Spatio-Temporal Data Analysis Project

2020-05-12

Patterns in foreign sims connected to OpenWiFi-Milan

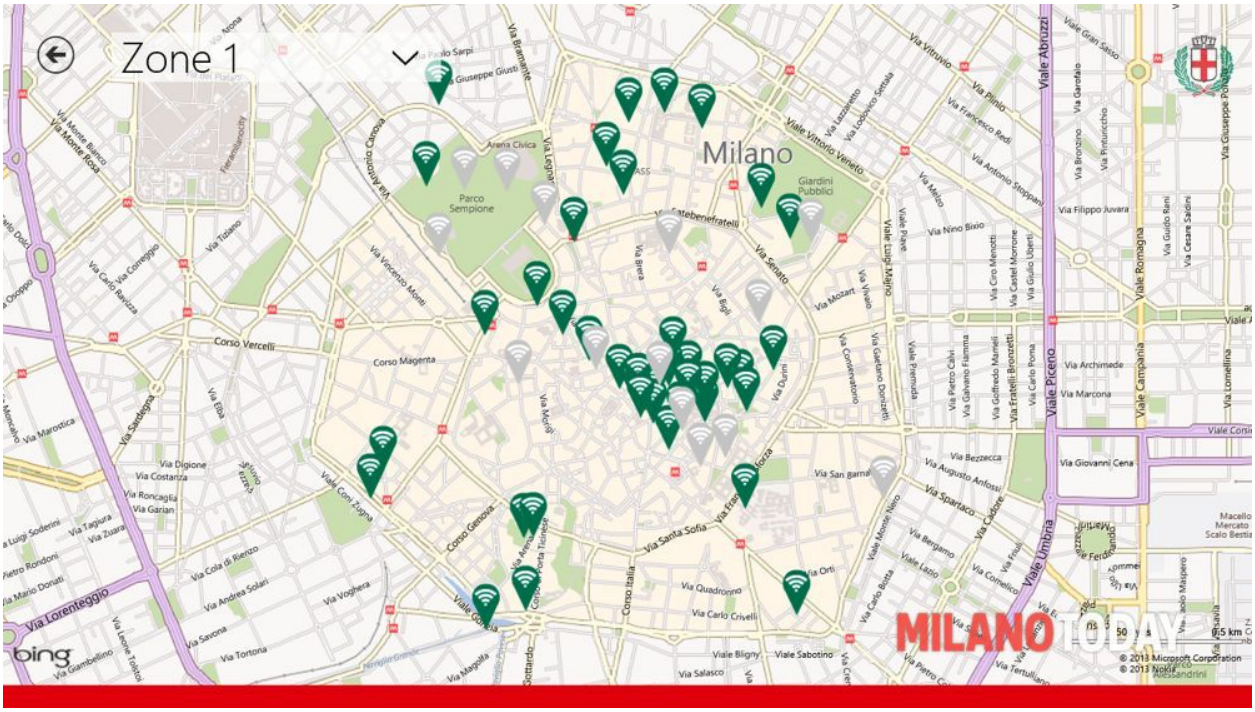
Author: Bernardi Riccardo - 864018

Professor: Isadora Antoniano-Villalobos

Contents

1 Introduction & Motivation

The project is about finding if some patterns are present in the way foreign people connects to the wifi of the city of Milan in Italy. This wifi was installed in the first days of august of the 2014 but the data if from the 5^{th} of July od the 2015. It was installed by the municipality of the city and it is public but it is only available in some areas of the city. We can see here below that the areas covered are the most attractive from the point of view of a tourist so for this reason we can believe that this is a good proxy for the number of people in the city.



It permits to a user that is logged in to use the free wifi for a maximum time of 60 minutes and 300MB of downloaded data. These restrictions are huge for a people living and working there but probably for a tourist that remains few days it can be enough. Obviously the wifi was created in a time in which the telecom companies were digging gold with high prices on internet connection but at the time I'm writing(year 2020) the fees are much much lower and the roaming no more exists. FO all these reasons we can agree with the fact that as the time

goes on the public wifi is going to be abandoned. This comes easily by the fact that all the people will be able to afford an internet connection on the smartphone.[1]

Now I'm going to tell the reader how works the data we have: we have two columns, the first one is the day in which the relevation occurred and the second column is about how many sim cards from foreign people were connected. I would like to let the reader knows that no NANs are present and there is exactly one observation per day. These facts are good because the time serie is easier to be analyzed if all the data is present, if some data was missing then are needed complex assumptions that can be also not valid. Th enumber of sims(the second column) is about the number of sims from all the possible countries in the world. We know that each sim is uniquely identified by the system so if the second columns tells us that 12 sims are connected in a certain date it means that exactly 12 unique and different sims are connected. These are all good facts but we cannot state that all the sims are independent one from the others, for example a group of tourists or a fmaily coming to visit the city should be counted as only one element or more? Until now they are counted separately since every sim is identified uniquely. Another problem that insist on the dataset is that there is no way to know in which part of the city the sims connected.

Is interesting to analyse this kind od data? Obviously yes! The city of Milan was the first city providing a free wifi and it is already now the only one that provides the relative data in the form of open data. This kind of initiatives in Italy are pretty rare so it worths to be studied. We should also remind that in the city are present many boutiques that are of great interest for the foreign people, public events about fashion, luxury and design, music events on the beaches near to Milan and so on. After these also we should remind that in the city is also present the italian stock exchange market and so it can be really interesting to investigate if a certain then in the city is linked to some events in the relative stock exchange market.

2 Data Inspection

The dataset comes from the open data provided by all the municipalities of Milan. This repository is available at dati.gov.it. From this repository I selected the data going from June of 2017 to October of the 2019.

Characteristics of the DataSet:

- the dataset contains 2 columns "Date, Number_of_Foreign_Sims"
- has 868 rows
- Dates goes from from 05/06/17 to 30/10/19 (~2 years)
- the datasets have no NA
- no lacking days
- the "Number_of_Foreign_Sims" is a discrete variable about total number of foreign sims in a certain Date connected to the OpenWifi of Milan

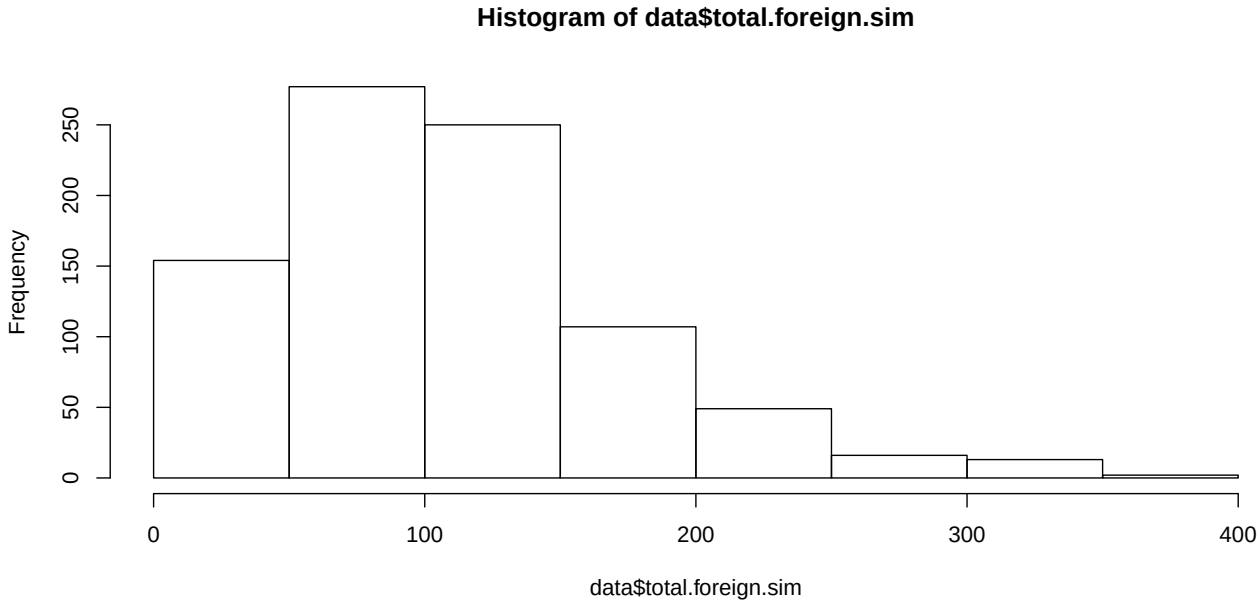
Here we can read which are the most important numbers for the data, these are called the Tukey's five numbers and they are the minimum that is also important to check if there are errors in the data since it should be greater or equal to zero. Zero seems infeasible and it can be a NAN. The data we can read here below are not bad in the sense that no negative values

are present but we can note that the space spanned by maximum - minimum is very large, probably the time serie suffers of great variability. The variability is a measure to calculate the volatility of a stock option. If the volatility is high this means the analysis will be more difficult for us since the stationarity is an assumptions for the model we are going to see in the next chapters.

```
## [1] "minimum, lower-hinge, median, upper-hinge, maximum)"
```

```
## [1]   1.0  61.5 101.0 141.0 378.0
```

We can check here the histogram of the data and we can see that the data is a bit skewed. In the technical jergon skewness is a measure of the fact that the data in the histogram is not symmetric over a certain accumulation point. For us the accumulation point is the mean and we can see it approximatively on the 100. We can note that the right tail is longer than the left tail. The skewness is not an appreciable characteristic for the time serie since it means that calculating the mean can have no sense and no assumptions on gaussianoty is feasible. Skewness also means that probaly constant variance assumption is not feasible and so no stationarity assumption can hold. These are all difficult facts to deal with. What we can do to cope with this problem? We can try to stabilize the data trying some transformations and checking which is the best one to have a more symmetrical distribution.



Here we are going to check the Nans and as we can see they are not present, this is a good fact because it means that there is non need of doing assumptions on the way missing values behave.

```
## number of nans in the date column:  0
```

```
## number of nans in the foreign sims column:  0
```

The mean of the values as stated before id around 100 and it is confirmed here below but it is useless to calculate it if the distribution is skewed.

```
## mean of the foreign sims column:  109.9228
```

Also the variance is calculated here:

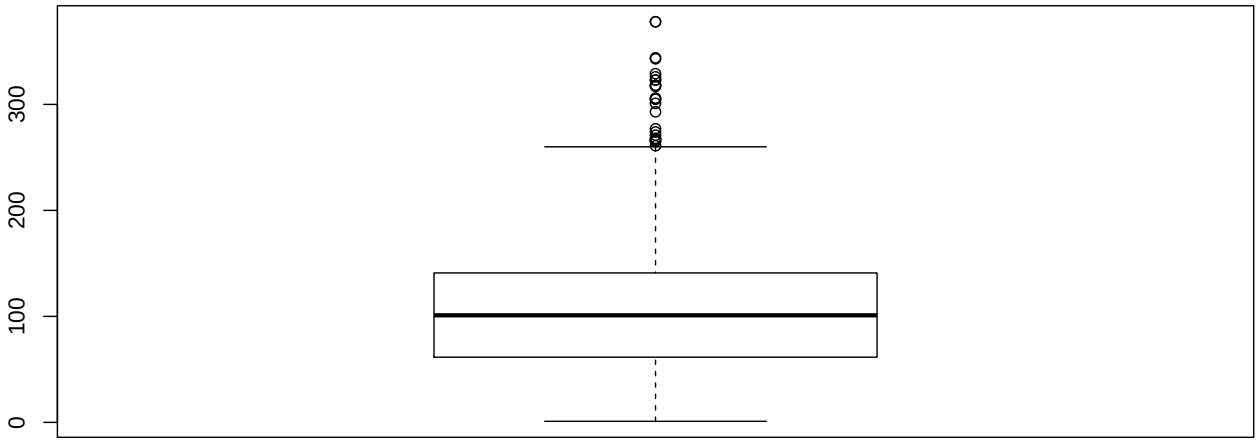
```
## mean of the foreign sims column:  63.63468
```

Elements that are good in our ts stand between mean±std

```
## upper confidence value of the foreign sims column:  173.5575
```

```
## lower confidence value of the foreign sims column:  46.28813
```

We are going here to check the outliers with the boxplot, also called the whisker's plot. It is useful for example here in this case to see that there are many outliers on the upper part and no outliers on the lower part. This mean that there are many off-scale values that are too big. This is not a certainty but the boxplot is a good tool for outliers analysis. It is applied on the column of the foreign sims.



Foreign sims

We know from an introspection of the data that at the end of the time serie are present some near-to-zero data due to some errors in the measuring way so we are going to eliminate them.

```
## here the data to investigate on: 39 78 91 39 1 2 2
```

```
## old length of the vector: 868
```

```
## here the data to be deleted: 1 2 2
```

```
## new length of the vector: 865
```

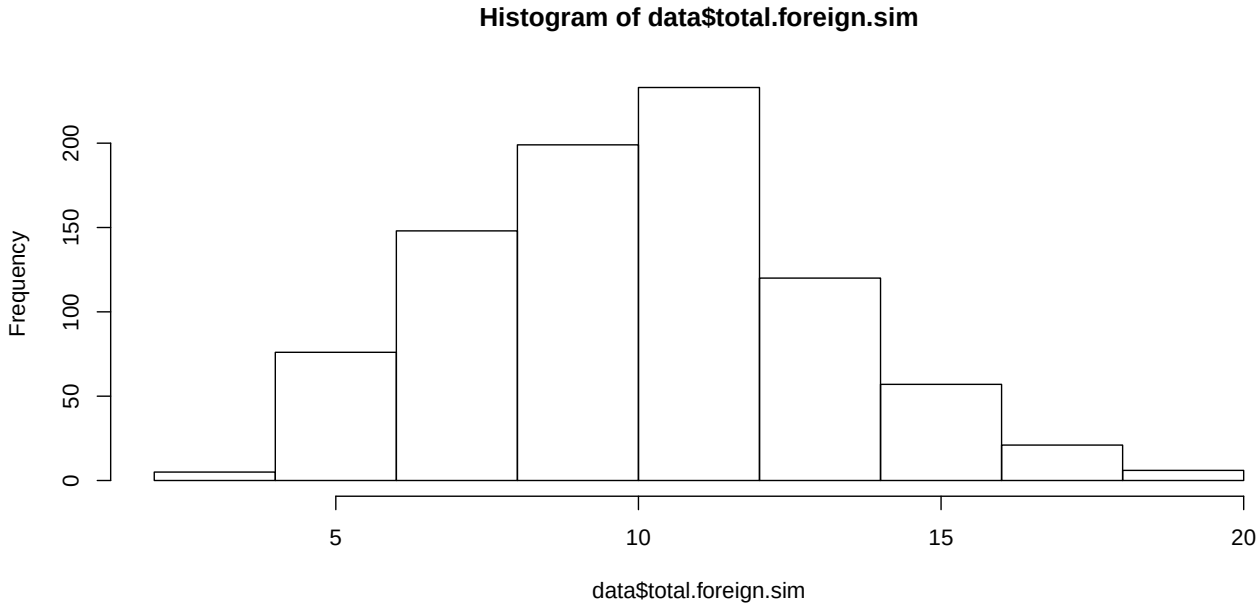
```
## the new last3 elements are: 78 91 39
```

We choose here to transform the data from the original scale to the square root scale. The transformation was chosen after many empirical tries with the most promising ones.

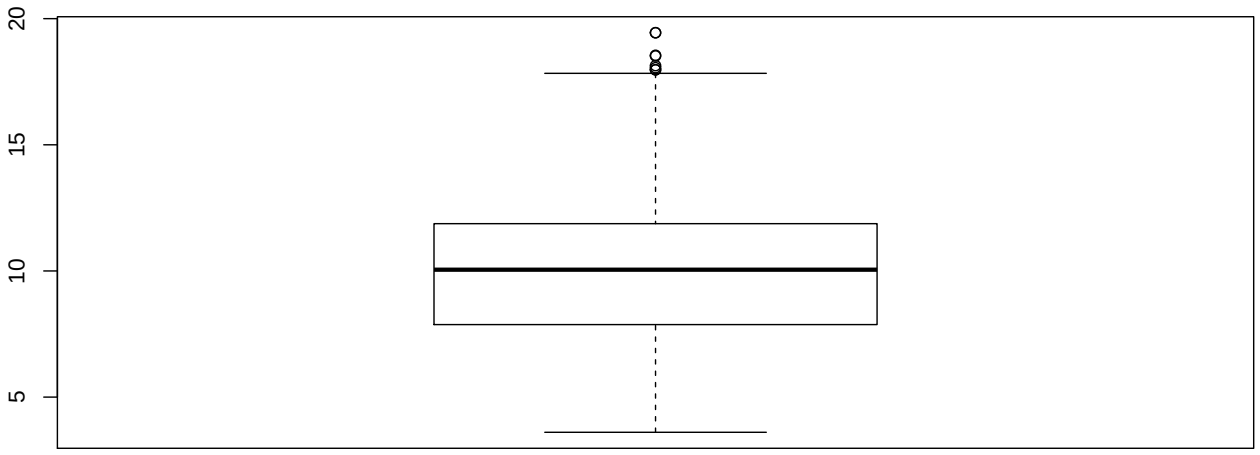
```
## before the transformation(data [0:5]): 73 139 174 97 156
```

```
## after the transformation(data [0:5]): 8.544004 11.78983 13.19091 9.848858 12.49
```

The hist after the transformation here below. As we can see it in better than before, in the sense that now it is bell-shaped so calculating the mean and the standard deviation is meaningful. The only requirement is that at the end the reult is put to the power of two since this is a transformation of the original data.



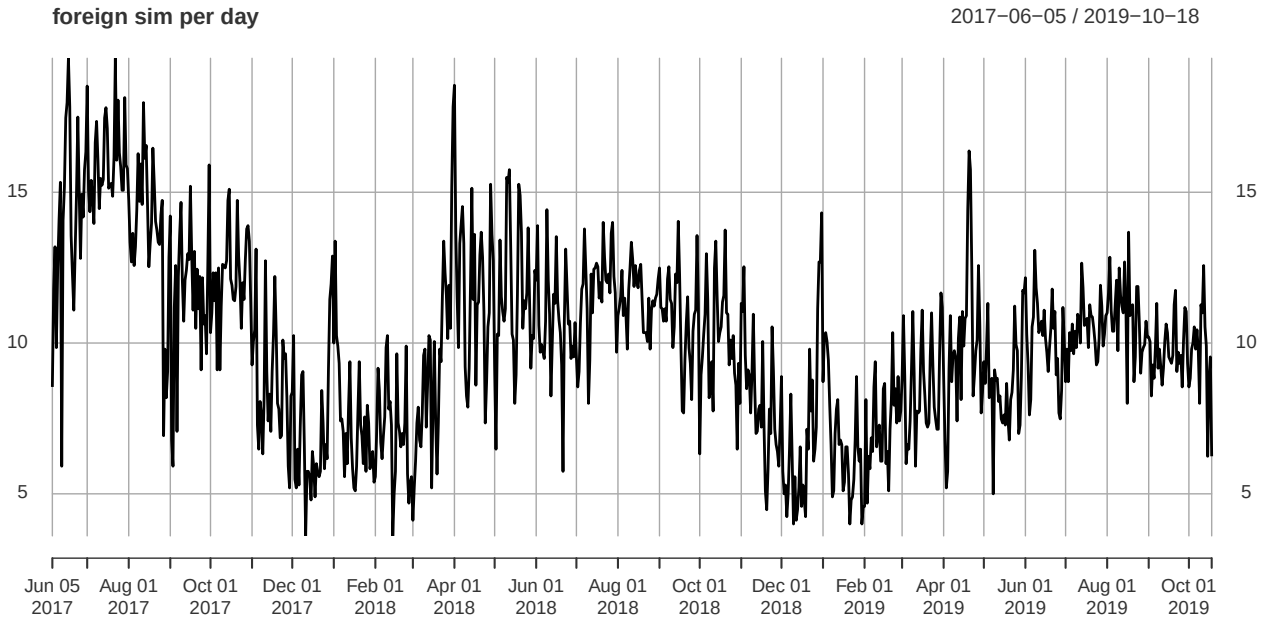
Now the boxplot is better because the overall number of outliers is less.



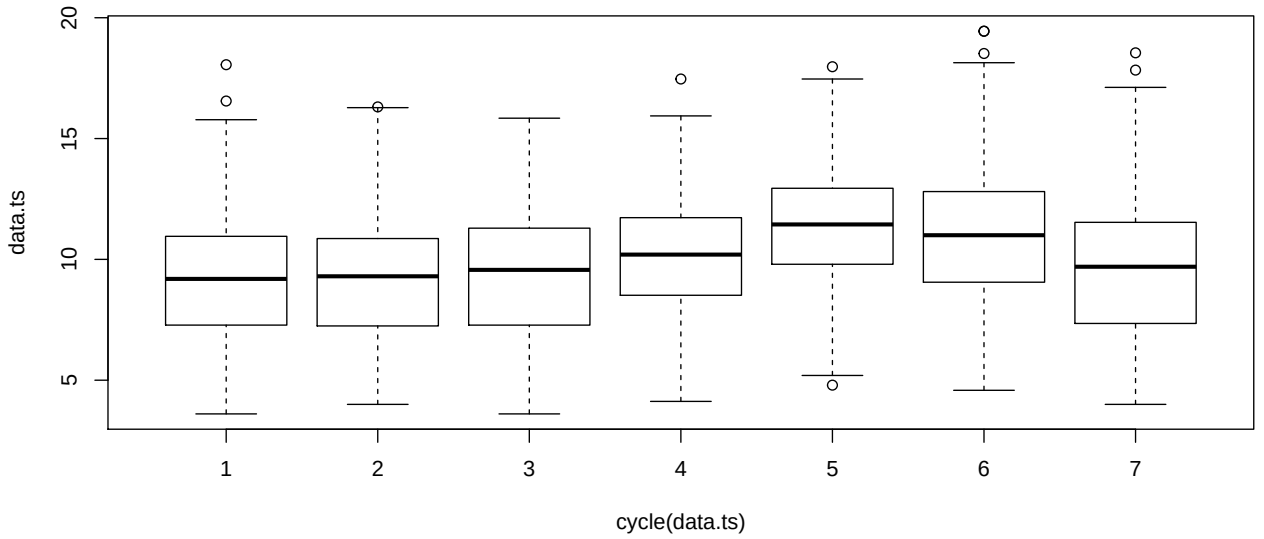
3 Time serie is built

Here the time serie is built

We loaded the dataset from the various datasets aggregating into only one dataset with 655 rows representing 2 years of data gathered. Starting from 05.06.2017 to 30.10.2019. Data is here:



As we can see the greatest number of foreign come after the mid of the week



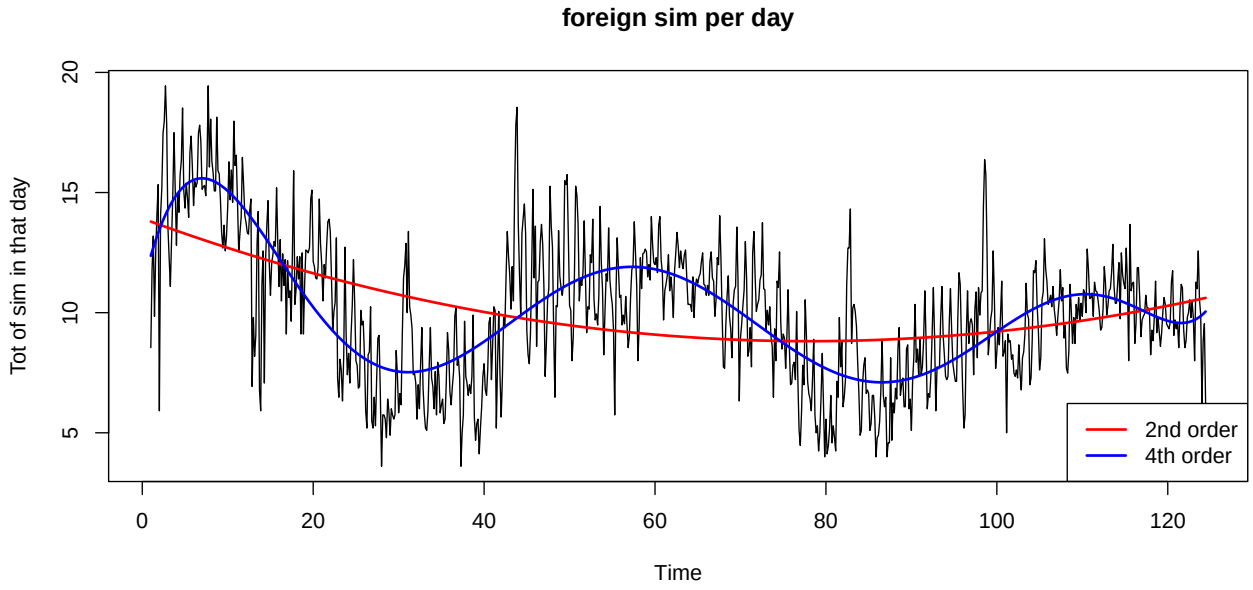
4 Peaks Explanation

Many peaks are present we would like to exaplin them and to cut them out to be able to predict with a simple arima

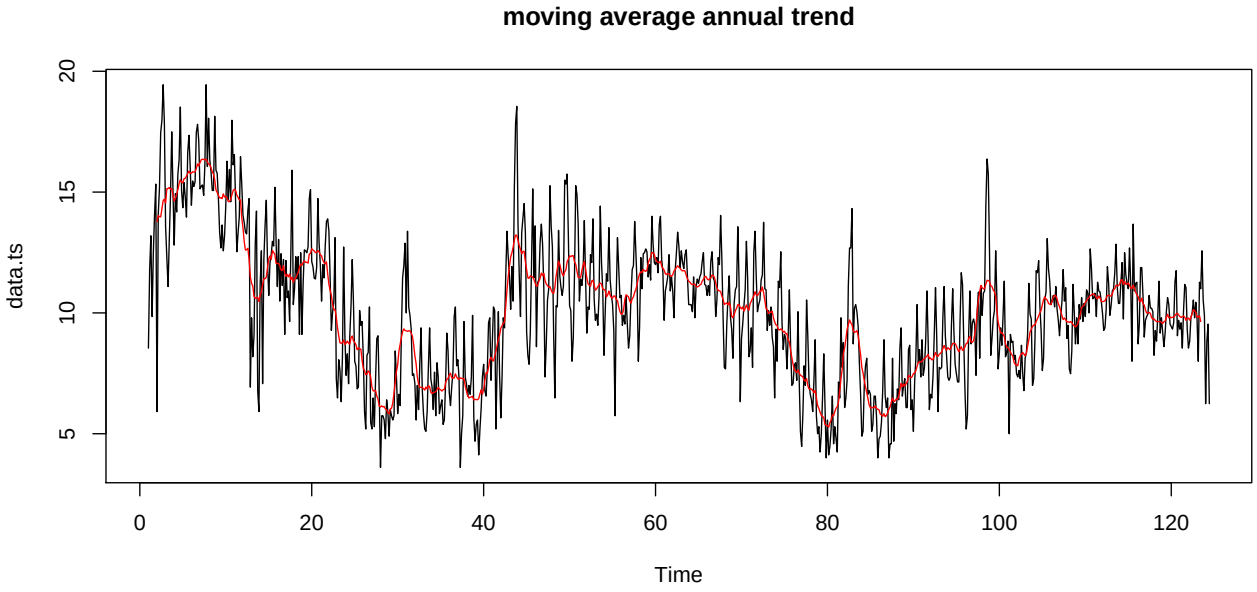
- automatic roaming [2]
- fashion week [3] february
- fashion week 2017 [4] february
- arch week [5]
- it was a saturday[6]

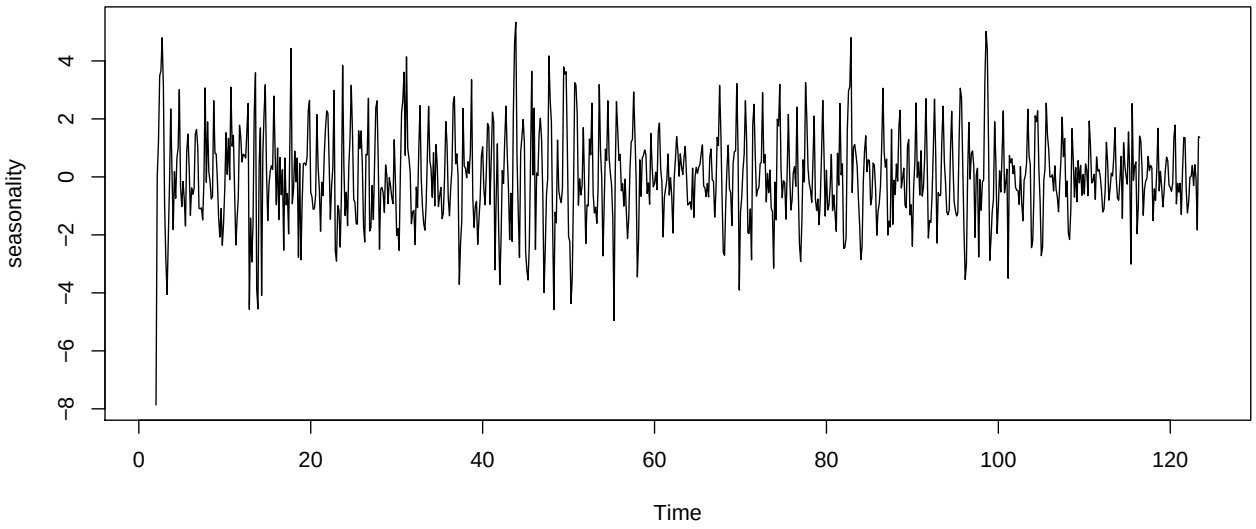
- it was the orient festival [6]
- many mucis events, samsara of papetee and others, folk's festivals, discounts [7]

5 Trend recognition



5.1 Smoothing





5.2 Splitting

6 Models

We tried many models, the most known is the arma but we tried: - arma - arima - sarima - var - rugarch - fgarch

and others, i'm reporting here only the best one and some ideas for the worst performing ones.

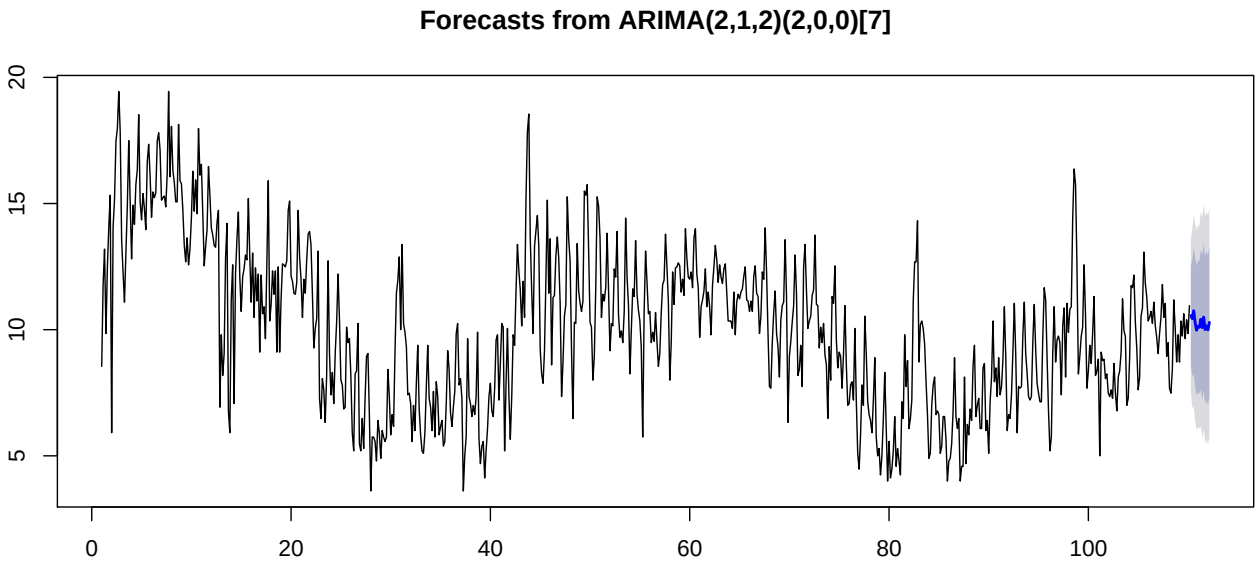
6.1 Arima

To find the best arima in a such complicate time serie we are going to exploit a grid search algorithm done via the auto arima function provided by the fpp package

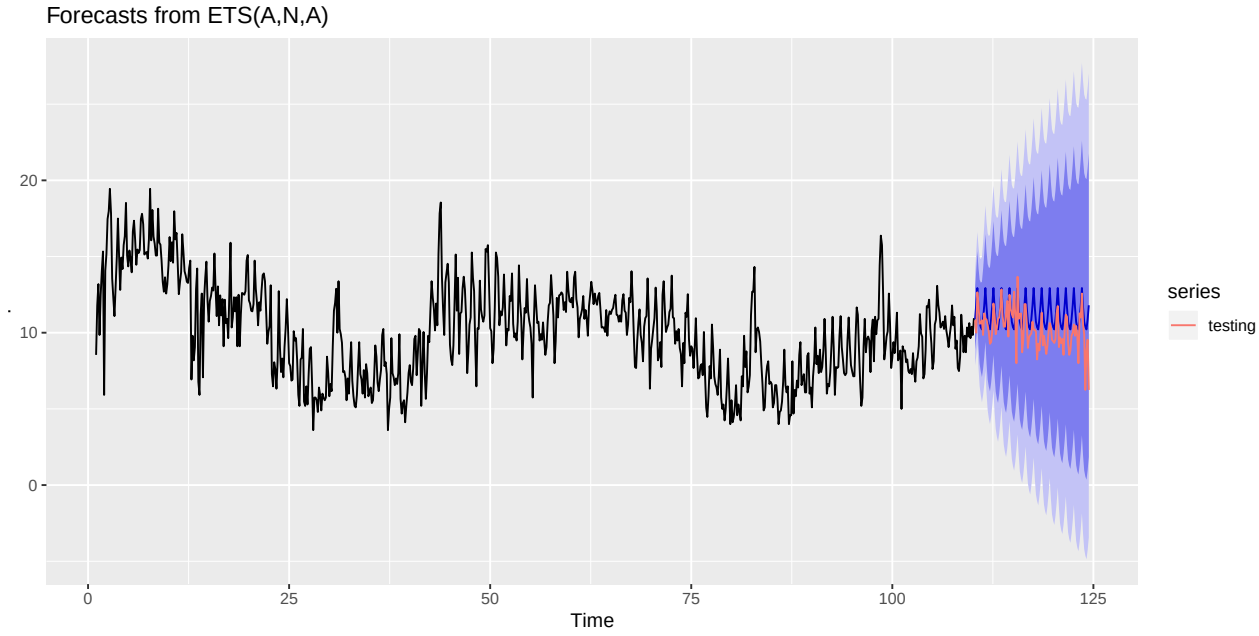
```
##
##  ARIMA(2,1,2)(1,0,1)[7] with drift         : Inf
##  ARIMA(0,1,0)           with drift         : 3150.243
##  ARIMA(1,1,0)(1,0,0)[7] with drift         : 3011.566
##  ARIMA(0,1,1)(0,0,1)[7] with drift         : 3043.261
##  ARIMA(0,1,0)                              : 3143.607
##  ARIMA(1,1,0)           with drift         : 3137.948
##  ARIMA(1,1,0)(2,0,0)[7] with drift         : 2968.604
##  ARIMA(1,1,0)(2,0,1)[7] with drift         : Inf
##  ARIMA(1,1,0)(1,0,1)[7] with drift         : Inf
##  ARIMA(0,1,0)(2,0,0)[7] with drift         : 3021.782
##  ARIMA(2,1,0)(2,0,0)[7] with drift         : 2963.8
##  ARIMA(2,1,0)(1,0,0)[7] with drift         : 3011.443
##  ARIMA(2,1,0)(2,0,1)[7] with drift         : Inf
##  ARIMA(2,1,0)(1,0,1)[7] with drift         : Inf
##  ARIMA(3,1,0)(2,0,0)[7] with drift         : 2944.413
```

```
##  ARIMA(3,1,0)(1,0,0)[7] with drift         : 2982.233
##  ARIMA(3,1,0)(2,0,1)[7] with drift         : Inf
##  ARIMA(3,1,0)(1,0,1)[7] with drift         : Inf
##  ARIMA(4,1,0)(2,0,0)[7] with drift         : 2948.849
##  ARIMA(3,1,1)(2,0,0)[7] with drift         : 2895.063
##  ARIMA(3,1,1)(1,0,0)[7] with drift         : 2922.809
##  ARIMA(3,1,1)(2,0,1)[7] with drift         : Inf
##  ARIMA(3,1,1)(1,0,1)[7] with drift         : Inf
##  ARIMA(2,1,1)(2,0,0)[7] with drift         : 2888.797
##  ARIMA(2,1,1)(1,0,0)[7] with drift         : 2921.181
##  ARIMA(2,1,1)(2,0,1)[7] with drift         : Inf
##  ARIMA(2,1,1)(1,0,1)[7] with drift         : Inf
##  ARIMA(1,1,1)(2,0,0)[7] with drift         : 2889.113
##  ARIMA(2,1,2)(2,0,0)[7] with drift         : 2887.637
##  ARIMA(2,1,2)(1,0,0)[7] with drift         : 2904.112
##  ARIMA(2,1,2)(2,0,1)[7] with drift         : Inf
##  ARIMA(1,1,2)(2,0,0)[7] with drift         : 2889.281
##  ARIMA(3,1,2)(2,0,0)[7] with drift         : 2896.624
##  ARIMA(2,1,3)(2,0,0)[7] with drift         : 2893.838
##  ARIMA(1,1,3)(2,0,0)[7] with drift         : 2895.647
##  ARIMA(3,1,3)(2,0,0)[7] with drift         : 2900.109
##  ARIMA(2,1,2)(2,0,0)[7]                    : 2881.177
##  ARIMA(2,1,2)(1,0,0)[7]                    : 2897.672
##  ARIMA(2,1,2)(2,0,1)[7]                    : Inf
##  ARIMA(2,1,2)(1,0,1)[7]                    : Inf
##  ARIMA(1,1,2)(2,0,0)[7]                    : 2882.866
##  ARIMA(2,1,1)(2,0,0)[7]                    : 2882.375
##  ARIMA(3,1,2)(2,0,0)[7]                    : Inf
##  ARIMA(2,1,3)(2,0,0)[7]                    : 2887.398
##  ARIMA(1,1,1)(2,0,0)[7]                    : 2882.612
##  ARIMA(1,1,3)(2,0,0)[7]                    : 2889.217
##  ARIMA(3,1,1)(2,0,0)[7]                    : 2888.624
##  ARIMA(3,1,3)(2,0,0)[7]                    : 2893.686
##
##  Best model: ARIMA(2,1,2)(2,0,0)[7]
```

A forecast on the training set looks like this one below:



Here we are going to plot a forecast on a part of the data that was never seen by the arima model, the plots looks not so bad.

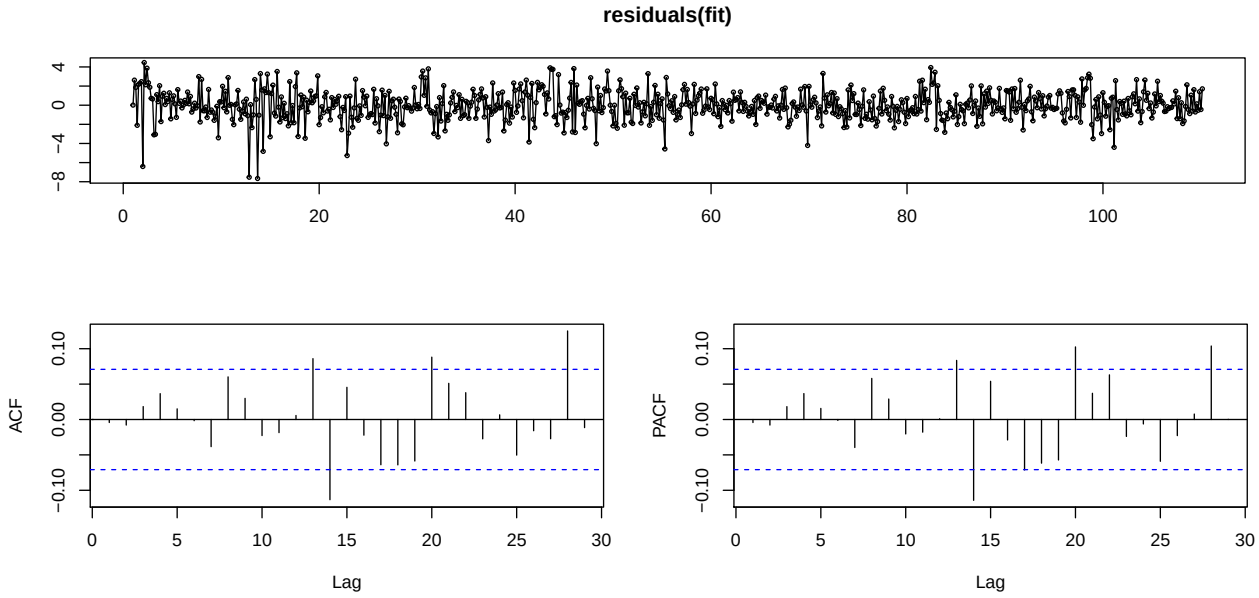


The accuracy on the test set is only two percentages points lower than on the training set. We are using 100 points.

```
##                       ME     RMSE       MAE         MPE      MAPE      MASE
## Training set -0.00785033 1.543384 1.1670287 -2.69172229 13.001429 0.6867895
## Test set      0.16977738 1.243377 0.9628649  0.04924476  9.671012 0.5666404
##                    ACF1 Theil's U
## Training set -0.004270222        NA
## Test set      0.286712600 0.8326098
```

Actually here below we can see that residuals are not pretty good but after many weeks

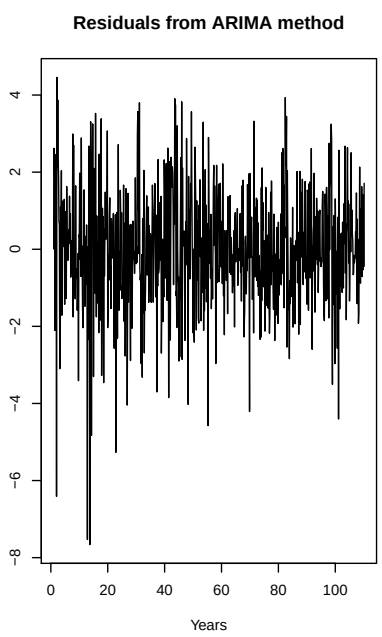
of attempts with many models, all the poossible seasonalities, all the decomposition, all the tricks available i'm pretty sure that this is the best trade-off between complexity of the models used, accuracy and processing of the dataset.

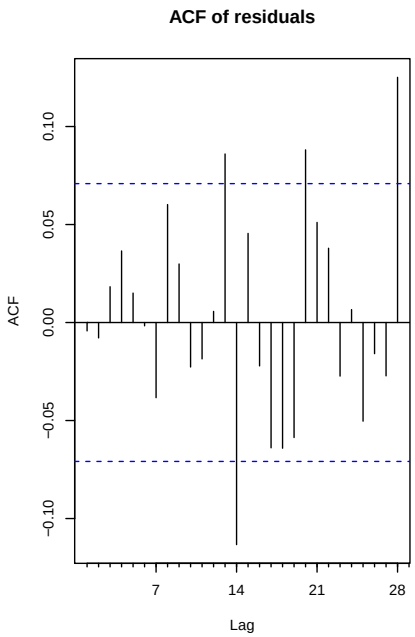


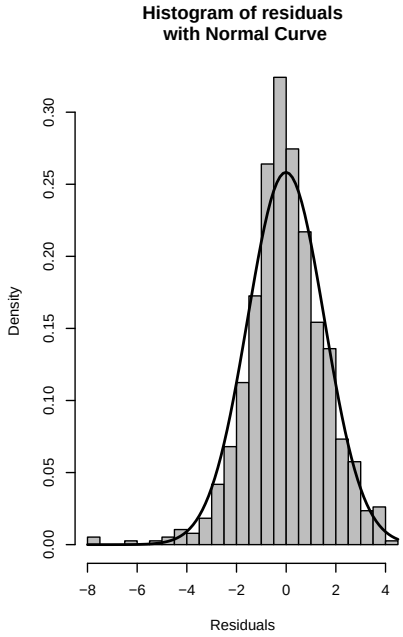

```
##
##  Box-Ljung test
##
## data:  residuals(fit)
## X-squared = 43.725, df = 24, p-value = 0.008212
```

```
##
##  Box-Pierce test
##
## data:  residuals(fit)
## X-squared = 42.744, df = 24, p-value = 0.01064
```

The diagnostic here below confirms at least that the auto arima chose the right differentiation parameter.

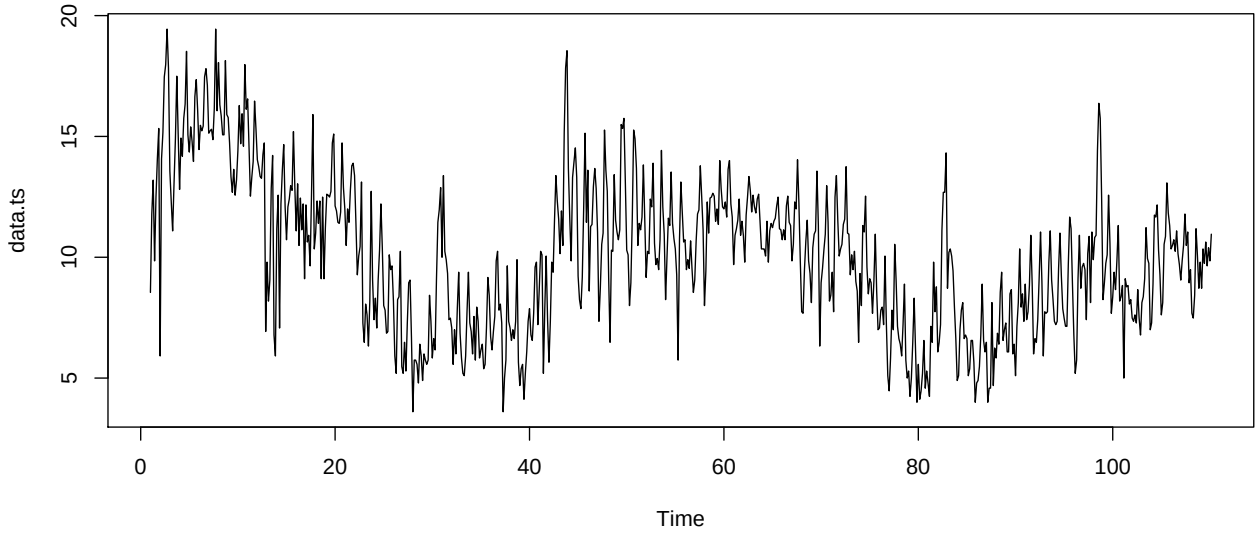







```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(data.ts)
## KPSS Level = 0.020778, Truncation lag parameter = 6, p-value = 0.1
```

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



The plot is not good but AIC and BIC are very high, we should try with a multi seasonal decomposition

```
## [1] 7
```

6.2 Searching for multi seasonalities

We searched for all the possible multi seasonalities using the msts package but at the end adding many seasonalities did not helped in finding better residuals or clearer trends and seasonalities. Adding more and more just created a complicated useless model that scored the same as the base one.

6.3 RUGARCH

We tried this model but it was not useful for great predictions, residuals were much more scattered than arima.

7 Conclusions

It was a hard work! Iwas really interesting! I tried a lot of methods to fit the data and obtain good forecasting and good residuals. I tried searching for multi seasonalities, difference some times to reach stationarity, detrending with lm and ma, smoothing. I tried by hand many arima models. I tried to decompose the time serie in many ways. I tried all the possible frequencies that can be thought as valid. Eventually it was really interesting! I experimented a lot.

The lesson I learned from this dataset is that the most foreign come after the mid of the week and the peak of foreign is unexpectedly on Friday and secondarily on Saturday. The number of outliers is very high due to programmed events that break the seasonality of the time serie. The number of foreign connected to the openwifi surely can be a good proxy for the overall number of foreign in milan since the antennas are in the central part of the city. In the days after the break down of the roaming policy the number of foreign surely increased but we cannot prove this fact since the data we have is not enough.

8 Bibliography

- [1] http://www.milanotoday.it/green/life/nuovi-hotspot-open-wifi-milano.html
- [2] https://www.mobileworld.it/2017/08/07/roaming-gratis-europa-condizioni-fair-use-114077/
- [3] https://www.cameramoda.it/it/milano-moda-donna/
- [4] https://www.milanoweekend.it/articoli/milano-fashion-week-2017-eventi-programma/
- [5] https://www.lastampa.it/milano/2017/06/17/news/milano-smart-city-del-futuro-se-ne-parla-all-ar 34584894?refresh_ce
- [6] https://www.wikieventi.it/milano/index.php?data_selezionata=2017-06-17
- [7] https://www.wikieventi.it/milano/index.php?data_selezionata=2017-07-22
- [8] k