456F2

```r
p = read.csv("PSY.csv",header =  T)
p = p[,5:28]
head(p)
```

```
##   V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20
## 1 20 31 12  3 40  7 23 22  9  78  74 115 229 170  86  96   6   9  16   3
## 2 32 21 12 17 34  5 12 22  9  87  84 125 285 184  85 100  12  12  10  -3
## 3 27 21 12 15 20  3  7 12  3  75  49  78 159 170  85  95   1   5   6  -3
## 4 32 31 16 24 42  8 18 21 17  69  65 106 175 181  80  91   5   3  10  -2
## 5 29 19 12  7 37  8 16 25 18  85  63 126 213 187  99 104  15  14  14  29
## 6 32 20 11 18 31  3 12 25  6 100  92 133 270 164  84 104   6   6  14   9
##   V21 V22 V23 V24
## 1  14  34   5  24
## 2  13  21   1  12
## 3   9  18   7  20
## 4  10  22   6  19
## 5  15  19   4  20
## 6   2  16  10  22
```

```r
##########
##Problem1
##########
library(stats)
#a.
#plugging in factor =6, I get that 6 factors can adeautely explain
#base on results from the scree plot and factanal analysis output
p = as.data.frame(scale(p))
res1p = factanal(p,factors = 6, rotation = "varimax", na.action = na.omit)
res1p
```

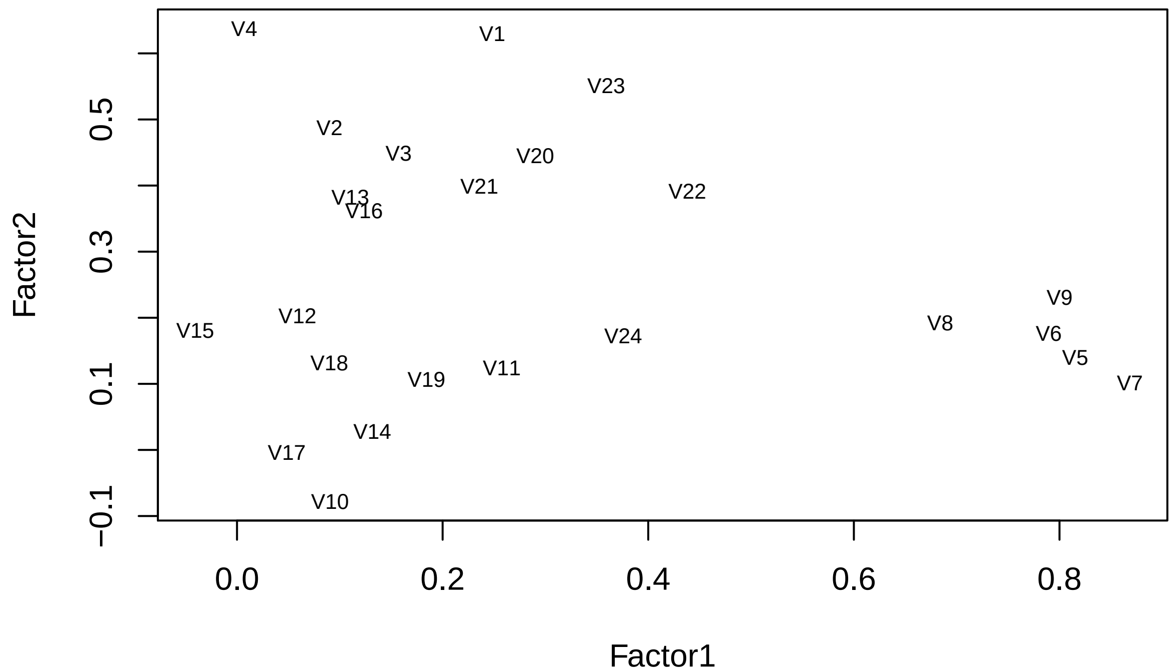
```
##
## Call:
## factanal(x = p, factors = 6, na.action = na.omit, rotation = "varimax")
##
## Uniquenesses:
##    V1    V2    V3    V4    V5    V6    V7    V8    V9   V10   V11   V12
## 0.488 0.739 0.763 0.547 0.299 0.311 0.214 0.442 0.278 0.037 0.469 0.592
##   V13   V14   V15   V16   V17   V18   V19   V20   V21   V22   V23   V24
## 0.447 0.575 0.555 0.589 0.612 0.726 0.609 0.597 0.530 0.536 0.426 0.468
##
## Loadings:
##     Factor1 Factor2 Factor3 Factor4 Factor5 Factor6
## V1   0.248   0.630           0.159           0.140
## V2           0.488                   0.107
## V3   0.157   0.449
## V4           0.638   0.101   0.182
## V5   0.815   0.140   0.119
## V6   0.790   0.176           0.143
## V7   0.869   0.102                           0.106
## V8   0.684   0.192           0.105   0.167
## V9   0.800   0.231                   0.125
```

```
## V10                   0.954   0.164           -0.105
## V11  0.258   0.124   0.453   0.279   0.128   0.386
## V12          0.204   0.539           0.113   0.235
## V13  0.110   0.383   0.414                   0.456
## V14  0.132                   0.617   0.162
## V15          0.181           0.627           -0.120
## V16  0.124   0.362           0.457   0.220
## V17                  0.259   0.552           0.102
## V18          0.132   0.153   0.443   0.124   0.117
## V19  0.184   0.106           0.275   0.502   0.130
## V20  0.290   0.445           0.218   0.243  -0.121
## V21  0.236   0.399   0.359   0.142   0.320
## V22  0.438   0.392           0.101   0.323
## V23  0.359   0.551   0.181   0.154   0.274  -0.101
## V24  0.376   0.173   0.345   0.192   0.449
##
##                Factor1 Factor2 Factor3 Factor4 Factor5 Factor6
## SS loadings      4.016   2.643   2.020   1.917   0.996   0.559
## Proportion Var   0.167   0.110   0.084   0.080   0.042   0.023
## Cumulative Var   0.167   0.277   0.362   0.441   0.483   0.506
##
## Test of the hypothesis that 6 factors are sufficient.
## The chi square statistic is 155.66 on 147 degrees of freedom.
## The p-value is 0.297
```

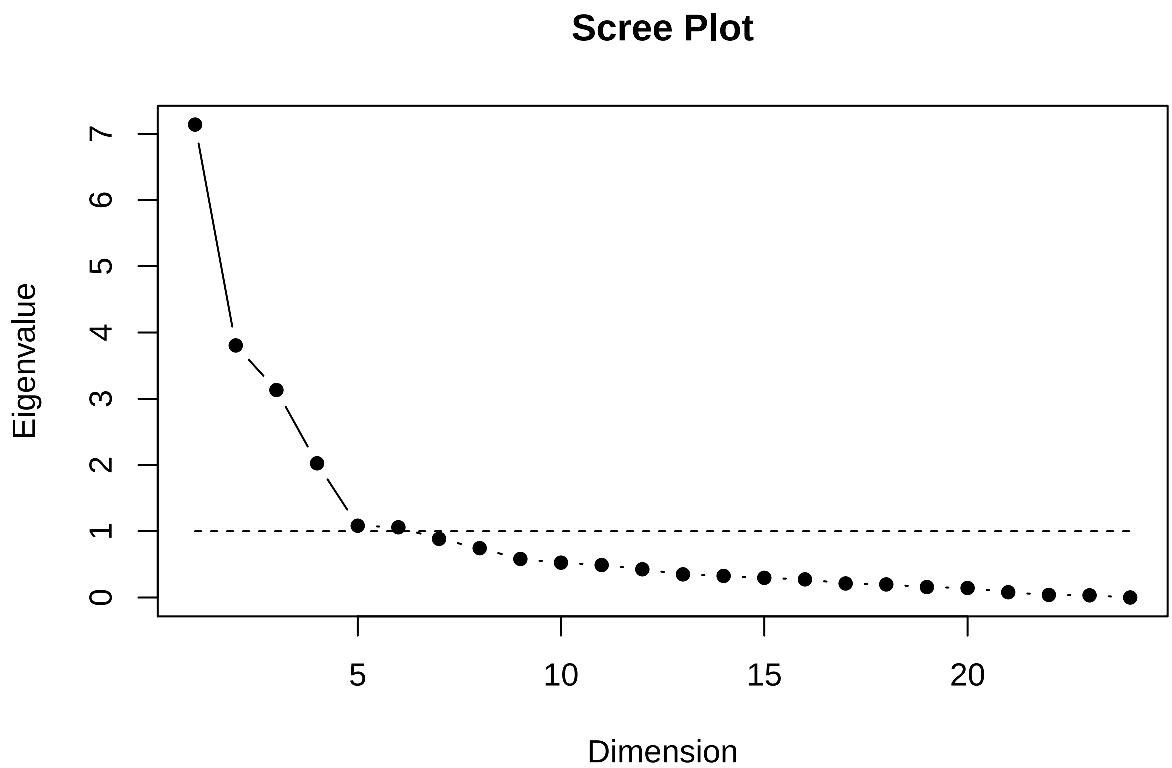
```r
#I look at the sums of squared (SS) loadings; these are the eigenvalues,
#or the variance in all variables which is accounted for by that factor
#(i.e., the eigenvalue/# of variables = proportion variance). If a factor has a
#"high" SS loading/eigenvalue, then it is helping to explain the variances in the
#variables. In the factanal() output, the factors are ordered by their eigenvalues,
#with higher eigenvalues first. As factor is important if its
#eigenvalue is greater than 1.

#Base on those information, factors 1-6 appear to be important.
#So six factor model can indeed explain the relationship
#among the variables

#Therefore, I might conclude that 6 factors are enough for this model
#Scree plot code
load1 = res1p$loadings
library(psy)
plot(load1,type="n")
text(load1,labels = names(p),cex=0.7)
```




```
#Determine # of factors to extract : 6 in this case
scree.plot(res1p$correlation)
```




```r
#choose 6 factors since their engivalues are larger than 1
```

```r
load1 = res1p$loadings[,1:2]
load1
```

```
##          Factor1      Factor2
## V1   0.248346042  0.630429035
## V2   0.089987640  0.488103474
## V3   0.157270612  0.449219675
## V4   0.007057571  0.637816210
## V5   0.815272541  0.140018254
## V6   0.789648169  0.176312699
## V7   0.868559515  0.102210686
## V8   0.684044352  0.192287914
## V9   0.800179986  0.230863259
## V10  0.090486883 -0.078182626
## V11  0.257605778  0.124361433
## V12  0.058818886  0.203651243
## V13  0.110282356  0.382597859
## V14  0.131677255  0.028161035
## V15 -0.040601309  0.180993829
## V16  0.123546691  0.362168707
## V17  0.048479326 -0.003177894
## V18  0.089805774  0.131595980
## V19  0.184343199  0.106448930
## V20  0.290130380  0.445141522
## V21  0.235735989  0.399384337
```

```
## V22  0.438044519  0.391772164
## V23  0.359196416  0.550931685
## V24  0.375657269  0.173021006
```

```
#Interpretation:
#
```

```
###############
#Note: I will give communalities and variance(uniqueness)
#for all 24 variables
###############

#engivalues for the 1st factor ==
loadings_fac1 = res1p$loadings[,1]
eigenv_fac1 = sum(loadings_fac1^2)
eigenv_fac1
```

```
## [1] 4.015961
```

```
#res1p$uniquenesses
loadings_fac1 = res1p$loading[1,]
communality_fac1 = sum(loadings_fac1^2)
#communalities = 1-uniqueness
#Answer to question(c)
communality_fac1
```

```
## [1] 0.5120692
```

```
uniqueness_fac1 = 1-communality_fac1
uniqueness_fac1
```

```
## [1] 0.4879308
```

```
#This is the answer for the (d): specific variances
#variance are simply the uniqueness;
```

```
#Give the four plots that helps explain
pcaCharts <- function(x) {
    x.var <- x$sdev ^ 2
    x.pvar <- x.var/sum(x.var)
    print("proportions of variance:")
    print(x.pvar)

    par(mfrow=c(2,2))
    plot(x.pvar,xlab="Principal component", ylab="Proportion of variance explained", ylim=c(0,1), type=
    plot(cumsum(x.pvar),xlab="Principal component", ylab="Cumulative Proportion of variance explained",
    screeplot(x)
    screeplot(x,type="l")
    par(mfrow=c(1,1))
}
#pcaCharts(res1p)
```

```
#####################
########Part2
#####################
e = read.csv("EFA.csv",header =  T)
head(e)
```

```
##   Price Safety Exterior_Looks Space_comfort Technology After_Sales_Service
## 1     4      4              5             4          3                   4
## 2     3      5              3             3          4                   4
## 3     4      4              3             4          5                   5
## 4     4      4              4             3          3                   4
## 5     5      5              4             4          5                   4
## 6     4      4              5             3          4                   5
##   Resale_Value Fuel_Type Fuel_Efficiency Color Maintenance Test_drive
## 1            5         4               4     2           4          2
## 2            3         4               3     4           3          2
## 3            5         4               5     4           5          4
## 4            5         5               4     4           4          2
## 5            5         3               4     5           5          5
## 6            3         4               3     2           3          2
##   Product_reviews Testimonials
## 1               4            3
## 2               2            2
## 3               4            3
## 4               5            3
## 5               5            2
## 6               2            3
```

```
##########
##Problem1
##########
library(stats)
#a.
#plugging in factor =6, I get that 6 factors can adeautely explain
#base on results from the scree plot and factanal analysis output
res2e = factanal(e,factors = 1, rotation = "varimax", na.action = na.omit)
res2e
```

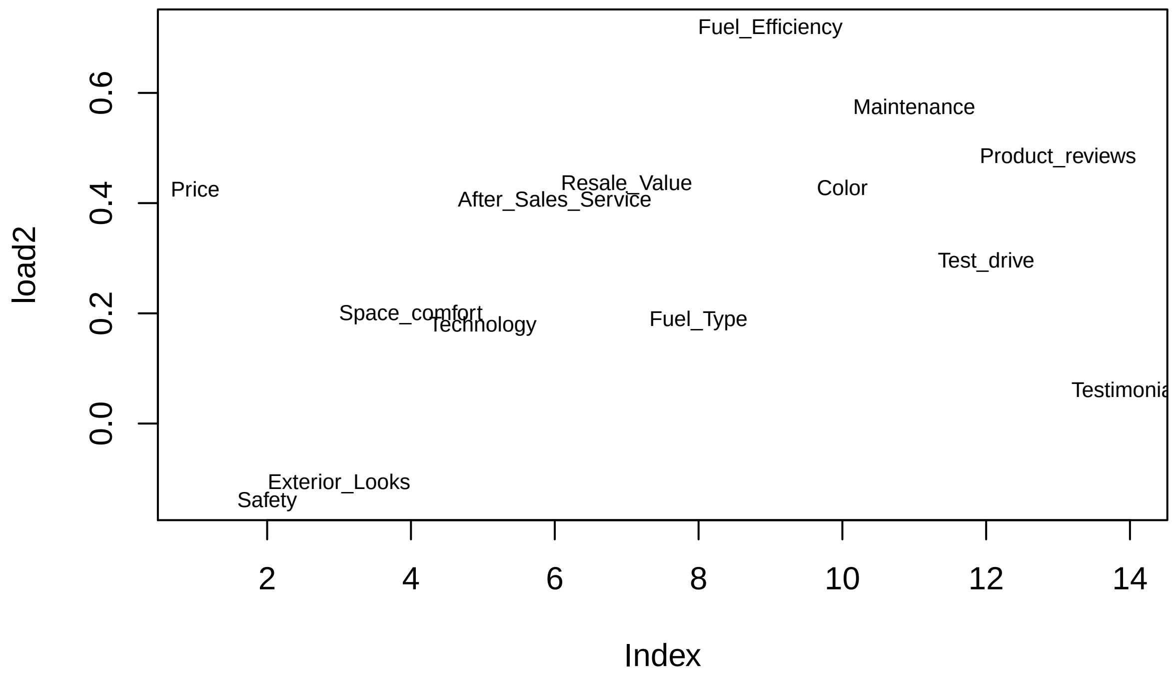
```
##
## Call:
## factanal(x = e, factors = 1, na.action = na.omit, rotation = "varimax")
##
## Uniquenesses:
##               Price              Safety      Exterior_Looks
##               0.819               0.980               0.988
##       Space_comfort          Technology After_Sales_Service
##               0.961               0.969               0.836
##        Resale_Value           Fuel_Type     Fuel_Efficiency
##               0.811               0.965               0.486
##               Color         Maintenance          Test_drive
##               0.817               0.669               0.913
##     Product_reviews        Testimonials
##               0.766               0.996
##
## Loadings:
##                     Factor1
## Price                0.425
## Safety              -0.141
## Exterior_Looks      -0.108
## Space_comfort        0.198
## Technology           0.176
```

```
## After_Sales_Service  0.405
## Resale_Value         0.435
## Fuel_Type            0.187
## Fuel_Efficiency      0.717
## Color                0.428
## Maintenance          0.575
## Test_drive           0.294
## Product_reviews      0.484
## Testimonials
##
##                 Factor1
## SS loadings       2.025
## Proportion Var    0.145
##
## Test of the hypothesis that 1 factor is sufficient.
## The chi square statistic is 159.92 on 77 degrees of freedom.
## The p-value is 9.35e-08
```

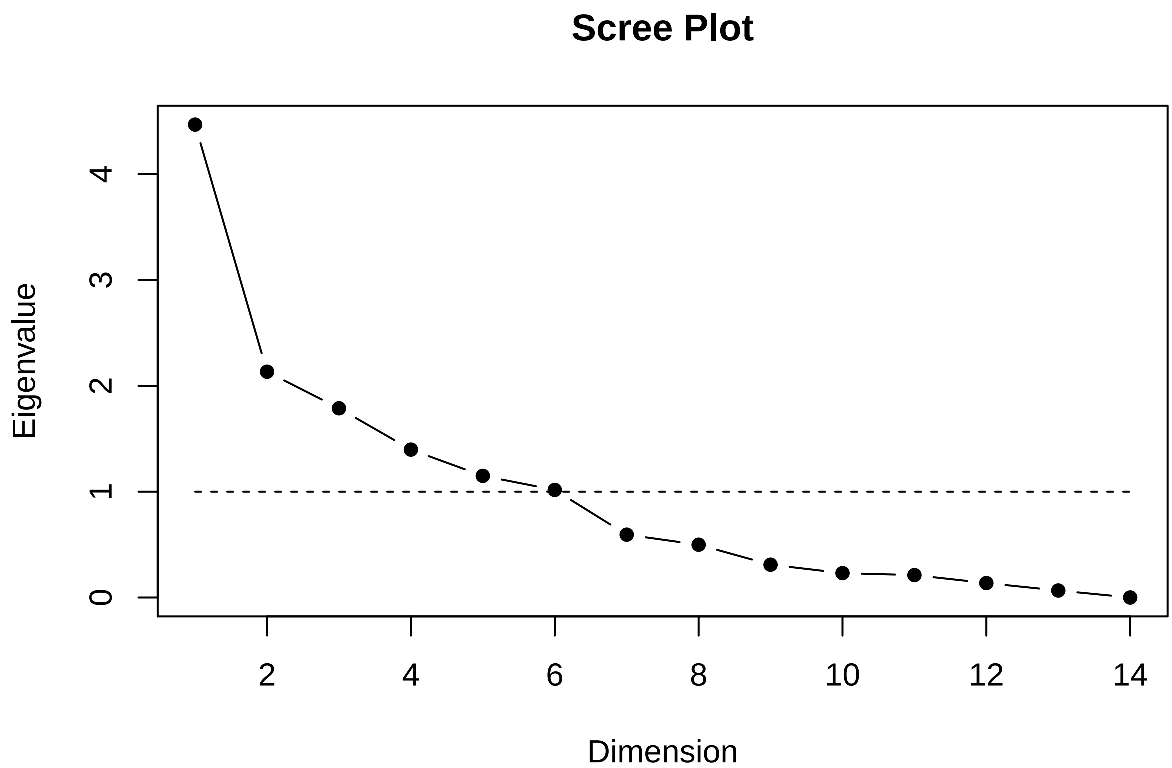
```r
#I look at the sums of squared (SS) loadings; these are the eigenvalues,
#or the variance in all variables which is accounted for by that factor
#(i.e., the eigenvalue/# of variables = proportion variance). If a factor has a
#"high" SS loading/eigenvalue, then it is helping to explain the variances in the
#variables. In the factanal() output, the factors are ordered by their eigenvalues,
#with higher eigenvalues first. As factor is important if its
#eigenvalue is greater than 1.

#Base on those information, factors 1-6 appear to be important.
#So six factor model can indeed explain the relationship
#among the variables

#Therefore, I might conclude that 6 factors are enough for this model
#Scree plot code
load2 = res2e$loadings
library(psy)
plot(load2,type="n")
text(load2,labels = names(e),cex=0.7)
```




```
#Determine # of factors to extract : 6 in this case
scree.plot(res2e$correlation)
```




```
#choose 6 factors since their engivalues are larger than 1
```