```
## ------------------------------------------------------------------
## Script name: Code homework 2 chapter 6.7 (3,4,5)
##
##
## Purpose of script: Learning Exercises
##
## Author: Michele G Coletta
##
## Date Created: 2022-02-07
##
## Email: mcoletta@student.hult.edu
##
## ------------------------------------------------------------------

#3)a)
##print(iris) to understand my dataset



iris$group <- gsub("setosa", 0, iris$Species)
iris$group <- gsub("versicolor", 0, iris$group)
iris$group <- gsub("virginica", 1, iris$group)

output:
100       5.7       2.8       4.1       1.3 versicolor     0
101       6.3       3.3       6.0       2.5  virginica     1

Sampling virginica as 1 and versicolor as 0.


iris$group <- as.numeric(iris$group) ## at first my logistic regression was not
# running since it said: Error in weights * y : non-numeric argument to binary
#operator


##print(iris) to check changes

#3)b)


#random sampling into training and testing
#step one creating our training index

training_index<- sample(1:nrow(iris), size=0.8*nrow(iris))
```

```
#step two: Doing testing and training sampling prior the regression.
training_iris <- iris[training_index,]
testing_iris <- iris[-training_index,]


#moving into predicting analysis

# building the logistic regression. "glm.fit: fitted probabilities
#numerically 0 or 1 occurred"
my_logit <- glm(group~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,
            data=training_iris, family= "binomial")

summary(my_logit)
output:
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -44.192     25.111  -1.760   0.0784 .
Sepal.Length   -1.626      2.396  -0.679   0.4973
Sepal.Width    -6.488      4.450  -1.458   0.1448
Petal.Length    8.977      4.786   1.876   0.0607 .
Petal.Width    17.167      9.520   1.803   0.0713 .


#calling caret for confusion matrix
library(caret)
# create a confusion matrix for training
pred_iris_train <- predict(my_logit, training_iris,
                           type="response")

confusionMatrix(data= as.factor(as.numeric(pred_iris_train>0.5)) ,
            reference= as.factor(as.numeric(training_iris$group)) )
# create a confusion matrix for testing
pred_iris_test <- predict(my_logit, testing_iris,
                      type="response")

confusionMatrix(data= as.factor(as.numeric(pred_iris_test>0.5)) ,
            reference= as.factor(as.numeric(testing_iris$group)) )
output:

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.96912  -0.00058   0.00000   0.00299   1.65225
```

#3)c) probability of new plant being Virginica with the following parametters

```
new_plant <- data.frame(Sepal.Width = 5, Sepal.Length = 9, Petal.Width = 7, Petal.Length = 10)
#Prediction came out with Virginica as 1 and others as 0
predict(my_logit, new_plant,
                    type="response")
```

output:

```
> predict(my_logit, new_plant,
+                     type="response")
1
1
```

Prediction came with 100% of assurance on having Virginica as 1.

```
# 4)installing rpart so we can go to kyphosis dataset
# 4)a)
## install.packages("rpart")
## library(rpart)

# kyphosis view kyphosis to understand

kyphosis$Kyphosis <- gsub("absent",0,kyphosis$Kyphosis)
kyphosis$Kyphosis <- gsub("present",1,kyphosis$Kyphosis)
# kyphosis view it to check changes
```

output:

```
kyphosis$Kyphosis <- gsub("absent",0,kyphosis$Kyphosis)
> kyphosis$Kyphosis <- gsub("present",1,kyphosis$Kyphosis)
> kyphosis
  Kyphosis Age Number Start
1        0  71      3     5
2        0 158      3    14
3        1 128      4     5
```

Sampling Absent as 0 and Present as 1.

```
kyphosis$Kyphosis <- as.numeric(kyphosis$Kyphosis) # converting to numeric since
#my regression would not work without converting it
```

4)b) building a logistic regression to check if kyphosis would present or absent.

random sampling into training and testing
step one creating our training index

```
train_index_k<- sample(1:nrow(kyphosis), size=0.8*nrow(kyphosis))
# step two. Doing testing and training sampling
train_k <- kyphosis[train_index_k,]
test_k <- kyphosis[-train_index_k,]
```

moving into predicting analysis

building the logistic regression. "glm.fit: fitted probabilities
#numerically 0 or 1 occurred"

##kyphosis checking data again
building a logistic regression to see the coefficients

```
logit_k <- glm(Kyphosis~+Age+Number+Start,
               data=train_k, family= "binomial")

summary(logit_k)
```

output:

```
Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.1533  -0.5480  -0.3932  -0.1809   2.0917

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.411654   1.572566  -0.898  0.36936
Age          0.008804   0.007522   1.171  0.24178
Number       0.332482   0.234651   1.417  0.15651
Start       -0.208303   0.070353  -2.961  0.00307 **
```

Testing and training first to run my logistic regression.
For example: for each unit of number the odds of business success would increase by 39%.

```
> exp(0.332482)-1
[1] 0.3944248
```

#4)c) what's probability of kyphosis being present with the following parametters:

```
k_prob <- data.frame(Age = 50, Number = 5, Start = 10)
predict(logit_k, k_prob,
      type="response")
```

output:

```
> predict(logit_k, k_prob,
+       type="response")
      1
0.1990791
```

Probability of kyphosis being present came out with 19%.

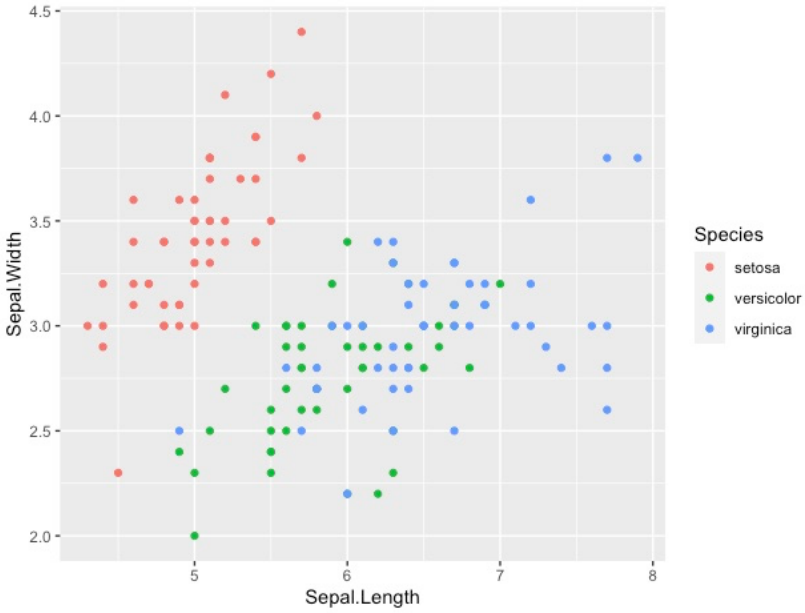
#5)

##plotting to check if variable pairs are homoscedastic or heteroscedastic

```
library(ggplot2)
```

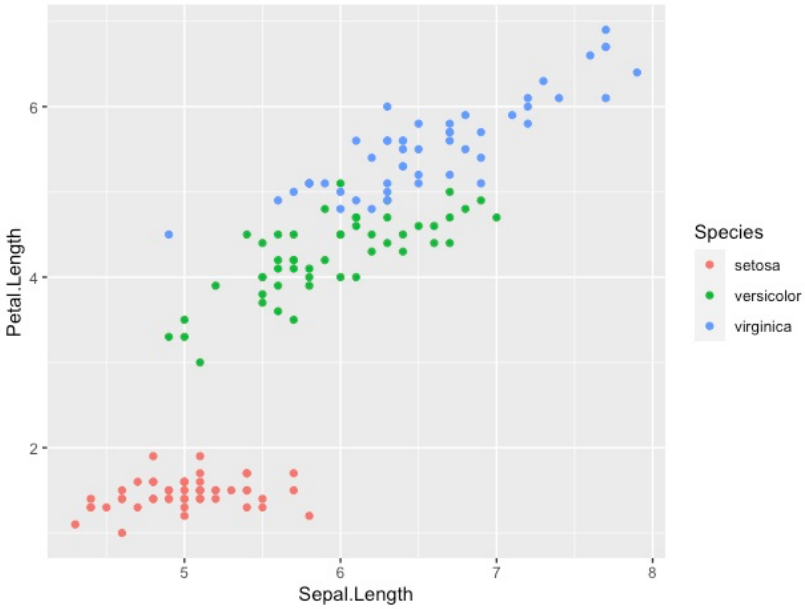
#using ggplot to do scatterplot with Y variable as Sepal.Width

```
ggplot(data=iris, aes(x=Sepal.Length, y=Sepal.Width, color=Species))+geom_point()
```

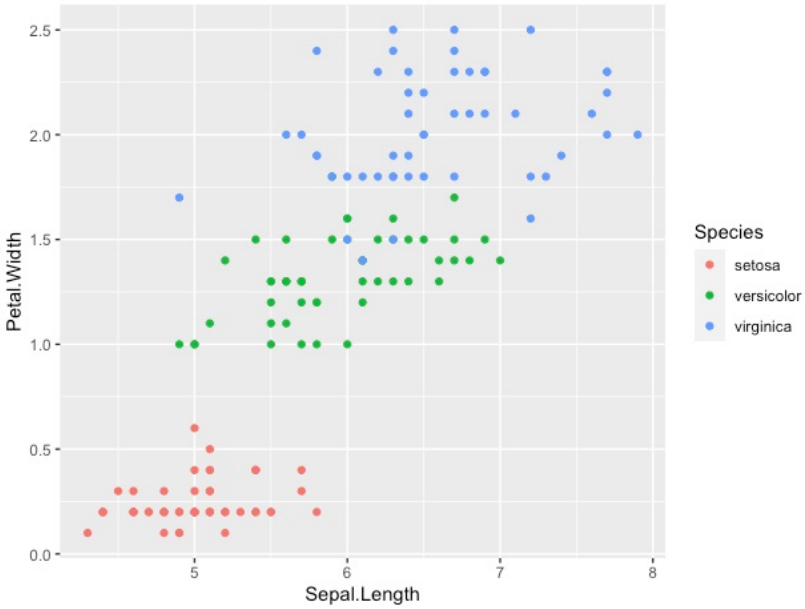


#using ggplot to do scatterplot with Y variable as Petal.Length

```
ggplot(data=iris, aes(x=Sepal.Length, y=Petal.Length, color=Species))+geom_point()
```



#using ggplot to do scatterplot with Y variable as Petal.Width
ggplot(data=iris, aes(x=Sepal.Length, y=Petal.Width, color=Species))+geom_point()



We can see that by plotting sepal.length on X axes with all different Y variables we can see how do they differenciate with each other. This is what our logistic regression is explaining us.