**Analysis by Simulation and Graphics – some examples**

**This expository talk is intended to demonstrate the utility of simulation and graphics for four selected situations. The first illustrates the power of smoothing, the second the validity of the bootstrap, the third the power of creative use of simulation and graphics to amplify the utility of meagre data, and the fourth to reveal a surprising property of panel reviews.**

**The applications discussed are respectively, gasoline consumption, BMI, bakery deliveries, and student application reviews. The combined message of these examples is to illustrate the utility of some simple and non-standard techniques that could be included in undergraduate statistics courses.**

Here is a bit more info about what I would do with these examples. I have R programs for everything …

Gas & smoothing demo – I just show how smoothing reveals the seasonal pattern not otherwise discernible.

SEM-bootstrap - I show that simulation can generate the square root law via simulation

Bimbo – I show how a few brave assumptions can produce valuable information from scanty data

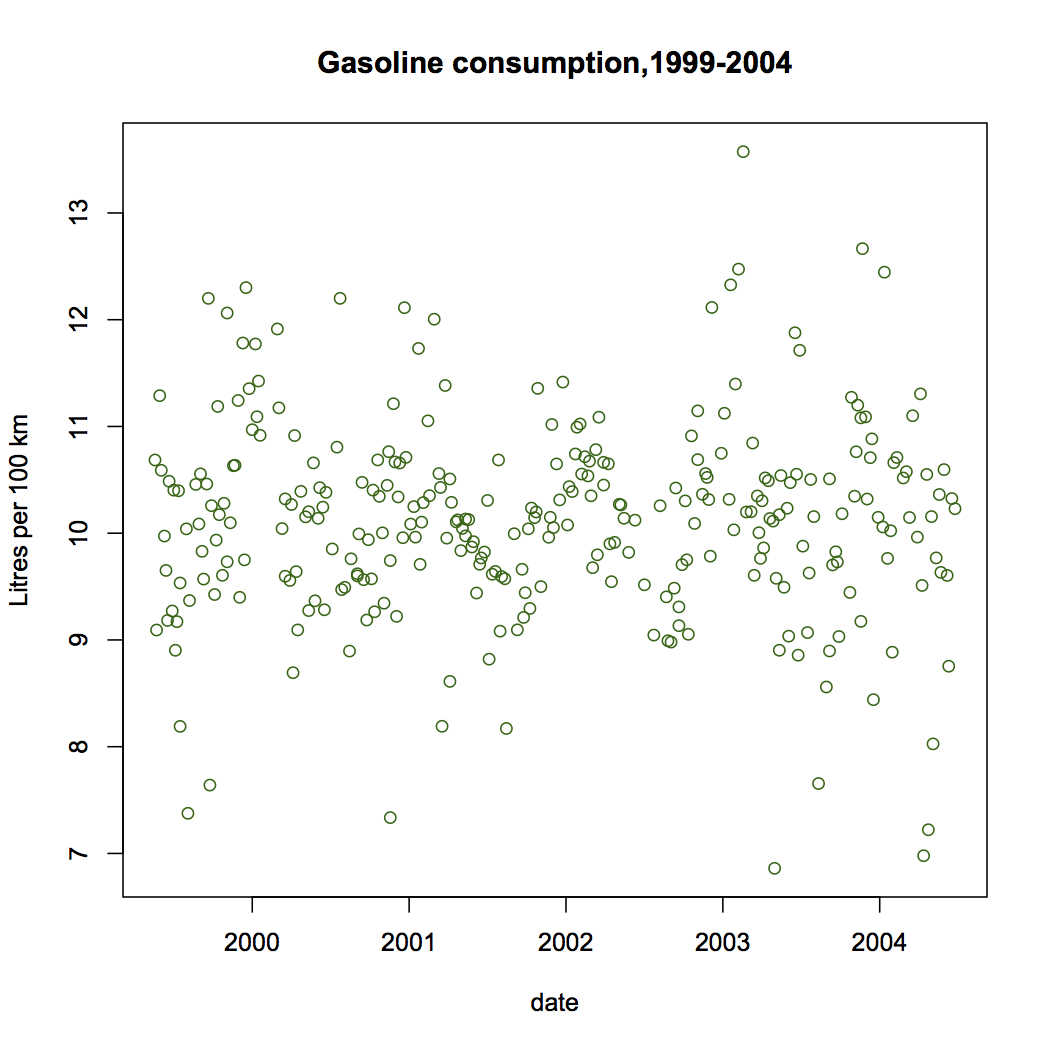
Panel Review – I show that SFUs alternate entry program had a problem in its early days that revealed an inherent weakness of peer review.

**The Power of Smoothing**

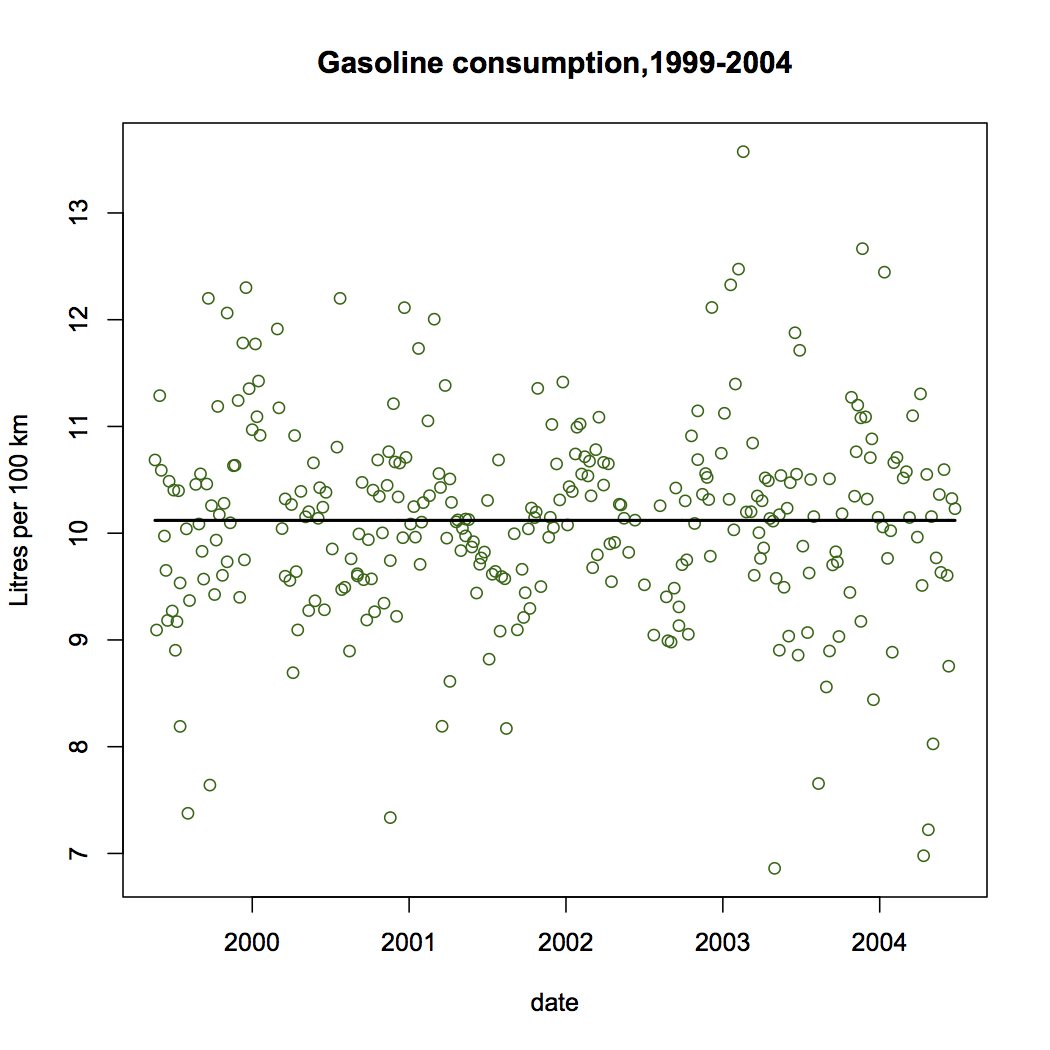
Real statisticians like data, so maybe that is why I collected my gasoline consumption for five years. This involved writing down, and each gasoline refill, the kilometers travelled since the last refill, and the kilometers travelled since the last refill. Here is the data displayed as a time series:

<use gas()>



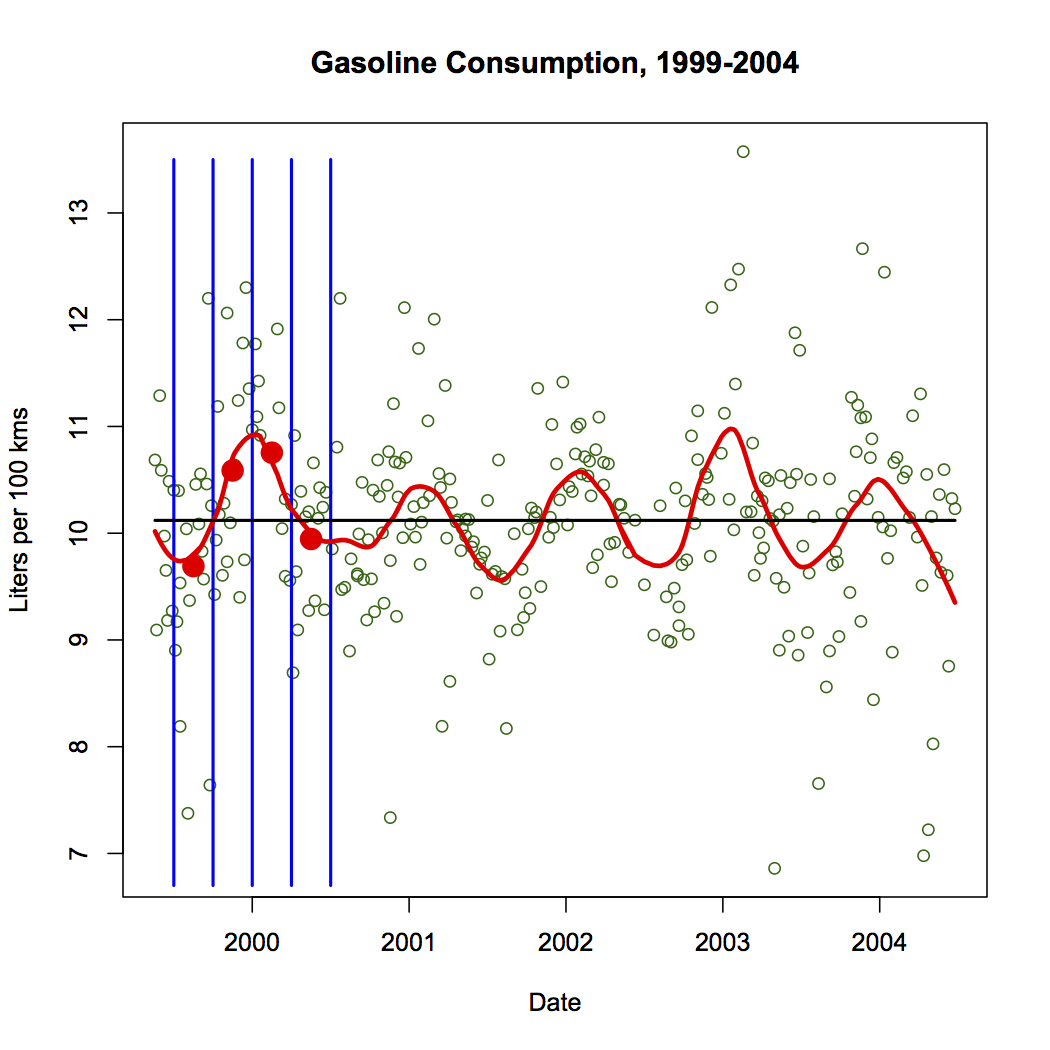


There does not seem to be much trend up or down over time. In fact the regression line would look like this:



You might conclude that the effort to collect this data resulted in a pretty boring result! But note that there is quite a bit of variability in the data, and we would like to explain at least some of that. Actually, the driving situation did not change much over time: I was commuting from Tsawwassen to and from SFU each weekday, and not much other driving with this car. So the program to discover what affects gas consumption is likely to be a dead end. The variability of the gas consumption rate is surprising given the constancy of the driving regime. Perhaps the problem is measurment error. If so, lets do a bit of averaging:

<use gasplot()>



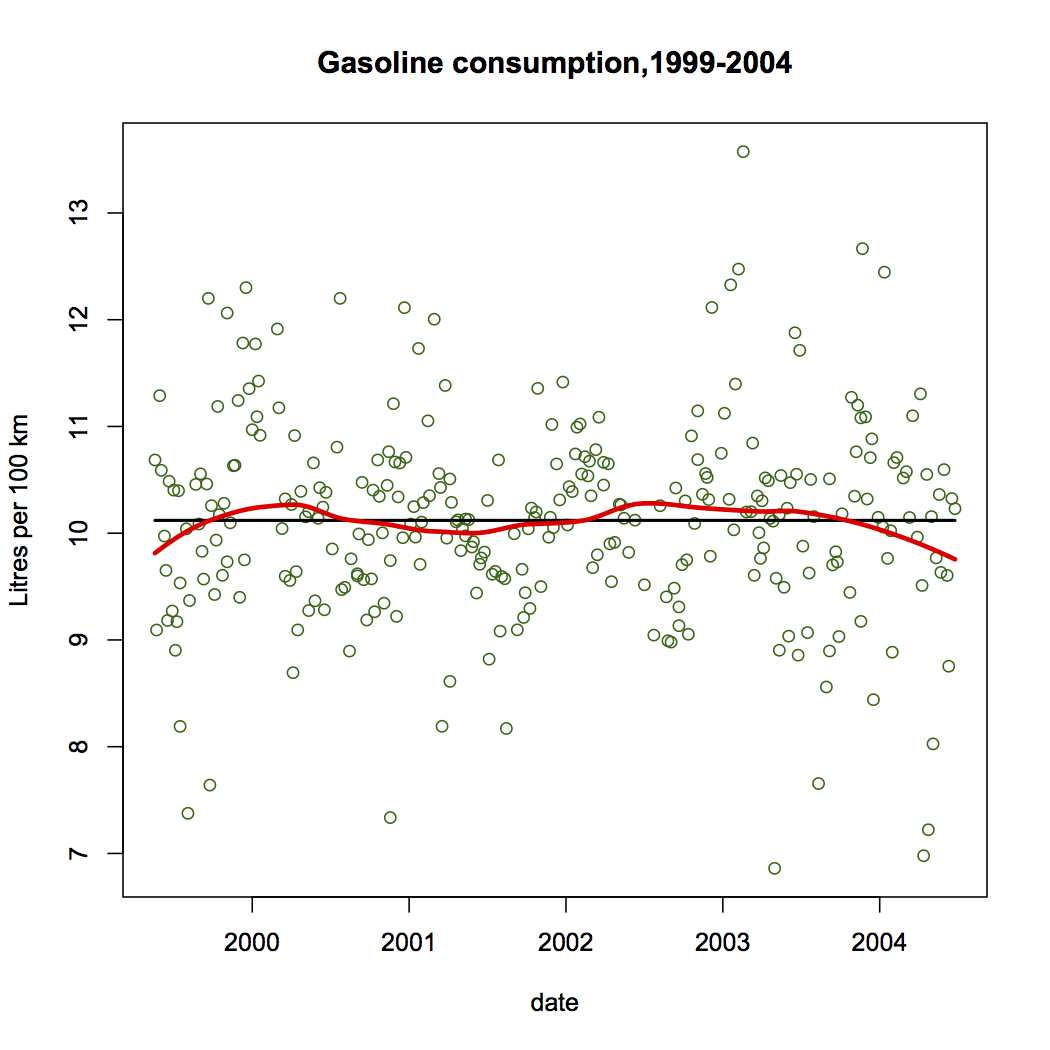
Averaging in vertical strips is the simplest lind of smoothing, and one can see here that it produces a seasonal pattern. Note that the thickness of the strip determines the amount of smoothing applied. Too narrow strips would produce a chaotic pattern like the data itself, while too thick strips would produce an almost flat curve.

<use gas.v.run>

This smooth is not enough:



while this one is over-smoothed:



The criteria that determines the correct amount of smoothing is the believability of features displayed, in the light of general knowledge about the data measurements. That good statistics practice involves subjectivity is a fact that applied statisticians know is true.

There are more sophisticated methods of smoothing, but when simple is good enough, it is better because it is easy to explain to the audience that is interested in the result.

**Confirmation of the Utility of the Bootstrap**

The bootstrap is a method for estimating variability of a statistic based on a single sample (of n data points selected at random from a population). If the statistic is t(x) where x is an n-tuple of data, then the variability of t(x) is estimated by computing the variability of t(x\*), where x\* is a sample selected with replacment from x. Clearly we can sample as many x\* as we need, and as we include more and more x\* in our estimate, the variability estimate will usually stabilize to a single value.

There is one situation where we know what the variability should be, and so let’s see if the bootstrap gives us a good estimate. For a random sample from any population, the variability of a sample mean of n observations from the population will be the boot , where 𝜎 is the population SD. In practice we do not usually know 𝜎 but we estimate it from s, and use s / √n as our estimate of the SD of the sample mean.

Our experiment here is to try to get an estimate close to 𝜎 / √n using only the data in one single sample of n observations, using the bootstrap technique.

use boot.demo.slow()

We start with a population – of course this would not usually be known, but for our experiment, it is known so that we can check the outcome.

Next we select a random sample – and note the sample mean and sample SD.

Often when we take a sample we want to use it to estimate the population mean, but when we do this we also like to know how accurate this estimate of the mean is. The sample SD measure the spread of the sample but what we want is the accuracy of the sample mean as an estimate of the population mean. Now we have theory that tells us to use s / √n to indicate the accuracy of the sample mean but we are going to pretend we do not know this theory, and instead use the bootstrap on this one sample to inform us about the accuracy of the sample mean. Talking re-samples from our original sample, with replacment, produces samples that have many of the same values as were in the original sample, but with varying multiplicity. The result gives an estimate very close to s / √n, which should be surprising!

The following graph shows how this procedure works over several sample sizes, with one experiment at each sample size. Note the bootstrap works as well as s / √n.

use boot.SEM.demo()

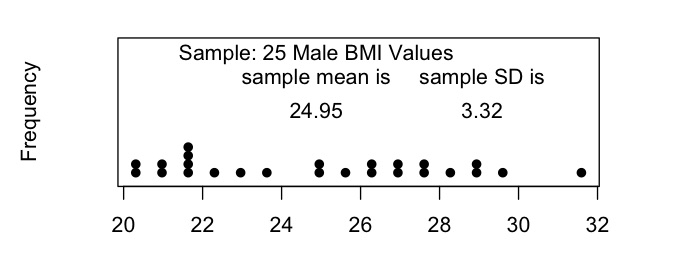
Is there a point here, other than an explanation of the bootstrap? Note that the bootstrap procedure sidestepped the requirement of know the theoretical formula for estimating the variability of a statistic, and suggests that it might work in more complicated situations in which the theoretical approach is not known.

Why does the bootstrap work? The Probability Integral transformation provides a general method for sasmpling from a population defined by the CDF. If we use the ECDF as a proxy for the CDF in this, “random sampling” becomes Bootstrap sampling. So any statistic value computed from a bootstrap sample will approximate the value computed from the actual population sampled. Since the ECDF is the main information we have about the CDF, its use as an estimate is natural and avoids further assumptions – but note the important assumption of “random” sampling,

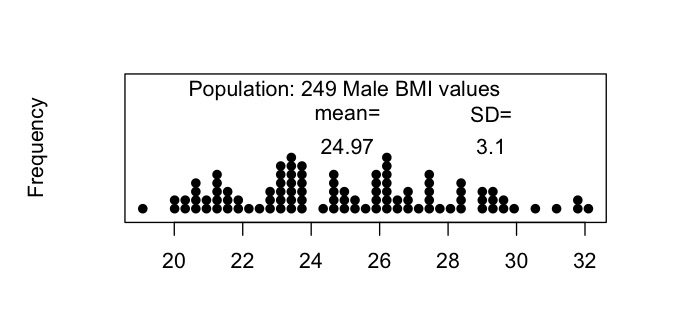
Let’s take a look at a data-based application.

Here is a sample of 25 BMI values for men.

BMI is (weight in kgms)/(height in meters)2 It tries to measure obesity.



Suppose we want to know how well we have estimated the population mean BMI for the population underlying this random sample. We need to estimate the SEM of the population mean and we now have two ways to do that. The square root formula gives 0.66 and the bootstrap method gives .60 for the SEM of the estimated mean. However, in this experiement we do actually know the true SEM based on the population data:



so the population SEM is 3.1/√25 = 0.62. So both estimates (.66 and .60) were very good in this case.

Here is one situation where you may not have an easy formula for the SD of an estimate: Consider a sample of size n=30, a poulation N(0,1), and a statistic such as the 4rth largest value (the 90th percentile). The true sd of the estimated percentile is about .30. The estimates using bootstrap on samples of size 30 produce values like (in 10 separate experiments)

[1] 0.3457481

[1] 0.2321129

[1] 0.4613471

[1] 0.4148128

[1] 0.2928874

[1] 0.165316

[1] 0.2888042

[1] 0.1309033

[1] 0.2617386

[1] 0.2895535

Not perfect , but not bad for a sample of only 30 values – and, no further theory required!

The bootstrap has an important role of providing the precision of estimates in complex or non-standard applications.

**Exploration of Simulation and Graphics for Operations Research**

(The case of the Bimbo Bakery)

Bimbo Group is an global bakery company based in Mexico. I was once sent some data on a tiny piece of the bakery operations, and asked if there were a way to use it to improve operations. The data concerned one retail outlet, one product (a loaf of bread), the deliveries of that product to the outlet each day for a year, and the sales of that product that day. So the main question to address with this data is whether the delivery amounts could be improved: some days the deliveries are too small to meet demand, and some days the deliveries exceed the demand and there is leftover waste. Of course, crucial to this analysis is the cost of unfulfilled demand, and the cost of leftovers. We may not know this but presumably someone from the Bimbo company could provide these costs. Let’s proceed with some guesses for these costs and see whether there is some advice about deliveries that would flow from our analysis. Lets suppose each loaf sold results in a gross profit of $0.50, each leftover results in a loss of $0.25, and each unfulfilled demand results in a loss of $2.00. The idea behind this last parameter is that unfulfilled demand could result in a loss of future sales, perhaps with a small probability.

The data consists of two numbers for each day of the year, except Sundays. The two numbers are: the number of leftovers loaves - L

the number of loaves sold – S

From this we can deduce the number of loaves delivered – D

We also need a symbol for unfulfilled demand – U. Obviously, U is not observed, and this is where we need to use some creative techniques to judge its influence on the outlet’s daily profit.

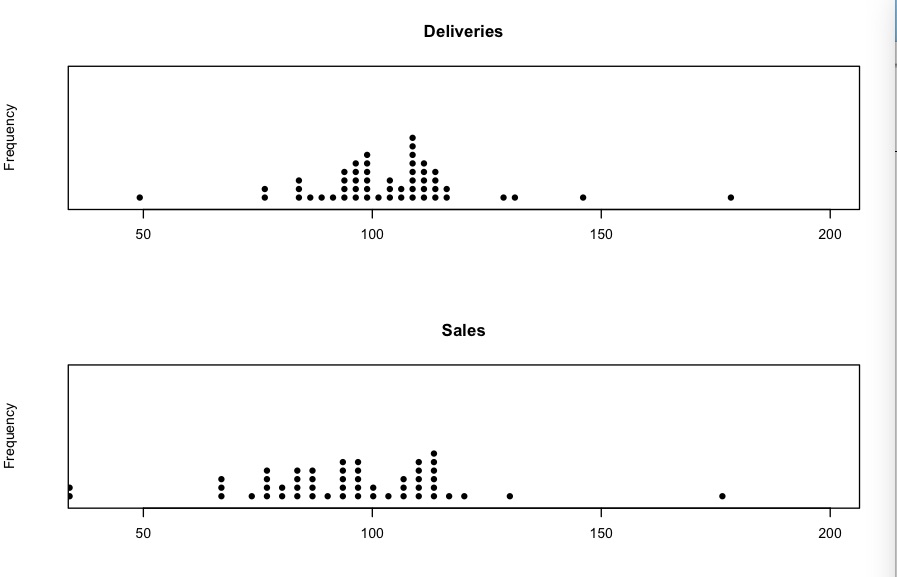
There is a fairly obvious weekly pattern, and so we might simplify the problem by concentrating on one day-of-the-week, say Wednesday. So now we have 53 daily delivery quantities and 53 daily sales quantities. Often the two numbers will be equal indicating that there may have been some unfulfilled demand – in this case the leftovers L = 0. When they are unequal, the deliveries D > the sales. On any particular day, the gross profit is S\*0.50 –L\*0.25 – U\*2.00. Each day, either L=0 or U=0.

If we could estimate the daily demand distribution, then we could run through a year of observed deliveries and by comparing simulated demand with actual deliveries, we could simulate an annual sales distribution. In fact, if we propose a certain distribution as a demand distribution, we can check our proposal by comparing the simulated sales distribution with the actual sales distribution. We will use this method to estimate the actual demand distribution for a particular day of the week – eg. Wednesday.

Lets first look at the Wednesday data:

It is clear that Wednesday deliveries vary considerably over the year, and that demand is also erratic (when it has a chance to show it). Lets look at the distributions of deliveries and demand:

Use bimbo.guess.plots(115,35,xd=wd,xs=ws)



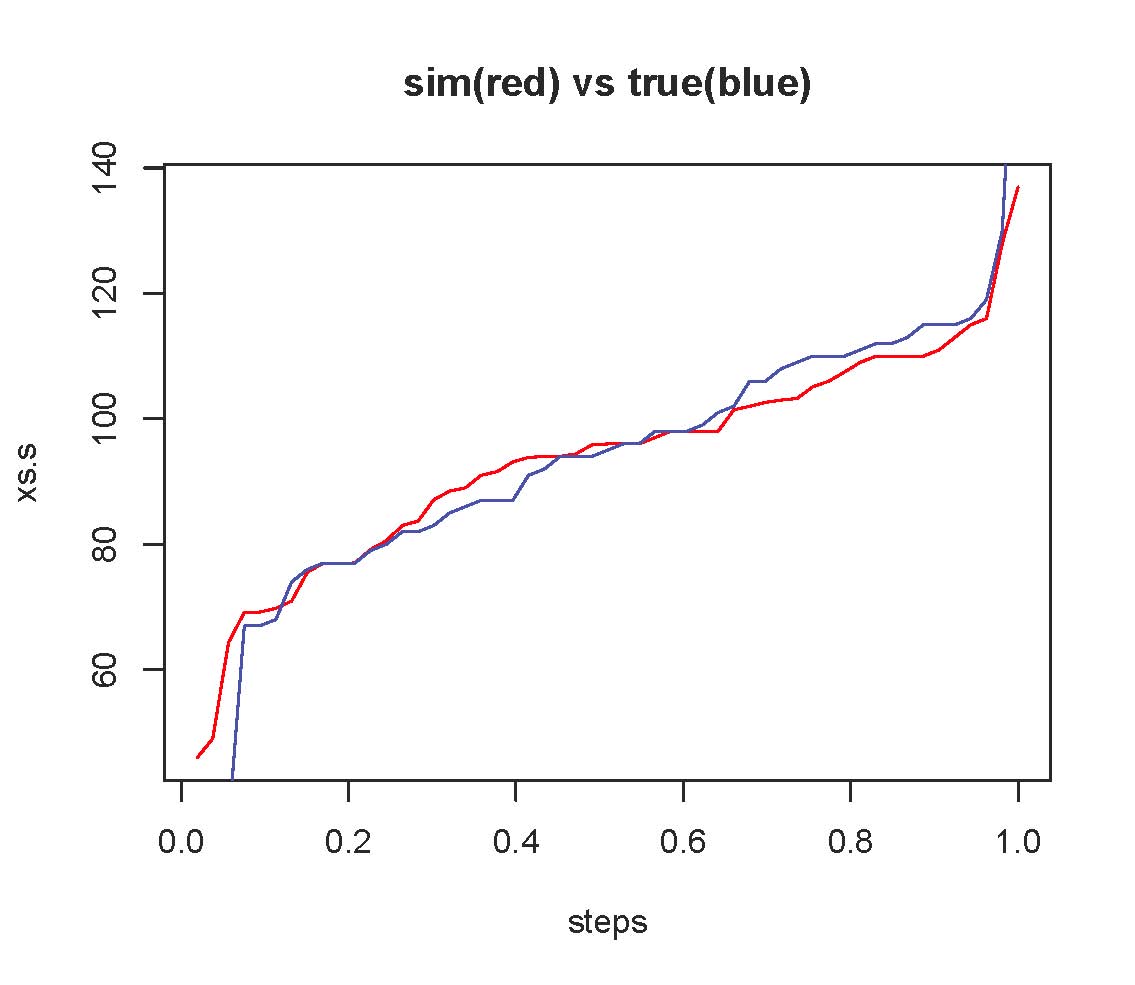
The dotplots suggest that the upper part of the demand distribution is hidden because the sales cannot exceed the deliveries. Our next step is to guess the demand distribution – we can check it by 1. Computing the simulated sales using the simulated demand and the actual deliveries

2. Comparing the simulated sales distribution with the actual sales distribution

<use bimbo.test(m,s,ws,wd) with m=115 and s=35

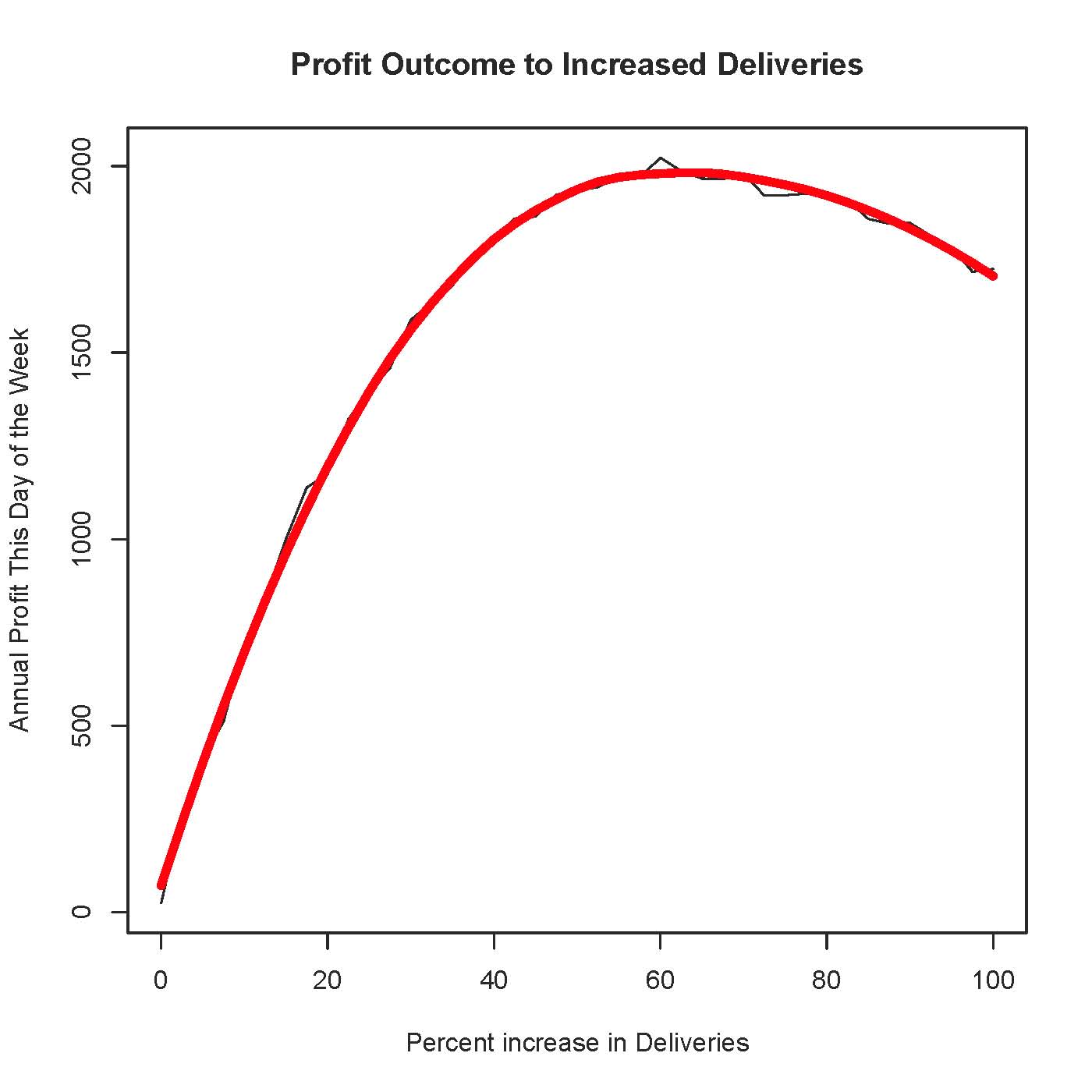
The result by trial and error is that the demand distribution is well-approximated by a N(mean=115, sd = 35) distribution. Here is a typical plot of the ecdfs of the two distributions:

(But try a few bimbo.test(115,35,ws,wd))



So we have found the hidden demand distribution. Now we consider what this implies for advice to the Bimbo delivery plan. Presumably there is someone deciding on the delivery amounts, and local conditions must inform this. But we can see if the delivery amounts tend to be to large or too small by seeing how the profit would change if the delivery amounts were altered by a constant percentage over the year.

Use bimbo.profit.graph(115,35,wd,smooth=F) then with smooth =T



It looks like a 60% increase in delivery amounts will maximize profits, and furthermore, the increase in profit from the previous regime is almost $2000 per annum. (the value of the graph at 0 increase is very small).

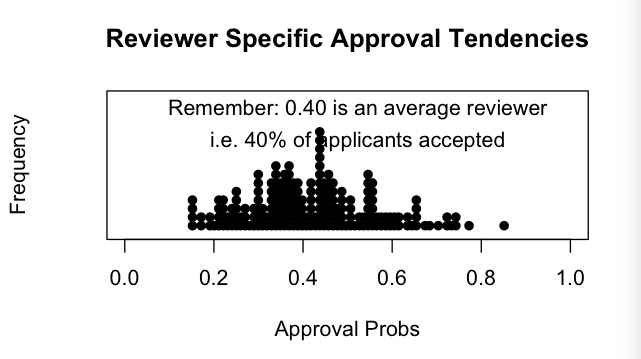
Presumably this procedure, including the trial and error fitting of the demand distribution, could be automated, and perfomed on every product at every retail location, and for every day of the week!

**A Surprising Feature of Reviews-by-Committee**

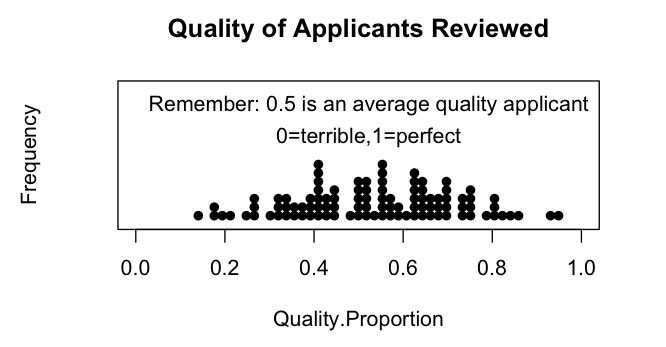
Simon Fraser has a category of student admission called “Diverse Qualifications Admissions”. Students without the normal academic achievement can make their case for admission in writing, and these applications are then reviewed by a committee. In practice, the committee assigns each applicant to two committee members, and the two committee members present their decision to a meeting of the whole committee. Committee members vary in their attitude toward these cases, with some feeling that only the most exceptional stories deserve a positive response, and others tending to accept anyone they think has a reasonable chance of surviving undergraduate work. Consequently, the review style of the particular committee members assigned to a case can make the difference of acceptance or rejection of individual cases they are assigned. We will simulate this process to examine the effect of varying rigor among “referees”.

We suppose that the committee members have acceptance tendencies between about 20% and 80%, with an average of 40%. Naturally, we want to build in the feature that higher quality applicants have a better chance of acceptance, and we do this by using the quality of the applicant to modify the probability of acceptance. We use the distance of the quality from 0.5 to stretch the probability towards 0 or 1. Here is a typical outcome of this simulation:

<use peer.mod >

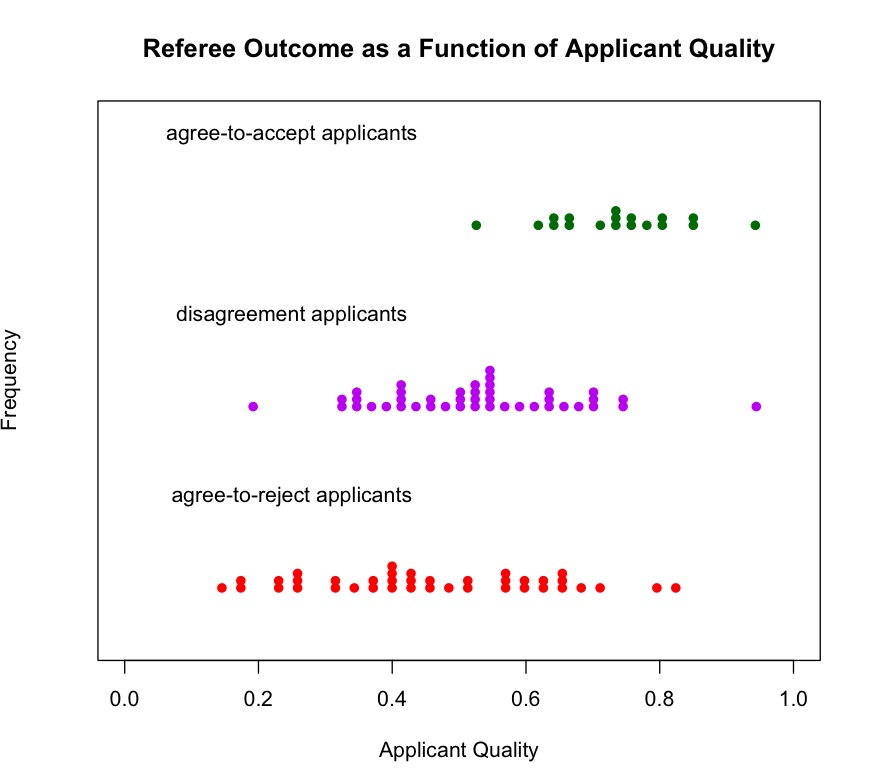


The chart shows that the reviewers have acceptance probabilities that range from about 20% to about 80%. This is what introduces an element of “luck” into the review process.

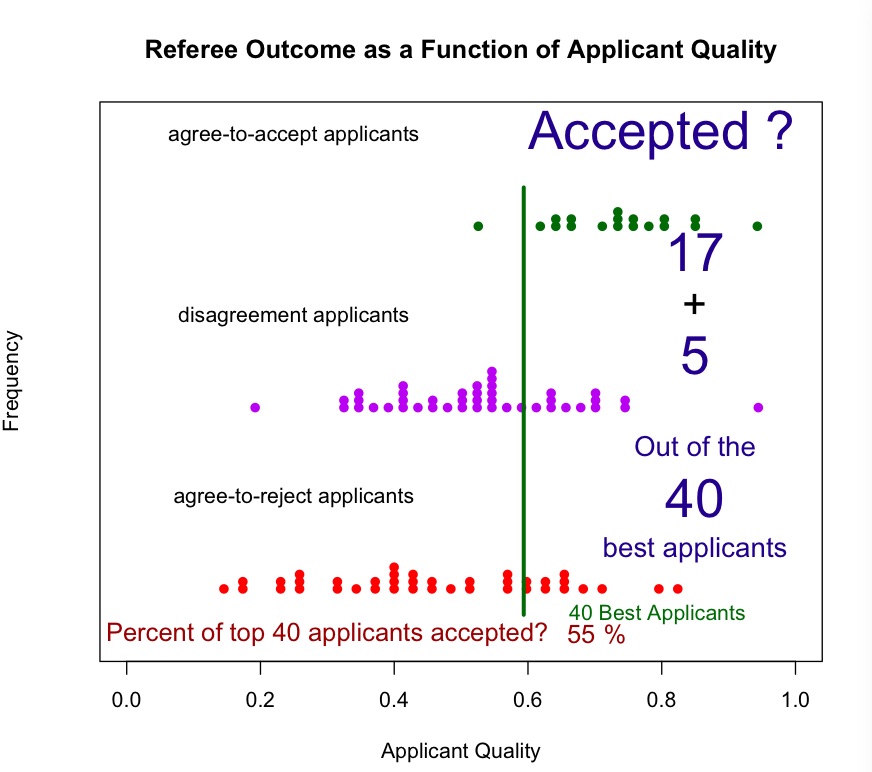


We suppose that the applicants have cases that vary widely in “quality”, as one would expect.

The result of the review process is shown below, where two reviewers determine the the accept/reject outcome if they agree, and if they do not agree, the decision falls to a similar third reviewer.



As expected, accepted applicants tend to be better than rejected applicants, but there is not a high correlation. It turns out that the number published in this simulation is 18 + 19 = 37, so 37% of the candidates are accepted. But many of those rejected are high quality applicants. Another way to look at this is to focus on the top 40% of applicants based on “quality”. Only 55% of these top applicants end up “accepted”. See below:



Of course, this 55% is a simulated result, but it is typical. If the process is repeated many times, it turns out that this percentage averages about 52% with an SD of 8%, and a range of 40% to 65% contains most of the simulated values.

The peer review referee system is considered a “best practice” by many academics. However, on the one hand, it fails to reliably reject poor submissions, and on the other hand, it fails to reliably identify the best submissions.

How valid is this simulation? Does it do a reasonable job of mimicking the true review process. Lets consider a few particular situations. As example 1, suppose an applicant has quality 0.7 (quite good). His referees might have acceptance probabilities of 0.15 and 0.4. The probability used by the program for referee 1 would be .575, and for referee 2 would be .70. Example 2 is an applicant of average quality 0.5. Referee 1 would accept with probability 0.15, and referee 2 would accept with probability 0.4. A third applicant with quality 0.3 would be accepted with probabilties 0.09 and 0.24. The mix of referee rigour and applicant quality is designed to give the applicant quality a larger say than the referee rigour, as might be expected in real life. Even so, the inconsistentcy of referees tendencies is enough to produce a disappointing relationship to the applicant quality. The conclusion is that to use this review system, one must try very hard to standardize the acceptance rates of selected referees for a given applicant quality. In fact, the SFU review process for “diverse qualifications” applicants was modified to ensure referee consistency, once the problem was identified.

Note that this same problem would apply to the review of academic research papers for refereed journals.

(KLW 2017/01/31)