

R for Data Science

Bài 14: Time Series Analysis

Phòng LT & Mạng

https://csc.edu.vn/lap-trinh-va-cSDL/R-Programming-Language-for-Data-Science_190

2020



Nội dung



1. Giới thiệu
2. Holt-Winters

Giới thiệu

- Một trong những dữ liệu phổ biến là Time Series Data, dữ liệu theo thời gian.
 - Ví dụ: Lượng mưa (mm) 60 tháng từ năm 2014 đến năm 2018 ở Hà Nội
- Những dữ liệu theo thời gian có đặc điểm là chúng biến chuyển theo chu kỳ và có xu hướng.
- R cung cấp các phương thức để tạo, thao tác và vẽ biểu đồ time series data. Dữ liệu time series được lưu trữ trong time-series object, có thể là vector hoặc data frame.

R programming language for Data Science

3

Giới thiệu



- Đối tượng time series được tạo ra bởi ts() function:

```
timeseries.object.name <- ts(data,  
start, end, frequency)
```

- Trong đó:

- data: vector hoặc matrix sử dụng trong time series
- start: thời gian bắt đầu cho lần quan sát đầu tiên
- end: thời gian kết thúc cho lần quan sát cuối cùng
- frequency: tần suất (số lần quan sát)

Giới thiệu

- frequency: tần suất (số lần quan sát)
 - frequency = 12: 12 dữ liệu ứng với 12 tháng trong năm
 - frequency = 4: 4 dữ liệu ứng với mỗi quý trong năm
 - frequency = 6: 6 dữ liệu ứng với 1 giờ (10 phút/dữ liệu * 6)
 - frequency = 24*6: dữ liệu ứng với 1 ngày (10 * 6 * 24)



Giới thiệu

- Ví dụ: Lượng mưa (mm) 60 tháng từ tháng 01/2014 đến tháng 12/2018 ở Hà Nội

```
rainfall = scan("rain_fall.txt")
```

```
rainfall
```

0.7	16.1	68.6	170.4	106.1	221.7	357.3	314.7	237.3	119.4	36.5	11.8	25.6	12.5	59.4	21.6	74.2	241.1	96.8	354.2	345.4	99.7	
158	31.5	96.9	4.2	24.7	104.5	249	95.1	280.4	534.5	178.5	45	9.3	9	70.9	12.3	112.4	19.1	105.4	212.9	449.1	283.2	266.9
259.7	19.4	47.5	16.6	10	34	58.8	209	188.5	428.1	313.4	229.7	94.4	28.2	84.2								

- T/h dữ liệu không có cột tháng/năm, với các giá trị sắp theo theo trình tự thời gian.
- Với dữ liệu này bắt đầu từ 01/2014 đến 12/2019. 1 năm có 12 lần đo đặc liên quan đến 12 tháng.



Giới thiệu

```
ts.rainfall = ts(rainfall, frequency = 12, start = c(2014,1))
```

```
ts.rainfall
```

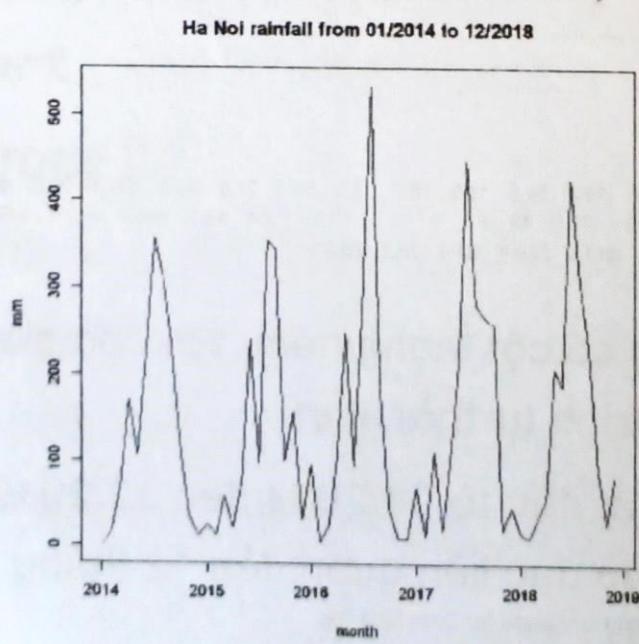
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2014	0.7	16.1	68.6	170.4	106.1	221.7	357.3	314.7	237.3	119.4	36.5	11.8
2015	25.6	12.5	59.4	21.6	74.2	241.1	96.8	354.2	345.4	99.7	158.0	31.5
2016	96.9	4.2	24.7	104.5	249.0	95.1	280.4	534.5	178.5	45.0	9.3	9.0
2017	70.9	12.3	112.4	19.1	105.4	212.9	449.1	283.2	266.9	259.7	19.4	47.5
2018	16.6	10.0	34.0	58.8	209.0	188.5	428.1	313.4	229.7	94.4	28.2	84.2



Giới thiệu

- Biểu đồ: sử dụng `plot.ts()` hoặc `plot()` để vẽ biểu đồ sau khi có dữ liệu theo time series

```
plot.ts(ts.rainfall, main="Ha Noi rainfall from 01/2014 to 12/2018",
       ylab="mm", xlab="month", col="blue")
```



Có thể thấy biểu đồ này mang tính theo mùa: lượng mưa thường cao nhất vào mùa hè và giảm thấp vào mùa đông.



Giới thiệu

□ Phân tích các thành phần trong dữ liệu theo thời gian

- Trend (xu hướng lâu dài): là chiều hướng biến đổi, tăng hay giảm trong dữ liệu và không nhất thiết phải tuyến tính.
- Seasonal (theo mùa): xảy ra khi số liệu chịu ảnh hưởng bởi các yếu tố mùa (quý, tháng, ngày). Biến đổi theo mùa xảy ra trong một quãng thời gian nhất định.
- Cyclic (chu kỳ): biến đổi theo chu kỳ tồn tại khi dữ liệu thể hiện những biến chuyển (tăng/ giảm) mà không theo một thời gian cố định

R programming language for Data Science

9

Giới thiệu

□ Phân biệt giữa seasonal và cyclic

- Nếu dao động xảy ra theo thời gian cố định (tuần/ tháng/quý) thì đó là seasonal.
- Nếu dao động không xảy ra theo thời gian cố định thì đó là cyclic
- Thời gian biến chuyển của cyclic thường dài hơn của seasonal
- Mức độ biến chuyển của cyclic cũng thường cao hơn seasonal.

Nội dung

1. Giới thiệu

2. Holt-Winters



Holt-Winters

❑ Exponential Smoothing

• Simple Exponential Smoothing (SES)

- Nếu dữ liệu có thể mô tả bằng một mô hình cộng hưởng và không có yếu tố mùa thì dùng phương pháp SES. SES là cách tiên lượng trong khoảng thời gian ngắn.



Holt-Winters

□ Sử dụng HoltWinters() để tính toán dự đoán SES:

```
forecast =  
HoltWinters(timeseries.object.name, beta = F,  
gamma = F)
```

- Hàm này cho kết quả chứa trong biến predict có tên là fitted.
 - Xem kết quả bằng forecast\$fitted

Holt-Winters

- Alpha: được gọi là smoothing parameter (có giá trị dao động từ 0 đến 1; nếu gần 0 thì trọng số cho số liệu trước đó không cao trong dự đoán giá trị kế tiếp; nếu gần 1 thì cao)

```
forecast.rainfall
```

```
Holt-Winters exponential smoothing without trend and without seasonal component.
```

```
Call:  
HoltWinters(x = ts.rainfall, beta = F, gamma = F)
```

```
Smoothing parameters:
```

```
alpha: 0.8286106
```

```
beta : FALSE
```

```
gamma: FALSE
```

```
Coefficients:
```

```
[,1]
```

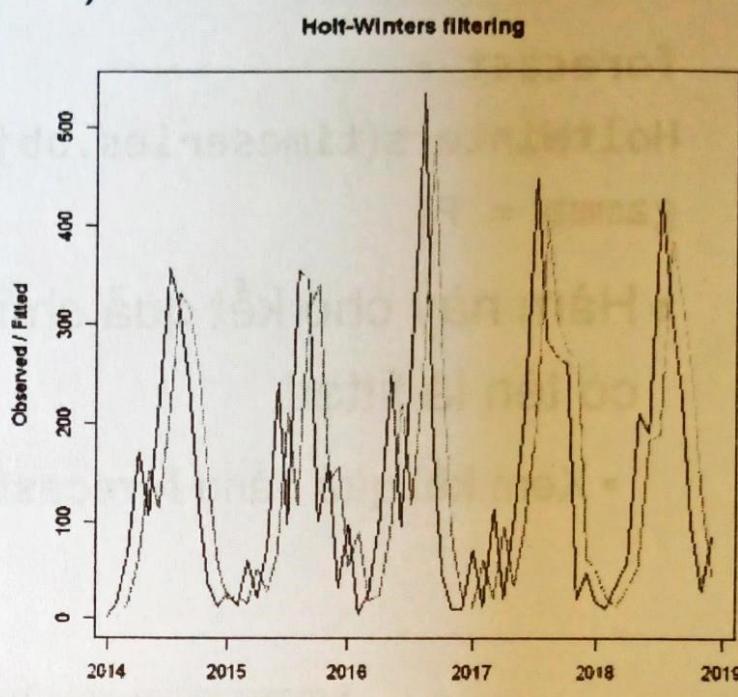
```
a 77.31102
```

Holt-Winters

- Biểu đồ biểu diễn giá trị thực tế và giá trị dự đoán (đường màu đỏ).

Có thể tính tổng bình phương sai số (sum of squared errors) bằng SSE:

```
forecast.rainfall$SSE  
881132.694466033
```



R programming

15

Holt-Winters

- Tuy nhiên, HoltWinters() trên chỉ dự đoán khoảng thời gian của dữ liệu gốc. Để dự đoán trong tương lai chúng ta cài install.packages("forecast") và sử dụng:

```
library("forecast")  
forecast:::forecast.HoltWinters(timeseries.object.name.forecasts, h="step")
```



Holt-Winters

- Ví dụ: Dự đoán lượng mưa Hà Nội 5 tháng tiếp theo:

`forecast.rainfall.new`

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2019	77.31102	-80.6344	235.2564	-164.2457	318.8677
Feb 2019	77.31102	-127.8111	282.4331	-236.3962	391.0182
Mar 2019	77.31102	-166.0065	320.6285	-294.8110	449.4331
Apr 2019	77.31102	-198.9710	353.5930	-345.2259	499.8479
May 2019	77.31102	-228.4014	383.0234	-390.2358	544.8578

HoltWinters cung cấp giá trị dự đoán và khoảng tin cậy 80% đến 95%. Ví dụ: Lượng mưa tháng 01/2019 là 77.31, khoảng tin cậy 95% từ -164.25 đến 318.86, khoảng tin cậy 80% từ -80.63 đến 235.25

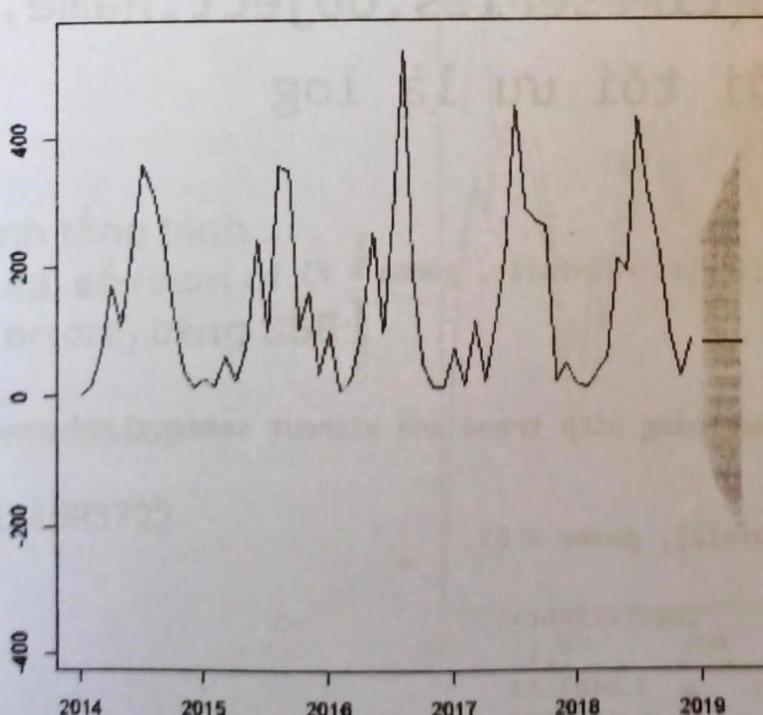
R programming language for Data Science

17

Holt-Winters

`plot.forecast(forecast.rainfall.new)`

Forecasts from HoltWinters



Biểu đồ vẽ phần dự đoán màu xanh, còn khoảng tin cậy 80% đến 95% là vùng màu xám

Với kết quả trên, cho thấy, SES không phải là phương pháp thích hợp cho dữ liệu này.

R programming language for Data Science

18

□ Holt Exponential Smoothing (HES)

- Nếu dữ liệu không có yếu tố mùa nhưng có xu hướng tăng hoặc giảm thì dùng phương pháp HES. Đây là cách dự đoán trong khoảng thời gian ngắn
- HES sử dụng 2 tham số là alpha, và beta với alpha là trọng số của giá trị hiện tại, beta thể hiện độ dốc cho phần trend ngay tại thời điểm hiện tại. 2 tham số đều có giá trị trong khoảng [0, 1]



- Sử dụng **forecast = HoltWinters(timeseries.object.name, gamma=F)** với tối ưu là log

- Ví dụ:

```
HES.forecast = HoltWinters(log(ts.rainfall), gamma = F)
```

```
HES.forecast
```

```
Holt-Winters exponential smoothing with trend and without seasonal component.
```

```
Call:
```

```
HoltWinters(x = log(ts.rainfall), gamma = F)
```

```
Smoothing parameters:  
alpha: 0.6848052  
beta : 0.6830466  
gamma: FALSE
```

```
Coefficients:  
[,1]  
a 3.8463113  
b -0.1694587
```



Holt-Winters

□ Giải thích

- Alpha (~0.685) khá cao: cho thấy giá trị hiện tại có trọng số khá cao trong việc dự đoán giá trị tương lai
- Beta (~0.68) khá cao: cho thấy độ dốc của trend được cập nhật trong thời gian qua

Call:
`HoltWinters(x = log(ts.rainfall), gamma = F)`

Smoothing parameters:
`alpha: 0.6848052`
`beta : 0.6830466`
`gamma: FALSE`

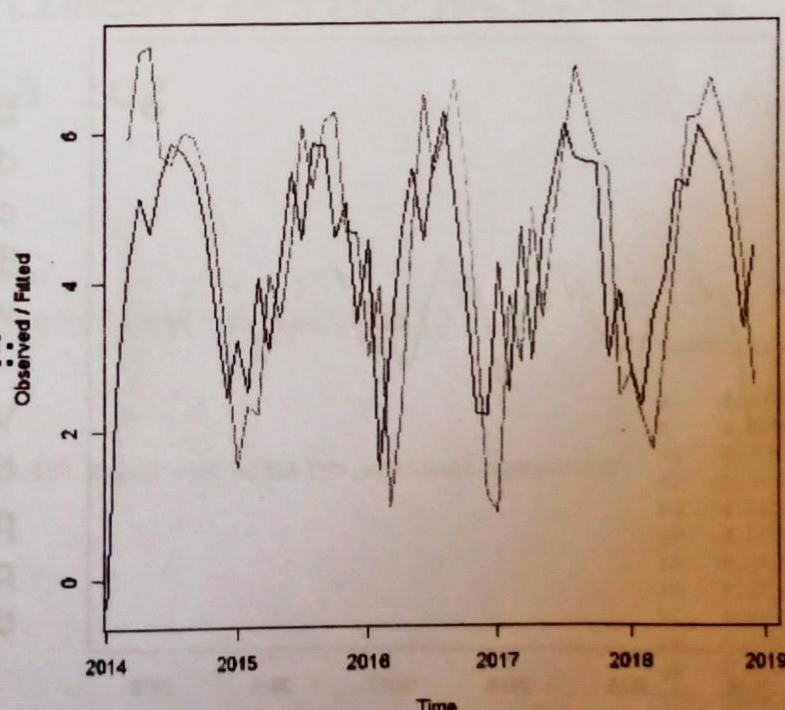
21

R programming language for Data Science

Holt-Winters

`plot(HES.forecast)`

Holt-Winters filtering



Có thể tính tổng bình phương sai số (sum of squared errors) bằng SSE:

`HES.forecast$SSE`

108.584324593722

22

R programming language for Data Science

Holt-Winters

```
HES.forecast.new = forecast.HoltWinters(HES.forecast, h=5)
```

```
HES.forecast.new
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2019	3.676853	1.9150629	5.438642	0.9824277	6.371277
Feb 2019	3.507394	0.8190666	6.195721	-0.6040479	7.618836
Mar 2019	3.337935	-0.5833061	7.259176	-2.6590857	9.334956
Apr 2019	3.168476	-2.2082529	8.545206	-5.0545215	11.391474
May 2019	2.999018	-4.0141548	10.012190	-7.7267041	13.724739

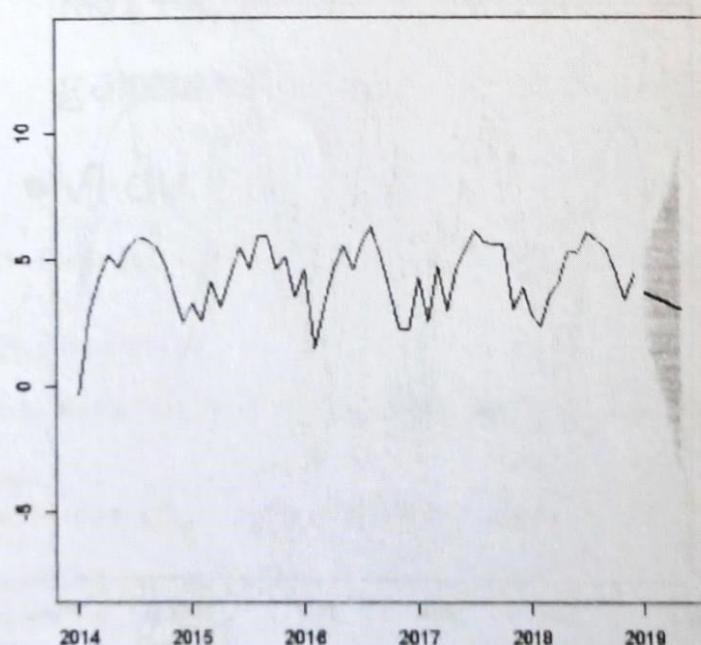
HoltWinters cung cấp giá trị dự đoán và khoảng tin cậy 80% đến 95%. Ví dụ: Lượng mưa tháng 01/2019 là 3.6768 (log), khoảng tin cậy 95% từ 0.982 đến 6.37



Holt-Winters

```
plot(HES.forecast.new)
```

Forecasts from HoltWinters



Biểu đồ vẽ phần
dự đoán màu xanh,
còn khoảng tin cậy
80% đến 95% là
vùng màu xám

Với kết quả trên, cho
thấy, HES không
phải là phương
pháp thích hợp cho
dữ liệu này.



□ Holt-Winters Exponential Smoothing

- Nếu dữ liệu có thể mô tả bằng một mô hình cộng hưởng và có yếu tố mùa thì dùng phương pháp Holt-Winters ES. Đây là cách dự đoán trong khoảng thời gian ngắn.
- Holt-Winters Exponential Smoothing sử dụng 3 tham số là alpha, beta, và gamma với alpha là trọng số của giá trị hiện tại, beta thể hiện độ dốc của giá trị hiện tại, gamma thể hiện seasonality. 3 tham số đều có giá trị trong khoảng [0, 1]



Holt-Winters

- Sử dụng **forecast = HoltWinters(timeseries.object.name)** với tối ưu là log

- Ví dụ:

```
# use log to optimization
HWES.forecast = HoltWinters(log(ts.rainfall))

HWES.forecast
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = log(ts.rainfall))

Smoothing parameters:
alpha: 0.008543883
beta : 1
gamma: 0.09435026
```

Coefficients:	[,1]
a	4.47766468
b	0.02414736
s1	-0.77223112
s2	-1.71165645
s3	-0.12523136
s4	-0.87068073
s5	0.34080168
s6	1.12211658
s7	1.54813501
s8	1.43928012
s9	1.13742642
s10	0.45545767
s11	-0.70032052
s12	-1.39168999



Holt-Winters

● Giải thích

- Alpha (~0.009) rất thấp: cho thấy giá trị hiện tại phụ thuộc vào các giá trị gần đây và một số giá trị trong quá khứ xa
- Beta (=1) rất cao: cho thấy độ dốc của trend được cập nhật trong thời gian qua
- Gamma (=0.094) thấp: cho thấy ảnh hưởng của yếu tố mùa tại thời điểm hiện tại không phụ thuộc vào giá trị gần nhất

Call:
HoltWinters(x = log(ts.rainfall))

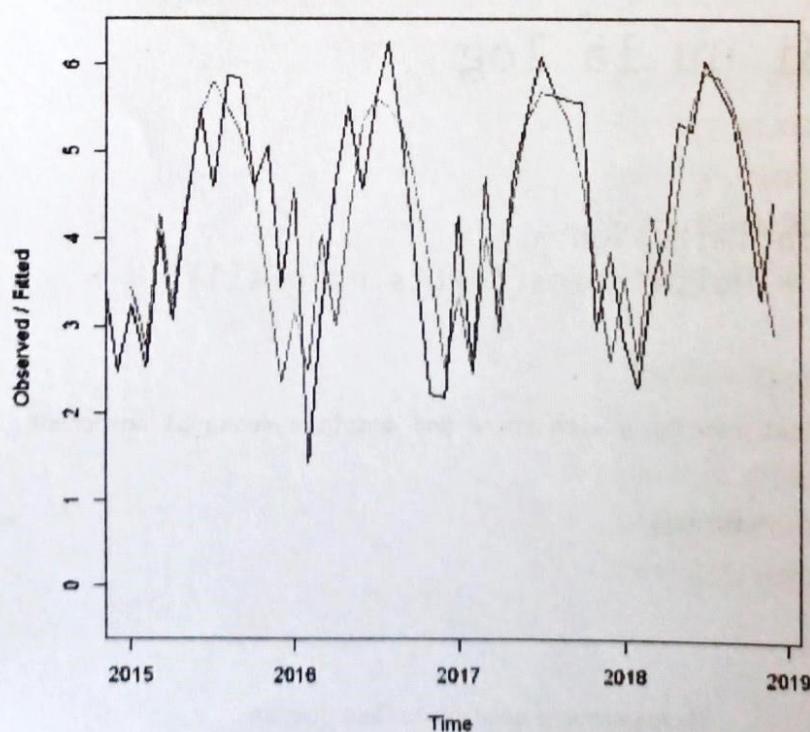
Smoothing parameters:
alpha: 0.008543883
beta : 1
gamma: 0.09435026



Holt-Winters

plot(HWES.forecast)

Holt-Winters filtering



Holt-Winters

- Ví dụ: Dự đoán cho 5 tháng tiếp theo

```
HWES.forecast.new = forecast.HoltWinters(HWES.forecast, h=5)
```

HWES.forecast.new

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2019	3.729581	2.747478	4.711684	2.227584	5.231578
Feb 2019	2.814303	1.832057	3.796549	1.312087	4.316519
Mar 2019	4.424875	3.442306	5.407444	2.922166	5.927585
Apr 2019	3.703573	2.720431	4.686715	2.199987	5.207159
May 2019	4.939203	3.955166	5.923240	3.434249	6.444158

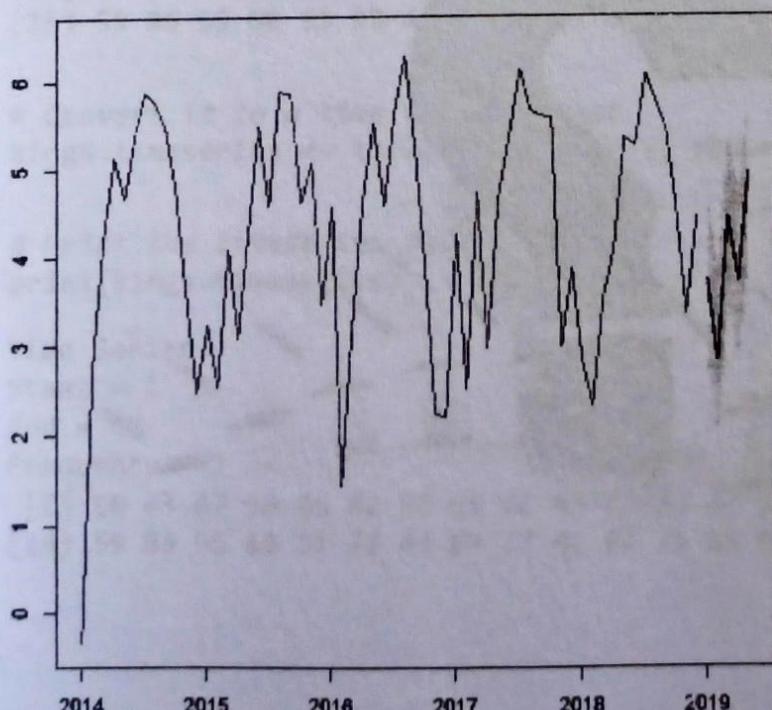
HoltWinters cung cấp giá trị dự đoán và khoảng tin cậy 80% đến 95%. Ví dụ: Lượng mưa tháng 01/2019 là 3.729 (log), khoảng tin cậy 95% từ 2.228 đến 5.232



Holt-Winters

```
plot(HWES.forecast.new)
```

Forecasts from HoltWinters



Biểu đồ vẽ
phần dự đoán
màu xanh, còn
khoảng tin cậy
80% đến 95% là
vùng màu xám



Holt-Winters

```
x = data.frame(HwES.forecast.new)
```

x

	Point.Forecast	Lo.80	Hi.80	Lo.95	Hi.95
Jan 2019	3.729581	2.747478	4.711684	2.227584	5.231578
Feb 2019	2.814303	1.832057	3.796549	1.312087	4.316519
Mar 2019	4.424875	3.442306	5.407444	2.922166	5.927585
Apr 2019	3.703573	2.720431	4.686715	2.199987	5.207159
May 2019	4.939203	3.955166	5.923240	3.434249	6.444158

```
x$Forecast = exp(x$Point.Forecast)
```

x

	Point.Forecast	Lo.80	Hi.80	Lo.95	Hi.95	Forecast
Jan 2019	3.729581	2.747478	4.711684	2.227584	5.231578	41.66164
Feb 2019	2.814303	1.832057	3.796549	1.312087	4.316519	16.68154
Mar 2019	4.424875	3.442306	5.407444	2.922166	5.927585	83.50240
Apr 2019	3.703573	2.720431	4.686715	2.199987	5.207159	40.59210
May 2019	4.939203	3.955166	5.923240	3.434249	6.444158	139.65892



Chapter 14: Time Series Analysis

Exercise 1: Kings

- Cung cấp tập tin kings.dat.txt
- Đọc dữ liệu từ tập tin và in dữ liệu
- Chuyển dữ liệu này thành Time Series object => in Time Series object
- Vẽ Time Series object vừa tạo

Exercise 2: Births

- Cung cấp tập tin mybirths.dat.txt
- Đọc dữ liệu từ tập tin, in dữ liệu
- Chuyển dữ liệu này thành Time Series object => in Time Series object
- Vẽ Time Series object vừa tạo
- Thực hiện việc dự báo và vẽ biểu đồ so sánh với thực tiễn
- Dự đoán số lượng sinh cho 6 tháng tiếp theo

Gợi ý:

Exercise 1: Kings

In [1]: # Get the data from file

```
kings <- scan("kings.dat.txt",skip=3)
print(kings)
```

```
[1] 60 43 67 50 56 42 50 65 68 43 65 34 47 34 49 41 13 35 53 56 16 43 69 59 48
[26] 59 86 55 68 51 33 49 67 77 81 67 71 81 68 70 77 56
```

In [2]: # Convert it to a time series object.

```
kings.timeseries <- ts(kings,start = 1, end = length(kings), frequency = 1)
```

In [3]: # Print the timeseries data.

```
print(kings.timeseries)
```

Time Series:

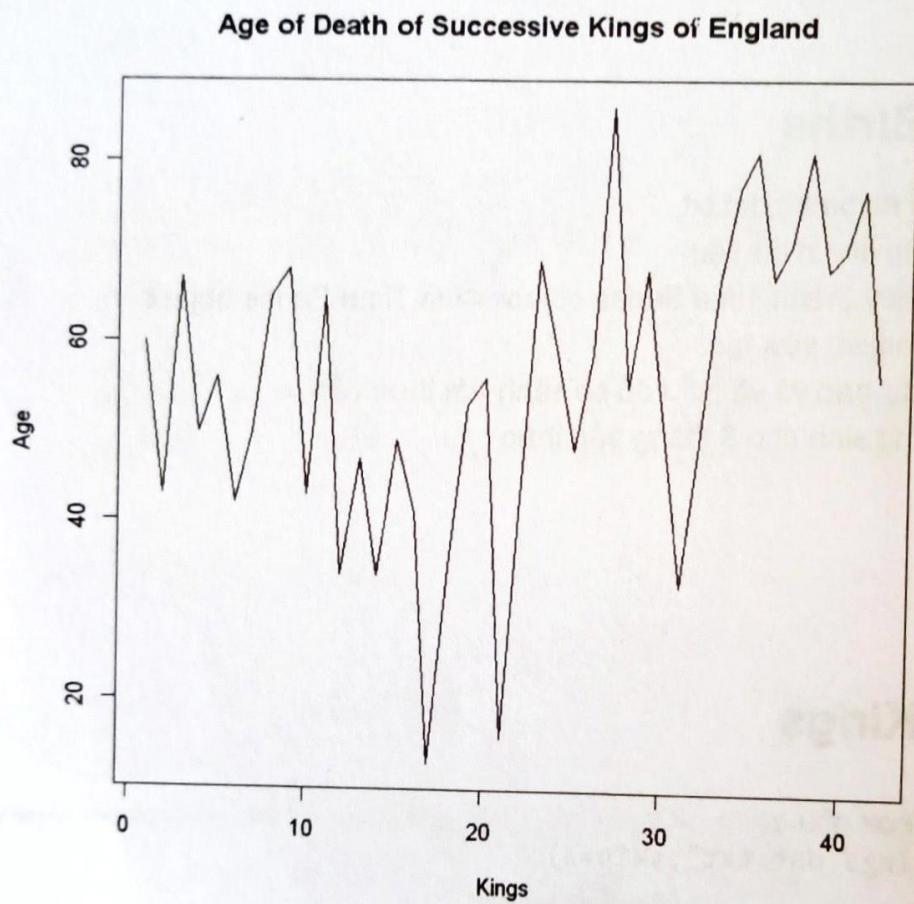
Start = 1

End = 42

Frequency = 1

```
[1] 60 43 67 50 56 42 50 65 68 43 65 34 47 34 49 41 13 35 53 56 16 43 69 59 48
[26] 59 86 55 68 51 33 49 67 77 81 67 71 81 68 70 77 56
```

```
In [4]: # Give the chart file a name.  
#png(file = "kings.png")  
  
# Plot a graph of the time series.  
plot(kings.timeseries, main='Age of Death of Successive Kings of England',  
      col='blue',  
      xlab='Kings', ylab='Age')  
  
# Save the file.  
#dev.off()
```



- Mô hình trên không có chu kỳ rõ ràng
- Độ dao động ngẫu nhiên có vẻ độc lập với thời gian

Exercise 2: Births



```
In [5]: # data set of the number of births per month in New York city, from January  
# Get the data from file  
births <- scan("mybirths.dat.txt",skip=3)  
print(births)
```

```
[1] 24.740 25.806 24.364 24.477 23.901 23.175 23.227 21.672 21.870 21.439  
[11] 21.089 23.709 21.669 21.752 20.761 23.479 23.824 23.105 23.110 21.759  
[21] 22.073 21.937 20.035 23.590 21.672 22.222 22.123 23.950 23.504 22.238  
[31] 23.142 21.059 21.573 21.548 20.000 22.424 20.615 21.761 22.874 24.104  
[41] 23.748 23.262 22.907 21.519 22.025 22.604 20.894 24.677 23.673 25.320  
[51] 23.583 24.671 24.454 24.122 24.252 22.084 22.991 23.287 23.049 25.076  
[61] 24.037 24.430 24.667 26.451 25.618 25.014 25.110 22.964 23.981 23.798  
[71] 22.270 24.775 22.646 23.988 24.737 26.276 25.816 25.210 25.199 23.162  
[81] 24.707 24.364 22.644 25.565 24.062 25.431 24.635 27.009 26.606 26.268  
[91] 26.462 25.246 25.180 24.657 23.304 26.982 26.199 27.210 26.122 26.706  
[101] 26.878 26.152 26.379 24.712 25.688 24.990 24.239 26.721 23.475 24.767  
[111] 26.219 28.361 28.599 27.914 27.784 25.693 26.881 26.217 24.218 27.914  
[121] 26.975 28.527 27.139 28.982 28.169 28.056 29.136 26.291 26.987 26.589  
[131] 24.848 27.543 26.896 28.878 27.390 28.065 28.141 29.048 28.484 26.634  
[141] 27.735 27.132 24.924 28.963 26.589 27.931 28.009 29.229 28.759 28.405  
[151] 27.945 25.912 26.619 26.076 25.286 27.660 25.951 26.398 25.565 28.865  
[161] 30.000 29.261 29.012 26.992 27.897
```

T

```
In [6]: # Convert it to a time series object.  
print("Bitrhs Time Series:")  
births.timeseries <- ts(births, start = c(1946,1), frequency = 12)  
  
# Print the timeseries data.  
births.timeseries  
  
[1] "Bitrhs Time Series:"
```

In [7]: # Give the chart file a name.

```
#png(file = "births.png")
```

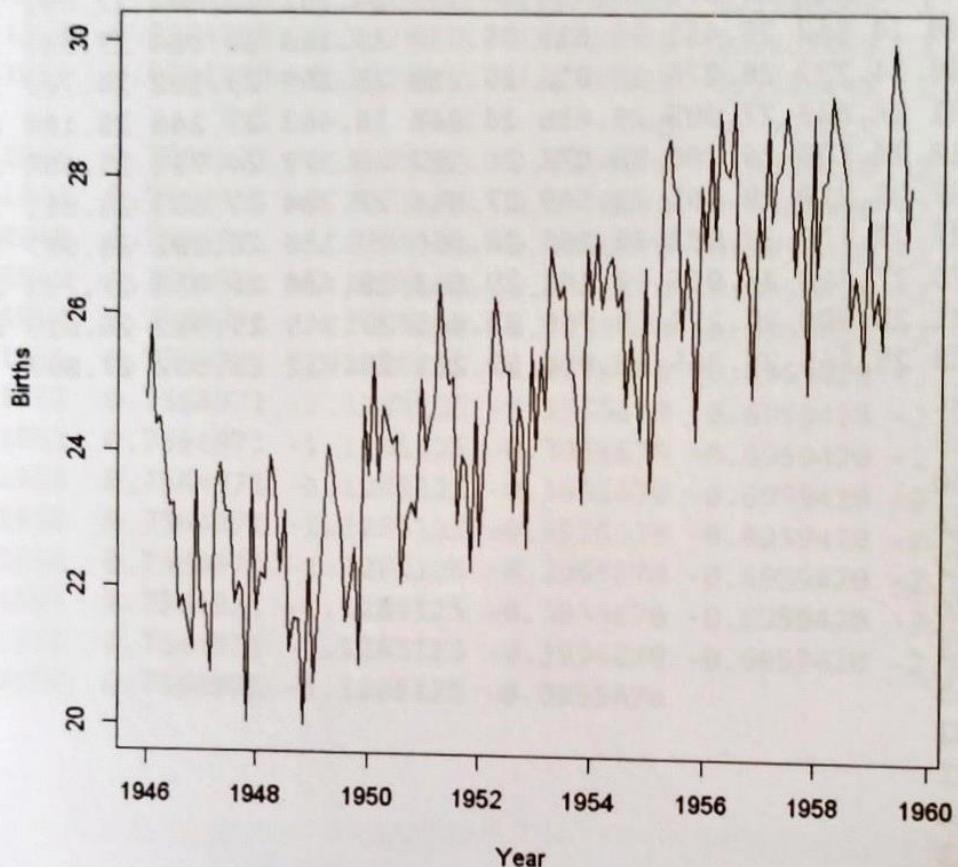
Plot a graph of the time series.

```
plot(births.timeseries,
      main='Births per month in New York city, from January 1946 to December 1959',
      col='red',
      xlab='Year', ylab='Births')
```

Save the file.

```
#dev.off()
```

Births per month in New York city, from January 1946 to December 1959





- Từ năm 1950, số trẻ sinh ra có xu hướng tăng theo năm

```
In [8]: # Xem xet cac thanh phan trong du lieu  
comp = decompose(births.timeseries)
```

In [9]: comp\$x



In [10]: comp\$seasonal

	Jan	Feb	Mar	Apr	May	Jun
1946	-0.8204260	0.2329041	-0.1720029	1.5139845	1.2193943	0.7476026
1947	-0.8204260	0.2329041	-0.1720029	1.5139845	1.2193943	0.7476026
1948	-0.8204260	0.2329041	-0.1720029	1.5139845	1.2193943	0.7476026
1949	-0.8204260	0.2329041	-0.1720029	1.5139845	1.2193943	0.7476026
1950	-0.8204260	0.2329041	-0.1720029	1.5139845	1.2193943	0.7476026
1951	-0.8204260	0.2329041	-0.1720029	1.5139845	1.2193943	0.7476026
1952	-0.8204260	0.2329041	-0.1720029	1.5139845	1.2193943	0.7476026
1953	-0.8204260	0.2329041	-0.1720029	1.5139845	1.2193943	0.7476026
1954	-0.8204260	0.2329041	-0.1720029	1.5139845	1.2193943	0.7476026
1955	-0.8204260	0.2329041	-0.1720029	1.5139845	1.2193943	0.7476026
1956	-0.8204260	0.2329041	-0.1720029	1.5139845	1.2193943	0.7476026
1957	-0.8204260	0.2329041	-0.1720029	1.5139845	1.2193943	0.7476026
1958	-0.8204260	0.2329041	-0.1720029	1.5139845	1.2193943	0.7476026
1959	-0.8204260	0.2329041	-0.1720029	1.5139845	1.2193943	0.7476026
	Jul	Aug	Sep	Oct	Nov	Dec
1946	0.7564971	-1.1285125	-0.3955670	-0.6959420	-2.1017080	0.8437759
1947	0.7564971	-1.1285125	-0.3955670	-0.6959420	-2.1017080	0.8437759
1948	0.7564971	-1.1285125	-0.3955670	-0.6959420	-2.1017080	0.8437759
1949	0.7564971	-1.1285125	-0.3955670	-0.6959420	-2.1017080	0.8437759
1950	0.7564971	-1.1285125	-0.3955670	-0.6959420	-2.1017080	0.8437759
1951	0.7564971	-1.1285125	-0.3955670	-0.6959420	-2.1017080	0.8437759
1952	0.7564971	-1.1285125	-0.3955670	-0.6959420	-2.1017080	0.8437759
1953	0.7564971	-1.1285125	-0.3955670	-0.6959420	-2.1017080	0.8437759
1954	0.7564971	-1.1285125	-0.3955670	-0.6959420	-2.1017080	0.8437759
1955	0.7564971	-1.1285125	-0.3955670	-0.6959420	-2.1017080	0.8437759
1956	0.7564971	-1.1285125	-0.3955670	-0.6959420	-2.1017080	0.8437759
1957	0.7564971	-1.1285125	-0.3955670	-0.6959420	-2.1017080	0.8437759
1958	0.7564971	-1.1285125	-0.3955670	-0.6959420	-2.1017080	0.8437759
1959	0.7564971	-1.1285125	-0.3955670			

In [11]: comp\$trend

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
1946	NA	NA	NA	NA	NA	NA	23.16112	22.86425
1947	22.29479	22.29354	22.30562	22.33483	22.31167	22.26279	22.25796	22.27767
1948	22.35242	22.32458	22.27458	22.23754	22.21988	22.16983	22.07721	22.01396
1949	22.16604	22.17542	22.21342	22.27625	22.35750	22.48862	22.70992	22.98563
1950	23.42679	23.50638	23.57017	23.63888	23.75713	23.86354	23.89533	23.87342
1951	24.28208	24.35450	24.43242	24.49496	24.48379	24.43879	24.36829	24.29192
1952	24.30129	24.31325	24.35175	24.40558	24.44475	24.49325	24.58517	24.70429
1953	25.02362	25.16308	25.26963	25.30154	25.34125	25.42779	25.57588	25.73904
1954	25.92137	25.89567	25.89458	25.92963	25.98246	26.01054	25.88617	25.67087
1955	26.06388	26.16329	26.25388	26.35471	26.40496	26.45379	26.64933	26.95183
1956	27.26925	27.35050	27.37983	27.39975	27.44150	27.45229	27.43354	27.44488
1957	27.45717	27.44429	27.48975	27.54354	27.56933	27.63167	27.67804	27.62579
1958	27.71037	27.65783	27.58125	27.49075	27.46183	27.42262	27.34175	27.25129
1959	27.17263	27.26208	27.36033	NA	NA	NA	NA	NA
	Sep	Oct	Nov	Dec				
1946	22.54521	22.35350	22.30871	22.30258				
1947	22.35400	22.43038	22.43667	22.38721				
1948	22.02604	22.06375	22.08033	22.13317				
1949	23.16346	23.21663	23.26967	23.33492				
1950	23.88150	24.00083	24.12350	24.20917				
1951	24.27642	24.27204	24.27300	24.28942				
1952	24.76017	24.78646	24.84992	24.92692				
1953	25.87513	25.92446	25.92317	25.92967				
1954	25.57312	25.64612	25.78679	25.93192				
1955	27.14683	27.21104	27.21900	27.20700				
1956	27.46996	27.44221	27.40283	27.44300				
1957	27.61212	27.68642	27.76067	27.75963				
1958	27.08558	26.96858	27.00512	27.09250				
1959	NA							

- Trong phần seasonal, yếu tố seasonal cao nhất là tháng 4 và thấp nhất là tháng 11. Như vậy tháng có nhiều trẻ sinh nhất là tháng 4, và thấp nhất là tháng 7

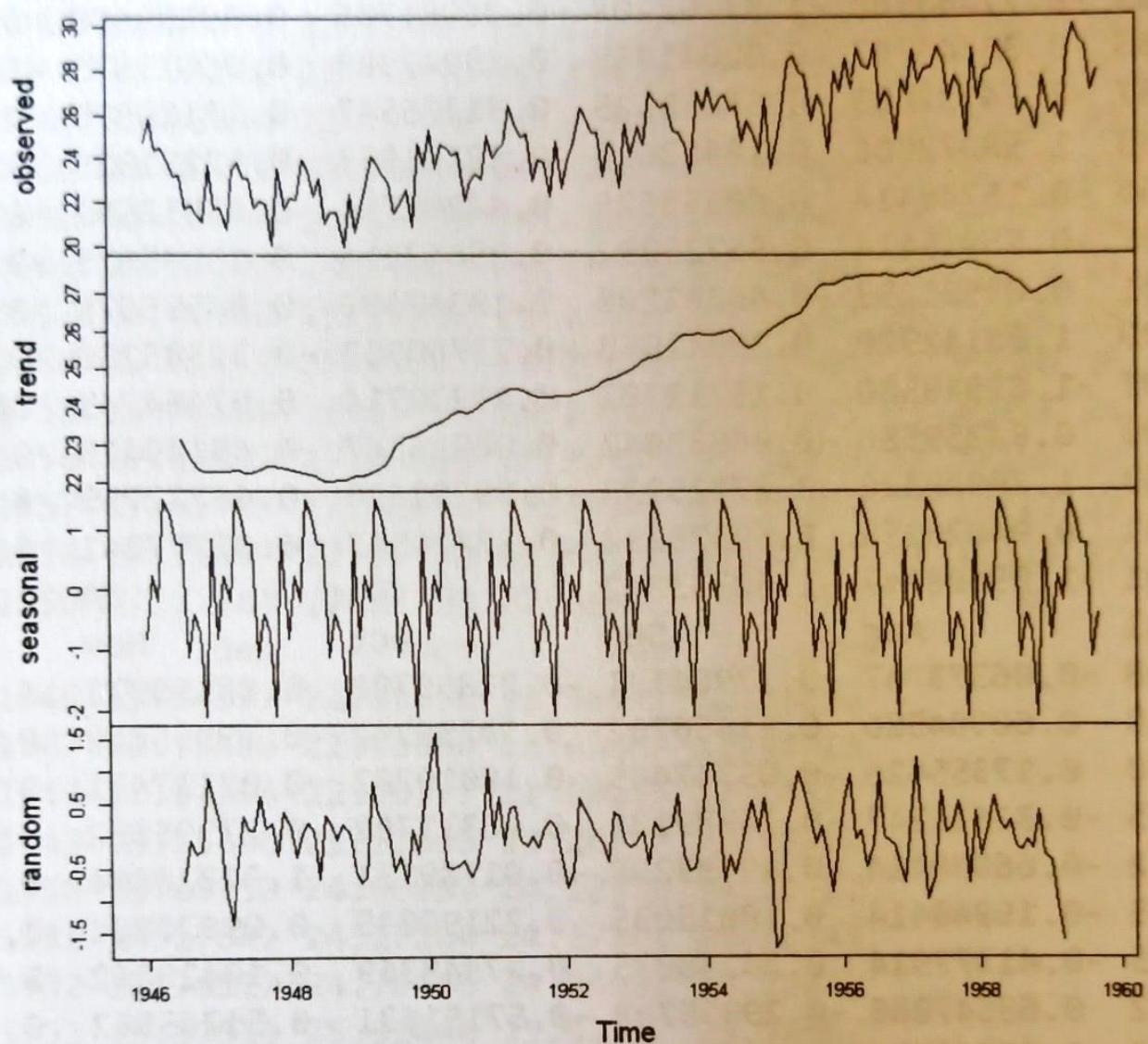
In [12]: comp\$random



	Jan	Feb	Mar	Apr	May	Jun
	NA	NA	NA	NA	NA	NA
1946	NA	NA	NA	NA	NA	NA
1947	0.19463433	-0.77444580	-1.37262208	-0.36981786	0.29293908	0.09460575
1948	0.14000933	-0.33548747	0.02041958	0.19847380	0.06473075	-0.67943592
1949	-0.73061567	-0.64732080	0.83258625	0.31376547	0.17110575	0.02577241
1950	1.06663433	1.58072086	0.18483625	-0.48185953	-0.52251925	-0.48914425
1951	0.57534266	-0.15740414	0.40658625	0.44205714	-0.08518592	-0.17239425
1952	-0.83486567	-0.55815414	0.55725292	0.35643214	0.15185575	-0.03085259
1953	-0.14119901	0.03501253	-0.46262208	0.19347380	0.04535575	0.09260575
1954	1.09805099	1.08142920	0.39941958	-0.73760953	-0.32385259	-0.60614425
1955	-1.76844901	-1.62919580	0.13712792	0.49230714	0.97464741	0.71260575
1956	0.52617599	0.94359586	-0.06883042	0.06826547	-0.49189425	-0.14389425
1957	0.25925933	1.20080420	0.07225292	-0.99252620	-0.64772759	0.66873075
1958	-0.30094901	0.04026253	0.59975292	0.22426547	0.07777241	0.23477241
1959	-0.40119901	-1.09698747	-1.62333042	NA	NA	NA
	Jul	Aug	Sep	Oct	Nov	Dec
1946	-0.69062208	-0.06373747	-0.27964131	-0.21855798	0.88199971	0.56264074
1947	0.09554458	0.60984586	0.11456702	0.20256702	-0.29995862	0.35901574
1948	0.30829458	0.17355420	-0.05747465	0.18019202	0.02137471	-0.55294260
1949	-0.55941375	-0.33811247	-0.74289131	0.08331702	-0.27395862	0.49830740
1950	-0.39983042	-0.66090414	-0.49493298	-0.01789131	1.02720804	0.02305740
1951	-0.01478875	-0.19940414	0.10015035	0.22190035	0.09870804	-0.35819260
1952	-0.14266375	-0.41377914	0.34240035	0.27348369	-0.10420862	-0.20569260
1953	0.12962792	0.63547086	-0.29955798	-0.57151631	-0.51745862	0.20855740
1954	-0.26366375	0.16963753	0.51044202	0.03981702	0.55391638	-0.05469260
1955	0.37816958	-0.13032080	0.12973369	-0.29809965	-0.89929196	-0.13677593
1956	0.94596125	-0.02536247	-0.08739131	-0.15726631	-0.45312529	-0.74377593
1957	0.04946125	0.13672086	0.51844202	0.14152535	-0.73495862	0.35959907
1958	-0.15324708	-0.21077914	-0.07101631	-0.19664131	0.38258304	-0.27627593
1959	NA	NA	NA			

In [13]: `plot(comp)`

Decomposition of additive time series



```
In [14]: # Forecasting  
births.timeseries.forecasts <- HoltWinters(births.timeseries)  
print(births.timeseries.forecasts)
```

Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:

```
HoltWinters(x = births.timeseries)
```

Smoothing parameters:

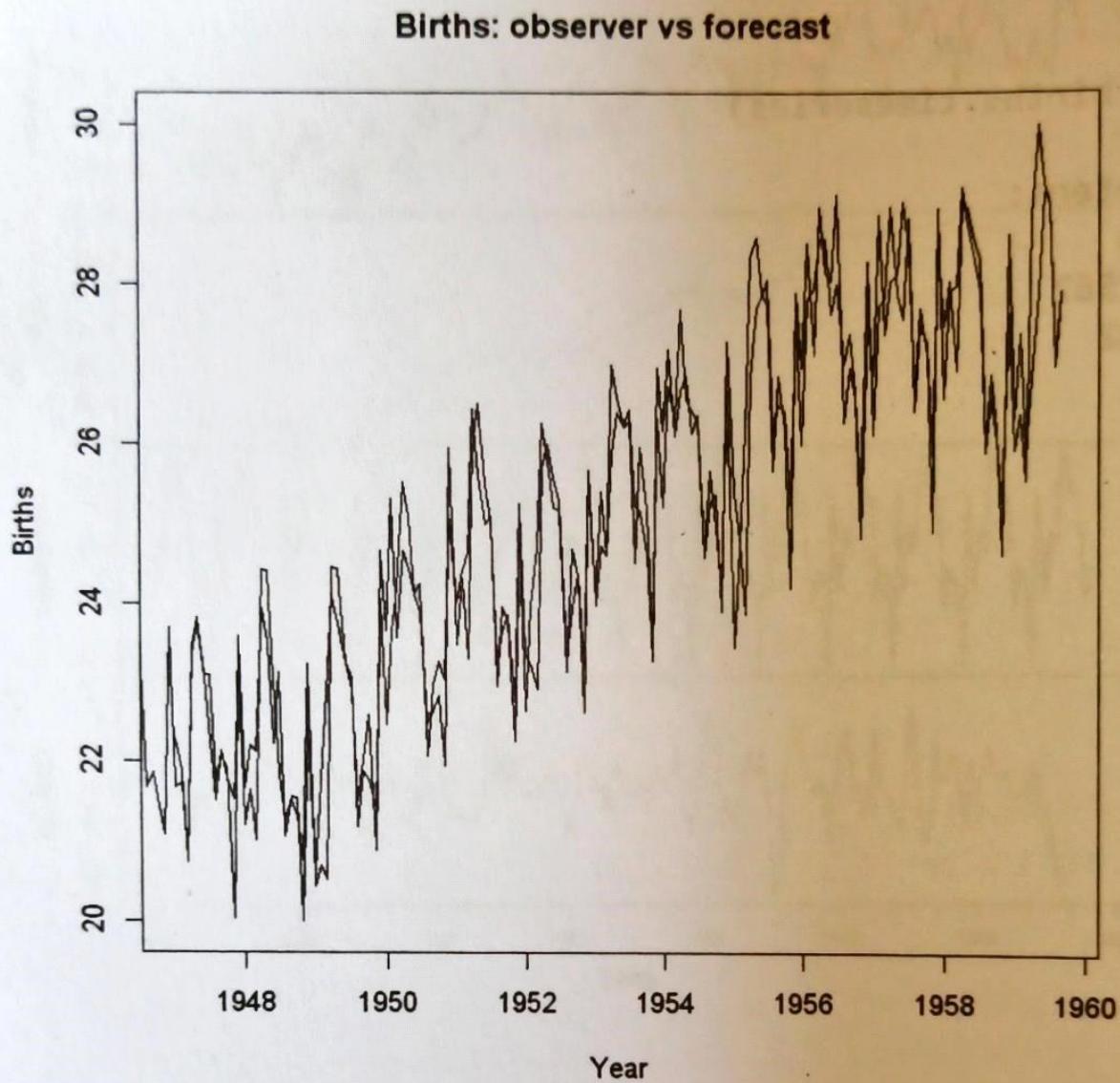
```
alpha: 0.686454  
beta : 0.008857567  
gamma: 0.6828562
```

Coefficients:

```
      [,1]  
a    28.23526659  
b    0.01404313  
s1   -0.77365521  
s2   -2.08404489  
s3   1.11945752  
s4   -0.37371601  
s5   0.83714172  
s6   0.20710404  
s7   1.88468799  
s8   1.59554494  
s9   0.96254105  
s10  0.74685563  
s11 -1.24615200  
s12 -0.33493561
```

- alpha ~ 0.7: khá cao, cho thấy giá trị hiện tại phụ thuộc vào các giá trị gần đây và các giá trị trong quá khứ gần
- beta ~ 0: độ dốc của trend không cập nhập trong thời gian qua
- gamma ~ 0.7: ảnh hưởng của yếu tố mùa tại thời điểm hiện tại có phụ thuộc vào các giá trị gần

```
In [15]: plot(births.timeseries.forecasts,  
            main = "Births: observer vs forecast",  
            xlab='Year', ylab='Births')
```



```
In [16]: births.timeseries.forecasts2 <- forecast:::forecast.HoltWinters(births.timeseries  
print("Next 6 months:")  
print(births.timeseries.forecasts2)  
.forecasts, h=6)
```

[1] "Next 6 months:"

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Oct 1959	27.47565	26.53108	28.42023	26.03105	28.92026
Nov 1959	26.17931	25.03033	27.32828	24.42210	27.93651
Dec 1959	29.39685	28.07186	30.72185	27.37045	31.42326
Jan 1960	27.91772	26.43493	29.40052	25.64998	30.18546
Feb 1960	29.14262	27.51491	30.77033	26.65326	31.63199
Mar 1960	28.52663	26.76370	30.28956	25.83045	31.22280

```
In [17]: plot(births.timeseries.forecasts2, main = "Births: forecast next six months",  
           xlab='Year', ylab='Births')
```

