Assignment2

Michael Nefiodovas(22969312) Carmen Leong(22789943)
Nicholas Choong(21980614)

Question 1

(a) Why should you not automatically scale the data prior to a PCA or FA? Restrict your answer to one or two concise sentences.

Scaling can possibly cause information loss especially for variables with high magnitude. If the variables with high magnitude are important, scaling is not advised since scaling centres all the variables and transforms their variability and range to more comparable ranges. Moreover, if the variables in the data set have same units of measurement, scaling is not necessarily required.

(b) The dataset ass2pop.csv is available in the LMS folder 'Data sets'. For a description of the data see Assignment 1. Here we work with a part of the dataset only. Let Σ be the covariance matrix consisting of rows 1:11, and columns 3:13. Read the data into R. The value for $\Sigma[1, 1]$ should be 0.8266. In your answer show the R commands you use to calculate the following and show the results stating clearly what each part is.

```
ass2pop <- read.csv("ass2pop.csv", header = FALSE)
S0 <- as.matrix(ass2pop[1:11, 3:13])
dim(S0)
```

```
## [1] 11 11
```

The covariance matrix is a 11x11 square matrix giving the covariance between each pair of the first 11 variables from the first population.

i. the eigenvalues of Σ;

```
ev <- eigen(S0)
vals <- ev$values
knitr::kable(round(vals, digits = 4))
```

x
4.8414
3.5168

x
0.8050
0.6695
0.4372
0.2438
0.1240
0.0466
0.0421
0.0079
0.0005

The eigenvalues of the covariance matrix encode the variability of the data in an orthogonal basis that captures as much of the data's variability as possible.

ii. the matrix $\Sigma^{2/3}$;

```
V1 <- ev$vectors
vals2 <- diag(vals^(2 / 3))
S1 <- V1 %*% vals2 %*% t(V1)
knitr::kable(round(S1, digits = 4))
```

0.5510	0.0438	0.5320	0.5971	-0.0821	0.0748	0.2192	0.3190	-0.0529	-0.2479	0.3658
0.0438	0.8358	0.0475	0.0489	-0.0851	0.0300	0.0445	-0.0020	-0.0546	-0.0256	0.0147
0.5320	0.0475	0.5309	0.5853	-0.0551	0.1321	0.2589	0.3407	-0.0215	-0.1996	0.3704
0.5971	0.0489	0.5853	0.7278	-0.0686	0.0759	0.2607	0.3572	-0.0725	-0.2604	0.4790
-0.0821	-0.0851	-0.0551	-0.0686	0.7015	0.3198	0.3202	0.2787	0.2735	0.4384	0.0764
0.0748	0.0300	0.1321	0.0759	0.3198	0.7711	0.4925	0.3737	0.4133	0.5291	0.1391
0.2192	0.0445	0.2589	0.2607	0.3202	0.4925	0.6541	0.4535	0.2714	0.2616	0.2635
0.3190	-0.0020	0.3407	0.3572	0.2787	0.3737	0.4535	0.5657	0.2146	0.1468	0.2996
-0.0529	-0.0546	-0.0215	-0.0725	0.2735	0.4133	0.2714	0.2146	0.8877	0.3808	0.0589
-0.2479	-0.0256	-0.1996	-0.2604	0.4384	0.5291	0.2616	0.1468	0.3808	0.8781	-0.0615
0.3658	0.0147	0.3704	0.4790	0.0764	0.1391	0.2635	0.2996	0.0589	-0.0615	1.2128

iii. the matrix $2\Sigma^{-1/4}\Sigma\Sigma^{-1/4}$and its eigenvalues

```
vals3 <- diag(vals^(-1 / 4))
S2 <- V1 %*% vals3 %*% t(V1)
mat <- 2 * S2 %*% S0 %*% S2
knitr::kable(round(mat, digits = 4))
```

0.9537	0.0576	0.8818	0.9432	-0.1414	0.0979	0.2978	0.4750	-0.0818	-0.3816	0.4798
0.0576	1.7438	0.0651	0.0646	-0.1381	0.0540	0.0763	-0.0121	-0.0847	-0.0294	0.0141
0.8818	0.0651	0.9087	0.9222	-0.1081	0.2010	0.3672	0.5069	-0.0393	-0.3080	0.4873
0.9432	0.0646	0.9222	1.2679	-0.1115	0.0829	0.3634	0.5151	-0.1218	-0.3856	0.6637
-0.1414	-0.1381	-0.1081	-0.1115	1.4128	0.4138	0.4753	0.4389	0.3642	0.6492	0.1079
0.0979	0.0540	0.2010	0.0829	0.4138	1.4363	0.7426	0.5375	0.5919	0.8175	0.1644

0.2978	0.0763	0.3672	0.3634	0.4753	0.7426	1.2474	0.6840	0.3677	0.3596	0.3392
0.4750	-0.0121	0.5069	0.5151	0.4389	0.5375	0.6840	1.0794	0.2985	0.1988	0.3845
-0.0818	-0.0847	-0.0393	-0.1218	0.3642	0.5919	0.3677	0.2985	1.7459	0.5233	0.0773
-0.3816	-0.0294	-0.3080	-0.3856	0.6492	0.8175	0.3596	0.1988	0.5233	1.6351	-0.0800
0.4798	0.0141	0.4873	0.6637	0.1079	0.1644	0.3392	0.3845	0.0773	-0.0800	2.2294

```
vals4 <- eigen(mat)$values
vals4
```

```
##  [1] 4.40065581 3.75063249 1.79442970 1.63648238 1.32247911 0.98757168
##  [7] 0.70420091 0.43163061 0.41057352 0.17738881 0.04429074
```

Question 2

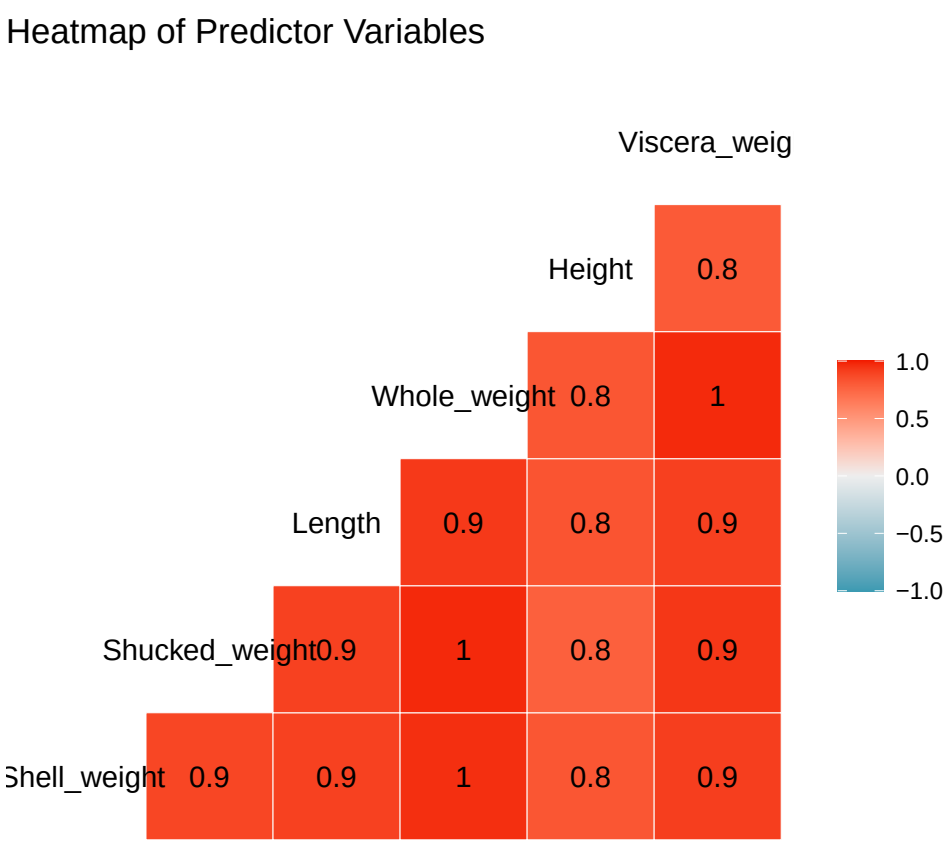
Consider the abalone data. We want to compare the performance of linear regression and PCR for the raw abalone data following the description given in Q3 of Lab 3. In the analysis we use the predictor variables Length, Height, Whole Weight, Shucked Weight, Viscera Weight and Dried-Shell Weight and we consider Rings as the response variable. Hint. Note the change of predictor variables used in Q2 compared to the variables in the Lab.

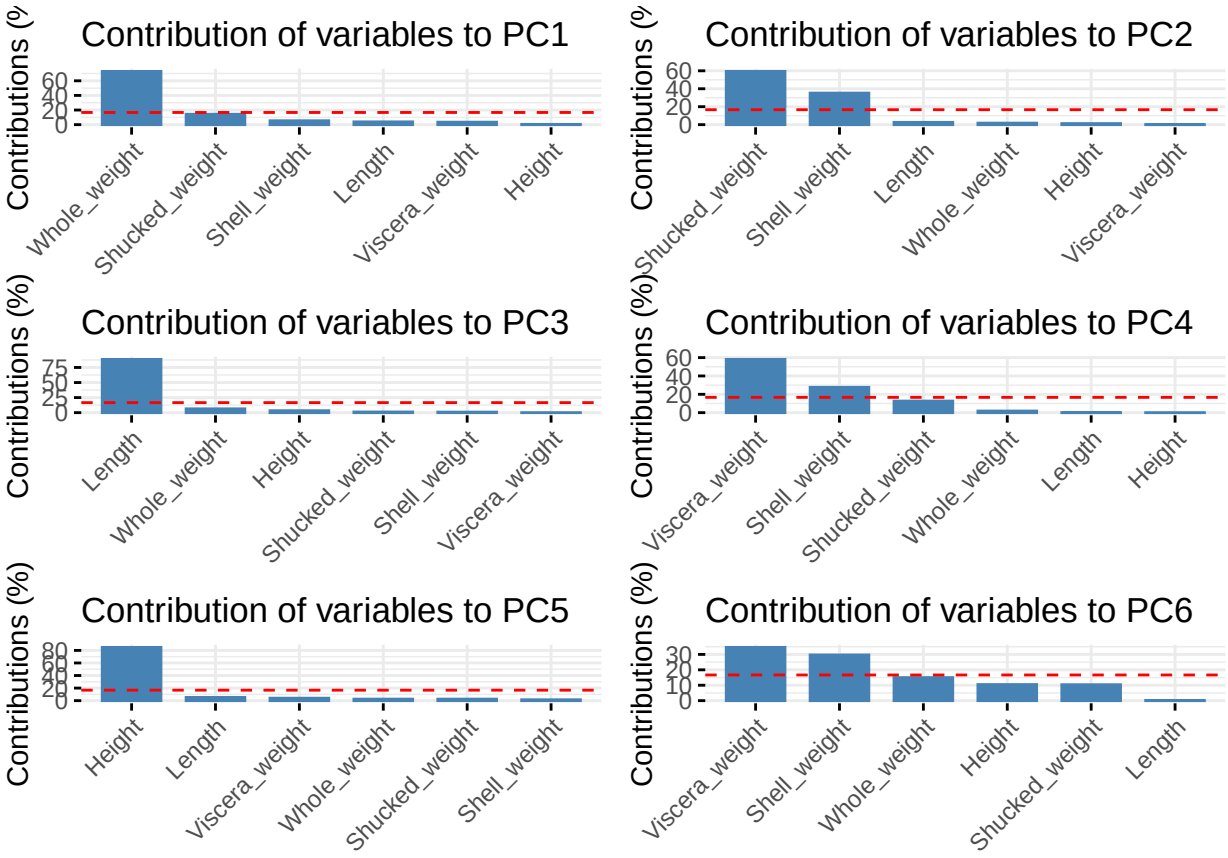
(a) For the regular linear regression use forward selection and state the order in which the variables are chosen. Calculate the residual standard deviation for each number of predictors. Hint. you may make use of the code in Lab 3.

	modelno	sigma
(Intercept)	0	3.224169
Shell_weight	1	2.510500
Shucked_weight	2	2.339087
Length	3	2.288719
Whole_weight	4	2.259298
Height	5	2.242614
Viscera_weight	6	2.227005

Shell_weight, Shucked_weight, Length, Whole_weight, Height, Viscera_weight

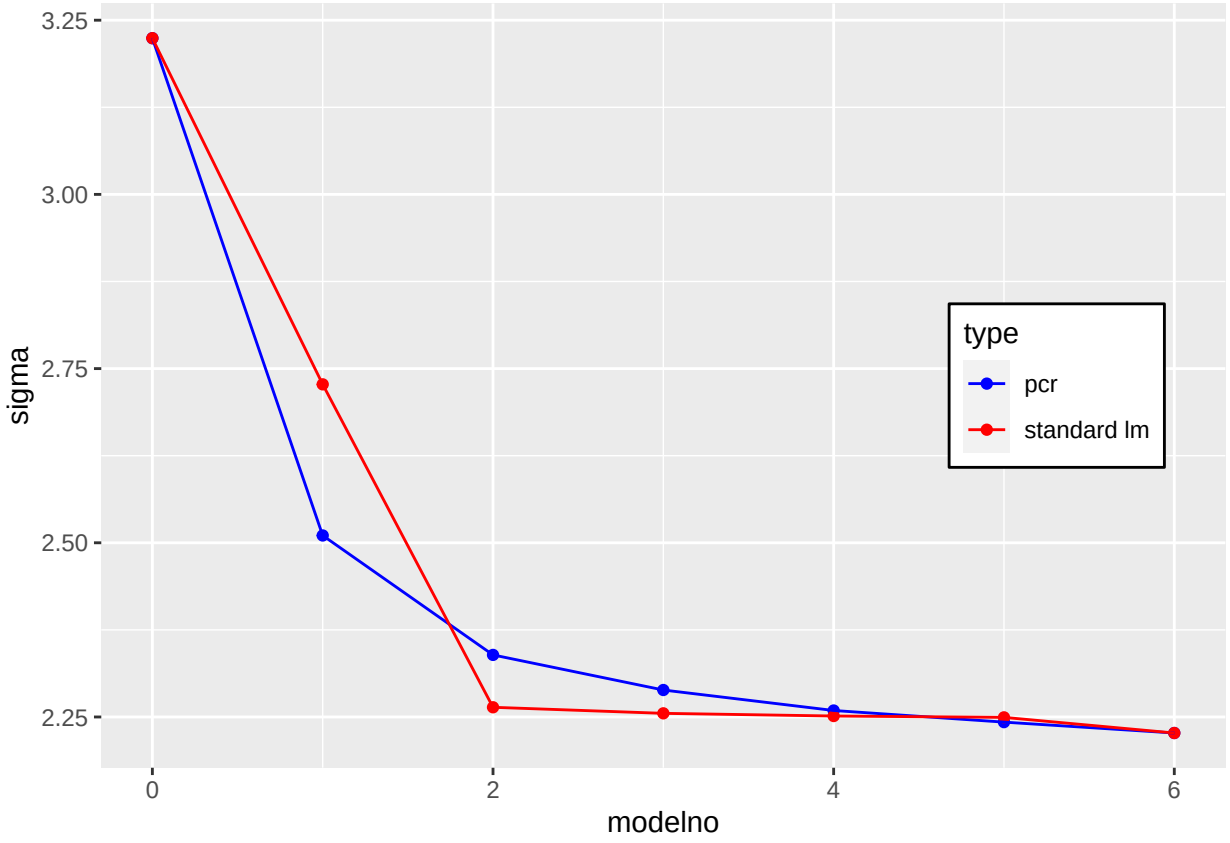
(b) Carry out PCR on the raw data using the same variables and response as in part (a). For each additional principal component you add to the regression model as predictor, calculate the residual standard deviation and list which of the variables has the highest absolute weight in the respective principal component.





	modelno	variable	sigma
(Intercept)	0		3.224169
1 comps	1	Whole_weight	2.727390
2 comps	2	Shucked_weight	2.263911
3 comps	3	Length	2.255230
4 comps	4	Viscera_weight	2.251380
5 comps	5	Height	2.249440
6 comps	6	Viscera_weight	2.227005

(c) In a single graph show plots of residual standard deviation resulting from your models on the y -axis against the number of variables/PC components on the x-axis.



(d) Explain why you do not require to a variable selection method when selecting the predictors in PCR.

PCR does not require us to choose which predictor variables to add to the model since each principal component uses a linear combination of all the predictor variables. Moreover, the principal components are arranged in descending order of variance, so the first principal component always has the largest variance, followed by the second and so on. Hence, a variable selection method is not required as the principal components can just be added to the model in order of their variances.

(e) Comment on your findings and in particular on what approaches work better for these data and why.

Based on the correlation heatmap of the predictor variables, we can see that they are highly collinear. The lines plot above shows that the standard linear regression has a smaller residual standard deviation than the principal component regression when the models with only one variable, but for models with two to four variables, the principal component regression has smaller residual standard deviations than the standard linear regression. As for models with five or more variables, the differences in the residual standard deviations between the two regressions are either small or zero.

For the standard linear regression, the predictor variables are added to the model using forward stepwise regression with the largest F-value. Stepwise regression might not be suitable as the F-statistics do not have the claimed distribution, and collinearity problems are exacerbated. For the principal component regression,

the predictor variables are added to the model based on the variances of the principal components. PCR can perform well even when the predictor variables are highly collinear, as it produces principal components that are orthogonal to each other.

In conclusion, for this data set, standard linear regression should be used for one variable, and principal component regression should be used for two or more variables.

Question 3

We consider the 13-dimensional wine recognition data of Example 4.6 and Lab 4. The data are available in the Data Sets folder. Here we want to compare a factor analysis of all observations with those obtained from cultivar 1 and cultivar 2. The cultivar membership of the observations is given in column 1 of the data set. For part of this analysis you may report the relevant results obtained in the lab. You may find it useful to create two data frames: one for the complete data and a separate one for the first two cultivars of the data. We refer to the latter as the *cultivar12* data. Hint. use the R command factanal from the stats library.

```
cultivar <- read.table(file = "wine.tsv", sep = ",")

cultivar12 <- cultivar[cultivar$V1 != 3, 2:14]
cultivar12

cultivar <- cultivar[, 2:14]
cultivar
```

(a) Scale the data and work with the scaled data. How many observations are in the cultivar12 data?

```
cultivar_scaled <- scale(cultivar, center = TRUE)

cultivar12_scaled <- scale(cultivar12, center = TRUE)

dim(cultivar12_scaled)
```

There are 130 observations in the cultivar12 data.

(b) Separately for the complete and for the cultivar12 data, carry out, display and report the results of the following:

i. Calculate the sample covariance matrix of the scaled data and the eigenvalues of this matrix. What is the value of $\hat{\sigma}^2$ for k=2? How different are the values of $\hat{\sigma}^2$ for the complete and the two cultivar12 datasets? Hint. You may use the information Box 6.7 in your calculations.

```r
S1 <- cov(cultivar_scaled)
val1 <- eigen(S1)$values

S2 <- cov(cultivar12_scaled)
val2 <- eigen(S2)$values

sigma_hat_sq1 <- (1 / (13 - 2)) * (sum(val1[3:13]))
sigma_hat_sq2 <- (1 / (13 - 2)) * (sum(val2[3:13]))

sigma_hat_sq1
```

```
## [1] 0.527016
```

```r
sigma_hat_sq2
```

```
## [1] 0.5849134
```

The value of $\hat{\sigma}^2$ for the complete dataset is 0.57897 lower that the one for cultivar12 dataset.

ii. Calculate and list the factor loadings for the 2-factor principal axis factoring using the value of $\hat{\sigma}^2$ calculated in the previous part.

```r
# Factor loading for complete cultivar dataset
Om1 <- diag(rep(sigma_hat_sq1, 13))
S_A1 <- S1 - Om1

eig_A1 <- eigen(S_A1)

Gamma_hat_1 <- eig_A1$vectors[, 1:2]
Lambda_hat_1 <- diag(eig_A1$values[1:2]^(1 / 2))
Ahat1 <- Gamma_hat_1 %*% Lambda_hat_1
Ahat1
```

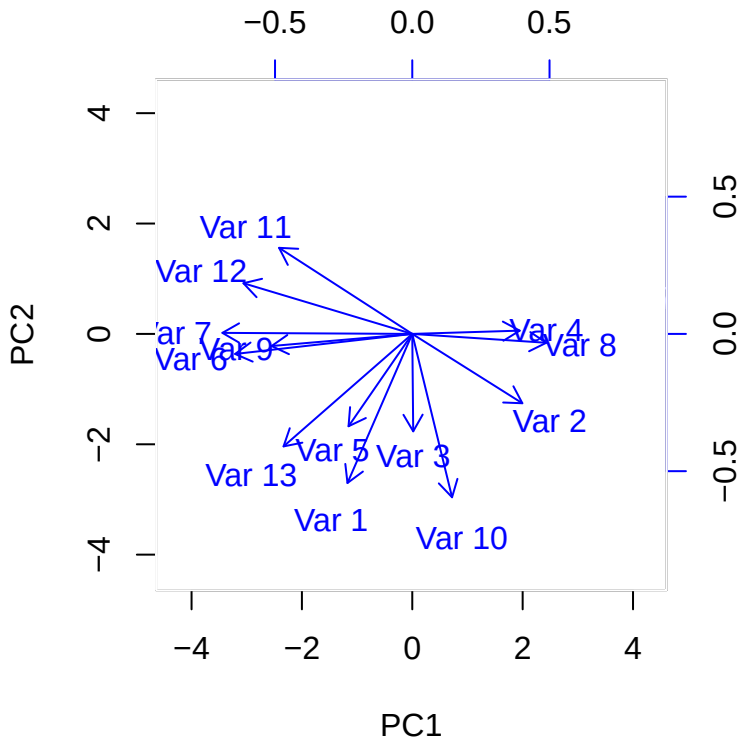
```
##              [,1]        [,2]
##  [1,] -0.29504100 -0.67883001
##  [2,]  0.50121729 -0.31570222
##  [3,]  0.00419282 -0.44361896
##  [4,]  0.48922349  0.01486432
##  [5,] -0.29026293 -0.42055185
##  [6,] -0.80677348 -0.09128633
##  [7,] -0.86457063  0.00471567
##  [8,]  0.61026726 -0.04039350
##  [9,] -0.64071874 -0.05516200
## [10,]  0.18115202 -0.74387639
## [11,] -0.60654976  0.39192100
## [12,] -0.76896884  0.23087893
## [13,] -0.58618456 -0.51216004
```

```r
# Factor loading for cultivar12 datasets
Om2 <- diag(rep(sigma_hat_sq2, 13))
S_A2 <- S2 - Om2

eig_A2 <- eigen(S_A2)

Gamma_hat_2 <- eig_A2$vectors[, 1:2]
Lambda_hat_2 <- diag(eig_A2$values[1:2]^(1 / 2))
Ahat2 <- Gamma_hat_2 %*% Lambda_hat_2
Ahat2
```

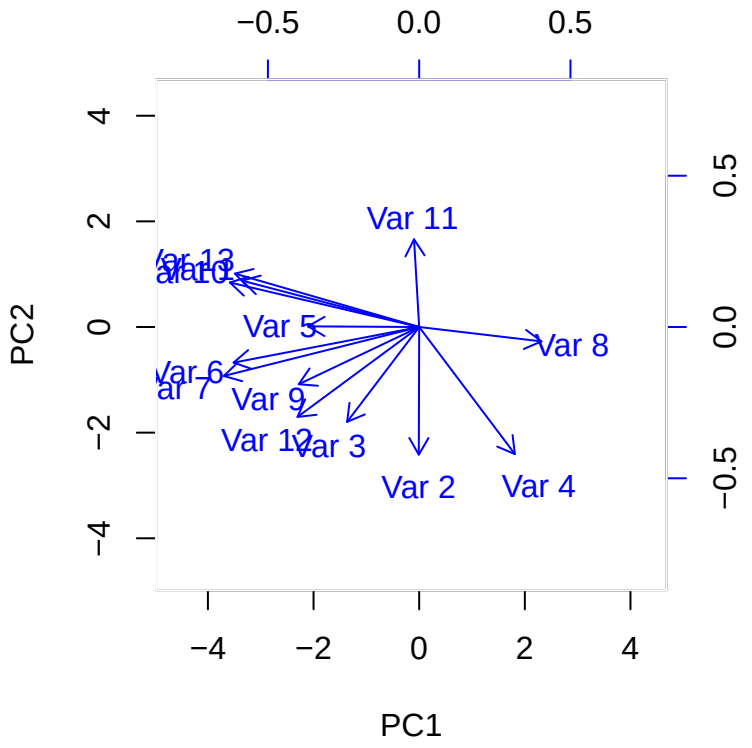
```
##               [,1]         [,2]
##  [1,] -0.731306139  0.191368509
##  [2,] -0.001599091 -0.527327970
##  [3,] -0.297809477 -0.391927572
##  [4,]  0.395377590 -0.525165136
##  [5,] -0.459622750  0.002654609
##  [6,] -0.766785883 -0.147231579
##  [7,] -0.807245450 -0.202261191
##  [8,]  0.505090569 -0.059404908
##  [9,] -0.497678257 -0.236450111
## [10,] -0.782286987  0.183898642
## [11,] -0.021329329  0.362116482
## [12,] -0.502888280 -0.371551565
## [13,] -0.761294450  0.220492655
```

iii. Show biplots of the factor loadings.

```r
wine_pr1 <- prcomp(cultivar_scaled, scale = F)
wine_pr2 <- prcomp(cultivar12_scaled, scale = F)

biplot(wine_pr1$x, Ahat1, col = c("white", "blue"))
```




```
biplot(wine_pr2$x, Ahat2, col = c("white", "blue"))
```



iv. Compare the results obtained from the complete data and the cultivar12 data and comment on the main differences, similarities etc.

The eigenvalues of complete data and cultivar12 data are very similar. Hence, the values of $\hat{\sigma}^2$ for both data are very similar as $\hat{\sigma}^2$ is calculated based on the eigenvalues. The eigenvectors and the factor loadings for the complete data and cultivar12 data are quite different. The factor loadings for both dataset differ in terms of absolute value, relative order by size and the sign. This difference can also be seen in the biplots where the variables are grouped differently and have different angles.

(c) We next turn to ML factor loadings and testing. In your calculations use the option "none" for rotation. If you use other commands, you may not achieve full marks for this question. Separately for the complete and for the cultivar12 data, carry out, display and report the results of the following:

i. Calculate the factor loadings for the 2-factor ML without rotation. List your factor loadings and show biplots of the factor loadings.

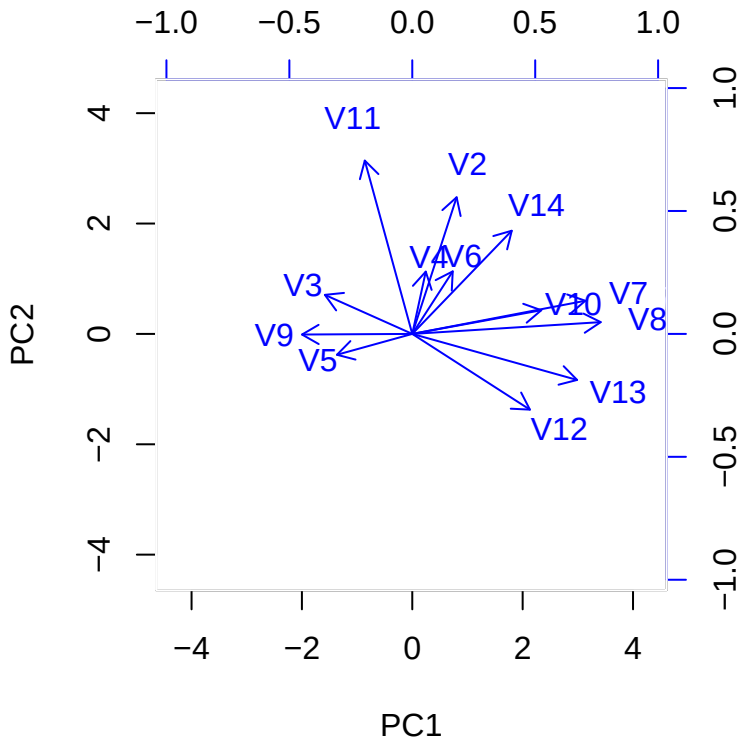
```
fa1 <- factanal(cultivar_scaled, factors = 2, rotation = "none")
fa2 <- factanal(cultivar12_scaled, factors = 2, rotation = "none")

fa1$loadings
```

```
##
## Loadings:
##     Factor1 Factor2
## V2    0.225   0.695
## V3   -0.444   0.198
## V4            0.317
## V5   -0.383  -0.106
## V6    0.206   0.318
## V7    0.880   0.168
## V8    0.958
## V9   -0.561
## V10   0.656   0.120
## V11 -0.242   0.881
## V12   0.598  -0.385
## V13   0.838  -0.234
## V14   0.506   0.525
##
##                Factor1 Factor2
## SS loadings      4.253   2.036
## Proportion Var   0.327   0.157
## Cumulative Var   0.327   0.484
```

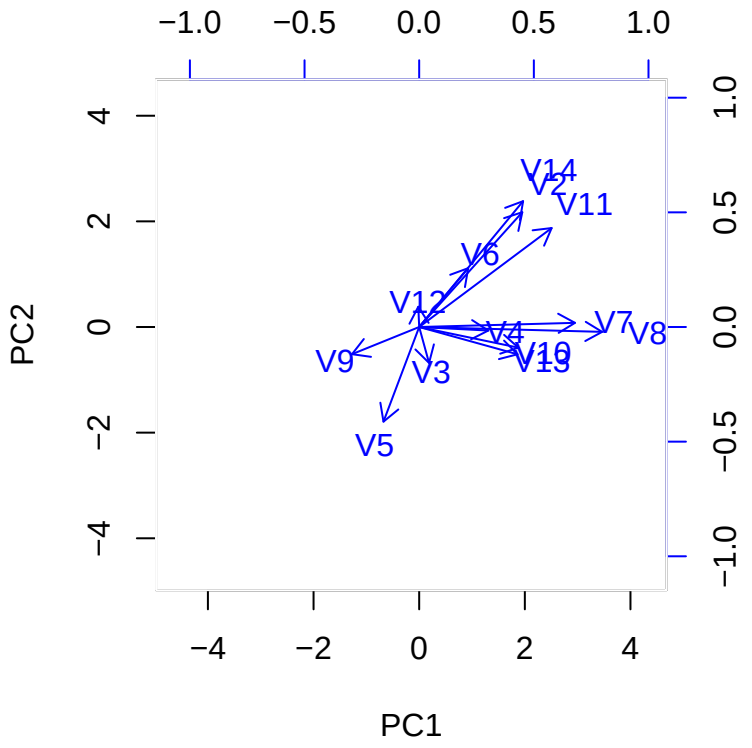
```
fa2$loadings
```

```
##
## Loadings:
##     Factor1 Factor2
## V2    0.562   0.626
## V3           -0.196
## V4    0.376
## V5   -0.195  -0.516
## V6    0.267   0.321
## V7    0.849
## V8    0.997
## V9   -0.366  -0.148
## V10   0.541  -0.113
## V11   0.722   0.539
## V12           0.111
## V13   0.536  -0.149
## V14   0.566   0.686
##
##                Factor1 Factor2
## SS loadings      3.842   1.631
## Proportion Var   0.296   0.125
## Cumulative Var   0.296   0.421
```

```
biplot(wine_pr1$x, fa1$loadings, col = c("white", "blue"))
```




```
biplot(wine_pr2$x, fa2$loadings, col = c("white", "blue"))
```



ii. Carry out a sequence of hypothesis tests starting with the one-factor model.

A. What is the largest number $\hat{\sigma}^2$ of factors you can test with these data? Why can we not exceed this number?

B. For each $k \leq k_{max}$, state the number of degrees of freedom of the χ^2 distribution, the limiting distribution of the test statistic $-2logLR_k$, and report the p-value for each set of tests. Complete data

```
##   k dof      p_value
## 1 1  65 1.466033e-80
## 2 2  53 1.485595e-32
## 3 3  42 1.959095e-15
## 4 4  32 1.444642e-05
## 5 5  23 2.056416e-02
## 6 6  15 3.093393e-01
```

Cultivar12 data

```
##   k dof      p_value
## 1 1  65 5.317439e-48
## 2 2  53 7.741033e-21
## 3 3  42 9.242414e-10
## 4 4  32 1.509757e-04
```

```
## 5 5  23 3.069014e-02
## 6 6  15 1.901765e-01
```

C. What is the appropriate k-factor model for the complete and cultivar12 data? At k = 5, the hypothesis tests of the complete data and cultivar12 are smaller than 0.05, which is the significance value, so the suitable model for both data sets is a 6-factor model. A 5-factor model is ideal if the significant level is at 0.01.

iii. Compare the results of parts (b) and (c).

Question 4

Consider the Boston Housing data which are available from

library (MASS) Boston attach (Boston) In Lab 5 we used these data with the 11 variables shown in Table 7.3 of Chapter 7.

```
attach(Boston)
Boston
```

(a) Use the split of the 11 variables as in Q3 of Lab 5. Calculate canonical correlation scores. List the strength of the four correlations and show the four CC score plots corresponding to ($U_{\bullet j}$, $V_{\bullet j}$) for j = 1, . . . , 4.

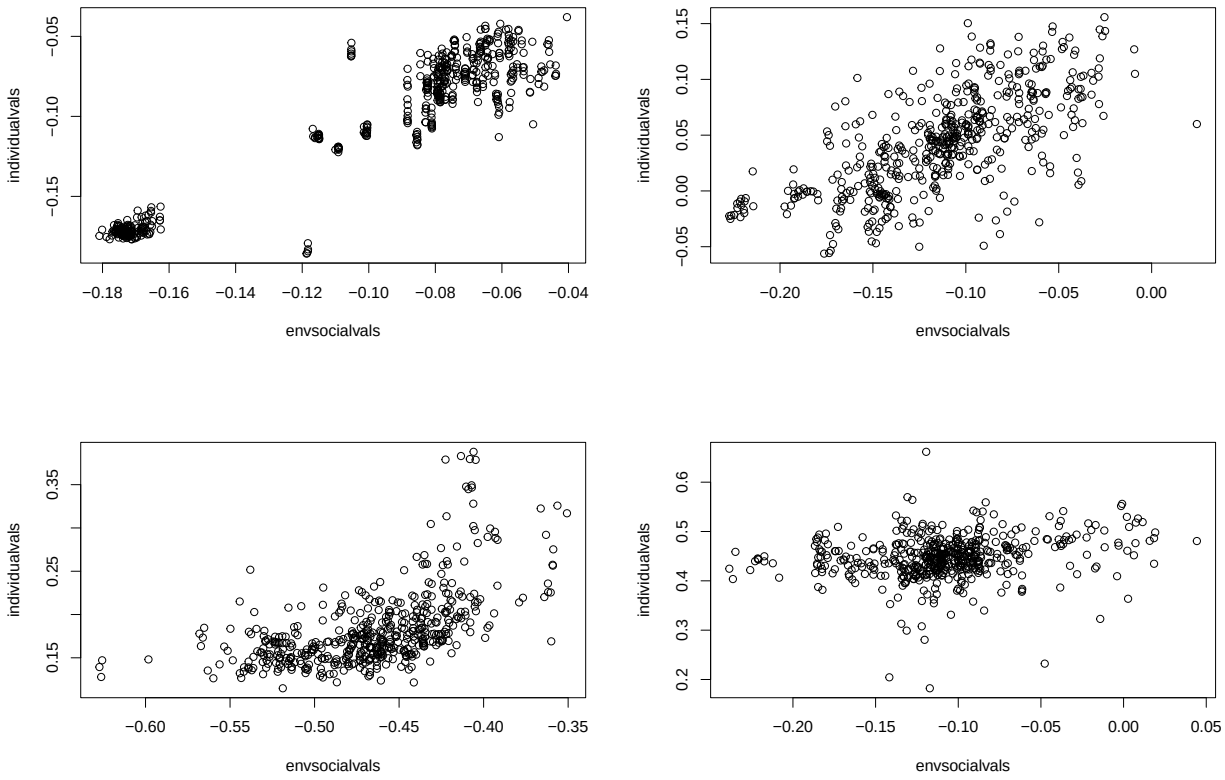
```
Boston.rearranged <- Boston %>% dplyr::select(
  "crim",
  "indus",
  "nox",
  "dis",
  "rad",
  "ptratio",
  "black",
  "rm",
  "age",
  "tax",
  "medv"
)
envsocial <- Boston.rearranged[, 1:7]
individual <- Boston.rearranged[, 8:11]

boston.CC <- cancor(envsocial, individual)

print("The strength of the four correlations: ")
```

```
## [1] "The strength of the four correlations: "
```

```
print(boston.CC$cor)
```

```
## [1] 0.9451239 0.6786623 0.5714338 0.2009740
```

```
for (i in 1:4) {
  envsocialvals <- as.matrix(envsocial) %*% as.matrix(boston.CC$xcoef[, i])

  individualvals <- as.matrix(individual) %*% as.matrix(boston.CC$ycoef[, i])

  plot(envsocialvals, individualvals)
}
```



(b) Comment on the plots and anything unusual you notice.

The first CC score plot has the unusual property that the data splits into two separate clusters.
The second and third CC score plots don't exhibit any interesting behavior.
The fourth CC score plot shows very little correlation.s

(c) Use all variables of the Boston Housing data other than chas and add the extra variables zn and lstat to the previous $X^{[2]}$ data to increase these to 6-dimensional data. Use the $X^{[1]}$ data of part (a). Repeat the calculations and graphics of part (a) for these data.

```
Boston.rearranged <- Boston %>% dplyr::select(
  "crim",
  "indus",
  "nox",
  "dis",
  "rad",
```

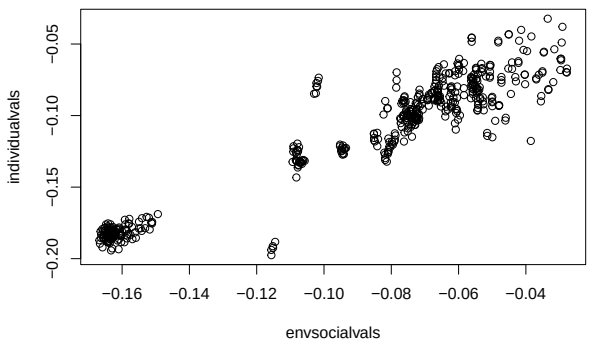
```
  "ptratio",
  "black",
  "rm",
  "age",
  "tax",
  "medv",
  "zn",
  "lstat"
)
envsocial <- Boston.rearranged[, 1:7]
individual <- Boston.rearranged[, 8:13]

boston.CC <- cancor(envsocial, individual)

print("The strength of the four correlations: ")
```

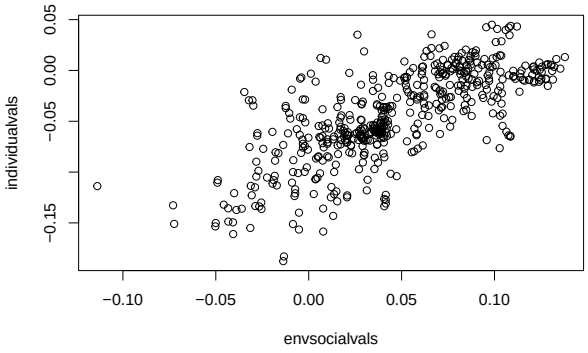
```
## [1] "The strength of the four correlations: "
```

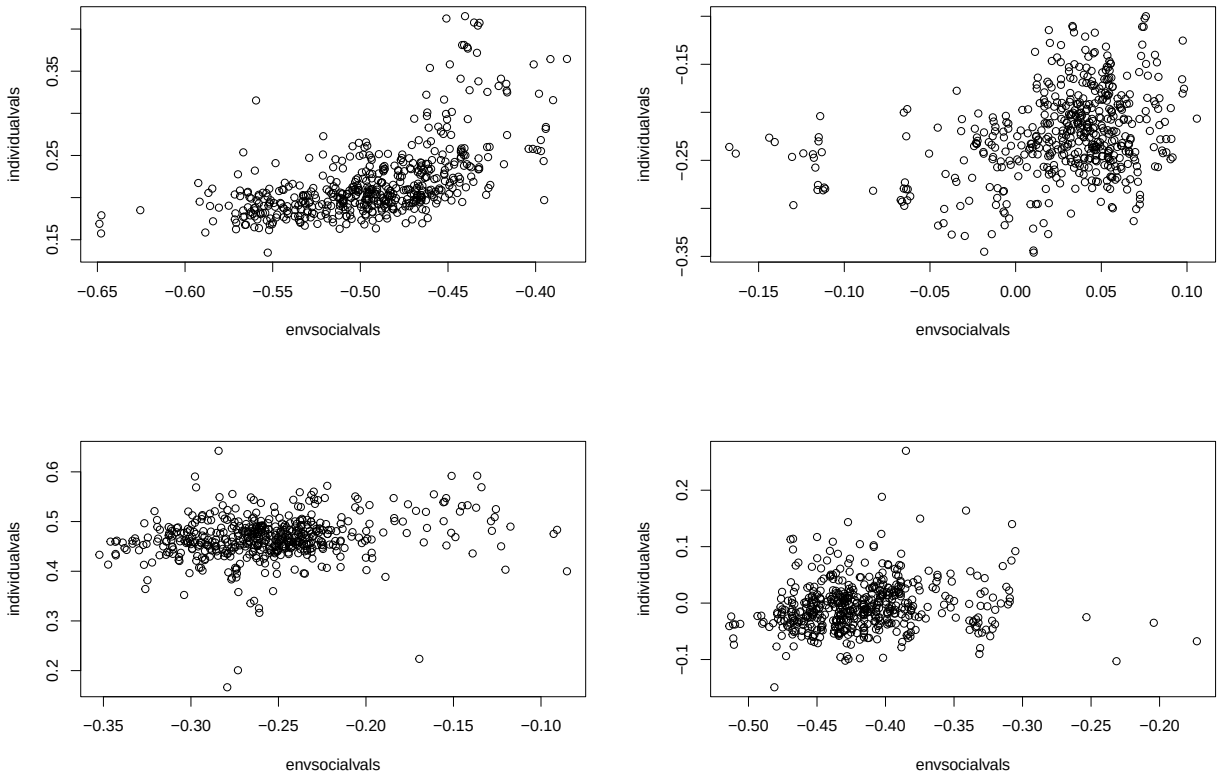
```
print(boston.CC$cor)
```

```
## [1] 0.9537171 0.7344798 0.5778651 0.3259343 0.2076161 0.1257320
```

```
for (i in 1:6) {
  envsocialvals <- as.matrix(envsocial) %*% as.matrix(boston.CC$xcoef[, i])

  individualvals <- as.matrix(individual) %*% as.matrix(boston.CC$ycoef[, i])

  plot(envsocialvals, individualvals)
}
```







(d) Compare the results of parts (a) and (c) and comment on the differences and why they could occur.

Firstly, the correlation scores are uniformly higher in part (c). This is likely because the variance of the additional two variables can be used to find correlations in the X1 data.

There are also now 6 total pairs of CC scores. This is possible because the rank of the data matrix has increased to 6.

(e) Carry out a hypothesis test for the data described in part (c) using the statistic T_k of Lecture 5 and the values of the correlation strengths obtained in part (c). Calculate the p-values for each statistic and report these p-values.

```
Tk <- function(n, d1, d2, cor, k) {
  constant <- n - 0.5 * (d1 + d2 + 3)
  terms <- na.omit(1 - cor[k + 1:length(cor)]^2)
  logterm <- log(prod(terms))
  result <- -constant * logterm
  return(result)
}

n <- 506
d1 <- 7
d2 <- 6
for (k in 1:5) {
  tk <- Tk(n, d1, d2, c(boston.CC$cor), k)
```

```
  df <- (d1 - k) * (d2 - k)
  print(paste0("k=", k, " p-value = ", pchisq(tk, df, lower.tail = FALSE)))
}
```

```
## [1] "k=1 p-value = 1.10813747800098e-122"
## [1] "k=2 p-value = 2.08614517693533e-49"
## [1] "k=3 p-value = 3.18145252539475e-13"
## [1] "k=4 p-value = 4.14651696567399e-05"
## [1] "k=5 p-value = 0.0189154942592372"
```

(f) Using a 1% significance level, make a decision regarding the number of nonzero correlation coefficients of the population model based on your results in part (e).

Looking at the results. At a 1% significance value, we would only retain the null hypothesis for k=5.

The hypothesis test at k=5 is $H_0^5 : v_1 \neq 0, \ldots, v_5 \neq 0, v_6 = 0$ vs $H_1^5 : v_1 \neq 0, \ldots, v_6 \neq 0$. Since we fail to reject the null hypothesis, we assume that the 6th correlation coefficient is zero.

Therefore we conclude that 1,. . . ,5 are nonzero correlation coefficients (thus there are 5).

(g) Does the decision change if you replace the 1% significance level by a 5% significance level? If yes, how? Comment.

Yes, it does change. No decisions would be rejected and therefore we would assume that there are 6 nonzero correlation coefficients.