Homework 3

Zahlen Zbinden

1. Glass samples from crime scenes are analyzed to identify their source. Data was gathered from 214 samples of glass from three different types of sources, each sample is analyzed and five different variables are measured.

The data set has been divided into a training set with 150 observations, and a test set, with 64 observations.

a. Fit a linear descriminant analysis model to the training data, and predict the source for each of the observations in the test data. Report the classification error rate on the test data.

First we need to fit a model and run the predictions on the test data

```
lda_fit <- lda(NewType ~ ., data = glass_train)
lda_predict <- predict(lda_fit, newdata = glass_test)
```

Now we can calculate the error rate on the test data

```
error_rate <- mean(lda_predict$class != glass_test$NewType)
error_rate
```

```
[1] 0.265625
```

b. Fit a quadratic discriminant analysis model to the training data, and predict the source for each of the observations in the test data. Report the classification error rate on the test data.

first we need to fit a model and run the predictions on the test data

```
quad_fit <- qda(NewType ~., data = glass_train)
quad_preds <- predict(quad_fit, newdata = glass_test)
```

Now we can calculate the error rate:

```
error_rate <- mean(quad_preds$class != glass_test$NewType)
error_rate
```

```
[1] 0.40625
```

c. fit a k-nearest neighbors classification model, and report the classification error rate on the test data.

First we need to fit the model

```
knn_preds <- knn(
    train = glass_train[1:5],
    test = glass_test[1:5],
    cl = as.factor(glass_train$NewType),
    k = 5
)
```

Now we can calculate the error rate

```
error_rate <- mean(knn_preds != glass_test$NewType)
error_rate
```

```
[1] 0.25
```

d. Fit a CART classification model, with a minimum node size of 10 to split and a minimum leaf size of Predict the source for each of the observations in the test data and report the classification error rate on the test set.

Fist we need to fit the model

```
cart_fit <- rpart(NewType ~., data = glass_train, method = "class")
cart_preds <- predict(cart_fit, newdata = data.frame(glass_test), type = "class")
```

now we can calculate the error rate

```
error_rate <- mean(cart_preds != glass_test$NewType)
error_rate
```

```
[1] 0.3125
```

e. Which classification approach do you like best for this data and why?

For this data I like the KNN classification approach the best, as it was the easiest to change the hyperparameters for and got the best results (lowest error rate).

2. Data from a study of a comparison of non-diabetic and diabetic patients was obtained for three primary variables, and two secondary variables. The data for n = 64 non-diabetic patients yields a covariance matrix. Determine the canoncial variates and their correlations. Try to interpret these quantities.

First we need to make the matrices

```
s11 <- matrix(
    c(1106, 396.7, 108.4, 396.7, 2382, 1143, 108.4, 1143, 2136),
    ncol = 3
)
s12 <- matrix(
    c(.79, -.21, 2.19, 26.23, -23.96, -20.84),
    ncol = 2
)
s21 <- t(s12)
s22 <- matrix(c(.02, .22, .22, 70.56), ncol = 2)
```

First we need to build the canoncial model, we need to have our two sets of variables X1 and X2

```
X1 = rbind(s11, s21)
X2 = rbind(s12, s22)
```

```
pats <- cc(X1, X2)
```

We can now pull out the canonical variates a1, and b1

```
pats$xcoef
```

```
              [,1]          [,2]
[1,] -0.0008258058  4.261537e-04
[2,]  0.0009284869 -1.055899e-03
[3,] -0.0015073509  9.705469e-05
```

```
pats$ycoef
```

```
             [,1]       [,2]
[1,] -1.007367179 0.36144728
[2,]  0.002745631 0.02615407
```

We can also pull out the correlations

```
pats$cor[1]
```

```
[1] 1
```

The interpretation of the correlation is that of the new eigen space that we can do analysis on. It shows us a linear combination of both sets of variables that has the highest correlation. We can see from the canonical variates that Weight and Insulin Resistance are valued the highest for the first set, and that Insulin Reposnse to oral glucose and relative weight are the most highly weighted in the next pair.

3. Data on national track records for men from 55 diferent countries are given. For each country, the national record for eight different race distances is recorded. The first column of the datset contains the country name, and the second column of the data set contains the three_letter abbreviations for the county, the next eight columns contain the nation records for each of the eight distances. The first three distances are recorded in seconds and the remaining five are recorded in minutes.

a. Find the sample covariance matrix S and the sample correlation matrix R for the distance records. Which of these matrices would you find more interesting/appropriate to use for a principal componenent analysis and why?

When using scaled data for all of the running times, both the covariance matrix and the correlation matrix are the same, so it doesn't make a difference which one we choose to use.

First lets scale the data so that larger numbers are going to influence our calculation, especially since some of the run data is in Min and some is in Sec

```
run[3:9] <- scale(run[3:9])
```

```
S <- cov(run[3:9])
R <- cor(run[3:9])
```

b. Find the eigenvalues and egenvectors of S

```
S_eig <- eigen(S)
S_eig
```

```
eigen() decomposition
$values
[1] 5.88027742 0.70208467 0.15185179 0.11501149 0.07173128 0.05483367 0.02420968

$vectors
           [,1]        [,2]        [,3]       [,4]        [,5]        [,6]
[1,] -0.3499509 -0.58034794  0.25802997 -0.1375930  0.61565817  0.26077917
[2,] -0.3681936 -0.45323211  0.37952293  0.2149162 -0.59843281 -0.31596306
[3,] -0.3831991 -0.19989270 -0.68549393 -0.5374357 -0.23289150 -0.01492540
[4,] -0.3918442  0.07951855 -0.43053980  0.6686476  0.32132768 -0.32269430
[5,] -0.3926692  0.25356899  0.01436636  0.2665510 -0.27909253  0.78940392
[6,] -0.3779614  0.42232534  0.27529735 -0.2485874  0.04563437 -0.30953206
[7,] -0.3801849  0.41090230  0.24108381 -0.2571135  0.15930510 -0.09401352
             [,7]
[1,] -0.090435322
[2,]  0.104081953
[3,] -0.001063656
[4,]  0.017255005
[5,] -0.095951311
[6,] -0.665814973
[7,]  0.726755324
```

c. Find the eigenvalues and eigenvectors of R

```
R_eig <- eigen(S)
R_eig
```

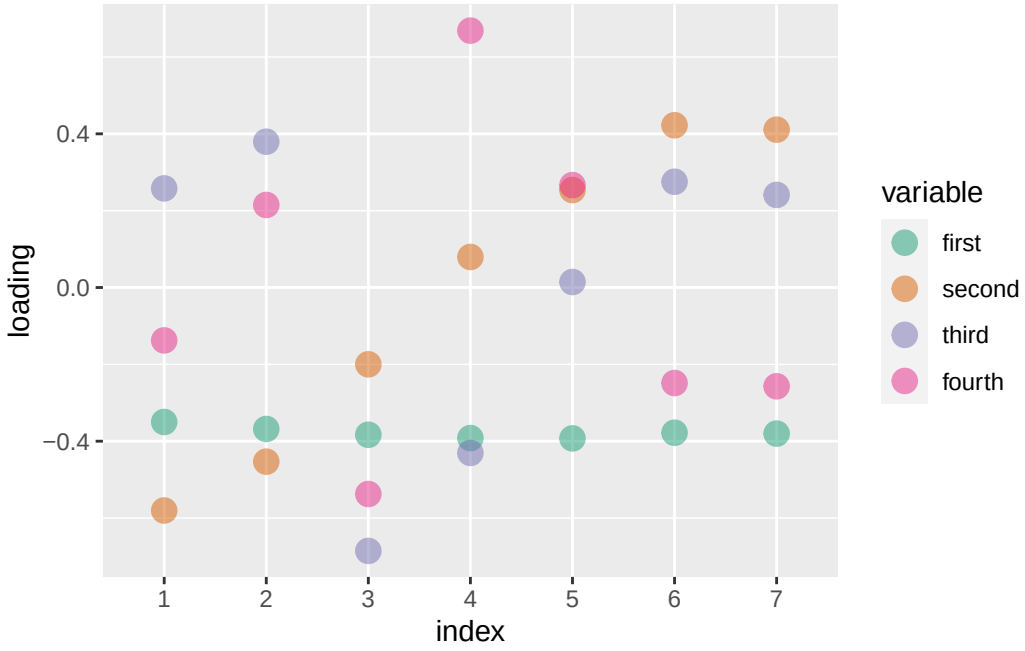
```
eigen() decomposition
$values
[1] 5.88027742 0.70208467 0.15185179 0.11501149 0.07173128 0.05483367 0.02420968

$vectors
           [,1]        [,2]        [,3]       [,4]        [,5]        [,6]
[1,] -0.3499509 -0.58034794  0.25802997 -0.1375930  0.61565817  0.26077917
[2,] -0.3681936 -0.45323211  0.37952293  0.2149162 -0.59843281 -0.31596306
[3,] -0.3831991 -0.19989270 -0.68549393 -0.5374357 -0.23289150 -0.01492540
[4,] -0.3918442  0.07951855 -0.43053980  0.6686476  0.32132768 -0.32269430
[5,] -0.3926692  0.25356899  0.01436636  0.2665510 -0.27909253  0.78940392
[6,] -0.3779614  0.42232534  0.27529735 -0.2485874  0.04563437 -0.30953206
[7,] -0.3801849  0.41090230  0.24108381 -0.2571135  0.15930510 -0.09401352
             [,7]
[1,] -0.090435322
```

```
[2,]  0.104081953
[3,] -0.001063656
[4,]  0.017255005
[5,] -0.095951311
[6,] -0.665814973
[7,]  0.726755324
```

d. Construct plots of the loadings(coefficient vectors) for the first found principal componenets computed using cample covariance matrix S

```r
df_load <- data.frame(
    index = c("1", "2", "3", "4", "5", "6", "7"),
    first = S_eig$vectors[,1],
    second = S_eig$vectors[,2],
    third = S_eig$vectors[,3],
    fourth = S_eig$vectors[,4]
)
```

```r
df_load %>%
    melt(id.vars = "index", variable.name = "variable", value.name = "loading") %>%
    ggplot(aes(x = index, y = loading, color = variable)) +
        geom_point(size = 4, alpha = .5) +
        scale_color_brewer(type = "qual", palette = 2)
```



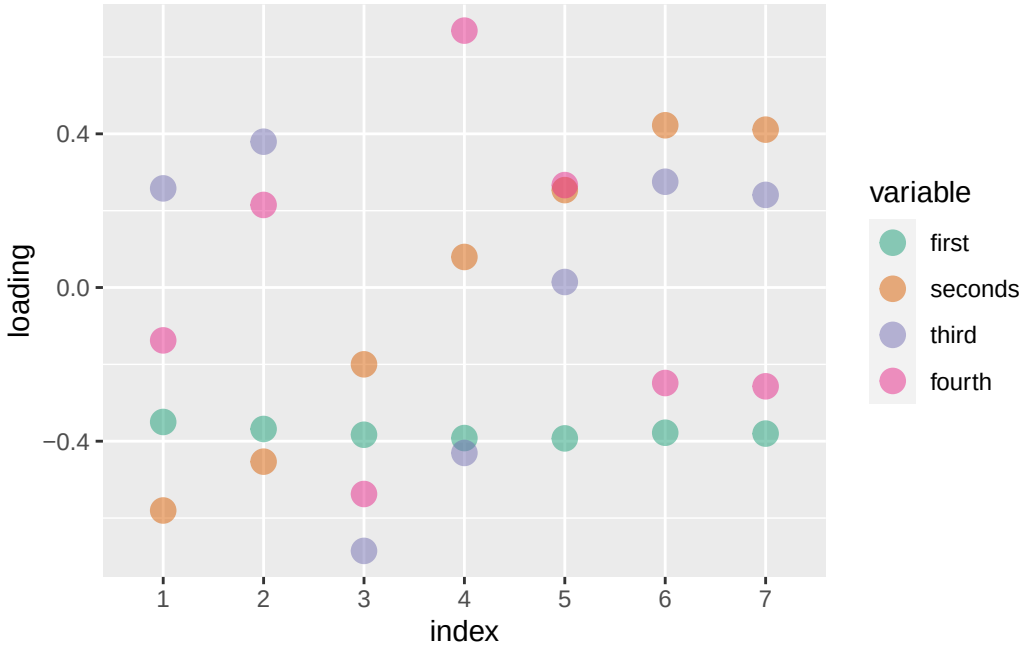
e. How would you interpret the loadings for the first principal component found using S?

It weights all of the variables roughly the same to determine the vector that contains most of the spread of the multivariate distribution.

f. Construct plots of the loadings for the first four principal components computed using the sample correlation matrix R.

```
df_load <- data.frame(
    index = c("1", "2", "3" , "4" , "5", "6", "7"),
    first = R_eig$vectors[,1],
    seconds = R_eig$vectors[,2],
    third = R_eig$vectors[,3],
    fourth = R_eig$vectors[,4]
)
```

```
df_load %>%
    melt(id.vars = "index", variable.name = "variable", value.name = "loading") %>%
    ggplot(aes(x = index, y = loading, color = variable)) +
    geom_point(size = 4, alpha = .5) +
    scale_color_brewer(type = "qual", palette = 2)
```



g. How would you interpret the loadings for the first principal component found using R?

It weights all of the variables roughly the same to determine the vector that contains most of the spread of the multivariate distribution.

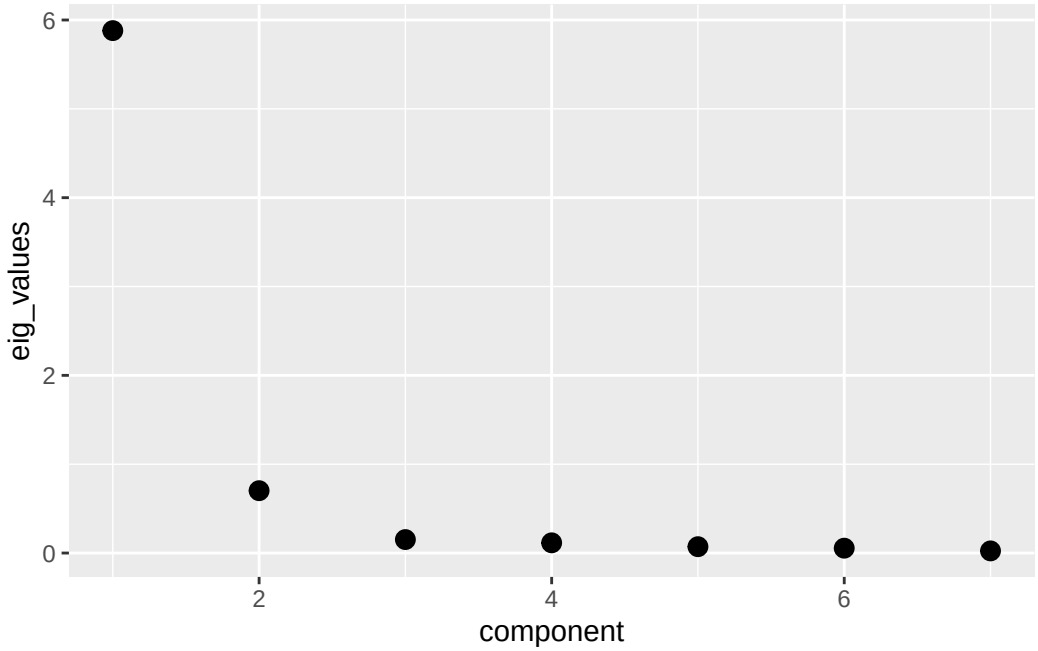
h. How would you interpret the loadings for the second principal component found using S.

The second principal component values the weights of the first 2 variables (100m and 200m) and the wieghts of the last 2 variables (5000, and 10000) much higher than the other variables.

i. Plot the scree plot and cumulative variance plot for the principal components found using R

```
df_scree <- data.frame(
    "eig_values" = R_eig$values,
    "component" = seq(from = 1, to = 7)
)
```

```
ggplot(df_scree, aes(x = component, y = eig_values)) +
    geom_point(size = 3)
```



j. How many principal components or R would you want to keep to explain most of the signal in this data? Explain your choice

I would choose to keep the first two, as the scree plot shows an "elbow" at the third component which means the amount of variance explained by variables 3 and on doesn't increase by much.