Stochastic processes I

The ASTA team

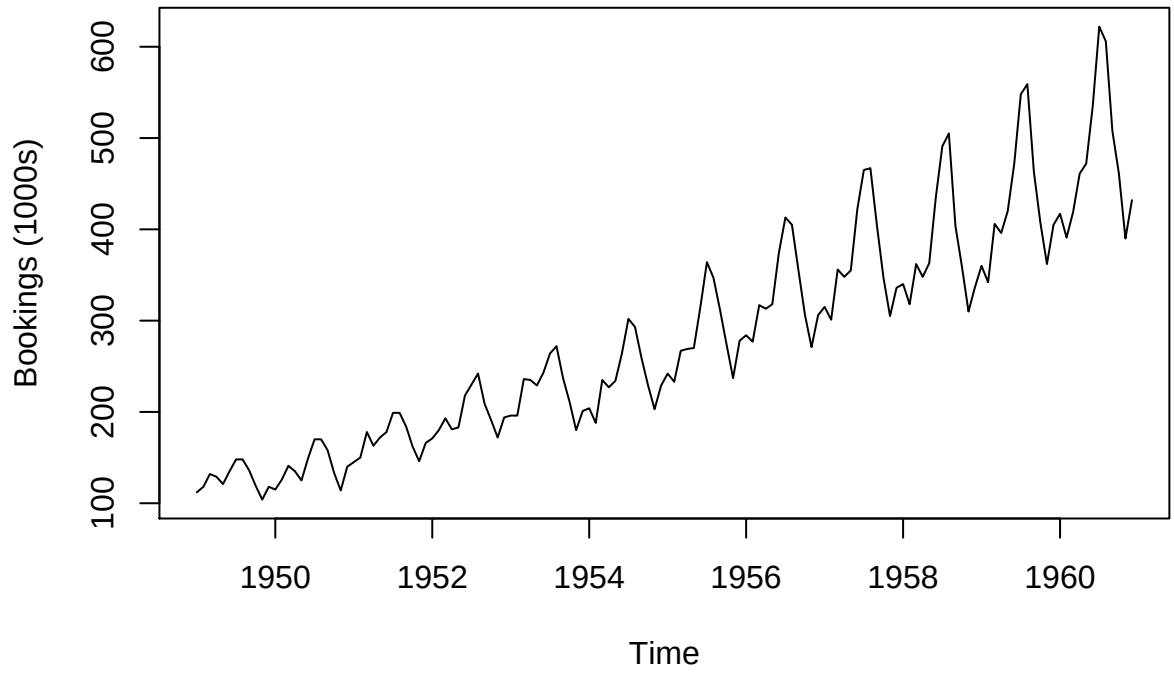
Contents

1 Concepts, terminology and examples

1.1 Plotting, manipulation, trends and seasonality

- A time series is a collection of the same variable measured at different points in time. We will always assume the data is observed at equidistant points in time (i.e. same time difference between consecutive observations).
- To get appropriate summaries and plots of the data we must tell R that the data is a time series, which is called a `ts` object in R. A built-in example is the dataset `AirPassengers` which we will abbreviate to `AP` to save typing:

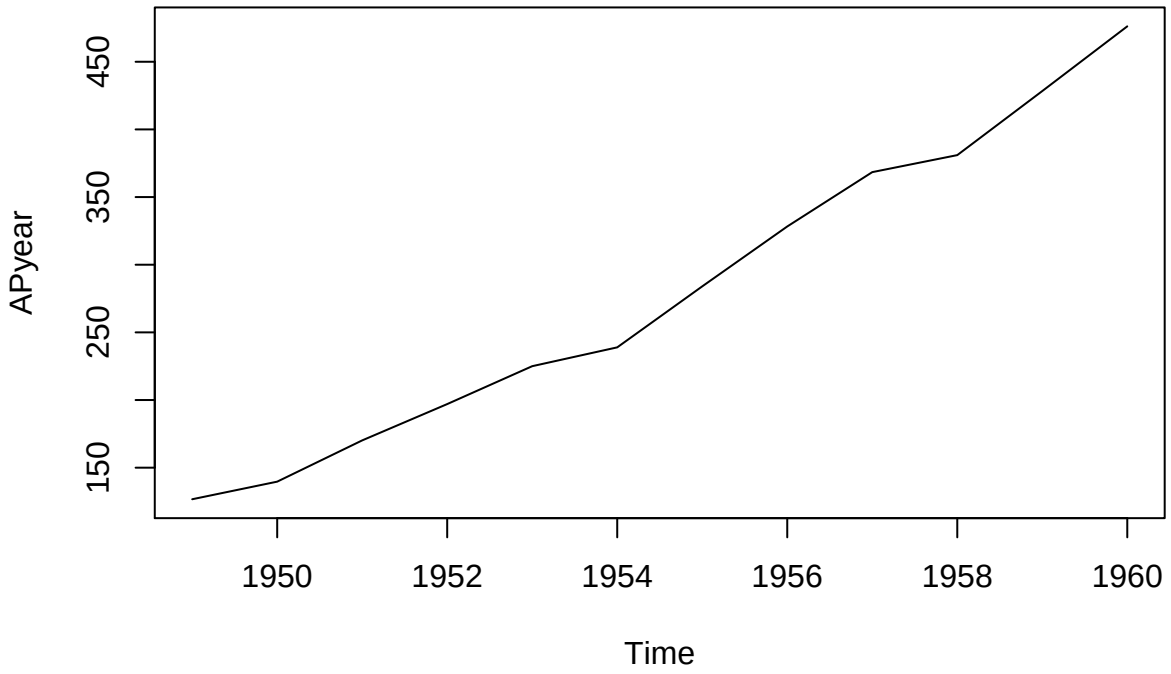
```
AP <- AirPassengers
plot(AP, ylab = "Bookings (1000s)")
```



- The generic function `plot` automatically detected that `AP` was a time series (`ts` object) and plotted the correct time on the horizontal axis (by calling `plot.ts` automatically).

- We can aggregate the time series over each year (sums the values by default, so we add `FUN = mean` to get mean number of monthly bookings for each year), which clearly shows an increasing trend in the number of passenger bookings:

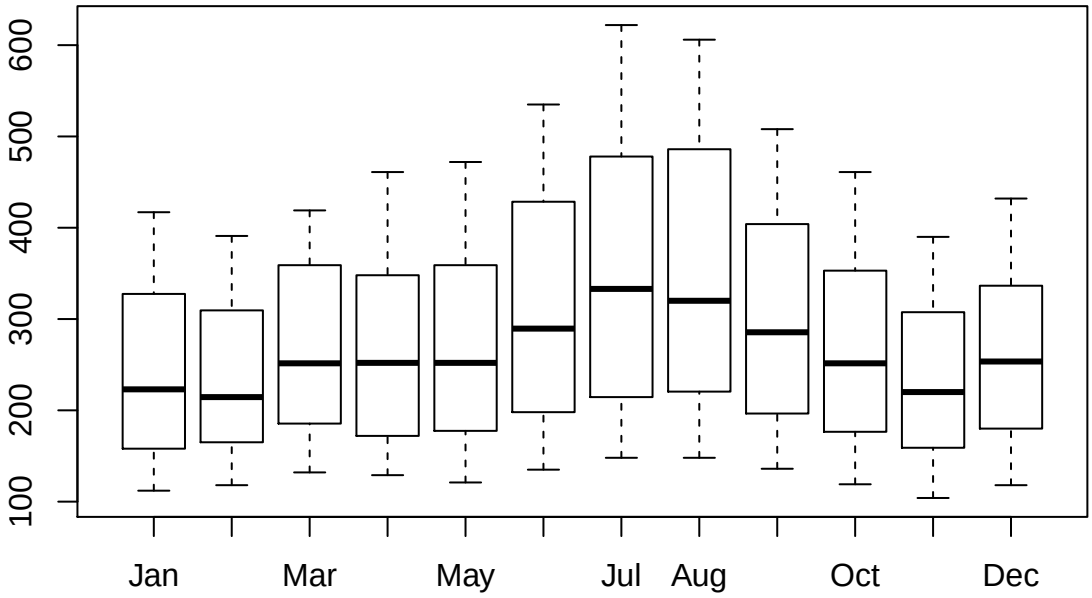
```
APyear <- aggregate(AP, FUN = mean)
plot(APyear)
```



Such a systematic change in a time series that is not periodic is known as the **trend** of the series.

- The repeating (periodic) pattern within each year seen in the original data is known as **seasonality** or **seasonal variation**. The function `cycle` indicates which cycle (aka season) each observation belongs to (in this case 1-12). A boxplot of the number of bookings for each cycle component (month) clearly shows the seasonality:

```
cyc <- cycle(AP)
cyc <- factor(cyc, labels = month.abb)
boxplot(AP ~ cyc)
```



- If we only want to consider a subset of the data the function `window` is used:

```
APJun55Dec56 <- window(AP, start = c(1955, 6), end = c(1956, 12))
APJun55Dec56
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1955                     315 364 347 312 274 237 278
## 1956 284 277 317 313 318 374 413 405 355 306 271 306
```

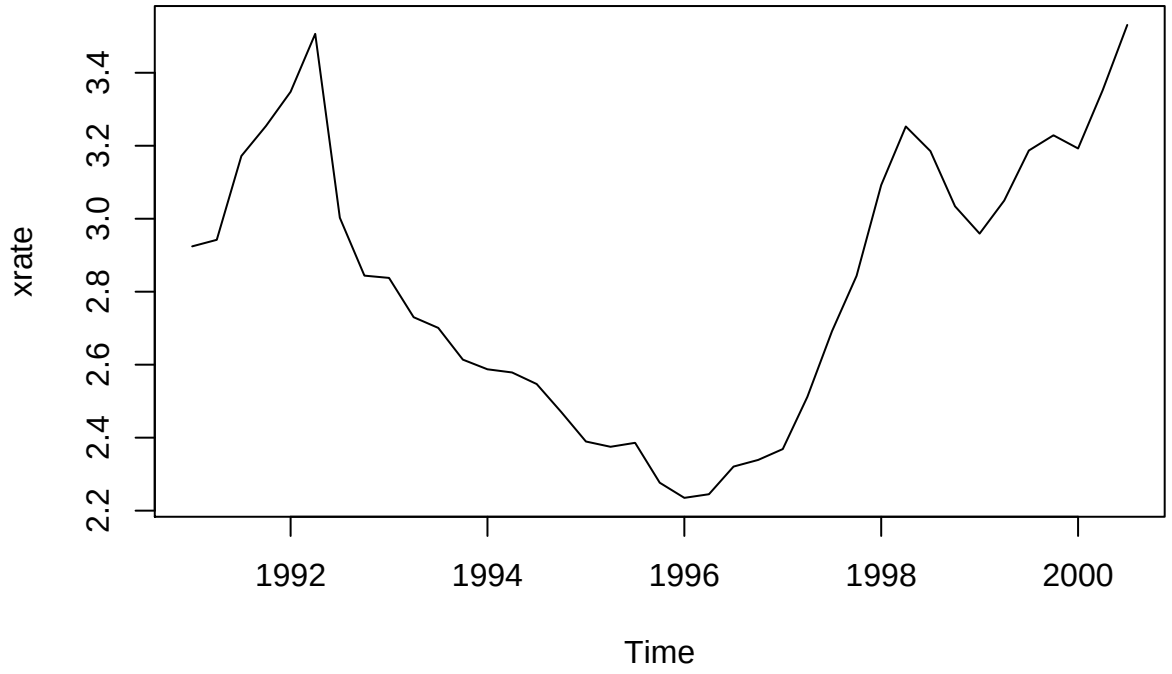
- The window function can also sub-sample the time series according to the frequency:

```
APAug <- window(AP, start = c(1949, 8), end = end(AP), freq = TRUE)
```

1.2 Stochastic trend

- The trend for the passenger bookings data was very clear from the data (as well as the seasonal pattern), and a reasonable explanation may be increasing population, increasing prosperity and technological advances making tickets cheaper.
- In other situations the behaviour of the trend may be less clear and we will need to handle it differently. For the quarterly exchange rate between GBP and NZD the picture is less clear:

```
www <- "https://asta.math.aau.dk/eng/static/datasets?file=pounds_nz.dat"
exchange_data <- read.table(www, header = TRUE)
exchange <- ts(exchange_data, start = 1991, freq = 4)
plot(exchange)
```

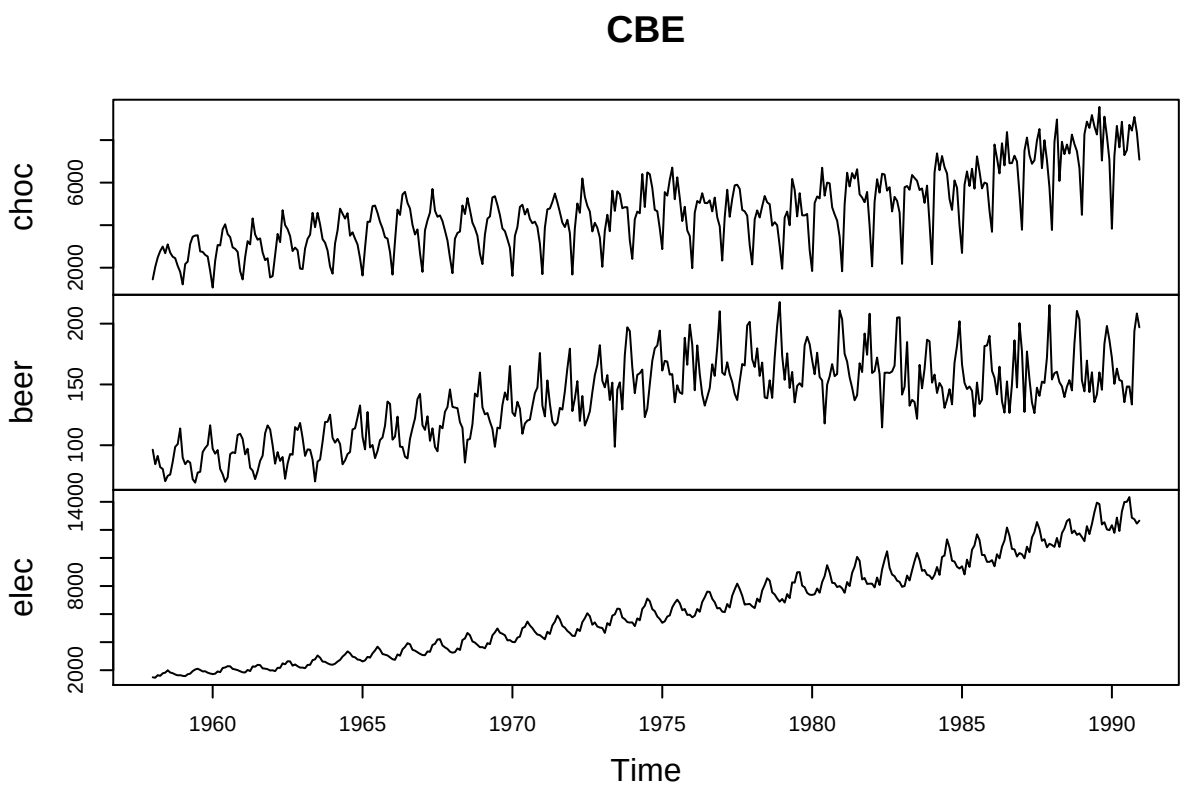


- In general it is dangerous to extrapolate the trend, and this is even more so the case for a series with a stochastic trend, which this example may have.
- Especially for financial time series stochastic trends with abrupt changes to the trend are common.

1.3 Multiple series

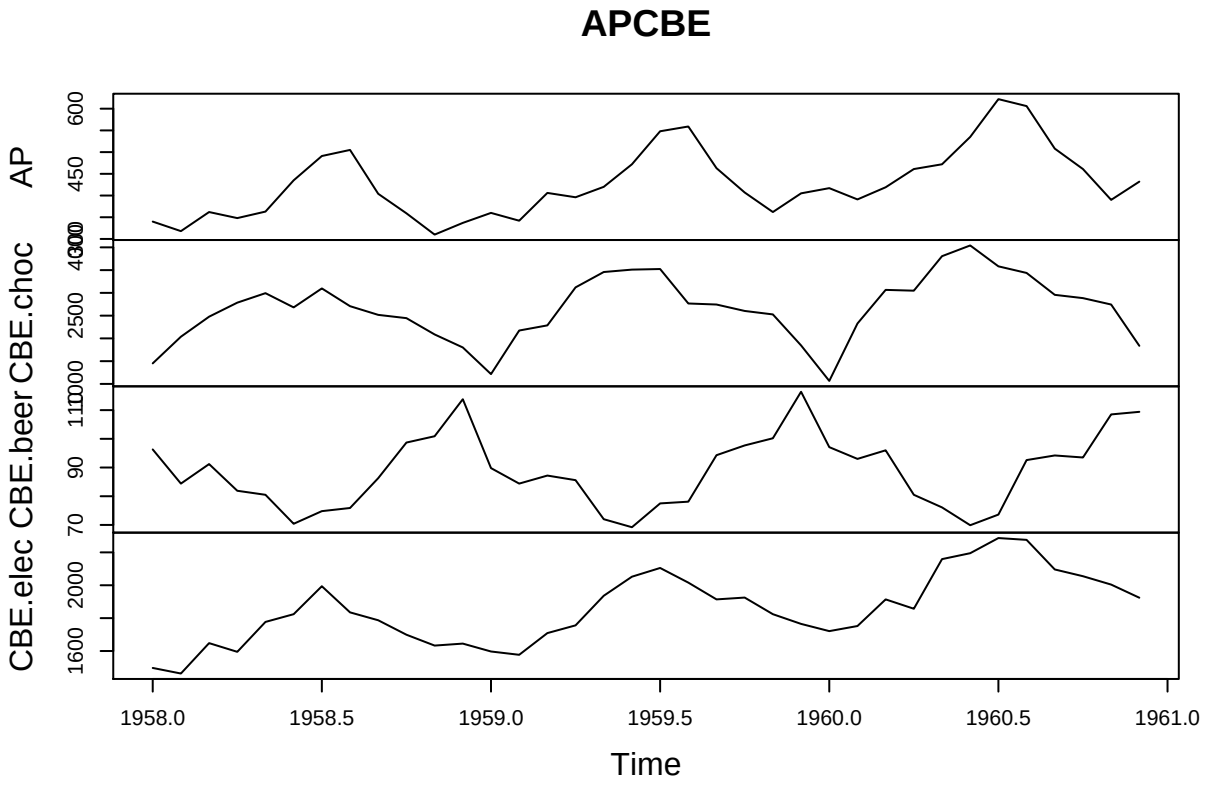
- Monthly time series from Jan. 1958 to Dec. 1990 of supply of three goods in Australia:
  - Electricity (Giga Watt hours)
  - Beer (Mega liters)
  - Chocolate (tonnes)

```
CBEdata <- read.table("https://asta.math.aau.dk/eng/static/datasets?file=cbe.dat",
                      header = TRUE)
CBE <- ts(CBEdata, start = 1958, freq = 12)
plot(CBE)
```



- For overlapping time series **with the same frequency** we can get the data for the overlapping period like this:

```
APCBE <- ts.intersect(AP, CBE)
plot(APCBE)
```



- In the realm of time series people often find spurious correlations and mistake them for causal relationships. E.g. the number of passenger bookings in USA looks very similar to electricity consumption in Australia and there is a strong correlation:

```
cor(APCBE)
```

```
##                  AP   CBE.choc   CBE.beer   CBE.elec
## AP        1.0000000  0.6375808 -0.4434747  0.8841668
## CBE.choc  0.6375808  1.0000000 -0.6204467  0.7644756
## CBE.beer -0.4434747 -0.6204467  1.0000000 -0.3120179
## CBE.elec  0.8841668  0.7644756 -0.3120179  1.0000000
```

However, this does not imply that US air passengers really influence the electricity consumption in Australia (or vice versa)! Time series with similar trends and seasonality will typically show strong correlation even though they are unrelated. Better explanations may be due to similar population growth, seasonal patterns, etc.

1.4 Decomposition

1.4.1 Decomposition - trend term

- It can be useful to decompose the time series x_t into *the trend* m_t, *the seasonal variation* s_t, and *an error term* z_t. Here we will focus on an additive decomposition $x_t = m_t + s_t + z_t$, but multiplicative models $x_t = m_t \cdot s_t \cdot z_t$ are also available.

- To estimate the trend a simple solution is to use a moving average by averaging the preceding and following L values relative to the current point in time, where L is chosen to average out the season effect. So if the season effect is weekly we choose $L = 3$ such that

$$\hat{m}_t = (x_{t-3} + x_{t-2} + x_{t-1} + x_t + x_{t+1} + x_{t+2} + x_{t+3})/7$$

  E.g. if the current time t is a Tuesday the moving average averages today's value with the ones from the preceding Sat., Sun. and Mon. and the following Wed., Thu. and Fri. For an even season length like 12 months we have to do something else if we want an equal amount of past and future in the moving average, and we use half of the value six months in the past/future:

$$\hat{m}_t = \frac{\frac{1}{2}x_{t-6} + x_{t-5} + x_{t-4} + \cdots + x_0 + \cdots + x_{t+4} + x_{t+5} + \frac{1}{2}x_{t+6}}{12}$$

  E.g., the moving average for Jul. 1958 uses half the value in Jan. 1958 and half the value in Jan. 1959 and all the 11 values Feb.-Dec. 1958:

```
choc <- CBE[, "choc"]
i <- 7
(choc[i-6]/2 + sum(choc[(i-5):(i+5)]) + choc[i+6]/2)/12
```

```
## [1] 2413.625
```

This computation can be done for all time points (except the first six and last six) using `decompose`, which returns a list with a component `trend` (and other things to be discussed shortly):

```
choc_decomp <- decompose(choc)
choc_trend <- choc_decomp$trend
window(choc_trend, end = c(1958, 8))
```

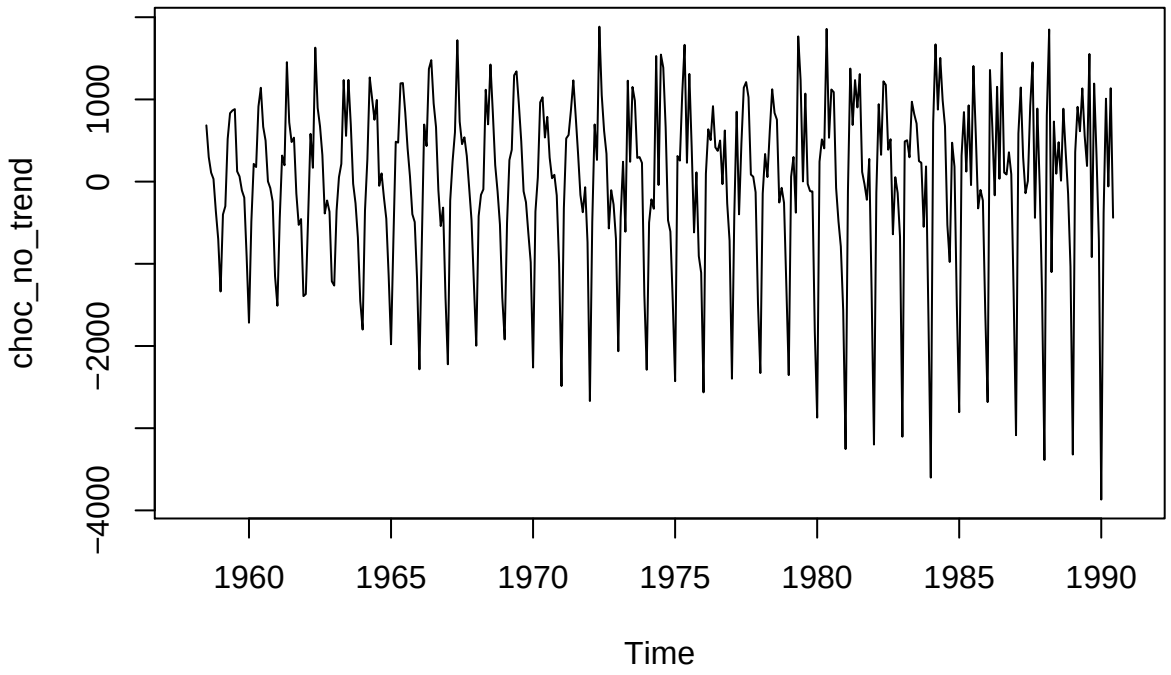
```
##           Jan      Feb      Mar      Apr      May      Jun      Jul
## 1958       NA       NA       NA       NA       NA       NA 2413.625
##           Aug
## 1958 2409.500
```

- Note that the procedure automatically chooses the appropriate size of the moving average based on the frequency of the time series, which was set to 12 in this case.

1.4.2 Decomposition - seasonal term

- To estimate the seasonal effect we subtract the trend and average for each period in the season (e.g. each month):

```
choc_no_trend <- choc - choc_trend
plot(choc_no_trend)
```




```
choc_jan <- mean(window(choc_no_trend, st = c(1958, 1), freq = TRUE), na.rm = TRUE)
choc_jan
```

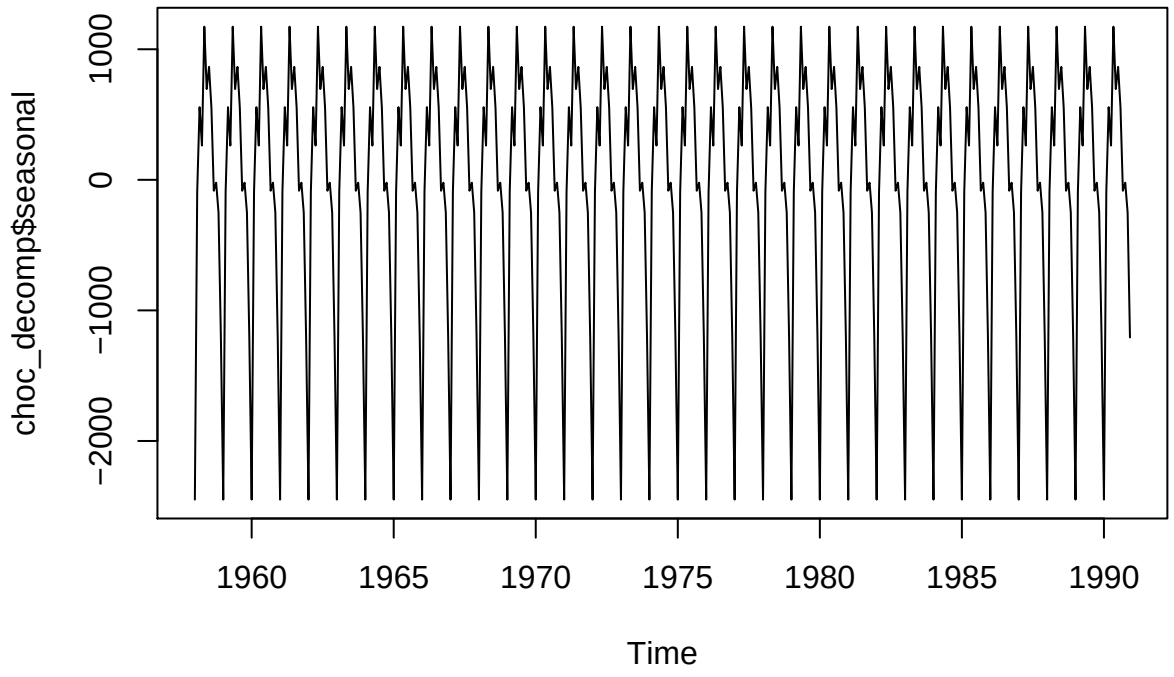
```
## [1] -2450.538
```

```
choc_may <- mean(window(choc_no_trend, st = c(1958, 5), freq = TRUE), na.rm = TRUE)
choc_may
```

```
## [1] 1171.215
```

It appears that chocolate consumption typically is much higher in May than in January. The season effects are also calculated by `decompose` and stored in the list as `seasonal`:

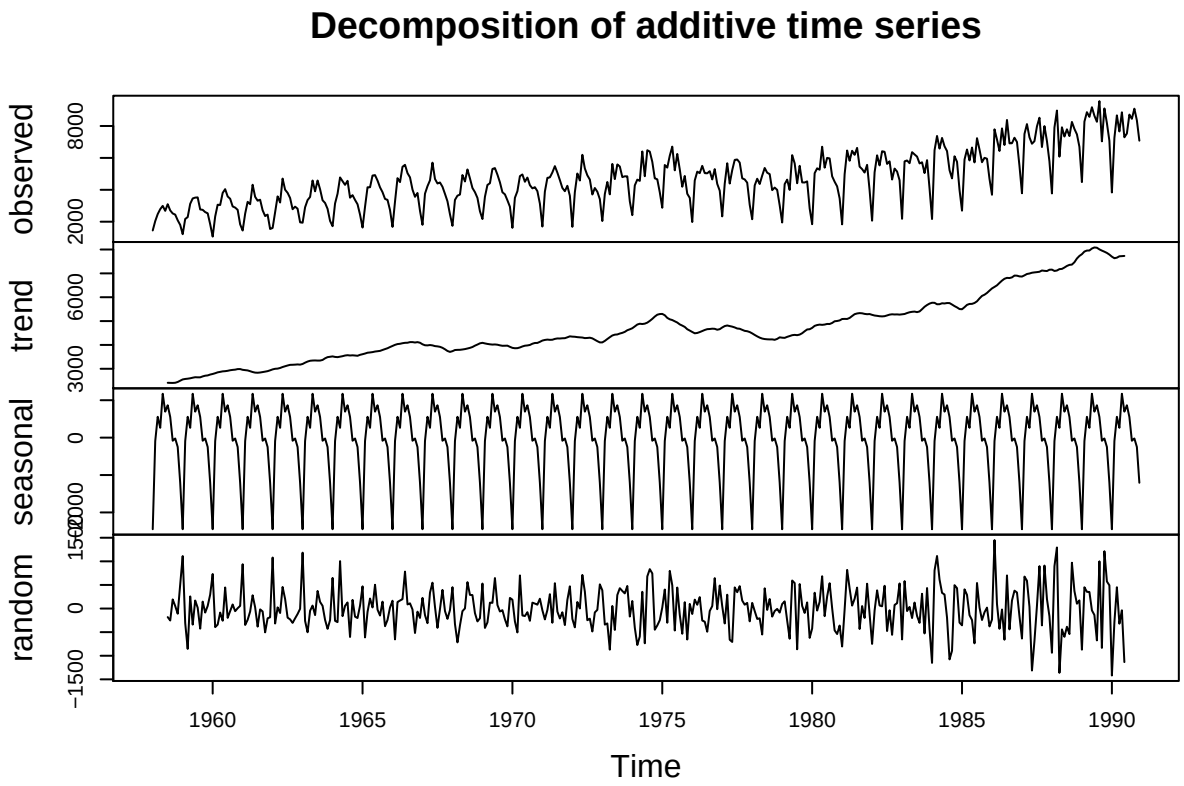
```
plot(choc_decomp$seasonal)
```



1.4.3 Decomposition - random term

- The entire decomposition can also be plotted:

```
plot(choc_decomp)
```

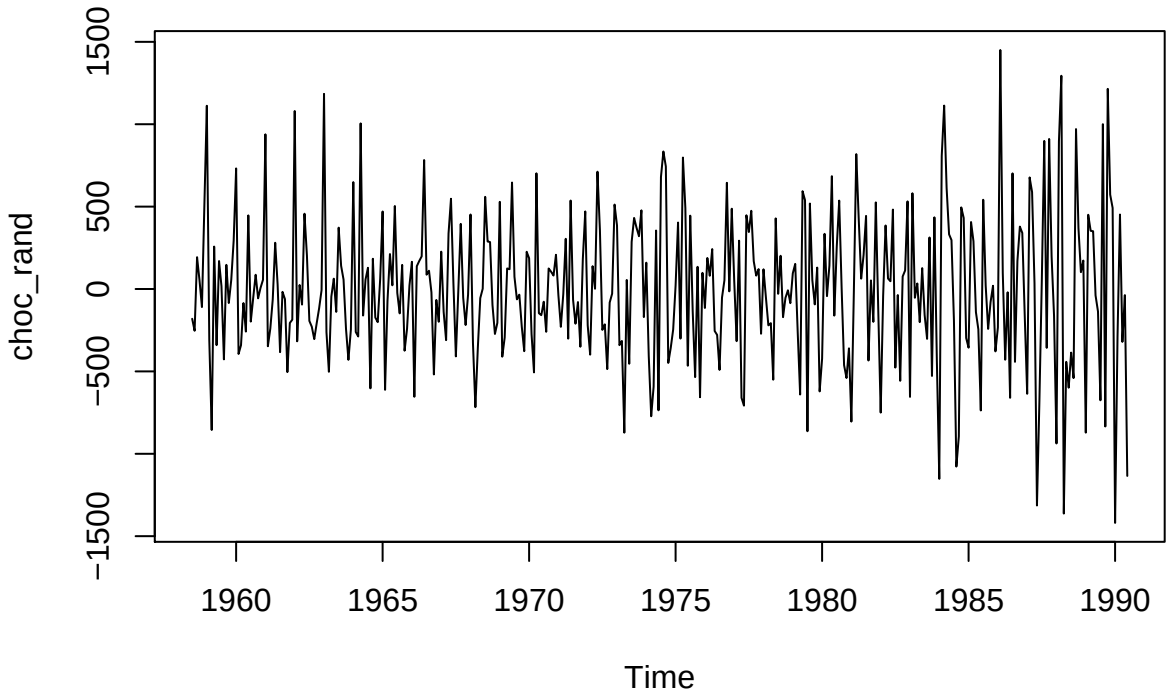


- Here the random term is simply given by $z_t = x_t - m_t - s_t$, and this can be accessed through the `random` entry of the decomposition output `choc_decomp$random`.

2 Stationarity and autocorrelation

- Consider the excess chocolate consumption data after adjusting for trend and season (and omitting the first six and last six entries which are `NA`):

```
choc_rand <- na.omit(choc_decomp$random)
plot(choc_rand)
```



This is a single realization of something we may think of as a random experiment. If things had been different (different weather, different chocolate advertisements, different economic fluctuations, ...) we could imagine another series.

This hypothetical scenario applies to any time series x_t where we imagine our dataset is somehow randomly selected among an entire ensemble of possible time series.

- Calculating the mean of all such hypothetical series would give us the mean function $\mu(t) = E(x_t)$ for all t. Since we already adjusted for trend and season we expect a mean value around zero at each time point, and we might assume $\mu(t) = \mu$ for all t, where μ is a fixed constant (zero in this case). If this is truly the case we say the series is stationary in the mean (first order stationary). In general we can estimate μ by the sample mean as we have been used to:

$$\hat{\mu} = \bar{x} = \frac{1}{n}\sum_{t=1}^{n} x_t.$$

- The variance (square of std. dev.) is in general also a function of time $\sigma^2(t) = E[(x_t - \mu(t))^2]$. If we assume that we also have stationarity of the variance, $\sigma^2(t) = \sigma^2$, we can estimate σ^2 by the sample variance

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

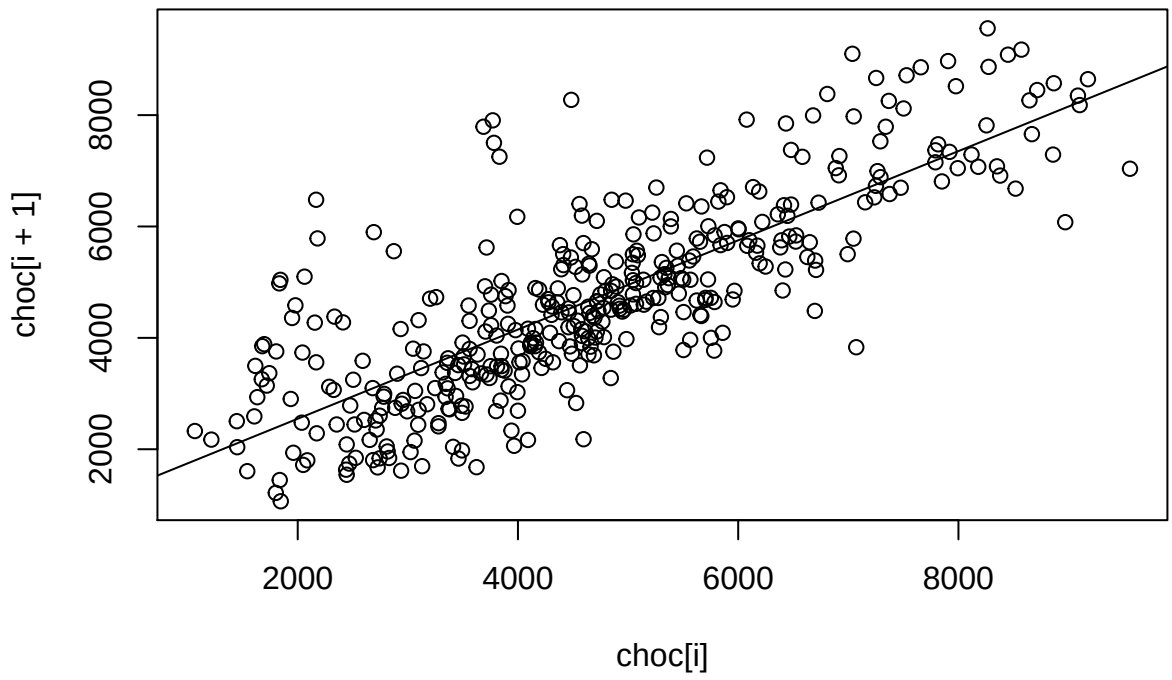
- Now consider a process that is stationary in the mean and variance. The variables at different times may be correlated and the process is called second order stationary if the covariance only depends on the number of time steps between the variables, $Cov(x_t, x_{t+k}) = \gamma(k)$ for all t, and γ is called the autocovariance function. The number of time steps k is called the lag. The autocovariance can be estimated from data:

$$\hat{\gamma}(k) = c_k = \frac{1}{n}\sum_{t=1}^{n-k}(x_t - \bar{x})(x_{t+k} - \bar{x})$$

- The more useful autocorrelation function (acf) is the normalized version that takes values between -1 and +1: $\rho(k) = \gamma(k)/\sigma^2$, which is estimated based on the estimate of the autocovariance above: $\hat{\rho}(k) = r_k = \frac{c_k}{c_0}$.

- The autocorrelation at lag 0 is always equal to 1, $\rho(0) = 1$ and $r_0 = 1$.
- For lag 1 it is the correlation between today's value x_t and tomorrows value x_{t+1}. If we look directly at the chocolate data this correlation is very strong (simply because there is an increasing trend, so a high value is typically followed by a high value):

```
i <- 1:(length(choc)-1)
plot(choc[i], choc[i+1])
abline(lsfit(choc[i], choc[i+1]))
```

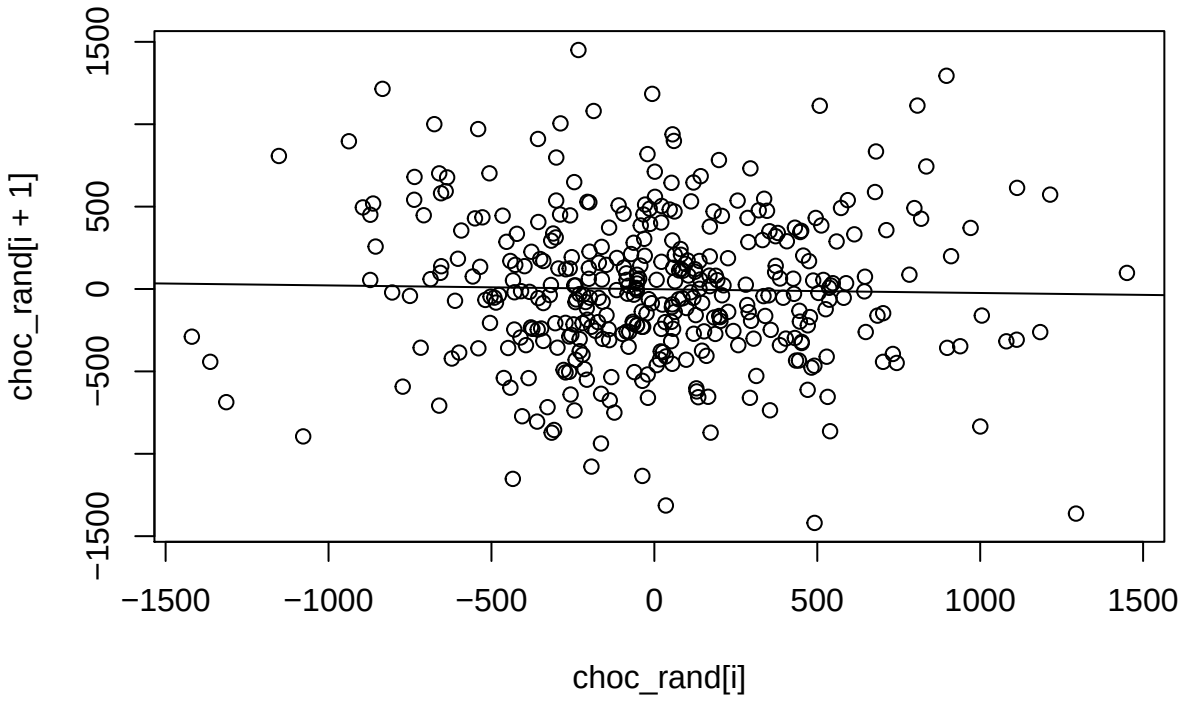



```
cor(choc[i], choc[i+1])
```

```
## [1] 0.8025919
```

- After removing the trend and seasonality there isn't really any dependence on the preceding value:

```
i <- 1:(length(choc_rand)-1)
plot(choc_rand[i], choc_rand[i+1])
abline(lsfit(choc_rand[i], choc_rand[i+1]))
```



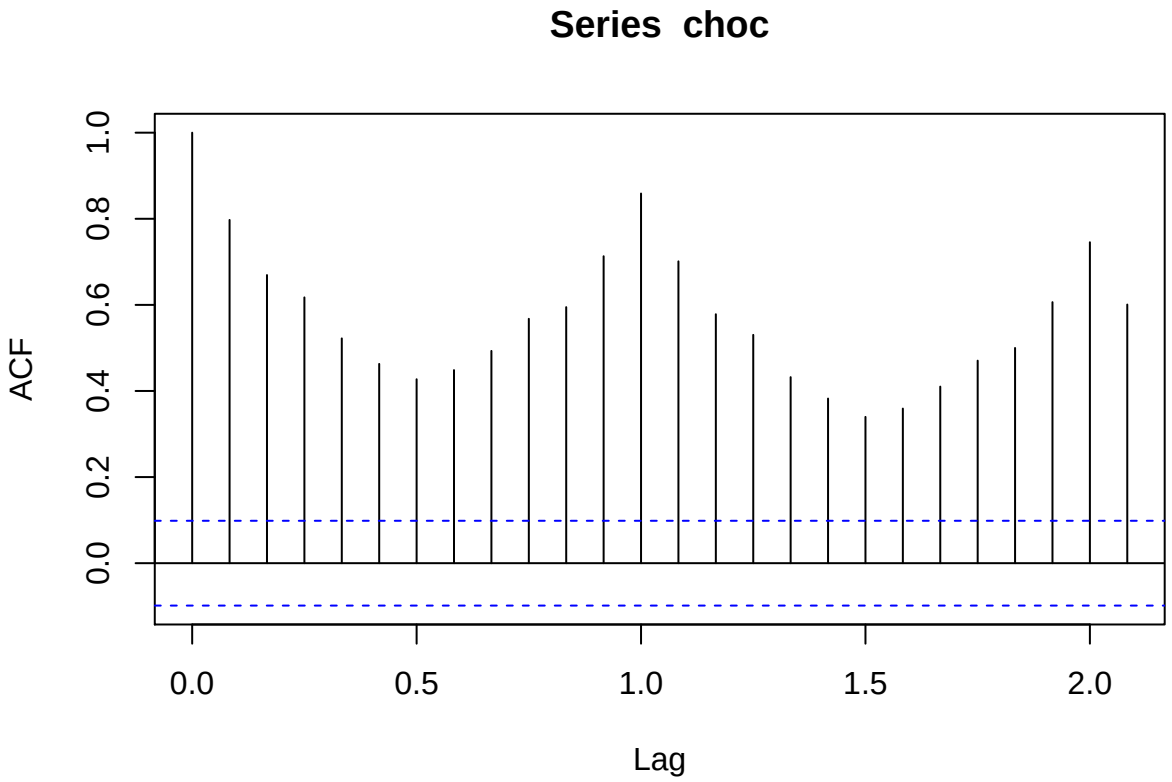

```
cor(choc_rand[i], choc_rand[i+1])
```

```
## [1] -0.02278652
```

2.1 Correlogram (empirical acf)

- To calculate the empirical autocorrelation at many lags we use `acf` which produces a plot by default:

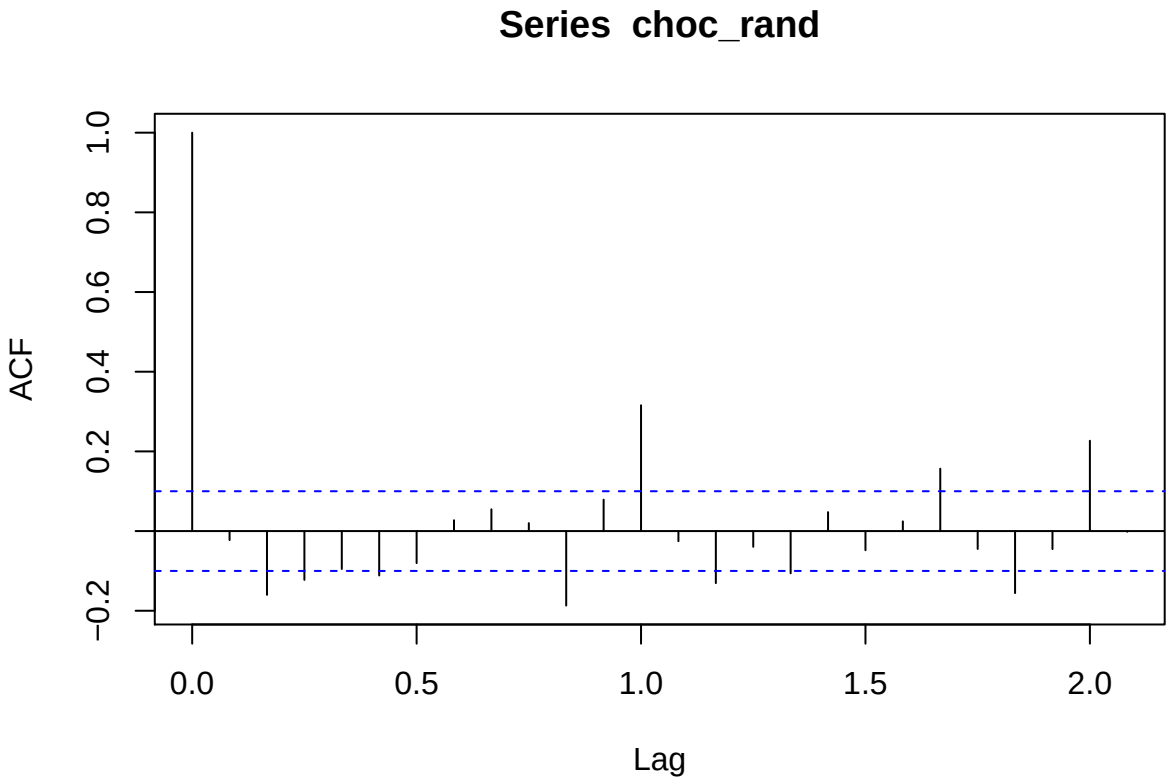
```
choc_acf <- acf(choc)
```



This is called the correlogram.

- It is possible to calculate the correlogram for any time series (also non-stationary). However, the theoretical properties only make sense for a stationary process.
- The correlogram value at lag k, r_k is an estimate of the true autocorrelation $\rho(k)$.
- In particular if $\rho(k) = 0$ then r_k will usually not be exactly zero. Rather it is approximately normally distributed with a mean close to zero and standard deviation $1/\sqrt{n}$, where n is the length of the series. Based on this approximate significance bands are drawn at $\pm 1.96/\sqrt{n}$.
- The significance bands are only valid for a single lag! Even if $\rho(k) = 0$ for all lags $k = 1, 2, \ldots$ we expect $1/20$ of the r_k's to fall outside the line. Furthermore the estimates are heavily correlated so if one falls outside it is also more likely that the one next does.
- It is typical to look at a few specific lags. E.g. lag 1 for the immediate past and lag 12 if a season of length 12 is expected. A spike at lag 12 indicates that there is a seasonal pattern that has not been adequately accounted for. It appears the seasonal adjustment for the chocolate data hasn't been perfect since there is a somewhat large autocorrelation at lag 12 (1 year):

```
acf(choc_rand)
```

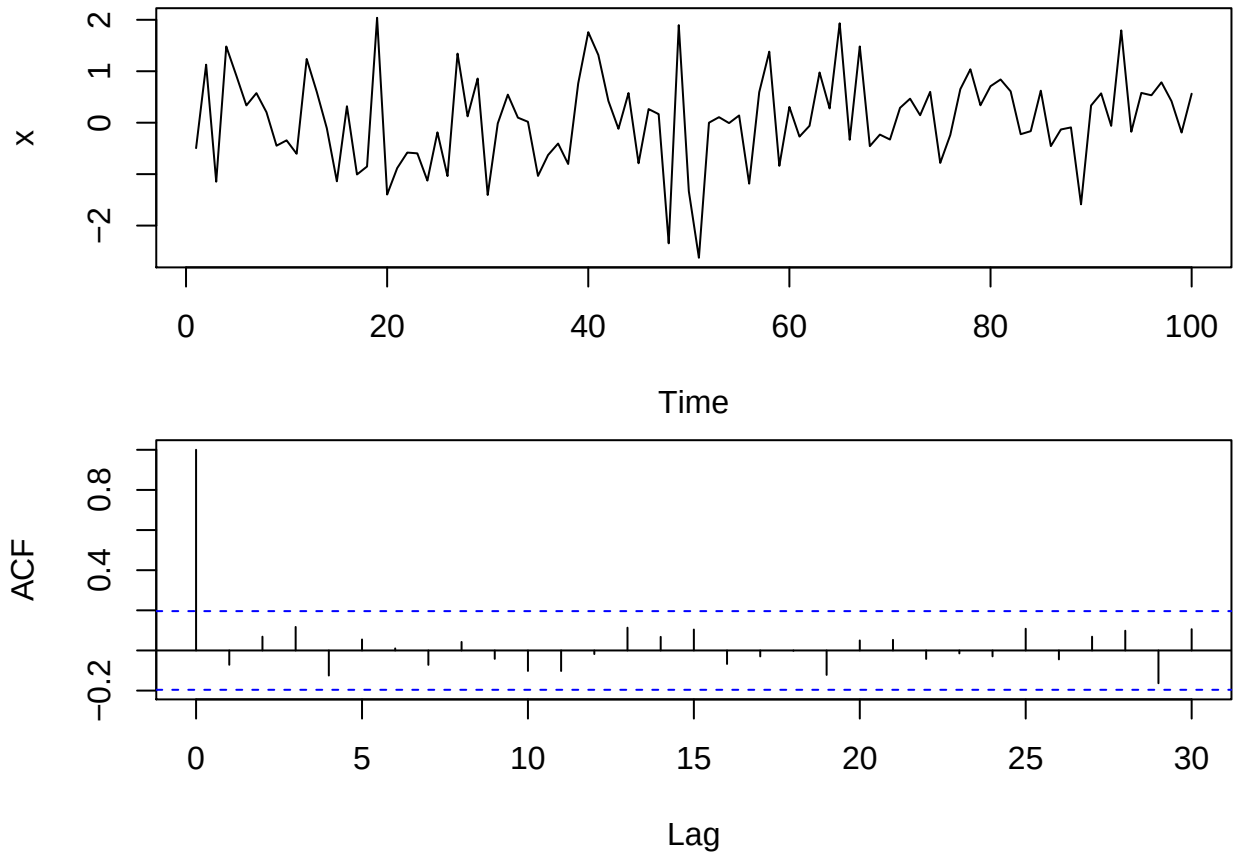


2.2 Typical patterns in correlograms

It can be illustrative to consider a few specific examples of correlograms:

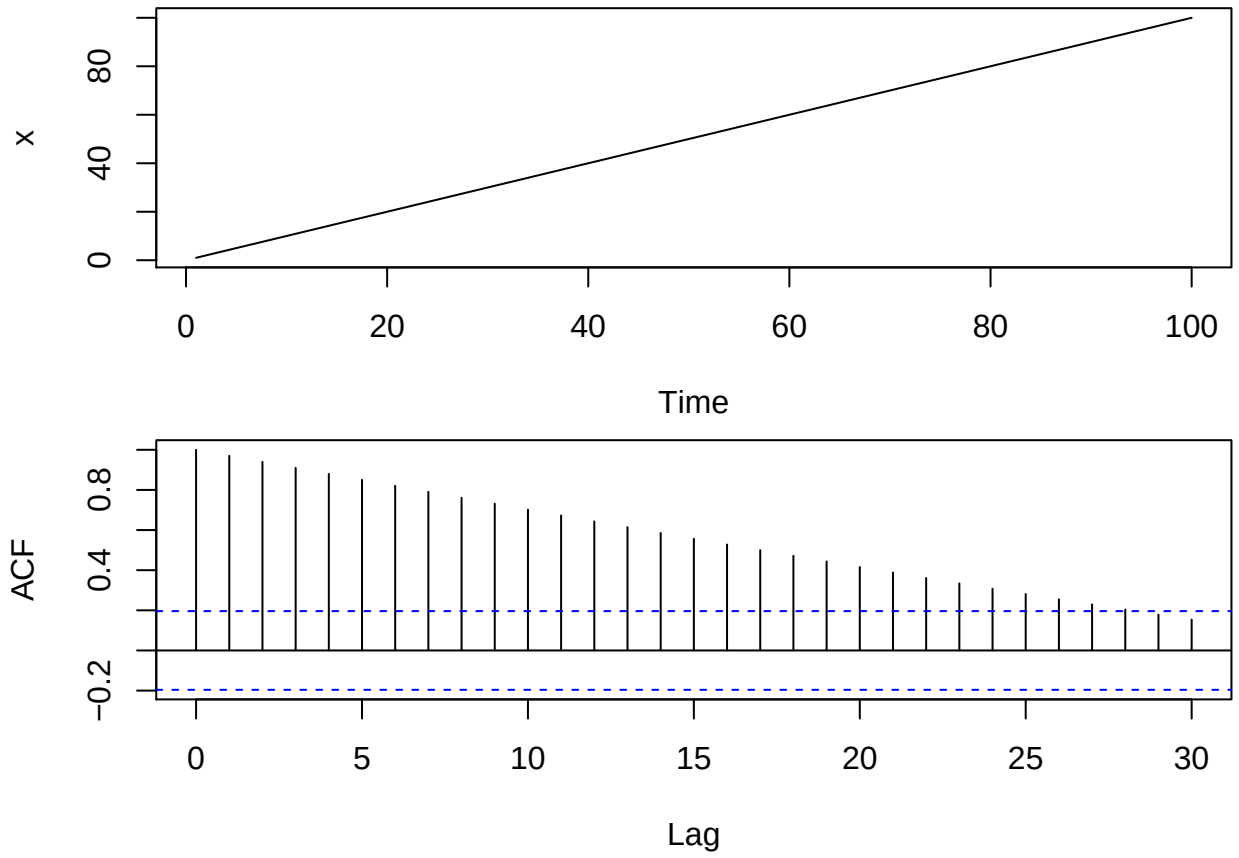
- Completely random:

```
x <- rnorm(100)
par(mfrow = c(2,1), mar = c(4,4,0.5,0.5))
ts.plot(x)
acf(x, main = "", lag.max = 30)
```



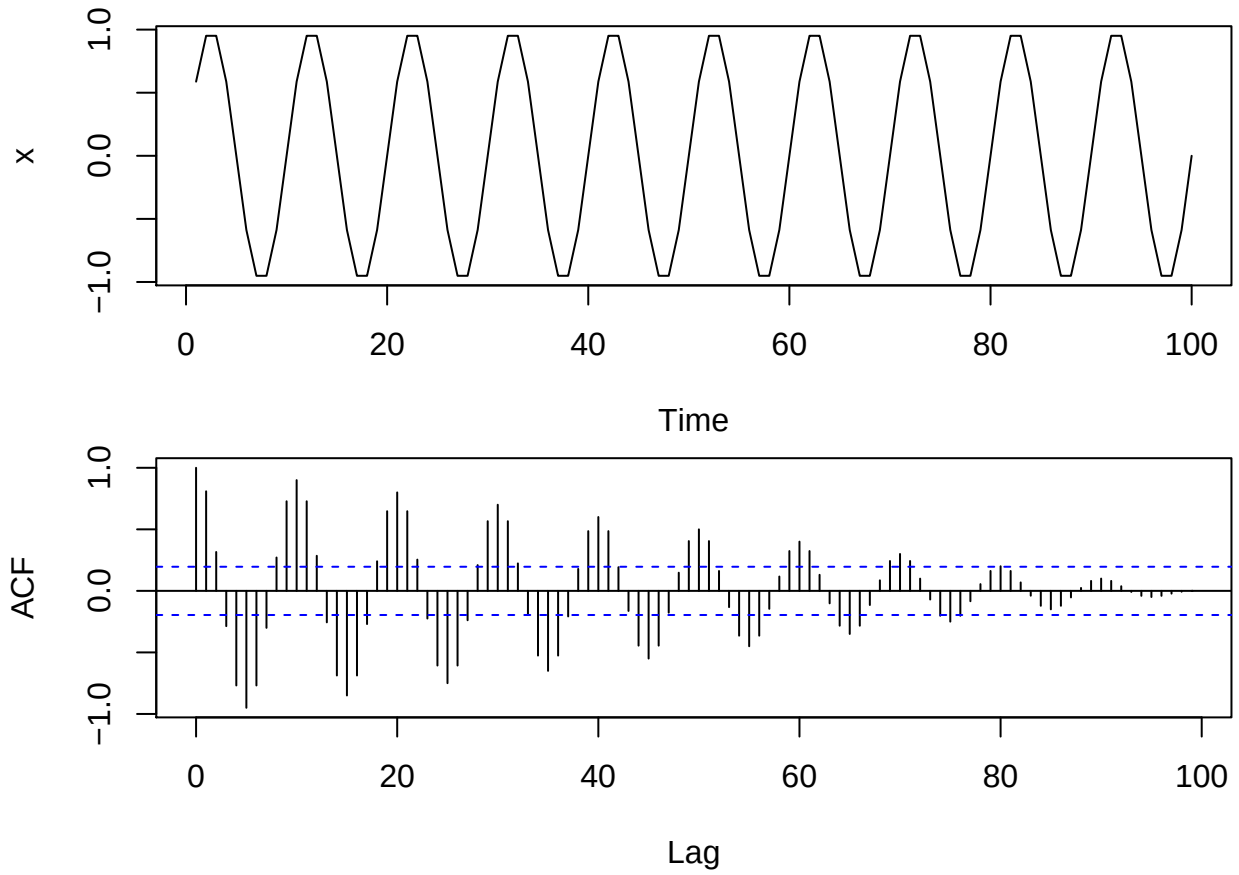
- Trend only:

```
x <- 1:100
par(mfrow = c(2,1), mar = c(4,4,0.5,0.5))
ts.plot(x)
acf(x, main = "", lag.max = 30)
```



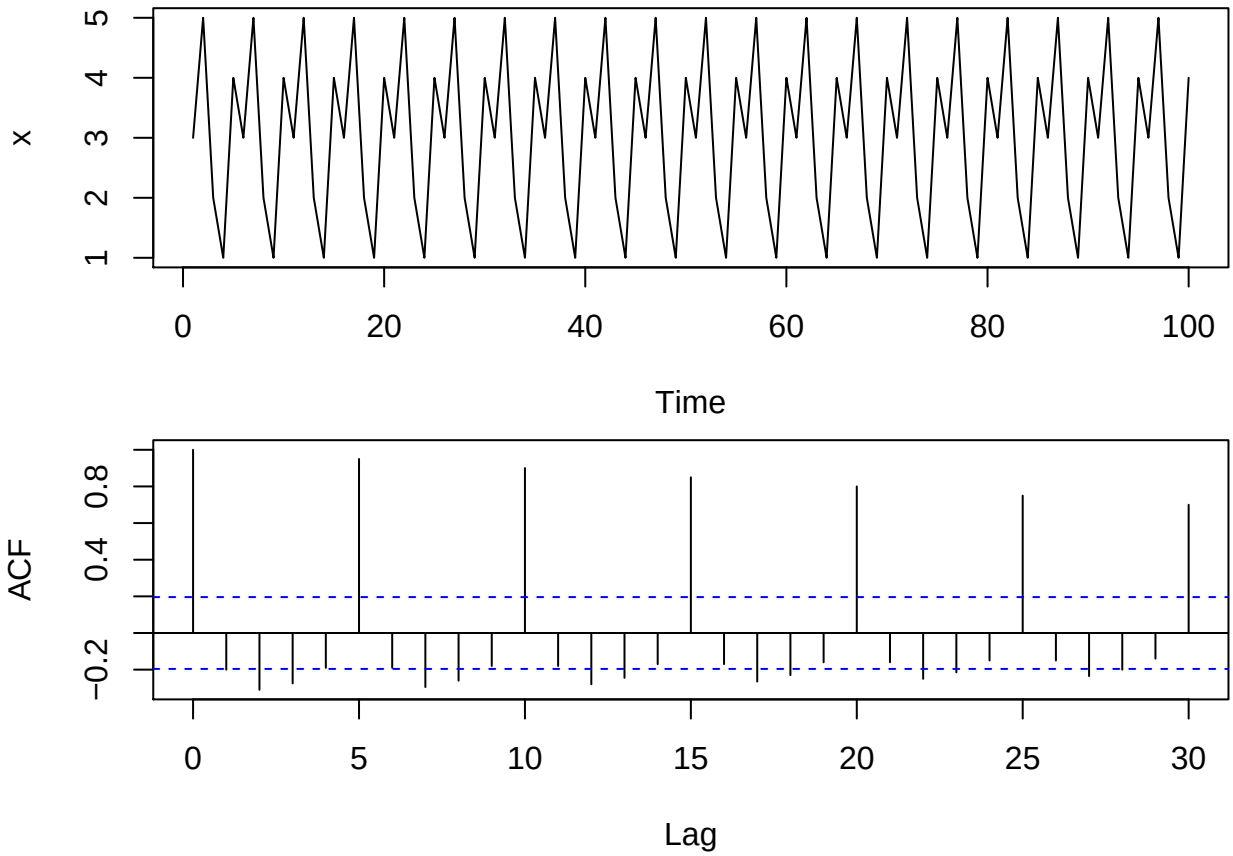
- Sinusoidal wave:

```
x <- sin((1:100)*2*pi/10)
par(mfrow = c(2,1), mar = c(4,4,0.5,0.5))
ts.plot(x)
acf(x, main = "", lag.max = 100)
```



- Repeated pattern:

```
x0 <- c(3,5,2,1,4)
x <- rep(x0, 20)
par(mfrow = c(2,1), mar = c(4,4,0.5,0.5))
ts.plot(x)
acf(x, main = "", lag.max = 30)
```



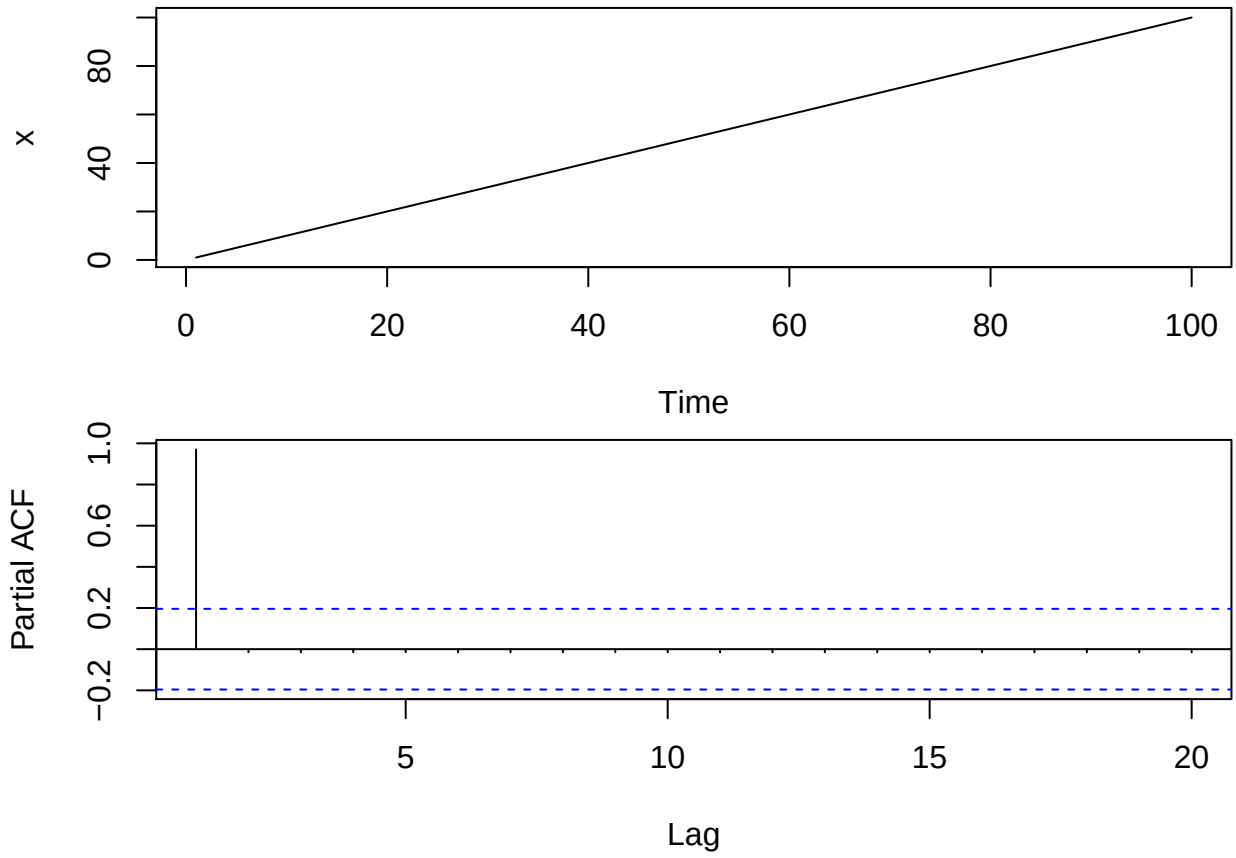
2.3 Partial autocorrelation function

- If a time series has a strong correlation with the values lagged by one time step it most likely also has a strong correlation with the values lagged by two time steps: When today's value is highly influenced by yesterdays, then yesterdays will be highly influenced by the day before, and thus the lag 2 value influences the current value through the lag 1 value.
- The partial autocorrelation (pacf) at lag 2 is the correlation between the time series and the lag 2 values **after controlling for the lag 1 values**.
- This is analogous to the multiple regression setup

$$x_t = \beta_0 + \beta_1 x_{t-1} + \beta_2 x_{t-2} + \epsilon$$

  where β_2 is a partial effect of the corresponding predictor (x_{t-2}) **when controlling for the other predictor** (x_{t-1}).
- In general the pacf at lag k is the correlation of x_t and x_{t-k} after controlling for the intermediate variables $x_{t-1}, \ldots, x_{t-k-1}$.
- A more detailed description of partial correlation in the multiple regression setup is in Section 11.7 of Agresti (2013).
- For a strong deterministic trend the pacf only shows strong correlation in the first lag and after controlling for this the following lags show no extra correlation:

```r
x <- 1:100
par(mfrow = c(2,1), mar = c(4,4,0.5,0.5))
ts.plot(x)
pacf(x, main = "")
```



3 Models for time series data

There are two fundamentally different model classes for time series data.

- Discrete time stochastic processes
- Continuous time stochastic processes

Next lecture will focus on discrete time models and in the remainder we will touch upon continuous time models.

3.1 Continuous time stochastic processes

In this setup we see the underlying X_t as a continuous function of t for t in some interval $[0, T]$.

In principle we imagine that there are infinitely many data points, simply because there are infinitely many time points between 0 and T. Of course this is never true: In practice we will always only have finitely many data points.

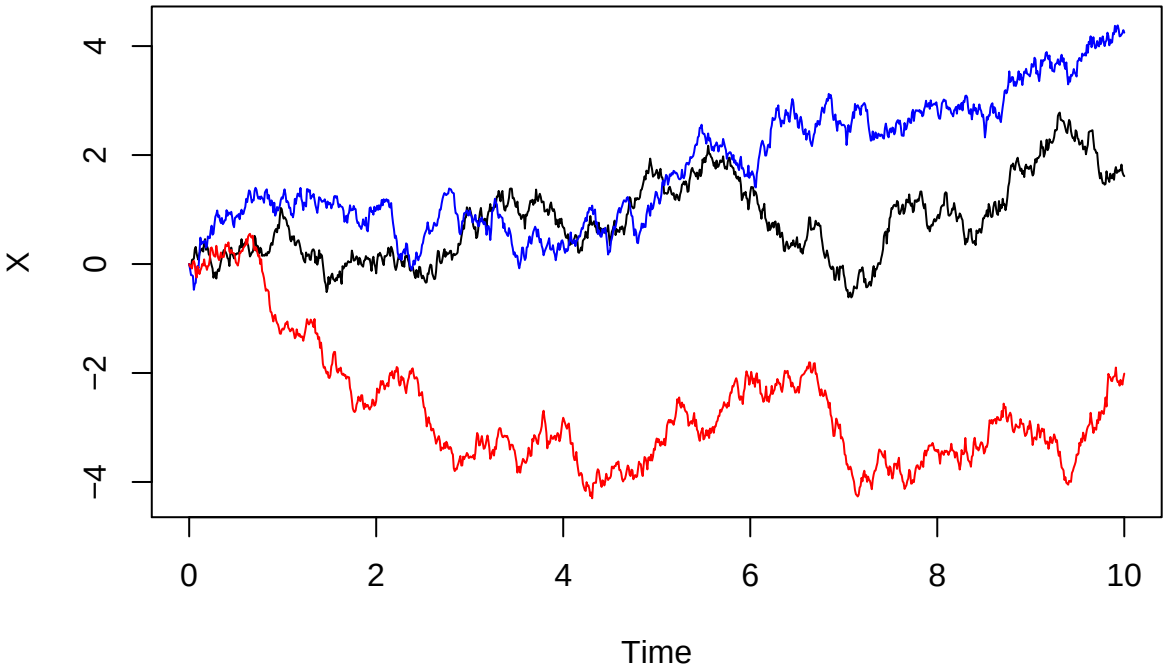
But it makes sense to believe that the real data actually contains all the data points. We are just not able to measure them (and to store them in a computer).

With a model for all datapoints, we are - through simulation - able to describe the behaviour of data. Also between the observations.

3.2 Wiener process

A key example of a process in continuous time will be the so–called Wiener process.

Three simulated realizations (black, blue and red) of this process can be seen here



A Wiener process has the following properties:

- It starts in 0: $W_0 = 0$.
- It has independent increments: For $0 < s < t$ it holds that $W_t - W_s$ is independent of everything that has happened up to time s, that is W_u for all $u \leq s$.
- It has normally distributed increments: For $0 < s < t$ it holds that the increment $W_t - W_s$ is normally distributed with variance $t - s$:

$$W_t - W_s \sim \text{Normal}(\mu = 0, \sigma^2 = t - s).$$

The intuition of this process is that it somehow changes direction all the time: How the process changes after time s will be independent of what has happened before time s. So whether the process should increase or decrease after s will not be affected by how much it was increasing or decreasing before.

This gives the very bumpy behaviour over time.

3.3 Stochastic differential equations

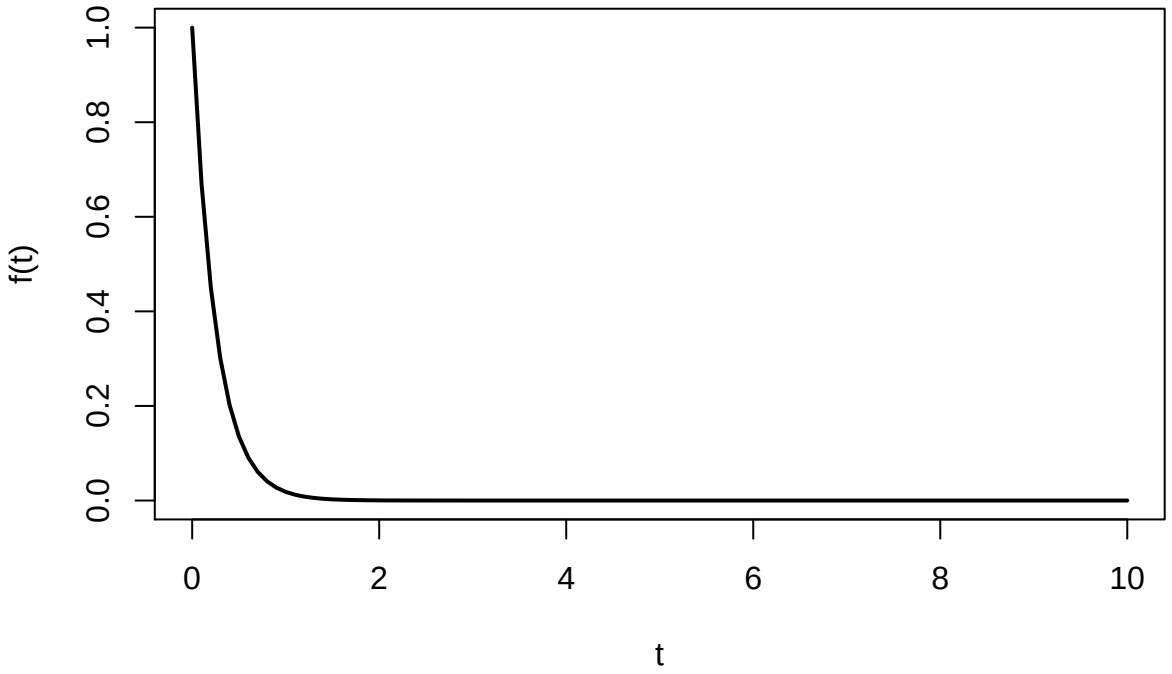
A common way to define a continuous time stochastic process model is through a stochastic differential equation (SDE) which we will turn to shortly, but before doing so we will recall some basic things about ordinary differential equations.

Suppose f is an unknown differentiable function and $f(0) = 1$. Recall the mathematical description of a differential equation

$$\frac{df(t)}{dt} = -4f(t)$$

With the condition $f(0) = 1$ this equation has the solution

$$f(t) = \exp(-4t)$$



With a slightly unusual notation we can rewrite this as

$$df(t) = -4 \cdot f(t)dt$$

This equation has the following (hopefully intuitive) interpretation:

- When time increases by a small amount dt the value of f changes (approximately) by $-4f(t)\,dt$.

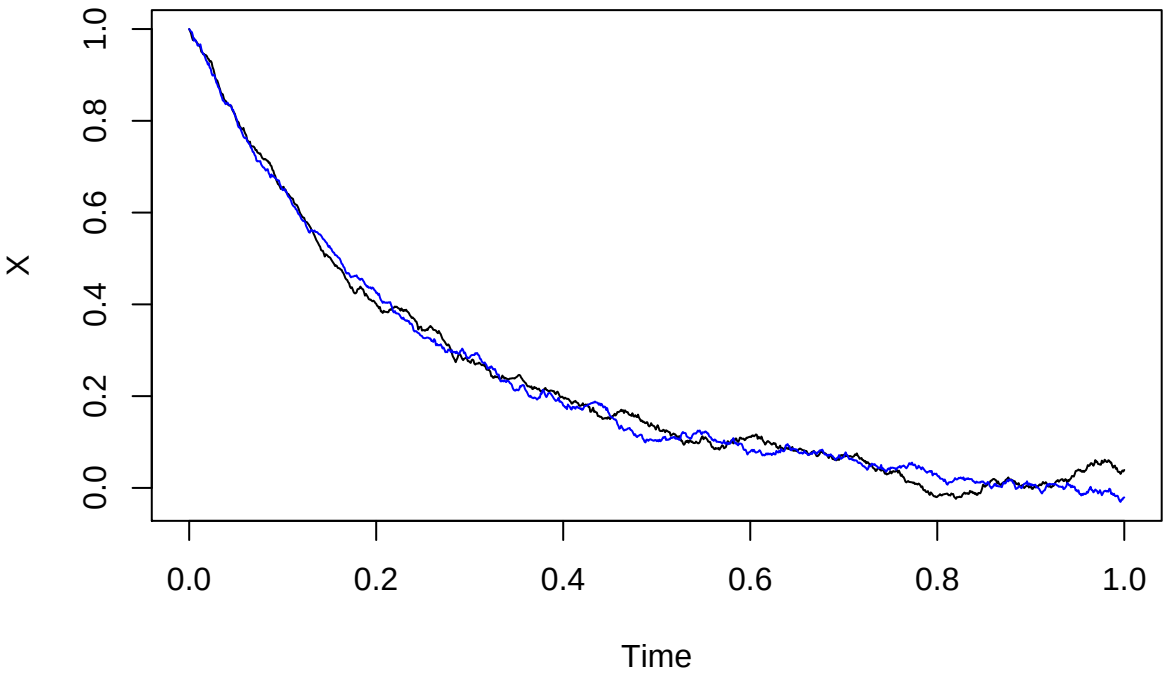
So when t is increased, then f is decreased. And the decrease is determined by the value of $f(t)$. That is why f decreases slower and slower, when t is increased.

We say that the function has a **drift** towards zero, and this drift is determined by the value of the function.

3.4 Stochastic differential equations

It will probably never be true that data behaves exactly like the exponentially decreasing curve on the previous slide.

Instead we will consider a model, where some random noise from a Wiener process has been added. Two different (black/blue) simulated realizations can be seen below



The type of process that is simulated above is described formally by the equation

$$dX_t = -4X_t dt + 0.1 dW_t$$

This is called a **Stochastic Differential Equation** (SDE), and the processes simulated above are called solutions of the stochastic differential equation.

The SDE $dX_t = -4X_t dt + 0.1 dW_t$ has two terms:

- $-4X_t dt$ is the **drift term**.
- $0.1 dW_t$ is the **diffusion term**.

The intuition behind this notation is very similar to the intuition in the equation $df(t) = -4 \cdot f(t)\ dt$ for an ordinary differential equation. When the time is increased by the small amount dt, then the process X_t is increased by $-4X_t\, dt$ AND by how much the process $0.1W_t$ has increased on the time interval $[t, t + dt]$.

So this process has a **drift** towards zero, but it is also pushed in a random direction (either up or down) by the Wiener process (more precisely, the process $0.1W_t$)