Homework 3

Eleana Cabello

09-15-2022

Glass Data

Part a

i. Correlation matrix of the Glass data set.

```
corMat <- cor(x = Glass[, 1:9])
corMat
```

```
##               RI          Na           Mg          Al          Si            K
## RI  1.0000000000 -0.19188538 -0.122274039 -0.40732603 -0.54205220 -0.289832711
## Na -0.1918853790  1.00000000 -0.273731961  0.15679367 -0.06980881 -0.266086504
## Mg -0.1222740393 -0.27373196  1.000000000 -0.48179851 -0.16592672  0.005395667
## Al -0.4073260341  0.15679367 -0.481798509  1.00000000 -0.00552372  0.325958446
## Si -0.5420521997 -0.06980881 -0.165926723 -0.00552372  1.00000000 -0.193330854
## K  -0.2898327111 -0.26608650  0.005395667  0.32595845 -0.19333085  1.000000000
## Ca  0.8104026963 -0.27544249 -0.443750026 -0.25959201 -0.20873215 -0.317836155
## Ba -0.0003860189  0.32660288 -0.492262118  0.47940390 -0.10215131 -0.042618059
## Fe  0.1430096093 -0.24134641  0.083059529 -0.07440215 -0.09420073 -0.007719049
##            Ca            Ba           Fe
## RI  0.8104027 -0.0003860189  0.143009609
## Na -0.2754425  0.3266028795 -0.241346411
## Mg -0.4437500 -0.4922621178  0.083059529
## Al -0.2595920  0.4794039017 -0.074402151
## Si -0.2087322 -0.1021513105 -0.094200731
## K  -0.3178362 -0.0426180594 -0.007719049
## Ca  1.0000000 -0.1128409671  0.124968219
## Ba -0.1128410  1.0000000000 -0.058691755
## Fe  0.1249682 -0.0586917554  1.000000000
```

ii. Eigenvalues and eigenvectors of the data set.

```
eigenVals_Vecs <- eigen(corMat)
eigenVals_Vecs
```

```
## eigen() decomposition
## $values
## [1] 2.511163726 2.050072185 1.404843994 1.157862446 0.914002247 0.527635193
## [7] 0.368958443 0.063852948 0.001608818
##
## $vectors
##             [,1]        [,2]          [,3]        [,4]         [,5]        [,6]
##  [1,]  0.5451766 -0.28568318 -0.0869108293  0.14738099  0.073542700 -0.11528772
##  [2,] -0.2581256 -0.27035007  0.3849196197  0.49124204 -0.153683304  0.55811757
##  [3,]  0.1108810  0.59355826 -0.0084179590  0.37878577 -0.123509124 -0.30818598
```

```
##  [4,] -0.4287086 -0.29521154 -0.3292371183 -0.13750592 -0.014108879  0.01885731
##  [5,] -0.2288364  0.15509891  0.4587088382 -0.65253771 -0.008500117 -0.08609797
##  [6,] -0.2193440  0.15397013 -0.6625741197 -0.03853544  0.307039842  0.24363237
##  [7,]  0.4923061 -0.34537980  0.0009847321 -0.27644322  0.188187742  0.14866937
##  [8,] -0.2503751 -0.48470218 -0.0740547309  0.13317545 -0.251334261 -0.65721884
##  [9,]  0.1858415  0.06203879 -0.2844505524 -0.23049202 -0.873264047  0.24304431
##              [,7]        [,8]        [,9]
##  [1,]  0.08186724  0.75221590  0.02573194
##  [2,]  0.14858006  0.12769315 -0.31193718
##  [3,] -0.20604537  0.07689061 -0.57727335
##  [4,] -0.69923557  0.27444105 -0.19222686
##  [5,]  0.21606658  0.37992298 -0.29807321
##  [6,]  0.50412141  0.10981168 -0.26050863
##  [7,] -0.09913463 -0.39870468 -0.57932321
##  [8,]  0.35178255 -0.14493235 -0.19822820
##  [9,]  0.07372136  0.01627141 -0.01466944
```

iii. Principle component analysis of the Glass data set.

```
prinComp <- prcomp(Glass[, 1:9], scale. = TRUE)
prinComp
```

```
## Standard deviations (1, .., p=9):
## [1] 1.58466518 1.43180731 1.18526115 1.07604017 0.95603465 0.72638502 0.60741950
## [8] 0.25269141 0.04011007
##
## Rotation (n x k) = (9 x 9):
##           PC1         PC2           PC3         PC4          PC5         PC6
## RI -0.5451766  0.28568318 -0.0869108293 -0.14738099  0.073542700 -0.11528772
## Na  0.2581256  0.27035007  0.3849196197 -0.49124204 -0.153683304  0.55811757
## Mg -0.1108810 -0.59355826 -0.0084179590 -0.37878577 -0.123509124 -0.30818598
## Al  0.4287086  0.29521154 -0.3292371183  0.13750592 -0.014108879  0.01885731
## Si  0.2288364 -0.15509891  0.4587088382  0.65253771 -0.008500117 -0.08609797
## K   0.2193440 -0.15397013 -0.6625741197  0.03853544  0.307039842  0.24363237
## Ca -0.4923061  0.34537980  0.0009847321  0.27644322  0.188187742  0.14866937
## Ba  0.2503751  0.48470218 -0.0740547309 -0.13317545 -0.251334261 -0.65721884
## Fe -0.1858415 -0.06203879 -0.2844505524  0.23049202 -0.873264047  0.24304431
##            PC7         PC8         PC9
## RI -0.08186724 -0.75221590 -0.02573194
## Na -0.14858006 -0.12769315  0.31193718
## Mg  0.20604537 -0.07689061  0.57727335
## Al  0.69923557 -0.27444105  0.19222686
## Si -0.21606658 -0.37992298  0.29807321
## K  -0.50412141 -0.10981168  0.26050863
## Ca  0.09913463  0.39870468  0.57932321
## Ba -0.35178255  0.14493235  0.19822820
## Fe -0.07372136 -0.01627141  0.01466944
```

iv. The eigenvectors and principal components seem to be equal in magnitude but different in their direction. However, based on our groups personal learning these vectors should be the same.

v. Proving components 1 and 2 are orthogonal.

```
dotProd <- dot(prinComp$rotation[, 1], prinComp$rotation[, 2])

dotProd
```

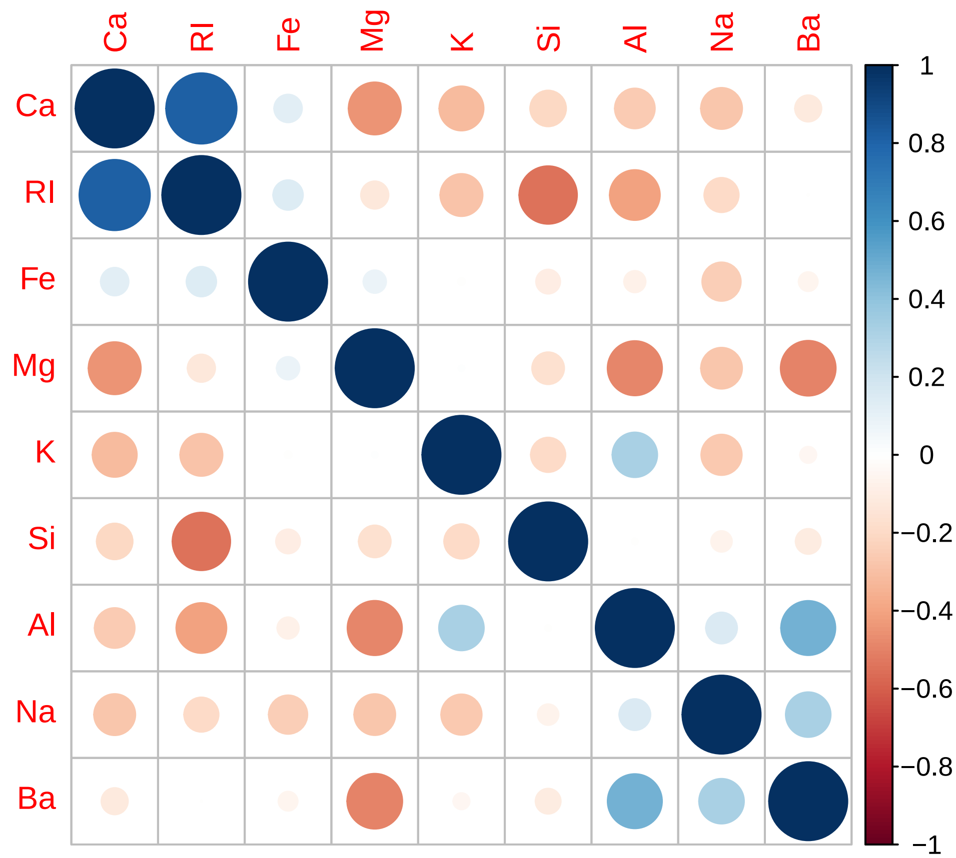
```
## [1] 3.469447e-18
```

The dot product of the two principle components is essentially zero proving they are orthogonal.

Part b

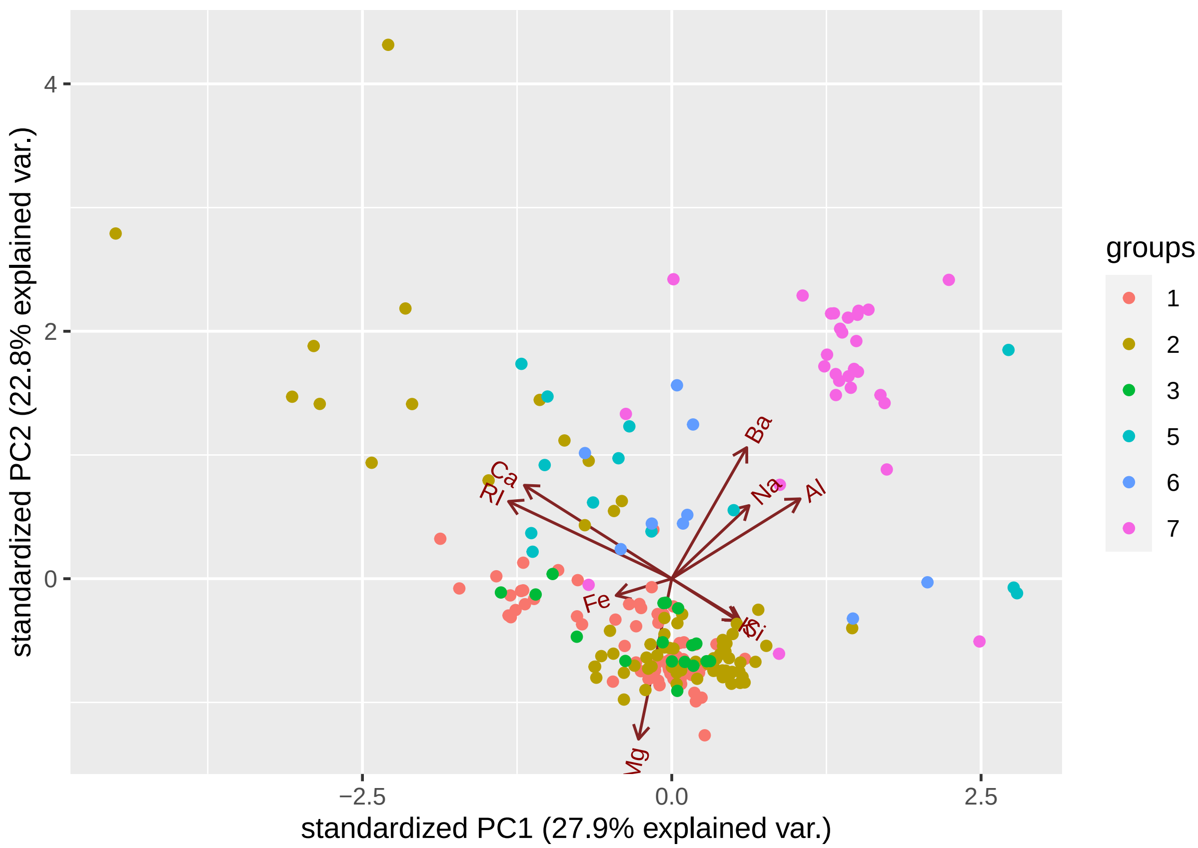
i. Correlation plot of the correlation matrix.

```r
corrplot(corMat, method = "circle", order = "AOE")
```



ii. Biplot of components 1 and 2.

```r
# Visual representation of principle components 1 vs 2.
ggbiplot(prinComp, choices = 1:2, groups = Glass$Type)
```



iii. Together principle components 1 and 2 explain about 50.70% of the variance in the original data. Component 1 has notable negative associations with CA and RI. It also has notable positive associations with Ba, Na, and Al. Component 2 has a large negative association with Mg. These two components were not able to separate or distinguish glass types very well. There is still severe overlap between some groups.

iv. Based on the results above, we do not believe the data can be reduced with PCA while still preserving the variance in the data. PCA is not the ideal dimension reduction method for this data set. PCA's ineffectiveness in dimension reduction for the glass data set could be due to the fact that PCA does not consider the class label or outliers in the data that were not addressed.

Part c

i. Linear discriminant analysis of the Glass data set.

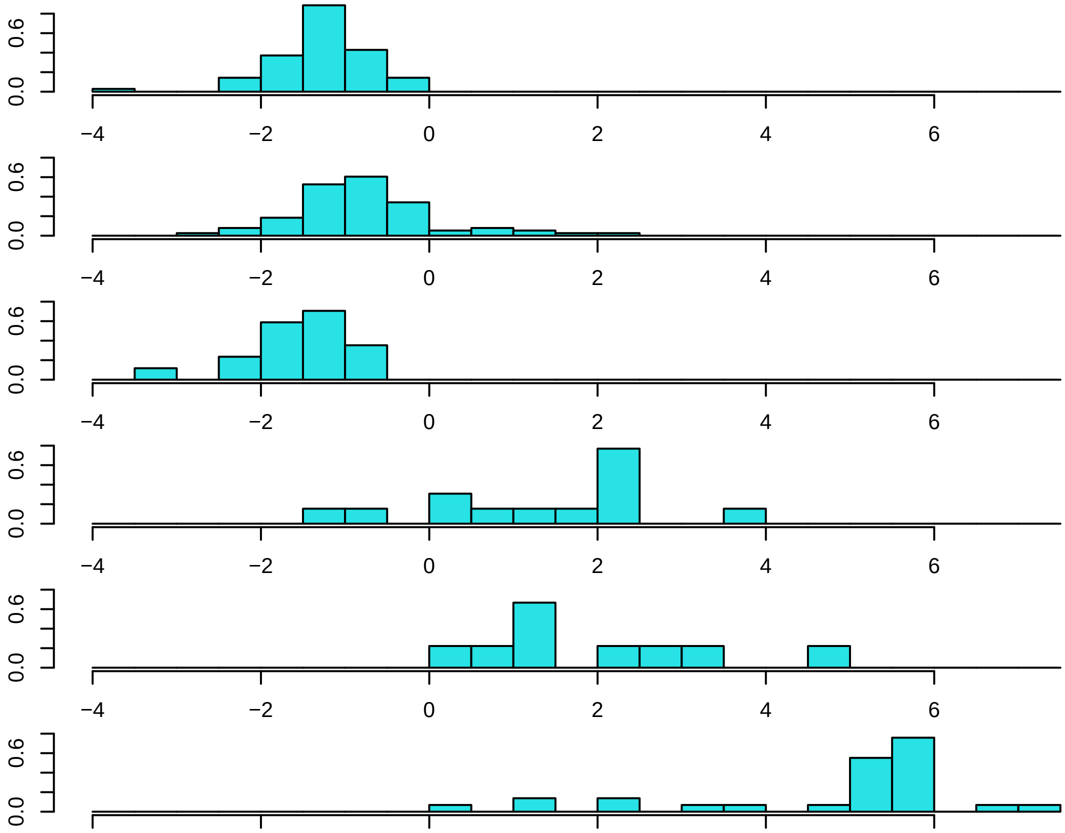
```
ldaAnalysis <- lda(x = Glass[, 1:9], grouping = Glass$Type)
ldaAnalysis
```

```
## Call:
## lda(Glass[, 1:9], grouping = Glass$Type)
##
## Prior probabilities of groups:
##          1          2          3          5          6          7
## 0.32710280 0.35514019 0.07943925 0.06074766 0.04205607 0.13551402
##
## Group means:
##         RI       Na        Mg       Al       Si         K        Ca          Ba
## 1 1.518718 13.24229 3.5524286 1.163857 72.61914 0.4474286  8.797286 0.012714286
```

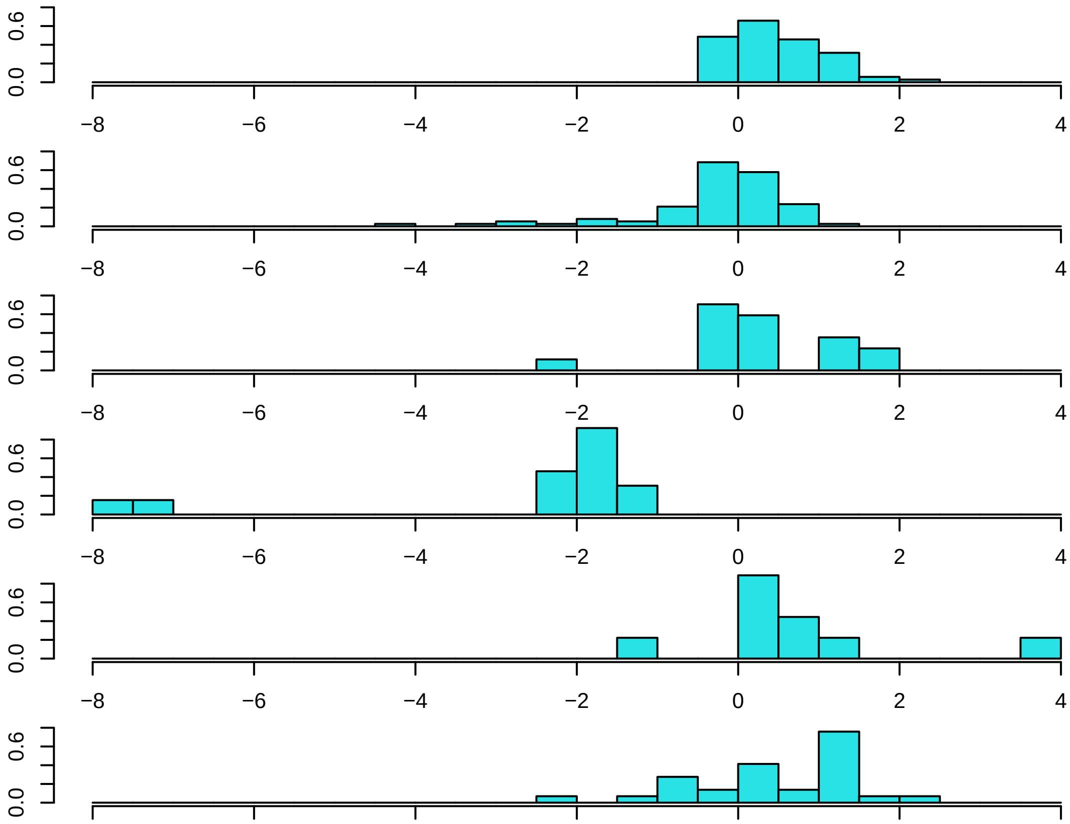
```
## 2 1.518619 13.11171 3.0021053 1.408158 72.59803 0.5210526  9.073684 0.050263158
## 3 1.517964 13.43706 3.5435294 1.201176 72.40471 0.4064706  8.782941 0.008823529
## 5 1.518928 12.82769 0.7738462 2.033846 72.36615 1.4700000 10.123846 0.187692308
## 6 1.517456 14.64667 1.3055556 1.366667 73.20667 0.0000000  9.356667 0.000000000
## 7 1.517116 14.44207 0.5382759 2.122759 72.96586 0.3251724  8.491379 1.040000000
##           Fe
## 1 0.05700000
## 2 0.07973684
## 3 0.05705882
## 5 0.06076923
## 6 0.00000000
## 7 0.01344828
##
## Coefficients of linear discriminants:
##            LD1        LD2         LD3          LD4          LD5
## RI 311.6912516 29.3910394 356.0188308 246.85720802 -804.6553938
## Na   2.3812158  3.1650800   0.4596785   6.92435141    2.3987509
## Mg   0.7403818  2.9858720   1.5728838   6.84983896    2.8002951
## Al   3.3377416  1.7247396   2.2024668   6.41923638    0.9371345
## Si   2.4516520  3.0063507   1.7026191   7.54220302    0.9562989
## K    1.5714954  1.8620159   1.2861127   8.07611300    2.8209927
## Ca   1.0063101  2.3729126   0.6475200   6.69663574    3.7110859
## Ba   2.3140953  3.4431987   2.5964981   6.43849270    4.4077058
## Fe  -0.5114573  0.2166388   1.2026071  -0.04474935   -1.3029207
##
## Proportion of trace:
##    LD1    LD2    LD3    LD4    LD5
## 0.8145 0.1169 0.0413 0.0163 0.0111
```

ii. Linear discriminate 1 was able to achieve 81.45% separation between the types of glass.

```
pred <- predict(ldaAnalysis, Glass[, 1:9])
par(mar = c(1, 4, 1, 4))
ldahist(data = pred$x[, 1], g = Glass$Type)
```




```r
ldahist(data = pred$x[, 2], g = Glass$Type)
```



iii. Based on the histograms above it is easy to see that linear discriminant 1 is able to separate the 6 types of glass effectively. There is small overlap between some of the classes histograms but overall each type falls in range associated with its class. The scatter plot below can also be used to visualize this separation. For discriminant 2, the histograms of the types of glass lie in similar regions with one another making it hard to distinguish them.

Principal Components for Dimension Reduction

Part a

```
data(heptathlon)

# All events were examined using the grubbs test. The following below are the
# results that were significant with a p-value above 0.05.

grubbs.test(heptathlon[, 1])  #hurdles
```

```
##
##  Grubbs test for one outlier
##
## data:  heptathlon[, 1]
## G = 3.5024, U = 0.4676, p-value = 0.000436
## alternative hypothesis: highest value 16.42 is an outlier
```

```
grubbs.test(heptathlon[, 2])  #highjump
```

```
##
##  Grubbs test for one outlier
```

```
##
## data:  heptathlon[, 2]
## G = 3.61806, U = 0.43184, p-value = 0.0001698
## alternative hypothesis: lowest value 1.5 is an outlier
```

```
grubbs.test(heptathlon[, 5])  #longjump
```

```
##
##  Grubbs test for one outlier
##
## data:  heptathlon[, 5]
## G = 2.68319, U = 0.68752, p-value = 0.04594
## alternative hypothesis: lowest value 4.88 is an outlier
```

```
grubbs.test(heptathlon[, 7])  #run800m
```

```
##
##  Grubbs test for one outlier
##
## data:  heptathlon[, 7]
## G = 3.30186, U = 0.52681, p-value = 0.001808
## alternative hypothesis: highest value 163.43 is an outlier
```

```
grubbs.test(heptathlon[, 8])  #Score
```

```
##
##  Grubbs test for one outlier
##
## data:  heptathlon[, 8]
## G = 2.68194, U = 0.68781, p-value = 0.04618
## alternative hypothesis: lowest value 4566 is an outlier
```

```
# Removing Launa (PNG) from data frame.
heptathlon <- heptathlon[!(row.names(heptathlon) %in% c("Launa (PNG)")), ]
```

Part b

```
heptathlon$hurdles = max(heptathlon[, 1]) - heptathlon$hurdles

heptathlon$run200m = max(heptathlon[, 4]) - heptathlon$run200m

heptathlon$run800m = max(heptathlon[, 7]) - heptathlon$run800m
```

Part c

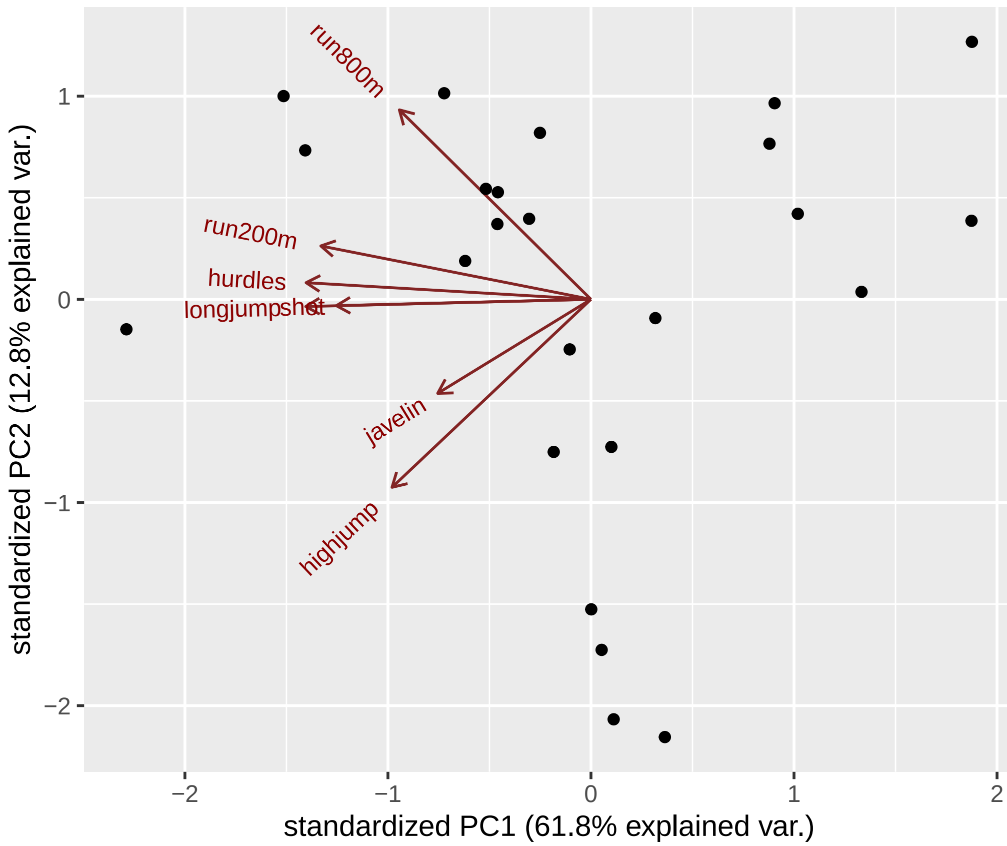
```
Hpca <- prcomp(heptathlon[, 1:7], scale. = TRUE, center = TRUE)
Hpca
```

```
## Standard deviations (1, .., p=7):
## [1] 2.0793370 0.9481532 0.9109016 0.6831967 0.5461888 0.3374549 0.2620420
##
## Rotation (n x k) = (7 x 7):
##                PC1         PC2        PC3         PC4         PC5         PC6
## hurdles  -0.4503876  0.05772161 -0.1739345  0.04840598 -0.19889364  0.84665086
## highjump -0.3145115 -0.65133162 -0.2088272 -0.55694554  0.07076358 -0.09007544
## shot     -0.4024884 -0.02202088 -0.1534709  0.54826705  0.67166466 -0.09886359
```

```
## run200m  -0.4270860  0.18502783  0.1301287  0.23095946 -0.61781764 -0.33279359
## longjump -0.4509639 -0.02492486 -0.2697589 -0.01468275 -0.12151793 -0.38294411
## javelin  -0.2423079 -0.32572229  0.8806995  0.06024757  0.07874396  0.07193437
## run800m  -0.3029068  0.65650503  0.1930020 -0.57418128  0.31880178 -0.05217664
##                  PC7
## hurdles  -0.06961672
## highjump  0.33155910
## shot      0.22904298
## run200m   0.46971934
## longjump -0.74940781
## javelin  -0.21108138
## run800m   0.07718616
```

Part d

```
ggbiplot(Hpca, choices = 1:2)
```

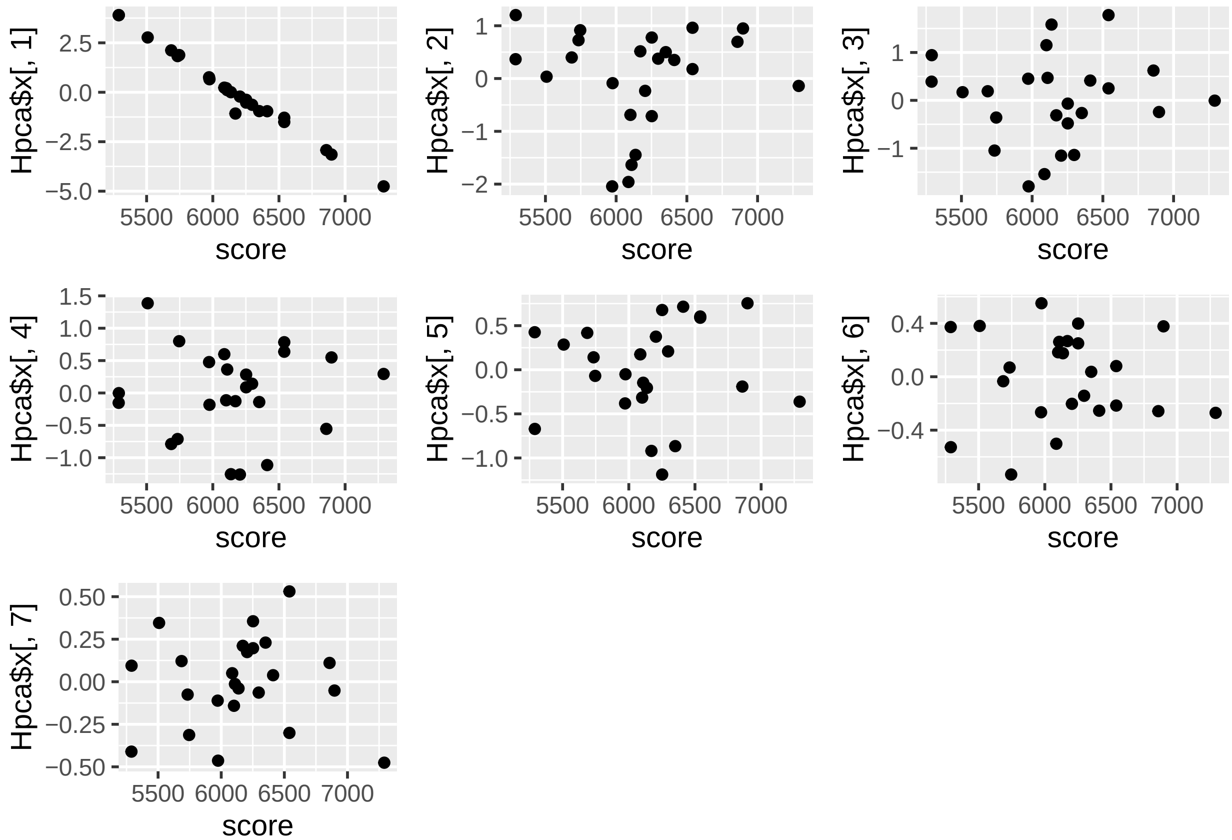


Principal components 1 and 2 shown above are able to explain 80.80% of the variance in the original data. Component 1 has positive associations with the hurdles, run200m, and run800m event results. It also has some large negative associations with the highjump, long jump, and shot event results. Component 2 has a large negative association with the javelin event results.

Part e

```
comp_1 <- ggplot(heptathlon) + geom_point(mapping = aes(x = score, y = Hpca$x[, 1]))
comp_2 <- ggplot(heptathlon) + geom_point(mapping = aes(x = score, y = Hpca$x[, 2]))
comp_3 <- ggplot(heptathlon) + geom_point(mapping = aes(x = score, y = Hpca$x[, 3]))
```

```
comp_4 <- ggplot(heptathlon) + geom_point(mapping = aes(x = score, y = Hpca$x[, 4]))
comp_5 <- ggplot(heptathlon) + geom_point(mapping = aes(x = score, y = Hpca$x[, 5]))
comp_6 <- ggplot(heptathlon) + geom_point(mapping = aes(x = score, y = Hpca$x[, 6]))
comp_7 <- ggplot(heptathlon) + geom_point(mapping = aes(x = score, y = Hpca$x[, 7]))

grid.arrange(comp_1, comp_2, comp_3, comp_4, comp_5, comp_6, comp_7, nrow = 3)
```



Component 1 seems to have the strongest association between it and the overall score of participants. There also seems to be significant negative association between component 2 and the overall score of participants. However, all other components seem to have no clear or significant correlations in their plots.

##Housing data dimension reduction and exploration

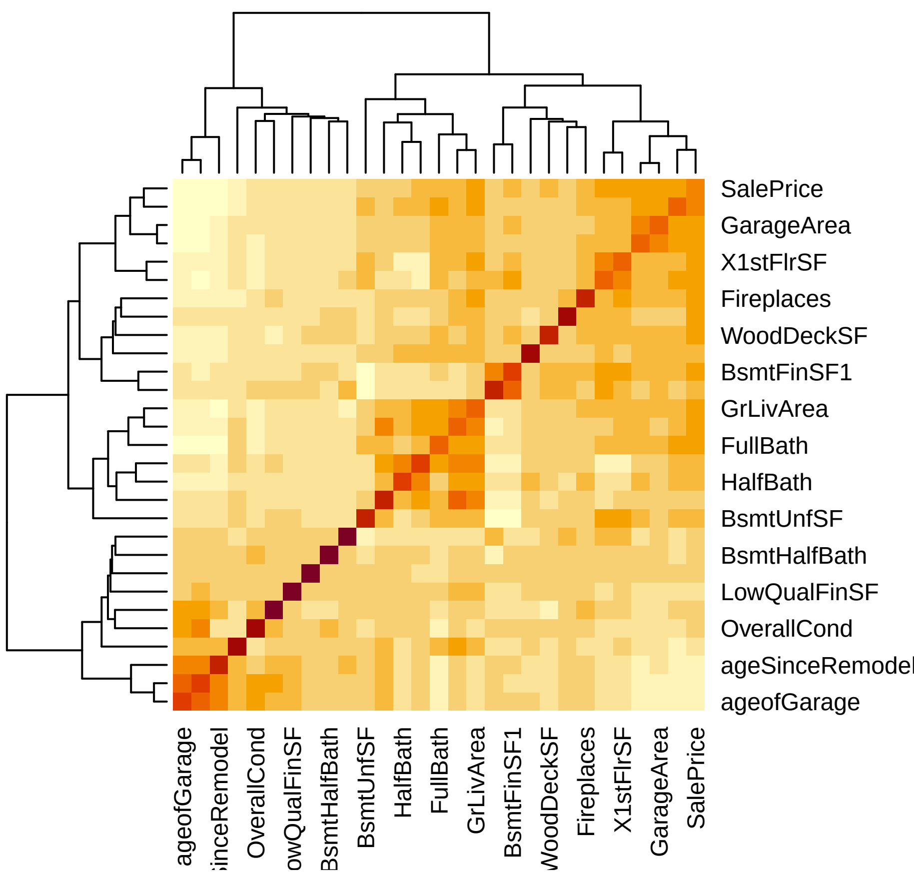
```
library(dplyr)
library(tidyverse)

hd <- read.csv("housingData.csv") %>%
    select_if(is.numeric) %>%
    dplyr::mutate(age = YrSold - YearBuilt, ageSinceRemodel = YrSold - YearRemodAdd,
        ageofGarage = ifelse(is.na(GarageYrBlt), age, YrSold - GarageYrBlt)) %>%
    dplyr::select(!c(Id, MSSubClass, LotFrontage, GarageYrBlt, MiscVal, YrSold, MoSold,
        YearBuilt, YearRemodAdd, MasVnrArea))
```

###PCA of the entire housing data set

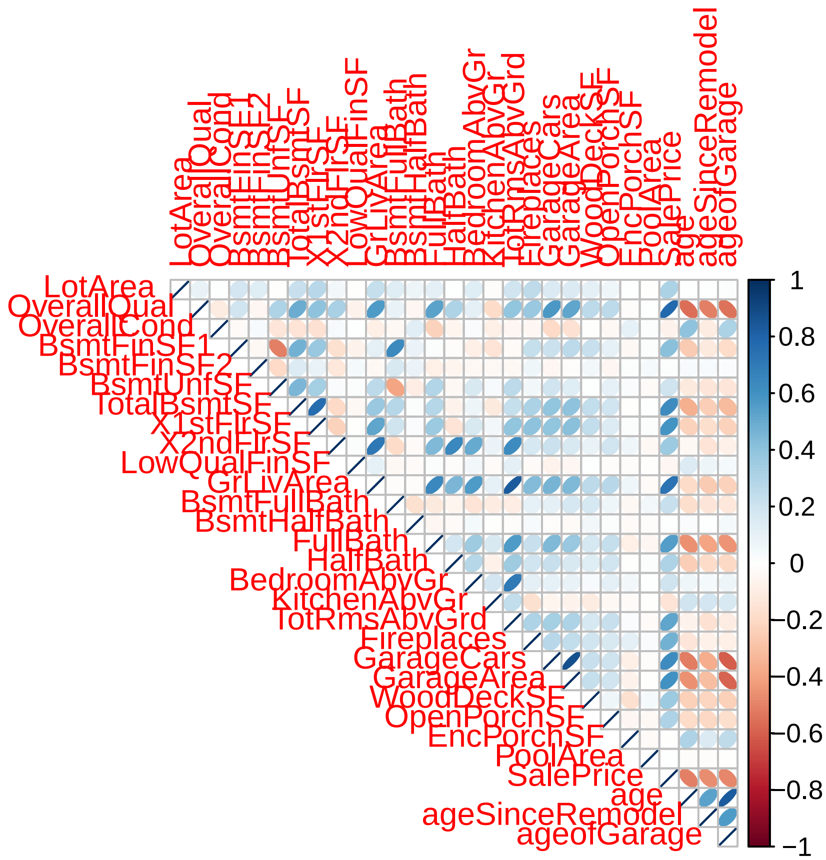
```
# PCA of the entire housing data frame
pc <- prcomp(hd[, ], scale = TRUE)
summary(pc)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.6859 1.8094 1.51612 1.39196 1.17462 1.09640 1.04475
## Proportion of Variance 0.2487 0.1129 0.07926 0.06681 0.04758 0.04145 0.03764
## Cumulative Proportion  0.2487 0.3616 0.44090 0.50772 0.55529 0.59674 0.63438
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     1.02853 1.00509 0.97773 0.96770 0.93850 0.91592 0.86969
## Proportion of Variance 0.03648 0.03483 0.03296 0.03229 0.03037 0.02893 0.02608
## Cumulative Proportion  0.67086 0.70570 0.73866 0.77095 0.80132 0.83025 0.85633
##                           PC15    PC16   PC17   PC18    PC19    PC20    PC21
## Standard deviation     0.83094 0.79780 0.7423 0.6573 0.60629 0.56912 0.52440
## Proportion of Variance 0.02381 0.02195 0.0190 0.0149 0.01268 0.01117 0.00948
## Cumulative Proportion  0.88014 0.90209 0.9211 0.9360 0.94866 0.95983 0.96931
##                          PC22    PC23    PC24    PC25   PC26   PC27      PC28
## Standard deviation     0.4725 0.45274 0.38294 0.36394 0.3186 0.2848 1.193e-15
## Proportion of Variance 0.0077 0.00707 0.00506 0.00457 0.0035 0.0028 0.000e+00
## Cumulative Proportion  0.9770 0.98408 0.98914 0.99370 0.9972 1.0000 1.000e+00
##                             PC29
## Standard deviation     6.918e-16
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

```r
# Correlation matrix for the entire data frame
cMat <- cor(hd[, ])

# Heatmap of the correlation map
heatmap(cor(hd[, ]))
```



```r
# Correlation plot of the data frame
corrplot(cMat, method = "ellipse", type = "upper")
```

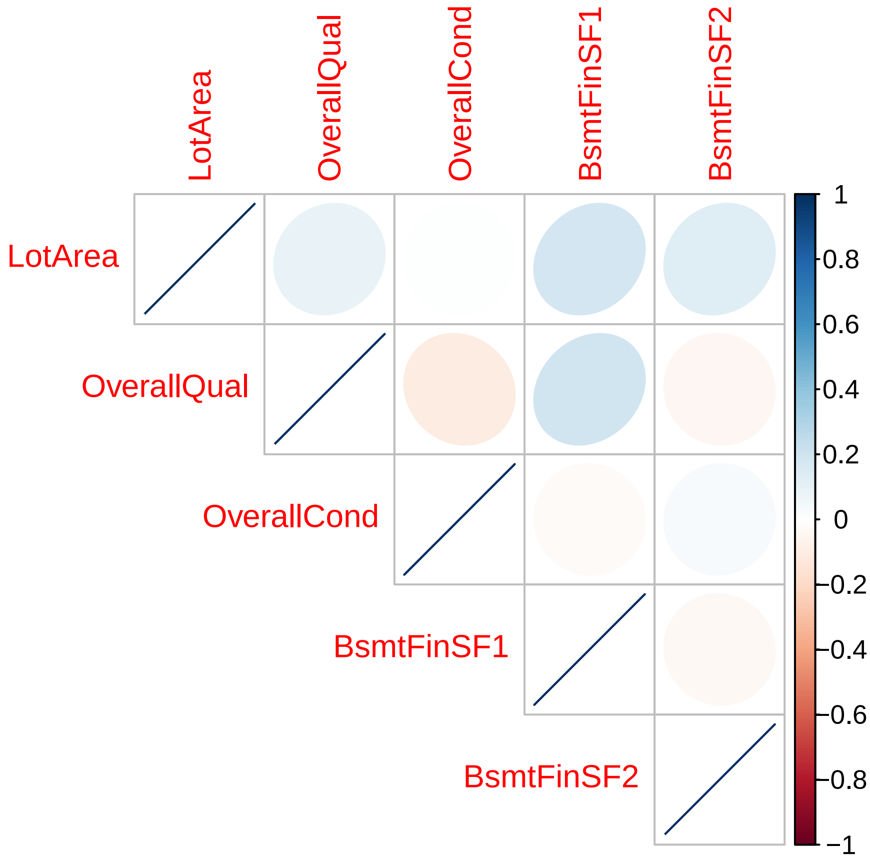


###PCA of a subset of the housing data

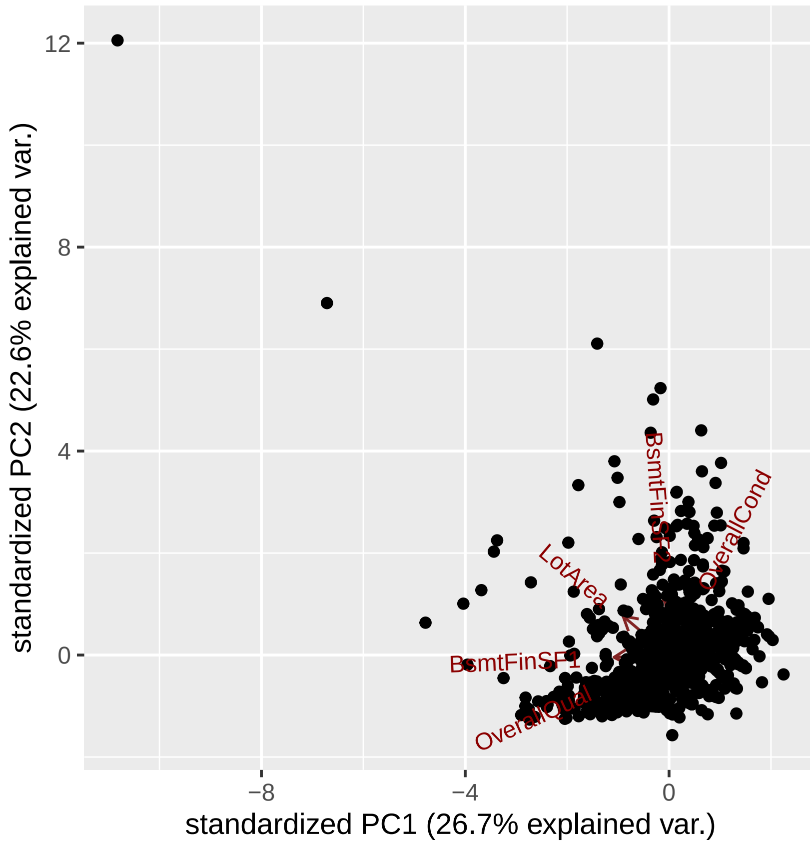
```r
# PCA of the first 5 variables of the data set: LotArea, OverallQual,
# OverallCond, BsmtFinSF1, BsmtFinSF2
pc <- prcomp(hd[, 1:5], scale = TRUE)
summary(pc)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5
## Standard deviation     1.1556 1.0620 0.9852 0.9022 0.8673
## Proportion of Variance 0.2671 0.2256 0.1941 0.1628 0.1504
## Cumulative Proportion  0.2671 0.4927 0.6868 0.8496 1.0000
```

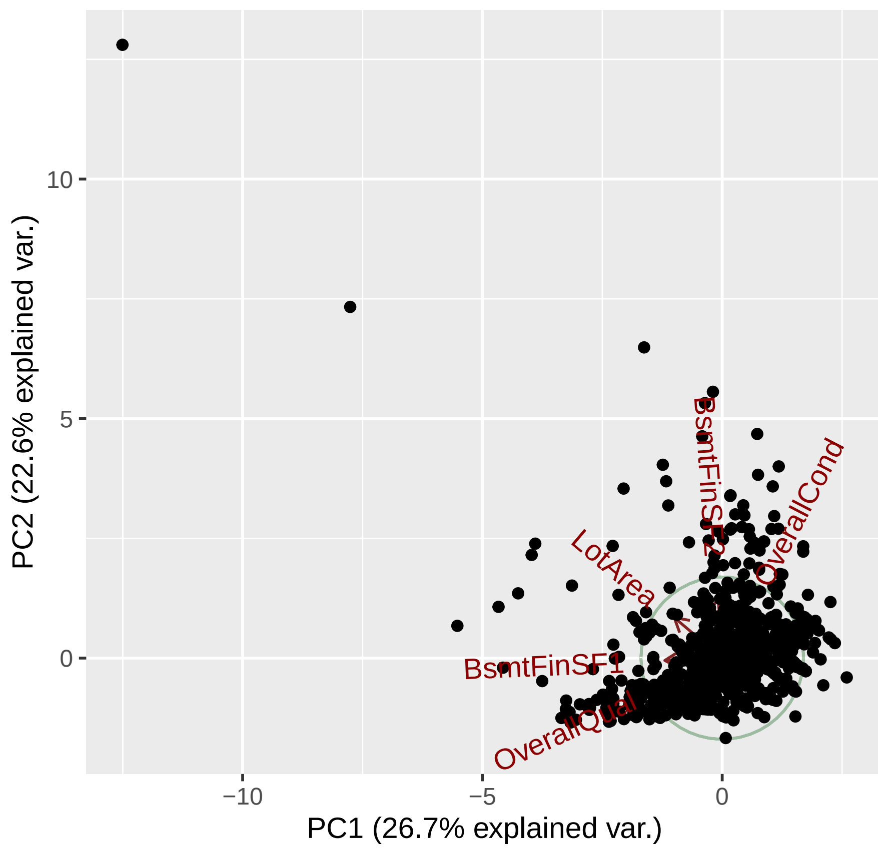
```r
# Correlation matrix of the subset
c2Mat <- cor(hd[, 1:5])

# Correlation plot of the subset based on the correlation matrix
corrplot(c2Mat, method = "ellipse", type = "upper")
```




```
# Plot of the principal components 1 and 2
ggbiplot(pc)
```




```
ggbiplot(pc, obs.scale = 1, var.scale = 1, varname.size = 4, labels.size = 10, circle = TRUE)
```



Together components 1 and 2 explain about 49.3% of the variance in the original housing data. From the biplot created, we are not able to observe clear and strong varability between the variables. In conclusion, we were not able to find anything worthwhile or notable from this PCA analysis about the housing data.