Serie Temporali

Analisi Dati e Statistica, 2025–26

Paolo Bosetti
Università di Trento, Dipartimento di Ingegneria Industriale

Ultimo aggiornamento: 17/06/2026

Indice

```
options(width = 60)
set.seed(0)
library(latex2exp)
library(ggpubr)
```

```
Loading required package: ggplot2
```

```
library(glue)
library(tidyverse)
```

```
── Attaching core tidyverse packages ──── tidyverse
2.0.0 ──
✔ dplyr     1.2.1     ✔ readr     2.2.0
✔ forcats   1.0.1     ✔ stringr   1.6.0
✔ lubridate 1.9.5     ✔ tibble    3.3.1
✔ purrr     1.2.2     ✔ tidyr     1.3.2
```

```
──          Conflicts          ────────────────────
tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.
org/>) to force all conflicts to become errors
```

```
library(ggfortify)
library(modelr)
library(tsbox)
library(xts)
```

```
Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric


########################## Warning from 'xts' package
##########################
#
#
# The dplyr lag() function breaks how base R's lag()
function is supposed to  #
# work, which breaks lag(my_xts). Calls to lag(my_xts)
that you type or       #
# source() into this session won't work correctly.
#
#
#
# Use stats::lag() to make sure you're not using
dplyr::lag(), or you can add #
#   conflictRules('dplyr',   exclude   =   'lag')   to
your .Rprofile to stop           #
#  dplyr  from  breaking  base  R's  lag()  function.
#
#
#
# Code in packages is not affected. It's protected by
R's namespace mechanism #
# Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to
suppress this warning.  #
#
#
################################################################################
```

```
Attaching package: 'xts'

The following objects are masked from 'package:dplyr':

    first, last
```

```
library(astsa)
library(patchwork)
theme_set(theme_gray()+theme(legend.position        =
"bottom"))
```

1 Serie Temporali

Una **serie temporale** è costituita da una serie di osservazioni di una variabile aleatoria tale per cui l'influenza di un'osservazione sulle seguenti non possa essere trascurata e—quindi—tale che la dipendenza dal tempo risulti essenziale

1.1 Statistica delle serie temporali

Tutti i metodi di regressione visti fin ora sono basati sull'assunzione che la variabile aleatoria sia $x \overset{IID}{\sim} \mathcal{N}(\mu, \sigma^2)$. Cioè tutte le osservazioni **devono essere non-autocorrelate**

Quest'assunzione è, tra l'altro, alla base della raccomandazione di casualizzazione della sequenza operativa

Supponiamo di poter considerare una misura come un segnale tempo-dipendente. È evidente che riducendo l'intervallo di campionamento del segnale prima o poi ogni campione sarà correlato al precedente

Esiste quindi una **frequenza di campionamento massima** al di sopra della quale ogni misura risulta essere autocorrelata, cioè **le osservazioni di x non sono più IID**

Questa situazione sussiste quando la **dinamica** propria dello strumento di misura o del misurando stesso—che sono **sempre finite**—sono più lente dell'intervallo temporale in cui si effettuano le misure

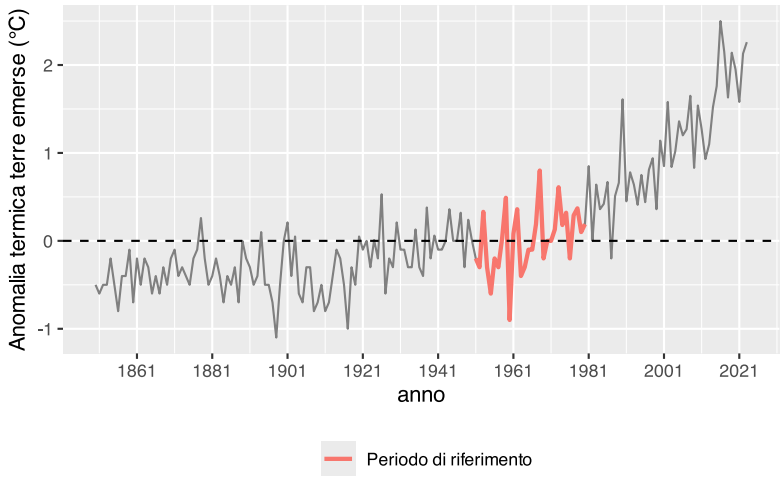
1.2 Statistica delle serie temporali

Consideriamo ad esempio la serie temporale in figura che riporta la differenza tra la temperatura media delle terre emerse e il corrispondente valore medio nel periodo 1951–1980

È evidente che osservazioni vicine sono più correlate di osservazioni lontane

Inoltre è evidente (**ed è di interesse**) valutare la dipendenza della v.a. considerata dal tempo allo scopo di **effettuare delle previsioni future**

```
ts <- ts_xts(gtemp_land)
ref <- ts["1951/1980"]
ggplot(ts, aes(x=Index, y=value)) +
  geom_line(color=gray(0.5)) +
   geom_line(data=ref, aes(y=value, color="Periodo di
riferimento"), linewidth=1) +
  # geom_point(shape=4) +
  geom_hline(yintercept=0, linetype=2) +
  scale_x_date(breaks="20 years", date_labels = "%Y") +
    labs(x="anno", y="Anomalia termica terre emerse
(°C)", color="") +
  theme(legend.position = "bottom")
```



1.3 Autocovarianza e autocorrelazione

Abbiamo visto come gli operatori **covarianza** e **correlazione** servano a stimare l'indipendenza di due campioni

Considerando un segnale tempo-dipendente $x = x(t)$, è interessante considerare la covarianza del segnale con se stesso, traslato nel tempo

Definiamo la **funzione autocovarianza** $\gamma(s,t)$ come la funzione che valuta la covarianza di un segnale temporale con se stesso valutato **iniziando** ai tempi s e t:

$$\gamma_x(s,t) = \sigma_{x_s,x_t} = E[(x_s - \mu_s)(x_t - \mu_t)]$$

È evidente che $\gamma_x(s,s) = \sigma^2(x_s)$

La **funzione di autocorrelazione** (ACF), di conseguenza, è definita come:

$$\rho_x(s,t) = \frac{\gamma_x(s,t)}{\sqrt{\gamma_x(s,s)\gamma_x(t,t)}}$$

ed ha il vantaggio di essere sempre compresa in $[-1,1]$. È inoltre evidente che $\rho_x(s,s) = 1$

1.4 Autocorrelazione

Se campioniamo un segnale continuo a intervalli fissi Δt per una durata complessiva T, otteniamo una **serie temporale finita** di $N = T/\Delta t$ osservazioni: $x_0 = \langle x_1, x_2, ..., x_N \rangle$

Possiamo estendere la definizione stabilendo che sia $s = t_0 = 0$ l'istante iniziale della serie e che sia $t = s + \tau$ un generico momento successivo tale per cui

$$\tau = \ell \Delta t$$

dove ℓ è il ritardo o *lag*, e allora l'**autocovarianza** e l'**autocorrelazione per una s.t. finita** risultano:

$$\gamma_x(\ell) = \frac{\sum_{i=1}^{N-\ell}(x_i - \bar{x}_0)(x_{i+\ell} - \bar{x}_\ell)}{N - \ell - 2}, \quad \rho_x(\ell) = \frac{\gamma_x(\ell)}{\sqrt{\sigma_x(0)\sigma_x(\ell)}}$$

dove

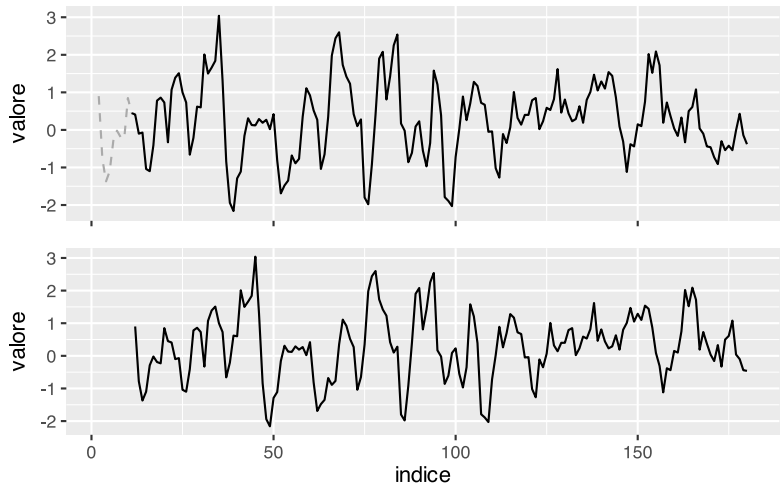
$$\bar{x}_0 = \frac{1}{N-\ell}\sum_{i=1}^{N-\ell} x_i, \quad \bar{x}_\ell = \frac{1}{N-\ell}\sum_{i=\ell}^{N} x_i$$

1.5 Autocorrelogramma (ACF)

```
ell <- 10
c <- ts_xts(chicken) %>% diff() %>% fortify() %>% mutate(i=1:n(), .before=Index) %>% filter(!is.na(value))

ps <- ggplot(c[ell:length(c$i),]) +
```

```
  geom_line(aes(x=i, y=value)) +
     geom_line(data=c[1:ell,],   aes(x=i,   y=value),
color=gray(2/3), linetype=2) +
  coord_cartesian(xlim=range(c$i)) +
  theme(axis.title.x = element_blank(), axis.text.x =
element_blank()) +
  labs(y="valore")
pt <- ggplot(lag(c, ell)) +
  geom_line(aes(x=i+ell, y=value)) +
  coord_cartesian(xlim=range(c$i)) +
  labs(x="indice", y="valore")
ggarrange(ps, pt, nrow=2, heights=c(1,1.08))
```

```
Warning: Removed 10 rows containing missing values or
values outside
the scale range (`geom_line()`).
```



Per costruire il grafico ACF di una s.t. si trasla l'ascissa di ℓ campioni trascurando i primi ℓ campioni nella s.t. non traslata e gli ultimi ℓ campioni in quella traslata

Poi si calcola l'autocorrelazione tra le due serie

Il processo viene ripetuto per $\ell = 0, 1, ..., n$, con n scelto in funzione della dimensione della serie, tipicamente pari ad almeno 50 e comunque non oltre la metà della lunghezza della s.t.

1.6 Autocorrelogramma (ACF)

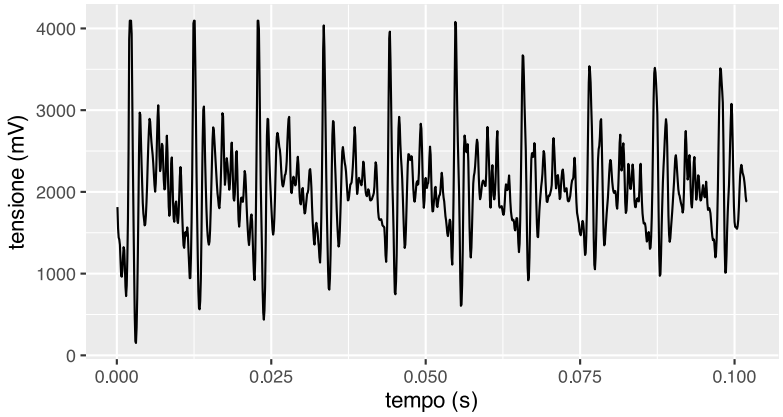
La **funzione di autocorrelazione** $\rho_x(\ell)$ che è una funzione **a valori discreti**, può essere messa in grafico per studiare il *lag* massimo al di sopra del quale la serie storica x non è più autocorrelata

In figura il segnale di un microfono che registra il suono "AAHH". La serie è **evidentemente periodica** ogni 0.01 s

L'autocorrelogramma mostra **autocorrelazione elevata** fino a $\ell = 4$. Poi l'andamento è **periodico**, a confermare che la s.t. è autocorrelata con se stessa ogni circa 10 lag

1.6.a Serie storica

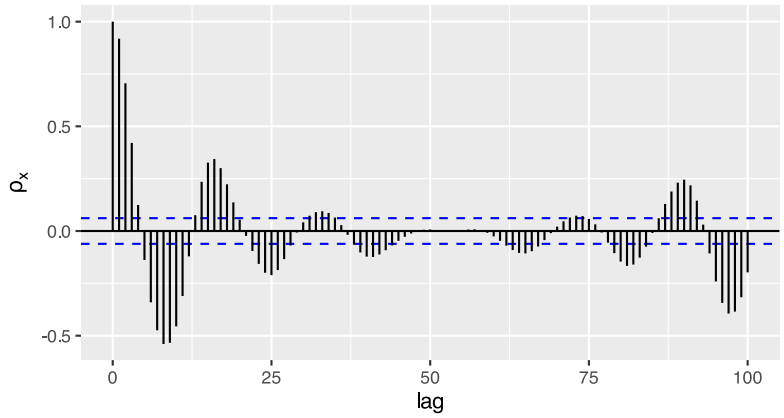
```
s <- ts_data.frame(speech) %>% slice(seq(1,n(),by=1))
%>%
  mutate(time=seq_along(time)/10000)
s %>%
ggplot(aes(x=time, y=value)) +
  geom_line() +
  labs(x="tempo (s)", y="tensione (mV)")
```



1.6.b ACF

```
with(acf(s$value,    lag.max    =    100,    plot=F),
data.frame(lag, acf)) %>%
  ggplot(aes(x=lag, y=acf)) +
  geom_hline(aes(yintercept=0)) +
   geom_hline(yintercept=c(-1,1)*qnorm((1 + 0.95)/2)/
sqrt(1020),
             linetype=2, color="blue") +
```

```
  geom_segment(mapping = aes(xend = lag, yend = 0)) +
  labs(x="lag", y=TeX("$\\rho_x$"))
```



ACF sta per *Auto-Correlation Function*; la banda evidenziata in blu è l'intervallo di confidenza al 95%. Inoltre, in generale vale sempre $\rho_x(0) = 1$

1.7 Autocorrelogramma (ACF)

Consideriamo gli stessi dati della s.t. precedente, ma campionati in istanti casuali

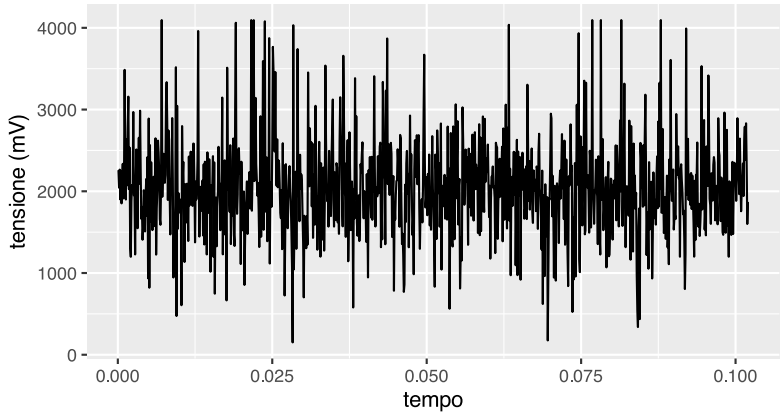
Allora $y \overset{IID}{\sim} \mathcal{N}(\mu, \sigma^2)$ e quindi $\rho_y(\ell) = 0 \; \forall \ell > 0$

Come atteso, l'unico valore della ACF fuori dall'initervallo di confidenza è $\rho_x(0)$

In questo caso si dice anche che la s.t. è un ***random walk***

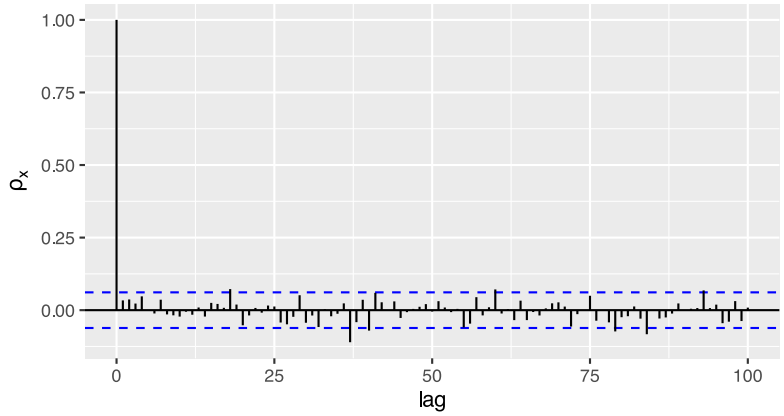
1.7.a Serie storica

```
s <- s %>% mutate(value = sample(value))
s %>%
ggplot(aes(x=time, y=value)) +
  geom_line() +
  labs(x="tempo", y="tensione (mV)")
```



1.7.b ACF

```
with(acf(s$value,    lag.max    =    100,    plot=F),
data.frame(lag, acf)) %>%
  ggplot(aes(x=lag, y=acf)) +
  geom_hline(aes(yintercept=0)) +
   geom_hline(yintercept=c(-1,1)*qnorm((1 + 0.95)/2)/
sqrt(1020),
             linetype=2, color="blue") +
  geom_segment(mapping = aes(xend = lag, yend = 0)) +
  labs(x="lag", y=TeX("$\\rho_x$"))
```



1.8 Serie temporali stazionarie

Una s.t. può essere stazionaria o meno. Si definiscono:

Serie temporale stazionaria in senso ampio

È una serie temporale per cui il comportamento probabilistico di una qualsiasi collezione di valori $\langle x_{t_1}, x_{t_2}, ..., x_{t_k} \rangle$ è identico a quello della collezione traslata $\langle x_{t_1+h}, x_{t_2+h}, ..., x_{t_k+h} \rangle$, cioè:

$$\Pr(x_{t_1} \leq c_1, ..., x_{t_k} \leq c_k) = \Pr(x_{t_1+h} \leq c_1, ..., x_{t_k+h} \leq c_k).$$

Serie temporale stazionaria in senso stretto

È una serie temporale per cui il valor medio della serie temporale è costante (tempo-indipendente) e la funzione di autocovarianza $\gamma(s,t)$ dipende da s e t solo tramite la loro differenza $|\ s-t\ |$

Per una **serie stazionaria in senso ampio** si può assumere $\sigma_x(0) = \sigma_x(\ell) = \sigma_x$ e $\bar{x}_0 = \bar{x}_\ell$ e, quindi, $\rho_x(\ell) = \gamma_x(\ell)/\sigma_x$

1.9 Stabilizzazione

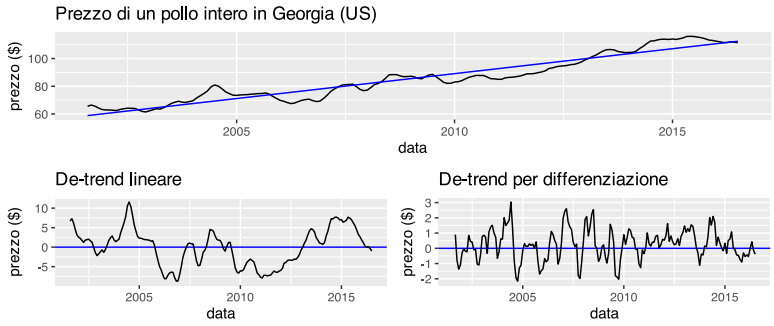
- Le ST stazionarie almeno in senso ampio sono più semplici da trattare
- È quindi utile cercare di **stabilizzare** la ST separandola in un termine di **tendenza** (*trend*) x_t più un termine **stazionario** x_s
- la stabilizzazione può essere fatta in due modi:
  - *detrending* mediante regressione lineare: $x_t = x_{l,t} + x_{s,t}$, dove $x_{l,t} = a + bt$ e, quindi, $x_{s,t}$ risulta essere la **serie dei residui** della regressione lineare di x_t
  - *detrending* per differenziazione
- La ST stabilizzata può poi essere analizzata e quindi ri-trasformata mediante l'operazione inversa:
  - somma del termine di tendenza
  - integrazione (cioè somma cumulativa)

1.10 Stabilizzazione (esempio)

```
m <- lm(chicken~time(chicken))
f <- xts(data.frame(
    line = predict(m),
    res = residuals(m)
  ), order.by=as.Date(time(chicken)))
p1 <- ggplot(ts_xts(chicken)) +
  geom_line(aes(x=Index, y=value)) +
 geom_line(aes(x=Index, y=line), data=f, col="blue") +
   labs(title="Prezzo di un pollo intero in Georgia
(US)", x="data", y="prezzo ($)")
```

```
p2 <- ggplot(f) +
  geom_hline(yintercept = 0, color="blue") +
  geom_line(aes(x=Index, y=res)) +
   labs(title="De-trend lineare", x="data", y="prezzo
($)")

p3 <- ggplot(diff(ts_xts(chicken))) +
  geom_hline(yintercept = 0, color="blue") +
  geom_line(aes(x=Index, y=value)) +
      labs(title="De-trend   per   differenziazione",
x="data", y="prezzo ($)")

p1 / (p2 + p3)
```

```
Warning: Removed 1 row containing missing values or
values outside
the scale range (`geom_line()`).
```



1.11 Operatori di differenziazione

In generale, la stabilizzazione per differenziazione dà risultati migliori ed è anche più pratica: se la ST differenziata non è stabile, è possibile aumentare l'ordine di differenziazione fino a raggiungere la stabilità

Come si differenzia una ST?

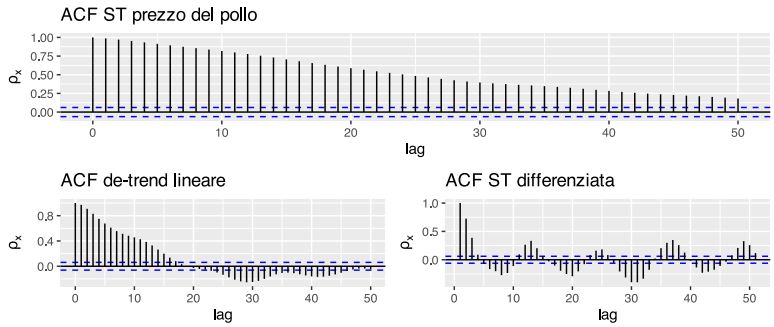
Si definiscono:

- **Operatore *backshift***: è l'operatore B^n tale per cui $B^n x_t := x_{t-n}$
- **Operatore differenza**: è l'operatore ∇ tale per cui $\nabla x_t := x_t - x_{t-1} = (1-B)x_t$. Risulta quindi che $\nabla^d x_t = (1-B)^d x_t$, e quindi ad esempio $\nabla^2 x_t = (1-B)^2 x_t = x_t - 2x_{t-1} + x_{t-2}$

Quindi ad esempio la differenziazione $\nabla^2 x_t$ è l'equivalente discreto della derivata seconda $\frac{d^2}{dt^2}x(t)$ per la funzione continua $x(t)$

1.12 Stabilizzazione e ACF

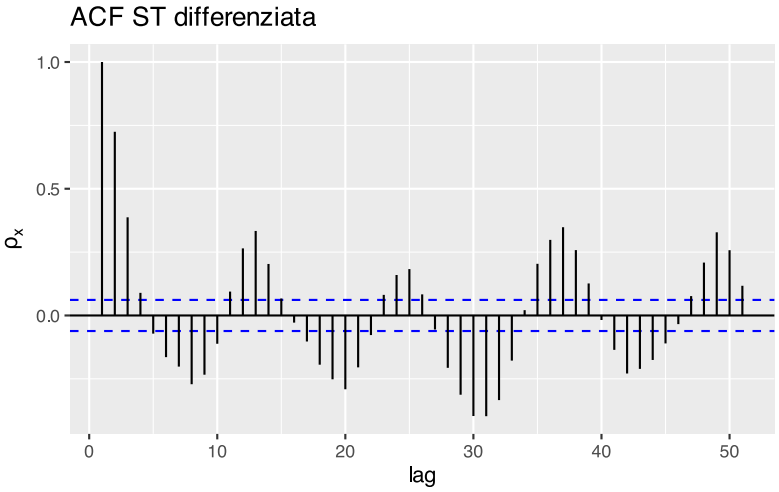
```
p1 <- with(acf(as.numeric(chicken), lag.max = 50,
plot=F), data.frame(lag, acf)) %>%
  ggplot(aes(x=lag, y=acf)) +
  geom_hline(aes(yintercept=0)) +
   geom_hline(yintercept=c(-1,1)*qnorm((1 + 0.95)/2)/
sqrt(1020),
              linetype=2, color="blue") +
  geom_segment(mapping = aes(xend = lag, yend = 0)) +
    labs(title="ACF ST prezzo del pollo", x="lag",
y=TeX("$\\rho_x$"))

p2 <- with(acf(f$res, lag.max = 50, plot=F),
data.frame(lag, acf)) %>%
  ggplot(aes(x=lag, y=acf)) +
  geom_hline(aes(yintercept=0)) +
   geom_hline(yintercept=c(-1,1)*qnorm((1 + 0.95)/2)/
sqrt(1020),
              linetype=2, color="blue") +
  geom_segment(mapping = aes(xend = lag, yend = 0)) +
  labs(title="ACF de-trend lineare", x="lag", y=TeX("$
\\rho_x$"))

p3 <- with(acf(diff(chicken), lag.max = 50, plot=F),
data.frame(lag=1:51, acf)) %>%
  ggplot(aes(x=lag, y=acf)) +
  geom_hline(aes(yintercept=0)) +
   geom_hline(yintercept=c(-1,1)*qnorm((1 + 0.95)/2)/
sqrt(1020),
              linetype=2, color="blue") +
  geom_segment(mapping = aes(xend = lag, yend = 0)) +
  labs(title="ACF ST differenziata", x="lag", y=TeX("$
\\rho_x$"))

p1 / (p2 + p3)
```



1.13 Stabilizzazione e ACF

- La ACF di una serie non stabilizzata mostra sempre una correlazione anche a *lag* elevati
- La ACF della serie dei residui di una regressione lineare decresce più rapidamente, ma mantiene comunque una correlazione anche a *lag* elevati
- La ACF della serie differenziata, inoltre, si smorza molto rapidamente (i *lag* 1, 2, 3, … dovrebbero essere esponenziali), dopodiché mostra oscillazioni armoniche, indice di una **periodicità** nella ST originale

```
p3
```



2 Modelli statistici

La regressione classica applicata alle serie temporali **è spesso insufficiente**. Ad esempio, nel caso del prezzo del pollo l'analisi

Possiamo quindi dire che la serie storica del prezzo del pollo mostra un *trend* che si **stabilizza al primo ordine** di differenziazione, ha una dinamica che mostra **autocorrelazione fino ai tre punti precedenti**, e un andamento **periodico** con periodo pari a 12 *lag*.

della autocorrelazione mostra un comportamento ciclico che la regressione classica non riesce ad evidenziare

È quindi necessario sviluppare delle tecniche che consentano di modellare i dettagli di una serie temporale, in particolare tenendo in considerazione anche **l'autocorrelazione** che può caratterizzare le serie temporali

2.1 Modelli AR — definizione

Un modello Auto-Regressivo (AR) esprime una determinata osservazione x_t al tempo t come combinazione lineare di p valori precedenti $x_{t-1}, x_{t-2}, ..., x_{t-p}$. Un modello AR di ordine p, abbreviato in $\mathrm{AR}(p)$, ha la forma: $$ x_t=\phi_1 x_{t-1} + \phi_2 x_{t-2}+ \dots+\phi_p x_{t-p} + w_t \label{eq:AR} $$

- x_t è stazionaria in senso ampio e $w_t \sim \mathcal{N}(0, \sigma_w^2)$
- $\phi_1, \phi_2, ..., \phi_p$ sono costanti e $\phi_p \neq 0$

Se la media di x_t non è nulla, si sostituisce x_t con $x_t - \mu$ per ottenere:

$$x_t - \mu = \phi_1(x_{t-1} - \mu) + \phi_2(x_{t-2} - \mu) + ... + \phi_p(x_{t-p} - \mu) + w_t$$

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + ... + \phi_p x_{t-p} + w_t, \;\; \alpha = \mu(1 - \phi_1 - \phi_2 - ... - \phi_p)$$

Le w_t sono anche chiamate **innovazioni**, dato che sono l'unico contributo **nuovo** di un punto rispetto ai precedenti.

2.2 Modelli AR — regressione

Ricordando la definizione dell'**operatore *backshift***, la definizione di w_t può essere scritta come:

$$(1 - \phi_1 B - \phi_2 B^2 - ... - \phi_p B^p) x_t = w_t$$

o ancora più concisamente come:

$$\Phi_p(B) x_t = w_t$$

dove $\Phi_p(B) \coloneqq 1 - \phi_1 B - \phi_2 B^2 - ... - \phi_p B^p$ è detto **operatore autoregressivo**.

Effettuare la regressione di un modello $\mathrm{AR}(p)$ su una serie storica x_t significa quindi adattare il modello $\Phi_p(B)\hat{x_t} = w_t$ identificando i coefficienti di $\Phi_p(B)$ che minimizzano i residui quadratici medi, essendo i residui $\varepsilon_t = x_t - \hat{x_t} = x_t - \Phi_p^{-1}(B) w_t$ (ammesso che esista l'inversa $\Phi_p^{-1}(B)$).

2.3 Modelli MA — definizione

Alternativamente, è possibile immaginare il caso in cui la generica osservazione x_t è espressa come combinazione lineare **del**

disturbo agente sulle q osservazioni precedenti. Un modello MA di ordine q, abbreviato come $\mathrm{MA}(q)$ è definito come

$$x_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + ... + \theta_q w_{t-q}$$

- $w_t \sim \mathcal{N}(0, \sigma_w^2)$
- $\theta_1, \theta_2, ..., \theta_q$ sono parametri costanti con $\theta_q \neq 0$

2.4 Modelli MA — regressione

Analogamente al caso $\mathrm{AR}(p)$, per $\mathrm{MA}(q)$ è possibile definire l'**operatore media mobile** di ordine q:

$$\Theta_q(B) = (1 + \theta_1 B + \theta_2 B^2 + ... + \theta_q B^q)$$

tale per cui la equazione per x_t può essere scritta come:

$$x_t = \Theta_q(B) w_t$$

Come sopra, regredire un modello $\mathrm{MA}(q)$ ad una serie storica x_t significa identificare i termini di $\Theta_q(B)$ che minimizzano i residui quadratici medi, definiti come $\varepsilon_t = x_t - \hat{x_t} = x_t - \Theta_q(B)$ (si noti che questa volta non c'è l'inversa!)

2.5 Modelli MA e ACF

```
set.seed(0)
tsma <- arima.sim(model=list(ma=c(-0.5, 0.7, -0.75)),
n=200)
```

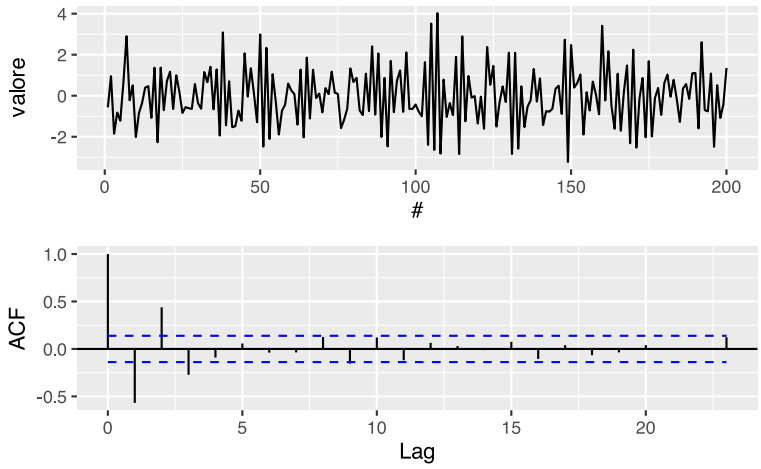
Una ST $MA(q)$ ha una memoria che si estende fino al *lag* q, nel senso che le innovazioni a distanze superiori a q non hanno alcun effetto sull'ultima osservazione

Quindi, data una serie temporale di tipo $MA(q)$ si può spesso identificare l'ordine dalla sua ACF, contando i picchi dopo quello a *lag* 0:

- tre picchi fuori dalla banda di confidenza significano un modello $MA(3)$
- i segni dei picchi corrispondono ai segni dei coefficienti

```
autoplot(tsma, xlab="#", ylab="valore") /
(autoplot(acf(tsma, plot=F)) +
  geom_hline(yintercept=0))
```

```
Warning: `aes_string()` was deprecated in ggplot2
3.0.0.
ℹ Please use tidy evaluation idioms with `aes()`.
ℹ See also `vignette("ggplot2-in-packages")` for more
  information.
ℹ The deprecated feature was likely used in the
ggfortify
  package.
  Please report the issue at
  <https://github.com/sinhrks/ggfortify/issues>.
```



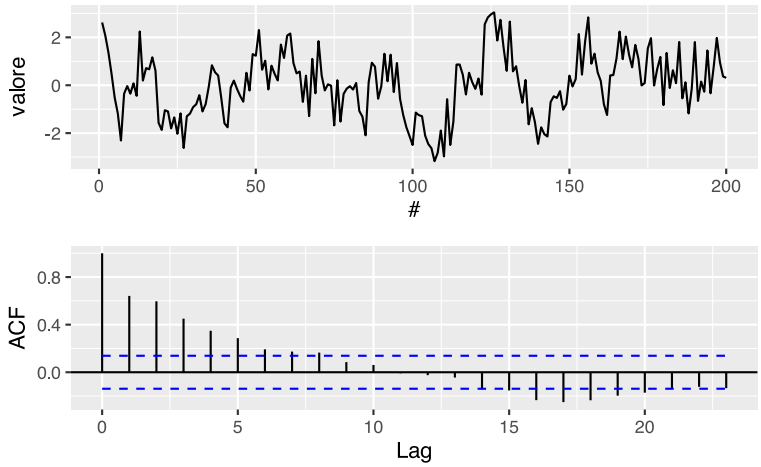
2.6 Modelli MA e ACF

```
set.seed(0)
tsar <- arima.sim(model=list(ar=c(0.5, 0.3)), n=200)
```

Se invece la serie temporale è di tipo $AR(p)$, ogni osservazione dipende dall'innovazione e da tutte le osservazioni precedenti, in modo ricorsivo

In questo caso l'autocorrelogramma riporterà un decadimento esponenziale seguito eventualmente da oscillazioni armoniche

```
autoplot(tsar, xlab="#", ylab="valore") /
(autoplot(acf(tsar, plot=F)) +
  geom_hline(yintercept=0))
```



2.7 Modelli AR e PACF

Per i modelli di tipo $AR(p)$ è comunque possibile identificare l'ordine mediante la **funzione di autocorrelazione parziale** o PACF, così definita:

$$\begin{aligned}\text{PACF}_1 &= \text{ACF}(z_{t+1}, z_t)\\ \text{PACF}_k &= \text{ACF}(z_{t+k} - \widehat{z_{t+k}}, z_t - \hat{z}_t), \quad k \geq 2\end{aligned}$$

in cui $\widehat{z_{t+k}}$ e $\hat{z}_t$ sono combinazioni lineari di $\{z_{t+1}, z_{t+2}, ..., z_{t+k-1}\}$ che minimizzano l'errore quadratico medio di z_{t+k} e z_t, rispettivamente

2.8 Modelli AR e PACF

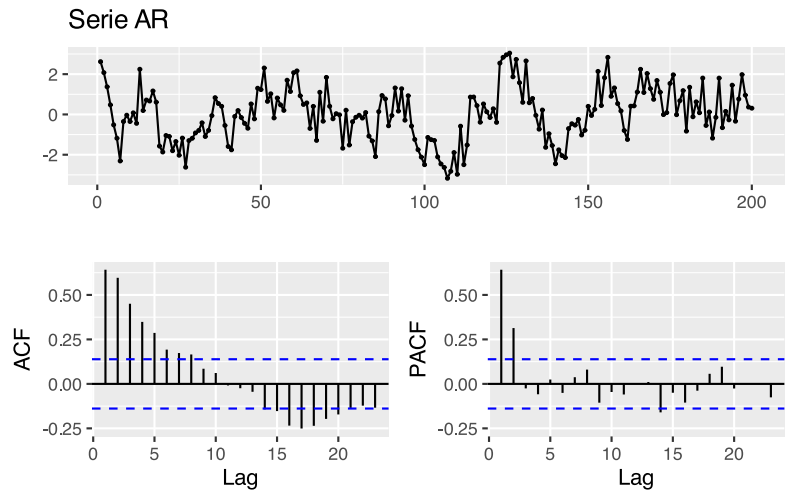
In generale, quindi, se la ACF mostra una memoria infinita (modello AR) e la PACF mostra pochi picchi, il numero di picchi è l'ordine del modello AR

Attenzione: non si considerano i picchi dopo il primo *cut-off*, cioè il *lag* in corrispondenza del quale l'autocorrelazione scende sotto il limite di confidenza per la prima volta

```
forecast::ggtsdisplay(tsar, main="Serie AR")
```

```
Registered S3 methods overwritten by 'forecast':
  method                 from
  autoplot.Arima         ggfortify
  autoplot.acf           ggfortify
  autoplot.ar            ggfortify
  autoplot.bats          ggfortify
```

```
autoplot.decomposed.ts ggfortify
autoplot.ets           ggfortify
autoplot.forecast      ggfortify
autoplot.stl           ggfortify
autoplot.ts            ggfortify
fitted.ar              ggfortify
fortify.ts             ggfortify
residuals.ar           ggfortify
```



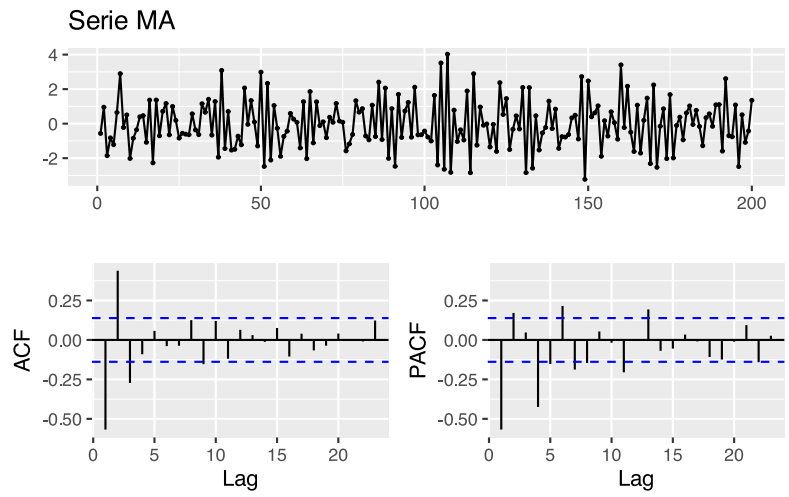
Attenzione: in questi grafici ACF e PACF il *lag* comincia da 1!

2.9 Modelli MA e PACF

Confrontando sia ACF che PACF

- puro rumore: ACF e PACF sono nulle per *lag* maggiore di 0
- $AR(p)$: ACF decresce lentamente, PACF non-nulla per *lag* minori o uguali a p, nulla altrimenti
- $MA(q)$: ACF mostra q picchi; se $PACF(1) > 0$, PACF oscilla a 0, altrimenti decade geometricamente a 0

```
forecast::ggtsdisplay(tsma, main="Serie MA")
```



Spesso, in pratica queste indicazioni sono difficili da riscontrare ed è quindi difficile individuare p, q in maniera certa e univoca

2.10 Modelli ARMA

L'ovvia estensione risulta dalla combinazione dei modelli $\mathrm{AR}(p)$ e $\mathrm{MA}(q)$

Un modello ARMA di ordine p, q, abbreviato come $\mathrm{ARMA}(p, q)$ è definito come:

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + ... + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + ... + \theta_q w_{t-q}$$

con ϕ_p e θ_q non nulli, ovvero, più brevemente:

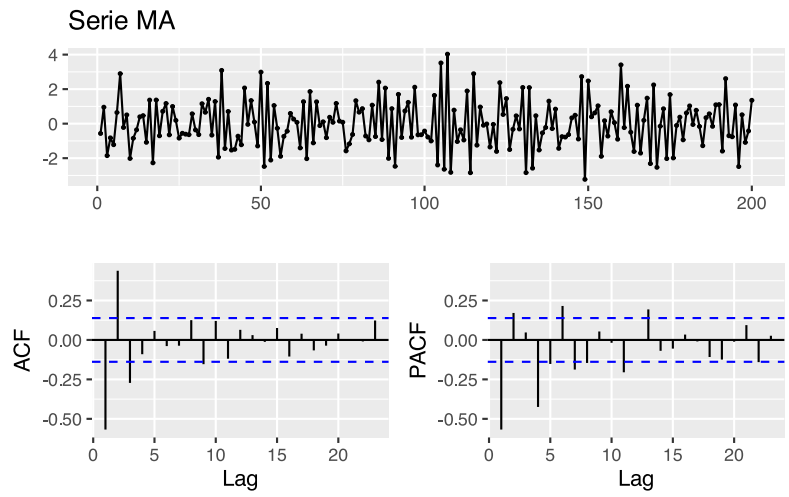
$$\Phi_p(B) x_t = \Theta_q(B) w_t$$

2.11 Modelli MA e PACF

Confrontando sia ACF che PACF

- puro rumore: ACF e PACF sono nulle per *lag* maggiore di 0
- $AR(p)$: ACF decresce lentamente, PACF non-nulla per *lag* minori o uguali a p, nulla altrimenti
- $MA(q)$: ACF mostra q picchi; se $PACF(1) > 0$, PACF oscilla a 0, altrimenti decade geometricamente a 0
- $ARMA(p, q)$: ACF mostra q picchi; PACF decade geometricamente a 0 solo dopo *lag* p

```
forecast::ggtsdisplay(tsma, main="Serie MA")
```



Spesso, in pratica queste indicazioni sono difficili da riscontrare ed è quindi difficile individuare p, q in maniera certa e univoca

2.12 Modelli ARIMA

I processi $\mathrm{ARMA}(p,q)$ sono adatti solo a descrivere serie **stazionarie** in senso ampio

Abbiamo visto però che un processo non stazionario può essere reso tale per differenziazione di un opportuno grado d

Un processo x_t è detto $\mathrm{ARIMA}(p,d,q)$ quando $\nabla^d x_t = (1-B)^d x_t$ 2è un processo $\mathrm{ARMA}(p,q)$

In generale, quando $E(\nabla^d x_t) = 0$ il processo $\mathrm{ARIMA}(p,d,q)$ può essere scritto come:

$$\Phi_p(B)\nabla^d x_t = \Theta_q(B) w_t$$

Se invece il valore atteso $E(\nabla^d x_t) = \mu$, allora:

$$\Phi_p(B)\nabla^d x_t = \delta + \Theta_q(B) w_t, \quad \delta = \mu(1-\phi_1-\ldots-\phi_p)$$

2.13 Stabilità e unicità dei modelli

Consideriamo un modello $\mathrm{AR}(1)$: possiamo sviluppare la formula ricorsiva come:

$$\begin{aligned} x_t &= \phi x_{t-1} + w_t = \phi(\phi x_{t-2} + w_{t-1}) + w_t \\ &= \sum_{j=0}^{+\infty} \phi^j w_{t-j} \end{aligned}$$

È quindi evidente che la serie temporale x_t **è stabile solo se** $\mid \phi \mid < 1$

Nel caso generale del modello $\mathrm{AR}(p)$, si ha la condizione di stabilità:

$$\left|\sum_{i=1}^{p} \phi_i\right| < 1$$

Un modello AR non stabile è detto anche **anti-causale**, perché si può dimostrare che per essere stabilizzato richiede la conoscenza delle osservazioni future

2.14 Stabilità e unicità dei modelli

Per un modello $\mathrm{MA}(1)$, invece, consideriamo due modelli:

$$x_t = w_t + \theta w_{t-1}, w_t \sim \mathcal{N}(0,1)$$
$$y_t = \nu_t + 1/\theta \nu_{t-1}, \nu_t \sim \mathcal{N}(0,\theta^2)$$

È evidente che x_t e ν_t hanno la stessa ACF e sono indistinguibili, dato che noi non conosciamo le innovazioni w_t e ν_t ma solo le due serie. Si può dimostrare che il risultato è generale (cioè non dipende dell'ordine q).

È quindi necessario scegliere una delle due forme alternative. Per individuare quale, riscriviamo la serie come $w_t = -\theta x_{t-1} + x_t$, che ha la stessa forma della $\mathrm{AR}(1)$. Possiamo quindi scegliere l'alternativa che rispetta lo stesso criterio di *stabilità*:

$$\left|\sum_{i=1}^{q} \theta_i\right| < 1$$

Tale alternativa si dice **invertibile**

2.15 Stabilità e unicità dei modelli

In generale, si può dimostrare che i criteri visti sopra corrispondono a imporre il requisito che le radici complesse dei polinomi

$$\Phi_p(z) = 1 - \phi_1 z - \phi_2 z^2 - \ldots - \phi_p z^p$$
$$\Theta_q(z) = 1 + \theta_1 z + \theta_2 z^2 + \ldots + \theta_q z^q$$

siano **tutte strettamente fuori dal cerchio unitario** sul piano complesso

Ad esempio: $\mathrm{ARMA}(2,2)$ con $\Phi_2(z) = 1 + 0.9z + 0.1z^2$ e $\Theta_q(z) = 1 + 2z + 15z^2$:

```
abs(polyroot(c(1, -0.9, -0.1)))
```

```
[1]  1 10
```

```
abs(polyroot(c(1, 2, 15)))
```

```
[1] 0.2581989 0.2581989
```

Cioè il termine AR non è causale e il termine MA non è invertibile (ma lo sarebbe $\theta_q(z) = 1 + 1/2z + 1/15z^2$)

2.16 Ridondanza

Consideriamo il processo $x_t = w_t$, con $w_t \sim \mathcal{N}(0, 1)$: si tratta ovviamente di puro rumore casuale

Moltiplichiamo entrambi i lati per $1 - 0.5B$ per ottenere:

$$x_t = 0.5x_{t-1} - 0.5w_{t-1} + w_t$$

che **sembra** un processo $\text{ARMA}(1, 1)$ ma ovviamente è sempre lo stesso rumore casuale. **Come distinguere questi casi?**

- si scompongono in fattori i polinomi Φ_p e Θ_q
- si eliminano i **fattori comuni**
- si sicompongono i fattori per ottenere il **modello non ridondante**

In R si può ancora usare `polyroot()`

2.17 Ridondanza

Ad esempio, consideriamo il processo $x_t = 0.4x_{t-1} + 0.45x_{t-2} + w_t + w_{t-1} + 0.25w_{t-2}$, che usando l'operatore B diventa:

$$(1 - 0.4B - 0.45B^2)x_t = (1 + B + 0.25B^2)w_t.$$

In questa forma il processo sembra $\text{ARMA}(2, 2)$, tuttavia possiamo scomporre i due polinomi in fattori, usando `polyroot()` per calcolare le radici:

```
# per Phi:
-1/polyroot(c(1, -0.4, -0.45))
```

```
[1] -0.9-6.887768e-21i  0.5+2.125854e-21i
```

```
# per Theta:
-1/polyroot(c(1, 1, 0.25))
```

```
[1] 0.5+0i 0.5+0i
```

Nota: se $z_i, i = 1, ..., n$ sono le radici del polinomio $p(z)$ di grado n, allora il polinomio può essere scomposto nei fattori $(1 - 1/z_1 z) \cdot (1 - 1/z_2 z) ... (1 - 1/z_n z)$.

2.18 Ridondanza

Come si vede, si può scrivere

$$\Phi_p(B) = (1 - 0.9B)(1 + 0.5B)$$
$$\Theta_q(B) = (1 + 0.5B)^2$$

Eliminando il fattore comune $(1 + 0.5B)$ otteniamo il modello

$$x_t = 0.9x_{t-1} + 0.5w_{t-1} + w_t$$

che è un ARMA$(1, 1)$.

Quindi ai criteri di **causalità** e di **invertibilità** si aggiunge il criterio di **non ridondanza** dei parametri, che si verifica eliminando ogni fattore comune dalla scomposizione in fattori dei polinomi $\Phi_p(B)$ e $\Theta_q(B)$.

2.19 Regressione (S)ARIMA

Per quanto detto sopra, un processo, o serie temporale, x_t può essere regredita mediante un modello ARIMA identificandone i parametri per **minimizzazione degli scarti quadratici**

Tuttavia, prima della regressione è necessario definire gli indici p, d, q adeguati, tali che non si abbia **né sotto- né sovra-adattamento**

In certi casi, inoltre, le serie sono **periodiche**: oltre ad un possibile **trend** sono soggette anche a ciclici andamenti oscillanti. In questi casi:

- si rende la ST stazionaria per differenziazione
- si separa un contributo a bassa frequenza, chiamato **stagionale**, da un contributo ad alta frequenza
- si effettua separatamente la regressione del contributo stagionale e del contributo non stagionale come due processi ARIMA distinti e sovrapposti: tale regressione si chiama *Seasonal ARIMA* o **SARIMA**

2.20 Regressione (S)ARIMA

Inoltre, abbiamo visto che i modelli ARIMA si basano sull'ipotesi di serie temporali stazionarie in senso ampio

Quindi, è necessario che sia il **valor medio che la varianza siano costanti nel tempo**

- La media si stabilizza per differenziazione
- La varianza si può stabilizzare mediante **trasformazioni Box-Cox**

Quindi, nel caso più generale il modello è $\text{SARIMA}(p, d, q, p_s, d_s, q_s, \lambda)$

Prima di eseguire una regressione è quindi necessario definire i valori dei sette parametri, evitando sovra- e sotto-adattamento

- d, d_s possono essere identificati per tentativi, aumentando gradualmente i valori (prima di d e poi di d_s) finché la ACF è soddisfacente
- gli altri termini possono essere individuati verificando la bontà della regressione su una griglia di combinazioni e valutando un opportuno **indice di merito**

2.21 Akaike Information Criterion

L'indice di merito più usato nella regressione SARIMA è l'**Akaike Information Criterion**, o AIC

In genere la qualità di una regressione con k parametri è misurata dalla somma quadratica dei residui, $SS_E(k)$, normalizzata per la dimensione del campione n:

$$\widehat{\sigma_k^2} = \frac{SS_E(k)}{n}$$

Questo indicatore si chiama ***Maximum Likelihood Estimator*** (MLE). Più questo valore è piccolo, meglio il modello si adatta ai dati

Tuttavia aumentando k si ha una diminuzione di MLE, a rischio di sovra-adattamento. Per questo motivo Akaike ha proposto di penalizzare MLE con il numero di parametri:

$$\text{AIC} = \log(\widehat{\sigma_k^2}) + \frac{n + 2k}{n}$$

L'AIC va ovviamente **minimizzato**

2.22 Altri criteri di informazione

Oltre all'AIC esistono anche l'AIC corretto e il *Bayesian Information Criterion*, o BIC:

- AIC corretto: $\text{AIC}_c = \log(\widehat{\sigma_k^2}) + \frac{n+k}{n-k-2}$
- BIC: $\text{BIC} = \log(\widehat{\sigma_k^2}) + \frac{k\log(n)}{n}$

Il BIC **penalizza maggiormente la dimensione del modello**, per cui è preferito per campioni molto grandi (migliaia di osservazioni), per i quali AIC e AICc tenderebbero a favorire modelli inutilmente complessi (troppi parametri, sovra-adattamento)

Come l'AIC, anche AICc e BIC vanno **minimizzati**

2.23 Regressione

La libreria R `forecast` fornisce la funzione `auto.arima()` che valuta gli indicatori su una griglia di combinazioni dei sette parametri e fornisce la regressione migliore:

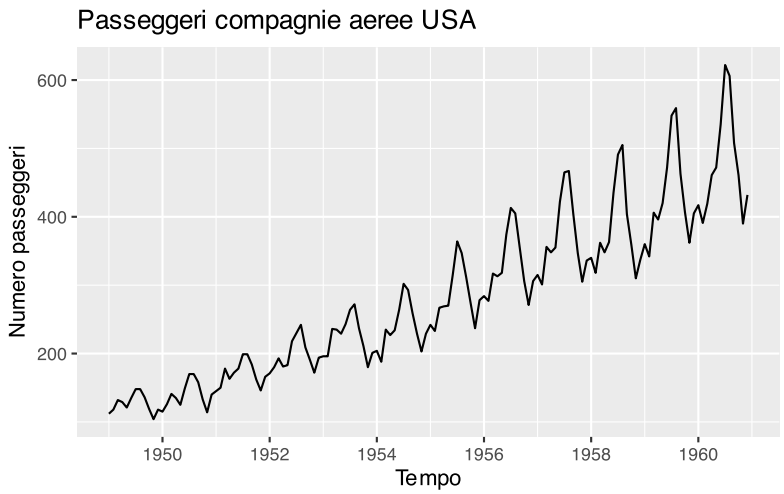
```
forecast::auto.arima(AirPassengers, lambda="auto", trace=T)
```

```
 ARIMA(2,1,2)(1,1,1)[12]                    : -890.0522
 ARIMA(0,1,0)(0,1,0)[12]                    : -845.0766
 ARIMA(1,1,0)(1,1,0)[12]                    : -885.6939
 ARIMA(0,1,1)(0,1,1)[12]                    : -896.9901
 ARIMA(0,1,1)(0,1,0)[12]                    : -860.1426
 ARIMA(0,1,1)(1,1,1)[12]                    : -895.2944
 ARIMA(0,1,1)(0,1,2)[12]                    : -895.3558
 ARIMA(0,1,1)(1,1,0)[12]                    : -889.5331
 ARIMA(0,1,1)(1,1,2)[12]                    : Inf
 ARIMA(0,1,0)(0,1,1)[12]                    : -880.0685
 ARIMA(1,1,1)(0,1,1)[12]                    : -896.1031
 ARIMA(0,1,2)(0,1,1)[12]                    : -895.698
 ARIMA(1,1,0)(0,1,1)[12]                    : -893.2768
 ARIMA(1,1,2)(0,1,1)[12]                    : -894.0835

 Best model: ARIMA(0,1,1)(0,1,1)[12]
```

```
Series: AirPassengers
ARIMA(0,1,1)(0,1,1)[12]
Box Cox transformation: lambda= -0.2947046

Coefficients:
          ma1     sma1
      -0.4355  -0.5847
s.e.   0.0908   0.0725

sigma^2 = 5.856e-05:  log likelihood = 451.59
AIC=-897.18   AICc=-896.99   BIC=-888.55
```

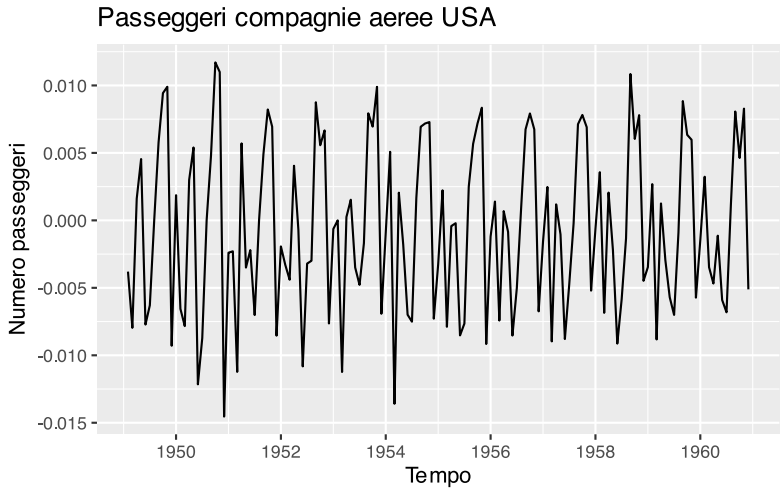
2.23.a Serie temporale

```
AirPassengers %>% autoplot() +
        labs(x="Tempo",    y="Numero    passeggeri",
title="Passeggeri compagnie aeree USA")
```



2.23.b Stabilizzata

```
diff(AirPassengers^-0.295) %>% autoplot() +
        labs(x="Tempo",    y="Numero    passeggeri",
title="Passeggeri compagnie aeree USA")
```



2.24 Previsione

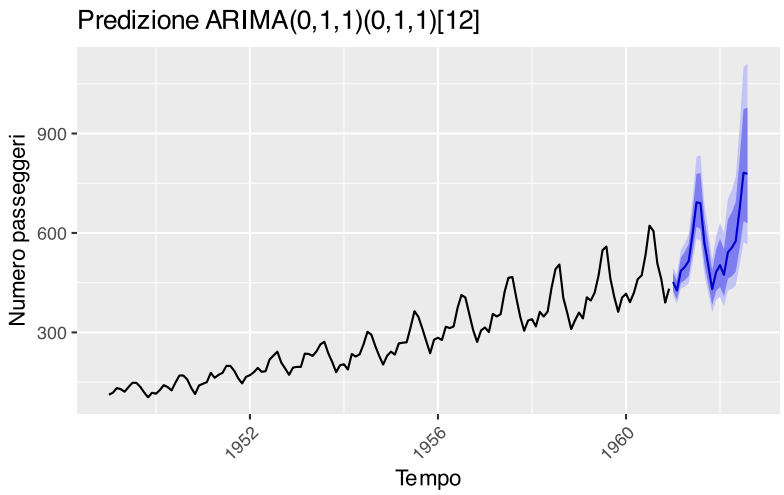
Una volta selezionato il modello più adatto (quello con **AIC minimo**) si può procedere a regressione e predizione

La predizione può essere ottenuta con la funzione `forecast()`

Tale funzione riporta anche le bande di confidenza al 95%

3 Predizione

```
fit      <-      forecast::auto.arima(AirPassengers,
lambda="auto")
autoplot(forecast::forecast(fit,h=20)) +
  theme(axis.text.x = element_text(angle = 45,
        hjust = 1)) +
    labs(title="Predizione  ARIMA(0,1,1)(0,1,1)[12]",
x="Tempo", y="Numero passeggeri")
```



4 Dettaglio

```
fit      <-      forecast::auto.arima(AirPassengers,
lambda="auto")
forecast::forecast(fit,h=20) %>%
  ggplot(aes(x=Index, y=Data)) +
  geom_line() +
     geom_ribbon(aes(ymin=`Lo  95`,  ymax=`Hi  95`),
alpha=1/3) +
     geom_ribbon(aes(ymin=`Lo  80`,  ymax=`Hi  80`),
alpha=1/3) +
  geom_line(aes(y=`Point Forecast`), color="blue") +
       scale_x_date(limits=c(ymd("1960/1/1"),    NA),
date_minor_breaks = "1 month")+
  theme(axis.text.x = element_text(angle = 45,
        hjust = 1)) +
```

```
    labs(title="Predizione  ARIMA(0,1,1)(0,1,1)[12]",
x="Tempo", y="Numero passeggeri")
```