CURS 9 - SERII DE TIMP (continuare)

EXEMPLE

```
## Varsta mortii regilor succesivi ai Angliei incepand cu William Cuceritorul
# #Sursa: McNeill, "Interactive Data Analysis"
kings<-read.csv("E:\\29_03_2018_Toshiba\\Alina_27.03.2018\\0_Docum univ\\2019-2020\\Cursuri\\Cursuri inregistrate\\Info 3\\Kings.csv", sep=",", header=F)
kings
plot(kings)
kingstimeseries <- ts(kings) ## stocarea datelor ca serie de timp
```

Uneori seria de timp este colectata lune sau trimestrial, caz in care va trebui sa introduceti un parametru care sa arate frecvanta colectarii: 12, pentru date lunare si 4 pentru celel trimestriale. Puteti specifica primul moment in care au fost colectate datele . De exemplu, daca prima data a fost colectata in trimestrul al doilea din 1986, vom introduce ca parametru de start=c(1986,2).

Exemplul al doilea (http://robjhyndman.com/tsdldata/data/nybirths.dat) contine numarul de nasteri pe luna in New York city, din Ian. 1946 pana in Dec. 1959. Iata datele:

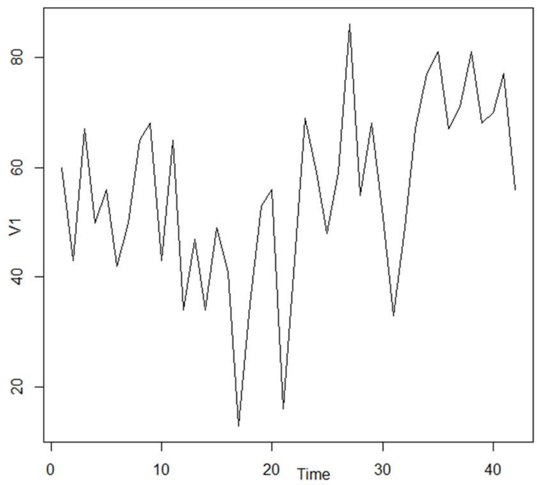
```
births <- scan("http://robjhyndman.com/tsdldata/data/nybirths.dat")
births
birthstimeseries <- ts(births, frequency=12, start=c(1946,1))
```

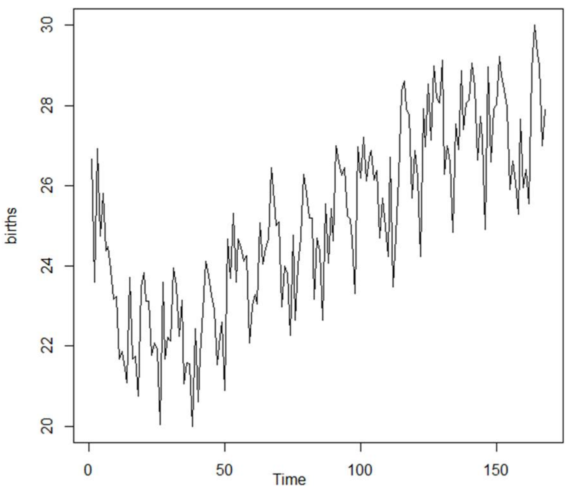
Similar, exemplul al treilea (http://robjhyndman.com/tsdldata/data/fancy.dat) contine vanzarile lunare de souveniruri dintr-un magazine de pe plaja, in Queensland, Australia, pt Ian 1987- Dec.1993.

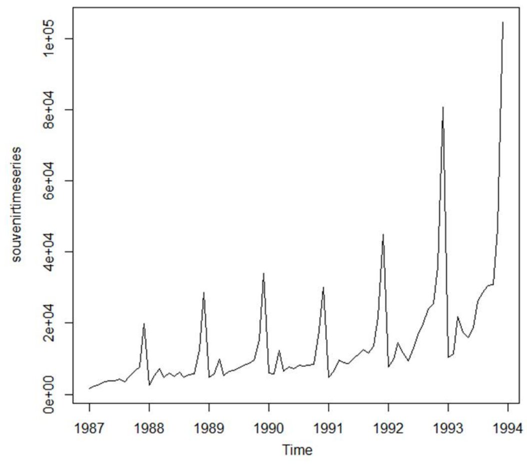
```
souvenir <- scan("http://robjhyndman.com/tsdldata/data/fancy.dat")
souvenirtimeseries <- ts(souvenir, frequency=12, start=c(1987,1))
souvenirtimeseries
```

1. ***Plotarea***

```
plot.ts(kingstimeseries)
plot(births)
plot.ts(souvenirtimeseries)
```





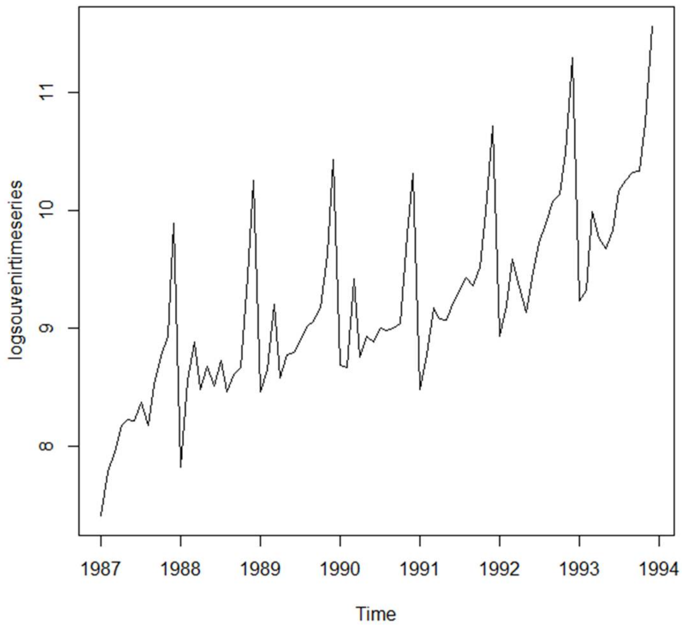


Putem sa incercam un model aditiv pt **Kings** fiindca variatiile aleatoare ale datelor sunt aproape constante ca marime in timp

Pt seria Births exista o anumita sezonalitate: exista unui varf in fiecare vara si un minim in fiecare iarna. Seria poate fi descrisa de un model aditiv deoarece fluctuatiile sezoniere sunt aproape constant in timp si nu par sa depinda de nivelul seriei, iar variatiile par a fi constant ca marime in timp.

In acest caz se pare ca un model aditiv nu este bun deoarece fluctuatiile sezoniere par sa creasca in nivel. Ca urmare am putea face o transformare pentru a ajunge la o series de timp pentru care am putea utiliza un model aditiv. Logaritmam (in baza e):

```
logsouvenirtimeseries <- log(souvenirtimeseries)
plot.ts(logsouvenirtimeseries)
```



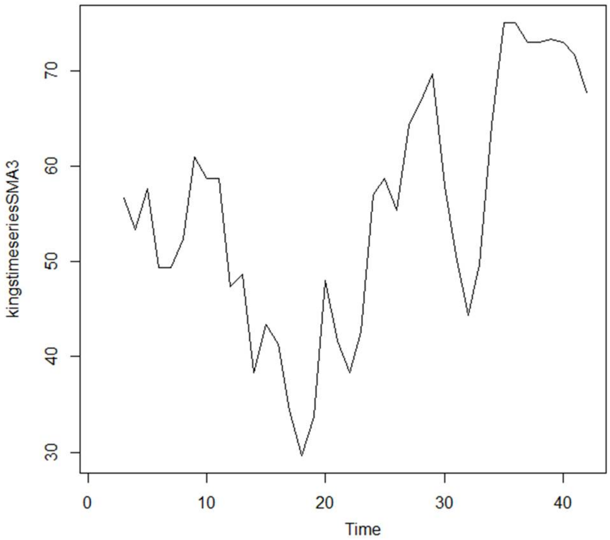
2. Descompunerea seriilor de timp

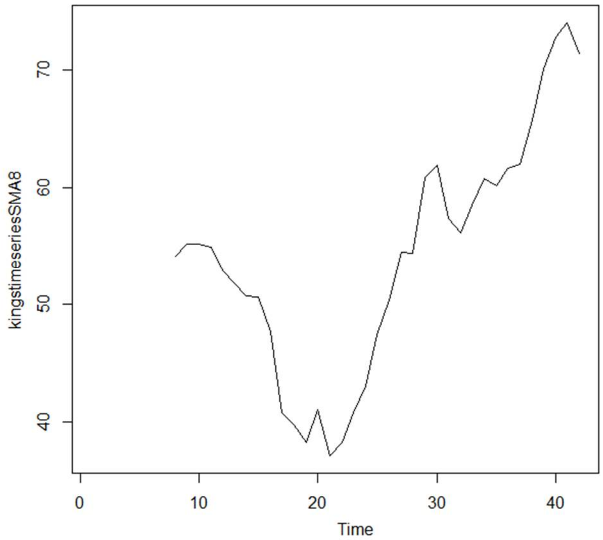
2.1. Descompunerea seriilor nesezonale – trend si componenta aleatoare

Pentru estimarea componentei nesezonale se poate folosi o metoda de netezire, ca de exemplu method mediilor mobile – se foloseste functia SMA() function din package TTR

```
library("TTR")
```

```
par(mfrow=c(2,2))
kingstimeseriesSMA3 <- SMA(kingstimeseries,n=3)
plot.ts(kingstimeseriesSMA3)
kingstimeseriesSMA8 <- SMA(kingstimeseries,n=8)
plot.ts(kingstimeseriesSMA8)
```



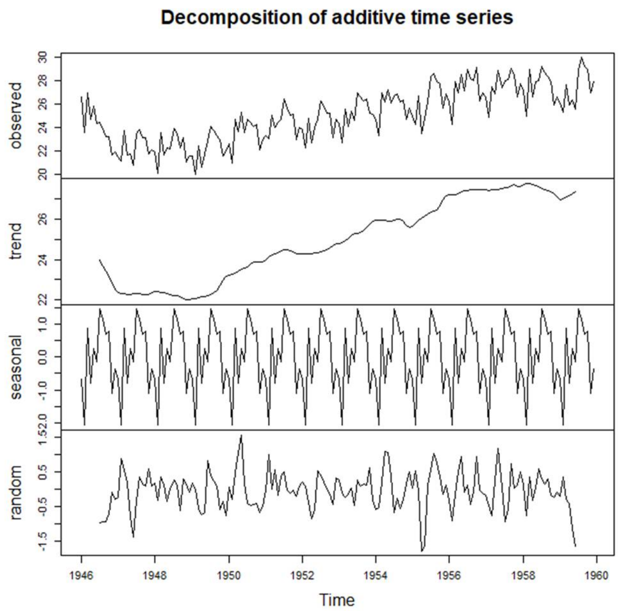


Seria netezita folosind o medie mobile de 8 da o imagine mai clara a componentei trend. Se vede ca varsta regilor a descercut de la aprox 55 la 38 ani in timpul domniei primilor 20 regi, apoi a crecut la aprox 73 ani spre sfarsitul domniei celui de al 40-lea rege.

Tema. *Descoperiti care este variatia aleatoare a seriei kings si analizati aceasta serie reziduala.*

2.2. Descompunerea seriilor sezoniere

```
birthstimeseriescomponents <- decompose(birthstimeseries)
birthstimeseriescomponents$seasonal        # da componenta  sezoniera
plot(birthstimeseriescomponents)
```



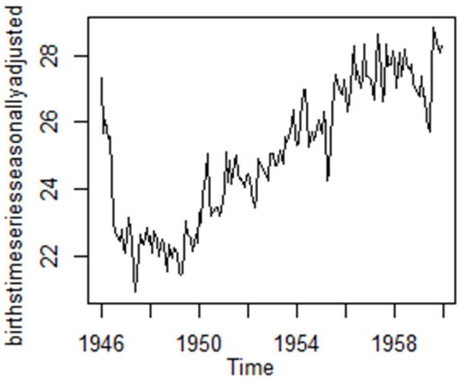
Factorii de sezonalitate estimatI sunt dati pentru lunile ianuarie-decembrie si sunt aceiasi in fiecare an. Cel mai mare este pentru iulie (aprox 1.46), cel mai mic pentru februarie (aprox. -2.08), indicand existenta unui varf de nasteri in iulie si cele mai putine nasteri in februarie.

2.3. Ajustarea sezoniera

Data o serie de timp care poate fi descrisa cu ajutorul unui model aditiv, ea poate fi ajustata sezonal prin estimarea componentei sezoniere, si scaderea componentei sezoniere din seria initiala. Se poate face acest lucru folosind functia decompose()

De exemplu, pentru seria nasterilor:

```
birthstimeseriescomponents <- decompose(birthstimeseries)
birthstimeseriesseasonallyadjusted <- birthstimeseries -  birthstimeseriescomponents$seasonal
plot(birthstimeseriesseasonallyadjusted)
```



Variatiile sezoniere au fost inlaturate din seria ajustata sezonier. Seria ajustata sezonier contine acum numai trendul si componenta aleatoare.

```
souvenir <- scan("http://robjhyndman.com/tsdldata/data/fancy.dat")
```

Read 84 items

```
souvenirtimeseries <- ts(souvenir, frequency=12, start=c(1987,1))
```

souvenirtimeseries

	Jan	Feb	Mar	Apr	May	Jun	Jul
1987	1664.81	2397.53	2840.71	3547.29	3752.96	3714.74	4349.61
1988	2499.81	5198.24	7225.14	4806.03	5900.88	4951.34	6179.12
1989	4717.02	5702.63	9957.58	5304.78	6492.43	6630.80	7349.62
1990	5921.10	5814.58	12421.25	6369.77	7609.12	7224.75	8121.22
1991	4826.64	6470.23	9638.77	8821.17	8722.37	10209.48	11276.55
1992	7615.03	9849.69	14558.40	11587.33	9332.56	13082.09	16732.78
1993	10243.24	11266.88	21826.84	17357.33	15997.79	18601.53	26155.15

	Aug	Sep	Oct	Nov	Dec
1987	3566.34	5021.82	6423.48	7600.60	19756.21
1988	4752.15	5496.43	5835.10	12600.08	28541.72
1989	8176.62	8573.17	9690.50	15151.84	34061.01
1990	7979.25	8093.06	8476.70	17914.66	30114.41
1991	12552.22	11637.39	13606.89	21822.11	45060.69
1992	19888.61	23933.38	25391.35	36024.80	80721.71
1993	28586.52	30505.41	30821.33	46634.38	104660.67

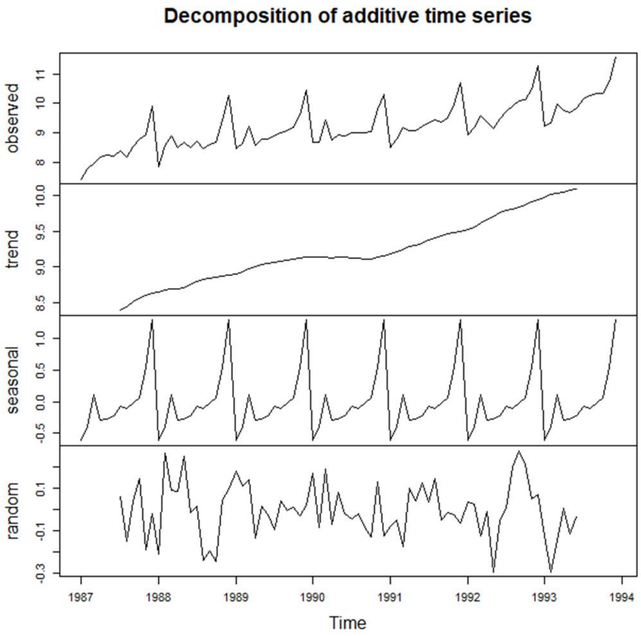
```
ST<-ts(souvenir)
```

```
ST
Time Series:
Start = 1
End = 84
Frequency = 1
 [1]   1664.81   2397.53   2840.71   3547.29   3752.96   3714.74   4349.61
 [8]   3566.34   5021.82   6423.48   7600.60  19756.21   2499.81   5198.24
[15]   7225.14   4806.03   5900.88   4951.34   6179.12   4752.15   5496.43
[22]   5835.10  12600.08  28541.72   4717.02   5702.63   9957.58   5304.78
[29]   6492.43   6630.80   7349.62   8176.62   8573.17   9690.50  15151.84
[36]  34061.01   5921.10   5814.58  12421.25   6369.77   7609.12   7224.75
[43]   8121.22   7979.25   8093.06   8476.70  17914.66  30114.41   4826.64
[50]   6470.23   9638.77   8821.17   8722.37  10209.48  11276.55  12552.22
[57]  11637.39  13606.89  21822.11  45060.69   7615.03   9849.69  14558.40
[64]  11587.33   9332.56  13082.09  16732.78  19888.61  23933.38  25391.35
[71]  36024.80  80721.71  10243.24  11266.88  21826.84  17357.33  15997.79
[78]  18601.53  26155.15  28586.52  30505.41  30821.33  46634.38 104660.67
```

```
ST<-ts(souvenir,start=c(1987,1))
```

```
ST
Time Series:
Start = 1987
End = 2070
Frequency = 1
 [1]   1664.81   2397.53   2840.71   3547.29   3752.96   3714.74   4349.61
 [8]   3566.34   5021.82   6423.48   7600.60  19756.21   2499.81   5198.24
[15]   7225.14   4806.03   5900.88   4951.34   6179.12   4752.15   5496.43
[22]   5835.10  12600.08  28541.72   4717.02   5702.63   9957.58   5304.78
[29]   6492.43   6630.80   7349.62   8176.62   8573.17   9690.50  15151.84
[36]  34061.01   5921.10   5814.58  12421.25   6369.77   7609.12   7224.75
[43]   8121.22   7979.25   8093.06   8476.70  17914.66  30114.41   4826.64
[50]   6470.23   9638.77   8821.17   8722.37  10209.48  11276.55  12552.22
[57]  11637.39  13606.89  21822.11  45060.69   7615.03   9849.69  14558.40
[64]  11587.33   9332.56  13082.09  16732.78  19888.61  23933.38  25391.35
[71]  36024.80  80721.71  10243.24  11266.88  21826.84  17357.33  15997.79
[78]  18601.53  26155.15  28586.52  30505.41  30821.33  46634.38 104660.67
```

```
plot.ts(souvenirtimeseries)
logsouvenirtimeseries <- log(souvenirtimeseries)
plot.ts(logsouvenirtimeseries)
DLSTS<- decompose(logsouvenirtimeseries)
plot(DLSTS)
```




```
logsouvenirseriesseasonallyadjusted <- logsouvenirtimeseries -  DLSTS$seasonal
logsouvenirseriesseasonallyadjusted
```

	Jan	Feb	Mar	Apr	May	Jun	Jul
1987	8.034210	8.167629	7.848181	8.468944	8.502825	8.443471	8.447853
1988	8.440714	8.941510	8.781693	8.772632	8.955382	8.730821	8.798942
1989	9.075677	9.034118	9.102461	8.871369	9.050917	9.022888	8.972415
1990	9.303022	9.053559	9.323535	9.054324	9.209628	9.108675	9.072247
1991	9.098650	9.160402	9.069920	9.379915	9.346171	9.454479	9.400492
1992	9.554623	9.580630	9.482295	9.652673	9.413790	9.702407	9.795136
1993	9.851117	9.715058	9.887267	10.056775	9.952731	10.054406	10.241812

	Aug	Sep	Oct	Nov	Dec
1987	8.288033	8.545404	8.701714	8.398493	8.602622
1988	8.575090	8.635710	8.605645	8.903969	8.970521
1989	9.117772	9.080249	9.112900	9.088388	9.147307
1990	9.093338	9.022619	8.979075	9.255885	9.024158
1991	9.546391	9.385835	9.452330	9.453189	9.427164
1992	10.006641	10.106886	10.076162	9.954473	10.010161
1993	10.369429	10.349516	10.269960	10.212604	10.269877

3. PREDICŢIA SCR

În aceasta parte a cursului vom prezenta o parte dintre metodele de predicţie a seriilor de timp. Acestea vor fi clasificate în:

- metode cantitative, dintre care: extrapolarea, metoda mediilor mobile (MMM), netezirea exponeţială simplă şi dublă (NES, NED), metodele Holt şi Winter,
- metode calitative, care utilizează judecata experţilor.

Vom analiza numai metodele cantitative, celelalte nefăcând obiectul acestui referat.

3.1. Extrapolarea

Aceasta este un procedeu de previziune simplu, bazat pe ipoteza că fenomenul studiat se modifică constant, cu aceeaşi cantitate şi în viitor, cunoscând evoluţia acestuia în câteva puncte din trecut. Atunci, extrapolarea ("determinarea") termenului y_t va fi făcută prin relaţia:

$$y_t = y_1 + (t-1)\cdot \bar{\Delta}\ ,\ t \in \mathbf{N}\ ,\ t > n, \qquad (1)$$

y_1 şi y_n fiind cunoscuţi.

Dacă SCR este mai aproape de o evoluţie exponenţială, estimabilă cu o progresie geometrică, cu raţie aproximativ egală cu indicele mediu de modificare relativă, $\bar{I}$, atunci:

$$y_t = y_1 \cdot \bar{I}^{(t-1)},\ t > n,\ t \in \mathbf{N}^*. \qquad (2)$$

Dacă se cunosc mai mult de doi termeni, calculele estimative sunt făcute folosind alte metode.

3.2. Metoda mediilor mobile

Metoda constă în calculul unei medii a celor mai recente n valori ale SCR. Aceasta (media) va reprezenta estimaţia făcută pentru valoarea a (n+1)-a din SCR.

Termenul "MM" se bazează pe faptul că dacă o nouă observaţie devine disponibilă pentru SCR, ea va înlocui cea mai veche dată şi calculează o nouă medie.

Un aspect important în folosirea oricărei metode de previziune este acurateţea estimaţiei. Se doreşte, aşadar, ca eroarea de estimaţie să fie cât mai mică.

Dacă notăm prin: f_t = N- media mobilă a ultimilor N date, adică:

$$f_t = \frac{y_t + y_{t-1} + \ldots + y_{t-N-1}}{N} \qquad (3)$$

şi prin :

$$\varepsilon_t = y_t - f_t\ ,\ t \in \overline{1,n}\ ,$$

eroarea de predicţie, vom defini ca măsură a acurateţii prognozei, ***deviaţia medie absolută*** (MAD), ca fiind media valorilor absolute ale erorilor de predicţie, adică:

$$MAD = \frac{|\varepsilon_1| + |\varepsilon_2| + \ldots + |\varepsilon_n|}{n} \qquad (4)$$

O altă măsură a preciziei estimaţiei este ***media pătratelor erorilor*** (MSE):

$$MSE = \frac{\varepsilon_1^2 + \varepsilon_2^2 + \ldots + \varepsilon_n^2}{n} \qquad (5)$$

Diferenţa dintre MSE şi MAD este că MSE este infuenţată mai mult de valorile extreme ale erorilor. În ceea ce urmează,vom folosi MAD.

Revenind la alegerea numărului N, acesta va fi determinat ca fiind cel care satisface condiţia minimizării MAD.

Un exemplu de aplicaţie a MMM pentru predicţia vânzărilor se află în Anexă.

În MMM, fiecare observaţie primeşte în calculul MM aceaşi pondere. Dacă se conferă ponderi diferite fiecăreia dintre datele care intervin în calculul MM rezultă o variantă a metodei şi anume **MMM ponderate.**

Aceste metode dau rezultate bune pentru serii de timp care variază de la un nivel de bază, adică dacă:

$$y_t = b + \varepsilon_t, \qquad (6)$$

unde b este nivelul de bază pentru serie şi ε_t este fluctuaţia aleatoare, de la nivelul de bază, în perioada t.

3.3. Netezirea exponenţială simplă (NES)

Dacă o serie de timp fluctuează de la un nivel de bază, NES poate fi folosită pentru a obţine previziuni bune pentru valorile viitoare ale seriei.

Modelul de bază este următorul:

$$F_{t+1} = \alpha \cdot y_t + (1-\alpha) \cdot F_t, \qquad (7)$$

unde:

F_{t+1} este valoarea prevăzută pentru perioada t+1,

y_t este valoarea actuală a seriei de timp pentru perioada t,

F_t este valoarea estimată a seriei de timp pentru perioada t,

α este constanta de netezire, $\alpha \in (0,1)$.

Putem spune că valoarea prezisă pentru perioada t+1 este o combinaţie convexă de valorile prevăzute şi de acea actuală în perioada t.

Într-adevăr, presupunând că avem trei date y_1, y_2, y_3, atunci previziunea pentru patru perioade devine:

$$F_4 = \alpha \cdot y_3 + (1-\alpha) \cdot F_3$$

Dar:

$$F_3 = \alpha \cdot y_2 + (1-\alpha) \cdot F_2 \Rightarrow F_4 = \alpha \cdot y_3 + \alpha \cdot (1-\alpha) \cdot y_2 + (1-\alpha)^2 \cdot F_3$$

$$F_2 = \alpha \cdot F_1 + (1-\alpha) \cdot F_1 \Rightarrow F_4 = \alpha \cdot y_3 + \alpha \cdot (1-\alpha) \cdot y_2 + \alpha \cdot (1-\alpha)^2 \cdot y_1 + (1-\alpha)^3 \cdot F_1$$

Pentru a iniţializa procedeul de predicţie trebuie cunoscută valoarea F_1. De obicei, aceasta este valoarea imediat precedentă valorii observate, adică y_0, ori poate fi considerat chiar y_1 (adică valoarea actuală în perioada 1).

Dacă: $F_1 = y_1$ atunci:

$$F_4 = \alpha \cdot y_3 + \alpha \cdot (1-\alpha) \cdot y_2 + \alpha \cdot (1-\alpha)^2 \cdot y_1$$

În general:

$$F_t = \alpha \cdot y_t + \alpha \cdot (1-\alpha) \cdot y_{t-1} + \alpha \cdot (1-\alpha)^2 \cdot y_{t-2} + \ldots + \alpha \cdot (1-\alpha)^2 \cdot y_{t-k} + \ldots$$

Întrucât : $\alpha + \alpha(1-\alpha) + \alpha(1-\alpha)^2 + \ldots = 1$,

relaţia precedentă arată că dacă se merge înapoi un număr "infinit" de perioade, previziunea pentru perioada următoare este o medie ponderată a tuturor observaţiilor, ponderea descrescând exponenţial cu factorul (1-α).

Un avantaj al procedeului este simplitatea sa şi faptul că necesită puţine date (pentru calculul lui y_{t+1} este necesară numai cunoaşterea lui y_t şi a lui F_t).

Se observă că estimările pentru perioada viitoare "netezesc" fluctuaţiile neregulate din SCR.

Prelucrând relaţia de definitie rezultă:

$$F_{t+1} = F_t + \alpha \cdot (y_t - F_t). \qquad (8)$$

Notând:

$$\varepsilon_t = y_t - F_t, \qquad (9)$$

eroarea estimaţiei din perioada t, rezultă că valoarea prevăzută pentru F_{t+1} este egală cu cea prevăzută prin perioada precedentă plus o ajustare, care este de α ori cea mai recentă eroare de estimare.

Observaţii:

Dacă $\alpha = \dfrac{2}{N+1}$, NES (cu parametrul de netezire α) şi MMM (cu o perioadă N a MM) vor da estimaţii similare. De exemplu, $\alpha = \dfrac{1}{3}$ este sensibil echivalent cu o 5-perioadă a MM.

1. În practică, α este ales, de obicei, 0.1, 0.3 sau 0.5. Dacă valoarea lui α care minimizează MAD depăşeşte 0.5, atunci este probabilă prezenţa trendului, a sezonalitatăţii sau a

variaţiilor ciclice şi NES nu este recomandată ca tehnică de predicţie. În astfel de cazuri rezultate superioare vor fi date de metoda lui Holt sau Winter.

Dacă seria de timp manifestă o variaţie aleatoare substanţială, este preferabilă o valoare mai mică a constantei de netezire. Raţiunea acestei alegeri este aceea că dacă cea mai mare parte a erorii de estimare este datorată variabilităţii aleatoare, nu se doreşte ajustarea prea rapidă a predicţiei.

Pentru o serie cronologică relativ stabilă, cu variabilitate relativ redusă, valorile mai mari ale constantei de netezire au avantajul ajustării rapide a previziunilor atunci când se produc erorile de previziune şi în consecinţă se permite predicţiilor să reacţioneze mai rapid la schimbarea condiţiilor.

2. Criteriul utilizat pentru a determina valoarea dorită pentru constanta de netezire α este acelaşi ca pentru determinarea numărului de perioade al datelor incluse în calculul MM (adică se alege α care minimizează MAD).

3. Chiar dacă o serie de timp nu variază de la un nivel de bază, NED poate să dea totuşi predicţii bune.

Dacă : $y_t = m_t + \varepsilon_t$ şi $m_t = m_{t-1} + \delta_t$,

unde ε_t şi δ_t sunt termeni eroare independenţi, fiecare având medie 0, atunci NES va da valori bune.

3.4. Netezirea exponeţială dublă (NED)

NES nu funcţionează în prezenţa trendului. De exemplu, dacă există un trend crescător în cerere, NES va da predicţii prea slabe. NED, care este mai complicată, include o ajustare a creşterii viitoare sau descreşterii trendului. O altă metodă de ajustare, care nu este inclusă aici (în NED) ţine seama de sezonalitate.

NED se bazează pe următoarele formule de calcul:

$$D_t = 2 \cdot S_{t-1} - F_{t-2}, \qquad (10)$$

unde:

D_t reprezintă previziunea pentru perioada viitoare t, prin NED,
S_{t-1} reprezintă previziunea pentru perioada t-1, prin NES,
F_{t-2} reprezintă funcţia de NED pentru perioada t-2,
y_t este cererea actuală,
α reprezintă constanta de netezire.

$$S_t = S_{t-1} + \alpha \cdot (y_{t-1} - S_{t-1}) = \alpha \cdot y_{t-1} + (1 - \alpha) \cdot S_{t-1} \qquad (11)$$

$$F_t = \alpha \cdot S_t + (1- \alpha) \cdot F_{t-1} \qquad (12)$$

Netezirea adaptată ţine seama de ajustarea automatică a coeficientului de netezire, α, bazată pe o funcţie de previziune a erorii de măsurare. Cea mai folosită funcţie este ***semnalul urmă***, care este calculat ca raport între deviaţia cumulată (RSFE) şi MAD, unde RSFE este egală cu suma tuturor erorilor de predicţie.

O creştere a mărimii semnalului urmă reflectă o creştere continuă a erorii de predicţie în acceaşi direcţie, fie pozitivă sau fie negativă. Aceasta sugerează posibilitatea deplasării în model. Modelul nedeplasat va avea RSFE constantă, localizată în jurul lui zero.

Exemplu:

1. Fisierul http://robjhyndman.com/tsdldata/hurst/precip1.dat contine precipitatiile anuale din Londra, in perioada 1813-1912 (original data from Hipel and McLeod, 1994).

```
rain <- scan("http://robjhyndman.com/tsdldata/hurst/precip1.dat",skip=1)
rainseries <- ts(rain,start=c(1813))
```

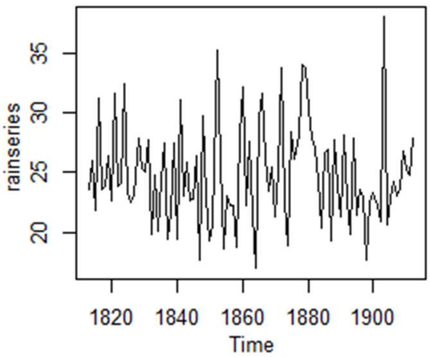
library(trend)
mk.test(rainseries)

Mann-Kendall trend test

data: rainseries
z = -0.28888, n = 100, p-value = 0.7727
alternative hypothesis: true S is not equal to 0
sample estimates:

S	varS	tau
-9.800000e+01	1.127480e+05	-1.980198e-02

plot.ts(rainseries)



Fluctuatiile seriei par sa fie constant in timp, deci e posibil sa descriem datele folosind un model aditiv si apoi sa facem o predictive folosind netezirea exponentiala simpla

- Functia “HoltWinters()” in R – cu parametrii beta=FALSE si gamma=FALSE

rainseriesforecasts <- HoltWinters(rainseries, beta=FALSE, gamma=FALSE)

rainseriesforecasts
Smoothing parameters:
alpha: 0.02412151
beta : FALSE
gamma: FALSE
Coefficients:
[,1]
a 24.67819

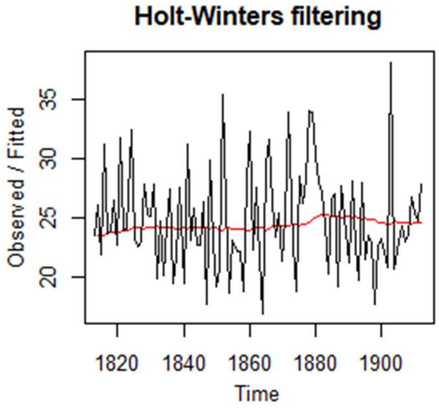
Predictiile sunt stocate in “fitted’’.

rainseriesforecasts$fitted
Time Series:
Start = 1814
End = 1912
Frequency = 1

	Xhat	level
1814	23.56000	23.56000
1815	23.62054	23.62054
1816	23.57808	23.57808

1817   23.76290   23.76290

plot(rainseriesforecasts)



Ca masura a acuratetii se calculeaza suma patratelor erorilor: MSE si se stocheaza in "SSE", care se apeleaza cu

rainseriesforecasts$SSE

[1] 1828.855

De obicei trebuie specificata prima valoare a seriei, care va fi folosita la netezire. Aceasta se face folosind parametrul "l.start"

HoltWinters(rainseries, beta=FALSE, gamma=FALSE, l.start=23.56)

Daca dorim sa facem predictie:

```
library("forecast")
y <- forecast:::forecast.HoltWinters(rainseriesforecasts, h=8)
# predictie pe 8 ani
y
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
1913	24.67819	19.17493	30.18145	16.26169	33.09470
1914	24.67819	19.17333	30.18305	16.25924	33.09715
1915	24.67819	19.17173	30.18465	16.25679	33.09960
1916	24.67819	19.17013	30.18625	16.25434	33.10204
1917	24.67819	19.16853	30.18785	16.25190	33.10449
1918	24.67819	19.16694	30.18945	16.24945	33.10694
1919	24.67819	19.16534	30.19105	16.24701	33.10938
1920	24.67819	19.16374	30.19265	16.24456	33.11182

plot(y)

Erorile sunt stocate in "residual"

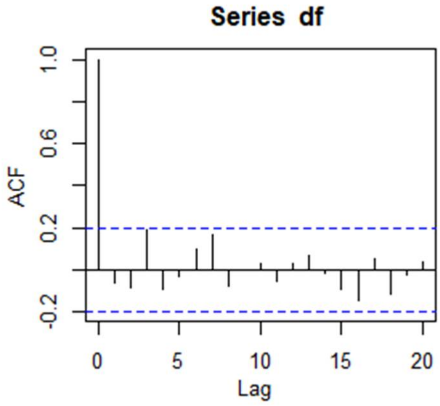
y$residuals
Time Series:
Start = 1813
End = 1912
Frequency = 1
[1]   NA 2.5100000 -1.7605450 7.6619220 -0.1128951 0.1198281

```
 [7]  2.6469377 -1.1569105  7.8909960 -0.1293468  0.1237733  8.4407877
[13] -0.9328169 -1.6003159 -1.1317139  3.7755848  1.1245120  0.8573870
[19]  3.5167056 -4.5081227  0.5606200 -4.1129030  0.2063065  3.2813300
[25] -4.7778206 -2.4725723  3.4470698 -4.6960787  7.1171978 -1.0944797
[31]  1.6919208 -1.5488909 -1.4115293  2.2325189 -6.4813328  5.7850067
[37] -1.2345364 -4.9147575 -3.3862062 11.4054742  1.6803570 -5.6001757
[43] -1.0550911 -1.8796407 -1.8643009 -5.2293312  4.3368082  8.2621979
[49] -1.9070988  3.4389033 -2.6240483 -7.2207523  5.5034232  7.4906723
[55]  1.9599860 -0.9372918  1.1053171 -3.0213449  0.7515345  9.5734064
[61] -1.8475186 -5.6529537  4.1034041  1.7244238  3.6928281  9.5137515
[67]  9.0242655  5.2665866  2.7795486  1.9325016 -0.8541132 -4.8835107
[73]  1.5242869  1.8575188 -5.9872873  2.6871351 -1.2676827 -3.8571042
[79]  3.1559349 -2.4601910 -5.2108475  3.0548460 -3.4888415 -1.3546853
[85] -1.9820083 -7.1041993 -2.0828352 -1.2925941 -2.3714148 -3.6442127
[91] 13.7036912 -4.0768625 -1.6585224 -0.3285164 -1.5705920 -0.8727070
[97]  2.2283440  0.7845930  0.1956674  3.2809476
```

```
df <- na.omit(y$residuals)
df
```

```
Time Series:
Start = 1814
End = 1912
Frequency = 1
 [1]  2.5100000 -1.7605450  7.6619220 -0.1128951  0.1198281  2.6469377
 [7] -1.1569105  7.8909960 -0.1293468  0.1237733  8.4407877 -0.9328169
[13] -1.6003159 -1.1317139  3.7755848  1.1245120  0.8573870  3.5167056
[19] -4.5081227  0.5606200 -4.1129030  0.2063065  3.2813300 -4.7778206
[25] -2.4725723  3.4470698 -4.6960787  7.1171978 -1.0944797  1.6919208
[31] -1.5488909 -1.4115293  2.2325189 -6.4813328  5.7850067 -1.2345364
[37] -4.9147575 -3.3862062 11.4054742  1.6803570 -5.6001757 -1.0550911
[43] -1.8796407 -1.8643009 -5.2293312  4.3368082  8.2621979 -1.9070988
[49]  3.4389033 -2.6240483 -7.2207523  5.5034232  7.4906723  1.9599860
[55] -0.9372918  1.1053171 -3.0213449  0.7515345  9.5734064 -1.8475186
[61] -5.6529537  4.1034041  1.7244238  3.6928281  9.5137515  9.0242655
[67]  5.2665866  2.7795486  1.9325016 -0.8541132 -4.8835107  1.5242869
[73]  1.8575188 -5.9872873  2.6871351 -1.2676827 -3.8571042  3.1559349
[79] -2.4601910 -5.2108475  3.0548460 -3.4888415 -1.3546853 -1.9820083
[85] -7.1041993 -2.0828352 -1.2925941 -2.3714148 -3.6442127 13.7036912
[91] -4.0768625 -1.6585224 -0.3285164 -1.5705920 -0.8727070  2.2283440
[97]  0.7845930  0.1956674  3.2809476
```

```
acf(df,lag.max=20)
```



Pentru a testa existenta unor corelatii nenule la intarzierile 1-20, se face un test Ljung-Box test folosind functia “Box.test()”. Intarzierea maxima pe care vrem sa o analizam este specificata de parametrul “lag” din functia Box.test().

Box.test(df, lag=20, type="Ljung-Box")

Box-Ljung test
data: df
X-squared = 17.401, df = 20, p-value = 0.6268

plot(df); hist(y)