Lab 5

Viktor Sjöberg

2023-12-11

0 Load data and relevent packages

```
setwd("/Users/viktorsjoberg/Desktop/high-dimensional/Asignment 5")
data <- read.delim("T8-4.DAT", header = FALSE, sep = "")
library(FactoMineR)
library(gt)
library(ggplot2)
library(scales)
```

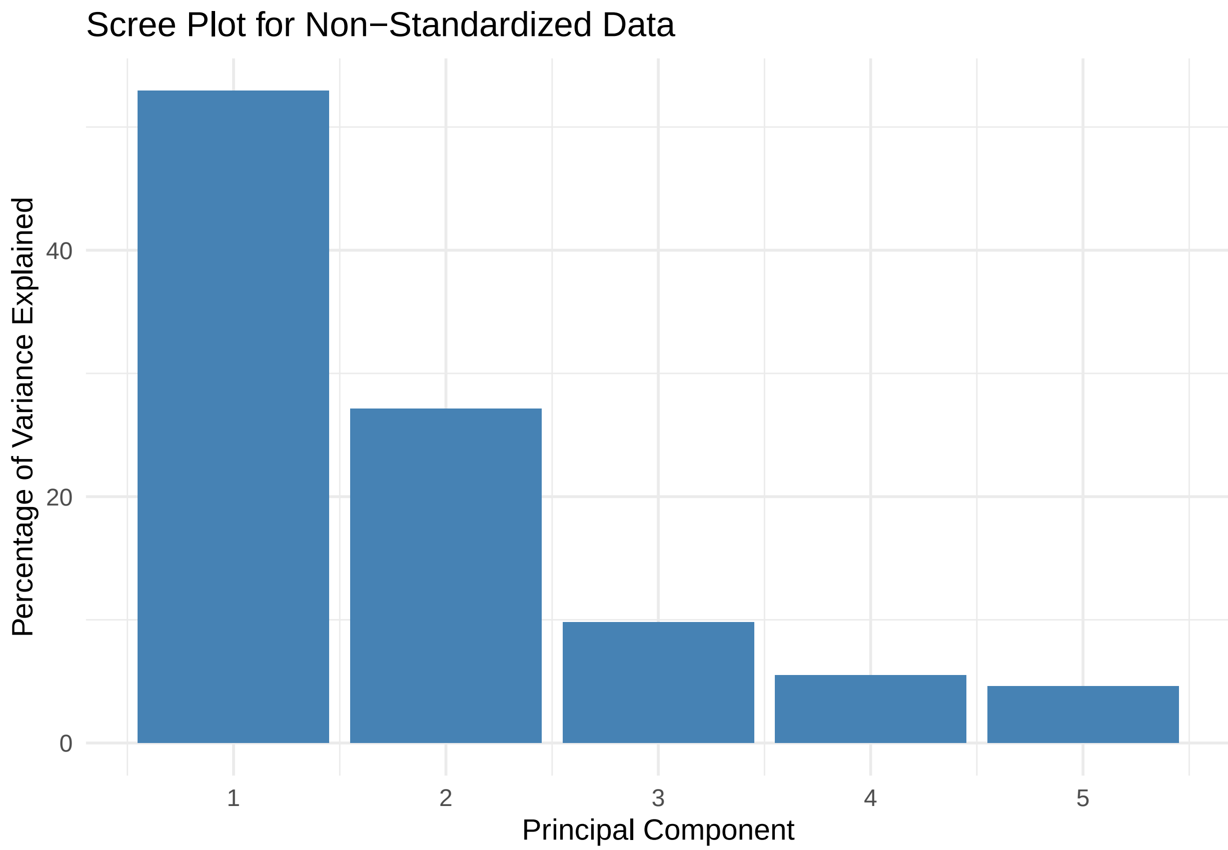
Task 1

1.1 Non Standardized Data

```
pca_results_non_std <- PCA(data, scale.unit=FALSE, graph=FALSE)
eig <- as.data.frame(pca_results_non_std$eig)
eig_gt <- gt(eig,
             caption = "Variance for Standardized Data ",
             rownames_to_stub = TRUE)
eig_gt
```

	eigenvalue	percentage of variance	cumulative percentage of variance
comp 1	0.0013543996	52.926066	52.92607
comp 2	0.0006943522	27.133298	80.05936
comp 3	0.0002513383	9.821584	89.88095
comp 4	0.0001412181	5.518400	95.39935
comp 5	0.0001177325	4.600652	100.00000

```
pca_data <- data.frame(PC = seq_along(pca_results_non_std$eig[, 'percentage of variance']),
                       Variance = pca_results_non_std$eig[, 'percentage of variance'])

ggplot(pca_data, aes(x = PC, y = Variance)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  xlab("Principal Component") +
  ylab("Percentage of Variance Explained") +
  ggtitle("Scree Plot for Non-Standardized Data") +
  theme_minimal()
```




```
corr <- pca_results_non_std$var$coord
corr_table <- as.data.frame(corr)
corr_gt <- gt(corr_table,
              caption = "Matrix of Correlations for Non-Standardized Data",
              rownames_to_stub = TRUE)
corr_gt
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
V1	0.008200363	0.016475058	0.0051700696	-0.007875911	0.001276660
V2	0.011308937	0.015030099	-0.0039569157	0.004920889	-0.006386670
V3	0.005697355	0.009077899	-0.0005967202	0.005906704	0.008466663
V4	0.023515408	-0.006533557	-0.0101859317	-0.003670450	0.001610810
V5	0.023954694	-0.008480871	0.0102392512	0.002571306	-0.001016881

In conclusion, the first three principal components appear to represent the majority of the variability in the data, capturing around 90% of the total variance. This suggests that for most purposes, we could consider only these three components in further analysis to reduce dimensionality while still retaining the core information in the dataset. The Scree Plot and cumulative variance metrics are particularly useful in supporting this decision. However, the choice of how many components to retain should also be informed by the specific context of the data and the needs of any subsequent analysis.

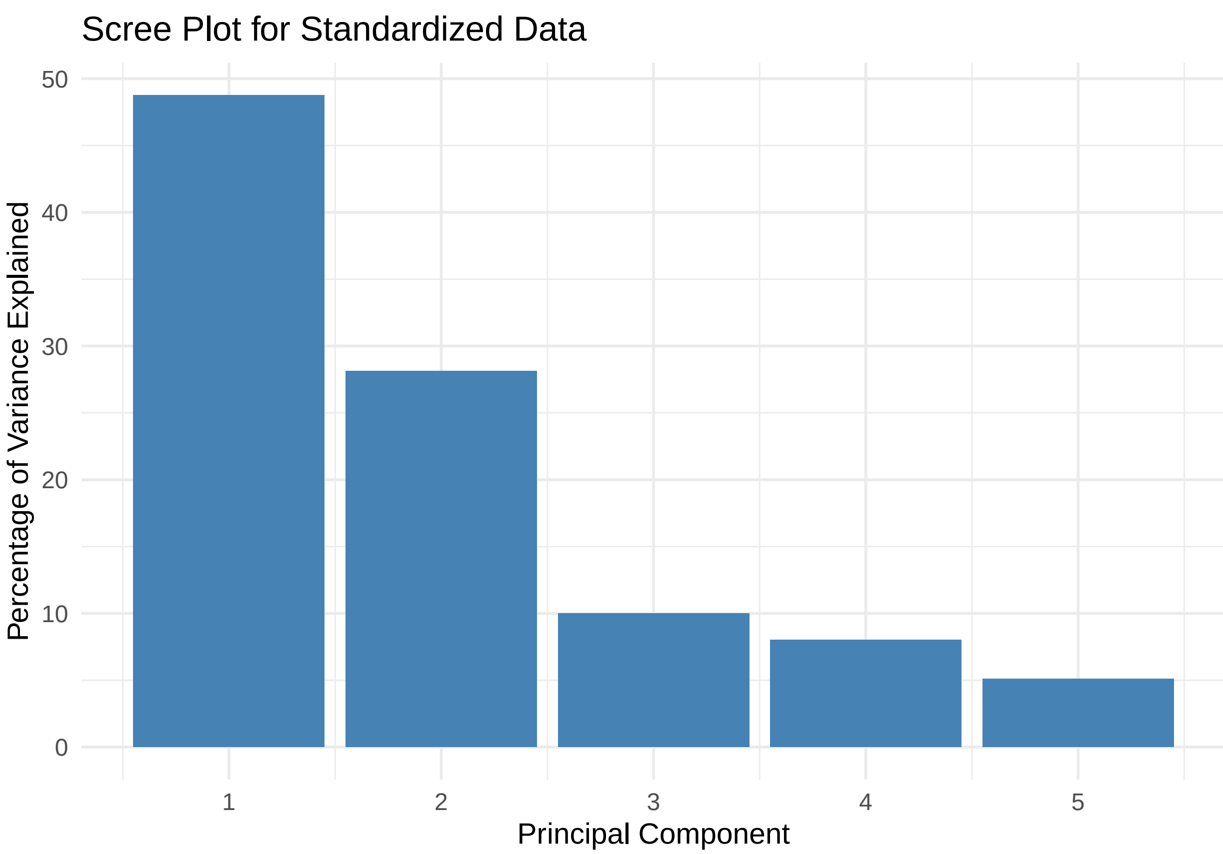
1.2 Standardized Data

```
pca_results_std <- PCA(data, scale.unit=TRUE, graph=FALSE)
```

```r
eig_std <- as.data.frame(pca_results_std$eig)
eig_std_gt <- gt(eig_std,
                 caption = "Variance for Standardized Data ",
                 rownames_to_stub = TRUE)
eig_std_gt
```

	eigenvalue	percentage of variance	cumulative percentage of variance
comp 1	2.4372731	48.745462	48.74546
comp 2	1.4070127	28.140253	76.88572
comp 3	0.5005127	10.010255	86.89597
comp 4	0.4000316	8.000632	94.89660
comp 5	0.2551699	5.103398	100.00000

```r
pca_std_data <- data.frame(PC = seq_along(pca_results_std$eig[, 'percentage of variance']),
                           Variance = pca_results_std$eig[, 'percentage of variance'])

ggplot(pca_std_data, aes(x = PC, y = Variance)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  xlab("Principal Component") +
  ylab("Percentage of Variance Explained") +
  ggtitle("Scree Plot for Standardized Data") +
  theme_minimal()
```




```r
corr_std <- pca_results_std$var$coord
corr_std_table <- as.data.frame(corr_std)
```

```r
corr_std_gt <- gt(corr_std_table,
                  caption = "Matrix of Correlations for Standardized Data",
                  rownames_to_stub = TRUE)
corr_std_gt
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
V1	0.7323218	-0.4365209	-0.42753444	0.2296048	0.19403650
V2	0.8311791	-0.2804859	-0.09629094	-0.3979617	-0.25064608
V3	0.7262022	-0.3738582	0.54604465	0.1827653	0.03595079
V4	0.6047155	0.6939569	0.06605069	-0.2411341	0.30039065
V5	0.5630885	0.7186401	-0.07699125	0.3120751	-0.25133495

In conclusion, while both non-standardized and standardized data capture a significant portion of the variance with the first few components, the standardized data seems to offer a more balanced structure with slightly stronger correlations between components and variables. Therefore, if the goal is to understand the underlying structure of the data and the variables have different scales, the standardized data would be more appropriate for PCA in this case.

2

2.1.1 Lab5HD1

Number of hidden latent variables (k): The script defines 5 hidden latent variables. Influence on a group of variables: The latent variables influence the observed variables through a factor loading matrix C. The script manipulates the loadings to create structured relationships between the latent variables and observed variables. For instance, the first latent variable has a stronger influence on the first 60 variables, as indicated by the multiplication by 5 of the first row and first 60 columns of C. Similar structured influence is defined for the other latent variables.

2.1.2 Lab5_varclust

Number of hidden latent variables (k): The script also defines 5 hidden latent variables for this simulation. Influence on a group of variables: This script sets up a more complex structure where the influence is clustered. It creates clusters of variables that are influenced by one or more latent variables but with zero influence from others. This is shown by assigning non-zero values to certain parts of C while setting other parts to zero.

2.1.3 Overall

n both scripts, the matrices F and C play a crucial role in simulating the data. F represents the latent variables (factors), and C represents how these latent variables influence the observed variables. The manipulation of C allows the simulation to mimic real-world scenarios where certain factors are known to influence specific groups of variables more than others.

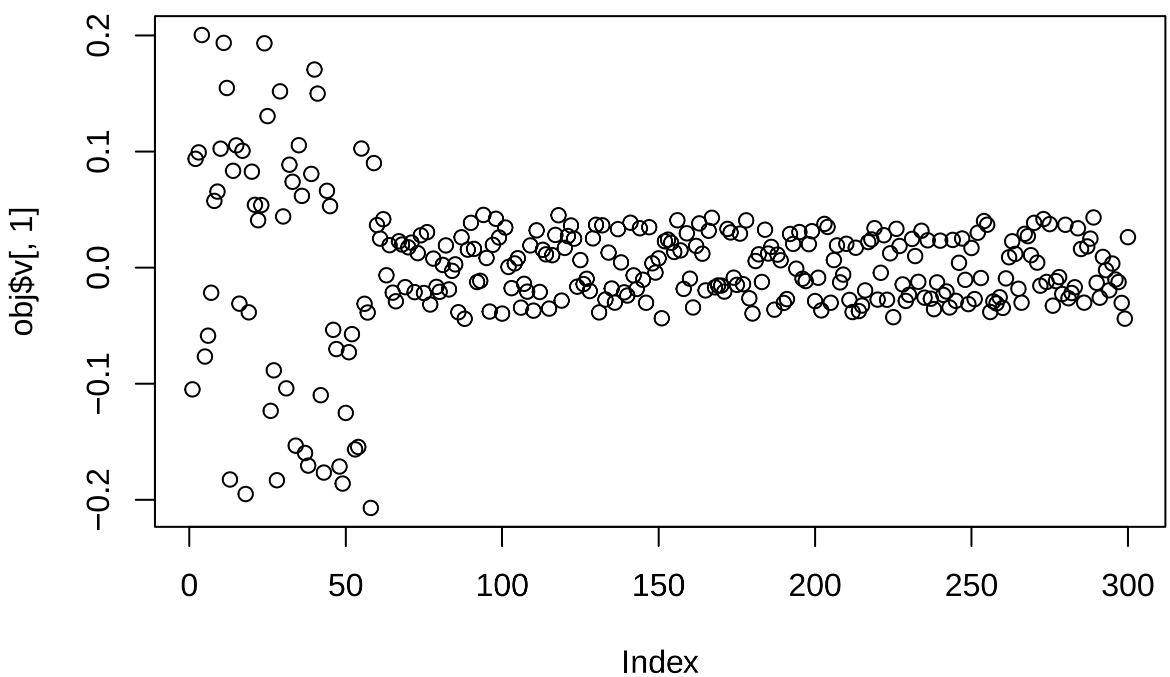
2.2.1 Lab5HD1

```r
#####SVD, number of PCs, sparse PCA


#####Data generation
```

```r
n=100;
p=300;
k=5;
F<-matrix(rnorm(n*k,0,1),nrow=n); ##factors

C1<-runif(k*p,0.1,1);
C2<-2*rbinom(k*p,1,0.5)-1;
C<-3*matrix(C1*C2,nrow=k);
C[1,1:60]<-5*C[1,1:60];
C[2,61:120]<-4*C[2,61:120];
C[3,121:180]<-3*C[3,121:180];
C[4,181:240]<-2*C[4,181:240];

M=F%*%C;

####SVD, correlation between factors and PCs

obj<-svd(M);
cor(obj$u[,1],F[,1])
```

```
## [1] -0.9990117
```

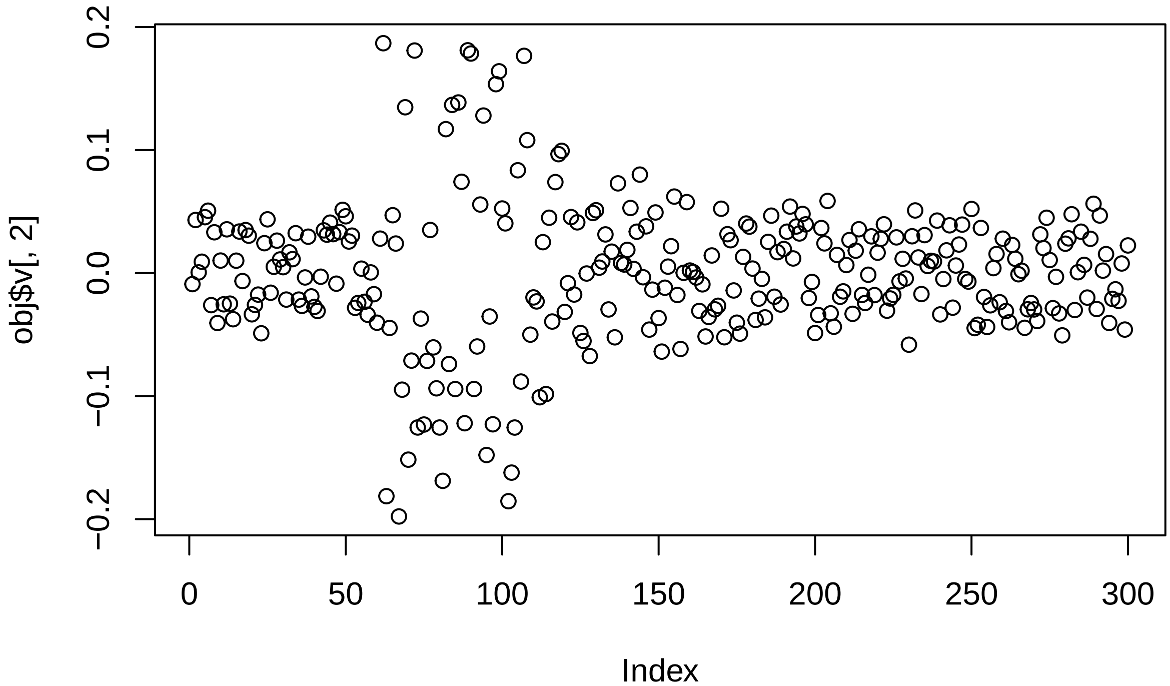
```r
plot(obj$v[,1]);
```




```r
cor(obj$u[,2],F[,2])
```

```
## [1] 0.9802821
```

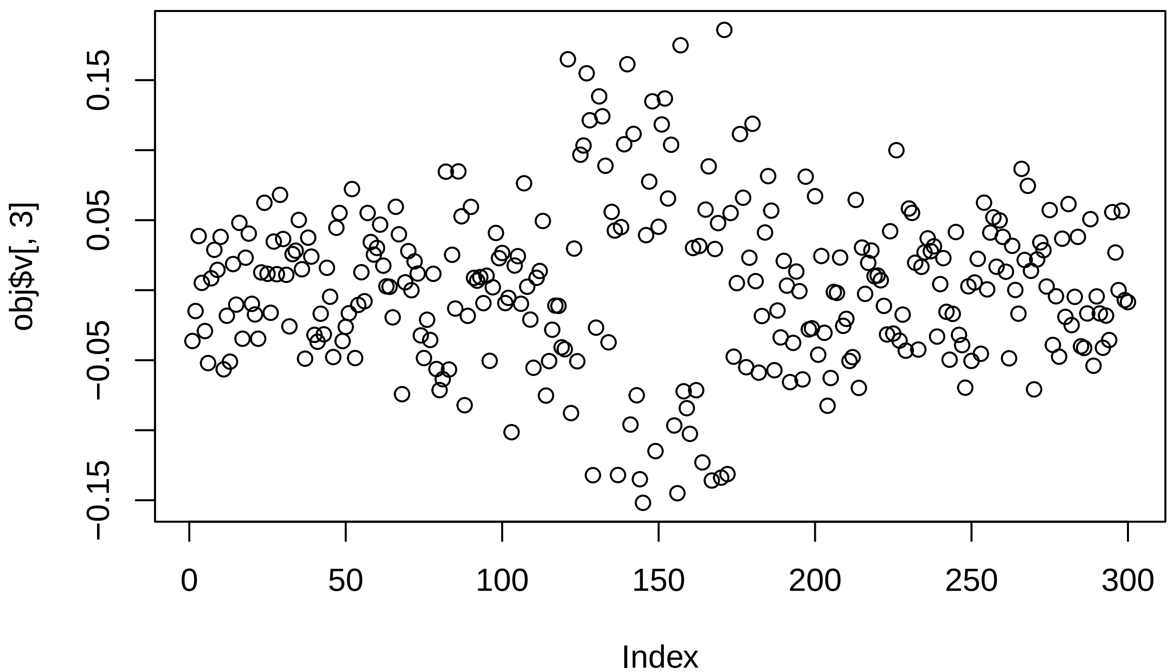
```r
plot(obj$v[,2]);
```




```r
cor(obj$u[,3], F[,3])
```

```
## [1] -0.9253647
```

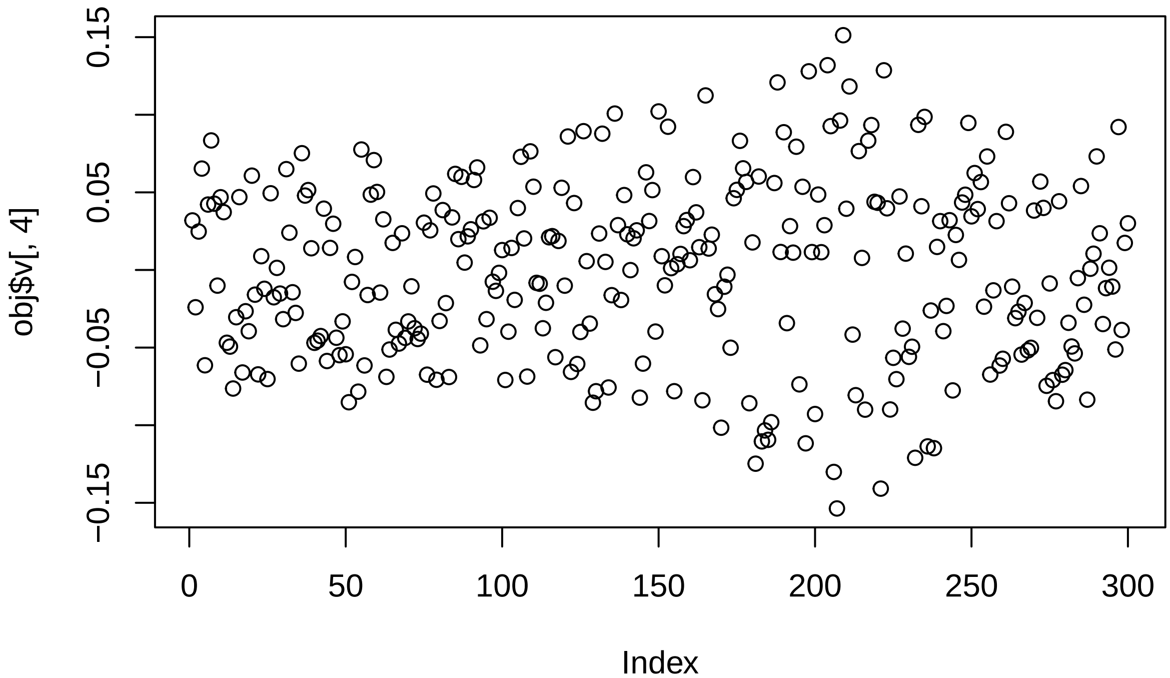
```r
plot(obj$v[,3]);
```




```r
cor(obj$u[,4], F[,4])
```

```
## [1] -0.915307
```

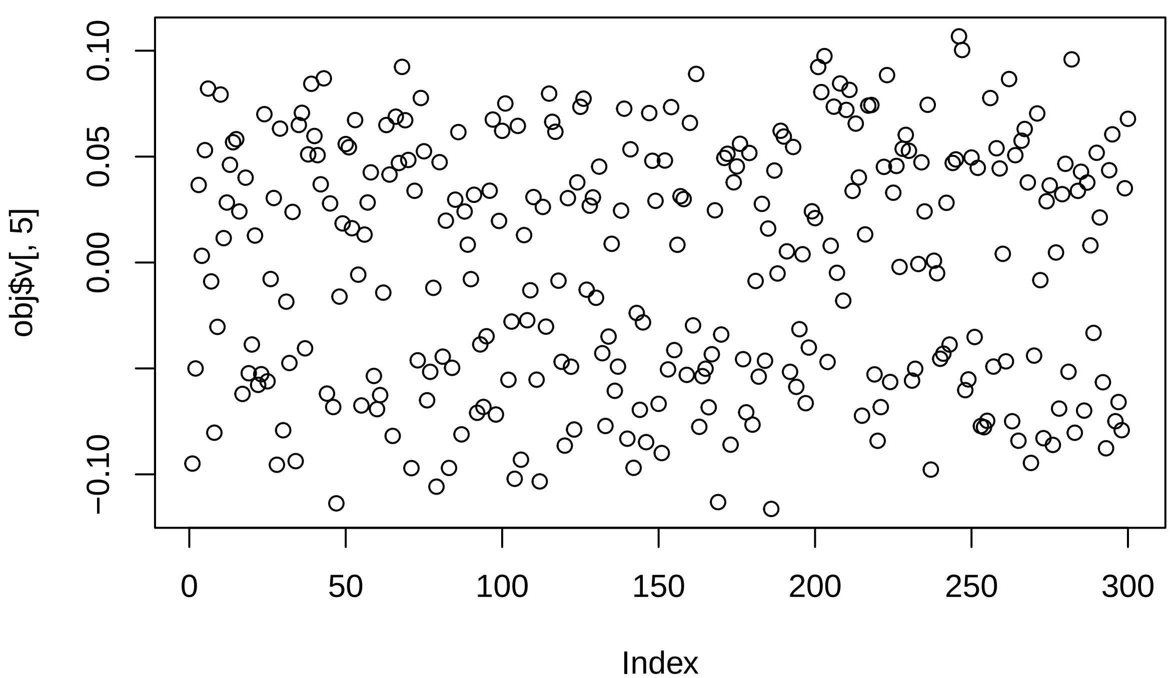
```r
plot(obj$v[,4]);
```




```
cor(obj$u[,5], F[,5])
```

```
## [1] 0.9490192
```

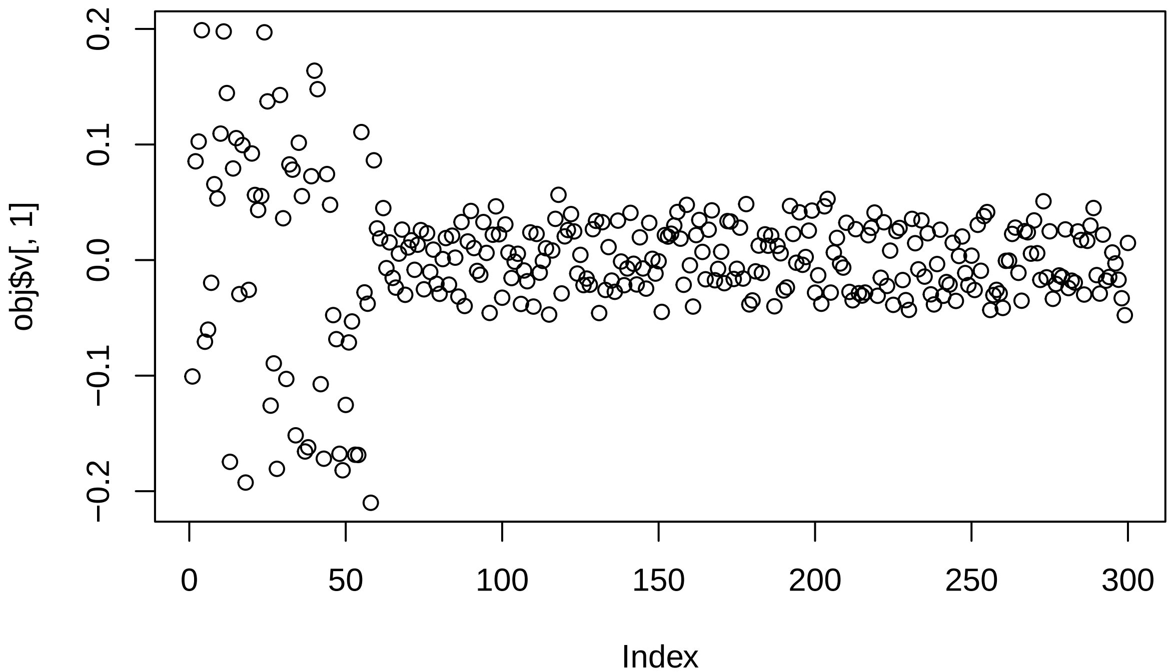
```
plot(obj$v[,5]);
```




```
####with noise

X<-M+5*matrix(rnorm(n*p,0,1), nrow=n);

obj<-svd(X);
cor(obj$u[,1],F[,1])
```

```
## [1] -0.9952225
```

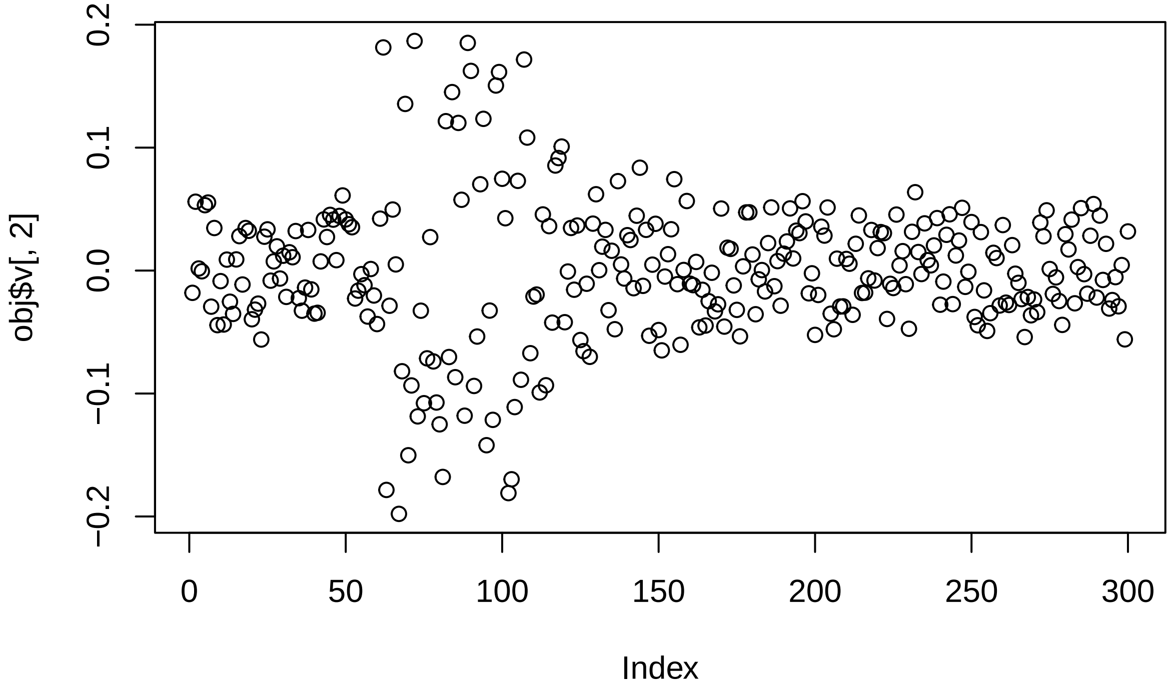
```r
plot(obj$v[,1]);
```




```r
cor(obj$u[,2],F[,2])
```

```
## [1] 0.9788553
```

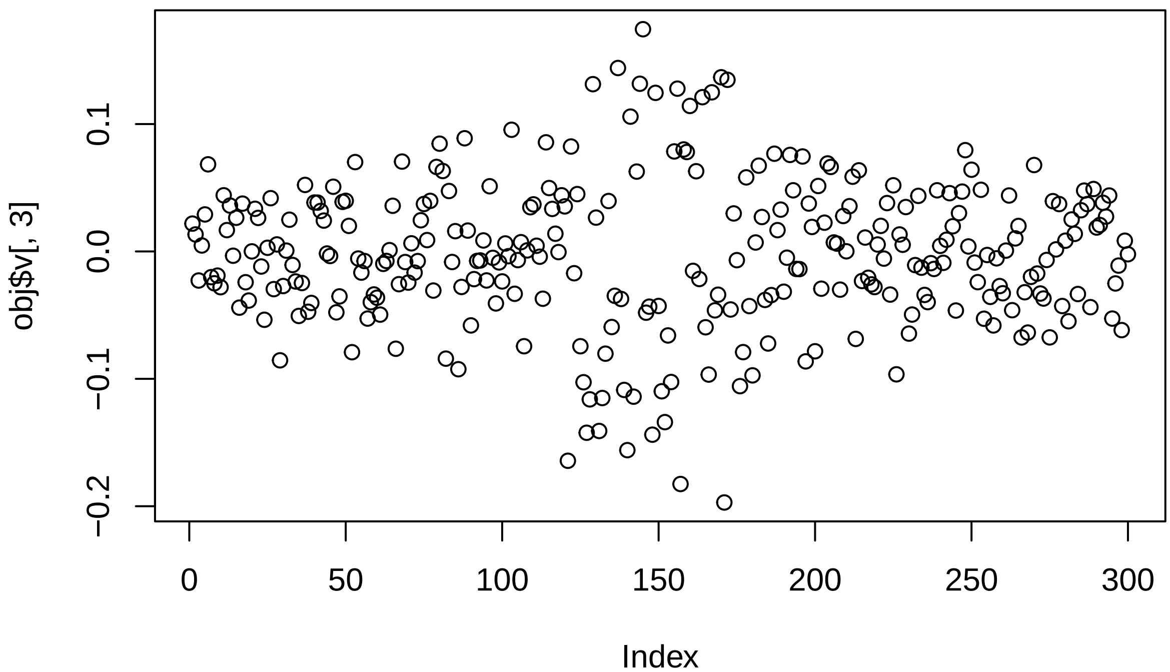
```r
plot(obj$v[,2]);
```




```r
cor(obj$u[,3], F[,3])
```

```
## [1] 0.9341411
```

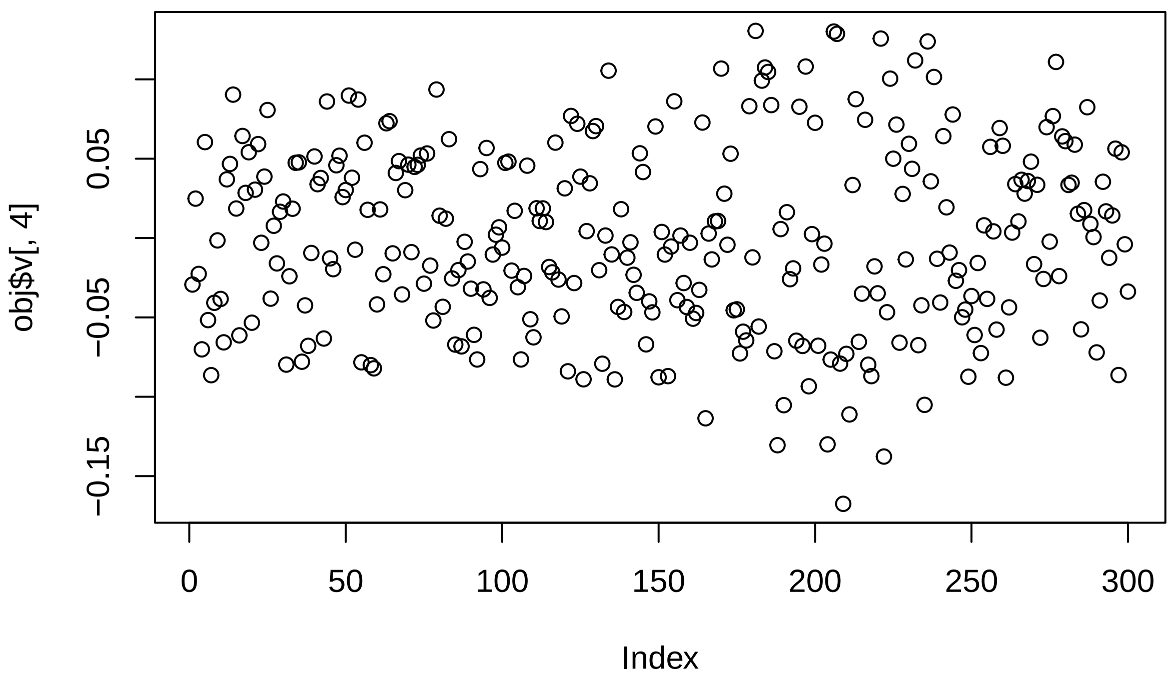
```r
plot(obj$v[,3]);
```




```r
cor(obj$u[,4], F[,4])
```

```
## [1] 0.9272089
```

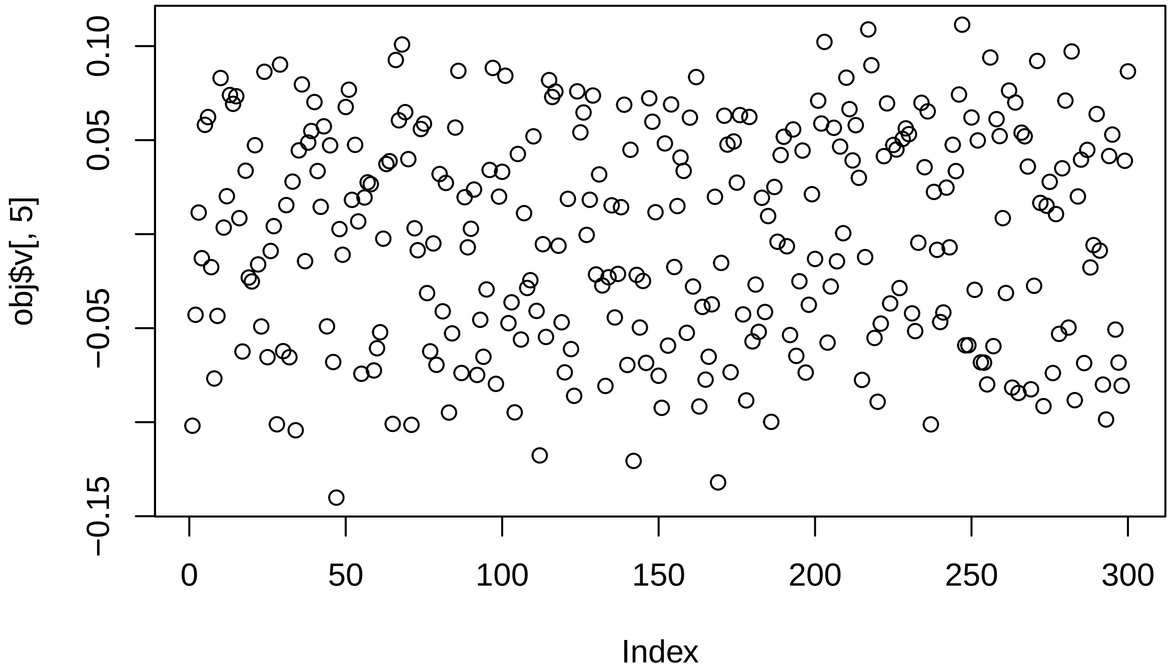
```r
plot(obj$v[,4]);
```



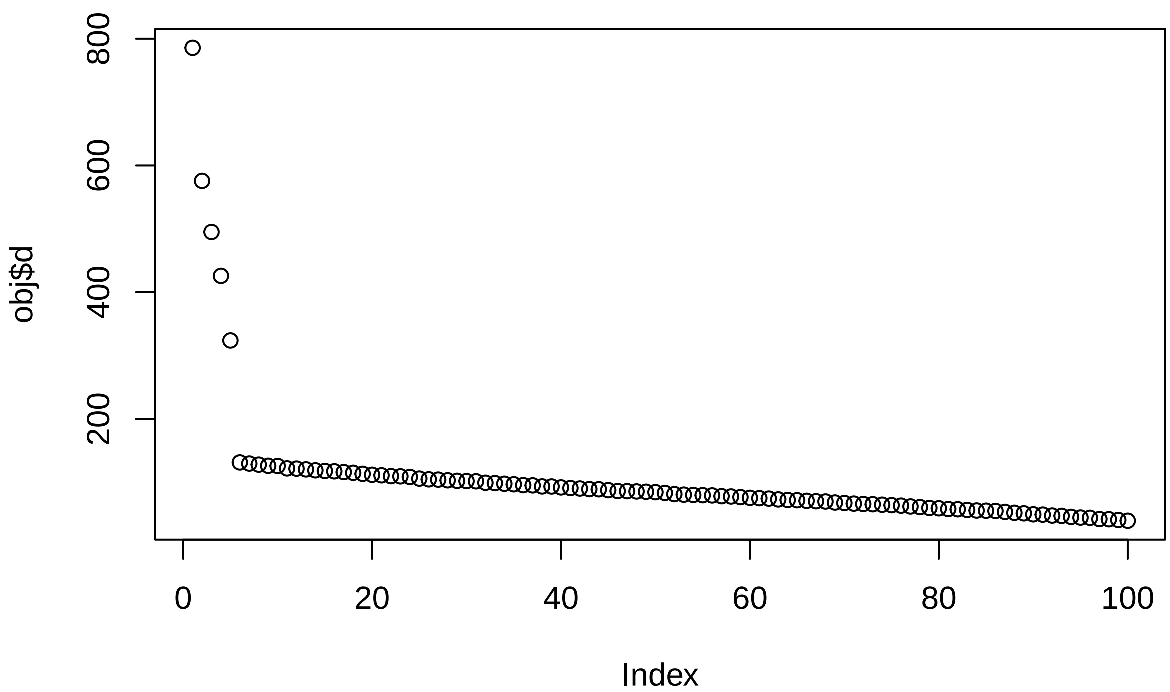

```r
cor(obj$u[,5], F[,5])
```

```
## [1] 0.9401334
```

```r
plot(obj$v[,5]);
```




```r
#########Estimating the dimension
plot(obj$d)
```



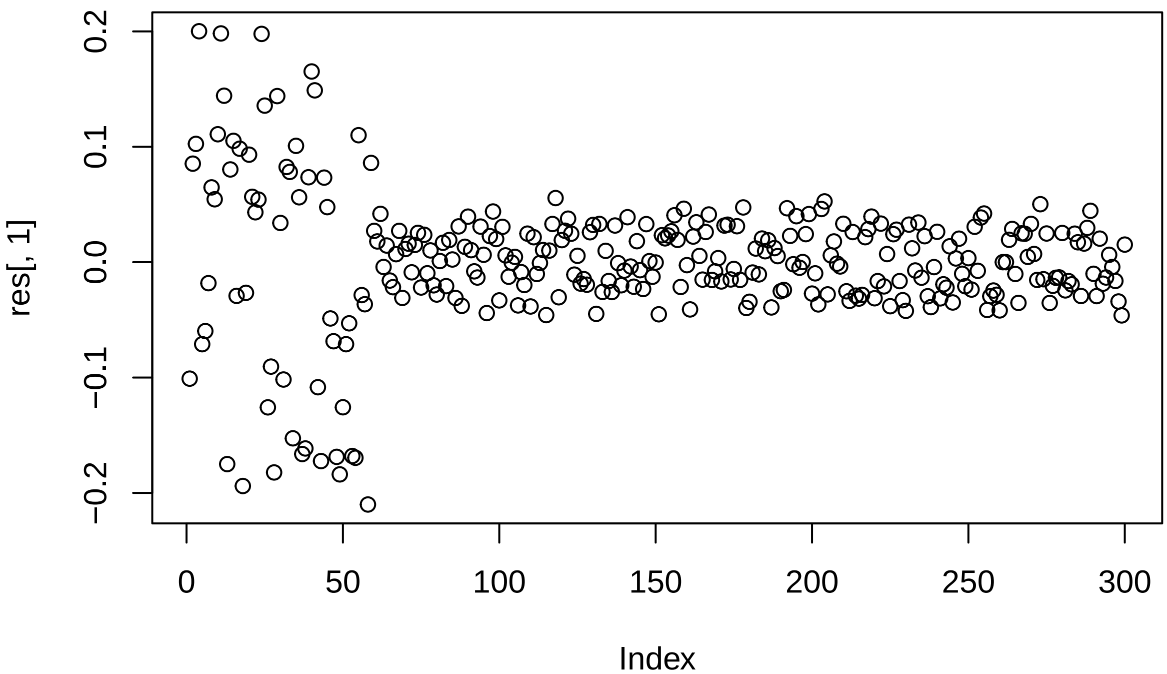

```r
library('pesel')
obj<-pesel(X);
obj$nPCs
```

```
## [1] 5
```

```r
########sparse PCA

library('sparsepca')

obj<-spca(X,5);
```

```
## [1] "Iteration:    1, Objective: 3.53466e+05, Relative improvement Inf"
```

```r
res<-obj$loadings;
plot(res[,1])
```

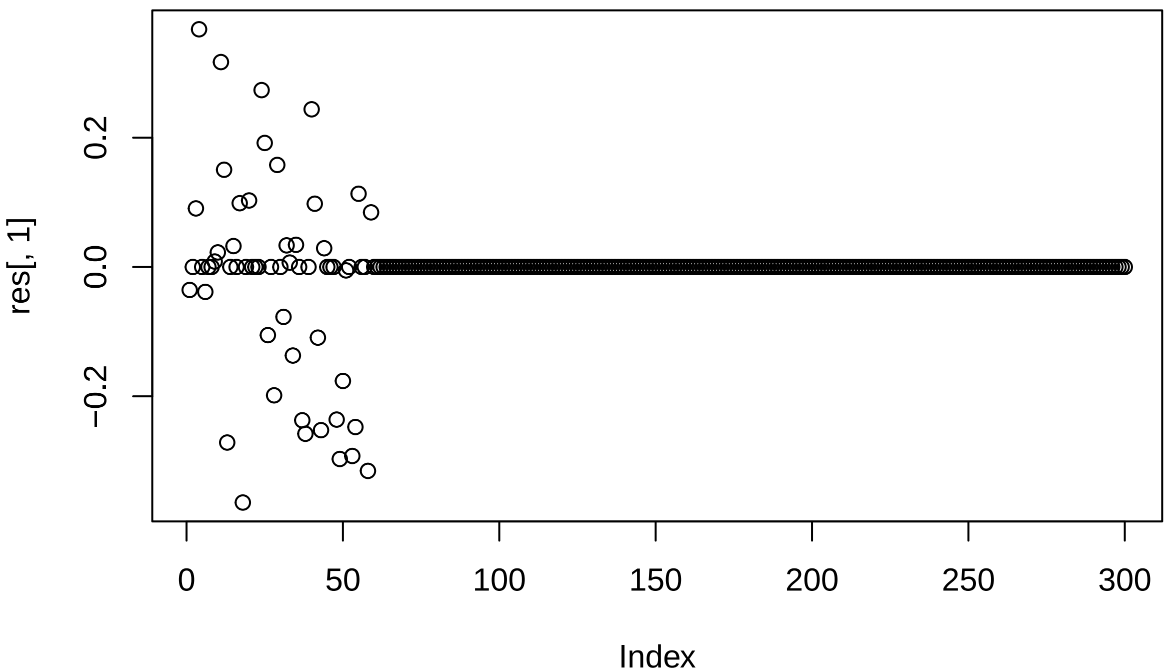



```r
obj<-spca(X,5, alpha=1e-3);
```

```
## [1] "Iteration:    1, Objective: 3.89524e+05, Relative improvement Inf"
## [1] "Iteration:   11, Objective: 3.86183e+05, Relative improvement 6.36767e-04"
## [1] "Iteration:   21, Objective: 3.84133e+05, Relative improvement 4.61889e-04"
## [1] "Iteration:   31, Objective: 3.82625e+05, Relative improvement 3.44723e-04"
## [1] "Iteration:   41, Objective: 3.81475e+05, Relative improvement 2.71036e-04"
## [1] "Iteration:   51, Objective: 3.80591e+05, Relative improvement 2.02727e-04"
## [1] "Iteration:   61, Objective: 3.79896e+05, Relative improvement 1.65373e-04"
## [1] "Iteration:   71, Objective: 3.79344e+05, Relative improvement 1.29951e-04"
## [1] "Iteration:   81, Objective: 3.78898e+05, Relative improvement 1.05947e-04"
## [1] "Iteration:   91, Objective: 3.78539e+05, Relative improvement 8.54578e-05"
## [1] "Iteration:  101, Objective: 3.78244e+05, Relative improvement 7.26845e-05"
## [1] "Iteration:  111, Objective: 3.77991e+05, Relative improvement 6.11048e-05"
## [1] "Iteration:  121, Objective: 3.77774e+05, Relative improvement 5.47429e-05"
## [1] "Iteration:  131, Objective: 3.77579e+05, Relative improvement 4.92149e-05"
## [1] "Iteration:  141, Objective: 3.77409e+05, Relative improvement 4.15286e-05"
## [1] "Iteration:  151, Objective: 3.77259e+05, Relative improvement 3.81045e-05"
## [1] "Iteration:  161, Objective: 3.77123e+05, Relative improvement 3.41350e-05"
## [1] "Iteration:  171, Objective: 3.77001e+05, Relative improvement 3.13828e-05"
## [1] "Iteration:  181, Objective: 3.76887e+05, Relative improvement 2.92714e-05"
## [1] "Iteration:  191, Objective: 3.76780e+05, Relative improvement 2.73421e-05"
```

```
## [1] "Iteration:  201, Objective: 3.76681e+05, Relative improvement 2.58583e-05"
## [1] "Iteration:  211, Objective: 3.76587e+05, Relative improvement 2.44109e-05"
## [1] "Iteration:  221, Objective: 3.76499e+05, Relative improvement 2.25532e-05"
## [1] "Iteration:  231, Objective: 3.76417e+05, Relative improvement 2.07597e-05"
## [1] "Iteration:  241, Objective: 3.76342e+05, Relative improvement 1.94050e-05"
## [1] "Iteration:  251, Objective: 3.76272e+05, Relative improvement 1.82735e-05"
## [1] "Iteration:  261, Objective: 3.76205e+05, Relative improvement 1.71571e-05"
## [1] "Iteration:  271, Objective: 3.76143e+05, Relative improvement 1.59637e-05"
## [1] "Iteration:  281, Objective: 3.76085e+05, Relative improvement 1.50480e-05"
## [1] "Iteration:  291, Objective: 3.76031e+05, Relative improvement 1.39498e-05"
## [1] "Iteration:  301, Objective: 3.75979e+05, Relative improvement 1.34212e-05"
## [1] "Iteration:  311, Objective: 3.75931e+05, Relative improvement 1.24984e-05"
## [1] "Iteration:  321, Objective: 3.75885e+05, Relative improvement 1.20324e-05"
## [1] "Iteration:  331, Objective: 3.75841e+05, Relative improvement 1.13247e-05"
## [1] "Iteration:  341, Objective: 3.75800e+05, Relative improvement 1.06503e-05"
## [1] "Iteration:  351, Objective: 3.75761e+05, Relative improvement 1.02171e-05"
```

```
res<-obj$loadings;
plot(res[,1])
```

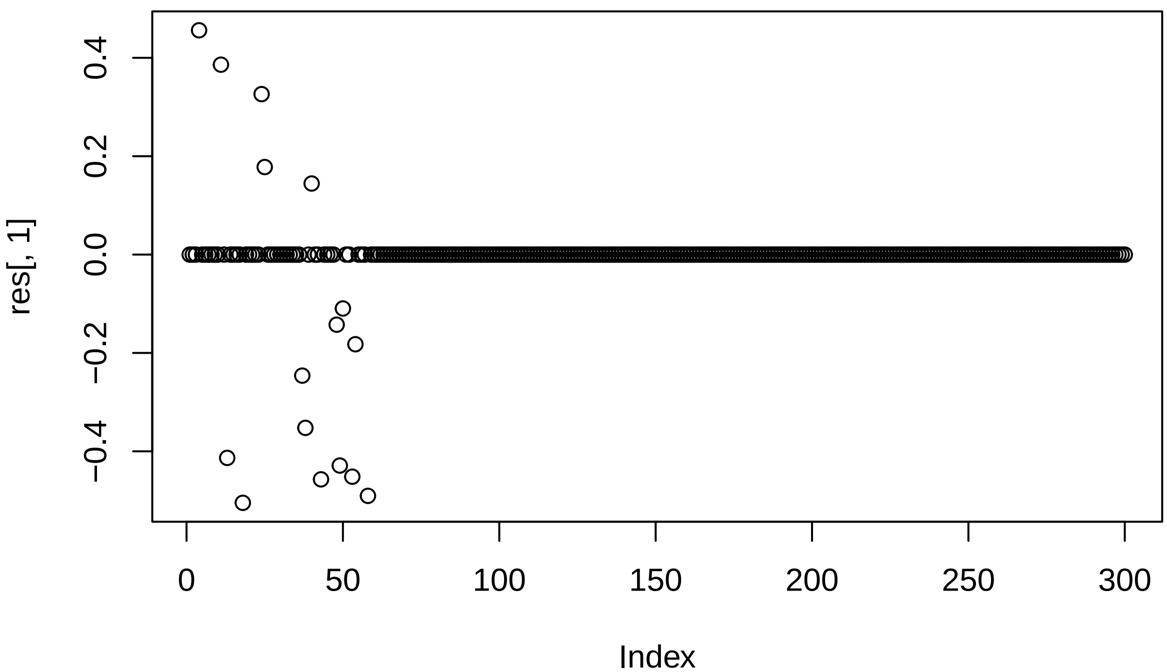



```
obj<-spca(X,5, alpha=5e-3);
```

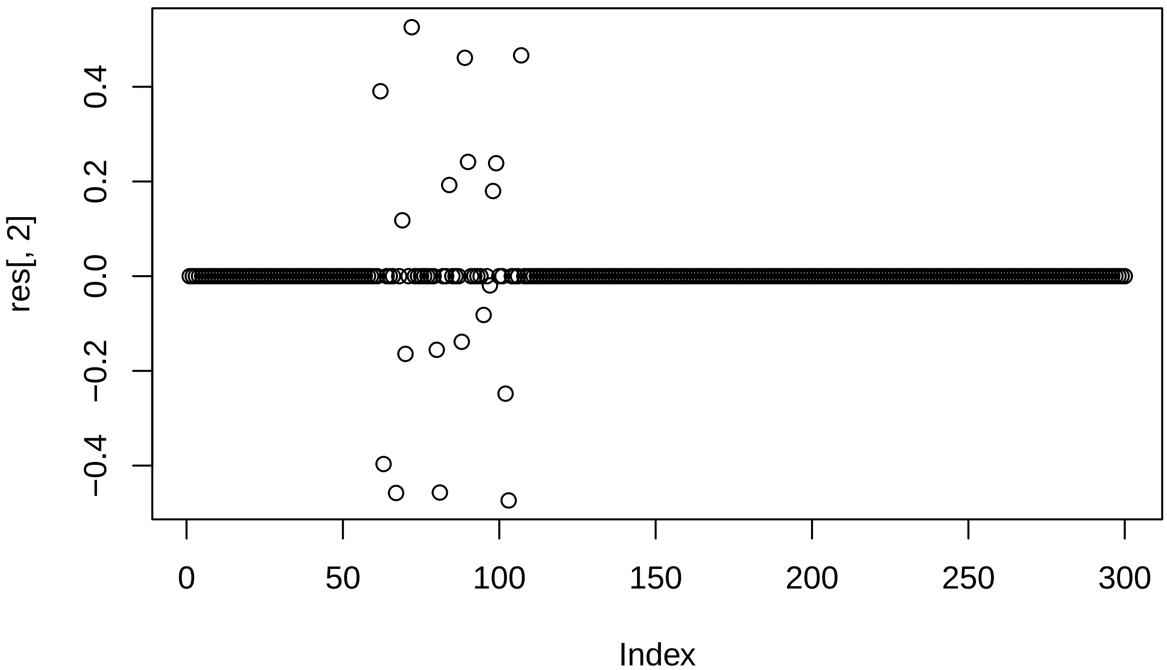
```
## [1] "Iteration:    1, Objective: 5.34940e+05, Relative improvement Inf"
## [1] "Iteration:   11, Objective: 4.85943e+05, Relative improvement 4.02094e-03"
## [1] "Iteration:   21, Objective: 4.74891e+05, Relative improvement 1.51992e-03"
## [1] "Iteration:   31, Objective: 4.69769e+05, Relative improvement 8.63179e-04"
## [1] "Iteration:   41, Objective: 4.66363e+05, Relative improvement 6.39669e-04"
## [1] "Iteration:   51, Objective: 4.63940e+05, Relative improvement 4.63147e-04"
## [1] "Iteration:   61, Objective: 4.62004e+05, Relative improvement 3.75547e-04"
## [1] "Iteration:   71, Objective: 4.60510e+05, Relative improvement 2.87252e-04"
## [1] "Iteration:   81, Objective: 4.59372e+05, Relative improvement 2.18113e-04"
## [1] "Iteration:   91, Objective: 4.58430e+05, Relative improvement 1.90716e-04"
## [1] "Iteration:  101, Objective: 4.57665e+05, Relative improvement 1.50919e-04"
## [1] "Iteration:  111, Objective: 4.57035e+05, Relative improvement 1.29168e-04"
```

```
## [1] "Iteration:  121, Objective: 4.56484e+05, Relative improvement 1.13792e-04"
## [1] "Iteration:  131, Objective: 4.56000e+05, Relative improvement 9.89102e-05"
## [1] "Iteration:  141, Objective: 4.55573e+05, Relative improvement 8.82836e-05"
## [1] "Iteration:  151, Objective: 4.55201e+05, Relative improvement 7.86192e-05"
## [1] "Iteration:  161, Objective: 4.54870e+05, Relative improvement 6.79534e-05"
## [1] "Iteration:  171, Objective: 4.54576e+05, Relative improvement 6.16199e-05"
## [1] "Iteration:  181, Objective: 4.54307e+05, Relative improvement 5.67474e-05"
## [1] "Iteration:  191, Objective: 4.54074e+05, Relative improvement 4.78138e-05"
## [1] "Iteration:  201, Objective: 4.53873e+05, Relative improvement 4.26750e-05"
## [1] "Iteration:  211, Objective: 4.53688e+05, Relative improvement 3.96597e-05"
## [1] "Iteration:  221, Objective: 4.53514e+05, Relative improvement 3.67072e-05"
## [1] "Iteration:  231, Objective: 4.53356e+05, Relative improvement 3.28553e-05"
## [1] "Iteration:  241, Objective: 4.53212e+05, Relative improvement 3.08121e-05"
## [1] "Iteration:  251, Objective: 4.53076e+05, Relative improvement 2.92972e-05"
## [1] "Iteration:  261, Objective: 4.52949e+05, Relative improvement 2.62456e-05"
## [1] "Iteration:  271, Objective: 4.52833e+05, Relative improvement 2.49470e-05"
## [1] "Iteration:  281, Objective: 4.52727e+05, Relative improvement 2.20810e-05"
## [1] "Iteration:  291, Objective: 4.52631e+05, Relative improvement 2.07645e-05"
## [1] "Iteration:  301, Objective: 4.52545e+05, Relative improvement 1.72713e-05"
## [1] "Iteration:  311, Objective: 4.52472e+05, Relative improvement 1.53824e-05"
## [1] "Iteration:  321, Objective: 4.52406e+05, Relative improvement 1.28184e-05"
## [1] "Iteration:  331, Objective: 4.52351e+05, Relative improvement 1.16353e-05"
## [1] "Iteration:  341, Objective: 4.52300e+05, Relative improvement 1.08684e-05"
## [1] "Iteration:  351, Objective: 4.52253e+05, Relative improvement 1.02292e-05"
```

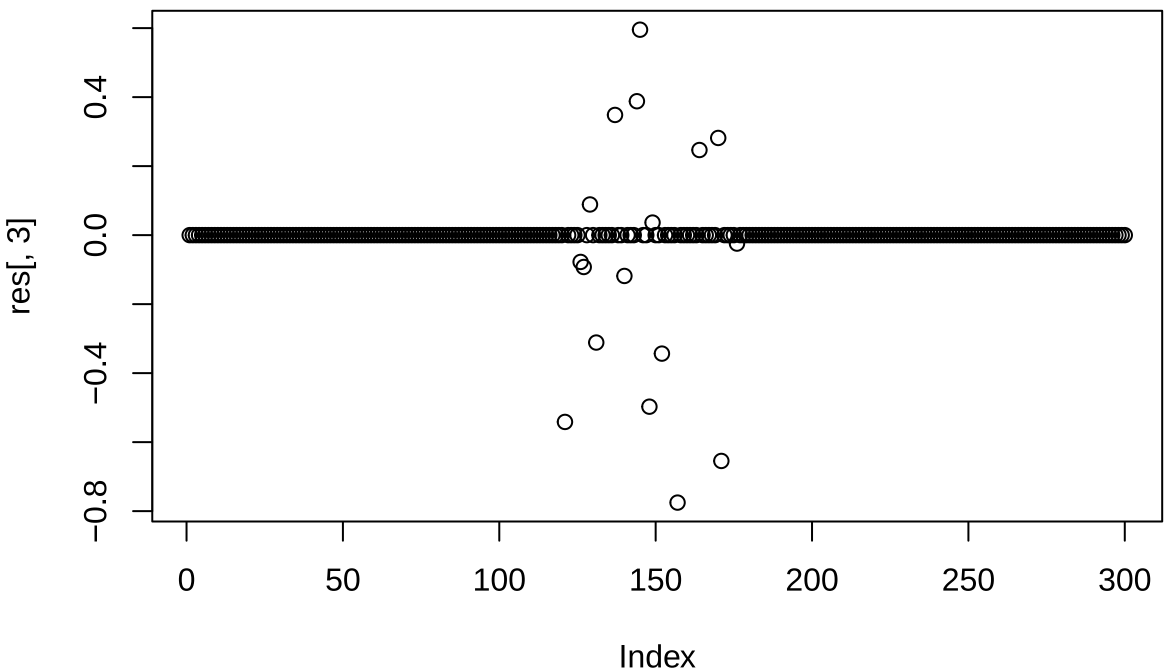
```r
res<-obj$loadings;
plot(res[,1])
```



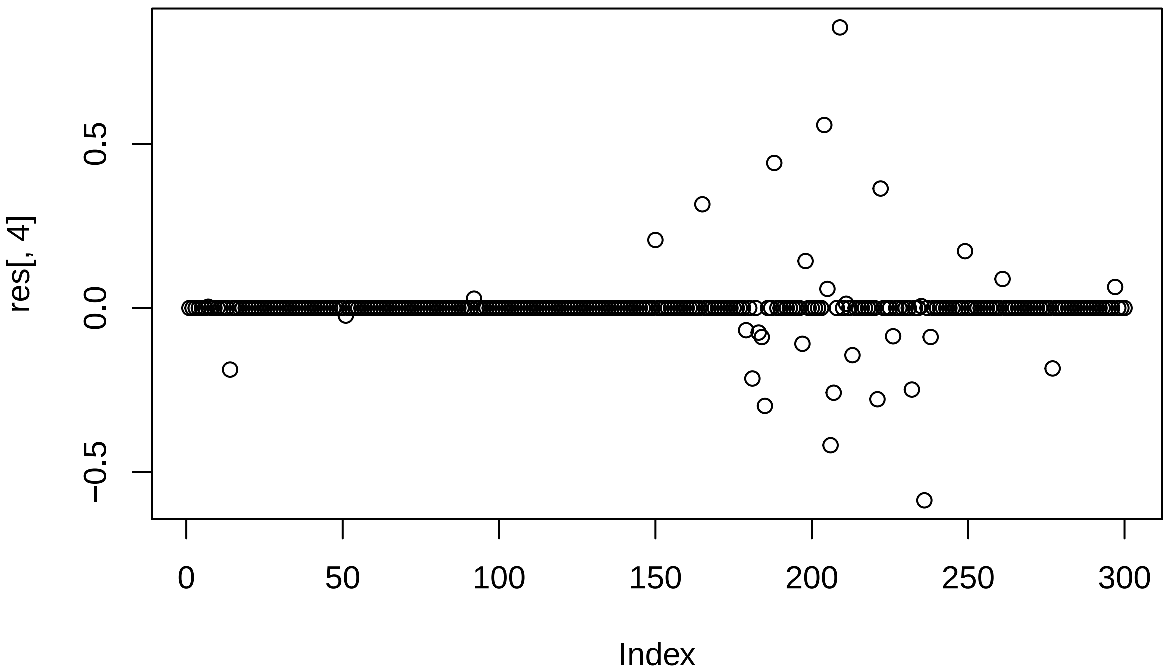

```r
plot(res[,2])
```



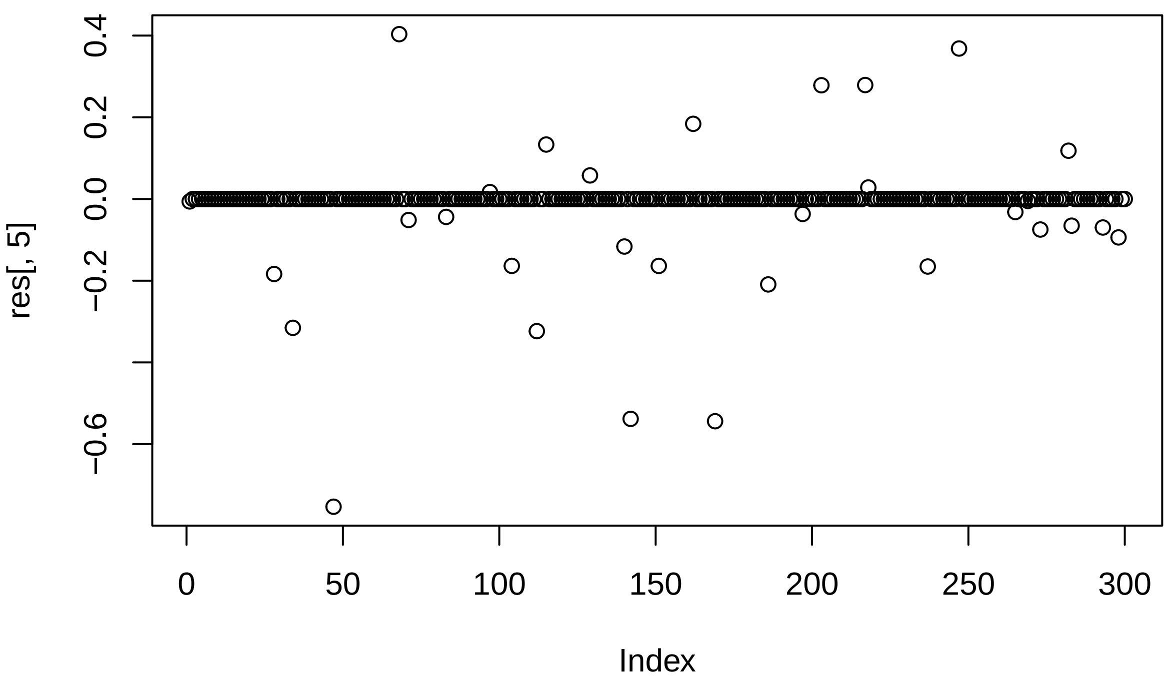

```r
plot(res[,3])
```




```r
plot(res[,4])
```




```
plot(res[,5])
```



2.2.2 Lab_5varclust

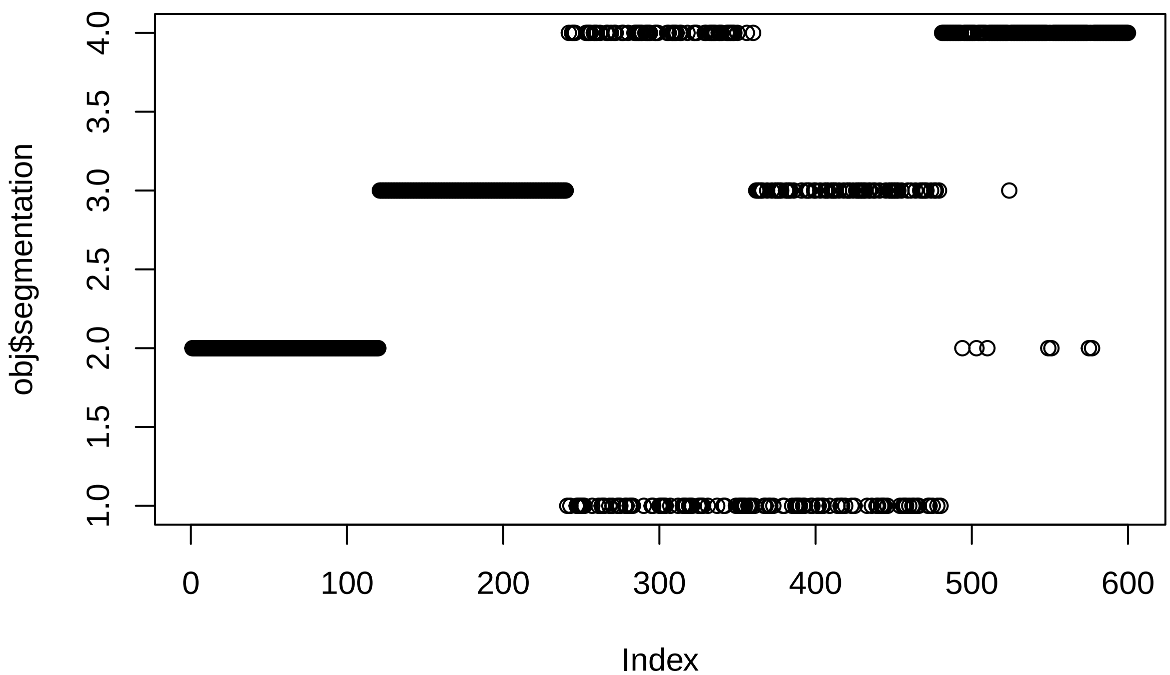
```
n=100;
p=600;
k=5;
F<-matrix(rnorm(n*k,0,1),nrow=n);
C1<-2*rbinom(k*p,1,0.5)-1;
C<-matrix(runif(k*p,0.1,1)*C1,nrow=k);
```

```r
C[1,1:200]<-5*C[1,1:200];
C[1,201:p]<-0;
C[2,201:400]<-4*C[2,201:400];
C[2,1:200]=C[2,401:600]=0;
C[3,201:400]<-3*C[3,201:400];
C[3,1:200]=C[3,401:600]=0;
C[4,401:600]=2*C[4,401:600];
C[4,1:400]=C[5,1:400]=0;

####three separate clusters of dimensions 1,2,2

M=F%*%C;


X<-M+matrix(rnorm(n*p,0,1), nrow=n);

library('varclust')
obj<-mlcc.bic(X,numb.clusters=1:5)

#####more difficult example: 5 clusters with shared factors

C1<-2*rbinom(k*p,1,0.5)-1;
C<-3*matrix(runif(k*p,0.1,1)*C1,nrow=k);

C[1,1:120]<-5*C[1,1:120];
C[1,121:p]<-0;
C[2,121:240]<-4*C[2,121:240];
C[2,361:480]<-3*C[2,361:480];
C[2,1:120]=C[2,241:360]=C[2,481:p]=0;
C[3,121:240]<-4*C[3,121:240];
C[3,481:p]<-2*C[3,481:p];
C[3,1:120]=C[3, 241:360]=C[3,361:480]=0
C[4,241:480]=3*C[4,241:480];
C[4,1:120]=C[4,121:240]=C[4,481:p]=0;
C[5,1:120]=C[5,121:240]=C[5,361:480]=0;

####group1:F1, group2:F2,F3,  group3:F5,F4, group4:F2,F4  group 5:F3,F5

M=F%*%C;

X<-M+matrix(rnorm(n*p,0,1), nrow=n);

library('varclust')
obj<-mlcc.bic(X,numb.clusters=3:7)
obj
```

```
## $nClusters:  4
## $segmentation:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [75] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [112] 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [149] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

```
## [186] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [223] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 4 1 4 4 4 1 1 1 1 1 1 4 4 4 4 1 4 4
## [260] 4 1 4 1 1 1 4 4 1 4 1 4 4 1 1 1 4 4 1 1 4 1 1 1 4 4 4 4 4 4 1 4 4 4 4 1 1
## [297] 4 4 4 1 1 1 1 1 4 4 1 4 4 4 4 1 4 4 1 1 1 4 1 1 1 4 4 4 1 1 1 1 4 4 1 4 4
## [334] 4 4 4 1 4 4 4 1 1 4 4 4 4 4 4 1 4 1 1 1 1 1 4 1 1 1 4 1 3 3 3 3 3 1 1 3 1
## [371] 1 3 1 3 3 3 3 3 1 1 3 3 3 3 1 3 1 1 1 1 3 1 1 3 3 3 1 1 3 3 1 1 3 1 1 3 3
## [408] 3 1 3 3 3 3 1 3 1 1 3 1 3 3 3 1 3 1 3 3 3 3 3 3 3 1 3 3 1 3 3 1 1 3 1 1 1
## [445] 3 1 3 3 3 3 3 3 3 1 3 1 1 1 3 1 3 1 1 3 1 1 3 3 3 3 3 1 1 3 1 3 3 1 3 1 4
## [482] 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4
## [519] 4 4 4 4 4 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 2 4 4 4 4
## [556] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [593] 4 4 4 4 4 4 4 4
## $BIC:  3706.287
## $subspacesDimensions:
##  3 4 4 3
```

```r
plot(obj$segmentation)
```



2.2.3 Analysis

In the section labeled "2.2.2 Lab_5varclust"the method applied is variable clustering with the varclust library in R. This method is used to identify groups of variables that are influenced by the same underlying latent factors. The output suggests that the best number of clusters for the data is 4, as indicated by the variable segmentation and the Bayesian Information Criterion (BIC) value provided.

In the "2.2.1 Lab5HD1" section, the method used is Singular Value Decomposition (SVD), which is used to decompose the data matrix and identify principal components. The correlations between the original factors and the principal components derived from SVD are listed, along with plots of the principal components, indicating how each variable contributes to the principal components.