Team Project

José Ignacio Díez Ruiz 100487766
Carlos Roldán Piñero 100484904
Pablo Vidal Fernández 100483912

2022-12-10

Descriptive analysis

Reading the data and setting the NAs

First, we read the data using the `read.csv` function:

```
data <- read.csv("diabetes.csv")
```

The names of the variables are the following ones:

```
colnames(data)
```

```
## [1] "Pregnancies"              "Glucose"
## [3] "BloodPressure"            "SkinThickness"
## [5] "Insulin"                  "BMI"
## [7] "DiabetesPedigreeFunction" "Age"
## [9] "Outcome"
```

We also want to transform the *Outcome* variable to a factor variable. We can do that using the `factor` function:

```
data$Outcome <- factor(data$Outcome, c(0,1))
```

After that, we are going to treat as NAs (Not Available data) all 0 values of the following variables:

- Glucose
- BloodPressure
- SkinThickness
- Insulin
- BMI

In order to do that, we define the following function:

```r
setNAs <- function(data, fields){
  percentage <- list()
  for (field in fields){
    data[[field]][data[[field]] == 0] <- NA
    percentage[[field]] <- 100*sum(is.na(data[[field]]))/nrow(data)
  }
  return (list(data = data, percentage = percentage))
}
```

Once the function is defined, we set the NAs, modify the data and save the NA percentages for each of the five variables:

```r
NAfields    <- c("Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI")
dataNA      <- setNAs(data, NAfields)
data        <- dataNA$data
percentages <- dataNA$percentage
```

Outliers

Esto no es mega correcto no?? Para variables no normales no podemos obtener resultados algo xd??

We define the function `findOutliers`:

```r
findOutliers <- function(data, fields){
  outliers <- list()
  for (field in fields){
    qs  <- quantile(data[[field]], c(0.25, 0.75), na.rm = TRUE)
    iqr <- qs[2] - qs[1]
    lq  <- qs[1] - 1.5*iqr
    hq  <- qs[2] + 1.5*iqr
    outliers[[field]] <- which((data[[field]] < lq) & (data[[field]] > hq))
  }
  return (outliers)
}
```

And we save the outliers in a new variable:

```r
outliers <- findOutliers(data, names(data)[names(data) != "Outcome"])
```

Fill NAs (Predictive Mean Matching)

If we look at the variable **percentages** (that contain the percentage of NAs for each one of the five variables contained in **NAfields**), we can see that the percentage of NAs for both the variable *Insulin* and the variable *SkinThickness* are big (48.7% and 29.56%). Performing imputation for variables with such a big number of NAs is not a good idea, so we are going to remove all instaces with NA values in that variables.

Fortunatelly, there is a overlaping in the instances with NA value in *SkinThickness* and *Insulin*. As we can see in the following chunk of code, all the instances with NA value for the *SkinThickness* also have NA value in the *Insulin* variable:

```
sum(is.na((data[is.na(data$SkinThickness),]$Insulin))) ==
  nrow(data[is.na(data$SkinThickness),])
```

```
## [1] TRUE
```

We keep only the instances with non-NA value for the *Insulin* variable:

```
data <- data[!is.na(data$Insulin),]
```

Finally, we impute the data using the `mice` package with the **Predictive Mean Matching** method (`method = "pmm"`):

```
require(mice)
```

```
## Loading required package: mice
```

```
## Warning: package 'mice' was built under R version 4.2.2
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
dataIm <- mice(data, m = 1, method = "pmm")
```

```
##
##  iter imp variable
##   1   1  Glucose  BMI
##   2   1  Glucose  BMI
##   3   1  Glucose  BMI
##   4   1  Glucose  BMI
##   5   1  Glucose  BMI
```

```
data2 <- complete(dataIm)
```