Group23_Ida_homework3 - Vignesh Murugan And Yazan AbuAwad

2024-09-06

Question 1

(a) (15 points) 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.

```r
# Compute the correlation matrix for the first 9 columns
CorMat <- cor(Glass[, 1:9])
```

---

ii. Compute the eigenvalues and eigenvectors of corMat.

```r
# Perform eigen decomposition on the correlation matrix
eigen_result <- eigen(CorMat)
eigen_values <- eigen_result$values
eigen_vector <- eigen_result$vectors
```

---

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

```r
# Perform Principal Component Analysis (PCA) on the first 9 columns
pca <- prcomp(Glass[,1:9], scale. = TRUE)
```

---

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

```r
# Compare eigenvalues to the variances explained by PCA components
print(pca$sdev^2)  # Squaring standard deviations to get variances
```

```
## [1] 2.510152168 2.058169337 1.407484057 1.144693344 0.914768873 0.528593040
## [7] 0.370262639 0.064267543 0.001608997
```

```r
print(eigen_values)
```

```
## [1] 2.510152168 2.058169337 1.407484057 1.144693344 0.914768873 0.528593040
## [7] 0.370262639 0.064267543 0.001608997
```

The results from (ii) and (iii) are same.

The eigenvalues from the correlation matrix's eigen decomposition represent the variance explained by each principal component, and these values are equivalent to the squared singular values (variances) calculated in prcomp. This equivalence arises because both approaches decompose the same matrix—correlation for scaled data, capturing how much variance each principal component accounts for. Thus, regardless of the method used, the key output—how variance is distributed across principal components—remains consistent, validating PCA's effectiveness in dimensionality reduction.

---

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)

```r
# Extract the first two principal components
pc1 <- pca$x[, 1]
pc2 <- pca$x[, 2]

# Compute the inner product (should be close to 0 if orthogonal)
inner_product <- sum(pc1 * pc2)
print(inner_product)
```

```
## [1] -3.392842e-13
```

The result of the inner product between the first two principal components, which is approximately -3.392842e-13, is essentially zero. This extremely small value, close to zero, confirms that the first two principal components are orthogonal.

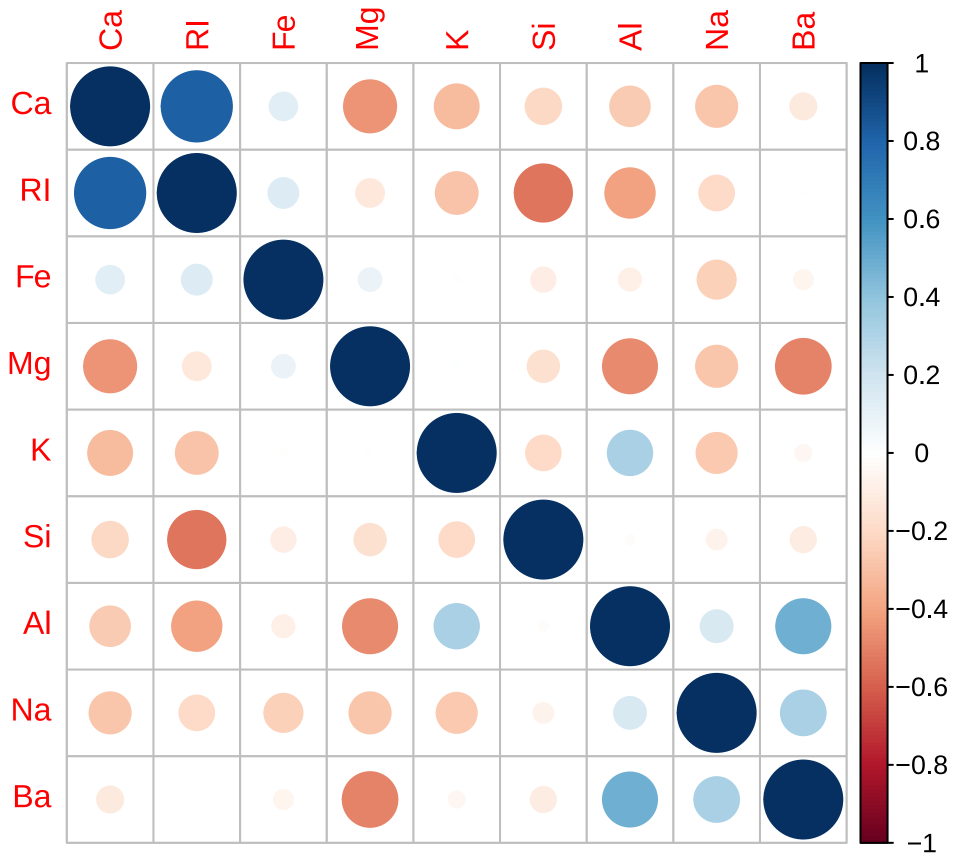
(b) (15 points) Application of PCA

i. Create a visualization of the corMat correlation matrix (i.e., a heatmap or variant) If you are interested and have time, consider the corrplot package for very nice options, https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html.

```r
library(corrplot)
```

```
## corrplot 0.94 loaded
```

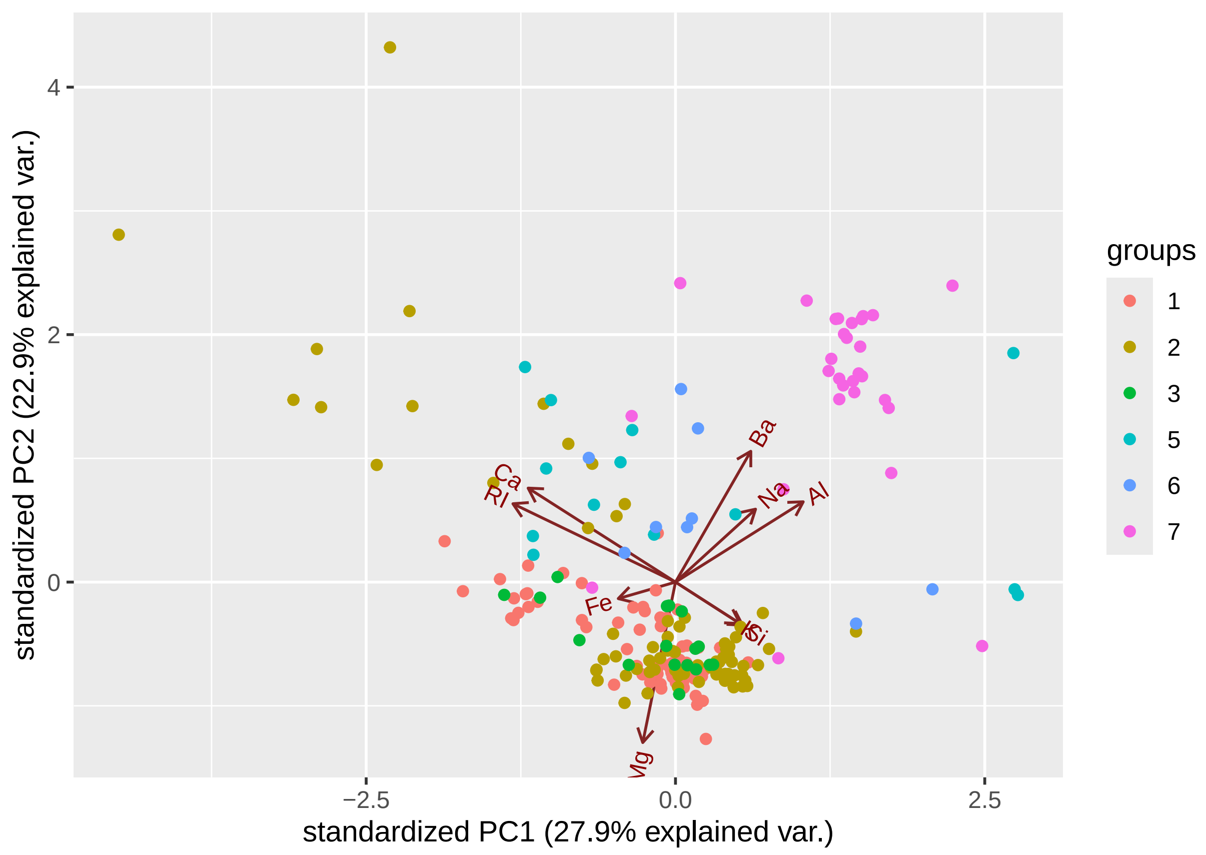
```r
# Visualize the correlation matrix
corrplot(CorMat, method = "circle", order = "AOE")
```



---

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

```r
# Visualize PCA using ggbiplot
ggbiplot(pca, groups = Glass$Type)
```



iii. Provide an interpretation of the first two prinicpal components the Glass data.

In PCA biplot,Variables with strong negative coefficients in PC1 are CA,RI and in PC2 are MG. Variables with strong positive coefficients in PC2 are Ba. The variables Ca and Ri have strong positive correlation and both have negative correlation with Si. The variable Mg has negative correlation with Ba.

```
print(pca)
```

```
## Standard deviations (1, .., p=9):
## [1] 1.58434597 1.43463213 1.18637433 1.06990343 0.95643550 0.72704404 0.60849210
## [8] 0.25351044 0.04011231
##
## Rotation (n x k) = (9 x 9):
##           PC1         PC2         PC3        PC4         PC5         PC6
## RI -0.5432231  0.28911804  0.08849541  0.1479796 -0.07670808  0.11455615
## Na  0.2676141  0.26909913 -0.36710090  0.5010669  0.14626769 -0.55790564
## Mg -0.1093261 -0.59215502  0.02295318  0.3842440  0.11610001  0.30585293
## Al  0.4269512  0.29636272  0.32602906 -0.1488756  0.01720068 -0.02014091
## Si  0.2239232 -0.15874450 -0.47979931 -0.6394962  0.01763694  0.08850787
## K   0.2156587 -0.15305116  0.66349177 -0.0733491 -0.30154622 -0.24107648
## Ca -0.4924367  0.34678973 -0.01380151 -0.2743430 -0.18431431 -0.14957911
## Ba  0.2516459  0.48262056  0.07649040  0.1299431  0.24970936  0.65986429
## Fe -0.1912640 -0.06089167  0.27223834 -0.2252596  0.87828176 -0.24066617
##            PC7         PC8         PC9
## RI -0.08223530 -0.75177166 -0.02568051
## Na -0.15419352 -0.12819398  0.31188932
## Mg  0.20691746 -0.07799332  0.57732740
## Al  0.69982052 -0.27334224  0.19041178
## Si -0.20945417 -0.38077660  0.29747147
## K  -0.50515516 -0.11064442  0.26075531
## Ca  0.09984144  0.39885229  0.57999243
## Ba -0.35043794  0.14497643  0.19853265
## Fe -0.07120579 -0.01650505  0.01459278
```

---

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?

Based on the results of the Principal Component Analysis (PCA), dimensionality reduction is not advisable in this instance. Typically, for PCA to be effective, the first few principal components should capture a substantial portion of the variance in the dataset. However, in this case, the first two principal components account for only 50% of the total variance, which is considerably low. This suggests that reducing dimensions may lead to significant information loss, thus impacting the integrity of the dataset.

```r
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6     PC7
## Standard deviation     1.5843 1.4346 1.1864 1.0699 0.9564 0.72704 0.60849
## Proportion of Variance 0.2789 0.2287 0.1564 0.1272 0.1016 0.05873 0.04114
## Cumulative Proportion  0.2789 0.5076 0.6640 0.7912 0.8928 0.95154 0.99268
##                            PC8     PC9
## Standard deviation     0.25351 0.04011
## Proportion of Variance 0.00714 0.00018
## Cumulative Proportion  0.99982 1.00000
```

(c) (15 points) 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.

```r
# Preprocess the data: center and scale
preproc.param <- Glass %>% preProcess(method = c("center", "scale"))

# Transform the data using the estimated parameters
transformed <- preproc.param %>% predict(Glass)

# Fit an LDA model to the transformed data
lda.model <- lda(Type ~., data = transformed)
lda.model
```

```
## Call:
## lda(Type ~ ., data = transformed)
##
## Prior probabilities of groups:
##          1          2          3          5          6          7
## 0.32394366 0.35680751 0.07981221 0.06103286 0.04225352 0.13615023
##
## Group means:
##            RI          Na         Mg          Al          Si           K
## 1  0.10586803 -0.21529537  0.6021709 -0.55566964 -0.03051835 -0.07127292
## 2  0.08928758 -0.35801082  0.2236651 -0.08333047 -0.07370057  0.03395576
## 3 -0.12668032  0.04037692  0.5986928 -0.50069475 -0.32346915 -0.14146475
## 5  0.19121411 -0.70579005 -1.3197807  1.17832829 -0.37327809  1.48675612
## 6 -0.29416453  1.52153712 -0.9514821 -0.16699477  0.71265830 -0.76375492
## 7 -0.40605128  1.27100801 -1.4829529  1.35761437  0.40154052 -0.26592900
##            Ca          Ba           Fe
## 1 -0.11782012 -0.32708816  0.005626553
## 2  0.08387772 -0.25209570  0.230146481
## 3 -0.12002632 -0.33526691 -0.002235610
## 5  0.82037781  0.02373083  0.035785004
## 6  0.28233912 -0.35297613 -0.586918470
## 7 -0.32450463  1.73435095 -0.449113729
##
## Coefficients of linear discriminants:
##           LD1        LD2       LD3          LD4        LD5
## RI -0.94568441 0.07527438 1.0863607 -0.818100075 -2.3954624
## Na -1.93653461 2.56835550 0.3710009 -5.620203296  2.1990082
## Mg -1.06291733 4.27171386 2.2661557 -9.823318476  4.4675280
## Al -1.65414159 0.83192958 1.1005447 -3.192430110  0.6006326
## Si -1.89406393 2.29624206 1.3286024 -5.857566773  0.9918384
## K  -1.02248031 1.19667679 0.8324237 -5.232631319  2.0664499
## Ca -1.42618954 3.35575714 0.9098493 -9.431977031  5.6893657
## Ba -1.14918330 1.69975242 1.2850261 -3.150403464  2.3353652
## Fe  0.04927568 0.01955888 0.1195656 -0.002693645 -0.1268396
##
## Proportion of trace:
##    LD1    LD2    LD3    LD4    LD5
## 0.8145 0.1168 0.0417 0.0158 0.0111
```

```r
# Predict using the LDA model
predictions <- lda.model %>% predict(transformed)
```

---

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

Elements with negative coefficients like Na, Mg, and Ca, which have large absolute values, heavily influence LD1 in the negative direction, indicating that higher values of these elements contribute to moving the discriminant score downward. Conversely, Fe, with a small positive coefficient, has a minor positive influence on LD1. These loadings help understand which elements are most important in distinguishing between the groups analyzed.

```r
# Display the scaling of the LDA model
lda.model$scaling
```

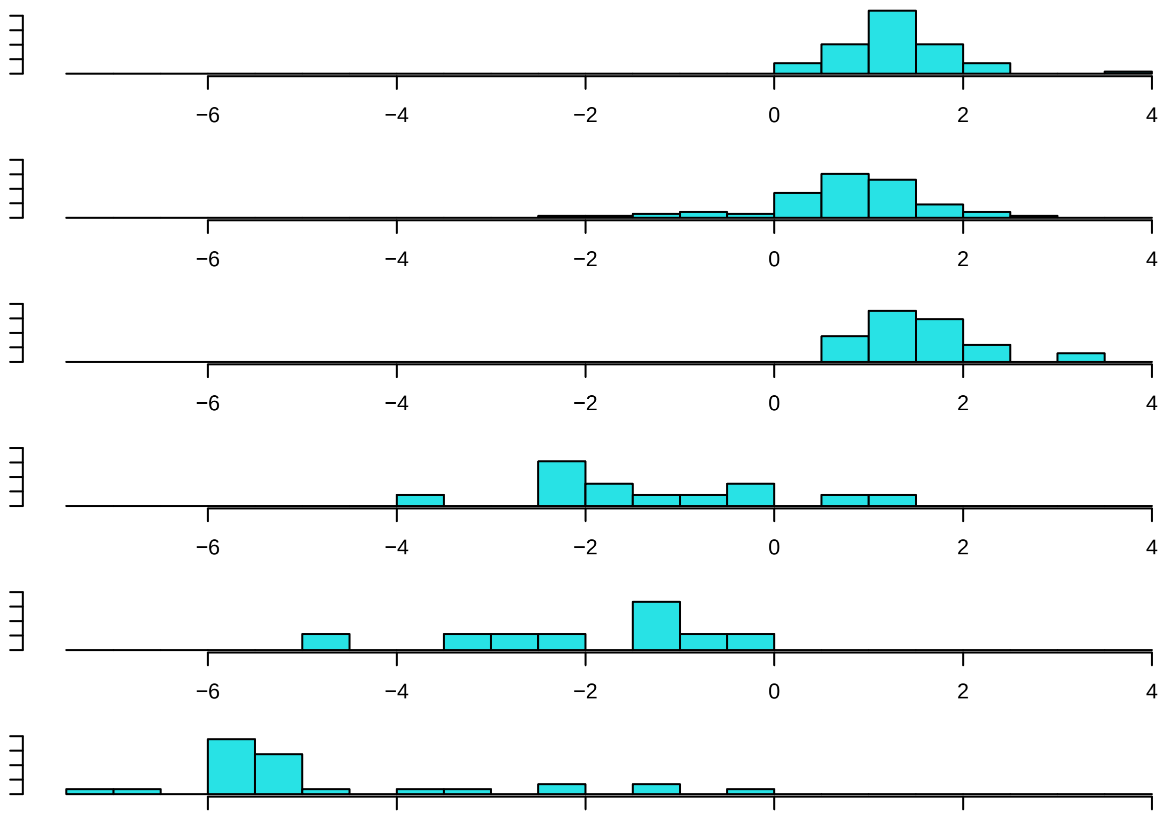
```
##              LD1        LD2       LD3          LD4        LD5
## RI -0.94568441 0.07527438 1.0863607 -0.818100075 -2.3954624
## Na -1.93653461 2.56835550 0.3710009 -5.620203296  2.1990082
## Mg -1.06291733 4.27171386 2.2661557 -9.823318476  4.4675280
## Al -1.65414159 0.83192958 1.1005447 -3.192430110  0.6006326
## Si -1.89406393 2.29624206 1.3286024 -5.857566773  0.9918384
## K  -1.02248031 1.19667679 0.8324237 -5.232631319  2.0664499
## Ca -1.42618954 3.35575714 0.9098493 -9.431977031  5.6893657
## Ba -1.14918330 1.69975242 1.2850261 -3.150403464  2.3353652
## Fe  0.04927568 0.01955888 0.1195656 -0.002693645 -0.1268396
```

---

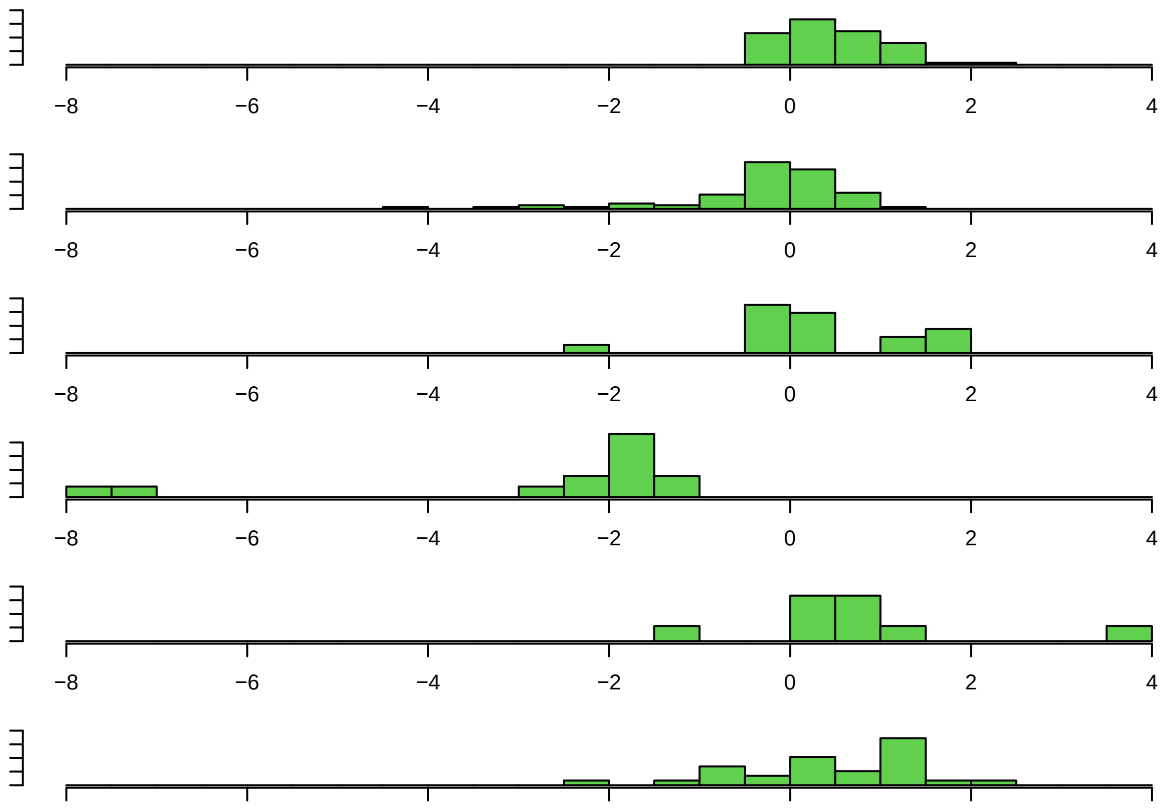
iii. Use the ldahist function from the MASS package to visualize the results for LD1 and LD2. Comment on the results.

In LD1, The type 1,2,3 are correlated, while 5,6,7 seem somewhat correlated.We can differentiate types 1,2,3 from 5,6,7. In LD2, only type 5 shows variation from other groups.

```r
par(mar=c(1,1,2,2))

# Histogram of the first LDA component
ldahist(predictions$x[,1], g = Glass$Type)
```



```r
# Histogram of the second LDA component
ldahist(predictions$x[,2], g = Glass$Type, col = 3)
```



Question 2.Principal components for dimension reduction

(a) (10 points) Examine the event results using the Grubb's test. According to this test there is one competitor who is an outlier for multiple events: Who is the competitor? And for which events is there statistical evidence that she is an outlier? Remove her from the data.

The Outlier is Launa (PNG). For Events High Jump, Long Jump, Run800 there is strong evidence(pvalue<0.05) that there is an Outlier in these variables.

```
# Detect outliers in each column using Grubb's test
outliers <- sapply(heptathlon[,1:7],function (x){
  test <- grubbs.test(x)
})

outlier(heptathlon[1:8])
```

```
##  hurdles highjump     shot  run200m longjump  javelin  run800m    score
##    16.42     1.50    10.00    22.56     4.88    47.50   163.43  4566.00
```

```
# Identify specific outliers in the 'hurdles' event
heptathlon[heptathlon$hurdles==outlier(heptathlon$hurdles),]
```

```
##             hurdles highjump  shot run200m longjump javelin run800m score
## Launa (PNG)   16.42      1.5 11.78   26.16     4.88   46.38  163.43  4566
```

```
# Removing outliler from dataset
heptathlon <- heptathlon[heptathlon$hurdles!=outlier(heptathlon$hurdles),]
```

(b) (5 points) 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.

```
# Transform running event times to scores
heptathlon_transformed <- heptathlon %>%
  mutate(
    hurdles = max(hurdles) - hurdles,
    run200m = max(run200m) - run200m,
    run800m = max(run800m) - run800m
  )
```

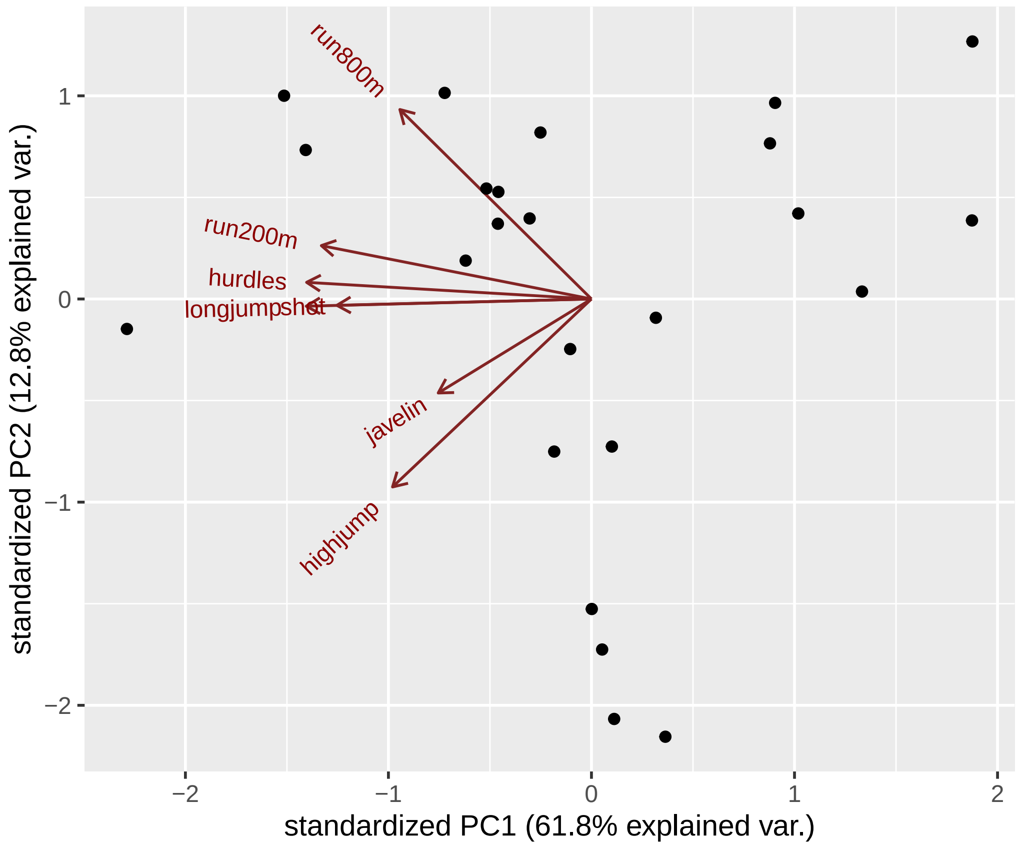
(c) (5 points) Perform a principal component analysis on the 7 event results and save the results of the prcomp function to a new variable Hpca.

```
# Perform PCA on the transformed heptathlon data
Hpca <- prcomp(heptathlon_transformed[, 1:7], scale. = TRUE)
```

(d) (10 points) Use ggibiplot to visualize the first two principal components. Provide a concise interpretation of the results.

The variance explained by Both PCA is 70% , which is kind of acceptable. The events longjump and javelin are highly correlated. Th events run200,hurdles,longjump,shot are highly correlated and these values are highy dependent/correlated with pc1.

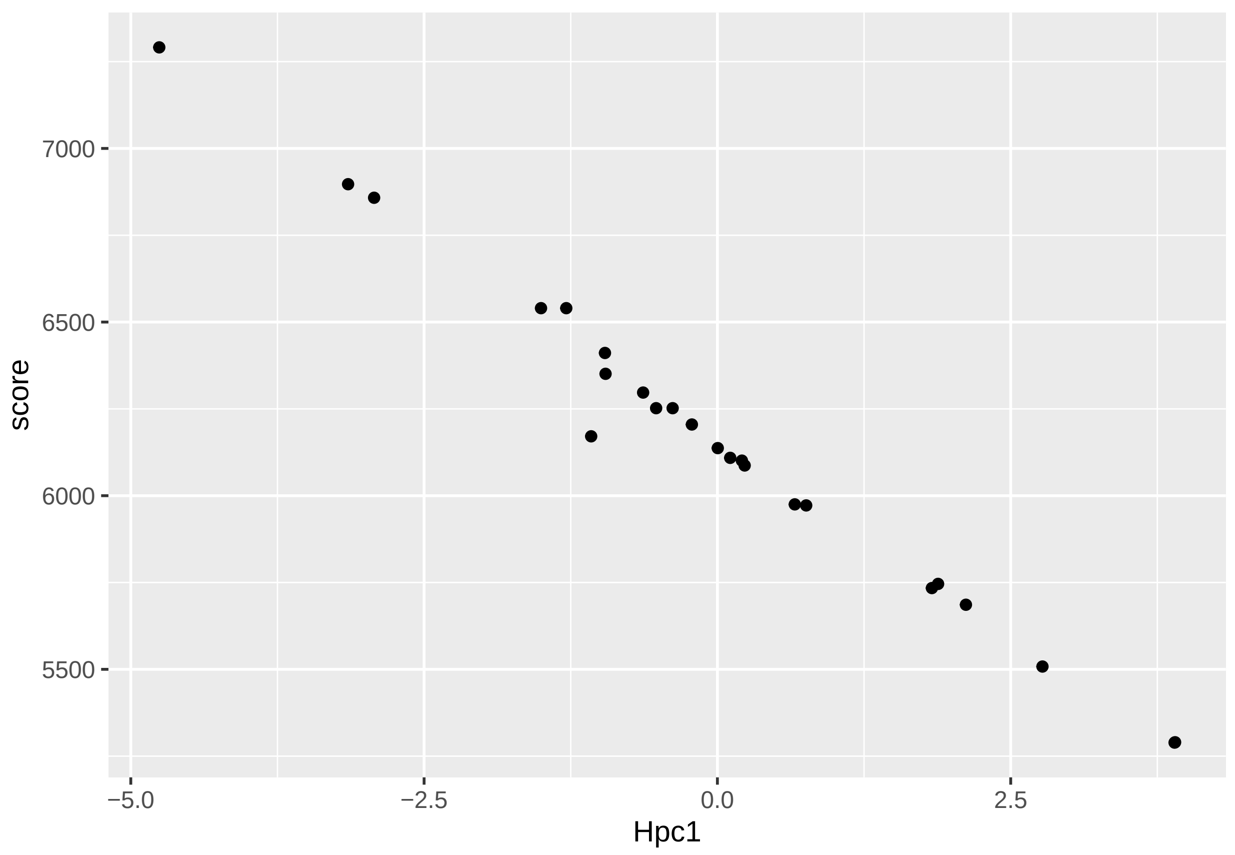
```r
# Visualize PCA results with ggbiplot
ggbiplot(Hpca)
```



(e) (10 points) 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.

As Hpc1 increases the score decreases.

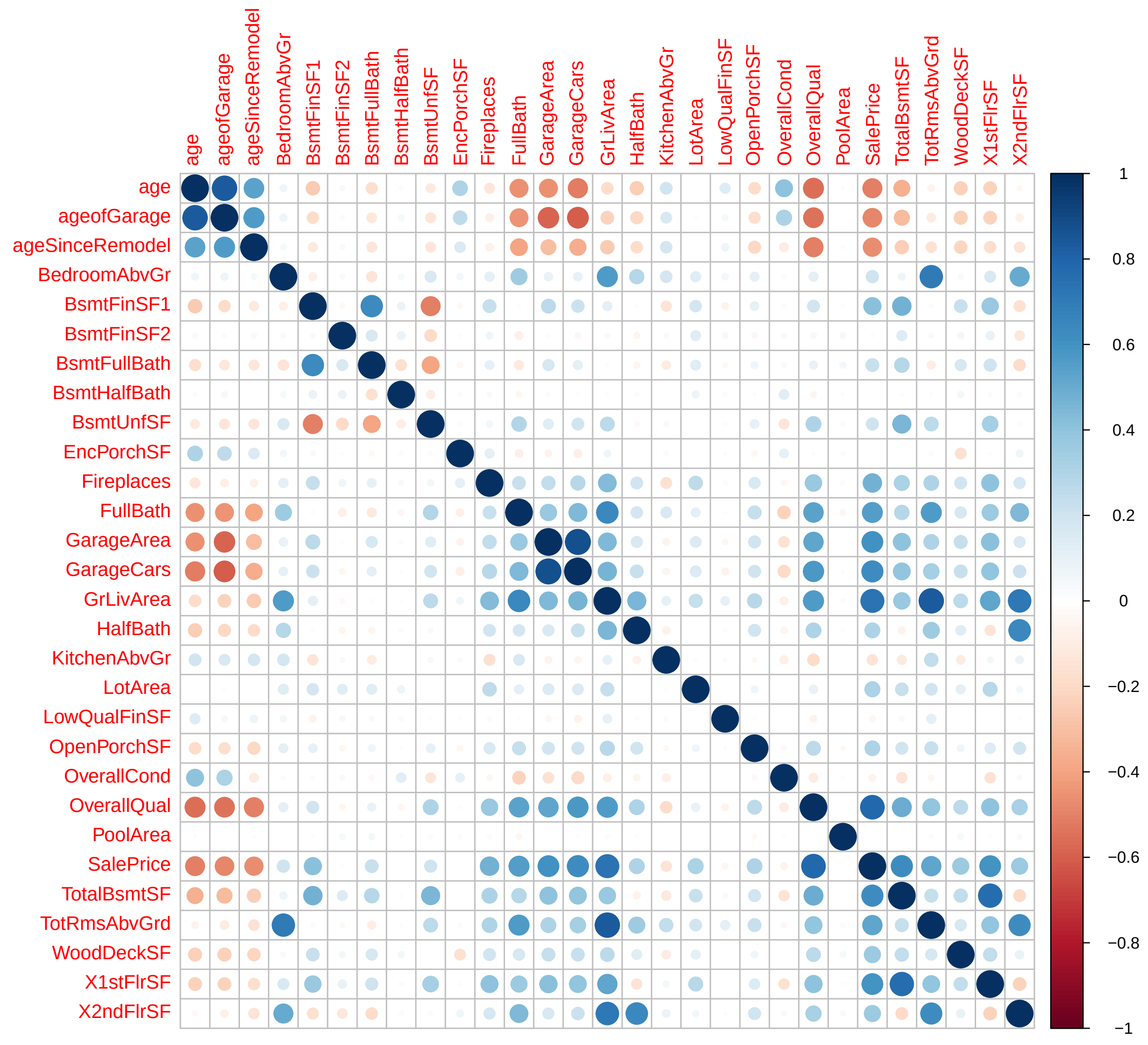
```r
Hpc1 <- Hpca$x[,1]

# Plot first principal component against the heptathlon score
ggplot(mapping = aes(Hpc1, heptathlon$score)) +
  geom_point() +
  labs(y = "score")
```



Question 3.Housing data dimension reduction and exploration

(15 points) Using this newly created data set hd, 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.

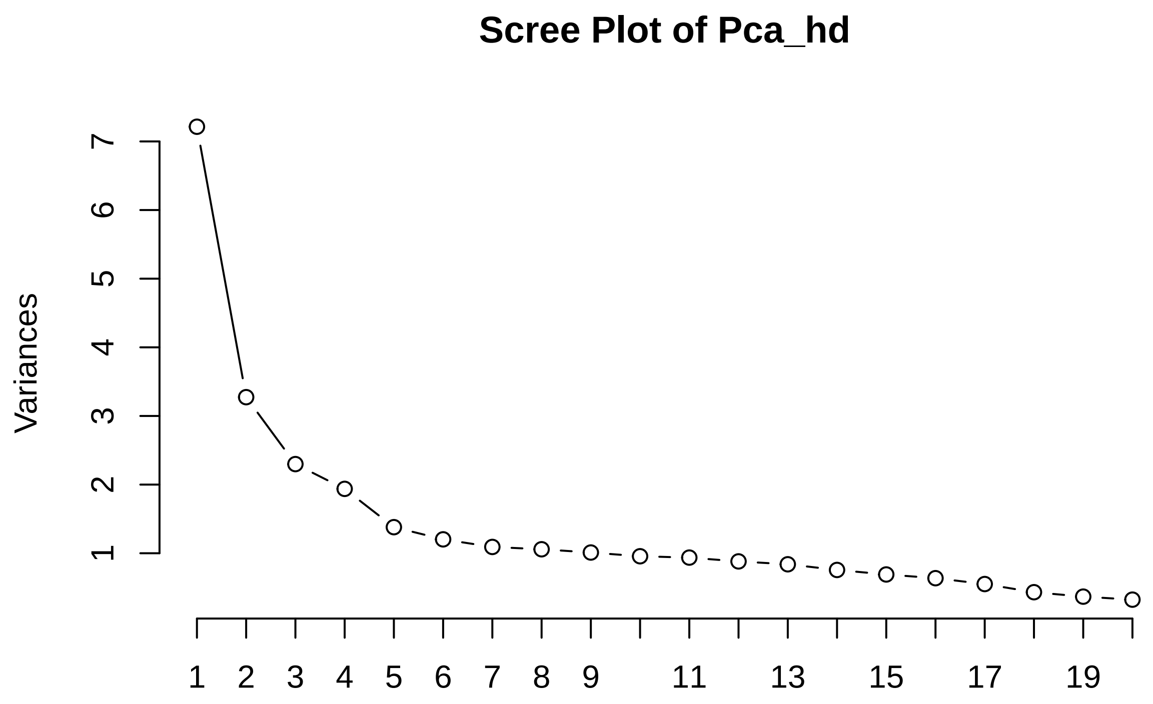
```r
# Compute the correlation matrix of the processed data
cor_hd <- cor(hd)
# Visualize the correlation matrix using corrplot
corrplot(cor_hd,method = "circle", order = "alphabet")
```



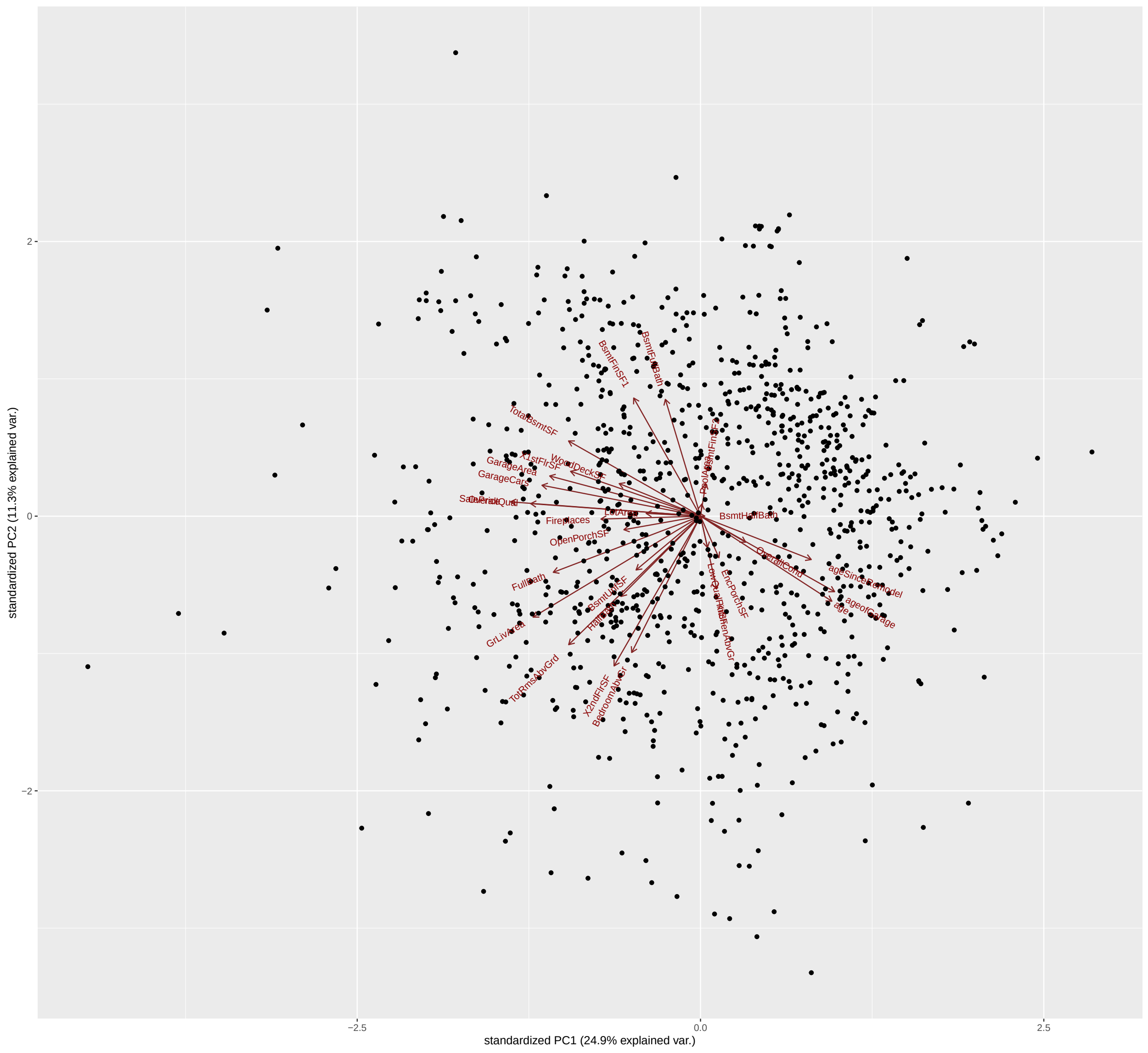
```r
# Perform PCA on the processed housing data
pca_hd <- prcomp(hd, scale. = TRUE)

# Display summary of PCA results
summary(pca_hd)
```

```
## 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     3.821e-16
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

```
# Plot a scree plot to show the variances explained by the principal components
screeplot(pca_hd, npcs = min(20, length(pca_hd$sdev)),
          type = "lines", main = "Scree Plot of Pca_hd")
```



```
# Visualize PCA results using ggbiplot
ggbiplot(pca_hd)
```



From the Correlation Plot , we can say that the variables :

1. Age, ageofGarage, ageSinceRemodel are negtively correlated with OverallQual.
2. Age is negatively correlated with saleprice.
3. OverallQual, Saleprice are positively correlated with Fullbath, GarageArea, GarageCars, GrLivArea.

From Biplot, The variances captured by first 2 pca is very low. Better not to make decisions on this. Reducing the dimensions may lead to significant information loss, thus impacting the integrity of the dataset.