USD-EURO exchange rate

- Load the data and have a look at it
- But the paths look different: look at higher dimensional distributions!
- All bi-dimensional distributions
- Correlations
- Partial correlations

Load the data and have a look at it

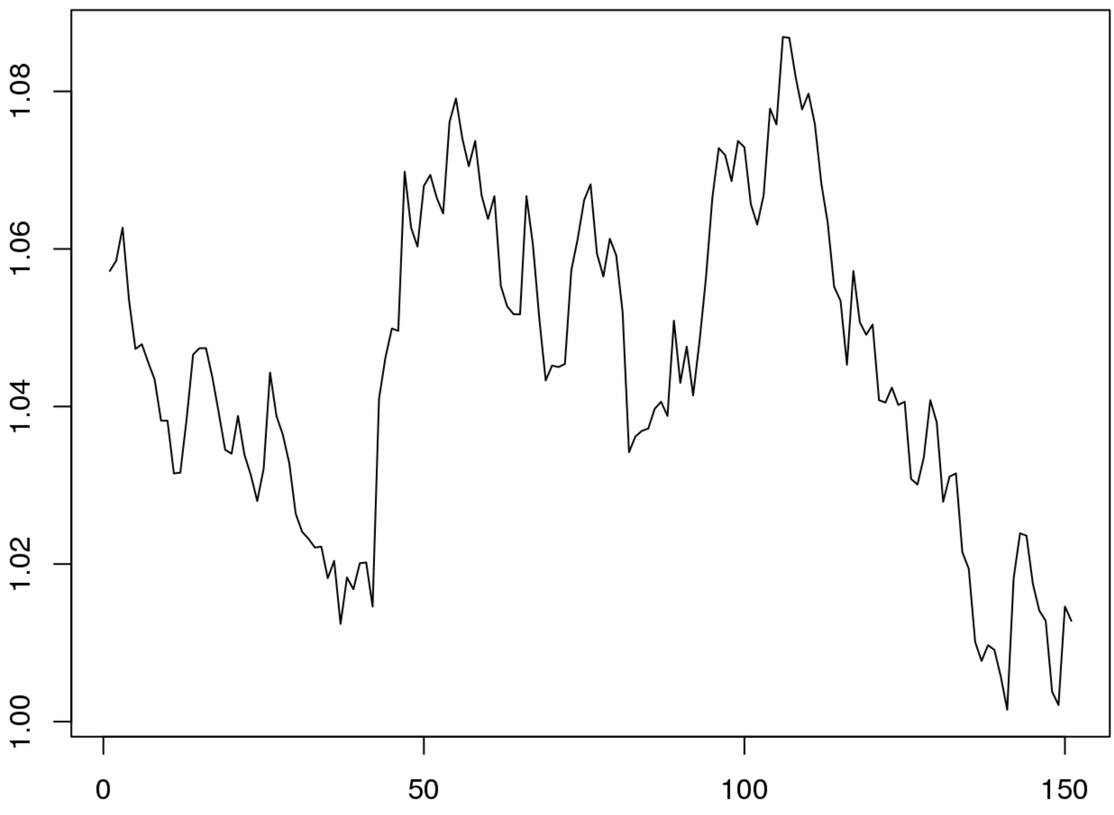
Download the csv file (requires Internet access).

```
cerdata <- url('https://mdi:343@perso.telecom-paristech.fr/roueff/data/usd-euro.csv')
cer <- read.table(cerdata,header=TRUE,sep=";")
attach(cer)
```

We work on a sub Window of length 150.

```
debut <- length(val)+1-100
fin <- length(val)+1-250
valsub <- val[debut:fin]

op <- par(mar=c(3,3,1,2))
ts.plot(valsub,xlab='',ylab='')
```

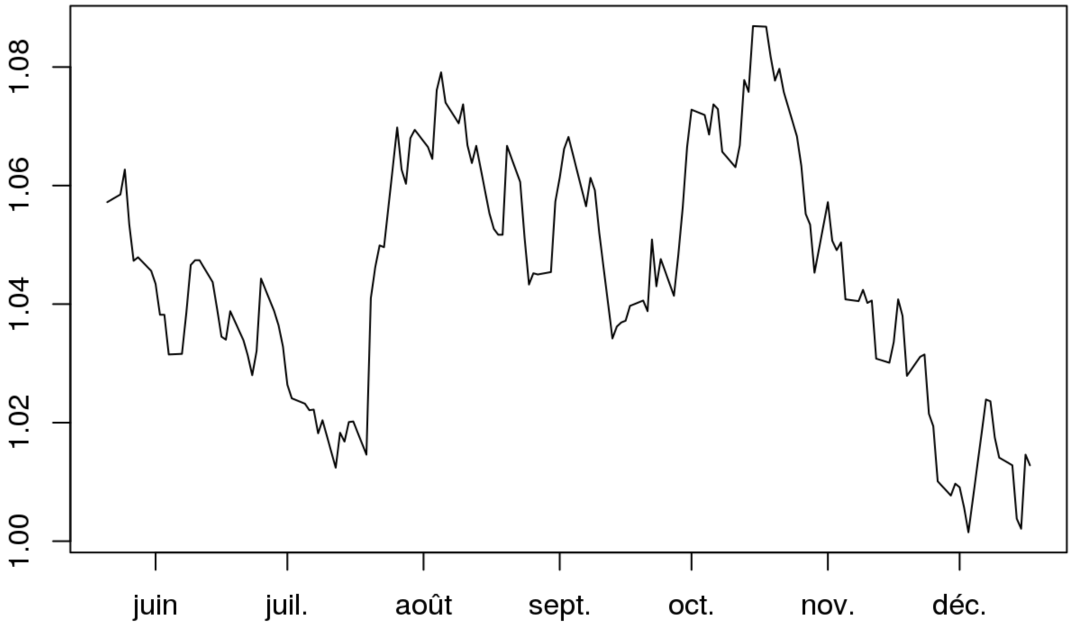


Plot with dates:

```
print(paste('dates between',as.POSIXct(date[debut]),'and',as.POSIXct(date[fin]),'included',sep=
' '))
```

```
## [1] "dates between 1999-05-21 and 1999-12-17 included"
```

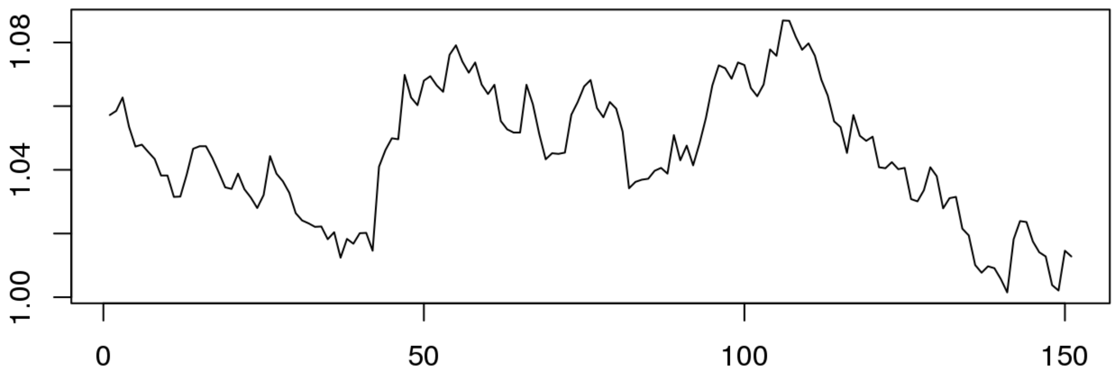
```
plot(as.POSIXct(date[debut:fin]),val[debut:fin], type='l',xlab='',ylab='')
```

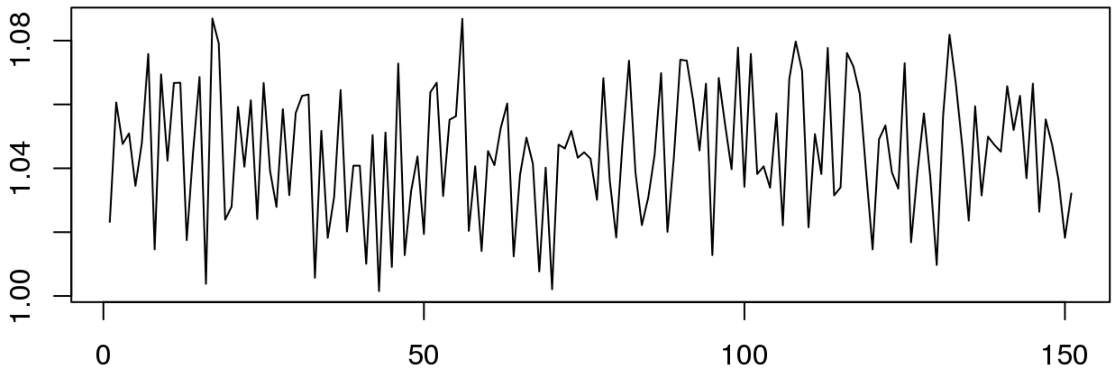


Let us compare the **paths** of this time setries with with the time series obtained by randomly schufling of dates.

```
valsub_shuffled <- sample(valsub, length(valsub) , replace = FALSE)

op <- par(mar=c(3,3,1,2),mfrow=c(2,1))
ts.plot(valsub,xlab='',ylab='')
ts.plot(valsub_shuffled,xlab='',ylab='')
```

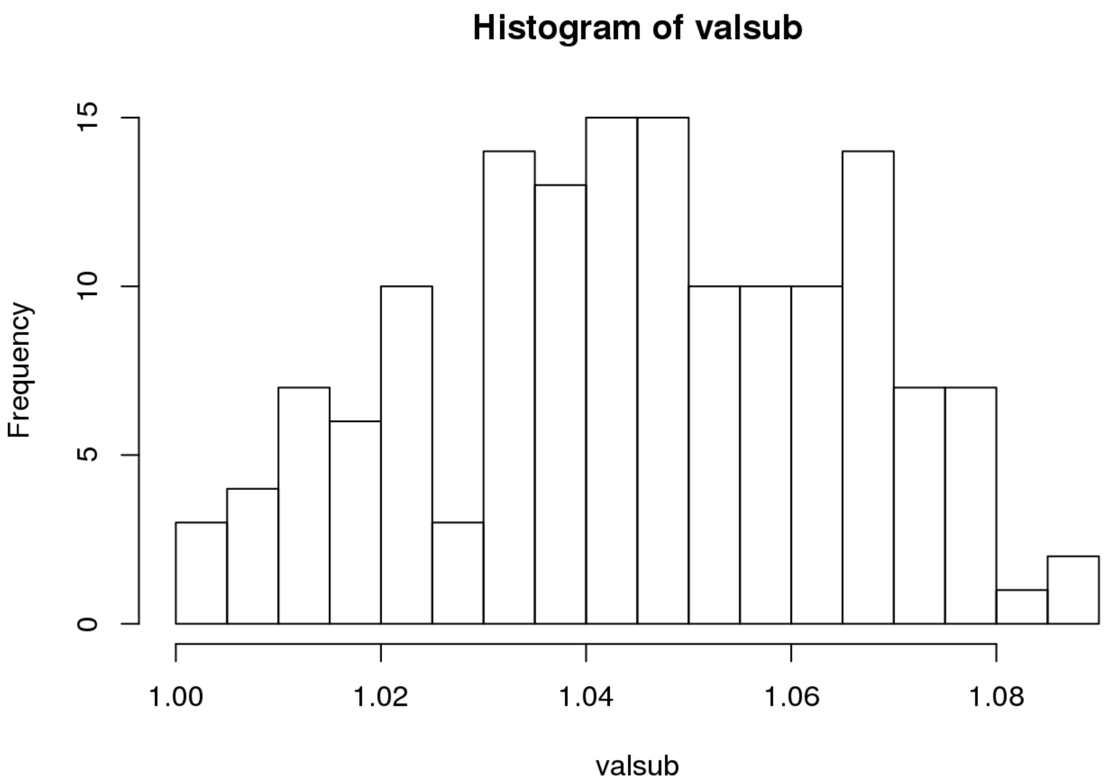





```
par(op)
```

Of course they have the same marginal histogram:

```
hist(valsub,30)
```



Here we assumed that the values at all time instants are drawn according to the **same distribution** and that a finite sample provides a **good representation** of it.

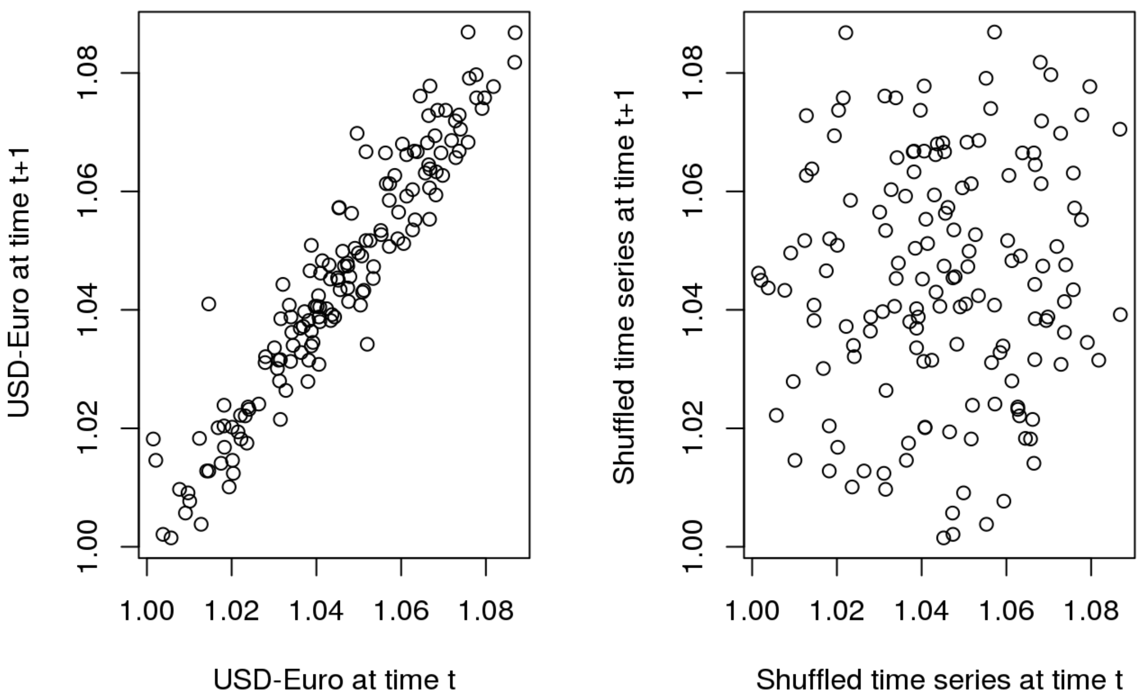
This is related to the **stationary** and **ergodic** properties of the data!

But the paths look different: look at higher dimensional distributions!

Do they have the same **two-dimensional distribution** for (X_t, X_{t+1}) ?

(still assuming the data is **stationary** and **ergodic**…)

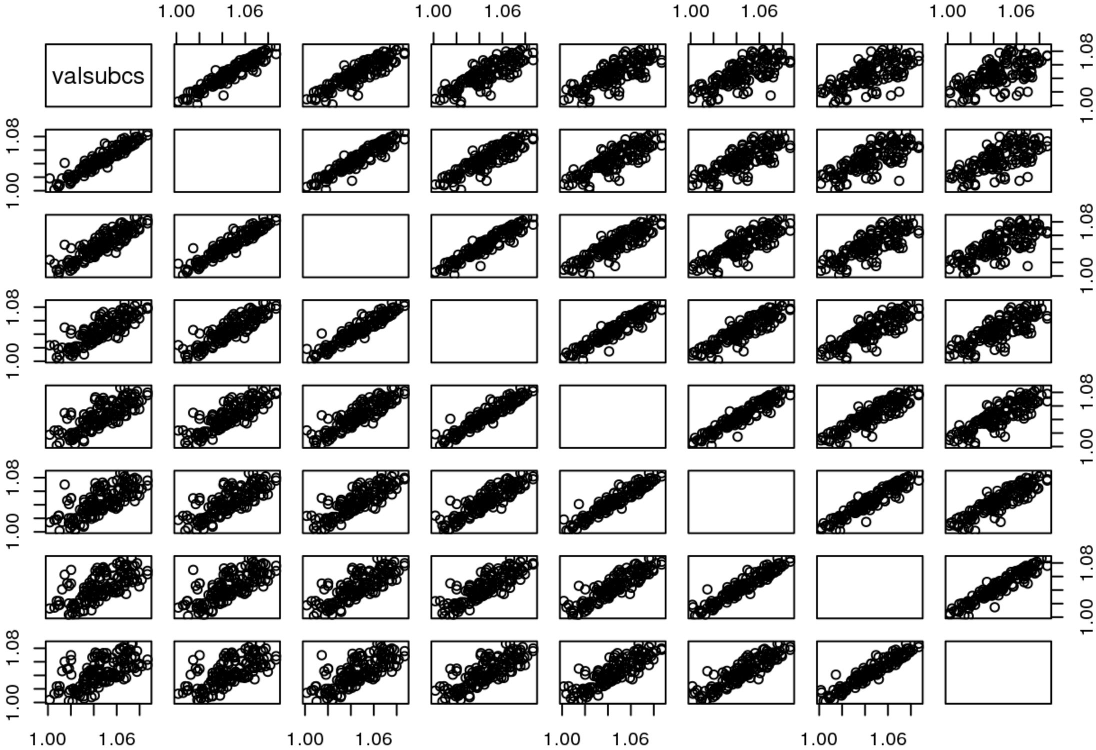
```
T <- length(valsub)
op <- par(mfrow=c(1,2))
valsubc <- t(rbind(valsub[1:T-1],valsub[2:T]))
plot(valsubc, xlab='USD-Euro at time t',
     ylab=' USD-Euro at time t+1')
valsubc_shuffled <- t(rbind(valsub_shuffled[1:T-1],valsub[2:T]))
plot(valsubc_shuffled, xlab='Shuffled time series at time t',
     ylab='Shuffled time series at time t+1')
```



All bi-dimensional distributions

We only looked at (X_t, X_{t+1}). What about all pair-wise distributions $(X_t, X_{t+h}), h = 1, 2, 3, \ldots$?

```
n <- 8
valsubcs <- valsub[1:(T-(n-1))]
for (i in 1:(n-1)){
  valsubcs <- rbind(valsubcs,valsub[(1+i):(T+i-(n-1))])
}
pairs(t(valsubcs))
```

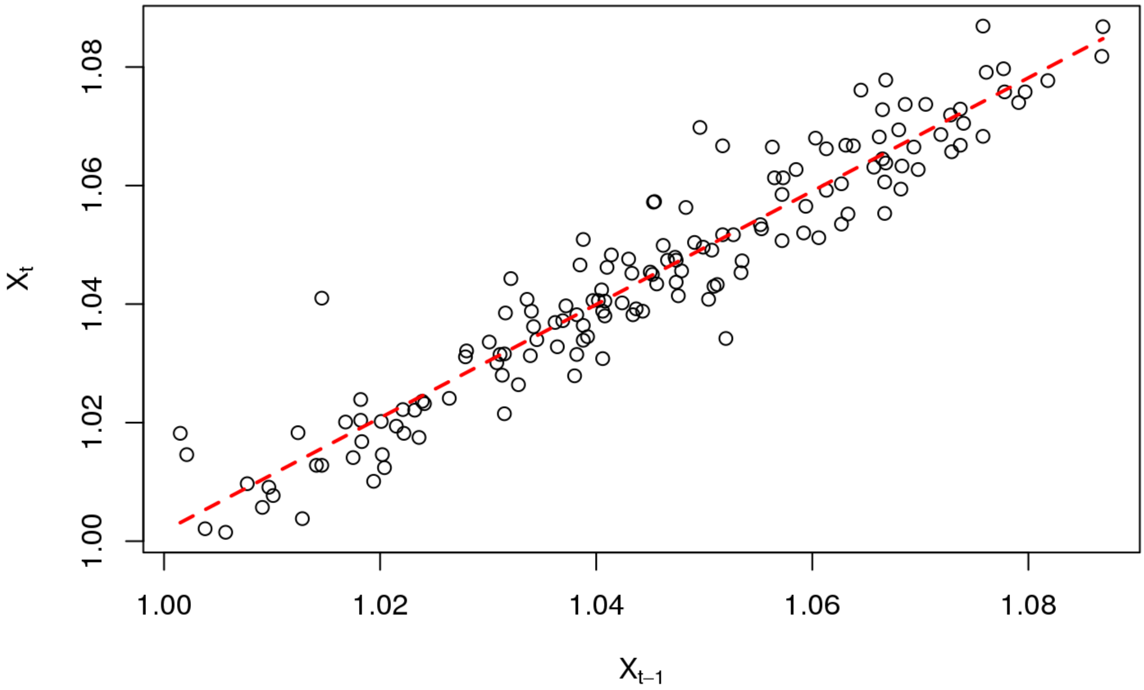


Can we do something simpler ?

Correlations

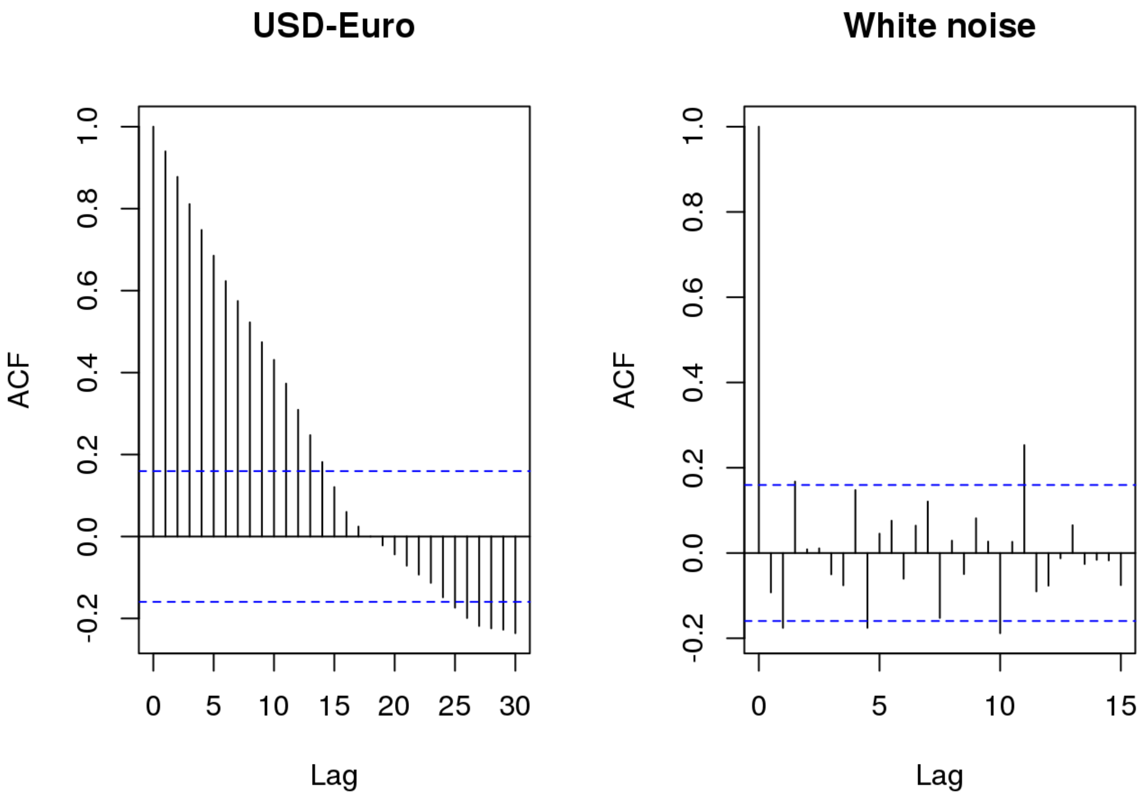
Let’s get back to the two-dimensional distribution of (X_t, X_{t+1}) and just add the regression line.

```
lmar <- lsfit(valsubc[,1],valsubc[,2])
par(op)
plot(valsubc,ylab=expression(X[t]),xlab=expression(X[t-1]))
x <- c(min(valsubc[,1]),max(valsubc[,1]))
lines(x, lmar$coefficients[1]+lmar$coefficients[2]*x,
      col=2, lty=2,lwd=2)
```



The slope is called the **correlation**. It is easy to get the correlations of all pairwise distributions (X_t, X_{t+h}), $h = 1, 2, 3, \ldots$! Here is the **auto-correlation function** up to $h = n$. We compare it to some IID variables.

```
n <- 30
op <- par(mfrow=c(1,2))
acf(valsub, lag.max=n, main='USD-Euro')
bm <-  ts(rnorm(n=length(valsub), mean=0, sd=1), frequency=2)
acf(bm, lag.max=n, main='White noise')
```



```
par(op)
```

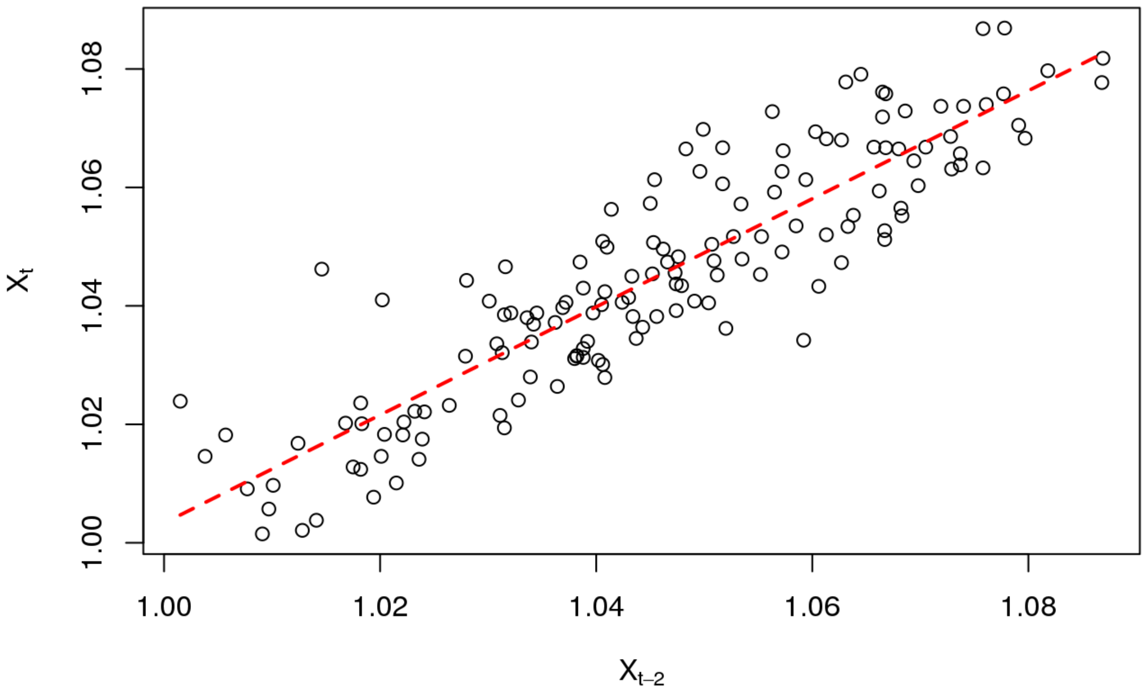
Partial correlations

There are more subtle ways to describe a sequence of correlations. Take the bivariate distribution with lag $n = 2$:

```
n <- 2
valsubcs <- valsub[1:(T-(n+1))]
for (i in 1:(n+1)){
  valsubcs <- cbind(valsubcs,valsub[(1+i):(T+i-(n+1))])
}
```

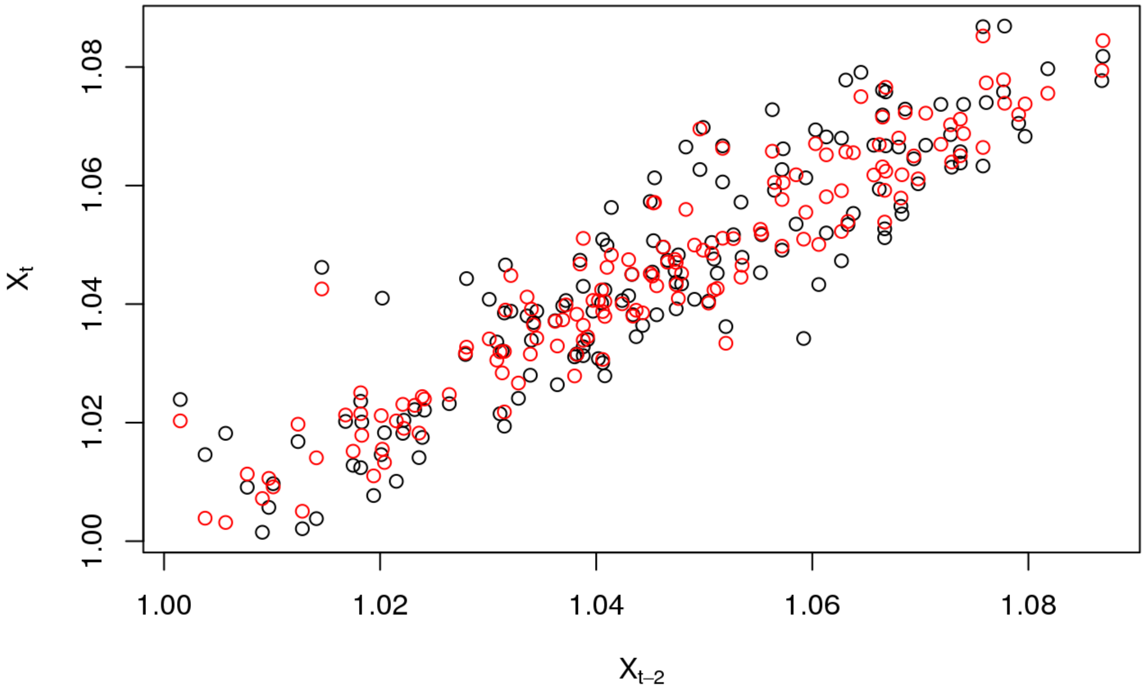
Let us use only X_{t-n} to predict X_t, and compare to the regression line:

```
myxlab <- bquote(expression(X[t-.(n)]))
lmar <- lsfit(valsubcs[,1],valsubcs[,n+1])
plot(valsubcs[,1],valsubcs[,n+1],ylab=expression(X[t]),xlab=eval(myxlab))
x <- c(min(valsubc[,1]),max(valsubc[,1]))
lines(x, lmar$coefficients[1]+lmar$coefficients[2]*x,
      col=2, lty=2,lwd=2)
```



Let us now use $X_{t-1}, \ldots, X_{t-n}$ to predict X_t and plot X_t (in black circles) and its predictor (in red circles):

```
lmar <- lsfit(valsubcs[,1:n],valsubcs[,n+1])
plot(valsubcs[,1],valsubcs[,n+1],ylab=expression(X[t]),xlab=eval(myxlab))
points(valsubcs[,1], lmar$coefficients[1]+valsubcs[,1:n]%*%lmar$coefficients[-1],
      col=2)
```



Finally plot the component of the last lag in this prediction, that is, we write

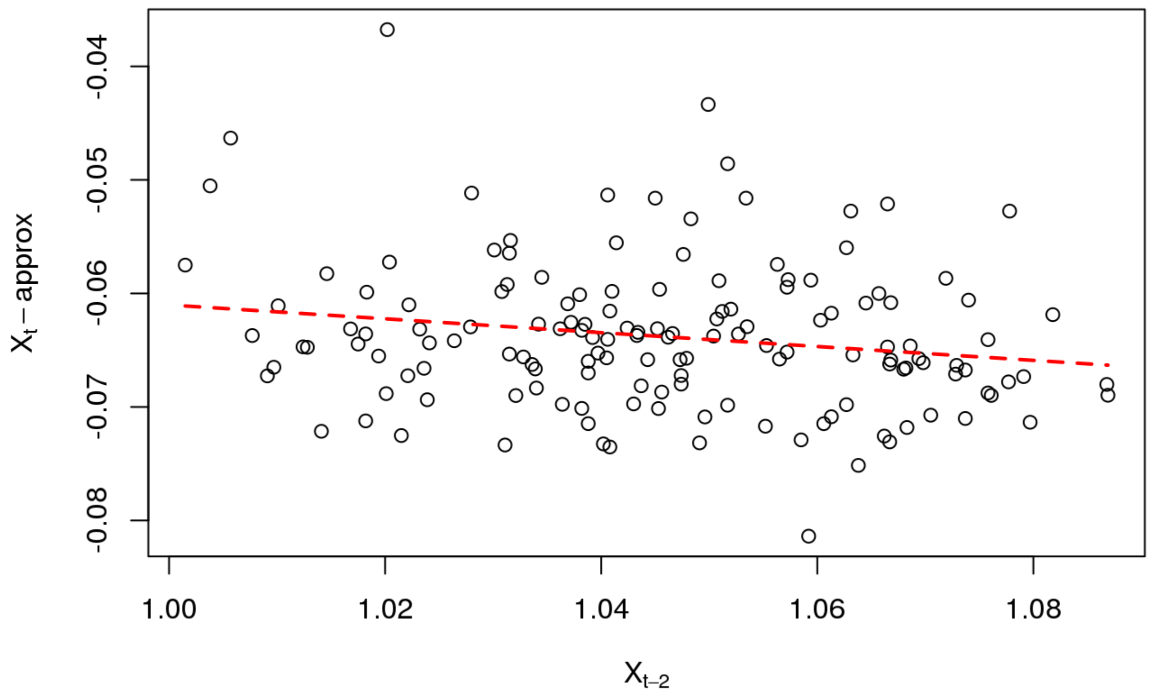
$$X_t = \phi_0 + \sum_{k=1}^{n-1} \phi_k X_{t-k} + \kappa(n) X_{t-n} + \epsilon_t$$

and we plot

$$X_t - \phi_0 - \sum_{k=1}^{n-1} \phi_k X_{t-k}$$

as a function of X_{t-n}, compared to the regression line $X_{t-n} \mapsto \kappa(n) X_{t-n}$:

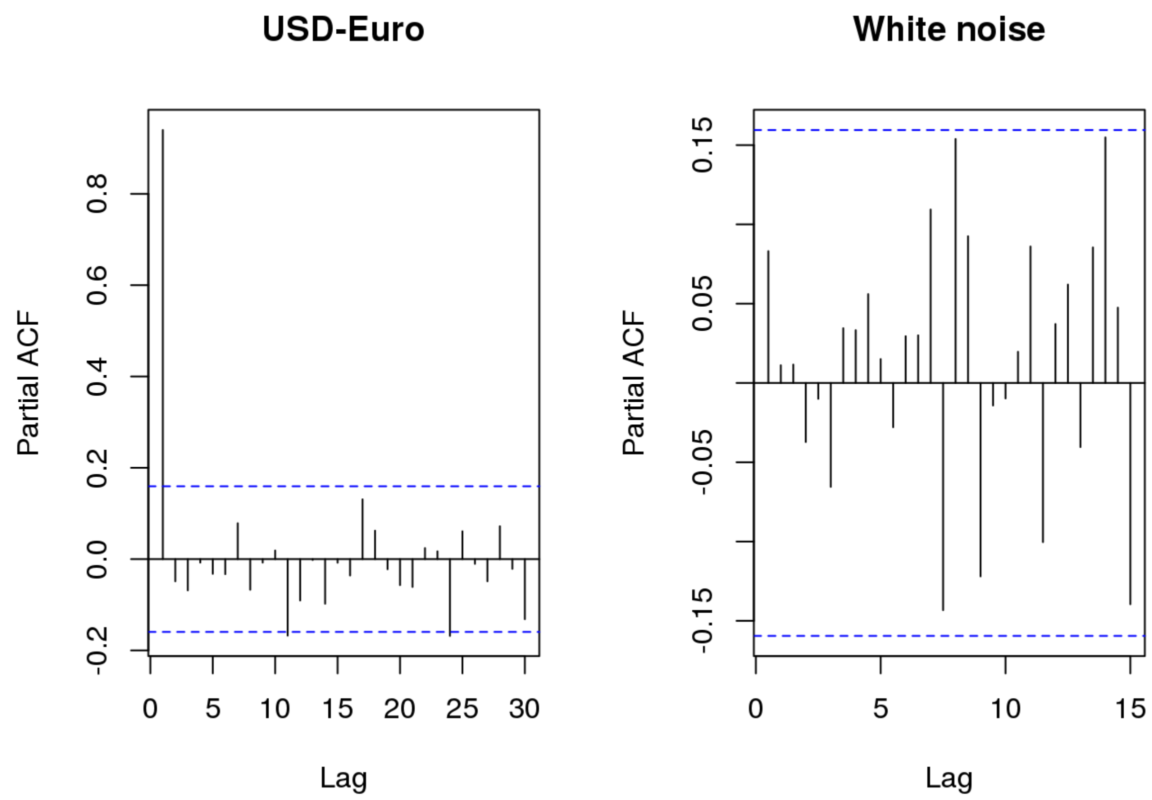
```
res <- valsubcs[,n+1]-lmar$coefficients[1]-as.matrix(valsubcs[,2:n]) %*% as.vector(lmar$coeffic
ients[3:(n+1)])
plot(valsubcs[,1],res,ylab=expression(X[t]-approx),xlab=eval(myxlab))
x <- c(min(valsubc[,1]),max(valsubc[,1]))
lines(x,lmar$coefficients[2]*x,
      col=2, lty=2,lwd=2)
```



The slope $\kappa(n)$ is called the **partial auto-correlation** at lag n.

Here is the **partial auto-correlation function up to** $h = n$

```
n <- 30
op <- par(mfrow=c(1,2))
pacf(valsub, lag.max=n, main='USD-Euro')
bm <-  ts(rnorm(n=length(valsub), mean=0, sd=1), frequency=2)
pacf(bm, lag.max=n, main='White noise')
```



```
par(op)
```