Glucose project

Van Nhut Ho

16/06/2020

Contents

1 EXECUTIVE SUMMARY The aim of this project is to show if the age and the weight influence on the glycemia (glucose). The Glycemia refers to the concentration of sugar or glucose in the blood. In many of the European countries blood glucose or sugar is also measured as millimol per decilitre (mmol/dl). These measurements are referred to as the SI units. If the blood glucose level above 6.1 mmol/l or 1.10 g/l (hyperglycemia) and less than 3.5 mmol/l or 0.54 g/l (hypoglycemia). When fasting blood glucose is above 7 mmol/l (1.26 g/l), the diagnosis of diabetes is made.

2 METHODS AND ANALYSIS 2.1 Data Analysis 2.1.1 Dataset exploration In this section, I'll present the methods and the analysis of the data. Before continuing the process, it's really important to discover the variables (names, types, counts, lenght etc.)

Getting descriptive statistics This dataset contains 10 variables and 284 observations.

```
## 'data.frame':    284 obs. of  10 variables:
##  $ id      : int  220 54 168 114 159 74 178 22 241 155 ...
##  $ chol    : int  163 173 200 219 215 170 222 228 165 271 ...
##  $ glyhb   : num  4.31 4.44 3.55 5.23 4.66 ...
##  $ location: Factor w/ 2 levels "Buckingham","Louisa": 1 1 1 1 2 2 1 1 2 1 ...
##  $ age     : int  29 76 40 76 78 41 51 24 22 55 ...
##  $ gender  : Factor w/ 2 levels "female","male": 1 1 1 2 2 1 1 1 1 1 ...
##  $ height  : num  157 155 157 163 165 ...
##  $ weight  : num  44.8 46.2 47.6 47.6 49.4 ...
##  $ waist   : int  30 31 26 29 33 29 28 33 28 30 ...
##  $ age_cat : Factor w/ 3 levels "20-39","40-59",..: 1 3 2 3 3 2 2 1 1 2 ...
```

Type of variable : - id "integer" = unique identification value per patient - chol "integer" = unique identification value of cholesterol per patient - glyhb "numeric" = unique identification value of glycemia per patient - location "factor" = specific city per patient - age "integer" = age of patient - gender "factor" = gender female or male - height "numeric" = unique identification value per patient in cm - weight "numeric" = unique identification value per patient in kg - waist "integer" = unique identification value per patient in cm - age_cat "factor" = age groups 20-39/40-59/60+ years

checking for missing variables This data set has no missing vaule.

```
##       id     chol    glyhb location      age   gender   height   weight
##        0        0        0        0        0        0        0        0
##    waist  age_cat
##        0        0
```

looking closer attributes and data values For all variables (except id, location, gender, age_cat because of nonsens), the mean and the median are close (= "egalitarian" distribution). We can see a large range for the variable chol and weight,.

```
##        id              chol           glyhb             location
##  Min.   :  1.00   Min.   : 78.0   Min.   :3.330   Buckingham:137
##  1st Qu.: 78.75   1st Qu.:174.8   1st Qu.:4.270   Louisa    :147
##  Median :157.50   Median :199.0   Median :4.660
##  Mean   :158.92   Mean   :200.5   Mean   :4.688
##  3rd Qu.:238.25   3rd Qu.:223.2   3rd Qu.:5.093
##  Max.   :318.00   Max.   :300.0   Max.   :6.210
##       age           gender        height          weight           waist
##  Min.   :20.00   female:171   Min.   :132.1   Min.   : 44.85   Min.   :26.0
##  1st Qu.:32.00   male  :113   1st Qu.:160.0   1st Qu.: 65.69   1st Qu.:33.0
##  Median :41.00                Median :167.6   Median : 77.01   Median :37.0
```

```
##  Mean   :43.82                Mean   :167.5   Mean   : 77.47   Mean   :36.8
##  3rd Qu.:54.25                3rd Qu.:175.3   3rd Qu.: 86.07   3rd Qu.:40.0
##  Max.   :79.00                Max.   :193.0   Max.   :116.42   Max.   :50.0
##   age_cat
##  20-39:120
##  40-59:109
##  60+  : 55
##
##
##
```

2.1.2 Univariate Plots Section One of the main goals of visualizing the data here is to observe which features are most helpful in predicting of type of glycemia (hyper-hypo glycemia). Now we look closer the attributes and data values for: **glyhb** Mean-Median are close ("egalitarian" distribution) Mean = 4.68 Median = 4.66 Skewness = 0.22 : asymmetry slightly to the right (-1 and +1) (+=to the right and 0=symmetry) coefficient of dyssimetry Kurtosis = -0.3 : flat curve, symmetry relative concentration of observations because < 0 ,flattening coefficient <0=platycurtic; >0=leptocurtic distribution therefore sharper curve=lower flattening Shape of the curve = Skewness+Kurtosis Conclusion: almost symmetric

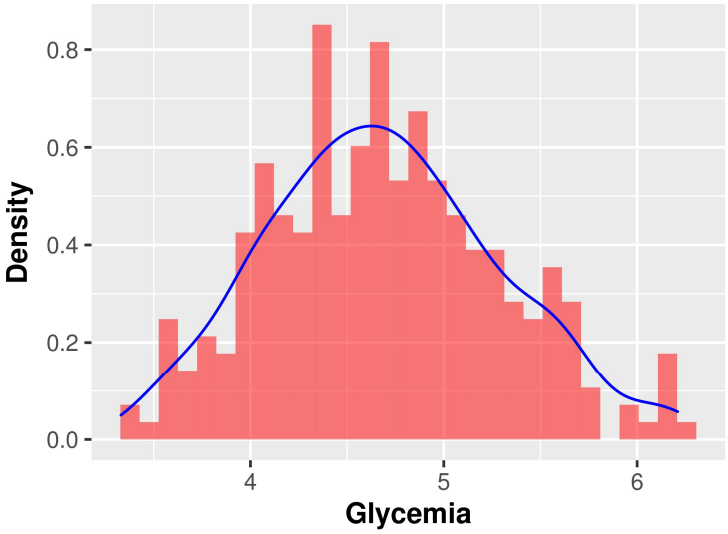
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.330   4.270   4.660   4.688   5.093   6.210
```

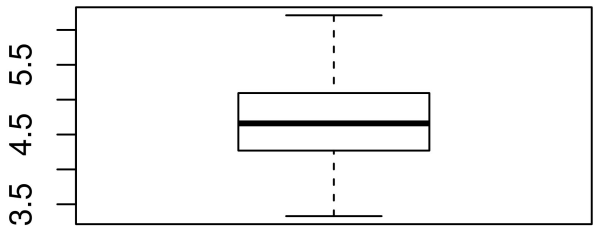
```
## [1] 0.2279742
## attr(,"method")
## [1] "moment"
```

```
## [1] -0.3428237
## attr(,"method")
## [1] "excess"
```

Histogram

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





Boxplot Outlier : none

weight

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   44.85   65.69   77.01   77.47   86.07  116.42
```

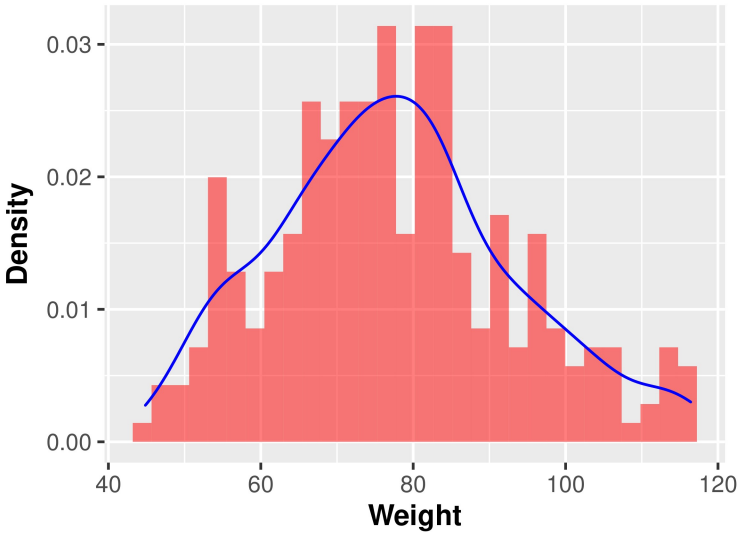
```
## [1] 0.3206615
## attr(,"method")
## [1] "moment"
```

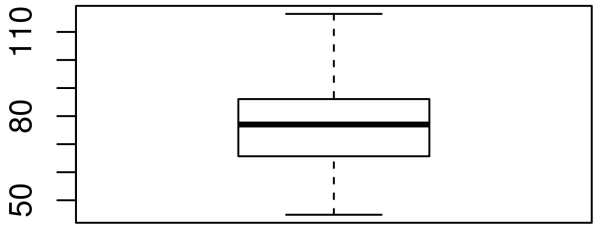
```
## [1] -0.2960106
## attr(,"method")
## [1] "excess"
```

Mean-Median are close (“egalitarian” distribution) Mean = 77.47 Median = 77.01 Skewness = 0.32 : asymmetry slightly to the right (-1 and +1) (+=to the right and 0=symmetry) coefficient of dyssimetry Kurtosis = -0.29 : flat curve, symmetry relative concentration of observations because < 0 ,flattening coefficient <0=platycurtic; >0=leptocurtic distribution therefore sharper curve=lower flattening Shape of the curve = Skewness+Kurtosis Conclusion: almost symmetric

Histogram

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





Boxplot Outlier : none

age

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   20.00   32.00   41.00   43.82   54.25   79.00
```

```
## [1] 0.382244
## attr(,"method")
## [1] "moment"
```

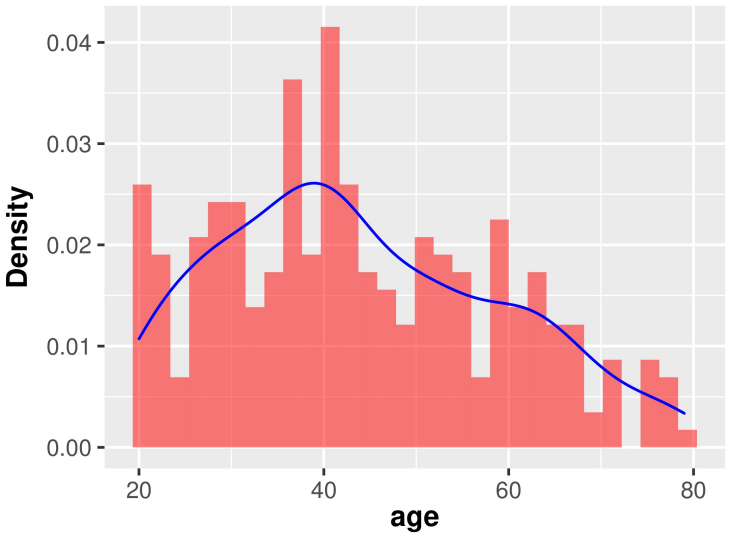
```
## [1] -0.7147143
## attr(,"method")
## [1] "excess"
```

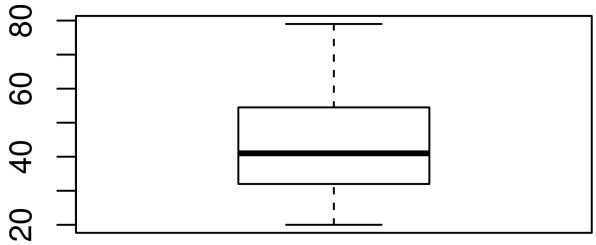
Mean-Median are close ("egalitarian" distribution) Mean = 41 Median = 43.82 Skewness = 0.38 : asymmetry slightly to the right (-1 and +1) (+=to the right and 0=symmetry) coefficient of dyssimetry Kurtosis = -0.71 : flat curve, symmetry relative concentration of observations because < 0 ,flattening coefficient

<0=platycurtic; >0=leptocurtic distribution therefore sharper curve=lower flattening Shape of the curve = Skewness+Kurtosis Conclusion: almost symmetric

Histogram

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





Boxplot Outlier : none

chol

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    78.0   174.8   199.0   200.5   223.2   300.0
```

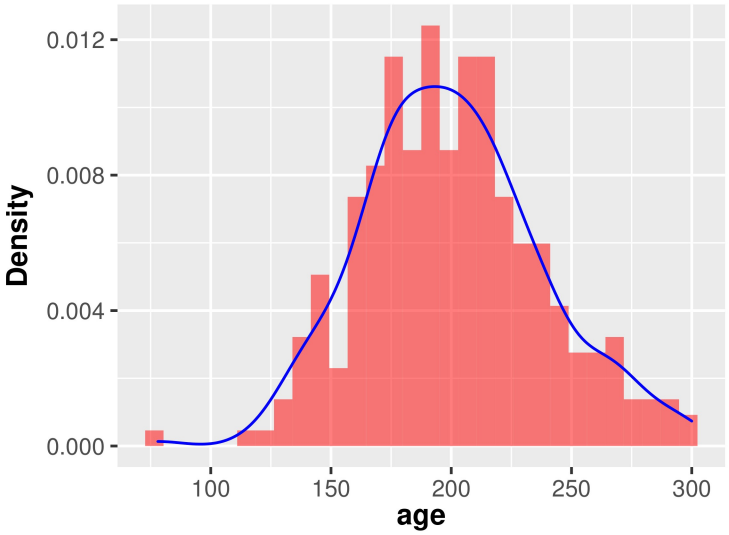
```
## [1] 0.2397935
## attr(,"method")
## [1] "moment"
```

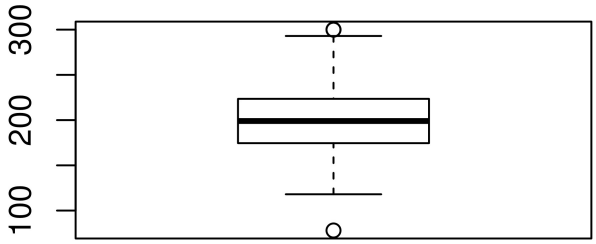
```
## [1] 0.07222744
## attr(,"method")
## [1] "excess"
```

Mean-Median are close ("egalitarian" distribution) Mean = 199.0 Median = 200.5 Skewness = 0.23 : asymmetry slightly to the right (-1 and +1) (+=to the right and 0=symmetry) coefficient of dyssimetry Kurtosis = 0.07 : weak flat curve, symmetry relative concentration of observations because = 0 ,flattening coefficient <0=platycurtic; >0=leptocurtic distribution therefore sharper curve=lower flattening Shape of the curve = Skewness+Kurtosis Conclusion: almost symmetric

Histogram

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





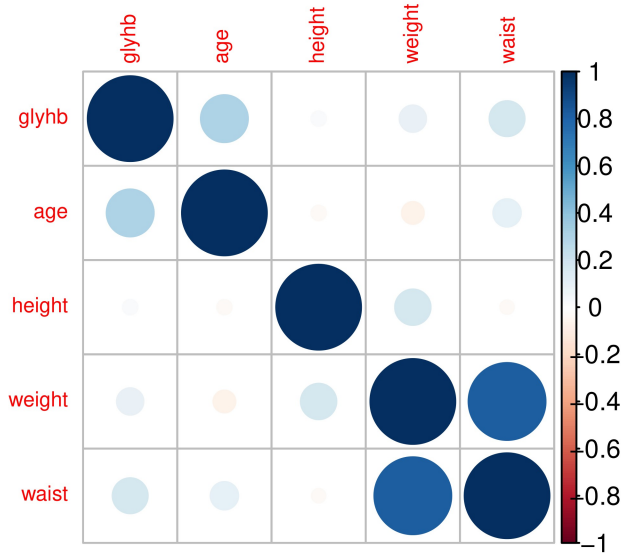
Boxplot Outlier : 1 Intlier : 1

2.1.3 Bivariate/Multivariate Analysis Correlation Plot with numeric variables: We are now interested in how the 6 predictors relate to each other. To see bivariate relationships among these predictors, we

calculate correlations between them. Correlations tell us: - whether this relationship is positive or negative - the strength of the relationship.

Value of r: Strength of relationship: -1.0 to -0.5 or 1.0 to 0.5 Strong -0.5 to -0.3 or 0.3 to 0.5 Moderate -0.3 to -0.1 or 0.1 to 0.3 Weak -0.1 to 0.1 None or very weak

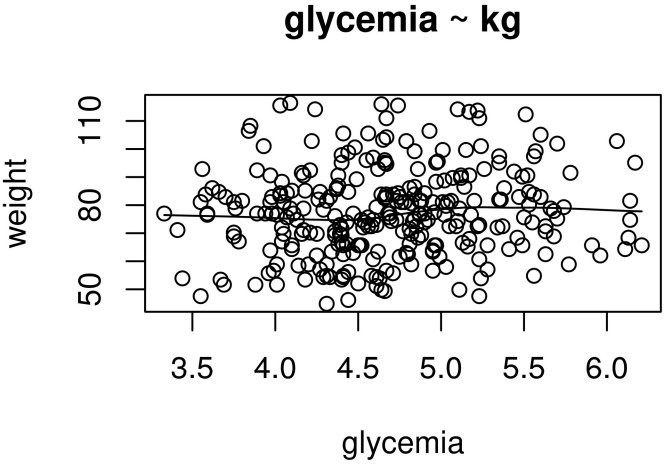
Calculate collinearity Fellowing this corrplot, the variable glyhb has a bivariate relationships with the variable age, weight and waist. Note there is "no" relation between glyhb and height.



Linear regression Now let's have a quantitative score throught a scatter plots and calculation that can help visualize any linear relationships between the dependent (response) variable and independent (predictor) variables.

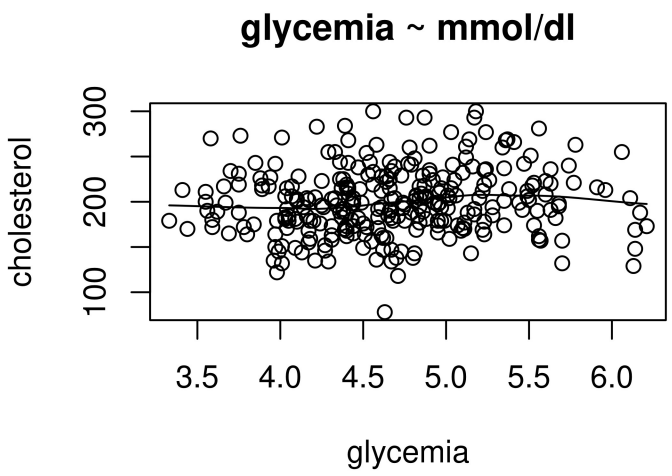
Linear regression between glycemia and weight: Following this score and the scatter plot, there is a weak positive linear relationships between these variables.

```
## [1] 0.09843643
```



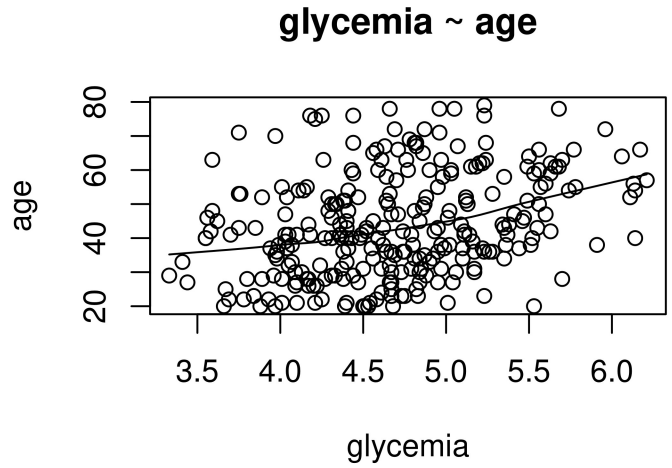
Linear regression between glycemia and cholesterol: As we can see, there is a weak positive linear regression.

```
## [1] 0.1131798
```



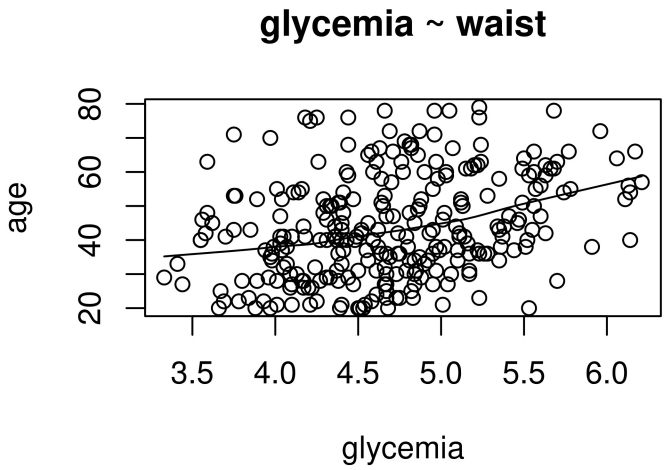
Linear regression between glyemia and age: Here we see a slight positive correlation between these variables.

```
## [1] 0.307663
```



Linear regression between glyemia and wast: The score and the scatter plot show us a slight positive correlation between these variables.

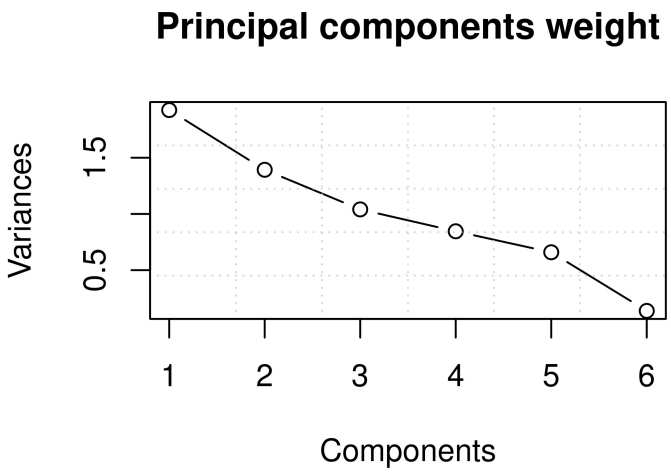
```
## [1] 0.1715594
```


glycemia ~ waist
age
20
40
60
80
3.5
4.0
4.5
5.0
5.5
6.0
glycemia

3 RESULTS*

3.1 Principal Components Analysis (PCA) transform Principal component analysis is a statistical technique that is used to analyze the interrelationships among a large number of variables and to explain these variables in terms of a smaller number of variables (called principal components) with a minimum loss of information. It constructs a set of orthogonal (non-collinear, uncorrelated, independent) variables and is used for making predictive models

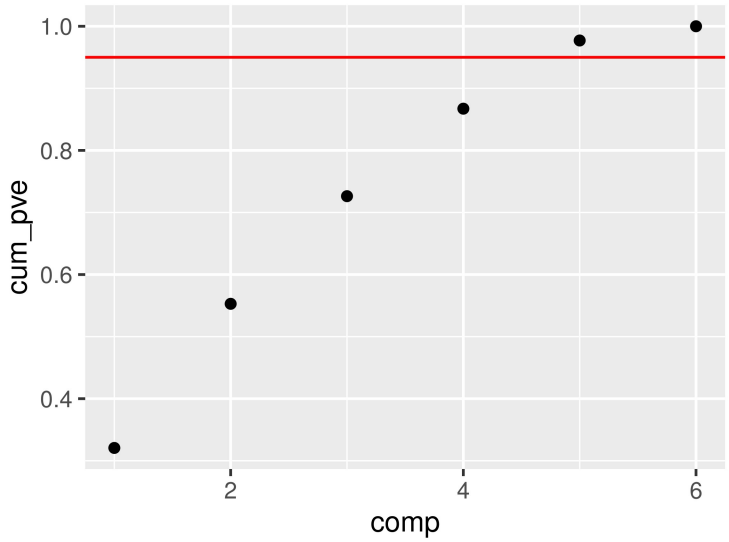
This plot show us the increase of components(variables) the decrease of variances:



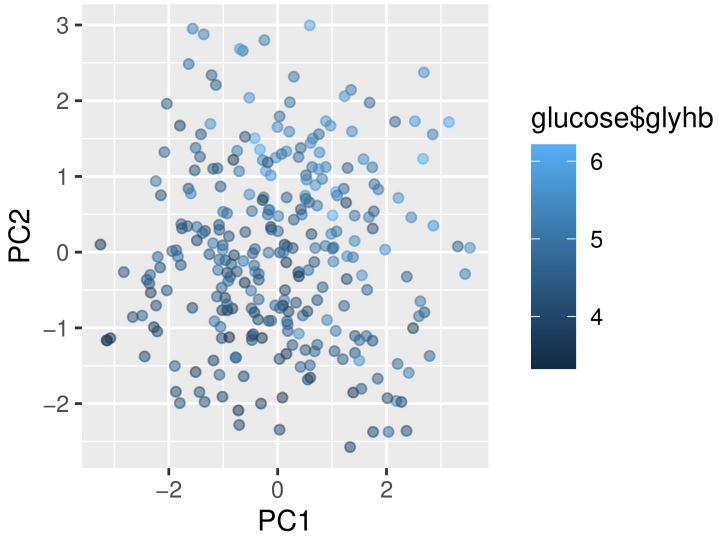

```
## Importance of components:
##                            PC1    PC2    PC3    PC4    PC5     PC6
## Standard deviation     1.3871 1.1802 1.0201 0.9196 0.8118 0.37086
## Proportion of Variance 0.3207 0.2322 0.1734 0.1410 0.1098 0.02292
## Cumulative Proportion  0.3207 0.5528 0.7263 0.8672 0.9771 1.00000
```

Following these results, the standard deviation of variables are low and indicate that the values tend to be close to the mean (also called the expected value) of the set.

Proportion of variance explained To explain more than 0.95 of the variance, it's required 5 principal components and 5 for 0.99.



The features with highest dimmensions or aligned with the leading principal component are the ones with



highest variance.

3.2 Machine learning Machine learning algorithms build a mathematical model based on sample data (training data), in order to make predictions or decisions making without being explicitly programmed to do so. **Split data into train and test sets** The division of the dataset into two parts makes it possible to check the performance of the learning machine. I will Split the available data into a train set (65%) and a test set (35).

train_set = 187 rows test_set = 97 rows

```
## [1] 187
```

```
## [1] 97
```

Overview train_set and test_set

	id	chol	glyhb	location	age	gender	height	weight	waist	age_cat
1	220	163	4.31	Buckingham	29	female	157.48	44.847	30	20-39
3	168	200	3.55	Buckingham	40	female	157.48	47.565	26	40-59
6	74	170	5.11	Louisa	41	female	154.94	49.830	29	40-59
7	178	222	4.64	Buckingham	51	female	167.64	49.830	28	40-59
8	22	228	4.61	Buckingham	24	female	154.94	51.189	33	20-39
9	241	165	3.69	Louisa	22	female	160.02	51.642	28	20-39

	id	chol	glyhb	location	age	gender	height	weight	waist	age_cat
2	54	173	4.44	Buckingham	76	female	154.94	46.206	31	60+
4	114	219	5.23	Buckingham	76	male	162.56	47.565	29	60+
5	159	215	4.66	Louisa	78	male	165.10	49.377	33	60+
11	245	226	3.88	Louisa	20	female	162.56	51.642	31	20-39
13	23	179	4.18	Buckingham	41	female	182.88	53.454	28	40-59
15	128	168	4.40	Buckingham	33	female	167.64	53.454	29	20-39

3.2.1 Naive Bayes In machine learning, naïve Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong independence assumptions between the features

Average of all glycemia

```
## [1] 4.687746
```

Predict the RMSE on the validation set

Finally a dataframe of result

model	RMSE
Naive Mean-Baseline Model	0.5481574

3.2.2 Model building on training data to predict the glycemia on test data From the model summary, the model p value and predictor's p value are less than the significance level = statistically significant model.

Calculate Akaike Information Criterion (AIC) The AIC is essentially an estimated measure of the quality of each of the available econometric models as they relate to one another for a certain set of data, making it an ideal method for model selection.

AIC between glyhb and weight:

```
##
## Call:
## lm(formula = glyhb ~ weight, data = glucose)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.35602 -0.39505 -0.02877  0.37948  1.56648
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.397030   0.178551  24.626   <2e-16 ***
## weight      0.003753   0.002259   1.661   0.0978 .
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5959 on 282 degrees of freedom
## Multiple R-squared:  0.00969,    Adjusted R-squared:  0.006178
## F-statistic: 2.759 on 1 and 282 DF,  p-value: 0.09781
```

```
##
## Call:
## lm(formula = glyhb ~ age, data = glucose)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33254 -0.39033 -0.04089  0.36364  1.49902
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.151304   0.104419   39.76  < 2e-16 ***
## age         0.012242   0.002255    5.43 1.22e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5698 on 282 degrees of freedom
## Multiple R-squared:  0.09466,    Adjusted R-squared:  0.09145
## F-statistic: 29.48 on 1 and 282 DF,  p-value: 1.218e-07
```

```
##        df      AIC
## lmMod   3 515.8976
## lmMod2  3 490.4217
```

Calculate Bayesian Information Criterion (BIC) The BIC is a variant of AIC with a stronger penalty for including additional variables to the model. BIC between glyhb and weight:

```
##        df      BIC
## lmMod   3 526.8445
## lmMod2  3 501.3686
```

AIC and BIC are small meaning that the econometric models is powerful.

Average prediction error rate Dividing the RSE by the average value of the outcome variable will give us the prediction error rate, which should be as small as possible:

```
## [1] 0.1271167
```

The average prediction error rate is 12.7%.

Calculate prediction accuracy and error rates A simple correlation between the actuals and predicted values can be used as a form of accuracy measure. A higher correlation accuracy implies that the actuals and predicted values have similar directional movement, i.e. when the actuals values increase the predicteds also increase and vice-versa.

```
##    actuals predicteds
## 2     4.44   4.570426
```

```
## 4     5.23   4.575526
## 5     4.66   4.582325
## 11    3.88   4.590825
## 13    4.18   4.597625
## 15    4.40   4.597625
```

min_max_accuracy The result of min max accuracy is hight:

```
## [1] 0.9103795
```

mean absolute percentage deviation The mean absolute percentage deviation is low:

```
## [1] 0.09756926
```

3.2.3 Cross-validation methods

The Validation set Approach When comparing two models, the one that produces the lowest test sample RMSE is the preferred model. Making predictions and computing the R2, RMSE and MAE:

```
##           R2      RMSE       MAE
## 1 0.01978701 0.5791717 0.4689291
```

The RMSE and the MAE are measured in the same scale as the outcome variable. Dividing the RMSE by the average value of the outcome variable will give us the prediction error rate, which should be as small as possible:

```
## [1] 0.1238938
```

Note that, the validation set method is only useful when you have a large data set that can be partitioned. Therefore, the test error rate can be highly variable, depending on which observations are included in the training set and which observations are included in the validation set.

K-fold cross-validation The k-fold cross-validation method evaluates the model performance on different subset of the training data and then calculate the average prediction error rate. The algorithm is as follow: - Randomly split the data set into k-subsets (or k-fold) (for example 5 or 10 subsets) - Reserve one subset and train the model on all other subsets - Test the model on the reserved subset and record the prediction error - Repeat this process until each of the k subsets has served as the test set. - Compute the average of the k recorded errors (cross-validation error) serving as the performance metric for the model. - K-fold cross-validation (CV) is a robust method for estimating the accuracy of a model.

Define training control and train the model

```
## Error in train(glyhb ~ ., data = glucose, method = "lm", trControl = train.control): arguments inuti
```

Summarize the results

```
##
## Call:
## lm(formula = glyhb ~ ., data = train_set)
##
## Coefficients:
##    (Intercept)             id           chol  locationLouisa            age
```

```
##      2.2745534      -0.0008731      -0.0001066       0.0932791       0.0222294
##     gendermale          height          weight           waist    age_cat40-59
##     -0.2044325       0.0077879       0.0003019       0.0118329      -0.2077641
##     age_cat60+
##     -0.2373397
```

Repeated K-fold cross-validation The process of splitting the data into k-folds could be repeatedas many times as wished and called repeated k-fold cross validation. The final model error is taken as the mean error from the number of repeats.

This following example uses 10-fold cross validation with 4 repeats: Define training control and model

```
## Error in train(glyhb ~ ., data = glucose, method = "lm", trControl = train.control2): arguments inut
```

Summarize the results

```
## Error in print(model2): objet 'model2' introuvable
```

Summarize RMSE table This table shows the RMSE results of models builted and trained. As we can see, Naive Mean-Baseline Model has the lowest RMSE score this is due to overtraining. Because the first applying K-fold cross-validation model had the lowest RMSE score.

model	RMSE
Naive Mean-Baseline Model	0.5481574
K-fold cross-validation	0.5791717

Conclusion - The glycemia has a slight positive correlation with waist (0.17) and with age (0.30). - The performant model metrics are powerfull (low AIC,BIC). - The Naive Mean-Baseline Model has a low RMSE (prediction errors) score. - The K-fold Cross-Validation has a low RMSE score.

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18362)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=French_Switzerland.1252  LC_CTYPE=French_Switzerland.1252
## [3] LC_MONETARY=French_Switzerland.1252 LC_NUMERIC=C
## [5] LC_TIME=French_Switzerland.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] mlr_2.17.1        ParamHelpers_1.14 corrplot_0.84     timeDate_3043.102
##  [5] kableExtra_1.1.0  data.table_1.12.8 caret_6.0-86      lattice_0.20-41
##  [9] forcats_0.5.0     stringr_1.4.0     dplyr_1.0.0       purrr_0.3.4
## [13] readr_1.3.1       tidyr_1.1.0       tibble_3.0.1      ggplot2_3.3.1
## [17] tidyverse_1.3.0   foreign_0.8-72
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-148         fs_1.4.1             lubridate_1.7.9
##  [4] webshot_0.5.2        httr_1.4.1           tools_3.6.2
##  [7] backports_1.1.7      R6_2.4.1             rpart_4.1-15
## [10] DBI_1.1.0            colorspace_1.4-1     nnet_7.3-14
## [13] withr_2.2.0          tidyselect_1.1.0     compiler_3.6.2
## [16] parallelMap_1.5.0    cli_2.0.2            rvest_0.3.5
## [19] xml2_1.3.2           labeling_0.3         scales_1.1.1
## [22] checkmate_2.0.0      digest_0.6.25        rmarkdown_2.2
## [25] pkgconfig_2.0.3      htmltools_0.4.0      dbplyr_1.4.4
## [28] rlang_0.4.6          readxl_1.3.1         rstudioapi_0.11
## [31] BBmisc_1.11          farver_2.0.3         generics_0.0.2
## [34] jsonlite_1.6.1       ModelMetrics_1.2.2.2 magrittr_1.5
## [37] Matrix_1.2-18        Rcpp_1.0.4.6         munsell_0.5.0
## [40] fansi_0.4.1          lifecycle_0.2.0      stringi_1.4.6
## [43] pROC_1.16.2          yaml_2.2.1           MASS_7.3-51.6
## [46] plyr_1.8.6           recipes_0.1.12       grid_3.6.2
## [49] blob_1.2.1           parallel_3.6.2       crayon_1.3.4
## [52] haven_2.3.1          splines_3.6.2        hms_0.5.3
## [55] knitr_1.28           pillar_1.4.4         reshape2_1.4.4
## [58] codetools_0.2-16     stats4_3.6.2         fastmatch_1.1-0
## [61] reprex_0.3.0         glue_1.4.1           evaluate_0.14
## [64] modelr_0.1.8         vctrs_0.3.1          foreach_1.5.0
## [67] cellranger_1.1.0     gtable_0.3.0         assertthat_0.2.1
## [70] xfun_0.14            gower_0.2.1          prodlim_2019.11.13
## [73] broom_0.5.6          class_7.3-17         survival_3.1-12
## [76] viridisLite_0.3.0    iterators_1.0.12     lava_1.6.7
## [79] ellipsis_0.3.1       ipred_0.9-9
```