We view each value y_t of a time series y as the realized value of a random variable.[1] The corresponding family of random variables is called a (**stochastic**) **process**.

Our notation does not distinguish between a random variable and the observed value that it takes. It will always be clear from the context whether the random variable or the observed value is meant.

A stochastic process is said to have a **trend** if not all of its random variables have the same mean.

Examples of trends:

Upward trend:	$s<t \Rightarrow Ey_s<Ey_t$
Downward trend:	$s<t \Rightarrow Ey_s>Ey_t$
Linear trend:	$Ey_t=a+bt,\ b\neq 0$
Exponential trend:	$Ey_t=\exp(a+bt),\ b\neq 0$

[1] In case of a process with domain $\mathbf{Z}$, we usually write y_t instead of $y(t)$.

A process x is called **white noise** if all of its random variables have the same mean and the same variance and are uncorrelated, i.e.,

$$Ex_s=Ex_t \ \forall s,t\in\mathbf{Z},$$
$$\mathrm{var}(x_s)=\mathrm{var}(x_t) \ \forall s,t\in\mathbf{Z},$$

and

$$s\neq t \Rightarrow \mathrm{cov}(x_s,x_t)=0 \ \forall s,t\in\mathbf{Z}.$$

If the process

$$y_t = a + bt + x_t.$$

is the sum of a linear trend and a white noise with mean zero and variance σ^2, we have

$$Ey_t = a + bt,\ \mathrm{var}(y_t) = \sigma^2.$$

The process

$$z_t = \tfrac{1}{2k+1} \sum_{j=-k}^{k} y_{t+j}$$

obtained by averaging each y_t with its k nearest neighbors in the past and its k nearest neighbors in the future has the same trend as the original process, because

$$Ez_t = \tfrac{1}{2k+1} \sum_{j=-k}^{k} Ey_{t+j} = \tfrac{1}{2k+1} \sum_{j=-k}^{k} (a + b(t+j) + \underbrace{Ex_{t+j}}_{=0})$$

$$= a + bt + \tfrac{b}{2k+1} \underbrace{\sum_{j=-k}^{k} j}_{=0}.$$

But its variance is much smaller because

$$\mathrm{var}(z_t) = \tfrac{1}{(2k+1)^2} \mathrm{var}\left(\sum_{j=-k}^{k} y_{t+j} \right)$$

$$= \tfrac{1}{(2k+1)^2} \mathrm{var}\left(\sum_{j=-k}^{k} (a + b(t+j) + x_{t+j}) \right)$$

$$= \tfrac{1}{(2k+1)^2} \mathrm{var}\left(\sum_{j=-k}^{k} (a + b(t+j)) + \sum_{j=-k}^{k} x_{t+j} \right)$$

$$= \tfrac{1}{(2k+1)^2} \mathrm{var}\left(\sum_{j=-k}^{k} x_{t+j} \right)$$

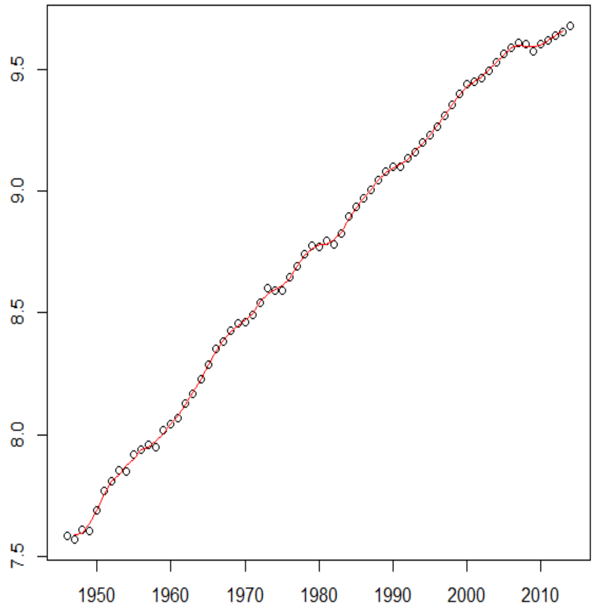
$$= \tfrac{1}{(2k+1)^2} \sum_{j=-k}^{k} \underbrace{\mathrm{var}(x_{t+j})}_{=\sigma^2}$$

$$= \tfrac{\sigma^2}{2k+1}.$$

Thus z_t is a suitable estimator for Ey_t. FS

Exercise: Estimate the trend of the log GDP by averaging over $2k+1$ neighboring values. Use $k=1$.[2]

```
S.1 <- (lag(y.ts,k=-1)+y.ts+lag(y.ts,k=1))/3
plot(y.ts,type="p",xlab=" ",ylab=" ")
lines(S.1,col="red")
```



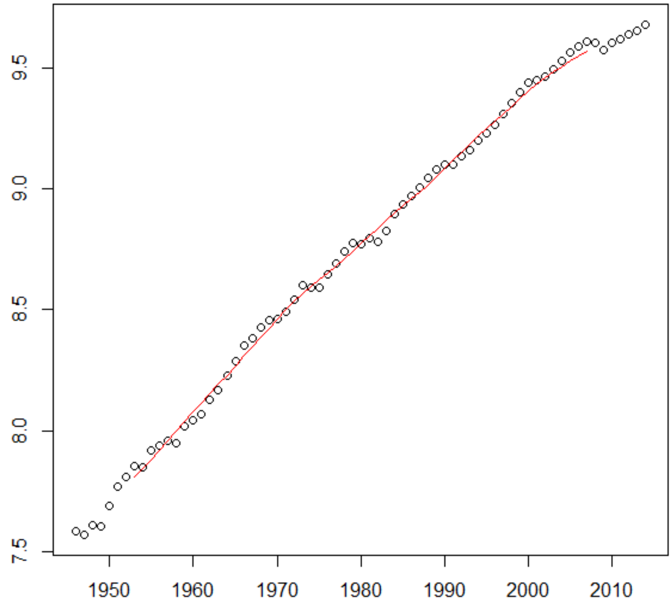
[2] Only the commands are shown, not the promts.

Exercise: Estimate the trend of the log GDP by averaging over $2k+1$ neighboring values. Use $k=7$.

```
S.7 <- y.ts
for (i in 1:7)
    { S.7 <- S.7+lag(y.ts,k=-i)+lag(y.ts,k=i) }
# for loop: everything within the curly brackets
#               is done 7 times with i taking the values 1 to 7.
S.7 <- S.7/15
plot(y.ts,type="p",xlab=" ",ylab=" ")
lines(S.7,col="red")
```

The larger the value of k, the smoother is the result. The graph obtained for $k=7$ supports the hypothesis of a break in the 1970s.

Note: The missing values are due to the fact that we cannot compute averages near the boundary.



Exercise: Estimate the trend of the log GDP by averaging over $2k+1$ neighboring values. Use different values of k.

Different smoothed versions of the log GDP can be obtained efficiently by defining a suitable function and applying this function repeatedly, each time using a different value of k.

```
smooth <- function(y,k)
  {  y.sm <- y
     for (i in 1:k) y.sm <- y.sm+lag(y,k=-i)+lag(y,k=i)
     y.sm <- y.sm/(2*k+1)
     return(y.sm)
  }
```

This function has two arguments. **y** is the time series to be smoothed and **k** determines the degree of smoothing. The function returns the smoothed time series. All other variables defined in the function (like **i**) disappear after the function is executed.

The function **smooth** is used like this:

```
S.7 <- smooth(y.ts,7)
plot(y.ts,type="p",xlab=" ",ylab=" ")
lines(S.7,col="red")
```

When the function is called, the real arguments **y.ts** and **7** are passed to the function, where they replace the formal (or dummy) arguments **y** and **k**.

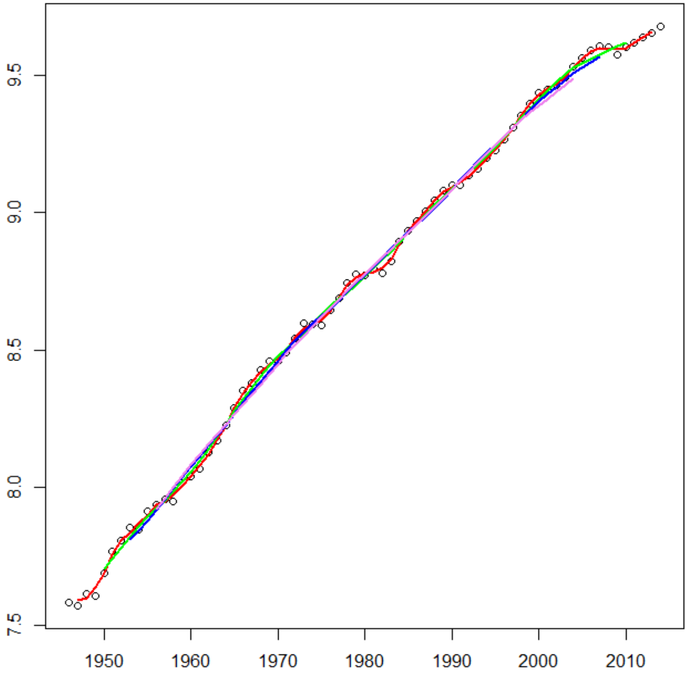
The smoothed versions S.1, S.4, S.7, S.10 of the log GDP could be obtained as follows.

```
K <- c(1,4,7,10)
for (k in K)
    { assign(paste("S",k,sep="."),smooth(GDP.lts,k)) }
```

Note:

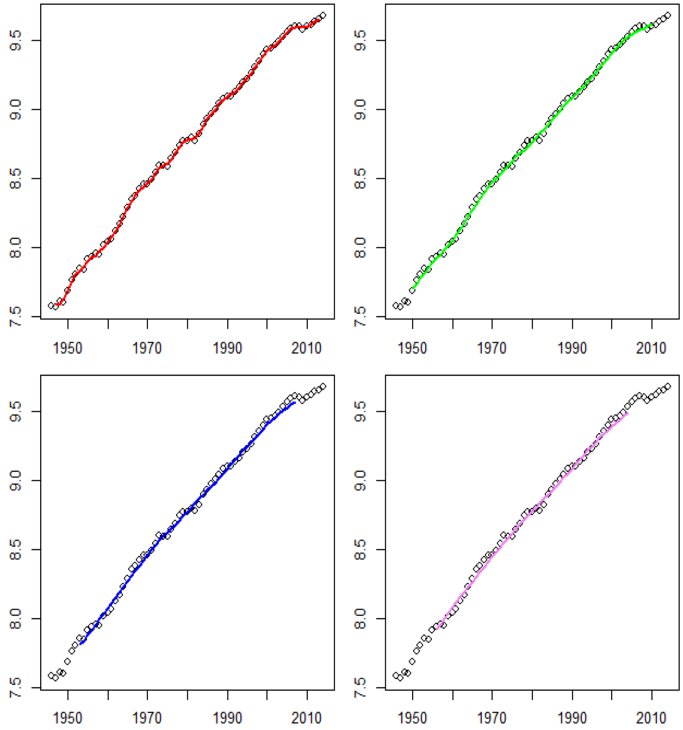
paste("S",7) concatenates the character **"S"** and the number **7** after coverting the number to a character.
paste("S",7,sep=".") separates the two elements by ".".
assign("S.7",3.5) assigns the value 3.5 to the name "S.7".

```
par(mar=c(2,2,1,1))
# setting graphical parameters:
#    mar(i,j,k,l) … lines of margin
#    i: bottom, j: left, k: top, l: right
plot(y.ts,type="p")
COL <- c("red","green","blue","violet")
for (i in 1:length(K)) {
    S <- smooth(y.ts,K[i])
    lines(S,col=COL[i],lwd=2) }
```



This plot is confusing because there are too many lines.

```
par(mfrow=c(2,2),mar=c(2,2,1,1))
#    mfrow=c(i,j) … i rows of j plots
for (i in 1:length(K))
   { plot(y.ts,type="p")
     k <- K[i]
     S <- smooth(y.ts,k)
     lines(S,col=COL[i],lwd=2)
   }
```



More generally, we can use weighted sums for the estimation of the trend:

$$z_t = \sum_{j=-k}^{k} w_j y_{t+j}$$

Such a transformation is called a **finite moving average (MA) filter**.

Usually the weights w_j are chosen such that

$$\sum_{j=-k}^{k} w_j = 1 .$$

If, in addition, the filter is **symmetric**, i.e.,

$$w_{-j}=w_j \ \forall j,$$

then a linear trend can pass without distortion.

Indeed, if

$$y_t=a+bt+x_t,$$

where

$$Ex_t=0 \ \forall t,$$

then

$$\begin{aligned} Ez_t &= \sum_{j=-k}^{k} w_j(a+b(t+j)+\underbrace{Ex_{t+j}}_{=0}) \\ &= (a+bt)\underbrace{\sum_{j=-k}^{k} w_j}_{=1} + b\sum_{j=-k}^{k} jw_j \\ &= a+bt + b(0\cdot w_0+(-1\cdot w_{-1}+1\cdot w_1)+\ldots+(-k\cdot w_{-k}+k\cdot w_k)) \\ &= a+bt . \end{aligned}$$

FW

Exercise: Show that the symmetric filter with weights

$$w_0 = \tfrac{1}{2}\,,\; w_1 = \tfrac{1}{3}\,,\; w_2 = -\tfrac{1}{12}$$

allows a quadratic trend

$$y_t = a + bt + ct^2$$

to pass without distortion. FQ

Exercise: Find another symmetric filter with $k=2$ that allows a quadratic trend to pass without distortion. F2

Exercise: Show that the (symmetric) **Spencer filter** with weights

$$w_0,\ldots,w_7:\; \tfrac{74}{320}\,,\; \tfrac{67}{320}\,,\; \tfrac{46}{320}\,,\; \tfrac{21}{320}\,,\; \tfrac{3}{320}\,,\; -\tfrac{5}{320}\,,\; -\tfrac{6}{320}\,,\; -\tfrac{3}{320}$$

allows a cubic trend

$$y_t = a + bt + ct^2 + dt^3$$

to pass without distortion. F3

Exercise: Apply both the simple MA filter with weights

$$w_{-1} = w_0 = w_1 = \tfrac{1}{3}$$

and the more sophisticated filter from Exercise FQ to the log GDP and compare the results.

Exercise: Apply both the simple MA filter with weights

$$w_{-2} = w_{-1} = w_0 = w_1 = w_2 = \tfrac{1}{5}$$

and the Spencer filter to the log GDP and compare the results.

The figures produced in the last two exercises show that methods with nice theoretical properties do not necessarily perform better than more primitive methods. In our case, even the opposite seems to be true, because the simple filters basically do the same job as the more sophisticated filters but produce fewer missing values at each end of the series.

A time series y_t exhibiting a roughly linear trend could also be smoothed by applying the **Hodrick-Prescott (HP) filter**, i.e., by solving the optimization problem

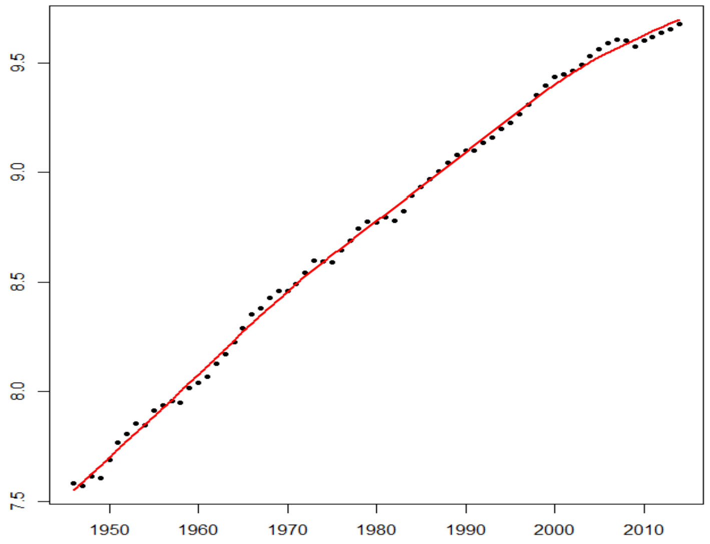
$$\min_{z_t} \{ \sum_{t=1}^{n} (y_t - z_t)^2 + \lambda \sum_{t=3}^{n} \left[(z_t - z_{t-1}) - (z_{t-1} - z_{t-2}) \right]^2 \}.$$

The first term vanishes if $z_t = y_t$ and the second term vanishes if z_t grows linearly, hence there is a trade-off between goodness of fit and smoothness. The larger the parameter λ, the more will deviations from linearity be penalized. For a sufficiently large λ, the solution z_t will virtually coincide with the least squares line.

Exercise: Apply the HP filter with λ=700 to the log GDP.[3]

```
library(mFilter)  # the package "mFilter" is loaded
par(mfrow=c(1,1),mar=c(2,2,1,1))
plot(D,y,pch=20) # pch=20: small solid circle
h <- hpfilter(y.ts,type="lambda",freq=700)
lines(D,h$trend,col="red",lwd=2)
```

[3] It might be helpful to run R as an administrator when you install the package "mFilter".



The tuning parameter λ of the Hodrick-Prescott filter plays a similar role as the parameter k of the MA filter.

However, the HP filter has the advantage that it does not produce missing values at each end of the series.