Statistical_Learning

Marios Choudetsanakis

2/3/2022

Data set

The goal of the project is to predict if an individual of the data set will have diabetes in the future.The data set contains 8 continuous metrics and 1 factor (positive or negative).The individuals are women and the prediction will be achieved through classification using 3 different methods (lda,qda,knn).

```
data.set=read.table("data.txt",sep = ',')
names.of.data=c("n.pregnant","glucose","b.p","s.t","inslunin",'b.m.index',"diabetes","age","class")
colnames(data.set)=names.of.data
data.A=data.set
data.B=data.set
#data set B without the missing values
data.B["glucose"][data.B["glucose"]==0]=NA
data.B["s.t"][data.B["s.t"]==0]=NA
data.B["b.p"][data.B["b.p"]==0]=NA
data.B["inslunin"][data.B["inslunin"]==0]=NA
data.B["b.m.index"][data.B["b.m.index"]==0]=NA
summary(data.B)
```

```
##    n.pregnant        glucose           b.p              s.t
##  Min.   : 0.000   Min.   : 44.0   Min.   : 24.00   Min.   : 7.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.00   1st Qu.:22.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :29.00
##  Mean   : 3.845   Mean   :121.7   Mean   : 72.41   Mean   :29.15
##  3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.00   3rd Qu.:36.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##                   NA’s   :5       NA’s   :35       NA’s   :227
##     inslunin        b.m.index        diabetes           age
##  Min.   : 14.00   Min.   :18.20   Min.   :0.0780   Min.   :21.00
##  1st Qu.: 76.25   1st Qu.:27.50   1st Qu.:0.2437   1st Qu.:24.00
##  Median :125.00   Median :32.30   Median :0.3725   Median :29.00
##  Mean   :155.55   Mean   :32.46   Mean   :0.4719   Mean   :33.24
##  3rd Qu.:190.00   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
##  Max.   :846.00   Max.   :67.10   Max.   :2.4200   Max.   :81.00
##  NA’s   :374      NA’s   :11
##      class
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
##
```

```
df=data.B[!(is.na(data.B$glucose) | is.na(data.B$b.p) | is.na(data.B$s.t) | is.na(data.B$inslunin) |
              is.na(data.B$b.m.index)),]
data.B=df
```

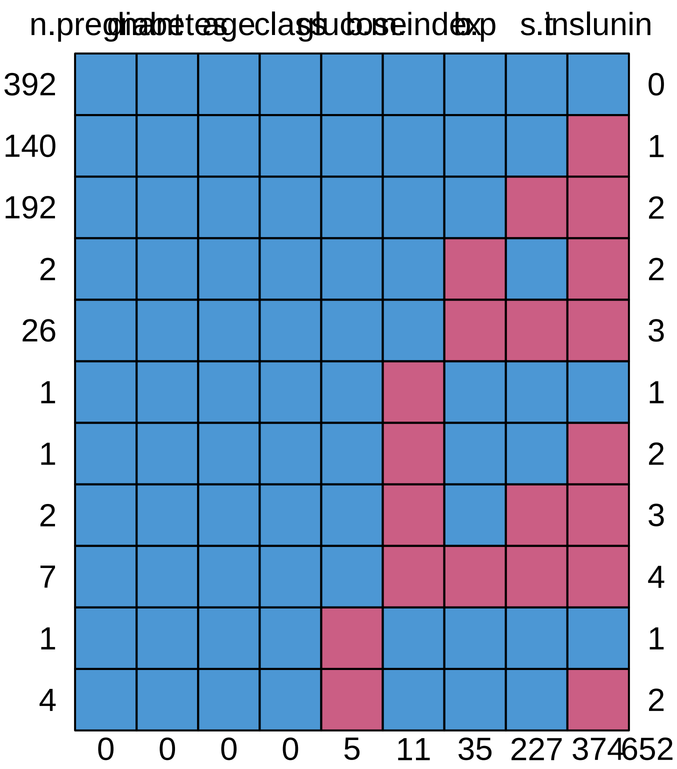
Data Imputation(data set A)

Data imputation for the data set A with 3 different methods

- PMM(predictive mean matching)
- Lasso linear regression
- Linear regression using bootstrap

Then, I compare the results of each method with the descriptive statistics of the original data.The Lasso linear regression gave values which are close to the values of the original data.

```
#install.packages("mice")
library(mice)
#convert the zeros to NA
data.A["glucose"][data.A["glucose"]==0]=NA
data.A["s.t"][data.A["s.t"]==0]=NA
data.A["b.p"][data.A["b.p"]==0]=NA
data.A["inslunin"][data.A["inslunin"]==0]=NA
data.A["b.m.index"][data.A["b.m.index"]==0]=NA
data.A["diabetes"][data.A["diabetes"]==0]=NA
summary(data.A)
md.pattern(data.A)
```



```
#Predictive mean matching
data.imp.1=mice(data.A,method = "pmm",maxit = 20)
f.data.A.1=complete(data.imp.1,3)
#Lasso linear regression
data.imp.2=mice(data.A,method = "lasso.norm",maxit = 20)
f.data.A.2=complete(data.imp.2,3)
#Linear regression using bootstrap
data.imp.3=mice(data.A,method = "norm.boot",maxit = 20)
f.data.A.3=complete(data.imp.3,3)
```

```
summary(f.data.A.1)
```

```
##    n.pregnant        glucose           b.p              s.t
##  Min.   : 0.000   Min.   : 44.0   Min.   : 24.0   Min.   : 7.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.0   1st Qu.:20.00
##  Median : 3.000   Median :117.0   Median : 72.0   Median :28.00
##  Mean   : 3.845   Mean   :121.6   Mean   : 72.3   Mean   :28.64
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.0   3rd Qu.:36.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.0   Max.   :99.00
##     inslunin        b.m.index        diabetes           age
##  Min.   : 14.0   Min.   :18.20   Min.   :0.0780   Min.   :21.00
##  1st Qu.: 76.0   1st Qu.:27.40   1st Qu.:0.2437   1st Qu.:24.00
##  Median :122.0   Median :32.30   Median :0.3725   Median :29.00
##  Mean   :151.5   Mean   :32.44   Mean   :0.4719   Mean   :33.24
##  3rd Qu.:182.2   3rd Qu.:36.62   3rd Qu.:0.6262   3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00
##      class
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
```

```
summary(f.data.A.2)
```

```
##    n.pregnant       glucose          b.p             s.t
##  Min.   : 0.000   Min.   : 44.0   Min.   : 24.00   Min.   : 6.4
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.00   1st Qu.:22.0
##  Median : 3.000   Median :117.0   Median : 72.00   Median :29.0
##  Mean   : 3.845   Mean   :121.7   Mean   : 72.36   Mean   :29.2
##  3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.00   3rd Qu.:36.0
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.0
##     inslunin        b.m.index        diabetes            age
##  Min.   :-145.4   Min.   :18.20   Min.   :0.0780   Min.   :21.00
##  1st Qu.:  76.0   1st Qu.:27.48   1st Qu.:0.2437   1st Qu.:24.00
##  Median : 138.2   Median :32.29   Median :0.3725   Median :29.00
##  Mean   : 156.2   Mean   :32.42   Mean   :0.4719   Mean   :33.24
##  3rd Qu.: 210.0   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
##  Max.   : 846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00
##      class
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
```

```
summary(f.data.A.3)
```

```
##    n.pregnant       glucose          b.p            s.t
##  Min.   : 0.000   Min.   : 44.0   Min.   : 24.0   Min.   : 1.308
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.0   1st Qu.:20.988
##  Median : 3.000   Median :117.0   Median : 72.0   Median :28.288
##  Mean   : 3.845   Mean   :121.7   Mean   : 72.2   Mean   :28.571
##  3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.0   3rd Qu.:35.703
##  Max.   :17.000   Max.   :199.0   Max.   :122.0   Max.   :99.000
##     inslunin         b.m.index        diabetes            age
##  Min.   :-181.16   Min.   :18.20   Min.   :0.0780   Min.   :21.00
##  1st Qu.:  74.75   1st Qu.:27.48   1st Qu.:0.2437   1st Qu.:24.00
##  Median : 130.00   Median :32.30   Median :0.3725   Median :29.00
##  Mean   : 152.28   Mean   :32.43   Mean   :0.4719   Mean   :33.24
##  3rd Qu.: 215.00   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
##  Max.   : 846.00   Max.   :67.10   Max.   :2.4200   Max.   :81.00
##      class
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
```

```
summary(data.A)
```

```
##    n.pregnant        glucose           b.p              s.t
##  Min.   : 0.000   Min.   : 44.0   Min.   : 24.00   Min.   : 7.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.00   1st Qu.:22.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :29.00
##  Mean   : 3.845   Mean   :121.7   Mean   : 72.41   Mean   :29.15
##  3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.00   3rd Qu.:36.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##                   NA's   :5       NA's   :35       NA's   :227
##     inslunin        b.m.index        diabetes           age
##  Min.   : 14.00   Min.   :18.20   Min.   :0.0780   Min.   :21.00
##  1st Qu.: 76.25   1st Qu.:27.50   1st Qu.:0.2437   1st Qu.:24.00
##  Median :125.00   Median :32.30   Median :0.3725   Median :29.00
##  Mean   :155.55   Mean   :32.46   Mean   :0.4719   Mean   :33.24
##  3rd Qu.:190.00   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
##  Max.   :846.00   Max.   :67.10   Max.   :2.4200   Max.   :81.00
##  NA's   :374      NA's   :11
##      class
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
##
```

```
imp.data.A=f.data.A.2
```

Statistical Inference(data set A)

```
model=glm(class ~ .,data = imp.data.A,family = binomial)
summary(model)
```

```
##
## Call:
## glm(formula = class ~ ., family = binomial, data = imp.data.A)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7614  -0.7165  -0.3877   0.7120   2.3593
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.0348124  0.8124308 -11.121  < 2e-16 ***
## n.pregnant   0.1193150  0.0324404   3.678 0.000235 ***
## glucose      0.0374781  0.0041913   8.942  < 2e-16 ***
## b.p         -0.0084647  0.0085146  -0.994 0.320154
## s.t          0.0215242  0.0119830   1.796 0.072457 .
## inslunin    -0.0004367  0.0009333  -0.468 0.639834
## b.m.index    0.0729276  0.0183532   3.974 7.08e-05 ***
## diabetes     0.8433998  0.2978005   2.832 0.004624 **
## age          0.0129713  0.0096004   1.351 0.176657
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 993.48  on 767  degrees of freedom
## Residual deviance: 710.19  on 759  degrees of freedom
## AIC: 728.19
##
## Number of Fisher Scoring iterations: 5
```

The summary of the model shows that the number of times pregnant(n.pregnant),the Plasma glucose concentration a 2 hours in an oral glucose tolerance test(glucose),the Body mass index(b.m.index) and the Diabetes pedigree function(diabetes) give p-values lower than 0.001,so they are statistically significant.The analysis will continue without the non-statistically significant variables.

```
f.model=glm(class ~ n.pregnant + glucose + b.m.index + diabetes,data = imp.data.A,family=binomial)
summary(f.model)
```

```
##
## Call:
## glm(formula = class ~ n.pregnant + glucose + b.m.index + diabetes,
##     family = binomial, data = imp.data.A)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8158  -0.7273  -0.4031   0.7198   2.4351
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.090653   0.698516 -13.014  < 2e-16 ***
## n.pregnant   0.141881   0.027495   5.160 2.47e-07 ***
## glucose      0.037045   0.003485  10.631  < 2e-16 ***
## b.m.index    0.085380   0.014595   5.850 4.92e-09 ***
## diabetes     0.874675   0.294338   2.972  0.00296 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 993.48  on 767  degrees of freedom
## Residual deviance: 715.97  on 763  degrees of freedom
## AIC: 725.97
##
## Number of Fisher Scoring iterations: 5
```

```
coef(f.model)
```

```
## (Intercept)  n.pregnant     glucose   b.m.index    diabetes 
## -9.09065302  0.14188075  0.03704502  0.08537962  0.87467524
```

Data Exploration(data set A)

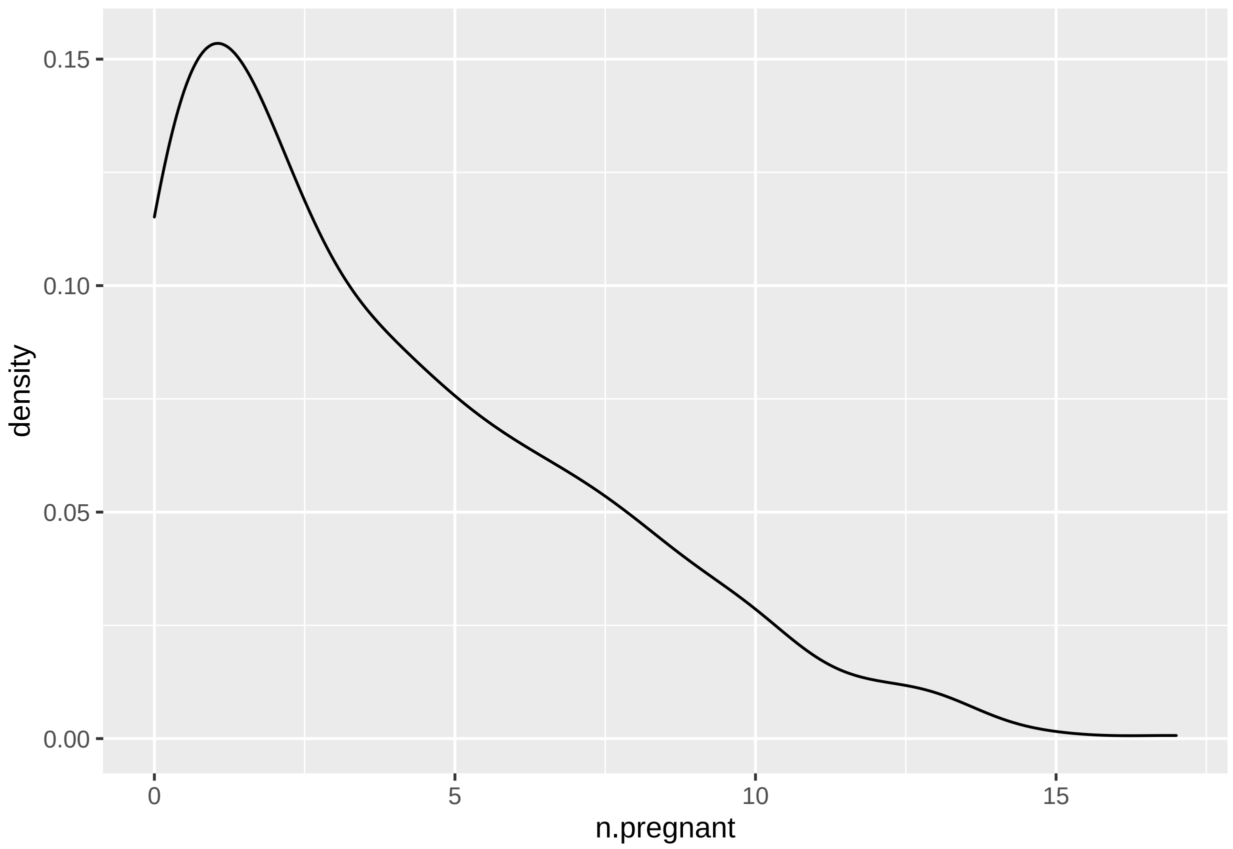
```
#install.packages("ggplot2")
#install.packages("dplyr")
library(ggplot2)
library(dplyr)
```

```
## 
## Attaching package: 'dplyr'
```

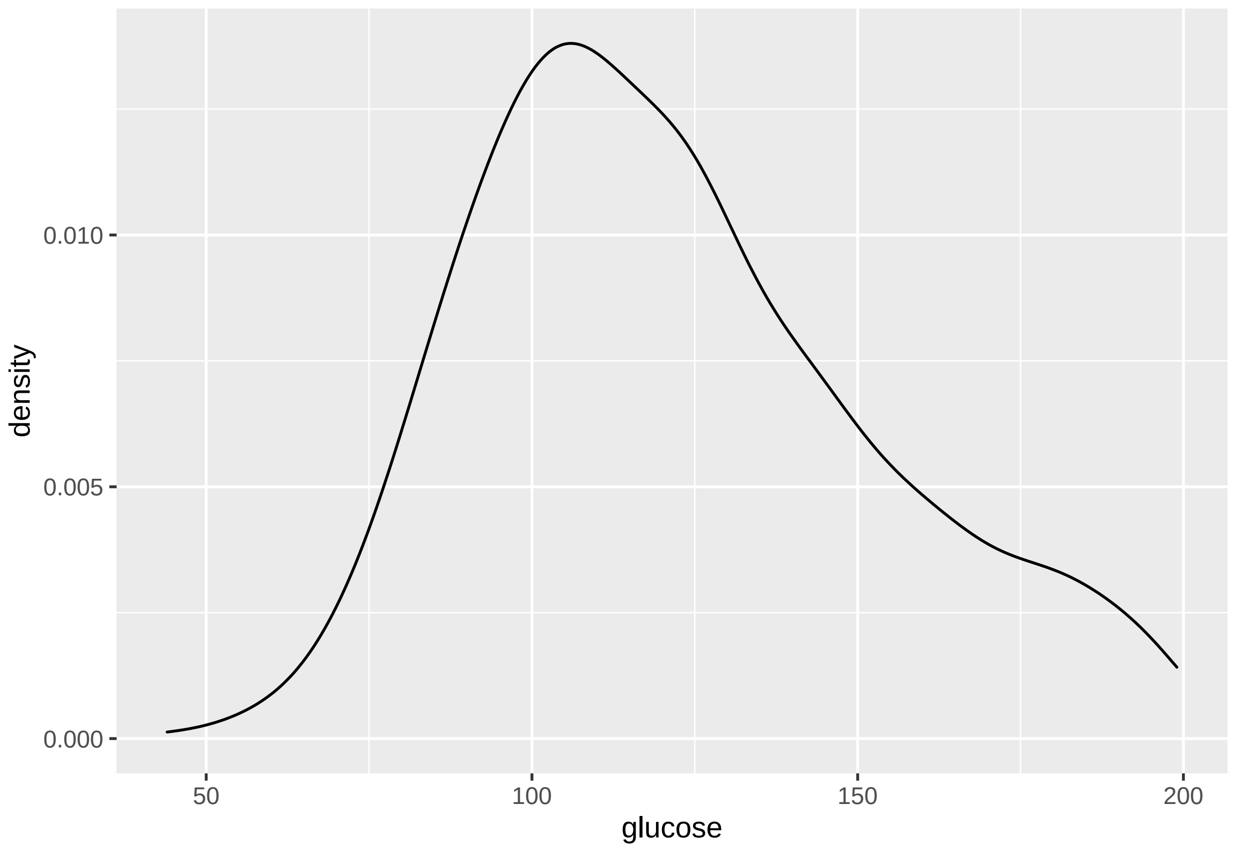
```
## The following objects are masked from 'package:stats':
## 
##     filter, lag
```

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

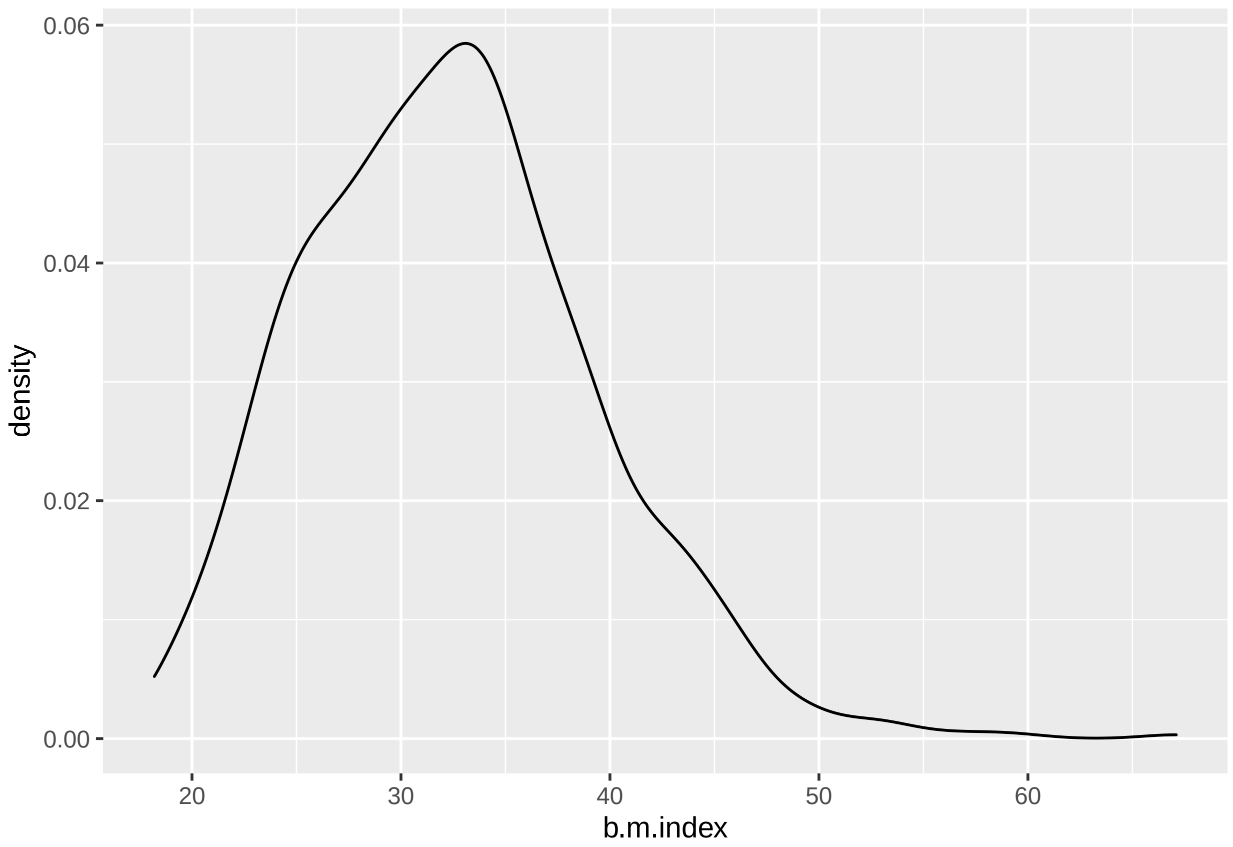
```
#data visualization
imp.data.A%>%ggplot(aes(n.pregnant))+geom_density()
```



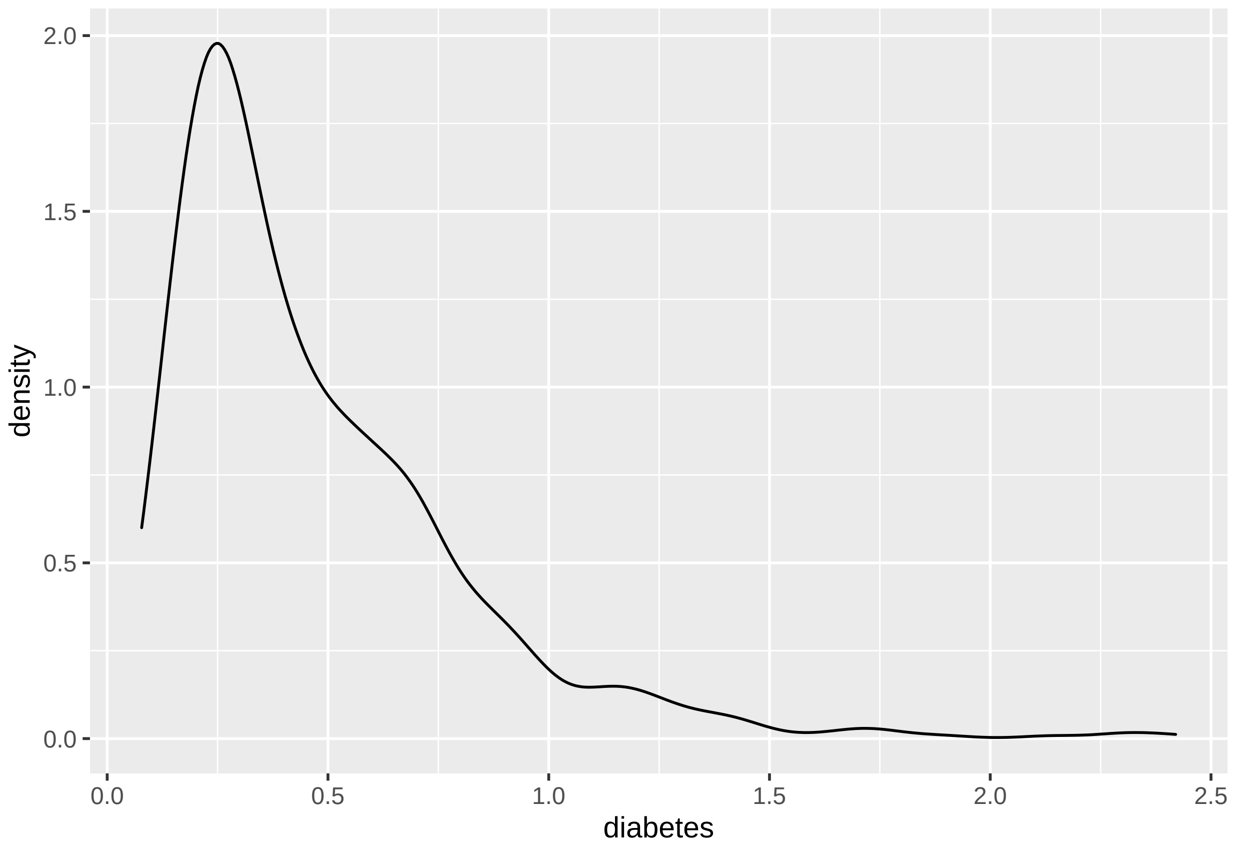
```
imp.data.A%>%ggplot(aes(glucose))+geom_density()
```



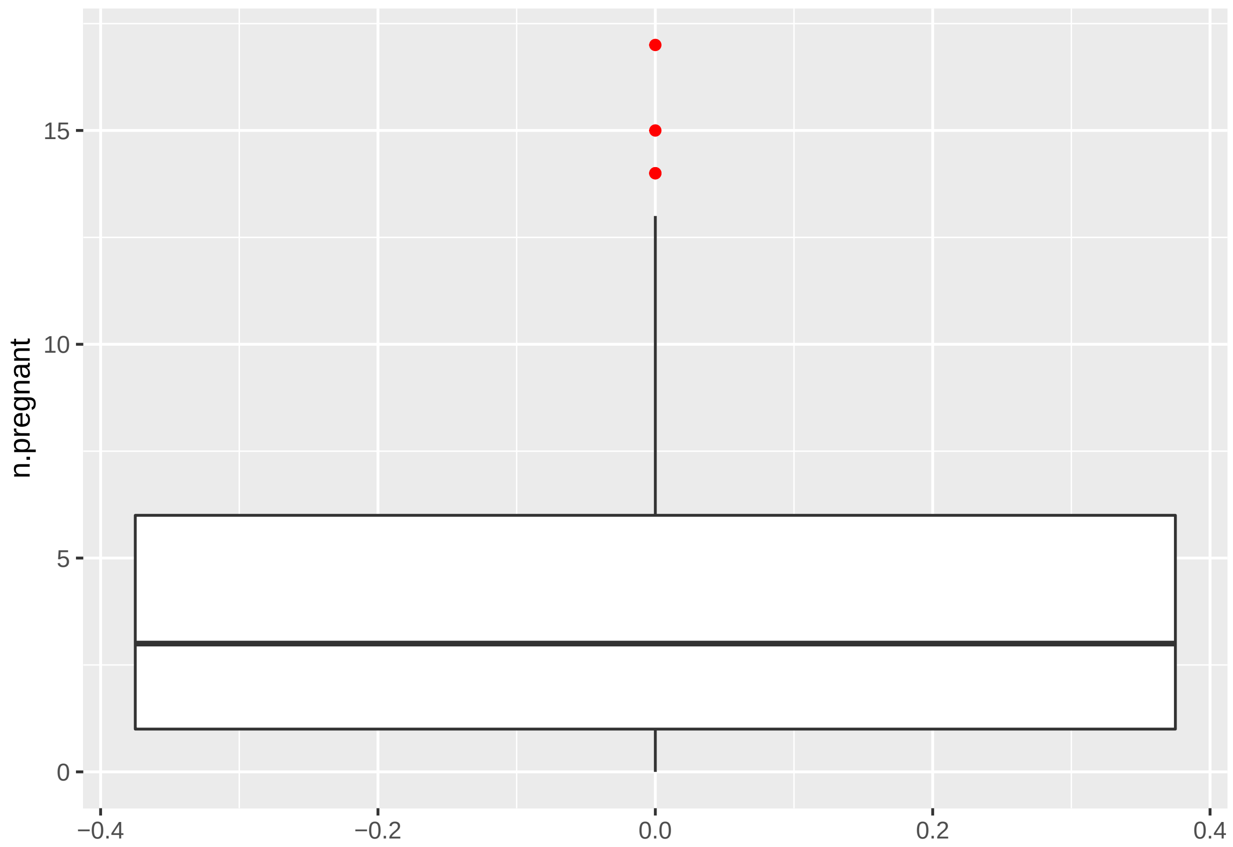
```
imp.data.A%>%ggplot(aes(b.m.index))+geom_density()
```



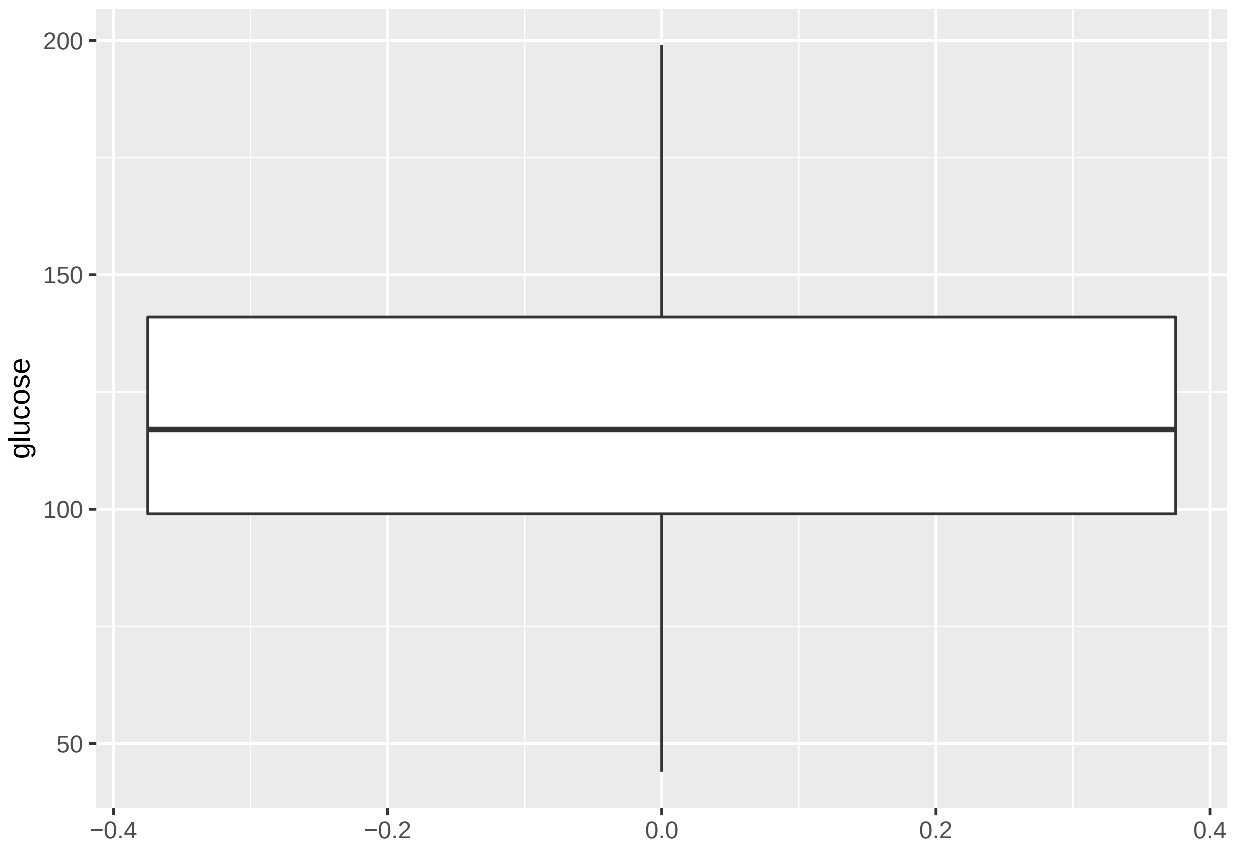
```
imp.data.A%>%ggplot(aes(diabetes))+geom_density()
```



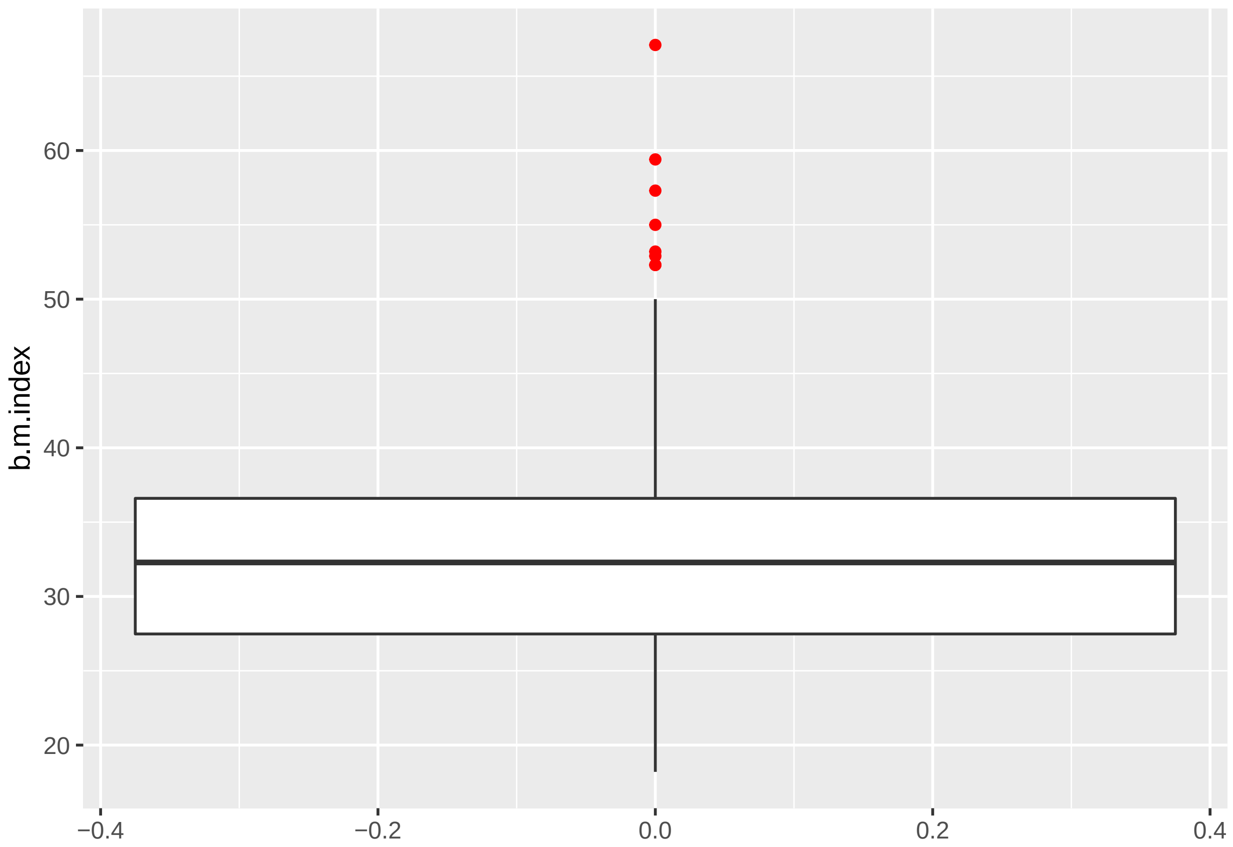
```
#The variable of the number of the times pregnant has 3 outliers
imp.data.A%>%ggplot(aes(y=n.pregnant))+geom_boxplot(outlier.colour = "red")
```



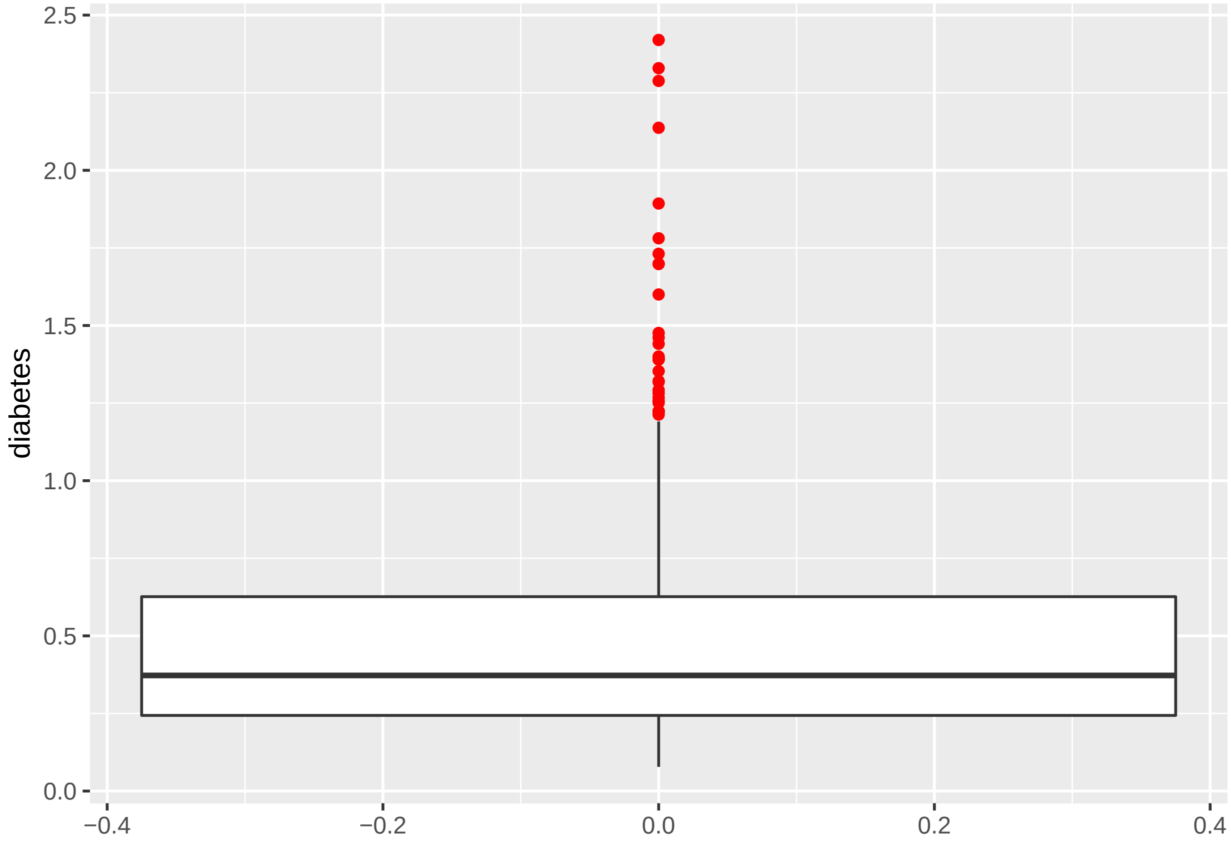
```
#the variable of the glucose has no outliers
imp.data.A%>%ggplot(aes(y=glucose))+geom_boxplot(outlier.colour = "red")
```



```
#the body mass index has 7 outliers
imp.data.A%>%ggplot(aes(y=b.m.index))+geom_boxplot(outlier.colour = "red")
```



```
#the diabetes has many outliers
imp.data.A%>%ggplot(aes(y=diabetes))+geom_boxplot(outlier.colour = "red")
```



```
#Outliers' removing with interquartile method
#diabetes
Q1 <- quantile(imp.data.A$diabetes, .25)
Q3 <- quantile(imp.data.A$diabetes, .75)
IQR <- IQR(imp.data.A$diabetes)
nod <- subset(imp.data.A, imp.data.A$diabetes> (Q1 - 1.5*IQR) & imp.data.A$diabetes< (Q3 + 1.5*IQR))
Q1 <- quantile(imp.data.A$b.m.index, .25)
Q3 <- quantile(imp.data.A$b.m.index, .75)
IQR <- IQR(imp.data.A$b.m.index)
nod <- subset(imp.data.A, imp.data.A$b.m.index> (Q1 - 1.5*IQR) & imp.data.A$b.m.index< (Q3 + 1.5*IQR))
dim(nod)
```

```
## [1] 760   9
```

```
dim(imp.data.A)
```

```
## [1] 768   9
```

```
#Normallity test
data.a=nod
shapiro.test(data.a$n.pregnant)
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  data.a$n.pregnant
## W = 0.90516, p-value < 2.2e-16
```

```
shapiro.test(data.a$glucose)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data.a$glucose
## W = 0.97018, p-value = 2.47e-11
```

```
shapiro.test(data.a$b.m.index)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data.a$b.m.index
## W = 0.99007, p-value = 5.226e-05
```

```
shapiro.test(data.a$diabetes)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data.a$diabetes
## W = 0.84615, p-value < 2.2e-16
```

```
#log transformation
shapiro.test(log(data.a$n.pregnant+1))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log(data.a$n.pregnant + 1)
## W = 0.94212, p-value < 2.2e-16
```

```
shapiro.test(log(data.a$glucose))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log(data.a$glucose)
## W = 0.99165, p-value = 0.0002748
```

```
shapiro.test(log(data.a$b.m.index))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log(data.a$b.m.index)
## W = 0.99158, p-value = 0.0002536
```

```
shapiro.test(log(data.a$diabetes))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log(data.a$diabetes)
## W = 0.9929, p-value = 0.001111
```

Discriminant Analysis(data set A)

```
#install.packages("MASS")
#install.packages("caret")
library(MASS)
```

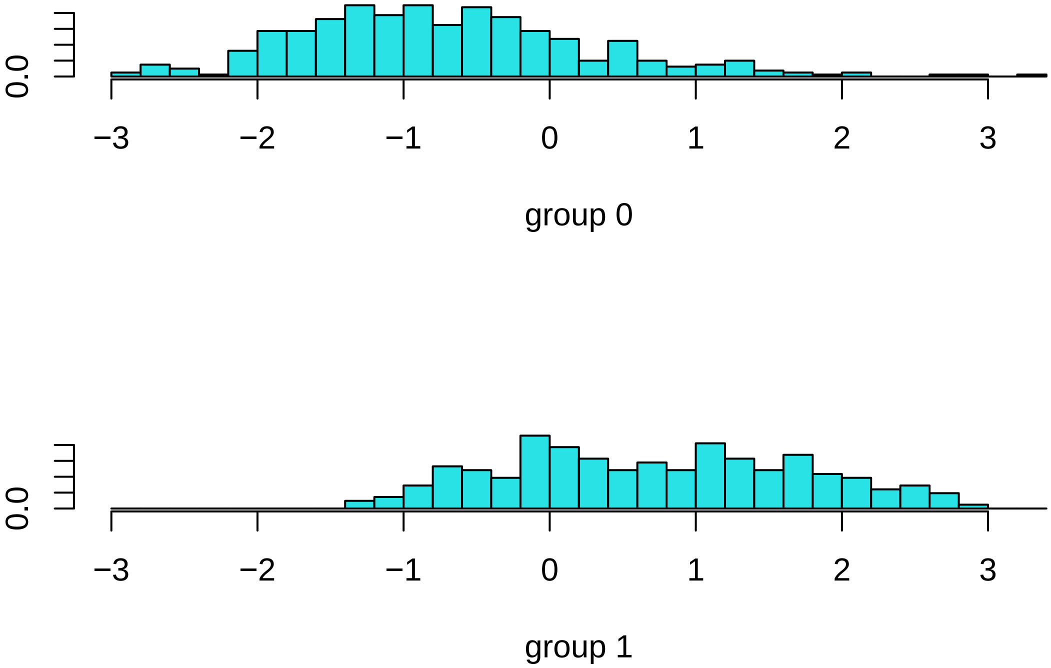
```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
#library(caret)
set.seed(2121)
# Create dummy for splitting (80 - 20)
split_dummy <- sample(c(rep(0, 0.8 * nrow(data.a)),
                        rep(1, 0.2 * nrow(data.a))))
data_train.a <- data.a[split_dummy == 0, ]
data_test.a <- data.a[split_dummy == 1, ]
#Linear Discrimination Analysis
lda.a=lda(class ~ n.pregnant + glucose + b.m.index + diabetes ,data = data_train.a)
lda.a
```

```
## Call:
## lda(class ~ n.pregnant + glucose + b.m.index + diabetes, data = data_train.a)
##
## Prior probabilities of groups:
##         0         1
## 0.6595395 0.3404605
##
## Group means:
##   n.pregnant  glucose b.m.index  diabetes
## 0   3.249377 110.8462  30.77471 0.4300723
## 1   5.120773 142.3760  34.76112 0.5402367
##
## Coefficients of linear discriminants:
##                   LD1
## n.pregnant 0.12733319
## glucose    0.02802660
## b.m.index  0.05712481
## diabetes   0.65010180
```

```
plot(lda.a)
```



```
#prediction with the test data set
lda.a.pred = predict(lda.a,data_test.a,type="response")
names(lda.a.pred)
```

```
## [1] "class"     "posterior" "x"
```

```
lda.a.class=lda.a.pred$class
table(lda.a.class,data_test.a$class)
```

```
##
## lda.a.class  0  1
##           0 92 26
##           1  5 29
```

```
#misclassification error
mle.a=(29+7)/148
mle.a
```

```
## [1] 0.2432432
```

```r
#sensitivity
sensitivity.lda.a = (81+31)/148
sensitivity.lda.a
```

```
## [1] 0.7567568
```

```r
#FNR
fnr.lda.a=30/(30+30)
fnr.lda.a
```

```
## [1] 0.5
```

```r
# Quadratic Discriminant Analysis
qda.a = qda(class ~ n.pregnant + glucose + b.m.index + diabetes ,data = data_train.a)
qda.a
```

```
## Call:
## qda(class ~ n.pregnant + glucose + b.m.index + diabetes, data = data_train.a)
##
## Prior probabilities of groups:
##         0         1
## 0.6595395 0.3404605
##
## Group means:
##   n.pregnant  glucose b.m.index  diabetes
## 0   3.249377 110.8462  30.77471 0.4300723
## 1   5.120773 142.3760  34.76112 0.5402367
```

```r
qda.a.pred = predict(qda.a,data_test.a)
qda.a.class=qda.a.pred$class
table(qda.a.class,data_test.a$class)
```

```
##
## qda.a.class  0  1
##           0 84 23
##           1 13 32
```

```r
#misclassification error
mqe.a=(26+11)/148
mqe.a
```

```
## [1] 0.25
```

```r
#sensitivity
sensitivity.qda.a = (77+34)/148
sensitivity.qda.a
```

```
## [1] 0.75
```

```
#FNR
fnr.qda.a=26/(26+34)
fnr.qda.a
```

```
## [1] 0.4333333
```

```
# K-Nearest Neighbors
library(class)
train.x.a=cbind(data_train.a$n.pregnant,data_train.a$glucose,data_train.a$b.m.index,data_train.a$diabete
test.x.a=cbind(data_test.a$n.pregnant,data_test.a$glucose,data_test.a$b.m.index,data_test.a$diabetes)
knn.a.pred=knn(train.x.a,test.x.a,data_train.a$class,k=3)
table(knn.a.pred,data_test.a$class)
```

```
##
## knn.a.pred  0  1
##          0 75 24
##          1 22 31
```

```
#misclassification error
mke.a=(31+14)/148
mke.a
```

```
## [1] 0.3040541
```

```
#sensitivity
sensitivity.knn.a=(74+29)/148
sensitivity.knn.a
```

```
## [1] 0.6959459
```

```
#FNR
fnr.knn.a=30/(30+30)
fnr.knn.a
```

```
## [1] 0.5
```

Conclusion for data set A

Since we have normality problems our results will not be perfect,but the lda and the qda methods give us 75% sensitivity which is high.On the other hand the K-nearest neighbord gives a smaller sensitivity (69%).

Statistical Inference(data set B)

```
model.b=glm(class ~.,data = data.B,family = binomial)
summary(model.b)
```

```
##
## Call:
## glm(formula = class ~ ., family = binomial, data = data.B)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7823  -0.6603  -0.3642   0.6409   2.5612
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.004e+01  1.218e+00  -8.246  < 2e-16 ***
## n.pregnant   8.216e-02  5.543e-02   1.482  0.13825
## glucose      3.827e-02  5.768e-03   6.635 3.24e-11 ***
## b.p         -1.420e-03  1.183e-02  -0.120  0.90446
## s.t          1.122e-02  1.708e-02   0.657  0.51128
## inslunin    -8.253e-04  1.306e-03  -0.632  0.52757
## b.m.index    7.054e-02  2.734e-02   2.580  0.00989 **
## diabetes     1.141e+00  4.274e-01   2.669  0.00760 **
## age          3.395e-02  1.838e-02   1.847  0.06474 .
## ---
## 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: 344.02  on 383  degrees of freedom
## AIC: 362.02
##
## Number of Fisher Scoring iterations: 5
```

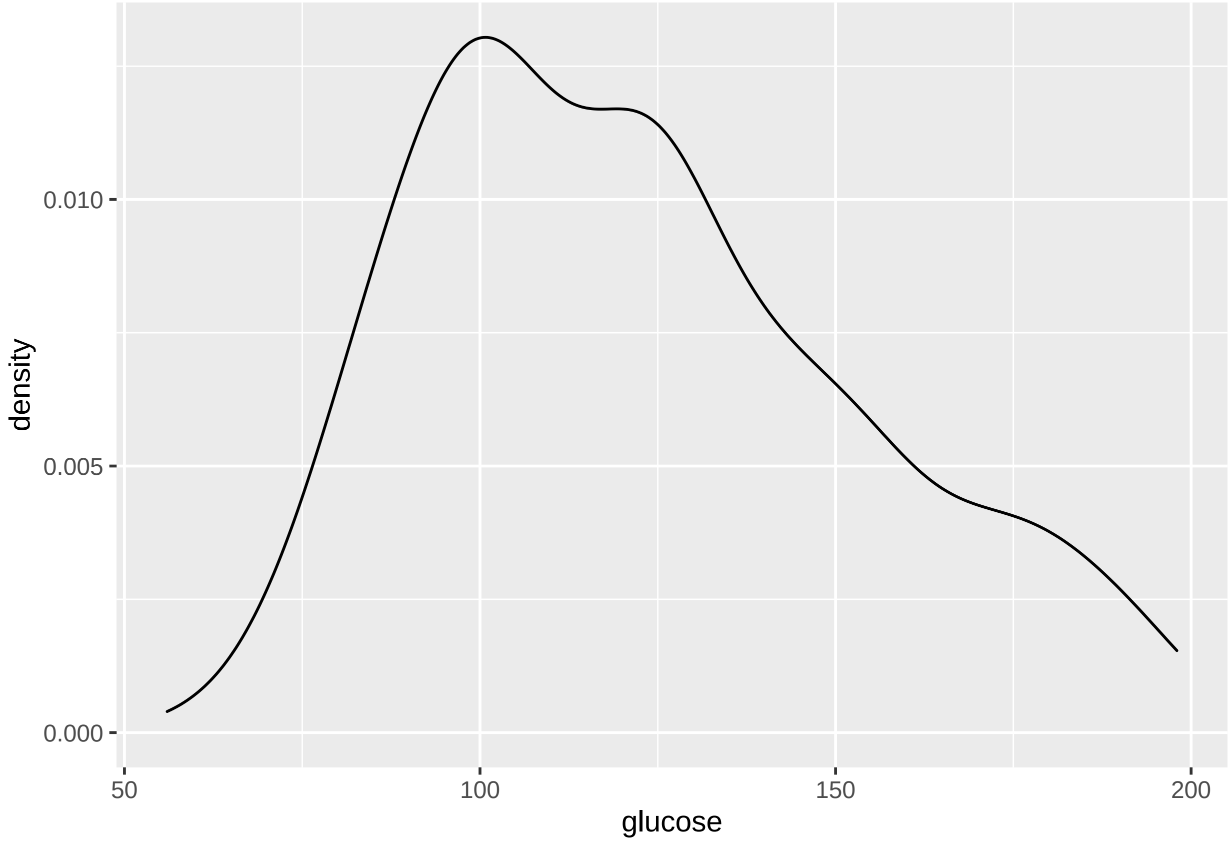
The data set B gives different statistically significant variables (glucose,body mass index ,diabetes,blood pressure and age(p- values close to 0.1)).So the model for the second analysis will be different from the first.

```
f.model.b=glm(class ~ glucose  + b.m.index + diabetes + age,data = data.B,family = binomial)
summary(f.model.b)
```

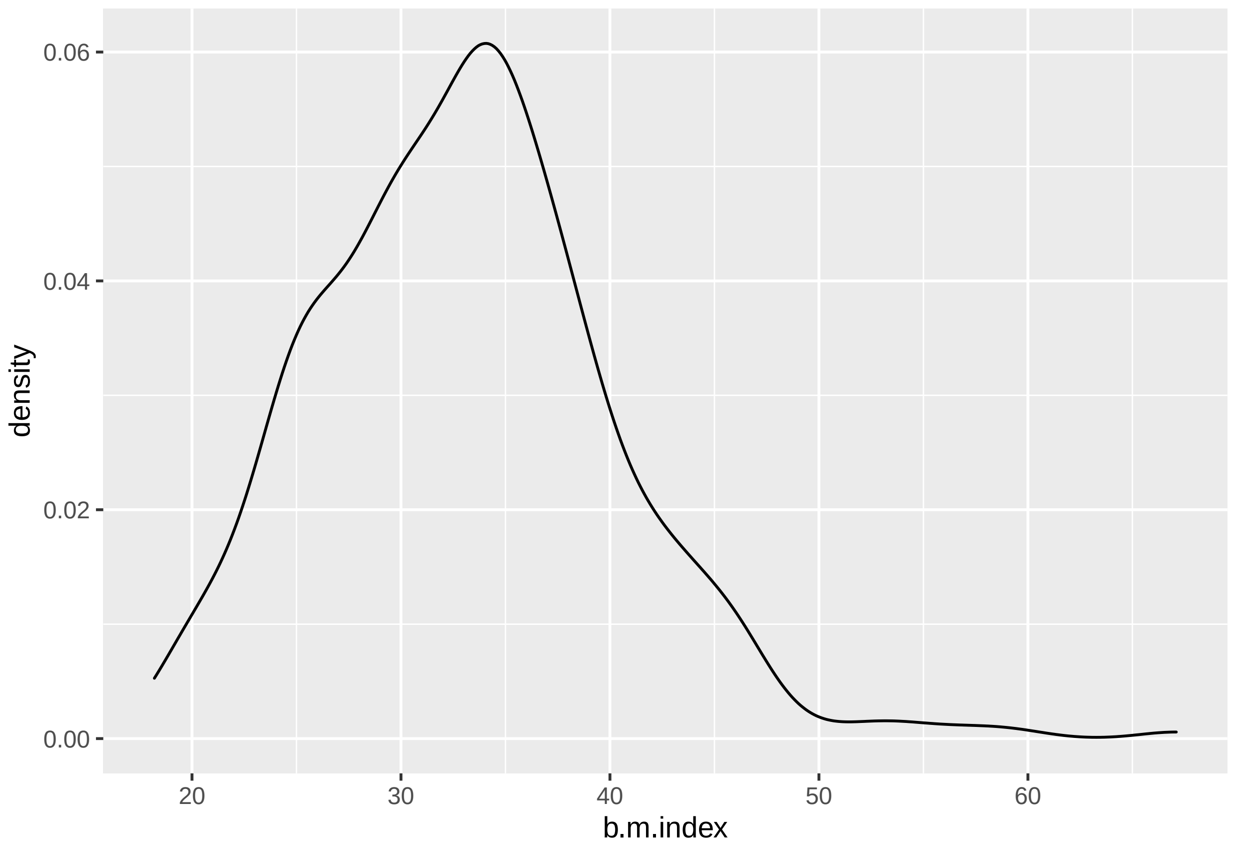
```
##
## Call:
## glm(formula = class ~ glucose + b.m.index + diabetes + age, family = binomial,
##     data = data.B)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8228  -0.6617  -0.3759   0.6702   2.5881
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.092018   1.080251  -9.342  < 2e-16 ***
## glucose       0.036189   0.004982   7.264 3.76e-13 ***
## b.m.index     0.074449   0.020267   3.673 0.000239 ***
## diabetes      1.087129   0.419408   2.592 0.009541 **
## age           0.053012   0.013439   3.945 8.00e-05 ***
## ---
## 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: 347.23  on 387  degrees of freedom
## AIC: 357.23
##
## Number of Fisher Scoring iterations: 5
```

Data Exploration(data set B)

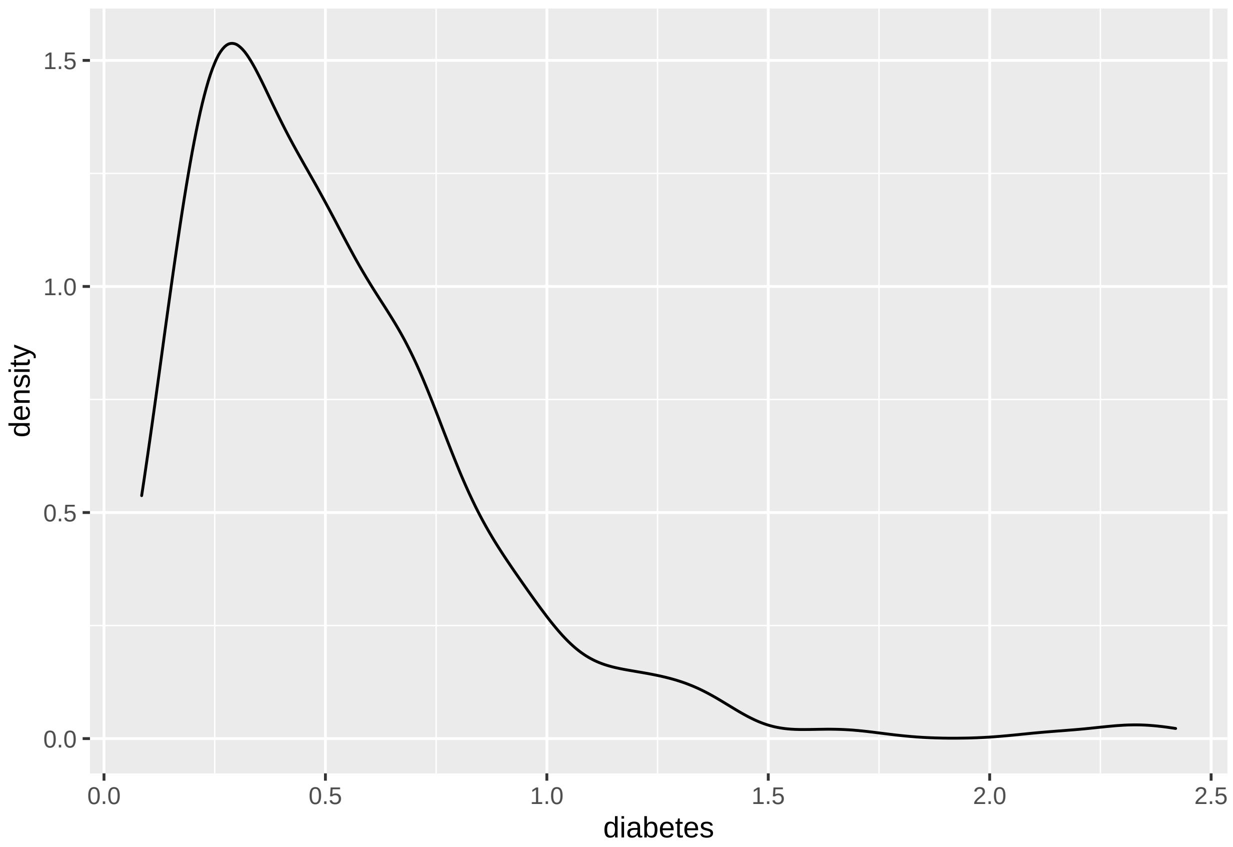
```
library(ggplot2)
library(dplyr)
#data visualization
data.B%>%ggplot(aes(glucose))+geom_density()
```



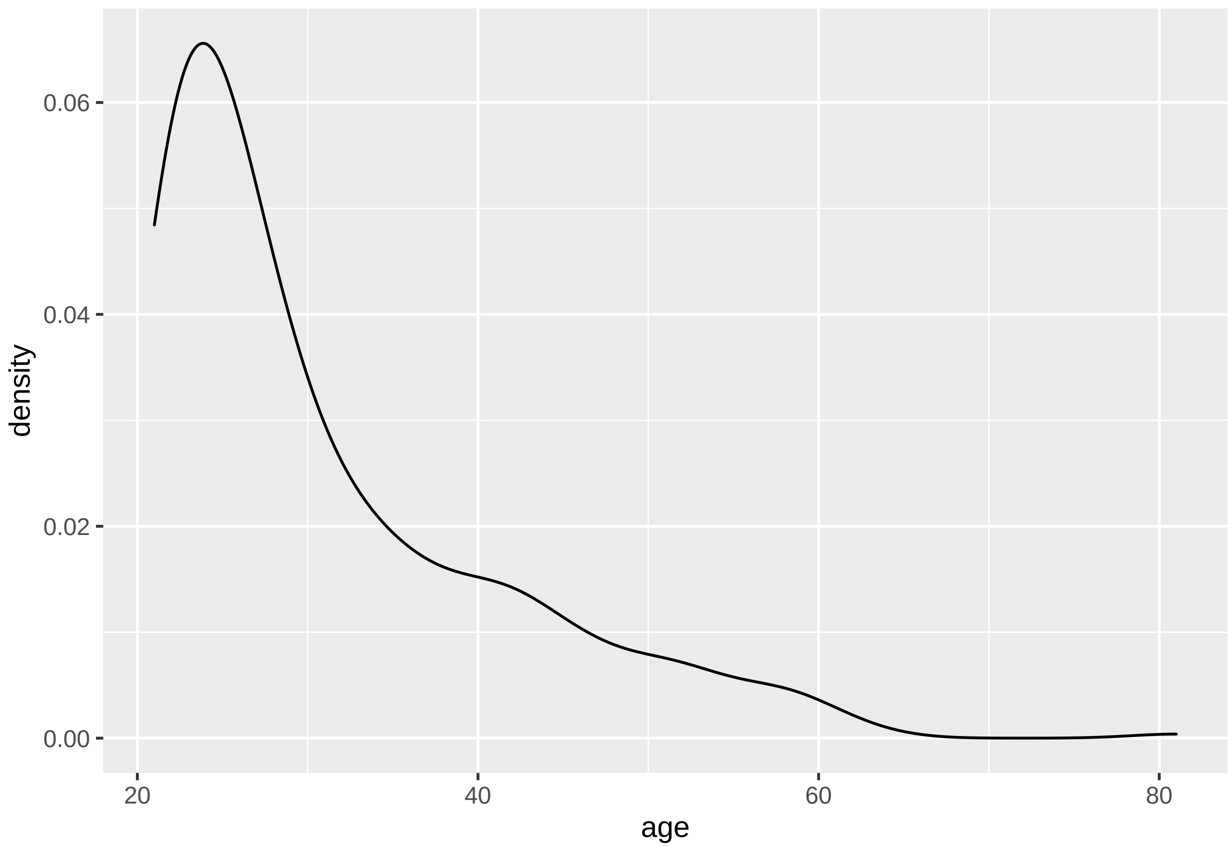
```
data.B%>%ggplot(aes(b.m.index))+geom_density()
```



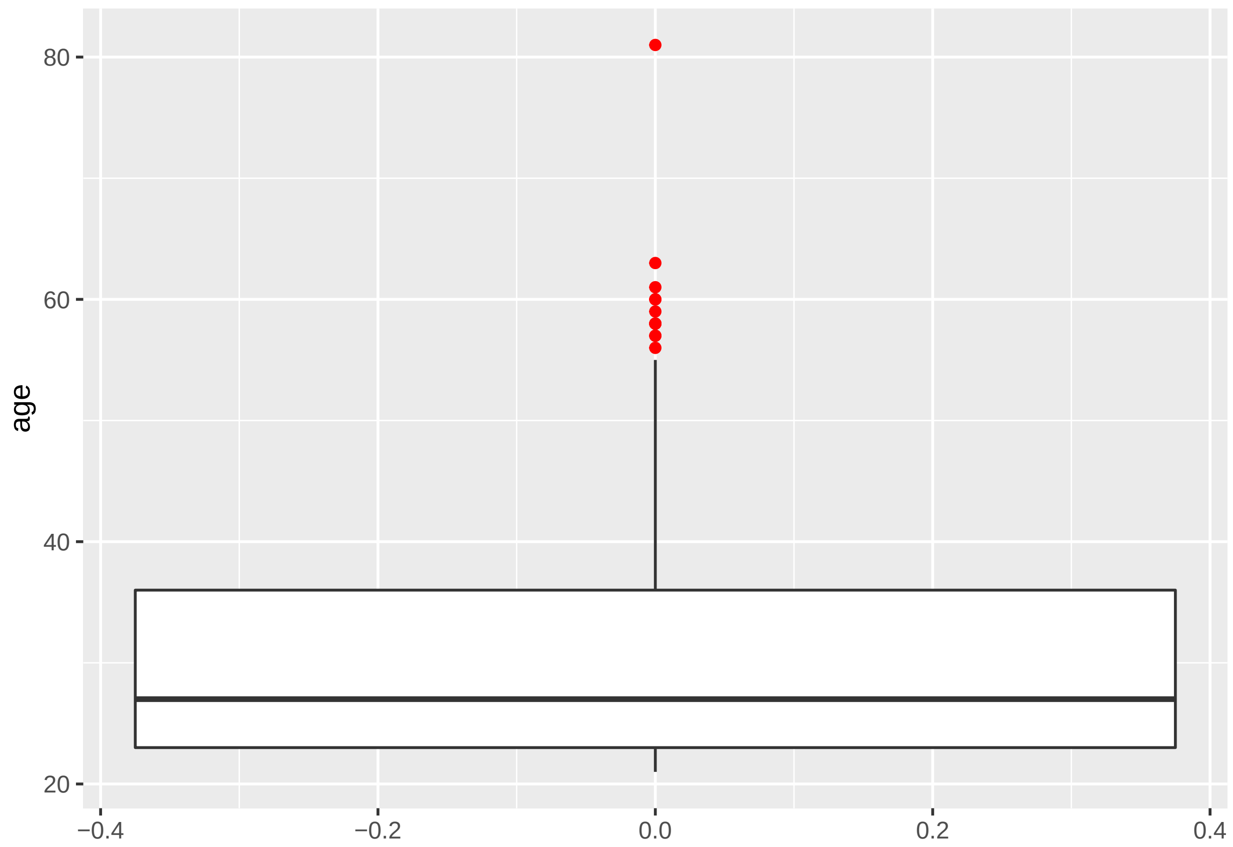
```
data.B%>%ggplot(aes(diabetes))+geom_density()
```



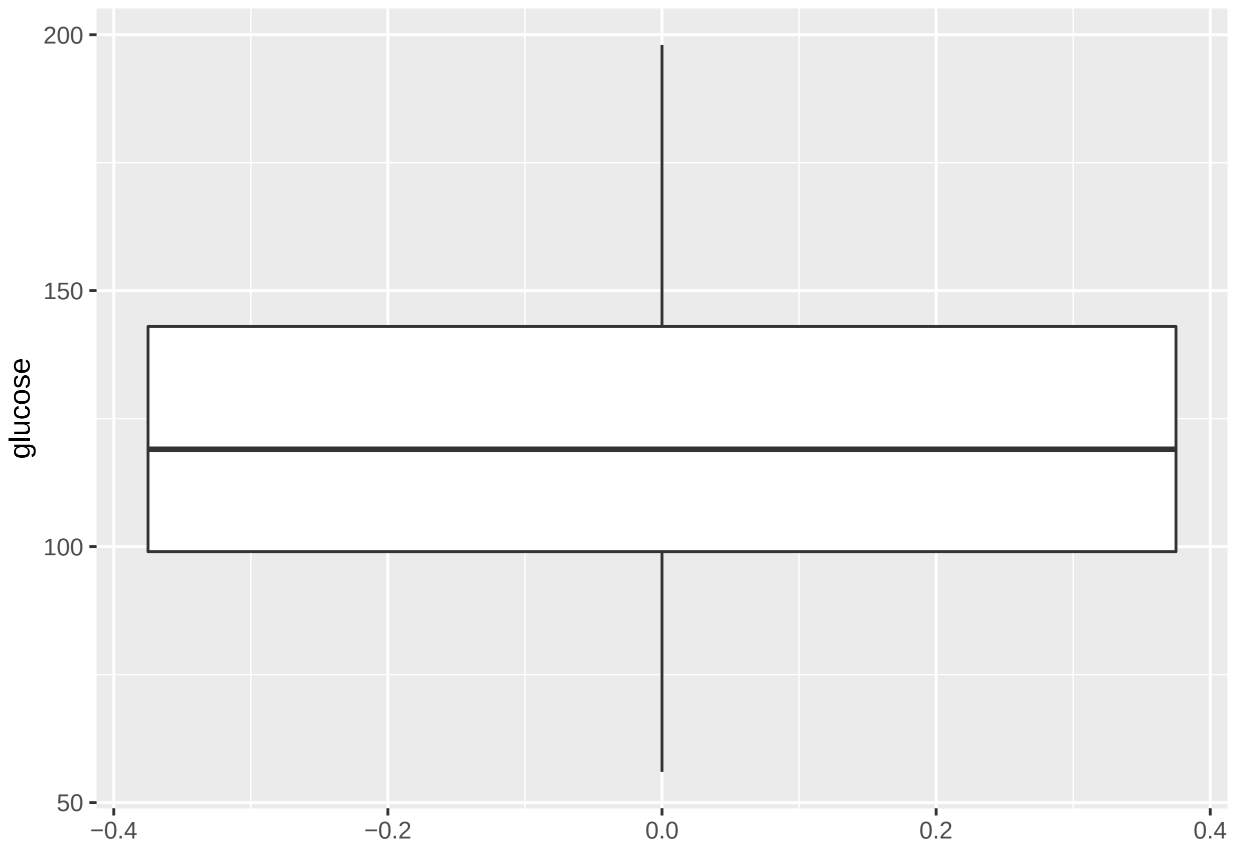
```
data.B%>%ggplot(aes(age))+geom_density()
```



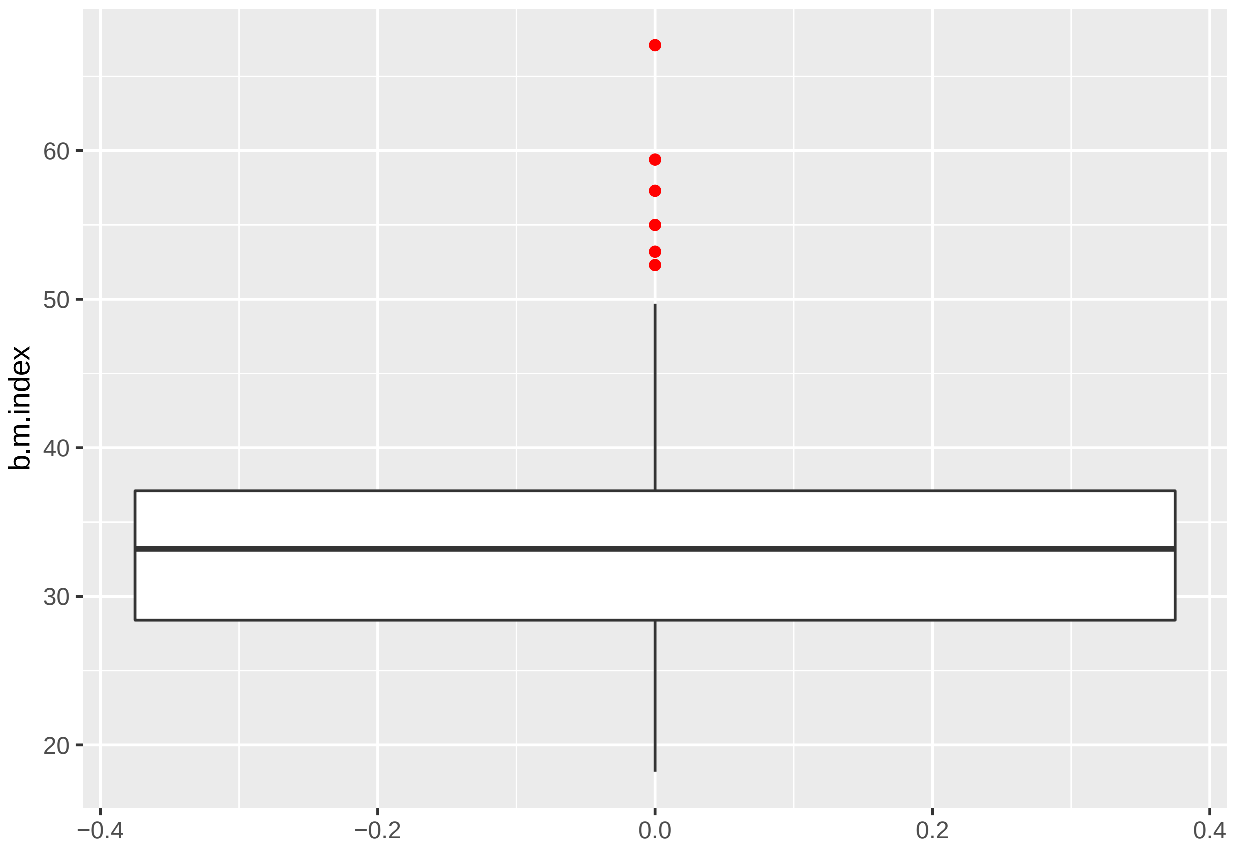
```
#The variable of the age has 8 outliers
data.B%>%ggplot(aes(y=age))+geom_boxplot(outlier.colour = "red")
```



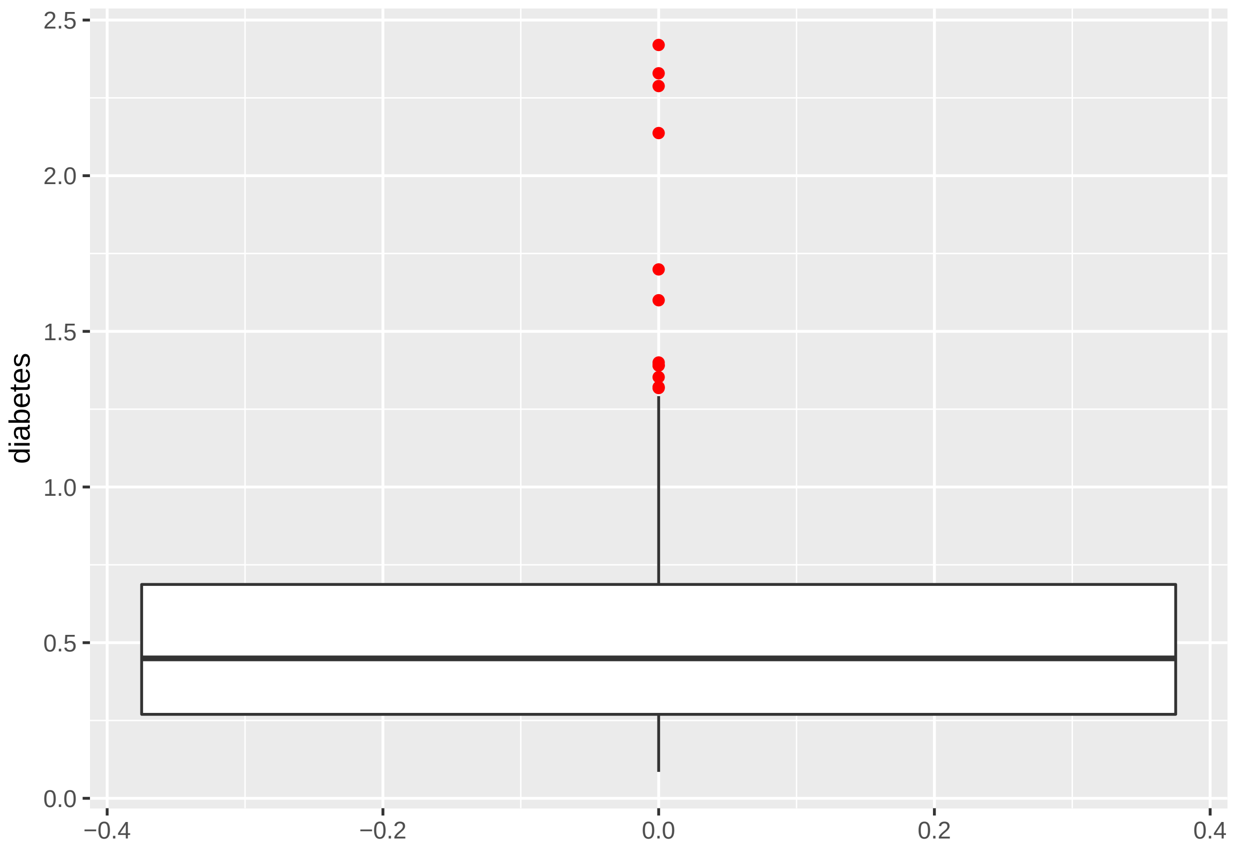
```
#the variable of the glucose has no outliers
data.B%>%ggplot(aes(y=glucose))+geom_boxplot(outlier.colour = "red")
```



```
#the body mass index has 6 outliers
data.B%>%ggplot(aes(y=b.m.index))+geom_boxplot(outlier.colour = "red")
```



```
#the diabetes has 10 outliers
data.B%>%ggplot(aes(y=diabetes))+geom_boxplot(outlier.colour = "red")
```



```
#Outliers' removing with interquartile method
#diabetes
Q1 <- quantile(data.B$diabetes, .25)
Q3 <- quantile(data.B$diabetes, .75)
IQR <- IQR(data.B$diabetes)
nb <- subset(data.B, data.B$diabetes> (Q1 - 1.5*IQR) & data.B$diabetes< (Q3 + 1.5*IQR))
dim(nb)
```

```
## [1] 380   9
```

```
dim(data.B)
```

```
## [1] 392   9
```

```
data.b=nb
```

```
shapiro.test(data.b$age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data.b$age
## W = 0.83698, p-value < 2.2e-16
```

```r
shapiro.test(data.b$glucose)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data.b$glucose
## W = 0.96394, p-value = 4.662e-08
```

```r
shapiro.test(data.b$b.m.index)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data.b$b.m.index
## W = 0.97776, p-value = 1.379e-05
```

```r
shapiro.test(data.b$diabetes)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data.b$diabetes
## W = 0.94157, p-value = 4.428e-11
```

```r
#none of the variables are normally distributed
#log transformation (not working)
shapiro.test(log(data.b$age))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log(data.b$age)
## W = 0.90108, p-value = 5.164e-15
```

```r
shapiro.test(log(data.b$glucose))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log(data.b$glucose)
## W = 0.98715, p-value = 0.001929
```

```r
shapiro.test(log(data.b$b.m.index))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log(data.b$b.m.index)
## W = 0.99221, p-value = 0.04436
```

```
shapiro.test(log(data.b$diabetes))
```

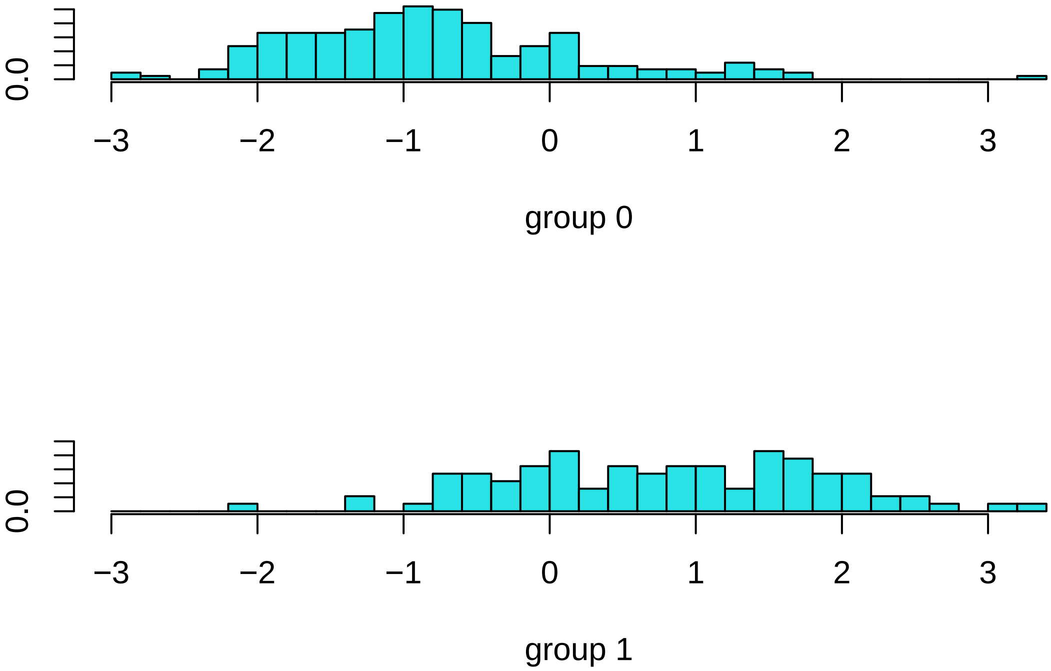
```
##
##  Shapiro-Wilk normality test
##
## data:  log(data.b$diabetes)
## W = 0.98338, p-value = 0.0002323
```

Discriminant Analysis(data set B)

```
library(MASS)
set.seed(2021)
# Create dummy for splitting (80-20)
split_dummy <- sample(c(rep(0, 0.8 * nrow(data.b)),
                        rep(1, 0.2 * nrow(data.b))))
data_train.b <- data.b[split_dummy == 0, ]
data_test.b <- data.b[split_dummy == 1, ]
#Linear Discrimination Analysis
lda.b=lda(class ~  glucose  + b.m.index + diabetes + age,data = data_train.b)
lda.b
```

```
## Call:
## lda(class ~ glucose + b.m.index + diabetes + age, data = data_train.b)
##
## Prior probabilities of groups:
##         0         1
## 0.6940789 0.3059211
##
## Group means:
##    glucose b.m.index  diabetes      age
## 0 110.8483  31.79242 0.4462275 28.36019
## 1 146.2258  35.62366 0.5597097 35.66667
##
## Coefficients of linear discriminants:
##                  LD1
## glucose   0.02943377
## b.m.index 0.04660949
## diabetes  0.87156469
## age       0.03526822
```

```
plot(lda.b)
```



```r
#prediction with the test data
lda.b.pred = predict(lda.b,data_test.b)
lda.b.class=lda.b.pred$class
table(lda.b.class,data_test.b$class)
```

```
##
## lda.b.class  0  1
##           0 39 12
##           1  8 17
```

```r
#sensitivity
sensitivity.lda.b=(39+17)/76
sensitivity.lda.b
```

```
## [1] 0.7368421
```

```r
#misclassification
mle.b=20/76
mle.b
```

```
## [1] 0.2631579
```

```
#FNR
fnr.lda.b=12/(12+17)
fnr.lda.b
```

```
## [1] 0.4137931
```

```
# Quadratic Discriminant Analysis
qda.b = qda(class ~  glucose  + b.m.index + diabetes + age,data = data_train.b)
qda.b
```

```
## Call:
## qda(class ~ glucose + b.m.index + diabetes + age, data = data_train.b)
##
## Prior probabilities of groups:
##         0         1
## 0.6940789 0.3059211
##
## Group means:
##    glucose b.m.index  diabetes      age
## 0 110.8483  31.79242 0.4462275 28.36019
## 1 146.2258  35.62366 0.5597097 35.66667
```

```
qda.b.pred = predict(qda.b,data_test.b)
qda.b.class=qda.b.pred$class
table(qda.b.class,data_test.b$class)
```

```
##
## qda.b.class  0  1
##           0 38 13
##           1  9 16
```

```
#sensitivity
sensitivity.qda.b=(38+16)/76
sensitivity.qda.b
```

```
## [1] 0.7105263
```

```
#misclassification
mqe.b=(9+13)/76
mqe.b
```

```
## [1] 0.2894737
```

```
#FNR
fnr.qda.b=13/(13+16)
fnr.qda.b
```

```
## [1] 0.4482759
```

```
# K-Nearest Neighbors
library(class)
train.x.b=cbind(data_train.b$n.pregnant,data_train.b$glucose,data_train.b$b.m.index,data_train.b$diabete
test.x.b=cbind(data_test.b$n.pregnant,data_test.b$glucose,data_test.b$b.m.index,data_test.b$diabetes)
knn.b.pred=knn(train.x.b,test.x.b,data_train.b$class,k=3)
table(knn.b.pred,data_test.b$class)
```

```
##
## knn.b.pred  0  1
##          0 43 15
##          1  4 14
```

```
#sensitivity
sensitivity.knn.b=(43+14)/76
sensitivity.knn.b
```

```
## [1] 0.75
```

```
#misclassification
mke.b=(4+15)/76
mke.b
```

```
## [1] 0.25
```

```
#FNR
fnr.knn.b=15/(14+15)
fnr.knn.b
```

```
## [1] 0.5172414
```

Conclusion for data set B

The lda(73%) and qda(71%) give close results for the data set B but the knn method gives a sensitivity of 75%.

Conclusion

```
nam=c("lda-a","qda-a","knn-a","lda-b","qda-b","knn-b")
sen=c(sensitivity.lda.a,sensitivity.qda.a,sensitivity.knn.a,sensitivity.lda.b,sensitivity.qda.b,sensitiv
ms=c(mle.a,mqe.a,mke.a,mle.b,mqe.b,mke.b)
fnr=c(fnr.lda.a,fnr.qda.a,fnr.knn.a,fnr.lda.b,fnr.qda.b,fnr.knn.b)
d=data.frame(nam,sen,ms,fnr)
knitr::kable(d,"pipe",col.names = c("Method-DataSet","Sensitivity","Misclassification Rate","False Nega
```

Method-DataSet	Sensitivity	Misclassification Rate	False Negative Rate
lda-a	0.7567568	0.2432432	0.5000000

Method-DataSet	Sensitivity	Misclassification Rate	False Negative Rate
qda-a	0.7500000	0.2500000	0.4333333
knn-a	0.6959459	0.3040541	0.5000000
lda-b	0.7368421	0.2631579	0.4137931
qda-b	0.7105263	0.2894737	0.4482759
knn-b	0.7500000	0.2500000	0.5172414

- LDA

The lda method gives similar results in sensitivity for both data sets but the false negative rate is 9% smaller in the data set B.So if we are going to use the lda method for prediction and we have NAs in our data set ,the best thing we can do is to exclude these individuals from our research.

- QDA

The qda method has higher sensitivity in the data set A and the false negative rate is almost the same in both data sets.So the best option for us is to impute the missing variables with the Lasso Linear Regression method.

*KNN

The sensitivity in the Knn method is 6% in the second data set and the false negative rate has only 1% difference.So if we are gong to use the knn method the best thing to do is to exclude the missing values.

In conclusion,the best method for prediction of the developing diabetes in women ,is the Quadratic Discriminant Analysis.This method gave the highest sensitivity and the lowest false negative rate which is really important for medical studies.Also the best results achieved with imputation of the missing values with the Lasso linear regression method.