Analisi evoluzione livello co2 Atmosferica.

Mariangela Tafuri, Vincenzo Picarelli, Paolo Simari

Giugno 2022

Scopo di questo report è quello di esplorare e analizzare l'evoluzione dei livelli co2 nell'aria dal 1958 fino al 2022, e come questo protrebbe evolversi nel tempo.

Parole chiave: *analisi dati, serie storica, co2, Arima.*

Il caso studio

L'obiettivo del report è quello di analizzare le dinamiche relative alle emissioni di CO2 nell'atmosfera terrestre, importante indicatore dell'inquinamento atmosferico. Per fare ciò utilizzeremo i dati raccolti dal Global Monitoring Laboratory dell'Earth System Research Laboratory (ESRL), https://gml.noaa.gov/ccgg/trends/data.html. I dati mostrano l'anidride carbonica media mensile e sono disponibili sino a maggio 2022. In particolare, ogni media mensile è la media delle medie giornaliere, basate a loro volta su medie orarie. Ogni record è riportato come frazione molecolare dell'aria, definita come il numero di molecole di anidride carbonica divise per il numero di tutte le molecole nell'aria, inclusa la stessa CO2. La frazione molecolare è espressa come parti per milione (ppm).

Analisi preliminare

Caricamento delle librerie e del dataset:

```r
library(timeSeries)
library(timeDate)
library(fBasics)
library(forecast)
library(ggplot2)
library(imputeTS)
library(urca)
library(tidyverse)
library(seasonal)
library(lubridate)
library(fpp)
library(seastests)
library(lmtest)
library(stats)

co2<-read.csv('co2.csv')
```

Si verifica se il dataset presenta dei *valori* mancanti

```r
sum(is.na(co2))
```

```
## [1] 0
```

```r
dim(co2)
```

```
## [1] 771   9
```

Il dataset contiene 771 record per 9 variabili, che sono:

```r
names(co2)
```

```
## [1] "Year"             "month"            "decimal_date"    "monthly_average"
## [5] "de.seasonalized" "days"             "st.devofday"     "mean_ofmon"
## [9] "data"
```

Tuttavia, ai fini delle analisi, si utilizzeranno solo alcune delle variabili presenti nel dataset, tra cui:

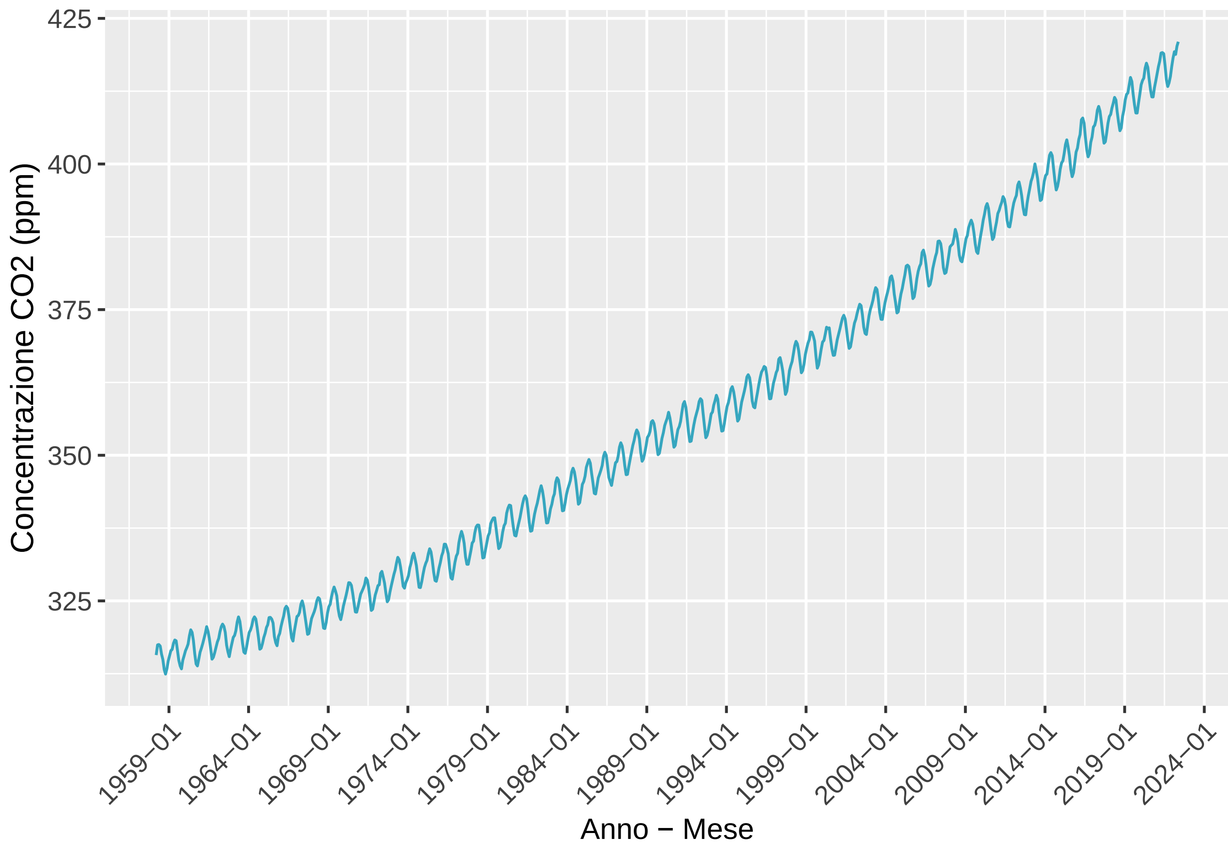
```r
head(co2)
```

```
##   Year month monthly_average       data
## 1 1958     3          315.70 1958-03-15
## 2 1958     4          317.45 1958-04-15
## 3 1958     5          317.51 1958-05-15
## 4 1958     6          317.24 1958-06-15
## 5 1958     7          315.86 1958-07-15
## 6 1958     8          314.93 1958-08-15
```

Come è possibile osserva la serie temporale risulta essere a frequenza mensile.

Visualizzazione

Prima di tutto si procede a visualizzare la serie storica, per osservare se vi è la presenza di qualche componente (evolutiva, stagionale o ciclica) che caratterizza la serie in esame.




```r
# Divido Train e Test
co2_test <- d %>% filter(Year > 2016)
co2_train <- d %>% filter(Year <= 2016)
```

```r
##
co2 <- ts(d$monthly_average, frequency = 12, start = c(1958,3))
co_train <- ts(co2_train$monthly_average, frequency = 12, start = c(1958,3))
co_test <- ts(co2_test$monthly_average, frequency = 12, start = c(2017,1))
```

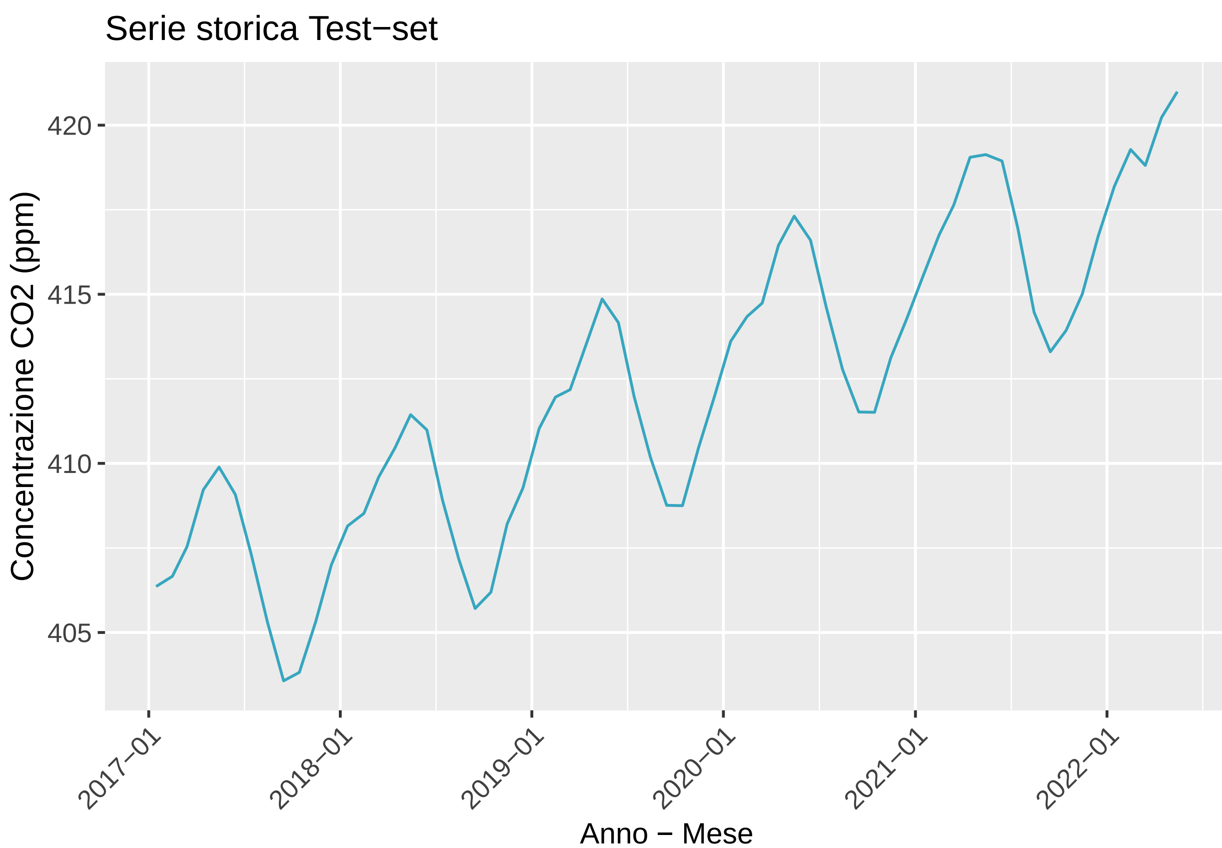
```r
head(co_train[])
```

```
##          Mar    Apr    May    Jun    Jul    Aug
## 1958 315.70 317.45 317.51 317.24 315.86 314.93
```

```r
## Rappresentazione Test e Train

ggplot(co2_train,aes(data, monthly_average)) +
  geom_line(color='#36A6BF') +
  xlab("Anno - Mese") +
  scale_x_date(date_labels = "%Y-%m", date_breaks = "5 year") +
  theme(axis.text.x = element_text( color = "#404040",
                                   size = 10, angle = 45, hjust = 1)) +
  ylab("Concentrazione CO2 (ppm)") +
  #scale_x_continuous(breaks = trans_breaks(identity, identity, n = 10))
```

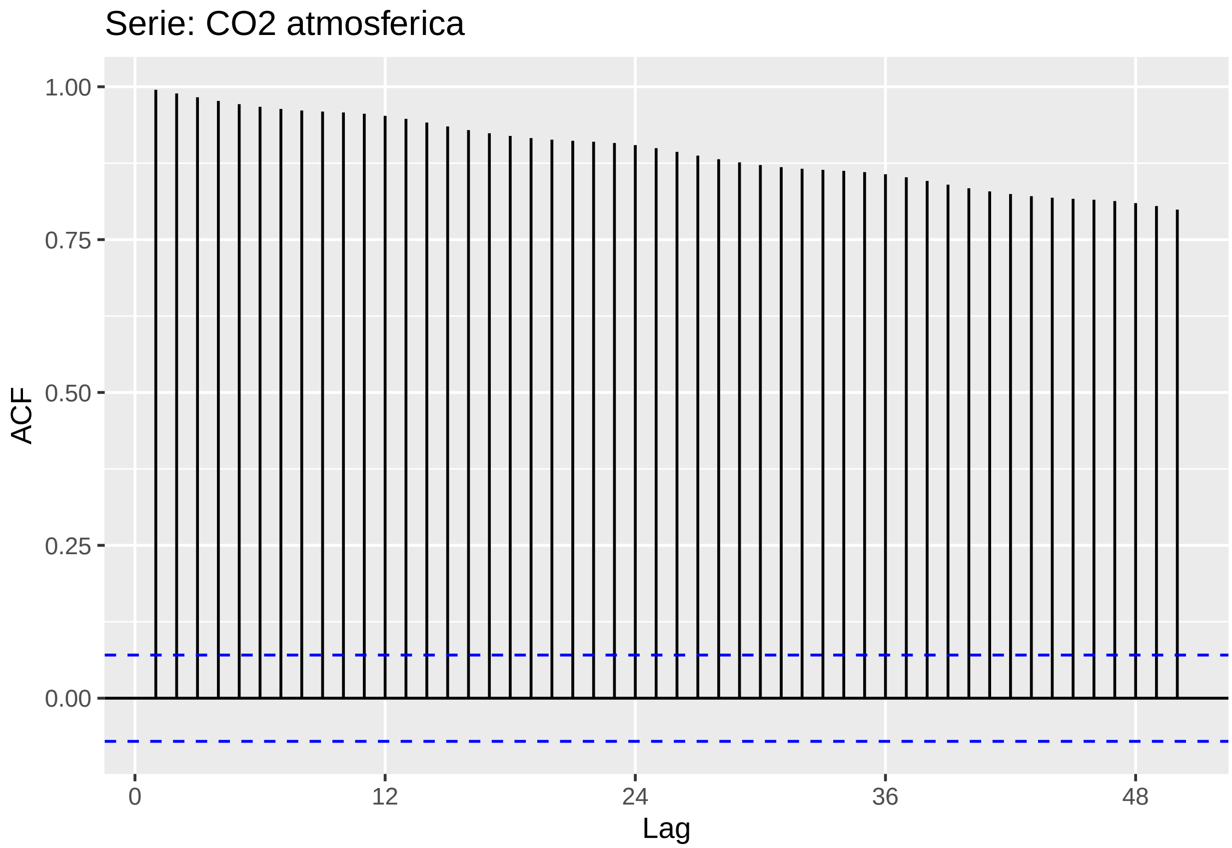
```
  scale_y_continuous() +
  theme(axis.text.y = element_text(color = "#404040",
                                   size = 10, hjust = 1), axis.title.y = element_text(size = 12)) +
  ggtitle("Serie storica Train-set")
```


```
ggplot(co2_test,aes(data, monthly_average)) +
  geom_line(color='#36A6BF') +
  xlab("Anno - Mese") +
  scale_x_date(date_labels = "%Y-%m", date_breaks = "1 year") +
  theme(axis.text.x = element_text(color = "#404040",
                                   size = 10, angle = 45, hjust = 1)) +
  ylab("Concentrazione CO2 (ppm)") +
  #scale_x_continuous(breaks = trans_breaks(identity, identity, n = 10))
  scale_y_continuous() +
  theme(axis.text.y = element_text(color = "#404040", size = 10, hjust = 1),
        axis.title.y = element_text(size = 12)) +
  ggtitle("Serie storica Test-set")
```

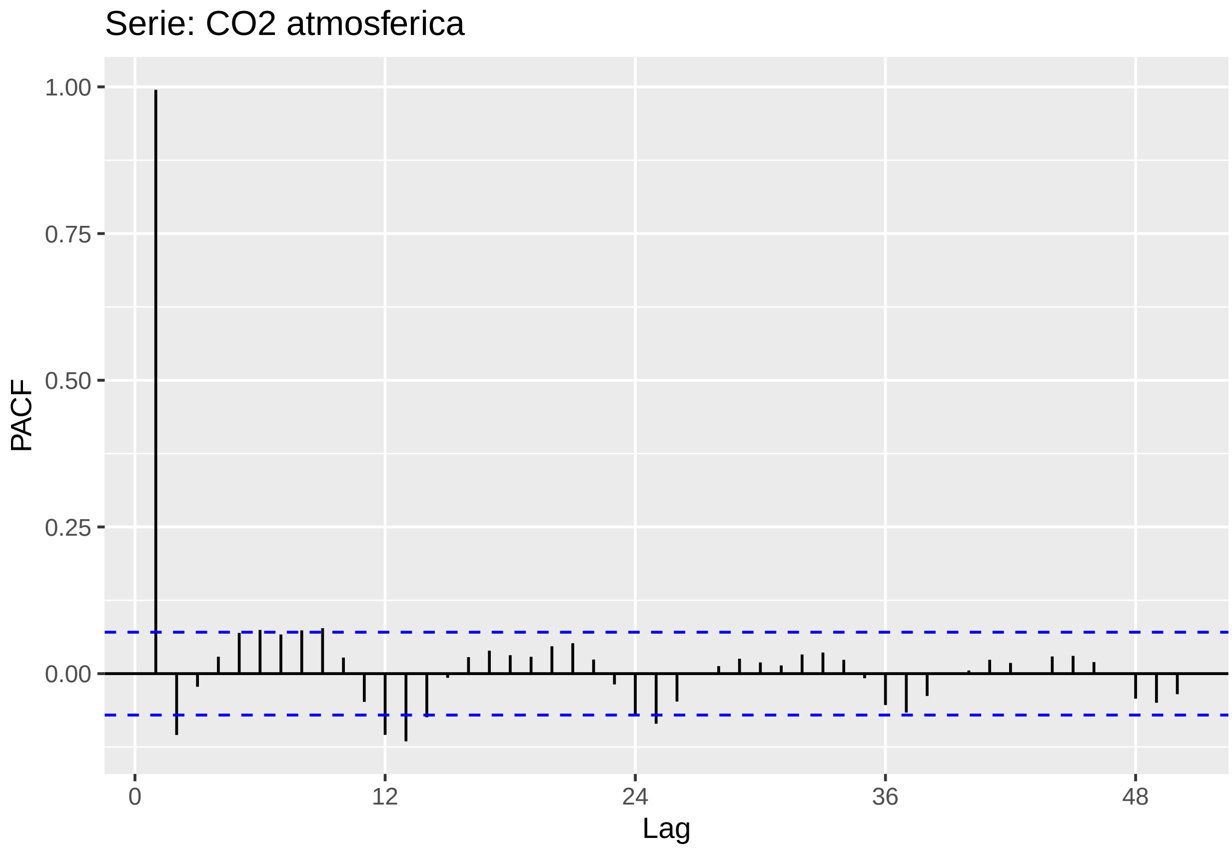


Funzione di autocorrelazione.

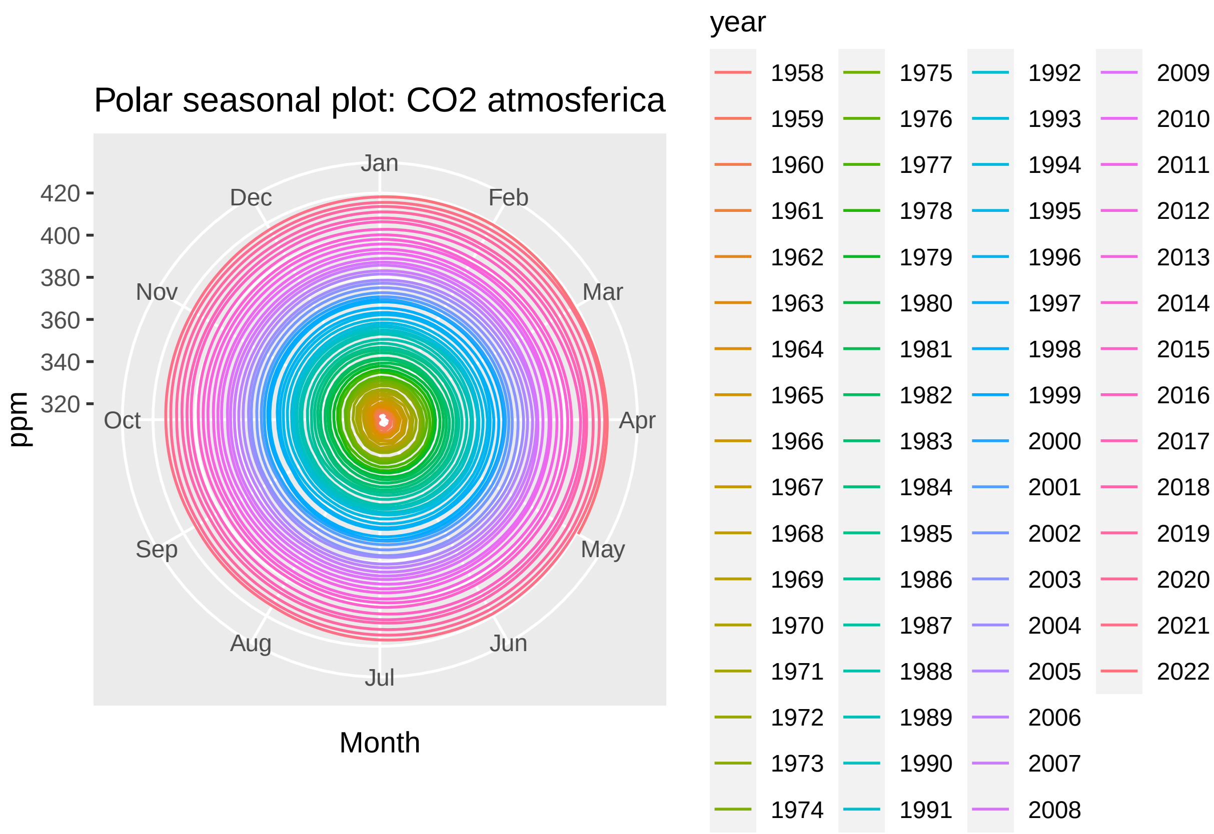
```
#### PACF E ACF su tutta la serie.
ggAcf(co2, lag.max = 50)+
  ggtitle("Serie: CO2 atmosferica ")
```

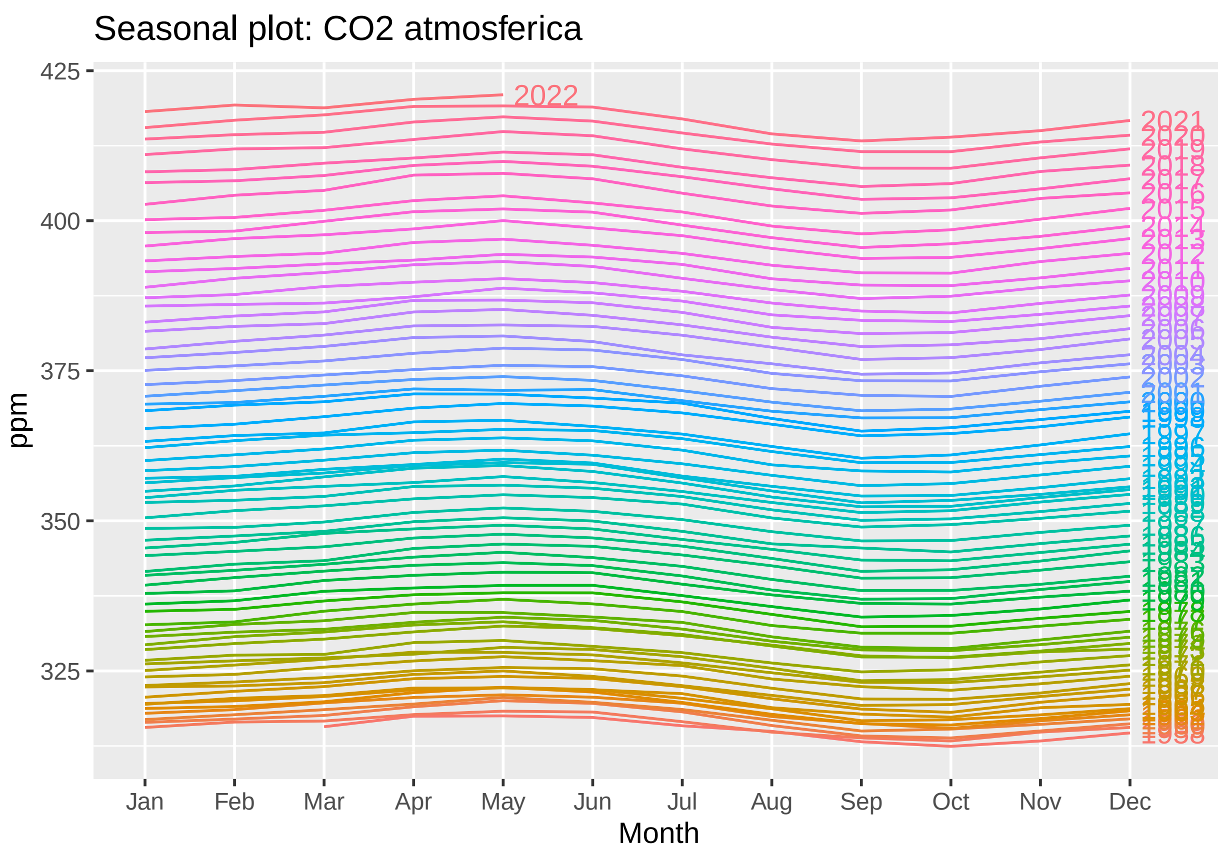


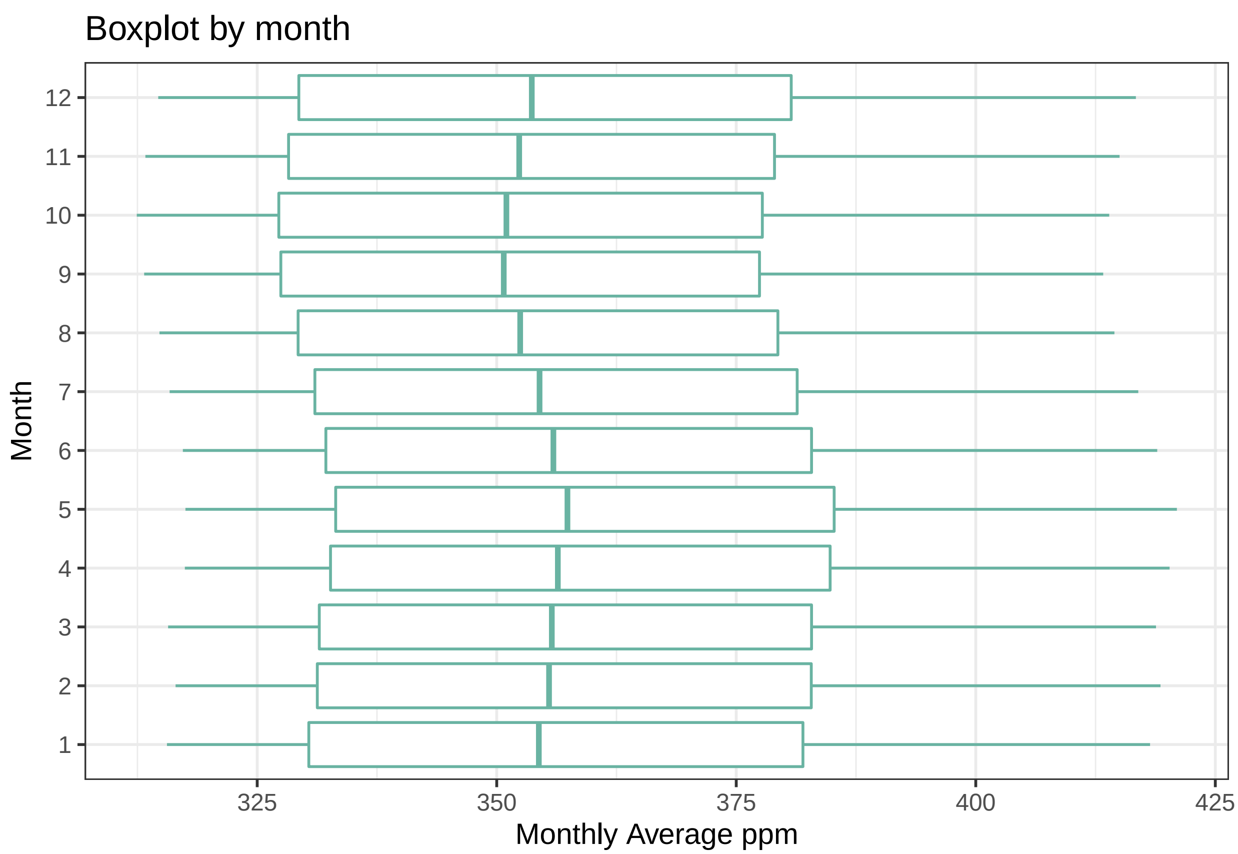

```r
ggPacf(co2, lag.max = 50)+
  ggtitle("Serie: CO2 atmosferica")
```



Come è possibile notare la serie presenta una forte consistenza nella correlazione. Questo è un indicatore del fattore che la serie presenta un trend.


Polar seasonal plot: CO2 atmosferica
ppm
420
400
380
360
340
320
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month
year
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022


Seasonal plot: CO2 atmosferica
ppm
425
400
375
350
325
2022
2021
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month



Confermiamo, attraverso i test, se vi è la presenza di un trend.

```
##
##  Augmented Dickey-Fuller Test
##
## data:  co_train
## Dickey-Fuller = -0.44415, Lag order = 8, p-value = 0.9844
## alternative hypothesis: stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  co_train
## Dickey-Fuller = -0.44415, Lag order = 8, p-value = 0.01564
## alternative hypothesis: explosive
```

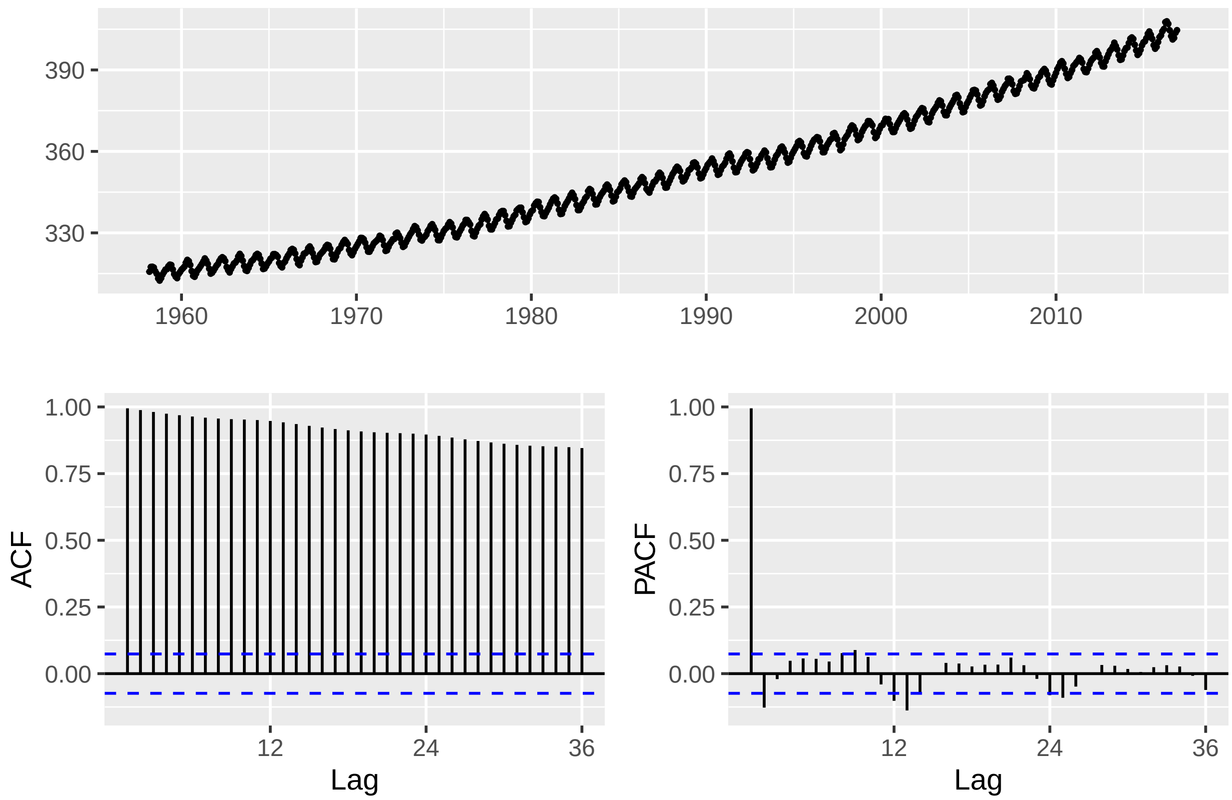
Test su stagionalità

```
# Test di Webel-Olleck per la stagionalità
isSeasonal(co_train, freq=12)
```

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

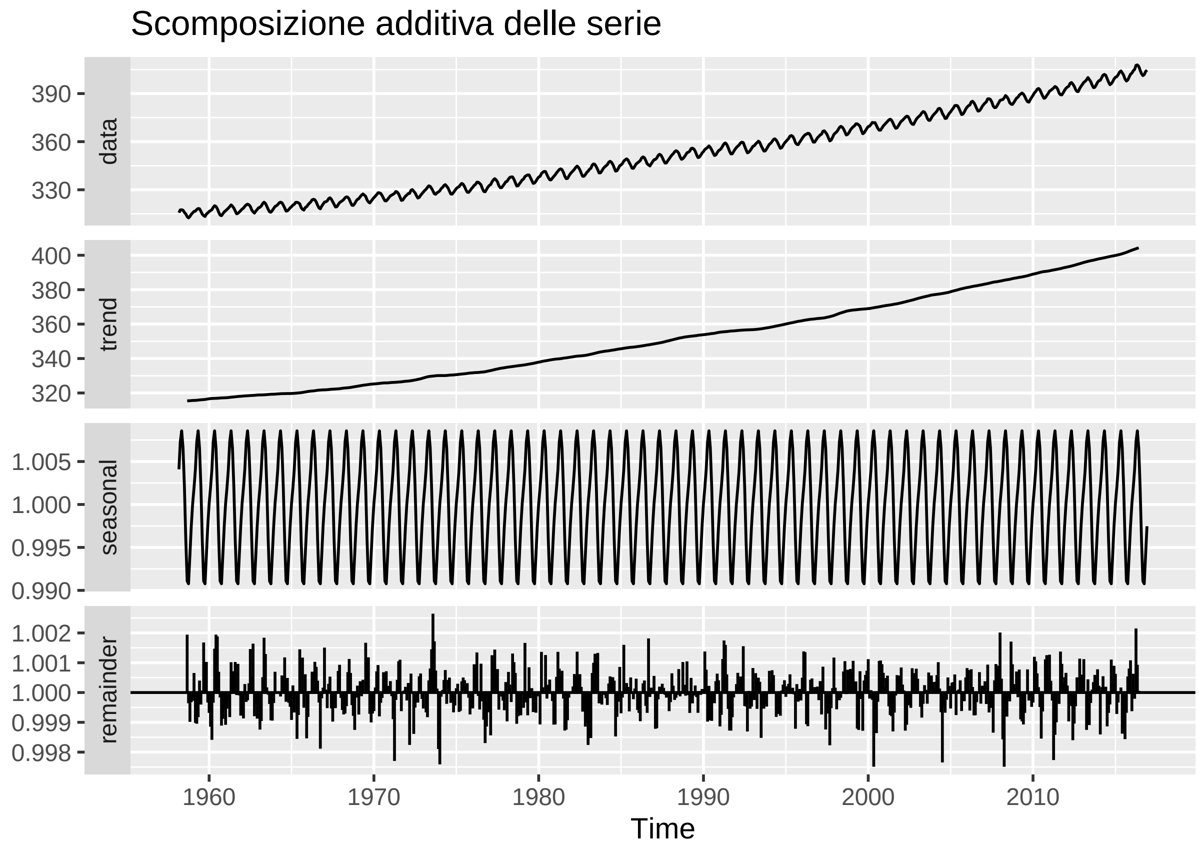
A riprova del fatto che vi è la presenza di una componente stagionale che caratterizza la serie. Graficamente viene ripresentata la funzione di autocorrelazione parziale e totale sul train set.

```r
# grafico ACF e PACF su train set
co_train %>% ggtsdisplay()
```

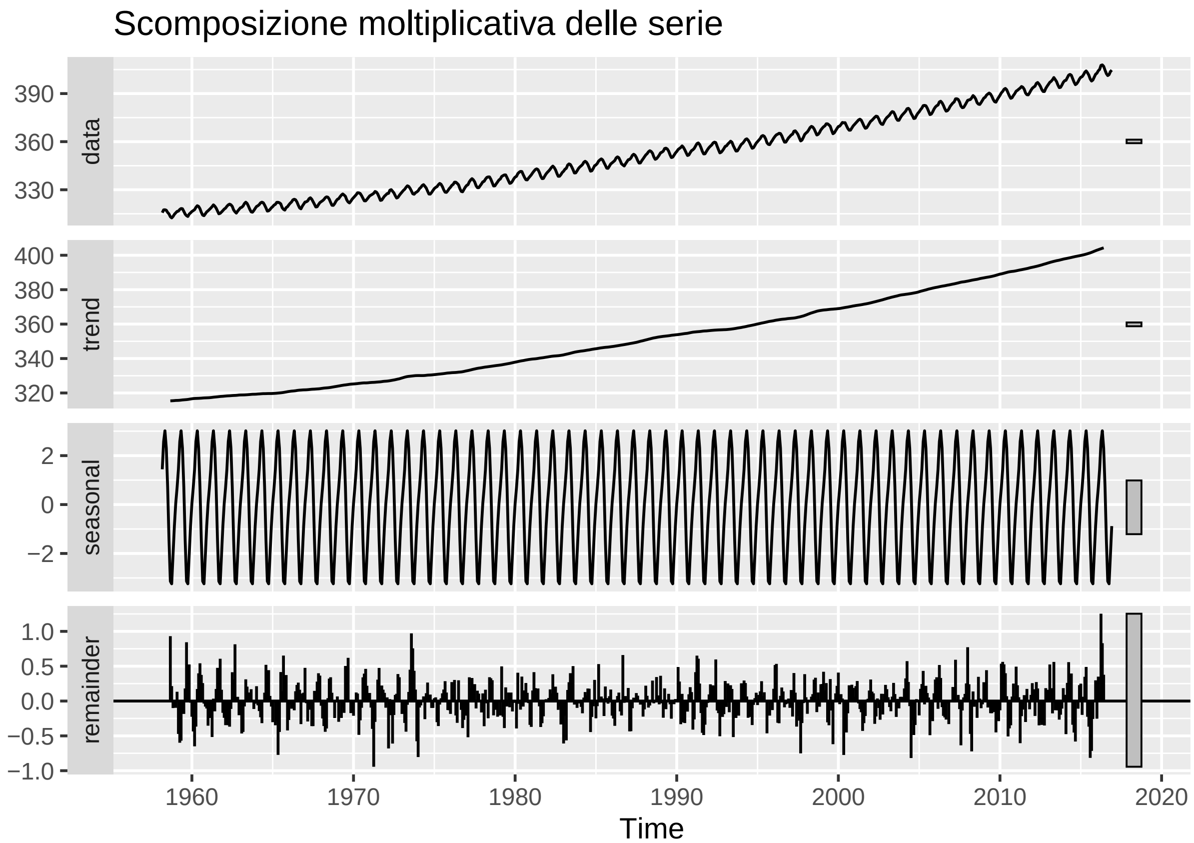


SCOMPOSIZIONE CLASSICA.

```r
mydata=decompose(co_train, "multiplicative")
mydata1=decompose(co_train, "additive")
autoplot(mydata, main="Scomposizione additiva delle serie")
```

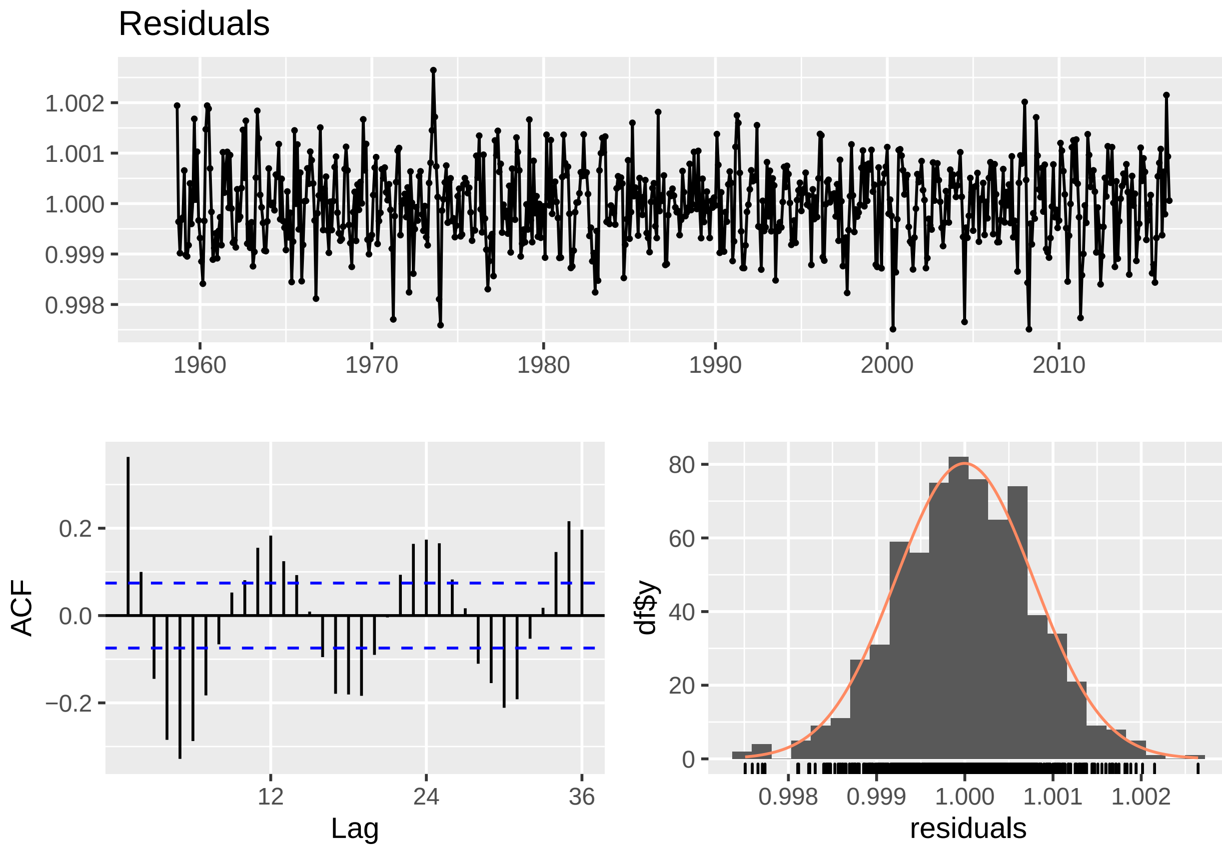



```
autoplot(mydata1, main="Scomposizione moltiplicativa delle serie")
```



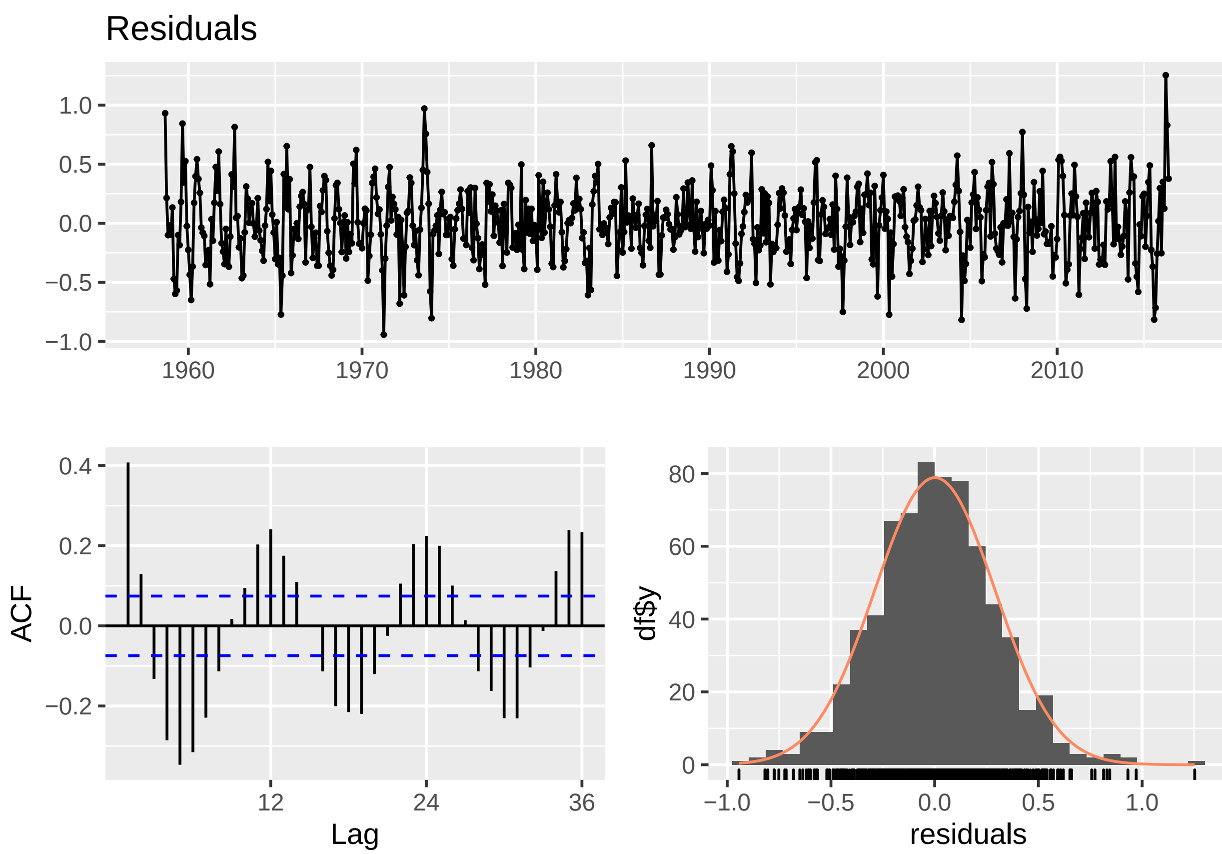
Analisi residui modello additivo.

```
checkresiduals(remainder(mydata))
```



Analisi residui modello moltiplicativo.

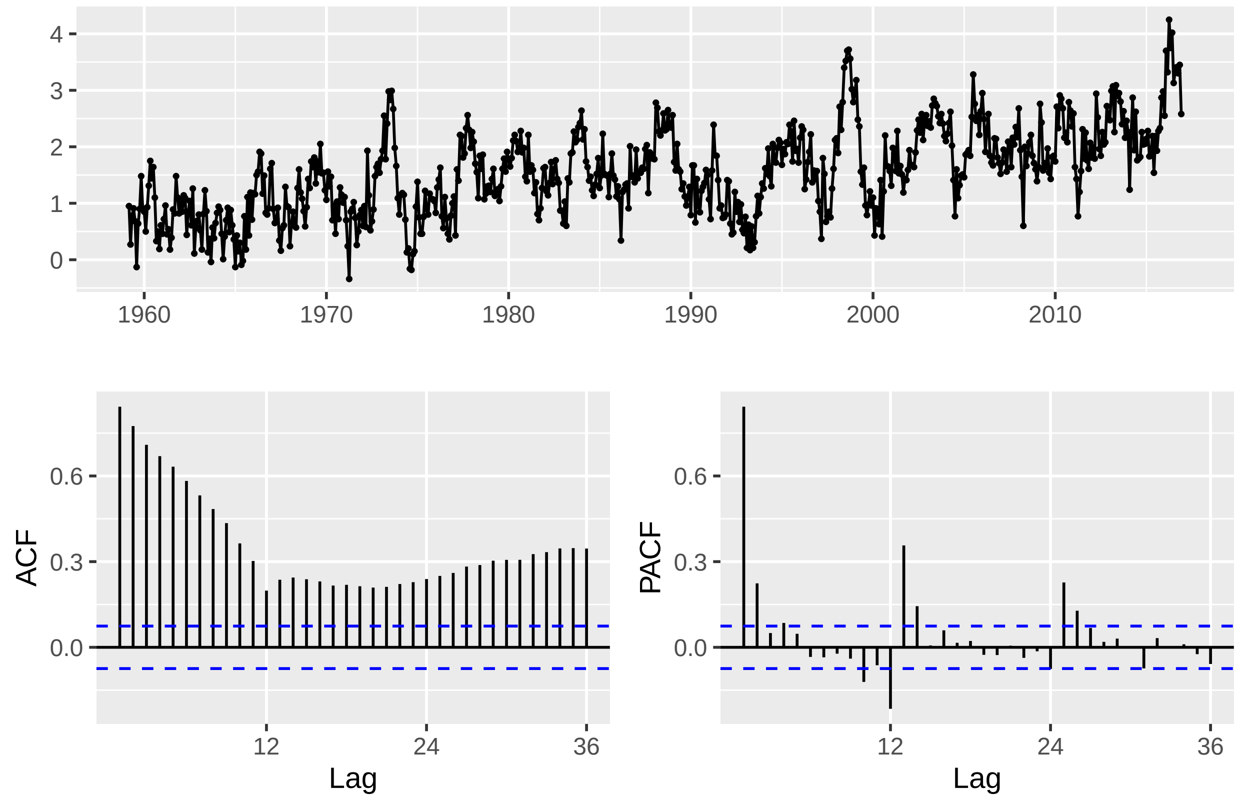
```
checkresiduals(remainder(mydata1))
```



Come emerge chiaramente dal correlogramma, la forte struttura di autocorrelazione presente all'interno della componente residua è sintomo dell'incapacità del modello additivo di catturare correttamente la componente di tendenza ciclica e la marcata stagionalità nei dati. Più in generale, in questo caso, l'approccio classico alla decomposizione non restituisce risultati soddisfacenti. Questo avviene anche applicando un modello moltiplicativo, cioè anche applicando tale modello, non siamo in grado di risolvere tutte le criticità precedentemente descritte.

Differenzazione. In presenza di *stagionalità* è necessario procedere a differenziare per il periodo di stagionalità, individuato in precedenza e pari a 12.

```
X_1= diff(co_train, 12)
ggtsdisplay(X_1)
```



E' possibile già intuire visivamente come la serie, a seguito della differenziazione, non risulta essere stazionaria, e cio viene confermato anche dal test che segue:

```
#Confermato anche dal test.
diff(co_train, 12) %>% ur.kpss() %>% summary
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 4.6527
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Il test di radice unitaria restituisce un p-value alto, per cui questo ci porta a non rigettare l'ipotesi nulla di non stazionarieta della serie. A seguito della differenziazione stagionale la serie non è ancora stazionaria, per cui è necessario effettuare un'ulteriore differenziazione.

```
d=ndiffs(co_train, test = "kpss")
d
```

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

Suggerisce d=1, ovvero procedere a differenziare per un periodo.

```
X = diff(X_1, diff=d)
diff(diff(co_train, 12),1) %>% ur.kpss() %>% summary
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 0.0081
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

In seguito ad un'ulteriore differenziazione, il test ci porta a rigettare l'ipotesi nulla di non stazionarietà della serie. Questo ci permette di affermare con un buon grado di fiducia che i dati differenziati risultano adesso stazionari. Viene confermato anche attraverso l'adf test.

```
adf.test(X, alternative = 'stationary') # Rifiuto H_0, serie stazionaria.
```

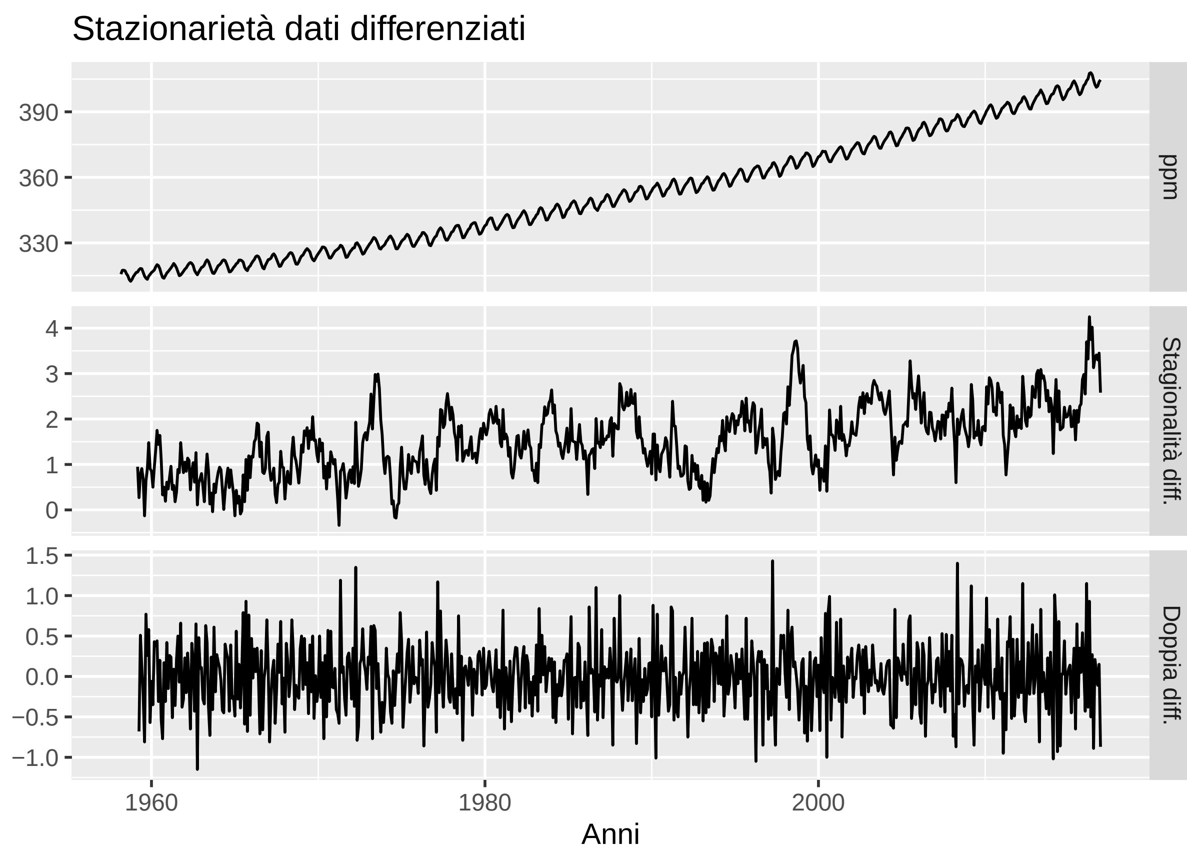
```
##
##  Augmented Dickey-Fuller Test
##
## data:  X
## Dickey-Fuller = -8.8063, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(X, alternative = 'explosive') #Non rifiuto H_0, serie non esposiva.
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  X
## Dickey-Fuller = -8.8063, Lag order = 8, p-value = 0.99
## alternative hypothesis: explosive
```

Vediamo l'andamento grafico del processo di differenziazione.

```
cbind("ppm" = co_train,
      "Stagionalità diff." = X_1 ,
      "Doppia diff." =  X ) %>%
  autoplot(facets=TRUE) +
  xlab("Anni") + ylab("") +
  ggtitle("Stazionarietà dati differenziati")
```



Determinazione del modello

Per curiosità inseriamo il modello consigliato in automatico dall'algoritmo auto.arima!

```
auto.arima(co_train, d = 1, D = 1,stationary = FALSE ,
           seasonal = TRUE,
           ic = "aicc", stepwise = TRUE,
           trace = FALSE,
           truncate = NULL, xreg = NULL,
           test = 'kpss',
           seasonal.test = 'seas' )
```

```
## Series: co_train
## ARIMA(1,1,2)(2,1,2)[12]
##
## Coefficients:
##          ar1      ma1     ma2     sar1     sar2     sma1     sma2
##       0.4809  -0.8280  0.1083  -0.2995  -0.0107  -0.5581  -0.2646
## s.e.  0.2517   0.2544  0.1099   2.5205   0.0718   2.5206   2.1904
##
## sigma^2 = 0.09857:  log likelihood = -178.67
## AIC=373.34   AICc=373.55   BIC=409.67
```

Inseriamo il modello consigliato dall'algoritmo e valutiamo diversi modelli in relazione a quello che ci viene indicato dal PACF e ACF. Proviamo diversi modelli e decidiamo sulla base dell'AIC (AIC Più Basso)

```r
fit1 <- Arima(co_train, order = c(0,1,1), seasonal = c(0,1,1))
fit2 <- Arima(co_train, order = c(1,1,1), seasonal = c(0,1,1)) # modello miglior in sample.
fit3 <- Arima(co_train, order = c(1,1,1), seasonal = c(1,1,1))
fit4 <- Arima(co_train, order = c(0,1,1), seasonal = c(2,1,1))
fit5 <- Arima(co_train, order = c(1,1,1), seasonal = c(2,1,1))
fit6 <- Arima(co_train, order = c(3,1,0), seasonal = c(2,1,0))
fit7 <- Arima(co_train, order = c(3,1,1), seasonal = c(2,1,1))
fit8 <- Arima(co_train, order = c(3,1,1), seasonal = c(2,1,1))
fit_auto <- Arima(co_train, order = c(1,1,2), seasonal = c(2,1,2))
```

```r
model_eva
```

```
##   Model_name      AICc
## 1       fit1 -1667.470
## 2       fit2 -1669.323
## 3       fit3 -1667.356
## 4       fit4 -1663.552
## 5       fit5 -1665.461
## 6       fit6 -1560.141
## 7       fit7 -1664.655
## 8       fit8 -1662.879
## 9   fit_auto -1662.198
```

```r
which.min(model_eva$AICc)
```

```
## [1] 2
```

Dunque, il modello con l'AICc più basso risulta il modello 2, per cui verrà selezionato.

```r
summary(fit2)
```

```
## Series: co_train
## ARIMA(1,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.2210  -0.5656  -0.8639
## s.e.  0.1003   0.0854   0.0202
##
## sigma^2 = 0.09816:  log likelihood = -179.21
## AIC=366.42   AICc=366.48   BIC=384.58
##
## Training set error measures:
##                      ME      RMSE       MAE        MPE      MAPE      MASE
## Training set 0.02503951 0.3097304 0.2407159 0.00682593 0.0684939 0.1560185
##                      ACF1
## Training set -0.004773924
```

Test significatività sui coefficienti:

```
coeftest(fit2, vcov(fit2))
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1   0.221012   0.100335   2.2027   0.02761 *
## ma1  -0.565581   0.085374  -6.6248 3.478e-11 ***
## sma1 -0.863919   0.020192 -42.7852 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Intevallo di confidenza ad un livello di confidenza del 95%.

```
coefci(fit2,level = 0.95)
```

```
##            2.5 %     97.5 %
## ar1   0.02435907  0.4176640
## ma1  -0.73291098 -0.3982519
## sma1 -0.90349449 -0.8243432
```

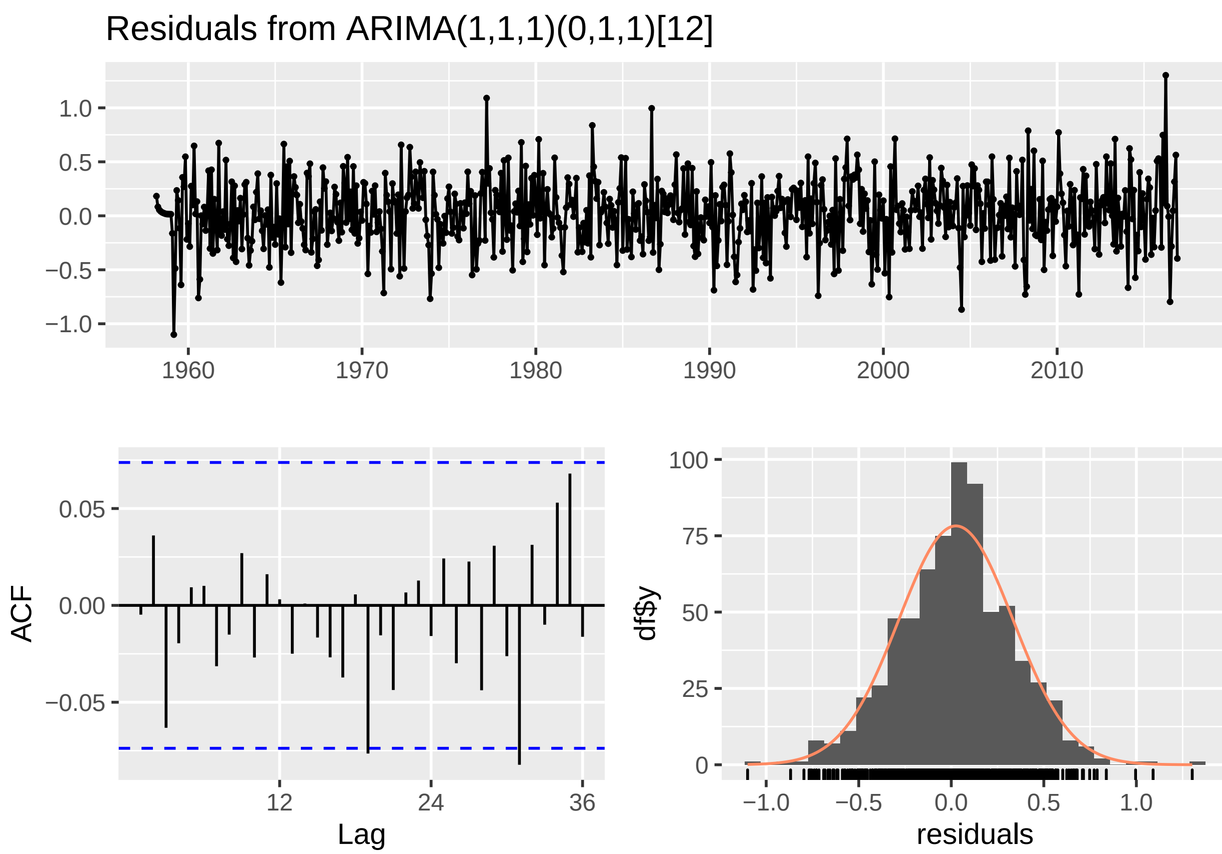
Viene inserita nella valutazione dei coefficienti anche il modello numero 1, poichè, come vedremo in seguito, risulterà essere il più performante fuori dal sample.

```
coeftest(fit1, vcov(fit1)) # Inserisco perchè so che è il migliore out of sample.
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ma1  -0.368701   0.040120  -9.190 < 2.2e-16 ***
## sma1 -0.863373   0.020508 -42.099 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

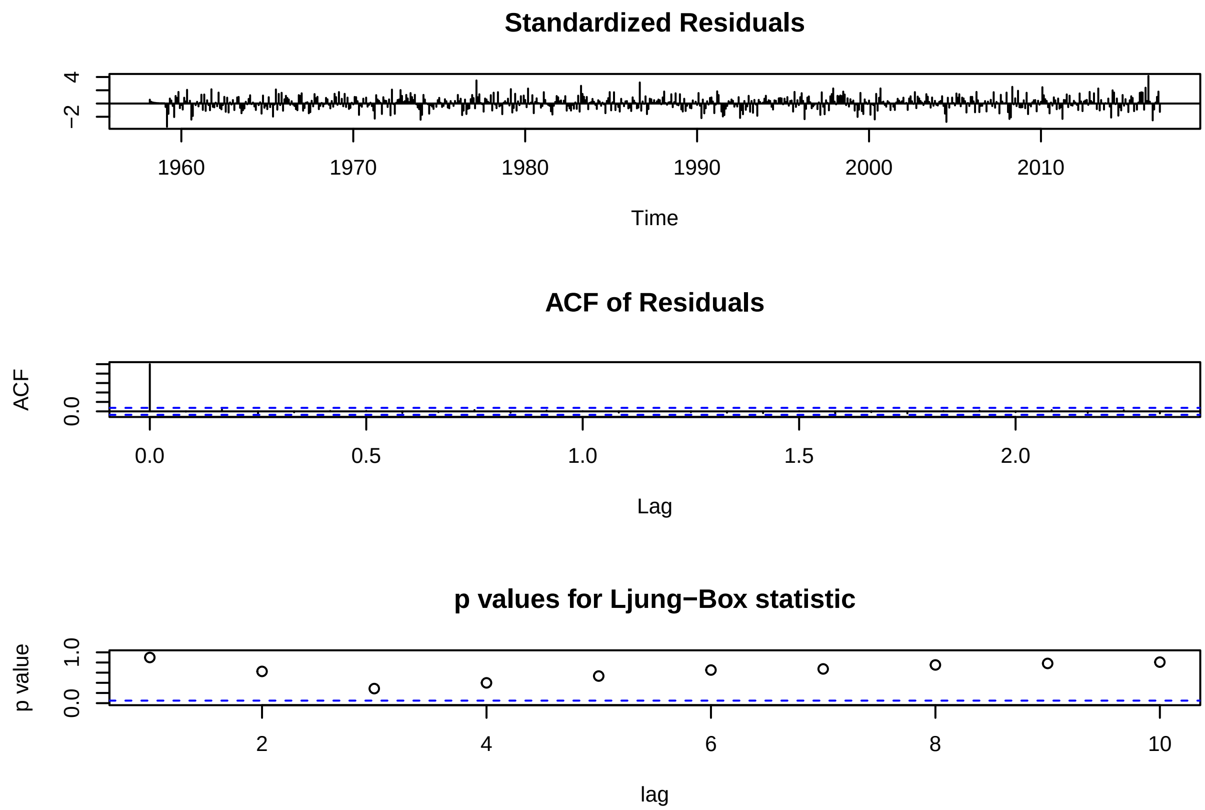
Analisi dei residui del fit2

```
checkresiduals(fit2)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)(0,1,1)[12]
## Q* = 14.64, df = 21, p-value = 0.8406
##
## Model df: 3.   Total lags used: 24
```

```
res= fit2$residuals
tsdiag(fit2 ,lag=30) # Per ogni lag rifiuto Ho.
```



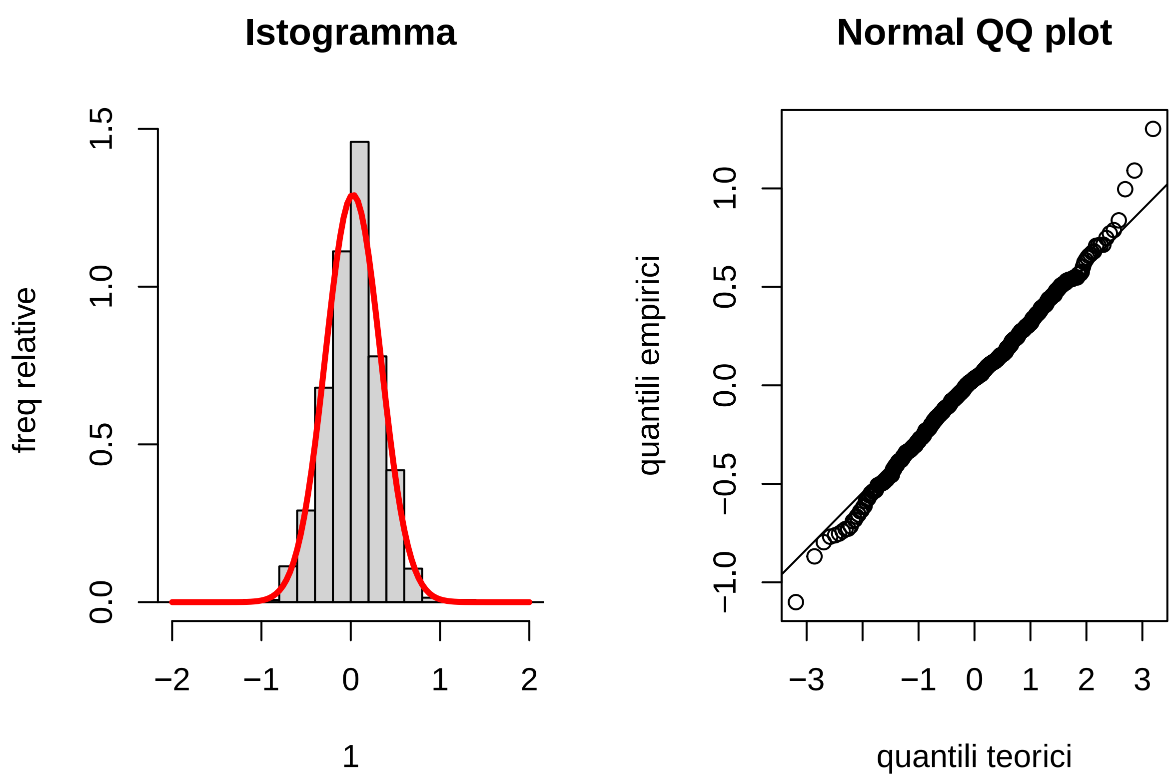
#il test conferma che gli errore non sono autocorrelati, - non rifiuto l'H_0.

```
Box.test(res, lag = 30, type = "Box-Pierce", fitdf= 3)
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 18.249, df = 27, p-value = 0.8957
```

#Test shapiro su normalità dell'errore.

```
par(mfrow = c(1,2))
hist(res, breaks=10, main="Istogramma", xlab= 1 ,ylab="freq relative ",
     ylim= c(0, 1.5), xlim=c(-2, 2), freq=F); abline(h=0)
curve(dnorm(x, mean(res), sd(res)), col="red", lwd=3, add =T)
qqnorm(res, main="Normal QQ plot", xlab="quantili teorici", ylab="quantili empirici");
qqline(res)
```




```r
#Graficamente sembra vi sià una distribuzione normale dei residui, tuttavia il test conferma tutt'altro
```

```r
shapiroTest(res, title = "Test shapiro-Wilk")  # rifiuto H_0, i residui non hanno distribuzione normale
```

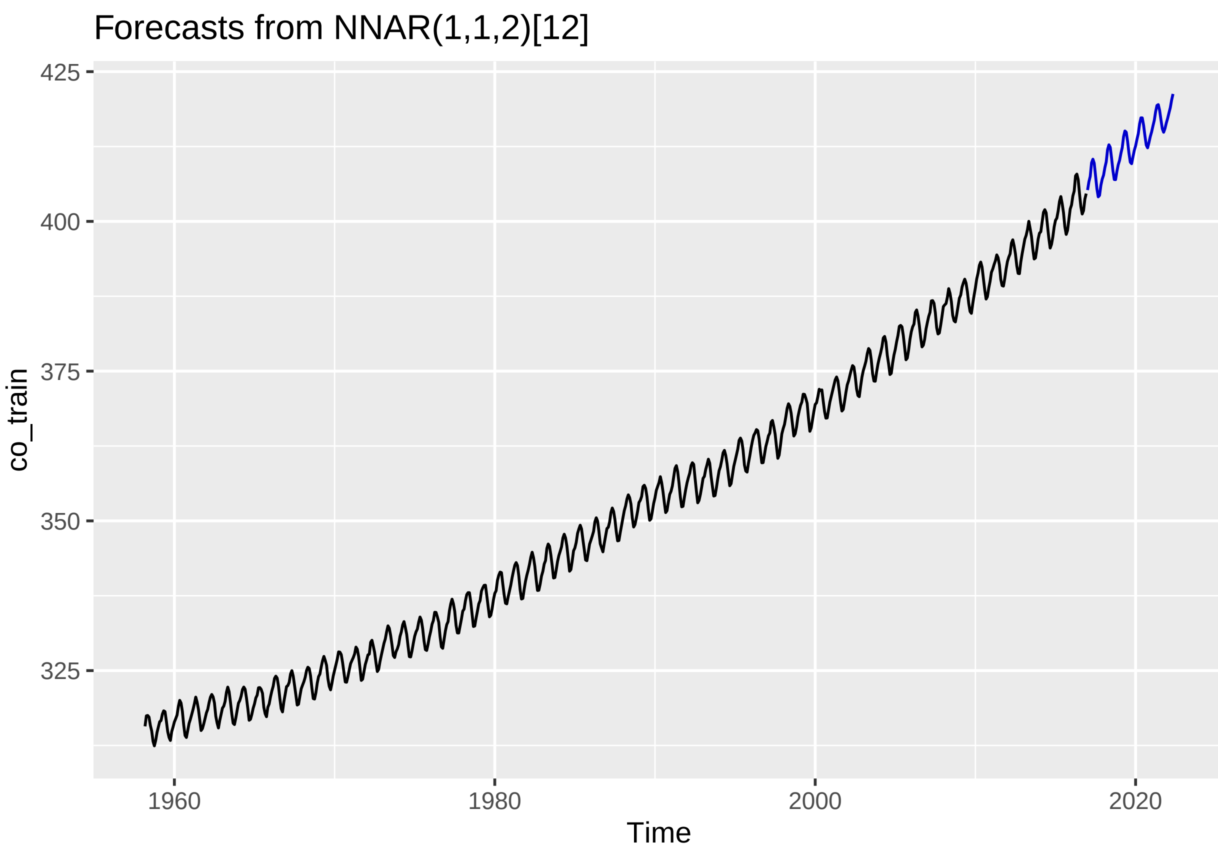
```
##
## Title:
##  Test shapiro-Wilk
##
## Test Results:
##   STATISTIC:
##     W: 0.996
##   P VALUE:
##     0.06678
##
## Description:
##  Fri Jun 24 15:01:02 2022 by user:
```

NNAR

```r
set.seed(2022)
fit_nnar <- nnetar(co_train)
fit_nnar$model
```

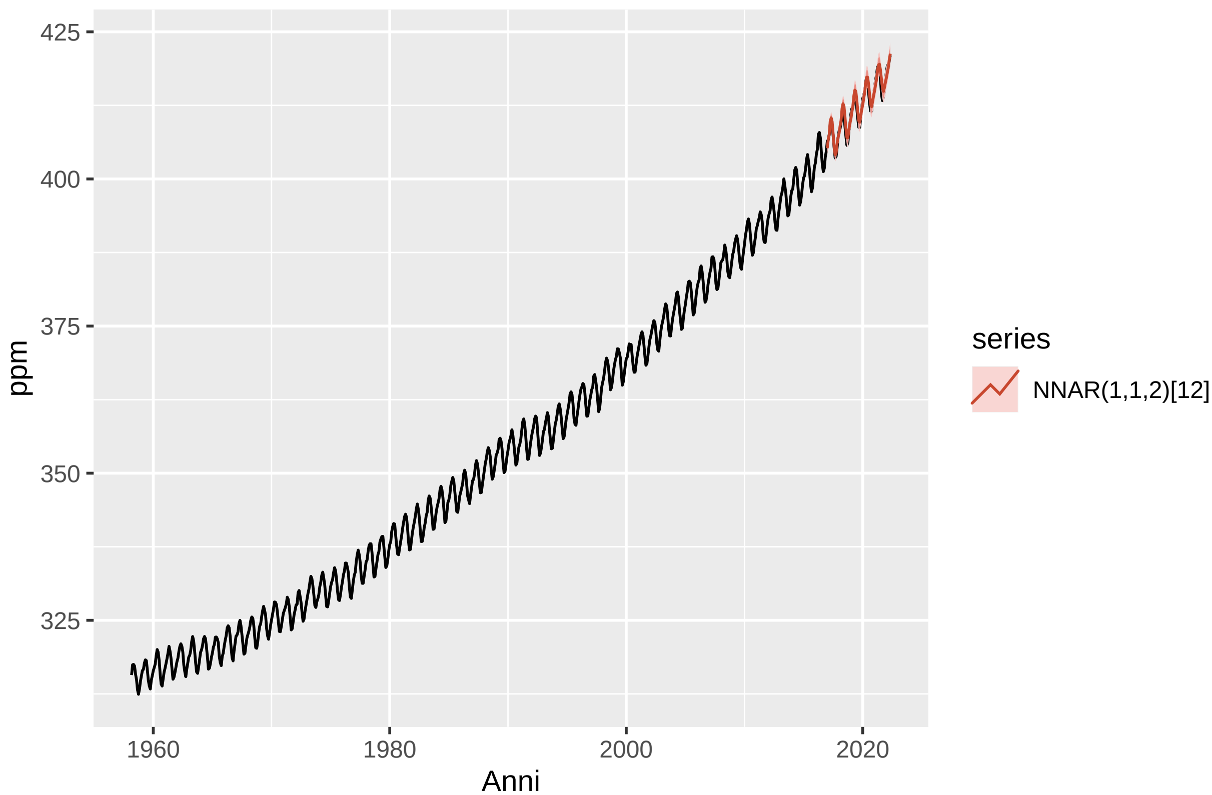
```
##
## Average of 20 networks, each of which is
## a 2-2-1 network with 9 weights
## options were - linear output units
```

```
### Previsione con NNAR.
prev <- forecast(fit_nnar, h=65)
autoplot(prev)
```



#Intervallo di previsone simulato

```
prev2 <- forecast(fit_nnar, PI=TRUE, npaths=1000, level=c(80, 95), h=65)
autoplot(co2) +
  autolayer(prev2, PI=T, series = "NNAR(1,1,2)[12]") +
  xlab("Anni")+ylab("ppm")+
  ggtitle("")
```



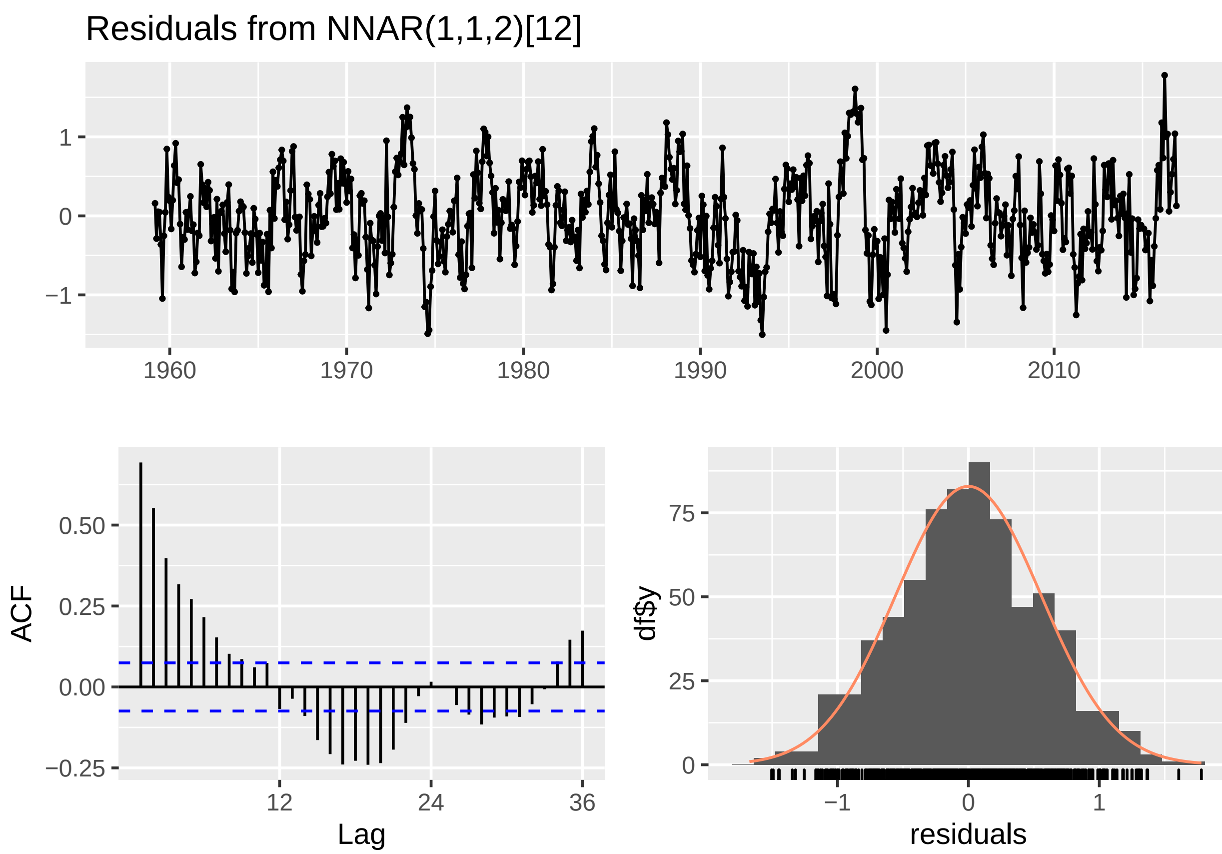
#Accuratezza previsioni, valutazione su test-set. RMSE piu elevato rispetto al migliore modello ARIMA, quindi non verrà selezionato.

```
a_nn <-round(accuracy(prev, co_test)[2, 2],3)
```

#Analisi redisuo NNAR

```
checkresiduals(fit_nnar)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```



NNAR Modello non buono, lo confermano i risultati. Non è correttamente identificato.

#Performance sul test set Si valuta la performance effettiva dei Modelli sul Test-set.

```
fc1 <- forecast(fit1, h=65, level = c(90,95))
fc2 <- forecast(fit2, h=65, level = c(90,95))
fc3 <- forecast(fit3, h=65, level = c(90,95))
fc4 <- forecast(fit4, h=65, level = c(90,95))
fc5 <- forecast(fit5, h=65, level = c(90,95))
fc6 <- forecast(fit6, h=65, level = c(90,95))
fc7 <- forecast(fit7, h=65, level = c(90,95))
fc_auto <- forecast(fit_auto, h=65, level = c(90,95))
fit9 <- rwf(co_train, drift = TRUE, h=65) # random walk
```

Valutazione modelli tramite RMSE.

```
a1 <- round(accuracy(fc1, co_test)[2, 2],3)
a2 <- round(accuracy(fc2, co_test)[2, 2],3) # Migliore
a3 <- round(accuracy(fc3, co_test)[2, 2],3)
a4 <- round(accuracy(fc4, co_test)[2, 2],3)
a5 <- round(accuracy(fc5, co_test)[2, 2],3)
a6 <- round(accuracy(fc6, co_test)[2, 2],3)
a7 <- round(accuracy(fc7, co_test)[2, 2],3)
```

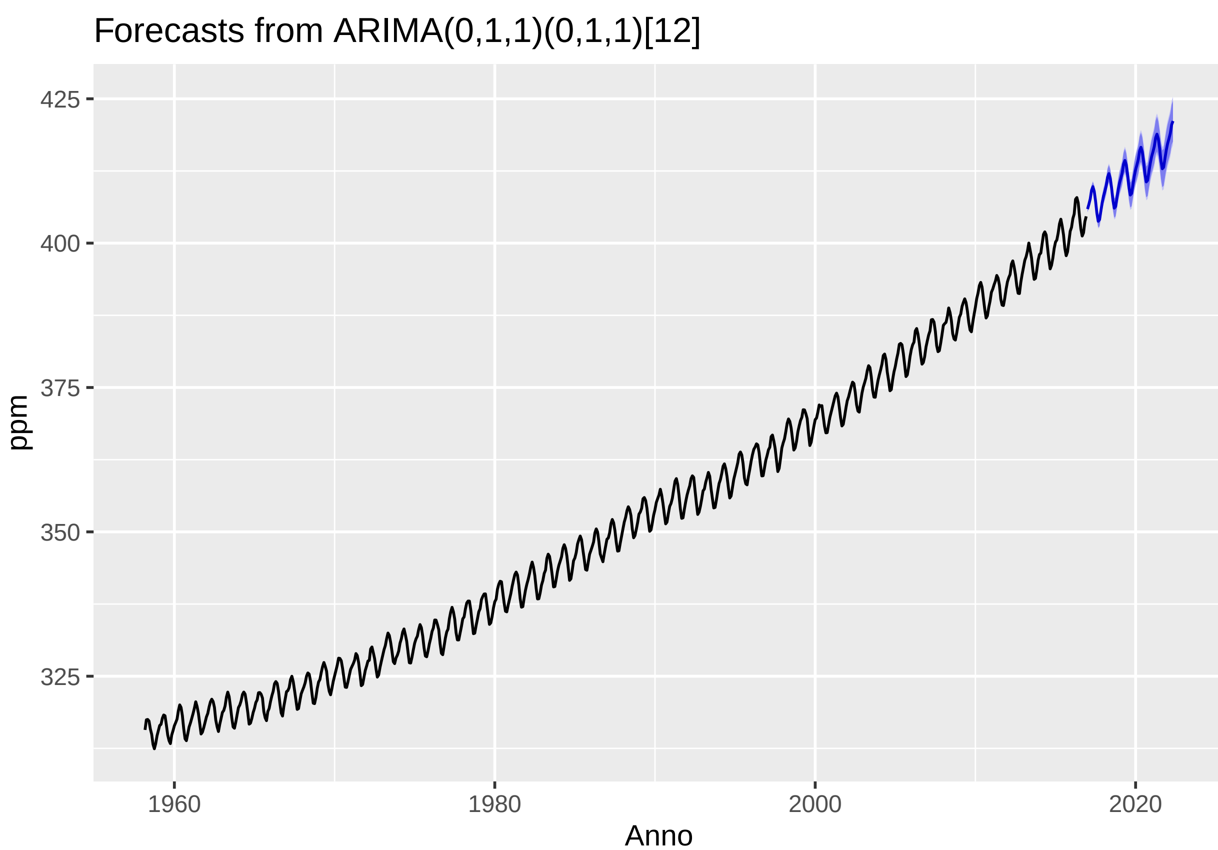
```
a9 <- round(accuracy(fit9, co_test)[2, 2],3)
a_auto <- round(accuracy(fc_auto,co_test)[2, 2],3)
```

Fuori dal sample il migliore modello, RMSE minore : ARIMA(0,1,1)(0,1,1)

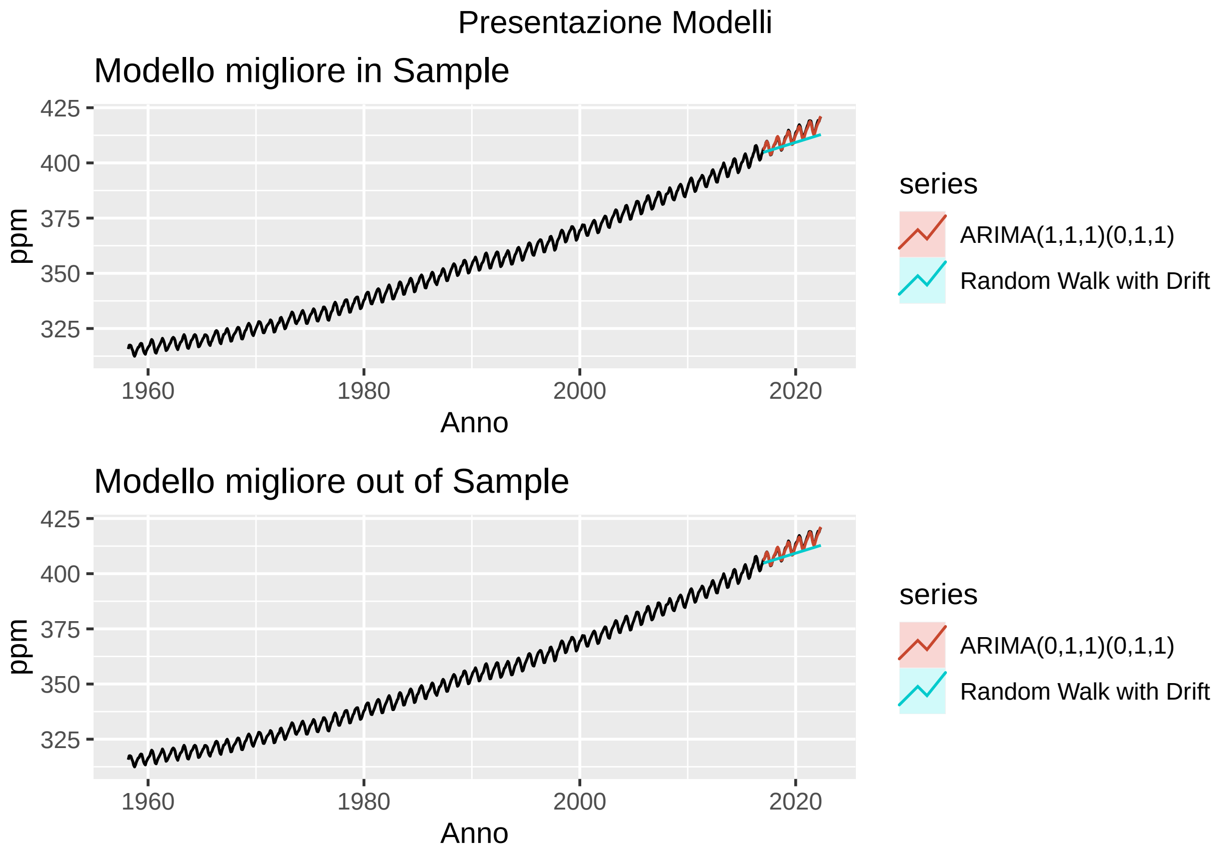
Plottiamo i risultati

Viene presentato il modello che nel test set performa meglio, il termini di RMSE minore.

```
autoplot(fc1) + xlab("Anno")+ylab("ppm")
```



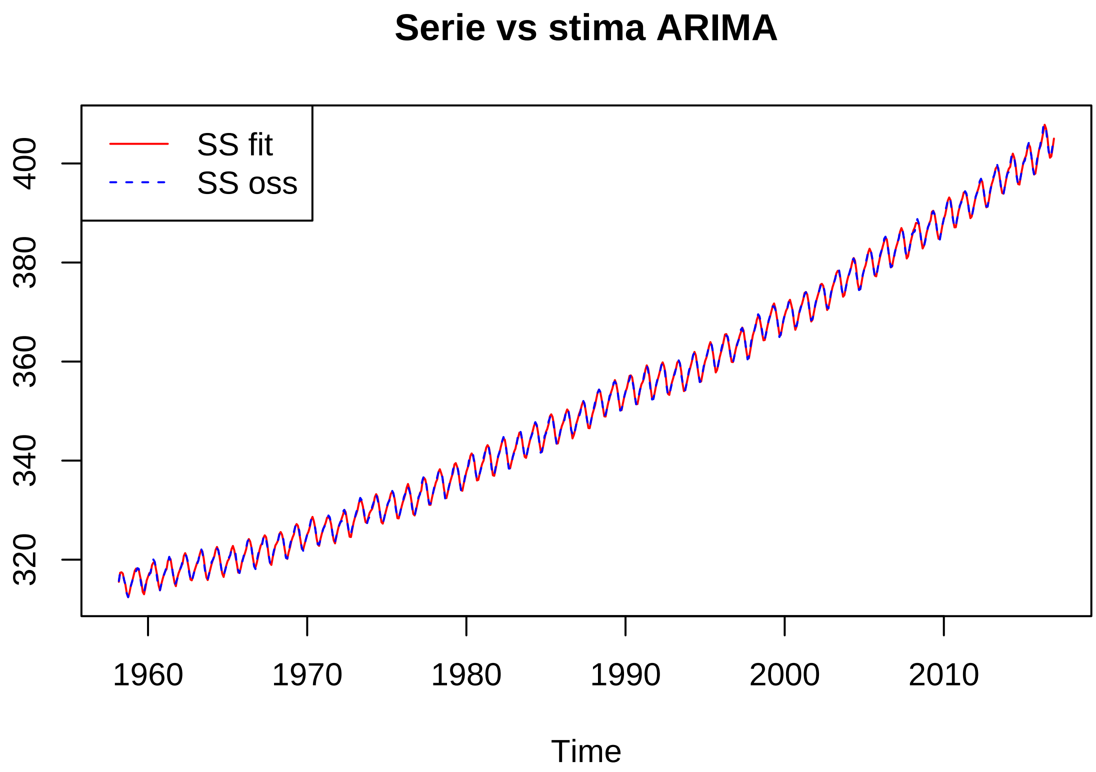

```
p4=autoplot(co2)+
  autolayer(fc1, PI=F, series = "ARIMA(0,1,1)(0,1,1)" ) +
  autolayer(fit9, PI=F, series = "Random Walk with Drift") +
  xlab("Anno")+ylab("ppm")+
  ggtitle("Modello migliore out of Sample")

p5=autoplot(co2)+
  autolayer(fc2, PI=F, series = "ARIMA(1,1,1)(0,1,1)") +
  autolayer(fit9, PI=F, series = "Random Walk with Drift") +
  xlab("Anno")+ylab("ppm")+
  ggtitle("Modello migliore in Sample")
gridExtra::grid.arrange(p5,p4, top = "Presentazione Modelli", newpage = TRUE)
```



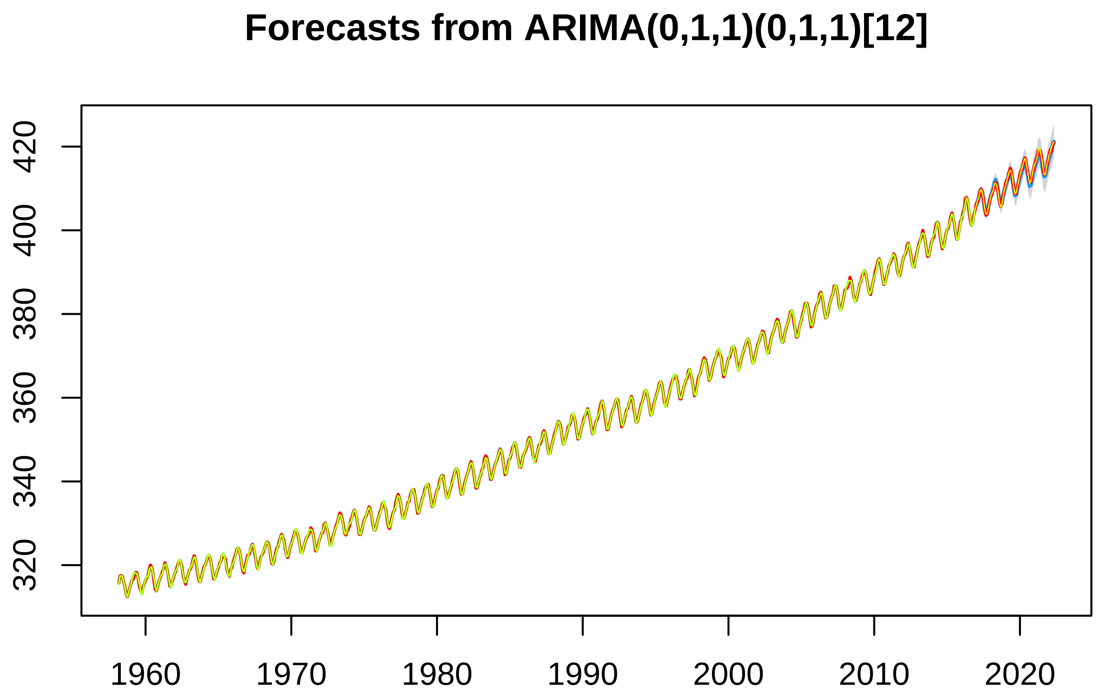
Plot dei risultati, si selezionerà comunque il modello che performa meglio fuori dal sample, che risulta essere anche il più semplice.

```
y_trainf1=fit1$fitted
fit1_tot = Arima(co2, order = c(0,1,1), seasonal = c(0,1,1))
y_tot = fit1_tot$fitted
```

```
ts.plot(y_trainf1, co_train, lty=c(1:2), col=c("red", "blue"), main="Serie vs stima ARIMA")
legend("topleft", legend=c("SS fit", "SS oss"), lty=c(1:2), col=c("red", " blue"))
```



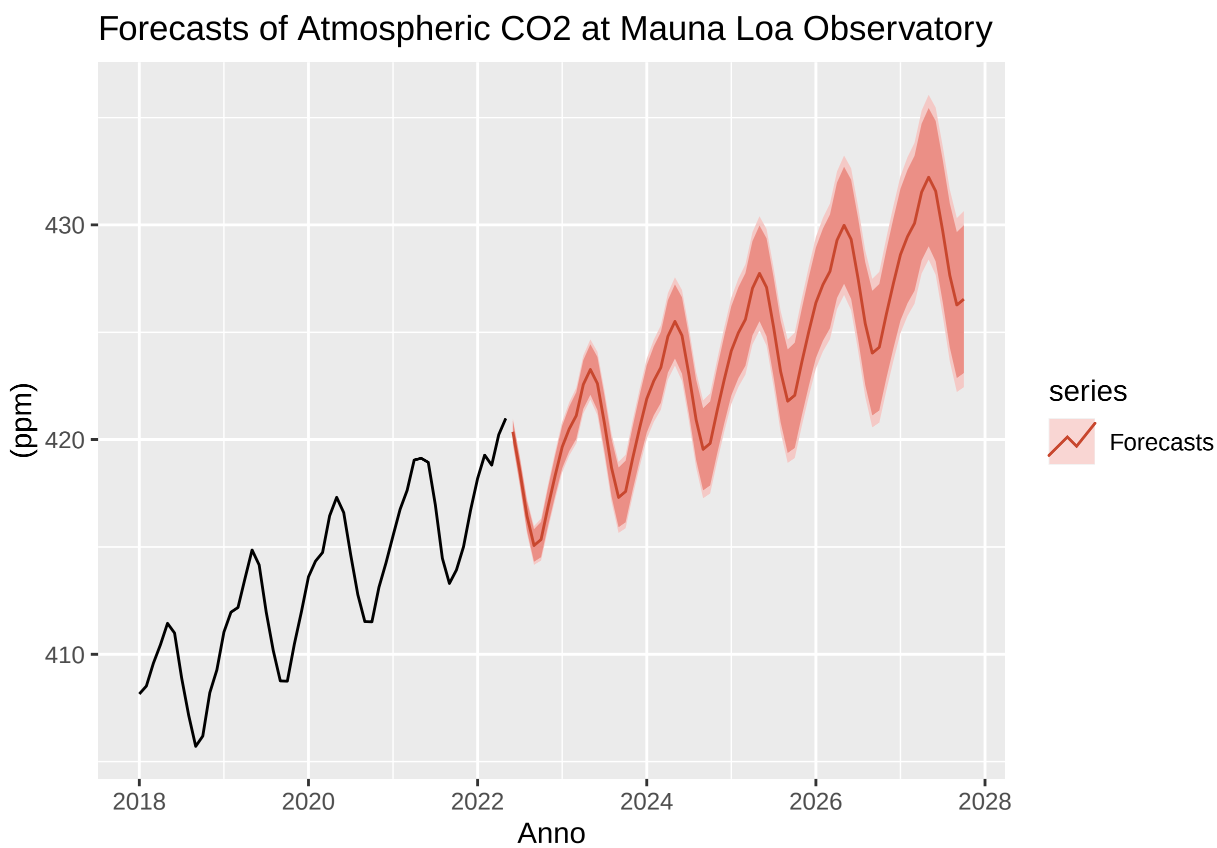

```
plot(fc1)
lines(co2 , lwd= 1.5, col="red")
lines(y_trainf1, lwd=1, col="green")
lines(y_tot, lwd=0.6, col="yellow")
```



Il modello sembra essersi adattato perfettamente ai dati, questo potrebbe indicare la presenza di overfitting.

Previsioni future.

```
prev_fut  <- Arima(co2, order = c(0,1,1), seasonal = c(0,1,1))
forecast_fut <- forecast(prev_fut, h = 65, level = c(90,95))

windo <- window(co2, start = c(2018,1))
autoplot(windo)+
  autolayer(forecast_fut, series = "Forecasts") +
  xlab("Anno")+ylab("(ppm)")+
  ggtitle("Forecasts of Atmospheric CO2 at Mauna Loa Observatory")
```



Forecasts of Atmospheric CO2 at Mauna Loa Observatory
(ppm)
410
420
430
2018
2020
2022
2024
2026
2028
Anno
series
Forecasts