36-315: Statistical Graphics and Visualization

Lab 7

Date: February 25, 2003 **Due: end of lab**

Interspersed throughout this lab are some useful thought questions. You will be asked about them at check-off.

1. Download the files for this lab from the course web page.
2. Open a Word document to record your work.

Start R

3. `Start -> Programs -> Class software -> R 1.5.1`
4. Set the working directory to `My Documents`:

   ```
   File -> Change dir...
   ```

5. Load the special functions for this lab:

   ```
   source("lab7.r")
   ```

Load the data

6. `frame = read.csv("lab6.csv")`

   (Yes, it is the same data from lab 6.) You will always be using the Stamford data, so to save typing it helps to rename it as `ozone`:

   ```
   names(frame)[2] = "ozone"
   n = nrow(frame)
   ```

   The variable `n` above is useful for setting spans.

Decompose the data

7. In lab 6, you decomposed the ozone data into a trend and residual. Now you will decompose it into three components: an irregular trend, a periodic oscillation, and noise. The process is illustrated on the last page.
8. First, make a scatterplot of ozone versus day with a prediction line. Use a span of 30/n and give the plot the correct aspect ratio (as for all plots in this lab). The purpose of this plot is to select and verify the span.
9. Second, compute the trend line with this span and add it to the frame along with its residual. This completes the first decomposition.

10. Make a scatterplot of `ozone.residual` with a prediction line, span 14/n.

11. Compute the trend line with this span and add it to the frame along with its residual. This completes the second decomposition.

Cycle plots

12. The goal is to analyze the periodic component (`ozone.residual.smooth`). Start by graphing it with a connected line, in the correct aspect ratio. *Does it look periodic?*

13. Plot the auto-correlation function of this time-series. *How periodic is it, and what is the period (to the nearest integer)?*

14. Return to your connected line graph and superpose a period grid. This is done as follows:

    ```
    period.grid(1,p)
    ```

    where p is your estimated period. The grid allows you to measure the spacing between cycles. *Do the cycles in the data follow your estimated period well? Can you spot regions when the peak spacing changes?*

15. Make a spiral plot using your estimated period. It may help to scale the window, or make the symbols bigger via `cex=3`. *Does this plot help your perception of the cycle variation? Using this plot, find a cycle whose peak came significantly earlier than expected, and a cycle whose peak came significantly later than expected.*

16. Make an image of the cycle matrix using your estimated period. The window size that produces square cells is generally best. *Does this plot help you answer the previous question?*

17. Show us your graphs.

How decomposition works:

`predict.plot` only makes a graph. It does not give you the trend line in numerical form, nor does it compute residuals.

`smooth` computes a trend line and returns it to you. It is called exactly the same way as `predict.plot`, but does not make a plot. The resulting `fit` object can be plotted, using `plot(fit)`, which will look the same as `predict.plot`.

`extend.with.fit(frame,fit)` takes the `fit` object from `smooth`, adds it to the frame, **and** adds a column of residuals to the frame. The names of these columns are determined from the name of the column that was smoothed. Type `frame[1,]` if you are ever in doubt about what columns are in `frame`.