```
# linear regression using diabetic data(BMI vs DBP or PGC)

View(Diabatic_data1)

data1<-Diabatic_data1

View(data1)


set.seed(2)

# sample.split function is present in this package

library(caTools)

#we divide the data with ratio 0.7

split<- sample.split(data1$HasRiskDiabetes, SplitRatio = 0.8)

# we will split data into training and testing data set

training_data1<-subset(data1, split="TRUE")    # as TRUE so training data set

testing_data1<- subset(data1, split="FALSE")   # as FALSE so testing data set

# see the summary of data1

summary(data1)

# data visualization

plot(data1$BodyMassIndex, data1$DiastolicBloodPressureInMMHG)

# Build model lm<- for linear regression (. means all variable will include)

data1_model<- lm(data1$HasRiskDiabetes~., data = training_data1) # DiastolicBloodPressureInMMHG -
> dependent var

summary(data1_model)                                          # if p value not <0.05%  then remove
that variable

# predict the value

data1_predict<- predict(data1_model, newdata = testing_data1)


data1_predict


TAB<-table(testing_data1$HasRiskDiabetes, data1_predict>0.05)
```

```
TAB
#   FALSE TRUE
#0     81  419
#1     3  265
(81+265)/(81+3+419+265)
# 0.4505208 = 45% acqurate

table(Actualvalue=testing_data1$HasRiskDiabetes, PredictedvalueBymodel=data1_predict> 0.05)

# acquracy of the model

#_____________________________________________________________
# To compare predicted values and actual(test_value) values, we can use plot
#plot(testing_data1$BodyMassIndex, type = "l", lty=1.8, col="green")
#lines(data1_predict, type = "l", col="blue")
#_____________________________________________________________________

plot(data1$BodyMassIndex, data1$DiastolicBloodPressureInMMHG)
# to draw regression line
plot(data1$BodyMassIndex, data1$DiastolicBloodPressureInMMHG)
abline(lm(data1$BodyMassIndex~data1$DiastolicBloodPressureInMMHG), col="red")

# to see corplot
cr<- cor(data1)
```

```
library(corrplot)


corrplot(cr, type = "lower")

corrplot(cr, method = "number")
```