Assignment 4

```
rm(list=ls())
library(foreign)
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(haven)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

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

```
library(dplyr)
library(ISLR)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-1
```

```
library(arm)
```

```
## Loading required package: lme4
```

```
##
## arm (Version 1.11-2, built: 2020-7-27)
```

```
## Working directory is /home/enminz/Graduate/Data-2020/Assignment4
```

```
library(coefplot)
```

```
##
## Attaching package: 'coefplot'
```

```
## The following objects are masked from 'package:arm':
##
##     coefplot, coefplot.default, invlogit
```

NAME: Enmin Zhou
DUE DATE: March 30th, 11:59pm

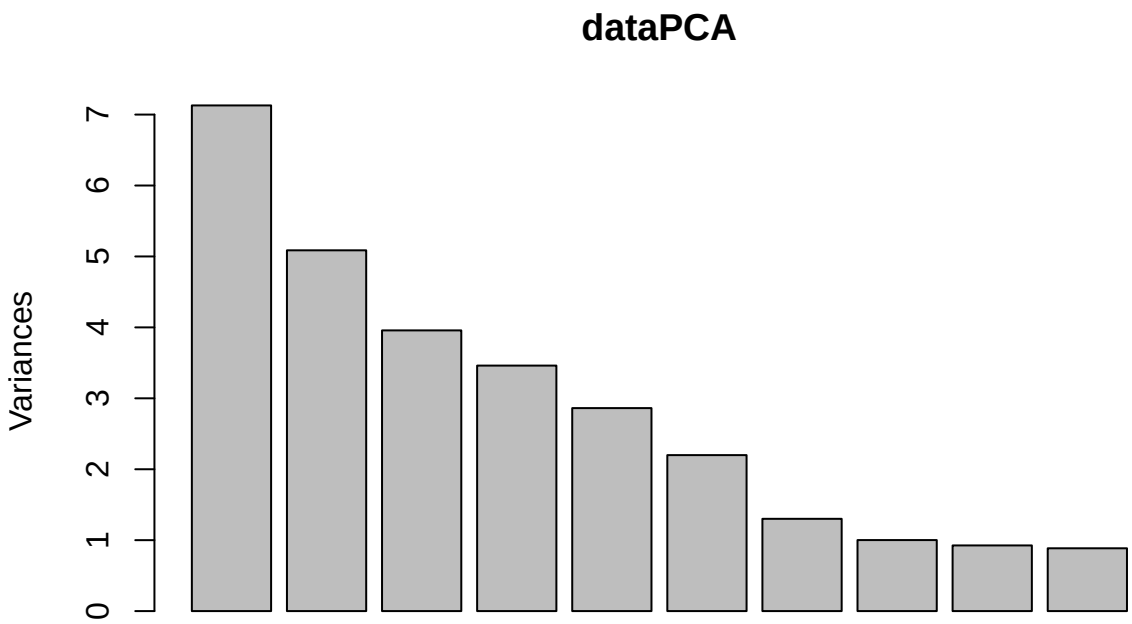
Problem 1 (100 pts)

In the folder Assignment 4, you will find the data set called data-final.csv. This data set is from the Five Personality Data Set, and it collects on-line personality test (take a look to the codebook.txt in the folder Assignment 4).

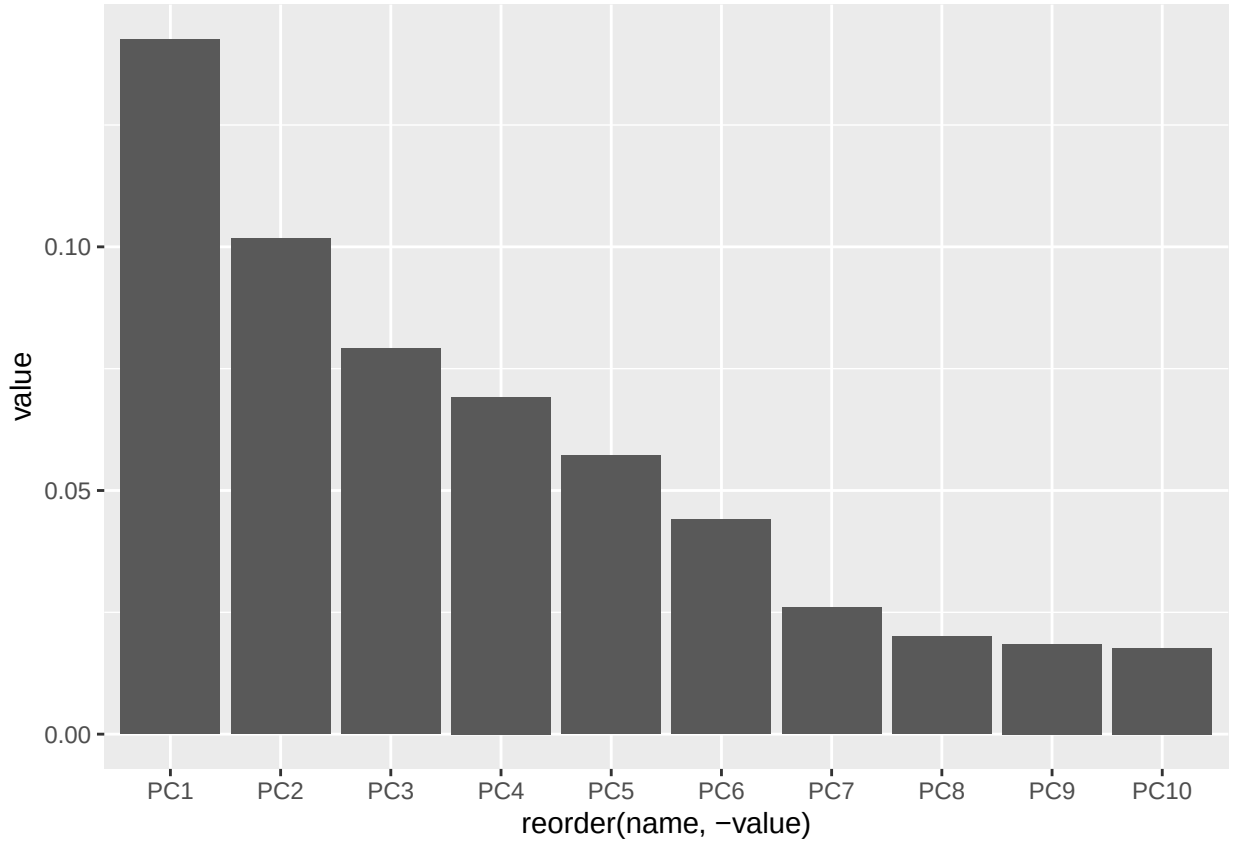
(a) (40 points) Consider the first 50 variables of this data set (this should correspond to the codebook.txt variables). Perform the Principal Component (PC) analysis after having scaled the data. How many components will you retain based on the total variance explained by each component? Plot a bar plot (in ggplot) showing the proportion of variance explained by each PC (consider just the first 10 PC). Then, plot the PC that you have chosen in an heatmap, choose your own colour in three different tonality (where one should be white). How can you interpret this plot and the PC? Is there any link with the name of the data set "the five big personalities"?

Answer: I will retain the first two PC, which are EXT1 and EXT2 because they have proportion of variance bigger than 0.1. This barplot ranks the first 10 important personalities from PCA analysis, with only 2 of theirs variance proportion higher than 0.1. There are in total 5 PC with variance proportion bigger than 0.05, which is the link with the name of the data set "the five big personalities".

```
data = read.delim("data-final.csv")
data = data[,1:50]
df <- data.matrix(data) %>% na.omit()
dataPCA <- prcomp(df, scale. = TRUE)
plot(dataPCA)
```




```
first10 <- summary(dataPCA)$importance[2,1:10]
df_10 <- enframe(first10) %>% unnest(cols=c('name', 'value'))
p1 <- ggplot(df_10, aes(x=reorder(name, -value), y=value)) + geom_bar(stat="identity")
p1
```




```
eigs_B <- dataPCA$sdev^2
eigs_B[1] / sum(eigs_B)
```

```
## [1] 0.1425808
```

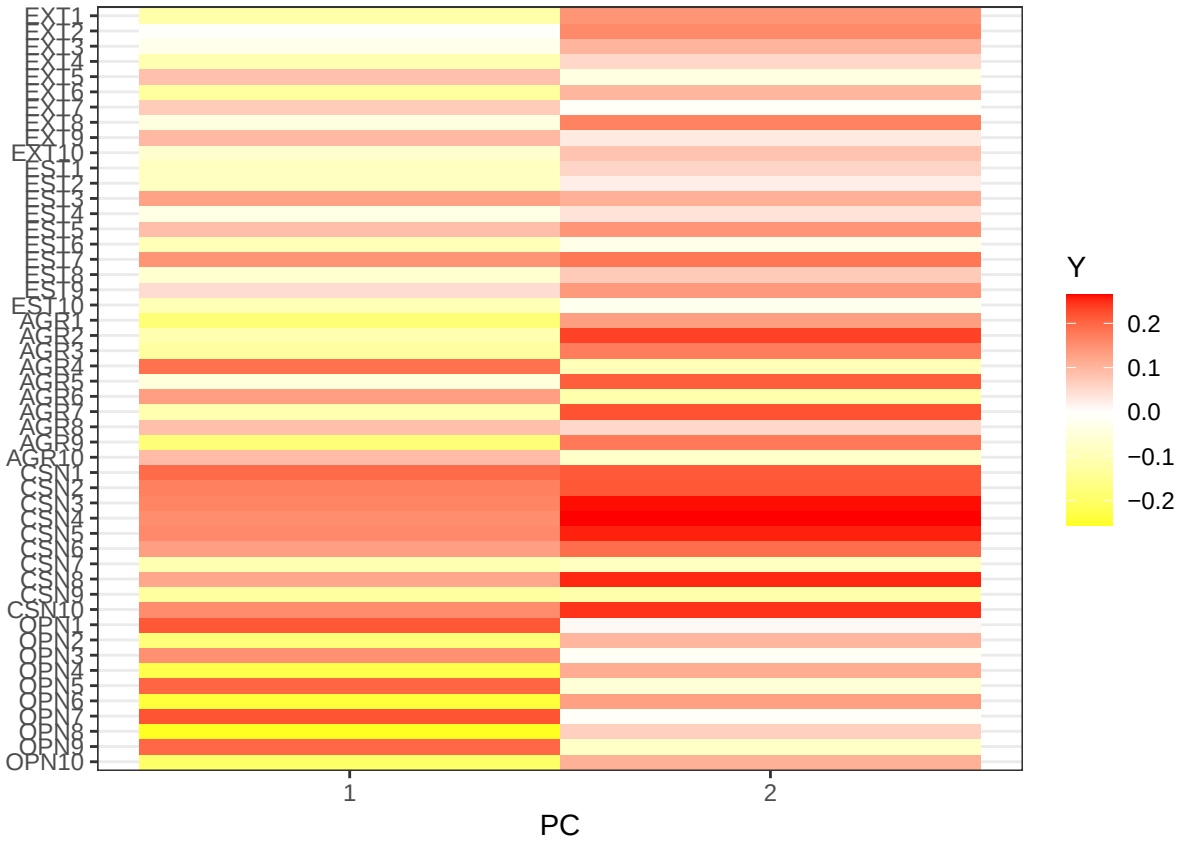
```
eigs_B[2] / sum(eigs_B)
```

```
## [1] 0.1017258
```

```
eigs_B[3] / sum(eigs_B)
```

```
## [1] 0.07914442
```

```
p=dim(dataPCA$rotation)[1]
label <- colnames(df)
data_plot <- data.frame(Row =rep(1:p, times= 2), Col = rep(x=c('1', '2'), each=p), Y= matrix(c(dataPCA$
heatmap <- ggplot(data_plot, aes(Col, Row)) + geom_tile(aes(fill=Y))
heatmap <- heatmap + scale_fill_gradient2(low="yellow", mid="white", high="red")
heatmap <- heatmap + scale_y_discrete('',limits=label[p:1]) + theme_bw() + scale_x_discrete('PC')
heatmap
```



(b) (40 points) Perform a factor analysis model with 5 factors with no rotation. How is the total variance explained from the model? Now perform the factor analysis model with 5 factors and with the varimax rotation (remember to not scale the data). Will you keep the model with 5 common factors or will you add another one? Explain why. Plot in an heatmap the matrix of factor loadings matrix (similar to Figure 1) Again choose your own colour by considering three different tonality. Interpretation: Now interpret the factors. Explain what each factor represents and give a name to each factor based on its high loadings.

Answer: I will keep with 5 common factors since the 6th common factor explains 0.03 portion of variance, which is smaller than 0.05. In the following results, we have 5 factors: Factor1, Factor2, Factor3, Factor4, Factor5. Each of them represent that they are the potential factor behind the variables that they correspond to in the loadings. If the color is deeper, this factor will explain more of the variance of that variable in the original dataset. To give each factor a name, Factor1 will be CSN, Factor 2 will be OPN, Factor 3 will be AGB, Factor4 will be EXT and Factor5 will be EST.

```
mod1=factanal(df, factors=5, rotation="none")
mod2=factanal(df, factors=5, rotation="varimax")
lort=loadings(mod2)
lort
```

```
##
## Loadings:
##       Factor1 Factor2 Factor3 Factor4 Factor5
## EXT1           0.690
## EXT2          -0.667  -0.121
## EXT3  -0.162   0.632   0.242           0.168
## EXT4   0.179  -0.700
## EXT5           0.696   0.201   0.124   0.136
```

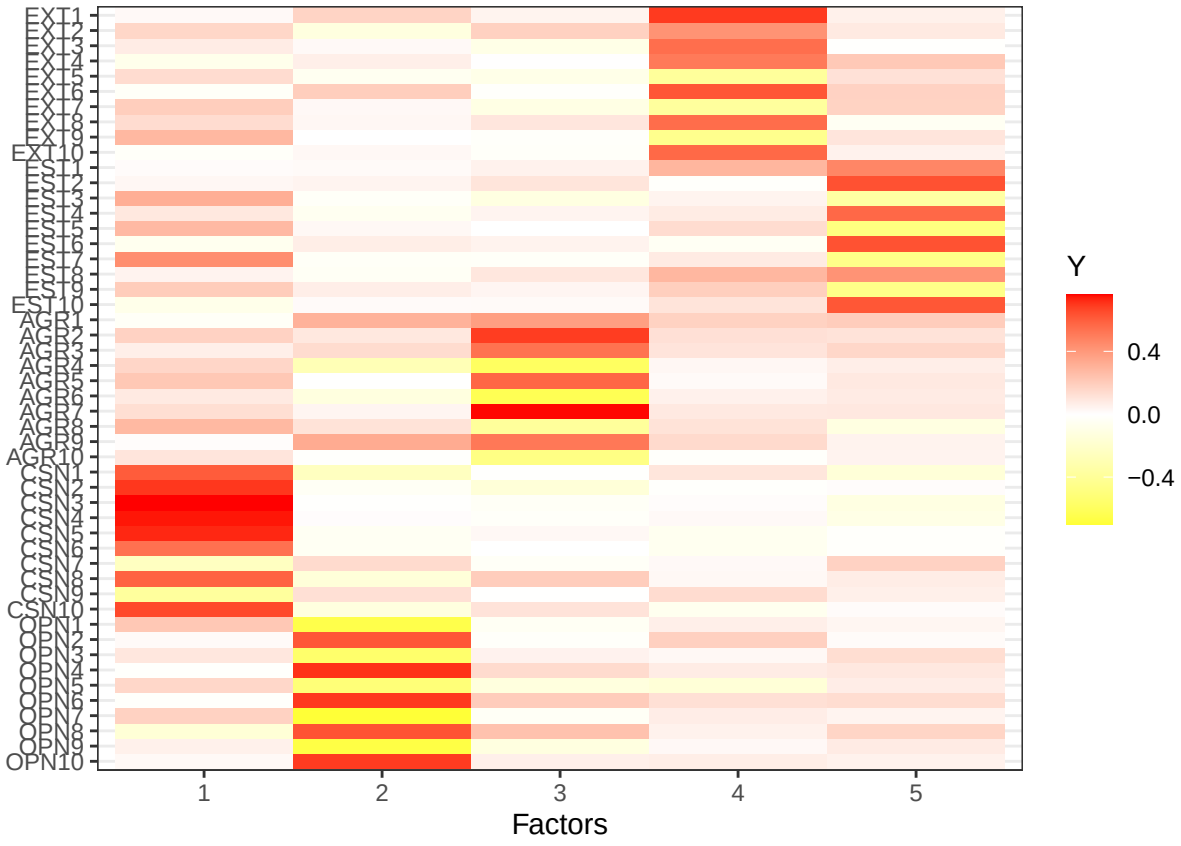
```
## EXT6   0.159  -0.516  -0.131  -0.157
## EXT7           0.711   0.144
## EXT8          -0.560                  0.131
## EXT9           0.620           0.185
## EXT10  0.219  -0.652
## EST1   0.657  -0.129   0.113
## EST2  -0.384   0.124           0.139
## EST3   0.586  -0.151   0.199
## EST4  -0.245   0.143                  0.177
## EST5   0.540
## EST6   0.726
## EST7   0.747
## EST8   0.761                         -0.115
## EST9   0.699          -0.154
## EST10  0.606  -0.247           0.101  -0.153
## AGR1   0.104          -0.473
## AGR2           0.332   0.516   0.146
## AGR3   0.278   0.113  -0.391   0.113  -0.119
## AGR4   0.130           0.758
## AGR5          -0.127  -0.628
## AGR6   0.220           0.575
## AGR7   0.161  -0.286  -0.593
## AGR8           0.141   0.538   0.107   0.161
## AGR9   0.178           0.685   0.122   0.111
## AGR10          0.304   0.378   0.181   0.198
## CSN1                           0.107   0.622
## CSN2   0.201                   0.191  -0.457
## CSN3                           0.284   0.421
## CSN4   0.439                          -0.457
## CSN5                                   0.635
## CSN6   0.276                   0.138  -0.487
## CSN7                                   0.568
## CSN8   0.320          -0.118          -0.359
## CSN9                   0.106           0.636
## CSN10                          0.290   0.468
## OPN1                           0.568
## OPN2   0.283                  -0.441   0.103
## OPN3   0.138           0.100   0.555
## OPN4   0.197          -0.105  -0.382   0.176
## OPN5           0.196           0.623   0.179
## OPN6   0.139                  -0.391   0.119
## OPN7                           0.511   0.214
## OPN8                           0.550
## OPN9   0.161  -0.127   0.184   0.423
## OPN10          0.172           0.693
##
##                Factor1 Factor2 Factor3 Factor4 Factor5
## SS loadings      4.796   4.788   3.558   3.229   3.148
## Proportion Var   0.096   0.096   0.071   0.065   0.063
## Cumulative Var   0.096   0.192   0.263   0.327   0.390
```

```
mod2_try6=factanal(df, factors=6, rotation="varimax")
lort_try6=loadings(mod2_try6)
lort_try6
```

```
##
## Loadings:
##       Factor1 Factor2 Factor3 Factor4 Factor5 Factor6
## EXT1  -0.666           0.101                   0.210
## EXT2   0.699                                   0.172
## EXT3  -0.602  -0.212   0.288           0.111   0.239
## EXT4   0.733   0.136                           0.163
## EXT5  -0.675           0.236   0.137           0.149
## EXT6   0.560                  -0.162           0.312
## EXT7  -0.685           0.185                   0.203
## EXT8   0.583                                   0.138
## EXT9  -0.598                   0.197           0.177
## EXT10  0.672   0.185                           0.130
## EST1   0.117   0.687   0.112
## EST2          -0.490           0.133           0.375
## EST3   0.148   0.600   0.207
## EST4  -0.105  -0.310                   0.106   0.316
## EST5           0.523                           0.125
## EST6           0.730
## EST7           0.736                  -0.130
## EST8           0.755                  -0.139
## EST9           0.702  -0.133
## EST10  0.242   0.613           0.105  -0.163
## AGR1                  -0.434                   0.296
## AGR2  -0.311           0.539   0.137
## AGR3           0.247  -0.355   0.124  -0.172   0.202
## AGR4                   0.776
## AGR5   0.151          -0.594                   0.332
## AGR6           0.180   0.600                   0.103
## AGR7   0.309   0.126  -0.561                   0.307
## AGR8  -0.113           0.570   0.101   0.107   0.117
## AGR9           0.142   0.706   0.112
## AGR10 -0.268           0.424   0.183   0.131   0.209
## CSN1                           0.143   0.601   0.101
## CSN2           0.126           0.168  -0.556   0.180
## CSN3                   0.122   0.305   0.385
## CSN4           0.374                  -0.549   0.194
## CSN5                                   0.619   0.140
## CSN6           0.196           0.114  -0.597   0.219
## CSN7           0.106           0.112   0.544   0.107
## CSN8           0.253                  -0.439   0.233
## CSN9                   0.131           0.606   0.135
## CSN10                          0.315   0.425   0.123
## OPN1                           0.569
## OPN2           0.229          -0.455           0.405
## OPN3           0.120   0.116   0.550
## OPN4           0.140          -0.390           0.444
## OPN5  -0.175                   0.628   0.126   0.104
## OPN6                          -0.385           0.311
## OPN7          -0.105           0.522   0.165
## OPN8                           0.553
## OPN9   0.138   0.147   0.198   0.423
## OPN10 -0.157                   0.690
##
```

```
##                Factor1 Factor2 Factor3 Factor4 Factor5 Factor6
## SS loadings      4.755   4.641   3.661   3.288   3.148   1.810
## Proportion Var   0.095   0.093   0.073   0.066   0.063   0.036
## Cumulative Var   0.095   0.188   0.261   0.327   0.390   0.426
```

```
data_plot <- data.frame(Row =rep(1:p, times= 5), Col = rep(x=c('1', '2', '3', '4', '5'), each=p), Y= ma
heatmap <- ggplot(data_plot, aes(Col, y=Row)) + geom_tile(aes(fill=Y))
heatmap <- heatmap + scale_fill_gradient2(low="yellow", mid="white", high="red")
heatmap <- heatmap + scale_y_discrete('',limits=label[p:1]) + theme_bw() + scale_x_discrete('Factors')
heatmap
```



(c) (20 points) Perform a bootstrap of 50 samples. For each of the bootstrapped sample save the proportion of variance explained by each factor (consider just the first five factors). Plot the proportion of variance explained by each of the five factors with a boxplot in the ggplot and then perform the histogram for each proportion. What can you say about these five distributions obtained? If we bootstrap the loadings we will obtain something no sense in a statistical framework. Explain why.

Answer: The boxplot shows that there are very little outliers except for my Factor3, and the size of box (space between Q1 and Q3) is small which means that the sample factors explain pretty much the same proportion of variance in the sample data. Therefore, the factors given by the factor analysis are stable and reliable. The 5 distributions give a nearly normal distribution with their proportion of variance as mean and very small variance, which matches what we see on the boxplot. The "factanal" always use "mle" to get the potential factors, "maximum likehood estimation" uses bootstrap to estimate the statistics. Therefore, if we do bootstrapping using "factanal" in this problem, it makes no sense as "factanal" loadings is a result of bootstrapping itself.

```
process <- function(data, idx){
  data <- data[idx,]
```

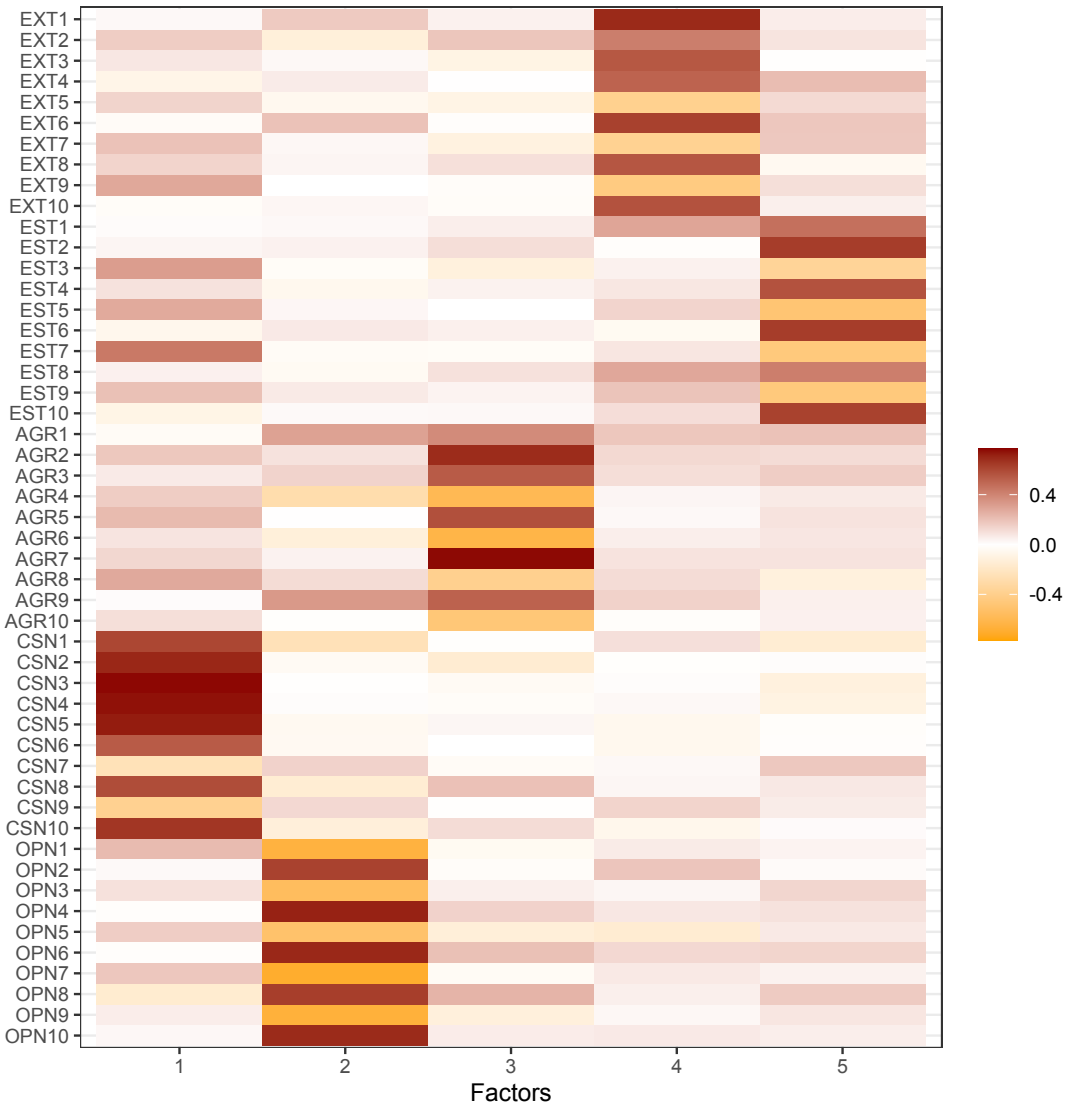
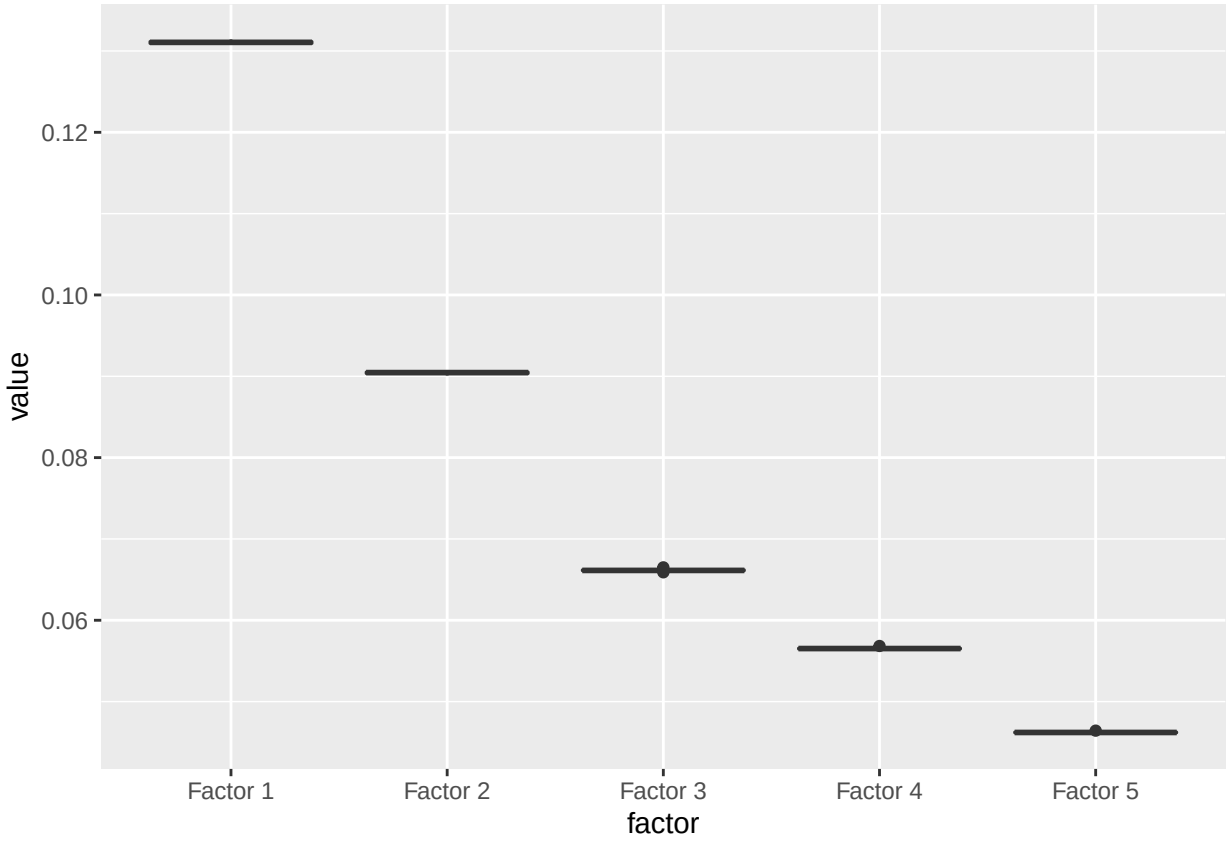


Figure 1: Estimate

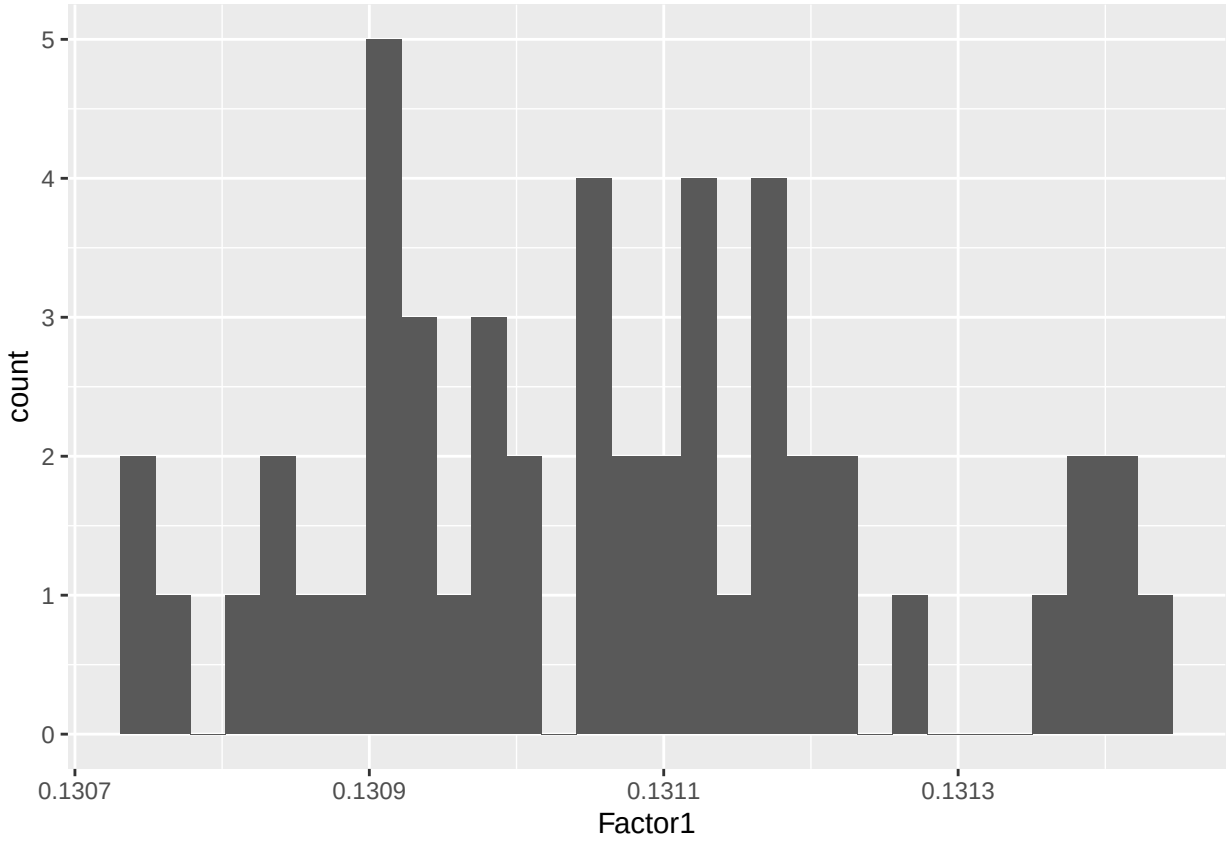
```
  mod <- factanal(data, factors=5, rotation="none")
  lort <- loadings(mod)
  return (colSums(lort^2)/nrow(lort))
}
sample <- boot::boot(data=df, statistic=process, R=50)
sample
```

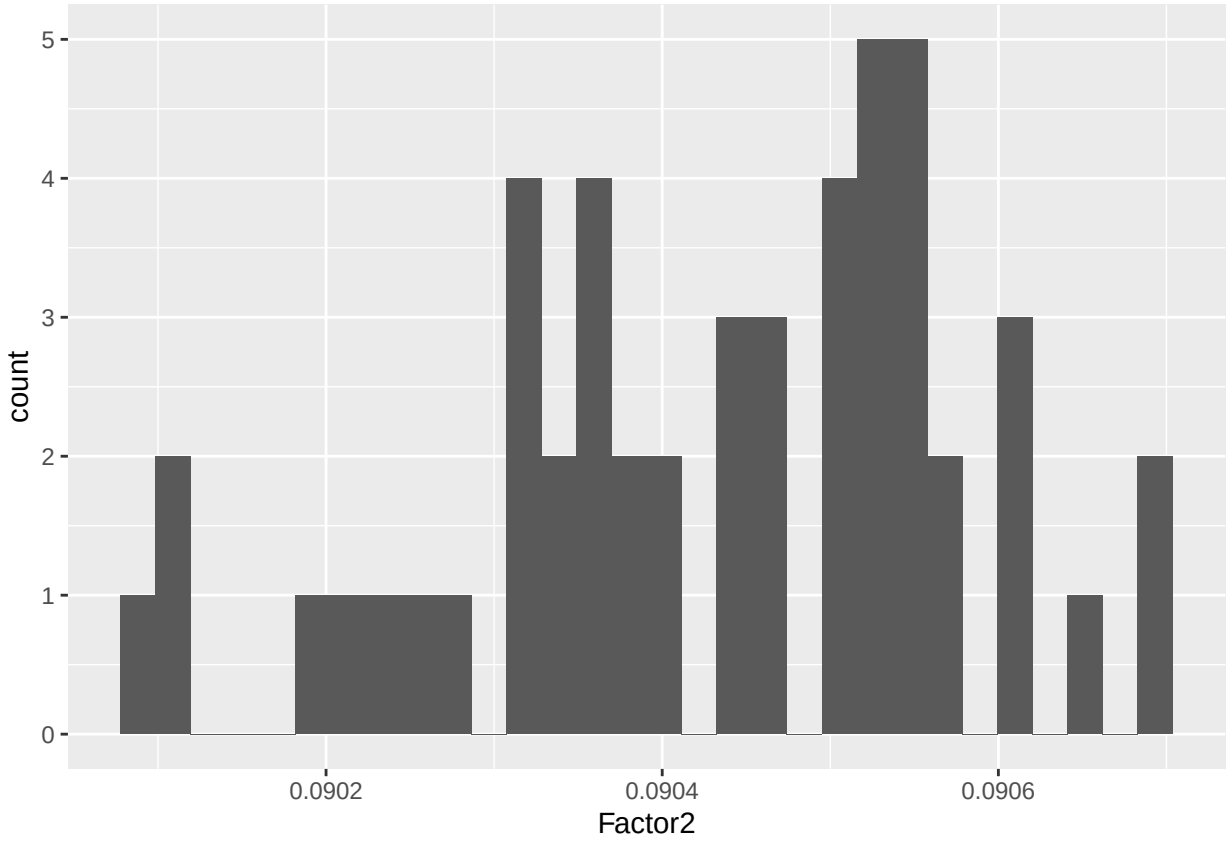
```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot::boot(data = df, statistic = process, R = 50)
##
##
## Bootstrap Statistics :
##       original        bias     std. error
## t1* 0.13111757 -5.406957e-05 1.813227e-04
## t2* 0.09038226  4.815605e-05 1.472065e-04
## t3* 0.06614332  9.527622e-07 1.150615e-04
## t4* 0.05654284 -2.203972e-05 9.499970e-05
## t5* 0.04618135  1.805438e-05 8.784117e-05
```

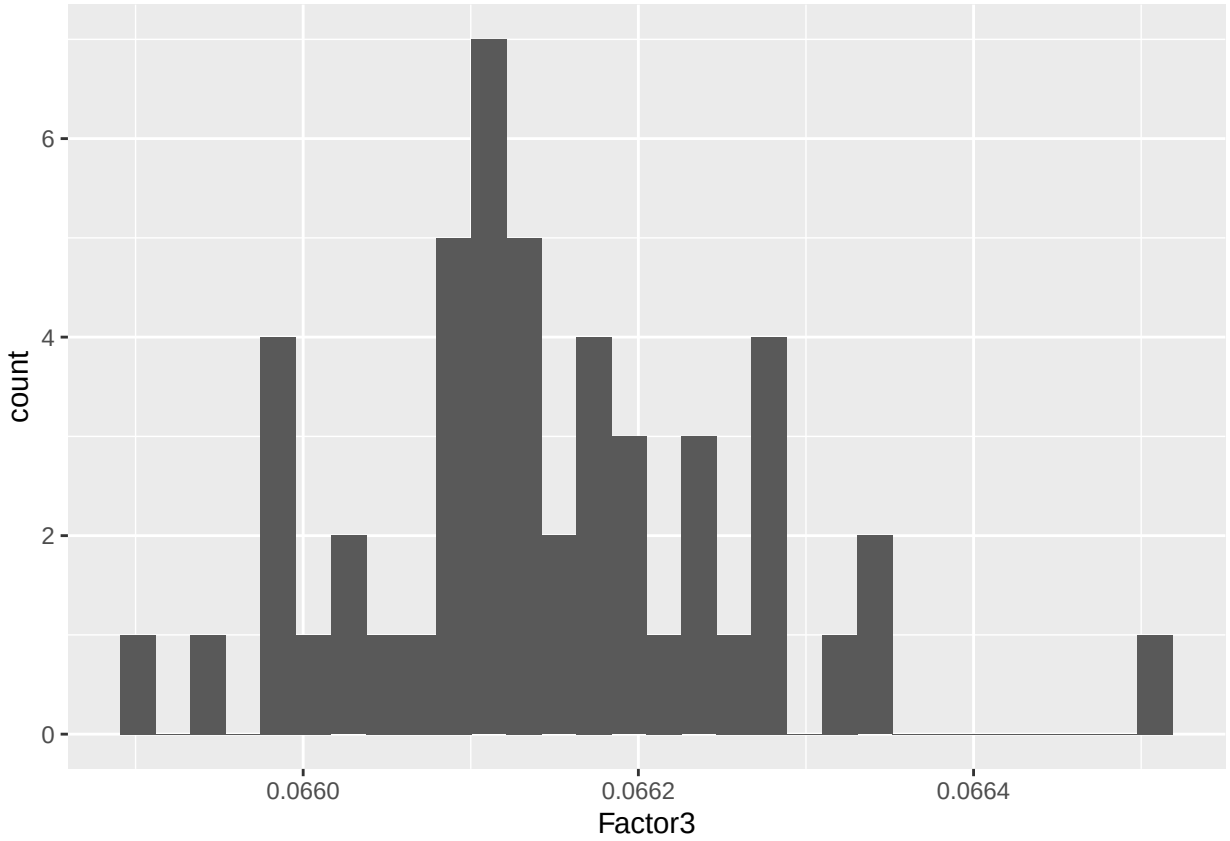
```
stat <- sample$t
df_3 <- data.frame(Factor1=stat[,1], Factor2=stat[,2], Factor3=stat[,3], Factor4=stat[,4], Factor5=stat
df_3_thin <- data.frame()
for (i in 1:5){
  df_3_thin <- rbind(data.frame(value=stat[,i], factor=paste("Factor", i)), df_3_thin)
}
ggplot(df_3_thin, aes(x=factor, y=value)) + geom_boxplot()
```

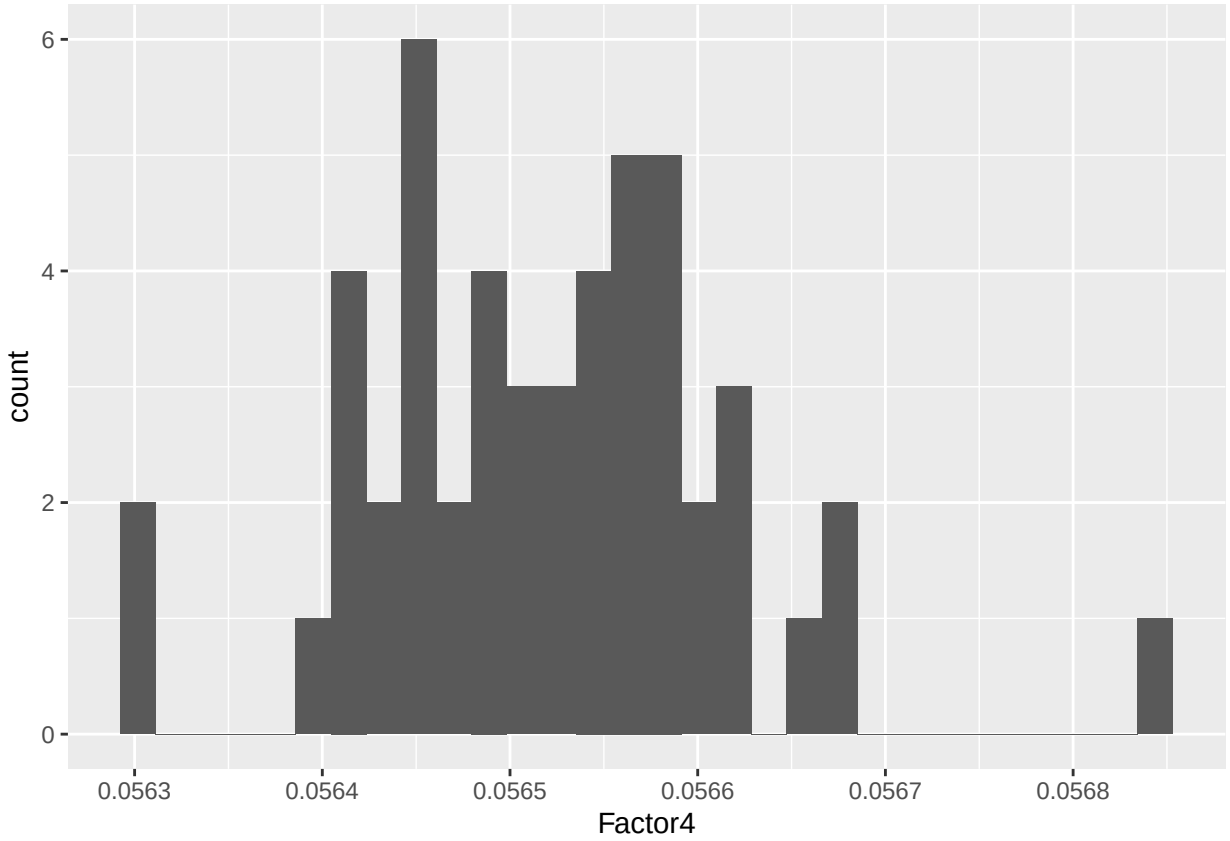


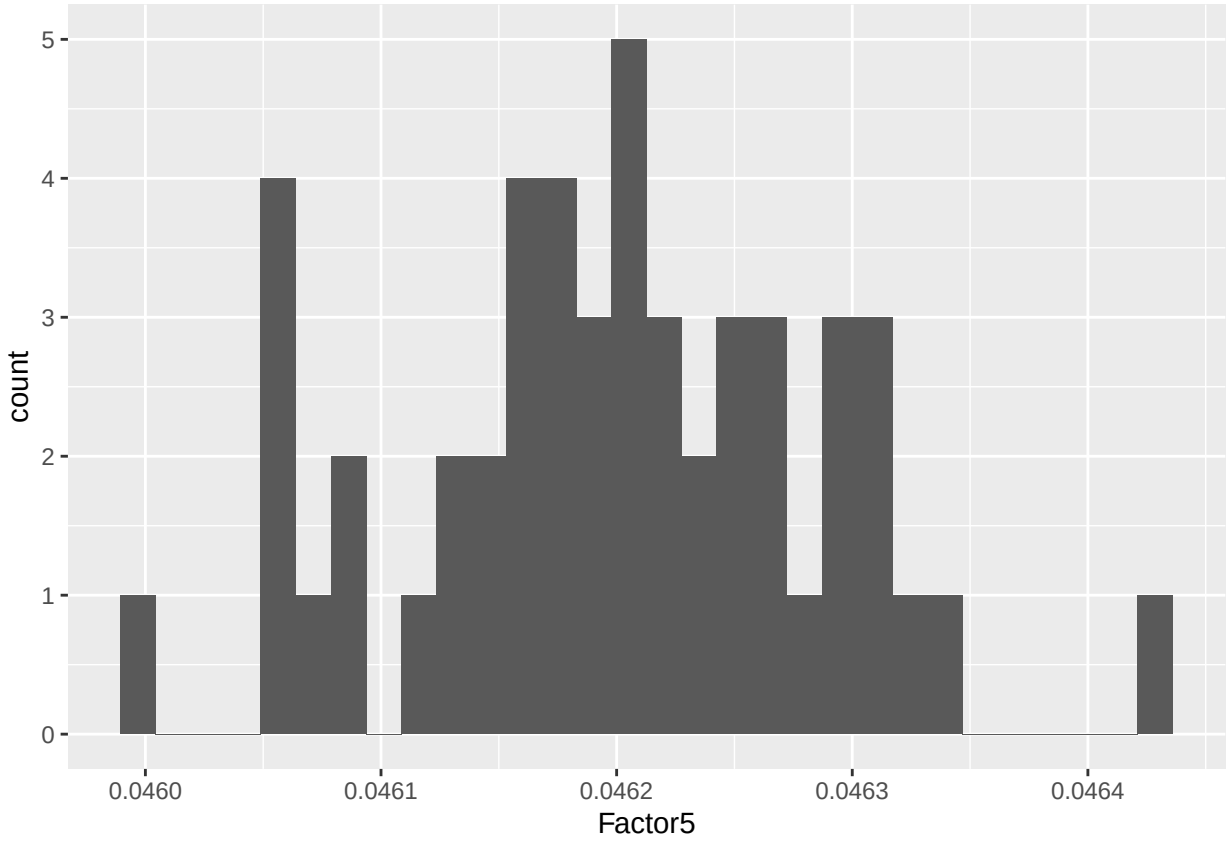

```
for (i in 1:5){
  p <- ggplot(df_3, aes_string(x=names(df_3)[i])) + geom_histogram(bins=30)
  plot(p)
}
```


count
5
4
3
2
1
0
0.1307
0.1309
0.1311
0.1313
Factor1


5
4
3
2
1
0
count
0.0902
0.0904
0.0906
Factor2


count
6
4
2
0
0.0660
0.0662
0.0664
Factor3


6
4
2
0
count
0.0563
0.0564
0.0565
0.0566
0.0567
0.0568
Factor4


count
5
4
3
2
1
0
0.0460
0.0461
0.0462
0.0463
0.0464
Factor5