ISE 5103 Intelligent Data Analytics
Homework #3

Group: Homework #3 6

Member:

1. Md Monjur Hossain Bhuiyan
2. Sujata Sahu

---

1. Glass Data

The study of classification of types of glass is motivated by criminological investigations. At the scene of a crime, the glass left can be used as evidence... if it is correctly identified.

The data set we consider consists of 213 unique glass samples labeled as one of six class categories[1]:

type	description
1	building windows float processed
2	building windows non-float processed
3	vehicle windows float processed
5	containers
6	tableware
7	headlamps

There are nine predictors, including the refractive index and percentages of the following eight elements found in the glass: Na (Sodium), Mg (Magnesium), Al (Aluminum), Si (Silicon), K (Potassium), Ca (Calcium), Ba (Barium), and Fe (Iron).

Problem 1(a): Mathematics of PCA

I. Create the correlation matrix of all the numerical attributes in the Glass data and store the results in a new object corMat.

Solution:

R code:

```
data("Glass")
GD<-(Glass[,-10])          ## Taking the numeric attributes from data frame
corMat<-cor(GD)            ## creating correlation matrix
View(corMat)
```

Output:

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

II. Compute the eigenvalues and eigenvectors of corMat.

Solution:

R code:

```
corMat_ev<-eigen(corMat)      ## Computing eigen values and vectors of corMat
corMat_ev
```

Output:

```
eigen() decomposition
$values
[1] 2.511163726 2.050072185 1.404843994 1.157862446 0.914002247 0.527635193 0.368958443 0.063852948 0.001608818

$vectors
           [,1]        [,2]          [,3]        [,4]         [,5]        [,6]        [,7]        [,8]        [,9]
 [1,]  0.5451766 -0.28568318 -0.0869108293  0.14738099  0.073542700 -0.11528772  0.08186724  0.75221590  0.02573194
 [2,] -0.2581256 -0.27035007  0.3849196197  0.49124204 -0.153683304  0.55811757  0.14858006  0.12769315 -0.31193718
 [3,]  0.1108810  0.59355826 -0.0084179590  0.37878577 -0.123509124 -0.30818598 -0.20604537  0.07689061 -0.57727335
 [4,] -0.4287086 -0.29521154 -0.3292371183 -0.13750592 -0.014108879  0.01885731 -0.69923557  0.27444105 -0.19222686
 [5,] -0.2288364  0.15509891  0.4587088382 -0.65253771 -0.008500117 -0.08609797  0.21606658  0.37992298 -0.29807321
 [6,] -0.2193440  0.15397013 -0.6625741197 -0.03853544  0.307039842  0.24363237  0.50412141  0.10981168 -0.26050863
 [7,]  0.4923061 -0.34537980  0.0009847321 -0.27644322  0.188187742  0.14866937 -0.09913463 -0.39870468 -0.57932321
 [8,] -0.2503751 -0.48470218 -0.0740547309  0.13317545 -0.251334261 -0.65721884  0.35178255 -0.14493235 -0.19822820
 [9,]  0.1858415  0.06203879 -0.2844505524 -0.23049202 -0.873264047  0.24304431  0.07372136  0.01627141 -0.01466944
```

III. Use prcomp to compute the principal components of the Glass attributes (make sure to use the scale option).

<u>Solution</u>:

<u>R code:</u>

```
pcofGlass<-prcomp((GD),scale=T)     ## computing principal components.
pcofGlass
summary(pcofGlass)
```

<u>Output:</u>

```
> pcofGlass
Standard deviations (1, .., p=9):
[1] 1.58466518 1.43180731 1.18526115 1.07604017 0.95603465 0.72638502 0.60741950 0.25269141 0.04011007

Rotation (n x k) = (9 x 9):
          PC1         PC2           PC3         PC4          PC5         PC6         PC7         PC8         PC9
RI -0.5451766  0.28568318 -0.0869108293 -0.14738099  0.073542700 -0.11528772 -0.08186724 -0.75221590 -0.02573194
Na  0.2581256  0.27035007  0.3849196197 -0.49124204 -0.153683304  0.55811757 -0.14858006 -0.12769315  0.31193718
Mg -0.1108810 -0.59355826 -0.0084179590 -0.37878577 -0.123509124 -0.30818598  0.20604537 -0.07689061  0.57727335
Al  0.4287086  0.29521154 -0.3292371183  0.13750592 -0.014108879  0.01885731  0.69923557 -0.27444105  0.19222686
Si  0.2288364 -0.15509891  0.4587088382  0.65253771 -0.008500117 -0.08609797 -0.21606658 -0.37992298  0.29807321
K   0.2193440 -0.15397013 -0.6625741197  0.03853544  0.307039842  0.24363237 -0.50412141 -0.10981168  0.26050863
Ca -0.4923061  0.34537980  0.0009847321  0.27644322  0.188187742  0.14866937  0.09913463  0.39870468  0.57932321
Ba  0.2503751  0.48470218 -0.0740547309 -0.13317545 -0.251334261 -0.65721884 -0.35178255  0.14493235  0.19822820
Fe -0.1858415 -0.06203879 -0.2844505524  0.23049202 -0.873264047  0.24304431 -0.07372136 -0.01627141  0.01466944
> summary(pcofGlass)
Importance of components:
                         PC1    PC2    PC3    PC4    PC5     PC6    PC7     PC8     PC9
Standard deviation     1.585 1.4318 1.1853 1.0760 0.9560 0.72639 0.6074 0.25269 0.04011
Proportion of Variance 0.279 0.2278 0.1561 0.1286 0.1016 0.05863 0.0410 0.00709 0.00018
Cumulative Proportion  0.279 0.5068 0.6629 0.7915 0.8931 0.95173 0.9927 0.99982 1.00000
```

<u>Explanation:</u> From the Cumulative Proportion, at the PC6 we reach 95.17%, and so more than 95% of the variability has been explained. This allows us to exclude PC7, PC8 and PC9.

IV. Compare the results from (ii) and (iii) – Are they the same? Different? Why?

Solution:

From ii, we get the eigen values and vectors. And from iii, we get the rotation matrix of the principal components along with the standard deviations. The results show that both eigen vectors and rotation matrix have the same values except their axes flipped with respect to each other, this means that one algorithm has (-) as an indicator where the other uses (+). As a matter of fact, they look different, but they hold the same information in the same manner, just with inverted axes. Moreover, we can get the eigen values by squaring the standard deviation values found from principal components.

V. Using R demonstrate that principal components 1 and 2 from (iii) are orthogonal. (Hint: the inner product between two vectors is useful in determining the angle between the two vectors)

Solution:

we can use the inner product to determine the angle between two vectors as per following equation.

ab = |a| |b| CosØ

For two vectors to be orthogonal, the angle between them should be 90 degree which makes the equation as follows

ab = 0

The product of the PC1 and PC2 is 1.040834e-17 This value is so small, that it can be attributed to machine precision errors. Thus, we can conclude that these two vectors are orthogonal.

R code:

```
product<-t(pcofGlass$rotation[,2]) %*%
pcofGlass$rotation[,1]
product
```

Output:

```
> product
             [,1]
[1,] 1.040834e-17
```

Problem 1(b): Application of PCA

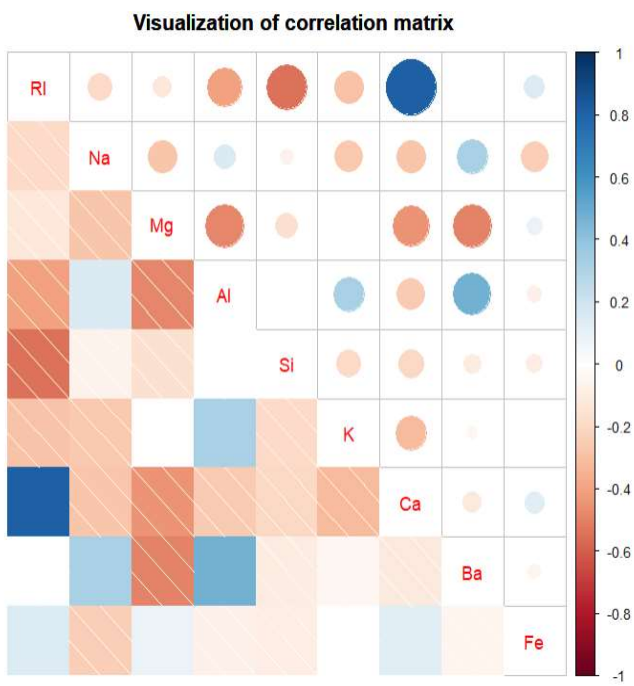
I. **Create a visualization of the corMat correlation matrix (i.e., a heatmap or variant)**

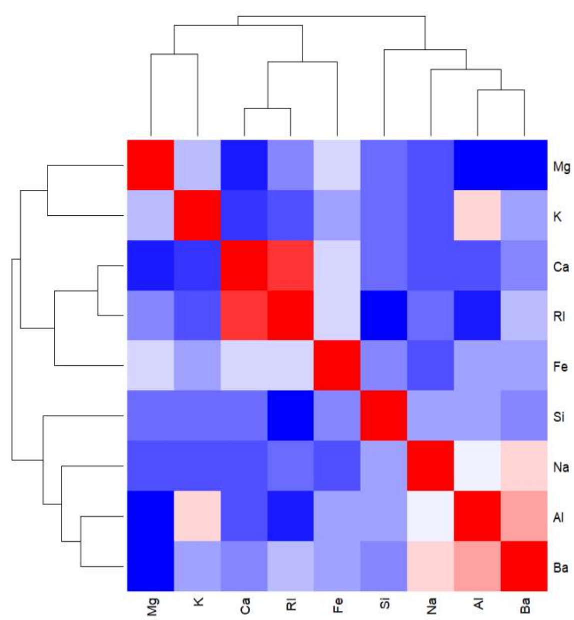
Solution:

R code:

```
corrplot.mixed(corMat,lower = 'shade', upper = 'circle',        ## Visualization of correlation matrix
title="Visualization of correlation matrix",mar=c(0,0,1,0))

col<- colorRampPalette(c("blue", "white", "red"))(20)           ## Creating heatmap
heatmap(corMat, col = col, symm = TRUE)
```

Output:





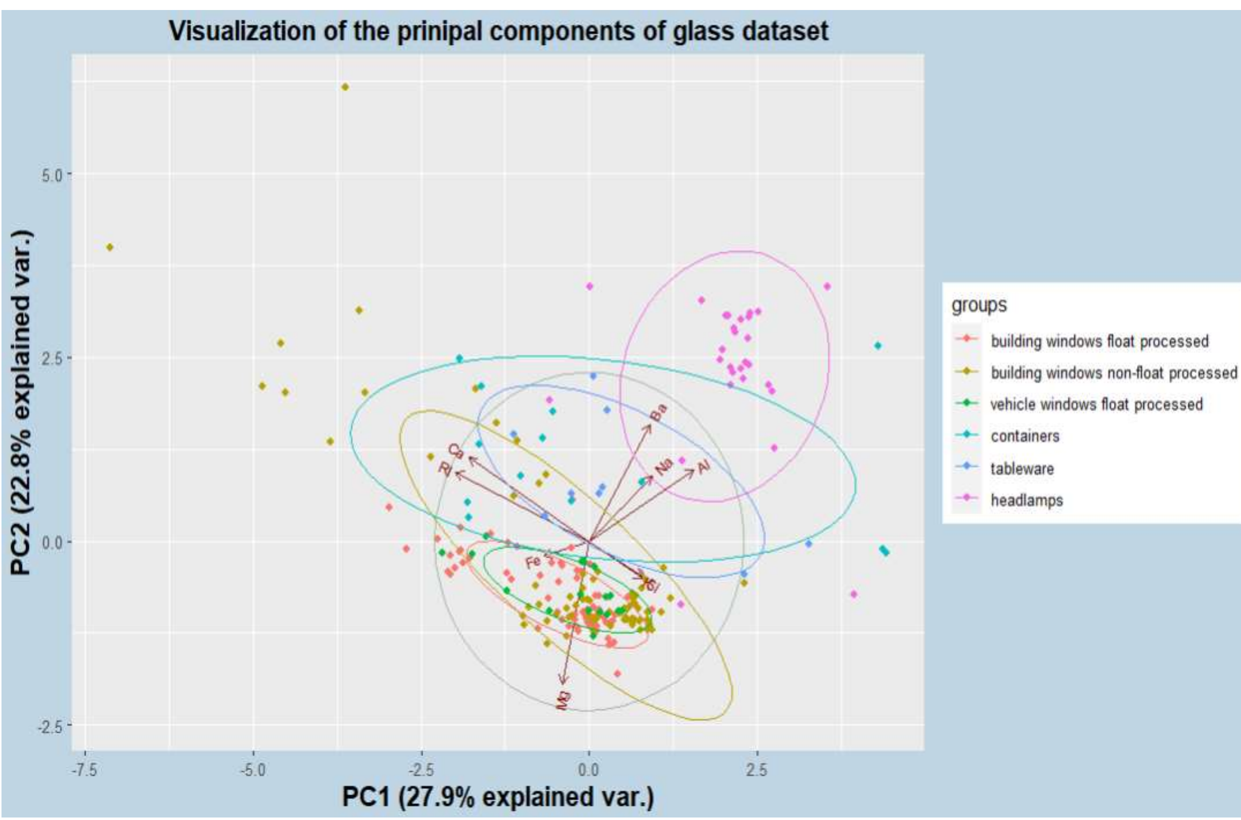
II. **Provide visualizations of the principal component analysis results from the Glass data. Consider incorporating the glass type to group and color your biplot.**

Solution:

R code:

```
## labeling the description based on type
levels(Glass$Type)[levels(Glass$Type)=='1'] <-'building windows float processed'
levels(Glass$Type)[levels(Glass$Type)=='2'] <-'building windows non-float processed'
levels(Glass$Type)[levels(Glass$Type)=='3'] <-'vehicle windows float processed'
levels(Glass$Type)[levels(Glass$Type)=='5'] <-'containers'
levels(Glass$Type)[levels(Glass$Type)=='6'] <-'tableware'
levels(Glass$Type)[levels(Glass$Type)=='7'] <-'headlamps'

 ggbiplot(pcofGlass, obs.scale = 1, var.scale = 1,
  groups = Glass$Type, ellipse = TRUE, circle = TRUE) +
  theme(legend.direction = 'vertical', legend.position = 'right')+
  labs(title="Visualization of the prinipal components of glass dataset")+
  theme(plot.title=element_text(face="bold",hjust=0.5,size = 15),
  axis.title.x = element_text(face="bold",size = 15),
  axis.title.y = element_text(face="bold",size = 15))+
  theme(plot.background=element_rect(fill="#BFD5E3"))
```

Output:



Explanation: From the Bi-Plot above we have PC1 on the x-axis and PC2 on the y-axis. The ellipses explain us each type of the data. Within the main circle there are arrows representing the features of our dataset. We can see that; Ca and RI have high correlation and, they have small correlation with Fe. Moreover, AL, Ba, and Na also corelated strongly. On the contrary, Mg is far away from the other features. If we look at the x-axis, we have Ba, Na, and Al on the right side, at a positive value of 1.5 and above, and this means that these variables are positive correlated.

III. Provide an interpretation of the first two principal components the Glass data.

Solution:

R code:
summary(pcofGlass)

Output:

```
> summary(pcofGlass)
Importance of components:
                         PC1    PC2    PC3    PC4    PC5     PC6    PC7     PC8
Standard deviation     1.585 1.4318 1.1853 1.0760 0.9560 0.72639 0.6074 0.25269
Proportion of Variance 0.279 0.2278 0.1561 0.1286 0.1016 0.05863 0.0410 0.00709
Cumulative Proportion  0.279 0.5068 0.6629 0.7915 0.8931 0.95173 0.9927 0.99982
                           PC9
Standard deviation     0.04011
Proportion of Variance 0.00018
Cumulative Proportion  1.00000
```

Explanation:
The first two principal components capture the most variance of the data. PC1 captures 27.9% of the variance, whereas PC2 captured 22.8 % of the variance of the data. Also, the summary above shows, after PC1 and PC2, the Cumulative Proportion of the variance is 0.5068. That is about 50.58% of the total variance can be explained by PC1 and PC2.

IV. Based on the PCA results, do you believe that you can effectively reduce the dimension of the data? If so, to what degree? If not, why?

Solution:
If we see the summarized PCA result, we can see the PC6 covers almost 95% of variance of the data based on what we can think of reducing the dimension. But we cannot go beyond that (PC3) because it only comprises 66% of data which will certainly be very likely to provide an inaccurate conclusion.

Problem 1(c): Application of LDA

I. **Since the Glass data is grouped into various labeled glass types, we can consider linear discriminant analysis (LDA) as another form of dimension reduction. Use the lda method from the MASS package to reduce the Glass data dimensionality**.

Solution:

R code:

```
LDA_Glass<-lda(Type ~ . , data = Glass)
LDA_Glass
```

Output:

```
lda(Type ~ ., data = Glass)

Prior probabilities of groups:
    building windows float processed building windows non-float processed
                          0.32710280                           0.35514019
     vehicle windows float processed                           containers
                          0.07943925                           0.06074766
                           tableware                            headlamps
                          0.04205607                           0.13551402

Group means:
                                          RI       Na        Mg       Al
building windows float processed     1.518718 13.24229 3.5524286 1.163857
building windows non-float processed 1.518619 13.11171 3.0021053 1.408158
vehicle windows float processed      1.517964 13.43706 3.5435294 1.201176
containers                           1.518928 12.82769 0.7738462 2.033846
tableware                            1.517456 14.64667 1.3055556 1.366667
headlamps                            1.517116 14.44207 0.5382759 2.122759
                                           Si         K        Ca          Ba
building windows float processed     72.61914 0.4474286  8.797286 0.012714286
building windows non-float processed 72.59803 0.5210526  9.073684 0.050263158
vehicle windows float processed      72.40471 0.4064706  8.782941 0.008823529
containers                           72.36615 1.4700000 10.123846 0.187692308
tableware                            73.20667 0.0000000  9.356667 0.000000000
headlamps                            72.96586 0.3251724  8.491379 1.040000000
                                             Fe
building windows float processed     0.05700000
building windows non-float processed 0.07973684
vehicle windows float processed      0.05705882
containers                           0.06076923
tableware                            0.00000000
headlamps                            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
```

Explanation: From the analysis, about 32.71% belongs to float processed building windows and 35.51% belongs to non-float processed building windows type of dataset. Headlamps consists of 13.55% of the dataset. The minimum of the dataset 4.2% belongs to the tableware groups. Also, from the proportion of trace, about 81.45% variances can be described with LD1.

II. How would you interpret the first discriminant function, LD1?

Solution:

The first discriminant function LD1 is a linear combination of all 9 numeric attributes of the dataset. The "proportion of trace"" is the percentage separation achieved by each discriminant function. The first discriminant function LD1 consist of 81.45 %; the maximum variance of dataset. It indicates the good separation between the glass types.

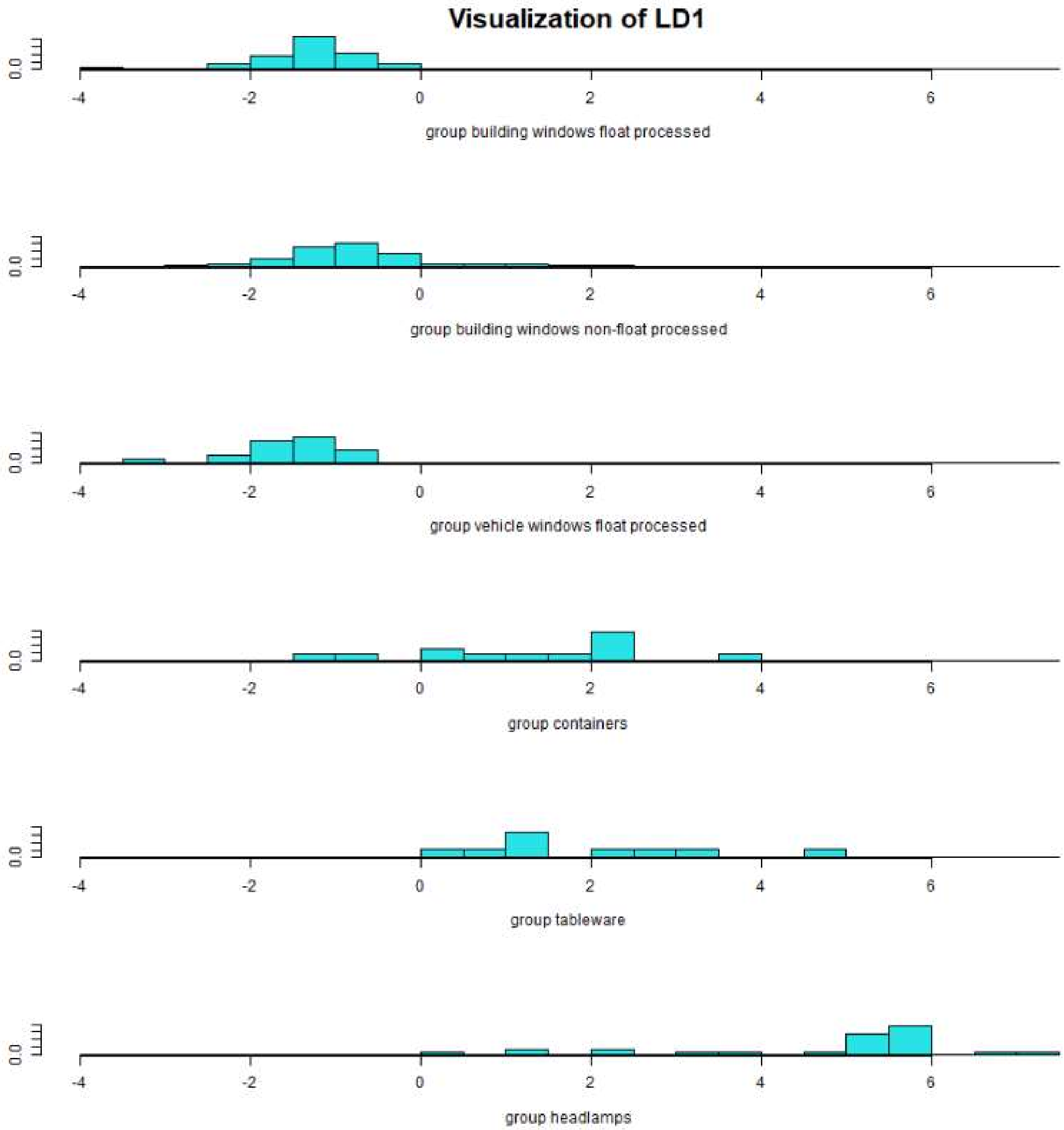
III. Use the ldahist function from the MASS package to visualize the results for LD1 and LD2. Comment on the results.

Solution:

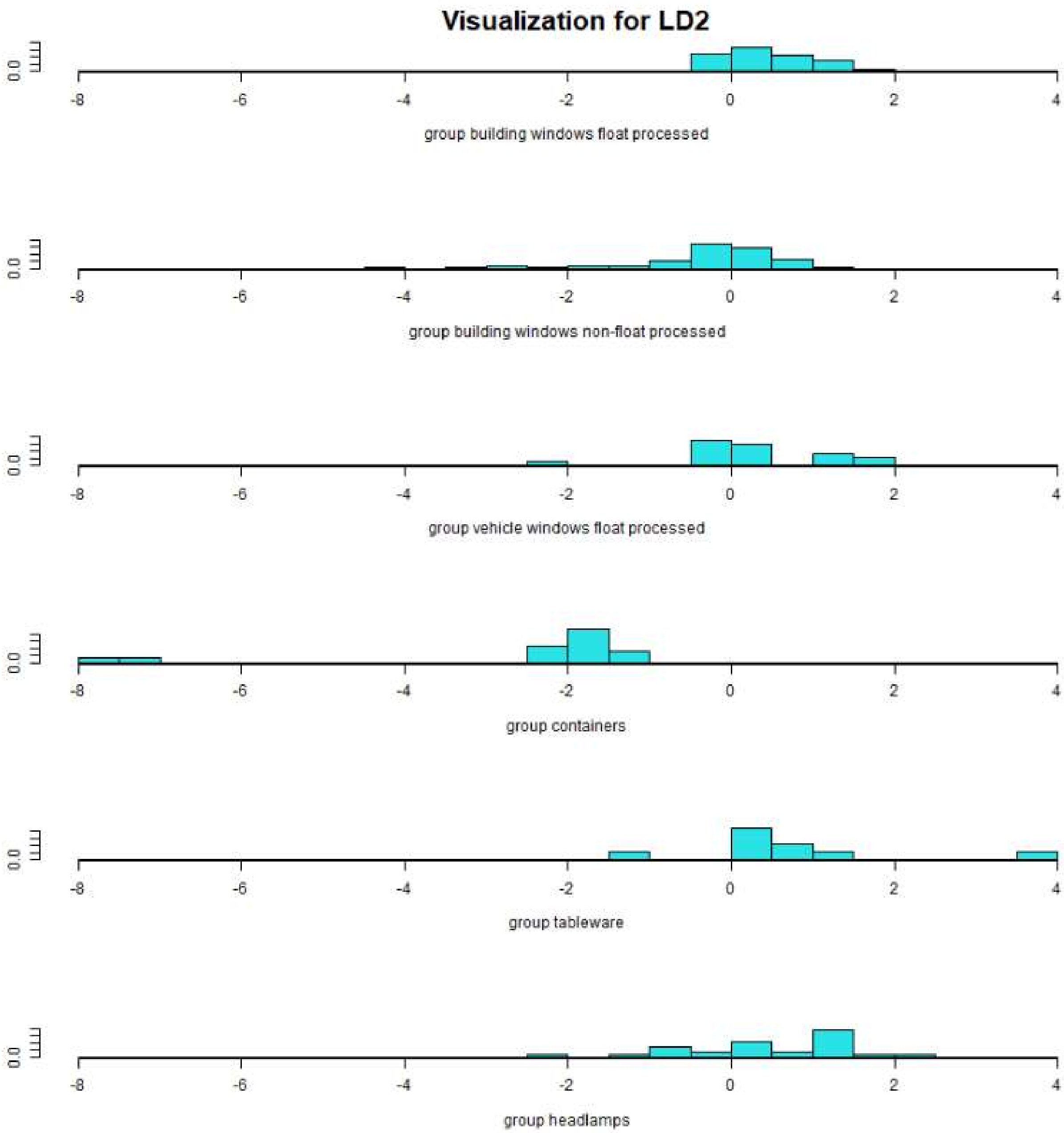
R code:

```
Glass.lda.values <- predict(LDA_Glass)
Glass.lda.values

ldahist(Glass.lda.values$x[,1], g= Glass$Type)+      ## Histogram for LD1
title(main="Visualization of LD1")

ldahist(Glass.lda.values$x[,2], g= Glass$Type)+       ## Histogram for LD2
title(main="Visualization for LD2")
```

Output:



Explanation: These histograms are based on LD1. We already know that about 81.45% separation can be achieved by LD1. It's clear that overlaps found within first, second and third types of the glass. Moreover, overlap also observed in containers, tableware, and headlamps groups separately. But no overlap observed with the first three types of the histogram with the rest three types.



Explanation: These histograms are based on LD2. In this case very high overlapping can be observed in between the different types of the glass dataset. And this kind of overlapping is not a good sign for analysis.

2. Principal components for dimension reduction

The HSAUR2 package contains the data heptathlon which are the results of the women's olympic heptathlon competition in Seoul, Korea from 1988. A scoring system is used to assign points to the results from each of the seven events and the winner is the woman who accumulates the most points over the two days.

<u>Problem 2(a)</u>: Examine the event results using the Grubb's test. According to this test there is one competitor who is an outlier multiple events: Who is the competitor? And for which events is there statistical evidence that she is an outlier? Remove her from the data.

<u>Solution</u>:

<u>R code:</u>

```
tests = lapply(heptathlon,grubbs.test)      ## Grubbs test for outlier

tests

x<-rm.outlier(heptathlon)                    ## Removing outlier

x
```

<u>Output:</u>

```
$hurdles

        Grubbs test for one outlier

data:  X[[i]]
G = 3.5024, U = 0.4676, p-value = 0.000436
alternative hypothesis: highest value 16.42 is an outlier


$highjump

        Grubbs test for one outlier

data:  X[[i]]
G = 3.61806, U = 0.43184, p-value = 0.0001698
alternative hypothesis: lowest value 1.5 is an outlier


$shot

        Grubbs test for one outlier

data:  X[[i]]
G = 2.08971, U = 0.81047, p-value = 0.3702
alternative hypothesis: lowest value 10 is an outlier


$run200m

        Grubbs test for one outlier

data:  X[[i]]
G = 2.15480, U = 0.79847, p-value = 0.3048
alternative hypothesis: lowest value 22.56 is an outlier


$longjump

        Grubbs test for one outlier

data:  X[[i]]
G = 2.68319, U = 0.68752, p-value = 0.04594
alternative hypothesis: lowest value 4.88 is an outlier
```

```
$javelin

        Grubbs test for one outlier

data:  X[[i]]
G = 1.69718, U = 0.87498, p-value = 1
alternative hypothesis: highest value 47.5 is an outlier


$run800m

        Grubbs test for one outlier

data:  X[[i]]
G = 3.30186, U = 0.52681, p-value = 0.001808
alternative hypothesis: highest value 163.43 is an outlier


$score

        Grubbs test for one outlier

data:  X[[i]]
G = 2.68194, U = 0.68781, p-value = 0.04618
alternative hypothesis: lowest value 4566 is an outlier
```

<u>Explanation:</u> Looking at the output of Grubbs test of individual sport Launa is the outlier as she has the lowest score in total that is 4566.

Looking at the Grubbs test she is the outlier for hurdles, high jump, long jump, run 800m. We removed the outlier as given code above and finally the modified data set is as follows.

	hurdles	highjump	shot	run200m	longjump	javelin	run800m	score
1	12.69	1.86	15.80	23.65	7.27	45.66	128.51	7291
2	12.85	1.80	16.23	23.10	6.71	42.56	126.12	6897
3	13.20	1.83	14.20	23.92	6.68	44.54	124.20	6858
4	13.61	1.80	15.23	23.93	6.25	42.78	132.24	6540
5	13.51	1.74	14.76	24.65	6.32	47.46	127.90	6540
6	13.75	1.83	13.50	23.59	6.33	42.82	125.79	6411
7	13.38	1.80	12.88	24.48	6.37	40.28	132.54	6351
8	13.55	1.80	14.13	24.86	6.47	38.00	133.65	6297
9	13.63	1.83	14.28	23.59	6.11	42.20	136.05	6252
10	13.25	1.77	12.62	25.03	6.28	39.06	134.74	6252
11	13.75	1.86	13.01	23.59	6.34	37.86	131.49	6205
12	13.24	1.80	12.88	24.87	6.37	40.28	132.54	6171
13	13.85	1.86	11.58	24.78	6.05	44.58	134.93	6137
14	13.71	1.83	13.16	24.61	6.12	45.44	142.82	6109
15	13.79	1.80	12.32	25.00	6.08	38.60	137.06	6101
16	13.93	1.86	14.21	25.47	6.40	35.76	146.67	6087
17	13.47	1.80	12.75	24.83	6.34	44.34	138.48	5975
18	14.07	1.83	12.69	24.92	6.13	37.76	146.43	5972
19	14.39	1.71	12.68	25.61	6.10	35.68	138.02	5746
20	14.04	1.77	11.81	25.69	5.99	39.48	133.90	5734
21	14.31	1.77	11.66	25.50	5.75	39.64	133.35	5686
22	14.23	1.71	12.95	25.23	5.50	39.14	144.02	5508
23	14.85	1.68	10.83	26.61	5.47	39.26	137.30	5290
24	14.53	1.71	11.78	26.16	5.50	46.38	139.17	5289

<u>Problem 2(b):</u> As is, some event results are "good" if the values are large (e.g. highjump), but some are "bad" if the value is large (e.g. time to run the 200 meter dash). Transform the running events (hurdles, run200m, run800m) so that large values are good. An easy way to do this is to subtract values from the max value for the event, i.e. $x_i \leftarrow x_{max} - x_i$.

<u>Solution</u>:

<u>R code:</u>

```
hurdlemax <- max(heptathlon$hurdles)          ##Taking the max values
hurdlemax
run200mmax <- max(heptathlon$run200m)
run200mmax
run800mmax <- max(heptathlon$run800m)
run800mmax

heptathlon$hurdles <- (hurdlemax-heptathlon$hurdles)    ##Subtracting values from max values
heptathlon$hurdles
heptathlon$run200m <- (run200mmax-heptathlon$run200m)
heptathlon$run200m
heptathlon$run800m <- (run800mmax-heptathlon$run800m)
heptathlon$run800m
```

Output:

```
> heptathlon$hurdles
 [1] 3.73 3.57 3.22 2.81 2.91 2.67 3.04 2.87 2.79 3.17 2.67 3.18 2.57 2.71 2.63 2.49 2.95 2.35 2.03 2.38 2.11 2.19 1.57 1.89 0.00
> heptathlon$run200m
 [1] 4.05 2.96 3.51 2.69 2.68 1.96 3.02 2.13 1.75 3.02 1.58 3.02 1.74 1.83 2.00 1.61 1.14 1.78 1.69 1.00 0.92 1.11 1.38 0.00 0.45
> heptathlon$run800m
 [1] 34.92 37.31 39.23 31.19 35.53 37.64 30.89 29.78 27.38 28.69 31.94 30.89 28.50 20.61 26.37 16.76 24.95 17.00 25.41 29.53 30.08 19.41 26.13 24.26  0.00
```

Problem 2(c): Perform a principal component analysis on the 7 event results and save the results of the prcomp function to a new variable Hpca.

Solution:

R code:

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

Hpca
```

Output:

```
Standard deviations (1, .., p=7):
[1] 2.0642116 1.0536510 0.7944775 0.7017471 0.5696168 0.3742378 0.2017162

Rotation (n x k) = (7 x 7):
                PC1         PC2         PC3         PC4         PC5        PC6        PC7
hurdles   0.4488384 -0.02995664  0.05883809 -0.06607130  0.44886757 -0.6874526 -0.3402149
highjump -0.3296102  0.48319930  0.53561202 -0.40048682  0.06287237 -0.2858172  0.3534986
shot     -0.4022277  0.03439308 -0.20373745  0.69497084  0.27085077 -0.3436843  0.3479333
run200m   0.4383653 -0.05216536 -0.04058024  0.08904143 -0.66715848 -0.3826853  0.4516537
longjump -0.4413143  0.19191799 -0.14655665  0.02582044 -0.50845055 -0.3422282 -0.6087869
javelin  -0.2014123 -0.70237460  0.64593824  0.15002942 -0.10630661 -0.0862457 -0.0872843
run800m   0.3172140  0.48120093  0.47727045  0.56670674 -0.07925490  0.2378909 -0.2366514
```

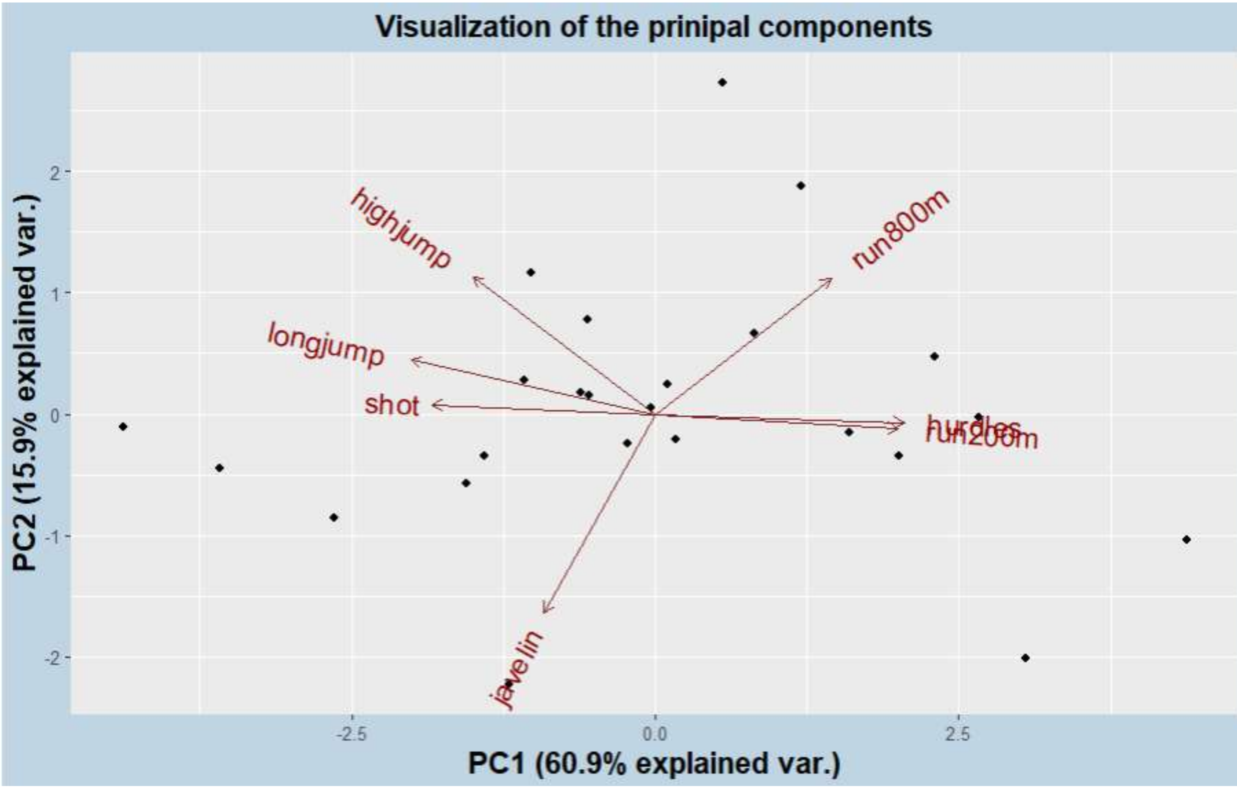
Problem 2(d): Use ggibiplot to visualize the first two principal components. Provide a concise interpretation of the results.

Solution:

R code:

```
ggbiplot(pcobj=Hpca,choices=c(1,2),obs.scale=1, var.scale=1,
varname.size=5,varname.abbrev=FALSE)+
 labs(title="Visualization of the prinipal components")+
 theme(plot.title=element_text(face="bold",hjust=0.5,size = 15),
 axis.title.x = element_text(face="bold",size = 15),
 axis.title.y = element_text(face="bold",size = 15))+
 theme(plot.background=element_rect(fill="#BFD5E3"))
```

Output:



Explanation: Here we can see that PC1 explains almost 60.9% of the variances while PC2 explains 15.9% of the total variances. From the plot we can see that hurdles and run 200m are positively correlated whereas the shot and run 200m are negatively correlated. The hurdles and run200m has high impact on PC1.Javelin throw are making the huge impact on both PC1 and PC2 as the line of javelin throw is the longest line which is almost at 45 degree.

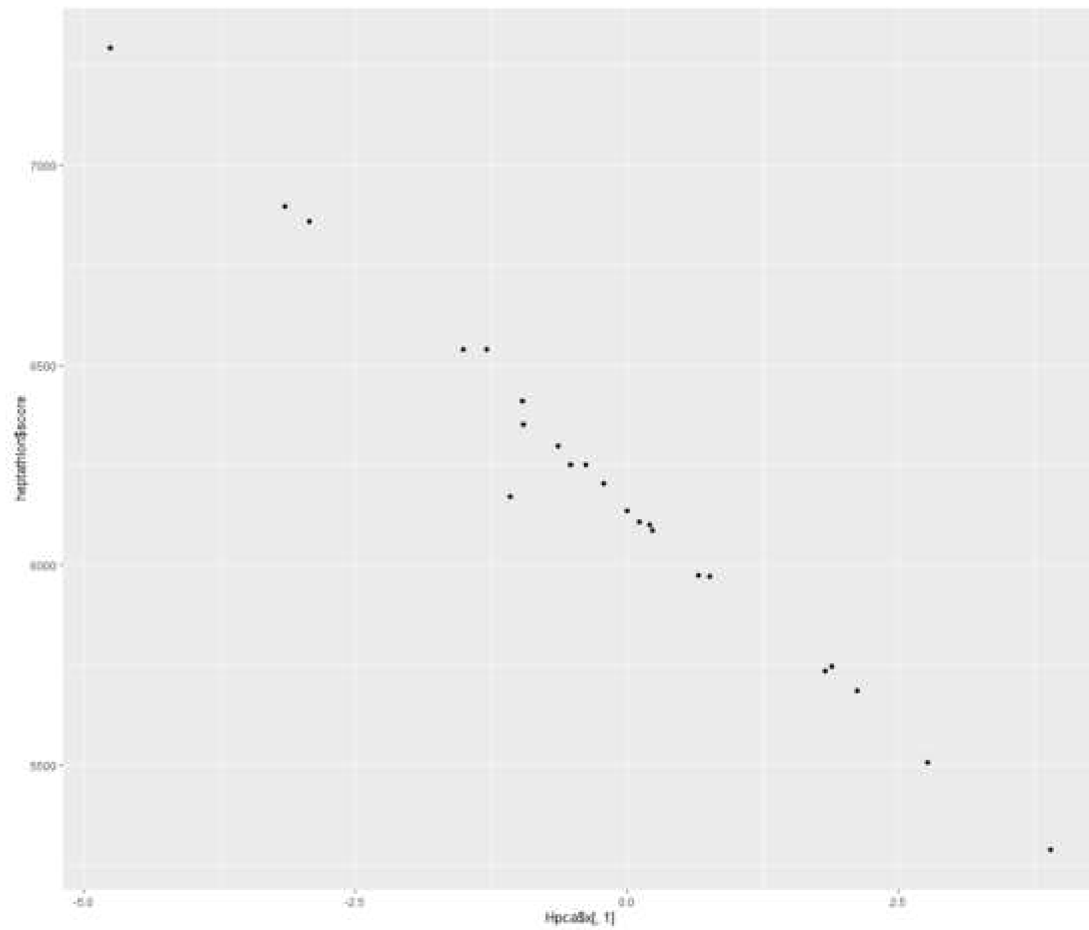
Problem 2(e): The PCA projections onto principal components 1, 2, 3, . . . for each competitor can now be accessed as Hpca$x[,1], Hpca$x[,2], Hpca$x[,3], ... Plot the heptathlon score against the principal component 1 projections. Briefly discuss these results.

Solution:

R code:

```
ggplot(Hpca)+
geom_point(mapping=aes(x=Hpca$x[,1],y=heptathlon$score))
```

Output:



Explanation: The plot depicts as the score increases PC1 decreases

---

3. Housing data dimension reduction and exploration

Using this newly created data set hd (as per given instruction), perform PCA and correlation analysis. Did you find anything worthwhile? Make sure to respond with visualizations and interpretations of at least the most important principal components.

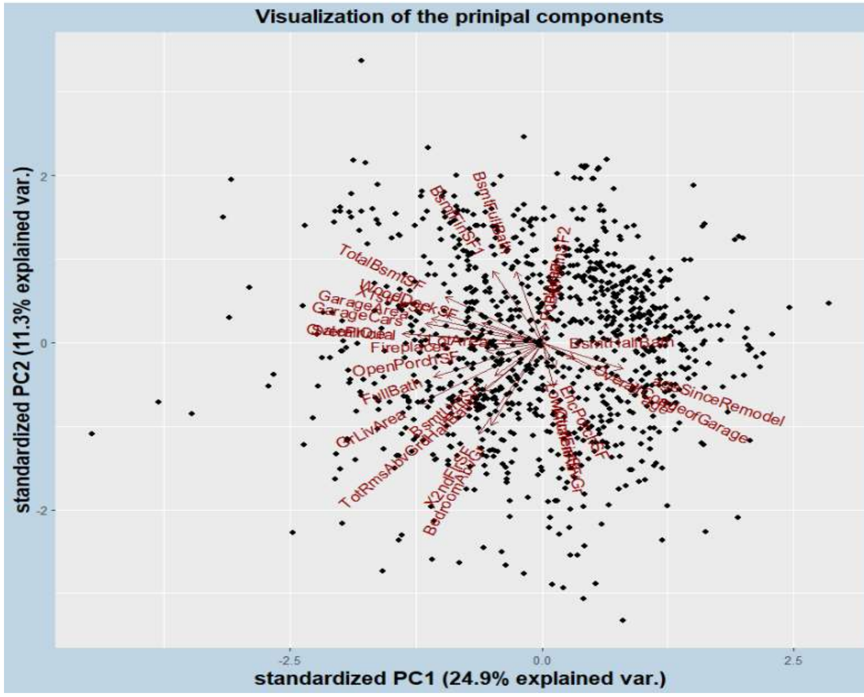
Solution:

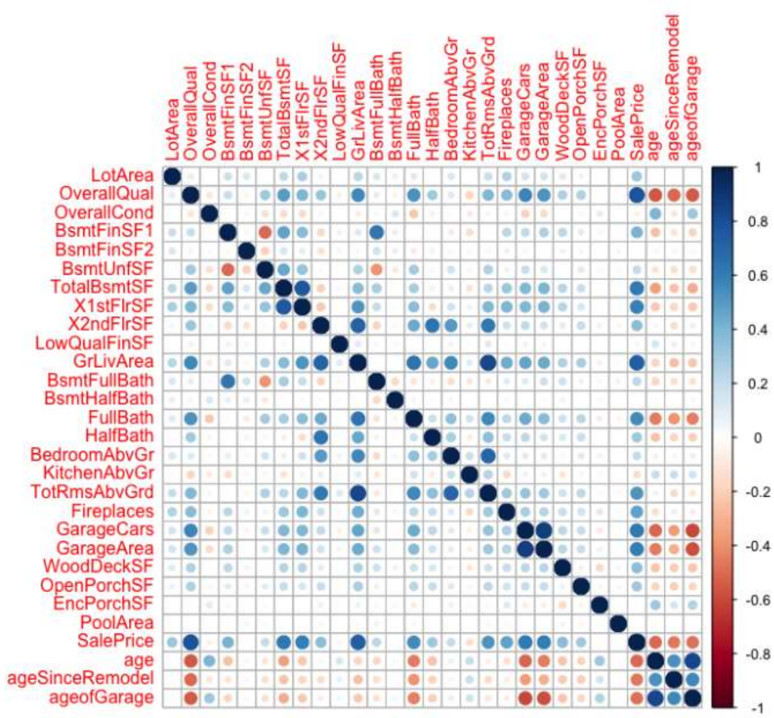
R code:

```
## reading "housingData" dataframe
housedata<-read.csv("housingData.csv")
view(housedata)
```

```
## select numeric columns
hd <- housedata %>%
 select_if(is.numeric) %>%

##creates new variables age, ageSinceRemodel, and ageofGarage
  dplyr::mutate(age = YrSold - YearBuilt,
              ageSinceRemodel = YrSold - YearRemodAdd,
              ageofGarage = ifelse(is.na(GarageYrBlt), age, YrSold - GarageYrBlt)) %>%

##removes a few columns that are not needed
  dplyr::select(!c(Id,MSSubClass, LotFrontage, GarageYrBlt,
                MiscVal, YrSold , MoSold, YearBuilt,
                YearRemodAdd, MasVnrArea))

## PCA for dataset hd
pc=prcomp(hd, scale.=T)
pc

## Visualization of PCA
ggbiplot(pcobj=pc,choices=c(1,2),varname.size=4,varname.abbrev=FALSE)+
 labs(title="Visualization of the prinipal components")+
 theme(plot.title=element_text(face="bold",hjust=0.5,size = 15),
 axis.title.x = element_text(face="bold",size = 15),
  axis.title.y = element_text(face="bold",size = 15))+
  theme(plot.background=element_rect(fill="#BFD5E3"))

## CRA
CRA<- cor(hd)
CRA

## plot the CRA visualisation
corrplot(CRA)
```

Output:





Explanation: Looking at the plot of principal component analysis the sale price and overall quality is positively correlated. Age of garage and garage area are correlated. The PC1 only covers the24.9% of total area and PC2 covers only 11.3 % of the variables Most of the data is scattered away. That's the reason of both PC1 and PC2 together only covers a total of 36.2% of the total variances.