36-315: Statistical Graphics and Visualization

Lab 5
Date: February 12, 2002 **Due: start of class February 18, 2002**

---

1 Introduction

In this lab, you will try out some time series plotting techniques. The lab is interspersed with question marks ("?") that you need to replace with the proper S-PLUS command.

For this lab you'll need the extra functions provided at

`www.stat.cmu.edu/~minka/courses/36-315/code/ts.s`

You can `source` this file into S-PLUS. The data will be the state populations dataset from

`www.stat.cmu.edu/~minka/courses/36-315/data/statepop.txt`

You can load it into S-PLUS with

```
statepop <- read.table("statepop.txt")
```

This is a data frame where columns are states and rows are years. The following commands assume that you have extracted the year names into `x` and the time series for your state into `y`, e.g.

```
x <- as.numeric(dimnames(statepop)[[1]])
y <- statepop[["my.state"]]
```

S-PLUS has turned spaces into dots, so that "West Virginia" is indexed as `"West.Virginia"`. Note the use of double brackets for extracting a vector from a data frame. If you had typed `statepop["my.state"]` you would have gotten a data frame whose one column was "my.state". If your state got started after 1790, it helps to focus on the timeline where it is defined. Fill in the "?" appropriately:

```
i <- !is.na(y)
x <- ?
y <- ?
```

2 Trend and oscillations

Because the time series is short, it should be plotted using lines *and* points:

```
plot(x,y,type="o")
```

Find a transformation of the form $\frac{y^p-1}{p}+1$ which makes the trend linear. Then superpose a smooth curve; as smooth as possible.

```
yp <- lowess(x,y,f=1)$y
lines(x,yp,col=2)
```

Turn in this plot, with proper axis labels. The purpose of this regression line is to model the trend, which you subtract to get a residual time series:

```
r <- ?
plot(x,r,type="o")
rp <- lowess(x,r,f=0.5)$y
lines(x,rp,col=2)
```

Note that you want to use less smoothing here.

> **Question:** It is typical for states to show a population dip during 1940–1950 and boom during 1960–1970. Does your state show this pattern?

3 Aspect ratio

The aspect ratio should generally be chosen to maximize the change in orientation across the time series. "Orientation" means the tangent of the slope. To get an idea of the proper aspect ratio for these plots, use the function `set.aspect`. It computes the slope at each point of the time series and sets the aspect ratio to maximize the change in orientation. As shown in class, this amounts to making the mean orientation equal to 45 degrees. Make a plot, call `set.aspect`, then make the plot again, like so:

```
plot(x,r,type="o")
opar <- set.aspect(x,r)
plot(x,r,type="o")
```

Turn in this plot. The aspect setting will also hold for subsequent plots. To undo the aspect setting, type `par(opar)`.

4 Comparing growth rates

Now you will compare your state to some others. Pick four other states which are geographically close, and make a reduced data frame containing the five states:

```
s <- statepop[c("my.state","other.state.1","other.state.2",?,?)]
```

Note that we are using single brackets this time.

When comparing states, it helps to transform to a standard scale. Instead of absolute population, use the log growth rate. A time series can be turned into a log growth rate series as follows:

```
g <- gradient(log(y))
plot(x,g,type="o")
```

To convert the entire frame into log growth rate, apply this function to each column:

```
rate <- function(x) gradient(log(x))
r <- apply.df(s,rate)
```

To visualize the state curves, you can pursue a strategy of simplification or alternate encodings. One alternate encoding is line thickness. With all time series in a single frame, you can say

```
ts.chart(r)
```

This displays all columns in parallel, using line thickness. If you want to see what it looks like without thickness, say

```
ts.chart(r,thick=F)
```

The curves are probably hard to compare because each state tends to have unusually large growth rate at its inception. To get rid of these effects, make a new frame containing only the years 1900–1990:

```
r2 <- r[?,]
ts.chart(r2)
```

To make the plot easier to read, the states should be ordered so that similar states are together. To reorder a data frame, you can select out columns in a particular order, for example `r2[c(5,4,3,2,1)]` will reverse the order of the states. Fill in a vector below to order your states:

```
ts.chart(r2[?])
```

> **Question:** What is the main difference between the state growth curves?

5 Simplification

The other way to compare multiple time series is simplification. If we simplify each curve into two numbers, we can make a scatterplot. The first number can be the median:

```
mr <- sapply(r2, function(x) median(x,na.rm=T))
```

The second number can be the slope, computed by least-squares:

```
trend <- function(y) {
  x <- 1:length(y)
  coef(lm(y~x))[[2]]
}
sr <- sapply(r2,trend)
```

Now make a scatterplot, with points labeled:

```
plot(?)
text(mr,sr,labels=names(mr))
```

The line corresponding to zero slope is an important reference. You can add it to the plot via `abline`:

```
abline(h=?,lty=3)
```

The line is dotted (`lty=3`) so that it doesn't dominate the plot.

> **Question:** Does the grouping in this plot agree with the one you made in the previous chart? If not, why not? What difference between the states is highlighted now?