Pima Indians Diabetes Database

Presented by The French Fries

Pima indians

- **WHERE:** The Pima are a group of Native Americans living in an area consisting of what is now central and southern Arizona, as well as northwestern Mexico in the states of Sonora and Chihuahua.

- **THE NAME:** The short name, "Pima", is believed to have come from the phrase pi 'añi mac or pi mac, meaning "I don't know," which they used repeatedly in their initial meetings with Spanish colonists.

The dataset

- **SOURCE:** National Institute of Diabetes and Digestive and Kidney Diseases.
- **AIM:** to diagnostically predict whether or not a patient has diabetes (type 2), based on certain diagnostic measurements included in the dataset.
- **PATIENTS:** females at least 21 years old of Pima Indian heritage.

What is diabetes?

- *Diabetes mellitus* is a metabolism (= how the body uses and digests food for growth and energy) disorder.
- Most of our consumed foods are broken down into glucose which is a type of sugar in the blood. Glucose is the main source of food for our bodies (our cells).
- Without the help of insulin, the glucose cannot enter our cells.
- Insulin is produced by Beta cells in the Islets of Langerhans, which are in the pancreas. After eating, the pancreas release an adequate amount of insulin to transport the blood glucose into the cells.

- IIf any one has diabetes, the glucose in the bloodstream does not enter the cells (at all or not enough), so glucose builds up until levels are too high, resulting in a condition called hyperglycemia. Hyperglycemia happens for one of two main reasons:
    - the body is producing no insulin which is known as "Diabetes Type 1";
    - the cells do not respond correctly to the insulin which is known as "Diabetes Type 2".

Our variables

- **PREGNANCIES**: number of times pregnant. Having diabetes should not affect fertility. If you have gestational diabetes during pregnancy, generally your blood sugar returns to its usual level soon after delivery. But if you've had gestational diabetes, you have a higher risk of getting type 2 diabetes.
- **GLUCOSE**: plasma glucose concentration a 2 hours in an oral glucose tolerance test.

2 hour GTT with 75 g intake	Normal	Hyperglycemia	
		Impaired glucose tolerance	Diabetes
	< 7.8 mmol/L (< 140 mg/dL)	$7.8 - 11.1$ mmol/L ($140 - 200$ mg/dL)	≥ 11.0 mmol/L (≥ 200 mg/dL)

- **BLOOD PRESSURE**: diastolic blood pressure (mm Hg). Most people with type 2 diabetes also suffer from high blood pressure.

Our variables

- **TRICEPS SKIN FOLD THICKNESS** (mm): from it total body fat and hence contribution of fat to body mass can be estimated..
- **INSULIN**: 2-Hour serum insulin (mu U/ml) appears to be a good indicator for insulin resistance, i.e when cells in your muscles, fat, and liver don't respond well to insulin and can't use glucose from your blood for energy.
- **BMI:** Body mass index (weight in kg/(height in m)^2. Severely obese people (BMI ≥40) were at an even greater risk of type 2 diabetes, when compared to obese people with a lower BMI (BMI 30- 39.9).
- **DIABETES PEDIGREE FUNCTION**: a function that represents how likely they are to get the disease by extrapolating from their ancestor's history.
- **AGE:** you can develop type 2 diabetes at any age, even during childhood. However, type 2 diabetes occurs most often in middle-aged and older people. You are more likely to develop type 2 diabetes if you are age 45 or older.

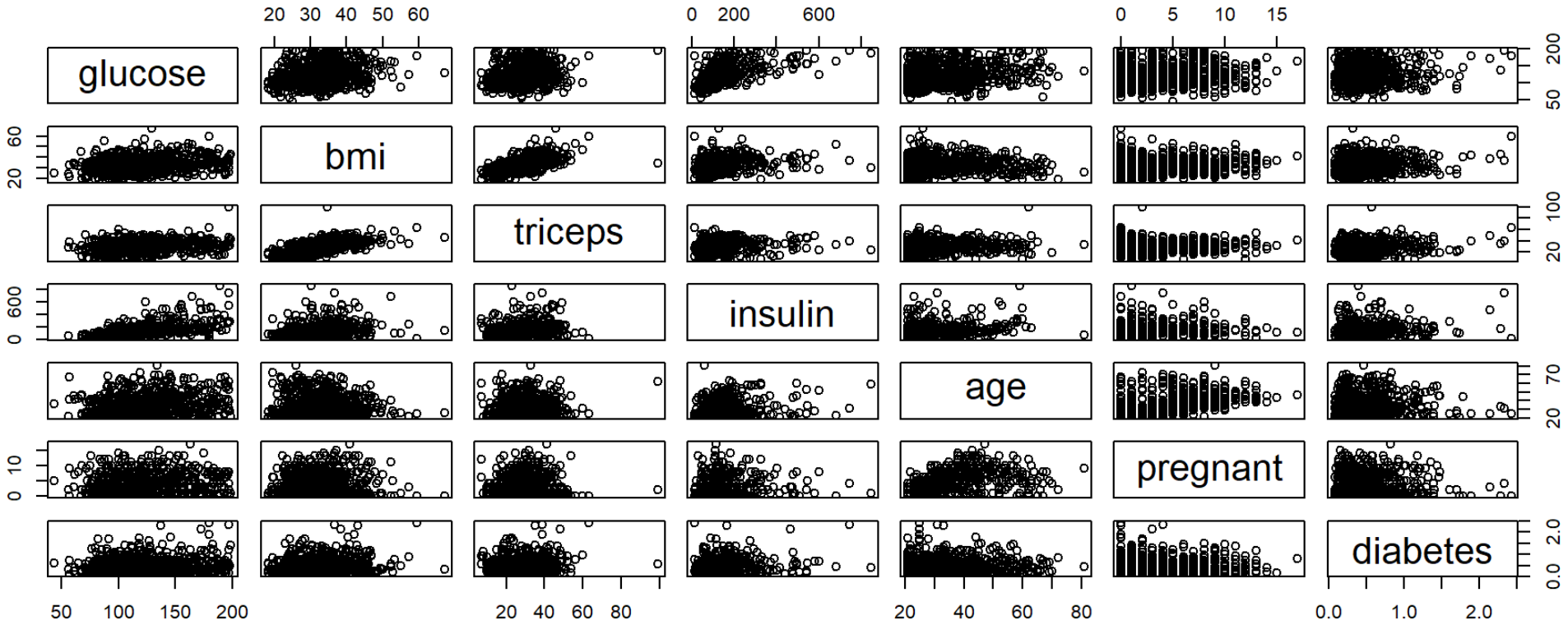
Data pre-processing

- Data pre-processing or data preparation is any type of manipulation that is performed on the raw data to prepare it for another data processing procedure. The first step that we are going to take to prepare our data is looking for missing values.

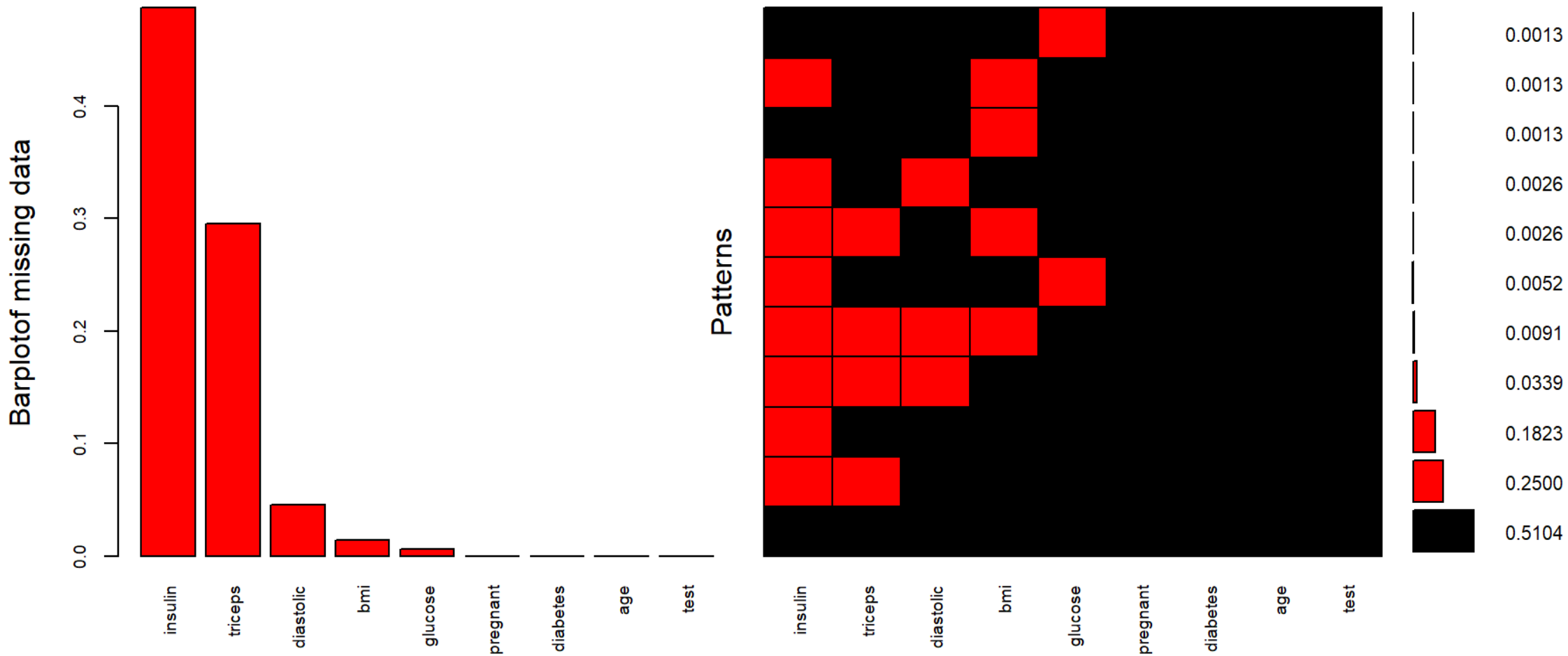
```
> library(faraway)
> data(pima)
> summary(pima)
    pregnant         glucose       diastolic         triceps          insulin     
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00   Min.   :  0.0  
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.0  
 Median : 3.000   Median :117.0   Median : 72.00   Median :23.00   Median : 30.5  
 Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54   Mean   : 79.8  
 3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:127.2  
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00   Max.   :846.0  
      bmi           diabetes          age             test       
 Min.   : 0.00   Min.   :0.0780   Min.   :21.00   Min.   :0.000  
 1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00   1st Qu.:0.000  
 Median :32.00   Median :0.3725   Median :29.00   Median :0.000  
 Mean   :31.99   Mean   :0.4719   Mean   :33.24   Mean   :0.349  
 3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00   3rd Qu.:1.000  
 Max.   :67.10   Max.   :2.4200   Max.   :81.00   Max.   :1.000
```

Visualization

```
> pairs(d[,c("glucose","bmi","triceps","insulin","age","pregnant","diabetes")])
```



```
> library(VIM)
> aggr<-aggr(d, col=c('black','red'), numbers=TRUE, sortVars=TRUE, labels=names(d),
 cex.axis=.7, gap=1, ylab=c("Barplotof missing data","Patterns"))
```

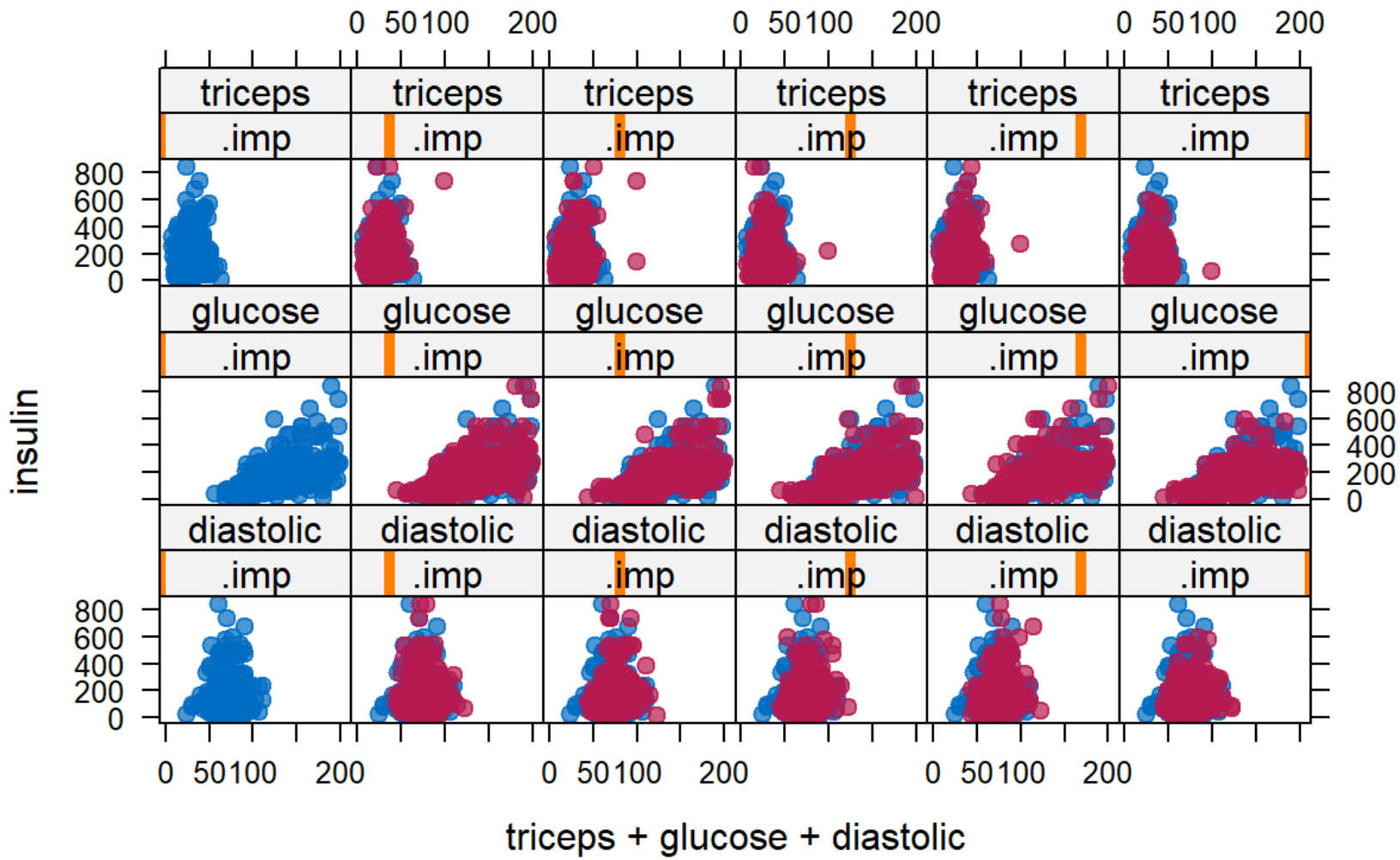


Missing value imputation

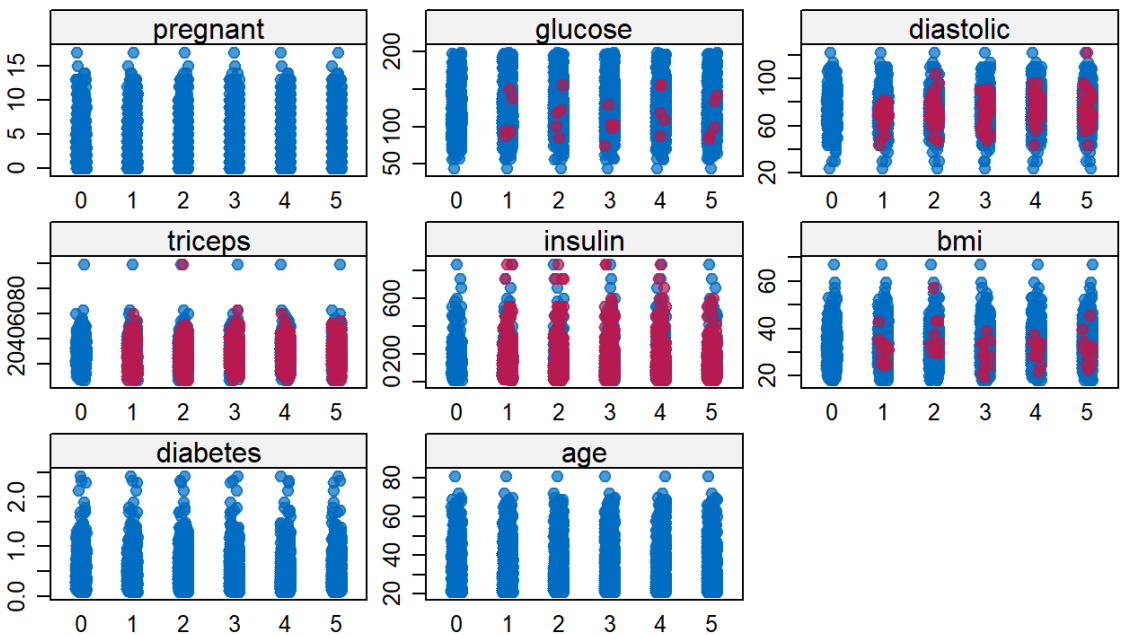
- In order to impute our missing values, from the Mice library, pmm algorithm is been used.
- Proposed by Rubin (1986) and Little (1988)
- Choose the variable with missing data and consider it as y
- Select the columns with complete data and call them x
- Regress y on x and obtain $\hat{\mathbf{y}}$
- $\hat{\mathbf{y}}_{obs}$ is the vector of estimates for observed $y_i's; i \in \{observed\}$
- $\hat{\mathbf{y}}_{miss}$ is the vector estimates for missing $y_j's; j \notin \{observed\}$
- The final estimate for the jth missing observation in the selected variable is :

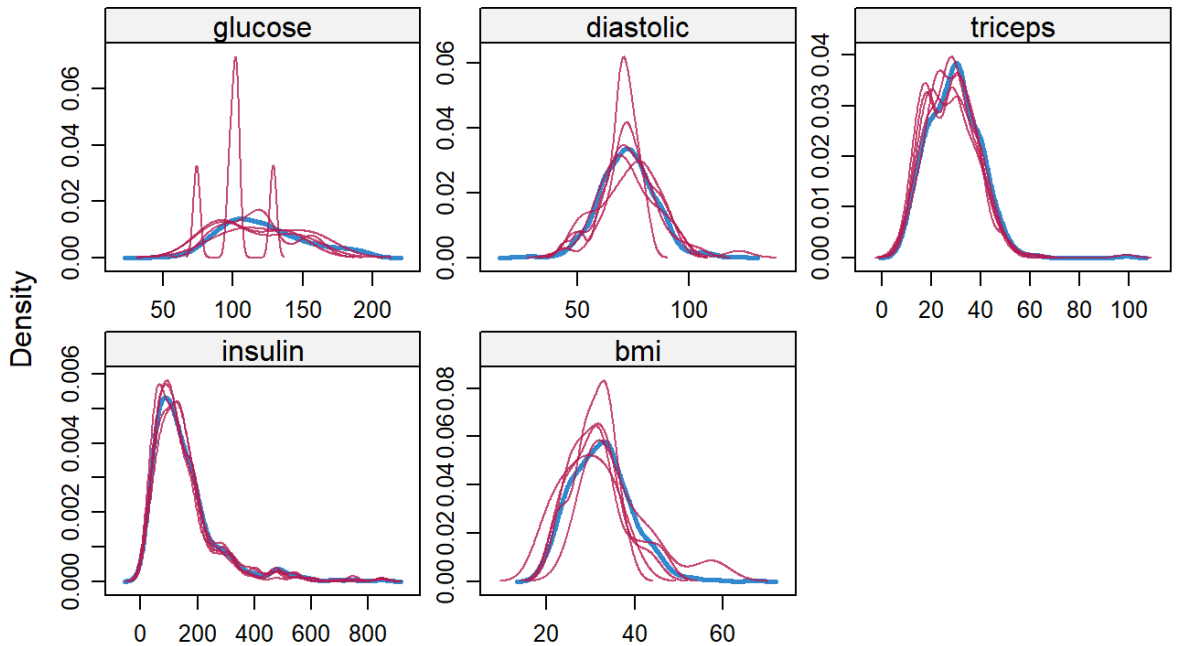
$$\hat{y}_{j,final} = \underset{y_{i,obs}}{\operatorname{argmin}} \left| \hat{y}_{j,miss} - \hat{y}_{i,obs} \right| ; i \in \{observed\}$$

```
> xyplot(impu, insulin ~ triceps+glucose+diastolic| .imp, pch= 20, cex= 1.4)
```



```
> stripplot(impu, pch= 20, cex= 1.2)
> densityplot(impu)
```





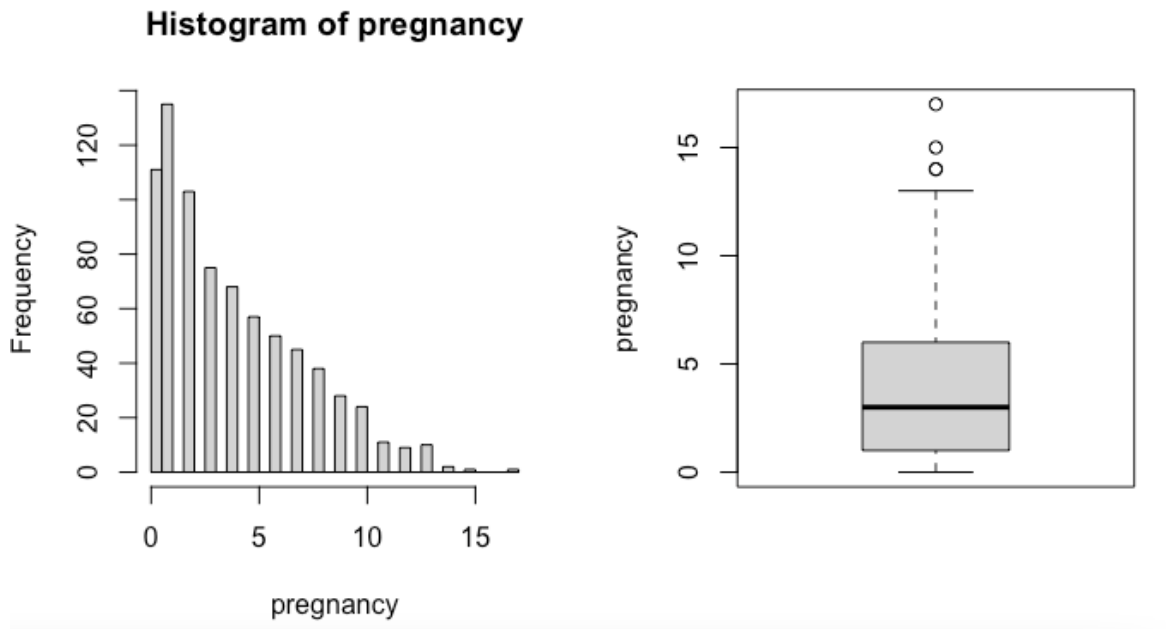
Outliers Detection

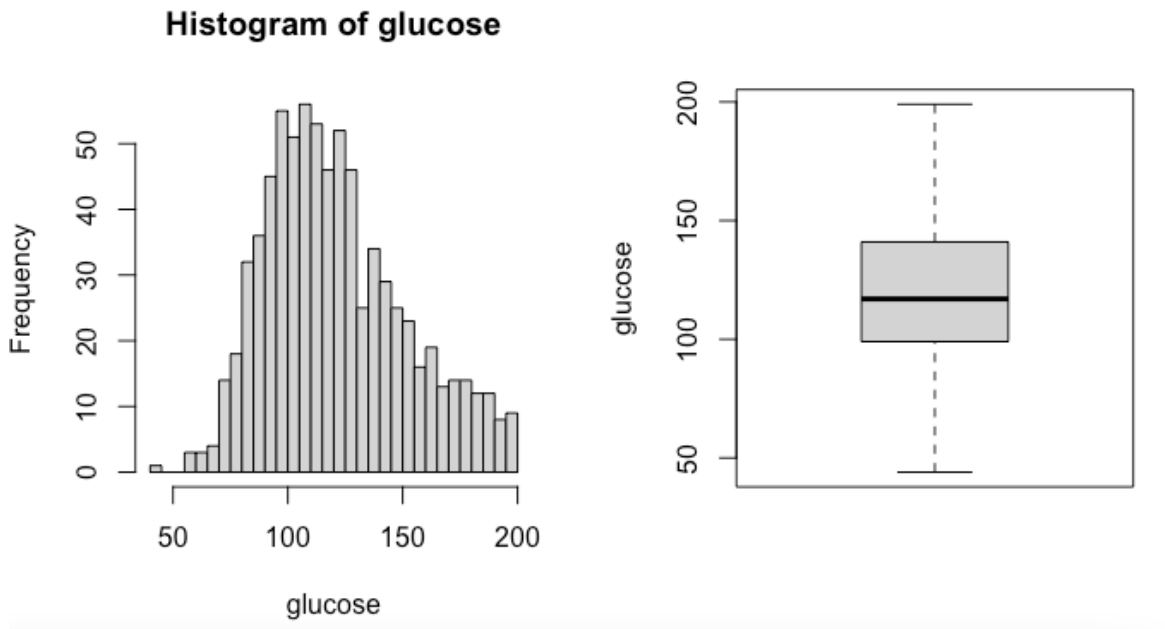
An outlier is a value or an observation that is distant from other observations (a data point that differs significantly from other data points). We will now start the detection of the outliers of our dataset using some descriptive statistics and in particular by finding the minimum, the maximum and the values of the quartiles.

```
> summary(DataSet)
    pregnant         glucose        diastolic        triceps          insulin     
 Min.   : 0.000   Min.   : 44.0   Min.   : 24.00   Min.   : 7.00   Min.   : 14.0  
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.00   1st Qu.:21.00   1st Qu.: 76.0  
 Median : 3.000   Median :117.0   Median : 72.00   Median :29.00   Median :120.0  
 Mean   : 3.845   Mean   :121.6   Mean   : 72.33   Mean   :28.60   Mean   :151.9  
 3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.00   3rd Qu.:35.25   3rd Qu.:190.0  
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00   Max.   :846.0  
      bmi           diabetes          age            test          
 Min.   :18.20   Min.   :0.0780   Min.   :21.00   Length:768        
 1st Qu.:27.50   1st Qu.:0.2437   1st Qu.:24.00   Class :character  
 Median :32.25   Median :0.3725   Median :29.00   Mode  :character  
 Mean   :32.42   Mean   :0.4719   Mean   :33.24                     
 3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00                     
 Max.   :67.10   Max.   :2.4200   Max.   :81.00                     
```

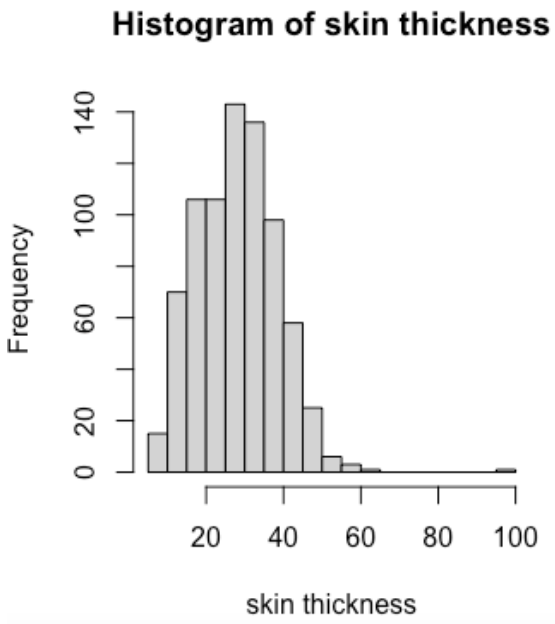
Outliers Detection

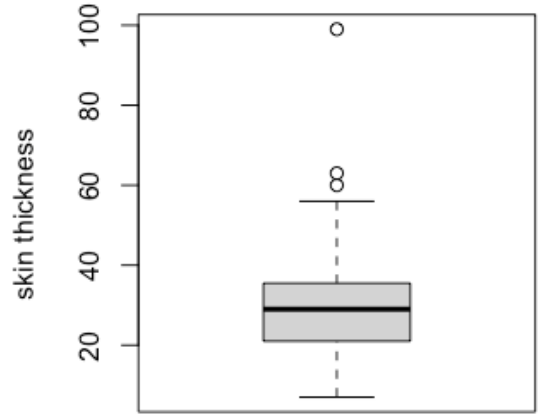
Another basic way to detect our potential outliers is to draw histograms or boxplots of the data.

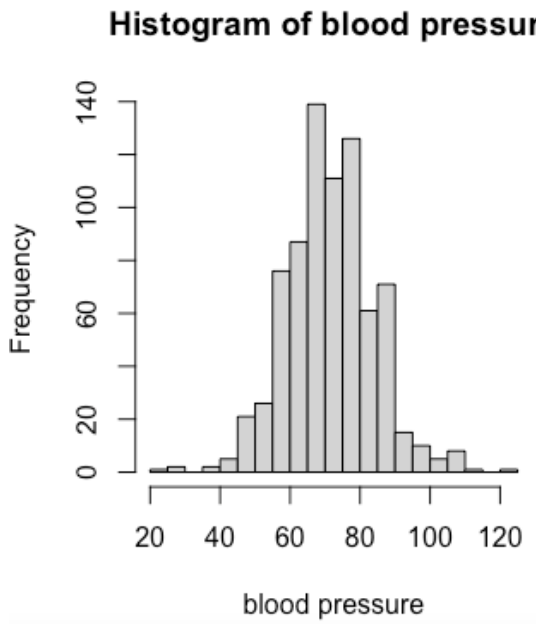


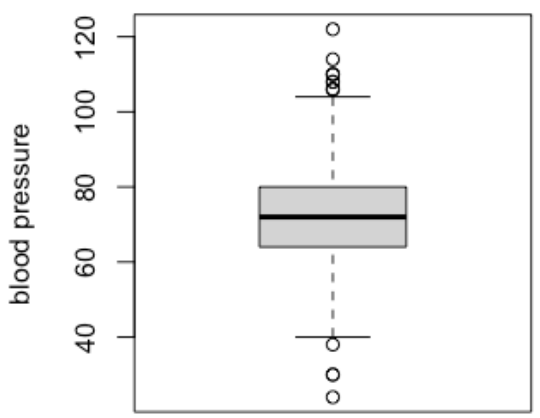


Outliers Detection

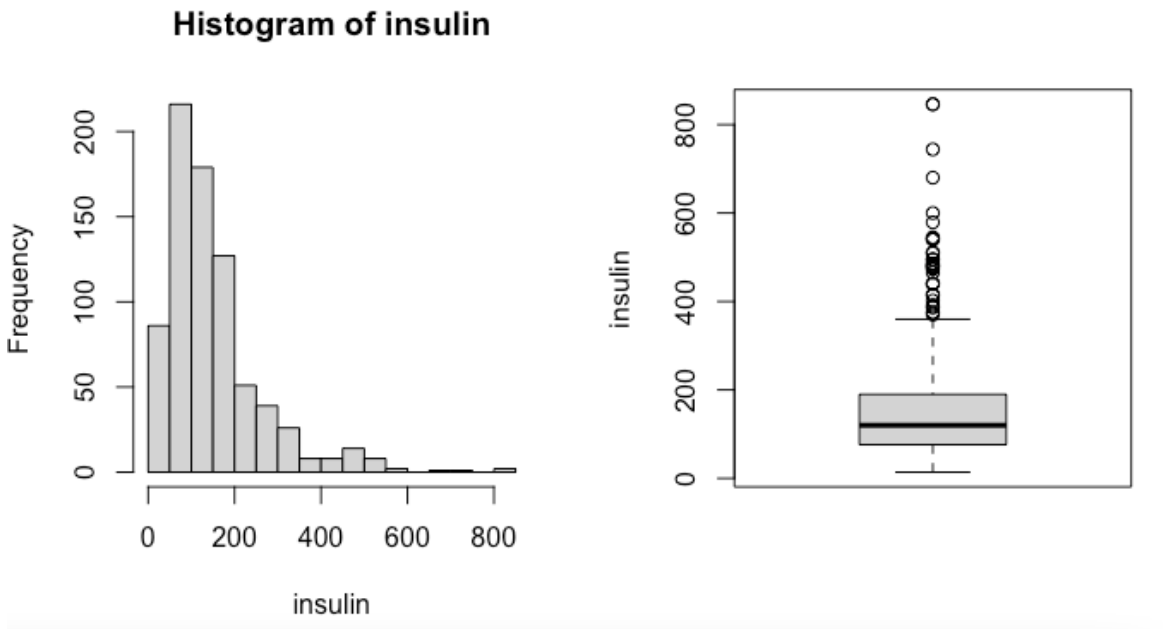


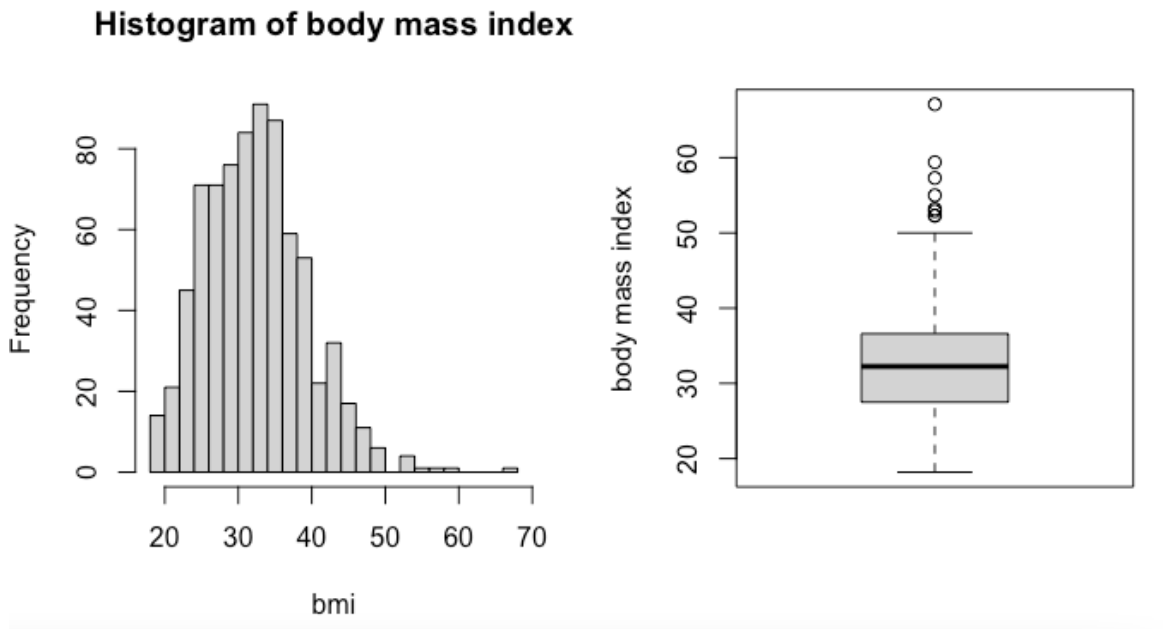




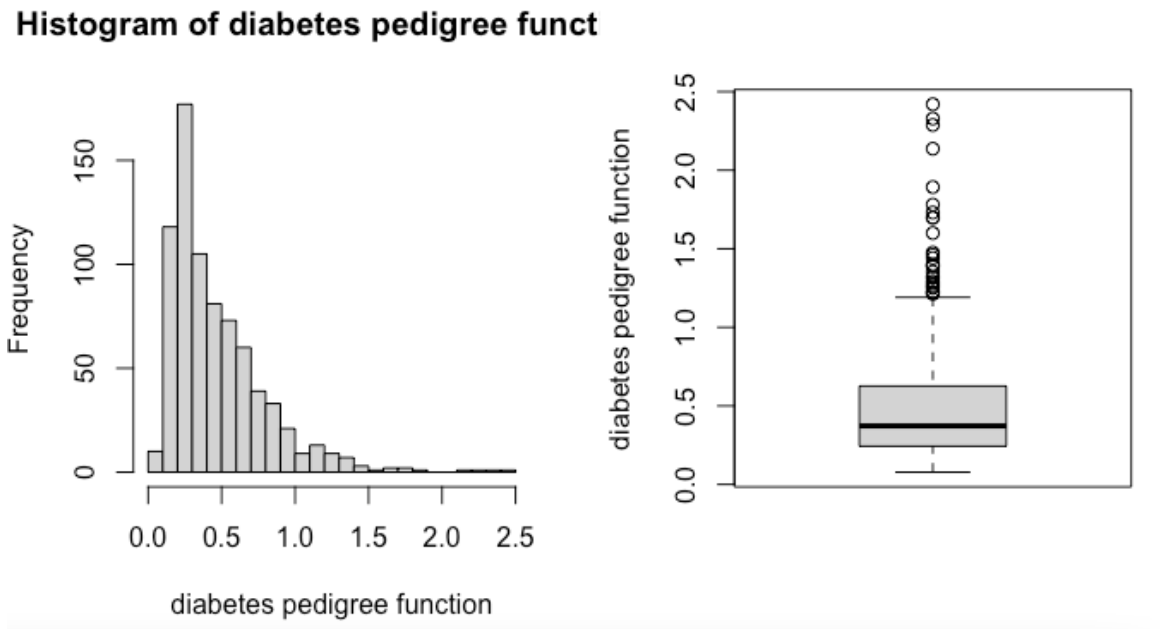


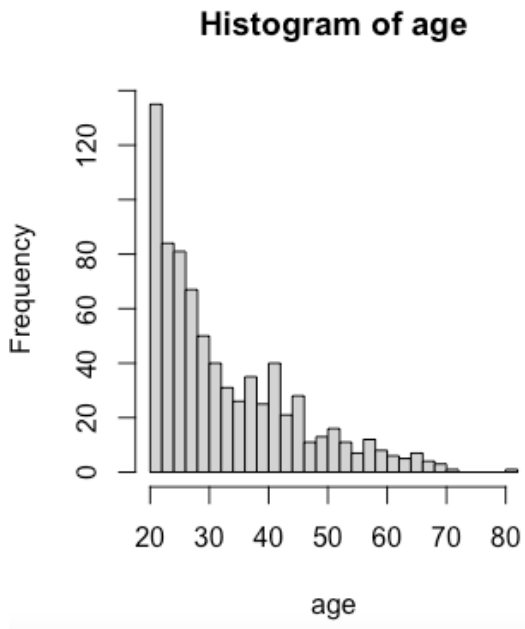
Outliers Detection

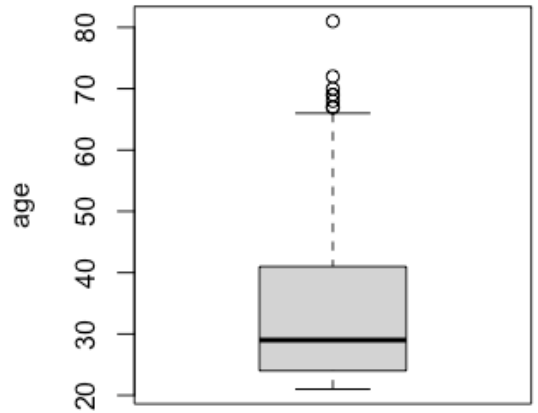




Outliers Detection







Outliers Detection

Now we are going to extract the values of the potential outliers using the boxplot.stats( )$out R function and their respective row number with which( ) function.

```
> pregnant_out <- boxplot.stats(pregnant)$out
> pregnant_out
[1] 15 17 14 14
> pregnant_out_ind <- which(pregnant %in% c(pregnant_out))
> pregnant_out_ind
[1]  89 160 299 456
> glucose_out <- boxplot.stats(glucose)$out
> glucose_out
integer(0)
> glucose_out_ind <- which(glucose %in% c(glucose_out))
> glucose_out_ind
integer(0)
> diastolic_out <- boxplot.stats(diastolic)$out
> diastolic_out
 [1]  30 110 108 122  30 110 108 110  24  38 106 106 106 114
> diastolic_out_ind <- which(diastolic %in% c(diastolic_out))
> diastolic_out_ind
 [1]  19  44  85 107 126 178 363 550 598 600 659 663 673 692
> triceps_out <- boxplot.stats(triceps)$out
> triceps_out
[1] 60 63 99
> triceps_out_ind <- which(triceps %in% c(triceps_out))
> triceps_out_ind
[1]  58 446 580
```

```
> insulin_out <- boxplot.stats(insulin)$out
> insulin_out
 [1] 543 846 387 440 440 495 480 485 495 485 480 478 744 370 510 680 402 375 545 543 465 415
[23] 510 495 579 474 375 480 415 485 600 440 540 540 415 846 480 474 387 392 440 510
> insulin_out_ind <- which(insulin%in% c(insulin_out))
> insulin_out_ind
 [1]   9  14  27  47  59 112 139 154 187 193 194 221 229 232 236 248 249 259 287 320 371 393
[23] 405 409 410 416 441 487 553 581 585 646 656 661 665 676 696 709 711 716 729 754
> bmi_out <- boxplot.stats(bmi)$out
> bmi_out
[1] 53.2 55.0 67.1 52.3 52.3 52.9 59.4 57.3
> bmi_out_ind <- which(bmi %in% c(bmi_out))
> bmi_out_ind
[1] 121 126 178 194 248 304 446 674
> diabetes_out <- boxplot.stats(diabetes)$out
> diabetes_out
 [1] 2.288 1.441 1.390 1.893 1.781 1.222 1.400 1.321 1.224 2.329 1.318 1.213 1.353 1.224
[15] 1.391 1.476 2.137 1.731 1.268 1.600 2.420 1.251 1.699 1.258 1.282 1.698 1.461 1.292
[29] 1.394
> diabetes_out_ind <- which(diabetes %in% c(diabetes_out))
> diabetes_out_ind
 [1]   5  13  40  46  59 101 148 188 219 229 244 246 260 293 309 331 371 372 384 396 446 535
[23] 594 607 619 622 623 660 662
> age_out <- boxplot.stats(age)$out
> age_out
[1] 69 67 72 81 67 67 70 68 69
> age_out_ind <-which(age %in% c(age_out))
> age_out_ind
[1] 124 364 454 460 490 538 667 675 685
```

Outliers Detection

Leverage Points

An observation is considered to have high leverage if it has a value (or values) for the predictor variables that are much more extreme compared to the rest of the observations in the dataset. We will take a closer look at the observations that have high leverage (if present) since they could have a large impact on the results of a given model.

We are going to follow some steps:

1) Build a regression model
2) Calculate the leverage for each observation
3) Visualize the leverage for each observation

```
> lr_model <- glm( test ~ pregnant + glucose + diastolic + triceps + insulin + bmi + diabetes + age,
 data = DataSet,family = binomial(link='logit'))
> summary(lr_model)

Call:
glm(formula = test ~ pregnant + glucose + diastolic + triceps +
    insulin + bmi + diabetes + age, family = binomial(link = "logit"),
    data = DataSet)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7094  -0.7231  -0.3977   0.7144   2.3764

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.0242679  0.8034183 -11.232  < 2e-16 ***
pregnant     0.1217835  0.0326610   3.729 0.000192 ***
glucose      0.0379551  0.0041634   9.116  < 2e-16 ***
diastolic   -0.0089549  0.0083578  -1.071 0.283971
triceps      0.0068988  0.0119743   0.576 0.564525
insulin     -0.0006137  0.0009525  -0.644 0.519351
bmi          0.0863331  0.0191294   4.513 6.39e-06 ***
diabetes     0.8389681  0.2965386   2.829 0.004666 **
age          0.0127901  0.0095945   1.333 0.182509
---
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: 712.45  on 759  degrees of freedom
AIC: 730.45

Number of Fisher Scoring iterations: 5
```

1) Build a regression model

First we will build a logistic regression model fitting the data

2) Calculate leverage for each observation

Next, we'll use the hatvalues() function to calculate the leverage for each observation in the model (only first twenties observations shown).

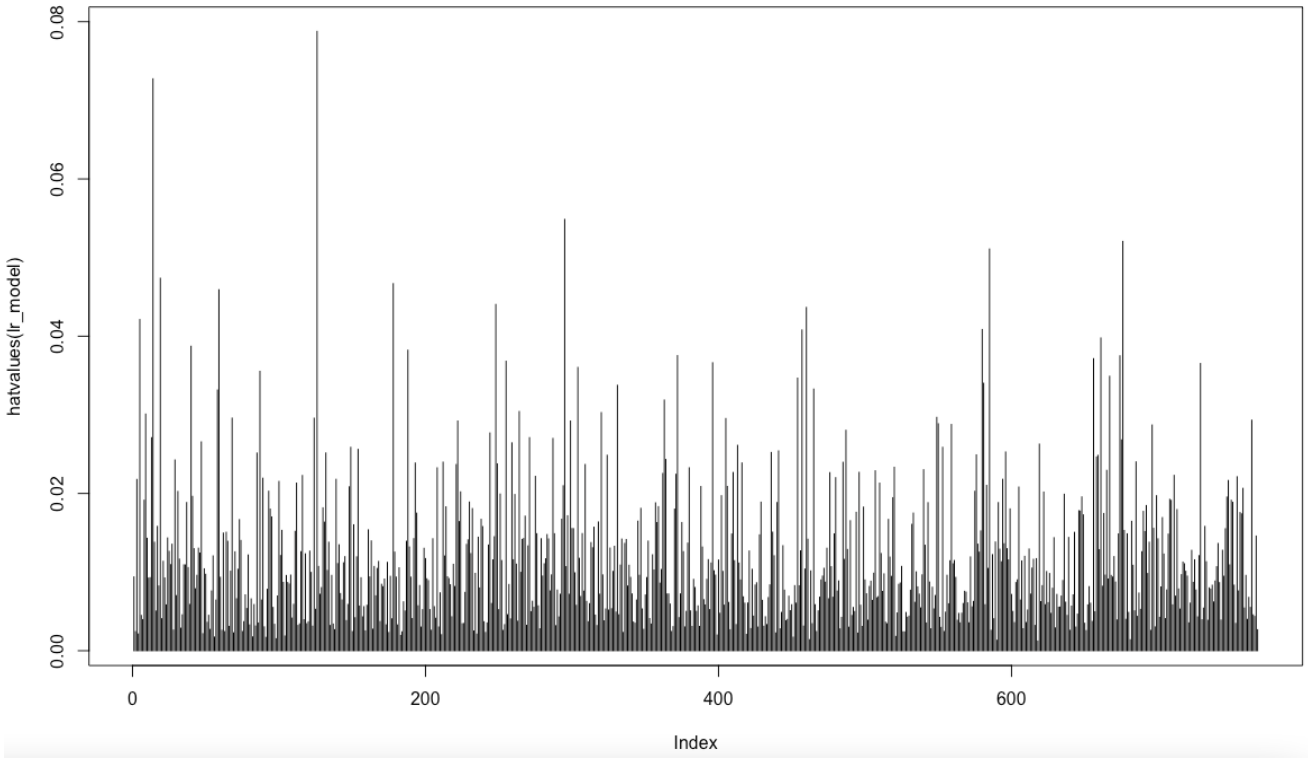
```
> hats <- as.data.frame(hatvalues(lr_model))
> hats
    hatvalues(lr_model)
1           0.009387937
2           0.002429332
3           0.021784597
4           0.002133742
5           0.042119481
6           0.004488471
7           0.003957692
8           0.019143382
9           0.030096240
10          0.014297690
11          0.009235796
12          0.009288698
13          0.027094924
14          0.072734146
15          0.013791217
16          0.005049031
17          0.015816222
18          0.008230647
19          0.047375263
20          0.004073226
```

We have to take a closer look at observations that have a leverage value greater than 2.

An easy way to do this is to sort the observations based on their leverage value, descending:

```
> hats[order(-hats['hatvalues(lr_model)']), ]
  [1] 0.078766747 0.072734146 0.054858231 0.052058198 0.051090185 0.047375263 0.046685504
  [8] 0.045911645 0.044041644 0.043660039 0.042119481 0.040853950 0.040796418 0.039774182
 [15] 0.038724171 0.038223998 0.037529193 0.037506137 0.037142747 0.036815073 0.036625714
 [22] 0.036524792 0.036030613 0.035532815 0.034904774 0.034667989 0.034021044 0.033759804
 [29] 0.033275428 0.033153125 0.031882724 0.030423879 0.030285112 0.030096240 0.029679753
 [36] 0.029584595 0.029574699 0.029530282 0.029323408 0.029211296 0.029209938 0.028864061
 [43] 0.028771948 0.028706778 0.028033307 0.027692485 0.027099133 0.027094924 0.026999150
 [50] 0.026808829 0.026558229 0.026440350 0.026280126 0.026125795 0.025870552 0.025868601
 [57] 0.025608623 0.025412228 0.025273129 0.025218617 0.025152878 0.025144877 0.024909051
 [64] 0.024858819 0.024852143 0.024639194 0.024324543 0.024259118 0.024019368 0.023981952
 [71] 0.023968911 0.023880681 0.023872531 0.023761357 0.023684901 0.023678526 0.023329854
 [78] 0.023267038 0.023261674 0.023020308 0.022938925 0.022884061 0.022687144 0.022672411
 [85] 0.022649687 0.022542067 0.022443453 0.022302229 0.022287314 0.022172137 0.022127576
 [92] 0.022006757 0.021944842 0.021806843 0.021799099 0.021784597 0.021652001 0.021528686
 [99] 0.021313492 0.021312425 0.021023895 0.020979984 0.020892173 0.020878860 0.020849183
[106] 0.020813262 0.020655441 0.020286276 0.020276468 0.020256098 0.020201950 0.020176392
[113] 0.019927252 0.019900498 0.019864613 0.019716320 0.019714897 0.019628638 0.019563165
[120] 0.019541677 0.019458400 0.019254221 0.019143382 0.019133266 0.019123308 0.018910925
[127] 0.018893223 0.018870927 0.018849585 0.018846504 0.018838346 0.018826391 0.018807977
[134] 0.018463391 0.018320972 0.018300093 0.018269528 0.018170673 0.018107988 0.018078381
[141] 0.018035474 0.018009576 0.018005564 0.017948186 0.017813856 0.017753413 0.017717365
[148] 0.017603561 0.017566820 0.017503896 0.017500571 0.017456407 0.017404789 0.017279052
```

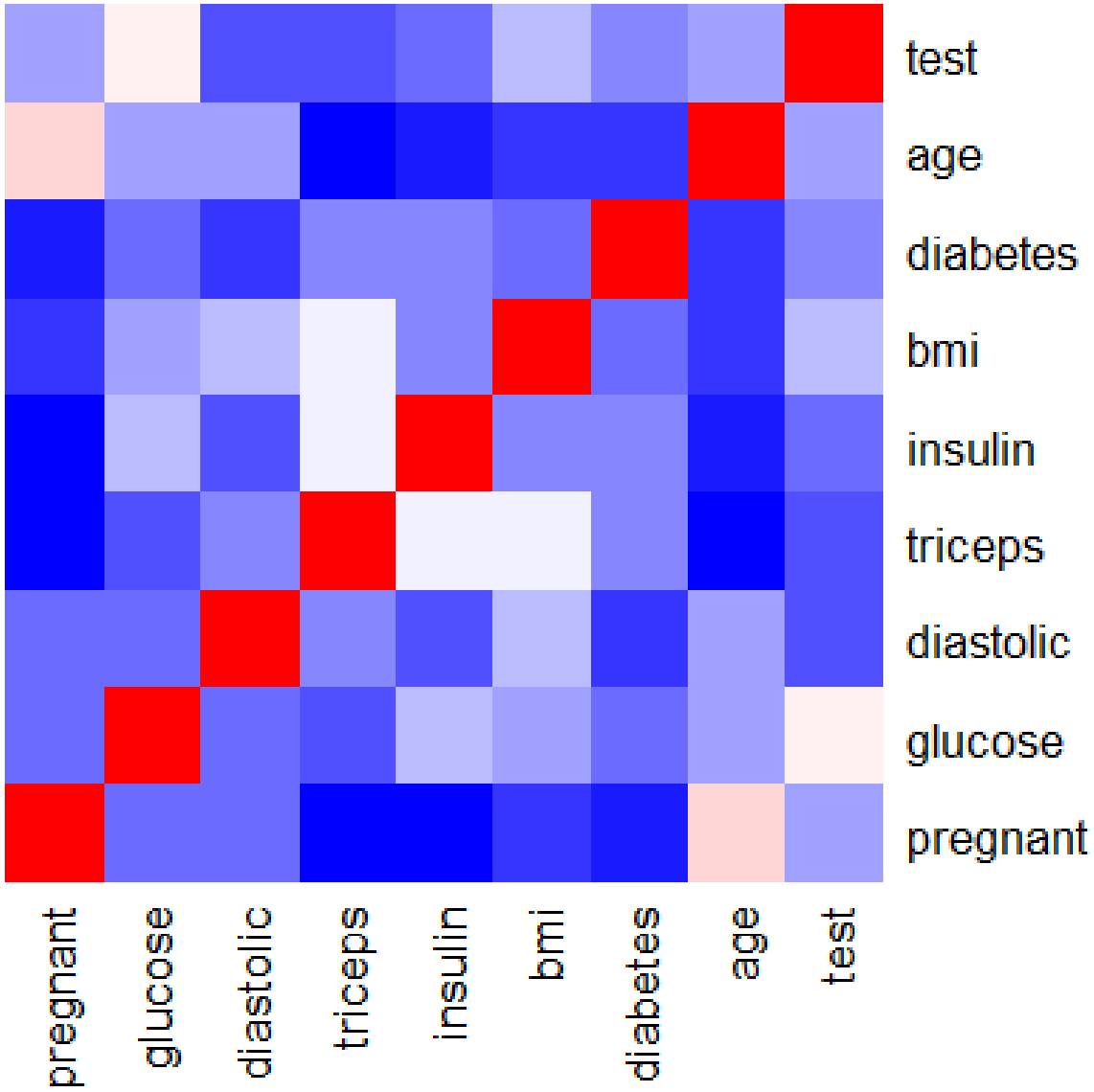
We can see that the largest leverage value is 0.079. Since this isn't greater than 2, we know that none of the observations in our dataset have high leverage.



3) Visualize leverage for each observation

Supervised Learning

Correlation between variables



Binomial regression

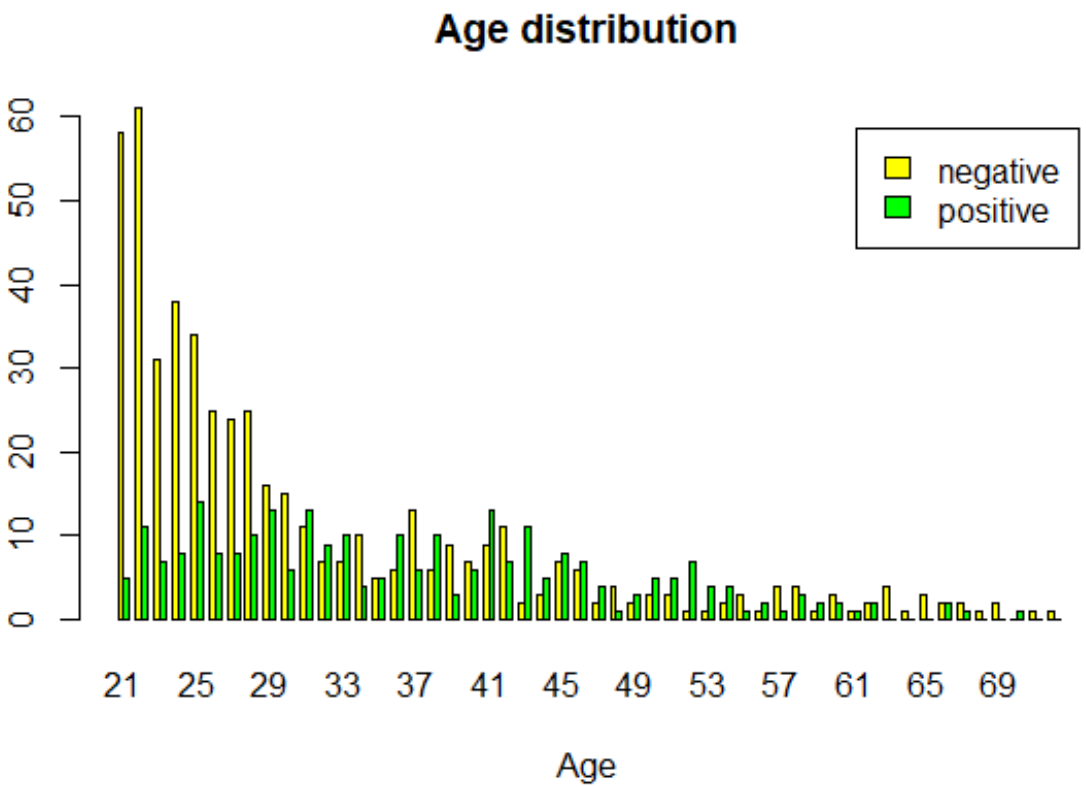
```
> summary(model)

Call:
glm(formula = d$test ~ ., family = "binomial", data = d)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7094  -0.7231  -0.3977   0.7144   2.3764

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.0242679  0.8034183 -11.232  < 2e-16 ***
pregnant     0.1217835  0.0326610   3.729 0.000192 ***
glucose      0.0379551  0.0041634   9.116  < 2e-16 ***
diastolic   -0.0089549  0.0083578  -1.071 0.283971
triceps      0.0068988  0.0119743   0.576 0.564525
insulin     -0.0006137  0.0009525  -0.644 0.519351
bmi          0.0863331  0.0191294   4.513 6.39e-06 ***
diabetes     0.8389681  0.2965386   2.829 0.004666 **
age          0.0127901  0.0095945   1.333 0.182509
---
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: 712.45  on 759  degrees of freedom
AIC: 730.45

Number of Fisher Scoring iterations: 5
```

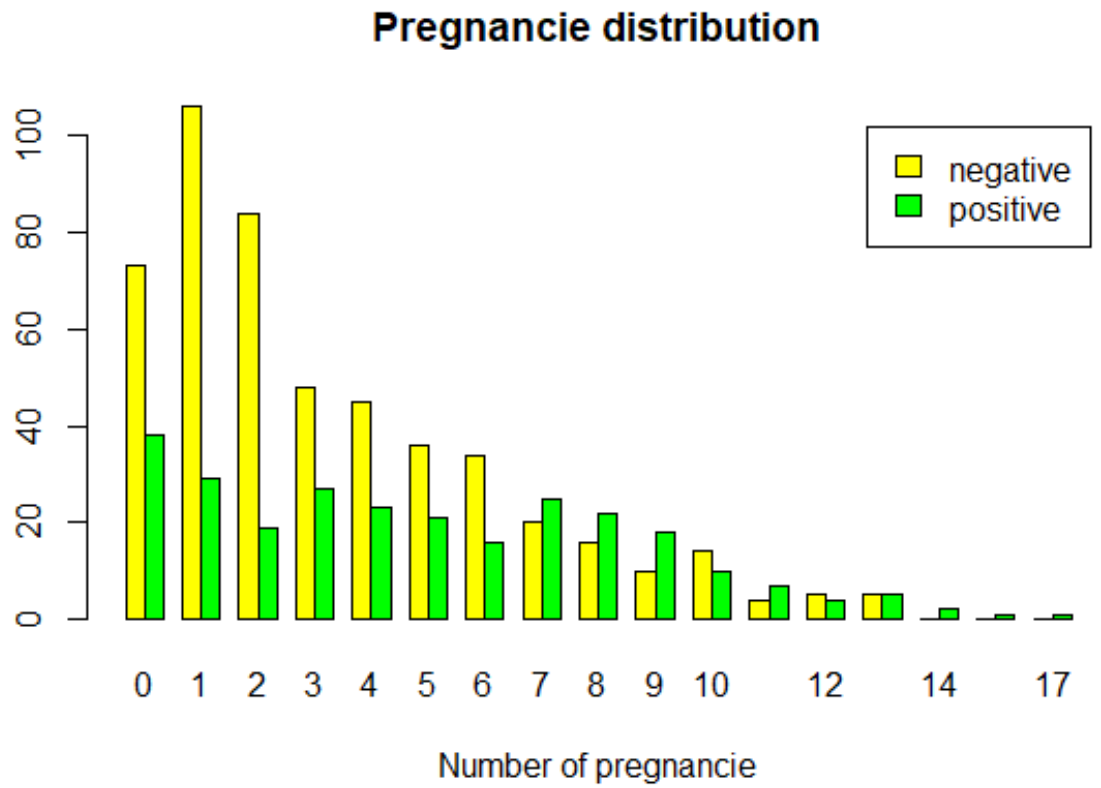
Interactions

```
Call:
glm(formula = d$test ~ . + pregnant * insulin, family = "binomial",
    data = d)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7226  -0.7288  -0.4029   0.7102   2.3734

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -8.9404156  0.8089558 -11.052  < 2e-16 ***
pregnant         0.0888746  0.0501922   1.771  0.07661 .
glucose          0.0381069  0.0041719   9.134  < 2e-16 ***
diastolic       -0.0086339  0.0083669  -1.032  0.30211
triceps          0.0076445  0.0120039   0.637  0.52423
insulin         -0.0014799  0.0013712  -1.079  0.28047
bmi              0.0861386  0.0191294   4.503  6.7e-06 ***
diabetes         0.8524385  0.2962664   2.877  0.00401 **
age              0.0125512  0.0096101   1.306  0.19154
pregnant:insulin 0.0002120  0.0002465   0.860  0.38977
---
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: 711.71  on 758  degrees of freedom
AIC: 731.71
```

```
Call:
glm(formula = d$test ~ . + age * insulin, family = "binomial",
    data = d)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6247  -0.7210  -0.3978   0.7109   2.3661

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.471e+00  8.483e-01  -9.986  < 2e-16 ***
pregnant     1.304e-01  3.334e-02   3.913 9.12e-05 ***
glucose      3.905e-02  4.229e-03   9.235  < 2e-16 ***
diastolic   -8.045e-03  8.395e-03  -0.958  0.33789
triceps      8.373e-03  1.207e-02   0.694  0.48796
insulin     -5.649e-03  2.677e-03  -2.110  0.03482 *
bmi          8.899e-02  1.931e-02   4.609 4.05e-06 ***
diabetes     8.718e-01  2.974e-01   2.931  0.00338 **
age         -1.409e-02  1.656e-02  -0.851  0.39485
insulin:age  1.400e-04  7.084e-05   1.976  0.04819 *
---
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: 708.26  on 758  degrees of freedom
AIC: 728.26
```

Variables distribution:





Collinearity check

Model with no interaction:

```
> sqrt(vif(model))>2
 pregnant   glucose diastolic   triceps   insulin       bmi  diabetes       age
    FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE
>
```

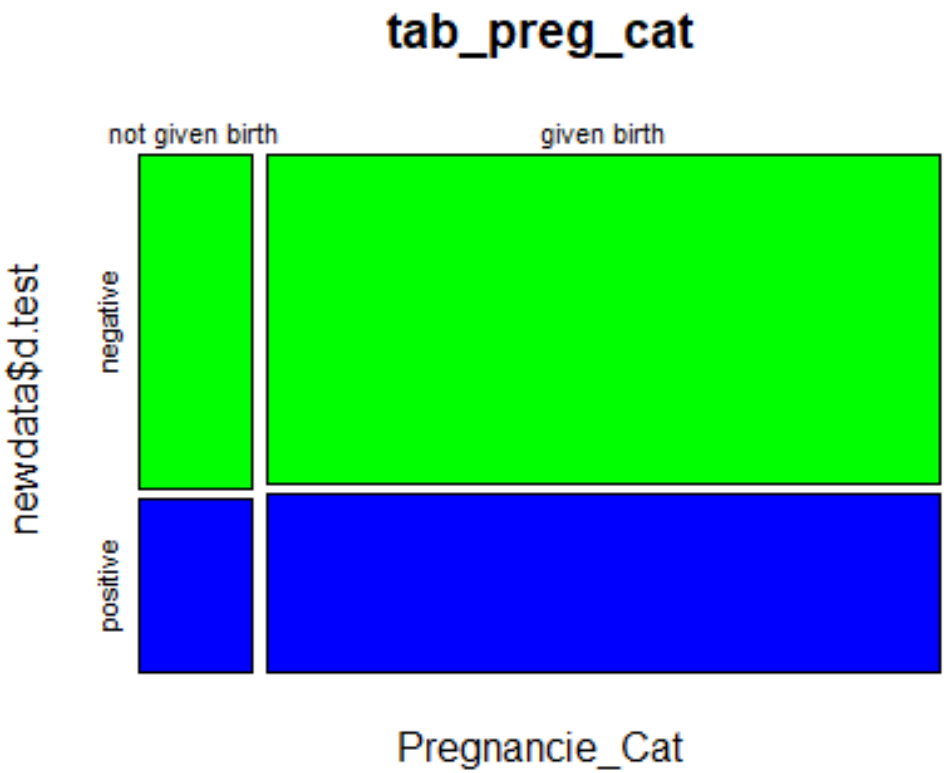
Model with interaction:

```
pregnant     glucose   diastolic     triceps     insulin         bmi    diabetes
   FALSE       FALSE       FALSE       FALSE        TRUE       FALSE       FALSE
     age insulin:age
    TRUE        TRUE
```

Regression with pregnancies as a categorical variable

```
Call:
glm(formula = newdata$d.test ~ ., family = "binomial", data = newdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5842  -0.7339  -0.4141   0.7318   2.3597

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)               -9.2181027  0.8474832 -10.877  < 2e-16 ***
Pregnancie_Catgiven birth  0.1475584  0.2786608   0.530 0.596440
d.glucose                  0.0378607  0.0041383   9.149  < 2e-16 ***
d.diastolic               -0.0073241  0.0082846  -0.884 0.376659
d.triceps                  0.0108879  0.0119762   0.909 0.363282
d.insulin                 -0.0009497  0.0009447  -1.005 0.314761
d.bmi                      0.0817245  0.0189190   4.320 1.56e-05 ***
d.diabetes                 0.7670355  0.2926961   2.621 0.008778 **
d.age                      0.0297979  0.0084975   3.507 0.000454 ***
---
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: 726.52  on 759  degrees of freedom
AIC: 744.52
```

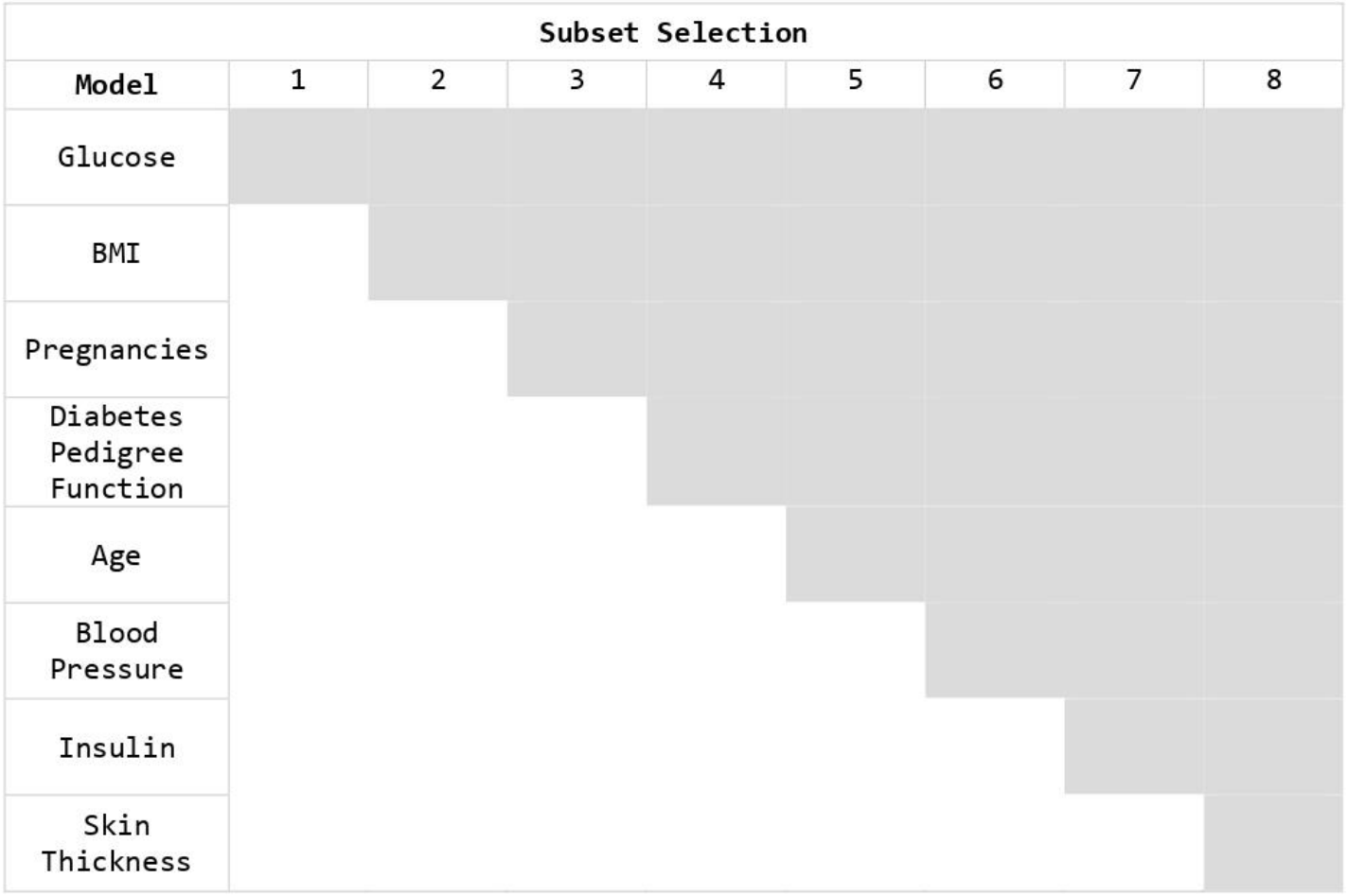
About the regression




```
RR_preg_cat=tab_preg_cat[2,2]/tab_preg_cat[1,2]
RR_preg_cat  ##6.052632
OR=(tab_preg_cat[1,1]*tab_preg_cat[2,2])/(tab_preg_cat[1,2]*tab_preg_cat[2,1])
OR  ##1.034759
```

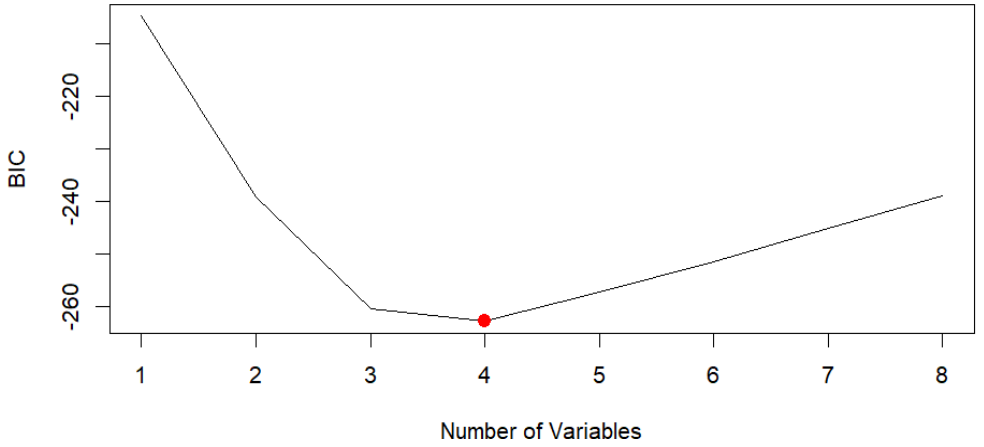
Model Selection

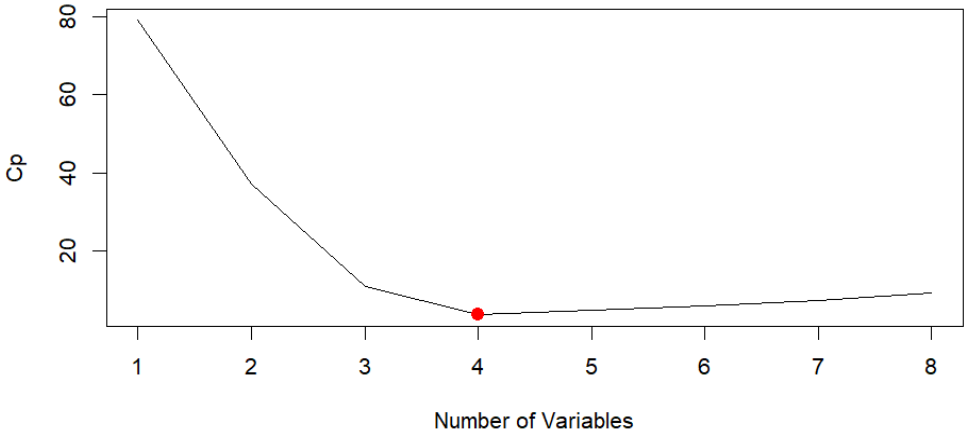
- Now, we wish to identify the variables that best explain the outcome of the test.
- To do so, we have used *subset selection*, through which we have identified a subset of the p variables that we believe to be related to the response.
- As result, we obtain 8 different models, one for each variable: in the first one, *glucose* is the only explanatory variable employed to describe the response; in the last model, all 8 variables are included (same results are obtained using forward selection).
- The results are hereby showed:

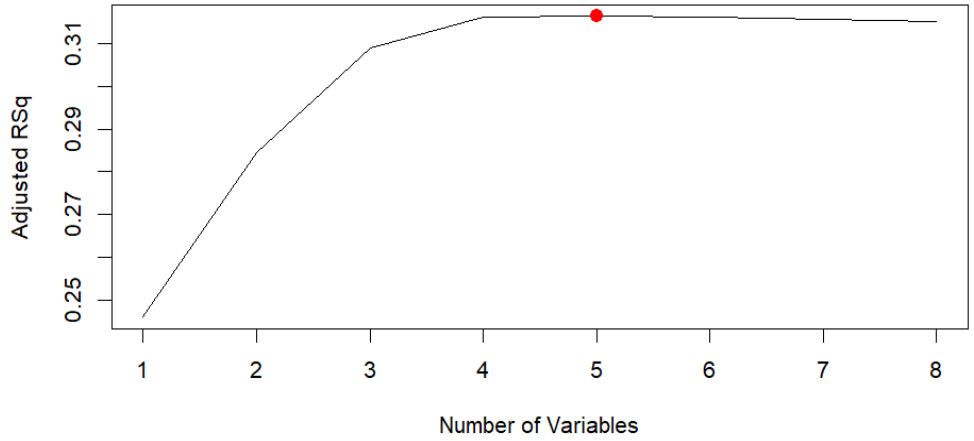


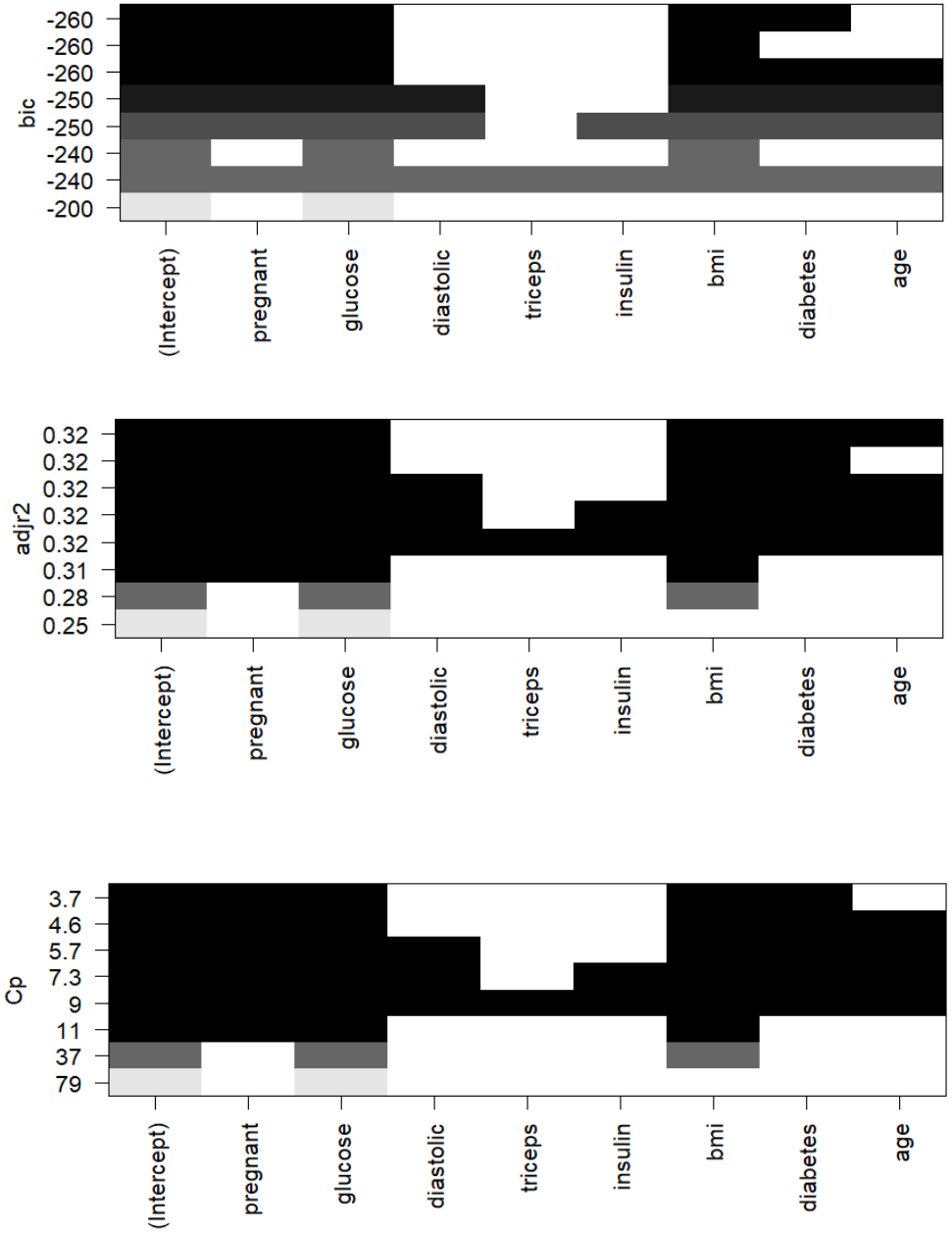
Model Selection

- To select the best model we use:
  - *Mallow's Cp*, and/or *BIC* for which we shall select the model with the lowest values.
  - In our case we'll select the model with 4 variables.
  - Or the *adjusted R-squared*, for which we shall select the model with the highest R-squared.
  - In our case, we'll select the model with 5 variables.









Model Selection

Here, some further representation of the 8 different models, using *black boxes*.

For each value of the adjusted R-squared, BIC and Cp we observe the corresponding model with the included variables.

For instance, the model with the 4 variables (pregnancies, glucose, bmi, and diabetes pedigree function,)is associated with the lowest values of the BIC (-260) and the Cp (3.7).

Model Selection

Finally, we have performed the *K-fold cross* validation to evalute the performance of the models out of the training sample.
We compare the original model with the models made up of 4 and 5 variables, in terms of accuracy and by means of Cohen's Kappa. We set $k = 10$.

- *Accuracy* returns the average accuracy from all training folds.
- *Kappa* returns the rate of agreement between the observed and the expected results.
  Our Kappa lies in the range between [0.46, 0.48] in all three models, which can be considered as *moderate* agreement.

N. of parameters	Accuracy	Kappa
8	0.7694805	0.4680287
4	0.772095	0.469602
5	0.7760595	0.4798639

Model Selection

Based on our results, we could conclude that the models that best describe the response variable are:

- The model with 4 variables: glucose, pregnancies, BMI, and diabetes pedigree function.
- The model with 5 variables: glucose, pregnancies, BMI, diabetes pedigree function, and age.

However, we can observe from the Accuracy and Kappa indexes that the two models do not improve by much the representation of the response variable with respect to the original model.

Regression with the selected model

```
Call:
glm(formula = d$test ~ +d$pregnant + d$bmi + d$glucose + d$diabetes,
    family = "binomial", data = d)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8196  -0.7256  -0.4015   0.7240   2.4343

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.105710   0.698353 -13.039  < 2e-16 ***
d$pregnant   0.141861   0.027541   5.151 2.59e-07 ***
d$bmi        0.085857   0.014609   5.877 4.17e-09 ***
d$glucose    0.037068   0.003489  10.626  < 2e-16 ***
d$diabetes   0.876393   0.294599   2.975  0.00293 **
---
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: 715.42  on 763  degrees of freedom
AIC: 725.42

Number of Fisher Scoring iterations: 5
```

Train and test sample:

```
require(caTools)
set.seed(3)
sample = sample.split(d$test, SplitRatio=0.75)
train_sample = subset(d, sample==TRUE)
train_sample
test_sample = subset(d, sample==FALSE)
nrow(train_sample) ##576
nrow(test_sample)  ##192
##Baseline model
table(d$test)
##Baseline accuracy
baseline_accu <- round(500/nrow(d),2)
baseline_accu  #0.65
##Predict test on training set data
Allvar <- glm(train_sample$test ~ ., data = train_sample, family = binomial)
Predicttrain_sample <- predict(Allvar, type = "response")
summary(Predicttrain_sample)
##Average prediction on each of the two test
tapply(PredictTrain_sample , train_sample$test, mean)
```

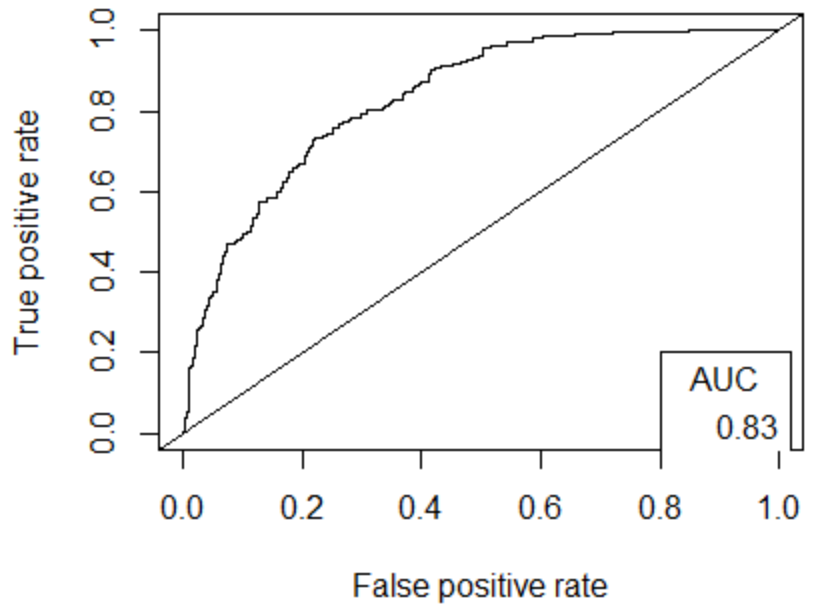
Building confusion matrix

```
##Build the confusion matrix with threshold value of 0.5
thres=table(train_sample$test,Predicttrain_sample>0.5)
thres
# Accuracy
accur <- round(sum(diag(thres))/sum(thres),2) ##0.76
# Mis-classification error rate
MC <- 1-accur  #0.24
sensi <- round(107/(94+107),2) #Sensitivity at 0.5 threshold: 0.54
specifi <- round(329/(329+46),2) #Specificity at 0.5 threshold: 0.89
sprintf("Sensitivity at 0.5 threshold: %s", sensi)
sprintf("Specificity at 0.5 threshold: %s", specifi)
```

```
> thres

          FALSE TRUE
  negative   329   46
  positive    94  107
>
```

ROC curve

```
##ROC Curve
install.packages("ROCR")
library(ROCR)
ROCRpred = prediction(Predicttrain_sample, train_sample$test)
ROCRperf = performance(ROCRpred, "tpr", "fpr")
plot(ROCRperf)
abline(a=0, b=1)
auc_train <- round(as.numeric(performance(ROCRpred, "auc")@y.values),2)
legend(.8, .2, auc_train, title = "AUC", cex=1)
```



Prediction on test set

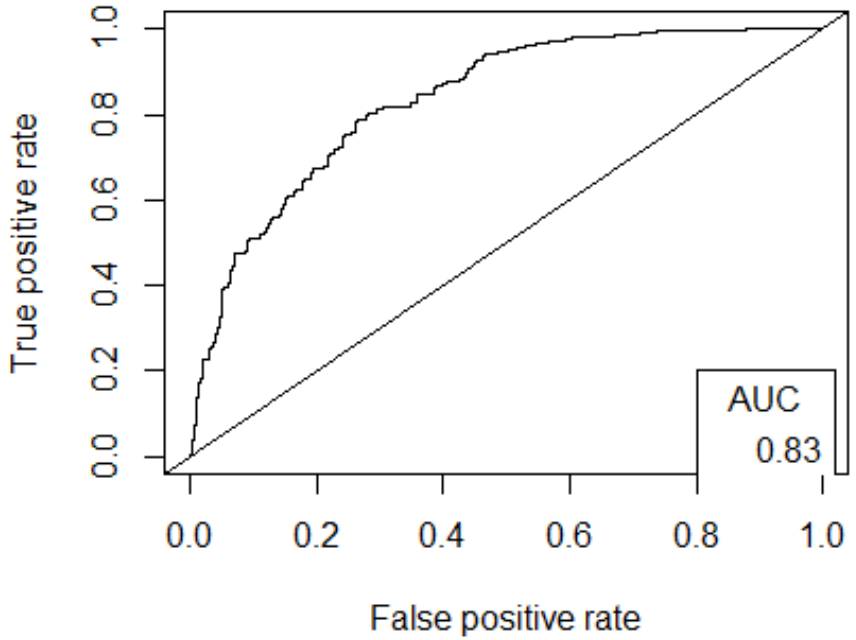
```r
##MAKE PREDICTION ON TEST SET
PredictTest <- predict(Allvar, type = "response", newdata = test_sample)
test_tab <- table(test_sample$test, PredictTest > 0.5)
test_tab
accuracy_test <- round(sum(diag(test_tab))/sum(test_tab),2)
#Accuracy on test set is 0.81"
```

```r
test_tab
```

```
          FALSE TRUE
negative    114   11
positive     25   42
```

ROC curve for the selected model

```
ROCRpred_sel = prediction(Predict_sel_Train_sample, train_sample$test)
ROCRperdf_sel = performance(ROCRpred_sel, "tpr", "fpr")
plot(ROCRperf_sel)
abline(a=0, b=1)
auc_train <- round(as.numeric(performance(ROCRpred_sel, "auc")@y.values),2)
legend(.8, .2, auc_train, title = "AUC", cex=1)
```



Unsupervised learning

Principal component analysis

- What is it
- When we generally use PCA
- What does it produce
- How we did implement it in our model

Variable scales

In order to understand if the variables have the same scale we calculate the mean and the standard deviation

	M	sigma
pregnant	3.85	3.37
glucose	121.64	30.52
diastolic	72.33	12.43
triceps	28.60	10.35
insulin	151.95	114.08
bmi	32.42	6.91
diabetes	0.47	0.33
age	33.24	11.76

Correlation matrix

	pregnant	glucose	diastolic	triceps	insulin	bmi	diabetes	age
pregnant	1.000	0.132	0.207	0.108	0.032	0.023	-0.034	0.544
glucose	0.132	1.000	0.228	0.199	0.570	0.240	0.139	0.270
diastolic	0.207	0.228	1.000	0.210	0.104	0.306	0.004	0.327
triceps	0.108	0.199	0.210	1.000	0.141	0.644	0.131	0.103
insulin	0.032	0.570	0.104	0.141	1.000	0.204	0.065	0.193
bmi	0.023	0.240	0.306	0.644	0.204	1.000	0.157	0.035
diabetes	-0.034	0.139	0.004	0.131	0.065	0.157	1.000	0.034
age	0.544	0.270	0.327	0.103	0.193	0.035	0.034	1.000

Eigen vector and eigenvalues

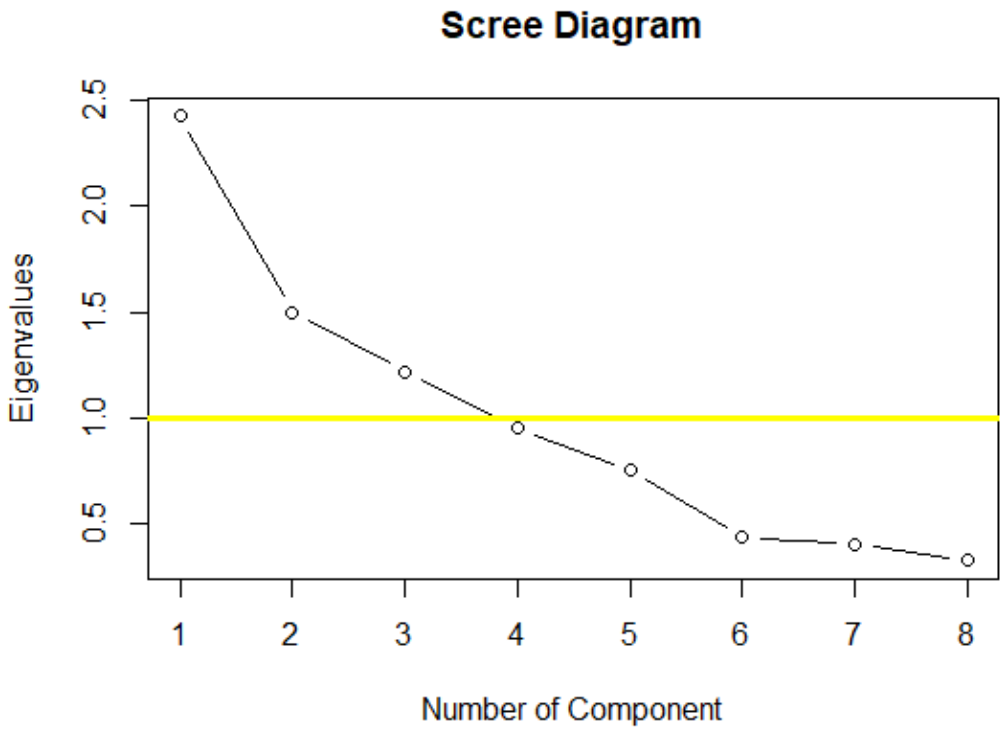
```
> eigen(rho)
eigen() decomposition
$values
[1] 2.4255096 1.4984874 1.2182756 0.9500406 0.7538878 0.4308355 0.3996302 0.3233333

$vectors
           [,1]         [,2]        [,3]        [,4]        [,5]         [,6]       [,7]         [,8]
[1,] -0.2721067  0.545808891  0.24858592 -0.16196061 -0.38093461 -0.466293439  0.4180726 -0.05195131
[2,] -0.4340428 -0.004283783 -0.50368318  0.06244666  0.03830247 -0.536891664 -0.5055775  0.09307929
[3,] -0.3649623  0.129801774  0.27267516  0.19714708  0.81192901 -0.061611921  0.1502255 -0.22614795
[4,] -0.3951136 -0.374454539  0.36387347  0.05412699 -0.36452238  0.100552549 -0.2581076 -0.59893950
[5,] -0.3594352 -0.056194511 -0.61522274  0.18795192 -0.11618147  0.345870545  0.5348070 -0.18622708
[6,] -0.4105478 -0.444679222  0.29219924  0.11125899 -0.08031833  0.002365204  0.1991211  0.69989008
[7,] -0.1430949 -0.246489114 -0.09897634 -0.92473554  0.19668483  0.005572935  0.1072270 -0.06017645
[8,] -0.3606012  0.532311266  0.06385075 -0.15884638 -0.05179168  0.600622877 -0.3784892  0.22498454
```

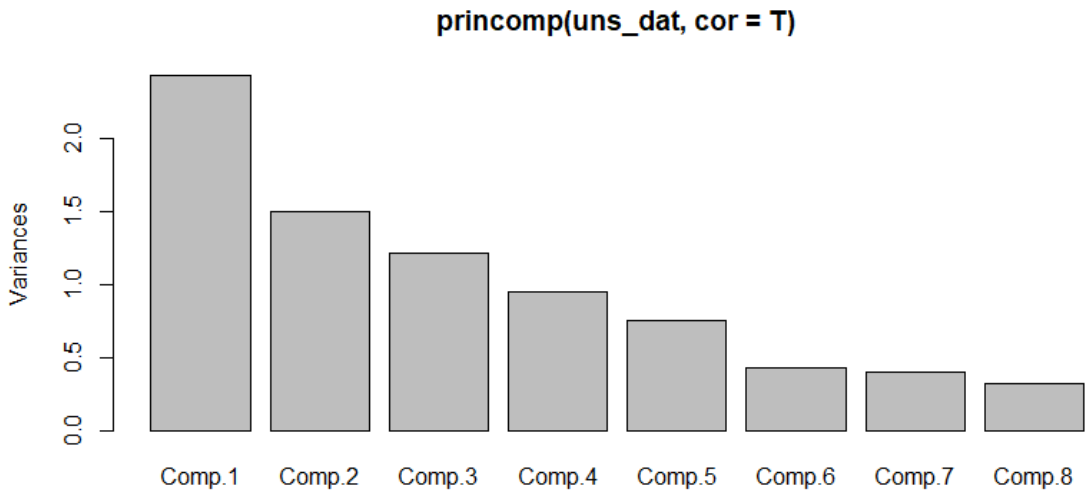
Choice of the components

In the table are shown the eigenvalues, the variance and the percentage of variance explained

```
> tab
      eigenvalues % variance % cum variance
[1,]         2.426      30.319          30.319
[2,]         1.498      18.731          49.050
[3,]         1.218      15.228          64.278
[4,]         0.950      11.876          76.154
[5,]         0.754       9.424          85.578
[6,]         0.431       5.385          90.963
[7,]         0.400       4.995          95.958
[8,]         0.323       4.042         100.000
```

Graphs

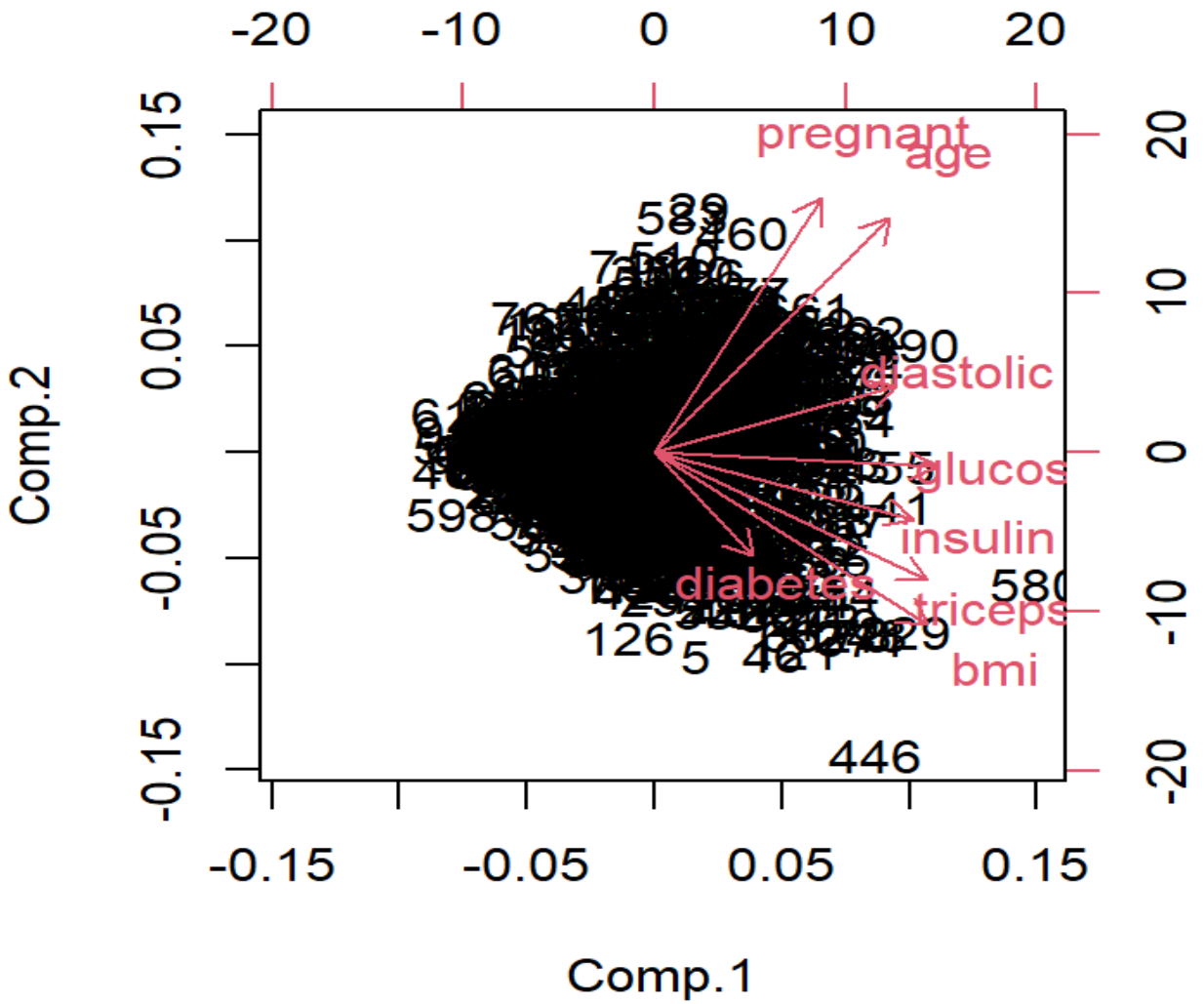

Scree Diagram
Eigenvalues
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7
8
Number of Component


princomp(uns_dat, cor = T)
Variances
0.0
0.5
1.0
1.5
2.0
Comp.1
Comp.2
Comp.3
Comp.4
Comp.5
Comp.6
Comp.7
Comp.8

Communality & scores plot

```
> comunality=comp[,1]^2+comp[,2]^2+comp[,3]^2
> comp=cbind(comp,comunality)
> comp
           Comp1  Comp2  comp3 comunality
pregnant   0.398 -0.693 -0.238   0.695297
glucose    0.685  0.014  0.546   0.767537
diastolic  0.547 -0.210 -0.343   0.460958
triceps    0.647  0.447 -0.373   0.757547
insulin    0.587  0.112  0.659   0.791394
bmi        0.639  0.522 -0.368   0.816229
diabetes   0.217  0.275  0.073   0.128043
age        0.553 -0.663 -0.031   0.746339
```

```
> plot(princomp(uns_dat, cor=T)$scores)
> abline(h=0, v=0)
> biplot(acp)
```



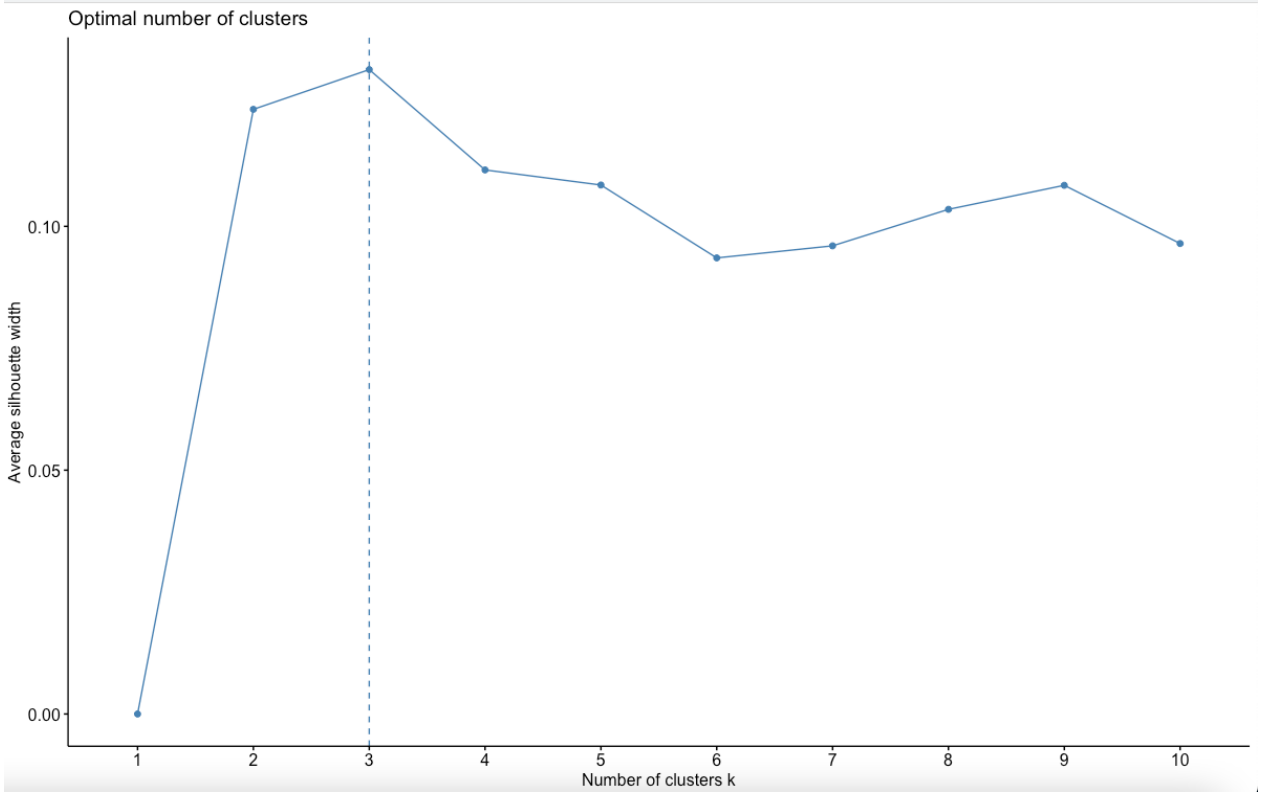
Hiearchical clustering

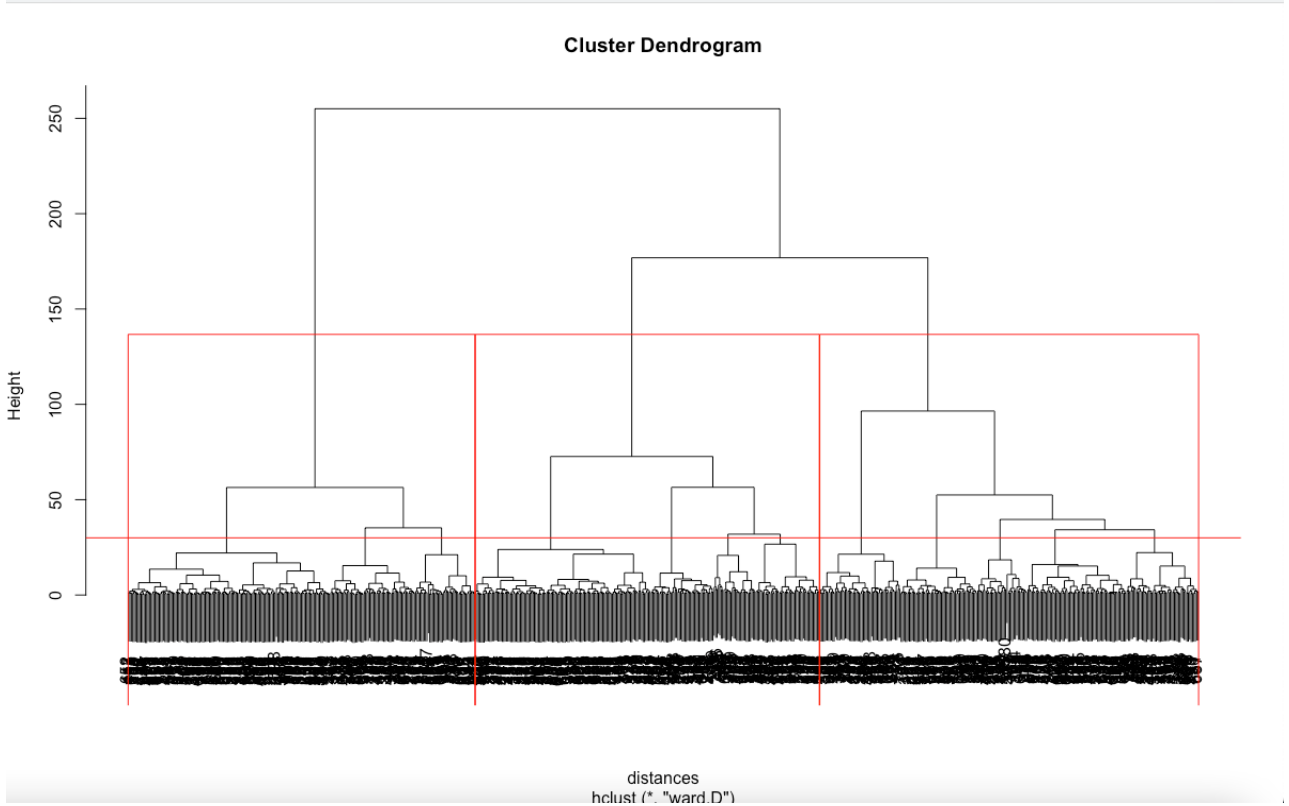
- The hierarchical approach is an alternative to the k-means clustering method that results in a treebase model representation called dendogram.

- We'll start our analysis by calculating the optimal number of clusters and then we'll continue by plotting the dendogram and visualizing which are the observations in every cluster.

Hierarchical clustering

We started with the standardization of the DataSet and then using fviz_nbclust( ) function of R we obtained the optimal number of the clusters that is 3.

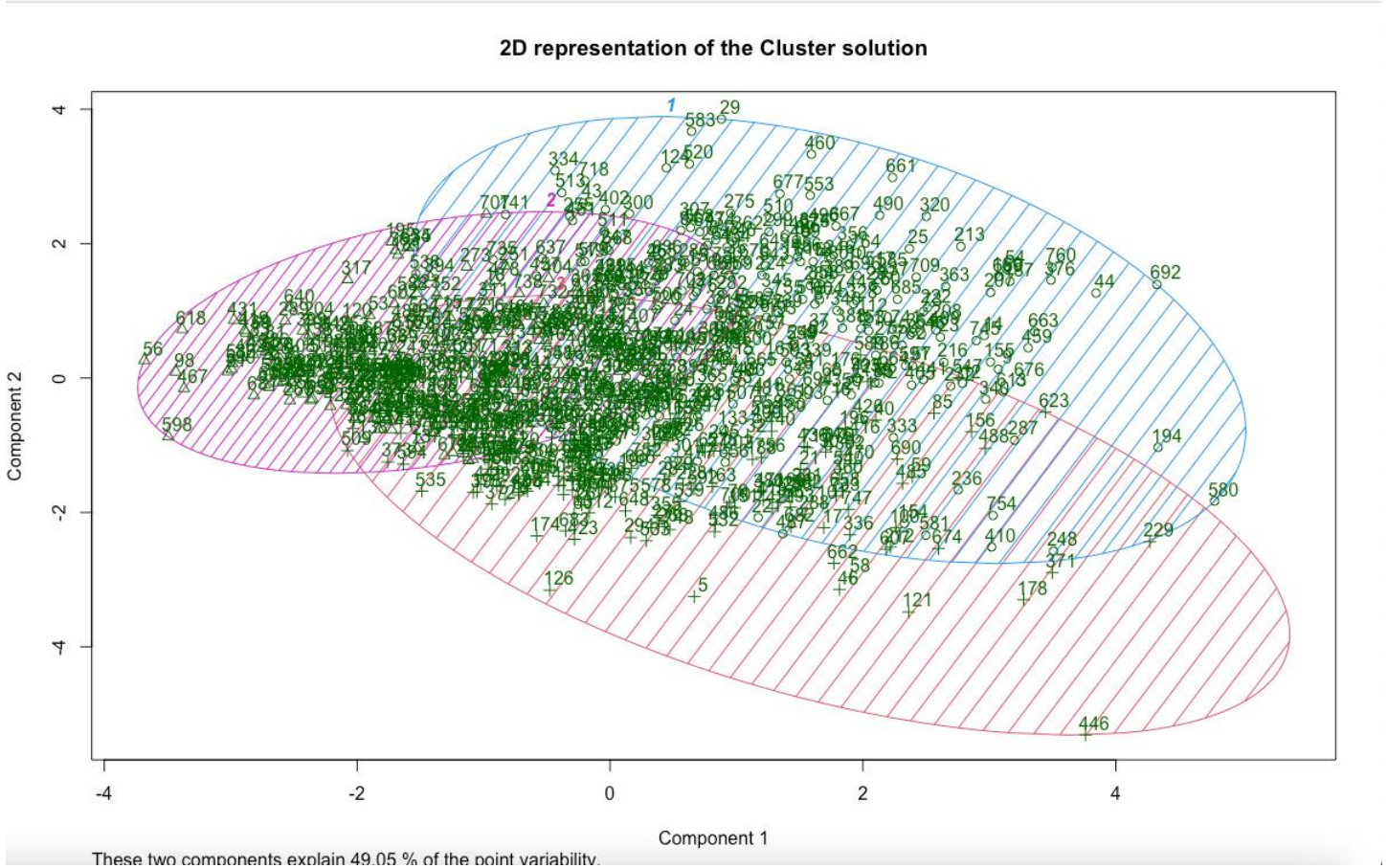




Hierarchical clustering

Then, we plotted the dendogram and to make the division of the clusters clearer we draw rectangles around the branches of the dendrogram to highlight them.

Hierarchical clustering



We can now visulize the observations divided into their respective clusters by this bivariate clusters plot.

Thank you for your attention

The French Fries:

- Di Giovanni Alessia
- Leo Folliero Alessia
- Ruggiero Luisa
- Taheri Mahnaz
- Vasilev Doina