Data analysis of the data regularite-mensuelle-ter

Loading packages

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflic
ts to become errors
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(ggplot2)
library(RColorBrewer)
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
library(klaR)
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(devtools)
```

```
## Loading required package: usethis
```

```
library(patchwork)
```

```
##
## Attaching package: 'patchwork'
##
## The following object is masked from 'package:MASS':
##
##     area
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

Import data and setting graphic settings

```
url = "https://ressources.data.sncf.com/api/explore/v2.1/catalog/datasets/regularite-mensuelle-ter/exports/csv?lang=fr&timezone=Europe%2FBerlin&use_labels=true&delimiter=%3B"

data = read.csv(url, sep = ";", header = TRUE)

mycol = brewer.pal(5, "Set1")
```

Data preparation

```
#Removing missing values
data = na.omit(data)

#Renaming col
colnames(data) = c("date", "region", "nbr_train_prog", "nbr_train_circ", "nbr_train_a
nn", "nbr_train_ret", "tx_reg", "prop", "comm")

#Making the region as factors
data$region = as.factor(data$region)

#Making the date variable as date in R
data$date = as.yearmon(data[,1], format = "%Y-%m")
```

Data exploration

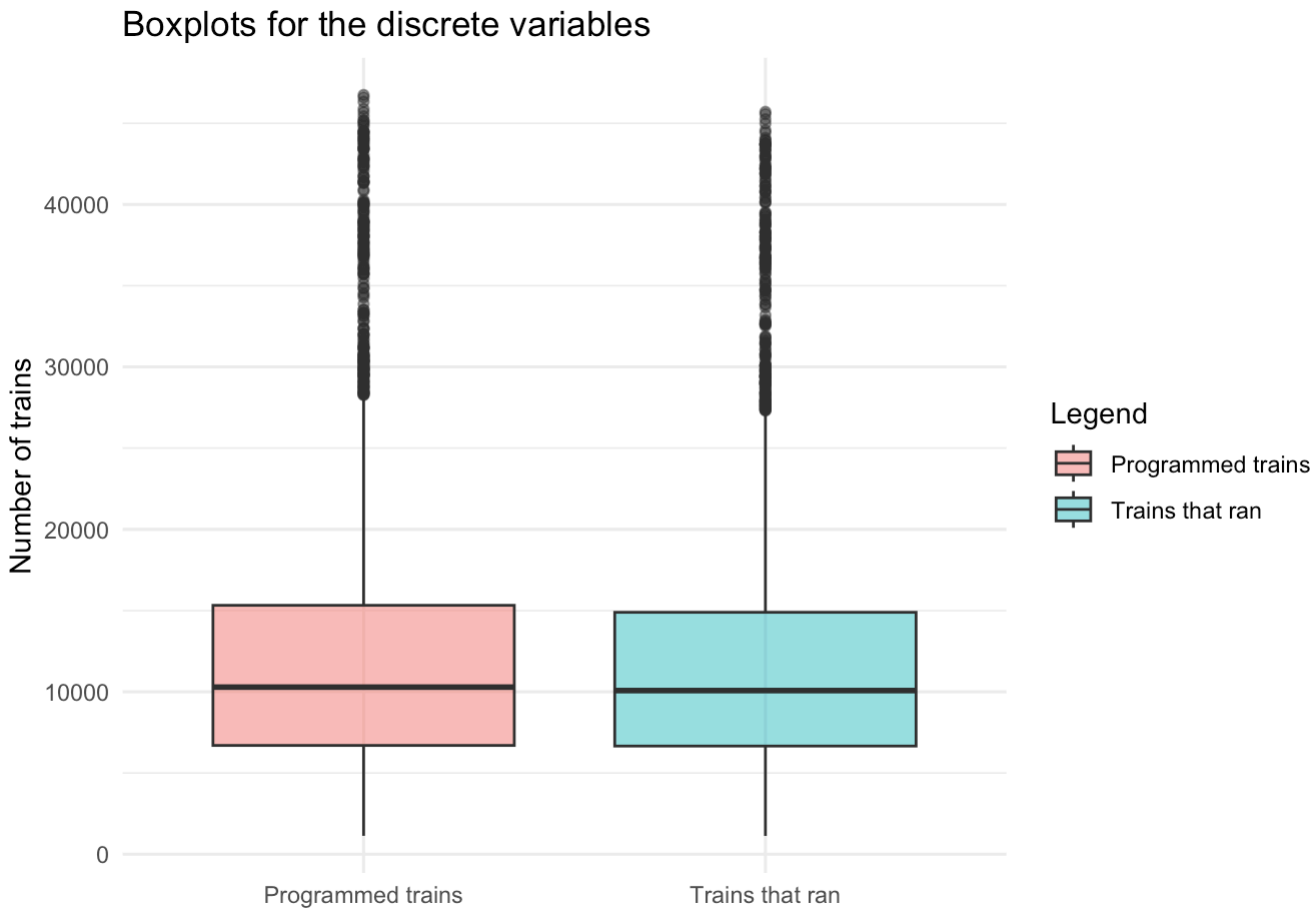
```
#Overview of the data
head(data)
```

```
##       date                     region nbr_train_prog nbr_train_circ
## 1 Jan 2013                  Bourgogne           8400           8332
## 2 Jan 2013           Pays-de-la-Loire          10407          10195
## 3 Jan 2013           Poitou Charentes           3269           3134
## 4 Feb 2013                   Limousin           3449           3406
## 5 Feb 2013           Pays-de-la-Loire           9238           9126
## 6 Feb 2013 Provence Alpes Côte d'Azur          12581          12142
##   nbr_train_ann nbr_train_ret   tx_reg      prop
## 1            68           625 92.49880 12.331200
## 2           212           713 93.00638 13.298738
## 3           135           205 93.45884 14.287805
## 4            43           219 93.57017 14.552511
## 5           112           503 94.48828 17.143141
## 6           439          1761 85.49662  5.894946
##
comm
## 1                                                    Un mois de janvier qui surpass
e les six exercices précédents en termes d'annulation et de ponctualité des trains.
## 2
## 3 Mouvements sociaux des agents du service commercial trains le ASCT le 1er et le
8 janvier. Fortes chutes de neige ayant entrainé des perturbations exceptionnelles.
## 4
## 5
## 6
```

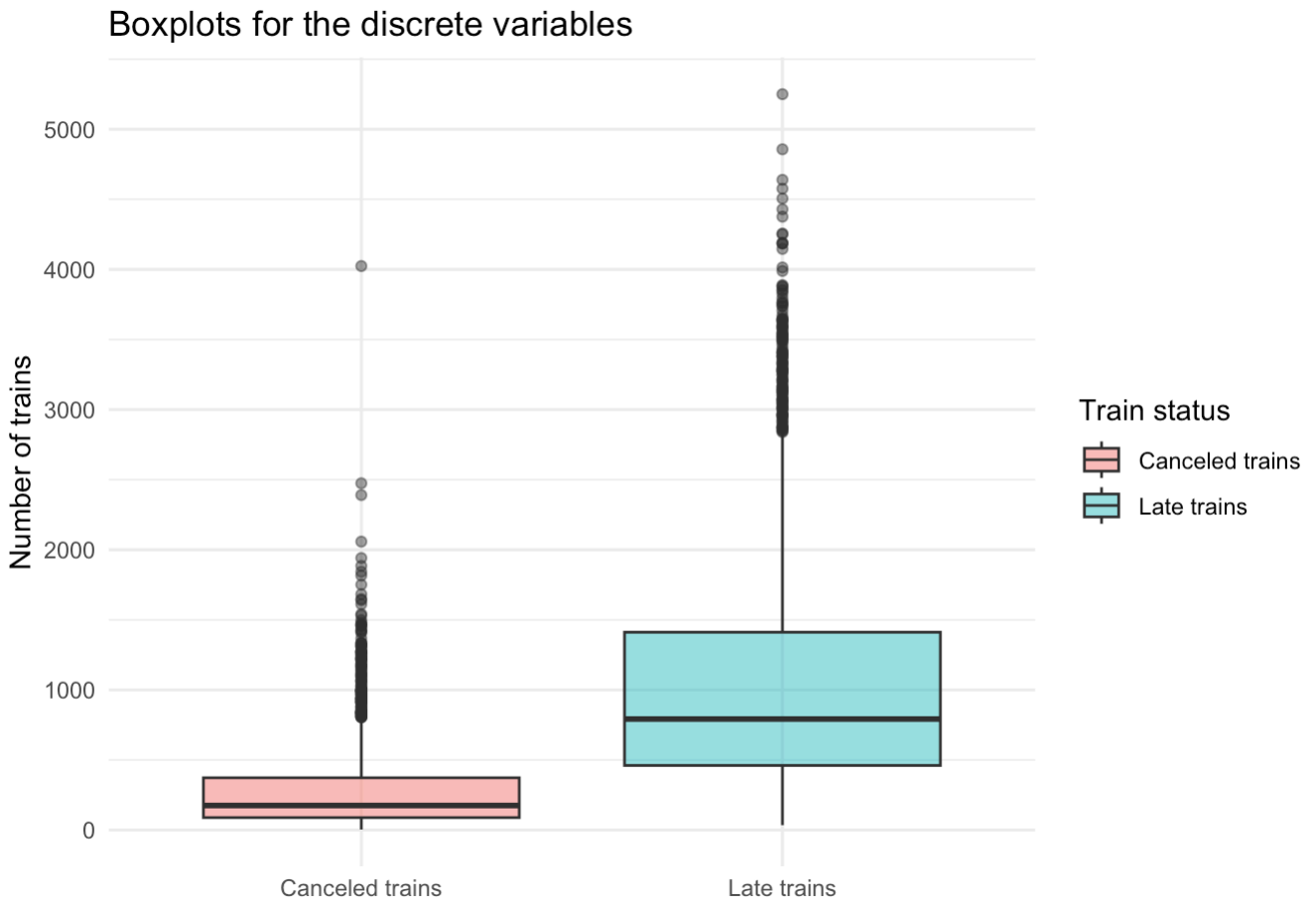
```
#How much data per region?
table(data$region)
```

```
##
##                     Alsace                  Aquitaine
##                         48                         92
##                   Auvergne       Auvergne-Rhône-Alpes
##                         60                         84
##            Basse Normandie                  Bourgogne
##                         60                         60
##    Bourgogne-Franche-Comté                   Bretagne
##                         84                        144
##                     Centre        Centre Val-de-Loire
##                         92                         52
##          Champagne Ardenne              Franche Comté
##                         48                         60
##                  Grand Est            Haute Normandie
##                         96                         60
##            Hauts-de-France       Languedoc Roussillon
##                         84                         60
##                   Limousin                   Lorraine
##                         60                         13
##              Midi Pyrénées         Nord Pas de Calais
##                         66                         58
##                  Normandie         Nouvelle Aquitaine
##                         84                         52
##                  Occitanie           Pays-de-la-Loire
##                         78                        144
##                    Picardie           Poitou Charentes
##                         60                         60
## Provence Alpes Côte d'Azur                Rhône Alpes
##                        144                         60
```

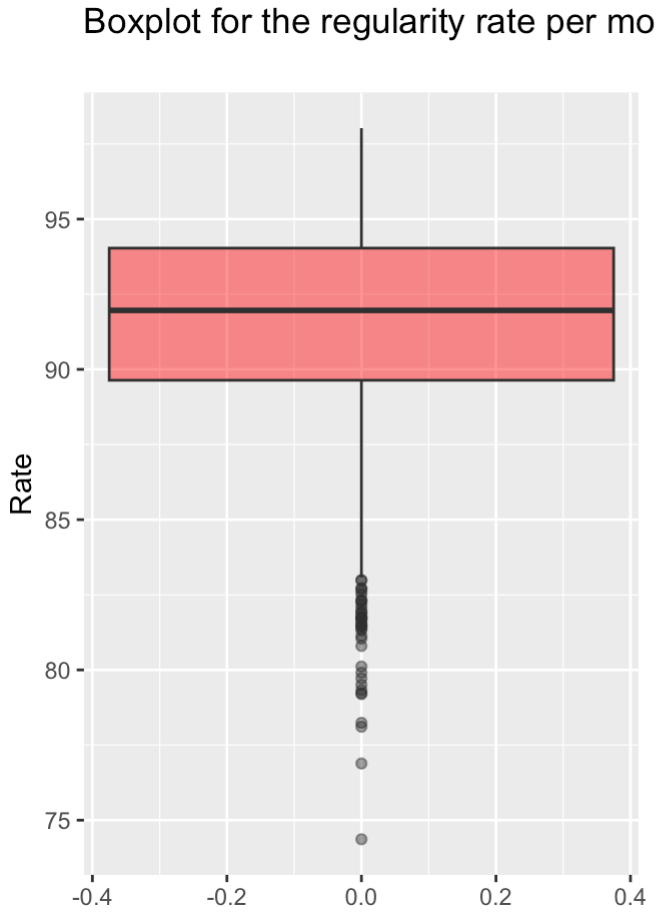
```
#The boxplot for discrete variables
discrete_var = colnames(data)[3:6]

#We put the data in a form that is convenient for the boxplots
data_train = stack(data[,3:4])
data_train$ind = factor(data_train$ind, labels = c("Programmed trains", "Trains that
ran"))

data_late = stack(data[,5:6])
data_late$ind = factor(data_late$ind, labels = c("Canceled trains", "Late trains"))

#The boxplots
ggplot(data_train, aes(x = ind, y = values, fill = ind)) +
  geom_boxplot(alpha=0.5) +
  theme_minimal() +
  labs(title = "Boxplots for the discrete variables",
       y = "Number of trains",
       x = "",
       fill = "Legend")
```

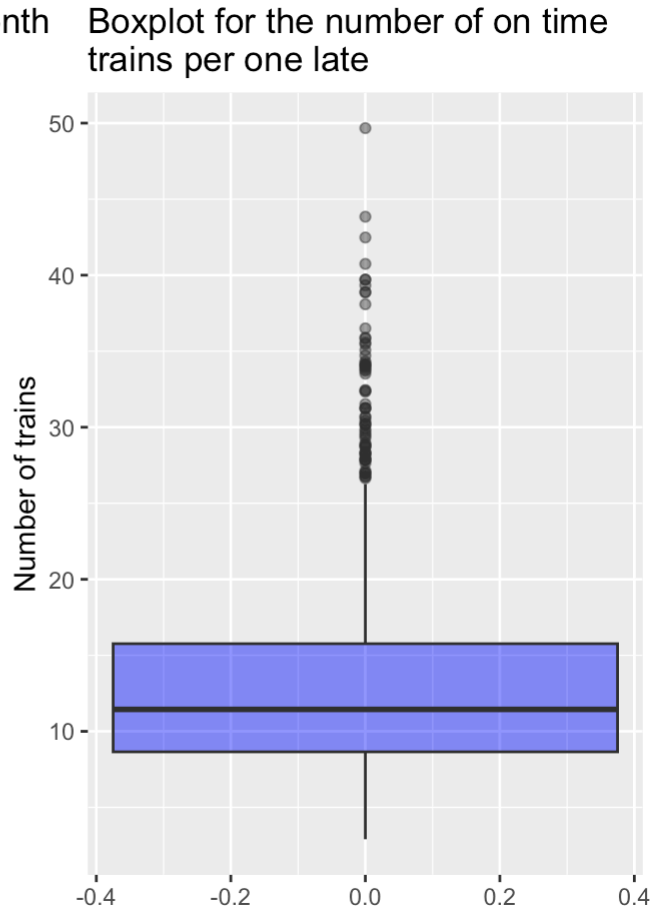



```
ggplot(data_late, aes(x = ind, y = values, fill = ind)) +
  geom_boxplot(alpha=0.5) +
  theme_minimal() +
  labs(title = "Boxplots for the discrete variables",
       y = "Number of trains",
       x = "",
       fill = "Train status")
```




```
#Boxplot for the rates
data_rate = data["tx_reg"]
data_prop = data["prop"]

g1 = ggplot(data_rate, aes(y = tx_reg)) +
  geom_boxplot(alpha = 0.5, show.legend = FALSE, fill = "red") +
  labs(title="Boxplot for the regularity rate per month",
       y = "Rate")

g2 = ggplot(data_prop, aes(y = prop)) +
geom_boxplot(alpha = 0.5, show.legend = FALSE, fill = "blue") +
labs(title="Boxplot for the number of on time\ntrains per one late",
     y = "Number of trains")

g1 + g2
```

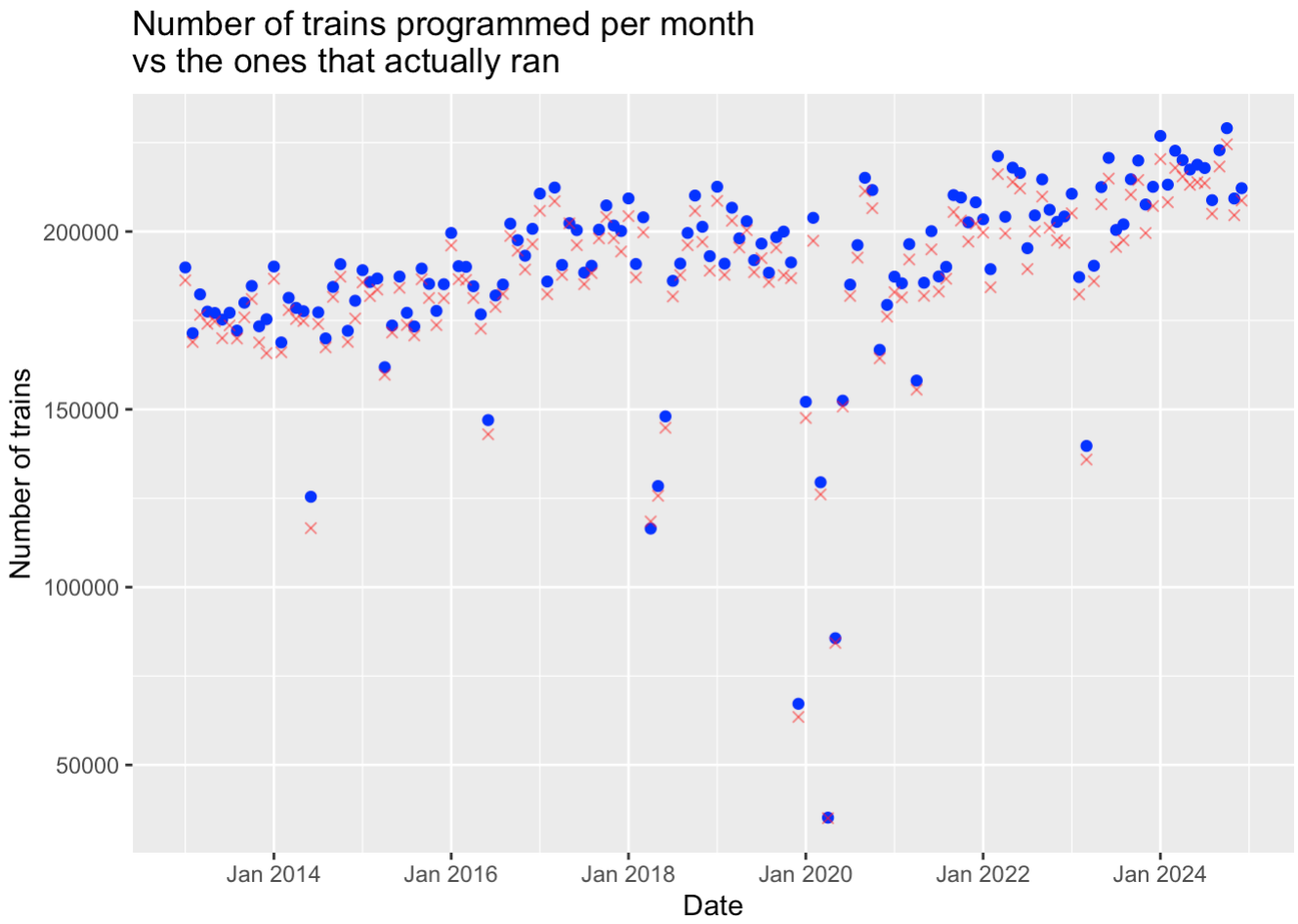





```
#Regrouping the data to have numbers across all region per month
train_prog = data %>%
  group_by(date) %>%
  summarize(train_prog = sum(nbr_train_prog))

train_circ = data %>%
  group_by(date) %>%
  summarize(train_circ = sum(nbr_train_circ))
```

The graph

```
#Points graph to visualize the difference between the trains that were programmed and
the ones that ran
ggplot(train_prog, aes(x=date, y=train_prog)) +
  geom_point(col = "blue") +
  geom_point(data = train_circ, aes(x=date,y=train_circ), col = "red", pch=4, alpha=
0.5) +
  labs(title = "Number of trains programmed per month\nvs the ones that actually ra
n",
       x = "Date",
       y = "Number of trains")
```



For the PACA region

For the following, we will be focusing in the PACA region

```
#Filter vector
data_paca = data[data$region == "Provence Alpes Côte d'Azur",]
data_paca = data_paca[,-2] #Removing the region column
head(data_paca)
```

```
##         date nbr_train_prog nbr_train_circ nbr_train_ann nbr_train_ret   tx_reg
## 6   Feb 2013          12581          12142           439          1761 85.49662
## 11  Mar 2013          13994          13042           952          2010 84.58825
## 34  Jul 2013          13918          13276           642          2762 79.19554
## 45  Sep 2013          13024          12465           559          1745 86.00080
## 105 Aug 2014          14404          13842           562          2228 83.90406
## 118 Apr 2013          13437          12168          1269          1660 86.35766
##         prop
## 6   5.894946
## 11  5.488557
## 34  3.806662
## 45  6.143266
## 105 5.212747
## 118 6.330120
##
comm
## 6
## 11  La non fiabilité dans la restitution de travaux amène l'activité à transférer
sur route les premiers trains qui suivent l'heure théorique de restitution des travau
x.
## 34
## 45
## 105
## 118                                                 Un éboulement dans un tunnel suit
e à des intempéries entraîne la suppression des trains sur une partie de leur parcour
s.
```

It could be interesting to investigate the mean number of late trains per year

```
#Summing the data for each year of late trains for the paca region
data_paca_year = data_paca %>%
  group_by(year = lubridate::year(date)) %>%
  summarize(nbr_train_prog = mean(nbr_train_prog), nrb_train_circ = mean(nbr_train_ci
rc), nbr_train_ann = mean(nbr_train_ann), nbr_train_ret = mean(nbr_train_ret))

head(data_paca_year)
```

```
## # A tibble: 6 × 5
##    year nbr_train_prog nrb_train_circ nbr_train_ann nbr_train_ret
##   <dbl>          <dbl>          <dbl>         <dbl>         <dbl>
## 1  2013         13313.         12337           976.         2083.
## 2  2014         13489.         12760.          728.         2332.
## 3  2015         15188.         14627           562.         2559.
## 4  2016         14541          14060.          481.         2125.
## 5  2017         15037.         14534.          503.         2324.
## 6  2018         13755.         13173.          582.         1769.
```

Data analysis

In this next section, I thought it could be interesting to perform an ANOVA to test whether the differences between the late trains of the years 2021 up to 2024 are significant

To this end, we have to explicit our variables IV : independent variable => the year, qualitative ordinal variable (4 modalities : 2021, 2022, 2023, 2024) DV : dependent variable => number of late trains, quantitative discrete variable

Null hypothesis H0 : there are no difference between the number of late trains of the years 2021 up to 2024 Alternative hypothesis H1 : there is a difference between the number of late trains of the years 2021 up to 2024

Before diving into the test, we have to curate our data

```
#Keeping only the date in our data using year function of the lubricate package
data_paca_aov = data.frame(date = year(data_paca$date), nbr_train_ret = data_paca$nbr
_train_ret)
data_paca_aov = data_paca_aov[order(data_paca_aov$date),]
data_paca_aov = data_paca_aov[data_paca_aov$date>=2021,]
head(data_paca_aov)
```

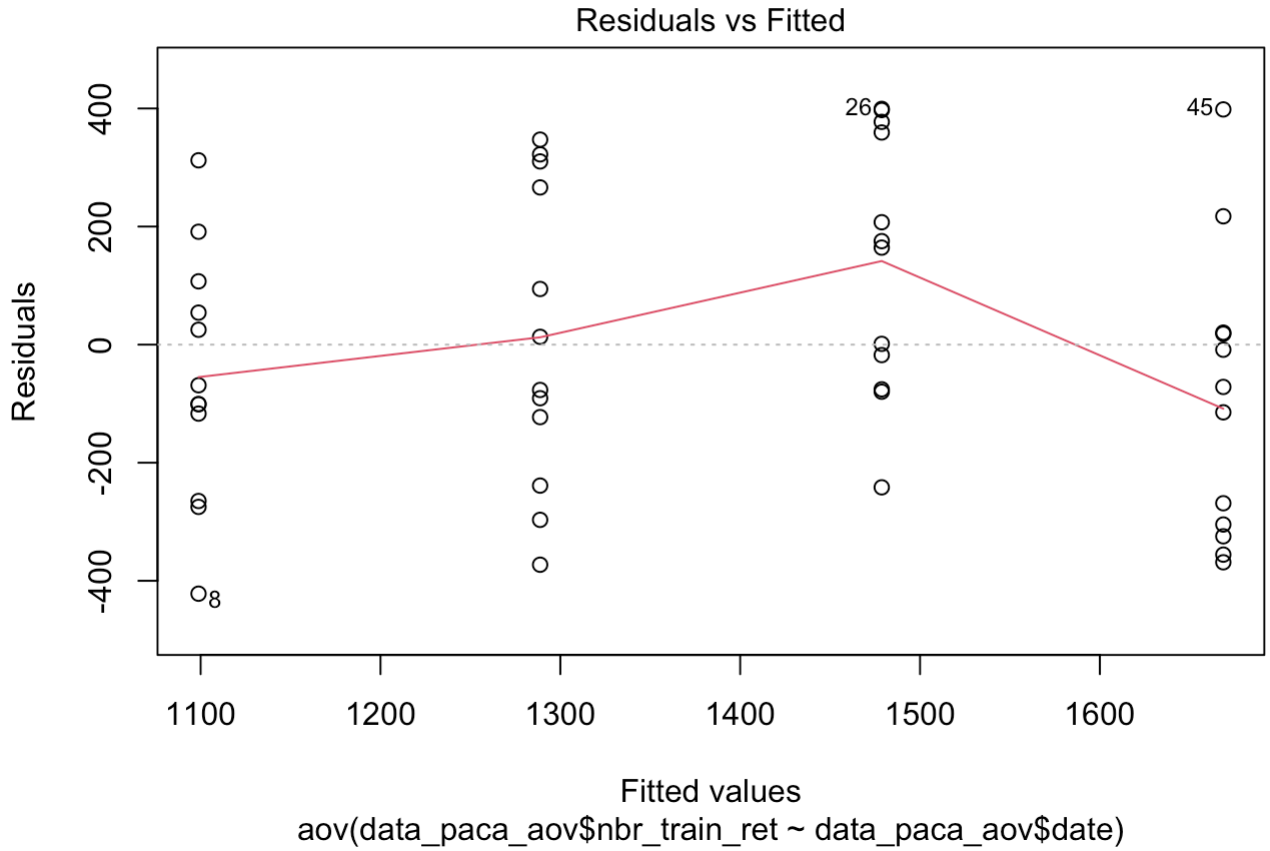
```
##    date nbr_train_ret
## 30 2021          1030
## 31 2021           824
## 32 2021           834
## 35 2021          1124
## 37 2021          1290
## 40 2021           982
```

```
#All is done, we can do the anova
anova = aov(data_paca_aov$nbr_train_ret~data_paca_aov$date)

summary(anova) #according to this, the difference is very strongly significant
```

```
##                   Df  Sum Sq Mean Sq F value   Pr(>F)
## data_paca_aov$date  1 2164670 2164670    36.3 2.66e-07 ***
## Residuals          46 2743219   59635
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

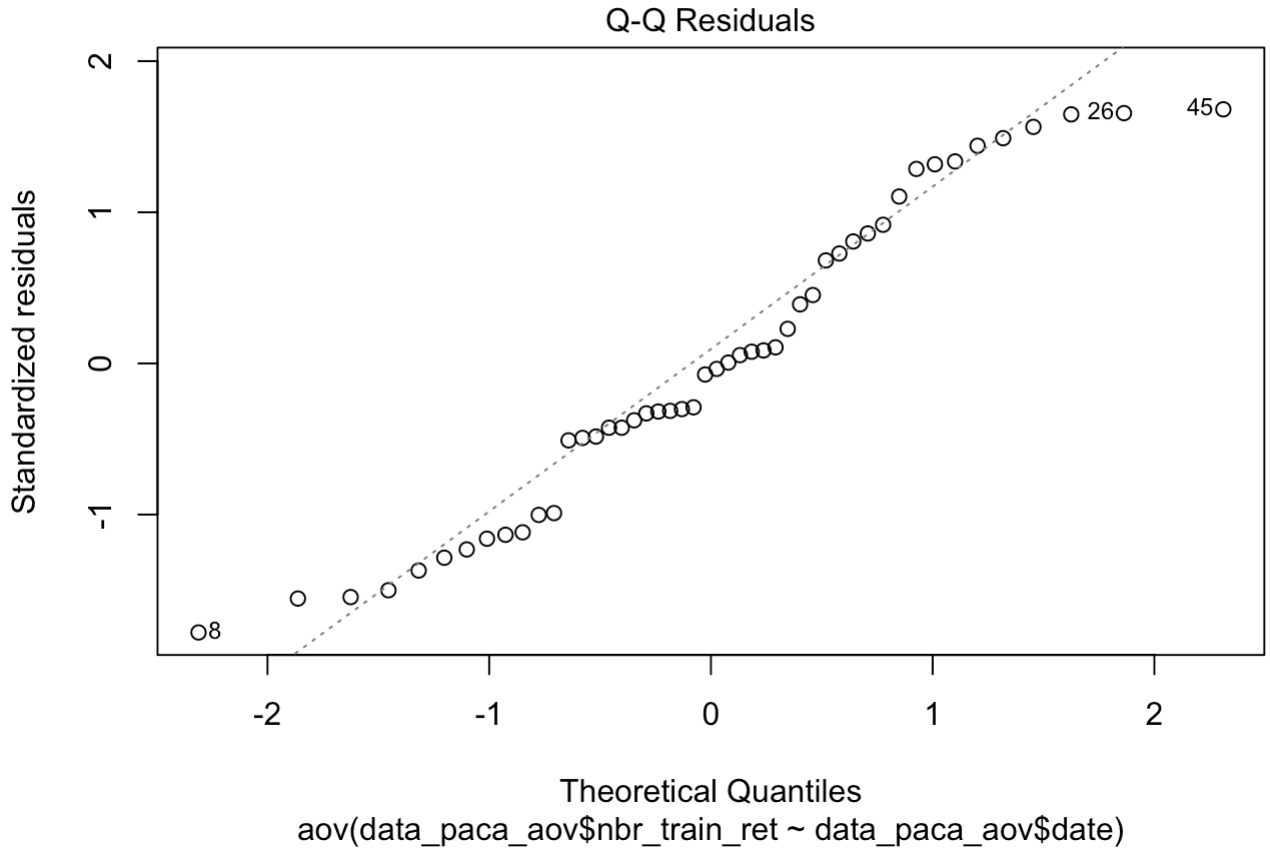
Residuals independence

```
#visual checking of the absence of correlation
plot(anova, 1)
```



Normal distribution of the residuals

```
#The residuals do not seem to be normally distributed (not following the line). Becau
se the p value of the shapiro test is only a little below 0.05, we question the valid
ity of our test
plot(anova, 2)
```




```
shapiro.test(anova$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  anova$residuals
## W = 0.95094, p-value = 0.0436
```

Linear regression

```
#Getting the linear regression from python, we want to know if the coefficients are s
ignificant to assess whether the regression is relevant or not
#=>As we can see, the R2 coefficient is quite low, 0.3039, the regression might not b
e that relevant
linear_reg_paca = lm(data_paca$nbr_train_prog~data_paca$nbr_train_ret)
summary(linear_reg_paca)
```

```
##
## Call:
## lm(formula = data_paca$nbr_train_prog ~ data_paca$nbr_train_ret)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7877.7  -594.5   298.3   969.6  2685.4
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.039e+04  4.414e+02  23.535  < 2e-16 ***
## data_paca$nbr_train_ret 1.896e+00  2.381e-01   7.964 4.87e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1677 on 142 degrees of freedom
## Multiple R-squared:  0.3088, Adjusted R-squared:  0.3039
## F-statistic: 63.43 on 1 and 142 DF,  p-value: 4.871e-13
```