Series 2

1. We would like to illustrate various methods for descriptive decomposition and elimination of trends using the data `hstart`.

This data contains monthly data on the start of residential construction in the USA within the time frame of January 1966 to January 1974. The data have undergone some transformation unknown to us (perhaps an index over some baseline value has been calculated, or perhaps the data are to be read as $x \cdot 10^?$ construction permits). In our opinion, this makes these data a good didactic choice for illustrating the theory.

(Source: U. S. Bureau of the Census, Construction Reports.)

Importing the data (without `header=T`!) and preparing them:

```
> hstart <- read.table("http://stat.ethz.ch/Teaching/Datasets/WBL/hstart.dat")
> hstart <- ts(hstart[,1], start=1966, frequency=12)
```

a) Make a time series plot. Is this a stationary time series? If not, what kind of non-stationarity is evident? Into which components might this time series be decomposed sensibly?

b) Modelling
Decompose the time series into the components specified in a) using a parametric model. Plot the time series, including fitted values, and comment on any differences. Choose the order of the polynomial according to how good it fits the real data. Compare the orders 3, 4 and 6.

R-Hint:

```
> Time <- 1:length(hstart)
> Months <- factor(rep(month.name, length(hstart)/12), levels=month.name)
> H.lm  <- lm(hstart ~ Months + Time + I(Time^2) + I(Time^3) + ...)
> H.fit <- ts(fitted(H.lm), start=1966, freq=12)
> lines(H.fit, lty=3, col=2)                    # H.fit is added to the plot from part a)
```

If you want to compare the different polynomials you can either plot them and compare how good they fit the real data or plot the residuals, `hstart-H.fit` versus Time and check if there is still a structure.

c) STL decomposition
Decompose the time series in trend, seasonal component and remainder using the non-parametric STL method, and add data from this decomposition to the plot from part a).

R-Hint:
The decomposition is made using `H.stl <- stl(hstart, s.window="periodic")`
Note: The smoothing parameter for the seasonal effect is chosen by means of `s.window`. If `s.window="periodic"`, the seasonal effect is estimated by averaging. It is also possible to specify a value for the smoothing parameter (an odd number). Try e.g. `H.stl.var <- stl(hstart, s.window = 15)`, and compare the result of this to `H.stl`. Incidentally, `summary()` can be used for displaying the values of `window`.
The trend estimation parameter can be set using `t.window`. Unlike `s.window`, this argument does have a default value (cf. the help file). Perhaps you could try to vary this parameter as well. The documentation for R and the help files give more details.
Trend, seasonal component and remainder of the STL are stored in

```
H.st$time.series[,"trend"]
H.st$time.series[,"seasonal"]
H.st$time.series[,"remainder"]
```

Have a look at the output of `str(H.stl)` for more details.
Plot (in the same plot) the residuals from your favorite model in b) and the remainder from the STL-Decomposition vs. time.

d) Plot the components of the STL decomposition and its estimates of monthly effects. Compare the monthly effects of the original decomposition `H.stl` to that of the decomposition with chosen smoothing parameter `H.stl.var`. What is better, constant or variable seasonal effects?

R-Hint:

```
> plot(H.stl)                                              # STL decomposition
> seasonal <- H.stl$time.series[,1]
> seasonal.var <- H.stl.var$time.series[,1]
                         # Alternative, cf. comment in the R hints for part c)
> monthplot(seasonal)                  # Monthly effects estimated by averages
> monthplot(seasonal.var)
```

e) The special filter

$$Y_t = \frac{1}{24}\Big(X_{t-6} + 2X_{t-5} + \ldots + 2X_t + \ldots + 2X_{t+5} + X_{t+6}\Big)$$

can be used for computing a trend estimate. Plot this, the STL trend and the data in a single plot. What are the differences between this fit and the previous ones? What is better, what is worse?

R-Hint:

```
> plot(hstart,lty=3)
> H.filt <- filter(hstart, c(1,rep(2,11),1)/24 )
> trend <- H.stl$time.series[,2]
> lines(trend, col=3)
> lines(H.filt, lty=2, col=2)
```

Labelling:

```
> legend(1966, 235, legend=c("Time series","Filter","STL"),
+                                         col=c(1,2,3), lty=c(3,2,1))
```

f) **Differences** (optional)
Try to remove the trend and seasonal effects by computing differences. After removing seasonal effects, choose some linear trend elimination method from the course notes and plot the outcome.

R-Hint:

```
> H.y <- diff(hstart, lag=12)      # time series differences Y_t = X_t - X_{t-k}, lag = k
> H.z <- diff(H.y, lag=...)
> par(mfrow=c(2,1))
> plot(H.y)
> plot(H.z)
```

2. In this exercise, we shall investigate the properties of various filters (running weighted means). To this end, we look at the seasonally adjusted times series from Problem 1, which we obtain by subtracting the estimated monthly effects (which sum to zero) from the original time series.

a) Plot the seasonally adjusted time series.

R-Hint:

```
> seasonal <- H.stl$time.series[,1]         # STL decomposition, cf. Problem 1
> hstart.ds <- hstart - seasonal
> plot(hstart.ds)
```

b) Compute the following 4 filters (X_t is the seasonally adjusted time series) .

1. $Y_t = \frac{1}{5}X_{t-2} + \frac{1}{5}X_{t-1} + \frac{1}{5}X_t + \frac{1}{5}X_{t+1} + \frac{1}{5}X_{t+2}$
2. $Z_t = \frac{1}{5}Y_{t-2} + \frac{1}{5}Y_{t-1} + \frac{1}{5}Y_t + \frac{1}{5}Y_{t+1} + \frac{1}{5}Y_{t+2}$
3. $A_t = \frac{1}{25}\big(X_{t-4} + 2X_{t-3} + 3X_{t-2} + 4X_{t-1} + 5X_t + 4X_{t+1} + 3X_{t+2} + 2X_{t+3} + X_{t+4}\big)$
4. $B_t = \frac{1}{9}\big(X_{t-4} + X_{t-3} + X_{t-2} + X_{t-1} + X_t + X_{t+1} + X_{t+2} + X_{t+3} + X_{t+4}\big)$

R-Hint:

```
> hstart.fil1 <- filter(hstart.ds, rep(1,5)/5)
> hstart.fil2 <- filter(hstart.fil1, rep(1,5)/5)
> hstart.fil3 <- filter(hstart.ds, c(1:4,5,4:1)/25)
> hstart.fil4 <- filter(hstart.ds, rep(1,9)/9)
```

c) Plot the seasonally adjusted data and the results from filters 3 and 4 on top of each other.

R-Hint:

```
> plot(hstart.ds, lty=3)
> lines(hstart.fil3, lty=1)
> lines(hstart.fil4, lty=2)
```

Labelling:

```
> legend(1966, 210, legend=c("Time series","Filter 3","Filter 4"), lty=c(3,1,2))
```

d) Repeat part c) with filters 2 and 3. Comments?

R-Hint:

```
> plot(hstart.ds, lty=3)
> lines(hstart.fil2, lty=1)
> lines(hstart.fil3, lty=2)
```

Labelling:

```
> legend(1966,210,legend=c("Time series","Filter 2","Filter 3"),lty=c(3,1,2))
```

3. The dataset `http://stat.ethz.ch/Teaching/Datasets/WBL/zeitsim.dat` contains the following four simulated time series.

`Y1`:	$Y_t = E_t - 0.5 \cdot E_{t-1}$,	$E_t \sim \mathcal{N}(0,1)$ i.i.d., $E_0 = 0$
`Y2`:	$Y_t = Y_{t-1} + E_t$,	$E_t \sim \mathcal{N}(0,1)$ i.i.d., $Y_0 = 0$
`Y3`:	$Y_t = 0.5 \cdot Y_{t-1} + E_t$,	$E_t \sim \mathcal{N}(0,1)$ i.i.d., $Y_0 = 0$
`Y4`:	$Y_t = Y_{t-1} \cdot E_t$,	$E_t \sim \mathcal{U}(0.95, 1.05)$ i.i.d., $Y_0 = 1$

Use computations or a time series plot to decide whether or not these processes are stationary.
If a process is not stationary, suggest a simple method to make it stationary, and try this method out in R .

R-Hint:
`zeitsim[,1]` is the first time series, Y_1.

Note:
The solution sheet contains the relevant commands for simulation using R. For those interested in the mathematical details, there are also some theoretical remarks on these 4 processes to be found there.

Exercise hour: Monday, February 24.