Question 4

A. Load the /public/bmort/R/heart.csv data set into a data frame. Are there any missing values? Perform any necessary data imputation on the data set.

In [15]:

```
## Loading the dataset
heart <- read.csv('/public/bmort/R/heart.csv')
head(heart,10)
```

A data.frame: 10 × 14

age	sex	pain	bp	chol	sugar	ecg	rate	angina	stv	sts	mvn	thal
<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<dbl>	<int>	<int>	<int>
63	1	1	145	233	1	2	150	0	2.3	3	0	6
67	1	4	160	286	0	2	108	1	1.5	2	3	3
67	1	4	120	229	0	2	129	1	2.6	2	2	7
37	1	3	130	250	0	0	187	0	3.5	3	0	3
41	0	2	130	204	0	2	172	0	1.4	1	0	3
56	1	2	120	236	0	0	178	0	0.8	1	0	3
62	0	4	140	268	0	2	160	0	3.6	3	2	3
57	0	4	120	354	0	0	163	1	0.6	1	0	3
63	1	4	130	254	0	2	147	0	1.4	2	1	7
53	1	4	140	203	1	2	155	1	3.1	3	0	7

In [16]:

```
## Finding missing values
## Locating the missing values
which(is.na(heart), arr.ind = TRUE)
```

A matrix:
0 × 2 of
type int

row	col

It can be seen that this data has no missing values.

In [17]:

```
### Converting the disease column in to a factor
heart$disease <- as.factor(heart$disease)
```

B. Produce a table of summary statistics on the data set. How do the ranges of the values in the columns compare? Does each column of data have similar magnitudes and ranges? Are there any outliers?

In [18]:

```
## summary statistics
summary(heart)
```

```
      age             sex             pain             bp             chol      
 Min.   :29.00   Min.   :0.00   Min.   :1.000   Min.   : 94.0   Min.   :12
6.0  
 1st Qu.:48.00   1st Qu.:0.00   1st Qu.:3.000   1st Qu.:120.0   1st Qu.:21
1.0  
 Median :56.00   Median :1.00   Median :3.000   Median :130.0   Median :24
1.5  
 Mean   :54.48   Mean   :0.68   Mean   :3.153   Mean   :131.6   Mean   :24
6.9  
 3rd Qu.:61.00   3rd Qu.:1.00   3rd Qu.:4.000   3rd Qu.:140.0   3rd Qu.:27
5.2  
 Max.   :77.00   Max.   :1.00   Max.   :4.000   Max.   :200.0   Max.   :56
4.0  
     sugar             ecg              rate           angina      
 Min.   :0.0000   Min.   :0.0000   Min.   : 71.0   Min.   :0.0000  
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.8   1st Qu.:0.0000  
 Median :0.0000   Median :0.5000   Median :153.0   Median :0.0000  
 Mean   :0.1467   Mean   :0.9867   Mean   :149.7   Mean   :0.3267  
 3rd Qu.:0.0000   3rd Qu.:2.0000   3rd Qu.:166.0   3rd Qu.:1.0000  
 Max.   :1.0000   Max.   :2.0000   Max.   :202.0   Max.   :1.0000  
      stv             sts             mvn             thal      disease
 Min.   :0.00   Min.   :1.000   Min.   :0.00   Min.   :3.000   0:162  
 1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.00   1st Qu.:3.000   1:138  
 Median :0.80   Median :2.000   Median :0.00   Median :3.000          
 Mean   :1.05   Mean   :1.603   Mean   :0.67   Mean   :4.727          
 3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.00   3rd Qu.:7.000          
 Max.   :6.20   Max.   :3.000   Max.   :3.00   Max.   :7.000          
```

Columns like sex,sugar, angina and ecg are said to have nominal outcomes like(0,1) and (0,2). Pain on the otherhand shows a ordinal outcomes and these variables are said to be categorical data. Same can be said for other qualitative variables in out data.

bp, chol, considering the 3rd quartile and maximum depicts the existence of upper outliers. rate , considering the 1st quartile and minimum shows a lower outlier.

Age has no outliers.

C. Partition the heart data set so that 80% will be used for training and 20% will be used for testing your machine learning model.

In [24]:

```
install.packages('caret')
install.packages('ggplot2')
install.packages('lattice')
library(caret)
library(ggplot2)
library(lattice)
```

In [20]:

```
### Splitting the dataset
sp_data <- createDataPartition(y = heart$disease, p = 0.8, list = FALSE)
# sp_data
```

In [21]:

```
## The training and testing data
tr_data <- heart[sp_data,]
te_data <- heart[-sp_data,]
```

D. Using logistic regression as provided by the Caret library in R, develop a model to predict heart disease diagnosis based on the 13 features provided in the data set for each patient.

In [23]:

```
## Fitting a logistic regression model
log_model <- train(disease~., data = tr_data, method = 'glm', family = 'binomial')
log_model
```

```
Generalized Linear Model

241 samples
 13 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 241, 241, 241, 241, 241, 241, ...
Resampling results:

  Accuracy   Kappa
  0.8249149  0.6463876
```

E. Generate a confusion matrix using the data from your test set to show the accuracy of the model.

In [29]:

```
## predicting the test data
pred_test1 <- predict(object = log_model, newdata = te_data)
pred_test1
```

```
    0  1  1  0  0  1  1  0  1  1  0  0  0  1  0  0  0  1  0  0  1  1  1  1
    1  0  0  0  0  1  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1
    0  1  0  0  0  0  0  1  1  1  0
```

▶ **Levels**:

In [30]:

```
confusionMatrix(table(pred_test1, te_data$disease))
```

```
Confusion Matrix and Statistics


pred_test1  0  1
         0 29  9
         1  3 18

               Accuracy : 0.7966
                 95% CI : (0.6717, 0.8902)
    No Information Rate : 0.5424
    P-Value [Acc > NIR] : 4.294e-05

                  Kappa : 0.583

 Mcnemar's Test P-Value : 0.1489

            Sensitivity : 0.9062
            Specificity : 0.6667
         Pos Pred Value : 0.7632
         Neg Pred Value : 0.8571
             Prevalence : 0.5424
         Detection Rate : 0.4915
   Detection Prevalence : 0.6441
      Balanced Accuracy : 0.7865

       'Positive' Class : 0
```

F. Write a few sentences providing commentary on the accuracy of the model. What percent are false positives? What percent are false negatives?

The accuracy of the log_model has a **79.66%** classification rate. From the confusion matrix above, we can see that the model correctly classified 29 non diseased patients as no disease whiles it wrongly classified 9 not diseased patients as disease. The model also wrongly classified 3 diseased patients as no disease whiles it correctly classified 18 diseased patients as disease.

In [32]:

```
conf_table <- table(pred_test1, te_data$disease)
conf_table
```

```
pred_test1  0  1
         0 29  9
         1  3 18
```

In [40]:

```
false_pos <- conf_table[2,1]
per_fp <- round((false_pos/sum(conf_table[2,]))*100,2)
per_fp
```

14.29

The percent of false positives are **14.29%**.

In [41]:

```
false_neg <- conf_table[1,2]
per_np <- round((false_neg/sum(conf_table[1,]))*100,2)
per_np
```

```
23.68
```

The percent of false negatives are **23.68%**.