Assignment 3

Pan Hao Chen_G2004315D

1.

(a.)

I put the information from question one into Excel and save it as “Problem 1.csv” and used R program to open it.

> library(foreign)

> setwd("~/Desktop/Data/Assignment 3")

> mydata= read.csv("Problem 1.csv")

> summary(mydata)

```
     record         age            income         education       default         
 Min.   : 1.0   Min.   :25.00   Min.   : 35000   Min.   : 9.00   Length:11         
 1st Qu.: 3.5   1st Qu.:29.00   1st Qu.: 52500   1st Qu.:12.00   Class :character  
 Median : 6.0   Median :33.00   Median : 70000   Median :12.00   Mode  :character  
 Mean   : 6.0   Mean   :38.36   Mean   : 84455   Mean   :13.09                     
 3rd Qu.: 8.5   3rd Qu.:46.00   3rd Qu.:111000   3rd Qu.:15.00                     
 Max.   :11.0   Max.   :59.00   Max.   :170000   Max.   :18.00                     
```

> (mydata[,2]-25)/(59-25)

>standard_age<c(1.00000000,0.47058824,0.11764706,0.00000000,0.70588235,0.14705882,0.23529412,0.11764706, 0.52941176, 0.91176471, 0.08823529)

> (mydata[,3]-35000)/(170000-35000)

> standard_income<-c(0.71851852, 0.37037037, 0.18518519, 0.00000000, 1.00000000, 0.25925926, 0.07407407, 0.55555556, 0.07407407, 0.57037037, 0.22222222)

> (mydata[,4]-9)/(18-9)

> standard_educ<-c(1.0000000, 0.5555556, 0.7777778, 0.1111111, 0.3333333, 0.0000000, 0.4444444, 0.3333333, 0.3333333, 0.7777778, 0.3333333)

> mydata<-cbind(mydata,standard_age, standard_income, standard_educ)

	record	age	income	education	default	standard_age	standard_income	standard_educ
1	1	59	132000	18	No	1.00000000	0.71851852	1.0000000
2	2	41	85000	14	Yes	0.47058824	0.37037037	0.5555556
3	3	29	60000	16	Yes	0.11764706	0.18518519	0.7777778
4	4	25	35000	10	Yes	0.00000000	0.00000000	0.1111111
5	5	49	170000	12	No	0.70588235	1.00000000	0.3333333
6	6	30	70000	9	Yes	0.14705882	0.25925926	0.0000000
7	7	33	45000	13	Yes	0.23529412	0.07407407	0.4444444
8	8	29	110000	12	No	0.11764706	0.55555556	0.3333333
9	9	43	45000	12	No	0.52941176	0.07407407	0.3333333
10	10	56	112000	16	No	0.91176471	0.57037037	0.7777778
11	11	28	65000	12	Yes	0.08823529	0.22222222	0.3333333

(b.)

> (45-25)/(59-25)

[1] 0.5882353

```
> (60000-35000)/(170000-35000)
[1] 0.1851852
> (15-9)/(18-9)
[1] 0.6666667
> new_ppl<-c(12,45,60000,15 ,"Don't know",0.5882353, 0.1851852, 0.6666667)
> mydata=rbind(mydata, new_ppl)
```

	record	age	income	education	default	standard_age	standard_income	standard_educ
1	1	59	132000	18	No	1	0.71851852	1
2	2	41	85000	14	Yes	0.47058824	0.37037037	0.5555556
3	3	29	60000	16	Yes	0.11764706	0.18518519	0.7777778
4	4	25	35000	10	Yes	0	0	0.1111111
5	5	49	170000	12	No	0.70588235	1	0.3333333
6	6	30	70000	9	Yes	0.14705882	0.25925926	0
7	7	33	45000	13	Yes	0.23529412	0.07407407	0.4444444
8	8	29	110000	12	No	0.11764706	0.55555556	0.3333333
9	9	43	45000	12	No	0.52941176	0.07407407	0.3333333
10	10	56	112000	16	No	0.91176471	0.57037037	0.7777778
11	11	28	65000	12	Yes	0.08823529	0.22222222	0.3333333
12	12	45	60000	15	Don't know	0.5882353	0.1851852	0.6666667

```
> data_set<-data.frame(mydata[,6:8])
> distance.matrix <- dist(data_set, method = "euclidean", diag = T)
> distance.matrix
           1         2         3         4         5         6         7         8         9        10        11        12
1  0.0000000
2  0.7739604 0.0000000
3  1.0546914 0.4563373 0.0000000
4  1.5186811 0.7457603 0.7018396 0.0000000
5  0.7811412 0.7079404 1.0988516 1.2440469 0.0000000
6  1.3922743 0.6524255 0.7818506 0.3180996 0.9859471 0.0000000
7  1.1439955 0.3943355 0.3705370 0.4146823 1.0445754 0.4894996 0.0000000
8  1.1178319 0.4563374 0.5785371 0.6098078 0.7372595 0.4469537 0.5079478 0.0000000
9  1.0398108 0.3750126 0.6159763 0.5789184 0.9425925 0.5399985 0.3144056 0.6335413 0.0000000
10 0.2812757 0.5329347 0.8826044 1.2653386 0.6515367 1.1342414 0.9027921 0.9101501 0.7681372 0.0000000
11 1.2337218 0.4663951 0.4469538 0.3264213 0.9931899 0.3405041 0.2364738 0.3346284 0.4653865 0.9984682 0.0000000
12 0.7517351 0.2459269 0.4835276 0.8300339 0.8881864 0.8028500 0.4316201 0.6853748 0.3562542 0.5151549 0.6020655 0.0000000
```

The last row shows the distance between the 12th person and the 11 observations.

(c.)

```
> library(foreign)
> setwd("~/Desktop/Data/Assignment 3")
> training= read.csv("Problem 1.csv")
> training$default<-as.factor(training$default)
> training<-training[,(2:5)]
> my_model<-train.kknn(default~., data = training, kmax = 5)
> new_ppl<-data.frame(age=45, income=60000, education=15)
> predict(my_model, new_ppl)
[1] Yes
Levels: No Yes
```

KNN model which is based on the 11 observations shows that this person with age=45, income=60000, and education=15, will default.

2.

(a.)

In market basket analysis, the term “support” means among all the transaction, how many times did a certain itemset occur. For example, if we are looking for the support for an itemset {x,y}, the formula should be numbers of transactions containing {x,y} divided by total number of transactions. So we could see this formula as the estimate probability of {x,y} among the randomly picked baskets. If the support of {x,y} is high, we could assume that there is a high association between {x} and {y} since they appear together quite often among all the transactions.

As for the term “confidence”, it shows for those who bought x, how many of them bought y at the same time. The formula would be P(x and y)/P(x). If the confidence of {x,y} is high, we could say that consumers are likely to buy {y} after they bought {x}.

(b.)

If minisup>0.1, then the support count should be greater than 1.

1-itemset	Count	2-itemset	Count	3-itemset	Count	4-itemset	Count
Milk	5	~~{Milk, Beer}~~	~~1~~	~~{Milk, Diaper, Milo}~~	~~1~~	{Milk, Milo, Bread, Egg}	2
Beer	3	{Milk, Diaper}	2	~~{Milk, Diaper, Bread}~~	~~0~~	~~{Beer, Diaper, Coke, Milk}~~	~~0~~
Diaper	4	{Milk, Milo}	3	~~{Milk, Diaper, Egg}~~	~~1~~	~~{Beer, Diaper, Coke, Milo}~~	~~0~~
Milo	4	{Milk, Bread}	2	~~{Milk, Diaper, Beer}~~	~~1~~	~~{Beer, Diaper, Coke, Bread}~~	~~0~~
Bread	4	{Milk, Egg}	4	~~{Milk, Diaper, Coke}~~	~~0~~	~~{Beer, Diaper, Coke, Egg}~~	~~1~~
Egg	6	~~{Milk, Coke}~~	~~1~~	{Milk, Milo, Bread}	2	~~{Beer, Diaper, Coke, Instant Noodle}~~	~~0~~
Coke	6	~~{Milk, Instant Noodle}~~	~~0~~	{Milk, Milo, Egg}	2		
~~Butter~~	~~1~~	{Beer, Diaper}	3	{Milk, Bread, Egg}	2		
Instant Noodle	2	~~{Beer, Milo}~~	~~1~~	~~{Milk, Bread, Coke}~~	~~0~~		
		~~{Beer, Bread}~~	~~0~~	~~{Milk, Egg, Coke}~~	~~1~~		
		~~{Beer, Egg}~~	~~1~~	{Beer, Diaper, Coke}	2		
		{Beer, Coke}	2	~~{Beer, Diaper, Egg}~~	~~1~~		
		~~{Beer, Instant Noodle}~~	~~0~~	~~{Beer, Coke, Instant Noodle}~~	~~0~~		
		~~{Diaper, Milo}~~	~~1~~	~~{Diaper, Egg, Coke}~~	~~1~~		
		~~{Diaper, Bread}~~	~~0~~	~~{Diaper, Coke, Instant Noodle}~~	~~0~~		
		{Diaper, Egg}	2	{Milo, Bread, Egg}	2		
		{Diaper, Coke}	2	~~{Milo, Bread, Coke}~~	~~0~~		
		~~{Diaper, Instant Noodle}~~	~~0~~	~~{Milo, Egg, Coke}~~	~~0~~		
		{Milo, Bread}	2	~~{Bread, Egg, Coke}~~	~~1~~		
		{Milo, Egg}	2	~~{Bread, Coke, Instant Noodle}~~	~~1~~		
		~~{Milo, Coke}~~	~~1~~	~~{Egg, Coke, Instant Noodle}~~	~~0~~		
		~~{Milo, Instant Noodle}~~	~~1~~				
		{Bread, Egg}	2				
		{Bread, Coke}	2				
		~~{Bread, Instant Noodle}~~	~~1~~				
		{Egg, Coke}	3				
		~~{Egg, Instant Noodle}~~	~~0~~				
		{Coke, Instant Noodle}	2				

From the above result, the maximum size of itemset would be 4.

(c.) Consider {Diaper}->{Beer},

Support should be 3/10. Confidence should be 3/4.

3.

(a.)

```
> library(foreign)
> setwd("~/Desktop/Data/Assignment 3")
> Bank<-read.csv("Bank.csv")
> install.packages("dplyr")
> library(dplyr)
> mydata<-select(Bank, PersonalLoan, Age, Experience, Income, Family, CCAvg,
Education, CD.Account)
```

(b.)

```
> library(psych)
> describe(mydata)
```

```
> describe(mydata)
             vars    n  mean    sd median trimmed   mad min max range  skew kurtosis   se
PersonalLoan    1 5000  0.10  0.29    0.0    0.00  0.00   0   1     1  2.74     5.52 0.00
Age             2 5000 45.34 11.46   45.0   45.38 14.83  23  67    44 -0.03    -1.15 0.16
Experience      3 5000 20.10 11.47   20.0   20.13 14.83  -3  43    46 -0.03    -1.12 0.16
Income          4 5000 73.77 46.03   64.0   68.83 43.00   8 224   216  0.84    -0.05 0.65
Family          5 5000  2.40  1.15    2.0    2.37  1.48   1   4     3  0.16    -1.40 0.02
CCAvg           6 5000  1.94  1.75    1.5    1.65  1.33   0  10    10  1.60     2.64 0.02
Education       7 5000  1.88  0.84    2.0    1.85  1.48   1   3     2  0.23    -1.55 0.01
CD.Account      8 5000  0.06  0.24    0.0    0.00  0.00   0   1     1  3.69    11.61 0.00
```

```
> summary(mydata)
```

```
  PersonalLoan        Age          Experience        Income           Family          CCAvg          Education        CD.Account
 Min.   :0.000   Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   :1.000   Min.   : 0.000   Min.   :1.000   Min.   :0.0000
 1st Qu.:0.000   1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:1.000   1st Qu.: 0.700   1st Qu.:1.000   1st Qu.:0.0000
 Median :0.000   Median :45.00   Median :20.0   Median : 64.00   Median :2.000   Median : 1.500   Median :2.000   Median :0.0000
 Mean   :0.096   Mean   :45.34   Mean   :20.1   Mean   : 73.77   Mean   :2.396   Mean   : 1.938   Mean   :1.881   Mean   :0.0604
 3rd Qu.:0.000   3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00   3rd Qu.:3.000   3rd Qu.: 2.500   3rd Qu.:3.000   3rd Qu.:0.0000
 Max.   :1.000   Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :4.000   Max.   :10.000   Max.   :3.000   Max.   :1.0000
```

From the above information, we could know that variables “PersonalLoan” and “CD. Account” are both dummy variables as the range is 1. Also, we can see that “Income” has the largest variation among these variables. However, there is a weird part in variable “Experience”, its mini is (-3) which is unusual, thus we could know that some data that we collected is useless since it does not reflect the fact.

(c.)

```
Eliminate_Bank_Wrong<- mydata[mydata$Experience >= 0, ]
```

(d.)

```
>   set.seed(5000)
>   sample<-sample(1:4948, 4500)
>   For_training<-Eliminate_Bank_Wrong[sample,]
>   validation<-Eliminate_Bank_Wrong[-sample,]
```

(e.)

```
> library(kknn)
> bank_model<- train.kknn(PersonalLoan~., data = For_training)
> summary(bank_model)
```

```
Call:
train.kknn(formula = PersonalLoan ~ ., data = For_training)

Type of response variable: continuous
minimal mean absolute error: 0.01533333
Minimal mean squared error: 0.0136839
Best kernel: optimal
Best k: 3
```

> prediction<-predict(bank_model, validation)

(f.)

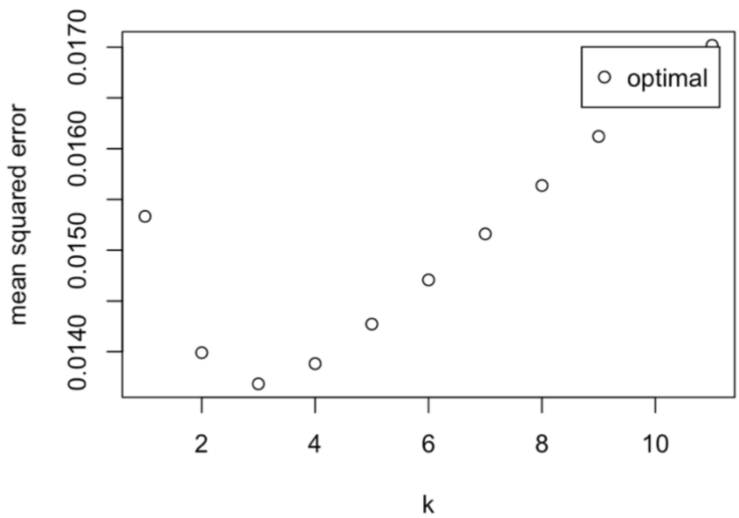
> actual<-validation[,1]

> Table<-table(actual, prediction)

> mean(prediction==actual)

[1] 0.9441964

> plot(bank_model)



(4.)

(a.)

> install.packages("MASS")

> library(MASS)

> mydata<-Pima.tr

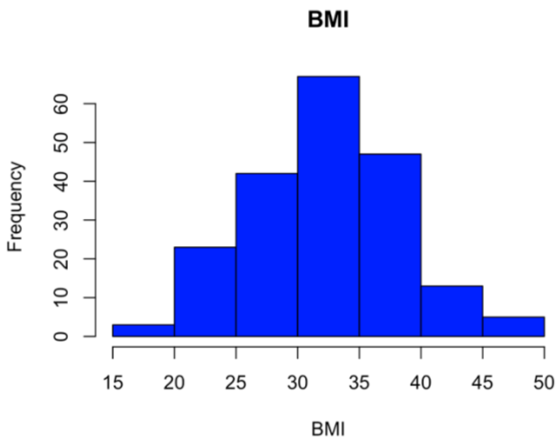
> summary(mydata)

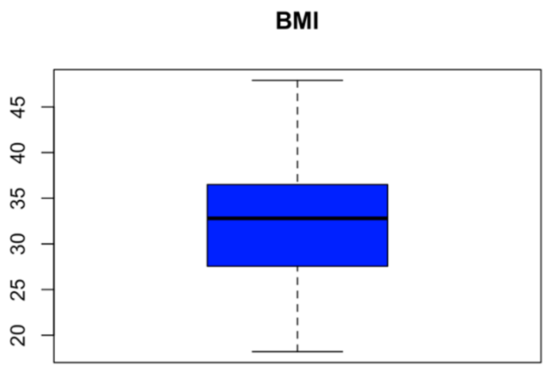
```
> summary(mydata)
     npreg            glu              bp              skin            bmi             ped              age          type
 Min.   : 0.00   Min.   : 56.0   Min.   : 38.00   Min.   : 7.00   Min.   :18.20   Min.   :0.0850   Min.   :21.00   No :132
 1st Qu.: 1.00   1st Qu.:100.0   1st Qu.: 64.00   1st Qu.:20.75   1st Qu.:27.57   1st Qu.:0.2535   1st Qu.:23.00   Yes: 68
 Median : 2.00   Median :120.5   Median : 70.00   Median :29.00   Median :32.80   Median :0.3725   Median :28.00
 Mean   : 3.57   Mean   :124.0   Mean   : 71.26   Mean   :29.21   Mean   :32.31   Mean   :0.4608   Mean   :32.11
 3rd Qu.: 6.00   3rd Qu.:144.0   3rd Qu.: 78.00   3rd Qu.:36.00   3rd Qu.:36.50   3rd Qu.:0.6160   3rd Qu.:39.25
 Max.   :14.00   Max.   :199.0   Max.   :110.00   Max.   :99.00   Max.   :47.90   Max.   :2.2880   Max.   :63.00
```

(b.)

> hist(mydata$bmi, main = "BMI", xlab = "BMI", col = 'blue')

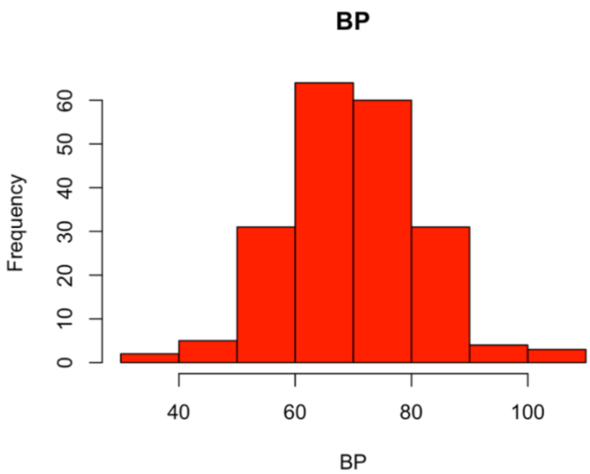
> boxplot(mydata$bmi, main = "BMI", col = 'Blue')

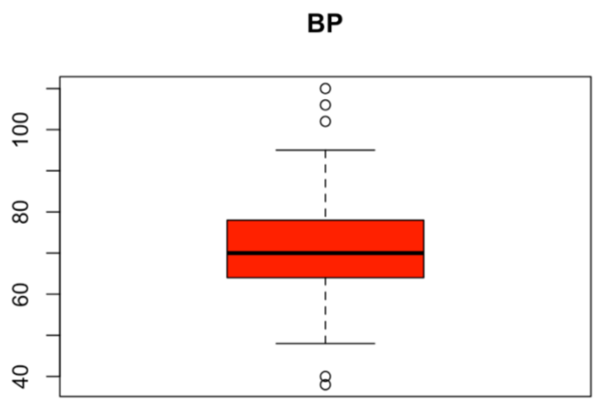




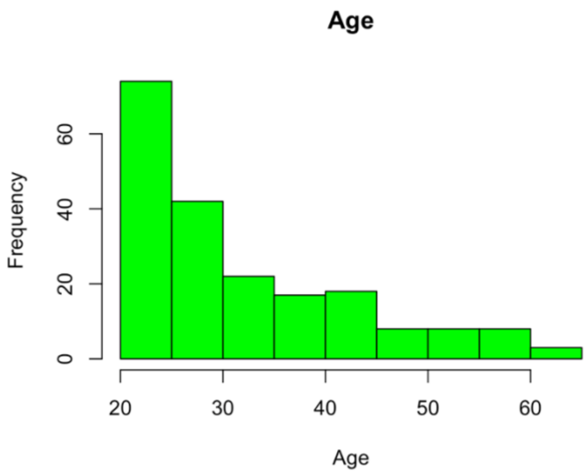
> hist(mydata$bp, main = "BP", xlab = "BP", col = 'red')

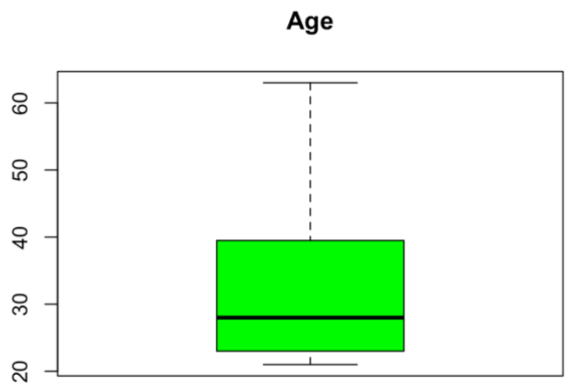
> boxplot(mydata$bp, main = "BP", col = 'red')






```
> hist(mydata$age, main = "Age", xlab = "Age", col = 'green')
> boxplot(mydata$age, main = "Age", col = 'green')
```






```
> library(moments)
> skewness(mydata$bmi)
[1] 0.02512996
```

> skewness(mydata$bp)
[1] 0.1652765
> skewness(mydata$age)
[1] 1.087979

Roughly speaking, we could say variable “bmi” and “bp” are pretty close to a normal distribution as their skewness are close to 0. However, as for “age”, it is positively skewed with skewness 1.087979.

(c.)

> mydata_lda<-lda(mydata$type~.,data=mydata)

(d.)

> print(mydata.lda)$class

```
> print(mydata.lda)$class
Call:
lda(mydata$type ~ ., data = mydata)

Prior probabilities of groups:
  No  Yes
0.66 0.34

Group means:
       npreg      glu       bp     skin      bmi       ped      age
No  2.916667 113.1061 69.54545 27.20455 31.07424 0.4154848 29.23485
Yes 4.838235 145.0588 74.58824 33.11765 34.70882 0.5486618 37.69118

Coefficients of linear discriminants:
                LD1
npreg  0.0794995781
glu    0.0240316424
bp    -0.0018125857
skin  -0.0008317413
bmi    0.0494891916
ped    1.2530603130
age    0.0314375125
NULL
```

For the prior probability, it shows the ratio of “Yes” and “No” in our 200 observations. Next, it calculates the mean of every independent variable.
Last part, it shows the linear combination of “type” and other independent variables.
LD1= 0.0794995781*npreg+ 0.0240316424*glu-0.0018125857*bp-0.0008317413*skin+ 0.0494891916*bmi+ 1.2530603130*ped+ 0.0314375125*age.

(e.)

> print(mydata)$type

```
> print(mydata)$type
   npreg glu  bp skin  bmi   ped age type
1      5  86  68   28 30.2 0.364  24   No
2      7 195  70   33 25.1 0.163  55  Yes
3      5  77  82   41 35.8 0.156  35   No
4      0 165  76   43 47.9 0.259  26   No
5      0 107  60   25 26.4 0.133  23   No
6      5  97  76   27 35.6 0.378  52  Yes
7      3  83  58   31 34.3 0.336  25   No
8      1 193  50   16 25.9 0.655  24   No
9      3 142  80   15 32.4 0.200  63   No
10     2 128  78   37 43.3 1.224  31  Yes
11     0 137  40   35 43.1 2.288  33  Yes
12     9 154  78   30 30.9 0.164  45   No
13     1 189  60   23 30.1 0.398  59  Yes
14    12  92  62    7 27.6 0.926  44  Yes
15     1  86  66   52 41.3 0.917  29   No
16     4  99  76   15 23.2 0.223  21   No
17     1 109  60    8 25.4 0.947  21   No
18    11 143  94   33 36.6 0.254  51  Yes
19     1 149  68   29 29.3 0.349  42  Yes

 [ reached 'max' / getOption("max.print") -- omitted 75 rows ]
  [1] No  Yes No  No  No  Yes No  No  No  Yes Yes No  Yes Yes No  No  No  Yes Yes No  No  No  No 
 [24] No  No  Yes No  Yes No  No  No  No  Yes No  Yes No  No  No  No  No  Yes No  No  No  No  No 
 [47] No  No  Yes Yes Yes No  No  Yes No  No  No  No  No  No  Yes Yes No  No  No  No  Yes Yes No  Yes
 [70] No  Yes Yes Yes No  Yes Yes No  No  Yes No  No  No  Yes Yes No  No  Yes No  No  No  No  No 
 [93] Yes No  No  Yes No  No  No  Yes No  Yes No  Yes No  No  No  Yes No  No  No  No  Yes Yes No 
[116] No  Yes Yes No  Yes No  No  Yes No  Yes No  No  No  No  Yes Yes No  No  No  No  No  No  No 
[139] No  No  Yes Yes No  No  No  No  No  Yes No  No  No  Yes Yes Yes No  Yes Yes No  No  Yes Yes
[162] No  No  No  No  No  Yes No  No  No  Yes No  Yes Yes Yes No  No  No  No  No  No  No  No  Yes
[185] No  Yes Yes Yes No  Yes No  No  Yes No  No  No  Yes No  No  Yes
Levels: No Yes
```

This shows the actual result of whether or not they have obesity.

(f.)

> prediction<-predict(mydata.lda)$class

> mean(mydata$type==prediction)

[1] 0.77

> table(mydata$type,prediction)

	prediction	
	No	Yes
No	115	17
Yes	29	39

The accuracy rate is 77%.

(g.)

> mydata.qda<-qda(mydata$type~.,data=mydata)

> print(mydata.qda)$class

```
> print(mydata.qda)$class
Call:
qda(mydata$type ~ ., data = mydata)

Prior probabilities of groups:
  No  Yes 
0.66 0.34 

Group means:
       npreg      glu       bp     skin      bmi       ped      age
No  2.916667 113.1061 69.54545 27.20455 31.07424 0.4154848 29.23485
Yes 4.838235 145.0588 74.58824 33.11765 34.70882 0.5486618 37.69118
NULL
```

For the prior probability, it shows the ratio of “Yes” and “No” in our 200 observations. Next, it calculates the mean of every independent variable. However, since QDA is a non-linear method, there is no regressor for each independent variable, which is different from the LDA.

> print(mydata)$type

```
> print(mydata)$type
    npreg glu  bp skin  bmi   ped age type
1       5  86  68   28 30.2 0.364  24   No
2       7 195  70   33 25.1 0.163  55  Yes
3       5  77  82   41 35.8 0.156  35   No
4       0 165  76   43 47.9 0.259  26   No
5       0 107  60   25 26.4 0.133  23   No
6       5  97  76   27 35.6 0.378  52  Yes
7       3  83  58   31 34.3 0.336  25   No
8       1 193  50   16 25.9 0.655  24   No
9       3 142  80   15 32.4 0.200  63   No
10      2 128  78   37 43.3 1.224  31  Yes
11      0 137  40   35 43.1 2.288  33  Yes
12      9 154  78   30 30.9 0.164  45   No
13      1 189  60   23 30.1 0.398  59  Yes
14     12  92  62    7 27.6 0.926  44  Yes
15      1  86  66   52 41.3 0.917  29   No
16      4  99  76   15 23.2 0.223  21   No
17      1 109  60    8 25.4 0.947  21   No
18     11 143  94   33 36.6 0.254  51  Yes
19      1 149  68   29 29.3 0.349  42  Yes

 [ reached 'max' / getOption("max.print") -- omitted 75 rows ]
  [1] No  Yes No  No  No  Yes No  No  No  Yes Yes No  Yes Yes No  No  No  Yes Yes No  No  No  No 
 [24] No  No  Yes No  Yes No  No  No  No  Yes No  Yes No  No  No  No  No  Yes No  No  No  No  No 
 [47] No  No  Yes Yes No  No  Yes No  No  No  No  No  No  Yes Yes No  No  No  No  Yes Yes No  Yes
 [70] No  Yes Yes Yes No  Yes Yes No  No  Yes No  No  No  Yes Yes No  No  Yes No  No  No  No  No 
 [93] Yes No  No  Yes No  No  No  Yes No  Yes No  Yes No  No  No  Yes No  No  No  No  Yes Yes No 
[116] No  Yes Yes No  Yes No  No  Yes No  Yes No  No  No  No  Yes Yes No  No  No  No  No  No  No 
[139] No  No  Yes Yes No  No  No  No  No  Yes No  No  No  Yes Yes Yes No  Yes Yes No  No  Yes Yes
[162] No  No  No  No  No  Yes No  No  No  Yes No  Yes Yes Yes No  No  No  No  No  No  No  No  Yes
[185] No  Yes Yes Yes No  Yes No  No  Yes No  No  No  Yes No  No  Yes
Levels: No Yes
```

This shows the actual result of whether or not they have obesity.

> prediction.qda<-predict(mydata.qda)$class

> mean(mydata$type==prediction.qda)

[1] 0.77

> table(mydata$type,prediction.qda)

	prediction.qda	
	No	Yes
No	114	18
Yes	28	40

The accuracy rate is also 77%, and if we see the tables of (mydata$type,prediction) and (mydata$type,prediction.qda), the results are roughly the same.

> Type_Yes<-mydata[mydata$type=="Yes",]

> Type_No<-mydata[mydata$type=="No",]

> cov(Type_Yes[,(1:7)])

	npreg	glu	bp	skin	bmi	ped	age
npreg	15.7794118	-8.19929763	6.4249342	-0.5179982	-1.0328797	-0.13301076	21.9343723
glu	-8.1992976	907.25021949	23.7111501	87.5153644	9.9815628	-0.08215891	58.1229148
bp	6.4249342	23.71115013	134.1861282	19.7058824	5.1589113	-0.79546971	26.3037752
skin	-0.5179982	87.51536435	19.7058824	151.3292362	28.2855136	0.34795083	26.6786655
bmi	-1.0328797	9.98156277	5.1589113	28.2855136	23.1452941	0.45769407	-7.9121598
ped	-0.1330108	-0.08215891	-0.7954697	0.3479508	0.4576941	0.12888309	-0.4173747
age	21.9343723	58.12291484	26.3037752	26.6786655	-7.9121598	-0.41737467	131.7987270

> cov(Type_No[,(1:7)])

	npreg	glu	bp	skin	bmi	ped	age
npreg	7.87849873	10.7722646	8.19083969	2.91030534	-0.03575064	-0.20734097	16.8288804
glu	10.77226463	709.5611844	81.43025677	13.23768217	19.03786722	-0.51860907	59.0130696
bp	8.19083969	81.4302568	122.84524636	33.87994448	16.61263012	-0.07718251	46.7869535
skin	2.91030534	13.2376822	33.87994448	119.44639139	50.12591950	0.07428175	18.4706801
bmi	-0.03575064	19.0378672	16.61263012	50.12591950	40.72299560	0.14524159	6.9999884
ped	-0.20734097	-0.5186091	-0.07718251	0.07428175	0.14524159	0.07138833	-0.5381376
age	16.82888041	59.0130696	46.78695350	18.47068008	6.99998843	-0.53813764	91.0818297

I also calculate the covariance matrix for both groups, and it is obvious that they have different covariance matrix.

(h.)

From both results, I would say both of them are not so different, so there is no such a strong recommendation of we should use LDA or QDA based on their results. But in (g.), I also did calculate the covariance matrix for both classes, and it turned out that two classes had different covariance matrix, so based on the assumptions of LDA and QDA, it seems QDA would be a better option.

(i.)

LPM, linear probability model is a case of binary regression model. Its dependent variable takes value of either 1 or 0. Even though it could be used for multiple classification, normally people would use it for binary classification. The output of LPM should be interpreted as probability. Hence, we would assume the value of dependent variable would lie within [0,1]. However, sometimes the probability of LPM would be greater than 1 or smaller than 0 which is not we should expect. Thus, we could use transform it into either probit model or logit model by applying cumulative standard normal density function or logistic function respectively.

As for LDA and QDA, both of them are based on Bayes theorem, but different in approach of classification. They both identify the distribution of all the input x for each class, then use Bayes theorem to flip the distribution and calculate the probability. But the assumption between LDA and QDA are different. For LDA, distribution of observation in each class is normal with a class-specific mean vector and common covariance matrix. For QDA, distribution of observation in each class is normal with a class-specific mean vector and class-specific covariance matrix. The difference is how they think of the covariance matrix.

To sum up, without the normal distribution assumption in our observations, LPM would have the advantage over LDA or QDA. However, if the output of dependent variable is fully separated and non-binary, LDA or QDA may be the better option. However, how to choose between LDA and QDA? Then it would depend on the training set. If the training set is small, then LDA would be a better choice since we need to contain and reduce the variance. On the other hand, if the training set is large enough, then the variance wouldn’t be a thing that we need to concern, so QDA will be the better option. In general, LPM would be used for binomial classification, and LDA and QDA would be more favorable when it comes to multiple classification.