Logistic Regression Analysis of Breast Cancer Coimbra Data Set

Andrew Tran, Edward Wang

2023-04-10

Load and transform the data

```
# Read data from csv file
bccdat <- read.csv("breast-cancer-coimbra-data-set.csv")

# Transforming the Response Variable to Conform with the LR: Healthy = 0, Patients = 1
bccdat$Classification <- bccdat$Classification - 1

# Splitting the data into training & testing set: 80% to train, 20% to test
set.seed(1046)
train_data <- slice_sample(bccdat, prop = 0.8)
test_data <- anti_join(bccdat, train_data)
```

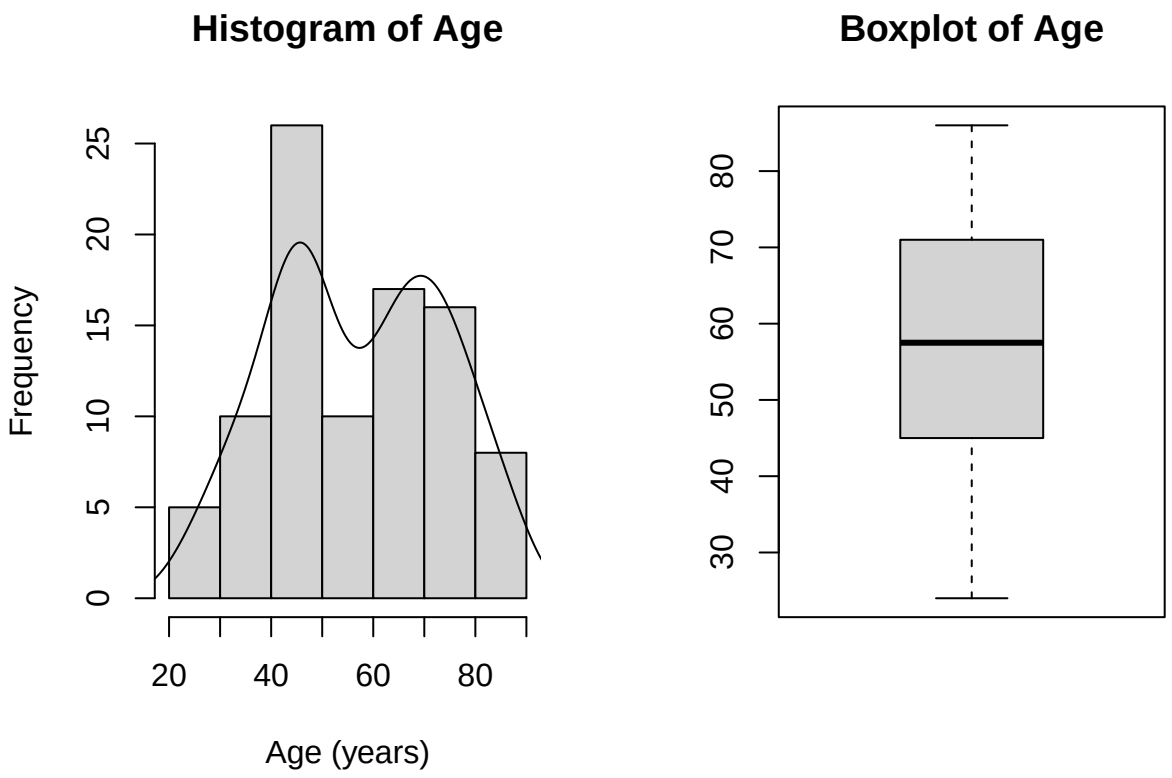
Summary statistics of the training data

```
attach(train_data)  # using summary(variable_name), sd(variable_name) to obtain the needed Statistics
```

Graphing of explanatory variables:

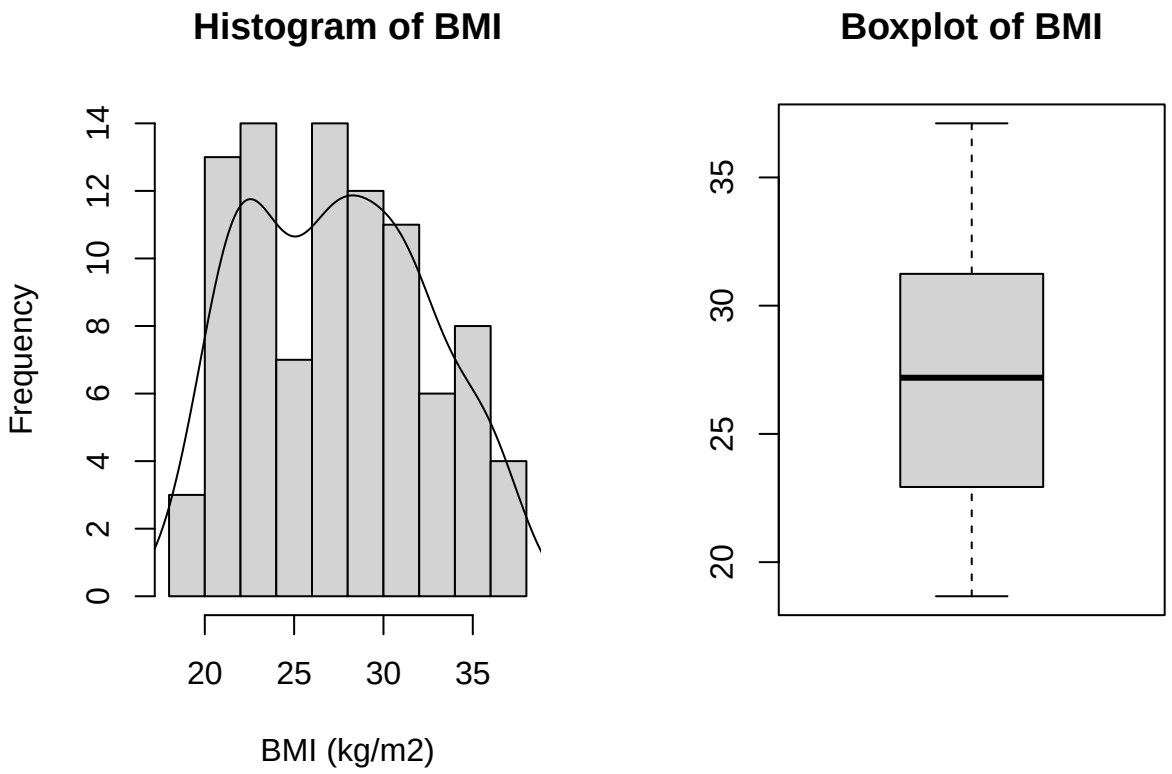
Age

```
hist <- hist(Age, main="histogram of Age", plot=FALSE)
multiplier <- hist$counts / hist$density
density <- density(Age)
density$y <- density$y * multiplier[1]
par(mfrow=c(1,2))
plot(hist, xlab = "Age (years)")
lines(density)
boxplot(Age, main="Boxplot of Age")
```



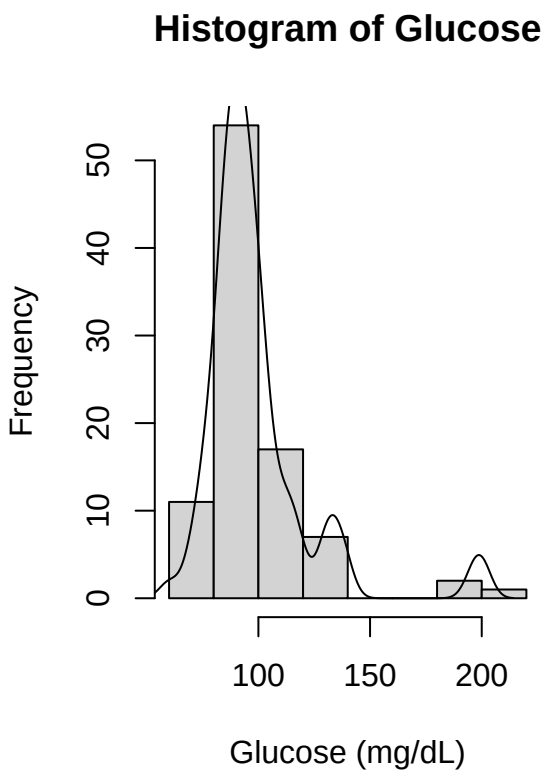
BMI

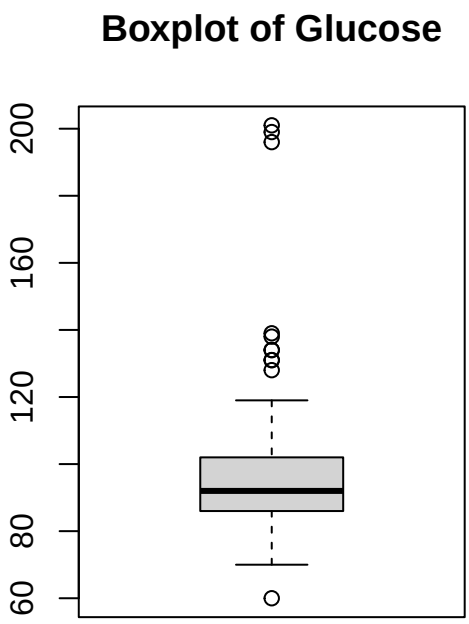
```
hist <- hist(BMI, main="histogram of BMI", plot=FALSE)
multiplier <- hist$counts / hist$density
density <- density(BMI)
density$y <- density$y * multiplier[1]
par(mfrow=c(1,2))
plot(hist, xlab = "BMI (kg/m2)")
lines(density)
boxplot(BMI, main="Boxplot of BMI")
```



Glucose

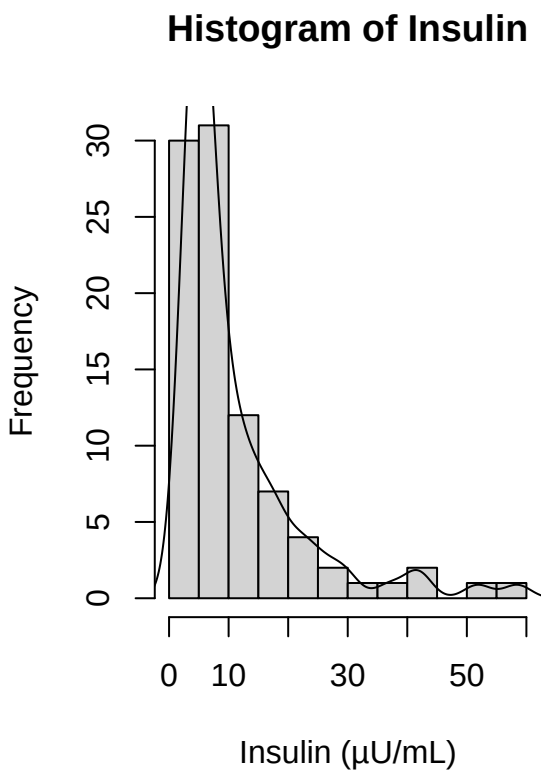
```
hist <- hist(Glucose , main="histogram of Glucose", plot=FALSE)
multiplier <- hist$counts / hist$density
density <- density(Glucose )
density$y <- density$y * multiplier[1]
par(mfrow=c(1,2))
plot(hist, xlab = "Glucose (mg/dL)")
lines(density)
boxplot(Glucose, main="Boxplot of Glucose")
```

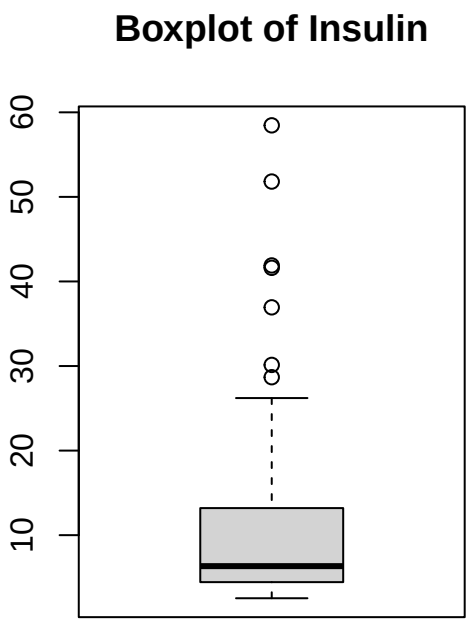




Insulin

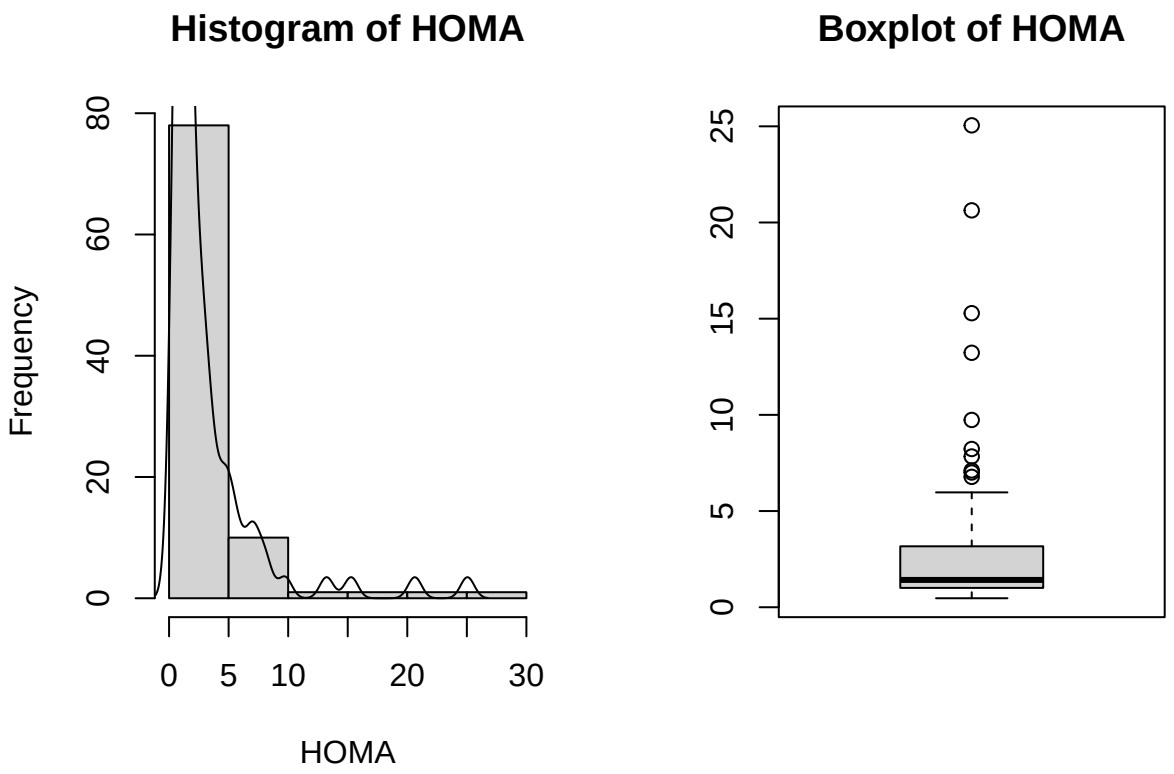
```
hist <- hist(Insulin , main="histogram of Insulin", plot=FALSE)
multiplier <- hist$counts / hist$density
density <- density(Insulin )
density$y <- density$y * multiplier[1]
par(mfrow=c(1,2))
plot(hist, xlab = "Insulin (µU/mL)")
lines(density)
boxplot(Insulin, main="Boxplot of Insulin")
```





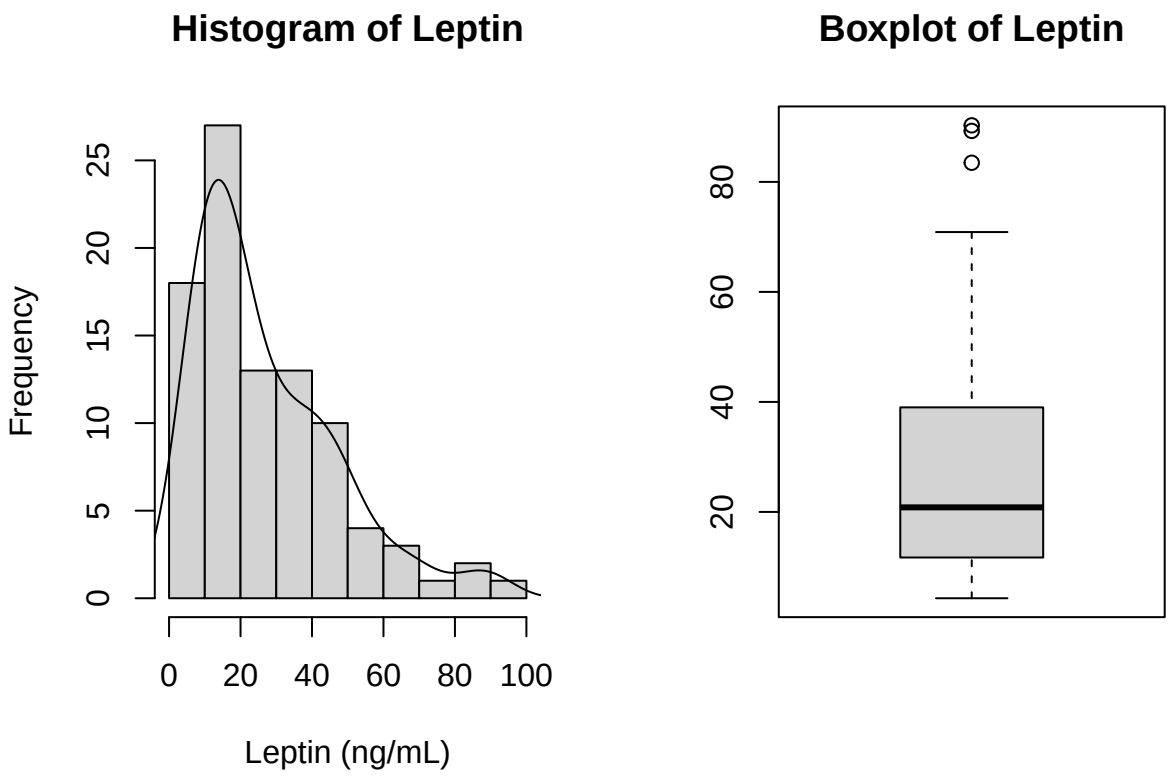
HOMA

```
hist <- hist(HOMA , main="histogram of HOMA", plot=FALSE)
multiplier <- hist$counts / hist$density
density <- density(HOMA )
density$y <- density$y * multiplier[1]
par(mfrow=c(1,2))
plot(hist, xlab = "HOMA")
lines(density)
boxplot(HOMA, main="Boxplot of HOMA")
```



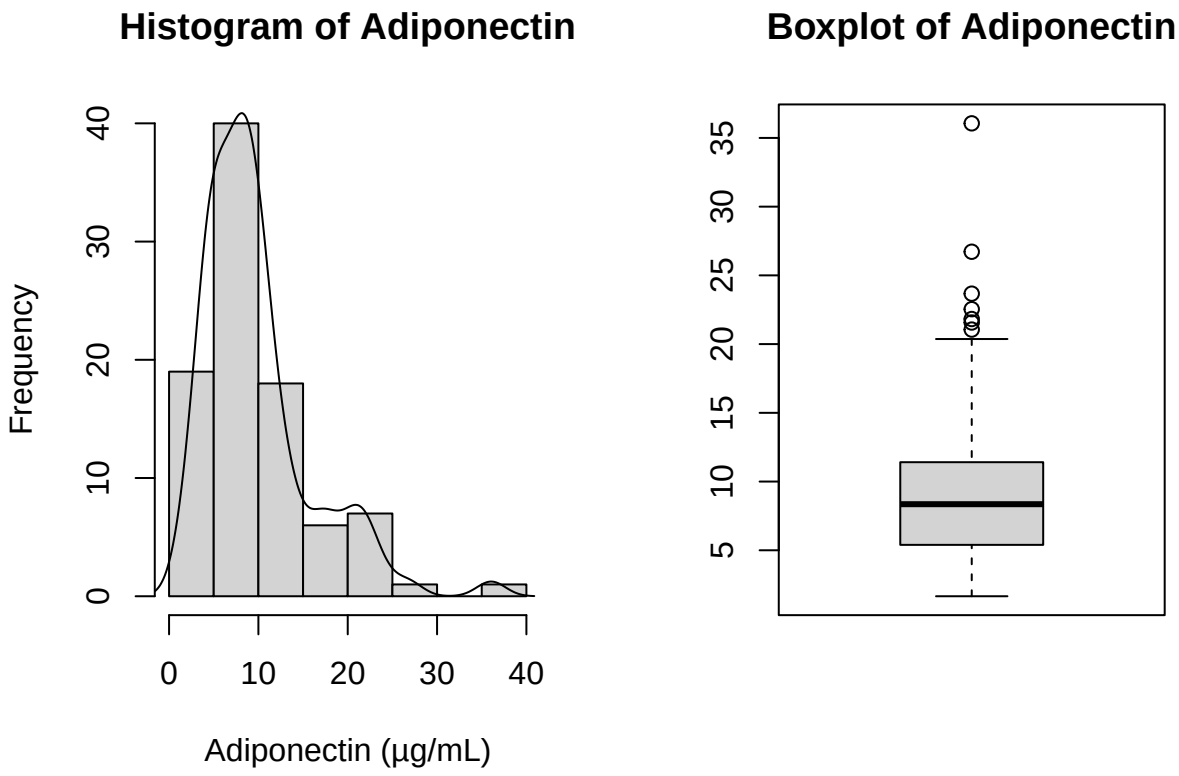
Leptin

```
hist <- hist(Leptin  , main="histogram of Leptin", plot=FALSE)
multiplier <- hist$counts / hist$density
density <- density(Leptin )
density$y <- density$y * multiplier[1]
par(mfrow=c(1,2))
plot(hist, xlab = "Leptin (ng/mL)")
lines(density)
boxplot(Leptin, main="Boxplot of Leptin")
```



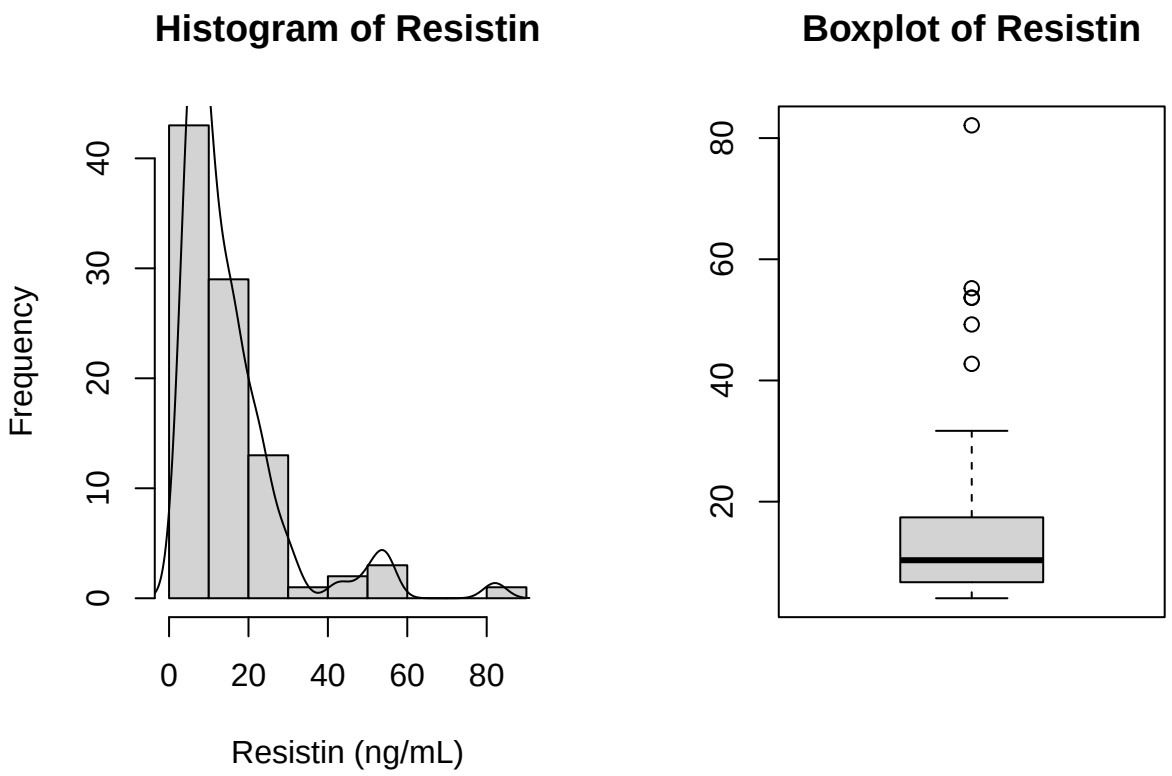
Adiponectin

```
hist <- hist(Adiponectin, main="histogram of Adiponectin", plot=FALSE)
multiplier <- hist$counts / hist$density
density <- density(Adiponectin)
density$y <- density$y * multiplier[1]
par(mfrow=c(1,2))
plot(hist, xlab = "Adiponectin (µg/mL)")
lines(density)
boxplot(Adiponectin, main="Boxplot of Adiponectin")
```



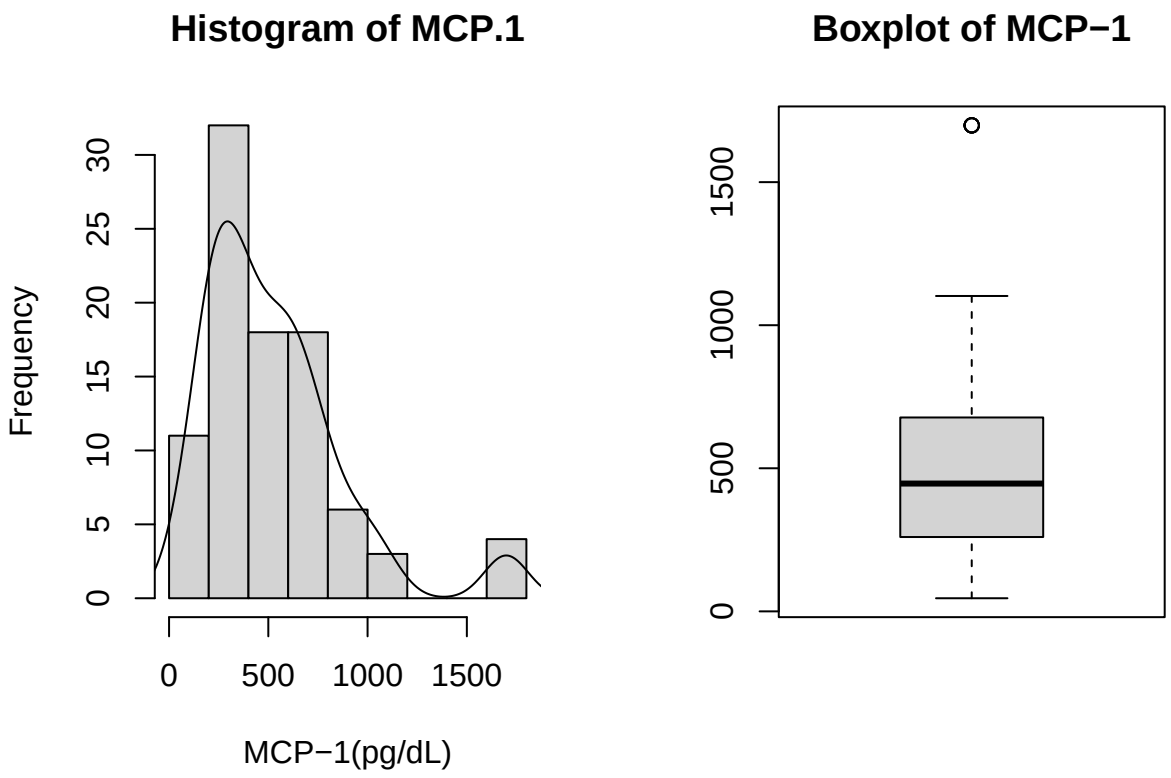
Resistin

```
hist <- hist(Resistin, main="histogram of Resistin", plot=FALSE)
multiplier <- hist$counts / hist$density
density <- density(Resistin)
density$y <- density$y * multiplier[1]
par(mfrow=c(1,2))
plot(hist, xlab = "Resistin (ng/mL)")
lines(density)
boxplot(Resistin, main="Boxplot of Resistin")
```



MCP-1

```
hist <- hist(MCP.1, main="histogram of MCP-1", plot=FALSE)
multiplier <- hist$counts / hist$density
density <- density(MCP.1)
density$y <- density$y * multiplier[1]
par(mfrow=c(1,2))
plot(hist, xlab = "MCP-1(pg/dL)")
lines(density)
boxplot(MCP.1, main="Boxplot of MCP-1")
```



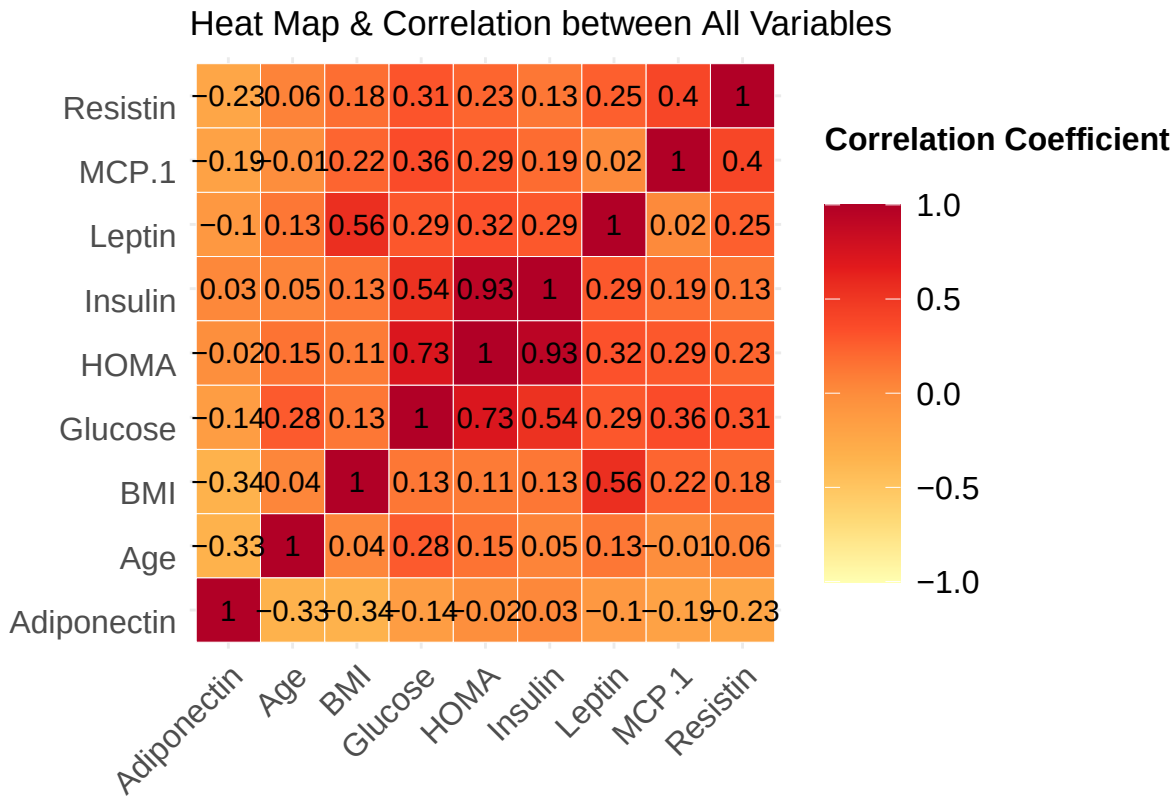
Correlation visualization

```
# Deselect the response variable
train_expl_data <- train_data[,!names(train_data) %in% c("Classification")]
```

Heat map

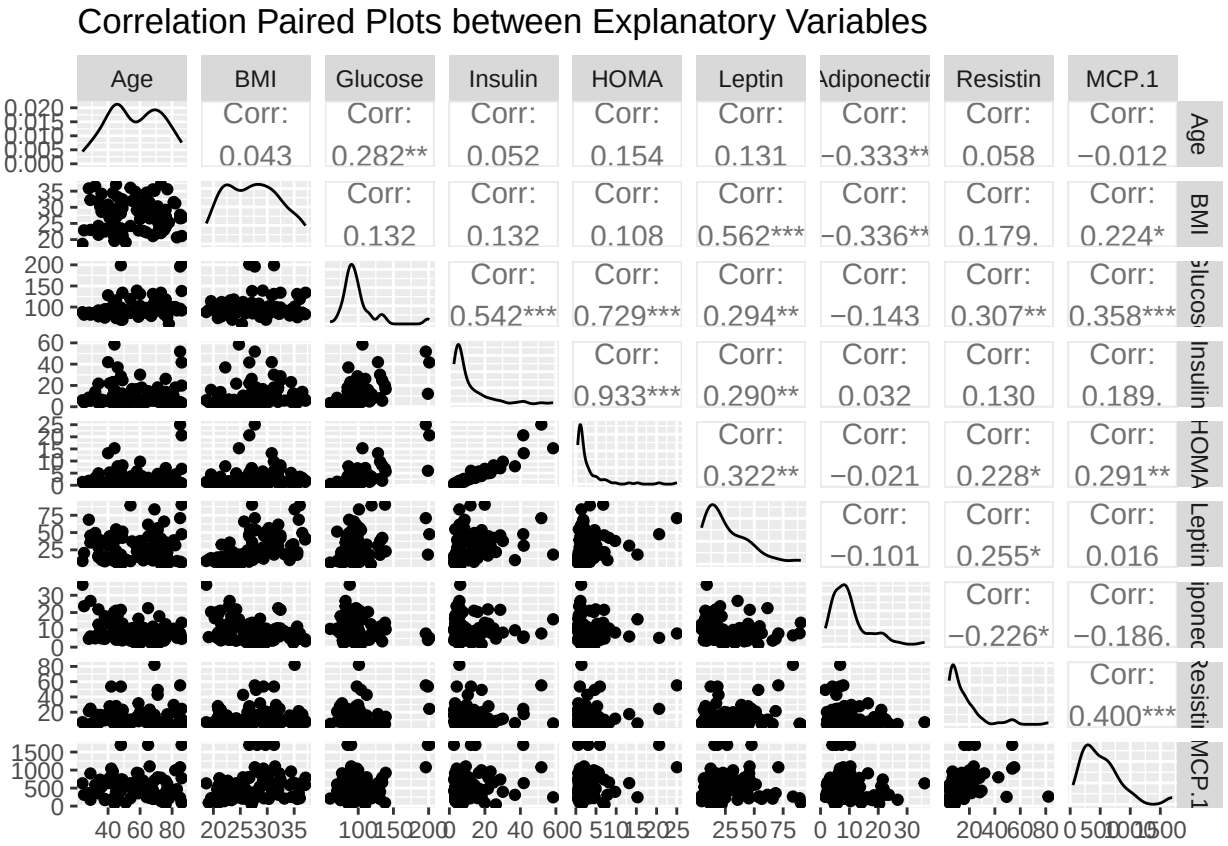
```
# obtain the correlation matrix
cor_mat <- train_expl_data %>%
           cor() %>%
           as.data.frame() %>%
           rownames_to_column("var1") %>%
           pivot_longer(-var1, names_to = "var2", values_to = "corr")
# create the heatmap between variables
plot_corr_matrix <- cor_mat %>%
  ggplot(aes(x = var1, y = var2)) +
  geom_tile(aes(fill = corr), color = "white") +
  scale_fill_distiller("Correlation Coefficient \n",
                       palette =  "YlOrRd",
                       direction = 1, limits = c(-1, 1)
  ) +
  labs(x = "", y = "") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(
      angle = 45, vjust = 1,
      size = 12, hjust = 1
    ),
```

```
    axis.text.y = element_text(
      vjust = 1,
      size = 12, hjust = 1
    ),
    legend.title = element_text(size = 12, face = "bold"),
    legend.text = element_text(size = 12),
    legend.key.size = unit(1, "cm")
  ) +
  coord_fixed() +
  geom_text(aes(var1, var2, label = round(corr, 2)), color = "black", size = 4) +
  ggtitle("Heat Map & Correlation between All Variables")

plot_corr_matrix
```



Correlation paired plots

```
pair_plot <- ggpairs(train_expl_data,
                     title = "Correlation Paired Plots between Explanatory Variables",
                     progress=FALSE)

pair_plot
```



Model selection

Worst-case model

```
# calculate the proportion of the Classification = 0 & that of Classification == 1
p0 <- length(Classification[Classification==0])/length(Classification)
p1 <- length(Classification[Classification==1])/length(Classification)

# calculate AIC for the worst case model
# this is the log-likelihood of the worst case model
log1 <- log(p1^Classification*p0^(1-Classification))
AIC1 <- -2 * (sum(log1) - 1)
AIC1
```

```
## [1] 127.9694
```

Baseline model (full model without interactions)

```
full_model <- glm(Classification ~ ., data=train_data, family=binomial(link="logit"))
summary(full_model)
```

```
##
## Call:
## glm(formula = Classification ~ ., family = binomial(link = "logit"),
##     data = train_data)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9536  -0.7693   0.1628   0.7447   2.0507
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.481034   4.231577  -1.295   0.1952
## Age         -0.030342   0.018992  -1.598   0.1101
## BMI         -0.142139   0.080881  -1.757   0.0789 .
## Glucose      0.102955   0.044090   2.335   0.0195 *
## Insulin      0.067374   0.344479   0.196   0.8449
## HOMA        -0.051645   1.462748  -0.035   0.9718
## Leptin      -0.004615   0.019885  -0.232   0.8165
## Adiponectin -0.022011   0.049267  -0.447   0.6550
## Resistin     0.041889   0.027859   1.504   0.1327
## MCP.1        0.001881   0.001044   1.802   0.0716 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 125.969  on 91  degrees of freedom
## Residual deviance:  85.862  on 82  degrees of freedom
## AIC: 105.86
##
## Number of Fisher Scoring iterations: 8
```

```
vif(full_model)
```

```
##         Age         BMI     Glucose     Insulin        HOMA      Leptin
##    1.420299    2.429897    3.399515   75.030439   82.105619    2.135637
## Adiponectin    Resistin       MCP.1
##    1.345140    1.210822    1.320200
```

Variables selection

Insulin & HOMA and Glucose & HOMA are collinear

Insulin & HOMA pair and Glucose & HOMA pair have very high correlations. This information suggests collinearity in these two pairs. In reality, this is true. HOMA is a method used to quantify Insulin resistance and beta-cell function. In this sense, HOMA is a direct function of Glucose and Insulin (source: https://en.wikipedia.org/wiki/Homeostatic_model_assessment). Thus, we can disregard HOMA when fitting the data.

```
model_without_homa <- glm(Classification ~ Age + BMI + Glucose + Insulin + Leptin + Adiponectin + Resis
                          data=train_data,
                          family=binomial(link="logit"))
summary(model_without_homa)
```

```
##
## Call:
## glm(formula = Classification ~ Age + BMI + Glucose + Insulin +
##     Leptin + Adiponectin + Resistin + MCP.1, family = binomial(link = "logit"),
##     data = train_data)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.9495  -0.7679   0.1559   0.7453   2.0514
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.381070   3.135525  -1.716 0.086132 .
## Age         -0.030321   0.018979  -1.598 0.110141
## BMI         -0.141830   0.080375  -1.765 0.077630 .
## Glucose      0.101798   0.029326   3.471 0.000518 ***
## Insulin      0.055321   0.043373   1.275 0.202147
## Leptin      -0.004679   0.019828  -0.236 0.813452
## Adiponectin -0.022066   0.049228  -0.448 0.653979
## Resistin     0.041845   0.027799   1.505 0.132251
## MCP.1        0.001882   0.001044   1.803 0.071376 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 125.969  on 91  degrees of freedom
## Residual deviance:  85.863  on 83  degrees of freedom
## AIC: 103.86
##
## Number of Fisher Scoring iterations: 6
```

```
vif(model_without_homa)
```

```
##         Age         BMI     Glucose     Insulin      Leptin Adiponectin
##    1.418151    2.401566    1.498869    1.183840    2.119825    1.343355
##    Resistin       MCP.1
##    1.208821    1.319340
```

Our AIC is smaller after removing HOMA so we have ground to remove it when fitting the data.

Glucose & Insulin is collinear

Insulin is the hormone that metabolizes Glucose. This information along with the correlation between Glucose & Insulin suggest that there is a functional relationship between them.

```
model_without_glucose <- glm(Classification ~ Age + BMI + Insulin + Leptin + HOMA + Adiponectin + Resis
                             data=train_data,
                             family=binomial(link="logit"))
summary(model_without_glucose)
```

```
##
## Call:
## glm(formula = Classification ~ Age + BMI + Insulin + Leptin +
##     HOMA + Adiponectin + Resistin + MCP.1, family = binomial(link = "logit"),
##     data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.91907  -0.82067   0.00897   0.90004   1.96000
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)  3.1702310  2.2966624   1.380  0.16747
## Age         -0.0216038  0.0175763  -1.229  0.21902
## BMI         -0.1271277  0.0744652  -1.707  0.08778 .
## Insulin     -0.7550704  0.2804363  -2.692  0.00709 **
## Leptin      -0.0056979  0.0194867  -0.292  0.76998
## HOMA         3.6323801  1.2456323   2.916  0.00354 **
## Adiponectin -0.0292021  0.0477030  -0.612  0.54043
## Resistin     0.0368921  0.0251333   1.468  0.14214
## MCP.1        0.0016559  0.0009712   1.705  0.08818 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 125.969  on 91  degrees of freedom
## Residual deviance:  91.311  on 83  degrees of freedom
## AIC: 109.31
##
## Number of Fisher Scoring iterations: 8
```

```
vif(model_without_glucose)
```

```
##         Age         BMI     Insulin      Leptin        HOMA Adiponectin
##    1.301507    2.306979   43.795327    2.079909   44.680604    1.345703
##    Resistin       MCP.1
##    1.219626    1.244650
```

```
model_without_insulin <- glm(Classification ~ Age + BMI + Glucose + Leptin + HOMA + Adiponectin + Resis
                             data=train_data,
                             family=binomial(link="logit"))
summary(model_without_insulin)
```

```
##
## Call:
## glm(formula = Classification ~ Age + BMI + Glucose + Leptin +
##     HOMA + Adiponectin + Resistin + MCP.1, family = binomial(link = "logit"),
##     data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9335  -0.7610   0.1281   0.7475   2.0559
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.946348   3.201453  -1.545   0.1223
## Age         -0.030271   0.018966  -1.596   0.1105
## BMI         -0.140353   0.080213  -1.750   0.0802 .
## Glucose      0.096804   0.030298   3.195   0.0014 **
## Leptin      -0.004924   0.019957  -0.247   0.8051
## HOMA         0.233987   0.186167   1.257   0.2088
## Adiponectin -0.022272   0.049171  -0.453   0.6506
## Resistin     0.041589   0.027643   1.505   0.1324
## MCP.1        0.001887   0.001042   1.811   0.0702 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 125.969  on 91  degrees of freedom
## Residual deviance:  85.899  on 83  degrees of freedom
## AIC: 103.9
##
## Number of Fisher Scoring iterations: 6
```

```
vif(model_without_insulin)
```

```
##         Age         BMI     Glucose      Leptin        HOMA Adiponectin
##    1.415487    2.400320    1.577684    2.132220    1.271294    1.342211
##    Resistin       MCP.1
##    1.209355    1.316150
```

The AIC increases when we remove Glucose but decreases when we remove Insulin. Thus, we should consider removing Insulin when fitting the data.

Leptin & BMI might represent duplicated info

We can also see that the correlation between Leptin & BMI is decently high. Leptin is a hormone your body releases that helps it regulate fat storage (source: https://en.wikipedia.org/wiki/Leptin). A lack in Leptin leads to overweight/obesity in most cases. However, unlike the relationship between HOMA and Insulin/Glucose, BMI is not a direct function of Leptin due to other factors (e.g. a person with low Leptin could exercise a lot, etc.). Though, Leptin and BMI might be representing the same aspect here in our data, which is physical fitness (weight/body fat/obesity/etc.). Thus, we should examine models without Leptin or BMI.

```
model_without_bmi <- glm(Classification ~ Age + Glucose + Insulin + HOMA + Leptin + Adiponectin + Resis
                         data=train_data,
                         family=binomial(link="logit"))
summary(model_without_bmi)
```

```
##
## Call:
## glm(formula = Classification ~ Age + Glucose + Insulin + HOMA +
##     Leptin + Adiponectin + Resistin + MCP.1, family = binomial(link = "logit"),
##     data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0873  -0.8217   0.1216   0.8578   2.1882
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.6201488  4.1053459  -2.100   0.0358 *
## Age         -0.0270505  0.0182613  -1.481   0.1385
## Glucose      0.0995910  0.0445891   2.234   0.0255 *
## Insulin     -0.0002413  0.3257077  -0.001   0.9994
## HOMA         0.1966486  1.4056375   0.140   0.8887
## Leptin      -0.0285699  0.0154216  -1.853   0.0639 .
## Adiponectin  0.0095126  0.0451538   0.211   0.8331
## Resistin     0.0455036  0.0276192   1.648   0.0994 .
## MCP.1        0.0013486  0.0009460   1.426   0.1540
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 125.969  on 91  degrees of freedom
## Residual deviance:  89.115  on 83  degrees of freedom
## AIC: 107.12
##
## Number of Fisher Scoring iterations: 8
```

```
vif(model_without_bmi)
```

```
##         Age     Glucose     Insulin        HOMA      Leptin Adiponectin
##    1.353640    3.371261   73.231615   81.251624    1.330422    1.170625
##    Resistin       MCP.1
##    1.208768    1.172532
```

```
model_without_leptin <- glm(Classification ~ Age + Glucose + Insulin + HOMA + BMI + Adiponectin + Resis
                        data=train_data,
                        family=binomial(link="logit"))
summary(model_without_leptin)
```

```
##
## Call:
## glm(formula = Classification ~ Age + Glucose + Insulin + HOMA +
##     BMI + Adiponectin + Resistin + MCP.1, family = binomial(link = "logit"),
##     data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9204  -0.7929   0.1688   0.7633   2.0101
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.255903   4.071859  -1.291   0.1968
## Age         -0.030268   0.018995  -1.593   0.1110
## Glucose      0.102869   0.043772   2.350   0.0188 *
## Insulin      0.073269   0.342555   0.214   0.8306
## HOMA        -0.079884   1.450857  -0.055   0.9561
## BMI         -0.154228   0.062244  -2.478   0.0132 *
## Adiponectin -0.023769   0.048485  -0.490   0.6240
## Resistin     0.040303   0.027189   1.482   0.1383
## MCP.1        0.001945   0.001014   1.918   0.0552 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 125.969  on 91  degrees of freedom
## Residual deviance:  85.916  on 83  degrees of freedom
## AIC: 103.92
##
## Number of Fisher Scoring iterations: 8
```

```
vif(model_without_leptin)
```

```
##         Age     Glucose     Insulin        HOMA         BMI Adiponectin
##    1.422582    3.369777   75.034227   81.734207    1.435202    1.311805
##    Resistin       MCP.1
##    1.139302    1.237184
```

The AIC increases when removing BMI but decreases when removing Leptin. Thus, we should consider removing Leptin when fitting the data.

Final variables selection

```
model_with_selected_vars <- glm(Classification ~ Age + Glucose + BMI + Adiponectin + Resistin + MCP.1,
                                data=train_data,
                                family=binomial(link="logit"))
summary(model_with_selected_vars)
```

```
##
## Call:
## glm(formula = Classification ~ Age + Glucose + BMI + Adiponectin +
##     Resistin + MCP.1, family = binomial(link = "logit"), data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0547  -0.8146   0.1838   0.8303   2.0810
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.055241   2.824254  -2.144   0.0320 *
## Age         -0.032490   0.018630  -1.744   0.0812 .
## Glucose      0.112082   0.028802   3.891 9.96e-05 ***
## BMI         -0.137037   0.059552  -2.301   0.0214 *
## Adiponectin -0.017573   0.047355  -0.371   0.7106
## Resistin     0.037104   0.026799   1.385   0.1662
## MCP.1        0.002086   0.001013   2.059   0.0395 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 125.969  on 91  degrees of freedom
## Residual deviance:  87.917  on 85  degrees of freedom
## AIC: 101.92
##
## Number of Fisher Scoring iterations: 6
```

```
vif(model_with_selected_vars)
```

```
##         Age     Glucose         BMI Adiponectin    Resistin       MCP.1
##    1.412275    1.393049    1.311346    1.297189    1.125742    1.251648
```

Interaction terms selection

After possible removals of HOMA, Insulin, and Leptin, we have the remaining 6 variables, which are Age, Glucose, BMI, Adiponectin, Resistin, and MCP.1. We will now proceed to exploring the interactions between them.

Age interacts with remaining explanatory variables

It is safe to assume that a person gets more prone to adverse effects of irregular biological indicators as they get older. So we might want to add interaction terms between Age and Glucose/BMI/Adiponectin/Resistin/MCP.1

```
model_with_age_interactions <- glm(Classification ~ Age*Glucose + Age*BMI + Age*Adiponectin + Age*Resis
                                   data=train_data,
                                   family=binomial(link="logit"))
summary(model_with_age_interactions)
```

```
##
## Call:
## glm(formula = Classification ~ Age * Glucose + Age * BMI + Age *
##     Adiponectin + Age * Resistin + Age * MCP.1, family = binomial(link = "logit"),
##     data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.11514  -0.52868   0.02631   0.66189   2.57879
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.111e+00  1.404e+01  -0.079  0.93695
## Age             -1.978e-01  2.274e-01  -0.870  0.38431
## Glucose          3.877e-01  1.509e-01   2.569  0.01019 *
## BMI             -8.983e-01  3.209e-01  -2.799  0.00513 **
## Adiponectin     -5.982e-01  2.404e-01  -2.488  0.01283 *
## Resistin         1.736e-01  1.937e-01   0.896  0.37019
## MCP.1           -4.548e-03  5.509e-03  -0.826  0.40907
## Age:Glucose     -3.801e-03  2.088e-03  -1.820  0.06876 .
## Age:BMI          1.276e-02  5.514e-03   2.315  0.02063 *
## Age:Adiponectin  1.096e-02  4.338e-03   2.527  0.01149 *
## Age:Resistin    -2.194e-03  2.919e-03  -0.751  0.45236
## Age:MCP.1        1.206e-04  9.536e-05   1.265  0.20593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 125.969  on 91  degrees of freedom
## Residual deviance:  67.258  on 80  degrees of freedom
## AIC: 91.258
##
## Number of Fisher Scoring iterations: 7
```

```
vif(model_with_age_interactions)
```

```
##             Age         Glucose             BMI     Adiponectin        Resistin
##       140.98027        32.12691        24.98454        19.21182        63.66493
##           MCP.1     Age:Glucose         Age:BMI Age:Adiponectin    Age:Resistin
##        23.31879       175.61318        87.84911        18.35431        71.50008
##       Age:MCP.1
##        24.50012
```

The AIC decreases as we let Age interacts with Glucose, BMI, Adiponectin, Resistin, and MCP.1. Thus, we should consider adding these interaction terms when fitting the data.

Adiponectin interacts with BMI and Glucose

By definition, Adiponectin is a protein hormone and adipokine that is involved in the process of regulating glucose and fatty acid breakdown (source: https://en.wikipedia.org/wiki/Adiponectin). Thus, we want to explore how Adiponectin interacts with BMI and Glucose.

```
model_with_adiponectin_interactions <- glm(Classification ~ Age + Resistin + MCP.1 + Adiponectin*BMI + A
                                           data=train_data,
                                           family=binomial(link="logit"))
summary(model_with_adiponectin_interactions)
```

```
##
## Call:
## glm(formula = Classification ~ Age + Resistin + MCP.1 + Adiponectin *
##     BMI + Adiponectin * Glucose, family = binomial(link = "logit"),
##     data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1809  -0.6381   0.1322   0.6885   2.2529
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          2.903254   5.005402   0.580  0.56190
## Age                 -0.056290   0.022294  -2.525  0.01157 *
## Resistin             0.041279   0.025067   1.647  0.09961 .
## MCP.1                0.002002   0.001032   1.939  0.05246 .
## Adiponectin         -0.969111   0.486810  -1.991  0.04651 *
## BMI                 -0.437267   0.141064  -3.100  0.00194 **
## Glucose              0.111418   0.053759   2.073  0.03822 *
## Adiponectin:BMI      0.032397   0.014030   2.309  0.02094 *
## Adiponectin:Glucose  0.001770   0.004744   0.373  0.70913
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 125.969  on 91  degrees of freedom
## Residual deviance:  77.576  on 83  degrees of freedom
## AIC: 95.576
##
## Number of Fisher Scoring iterations: 6
```

```
vif(model_with_adiponectin_interactions)
```

```
##                 Age            Resistin               MCP.1         Adiponectin
##            1.762535            1.122996            1.227835           96.191157
##                 BMI             Glucose     Adiponectin:BMI Adiponectin:Glucose
##            6.114585            4.719082           41.038710           73.154292
```

The AIC decreases as we let Adiponectin interacts with Glucose, BMI. Thus, we should consider adding these interaction terms when fitting the data.

Resistin interacts with BMI and Glucose

It is theorized that Resistin links obesity to diabetes (source: https://en.wikipedia.org/wiki/Resistin). Thus, it might be worth it to explore how Resistin interacts with BMI and Glucose.

```
model_with_resistin_interactions <- glm(Classification ~ Age + Adiponectin + MCP.1 + Resistin*BMI + Resi
                                        data=train_data,
                                        family=binomial(link="logit"))
summary(model_with_resistin_interactions)
```

```
##
## Call:
## glm(formula = Classification ~ Age + Adiponectin + MCP.1 + Resistin *
##     BMI + Resistin * Glucose, family = binomial(link = "logit"),
##     data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.64052  -0.61366   0.03486   0.53042   1.97466
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.115e+01  6.463e+00  -3.272  0.00107 **
## Age              -3.412e-02  2.126e-02  -1.605  0.10851
## Adiponectin       1.836e-02  5.326e-02   0.345  0.73026
## MCP.1             4.725e-04  1.067e-03   0.443  0.65796
## Resistin          1.413e+00  5.214e-01   2.710  0.00673 **
## BMI               2.226e-01  1.547e-01   1.439  0.15009
## Glucose           1.543e-01  4.801e-02   3.215  0.00131 **
## Resistin:BMI     -3.416e-02  1.485e-02  -2.299  0.02148 *
## Resistin:Glucose -2.744e-03  2.407e-03  -1.140  0.25441
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 125.969  on 91  degrees of freedom
## Residual deviance:  70.427  on 83  degrees of freedom
## AIC: 88.427
##
## Number of Fisher Scoring iterations: 8
```

```
vif(model_with_resistin_interactions)
```

```
##              Age      Adiponectin            MCP.1         Resistin
##         1.467576         1.418601         1.459538       129.567361
##              BMI          Glucose     Resistin:BMI Resistin:Glucose
##         6.452823         2.614982       120.976352        25.863046
```

The AIC decreases as we let Resistin interacts with Glucose, BMI. Thus, we should consider adding these interaction terms when fitting the data.

Best model fitted manually

```
manual_best_model <- glm(Classification ~ Age*Glucose + Age*BMI + Age*Adiponectin + Age*Resistin + Age*
                         data=train_data,
                         family=binomial(link="logit"))
summary(manual_best_model)
```

```
##
```

```
## Call:
## glm(formula = Classification ~ Age * Glucose + Age * BMI + Age *
##     Adiponectin + Age * Resistin + Age * MCP.1 + Adiponectin *
##     BMI + Adiponectin * Glucose + Resistin * BMI + Resistin *
##     Glucose, family = binomial(link = "logit"), data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.80376  -0.28543   0.01229   0.39773   2.04256
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         3.2522389 21.4576742   0.152   0.8795
## Age                -0.3632648  0.3162572  -1.149   0.2507
## Glucose             0.3376885  0.1718030   1.966   0.0493 *
## BMI                -0.7410608  0.5012549  -1.478   0.1393
## Adiponectin        -2.0130825  0.9969315  -2.019   0.0435 *
## Resistin            1.6549456  0.8037469   2.059   0.0395 *
## MCP.1              -0.0080955  0.0066450  -1.218   0.2231
## Age:Glucose        -0.0037269  0.0019853  -1.877   0.0605 .
## Age:BMI             0.0152563  0.0085781   1.779   0.0753 .
## Age:Adiponectin     0.0112145  0.0088843   1.262   0.2068
## Age:Resistin        0.0018132  0.0047068   0.385   0.7001
## Age:MCP.1           0.0001455  0.0001147   1.268   0.2048
## BMI:Adiponectin     0.0184718  0.0193524   0.954   0.3398
## Glucose:Adiponectin 0.0098245  0.0100674   0.976   0.3291
## BMI:Resistin       -0.0430745  0.0226612  -1.901   0.0573 .
## Glucose:Resistin   -0.0034566  0.0018264  -1.893   0.0584 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 125.969  on 91  degrees of freedom
## Residual deviance:  49.364  on 76  degrees of freedom
## AIC: 81.364
##
## Number of Fisher Scoring iterations: 8
```

```
vif(manual_best_model)
```

```
##                 Age             Glucose                 BMI         Adiponectin
##           166.28714            42.13098            41.79494           162.33109
##            Resistin               MCP.1         Age:Glucose             Age:BMI
##           211.07202            39.52076           114.59459           139.38002
##     Age:Adiponectin        Age:Resistin           Age:MCP.1     BMI:Adiponectin
##            45.41396            35.19782            39.59140            34.57630
## Glucose:Adiponectin        BMI:Resistin    Glucose:Resistin
##           159.96802           175.41112            18.86229
```

```
manual_best_model.pred <- predict(manual_best_model, newdata=test_data, type="response")
```

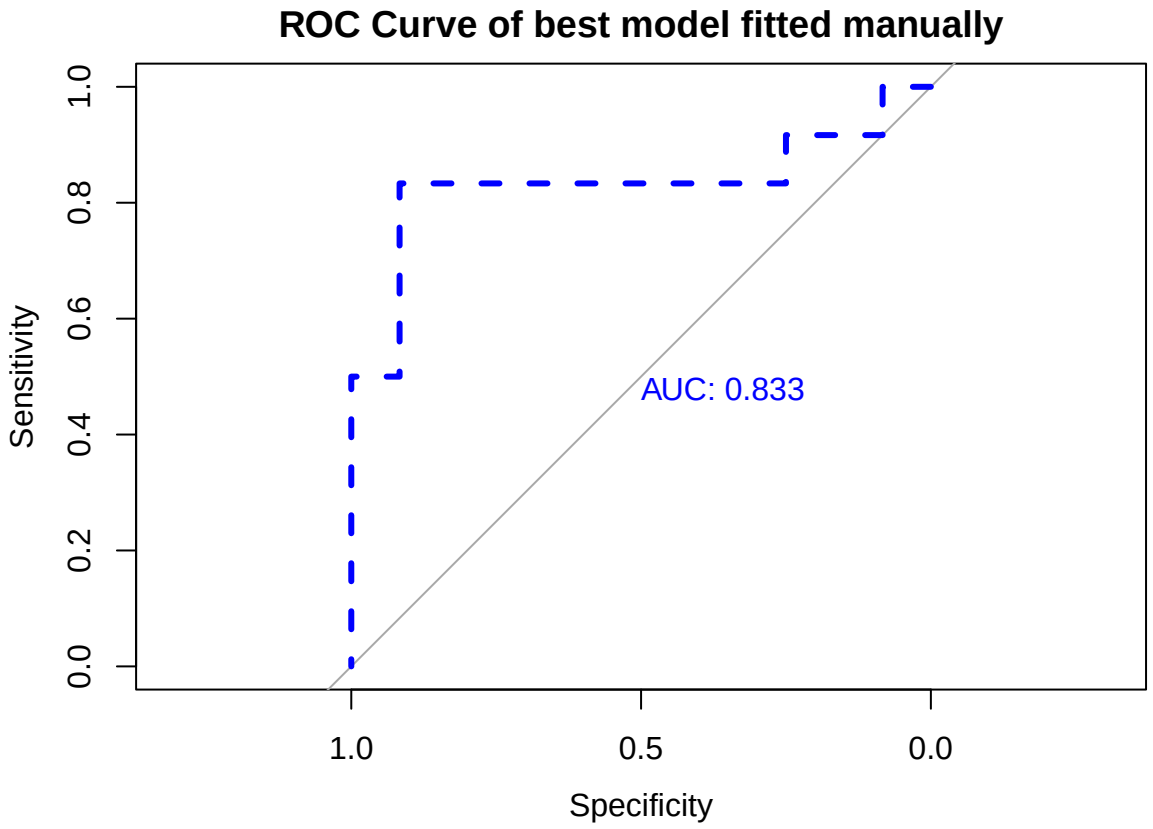
Confusion matrix

```
cut_off <- 0.5
manual_best_model.classified_pred <- as.integer(manual_best_model.pred > cut_off)
confusionMatrix(data=as.factor(manual_best_model.classified_pred),
                reference=as.factor(test_data$Classification),
                positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 10  2
##          1  2 10
##
##                Accuracy : 0.8333
##                  95% CI : (0.6262, 0.9526)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 0.0007719
##
##                   Kappa : 0.6667
##
##  Mcnemar's Test P-Value : 1.0000000
##
##             Sensitivity : 0.8333
##             Specificity : 0.8333
##          Pos Pred Value : 0.8333
##          Neg Pred Value : 0.8333
##              Prevalence : 0.5000
##          Detection Rate : 0.4167
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.8333
##
##        'Positive' Class : 1
##
```

ROC Curve and AUC

```
manual_best_model.roc_curve <- roc(response=test_data$Classification,
                                   predictor=manual_best_model.pred)
plot(manual_best_model.roc_curve,
     print.auc=TRUE, col="blue", lwd=3, lty=2,
     main="ROC Curve of best model fitted manually")
```



Exhaustive model selection using BestGlm method

```
train_Xy <- subset(train_data, select=-c(Insulin, HOMA, Leptin))
names(train_Xy)[names(train_Xy) == "Classification"] <- "y"
bestglm_best_model <- bestglm(Xy=train_Xy,
                        family=binomial(link="logit"),
                        IC="AIC",
                        method="exhaustive")$BestModel
summary(bestglm_best_model)
```

```
##
## Call:
## glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0757  -0.8179   0.1864   0.8083   2.0627
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.464731   2.601089  -2.485 0.012941 *
## Age         -0.030051   0.017389  -1.728 0.083959 .
## BMI         -0.130851   0.057137  -2.290 0.022014 *
## Glucose      0.111113   0.028585   3.887 0.000101 ***
## Resistin     0.038288   0.026893   1.424 0.154537
## MCP.1        0.002092   0.001015   2.061 0.039329 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 125.969  on 91  degrees of freedom
## Residual deviance:  88.055  on 86  degrees of freedom
## AIC: 100.06
##
## Number of Fisher Scoring iterations: 6
```

```
vif(bestglm_best_model)
```

```
##      Age      BMI  Glucose Resistin    MCP.1
## 1.223343 1.197218 1.373557 1.112440 1.256998
```

```
bestglm_best_model.pred <- predict(bestglm_best_model, newdata=test_data, type="response")
```

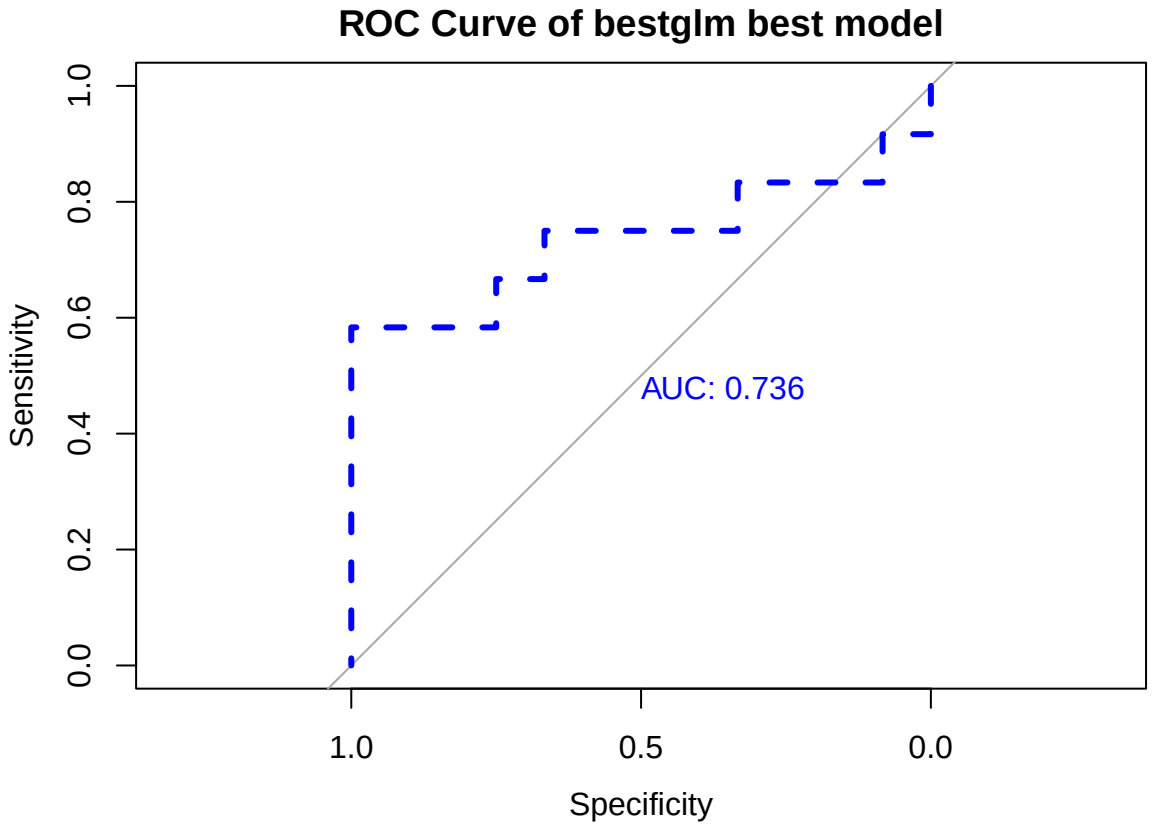
Confusion matrix

```
cut_off <- 0.5
bestglm_best_model.classified_pred <- as.integer(bestglm_best_model.pred > cut_off)
confusionMatrix(data=as.factor(bestglm_best_model.classified_pred),
                reference=as.factor(test_data$Classification),
                positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 7 3
##          1 5 9
##
##                Accuracy : 0.6667
##                  95% CI : (0.4468, 0.8437)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 0.07579
##
##                   Kappa : 0.3333
##
##  Mcnemar's Test P-Value : 0.72367
##
##             Sensitivity : 0.7500
##             Specificity : 0.5833
##          Pos Pred Value : 0.6429
##          Neg Pred Value : 0.7000
##              Prevalence : 0.5000
##          Detection Rate : 0.3750
##    Detection Prevalence : 0.5833
##       Balanced Accuracy : 0.6667
##
##        'Positive' Class : 1
##
```

ROC Curve and AUC

```
bestglm_best_model.roc_curve <- roc(response=test_data$Classification,
                                    predictor=bestglm_best_model.pred)
plot(bestglm_best_model.roc_curve,
     print.auc=TRUE, col="blue", lwd=3, lty=2,
     main="ROC Curve of bestglm best model")
```



Best model with interactions using glmulti

```
glmulti_best_model <- glmulti("Classification",
                              c("Age", "Glucose", "BMI", "Adiponectin", "Resistin", "MCP.1"),
                              data=train_data,
                              level=2,
                              method="h",
                              crit="aic",
                              confsetsize=6,
                              plotty=FALSE,
                              report=FALSE,
                              fitfunction="glm",
                              family=binomial(link="logit"))@objects[[1]]
summary(glmulti_best_model)
```

```
##
## Call:
## fitfunc(formula = as.formula(x), family = ..1, data = data)
##
```

```
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.25976  -0.43850   0.00642   0.38895   2.28001
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          12.969148   7.257428   1.787  0.07393 .
## BMI                  -1.003667   0.474774  -2.114  0.03452 *
## Adiponectin          -2.189171   0.715181  -3.061  0.00221 **
## Resistin              1.147080   0.405924   2.826  0.00472 **
## Glucose:Age          -0.003581   0.001177  -3.042  0.00235 **
## BMI:Age               0.009058   0.003464   2.615  0.00892 **
## BMI:Glucose           0.007626   0.003232   2.360  0.01828 *
## Adiponectin:Glucose   0.020256   0.007212   2.809  0.00497 **
## BMI:Resistin         -0.039867   0.013109  -3.041  0.00236 **
## Adiponectin:Resistin  0.029949   0.014444   2.073  0.03813 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 125.969  on 91  degrees of freedom
## Residual deviance:  54.079  on 82  degrees of freedom
## AIC: 74.079
##
## Number of Fisher Scoring iterations: 7
```

```
vif(glmulti_best_model)
```

```
##                  BMI          Adiponectin             Resistin
##            40.560512           151.362961            92.227657
##          Glucose:Age              BMI:Age          BMI:Glucose
##            57.265717            24.322010            29.331792
##  Adiponectin:Glucose         BMI:Resistin Adiponectin:Resistin
##           133.769765           104.899657             8.148751
```

```
glmulti_best_model.pred <- predict(glmulti_best_model, newdata=test_data, type="response")
```

Confusion matrix

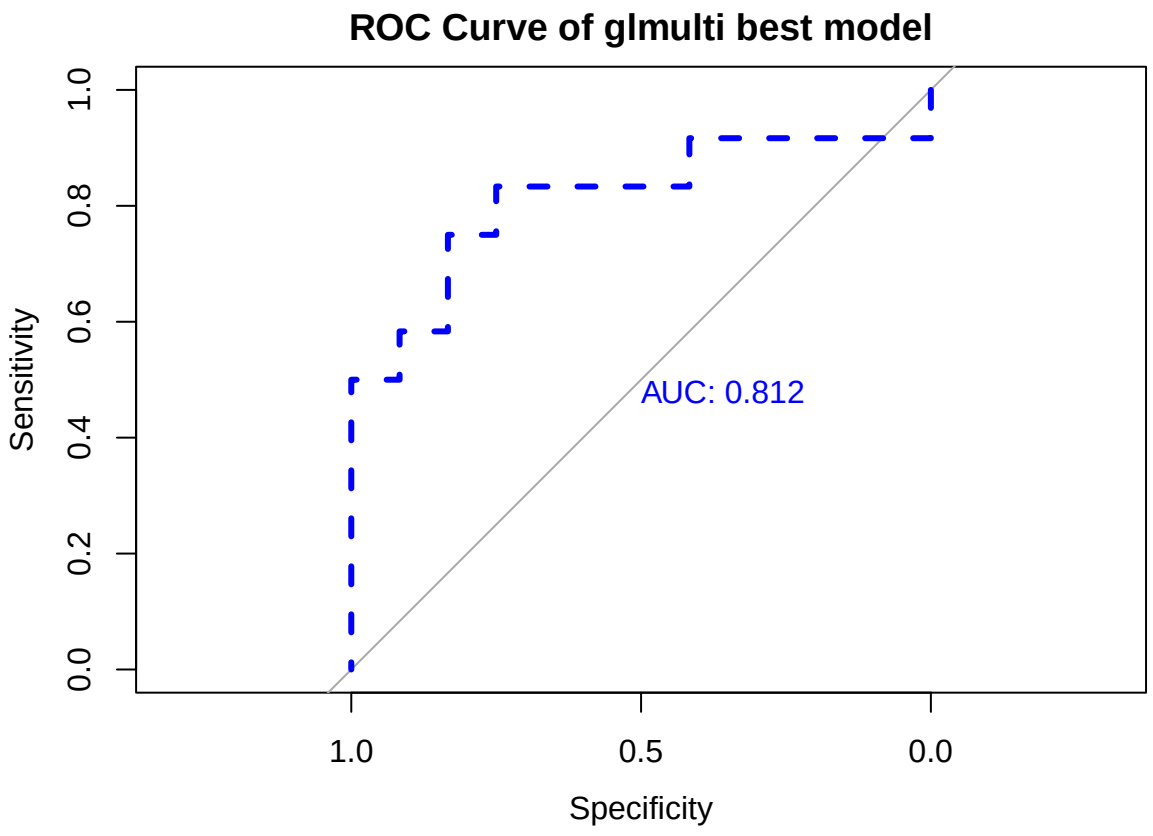
```
cut_off <- 0.5
glmulti_best_model.classified_pred <- as.integer(glmulti_best_model.pred > cut_off)
confusionMatrix(data=as.factor(glmulti_best_model.classified_pred),
                reference=as.factor(test_data$Classification),
                positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  9  2
##          1  3 10
##
##                Accuracy : 0.7917
##                  95% CI : (0.5785, 0.9287)
```

```
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 0.003305
##
##                   Kappa : 0.5833
##
##  Mcnemar's Test P-Value : 1.000000
##
##             Sensitivity : 0.8333
##             Specificity : 0.7500
##          Pos Pred Value : 0.7692
##          Neg Pred Value : 0.8182
##              Prevalence : 0.5000
##          Detection Rate : 0.4167
##    Detection Prevalence : 0.5417
##       Balanced Accuracy : 0.7917
##
##        'Positive' Class : 1
##
```

ROC Curve and AUC

```
glmulti_best_model.roc_curve <- roc(response=test_data$Classification,
                                    predictor=glmulti_best_model.pred)
plot(glmulti_best_model.roc_curve,
     print.auc=TRUE, col="blue", lwd=3, lty=2,
     main="ROC Curve of glmulti best model")
```



Comparisons between best model fitted manually, by bestglm, and by glmulti

Worst performing best model

In terms of AIC and AUC, best model fitted by bestglm has the worst performance (AIC=100.06, AUC=0.736). It should also be noted that this model does not include any interaction terms. Thus, it is reasonable to make an assumption that good candidate models should include some interactions between explanatory variables.

Manual fitting v. glmulti

In terms of AIC, best model fitted by glmulti performs better than the one fitted manually.

In terms of accuracy and AUC, best model fitted manually performs better than the one fitted by glmulti.

Cross-validation

```
set.seed(268)
fold1 <- slice_sample(bccdat, prop = 0.25)
fold2 <- slice_sample(anti_join(bccdat, fold1), prop = 1/3)
fold3 <- slice_sample(anti_join(bccdat, bind_rows(fold1, fold2)), prop = 1/2)
fold4 <- anti_join(bccdat, bind_rows(fold1, fold2, fold3))
folds <- list(fold1, fold2, fold3, fold4)
```

```
cv_results.accuracies <- data.frame(manual_best_model.accuracies=c(), glmulti_best_model.accuracies=c())
cv_results.aucs <- data.frame(manual_best_model.aucs=c(), glmulti_best_model.aucs=c())
```

```
local({
  for (i in 1:4) {
    cvsplit_i.train_data <- anti_join(bccdat, folds[[i]])
    cvsplit_i.test_data <- folds[[i]]

    cut_off <- 0.5

    cvsplit_i.manual_best_model <- glm(formula=manual_best_model$formula,
                                       data=cvsplit_i.train_data,
                                       family=binomial(link="logit"))
    cvsplit_i.manual_best_model.pred <- predict(cvsplit_i.manual_best_model,
                                                newdata=cvsplit_i.test_data,
                                                type="response")
    # accuracy of manual best model at split i
    cvsplit_i.manual_best_model.classified_pred <-
      as.integer(cvsplit_i.manual_best_model.pred > cut_off)
    cvsplit_i.manual_best_model.accuracy <-
      confusionMatrix(data=as.factor(cvsplit_i.manual_best_model.classified_pred),
                      reference=as.factor(cvsplit_i.test_data$Classification),
                      positive="1")$overall[[1]]
    # auc of manual best model at split i
    cvsplit_i.manual_best_model.roc_curve <-
      roc(response=cvsplit_i.test_data$Classification,
          predictor=cvsplit_i.manual_best_model.pred)


    cvsplit_i.glmulti_best_model <- glm(formula=glmulti_best_model$formula,
                                        data=cvsplit_i.train_data,
                                        family=binomial(link="logit"))
    cvsplit_i.glmulti_best_model.pred <- predict(cvsplit_i.glmulti_best_model,
```

```
                                                newdata=cvsplit_i.test_data,
                                                type="response")
    # accuracy of glmulti best model at split i
    cvsplit_i.glmulti_best_model.classified_pred <-
      as.integer(cvsplit_i.glmulti_best_model.pred > cut_off)
    cvsplit_i.glmulti_best_model.accuracy <-
      confusionMatrix(data=as.factor(cvsplit_i.glmulti_best_model.classified_pred),
                      reference=as.factor(cvsplit_i.test_data$Classification),
                      positive="1")$overall[[1]]
    # auc of glmulti best model at split i
    cvsplit_i.glmulti_best_model.roc_curve <-
      roc(response=cvsplit_i.test_data$Classification,
          predictor=cvsplit_i.glmulti_best_model.pred)

    cvsplit_i.result.accuracies <-
      data.frame(manual_best_model.accuracies=c(cvsplit_i.manual_best_model.accuracy),
                 glmulti_best_model.accuracies=c(cvsplit_i.glmulti_best_model.accuracy))
    cv_results.accuracies <<- bind_rows(cv_results.accuracies, cvsplit_i.result.accuracies)

    cvsplit_i.result.aucs <-
      data.frame(manual_best_model.aucs=c(cvsplit_i.manual_best_model.roc_curve$auc),
                 glmulti_best_model.aucs=c(cvsplit_i.glmulti_best_model.roc_curve$auc))
    cv_results.aucs <<- bind_rows(cv_results.aucs, cvsplit_i.result.aucs)
  }
})
```

```
colnames(cv_results.accuracies) <- c("Best model fitted manually", "Best model fitted by glmulti")
row.names(cv_results.accuracies) <- c("Accuracy of fold1", "Accuracy of fold2", "Accuracy of fold3", "A
knitr::kable((cv_results.accuracies), format = "simple", digits = 3)
```

	Best model fitted manually	Best model fitted by glmulti
Accuracy of fold1	0.793	0.828
Accuracy of fold2	0.621	0.690
Accuracy of fold3	0.759	0.655
Accuracy of fold4	0.793	0.897

```
colnames(cv_results.aucs) <- c("Best model fitted manually", "Best model fitted by glmulti")
row.names(cv_results.aucs) <- c("AUC of fold1", "AUC of fold2", "AUC of fold3", "AUC of fold4")
knitr::kable((cv_results.aucs), format = "simple", digits = 3)
```

	Best model fitted manually	Best model fitted by glmulti
AUC of fold1	0.818	0.909
AUC of fold2	0.611	0.764
AUC of fold3	0.774	0.631
AUC of fold4	0.904	0.938

Result

As we can see from our cross-validation results, the performance differentials between the best models fitted manually and by glmulti are pretty even. Best model fitted manually offers better interpretability, whereas best model fitted by glmulti is smaller in size. Based on personal preference, I'm choosing the best model fitted manually as the main logistic regression for the Breast Cancer Coimbra data set.

```r
main_model <- manual_best_model
summary(main_model)
```

```
##
## Call:
## glm(formula = Classification ~ Age * Glucose + Age * BMI + Age *
##     Adiponectin + Age * Resistin + Age * MCP.1 + Adiponectin *
##     BMI + Adiponectin * Glucose + Resistin * BMI + Resistin *
##     Glucose, family = binomial(link = "logit"), data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.80376  -0.28543   0.01229   0.39773   2.04256
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          3.2522389 21.4576742   0.152   0.8795
## Age                 -0.3632648  0.3162572  -1.149   0.2507
## Glucose              0.3376885  0.1718030   1.966   0.0493 *
## BMI                 -0.7410608  0.5012549  -1.478   0.1393
## Adiponectin         -2.0130825  0.9969315  -2.019   0.0435 *
## Resistin             1.6549456  0.8037469   2.059   0.0395 *
## MCP.1               -0.0080955  0.0066450  -1.218   0.2231
## Age:Glucose         -0.0037269  0.0019853  -1.877   0.0605 .
## Age:BMI              0.0152563  0.0085781   1.779   0.0753 .
## Age:Adiponectin      0.0112145  0.0088843   1.262   0.2068
## Age:Resistin         0.0018132  0.0047068   0.385   0.7001
## Age:MCP.1            0.0001455  0.0001147   1.268   0.2048
## BMI:Adiponectin      0.0184718  0.0193524   0.954   0.3398
## Glucose:Adiponectin  0.0098245  0.0100674   0.976   0.3291
## BMI:Resistin        -0.0430745  0.0226612  -1.901   0.0573 .
## Glucose:Resistin    -0.0034566  0.0018264  -1.893   0.0584 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 125.969  on 91  degrees of freedom
## Residual deviance:  49.364  on 76  degrees of freedom
## AIC: 81.364
##
## Number of Fisher Scoring iterations: 8
```