I. Linear filtering of time series data performs local, non-parametric smoothing as a way to estimate trends. (pp. 16-35 in Falk or 6.2 in Hyndman)

a. This method does not deal well with long term periodicity. Assume $Y_t = T_t + S_t + R_t$

b. The goal is to estimate the trend and seasonality with $\hat{T}_t$ and $\hat{S}_t$, then subtract these from $Y_t$ to get a new de-trended time series for further analysis.

c. A linear filter can be written as a set of $r+s+1$ arbitrary real numbers (also called weights), $a_i$, where $i$ is an integer from $-r$ to $+s$. The filter is combined with the time series data, say $Y_t$, to produce a new time series, say $Y^*$, as a linear combination of $a$ and $Y$ like this:

$$Y^*_t = \sum_{u=-r}^{s} a_u Y_{t+u} \quad t = s + 1, \ldots, n - r$$

d. Note that the vector of $Y^*_t$ values will be $r+s$ time periods shorter than $Y_t$.

e. The filter process can be thought of as a sliding, re-weighting, window of \textit{reverse(a)}:

E.g., $r=2$, $s=3$, $a=[0.2, 0.1, 0.3, 0.2, 0.1, 0.3, 0.2, 0.1]$ \hspace{1cm} $Y=[Y_1=4, 5, 7, 6, 8, 5, 2, 4, Y_9=7]$

Since $Y$ is of length 9 and $r+s=5$, the length of $Y^*$ is 4.

\begin{tabular}{cccccccc}
$Y_1$ & $Y_2$ & $Y_3$ & $Y_4$ & $Y_5$ & $Y_6$ & $Y_7$ & $Y_8$ & $Y_9$ \\
4 & 5 & 7 & 6 & 8 & 5 & 2 & 4 & 7 \\
\end{tabular}

$a$: 0.1 0.1 0.2 \hspace{0.5cm} \textbf{0.3} \hspace{0.5cm} 0.2 0.1

$Y'_4$ = 0.4+0.5+1.4+1.8+1.6+0.5 = 6.2

$a$: 0.1 0.1 0.2 \hspace{0.5cm} \textbf{0.3} \hspace{0.5cm} 0.2 0.1

$Y'_5$ = 0.5+0.7+1.2+2.4+1.0+0.2 = 6.0

$a$: 0.1 0.1 0.2 \hspace{0.5cm} \textbf{0.3} \hspace{0.5cm} 0.2 0.1

$Y'_6$ = 0.7+0.6+1.6+1.5+0.4+0.4 = 5.2

$a$: 0.1 0.1 0.2 \hspace{0.5cm} \textbf{0.3} \hspace{0.5cm} 0.2 0.1

$Y'_7$ = 0.6+0.8+1.0+0.6+0.8+0.7 = 4.5
f. The number of weights is called the order of the linear filter.

g. A linear filter whose weights add to one is called a moving average filter. One common form of moving average filter is made of symmetric equal weights, e.g., \( r=s=2, a=[1/5, 1/5, 1/5, 1/5, 1/5] \). Relatively short moving average filters, whether rectangular or diamond-shaped are called low-pass filters, because they remove rapid (high frequency) short term variations, leaving the “low frequency” and long-term trend components. What is the effect of increasing \( r \) and \( s \)?

II. Classical decomposition of time series

a. Apply a filter of width equal to the period (m) to the time series to get the trend estimate. Use \( \text{rep}(1/m, m) \) as the filter if \( m \) is odd or \( c(1/(2*m), \text{rep}(1/m, m-1), 1/(2*m)) \) if \( m \) is even

b. Perform detrending by subtracting the trend estimate from the data.

c. Find the seasonal “figure” by averaging the detrended value for each of the \( m \) time periods.

d. Find the residual component \( (R_t) \) by subtracting both the trend estimate \( (T_t) \) and the (appropriately replicated) figure \( (S_t) \) from the original data \( (y_t) \).

e. In R, \texttt{decompose(x)} does moving average seasonal and trend decomposition.

III. Example: Unemployed Females in the US 1960-1986

```r
FILENAME fem "&wd/data/female.txt";
TITLE1 'Simple Moving Average of Order 17 for Unemployed Females';
/* Read in the data and generate SAS-formatted date */
DATA data1;
  INFILE fem;
  INPUT upd @@;
  date = INTNX('month', '01jan61'd, _N_-1);
  FORMAT date yymon.;
RUN;
/* Compute the simple moving averages of order 17 */
/* with extension to estimate the ends, too. */
/* Note: METHOD defaults to cubic spline interpolation */
PROC EXPAND DATA=data1 OUT=data2 METHOD=NONE;
  ID date;
  CONVERT upd=ma17 / TRANSFORM=(CMOVAVE 17);
RUN;
Obs date ma17 upd
1 1961JAN 384.333 375
2 1961FEB 385.800 384
3 1961MAR 384.909 383 ...
/* Plot the data together with the simple moving average */
PROC SGPLOT DATA=data2;
  SERIES Y=upd X=date;
  SERIES Y=ma17 X=date / LINEATTRS=(COLOR='red');
  XAXIS MIN=0;
RUN;
```
f1 = ts(scan("data/female.txt", quiet=TRUE), start=1961, frequency=12)
f17 = filter(f1, filter=rev(rep(1/17,17)))
plot(f1, ylab="Female Unemployment"); lines(f17, col=2)

IV. The “Census X11 Program”

a. This is a result of a project of the US Census Bureau from the 1950s to automate and systematize the process of seasonal adjustment of economic data.
b. It is fully implemented in SAS, as is its successor X12 which incorporates a regression ARIMA model. X12 also adds ODS GRAPHICS.
c. Outline
   i. Compute a moving average of length 12 (to average over the seasonality) which is the initial trend estimate.
   ii. Compute original minus trend, which estimates S_t+R_t
   iii. Estimate the seasonal trend component of each month separately using the previous and following two years for that month and filter weights of (1,2,3,2,1)/9. Equivalently, compute the ordinary average of the Y_t-24,Y_t-12,Y_t and of Y_{t-12},Y_t,Y_{t+12} and of Y_t,Y_{t+12},Y_{t+24} and then average them.
   iv. Adjust the seasonal component by subtracting its mean.
   v. Subtract the seasonal trend from the original series to get an estimate of T_t+R_t.
   vi. Smooth with a Henderson filter. These filters (of which you can choose among orders 9, 13 or 23 in X11) are similar to loess smoothing and give a trend estimate.
   vii. Subtract the step vi trend estimate from the original data to get a new estimate of S_t+R_t.
   viii. Compute the seasonal trend for each month using three prior and subsequent months with the filter (1,2,3,3,2,1)/15, which is equivalent to the simple average of three simple order 5 averages.
   ix. Center the new seasonal estimates as in step iv.
x. Report the seasonally adjusted data as the original data minus the final seasonal estimates.

xi. Note: end corrections are needed for data at the start and end of the series.

xii. Note: the method essentially uses 7 years of data around any data point, corresponding to a typical business cycle in economics.

FILENAME un1 "$wd/data/unemployed1.txt";
TITLE1 'Original and X11 seasonal adjusted data';
TITLE2 'Unemployed1 Data';

/* Read in the data and generate SAS-formatted date */
DATA unemp1;
  INFILE un1;
  INPUT month $ t upd;
  date=INTNX('MONTH', '01jul75'd, _N_ -1);
  FORMAT date YYMON.;
RUN;

/* Simple seasonal adjustment */
PROC TIMESERIES DATA=unemp1 OUT=series SEASONALITY=12
  OUTDECOMP=deseason;
  VAR upd;
  DECOMP /MODE=ADD;
RUN;

/* Apply X-11-Program */
PROC X11 DATA=unemp1;
  MONTHLY DATE=date ADDITIVE;
  VAR upd;
  OUTPUT OUT=unemp2 B1=upd D11=updx11;
RUN;
DATA unemp2;
  MERGE unemp2 deseason(KEEP=SA);
  LABEL SA="Simple Seasonal Adjustment";
RUN;

/* Plot data and adjusted data */
PROC SGPLOT DATA=unemp2;
  SCATTER Y=upd X=date;
  SERIES Y=updx11 X=date / LINEATTRS=(COLOR="RED");
  SERIES Y=SA X=date / LINEATTRS=(COLOR="GREEN");
RUN;
In R, use package “X12”.

V. Local polynomial smoothing (loess)

a. The stl() function in R is a version of decompose() that uses loess.

b. In SAS, use, e.g., PROC UCM (unobserved components model).

c. Quote from Hyndman (section 6.5):

STL has several advantages over the classical decomposition method and X-12-ARIMA:

- Unlike X-12-ARIMA, STL will handle any type of seasonality, not only monthly and quarterly data.
- The seasonal component is allowed to change over time, and the rate of change can be controlled by the user.
- The smoothness of the trend-cycle can also be controlled by the user.
- It can be robust to outliers (i.e., the user can specify a robust decomposition). So occasional unusual observations will not affect the estimates of the trend-cycle and seasonal components. They will, however, affect the remainder component.

On the other hand, STL has some disadvantages. In particular, it does not automatically handle trading day or calendar variation, and it only provides facilities for additive decompositions.

It is possible to obtain a multiplicative decomposition by first taking logs of the data, and then back-transforming the components. Decompositions somewhere between additive and multiplicative can be obtained using a Box-Cox transformation of the data with $0<\lambda<1$. A value of $\lambda=0$ corresponds to the multiplicative decomposition while $\lambda=1$ is equivalent to an additive decomposition.
d. **STL Example:**

```
library(fpp) # Hyndman’s data library
fit <- stl(elecequip, t.window=15, s.window="periodic",
          robust=TRUE)
plot(fit)
```

```
TITLE1 'Electricity Data';
/* Read in the data, compute moving average of length as 12
   as well as first and second order differences */
DATA data1(KEEP=year sum delta1 delta2);
   INFILE 'data/electric.txt';
   INPUT year t jan feb mar apr may jun jul aug sep oct nov dec;
   sum=jan+feb+mar+apr+may+jun+jul+aug+sep+oct+nov+dec;
   delta1=DIF(sum);
   delta2=DIF(delta1);
RUN;
```
TITLE2 'Raw data';
PROC SGLOT DATA=data1;
  SERIES Y=sum X=year;
RUN;

TITLE2 'First order differences';
PROC SGLOT DATA=data1;
  SERIES Y=deltal X=year;
RUN;
TITLE2 'Second order differences';
PROC SGLOT DATA=data1;
   SERIES Y=delta2 X=year;
   REFLINE 0 / AXIS=Y;
RUN;

VII. Next: Exponential smoothing