Big Data

Logistic Regression

Name	Sec	BN	ID
Sara Gamal Gerges	1	20	9210455
Eman Ibrahim	1	14	9210265

Q1)

```r
#(Q1) Write the variable pairs that are not correlated at all to each other.
if (cor.mat[1,2] == 0) {
  print("Price and Income are not correlated")
}
if (cor.mat[1,3] == 0) {
  print("Price and Age are not correlated")
}
if (cor.mat[2,3] == 0) {
  print("Income and Age are not correlated")
}
```

```
       Price     Income        Age
Price      1 0.00000000 0.00000000
Income     0 1.00000000 0.09612083
Age        0 0.09612083 1.00000000
> #(Q1) Write the variable pairs that are not correlated at all to each other.
> if (cor.mat[1,2] == 0) {
+   print("Price and Income are not correlated")
+ }
[1] "Price and Income are not correlated"
> if (cor.mat[1,3] == 0) {
+   print("Price and Age are not correlated")
+ }
[1] "Price and Age are not correlated"
> if (cor.mat[2,3] == 0) {
+   print("Income and Age are not correlated")
+ }
> 
```

[1] "Price and Income are not correlated"

[1] "Price and Age are not correlated"

Q2)

No, there is no highly correlated variables because no correlation value is close to 1 or -1.

Q3)

```
> #(Q3): How many categories are there for the Price variable?
> ##There are 3 categories for the Price variable (10,20,30)
> levels(as.factor(Mydata$Price))
[1] "10" "20" "30"
> 
```

Q4)

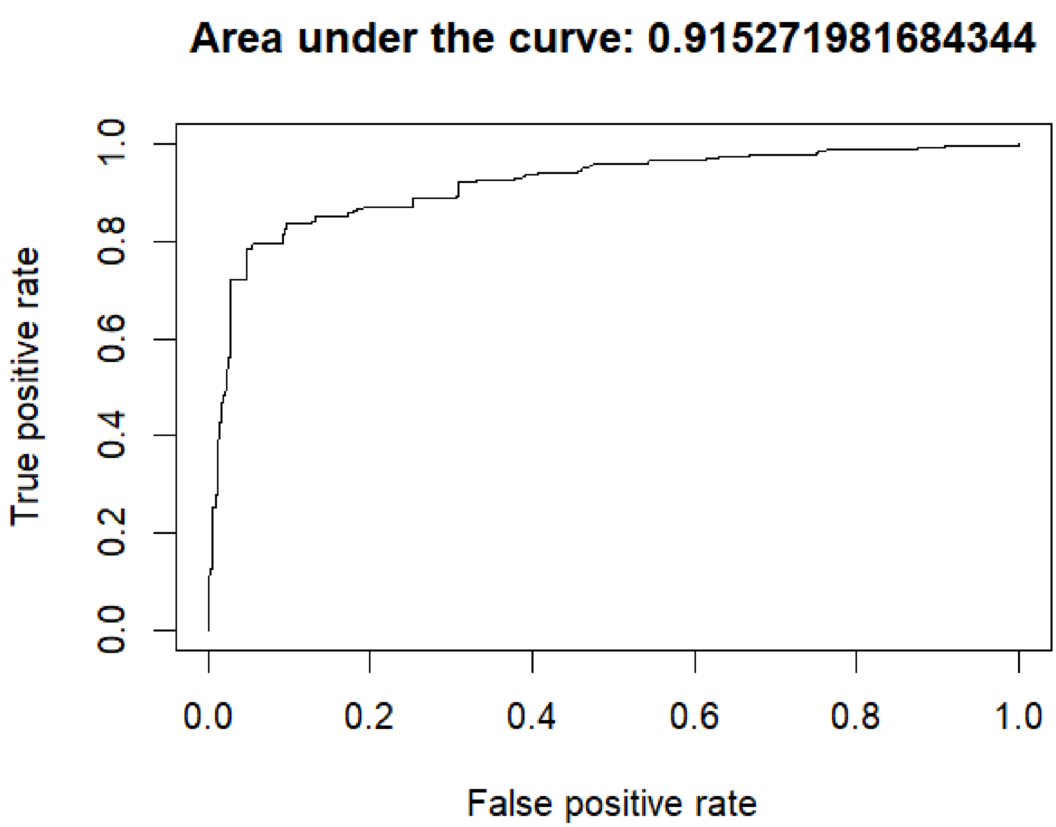
```
#(Q4): Why is it divided into two entries only in the model?
##It is divided into two entries because the Price variable is a categorical variable with 3 categories,
##for n categories, the model will have n-1 entries in the coefficients table, so we have 2 entries in the model.
##the model will use those 2 entries to know the effect of each category on the dependent variable.
```

Q5)

a. Write AUC value 0.915272
b. Maximum value of AUC is 1 (ideal case)

AUC ranges from 0 to 1, where:

- **1** → Perfect classification (ideal case).
- **0.5**→ Random guessing (no predictive power).
- < **0.5** → Worse than random (model is making wrong predictions more often).



Q6)

Each point in the ROC curve represents a different threshold used by the classifier.

What changes from one point to another?

The value that changes and drives both **True Positive Rate (TPR)** and **False Positive Rate (FPR)** is the **classification threshold**.

How does the threshold affect TPR & FPR?

- **Lower threshold → More positives predicted**
  - **Higher TPR**
  - **Higher FPR**
- **Higher threshold → More negatives predicted**
  - **Lower TPR**
  - **Lower FPR**

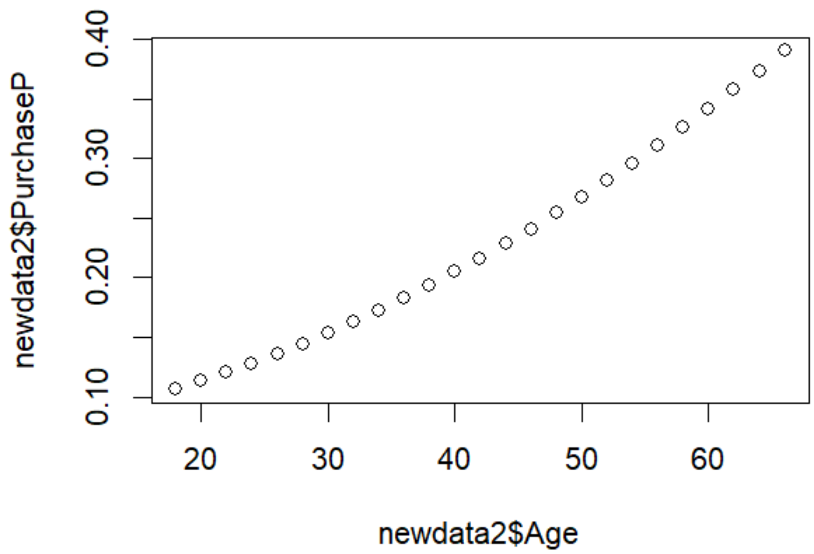
Q7)

```
> # [4] Predictions
> #Prediction - 1
> Price <- c(10,20,30)
> Age <- c(mean(Mydata$Age))
> Income <- c(mean(Mydata$Income))
> newdata1 <- data.frame(Income,Age,Price) # Note: The predict function requires the variables to be named exactly as in
the fitted model.
> newdata1
  Income    Age Price
1 42.492 35.976    10
2 42.492 35.976    20
3 42.492 35.976    30
> newdata1$PurchaseP <- predict (mylogit,newdata=newdata1,type="response")
> newdata1
  Income    Age Price PurchaseP
1 42.492 35.976    10 0.6707408
2 42.492 35.976    20 0.4918407
3 42.492 35.976    30 0.1826131
> 
```

As the price increases the predicted probability of purchase decreases.

Q8)

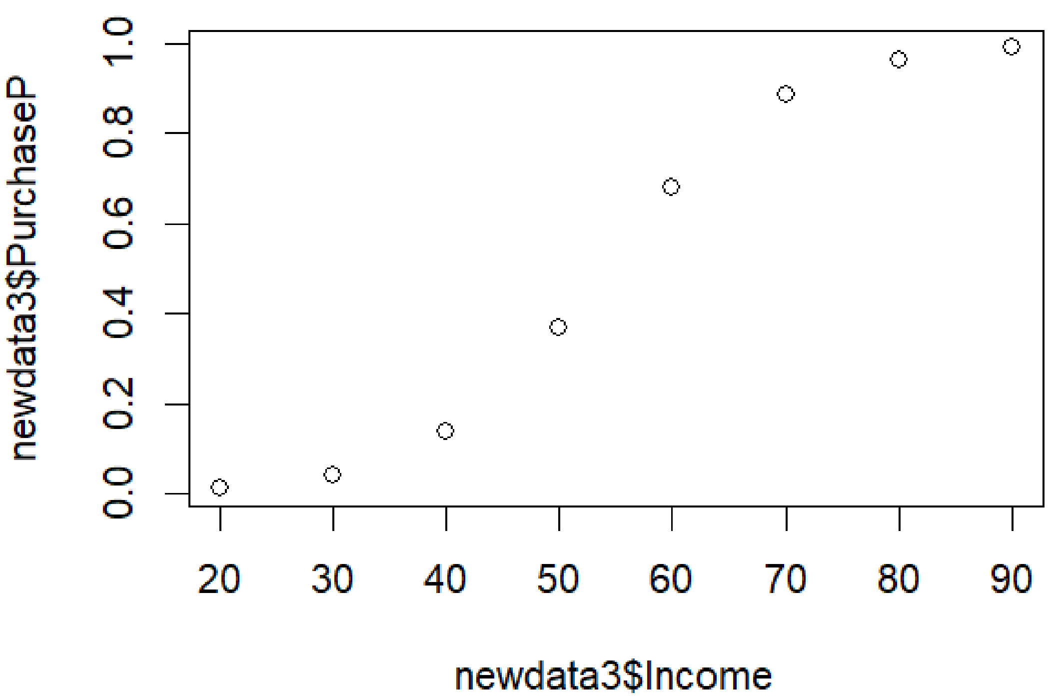
```
> cbind(newdata2$Age,newdata2$PurchaseP)
      [,1]      [,2]
 [1,]   18 0.1063052
 [2,]   20 0.1131540
 [3,]   22 0.1203845
 [4,]   24 0.1280103
 [5,]   26 0.1360445
 [6,]   28 0.1444993
 [7,]   30 0.1533864
 [8,]   32 0.1627160
 [9,]   34 0.1724975
[10,]   36 0.1827387
[11,]   38 0.1934457
[12,]   40 0.2046231
[13,]   42 0.2162731
[14,]   44 0.2283958
[15,]   46 0.2409892
[16,]   48 0.2540483
[17,]   50 0.2675657
[18,]   52 0.2815308
[19,]   54 0.2959303
[20,]   56 0.3107477
[21,]   58 0.3259636
[22,]   60 0.3415553
[23,]   62 0.3574973
[24,]   64 0.3737609
[25,]   66 0.3903150
> plot(newdata2$Age,newdata2$PurchaseP)
> 
```



It ‘s clear that as the age increases the probability of purchase increases.

Q9)

```
> #Prediction - 3
> newdata3 <- data.frame(Income= seq(20,90,10),Age=mean(Mydata$Age),Price=30)
> newdata3$PurchaseP<-predict(mylogit,newdata=newdata3,type="response")
> cbind(newdata3$Income,newdata3$PurchaseP)
     [,1]       [,2]
[1,]   20 0.01219091
[2,]   30 0.04281102
[3,]   40 0.13948050
[4,]   50 0.37004640
[5,]   60 0.68039246
[6,]   70 0.88525564
[7,]   80 0.96546923
[8,]   90 0.99022745
> plot(newdata3$Income,newdata3$PurchaseP)
> 
```



It ‘s clear that as the income increases the probability of purchase increases.