Binary and Multinomial Logistic Regression

Yin Zehui

Binary and Multinomial Logistic Regression

Part 1: Binary Logistic Regression

First, I set working directory.

```
setwd("C:/Users/zehuiyin/Desktop/analysis")
```

Then, I load the diabetes csv file and generate some descriptive statistics of it.

```
dia <- read.csv("diabetes.csv")
summary(dia)
```

```
##        X            pregnant         glucose         pressure     
##  Min.   :  4.0   Min.   : 0.000   Min.   : 56.0   Min.   : 24.00  
##  1st Qu.:204.8   1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00  
##  Median :385.5   Median : 2.000   Median :119.0   Median : 70.00  
##  Mean   :388.1   Mean   : 3.301   Mean   :122.6   Mean   : 70.66  
##  3rd Qu.:568.2   3rd Qu.: 5.000   3rd Qu.:143.0   3rd Qu.: 78.00  
##  Max.   :766.0   Max.   :17.000   Max.   :198.0   Max.   :110.00  
##     triceps         insulin            mass          pedigree     
##  Min.   : 7.00   Min.   : 14.00   Min.   :18.20   Min.   :0.0850  
##  1st Qu.:21.00   1st Qu.: 76.75   1st Qu.:28.40   1st Qu.:0.2697  
##  Median :29.00   Median :125.50   Median :33.20   Median :0.4495  
##  Mean   :29.15   Mean   :156.06   Mean   :33.09   Mean   :0.5230  
##  3rd Qu.:37.00   3rd Qu.:190.00   3rd Qu.:37.10   3rd Qu.:0.6870  
##  Max.   :63.00   Max.   :846.00   Max.   :67.10   Max.   :2.4200  
##       age          diabetes        
##  Min.   :21.00   Length:392        
##  1st Qu.:23.00   Class :character  
##  Median :27.00   Mode  :character  
##  Mean   :30.86                     
##  3rd Qu.:36.00                     
##  Max.   :81.00                     
```

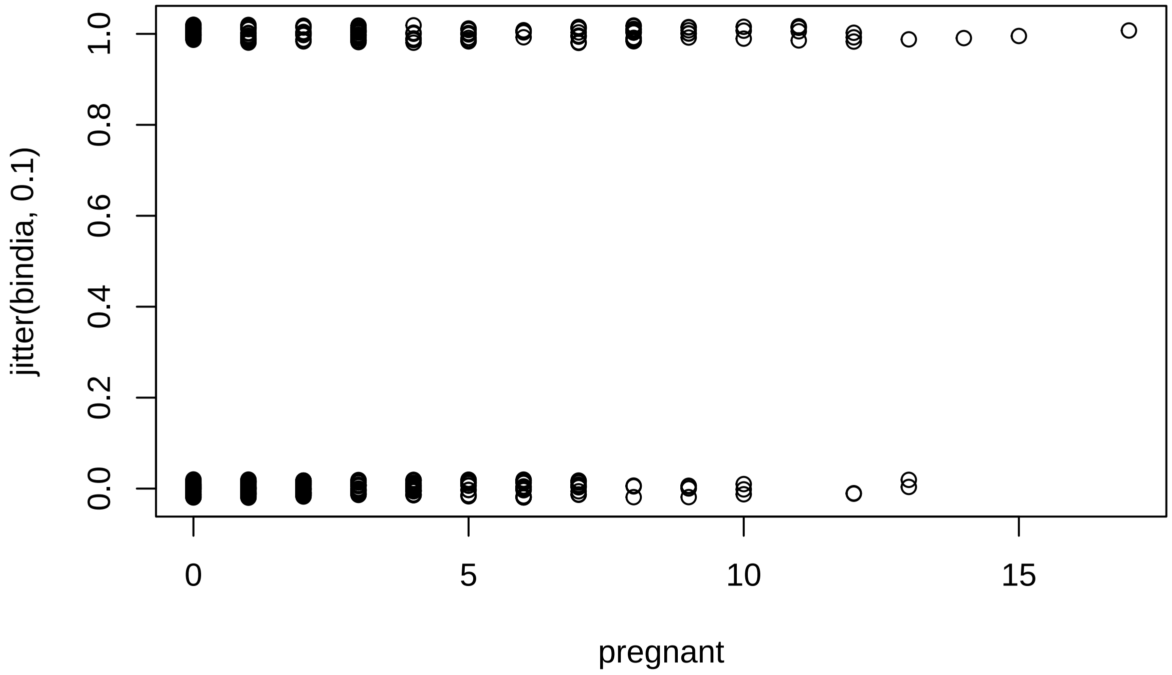
The information about whether the patients have diabetes or not is stored in the variable called "diabetes" with characters. I convert that variable into binary form.

```
dia$bindia[dia$diabetes == 'pos'] = 1
dia$bindia[dia$diabetes == 'neg'] = 0
attach(dia)
```

```
## The following object is masked from package:datasets:
##
##     pressure
```

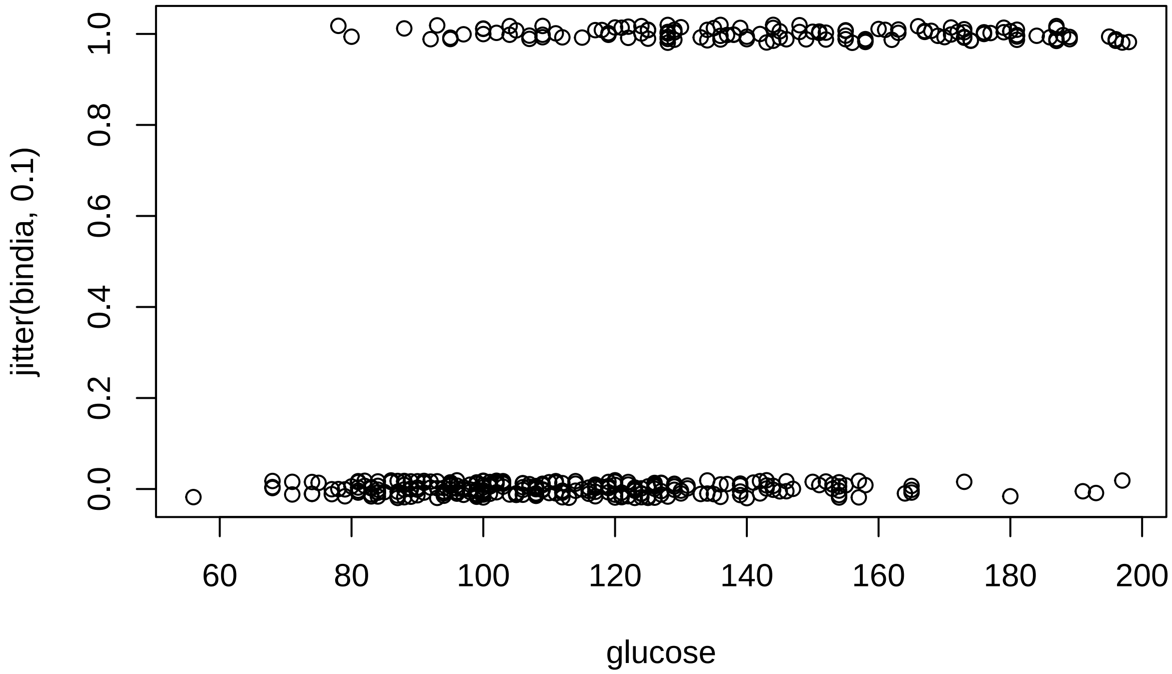
Since I am interested in predict whether a patient has diabetes or not. I plot the dependent variables against other variables to see which independent variable seem to have correlation with the dependent variable.

```
plot(pregnant, jitter(bindia, 0.1))
```



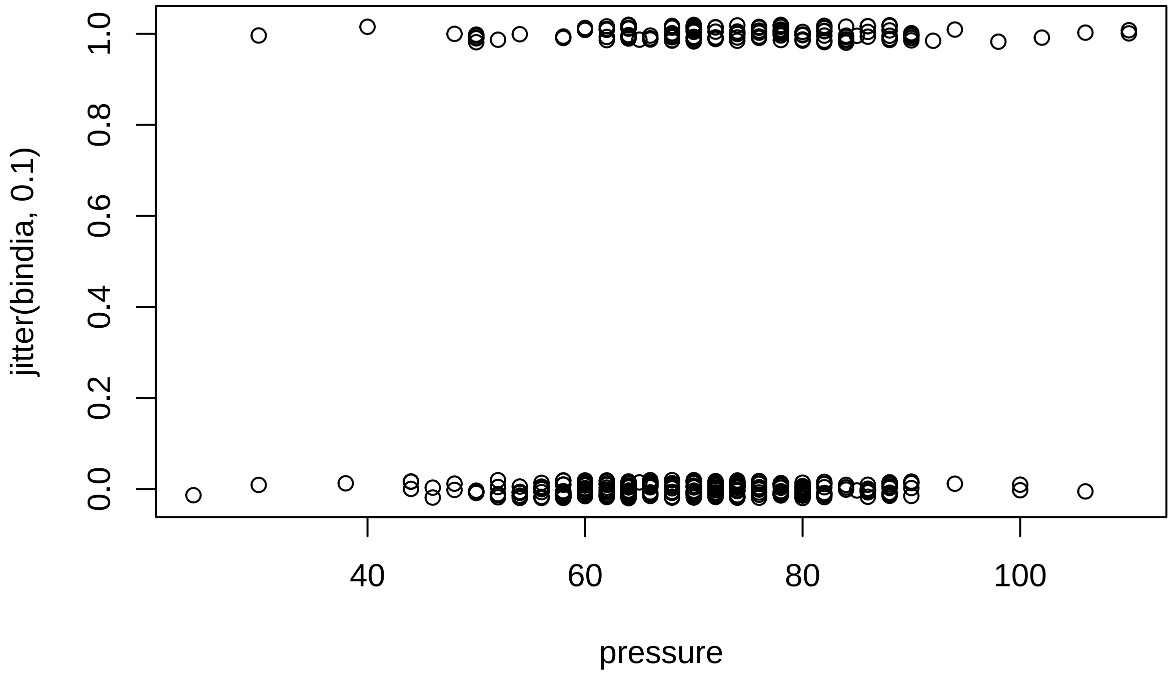
Based on the plot above, diabetes against pregnant. The amount of points at 0 and 1 at the same pregnant level are generally identical. Therefore, the variable pregnant is not correlated with diabetes.

```
plot(glucose, jitter(bindia, 0.1))
```



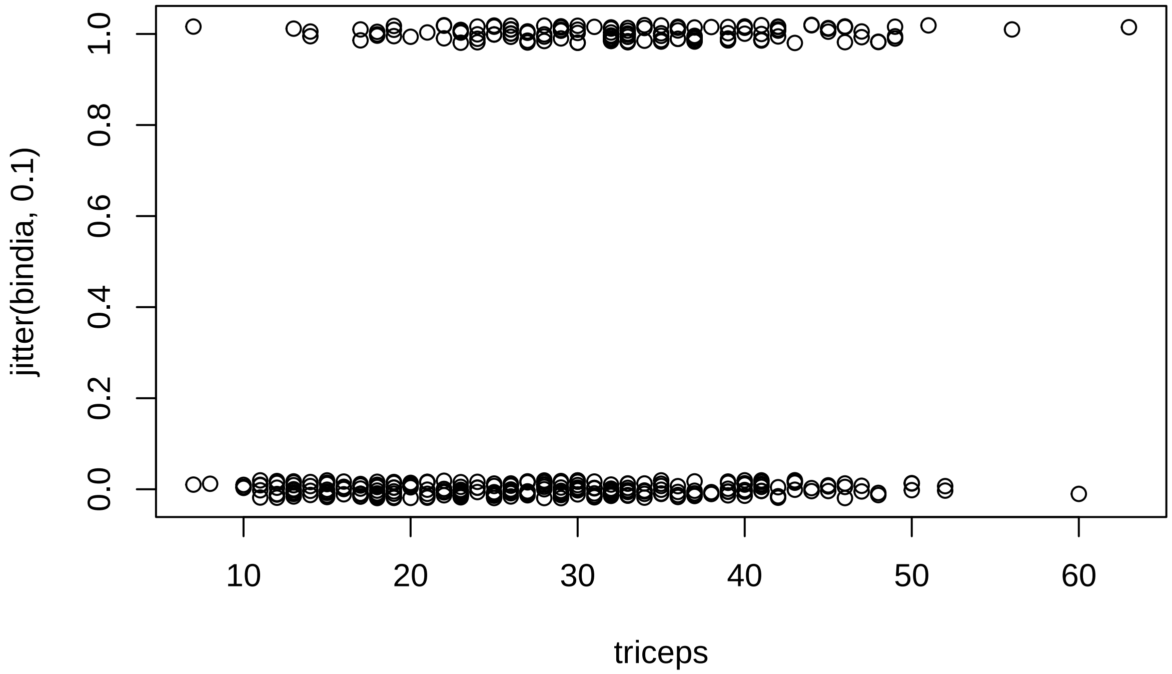
Based on the plot above, diabetes against glucose. There are a large amount of points clustered at the bottom left of the plot, therefore person without diabetes tend to have lower glucose level. Persons with diabetes tend to have higher level of glucose than the persons without. Based on this plot, the diabetes is correlated with glucose level.

```
plot(pressure, jitter(bindia, 0.1))
```



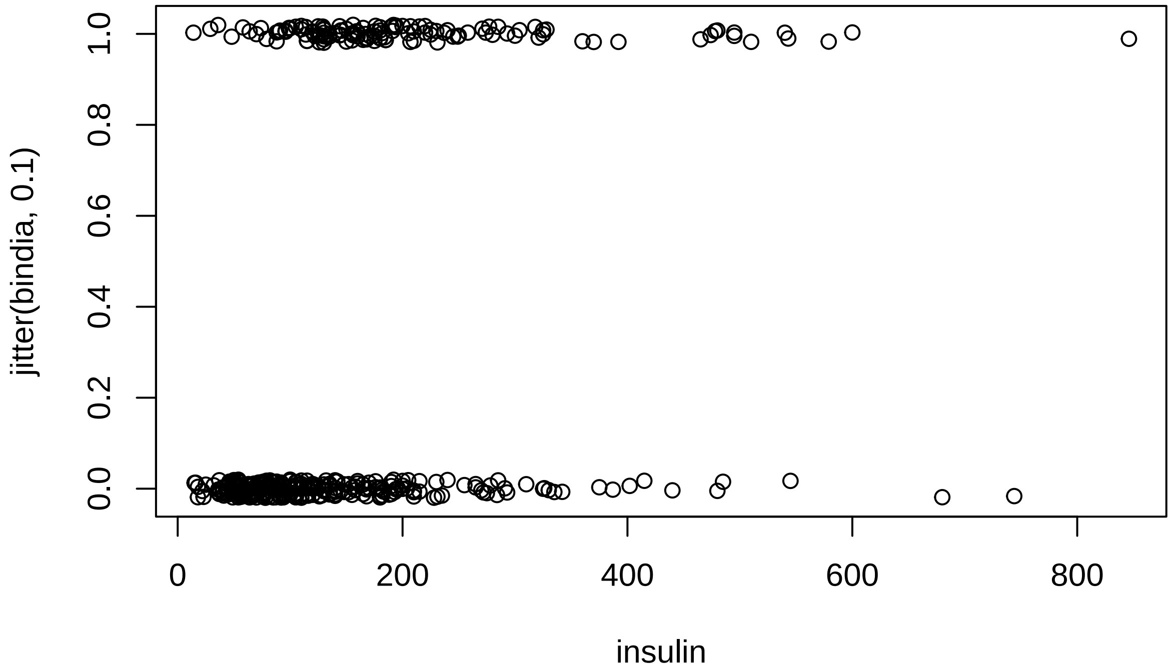
Based on the plot above, diabetes against pressure. The amount of points at 0 and 1 are relative identical at the same level of pressure. Therefore, the variable pressure is not correlated with diabetes.

```
plot(triceps, jitter(bindia, 0.1))
```



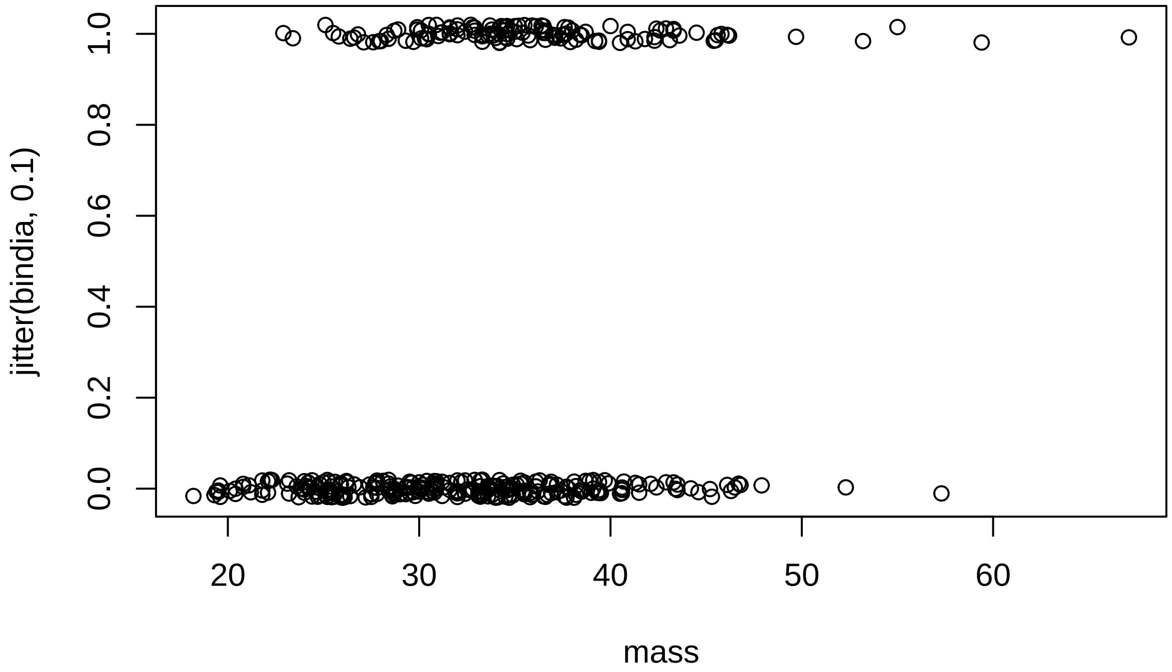
Based on the plot above, diabetes against skin fold thickness. There are more points at 0 level at the low level of skin fold thickness, the persons without diabetes tend to have relative low level of skin fold thickness. The persons with diabetes tend to have relative higher with smaller variance skin fold thickness. Therefore, the skin fold thickness is correlated with diabetes but less correlated than the glucose level with diabetes.

```
plot(insulin, jitter(bindia, 0.1))
```



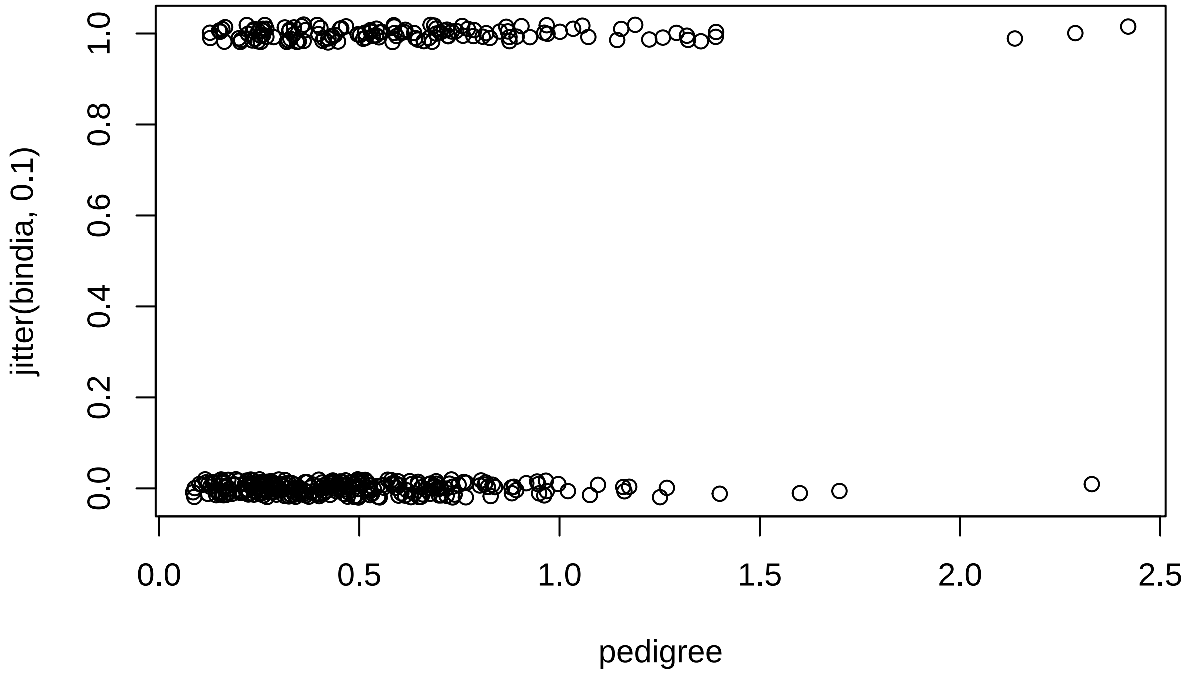
Based on the plot above, diabetes against insulin level. There are a lot of points clustered at the bottom left of the plot, the persons without diabetes tend to have lower insulin level. The persons with diabetes looks like tend to have slight higher insulin level. Therefore, the insulin is also slightly correlated with diabetes.

```
plot(mass, jitter(bindia, 0.1))
```



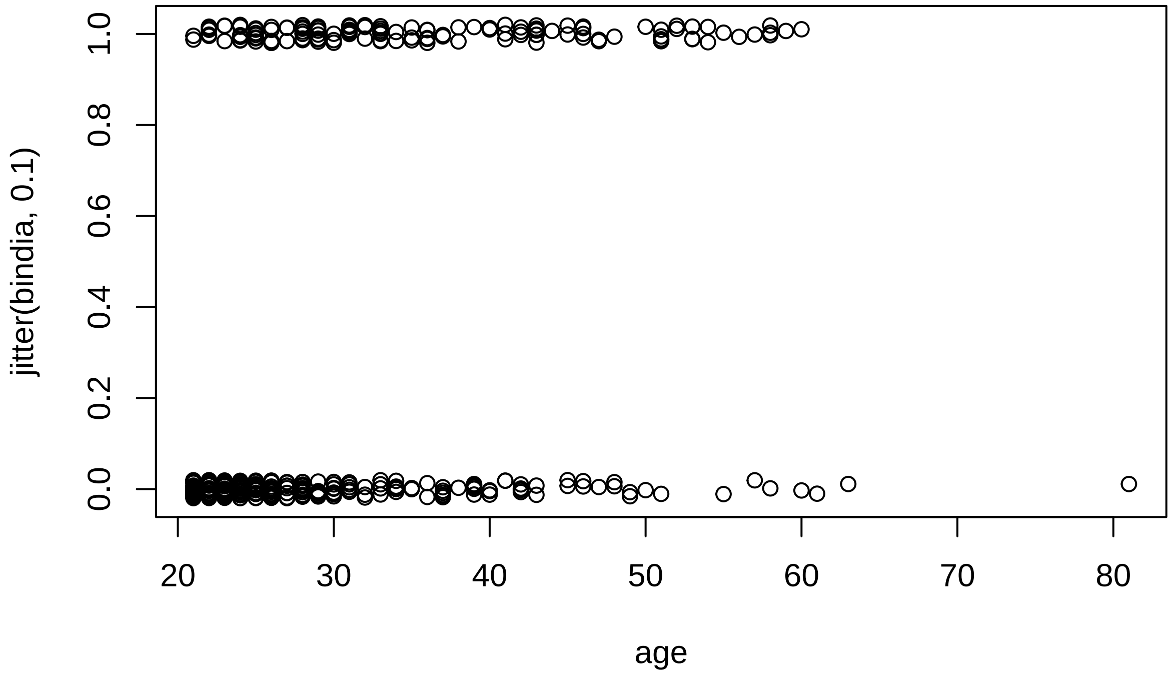
Based on the plot above, diabetes against mass. The variance of persons who do not have diabetes is much higher than the variance of the persons with diabetes. It looks like that the persons with diabetes are more likely to have high body mass index. Therefore, the body mass index is also correlated with diabetes.

```
plot(pedigree, jitter(bindia, 0.1))
```



Based on the plot above, diabetes against pedigree. The differences between persons with or without diabetes on this plot is not very significant. It seems that persons with out diabetes more tend to have low pedigree but again not very significant. I consider that the correlation between these two variables are relative small.

```
plot(age, jitter(bindia, 0.1))
```

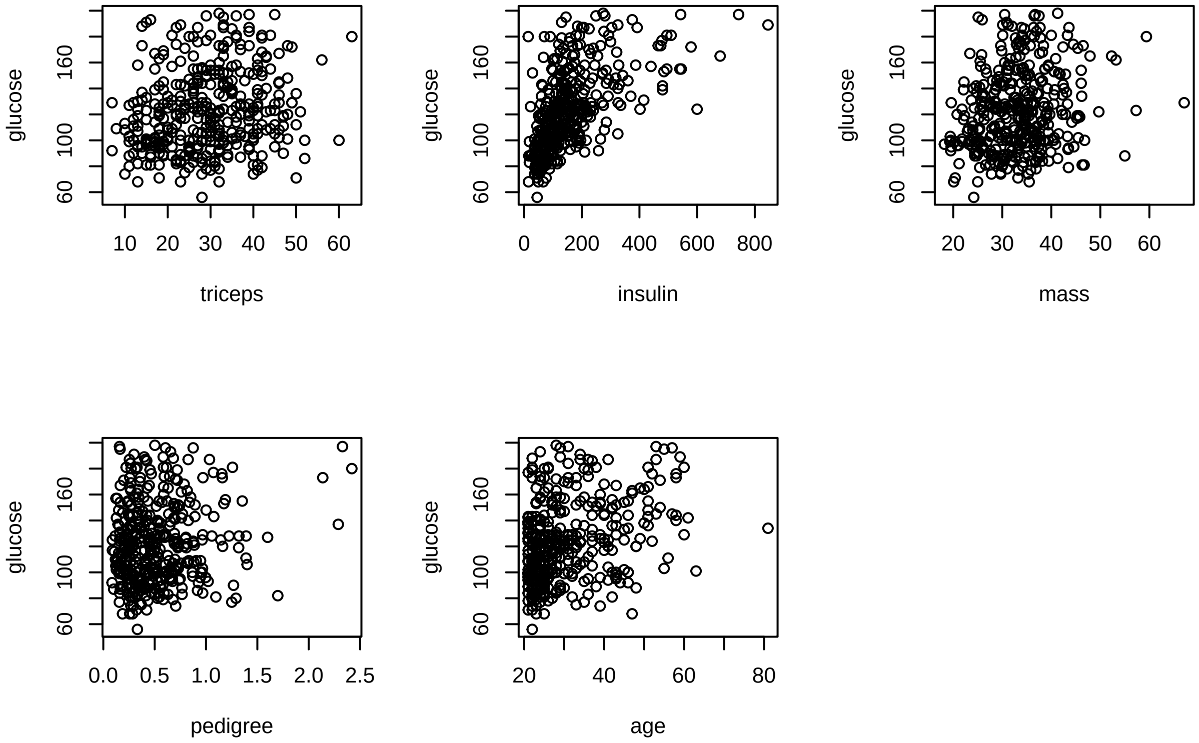


Based on the plot above, diabetes against age. The young people looks less likely to have diabetes and old people more tend to have diabetes. However, in the group of diabetes patients, the age distribution is relative more uniform. I would consider the correlation between these two variables are relative small.

Based on the previous plots, the variables seems to have some level of correlation with diabetes are: glucose, triceps, insulin, mass, pedigree, age. I then check whether there is correlation between these variables.

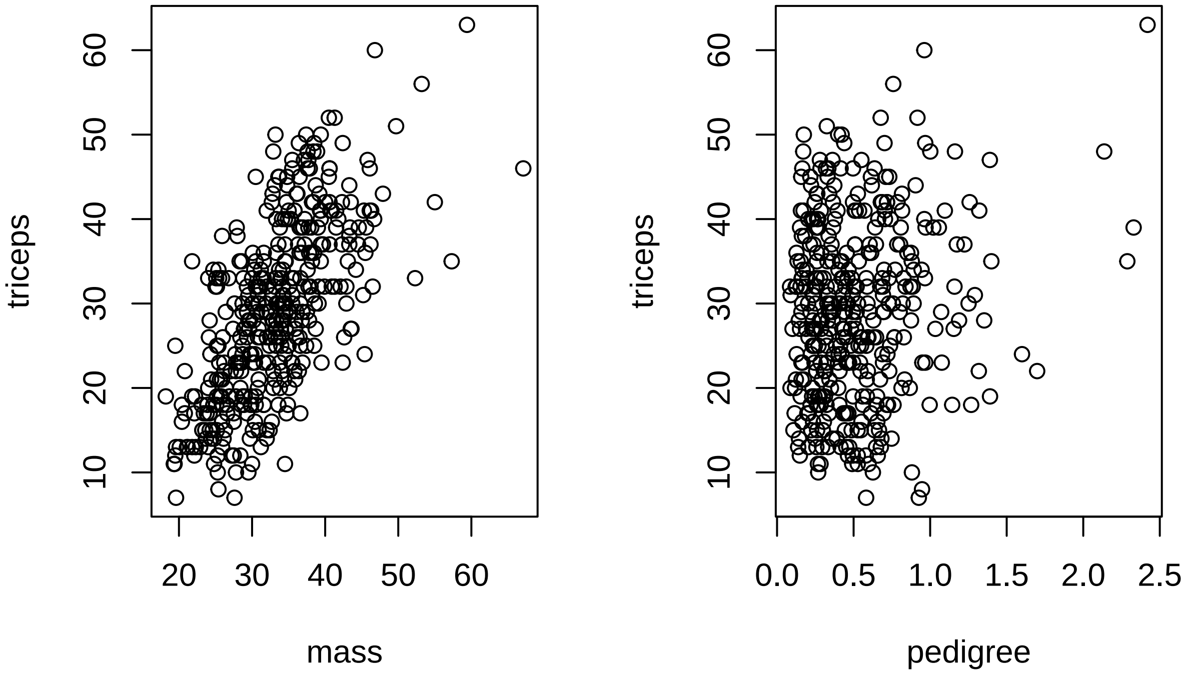
```
par(mfrow=c(2,3))
plot(glucose~triceps)
plot(glucose~insulin)
plot(glucose~mass)
plot(glucose~pedigree)
plot(glucose~age)
cor(glucose*3,cbind(triceps,insulin,mass,pedigree,age))
```

```
##        triceps  insulin      mass  pedigree       age
## [1,] 0.1988558 0.581223 0.2095159 0.1401802 0.3436415
```



Based on the plot above, the glucose is highly correlated with insulin and slightly correlated with age. Since the glucose has the highest correlation with diabetes based on the previous results, I would omit these two other correlated variables. Remove insulin and age from the independent variables list used in model.

```
par(mfrow=c(1,2))
plot(triceps~mass)
plot(triceps~pedigree)
```

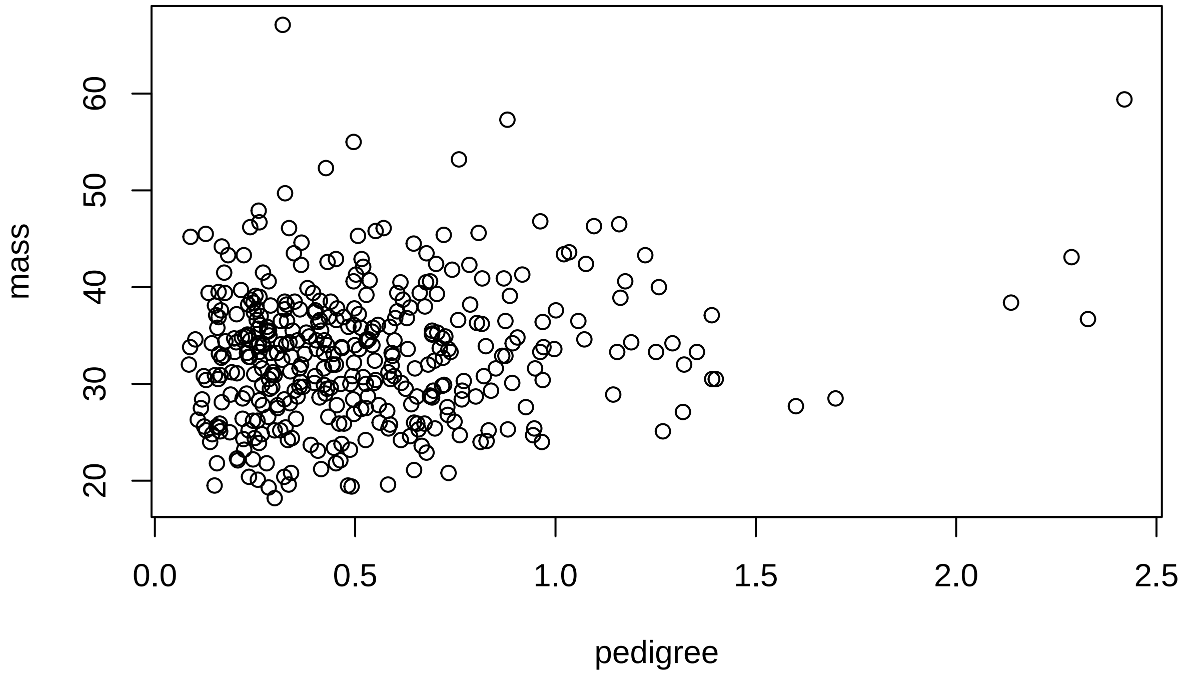


```
cor(triceps*2, cbind(mass, pedigree))
```

```
##           mass  pedigree
## [1,] 0.6643549 0.1604985
```

Based on the plots above, the variable triceps is highly correlated with mass. Based on the previous results, variable mass seems to have higher correlation with diabetes. I would remove triceps from the independent variable list used in model.

```
plot(mass~pedigree)
```

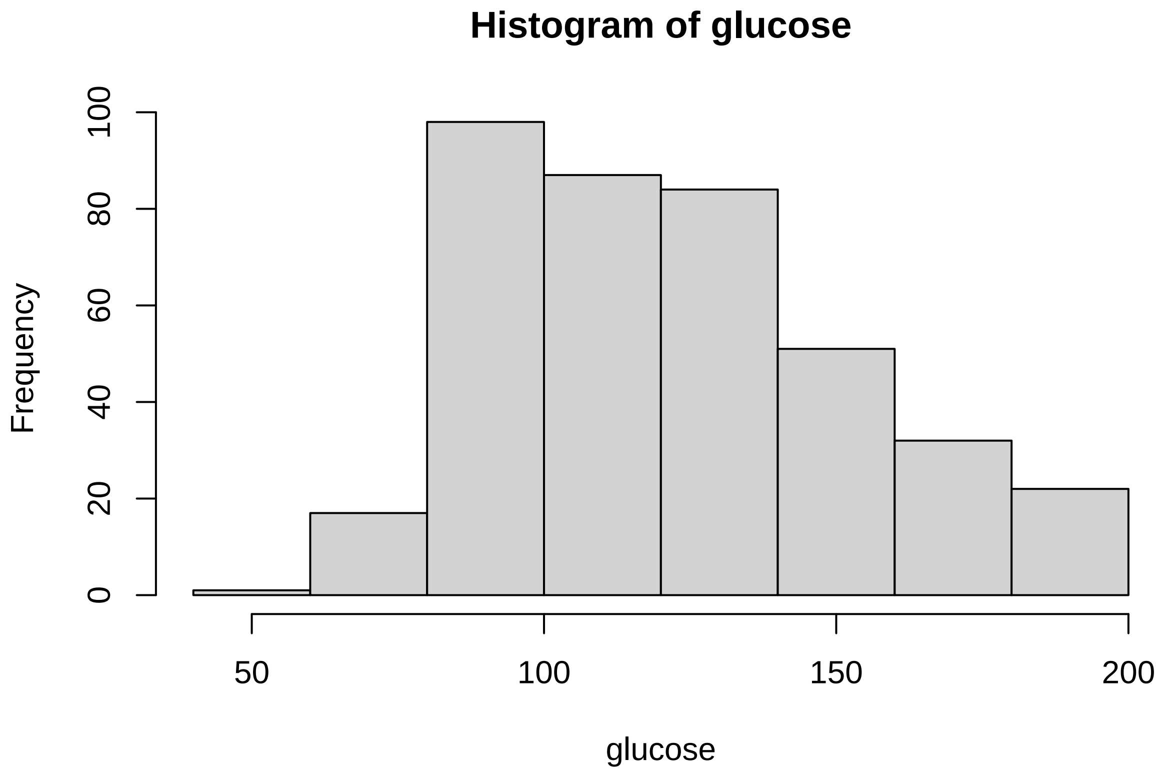


```
cor(mass,pedigree)
```

```
## [1] 0.158771
```

Based on the plot above, the two variables mass and pedigree is not correlated with each other. Then, the remaining independent variables are glucose, mass, pedigree. Then I generate descriptive statistic on these three variables and dependent variable.

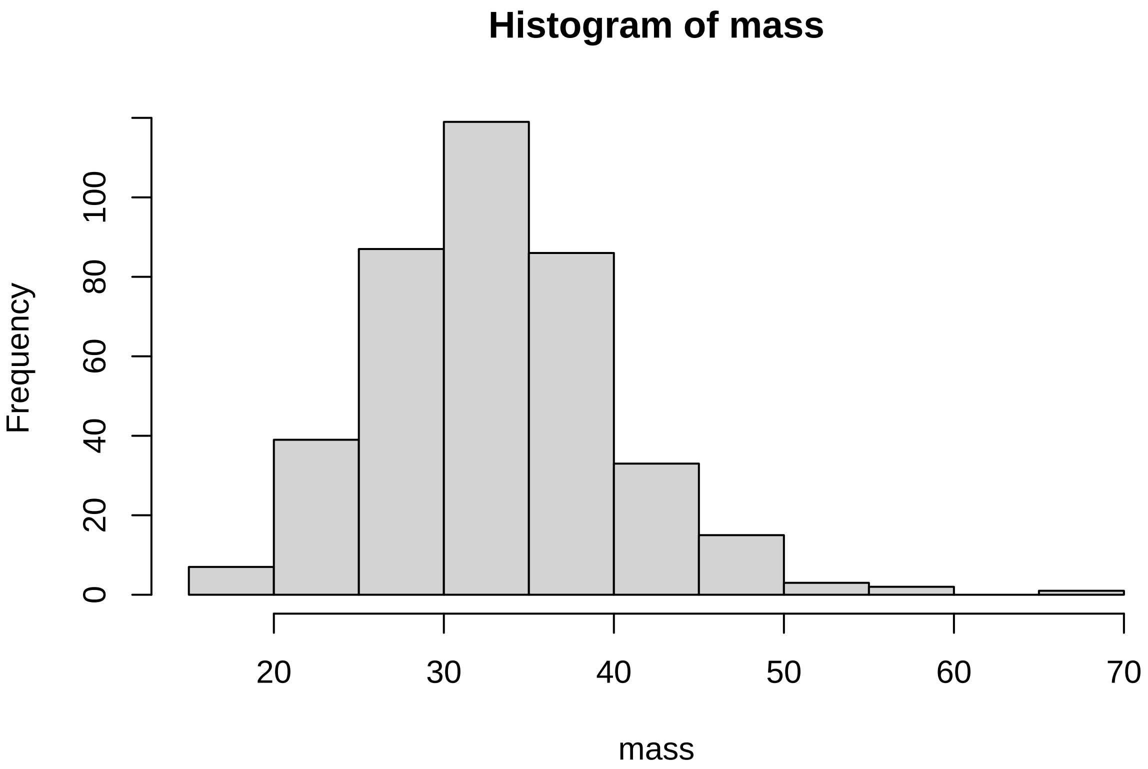
```
hist(glucose)
```



```
summary(glucose)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    56.0    99.0   119.0   122.6   143.0   198.0
```

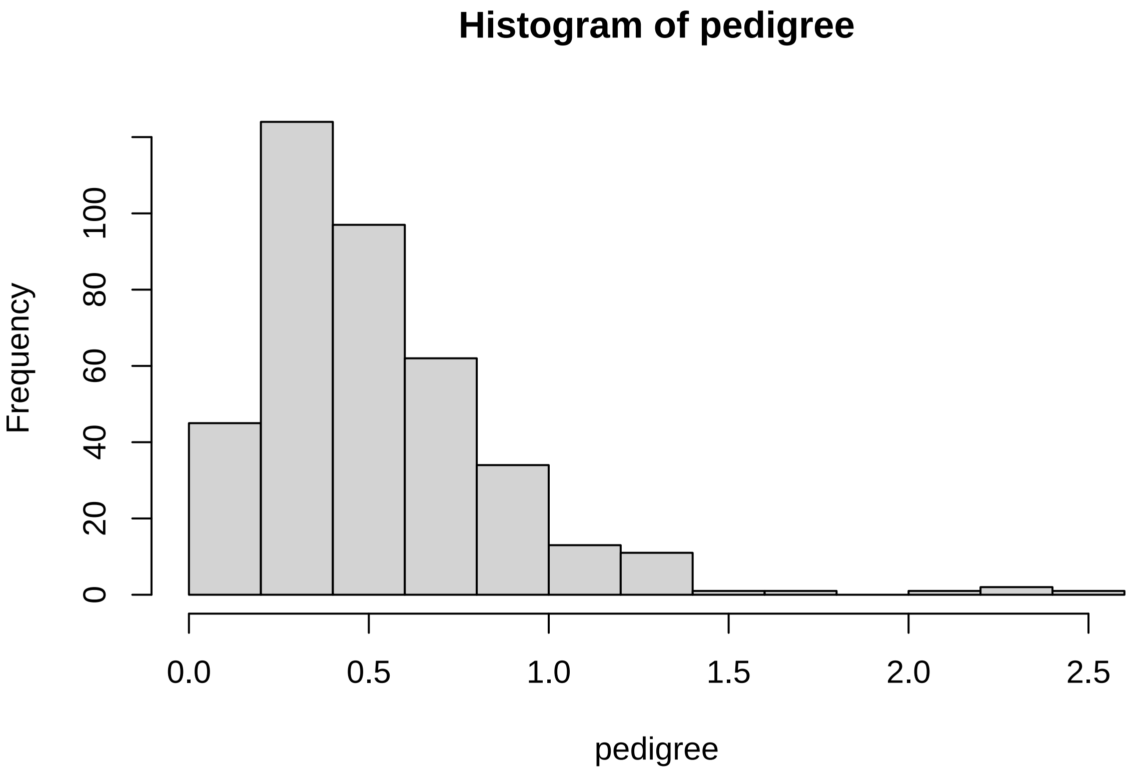
```
hist(mass)
```



```
summary(mass)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.20   28.40   33.20   33.09   37.10   67.10
```

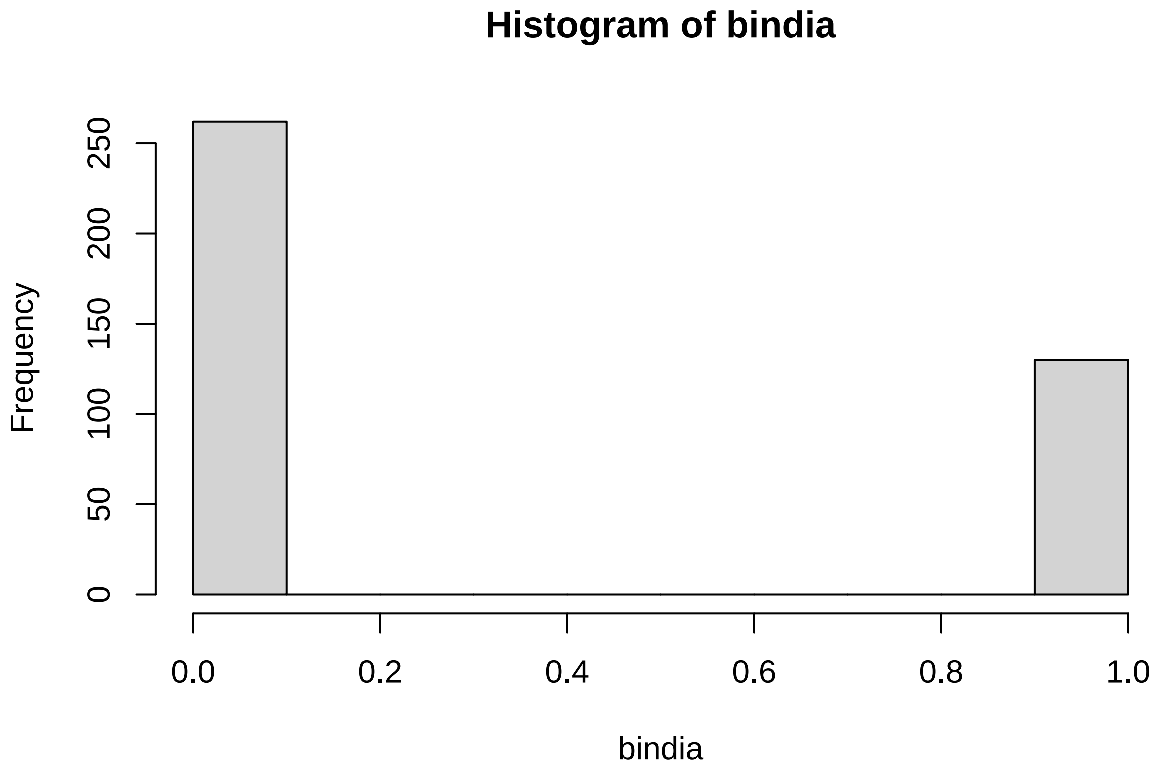
```
hist(pedigree)
```



```
summary(pedigree)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0850  0.2697  0.4495  0.5230  0.6870  2.4200
```

```
hist(bindia)
```



```
summary(bindia)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3316  1.0000  1.0000
```

Based on the plots above, all three independent variables are general normally distributed with some right skewdness. The variable pedigree has the highest level of left skewdness. The variable glucose has the highest median 119 and range 142. There are more people do not have diabetes than people have diabetes in the data set.

Then, I fit binary logistic regression model with these variables.

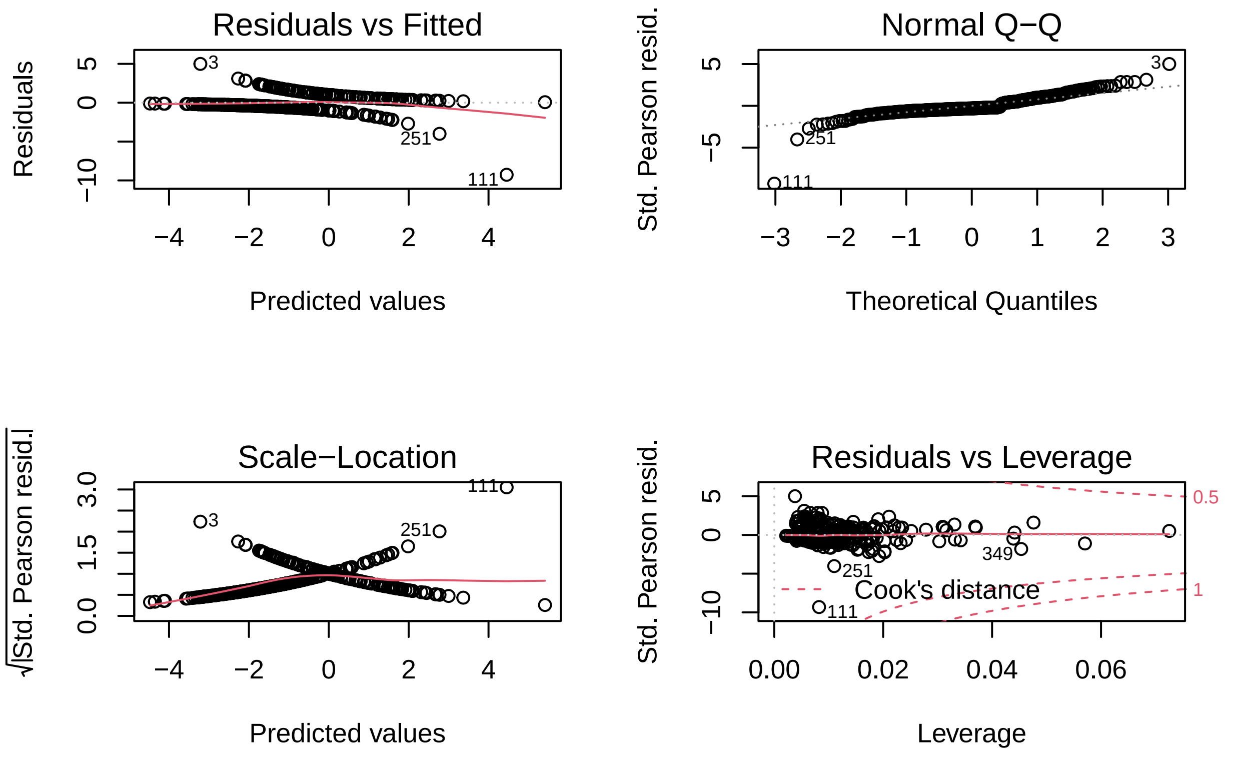
```
m1 <- glm(bindia ~ glucose + mass + pedigree, family = binomial(link = 'logit'))
summary(m1)
```

```
##
## Call:
## glm(formula = bindia ~ glucose + mass + pedigree, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9885  -0.6873  -0.4256   0.6689   2.5516
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.791977   0.972850  -9.037  < 2e-16 ***
## glucose      0.040708   0.004891   8.324  < 2e-16 ***
## mass         0.068073   0.019819   3.435 0.000593 ***
## pedigree     1.171465   0.414149   2.829 0.004675 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 498.1  on 391  degrees of freedom
## Residual deviance: 363.7  on 388  degrees of freedom
## AIC: 371.7
##
## Number of Fisher Scoring iterations: 5
```

All three independent variables have extreme small z-test p value which indicate that all three variables are significant in the model.The residual deviance 363.7 is also significantly smaller than null deviance 498.1 which indicate good validity of the model.

Plot basic diagnostic plots.

```
par(mfrow=c(2,2))
plot(m1)
```



Based on the qq plot, the residuals generally fits normal distribution with slightly off at the two tail area. Based on the right bottom corner plot, there is no bad leverage point in the model that have great leverage and deviance at the same time. Based on the left two plots, there are some pattern in the residuals plots which is inevitable for binary logistic regression. Generally the residuals are quite small. In summary, the diagnostic plots shows that the model have relative good validity.

Conduct an overall test of model fit using the deviance.

```
chisq<-m1$null.deviance-m1$deviance
pchisq(chisq,3,lower.tail=FALSE)
```

```
## [1] 6.103366e-29
```

The result is almost 0. It indicate that the logistic regression have extreme high chance better than the null model.

Generate deviance and pearson residuals of the model.

```
m1.dev_residuals = residuals(m1, type="deviance")
m1.pearson_residuals = residuals(m1, type="pearson")
```

Calculate the Deviance and Pearson X2 statistics of model fit from these.

```
sum(m1.dev_residuals^2)
```

```
## [1] 363.7016
```

```
sum(m1.pearson_residuals^2)
```

```
## [1] 445.5269
```

The deviance of the model calculated from deviance residuals is relative small. Also, the the Pearson Chi-Square calculated from pearson_residuals is relative small. They both indicate good fitness of the model.

The m1 model seems fits the data greatly according to the diagnostic tests. I then try to further improve the model by remove the variable with the largest z-test p value which is pedigree.

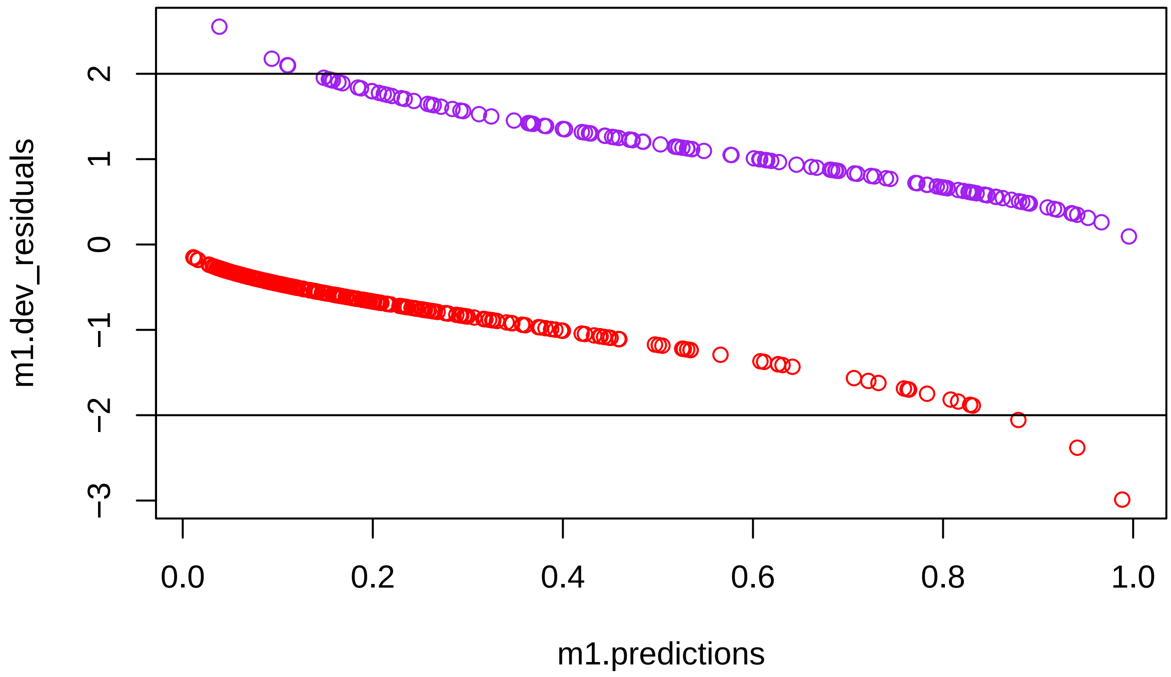
```
m2 <- glm(bindia ~ glucose + mass, family = binomial(link = 'logit'))
summary(m2)
```

```
##
## Call:
## glm(formula = bindia ~ glucose + mass, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2112  -0.7396  -0.4114   0.7009   2.4306
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.301460   0.927405  -8.951  < 2e-16 ***
## glucose      0.040713   0.004825   8.437  < 2e-16 ***
## mass         0.071794   0.019606   3.662  0.00025 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 498.10  on 391  degrees of freedom
## Residual deviance: 372.12  on 389  degrees of freedom
## AIC: 378.12
##
## Number of Fisher Scoring iterations: 4
```

The reduced model has higher residual deviance 372.12 and higher AIC 378.12, which indicate that the reduced model is worse fitted than the original model. Therefore, I would stick to original model m1 as my final model.

I then try to visualize my final model residuals.

```
m1.predictions <- predict(m1, type="response")
plot(m1.predictions,m1.dev_residuals, col=c("red","purple")[1+m1$y])
abline(2,0)
abline(-2,0)
```



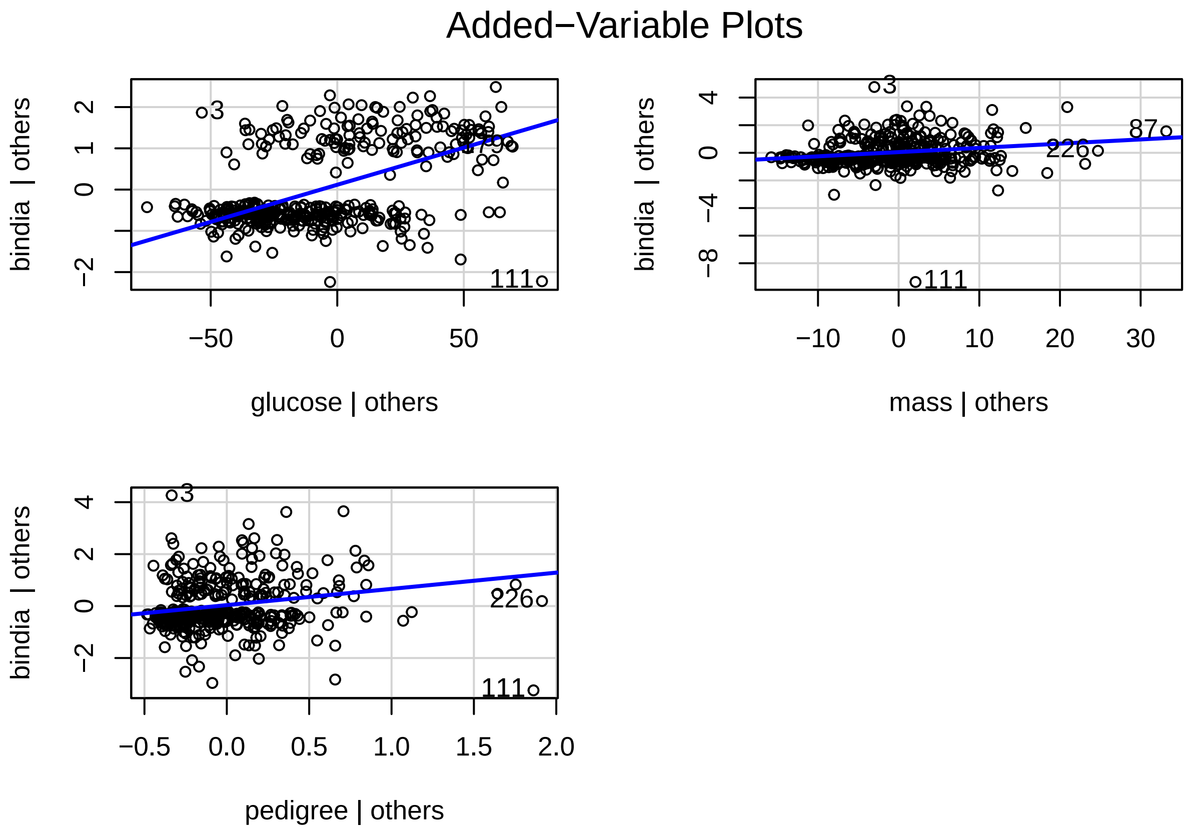
The residual plot follows the two line shape which is normal in a binary logistic regression model and the most of the residuals fall into the -2 to 2 range which is also a good sign that my model fitted the data well.

After that, I also tried the added-variable plots.

```
library(car)
```

```
## Loading required package: carData
```

```
avPlots(m1)
```



Based on the plot, the variable mass seem have little impact to the model. I try to improve the model by reduce variable mass from the model.

```
m3 <- glm(bindia ~ glucose + pedigree)
summary(m3)
```

```
##
## Call:
## glm(formula = bindia ~ glucose + pedigree)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.23968  -0.27077  -0.09759   0.27365   1.05902
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.6974108  0.0850704  -8.198 3.58e-15 ***
## glucose      0.0075781  0.0006614  11.458  < 2e-16 ***
## pedigree     0.1907298  0.0590764   3.229  0.00135 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for gaussian family taken to be 0.1596802)
##
##     Null deviance: 86.888  on 391  degrees of freedom
## Residual deviance: 62.116  on 389  degrees of freedom
## AIC: 398.28
##
## Number of Fisher Scoring iterations: 2
```

Based on the above results, the AIC is higher than the original model, which indicate the original model is better. However, the difference between null and residual residuals are smaller in this model, which may indicate opposite message.

To better check which model is better, I also conduct overall test of model fit using the deviance on m3. Also calculate the pseudo Rˆ2 for two model.

```
chisqm3<-m3$null.deviance-m3$deviance
pchisq(chisqm3,2,lower.tail=FALSE)
```

```
## [1] 4.176339e-06
```

```
m1_pseudoR <- 1 - (m1$deviance^2/m1$null.deviance^2)
m1_pseudoR
```

```
## [1] 0.4668355
```

```
m3_pseudoR <- 1 - (m3$deviance^2/m3$null.deviance^2)
m3_pseudoR
```

```
## [1] 0.4889254
```

The test p value is much higher than the original model. The pseudo Rˆ2 is rather similar in two models. According to the above results, most of them indicate the original model is better, therefore I consider that the original model is relative better than the reduced model. I would still stick to m1 as my final model.

Since I have conducted all the diagnostics and tried to improve my model by reduce some variables, the final model I come to is m1. Here I would do interpretation of the model.

```
summary(m1)
```

```
##
## Call:
## glm(formula = bindia ~ glucose + mass + pedigree, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9885  -0.6873  -0.4256   0.6689   2.5516
##
## Coefficients:
```

```
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.791977   0.972850  -9.037  < 2e-16 ***
## glucose      0.040708   0.004891   8.324  < 2e-16 ***
## mass         0.068073   0.019819   3.435 0.000593 ***
## pedigree     1.171465   0.414149   2.829 0.004675 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 498.1  on 391  degrees of freedom
## Residual deviance: 363.7  on 388  degrees of freedom
## AIC: 371.7
##
## Number of Fisher Scoring iterations: 5
```

Formula:

$$ln(\frac{\theta(x)}{1-\theta(x)}) = -8.791977 + 0.040708 * glucose + 0.068073 * mass + 1.171465 * pedigree$$

```
exp(coef(m1))
```

```
##  (Intercept)      glucose         mass     pedigree
## 0.0001519473 1.0415479391 1.0704432675 3.2267179061
```

```
exp(confint(m1))
```

```
## Waiting for profiling to be done...
```

```
##                    2.5 %       97.5 %
## (Intercept) 2.041603e-05 0.0009343387
## glucose     1.031997e+00 1.0520286705
## mass        1.030562e+00 1.1140945693
## pedigree    1.453547e+00 7.3829820192
```

Holding all else constant, for 1 unit increase in plasma glucose concentration, we would expect to see 1.0415479391 times increase in odds of being diabetes.

Holding all else constant, for 1 unit increase in plasma glucose concentration, we are 95 percent confident that we would expect to see 1.031997e+00 to 1.0520286705 times increase in odds of being diabetes.

Holding all else constant, for 1 unit increase in body mass index, we would expect to see 1.0704432675 times increase in odds of being diabetes.

Holding all else constant, for 1 unit increase in body mass index, we are 95 percent confident that we would expect to see 1.030562e+00 to 1.1140945693 times increase in odds of being diabetes.

Holding all else constant, for 1 unit increase in diabetes pedigree function, we would expect to see 3.2267179061 times increase in odds of being diabetes.

Holding all else constant, for 1 unit increase in diabetes pedigree function, we are 95 percent confident that we would expect to see 1.453547e+00 to 7.3829820192 times increase in odds of being diabetes.

The odds here is

$$\frac{\theta(x)}{1-\theta(x)}$$

Since I assigned 1 for being tested positive for diabetes, the odd is the ratio of the possibility of being diabetes and the possibility of not being diabetes. This odds make sense, because it indicate the magnitude relationship between possibility of being diabetes and not being diabetes. If the odds equals 1 indicate that the person have equal chance of being diabetes or not. If the odds greater than 1 indicate that the person has more chance having diabetes. If the odds less than 1 indicate that the person has less chance having diabetes.

Clear work space and detach dataset.

```
detach(dia)
rm(list = ls(all.names = TRUE))
```

Part 2: Multinomial Logistic Regression

Load library.

```
library(foreign)
library(nnet)
library(ggplot2)
library(reshape2)
```

Load dataset.

```
data(iris)
```

I firstly generate some descriptive statistics of the dataset.

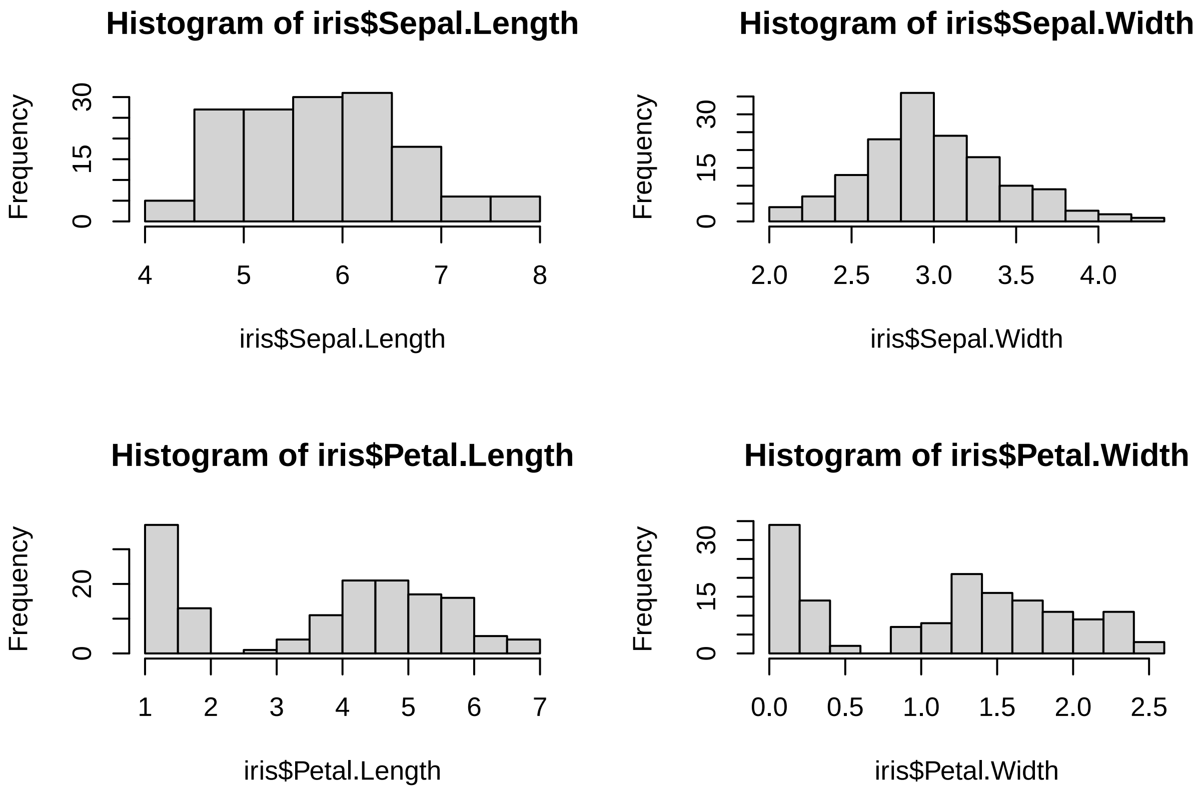
```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
## 
```

There are equally 50 observations in each species. The variable Petal.Length has the largest range of 5.9. The variables Sepal.Width and Petal.Width have the smallest range of 2.4. The variable Sepal.Length has the highest median of 5.8, while the variable Petal.Width has the smallest median of 1.3.

I then generate plots to check the distributions of the numeric variables.

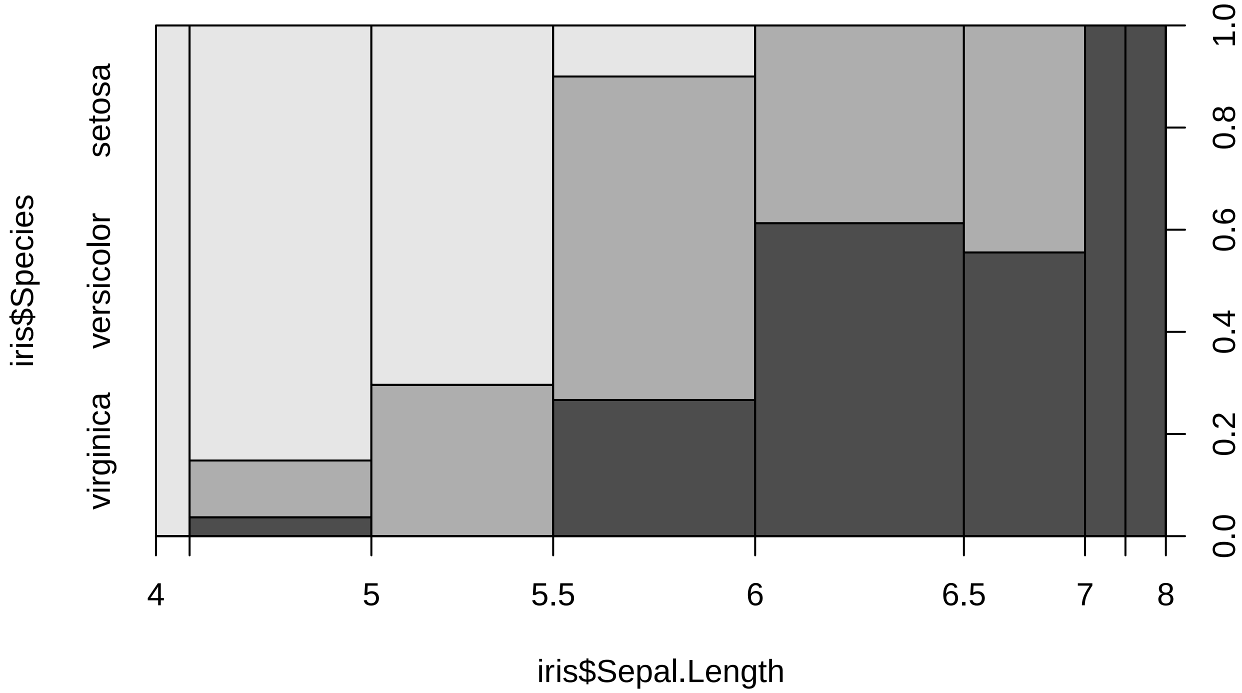
```
par(mfrow=c(2,2))
hist(iris$Sepal.Length)
hist(iris$Sepal.Width)
hist(iris$Petal.Length)
hist(iris$Petal.Width)
```



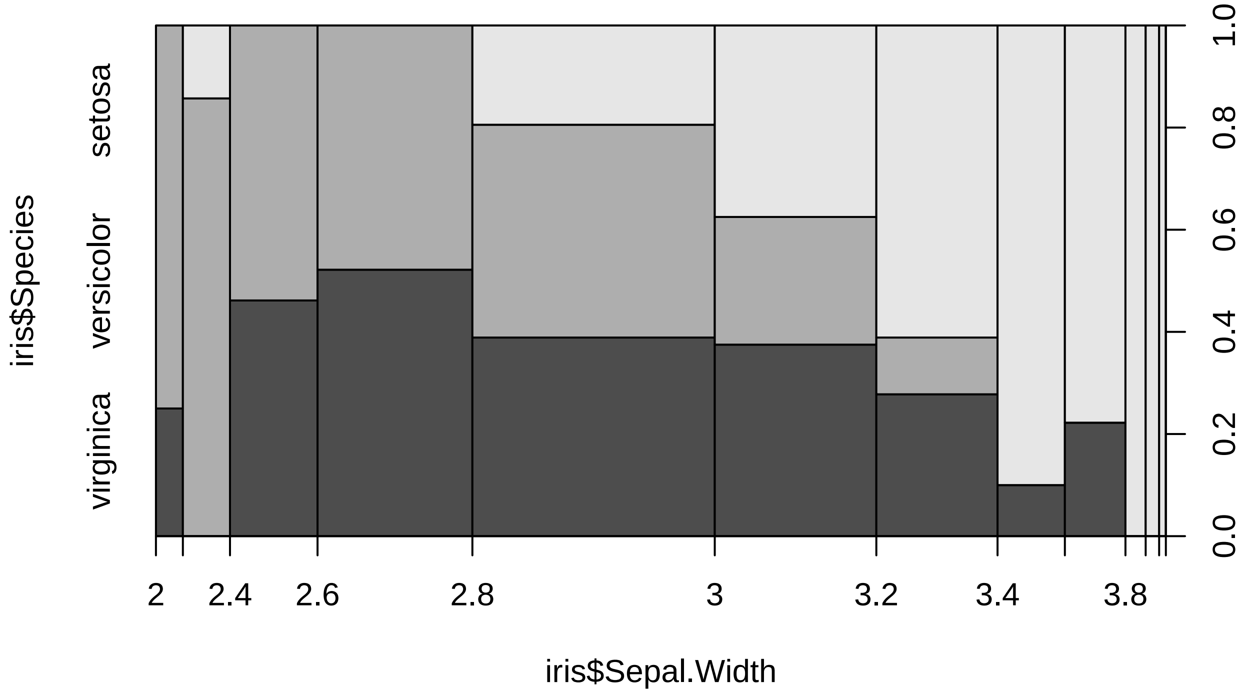
The variables Sepal.Length and Sepal.Width are generally normally distributed without significant skewdness. However, the variables Petal.Length and Petal.Width are bimodal with two peaks. These two variables both have one peak at 0 and the other at higher level. The general distribution patterns of variables Petal.Length and Petal.Width are very similar.

Since we are interested in predict the probability of being each type of species, I check the relations between these numeric variables and the categorical variable of interest.

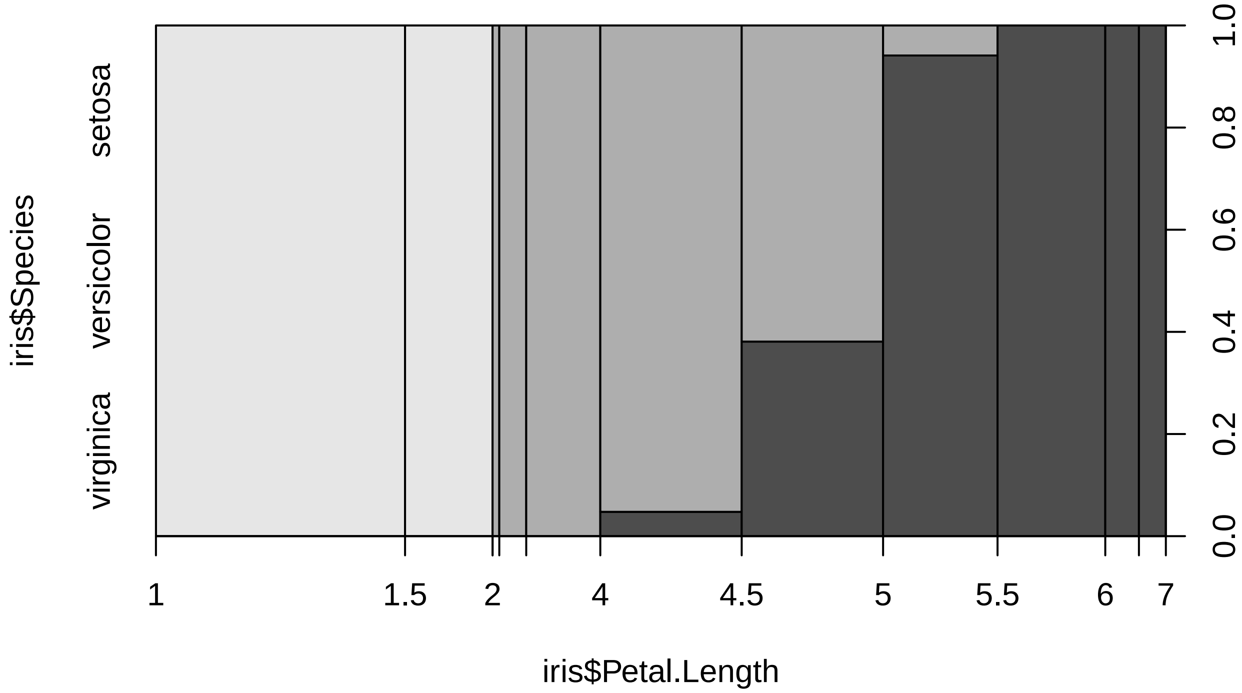
```
plot(iris$Species~iris$Sepal.Length)
```



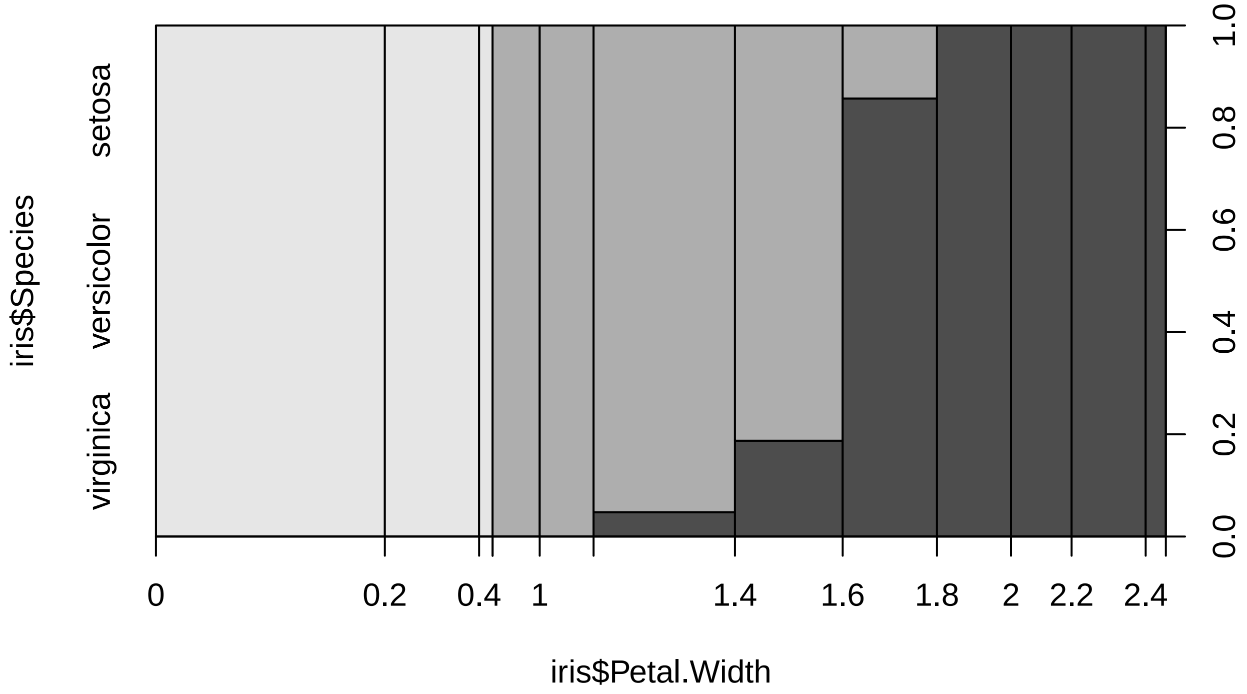
```
plot(iris$Species~iris$Sepal.Width)
```



```
plot(iris$Species~iris$Petal.Length)
```



```
plot(iris$Species~iris$Petal.Width)
```

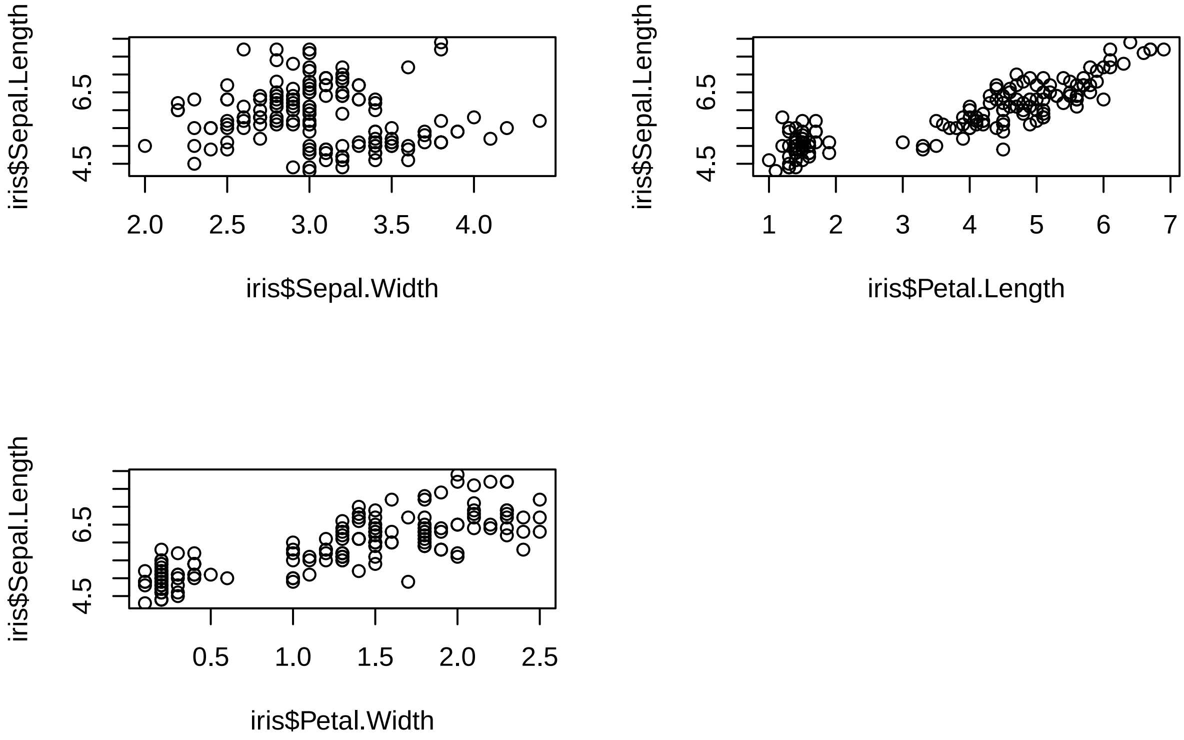


From the plots above, all variables seem to have some level of correlation with the categorical variable of interest. With the increase of Sepal.Length, Petal.Length and Petal.Width, the probability of being virginica seems to be increasing. With the increase of Sepal.Width, the probability of being setosa seems to be increasing.

Before fitting the model, I try to identify is there any correlation between these four numeric variables.

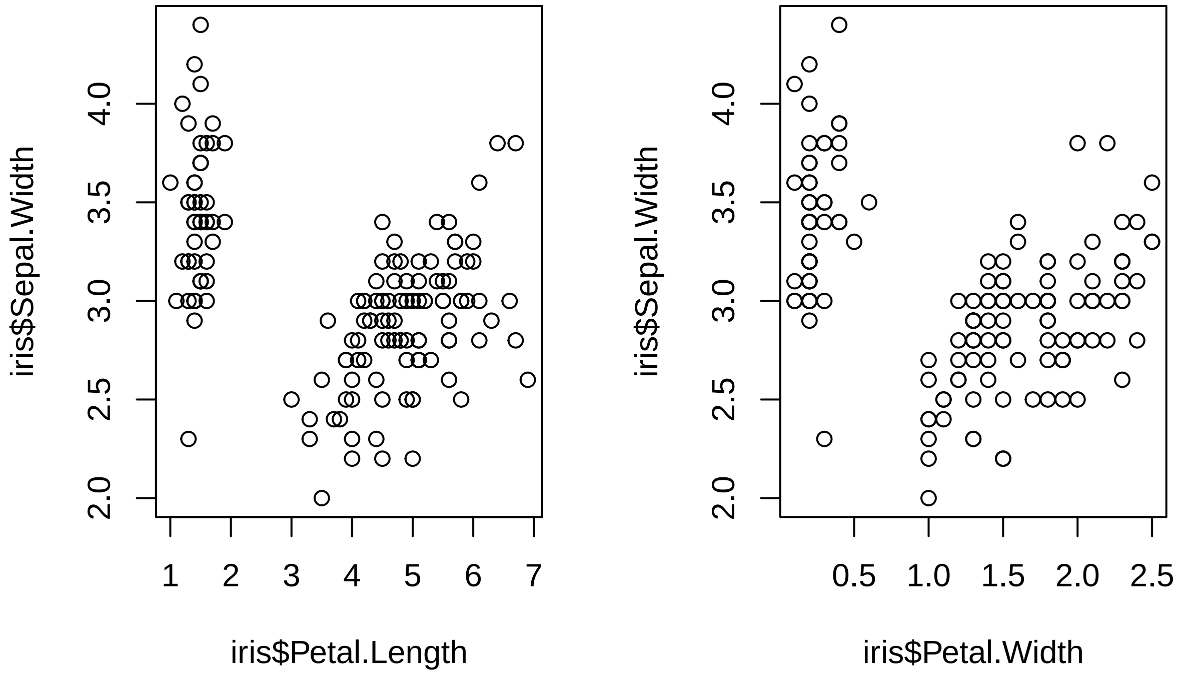
```
par(mfrow=c(2,2))
plot(iris$Sepal.Length~iris$Sepal.Width)
plot(iris$Sepal.Length~iris$Petal.Length)
plot(iris$Sepal.Length~iris$Petal.Width)
cor(iris$Sepal.Length*3, cbind(iris$Sepal.Width,
                                iris$Petal.Length,iris$Petal.Width))
```

```
##             [,1]      [,2]      [,3]
## [1,] -0.1175698 0.8717538 0.8179411
```



From the above result, the variable Sepal.Length is highly correlated with variables Petal.Length and Petal.Width. Therefore, these three variables can not be included in the model at the same time.

```
par(mfrow=c(1,2))
plot(iris$Sepal.Width~iris$Petal.Length)
plot(iris$Sepal.Width~iris$Petal.Width)
```

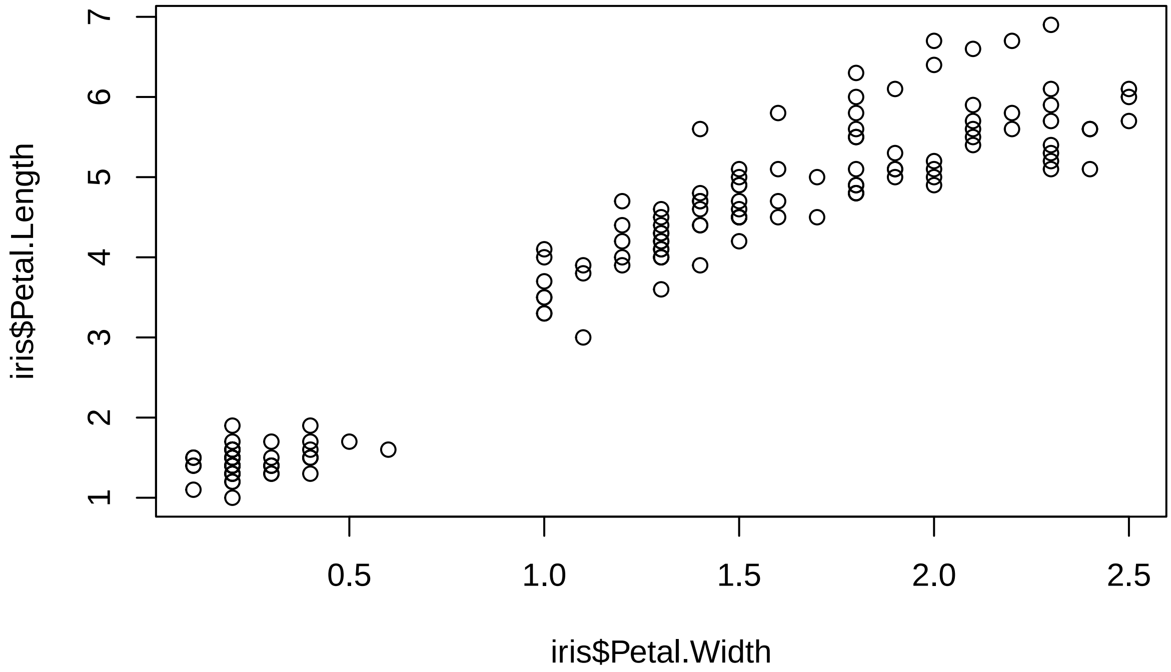


```
cor(iris$Sepal.Width*2, cbind(iris$Petal.Length,iris$Petal.Width))
```

```
##             [,1]       [,2]
## [1,] -0.4284401 -0.3661259
```

From the above result, the variable Sepal.Width is not significantly correlated with variables Petal.Length and Petal.Width.

```
plot(iris$Petal.Length~iris$Petal.Width)
```



```
cor(iris$Petal.Length,iris$Petal.Width)
```

```
## [1] 0.9628654
```

From the above result, the variables Petal.Width and Petal Length are extreme highly correlated. Therefore, these two variables can definitely not be included in the model at the same time.

Based on the previous findings, I would choose variables Sepal.Width and Petal.Length as my independent variables. Because these two variables are independent with each other, and the Petal.Length seems to be most highly correlated with the dependent variables. Also, I would set setosa species as my reference case for the dependent variable.

```
iris$Species2<-relevel(iris$Species, ref = "setosa")
m_spe<-multinom(iris$Species2 ~ iris$Sepal.Width + iris$Petal.Length)
```

```
## # weights:  12 (6 variable)
## initial  value 164.791843
## iter  10 value 27.458393
## iter  20 value 16.323670
## iter  30 value 15.804032
## iter  40 value 15.773169
## iter  50 value 15.765030
## iter  60 value 15.763657
## iter  70 value 15.761105
## iter  80 value 15.759926
## iter  90 value 15.758010
## final  value 15.757878
## converged
```

```
summary(m_spe)
```

```
## Call:
## multinom(formula = iris$Species2 ~ iris$Sepal.Width + iris$Petal.Length)
##
## Coefficients:
##            (Intercept) iris$Sepal.Width iris$Petal.Length
## versicolor    11.15223        -13.83385          10.59658
## virginica    -26.26734        -16.33225          19.76280
##
## Std. Errors:
##            (Intercept) iris$Sepal.Width iris$Petal.Length
## versicolor    15.00706         23.46767          35.11604
## virginica     15.73829         23.48803          35.20750
##
## Residual Deviance: 31.51576
## AIC: 43.51576
```

Calculate statistical test results.

```
z <- summary(m_spe)$coefficients/summary(m_spe)$standard.errors
z
```

```
##            (Intercept) iris$Sepal.Width iris$Petal.Length
## versicolor   0.7431325       -0.5894853         0.3017590
## virginica   -1.6690091       -0.6953434         0.5613237
```

```
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p
```

```
##            (Intercept) iris$Sepal.Width iris$Petal.Length
## versicolor  0.45740147        0.5555358         0.7628358
## virginica   0.09511558        0.4868401         0.5745769
```

Based on the diagnostic results, none of the coefficient is significant. The model seems to be invalid for some reason.

Try to change the specification of the model. Use Petal.Width instead of Petal.Length in the model.

```
m_spe2<-multinom(iris$Species2 ~ iris$Sepal.Width + iris$Petal.Width)
```

```
## # weights:  12 (6 variable)
## initial  value 164.791843
## iter  10 value 72.114113
## iter  20 value 24.910312
## iter  30 value 13.700981
## iter  40 value 13.699865
## iter  50 value 13.699755
## iter  60 value 13.699723
## iter  70 value 13.699697
## iter  80 value 13.699679
## iter  90 value 13.699670
## iter 100 value 13.699657
## final  value 13.699657
## stopped after 100 iterations
```

```
summary(m_spe2)
```

```
## Call:
## multinom(formula = iris$Species2 ~ iris$Sepal.Width + iris$Petal.Width)
##
## Coefficients:
##            (Intercept) iris$Sepal.Width iris$Petal.Width
## versicolor    6.302503        -16.42649         47.23637
## virginica    -8.080038        -20.33514         62.94101
##
## Std. Errors:
##            (Intercept) iris$Sepal.Width iris$Petal.Width
## versicolor    10.02487         26.08532         9.928271
## virginica     10.06567         26.09531         9.886234
##
## Residual Deviance: 27.39931
## AIC: 39.39931
```

```
z <- summary(m_spe2)$coefficients/summary(m_spe2)$standard.errors
z
```

```
##            (Intercept) iris$Sepal.Width iris$Petal.Width
## versicolor   0.6286865       -0.6297216         4.757764
## virginica   -0.8027325       -0.7792644         6.366531
```

```
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p
```

```
##            (Intercept) iris$Sepal.Width iris$Petal.Width
## versicolor   0.5295543        0.5288768     1.957492e-06
## virginica    0.4221294        0.4358240     1.933513e-10
```

Based on the results, the model improved. The p value of Petal.Width is extreme small which indicate the this variable is very significant in predicting the dependent variable.

I further tried to use Sepal.Length with Sepal.Width in the model.

```
m_spe3<-multinom(iris$Species2 ~ iris$Sepal.Width + iris$Sepal.Length)
```

```
## # weights:  12 (6 variable)
## initial  value 164.791843
## iter  10 value 62.715967
## iter  20 value 59.808291
## iter  30 value 55.445984
## iter  40 value 55.375704
## iter  50 value 55.346472
## iter  60 value 55.301707
## iter  70 value 55.253532
## iter  80 value 55.243230
## iter  90 value 55.230241
## iter 100 value 55.212479
## final  value 55.212479
## stopped after 100 iterations
```

```
summary(m_spe3)
```

```
## Call:
## multinom(formula = iris$Species2 ~ iris$Sepal.Width + iris$Sepal.Length)
##
## Coefficients:
##             (Intercept) iris$Sepal.Width iris$Sepal.Length
## versicolor   -92.09925        -40.58755          40.40326
## virginica   -105.10096        -40.18800          42.30095
##
## Std. Errors:
##             (Intercept) iris$Sepal.Width iris$Sepal.Length
## versicolor    26.27830         27.77772          9.142716
## virginica     26.37025         27.78874          9.131119
##
## Residual Deviance: 110.425
## AIC: 122.425
```

```
z <- summary(m_spe3)$coefficients/summary(m_spe3)$standard.errors
z
```

```
##             (Intercept) iris$Sepal.Width iris$Sepal.Length
## versicolor   -3.504764        -1.461155          4.419175
## virginica    -3.985588        -1.446197          4.632614
```

```
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p
```

```
##              (Intercept) iris$Sepal.Width iris$Sepal.Length
## versicolor 4.570127e-04        0.1439729      9.907831e-06
## virginica  6.731315e-05        0.1481220      3.610778e-06
```

The result is rather similar with the previous model, the variable Sepal.Width is not significant. Therefore, I think that the variable Sepal.Width seems to be not important for predicting species of the flowers. I would exclude variable Sepal.Width from the model. Also, since the remaining three variables are highly correlated with each other, I would tried to use only one independent variable in the modeling.

Use Sepal.Length as the independent variable.

```
m_spesl<-multinom(iris$Species2 ~ iris$Sepal.Length)
```

```
## # weights:  9 (4 variable)
## initial  value 164.791843
## iter  10 value 91.337114
## iter  20 value 91.035008
## final  value 91.033971
## converged
```

```
summary(m_spesl)
```

```
## Call:
## multinom(formula = iris$Species2 ~ iris$Sepal.Length)
##
## Coefficients:
##             (Intercept) iris$Sepal.Length
## versicolor   -26.08339          4.816072
## virginica    -38.76786          6.847957
##
## Std. Errors:
##             (Intercept) iris$Sepal.Length
## versicolor    4.889635         0.9069211
## virginica     5.691596         1.0223867
##
## Residual Deviance: 182.0679
## AIC: 190.0679
```

```
z <- summary(m_spesl)$coefficients/summary(m_spesl)$standard.errors
z
```

```
##             (Intercept) iris$Sepal.Length
## versicolor   -5.334424          5.310353
## virginica    -6.811422          6.698011
```

```
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p
```

```
##              (Intercept) iris$Sepal.Length
## versicolor 9.584830e-08      1.094128e-07
## virginica  9.663825e-12      2.112754e-11
```

Use Petal.Length as the independent variable.

```
m_spepl<-multinom(iris$Species2 ~ iris$Petal.Length)
```

```
## # weights:  9 (4 variable)
## initial  value 164.791843
## iter  10 value 18.070003
## iter  20 value 17.303746
## iter  30 value 16.853561
## iter  40 value 16.776006
## iter  50 value 16.763088
## iter  60 value 16.755329
## iter  70 value 16.750738
## iter  80 value 16.748059
## iter  90 value 16.745254
## iter 100 value 16.742943
## final  value 16.742943
## stopped after 100 iterations
```

```
summary(m_spepl)
```

```
## Call:
## multinom(formula = iris$Species2 ~ iris$Petal.Length)
##
## Coefficients:
##            (Intercept) iris$Petal.Length
## versicolor   -22.99873          9.194485
## virginica    -66.34860         18.110300
##
## Std. Errors:
##            (Intercept) iris$Petal.Length
## versicolor    23.18035          9.526659
## virginica     25.64033          9.789355
##
## Residual Deviance: 33.48589
## AIC: 41.48589
```

```
z <- summary(m_spepl)$coefficients/summary(m_spepl)$standard.errors
z
```

```
##            (Intercept) iris$Petal.Length
## versicolor  -0.9921648         0.9651321
## virginica   -2.5876657         1.8499993
```

```
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p
```

```
##            (Intercept) iris$Petal.Length
## versicolor 0.321117165        0.33447863
## virginica  0.009662871        0.06431365
```

Use Petal.Width as the independent variable.

```
m_spepw<-multinom(iris$Species2 ~ iris$Petal.Width)
```

```
## # weights:  9 (4 variable)
## initial  value 164.791843
## iter  10 value 17.736099
## iter  20 value 16.947974
## iter  30 value 16.808175
## iter  40 value 16.732572
## iter  50 value 16.723678
## iter  60 value 16.722619
## iter  70 value 16.722250
## iter  80 value 16.720992
## iter  90 value 16.720845
## iter 100 value 16.720654
## final  value 16.720654
## stopped after 100 iterations
```

```
summary(m_spepw)
```

```
## Call:
## multinom(formula = iris$Species2 ~ iris$Petal.Width)
##
## Coefficients:
##            (Intercept) iris$Petal.Width
## versicolor   -24.08868         31.44849
## virginica    -45.20657         44.39249
##
## Std. Errors:
##            (Intercept) iris$Petal.Width
## versicolor    38.78422         48.37552
## virginica     39.05553         48.45901
##
## Residual Deviance: 33.44131
## AIC: 41.44131
```

```
z <- summary(m_spepw)$coefficients/summary(m_spepw)$standard.errors
z
```

```
##            (Intercept) iris$Petal.Width
## versicolor  -0.6210949        0.6500910
## virginica   -1.1574947        0.9160833
```

```
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p
```

```
##            (Intercept) iris$Petal.Width
## versicolor   0.5345372        0.5156334
## virginica    0.2470703        0.3596232
```

Despite previous results indicate that the variable Sepal.Width may not be significant when together with other independent variables in the model, I still try out it as the only independent variables in the model.

```
m_spesw<-multinom(iris$Species2 ~ iris$Sepal.Width)
```

```
## # weights:  9 (4 variable)
## initial  value 164.791843
## iter  10 value 126.269225
## final  value 126.268479
## converged
```

```
summary(m_spesw)
```

```
## Call:
## multinom(formula = iris$Species2 ~ iris$Sepal.Width)
##
## Coefficients:
##             (Intercept) iris$Sepal.Width
## versicolor    18.85837        -6.118941
## virginica     12.99728        -4.079084
##
## Std. Errors:
##             (Intercept) iris$Sepal.Width
## versicolor    3.064283        0.9912226
## virginica     2.688309        0.8435572
##
## Residual Deviance: 252.537
## AIC: 260.537
```

```
z <- summary(m_spesw)$coefficients/summary(m_spesw)$standard.errors
z
```

```
##             (Intercept) iris$Sepal.Width
## versicolor    6.154254        -6.173125
## virginica     4.834741        -4.835576
```

```
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p
```

```
##              (Intercept) iris$Sepal.Width
## versicolor 7.543177e-10     6.695324e-10
## virginica  1.333189e-06     1.327609e-06
```

Based on the model results above, surprisingly both variables Sepal.Width and Sepal.Length are significant when each of them are the only independent variable in the model. It seems that although the variable Sepal.Width is not statistically correlated with the other three variables, the variable still contain significant duplicated information with other variables which cause it to be insignificant when together with other variables in one model. Moreover, the model with Sepal.Length has lower AIC and Residual deviance than the model with Sepal.Width, which indicate better model fitness.

Then, I calculate the deviance residuals and pearson residuals of the two models who has significant independent variable. Also, calculate the Deviance and Pearson X2 statistics of model fit from these.

```
m_spesl.dev_residuals = residuals(m_spesl, type="deviance")
m_spesl.pearson_residuals = residuals(m_spesl, type="pearson")
m_spesw.dev_residuals = residuals(m_spesw, type="deviance")
m_spesw.pearson_residuals = residuals(m_spesw, type="pearson")
sum(m_spesl.dev_residuals^2)
```

```
## [1] 56.65386
```

```
sum(m_spesl.pearson_residuals^2)
```

```
## [1] 56.65386
```

```
sum(m_spesw.dev_residuals^2)
```

```
## [1] 76.11534
```

```
sum(m_spesw.pearson_residuals^2)
```

```
## [1] 76.11534
```

The m_spesl model which is the model with Sepal.Length as the only independent variable has the lower general residuals for both deviance and pearson residuals. This indicate that the model with only Sepal.Length is the better model.

Therefore, the model with only Sepal.Length is the better model. I would stick to it (m_spesl) as my final model.

```
summary(m_spesl)
```

```
## Call:
## multinom(formula = iris$Species2 ~ iris$Sepal.Length)
##
## Coefficients:
##            (Intercept) iris$Sepal.Length
## versicolor   -26.08339          4.816072
## virginica    -38.76786          6.847957
##
## Std. Errors:
##            (Intercept) iris$Sepal.Length
## versicolor    4.889635         0.9069211
## virginica     5.691596         1.0223867
##
## Residual Deviance: 182.0679
## AIC: 190.0679
```

Functions of my final model:

$$ln(\frac{Pr(species = versicolor)}{Pr(species = setosa)}) = -26.08339 + 4.816072 * Sepal.Length$$

$$ln(\frac{Pr(species = virginica)}{Pr(species = setosa)}) = -38.76786 + 6.847957 * Sepal.Length$$

Interpretations:

```
exp(coef(m_spesl))
```

```
##             (Intercept) iris$Sepal.Length
## versicolor 4.700338e-12          123.4791
## virginica  1.456567e-17          941.9549
```

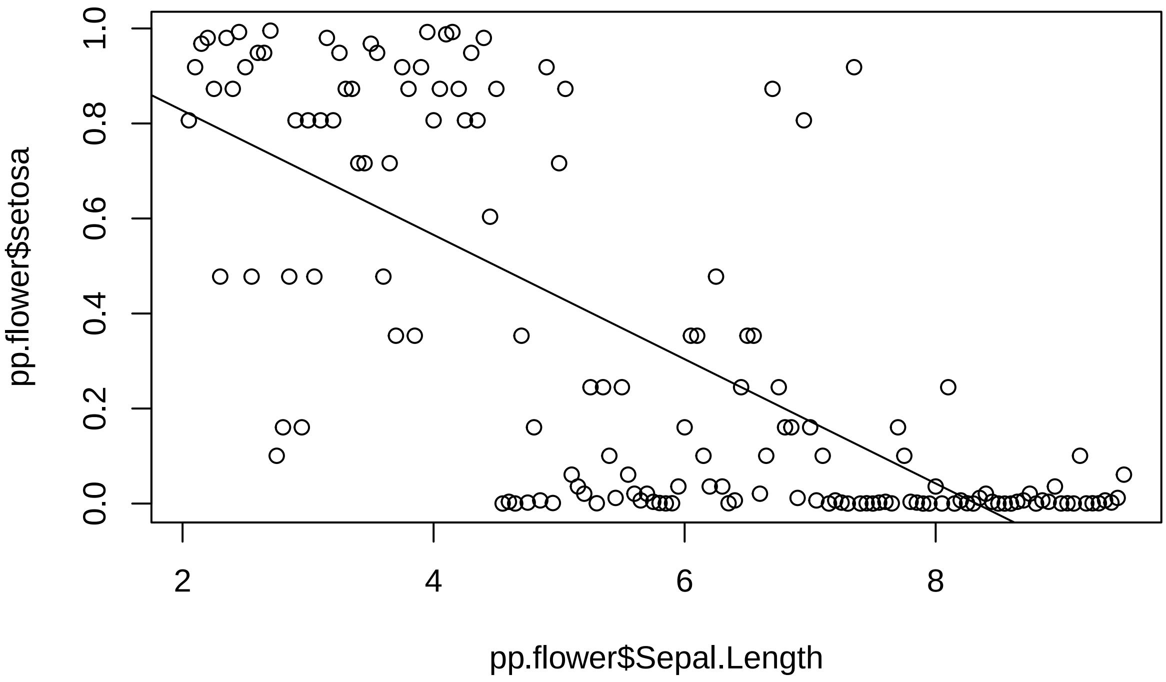
The relative risk ratio for a unit increase in the Sepal.Length is 123.4791 for versicolor species vs. setosa species. For every unit increase in the Sepal.Length, we expected to see 123.4791 times increase in the probability of being versicolor species than setosa species.

The relative risk ratio for a unit increase in the Sepal.Length is 941.9549 for virginica species vs. setosa species. For every unit increase in the Sepal.Length, we expected to see 941.9549 times increase in the probability of being virginica species than setosa species.

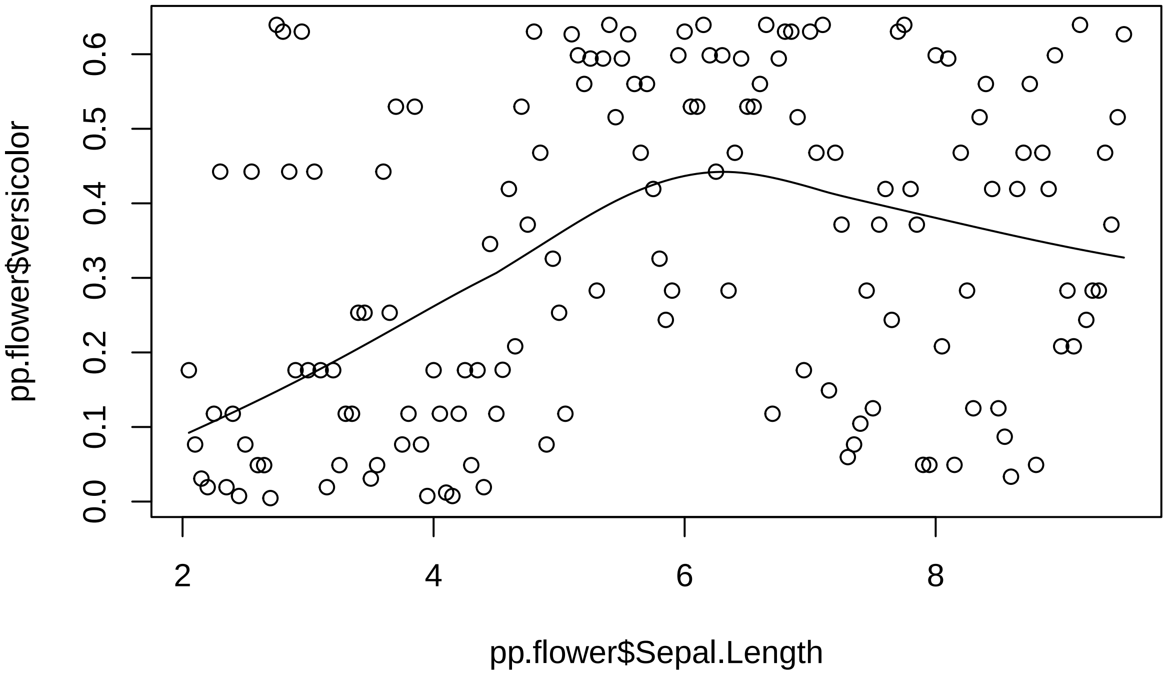
The “odds”, which technically not real odds, here in this model is the probability of being other species against the reference case which is being setosa species. It makes sense here, because the “odds” here gives us the ratio of chance the flowers in one species against the reference species. By looking at the “odds”, we can get the information whether this flower is more likely to be in the reference species or be in other particular species.

Next, use the “predicted probabilities” for each outcome to further interpret and visualize the model. Considering the range of Sepal.Length from the given data is 4.3 to 7.9. I artificially create a new dataset cover this range with 150 observations which is the same as in the model.

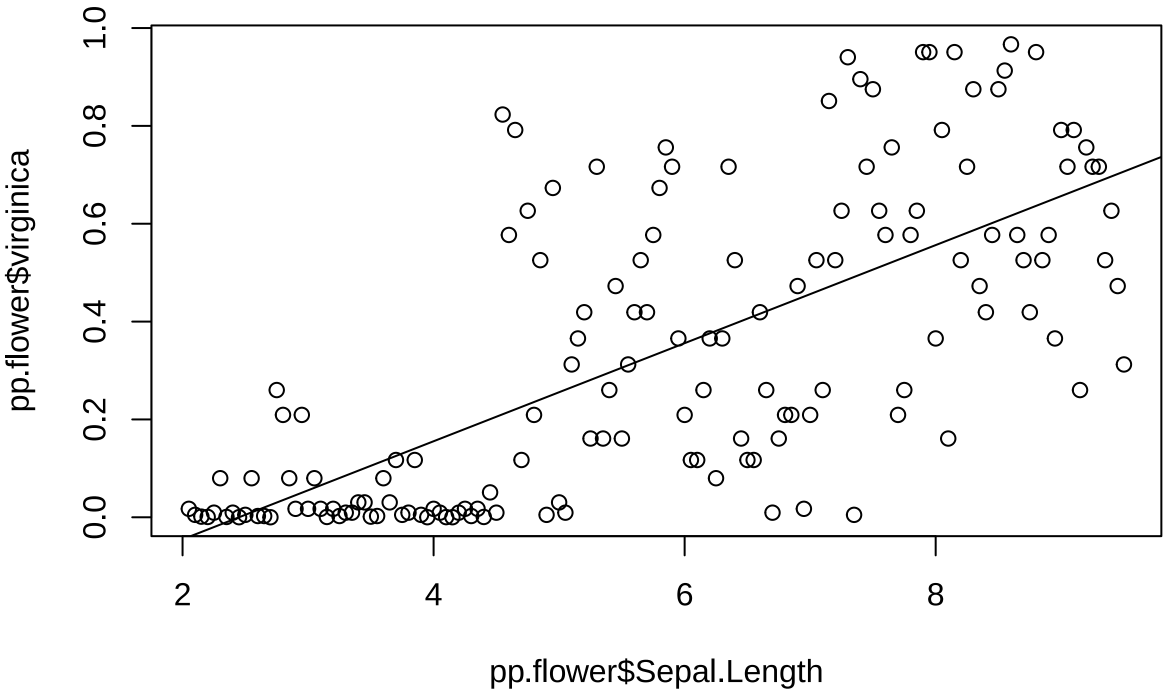
```r
dflower <- data.frame(Sepal.Length = seq(from = 2.05, to = 9.5, by = 0.05))
pp.flower <- cbind(dflower, predict(m_spesl, newdata = dflower, type = "probs", se = TRUE))
plot(pp.flower$setosa~pp.flower$Sepal.Length)
abline(lsfit(pp.flower$Sepal.Length, pp.flower$setosa))
```



```r
plot(pp.flower$versicolor~pp.flower$Sepal.Length)
lines(lowess(pp.flower$Sepal.Length,pp.flower$versicolor))
```



```
plot(pp.flower$virginica~pp.flower$Sepal.Length)
abline(lsfit(pp.flower$Sepal.Length, pp.flower$virginica))
```



According to the above plot, the setosa species flowers tends to have small sepal length, while the virginica species flowers tends to have big sepal length. The versicolor species flowers tends to have sepal length between other two species. With the increase of sepal length from 0, the probability of being setosa decreases, while the probability of being versicolor and virginica increases. With the further increase in sepal length, the probability of being sepal length starts to decrease as well. The probability of other two species keeps the same trend as before with the further increase in sepal length.