Factor analysis

Ajayraj pasi

2024-03-01

Library

```
library(psych)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(factoextra)

## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##     %+%, alpha

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

library(FactoMineR)
library(GPArotation)

##
## Attaching package: 'GPArotation'

## The following objects are masked from 'package:psych':
##
##     equamax, varimin
```

Data Import

```
# data loading
data<-read.csv('C:/Users/Apse360/Desktop/factor/Stat career.csv')
head(data)
```

```
##   stat_cry afraid_spss sd_excite nmare_pearson du_stat lexp_comp comp_hate
## 1        2           1         4             2       2         2         3
## 2        1           1         4             3       2         2         2
## 3        2           3         2             2       4         1         2
## 4        3           1         1             4       3         3         4
## 5        2           1         3             2       2         3         3
## 6        2           1         3             2       4         4         4
##   good_math frs_better_stat com_for_games bad_math spss_no_help
damaging_comp
## 1         1               1             2        1            2
2
## 2         2               5             2        2            3
1
## 3         2               2             2        3            3
2
## 4         2               2             4        2            2
2
## 5         2               4             2        2            3
3
## 6         2               4             3        2            4
3
##   comp_alive comp_getme weep_ct slip_coma spss_crash eb_looks no_sleep_ev
## 1          2          2       3         1          2        3           2
## 2          3          4       3         2          2        3           4
## 3          4          2       3         2          3        1           4
## 4          3          3       3         2          4        2           4
## 5          2          2       2         2          3        3           4
## 6          3          5       2         3          5        1           5
##   nm_normdist frs_better_spss stat_nerd career
## 1           2               2         5      1
## 2           4               4         2      0
## 3           3               2         2      1
## 4           4               4         3      0
## 5           2               4         4      0
## 6           3               1         4      0

# Removing dependent variable
df<-subset(data,select = -career)
colnames(df)<-
c('q1','q2','q3','q4','q5','q6','q7','q8','q9','q10','q11','q12','q13','q14',
'q15','q16','q17','q18','q19','q20','q21','q22','q23')
```

Factor test

```
# Checking significance for factor analysis
KMO(df)

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = df)
## Overall MSA =  0.93
## MSA for each item =
```

```
##   q1   q2   q3   q4   q5   q6   q7   q8   q9  q10  q11  q12  q13  q14  q15 
q16 
## 0.93 0.87 0.95 0.96 0.96 0.89 0.94 0.87 0.83 0.95 0.91 0.95 0.95 0.97 0.94 
0.93 
##  q17  q18  q19  q20  q21  q22  q23 
## 0.93 0.95 0.94 0.89 0.93 0.88 0.77
```

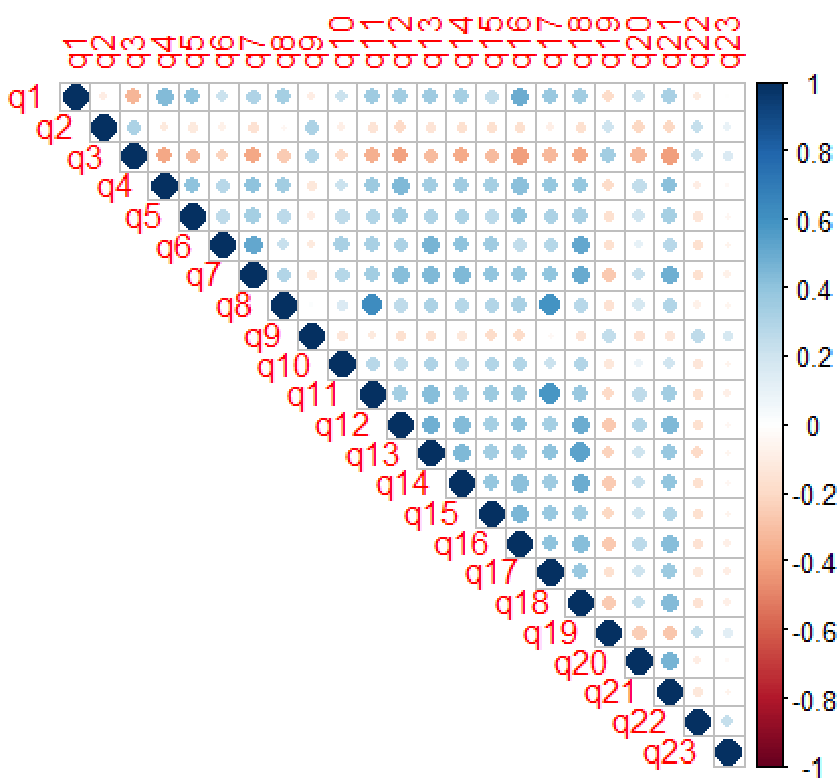
```r
bartlett.test(df)
```

```
## 
##  Bartlett test of homogeneity of variances
## 
## data:  df
## Bartlett's K-squared = 1277.9, df = 22, p-value < 2.2e-16
```

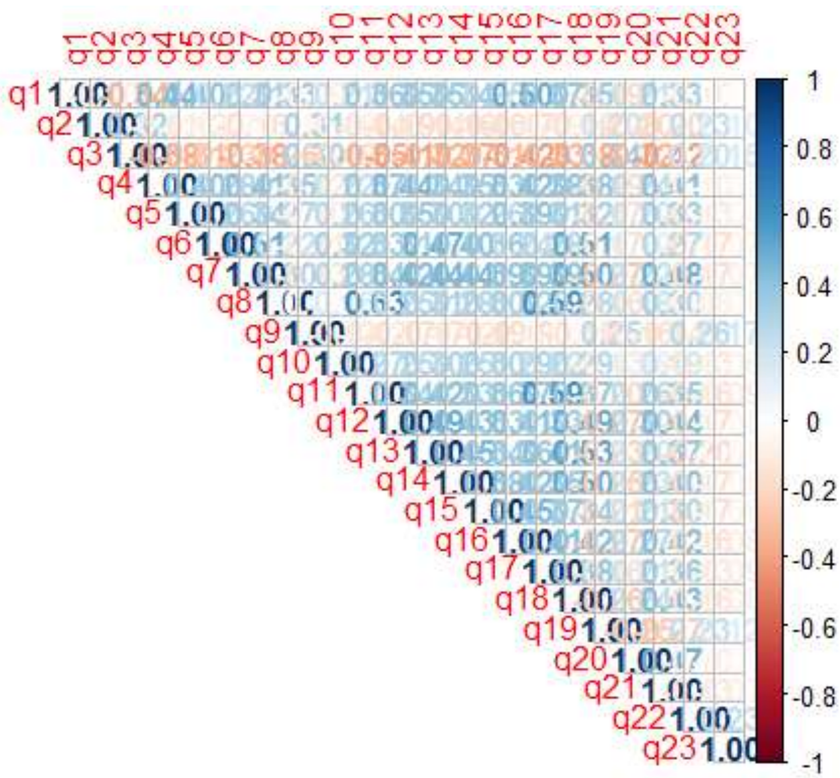
```r
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
corrplot(cor(df),type = 'upper')
```

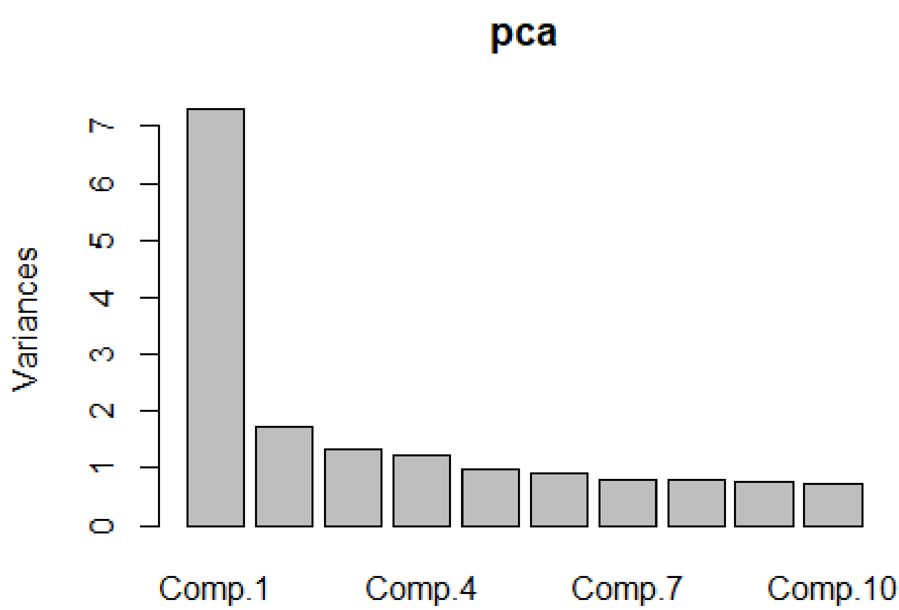



```r
corrplot(round(cor(df),2),method='number',type = 'upper')
```

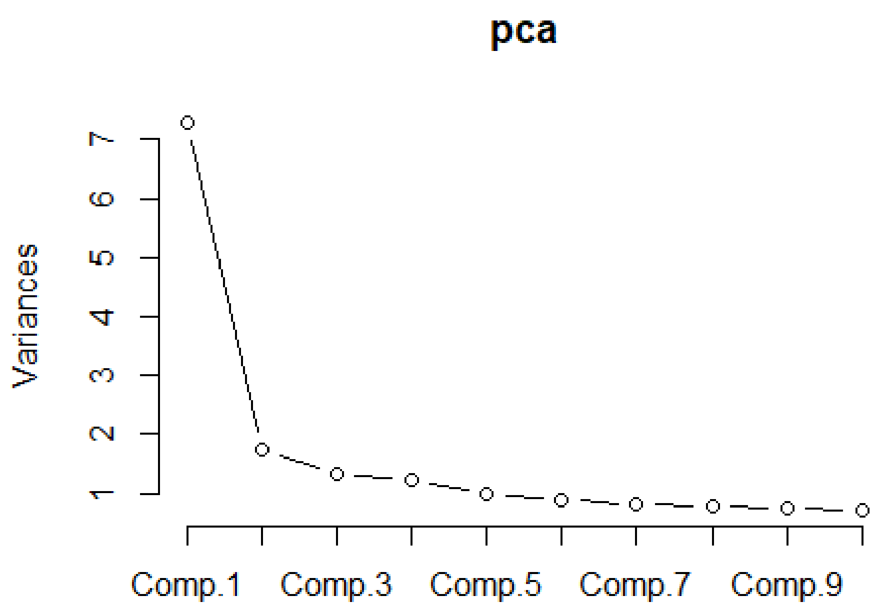



```
# Principal component analysis
pca<-princomp(df,cor = T)
summary(pca)

## Importance of components:
##                           Comp.1     Comp.2     Comp.3     Comp.4
Comp.5
## Standard deviation     2.7000087 1.31864656 1.14749794 1.10778976
0.99392047
## Proportion of Variance 0.3169586 0.07560125 0.05725007 0.05335644
0.04295121
## Cumulative Proportion  0.3169586 0.39255982 0.44980988 0.50316633
0.54611754
##                            Comp.6     Comp.7     Comp.8     Comp.9
Comp.10
## Standard deviation     0.94621901 0.89753016 0.88477112 0.86658594
0.84673356
## Proportion of Variance 0.03892741 0.03502436 0.03403565 0.03265092
0.03117207
## Cumulative Proportion  0.58504495 0.62006931 0.65410496 0.68675588
0.71792796
##                           Comp.11    Comp.12    Comp.13    Comp.14
Comp.15
## Standard deviation     0.82679364 0.81823075 0.78230274 0.76009061
0.74107189
## Proportion of Variance 0.02972121 0.02910876 0.02660859 0.02511903
0.02387772
```

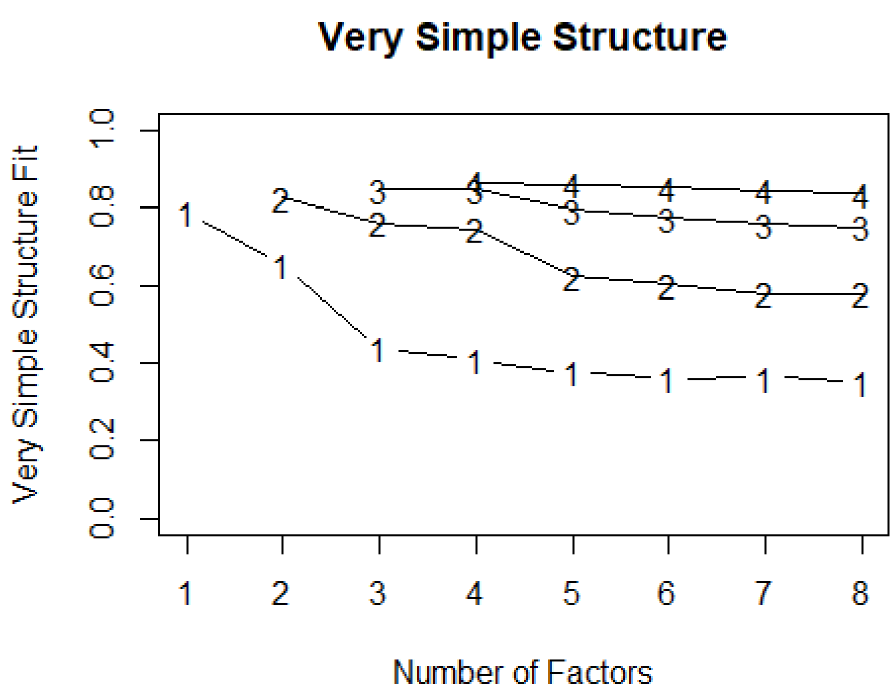
```
## Cumulative Proportion  0.74764916 0.77675793 0.80336652 0.82848555
0.85236327
##                            Comp.16    Comp.17    Comp.18    Comp.19
Comp.20
## Standard deviation     0.72329135 0.71301906 0.67523318 0.65100197
0.63858510
## Proportion of Variance 0.02274567 0.02210418 0.01982347 0.01842624
0.01773004
## Cumulative Proportion  0.87510894 0.89721312 0.91703659 0.93546283
0.95319287
##                            Comp.21    Comp.22    Comp.23
## Standard deviation     0.61601936 0.60334257 0.57711507
## Proportion of Variance 0.01649912 0.01582705 0.01448095
## Cumulative Proportion  0.96969200 0.98551905 1.00000000

screeplot(pca)
```




```
plot(pca,type='l')
```


pca
Variances
1
2
3
4
5
6
7
Comp.1
Comp.3
Comp.5
Comp.7
Comp.9

```
vss(df)
```


Very Simple Structure
Very Simple Structure Fit
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5
6
7
8
Number of Factors

```
##
## Very Simple Structure
## Call: vss(x = df)
## VSS complexity 1 achieves a maximimum of 0.79  with  1  factors
## VSS complexity 2 achieves a maximimum of 0.83  with  2  factors
##
## The Velicer MAP achieves a minimum of 0.01  with  1  factors
## BIC achieves a minimum of  -463.3  with  7  factors
## Sample Size adjusted BIC achieves a minimum of  -114.53  with  8  factors
##
## Statistics by number of factors
##   vss1 vss2   map dof chisq     prob sqresid  fit RMSEA  BIC SABIC complex
## 1 0.79 0.00 0.011 230  4463  0.0e+00    14.2 0.79 0.085 2657  3388     1.0
## 2 0.65 0.83 0.012 208  3161  0.0e+00    11.7 0.83 0.074 1528  2189     1.4
## 3 0.44 0.76 0.012 187  1957 3.8e-292    10.2 0.85 0.061  489  1083     1.8
## 4 0.41 0.74 0.013 167  1166 2.3e-149     8.9 0.87 0.048 -145   385     1.8
## 5 0.37 0.62 0.015 148   755  4.2e-82     8.2 0.88 0.040 -407    63     2.1
## 6 0.36 0.60 0.020 130   578  1.1e-57     7.7 0.88 0.037 -443   -30     2.1
## 7 0.36 0.58 0.025 113   424  1.6e-37     7.3 0.89 0.033 -463  -104     2.1
## 8 0.35 0.58 0.032  97   339  1.5e-28     7.2 0.89 0.031 -423  -115     2.4
##   eChisq  SRMR eCRMS eBIC
## 1   5547 0.065 0.068 3741
## 2   3114 0.049 0.054 1481
## 3   1789 0.037 0.043  320
## 4    880 0.026 0.032 -431
## 5    492 0.019 0.025 -670
## 6    364 0.017 0.023 -656
## 7    264 0.014 0.021 -623
## 8    199 0.012 0.020 -563
```

```r
# Selecting no of factors
r<-cor(df)
eval<-eigen(r)
eval$values
```

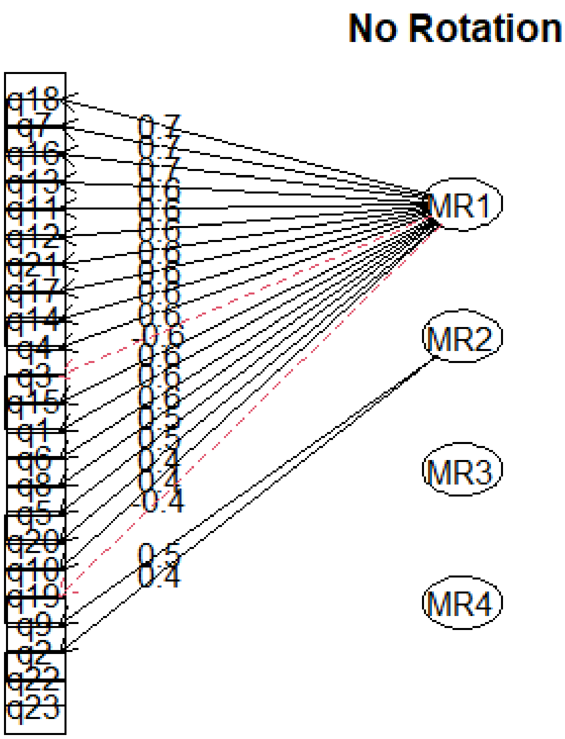
```
##  [1] 7.2900471 1.7388287 1.3167515 1.2271982 0.9878779 0.8953304 0.8055604
##  [8] 0.7828199 0.7509712 0.7169577 0.6835877 0.6695016 0.6119976 0.5777377
## [15] 0.5491875 0.5231504 0.5083962 0.4559399 0.4238036 0.4077909 0.3794799
## [22] 0.3640223 0.3330618
```

```r
nfactors=sum((eval$values>=1))
nfactors
```

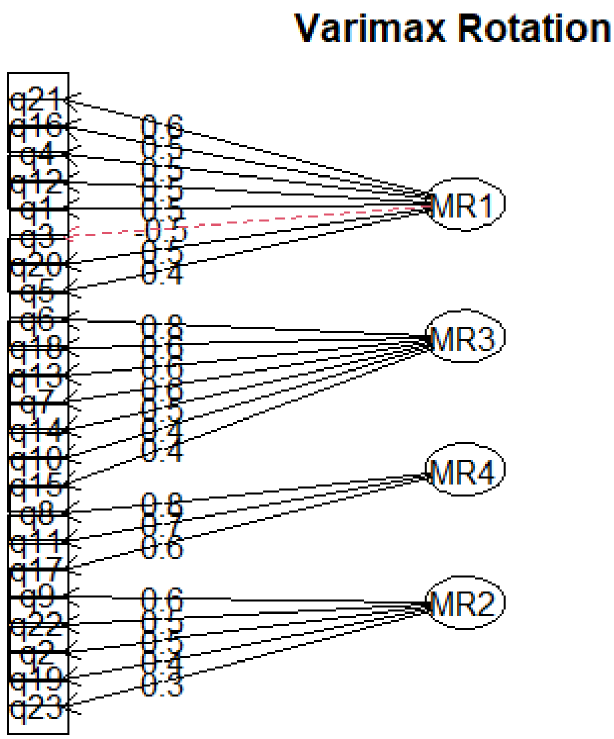
```
## [1] 4
```

Applying Rotation

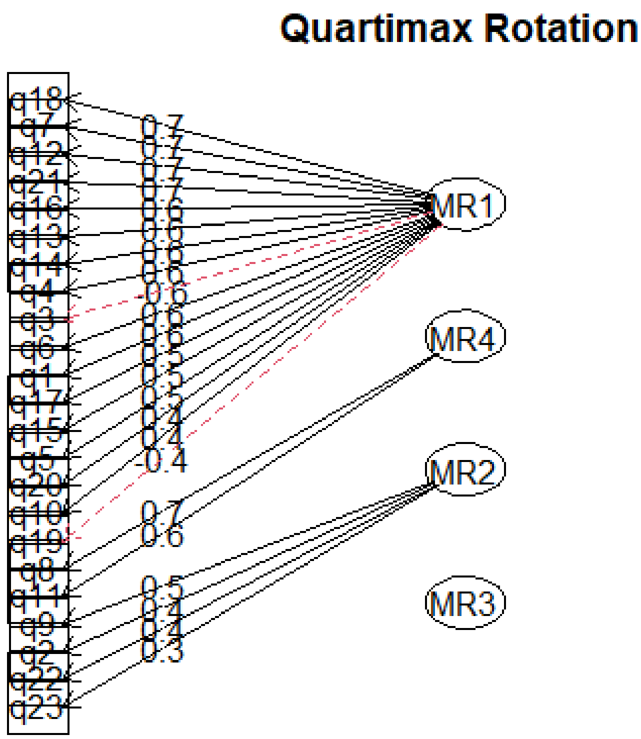
```r
# No rotation
library(psych)
fac<-fa(df,nfactors = 4,rotate = 'none')
load<-fac$loadings
fa.diagram(load,main = 'No Rotation')
```



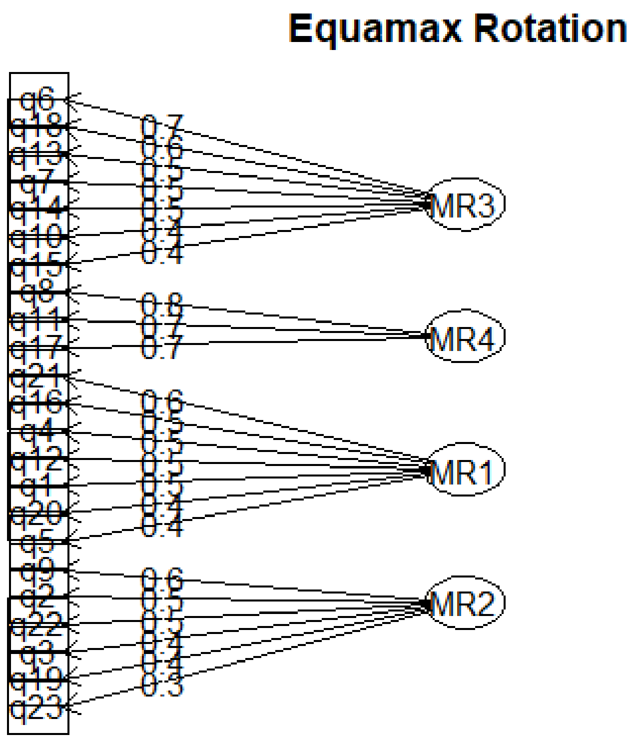

```r
# Varimax Rotation
fac1<-fa(df,nfactors = 4,rotate = 'varimax')
load1<-fac1$loadings
fa.diagram(load1,main = 'Varimax Rotation ')
```



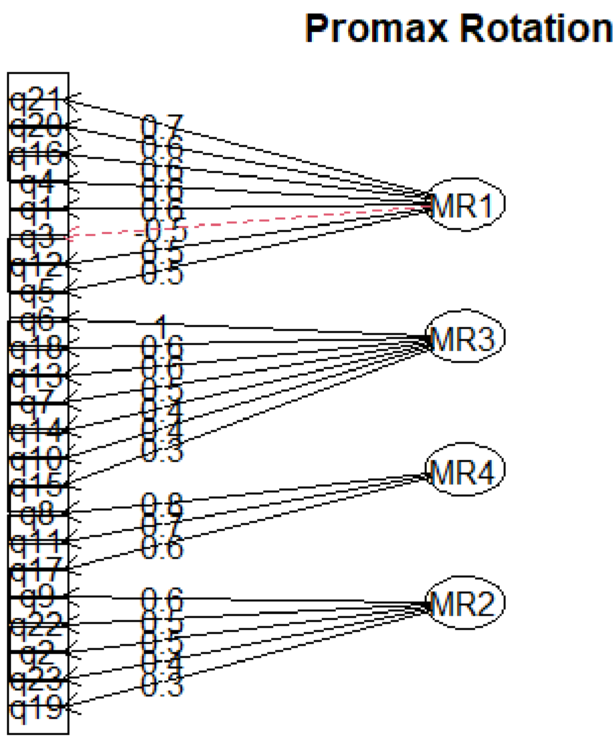

```
# Equamax Rotation
fac2<-fa(df,nfactors = 4,rotate = 'quartimax')
load2<-fac2$loadings
fa.diagram(load2,main = 'Quartimax Rotation')
```



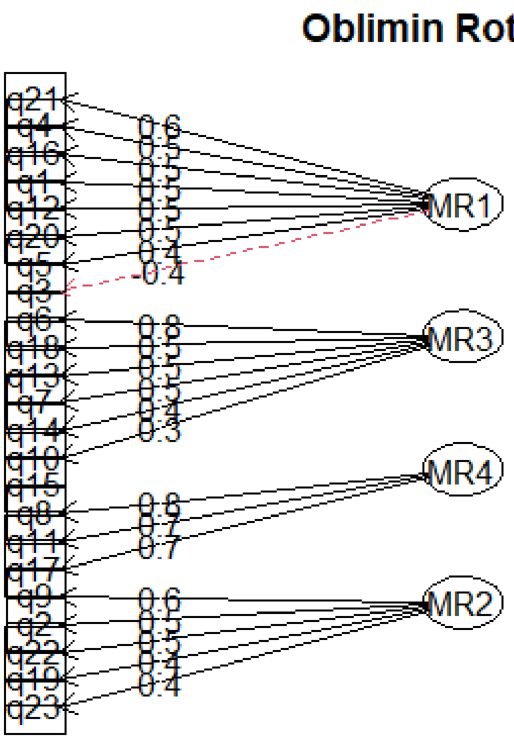

```r
# Quartimax Rotation
fac3<-fa(df,nfactors = 4,rotate = 'equamax')
load3<-fac3$loadings
fa.diagram(load3,main = 'Equamax Rotation')
```




```
# Promax Rotation
fac4<-fa(df,nfactors = 4,rotate = 'promax')
load4<-fac4$loadings
fa.diagram(load4,main = 'Promax Rotation')
```



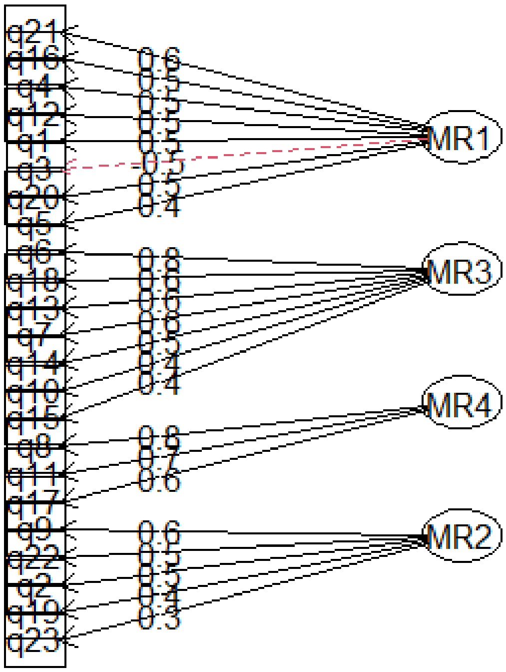

```r
# Oblimin Rotation
fac5<-fa(df,nfactors = 4,rotate = 'oblimin')
load5<-fac5$loadings
fa.diagram(load5,main = 'Oblimin Rotation')
```



Final Rotation

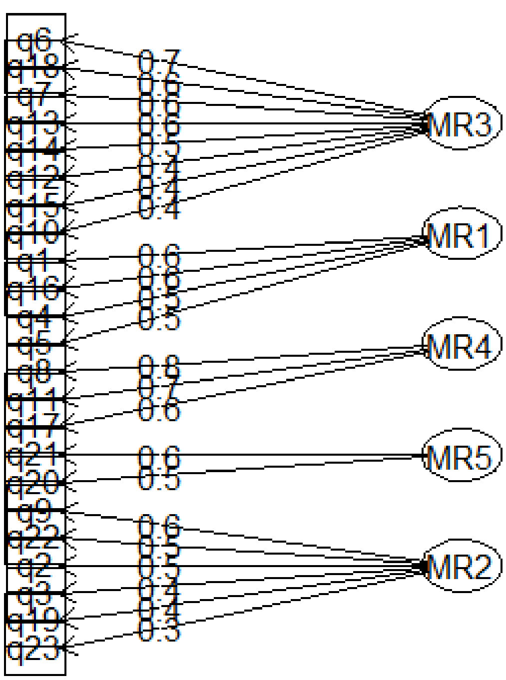
```
# Varimax Rotation
fac1<-fa(df,nfactors = 4,rotate = 'varimax')
load1<-fac1$loadings
fa.diagram(load1,main = 'Varimax Rotation 4')
```

Varimax Rotation 4

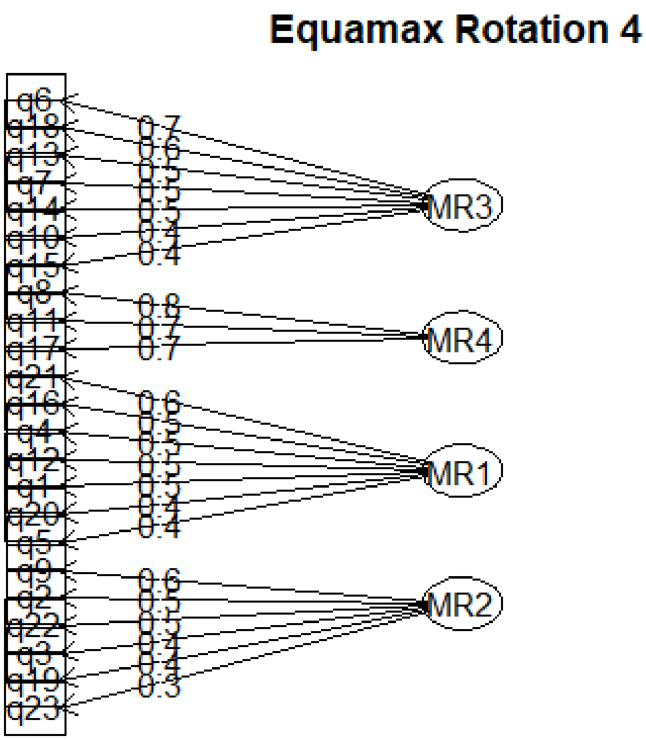



```r
fac1.2<-fa(df,nfactors = 5,rotate = 'varimax')
load1.2<-fac1.2$loadings
fa.diagram(load1.2,main = 'Varimax Rotation 5')
```

Varimax Rotation 5



```r
# Equamax Rotation
fac2<-fa(df,nfactors = 4,rotate = 'equamax')
load2<-fac2$loadings
fa.diagram(load2,main = 'Equamax Rotation 4')
```

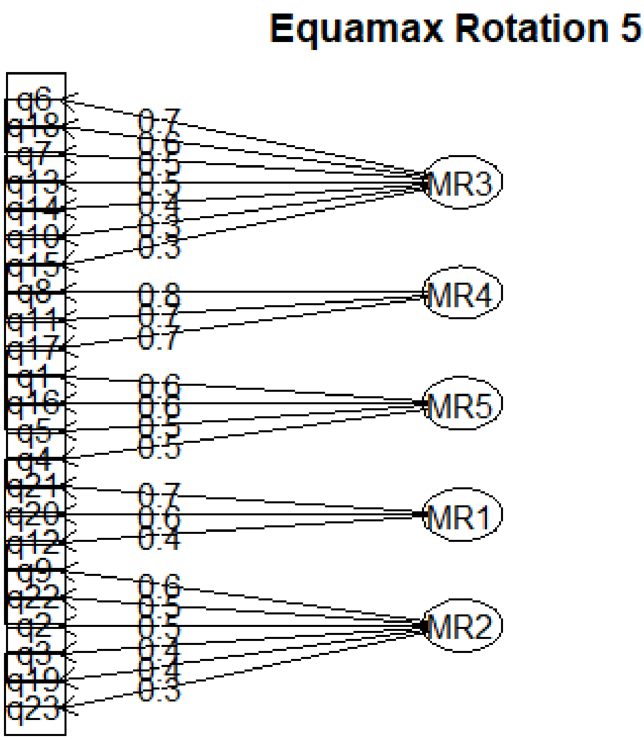



```r
load2
```

```
##
## Loadings:
##      MR3    MR4    MR1    MR2
## q1   0.185  0.335  0.473
## q2                -0.150  0.486
## q3  -0.165 -0.232 -0.435  0.449
## q4   0.249  0.323  0.497
## q5   0.242  0.252  0.411
## q6   0.747  0.173
## q7   0.540  0.235  0.343 -0.156
## q8          0.788  0.132
## q9                        0.570
## q10  0.370  0.174  0.122 -0.123
## q11  0.226  0.726  0.142 -0.172
## q12  0.377  0.191  0.484 -0.201
## q13  0.544  0.294  0.258 -0.157
## q14  0.466  0.221  0.365 -0.159
## q15  0.359  0.306  0.231 -0.216
## q16  0.250  0.324  0.497 -0.208
## q17  0.230  0.686  0.220
```

```
## q18  0.593  0.214  0.349 -0.154
## q19 -0.144 -0.105 -0.229  0.401
## q20         0.144  0.423 -0.256
## q21  0.239  0.234  0.558 -0.211
## q22 -0.166                0.459
## q23                       0.318
##
##                  MR3   MR4   MR1   MR2
## SS loadings    2.598 2.585 2.483 1.645
## Proportion Var 0.113 0.112 0.108 0.072
## Cumulative Var 0.113 0.225 0.333 0.405
```

```r
fac2.1<-fa(df,nfactors = 5,rotate = 'equamax')
load2.1<-fac2.1$loadings
fa.diagram(load2.1,main = 'Equamax Rotation 5')
```



Model Plotting

Model 1

```r
# Equamax Rotation 4 factor
d1<-fac2$scores
head(d1)
```

```
##            MR3         MR4         MR1        MR2
## 1  0.06057901 -1.50375962 -0.84483547 -0.2883108
```

```
## 2 -0.42972103 -0.25851828 -0.04427528  0.5098019
## 3 -0.56101995  0.05008314  0.01158021 -0.7351955
## 4  0.54530704 -0.50542679  0.73994020 -0.3874704
## 5  0.49227818 -0.42848073 -0.68069216  0.5801998
## 6  1.80786736 -0.37206286 -0.27661151 -0.3514139
```

```
colnames(d1)<-c('comp_fear','maths_fear','Stat_fear','peer_pressure')
a<-data[24]
head(a)
```

```
##   career
## 1      1
## 2      0
## 3      1
## 4      0
## 5      0
## 6      0
```

```
fdata1<-data.frame(d1,a)
fdata1$career<-as.factor(fdata1$career)
class(fdata1$career)
```

```
## [1] "factor"
```

```
head(fdata1)
```

```
##     comp_fear  maths_fear   Stat_fear peer_pressure career
## 1  0.06057901 -1.50375962 -0.84483547    -0.2883108      1
## 2 -0.42972103 -0.25851828 -0.04427528     0.5098019      0
## 3 -0.56101995  0.05008314  0.01158021    -0.7351955      1
## 4  0.54530704 -0.50542679  0.73994020    -0.3874704      0
## 5  0.49227818 -0.42848073 -0.68069216     0.5801998      0
## 6  1.80786736 -0.37206286 -0.27661151    -0.3514139      0
```

```
# splitting data
library(tidyverse) # to use %>% operator
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ lubridate 1.9.3     ✔ tibble    3.2.1
## ✔ purrr     1.0.2     ✔ tidyr     1.3.0
## ✔ readr     2.1.4
## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ ggplot2::%+%()   masks psych::%+%()
## ✖ ggplot2::alpha() masks psych::alpha()
## ✖ dplyr::filter()  masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
```

```
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to force
all conflicts to become errors
```

```r
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
set.seed(100)
ndata<-fdata1$career %>%
  createDataPartition(p=0.7, list=FALSE)
train_data1<-fdata1[ndata,]
test_data1<-fdata1[-ndata,]


# Regression model
model1<-
glm(career~comp_fear+maths_fear+Stat_fear+peer_pressure,family='binomial',tra
in_data1)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(model1)
```

```
##
## Call:
## glm(formula = career ~ comp_fear + maths_fear + Stat_fear + peer_pressure,
##     family = "binomial", data = train_data1)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -4.8994     0.3565  -13.74   <2e-16 ***
## comp_fear       -6.7632     0.4871  -13.89   <2e-16 ***
## maths_fear      -5.9170     0.4407  -13.43   <2e-16 ***
## Stat_fear       -6.9658     0.4989  -13.96   <2e-16 ***
## peer_pressure   -3.1200     0.2573  -12.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2325.41  on 1800  degrees of freedom
## Residual deviance:  464.83  on 1796  degrees of freedom
## AIC: 474.83
##
## Number of Fisher Scoring iterations: 9
```

```
# model accuracy checking
library(caret)
class(test_data1$career)

## [1] "factor"

pred1<-predict(model1,test_data1,type = 'response')
predicted1<-ifelse(pred1 >0.50,1,0)
predicted1<-as.factor(predicted1)
class(predicted1)

## [1] "factor"

# confusion matrix using formula
cm1<-confusionMatrix(data = predicted1, reference = test_data1$career)
cm1$table

##           Reference
## Prediction   0   1
##          0 484  24
##          1  19 243

# Accuracy
accuracy1<-(484+243)/(484+24+19+243)
accuracy1

## [1] 0.9441558
```

Model 2

```
# Equamax Rotation 5 factor
d2<-fac2.1$scores
head(d2)

##           MR3         MR4          MR5         MR1        MR2
## 1  0.02970382 -1.52662761 -0.087772604 -1.16837886 -0.5113565
## 2 -0.35555757 -0.18566857 -0.647544642  0.49669001  0.6610181
## 3 -0.57681013  0.02910983  0.005053944  0.03286161 -0.7439372
## 4  0.49539057 -0.58509527  0.565995101  0.54880799 -0.3279506
## 5  0.58604961 -0.29283897 -0.810617186 -0.31680406  0.5689039
## 6  1.87514229 -0.27795267 -0.584780983  0.25392457 -0.3160687

colnames(d2)<-
c('comp_fear','maths_fear','desc_stat','stat_application','peer_pressure')
b<-data[24]
head(b)

##   career
## 1      1
## 2      0
## 3      1
## 4      0
```

```
## 5      0
## 6      0

fdata2<-data.frame(d2,b)
fdata2$career<-as.factor(fdata2$career)
class(fdata2$career)

## [1] "factor"

head(fdata2)

##      comp_fear  maths_fear    desc_stat stat_application peer_pressure
career
## 1  0.02970382 -1.52662761 -0.087772604      -1.16837886    -0.5113565
1
## 2 -0.35555757 -0.18566857 -0.647544642       0.49669001     0.6610181
0
## 3 -0.57681013  0.02910983  0.005053944       0.03286161    -0.7439372
1
## 4  0.49539057 -0.58509527  0.565995101       0.54880799    -0.3279506
0
## 5  0.58604961 -0.29283897 -0.810617186      -0.31680406     0.5689039
0
## 6  1.87514229 -0.27795267 -0.584780983       0.25392457    -0.3160687
0

# splitting data
library(tidyverse) # to use %>% operator
library(caret)
set.seed(101)
ndata<-fdata2$career %>%
  createDataPartition(p=0.7, list=FALSE)
train_data2<-fdata2[ndata,]
test_data2<-fdata2[-ndata,]

# Regression model (equamax-5)
model2<-
glm(career~comp_fear+maths_fear+desc_stat+stat_application+peer_pressure,fami
ly='binomial',train_data2)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(model2)

##
## Call:
## glm(formula = career ~ comp_fear + maths_fear + desc_stat +
stat_application +
##     peer_pressure, family = "binomial", data = train_data2)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)       -5.1808     0.3785  -13.69   <2e-16 ***
## comp_fear         -7.0053     0.5128  -13.66   <2e-16 ***
## maths_fear        -5.7866     0.4473  -12.94   <2e-16 ***
## desc_stat         -6.2605     0.4682  -13.37   <2e-16 ***
## stat_application  -4.8374     0.3731  -12.97   <2e-16 ***
## peer_pressure     -3.7016     0.3029  -12.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2325.41  on 1800  degrees of freedom
## Residual deviance:  438.91  on 1795  degrees of freedom
## AIC: 450.91
##
## Number of Fisher Scoring iterations: 9
```

```
# model accuracy checking
library(caret)
class(test_data2$career)
```

```
## [1] "factor"
```

```
pred2<-predict(model2,test_data2,type = 'response')
predicted2<-ifelse(pred2 >0.50,1,0)
predicted2<-as.factor(predicted2)
class(predicted2)
```

```
## [1] "factor"
```

```
# confusion matrix using formula
cm2<-confusionMatrix(data = predicted2, reference = test_data2$career)
cm2$table
```

```
##           Reference
## Prediction   0   1
##          0 483  19
##          1  20 248
```

```
# Accuracy
accuracy2<-(483+248)/(483+20+19+248)
accuracy2
```

```
## [1] 0.9493506
```

Model 3

```
# Varimax rotation 4
d3<-fac1$scores
head(d3)
```

```
##          MR1         MR3         MR4        MR2
## 1 -0.9730817 -0.06019129 -1.40486535 -0.3710462
```

```
## 2 -0.1213199 -0.43101309 -0.21416197  0.5167146
## 3  0.1272467 -0.57627951  0.05835085 -0.7114799
## 4  0.6989150  0.52828661 -0.62627191 -0.3011235
## 5 -0.8087034  0.46052478 -0.35703599  0.4820252
## 6 -0.3353930  1.76374832 -0.44386922 -0.4318615
```

```
colnames(d3)<-c('Stat_fear','comp_fear','maths_fear','peer_pressure')
c<-data[24]
head(c)
```

```
##   career
## 1      1
## 2      0
## 3      1
## 4      0
## 5      0
## 6      0
```

```
fdata3<-data.frame(d3,c)
fdata3$career<-as.factor(fdata3$career)
class(fdata3$career)
```

```
## [1] "factor"
```

```
head(fdata3)
```

```
##    Stat_fear   comp_fear  maths_fear peer_pressure career
## 1 -0.9730817 -0.06019129 -1.40486535    -0.3710462      1
## 2 -0.1213199 -0.43101309 -0.21416197     0.5167146      0
## 3  0.1272467 -0.57627951  0.05835085    -0.7114799      1
## 4  0.6989150  0.52828661 -0.62627191    -0.3011235      0
## 5 -0.8087034  0.46052478 -0.35703599     0.4820252      0
## 6 -0.3353930  1.76374832 -0.44386922    -0.4318615      0
```

```
# splitting data
library(tidyverse) # to use %>% operator
library(caret)
set.seed(102)
ndata<-fdata3$career %>%
  createDataPartition(p=0.7, list=FALSE)
train_data3<-fdata3[ndata,]
test_data3<-fdata3[-ndata,]

# Regression model
model3<-
glm(career~Stat_fear+comp_fear+maths_fear+peer_pressure,family='binomial',tra
in_data3)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model3)
```

```
##
## Call:
## glm(formula = career ~ Stat_fear + comp_fear + maths_fear + peer_pressure,
##     family = "binomial", data = train_data3)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -5.0861     0.3741  -13.60   <2e-16 ***
## Stat_fear      -7.1600     0.5232  -13.69   <2e-16 ***
## comp_fear      -7.8590     0.5727  -13.72   <2e-16 ***
## maths_fear     -4.9138     0.3927  -12.51   <2e-16 ***
## peer_pressure  -3.8677     0.3110  -12.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2325.41  on 1800  degrees of freedom
## Residual deviance:  428.93  on 1796  degrees of freedom
## AIC: 438.93
##
## Number of Fisher Scoring iterations: 9
```

```r
# model accuracy checking
library(caret)
class(test_data3$career)
```

```
## [1] "factor"
```

```r
pred3<-predict(model3,test_data3,type = 'response')
predicted3<-ifelse(pred3 >0.50,1,0)
predicted3<-as.factor(predicted3)
class(predicted3)
```

```
## [1] "factor"
```

```r
# confusion matrix using formula
cm3<-confusionMatrix(data = predicted3, reference = test_data3$career)
cm3$table
```

```
##           Reference
## Prediction   0   1
##          0 479  26
##          1  24 241
```

```r
# Accuracy
accuracy3<-(479+241)/(479+26+24+241)
accuracy3
```

```
## [1] 0.9350649
```

model 4

```
# Varimax Rotation 5
d4<-fac1.2$scores
head(d4)

##          MR3         MR1         MR4        MR5        MR2
## 1 -0.1577716 -0.22326138 -1.50955861 -1.1628409 -0.5099698
## 2 -0.3504339 -0.63303479 -0.12215475  0.5478697  0.6520997
## 3 -0.5763189  0.08011783  0.03652082  0.0965459 -0.7340899
## 4  0.5483024  0.52827459 -0.65526149  0.4718816 -0.2917039
## 5  0.4934888 -0.90449936 -0.24542007 -0.3341677  0.5260952
## 6  1.8304723 -0.70834026 -0.32314993  0.1231236 -0.3574944

colnames(d4)<-
c('comp_fear','stat_application','maths_fear','desc_stat','peer_pressure')
d<-data[24]
head(d)

##   career
## 1      1
## 2      0
## 3      1
## 4      0
## 5      0
## 6      0

fdata4<-data.frame(d4,d)
fdata4$career<-as.factor(fdata4$career)
class(fdata4$career)

## [1] "factor"

head(fdata4)

##    comp_fear stat_application  maths_fear  desc_stat peer_pressure career
## 1 -0.1577716      -0.22326138 -1.50955861 -1.1628409    -0.5099698      1
## 2 -0.3504339      -0.63303479 -0.12215475  0.5478697     0.6520997      0
## 3 -0.5763189       0.08011783  0.03652082  0.0965459    -0.7340899      1
## 4  0.5483024       0.52827459 -0.65526149  0.4718816    -0.2917039      0
## 5  0.4934888      -0.90449936 -0.24542007 -0.3341677     0.5260952      0
## 6  1.8304723      -0.70834026 -0.32314993  0.1231236    -0.3574944      0

# splitting data
library(tidyverse) # to use %>% operator
library(caret)
set.seed(103)
ndata<-fdata4$career %>%
  createDataPartition(p=0.7, list=FALSE)
train_data4<-fdata4[ndata,]
test_data4<-fdata4[-ndata,]
```

```
# Regression model
model4<-
glm(career~comp_fear+maths_fear+desc_stat+stat_application+peer_pressure,fami
ly='binomial',train_data4)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(model4)

##
## Call:
## glm(formula = career ~ comp_fear + maths_fear + desc_stat +
stat_application +
##     peer_pressure, family = "binomial", data = train_data4)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -4.9376     0.3615  -13.66   <2e-16 ***
## comp_fear          -7.7953     0.5615  -13.88   <2e-16 ***
## maths_fear         -4.7480     0.3878  -12.24   <2e-16 ***
## desc_stat          -3.6850     0.3077  -11.97   <2e-16 ***
## stat_application  -5.8875     0.4431  -13.29   <2e-16 ***
## peer_pressure      -3.7599     0.3140  -11.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2325.41  on 1800  degrees of freedom
## Residual deviance:  428.31  on 1795  degrees of freedom
## AIC: 440.31
##
## Number of Fisher Scoring iterations: 9

# model accuracy checking
library(caret)
class(test_data4$career)

## [1] "factor"

pred4<-predict(model4,test_data4,type = 'response')
predicted4<-ifelse(pred4 >0.50,1,0)
predicted4<-as.factor(predicted4)
class(predicted4)

## [1] "factor"

# confusion matrix using formula
cm4<-confusionMatrix(data = predicted4, reference = test_data4$career)
cm4$table
```

```
##            Reference
## Prediction   0   1
##          0 479  23
##          1  24 244
```

```r
# Accuracy
accuracy4<-(479+244)/(479+23+24+244)
accuracy4
```

```
## [1] 0.938961
```

Best Model

```r
# Best model
model1$aic;accuracy1
```

```
## [1] 474.8296
```

```
## [1] 0.9441558
```

```r
model2$aic;accuracy2
```

```
## [1] 450.9133
```

```
## [1] 0.9493506
```

```r
model3$aic;accuracy3
```

```
## [1] 438.9283
```

```
## [1] 0.9350649
```

```r
model4$aic;accuracy4
```

```
## [1] 440.3084
```

```
## [1] 0.938961
```