Analysis of a time series

Introduction

The aim of this report is to analyze road traffic in Minnesota, USA, specifically on the section between Minneapolis and St. Paul, in relation to other factors such as weather, temperature, and holidays/weekends.

The research questions we aim to answer through time series analysis are the following:

- Is traffic affected by bad weather?
- Is traffic affected by temperature?
- Is traffic affected by holidays?

To address these questions, I first carried out an exploratory analysis of the time series, in which I examined normality, stationarity, and the presence of trends and seasonality. Subsequently, I built simple linear regression, ARIMA, SARIMA, and regARIMA models.

The dataset was taken from the **UCI Machine Learning Repository** and contains the following variables:

- `traffic_volume`: hourly traffic volume
- `clouds_all`: hourly percentage of cloud cover
- `temp`: average hourly temperature in Kelvin
- `holiday`: holiday for each day (e.g., *"Christmas Day"*; if no holiday, *"None"*)
- `rain_1h`: millimeters of rain in one hour
- `snow_1h`: millimeters of snow in one hour
- `date_time`: date and time
- `weather_main`: weather description (e.g., *"Clouds"*, *"Clear"*, ...)

Exploratory analysis of the dataset variables

I import the required libraries and load the dataset:

```
library(lubridate)
library(tsbox)
library(forecast)
library(tsibble)
library(fpp)
library(fpp2)
library(performance)
library(tidyverse)
```

```r
library(ggplot2)
library(ggpubr)
library(dplyr)
library(lubridate)
library(tseries)
library(feasts)
library(urca)
library(fable)
dataset <- read_csv("Metro_Interstate_Traffic_Volume.csv")
```

The variables of interest for the analysis are: **`traffic_volume`**, **`clouds_all`**, **`holiday`**, **`temp`**, and **`date`**.

First, I transformed *`holiday`* into a factor variable and *`date`* into a date variable.
Since the data are hourly, I then had to aggregate them to obtain daily values:

- I computed the **average** for *`clouds_all`*, *`traffic_volume`*, and *`temp`*
- I computed the **sum** of millimeters of precipitation for *`rain_1h`* and *`snow_1h`*
- For *`holiday`*, I kept the first value recorded each day
- For *`weather_main`*, I took the **mode**

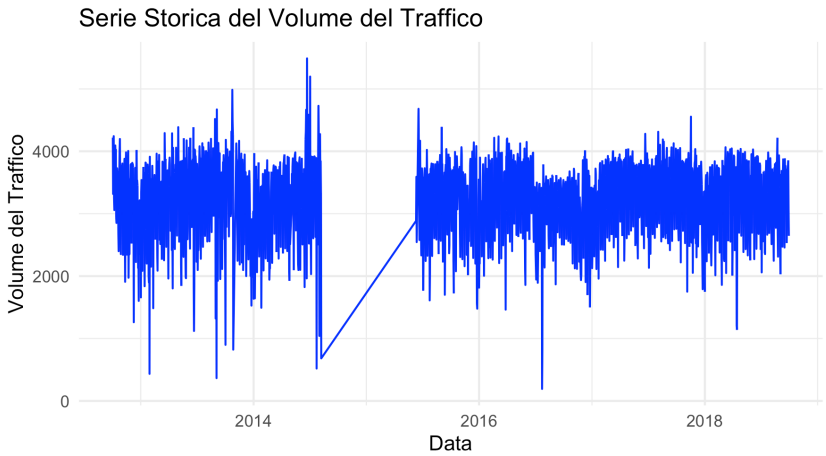
Finally, I created a variable called `is_holiday_day`, which takes value 1 if the given day is a holiday and 0 otherwise, and I removed the previous *`holiday`* variable.

```r
dataset <- dataset %>%
  mutate(date_time = as.Date(date_time))
# format "YYYY-MM-DD"


dataset$holiday <- factor(dataset$holiday)


clouds_mean<- dataset %>%
  group_by(date_time) %>%
  summarise(mean_clouds = mean(clouds_all, na.rm = TRUE))
dataset <- dataset %>%
  left_join(clouds_mean, by = "date_time")


traffic_mean<- dataset %>%
  group_by(date_time) %>%
  summarise(mean_traffic = mean(traffic_volume,na.rm=TRUE))
dataset <- dataset %>%
  left_join(traffic_mean, by = "date_time")


temp_mean<- dataset %>%
  group_by(date_time) %>%
  summarise(mean_temp = mean(temp,na.rm=TRUE))
dataset <- dataset %>%
  left_join(temp_mean, by = "date_time")
```

```r
daily_totals <- dataset %>%
  group_by(date_time) %>%
  summarise(
    total_rain = sum(rain_1h, na.rm = TRUE),
    total_snow = sum(snow_1h, na.rm = TRUE)
  )
dataset <- dataset %>%
  left_join(daily_totals, by = "date_time")


get_mode <- function(x) {
  ux <- unique(x)
  tab <- tabulate(match(x, ux))
  ux[which.max(tab)]  # Restituisce il primo valore in caso di pareggio
}

daily_weather <- dataset %>%
  group_by(date_time) %>%
  summarise(
    weather_main_mode = get_mode(weather_main)
  )
dataset <- dataset %>%
  left_join(daily_weather, by = "date_time")


daily_holiday <- dataset %>%
  group_by(date_time) %>%
  summarise(first_holiday = first(holiday))
dataset <- dataset %>%
  left_join(daily_holiday, by = "date_time")
dataset <- dataset %>%
  mutate(holiday_binary = if_else(holiday == "None", 0, 1))
```

As can be seen in the following plot, which shows the time series of the variable *mean_traffic*, there is a one-year gap in the data:

```r
ggplot(dataset, aes(x = date_time, y = mean_traffic, na.rm = TRUE)) +
  geom_line(color = "blue") +  # Usa una linea blu per il grafico
  labs(
    title = "Serie Storica del Volume del Traffico",
    x = "Data",
    y = "Volume del Traffico"
  ) +
  theme_minimal()
```



I therefore decided to consider only the years from late 2015 to late 2018.

```
dataset <- dataset %>%
  filter(date_time >= ymd("2015-06-11") & date_time <= ymd("2018-12-31"))
```

Exploratory analysis

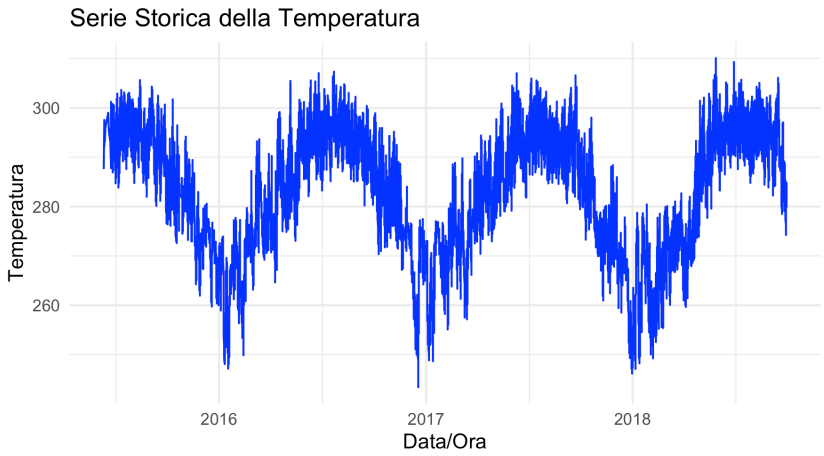
On the previously listed variables, I carried out an exploratory analysis including:

- Histogram and box plot to examine the distribution of the variable
- Handling of possible outliers
- Verification of normality using the Bera–Jarque test
- ACF and PACF plots to assess persistence and autocorrelation
- Ljung–Box and Box–Pierce tests to check whether the series is a realization of a white noise process
- Box–Cox transformation to evaluate whether a transformation is needed to make the distribution normal
- Augmented Dickey–Fuller test to verify the stationarity of the series

If the series is found to be non-stationary due to the presence of a trend, I performed detrending in the case of a deterministic trend and differencing in the case of a stochastic trend. If, instead, the series is seasonal, I deseasonalized it using harmonic regression or other techniques.

mean_temp variable

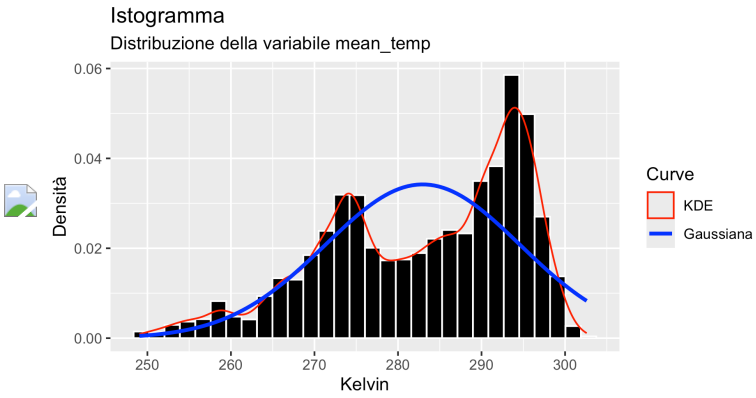
The first variable I analyze is the average temperature: it turns out to have an annual frequency.

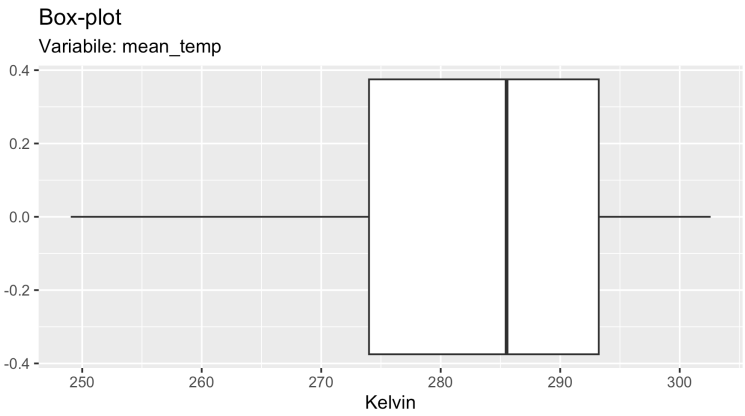


By observing the graph, we can expect the presence of seasonality in the series, as well as a growing linear trend.

Let's now look at the histogram and box plot to examine its distribution:

```
cc <- c("Dens"="#FF0000","Norm"="blue")
dataset %>%
  ggplot(data = ., aes(x = dataset$mean_temp)) +
  geom_histogram(aes(y =..density..),
                 colour="white",
                 fill = "black") +
  geom_density(aes(col="Dens")) +
  stat_function(fun = dnorm,
                args = list(mean = mean(dataset$mean_temp,na.rm=T),
                            sd = sd(dataset$mean_temp,na.rm = T)),
                aes(col="Norm"),
                size=1.1) +
  labs(title = "Istogramma ",
       subtitle = "Distribuzione della variabile mean_temp",
       x = "Kelvin",
       y = "Densità") +
  scale_color_manual("Curve",
                     values = cc,
                     breaks = c("Dens","Norm"),
                     labels = c("KDE","Gaussiana"))


dataset%>%
  ggplot(aes(x = dataset$mean_temp)) +
  geom_boxplot(outlier.colour="red",
               outlier.shape=8,
               outlier.size=4,
               notch=F) +
  labs(title = "Box-plot",
       subtitle = "Variabile: mean_temp",
       x = "Kelvin")
```





From the histogram, we can deduce that the variable does not have a normal distribution, as evident from the Gaussian curve and the kernel density not overlapping. There appears to be a strong skewness with a longer tail to the left, which is also confirmed by the box plot. This is unsurprising: during the year, the most frequent temperatures are between 0 and 20 degrees Celsius, with the median around 12–13°C (285K–273K).

The box plot also shows the absence of outliers.

To confirm what we have just observed from the two graphs, we perform the Bera-Jarque test, which evaluates the null hypothesis that the input data are normally distributed against the alternative hypothesis of non-normality.

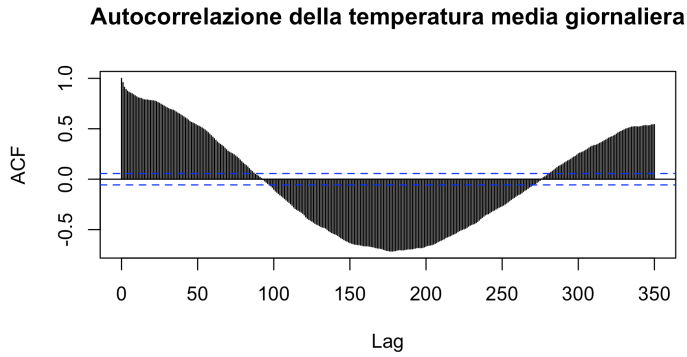
```
resultJB <- jarque.bera.test(dataset$mean_temp)
print(resultJB)
```

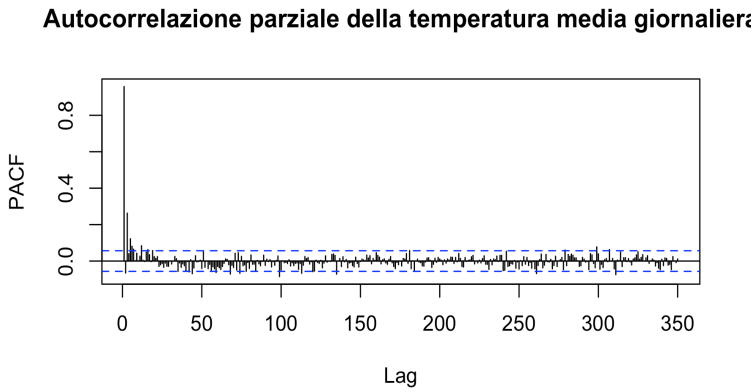
Obtaining a very low p-value, less than 2.2e-16, we reject the null hypothesis of normality and confirm what was previously assumed.

Now let us show the graphs related to the ACF and PACF:

```
dataset <- dataset %>%
  mutate(Date = ymd(date_time)) %>%
  group_by(Date) %>%
  summarise(mean_temp = mean(temp, na.rm = TRUE)) %>%
  ungroup() %>%
  as_tsibble(index = Date)


acf(dataset$mean_temp, main = "Autocorrelazione della temperatura
```

```
        media giornaliera",
        lag.max = 350, xlab = "Lag", ylab = "ACF")

pacf(dataset$mean_temp, main = "Autocorrelazione parziale della
     temperatura media giornaliera",
     lag.max = 350, xlab = "Lag", ylab = "PACF")
```





From the ACF plot, we notice the presence of seasonality as there is a strong correlation between the variable and itself at lag 1, lag 2, etc. This suggests that the series is seasonal with good persistence, also because the decay toward zero of the correlations appears to be quite slow. From the PACF plot, which measures the relationship between the variable at time t and itself at time t-k without the influence of intermediate lags, we observe that the most significant correlations occur at the initial lags: indeed, temperature is strongly correlated with that of previous days.

Furthermore, this series can be considered a non-ergodic process because the memory is long-term, meaning there is a strong dependence between observations that are far apart in time; in fact, the temperature of a given day can be influenced by factors occurring days, weeks, or even years earlier.

To confirm that the correlations are statistically significant, the Ljung-Box and Box-Pierce tests were also conducted:

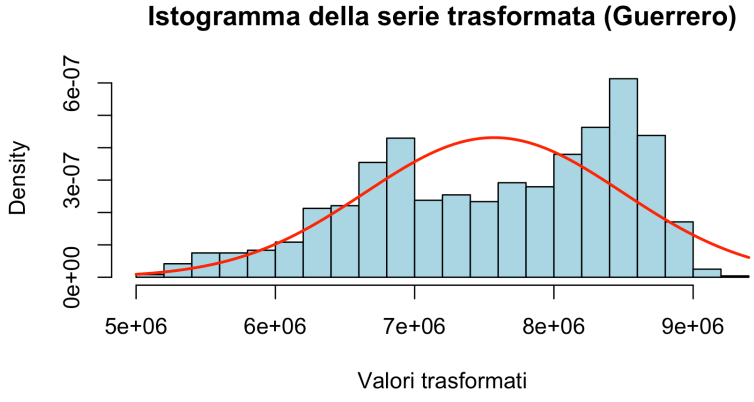
```
y1 <- ts(dataset$mean_temp, frequency = 365, start = c(2015, 162))

ljung_box(x = y1, lag = 1)
ljung_box(x = y1, lag = 10)
ljung_box(x = y1, lag = 20)
```

```r
ljung_box(x = y1, lag = 30)

box_pierce(x = y1, lag = 1)
box_pierce(x = y1, lag = 10)
box_pierce(x = y1, lag = 20)
box_pierce(x = y1, lag = 30)
```

In these tests, the alternative hypothesis states that at least one autocorrelation coefficient is significantly different from zero. Since in both cases the p-value is zero, we reject the hypothesis that the series is a realization of a White Noise process.

At this point, given that we have confirmed the non-normality of the variable *mean_temp*, we perform the Box-Cox transformation using the “log-likelihood” and “Guerrero” methods to check whether any transformation is needed to make the distribution normal:

```r
lambda_guer <- forecast::BoxCox.lambda(dataset$mean_temp,method = "guerrero",
                                       lower = -3,upper = 3)

lambda_loglik <- forecast::BoxCox.lambda(dataset$mean_temp,method = "loglik",
                                         lower = -3,upper = 3)

#In this case, both methods yield approximately the same result, around 3.
#For simplicity, we will consider only the Guerrero method.

dataset$mean_temp_bc_guer <- forecast::BoxCox(dataset$mean_temp,
                                              lambda = lambda_guer)


hist(dataset$mean_temp_bc_guer,
     main = "Istogramma della serie trasformata (Guerrero)",
     xlab = "Valori trasformati",
     col = "lightblue",
     border = "black",freq=F,breaks=20)

mu_guer <- mean(dataset$mean_temp_bc_guer, na.rm = TRUE)
sigma_guer <- sd(dataset$mean_temp_bc_guer, na.rm = TRUE)
curve(dnorm(x, mean = mu_guer, sd = sigma_guer),
      col = "red",
      lwd = 2,
      add = TRUE)
```



Since the transformation to the third power does not change the data distribution, it is not worth losing interpretability to perform a transformation; therefore, we proceed with the variable **mean_temp**.

We verify the stationarity of the series with the Augmented Dickey-Fuller (ADF) test, after having first removed the missing values. We start with the ADF test using `type="trend"`, i.e., assuming the presence of a deterministic linear trend:

```
dataset <- dataset %>%
  group_by(lubridate::year(Date)) %>%
  mutate(Year_mean = mean(mean_temp,na.rm=T)) %>%
  ungroup()

dataset <- dataset %>%
  filter(mean_temp > 0) %>%
  mutate(log_temp = ifelse(mean_temp > 0, log(mean_temp), NA))

log_temp <- dataset %>%
  select(log_temp) %>%
  ts_ts()

log_temp_ts <- ts(dataset$log_temp, start = c(min(year(dataset$Date))),
                  frequency = 365)



ADF_logtemp_const_trend <- urca::ur.df(y = log_temp_ts, type = "trend",
                                       selectlags = "AIC")
summary(ADF_logtemp_const_trend)
```

obtaining the following output:

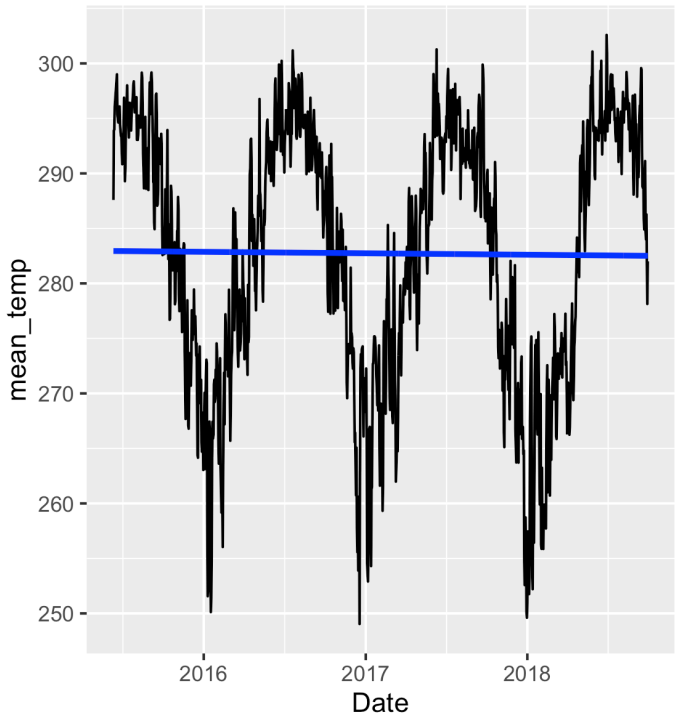
```
Residual standard error: 0.01232 on 1193 degrees of freedom
Multiple R-squared:  0.02619,   Adjusted R-squared:  0.02374
F-statistic: 10.69 on 3 and 1193 DF,  p-value: 6.139e-07


Value of test-statistic is: -5.4613 9.9527 14.9247

Critical values for test statistics:
      1pct  5pct 10pct
tau3 -3.96 -3.41 -3.12
phi2  6.09  4.68  4.03
phi3  8.27  6.25  5.34
```

- **tau3** tests the hypothesis that there is a unit root in the autoregressive part of the model. Since the critical values are lower than the test statistic, we reject H_0, meaning there is no unit root and the series is stationary.

- **phi2** is another test associated with the regression; from the obtained value we conclude that the trend is not significant.
- **phi3** tests the hypothesis that both the constant and the trend are null, but from the obtained values we conclude that at least one of them is significant.

  Since we reject H_0 in all three cases, the series is stationary around a deterministic linear trend. Therefore, it is not necessary to proceed with the other two types of ADF tests (type="drift" and type="None"). To make the series stationary around zero, we would need to detrend it (linearly), i.e., subtract the trend from the series.

```
mod_trend_lin <- dataset %>%
  model(m = TSLM(mean_temp ~ trend()))
report(mod_trend_lin)


# Serie detrendizzata
log_temp_detr_lin <- augment(mod_trend_lin) %>%
  select(log_temp_detr_lin = .resid, trend_lin = .fitted, mean_temp)
log_temp_detr_lin %>%
  ggplot(aes(x = Date)) +
  geom_line(aes(y=mean_temp)) +
  geom_line(aes(y=trend_lin), col="blue", size=1.1)
```



The blue line represents the linear trend, which is increasing but in an almost imperceptible way. Since the trend is not significant, we decide not to proceed with detrending and only perform deseasonalization.

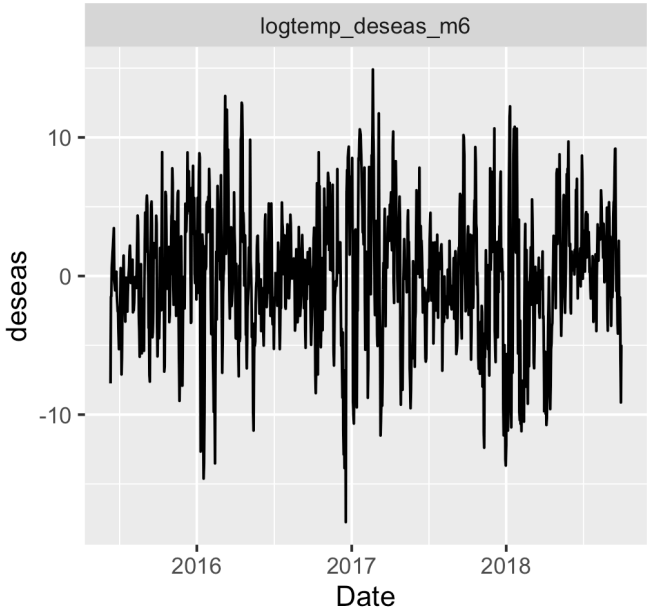
In this case, I deseasonalized the series using harmonic regression, employing the Fourier series to approximate sine and cosine with a polynomial of degree *k*. Specifically, we created six polynomials of degrees *k = 1, 2, 3, 4, 5, 6*.

```r
mean_temp_ts <- ts(dataset$mean_temp, start = c(2015, 1), frequency = 365)

K1 <- tslm(mean_temp_ts ~ trend + fourier(x = mean_temp_ts, K = 1))
K2 <- tslm(mean_temp_ts ~ trend + fourier(x = mean_temp_ts, K = 2))
K3 <- tslm(mean_temp_ts ~ trend + fourier(x = mean_temp_ts, K = 3))
K4 <- tslm(mean_temp_ts ~ trend + fourier(x = mean_temp_ts, K = 4))
K5 <- tslm(mean_temp_ts ~ trend + fourier(x = mean_temp_ts, K = 5))
K6 <- tslm(mean_temp_ts ~ trend + fourier(x = mean_temp_ts, K = 6))

logtemp_deseas <- dataset %>%
  mutate(logtemp_deseas_m1 = K1$residuals, seas_m1 = K1$fitted.values,
         logtemp_deseas_m2 = K2$residuals, seas_m2 = K2$fitted.values,
         logtemp_deseas_m3 = K3$residuals, seas_m3 = K3$fitted.values,
         logtemp_deseas_m4 = K4$residuals, seas_m4 = K4$fitted.values,
         logtemp_deseas_m5 = K5$residuals, seas_m5 = K5$fitted.values,
         logtemp_deseas_m6 = K6$residuals, seas_m6 = K6$fitted.values)


perf_m1 <- model_performance(model = K1)
perf_m2 <- model_performance(model = K2)
perf_m3 <- model_performance(model = K3)
perf_m4 <- model_performance(model = K4)
perf_m5 <- model_performance(model = K5)
perf_m6 <- model_performance(model = K6)
perf <- as.data.frame(rbind(perf_m1,perf_m2,perf_m3,perf_m4,perf_m5,perf_m6))
cbind(Model = c("M1","M2","M3","M4","M5","M6"),perf)
perf_cv_m1 <- CV(obj = K1)
perf_cv_m2 <- CV(obj = K2)
perf_cv_m3 <- CV(obj = K3)
perf_cv_m4 <- CV(obj = K4)
perf_cv_m5 <- CV(obj = K5)
perf_cv_m6 <- CV(obj = K6)
perf_cv <- as.data.frame(rbind(perf_cv_m1,perf_cv_m2,perf_cv_m3,perf_cv_m4,
                               perf_cv_m5,perf_cv_m6))
cbind(Model = c("M1","M2","M3","M4","M5","M6"),perf_cv)
```

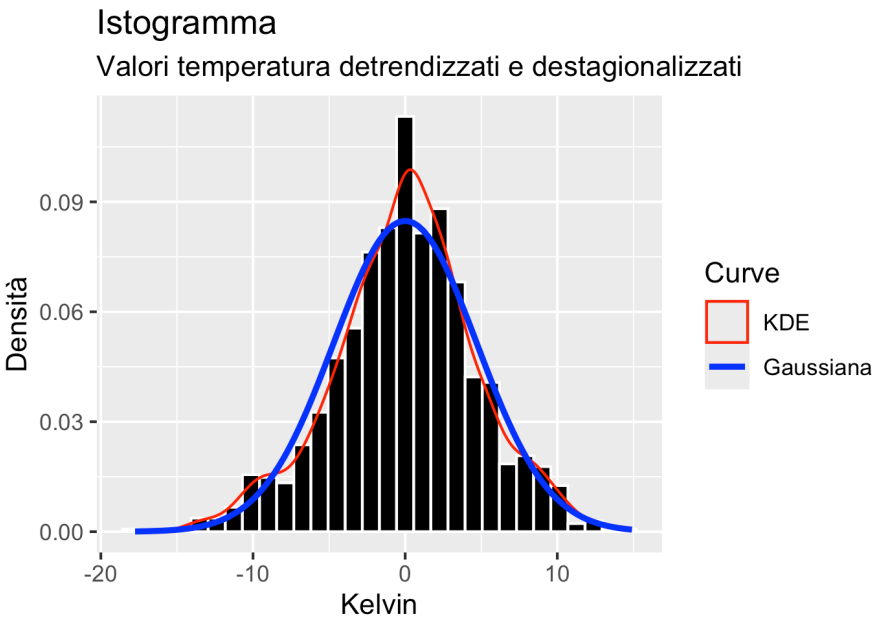
	Model	CV	AIC	AICc	BIC	AdjR2
perf_cv_m1	M1	24.28485	3826.610	3826.660	3852.056	0.8270272
perf_cv_m2	M2	23.43877	3783.838	3783.932	3819.463	0.8333659
perf_cv_m3	M3	23.09785	3766.022	3766.173	3811.825	0.8360957
perf_cv_m4	M4	22.68602	3744.229	3744.451	3800.211	0.8393141
perf_cv_m5	M5	22.66414	3742.808	3743.115	3808.968	0.8397694
perf_cv_m6	M6	22.65322	3741.947	3742.353	3818.286	0.8401484

Using the metrics CV, AIC, AICc, BIC, and adjusted R^2, I selected the polynomial of degree 6 as the one that best approximates the seasonal pattern of the series, achieving a very good fit to the data of 84%.

Therefore, the deseasonalized series is:

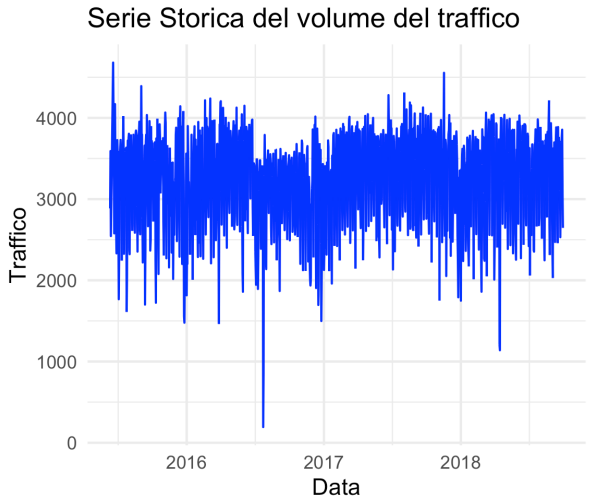


The deseasonalized series appears to be stationary with a mean of zero, as confirmed by the ADF test with type="None" performed on the series, which led to the rejection of H_0. It also has a normal distribution, as shown in the following histogram:



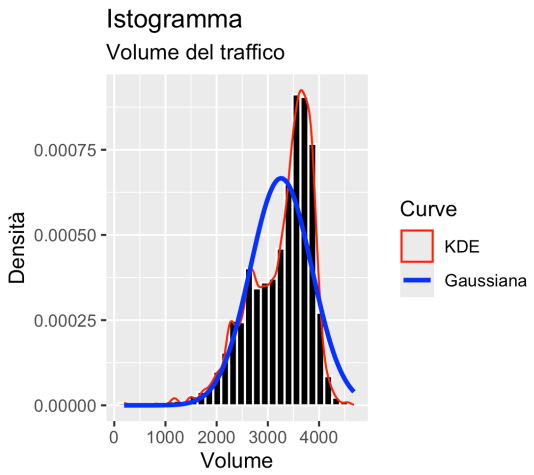
mean_traffic variable

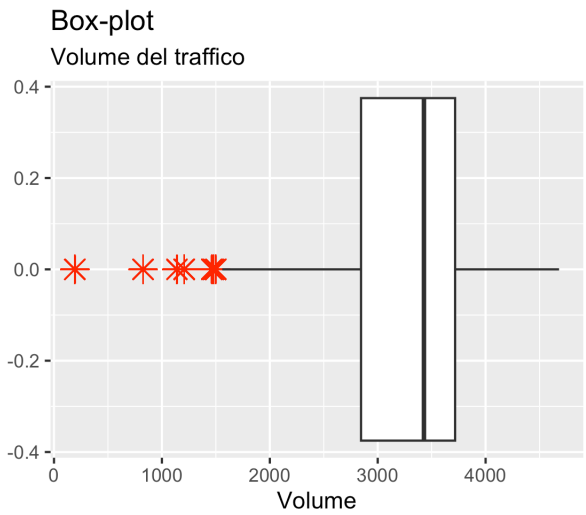
Omitting the code used, which is similar to what was shown earlier, the second variable to analyze is the traffic volume, shown in the following graph:



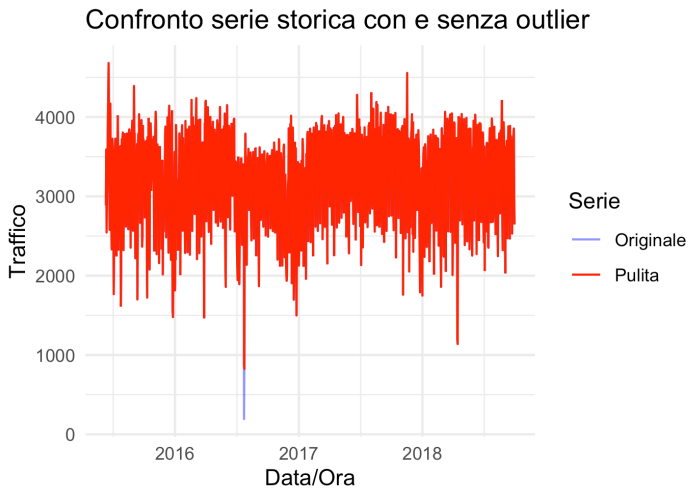
From this, a weekly seasonality emerges and no trend is observed. We also note the presence of a very low peak around mid-2016.

Now let's examine the distribution of traffic_volume using the histogram and box plot:





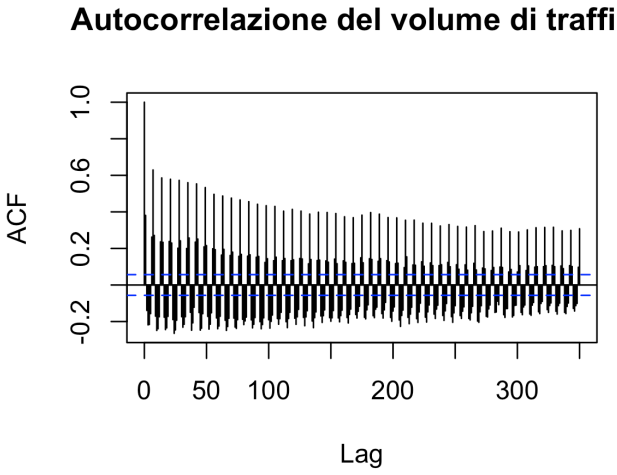
From the histogram, we notice that the traffic_volume series does not follow a normal distribution, as indicated by the kernel density, but instead shows strong skewness with a long left tail. This is also confirmed by the box plot, which reveals the presence of several lower outliers. To address this issue, we decide to use the `tsclean()` function in R, obtaining the following cleaned series:

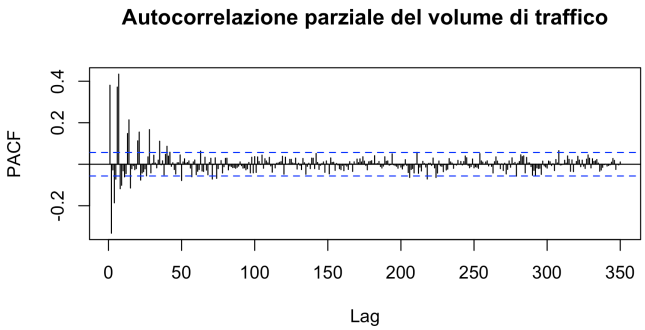


From which we note the very low value between 2016 and 2017 already highlighted earlier. A deeper analysis of the dataset shows that this is explained by some missing values, which could create problems with deseasonalization.

What is observed in the graphs is confirmed by the Bera-Jarque test, and the null hypothesis of normality for the time series distribution is therefore rejected.

We now investigate the persistence of the series through the ACF and PACF plots.





From the ACF plot, we note a high persistence of the series, as the correlations tend to zero very slowly. Moreover, we observe an interesting aspect due to the alternation of negatively and positively correlated lags: specifically, the first three lags are negatively correlated; the next four are

positively correlated; the following three are negatively correlated, and so on — confirming the presence of a weekly seasonality.

To confirm the significance of these correlations, the Ljung-Box and Box-Pierce tests were also performed. Given that the p-values in both cases are equal to 0.00, we reject the null hypothesis of normality for the series.

Similarly to what was done for the variable *mean_temp*, the transformations obtained using the 'log-likelihood' and 'Guerrero' methods result in distributions that change only slightly compared to the original. Therefore, it was decided to proceed with the original series.

From the values obtained with the ADF test:

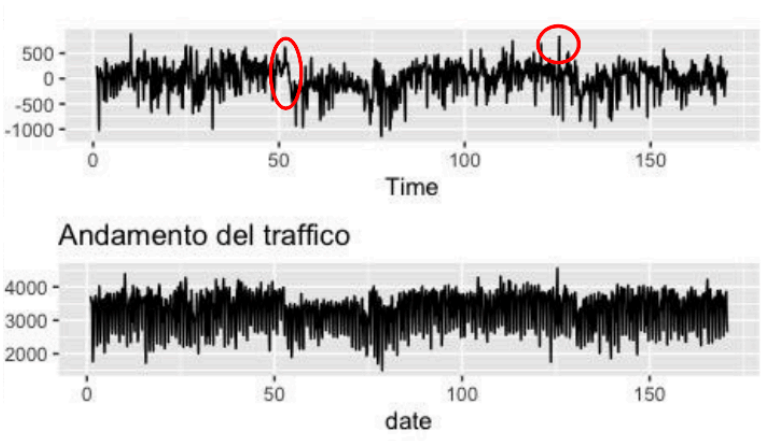
```
Residual standard error: 457.5 on 1183 degrees of freedom
Multiple R-squared:  0.4254,    Adjusted R-squared:  0.424
F-statistic:   292 on 3 and 1183 DF,  p-value: < 2.2e-16


Value of test-statistic is: -29.4565 289.2313 433.8451

Critical values for test statistics:
      1pct  5pct 10pct
tau3 -3.96 -3.41 -3.12
phi2  6.09  4.68  4.03
phi3  8.27  6.25  5.34
```

The series turns out to be stationary around a deterministic trend. To make the series stationary around zero, we need to perform a detrending, in this case accompanied by deseasonalization. I created three polynomials K1, K2, and K3 of degrees 1, 2, and 3 respectively, in which the variable *traffic* is expressed as a function of the trend and the harmonic regression carried out via Fourier. As we did previously, we choose the best model by comparing the following metrics:

	Model	CV	AIC	AICc	BIC	AdjR2
perf_cv_m1	M1	742.7931	7841.887	7841.921	7862.201	0.04712032
perf_cv_m2	M2	740.9542	7838.875	7838.947	7869.345	0.05113329
perf_cv_m3	M3	739.3720	7836.256	7836.379	7876.883	0.05481350

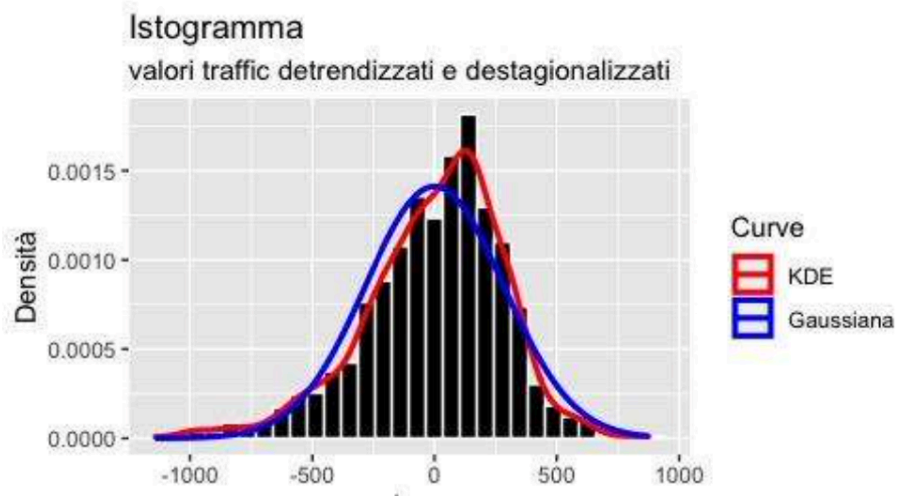
We choose polynomial M3, which has the lowest values for all metrics except BIC and the highest adjusted R^2, although in general the fit is very low, equal to 0.05%. Thus, the polynomial that best models both the seasonality and the trend of the series is of third degree.

Therefore, the deseasonalized and detrended series is:



Both from the graph of the detrended and deseasonalized series and from the original one, we notice the presence of a high traffic value on November 17, 2017, probably due to an important

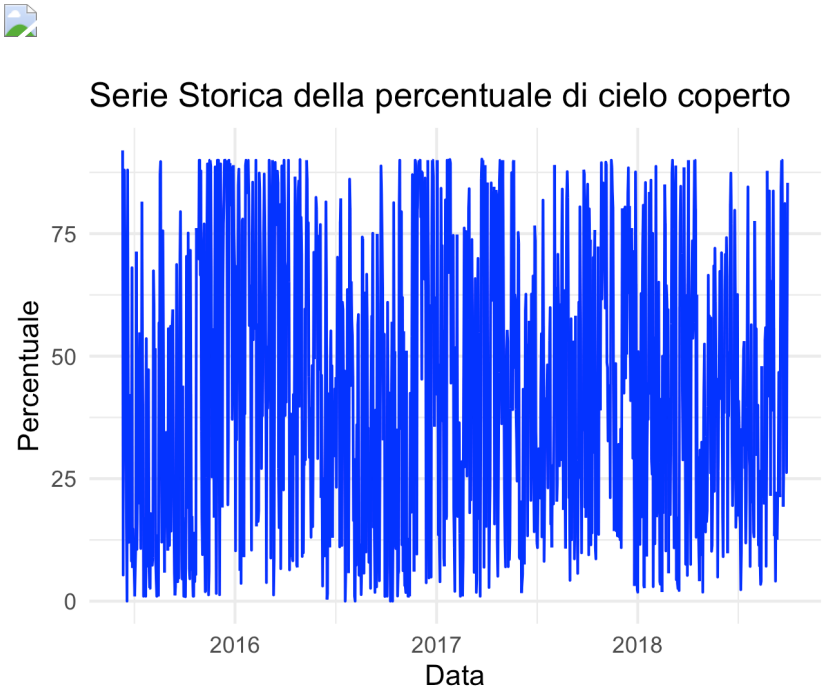
hockey game held in St. Paul. Furthermore, there is a particular pattern highlighted by the first circle corresponding to the days June 24, 25, and 26, 2016 — a Friday, Saturday, and Sunday, respectively. Generally, traffic on Saturdays and Sundays is lower compared to weekdays, but in this case the negative peak is more evident because on June 25 and 26 there was a storm that caused people to travel less.

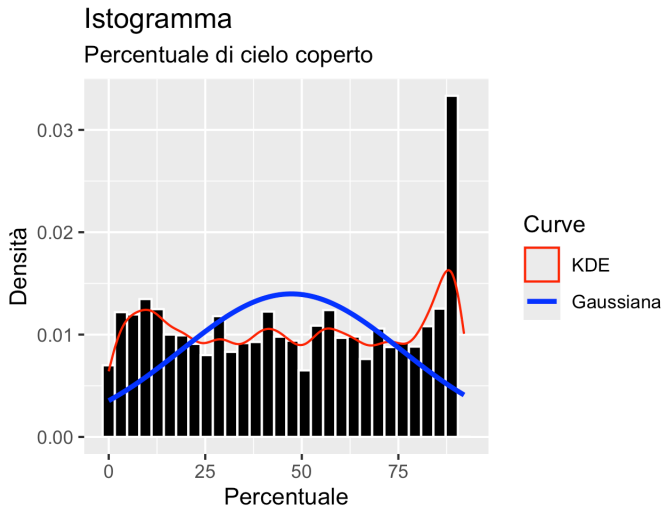


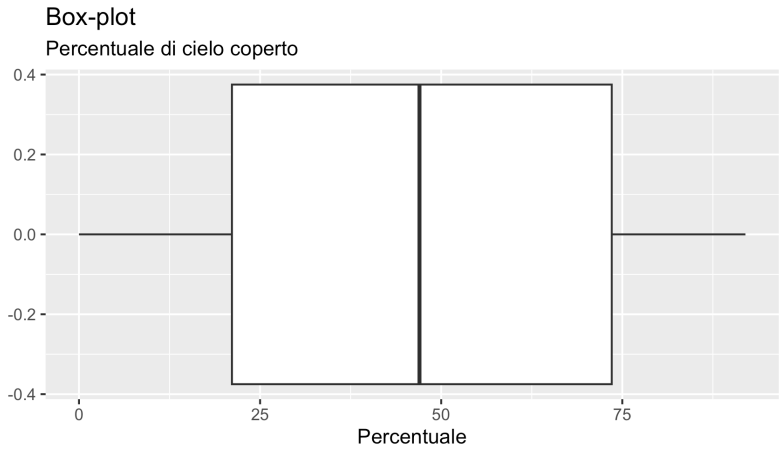
As we can see from the graph, the series has an almost normal distribution, and the ADF test with type="none" confirms the stationarity of the series.

clouds_all variable

The third series on which we perform the exploratory analysis is the percentage of cloud cover, which has an annual frequency.



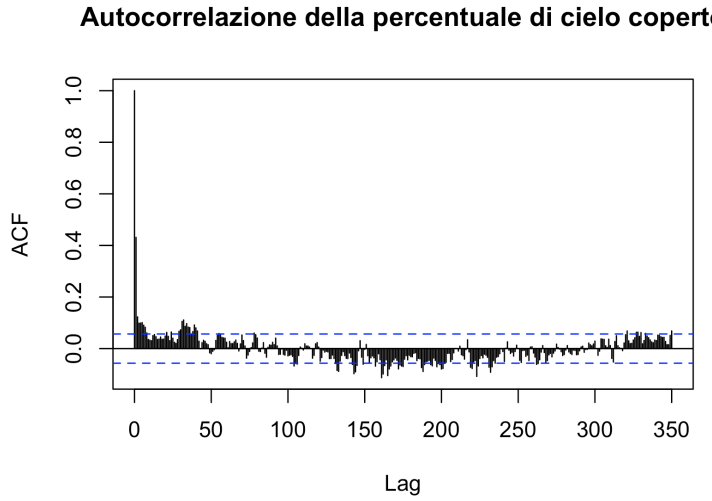




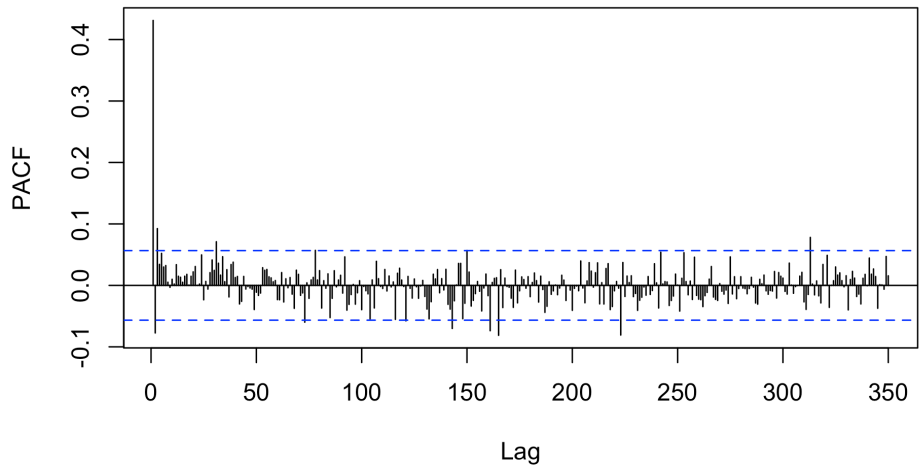
From the histogram plot, we note that the variable describing the percentage of cloud cover **does not have a normal distribution**, as indicated by the kernel density curve, but instead appears to be **uniform**.

This is also confirmed by the box plot, where we can observe that the variable is **symmetric**, with the median around **40% cloud cover**. There do not appear to be any outliers.

The output of the Bera-Jarque test confirms these observations.



Autocorrelazione parziale della percentuale di cielo coperto

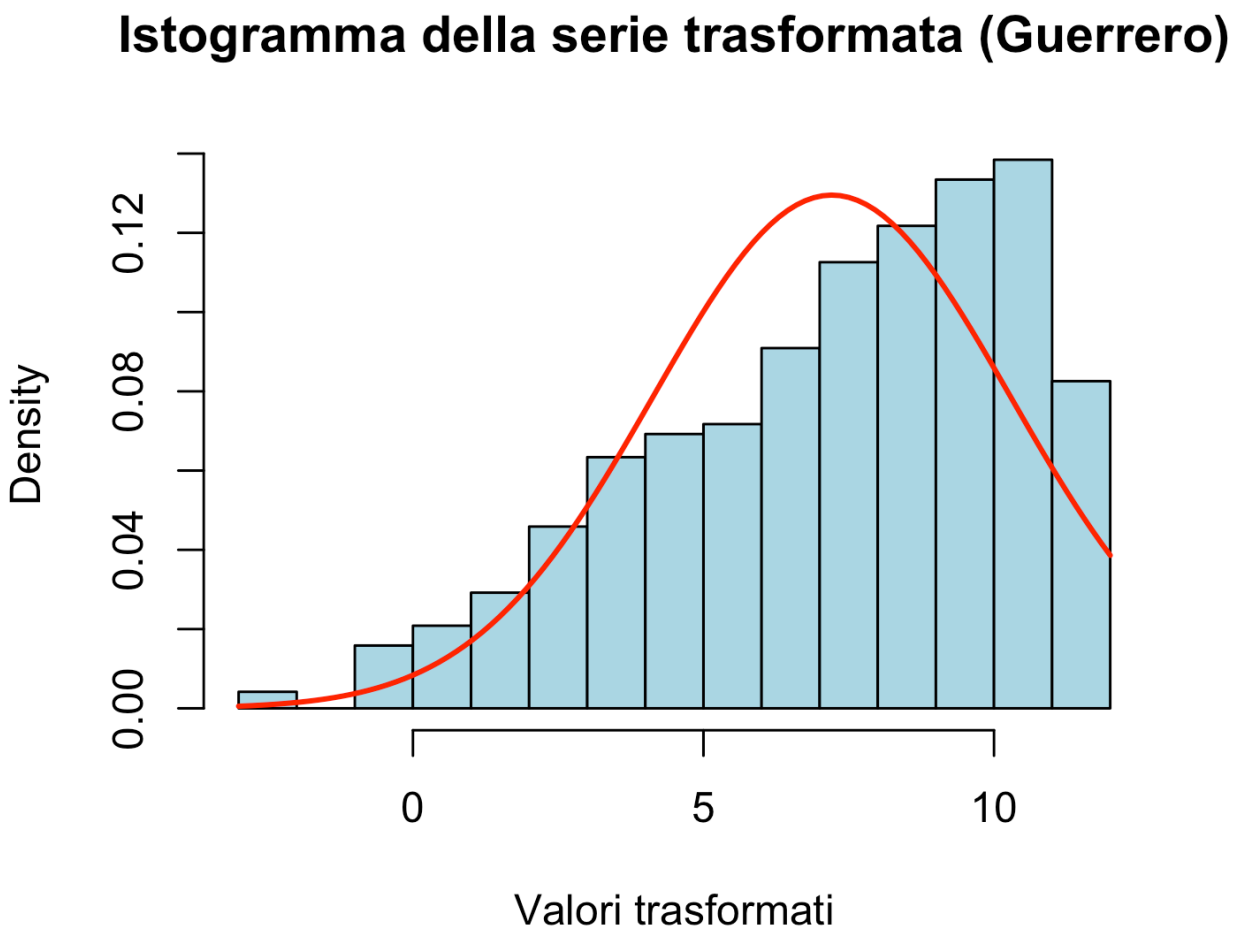


From the ACF plot, we can observe that the series is **not very persistent**, as the autocorrelations tend to zero very quickly.

Now let's look at the PACF plot, from which we see that the partial correlations are all not significant, except for the first lag, which has a negative correlation. However, this should not be a particular issue given the very small value of **-0.1**.

We then perform the Ljung-Box and Box-Pierce tests, obtaining a very low p-value, which leads us to reject the null hypothesis that the series is a realization of a White Noise process. Therefore, there is some persistence in the series, although limited.

Similarly to what was done for the other variables, Box-Cox transformations using the 'log-likelihood' and 'Guerrero' methods suggest unsuitable transformations that would cause the variable to lose its symmetry. For example, consider the result obtained with the 'Guerrero' method.



where the variable becomes strongly skewed with a longer left tail.

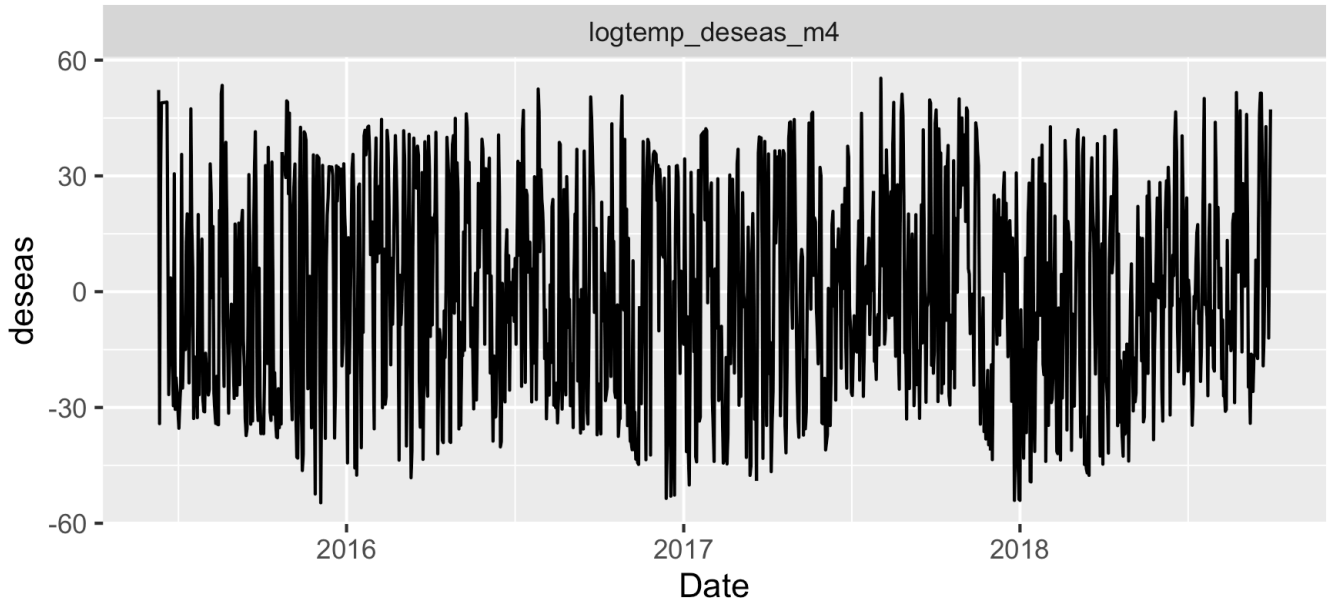
Also in this case, from the output of the ADF test, the series turns out to be **stationary around a deterministic trend**.

```
Residual standard error: 24.88 on 1180 degrees of freedom
Multiple R-squared:  0.2837,    Adjusted R-squared:  0.2819
F-statistic: 155.8 on 3 and 1180 DF,  p-value: < 2.2e-16


Value of test-statistic is: -19.9109 132.151 198.2217

Critical values for test statistics:
      1pct  5pct 10pct
tau3 -3.96 -3.41 -3.12
phi2  6.09  4.68  4.03
phi3  8.27  6.25  5.34
```

To make the series stationary around zero, we would need to perform detrending. However, since the trend of the series is not clearly visible in the plot, we decide to apply only **deseasonalization**, again using harmonic regression with Fourier series.

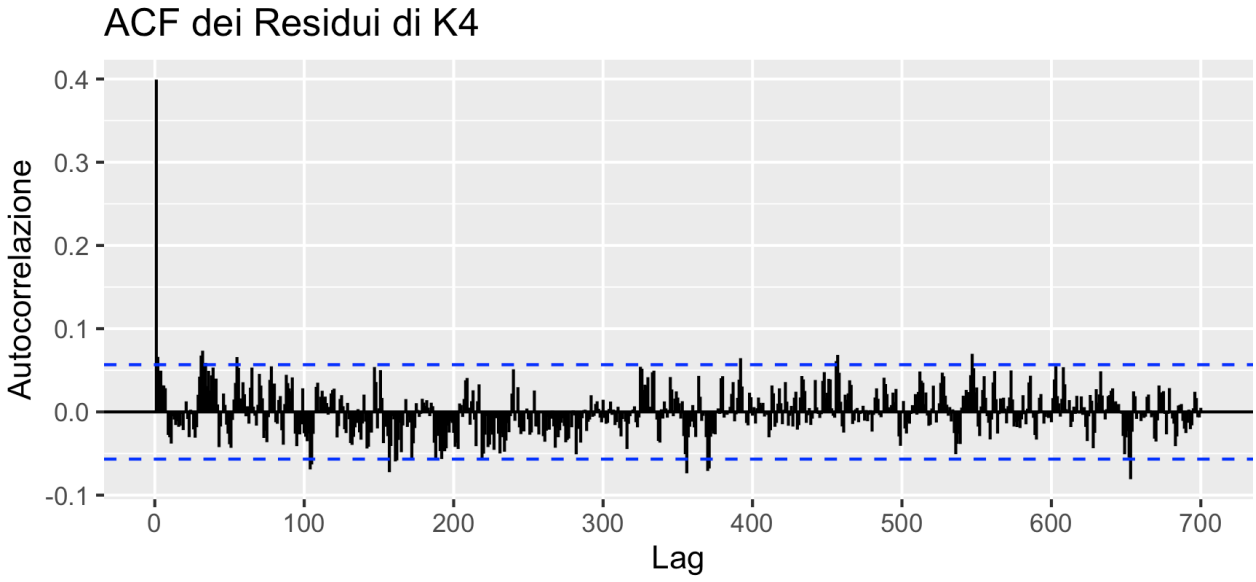
	Model	CV	AIC	AICc	BIC	AdjR2
perf_cv_m1	M1	746.2625	7900.380	7900.430	7925.805	0.04267828
perf_cv_m2	M2	744.7515	7897.876	7897.971	7933.472	0.04627540
perf_cv_m3	M3	744.0435	7896.645	7896.797	7942.410	0.04884363
perf_cv_m4	M4	743.1616	7895.140	7895.363	7951.076	0.05161949
perf_cv_m5	M5	745.6968	7899.097	7899.405	7965.203	0.05004919
perf_cv_m6	M6	746.5955	7900.416	7900.824	7976.692	0.05057310

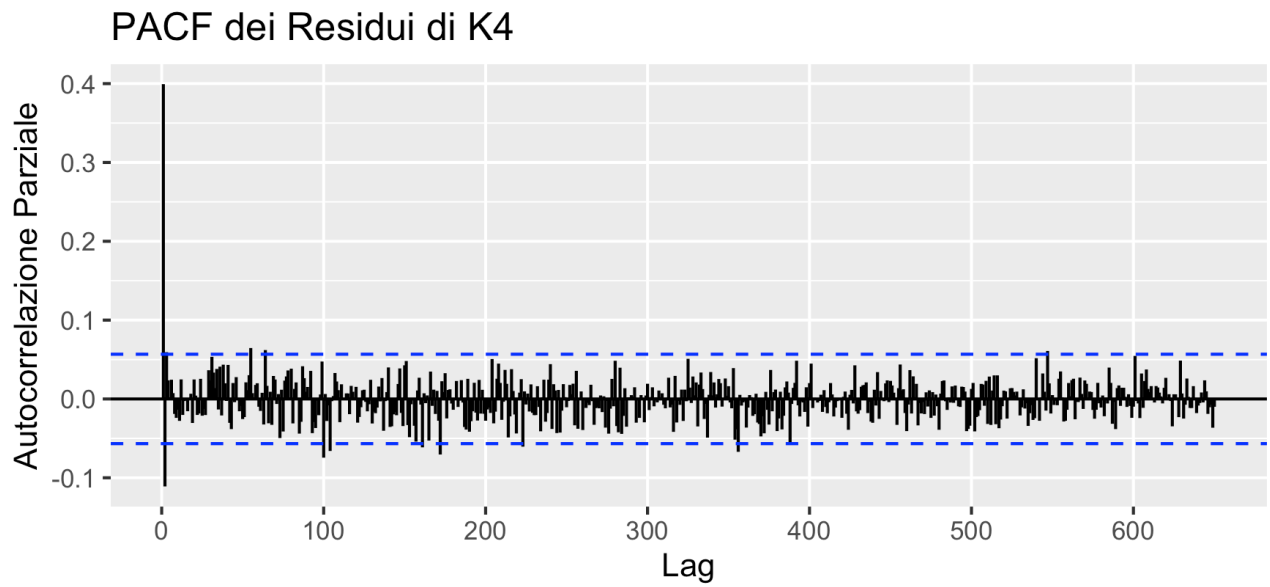
The chosen polynomial is of degree 4, since it has the lowest CV, AIC, and AICc, and the highest R^2_adj, although we can note that the overall fit is very low, at just 0.05%.



The behavior of the series seems to be more harmonic, indicating the continued presence of seasonality.

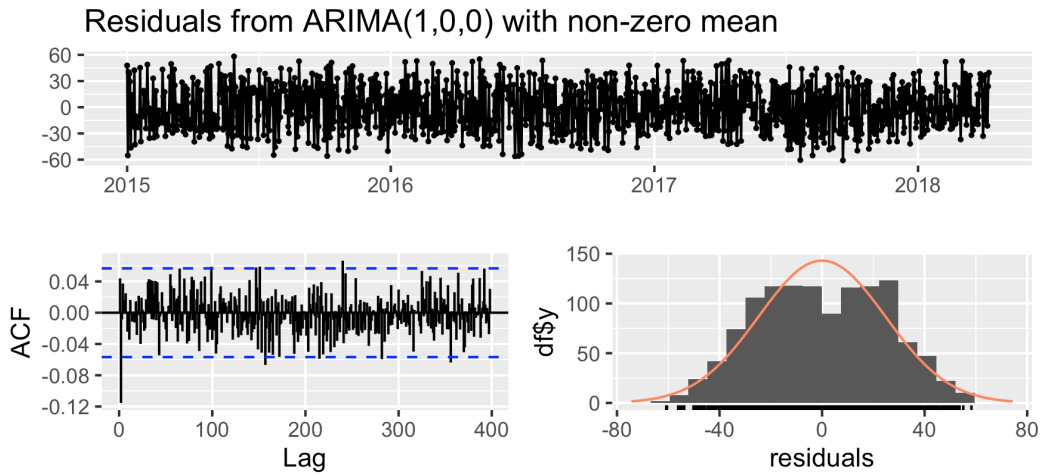
We therefore plot the residuals of model 4 to check whether seasonality is indeed present, obtaining:





Therefore, seasonality seems to be minimal, and the only lag with significant correlation is the first one. We can try to fit an AR(1) model:

```
mod_ar1 <- Arima(logtemp_deseas$logtemp_deseas_m4, order = c(1,0,0))

checkresiduals(mod_ar1)
```



From these graphs, we observe that all autocorrelations appear to be insignificant. Therefore, we can conclude that the clouds_all series actually did not have seasonality, which was instead captured by the autoregressive component.

Moreover, the series now has a distribution that more closely resembles a normal distribution compared to the original one, which was uniform, although the Bera-Jarque test shows a p-value of 9.096e-09, leading us to reject the null hypothesis of normality. Additionally, the process is stationary, as confirmed by the output of the ADF test with type="None".

Decompositions

We now proceed with the decomposition of the series with the goal of isolating the **trend**, **seasonality**, and **residual**components. This decomposition will be performed only on the traffic and

temperature series, as they both exhibit trend and seasonality.

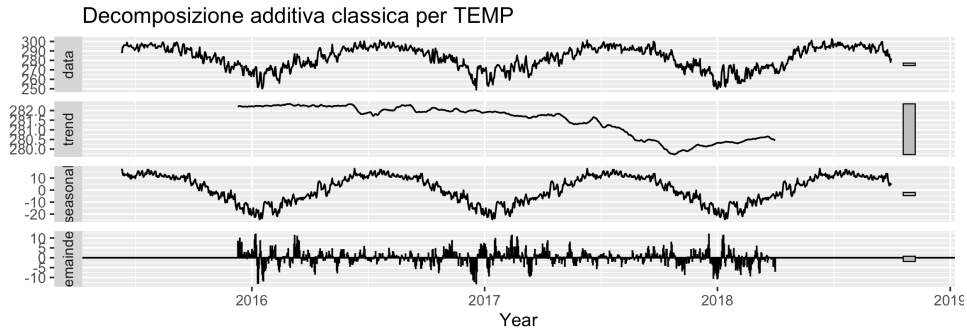
For the cloud cover percentage series, decomposition is not necessary. From our exploratory analysis, there is no evident trend — only a barely perceptible one — and although seasonality seems to be present visually and logically, it is in fact absent because it is covered by the autoregressive component.

mean_temp variable

In this case, we apply a classical additive decomposition because the strength of the seasonal fluctuations around the trend remains constant and does not change with the level of the series. Thus, the historical series **mean_temp** will be expressed as the sum of all its components:

mean_temp=T_t+S_t + ε_t

```
source('FN - TS_custom_aggregate.txt')
source('FN - Sequence of dates.txt')

dataset <- dataset %>%
  mutate(Date = ymd(date_time)) %>%
  group_by(Date) %>%
  summarise(mean_temp = mean(temp, na.rm = TRUE)) %>%
  ungroup() %>%
  as_tsibble(index = Date)

dataset_TS <- ts_ts(dataset)

TEMP <- dataset_TS

library(zoo)
TEMP <- na.locf(dataset_TS)
autoplot(TEMP) +
  labs(y  = latex2exp::TeX("Kelvin"),
       title = "serie originale")

dec_add <- TEMP %>%
  decompose(type="additive")
dec_add

TEMP_add_destag <- dec_add$trend + dec_add$random
TEMP_add_detrend <- dec_add$seasonal + dec_add$random

autoplot(dec_add) +
  xlab("Year") +
  ggtitle("Decomposizione additiva classica per TEMP")
```



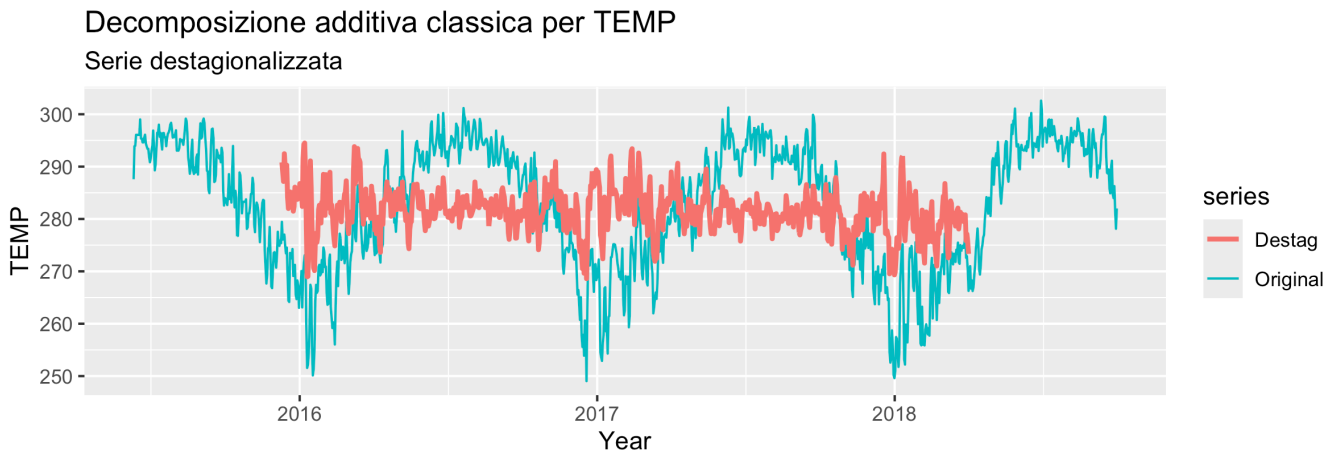
From the graph, we observe a very strong seasonality, as already seen in the exploratory analysis. We also notice the presence of a negative trend, which contradicts the earlier analysis; however, looking at the scale of values, ranging from 280.5 to 282, this trend appears to be negligible. This contradiction can be explained by the fact that, while in the EDA the trend was linear, here the trend component T is calculated using the moving average.

Regarding the residuals, the graph shows that they appear to fluctuate randomly around zero and seem to behave like white noise, with a mean equal to 0 and constant variance.

Regarding the detrended series, i.e., mean_temp=St+εt, it shows the same pattern as the original series because, as mentioned, the observed trend is minimal and therefore not significant.

As for the deseasonalization, we can better observe how the series looks once the seasonal component has been extracted, i.e., mean_temp=Tt+εt, through:

```
autoplot(TEMP,series = "Original") +
  autolayer(TEMP_add_destag,series = "Destag", size=1.01) +
  labs(x="Year",
       title = "Decomposizione additiva classica per CO",
       subtitle = "Serie destagionalizzata")
```



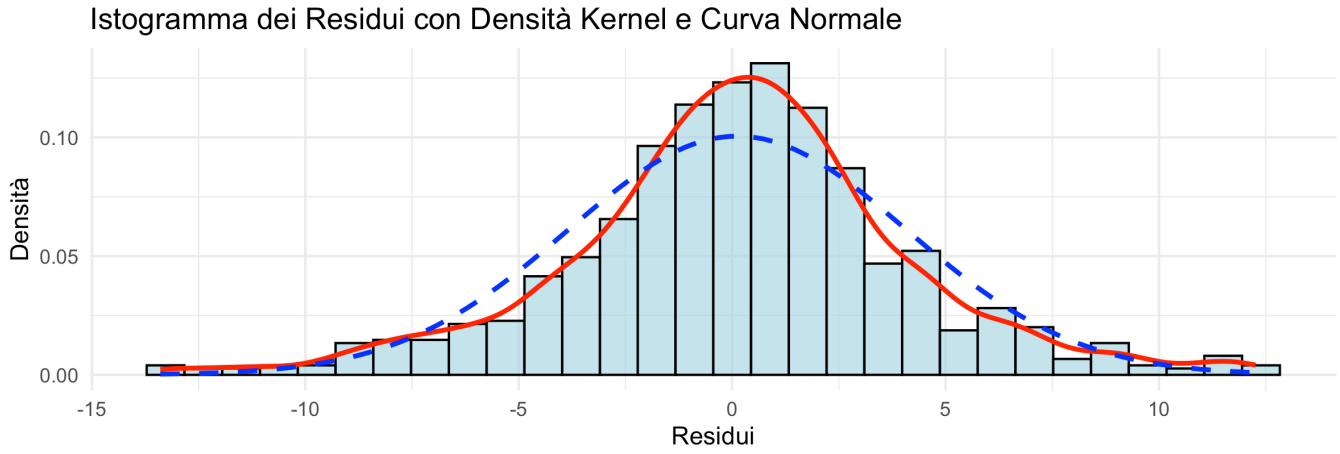
Deseasonalization performed through decomposition resolves the seasonality issues.

Instead, analyzing the distribution of the residual component, given by:

ε_t= mean_temp - T_t - S_t

```
residui_add <- dec_add$random
```

```r
residui_add <- na.omit(residui_add)

mu <- mean(residui_add)
sigma <- sd(residui_add)

ggplot(data = data.frame(Residui = residui_add), aes(x = Residui)) +
  # Istogramma normalizzato per la densità
  geom_histogram(aes(y = ..density..), bins = 30, fill = "lightblue",
                 color = "black", alpha = 0.7) +
  geom_density(color = "red", size = 1) +
  stat_function(fun = dnorm, args = list(mean = mu, sd = sigma),
                color = "blue", linetype = "dashed", size = 1) +
  ggtitle("Istogramma dei Residui con Densità Kernel e Curva Normale") +
  xlab("Residui") +
  ylab("Densità") +
  theme_minimal()
```



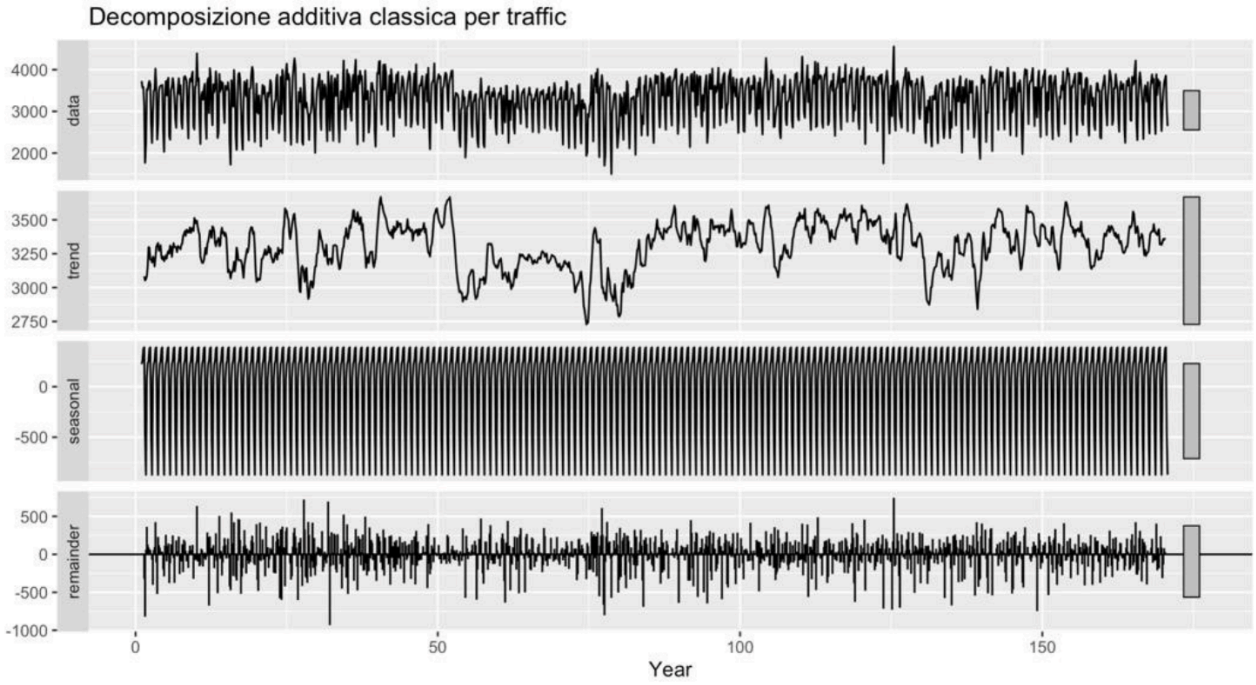
We can confirm the normality of the distribution and conclude that the residuals are white noise with mean 0 and constant variance.

mean_traffic variable

In this case as well, we perform a classical additive decomposition. Therefore, the historical series *mean_traffic* will be the sum of all its components:

mean_traffic=Tt+St+εt

Similarly to before, we represent both the historical series and its components:

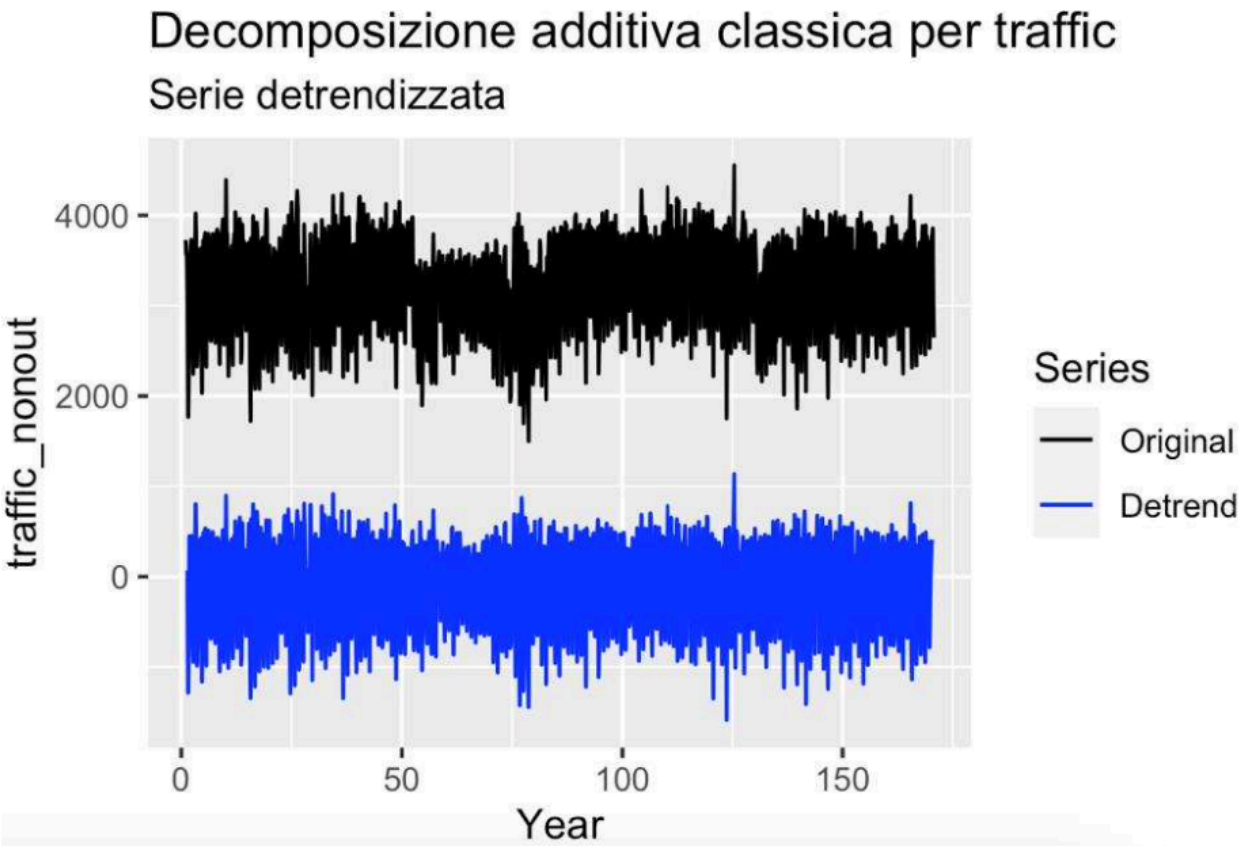


From the graph, we notice a slightly increasing trend, although minimal, and a weekly seasonality with lower peaks during weekends, as expected. Regarding the residuals, we observe that they fluctuate randomly around 0 and thus seem to be distributed like White Noise with mean 0 and constant variance, but we will confirm this later by testing the normality of their distribution.

We now look at the detrended series graphically, given by:

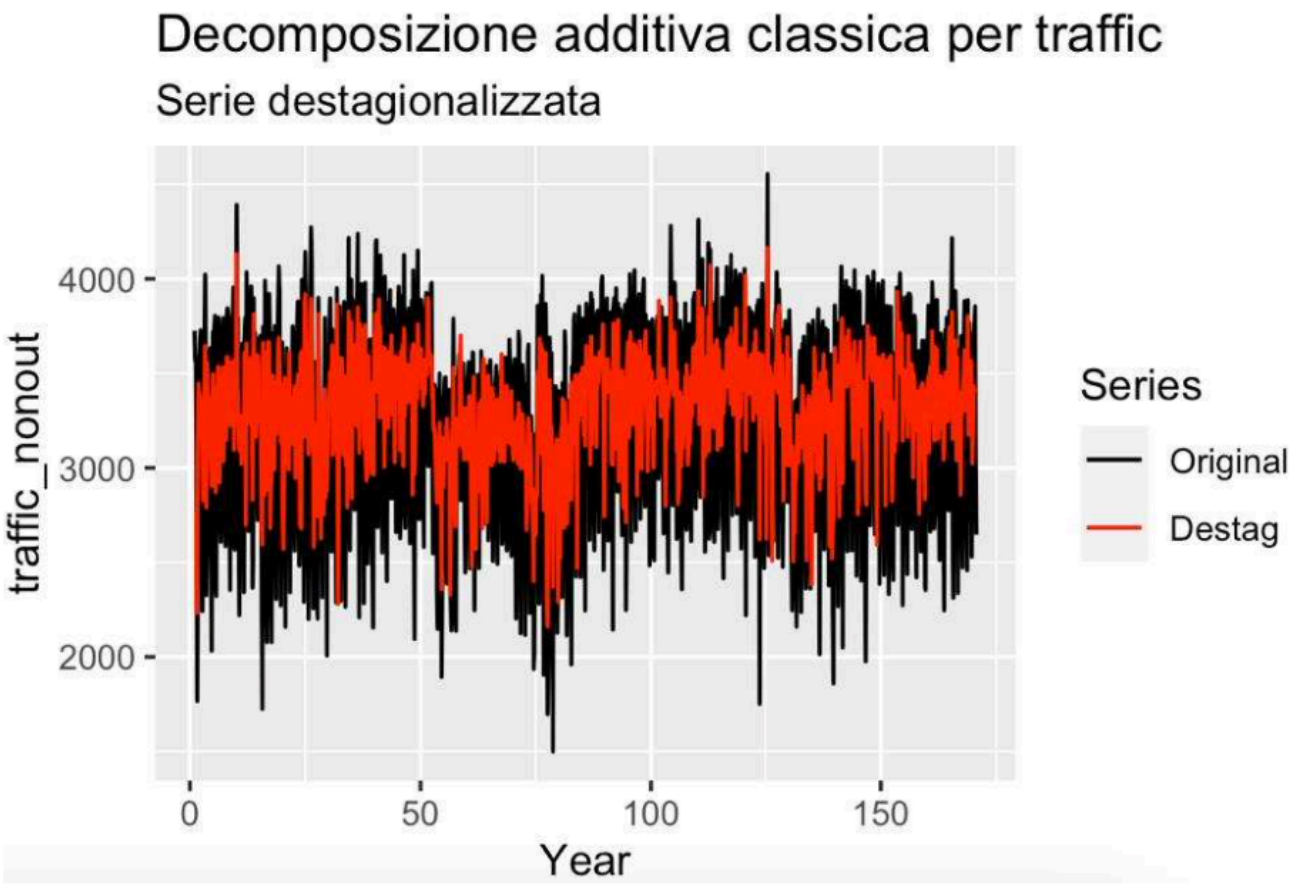
mean_traffic=St+εt

where the trend component Tt has been isolated.



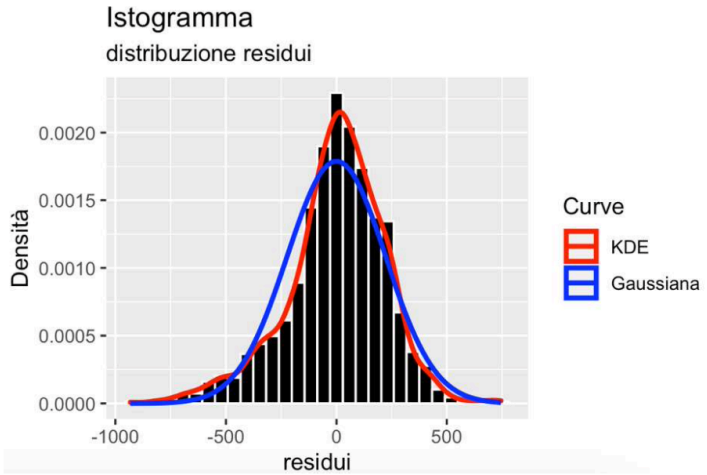
From the graph, we can see that the decomposition has worked well in this case. The series has indeed been centered around 0 and appears to have a much more stationary behavior compared to the original one.

Now, on the detrended series, we perform the deseasonalization. We isolate the seasonal component St, which is calculated by averaging the values belonging to the same period, thus obtaining: mean_traffic=$T_t + \varepsilon_t$:



Looking at the graph, we notice that the deseasonalized series appears more centered and less variable compared to the original series.

As for the residual component, given by εt=mean_traffic−Tt−St, the distribution of the residuals is practically normal, and thus we can confirm that they behave like White Noise with mean 0 and constant variance:

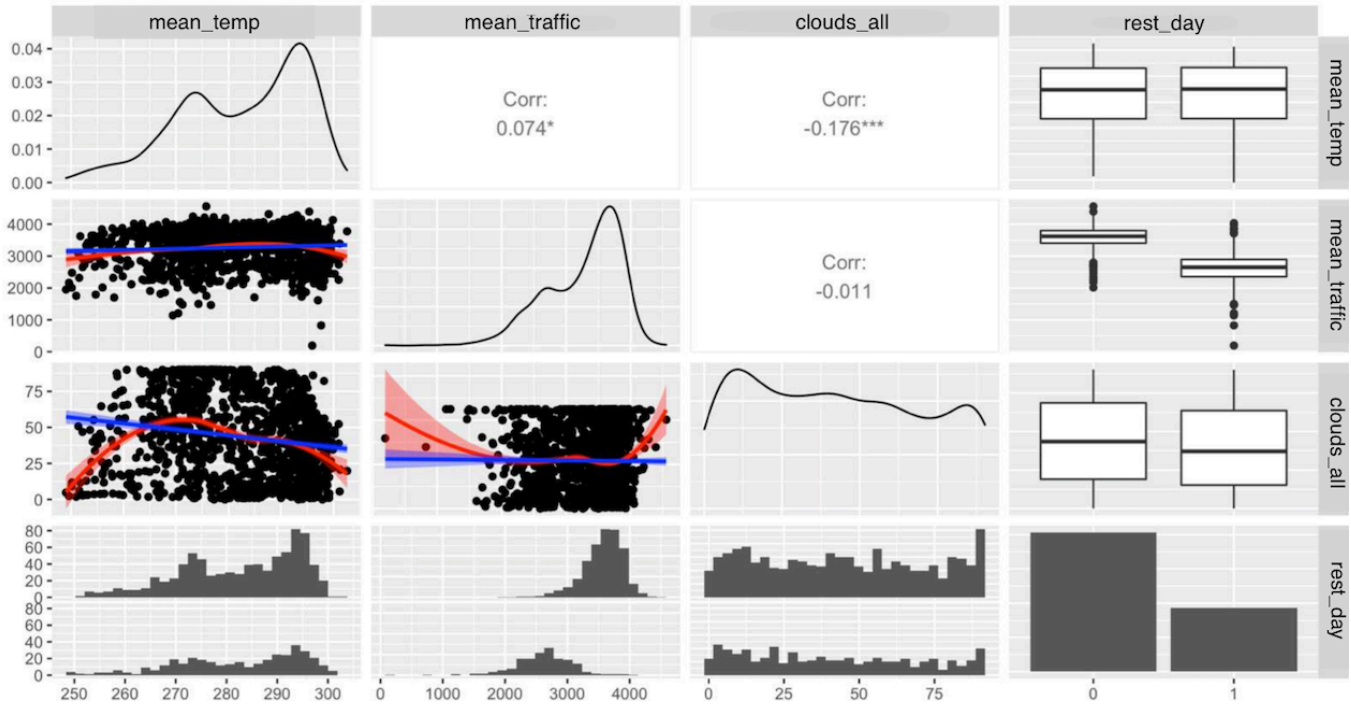


Regression analysis and residual analysis

Before performing the regression, I created a variable called `rest_day` to add to the dataset, which takes the value 0 when the day is a working day (Monday to Friday) and the value 1 when it is a weekend and/or a public holiday.

```
dataset <- dataset %>%
  mutate(
    Date = ymd(date_time),
    weekday = wday(Date, week_start = 1),
    # 1 = Lunedì, ..., 7 = Domenica
    rest_day = ifelse(weekday >= 6 | holiday_binary == 1, 1, 0)
    # Weekend o festivo = 1
  ) %>%
  select(-weekday)

rest_day <- dataset %>%
  group_by(date_time)
```

obtaining scatterplots and correlations between the variables of interest.



From the graph, we can immediately see that the only statistically significant correlation is between `mean_temp` and `clouds_all`, although it is low at -0.176: as temperatures rise, indicating the arrival of warmer seasons, the percentage of cloud cover decreases. The low correlation value is due to the fact that summer days can still be cloudy, even though, on average, they are fewer compared to autumn or winter.

The other correlations are very low, with values close to zero.

Regarding `rest_day`, since this is a factor variable, histograms and box plots are shown, from which we can observe that traffic is higher on working days and decreases on weekends and public holidays.

The blue lines in the graph represent the linear regression, while the red lines are LOESS curves. In the case of `mean_temp` and `mean_traffic`, the lines overlap very well, as expected.

After analyzing the correlations, we can proceed with the regression models. We identify three models where, for simplicity, we define T = `mean_traffic`; C = `clouds_all`; Te = `mean_temp`.

$$\begin{cases} T = \alpha_0 + \alpha_1 C + \alpha_2 Te + \alpha_3 R + \varepsilon_t \\ T = \beta_0 + \beta_1 Te + \varepsilon_t \\ T = \gamma_0 + \gamma_1 Te + \gamma_2 R + \varepsilon_t \end{cases}$$

Compare the following metrics:

	Model	CV	AIC	AICc	BIC	AdjR2
perf_cv_m1	M1	135089.4	14012.11	14012.16	14037.50	0.598961538
perf_cv_m2	M2	334519.4	15088.22	15088.24	15103.46	0.004659193
perf_cv_m3	M3	135788.4	14018.44	14018.47	14038.75	0.596477076

We note that the best model is the full model, with the lowest values of AIC, AICc, BIC, and CV, and a fit to the data equal to 59%.

Let's examine the parameters of this model through the summary.

```
Residuals:
     Min       1Q   Median       3Q      Max
-2437.50  -196.88    30.13   233.73  1429.37

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 2747.5317   262.6295  10.462  < 2e-16 ***
perc_cloud    -1.1231     0.3892  -2.886 0.003973 **
rest_day1   -963.8666    22.9977 -41.911  < 2e-16 ***
temp_K         3.1027     0.9154   3.390 0.000723 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 366.8 on 1182 degrees of freedom
Multiple R-squared:    0.6,     Adjusted R-squared:  0.599
F-statistic: 590.9 on 3 and 1182 DF,  p-value: < 2.2e-16
```

We observe that the residuals have a minimum value of -2437 and a maximum of 1429, and that the median relative to the residual standard error (30/366.8 = 0.1) is close to 0. We also notice a slight asymmetry, as the minimum is larger in magnitude than the maximum.

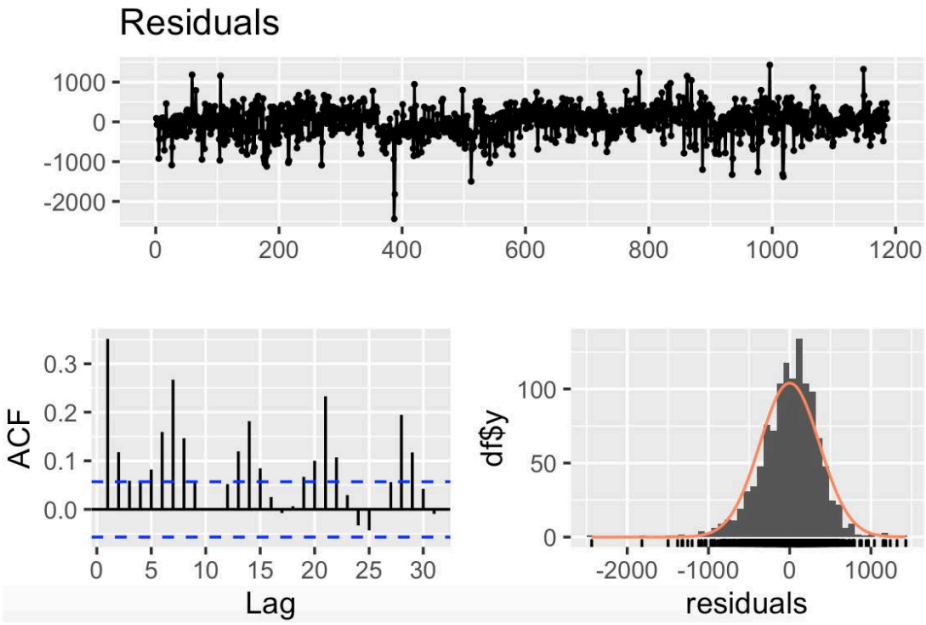
The intercept is approximately 2747, meaning that the average traffic volume when all other variables are zero is 2747. Of course, this does not make interpretative sense, since it is not possible to have a temperature of 0 Kelvin.

The variable `perc_cloud` has a value of about -1.12, which means that for each 1% increase in cloud cover, traffic decreases by 1.12. This confirms the negative correlation between the two variables and shows that traffic decreases on cloudy days.

When the variable `rest_day` takes the value 1, i.e., on non-working days, traffic decreases on average by about 983. This confirms what was already observed in the correlation plot, namely that traffic decreases on non-working days.

The variable `mean_temp` has a value of 3.10, meaning that for each 1-degree increase, traffic increases on average by 3.10. Of course, an increase of one Kelvin degree corresponds to an increase of one Celsius degree, since the transformation is simply a translation of 273.15.

We can see the distribution of residuals in the following graph:



From the graph, we observe that the residuals appear to fluctuate randomly around 0 and therefore resemble white noise (WN) with mean 0 and constant variance. The distribution, as seen from the histogram, is practically normal, although slightly asymmetric as mentioned earlier. However, from the ACF plot, we notice that the residuals are autocorrelated, meaning the white noise assumption does not hold. Therefore, we conclude that the residuals are normally distributed with mean zero and constant variance, but they are not white noise.

ARIMA models and Box-Jenkins analysis

In addition to the regression model, we decide to also estimate ARIMA, SARIMA, and regARIMA models to determine which best describes the behavior of the time series of interest.

We use the Box-Jenkins analysis to estimate these models. The procedure is divided into three steps:

- Identification of the optimal model through a graphical analysis of the series, data transformation if necessary, and analysis of stationarity and seasonality;
- Estimation of parameters using maximum likelihood;
- Model fitting to evaluate how well the model adapts to the data.

The codes used in this last part employ functions provided separately, which were studied during the Time Series course I attended at the University of Milano-Bicocca.

ARIMA

We apply the Box-Jenkins method to the residuals of the seasonally adjusted and detrended traffic series obtained in the exploratory analysis.

```
ARIMA(2,0,1) with zero mean

Coefficients:
          ar1      ar2      ma1
      1.2116  -0.2320  -0.9124
s.e.  0.0413   0.0352   0.0271

sigma^2 = 65484:  log likelihood = -8278.58
AIC=16565.16   AICc=16565.2   BIC=16585.49
```

From the output, we observe that the ARIMA model that best describes the behavior of our series is an ARIMA(2,0,1), meaning an autoregressive order of p = 2, a differencing order of d = 0 (thus stationary), and a moving average order of q = 1. Therefore, the process is influenced by its past up to lag t-2 and by exogenous shocks up to lag t-1.

We represent the ARIMA(2,1) model in expanded form:

$$Y_t = \mu + \phi_1 * Y_{t-1} + \phi_2 * Y_{t-2} + \epsilon_t + \theta_1 * \epsilon_{t-1}$$

The parameters all appear to be statistically significant. In particular, the parameters of the autoregressive part are $\phi 1=1.2116$ and $\phi 2=-0.2320$, while the parameter of the moving average part is $\theta 1=-0.9124$. Since the absolute value of $\theta 1$ is less than 1, the process is invertible.

SARIMA

We also estimate a SARIMA model, given that our series exhibits seasonal behavior. We again apply the Box-Jenkins method, but this time to the original traffic series (not seasonally adjusted or detrended), obtaining:

```
Series: traffic_nonout
ARIMA(1,0,0)(2,0,0)[7] with non-zero mean

Coefficients:
          ar1    sar1    sar2      mean
      0.2867  0.4723  0.3805  3309.206
s.e.  0.0278  0.0269  0.0270    78.438

sigma^2 = 88963:  log likelihood = -8464.21
AIC=16938.41   AICc=16938.47   BIC=16963.82
```

The optimal SARIMA that best captures the behavior of our series is obtained by multiplying an ARIMA(1,0,0) with an ARIMA(2,0,0) model that captures the seasonality of the series.

The resulting SARIMA thus represents a time series forecasting model that includes an autoregressive term of order 1 and no seasonal moving average term, with a seasonality of 7. We represent it in compact form:

$(1 - \psi_1 * L^7 - \psi_2 * L^{14}) * \phi_1(L) * Y_t = \epsilon_t$

The coefficients of the model are $\psi_1 = 0.4723, \psi_2 = 0.3805, \phi_1 = 0.2867$ all statistically significant.

regARIMA

The final model estimated using Box-Jenkins is the RegARIMA, with the original series 'traffic' as the response variable, and the original series 'temp', 'clouds_all', and 'rest_day' as the regressors.

```
Series: metro_col$traffic
Regression with ARIMA(1,1,2) errors

Coefficients:
         ar1      ma1      ma2   xreg1    xreg2      xreg3
      0.1375  -0.7938  -0.1633  2.3301  -1.0706  -925.5439
s.e.  0.0935   0.0926   0.0861  1.9832   0.3982    22.7812

sigma^2 = 115795:  log likelihood = -8587.8
AIC=17189.6   AICc=17189.7   BIC=17225.15
```

What we obtain is a multiple linear regression model with errors that are distributed as ARIMA(1,1,2), i.e., with an autoregressive part of order 1, with a non-stationary distribution because d=1, and with a moving average part of order 2.

We can write the regression with ARIMA errors as a linear regression with ARMA errors by applying the d-th difference (d=1) to every variable in the following way:

$$T^* = \alpha_1 * C^* + \alpha_2 * Te^* + \alpha_3 * R^* + \frac{\theta_2(L)}{\theta_1(L)} * \epsilon_t$$

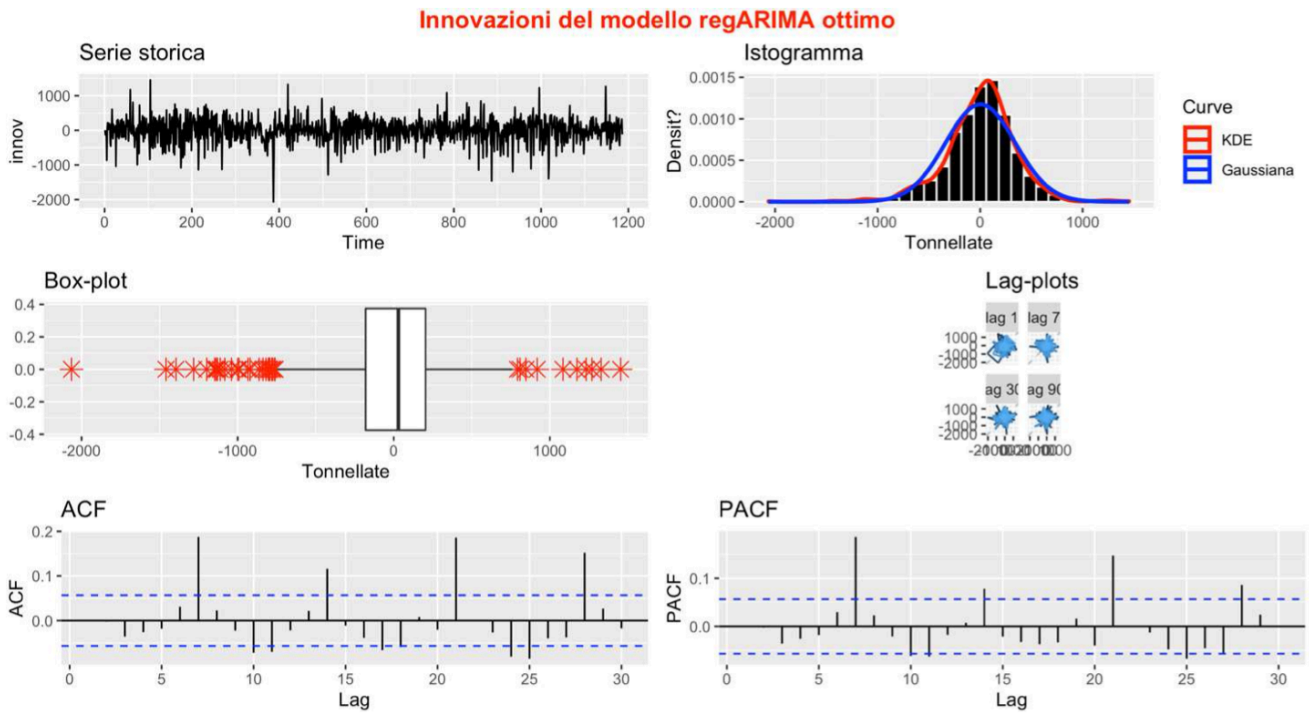
With $\alpha_1 = 2.3301, \alpha_2 = -1.0706, \alpha_3 = -925.5439$ statistically significant and very similar to those obtained in the full regression model but not equal, as they were estimated with MLE, and the AR and MA coefficients have values of, respectively $\psi_1 = 0.137, \theta_1 = -0.7938, \theta_2 = -0.1633$ all statistically significant.

Once all the models have been estimated, we determine which is the best model among ARIMA, SARIMA, RegARIMA, and the full regression model, using the optimization criteria:

Modello	AIC	AICc	BIC	R^2
regr_completa	14012.11	14012.16	14037.50	0.598
ARIMA	16565.16	16565.2	16585.49	0.637
SARIMA	16938.41	16938.47	16963.82	0.572
regARIMA	17189.6	17189.7	17225.15	0.658

The model that maximizes all the metrics and has a better fit to the data is the RegARIMA.

Now we proceed with the analysis of the innovations to assess the model's adaptability to the data:



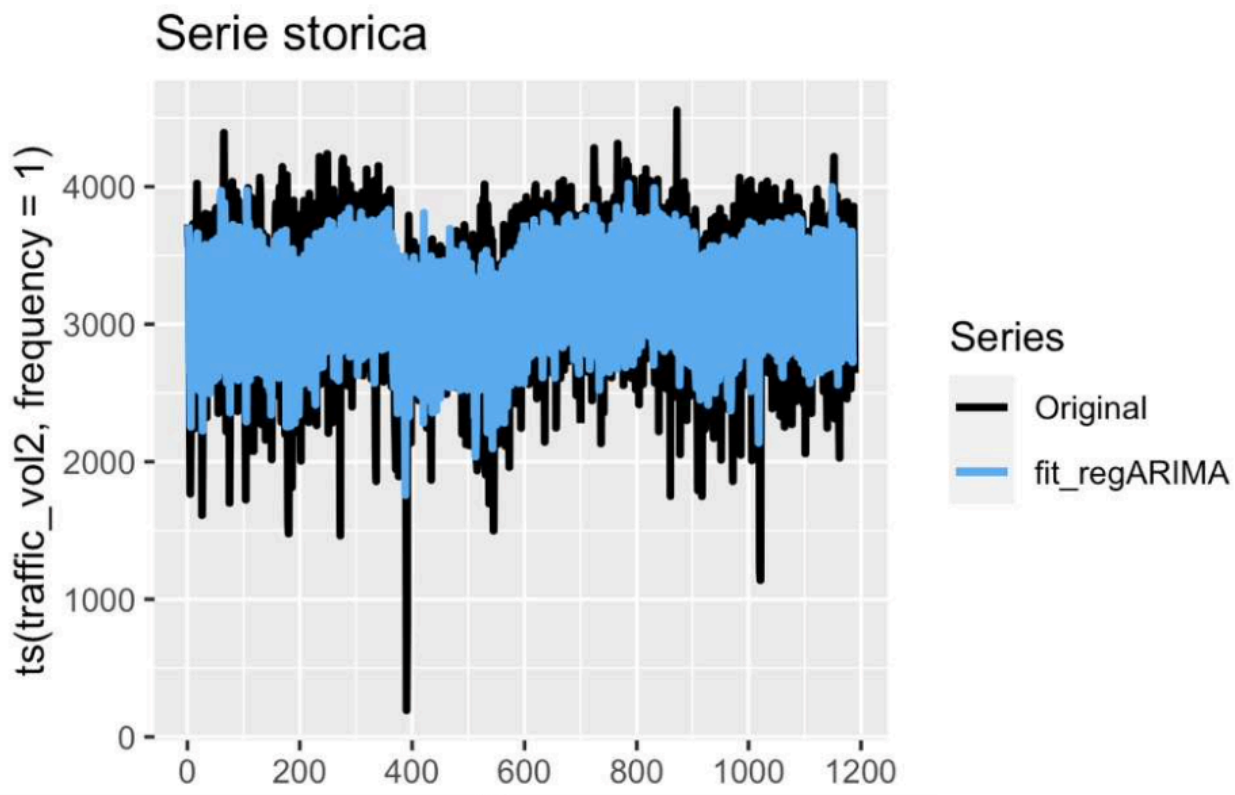
Several plots are reported, including the residual trend, the histogram, the Box-plot, the lag plot, and the autocorrelation plots. From the first plot, we note that the residuals show a seemingly random trend around zero, leading us to believe they might be distributed as WN with a mean of 0 and

constant variance. From the histogram, we observe that the residual distribution is normal, symmetrical around the mean of 0, which is also confirmed in the box-plot.

From the ACF and PACF plots, we observe that the residuals are uncorrelated with each other, and we can also note this in the lag plots, which represent the joint values of the lagged variables in pairs.

Therefore, we can conclude that the residuals are white noise.

Now we calculate the fitted values of the model and compare them with the original series:



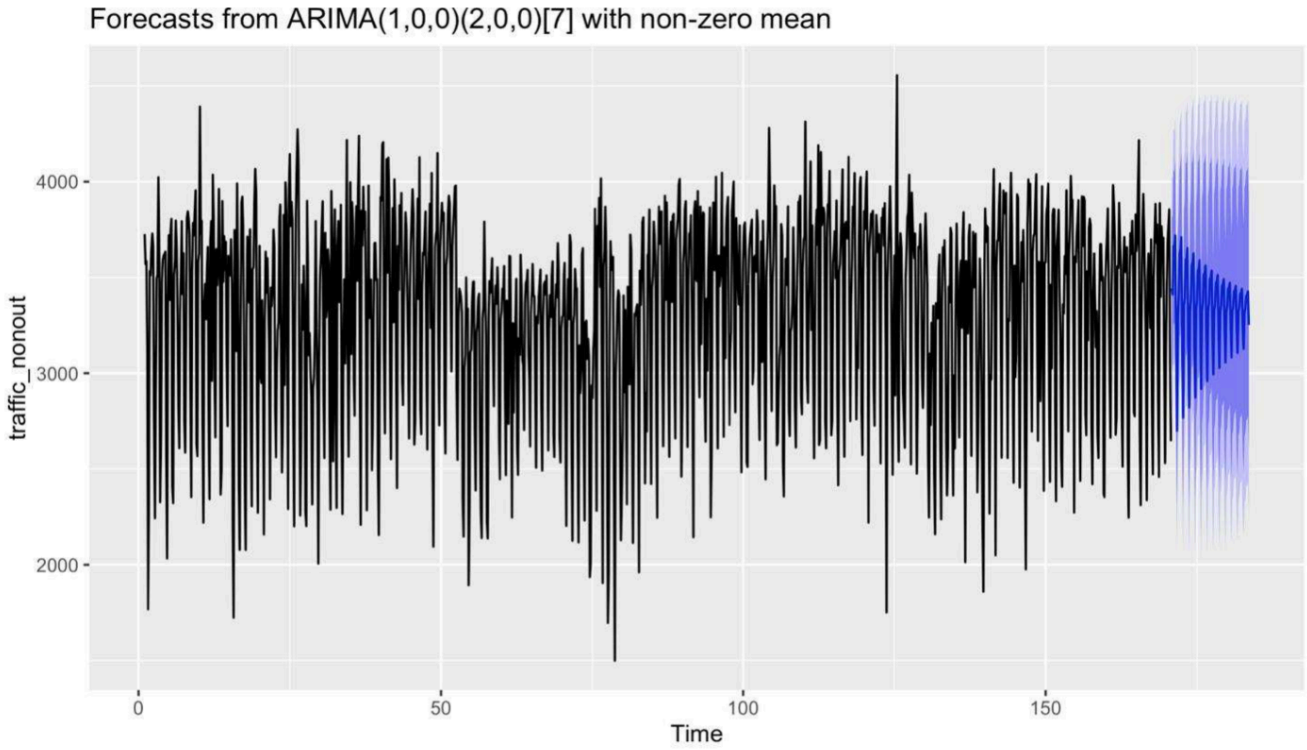
The model seems to describe the trend of the original series very well, even capturing the lower and upper peaks. This is also confirmed by the R2 value of approximately 66%, which means the RegARIMA explains 66% of the total variance, which is good.

Forecasting

As a final step, we are going to look at the short-term forecasts of future values made by the best model chosen in the previous section.

Since the best model is the RegARIMA, to which we cannot apply the `forecast::forecast` function in R because it requires both a training and a testing dataset to perform the predictions, we choose the second-best model, the SARIMA.

What we obtain are the forecasts for the three months following the end of our data:



From the plot, we observe that the blue lines represent the forecasts, while the surrounding purple lines represent the 95% confidence interval for the outermost one, and the 80% confidence interval for the innermost one. We note that the confidence interval becomes increasingly wider as time increases, and this is due to the fact that the further ahead one goes in time, the more uncertain and less precise the predicted values become.

Despite this, we can observe that the SARIMA forecasts a trend similar to that present in the original series for the closest days and weeks, while as time progresses, the predicted value for traffic volume appears to decrease steadily. This is likely due to what was mentioned before: the further ahead one goes in time, the less precise the forecasts are, and they tend to assume a value close to the mean.

Conclusions

In order to answer the questions we posed at the beginning of this paper, we performed several analyses. We can conclude that the results obtained from these analyses are very good.

This is particularly true for the exploratory analysis, which allowed us to answer our initial questions right away. The estimated models—ARIMA, SARIMA, RegARIMA, and the regression—also provided us with optimal results; in fact, they all had a data fit of approximately 60%, which we can consider good. The only technique that did not provide satisfactory results was the regression model with traffic as a function of time and the `rest_day` variable, from which we expected a very good data fit, not 0.5%, especially since traffic seemed to be highly influenced by both temperature and the type of day of the week (working or not).

Despite this, we were still able to optimally answer the research questions. We have indeed seen how traffic is influenced by bad weather, as it decreases with increasing cloudiness and lowering temperatures. We then noted that traffic is greater on working days than on weekends or holidays, and this is due to several factors: commuting (people are obliged to travel during the week to reach

their workplace), school or sports activities, but also the traffic produced by commercial activities that ship or receive supplies.

It was also noted that among weekdays, the one with the most traffic is Friday, and this is likely due to all those people who work in other cities during the week and return home on the weekend.

We then confirmed the fact that if the weather is good on the weekend, traffic increases, and this is because people tend to travel more, perhaps for day trips.

Finally, we also noted how the SARIMA model predicts the traffic trend quite well in the days following the end of our dataset, although the values are very uncertain and unreliable, especially in the long term, as we cannot predict atmospheric phenomena—especially in this period of climate crisis—that could affect road traffic.