Time series forecasting

Introduction and exponential smoothing

Julien JACQUES

Universite Lumière Lyon 2

Time series with R

Descriptive statistics for time series

Exponential Smoothing

Times series

A **time series** is:

- a series of data points indexed in time order
- a sequence taken at successive equally spaced points in time.
- it is a sequence of discrete-time data

$$(x_t)_{1 \leq t \leq n} = (x_1, \ldots, x_n)$$

where t is time (seconde, day, year...).

Times series

A **time series** is:

- a series of data points indexed in time order
- a sequence taken at successive equally spaced points in time.
- it is a sequence of discrete-time data

$$(x_t)_{1 \leq t \leq n} = (x_1, \dots, x_n)$$

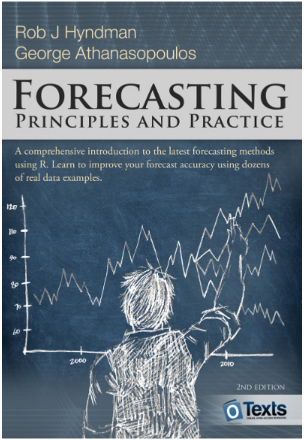
where t is time (seconde, day, year...).

Our goal is to **forecast the future** of the time series

$$x_{n+1}, x_{n+2}, \dots$$

Reference

Hyndman R.J. and Athanasopoulos G. *Forecasting: Principles and Practice*, OTexts, 2013.
https://robjhyndman.com/uwafiles/fpp-notes.pdf



Exemples of times series

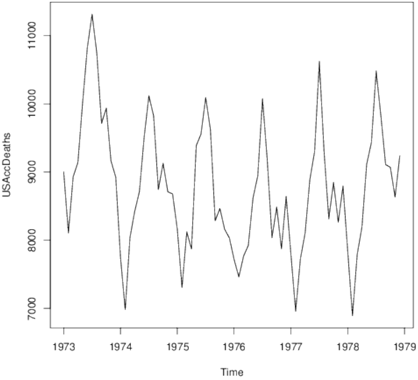


Figure 1: Number of accidental deaths in USA from 1973 to 1978

Exemples of times series

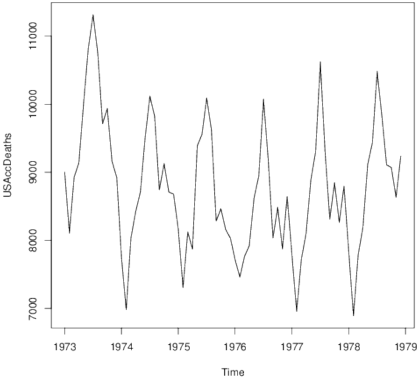


Figure 1: Number of accidental deaths in USA from 1973 to 1978

it seems to be a *periodicity*: we talk about **seasonal pattern**, which occurs when time series are affected by seasonal factor (day of the week, month of the year. . .). The frequency is fixed and knwon.

Exemples of times series

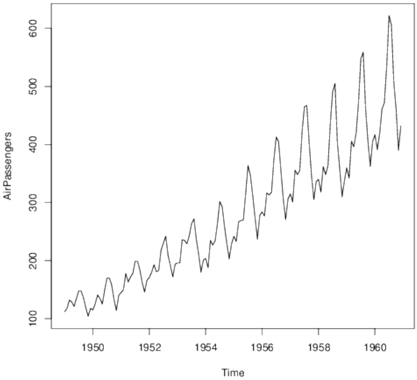


Figure 2: Monthly Airline Passenger Numbers 1949-1960

Exemples of times series

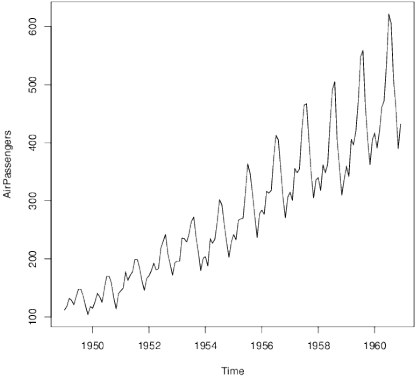


Figure 2: Monthly Airline Passenger Numbers 1949-1960

it seems to be a *seasonal* pattern but also a **trend pattern** (long-time increase or decrease, not necessarily linear)

Exemples of times series

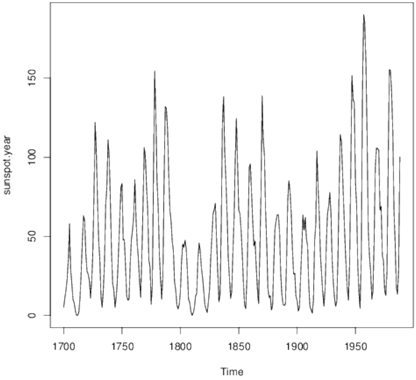


Figure 3: Annual number of sunspots observed on the surface of the sun from 1700 to 1980

Exemples of times series

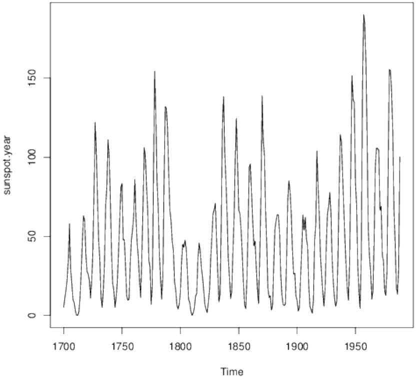


Figure 3: Annual number of sunspots observed on the surface of the sun from 1700 to 1980

it seems to be a *seasonal* pattern or maybe **cyclic pattern** (rises and falls that are not of fixed frequency)

Exemples of times series

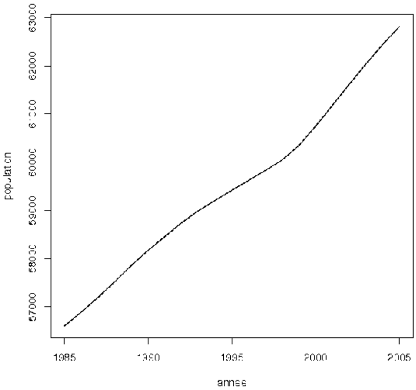


Figure 4: French population from 1985 to 2005

Exemples of times series

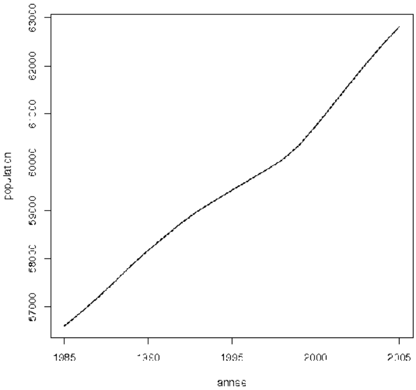


Figure 4: French population from 1985 to 2005

it seems to be a linear *trend*

Exemples of times series

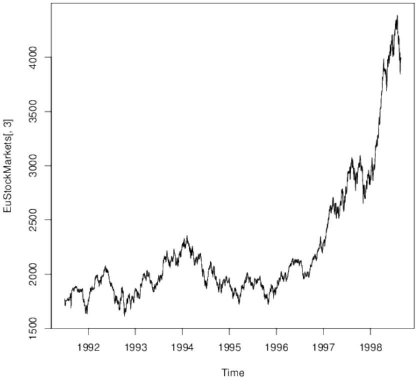


Figure 5: Daily closing values of the CAC40 from 1991 to 1998

Exemples of times series

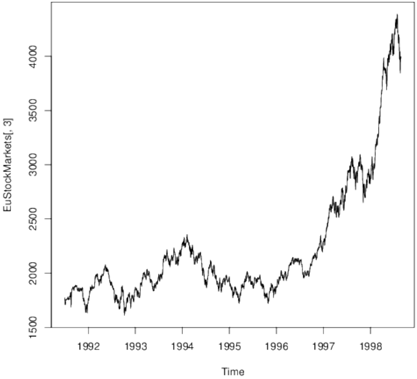


Figure 5: Daily closing values of the CAC40 from 1991 to 1998

it seems to be nothing regular...

Exemples of times series

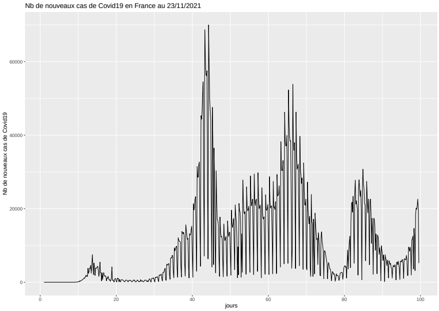


Figure 6: Covid19 number of new cases

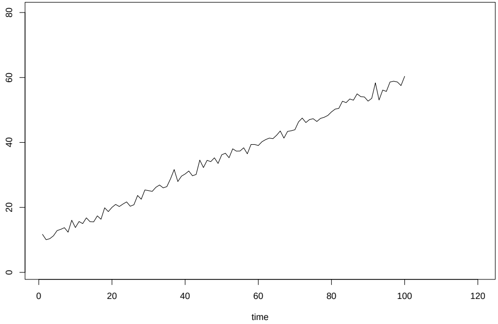
An already known forecasting method

Now, we will start with some simple forecasting method, that you already know!

An already known forecasting method

Load and plot the following series

```
data=read.table(file="http://eric.univ-lyon2.fr/jjacques/Download/DataSet/serie1.txt")
plot(data$V1,type='l',xlim=c(1,120),ylim=c(1,80),xlab='time',ylab='')
```

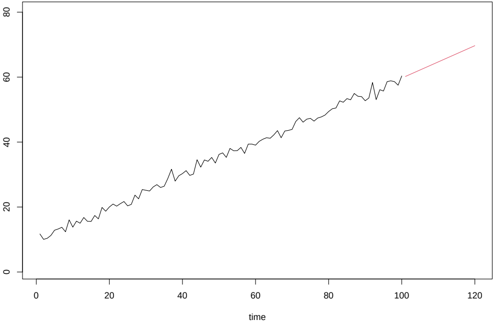


To do: Forecast this series for the next 20 times!

An already known forecasting method

We can use linear regression

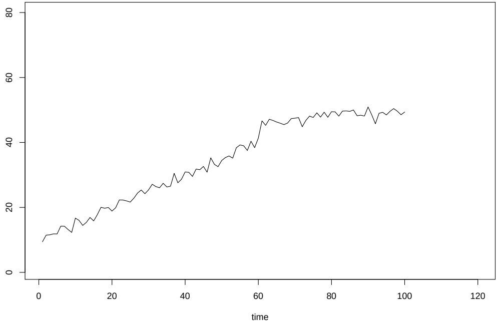
```
t=1:100;x=data$V1
model=lm(x~t)
newt=data.frame(t=101:120)
p=predict(model,newt)
plot(t,x,type='l',xlim=c(1,120),ylim=c(1,80),xlab='time',
     ylab='')
lines(newt$t,p,col=2)
```



An already known forecasting method

Load and plot the following series

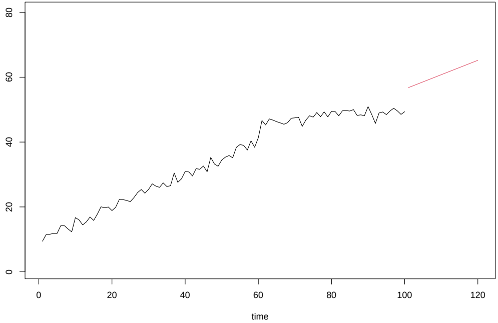
```
data=read.table(file="http://eric.univ-lyon2.fr/jjacques/Download/DataSet/serie2.txt")
plot(data$V1,type='l',xlim=c(1,120),ylim=c(1,80),xlab='time',ylab='')
```



To do: Forecast this series for the next 20 times!

An already known forecasting method

```
t=1:100;x=data$V1
model=lm(x~t)
newt=data.frame(t=101:120)
p=predict(model,newt)
plot(t,x,type='l',xlim=c(1,120),ylim=c(1,80),xlab='time',
     ylab='')
lines(newt$t,p,col=2)
```



Linear regression is not efficient since each observations have the same weight: we should be able to weight the data according to their age. . .

Time series with R

Time series R

In R, the `ts` object is dedicated to time series:

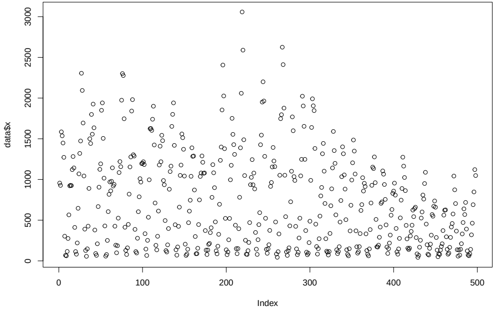
```
data("AirPassengers")
str(AirPassengers)
```

```
##  Time-Series [1:144] from 1949 to 1961: 112 118 132 129
```

Creation of ts object

We load the data from any format (here csv for instance)

```
data=read.csv(file="http://eric.univ-lyon2.fr/jjacques/Download/DataSet/varicelle.csv")
plot(data$x)
```

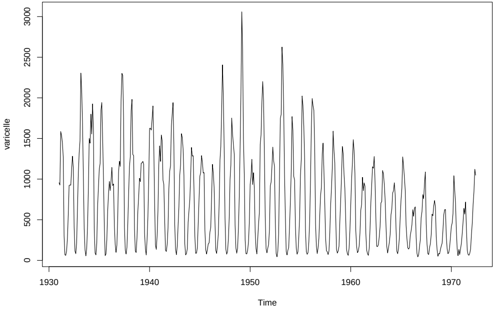


Creation of ts object

We indicate to R the specificity of the ts object:

- monthly data with annual seasonality: `freq=12`
- start in January 1931: `start=c(1931,1)`
- end in June 1972: `end=c(1972,6)`

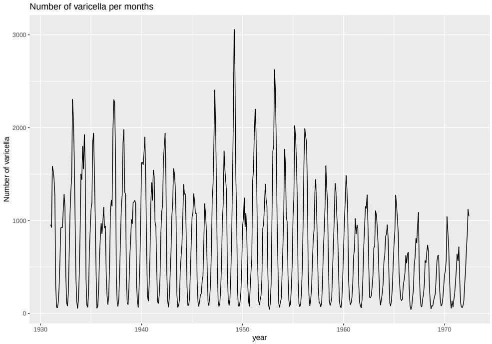
```
varicelle<-ts(data$x,start=c(1931,1),end=c(1972,6),freq=12)
plot(varicelle)
```



Plot with forecast

The forecast library proposes nice plots

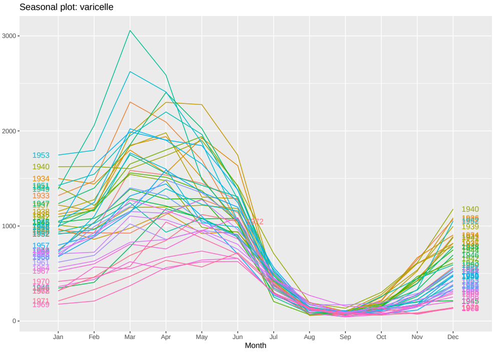
```
library(forecast)
library(ggplot2)
autoplot(varicelle) +
  ggtitle('Number of varicella per months')+
  xlab('year')+
  ylab('Number of varicella')
```



Plot with `forecast`

It could be convenient to use seasonal plot

```
ggseasonplot(varicelle,year.labels= TRUE,year.labels.left=TRUE)
```

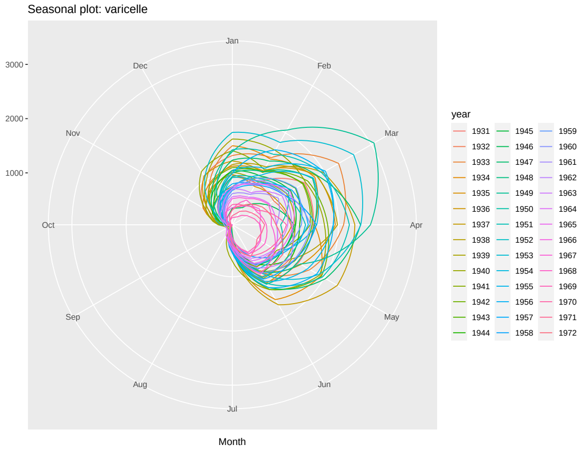


in particular for checking the size of the seasonality

Plot with forecast

or also with the polar option

```
ggseasonplot(varicelle,polar=TRUE)
```

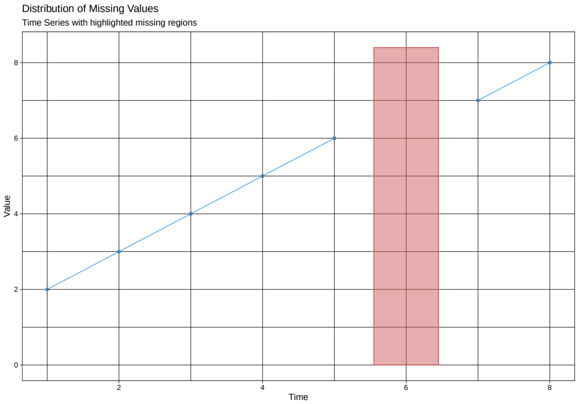


Missing data imputation

Some time series could have missing data.

The following package proposes imputation method.

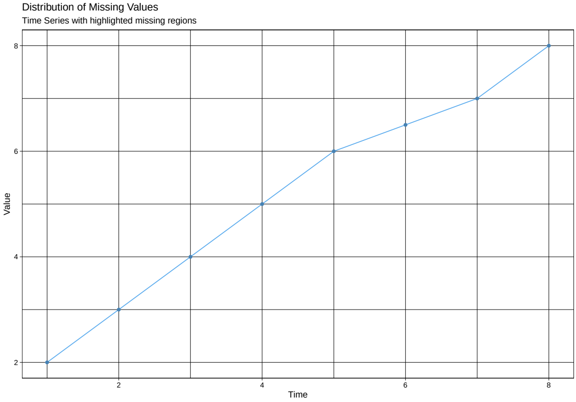
```r
library(imputeTS)
x <- ts(c(2, 3, 4, 5, 6, NA, 7, 8))
ggplot_na_distribution(x)
```



Missing data imputation

Most simple imputation method is interpolation (linear, spline...):

```
x=na_interpolation(x)
ggplot_na_distribution(x)
```



Descriptive statistics for time series

Descriptive statistics for time series

Empirical **mean**:

$$\bar{x}_n = \frac{1}{n}\sum_{t=1}^{n} x_t$$

```r
mean(varicelle)
```

```
## [1] 732.4076
```

Empirical **variance**:

$$\hat{\sigma}_n(0) = \frac{1}{n}\sum_{t=1}^{n}(x_t - \bar{x}_n)^2$$

```r
var(varicelle)
```

```
## [1] 347785.4
```

Descriptive statistics for time series

Empirical **auto-covariance** of order h (*covariance between lagged values*):

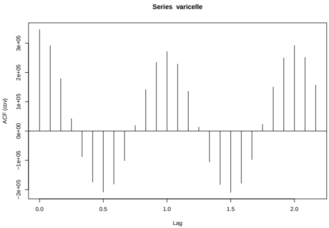
$$\hat{\sigma}_n(h) = \frac{1}{n-h}\sum_{t=1}^{n-h}(x_t - \bar{x}_n)(x_{t+h} - \bar{x}_n),$$

It measures the linear covariance between x_t and x_{t-h}

```
tmp=acf(varicelle,type="cov",plot = FALSE)
tmp$acf[1:3,1,1]
```

```
## [1] 347087.0 291348.5 179126.1
```

```
plot(tmp)
```



Descriptive statistics for time series

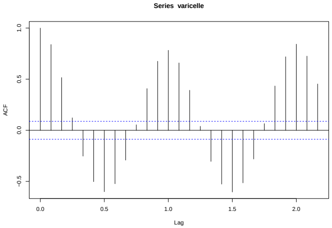
Empirical **auto-correlation** of order h:

$$\hat{\rho}_n(h) = \frac{\hat{\sigma}_n(h)}{\hat{\sigma}_n(0)} \quad \in [-1, 1]$$

```
tmp=acf(varicelle,type="cor",plot = FALSE)
tmp$acf[1:3,1,1]
```

```
## [1] 1.0000000 0.8394105 0.5160841
```

```
plot(tmp)
```



The plot is known as **correlogram**. Values into the blue lines ($\pm 2/\sqrt{n}$) are not significantly different from zero.

Auto-correlation significativity

The Box test tests if there exists at least one among the first `lag` autocorrelations which is significant

```
Box.test(varicelle,lag=10,type="Box-Pierce")
```

```
##
##  Box-Pierce test
##
## data:  varicelle
## X-squared = 1091.7, df = 10, p-value < 2.2e-16
```

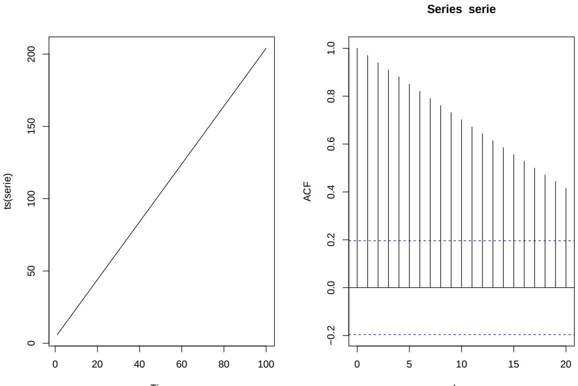
Here the p-value is very low, below than 0.05, that means that there is some significant autocorrelations among the 10 first order autocorrelations.

Auto-correlation properties

- if the time serie $(x_t)_{1\leq t\leq n}$ is a pure linear trend $x_t = at + b$, then for all h:

$$\hat{\rho}_n(h) \xrightarrow[n\to\infty]{} 1$$

```r
serie=2*(1:100)+4
par(mfrow=c(1,2))
plot(ts(serie))
acf(serie)
```

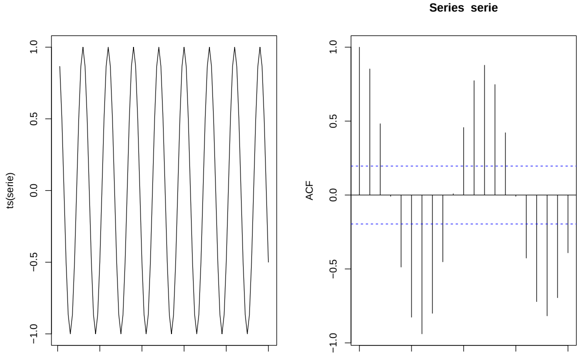


Auto-correlation properties

- if the time serie $(x_t)_{1 \leq t \leq n}$ is a pure seasonal pattern, for instance $x_t = a \cos \frac{2t\pi}{T}$, then for all h:

$$\hat{\rho}_n(h) \xrightarrow[n \to \infty]{} \cos \frac{2h\pi}{T}$$

```
serie=cos(2*pi/12*(1:100))
par(mfrow=c(1,2))
plot(ts(serie))
acf(serie)
```



Auto-correlation properties

Thus, the presence of *trend* and *season pattern* are observable in the auto-correlation plots.

We can also use this plot to check the value of the periodicity...

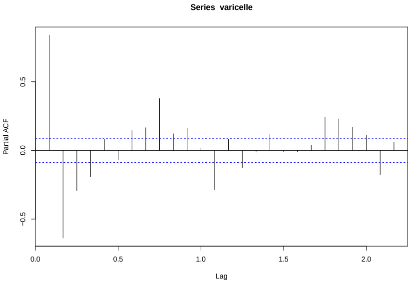
Descriptive statistics for time series

Empirical **partial auto-correlation** of order h measures the linear correlation between x_t and x_{t-h}, **but removing the effect of** $x_{t-1}, \ldots, x_{t-h+1}$

```r
tmp=pacf(varicelle,type="cor",plot = FALSE)
tmp$acf[1:3,1,1]
```

```
## [1]  0.8394105 -0.6382268 -0.2944475
```

```r
plot(tmp)
```



Exercice

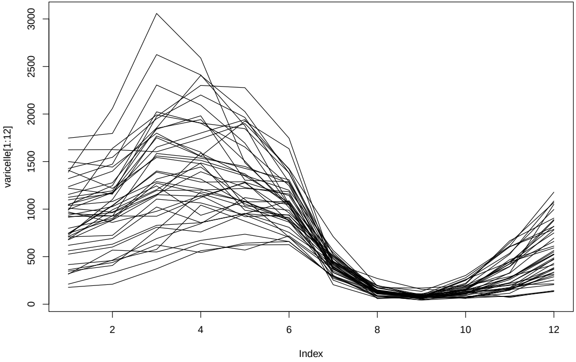
File "http://eric.univ-lyon2.fr/jjacques/Download/DataSet/varicelle.csv" contains the mensual number of varicella from January 1931 to June 1972.

- Load this data set and build a `ts` object.
- Plot the time series.
- Is there some trend, seasonal pattern or cyclic pattern?
- What is the mean mensual number of varicella?
- Plot the correlogram and interpret it.
- Plot the seasonal plot.
- Compute the annual numbers of varicella and plot them from 1931 to 1972.
- What can you say from this two latter graphs?

Correction

- Manual seasonal plot

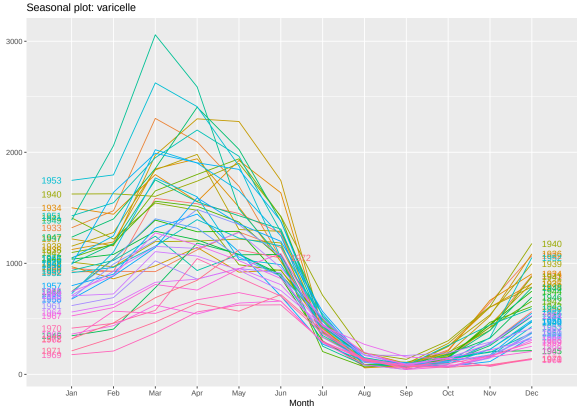
```
plot(varicelle[1:12],type="l",ylim=c(min(varicelle),max(var
for (i in 1:41) lines(varicelle[(1+12*i):(12*(i+1))])
```



Correction

- Automatic seasonal plot

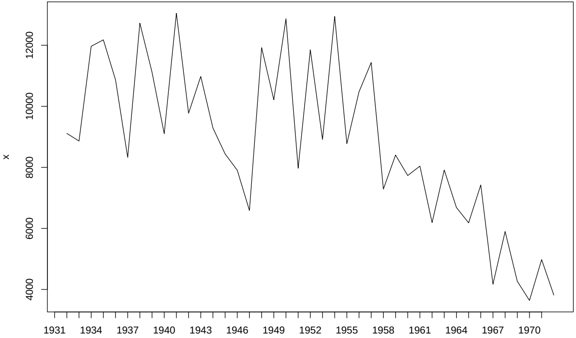
```
ggseasonplot(varicelle,year.labels= TRUE,year.labels.left=TRUE)
```



Correction

- Annual evolution (manually)

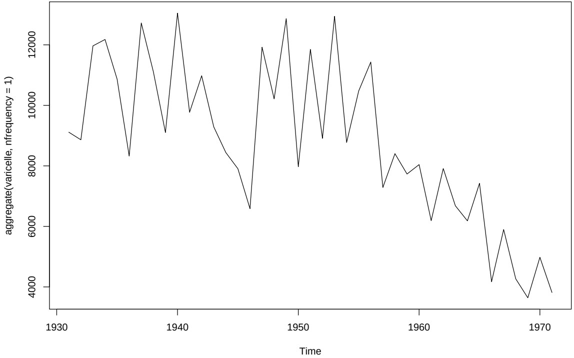
```
x=rep(0,41)
for (i in 0:40) x[i+1]<-sum(varicelle[(1+12*i):(12*(i+1))])
plot(x,type='l',xaxt='n',xlab='')
axis(1,at = 0:40,labels = 1931:1971)
```



Correction

- Annual evolution (automatically)

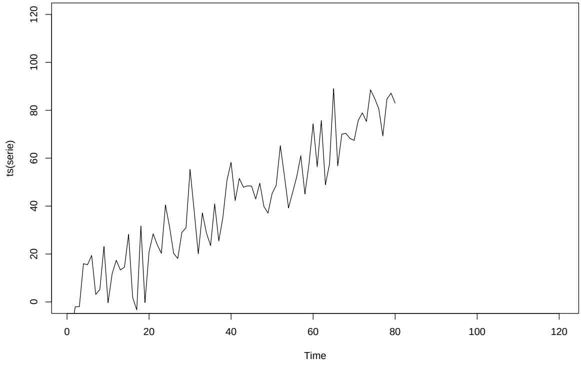
```
plot(aggregate(varicelle,nfrequency=1))
```



Exponential Smoothing

Introduction to exponential smoothing

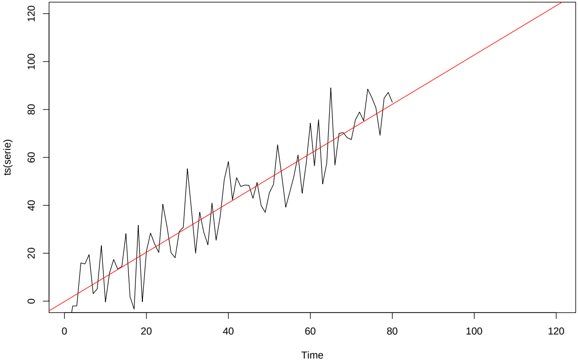
Have you an idea of forecasting model?



Introduction to exponential smoothing

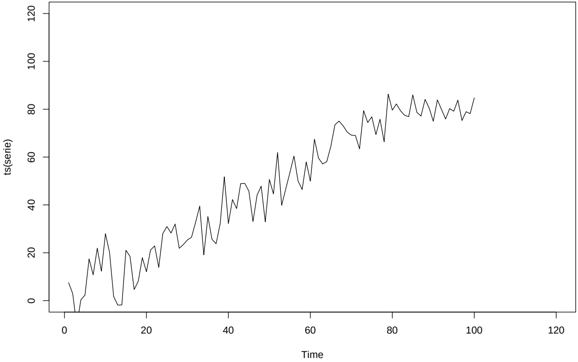
Maybe a linear regression?

```
mod=lm(serie~temps)
plot(ts(serie),xlim=c(1,120),ylim=c(0,120))
abline(mod$coefficients,col="red")
```



Introduction to exponential smoothing

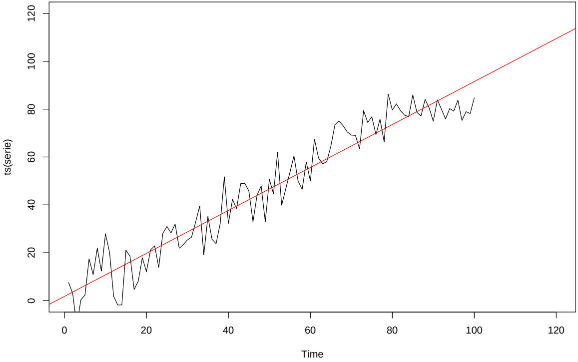
And now?



Introduction to exponential smoothing

Linear regression again?

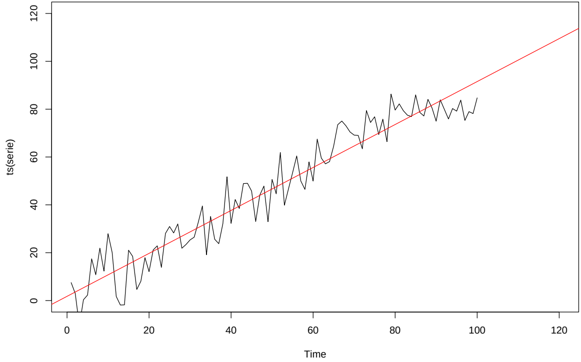
```
mod=lm(serie~temps)
plot(ts(serie),xlim=c(1,120),ylim=c(0,120))
abline(mod$coefficients,col="red")
```



Introduction to exponential smoothing

Linear regression again?

```
mod=lm(serie~temps)
plot(ts(serie),xlim=c(1,120),ylim=c(0,120))
abline(mod$coefficients,col="red")
```



Linear regression is not efficient since each observations have the same weight: we should be able to weight the data according to their age...

Exponential Smoothing

Exponential Smoothing is a collection of models (constant, linear, seasonal. . .) in which **the importance of the observed data decreases with their age**

Simple Exponential Smoothing (SES)

Given a smoothing constant $0 < \alpha < 1$, forecast with **Simple Exponential Smoothing** is:

$$\hat{x}_{n,h} = \alpha \sum_{j=0}^{n-1} (1-\alpha)^j x_{n-j}.$$

With this model, forecast is:

- **constant** (does not depend of horizon h),
- a **ponderated mean** of past observations,
- the closer is α to 1, the faster the weight of past observations decreases.

Constant α should be tuned on the data.

Evaluating forecast accuracy

How α can be chosen? More generaly, how to compare different forecasting models?

The data set $x_1, \ldots, x_n$ is separated into **train and test subsets**:

- about 80% for the training data, to estimate model parameter
- the 20% most recents, to evaluate forecast accuracy

Evaluating forecast accuracy

The size of 20% for the test data can be adapted according to

- the forecasting horizon h we want to predict
- the size of the season pattern (select 1 or 2 season in the test dataset)

Warning: once the forecasting model is selected, it should be estimated again on the whole dataset $x_1, \ldots, x_n$ before to forecast the future.

Evaluating forecast accuracy

On the test data, several indicators can be computed:

- **Root Mean Square Error**:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n-m}\sum_{h=1}^{n-m}(\hat{x}_{m,h} - x_{m+h})^2}$$

  where m is the size of training set.

- **Mean Absolute Percentage Error**:

$$\mathrm{MAPE} = \frac{100}{n-m}\sum_{h=1}^{n-m}\frac{|\hat{x}_{m,h} - x_{m+h}|}{x_{m+h}}$$

Tools to subset a time series

Extract all data from 1950

```
window(varicelle,start=1950)
```

Extract the first or last observations

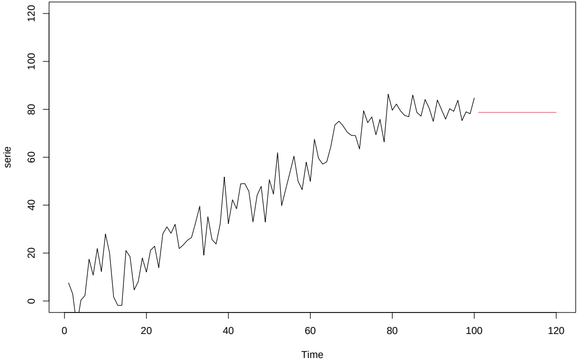
```
head(varicelle,12)
tail(varicelle,6)
```

The `subset` function allows more type of subsetting.

Simple Exponential Smoothing (SES)

SES forecast with $\alpha = 0.1$

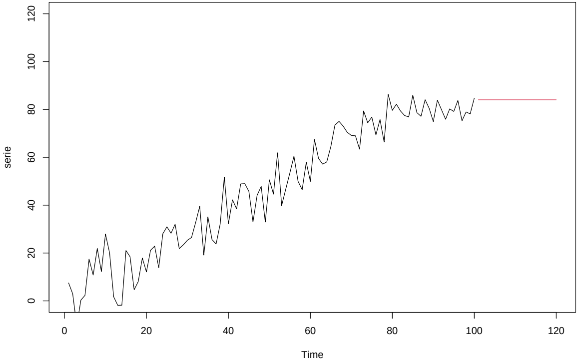
```
serie=ts(serie)
LES=HoltWinters(serie,alpha=0.1,beta=FALSE,gamma=FALSE)
plot(serie,xlim=c(1,120),ylim=c(0,120))
p<-predict(LES,n.ahead=20)
lines(p,col=2)
```



Simple Exponential Smoothing (SES)

SES forecast with $\alpha = 0.9$

```
LES=HoltWinters(serie,alpha=0.9,beta=FALSE,gamma=FALSE)
plot(serie,xlim=c(1,120),ylim=c(0,120))
p<-predict(LES,n.ahead=20)
lines(p,col=2)
```



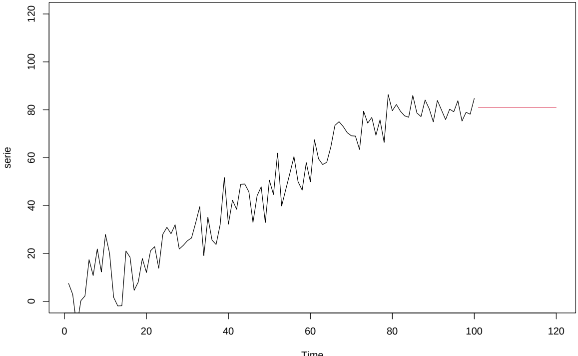
Simple Exponential Smoothing (SES)

For an automatic and optimal choice of α, let choose option `alpha=NULL`:

```
LES=HoltWinters(serie,alpha=NULL,beta=FALSE,gamma=FALSE)
print(LES$alpha)
```

```
## [1] 0.3724549
```

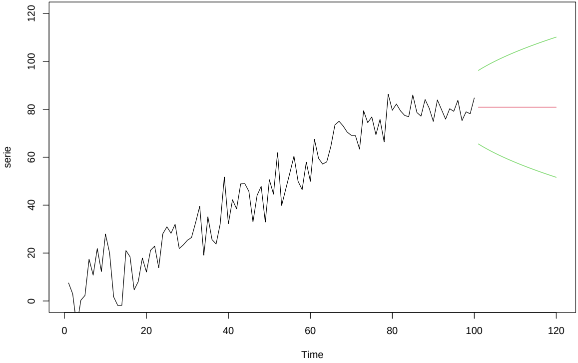
```
plot(serie,xlim=c(1,120),ylim=c(0,120))
p<-predict(LES,n.ahead=20)
lines(p,col=2)
```



Simple Exponential Smoothing (SES)

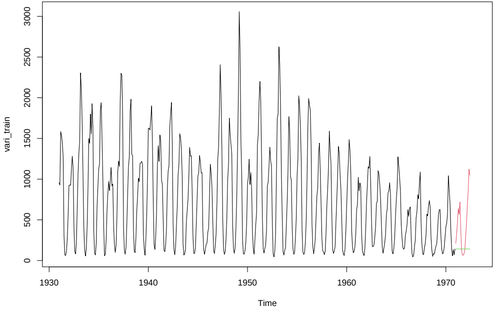
We can also add a forecasting interval.

```
LES=HoltWinters(serie,alpha=NULL,beta=FALSE,gamma=FALSE)
plot(serie,xlim=c(1,120),ylim=c(0,120))
p<-predict(LES,n.ahead=20,prediction.interval = TRUE)
lines(p[,1],col=2)
lines(p[,2],col=3);lines(p[,3],col=3);
```



Varicella forecasting with SES

```r
data=read.csv(file="http://eric.univ-lyon2.fr/jjacques/Download/DataSet/varicelle.csv")
vari_train<-ts(data$x[1:480],start=c(1931,1),end=c(1970,12),freq=12)
vari_test<-ts(data$x[481:498],start=c(1971,1),end=c(1972,6),freq=12)
plot(vari_train,xlim=c(1931,1973))
lines(vari_test,col=2)
SES=HoltWinters(vari_train,alpha=NULL,beta=FALSE,gamma=FALSE)
p1<-predict(SES,n.ahead=18)
lines(p1,col=3)
```



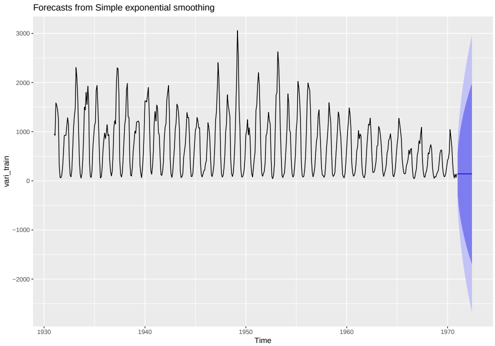
Varicella forecasting with SES

SES is also available through the `forecast` package

```
SES=ses(vari_train,h=18)
round(accuracy(SES),2)
```

```
##                ME   RMSE    MAE    MPE MAPE MASE ACF1
## Training set -1.6 338.15 251.29 -24.53   61 1.04 0.51
```

```
autoplot(SES)
```



Different models of exponential smoothing

- Simple Exponential Smoothing: forecasting with a constant
- Non seasonal Holt-Winters smoothing: forecasting with a linear trend
- Additive seasonal Holt-Winters: forecasting with a linear trend plus a seasonal pattern
- Multiplicative seasonal Holt-Winters: forecasting with a linear trend time a seasonal pattern

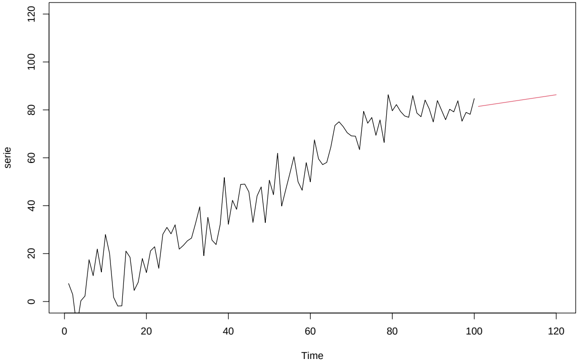
Non seasonal Holt-Winters smoothing

Forecasting is done with the linear trend

$$\hat{x}_{n,h} = \hat{a}_1 + \hat{a}_2 h.$$

This model has two smoothing constants (α, β) acting on a_1 and a_2.

```
LES=HoltWinters(serie,alpha=NULL,beta=NULL,gamma=FALSE)
plot(serie,xlim=c(1,120),ylim=c(0,120))
p<-predict(LES,n.ahead=20)
lines(p,col=2)
```



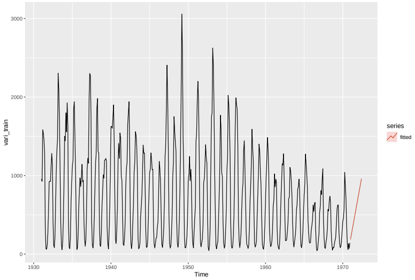
Varicella forecasting with non seasonal HW smoothing

We can also use the `forecast` package

```
HOLT=holt(vari_train,h=18)
round(accuracy(HOLT),2)
```

```
##                ME   RMSE    MAE   MPE  MAPE MASE ACF1
## Training set 0.22 330.42 259.65 54.15 89.34 1.08 0.01
```

```
autoplot(vari_train) + autolayer(HOLT,series='fitted',PI=FALSE)
```



the option `PI=FALSE` remove the prediction interval

Damped non seasonal Holt-Winters smoothing

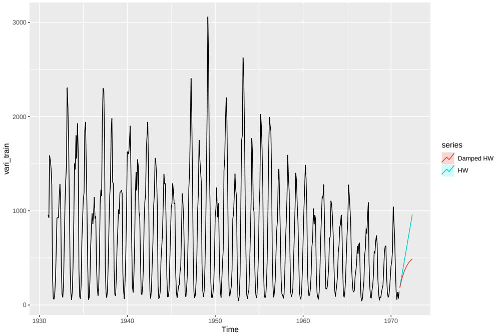
It is possible to add a damping parameter $0 < \phi < 1$ in order to dampen the trend

$$\hat{x}_{n,h} = \hat{a}_1 + \hat{a}_2(\phi + \phi^2 + \ldots + \phi^h).$$

- $\phi = 1$ lead to the usual non seasonal HW
- using $0 < \phi < 1$ dampens the trend so that it approaches a constant in the future

Varicella forecasting with non seasonal HW smoothing

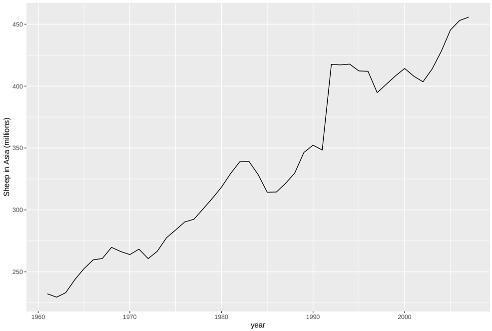
```
HOLT1=holt(vari_train,h=18)
HOLT2=holt(vari_train,damped=TRUE,phi=0.9,h=18)
autoplot(vari_train) +
  autolayer(HOLT1,series='HW',PI=FALSE) +
  autolayer(HOLT2,series='Damped HW',PI=FALSE)
```



Example: Sheep livestock in Asia

We will compare SES, HW and damped HW for forecasting the sheep livestock population in Asia.

```r
library(fpp)
data(livestock)
autoplot(livestock) +
  xlab("year") +
  ylab("Sheep in Asia (millions)")
```



Example: Sheep livestock in Asia

To compare the method we can divide the time series into train / test subset, but we can also use time series cross validation implemented in `tsCV`:

```
e1 <- tsCV(livestock, ses, h=1)
e2 <- tsCV(livestock, holt, h=1)
e3 <- tsCV(livestock, holt, damped=TRUE, h=1)
```

To compare MSE:

```
mean(e1^2, na.rm=TRUE)
```

```
## [1] 178.2531
```

```
mean(e2^2, na.rm=TRUE)
```

```
## [1] 173.365
```

```
mean(e3^2, na.rm=TRUE)
```

```
## [1] 162.6274
```

The best model seems to be the Damped HW

Example: Sheep livestock in Asia

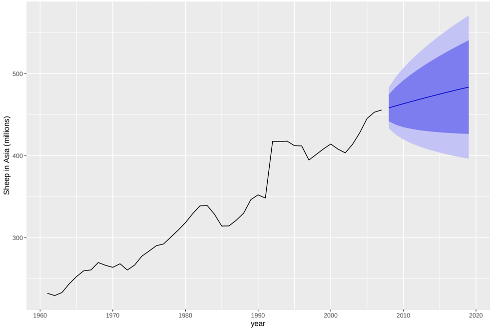
```r
HWd=holt(livestock,damped=TRUE,h=12)
HWd[["model"]]
```

```
## Damped Holt's method
##
## Call:
##  holt(y = livestock, h = 12, damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 3e-04
##     phi   = 0.9798
##
##   Initial states:
##     l = 223.35
##     b = 6.9046
##
##   sigma:  12.8435
##
##      AIC     AICc      BIC
## 427.6370 429.7370 438.7379
```

Example: Sheep livestock in Asia

Forecasting with the Damped HW

```
autoplot(livestock) +
  autolayer(HWd) +
  xlab("year") +
  ylab("Sheep in Asia (millions)")
```



Additive seasonal Holt-Winters

Now, we will add a seasonal pattern to the HW linear trend:

$$y_t = a_1 + a_2(t - n) + s_t,$$

where s_t is a seasonal pattern of period T.

Forecasting are:

$$\begin{aligned} \hat{x}_{n,h} &= \hat{a}_1 + \hat{a}_2 h + \hat{s}_{n+h-T} \quad 1 \leq h \leq T, \\ \hat{x}_{n,h} &= \hat{a}_1 + \hat{a}_2 h + \hat{s}_{n+h-2T} \quad T+1 \leq h \leq 2T, \end{aligned}$$

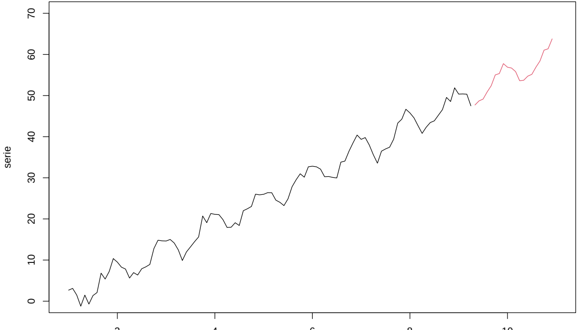
and so on for $h \geq 2T$.

This model has 3 smoothing constant α, β et γ: greater they are, lower are the importance of oldest observations. They act repectively on a_1, a_2 and s_t.

Additive seasonal Holt-Winters

In order to estimate the seasonal pattern, we should precise the corresponding period

```
serie=0.5*(1:100)+rnorm(100,0,1)+3*cos(pi/6*(1:100))
serie=ts(serie,start=c(1,1),end=c(9,4),frequency = 12)
LES=HoltWinters(serie,alpha=NULL,beta=NULL,gamma=NULL)
plot(serie,xlim=c(1,11),ylim=c(0,70))
p<-predict(LES,n.ahead=20)
lines(p,col=2)
```



Multiplicative seasonal Holt-Winters

The multiplicative seasonal Holt-Winters model is

$$y_t = [a_1 + a_2(t-n)] \times s_t,$$

where s_t is a seasonal pattern of period T.

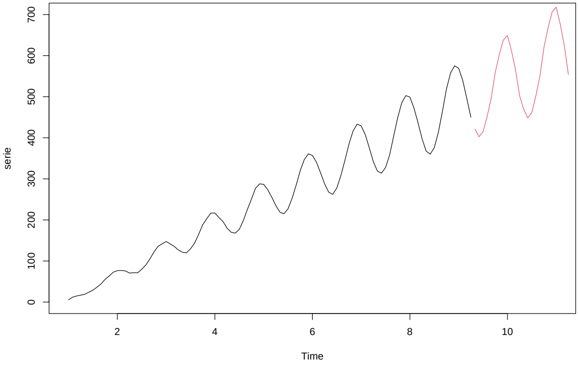
Forecasting are:

$$\begin{aligned}\hat{x}_{n,h} &= [\hat{a}_1 + \hat{a}_2 h]\hat{s}_{n+h-T} \quad 1 \leq h \leq T, \\ \hat{x}_{n,h} &= [\hat{a}_1 + \hat{a}_2 h]\hat{s}_{n+h-2T} \quad T+1 \leq h \leq 2T,\end{aligned}$$

and so on for $h \geq 2T$.

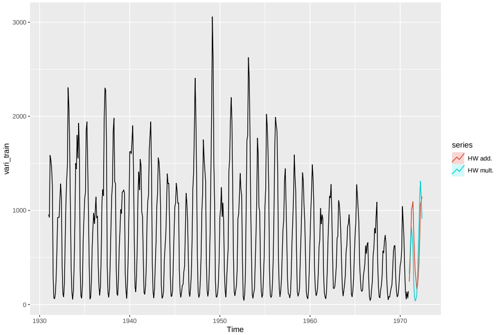
Multiplicative seasonal Holt-Winters

```
serie=5*(1:100)+rnorm(100,0,1)+cos(pi/6*(1:100))*(1:100)
serie=ts(serie,start=c(1,1),end=c(9,4),frequency = 12)
LES=HoltWinters(serie,alpha=NULL,beta=NULL,gamma=NULL,
                seasonal = "multi")
plot(serie,xlim=c(1,11),ylim=c(0,700))
p<-predict(LES,n.ahead=24)
lines(p,col=2)
```



Varicella forecasting with seasonal HW smoothing

```
fit1=hw(vari_train,seasonal='additive',h=18)
fit2=hw(vari_train,seasonal='multiplicative',h=18)
autoplot(vari_train) +
  autolayer(fit1,series='HW add.',PI=FALSE) +
  autolayer(fit2,series='HW mult.',PI=FALSE)
```



Varicella forecasting with seasonal HW smoothing

We can compute the RMSE of both model

```r
print(sqrt(mean((fit1$mean-vari_test)^2)))
```

```
## [1] 238.2674
```

```r
print(sqrt(mean((fit2$mean-vari_test)^2)))
```

```
## [1] 214.7901
```

The multiplicative seasonal Holt-Winters seems to be the best.

Varicella forecasting with seasonal HW smoothing

We can also compare with damped version of the seasonal HW, but the results are not better:

```
fit3=hw(vari_train,seasonal='additive',damped=TRUE,h=18)
fit4=hw(vari_train,seasonal='multiplicative',damped=TRUE,h=18)
print(sqrt(mean((fit1$mean-vari_test)^2)))
```

```
## [1] 238.2674
```

```
print(sqrt(mean((fit2$mean-vari_test)^2)))
```

```
## [1] 214.7901
```

```
print(sqrt(mean((fit3$mean-vari_test)^2)))
```

```
## [1] 279.7424
```

```
print(sqrt(mean((fit4$mean-vari_test)^2)))
```

```
## [1] 375.6358
```

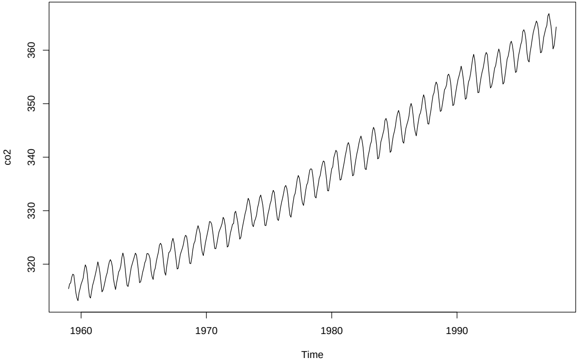
Exercice: forecasting of co2 concentration

File co2 contains CO2 concentrations near Mauna Loa volcano (Hawaï) from 1959 to 1997.

1. plot the data
2. which exponential smoothing model could be appropriate?
3. In order to validate your choice, evaluate your forecast on data between 1990 and 1997 using as training set data from 1959 to 1989.
4. If your forecast seems correct, let use this model to forecast co2 concentration from 1997 to 2007. If not, try other exponential smoothing models.

Exercice: forecasting of co2 concentration

```
plot.ts(co2)
```



Exponential smoothing: conclusion

Exponential smoothing is an effective forecasting method, which takes into account:

- a linear trend in the series
- a seasonal pattern

These components of the series are **deterministic** (i.e. not stochastic).

In the sequel, we will see models for the **stochastic** part of the time series.

San Francisco precipitation forecast

San Fransisco precipitation from 1932 to 1966 are available here:
http://eric.univ-lyon2.fr/jjacques/Download/DataSet/sanfran.dat

- Data until 1963 will be used as training set, in order to forecast precipitations for 1964, 1965 and 1966.
- Test several exponential smoothing models, and plot on the same graph the forecast and actual values.
- Which model seems to be graphically the best? And for RMSE?
- Interpret the value of the smoothing (and eventually damping) parameters.