DA410_Exam2_MattGraham

```
library(nnspat)  # used for dist2full()
library("dplyr")  # used to select numeric datatypes
library("ggplot2")
library(reshape)  # used for melting matricies
library(klaR)
library(ggvis)
library(class)
library(gmodels)
library(MASS)
library(readxl)
library(psych)
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.2.2
```

```
library(lavaan)
```

```
## Warning: package 'lavaan' was built under R version 4.2.2
```

```
library(semPlot)
```

```
## Warning: package 'semPlot' was built under R version 4.2.2
```

```
library(semTable)
```

```
## Warning: package 'semTable' was built under R version 4.2.2
```

```
library(kutils)
```

```
## Warning: package 'kutils' was built under R version 4.2.2
```

Problem 1

Get data

```
cov.mat <- data.frame(c(5, 0, 0), c(0, 9,0), c(0,0,9))
cov.mat
```

c.5..0..0. <dbl>	c.0..9..0. <dbl>	c.0..0..9. <dbl>
5	0	0
0	9	0
0	0	9

3 rows

A) Find eigenvalues and vectors

```
cov.mat.vals <- eigen(cov.mat)$values
cov.mat.vals
```

```
## [1] 9 9 5
```

```
cov.mat.vects <- eigen(cov.mat)$vectors
cov.mat.vects
```

```
##      [,1] [,2] [,3]
## [1,]    0    0    1
## [2,]    0    1    0
## [3,]    1    0    0
```

B) Find variance explained

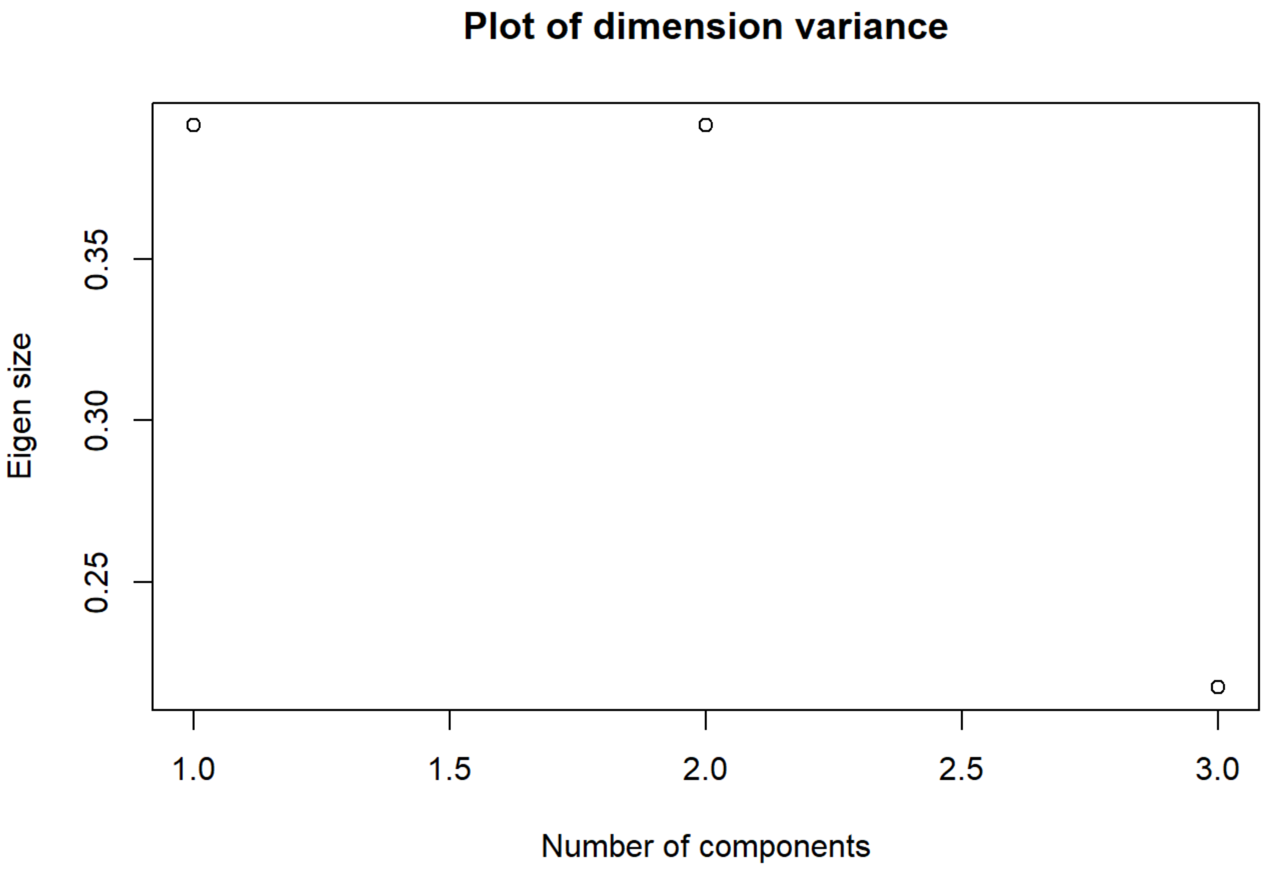
```
for (r in cov.mat.vals) {
  print(r/sum(cov.mat.vals))
}
```

```
## [1] 0.3913043
## [1] 0.3913043
## [1] 0.2173913
```

We can see that of our eigen values, ~80% of our variance is explained with 2 dimensions, while ~20% is explained with 1 dimension. This can be seen below.

Plot

```
plot(cov.mat.vals/sum(cov.mat.vals), xlab = 'Number of components', ylab='Eigen size', main='
Plot of dimension variance')
```



###C) Decision Ultimately, we will want to select 2 components in our analysis.

Problem 2

Assumption check:

Variables used should be metric. Dummy variables can also be considered, but only in special cases. -> check

Sample size: Sample size should be more than 200. -> check

Homogeneous sample: A sample should be homogenous. Violation of this assumption increases the sample size as the number of variables increases. Reliability analysis is conducted to check the homogeneity between variables.

Correlation: At least 0.30 correlations are required between the research variables.

```
french <- c(1, .44, .41, .29, .33, .25)
english <- c(.44, 1, .35, .35, .32, .33)
history <- c(.41, .35, 1, .16, .19, .18)
arithmetic <- c(.29, .35, .16, 1, .59, .47)
algebra <- c(.33, .32, .19, .59, 1, .46)
geometry <- c(.25, .33, .18, .47, .46, 1)

subject.cor <- cbind(french, english, history, arithmetic, algebra, geometry)
row.names(subject.cor) <- c('french', 'english', 'history', 'arithmetic', 'algebra', 'geometr
y')
as.data.frame(subject.cor)
```

	french	english	history	arithmetic	algebra	geometry
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
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

6 rows

We have a few correlations that are unable to be compared, and will be noted through analysis

Since we do not have a raw dataset, we assume there are no outliers.

Running fa

```
solution <- fa(r = subject.cor, nfactors = 2, rotate = "oblimin", fm="pa")
```

```
## Loading required namespace: GPArotation
```

```
## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : I
## am sorry, to do these rotations requires the GPArotation package to be installed
```

```
solution
```

```
## Factor Analysis using method =  pa
## Call: fa(r = subject.cor, nfactors = 2, rotate = "oblimin", fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##             PA1   PA2   h2   u2 com
## french     0.59  0.37 0.49 0.51 1.7
## english    0.59  0.23 0.41 0.59 1.3
## history    0.43  0.41 0.36 0.64 2.0
## arithmetic 0.71 -0.34 0.62 0.38 1.4
## algebra    0.70 -0.27 0.56 0.44 1.3
## geometry   0.58 -0.18 0.38 0.62 1.2
## 
##                        PA1  PA2
## SS loadings           2.22 0.59
## Proportion Var        0.37 0.10
## Cumulative Var        0.37 0.47
## Proportion Explained  0.79 0.21
## Cumulative Proportion 0.79 1.00
## 
## Mean item complexity =  1.5
## Test of the hypothesis that 2 factors are sufficient.
## 
## The degrees of freedom for the null model are  15  and the objective function was  1.43
## The degrees of freedom for the model are 4  and the objective function was  0.01 
## 
## The root mean square of the residuals (RMSR) is  0.01 
## The df corrected root mean square of the residuals is  0.03 
## 
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy             
##                                                    PA1  PA2
## Correlation of (regression) scores with factors   0.90 0.73
## Multiple R square of scores with factors          0.82 0.53
## Minimum correlation of possible factor scores     0.63 0.06
```

Overall, our model does a great job explaining ~90% of variation when using 2 factors. Our most-ideal values to model from would be arithmetic and algebra. We can also see in our console output that hypothesis tests with 2 factors are sufficient. Neither of these have correlations below .30.

Problem 3

Get data

```
food.stuff <- read.table("C:/mattgraham93.github.io/school/22_3_DA410/data/foodstuff.dat", header=TRUE)
food.stuff <- food.stuff[-1]
food.stuff
```

Energy	Protein	Fat	Calcium	Iron
<int>	<int>	<int>	<int>	<dbl>

Energy <int>	Protein <int>	Fat <int>	Calcium <int>	Iron <dbl>
340	20	28	9	2.6
245	21	17	9	2.7
420	15	39	7	2.0
375	19	32	9	2.5
180	22	10	17	3.7
115	20	3	8	1.4
170	25	7	12	1.5
160	26	5	14	5.9
265	20	20	9	2.6
300	18	25	9	2.3

1-10 of 27 rows Previous **1** 2 3 Next

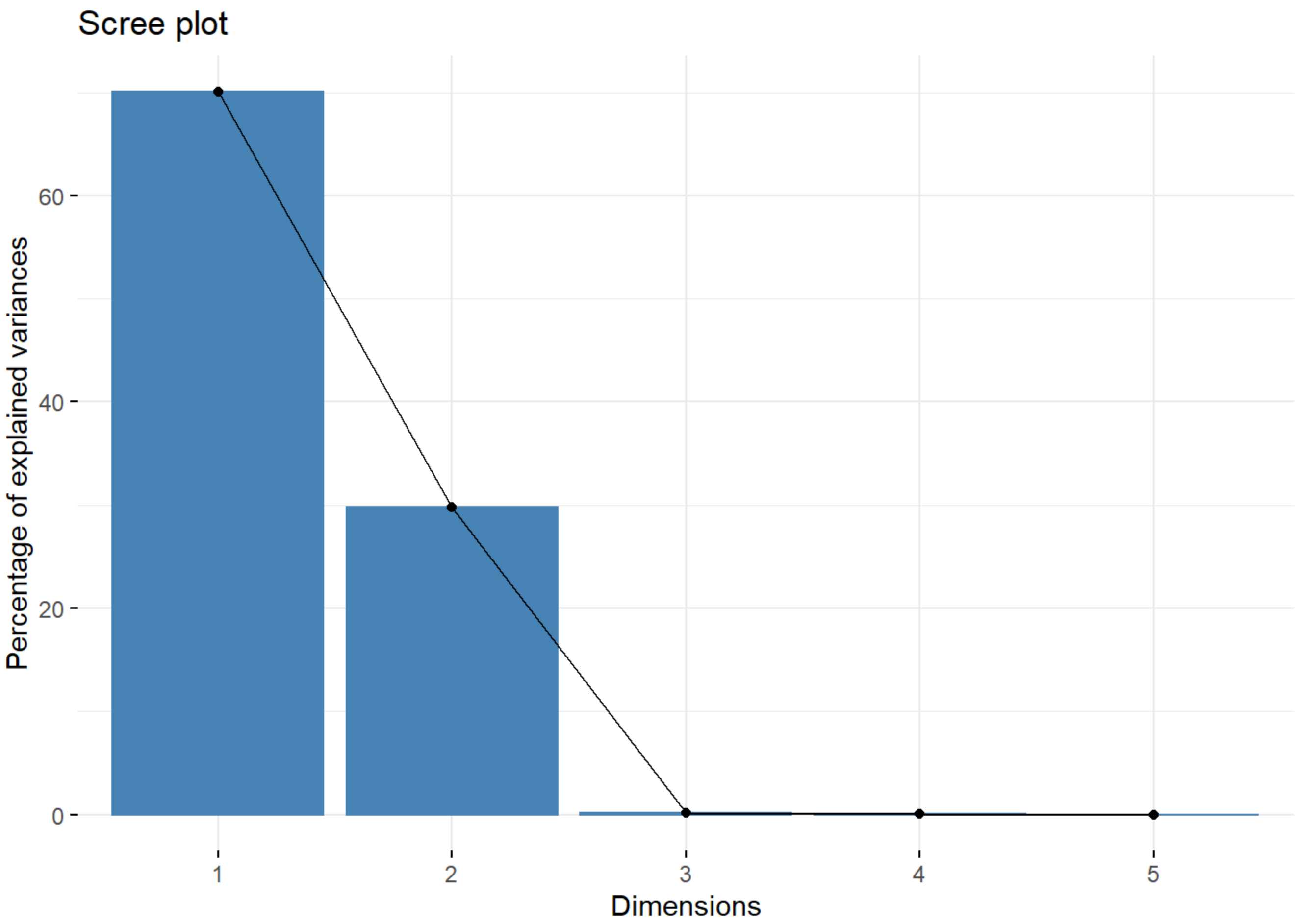
a. Determine factors to use

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.2.2
```

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

```
food.stuff.pca <- prcomp(food.stuff)
fviz_eig(food.stuff.pca)
```



We will ultimately use 2 dimensions when concluding our analysis.

B - obtaining loadings

```
S <- cov(food.stuff)
R <- cor(food.stuff)
```

S

```
as.data.frame(S)
```

	Energy	Protein	Fat	Calcium	Iron
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Energy	10243.01994	74.807692	1124.5655271	-2530.292023	-14.7521368
Protein	74.80769	18.076923	1.1923077	-28.230769	-1.0846154
Fat	1124.56553	1.192308	126.7207977	-270.673789	-0.9965812
Calcium	-2530.29202	-28.230769	-270.6737892	6089.344729	5.0491453
Iron	-14.75214	-1.084615	-0.9965812	5.049145	2.1341026

5 rows

R

```
as.data.frame(R)
```

	Energy <dbl>	Protein <dbl>	Fat <dbl>	Calcium <dbl>	Iron <dbl>
Energy	1.00000000	0.17384812	0.98706740	-0.32038440	-0.09977765
Protein	0.17384812	1.00000000	0.02491163	-0.08508934	-0.17462478
Fat	0.98706740	0.02491163	1.00000000	-0.30813212	-0.06060118
Calcium	-0.32038440	-0.08508934	-0.30813212	1.00000000	0.04429196
Iron	-0.09977765	-0.17462478	-0.06060118	0.04429196	1.00000000

5 rows

Get eigenvalues and eigenvectors of S and R

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

Eigen S

```
eig.S
```

```
## eigen() decomposition
## $values
## [1] 1.155253e+04 4.903923e+03 2.042503e+01 2.066907e+00 3.516836e-01
## 
## $vectors
##              [,1]          [,2]         [,3]          [,4]          [,5]
## [1,]  0.901061141  0.4195897978 -0.034918237 -0.0089992248  0.1035999595
## [2,]  0.006887716  0.0011983568 -0.924379029  0.1023111641 -0.3674329322
## [3,]  0.098689332  0.0474125325  0.374781853  0.0877224332 -0.9164364709
## [4,] -0.422255831  0.9064738840 -0.002194532 -0.0001874493  0.0005097596
## [5,] -0.001344580 -0.0003389534  0.061950579  0.9908361011  0.1200167645
```

Eigen R

```
eig.R
```

```
## 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
```

c. Obtain scores

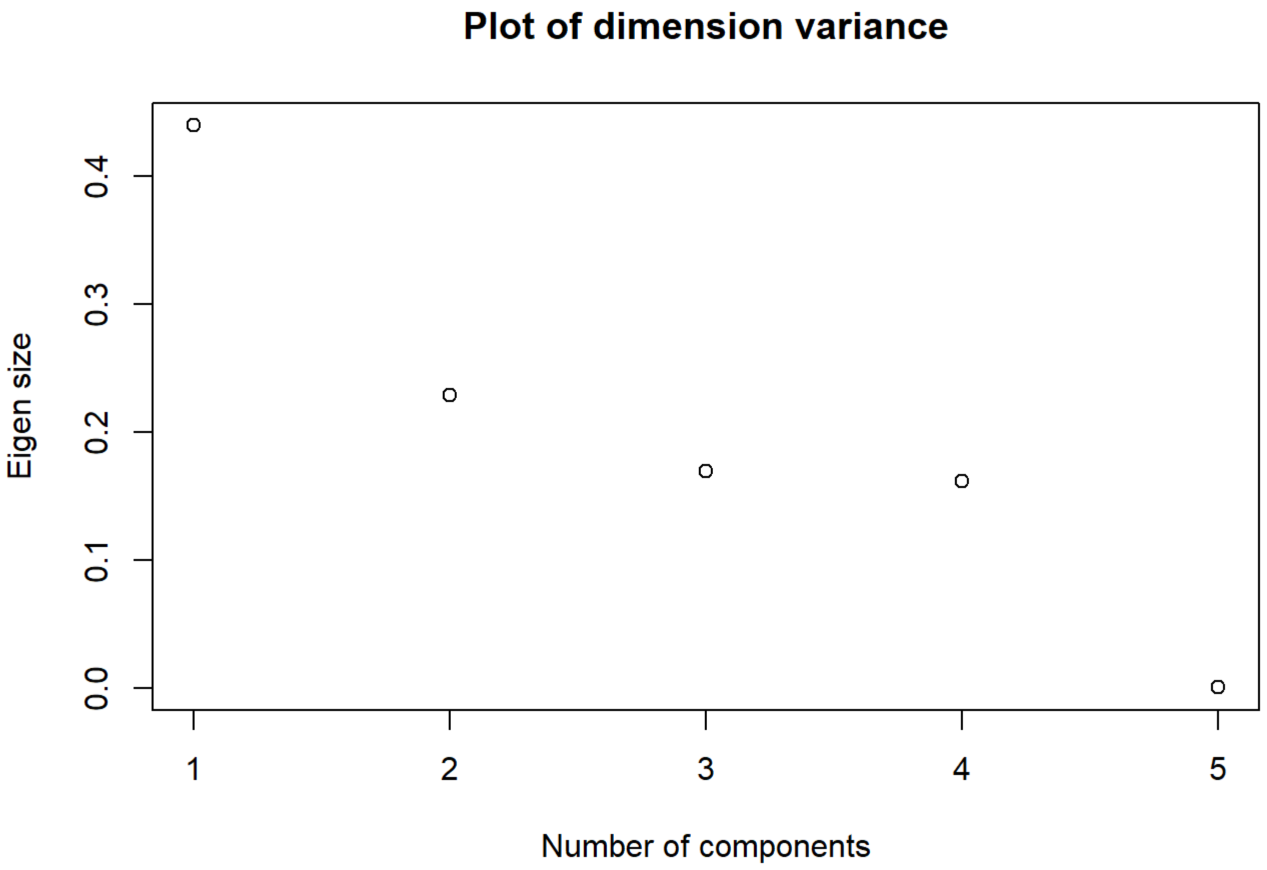
```
for (r in eig.R$values) {
  print(r/sum(eig.R$values))
}
```

```
## [1] 0.4395555
## [1] 0.228841
## [1] 0.1697149
## [1] 0.1615686
## [1] 0.0003200338
```

We can see that of our eigen values, ~65% of our variance is explained with just two dimensions, and inerestingly enough going with all 5 shows almost no meaningful value. We can see this below.

Plot

```
plot(eig.R$values/sum(eig.R$values), xlab = 'Number of components', ylab='Eigen size', main='
Plot of dimension variance')
```



e. Interpretation

Over the impact of foods' macros pertaining to total energy, as we model our data, we can conclude that most of our variation happens within the first 2 measures compared to subsequent ones. This makes sense as protein and fat are our primary determinate for overall macro tracking and impact caloric intake.

Problem 4

```
scores <- read.table("C:/mattgraham93.github.io/school/22_3_DA410/data/test_score.dat", heade
r=TRUE)
scores <- scores[-1]
scores
```

math	reading	sex
<dbl>	<dbl>	<chr>
83.16	79.67	boy
102.51	101.13	boy
81.63	80.53	boy
88.25	84.58	boy

math <dbl>	reading <dbl>	sex <chr>
81.47	76.52	boy
87.19	84.70	boy
88.66	85.86	boy
79.35	81.03	boy
83.35	80.44	boy
86.58	84.67	boy

1-10 of 62 rows Previous **1** 2 3 4 5 6 7 Next

Hotelling’s test

```
summary(manova(cbind(math, reading) ~sex, data=scores), test="Hotelling")
```

```
##           Df Hotelling-Lawley approx F num Df den Df    Pr(>F)    
## sex        1          0.30593   9.0249      2     59 0.0003805 ***
## Residuals 60                                                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hotelling’s Analysis

At alpha = 0.05 and p-value < 0.05, we can conclude there is sufficient evidence to state there are differences between mean math and reading scores between the recorded sexes.

Problem 5

Get data

```
glucose <- read.table("C:/mattgraham93.github.io/school/22_3_DA410/data/T3_5_DIABETES.DAT", h
eader=FALSE)[1:34,]
glucose <- glucose[-1]
colnames(glucose) <- c('rel_wt', 'fst_pls_glu', 'gl_int', 'ins_resp', 'ins_resist')

ys <- glucose[1:2]
xs <- glucose[3:5]

glucose
```

	rel_wt <dbl>	fst_pls_glu <int>	gl_int <int>	ins_resp <int>	ins_resist <int>
1	0.81	80	356	124	55

	rel_wt <dbl>	fst_pls_glu <int>	gl_int <int>	ins_resp <int>	ins_resist <int>
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

1-10 of 34 rows Previous **1** 2 3 4 Next

Find means

```
x.bar <- colMeans(xs)
x.bar
```

```
##     gl_int   ins_resp ins_resist
##  341.47059  175.32353   99.11765
```

```
y.bar <- colMeans(ys)
y.bar
```

```
##      rel_wt fst_pls_glu
##   0.9164706  89.6470588
```

a - Find canonical correlations

```
cancor2<-function(x,y,dec=4){
#Canonical Correlation Analysis to mimic SAS PROC CANCOR output.
#Basic formulas can be found in Chapter 10 of Mardia, Kent, and Bibby (1979).
# The approximate F statistic is exercise 3.7.6b.
    x<-as.matrix(x)
    y<-as.matrix(y)

    n<-dim(x)[1]
    q1<-dim(x)[2]
    q2<-dim(y)[2]
    q<-min(q1,q2)

    S11<-cov(x)
    S12<-cov(x,y)
    S21<-t(S12)
    S22<-cov(y)

    E1<-eigen(solve(S11)%*%S12%*%solve(S22)%*%S21)
    E2<-eigen(solve(S22)%*%S21%*%solve(S11)%*%S12)

    rsquared<-as.double(E1$values[1:q])

    LR<-NULL;pp<-NULL;qq<-NULL;tt<-NULL

    for (i in 1:q){
        LR<-c(LR,prod(1-rsquared[i:q]))
        pp<-c(pp,q1-i+1)
        qq<-c(qq,q2-i+1)
        tt<-c(tt,n-1-i+1)}

    m<-tt-0.5*(pp+qq+1);lambda<-(1/4)*(pp*qq-2);s<-sqrt((pp^2*qq^2-4)/(pp^2+qq^2-5))
    F<-((m*s-2*lambda)/(pp*qq))*((1-LR^(1/s))/LR^(1/s))
    df1<-pp*qq;df2<-(m*s-2*lambda)
    pval<-1-pf(F,df1,df2)
    outmat<-round(cbind(sqrt(rsquared),rsquared,LR,F,df1,df2,pval),dec)

    colnames(outmat) <- list("R","RSquared","LR","ApproxF","NumDF","DenDF","pvalue")
    rownames(outmat) <- as.character(1:q)
    xrels<-round(cor(x,x%*%E1$vectors)[,1:q],dec)
    colnames(xrels)<-apply(cbind(rep("U",q),as.character(1:q)),1,paste,collapse="")
    yrels<-round(cor(y,y%*%E2$vectors)[,1:q],dec)
    colnames(yrels)<- apply(cbind(rep("V",q),as.character(1:q)),1,paste,collapse="")
    list(Summary=outmat,
         a.Coefficients=E1$vectors,
         b.Coefficients=E2$vectors,
         XUCorrelations=xrels,YVCorrelations=yrels
     )
   }
## END FUNCTION
#################################################
```

b - Find standard coefficients

For canonical variables

Fasting coefficients

```
before.coefficients <- cancor2(xs, ys)$a.Coefficients
after.coefficients <- cancor2(xs, ys)$b.Coefficients

diag(before.coefficients)
```

```
## [1] 0.42680635 0.07577154 0.27703237
```

Post-consumption coefficients

```
diag(after.coefficients)
```

```
## [1] 0.99999176 0.09433123
```

c - Test significance for reach canonical correlation

```
cancor2(xs, ys)
```

```
## $Summary
##        R RSquared     LR ApproxF NumDF DenDF pvalue
## 1 0.6024   0.3628 0.6353  2.4616     6    58 0.0345
## 2 0.0546   0.0030 0.9970     NaN     2   NaN    NaN
## 
## $a.Coefficients
##            [,1]        [,2]       [,3]
## [1,]  0.4268063 -0.96431194 0.06551464
## [2,] -0.5231307  0.07577154 0.95862448
## [3,]  0.7376792  0.25369501 0.27703237
## 
## $b.Coefficients
##              [,1]       [,2]
## [1,]  0.999991761 0.99554087
## [2,] -0.004059221 0.09433123
## 
## $XUCorrelations
##                 U1      U2
## gl_int      0.2754 -0.9065
## ins_resp   -0.2448 -0.0518
## ins_resist  0.7781  0.3186
## 
## $YVCorrelations
##                  V1     V2
## rel_wt       0.9691 0.2465
## fst_pls_glu -0.1661 0.9861
```

```
# It produces two other pieces of information:  An F-test for the
# significance of each canonical correlation, and the correlations between
# the original variables and the corresponding canonical variates.
```

Interpretation

Given all our p-values < 0.05, there is enough evidence to conclude that there is at least one non-zero canonical correlation between relative weight and plasma glucose across glucose intolerance, insulin response to oral glucose, and insulin resistance.This means our subjects had different responses to ingesting glucose. This makes sense as diabetes and insulin are directly correlated.

Problem 6

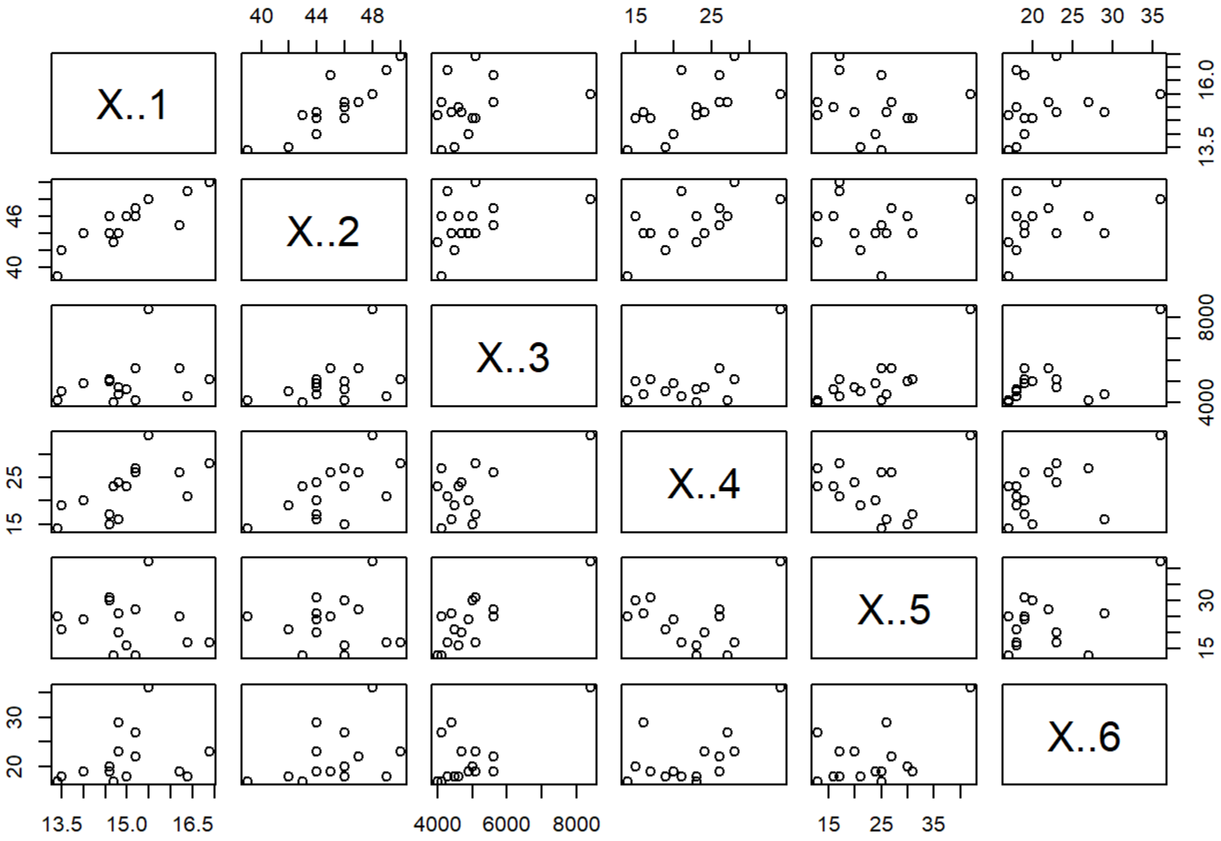
```
hematology <- read.csv('C:/mattgraham93.github.io/school/22_3_DA410/data/hematology.csv', hea
der=TRUE)
hematology
```

Observation.number	X..1	X..2	X..3	X..4	X..5	X..6
<int>	<dbl>	<int>	<int>	<int>	<int>	<int>
1	13.4	39	4100	14	25	17

Observation.number	X..1	X..2	X..3	X..4	X..5	X..6
<int>	<dbl>	<int>	<int>	<int>	<int>	<int>
2	14.6	46	5000	15	30	20
3	13.5	42	4500	19	21	18
4	15.0	46	4600	23	16	18
5	14.6	44	5100	17	31	19
6	14.0	44	4900	20	24	19
7	16.4	49	4300	21	17	18
8	14.8	44	4400	16	26	29
9	15.2	46	4100	27	13	27
10	15.5	48	8400	34	42	36

1-10 of 15 rows Previous **1** 2 Next

```
pairs(hematology[-1])
```

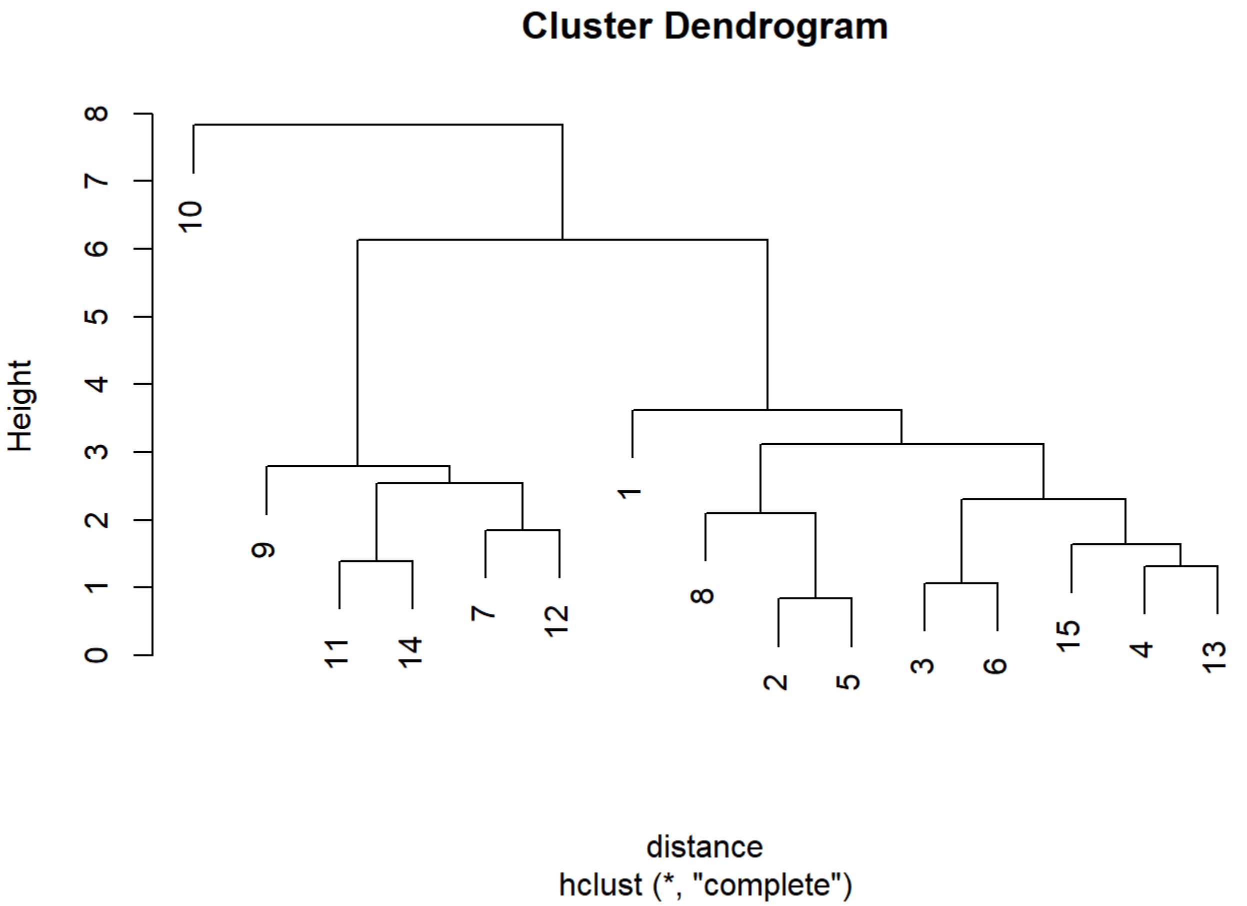


Normalizing

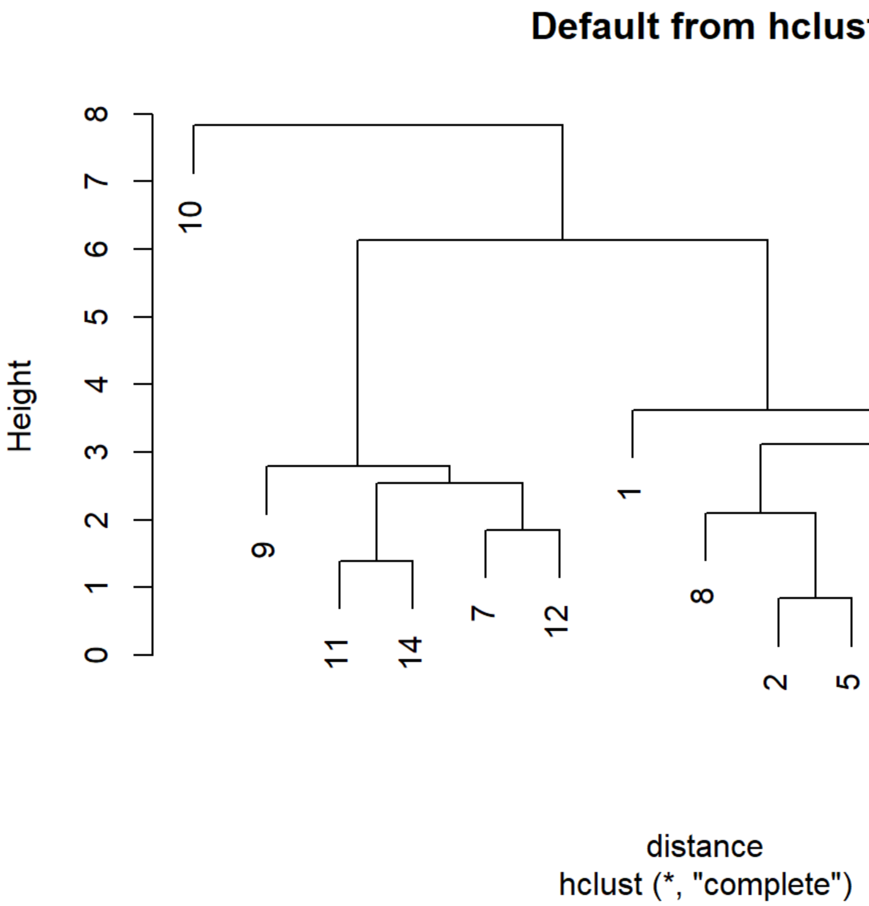
```
z <- hematology[,-c(1,1)]
means <- apply(z,2,mean)
sds <- apply(z,2,sd)
nor <- scale(z,center=means,scale=sds)

distance = dist(nor)
```

Plotting

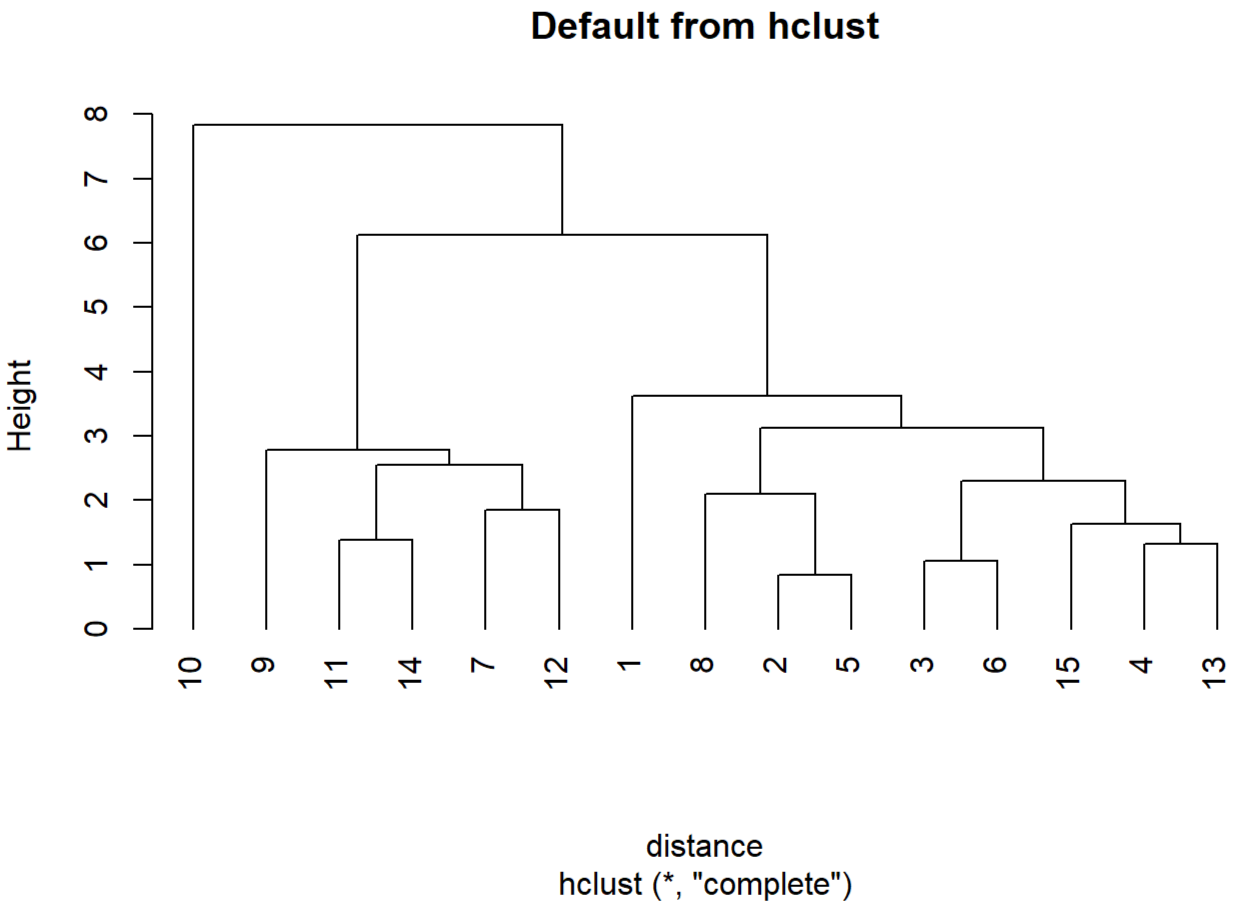
```
mydata.hclust = hclust(distance)
plot(mydata.hclust)
```



```
plot(mydata.hclust,labels=hematology$Observation.number,main='Default from hclust')
```

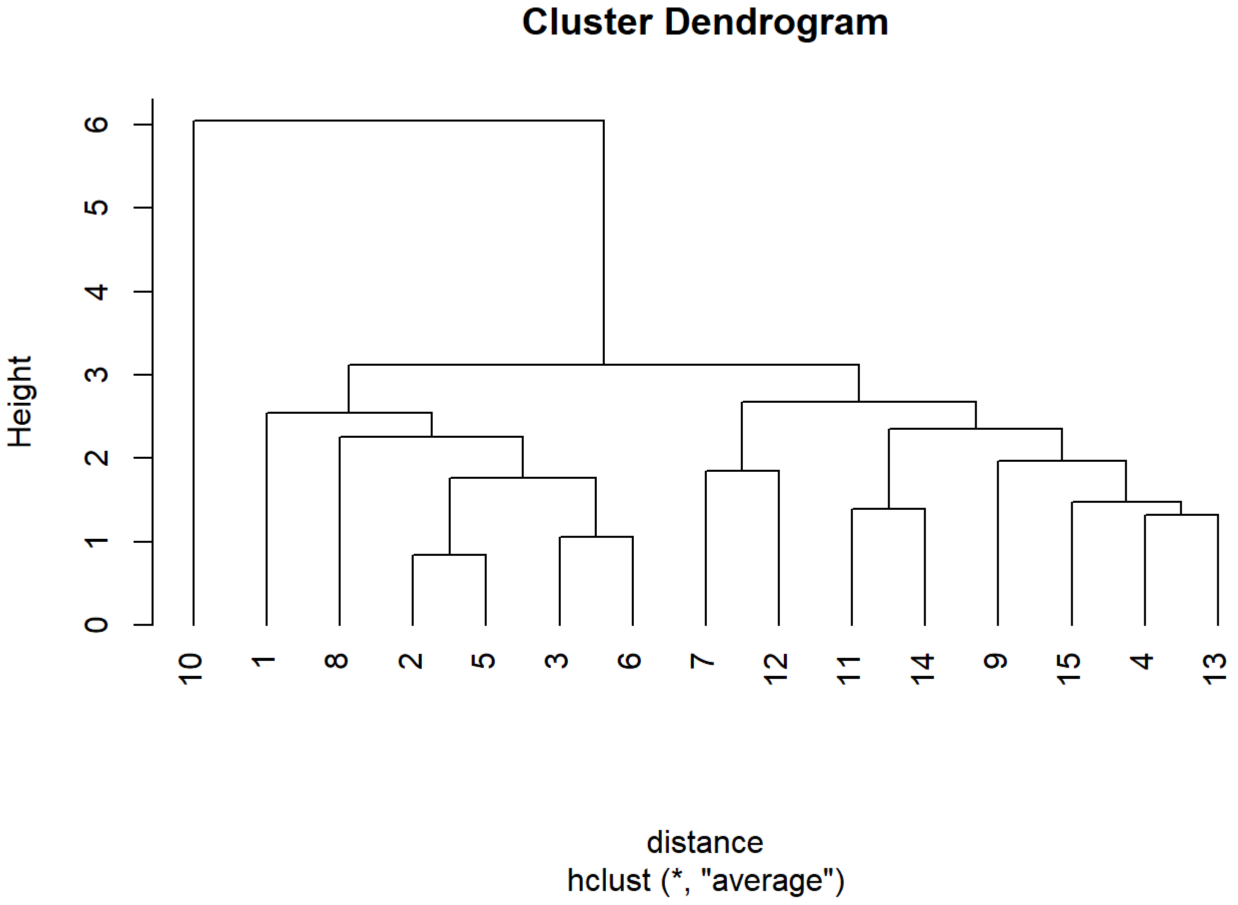


```
plot(mydata.hclust,hang=-1, labels=hematology$Observation.number,main='Default from hclust')
```



Average linkage

```
mydata.hclust<-hclust(distance,method="average")
plot(mydata.hclust,hang=-1)
```


Cluster Dendrogram
Height
0
1
2
3
4
5
6
10
1
8
2
5
3
6
7
12
11
14
9
15
4
13
distance
hclust (*, "average")