

Analisi di una serie storica

Introduzione

L'obiettivo dell'elaborato è quello di andare ad analizzare il traffico stradale in Minnesota, USA, in particolare nel tratto che va da Minneapolis a St. Paul, in relazione ad altri fattori come il meteo, la temperatura e i giorni festivi/week end.

Le domande di ricerca, alle quali vogliamo dare una risposta tramite l'analisi di serie storiche, sono le seguenti:

1. Il traffico è influenzato dal maltempo?
2. Il traffico è influenzato dalla temperatura?
3. Il traffico è influenzato dalle festività?

Per rispondere alle domande ho effettuato prima una analisi esplorativa delle serie storiche, in cui ho verificato la normalità della serie, la stazionarietà ed eventuali trend e stagionalità. In seguito, ho creato i modelli di regressione lineare semplice, ARIMA, SARIMA e regARIMA.

Il Dataset è stato preso da "UCI Machine Learning repository" e contiene le seguenti variabili:

- traffic_volume: volume del traffico orario
- clouds_all: percentuale oraria di cielo coperto
- temp: temperatura media oraria in Kelvin
- holiday: festività per ogni giorno (es: 'Christmas Day'; se non c'è nessuna festa 'None')
- rain_1h: mm di pioggia caduta in un'ora
- snow_1h: mm di neve caduti in un'ora
- date_time: data e ora
- weather_main: descrizione del tempo (es: 'Clouds' ; 'Clear'...)

Studio delle variabili del dataset

Importo le librerie necessarie e carico il dataset:

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

```

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

```

Le variabili di interesse per svolgere le analisi sono: traffic_volume; clouds_all; holiday; temp; date. Per prima cosa ho trasformato 'Holiday' in una variabile fattoriale e 'date' in una variabile in formato data. In seguito, avendo dati orari ho dovuto aggregarli per ottenerli giornalieri:

- Ho fatto una media per clouds_all, traffic_volume e temp
- Ho fatto una somma dei mm di pioggia/neve sia per rain_1h che per snow_1h
- Per Holiday ho preso il primo valore che c'era ogni giorno
- Per weather_main ho fatto la moda
- Ho poi creato una variabile chiamata 'is_holiday_day' che assume valore 1 se il giorno preso in considerazione è festivo e valore 0 se non lo è, ed eliminato la variabile 'holiday' precedente.

```

dataset <- dataset %>%
  mutate(date_time = as.Date(date_time)) # Mantiene solo "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")

```

```

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))

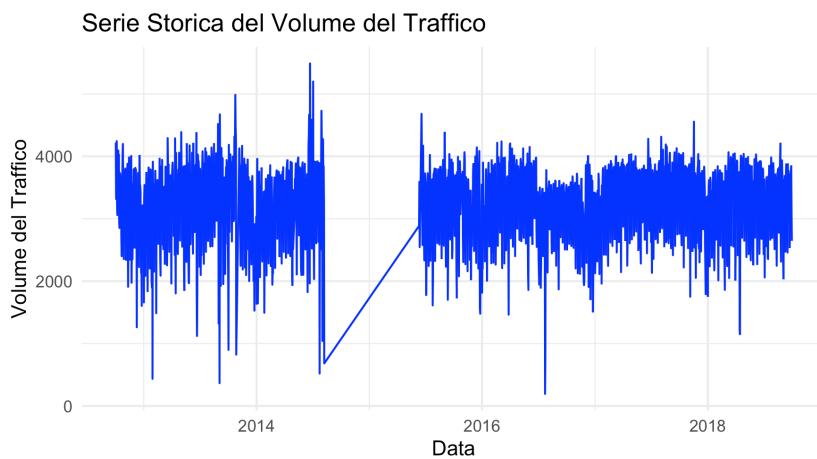
```

Come si può vedere nel seguente grafico in cui viene rappresentata la serie storica della variabile mean_traffic, c'è un buco di un anno nei dati:

```

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()

```



Ho quindi deciso di tenere in considerazione solo gli anni che vanno da fine 2015 a fine 2018.

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

Analisi esplorativa

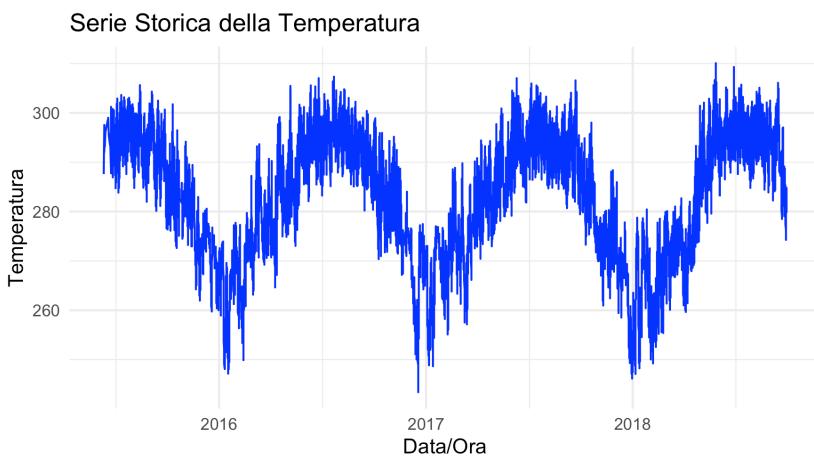
Sulle variabili precedentemente elencate ho fatto un'analisi esplorativa comprendente:

- Istogramma e box plot per vedere la distribuzione della variabile
- Gestione possibili outlier
- Verifica della normalità della variabile analizzata tramite il test di Bera-Jarque
- Grafici ACF e PACF per vedere la persistenza e l'autocorrelazione
- Test Ljung-Box e Box-Pierce per verificare se la serie è la realizzazione di un processo whitenoise
- Box-Cox per vedere se è necessario applicare una trasformazione alla variabile per rendere la distribuzione normale
- Test di Dickey Fuller Aumentato per verificare la stazionarietà della serie.

Se la serie risulta non stazionaria a causa della presenza di un trend ho eseguito una detrendizzazione nel caso di trend deterministico e una differenziazione nel caso stocastico. Se invece la serie risulta stagionale, l'ho destagionalizzata tramite regressione armonica o altre tecniche.

Variabile mean_temp

La prima variabile che analizzo è la temperatura media: essa risulta avere frequenza annuale:



Osservando il grafico possiamo aspettarcici la presenza di stagionalità nella serie e anche di un trend lineare crescente.

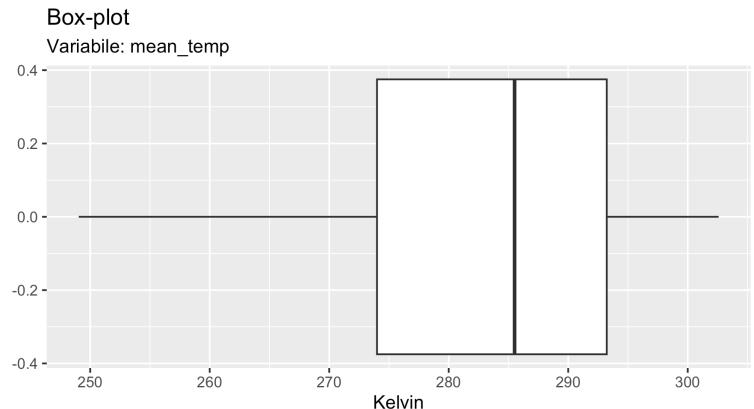
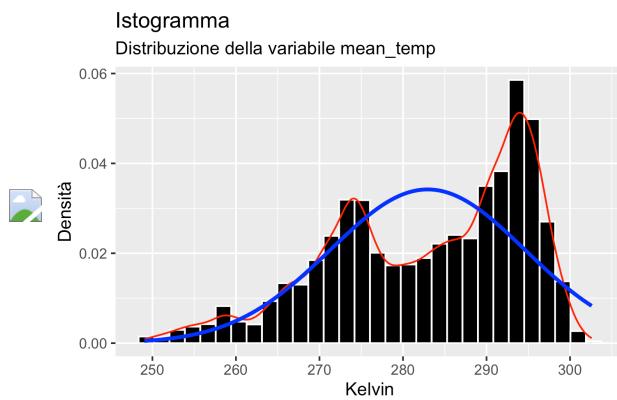
Vediamo ora l'istogramma e il box plot per studiare la sua distribuzione:

```

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")

```



Dall'istogramma possiamo dedurre che la variabile non ha una distribuzione normale, come si nota dalla curva gaussiana e dalla kernel density che non si sovrappongono. Sembra esserci una forte asimmetria con coda più lunga a sinistra, confermata anche dal box-plot. Questo non ci stupisce: durante l'anno le temperature più frequenti si trovano tra gli 0 e i 20 gradi con la mediana attorno ai 12/13°C (285K-273K).

Dal box plot notiamo anche l'assenza di outlier.

Per confermare quanto appena osservato dai due grafici, facciamo il test di Bera-Jarque che valuta l'ipotesi nulla che i dati di input si distribuiscono normalmente contro l'ipotesi alternativa di non normalità.

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

Ottenendo un p-value molto basso, inferiore a 2.2e-16, rifiuto l'ipotesi di normalità e confermo quanto supposto in precedenza.

Mostriamo ora i grafici relativi a ACF e PACF:

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

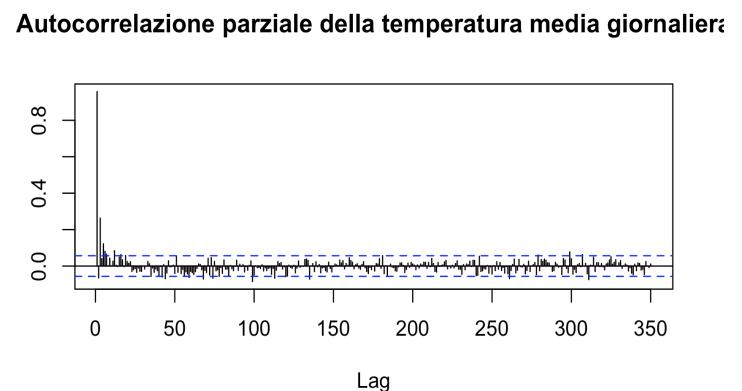
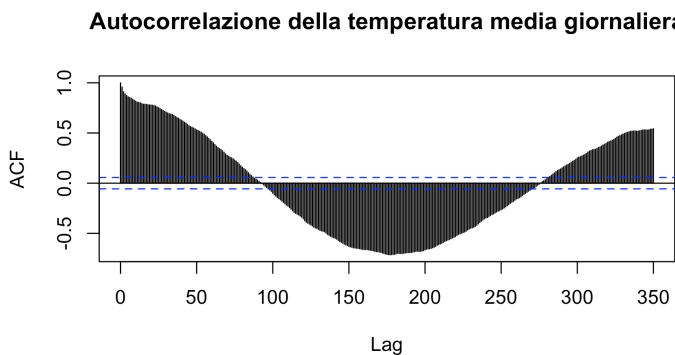
acf(dataset$mean_temp, main = "Autocorrelazione della temperatura media giornaliera")
```

```

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

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

```



Dal grafico dell'ACF notiamo la presenza di stagionalità in quanto c'è una forte correlazione tra la variabile e sé stessa al lag1, lag2,... sembra quindi essere una serie stagionale con buona persistenza anche perché il decadimento verso lo 0 delle correlazioni sembra essere abbastanza lento. Dal grafico delle PACF, che misura il legame tra la variabile al tempo t e se stessa al tempo t-k senza l'influenza dei ritardi intermedi, notiamo che le correlazioni più problematiche si trovano ai lag iniziali: effettivamente la temperatura è strettamente correlata a quella dei giorni precedenti.

Questa serie, inoltre, può essere considerata un processo non ergodico in quanto la memoria è a lungo termine, cioè c'è una forte dipendenza tra le osservazioni in tempi molto distanti; infatti, la temperatura di un determinato giorno può essere influenzata da fattori avvenuti giorni, settimane e addirittura anni prima.

A conferma del fatto che le correlazioni sono statisticamente significative si sono svolti anche i test di Ljung Box e Box Pierce:

```

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)
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 questi test l'ipotesi alternativa afferma che almeno un coefficiente di autocorrelazione risulti significativamente diverso da zero. Poichè in entrambi il valore del p-value è nullo, si rifiuta l'ipotesi che la serie sia la realizzazione di un processo White Noise.

A questo punto, dato che abbiamo confermato la non normalità della variabile mean_temp, facciamo Box-Cox con i metodi della 'log-likelihood' e di 'guerrero' per vedere se bisogna applicare qualche trasformazione per rendere la distribuzione normale:

```

lambda_guer <- forecast::BoxCox.lambda(dataset$mean_temp, method = "guerrero", 1)

lambda_loglik <- forecast::BoxCox.lambda(dataset$mean_temp, method = "loglik", 1

#in questo caso vengono circa uguali, circa #3, per semplicità consideriamo sc

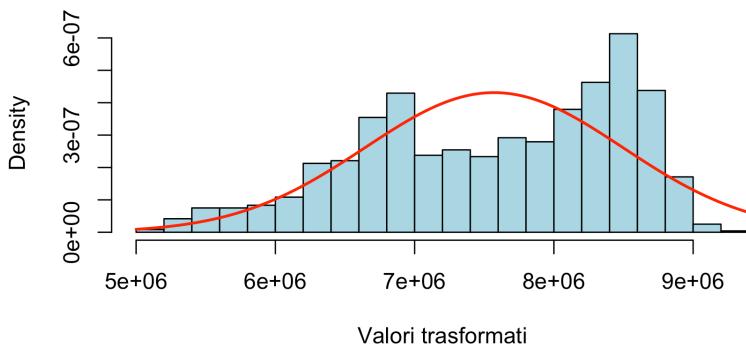
dataset$mean_temp_bc_guer <- forecast::BoxCox(dataset$mean_temp, lambda = lamb

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)

```

Istogramma della serie trasformata (Guerrero)



Dato che per la trasformazione alla terza la distribuzione dei dati non cambia, non vale la pena perdere interpretabilità per eseguire una trasformazione, pertanto proseguiamo con la variabile mean_temp.

Verifichiamo con il test ADF la stazionarietà della serie alla quale però abbiamo dovuto prima togliere i valori mancanti. Partiamo dal test ADF con type="trend", cioè con la presenza di un trend lineare deterministico:

```
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=12)

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

ottenendo i seguenti 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:
```

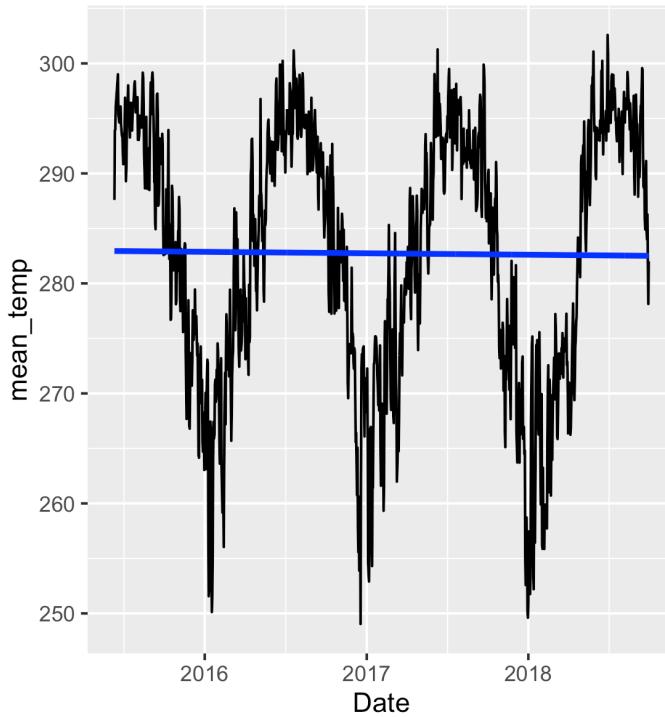
	1pct	5pct	10pct
tau3	-3.96	-3.41	-3.12
phi2	6.09	4.68	4.03
phi3	8.27	6.25	5.34

- tau3 verifica l'ipotesi che ci sia una radice unitaria nella parte autoregressiva del modello. Dato che i valori critici sono minori della statistica test, rifiutiamo H₀, quindi non abbiamo una radice unitaria e la serie risulta stazionaria.
- phi2 è un altro test associato alla regressione, dal valore ottenuto concludiamo che il trend non è significativo.
- phi3 serve per testare l'ipotesi secondo cui sia la costante che il trend siano nulli, ma dai valori ottenuti concludo che almeno uno dei due sia significativo.

Dato che rifiutiamo in tutti e tre i casi H0, la serie è stazionaria intorno al trend deterministico lineare. Non è quindi necessario continuare con gli altri due tipi di test ADF (type="drift" e type="None"). Per rendere la serie stazionaria intorno allo 0 dovrei fare una detrendizzazione (lineare), ovvero sottrarre il trend dalla serie.

```
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)
```



La retta blu è il trend lineare, che è crescente ma in maniera quasi impercettibile. Dato che il trend non è significativo, decidiamo di non proseguire con la detrendizzazione e di fare solo la destagionalizzazione.

In questo caso ho destagionalizzato la serie tramite la regressione armonica usando la serie di Fourier per approssimare seno e coseno con un polinomio di grado k. In questo caso abbiamo creato 6 polinomi di grado k=1,2,3,4,5,6.

```
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)

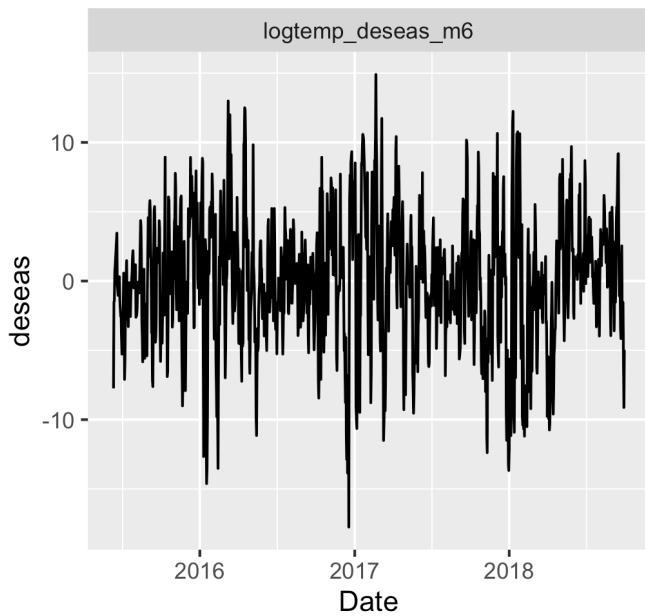
perf_cv_m1 <- CV(obj = K1)
perf_cv_m2 <- CV(obj = K2)
perf_cv_m3 <- CV(obj = K3)
perf_cv_m4 <- CV(obj = K4)
perf_cv_m5 <- CV(obj = K5)
perf_cv_m6 <- CV(obj = K6)
perf_cv <- as.data.frame(rbind(perf_cv_m1, perf_cv_m2, perf_cv_m3, perf_cv_m4, perf_cv_m5, perf_cv_m6))
cbind(Model = c("M1", "M2", "M3", "M4", "M5", "M6"), perf_cv)

```

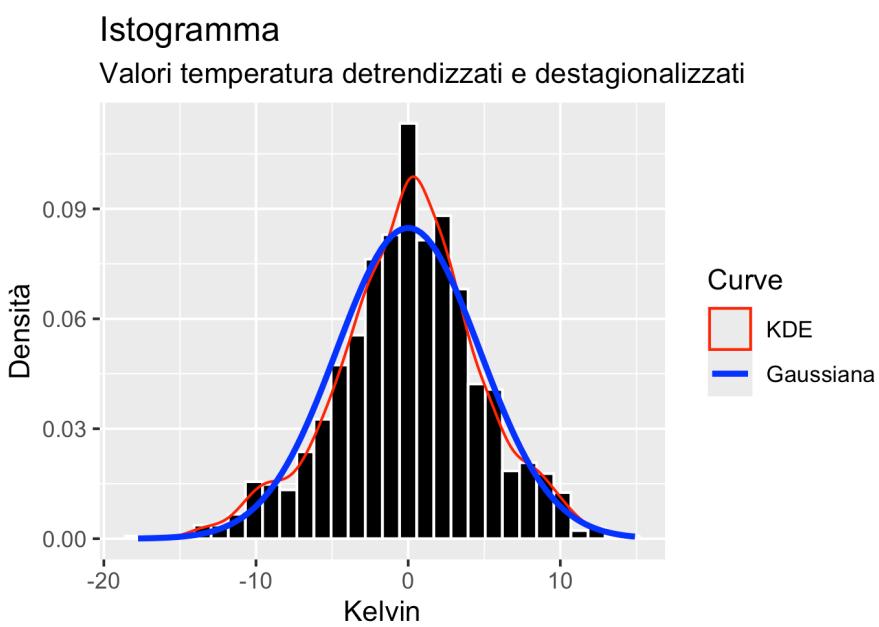
	Model	CV	AIC	AICc	BIC	AdjR2
perf_cv_m1	M1	24.28485	3826.610	3826.660	3852.056	0.8270272
perf_cv_m2	M2	23.43877	3783.838	3783.932	3819.463	0.8333659
perf_cv_m3	M3	23.09785	3766.022	3766.173	3811.825	0.8360957
perf_cv_m4	M4	22.68602	3744.229	3744.451	3800.211	0.8393141
perf_cv_m5	M5	22.66414	3742.808	3743.115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

Tramite le metriche del CV, AIC, AICc, BIC e R2_adj ho scelto il polinomio di grado 6 come quello che meglio approssima l'andamento della serie stagionale con un adattamento ai dati molto buono pari all'84%.

Pertanto, la serie destagionalizzata risulta essere:

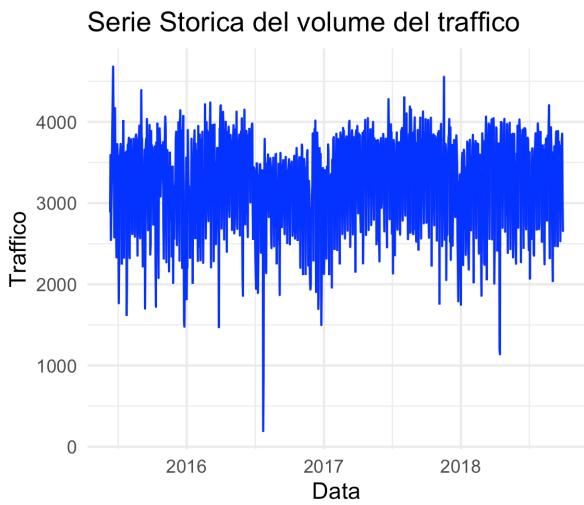


La serie destagionalizzata sembra essere stazionaria con media zero, come confermato dal test ADF di type="None" svolto sulla serie che ha portato al rifiuto di H₀, ed ha distribuzione normale, come possiamo vedere dal seguente istogramma:



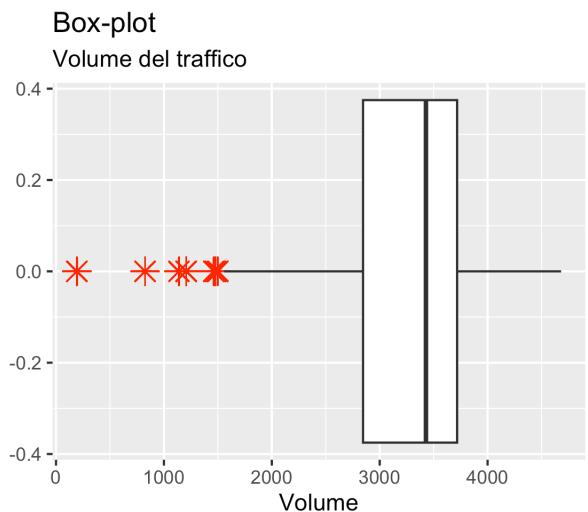
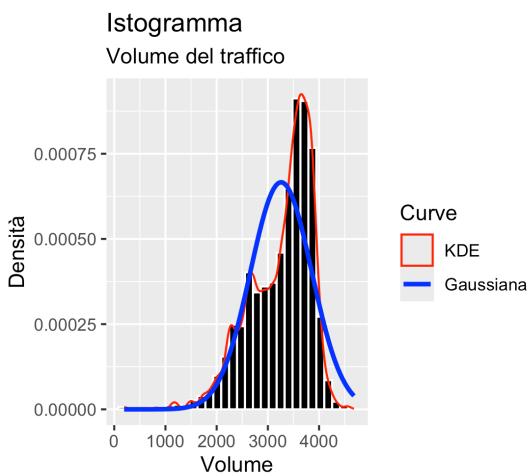
Variabile mean_traffic

Omettendo i codici utilizzati, analoghi a quelli riportati in precedenza, la seconda variabile da analizzare è quella del volume del traffico con grafico:



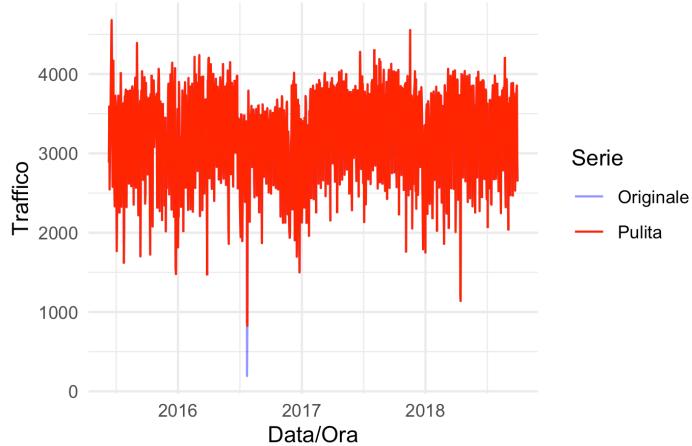
da cui risulta esserci una stagionalità settimanale e nessun trend. Notiamo anche la presenza di un picco bassissimo a circa metà del 2016.

Vediamo ora la distribuzione di traffic_volume con l'istogramma e il box-plot:



Dall'istogramma notiamo che la serie di Traffic non ha una distribuzione normale, come ci indica la kernel density, ma risulta fortemente asimmetrica con una coda lunga a sinistra. Quanto detto è confermato anche dal grafico del box-plot, dal quale notiamo la presenza di diversi outlier inferiori. Si decide di risolvere il problema usando la funzione tsClean() di R e ciò che si ottiene è la seguente serie:

Confronto serie storica con e senza outlier

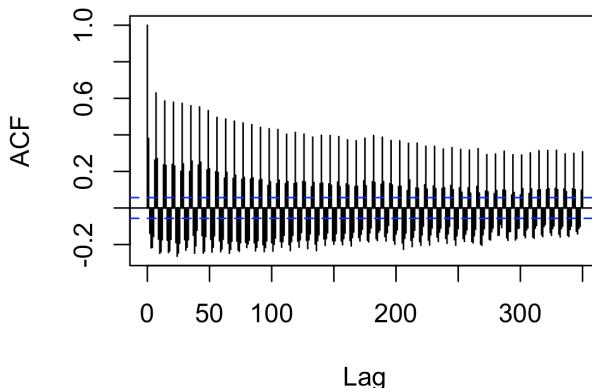


Dalla quale notiamo che il valore bassissimo tra il 2016 e il 2017 già messo in evidenza precedentemente. Da un analisi più approfondita del dataset questo viene spiegato da alcuni valori mancanti che potrebbero creare problemi con la destagionalizzazione.

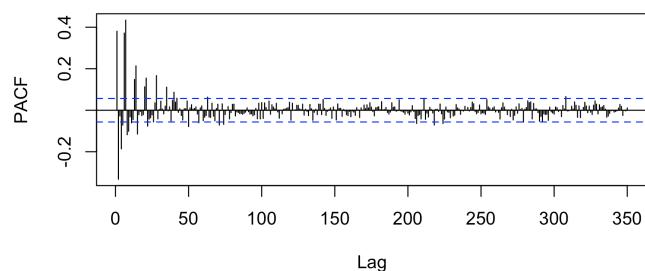
Quanto osservato nei grafici viene confermato dal test di Bera-Jarque e si rifiuta quindi l'ipotesi nulla di normalità della distribuzione della serie storica.

Andiamo ora a indagare la persistenza della serie tramite i grafici dell'ACF e del PACF.

Autocorrelazione del volume di traffico



Autocorrelazione parziale del volume di traffico



Dal grafico delle ACF notiamo che c'è una elevata persistenza delle serie, in quanto le correlazioni tendono a zero molto lentamente. Inoltre, notiamo un aspetto interessante dovuto all'alternanza di lag correlati negativamente e lag correlati positivamente e più nello specifico: i primi tre lag sono

correlati negativamente; i successivi quattro positivamente; i successivi tre negativamente etc... a riconferma quindi della presenza di una stagionalità settimanale.

A conferma della significatività delle correlazioni si sono svolti anche i test di Ljung Box e Box Pierce che, dato il valore del p-value in entrambi i casi pari 0.00, ci hanno portato al rifiuto dell'ipotesi di normalità della serie.

Analogamente a quanto fatto per la variabile mean_temp, le trasformazioni ottenute con i metodi della 'log-likelihood' e di 'guerrero' si ottengono distribuzioni che cambiano leggermente rispetto all'originale. Si decide quindi di proseguire con questa.

Dai valori ottenuti con il test ADF:

```
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:
```

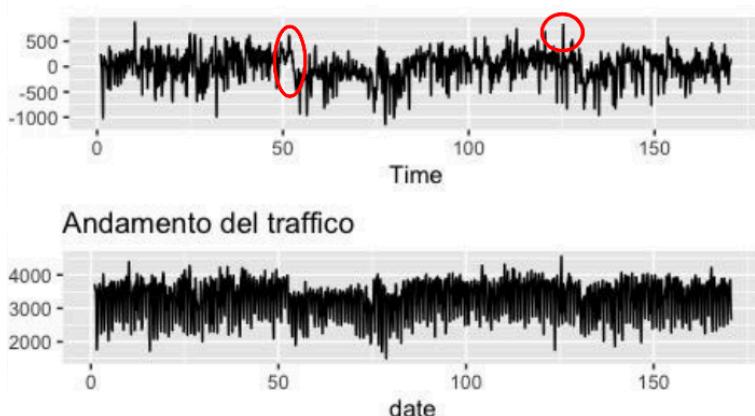
	1pct	5pct	10pct
tau3	-3.96	-3.41	-3.12
phi2	6.09	4.68	4.03
phi3	8.27	6.25	5.34

la serie risulta essere stazionaria intorno ad un trend deterministico. Per rendere la serie stazionaria intorno allo 0 dobbiamo fare una detrendizzazione, in questo caso accompagnata dalla destagionalizzazione. Ho creato tre polinomi K1, K2, K3 con grado rispettivamente 1,2,3 in cui la variabile traffic si trova in funzione del trend e della regressione armonica svolta tramite fourier. Come abbiamo fatto in precedenza, sceglieremo il modello migliore tramite il confronto delle seguenti metriche:

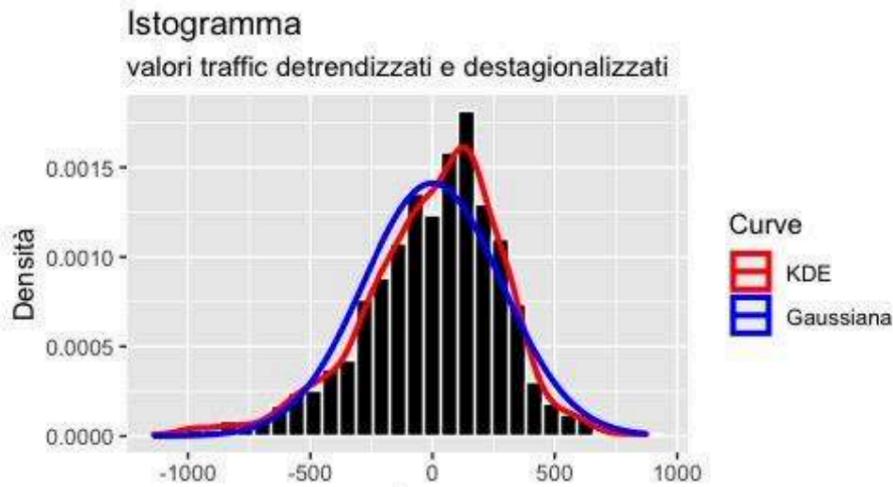
	Model	CV	AIC	AICc	BIC	AdjR2
perf_cv_m1	M1	742.7931	7841.887	7841.921	7862.201	0.04712032
perf_cv_m2	M2	740.9542	7838.875	7838.947	7869.345	0.05113329
perf_cv_m3	M3	739.3720	7836.256	7836.379	7876.883	0.05481350

Sceglieremo il polinomio M3 che ha tutte le metriche tranne il BIC più basse e l'R2_adj più alto, anche se in generale l'adattamento è molto basso, pari allo 0.05%. Quindi, il polinomio che va a modellare meglio la stagionalità della serie ed anche il trend è di terzo grado.

Pertanto, la serie destagionalizzata e detrendizzata risulta essere:



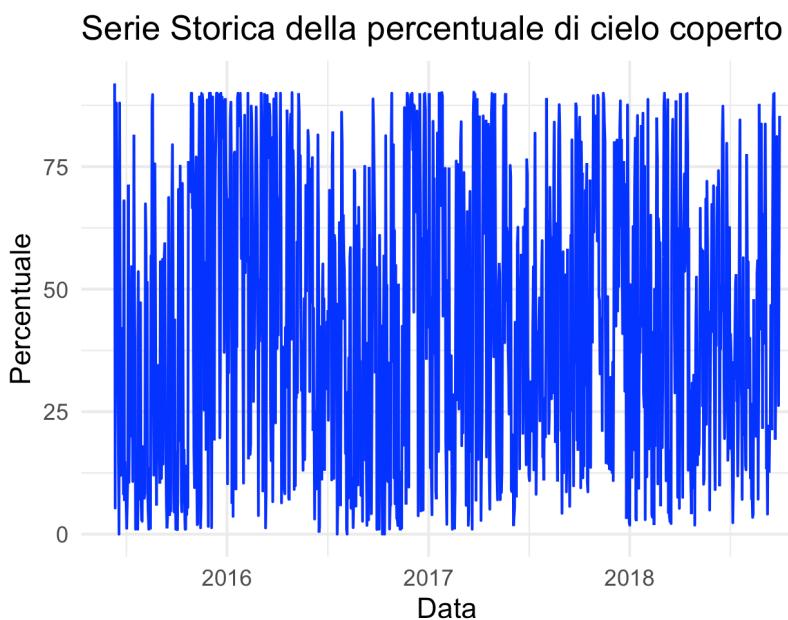
Sia dal grafico della serie detrendizzata e destagionalizzata che da quella originale notiamo la presenza di un valore elevato del traffico relativo al giorno 17/11/2017 probabilmente dovuto allo svolgimento di un importante partita di Hockey tenutasi a St. Paul. Inoltre, notiamo anche un andamento particolare evidenziato dal primo cerchio relativo ai giorni 24, 25 e 26 giugno 2016 rispettivamente un venerdì, sabato e domenica. Generalmente si nota che il traffico il sabato e la domenica è inferiore rispetto ai giorni lavorativi, ma in questo caso il picco in negativo è più evidente perché il 25 e il 26 giugno c'è stata una tempesta che ha portato le persone a spostarsi di meno.

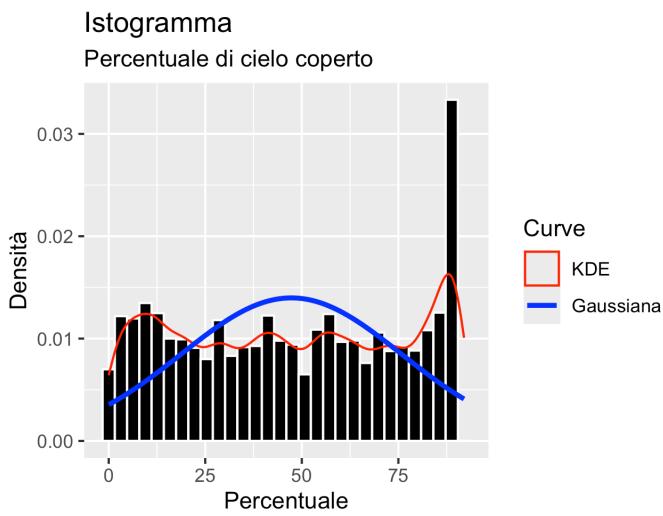


Come possiamo vedere dal grafico la serie ha distribuzione praticamente normale e il test ADF di type="none" conferma la stazionarietà della serie.

Variabile clouds_all

La terza serie sulla quale facciamo l'esplorativa è la percentuale di cielo coperto, che ha frequenza annuale:

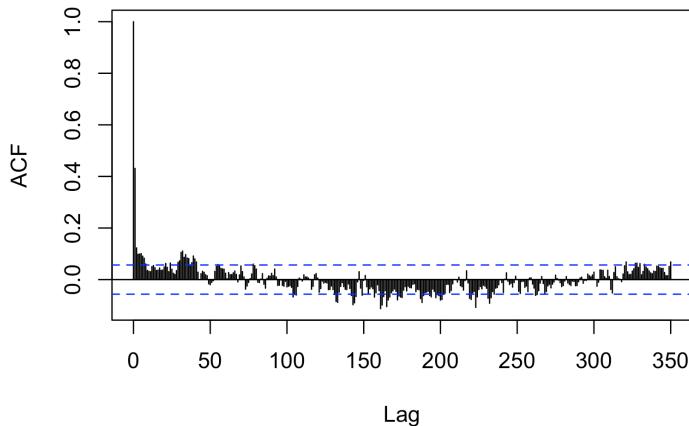




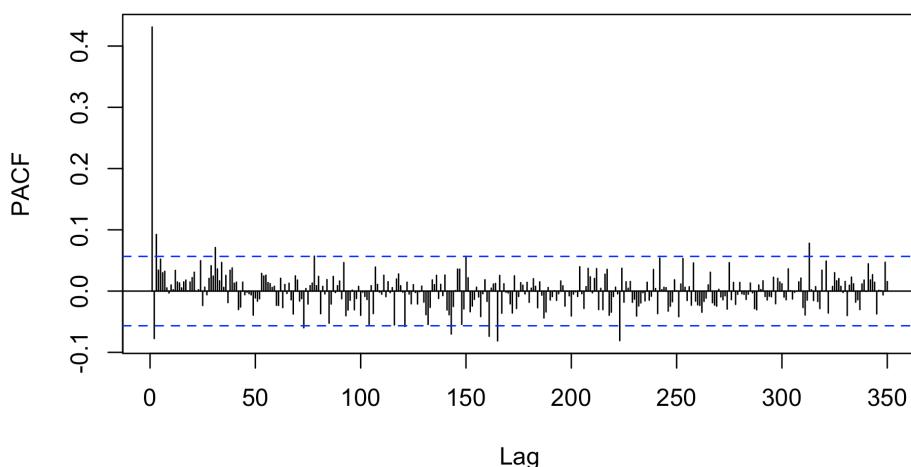
```
{width="356"}
```

Dal grafico dell'istogramma notiamo che la variabile che descrive la percentuale di nuvolosità non ha una distribuzione normale, come ci indica la kernel density, ma anzi sembra essere uniforme. Questo è confermato dal box-plot in cui possiamo anche notare che la variabile è simmetrica con la mediana che si trova intorno al 40% di nuvolosità. Non sembrano esserci outlier. Dall'output del test di Bera-Jarque confermiamo quanto detto.

Autocorrelazione della percentuale di cielo coperto



Autocorrelazione parziale della percentuale di cielo coperto



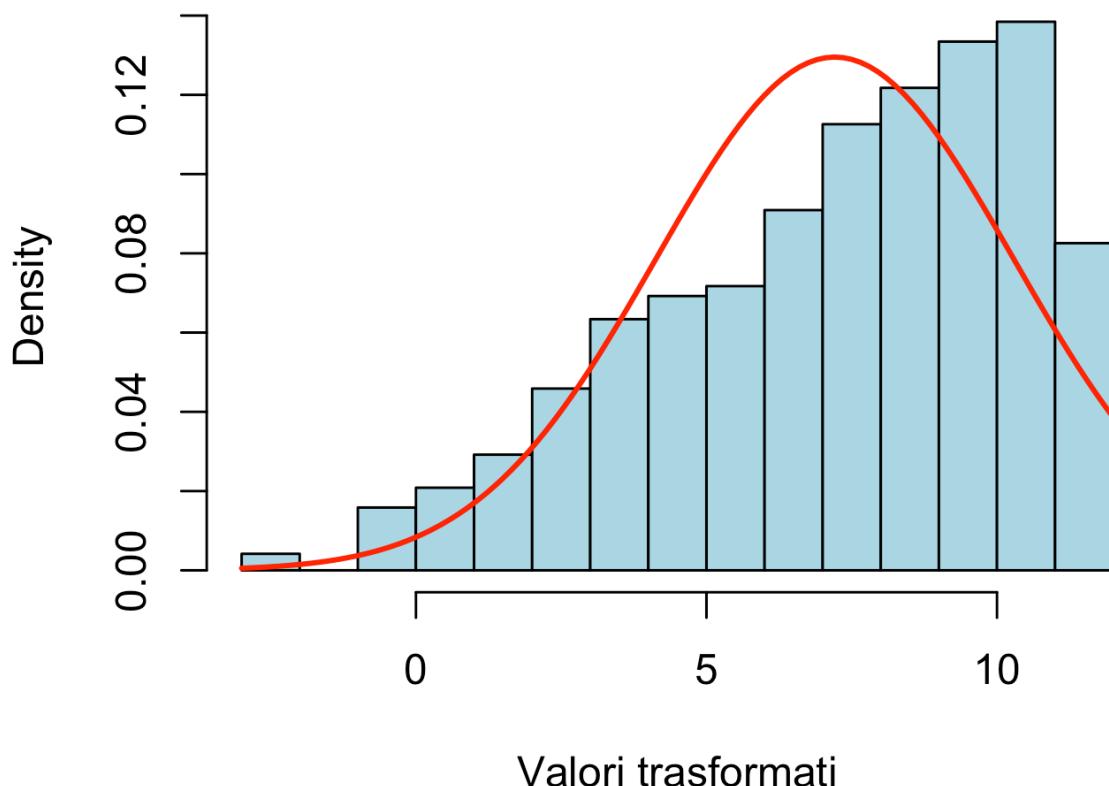
Dal grafico delle ACF possiamo notare che la serie è poco persistente, in quanto le autocorrelazioni tendono a zero molto velocemente.

Guardiamo ora il grafico delle PACF, dal quale notiamo che le correlazioni parziali sono tutte non significative ad eccezione del primo lag che ha correlazione negativa. Questo però non dovrebbe essere un particolare problema dato il valore molto piccolo pari a -0,1.

Svogliamo ora i test di Ljung-Box e Box-Pierce dalla quale otteniamo un p-value bassissimo il quale ci porta al rifiuto dell'ipotesi nulla che la serie sia realizzazione di un processo White Noise. Pertanto nella serie c'è persistenza, anche se poca.

Similmente a quanto fatto per le altre variabili, Box Cox con i metodi della 'log-likelihood' e di 'guerrero' ci suggeriscono trasformazioni inadatte che farebbero perdere la simmetria della variabile: si osservi per esempio quanto ottenuto con 'guerrero'

Istogramma della serie trasformata (Guerrero)



dove la variabile diventa fortemente asimmetrica con una coda più lunga a sinistra.

Anche in questo caso dall'output del test ADF, la serie risulta essere stazionaria attorno a un trend deterministico:

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

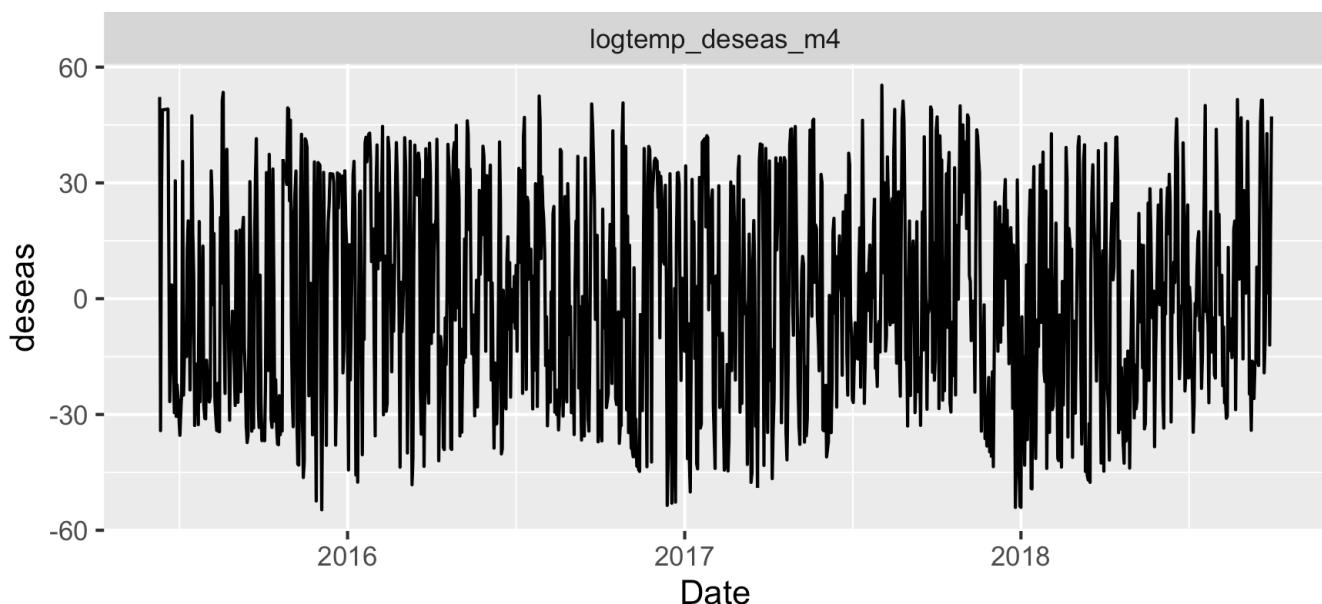
Critical values for test statistics:

	1pct	5pct	10pct
tau3	-3.96	-3.41	-3.12
phi2	6.09	4.68	4.03
phi3	8.27	6.25	5.34

Per rendere la serie stazionaria intorno allo 0 dobbiamo fare una detrendizzazione, ma dato che dal grafico dell'andamento della serie non si vede un chiaro trend, decidiamo di fare solo la destagionalizzazione sempre tramite la regressione armonica usando le serie di Fourier.

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

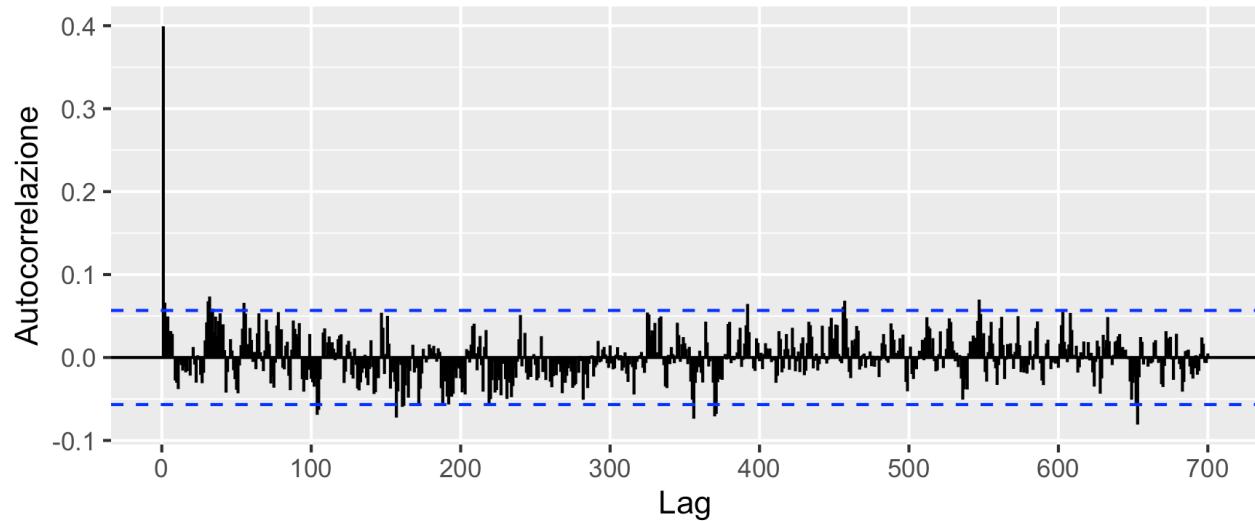
Il polinomio scelto è quello di grado 4, in quanto ha CV, AIC, AICc più bassi e R2_adj più alto, anche se possiamo notare che l'adattamento è in generale molto basso pari allo 0.05%.



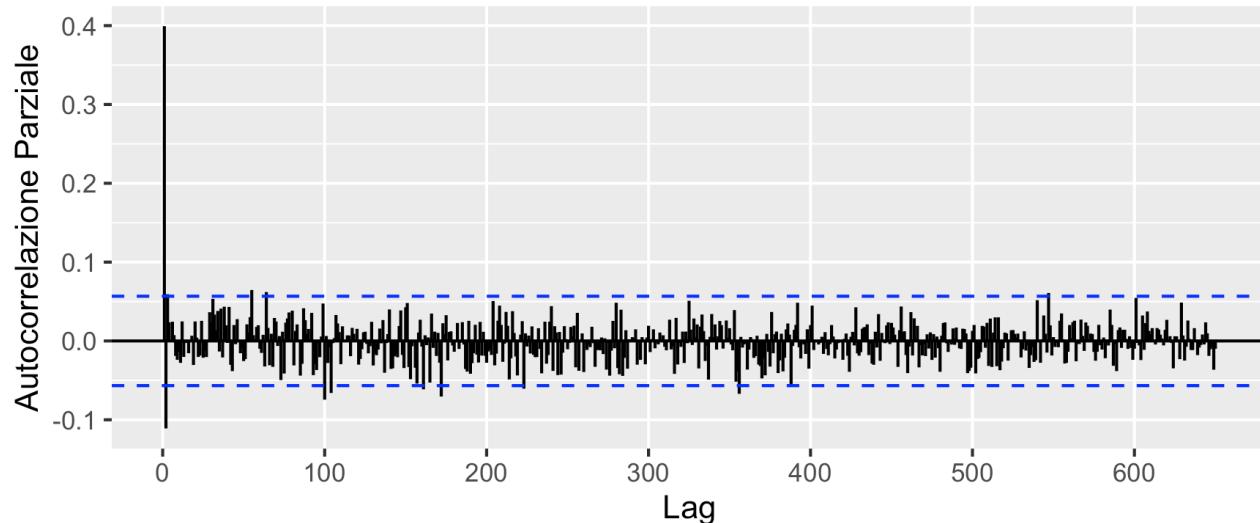
L'andamento della serie sembra essere più armonico ad indicare ancora presenza di stagionalità.

Facciamo quindi un grafico dei residui del modello 4 per vedere se c'è effettivamente stagionalità, ottenendo:

ACF dei Residui di K4



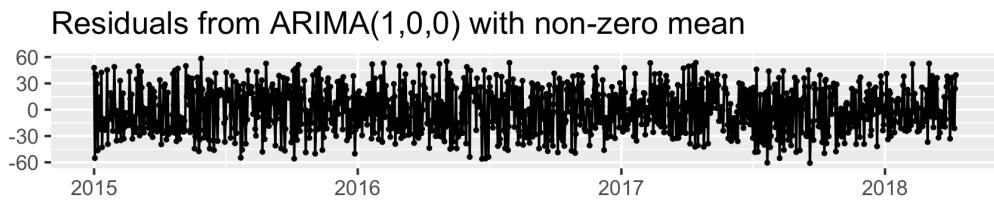
PACF dei Residui di K4



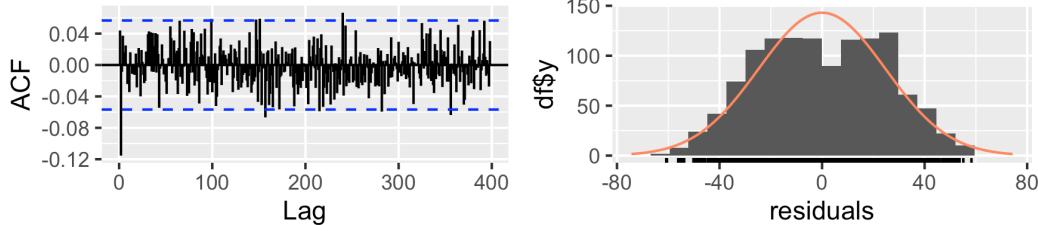
dunque la stagionalità sembra essere minima e l'unico lag con correlazione significativa è il primo. Possiamo provare a fare un AR(1):

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

checkresiduals(mod_ar1)
```



ottenendo:



Da questi grafici notiamo che tutte le autocorrelazioni sembrano non essere significative. Quindi possiamo concludere che in realtà nella serie relativa a clouds_all non c'era stagionalità che era coperta dalla parte autoregressiva.

Inoltre la serie ha una distribuzione che somiglia di più a quella di una normale rispetto a quella di partenza che era uniforme, anche se il test di Bera-Jarque mostra un p-value pari a 9.096e-09 che ci porta al rifiuto dell'ipotesi nulla di normalità della distribuzione. Inoltre, il processo è anche stazionario come ottenuto dall'output del test ADF di type="None".

Decomposizioni

Eseguiamo ora la decomposizione delle serie con lo scopo di andare ad isolare le componenti trend, stagionalità e i residui. La eseguiremo solo sulle serie relative a traffico e temperatura che hanno sia trend che stagionalità, mentre per quanto riguarda la percentuale di nuvole non c'è bisogno di fare la decomposizione. Infatti, da quello che abbiamo visto nell'esplorativa, in essa non sembra esserci un trend evidente, ma solo impercettibile, e la stagionalità, che visivamente e logicamente sembra esserci, in realtà non c'è in quanto è coperta dalla parte autoregressiva.

Variabile mean_temp

In questo caso applichiamo una decomposizione classica di tipo additivo in quanto la forza delle fluttuazioni stagionali intorno al trend rimangono costanti e non cambiano con il livello della serie. Quindi avremo che la serie storica mean_temp sarà funzione della somma di tutte le componenti:

$$\text{mean_temp} = T_t + S_t + \varepsilon_t$$

```

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

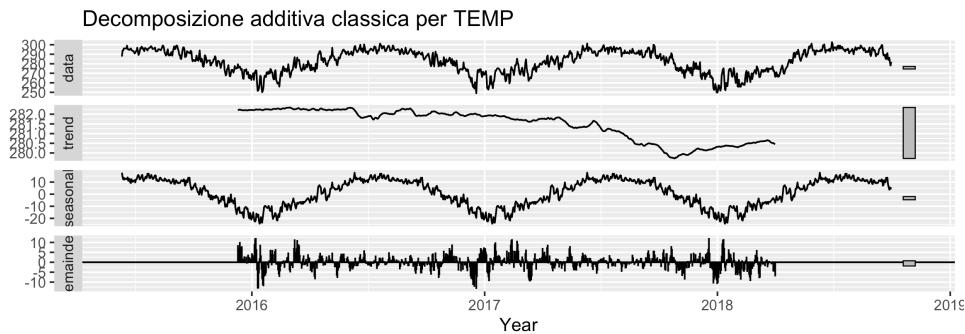
library(zoo)
TEMP <- na.locf(dataset_TS)
autoplot(TEMP) +
  labs(y = latex2exp::TeX("Kelvin"),
       title = "serie originale")

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")

```



Dal grafico notiamo la presenza di una stagionalità molto forte, come già visto nell'analisi esplorativa. Notiamo inoltre la presenza di un trend negativo, in contraddizione con quanto detto in precedenza, che però guardando la scala dei valori che va da 280.5 a 282, risulta essere poco significativo. Questa contraddizione la possiamo spiegare con il fatto che mentre nell'EDA il trend era lineare, in questo caso la componente T è calcolata usando la media mobile.

Nel caso dei residui invece, possiamo notare dal grafico che questi sembrano avere un andamento casuale attorno allo 0 e sembrano distribuirsi come dei White Noise con media pari a 0 e varianza costante.

Per quanto riguarda la serie detrendizzata, cioè: $\text{mean_temp} = S_t + \varepsilon_t$, questa risulta avere lo stesso andamento della serie originale perché, come detto, il trend osservato è minimo e quindi non significativo.

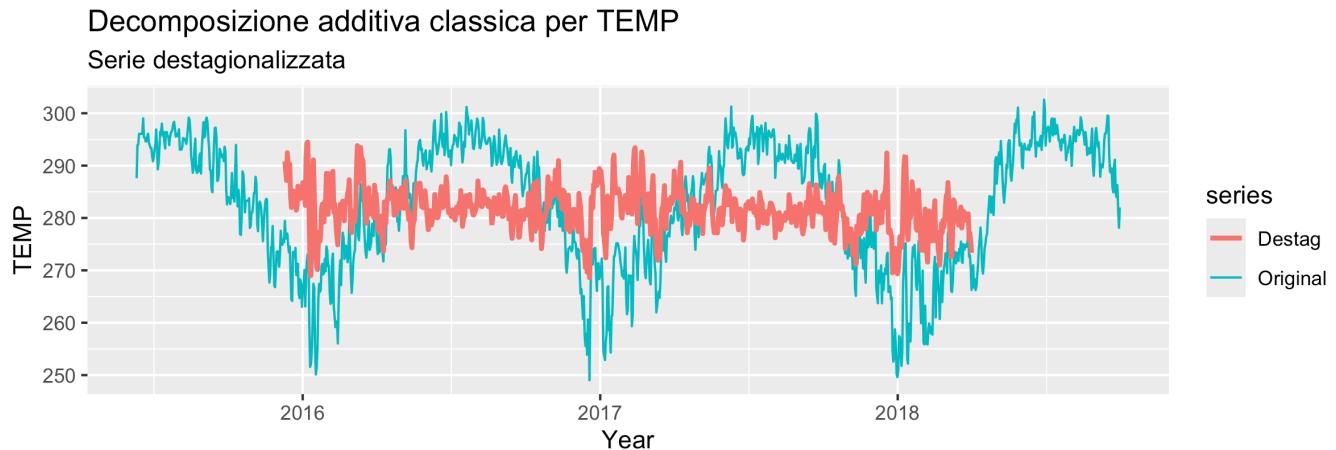
Mentre per quanto riguarda la destagionalizzazione, vediamo meglio come risulta la serie una volta estratta la componente stagionale, cioè $\text{mean_temp} = T_t + \varepsilon_t$, attraverso:

```

autoplot(TEMP,series = "Original") +
  autolayer(TEMP_add_destag,series = "Destag", size=1.01) +
  labs(x="Year",
       title = "Decomposizione additiva classica per C0",

```

```
subtitle = "Serie destagionalizzata")
```



la destagionalizzazione fatta tramite decomposizione risolve i problemi di stagionalità.

Invece analizzando la distribuzione della componente residuale, data da:

$$\varepsilon_t = \text{mean_temp} - T_t - S_t$$

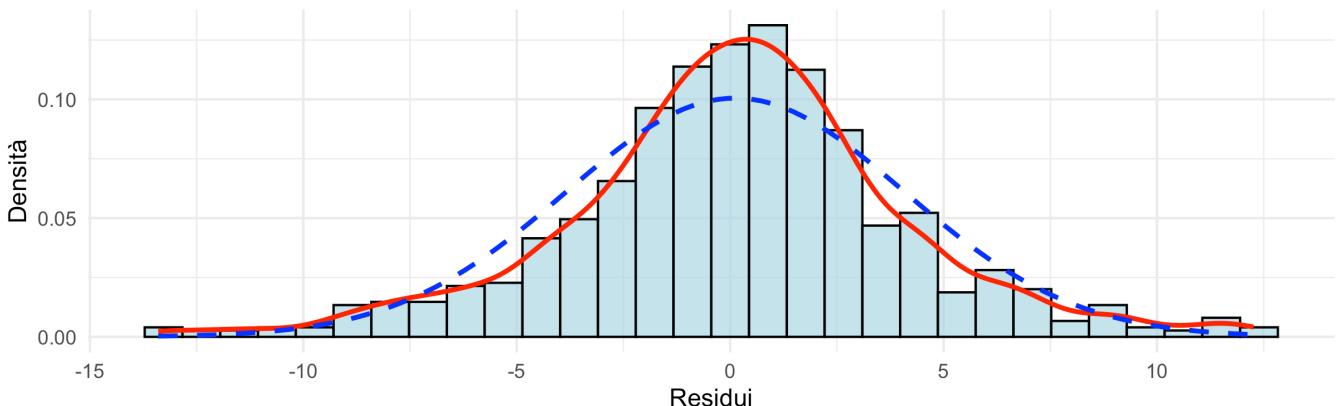
```
residui_add <- dec_add$random

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 =
  geom_density(color = "red", size = 1) +
  stat_function(fun = dnorm, args = list(mean = mu, sd = sigma), color = "blue")
  ggttitle("Istogramma dei Residui con Densità Kernel e Curva Normale") +
  xlab("Residui") +
  ylab("Densità") +
  theme_minimal()
```

Istogramma dei Residui con Densità Kernel e Curva Normale

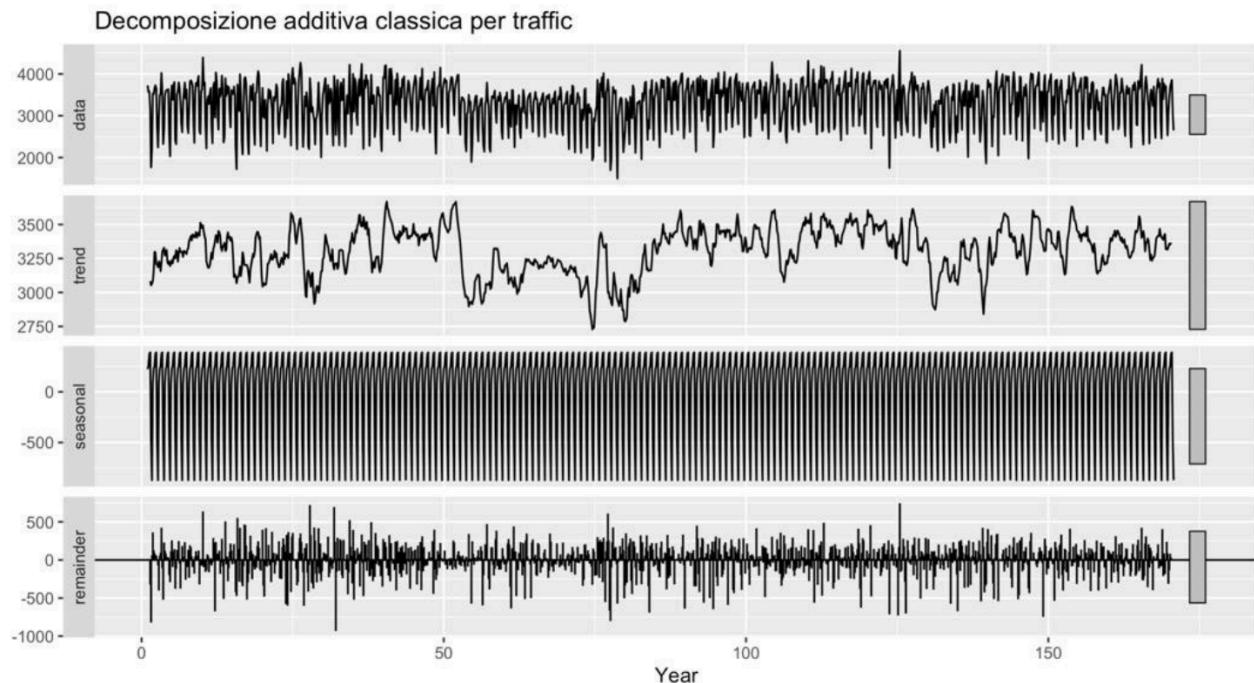


possiamo confermarne la normalità della distribuzione normale e concludiamo dicendo che i residui sono White Noise con media 0 e varianza costante.

Variabile mean_traffic

Anche in questo caso facciamo una decomposizione classica di tipo additivo. Quindi avremo che la serie storica mean_traffic sarà funzione della somma di tutte le componenti: $\text{mean_traffic} = T_t + S_t + \varepsilon_t$

Analogamente a prima rappresentiamo sia la serie storica che le sue componenti:

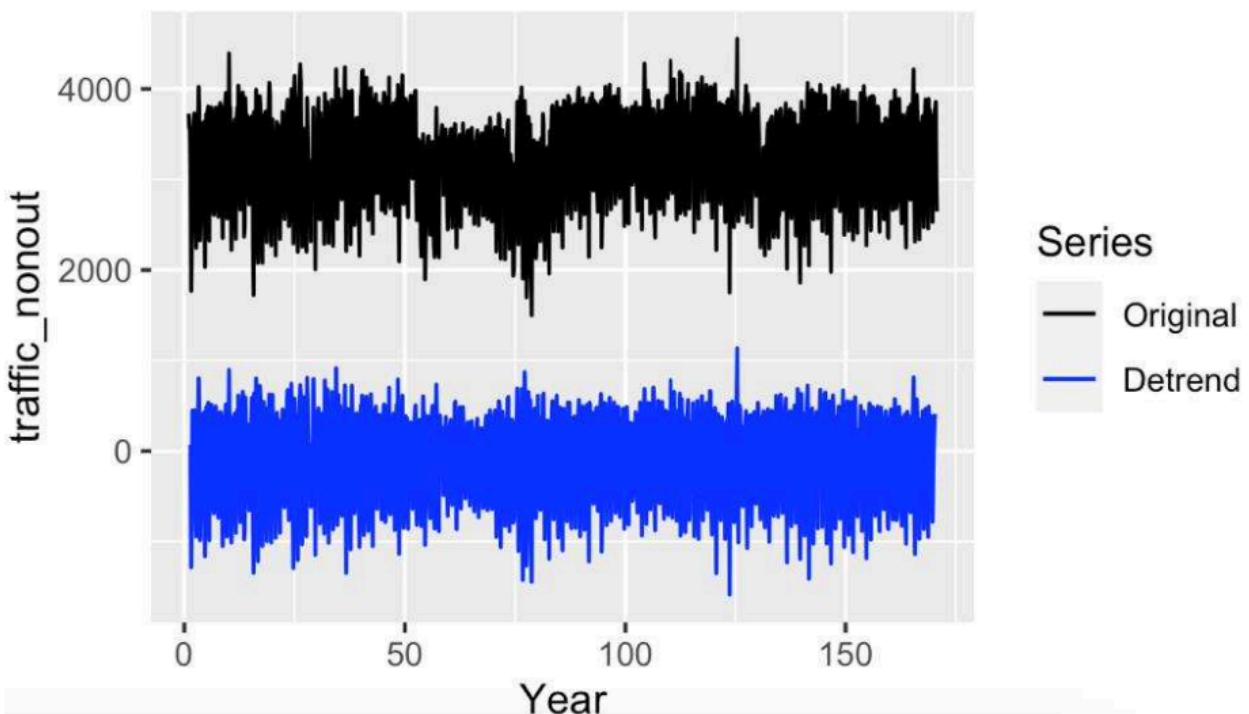


Dal grafico notiamo che sembra esserci un trend crescente, anche se minimo, e una stagionalità settimanale con picchi inferiori relativi al week end, come ci aspettavamo. Per quanto riguarda i residui notiamo che hanno un andamento casuale attorno allo 0 e sembrano quindi distribuirsi come un Withe Noise con media 0 e varianza costante, ma questo lo confermeremo dopo andando a verificare la normalità della loro distribuzione.

Vediamo graficamente la serie detrendizzata, data da $\text{mean_traffic} = S_t + \varepsilon_t$, dove la componente trend T è stata isolata:

Decomposizione additiva classica per traffic

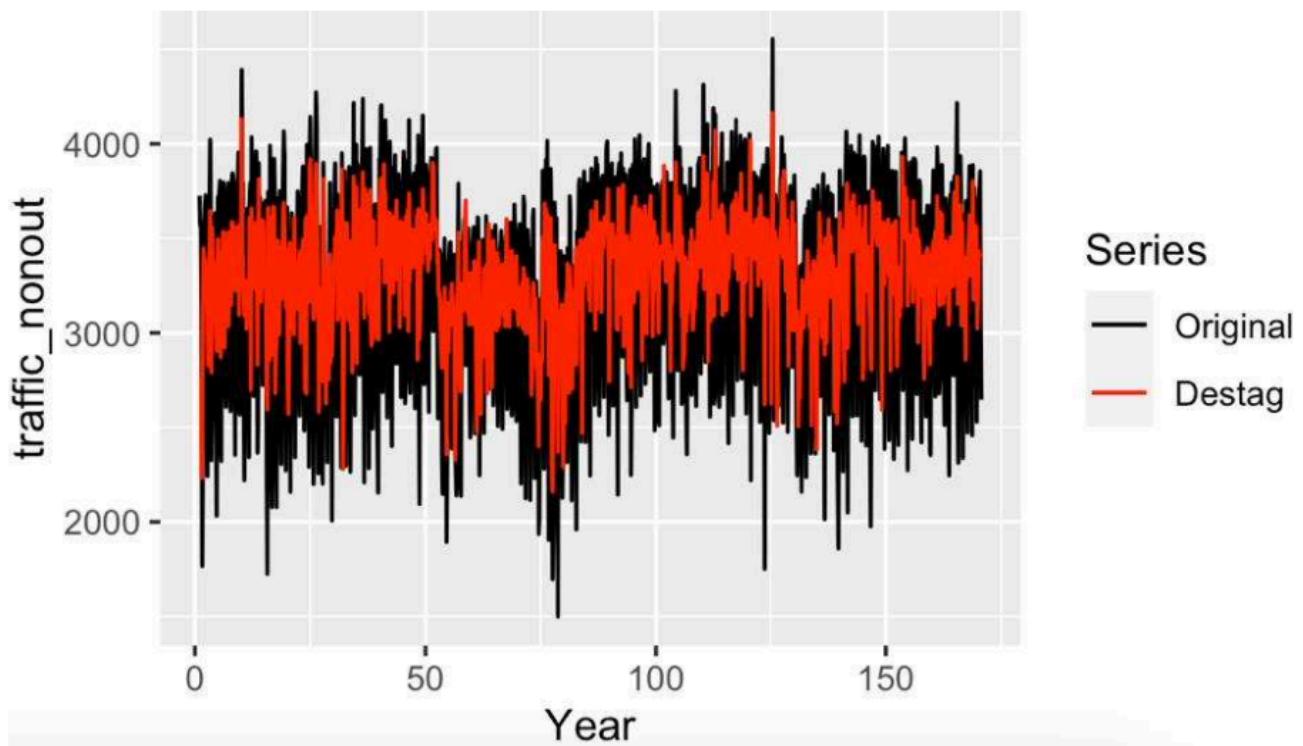
Serie detrendizzata



Dal grafico notiamo come la decomposizione abbia funzionato bene in questo caso. La serie, infatti, è stata centrata sullo 0 e sembra avere un andamento molto più stazionario rispetto a quella originale.

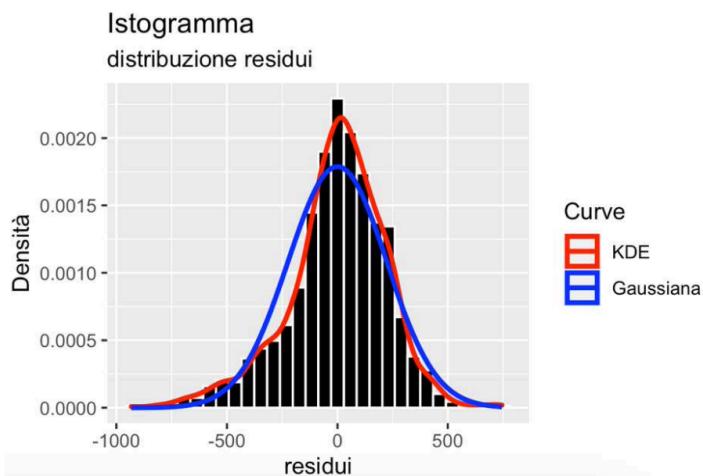
Ora sulla serie detrendizzata viene svolta la destagionalizzazione. Isoliamo la componente S_t che viene calcolata facendo una media dei valori appartenenti allo stesso periodo, ottenendo così $\text{mean_traffic} = T_t + \varepsilon_t$:

Decomposizione additiva classica per traffic Serie destagionalizzata



Guardando il grafico notiamo che la serie destagionalizzata sembra più centrata e meno variabile rispetto alla serie originale.

Invece per la componente residuale, data da $\varepsilon_t = \text{mean_traffic} - T_t - S_t$ la distribuzione dei residui è praticamente normale e quindi possiamo confermare che si distribuiscono come dei White Noise con media 0 e varianza costante:



Analisi di regressione e analisi dei residui

Prima di eseguire la regressione, ho creato una variabile chiamata `rest_day` da aggiungere al dataset, che assume valore 0 quando il giorno è lavorativo (dal lunedì al venerdì) e valore 1 quando è weekend e/o giorno festivo.

```
dataset <- dataset %>%
  mutate(
```

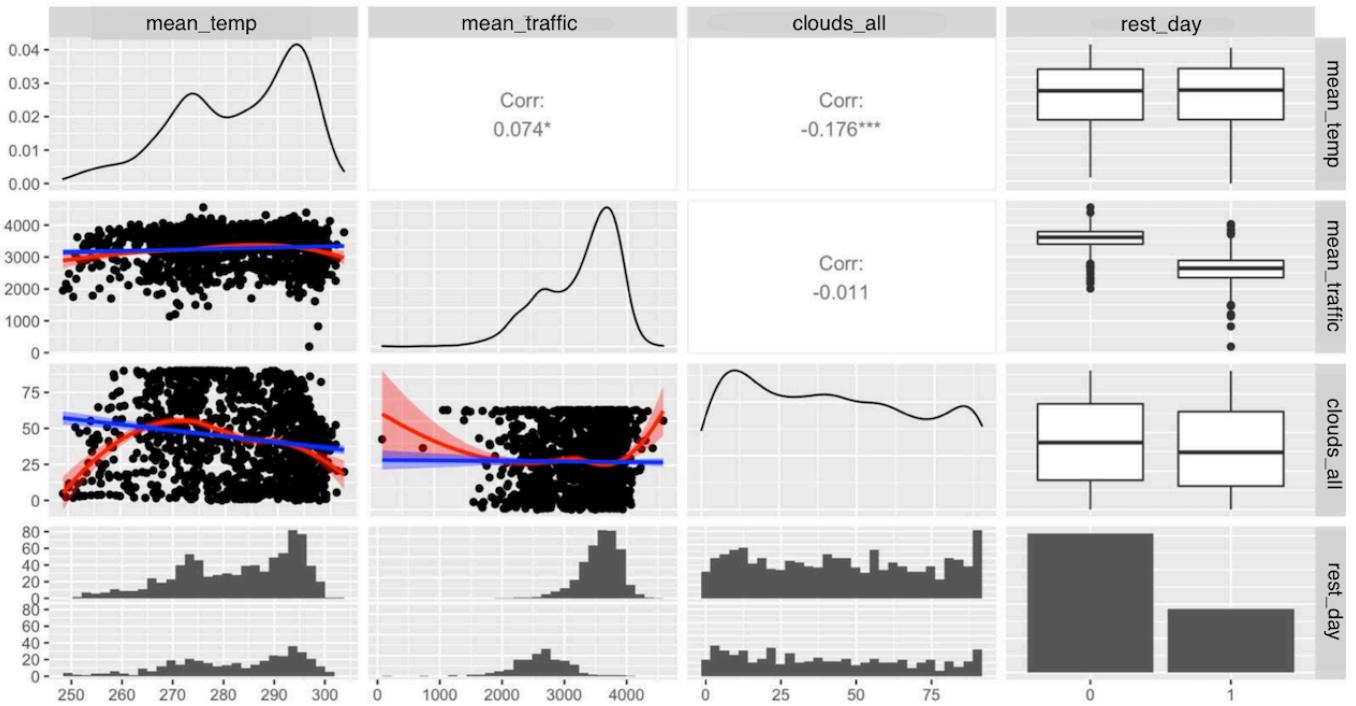
```

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 f
) %>%
select(-weekday)

rest_day <- dataset %>%
group_by(date_time)

```

ottenendo gli scatterplot e le correlazioni tra le variabili d'interesse:



Dal grafico notiamo subito che l'unica correlazione statisticamente significativa è quella tra mean_temp e clouds_all anche se con un valore basso pari a -0.176: all'aumentare delle temperature, che indicano l'arrivo della bella stagione, la percentuale di cielo coperto diminuisce. Il valore basso della correlazione è dovuto al fatto che i giorni estivi possono essere nuvolosi anche se mediamente saranno di meno rispetto alla stagione autunnale o invernale.

Le altre correlazioni, invece, sono molto basse con valori prossimi allo zero.

Per quanto riguarda rest_day, essendo questa una variabile factor, vengono riportati gli histogrammi e il box plot dalla quale possiamo notare che il traffico è maggiore nei giorni lavorativi e diminuisce nel week-end e giorni festivi.

Le linee blu nel grafico rappresentano la regressione lineare mentre quelle rosse sono loess. Nel caso di mean_temp e mean_traffic le linee si sovrappongono molto bene, come ci aspettavamo.

Analizzate le correlazioni possiamo procedere con i modelli di regressione. Identifichiamo tre modelli dove per semplicità chiamiamo T = mean_traffic; C = clouds_all; Te = mean_temp;

$$\left\{ \begin{array}{l} T = \alpha_0 + \alpha_1 C + \alpha_2 Te + \alpha_3 R + \varepsilon_t \\ T = \beta_0 + \beta_1 Te + \varepsilon_t \\ T = \gamma_0 + \gamma_1 Te + \gamma_2 R + \varepsilon_t \end{array} \right.$$

Confrontate le seguenti metriche:

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

Notiamo che il modello migliore è quello completo con valore minimo di AIC, AICc, BIC e CV e un adattamento ai dati pari al 59%.

Vediamo i parametri di questo modello attraverso il 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
```

Osserviamo che i residui assumono un valore minimo di -2437 e un max di 1429 e che la mediana in proporzione al residual std error ($30/366.8=0.1$) è vicina allo 0. Notiamo anche una leggera asimmetria in quanto il minimo è più elevato del massimo.

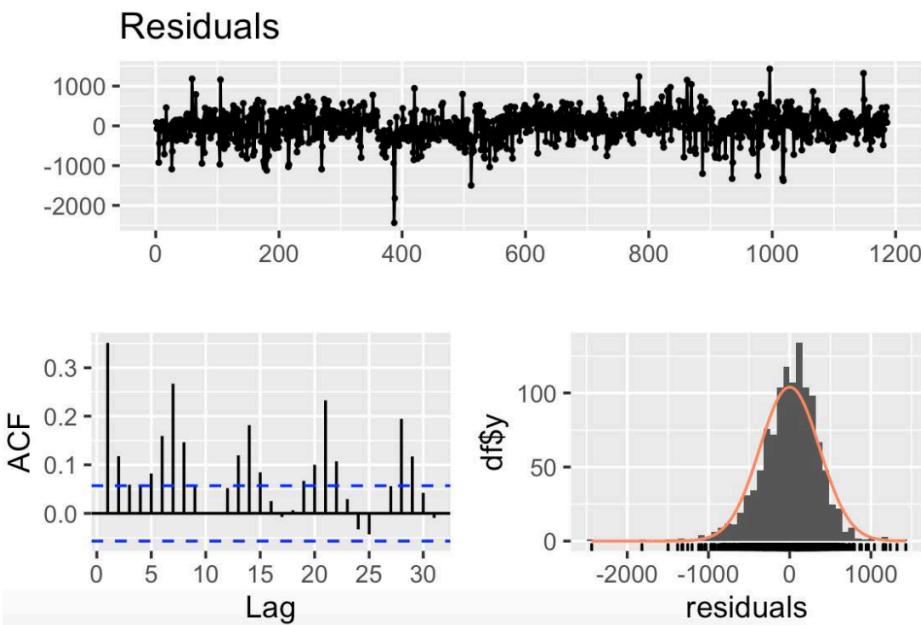
L'intercetta è circa 2747, quindi il volume medio del traffico quando le altre variabili sono nulle è 2747. Ovviamente non ha senso da un punto di vista interpretativo in quanto non è possibile avere una temperatura pari a 0 Kelvin.

La variabile perc_cloud ha un valore di -1.12 circa, il che significa che all'aumentare dell'1% di nuvolosità del cielo, il traffico diminuisce di 1.12, confermando quindi la correlazione negativa tra le due variabili e confermando che nel caso di giorni nuvolosi il traffico diminuisce.

Quando la variabile rest_day assume valore 1, cioè in caso di giorni non lavorativi, il traffico diminuisce mediamente di 983 circa. Questo conferma quanto già detto nel grafico delle correlazioni, e cioè che nei giorni non lavorativi il traffico diminuisce.

La variabile mean_temp ha un valore di 3.10, il che significa che all'aumentare di un 1 grado il traffico aumenta mediamente di 3.10. Ovviamente l'aumento della temperatura di un grado kelvin corrisponde all'aumento di un grado celsius perché la trasformazione è una semplice traslazione di 273,15.

Possiamo vedere la distribuzione dei residui nel seguente grafico:



Dal grafico notiamo che l'andamento dei residui sembra essere casuale attorno allo 0 e sembrano quindi dei WN con media 0 e varianza costante. La distribuzione, come si vede dall'istogramma, è praticamente normale anche se leggermente asimmetrica come detto prima. Dal grafico delle ACF notiamo però che i residui sono autocorrelati tra loro e quindi cade l'ipotesi di distribuzione White Noise. Pertanto, concludiamo che i residui si distribuiscono solo normalmente con media zero e varianza costante.

Modelli ARIMA e ANALISI DI BOX-JENKINS

Oltre al modello di regressione, decidiamo di stimare anche i modelli ARIMA, SARIMA e regARIMA e vediamo quale tra questi va a descrivere meglio l'andamento della serie storica di interesse.

Usiamo l'analisi di Box-Jenkins per andare a stimare questi modelli. La procedura si articola in tre passi:

- Identificazione del modello ottimo tramite un'analisi grafica della serie, la trasformazione dei dati se necessario e l'analisi della stazionarietà e della stagionalità;
- Stima dei parametri tramite la massima verosimiglianza;
- Model fitting per vedere quanto il modello si adatta bene ai dati.

I codici utilizzati in quest'ultima parte utilizzano funzioni riportate a parte, analizzate durante il corso di Serie Storiche che ho seguito presso Università degli Studi di Milano-Bicocca.

ARIMA

Applichiamo Box-Jenkins ai residui della serie traffic destagionalizzata e detrendizzata ottenuta nell'esplorativa:

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

Dall'output osserviamo che il modello ARIMA che meglio identifica l'andamento della nostra serie è un ARIMA(2,0,1), cioè con ordine della parte auto regressiva $p=2$, con ordine di differenziazione $d=0$, quindi stazionario, e con l'ordine della parte a media mobile $q=1$. Quindi il processo è influenzato dal suo passato fino al ritardo $t-2$ e da shock esogeni fino al ritardo $t-1$.

Rappresentiamo l'ARIMA(2,1) in forma estesa:

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

I parametri sembrano essere tutti statisticamente significativi e in particolare avremo che i parametri della parte autoregressiva hanno valore $\phi_1 = 1.2116$, $\phi_2 = -0.2320$ mentre quello della parte a media mobile ha valore $\theta_1 = -0.9124$, che essendo in modulo inferiore a 1 rende il processo invertibile.

SARIMA

Stimiamo anche un modello SARIMA, dato che la nostra serie presenta un andamento stagionale. Applichiamo sempre Box-Jenkins ma questa volta alla serie traffic originale (non destagionalizzata e detrendizzata) ottenendo:

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

Il SARIMA ottimale che va a catturare al meglio l'andamento della nostra serie è dato dalla moltiplicazione tra l'ARIMA(1,0,0) e il modello ARIMA(2,0,0) che cattura la stagionalità della serie.

Il SARIMA ottenuto va quindi a rappresentare un modello di previsioni temporali che tiene conto di un termine autoregressivo di ordine 1 e nessun termine di media mobile stagionale, e una stagionalità pari a 7. Rappresentiamolo in forma compatta:

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

I coefficienti del modello sono $\psi_1 = 0.4723$, $\psi_2 = 0.3805$, $\phi_1 = 0.2867$ tutti significativi.

regARIMA

L'ultimo modello stimato con Box_jenkins è il regARIMA con variabile di risposta la serie originale 'traffic', e come regressori le serie originali 'temp', 'clouds_all', 'rest_day':

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

Ciò che otteniamo è un modello di regressione lineare multipla con errori che si distribuiscono come ARIMA (1,1,2), cioè con parte autoregressiva di ordine 1, con distribuzione non stazionaria in quanto d=1 e con ordine della parte a media mobile pari a 2.

Possiamo scrivere la regressione con errori ARIMA come una regressione lineare con errori ARMA applicando la differenza d-esima ($d=1$) ad ogni variabile nel seguente modo:

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

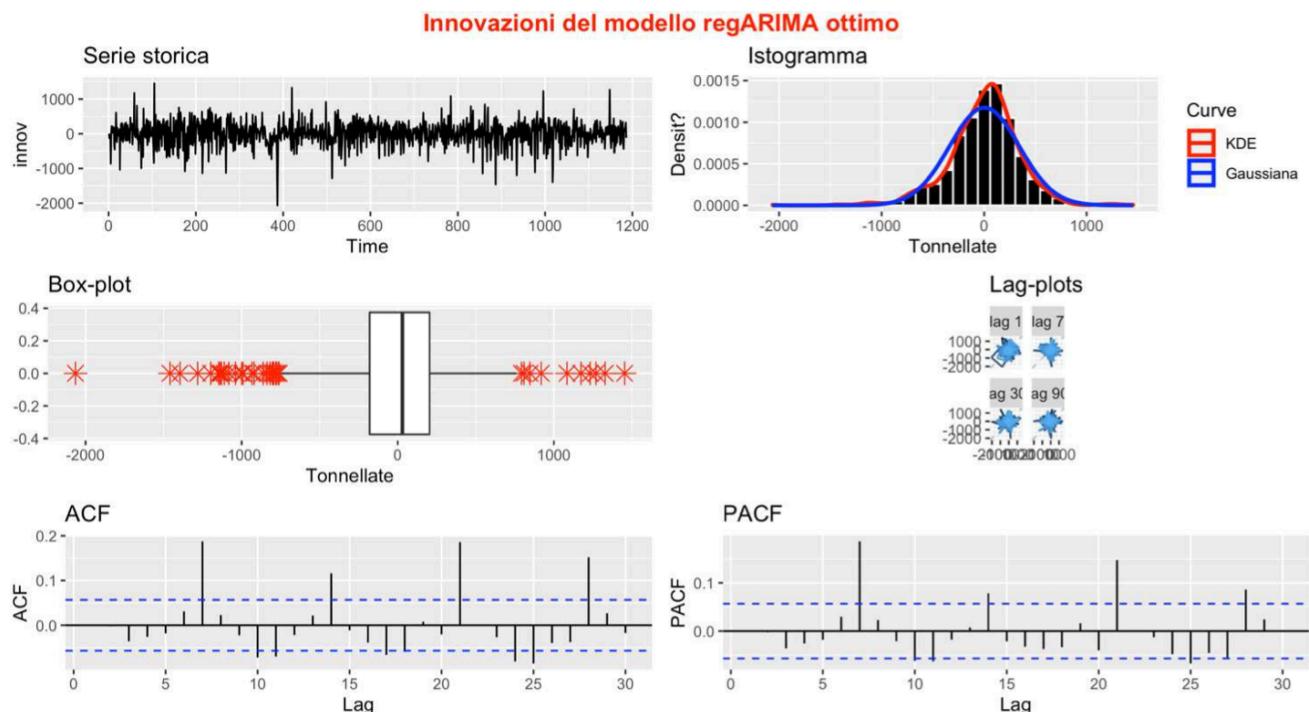
Con $\alpha_1 = 2.3301$, $\alpha_2 = -1.0706$, $\alpha_3 = -925.5439$ significativi e molto simili a quelli ottenuti nel modello di regressione completo ma non uguali in quanto stimati con MLE, e i coefficienti AR e MA con valori rispettivamente di $\psi_1 = 0.137$, $\theta_1 = -0.7938$, $\theta_2 = -0.1633$ tutti significativi.

Una volta stimati tutti i modelli vediamo tramite i criteri di ottimizzazione quale tra ARIMA, SARIMA, regARIMA e modello di regressione completo risulta il modello migliore:

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

Il modello che massimizza tutte le metriche e ha un adattamento migliore ai dati è il regARIMA.

Ora procediamo con l'analisi delle innovazioni per vedere l'adattabilità del modello ai dati:

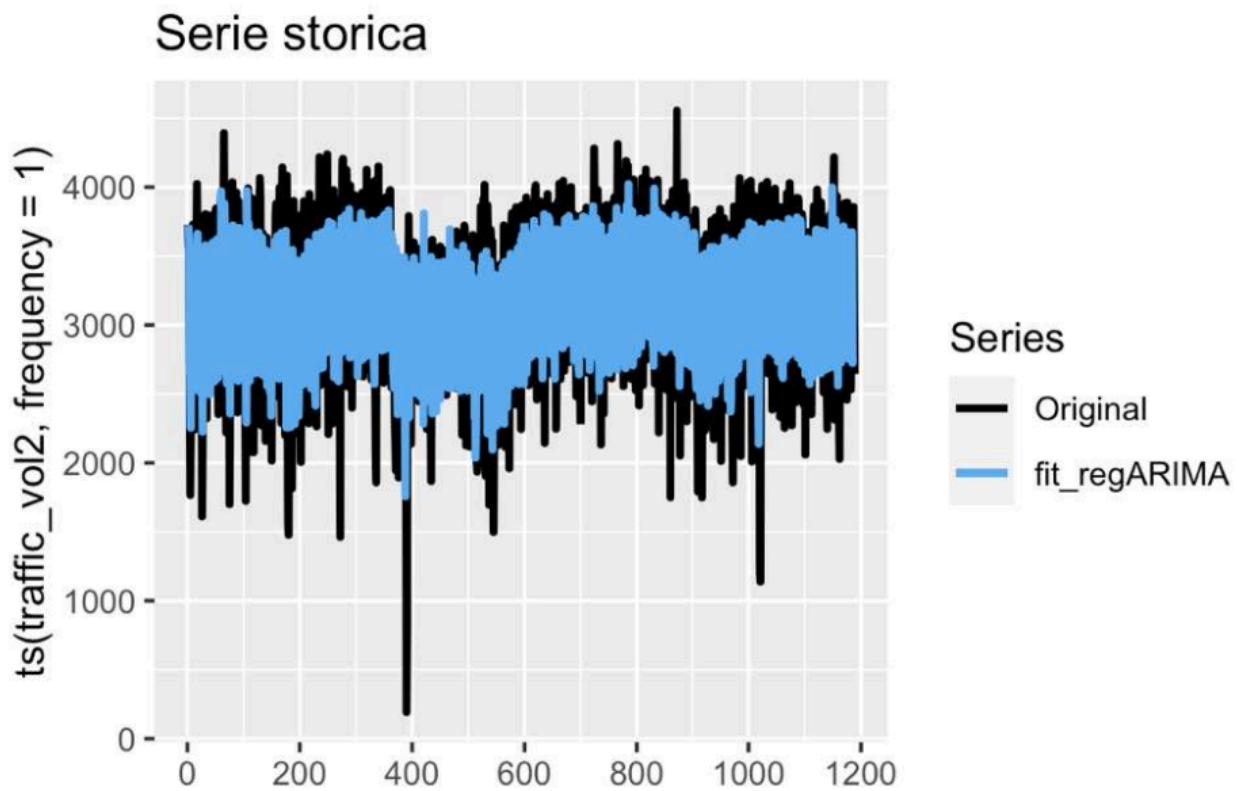


Vengono riportati diversi grafici, tra cui l'andamento dei residui, l'istogramma, il Box-plot, lag plot e i grafici delle autocorrelazioni. Dal primo grafico notiamo che i residui hanno un andamento che sembra essere casuale attorno allo 0 e questo ci porta a pensare che potrebbero distribuirsi come dei WN a media 0 e varianza costante. Dall'istogramma notiamo che la distribuzione dei residui è normale, simmetrica attorno alla media pari a 0 e questo è confermato anche nel box-plot.

Dal grafico delle ACF e PACF osserviamo che i residui sono incorrelati fra loro, e questo possiamo anche notarlo nei lag plots che rappresentano i valori congiunti delle variabili ritardate a coppie.

Pertanto possiamo concludere che i residui sono white noise.

Calcoliamo ora i valori fittati del modello e andiamo a confrontarli con la serie originale:



Il modello sembra descrivere molto bene l'andamento della serie originale andando a cogliere anche i picchi inferiori e superiori. Questo è confermato anche dal valore del R² pari al 66% circa, ciò significa che il regARIMA spiega il 66% della varianza totale, che è buono.

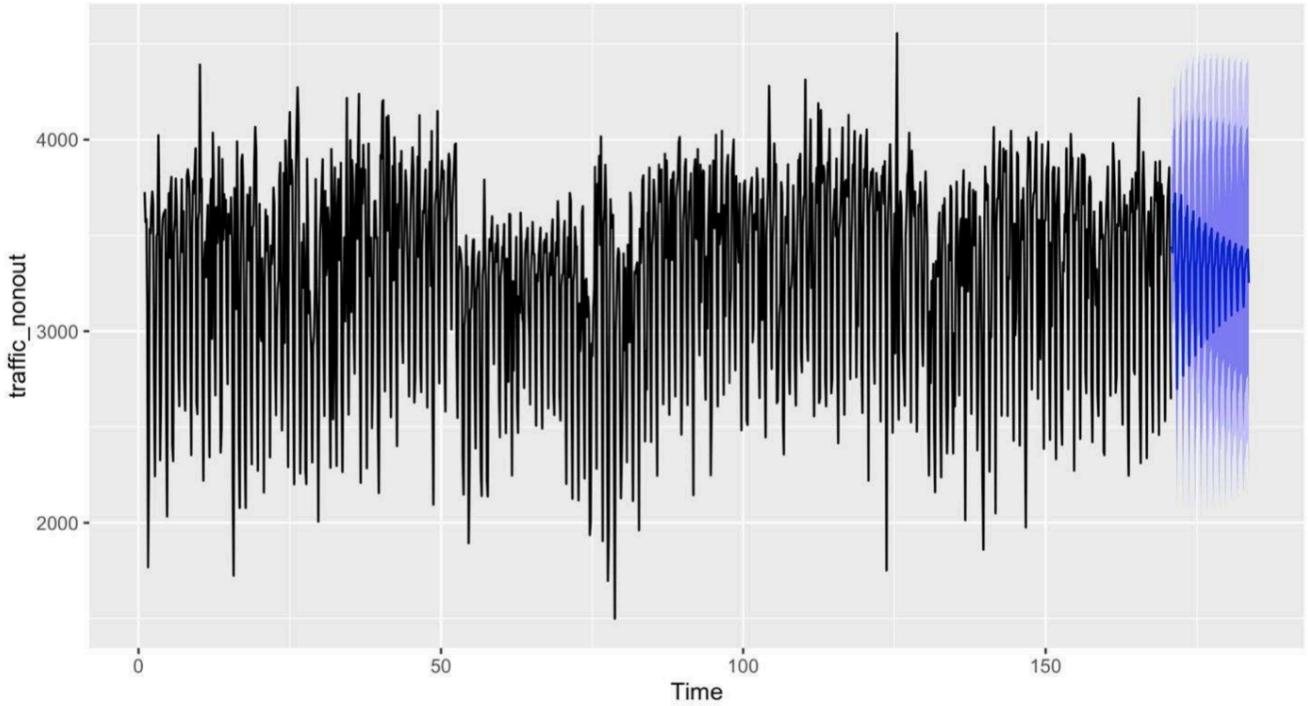
PREVISIONI

Come ultimo passo, andiamo a vedere le previsioni a breve termine dei valori futuri che vengono fatte dal modello migliore scelto al punto precedente.

Poiché il modello migliore è il regARIMA alla quale non possiamo applicare la funzione `forecast::forecast` di R in quanto necessita di un dataset di train e uno di test sulla quale fare le previsioni, scegliamo il secondo modello migliore, il SARIMA.

Ciò che otteniamo sono le previsioni relative ai tre mesi successivi dalla fine dei nostri dati:

Forecasts from ARIMA(1,0,0)(2,0,0)[7] with non-zero mean



Osserviamo dal grafico che le linee blu sono le previsioni, mentre le linee viola attorno rappresentano l'intervallo di confidenza al 95%, per quello più esterno, e all'80% per quello più interno. Notiamo che l'intervallo di confidenza diventa sempre più ampio all'aumentare del tempo e questo è dovuto al fatto che più si va avanti nel tempo più i valori previsti sono incerti e meno precisi. Nonostante ciò, possiamo osservare come il SARIMA preveda per i giorni e le settimane più vicine un andamento simile a quello presente nella serie originale, mentre mano a mano che si va avanti nel tempo sembra che il valore previsto per il volume del traffico diminuisca sempre di più, e questo è probabilmente dovuto a quanto detto prima, ovvero che più si va avanti nel tempo, più le previsioni sono imprecise e tendono ad assumere un valore prossimo alla media.

Conclusioni

Al fine di rispondere ai quesiti che ci siamo posti all'inizio di questo elaborato abbiamo svolto diverse analisi. Possiamo concludere che i risultati ottenuti da queste analisi sono molto buoni.

Questo vale in particolare per l'analisi esplorativa che ci ha permesso sin da subito di rispondere alle domande che ci siamo posti. Anche i modelli stimati ARIMA, SARIMA, regARIMA e la regressione ci hanno fornito dei risultati ottimali, infatti avevano tutti un adattamento di circa il 60% ai dati che possiamo ritenere buono. L'unica tecnica che non ci ha fornito risultati soddisfacenti è stato il modello di regressione con il traffico in funzione del tempo e della variabile rest_day, dalla quale ci aspettavamo un adattamento ai dati molto buono e non dello 0.5% anche perché il traffico sembrava essere molto influenzato sia dalla temperatura che dal tipo di giorno della settimana, se lavorativo o no. Ciò nonostante, siamo comunque riuscite a rispondere in maniera ottimale alle domande di ricerca. Abbiamo infatti visto come il traffico sia influenzato dal brutto tempo in quanto all'aumentare della nuvolosità e all'abbassarsi della temperatura questo diminuisce. Abbiamo poi notato come nei giorni lavorativi il traffico sia maggiore che nei week-end o nei giorni festivi e questo è dovuto a diversi fattori: al pendolarismo, cioè le persone in settimana sono obbligate a spostarsi per raggiungere il luogo di lavoro; alla scuola o alle attività sportive, ma anche al traffico prodotto dalle attività commerciali che spediscono i loro prodotti oppure vengono rifornite.

Si è anche notato come tra i giorni feriali, quello che maggior traffico è il venerdì e questo probabilmente può essere dovuto a tutte quelle persone che lavorano in settimana in altre città e nel week-end tornano a casa.

Abbiamo poi confermato il fatto che se nel week-end è bel tempo il traffico aumenta e questo perché ci si tende a spostare maggiormente magari per gite fuori porta.

Infine, abbiamo anche notato come il modello SARIMA preveda abbastanza bene l'andamento del traffico nei giorni successivi alla fine del nostro dataset, anche se i valori sono molto incerti e poco attendibili soprattutto nel lungo periodo in quanto non possiamo prevedere i fenomeni atmosferici, soprattutto in questo periodo di crisi climatica, che potrebbero influire sul traffico stradale.