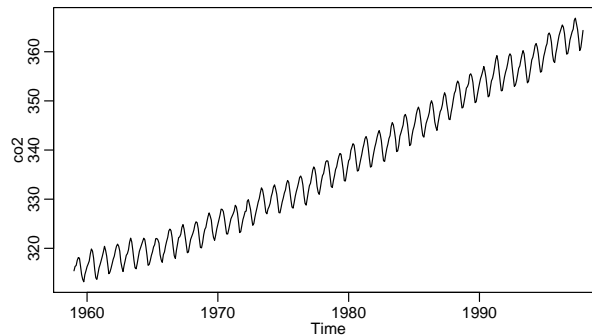


## Solution to Series 3

1. a) `> data(co2)`  
`> plot(co2)`



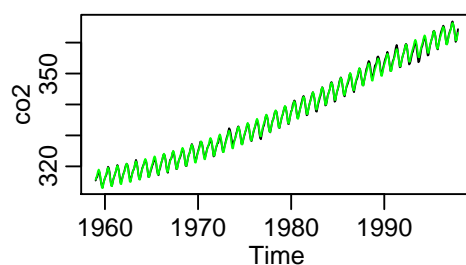
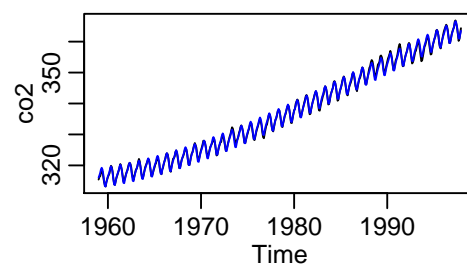
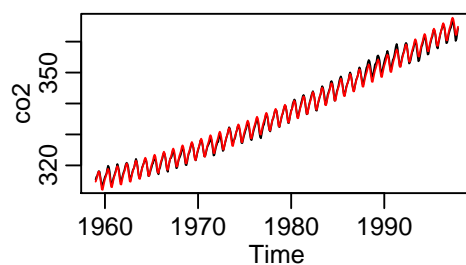
The time series under investigation is non-stationary. Its non-stationary properties consist of a trend and deterministic seasonal fluctuations (by month). We can decompose this time series into its trend, its seasonal fluctuations (month effects) and the remainder:

$$X_t = m_t + \alpha_{i(t)} + E_t$$

b) If the trend can be described by a polynomial of degree  $k$ , we then have

$$m_t = \beta_0 + \beta_1 \cdot t + \beta_2 \cdot t^2 + \dots + \beta_k \cdot t^k$$

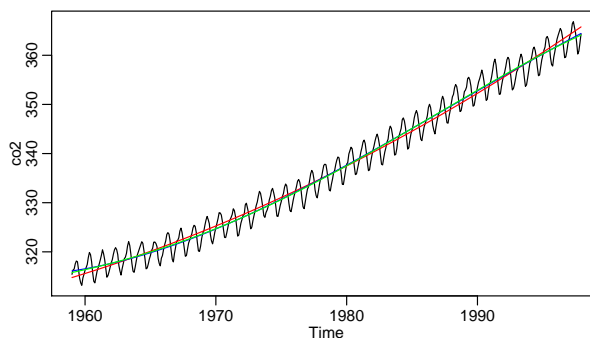
```
> tr <- as.numeric(time(co2))
> tc <- tr-mean(tr) # centering x-values to mitigate potential collinearity
> mm <- factor(rep(month.name, length(co2)/12), levels=month.name)
> fit2 <- lm(co2~tc+I(tc^2)+mm)
> fit3 <- lm(co2~tc+I(tc^2)+I(tc^3)+mm)
> fit4 <- lm(co2~tc+I(tc^2)+I(tc^3)+I(tc^4)+mm)
> par(mfrow=c(2,2))
> plot(co2)
> lines(tr,fitted(fit2), col="red")
> plot(co2)
> lines(tr, fitted(fit3),col="blue")
> plot(co2)
> lines(tr, fitted(fit4), col="green")
```



```

c) > cf2 <- coef(fit2)
> cf3 <- coef(fit3)
> cf4 <- coef(fit4)
> t.est2 <- cf2[1]+cf2[2]*tc+cf2[3]*tc^2
> t.est3 <- cf3[1]+cf3[2]*tc+cf3[3]*tc^2+cf3[4]*tc^3
> t.est4 <- cf4[1]+cf4[2]*tc+cf4[3]*tc^2+cf4[4]*tc^3+cf4[5]*tc^4
> t.adj2 <- t.est2-mean(t.est2)+mean(co2)
> t.adj3 <- t.est3-mean(t.est3)+mean(co2)
> t.adj4 <- t.est4-mean(t.est4)+mean(co2)
> plot(co2)
> lines(tr, t.adj2, col="red")
> lines(tr, t.adj3, col="blue")
> lines(tr, t.adj4, col="green")

```

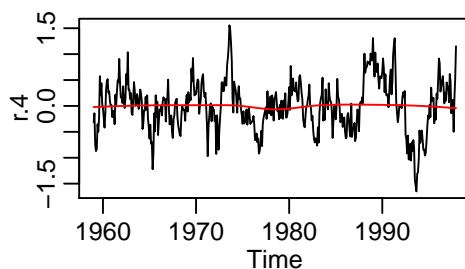
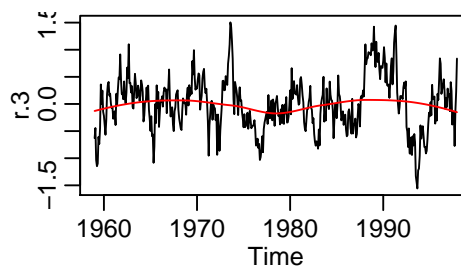
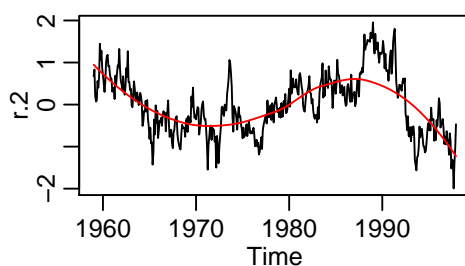


With this plot its hard to tell which polynomial fits best.

```

d) > par(mfrow=c(2,2))
> r.2 <- co2-fitted(fit2)
> plot(r.2)
> fit <- loess(r.2~tr)
> lines(tr, fitted(fit), col="red")
> r.3 <- co2-fitted(fit3)
> plot(r.3)
> fit <- loess(r.3~tr)
> lines(tr, fitted(fit), col="red")
> r.4 <- co2-fitted(fit4)
> plot(r.4)
> fit <- loess(r.4~tr)
> lines(tr, fitted(fit), col="red")

```



The polynomial of degree 4 seems to fit the data best.

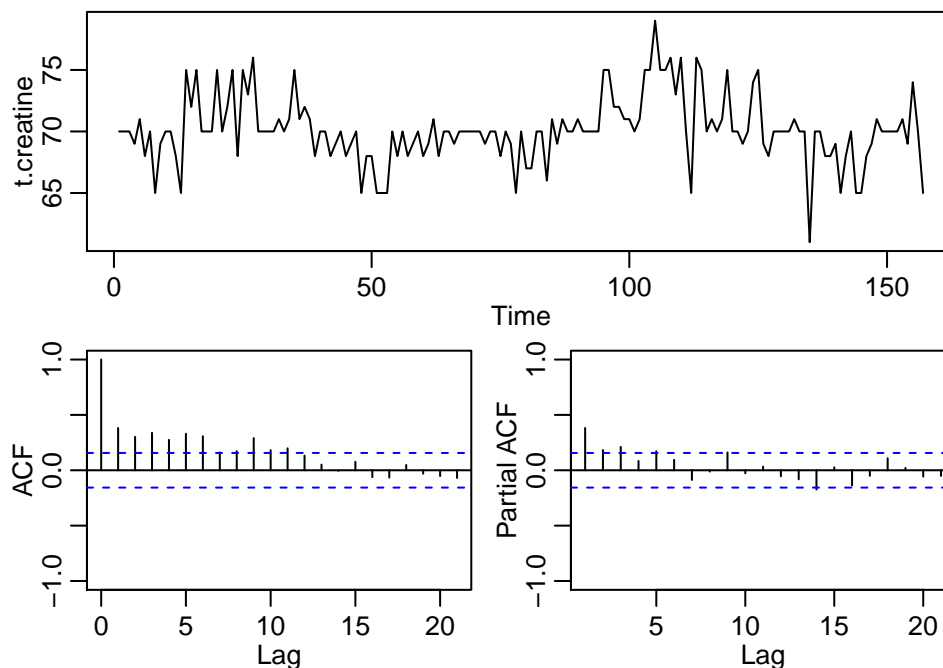
2. a) If the analyzing machine works correctly, the measurement it takes at time  $t$  is independent of the previous measurements:

$$X_t = \mu + E_t, \quad E_t \text{ i.i.d.},$$

where  $\mu$  denotes the creatine content of a standard sample.

- b) 

```
> d.creatine <- read.table("http://stat.ethz.ch/Teaching/Datasets/WBL/kreatin.dat",
  header = TRUE)
> t.creatine <- ts(d.creatine[, 2], start = 1, frequency = 1)
> layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
> plot(t.creatine)
> acf(t.creatine, plot = TRUE, ylim=c(-1,1))
> pacf(t.creatine, plot = TRUE, ylim=c(-1,1))
```



Since the data exhibit strong time-based correlation, the ideal model does *not* fit.

Thus the next steps would be to check the machine (perhaps some residue from the samples remains inside) and to ensure the standard samples really are manufactured in a way that rules out correlation.

**Note:** If you consider this time series to be non-stationary (not having a constant level, exhibiting standard autocorrelations that decay very slowly), then you are not the only one to think this way. If this is true, the machine must in any case be regarded as useless: since the standard samples all have the same creatin content, the measurements should have constant expectation.

3. The **time series plot 1** belongs to correlogram II. The autocorrelations reflect the periodic fluctuations of the series.

The **time series plot 2** belongs to correlogram I. On average, values from this time series are around the same as their predecessors. Thus the autocorrelations are positive for small  $k$ , and then become 0.

The **time series plot 3** belongs to correlogram III. The strong oscillation exhibited by this time series is evident in the changing signs of its autocorrelations.

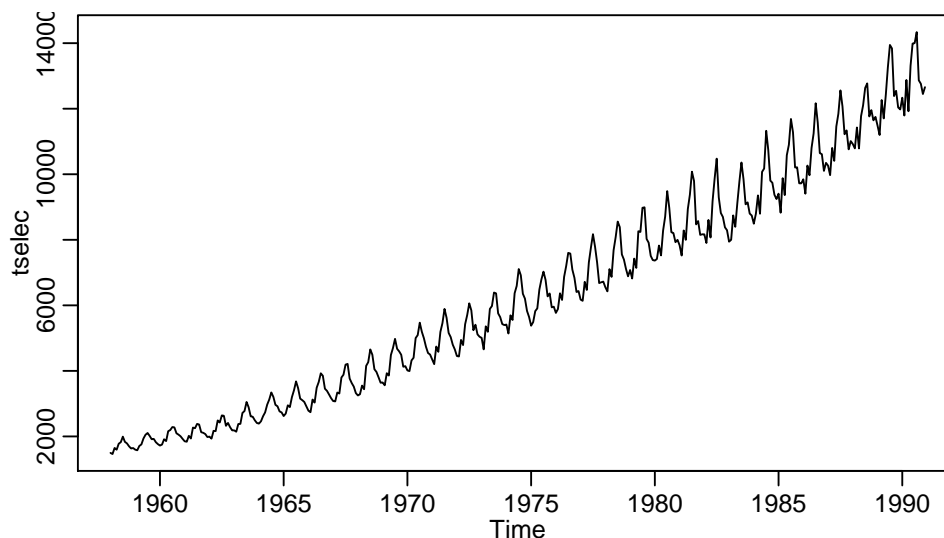
In order to identify which correlogram now belongs to which set of partial autocorrelations, we use the fact that the standard ( $\hat{\rho}(k)$ ) and partial ( $\hat{\pi}(k)$ ) autocorrelations are identical at lag 1:

$$\hat{\pi}(1) = \hat{\rho}(1)$$

It is thus clear which set of partial correlations belongs to which correlogram:

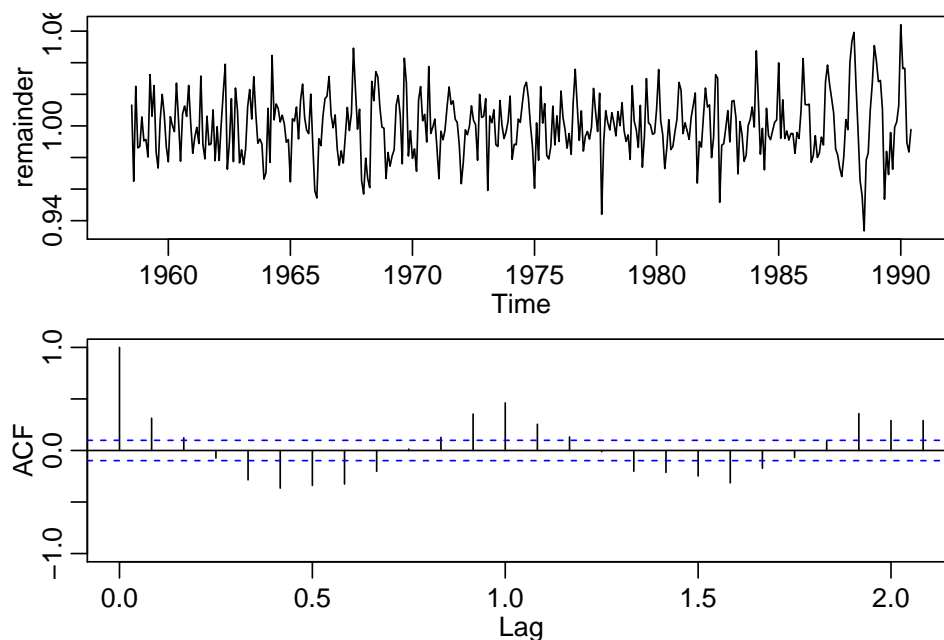
Time series	std. autocorr.	part. autocorr.
1	II	C
2	I	A
3	III	B

4. a) `> plot(tselec)`



The time series shows a clear trend and seasonal effect. However, the estimated ACF plotted in the correlogram is based on the assumption of (weak) stationarity of the time series ( $X_t$ ). If one calculates and plots the ACF of a time series with trend and/or seasonal component anyway, one mainly sees those strong non-random effects in the correlogram, while the weaker dependency structure of the random remainder of the time series is almost completely hidden.

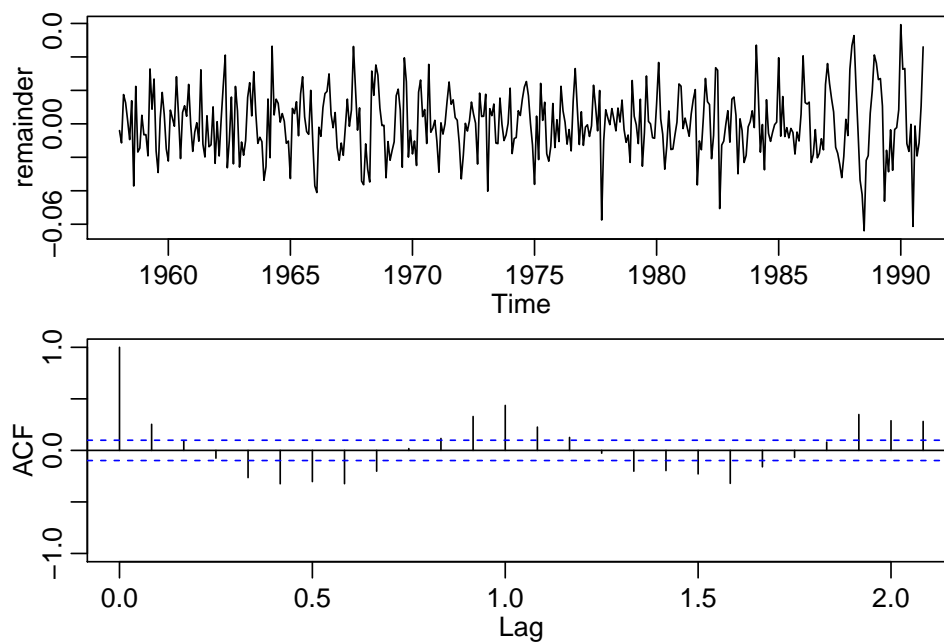
b) `> decomp <- decompose(tselec, type = "multiplicative")`  
`> par(mfrow = c(2, 1))`  
`> plot(decomp$random, ylab = "remainder")`  
`> acf(decomp$random, na.action = na.pass, plot = TRUE, ylim=c(-1,1))`



The trend is removed from the time series, but we still see a periodic behaviour in the computed remainder. The period of one year equals that of the seasonal component of the original time series. These effects are also visible in the damped harmonic behaviour of the correlogram.

c) We start by the averaging approach:

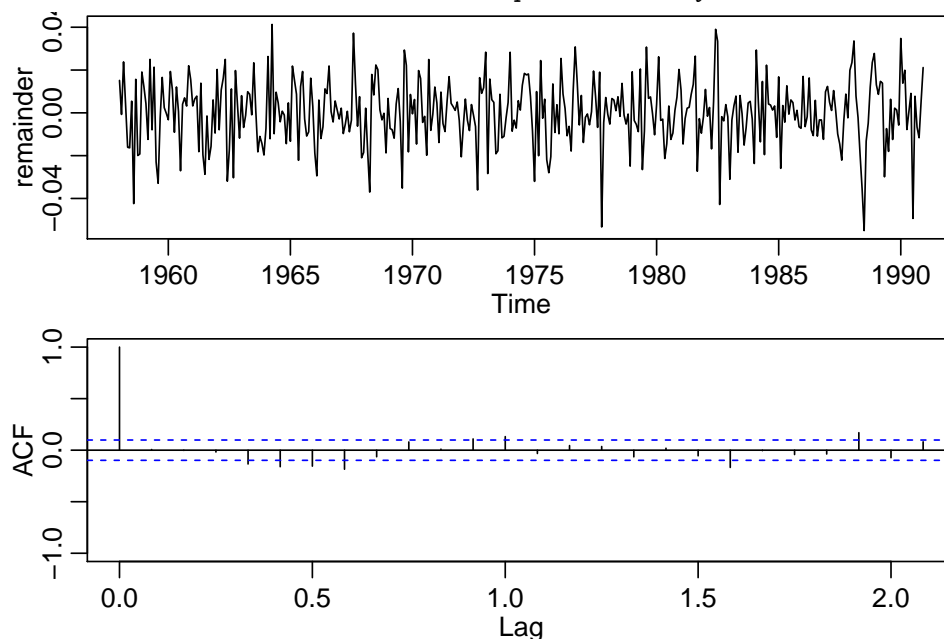
`> elec.stl <- stl(log(tselec), s.window = "periodic")`  
`> par(mfrow = c(2, 1))`  
`> plot(elec.stl$time.series[, "remainder"], ylab = "remainder")`  
`> acf(elec.stl$time.series[, "remainder"], plot = TRUE, ylim=c(-1,1))`



The outcome is very similar to that of `decompose()`; we still see a slight periodic behaviour with the period of one year, as well in the plot as in the correlogram.

Using a smoothing window of 19 (gives a reasonable monthplot; other values of similar magnitude are also fine), we find the following picture:

```
> elec.stl <- stl(log(tselec), s.window = 19)
> par(mfrow = c(2, 1))
> plot(elec.stl$time.series[, "remainder"], ylab = "remainder")
> acf(elec.stl$time.series[, "remainder"], plot = TRUE, ylim=c(-1,1))
```



Now, the periodicity is removed much better from the time series; the estimated ACF at lag 1 is not significant any more.

The difference between the averaging and the smoothing window approach goes along with different concepts behind the two methods. In the averaging approach, we model the seasonal component as truly periodic and explain the slight deviations from this fitted component as a stationary random process, which in our case has the same periodicity as the underlying seasonal effect. In the smoothing window approach, we allow the seasonal component to (slightly) vary over time, which allows to get rid of effects with the frequency of the seasonal component in the remainder. On the other hand, this approach is more susceptible to overfitting, since the slight deviations from a truly periodic behaviour in the seasonal component get the interpretation of being “deterministic”.

- d) The seasonal oscillations of the time series (see plot in Task a)) are approximately proportional to the trend. Therefore, a multiplicative model is appropriate for describing the time series:

$$X_t = T_t \cdot S_t \cdot I_t .$$

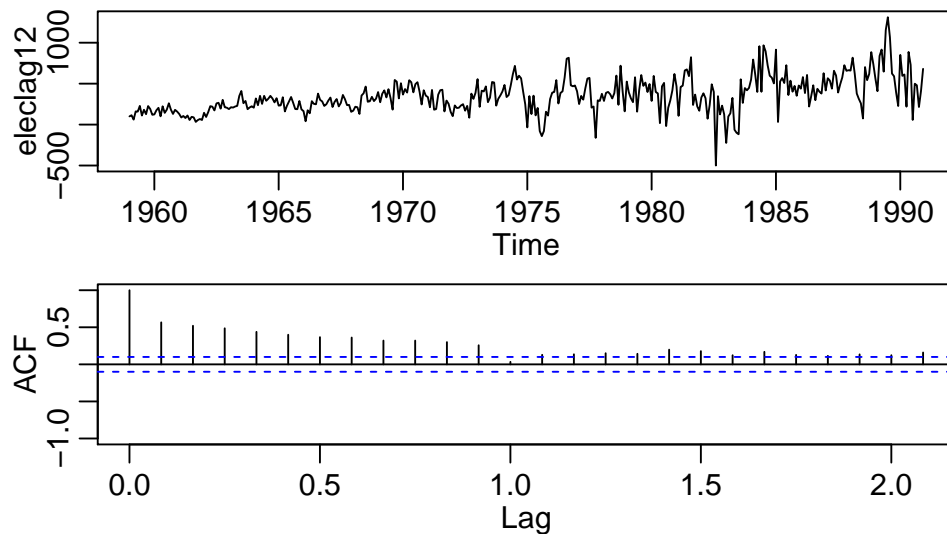
A log-transformation maps a multiplicative to an additive model:

$$\log(X_t) = \log(T_t) + \log(S_t) + \log(I_t) .$$

The R function `decompose()` can handle multiplicative models directly, whereas `stl()` requires an additive model.

- e) 

```
> eleclag12 <- diff(tselec, lag=12)
> par(mfrow=c(2,1))
> plot(eleclag12)
> acf(eleclag12, ylim=c(-1,1))
```



The acf looks quite different. We have strong autocorrelation. It looks like we could not get rid of the whole trend. It is probably not linear. This method is not appropriate here.