Assignment 1 - Medical Cost
Statistical Inference and Modelling - SIM
1st Semester 2022

Ander Barrio Campos, Odysseas Kyparissis

2022-11-18

Contents

1 Explanatory Data Analysis - EDA

1.1 Loading Insurance Data

In this part of the report, setting up the working environment and loading of the data into R are taking place. Additionally, a first look at the summary of the raw insurance data set is taken.

```
df <- read.csv("insurance.csv")
summary(df)
```

```
##       age            sex                 bmi           children
##  Min.   :18.00   Length:1338        Min.   :15.96   Min.   :0.000
##  1st Qu.:27.00   Class :character   1st Qu.:26.30   1st Qu.:0.000
##  Median :39.00   Mode  :character   Median :30.40   Median :1.000
##  Mean   :39.21                      Mean   :30.66   Mean   :1.095
##  3rd Qu.:51.00                      3rd Qu.:34.69   3rd Qu.:2.000
##  Max.   :64.00                      Max.   :53.13   Max.   :5.000
##     smoker             region             charges
##  Length:1338        Length:1338        Min.   : 1122
##  Class :character   Class :character   1st Qu.: 4740
##  Mode  :character   Mode  :character   Median : 9382
##                                        Mean   :13270
##                                        3rd Qu.:16640
##                                        Max.   :63770
```

1.2 Data Types

To begin with, the types of the raw variables contained into the data set are being checked.

```
typeof(df$age)
```

```
## [1] "integer"
```

```
typeof(df$sex)
```

```
## [1] "character"
```

```
typeof(df$bmi)
```

```
## [1] "double"
```

```
typeof(df$children)
```

```
## [1] "integer"
```

```
typeof(df$smoker)
```

```
## [1] "character"
```

```
typeof(df$region)
```

```
## [1] "character"
```

```
typeof(df$charges)
```

```
## [1] "double"
```

It is clear, that the data set consists of 4 numerical variables and 3 categorical ones. The numeric variables are the following: age, BMI, children and charges, while the categorical ones are: sex, smoker and region. In the following sections, categorical variables will be transformed into labeled factors, as well as, new derived factors will be produced from the numerical values in order to see their performance on the regression modelling process.

1.3 Checking for Missing Data

To continue with, a check for missing data is conducted on the raw data set. Considering the summary of the data set presented before, there are no NA values in the variables of the data set. The same conclusion is derived when a check is completed for each individual variable.

1.4 Checking for Duplicates

By checking if there are duplicate rows inside the raw data set, the result indicates that a single occurrence of a duplicate exists and its index is equal to 582 as it is shown below.

```
dupli <- duplicated(df)
dupli_ind <- which(dupli)
dupli_ind; length(dupli_ind)
```

```
## [1] 582
```

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

With the following command, a closer look can be taken into the values of the duplicate row.

```
df [dupli_ind,]
```

```
##     age  sex   bmi children smoker    region  charges
## 582  19 male 30.59        0     no northwest 1639.563
```

To continue with the explanatory data analysis, the duplicate row is removed from the raw data set. Finally, a check for the existence of duplicates is taking place which results in a FALSE statement.

```
df<- df[-dupli_ind, ]
any(duplicated(df))
```

```
## [1] FALSE
```

1.5 Creating Factors for Qualitative Variables

In this subsection of EDA, all qualitative variables are transformed into labeled factors. The qualitative variables, as mentioned before, are sex, smoker and region. First of all, the unique values of these 3 variables are presented below:

```
unique(df$sex)
```

```
## [1] "female" "male"
```

```
unique(df$smoker)
```

```
## [1] "yes" "no"
```

```r
unique(df$region)
```

```
## [1] "southwest" "southeast" "northwest" "northeast"
```

The next step includes the creation of the labeled factors based on the unique values of the categorical variables. Following the practice below, in case a categorical variable includes NA values, they will be transformed into zeros, which is an incorrect approach. In this case, once missing values check indicated that there are no missing data, proceeding with this practice does not result in erroneous data.

1.5.1 Sex to Labeled Factor

```r
# 1 - Initialize a variable with all values setted at 0
df$f.sex<-0
# 2 - Change its value for the cases where type of sex equals "male"
df$f.sex[df$sex=="male"]<-1
# 3 - Convering f.sex to labeled factor
df$f.sex<-factor(df$f.sex, labels=c("F","M"))
```

1.5.2 Smoker to Labeled Factor

```r
# 1 - Initialize a variable with all values setted at 0
df$f.smok<-0
# 2 - Change its value for the cases where type of smoker equals "yes"
df$f.smok[df$smoker=="yes"]<-1
# 3 - Convering f.smok to labeled factor
df$f.smok<-factor(df$f.smok, labels=c("No","Yes"))
```

1.5.3 Region to Labeled Factor

```r
# 1 - Initialize a variable with all values setted at 0
df$f.reg<-0
# 2 - Change its value for the cases where type of region equals:
# "southwest", "southeast", "northwest"
df$f.reg[df$region=="southwest"]<-3
df$f.reg[df$region=="southeast"]<-2
df$f.reg[df$region=="northwest"]<-1
# 3 - Convering f.reg to labeled factor
df$f.reg<-factor(df$f.reg,labels=c("NE","NW","SE","SW"))
```

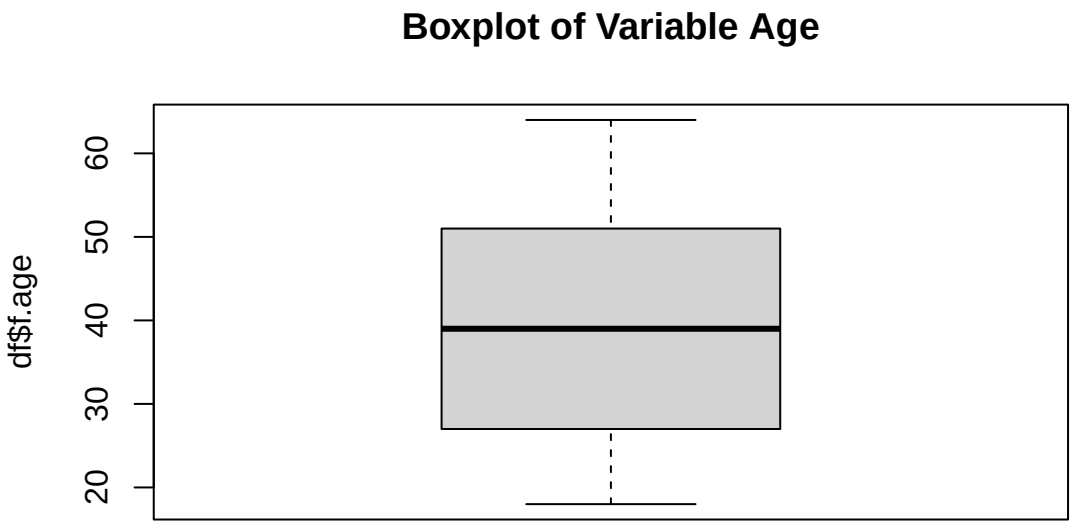
1.6 Creating Factors for Numerical Variables

This step is created in order to extract factors from numerical variables. This approach's goal is to check if some variables are more descriptive as a factor rather than as a numeric feature while training linear models. From the numerical variables of the data set, only age and BMI will be converted to factors, not the target variable (charges). Firstly, the discretization of the variable's values is taking place followed by the assigning of a label for each divided group.

1.6.1 Age to Labeled Factor

```
# 1 - Copying column age to a new column named f.age
df$f.age<-df$age

# 2 - Checking distribution of sample for variable age
# in order to decide how to discretize the values
Boxplot(df$f.age, main= "Boxplot of Variable Age")
```




```
# Once age distribution is almost equally devided
# into its range values, it is decided to discretize
# by using equal intervals
df$f.age<-discretize(df$f.age, method = "interval", breaks = 3,
                     labels = c("Young","Medium", "Old"))
```

The result of the discretization for age is calculated by separating the values of the variable into 3 equally-interval groups with labels: “Young”, “Medium” and “Old” respectively. The interval is equal to 15 years. Thus, group “Young” contain people in ages [18,33], group “Medium” contain individuals with ages [34,48] and finally “Old” group consist of people with ages [49,64].

1.6.2 BMI to Labeled Factor

For BMI, the discretization of the numerical value will be completed by using the labels “Low”, “Normal” and “High”. The values for creating the groups in this step are selected base on the Adult Body Mass Index values from healthcare bibliography.

```
# 1 - Copying column bmi to a new column named f.bmi
df$f.bmi<-df$bmi
# 2- Discretizing directly based on normal BMI values of Bibliography
df$f.bmi<-discretize(df$f.bmi, method = "fixed", breaks = c(-1,18.5,24.9,1000),
                     labels = c("Low","Normal", "High"))
```

1.7 Factor Conversion Check

After checking both manually and by executing commands on the terminal, the conversion of the categorical and numerical variables to factors has been completed correctly. In addition, while the categorical variables sex, region and smoker have been transformed into labeled factors, their old versions of type "chr" are discarded from the data frame. Below is presented the new structure of the data frame.

```
#We will only continue with those factor so we delete the previous variables
df$sex <- NULL #delete sex
df$region <- NULL #delete region
df$smoker <- NULL #delete smoker
str(df)
```

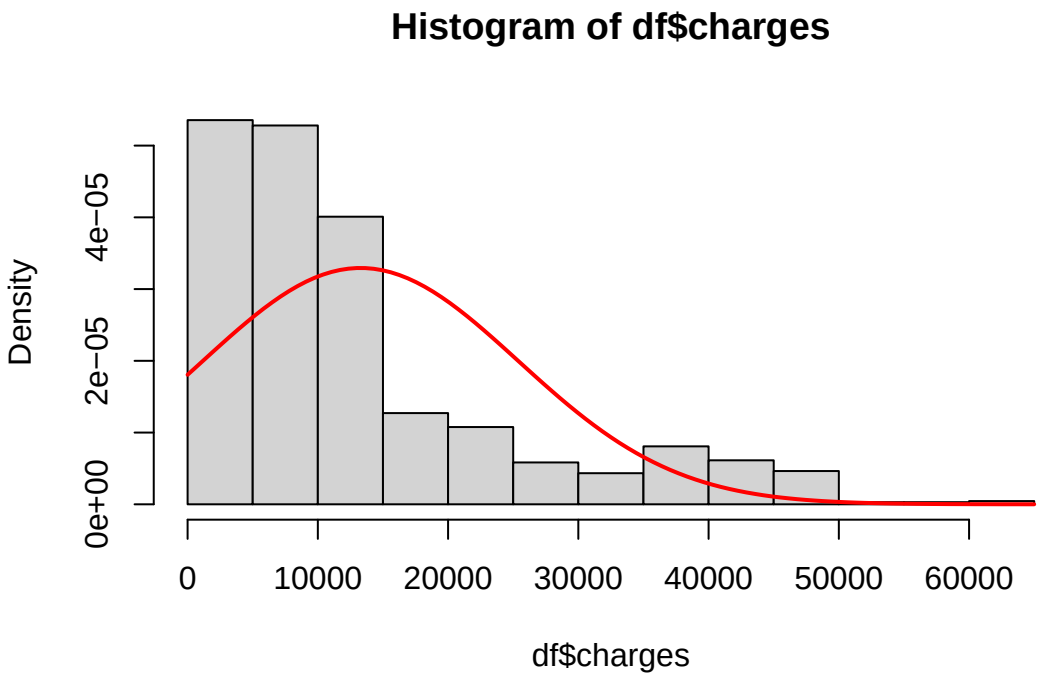
```
## 'data.frame':    1337 obs. of  9 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ charges : num  16885 1726 4449 21984 3867 ...
##  $ f.sex   : Factor w/ 2 levels "F","M": 1 2 2 2 2 1 1 1 2 1 ...
##  $ f.smok  : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ f.reg   : Factor w/ 4 levels "NE","NW","SE",..: 4 3 3 2 2 3 3 2 1 2 ...
##  $ f.age   : Factor w/ 3 levels "Young","Medium",..: 1 1 1 1 1 1 2 2 2 3 ...
##   ..- attr(*, "discretized:breaks")= num [1:4] 18 33.3 48.7 64
##   ..- attr(*, "discretized:method")= chr "interval"
##  $ f.bmi   : Factor w/ 3 levels "Low","Normal",..: 3 3 3 2 3 3 3 3 3 3 ...
##   ..- attr(*, "discretized:breaks")= num [1:4] -1 18.5 24.9 1000
##   ..- attr(*, "discretized:method")= chr "fixed"
```

1.8 Normal Distribution Test for Target Variable (charges)

By taking a look at the histogram of the target variable and the density curve that describe a normal distribution with mean and standard deviation equal to the respective values of the data set, one can understand that the target variable does not follow a normal distribution. In order to be precise, by running the Shapiro test, the result indicate a value less than 0.05. Thus, the null hypothesis can be rejected and conclude that the target variable does not follow a normal distribution.

```
## [1] 13279.12
```

```
## [1] 12110.36
```

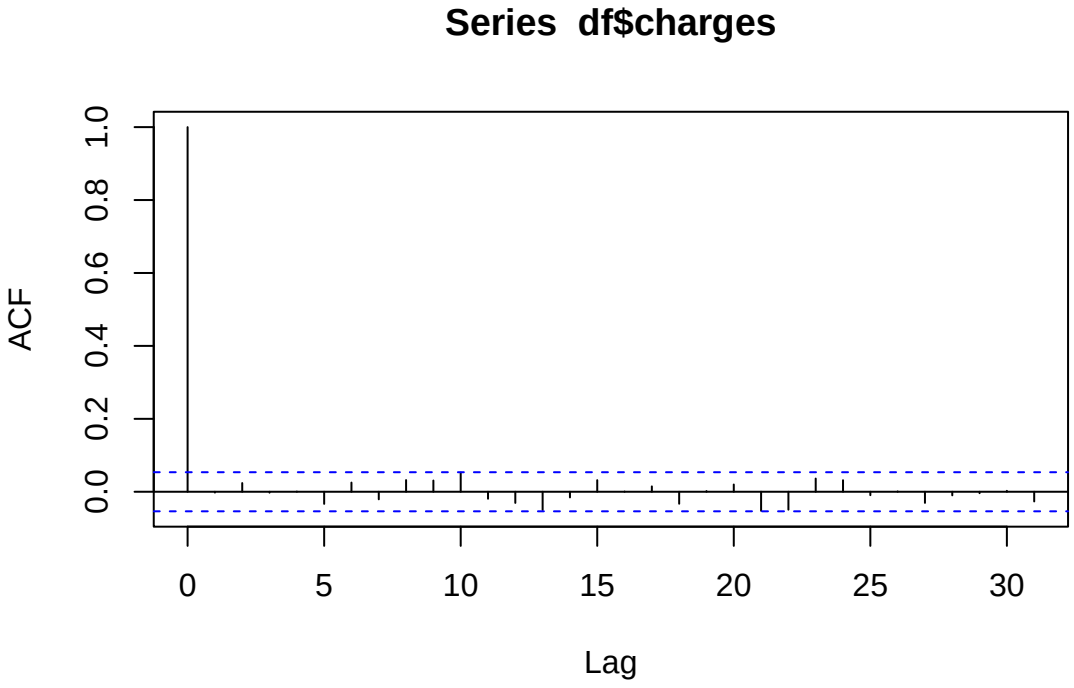



```
shapiro.test(df$charges)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$charges
## W = 0.81476, p-value < 2.2e-16
```

1.9 Serial Correlation

In order to address the serial correlation for the target variable two different approaches were followed. Firstly, the autocorrelation function was used which produces the ACF graph shown below.



From the graph, one can understand that all the vertical lines are inside the two horizontal blue lines except for the first one. The interpretation of this result is that there is no serial correlation for the target variable. Furthermore, for using statistical methods to address the same problem, the Durbin-Watson (DW) test was applied. The result is presented here:

```
dwtest(df$charges~1)
```

```
##
##  Durbin-Watson test
##
## data:  df$charges ~ 1
## DW = 2.0033, p-value = 0.5244
## alternative hypothesis: true autocorrelation is greater than 0
```

Once the resulting p-value is equal to 0.5 approximately, it means that the null hypothesis can not be rejected. The Durbin-Watson test has the null hypothesis that the autocorrelation of the disturbances is 0, thus serial correlation for the target variable is discarded.

1.10 Outliers Detection

In the following subsections both uni-variate and multivariate outliers will be detected and treated.

1.10.1 Univariate Outliers

To start with, in the following subsection the uni-variate outliers will be detected for the numerical variables: age, bmi, children and charges with the respective order. It is crucial to mention here, that only severe outliers were taken into account and not mild ones. Now, concerning variable age, as it was depicted before in the boxplot of the variable, outliers did not exist. The same result is derived after trying to detect outliers using the IQR method, which is implemented by function calcQ.

```
# 1 - AGE:
var_out<-calcQ(df$age)
llout_age<-which((df$age<var_out$souti)|(df$age>var_out$souts))
length(llout_age)
```

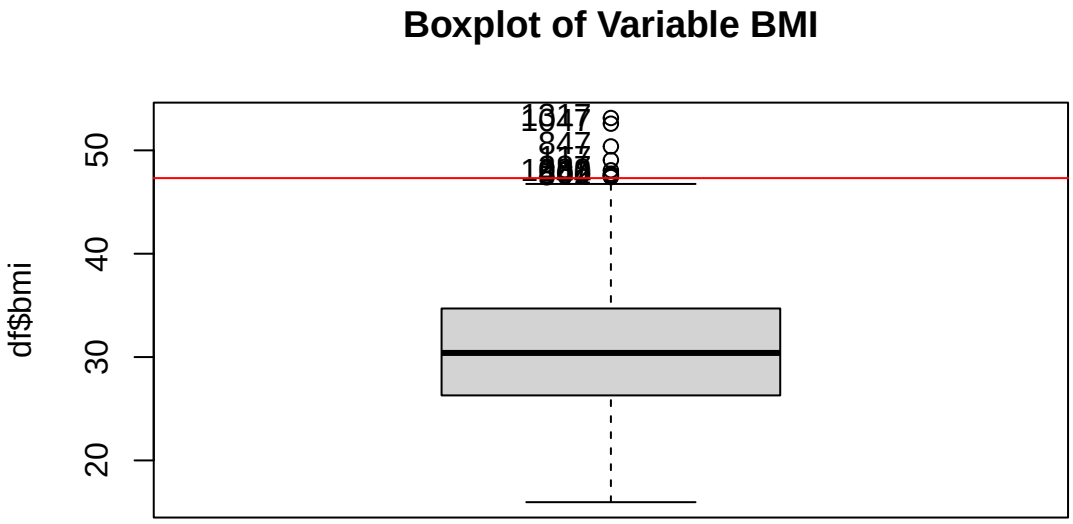
```
## [1] 0
```

This number indicates the number of indexes belonging to outlier observations for variable age, thus while it is zero it means that there are no severe outliers for variable age. Following by, the same approach is followed for variable BMI.

```
# 2 - BMI:
Boxplot(df$bmi, main = "Boxplot of Variable BMI")
```

```
## [1]  117  287  402  544  847  860 1047 1088 1317
```

```
var_out<-calcQ(df$bmi)
abline(h=var_out$souts,col="red")
abline(h=var_out$souti,col="red")
abline(h=var_out$mouts,col="red")
```



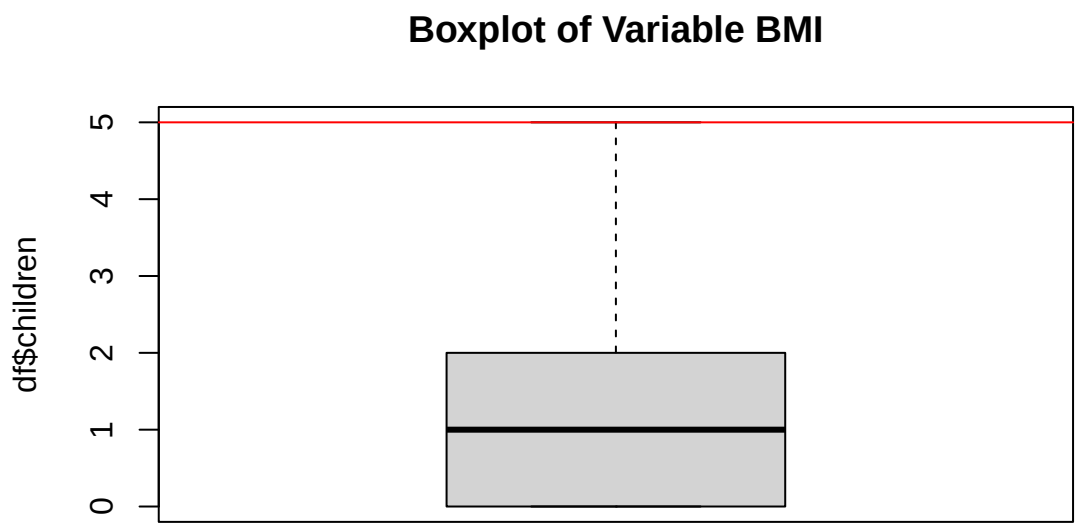

```
llout_bmi<-which((df$bmi<var_out$souti)|(df$bmi>var_out$souts))
length(llout_bmi)
```

```
## [1] 0
```

The results are the same, there are no severe outliers for variable BMI as well, but in this case some mild ones appear but will not be treated. To continue with, same technique is used for variable children.

```
# 2 - CHILDREN:
Boxplot(df$children, main = "Boxplot of Variable BMI")
var_out<-calcQ(df$children)
```

```
abline(h=var_out$souts,col="red")
abline(h=var_out$souti,col="red")
abline(h=var_out$mouts,col="red")
```




```
llout_children<-which((df$children<var_out$souti)|(df$children>var_out$souts))
length(llout_children)
```

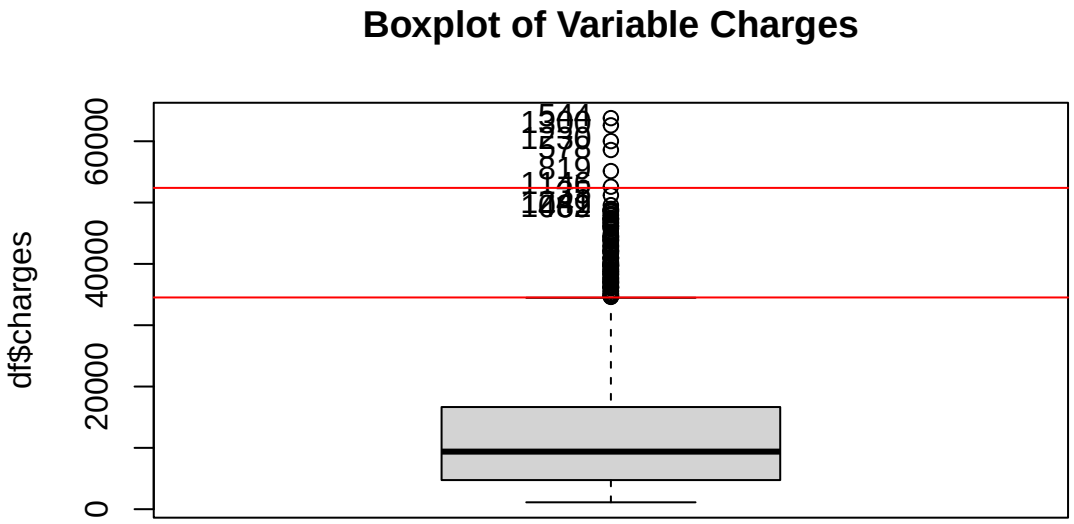
```
## [1] 0
```

And the results are again the same, there are not outliers for this variable as well. Finally, the outlier detection for the target variable is taking place.

```
# 4 - CHARGES:
Boxplot(df$charges, main = "Boxplot of Variable Charges")
```

```
##  [1]  544 1300 1230  578  819 1146   35 1241 1062  489
```

```
var_out<-calcQ(df$charges)
abline(h=var_out$souts,col="red")
abline(h=var_out$souti,col="red")
abline(h=var_out$mouts,col="red")
```




```
llout_charges<-which((df$charges<var_out$souti)|(df$charges>var_out$souts))
length(llout_charges)
```

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

In this case, there are 6 extreme outliers for the target variable, which are presented below.

```
df[llout_charges,]
```

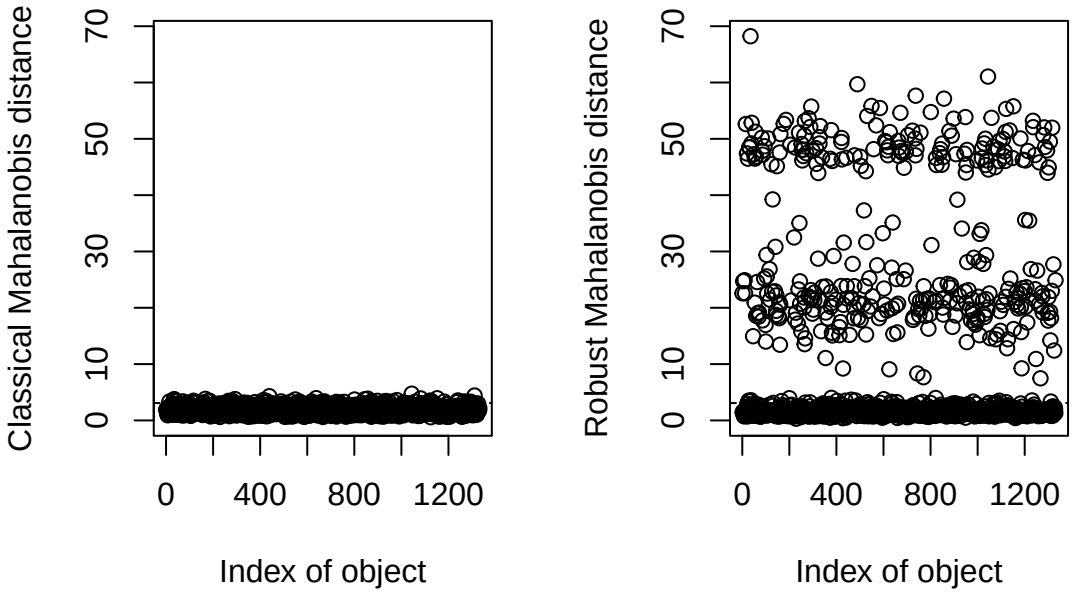
```
##      age    bmi children  charges f.sex f.smok f.reg  f.age f.bmi
## 544   54 47.410        0 63770.43     F    Yes    SE    Old  High
## 578   31 38.095        1 58571.07     F    Yes    NE  Young  High
## 820   33 35.530        0 55135.40     F    Yes    NW  Young  High
## 1147  60 32.800        0 52590.83     M    Yes    SW    Old  High
## 1231  52 34.485        3 60021.40     M    Yes    NW    Old  High
## 1301  45 30.360        0 62592.87     M    Yes    SE Medium  High
```

Due to the fact that the outliers are presented for the target variable, treating them would not be an ideal approach. By checking the values of the data set for those 6 cases, one conclusion that can be drawn is that all of them consern people who smoke and have a high BMI. The values of the other variables for those 6 cases are fluctuating. For this reason, it was decided to remove those observations from our further analysis.

```
df <- df[-llout_charges,]
```

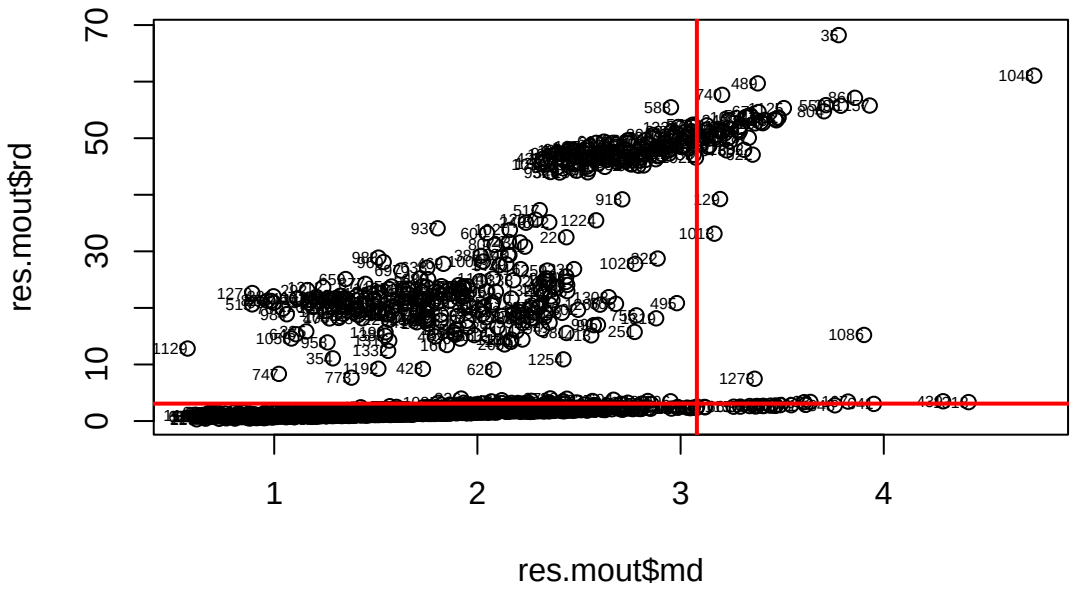
1.10.2 Multivariate Outliers

```
res.mout <- Moutlier( df[ , c(1:4)], quantile = 0.95 )
```




```
#str(res.mout)
#res.mout

par(mfrow=c(1,1))
plot( res.mout$md, res.mout$rd )
text(res.mout$md, res.mout$rd, labels=rownames(df),adj=1, cex=0.5)
abline( h=res.mout$cutoff, lwd=2, col="red")
abline( v=res.mout$cutoff, lwd=2, col="red")
```




```
llmout <- which( ( res.mout$md > res.mout$cutoff )
                 & (res.mout$rd > res.mout$cutoff) );llmout
```

```
##   15   33   35   40   95  110  129  162  167  176  186  252  264  266  282  293
##   15   33   35   40   95  110  129  162  167  176  186  252  264  266  282  293
##  329  331  378  421  422  439  489  531  550  622  669  675  678  726  740  760
##  329  331  378  421  422  439  489  531  549  619  666  672  675  723  737  757
##  804  861  884  902  952  985 1013 1048 1063 1086 1112 1123 1125 1140 1157 1187
##  801  857  880  898  948  981 1009 1044 1059 1082 1108 1119 1121 1136 1152 1182
## 1241 1242 1273 1289 1302 1318
## 1235 1236 1267 1283 1295 1311
```

```
df[llmout,]
```

```
##      age    bmi children   charges f.sex f.smok f.reg  f.age f.bmi
## 15    27 42.130        0 39611.758     M    Yes    SE  Young  High
## 33    19 28.600        5  4687.797     F     No    SW  Young  High
## 35    28 36.400        1 51194.559     M    Yes    SW  Young  High
## 40    60 39.900        0 48173.361     M    Yes    SW    Old  High
## 95    64 31.300        2 47291.055     F    Yes    SW    Old  High
## 110   63 35.090        0 47055.532     M    Yes    SE    Old  High
## 129   32 17.765        2 32734.186     F    Yes    NW  Young   Low
## 162   18 36.850        0 36149.484     F    Yes    SE  Young  High
## 167   20 37.000        5  4830.630     F     No    SW  Young  High
## 176   63 37.700        0 48824.450     F    Yes    SW    Old  High
## 186   36 41.895        3 43753.337     M    Yes    NE Medium  High
## 252   63 32.200        2 47305.305     F    Yes    SW    Old  High
## 264   19 36.955        0 36219.405     M    Yes    NW  Young  High
## 266   46 42.350        3 46151.124     M    Yes    SE Medium  High
## 282   54 40.565        3 48549.178     M    Yes    NE    Old  High
## 293   25 45.540        2 42112.236     M    Yes    SE  Young  High
```

```
## 329   64 33.800        1 47928.030     F    Yes    SW    Old  High
## 331   61 36.385        1 48517.563     F    Yes    NE    Old  High
## 378   24 40.150        0 38126.247     M    Yes    SE  Young  High
## 421   64 33.880        0 46889.261     M    Yes    SE    Old  High
## 422   61 35.860        0 46599.108     M    Yes    SE    Old  High
## 439   52 46.750        5 12592.534     F     No    SE    Old  High
## 489   44 38.060        0 48885.136     F    Yes    SE Medium  High
## 531   57 42.130        1 48675.518     M    Yes    SE    Old  High
## 550   43 46.200        0 45863.205     F    Yes    SE Medium  High
## 622   37 34.100        4 40182.246     M    Yes    SW Medium  High
## 669   62 32.015        0 45710.208     M    Yes    NE    Old  High
## 675   44 43.890        2 46200.985     F    Yes    SE Medium  High
## 678   60 31.350        3 46130.526     M    Yes    NW    Old  High
## 726   30 39.050        3 40932.429     F    Yes    SE  Young  High
## 740   29 35.500        2 44585.456     M    Yes    SW  Young  High
## 760   18 38.170        0 36307.798     M    Yes    SE  Young  High
## 804   18 42.240        0 38792.686     F    Yes    SE  Young  High
## 861   37 47.600        2 46113.511     F    Yes    SW Medium  High
## 884   51 37.050        3 46255.113     F    Yes    NE    Old  High
## 902   60 40.920        0 48673.559     M    Yes    SE    Old  High
## 952   51 42.900        2 47462.894     M    Yes    SE    Old  High
## 985   20 30.115        5  4915.060     M     No    NE  Young  High
## 1013  61 33.330        4 36580.282     F     No    SE    Old  High
## 1048  22 52.580        1 44501.398     M    Yes    SE  Young  High
## 1063  59 41.140        1 48970.248     M    Yes    SE    Old  High
## 1086  39 18.300        5 19023.260     F    Yes    SW Medium   Low
## 1112  38 38.390        3 41949.244     M    Yes    SE Medium  High
## 1123  53 36.860        3 46661.442     F    Yes    NW    Old  High
## 1125  23 42.750        1 40904.200     F    Yes    NE  Young  High
## 1140  19 32.490        0 36898.733     F    Yes    NW  Young  High
## 1157  19 44.880        0 39722.746     M    Yes    SE  Young  High
## 1187  20 35.625        3 37465.344     M    Yes    NW  Young  High
## 1241  52 41.800        2 47269.854     M    Yes    SE    Old  High
## 1242  64 36.960        2 49577.662     M    Yes    SE    Old  High
## 1273  43 25.520        5 14478.330     M     No    SE Medium  High
## 1289  20 39.400        2 38344.566     M    Yes    SW  Young  High
## 1302  62 30.875        3 46718.163     M    Yes    NW    Old  High
## 1318  18 53.130        0  1163.463     M     No    SE  Young  High
```

```
#res.mout$md[llmout]
df$mout <- 0
df$mout[ llmout ] <- 1
df$mout <- factor( df$mout, labels = c("MvOut.No","MvOut.Yes"))
summary(df)
```

```
##       age             bmi           children        charges      f.sex
##  Min.   :18.00   Min.   :15.96   Min.   :0.000   Min.   : 1122   F:659
##  1st Qu.:26.50   1st Qu.:26.22   1st Qu.:0.000   1st Qu.: 4729   M:672
##  Median :39.00   Median :30.30   Median :1.000   Median : 9305
##  Mean   :39.19   Mean   :30.64   Mean   :1.098   Mean   :13074
##  3rd Qu.:51.00   3rd Qu.:34.64   3rd Qu.:2.000   3rd Qu.:16436
```

```
##  Max.   :64.00   Max.   :53.13   Max.   :5.000   Max.   :51195
##  f.smok     f.reg       f.age        f.bmi             mout
##  No :1063   NE:323   Young :520   Low    :  20   MvOut.No :1277
##  Yes: 268   NW:322   Medium:401   Normal: 222   MvOut.Yes:  54
##             SE:362   Old   :410   High   :1089
##             SW:324
##
##
```

1.11 Preliminary Exploratory Analysis

```
res.con<-condes(df,4, proba = 1)
#str(res.con)
#names(res.con)
res.con$quanti
```

```
##          correlation      p.value
## age       0.30062715 3.329401e-29
## bmi       0.18862792 3.988446e-12
## children  0.07587579 5.613221e-03
```

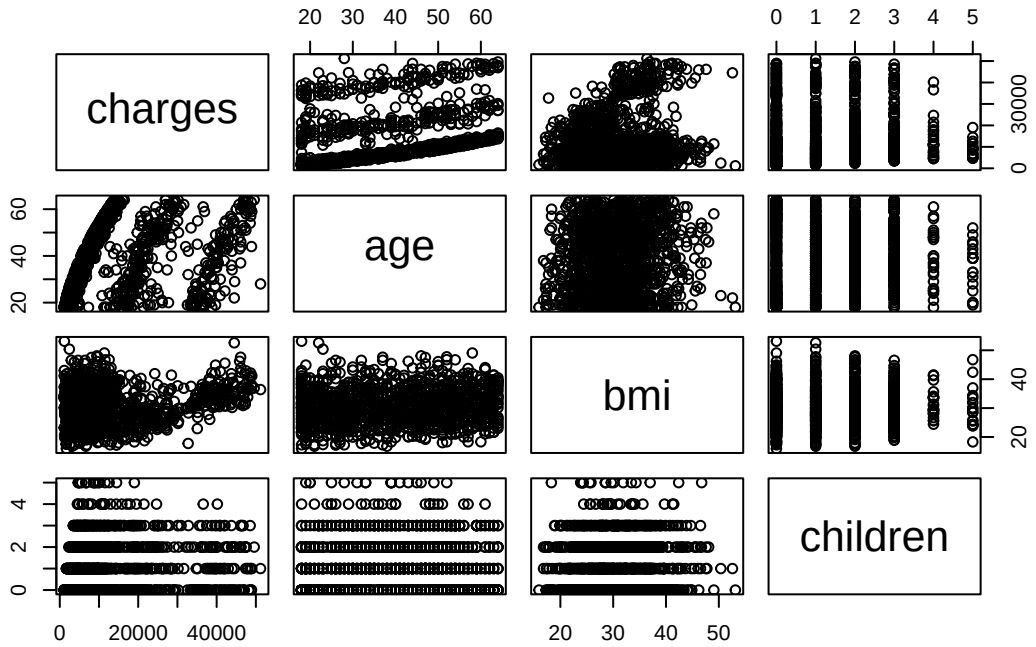
```
res.con$quali
```

```
##                 R2       p.value
## f.smok 0.618076780 4.650154e-280
## mout   0.213811211  1.784343e-71
## f.age  0.077526782  5.361680e-24
## f.bmi  0.013067483  1.610172e-04
## f.sex  0.003654488  2.742457e-02
## f.reg  0.006613994  3.193769e-02
```

```
res.con$category
```

```
##                   Estimate       p.value
## f.smok=Yes      11505.5682 4.650154e-280
## mout=MvOut.Yes  13754.4679  1.784343e-71
## f.age=Old        3842.5818  1.736319e-18
## f.bmi=High       2723.3239  3.435144e-05
## f.reg=SE         1435.8572  8.076659e-03
## f.sex=M           709.5889  2.742457e-02
## f.age=Medium      152.3176  3.240866e-01
## f.reg=NE          234.6603  7.350039e-01
## f.reg=SW         -809.1676  1.335872e-01
## f.reg=NW         -861.3500  1.128202e-01
## f.bmi=Low       -2125.3115  1.052167e-01
## f.sex=F          -709.5889  2.742457e-02
## f.bmi=Normal     -598.0124  1.743093e-04
## f.age=Young     -3994.8994  2.095461e-20
## mout=MvOut.No  -13754.4679  1.784343e-71
## f.smok=No      -11505.5682 4.650154e-280
```

```r
plot(df[,c(4,1:3)])
```




```r
cor(df[,c(4,1:3)],method="spearman") # Non Parametric version for general variables
```

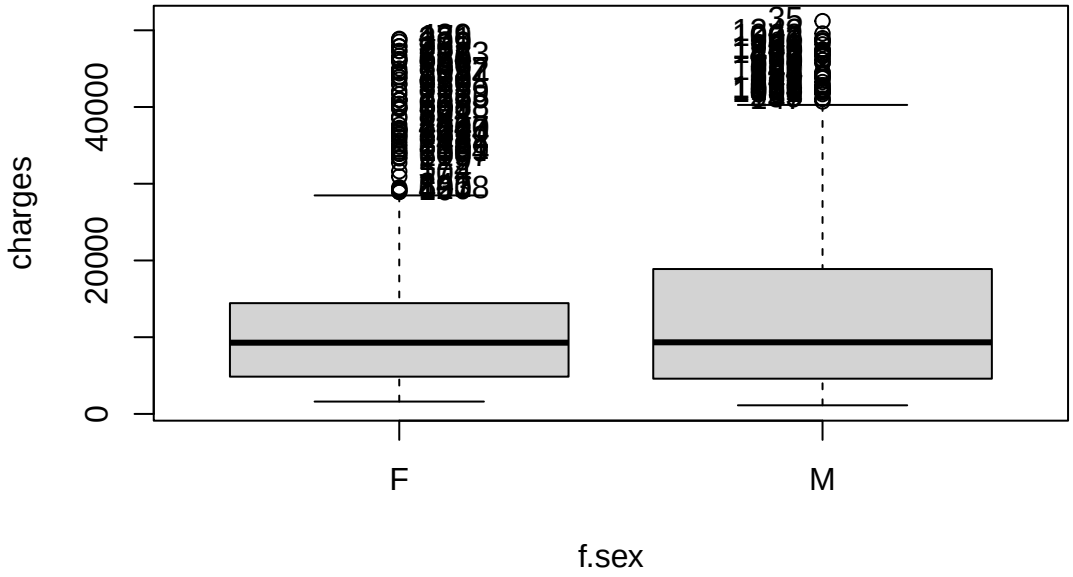
```
##             charges        age        bmi    children
## charges   1.0000000 0.53429693 0.11365380 0.13672344
## age       0.5342969 1.00000000 0.10664431 0.05742152
## bmi       0.1136538 0.10664431 1.00000000 0.01717882
## children 0.1367234 0.05742152 0.01717882 1.00000000
```

```r
#INTERACTION BETWEEN CATEGORICAL AND NUMERICAL VARIABLES
# --> It can be done with condes$quali

#SEX
# Numeric
with(df, tapply(charges,f.sex,summary))
```

```
## $F
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1608    4857    9284   12357   14438   48885
##
## $M
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1122    4619    9333   13777   18845   51195
```

```r
# Graphics
Boxplot(charges~f.sex,data=df, id=list(n=Inf,labels=row.names(df))) # It does work (Bec
```



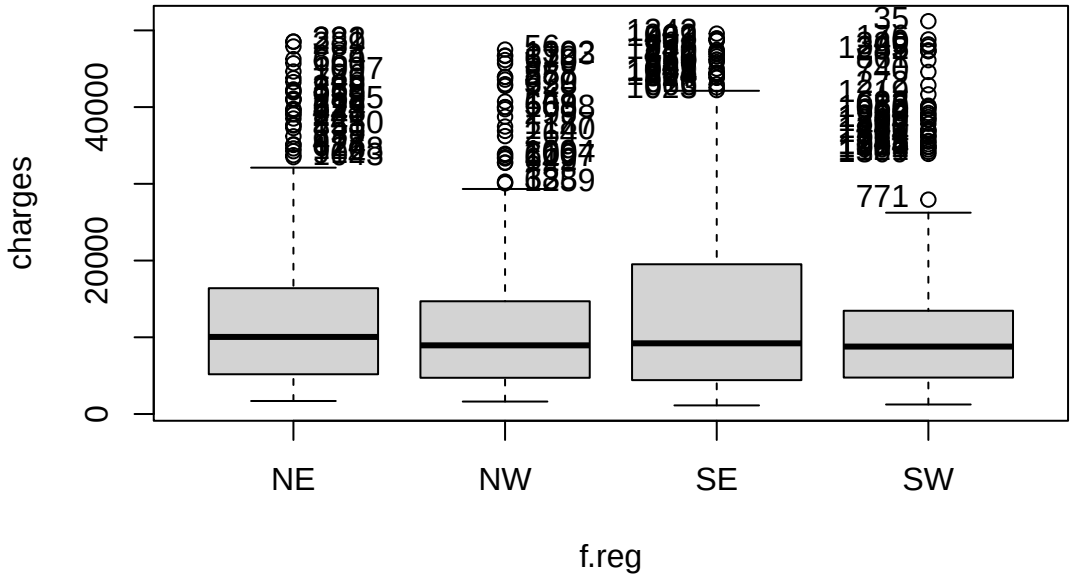

```
##   [1] "10"   "24"   "85"   "87"   "95"   "104"  "129"  "162"  "176"  "204"
##  [11] "241"  "243"  "245"  "252"  "289"  "315"  "329"  "331"  "420"  "442"
##  [21] "489"  "550"  "559"  "574"  "588"  "600"  "616"  "619"  "630"  "668"
##  [31] "675"  "697"  "707"  "726"  "737"  "804"  "843"  "846"  "851"  "853"
##  [41] "857"  "861"  "884"  "891"  "1013" "1022" "1032" "1038" "1094" "1097"
##  [51] "1121" "1123" "1125" "1140" "1153" "1197" "1207" "1219" "1309" "1314"
##  [61] "1324" "1338" "35"   "40"   "56"   "110"  "147"  "186"  "253"  "255"
##  [71] "257"  "266"  "272"  "282"  "293"  "313"  "328"  "339"  "382"  "421"
##  [81] "422"  "531"  "570"  "666"  "669"  "678"  "740"  "743"  "827"  "894"
##  [91] "902"  "952"  "957"  "959"  "1023" "1048" "1063" "1091" "1112" "1241"
## [101] "1242" "1285" "1302"
```

```
#REGION
# Numeric
with(df, tapply(charges,f.reg,summary))
```

```
## $NE
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1695    5179   10043   13267   16398   48549
##
## $NW
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1621    4724    8948   12171   14626   47496
##
## $SE
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1122    4424    9212   14468   19498   49578
##
## $SW
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##    1242    4750    8791   12223   13454   51195
```

```
# Graphics
Boxplot(charges~f.reg,data=df, id=list(n=Inf,labels=row.names(df))) # It does work (Bec
```

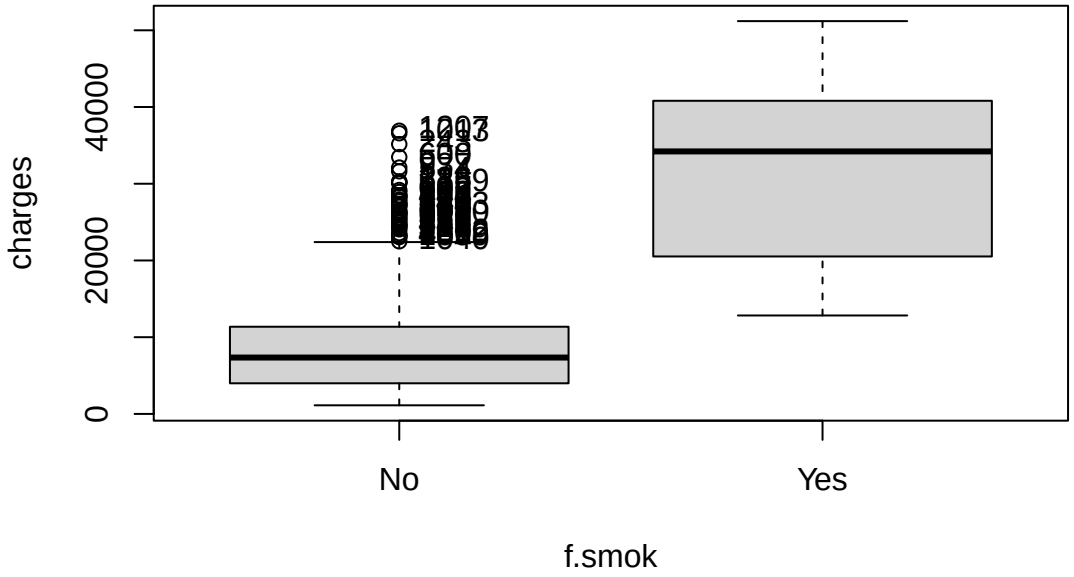



```
##   [1] "24"   "39"   "124"  "186"  "241"  "255"  "282"  "289"  "331"  "339"
##  [11] "382"  "423"  "477"  "624"  "669"  "739"  "743"  "829"  "851"  "853"
##  [21] "884"  "912"  "918"  "948"  "959"  "1043" "1097" "1125" "1250" "1268"
##  [31] "56"   "63"   "87"   "129"  "147"  "257"  "260"  "264"  "299"  "328"
##  [41] "388"  "559"  "570"  "588"  "600"  "630"  "642"  "668"  "678"  "1038"
##  [51] "1094" "1123" "1140" "1187" "1197" "1259" "1302" "110"  "253"  "266"
##  [61] "421"  "422"  "489"  "531"  "550"  "616"  "666"  "675"  "707"  "827"
##  [71] "846"  "894"  "902"  "952"  "1023" "1032" "1048" "1063" "1241" "1242"
##  [81] "1324" "20"   "30"   "31"   "35"   "40"   "85"   "95"   "176"  "224"
##  [91] "243"  "252"  "272"  "315"  "323"  "329"  "374"  "442"  "501"  "610"
## [101] "622"  "683"  "740"  "771"  "861"  "954"  "1002" "1050" "1121" "1207"
## [111] "1208" "1219" "1285" "1289" "1292" "1304" "1309" "1314"
```

```
#SMOKER
# Numeric
with(df, tapply(charges,f.smok,summary))
```

```
## $No
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1122    3989    7346    8441   11363   36911
##
## $Yes
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12829   20634   34210   31452   40766   51195
```

```
# Graphics
Boxplot(charges~f.smok,data=df, id=list(n=Inf,labels=row.names(df))) # It does work (Be
```




```
##  [1] "10"   "63"   "116"  "139"  "141"  "220"  "228"  "243"  "246"  "290"
## [11] "322"  "356"  "380"  "388"  "431"  "444"  "469"  "492"  "517"  "521"
## [21] "527"  "540"  "574"  "600"  "638"  "659"  "689"  "697"  "771"  "807"
## [31] "877"  "926"  "937"  "960"  "965"  "981"  "988"  "1009" "1013" "1020"
## [41] "1040" "1099" "1143" "1207" "1212" "1259"
```