FDA\_Sanket\_Assignment\_2

Sanket Patil

2023-10-08

# Problem 1

# Importing data in R

Data = read.csv("BankData.csv", header = T)  
  
dim(Data)

## [1] 690 13

Data <- na.omit(Data)  
dim(Data)

## [1] 666 13

# Question 1 :

# Applying Visualization techniques to create graphs..

# Checking class of all variables

lapply(Data,class)

## $X  
## [1] "integer"  
##   
## $cont1  
## [1] "numeric"  
##   
## $cont2  
## [1] "numeric"  
##   
## $cont3  
## [1] "numeric"  
##   
## $bool1  
## [1] "character"  
##   
## $bool2  
## [1] "character"  
##   
## $cont4  
## [1] "integer"  
##   
## $bool3  
## [1] "character"  
##   
## $cont5  
## [1] "integer"  
##   
## $cont6  
## [1] "integer"  
##   
## $approval  
## [1] "character"  
##   
## $credit.score  
## [1] "numeric"  
##   
## $ages  
## [1] "integer"

ls(Data)

## [1] "ages" "approval" "bool1" "bool2" "bool3"   
## [6] "cont1" "cont2" "cont3" "cont4" "cont5"   
## [11] "cont6" "credit.score" "X"

# Separating all numeric variables into another dataframe

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

num\_data <- select\_if(Data, is.numeric)  
ls(num\_data)

## [1] "ages" "cont1" "cont2" "cont3" "cont4"   
## [6] "cont5" "cont6" "credit.score" "X"

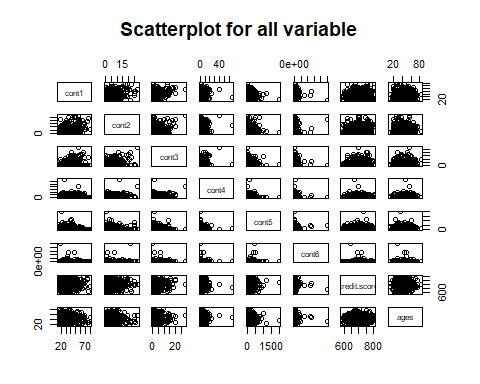
# Verifying class of all numeric variables

lapply(num\_data,class)

## $X  
## [1] "integer"  
##   
## $cont1  
## [1] "numeric"  
##   
## $cont2  
## [1] "numeric"  
##   
## $cont3  
## [1] "numeric"  
##   
## $cont4  
## [1] "integer"  
##   
## $cont5  
## [1] "integer"  
##   
## $cont6  
## [1] "integer"  
##   
## $credit.score  
## [1] "numeric"  
##   
## $ages  
## [1] "integer"

# Creating Scatter plot matrix for all numeric variables

pairs(~cont1+cont2+cont3+cont4+cont5+cont6+credit.score+ages,data = Data,main = "Scatterplot for all variable")

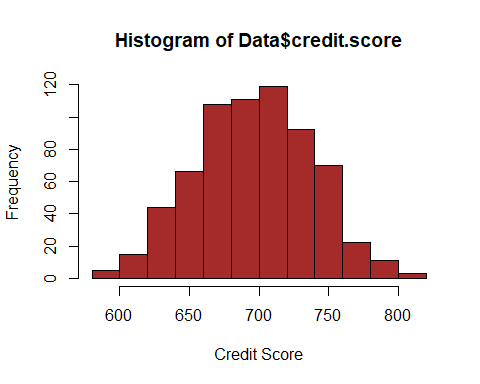
 # By looking at the scatterplot matrix, we can say there is no linear relationship between any of the numeric variable.

library(ggplot2)  
library(tidyverse)

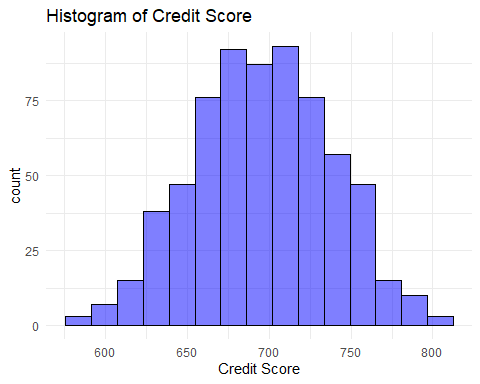
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.0  
## ✔ readr 2.1.4   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Creating Histogram..

hist(Data$credit.score,xlab = "Credit Score",col = "brown",border = "black")

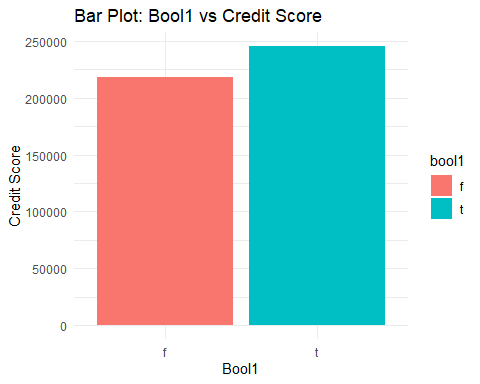


Data %>% ggplot(aes(credit.score)) +  
 geom\_histogram(fill = "blue", bins = 15, color = "black", alpha = 0.5) +  
 labs( x = "Credit Score", title = "Histogram of Credit Score") +  
 theme\_minimal()

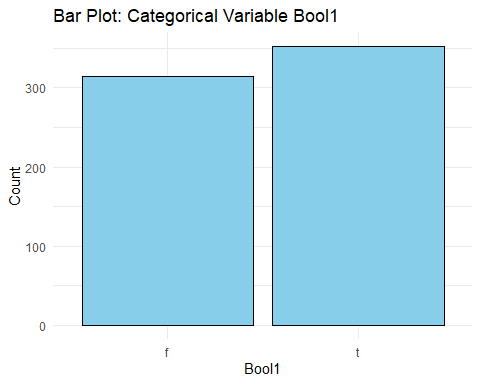
 # Used two methods to create histogram. By looking at the histogram, we can see that distribution is normal. We can see a perfect bell curve

# Creating a Bar plot

ggplot(Data, aes(x = bool1, y = credit.score, fill = bool1)) +  
 geom\_bar(stat = "identity") +  
 labs(x = "Bool1", y = "Credit Score", title = "Bar Plot: Bool1 vs Credit Score") +  
 theme\_minimal()



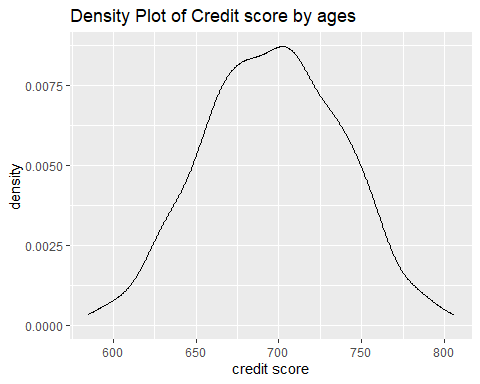
ggplot(Data, aes(x = bool1)) +  
 geom\_bar(fill = "skyblue", color = "black") +  
 labs(x = "Bool1", y = "Count", title = "Bar Plot: Categorical Variable Bool1") +  
 theme\_minimal()

 # By looking at the bar plot, we can see that the the category distribution is almost similar. There is no biased category. # Also there are more number of records for True as compared to False.

# Creating Density plot

ggplot(Data, aes(x = credit.score, fill = ages)) +  
 geom\_density(alpha = 0.5) +  
 labs(title = "Density Plot of Credit score by ages", x = "credit score", fill = "Credit Score")

## Warning: The following aesthetics were dropped during statistical transformation: fill  
## ℹ This can happen when ggplot fails to infer the correct grouping structure in  
## the data.  
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical  
## variable into a factor?



# Question 2 :

# Z-score normalization

Normalized\_cont1 <- scale(Data$cont1)

# Z-score normalization

Normalized\_cont1 = (Data$cont1 - mean(Data$cont1)) / sd(Data$cont1)

# Old Min max before Z-score normalization

max(Data$cont1)

## [1] 80.25

min(Data$cont1)

## [1] 13.75

# New min max values after Z-score normalization

max(Normalized\_cont1)

## [1] 4.083912

min(Normalized\_cont1)

## [1] -1.494865

# Min-max normalization

# Min-max values before normalization

max(Data$cont2)

## [1] 28

min(Data$cont2)

## [1] 0

Normalized\_cont2 = (Data$cont2 - min(Data$cont2)) / (max(Data$cont2) - min(Data$cont2))

# Min-max values after normalization

max(Normalized\_cont2)

## [1] 1

min(Normalized\_cont2)

## [1] 0

# Decimal scaling normalization

max(Data$cont3)

## [1] 28.5

decimal\_places<-2   
  
Normalized\_cont3 <- Data$cont3 / 10^decimal\_places

# Min-max values before normalization

min(Data$cont3)

## [1] 0

max(Data$cont3)

## [1] 28.5

# Min-max values after normalization

min(Normalized\_cont3)

## [1] 0

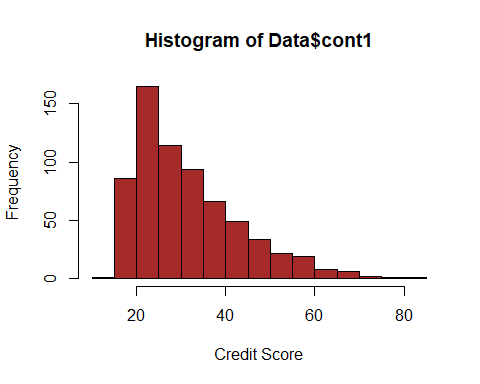
max(Normalized\_cont3)

## [1] 0.285

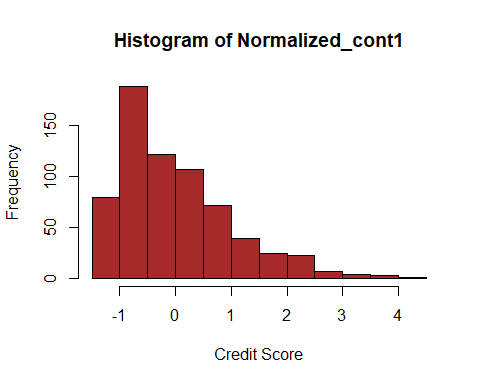
# Question 3:

# Z-score Normalization :

hist(Data$cont1,xlab = "Credit Score",col = "brown",border = "black")



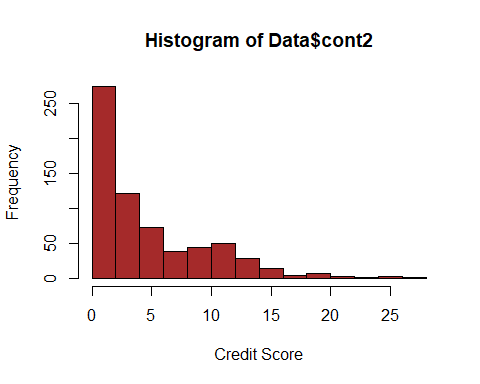
hist(Normalized\_cont1,xlab = "Credit Score",col = "brown",border = "black")

 # After Z-score normalization for the variable count1, there is change in min and max value but there is no change in normality of the data as we can see histogram looks similar as it was previously.

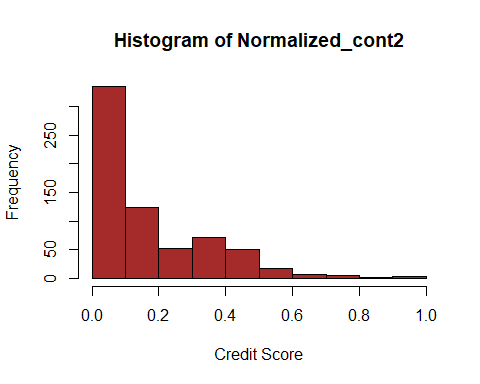
# Decimal Normalization :

# Histogram before normalization

hist(Data$cont2,xlab = "Credit Score",col = "brown",border = "black")

 # Histogram after normalization

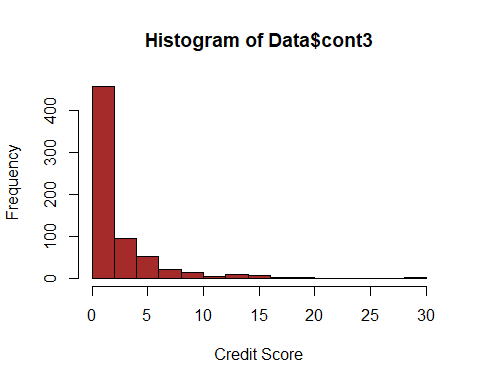
hist(Normalized\_cont2,xlab = "Credit Score",col = "brown",border = "black")

 # After min-max normalization for the variable count2, there is change in min and max value but there is no change in normality of the data as we can see histogram looks similar as it was previously.

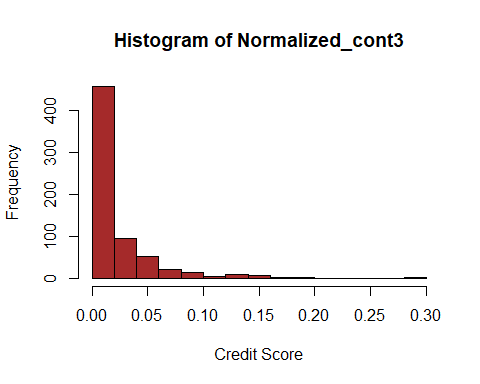
# Decimal Scaling :

# Histogram before normalization

hist(Data$cont3,xlab = "Credit Score",col = "brown",border = "black")

 # Histogram after normalization

hist(Normalized\_cont3,xlab = "Credit Score",col = "brown",border = "black")

 # After Decimal scaling for the variable count3, there is change in min and max value but there is no change in normality of the data as we can see histogram looks similar as it was previously.

# Question 4 :

# Equal depth (quantile) binning

Data$credit.score\_bins <- cut(Data$credit.score, breaks=quantile(Data$credit.score, probs=0:3/3), labels=c("Low", "Medium", "High"), include.lowest=TRUE)  
  
head(Data)

## X cont1 cont2 cont3 bool1 bool2 cont4 bool3 cont5 cont6 approval credit.score  
## 1 1 30.83 0.000 1.25 t t 1 f 202 0 + 664.60  
## 2 2 58.67 4.460 3.04 t t 6 f 43 560 + 693.88  
## 3 3 24.50 0.500 1.50 t f 0 f 280 824 + 621.82  
## 4 4 27.83 1.540 3.75 t t 5 t 100 3 + 653.97  
## 5 5 20.17 5.625 1.71 t f 0 f 120 0 + 670.26  
## 6 6 32.08 4.000 2.50 t f 0 t 360 0 + 672.16  
## ages credit.score\_bins  
## 1 42 Low  
## 2 54 Medium  
## 3 29 Low  
## 4 58 Low  
## 5 65 Low  
## 6 61 Low

# We have chosen variable credit.score. By looking at the distribution of the data, we decided to choose equal depth binning technique as data is normally distributed.

# Question 5 :

# To calculate mean of each bin

mean\_bins <- tapply(Data$credit.score, Data$credit.score\_bins, mean)

# Replace original values with bin mean

Data$Smoothed\_credit.score <- as.numeric(mean\_bins[Data$credit.score\_bins])  
  
head(Data)

## X cont1 cont2 cont3 bool1 bool2 cont4 bool3 cont5 cont6 approval credit.score  
## 1 1 30.83 0.000 1.25 t t 1 f 202 0 + 664.60  
## 2 2 58.67 4.460 3.04 t t 6 f 43 560 + 693.88  
## 3 3 24.50 0.500 1.50 t f 0 f 280 824 + 621.82  
## 4 4 27.83 1.540 3.75 t t 5 t 100 3 + 653.97  
## 5 5 20.17 5.625 1.71 t f 0 f 120 0 + 670.26  
## 6 6 32.08 4.000 2.50 t f 0 t 360 0 + 672.16  
## ages credit.score\_bins Smoothed\_credit.score  
## 1 42 Low 650.0637  
## 2 54 Medium 696.2586  
## 3 29 Low 650.0637  
## 4 58 Low 650.0637  
## 5 65 Low 650.0637  
## 6 61 Low 650.0637

# Problem 2 :

# Reading data in R

df = read.csv("BankData.csv", header = T)  
head(df)

## X cont1 cont2 cont3 bool1 bool2 cont4 bool3 cont5 cont6 approval credit.score  
## 1 1 30.83 0.000 1.25 t t 1 f 202 0 + 664.60  
## 2 2 58.67 4.460 3.04 t t 6 f 43 560 + 693.88  
## 3 3 24.50 0.500 1.50 t f 0 f 280 824 + 621.82  
## 4 4 27.83 1.540 3.75 t t 5 t 100 3 + 653.97  
## 5 5 20.17 5.625 1.71 t f 0 f 120 0 + 670.26  
## 6 6 32.08 4.000 2.50 t f 0 t 360 0 + 672.16  
## ages  
## 1 42  
## 2 54  
## 3 29  
## 4 58  
## 5 65  
## 6 61

dim(df)

## [1] 690 13

# Removing rows which has NA values

df <- na.omit(df)  
dim(df)

## [1] 666 13

#Load the required library

library(e1071)  
  
unique(df$approval)

## [1] "+" "-"

# Replacing the values of target to 1 and 0.

df$approval <- as.factor(df$approval)  
head(df)

## X cont1 cont2 cont3 bool1 bool2 cont4 bool3 cont5 cont6 approval credit.score  
## 1 1 30.83 0.000 1.25 t t 1 f 202 0 + 664.60  
## 2 2 58.67 4.460 3.04 t t 6 f 43 560 + 693.88  
## 3 3 24.50 0.500 1.50 t f 0 f 280 824 + 621.82  
## 4 4 27.83 1.540 3.75 t t 5 t 100 3 + 653.97  
## 5 5 20.17 5.625 1.71 t f 0 f 120 0 + 670.26  
## 6 6 32.08 4.000 2.50 t f 0 t 360 0 + 672.16  
## ages  
## 1 42  
## 2 54  
## 3 29  
## 4 58  
## 5 65  
## 6 61

# Train the SVM model

svm\_model <- svm(approval ~ ., data = df, kernel = "linear", cost = 1, cross = 10)

# Print the model details

print(svm\_model)

##   
## Call:  
## svm(formula = approval ~ ., data = df, kernel = "linear", cost = 1,   
## cross = 10)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 1   
##   
## Number of Support Vectors: 195

# Access cross-validation results

summary <- summary(svm\_model)  
print(summary)

##   
## Call:  
## svm(formula = approval ~ ., data = df, kernel = "linear", cost = 1,   
## cross = 10)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 1   
##   
## Number of Support Vectors: 195  
##   
## ( 101 94 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## - +  
##   
## 10-fold cross-validation on training data:  
##   
## Total Accuracy: 86.33634   
## Single Accuracies:  
## 87.87879 86.56716 87.87879 89.55224 83.58209 89.39394 80.59701 77.27273 88.0597 92.53731

# As this is a regression model, accuracy might not be a most appropriate approach. Hence we evaluated Correlation coefficient and MSE.

# The squared correlation coefficient suggests a moderate amount of relationship between predicted and actual values, and the model seems to have a reasonably low mean squared error.

# Question b.

#Use grid search to optimize the C parameter # Define a grid of C values to search

c\_values <- 10^(-2:2)

# Perform grid search with cross-validation

svm\_tune <- tune(svm, approval ~ ., data = df, kernel = "linear", ranges = list(cost = c\_values), tunecontrol = tune.control(sampling = "cross", cross = 10))

# Print the best parameter and its corresponding accuracy

C\_parameter <- svm\_tune$best.parameters$cost  
accuracy <- svm\_tune$best.performance

# Chosen C parameter for the model is 0.01.

# Best accuracy for the model is 0.1147595.

# Question c:

The results are different even if the value of C=1. This is because of below mentioned reasons. The folds can be different between the two runs, leading to slightly different training and validation sets. For every run, SVM may change its local minima hence SVM might converge differently. Also other hyperparameters can change differently for every iteration. To resolve this issue, we can use set.seed function. This function will reproduce any operation which was performed with randomness.

# Problem 3 :

# Install and load necessary libraries

library(dplyr)  
library(e1071)  
library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

# Loading starwars dataset

starwars <- dplyr::starwars  
colnames(starwars)

## [1] "name" "height" "mass" "hair\_color" "skin\_color"  
## [6] "eye\_color" "birth\_year" "sex" "gender" "homeworld"   
## [11] "species" "films" "vehicles" "starships"

# Removing variables given in problem statement

starwars <- starwars %>% select(-films, -vehicles, -starships, -name)  
colnames(starwars)

## [1] "height" "mass" "hair\_color" "skin\_color" "eye\_color"   
## [6] "birth\_year" "sex" "gender" "homeworld" "species"

# Dimensions before removing rows with NA

dim(starwars)

## [1] 87 10

# Removing rows which has NA values

starwars <- na.omit(starwars)

# Dimensions after removing rows with NA

dim(starwars)

## [1] 29 10

# Question a:

#Creating dummy variables except variable “gender”:

starwars\_dummies <- model.matrix(~. - gender, data = starwars)[, -1]  
  
gender<-starwars$gender

# Merging dummy variables with gender variable

starwars\_new <- cbind.data.frame(starwars\_dummies, gender=gender)  
  
colnames(starwars\_new)

## [1] "height" "mass" "hair\_colorblack"   
## [4] "hair\_colorblond" "hair\_colorbrown" "hair\_colorbrown, grey"   
## [7] "hair\_colorgrey" "hair\_colornone" "hair\_colorwhite"   
## [10] "skin\_colorbrown" "skin\_colorbrown mottle" "skin\_colordark"   
## [13] "skin\_colorfair" "skin\_colorgreen" "skin\_colorlight"   
## [16] "skin\_colororange" "skin\_colorpale" "skin\_colorred"   
## [19] "skin\_colortan" "skin\_colorunknown" "skin\_colorwhite"   
## [22] "skin\_coloryellow" "eye\_colorblue" "eye\_colorblue-gray"   
## [25] "eye\_colorbrown" "eye\_colorhazel" "eye\_colororange"   
## [28] "eye\_colorred" "eye\_coloryellow" "birth\_year"   
## [31] "sexmale" "homeworldBespin" "homeworldCerea"   
## [34] "homeworldConcord Dawn" "homeworldCorellia" "homeworldDathomir"   
## [37] "homeworldDorin" "homeworldEndor" "homeworldHaruun Kal"   
## [40] "homeworldKamino" "homeworldKashyyyk" "homeworldMirial"   
## [43] "homeworldMon Cala" "homeworldNaboo" "homeworldRyloth"   
## [46] "homeworldSerenno" "homeworldSocorro" "homeworldStewjon"   
## [49] "homeworldTatooine" "homeworldTrandosha" "speciesEwok"   
## [52] "speciesGungan" "speciesHuman" "speciesKel Dor"   
## [55] "speciesMirialan" "speciesMon Calamari" "speciesTrandoshan"   
## [58] "speciesTwi'lek" "speciesWookiee" "speciesZabrak"   
## [61] "gender"

dim(starwars\_new)

## [1] 29 61

summary(starwars\_new)

## height mass hair\_colorblack hair\_colorblond   
## Min. : 88 Min. : 20.00 Min. :0.0000 Min. :0.00000   
## 1st Qu.:170 1st Qu.: 75.00 1st Qu.:0.0000 1st Qu.:0.00000   
## Median :180 Median : 79.00 Median :0.0000 Median :0.00000   
## Mean :178 Mean : 77.77 Mean :0.2069 Mean :0.06897   
## 3rd Qu.:188 3rd Qu.: 83.00 3rd Qu.:0.0000 3rd Qu.:0.00000   
## Max. :228 Max. :136.00 Max. :1.0000 Max. :1.00000   
## hair\_colorbrown hair\_colorbrown, grey hair\_colorgrey hair\_colornone   
## Min. :0.0000 Min. :0.00000 Min. :0.00000 Min. :0.0000   
## 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.0000   
## Median :0.0000 Median :0.00000 Median :0.00000 Median :0.0000   
## Mean :0.2414 Mean :0.03448 Mean :0.03448 Mean :0.3103   
## 3rd Qu.:0.0000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.00000 Max. :1.00000 Max. :1.0000   
## hair\_colorwhite skin\_colorbrown skin\_colorbrown mottle skin\_colordark   
## Min. :0.00000 Min. :0.00000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :0.00000 Median :0.00000 Median :0.00000 Median :0.00000   
## Mean :0.06897 Mean :0.03448 Mean :0.03448 Mean :0.06897   
## 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.00000   
## skin\_colorfair skin\_colorgreen skin\_colorlight skin\_colororange   
## Min. :0.0000 Min. :0.00000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.00000   
## Median :0.0000 Median :0.00000 Median :0.0000 Median :0.00000   
## Mean :0.2414 Mean :0.03448 Mean :0.2069 Mean :0.06897   
## 3rd Qu.:0.0000 3rd Qu.:0.00000 3rd Qu.:0.0000 3rd Qu.:0.00000   
## Max. :1.0000 Max. :1.00000 Max. :1.0000 Max. :1.00000   
## skin\_colorpale skin\_colorred skin\_colortan skin\_colorunknown  
## Min. :0.00000 Min. :0.00000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :0.00000 Median :0.00000 Median :0.00000 Median :0.00000   
## Mean :0.06897 Mean :0.03448 Mean :0.03448 Mean :0.03448   
## 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.00000   
## skin\_colorwhite skin\_coloryellow eye\_colorblue eye\_colorblue-gray  
## Min. :0.00000 Min. :0.00000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.00000   
## Median :0.00000 Median :0.00000 Median :0.0000 Median :0.00000   
## Mean :0.03448 Mean :0.06897 Mean :0.2759 Mean :0.03448   
## 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:1.0000 3rd Qu.:0.00000   
## Max. :1.00000 Max. :1.00000 Max. :1.0000 Max. :1.00000   
## eye\_colorbrown eye\_colorhazel eye\_colororange eye\_colorred   
## Min. :0.0000 Min. :0.00000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :0.0000 Median :0.00000 Median :0.00000 Median :0.00000   
## Mean :0.3448 Mean :0.06897 Mean :0.06897 Mean :0.03448   
## 3rd Qu.:1.0000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :1.0000 Max. :1.00000 Max. :1.00000 Max. :1.00000   
## eye\_coloryellow birth\_year sexmale homeworldBespin   
## Min. :0.0000 Min. : 8.00 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.0000 1st Qu.: 31.00 1st Qu.:1.0000 1st Qu.:0.00000   
## Median :0.0000 Median : 46.00 Median :1.0000 Median :0.00000   
## Mean :0.1379 Mean : 51.29 Mean :0.7931 Mean :0.03448   
## 3rd Qu.:0.0000 3rd Qu.: 57.00 3rd Qu.:1.0000 3rd Qu.:0.00000   
## Max. :1.0000 Max. :200.00 Max. :1.0000 Max. :1.00000   
## homeworldCerea homeworldConcord Dawn homeworldCorellia homeworldDathomir  
## Min. :0.00000 Min. :0.00000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :0.00000 Median :0.00000 Median :0.00000 Median :0.00000   
## Mean :0.03448 Mean :0.03448 Mean :0.06897 Mean :0.03448   
## 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.00000   
## homeworldDorin homeworldEndor homeworldHaruun Kal homeworldKamino   
## Min. :0.00000 Min. :0.00000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :0.00000 Median :0.00000 Median :0.00000 Median :0.00000   
## Mean :0.03448 Mean :0.03448 Mean :0.03448 Mean :0.03448   
## 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.00000   
## homeworldKashyyyk homeworldMirial homeworldMon Cala homeworldNaboo   
## Min. :0.00000 Min. :0.00000 Min. :0.00000 Min. :0.0000   
## 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.0000   
## Median :0.00000 Median :0.00000 Median :0.00000 Median :0.0000   
## Mean :0.03448 Mean :0.06897 Mean :0.03448 Mean :0.1034   
## 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.0000   
## Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.0000   
## homeworldRyloth homeworldSerenno homeworldSocorro homeworldStewjon   
## Min. :0.00000 Min. :0.00000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :0.00000 Median :0.00000 Median :0.00000 Median :0.00000   
## Mean :0.03448 Mean :0.03448 Mean :0.03448 Mean :0.03448   
## 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.00000   
## homeworldTatooine homeworldTrandosha speciesEwok speciesGungan   
## Min. :0.0000 Min. :0.00000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :0.0000 Median :0.00000 Median :0.00000 Median :0.00000   
## Mean :0.2069 Mean :0.03448 Mean :0.03448 Mean :0.03448   
## 3rd Qu.:0.0000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :1.0000 Max. :1.00000 Max. :1.00000 Max. :1.00000   
## speciesHuman speciesKel Dor speciesMirialan speciesMon Calamari  
## Min. :0.0000 Min. :0.00000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :1.0000 Median :0.00000 Median :0.00000 Median :0.00000   
## Mean :0.6207 Mean :0.03448 Mean :0.06897 Mean :0.03448   
## 3rd Qu.:1.0000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :1.0000 Max. :1.00000 Max. :1.00000 Max. :1.00000   
## speciesTrandoshan speciesTwi'lek speciesWookiee speciesZabrak   
## Min. :0.00000 Min. :0.00000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :0.00000 Median :0.00000 Median :0.00000 Median :0.00000   
## Mean :0.03448 Mean :0.03448 Mean :0.03448 Mean :0.03448   
## 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.00000   
## gender   
## Length:29   
## Class :character   
## Mode :character   
##   
##   
##

# Note that after merging gender with dummies data, all variables converted to class character. We need to convert those variables into numeric data.

# Head of new data after creating dummy variables except gender

head(starwars\_new)

## height mass hair\_colorblack hair\_colorblond hair\_colorbrown  
## 1 172 77 0 1 0  
## 2 202 136 0 0 0  
## 3 150 49 0 0 1  
## 4 178 120 0 0 0  
## 5 165 75 0 0 1  
## 6 183 84 1 0 0  
## hair\_colorbrown, grey hair\_colorgrey hair\_colornone hair\_colorwhite  
## 1 0 0 0 0  
## 2 0 0 1 0  
## 3 0 0 0 0  
## 4 1 0 0 0  
## 5 0 0 0 0  
## 6 0 0 0 0  
## skin\_colorbrown skin\_colorbrown mottle skin\_colordark skin\_colorfair  
## 1 0 0 0 1  
## 2 0 0 0 0  
## 3 0 0 0 0  
## 4 0 0 0 0  
## 5 0 0 0 0  
## 6 0 0 0 0  
## skin\_colorgreen skin\_colorlight skin\_colororange skin\_colorpale skin\_colorred  
## 1 0 0 0 0 0  
## 2 0 0 0 0 0  
## 3 0 1 0 0 0  
## 4 0 1 0 0 0  
## 5 0 1 0 0 0  
## 6 0 1 0 0 0  
## skin\_colortan skin\_colorunknown skin\_colorwhite skin\_coloryellow  
## 1 0 0 0 0  
## 2 0 0 1 0  
## 3 0 0 0 0  
## 4 0 0 0 0  
## 5 0 0 0 0  
## 6 0 0 0 0  
## eye\_colorblue eye\_colorblue-gray eye\_colorbrown eye\_colorhazel  
## 1 1 0 0 0  
## 2 0 0 0 0  
## 3 0 0 1 0  
## 4 1 0 0 0  
## 5 1 0 0 0  
## 6 0 0 1 0  
## eye\_colororange eye\_colorred eye\_coloryellow birth\_year sexmale  
## 1 0 0 0 19.0 1  
## 2 0 0 1 41.9 1  
## 3 0 0 0 19.0 0  
## 4 0 0 0 52.0 1  
## 5 0 0 0 47.0 0  
## 6 0 0 0 24.0 1  
## homeworldBespin homeworldCerea homeworldConcord Dawn homeworldCorellia  
## 1 0 0 0 0  
## 2 0 0 0 0  
## 3 0 0 0 0  
## 4 0 0 0 0  
## 5 0 0 0 0  
## 6 0 0 0 0  
## homeworldDathomir homeworldDorin homeworldEndor homeworldHaruun Kal  
## 1 0 0 0 0  
## 2 0 0 0 0  
## 3 0 0 0 0  
## 4 0 0 0 0  
## 5 0 0 0 0  
## 6 0 0 0 0  
## homeworldKamino homeworldKashyyyk homeworldMirial homeworldMon Cala  
## 1 0 0 0 0  
## 2 0 0 0 0  
## 3 0 0 0 0  
## 4 0 0 0 0  
## 5 0 0 0 0  
## 6 0 0 0 0  
## homeworldNaboo homeworldRyloth homeworldSerenno homeworldSocorro  
## 1 0 0 0 0  
## 2 0 0 0 0  
## 3 0 0 0 0  
## 4 0 0 0 0  
## 5 0 0 0 0  
## 6 0 0 0 0  
## homeworldStewjon homeworldTatooine homeworldTrandosha speciesEwok  
## 1 0 1 0 0  
## 2 0 1 0 0  
## 3 0 0 0 0  
## 4 0 1 0 0  
## 5 0 1 0 0  
## 6 0 1 0 0  
## speciesGungan speciesHuman speciesKel Dor speciesMirialan speciesMon Calamari  
## 1 0 1 0 0 0  
## 2 0 1 0 0 0  
## 3 0 1 0 0 0  
## 4 0 1 0 0 0  
## 5 0 1 0 0 0  
## 6 0 1 0 0 0  
## speciesTrandoshan speciesTwi'lek speciesWookiee speciesZabrak gender  
## 1 0 0 0 0 masculine  
## 2 0 0 0 0 masculine  
## 3 0 0 0 0 feminine  
## 4 0 0 0 0 masculine  
## 5 0 0 0 0 feminine  
## 6 0 0 0 0 masculine

# Question b:

# Replacing values from gender variable to 1 and

starwars\_new$gender <- as.factor(starwars\_new$gender)

# Split the data into training and testing

set.seed(123)  
train\_indices <- sample(seq\_len(nrow(starwars\_new)), size = 0.80 \* nrow(starwars\_new))

# Create training and testing datasets

train\_data <- starwars\_new[train\_indices, ]  
test\_data <- starwars\_new[-train\_indices, ]

# Applying svm model

svm\_model <- svm(gender ~ ., data = train\_data)

## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'hair\_colorgrey' and 'skin\_colorpale' and 'skin\_colortan' and  
## 'skin\_colorwhite' and 'homeworldCerea' and 'X.homeworldConcord.Dawn.' and  
## 'homeworldDorin' and 'X.speciesKel.Dor.' constant. Cannot scale data.

svm\_predictions <- predict(svm\_model, test\_data)

# Accuracy

accuracy <- mean(svm\_predictions == test\_data$gender)  
print(accuracy)

## [1] 1

# Confusion Matrix

confusionMatrix(svm\_predictions, test\_data$gender)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction feminine masculine  
## feminine 0 0  
## masculine 0 6  
##   
## Accuracy : 1   
## 95% CI : (0.5407, 1)  
## No Information Rate : 1   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : NaN   
##   
## Mcnemar's Test P-Value : NA   
##   
## Sensitivity : NA   
## Specificity : 1   
## Pos Pred Value : NA   
## Neg Pred Value : NA   
## Prevalence : 0   
## Detection Rate : 0   
## Detection Prevalence : 0   
## Balanced Accuracy : NA   
##   
## 'Positive' Class : feminine   
##

# SVM has predicted all the values correctly hence we can say the accuracy is 100%.

# Question c: ——— New Latest

starwars\_no\_gender <- starwars\_new %>% select(-gender)  
  
starwars\_no\_gender <- starwars\_no\_gender %>% mutate\_all(as.numeric)

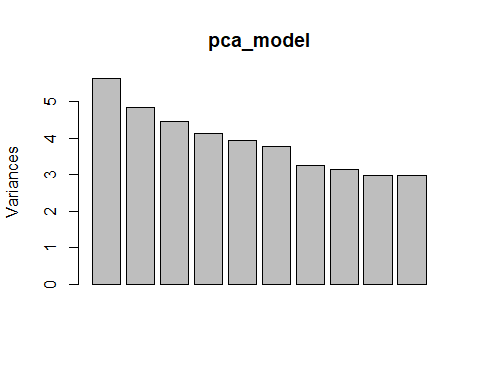
# Perform PCA

pca\_model <- prcomp(starwars\_no\_gender, center = TRUE, scale. = TRUE)

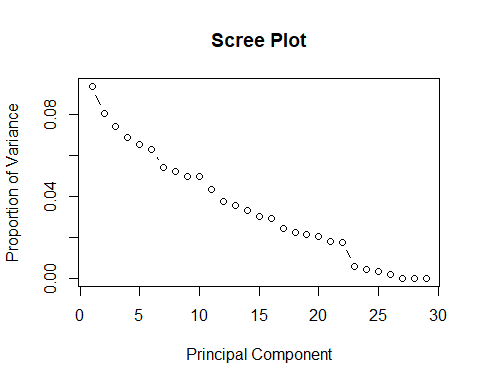
# Determine the number of principal components

# Plotting the PCA components on graph

plot(pca\_model)



plot(pca\_model$sdev^2/sum(pca\_model$sdev^2), type="b", main="Scree Plot", xlab="Principal Component", ylab="Proportion of Variance")

 # From the scree plot, you’d typically choose the number of components before the “elbow”, or where the curve starts leveling off. Let’s assume it’s ‘m’ components.

# Calculating the variance explained to choose components

variance\_explained<-cumsum(pca\_model$sdev^2)/sum(pca\_model$sdev^2)

# Choose a threshold

n\_components <- sum(variance\_explained >= 0.95)  
cat("Number of components to retain:", n\_components, "\n")

## Number of components to retain: 9

# Reduce data to the selected number of components

reduced\_starwars\_pca <- as.data.frame(pca\_model$x[, 1:n\_components])  
  
reduced\_starwars\_pca$gender <- starwars\_new$gender  
  
summary(reduced\_starwars\_pca)

## PC1 PC2 PC3 PC4   
## Min. :-6.5241 Min. :-4.81025 Min. :-1.9659 Min. :-3.5411   
## 1st Qu.:-1.2075 1st Qu.:-0.06105 1st Qu.:-1.3024 1st Qu.:-0.3668   
## Median :-0.1281 Median : 0.52554 Median :-0.7787 Median : 0.3104   
## Mean : 0.0000 Mean : 0.00000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 1.7571 3rd Qu.: 1.04134 3rd Qu.: 0.4645 3rd Qu.: 0.5468   
## Max. : 5.3015 Max. : 5.94949 Max. : 6.4552 Max. : 8.6229   
## PC5 PC6 PC7 PC8   
## Min. :-1.4566 Min. :-6.4010 Min. :-3.0764 Min. :-4.07115   
## 1st Qu.:-1.2372 1st Qu.:-0.2707 1st Qu.:-1.2000 1st Qu.:-1.07535   
## Median :-0.5943 Median : 0.3659 Median : 0.2181 Median :-0.01017   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.00000   
## 3rd Qu.: 0.2889 3rd Qu.: 0.6494 3rd Qu.: 1.4615 3rd Qu.: 0.72032   
## Max. : 6.5217 Max. : 5.2161 Max. : 4.7265 Max. : 3.68554   
## PC9 gender   
## Min. :-3.7233 feminine : 6   
## 1st Qu.:-0.7504 masculine:23   
## Median :-0.0912   
## Mean : 0.0000   
## 3rd Qu.: 0.7301   
## Max. : 4.7856

# Head of final reduced data with target variable

head(reduced\_starwars\_pca)

## PC1 PC2 PC3 PC4 PC5 PC6  
## 1 -0.5962699 1.46575551 -0.9067580 0.61626902 -1.3680614 0.6494367  
## 2 2.4529267 0.64739144 -1.4111843 -0.07990673 -1.4447142 -1.0001053  
## 3 -2.4521559 -0.06104799 -0.4777368 0.31044906 0.3395622 0.3658532  
## 4 0.2842760 1.99027146 -0.5754785 0.69694113 -1.2371833 0.7089565  
## 5 -1.3821668 1.25945903 0.2377471 0.38985984 -0.1587080 0.5726795  
## 6 -1.0581133 0.90354958 -0.8797730 0.69056284 -1.2441201 0.3662087  
## PC7 PC8 PC9 gender  
## 1 1.4615036 -1.44520656 -0.3100082 masculine  
## 2 0.9311200 -1.24791413 0.1875646 masculine  
## 3 0.6334615 -0.01017403 -0.1219860 feminine  
## 4 1.6961580 -2.06768100 0.3822544 masculine  
## 5 1.7548023 -1.07535389 -0.2345918 feminine  
## 6 -0.1238795 -0.68035224 0.7301183 masculine

# Question D :

# Train-test split on PCA data

index <- sample(1:nrow(reduced\_starwars\_pca), nrow(reduced\_starwars\_pca)\*0.8)  
  
set.seed(456)  
train\_data\_pca <- reduced\_starwars\_pca[index, ]  
test\_data\_pca <- reduced\_starwars\_pca[-index, ]

# Grid search on the C parameter

tune\_result <- tune.svm(gender ~ ., data = train\_data\_pca, cost = 10^(-1:2))

# Finding Best model

best\_svm\_model <- svm(gender ~ ., data = train\_data\_pca, cost = tune\_result$best.parameters$cost)

# Predictions on test data

predictions\_pca <- predict(best\_svm\_model, test\_data\_pca)

# confusion matrix

confusionMatrix(predictions\_pca,test\_data\_pca$gender)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction feminine masculine  
## feminine 0 0  
## masculine 2 4  
##   
## Accuracy : 0.6667   
## 95% CI : (0.2228, 0.9567)  
## No Information Rate : 0.6667   
## P-Value [Acc > NIR] : 0.6804   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : 0.4795   
##   
## Sensitivity : 0.0000   
## Specificity : 1.0000   
## Pos Pred Value : NaN   
## Neg Pred Value : 0.6667   
## Prevalence : 0.3333   
## Detection Rate : 0.0000   
## Detection Prevalence : 0.0000   
## Balanced Accuracy : 0.5000   
##   
## 'Positive' Class : feminine   
##

# Accuracy

accuracy <- mean(predictions\_pca == test\_data\_pca$gender)  
print(accuracy)

## [1] 0.6666667

# Question e:

PCA has reduced the dimensionality of the dataset. Hence we are working on reduced number of principal components instead of working on all variables. This will increase the computational efficiency of the model and model will work faster. It will avoid overfitting.

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.