Multivariate Analysis Final Exam

DA 410

Marjorie Blanco

Problem 1

Suppose our multivariate data have `covariance` matrix

S =

5	0	0
0	9	0
0	0	8

(a) Find the eigenvalues and eigenvectors of S.

The off-diagonal values are the covariances between variables.

```
res.eig <- eigen(S)
res.eig
```

```
## eigen() decomposition
## $values
## [1] 9 8 5
##
## $vectors
##      [,1] [,2] [,3]
## [1,]    0    0    1
## [2,]    1    0    0
## [3,]    0    1    0
```

The eigenvalues of S are $\lambda_1 = 9$, $\lambda_2 = 8$, and $\lambda_3 = 5$. The eigenvectors of S are:

$a_1 = 0, 1, 0$

$a_2 = 0, 0, 1$

$a_3 = 1, 0, 0$

The sum of the eigenvalues from a correlation matrix will equal the number of variables. The sum of the eigenvalues will equal 22.

The first two principal components are:

$Z_l = 1\ y_2$

$Z_2 = 1\ y_3$

$Z_3 = 1\ y_1$

(b) Show the percent of variance explained.

```
res.eig$values/(sum(res.eig$values))
```

```
## [1] 0.4090909 0.3636364 0.2272727
```

The SS loadings = 9, which is the eigenvalue for the first principal component. The proportion variance = $\frac{9}{22} = 41\%$.

The SS loadings = 8, which is the eigenvalue for the second principal component. The proportion variance = $\frac{8}{22} = 36\%$.

The SS loadings = 5, which is the eigenvalue for the third principal component. The proportion variance = $\frac{5}{22} = 23\%$.

(c) Decide how many components to retain.

Method 1: % of variance

An appropriate threshold percentage should be selected prior to starting the process. If we want to explain at least 50% of variance then we would select PC1 and PC2. Selecting PC3 is not recomended.

Problem 2

The correlation matrix given below arises from the scores of 220 boys in six school subjects: (1) French, (2) English, (3) history, (4) arithmetic, (5) algebra, and (6) geometry. Obtain principal component loadings for three factors.

```
# grades correlation matrix
gr <- c(0.44,
        0.41, 0.35,
        0.29, 0.35, 0.16,
        0.33, 0.32, 0.19, 0.59,
        0.25, 0.33, 0.18, 0.47, 0.46)
```

```
grades.cor <- diag(6) / 2
grades.cor[upper.tri(grades.cor)] <- gr
grades.cor <- grades.cor + t(grades.cor)
rownames(grades.cor) <- colnames(grades.cor) <-
  c("French", "English", "History", "Arithmetic", "Algebra", "Geometry")
```

Step 1: Find correlation matrix R.

R =

	French	English	History	Arithmetic	Algebra	Geometry
French	1.00	0.44	0.41	0.29	0.33	0.25
English	0.44	1.00	0.35	0.35	0.32	0.33
History	0.41	0.35	1.00	0.16	0.19	0.18
Arithmetic	0.29	0.35	0.16	1.00	0.59	0.47
Algebra	0.33	0.32	0.19	0.59	1.00	0.46
Geometry	0.25	0.33	0.18	0.47	0.46	1.00

```
s <- factanal(covmat = grades.cor,
              factors = 3,
              method = "mle",
              n.obs = 220)
s
```

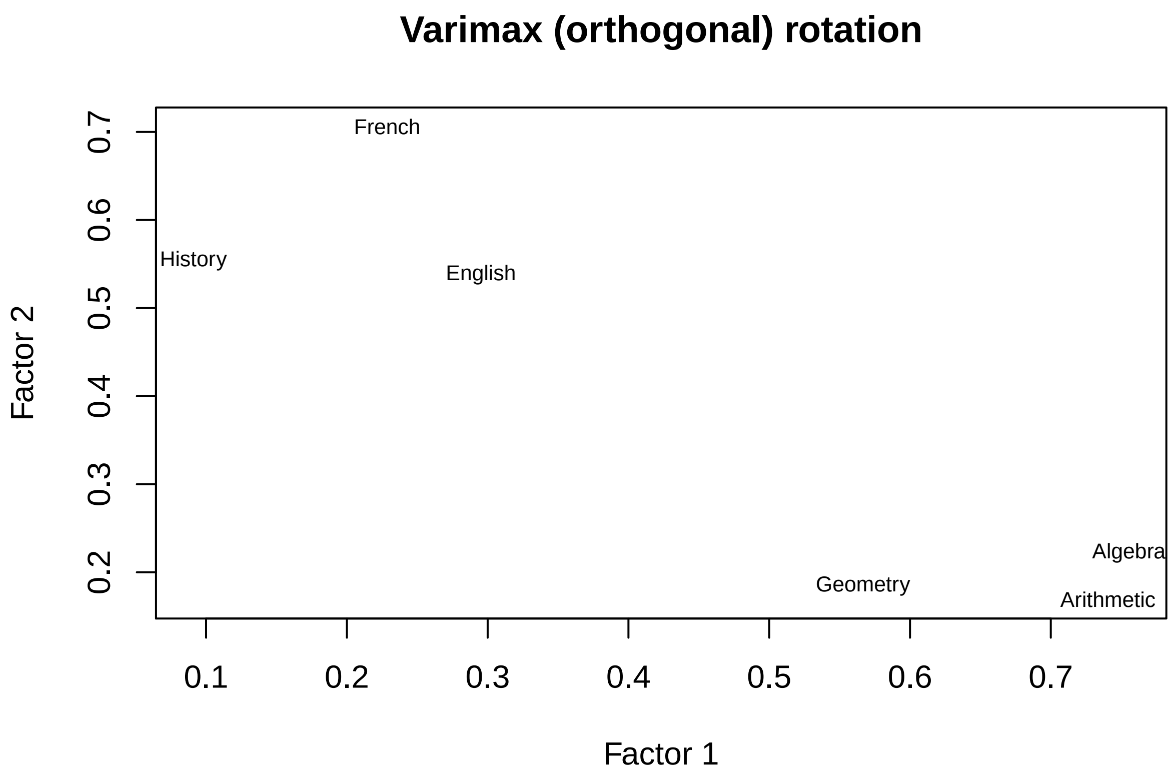
```
##
## Call:
## factanal(factors = 3, covmat = grades.cor, n.obs = 220, method = "mle")
##
## Uniquenesses:
##     French    English    History Arithmetic    Algebra   Geometry
##      0.448      0.497      0.679      0.411      0.376      0.611
##
## Loadings:
##            Factor1 Factor2 Factor3
## French      0.229   0.706
## English     0.295   0.538   0.355
## History             0.554
## Arithmetic  0.741   0.169   0.109
## Algebra     0.756   0.222
## Geometry    0.567   0.185   0.183
##
##                Factor1 Factor2 Factor3
## SS loadings      1.588   1.207   0.181
## Proportion Var   0.265   0.201   0.030
## Cumulative Var   0.265   0.466   0.496
##
## The degrees of freedom for the model is 0 and the fit was 0.001
```

```
loadings(s)
```

```
##
## Loadings:
##            Factor1 Factor2 Factor3
## French      0.229   0.706
## English     0.295   0.538   0.355
## History             0.554
## Arithmetic  0.741   0.169   0.109
## Algebra     0.756   0.222
## Geometry    0.567   0.185   0.183
##
##                Factor1 Factor2 Factor3
## SS loadings      1.588   1.207   0.181
## Proportion Var   0.265   0.201   0.030
## Cumulative Var   0.265   0.466   0.496
```

Algebra and Arithmetic dominates the first factor. French dominates the second factor. English weakly dominates the third factor.

```
plot(s$loadings[,1], s$loadings[,2],
     type = "n", xlab = "Factor 1",
     ylab = "Factor 2",
     main="Varimax (orthogonal) rotation")

text(s$loadings[,1], s$loadings[,2],
     rownames(grades.cor), cex = 0.7, xlim = c(-2, 1.5))
```



Math skills such Arithmetic, Algebra and Geometry dominate the first factor. Non-math skills such as French, History and English dominate the second factor.

Step 2: Find the eigenvalue D and eigenvectors C of R.

```
# Then use that correlation matrix to calculate eigenvalues
res.eig <- eigen(grades.cor, symmetric = FALSE)
res.eig
```

```
## eigen() decomposition
## $values
## [1] 2.7286835 1.1287922 0.6152914 0.6028089 0.5225144 0.4019097
##
## $vectors
##           [,1]       [,2]        [,3]       [,4]       [,5]         [,6]
## [1,] 0.4000324  0.4183504  0.20734324 -0.4554101  0.6288251 -0.138370623
## [2,] 0.4167784  0.2727390  0.67736728  0.3473498 -0.3831532  0.160076491
## [3,] 0.3125882  0.6019621 -0.66187338  0.1452987 -0.2825319 -0.030296030
## [4,] 0.4453246 -0.3925077 -0.02272541 -0.2333619 -0.3360834 -0.692601924
## [5,] 0.4491070 -0.3508495 -0.16947568 -0.3937351 -0.1299707  0.688887359
## [6,] 0.4105452 -0.3332870 -0.17569157  0.6643440  0.4981003 -0.006929692
```

Step 3: Find C_1 and D_1

```
c.1 <- res.eig$vectors[,1:3]
d.1 <- diag(res.eig$values[1:3])
```

C_1=

0.4000324	0.4183504	0.2073432
0.4167784	0.2727390	0.6773673
0.3125882	0.6019621	-0.6618734
0.4453246	-0.3925077	-0.0227254
0.4491070	-0.3508495	-0.1694757
0.4105452	-0.3332870	-0.1756916

D_1=

2.728683	0.000000	0.0000000
0.000000	1.128792	0.0000000
0.000000	0.000000	0.6152914

Step 4: Find $C_1D_1^{1/2}$

```
l <- as.data.frame(c.1 %*% sqrt(d.1))
```

$C_1D_1^{1/2}$=

V1	V2	V3
0.66	0.44	0.16
0.69	0.29	0.53
0.52	0.64	-0.52
0.74	-0.42	-0.02
0.74	-0.37	-0.13
0.68	-0.35	-0.14

Step 5: Obtain loadings

```
l[,4] <- l[,1]^2 + l[,2]^2 + l[,3]^2
l[,5] <- 1 - l[,4]
```

```
prop <- res.eig$values[1:3]/sum(res.eig$values)
cumprop <- c(prop[1], sum(prop))
cumulative.proportion <- 0
prop <- c()
cumulative <- c()
for (i in res.eig$values) {
  proportion <- i / dim(data)[2]
  cumulative.proportion <- cumulative.proportion + proportion
  prop <- append(prop, proportion)
  cumulative <- append(cumulative, cumulative.proportion)
}
data.frame(cbind(prop, cumulative))
```

```
## [1] prop       cumulative
## <0 rows> (or 0-length row.names)
```

```
factors <- t(t(res.eig$vectors[,1:3]) * sqrt(res.eig$values[1:3]))
round(factors, 2)
```

```
##      [,1]  [,2]  [,3]
## [1,] 0.66  0.44  0.16
## [2,] 0.69  0.29  0.53
## [3,] 0.52  0.64 -0.52
## [4,] 0.74 -0.42 -0.02
## [5,] 0.74 -0.37 -0.13
## [6,] 0.68 -0.35 -0.14
```

```
#compute the communality remains the same as in the covariance setting.
h2 <- rowSums(factors^2)
h2
```

```
## [1] 0.6606702 0.8402628 0.9451954 0.7153583 0.7069891 0.6042914
```

```
u2 <- 1 - h2
u2
```

```
## [1] 0.33932984 0.15973722 0.05480456 0.28464172 0.29301094 0.39570863
```

```
com <- rowSums(factors^2)^2 / rowSums(factors^4)
com
```

```
## [1] 1.894459 2.267232 2.872228 1.583977 1.549758 1.604416
```

```
mean(com)
```

```
## [1] 1.962012
```

	PC1	PC2	PC3	h2	u2	com
French	0.66	0.44	0.16	0.66	0.339	1.9
English	0.69	0.29	0.53	0.84	0.160	2.3
History	0.52	0.64	-0.52	0.95	0.055	2.9
Arithmetic	0.74	-0.42	-0.02	0.72	0.285	1.6
Algebra	0.74	-0.37	-0.13	0.71	0.293	1.5
Geometry	0.68	-0.35	-0.14	0.60	0.396	1.6

```
grades.pc <- principal(grades.cor, nfactors = 3, rotate = 'none', covar = TRUE)
grades.pc
```

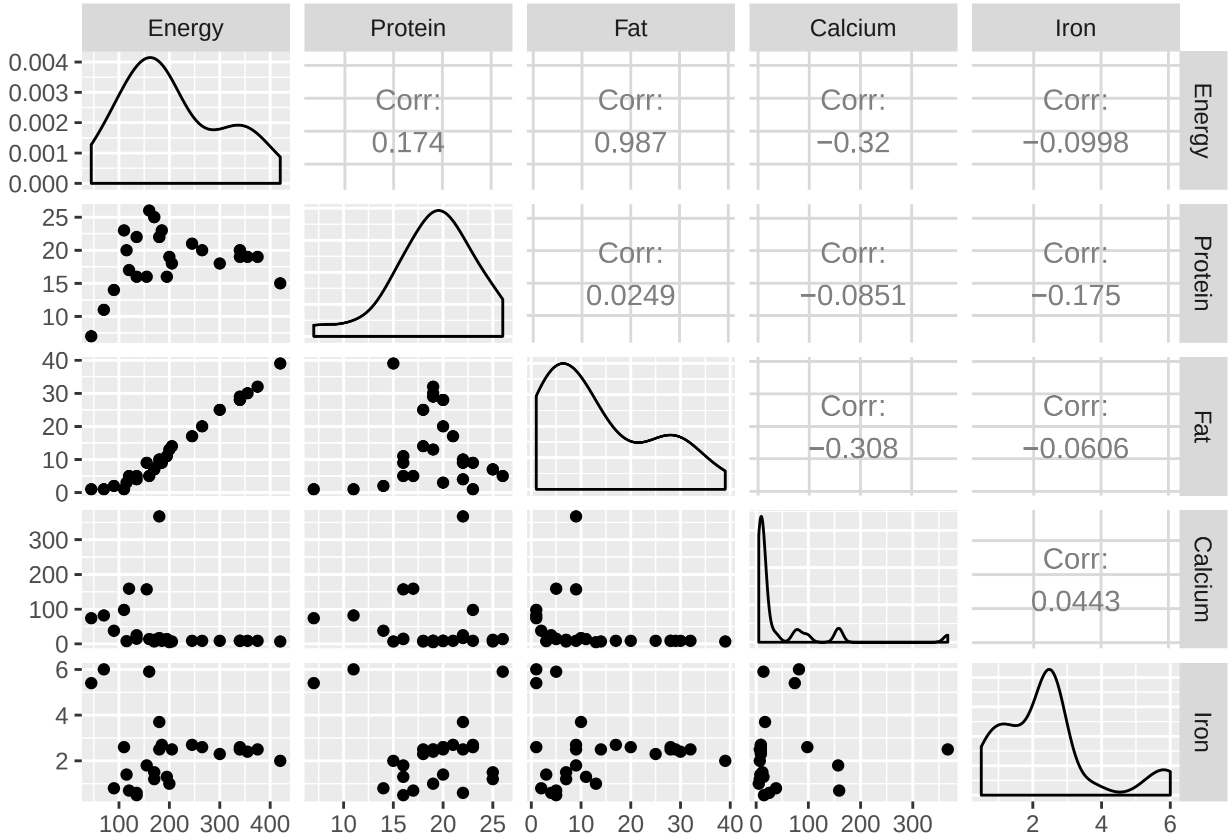
```
## Principal Components Analysis
## Call: principal(r = grades.cor, nfactors = 3, rotate = "none", covar = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##             PC1   PC2   PC3   h2    u2 com
## French     0.66  0.44 -0.16 0.66 0.339 1.9
## English    0.69  0.29 -0.53 0.84 0.160 2.3
## History    0.52  0.64  0.52 0.95 0.055 2.9
## Arithmetic 0.74 -0.42  0.02 0.72 0.285 1.6
## Algebra    0.74 -0.37  0.13 0.71 0.293 1.5
## Geometry   0.68 -0.35  0.14 0.60 0.396 1.6
##
##                        PC1  PC2  PC3
## SS loadings           2.73 1.13 0.62
## Proportion Var        0.45 0.19 0.10
## Cumulative Var        0.45 0.64 0.75
## Proportion Explained  0.61 0.25 0.14
## Cumulative Proportion 0.61 0.86 1.00
##
## Mean item complexity =  2
## Test of the hypothesis that 3 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.11
##
## Fit based upon off diagonal values = 0.92
```

Problem 3

For the Foodstuff Contents data set below:

(a) Discuss your choice of the number of factors.

```
foodstuff <- read_csv("data/Problem3-dataset.csv")
foodstuff <- foodstuff %>% remove_rownames %>% column_to_rownames(var="Food")
ggpairs(foodstuff, progress = FALSE)
```



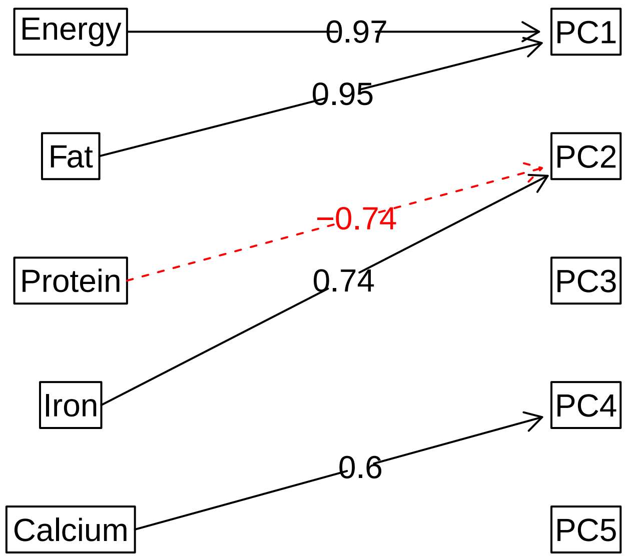
```
s <- factanal(foodstuff,
              factors = 2,
              #method = "mle",
              rotation = "none",
              n.obs = nrow(foodstuff))
s
```

```
##
## Call:
## factanal(x = foodstuff, factors = 2, n.obs = nrow(foodstuff),     rotation = "none")
##
## Uniquenesses:
##  Energy Protein     Fat Calcium    Iron
##   0.005   0.005   0.005   0.897   0.965
##
## Loadings:
##         Factor1 Factor2
## Energy   0.998
## Protein  0.197   0.978
## Fat      0.983  -0.172
## Calcium -0.319
## Iron            -0.160
##
##                Factor1 Factor2
## SS loadings      2.113   1.013
## Proportion Var   0.423   0.203
## Cumulative Var   0.423   0.625
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 10.67 on 1 degree of freedom.
## The p-value is 0.00109
```

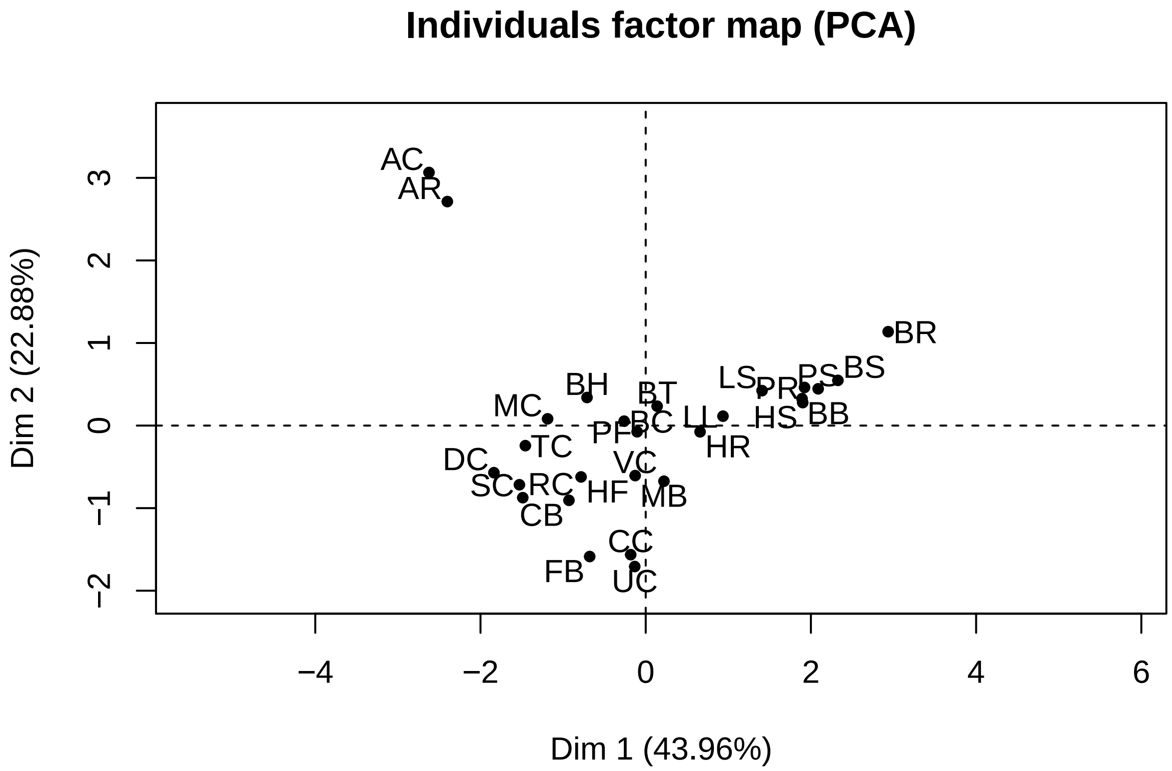
Energy and Fat dominates the first factor. Protein dominates the second factor.

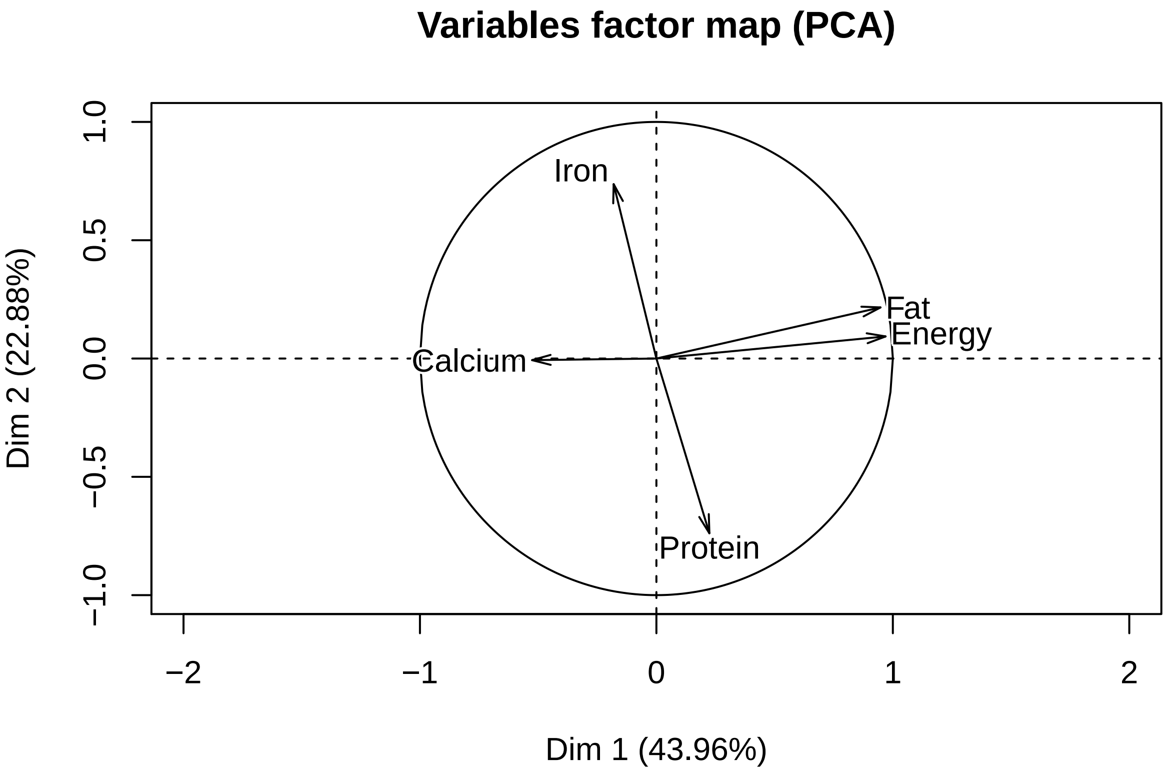
```
foodstuff.pc <- principal(foodstuff, nfactors = 5, rotate = 'none', covar = FALSE)
foodstuff.pc
```

```
## Principal Components Analysis
## Call: principal(r = foodstuff, nfactors = 5, rotate = "none", covar = FALSE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##           PC1   PC2   PC3  PC4   PC5 h2       u2 com
## Energy   0.97  0.09  0.14 0.18 -0.03  1  0.0e+00 1.1
## Protein  0.22 -0.74 -0.43 0.47  0.00  1 -3.8e-15 2.6
## Fat      0.95  0.22  0.20 0.12  0.03  1 -2.2e-15 1.2
## Calcium -0.53 -0.01  0.60 0.60  0.00  1  1.8e-15 3.0
## Iron    -0.18  0.74 -0.50 0.42  0.00  1 -3.8e-15 2.6
##
##                        PC1  PC2  PC3  PC4 PC5
## SS loadings           2.20 1.14 0.85 0.81   0
## Proportion Var        0.44 0.23 0.17 0.16   0
## Cumulative Var        0.44 0.67 0.84 1.00   1
## Proportion Explained  0.44 0.23 0.17 0.16   0
## Cumulative Proportion 0.44 0.67 0.84 1.00   1
##
## Mean item complexity =  2.1
## Test of the hypothesis that 5 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```



Components Analysis


Individuals factor map (PCA)
Dim 1 (43.96%)
Dim 2 (22.88%)
−4
−2
0
2
4
6
−2
−1
0
1
2
3
AC
AR
BR
BS
PS
PR
LS
BB
HS
LI
HR
BT
BC
PF
BH
MC
TC
DC
SC
RC
HF
CB
VC
MB
CC
FB
UC



The first component consist of the average energy and fat content of the food. The second component consist of the average protein and Iron content of the food.

Method 1: % of variance

An appropriate threshold percentage should be selected prior to starting the process. If we want to explain at least 50% of variance then we would select PC1 and PC2.

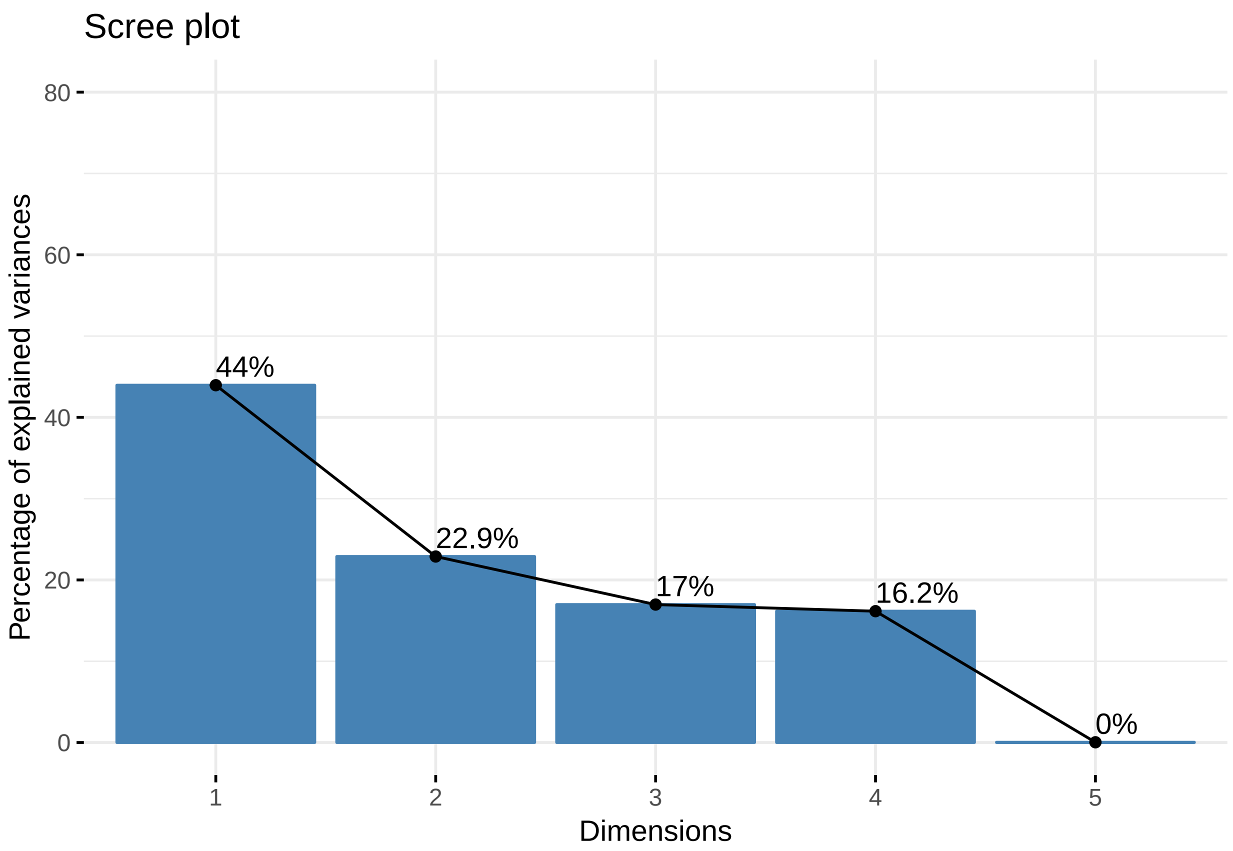
Method 2: Kaiser's criterion

Components with SS loadings > 1. In the foodstuff data, retaining only PC1 and PC2 is recomended. The SS loading for PC3, PC4 and PC5 is < 1.

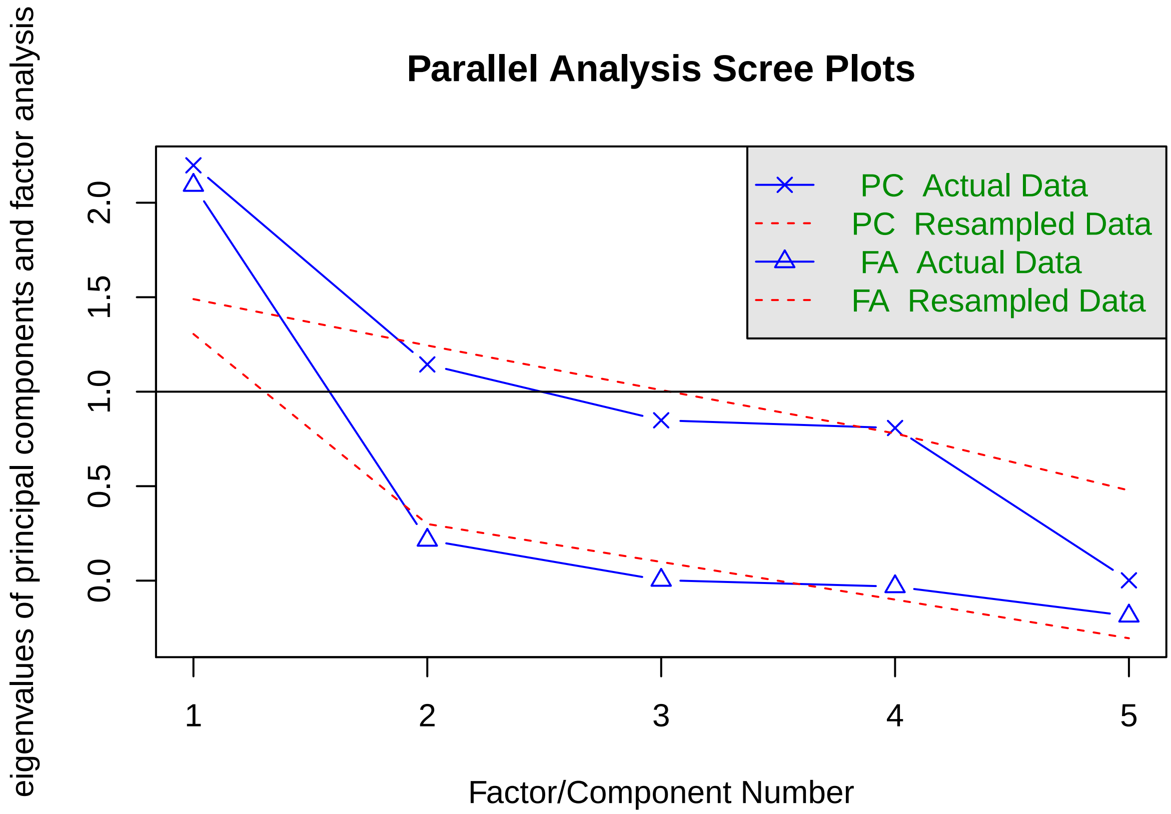
Method 3: Scree plot

The number of points after point of inflexion. For this plot, retaining PC1 and PC2 is recomended.

```
foodstuff.pca <- prcomp(foodstuff, center = TRUE, scale = TRUE)
fviz_eig(foodstuff.pca, addlabels = TRUE, ylim = c(0, 80))
```



```
parallel <- fa.parallel(foodstuff, fm = 'pa', fa = 'both', sim=FALSE)
```



```
## Parallel analysis suggests that the number of factors =  1  and the number of components =  1
```

Method 1, 2,and 3 recomend that only PC1 and PC2 are retained. I would only include PC3 if the minimum explained variance % is selected to be at least 80%. Selecting PC4 should be avoided as it explain 99.97% and this could indicate possible overfitting. None of the methods recomended PC3-PC5 to be retained.

(b) Obtain principal component loadings.

```
foodstuff.pc$loadings
```

```
##
## Loadings:
##          PC1    PC2    PC3    PC4    PC5
## Energy    0.969         0.137  0.178
## Protein   0.224 -0.739 -0.426  0.471
## Fat       0.948  0.216  0.199  0.120
## Calcium  -0.526         0.601  0.602
## Iron     -0.181  0.737 -0.497  0.420
##
##                  PC1   PC2   PC3   PC4   PC5
## SS loadings    2.198 1.144 0.849 0.808 0.002
## Proportion Var 0.440 0.229 0.170 0.162 0.000
## Cumulative Var 0.440 0.668 0.838 1.000 1.000
```

```
#Use R
foodstuff.scaled <- scale(foodstuff, center = TRUE, scale = TRUE)

#Correlation matrix
foodstuff_cor <- cor(foodstuff.scaled)

res.eig <- eigen(foodstuff_cor)
res.eig
```

```
## eigen() decomposition
## $values
## [1] 2.197777619 1.144204758 0.848574671 0.807842783 0.001600169
##
## $vectors
##            [,1]        [,2]       [,3]       [,4]         [,5]
## [1,] -0.6539155  0.08725829 -0.1490040 -0.1985936  0.709322816
## [2,] -0.1511882 -0.69052953  0.4629211 -0.5245825 -0.104059181
## [3,] -0.6394332  0.20196122 -0.2157528 -0.1336768 -0.697078234
## [4,]  0.3546581 -0.00633049 -0.6521357 -0.6699900  0.003161132
## [5,]  0.1219811  0.68900403  0.5400663 -0.4675657  0.010235855
```

The sum of the eigenvalues from a correlation matrix will equal the number of variables. The sum of the eigenvalues will equal 5.

The SS loadings = 2.1977776, which is the eigenvalue for the first principal component. The proportion variance = 44%.

The SS loadings = 1.1442048, which is the eigenvalue for the second principal component. The proportion variance = 23%.

The SS loadings = 0.8485747, which is the eigenvalue for the third principal component. The proportion variance = 17%.

The SS loadings = 0.8078428, which is the eigenvalue for the fourth principal component. The proportion variance = 16%.

The SS loadings = 0.0016002, which is the eigenvalue for the fifth principal component. The proportion variance = 0%.

The first principal component accounts for the most variable variance (2.1977776 / 5 = 44%) with the remaining components in lesser and lesser amounts. This leaves 56% unexplained variance. This could be due to another principal component or residual error variance.

The first and second principal component accounts for the most variable variance (3.3419824 / 5 = 66.8%) with the remaining components in lesser and lesser amounts. This leaves 33.2% unexplained variance. This could be due to another principal component or residual error variance.

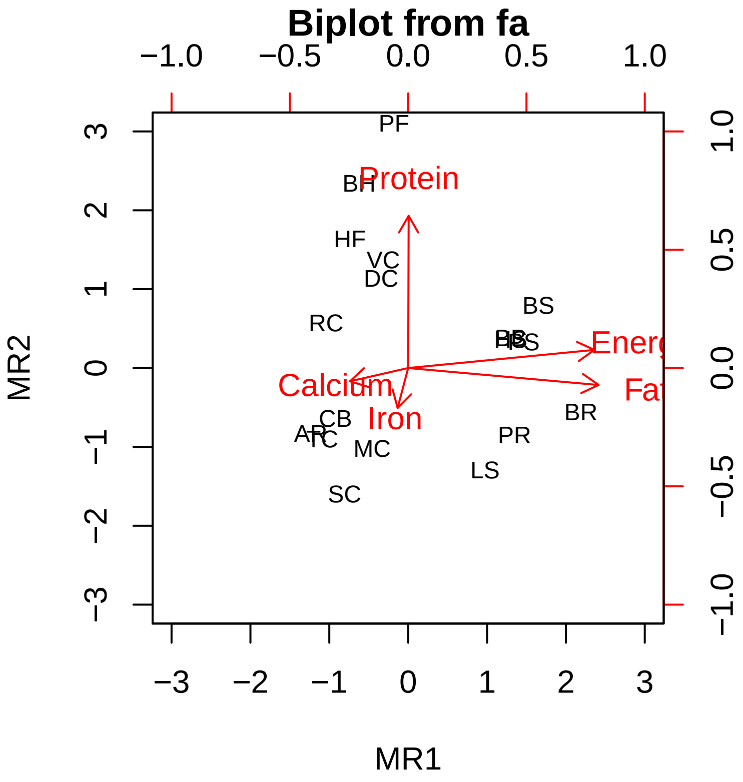
Positively correlated variables are grouped together: fat and energy.

Negatively correlated variables are positioned on opposite sides of the plot origin (opposed quadrants): iron and protein.

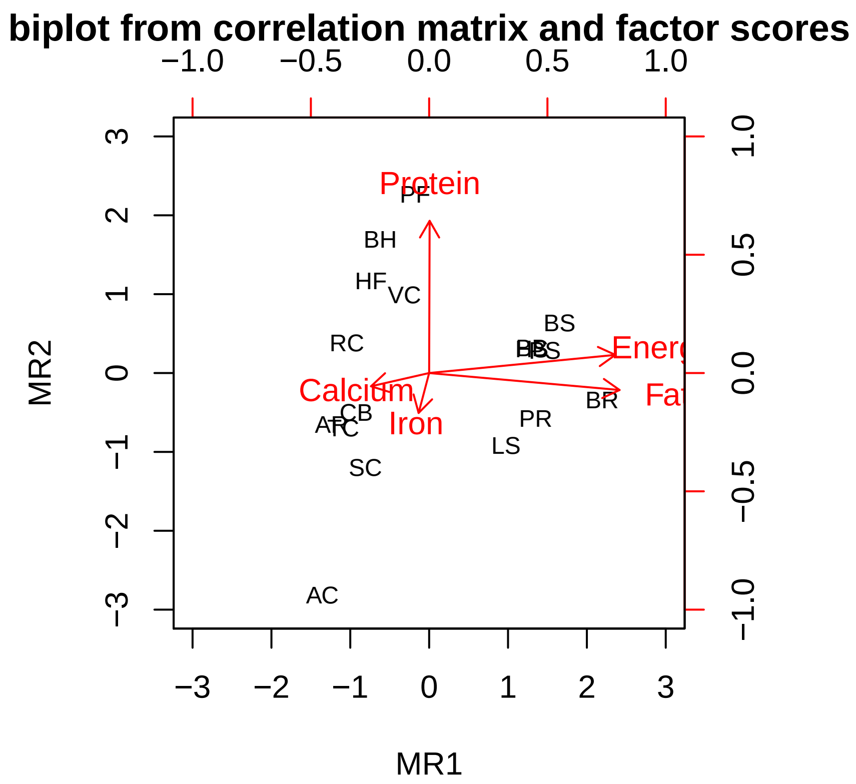
The distance between variables and the origine measures the quality of the variables on the factor map. Variables that are away from the origin are well represented on the factor map.

(c) Calculate percent of variance explained for each factor, plot the factor scores using appropriate plot(s), and decide how many components to retain.

```
fa2 <- fa(foodstuff, 2, scores=TRUE, covar = FALSE)
biplot(fa2, labels=rownames(foodstuff))
```



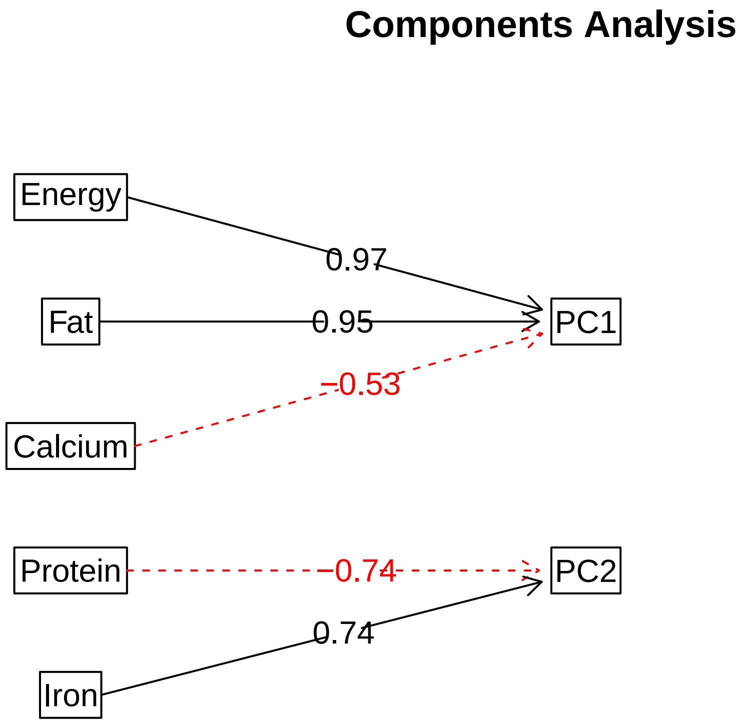
```
x <- list()
x$scores <- factor.scores(foodstuff, fa2)
x$loadings <- fa2$loadings
class(x) <- c('psych','fa')
biplot(x, main="biplot from correlation matrix and factor scores",
       labels = rownames(foodstuff))
```



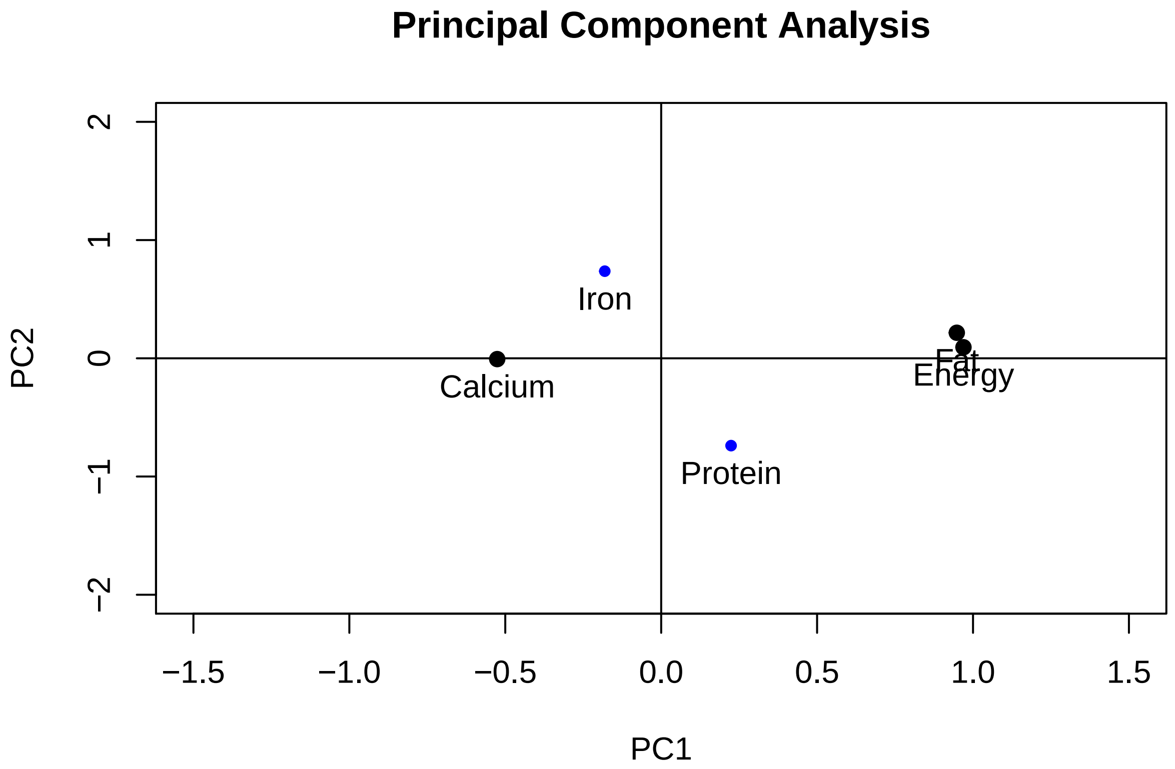
Method 1, 2,and 3 recomend that only PC1 and PC2 are retained.

```
foodstuff.pc <- principal(foodstuff, nfactors = 2, rotate = 'none', covar = FALSE)
foodstuff.pc
```

```
## Principal Components Analysis
## Call: principal(r = foodstuff, nfactors = 2, rotate = "none", covar = FALSE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##           PC1   PC2   h2    u2 com
## Energy   0.97  0.09 0.95 0.052 1.0
## Protein  0.22 -0.74 0.60 0.404 1.2
## Fat      0.95  0.22 0.95 0.055 1.1
## Calcium -0.53 -0.01 0.28 0.724 1.0
## Iron    -0.18  0.74 0.58 0.424 1.1
##
##                        PC1  PC2
## SS loadings           2.20 1.14
## Proportion Var        0.44 0.23
## Cumulative Var        0.44 0.67
## Proportion Explained  0.66 0.34
## Cumulative Proportion 0.66 1.00
##
## Mean item complexity =  1.1
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.16
##  with the empirical chi square  13.51  with prob <  0.00024
##
## Fit based upon off diagonal values = 0.8
```



```
plot(foodstuff.pc, labels=names(foodstuff),  ylim = c(-2,2),  xlim = c(-1.5,1.5))
```



Problem 4

The data below measures in five variables in comparison of normal patients and diabetics:

x1 : glucose intolerance

x2 : insulin response to oral glucose

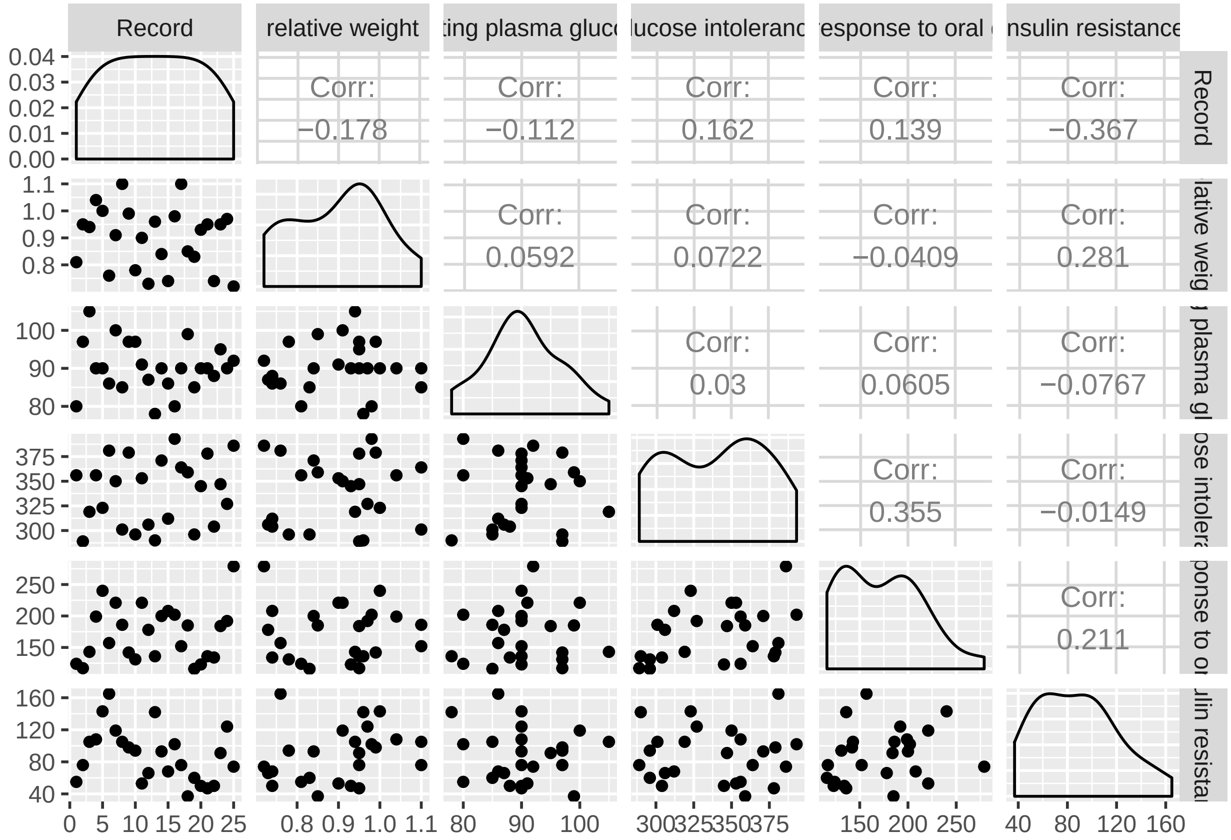
x3 : insulin resistance

y1 : relative weight

y2 : fasting plasma glucose

Record	relative weight	fasting plasma glucose	glucose intolerance	insulin response to oral glucose	insulin resistance
1	0.81	80	356	124	55
2	0.95	97	289	117	76
3	0.94	105	319	143	105
4	1.04	90	356	199	108
5	1.00	90	323	240	143
6	0.76	86	381	157	165
7	0.91	100	350	221	119
8	1.10	85	301	186	105
9	0.99	97	379	142	98
10	0.78	97	296	131	94
11	0.90	91	353	221	53
12	0.73	87	306	178	66
13	0.96	78	290	136	142
14	0.84	90	371	200	93
15	0.74	86	312	208	68
16	0.98	80	393	202	102
17	1.10	90	364	152	76
18	0.85	99	359	185	37
19	0.83	85	296	116	60
20	0.93	90	345	123	50
21	0.95	90	378	136	47
22	0.74	88	304	134	50
23	0.95	95	347	184	91
24	0.97	90	327	192	124
25	0.72	92	386	279	74

```r
ggpairs(diabetes, progress = FALSE)
```



(a) Find the canonical correlation between (y1, y2) and (x1, x2, x3).

Correlations

```
matcor(y, x)
```

```
## $Xcor
##                          relative weight fasting plasma glucose
## relative weight               1.00000000             0.05920816
## fasting plasma glucose        0.05920816             1.00000000
##
## $Ycor
##                                    glucose intolerance
## glucose intolerance                         1.00000000
## insulin response to oral glucose            0.35547847
## insulin resistance                         -0.01492231
##                                    insulin response to oral glucose
## glucose intolerance                                       0.3554785
## insulin response to oral glucose                          1.0000000
## insulin resistance                                        0.2108386
##                                    insulin resistance
## glucose intolerance                       -0.01492231
## insulin response to oral glucose           0.21083862
## insulin resistance                         1.00000000
##
```

```
## $XYcor
##                                  relative weight fasting plasma glucose
## relative weight                       1.00000000             0.05920816
## fasting plasma glucose                0.05920816             1.00000000
## glucose intolerance                   0.07217594             0.02997456
## insulin response to oral glucose     -0.04093194             0.06045515
## insulin resistance                    0.28061176            -0.07672352
##                                  glucose intolerance
## relative weight                           0.07217594
## fasting plasma glucose                    0.02997456
## glucose intolerance                       1.00000000
## insulin response to oral glucose          0.35547847
## insulin resistance                       -0.01492231
##                                  insulin response to oral glucose
## relative weight                                       -0.04093194
## fasting plasma glucose                                 0.06045515
## glucose intolerance                                    0.35547847
## insulin response to oral glucose                       1.00000000
## insulin resistance                                     0.21083862
##                                  insulin resistance
## relative weight                          0.28061176
## fasting plasma glucose                  -0.07672352
## glucose intolerance                     -0.01492231
## insulin response to oral glucose         0.21083862
## insulin resistance                       1.00000000
```

The canonical correlations:

```r
# computer canonical correlations
cc1 <- cc(y, x)
# display the canonical correlations
cc1$cor
```

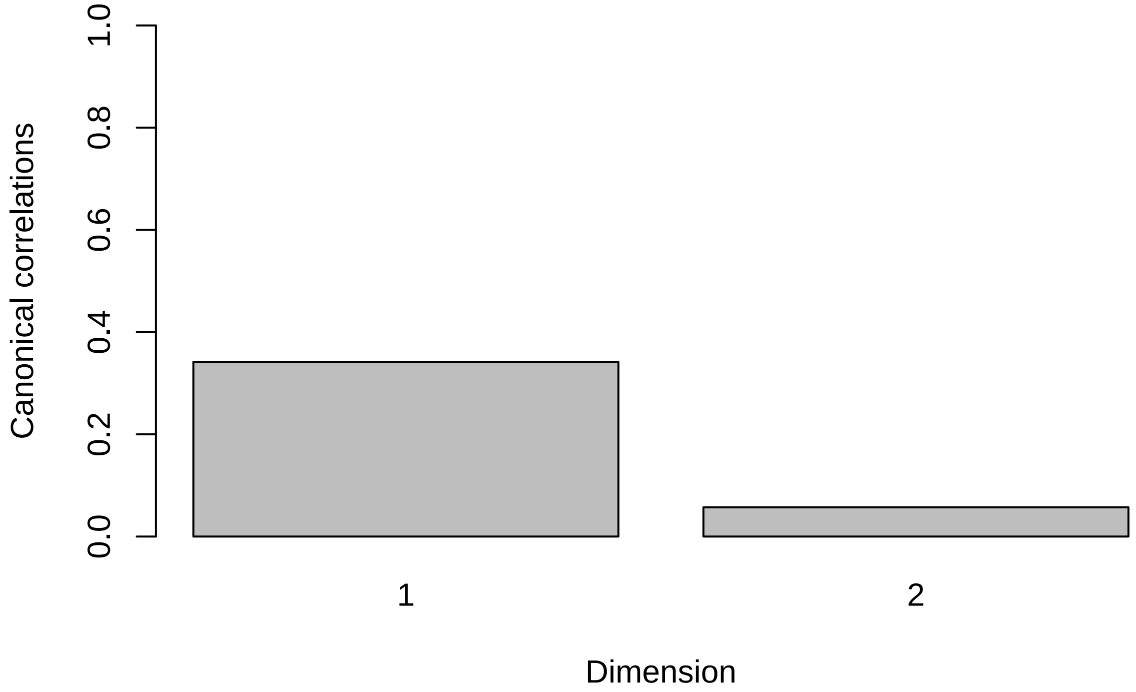
```
## [1] 0.34192472 0.05719007
```

$r_1 = 0.3419247$

$r_2 = 0.0571901$

The first canonical variate captures the most explained variance, canonical r = 0.3419247.

The scree graph of canonical correlations:



Raw canonical coefficients

```
# raw canonical coefficients
cc1[3:4]
```

```
## $xcoef
##                              [,1]      [,2]
## relative weight        -8.49151099 -2.454277
## fasting plasma glucose  0.05134587 -0.144875
##
## $ycoef
##                                        [,1]          [,2]
## glucose intolerance              -0.01106031 -0.0191559262
## insulin response to oral glucose  0.01190259 -0.0133269209
## insulin resistance               -0.02911410  0.0003958763
```

Compute canonical loadings

```
# compute canonical loadings
cc2 <- comput(y, x, cc1)
# display canonical loadings
cc2[3:6]
```

```
## $corr.X.xscores
##                             [,1]       [,2]
## relative weight        -0.9425535 -0.3340551
## fasting plasma glucose  0.2776622 -0.9606788
##
## $corr.Y.xscores
##                                        [,1]         [,2]
## glucose intolerance               -0.0594290 -0.048378036
## insulin response to oral glucose   0.0596224 -0.045697135
## insulin resistance                -0.2957265 -0.005609106
##
## $corr.X.yscores
##                               [,1]        [,2]
## relative weight        -0.32228235 -0.01910464
## fasting plasma glucose  0.09493958 -0.05494129
##
## $corr.Y.yscores
##                                        [,1]        [,2]
## glucose intolerance               -0.1738072 -0.84591670
## insulin response to oral glucose   0.1743729 -0.79903966
## insulin resistance                -0.8648876 -0.09807832
```

Table 1: Canonical Coefficients

	Dimension 1	Dimension 2
relative weight	-8.4915110	-2.4542771
fasting plasma glucose	0.0513459	-0.1448750
glucose intolerance	-0.0110603	-0.0191559
insulin response to oral glucose	0.0119026	-0.0133269
insulin resistance	-0.0291141	0.0003959

Table 1 presents the canonical coefficients for the first two dimensions across both sets of variables.

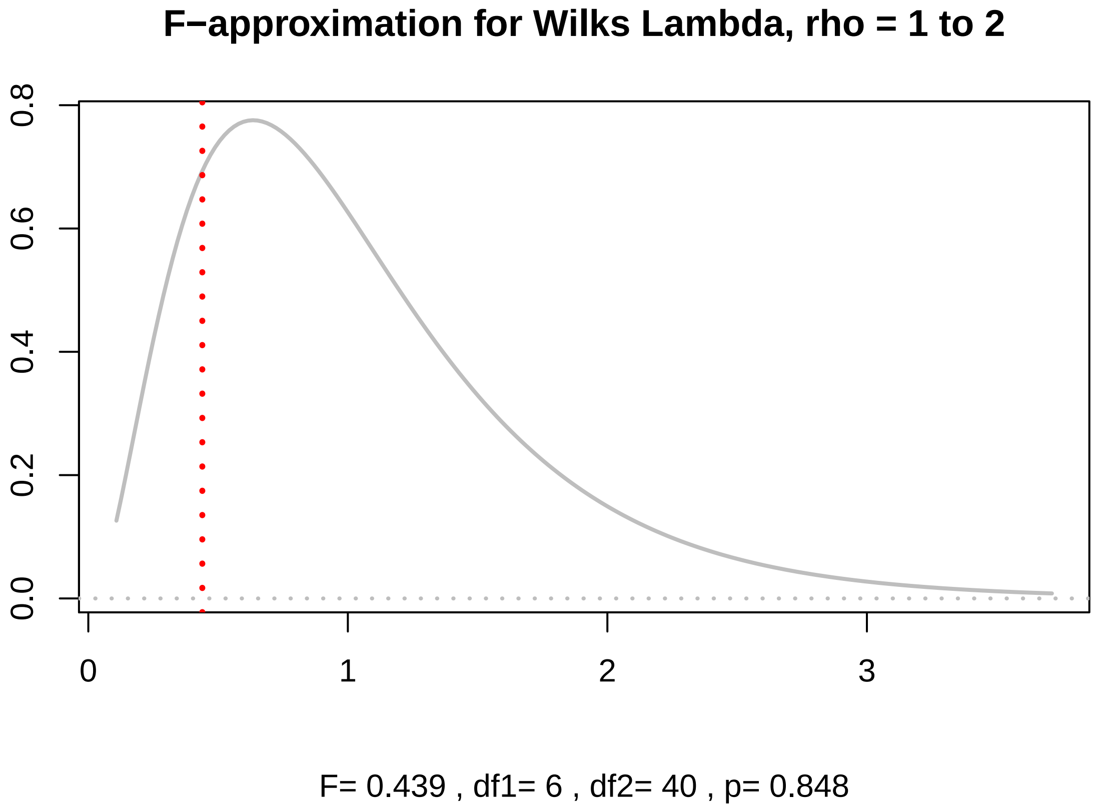
(b) Test the significance of each canonical correlation.

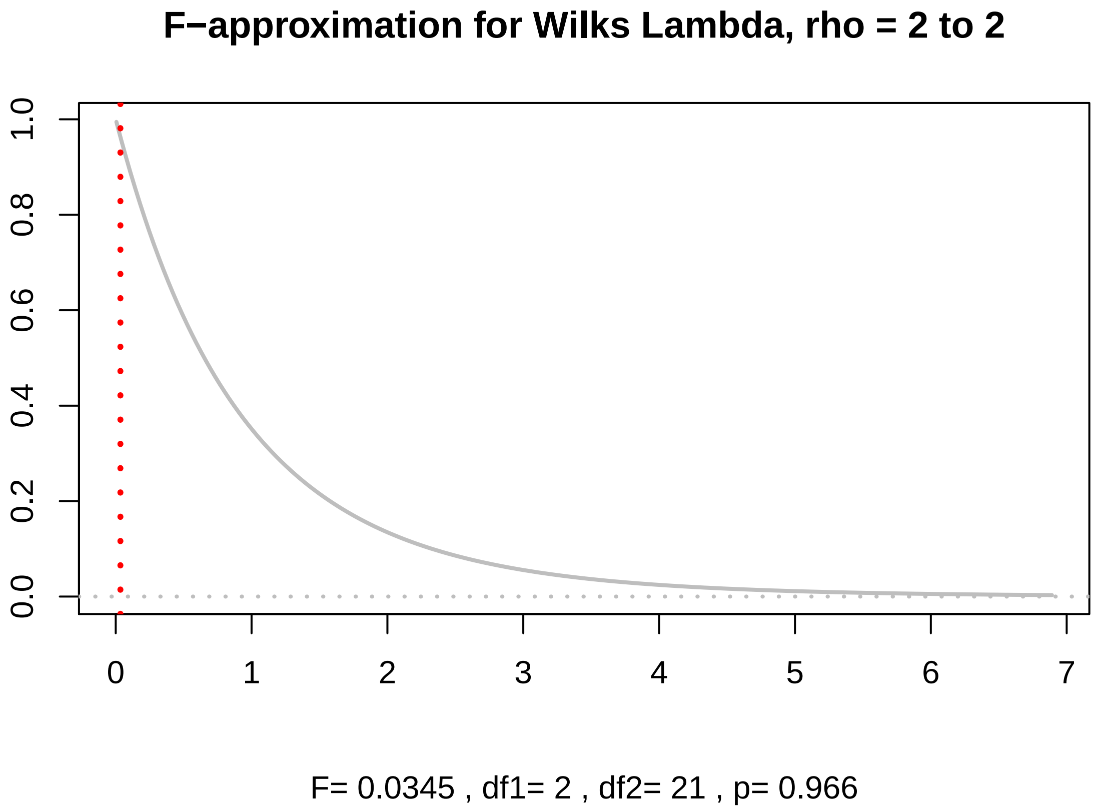
Table 2: Tests of Canonical Dimensions

```r
# tests of canonical dimensions
rho <- cc1$cor
## Define number of observations,
n <- dim(y)[1]
# number of variables in first set
p <- length(y)
# number of variables in the second set
q <- length(x)

## Calculate p-values using the F-approximations of different test statistics:
res <- p.asym(rho, n, p, q, tstat = "Wilks")
```

```
## Wilks' Lambda, using F-approximation (Rao's F):
##               stat     approx df1 df2   p.value
## 1 to 2:  0.8801992 0.43921980   6  40 0.8481513
## 2 to 2:  0.9967293 0.03445509   2  21 0.9661862
```

Plot the F-approximation for Wilks' Lambda, considering 2 or 1 canonical correlation(s):





H_0: there is no (linear) relationship between the y's and the x's - all canonical correlations r_1, r_2 are non-significant.

H_1: there is (linear) relationship between the y's and the x's at least one canonical correlations r_1, r_2 is significant.

We fail to reject the null hypothesis in favor of the alternative. This implies that r_1^2 and r_2^2 is not significantly different from zero.

We conclude that $r_1 = 0.3419247$ (F =0.4392198) is not significant since the p-value (0.8481513) $>$ 0.05

We conclude that $r_2 = 0.0571901$ (F =0.0344551) is not significant since the p-value (0.9661862) $>$ 0.05

Tests of dimensionality for the canonical correlation analysis, as shown in Table 2, indicate that the two canonical dimensions are not statistically significant at the 0.05 level. Dimension 1 has a canonical correlation of 0.3419247 between the sets of variables, while for dimension 2 the canonical correlation was much lower at 0.0571901

Problem 5

The data consists of mental ability test scores of seventh- and eighth-grade children from two different schools (Pasteur and Grant-White). In our version of the dataset, only 9 out of the original 26 tests are included. A CFA model that is often proposed for these 9 variables consists of three latent variables (or factors), each with three indicators:

- a visual factor measured by 3 variables: x1 and x2
- a textual factor measured by 4 variables: x3, x4, x5 and x6

- a speed factor measured by 3 variables: x7, x8 and x9
- a visual factor and a textual factor have zero correlation

(a) Please draw a figure contains a graphical representation of the three-factor model.

x1: Visual perception

x2: Cubes

x3: Lozenges

x4: paragraph comprehension

x5: Sentence completion

x6: Word meaning

x7: Speeded addition

x8: Speeded counting of dots

x9: Speeded discrimination straight and curved capitals

```
# specify the model
HS.model <- ' Visual  =~ x1 + x2
              Textual =~ x3 + x4 + x5 + x6
              Speed   =~ x7 + x8 + x9
              Visual ~~ 0 * Textual'

fit <- lavaan::cfa(HS.model, data=HolzingerSwineford1939)
summary(fit)
```

```
## lavaan 0.6-3 ended normally after 53 iterations
##
##   Optimization method                           NLMINB
##   Number of free parameters                         20
##
##   Number of observations                           301
##
##   Estimator                                         ML
##   Model Fit Test Statistic                     200.854
##   Degrees of freedom                                25
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
## Latent Variables:
##                     Estimate  Std.Err  z-value  P(>|z|)
##   Visual =~
##     x1                 1.000
##     x2                 0.341    0.222    1.537    0.124
##   Textual =~
```

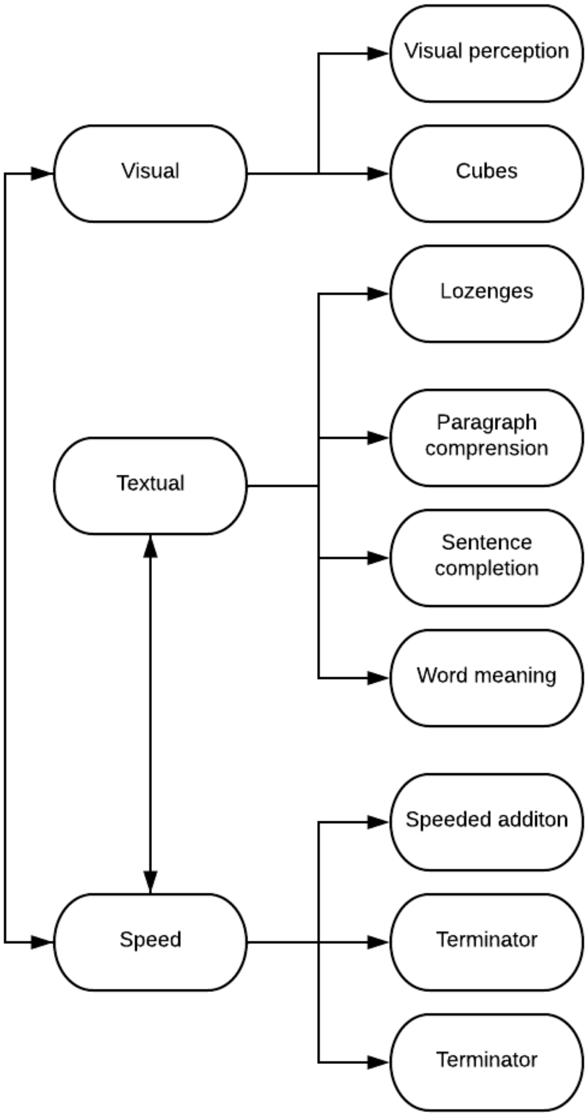
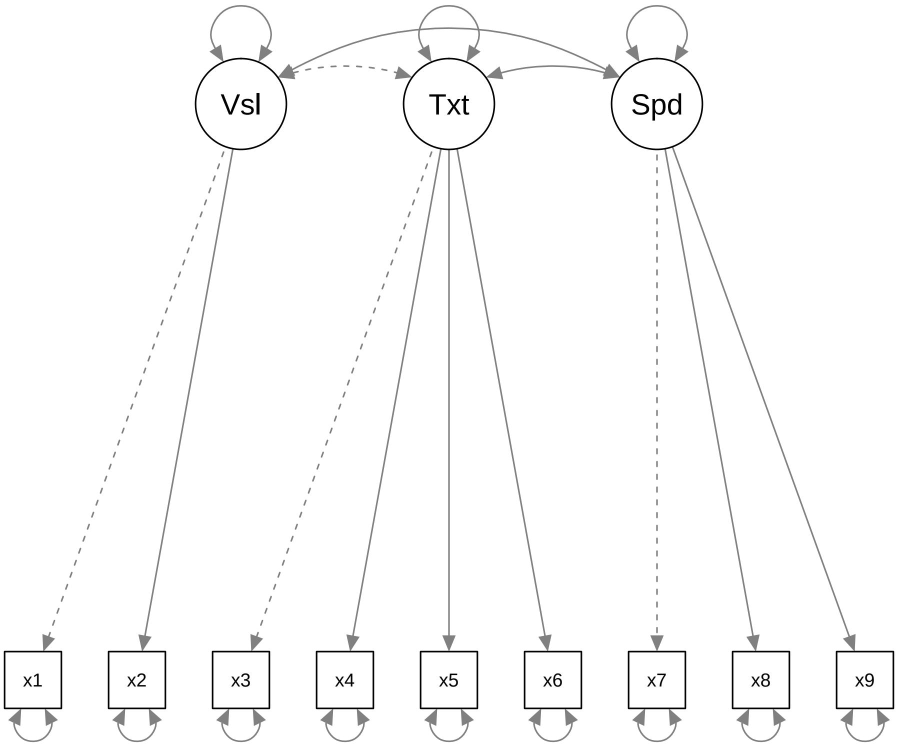


Figure 1: Three-factor model

```
##     x3                1.000
##     x4                5.193    1.878    2.765    0.006
##     x5                5.846    2.113    2.766    0.006
##     x6                4.816    1.743    2.763    0.006
##   Speed =~
##     x7                1.000
##     x8                1.176    0.170    6.915    0.000
##     x9                1.012    0.145    7.001    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   Visual ~~
##     Textual           0.000
##     Speed             0.216    0.056    3.886    0.000
##   Textual ~~
##     Speed             0.019    0.011    1.745    0.081
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                0.163    0.754    0.216    0.829
##    .x2                1.243    0.134    9.284    0.000
##    .x3                1.239    0.101   12.222    0.000
##    .x4                0.380    0.049    7.811    0.000
##    .x5                0.429    0.059    7.278    0.000
##    .x6                0.361    0.044    8.270    0.000
##    .x7                0.777    0.082    9.502    0.000
##    .x8                0.461    0.078    5.926    0.000
##    .x9                0.599    0.071    8.429    0.000
##     Visual            1.195    0.761    1.570    0.116
##     Textual           0.036    0.026    1.383    0.167
##     Speed             0.392    0.088    4.454    0.000
```

```
semPaths(fit) #, title = FALSE, curvePivot = TRUE)
```



(b) Please write out the corresponding syntax for specifying this model.

Visual =~ x1 + x2

Textual =~ x3 + x4 + x5 + x6

Speed =~ x7 + x8 + x9

Visual ~~ 0 * Textual

Problem 6

Make a conclusion for this class DA 410, make sure you include the following aspects:

(a) How many models you have learnt, use 3 to 5 sentences to explain each of them.

Multivariate Analysis of Variance

MANOVA was used to tests for the difference in two or more vectors of means. For this model, total of the ith sample, overall total, and mean of the ith sample is calculated. These calculations are then used to create a model for each observation vector. There are four test statistic used to test: Wilks, Roy, Pillai and Lawley-Hotelling. MANOVA can be conducted as one-way or as two-way. The number of factor variables involved distinguish a one-way MANOVA from a two-way MANOVA.

Discriminant Analysis

Discriminant Analysis is used to perform a multivariate test of differences between groups and can also be used to determine the minimum number of dimensions needed to describe the differences. LDA: The mean and the variance of the variable for each class is calculated for every single input variable (x). For multiple variables, the mean and the covariance matrix is calculated over the multivariate Gaussian. These statistical properties are estimated are used in the LDA equation to make predictions. QDA is an extension of LDA.

K-Nearest neighbor

For K-Nearest neighbor each new instance (x) is predicted by searching through the entire training set for the K most similar instances (neighbors) and summarizing the output variable for those K instances. To predict iterate from 1 to total number of training data points and calculate the distance (such as Euclidean) between them. Then sort the calculated distances in ascending order based on distance values, get top k rows from the sorted array, determine the most frequent class of the rows and then return the predicted class.

Canonical correlation analysis

Canonical correlation analysis is a method for exploring the relationships between two multivariate sets of variables (vectors). A correlation matrix is computed and the following information is extracted:

- Rxx: The correlations among the X variables.
- Ryy: The correlations among the Y variables.
- Rxy: The correlations between the X and Y variables.
- Ryx: The correlations between the Y and X variables.

The Canonical correlation coefficients are defined using the singular value decomposition of a matrix C.

Principal Component Analysis

Principal Component Analysis is a data reduction methods that identify the variance in variables in a smaller set. Find the covariance matrix S (covariance matrix) or R (correlation matrix) to compute eigenvalues and eigenvectors. Determine the total variance for each principal components. Determine components to keep based on total variance explained, keep only the components whose eigenvalues is at least greater than the average eigenvalue and the average sample variance of the original variables.

Factor Analysis

Factor Analysis is a data reduction methods that identify the variance in variables in a smaller set.

Exploratory Factor Analysis main goals are to explore the underlying theoretical structure and dentify the structure of the relationship between the variable and the respondent. The are two methods: Principle component actor analysis and Common factor analysis. The five steps are:

1- Find correlation matrix

2- Find eigenvalues and eigenvectors.

3 - Find C_1 and D_1

4 - Find $C_1 D_1^{1/2}$

5- Find component loading.

Confirmatory Factor Analysis is used to verify the factor structure of a set of observed variables (i.e. test the model). The model defines the latent variable with set of observed (manifest/indicators) variables. CFA test whether the indicators that have been classified previously into a group will be consistent. The initial assumption is that indicators fit into a certain latent variables.

Cluster Analysis

K-means

1- Each data point is assigned to its nearest centroid, based on the squared Euclidean distance.

2- The centroids are recomputed.

3- Iterates between steps 1 and 2 until a stopping criteria is met (no data points change clusters, the sum of the distances is minimized, or some maximum number of iterations is reached).

(b) Which one really impressed you when you learnt and why

While I had been previously exposed to PCA as a data reduction methods, I was really impressed with Factor Analysis and Canonical Correlation Analysis. I was originally confused at first specially given that FA has different methods to extract factors: principal component analysis, common factor analysis (SEM), image factoring (OLS), maximum likelihood method and others. Factor analyis can be Exploratory factor analysis or Confirmatory factor analysis.

The assumption in exploratory factor analysis (EFA) is that any indicator or variable may be associated with any factor. Confirmatory factor analysis (CFA) is used to determine the factor and factor loading of measured variables. The analysis is based on a prior theory and one of the goals is to confirm it. CFA assumes that each factor is associated with a specified subset of measured variables.

(c) Which one is your favorite one and why

We did not spend too much time on K-means, but his is one of my favorite modeling techniques. This method is simple to even a novice can easily understand. The K-means clustering algorithm can be used to find groups which have not been explicitly labeled in the data. This is a versatile algorithm that can be used for in the following business cases:

- Inventory categorization
- Detecting bots or anomalies
- Behavioral segmentation
- Sorting sensor measurements f

(d) Select two models out, make a comparison. Show the differences and similarities between them.

I will discuss the fundamental difference between Principal Component Analysis (PCA) and Factor Analysis (FA). Both are data reduction methods that identify the variance in variables in a smaller set. The steps to conducted PCA or FA are similar: extraction, interpretation, rotation, choosing the number of components/factors.

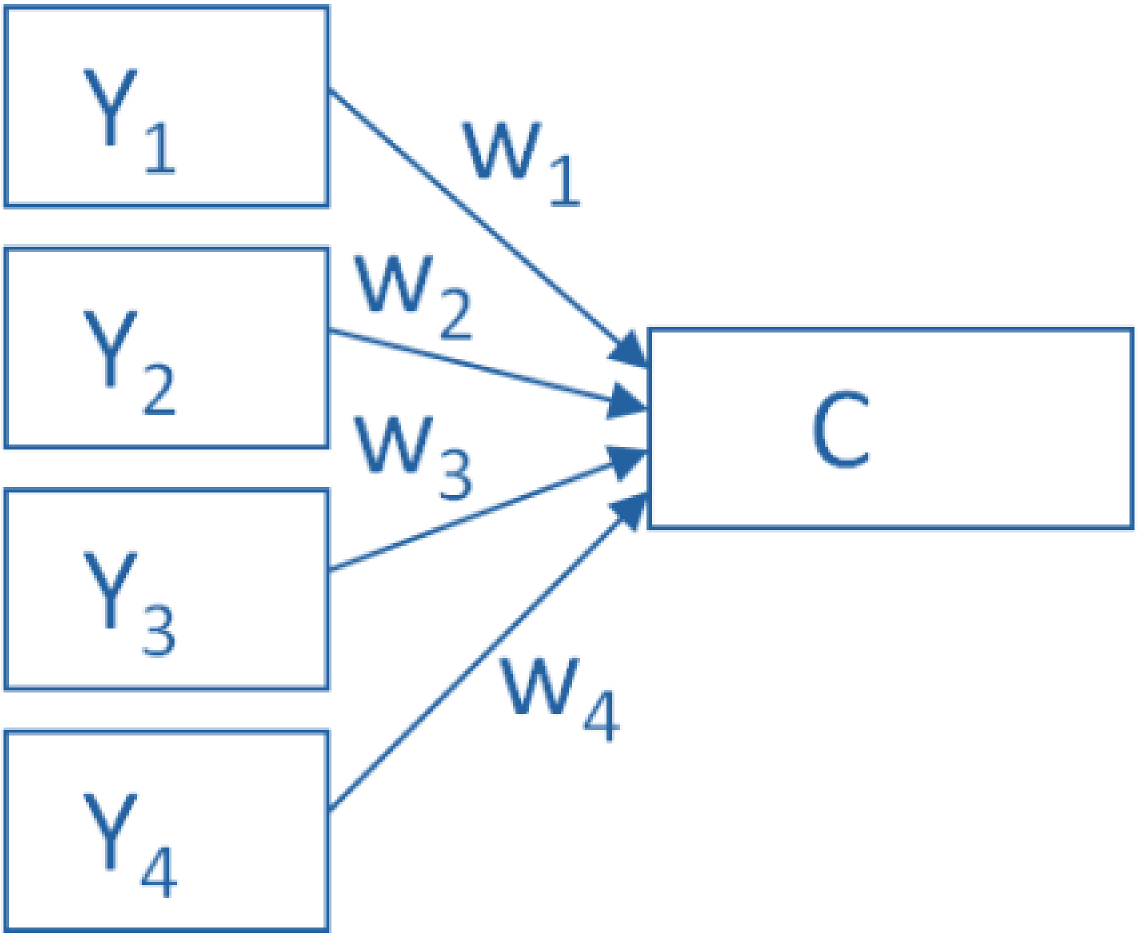


Figure 2: CPCA

PCA is a linear combination of variables. Factor Analysis is a measurement model of a latent variable. PCA creates one or more components variables from a larger set of measured variables by using a linear combination (weighted average) of a set of variables. This method seeks to find the optimal number of components and weights. The relationships between C and each Y are weighted.

In this example, 4 measured $(Y_1, Y_2, Y_3, and Y_4)$ variables are combined into a single component (C). The direction of the arrows shows that Y variables contribute to the component variable. The weights $(w1, w_2, w_3, and w_4)$ shows how Y variables contribute more than others.

$C = w_1(Y_1) + w_2(Y_2) + w_3(Y_3) + w_4(Y_4)$

Factor Analysis approaches data reduction by modeling the measurement of a latent variable. Latent variable cannot be directly measured with a single variable (such as social anxiety) but can be seen through the relationships it causes in a set of Y variables.

Measuring social anxiety is not possible but social anxiety can be measured as high or low by ansering question such as "I am uncomfortable in large groups" and "I get nervous talking with strangers." Those experiencing high social anxiety will most likely answer similarly to others with high social anxiety. Likewise, those experiencing low social anxiety will most likely answer similarly to others with low social anxiety.

In this example F, the latent factor, is causing the responses on the four measured Y variables. The arrows go in the opposite direction from PCA. The relationships between F and each Y are weighted. During factor

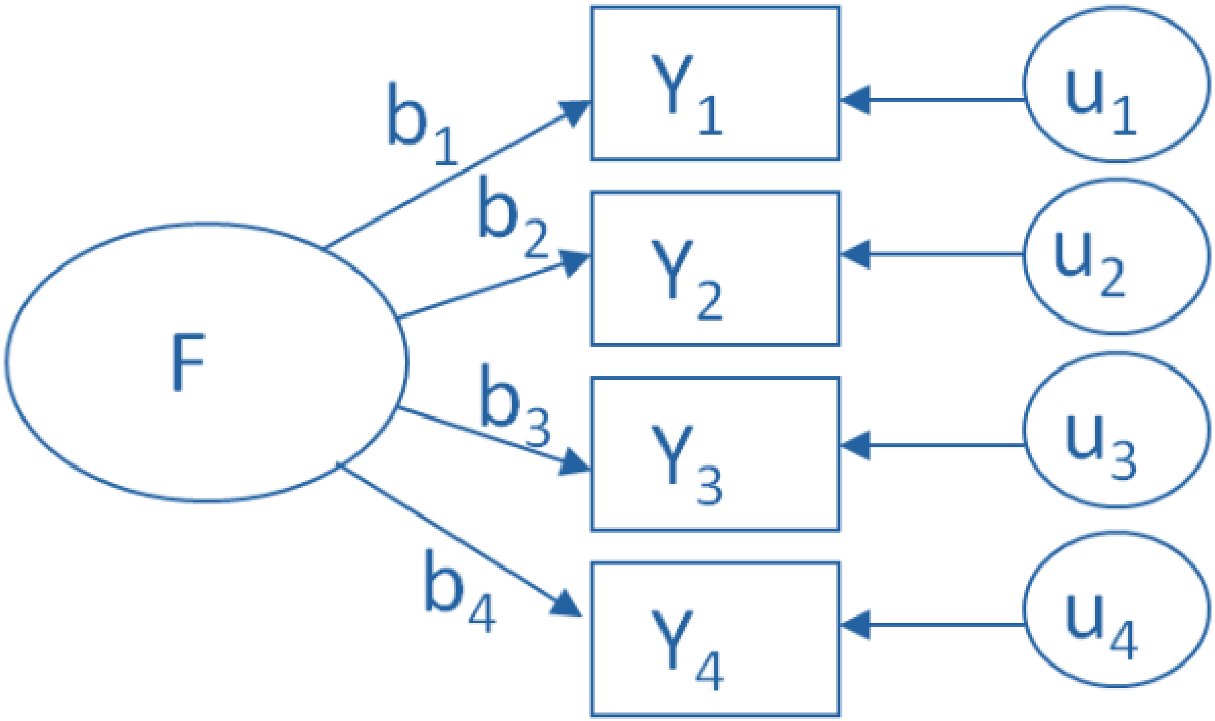


Figure 3: FA

analysis the optimal weights are identified. The u's are the set of error terms which measures variance in each Y that is unexplained by the factor.

$Y_1 = b_1 * F + u_1$

$Y_2 = b_2 * F + u_2$

$Y_3 = b_3 * F + u_3$

$Y_4 = b_4 * F + u_4$

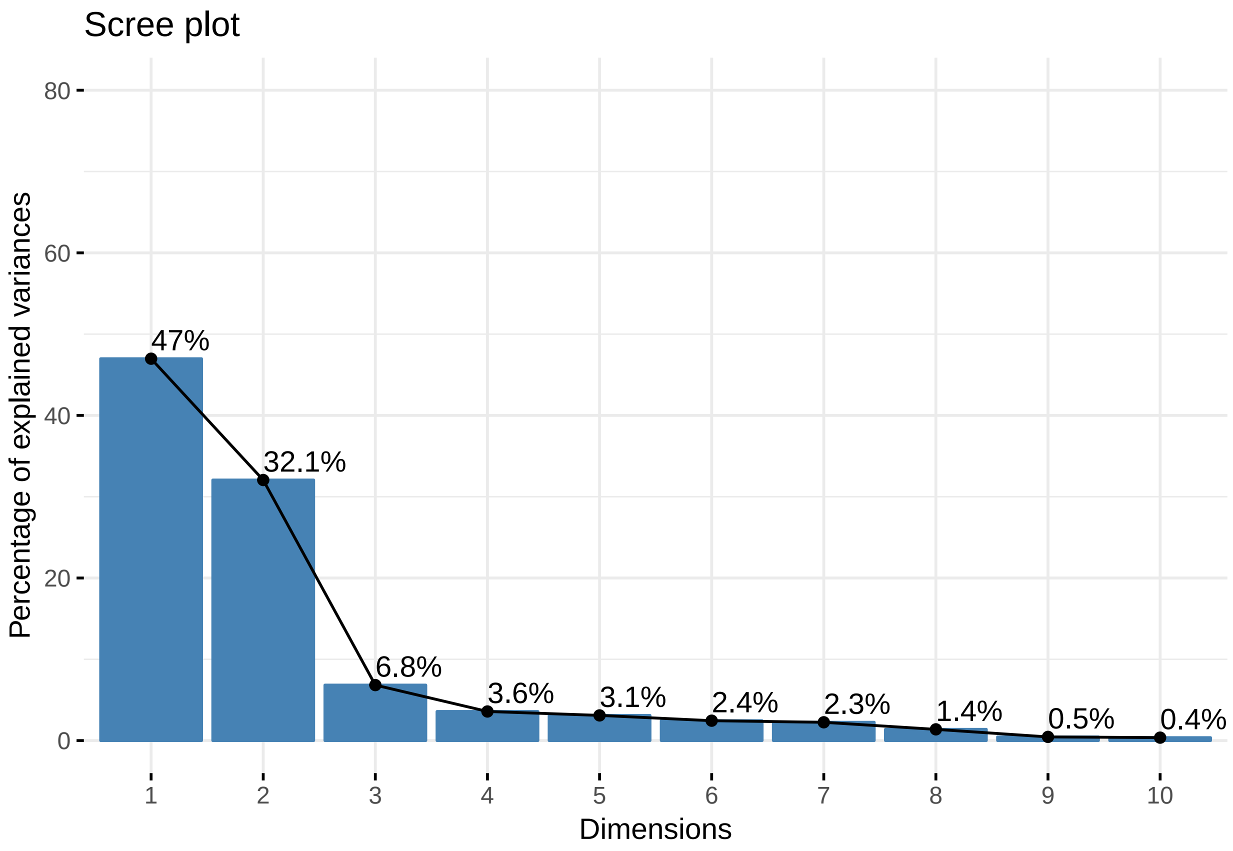
(e) If you will build up a project to solve some real problem using one of them, which one you would like to you, and what kind of project you will like to build. (200 words)

Currently I am analazying the Boston HMDA Data to create a mortgage lending decisson model. The Boston HMDA data set was collected by researchers at the Federal Reserve Bank of Boston. The data set combines information from mortgage applications and a follow-up survey of the banks/lending institutions that received these mortgage applications. The data constains information on mortgage applications made in 1990 in the greater Boston metropolitan area. The full data set has 2925 observations, consisting of all mortgage applications by blacks and Hispanics and a random sample of mortgage applications by whites. My goal is to create both logit and probit models to determine if lending institution showed bias toward minorities (Black and Hispanics). I performed EDA to select the variables to include in my model but I would like to go back and perform principal component analysis as a way to reduce dimensionality. This is a practical and real world example of a model that I could build in the near future.

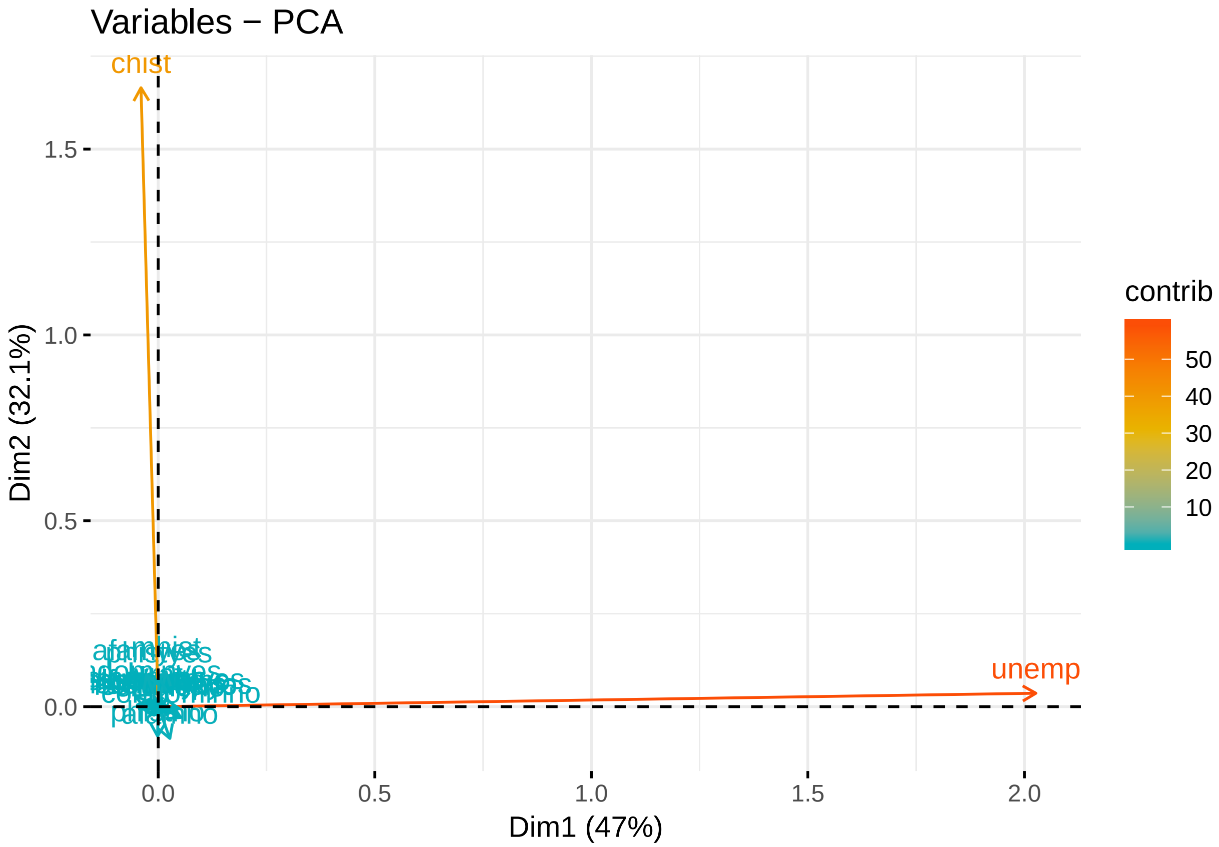
```r
#using HMDA from AER package
data(HMDA)
#create a dummy data frame
HMDA <- dummy.data.frame(HMDA, names = c("phist",
                                         "selfemp", "insurance",
                                         "condomin","afam", "single", "hschool"))

HMDA$chist <- as.integer(HMDA$chist)
HMDA$mhist <- as.integer(HMDA$mhist)

#principal component analysis
prin_comp <- prcomp(HMDA[,-1])
prin_comp
```

```
## Standard deviations (1, .., p=20):
##  [1] 2.030228e+00 1.677064e+00 7.741428e-01 5.606719e-01 5.216303e-01
##  [6] 4.636241e-01 4.446554e-01 3.483996e-01 1.999904e-01 1.789941e-01
## [11] 1.705788e-01 1.307716e-01 4.662084e-02 1.053630e-15 3.954764e-16
## [16] 3.002762e-16 1.942465e-16 1.723408e-16 1.183727e-16 9.526222e-17
##
## Rotation (n x k) = (20 x 20):
##                        PC1           PC2           PC3          PC4
## pirat         0.0024018888  0.0044893195 -0.0041840614  0.005823733
## hirat         0.0017655618  0.0005867931 -0.0055180552  0.016681209
## lvrat        -0.0033636841  0.0128023385 -0.0215854398  0.001634141
## chist        -0.0196088959  0.9928520345  0.0478542683  0.016745430
## mhist         0.0089166720  0.0560359401 -0.2141556801  0.234558324
## phistno      -0.0007667758 -0.0472703010  0.0051685694  0.023080767
## phistyes      0.0007667758  0.0472703010 -0.0051685694 -0.023080767
## unemp         0.9981308238  0.0215165364 -0.0372120980 -0.021821894
## selfempno    -0.0236820842  0.0029698170 -0.0198498420  0.011629887
## selfempyes    0.0236820842 -0.0029698170  0.0198498420 -0.011629887
## insuranceno  -0.0008156054 -0.0046591370  0.0045641976 -0.003925678
## insuranceyes  0.0008156054  0.0046591370 -0.0045641976  0.003925678
## condominno    0.0257608312 -0.0173370987  0.4119520199  0.542099847
## condominyes  -0.0257608312  0.0173370987 -0.4119520199 -0.542099847
## afamno        0.0132575886 -0.0510382934  0.1141321360  0.138793415
## afamyes      -0.0132575886  0.0510382934 -0.1141321360 -0.138793415
## singleno      0.0136618402 -0.0072477140  0.5401871301 -0.397348132
## singleyes    -0.0136618402  0.0072477140 -0.5401871301  0.397348132
## hschoolno     0.0060989178  0.0023885632  0.0001857499 -0.015833149
## hschoolyes   -0.0060989178 -0.0023885632 -0.0001857499  0.015833149
##                       PC5          PC6           PC7          PC8
## pirat        -0.009299362  0.015944382 -0.0145869382  0.024507008
## hirat        -0.017823420  0.009647270 -0.0040982128  0.016215845
## lvrat        -0.071034612  0.030966219 -0.0064420765  0.032109738
## chist         0.071391045 -0.052344740  0.0230472316 -0.053899950
## mhist        -0.872269324 -0.316801168 -0.1740372896 -0.007302052
## phistno       0.026944779 -0.103822343  0.0775061937 -0.690496117
## phistyes     -0.026944779  0.103822343 -0.0775061937  0.690496117
## unemp         0.003222351  0.014130388  0.0331214508 -0.001944552
## selfempno    -0.156195171  0.078715724  0.6817056388  0.057999504
## selfempyes    0.156195171 -0.078715724 -0.6817056388 -0.057999504
## insuranceno   0.017315372 -0.027450464  0.0047008522 -0.034103818
## insuranceyes -0.017315372  0.027450464 -0.0047008522  0.034103818
## condominno   -0.019108882  0.185204863 -0.0212131870 -0.009293529
## condominyes   0.019108882 -0.185204863  0.0212131870  0.009293529
## afamno        0.206964592 -0.625710322  0.1113197466  0.126821656
## afamyes      -0.206964592  0.625710322 -0.1113197466 -0.126821656
## singleno     -0.212794169 -0.064183986 -0.0176582965 -0.008988386
## singleyes     0.212794169  0.064183986  0.0176582965  0.008988386
## hschoolno    -0.019771739  0.005743303  0.0002902126 -0.002960013
## hschoolyes    0.019771739 -0.005743303 -0.0002902126  0.002960013
##                        PC9         PC10          PC11         PC12
## pirat        -4.405716e-02 -0.151879478 -0.1343871539  0.722389462
## hirat        -2.477236e-02 -0.139745242 -0.1066391688  0.633834602
## lvrat        -3.105053e-01 -0.217761891 -0.8930981987 -0.225234460
## chist         7.333806e-04  0.001183982  0.0027186292  0.001371278
## mhist         3.017814e-02  0.032952816  0.0475464792 -0.001617680
## phistno      -4.703019e-02 -0.011032156 -0.0170530122  0.016558130
## phistyes      4.703019e-02  0.011032156  0.0170530122 -0.016558130
## unemp         8.192336e-06  0.008824324 -0.0044011022 -0.001649279
## selfempno     8.072174e-03  0.006411031  0.0069724407  0.006163999
## selfempyes   -8.072174e-03 -0.006411031 -0.0069724407 -0.006163999
## insuranceno   6.649043e-01  0.006921680 -0.2356145862 -0.004224099
## insuranceyes -6.649043e-01 -0.006921680  0.2356145862  0.004224099
## condominno    8.145307e-03 -0.007608096  0.0001790002 -0.010886070
## condominyes  -8.145307e-03  0.007608096 -0.0001790002  0.010886070
## afamno       -3.294768e-02 -0.024077755 -0.0208520271  0.001885818
## afamyes       3.294768e-02  0.024077755  0.0208520271 -0.001885818
## singleno     -4.695978e-03  0.012534194 -0.0031353050  0.009141978
## singleyes     4.695978e-03 -0.012534194  0.0031353050 -0.009141978
## hschoolno     6.716797e-02 -0.673352656  0.1703815050 -0.110901053
## hschoolyes   -6.716797e-02  0.673352656 -0.1703815050  0.110901053
##                       PC13          PC14          PC15          PC16
## pirat         0.6586757398  3.568416e-16  1.833482e-16 -5.190040e-17
## hirat        -0.7521547157 -8.089401e-16 -2.561360e-16  4.131625e-16
## lvrat        -0.0094916067  2.099959e-16  4.774829e-16  2.499309e-17
## chist        -0.0023758745  1.508810e-17  2.500633e-17  3.926007e-18
## mhist         0.0090143833  6.664376e-17  1.785244e-16  1.387547e-17
## phistno       0.0031261030 -1.433194e-04  2.883770e-02  1.880945e-02
## phistyes     -0.0031261030 -1.433194e-04  2.883770e-02  1.880945e-02
## unemp        -0.0003743581 -5.281829e-17 -1.121133e-17  1.497320e-17
## selfempno     0.0053460624  1.700148e-03 -1.054719e-02  3.162855e-02
## selfempyes   -0.0053460624  1.700148e-03 -1.054719e-02  3.162855e-02
## insuranceno   0.0050112914  1.646742e-03 -1.090408e-02  6.162540e-03
## insuranceyes -0.0050112914  1.646742e-03 -1.090408e-02  6.162540e-03
## condominno    0.0037371522  7.069760e-01 -9.713327e-04  5.330822e-03
## condominyes  -0.0037371522  7.069760e-01 -9.713327e-04  5.330822e-03
## afamno        0.0015341145 -1.904146e-03 -7.030967e-01 -2.587447e-02
## afamyes      -0.0015341145 -1.904146e-03 -7.030967e-01 -2.587447e-02
## singleno     -0.0026529702  1.210942e-02 -6.235429e-02 -6.360053e-03
## singleyes     0.0026529702  1.210942e-02 -6.235429e-02 -6.360053e-03
## hschoolno     0.0054208272 -5.388674e-03 -2.653815e-02  7.055987e-01
## hschoolyes   -0.0054208272 -5.388674e-03 -2.653815e-02  7.055987e-01
##                       PC17          PC18          PC19          PC20
## pirat         5.147271e-16  8.475169e-16 -2.449437e-16  1.483173e-16
## hirat        -7.110921e-16 -1.214572e-15  4.255831e-17 -3.181410e-16
## lvrat        -2.237530e-17  3.578476e-16  2.919675e-16  3.462399e-17
## chist         4.245274e-18 -8.184020e-18 -6.680107e-18 -2.779504e-18
## mhist         1.540225e-17  5.258765e-17 -7.310921e-17  3.716168e-17
## phistno       6.358764e-02 -2.012870e-01  6.625108e-01  1.238311e-01
## phistyes      6.358764e-02 -2.012870e-01  6.625108e-01  1.238311e-01
## unemp        -2.227537e-17  4.329427e-18  1.264288e-17  3.697358e-18
## selfempno     1.441404e-01 -6.533677e-01 -1.897178e-01 -1.233973e-01
## selfempyes    1.441404e-01 -6.533677e-01 -1.897178e-01 -1.233973e-01
## insuranceno   6.576361e-03 -8.286988e-02 -1.535606e-01  6.850904e-01
## insuranceyes  6.576361e-03 -8.286988e-02 -1.535606e-01  6.850904e-01
## condominno    1.147967e-02  4.696649e-03  5.882567e-04 -1.172985e-03
## condominyes   1.147967e-02  4.696649e-03  5.882567e-04 -1.172985e-03
## afamno        6.165708e-02  1.534064e-02  3.061020e-02 -2.828412e-03
## afamyes       6.165708e-02  1.534064e-02  3.061020e-02 -2.828412e-03
## singleno     -6.863338e-01 -1.558178e-01  2.294083e-02 -8.081974e-03
## singleyes    -6.863338e-01 -1.558178e-01  2.294083e-02 -8.081974e-03
## hschoolno    -1.222579e-02  3.449942e-02 -6.490729e-03 -3.920851e-03
## hschoolyes   -1.222579e-02  3.449942e-02 -6.490729e-03 -3.920851e-03
```

```
fviz_eig(prin_comp, addlabels = TRUE, ylim = c(0, 80))
```



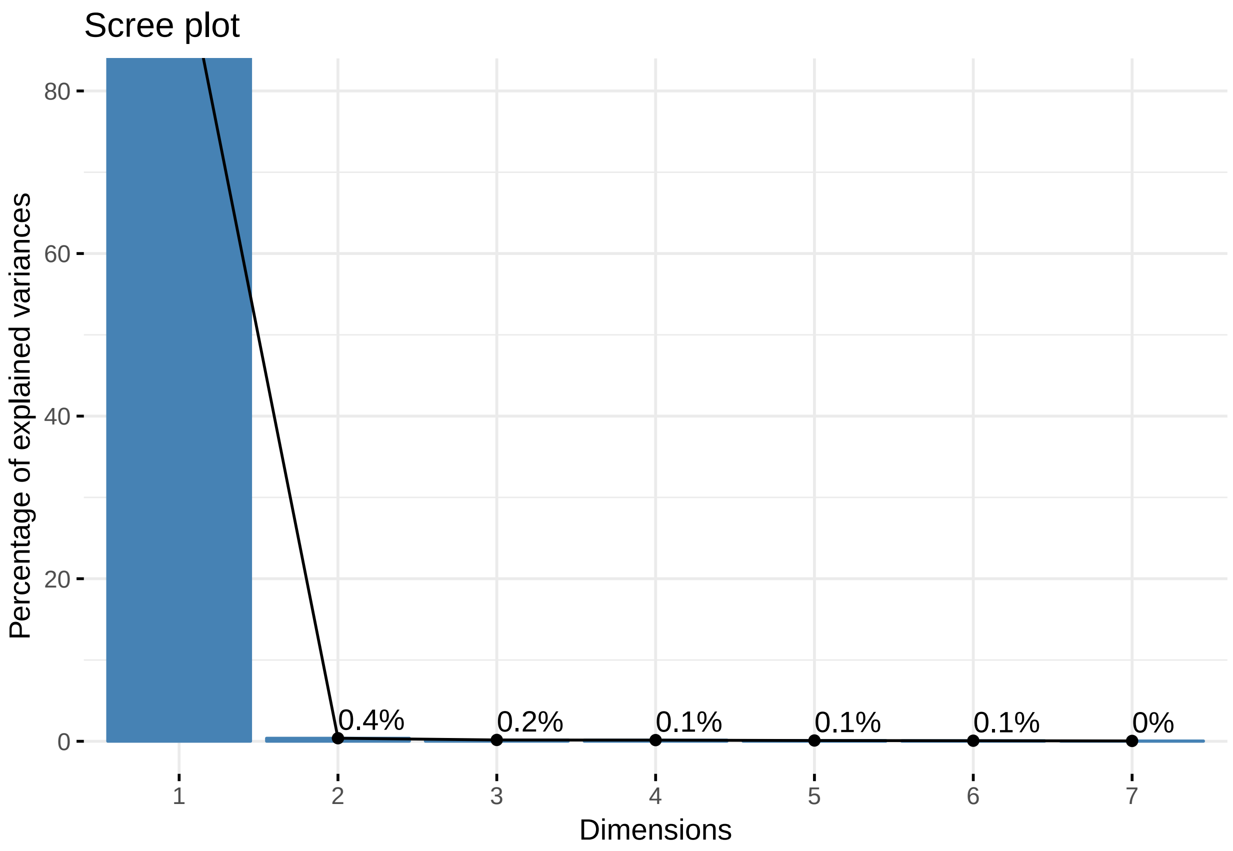
```
fviz_pca_var(prin_comp, col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"))
```



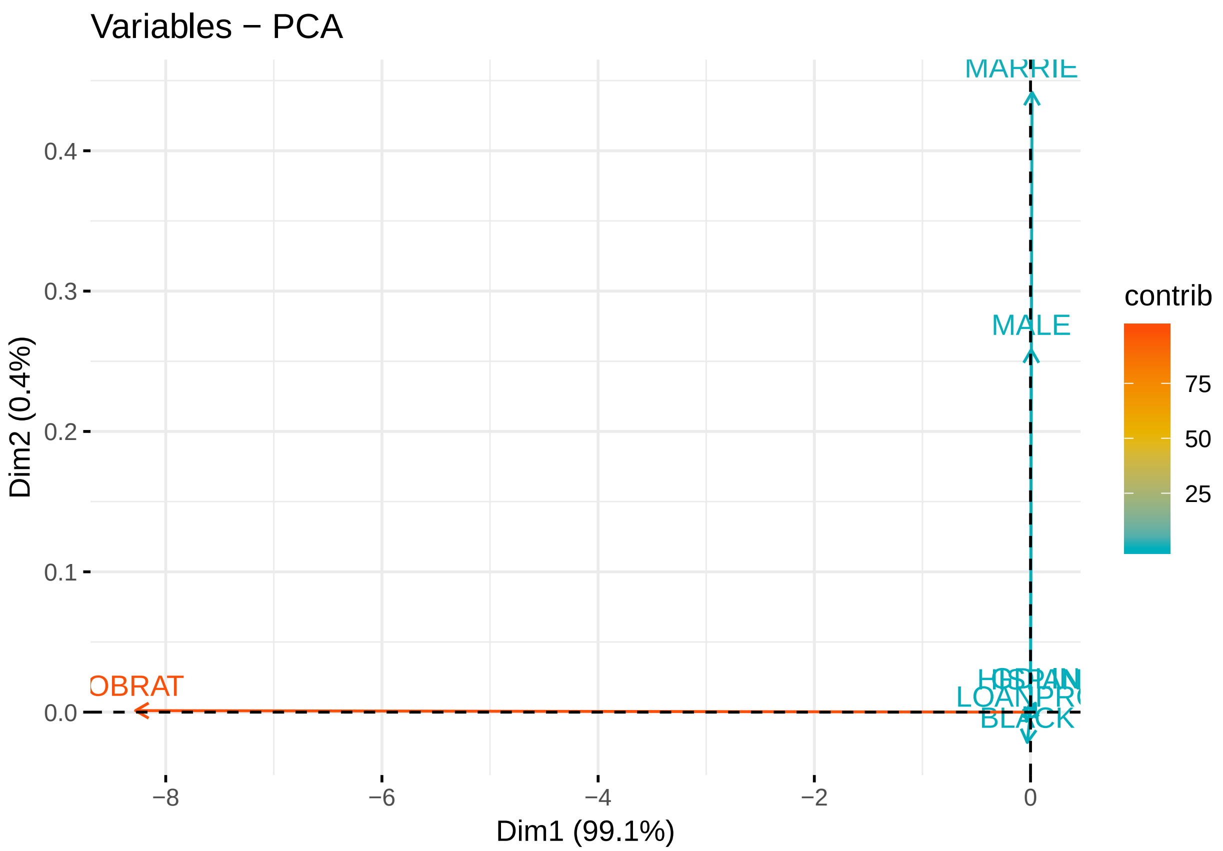
```
prin_comp <- prcomp(HMDA[,-7])
prin_comp
```

```
## Standard deviations (1, .., p=7):
## [1] 8.2796242 0.5122800 0.3394519 0.3178872 0.2590535 0.2243570 0.1796595
##
## Rotation (n x k) = (7 x 7):
##                   PC1          PC2           PC3          PC4          PC5
## MARRIED  0.0016794520  0.862302698 -0.5034846097  0.039780973 -0.016037536
## GDLIN    0.0055122577  0.012479946  0.0297255174  0.587264491  0.748789646
## OBRAT   -0.9999651848  0.002141502  0.0006815561  0.006507615  0.002899731
## BLACK   -0.0036299276 -0.042278448 -0.1682634679 -0.784387058  0.536108097
## HISPAN  -0.0008628778  0.011073814 -0.0227044975  0.048779084 -0.376490509
## MALE     0.0008335471  0.504179593  0.8466151401 -0.152327794  0.060078259
## LOANPRC -0.0046698959 -0.012905945 -0.0058217350 -0.112469241 -0.079259461
##                  PC6          PC7
## MARRIED -0.027378492 -0.018578868
## GDLIN    0.300175169 -0.057093000
## OBRAT   -0.001082159  0.003556696
## BLACK    0.236106234  0.107077821
## HISPAN   0.891972791  0.244174465
## MALE     0.045648884  0.012168519
## LOANPRC  0.236005679 -0.961845966
```

```r
fviz_eig(prin_comp, addlabels = TRUE, ylim = c(0, 80))
```



```r
fviz_pca_var(prin_comp, col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"))
```



The logit and probit models indicate that approval rates are higher for Whites than for Hispanic and Black applicants holding constant the loan to value, other obligations and creditworthiness.

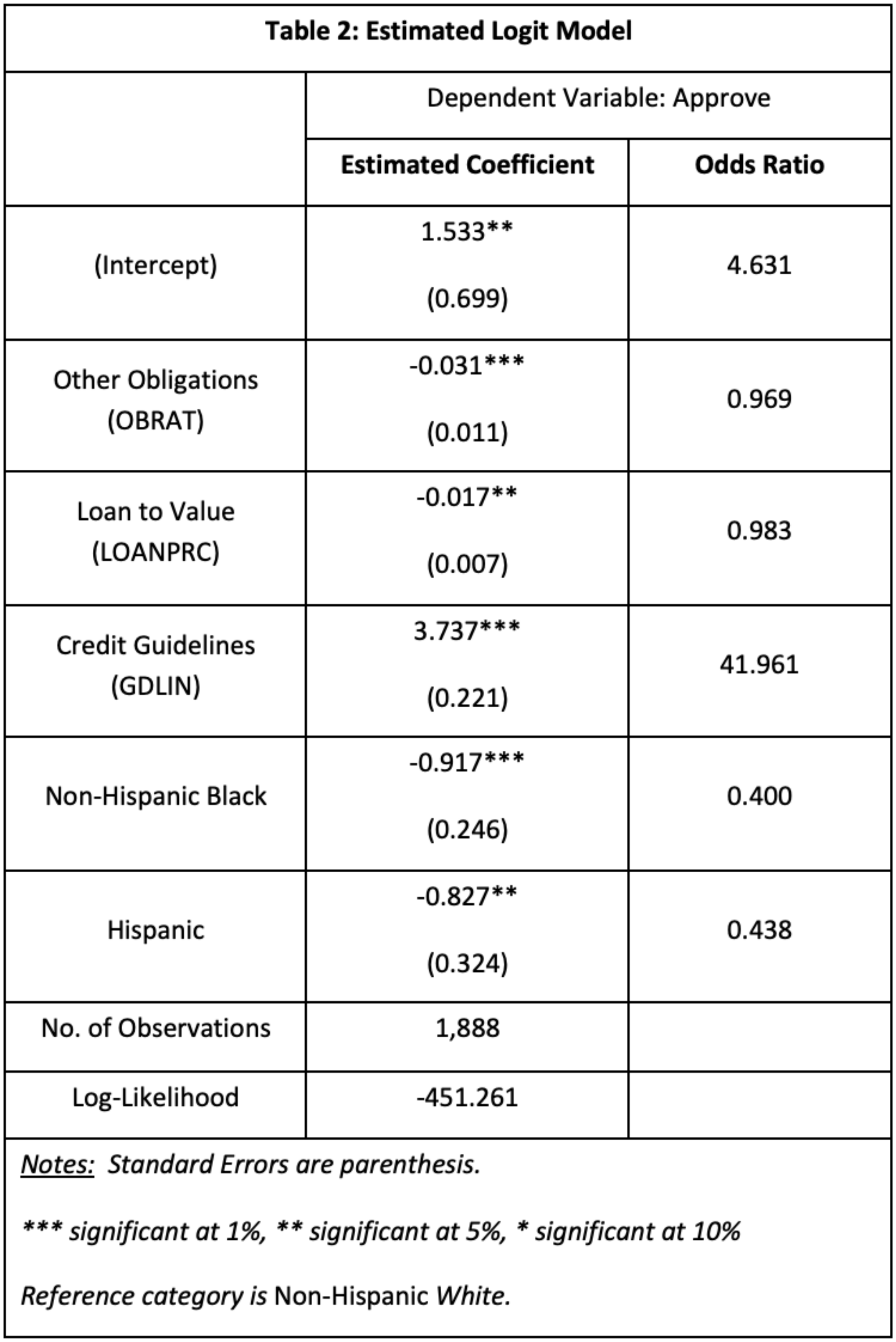


Table 2: Estimated Logit Model

	Dependent Variable: Approve	
	Estimated Coefficient	**Odds Ratio**
(Intercept)	1.533** (0.699)	4.631
Other Obligations (OBRAT)	-0.031*** (0.011)	0.969
Loan to Value (LOANPRC)	-0.017** (0.007)	0.983
Credit Guidelines (GDLIN)	3.737*** (0.221)	41.961
Non-Hispanic Black	-0.917*** (0.246)	0.400
Hispanic	-0.827** (0.324)	0.438
No. of Observations	1,888	
Log-Likelihood	-451.261	

Notes: Standard Errors are parenthesis.

****** significant at 1%, ** significant at 5%, * significant at 10%*

Reference category is Non-Hispanic *White.*

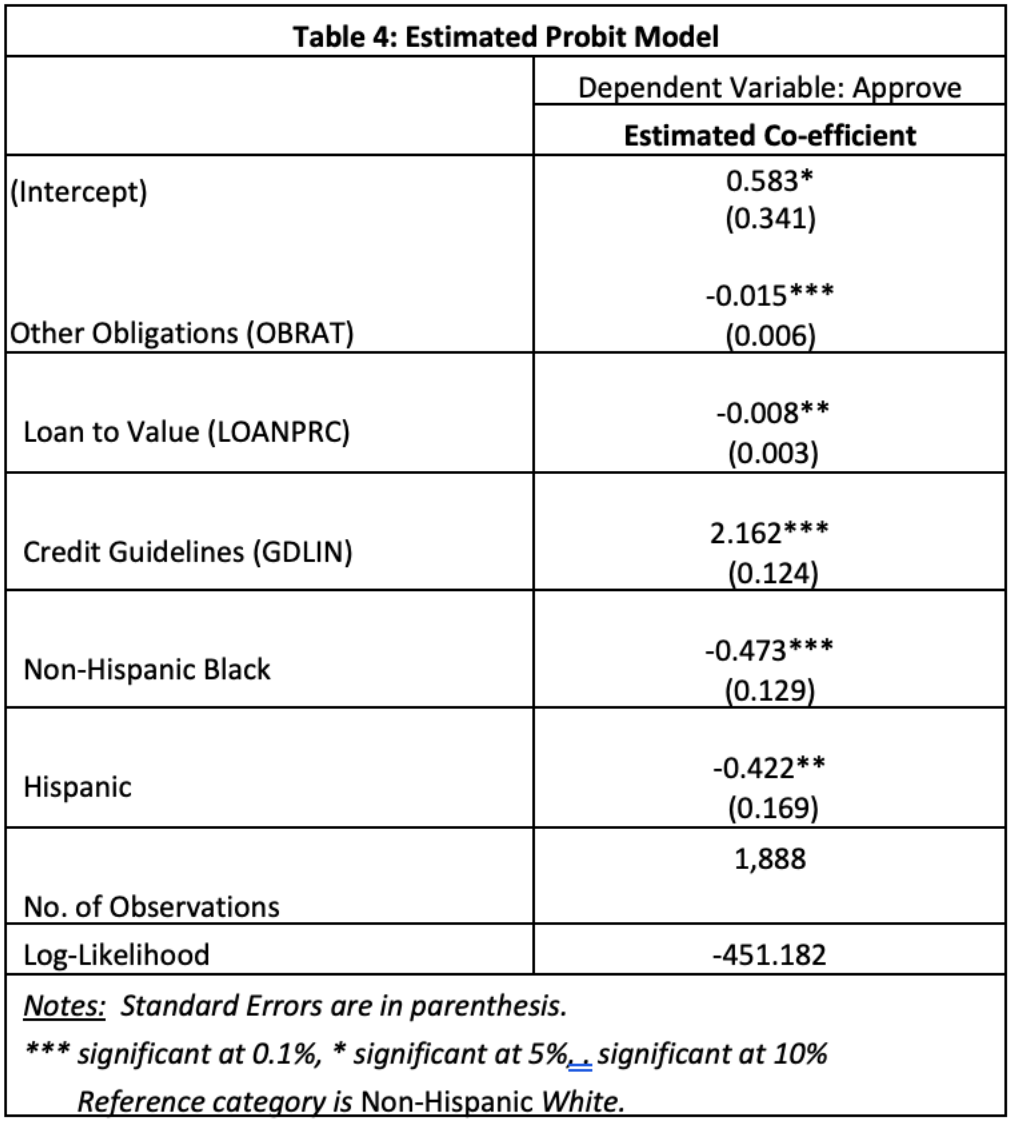


Table 4: Estimated Probit Model

	Dependent Variable: Approve
	Estimated Co-efficient
(Intercept)	0.583* (0.341)
Other Obligations (OBRAT)	-0.015*** (0.006)
Loan to Value (LOANPRC)	-0.008** (0.003)
Credit Guidelines (GDLIN)	2.162*** (0.124)
Non-Hispanic Black	-0.473*** (0.129)
Hispanic	-0.422** (0.169)
No. of Observations	1,888
Log-Likelihood	-451.182

Notes: Standard Errors are in parenthesis.

*** significant at 0.1%, * significant at 5%, . significant at 10%

Reference category is Non-Hispanic White.

Reference:

http://wise.xmu.edu.cn/course/gecon/hmda.pdf

Mortgage Lending in Boston: Interpreting HMDA Data," American Economic Review, 1996, pp. 25 – 53