Solution to Series 6

1. a) The plot is shown in part c)

b) The state space model of the package `sspir` is given by

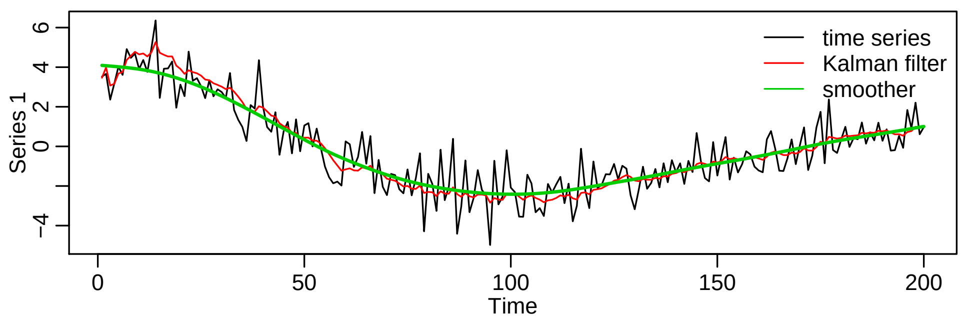
$$\begin{aligned} Z_t &= G_t Z_{t-1} + E_t, \quad E_t \sim \mathcal{N}(0, W_t) \text{ (state equation)}, \\ Y_t &= F_t^T Z_t + U_t, \quad U_t \sim \mathcal{N}(0, V_t) \text{ (observation equation)}. \end{aligned}$$

ϕ is a free parameter vector that can have influence on the matrices F_t, G_t, V_t and W_t. In our model, we have the dependencies

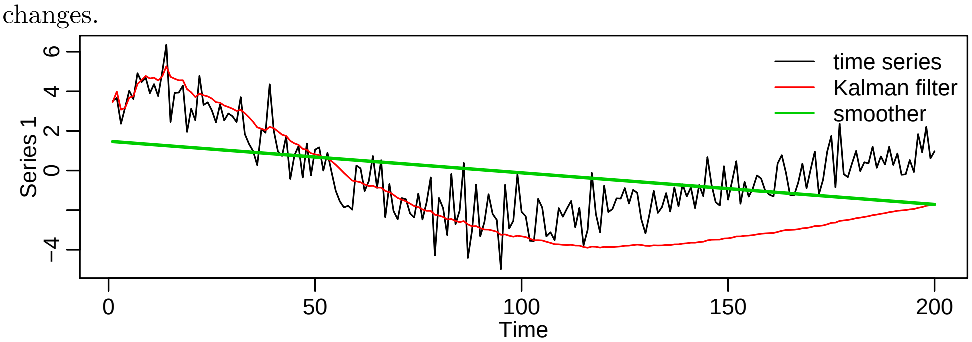
$$V_t = \phi_1, \quad W_t = \begin{pmatrix} \phi_2 & 0 \\ 0 & 0 \end{pmatrix}.$$

The initialization of the state vector Z (called θ in the package documentation) is $Z_0 \sim \mathcal{N}(m_0, C_0)$.

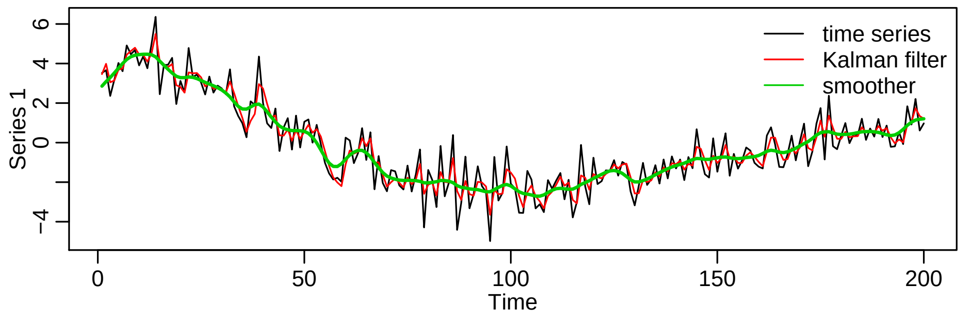
c)



d) When setting the error variance of the AR(2) process to zero (next figure), the second derivative of the process must be 0, that is, it must be a straight line. The *smoother* considers all data points and fits a straight line through them, which estimates the process quite badly. The *filter* fits the data better in the beginning, but runs into troubles as soon as the trend of the time series changes.

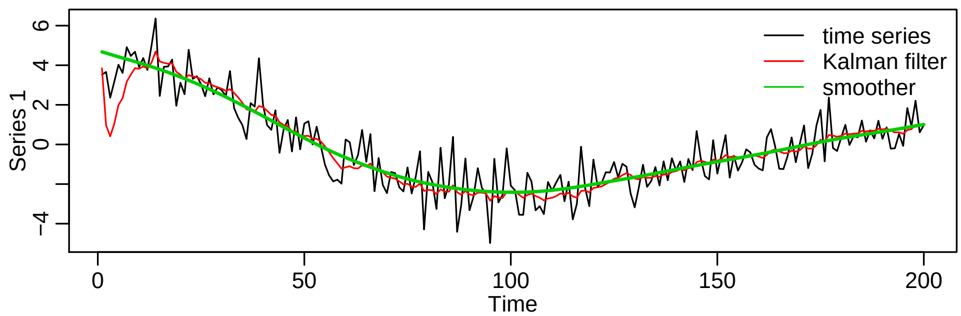


Setting the error variance of the AR(2) process to 0.06 (next figure) has the effect that the AR process doesn't have to be very smooth any more. The *filtered* as well as the *smoothed* series are now closer to the data.

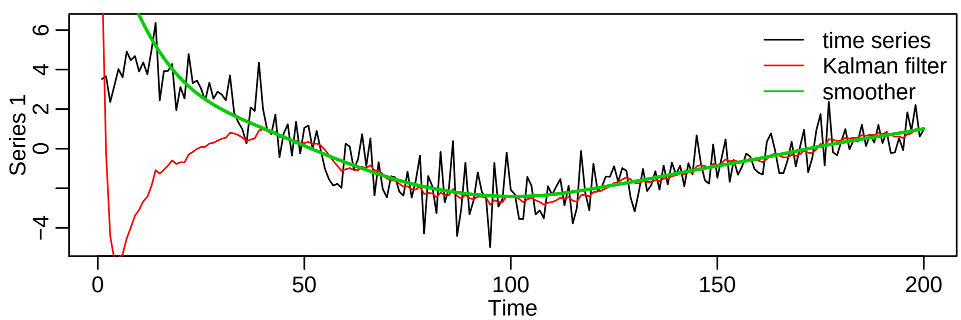


e) Despite the very unrealistic start value of 40, filter and smoother adapt to the data quite fast (first plot). Only by assigning this start value a high accuracy (a low variance), the adaptation takes more time (second plot). This is the reason why the variance of the initial state is usually chosen to be rather large.

```
> phi(ss.model) <- c(0.9, 0.0001)
> m0(ss.model) <- matrix(c(40, 40), ncol = 2)
> C0(ss.model) <- diag(c(20, 20))
```




```
> C0(ss.model) <- diag(c(1, 1))
```

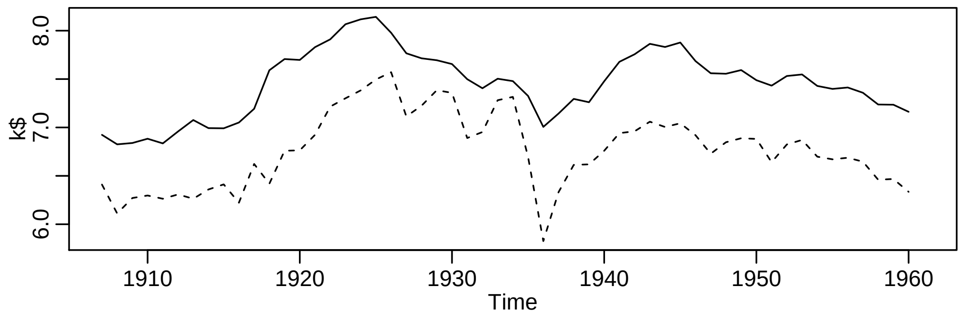


f) In order to get only positive values as variances (cf. part b)), we optimize the component-wise logarithm of the parameter vector ϕ. We minimize the negative log-likelihood with the function `nlm()`. The result is then

```
> exp(log.phi.min$estimate)
[1] 0.899998212 0.000099124
```

(Note that the model `ss.model` must be initialized as in part b) to avoid getting stuck in local optima.)

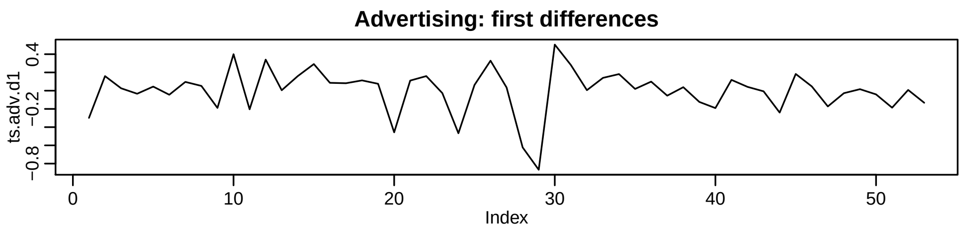
2. a) The plots clearly show that the time series are *not* stationary:

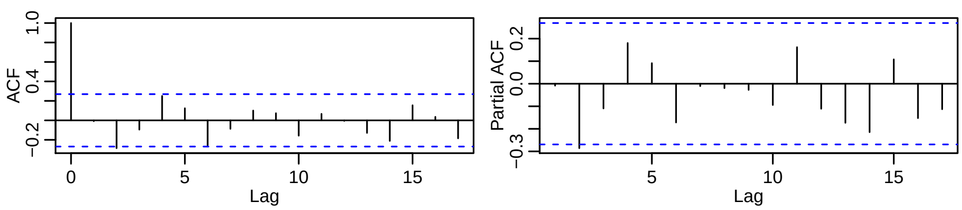


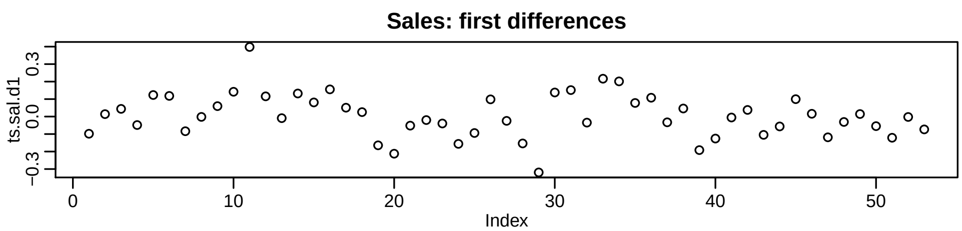
b) We first remove the missing values (last entry of the time series) and then calculate the first differences:

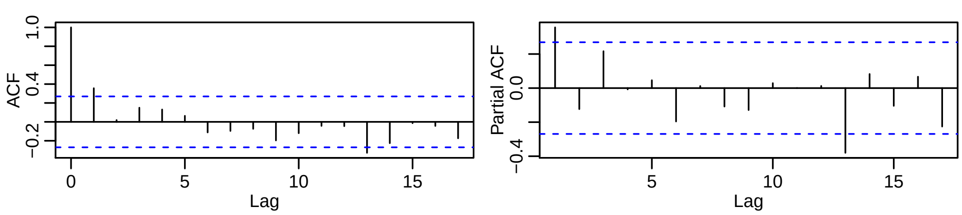
```
> ts.adv.d1 <- diff(ts.advert[!is.na(ts.advert)])
> ts.sal.d1 <- diff(ts.sales[!is.na(ts.sales)])
```

By differencing we can achieve stationarity as the following plots show (more or less):









c) The transfer function model

$$Y_{2,t} = \sum_{j=0}^{\infty} \nu_j Y_{1,t-j} + E_t$$

makes the assumption that a change in the advertising expenditures ($Y_{1,t}$) causes a change in the (future) sales ($Y_{2,t}$), but *not* vice versa.

d)
- From the correlogram of `d.adv.d1` we see that the input series $Y_{1,t} = X_{1,t} - X_{1,t-1}$ can be described as an AR(2) model. We fit it as follows:

```
> (r.fit.adv <- arima(ts.adv.d1, order = c(2, 0, 0)))
```

```
Call:
arima(x = ts.adv.d1, order = c(2, 0, 0))

Coefficients:
          ar1      ar2  intercept
      -0.0066  -0.2875    -0.0003
s.e.   0.1331   0.1314     0.0244

sigma^2 estimated as 0.05171:  log likelihood = 3.21,  aic = 1.59
```

Hence we get the model

$$Y_{1,t} = -0.0066 \cdot Y_{1,t-1} - 0.2875 \cdot Y_{1,t-2} + D_t \,,$$

where D_t is a white noise with variance $\widehat{\sigma}_D^2 = 0.052$ (see component `r.fit.adv$sigma2`). The mean of the time series can be regarded as zero (one gets an estimate of -0.0014).

Remark: One could also fit the AR(2) model of the first differences with the function `ar.burg()` or `ar.yw()`, resp. The estimates of the coefficients are quite similar, though.

- We apply the transformation as in the lecture:

```
> ts.D <- resid(r.fit.adv)
> ts.Z <- filter(ts.sal.d1, c(1, -r.fit.adv$model$phi), sides = 1)
```

In the transformed model

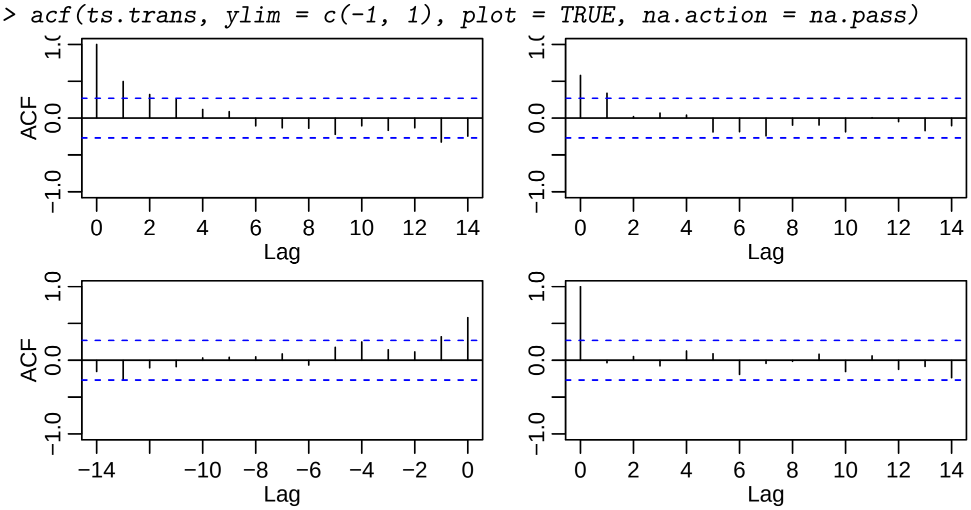
$$Z_t = \sum_{j=0}^{\infty} \nu_j D_{t-j} + U_t \ ,$$

the coefficients are the same as in the original transfer function model of part c). However, the time series D_t is *uncorrelated* here. Hence we can estimate the coefficients ν_j by

$$\widehat{\nu}_k = \frac{\widehat{\gamma}_{21}(k)}{\widehat{\sigma}_D^2}, \quad k \geq 0$$

where $\widehat{\rho}_{21}(k)$ denotes the empirical cross correlations of D_t and Z_t. The estimated coefficients $\widehat{\nu}_k$ are hence proportional to the empirical cross correlations $\widehat{\rho}_{21}(k)$ shown in the following plot.

```
> ts.trans <- ts.intersect(ts.Z, ts.D)
> acf(ts.trans, ylim = c(-1, 1), plot = TRUE, na.action = na.pass)
```



We see that $\widehat{\rho}_{21}(0)$ has the largest value. We find another large value at lag $k = -1$. This shows that, *contrary to our assumption* in part c), there is an influence of $Y_{2,t}$ on $Y_{1,t}$. Hence the modeling approach is not allowed since the prerequisites are not fulfilled. However, our analysis shows that there is a mutual influence between $Y_{2,t}$ and $Y_{1,t}$.

A change in the sales hence also causes a change in the advertising expenditures. This seems to be plausible in practice: the budget for advertising is usually established based on past sales, e.g. as a percentage of last year's sales.

- Estimation of the coefficients ν_j in R :

```
> gamma21 <- acf(ts.trans, plot = FALSE, type = "covariance",
+   na.action = na.pass)$acf[, 1, 2]
> round(gamma21/r.fit.adv$sigma2, 2)[1:6]
[1]  0.33  0.20  0.01  0.04  0.02 -0.11
```