Gym-Members-Clustering

Paula Ramirez

15/10/2024

Gym-Members-Clustering

1. Data Transformation and Descriptive Analysis

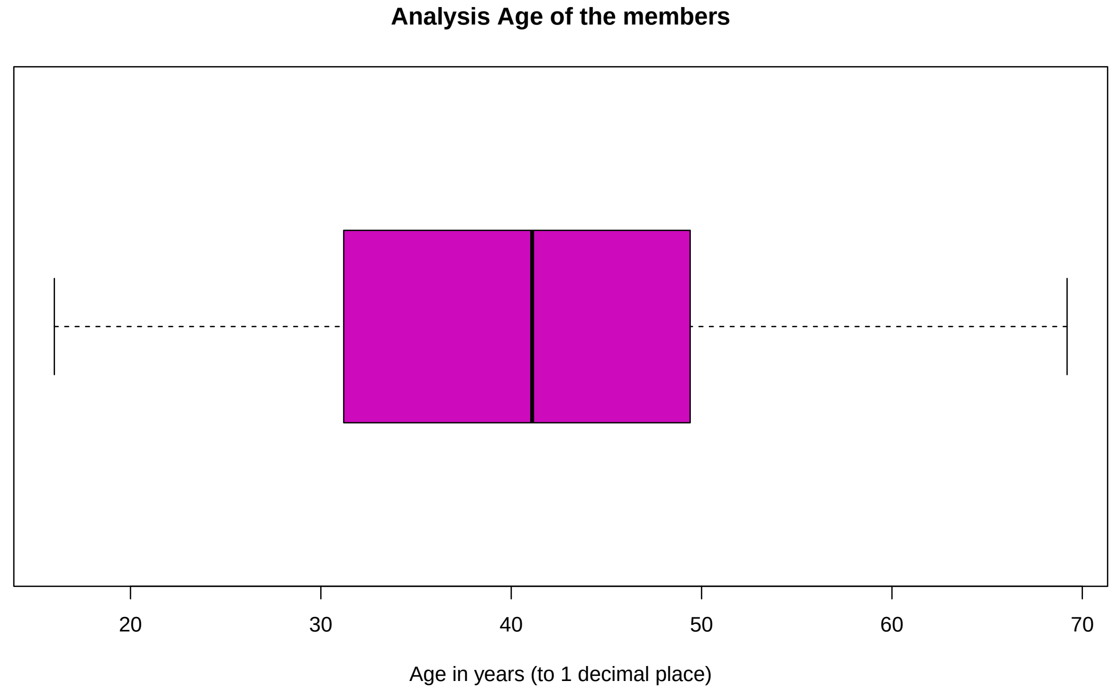
1. Rename all variables

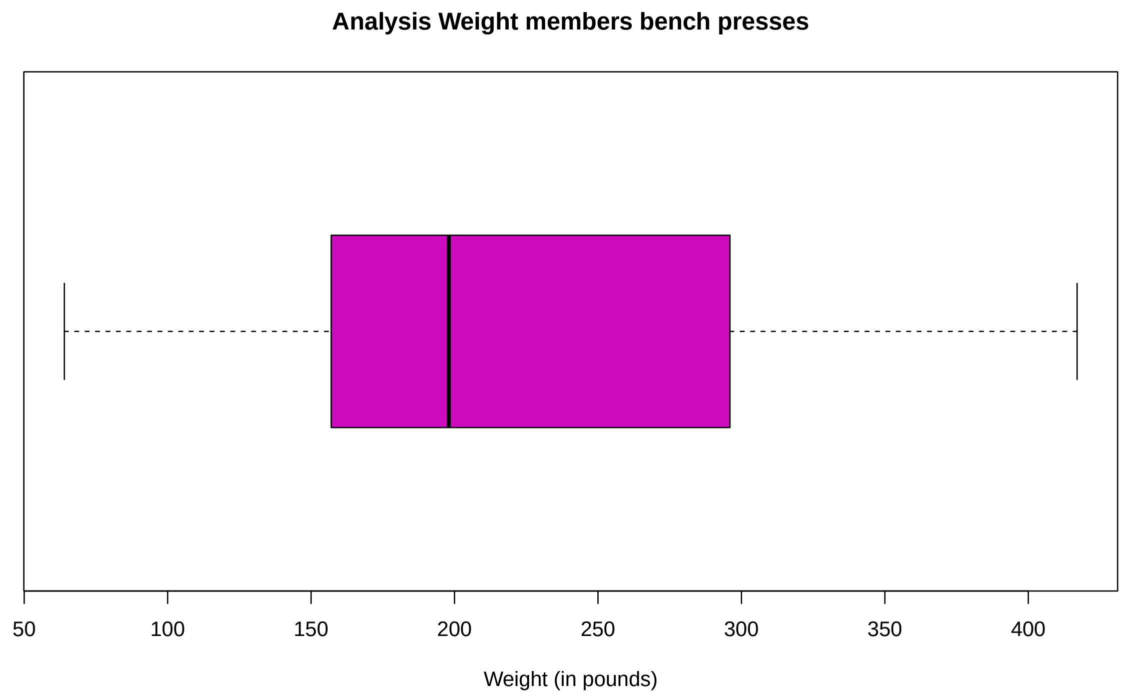
```r
#Reading the file
gym_data_PR <- read.table(here("Gym-Members-Clustering", "Data_Gym.txt"),
                                    header = TRUE, sep = ",")
#Converting it to dataframe.
gym_data_PR <- as.data.frame(gym_data_PR)
#Append PR initials to all variables in the dataframe
colnames(gym_data_PR) <- paste(colnames(gym_data_PR), "PR", sep = "_")
#Showing first results
head(gym_data_PR)
```

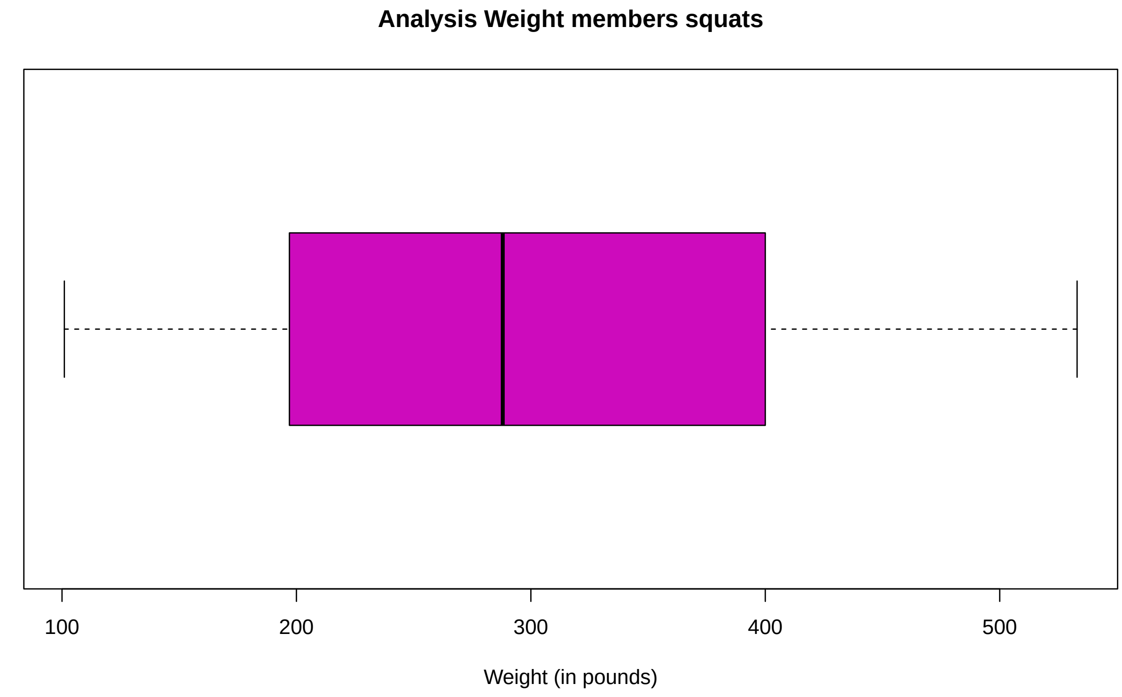
```
##   Age_PR BP_PR Sq_PR DL_PR PU_PR
## 1   35.2   398   375   493    39
## 2   63.7   261   531   514    14
## 3   23.5   157   134   272     3
## 4   37.2   160   114   236    10
## 5   45.9   128   156   354    14
## 6   47.7   313   350   430    35
```

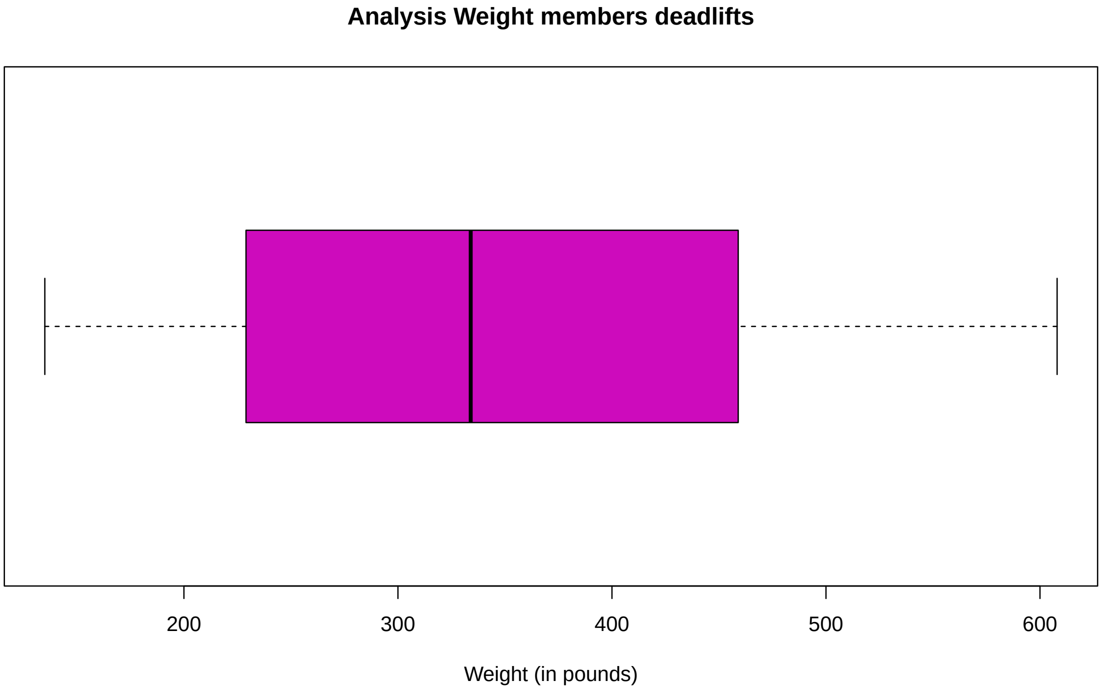
2. Graphical summaries

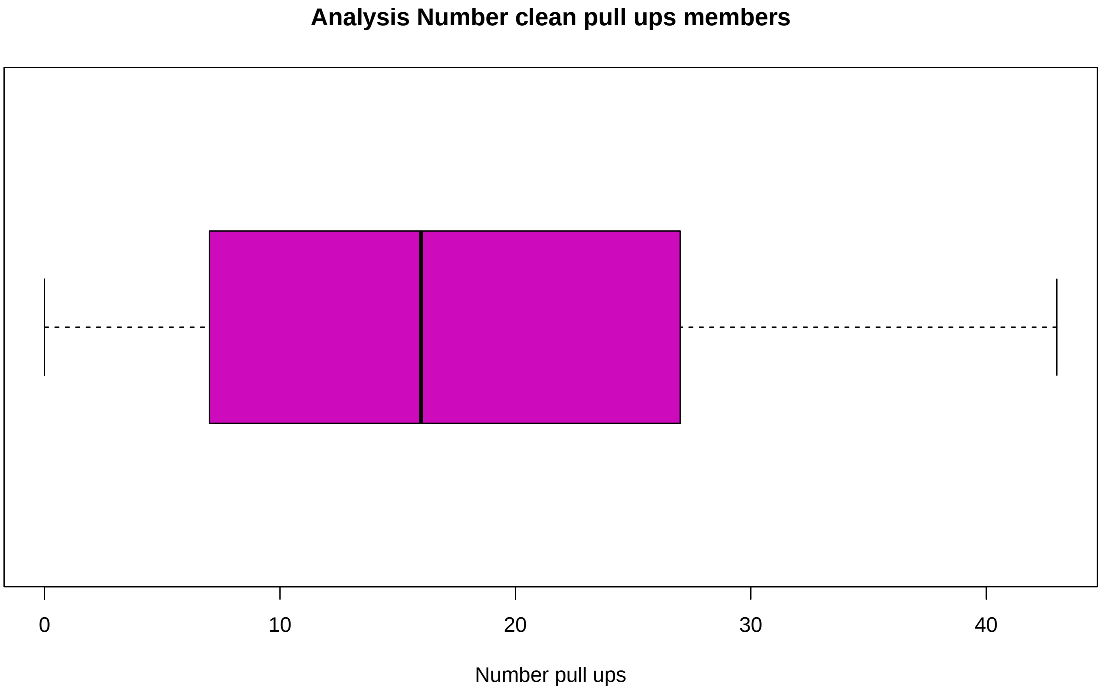
```r
# Creating vector to names for each chart
names_variables_PR <- c("Age of the members",
                        "Weight members bench presses",
                        "Weight members squats",
                        "Weight members deadlifts",
                        "Number clean pull ups members")

# Creating x label for each chart
x_labels_PR <- c("Age in years (to 1 decimal place)",
                 "Weight (in pounds)",
                 "Weight (in pounds)",
                  "Weight (in pounds)",
                  "Number pull ups")

for (i in 1:ncol(gym_data_PR)) {
  boxplot(gym_data_PR[,i],
          main=paste("Analysis", names_variables_PR[i]),
          xlab=x_labels_PR[i],
          horizontal=TRUE,
          pch=20,
          col=6)
}
```




Analysis Weight members bench presses
50
100
150
200
250
300
350
400
Weight (in pounds)


Analysis Weight members squats
100
200
300
400
500
Weight (in pounds)





Observations: In general, there are not outliers in any variable

Aditionally:

- *Age of the members*: The gym members are between ~15 and ~ 69 years old. Most of the gym members are between ~31 and 50 years old, this is the IQR or 50% of the data set.

- *Weight members bench presses*: About member that lift bench presses, there are members who lift heavier weights compared to the median. Most of the gym members lift between 150 and 300 pounds on the bench press.
- *Weight members squats*: The gym members can lift between 100 and ~ 550 pounds. 50% of them lift between 200 and 400 pounds.(IQR - 50% data set).
- *Weight members deadlifts*: About gym members who do deadlifts, they lift approximately between ~230 and ~460 pounds. The median is 320 approximately.
- *Number clean pull ups members*: There are gym members who do until ~43 numbers of pull-ups. However, most of them do between 9 and 28 aprox.

3. Standardize all of the variables using min and max method

```
# Set a function to re-scale data
sta_pr <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}
# standardizing all variables with the same method
gym_data_PR$Age_PR_std <-- sta_pr(gym_data_PR$Age_PR)
gym_data_PR$BP_PR_std <-- sta_pr(gym_data_PR$BP_PR)
gym_data_PR$Sq_PR_std <-- sta_pr(gym_data_PR$Sq_PR)
gym_data_PR$DL_PR_std <-- sta_pr(gym_data_PR$DL_PR)
gym_data_PR$PU_PR_std <-- sta_pr(gym_data_PR$PU_PR)
# printing news standardized variables
head(gym_data_PR)
```

```
##   Age_PR BP_PR Sq_PR DL_PR PU_PR Age_PR_std  BP_PR_std   Sq_PR_std  DL_PR_std
## 1   35.2   398   375   493    39 -0.3609023 -0.9461756 -0.63425926 -0.7568710
## 2   63.7   261   531   514    14 -0.8966165 -0.5580737 -0.99537037 -0.8012685
## 3   23.5   157   134   272     3 -0.1409774 -0.2634561 -0.07638889 -0.2896406
## 4   37.2   160   114   236    10 -0.3984962 -0.2719547 -0.03009259 -0.2135307
## 5   45.9   128   156   354    14 -0.5620301 -0.1813031 -0.12731481 -0.4630021
## 6   47.7   313   350   430    35 -0.5958647 -0.7053824 -0.57638889 -0.6236786
##     PU_PR_std
## 1 -0.90697674
## 2 -0.32558140
## 3 -0.06976744
## 4 -0.23255814
## 5 -0.32558140
## 6 -0.81395349
```

To re-scale the variables I have used the minimum and maximum method (second method) because the data does not have outliers, according the bloxplots.

2. Clustering

1. Create segmentation/cluster schemes for k=2,3,4,5,6,7.

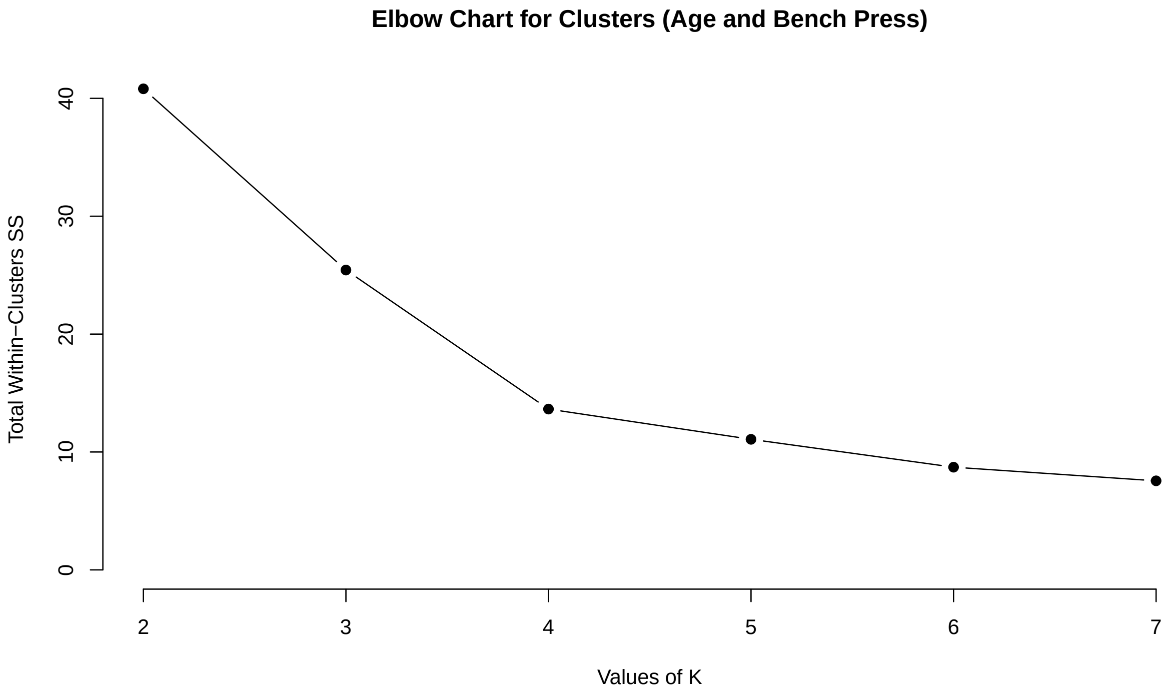
```
# Set Up for Clusters
# Variable for Elbow Chart for k=2,3,4,5,6,7
maxk_pr <- 7
# replicating with 0
wss_pr <- rep(0,maxk_pr-1)
# Creating Clusters with loop to iterate in clusters
for (k_pr in 2:maxk_pr){
  ClstrGym_PR <- kmeans(gym_data_PR[,c(6:7)], iter.max=10, centers=k_pr, nstart=10)
  #put tot.withinss in each # cluster [value in k-1]
  wss_pr[k_pr-1] <- ClstrGym_PR$tot.withinss
}
# SS
wss_pr
```

```
## [1] 40.807267 25.435862 13.641372 11.076715  8.711696  7.554256
```

Creating set up for cluster to ilustrate in a WSS plot

2. Create the WSS plots

```
#Show results in wss plot
plot(2:maxk_pr, wss_pr,
     type="b", pch = 19, frame = FALSE,
     main="Elbow Chart for Clusters (Age and Bench Press)",
     xlab="Values of K",
     ylab="Total Within-Clusters SS",
     ylim=c(0,max(wss_pr)))
```

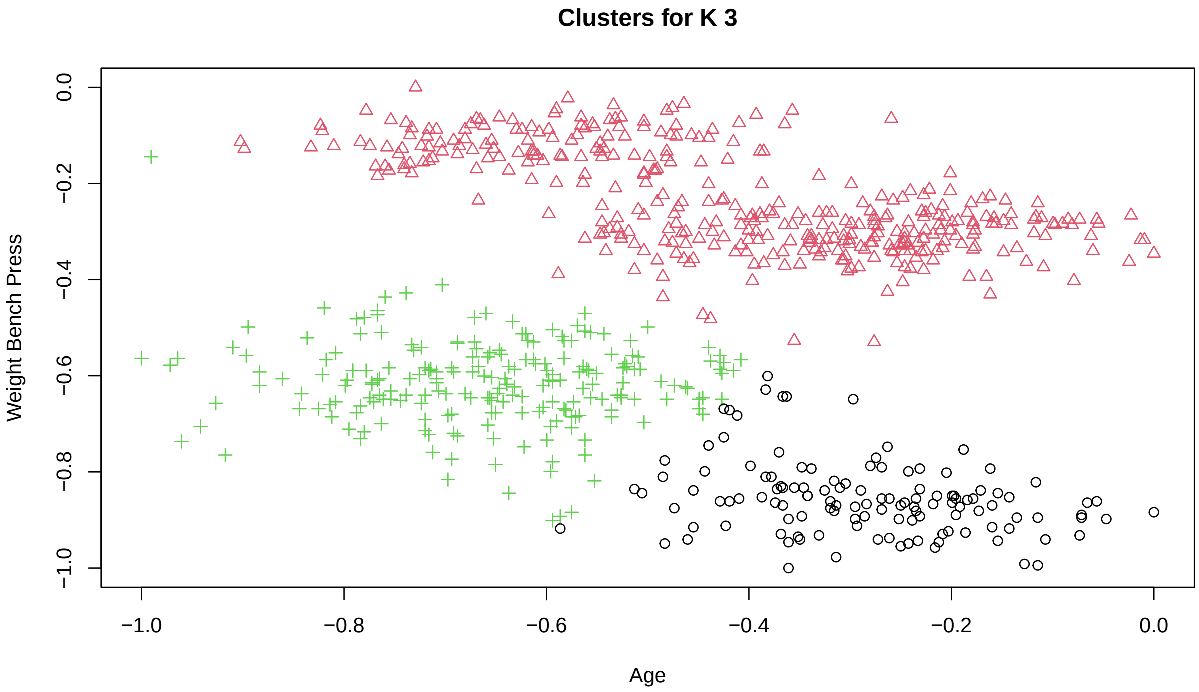


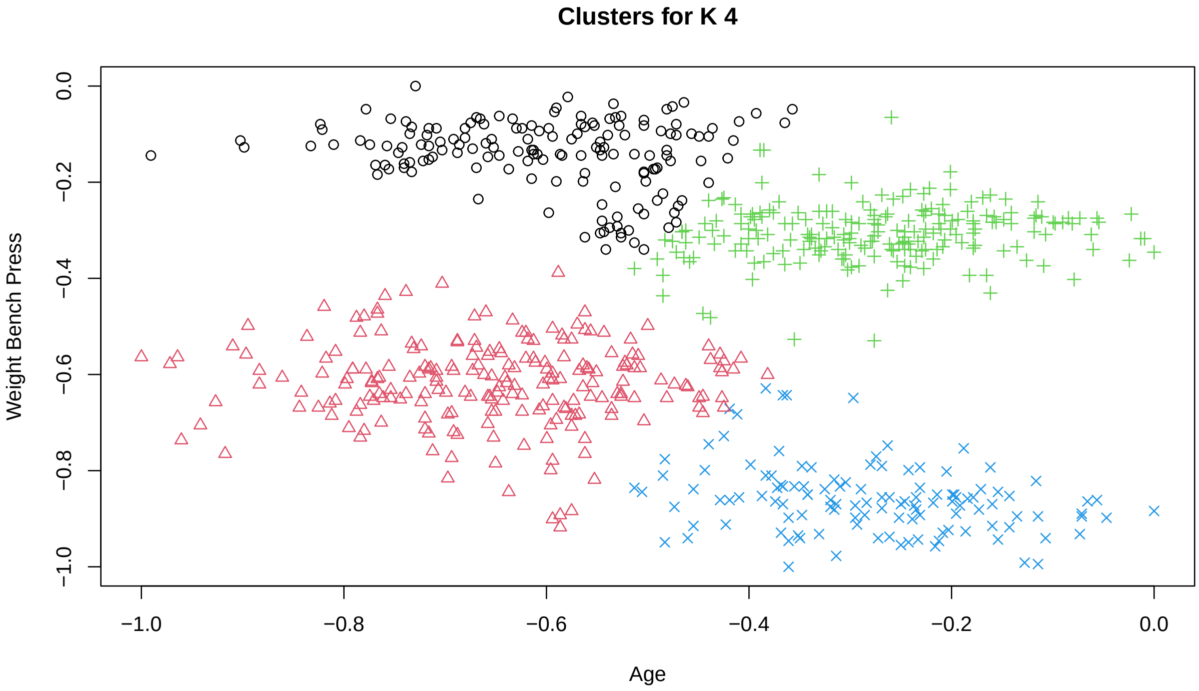
Base on WSS plot I selected 4 clusters, because is where the plot shows the "elbow" point

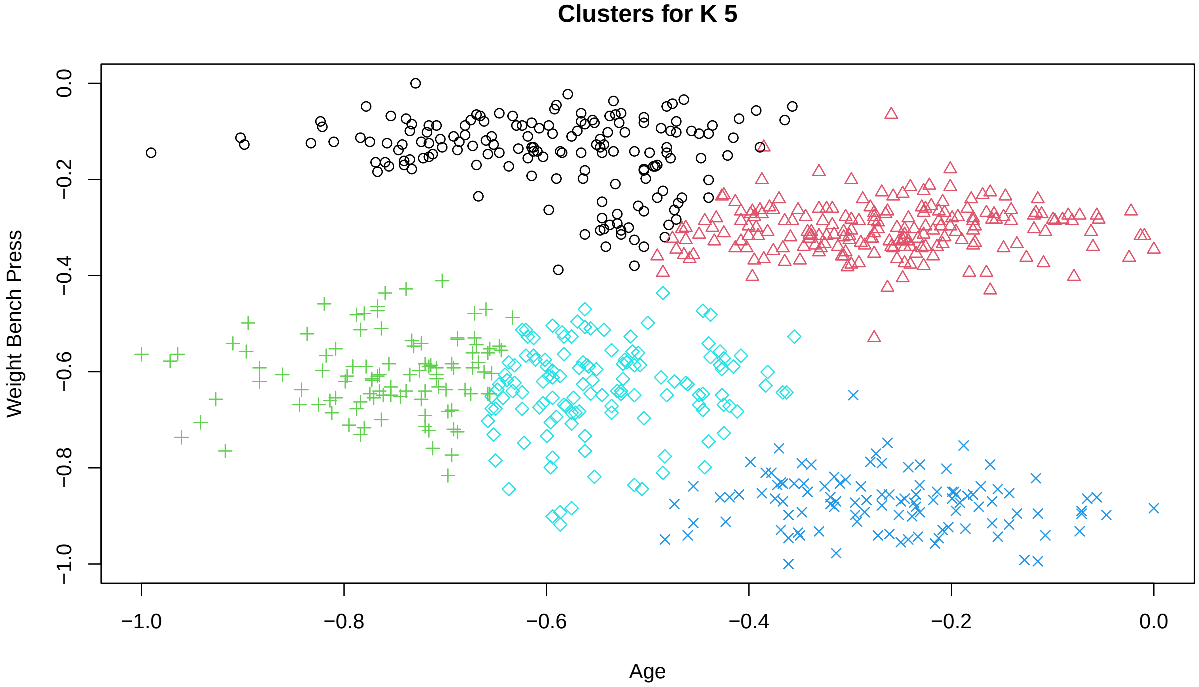
3. Evaluation of Clusters

1. creating a scatter plots for "k-1", "k", "k+1".

```
# Cluster selected = 4
k_selected_pr = 4
#Loop to iterate in k-1, k and k+1
for (k_pr in (k_selected_pr-1):(k_selected_pr+1)){
  ClstrGym_PR <- kmeans(gym_data_PR[,c(6:7)], iter.max=10, centers=k_pr, nstart=10)
  gym_data_PR$Cluster_PR <- factor(ClstrGym_PR$cluster)

#Creating charts for each # of clusters
plot(gym_data_PR$Age_PR_std, gym_data_PR$BP_PR_std,
     col=gym_data_PR$Cluster_PR, pch=as.numeric(gym_data_PR$Cluster_PR),
     main=paste("Clusters for K",k_pr), xlab="Age", ylab="Weight Bench Press"
     )

}
```




Clusters for K 4


Clusters for K 5

```
head(gym_data_PR)
```

```
##   Age_PR BP_PR Sq_PR DL_PR PU_PR Age_PR_std  BP_PR_std   Sq_PR_std  DL_PR_std
## 1   35.2   398   375   493    39 -0.3609023 -0.9461756 -0.63425926 -0.7568710
## 2   63.7   261   531   514    14 -0.8966165 -0.5580737 -0.99537037 -0.8012685
## 3   23.5   157   134   272     3 -0.1409774 -0.2634561 -0.07638889 -0.2896406
## 4   37.2   160   114   236    10 -0.3984962 -0.2719547 -0.03009259 -0.2135307
## 5   45.9   128   156   354    14 -0.5620301 -0.1813031 -0.12731481 -0.4630021
## 6   47.7   313   350   430    35 -0.5958647 -0.7053824 -0.57638889 -0.6236786
##     PU_PR_std Cluster_PR
## 1 -0.90697674          4
## 2 -0.32558140          3
## 3 -0.06976744          2
## 4 -0.23255814          2
## 5 -0.32558140          1
## 6 -0.81395349          5
```

2. choose the best set of cluster that describes the data.

Based on the WSS plot and the cluster charts, I think the set of clusters K=4 represents a more suitable grouping, showing the information in a short of quadrant were differences in both weight and age are evident to the eye.

With cluster K = 3, the highest bench press weights are grouped together without considering the age.

Finally, the clusters K=5 divides the lower weight in three groups, while dividing the higher weight in only two groups, making it harder to make conclusions and take decisions.

3. Summary cluster.

```r
# data with only 4 cluster based on my previos reflection
ClstrGym_PR <- kmeans(gym_data_PR[,c(6:7)], iter.max=10, centers=4, nstart=10)
gym_data_PR$Cluster_PR <- factor(ClstrGym_PR$cluster)
#summary table for the segmentation/clustering scheme = 4
SummClusters <- aggregate(cbind(Age_PR, BP_PR, Sq_PR, DL_PR, PU_PR) ~ Cluster_PR,
                          gym_data_PR, FUN=function(x) round(mean(x), 0))
# result
SummClusters
```

```
##   Cluster_PR Age_PR BP_PR Sq_PR DL_PR PU_PR
## 1          1     31   367   428   502    30
## 2          2     48   115   189   210     6
## 3          3     30   172   231   276    13
## 4          4     51   282   380   427    25
```

Creating the summary table for 4 clusters

4. Descriptive names for each cluster.

Custer 1: Young Avg Performance

Custer 2: Mature High Performance

Custer 3: Young High Performance

Custer 4: Mature Low Performance

5. Possible uses for this clustering scheme

With this classification, it is possible creating training plans or memberships. Gym members could be segmented into groups with different necessities and physical abilities.

References

Conestoga College. (2024). PROG8435 – Data Analysis, Modeling and Algorithms - LECTURE 6 – UNSUPERVISED LEARNING:K-MEANS CLUSTERING [PowerPoint slides]. eConestoga.