pima\_indians\_diabetes\_study

Amit k singh

1 October 2017

library(readr)  
diabetes <- read\_csv("D:/R Working Directory/pima\_indians\_diabetes\_study/diabetes.csv")

## Parsed with column specification:  
## cols(  
## prgnt = col\_integer(),  
## glucose = col\_integer(),  
## bp = col\_integer(),  
## skin = col\_integer(),  
## Insulin = col\_integer(),  
## BMI = col\_double(),  
## pedigree = col\_double(),  
## age = col\_integer(),  
## class = col\_integer()  
## )

View(diabetes)  
dim(diabetes)

## [1] 768 9

str(diabetes)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 768 obs. of 9 variables:  
## $ prgnt : int 6 1 8 1 0 5 3 10 2 8 ...  
## $ glucose : int 148 85 183 89 137 116 78 115 197 125 ...  
## $ bp : int 72 66 64 66 40 74 50 0 70 96 ...  
## $ skin : int 35 29 0 23 35 0 32 0 45 0 ...  
## $ Insulin : int 0 0 0 94 168 0 88 0 543 0 ...  
## $ BMI : num 33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...  
## $ pedigree: num 0.627 0.351 0.672 0.167 2.288 ...  
## $ age : int 50 31 32 21 33 30 26 29 53 54 ...  
## $ class : int 1 0 1 0 1 0 1 0 1 1 ...  
## - attr(\*, "spec")=List of 2  
## ..$ cols :List of 9  
## .. ..$ prgnt : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ glucose : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ bp : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ skin : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ Insulin : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ BMI : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_double" "collector"  
## .. ..$ pedigree: list()  
## .. .. ..- attr(\*, "class")= chr "collector\_double" "collector"  
## .. ..$ age : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ class : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## ..$ default: list()  
## .. ..- attr(\*, "class")= chr "collector\_guess" "collector"  
## ..- attr(\*, "class")= chr "col\_spec"

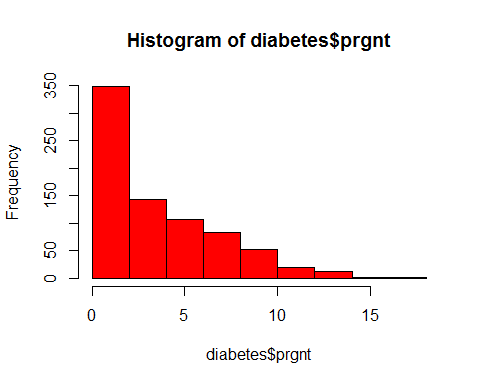
colnames(diabetes)

## [1] "prgnt" "glucose" "bp" "skin" "Insulin" "BMI"   
## [7] "pedigree" "age" "class"

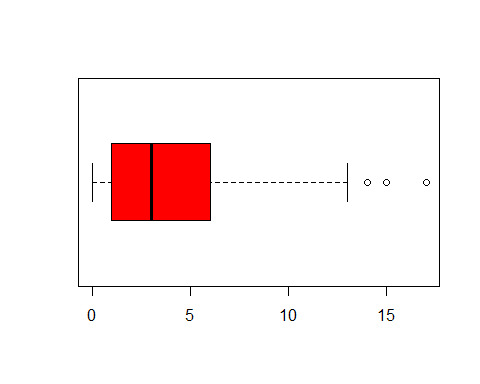
table(diabetes$class)

##   
## 0 1   
## 500 268

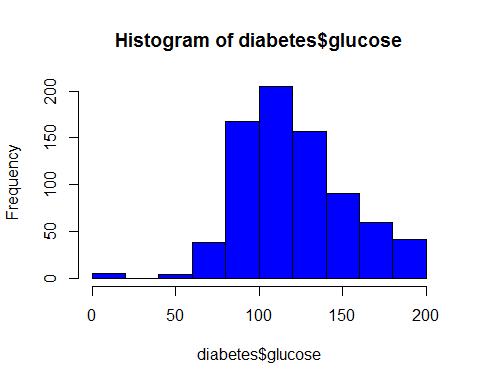
hist(diabetes$prgnt, col = "red")



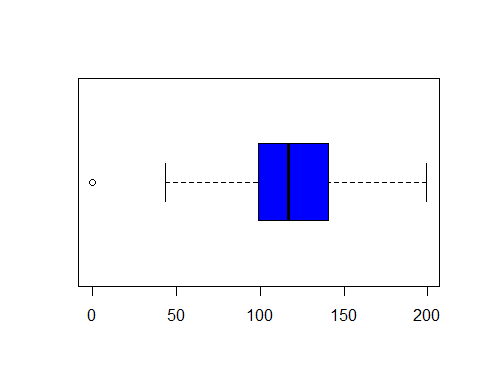
boxplot(diabetes$prgnt, col = "red", horizontal = T)



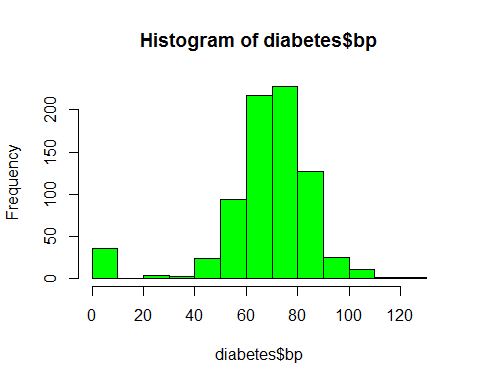
hist(diabetes$glucose, col = "blue")



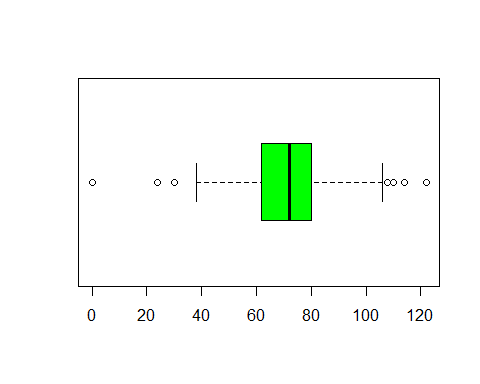
boxplot(diabetes$glucose, col = "blue", horizontal = T)



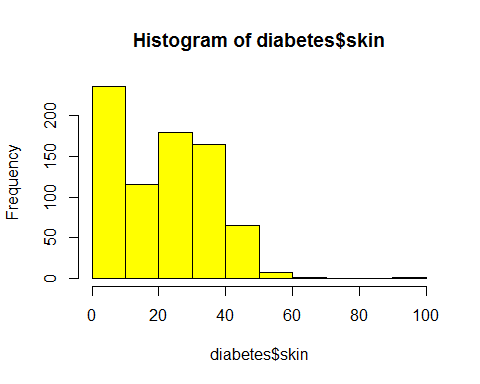
hist(diabetes$bp, col = "green")



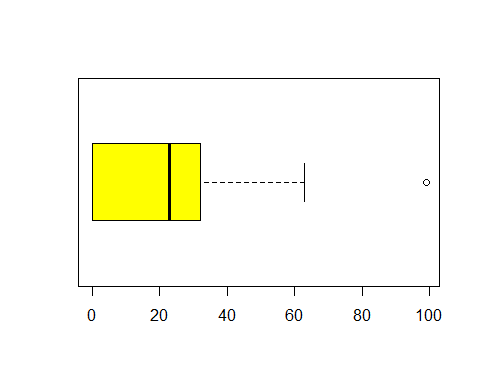
boxplot(diabetes$bp, col = "green", horizontal = T)



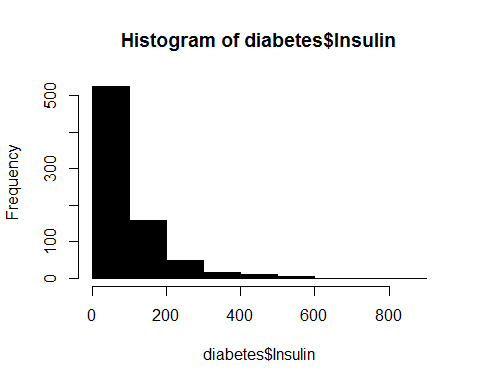
hist(diabetes$skin, col = "yellow")



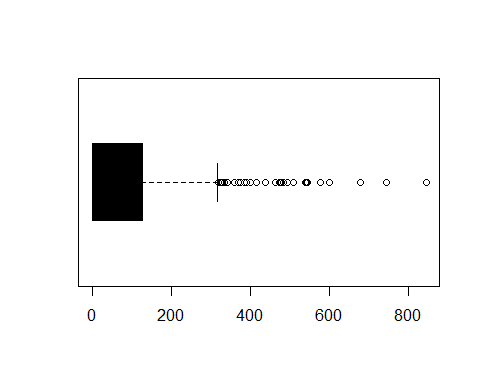
boxplot(diabetes$skin, col = "yellow", horizontal = T)



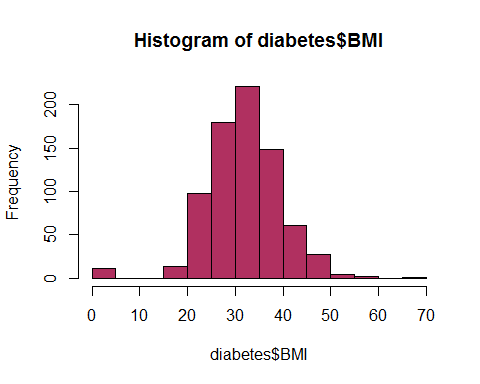
hist(diabetes$Insulin, col = "black")



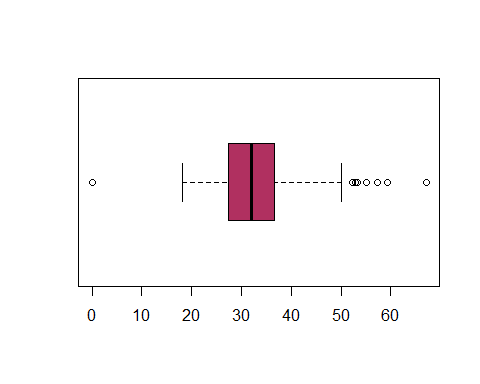
boxplot(diabetes$Insulin, col = "black", horizontal = T)



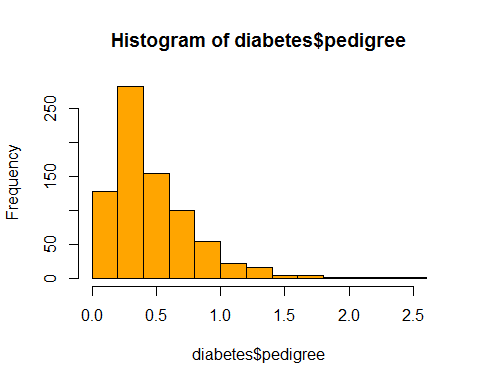
hist(diabetes$BMI, col = "maroon")



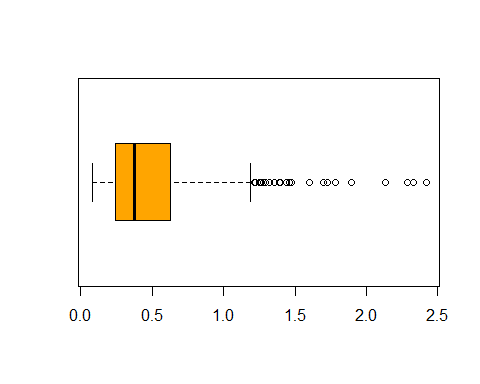
boxplot(diabetes$BMI, col = "maroon", horizontal = T)



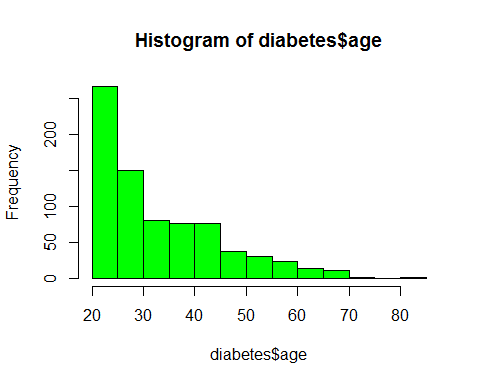
hist(diabetes$pedigree, col = "orange")



boxplot(diabetes$pedigree, col = "orange", horizontal = T)



hist(diabetes$age, col = "green")



boxplot(diabetes$age, col = "green", horizontal = T)  
  
library(caTools)  
set.seed(88)  
split<- sample.split(diabetes$class, SplitRatio = 0.75)  
diabetesTrain<- subset(diabetes, split == T)  
diabetesTest<- subset(diabetes, split == F)  
dim(diabetesTrain) # 576 by 9

## [1] 576 9

dim(diabetesTest)# 192 by 9

## [1] 192 9

str(diabetesTrain)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 576 obs. of 9 variables:  
## $ prgnt : int 6 1 8 1 0 3 10 10 5 0 ...  
## $ glucose : int 148 85 183 89 137 78 168 139 166 118 ...  
## $ bp : int 72 66 64 66 40 50 74 80 72 84 ...  
## $ skin : int 35 29 0 23 35 32 0 0 19 47 ...  
## $ Insulin : int 0 0 0 94 168 88 0 0 175 230 ...  
## $ BMI : num 33.6 26.6 23.3 28.1 43.1 31 38 27.1 25.8 45.8 ...  
## $ pedigree: num 0.627 0.351 0.672 0.167 2.288 ...  
## $ age : int 50 31 32 21 33 26 34 57 51 31 ...  
## $ class : int 1 0 1 0 1 1 1 0 1 1 ...

str(diabetesTest)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 192 obs. of 9 variables:  
## $ prgnt : int 5 10 2 8 4 1 7 3 3 11 ...  
## $ glucose : int 116 115 197 125 110 189 100 158 88 138 ...  
## $ bp : int 74 0 70 96 92 60 0 76 58 76 ...  
## $ skin : int 0 0 45 0 0 23 0 36 11 0 ...  
## $ Insulin : int 0 0 543 0 0 846 0 245 54 0 ...  
## $ BMI : num 25.6 35.3 30.5 0 37.6 30.1 30 31.6 24.8 33.2 ...  
## $ pedigree: num 0.201 0.134 0.158 0.232 0.191 0.398 0.484 0.851 0.267 0.42 ...  
## $ age : int 30 29 53 54 30 59 32 28 22 35 ...  
## $ class : int 0 0 1 1 0 1 1 1 0 0 ...

library("mosaic")

## Warning: package 'mosaic' was built under R version 3.4.1

## Loading required package: dplyr

## Warning: package 'dplyr' was built under R version 3.4.1

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

## Loading required package: lattice

## Loading required package: ggformula

## Warning: package 'ggformula' was built under R version 3.4.1

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.4.1

##   
## New to ggformula? Try the tutorials:   
## learnr::run\_tutorial("introduction", package = "ggformula")  
## learnr::run\_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

## Warning: package 'mosaicData' was built under R version 3.4.1

## Loading required package: Matrix

##   
## The 'mosaic' package masks several functions from core packages in order to add   
## additional features. The original behavior of these functions should not be affected by this.  
##   
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.

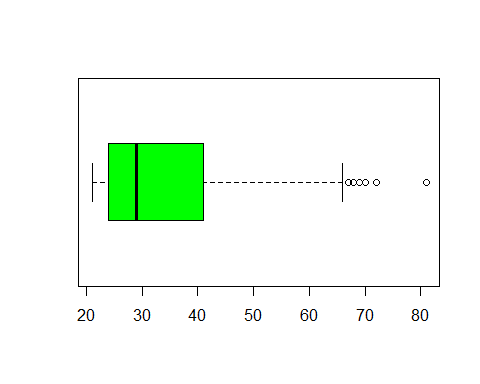
##   
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':  
##   
## mean

## The following objects are masked from 'package:dplyr':  
##   
## count, do, tally

## The following objects are masked from 'package:stats':  
##   
## binom.test, cor, cor.test, cov, fivenum, IQR, median,  
## prop.test, quantile, sd, t.test, var

## The following objects are masked from 'package:base':  
##   
## max, mean, min, prod, range, sample, sum



inspect (diabetesTrain)

##   
## quantitative variables:   
## name class min Q1 median Q3 max mean  
## 1 prgnt integer 0.000 1.0000 3.000 6.00000 17.00 3.8402778  
## 2 glucose integer 0.000 99.0000 118.000 142.25000 199.00 121.8246528  
## 3 bp integer 0.000 62.0000 72.000 80.00000 122.00 68.8888889  
## 4 skin integer 0.000 0.0000 23.000 33.00000 63.00 20.5798611  
## 5 Insulin integer 0.000 0.0000 37.000 130.00000 744.00 83.0850694  
## 6 BMI numeric 0.000 27.3750 31.950 36.42500 67.10 31.9190972  
## 7 pedigree numeric 0.078 0.2395 0.365 0.61225 2.42 0.4730521  
## 8 age integer 21.000 24.0000 29.000 41.00000 81.00 33.3072917  
## 9 class integer 0.000 0.0000 0.000 1.00000 1.00 0.3489583  
## sd n missing  
## 1 3.2712996 576 0  
## 2 31.9518466 576 0  
## 3 19.7716041 576 0  
## 4 15.9234214 576 0  
## 5 116.4421620 576 0  
## 6 8.1223246 576 0  
## 7 0.3417397 576 0  
## 8 11.5527230 576 0  
## 9 0.4770551 576 0

inspect(diabetesTest)

##   
## quantitative variables:   
## name class min Q1 median Q3 max mean  
## 1 prgnt integer 0.000 1.00000 3.000 6.00000 15.000 3.8593750  
## 2 glucose integer 0.000 99.75000 113.000 134.50000 198.000 118.1041667  
## 3 bp integer 0.000 64.00000 70.000 78.00000 110.000 69.7552083  
## 4 skin integer 0.000 0.00000 22.000 32.00000 99.000 20.4062500  
## 5 Insulin integer 0.000 0.00000 0.000 105.25000 846.000 69.9427083  
## 6 BMI numeric 0.000 27.17500 32.750 36.95000 52.900 32.2130208  
## 7 pedigree numeric 0.085 0.24775 0.391 0.66525 1.893 0.4683490  
## 8 age integer 21.000 23.00000 28.000 40.00000 69.000 33.0416667  
## 9 class integer 0.000 0.00000 0.000 1.00000 1.000 0.3489583  
## sd n missing  
## 1 3.6575985 192 0  
## 2 31.9556234 192 0  
## 3 18.0849500 192 0  
## 4 16.0793700 192 0  
## 5 111.2870062 192 0  
## 6 7.1374968 192 0  
## 7 0.2987325 192 0  
## 8 12.3910259 192 0  
## 9 0.4778869 192 0

head(diabetesTrain, 3)

## # A tibble: 3 x 9  
## prgnt glucose bp skin Insulin BMI pedigree age class  
## <int> <int> <int> <int> <int> <dbl> <dbl> <int> <int>  
## 1 6 148 72 35 0 33.6 0.627 50 1  
## 2 1 85 66 29 0 26.6 0.351 31 0  
## 3 8 183 64 0 0 23.3 0.672 32 1

head(diabetesTest, 3)

## # A tibble: 3 x 9  
## prgnt glucose bp skin Insulin BMI pedigree age class  
## <int> <int> <int> <int> <int> <dbl> <dbl> <int> <int>  
## 1 5 116 74 0 0 25.6 0.201 30 0  
## 2 10 115 0 0 0 35.3 0.134 29 0  
## 3 2 197 70 45 543 30.5 0.158 53 1

diabetes$class<-as.factor(diabetes$class)  
diabetesTrain$class<-as.factor(diabetesTrain$class)  
diabetesTest$class<-as.factor(diabetesTest$class)  
  
str(diabetes)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 768 obs. of 9 variables:  
## $ prgnt : int 6 1 8 1 0 5 3 10 2 8 ...  
## $ glucose : int 148 85 183 89 137 116 78 115 197 125 ...  
## $ bp : int 72 66 64 66 40 74 50 0 70 96 ...  
## $ skin : int 35 29 0 23 35 0 32 0 45 0 ...  
## $ Insulin : int 0 0 0 94 168 0 88 0 543 0 ...  
## $ BMI : num 33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...  
## $ pedigree: num 0.627 0.351 0.672 0.167 2.288 ...  
## $ age : int 50 31 32 21 33 30 26 29 53 54 ...  
## $ class : Factor w/ 2 levels "0","1": 2 1 2 1 2 1 2 1 2 2 ...  
## - attr(\*, "spec")=List of 2  
## ..$ cols :List of 9  
## .. ..$ prgnt : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ glucose : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ bp : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ skin : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ Insulin : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ BMI : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_double" "collector"  
## .. ..$ pedigree: list()  
## .. .. ..- attr(\*, "class")= chr "collector\_double" "collector"  
## .. ..$ age : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ class : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## ..$ default: list()  
## .. ..- attr(\*, "class")= chr "collector\_guess" "collector"  
## ..- attr(\*, "class")= chr "col\_spec"

str(diabetesTrain)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 576 obs. of 9 variables:  
## $ prgnt : int 6 1 8 1 0 3 10 10 5 0 ...  
## $ glucose : int 148 85 183 89 137 78 168 139 166 118 ...  
## $ bp : int 72 66 64 66 40 50 74 80 72 84 ...  
## $ skin : int 35 29 0 23 35 32 0 0 19 47 ...  
## $ Insulin : int 0 0 0 94 168 88 0 0 175 230 ...  
## $ BMI : num 33.6 26.6 23.3 28.1 43.1 31 38 27.1 25.8 45.8 ...  
## $ pedigree: num 0.627 0.351 0.672 0.167 2.288 ...  
## $ age : int 50 31 32 21 33 26 34 57 51 31 ...  
## $ class : Factor w/ 2 levels "0","1": 2 1 2 1 2 2 2 1 2 2 ...

str(diabetesTest)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 192 obs. of 9 variables:  
## $ prgnt : int 5 10 2 8 4 1 7 3 3 11 ...  
## $ glucose : int 116 115 197 125 110 189 100 158 88 138 ...  
## $ bp : int 74 0 70 96 92 60 0 76 58 76 ...  
## $ skin : int 0 0 45 0 0 23 0 36 11 0 ...  
## $ Insulin : int 0 0 543 0 0 846 0 245 54 0 ...  
## $ BMI : num 25.6 35.3 30.5 0 37.6 30.1 30 31.6 24.8 33.2 ...  
## $ pedigree: num 0.201 0.134 0.158 0.232 0.191 0.398 0.484 0.851 0.267 0.42 ...  
## $ age : int 30 29 53 54 30 59 32 28 22 35 ...  
## $ class : Factor w/ 2 levels "0","1": 1 1 2 2 1 2 2 2 1 1 ...

model<- glm(class~prgnt+glucose+bp+skin+Insulin+BMI+pedigree+age, data = diabetesTrain, family = binomial)  
model

##   
## Call: glm(formula = class ~ prgnt + glucose + bp + skin + Insulin +   
## BMI + pedigree + age, family = binomial, data = diabetesTrain)  
##   
## Coefficients:  
## (Intercept) prgnt glucose bp skin   
## -8.423234 0.136613 0.039187 -0.010925 0.001172   
## Insulin BMI pedigree age   
## -0.001697 0.084260 0.827628 0.001463   
##   
## Degrees of Freedom: 575 Total (i.e. Null); 567 Residual  
## Null Deviance: 745.1   
## Residual Deviance: 540 AIC: 558

summary(model)

##   
## Call:  
## glm(formula = class ~ prgnt + glucose + bp + skin + Insulin +   
## BMI + pedigree + age, family = binomial, data = diabetesTrain)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4409 -0.7368 -0.4139 0.7204 2.9541   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -8.423234 0.827323 -10.181 < 2e-16 \*\*\*  
## prgnt 0.136613 0.038832 3.518 0.000435 \*\*\*  
## glucose 0.039187 0.004508 8.694 < 2e-16 \*\*\*  
## bp -0.010925 0.006057 -1.804 0.071265 .   
## skin 0.001172 0.008227 0.142 0.886760   
## Insulin -0.001697 0.001087 -1.561 0.118570   
## BMI 0.084260 0.017394 4.844 1.27e-06 \*\*\*  
## pedigree 0.827628 0.339732 2.436 0.014846 \*   
## age 0.001463 0.011098 0.132 0.895085   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 745.11 on 575 degrees of freedom  
## Residual deviance: 540.00 on 567 degrees of freedom  
## AIC: 558  
##   
## Number of Fisher Scoring iterations: 5

predictTrain<- predict(model, type = "response")  
head(predictTrain,3)

## 1 2 3   
## 0.70552024 0.04454686 0.84656517

cm<-table(diabetesTrain$class, predictTrain>= 0.58)  
cm

##   
## FALSE TRUE  
## 0 345 30  
## 1 95 106

#this model can 77% times predict as 1 which is True positive rate(106/(106+30))  
accuracy <- sum(diag(cm))/sum(cm)  
accuracy #78% is good or not.

## [1] 0.7829861

#lets check this model can only give you 48% confidence of person not havic diabetic hence accuracy of   
#predicted model more than this accepted.  
table(diabetesTrain$class)

##   
## 0 1   
## 375 201

error<-1-accuracy  
error

## [1] 0.2170139

#model performance Evaluation  
library(ROCR)

## Warning: package 'ROCR' was built under R version 3.4.1

## Loading required package: gplots

## Warning: package 'gplots' was built under R version 3.4.1

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

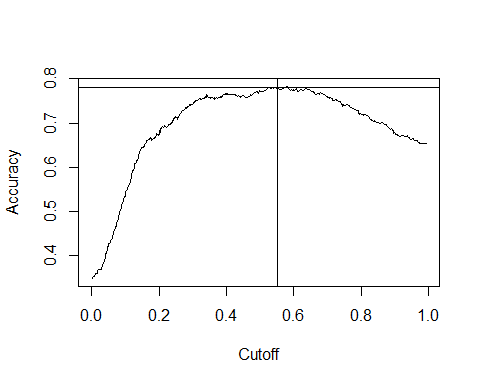
predmod<-predict(model,diabetesTrain,type = "response")  
head(predmod)

## 1 2 3 4 5 6   
## 0.70552024 0.04454686 0.84656517 0.04245449 0.86269300 0.05955055

head(diabetesTrain)

## # A tibble: 6 x 9  
## prgnt glucose bp skin Insulin BMI pedigree age class  
## <int> <int> <int> <int> <int> <dbl> <dbl> <int> <fctr>  
## 1 6 148 72 35 0 33.6 0.627 50 1  
## 2 1 85 66 29 0 26.6 0.351 31 0  
## 3 8 183 64 0 0 23.3 0.672 32 1  
## 4 1 89 66 23 94 28.1 0.167 21 0  
## 5 0 137 40 35 168 43.1 2.288 33 1  
## 6 3 78 50 32 88 31.0 0.248 26 1

#histogram suggest that maiximum data is between 0 to 0.5 hence changing threshold value might not heavily impact on accuracy.  
predmodel<-prediction(predmod,diabetesTrain$class)  
eval<-performance(predmodel,"acc")  
plot(eval)  
abline(h=0.78,v=0.55)



#if you look graph there is almost no change in accuracy and threshold.  
#Identify Best Values  
max<-which.max(slot(eval,"y.values")[[1]])  
accu<-slot(eval,"y.values")[[1]][max]  
cut<-slot(eval,"x.values")[[1]][max]  
print(c(accuracy=accu,Cutoff=cut))

## accuracy Cutoff.565   
## 0.7829861 0.5810861

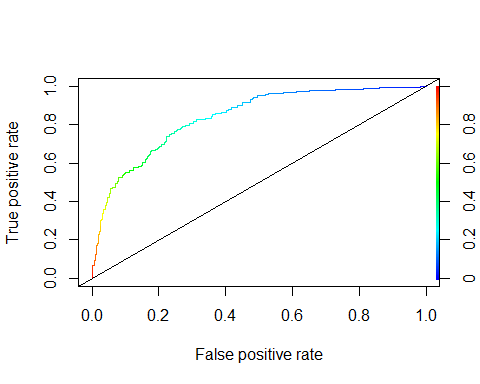
#Above is best cutoff where maximum accuracy of model can be achived.  
#Receiver operating characteristics  
summary(predmod)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.001416 0.117816 0.268457 0.348958 0.533577 0.992406

summary (predmodel)

## Length Class Mode   
## 1 prediction S4

ROC<-performance(predmodel,"tpr","fpr")  
plot(ROC,colorize=T)  
abline(a=0,b=1)



test\_results<- predict(model,diabetesTest,type = "response")  
head(test\_results, 3) # 0.163 0.64 0.67

## 1 2 3   
## 0.1630588 0.6404027 0.6714463

cmt<-table(diabetesTest$class,test\_results >= 0.5)  
cmt

##   
## FALSE TRUE  
## 0 113 12  
## 1 29 38

accuracy <- sum(diag(cmt))/sum(cmt)  
accuracy

## [1] 0.7864583

error<-1-accuracy  
error

## [1] 0.2135417

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.