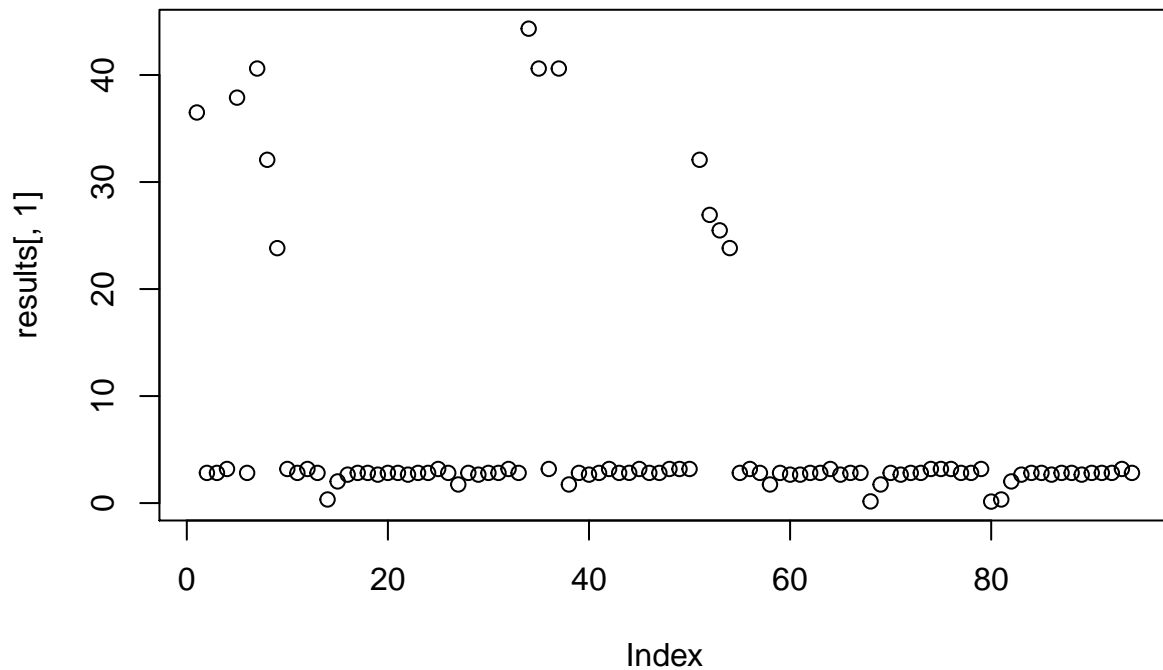


```
## 1 factor completed in 0.00419 minutes. Estimated time of completion: 2020-04-29 13:48:29 [1]
## [6] 2.998582e-06 3.000000e+01 4.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432866e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 40.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

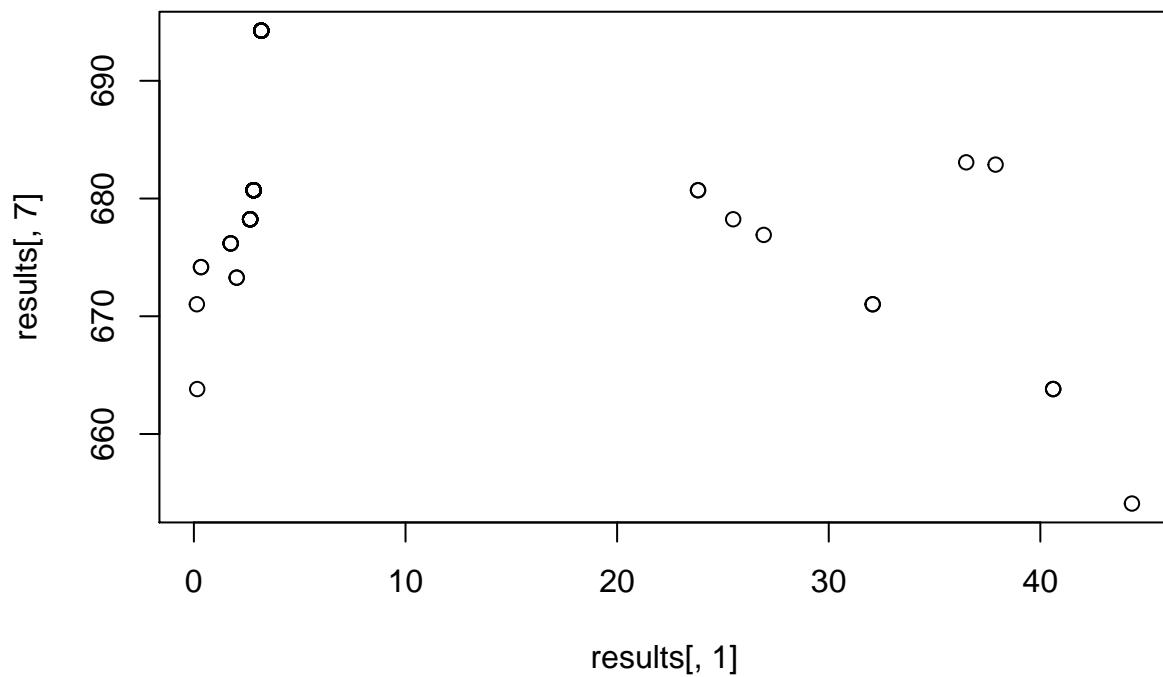


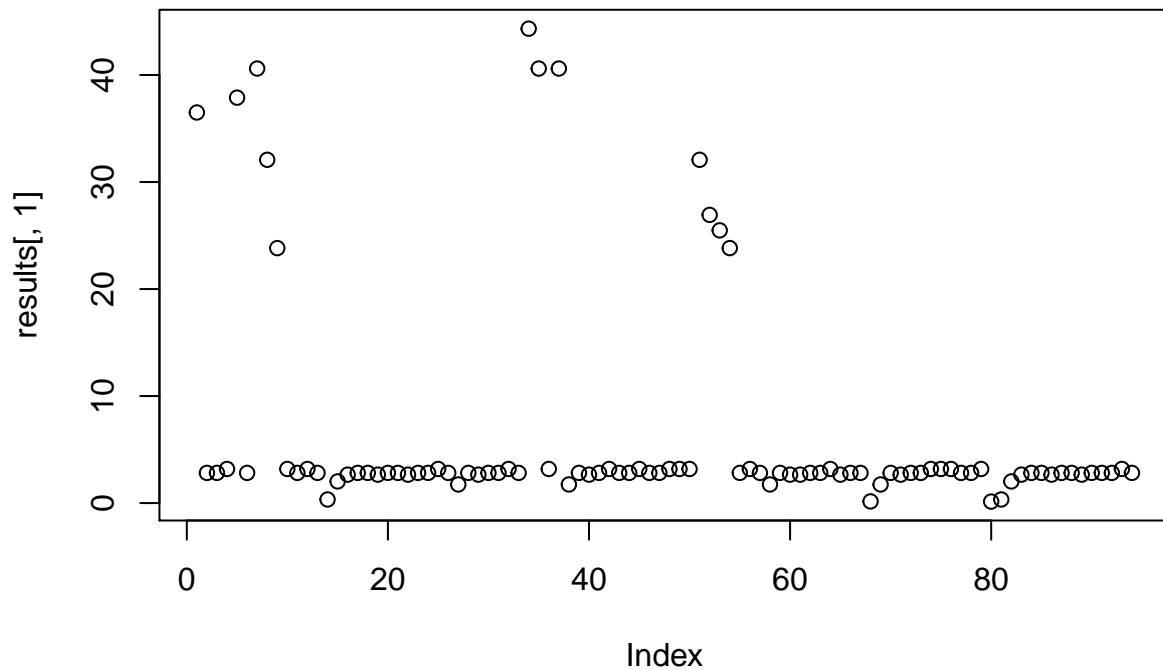


```
## 1 factor completed in 0.00475 minutes. Estimated time of completion: 2020-04-29 13:48:31 [1]
## [6] 1.264352e-08 4.000000e+01 4.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 50.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

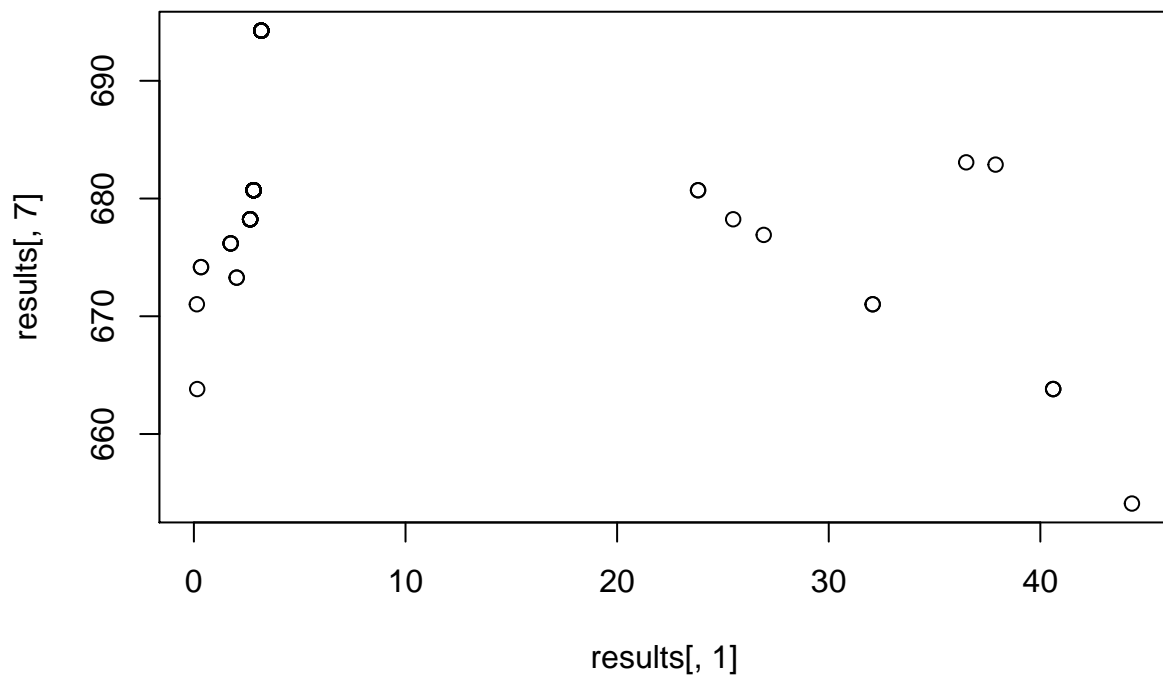


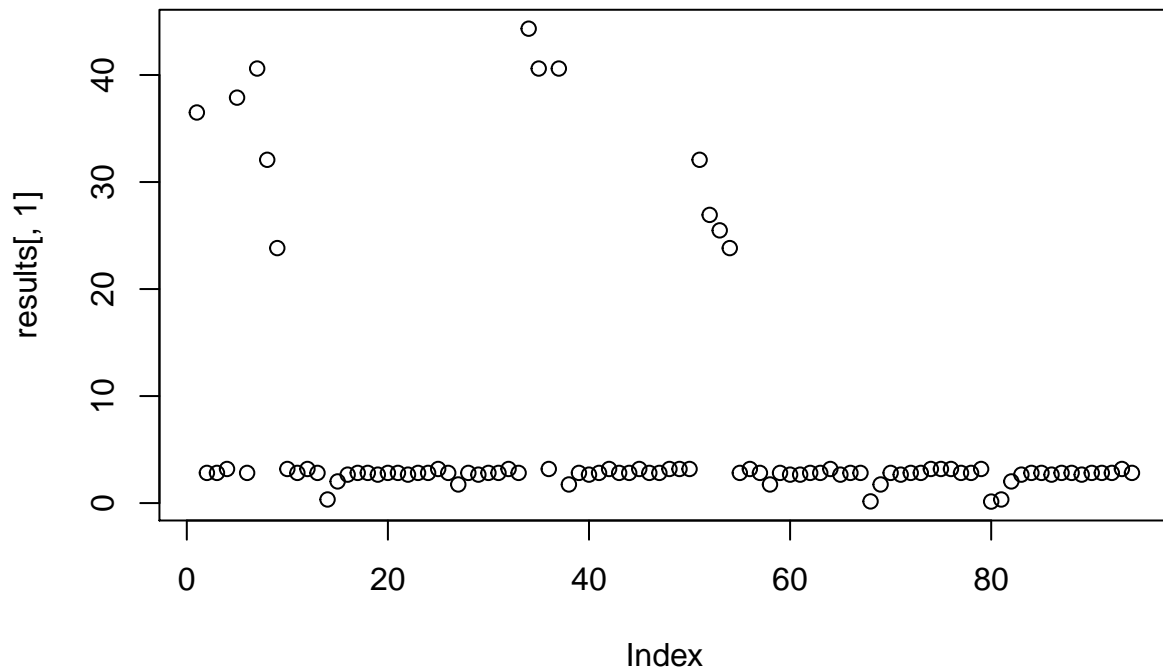


```
## 1 factor completed in 0.00671 minutes. Estimated time of completion: 2020-04-29 13:48:33 [1]
## [6] 5.331139e-11 5.000000e+01 4.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 60.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

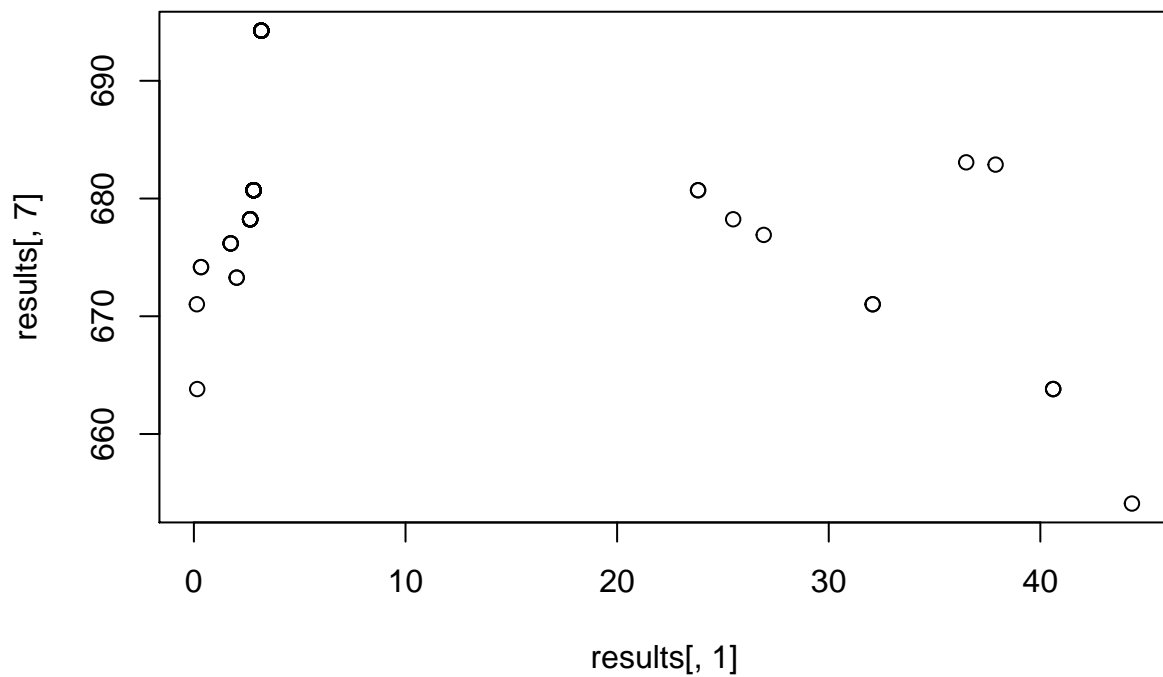


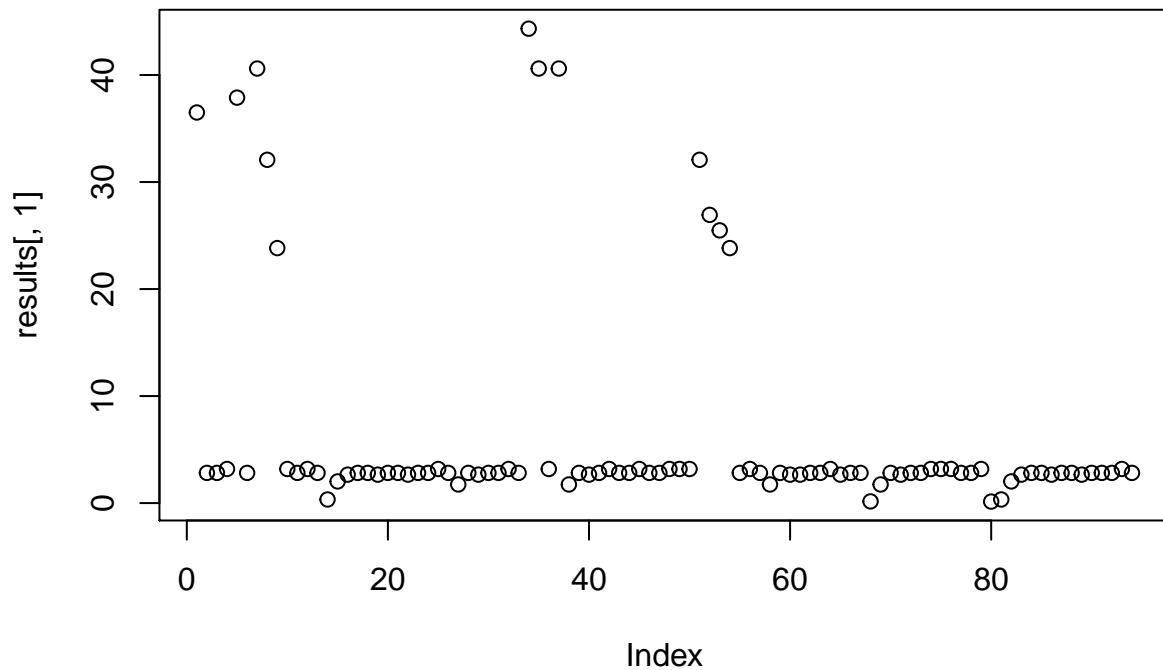


```
## 1 factor completed in 0.00552 minutes. Estimated time of completion: 2020-04-29 13:48:36 [1]
## [6] 2.247875e-13 6.000000e+01 4.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 70.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

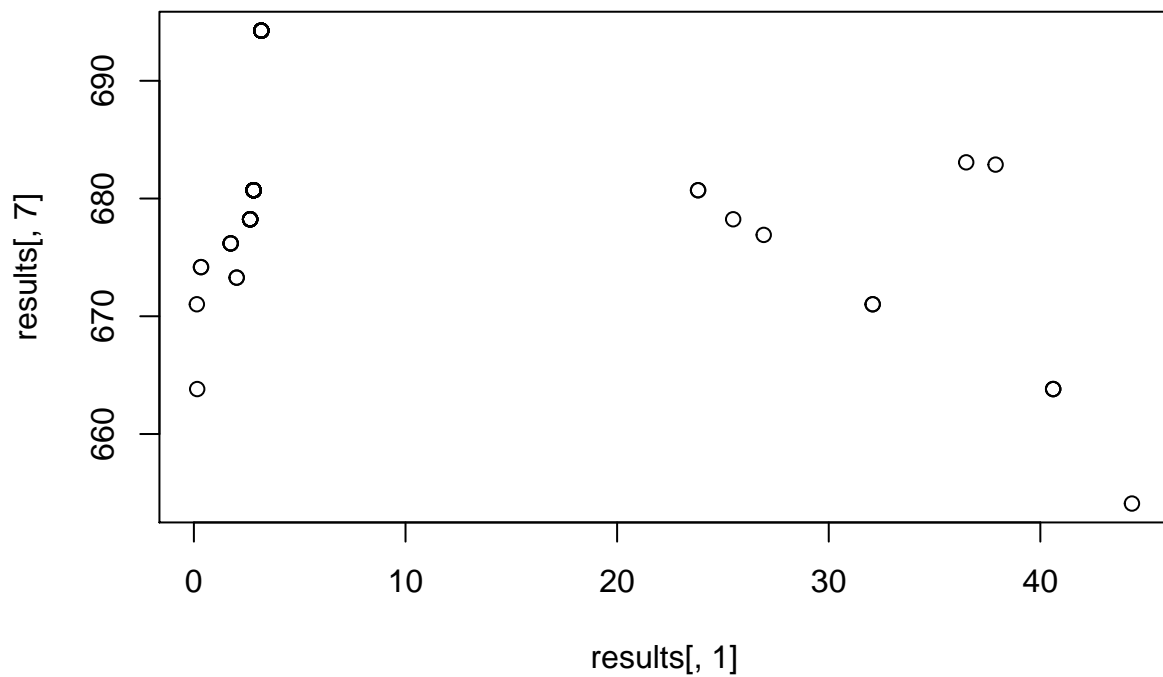


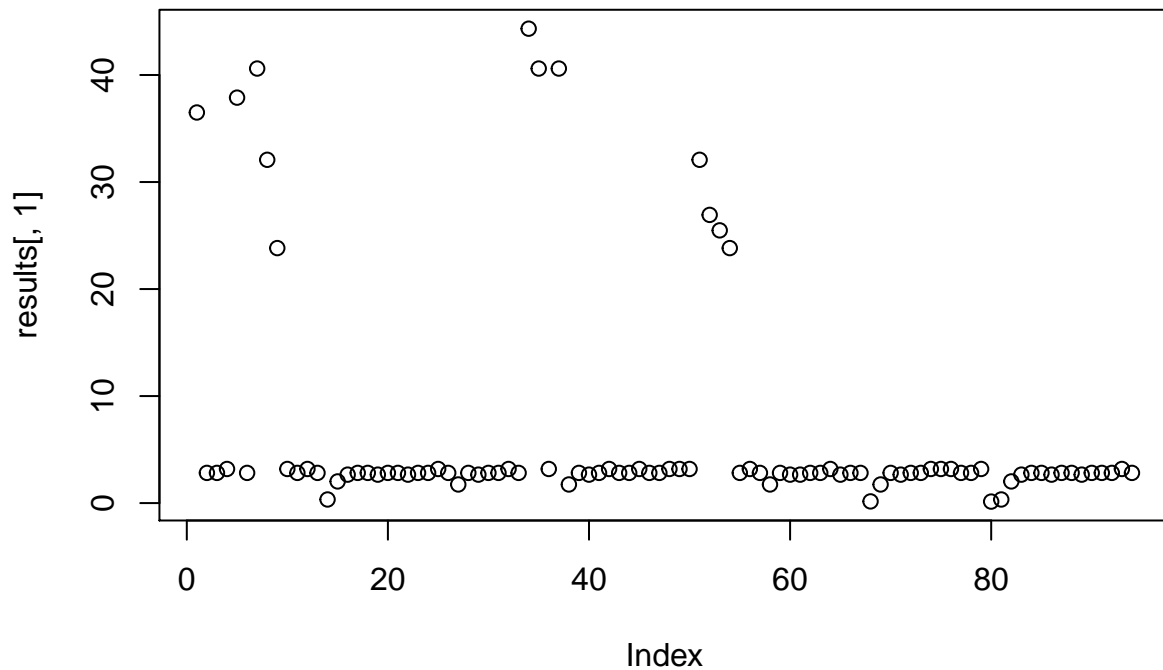


```
## 1 factor completed in 0.00688 minutes. Estimated time of completion: 2020-04-29 13:48:39 [1]
## [6] 9.478162e-16 7.000000e+01 4.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 80.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

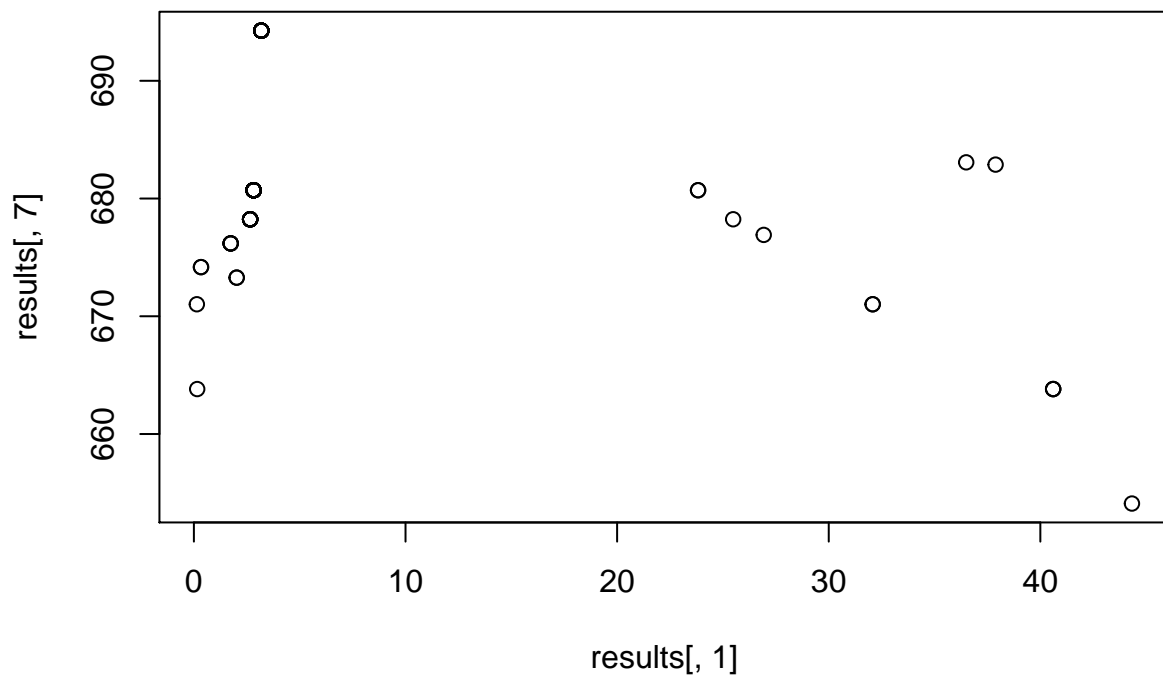


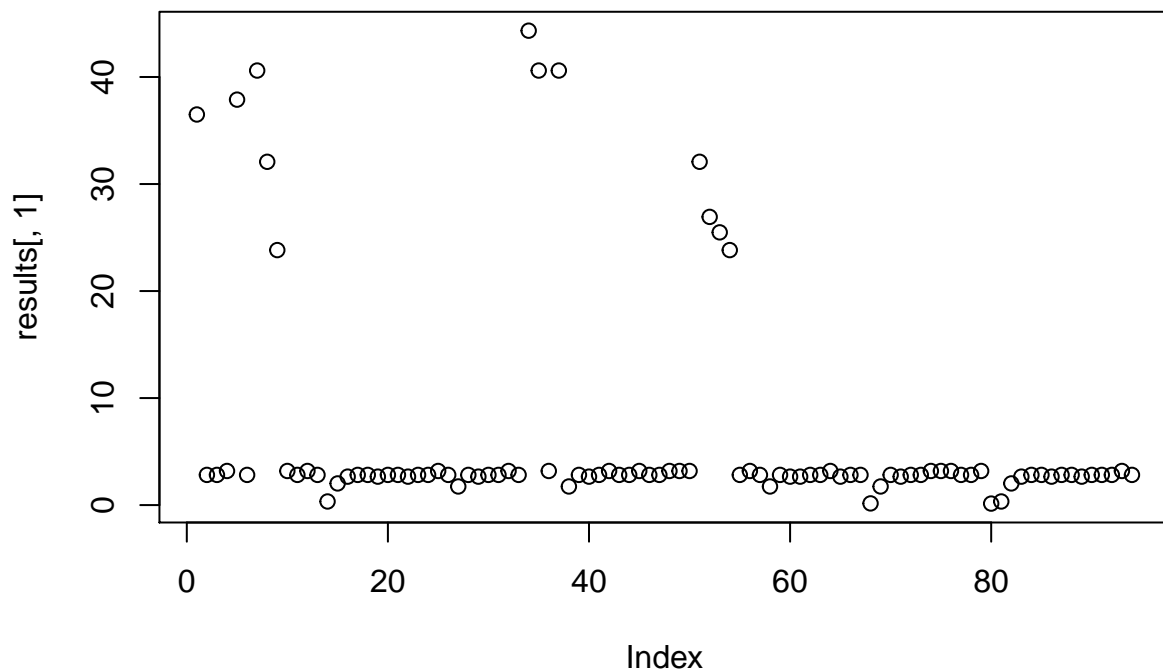


```
## 1 factor completed in 0.00397 minutes. Estimated time of completion: 2020-04-29 13:48:41 [1]
## [6] 3.996467e-18 8.000000e+01 4.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 90.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

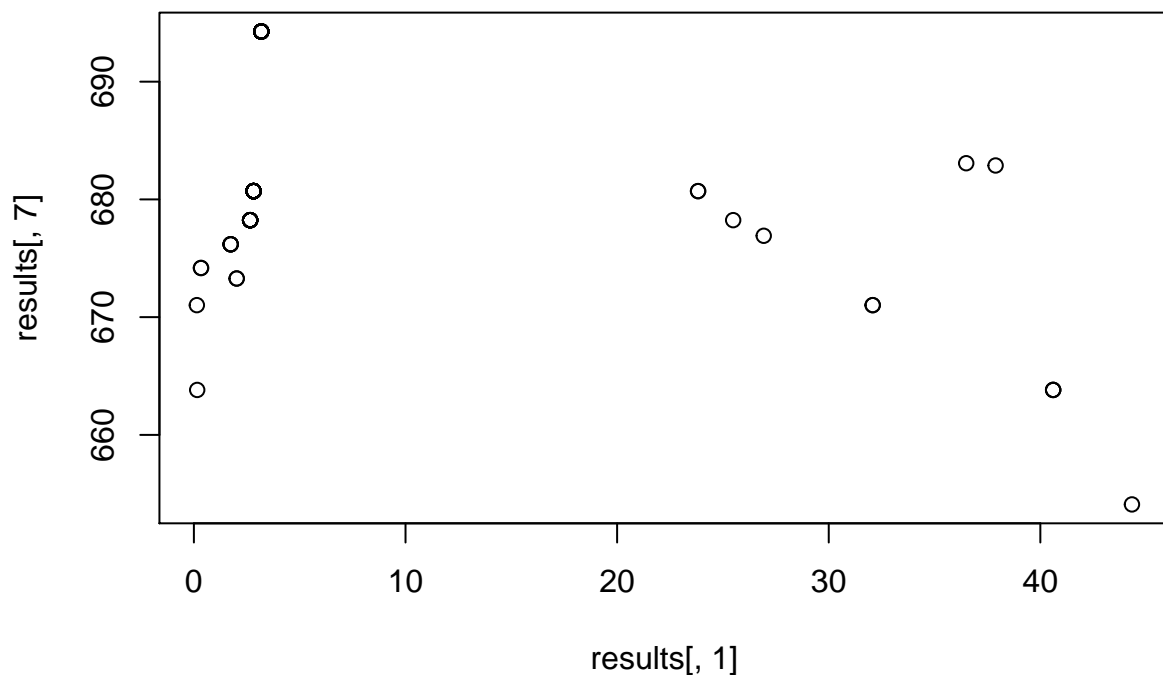


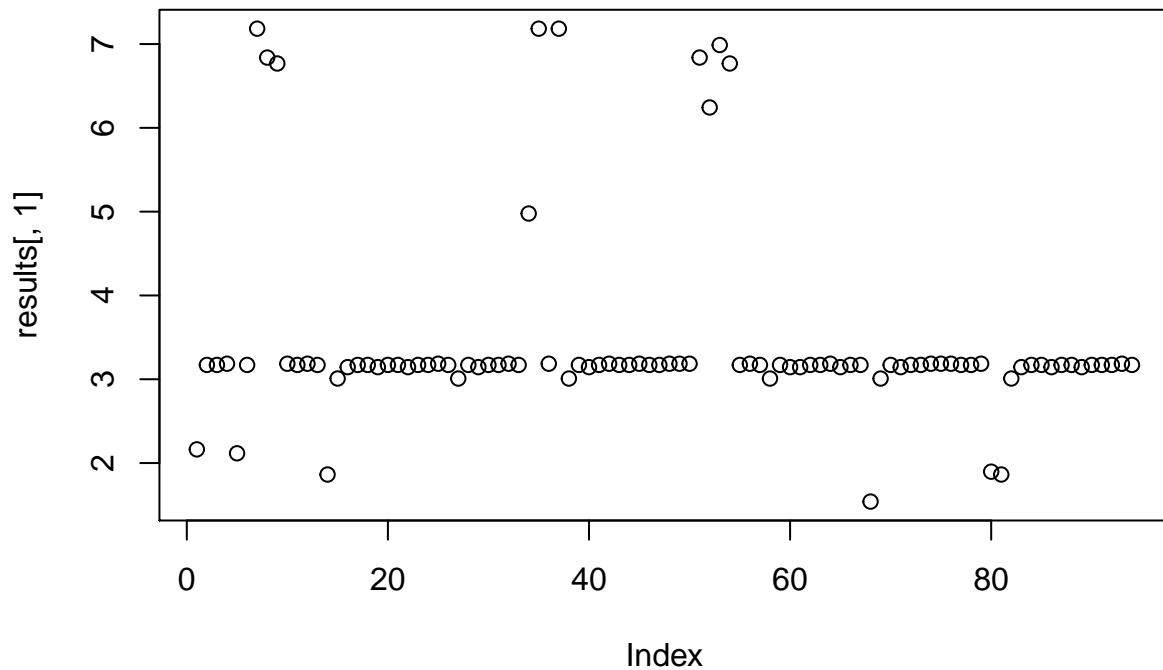


```
## 1 factor completed in 0.00393 minutes. Estimated time of completion: 2020-04-29 13:48:43 [1]
## [6] 1.685110e-20 9.000000e+01 4.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "X = 50.000001"
## [1] "j = 1e-06"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

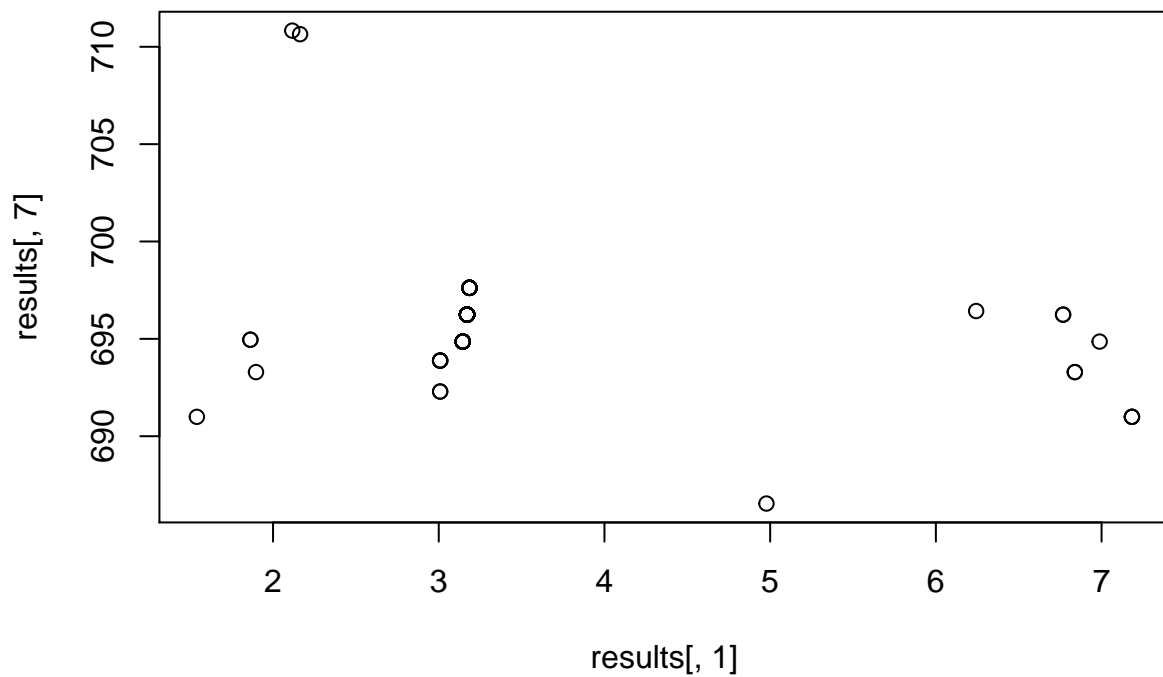


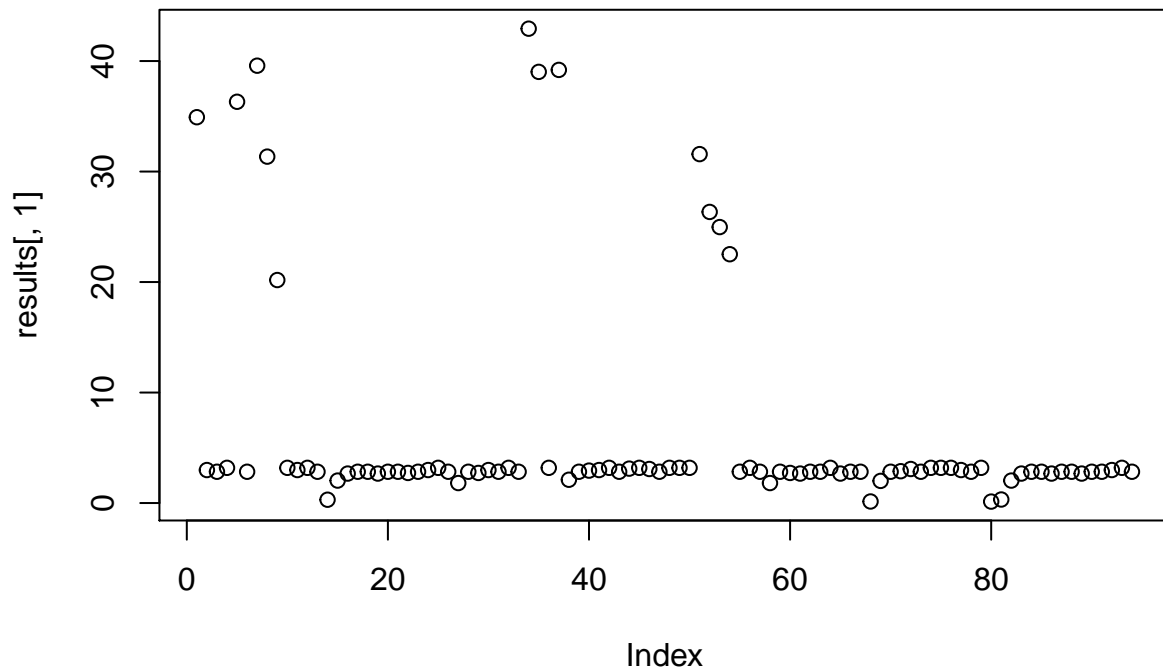


```
## 1 factor completed in 0.00373 minutes. Estimated time of completion: 2020-04-29 13:48:45 [1]
## [7] 0.0000010 50.0000010 7.8130568 690.9940183 4.9775769 686.5415229
## [13] 68.0000000 34.0000000
## [1] "j = 10.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

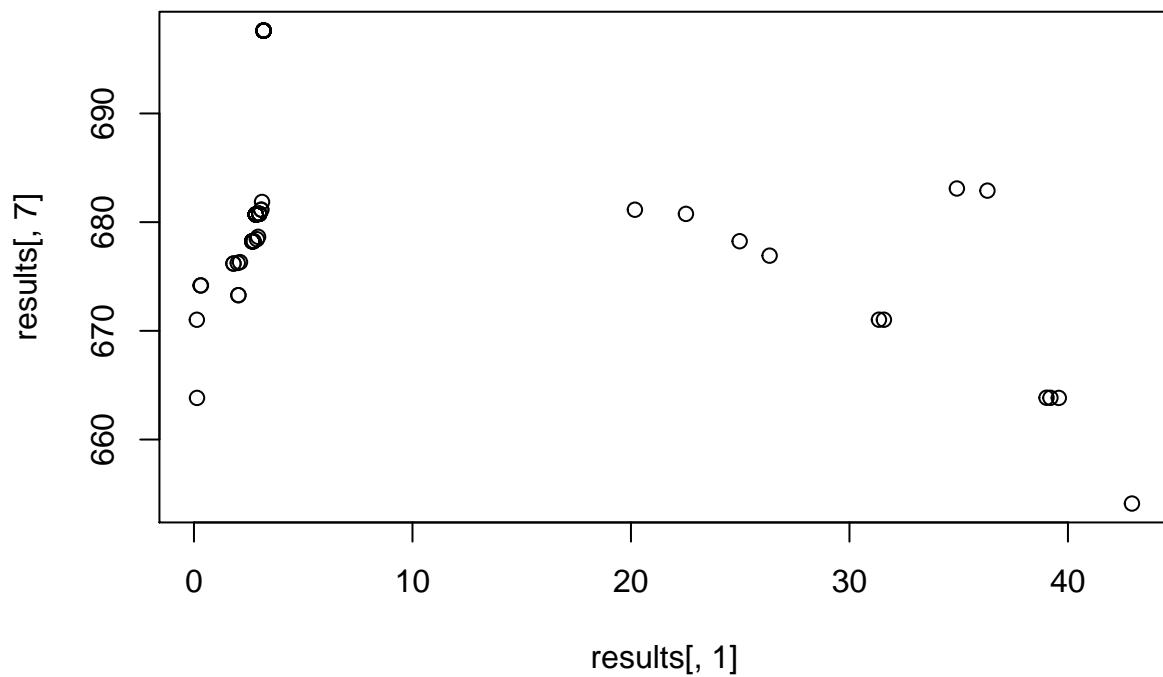


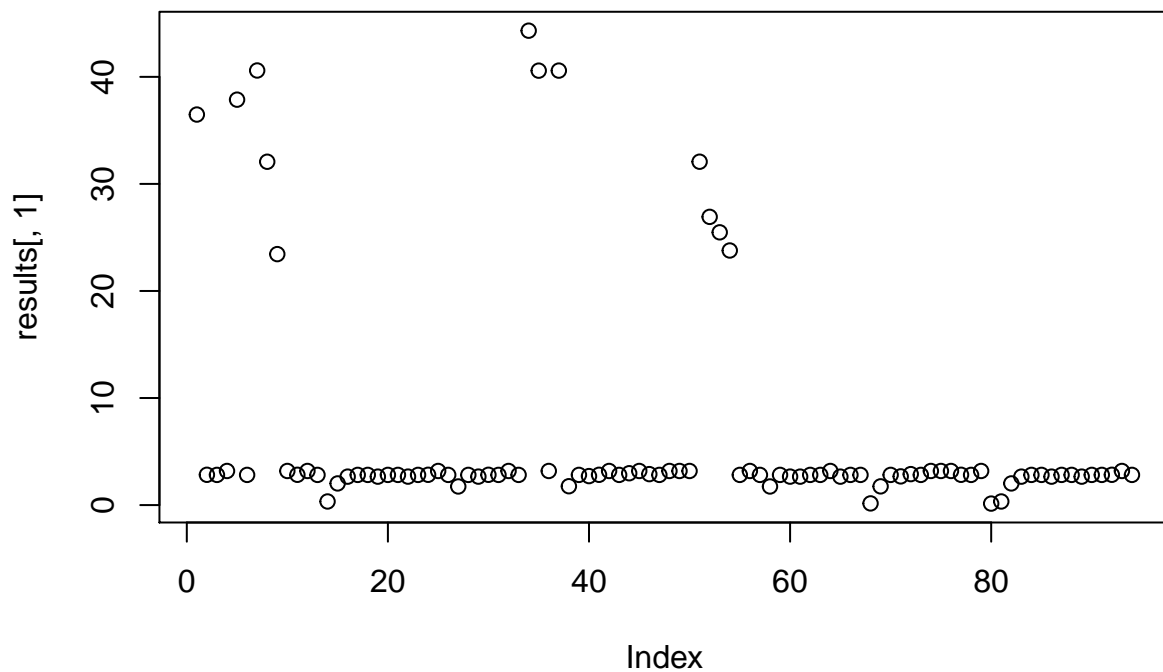


```
## 1 factor completed in 0.0039 minutes. Estimated time of completion: 2020-04-29 13:48:47 [1]
## [7] 10.0000010 50.0000010 7.8130568 671.0213336 42.9289653 654.1136470
## [13] 80.0000000 34.0000000
## [1] "j = 20.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

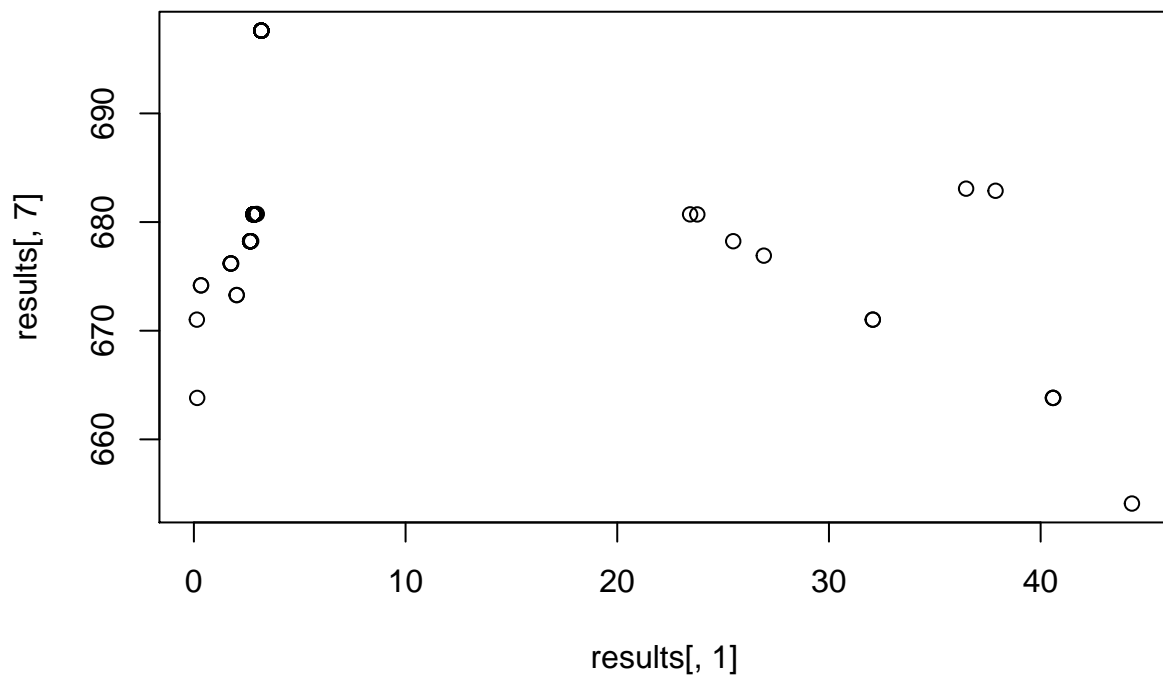


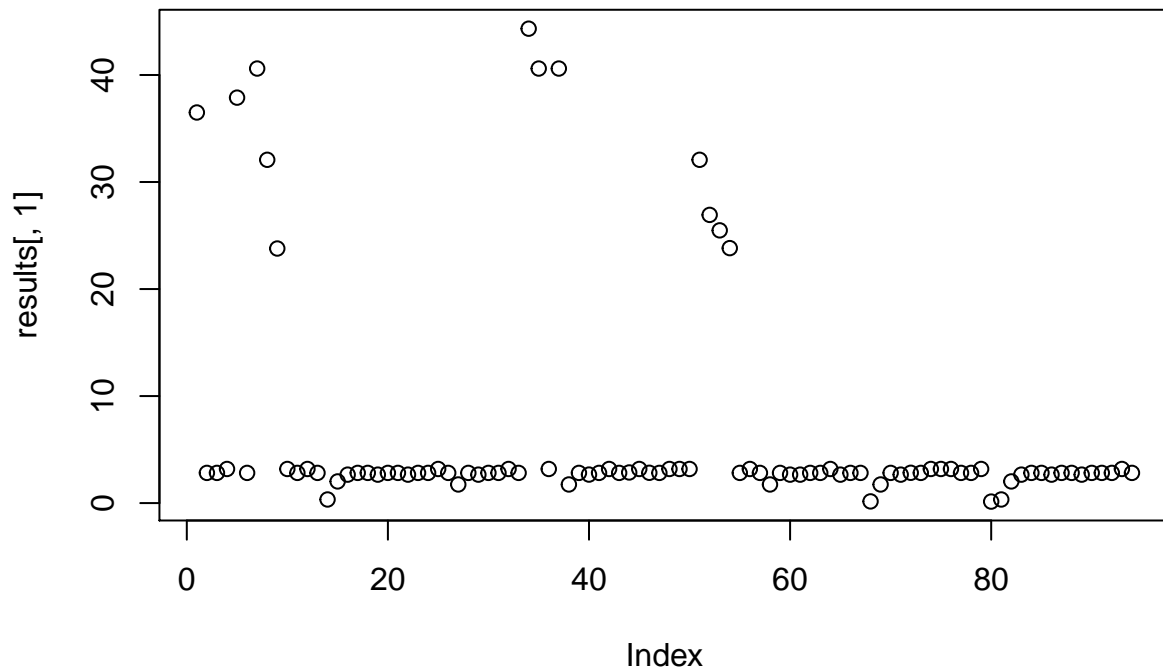


```
## 1 factor completed in 0.00393 minutes. Estimated time of completion: 2020-04-29 13:48:49 [1]
## [6] 8.889429e-04 2.000000e+01 5.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.431374e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 30.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

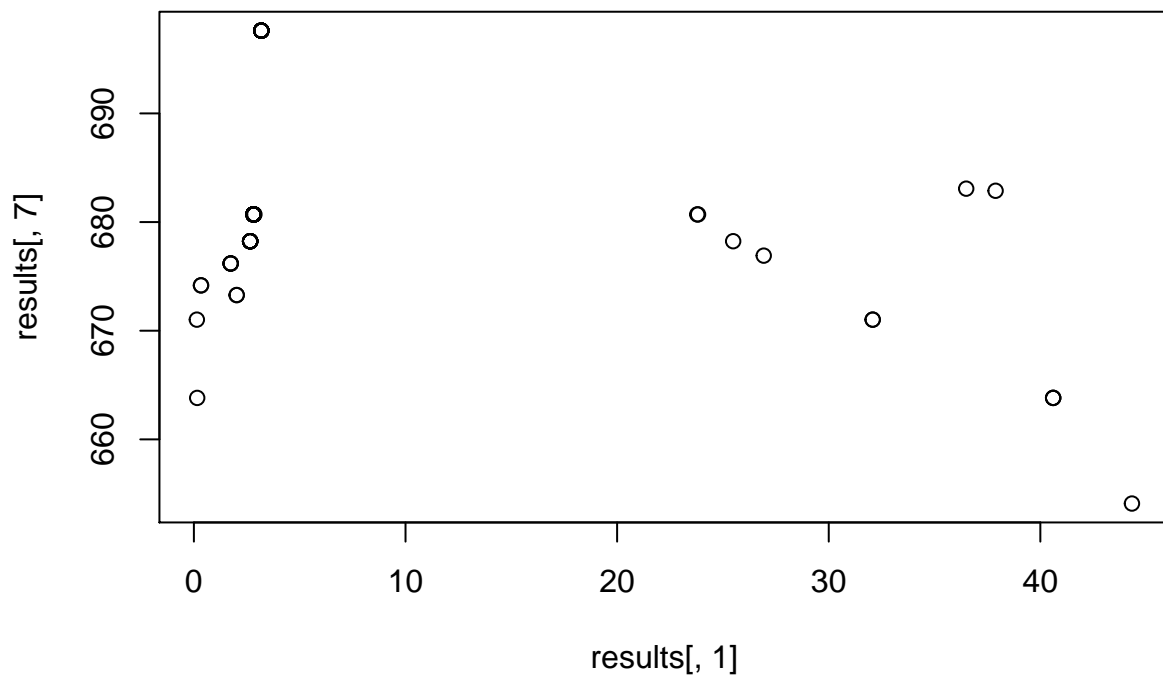


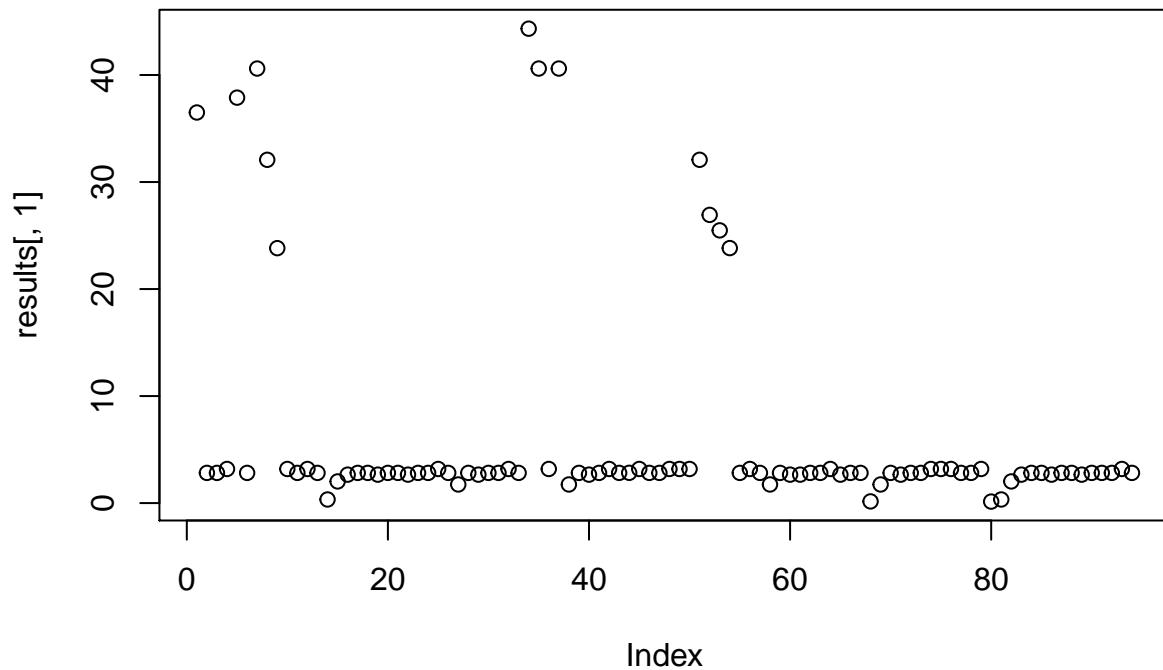


```
## 1 factor completed in 0.00385 minutes. Estimated time of completion: 2020-04-29 13:48:51 [1]
## [6] 3.748227e-06 3.000000e+01 5.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432863e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 40.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

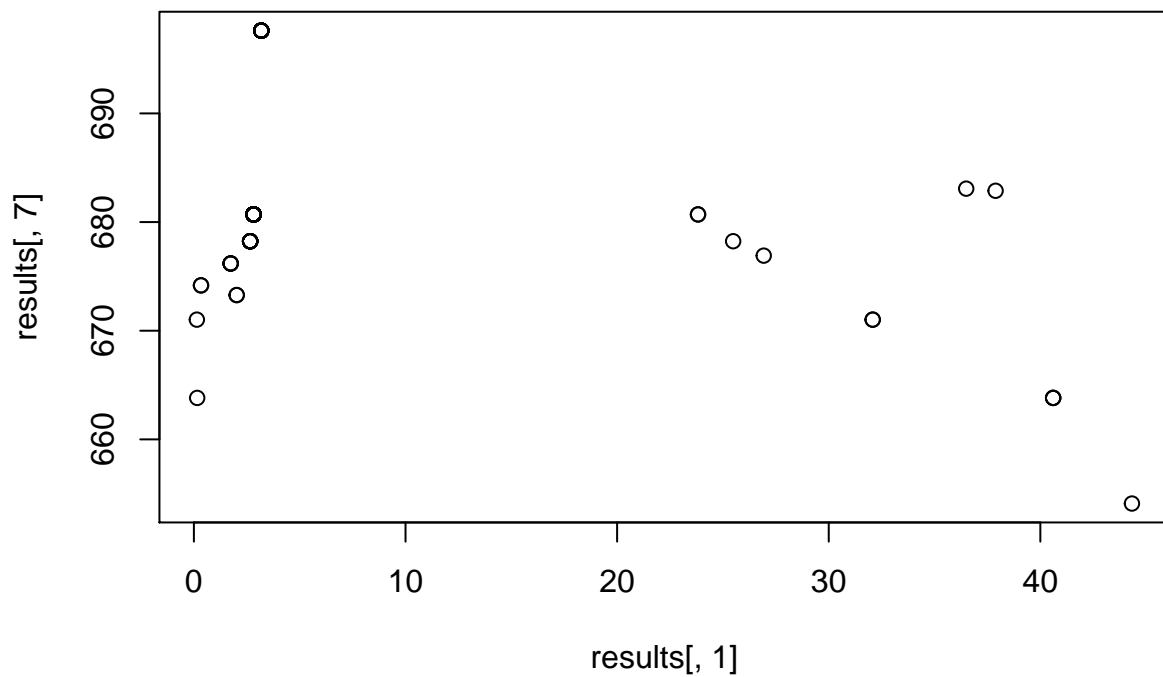


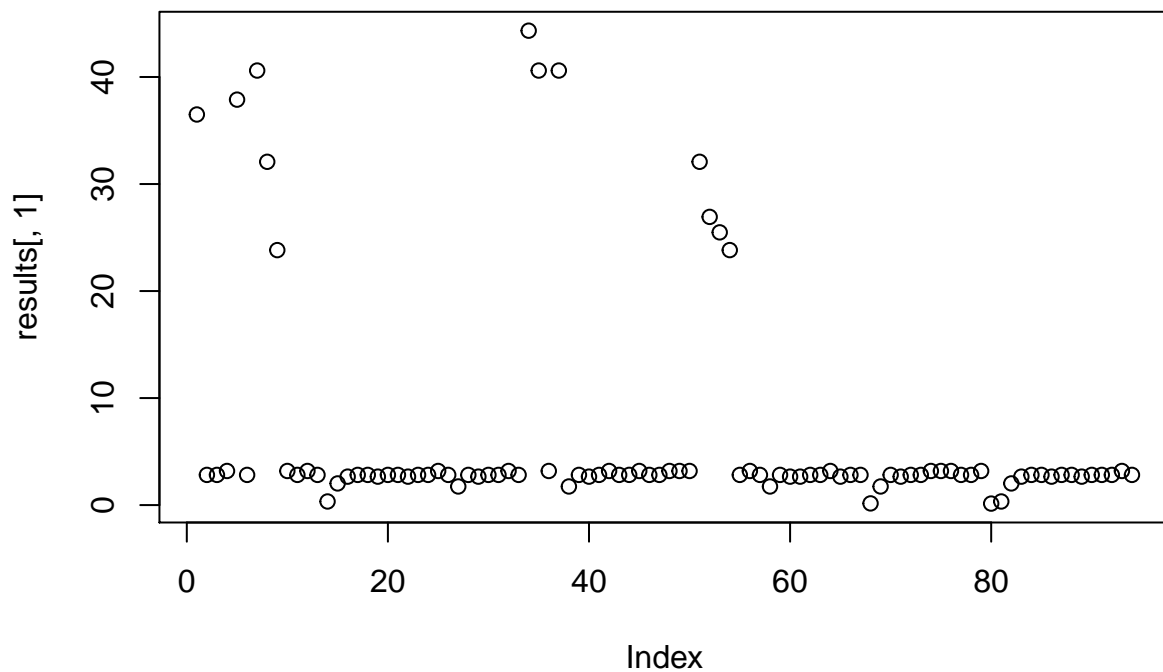


```
## 1 factor completed in 0.0044 minutes. Estimated time of completion: 2020-04-29 13:48:53 [1]
## [6] 1.580440e-08 4.000000e+01 5.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 50.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

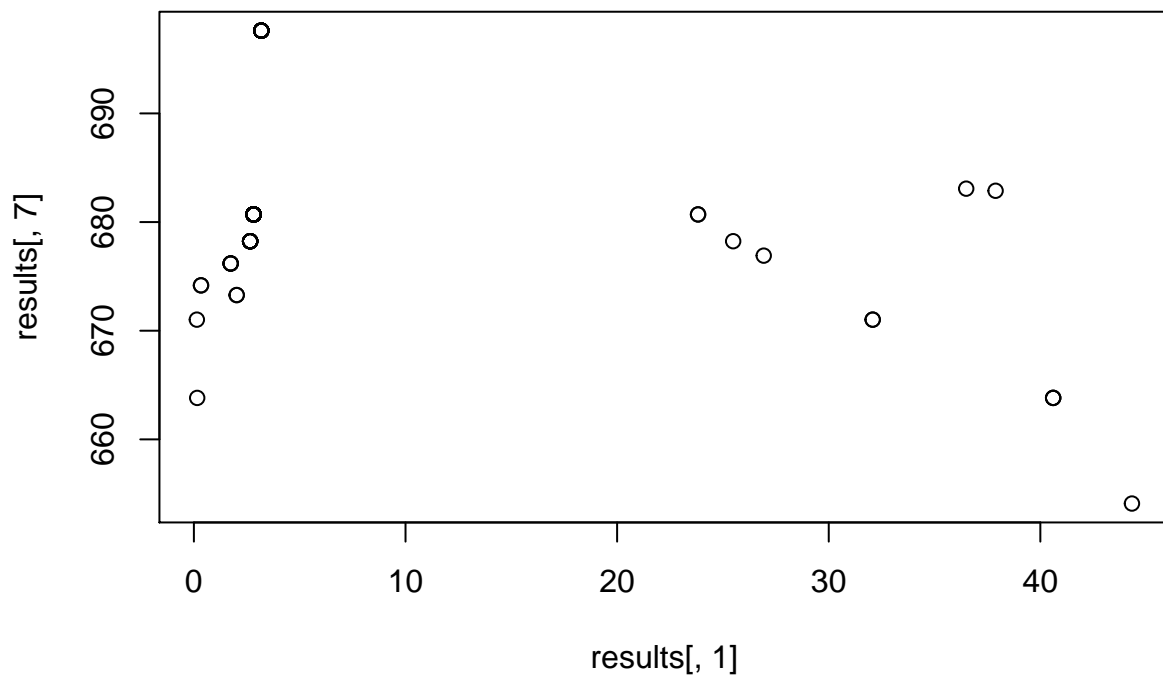


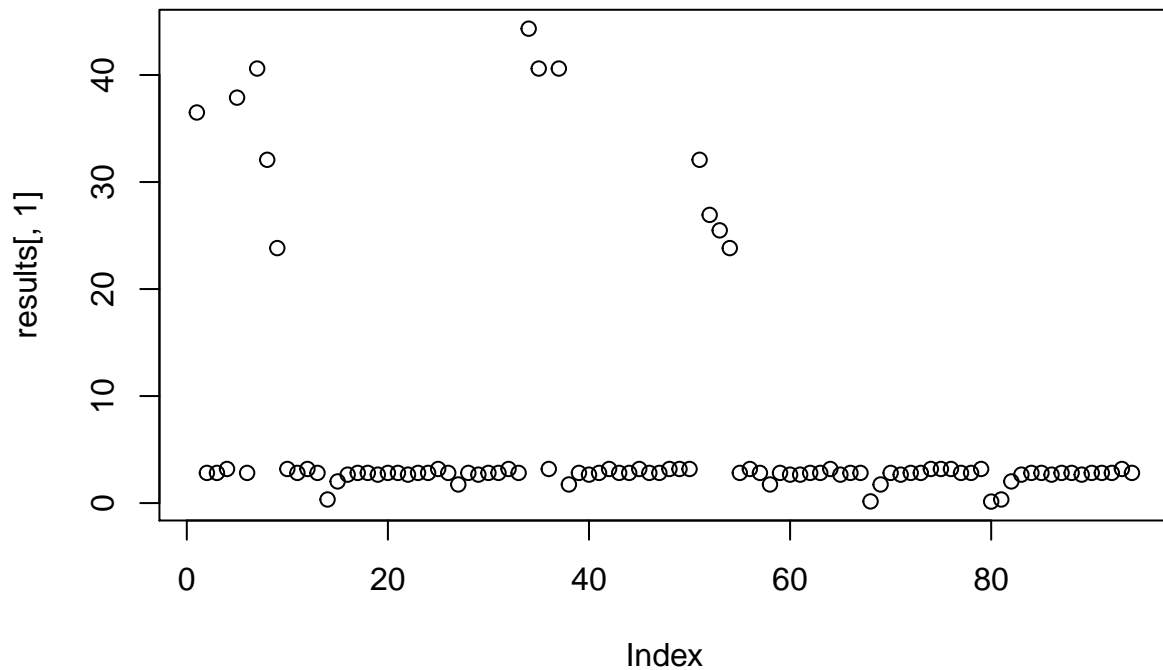


```
## 1 factor completed in 0.00558 minutes. Estimated time of completion: 2020-04-29 13:48:55 [1]
## [6] 6.663924e-11 5.000000e+01 5.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 60.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

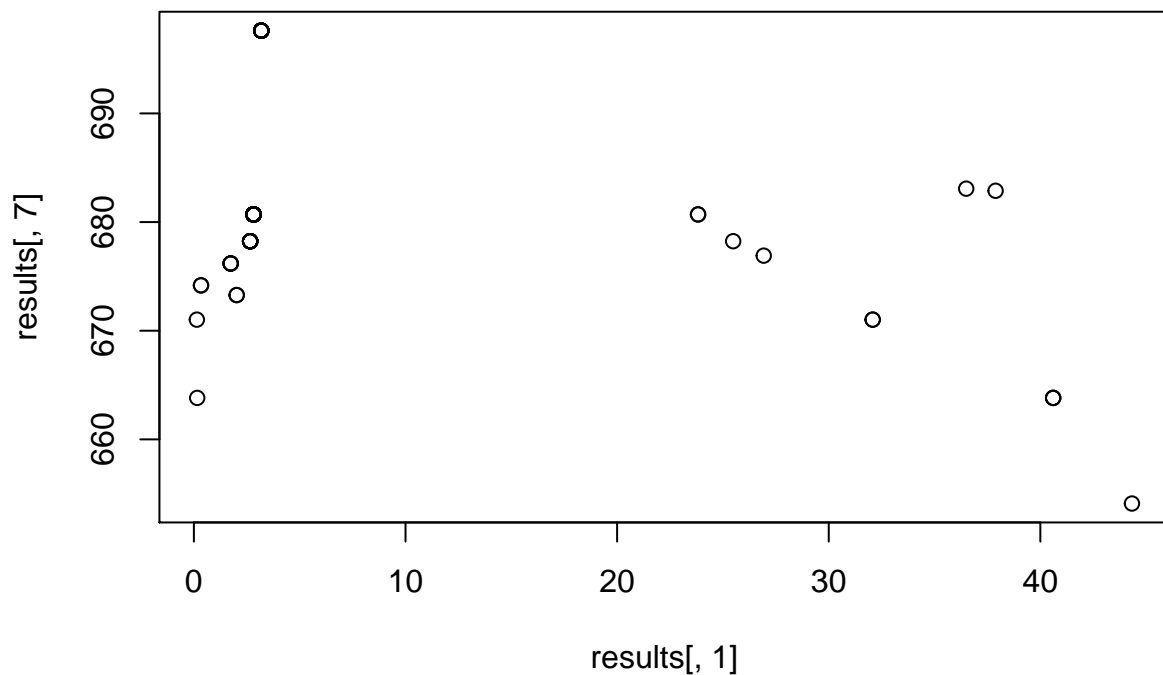


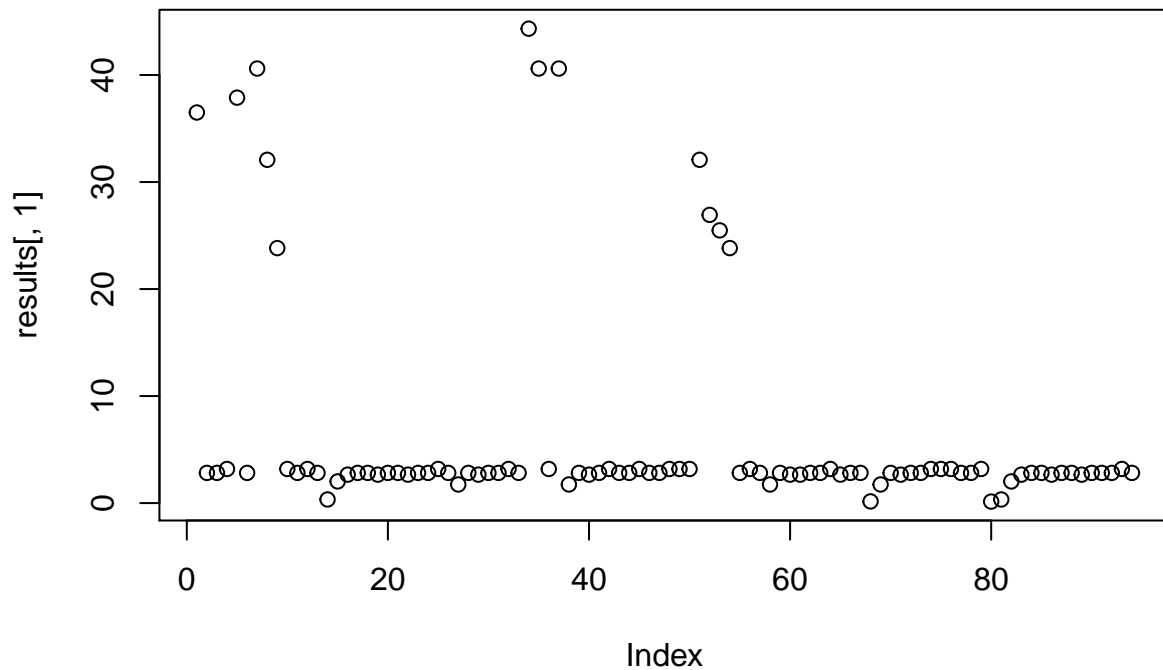


```
## 1 factor completed in 0.00379 minutes. Estimated time of completion: 2020-04-29 13:48:58 [1]
## [6] 2.809843e-13 6.000000e+01 5.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 70.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

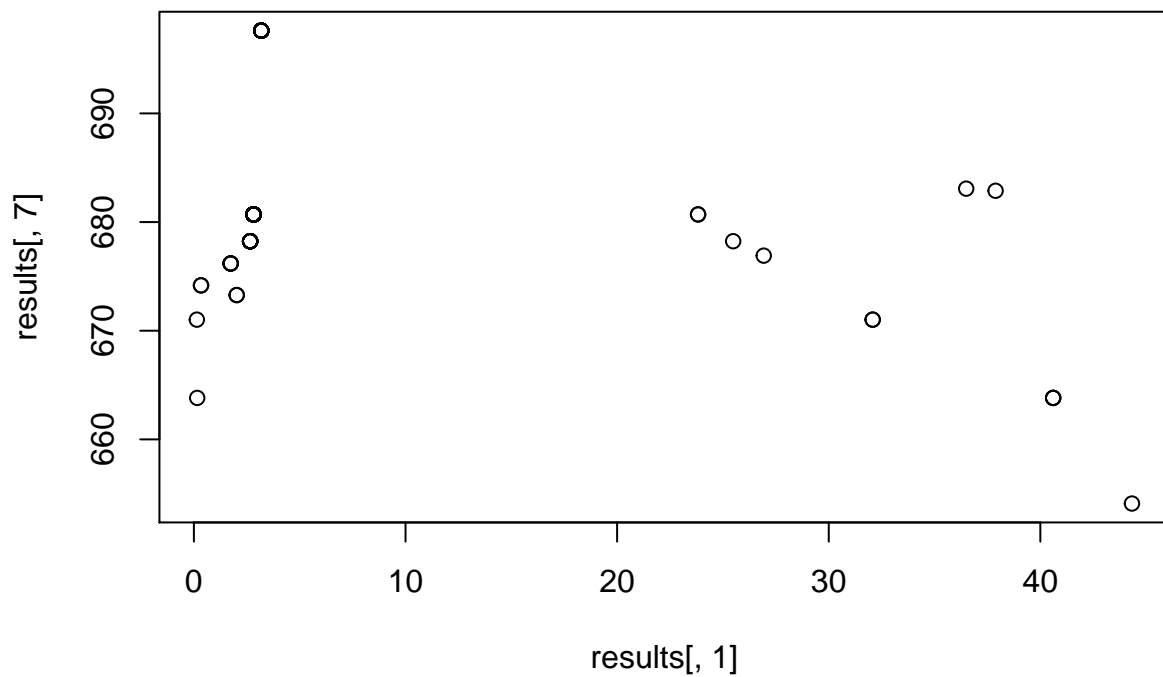


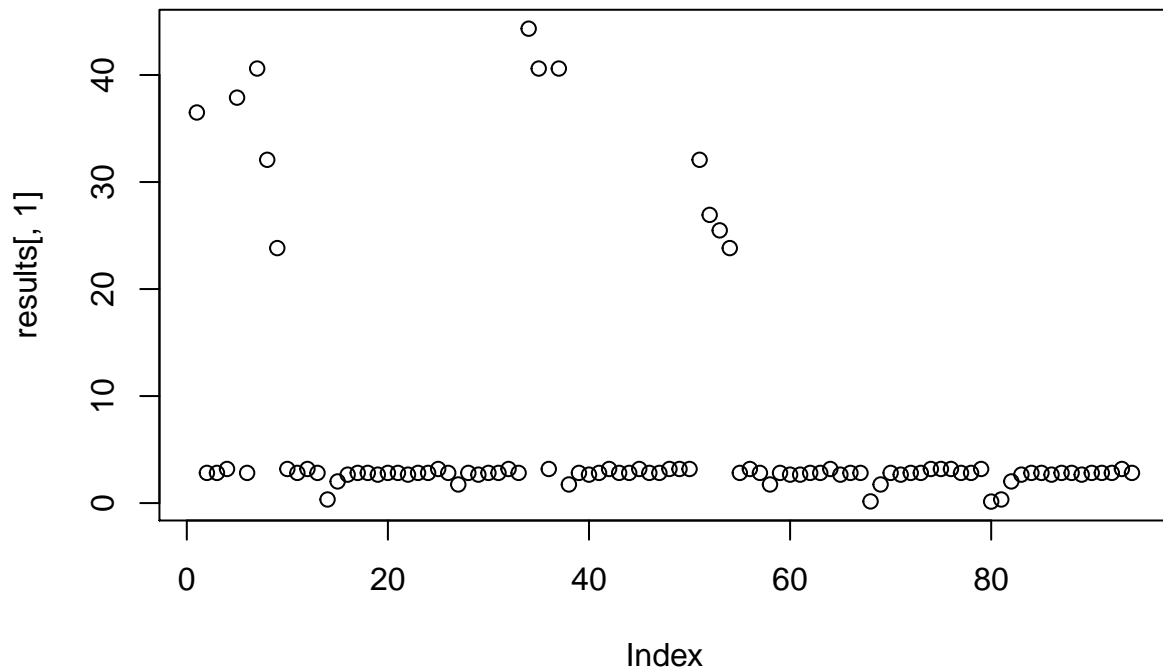


```
## 1 factor completed in 0.00383 minutes. Estimated time of completion: 2020-04-29 13:48:59 [1]
## [6] 1.184770e-15 7.000000e+01 5.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 80.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

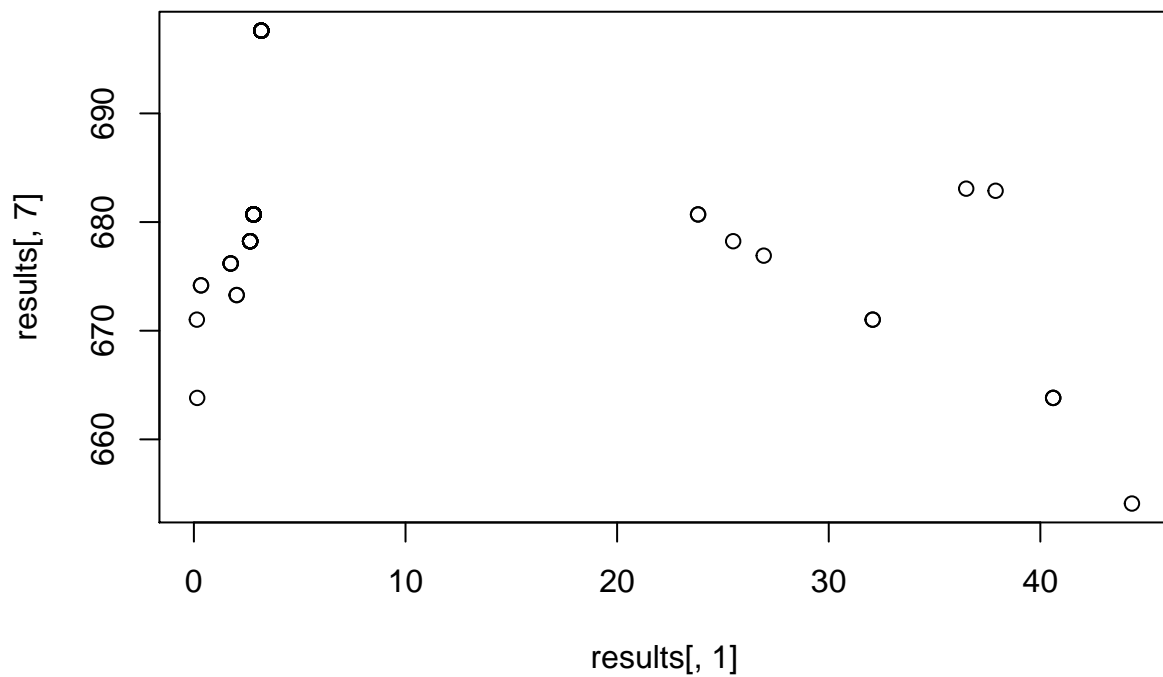


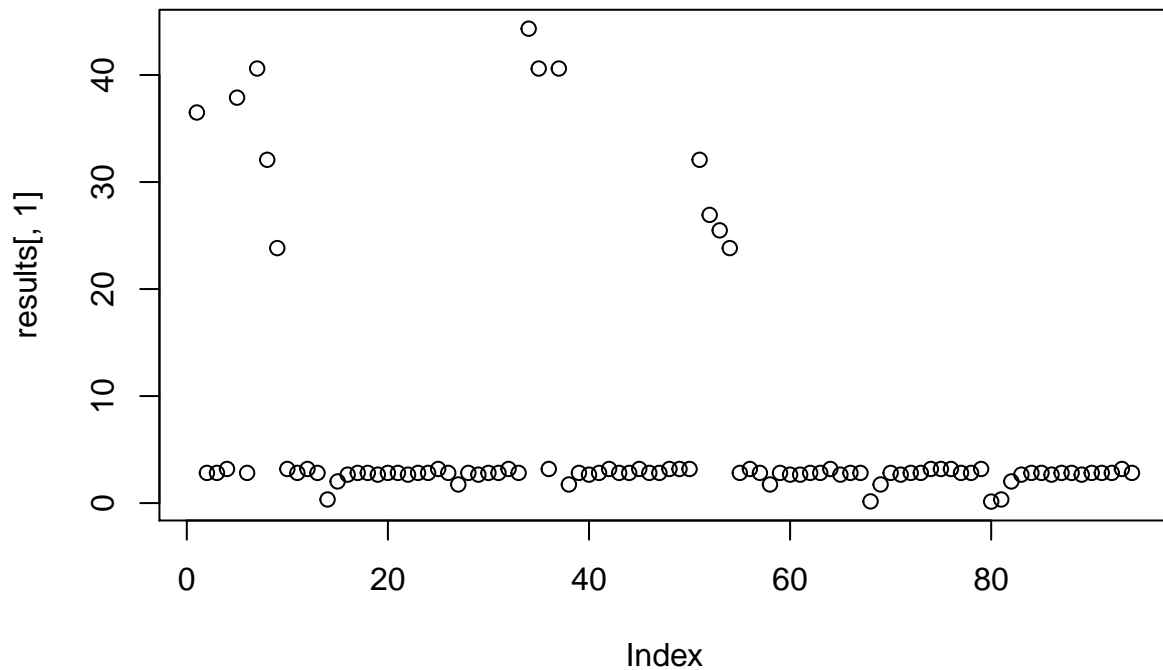


```
## 1 factor completed in 0.00383 minutes. Estimated time of completion: 2020-04-29 13:49:01 [1]
## [6] 4.995583e-18 8.000000e+01 5.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 90.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

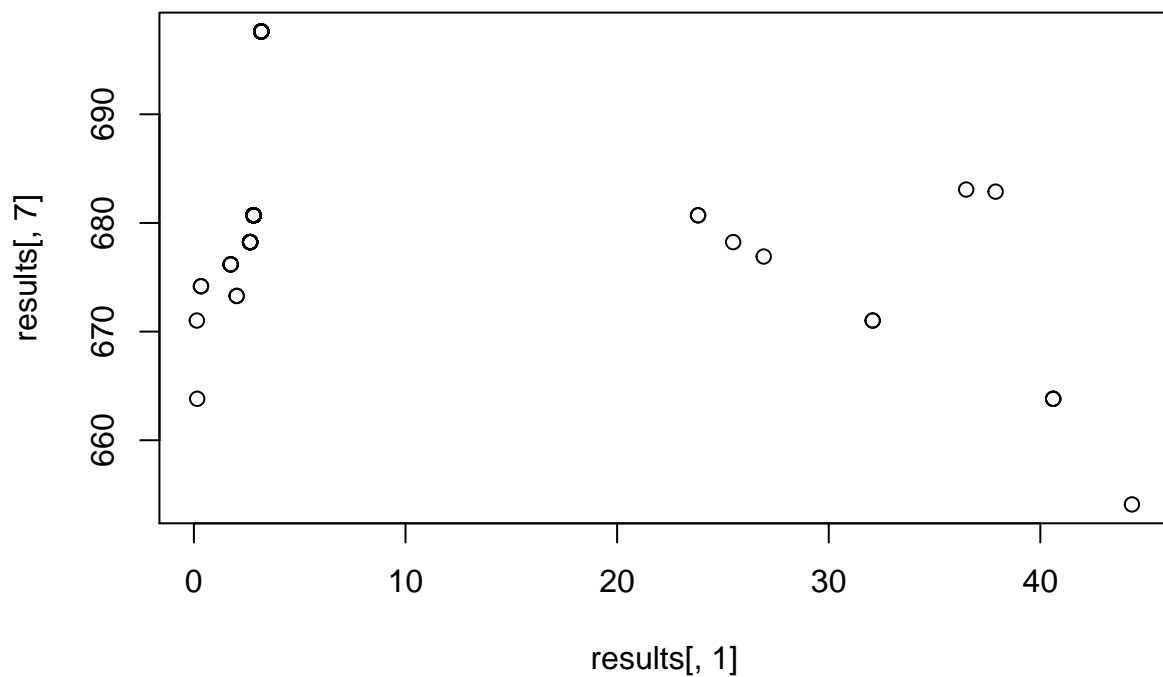


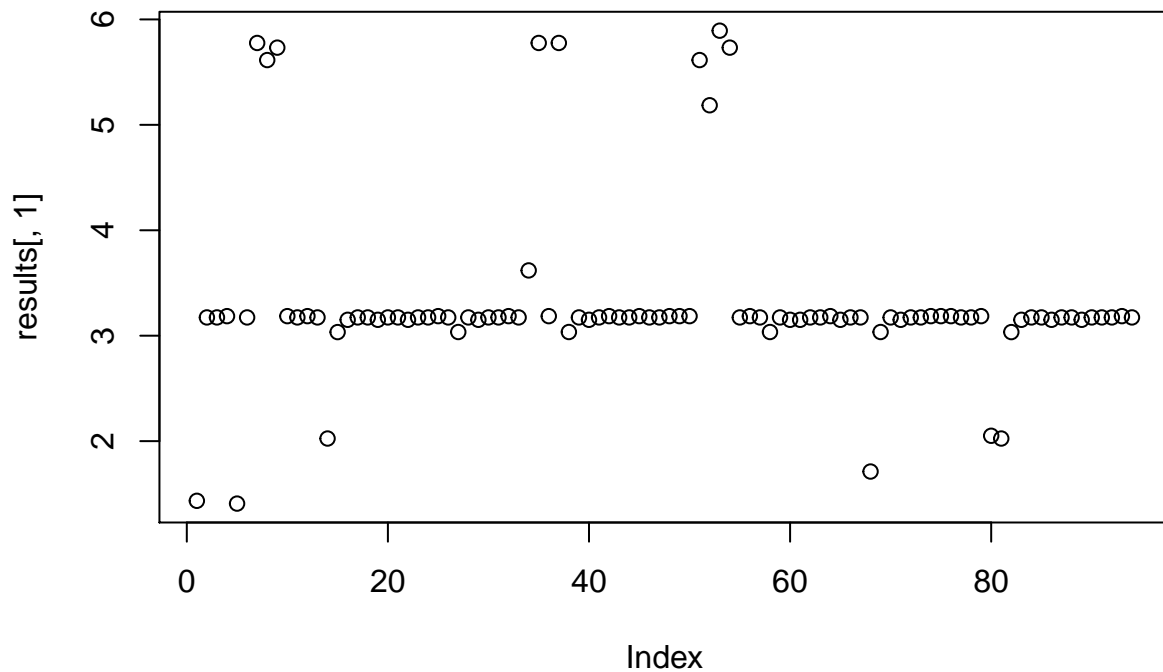


```
## 1 factor completed in 0.00387 minutes. Estimated time of completion: 2020-04-29 13:49:03 [1]
## [6] 2.106387e-20 9.000000e+01 5.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "X = 60.000001"
## [1] "j = 1e-06"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

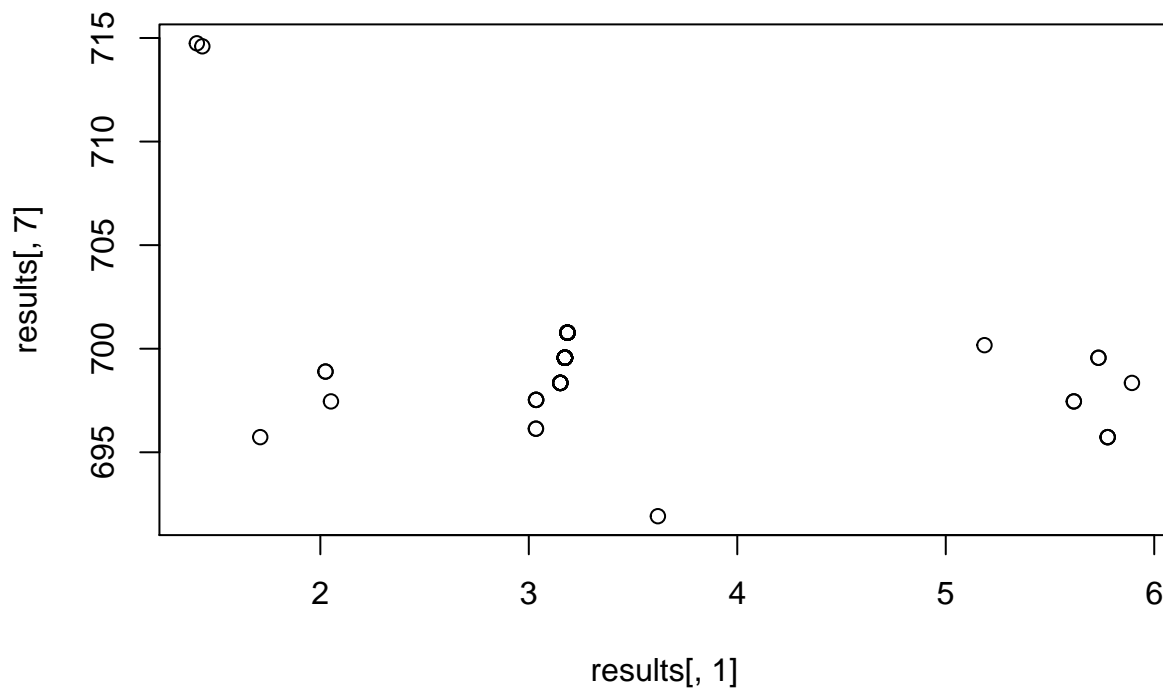


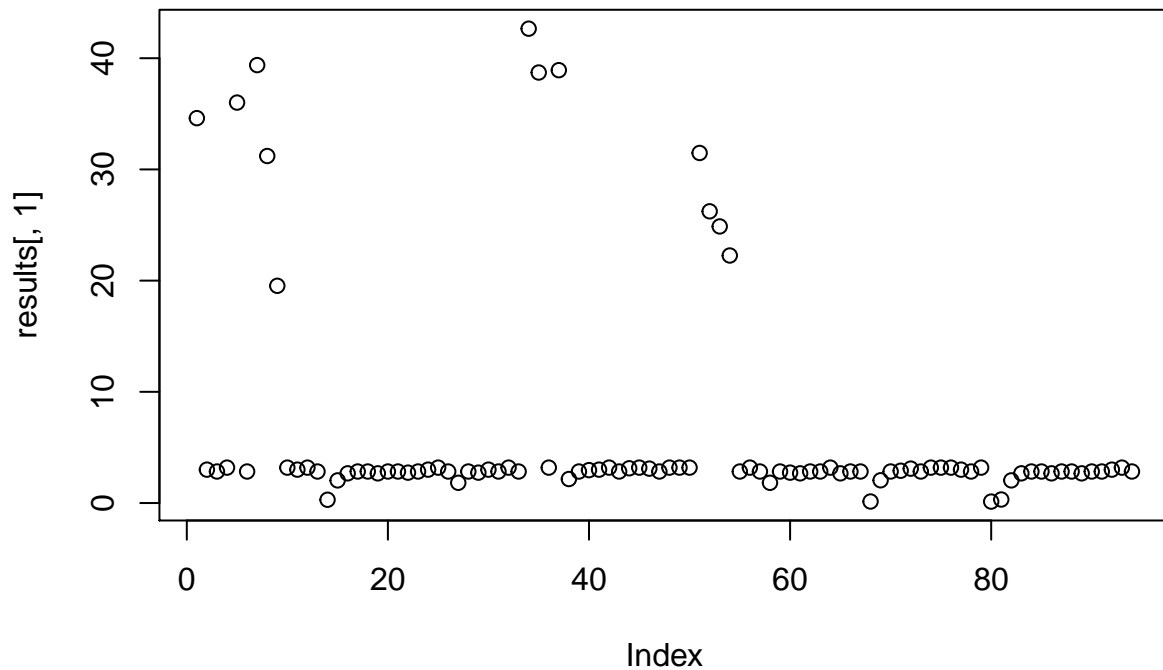


```
## 1 factor completed in 0.00408 minutes. Estimated time of completion: 2020-04-29 13:49:05 [1]
## [7] 0.0000010 60.0000010 7.8130568 714.7428145 3.6192305 691.9150507
## [13] 5.0000000 34.0000000
## [1] "j = 10.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

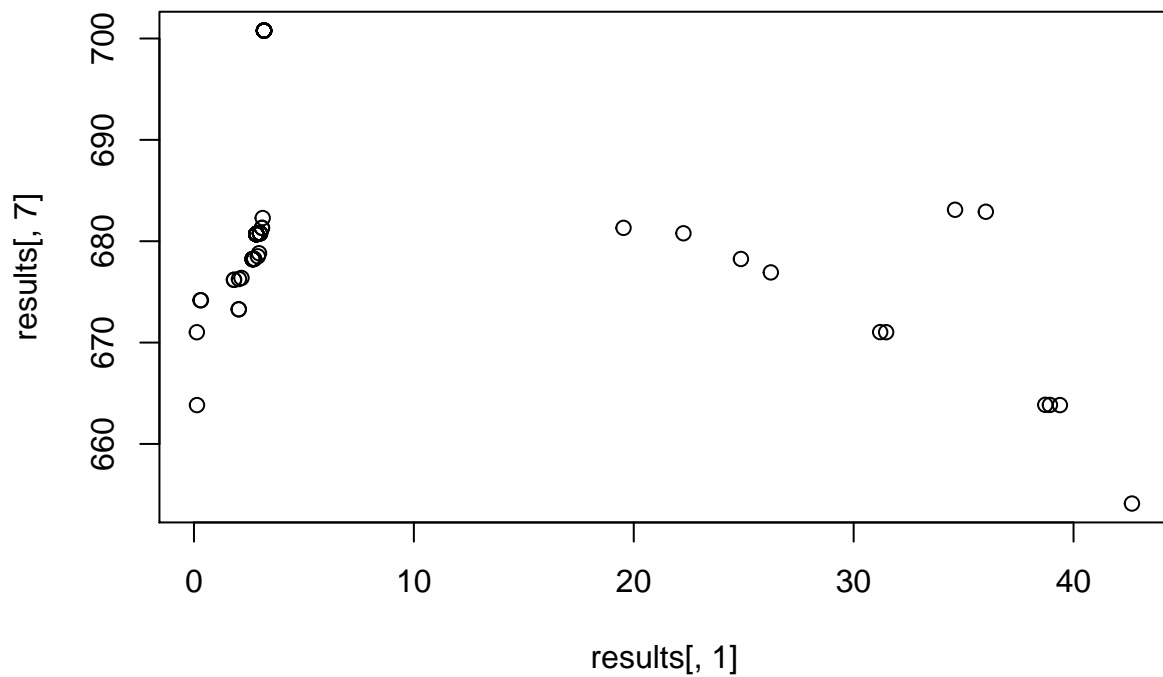


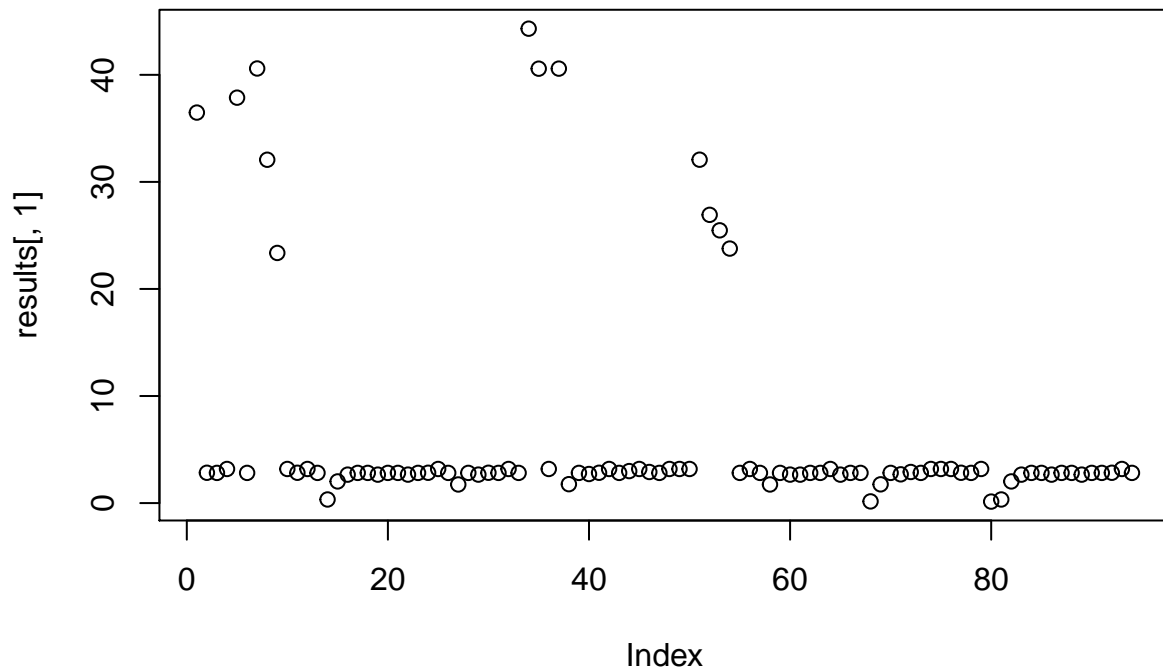


```
## 1 factor completed in 0.00372 minutes. Estimated time of completion: 2020-04-29 13:49:07 [1]
## [7] 10.0000010 60.0000010 7.8130568 671.0220966 42.6561206 654.1214149
## [13] 80.0000000 34.0000000
## [1] "j = 20.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

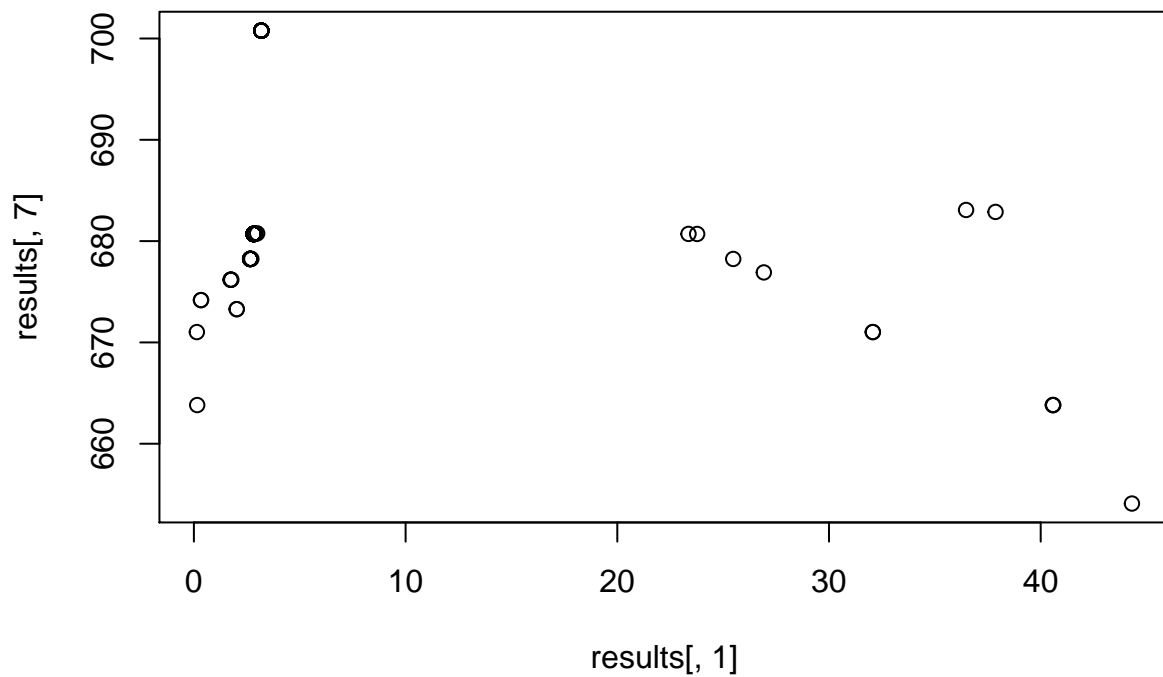


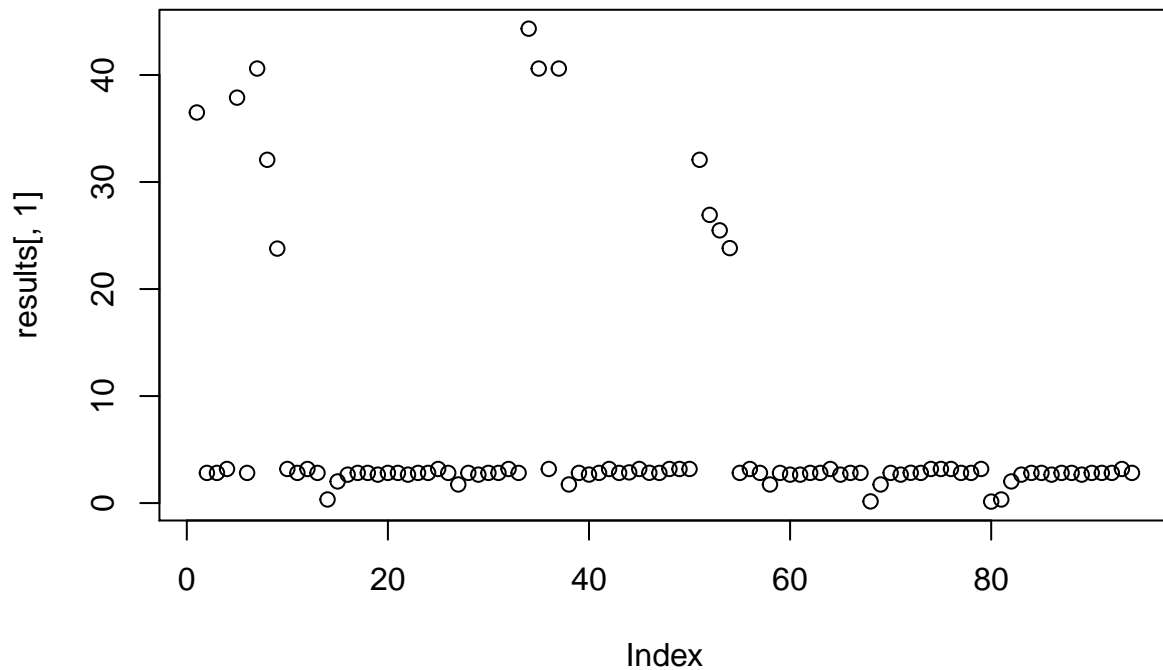


```
## 1 factor completed in 0.00403 minutes. Estimated time of completion: 2020-04-29 13:49:10 [1]
## [6] 0.001066732 20.000001000 60.000001000 7.813056846 671.019586862
## [11] 44.310734556 654.095685670 80.000000000 34.000000000
## [1] "j = 30.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

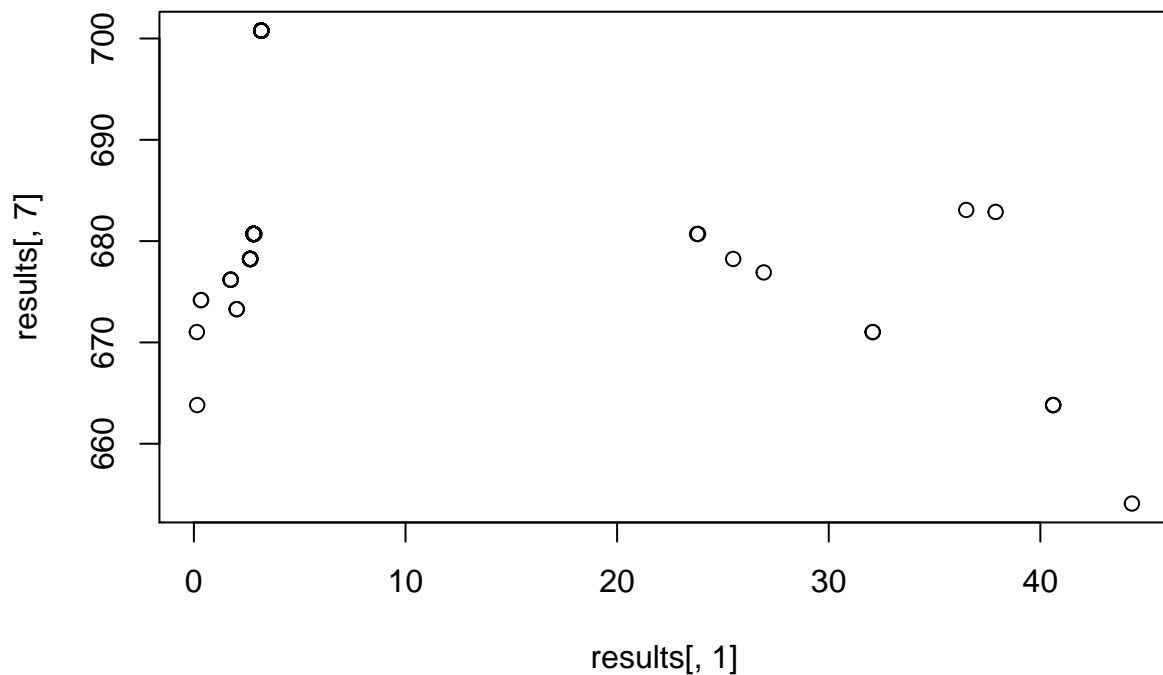


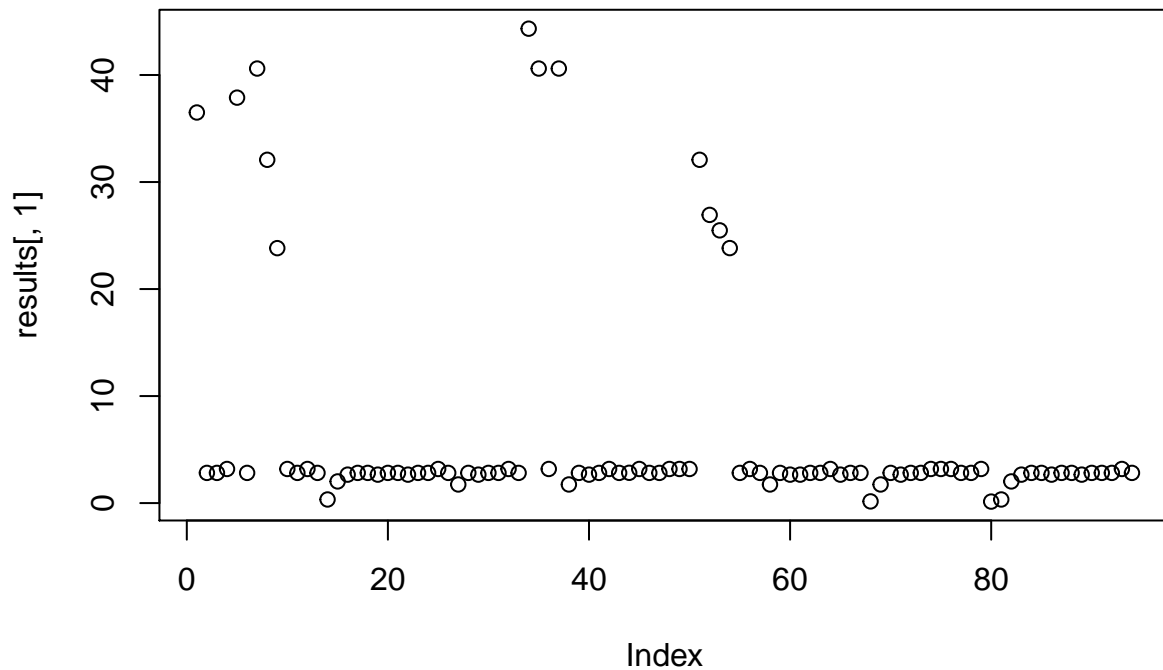


```
## 1 factor completed in 0.00414 minutes. Estimated time of completion: 2020-04-29 13:49:12 [1]
## [6] 4.497873e-06 3.000000e+01 6.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432860e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 40.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

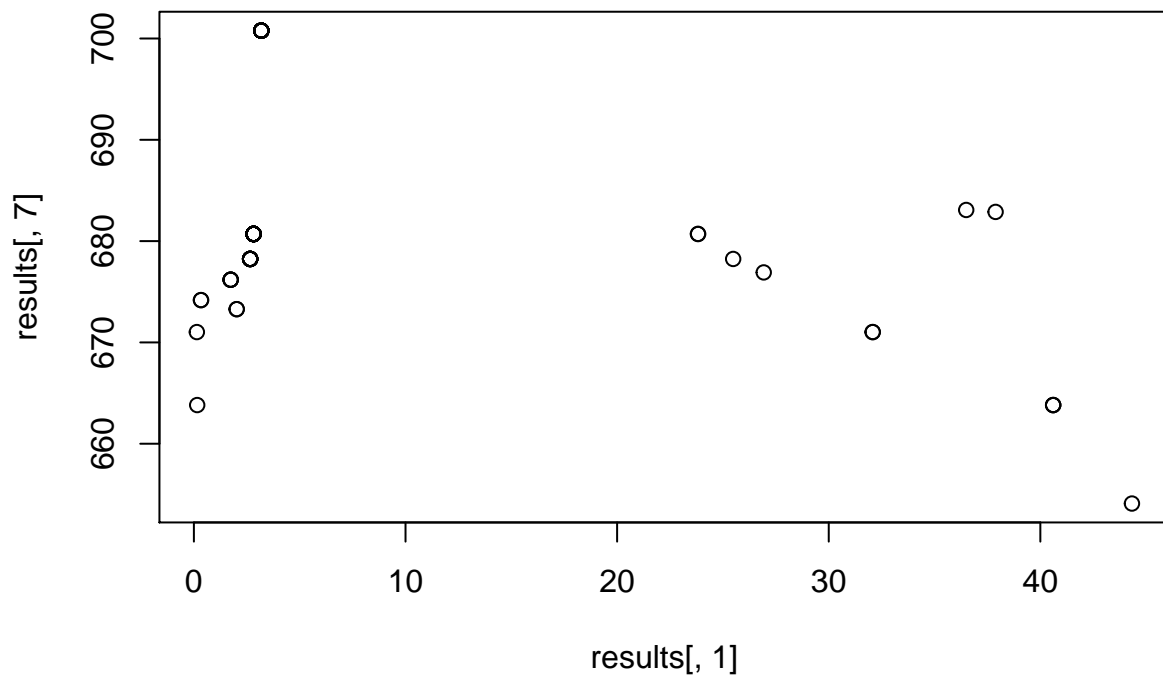


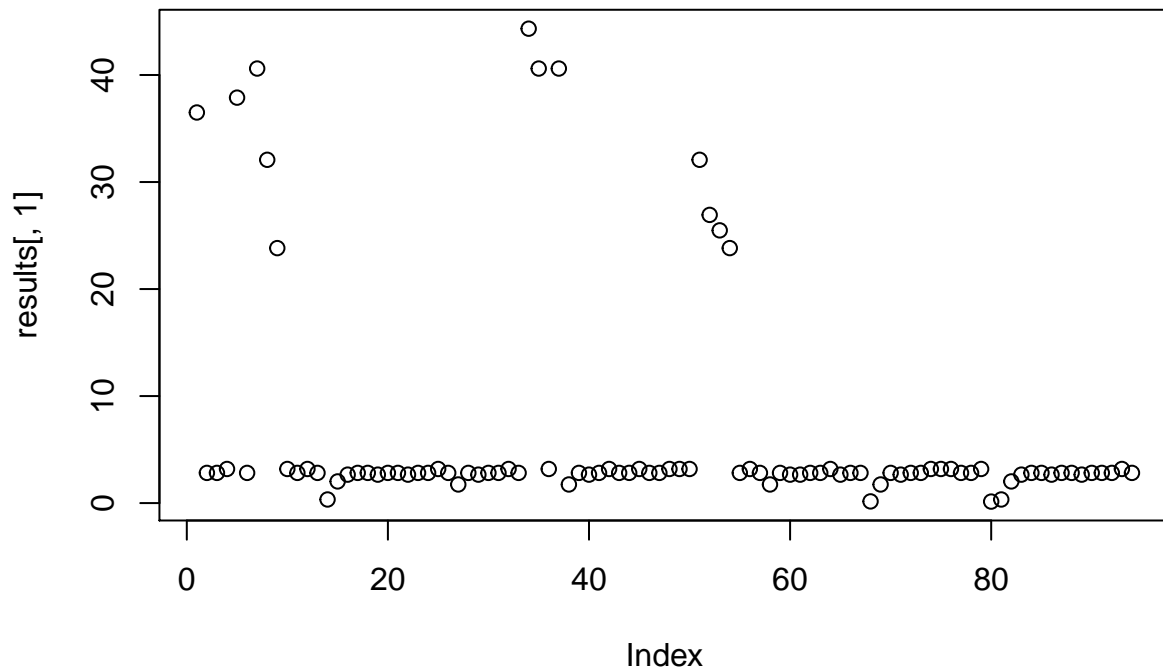


```
## 1 factor completed in 0.00384 minutes. Estimated time of completion: 2020-04-29 13:49:14 [1]
## [6] 1.896528e-08 4.000000e+01 6.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 50.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

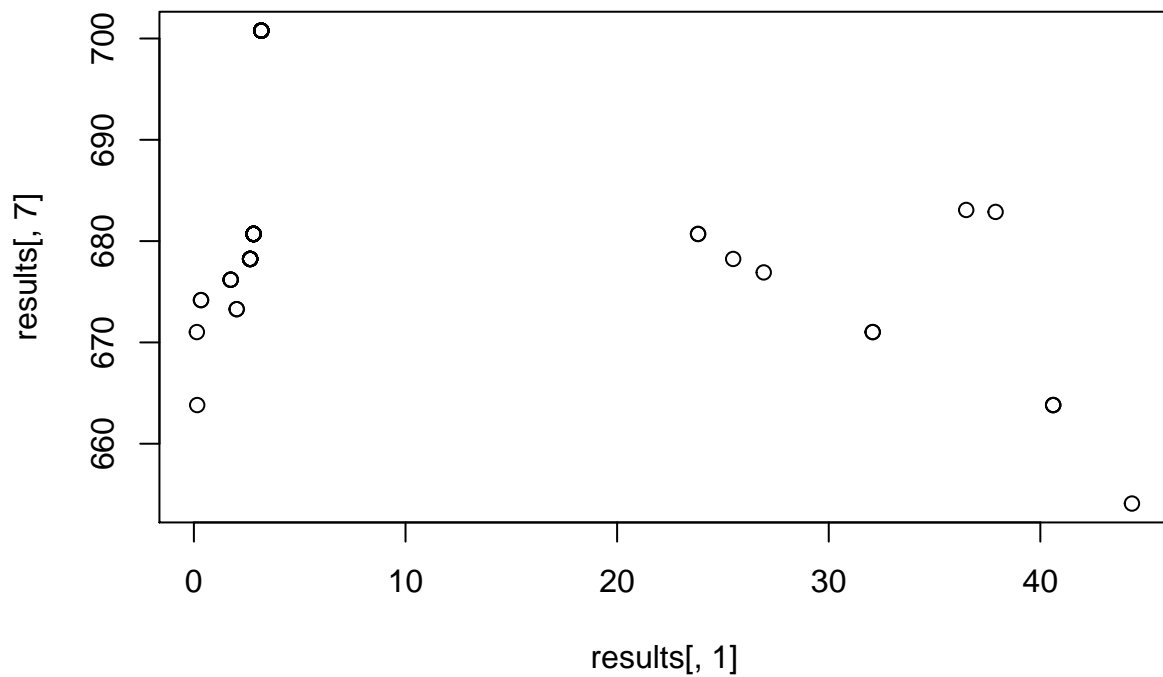


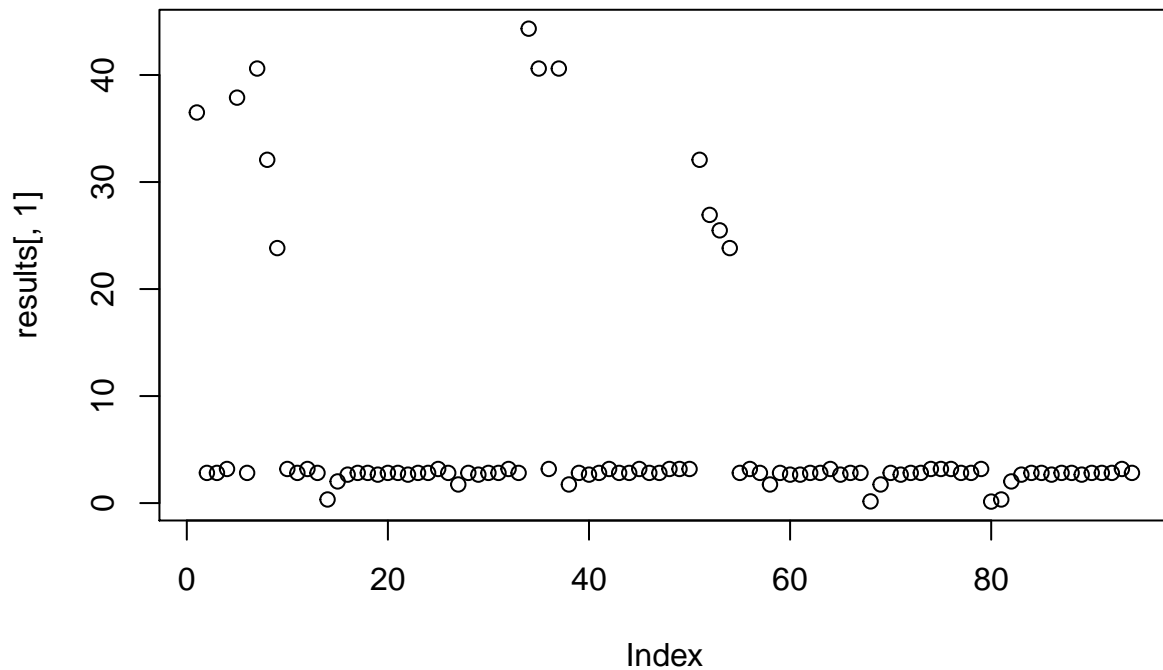


```
## 1 factor completed in 0.00383 minutes. Estimated time of completion: 2020-04-29 13:49:16 [1]
## [6] 7.996709e-11 5.000000e+01 6.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 60.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

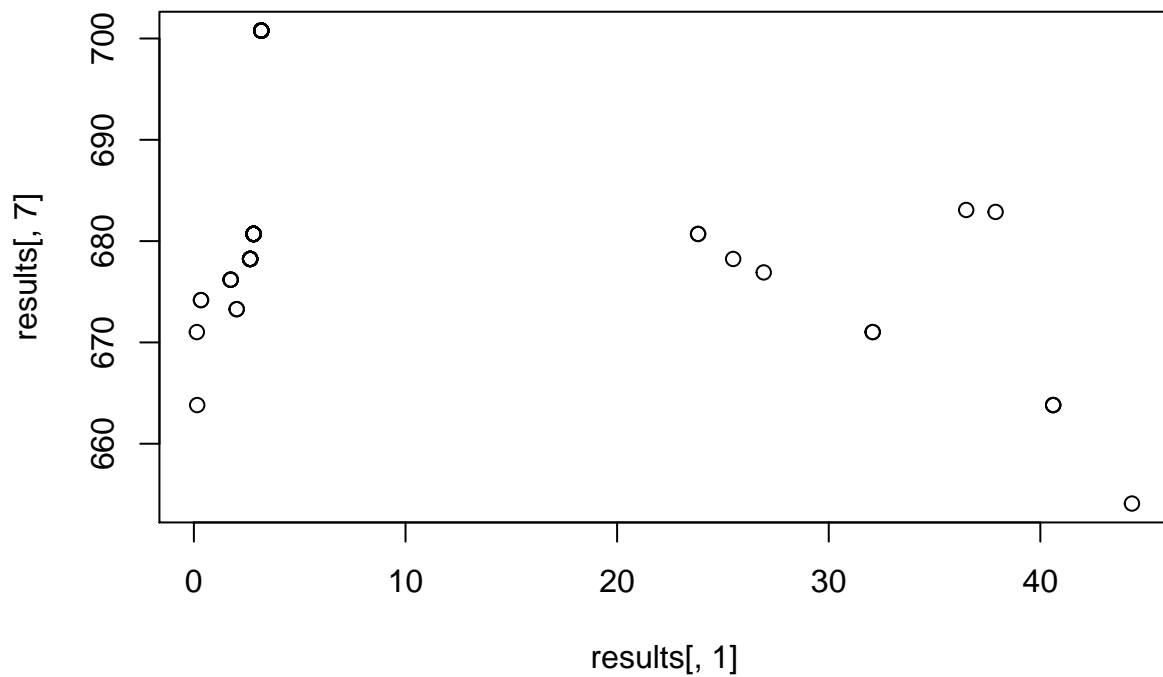


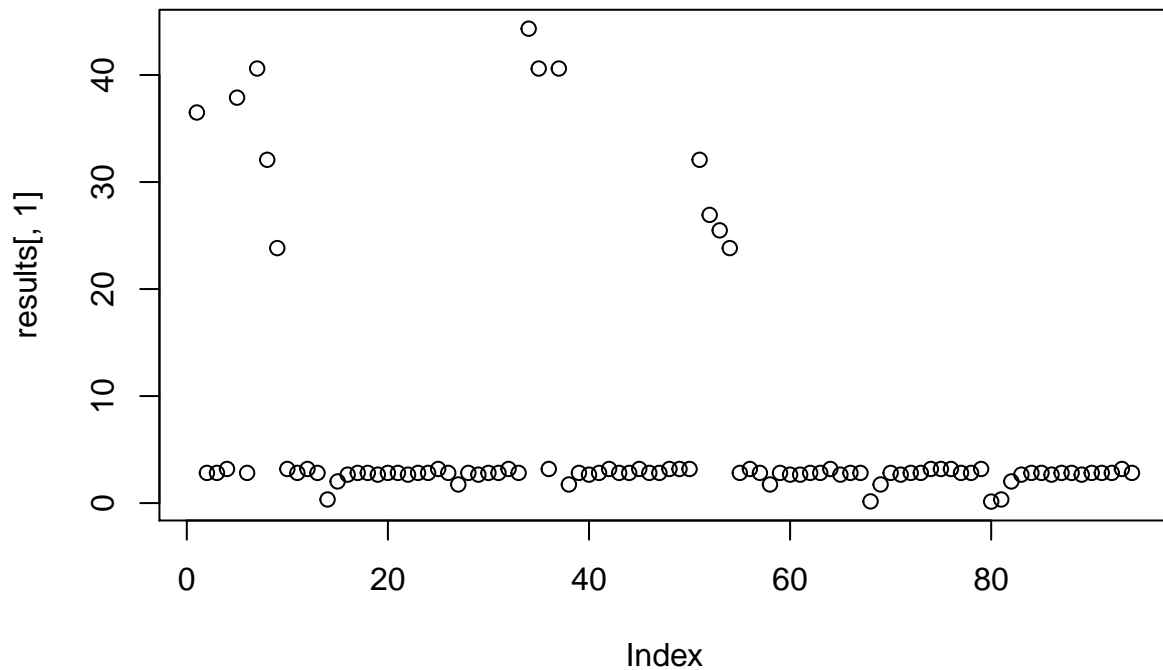


```
## 1 factor completed in 0.00382 minutes. Estimated time of completion: 2020-04-29 13:49:18 [1]
## [6] 3.371812e-13 6.000000e+01 6.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 70.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

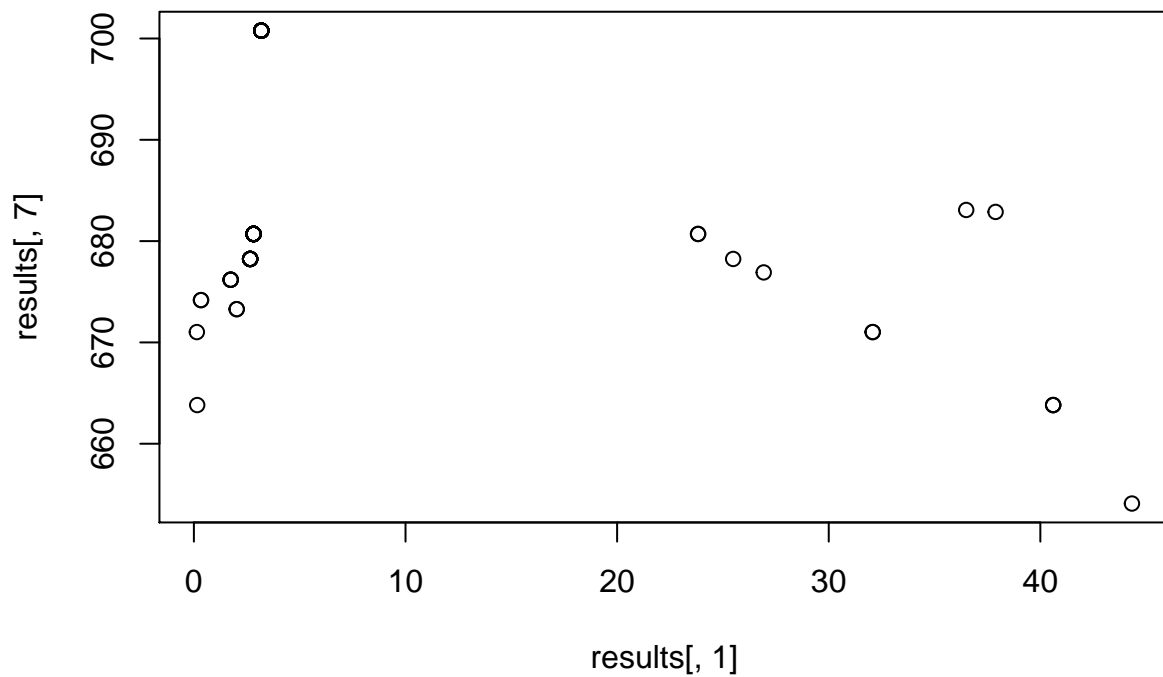


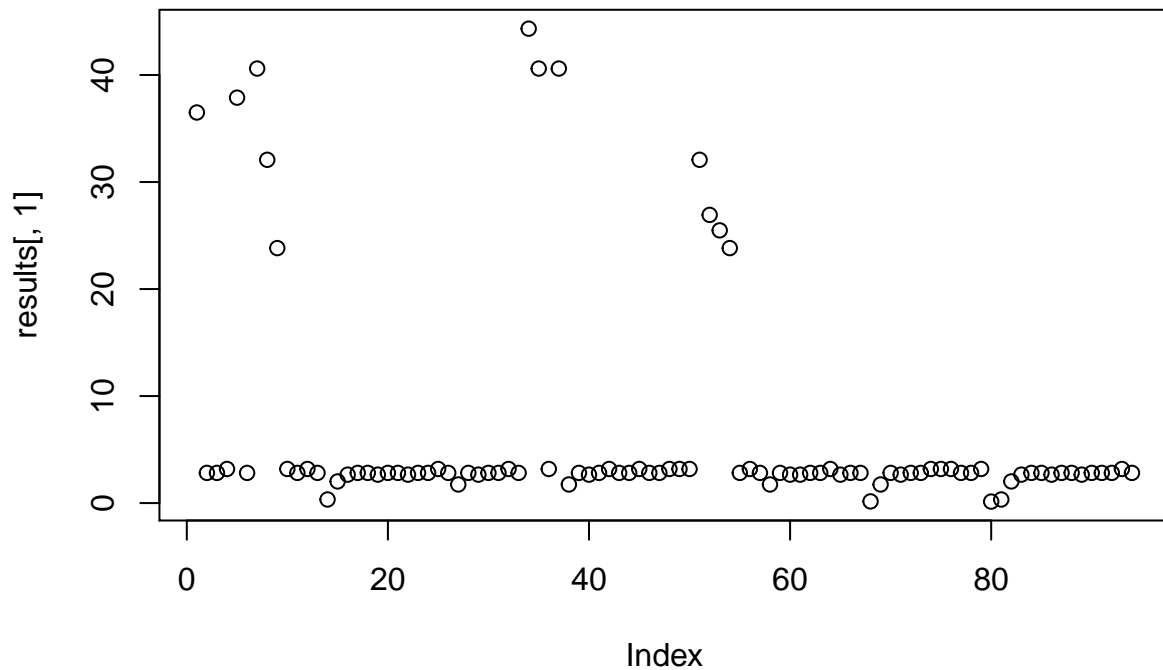


```
## 1 factor completed in 0.00395 minutes. Estimated time of completion: 2020-04-29 13:49:20 [1]
## [6] 1.421724e-15 7.000000e+01 6.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 80.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

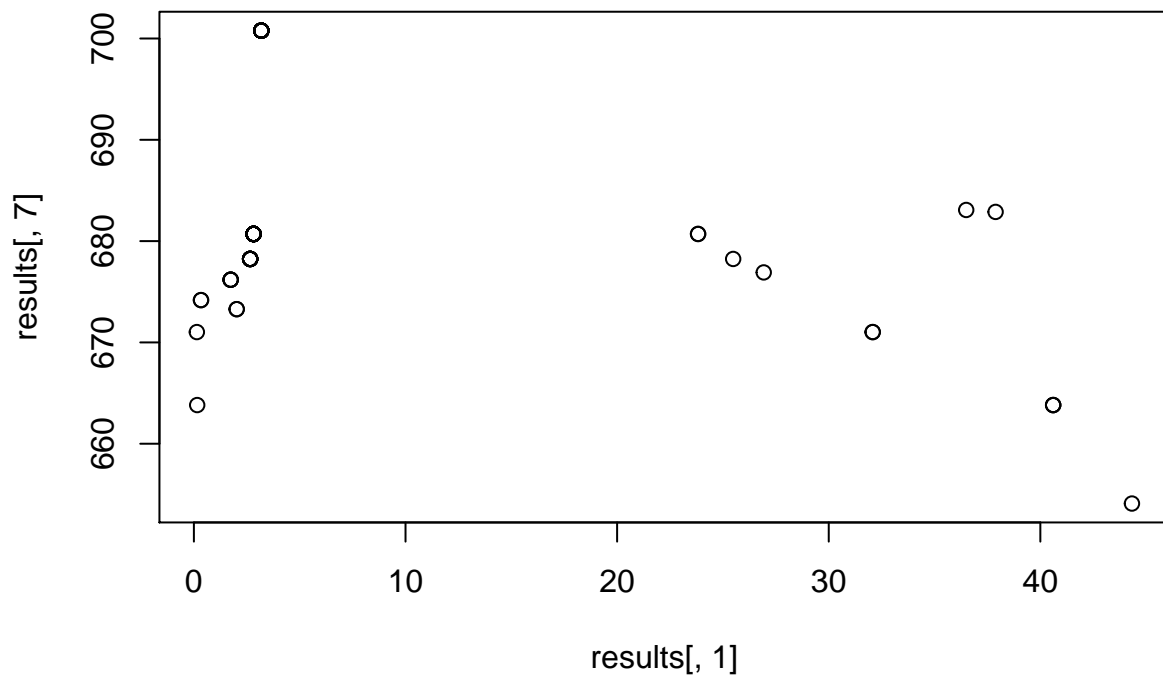


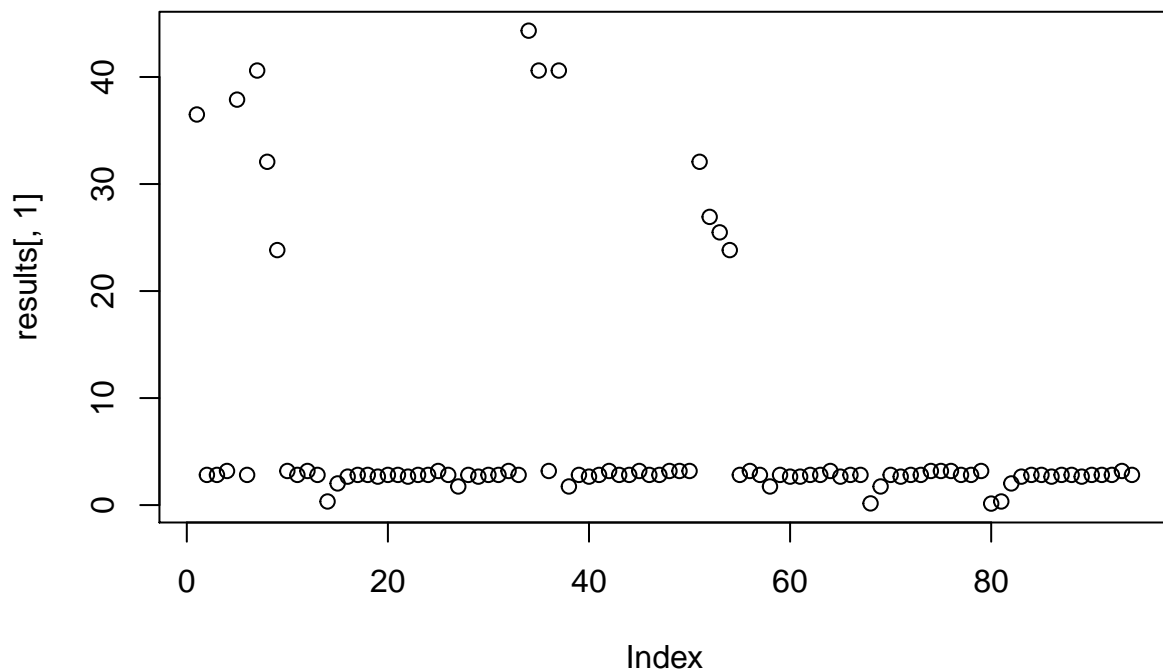


```
## 1 factor completed in 0.00382 minutes. Estimated time of completion: 2020-04-29 13:49:22 [1]
## [6] 5.994700e-18 8.000000e+01 6.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 90.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

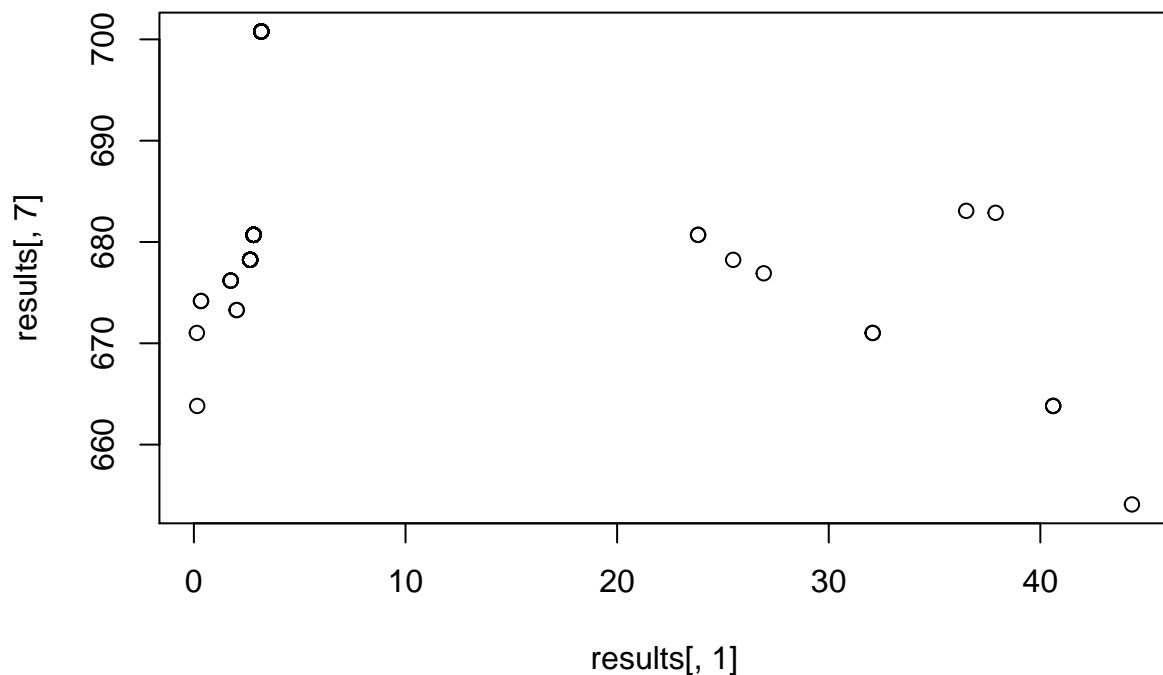


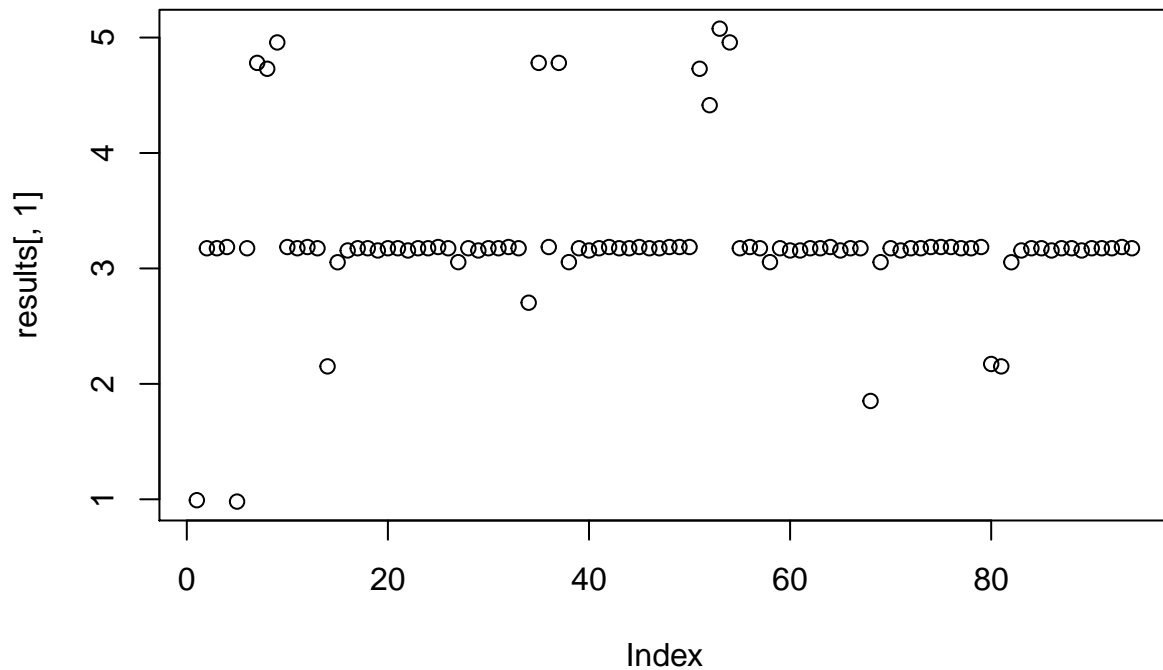


```
## 1 factor completed in 0.00385 minutes. Estimated time of completion: 2020-04-29 13:49:24 [1]
## [6] 2.527665e-20 9.000000e+01 6.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "X = 70.000001"
## [1] "j = 1e-06"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

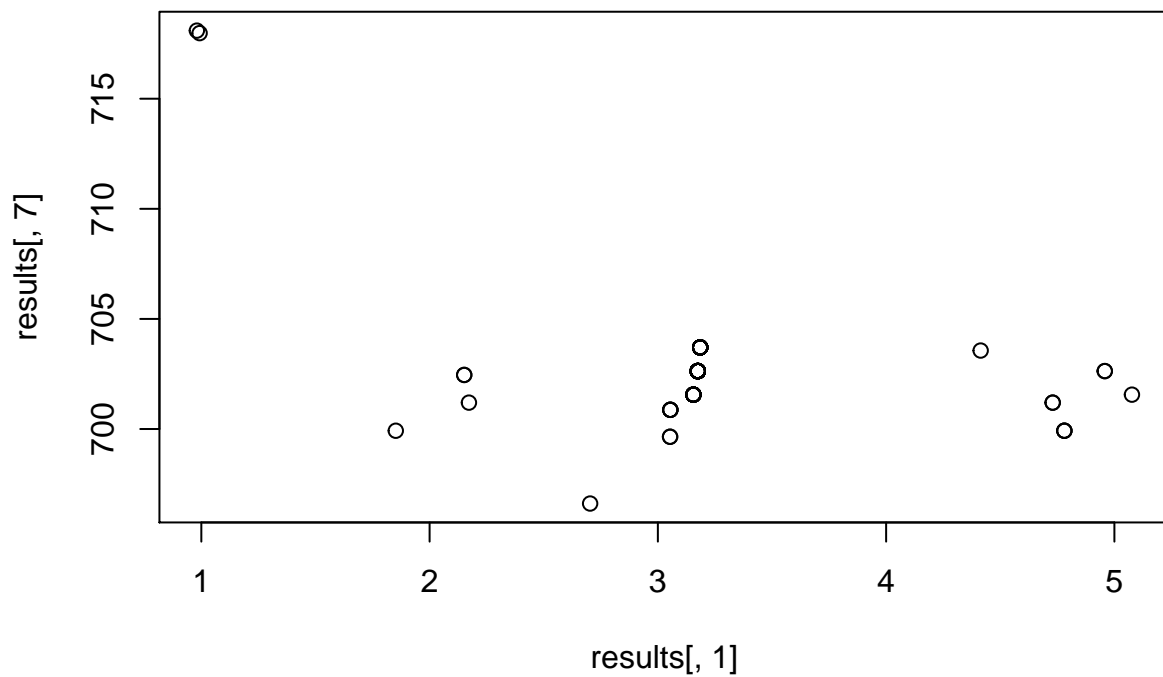


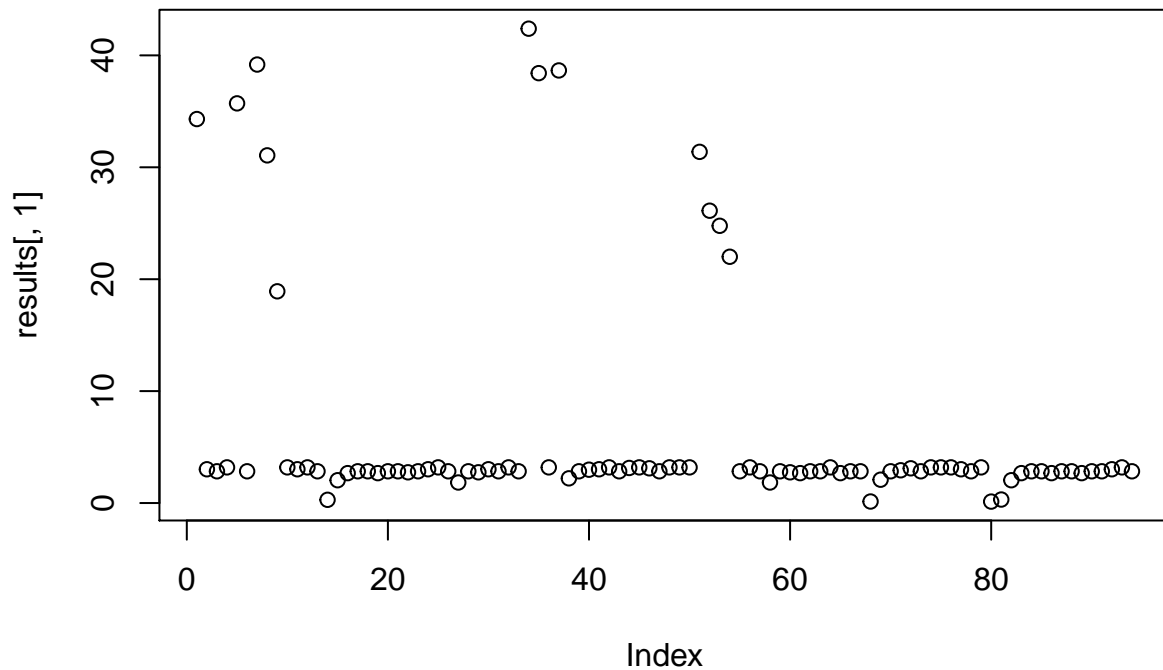


```
## 1 factor completed in 0.00386 minutes. Estimated time of completion: 2020-04-29 13:49:25 [1]
## [7] 0.0000010 70.0000010 7.8130568 718.0914213 2.7031879 696.6160156
## [13] 5.0000000 34.0000000
## [1] "j = 10.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

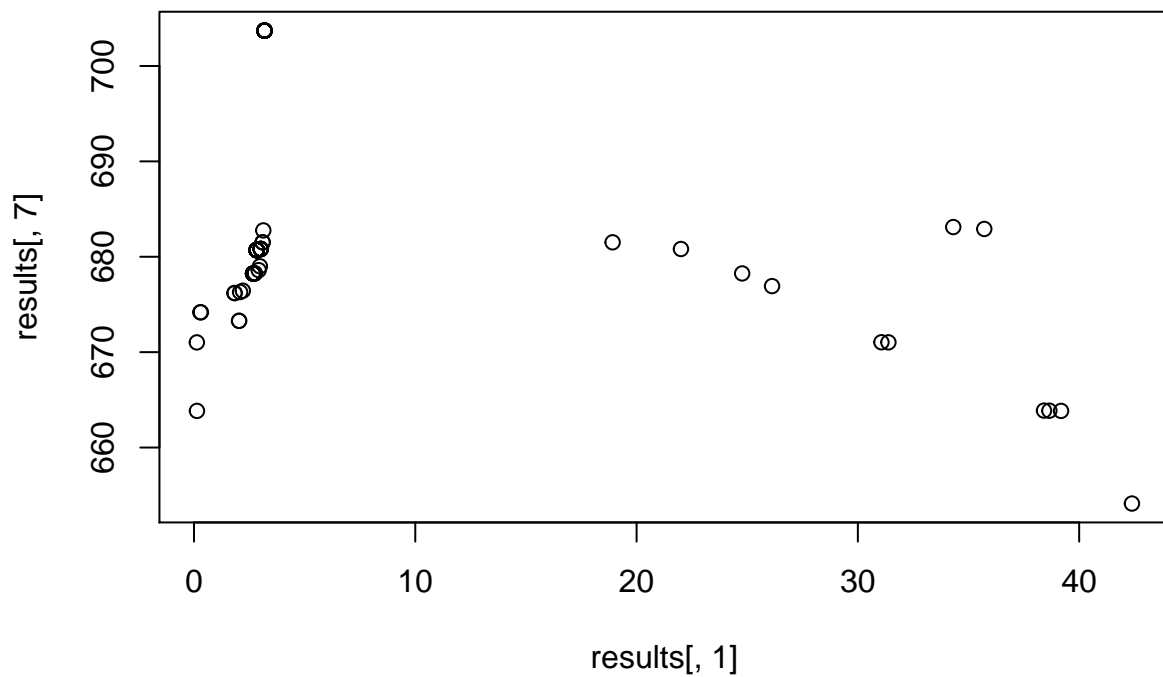


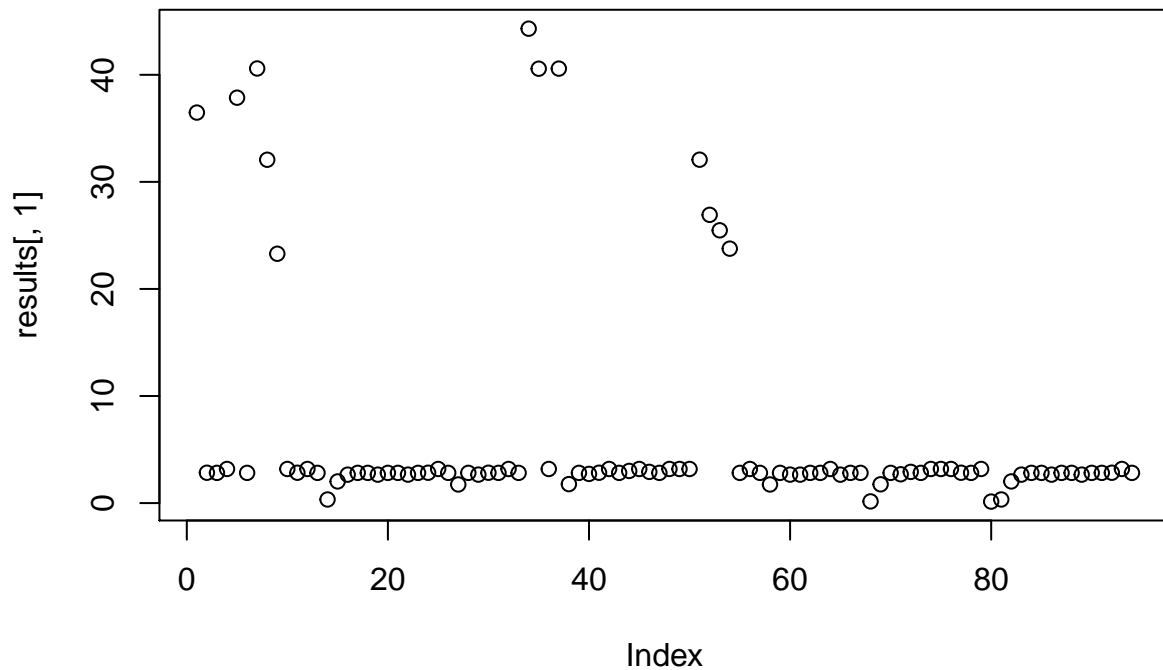


```
## 1 factor completed in 0.00383 minutes. Estimated time of completion: 2020-04-29 13:49:27 [1]
## [7] 10.0000010 70.0000010 7.8130568 671.0229954 42.3855888 654.1305230
## [13] 80.0000000 34.0000000
## [1] "j = 20.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

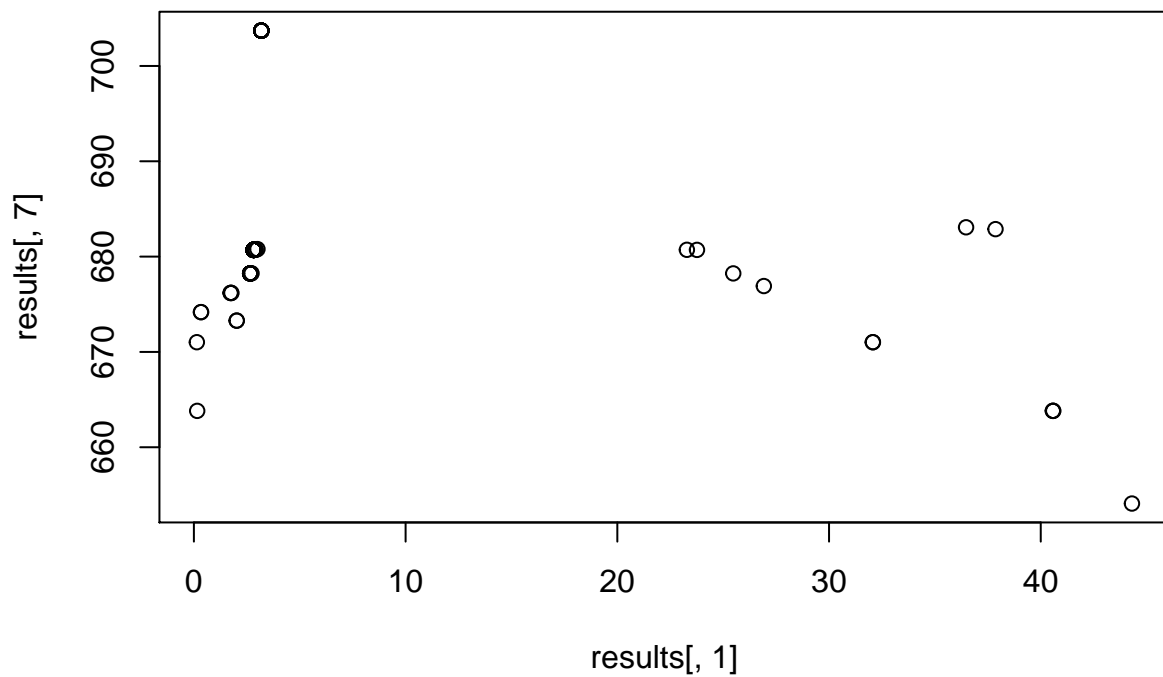


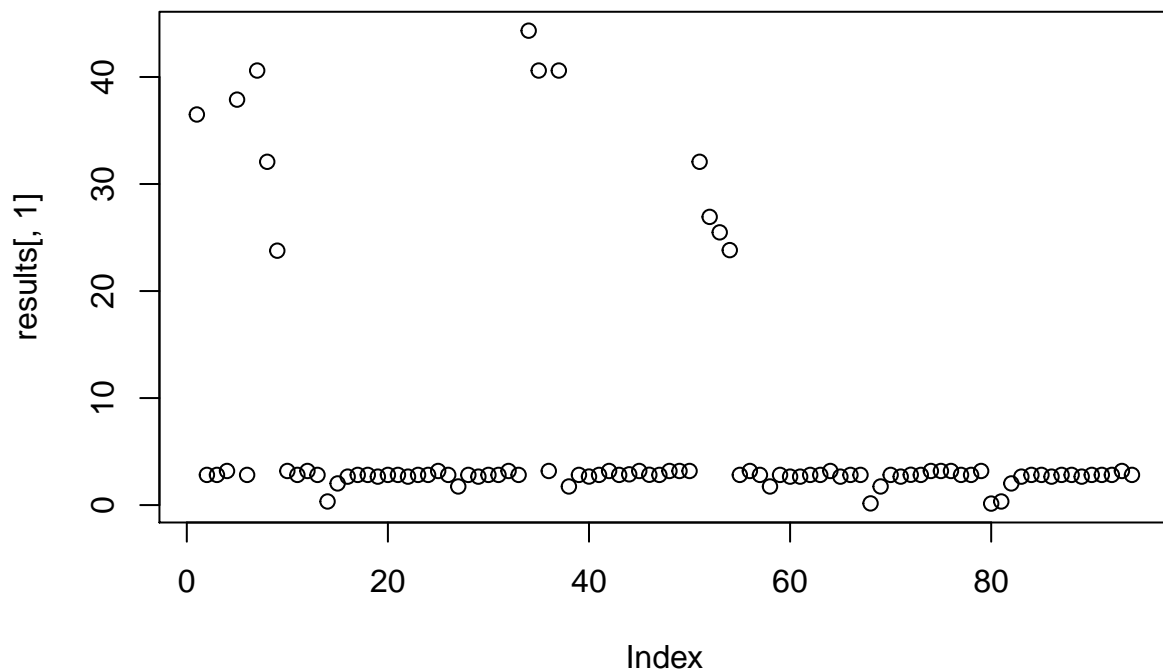


```
## 1 factor completed in 0.00393 minutes. Estimated time of completion: 2020-04-29 13:49:29 [1]
## [6] 0.00124452 20.00000100 70.00000100 7.81305685 671.01958688
## [11] 44.30772608 654.09568673 80.00000000 34.00000000
## [1] "j = 30.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

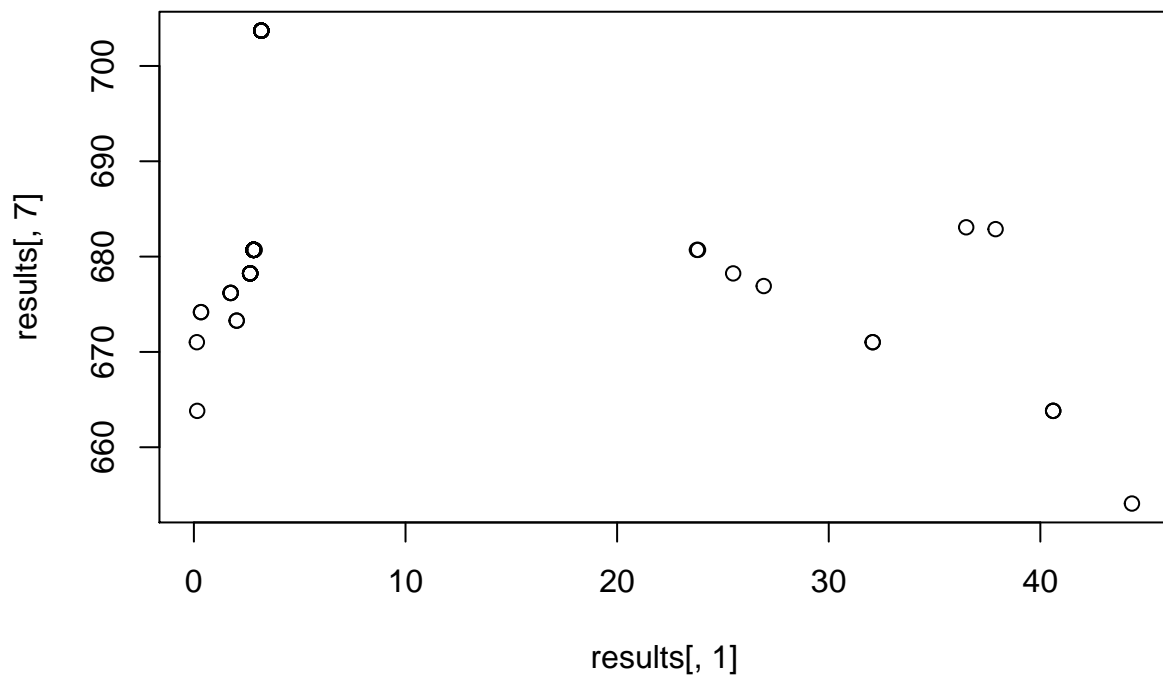


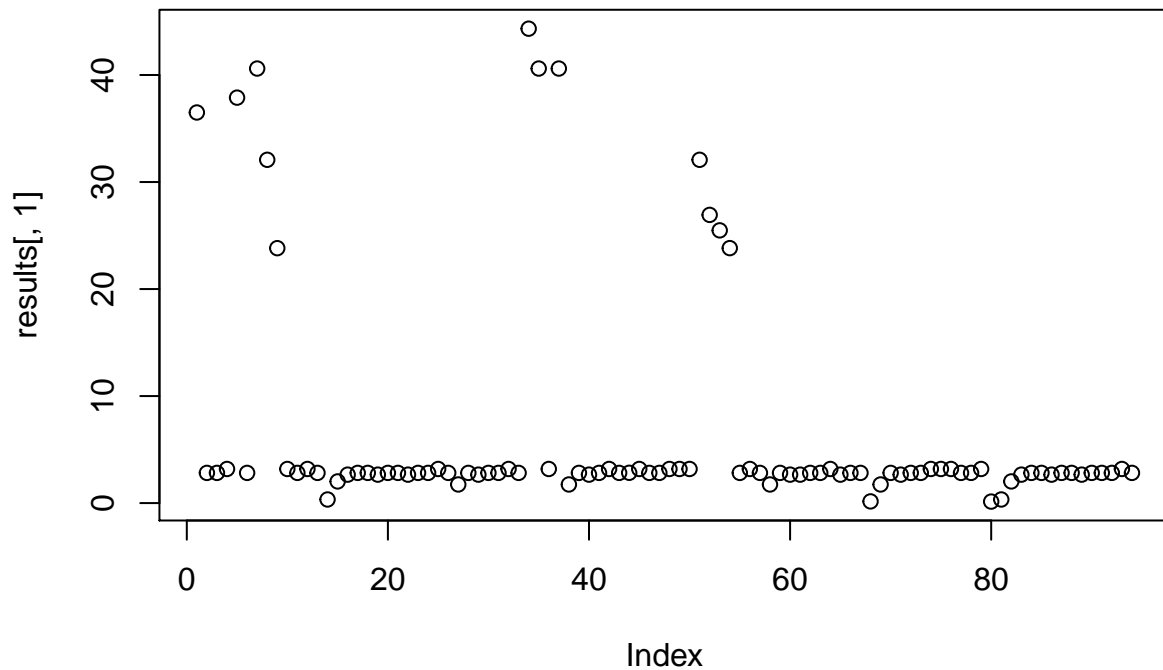


```
## 1 factor completed in 0.00385 minutes. Estimated time of completion: 2020-04-29 13:49:31 [1]
## [6] 5.247518e-06 3.000000e+01 7.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432857e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 40.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

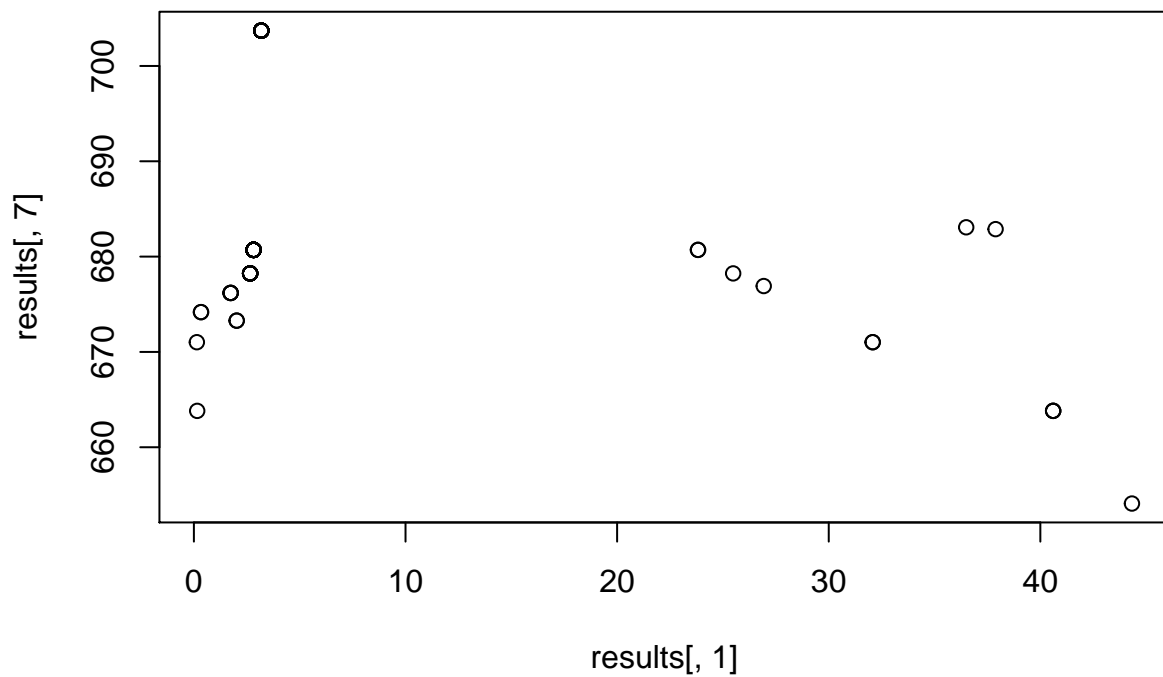


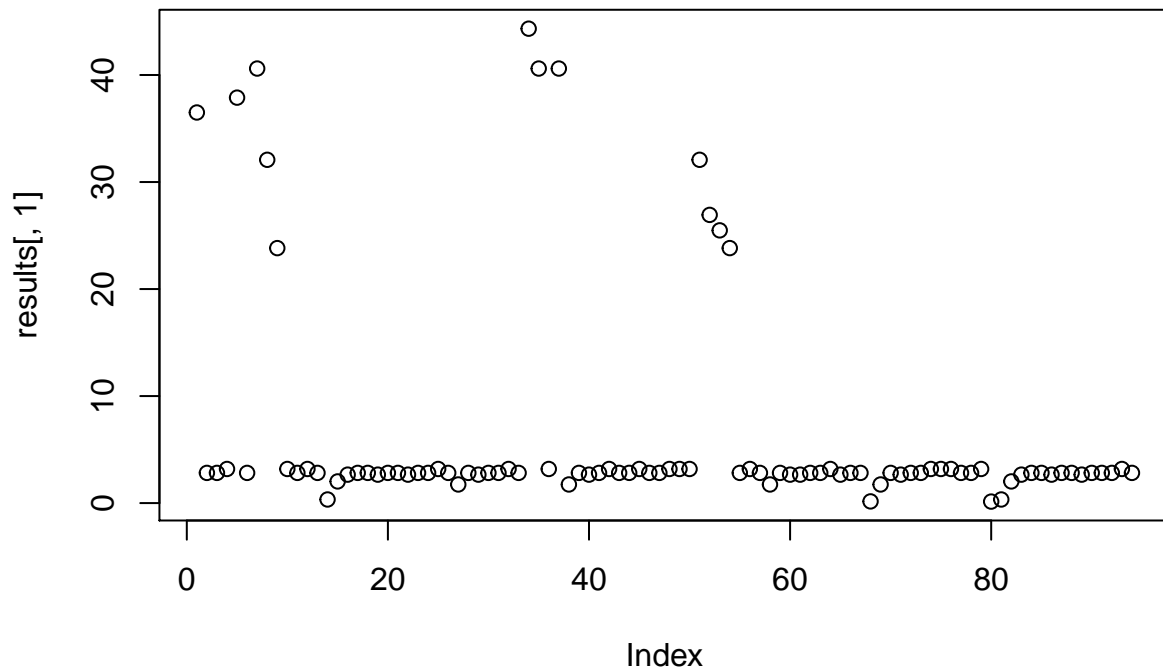


```
## 1 factor completed in 0.00374 minutes. Estimated time of completion: 2020-04-29 13:49:33 [1]
## [6] 2.212616e-08 4.000000e+01 7.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 50.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

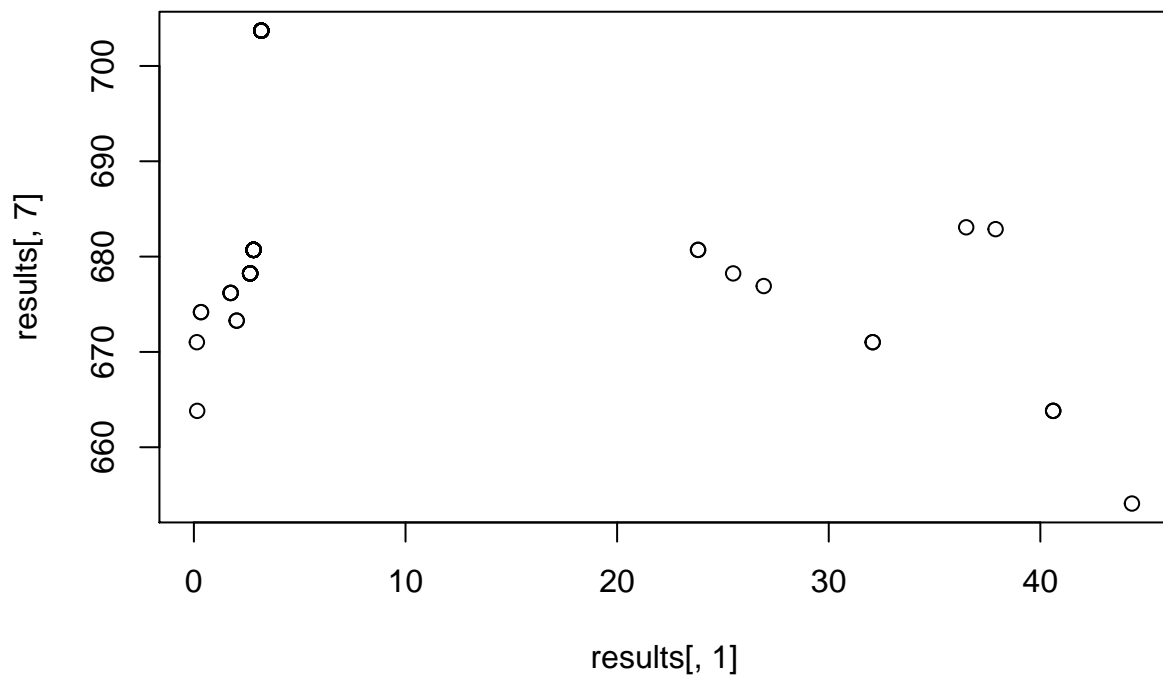


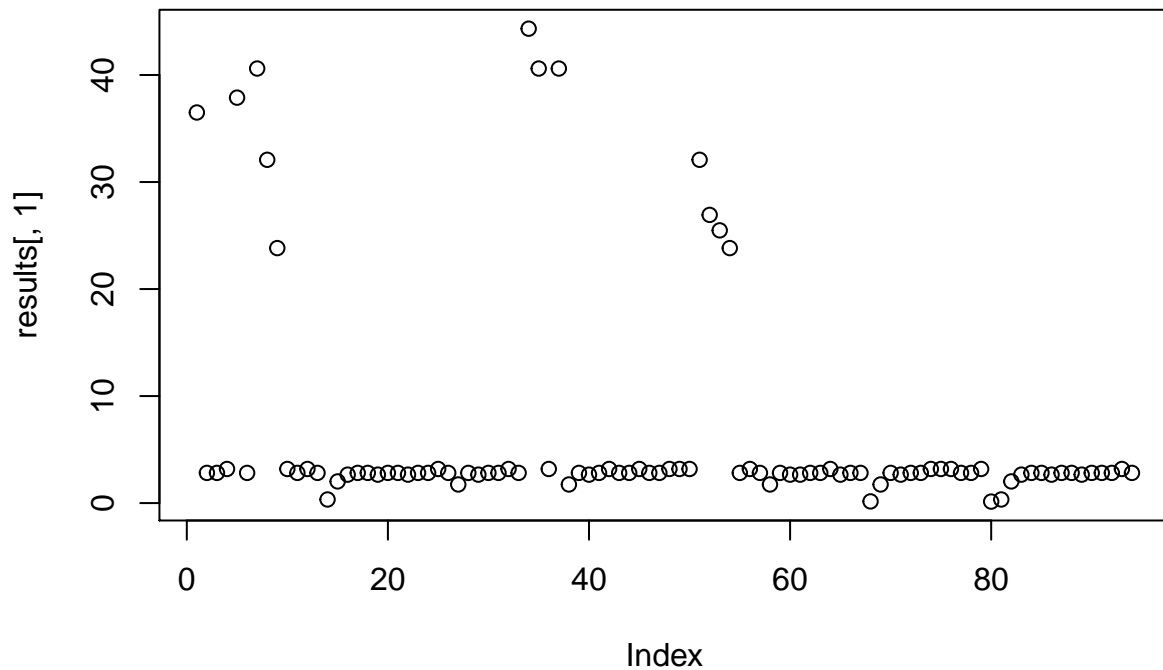


```
## 1 factor completed in 0.00385 minutes. Estimated time of completion: 2020-04-29 13:49:35 [1]
## [6] 9.329494e-11 5.000000e+01 7.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 60.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

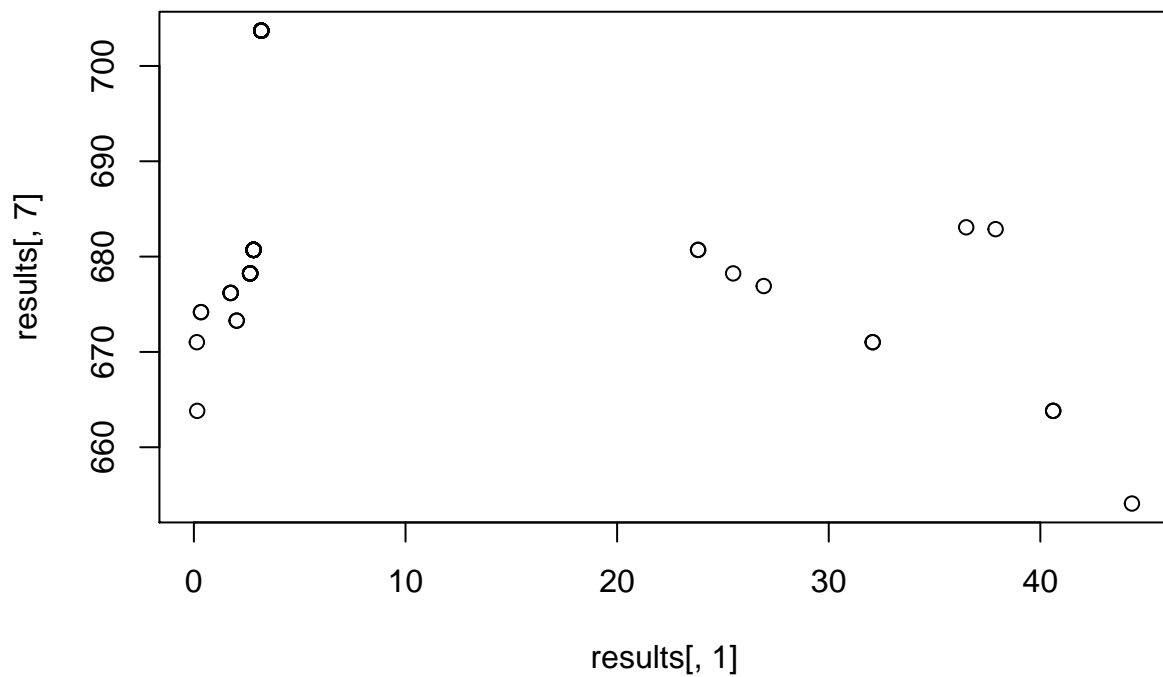


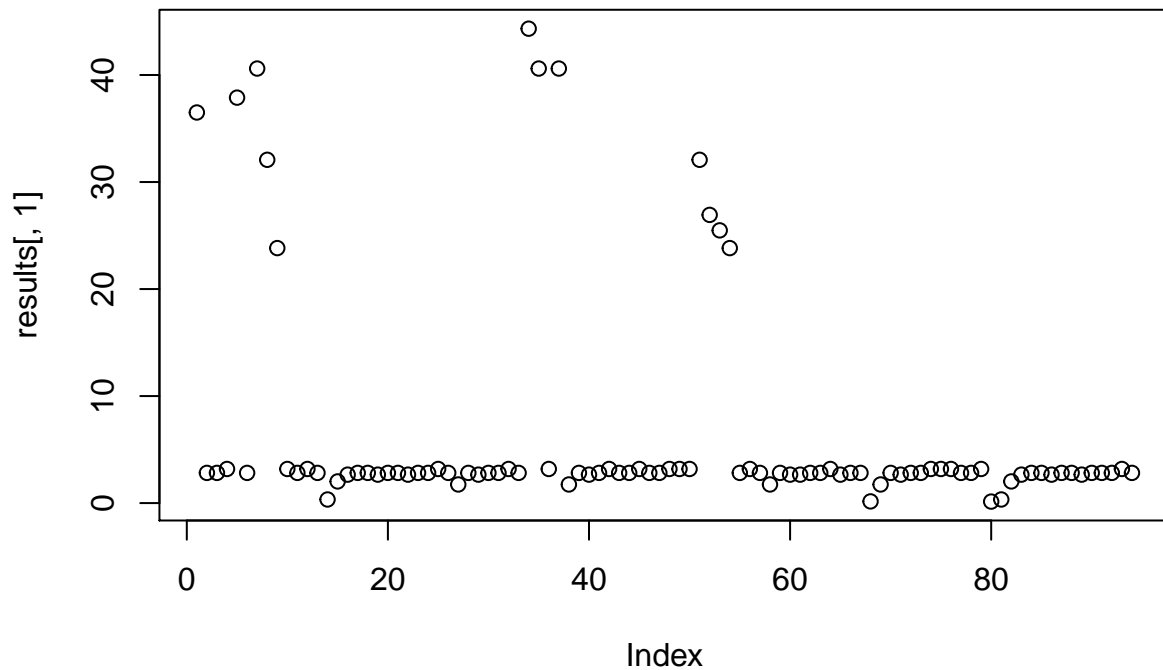


```
## 1 factor completed in 0.00383 minutes. Estimated time of completion: 2020-04-29 13:49:37 [1]
## [6] 3.933780e-13 6.000000e+01 7.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 70.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

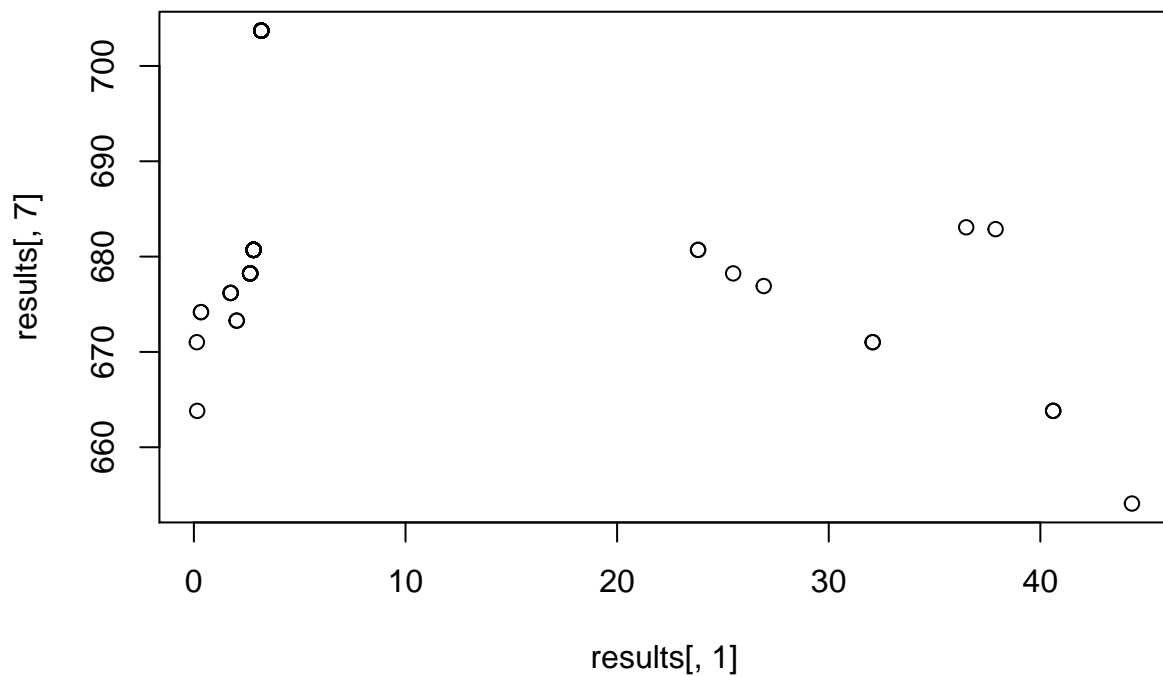


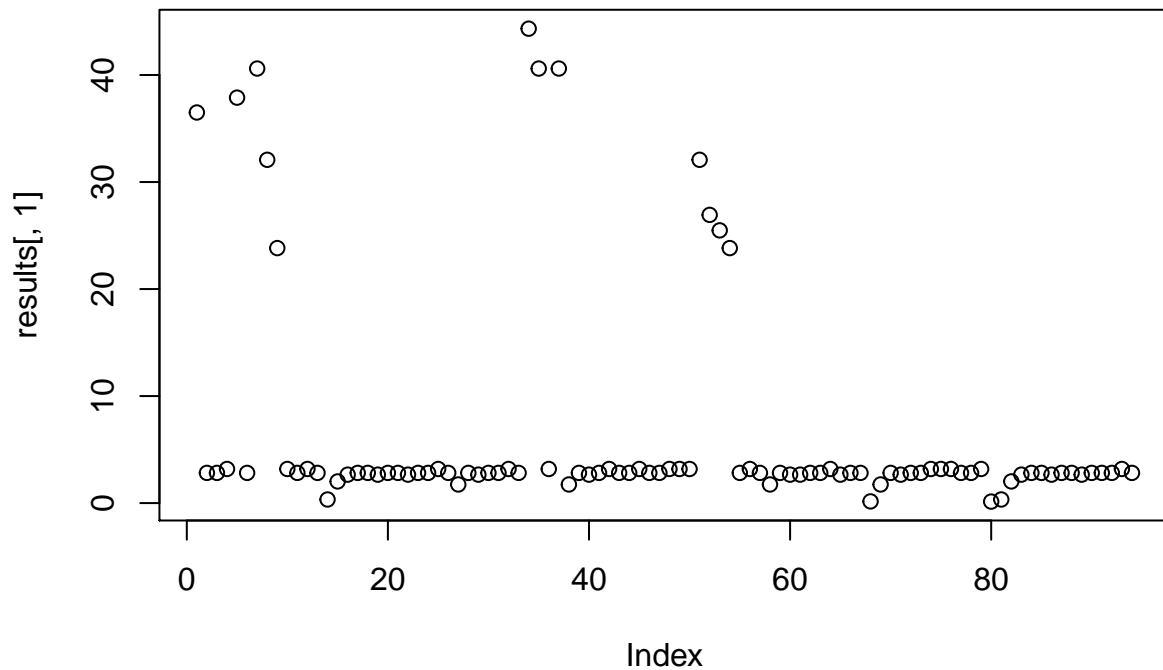


```
## 1 factor completed in 0.00389 minutes. Estimated time of completion: 2020-04-29 13:49:39 [1]
## [6] 1.658678e-15 7.000000e+01 7.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 80.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

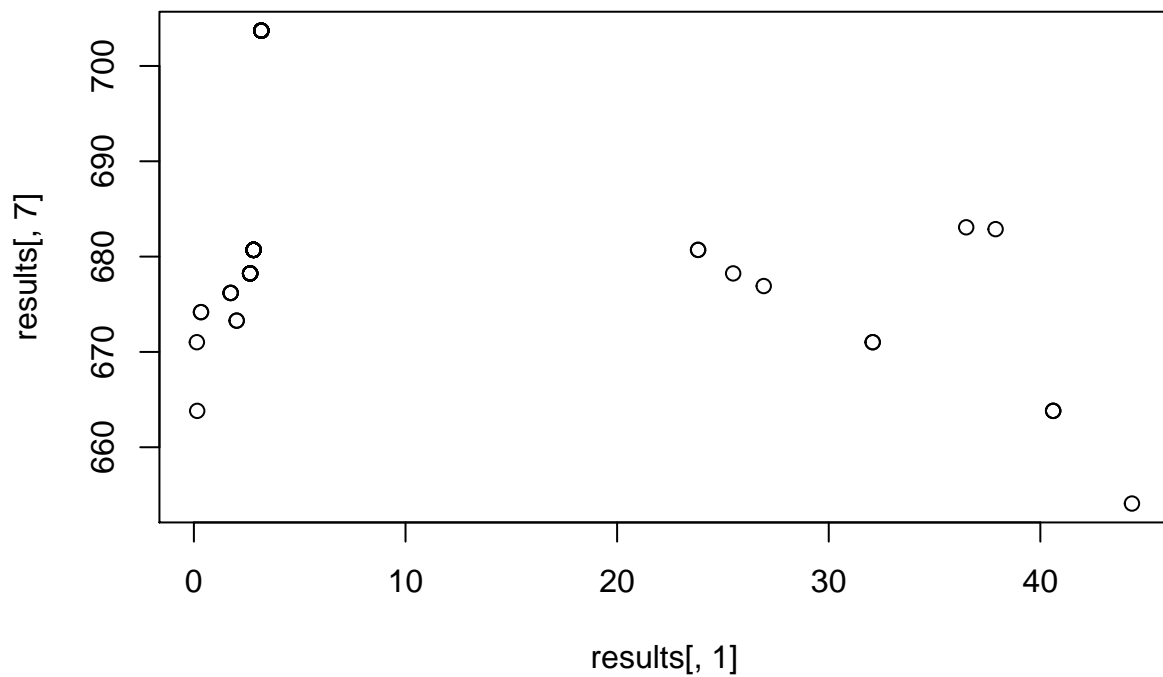


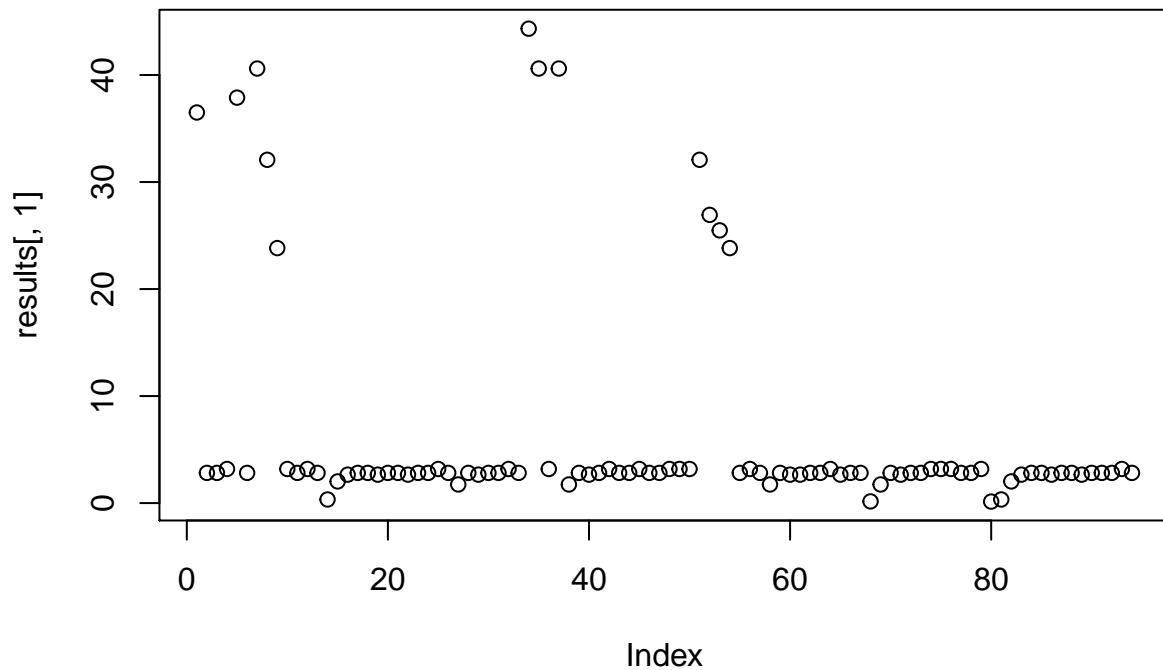


```
## 1 factor completed in 0.00391 minutes. Estimated time of completion: 2020-04-29 13:49:41 [1]
## [6] 6.993816e-18 8.000000e+01 7.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 90.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

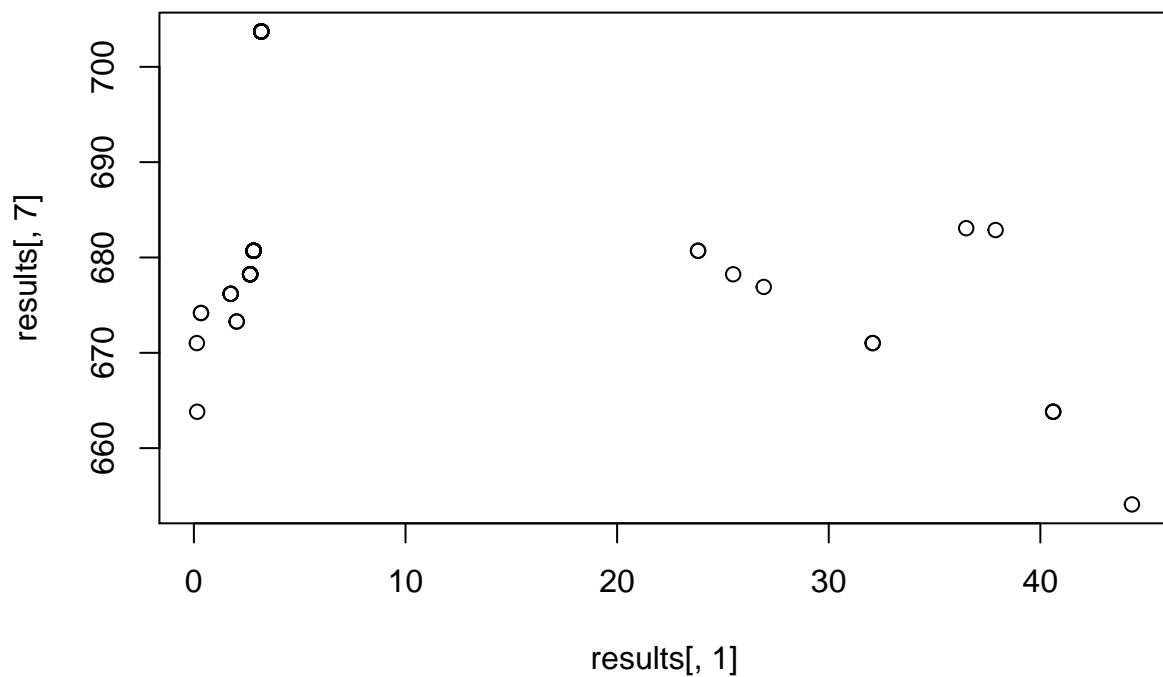


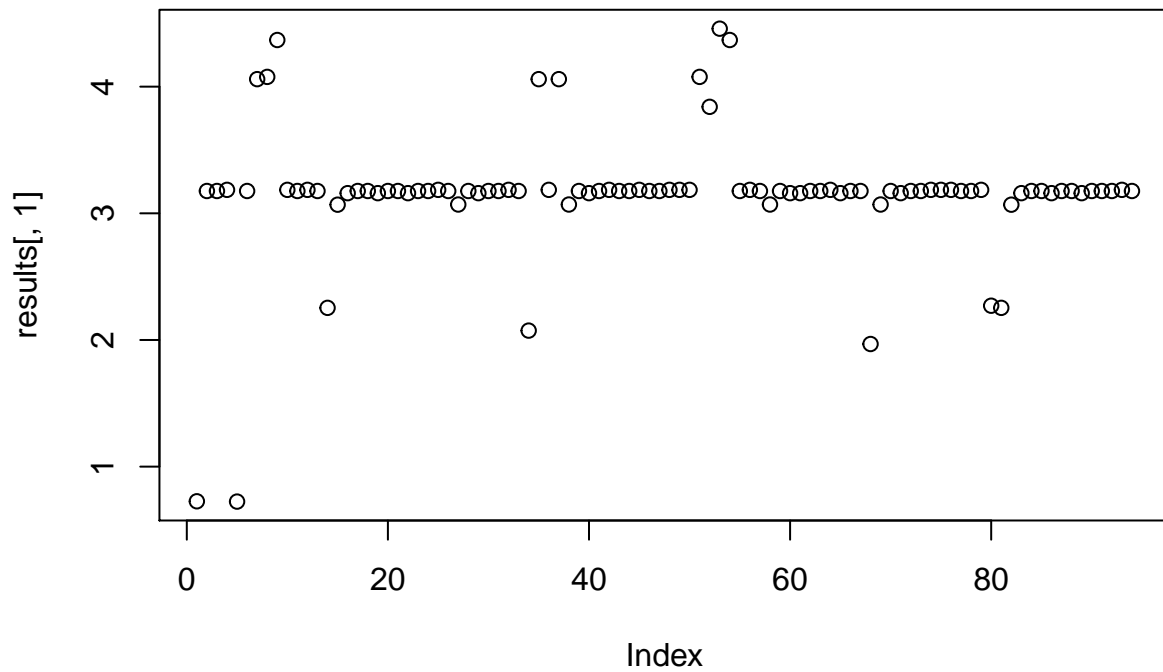


```
## 1 factor completed in 0.00382 minutes. Estimated time of completion: 2020-04-29 13:49:43 [1]
## [6] 2.948942e-20 9.000000e+01 7.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "X = 80.000001"
## [1] "j = 1e-06"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

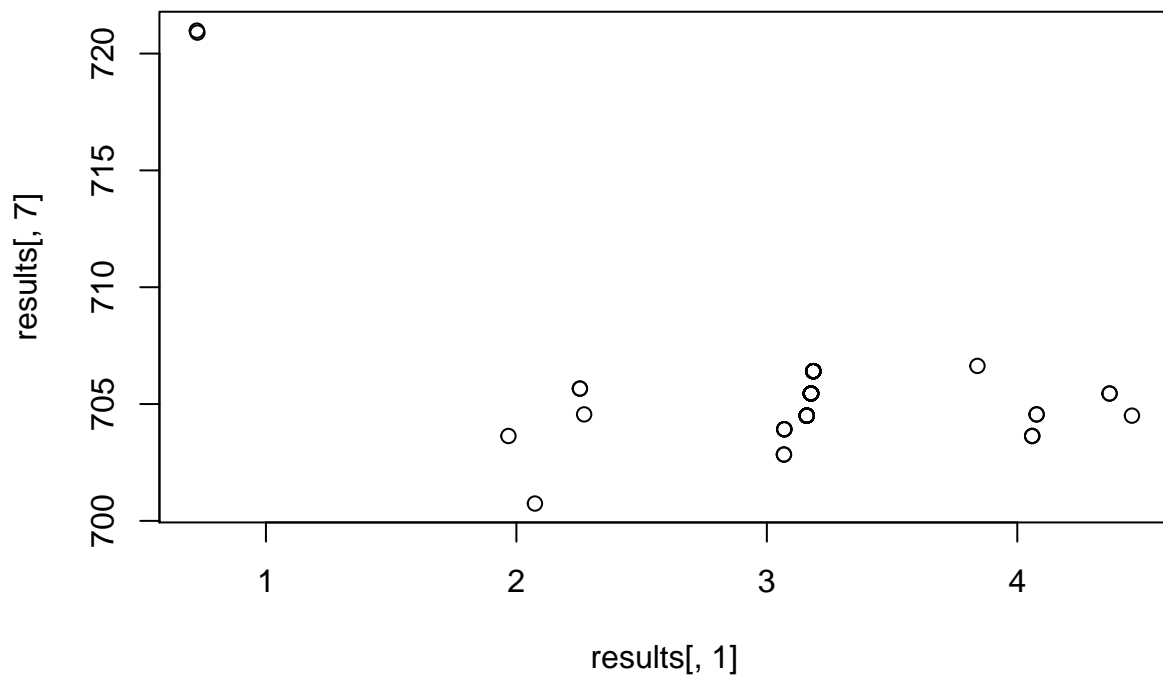


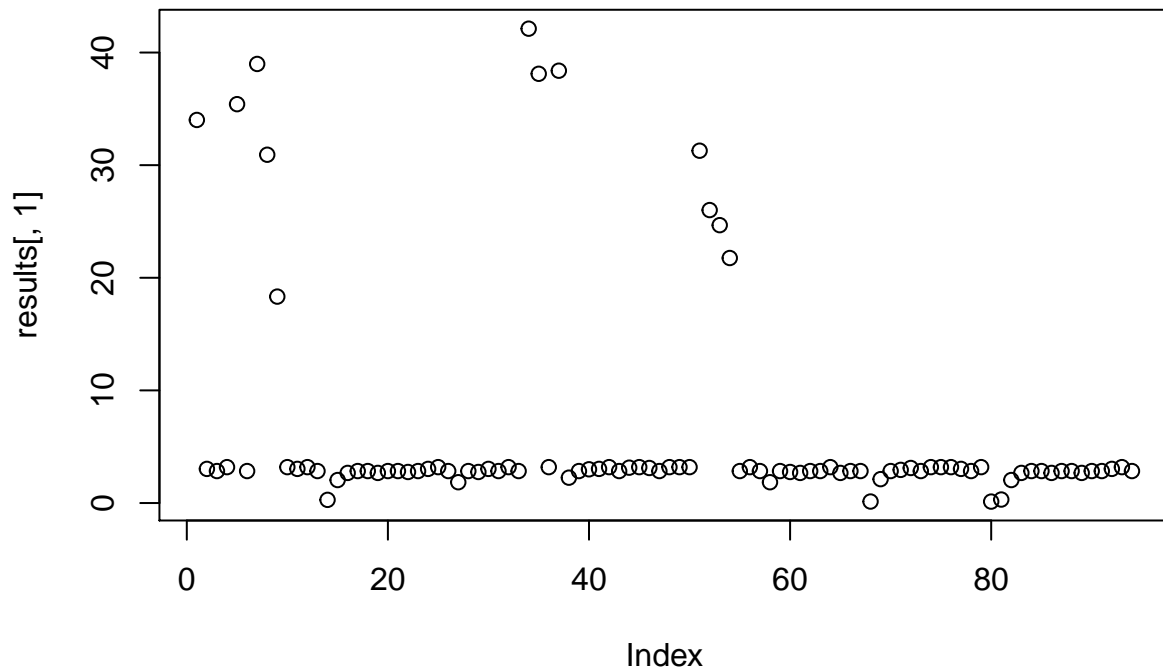


```
## 1 factor completed in 0.00388 minutes. Estimated time of completion: 2020-04-29 13:49:45 [1]
## [7] 0.0000010 80.0000010 7.8130568 720.9816454 2.0739924 700.7426895
## [13] 5.0000000 34.0000000
## [1] "j = 10.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

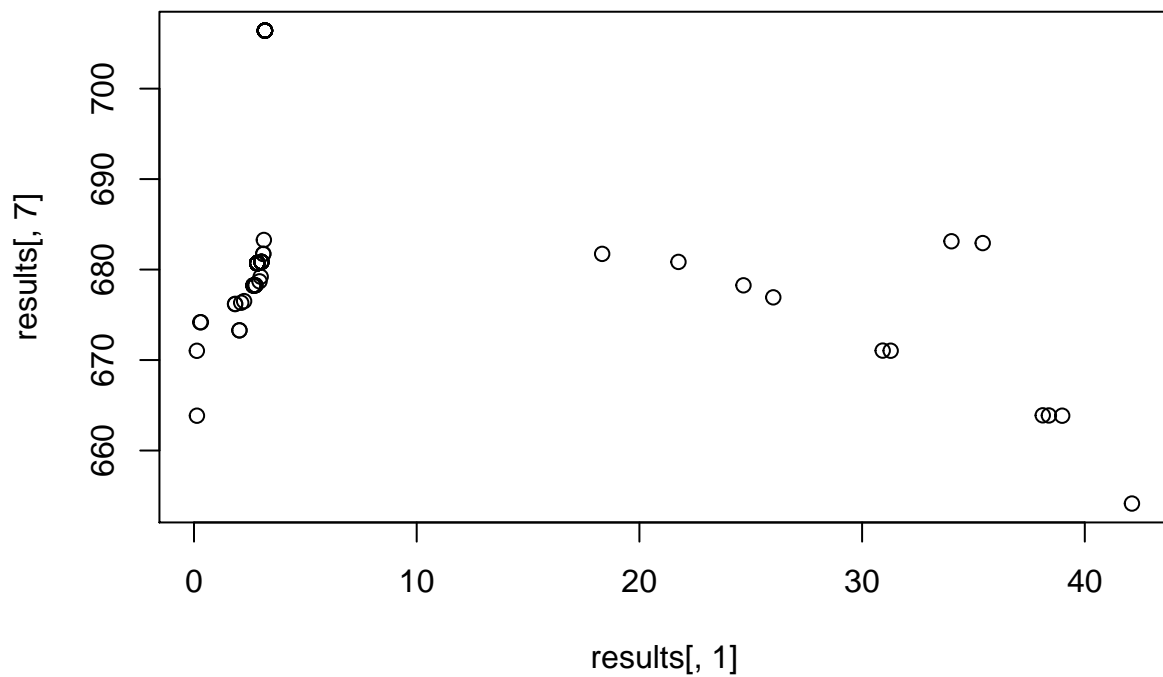


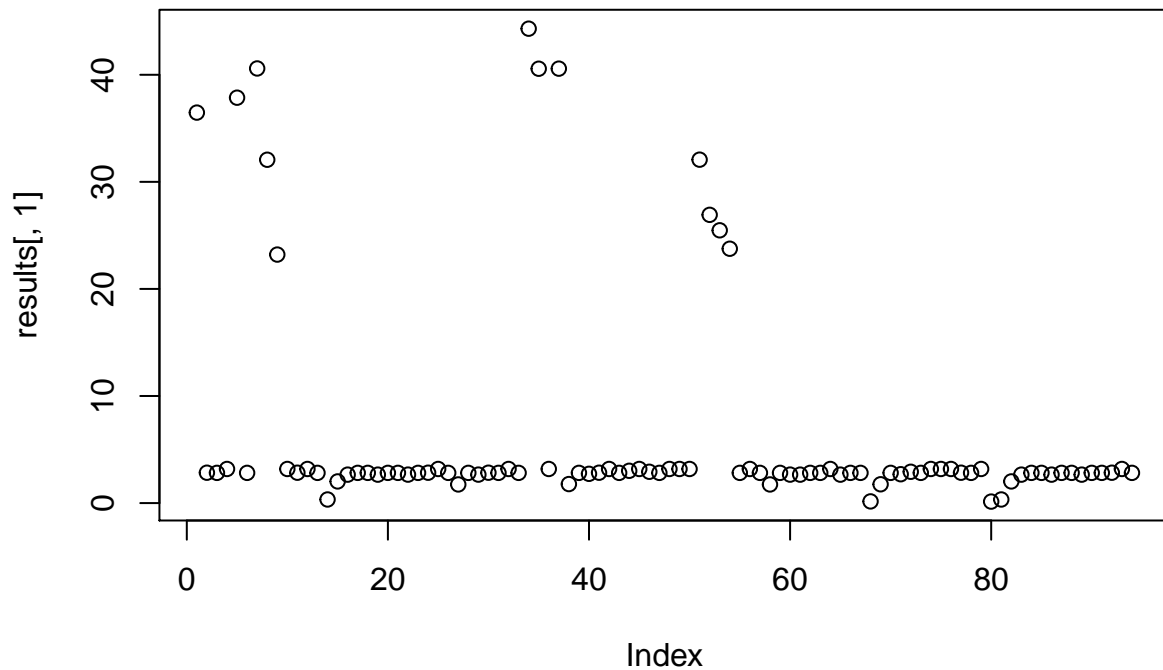


```
## 1 factor completed in 0.00384 minutes. Estimated time of completion: 2020-04-29 13:49:47 [1]
## [7] 10.0000010 80.0000010 7.8130568 671.0240291 42.1173445 654.1409498
## [13] 80.0000000 34.0000000
## [1] "j = 20.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

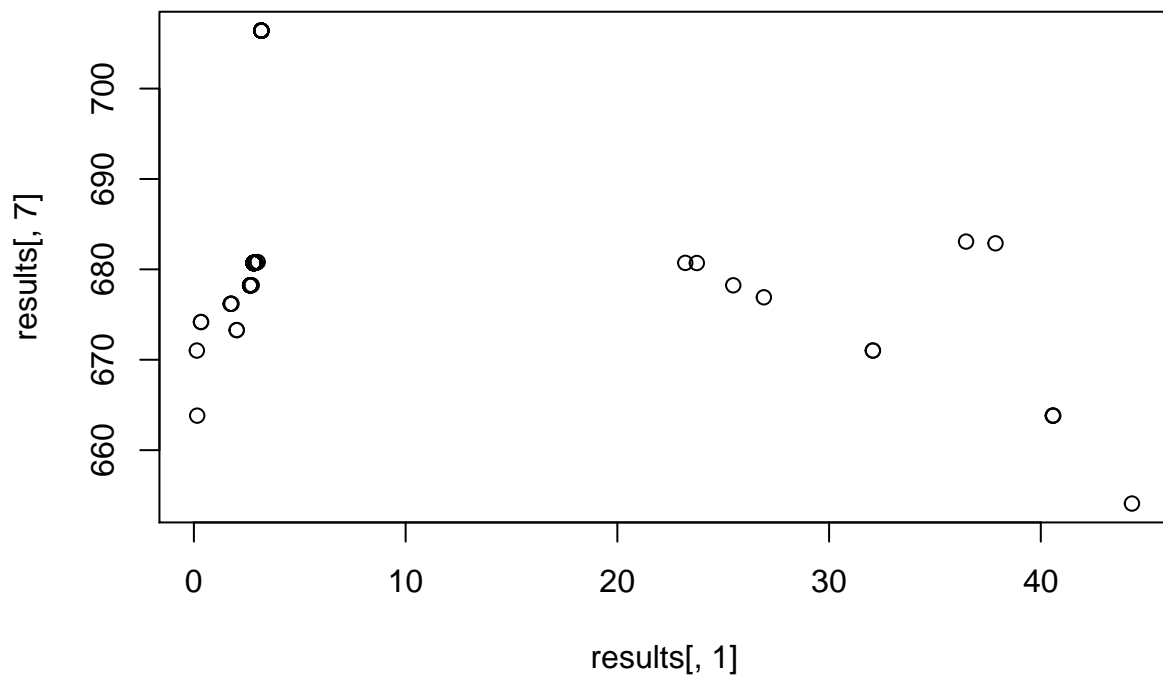


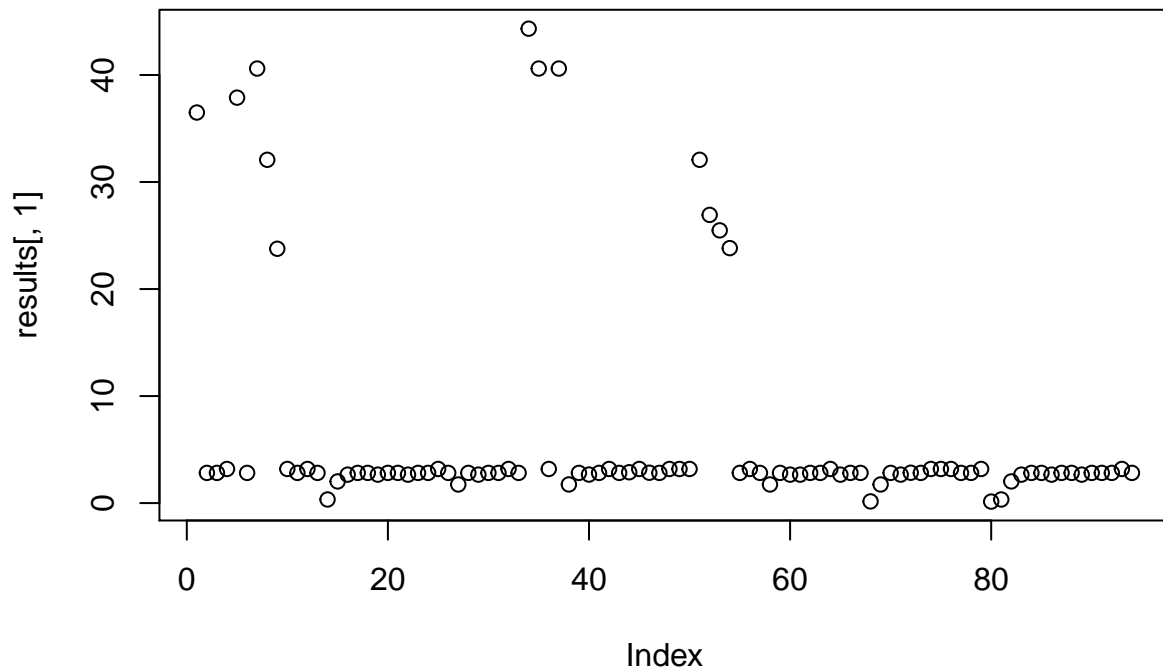


```
## 1 factor completed in 0.00412 minutes. Estimated time of completion: 2020-04-29 13:49:50 [1]
## [6] 0.001422309 20.000001000 80.000001000 7.813056846 671.019586897
## [11] 44.304717883 654.095687958 80.000000000 34.000000000
## [1] "j = 30.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

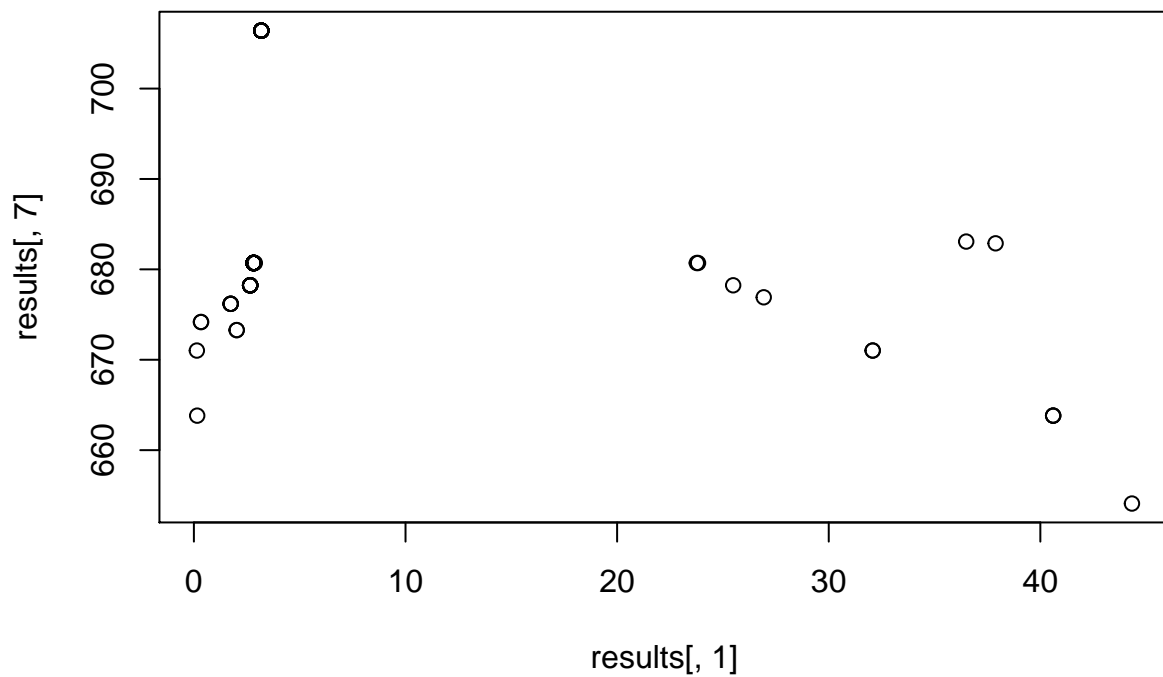


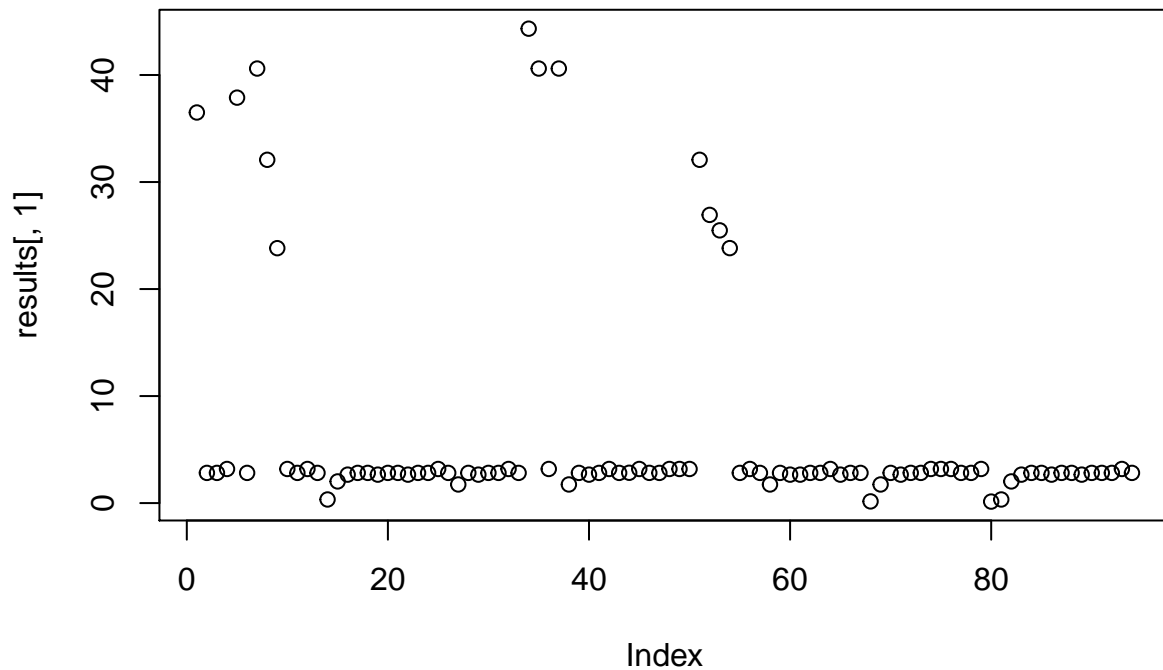


```
## 1 factor completed in 0.00386 minutes. Estimated time of completion: 2020-04-29 13:49:51 [1]
## [6] 5.997164e-06 3.000000e+01 8.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432854e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 40.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

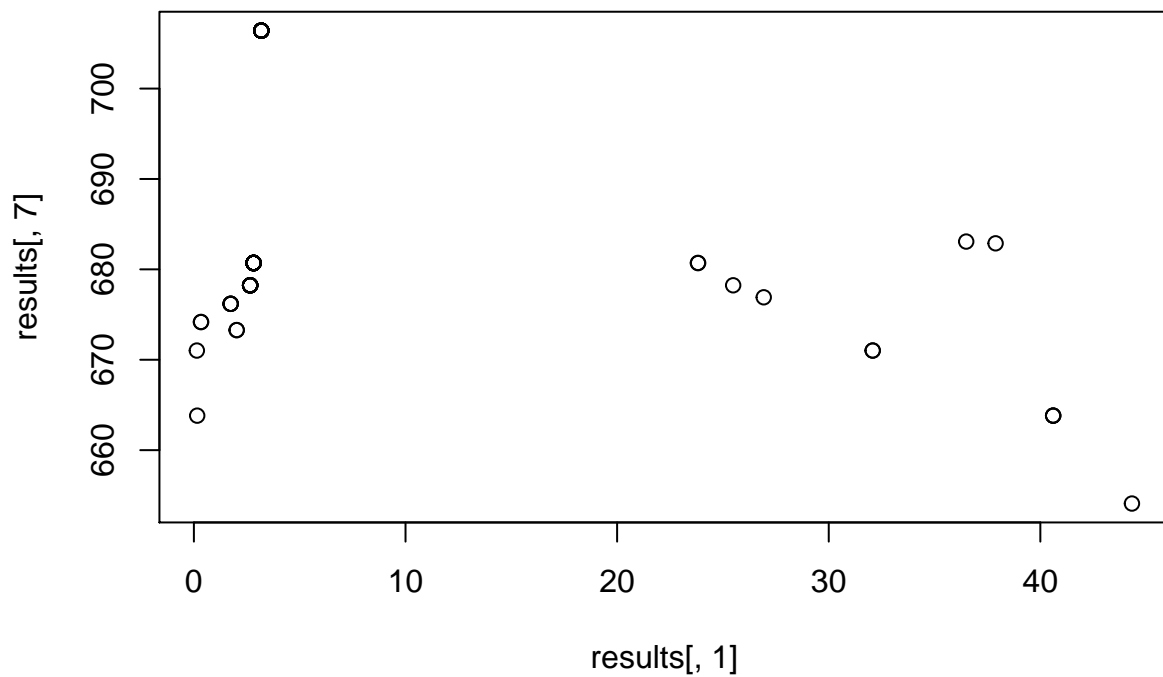


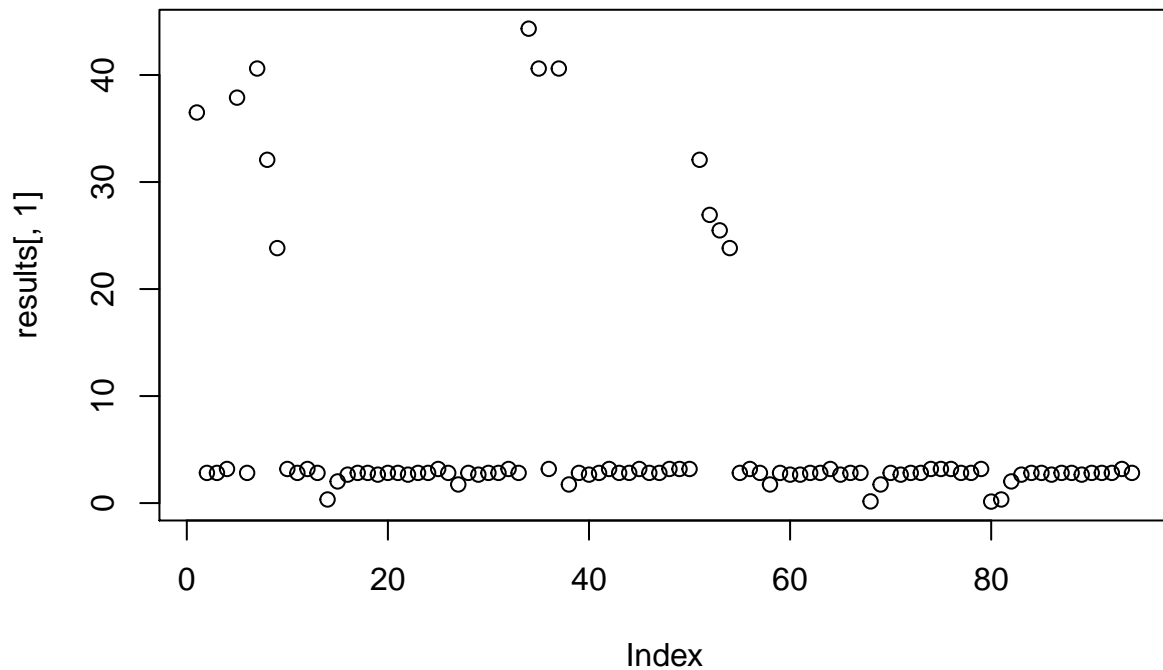


```
## 1 factor completed in 0.00386 minutes. Estimated time of completion: 2020-04-29 13:49:53 [1]
## [6] 2.528704e-08 4.000000e+01 8.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 50.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

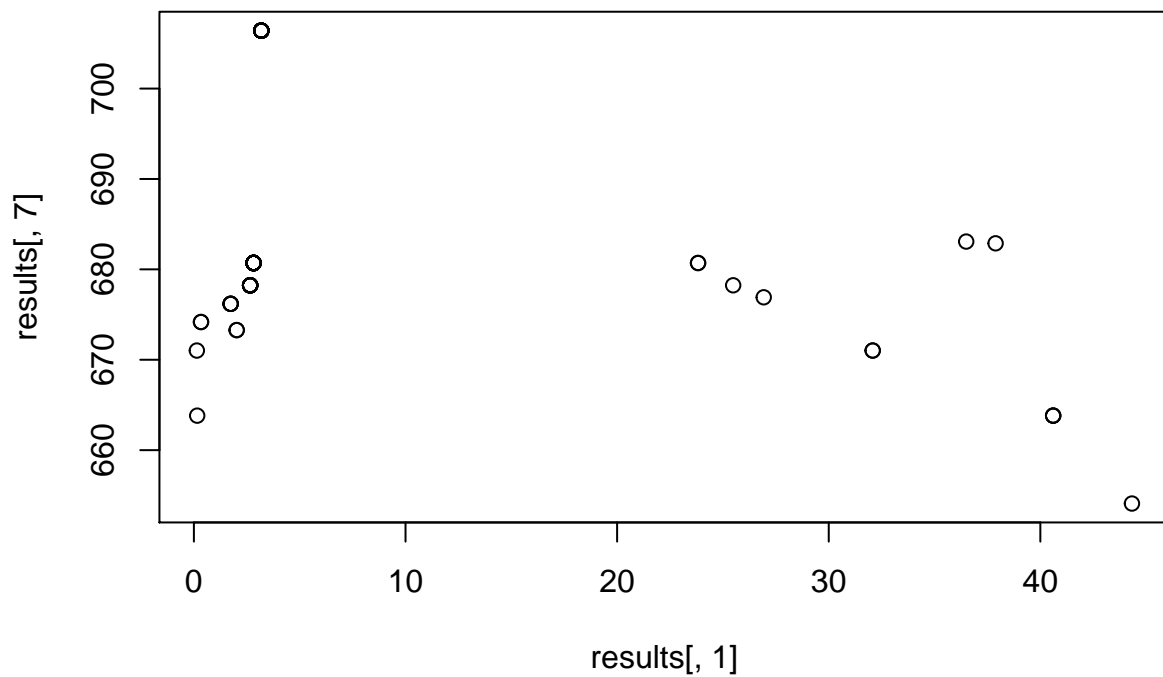


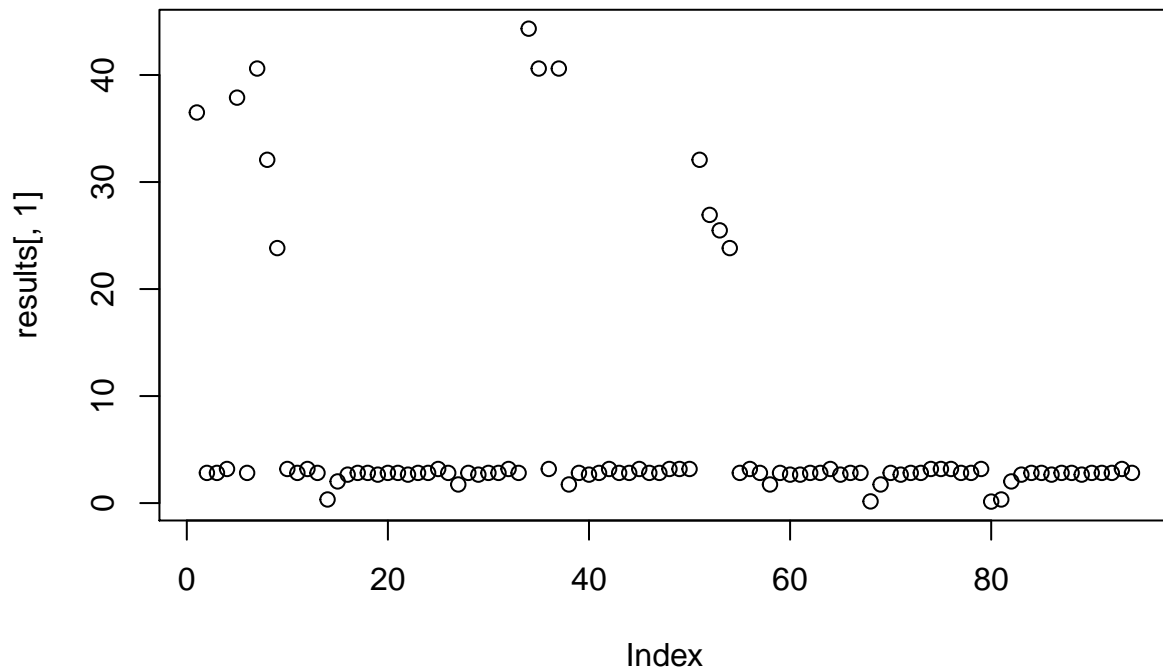


```
## 1 factor completed in 0.00394 minutes. Estimated time of completion: 2020-04-29 13:49:55 [1]
## [6] 1.066228e-10 5.000000e+01 8.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 60.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

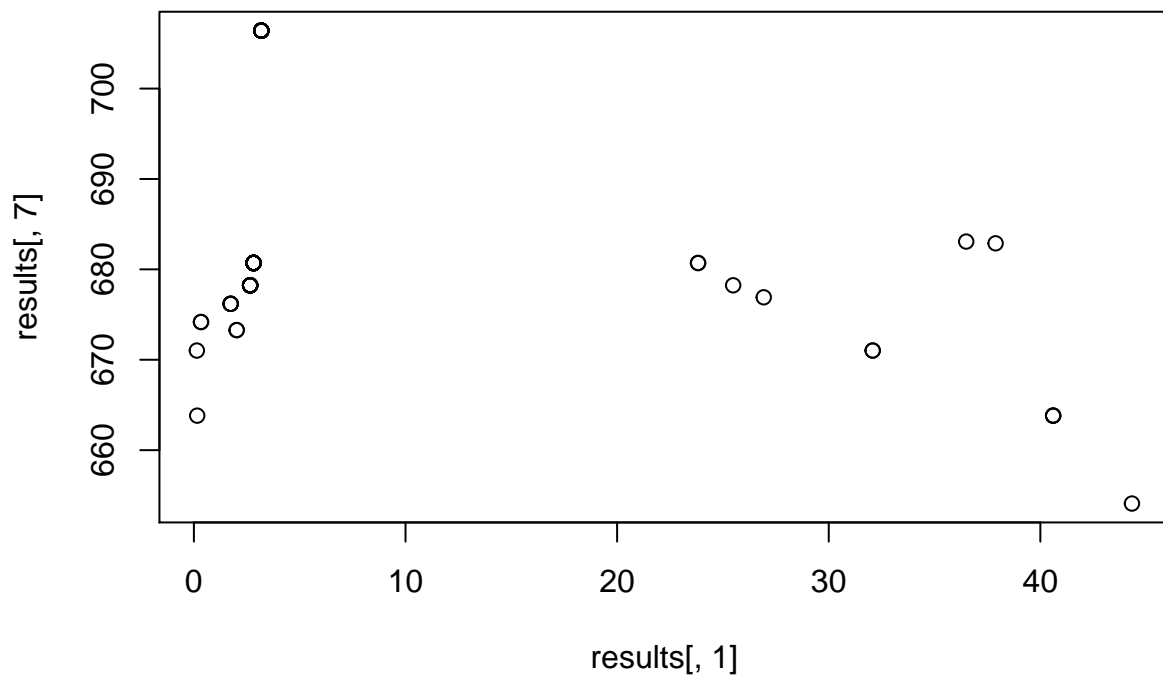


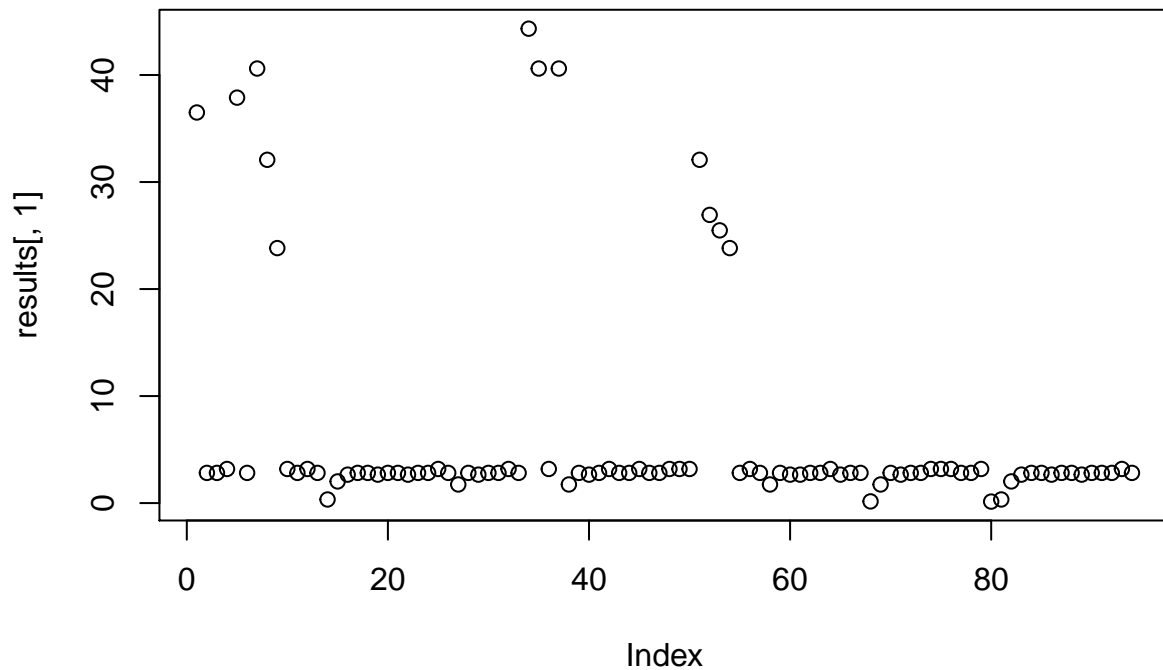


```
## 1 factor completed in 0.00387 minutes. Estimated time of completion: 2020-04-29 13:49:57 [1]
## [6] 4.495749e-13 6.000000e+01 8.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 70.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

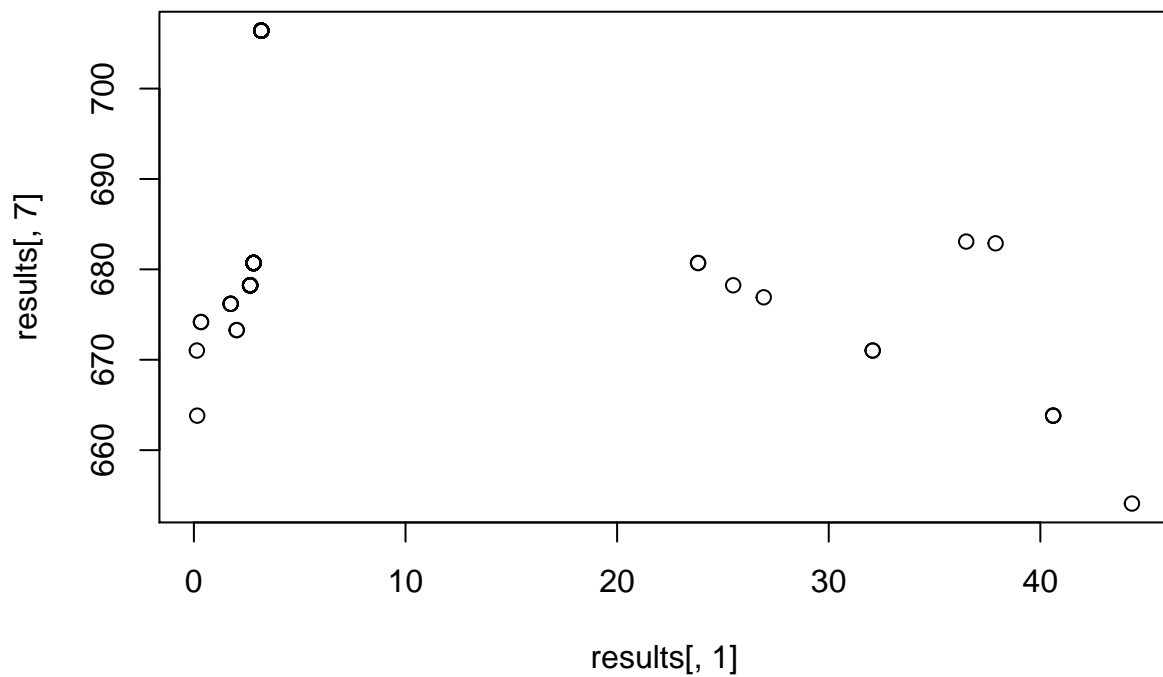


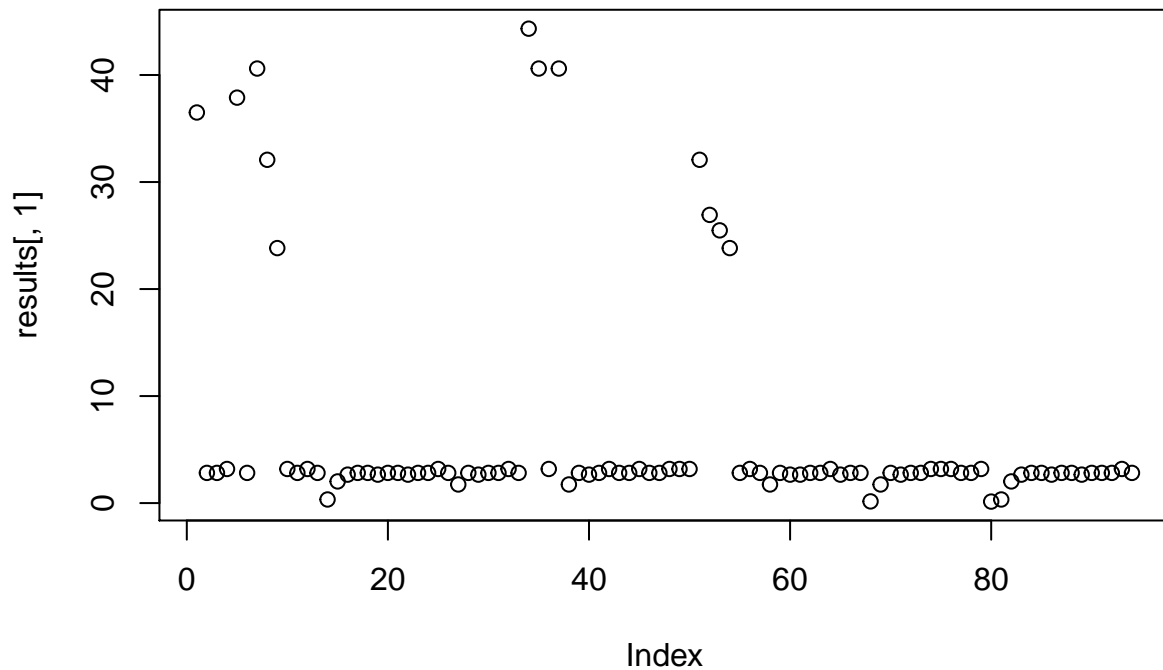


```
## 1 factor completed in 0.00403 minutes. Estimated time of completion: 2020-04-29 13:49:59 [1]
## [6] 1.895632e-15 7.000000e+01 8.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 80.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

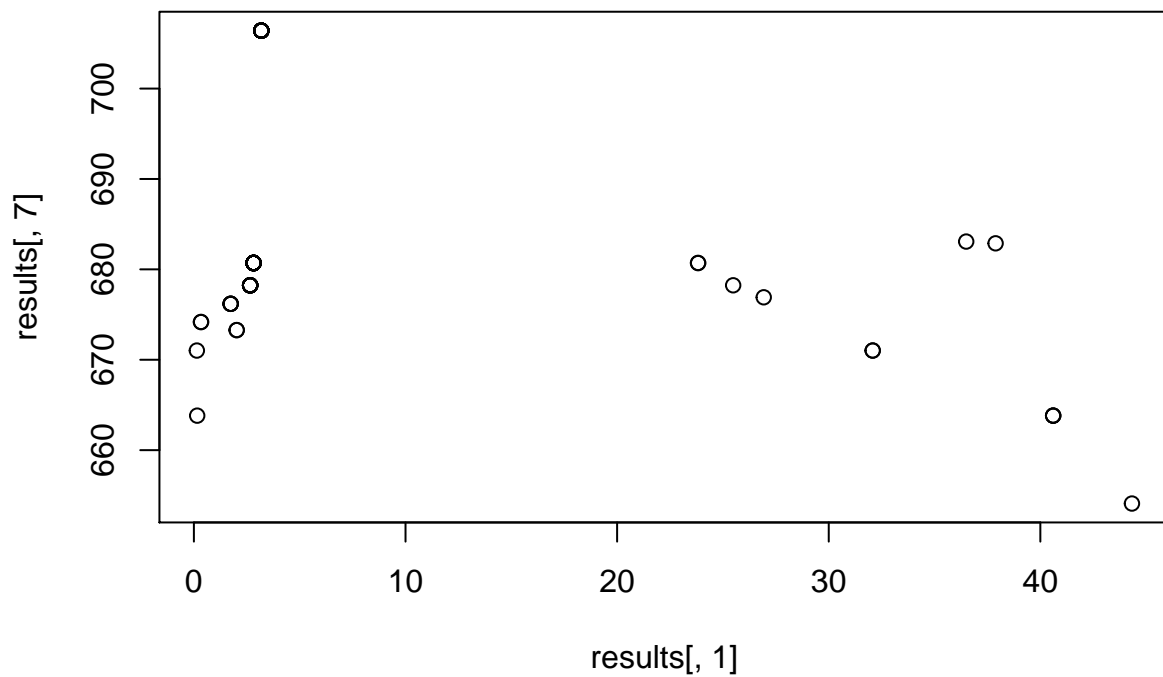


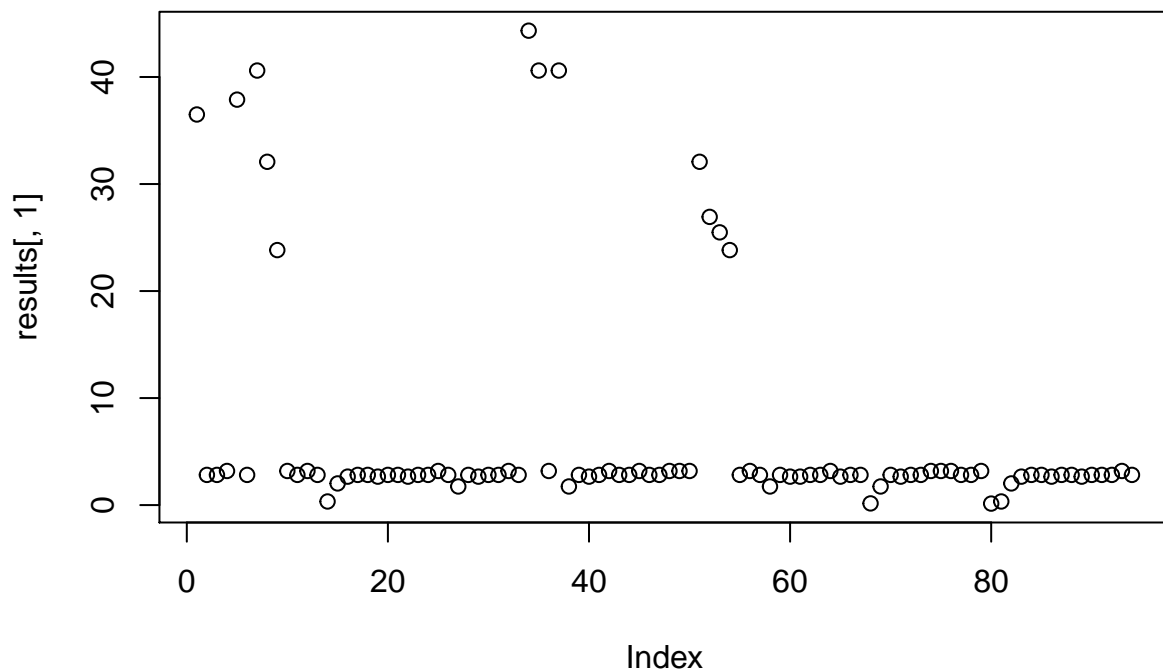


```
## 1 factor completed in 0.00371 minutes. Estimated time of completion: 2020-04-29 13:50:01 [1]
## [6] 7.992933e-18 8.000000e+01 8.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 90.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

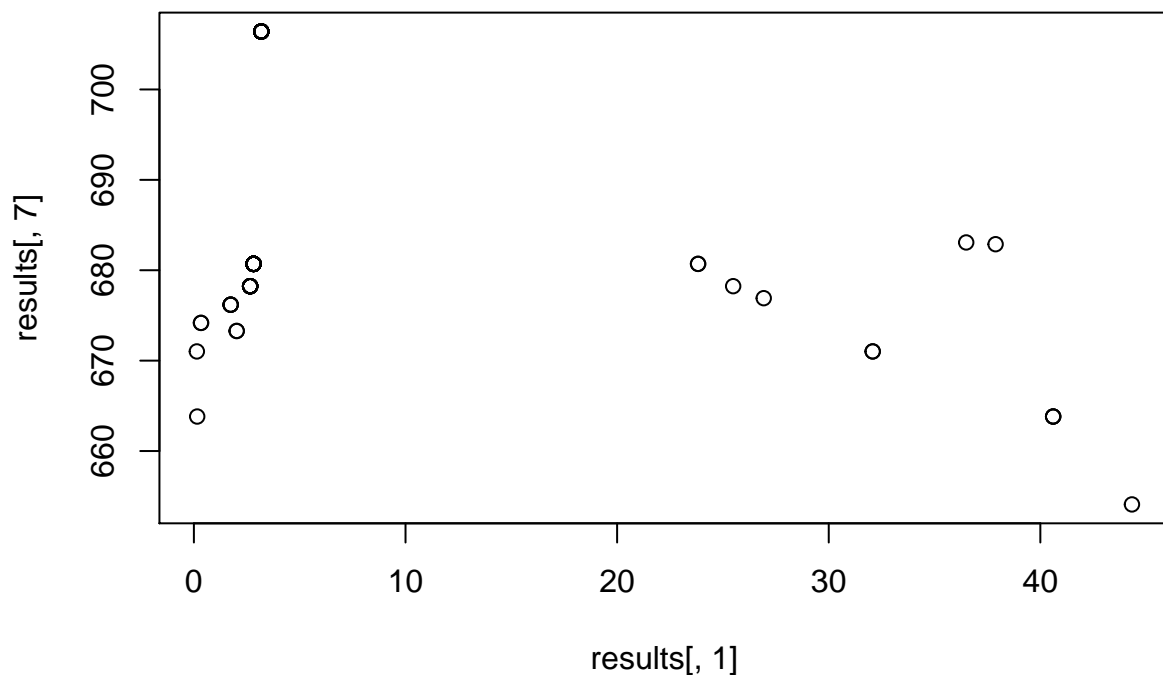


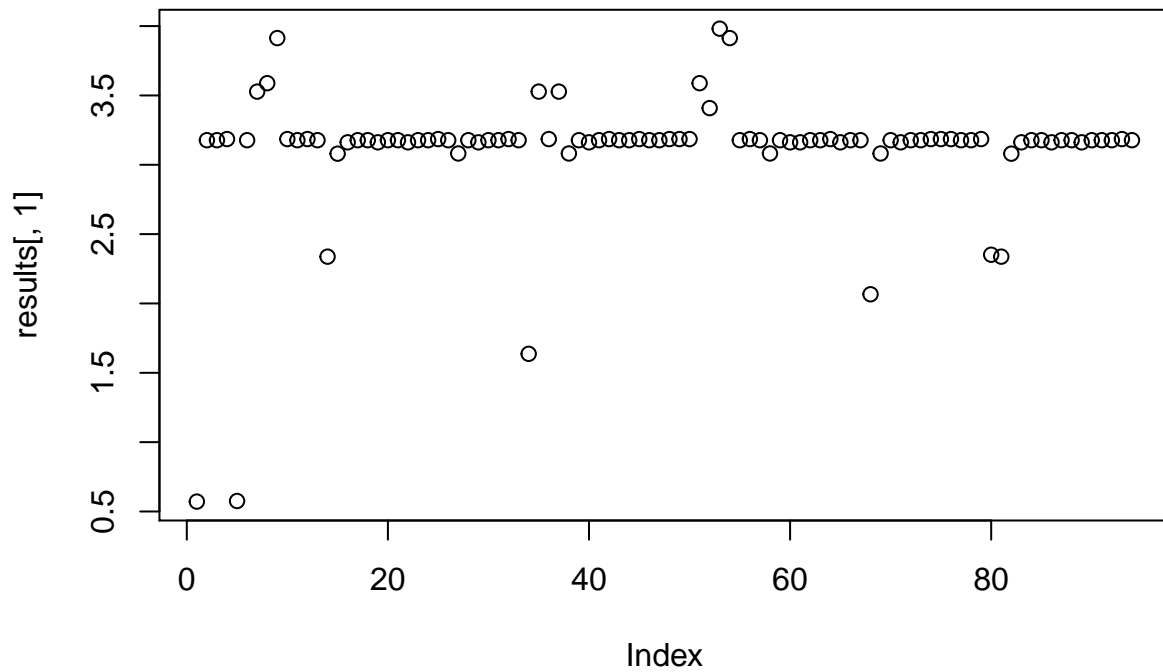


```
## 1 factor completed in 0.004 minutes. Estimated time of completion: 2020-04-29 13:50:03 [1] 1
## [6] 3.370220e-20 9.000000e+01 8.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "X = 90.000001"
## [1] "j = 1e-06"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

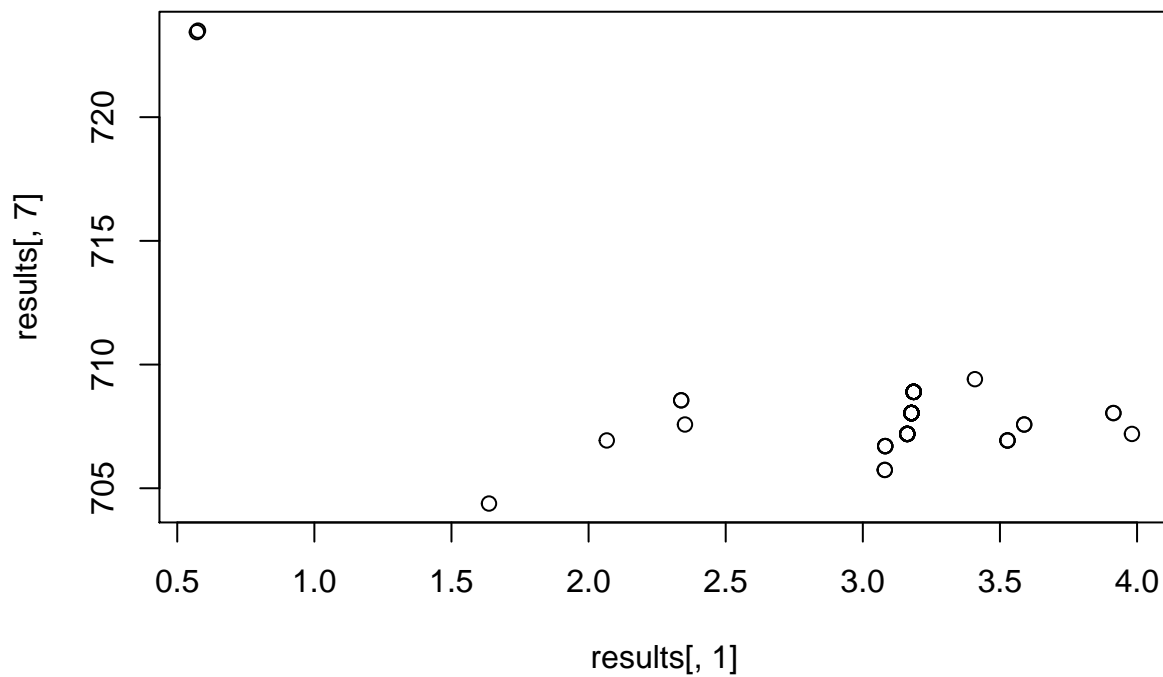


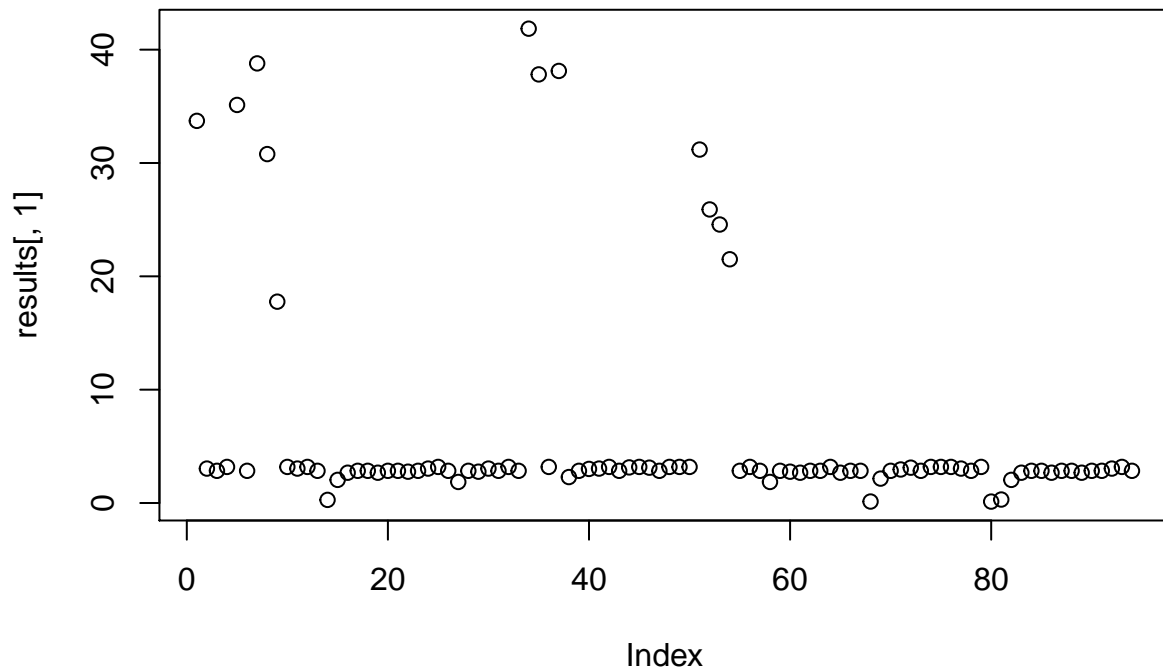


```
## 1 factor completed in 0.00389 minutes. Estimated time of completion: 2020-04-29 13:50:05 [1]
## [7] 0.0000010 90.0000010 7.8130568 723.4369699 1.6368055 704.3829845
## [13] 1.0000000 34.0000000
## [1] "j = 10.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

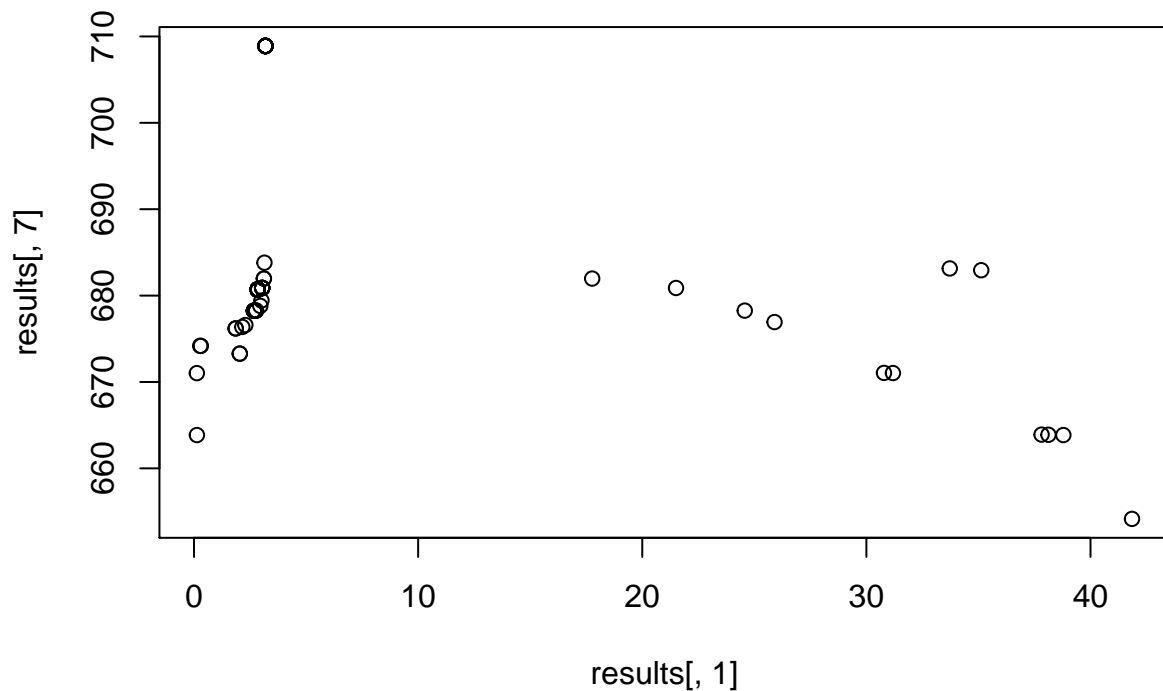


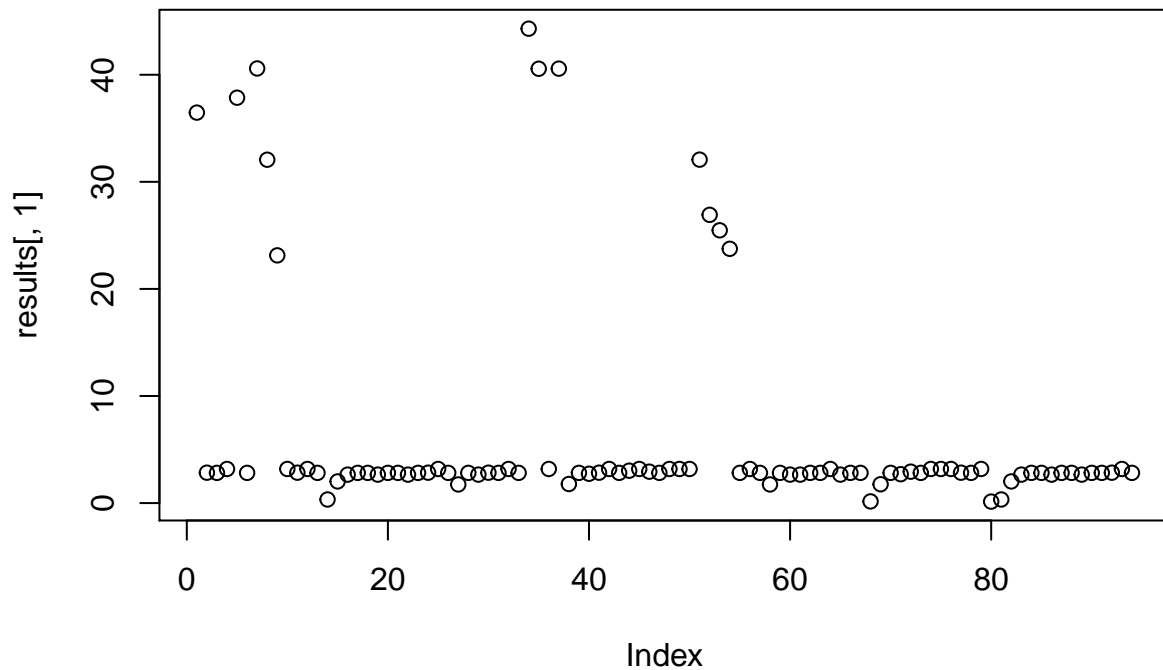


```
## 1 factor completed in 0.00382 minutes. Estimated time of completion: 2020-04-29 13:50:08 [1]
## [7] 10.0000010 90.0000010 7.8130568 671.0251968 41.8513630 654.1526742
## [13] 80.0000000 34.0000000
## [1] "j = 20.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

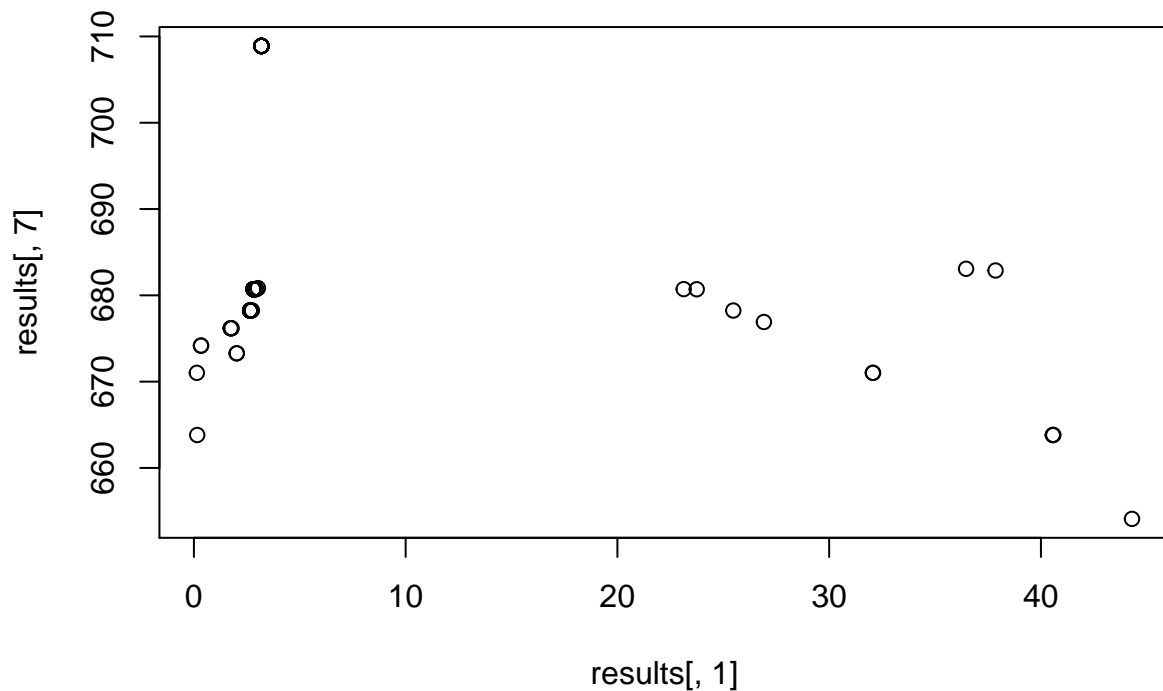


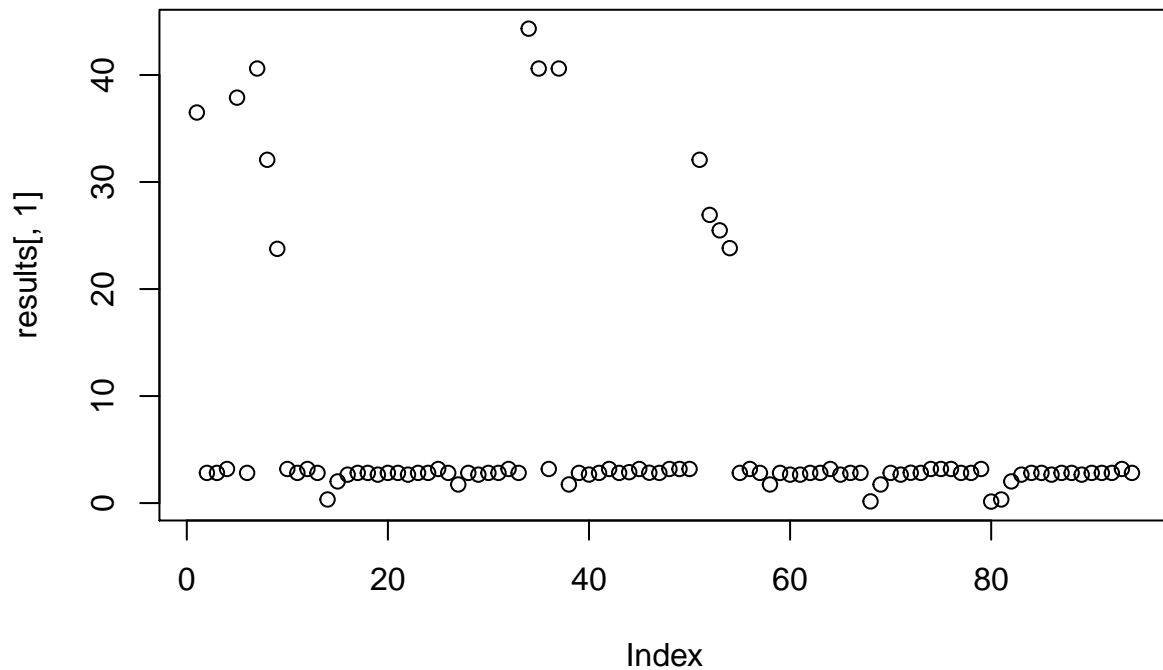


```
## 1 factor completed in 0.00411 minutes. Estimated time of completion: 2020-04-29 13:50:09 [1]
## [6] 0.001600097 20.000001000 90.000001000 7.813056846 671.019586919
## [11] 44.301709957 654.095689346 80.000000000 34.000000000
## [1] "j = 30.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

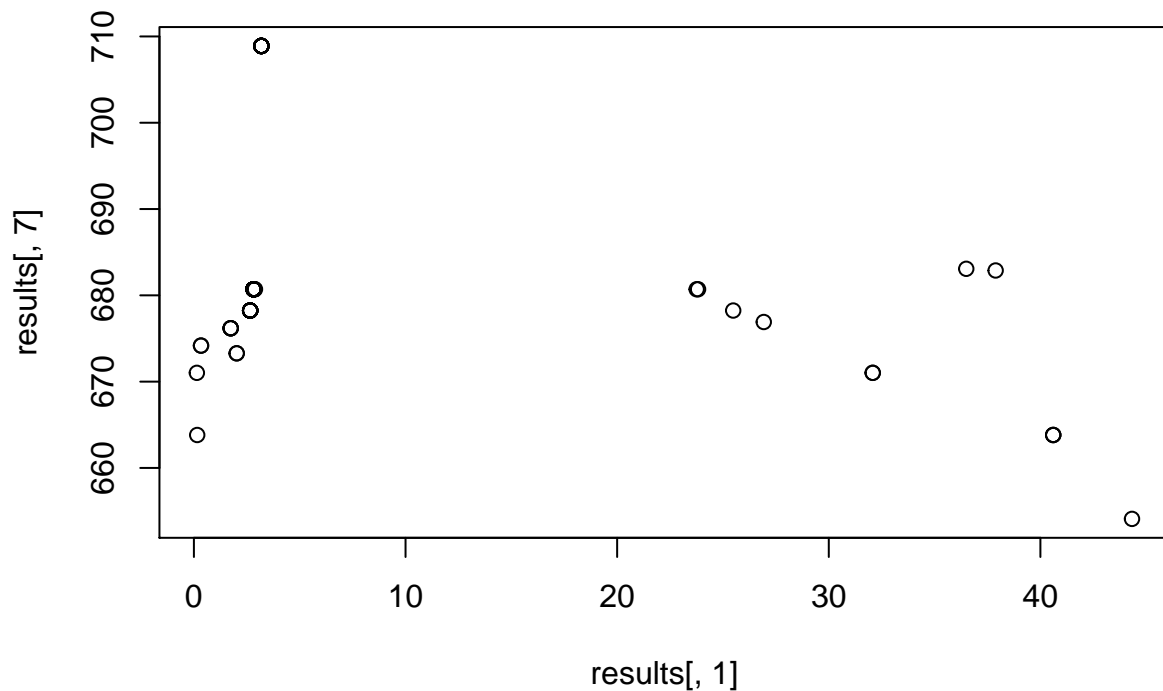


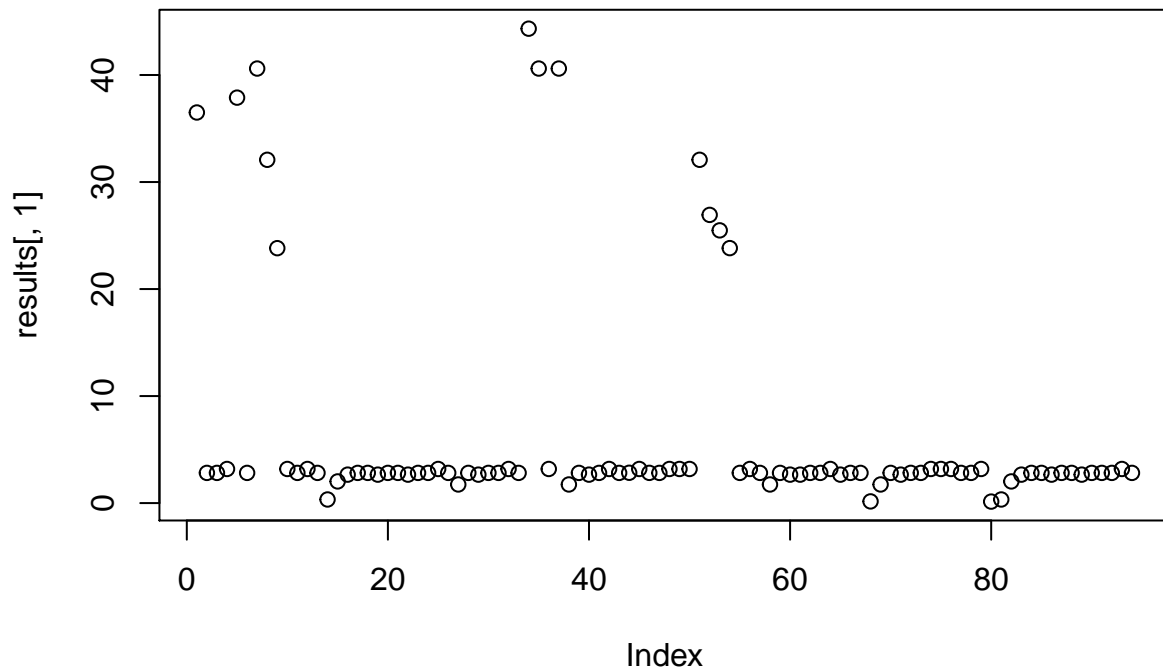


```
## 1 factor completed in 0.00392 minutes. Estimated time of completion: 2020-04-29 13:50:11 [1]
## [6] 6.746809e-06 3.000000e+01 9.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432851e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 40.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

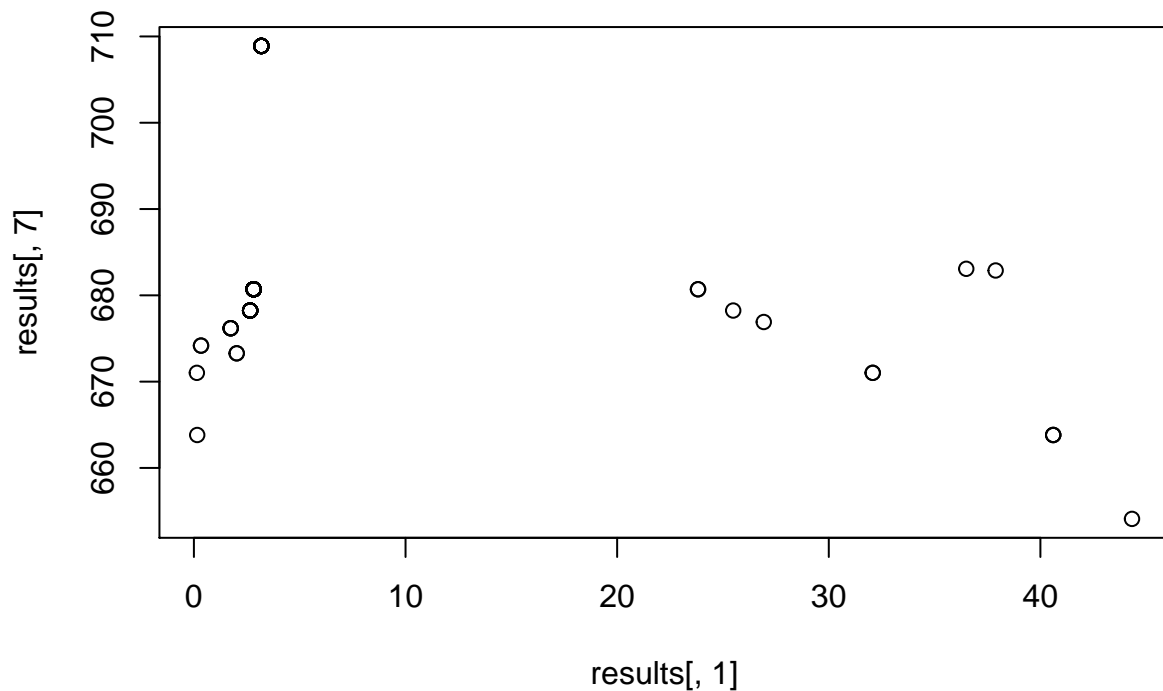


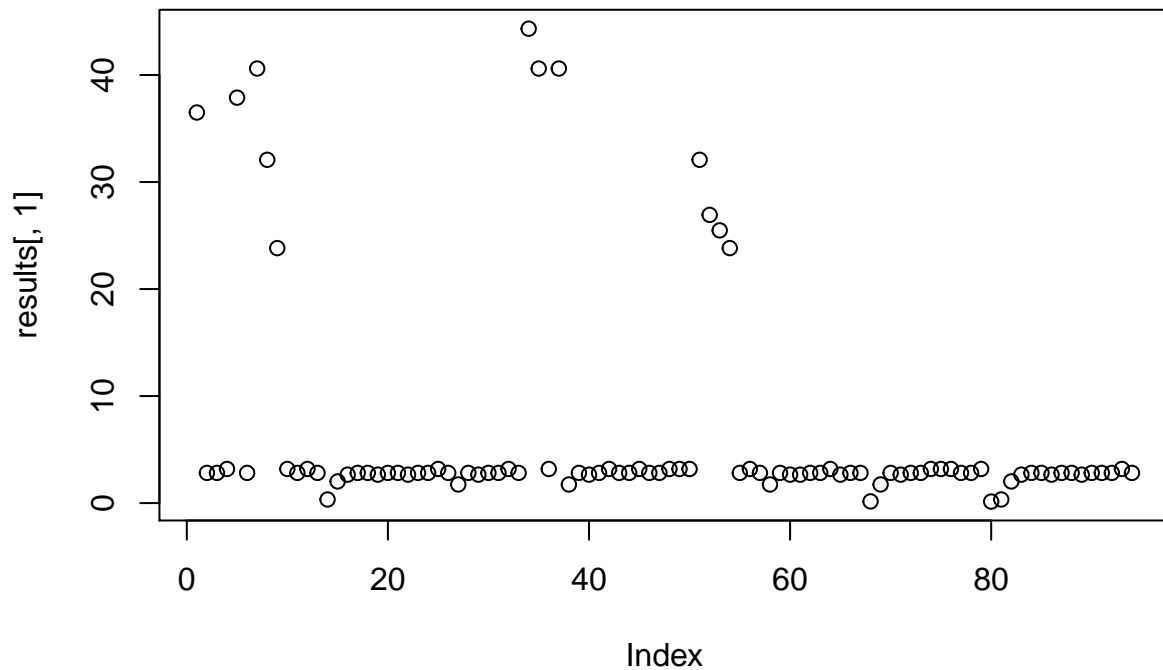


```
## 1 factor completed in 0.00396 minutes. Estimated time of completion: 2020-04-29 13:50:13 [1]
## [6] 2.844792e-08 4.000000e+01 9.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 50.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

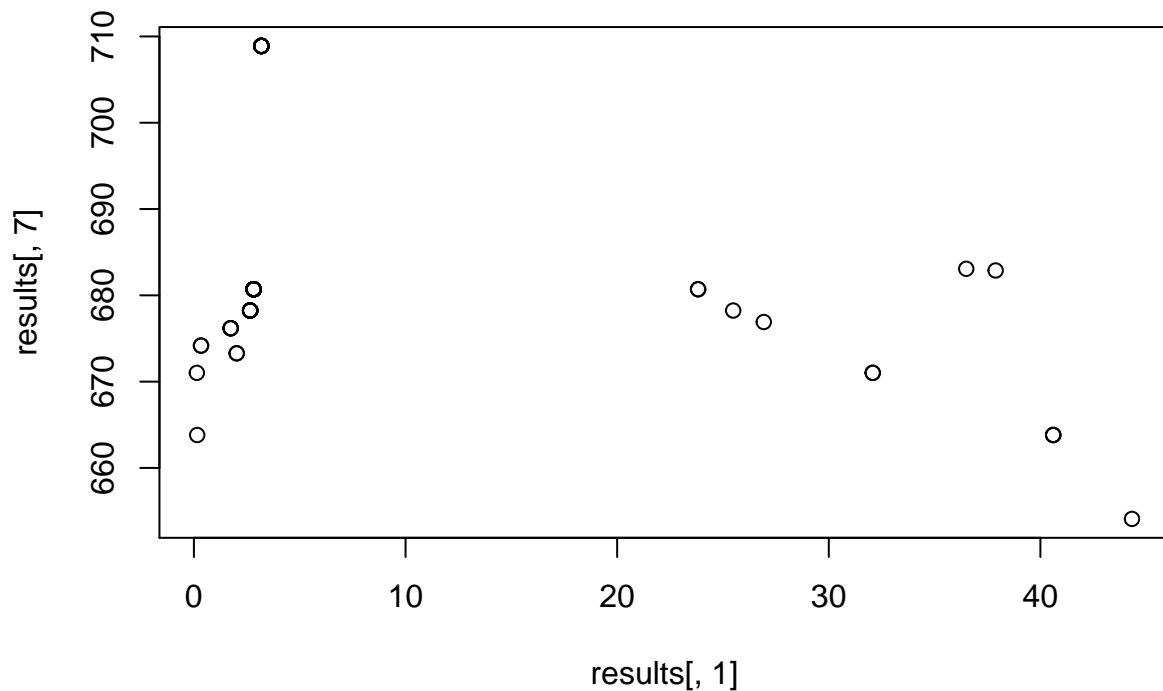


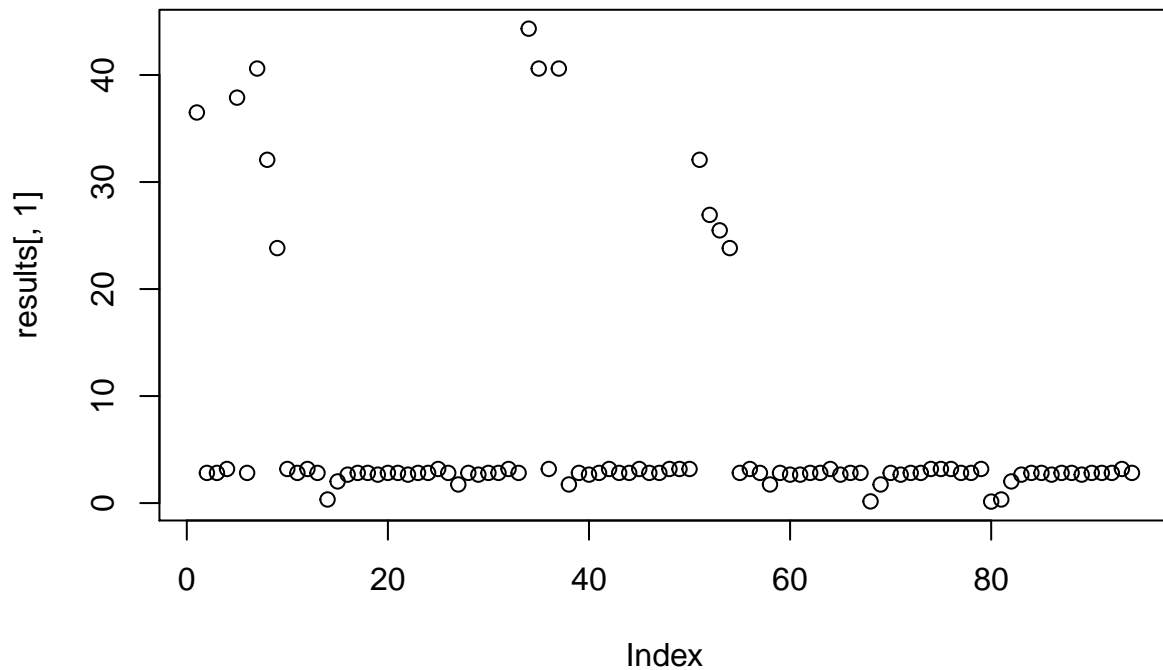


```
## 1 factor completed in 0.00399 minutes. Estimated time of completion: 2020-04-29 13:50:15 [1]
## [6] 1.199506e-10 5.000000e+01 9.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 60.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

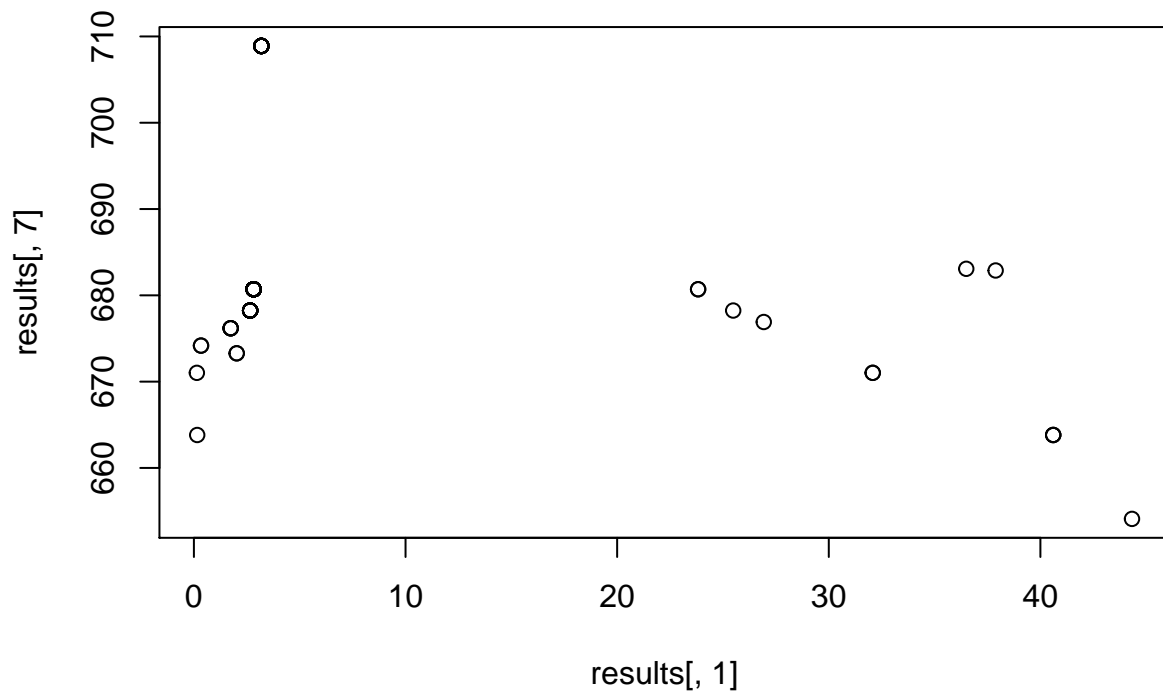


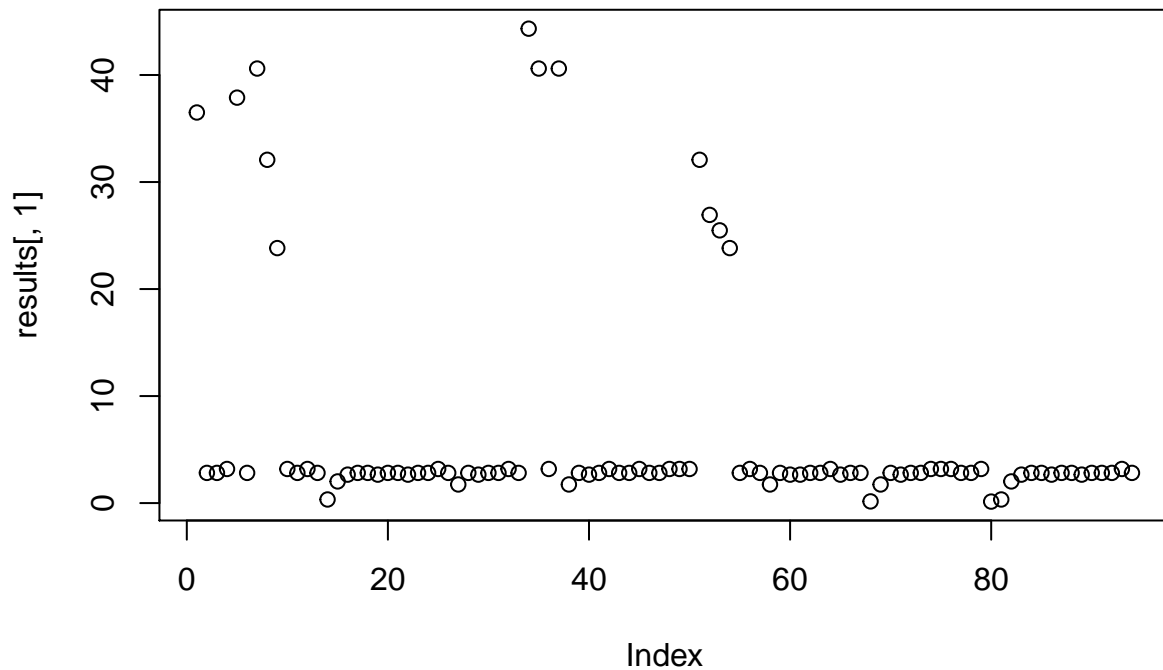


```
## 1 factor completed in 0.00393 minutes. Estimated time of completion: 2020-04-29 13:50:17 [1]
## [6] 5.057718e-13 6.000000e+01 9.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 70.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

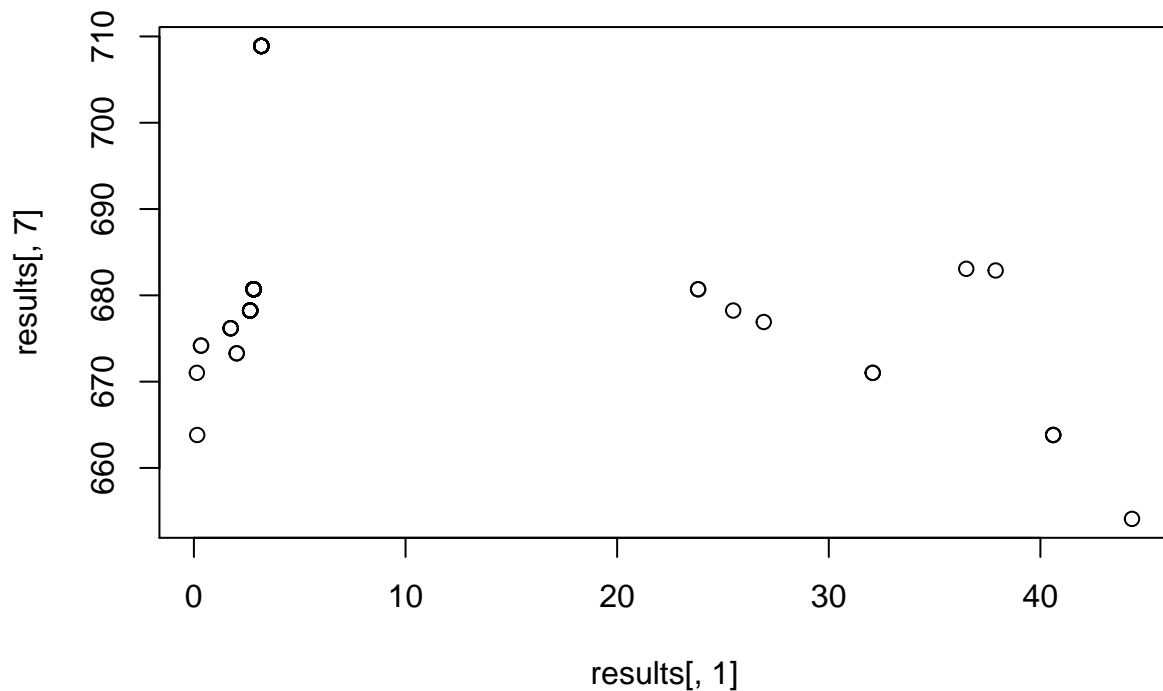


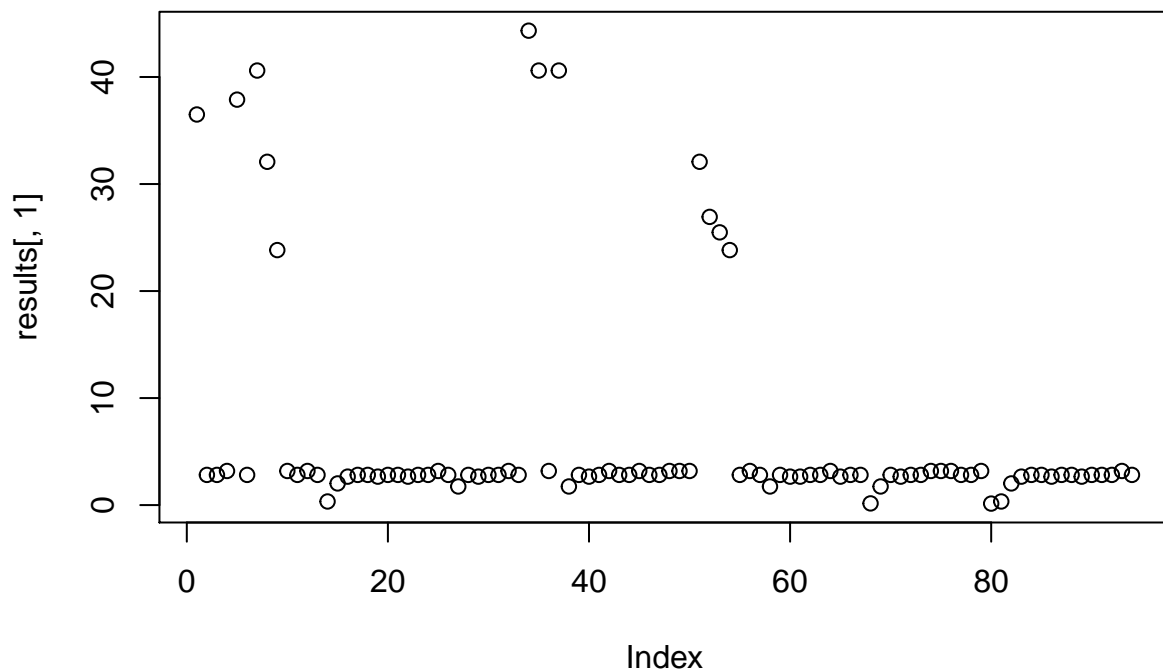


```
## 1 factor completed in 0.00425 minutes. Estimated time of completion: 2020-04-29 13:50:19 [1]
## [6] 2.132586e-15 7.000000e+01 9.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 80.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```

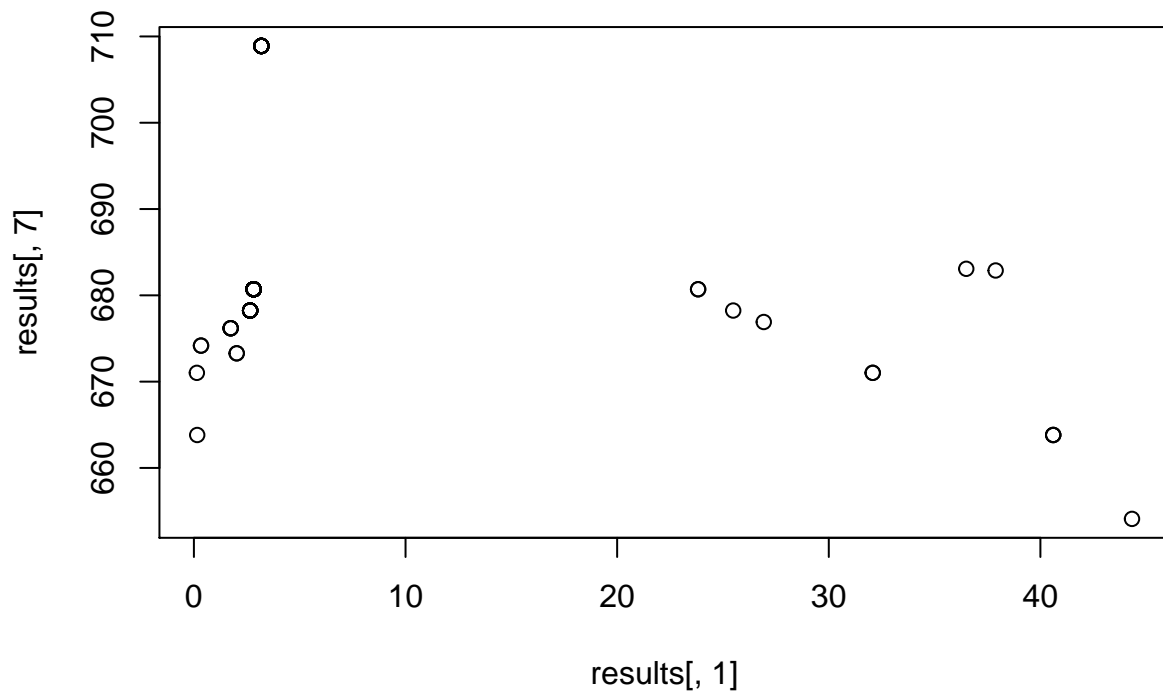


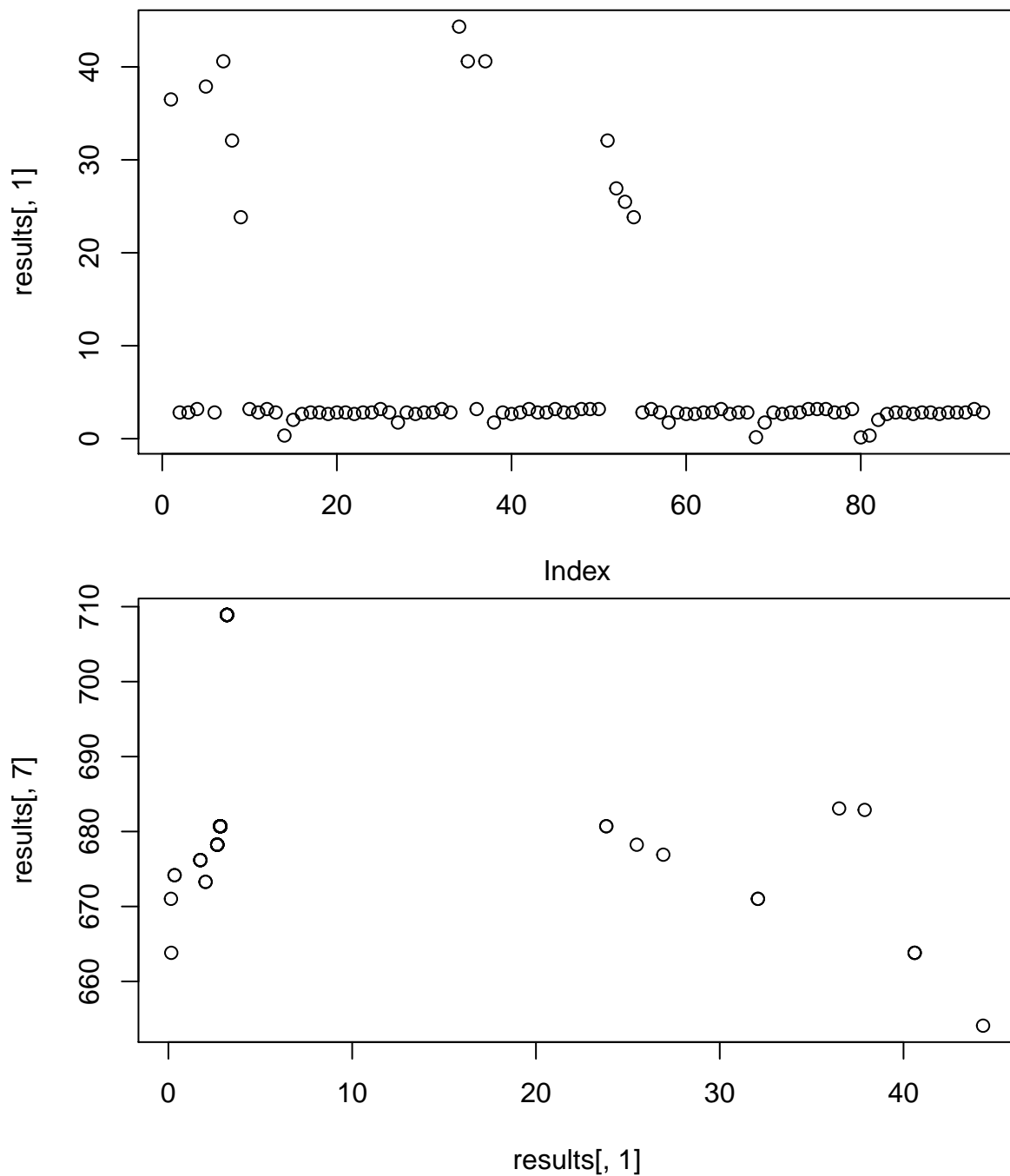


```
## 1 factor completed in 0.00389 minutes. Estimated time of completion: 2020-04-29 13:50:21 [1]
## [6] 8.992050e-18 8.000000e+01 9.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
## [1] "j = 90.000001"

## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.

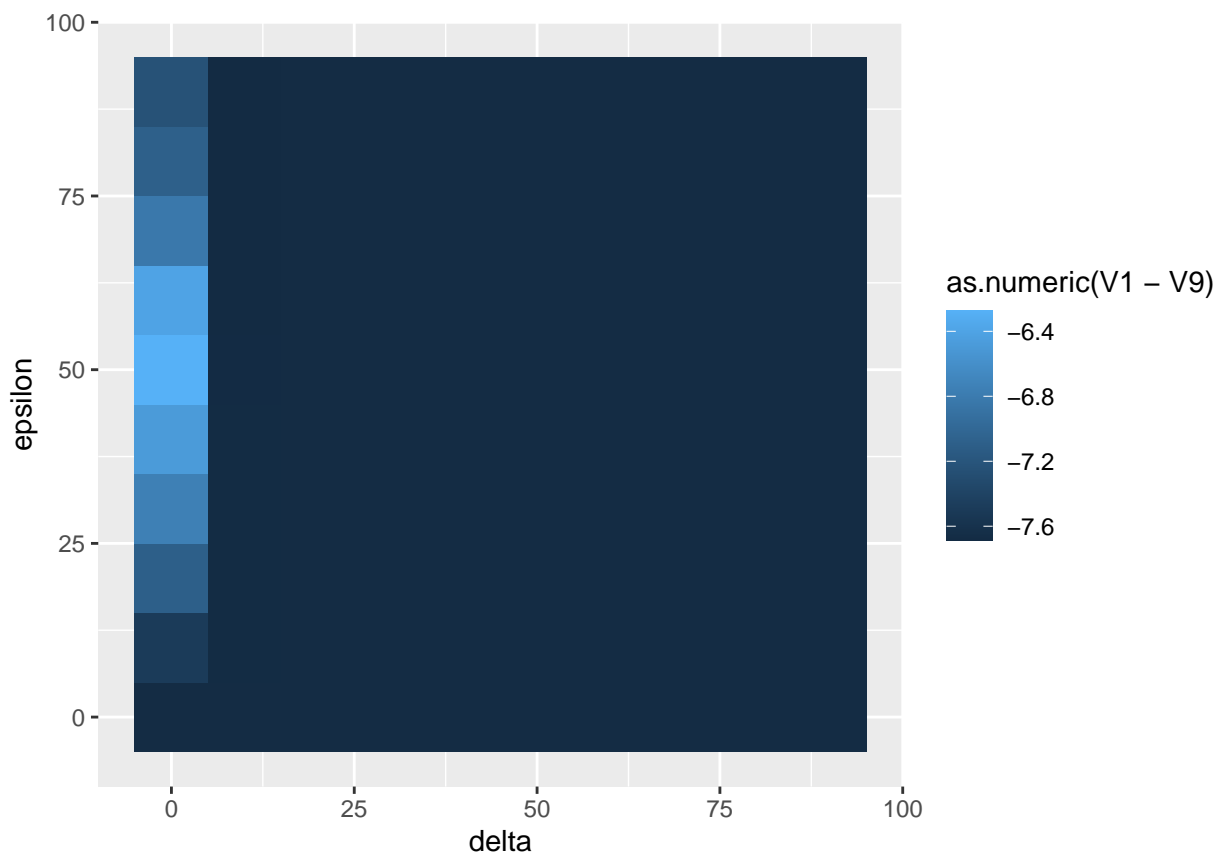
## Warning in test_BodySize[, "intercept"] * beta_ridge[2] + t(basis_map) * : Recycling array of length
## Use c() or as.vector() instead.
```



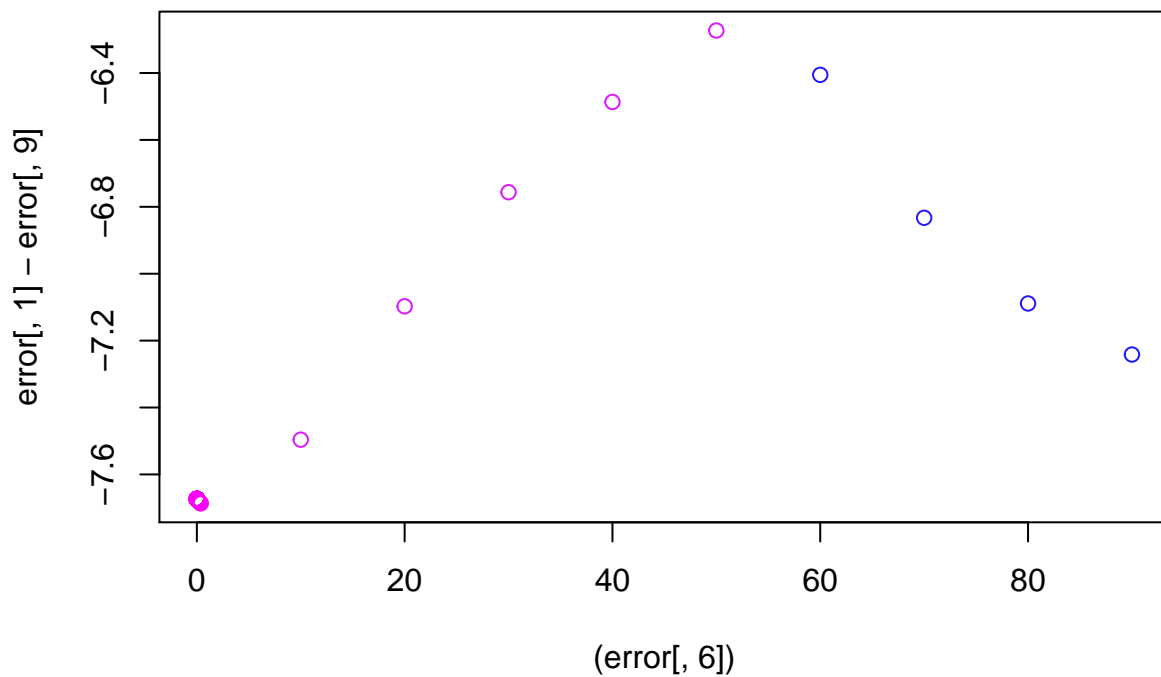


```
## 1 factor completed in 0.00386 minutes. Estimated time of completion: 2020-04-29 13:50:23 [1]
## [6] 3.791497e-20 9.000000e+01 9.000000e+01 7.813057e+00 6.710196e+02
## [11] 4.432879e+01 6.540957e+02 8.000000e+01 3.400000e+01
```

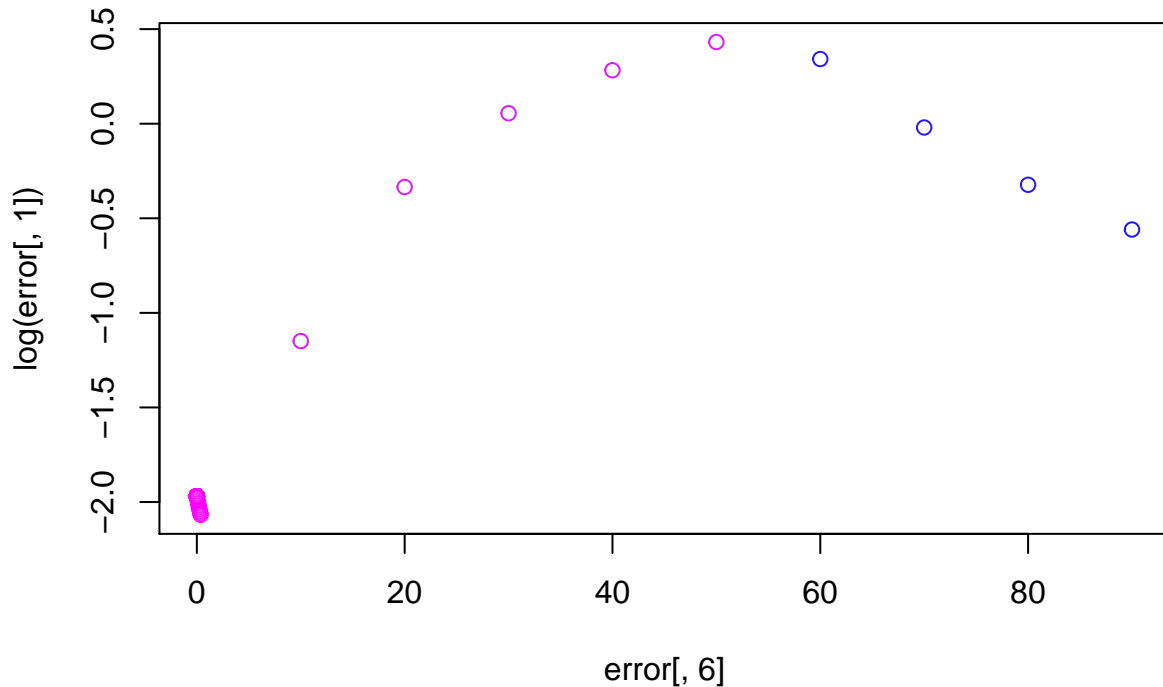
```
error %>%
  as.data.frame() %>%
  ggplot2::ggplot(aes(V7, V8)) +
  ggplot2::geom_tile(aes(fill= as.numeric(V1 - V9) )) +
  ggplot2::xlab("delta") +
  ggplot2::ylab("epsilon")
```



```
plot(y = error[,1] - error[,9], x = (error[,6]), col = rgb(red = error[,13]/max(error[,13]), green = 0,
```



```
plot(y = log(error[,1]), x = error[,6], col = rgb(red = error[,13]/max(error[,13]), green = 0, blue = 1
```



```
error[,13]
```

```
## [1] 80 80 80 80 80 80 80 80 80 80 80 68 80 80 80 80 80 80 80 80 68 80 80 80 80
## [26] 80 80 80 80 80 68 80 80 80 80 80 80 80 80 80 68 80 80 80 80 80 80 80 80 80
## [51] 68 80 80 80 80 80 80 80 80 80 80 5 80 80 80 80 80 80 80 80 80 5 80 80 80 80
## [76] 80 80 80 80 80 5 80 80 80 80 80 80 80 80 80 1 80 80 80 80 80 80 80 80 80
```

```
error[,14]
```

```
## [1] 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34
## [26] 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34
## [51] 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34
## [76] 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34
```

```
train_BodySize[getPhyloGroups(train_tree)[unique(error[,13])[1]][[1]][[1]],]$Species
```

```
## [1] "43" "44" "45" "46" "47" "48" "49" "50"
```

```
train_BodySize[getPhyloGroups(train_tree)[unique(error[,13])[2]][[1]][[1]],]$Species
```

```
## [1] "37" "38" "39" "40" "41" "42" "43" "44" "45" "46" "47" "48" "49" "50"
```

```
train_BodySize[getPhyloGroups(train_tree)[unique(error[,13])[3]][[1]][[1]],]$Species
```

```
## [1] "3" "4" "6" "7" "8" "9" "10" "11" "12" "13" "14" "15" "16" "17" "18"
```

```
train_BodySize[getPhyloGroups(train_tree)[unique(error[,13])[4]][[1]][[1]],]$Species
```

```
## [1] "1" "2" "3" "4" "6" "7" "8" "9" "10" "11" "12" "13" "14" "15" "16"
```

```
## [16] "17" "18"
```

```
train_BodySize[getPhyloGroups(train_tree)[unique(error[,13])[5]][[1]][[1]],]$Species
```

```
## character(0)
```

```
train_BodySize[getPhyloGroups(train_tree)[unique(error[,13])[6]][[1]][[1]],]$Species
```

```

## character(0)
train_BodySize[getPhyloGroups(train_tree)[unique(error[,13])[7]][[1]][[1]],]$Species

## character(0)
train_BodySize[getPhyloGroups(train_tree)[unique(error[,13])[8]][[1]][[1]],]$Species

## character(0)
train_BodySize[getPhyloGroups(train_tree)[unique(error[,13])[9]][[1]][[1]],]$Species

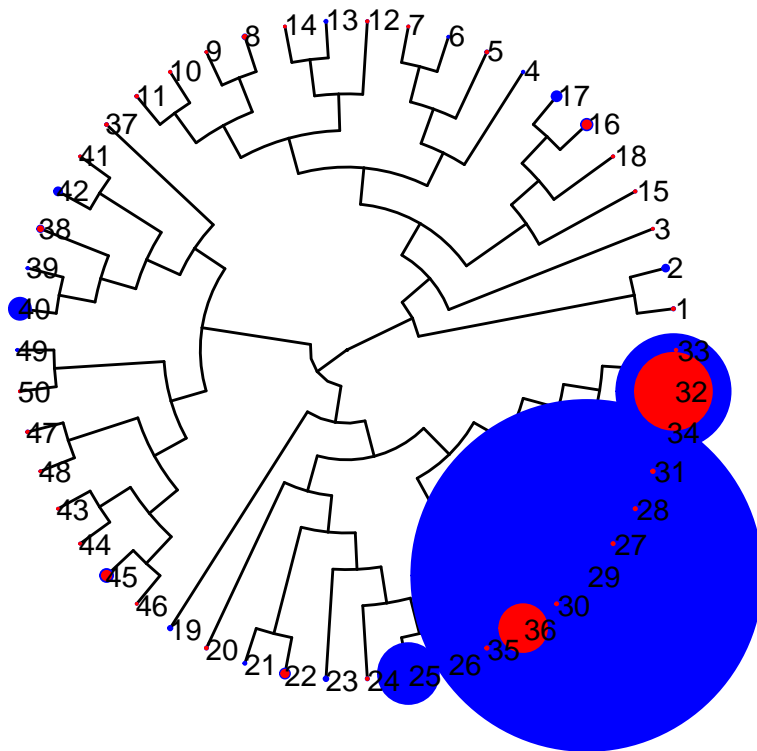
## character(0)
train_BodySize[getPhyloGroups(train_tree)[unique(error[,13])[10]][[1]][[1]],]$Species

## character(0)
train_BodySize[getPhyloGroups(train_tree)[unique(error[,13])[11]][[1]][[1]],]$Species

## character(0)
train_BodySize[getPhyloGroups(train_tree)[unique(error[,13])[12]][[1]][[1]],]$Species

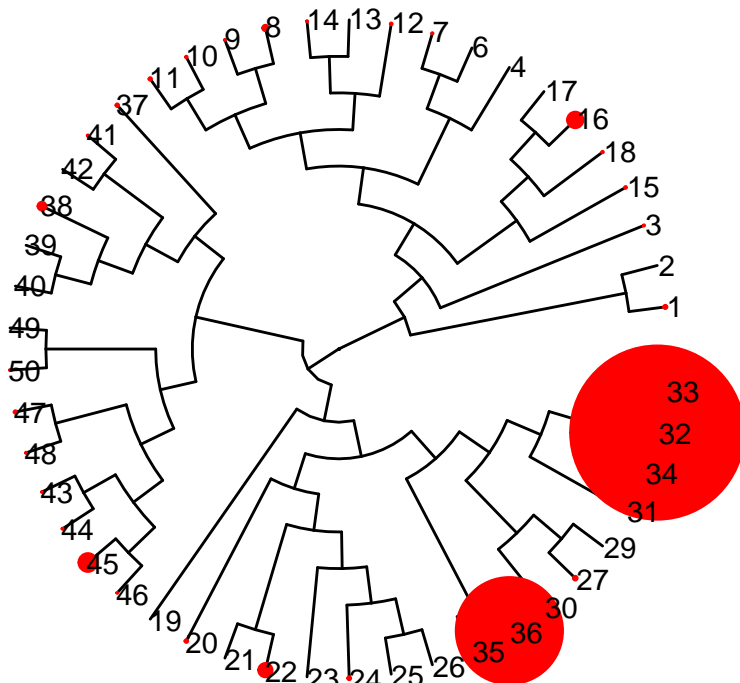
## character(0)


```



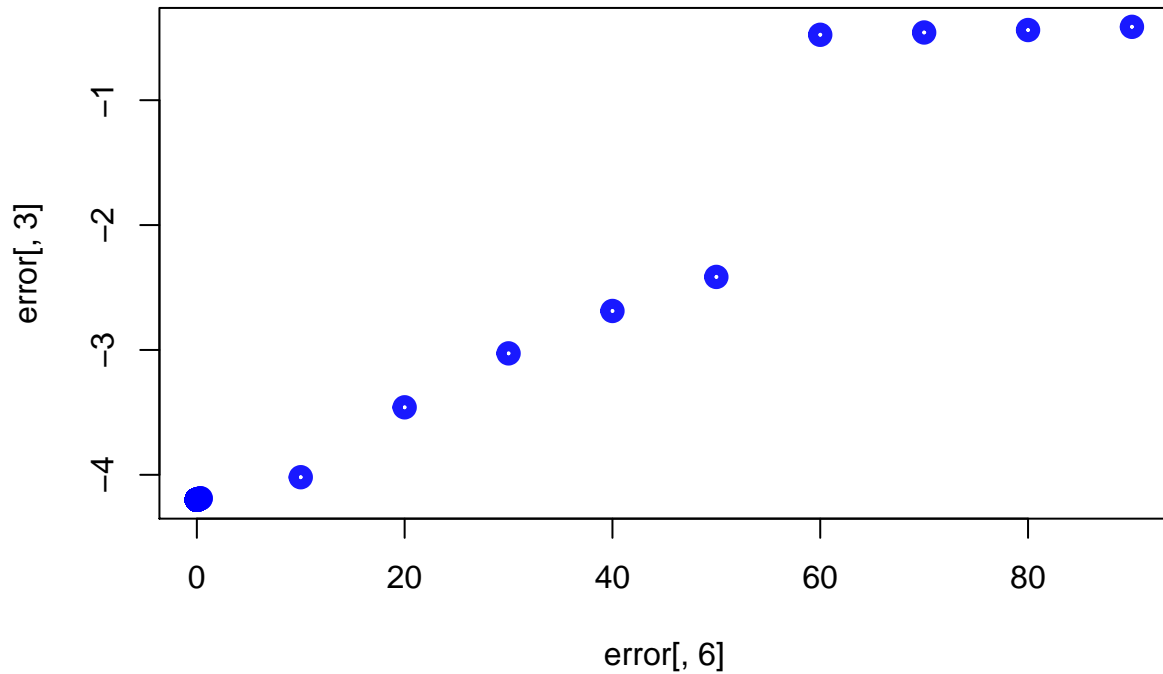
```
ggtree::ggtree(train_tree, branch.length = 'none', layout = 'fan') +
  ggtree::geom_tippoint(size=.23*as.numeric(as.character(train_BodySize$BodySize_miss)),col='red') +
  ggtree::geom_tiplab()
```

Warning: Removed 16 rows containing missing values (geom_point_g_gtree).



```
# plot((error[,1]), type = 'l', lwd = 5, col = rgb(red = 0, green = 0, blue = 1, alpha = 0.9))
# plot(error[,1], error[,7], lwd = 5, col = rgb(red = 0, green = 0, blue = 1, alpha = 0.9))
```

```
# plot(error[,1], error[,8], lwd = 5, col = rgb(red = 0, green = 0, blue = 1, alpha = 0.9))
# plot(error[,1], error[,6], lwd = 5, col = rgb(red = 0, green = 0, blue = 1, alpha = 0.9))
#
plot(y =error[,3], x=error[,6], lwd = 5, col = rgb(red = 0, green = 0, blue = 1, alpha = 0.9))
```



```
#
# plot((error[,10]), type = 'l', lwd = 5, col = rgb(red = 1, green = 0, blue = 1, alpha = 0.9))
#
# plot((error[,11]), type = 'l', lwd = 5, col = rgb(red = 0, green = 0, blue = 1, alpha = 0.9))
# plot((error[,12]), type = 'l', lwd = 5, col = rgb(red = 0, green = 0, blue = 1, alpha = 0.9))

plot(error[,11] - error[,9], col = rgb(red = 0, green = 0, blue = 1, alpha = 0.9))
```

