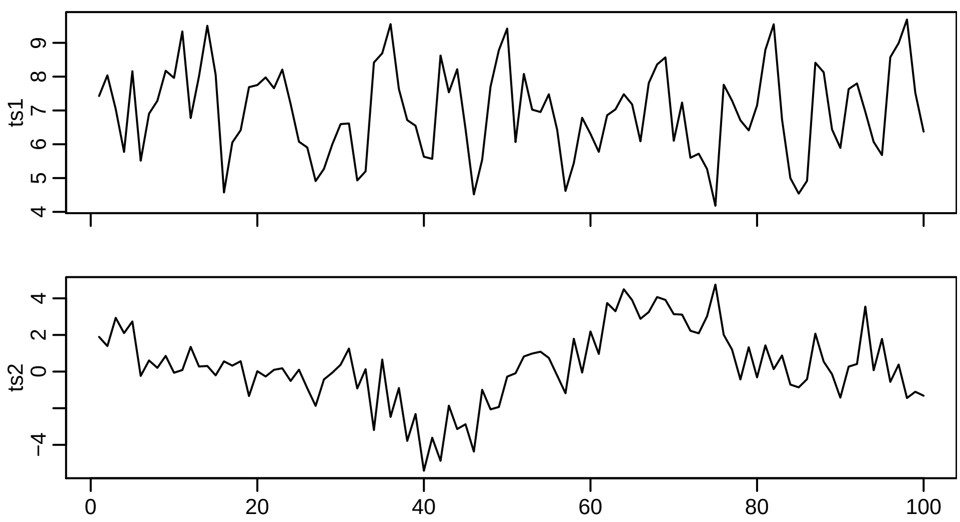
Solution to Series 4

1. a) Plots of the two time series:

```
> par(mfrow = c(2, 1), mar = c(0.5, 3, 2, 0), oma = c(2, 0, 0, 0),
    cex = 0.66, mgp = c(1.5, 0.6, 0))
> plot(ts1, xlab = "", xaxt = "n")
> axis(1, at = (0:5)*20, labels = rep(NA, 6))
> plot(ts2)
```



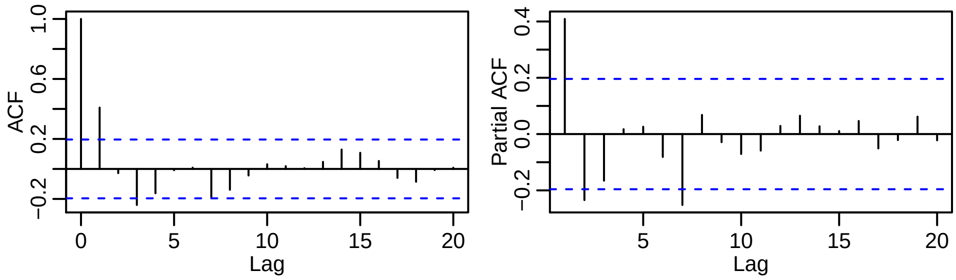
The **first time series**, ts1, is actually simulated from a stationary $\mathrm{AR}(2)$ process with coefficients $\alpha_1 = 0.5$ and $\alpha_2 = -0.2$, shifted by the constant amount 7. (Strictly speaking, stationarity of an $\mathrm{AR}(p)$ process demands that the mean be 0 throughout, which is not the case due to the shift. However, we can subtract the estimated mean and then still call it a stationary process, since constant vertical shifts do not affect the structure or dependencies involved in a time series.)
When looking at the plot, the stationarity of this process is nicely evident in the "identical behaviour" over 100 points in time: we see neither trend nor periodicity.
The **second time series** is also simulated from a stationary (!) $\mathrm{AR}(4)$ process with coefficients $\alpha_1 = 0.5$, $\alpha_2 = 0.5$, $\alpha_3 = -0.4$ and $\alpha_4 = 0.3$.
When looking at this plot, the stationarity shows up in rather unaccustomed way. It looks as if there is a nonlinear trend in the time series; thus even a stationary time series need not act quite as nicely as we would hope it to.
As we know the generating coefficients α_i for the two processes (which is not the case in reality) we can here proof the stationarity of the $\mathrm{AR}(p)$ process using the unit root test:

```
> t.ar1 <- c(0.5, -0.2)
> abs(polyroot(c(1, -t.ar1)))
[1] 2.236068 2.236068
> t.ar2 <- c(0.5, 0.5, -0.4, 0.3)
> abs(polyroot(c(1, -t.ar2)))
[1] 1.063564 1.092643 1.693630 1.693630
```

The absolute values of the complex roots of the polynomials $1 - \sum_{i=1}^{p} \alpha_i z^i$ are larger than 1, as required for stationarity.
Nevertheless for the second time series the absolute values of the complex roots are close to 1 and only exceed it by little. Therefore, the process is very close to non-stationarity (although it still is a stationary process).
In summary, we see that it is very difficult to recognize stationarity from a mere glance at a time series.

b) **First time series:**

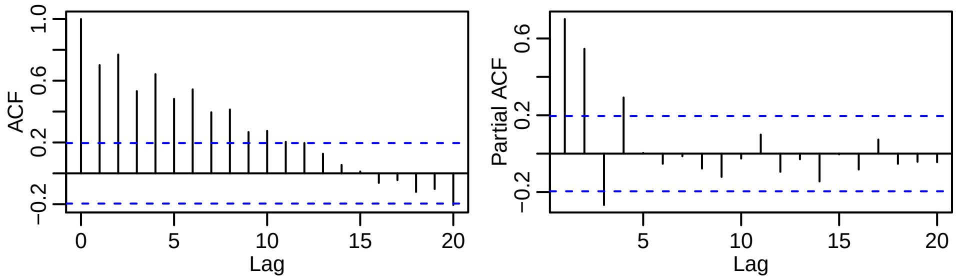
```
> acf(ts1)
> pacf(ts1)
```



The autocorrelations at lags 1 and 3 do seem to differ from zero; apart from this, a dampened sine wave seems to fit the autocorrelations at least plausibly well. As expected (when the model is known), the partial autocorrelations at lags 1 and 2 show significant deviations from zero, indicating that we should fit an $\mathrm{AR}(2)$ model. However, since the PACF at lag 7 is also slightly significant, an $\mathrm{AR}(7)$ would perhaps be more appropriate. In this case we would try to fit an $\mathrm{AR}(2)$ model first and analyse the residuals. If those still show some dependencies, it then would be natural to try fitting an $\mathrm{AR}(7)$ model.

Second time series:

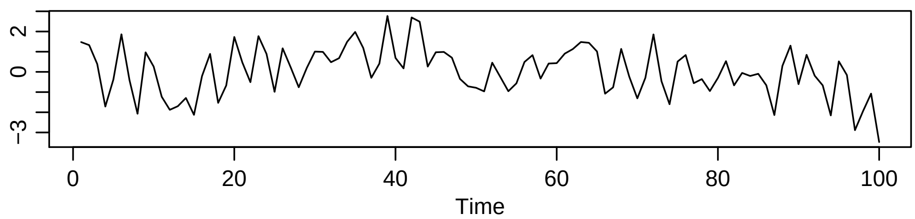
```
> acf(ts2)
> pacf(ts2)
```



Lags up to 12 all have positive autocorrelations, as evidenced by the series tending to stay where it already is for longer than e.g. `ts1`. However, its decay is not exponential, but much slower.
The partial autocorrelations at lags 1 to 4 are significant, which fits the actual model $\mathrm{AR}(4)$. All subsequent autocorrelations are much smaller.

2. a) The time series looks stationary:

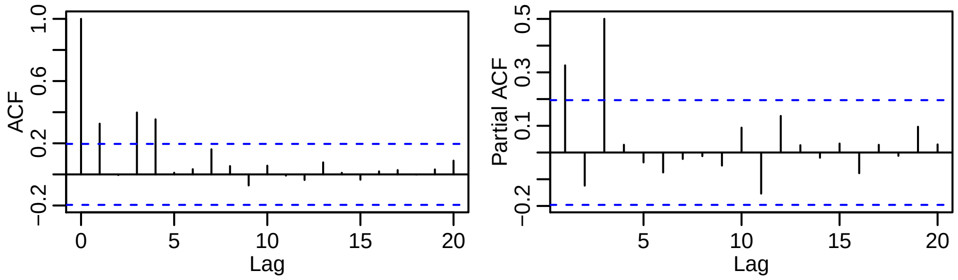
```
> plot(ts.sim, ylab = "")
```



b)
```
> abs(polyroot(c(1, - ar.coef)))
[1] 1.170785 1.193125 1.193125
```

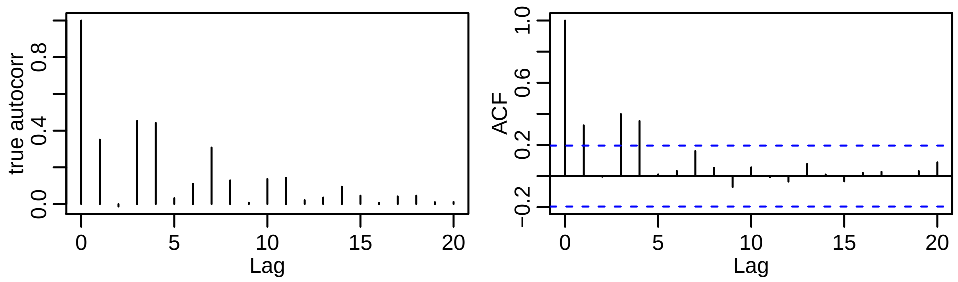
Indeed, all roots lie outside the unit circle, so the AR process is stationary.

c)
```
> acf(ts.sim, plot=TRUE)
> acf(ts.sim, type="partial", plot=TRUE)
```



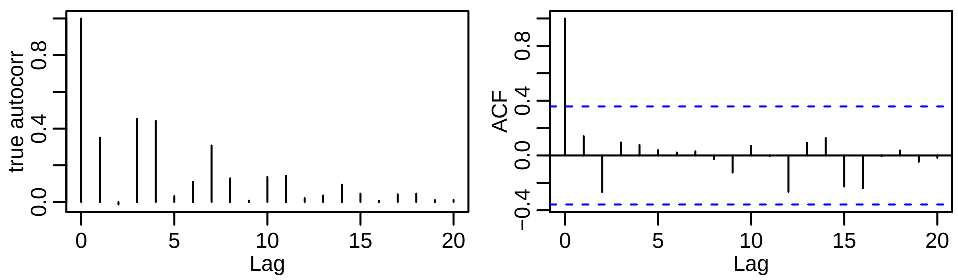
With the exception of lag 7, its autocorrelations decay rapidly. The partial autocorrelations are significant at lags 1 and 3, indicating an AR model of order at least 3. Subsequent lags are insignificant, although lags 10, 11 and 12 are noticeable.

```
> autocorr <- ARMAacf(ar=c(0.5, -0.4, 0.6), lag.max=20)
> plot(0:20, autocorr, type="h", xlab="Lag", ylab="true autocorr")
> acf(ts.sim, plot=T)
```



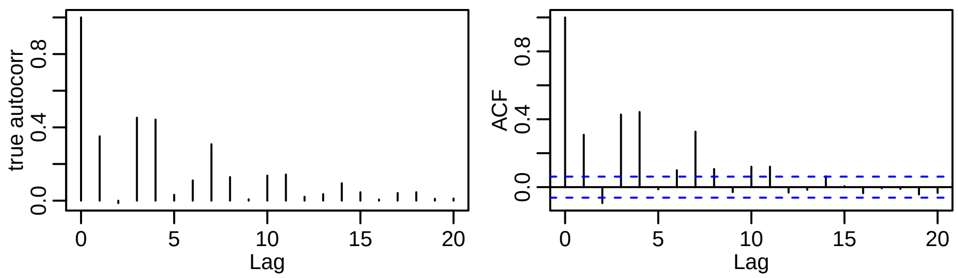
d) Time series of length 30:

```
> ts.sim2 <- arima.sim(list(ar = ar.coef), n = 30)
> plot(0:20, autocorr, type="h", xlab="Lag", ylab="true autocorr")
> acf(ts.sim2, plot=T, lag.max=20)
```



Time series of length 1000:

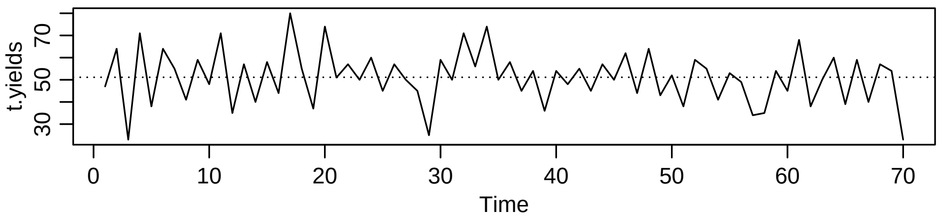
```
> ts.sim3 <- arima.sim(list(ar = ar.coef), n = 1000)
> plot(0:20, autocorr, type="h", xlab="Lag", ylab="true autocorr")
> acf(ts.sim3, plot=T, lag.max=20)
```



It is visible that the estimated autocorrelations get more precise the longer the time series is.

3. **a)** Plotting and calculating the mean:

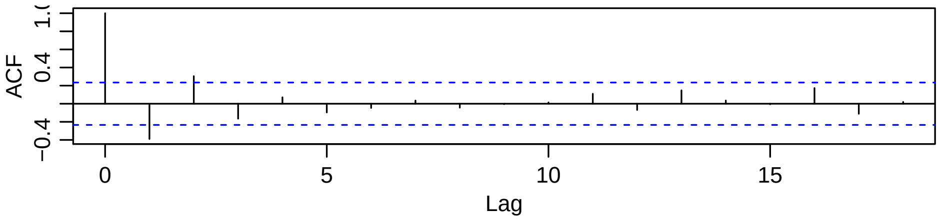
```
> mu <- mean(t.yields)
> plot(t.yields)
> abline(h = mu, lty=3)
```



We can regard this time series as being stationary.

b) Plotting the ACF:

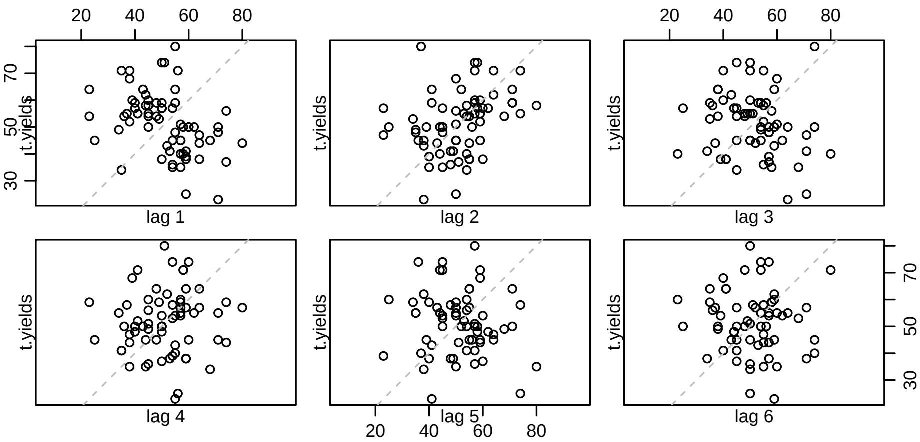
```
> acf(t.yields, plot = TRUE)
```



The correlogram shows us that for lags $k \geq 3$, the estimated autocorrelations $\widehat{\rho}\langle k\rangle$ do not differ significantly from 0. The first of these autocorrelations is negative; as the time series oscillates very noticeably, this negativity is not at all surprising.

Looking at lagged scatterplots:

```
> lag.plot(t.yields, lag = 6, layout = c(2, 3), do.lines = FALSE)
```



In the lagged scatterplot with lag 1 the pairs $[x_t, x_{t+1}]$ show the negative linear relationship we expected from the correlogram. For lag 2, however, the lagged scatterplot shows up a positive linear relationship, and for lag $k \geq 4$ we see no further correlation. The pairs $[x_t, x_{t+3}]$ (lagged scatterplot at lag 3) still have a slightly negative connection, but the correlogram tells us that we can assume $\widehat{\rho}\langle 3\rangle = 0$.

c) The variance of the arithmetic mean $\widehat{\mu}$ is

$$\mathrm{Var}(\widehat{\mu}) = \frac{1}{n^2}\,\gamma(0)\left(n + 2\sum_{k=1}^{n-1}(n-k)\,\rho(k)\right).$$

From the correlogram in Part b) we see that the estimated autocorrelations $\widehat{\rho}\langle k\rangle$ do not differ significantly from 0 for lags $k \geq 3$. Thus we can set all the autocorrelations $\rho(k)$ for $k \geq 3$ to 0. We obtain

$$\mathrm{Var}(\widehat{\mu}) = \frac{1}{n^2}\,\gamma(0)\Big(n + 2(n-1)\rho(1) + 2(n-2)\rho(2)\Big).$$

To estimate the variance of $\widehat{\mu}$, we replace $\gamma(0), \rho(1)$ and $\rho(2)$ by their estimates.

R code:

```
> n <- length(t.yields)
> gamma0 <- var(t.yields) * (n - 1)/n
> rho <- acf(t.yields, plot=F)$acf
> Var.mu <- n^(-2) * gamma0 * (n + 2*sum((n - 1:2)*rho[2:3]))
```

This yields an estimated variance of $\widehat{\mathrm{Var}}(\widehat{\mu}) = 1.643$.
The bounds of an approximate 95% confidence interval for the mean yield are then given by

$$\widehat{\mu} \pm 1.96 \cdot \mathrm{se}(\widehat{\mu}) = \widehat{\mu} \pm 1.96 \cdot \sqrt{\widehat{\mathrm{Var}}(\widehat{\mu})}\ .$$

In our case, we get a confidence interval of $[48.62, 53.64]$.
If we assume independence, the variance of $\widehat{\mu}$ is estimated as

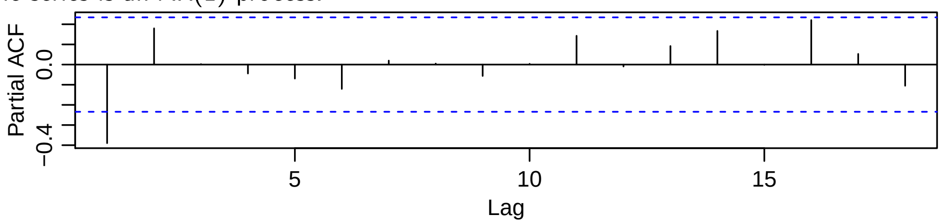
$$\widehat{\mathrm{Var}}(\widehat{\mu}) = \frac{1}{n^2} \sum_{s=1}^{n} \widehat{\mathrm{Var}}(X_s) = \frac{\widehat{\gamma}(0)}{n} = 1.997\ .$$

Under this independence assumption, therefore, an approximate 95% confidence interval for the mean yield is given by

$$[\widehat{\mu} - 1.96 \cdot \mathrm{se}(\widehat{\mu}), \widehat{\mu} + 1.96 \cdot \mathrm{se}(\widehat{\mu})] = [48.36, 53.90]\ .$$

Thus the correct specification of the independence structure here leads to a confidence interval which is 10% narrower.

d) Only the first partial autocorrelation is significantly different from zero. Thus we can assume that our time series is an AR(1) process.



e) For an AR(1) model, the Yule-Walker equations simplify to

$$\widehat{\rho}(1) = \widehat{\alpha} \cdot \widehat{\rho}(0), \quad \widehat{\rho}(0) = 1$$

σ^2 can be estimated by $\widehat{\sigma}^2 = \widehat{\sigma}_X^2 \cdot (1 - \widehat{\alpha}^2)$. Here, we get $\widehat{\alpha} = \widehat{\rho}(1) = -0.390$ and $\widehat{\sigma}^2 = 120.266$.
Determining parameters with the help of R:

```
> r.yw <- ar(t.yields, method = "yw", order.max = 1)
> r.yw$ar
[1] -0.3898783
> r.yw$var.pred
[1] 122.0345
```