Università degli Studi di Padova

Dipartimento di Scienze Statistiche

Corso di Laurea Triennale in

Statistica per l'Economia e l'Impresa

RELAZIONE FINALE

IMPLEMENTAZIONE DI ALGORITMI DI PREVISIONE PER SERIE STORICHE, UN PROGETTO STAGE CON ZUCCHETTI S.p.A.

Relatore:

Prof.ssa Luisa Bisaglia

Dipartimento di Scienze Statistiche

Laureanda:

Maria Grazia Biasco

Matricola n: 1169146

ANNO ACCADEMICO 2020/2021

Indice

Sommario

Il seguente lavoro di tesi è stato elaborato in seguito all'esperienza di stage lavorativo svolto con l'azienda Zucchetti S.p.A. e si pone come scopo quello di analizzare nel dettaglio il progetto realizzato.
L'obiettivo principale del lavoro di stage consisteva nell'implementazione di algoritmi basati su tecniche e modelli predittivi da applicare nell'ambito dell'analisi delle serie storiche. In particolare, il lavoro realizzato è pensato per entrare a far parte di un software più completo e interattivo che l'azienda metterebbe a disposizione dei propri clienti.
Nel primo capitolo verrà presentata l'azienda ospitante e verrà descritto più ampiamente il progetto in esame in relazione anche ai linguaggi e le tecnologie usati dall'azienda. Nel secondo capitolo verranno illustrate le metodologie statistiche utilizzate, ossia i metodi del Lisciamento Esponenziale e i modelli ARIMA. Il terzo capitolo verrà dedicato alla realizzazione effettiva del progetto, mostrandone un'applicazione a titolo esemplificativo su una serie storica. In Appendice verrà riportato, infine, un "Manuale utente", nel quale sono state descritte nel dettaglio tutte le funzioni implementate.

Capitolo 1

1.1 L'Azienda

Zucchetti S.p.A. è la prima *software-house* italiana per storia e dimensione, nonché una delle più importanti nel settore ICT *(Information and Communication Technology)*. Zucchetti S.p.A. fa parte del Gruppo Zucchetti il quale offre soluzioni *software* gestionali, *hardware* e servizi innovativi per soddisfare le specifiche esigenze del cliente.
L'azienda copre un *target* di clienti molto ampio, lavorando per aziende, banche, assicurazioni, professionisti e associazioni di categoria, CAF e Pubblica Amministrazione; operando pertanto in svariati settori, fornendo prodotti gestionali aziendali e soluzioni *ERP* (gestione del personale, portali e comunicazione aziendali, rilevazione presenze e accessi, etc.), lavora inoltre nel campo dell'*e-business* e della *business intelligence*, e, infine, offre servizi per robotica, automazione, sicurezza informatica e logica, *server farm*.

1.2 Descrizione dello stage e obiettivi

L'analisi dei dati a fini previsivi è sempre più rilevante anche in ambito aziendale e, pertanto, diventa fondamentale lo sviluppo di *software* e applicativi che possano aiutare a manipolare, modellare e visualizzare dati e risultati di analisi.
Il progetto di stage è centrato sulla previsione puntuale delle serie storiche; in particolare, si vogliono implementare delle funzioni che permettano di ottenere delle previsioni sfruttando i metodi del Lisciamento Esponenziale e la classe dei

modelli ARIMA. Si vogliono, inoltre, integrare i risultati con delle analisi sulla valutazione delle capacità previsive, quali intervalli di previsione e indici appositi. I risultati verranno visualizzati qualitativamente con dei grafici interattivi visibili su *browser*, nei quali verranno riportate la serie dei dati osservati, le previsioni ottenute e i relativi intervalli di previsione, inoltre, tutti i risultati numerici verranno stampati sulla *console*.

1.3 Tecnologie

Il progetto è stato realizzato per essere presentato interamente in una pagina web, per cui le tre tecnologie sulle quali si fonda e che l'azienda utilizza abitualmente, sono: *JavaScript*, *HTML* e *CSS*.

JavaScript

JavaScript è un linguaggio di programmazione interpretato ed orientato agli oggetti e agli eventi. Per poter essere compilato, il codice sorgente viene integrato all'interno di altro codice (in questo caso, l'HTML) e quindi utilizzato come linguaggio di *scripting*. In particolare, l'uso del JavaScript "basato sul web" permette di curare l'interazione all'interno del progetto web, ossia fornisce le istruzioni da eseguire per tutti gli eventi che seguono da una particolare operazione dell'utente (come ad esempio un click o la digitazione da tastiera), rendendo così la pagina dinamica.
Esistono numerosi linguaggi molto convenienti da utilizzare per effettuare analisi statistiche, quali R, Python o Julia per citarne alcuni, tuttavia JavaScript risulta essenziale per sviluppare delle interfacce *web-based*. Si tratta, infatti, di un linguaggio versatile che permette di condividere il lavoro realizzato su qualsiasi dispositivo senza richiedere nessun tipo di installazione, in più il codice sorgente è scaricato "in chiaro" ed è accessibile a chiunque voglia leggerlo, rendendo possibile una comprensione più approfondita del lavoro finale. Gli svantaggi nell'utilizzo di questo linguaggio riguardano soprattutto il fatto che non comprenda

pacchetti che effettuino analisi statistiche, inoltre bisogna sottolineare il fatto che le operazioni dell'algebra lineare non vengono svolte in automatico ma mediante cicli *"for"* annidati computazionalmente onerosi.

HTML

L'HTML (*HyperText Markup Language*) è un linguaggio di *markup* nato per la formattazione, l'impaginazione e la visualizzazione grafica di documenti web. Esso codifica opportunamente il contenuto logico delle pagine, definendone la struttura in modo statico.

CSS

Il CSS *(Cascading Style Sheets o Fogli di stile)* è il linguaggio usato per definire la formattazione e la personalizzazione della pagina, usato affiancandolo ad HTML. Attraverso il CSS si stabilisce la presentazione e lo stile del documento, definendone ad esempio colori, font, bordi e margini, etc.

D3.js

La libreria JavaScript *d3.js* permette di manipolare documenti basati su dati[1]. Applicata al presente lavoro, il suo utilizzo comporta l'implementazione di ogni singola componente del grafico (gli assi e il loro orientamento, la funzione della linea, i punti, la griglia, etc.) utilizzando l'SVG (*Scalable Vector Graphics*), ossia una specificazione per la descrizione di grafici vettoriali bidimensionali, integrato direttamente nel documento HTML e, pertanto, nel progetto web.

Capitolo 2

Metodologie utilizzate

2.1 Le serie storiche

Si definisce serie storica una successione di valori numerici x_t ordinati in base all'indice t, dove col pedice t si vuole indicare la dipendenza delle osservazioni dal tempo: ogni dato è registrato nell'istante t ed esiste dipendenza tra osservazioni successive.

L'assunzione sulla quale si basa l'analisi di una serie storica, relativa ad una variabile Y, definisce la serie osservata come la realizzazione di un processo generatore definito dal modello stocastico $Y_t = f(t) + u_t$. Quest'ultimo è la composizione di una parte sistematica e completamente deterministica, $f(t)$, e di una parte stocastica, u_t.

A seconda di come venga considerata la componente stocastica, si distinguono due filoni principali nell'analisi delle serie storiche:

- se u_t viene considerato un processo a componenti incorrelate, e pertanto l'analisi è focalizzata su $f(t)$, allora si parla di *approccio classico*;
- se u_t viene considerato un processo a componenti correlate, e pertanto l'analisi è focalizzata su di esso, si parla di *approccio moderno*.

Generalmente, la componente deterministica $f(t)$ è la risultante dell'azione congiunta di tre componenti: il trend, il ciclo e la stagionalità; tuttavia si prenderanno in considerazione solo trend e stagionalità.

Trend	**Stagionalità**		
	Nessuna	*Additiva*	*Moltiplicativa*
Nessuno	N,N	N,A	N,M
Additivo	A,N	A,A	A,M
Additivo "damped"	A_d, N	A_d, A	A_d, M
Moltiplicativo	M,N	M,A	M,M
Moltiplicativo "damped"	M_d, N	M_d, A	M_d, M

Tabella 2.1: *Specificazione dei quindici diversi metodi di Lisciamento Esponenziale. Le celle evidenziate sono quelle trattate in questo lavoro. Fonte:[1]*

2.2 Lisciamento Esponenziale

Il Lisciamento Esponenziale è una classe di metodi di previsione per serie storiche univariate che ricade nell'approccio classico dell'analisi delle serie storiche, particolarmente usata per le previsioni nel breve periodo grazie alla sua flessibilità e semplicità d'uso. L'idea su cui si fonda tale metodologia è che la previsione al tempo $t + k$ sia il risultato di una combinazione pesata di tutte le osservazioni giunte al tempo $t + k - 1$, con pesi esponenzialmente decrescenti al decrescere dell'ordine temporale delle osservazioni (da qui l'appellativo di *Esponenziale*).
Si possono individuare quindici diversi metodi di Lisciamento, riassunti nella Tabella 2.1, ottenuti distinguendo tra la diversa "natura" delle componenti del trend e della stagionalità di una serie storica. La componente erratica, che può essere di tipo additivo o moltiplicativo, viene qui ignorata poiché questa distinzione non fa differenza nella previsione puntuale [1].

I previsori basati sul Lisciamento Esponenziale non consentono di formulare affermazioni probabilistiche, data la natura non parametrica del metodo. Per la costruzione di intervalli di previsione pertanto si rende necessario ricorrere a tecniche di simulazione. Tale metodologia si poggia sulla specificazione *state-space* dei modelli, secondo la quale una serie storica $y_1, y_2, ..., y_n$ è associata ad una serie di vettori non osservabili, che descrive lo stato della variabile Y_{t+1} al tempo $t + 1$ in funzione dei precedenti stati Y_t e di un termine di errore[1]. Tale relazione vie-

ne specificata tramite una *equazione di misurazione*, mentre l'*equazione di stato* descrive l'evoluzione del sistema dinamico nel tempo [2].

Nei modelli *state-space* vengono distinti gli errori moltiplicativi da quelli additivi, poiché portano ad intervalli di previsione differenti, per questo motivo nella specificazione del modello viene aggiunta una lettera che denota la natura additiva (A) o moltiplicativa (M) degli errori. Nel seguente lavoro verranno presi in considerazione i modelli lineari con una singola fonte di errore (*SSOE*) e omoschedastici, per questo motivo verrà escluso il calcolo degli intervalli di previsione per il Lisciamento Esponenziale Triplo con stagionalità moltiplicativa, in cui l'equazione per le osservazioni è non-lineare e gli errori sono eteroschedastici. Inoltre, per comodità verranno implementati solo i modelli con errori additivi.

Gli intervalli di previsione tramite simulazione prevedono di simulare M "traiettorie" dal modello *state-space* della serie storica (con M numero elevato, anche in base alle prestazioni dell'*hardware* a disposizione), ottenendo quindi k vettori $(\hat{y}_{t+1}, ..., \hat{y}_{t+k})$ M-dimensionali condizionatamente a y_t, generando per ϵ_t un valore casuale da una distribuzione appropriata (generalmente, assumendo che ϵ_t sia Gaussiana di media 0 e varianza σ_ϵ^2, si ottengono dei risultati equivalentemente affidabili[1]). Gli intervalli di previsione al $100(1-\alpha)\%$ per $\hat{y}_{t+k}$ sono dati pertanto dai quantili $\frac{\alpha}{2}, 1-\frac{\alpha}{2}$ delle M osservazioni simulate in quel punto.

2.2.1 Simple Exponential Smoothing, (N,N)

La più basilare tra le tecniche di previsione è il Lisciamento Esponenziale Semplice (*SES, Simple Exponential Smoothing*), da applicare a serie che non presentano stagionalità, trend sistematico o marcate fluttuazioni, ma che invece presentano un livello che cambia in modo stocastico. Si tratta di una tecnica che consiste nell'effettuare una media ponderata di tutte le osservazioni disponibili, con peso variabile in base alla prossimità temporale. La previsione $\hat{y}_{n+1}$ ottenuta in questo modo risulta pertanto

$$\hat{y}_{n+1} = \alpha\hat{y}_n + (1-\alpha)y_n,$$

ossia la previsione $\hat{y}_{n+1}$ è una media ponderata tra $\hat{y}_n$, previsione di y_n fatta al tempo $n-1$, e l'ultima osservazione disponibile, y_n, con pesi rispettivamente α e $1-\alpha$.

Il Lisciamento Esponenziale Semplice fornisce delle previsioni costanti, ragion per cui è sufficiente calcolare la previsione della serie storica in esame un solo passo avanti. Quindi, finché non giungono nuove informazioni, si avrà $\hat{y}_{n+k} = \hat{y}_{n+1}, \forall k \geq 1$. [3]

Essendo uno schema di calcolo ricorsivo, è necessario inizializzare la successione di previsioni $\hat{y}_n$ e individuare il valore ottimale da attribuire ad α. Relativamente al valore iniziale di $\hat{y}_n$, si può porre:

$$\hat{y}_1 = \frac{1}{n}\sum_{t=1}^{n} y_t,$$

in quanto di utilizzo molto diffuso, ossia $\hat{y}_1$ è la media aritmetica degli n valori della serie storica; mentre il valore ottimale per α è quel valore che minimizza la somma degli errori di previsione al quadrato, ossia:

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}}\{S(\alpha) = \sum_{t=1}^{n-1}(y_{t+1} - \hat{y}_{t+1})^2\}.$$

La specificazione *state-space* del Lisciamento Esponenziale con errore additivo, ossia le equazioni per il modello ETS(A,N,N), è:

$$Y_t = l_{t-1} + \epsilon_t$$

$$l_t = l_{t-1} + \alpha\epsilon_t$$

$$\mu_t = \hat{y}_{t+h|t} = l_t,$$

in cui l_t la componente del livello, $\epsilon_t \sim NID(0, \sigma^2)$.

2.2.2 Double Exponential Smoothing, (A,N)

Il Lisciamento Esponenziale Doppio (Lisciamento Esponenziale di Holt, metodo non stagionale lineare) è una generalizzazione del Lisciamento Esponenziale Semplice, applicato a serie storiche che presentano un trend sistematico (di tipo

additivo), ma non una componente stagionale. In accordo con questo metodo, in prossimità di n la serie può essere approssimata da una retta di equazione

$$L_n + T_n(t - n),$$

dove l'intercetta stimata $\hat{L}_n$ rappresenta il livello della serie nel periodo n, mentre la stima del coefficiente angolare $\hat{T}_n$ ne rappresenta "la crescita", ovvero la pendenza.
Il Lisciamento Esponenziale Doppio è, analogamente a quello Semplice, uno schema di calcolo ricorsivo: dopo aver inizializzato le successioni di $[\hat{L}_t, \hat{T}_t]$, l'algoritmo aggiorna ricorsivamente le loro stime seguendo lo schema [2]:

$$\hat{L}_n = \alpha y_n + (1 - \alpha)(\hat{L}_{n-1} + \hat{T}_{n-1}), \qquad 0 < \alpha < 1$$

$$\hat{T}_n = \beta(\hat{L}_n - \hat{L}_{n-1}) + (1 - \beta)\hat{T}_{n-1}, \qquad 0 < \beta < 1$$

Il processo di inizializzazione per questo metodo richiede di individuare i valori di $[\hat{L}_1$ e $\hat{T}_1]$. A tal fine si pone, in genere:

$$\hat{L}_1 = y_1 \quad e \quad \hat{T}_1 = \frac{(y_2 - y_1) + (y_3 - y_2) + (y_4 - y_3)}{3},$$

cioè il valore iniziale del livello della serie è pari alla prima osservazione, e il valore iniziale del trend è pari alla media delle prime tre osservazioni della serie storica ottenuta dopo la rimozione della componente del trend.
Questo metodo fa uso di due costanti, α e β, le quali possono essere ottimizzate minimizzando, similmente al Lisciamento Esponenziale Semplice, la somma degli errori quadratici tra la serie originale e la serie ottenuta col Lisciamento Esponenziale Doppio:

$$\hat{\alpha}, \hat{\beta} = \underset{\alpha,\beta}{\operatorname{argmin}}\{S(\alpha, \beta) = \sum_{t=1}^{n-1}(y_{t+1} - \hat{y}_{t+1})^2\}.$$

Dopo aver ottenuto le n stime di $[\hat{L}_t, \hat{T}_t]$, il Lisciamento Esponenziale Doppio fornisce le previsioni della serie:

$$\hat{y}_{n+k} = \hat{L}_n + \hat{T}_n k, \qquad k \geq 1.$$

Il Lisciamento Esponenziale Doppio corrisponde nella forma dei modelli *state-space* al modello ETS(A,A,N) in caso di errori additivi e al modello ETS(M,A,N) in caso di errori moltiplicativi. Si riporta di seguito la scrittura per il caso additivo:

$$Y_t = l_{t-1} + b_{t-1} + \epsilon_t$$

$$l_t = l_{t-1} + b_{t-1} + \alpha\epsilon_t$$

$$b_t = b_{t-1} + \alpha\beta\epsilon_t$$

$$\mu_t = \hat{y}_{t+h|t} = l_t + hb_t,$$

con l_t la componente del livello, b_t la componente di "crescita", $\epsilon_t \sim NID(0, \sigma^2)$.

Double Damped Exponential Smoothing, $(\mathbf{A}_d, N)$[1]

Il Lisciamento Esponenziale Doppio porta a delle previsioni che crescono o decrescono indefinitamente nel tempo, risultando empiricamente "esagerate". Per questo motivo esiste una versione "smorzata" del metodo di Holt (*"damped"*), in grado di smorzare la crescita del trend all'aumentare dell'orizzonte temporale di previsione, grazie all'introduzione di una costante di smorzamento ϕ, compresa tra zero e uno.

$$\hat{L}_n = \alpha y_n + (1-\alpha)(\hat{L}_{n-1} + \phi\hat{T}_{n-1}), \qquad 0 < \alpha < 1$$

$$\hat{T}_n = \beta(\hat{L}_n - \hat{L}_{n-1}) + (1-\beta)\phi\hat{T}_{n-1}, \qquad 0 < \beta < 1$$

$$\hat{y}_{n+k} = \hat{L}_n + \sum_{i=1}^{h} \phi^i \hat{T}_n, \qquad k \geq 1.$$

Il modello "Damped" è definito in forma *state-space* come:

$$Y_t = l_{t-1} + b_{t-1} + \epsilon_t$$

$$l_t = l_{t-1} + b_{t-1} + \alpha\epsilon_t$$

$$b_t = \phi b_{t-1} + \alpha\beta\epsilon_t$$

$$\mu_t = \hat{y}_{t+h|t} = l_t + \phi b_t.$$

2.2.3 Holt Winters' (Triple) Exponential Smoothing, (A,A),(A,M) [1]

Il Lisciamento Esponenziale Triplo (metodo Holt-Winters stagionale) è la metodologia più indicata per ottenere delle previsioni puntuali nel caso di serie storiche che comprendono un trend sistematico (additivo) e una stagionalità di periodo s, il quale indica dopo quanti istanti la fluttuazione della componente stagionale si ripete in maniera pressoché invariata.
La componente stagionale può essere di tipo additivo o moltiplicativo: nel primo caso, l'ampiezza della fluttuazione stagionale è indipendente dal livello della serie, ossia la stagionalità varia all'interno di un range di valori la cui ampiezza rimane fissa; nel modello moltiplicativo, invece, le dimensioni delle fluttuazioni stagionali variano a seconda del livello della serie, ovvero la stagionalità assume valori all'interno di un range la cui ampiezza dipende dal livello. È definito un metodo di Lisciamento diverso per ognuna di queste tipologie.
Per poter inizializzare gli algoritmi, un metodo basato sulla decomposizione della serie che permette di ottenere le stime di trend, livello e stagionalità a partire dall'istante 1, è definito dal seguente schema [1]:

1. Si adatta ai primi due anni di osservazioni una media mobile pesata $2 \times s$ (quindi una media mobile di ordine $s + 1$), con peso $\frac{1}{2s}$ per la prima e l'ultima osservazione, e con peso $\frac{1}{s}$ per tutte le altre.

2. Si ottiene la serie privata della componente del trend sottraendo (in caso di modello additivo) o dividendo (in caso di modello moltiplicativo) alla serie originale la serie "lisciata" con la media mobile.

3. Si ottengono i primi s valori della stagionalità come media dei valori ottenuti al punto precedente ad intervalli di s. Ossia, nel caso di serie mensili, il primo valore di stagionalità (corrispondente a Gennaio) si calcola come la media dei valori di Gennaio dei vari anni della serie priva del trend, il secondo valore di S (Febbraio) è la media dei valori di Febbraio, e così via.

4. Si ottiene la serie destagionalizzata sottraendo (in caso di modello additivo) o dividendo (in caso di modello moltiplicativo) alla serie originale la successione degli indici di stagionalità.

5. Si effettua una semplice regressione lineare (OLS) sui dati destagionalizzati, dalla quale si ottiene il valore iniziale per il livello L_0 come intercetta della retta, calcolata come: $\hat{L}_0 = \frac{\sigma_{xy}}{\sigma_x^2}$, e il valore iniziale per il trend T_0 come coefficiente angolare della retta, calcolato come: $\hat{T}_0 = \overline{y} - \hat{L}_0\overline{x}$ (in questo caso, le x sono i valori che vanno da 1 a n mentre le y sono le osservazioni della serie storica).

Nel caso di stagionalità additiva, abbiamo che la serie storica $\{y_t\}_{t=1}^n$ è esprimibile, in prossimità di n, come

$$L_n + T_n(t-n) + S_n.$$

Dopo aver inizializzato l'algoritmo, le formule di aggiornamento per le successioni del livello, del trend e della stagionalità sono:

$$\hat{L}_n = \alpha[y_n - \hat{S}_{n-s}] + (1-\alpha)(\hat{L}_{n-1} + \hat{T}_{n-1})$$

$$\hat{T}_n = \beta[\hat{L}_n - \hat{L}_{n-1}] + (1-\beta)\hat{T}_{n-1}$$

$$\hat{S}_n = \gamma[y_n - \hat{L}_{n-1} - \hat{T}_{n-1}] + (1-\gamma)\hat{S}_{n-s}$$

con $0 < \alpha, \beta, \gamma < 1$.
A questo punto, le previsioni k passi in avanti sono date da:

$$\hat{y}_{n+k} = \hat{L}_n + \hat{T}_n k + \hat{S}_{n+k-s(h+1)}$$

dove h è la parte intera della divisione $(k-1)/s$, che garantisce che le stime degli indici stagionali usati per la previsioni siano dell'ultimo anno del campione.[1]
Nel caso di stagionalità moltiplicativa, la serie storica $\{y_t\}_{t=1}^n$ è esprimibile, in prossimità di n, come

$$[L_n + T_n(t-n)]S_n.$$

Le formule di aggiornamento diventano quindi:

$$\hat{L}_n = \alpha[y_n/\hat{S}_{n-s}] + (1-\alpha)(\hat{L}_{n-1} + \hat{T}_{n-1})$$

$$\hat{T}_n = \beta[\hat{L}_n - \hat{L}_{n-1}] + (1-\beta)\hat{T}_{n-1}$$

$$\hat{S}_n = \gamma[y_n/(\hat{L}_{n-1} - \hat{T}_{n-1})] + (1-\gamma)\hat{S}_{n-s}$$

con $0 < \alpha, \beta, \gamma < 1$.

Le previsioni sono invece date da:

$$\hat{y}_{n+k} = (\hat{L}_n + \hat{T}_n k)\hat{S}_{n+k-s(h+1)}.$$

Il caso del modello additivo corrisponde al modello *state-space* ETS(A,A,A), ossia quello definito dalle seguenti equazioni:

$$Y_t = l_{t-1} + b_{t-1} + s_{t-s} + \epsilon_t$$

$$l_t = l_{t-1} + b_{t-1} + \alpha\epsilon_t$$

$$b_t = b_{t-1} + \alpha\beta\epsilon_t$$

$$s_t = s_{t-s} + \gamma\epsilon_t$$

$$\mu_t = \hat{y}_{t+h|t} = l_t + hb_t + s_{t-s+1+(h-1)}.$$

2.3 Processi ARIMA

Nell'ambito dell'approccio moderno all'analisi delle serie storiche, si suppone che la serie storica osservata sia la realizzazione finita di un processo stocastico generatore dei dati.

I principali processi individuabili sono:

- Il *processo a media mobile di ordine q (MA(q))*, per il quale la variabile Y al tempo t è funzione lineare di q disturbi passati (osservati) *white noise*, compreso il disturbo al tempo corrente; formalmente, $Y_t = \epsilon_t - \theta_1\epsilon_{t-1} - ... - \theta_q\epsilon_{t-q}$, con $\theta_i, i = 0, ..., q$, parametri ed $\{\epsilon_t\}$ processo *white noise* di media 0 e varianza σ^2_ϵ;

- Il *processo autoregressivo di ordine p (AR(p))*, per il quale la variabile Y_t regredisce sui suoi p valori passati (da cui il prefisso *auto*) e ai quali viene sommato uno shock casuale al tempo corrente; formalmente, $Y_t = \phi_0 + \phi_1 Y_{t-1} + ... + \phi_p Y_{t-p} + \epsilon_t$, con $\phi_i, i = 0, ..., p$, parametri ed $\{\epsilon_t\} \sim WN(0, \sigma^2_\epsilon)$.

Per poter ottenere una maggiore flessibilità nella modellazione delle serie storiche, facendo attenzione a seguire il "principio della parsimonia", si è reso necessario estendere i modelli appena descritti, definendo il modello misto *autoregressivo a media mobile (ARMA(p,q))*. Questo processo descrive la variabile Y_t come una combinazione dei processi AR ed MA, ovvero:

$$Y_t - \sum_{i=1}^{p} \phi_i Y_{t-i} = \phi_0 + \epsilon_t - \sum_{j=1}^{q} \theta_j \epsilon_{t-j},$$

con ϕ_i e $\theta_j, i = 0, ..., p$ e $j = 1, ..., q$ parametri ed $\epsilon_t \sim WN(0, \sigma_\epsilon^2)$. I modelli *ARMA(p,q)* si adattano bene a serie storiche stazionarie, vale a dire quei processi per i quali media e varianza non variano col variare del tempo.

Le serie non-stazionarie, casistica molto più frequente rispetto alla controparte, possono comunque essere stabilizzate tramite delle opportune trasformazioni. In particolare, se un processo non-stazionario diventa stazionario a seguito di d differenziazioni successive, si dice che il processo è *non-stazionario omogeneo di ordine d*. Per poter rappresentare questi processi ci si serve di un'estensione dei modelli *ARMA(p,q)* che prende il nome di *processo autoregressivo integrato a media mobile (ARIMA(p,d,q))*. Si ha quindi che $\{Y_t\}$ segue un *ARIMA(p,d,q)* se $\{X_t\} = (1 - B)^d Y_t$, dove con B indichiamo l'operatore ritardo, è un processo *ARMA(p,q)*.

Formalmente, la classe dei modelli *ARIMA(p,d,q)* è rappresentata da:

$$X_t = (1 - B)^d Y_t$$

$$X_t = \phi_0 + \sum_{i=1}^{p} \phi_i X_{t-i} + \epsilon_t - \sum_{j=1}^{q} \theta_j \epsilon_{t-j}.$$

In un contesto di analisi, in prima battuta si effettua un'accurata specificazione del processo atto a modellare la serie in esame. A tal fine, si può procedere con un'identificazione strutturale, ossia osservando le strutture delle funzioni di autocorrelazione e di autocorrelazione parziale empiriche, oppure si può ricorrere a dei criteri di informazione automatica volti a penalizzare il numero dei parametri del modello, evitando il rischio di *overfitting*.

Tra i criteri più comuni da minimizzare si citano il *criterio di informazione di*

Akaike, AIC definito come $-\frac{2}{n}\log(L(\hat{\delta}))+\frac{2}{n}k$, e il *criterio di informazione Bayesiano, BIC*, definito come $-\frac{2}{n}(\hat{\delta})+\frac{\log(n)}{n}k$, in cui k è il numero di parametri stimati, $\hat{\delta}$ è il vettore $(n\times 1)$ contenente le loro stime e $L(\hat{\delta})$ è la funzione di verosimiglianza calcolata in $\hat{\delta}$ e in ipotesi di Gaussianità del white noise[3].

La stima di un modello *ARIMA(p,d,q)* consiste nella stima dei suoi $p+q+2$ parametri, generalmente mediante il metodo della massima verosimiglianza, date le buone proprietà statistiche degli stimatori risultanti. In particolare, si tratta di massimizzare la funzione di verosimiglianza approssimata, in assunzione di Normalità, condizionandosi alle prime p osservazioni e a particolari valori delle ϵ. Successivamente è fondamentale effettuare una validazione del modello per verificarne l'adeguatezza, analizzando la correlazione tra i residui.

Per quanto riguarda la previsione, dato il contesto probabilistico, si è in grado di ottenere dei previsori che godono di ottime proprietà e che possono essere affiancati da intervalli di previsione. Nella fattispecie, il previsore ottimo $\hat{Y}_{n+k}$ è il valore atteso condizionato all'informazione passata:

$$\hat{Y}_{n+k}=E[Y_{n+k}|\mathcal{I}_n]$$

$$E[Y_{t+j}|Y_t=y_t,...,Y_1=y_1]=\begin{cases}y_{t+j} & j\leq 0\\ \hat{y}_{t+j} & j\geq 0\end{cases}$$

$$E[\epsilon_{t+j}|Y_t=y_t,...,Y_1=y_1]=\begin{cases}e_{t+j} & j\leq 0\\ 0 & j\geq 0\end{cases},$$

ponendo $e_t=y_t-\hat{y}_t$.

Gli intervalli di previsione si ottengono nel modo seguente:

$$\hat{y}_{n+k}\pm z_{1-\frac{\alpha}{2}}\sigma[e_{t+k}],$$

dove $z_{1-\frac{\alpha}{2}}$ indica il percentile di ordine $1-\frac{\alpha}{2}$ di una distribuzione $N(0,1)$.

2.4 Qualità delle previsioni

La qualità delle previsioni si valuta generalmente *ex post*, cioè confrontando le stime delle previsioni con i valori effettivamente realizzati. Dal punto di vista

operativo, è utile suddividere il campione in un sottoinsieme di stima, che verrà usato per "costruire" il metodo o modello previsivo, e un sottoinsieme di convalida, col quale verrà valutata la bontà delle previsioni.
Per poter valutare in maniera quantitativa la qualità previsiva del modello specificato, si può ricorrere a diversi indici che considerano l'errore di previsione e_t, vale a dire lo scarto tra valori realizzati y_t e previsti $\hat{y}_t$:

- **EM**, errore medio di previsione $= \frac{1}{n}\sum_{t=1}^{n} e_t$
- **EAM**, errore assoluto di previsione $= \frac{1}{n}\sum_{t=1}^{n} |e_t|$
- **EQM**, errore quadratico medio di previsione $= \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$
- **ES**, errore sistematico $= \frac{(\bar{\hat{y}}-\bar{y})^2}{EQM^2}$
- **EV**, errore nelle variabilità $= \frac{(\sigma_{\hat{y}}-\sigma_y)^2}{EQM^2}$
- **EC**, errore nelle covarianze $= \frac{2\sigma_{\hat{y}}\sigma_y(1-\rho_{\hat{y}y})}{EQM^2}$

Valori vicini (in valore assoluto) degli indici EAM ed EM, indicano che vi è sistematicità negli errori di previsione[3].
Inoltre, dal momento che $EQM^2 = EQM^2(ES + EV + EC)$, abbiamo informazioni sulla natura dell'errore complessivo:
-**ES** indica la frazione di errore dovuta alla diversa media dei valori previsti e realizzati;
-**EV** indica la frazione di errore dovuta alla diversa variabilità dei valori previsti e realizzati;
-**EC** infine, misura l'errore dovuto all'imperfetta correlazione lineare fra i valori previsti e quelli realizzati, che può venire attribuita a fattori accidentali [3].

Capitolo 3

Realizzazione

3.1 Preparazione dei dati e visualizzazione

Si è scelto di utilizzare a titolo esemplificativo la serie storica *Crime2* della libreria *tsdl* già compresa in R, creata da Rob Hyndman, professore della Monash University (Australia). Si tratta di una serie mensile che va da Gennaio 1966 a Ottobre 1975, conta quindi 118 osservazioni (valori interi positivi, con nessun valore mancante).

Nell'ipotesi di ottenere previsioni per una finestra temporale di 12 istanti, la serie storica è stata distinta in un sottoinsieme di stima, che conterrà le prime 106 osservazioni da dare in input al modello, e un sottoinsieme di convalida, che conterrà le ultime 12 osservazioni, con le quali si potrà valutare l'accuratezza delle previsioni *in-sample* ottenute.

Viene creato un progetto web per ogni diversa metodologia di previsione: in ogni grafico (visibile attraverso il *browser*) si riportano i valori della serie, tracciando la linea riferita al sottoinsieme di stima in nero e quella riferita al sottoinsieme di convalida in grigio, sulla quale verranno poi sovrapposte le linee riferite alle previsioni insieme ai relativi intervalli di previsione. Inoltre vengono stampati su *console* i valori delle osservazioni della serie, distinti nei due sottoinsiemi creati.

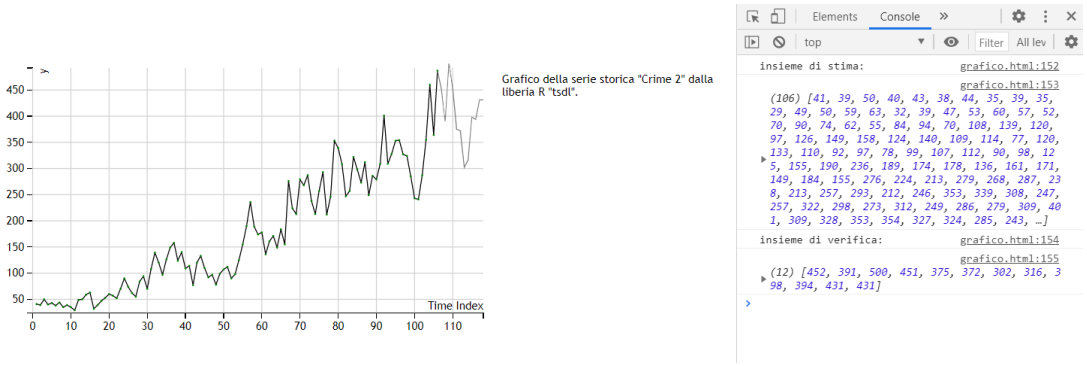

Figura 3.1: *Grafico della serie storica "Crime2" dalla libreria R "tsdl". Visualizzazione su browser.*

3.2 Indici per la qualità delle previsioni

Come già descritto nella sezione **2.4**, per poter formulare una valutazione circa l'accuratezza delle previsioni ottenute seguendo un determinato metodo o modello è utile ricorrere a degli indici specifici. Nella stessa sezione sono stati elencati e descritti gli indicatori più usati, i quali vengono pertanto implementati all'interno della stessa funzione. Inserendo tra i parametri di input il vettore delle previsioni p_t e il vettore del sottoinsieme di verifica della serie storica r_t, tale metodo si basa sull'errore di previsione $e_t = r_t - p_t$ che concorre nel calcolo degli indici.
La funzione restituisce un oggetto contenente al suo interno le sigle degli indici affiancate ai loro risultati, ordinate secondo un criterio alfabetico, come mostrato nella Figura 3.2.

3.3 Lisciamento Esponenziale

3.3.1 Intervalli di previsione simulati

Gli **intervalli di previsione**, nel caso delle stime ottenute seguendo i metodi del Lisciamento Esponenziale, si possono ricavare tramite simulazione seguendo l'approccio *state-space*. Pertanto, per ogni diversa metodologia di Lisciamento Esponenziale è stata realizzata una funzione in cui si passano in input la serie di

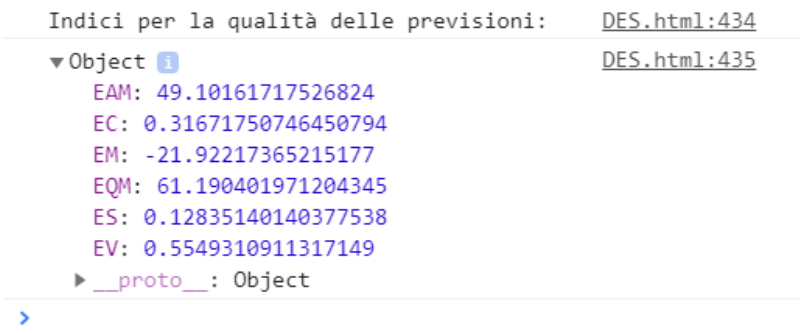

Figura 3.2: *Oggetto contenente gli indici per la bontà delle previsioni calcolati sull'applicazione del Lisciamento Esponenziale Doppio sulla serie storica "Crime2" dalla libreria R "tsdl".*

stima dei dati, i valori dei parametri riferiti ai metodi di Lisciamento (α, β, γ), il numero k di istanti temporali da prevedere e la deviazione standard σ_ϵ riferita alla componente erratica della serie storica in esame. Ognuna di queste funzioni si avvia con la stima ricorsiva delle componenti di L, T, S, come definito nelle funzioni per il Lisciamento Esponenziale, delle quali si utilizzano $\hat{L}_n, \hat{T}_n, \hat{S}_n$ per inizializzare le componenti di l_t, b_t, s_t, che vengono poi aggiornate mediante le specifiche equazioni riportate nel Capitolo 2.

Al fine di sviluppare il modello *state-space* è necessario individuare la componente erratica $e_t \sim N(0, \sigma_\epsilon^2)$ ricorrendo alla generazione di valori pseudo-casuali da una distribuzione Normale di media 0 e varianza σ_ϵ^2. A questo scopo è stata creata una funzione apposita che, attraverso la Trasformazione di Box-Muller nella sua forma principale, restituisce un valore $e_i \sim N(\mu, \sigma^2)$, con i parametri μ e σ specificati in input.

A questo punto si creerà una matrice di dimensione $(M \times k)$ per ognuna di queste componenti, dove con M si indica il numero di simulazioni (iterazioni) da eseguire mentre k è il numero di previsioni da ottenere. Infine, dalle equazioni di previsione relative ad ogni metodo di Lisciamento descritto, ossia combinando opportunamente le matrici l_t, b_t, s_t, deriva la matrice delle stime $\hat{Y}$ di dimensione $(M \times k)$, delle cui colonne vengono calcolati i quantili di ordine $\frac{\alpha}{2}$ e $1 - \frac{\alpha}{2}$.

Tali funzioni restituiscono pertanto k estremi degli intervalli di previsione simulati in una coppia di *array*, che vengono sovrapposti alla linea riferita alle previsioni

in ogni grafico realizzato.

3.3.2 Lisciamento Esponenziale Semplice

Il Lisciamento Esponenziale Semplice è stato implementato in JavaScript con una funzione che, data la serie storica di lunghezza n e il valore da attribuire al parametro α, restituisce il vettore di lunghezza $(n + k)$ della serie storica *perequata*, dove con k indichiamo l'orizzonte temporale di previsione. La funzione inizializza l'algoritmo e segue le formule di aggiornamento già descritte in **2.2.1**. All'interno del grafico creato nel progetto HTML viene disegnata quindi la linea corrispondente alle k previsioni calcolate in questo modo, insieme agli intervalli di previsione simulati ottenuti come descritto in **3.2.1**.

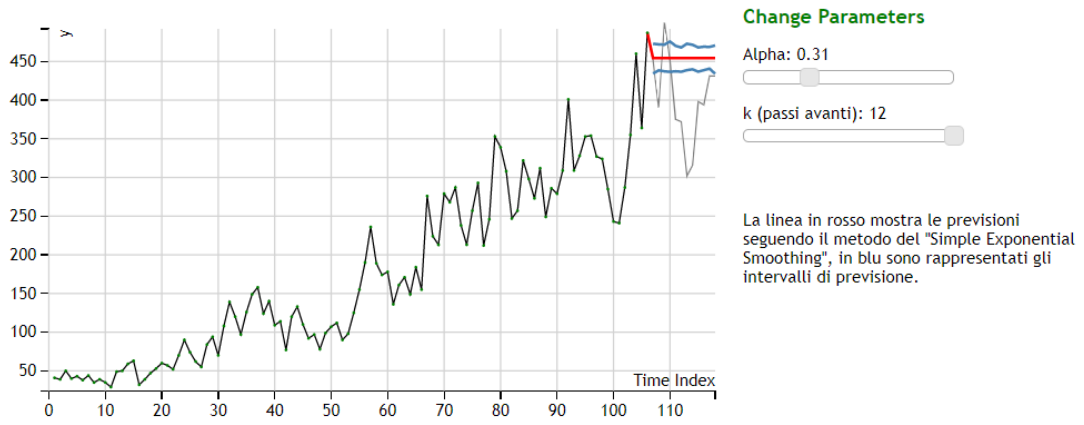

Figura 3.3: *Grafico della serie storica "Crime2" dalla libreria R "tsdl" con previsioni ottenute dal Lisciamento Esponenziale Semplice. Visualizzazione su browser.*

Si vede anche graficamente che le previsioni ottenute con il Lisciamento Esponenziale Semplice sono costanti e lo sono in maniera approssimativa anche i relativi intervalli di previsione, i quali non "contengono" neppure i valori osservati della serie nel sottoinsieme di verifica. A riprova di ciò, dai risultati degli indici per valutare la qualità delle previsioni risulta che l'errore è dovuto in maggior parte

alla diversa variabilità della serie osservata nel sottoinsieme di verifica e delle previsioni, le quali, essendo costanti, hanno varianza pari a zero (Figura 3.4). Questo è dovuto alla presenza di una componente di trend e di una di stagionalità nella serie in esame, per la quale questo metodo di previsione risulta chiaramente poco adeguato.

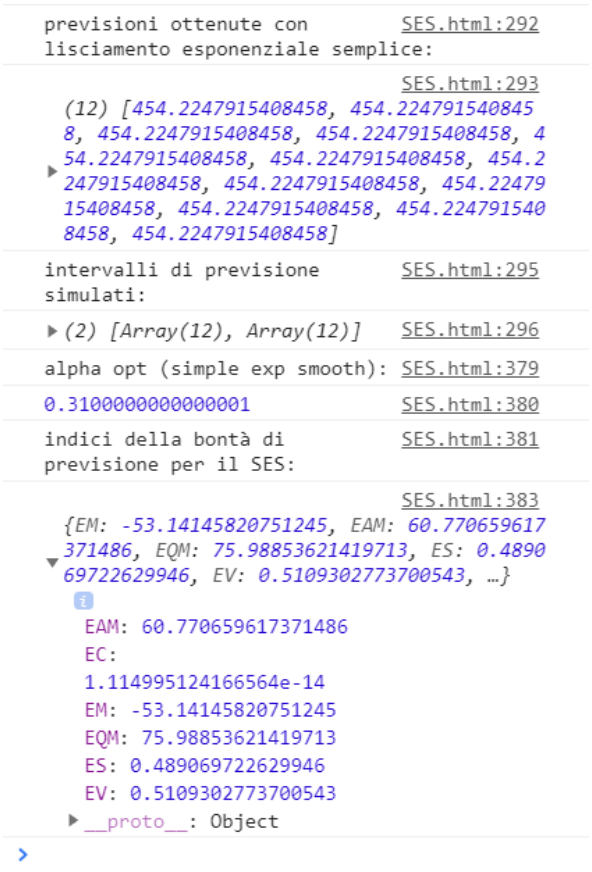

Figura 3.4: *Risultati numerici stampati su console, ottenuti dall'applicazione del Lisciamento Esponenziale Semplice sulla serie "Crime2" dalla libreria R "tsdl".*

Per quanto riguarda il valore ottimo del parametro α, è stata creata una funzione specifica in cui si calcola la somma degli errori al quadrato tra il vettore della serie di stima e quello risultante dall'applicazione del Lisciamento Esponenziale Semplice con un valore di α fissato sulla serie, ripetendo quindi questo calcolo per i valori di α compresi tra 0.01 e 0.99 (incrementando sequenzialmente il parametro di 0.01) e confrontando ad ogni ciclo il valore della somma degli

errori al quadrato calcolato con quello precedente, individuandone il minimo. La funzione, di conseguenza, restituisce e stampa sulla *console* il valore di α che minimizza la funzione obiettivo (Figura 3.4).
Tra i dettagli grafici implementati, si è reso necessario inserire uno *slider* (cursore) interattivo che regoli il valore assunto dal parametro α e uno che regoli il numero di istanti temporali da prevedere, k. In questo modo il cliente è in grado di "aggiustare" i valori dei parametri manualmente, potendo osservare ad ogni modifica l'aggiornamento sia delle stime numeriche che delle linee sul grafico, inoltre questo permette di modificare facilmente tale valore al giungere di nuove osservazioni, ricalcolandolo ad ogni tempo.

3.3.3 Lisciamento Esponenziale Doppio

Questa metodologia di Lisciamento è stata implementata in una funzione che, analogamente al Lisciamento Esponenziale Semplice, riceve in input il vettore della serie dei dati, i valori dei parametri α e β e il numero k di passi avanti; dopo aver inizializzato le successioni di $[\hat{L}_t, \hat{T}_t]$, aggiorna ricorsivamente le loro stime seguendo lo schema illustrato nella sezione **2.2.2**. La funzione restituisce il vettore di lunghezza k che viene rappresentato graficamente nella linea delle previsioni, sovrapposta al quella delle osservazioni del sottoinsieme di stima (Figura 3.5).
In questo grafico viene aggiunta anche la linea riferita alle previsioni *in-sample* ottenute con la variante *"Damped"* del Lisciamento Esponenziale Doppio, tramite una funzione alla quale bisogna passare in input, oltre ai parametri propri del Lisciamento Esponenziale Doppio, anche il valore da attribuire al parametro ϕ, seguendo quanto riportato nel paragrafo relativo.

Le previsioni ottenute seguendo il Lisciamento Doppio sono rappresentate da una retta inclinata positivamente, accompagnata da intervalli di previsione più ampi all'aumentare dell'orizzonte temporale, che denotano quindi una previsione meno affidabile nel lungo periodo. La frazione di errore più rilevante è quella dovuta alla variabilità, come si vede dal calcolo degli indici circa la natura dell'errore complessivo (Figura 3.6).
Ad ogni modifica dei valori dei parametri α, β e ϕ tramite gli appositi *slider*, si

possono osservare nel grafico gli aggiornamenti delle linee delle previsioni assieme agli intervalli di previsione simulati, questi ultimi riferiti solo alla versione "non smorzata".

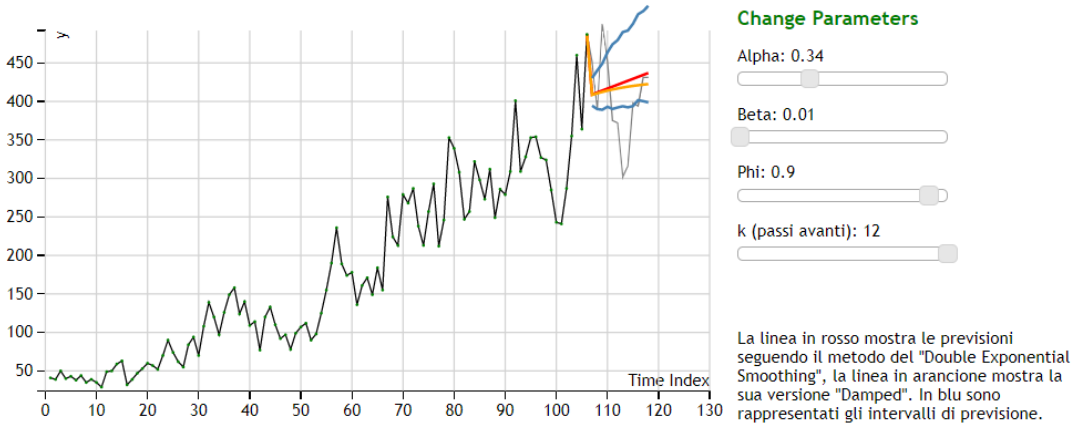

Figura 3.5: *Grafico della serie storica "Crime2" dalla libreria R "tsdl" con previsioni ottenute dal Lisciamento Esponenziale Doppio. Visualizzazione su browser.*

Il valore da attribuire ai parametri viene ottimizzato minimizzando rispetto ad α e β la somma degli errori quadratici tra la serie di test e la serie delle previsioni ottenuta col Lisciamento Esponenziale Doppio, analogamente a quanto fatto per il Lisciamento Semplice. Tuttavia, dal momento che in questo caso bisogna effettuare un'ottimizzazione in due parametri, la ricerca del minimo è stata implementata mediante una *grid-search*, una tecnica di ottimizzazione di parametri e iper-parametri molto diffusa per la sua semplicità ed efficacia.
Seguendo questo metodo, si dividono gli intervalli in cui possono assumere valori i parametri in valori equi-spaziati, e si salvano in una griglia contenente tutte le possibili combinazioni di valori. Dividendo l'intervallo $[0, 1]$ in 100 valori, si ottiene quindi una matrice 100x100 in cui ogni cella contiene la coppia dei valori dei parametri α, β, che vengono incrementati sequenzialmente di 0.01 (più propriamente, si ottiene un *array* a 3 dimensioni, in cui la terza dimensione è data dal

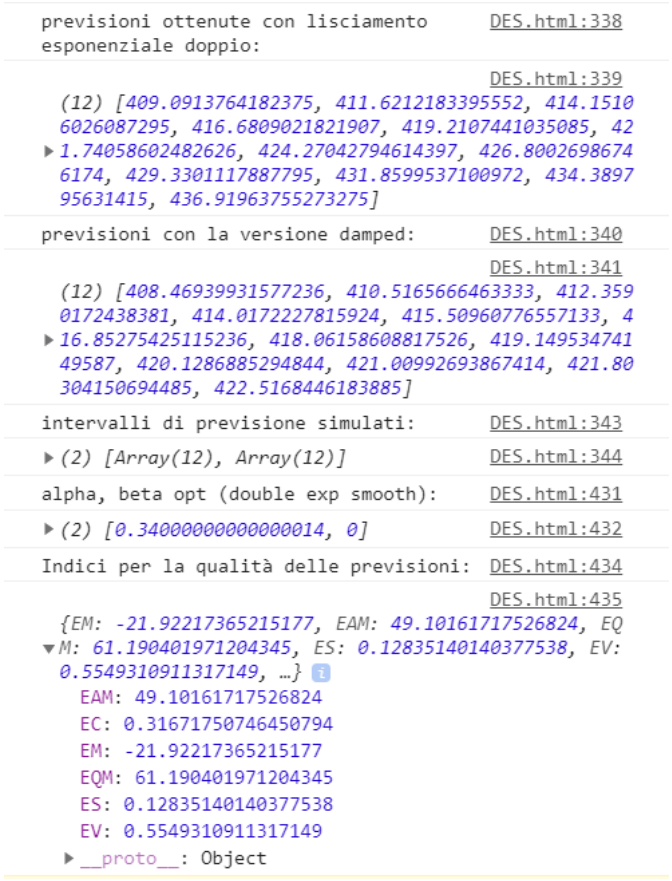

Figura 3.6: *Risultati numerici stampati su console, ottenuti dall'applicazione del Lisciamento Esponenziale Doppio sulla serie storica "Crime2" dalla libreria R "tsdl".*

fatto che ogni cella della matrice sia in realtà un vettore). Viene calcolato quindi il valore della somma degli errori al quadrato per ogni cella, effettuando 100×100 confronti col fine di individuare la cella corrispondente al minimo assoluto. La funzione infine stampa la coppia dei valori ottimali di (α, β) sulla *console* (Figura 3.6).

Per quanto riguarda il parametro di smorzamento ϕ, è consigliato impostare il suo valore intorno allo 0.9 poiché per valori minori si ottiene uno "smorzamento" troppo pronunciato.

3.3.4 Lisciamento Esponenziale Triplo

Per quanto riguarda il metodo di Holt-Winters, essendo distinto in *modello stagionale additivo* e *modello stagionale moltiplicativo*, sono stati creati due diversi grafici per presentare le funzioni realizzate per ciascun metodo di previsione.
In entrambi i casi le funzioni ricevono in ingresso la serie storica in esame, i valori dei parametri α, β e γ, il numero k di passi avanti da prevedere e il valore s corrispondente al periodo della stagionalità.
Le funzioni inizializzano gli algoritmi come descritto nella sezione **2.2.3** per poi aggiornare ricorsivamente le stime delle componenti del trend, livello e stagionalità ed, infine, calcolare le k previsioni. Queste ultime vengono restituite in un vettore che, come per i grafici realizzati precedentemente, viene aggiunto nella linea delle previsioni sovrapposta a quella del sottoinsieme di verifica. (Figure 3.7 e 3.8)

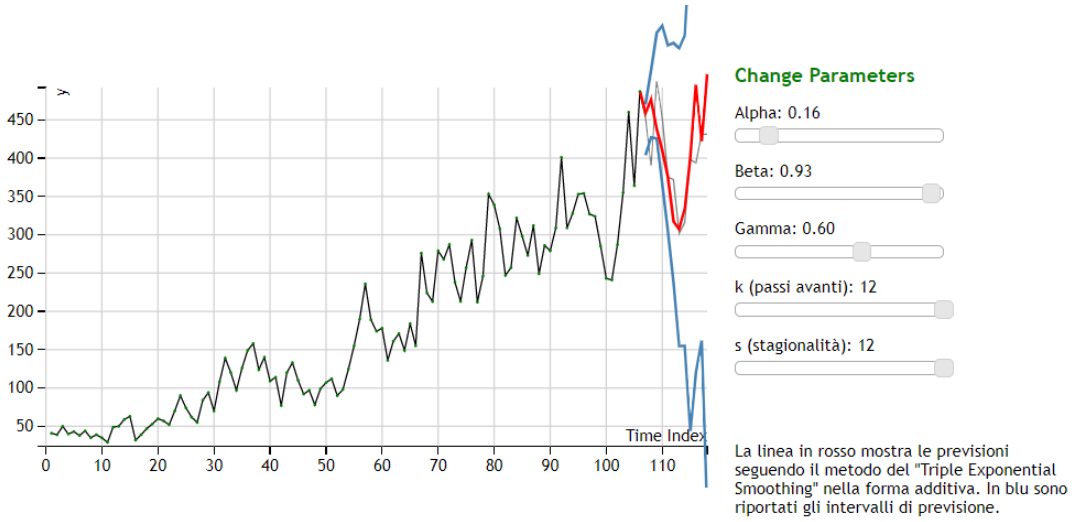

Figura 3.7: *Grafico della serie storica "Crime2" dalla libreria R "tsdl" con previsioni ottenute dal Lisciamento Esponenziale Triplo con stagionalità additiva. Visualizzazione su browser.*

La linea delle previsioni ottenuta con il metodo di Holt-Winters è chiaramente quella che meglio si adatta alla linea delle osservazioni del sottoinsieme di verifica della serie, producendo l'errore di previsione più basso: si ha infatti un errore

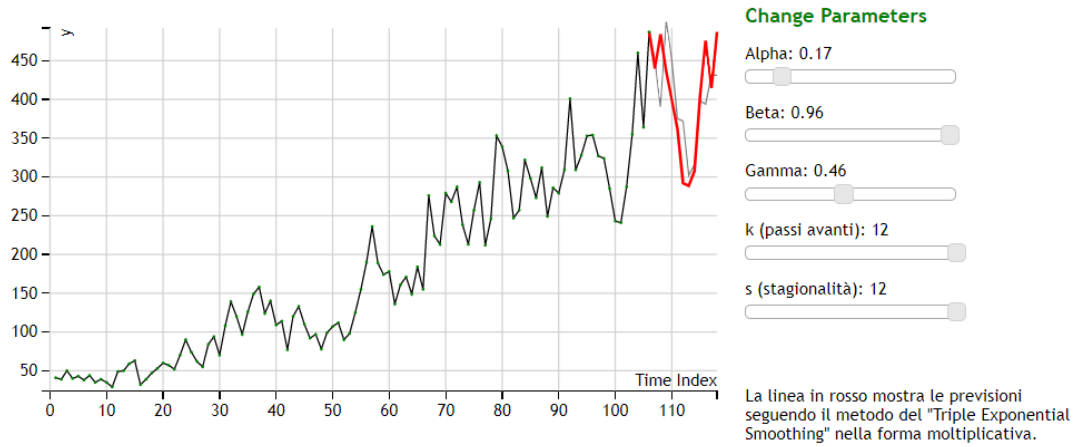

Figura 3.8: *Grafico della serie storica "Crime2" dalla libreria R "tsdl" con previsioni ottenute dal Lisciamento Esponenziale Triplo con stagionalità moltiplicativa. Visualizzazione su browser.*

assoluto di previsione (EAM) pari a 38.92 per il modello additivo e pari a 41.29 per il modello moltiplicativo, contro i valori di 60.77 e 49.10 rispettivamente per il Lisciamento Esponenziale Semplice e Doppio. Inoltre, il 91 $\sim$ 92% dell'errore quadratico medio di previsione al quadrato (EQM^2) per i metodi di Holt-Winters è dato dall'errore nelle covarianze: questo indica che la quasi totalità dell'errore può essere attribuito a fattori accidentali.

Anche per questo metodo, in maniera del tutto analoga alla funzione realizzata per il Lisciamento Esponenziale Doppio, si include una funzione che permetta di ottimizzare simultaneamente i valori dei parametri α, β e γ tramite una *grid-search*, con la quale si crea una "griglia" in tre dimensioni in cui ogni cella contiene la tripla di valori corrispondenti ai tre parametri. L'algoritmo di Lisciamento viene ripetuto 100^3 volte e vengono eseguiti altrettanti confronti tra i valori della somma degli errori quadratici di previsione; infine viene restituita e stampata su *console* la tripla relativa al minimo assoluto.

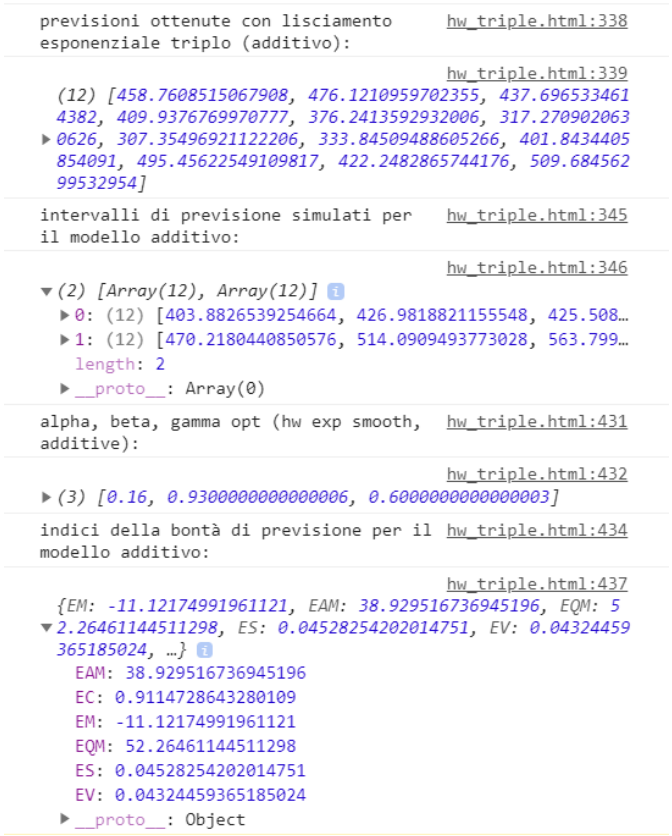

Figura 3.9: *Risultati numerici stampati su console, ottenuti dall'applicazione del Lisciamento Esponenziale Triplo con stagionalità additiva sulla serie "Crime2" dalla libreria R "tsdl"*

3.4 Processi ARIMA

La stima dei parametri di un modello ARIMA coinvolge, come accennato nella sezione 2.3, la funzione di verosimiglianza condizionata, la cui stesura ed implementazione risultano essere molto complicate, specie per valori elevati dei parametri p, d, q. Per questo motivo non è stata implementata nessuna funzione che stimasse tale modello, piuttosto si è preferito sfruttare la libreria arima [1] già esistente. Si tratta in realtà di un pacchetto chiamato *"ctsa"*, scritto in C e re-implementato tramite *Emscripten* in modo che possa essere eseguito nei browser web.

[1] si fa riferimento alla vecchia versione, la nuova versione è stata creata dopo il termine dello stage

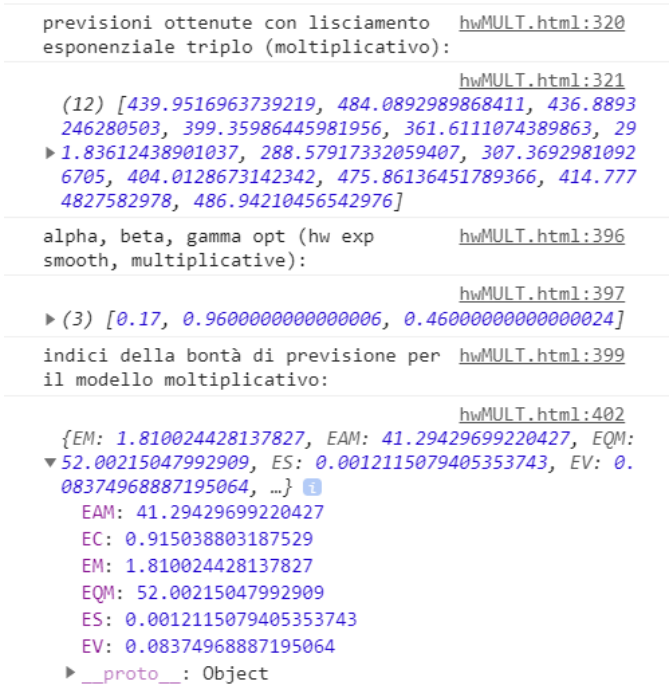

Figura 3.10: *Risultati numerici stampati su console, ottenuti dall'applicazione del Lisciamento Esponenziale Triplo con stagionalità moltiplicativa sulla serie "Crime2" dalla libreria R "tsdl"*

La relativa funzione *arima* si presenta come:

```
function arima(serie_train,k, {
        method: 0,
        optimizer: 6,
        p: 1,
        d: 0,
        q: 1,
        verbose: true})
```

Essa riceve in input il vettore di osservazioni della serie storica, il numero k di passi avanti per la previsione e i valori di p, d, q del modello. La funzione utilizza di default il metodo della Massima Verosimiglianza Esatta e un algoritmo di ottimizzazione numerica a memoria limitata, il *Limited-memory BFGS*, per la stima dei coefficienti del modello; tuttavia si tratta di impostazioni che possono essere modificate inserendo un valore diverso da quello di default rispettivamente nelle

variabili "*method*" e "*optimizer*".

La funzione restituisce in output il vettore delle k previsioni ottenute e il vettore degli errori quadratici medi ($MSE[e_{t+k}]$); inoltre, vengono stampati sulla console i coefficienti stimati del modello con la rispettiva deviazione standard, la varianza degli errori, la forma esplicita dell'equazione del modello, il valore della log-verosimiglianza ed infine il valore del *Criterio di Informazione di Akaike (AIC)*. In aggiunta, questi risultati vengono integrati stampando su *console* anche la serie storica in esame nella sua distinzione tra sottoinsieme di stima e di verifica, la serie delle previsioni e gli intervalli di previsione ottenuti (Figure 3.12 e 3.13).

Analogamente a quanto fatto con il Lisciamento Esponenziale, dopo aver disegnato il grafico della serie storica distinta in serie di stima e serie di verifica, si sovrappongono la linea corrispondente alle previsioni (in rosso) insieme ai rispettivi intervalli di previsione (in blu), ottenuti tramite la consueta formula:

$$\hat{y}_{n+k} \pm z_{1-\frac{\alpha}{2}} \sigma[e_{t+k}],$$

dove $z_{1-\frac{\alpha}{2}}$ indica il percentile di ordine $1 - \frac{\alpha}{2}$ di una distribuzione Normale Standard (in questo caso, è stato usato $\alpha = 0.05$).

All'interno della pagina web si è scelto di inserire nuovamente dei cursori (*slider*) per regolare i valori dei parametri p, d, q del modello ARIMA e il numero k di passi avanti, di conseguenza il grafico e ogni risultato si aggiornano in automatico.

L'individuazione del modello adatto a rappresentare il processo generatore dei dati in esame è una fase fondamentale nell'applicazione dei modelli ARIMA su una serie storica, ragion per cui, specie in un contesto aziendale, è conveniente servirsi di una funzione che sia in grado di effettuare una selezione automatica del modello tramite specifici criteri che aiutino ad avere un buon adattamento del modello sui dati, senza tuttavia avere un caso di *overfitting*. Per questa ragione si utilizzano criteri che penalizzino il numero di parametri, come l'AIC o il BIC, a differenza della funzione della somma degli errori quadratici, utilizzata per il Lisciamento Esponenziale, che invece porterebbe ad un adattamento eccessivo

ai dati con una variabilità della stima delle previsioni conseguentemente molto ampia.

Utilizzando la libreria ARIMA citata, non è possibile estrarre il valore dell'AIC stampato sulla *console*, ragion per cui non è stata realizzata una funzione "auto-arima" in grado di minimizzare tale criterio per la selezione del modello.

In questo caso specifico, la stima di un modello ARIMA(p,d,q) non risulta adeguata dal momento che la serie in esame presenta una stagionalità di periodo 12 e sarebbe pertanto più opportuno adattare un modello ARIMA stagionale, ovvero un SARIMA(p,d,q)(P,D,Q).

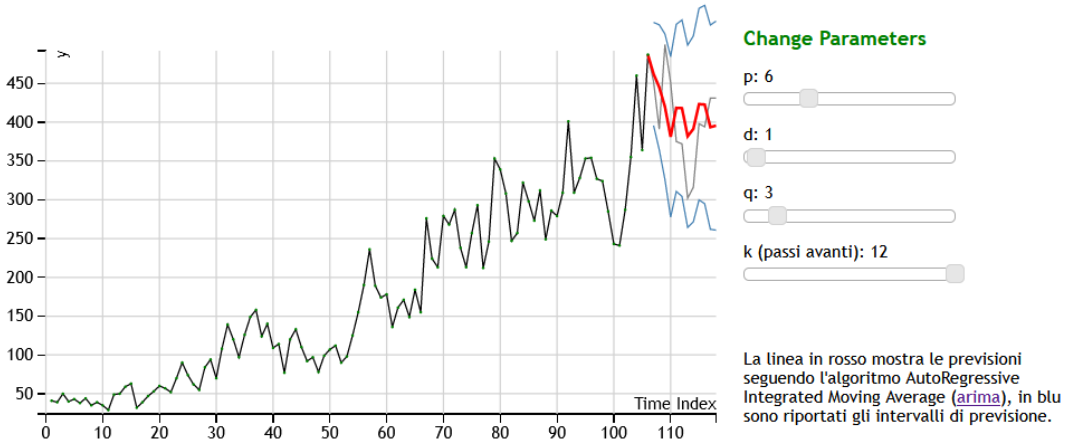

Figura 3.11: *Grafico della serie storica "Crime2" dalla libreria R "tsdl" con previsioni ottenute dall'applicazione del modello ARIMA(6,1,3). Visualizzazione su browser.*

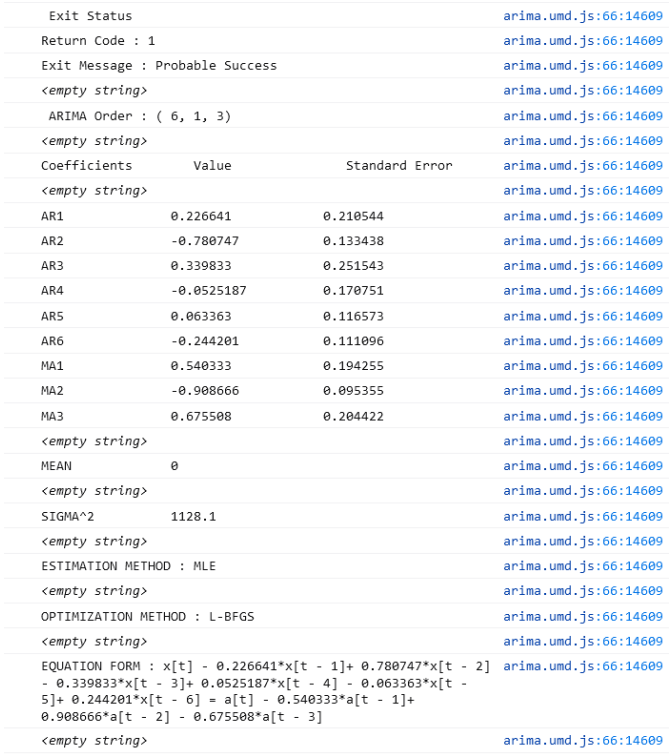

Figura 3.12: *Risultati numerici stampati su console, ottenuti dall'applicazione di un ARIMA(6,1,3) sulla serie "Crime2" dalla libreria R "tsdl".*

Figura 3.13: *Risultati numerici stampati su console, ottenuti dall'applicazione di un ARIMA(6,1,3) sulla serie "Crime2" dalla libreria R "tsdl".*

Conclusioni

In questa relazione si descrive nel dettaglio quanto realizzato durante l'esperienza di stage presso Zucchetti S.p.A.. In particolare, sono state realizzate delle interfacce veloci ed intuitive in grado di restituire e visualizzare delle previsioni in serie storiche, tramite l'applicazione di metodi di previsione quali la classe di modelli ARIMA e il Lisciamento Esponenziale, molto utilizzato in ambito aziendale per la sua flessibilità e accessibilità d'impiego.
La visualizzazione immediata dei risultati sul grafico, affiancata agli indici per la valutazione delle qualità previsive e agli intervalli di previsione, rendono più semplice il confronto tra diversi algoritmi permettendo di selezionare quello più adeguato a rappresentare il processo generatore dei dati. Tali strumenti risultano utili e convenienti per tutti i clienti che vogliano visualizzare diversi scenari di sviluppo in uno specifico fenomeno. In aggiunta, il codice JavaScript viene eseguito dal browser dell'utente (ossia dal *client*), per cui non si verifica nessun sovraccarico del *server*, riducendo i tempi di caricamento della pagine.
Chiaramente, le interfacce realizzate rappresentano uno "scheletro" per un unico *software* più ampio, dalla grafica completa ed ottimizzata, implementando altre funzionalità statistiche circa l'analisi della serie storica. Si fa notare, inoltre, che la libreria utilizzata per la stima del modello ARIMA è stata recentemente aggiornata ed ampliata con l'integrazione dei modelli SARIMA, SARIMAX e della funzionalità di selezione automatica del modello (auto-arima).

Appendice

In questa sezione vengono riportate e descritte tutte le funzioni implementate in JavaScript ed utilizzate ai fini del progetto di stage.

3.5 Misure statistiche

3.5.1 Somma

```
sum(array,inizio,fine)
```

- **Input:**
  - *array*, vettore numerico di dimensione n
  - *inizio*, intero, indice dal quale partire
  - *fine*, intero, indice finale
- **Descrizione:** Calcola la somma di un vettore di numeri, dal j-esimo valore, con j indicato nella variabile *inizio*, fino al k-esimo valore, con k indicato nella variabile *fine*.
- **Formula:**

$$\sum_{i=j}^{k} y_i$$

- **Output:** scalare

3.5.2 Somma parziale di una serie geometrica

```
geom_parziale(phi,h)
```

- **Input:**
  - *phi*, scalare
  - *h*, intero, indice finale
- **Descrizione:** Calcola la somma parziale di una serie geometrica di ragione ϕ, ossia una serie geometrica finita con somma dall'indice 1 all'indice h.

- **Formula:**

$$\sum_{i=1}^{h} \phi^i$$

- **Output:** scalare

3.5.3 Media Aritmetica

```
mean(array)
```

- **Input:**

  - *array*, vettore numerico di dimensione n

- **Descrizione:** Calcola la media aritmetica di un vettore di numeri.

- **Formula:**

$$\frac{1}{n}\sum_{i=1}^{n} y_i$$

- **Output:** scalare

3.5.4 Scarto quadratico medio

```
sigma(array)
```

- **Input:**

  - *array*, vettore numerico di dimensione n

- **Descrizione:** Calcola la deviazione standard non corretta, ossia lo scarto quadratico medio di un vettore di numeri.

- **Formula:**

$$\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2\right)^{1/2} = \sigma_y$$

- **Output:** scalare

3.5.5 Somma degli errori al quadrato

```
SSE(array1,array2)
```

- **Input:**

  - *array1*, vettore numerico di dimensione n

  - *array2*, vettore numerico di dimensione n

- **Descrizione:** Calcola la somma degli errori al quadrato tra due vettori di numeri. Misura la discrepanza tra i dati osservati (array1) e una loro stima (array2).

- **Formula:**

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **Output:** scalare

3.5.6 Regressione Lineare

```
linearRegression(y,x)
```

- **Input:**

  - y, vettore numerico di lunghezza n, variabile dipendente

  - x, vettore numerico di lunghezza, variabile indipendente n

- **Descrizione:** Si esegue una regressione lineare semplice secondo il metodo dei minimi quadrati ordinari (OLS) tra due vettori di numeri, Y_i variabile dipendente e X_i regressore.

- **Formula:**

$$modello\ lineare: \quad Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

$$retta\ di\ regressione: \beta_0 + \beta_1 X$$

$$\hat{\beta}_1 = \frac{n\sum_i x_i y_i - \sum_i x_i \sum_i y_i}{n\sum_i x_i^2 - \left(\sum_i x_i\right)^2} = \frac{\sigma_{x,y}}{\sigma_x^2}$$

$$\hat{\beta}_0 = \bar{y} - \beta_1 \bar{x}$$

- **Output:** oggetto *lr*:

  - *slope*: scalare, $\hat{\beta}_1$

  - *intercept*: scalare, $\hat{\beta}_0$

3.5.7 Generazione valore casuale distribuito Normalmente

```
rnorm(mean,sd)
```

- **Input:**

  - *mean*, scalare, la media μ della distribuzione da cui campionare

  - *sd*, scalare, la deviazione standard σ della distribuzione da cui campionare

- **Descrizione:** Genera un valore pseudo-casuale da una distribuzione Normale di media μ e varianza σ^2.

  JavaScript possiede, tra gli oggetti standard predefiniti, la funzione *Math.random()* che ritorna un numero pseudo-casuale in virgola mobile compreso nell'intervallo [0, 1). Si sfrutta quindi la Trasformazione di Box-Muller per ricavare, partendo da due valori distribuiti Uniformemente sull'intervallo $(0, 1]$, un valore distribuito come una Normale Standard, che viene poi opportunamente riscalato per seguire una distribuzione Normale di media μ e varianza σ^2.

- **Formula:**

$$Z = \sqrt{-2 \ln U_1} \cos 2\pi U_2,$$

  dove U_1, U_2 sono due variabili casuali indipendenti ed uniformemente distribuite nell'intervallo $(0, 1]$.

- **Output:** scalare, $x \sim N(\mu, \sigma^2)$.

3.6 Modelli serie storiche

3.6.1 Simple Exponential Smoothing

```
SES(serie,alpha,k)
```

- **Input:**
  - *serie*, vettore numerico di lunghezza n, l'insieme di stima
  - *alpha*, scalare, valore della costante di Lisciamento
  - k, intero, numero di istanti di previsione
- **Descrizione:** Applica il Lisciamento Esponenziale Semplice alla serie di dati inserita, producendo k previsioni.
- **Formula:** Si rimanda alla sezione 2.2.1.
- **Output:** vettore numerico di lunghezza $(n+k)$, contenente le n osservazioni "perequate" e le k previsioni.

3.6.2 Intervalli di previsione simulati

```
pred_int_SES(serie,alpha,k,sd)
```

- **Input:**
  - *serie*, vettore numerico di lunghezza n, l'insieme di stima
  - *alpha*, scalare, valore della costante di Lisciamento
  - k, intero, numero di istanti di previsione
  - *sd*, scalare, deviazione standard dell'errore
- **Descrizione:** Stima k estremi degli intervalli di previsione simulati riferiti alle previsioni ottenute con il Lisciamento Esponenziale Semplice.
- **Formula:** Si rimanda alla sezione 2.2.
- **Output:** coppia di array ciascuno di lunghezza k, contenente gli estremi degli intervalli di previsione.

3.6.3 Stima di alpha

```
alpha_opt(serie)
```

- **Input:**
  - *serie*, vettore numerico di lunghezza n, l'insieme di stima

- **Descrizione:** Stima il valore di α ottimale, minimizzando la somma degli errori al quadrato tra la serie originale e la serie ottenuta con la funzione descritta sopra.
- **Formula:** Si rimanda alla sezione 2.2.1.
- **Output:** scalare, $\hat{\alpha}$.

3.6.4 Double Exponential Smoothing

```
Holt(serie,alpha,beta,k)
```

- **Input:**
  - *serie*, vettore numerico di lunghezza n, l'insieme di stima
  - *alpha*, scalare, valore della costante di Lisciamento per il livello
  - *beta*, scalare, valore della costante di Lisciamento per il trend
  - k, intero, numero di istanti di previsione
- **Descrizione:** Applica il Lisciamento Esponenziale Doppio alla serie di dati inserita, producendo k previsioni.
- **Formula:** Si rimanda alla sezione 2.2.2.
- **Output:** vettore numerico di lunghezza k, contenente le k previsioni.

3.6.5 Double Damped Exponential Smoothing

```
Damped(serie,alpha,beta,phi,k)
```

- **Input:**
  - *serie*, vettore numerico di lunghezza n, l'insieme di stima
  - *alpha*, scalare, valore della costante di Lisciamento per il livello
  - *beta*, scalare, valore della costante di Lisciamento per il trend
  - *phi*, scalare, valore della costante di smorzamento
  - k, intero, numero di istanti di previsione
- **Descrizione:** Applica il Lisciamento Esponenziale Doppio nella sua variante "smorzata" alla serie di dati inserita, producendo k previsioni.
- **Formula:** Si rimanda al paragrafo "Damped ..." nella sezione 2.2.2.
- **Output:** vettore numerico di lunghezza k, contenente le k previsioni.

3.6.6 Intervalli di previsione simulati

```
pred_int_DES(serie,alpha,beta,k,sd)
```

- **Input:**
  - *serie*, vettore numerico di lunghezza n, l'insieme di stima
  - *alpha*, scalare, valore della costante di Lisciamento per il livello
  - *beta*, scalare, valore della costante di Lisciamento per il trend
  - k, intero, numero di istanti di previsione
  - *sd*, scalare, deviazione standard dell'errore
- **Descrizione:** Stima k estremi degli intervalli di previsione simulati riferiti alle previsioni ottenute con il Lisciamento Esponenziale Doppio.
- **Formula:** Si rimanda alla sezione 2.2.
- **Output:** coppia di array ciascuno di lunghezza k, contenente gli estremi degli intervalli di previsione.

3.6.7 Stima di alpha e beta

```
ab_opt(serie)
```

- **Input:**
  - *serie*, vettore numerico di lunghezza n, l'insieme di stima
- **Descrizione:** Stima i valori di α e β ottimali per il Lisciamento Esponenziale Doppio, minimizzando tramite grid-search la somma degli errori al quadrato tra la serie originale e la serie ottenuta con la funzione descritta sopra.
- **Formula:** Si rimanda alla sezione 2.2.2.
- **Output:** array contenente due valori, rispettivamente $\hat{\alpha}$ e $\hat{\beta}$.

3.6.8 Triple Exponential Smoothing, additive

```
HoltWint_ADD(serie,alpha,beta,gamma,k,s)
```

- **Input:**
  - *serie*, vettore numerico di lunghezza n, l'insieme di stima
  - *alpha*, scalare, valore del primo parametro per il livello
  - *beta*, scalare, valore del secondo parametro per il tren
  - *gamma*, scalare, valore del parametro per la componente stagionale
  - k, intero, numero di istanti di previsione
  - s, intero, periodo della stagionalità
- **Descrizione:** Applica il Lisciamento Esponenziale Triplo alla serie di dati inserita, assumendo stagionalità di tipo additivo. Produce k previsioni.
- **Formula:** Si rimanda alla sezione 2.2.3.
- **Output:** vettore numerico di lunghezza k, contenente le k previsioni.

3.6.9 Triple Exponential Smoothing, multiplicative

```
HoltWint_MULT(serie,alpha,beta,gamma,k,s)
```

- **Descrizione:** Applica il Lisciamento Esponenziale Triplo alla serie di dati inserita, assumendo stagionalità di tipo moltiplicativo. Produce k previsioni. Il resto è uguale alla funzione *HoltWint_ADD()* in 3.6.8.

3.6.10 Intervalli di previsione simulati

```
pred_int_HWA(serie,alpha,beta,gamma,k,s,sd)
```

- **Input:**
  - *serie*, vettore numerico di lunghezza n, l'insieme di stima
  - *alpha*, scalare, valore della costante di Lisciamento per il livello
  - *beta*, scalare, valore della costante di Lisciamento per il trend
  - *gamma*, scalare, valore della costante di Lisciamento per la stagionalità
  - k, intero, numero di istanti di previsione
  - s, scalare, periodo della stagionalità
  - *sd*, scalare, deviazione standard dell'errore
- **Descrizione:** Stima k estremi degli intervalli di previsione simulati riferiti alle previsioni ottenute con il Lisciamento Esponenziale Triplo assumendo stagionalità additiva.
- **Formula:** Si rimanda alla sezione 2.2.
- **Output:** coppia di array ciascuno di lunghezza k, contenente gli estremi degli intervalli di previsione.

3.6.11 Stima di alpha, beta e gamma, modello additivo

```
abg_add_opt(serie_tr,serie_te)
```

- **Input:**

  - *serie_tr*, vettore numerico di lunghezza n, l'insieme di stima

  - *serie_te*, vettore numerico di lunghezza $N-n$, dove N indica la lunghezza della serie completa; l'insieme di verifica

- **Descrizione:** Stima i valori ottimali di α, β e γ per il Lisciamento Esponenziale Triplo nel caso additivo, minimizzando tramite grid-search la somma degli errori al quadrato tra la serie di stima e la serie ottenuta con la funzione in 3.6.8.

- **Formula:** Si rimanda alla sezione 2.2.3.

- **Output:** array con tre valori, rispettivamente $\hat{\alpha}$, $\hat{\beta}$ e $\hat{\gamma}$.

3.6.12 Stima di alpha, beta e gamma, modello moltiplicativo

```
abg_mult_opt(serie_tr,serie_te)
```

- **Descrizione:** Stima i valori ottimali di α, β e γ per il Lisciamento Esponenziale Triplo nel caso moltiplicativo, analogamente alla funzione *abg_add_opt() in 3.6.11.*

3.6.13 Qualità delle previsioni

```
bonta_prev(serie_test,previsioni)
```

- **Input:**

  - *serie_test*, vettore numerico di lunghezza $(N-n)$, dove N indica la lunghezza della serie completa, il testing set. In questo caso, $(N-n)$ deve coincidere con k.

  - *previsioni*, vettore numerico di lunghezza k, le previsioni stimate

- **Descrizione:** Calcola alcuni importanti indici per misurare la qualità delle previsioni ottenute.
- **Formula:** Si rimanda al paragrafo "Qualità delle previsioni".
- **Output:** oggetto *bp*, contenente:
  - *EM*, scalare, errore medio di previsione
  - *EAM*, scalare, errore assoluto di previsione
  - *EQM*, scalare, errore quadratico medio di previsione
  - *ES*, scalare, errore sistematico
  - *EV*, scalare, errore nelle variabilità
  - *EC*, scalare, errore nelle covarianze

Bibliografia

Capitolo 1

[1] *D3.js Documentation*. 2020. URL: https://github.com/d3/d3/wiki.

Capitolo 2

[1] Rob J Hyndman et al. *Forecasting with Exponential Smoothing, The State Space Approach*. Springer, 2008. ISBN: 978-3-540-71918-2.

[2] George E.P. Box et al. *Time Series Analysis, Forecasting and Control*. John Wiley & Sons Inc, 2015. ISBN: 978-1-118-67502-1.

[3] Tommaso Di Fonzo e Francesco Lisi. *Serie storiche economiche, analisi statistiche e applicazioni*. Carocci editore, 2005. ISBN: 88-430-3423-5.