INTRODUCTION

author: "BADOHOUN IDIR" title: 'Projet : Analyse de Donnees2' date: "25 avril 2018"

INTRODUCTION

Nous avons vu en cours des mÃ©thodes pour faire l'analyse de donnÃ©es allant de la collecte des donnees , nettoyage des donnÃ©es Ã l'analyse exploratoire des donnees .

Dans ce projet Notre analyse portera sur un ensemble de donnÃ©es de qualitÃ© de l'air . Il contient les reponses d'un dispositif multicapteur a gaz deploye dans une ville italienne.

L'ensemble de donnees contient 9358 occurrences de reponses horaires moyennes d'un ensemble de 5 capteurs chimiques d'oxyde metallique incorpores dans un dispositif multicapteur chimique de qualite de l'air. L'appareil etait situe sur le terrain dans une zone fortement polluee, au niveau de la route, dans une ville italienne. Les donnees ont ete enregistrees de mars 2004 Ã fevrier 2005 (un an).

cela reprÃ©sente les enregistrements les plus longs disponibles gratuitement des rÃ©ponses des dispositifs de detection chimique de qualitÃ© de l'air dÃ©ployÃ©s sur le terrain.

les deux premieres variable donnent la date et l'heure ,les 10 suivantes representent des concentrations de produit chimique dans l 'air,T la temperature,RH et AH representent respectivement l'humidite relative et absolue.

##1 Extraction des donnÃ©es:

la recupÃ©ration d'un jeu de donnees depuis le site et chargement de donnees Le jeu de donnees se trouve sur le site "https://archive.ics.uci.edu/ml/datasets/Air+quality". Ce site contient des donnÃ©es etudies dans les centres d'apprentissage automatiques. Ce site repertorie tous les jeux de donnees d'interet en fonction des problematiques en machine learning : regression, classification notamment . A noter que nous allons utiliser le package tidyverse et ses differentes fonctions dans le cadre de notre analyse .

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)

td <- tempdir()
 tf <- tempfile(tmpdir=td, fileext=".zip")
 download.file("https://archive.ics.uci.edu/ml/machine-learning-databases/00360/AirQualityU
 fname <- unzip(tf, list=TRUE)$Name[1]
 unzip(tf, files=fname, exdir=td, overwrite=TRUE)
 fpath <- file.path(td, fname)
 data <-read.csv2(fpath)

head(data,10)
```

```
##          Date     Time CO.GT. PT08.S1.CO. NMHC.GT. C6H6.GT. PT08.S2.NMHC.
## 1  10/03/2004 18.00.00    2.6        1360      150     11.9          1046
## 2  10/03/2004 19.00.00    2.0        1292      112      9.4           955
## 3  10/03/2004 20.00.00    2.2        1402       88      9.0           939
## 4  10/03/2004 21.00.00    2.2        1376       80      9.2           948
## 5  10/03/2004 22.00.00    1.6        1272       51      6.5           836
## 6  10/03/2004 23.00.00    1.2        1197       38      4.7           750
## 7  11/03/2004 00.00.00    1.2        1185       31      3.6           690
## 8  11/03/2004 01.00.00    1.0        1136       31      3.3           672
## 9  11/03/2004 02.00.00    0.9        1094       24      2.3           609
## 10 11/03/2004 03.00.00    0.6        1010       19      1.7           561
##    NOx.GT. PT08.S3.NOx. NO2.GT. PT08.S4.NO2. PT08.S5.O3.    T   RH     AH
## 1      166         1056     113         1692        1268 13.6 48.9 0.7578
## 2      103         1174      92         1559         972 13.3 47.7 0.7255
## 3      131         1140     114         1555        1074 11.9 54.0 0.7502
## 4      172         1092     122         1584        1203 11.0 60.0 0.7867
## 5      131         1205     116         1490        1110 11.2 59.6 0.7888
## 6       89         1337      96         1393         949 11.2 59.2 0.7848
## 7       62         1462      77         1333         733 11.3 56.8 0.7603
## 8       62         1453      76         1333         730 10.7 60.0 0.7702
## 9       45         1579      60         1276         620 10.7 59.7 0.7648
## 10    -200         1705    -200         1235         501 10.3 60.2 0.7517
##     X X.1
## 1  NA  NA
## 2  NA  NA
## 3  NA  NA
## 4  NA  NA
## 5  NA  NA
## 6  NA  NA
## 7  NA  NA
## 8  NA  NA
## 9  NA  NA
## 10 NA  NA
```

##2 Mise en forme sous format tibble et prÃ©paration du jeu de donnÃ©es:

les donnees d'entrees doivent etre traites avant de les utiliser dans notre analyse exploratoire . Cela signifie supprimer des lignes sans valeurs , verifier la corrÃ©lation et les valeurs aberrantes. Lors de la construction du modÃ¨le, R prend en charge les valeurs nulles et supprime les lignes ou les donnees manquantes . Cela entraine une eventuelle perte des donnees et donc d'information. Nous utilisons ici dans notre analyse des tibbles et non des dataframes habituels pour notre jeu de donnees car les tibbles presente une methode d'impression raffinee qui montre seulement les 10 premieres lignes, et toutes les colonnes qui correspondent Ã l'ecran. Cela rend beaucoup plus facile le travail avec des donnees de grandes dimensions . En plus de son nom, chaque colonne rapporte son type, une belle caracteristique empruntee Ã la fonction str .

```
 data=as.tibble(data)
str(data)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    9471 obs. of  17 variables:
##  $ Date         : Factor w/ 392 levels "","01/01/2005",..: 116 116 116 116 116 116 129 12
##  $ Time         : Factor w/ 25 levels "","00.00.00",..: 20 21 22 23 24 25 2 3 4 5 ...
##  $ CO.GT.       : num  2.6 2 2.2 2.2 1.6 1.2 1.2 1 0.9 0.6 ...
##  $ PT08.S1.CO.  : int  1360 1292 1402 1376 1272 1197 1185 1136 1094 1010 ...
##  $ NMHC.GT.     : int  150 112 88 80 51 38 31 31 24 19 ...
##  $ C6H6.GT.     : num  11.9 9.4 9 9.2 6.5 4.7 3.6 3.3 2.3 1.7 ...
##  $ PT08.S2.NMHC.: int  1046 955 939 948 836 750 690 672 609 561 ...
##  $ NOx.GT.      : int  166 103 131 172 131 89 62 62 45 -200 ...
##  $ PT08.S3.NOx. : int  1056 1174 1140 1092 1205 1337 1462 1453 1579 1705 ...
##  $ NO2.GT.      : int  113 92 114 122 116 96 77 76 60 -200 ...
##  $ PT08.S4.NO2. : int  1692 1559 1555 1584 1490 1393 1333 1333 1276 1235 ...
##  $ PT08.S5.O3.  : int  1268 972 1074 1203 1110 949 733 730 620 501 ...
##  $ T            : num  13.6 13.3 11.9 11 11.2 11.2 11.3 10.7 10.7 10.3 ...
##  $ RH           : num  48.9 47.7 54 60 59.6 59.2 56.8 60 59.7 60.2 ...
##  $ AH           : num  0.758 0.726 0.75 0.787 0.789 ...
##  $ X            : logi  NA NA NA NA NA NA ...
##  $ X.1          : logi  NA NA NA NA NA NA ...
```

Nettoyage des donnees: En regardant de plus pres notre tibble on se rend compte de l'existence de plusieurs lignes(de 9358 jusqu'a 9471) et colonnes (les 2 dernieres) completements vides. Dans la description du jeu de donnees , la valeur -200 a etÃ© atribuee a toutes les mesures aberrantes,on va remplacer ces dernieres par la moyenne de la variable.

```
data=data[-c(9358:9471),-c(16,17)]


for(i in 3:dim(data)[2])

 data[which(data[,i]==-200),i]=sum(data[ - which(data[,i]==(-200)),i])/dim(data[ - which(dat
```

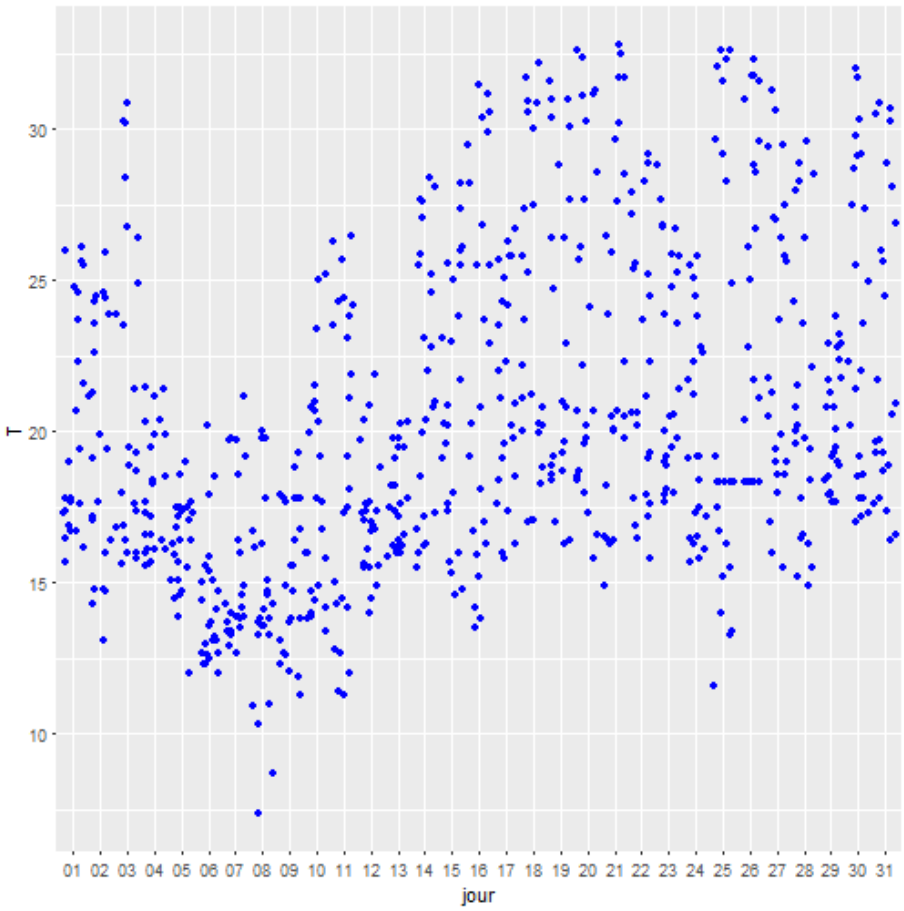
##3 Differentes formes de visualisations rapides de nos donnees en utilisant la

librairie des graphiques ggplot et quelques unes de ses pallettes intÃ©ressantes dans le cadre de notre analyse

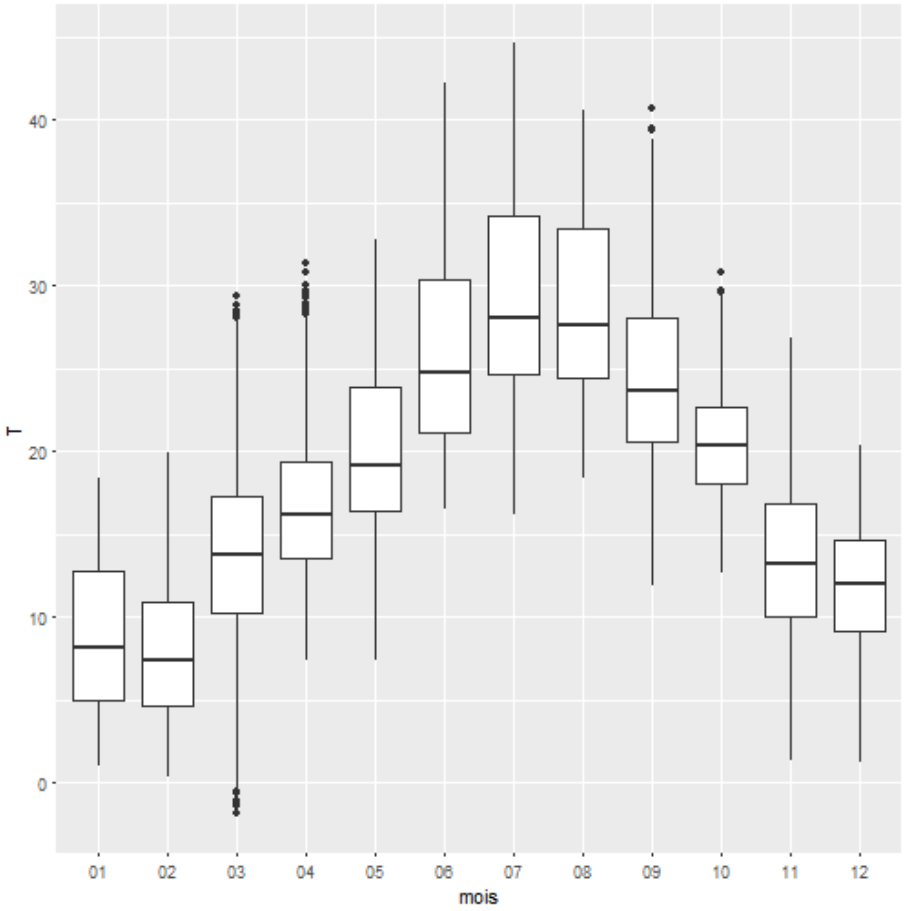
```
Date1=as.Date.character(format = "%d/%m/%Y" ,x = data$Date )
data=mutate(data,mois=format(Date1, format="%m"))
data=mutate(data,jour=format(Date1, format="%d"))

month5=filter(data,mois=="05")

month5
```

```
## # A tibble: 744 x 17
##    Date   Time  CO.GT. PT08.S1.CO. NMHC.GT. C6H6.GT. PT08.S2.NMHC. NOx.GT.
##    <fct>  <fct>  <dbl>       <dbl>    <dbl>    <dbl>         <dbl>   <dbl>
##  1 01/05~ 00.0~  3.50        1425.     275.    15.2          1155.   185.
##  2 01/05~ 01.0~  2.40        1179.     219.     9.30          951.   129.
##  3 01/05~ 02.0~  1.60        1047.     219.     6.50          833.    85.0
##  4 01/05~ 03.0~  1.30        1009.     219.     6.30          825.   247.
##  5 01/05~ 04.0~  2.15         882.     219.     3.50          685.    35.0
##  6 01/05~ 05.0~  0.600        858.     219.     2.60          624.    24.0
##  7 01/05~ 06.0~  0.700        875.     219.     3.10          659.    30.0
##  8 01/05~ 07.0~  1.20        1012.     219.     6.50          833.    75.0
##  9 01/05~ 08.0~  1.50        1049.     219.     6.50          836.    79.0
## 10 01/05~ 09.0~  1.80        1131.     219.     8.80          932.   104.
## # ... with 734 more rows, and 9 more variables: PT08.S3.NOx. <dbl>,
## #   NO2.GT. <dbl>, PT08.S4.NO2. <dbl>, PT08.S5.O3. <dbl>, T <dbl>,
## #   RH <dbl>, AH <dbl>, mois <chr>, jour <chr>
```

```
ggplot(data = month5) +
  geom_point(mapping = aes(x = jour, y = T), position = "jitter",color="blue")
```



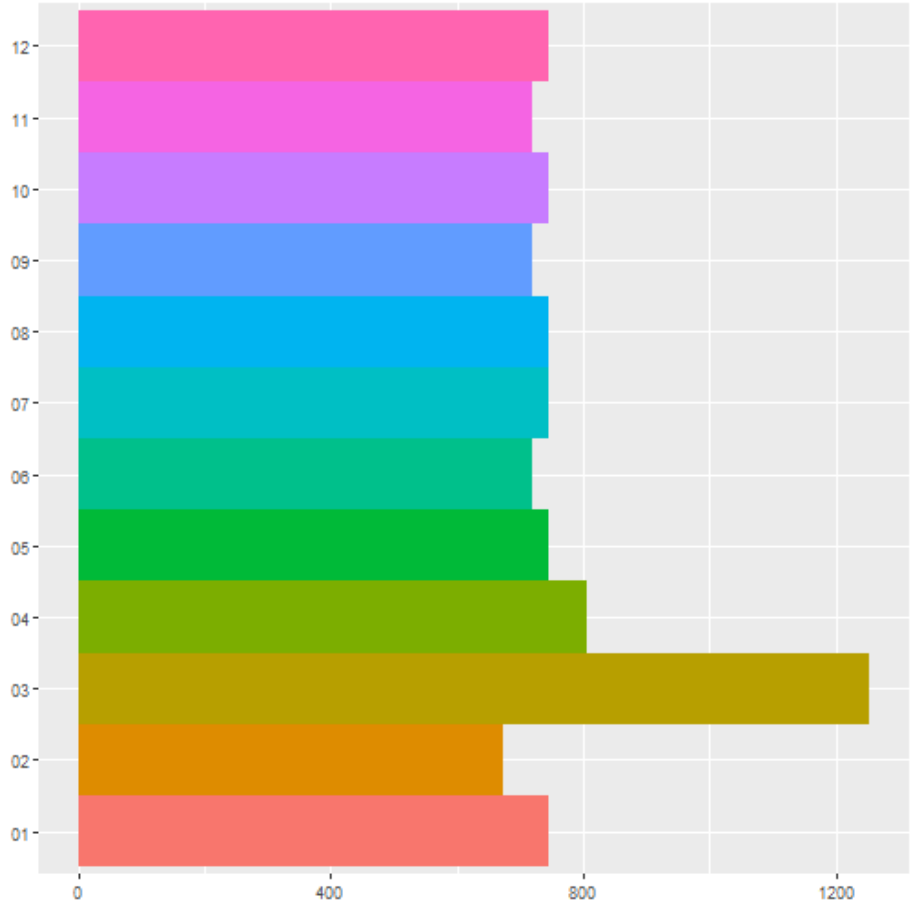

```
ggplot(data = data , mapping = aes(x =mois, y = T)) +
  geom_boxplot()
```



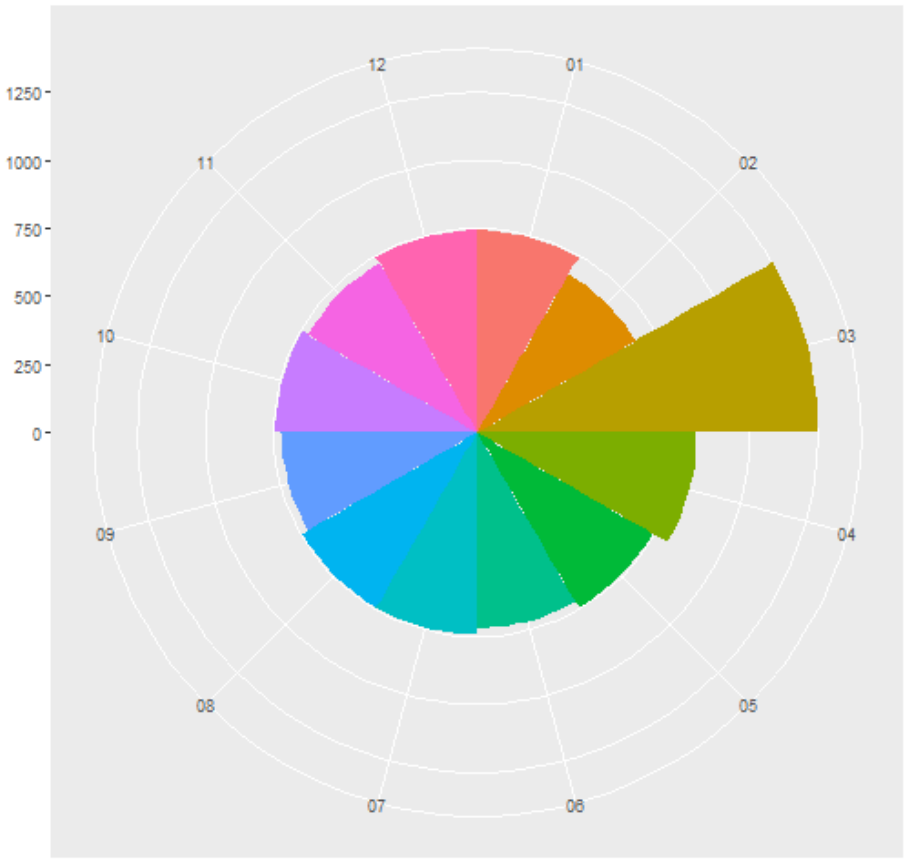
on a pris comme sous echantillons les donnees du mois de mai,et visualiser la temperature selon jour,on apprend que du 4 au 10 mai ete les jours les plus frais de mai.

On constate l'augmentation de la temperature en ete comme attendue.

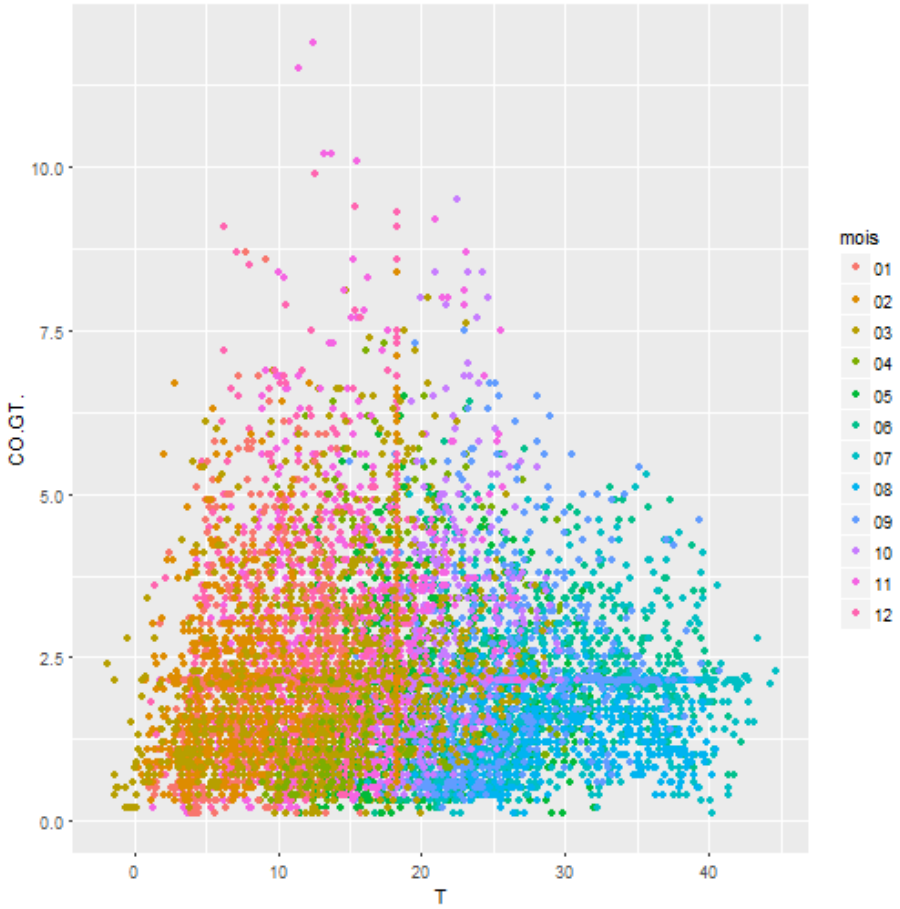
```
barmois <- ggplot(data = data) +
  geom_bar(
    mapping = aes(x = mois, fill = mois),
    show.legend = FALSE,
    width = 1
  ) +
  theme(aspect.ratio = 1) +
  labs(x = NULL, y = NULL)
barmois + coord_flip()
```




```
barmois + coord_polar()
```




```
ggplot(data = data) +
  geom_point(mapping = aes(x = T, y = CO.GT.,color=mois))
```



. On remarque que la concentration du CO explose en hiver.

4 Analyse exploratoire et Visualisation des donnees

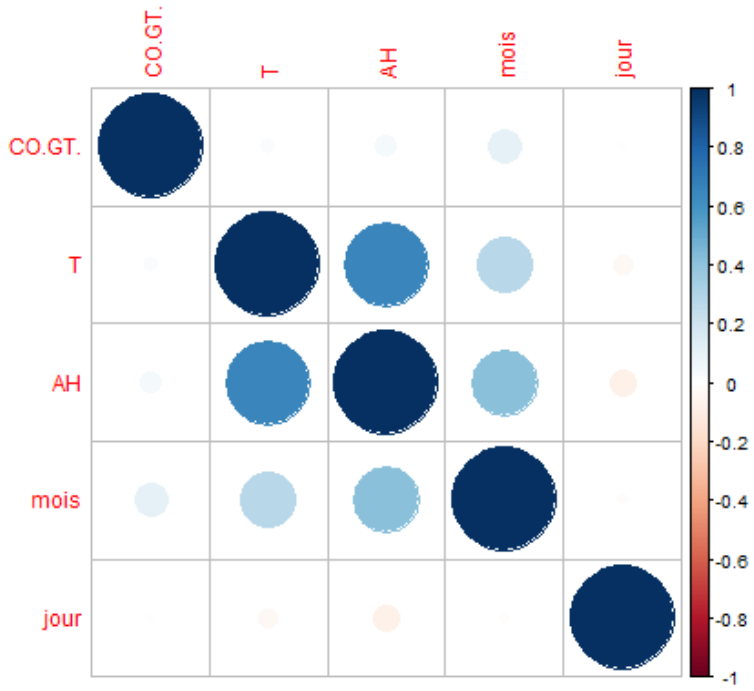
Dans cette section nous allons utiliser la visualisation et la transformation pour explorer les donnees : c'est une etape indispensable dans toute analyse statistique . Cela nous permettra par ailleurs de rÃ©soudre des questions sur nos donnees . En effet on se pose des questions si les donnees rÃ©pondent Ã nos attentes ou non . Nous allons donc deployer tous les outils de l'analyse exploratoire des donnees : la visualisation , la transformation et la modelisation . Chaque variable dans notre jeu de donnees a son propre modÃ¨le de variation, elle peut nous reveler des informations interessantes. La meilleure faÃ§on de comprendre ce modele est de visualiser la distribution des valeurs de la variable.

4.1 La corrÃ©lation entre les variables

On commence par regarder les correlations entre les variables On utilise une bibliothÃ¨que en plus ici qui nous donne une meilleure rÃ©prÃ©sentation des correlations entre les variables indÃ©pendantes c'est la bibliotheque corrplot

Avant de passer a la visualisation entre les variables certaines transformations sont nÃ©cessaires on cree deux nouvelles colonnes contenant les jours et les mois numerotes Ã l'aide de la la fonction mutate .

```
library(corrplot)

data$jour=as.numeric(data$jour)
data$mois=as.numeric(data$mois)

corrplot(cor(data[,c(3,13,15,16,17)]),method='circle')
```



le graphique ci dessus nous montre les differentes correlations entre nos variables principales,et on en deduit l'inexistance de correlation entre elles (a part entre la temperature et l'humidite)

##4.2 Analyse exploratoire des donnees

Dans cette section nous allons utiliser la visualisation et la transformation pour

explorer les donnees : c'est une etape indispensable dans toute analyse statistique . Cela nous permettra par ailleurs de resoudre des questions sur nos donnees . En effet on se pose des questions si les donnees repondent Ã nos attentes ou non . Nous allons donc deployer tous les outils de l'analyse exploratoire des donnees : la visualisation , la transformation et la modelisation . Chaque variable dans notre jeu de donnÃ©es a son propre modele de variation, elle peut nous reveler des informations interessantes.

La faÃ§on dont on visualise la distribution d'une variable depend de la nature de la variable, qu'elle soit categorielle ou continue. Pour examiner la distribution d'une variable categorielle, on utilise un graphique a barres: c'est ce que nous faisons avec la variable tempÃ©rature ici dans notre analyse . La hauteur des barres indique combien d'observations se sont produites avec chaque valeur x. Ici on le calcule aussi manuellement .

Par ailleurs pour examiner la distribution d'une variable continue, on utilise un histogramme. On fait de meme le calcul manuellement en combinant cut et count .

On remarque par ailleurs que les temperatures tournent autour de 20.

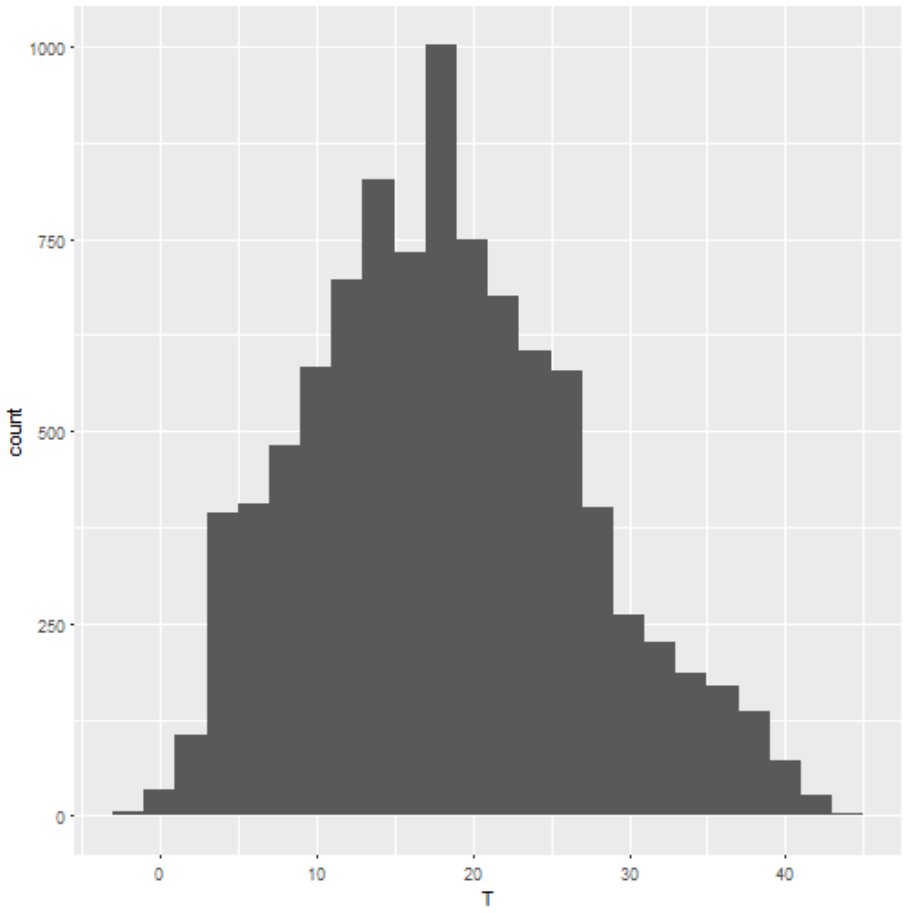
Ici on a zoomer sur les faibles temperatures en choisissant les tempÃ©ratures inferieures Ã 3 Nous avons aussi superposer plusieurs histogrammes dans le meme trace.

Pour rendre la tendance plus facile Ã voir nous pouvons reorganiser le mois en fonction de la valeur mediane de la variables de concentration co.gt .

```
data %>%
  count(mois)
```

```
## # A tibble: 12 x 2
##     mois     n
##    <dbl> <int>
##  1    1.   744
##  2    2.   672
##  3    3.  1254
##  4    4.   807
##  5    5.   744
##  6    6.   720
##  7    7.   744
##  8    8.   744
##  9    9.   720
## 10   10.   744
## 11   11.   720
## 12   12.   744
```

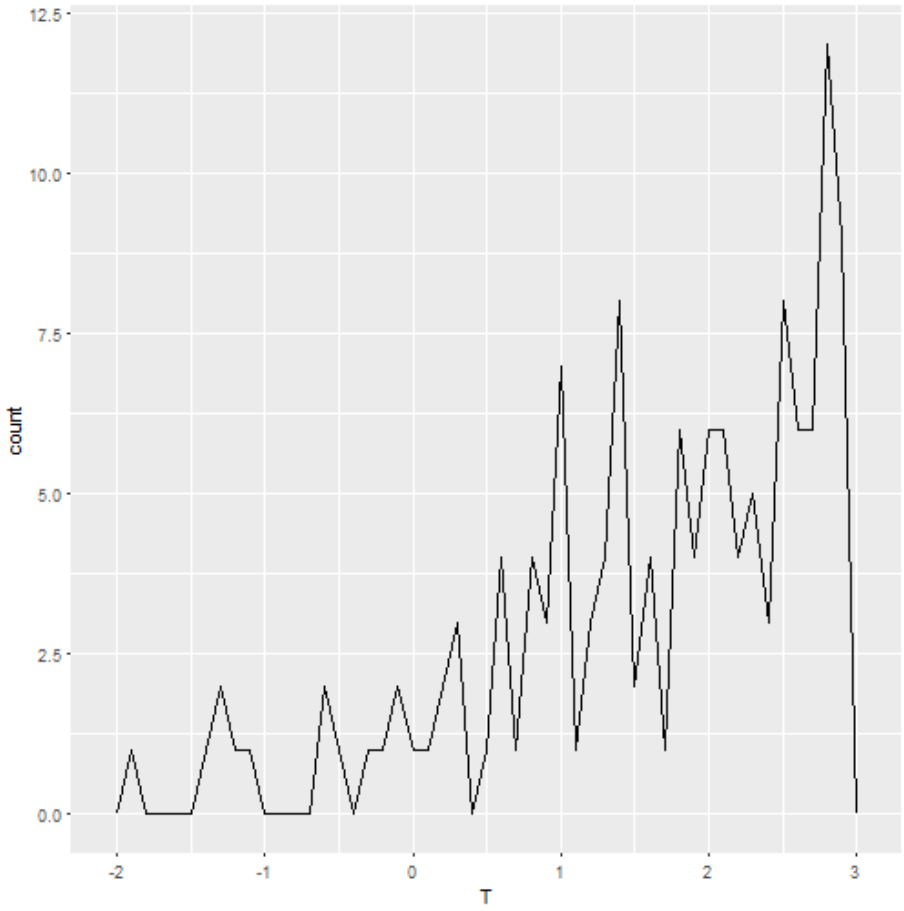
```
ggplot(data = data) +
  geom_histogram(mapping = aes(x = T), binwidth = 2)
```



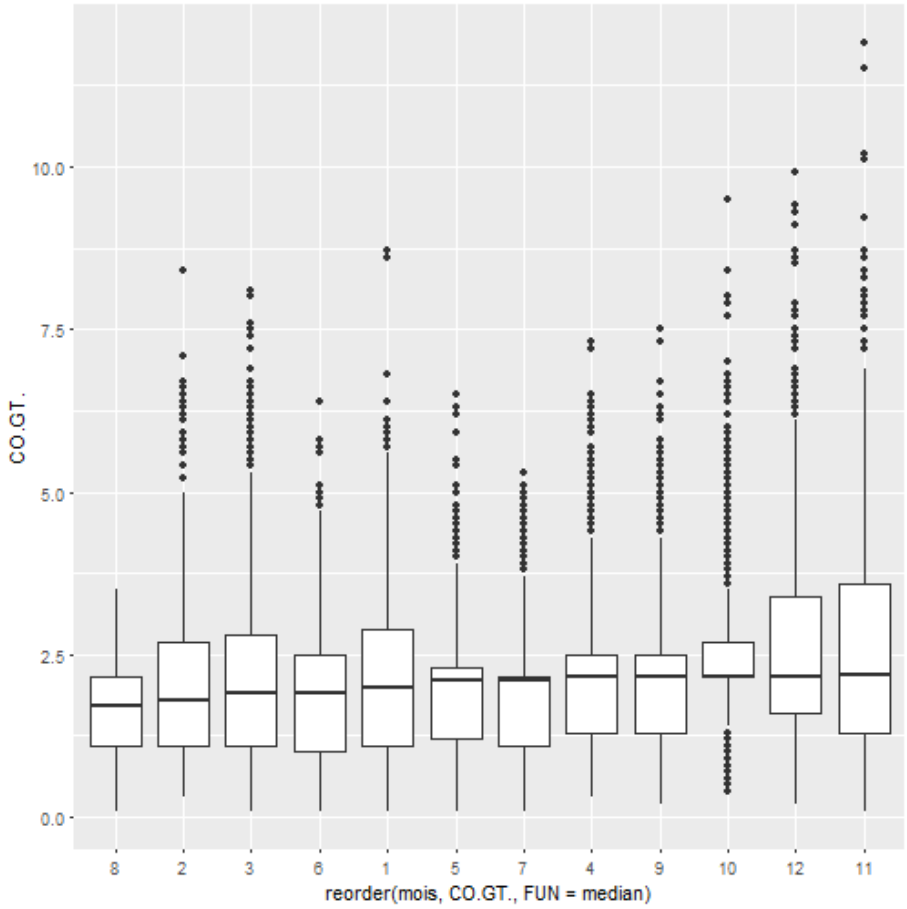

```
data %>%
  count(cut_width(T, 5))
```

```
## # A tibble: 10 x 2
##    `cut_width(T, 5)`     n
##    <fct>             <int>
##  1 [-2.5,2.5]          105
##  2 (2.5,7.5]           946
##  3 (7.5,12.5]         1484
##  4 (12.5,17.5]        1891
##  5 (17.5,22.5]        2114
##  6 (22.5,27.5]        1458
##  7 (27.5,32.5]         716
##  8 (32.5,37.5]         453
##  9 (37.5,42.5]         181
## 10 (42.5,47.5]           9
```

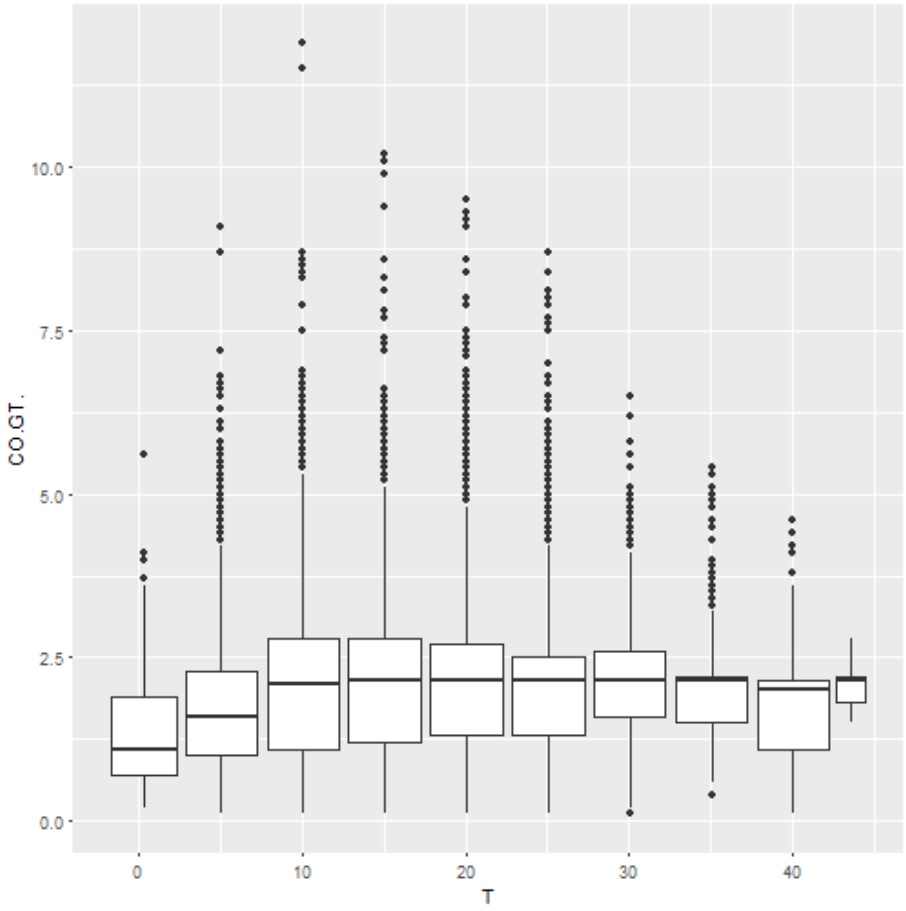
```
petiteT  <-data %>%
  filter(T <3)

ggplot(data = petiteT, mapping = aes(x = T, colour = mois)) +
  geom_freqpoly(binwidth = 0.1)
```




```
ggplot(data =data) +
  geom_boxplot(mapping = aes(x = reorder(mois, CO.GT., FUN = median), y =CO.GT.))
```




```
ggplot(data = data, mapping = aes(x = T, y = CO.GT. )) +
  geom_boxplot(mapping = aes(group = cut_width(T, 5)))
```



##5 REGRESSION LINEAIRE : voyons l'effet de la temperature ,l'humidite, la saison et l'heure sur la concentration du CO en mg/m^3 (CO.GT) en effectuant une regression lineaire. Avant cela on va creer une nouvelle variable 'weekend' qui aura comme valeur TRUE si c'est un jour de weekend et False si c'est un jour de semaine,sachant que les donnees on etÃ© recoltees en 2004 et que le jour de l'an est un jeudi.

```
X=c()
X=(   (data$jour-3)%%7==0      |    (data$jour-3)%%7==1     )
data=mutate(data,weekend=as.numeric(X))
data[,c(1,18)]
```

```
## # A tibble: 9,357 x 2
##    Date       weekend
##    <fct>        <dbl>
##  1 10/03/2004      1.
##  2 10/03/2004      1.
```

```
##  3 10/03/2004      1.
##  4 10/03/2004      1.
##  5 10/03/2004      1.
##  6 10/03/2004      1.
##  7 11/03/2004      1.
##  8 11/03/2004      1.
##  9 11/03/2004      1.
## 10 11/03/2004      1.
## # ... with 9,347 more rows
```

```
Model_lm1=lm(CO.GT.~T+as.factor(weekend)+RH+as.numeric(data$Time),data=data)
summary(Model_lm1)
```

```
##
## Call:
## lm(formula = CO.GT. ~ T + as.factor(weekend) + RH + as.numeric(data$Time),
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1480 -0.8828 -0.1593  0.5154  8.9243
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.3984571  0.0790828   5.038 4.78e-07 ***
## T                      0.0074141  0.0018032   4.112 3.96e-05 ***
## as.factor(weekend)1    0.0148089  0.0281965   0.525    0.599
## RH                     0.0137432  0.0009407  14.610  < 2e-16 ***
## as.numeric(data$Time) 0.0694620  0.0019207  36.165  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.23 on 9352 degrees of freedom
## Multiple R-squared:  0.127,  Adjusted R-squared:  0.1267
## F-statistic: 340.2 on 4 and 9352 DF,  p-value: < 2.2e-16
```

Avant d'interpreter les resultats,on va essayer d'optimiser notre modele: Le critere d'information Akaike (AIC) est une mesure de la qualite relative des modeles statistiques pour un ensemble donne de donnees. Etant donne une collection de modeles pour les donnees, AIC estime la qualite de chaque modele, par rapport Ã chacun des autres modeles. Par consequent, AIC fournit un moyen de selection de modele.

```
Model_lm_best=step(Model_lm1)
```

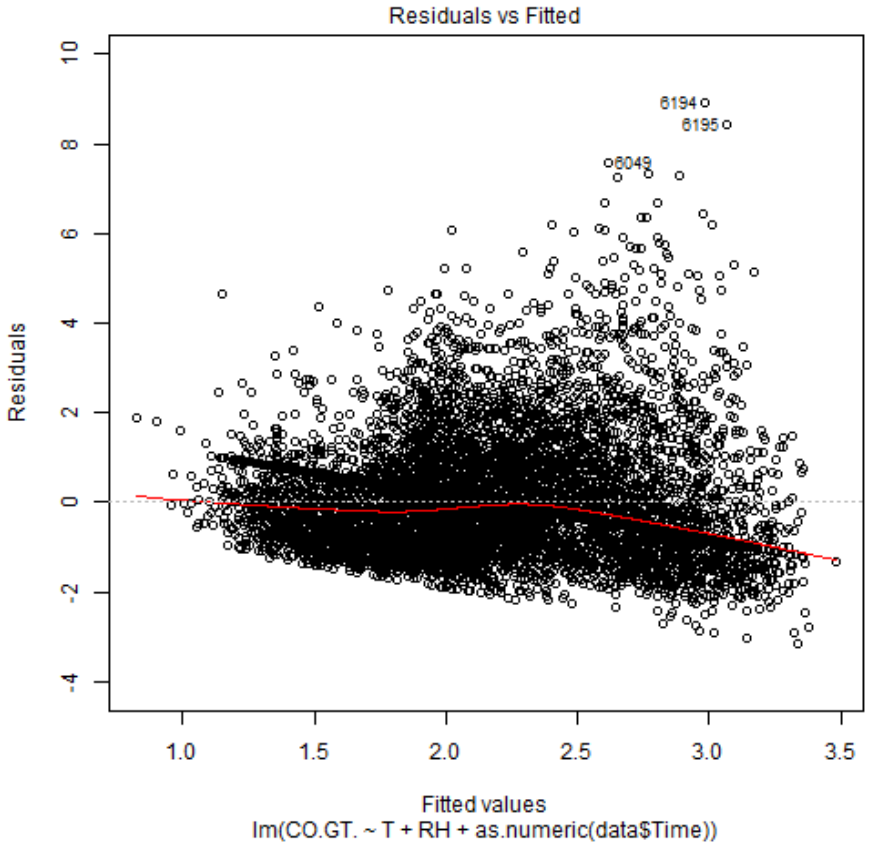
```
## Start:  AIC=3877.47
## CO.GT. ~ T + as.factor(weekend) + RH + as.numeric(data$Time)
##
```

```
##                           Df Sum of Sq   RSS    AIC
## - as.factor(weekend)       1      0.42 14147 3875.7
## <none>                                  14146 3877.5
## - T                        1     25.57 14172 3892.4
## - RH                       1    322.89 14469 4086.6
## - as.numeric(data$Time)    1   1978.34 16125 5100.3
##
## Step:  AIC=3875.75
## CO.GT. ~ T + RH + as.numeric(data$Time)
##
##                           Df Sum of Sq   RSS    AIC
## <none>                                  14147 3875.7
## - T                        1     25.78 14172 3890.8
## - RH                       1    323.11 14470 4085.1
## - as.numeric(data$Time)    1   1978.14 16125 5098.4
```

```
summary(Model_lm_best)
```

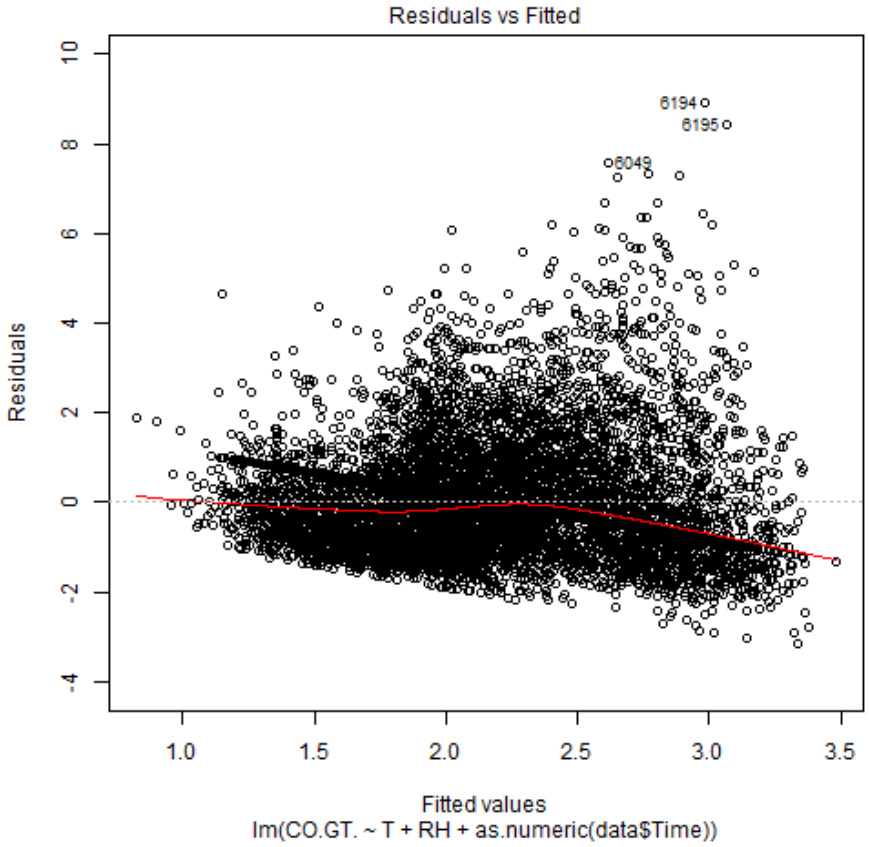
```
##
## Call:
## lm(formula = CO.GT. ~ T + RH + as.numeric(data$Time), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1373 -0.8837 -0.1612  0.5162  8.9202
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.4020279  0.0787869   5.103 3.41e-07 ***
## T                       0.0074408  0.0018024   4.128 3.69e-05 ***
## RH                      0.0137474  0.0009406  14.616  < 2e-16 ***
## as.numeric(data$Time) 0.0694579  0.0019206  36.164  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.23 on 9353 degrees of freedom
## Multiple R-squared:  0.127,  Adjusted R-squared:  0.1267
## F-statistic: 453.6 on 3 and 9353 DF,  p-value: < 2.2e-16
```

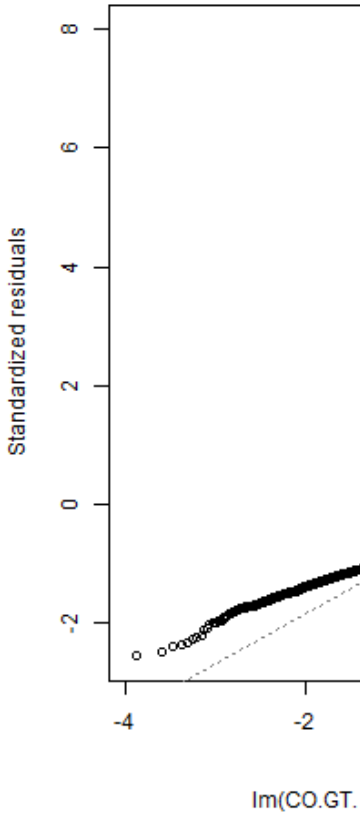
```
plot(Model_lm_best)[1]
```




```
## NULL
```

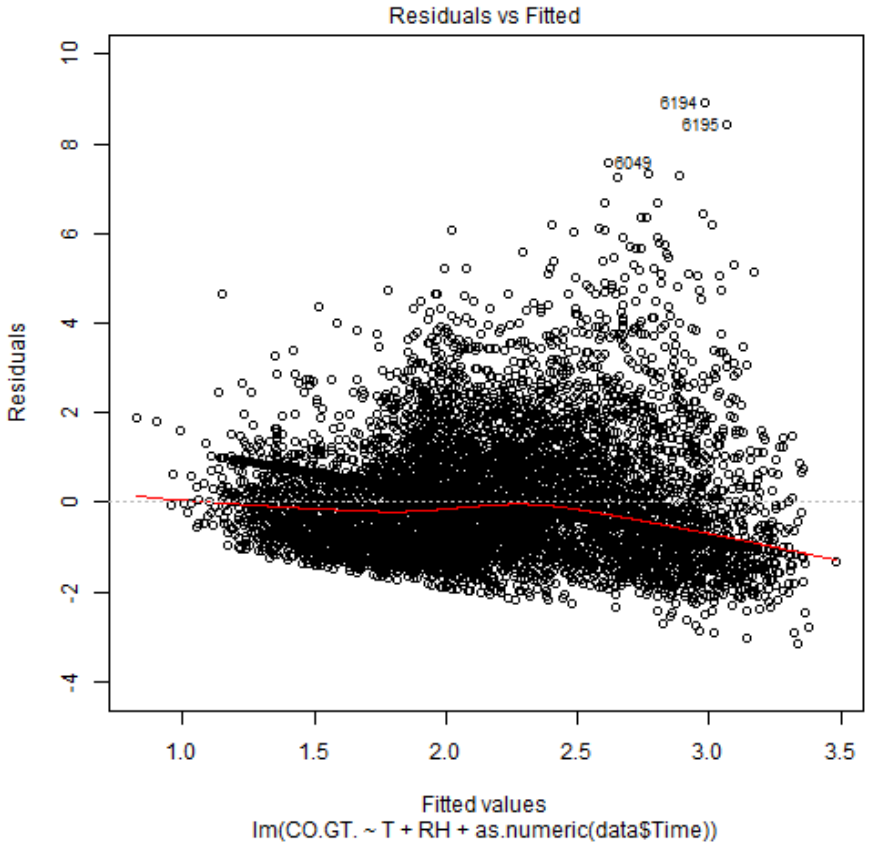
```
plot(Model_lm_best)[2]
```

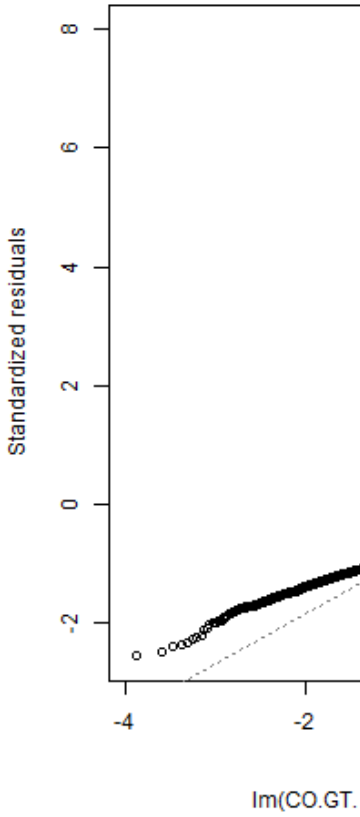





```
## NULL
```

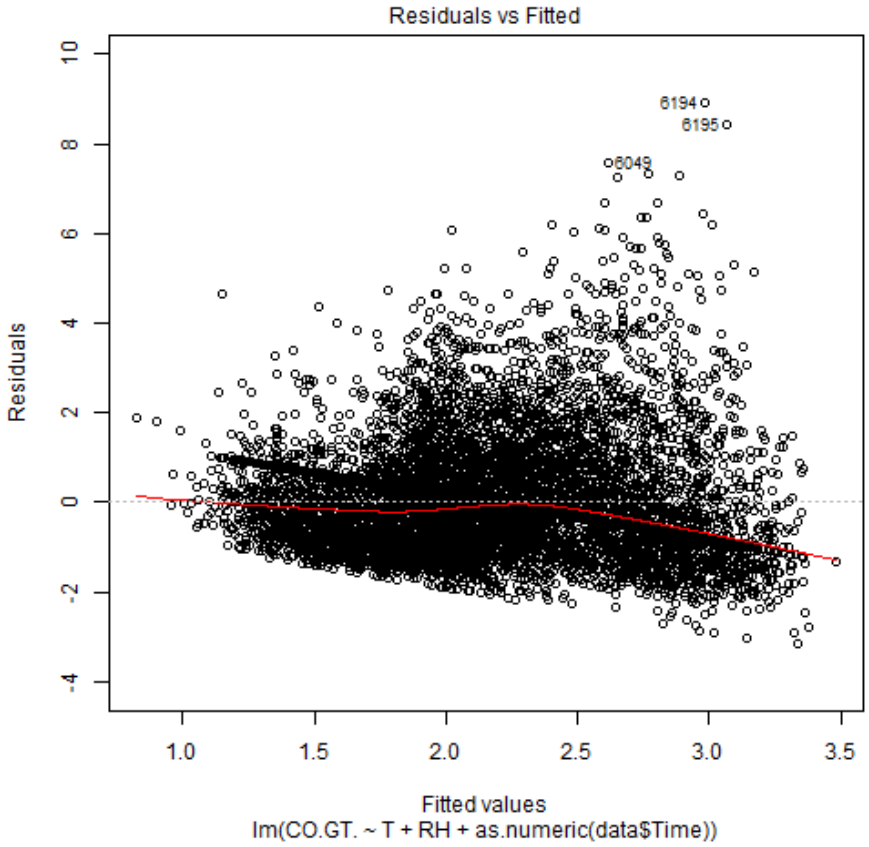
```
plot(Model_lm_best)[3]
```






```
## NULL
```

```
plot(Model_lm_best)[4]
```




```
## NULL
```

On conclue de cette analyse que la concentration du CO dans l'air est influencee par la temperature ,l'humidite et le moment de la journee, en revanche. il n'y a pas d'effet week-end/jr de semaine sur la concentration du CO.