SDM_Assignment3_3

Sri Balaji Muruganandam

25/10/2021

Setting Working Directory

```
rm(list = ls())
setwd("G:\\SDM_Sem01\\Assignment3")
```

Importing necessary libraries

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.1.1
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.1
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.1
```

```
## Loading required package: lattice
```

```
library(class)
library(ggplot2)
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.1.1
```

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.1.1
```

```
library(mvtnorm)
```

```
## Warning: package 'mvtnorm' was built under R version 4.1.1
```

```
library(MASS)
```

Loading the diabetes dataset

```
load("Diabetes.RData")
dim(Diabetes)
```

```
## [1] 145   6
```

Exploring the high level overview of the data

```
head(Diabetes,5)
```

```
##   relwt glufast glutest instest sspg  group
## 1  0.81      80     356     124   55 Normal
## 2  0.95      97     289     117   76 Normal
## 3  0.94     105     319     143  105 Normal
## 4  1.04      90     356     199  108 Normal
## 5  1.00      90     323     240  143 Normal
```

```
str(Diabetes)
```

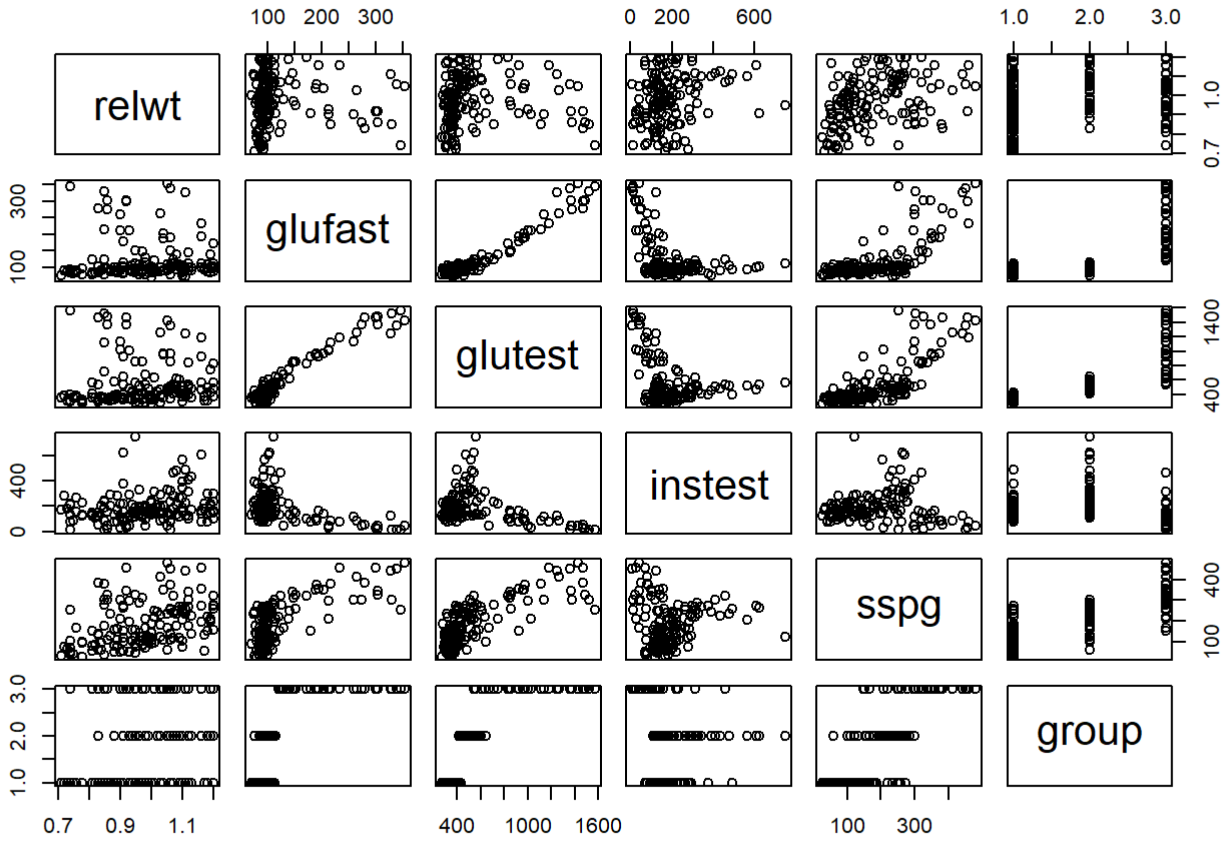
```
## 'data.frame':    145 obs. of  6 variables:
##  $ relwt  : num  0.81 0.95 0.94 1.04 1 0.76 0.91 1.1 0.99 0.78 ...
##  $ glufast: int  80 97 105 90 90 86 100 85 97 97 ...
##  $ glutest: int  356 289 319 356 323 381 350 301 379 296 ...
##  $ instest: int  124 117 143 199 240 157 221 186 142 131 ...
##  $ sspg   : int  55 76 105 108 143 165 119 105 98 94 ...
##  $ group  : Factor w/ 3 levels "Normal","Chemical_Diabetic",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(Diabetes)
```

```
##      relwt           glufast       glutest          instest
##  Min.   :0.7100   Min.   : 70   Min.   : 269.0   Min.   : 10.0
##  1st Qu.:0.8800   1st Qu.: 90   1st Qu.: 352.0   1st Qu.:118.0
##  Median :0.9800   Median : 97   Median : 413.0   Median :156.0
##  Mean   :0.9773   Mean   :122   Mean   : 543.6   Mean   :186.1
##  3rd Qu.:1.0800   3rd Qu.:112   3rd Qu.: 558.0   3rd Qu.:221.0
##  Max.   :1.2000   Max.   :353   Max.   :1568.0   Max.   :748.0
##       sspg                     group
##  Min.   : 29.0   Normal           :76
##  1st Qu.:100.0   Chemical_Diabetic:36
##  Median :159.0   Overt_Diabetic   :33
##  Mean   :184.2
##  3rd Qu.:257.0
##  Max.   :480.0
```

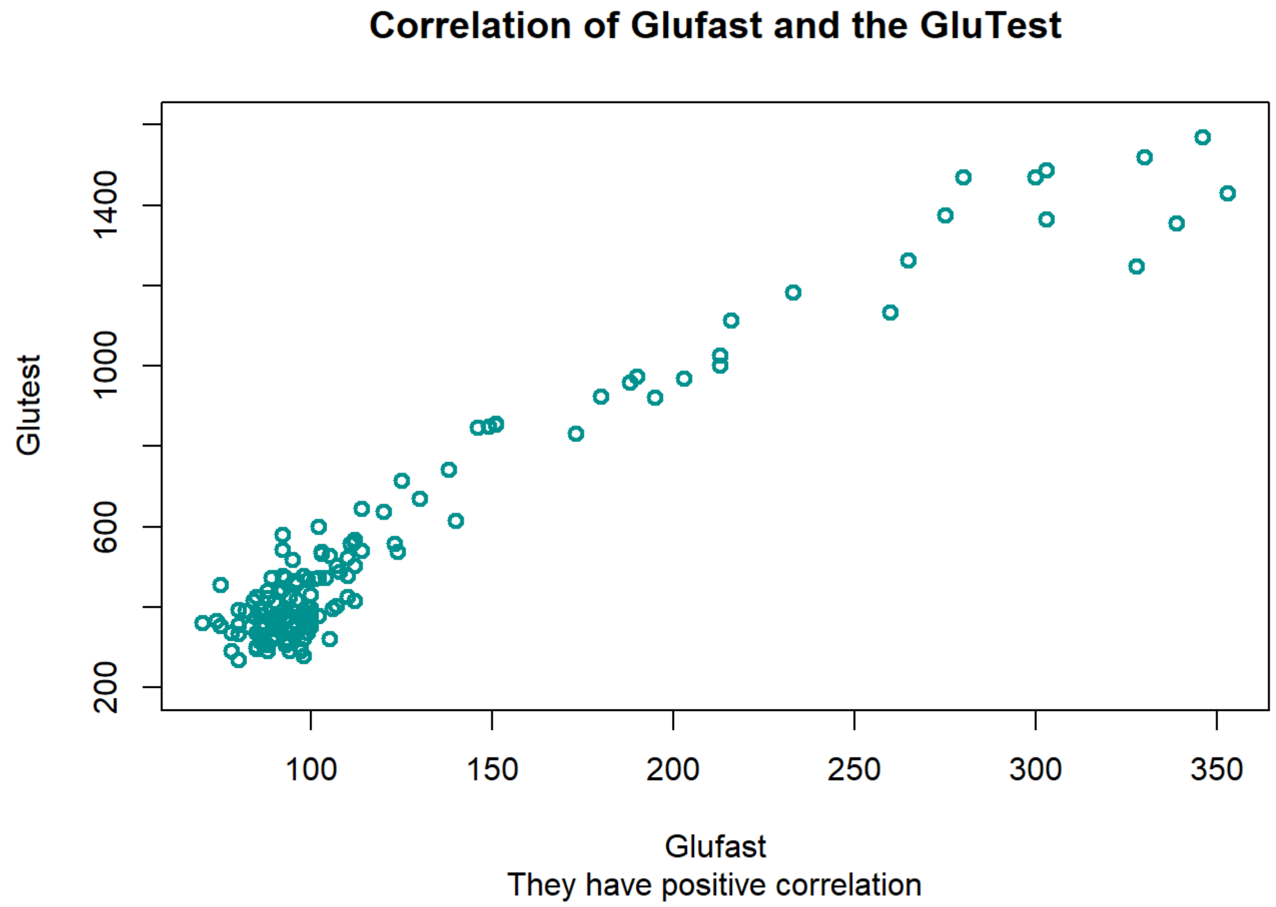
Plotting Scatter Plot

```
plot(Diabetes)
```



### There is a positive correlation between Glufast and the GluTest

```
plot(Diabetes$glufast, Diabetes$glutest,col="#00918E", xlab = "Glufast", ylab = "Glutest", yl
im = c(200,1600), main = "Correlation of Glufast and the GluTest", sub="They have positive co
rrelation", lwd = 2.3)
```



## Scatter plot of five variables - relwt, glufast, glutest, instest, sspg

```
table(Diabetes$group)
```

```
## 
##             Normal Chemical_Diabetic    Overt_Diabetic 
##                 76                36                33
```

```
#Diabetes$group = as.numeric(Diabetes$group)
```

relwt

```
d1 = qplot(Diabetes$relwt, Diabetes$glutest, colour = Diabetes$group ,data = Diabetes)
d2 = qplot(Diabetes$relwt, Diabetes$instest, colour = Diabetes$group ,data = Diabetes)
d3 = qplot(Diabetes$relwt, Diabetes$sspg, colour = Diabetes$group ,data = Diabetes)
d4 = qplot(Diabetes$relwt, Diabetes$glufast, colour = Diabetes$group ,data = Diabetes)

grid.arrange(d1, d2, d3, d4, nrow = 2, ncol=2)
```

```
## Warning: Use of `Diabetes$relwt` is discouraged. Use `relwt` instead.
```

```
## Warning: Use of `Diabetes$glutest` is discouraged. Use `glutest` instead.
```

```
## Warning: Use of `Diabetes$group` is discouraged. Use `group` instead.
## Warning: Use of `Diabetes$relwt` is discouraged. Use `relwt` instead.
```

```
## Warning: Use of `Diabetes$instest` is discouraged. Use `instest` instead.
```

```
## Warning: Use of `Diabetes$group` is discouraged. Use `group` instead.
```

```
## Warning: Use of `Diabetes$relwt` is discouraged. Use `relwt` instead.
```

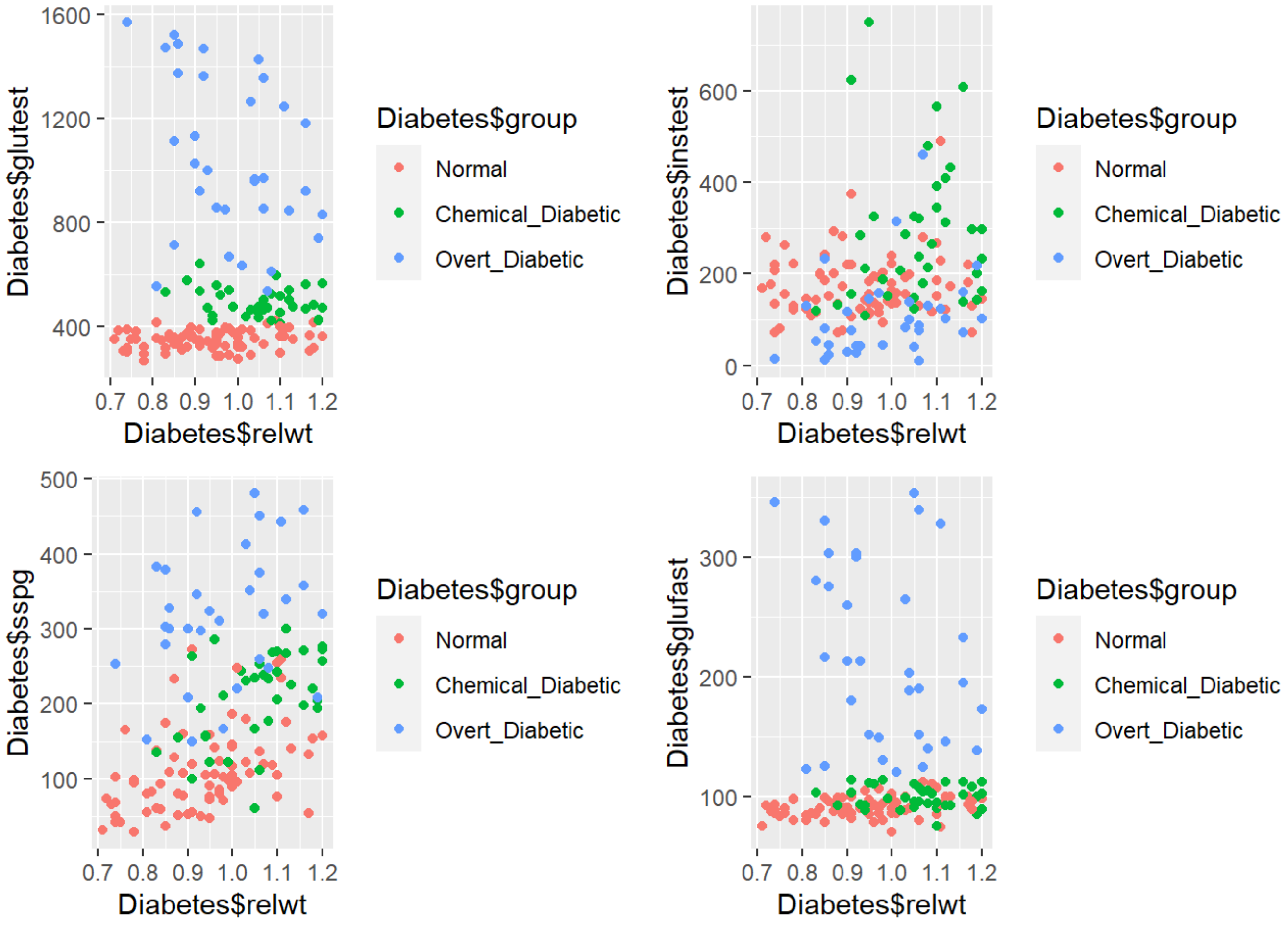
```
## Warning: Use of `Diabetes$sspg` is discouraged. Use `sspg` instead.
```

```
## Warning: Use of `Diabetes$group` is discouraged. Use `group` instead.
```

```
## Warning: Use of `Diabetes$relwt` is discouraged. Use `relwt` instead.
```

```
## Warning: Use of `Diabetes$glufast` is discouraged. Use `glufast` instead.
```

```
## Warning: Use of `Diabetes$group` is discouraged. Use `group` instead.
```



glufast

```
d1 = qplot(Diabetes$glufast,Diabetes$glutest, colour = Diabetes$group ,data = Diabetes)
d2 = qplot(Diabetes$glufast,Diabetes$instest, colour = Diabetes$group ,data = Diabetes)
d3 = qplot(Diabetes$glufast,Diabetes$sspg, colour = Diabetes$group ,data = Diabetes)

grid.arrange(d1, d2, d3, nrow = 2, ncol=2)
```

```
## Warning: Use of `Diabetes$glufast` is discouraged. Use `glufast` instead.
```

```
## Warning: Use of `Diabetes$glutest` is discouraged. Use `glutest` instead.
```

```
## Warning: Use of `Diabetes$group` is discouraged. Use `group` instead.
```

```
## Warning: Use of `Diabetes$glufast` is discouraged. Use `glufast` instead.
```

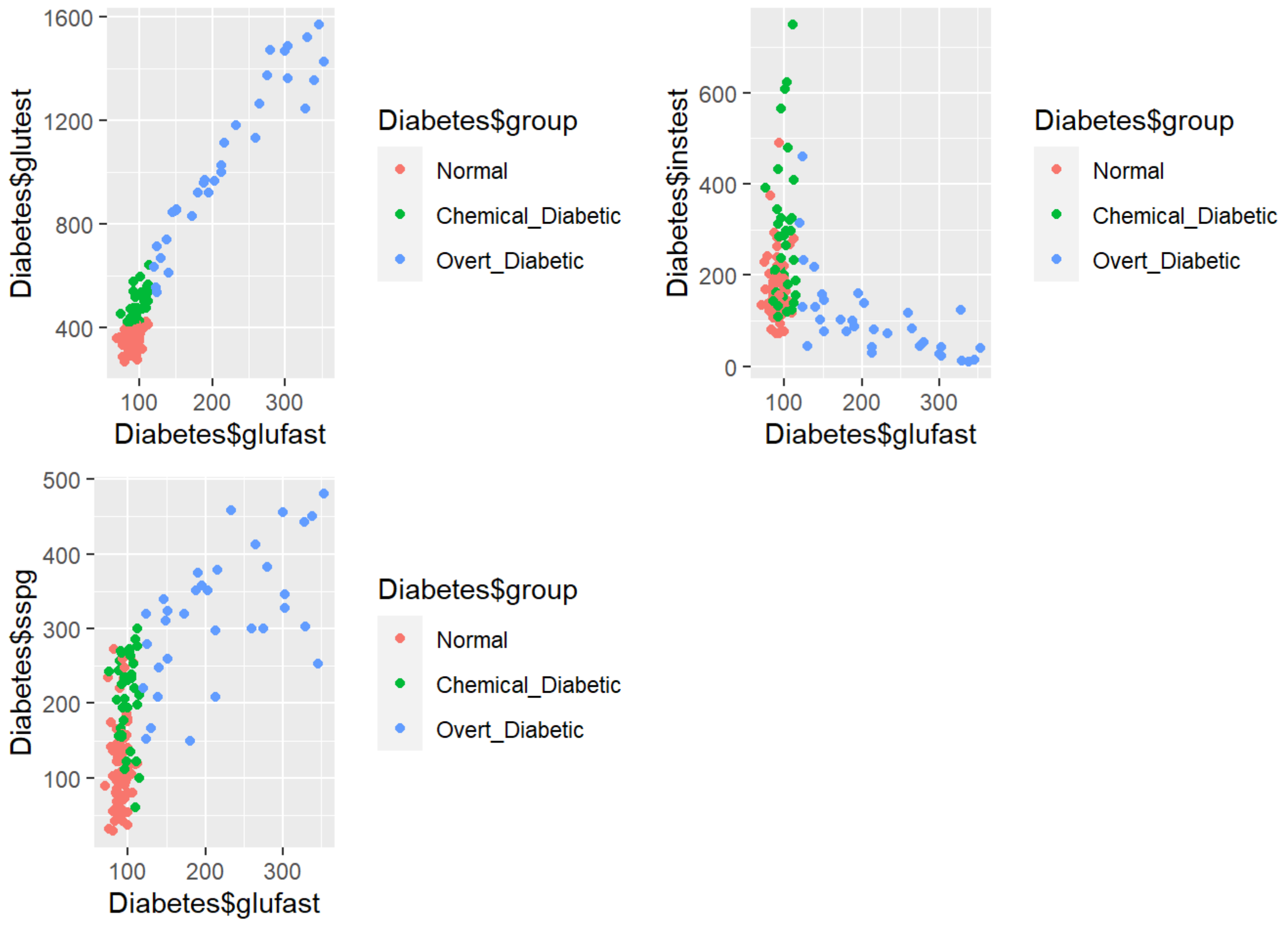
```
## Warning: Use of `Diabetes$instest` is discouraged. Use `instest` instead.
```

```
## Warning: Use of `Diabetes$group` is discouraged. Use `group` instead.
```

```
## Warning: Use of `Diabetes$glufast` is discouraged. Use `glufast` instead.
```

```
## Warning: Use of `Diabetes$sspg` is discouraged. Use `sspg` instead.
```

```
## Warning: Use of `Diabetes$group` is discouraged. Use `group` instead.
```



glutest

```
d1 = qplot(Diabetes$glutest,Diabetes$instest, colour = Diabetes$group ,data = Diabetes)
d2 = qplot(Diabetes$glutest,Diabetes$sspg, colour = Diabetes$group ,data = Diabetes)

grid.arrange(d1, d2, nrow = 1, ncol=2)
```

```
## Warning: Use of `Diabetes$glutest` is discouraged. Use `glutest` instead.
```

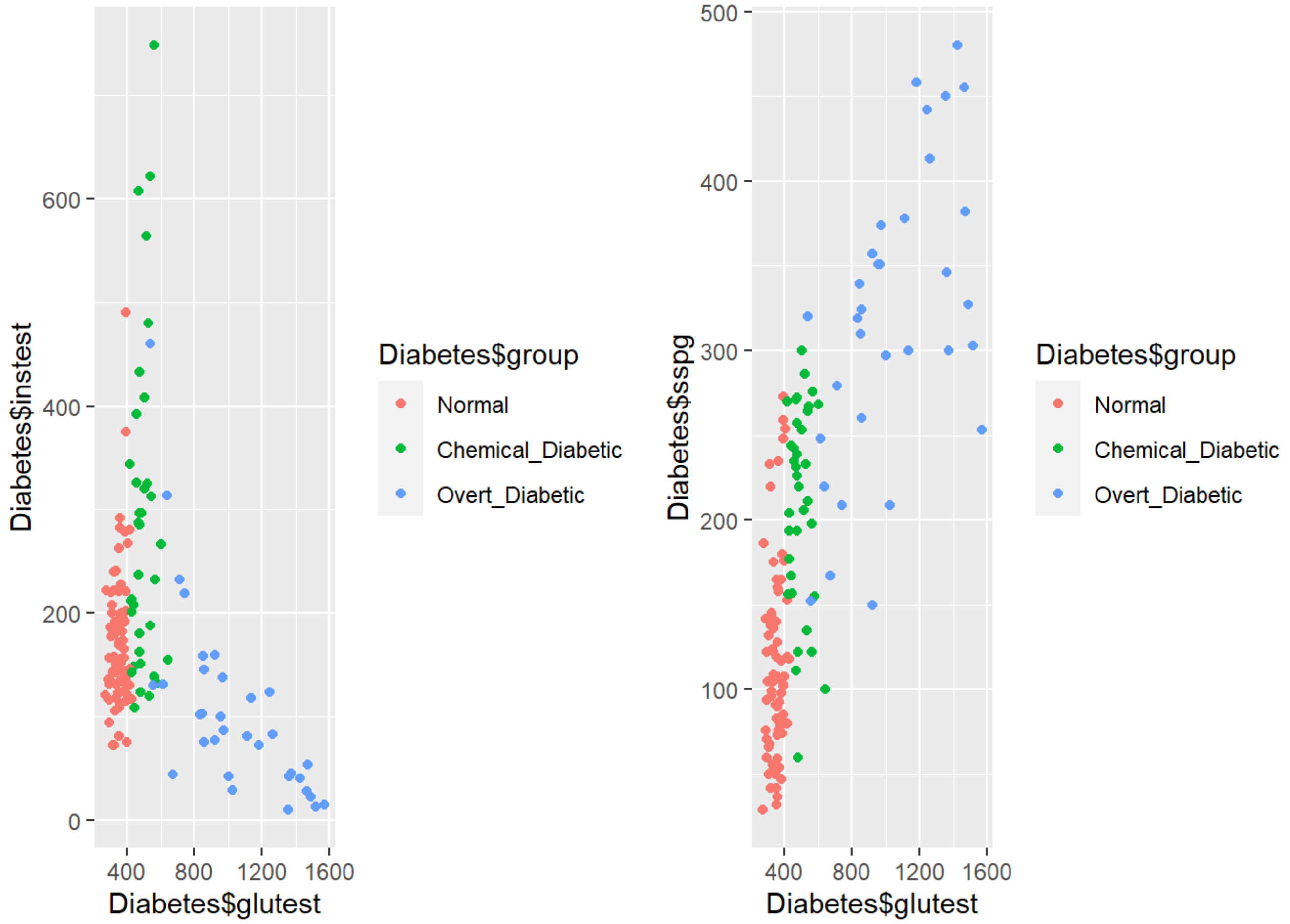
```
## Warning: Use of `Diabetes$instest` is discouraged. Use `instest` instead.
```

```
## Warning: Use of `Diabetes$group` is discouraged. Use `group` instead.
```

```
## Warning: Use of `Diabetes$glutest` is discouraged. Use `glutest` instead.
```

```
## Warning: Use of `Diabetes$sspg` is discouraged. Use `sspg` instead.
```

```
## Warning: Use of `Diabetes$group` is discouraged. Use `group` instead.
```



Covariance

```
data2 = Diabetes
Diabetes$group = as.numeric(Diabetes$group)
covariance = cov(Diabetes)
print(covariance)
```

```
##               relwt       glufast       glutest       instest         sspg
## relwt    0.01670174 -7.281513e-02  9.824262e-01      3.473373     5.266255
## glufast -0.07281513  4.087097e+03  1.954606e+04  -3063.463649  4849.905651
## glutest  0.98242625  1.954606e+04  1.004578e+05 -12918.162739 25908.490182
## instest  3.47337308 -3.063464e+03 -1.291816e+04  14625.312548   101.482519
## sspg     5.26625479  4.849906e+03  2.590849e+04    101.482519 11242.331897
## group    0.02266906  3.818338e+01  2.168152e+02    -11.249713    67.915948
##                group
## relwt     0.02266906
## glufast  38.18338123
## glutest 216.81522989
## instest -11.24971264
## sspg     67.91594828
## group     0.66839080
```

Classes have strong correlation with the predictor glutest and sspg

calculating multivariate normal

```
norm_d = dmvnorm(x=Diabetes$group)
print(norm_d)
```

```
## [1] 7.359291e-171
```

(b) Apply linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA). How does the performance of QDA compare to that of LDA in this case?

Preparing data

```
#data2 = Diabetes
#set.seed(123)
#set.seed(121)
set.seed(123)
random_index = sample(c(1:nrow(Diabetes)), size = round(8/10 * nrow(Diabetes)), replace = FAL
SE)
train_data2 <- data2[random_index,]
test_data2 <- data2[-random_index,]

train_data2 = data.frame(train_data2)
test_data2 = data.frame(test_data2)
data2 = data.frame(data2)


y_train_data2 <- as.numeric(train_data2$group)-1
y_test_data2 <- as.numeric(test_data2$group)-1

#y_train_data2 <- train_data2$group
#y_test_data2 <- test_data2$group
```

Modelling LDA

```
lda_model = lda(group~., data = train_data2)
summary(lda_model)
```

```
##         Length Class  Mode
## prior    3     -none- numeric
## counts   3     -none- numeric
## means   15     -none- numeric
## scaling 10     -none- numeric
## lev      3     -none- character
## svd      2     -none- numeric
## N        1     -none- numeric
## call     3     -none- call
## terms    3     terms  call
## xlevels  0     -none- list
```

Predicting for test data

```
lda_predict_train = predict(lda_model, newdata = train_data2)
lda_predict_test = predict(lda_model, newdata = test_data2)
res_train = as.numeric(lda_predict_train$class)-1
res_test = as.numeric(lda_predict_test$class)-1
#res_train = lda_predict_train$class
#res_test = lda_predict_test$class
```

Calculating the train and test errors

```
lda_result_train = which(res_train==y_train_data2)
lda_train_error=length(lda_result_train) / length(y_train_data2)
print(lda_train_error)
```

```
## [1] 0.9051724
```

```
lda_result_test = which(res_test==y_test_data2)
lda_test_error=length(lda_result_test) / length(y_test_data2)
print(lda_test_error)
```

```
## [1] 0.862069
```

The accuracy for training set is 90.51% and the accuracy for testing set is 86.20% when predicted using LDA.

Modelling QDA

```
qda_model = qda(group~., data = train_data2)
summary(qda_model)
```

```
##         Length Class  Mode
## prior    3     -none- numeric
## counts   3     -none- numeric
## means   15     -none- numeric
## scaling 75     -none- numeric
## ldet     3     -none- numeric
## lev      3     -none- character
## N        1     -none- numeric
## call     3     -none- call
## terms    3     terms  call
## xlevels  0     -none- list
```

Predicting for test data

```
qda_predict_train = predict(qda_model, newdata = train_data2)
qda_predict_test = predict(qda_model, newdata = test_data2)
res_trainq = as.numeric(qda_predict_train$class)-1
res_testq = as.numeric(qda_predict_test$class)-1
#res_trainq = qda_predict_train$class
#res_testq = qda_predict_test$class
```

Calculating the train and test errors

```
qda_result_train = which(res_trainq==y_train_data2)
qda_train_error=length(qda_result_train) / length(y_train_data2)
print(qda_train_error)
```

```
## [1] 0.9655172
```

```
qda_result_test = which(res_testq==y_test_data2)
qda_test_error=length(qda_result_test) / length(y_test_data2)
print(qda_test_error)
```

```
## [1] 0.8965517
```

The accuracy for training set is 96.55% and the accuracy for testing set is 89.65% when predicted using QDA.

QDA provides better prediction than the LDA

(c) Suppose an individual has (glucose test/intolerence= 68, insulin test=122, SSPG = 544. Relative weight = 1.86, fasting plasma glucose = 184). To which class does LDA assign this individual? To which class does QDA?

```
#to_predict = c(1.86, 184, 68, 122, 544)
new_data1 <-data.frame(1.86, 184, 68, 122, 544)

colnames(new_data1)<-c("relwt","glufast","glutest","instest","sspg")
head(new_data1)
```

```
##   relwt glufast glutest instest sspg
## 1  1.86     184      68     122  544
```

```
new_data_predict = predict(qda_model, newdata = new_data1)
print(new_data_predict$class)
```

```
## [1] Overt_Diabetic
## Levels: Normal Chemical_Diabetic Overt_Diabetic
```

The class is found to be Overt_Diabetic for the given predictor values when predicted using QDA model

```
new_data_predict_lda = predict(lda_model, newdata = new_data1)
print(new_data_predict_lda$class)
```

```
## [1] Normal
## Levels: Normal Chemical_Diabetic Overt_Diabetic
```

The class is found to be Normal for the given predictor values when predicted using LDA model