# Lab Report 2

*Theo Lambert and William Didier*

*2 octobre 2017*

## Data Cleaning

### Question 1

```
f <- read.table("~/Stats/TP2/group1_forestfires.csv", sep=",", dec = ".", header = TRUE)
for (i in 2:20)
  name <- paste("~/Stats/TP2/group",i,"_forestfires.csv", sep = "")
  f <- rbind(f, read.table(name, sep=",", dec = ".", header = TRUE))
```

### Question 2

```
##   X Y month day temp RH wind rain area
## 1 7 5   mar fri  8.2 51  6.7  0.0    0
## 2 7 4   oct sat 14.6 33  1.3  0.0    0
## 3 8 6   mar fri  8.3 97  4.0  0.2    0
## 4 8 6   mar sun 11.4 99  1.8  0.0    0
```

```
## [1] "..."
```

```
##     X Y month day temp RH wind rain  area
## 596 6 5   aug fri 18.2 62  5.4    0  0.43
## 597 2 4   aug sun 21.9 71  5.8    0 54.29
## 598 7 4   aug sun 21.2 70  6.7    0 11.16
## 599 1 4   aug sat 25.6 42  4.0    0  0.00
## 600 6 3   nov tue 11.8 31  4.5    0  0.00
```

We start by saving this dataset to a file so that we can compare it to the cleaned version we will produce during this Lab Session.

```
save(f,file="~/Stats/TP2/Lab2_uncleaned_dataset.Rda")
```

We can see that some months aren't filled properly. We could imagine an iterative method that would check if any argument is not as expected, or even repair the mistakes (for example replace April by apr) We could'nt find en elegant way of replacing the set of bad values, so we decided to apply, for each variable, a simple method without looping. We start by a quick check on how many different values are in the day and month columns :

```
summary(f[,"day"])
```

```
##       fri       mon    Monday       sat  Saturday       sun       thu
##       104        88         2        89         2       115        75
##  Thursday       tue       wed Wednesday
##         1        67        56         1
```

```
summary(f[,"month"])
```

```
##      apr      aug      dec      feb      jul     July      jun      mar
##       10      213       11       18       32        1       24       66
##      may      May November      oct      sep    April December      jan
##        1        1        1       19      197        1        1        2
##      nov  October
##        1        1
```

```
f[,"day"][f[,"day"]=="Monday"] <- "mon"
f[,"day"][f[,"day"]=="Tuesday"] <- "tue"
f[,"day"][f[,"day"]=="Wednesday"] <- "wed"
f[,"day"][f[,"day"]=="Thursday"] <- "thu"
f[,"day"][f[,"day"]=="Friday"] <- "fri"
f[,"day"][f[,"day"]=="Saturday"] <- "sat"
f[,"day"][f[,"day"]=="Sunday"] <- "sun"
```

We use the same method for the months of the year

```
f[,"month"][f[,"month"]=="January"] <- "jan"
f[,"month"][f[,"month"]=="February"] <- "feb"
f[,"month"][f[,"month"]=="March"] <- "mar"
f[,"month"][f[,"month"]=="April"] <- "apr"
f[,"month"][f[,"month"]=="May"] <- "may"
f[,"month"][f[,"month"]=="June"] <- "jun"
f[,"month"][f[,"month"]=="July"] <- "jul"
f[,"month"][f[,"month"]=="August"] <- "aug"
f[,"month"][f[,"month"]=="September"] <- "sep"
f[,"month"][f[,"month"]=="October"] <- "oct"
f[,"month"][f[,"month"]=="November"] <- "nov"
f[,"month"][f[,"month"]=="December"] <- "dec"
```

We now check that there are only 7 different values in the day column, and 12 in the month column :

```
summary(f[,"day"])
```

```
##       fri       mon    Monday       sat  Saturday       sun       thu
##       104        90         0        91         0       115        76
##  Thursday       tue       wed Wednesday
##         0        67        57         0
```

```
summary(f[,"month"])
```

```
##      apr      aug      dec      feb      jul     July      jun      mar
##       11      213       12       18       33        0       24       66
##      may      May November      oct      sep    April December      jan
##        2        0        0       20      197        0        0        2
##      nov  October
##        2        0
```

### Question 3

```
summary(f)
```

```
##        X               Y              month          day           temp
##  Min.   :1.0   Min.   :2.000   aug    :213   sun    :115   Min.   : 0.20
##  1st Qu.:3.0   1st Qu.:4.000   sep    :197   fri    :104   1st Qu.:15.72
##  Median :4.0   Median :4.000   mar    : 66   sat    : 91   Median :19.50
##  Mean   :4.6   Mean   :4.277   jul    : 33   mon    : 90   Mean   :19.41
##  3rd Qu.:6.0   3rd Qu.:5.000   jun    : 24   thu    : 76   3rd Qu.:22.80
##  Max.   :9.0   Max.   :9.000   oct    : 20   tue    : 67   Max.   :81.50
##                                (Other): 47   (Other): 57   NA's   :2
##        RH              wind             rain              area
##  Min.   :  2.00   Min.   :  0.400   Min.   :0.00000   Min.   :   0.000
##  1st Qu.: 33.00   1st Qu.:  2.700   1st Qu.:0.00000   1st Qu.:   0.000
##  Median : 42.00   Median :  4.000   Median :0.00000   Median :   0.450
##  Mean   : 44.19   Mean   :  4.454   Mean   :0.01833   Mean   :  13.046
##  3rd Qu.: 53.00   3rd Qu.:  5.400   3rd Qu.:0.00000   3rd Qu.:   6.365
##  Max.   :100.00   Max.   :130.800   Max.   :6.40000   Max.   :1090.840
##                   NA's   :2
```

We can see that some data has some NA values : we need to remove those lines because the data is not exploitable. We decide to use the na.omit function to suppress the unexploitable data.

```
f <- na.omit(f)
```

lets check that everything is now in order :

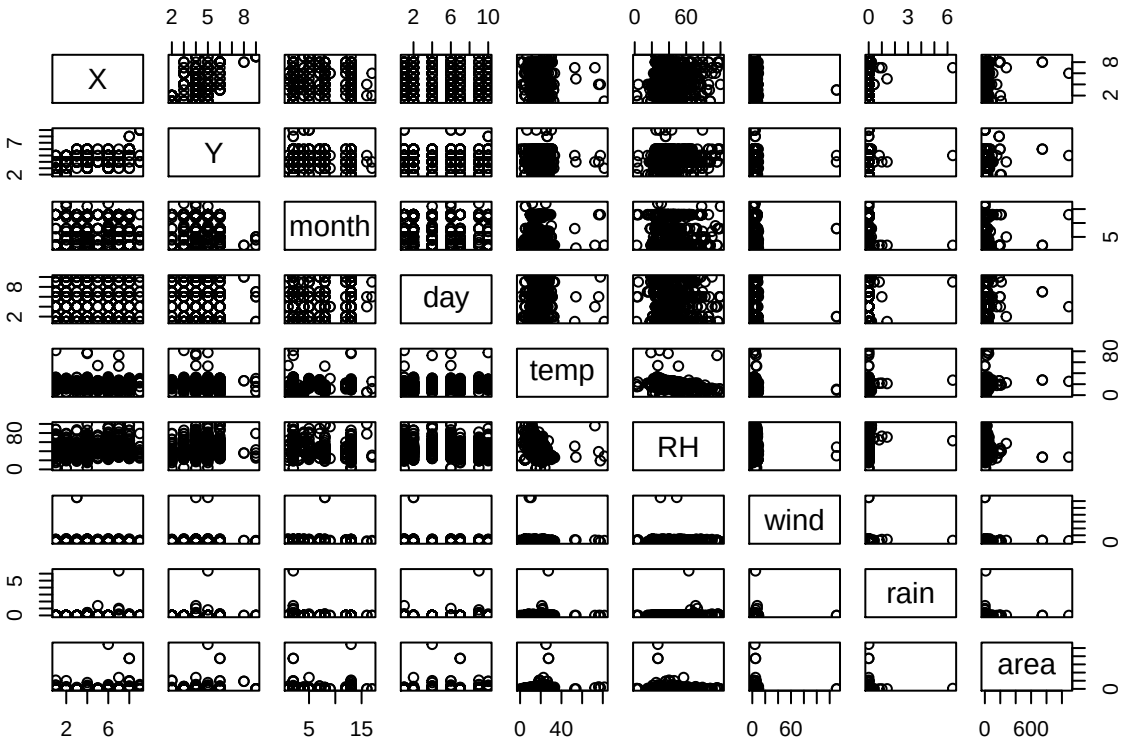
```
summary(f)
```

```
##        X               Y              month          day
##  Min.   :1.000   Min.   :2.000   aug    :213   sun    :115
##  1st Qu.:3.000   1st Qu.:4.000   sep    :196   fri    :103
##  Median :4.000   Median :4.000   mar    : 64   sat    : 91
##  Mean   :4.611   Mean   :4.273   jul    : 33   mon    : 90
##  3rd Qu.:6.000   3rd Qu.:5.000   jun    : 24   thu    : 76
##  Max.   :9.000   Max.   :9.000   oct    : 19   tue    : 65
##                                  (Other): 47   (Other): 56
##       temp             RH              wind             rain
##  Min.   : 0.20   Min.   :  2.00   Min.   :  0.400   Min.   :0.00000
##  1st Qu.:15.70   1st Qu.: 33.00   1st Qu.:  2.700   1st Qu.:0.00000
##  Median :19.50   Median : 42.00   Median :  4.000   Median :0.00000
##  Mean   :19.42   Mean   : 44.25   Mean   :  4.445   Mean   :0.01846
##  3rd Qu.:22.80   3rd Qu.: 53.00   3rd Qu.:  5.400   3rd Qu.:0.00000
##  Max.   :81.50   Max.   :100.00   Max.   :130.800   Max.   :6.40000
##
##       area
##  Min.   :   0.000
##  1st Qu.:   0.000
##  Median :   0.520
##  Mean   :  13.134
##  3rd Qu.:   6.393
##  Max.   :1090.840
##
```

### Question 4

We use the pairs function to build scatterplot matrixes.

```
pairs(f[,1:9])
```




The scatterplot clearly outlines some extreme values such as heavy rain while a significant area burned (observation n°592). Although this might be surprising, we cannot discredit this data as it is likely to simply be an extreme observation rather than a mistake. We also noticed two extreme observations regarding the area burned in a single fire. When investigating, we realised that two observations (n°240 and n°544) were exactly the same. this time, we can suppose that there has been a mistake in the entry of the data. We will investigate a method to suppress doubled data.

### Question 5

After a deeper analysis of the previous scatterplot matrixes, we found some extreme temperature values. After investigation, we saw that 6 values (n° 586,430,125,117,276,301) were clearly above normal temperatures in Celsius. We decided to delete these observations from our dataset.

```
f <- f[!f["temp"]>50,]
```

Also, we found some humidity values that aren't right. It is said that humidity is supposed to be a numerical value between 15 and 100.

```
f <- f[!f["RH"]<15,]
f <- f[!f["RH"]>100,]
```

We also found some extreme wind values (n°377 and 64). Even though the values are more than 10 times bigger than any other one, it's hard to conclude it's an error without having more info about the terrain. Indeed, having 130 km/h of wind is possible in many parts of the globe. We used the unique() function to

suppress all the doubled data in our set. We decides to keep the extreme values that we found with the scatterplot matrix, as we could'nt be sure wether they were extreme observations or mistakes.

```
f <- unique(f)
save(f,file="~/Stats/TP2/Lab2_cleaned_dataset.Rda")
```

Our cleaned dataset is in the Rda file attached to the same email as the on this file was attached to.

### Question 6

```
summary(f[,"X"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   4.000   4.681   7.000   9.000
```

```
summary(f[,"Y"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   4.000   4.000   4.305   5.000   9.000
```

```
table(f[,"X"],f[,"Y"])
```

```
##
##      2  3  4  5  6  8  9
##   1 15  6 13  4  0  0  0
##   2 22  1 23 13  0  0  0
##   3  0  1 36  8  3  0  0
##   4  0 18 28 23  8  0  0
##   5  0  0 21  2  2  0  0
##   6  0 21  8 41  3  0  0
##   7  0  2 34  9  1  0  0
##   8  0  3  1  2 45  1  0
##   9  0  0  4  2  0  0  5
```

We can see that individually, X and Y seem to be quite evenly distributed in the analysed area. But once we analyse the couple (X,Y) we can see that its not the case anymore : the fires seem to declare only in the diagonal and not evenly in the area. We could already guess that by looking at the scatterplot matrix we built earlier. We will now analyse the couples of variables through the correlation between them.

```
cor(f[,"X"],f[,"Y"])
```

```
## [1] 0.5343046
```

```
cor(f[,"X"],f[,"temp"])
```

```
## [1] -0.07277773
```

```
cor(f[,"X"],f[,"RH"])
```

```
## [1] 0.09501464
```

```
cor(f[,"X"],f[,"wind"])
```

```
## [1] -0.03637818
```

```
cor(f[,"X"],f[,"rain"])
```

```
## [1] 0.06397025
```

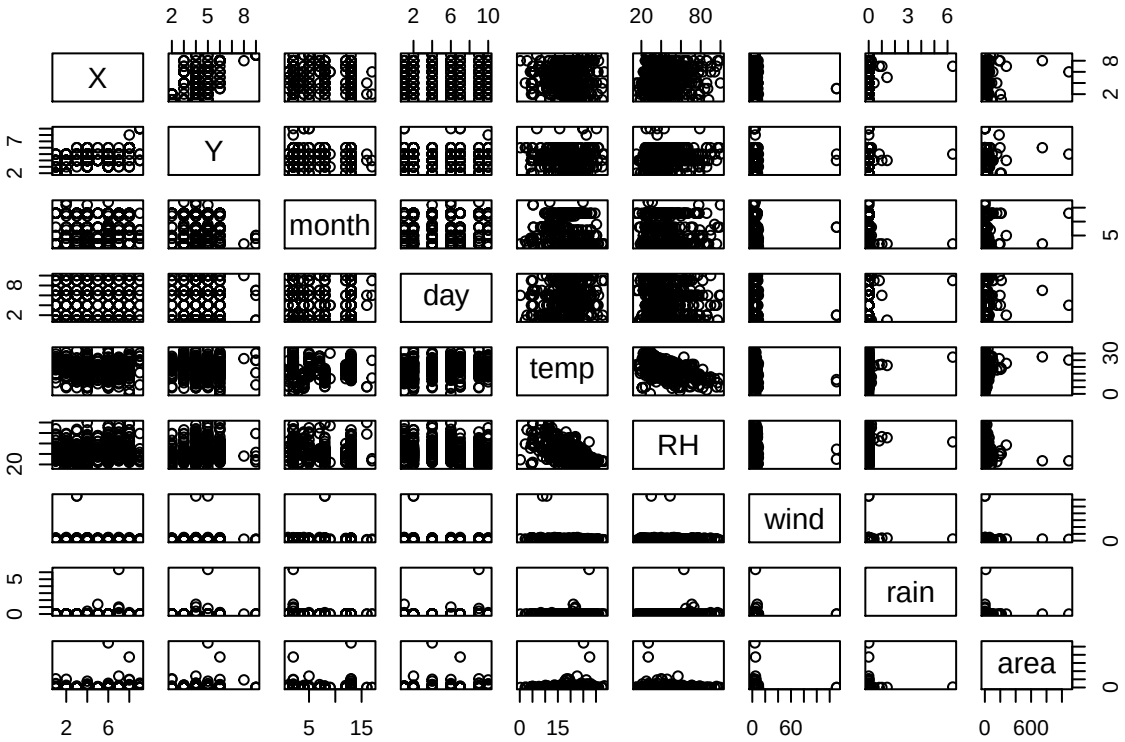
```
cor(f[,"X"],f[,"area"])
```

```
## [1] 0.07910818
```

Besides the correlation between X and Y that we had identified earlier, there's no variable that seems to be correlated to X.

Rather than testing one by one each couple, we will have another look at the scatterplot matrix now that we've cleaned the dataset. Once we identify a couple of variables that look like they have a relation, we quantify it with the correlation function.

```
pairs(f[,1:9])
```




We can identify a correlation between Rh and temp, which seems quite logical. Lets check this with the correlation function

```r
cor(f[,"temp"],f[,"RH"])
```

```
## [1] -0.5028369
```

Indeed, the correlation coefficient is quite high in absolute value.

On the other hand, it seems that there is no correlation whatsoever between RH and wind. Lets check it too.

```r
cor(f[,"wind"],f[,"RH"])
```

```
## [1] -0.008625523
```

As we thought, there is no correlation between these 2 variables.

The data we analyse is about forest fires. We think that the most important data to check its correlation with is the area of forest that burned. We will end this Lab Session by checking the value of the correlation between area and all the other variables.

```r
cor(f[,"area"],f[,"X"])
```

```
## [1] 0.07910818
```

```r
cor(f[,"area"],f[,"Y"])
```

```
## [1] 0.05782916
```

```r
cor(f[,"area"],f[,"temp"])
```

```
## [1] 0.1043539
```

```r
cor(f[,"area"],f[,"rain"])
```

```
## [1] -0.007075314
```

```r
cor(f[,"area"],f[,"RH"])
```

```
## [1] -0.08150537
```

```r
cor(f[,"area"],f[,"wind"])
```

```
## [1] -0.009575703
```

None of these values are significant : we can't conclude to any correlation between the area burned and other variables.