STAT 400 Data Analysis September 28, 2005

Today: Continuation: Bimbo Bakery Data Analysis
Seasonal Adjustment - review
Estimating Demand for a particular day of the week.
Optimizing Delivery Schedule for Maximum Profit

Bimbo Bakery Assignment for Wed, Oct 5:

For the day-of-the-week assigned to you, use the seasonally adjusted data to estimate the daily demand function. Then use your estimated demand function, along with actual deliveries, to simulate the profit for the year on your particular day-of-the-week. By modifying your deliveries by various percentages (but constant over the year), select the percentage that maximizes yearly profit, on average. Finally, estimate the annual profit gain from changed deliveries on your day-of-the-week.
Write a paragraph commenting on the strengths and weaknesses of this procedure to advise Bimbo of opportunities for improving profits through delivery policy modification. (Feel free to suggest and use improvements of the programs).

I will send you programs to do most of the work.

Background:

The data set consists of 53 weeks of daily data (Mon-Sat each week): The number of loaves of bread delivered to a particular retail outlet are recorded, as well as the number of loaves that were unsold at the end of the day. (In other words we know the actual sales). Loaves cost $0.50 to make and sell for $1.00. The cost to the bakery of a loaf being unsold on the day it was delivered is $0.25 (as it would be if half the old loaves were sold at half price, and the rest thrown out.) The cost of a loaf being demanded by a customer when it there are none left is $2.00 per customer (since there is a chance the retail outlet may lose the customer.)

The objective is to advise the bakery on whether its delivery amounts are optimal to maximize gross profits to the bakery. For example, if 100 loaves are delivered, and 90 are sold, gross profit is 90*(1.00-0.50) – 10*(0.25) = $42.50. If 100 loaves are delivered but 110 were demanded, the gross profit would be 100*(1.00-0.50)-10*(2.00) = $30.00.

The catch is that we do not know the demand in those cases where the number of unsold loaves is zero. How can we use the data to judge whether the delivery amounts should be increased or decreased?

This is a situation where we need to do some modeling to help fill in the missing data. One way to do this is to guess the demand distribution, simulate the effect it would have over the year of the data using the delivery amounts, and compare the simulated sales distribution with the actual sales distribution. By trial and error, we can fit a demand distribution to the data in this way. Once we have a good fit, we can again run through the year of data increasing or decreasing the delivery amounts by a certain percentage. By computing the gross profit for several percentage changes in the delivery amount, we can estimate the optimal percentage change (that maximizes gross profit), and recommend this to the bakery.

A preliminary problem is to deal with the seasonal component of the data. Actually, with one year of data, we have no way of knowing if it is really "seasonal" but if there is a trend in the data, we need to allow for it – the reason is that we want to consider the 53 Mondays (or any particular day) as if they were independent. If there were a time-trend over the series of Mondays, this independence would not hold. For simplicity of description, let's call this trend seasonal. A Monday in January would differ from a Monday in July by not only random error, but also by this seasonal trend. The way to do this is to fit the trend, and subtract it from the data, but add back in the yearly average so the numbers look like the seasonally adjusted numbers.

The data to smooth could be sales or deliveries. Since sales are truncated by deliveries, I think it is best to use deliveries as the seasonal indicator. Then adjust the sales by the same proportion that the deliveries are adjusted. (Have you a better idea?). Once you have the sales and deliveries adjusted for seasonality, you can use the modeling ideas mentioned above.

Again, the steps are:

Guess the demand distribution for your day.

Simulate 53 values from this distribution.

Use these 53 values along with the 53 (actual adjusted) deliveries to compute what the simulated sales (SS) would be on each of these 53 days. Then compare this SS distribution with the distribution of the actual observed adjusted sales. A close match means you have a good guess of the demand distribution. A poor match means you should try some different parameters in your guessed model or perhaps even a different model. This comparison can be done with a Q-Q plot or overlaying two ecdf plots.

I'll give you some programs to help with this via e-mail and these notes.

Of course, this is just the first step. But the next step is a bit easier. It is to use your demand distribution to assess the annual profit for the company (for this product at this retail outlet). Then by re-running this analysis with the deliveries modified by a ratio r (eg. .95 or 1.05) you can see how to choose r to maximize the profit. A graph is a good way to do this.

Programs:

```
> bimbo.wed.plot
function (m,s,...)
{
	op=par(mfrow=c(3,1))
	x=rnorm(53,m,s)
	my.dotplot(wd,main="Wednesday deliveries",xlim.min=40,xlim.max=200 )
	my.dotplot(ws,main="Wednesday sales",xlim.min=40,xlim.max=200)
	my.dotplot(x,main="Guessed Wed. demand distribution
",xlim.min=40,xlim.max=200)
	par(op)
}

>bimbo.test
function (m,s,xs,xd)
{
	xs.r=rnorm(53,m,s)
	a=xs.r[xs.r<=xd]
	b=xd[xs.r>xd]
	xs.s=sort(c(a,b))
	steps=(1:53)/53
	plot(steps,xs,type="l",col="red",main="sim(red) vs true(blue)")
	xs=sort(xs)
	lines(steps,xs,col="blue")
	a=loess(steps~xs,span=.25)
	b=loess(steps~xs.s,span=.25)
	c=abs(a$fitted-b$fitted)
	invisible(c)
	}

> bimbo.profit
function (m,s,xd)
{
	potential=round(rnorm(53,m,s))
	sales=pmin(potential,xd)
	unmet=potential-sales
	leftover=xd-sales
	profit=.50*sales-2*unmet-.25*leftover
	return(sum(profit))
}
```

```
>bimbo.profit.run
function (m,s,xd,f)
{
	a=0
	#
	dels=round(xd*f)
	for (i in 1:1000) a=bimbo.profit(m,s,dels)+a
	b=a/1000
	return(b)
}

>bimbo.profit.graph
function (m,s,xd)
{
	b=0
	c=1+(0:20)/40
	for (i in c) {a=bimbo.profit.run(m,s,xd,i);b=c(b,a)}
	b=b[-1]
	d=100*(c-1)
	plot(d,b,typ="l",xlab="Percent increase in Demand",ylab="Annual Profit This 
Day of the Week",main="Profit Outcome to Increased Deliveries")
	e=loess(b~d)
	lines(d,e$fitted,col="red",lwd=5)
	max=max(b)
	return(list(b[1],max,(max-b[1])))
}
```