Real Data applicaion of flashtpx: Bird Abundance Data due to Price et al

Wei Wang, Kushal K Dey

April 1, 2016

Contents

Overview

We now apply `flashtpx` on a real data. This data is collected by Trevor Price et al and does not yet have open access. But the good thing about this data is that $K = 2$ and $K = 3$ clusters, the standard GoM model gave pretty interesting results where it seemed to pick up the elevation gradient for $K = 2$ and the geographic gradient (East and West) for $K = 3$. The other good thing is that the dataset is pretty small and since there is an issue about the scaling of `flashtpx` for large datasets, this seems like the ideal first real data to apply this method on.

We also compare the results from standard goM model fit, the `flashtpx` model fit and the PMA model fit.

Data Preparation

```
rm(list = ls())
library(ashr)
library(irlba)
library(PMA)
library(maptpx)
source("../R/flash.R")
```

```
data = read.csv("../external_data/Himalayan_grid_matrix.csv",
    header = TRUE)

counts = as.matrix(data[, -1])
```

```
rownames(counts) = data[, 1]

new_data1 <- data.frame(read.csv("../external_data/MohanGrids.csv"))
new_data2 <- data.frame(read.csv("../external_data/MohanGrids2.csv"))

bird_species <- union(as.vector(colnames(counts)),
    union(as.vector(new_data1[, 1]), as.vector(new_data2[,
        1])))

new_data <- matrix(0, dim(counts)[1] + 3, length(bird_species))
new_data[1, match(new_data1[, 1], bird_species)] = new_data1[,
    2]
new_data[2, match(new_data1[, 1], bird_species)] = new_data1[,
    3]
new_data[3, match(new_data2[, 1], bird_species)] = new_data2[,
    2]
new_data[4:(dim(counts)[1] + 3), match(colnames(counts),
    bird_species)] = counts

new_counts <- as.matrix(new_data)

rownames(new_counts) <- c(c("U1", "U2", "MA1"),
    rownames(counts))
colnames(new_counts) <- bird_species
new_counts <- new_counts[-(1:3), ]
```

Next we load the metadata on the samples.

```
metadata = read.csv("../external_data/Himalayan_grid_metadata.csv",
    header = TRUE)
elevation_metadata = metadata$Elevation[match(rownames(new_counts),
    metadata[, 1])]
east_west_dir = metadata$WorE[match(rownames(new_counts),
    metadata[, 1])]
```

Standard GoM model fitting

We now fit the topic model for $K = 2$

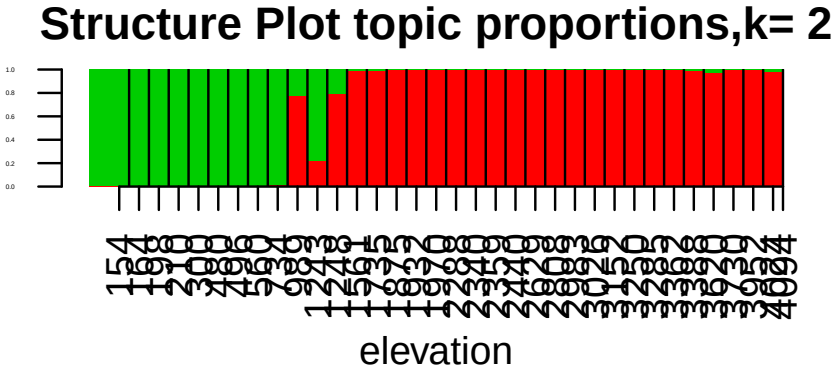
```
Topic_clus <- maptpx::topics(new_counts, K = 2)

##
## Estimating on a 35 document collection.
## Fitting the 2 topic model.
## log posterior increase: 281.4, 0.1, 1, 48.3, 0.1, done.
```

```
K <- 2
par(mfrow = c(1, 1))
par(mar = c(8, 2, 2, 2))
barplot(t(Topic_clus$omega[order(elevation_metadata),
    ]), col = 2:(K + 1), axisnames = F, space = 0,
    border = NA, main = "", las = 1, ylim = c(0,
        1), cex.axis = 0.2, cex.main = 1.4, xlab = "elevation")
title(main = paste("Structure Plot topic proportions,k=",
    K))
combo_patch_dir = paste0(round(elevation_metadata))
combo_patch_dir_ordered = combo_patch_dir[order(elevation_metadata)]

match_labs = match(unique(combo_patch_dir_ordered),
    combo_patch_dir_ordered)
match_labs_suffix = c(match_labs[2:length(unique(combo_patch_dir_ordered))],
    35)
match_labs_prefix = match_labs[1:(length(unique(combo_patch_dir_ordered)))]
labs = match_labs_prefix + 0.5 * (match_labs_suffix -
    match_labs_prefix)

axis(1, at = labs, unique(combo_patch_dir_ordered),
    las = 2)
abline(v = match_labs[2:length(match_labs)])
```



We perform the topic model for $K = 3$.

```
Topic_clus <- maptpx::topics(new_counts, K = 3)
```

```
##
## Estimating on a 35 document collection.
## Fitting the 3 topic model.
## log posterior increase: 411.3, 18.8, 2.9, 4.7, 9, 15, 10.9, 23.9, 0.3, 2.1, 1.3, done.
```

```
docweights <- Topic_clus$omega
par(mfrow = c(1, 1))
par(mar = c(8, 2, 2, 2))
east_west_elevation = paste0(metadata$WorE, "_",
    metadata$Elevation)

index1 <- which(metadata$WorE == "E")
index2 <- which(metadata$WorE == "W")
elevation1 <- metadata$Elevation[index1]
elevation2 <- metadata$Elevation[index2]
index_WE <- c(index1[order(elevation1)], index2[order(elevation2)])

K <- 3
barplot(t(docweights[index_WE, ]), col = 2:(K +
    1), axisnames = F, space = 0, border = NA,
    main = paste("No. of clusters=", 3), las = 1,
    ylim = c(0, 1), cex.axis = 0.2, cex.main = 1.4)

combo_patch_dir = paste0(east_west_elevation)
combo_patch_dir_ordered = combo_patch_dir[index_WE]

match_labs = match(unique(combo_patch_dir_ordered),
    combo_patch_dir_ordered)
match_labs_suffix = c(match_labs[2:length(unique(combo_patch_dir_ordered))],
    35)
match_labs_prefix = match_labs[1:(length(unique(combo_patch_dir_ordered)))]
labs = match_labs_prefix - 0.5 * (match_labs_suffix -
    match_labs_prefix)

axis(1, at = labs, unique(combo_patch_dir_ordered),
    las = 2)
abline(v = match_labs[1:length(match_labs)])
```

FLASH model fitting

We first derive the varaince weights.

```
lambda_star <- 1000 * (Topic_clus$omega %*% t(Topic_clus$theta))
lambda_star[lambda_star == 0] <- 1e-04
dim(lambda_star)
```

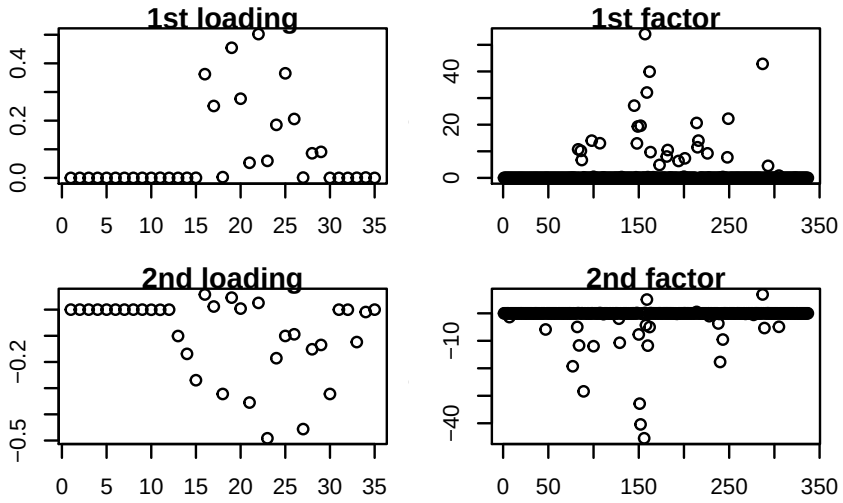
```
## [1]  35 337
```

```
g1 = suppressMessages(flash(new_counts, sigmae2_true = lambda_star))
f = g1$f
l = g1$l
res = new_counts - l %*% t(f)
# g_new = flash(res,nonnegative =
# TRUE,sigmae2_true = lambda)
g2 = suppressMessages(flash(res, sigmae2_true = lambda_star))
l = g2$l
f = g2$f
res = res - l %*% t(f)
g3 = suppressMessages(flash(res, sigmae2_true = lambda_star))
l = g3$l
f = g3$f
res = res - l %*% t(f)
g4 = suppressMessages(flash(res, sigmae2_true = lambda_star))
l = g4$l
f = g4$f
res = res - l %*% t(f)
g5 = suppressMessages(flash(res, sigmae2_true = lambda_star))
```

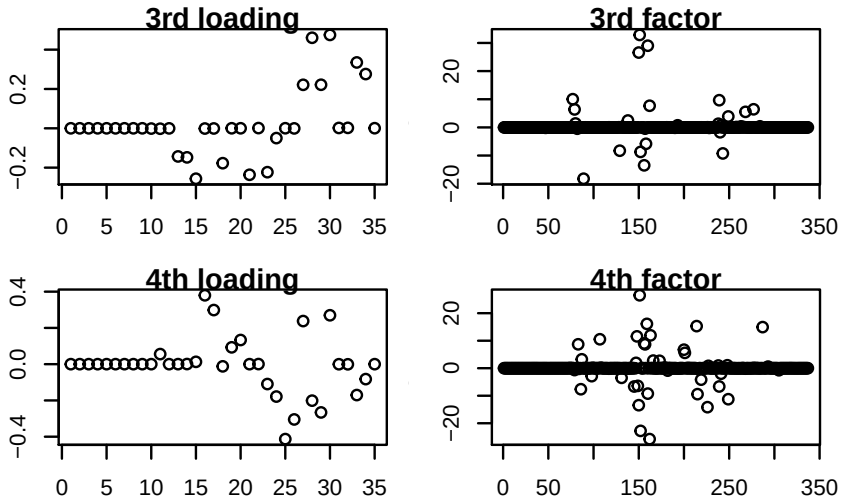
```
l = g5$l
f = g5$f
res = res - l %*% t(f)
g6 = suppressMessages(flash(res, sigmae2_true = lambda_star))
l = g6$l
f = g6$f
```

We plot the factor loadings in the order of elevation as in the structure plot for the GoM model with $K = 2$.

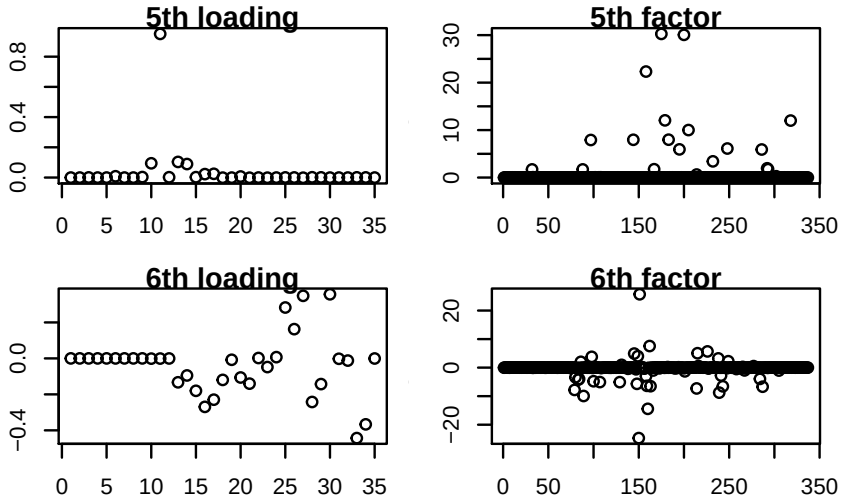
```
par(mfrow = c(2, 2))
par(cex = 0.6)
par(mar = c(3, 3, 0.8, 0.8), oma = c(1, 1, 1,
    1))
plot(g1$l[order(elevation_metadata)], main = "1st loading")
plot(g1$f, main = "1st factor")
plot(g2$l[order(elevation_metadata)], main = "2nd loading")
plot(g2$f, main = "2nd factor")
```




```
plot(g3$l[order(elevation_metadata)], main = "3rd loading")
plot(g3$f, main = "3rd factor")
plot(g4$l[order(elevation_metadata)], main = "4th loading")
plot(g4$f, main = "4th factor")
```

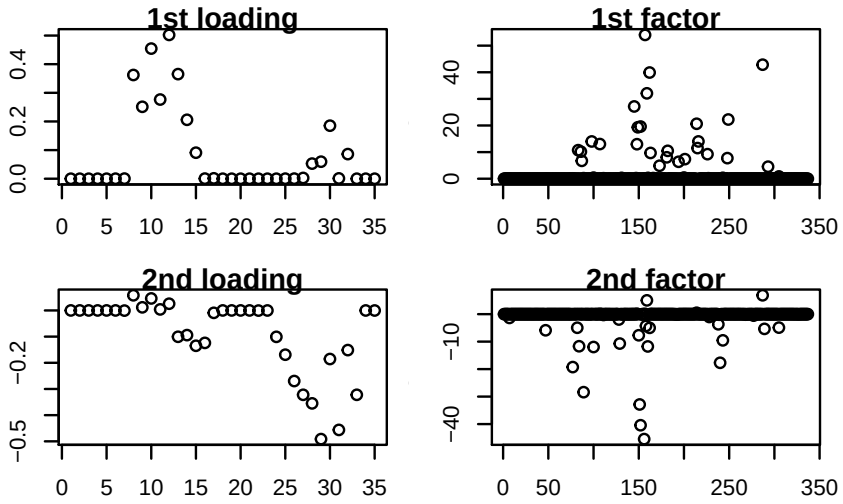



```
plot(g5$l[order(elevation_metadata)], main = "5th loading")
plot(g5$f, main = "5th factor")
plot(g6$l[order(elevation_metadata)], main = "6th loading")
plot(g6$f, main = "6th factor")
```

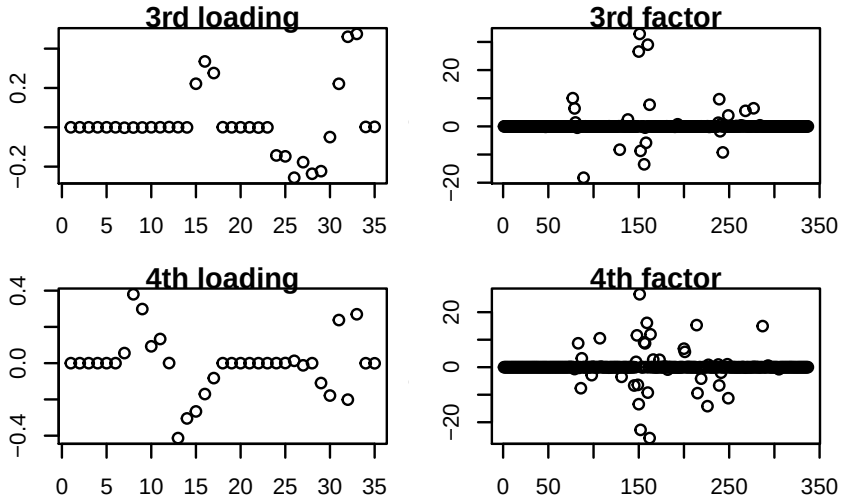


We plot the factor loadings in the order of both geography and elevation as in the structure plot for the GoM model with $K = 3$.

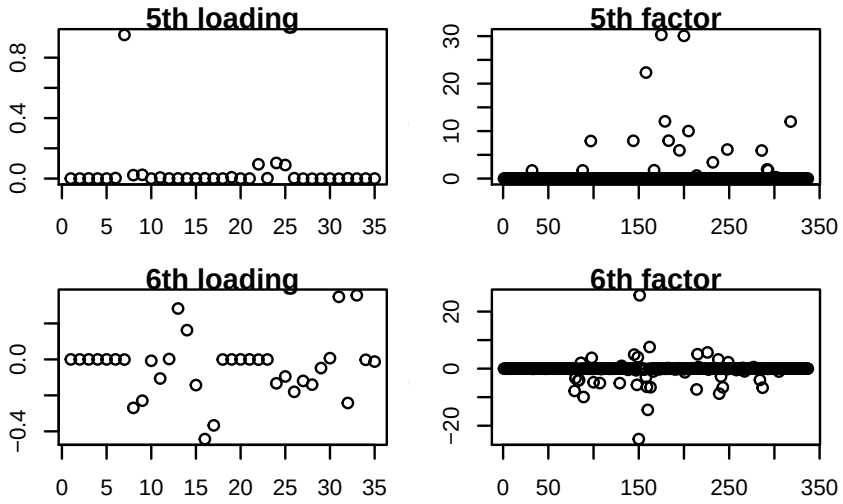
```
par(mfrow = c(2, 2))
par(cex = 0.6)
par(mar = c(3, 3, 0.8, 0.8), oma = c(1, 1, 1,
    1))
plot(g1$l[index_WE], main = "1st loading")
plot(g1$f, main = "1st factor")
plot(g2$l[index_WE], main = "2nd loading")
plot(g2$f, main = "2nd factor")
```




```
plot(g3$l[index_WE], main = "3rd loading")
plot(g3$f, main = "3rd factor")
plot(g4$l[index_WE], main = "4th loading")
plot(g4$f, main = "4th factor")
```




```
plot(g5$l[index_WE], main = "5th loading")
plot(g5$f, main = "5th factor")
plot(g6$l[index_WE], main = "6th loading")
plot(g6$f, main = "6th factor")
```



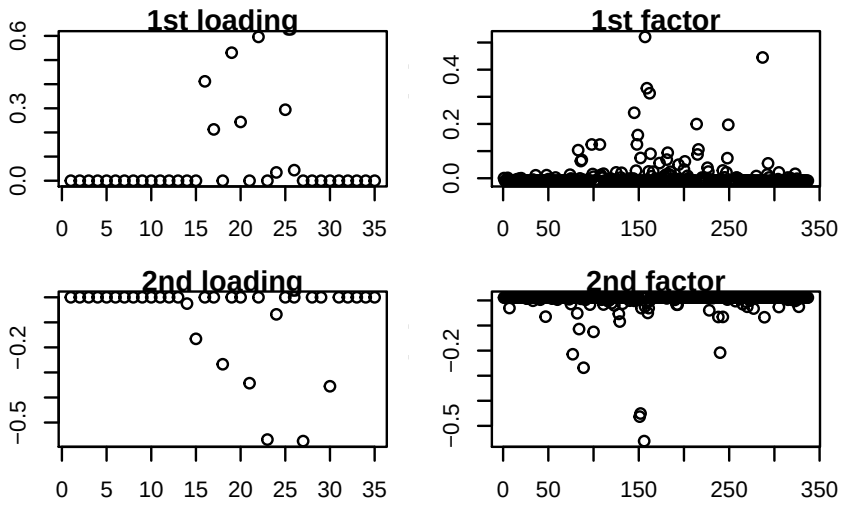
PMA model fitting

```
out <- PMD(new_counts, K = 6)
```

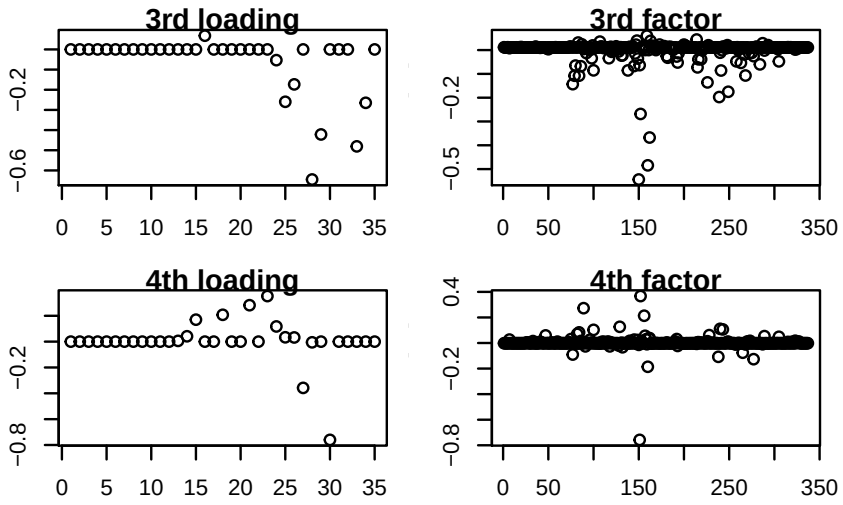
```
## 1234567891011121314151617181920
## 1234567891011121314151617181920
## 1234567891011121314151617181920
## 1234567891011121314151617181920
## 1234567891011121314151617181920
## 1234567891011121314151617181920
```

The loadings and factors ordered in the same order as the Structure plot for $K = 2$.

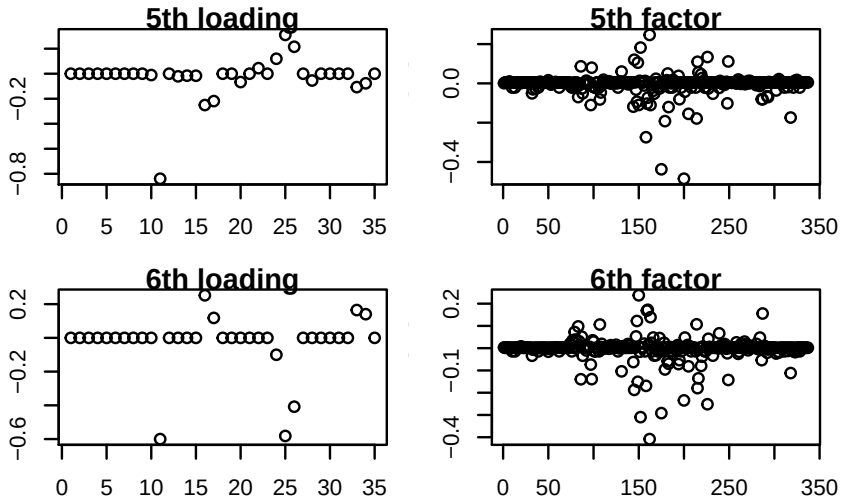
```
par(mfrow = c(2, 2))
par(cex = 0.6)
par(mar = c(3, 3, 0.8, 0.8), oma = c(1, 1, 1,
    1))
plot(out$u[order(elevation_metadata), 1], main = "1st loading")
plot(out$v[, 1], main = "1st factor")
plot(out$u[order(elevation_metadata), 2], main = "2nd loading")
plot(out$v[, 2], main = "2nd factor")
```




```
plot(out$u[order(elevation_metadata), 3], main = "3rd loading")
plot(out$v[, 3], main = "3rd factor")
plot(out$u[order(elevation_metadata), 4], main = "4th loading")
plot(out$v[, 4], main = "4th factor")
```

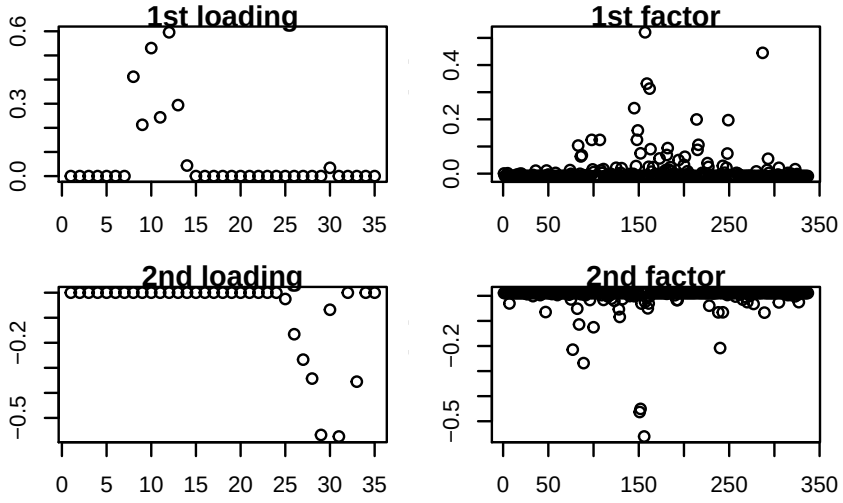



```
plot(out$u[order(elevation_metadata), 5], main = "5th loading")
plot(out$v[, 5], main = "5th factor")
plot(out$u[order(elevation_metadata), 6], main = "6th loading")
plot(out$v[, 6], main = "6th factor")
```

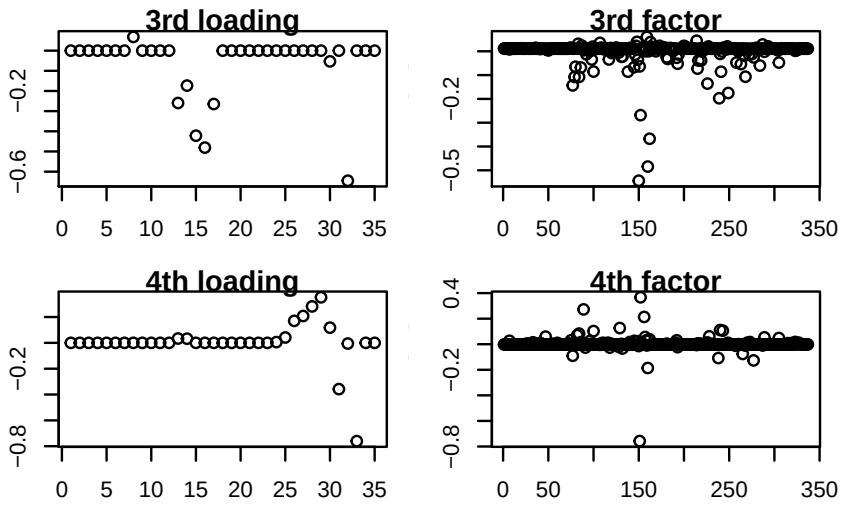


The loadings and factors ordered in the same order as the Structure plot for $K = 3$.

```
par(mfrow = c(2, 2))
par(cex = 0.6)
par(mar = c(3, 3, 0.8, 0.8), oma = c(1, 1, 1,
    1))
plot(out$u[index_WE, 1], main = "1st loading")
plot(out$v[, 1], main = "1st factor")
plot(out$u[index_WE, 2], main = "2nd loading")
plot(out$v[, 2], main = "2nd factor")
```




```
plot(out$u[index_WE, 3], main = "3rd loading")
plot(out$v[, 3], main = "3rd factor")
plot(out$u[index_WE, 4], main = "4th loading")
plot(out$v[, 4], main = "4th factor")
```




```
plot(out$u[index_WE, 5], main = "5th loading")
plot(out$v[, 5], main = "5th factor")
plot(out$u[index_WE, 6], main = "6th loading")
plot(out$v[, 6], main = "6th factor")
```