I. Loading and querying time series data in R and SAS
   a. R
      i. `scan(file="myfile.ext", what=double(), skip=0, sep="", quote="'", `quite`=FALSE)` reads any number of values per line into a vector
      ii. `read.table()` and `read.csv()` read rectangular data into data.frames.
      iii. `ts(data=NULL, start=1, end=numeric(), frequency=1)` converts a vector or data.frame column to a class “ts” object. When applied to a matrix or whole (numeric) data.frame it creates an “mts” multiple time series object (column-wise, with nice labels in the data.frame). Typical use is `ts(x, start=1960.5, frequency=12)` for monthly data starting in July 1960, with ‘end’ computed automatically.
      iv. `tsp(x)` when ‘x’ is a “ts” or “mts” object returns the ‘start’, ‘end’, and ‘frequency’ of the time series. Months, leap years, second in minutes, minutes in hours, etc. are not dealt with. `tsp(y) = tsp(x)` is useful for copying the tsp attributes to another series, e.g., in a “forecast” object from the “forecast” package, the $mean$ element is a “ts” with tsp relating to the desired future period, but the corresponding “lower” and “upper” prediction interval limits are not. The tsp attributes need to be added if you want to plot with them
      v. `plot.ts()` plots “ts” or “mts” objects using the time info on the x-axis
      vi. `acf(x, lag.max = NULL, type = c("correlation", "covariance", "partial"), plot = TRUE)` plots (or just computes) the acf of a numeric vector or a time series. For the latter lag=1 corresponds to “frequency”. For an “mts” object, acf() plots the acf of each series plus the cross-correlations.
      vii. `Acf()` in “forecast” suppresses the spike at lag=0.
      viii. `tsdisplay()` in “forecast” plot the series, acf and pacf.
   b. SAS
      i. INFORMATs such as MMDDYY10. (for 12/22/2014 or 3-5-2014) and MONYY8. (for Jan 2015 or Feb-2014) convert text file string dates into SAS dates which are just numeric values indicating days since Jan. 1, 1970. (See http://support.sas.com/documentation/cdl/en/leforinforref/64790/HTML/default/viewer.htm#n0verk17pchh4vn1akrrv0b5w3r0.htm.)
      ii. FORMATs such as DATE11. (for 12-Jan-2015) or MMDDYY. (for 1/12/2015) control how the data value appears in text, tables, and plots. A FORMAT statement in a DATA step sets the default. A FORMAT statement in any PROC overrides the default. (See
iii. SAS fully understands months, leap years, second in minutes, minutes in hours, etc. But see [http://www.sascommunity.org/wiki/Dates,_times,_and_datetimes_in_SAS](http://www.sascommunity.org/wiki/Dates,_times,_and_datetimes_in_SAS) for a discussion of historic data, where dates differ by between 10 and 13 days between countries over the years 1582 and 1929.

iv. Function INTCK(period, startdateortime, enddateortime, <method>) computes the interval between two date, time, or datetime values.

1. ‘period’ as a character variable, or constant, that describes the basic period and its start point offset (e.g. YEAR, QTR, or WEEK or more complex multi-intervals, see [http://support.sas.com/documentation/cdl/en/lefunctionsref/67398/HTML/default/viewer.htm#p0syn64amroombn14vrdksh459w.htm#n1w bckrlffgb8jn15eebgjpqm23c](http://support.sas.com/documentation/cdl/en/lefunctionsref/67398/HTML/default/viewer.htm#p0syn64amroombn14vrdksh459w.htm#n1w bckrlffgb8jn15eebgjpqm23c)).
2. ‘startdateortime’ and ‘enddateortime’ are date, time or datetime variables, (or constants like ’02SEP2013’D), that are used as the starting and ending points to calculate the number of periods. The TODAY() function is allowed.
3. ‘method’ is an optional character constant, enclosed in quotation marks, that specifies if counting is "CONTINUOUS" or "DISCRETE". If omitted the "DISCRETE" method is used. DISCRETE just counts the number of interval starts that are crossed.

v. Function INTNX(interval <multiple><.shift-index>, start-from, increment <, 'alignment'>) computes a new date, time or datetime value is a certain interval after (or before) the “start-from” value.

1. ‘interval’ is the ‘period’ as for INTCK (and optionally a multiple, so, e.g., YEAR2 specifies biannual periods). You can, e.g., use YEAR.3 to make the beginning of the time interval be March 1 instead of January 1 when ‘alignment’ is ‘B’, ‘M’, or ‘E’.
2. ‘start-from’ is as for INTCK
3. ‘increment’ is the number of periods to move (- is backwards)
4. Optional ‘alignment’ specifies if the answer is at the ‘BEGINNING’ of the period (default), ‘MIDDLE’, ‘END’, or ‘SAME’. Single letter abbreviations are OK.
5. **Example:**

```plaintext
DATA test;
  informat d1 MMDDYY10.;
  input d1 @@;
  ds = INTNX('YEAR', d1, 61, 'S');
  dm = INTNX('YEAR', d1, 61, 'm');
  db = INTNX('YEAR', d1, 61);
  format d1 ds dm db date11.;
  datalines;
  11-23-1954 08-01-1953 02-29-1956
run;
```

<table>
<thead>
<tr>
<th>d1</th>
<th>ds</th>
<th>dm</th>
<th>db</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-AUG-1953</td>
<td>01-AUG-2014</td>
<td>02-JUL-2014</td>
<td>01-JAN-2014</td>
</tr>
</tbody>
</table>

vi. **PROC ARIMA is the easiest way to get plots with ACF and PACF.**

```plaintext
PROC ARIMA DATA=myLib.myData;
  IDENTIFY VAR=myVar1 CENTER;
RUN;
  IDENTIFY VAR=myVar2;
RUN;
  IDENTIFY VAR=myVar2(1); /* first difference */
RUN;
QUIT;
```

II. **The Additive Model for a Time Series (pp. 1-13 in Falk TS book)**

a. \[ Y_t = T_t + Z_t + S_t + R_t, \text{ t}=1,...,n \]
b. \( T_t \) is a (possible) long term trend

c. \( Z_t \) is a (possible) long term cyclic component, such as a business cycle

d. \( S_t \) is a (possible) short term periodic component, such as yearly climate effects

e. \( R_t \) is residual “noise” or error, usually not i.i.d.
III. Definition of Durbin-Watson Statistic:

\[ d = \frac{\sum_{t=2}^{T} (e_t - e_{t-1})^2}{\sum_{t=1}^{T} e_t^2} \]

where \( T \) is the number of time points, and \( e_t \) are the residuals.

This statistic is approximately equal to \( 2(1-r) \), where \( r \) is the lag 1 autocorrelation. So we expect near 2 for no autocorrelation, near 0 for strong positive autocorrelation, and near 4 for strong negative autocorrelation.

To test for positive autocorrelation look up the values of \( d_L \) and \( d_U \) for “\( T \)” equal to the total number of data points and “\( K \)” equal to the number of regression parameters. Values less than \( d_L \) are good evidence of positive autocorrelation and values greater than \( d_U \) are considered good evidence of lack of positive autocorrelation. Value between the \( d_L \) and \( d_U \) cutoffs are considered inconclusive (colloquially, in the “grey zone”). For \( T \) below about 100, \( d_L \) is between 0.2 and 1.0, and \( d_U \) is between 1.3 and 3.0. (Bigger “\( K \)” and smaller “\( T \)” correspond to a wider “grey zone”.) When tables are not readily available, a rule-of-thumb is that values less than 1 are troublesome.

For testing negative autocorrelation, compute \( 4-d \) and compare to the same critical values of \( d_L \) and \( d_U \).

SAS GLM results for the unemployment regression:

<table>
<thead>
<tr>
<th>Sum of Residuals</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum of Squared Residuals</td>
<td>1314861464</td>
</tr>
<tr>
<td>Sum of Squared Residuals - Error SS</td>
<td>0</td>
</tr>
<tr>
<td>First Order Autocorrelation</td>
<td>1</td>
</tr>
<tr>
<td>Durbin-Watson D</td>
<td>0</td>
</tr>
</tbody>
</table>

\( AC(1)=1 \) and \( D=0 \) are suspicious – what could be wrong?

/* From: http://www.stanford.edu/~clint/bench/dwcrit.htm
   T   K   dL   dU
   51  13  1.03319  2.15258
   Since D=0.45 < 1.03, there is evidence of positive autocorrelation.
   (If 1st order AC were negative, we would look at 4 - 0.45 = 3.55 to check for evidence of negative autocorrelation.) */

IV. R Examples: in LNTS2.R
V. **SAS Examples: in LNTS2.sas**

Example of an equation for trend by a theoretical curve:
Three parameter logistic curve:

\[ y = f(t) = \frac{\beta_3}{(1 + \beta_2 \exp(-\beta_1 t))} \]

How do we approach an equation like this?

VI. **Conclusion:** EDA is critical because there is no “one size fits all” approach