

Progetto di Statistica II: Terza Relazione

Corso di Laurea Magistrale in
Artificial Intelligence and Data Engineering

Simone Landi

A.A. 2022-2023

Indice

Introduzione	1
Dataset	1
Analisi	1
Analisi Preliminare.....	1
Decomposizione	2
Analisi dei residui.....	3
Predizione e autovalutazione	5
Holt Winters	5
Autoregressione	5
Autoregressione manuale	5
Metodo di Yule-Walker	6
Metodo dei minimi quadrati	6
Confronto tra i metodi.....	7
Autovalutazione.....	7
Residui	7
Conclusioni	9
Appendice.....	10
Codice Analisi.....	10

Introduzione

Si ipotizza che la seguente analisi sia stata commissionata dal governo americano con lo scopo di riuscire a prevedere l'andamento delle miglia percorse nei prossimi anni.

Una previsione affidabile del trend è fondamentale per valutare quali siano i migliori momenti per incrementare l'offerta di percorsi stradali; inoltre la conoscenza della stagionalità delle miglia percorse permetterebbe di pianificare i lavori di manutenzione o di incremento delle strade.

Dataset

L'analisi è stata svolta utilizzando il seguente dataset:

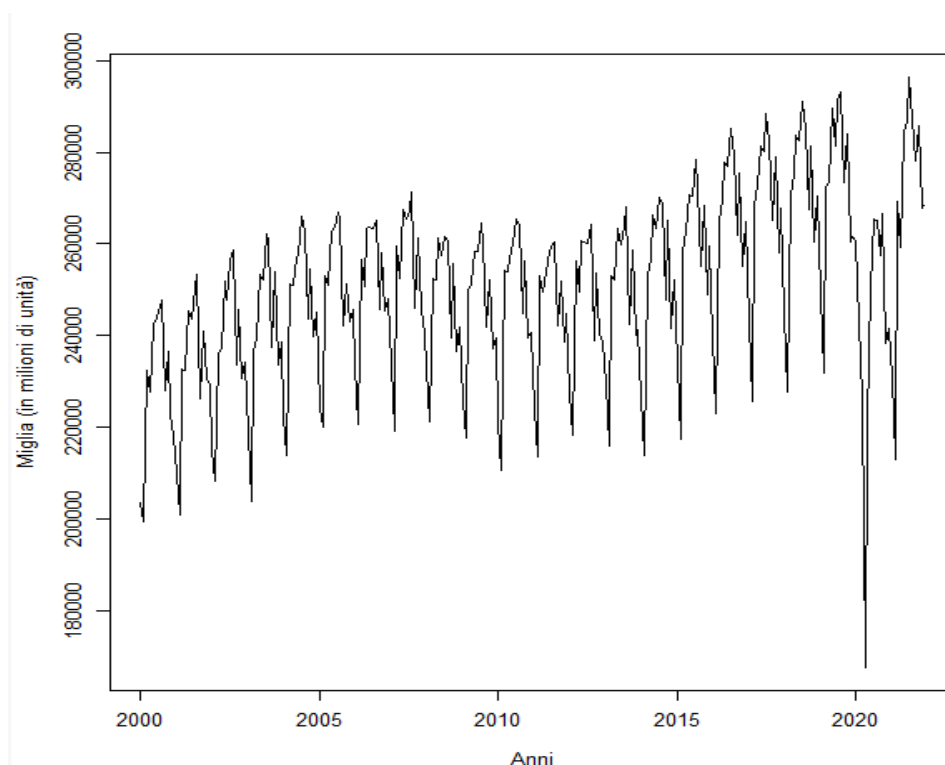
<https://fred.stlouisfed.org/series/TRFVOLUSM227NFWA>

Esso contiene una serie storica con osservazioni mensili riguardanti le miglia percorse dai veicoli negli Stati Uniti. Per l'analisi sono state considerate solo le osservazioni dal 1 gennaio 2000 al 1 dicembre 2021 (264 osservazioni).

Analisi

Analisi Preliminare

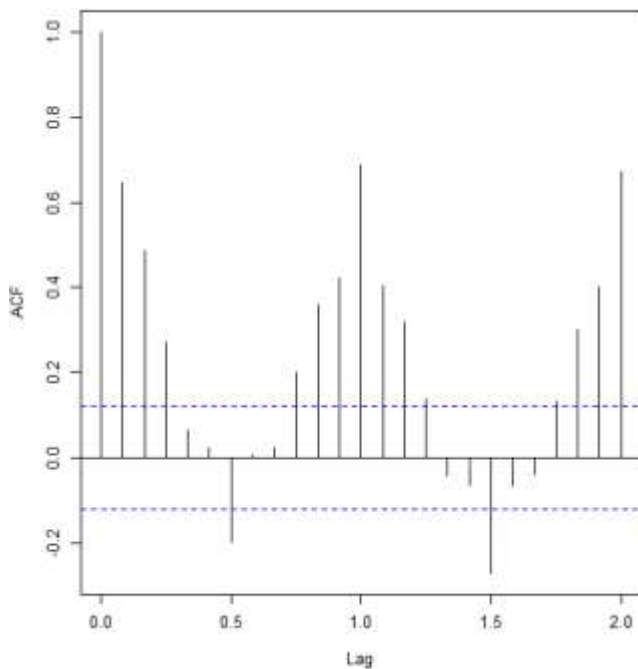
Una prima esplorazione grafica dei dati mostra una stagionalità annuale abbastanza evidente e un trend non troppo regolare: si ha infatti un andamento crescente fino al 2008, dove si registra un piccolo calo delle miglia percorse dovuto alla crisi globale conseguente allo scoppio della bolla immobiliare americana; dal 2008 al 2012 si registra invece un andamento costante. Dal 2012 si registra una crescita fino al 2020, dove viene registrato un picco negativo, questa volta dovuto alla pandemia di Covid-19.



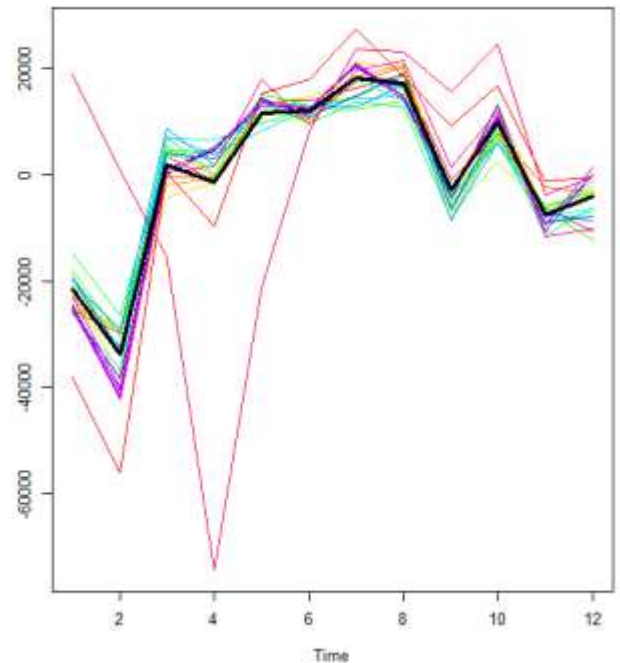
Per capire effettivamente quali siano le componenti della serie, si osserva per prima la sua funzione di autocorrelazione, questa sembra mostrare che sia presente una certa stagionalità nella serie dati i picchi positivi. Un'ulteriore verifica sulla presenza di stagionalità nella serie è rappresentare in un grafico l'andamento per anni: notiamo che i vari anni seguono un andamento piuttosto simile, perciò l'idea che sia

presente stagionalità è rafforzata. Le linee che si discostano maggiormente sono quelle relative agli anni più recenti, dovuto alla pandemia di Covid-19.

Autocorrelazione Miglia



Confronto stagionalità negli anni

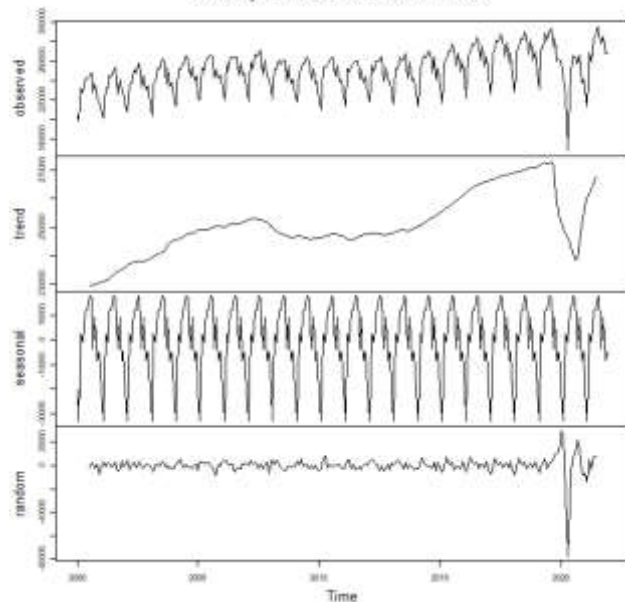


Decomposizione

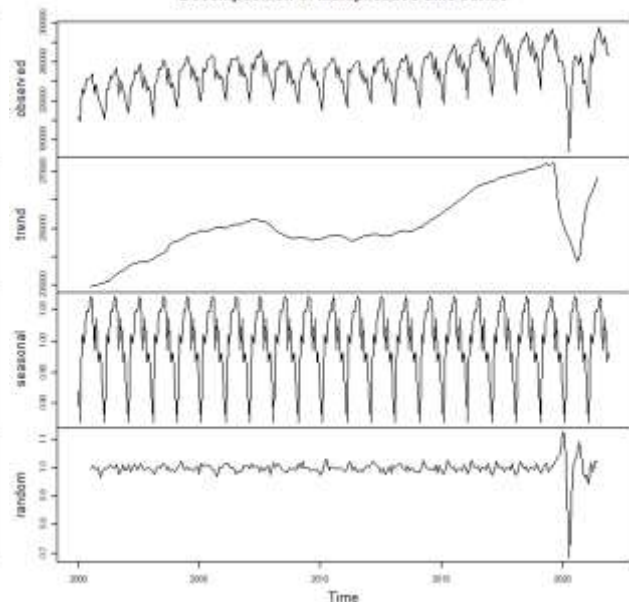
La decomposizione additiva presenta una stagionalità annuale che conferma le previsioni.

La decomposizione moltiplicativa ovviamente presenta lo stesso trend e stagionalità della decomposizione additiva, anche i residui sembrano non avere una struttura temporale come prima.

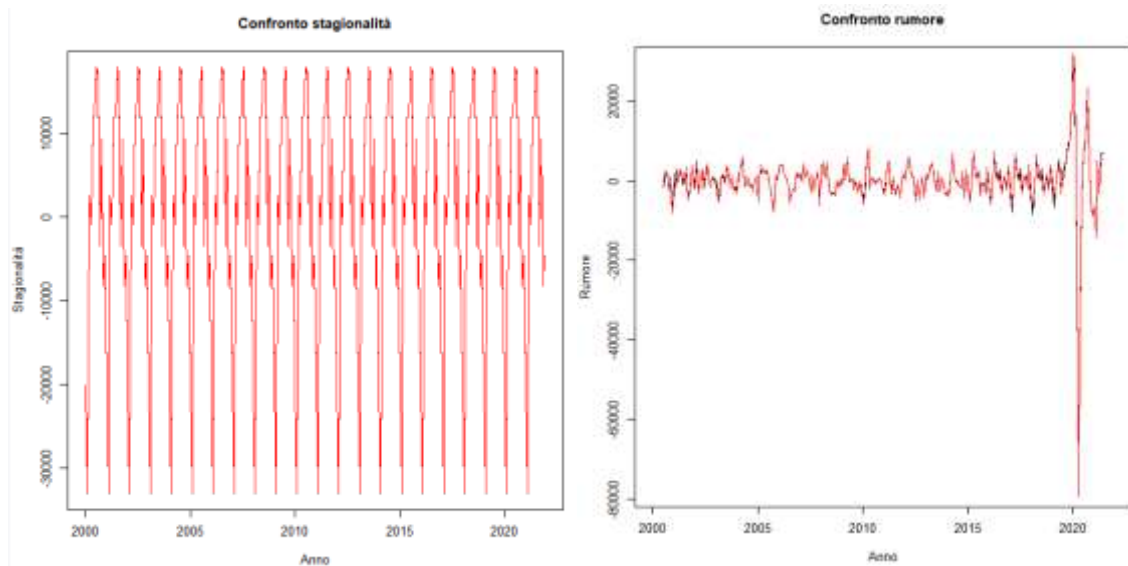
Decomposition of additive time series



Decomposition of multiplicative time series



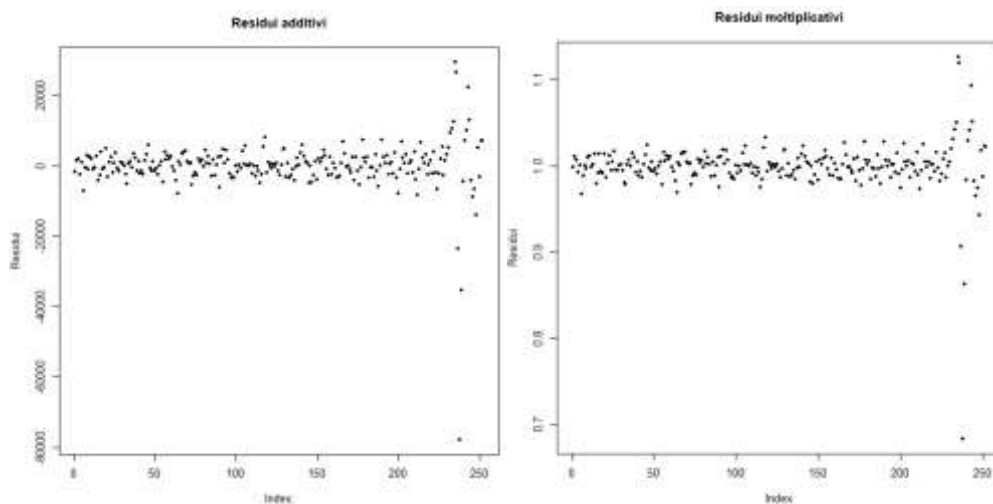
Confrontando le stagionalità e gli errori estratti dai due modelli, si nota che le differenze tra le componenti non risultano significative.



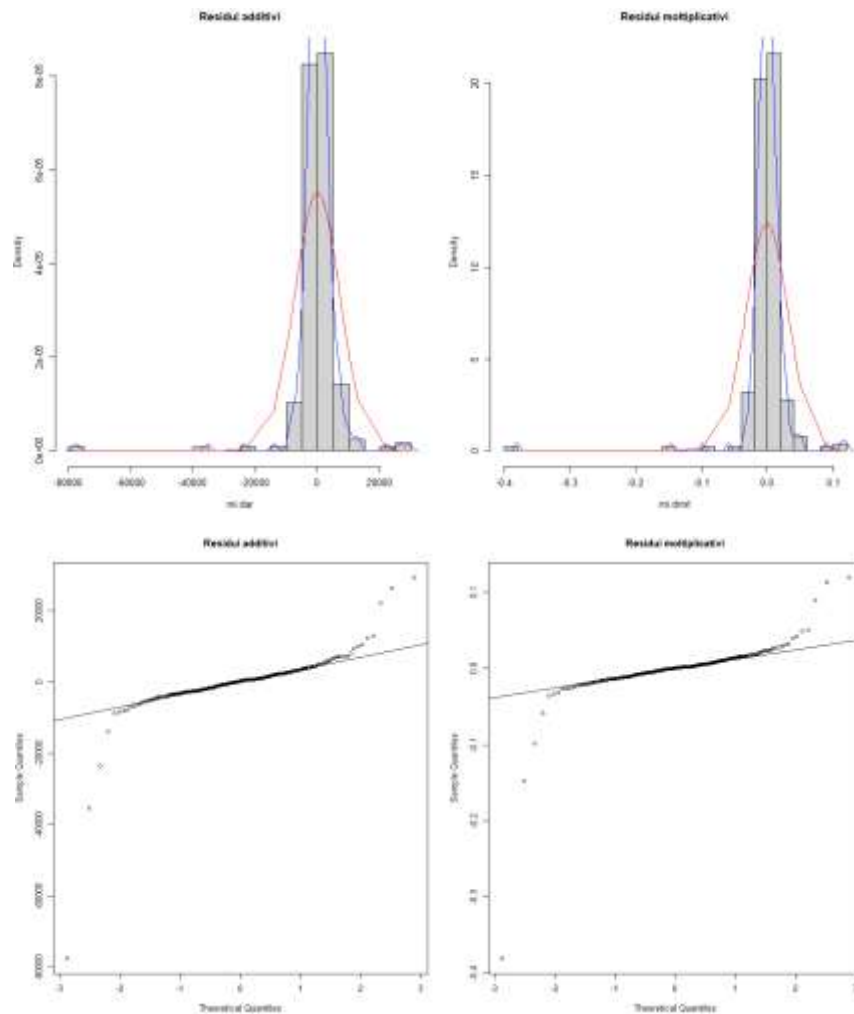
Analisi dei residui

Dai grafici sui residui è evidente come entrambi i modelli ottengono risultati molto simili.

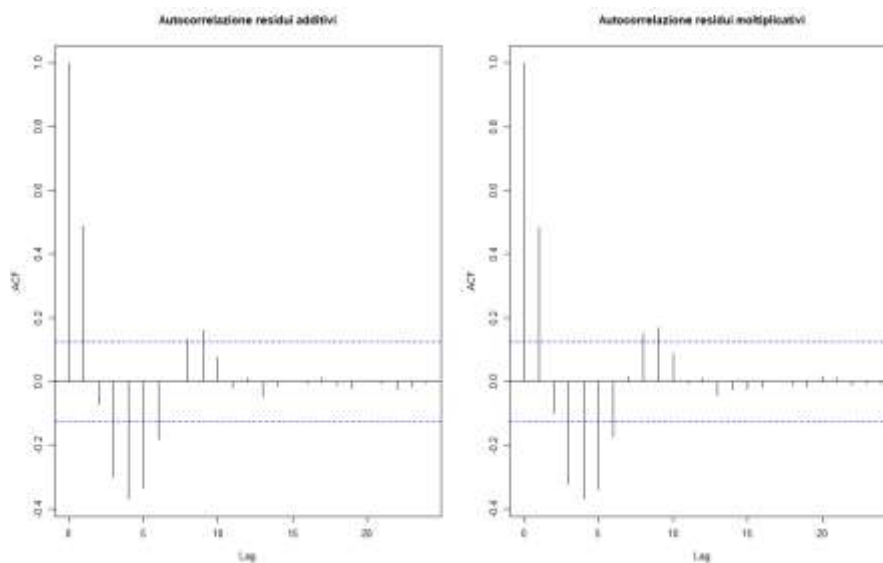
La varianza spiegata è dell'86% per il modello additivo e dell'84% per il modello moltiplicativo, entrambi valori ottimi.



L'analisi dei residui restituisce risultati simili per entrambi i modelli analizzati: i residui hanno una distribuzione casuale e non sembrano aver catturato stagionalità; in nessun caso, però, risultano essere Gaussiani dato che i test di Shapiro Wilk danno sempre pessimi p-value (entrambi $2.2e-16$, principalmente per la presenza di residui nelle code della gaussiana).



Osservando i grafici delle autocorrelazioni, si nota nuovamente un comportamento molto simile per entrambi i modelli.



Per scegliere quale dei due modelli utilizzare, è stato effettuato un confronto tra le deviazioni standard della funzione di autocorrelazione. Per le considerazioni fatte in precedenza e per i risultati del confronto delle deviazioni standard, la scelta è ricaduta sul modello di decomposizione additivo.

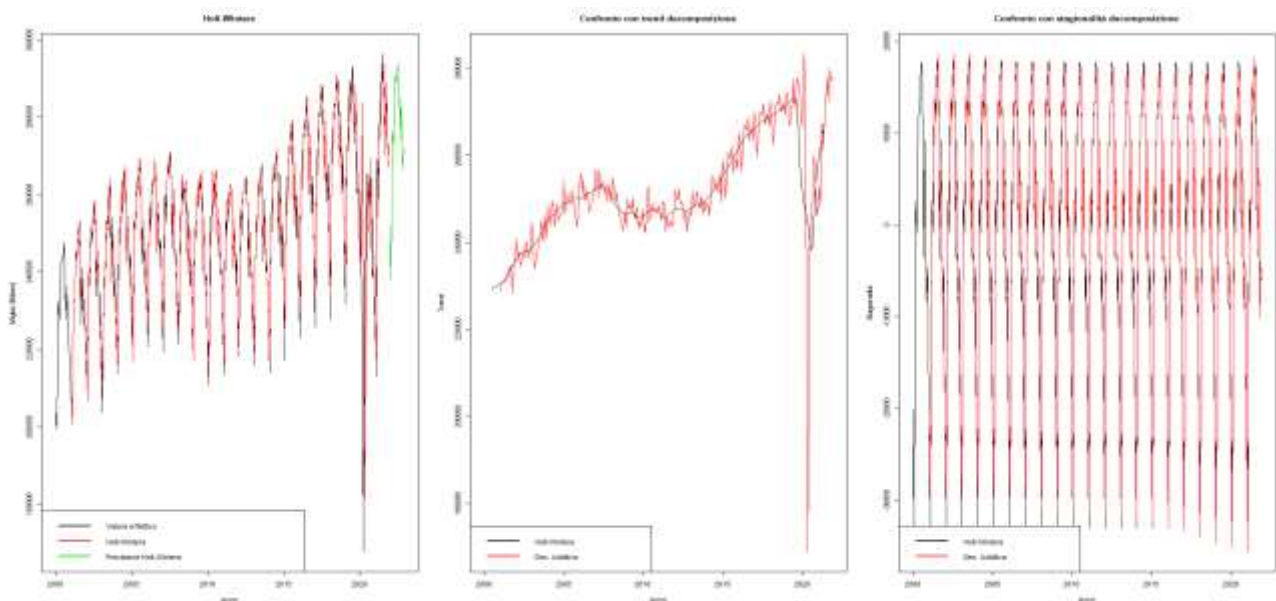
```
> # deviazione standard additivo
> sd(acf(mi.dar,plot=F)$acf)
[1] 0.2632164
> # deviazione standard moltiplicativo
> sd(acf(mi.dmr,plot=F)$acf)
[1] 0.2649652
```

Predizione e autovalutazione

Adesso valuteremo i vari metodi di previsione per serie storiche separatamente, per poi confrontarli in fase di autovalutazione.

Holt Winters

Il modello di Holt Winters considerato possiede i seguenti parametri calcolati automaticamente: $\alpha = 0.92$, $\beta = 0$, $\gamma = 1$. Confrontando il metodo di Holt-Winters e la decomposizione additiva possiamo notare come la stagionalità sia molto simile, così come il trend ad eccezione dell'anno 2020 dovuto alla pandemia di Covid-19. Di seguito i grafici di confronto con i risultati della decomposizione.



Autoregressione

Autoregressione manuale

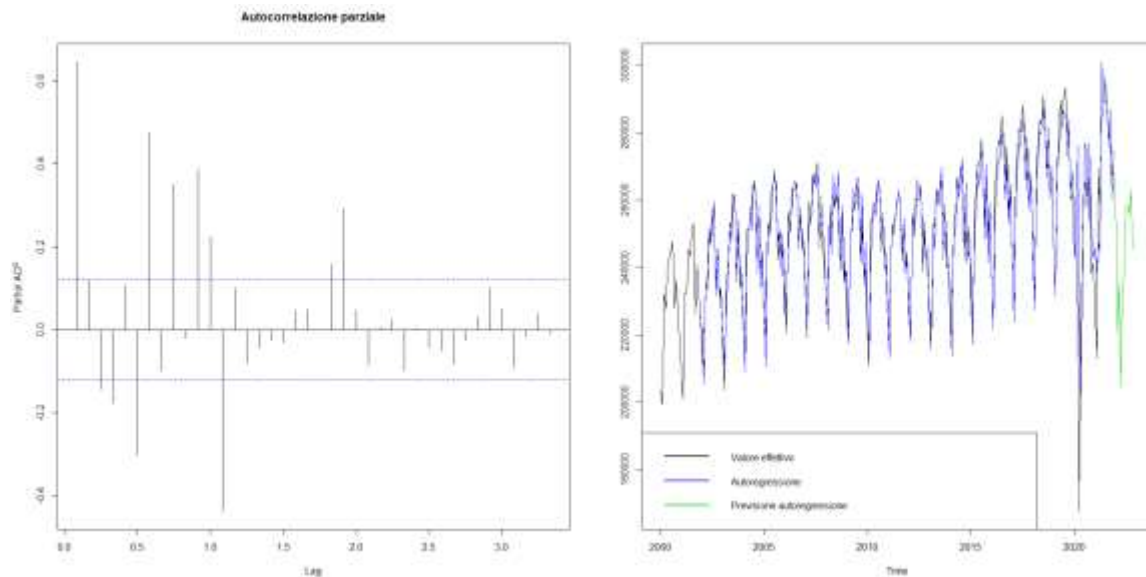
La funzione di autocorrelazione parziale evidenzia che il ritardo utile per la regressione arriva all'incirca a 23 mesi. Applicando una riduzione del modello lineare in base al p-value delle variabili d'ingresso otteniamo un modello ridotto con 7 variabili di ingresso e una varianza spiegata dell'85%.

```
Call:
lm(formula = X24 ~ X1 + X10 + X11 + X12 + X13 + X22 + X23, data = map)

Residuals:
    Min       1Q   Median       3Q      Max
-56906  -2560    749   3986  21987

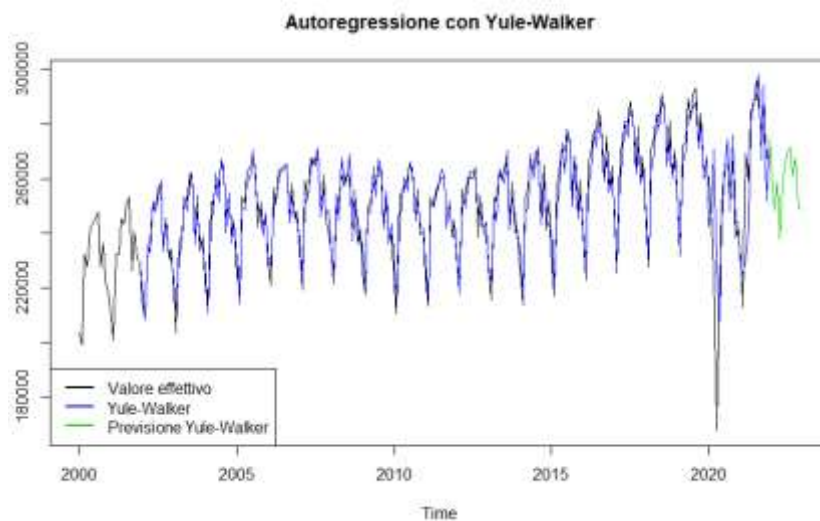
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.272e+04  8.780e+03   1.449   0.149
X1           3.195e-01  4.870e-02   6.560 3.43e-10 ***
X10          2.889e-01  5.786e-02   4.994 1.16e-06 ***
X11          -8.558e-01  6.313e-02  -13.557 < 2e-16 ***
X12           9.139e-01  3.870e-02  23.614 < 2e-16 ***
X13          -3.648e-01  5.289e-02   -6.898 4.93e-11 ***
X22          -2.756e-01  5.762e-02   -4.784 3.05e-06 ***
X23           9.266e-01  5.782e-02  16.025 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7609 on 233 degrees of freedom
Multiple R-squared:  0.8539,    Adjusted R-squared:  0.8495
F-statistic: 194.5 on 7 and 233 DF,  p-value: < 2.2e-16
```



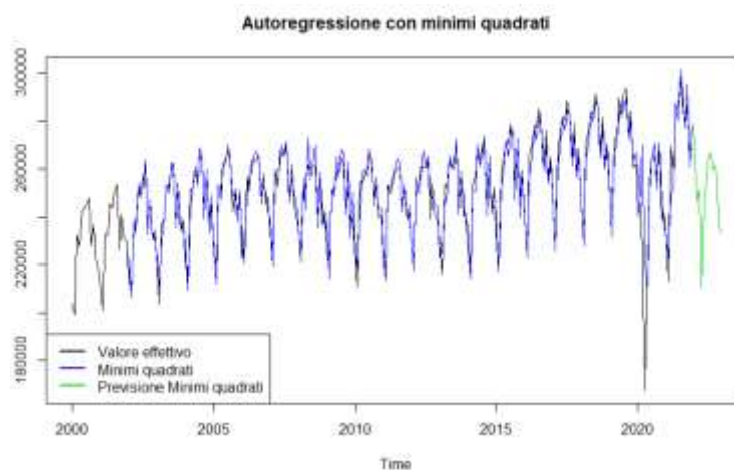
Metodo di Yule-Walker

Il metodo autoregressivo di Yule-Walker sceglie di utilizzare 23 lag. La varianza spiegata passa da 85% a 78%. Il modello non sembra particolarmente adatto per la nostra serie storica.



Metodo dei minimi quadrati

Anche questo caso il modello sceglie di utilizzare 23 lag in ingresso. Stavolta però il modello ha una varianza spiegata dell'88%, la previsione non risulta comunque migliore.

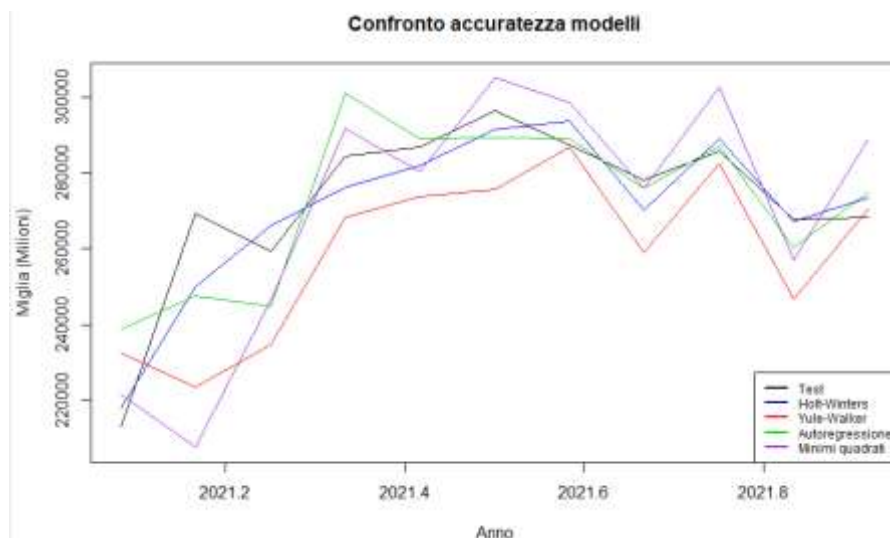


Confronto tra i metodi

Autovalutazione

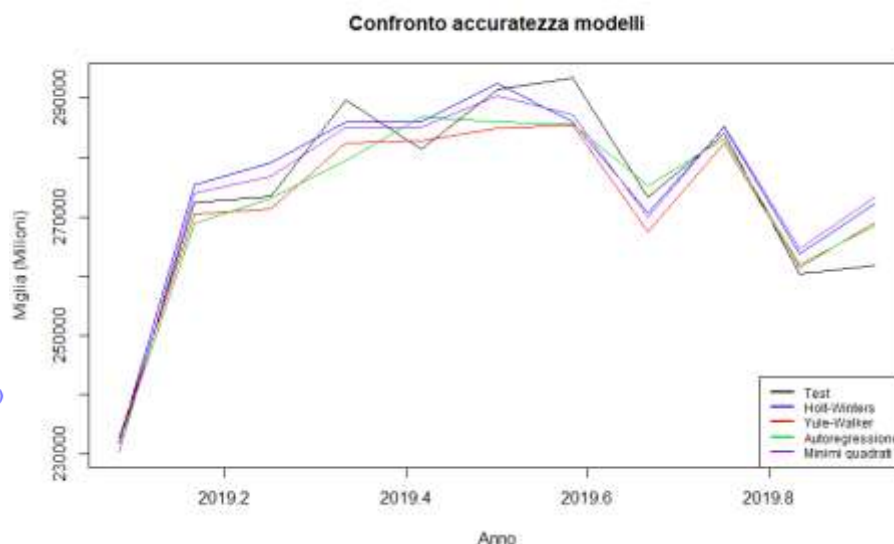
Per testare la capacità predittiva dei metodi analizzati, è stata effettuata un'autovalutazione sui dati dell'ultimo anno della serie, ovvero il 2021. I risultati ottenuti non sono stati dei migliori: l'errore medio inferiore è stato ottenuto con il metodo Holt-Winters, ma comunque siamo lontani da risultati soddisfacenti.

```
> sqrt(mean((test - res.hwm)^2))
[1] 8033.503
> sqrt(mean((test - res.ar)^2))
[1] 20809.36
> sqrt(mean((test - res.lm)^2))
[1] 12760.35
> sqrt(mean((test - res.ls)^2))
[1] 21691.21
```



L'accuratezza dei metodi valutata solo su dati relativi all'annata 2021 non è sufficientemente affidabile in quanto fortemente influenzata da fattori esterni che hanno indotto comportamenti anomali nei dati. Pertanto è stato ritenuto necessario ripetere il confronto considerando i dati del 2019 come test set (anno prima della pandemia). Questa volta i risultati sono stati decisamente migliori: il modello dei minimi quadrati risulta essere il migliore, seguono il modello di Holt-Winters e il modello di Yule Walker che hanno errori medi molto simili.

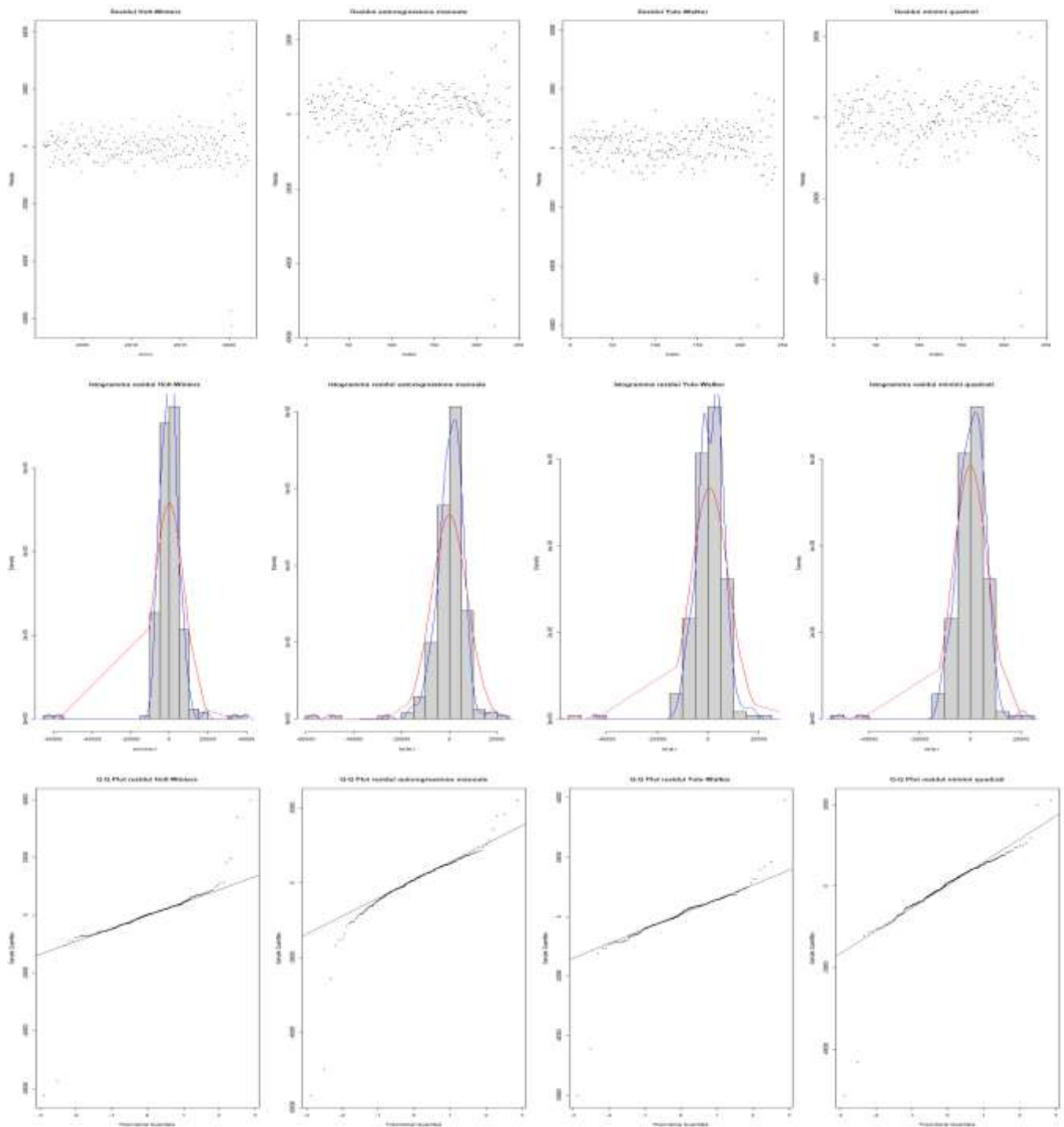
```
> sqrt(mean((test - res.hwm)^2))
[1] 4884.105
> sqrt(mean((test - res.ar)^2))
[1] 4844.212
> sqrt(mean((test - res.lm)^2))
[1] 5149.875
> sqrt(mean((test - res.ls)^2))
[1] 4797.715
```

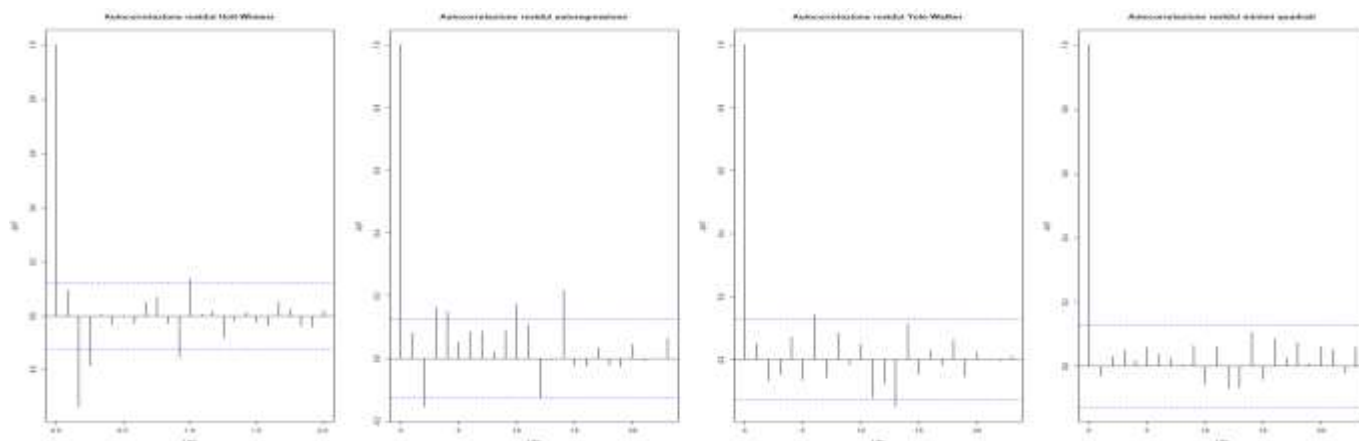


Residui

Come si può notare dai grafici, i residui dei modelli hanno tutti una struttura molto simile. Infatti tutti hanno residui più compatti all'inizio della serie che crescono negli anni finali. Questo è dovuto alla pandemia di Covid-19. Oltre ai grafici, il fatto che i residui non abbiano una distribuzione normale è evidenziato anche dal basso valore del p-value dei test di Shapiro-Wilk. Infatti il valore registrato da tutti i modelli è stato $2.2e-16$. Osservando l'autocorrelazione osserviamo che il modello migliore è quello dei minimi quadrati seguito da

Holt-Winters e Yule-Walker, mentre quello autoregressivo è il peggiore (l'ordine dei modelli nelle immagini è il seguente: Holt-Winters, autoregressione, Yule-walker, minimi quadrati).

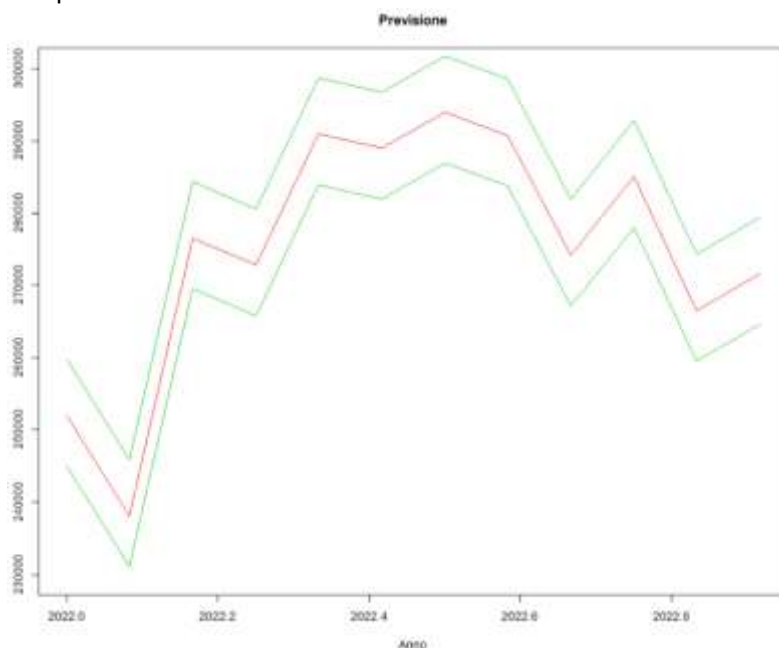




Conclusioni

In ultima analisi, i modelli dei minimi quadrati, Holt-Winter e Yule-Walker sono equiparabili. Dalle considerazioni fatte in precedenza però Holt-Winter ottiene risultati leggermente migliori e riporta anche una maggiore accuratezza nell'anno 2021, dunque ho deciso di sceglierlo come modello per poter fare una previsione sulle miglia annuali percorse in USA.

Il metodo prevede una crescita delle miglia percorse nell'anno 2022; per quanto riguarda la stagionalità, è presente un calo iniziale di circa 10000 milioni di unità, seguito poi da una crescita più o meno costante fino al mese di maggio, dove poi ci sarà nuovamente un alternarsi di crescita e decrescita fino alla fine dell'anno.



Appendice

Codice Analisi

```
# importo tabella

library(readr)
data = read.csv("tabella.csv", header = TRUE, sep = ",", stringsAsFactors = F)
colnames(data) <- c("Data", "Miglia")
head(data)
summary(data)

# creo time series
mi = ts(data$Miglia, frequency = 12, start = 2000, end = c(2021, 12))
plot(mi, xlab="Anni", ylab="Miglia (in milioni di unità)")

mi = na.omit(mi)

# AUTOCORRELAZIONE
layout(t(1:2))
acf(mi, main="Autocorrelazione Miglia") # netta periodicità
m_mi = matrix(mi, 12,22)
ts.plot(scale(m_mi,scale=F),col=rainbow(22), main="Confronto stagionalità negli
anni")
lines(rowMeans(scale(m_mi,scale=F)),lwd=3,col="black")
layout(1)

# DECOMPOSIZIONE
plot(decompose(mi))
plot(decompose(mi, "m"))

# confrontiamo le stagionalità
mi.da=decompose(mi)
mi.dm=decompose(mi,type="multiplicative")

plot(mi.da$seasonal, main="Confronto stagionalità", xlab="Anno",
ylab="Stagionalità")
lines(mean(mi.dm$trend,na.rm=T)*(mi.dm$seasonal-1),col="red")

# rumore

plot(mi.da$random, main="Confronto rumore", xlab="Anno", ylab="Rumore")
lines(mean(mi.dm$trend,na.rm=TRUE)*(mi.dm$random-1),col="red")

# RESIDUI
# additivo
mi.dar=as.vector(window(mi.da$random,c(2000,7),c(2021,6)))
plot(mi.dar,pch=20, main="Residui additivi", ylab="Residui")

# moltiplicativo
mi.dmr=as.vector(window(mi.dm$random,c(2000,7),c(2021,6)))
plot(mi.dmr,pch=20, main="Residui moltiplicativi", ylab="Residui")
mi.dmr1=log(mi.dmr)

# varianza spiegata additivo
1-var(mi.dar)/var(window(mi,c(2000,7),c(2021,6)))
# varianza spiegata moltiplicativo
1-var(mi.dmr1)/var(window(log(mi),c(2000,7),c(2021,6)))

# analisi residui
layout(t(1:2))
# istogramma additivo
hist(mi.dar,20,freq=F, main="Residui additivi")
lines(density(mi.dar),col="blue")
lines(sort(mi.dar),dnorm(sort(mi.dar),mean(mi.dar),sd(mi.dar)),col="red")
# istogramma moltiplicativo
```

```

hist(mi.dmr1,20,freq=F, main="Residui moltiplicativi")
lines(density(mi.dmr1),col="blue")
lines(sort(mi.dmr1),dnorm(sort(mi.dmr1),mean(mi.dmr1),sd(mi.dmr1)),col="red")
# qq plot additivo
qqnorm(mi.dar, main="Residui additivi")
qqline(mi.dar)
# qq plot additivo
qqnorm(mi.dmr1, main="Residui moltiplicativi")
qqline(mi.dmr1)
layout(1)
# Modello additivo
shapiro.test(mi.dar)
# Modello moltiplicativo
shapiro.test(mi.dmr1)

layout(t(1:2))
# autocorrelazione additivo
acf(mi.dar, main="Autocorrelazione residui additivi")
# autocorrelazione moltiplicativo
acf(mi.dmr, main="Autocorrelazione residui moltiplicativi")
layout(1)

# deviazione standard additivo
sd(acf(mi.dar,plot=F)$acf)
# deviazione standard moltiplicativo
sd(acf(mi.dmr,plot=F)$acf)

# HOLT-WINTERS
mi.hwm=Holtwinters(mi, seasonal="additive")

mi.hwm$alpha
mi.hwm$beta
mi.hwm$gamma

layout(t(1:3))
ts.plot(mi,mi.hwm$fitted[,1],predict(mi.hwm,12),col=c("black","red","green3"),
main="Holt-winters", xlab="Anno", ylab="Miglia (Milioni)")
legend(
  "bottomleft",
  legend = c("Valore effettivo", "Holt-winters", "Previsione Holt-winters"),
  col = c("black", "red", "green3"),
  lwd = 2,
)
ts.plot(mi.da$trend,mi.hwm$fitted[,2],col=c("black","red"), main="Confronto con
trend decomposizione", xlab="Anno", ylab="Trend")
legend(
  "bottomleft",
  legend = c("Holt-winters", "Dec. Additiva"),
  col = c("black", "red"),
  lwd = 2,
)
ts.plot(mi.da$seasonal,mi.hwm$fitted[,4],col=c("black","red"), main="Confronto
con stagionalità decomposizione", xlab="Anno", ylab="Stagionalità")
legend(
  "bottomleft",
  legend = c("Holt-winters", "Dec. Additiva"),
  col = c("black", "red"),
  lwd = 2,
)

# METODI AUTOREGRESSIVI
# AUTOREGRESSIONE MANUALE
layout(t(1:2))
pacf(mi, main="Autocorrelazione parziale",40)

```

```

L=length(mi)
l=23
map=matrix(nrow=L-l, ncol=l+1)
for (i in 1:(l+1)) {
  map[,i] = mi[i:(L-l-1+i)]
}
map <- data.frame(map)
mi.lm = lm(X24 ~ ., data = map)
summary(mi.lm)

mi.lm = lm(X24 ~ X1+X10+X11+X12+X13+X22+X23, data = map)
summary(mi.lm)

pt = rep(0, L+12)
pt[1:L] = mi
for (i in 1:12) {
  #xL+i-(l-m+1)
  pt[L+i] = coef(mi.lm) %*% c(1, pt[L+i-23], pt[L+i-14], pt[L+i-13], pt[L+i-12], pt[L+i-11], pt[L+i-2], pt[L+i-1])
}

mi.lm.pt = ts(pt, frequency = 12, start = c(2000,1))
mi.lm.a = window(mi, c(2001,12))-resid(mi.lm)
ts.plot(mi, mi.lm.a, window(mi.lm.pt, c(2021,12)), col = c("black", "blue", "green3"))
legend(
  "bottomleft",
  legend = c("Valore effettivo", "Autoregressione", "Previsione autoregressione"),
  col = c("black", "blue", "green3"),
  lwd = 2,
)
layout(1)

# YULE-WALKER
mi.ar = ar(mi)
mi.ar
1-mi.ar$var/var(mi[24:length(mi)])

mi.ar.p = predict(mi.ar, n.ahead = 12, se.fit=FALSE)

link.ar = rep(0,2)
link.ar[1] = mi[length(mi)]-mi.ar$resid[length(mi)]
link.ar[2] = mi.ar.p[1]

#layout(t(1:2))
ts.plot(mi, mi-mi.ar$resid, mi.ar.p, ts(link.ar, c(2021,12), frequency=12), col = c("black", "blue", "green3", "green3"), main="Autoregressione con Yule-walker")
legend(
  "bottomleft",
  legend = c("Valore effettivo", "Yule-walker", "Previsione Yule-walker"),
  col = c("black", "blue", "green3"),
  lwd = 2,
)

# MINIMI QUADRATI
mi.ls = ar(mi, method = "ols")
mi.ls$order
1-var(na.omit(mi.ls$resid))/var(mi[24:length(mi)])

mi.ls.p = predict(mi.ls, n.ahead = 12, se.fit = FALSE)

link.ls = rep(0,2)
link.ls[1] = mi[length(mi)]-mi.ls$resid[length(mi)]

```

```

link.ls[2] = mi.ls.p[1]

ts.plot(mi, mi - mi.ls$resid, mi.ls.p, ts(link.ls, c(2021,12), frequency=12),
col = c("black", "blue", "green3", "green3"), main="Autoregressione con minimi
quadrati")
legend(
  "bottomleft",
  legend = c("Valore effettivo", "Minimi quadrati", "Previsione Minimi
quadrati"),
  col = c("black", "blue", "green3"),
  lwd = 2,
)
layout(1)

# CONFRONTO MODELLI
# AUTOVALUTAZIONE
years = c(2021)
res.hwm=rep(0,11)
res.ar=rep(0,11)
res.lm=rep(0,11)
res.ls=rep(0,11)
mi.hwm.p=rep(0,11)
train.lm = lm(X24 ~ X1+X10+X11+X12+X13+X22+X23, data = map)
for (i in 1:11) {
  train_year = (i-1)%/%12 + 1
  train_period = i%%12
  if (train_period == 0) {
    train_period = 12
  }
  test_year = i%%12 + 1
  test_period = i%%12+1

  train = window(mi, end = c(years[train_year], train_period))
  test = window(mi, start = c(years[test_year], test_period))

  L = length(train)

  pt = rep(0, L)
  pt[1:L] = train[1:L]

  res.hwm[i] = predict(Holtwinters(train),1)
  res.ar[i] = predict(ar(train), n.ahead = 1, se.fit = F)
  res.ls[i] = predict(ar(train, method = "ols"), n.ahead = 1, se.fit = F)
  res.lm[i] = coef(train.lm) %*% c(1, pt[L-22], pt[L-13], pt[L-12], pt[L-11],
pt[L-10], pt[L-1], pt[L])
}
test = window(mi, start = c(2021,2))
sqrt(mean((test - res.hwm)^2))
sqrt(mean((test - res.ar)^2))
sqrt(mean((test - res.lm)^2))
sqrt(mean((test - res.ls)^2))
ts.plot(test, res.hwm, res.ar, res.lm, res.ls, col = c("black", "blue", "red",
"green3", "purple"), main="Confronto accuratezza modelli", xlab="Anno",
ylab="Migliaia (Milioni)")
legend(
  "bottomright",
  legend = c("Test", "Holt-winters", "Yule-walker", "Autoregressione", "Minimi
quadrati"),
  col = c("black", "blue", "red", "green3", "purple"),
  lwd = 2,
  cex=0.75
)

#prova
years = c(2020)

```

```

res.hwm=rep(0,11)
res.ar=rep(0,11)
res.lm=rep(0,11)
res.ls=rep(0,11)
mi.hwm.p=rep(0,11)
train.lm = lm(X24 ~ X1+X10+X11+X12+X13+X22+X23, data = map)
for (i in 1:11) {

  train_year = (i-1)%/%12 + 1
  train_period = i%%12
  if (train_period == 0) {
    train_period = 12
  }
  test_year = i%%12 + 1
  test_period = i%%12+1

  train = window(mi, end = c(years[train_year], train_period))
  test = window(mi, start = c(years[test_year], test_period))

  L = length(train)

  pt = rep(0, L)
  pt[1:L] = train[1:L]

  res.hwm[i] = predict(Holtwinters(train),1)
  res.ar[i] = predict(ar(train), n.ahead = 1, se.fit = F)
  res.ls[i] = predict(ar(train, method = "ols"), n.ahead = 1, se.fit = F)
  res.lm[i] = coef(train.lm) %*% c(1, pt[L-22], pt[L-13], pt[L-12], pt[L-11],
pt[L-10], pt[L-1], pt[L])
}
test = window(mi, start = c(2020,2), end=c(2020,12))
sqrt(mean((test - res.hwm)^2))
sqrt(mean((test - res.ar)^2))
sqrt(mean((test - res.lm)^2))
sqrt(mean((test - res.ls)^2))
ts.plot(test, res.hwm, res.ar, res.lm, res.ls, col = c("black", "blue", "red",
"green3", "purple"), main="Confronto accuratezza modelli", xlab="Anno",
ylab="Migliaia (Milioni)")
legend(
  "bottomright",
  legend = c("Test", "Holt-winters", "Yule-walker", "Autoregressione", "Minimi
quadrati"),
  col = c("black", "blue", "red", "green3", "purple"),
  lwd = 2,
  cex=0.75
)

#prova
years = c(2019)
res.hwm=rep(0,11)
res.ar=rep(0,11)
res.lm=rep(0,11)
res.ls=rep(0,11)
mi.hwm.p=rep(0,11)
train.lm = lm(X24 ~ X1+X10+X11+X12+X13+X22+X23, data = map)
for (i in 1:11) {

  train_year = (i-1)%/%12 + 1
  train_period = i%%12
  if (train_period == 0) {
    train_period = 12
  }
  test_year = i%%12 + 1
  test_period = i%%12+1

```



```

train = window(mi, end = c(years[train_year], train_period))
test = window(mi, start = c(years[test_year], test_period))

L = length(train)

pt = rep(0, L)
pt[1:L] = train[1:L]

res.hwm[i] = predict(Holtwinters(train),1)
res.ar[i] = predict(ar(train), n.ahead = 1, se.fit = F)
res.ls[i] = predict(ar(train, method = "ols"), n.ahead = 1, se.fit = F)
res.lm[i] = coef(train.lm) %*% c(1, pt[L-22], pt[L-13], pt[L-12], pt[L-11],
pt[L-10], pt[L-1], pt[L])
}
test = window(mi, start = c(2019,2), end=c(2019,12))
sqrt(mean((test - res.hwm)^2))
sqrt(mean((test - res.ar)^2))
sqrt(mean((test - res.lm)^2))
sqrt(mean((test - res.ls)^2))
ts.plot(test, res.hwm, res.ar, res.lm, res.ls, col = c("black", "blue", "red",
"green3", "purple"), main="Confronto accuratezza modelli", xlab="Anno",
ylab="Migliaia (Milioni)")
legend(
  "bottomright",
  legend = c("Test", "Holt-winters", "Yule-walker", "Autoregressione", "Minimi
quadrati"),
  col = c("black", "blue", "red", "green3", "purple"),
  lwd = 2,
  cex=0.75
)

# RESIDUI
layout(t(1:4))
# estrazione dei residui
mi.hwm.r=resid(mi.hwm)
mi.lm.r=resid(mi.lm)
mi.ls.r = as.double(na.omit(mi.ls$resid))
mi.ls.fitted = as.double(na.omit(mi - mi.ls$resid))
mi.ar.r = as.double(na.omit(mi.ar$resid))

# rappresentazione grafica rispetto al tempo
plot(mi.hwm.r, type = "p", pch = 20, main="Residui Holt-winters", xlab="Anno",
ylab="Residui")
plot(mi.lm.r, type = "p", pch = 20, main="Residui autoregressione manuale",
ylab="Residui")
plot(mi.ar.r, pch = 20, main="Residui Yule-walker", ylab="Residui")
plot(mi.ls.r, pch = 20, main="Residui minimi quadrati", ylab="Residui")
# densità empiriche
hist(mi.hwm.r, 20, freq = F, main="Istogramma residui Holt-winters")
lines(density(mi.hwm.r),col="blue")
lines(sort(mi.hwm.r), dnorm(sort(mi.hwm.r), mean(mi.hwm.r), sd(mi.hwm.r)), col
= "red")
hist(mi.lm.r, 20, freq = F, main="Istogramma residui autoregressione manuale")
lines(density(mi.lm.r),col="blue")
lines(sort(mi.lm.r), dnorm(sort(mi.lm.r), mean(mi.lm.r), sd(mi.lm.r)), col =
"red")
hist(mi.ls.r, 20, freq = F, main="Istogramma residui Yule-walker")
lines(density(mi.ar.r), col = "blue")
lines(sort(mi.ar.r), dnorm(sort(mi.ar.r), mean(mi.ar.r), sd(mi.ar.r)), col =
"red")
hist(mi.ls.r, 20, freq = F, main="Istogramma residui minimi quadrati")
lines(density(mi.ls.r), col = "blue")
lines(sort(mi.ls.r), dnorm(sort(mi.ls.r), mean(mi.ls.r), sd(mi.ls.r)), col =
"red")
# grafico quantile-quantile

```

```

qqnorm(mi.hwm.r, pch = 20, main="Q-Q Plot residui Holt-winters")
qqline(mi.hwm.r)
qqnorm(mi.lm.r, pch = 20, main="Q-Q Plot residui autoregressione manuale")
qqline(mi.lm.r)
qqnorm(mi.ar.r, pch = 20, main="Q-Q Plot residui Yule-walker")
qqline(mi.ar.r)
qqnorm(mi.ls.r, pch = 20, main="Q-Q Plot residui minimi quadrati")
qqline(mi.ls.r)
# autocorrelazione
acf(mi.hwm.r, main="Autocorrelazione residui Holt-winters")
acf(mi.lm.r, main="Autocorrelazione residui autoregressione")
acf(mi.ar.r, main="Autocorrelazione residui Yule-walker")
acf(mi.ls.r, main="Autocorrelazione residui minimi quadrati")
# test Shapiro
# Holt-winters
shapiro.test(mi.hwm.r)
# Autoregressione manuale
shapiro.test(mi.lm.r)
# Yule-walker
shapiro.test(mi.ar.r)
# Minimi quadrati
shapiro.test(mi.ls.r)
layout(1)

#CONCLUSIONE
ts.plot(predict(mi.hwm,12), col="red", main = "Previsione", xlab="Anno",
ylab="Miglia (Milioni)", ylim=c(230000, 300000))
lines(predict(mi.hwm,12)+quantile(mi.hwm.r,0.05),col="green3")
lines(predict(mi.hwm,12)+quantile(mi.hwm.r,0.95),col="green3")

```