Chapter 8: Lots of data

S520

These notes are written to accompany Trosset chapter 8.

Motivation

I take a simple random sample of 100 IU Bloomington faculty (as of 2014) and look up their salaries online. The data is in the file `faculty100.txt`. We can compute the sample average straightforwardly:

```r
salaries = scan("faculty100.txt")
mean(salaries)
```

```
## [1] 91108.09
```

So the mean of this sample of faculty salaries is exactly $91,108.09. But so what?

Another question we can easily answer: what proportion of the sample have salaries of at least $100,000?

```r
sum(salaries >= 100000)
```

```
## [1] 36
```

When we take a **sample**, the usual point is that we don't just want to draw conclusions about the sample – we want to know about the whole **population**. For example, here we might want to know about the mean salaries of *all* IU Bloomington faculty. We know that the expected value of the sample mean equals the population mean. But to work out to what extent our sample mean is actually useful, we need to answer questions like the following:

1. If we take a large enough sample, are we guaranteed to get the "right" answer?
2. The sample mean is calculated from a random sample, so it's random, so it has a probability distribution. What is that distribution?
3. If we use the sample mean as an **estimate** of the population mean, how big is the error likely to be?

We usually have to rely on theoretical results rather than getting exact answers to most of these questions, because we don't know the whole population. Except in this case we do! The file `faculty.txt` includes salaries for all 2478 IU Bloomington faculty salaries. We can thus get theoretical results and then compare then to what we actually see.

Weak law of large numbers

Trosset ch. 8.2.

It's obvious that if you sample *without* replacement from a finite population, you eventually get the right answer (your sample mean is the same as the population mean) because you eventually sample the entire population. But what if you're sampling with replacement, or if you're sampling from an infinite population? In these cases, you're taking an **independent and identically distributed** (iid) sample.

Let $X_1, \ldots, X_n$ be a sequence of iid random variables with finite mean μ and finite variance σ^2. Choose any positive number ϵ. Then as $n \to \infty$, the probability that the sample mean $\bar{X}_n$ is within ϵ of the population mean μ approaches 1:

$$P(\mu - \epsilon < \bar{X}_n < \mu + \epsilon) \to 1.$$

That is, no matter how small an ϵ you choose, I can choose a big enough n such that the probability that the sample mean is within ϵ of the population mean is arbitrarily close to 1. This is the **Weak Law of Large Numbers**. We say that the sample mean $\bar{X}_n$ **converges in probability** to the population mean μ. (There's also a Strong Law of Large Numbers, which says almost the same thing – the difference is important only to Ph.D. statisticians, and not even to all of them.)

Example. The true mean faculty salary is:

```r
salaries.all = scan("faculty.txt")
mean(salaries.all)
```

```
## [1] 95389.45
```

Suppose we sample with replacement. Can we pick a sample size that's large enough that we have a very high probability of getting a sample mean within a dollar of the population mean?

Let's trying a sample size of a million. To take such a sample, use the `sample()` function in R:

```r
x = sample(salaries.all, size=1e6, replace=TRUE)
mean(x)
```

```
## [1] 95294.54
```

Repeat this a few times:

```r
mean(sample(salaries.all, size=1e6, replace=TRUE))
```

```
## [1] 95364.62
```

```r
mean(sample(salaries.all, size=1e6, replace=TRUE))
```

```
## [1] 95387.05
```

```r
mean(sample(salaries.all, size=1e6, replace=TRUE))
```

```
## [1] 95362.92
```

```r
mean(sample(salaries.all, size=1e6, replace=TRUE))
```

```
## [1] 95389.4
```

It seems that a sample size of a million doesn't guarantee a sample mean within a dollar (or even ten dollars) of the population mean.

Well, let's try a hundred million:

```r
mean(sample(salaries.all, size=1e8, replace=TRUE))
```

```
## [1] 95392.98
```

With reasonable probability, we get within a few bucks of the right answer. Similarly, if we increased the sample size to a trillion, we'd be virtually certain of getting the right answer to within a few cents (you shouldn't try to take repeated samples of size one trillion unless you have terabytes of RAM, however.)

So the bad news is that we require prohibitively large samples to get near-exact results. But the good news we rarely care about near exact results. If we want to know the average salary of IU Bloomington faculty, we don't care about an answer to the nearest dollar – the nearest thousand (or even few thousand) will do.

Law of averages

Trosset p. 187.

A special case of the Law of Large Numbers is the **Law of Averages**. In this case, the iid random variables are Bernoulli – i.e. they only take the values 0 and 1. In this situation, the expected value of any one of the random variables is p, the probability of getting a 1. The sample mean is the proportion of the *sample* that takes the value 1. This **sample proportion** is still denoted as $\bar{X}_n$ in Trosset, while other sources call it $\hat{p}_n$ or just $\hat{p}$. (It's better practice to keep the subscript n to remind yourself the statistic is based on a sample of a certain size, but we'll get lazy.) The Law of Averages says that if the Bernoulli trials are iid, $\hat{p}$ converges (in probability) to p. The key assumption is independence – if the trials are dependent, then convergence might not happen.

Example. I toss a fair coin. Let $1 = X_1 = X_2 = X_3 = \ldots$ if the coin is heads, and let $0 = X_1 = X_2 = X_3 = \ldots$ if the coin is tails. Clearly the sample proportion $\hat{p}$ is going to be either 0 or 1, depending on whether the coin lands tails or heads. It's never going to converge to 1/2, the true value of p. The Law of Averages doesn't apply here because the trials are not independent: once you know X_1, you know all the rest of the random variables.

Example. In American roulette, the wheel has 18 red numbers, 18 black numbers, and 2 green numbers. If you bet c dollars on black, then if the wheel lands on a black number, you get your c dollars back plus another c dollars. If the wheel lands on red or green, the casino keeps your c dollar bet.

At a casino, you notice that at one roulette table, red has come up 10 times in a row. Does the Law of Averages imply that black is due and you should thus bet your life savings on black?

Answer. Please don't do this. We assume spins of the roulette wheel are iid (if they weren't, the casino might lose a lot of money, so they will go to a lot of effort to make sure this is the case.) So every spin has an 18/38 of landing on a black number, regardless of what's happened in the past – the wheel has no memory. Since this is less than half, the casino always has an advantage. The Law of Averages says that sample proportions will approach their true probabilities in the long run. It certainly doesn't say that things will even out in the short run.

The Central Limit Theorem

Trosset chapter 8.3.

We'll start off by stating the greatest theorem in statistics. Then we'll try to understand it, and only then will we try to work out why it's actually useful.

Roughly speaking: The average of any sufficiently large iid sample has an approximately normal distribution.

Formally: Let $X_1, \ldots, X_n$ be a sequence of iid random variables with finite mean and variance. Let

$$Z_n = \frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}}$$

Then as $n \to \infty$, the CDF of Z_n converges to the standard normal CDF.

This justifies approximating the distribution of the average of any sufficiently large iid sample by a normal distribution.

The key idea is that the **sample mean** is approximately normal for large samples. (Not the sample itself, which could have whatever distribution.) We constructed a random variable called Z_n that had an approximately standard normal distribution. But we recall the following properties of linear transformations of Normal random variables:

- Multiplying a Normal random variable by a constant c results in a new Normal random variable with c times the mean and c times the standard deviation of the original.
- Adding a constant d to a Normal random variable results in a new Normal random variable with c added to the expected value and no change to the standard deviation (since adding a constant doesn't change the spread.)

So to find the distribution of $\bar{X}_n$:

- Z_n is approximately Normal$(0, 1^2)$;
- Multiplying by $\sigma/sqrtn$ gives an approximately Normal$(0, [\sigma/sqrtn]^2)$ or Normal$(0, \sigma^2/n)$ random variable;
- Adding μ gives an approximately Normal$(\mu, \sigma^2/n)$ random variable.

That is: the mean of a large iid sample is approximately Normal with mean μ, variance σ^2/n, and standard deviation $\sigma/\sqrt{n}$.

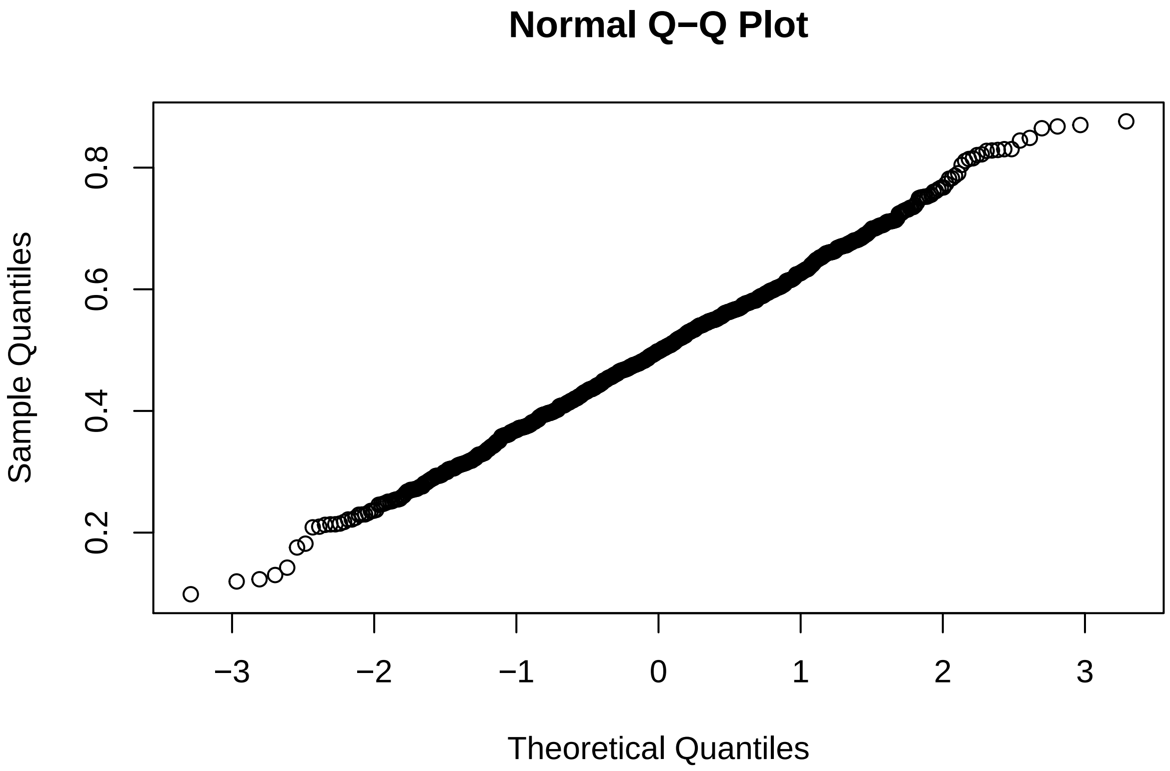
Furthermore, the **sum** of the n iid random variables is just n times the sample mean. So the sum of a large iid sample is approximately Normal with mean $n\mu$, variance $n\sigma^2$, and standard deviation $\sigma\sqrt{n}$.

Does the CLT work?

You might ask: How large is large? That is, how large a sample do you need to get an approximately normal distribution for the sample mean? The unsatisfactory answer is: it depends.

Let's start with samples from a uniform distribution. The uniform is well-behaved: it's symmetric and has no extreme values. Let's try taking samples from the uniform using `runif()`, finding the mean, then repeating this lots of times (say a thousand.) Then we'll have a sample of sample means. We can then check normality using the `qqnorm()` plot. With such a nice distribution, even the mean of a tiny sample of size 5 looks pretty normal, except perhaps at the tails (the extremes of the distribution):

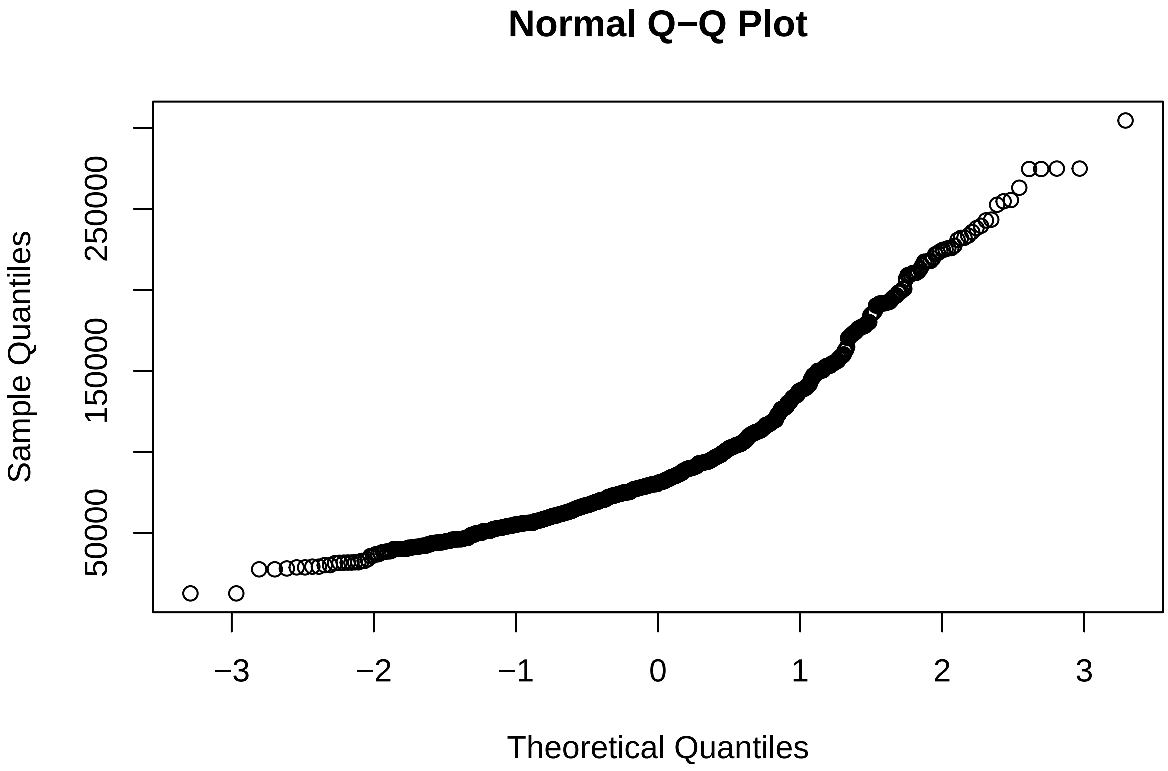
```
uniform.means = replicate(1000, mean(runif(5)))
qqnorm(uniform.means)
```



Let's now try taking samples from the faculty salaries population and checking for normality. So we'll take a sample of size 10, find the sample mean, repeat this a thousand times, and plot the results.

Firstly, look at samples of size 1:

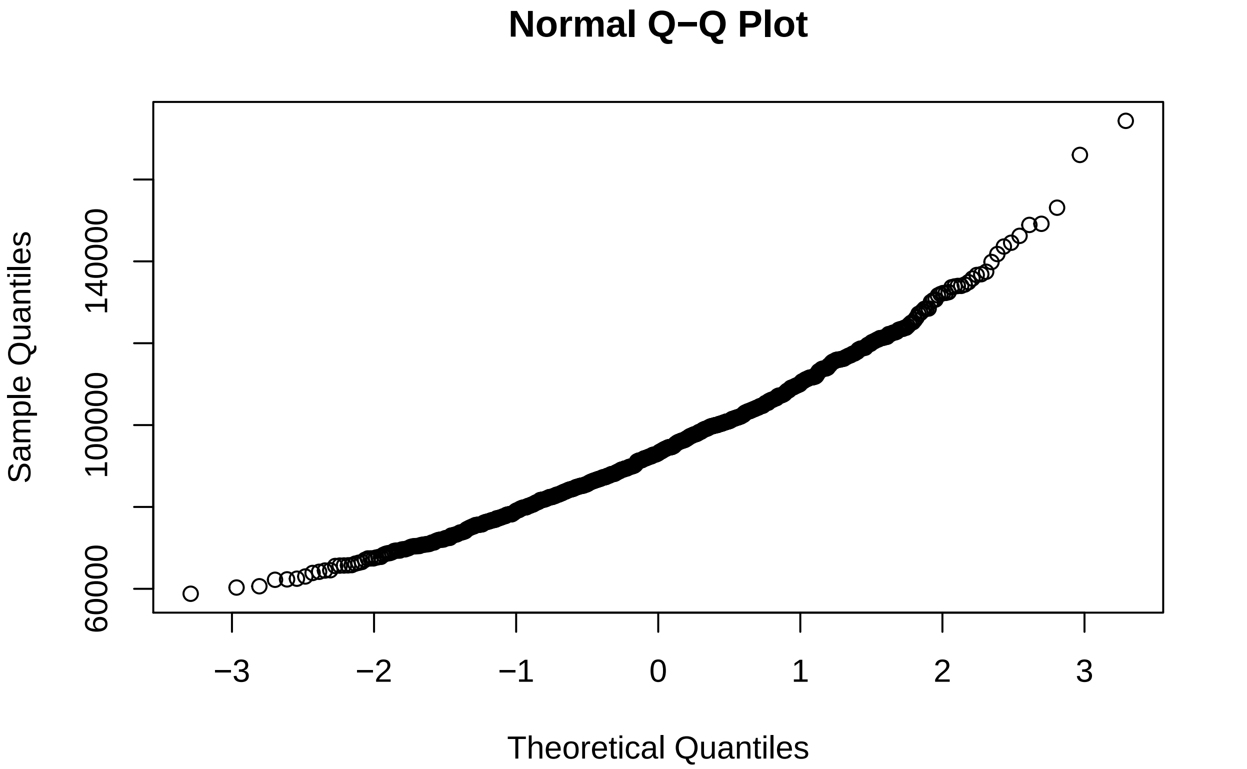
```
sample.means.1 = replicate(1000, mean(sample(salaries.all, size=1)))
qqnorm(sample.means.1)
```



The mean of a sample of size 1 is just the same as taking a sample of size 1. So the distribution of samples looks the same as the distribution of the population, which is right-skewed and strongly non-normal. The skewness and extreme values mean this distribution is not as nice as the uniform, and it's going to take a larger value of n before the sample mean is close to normal.

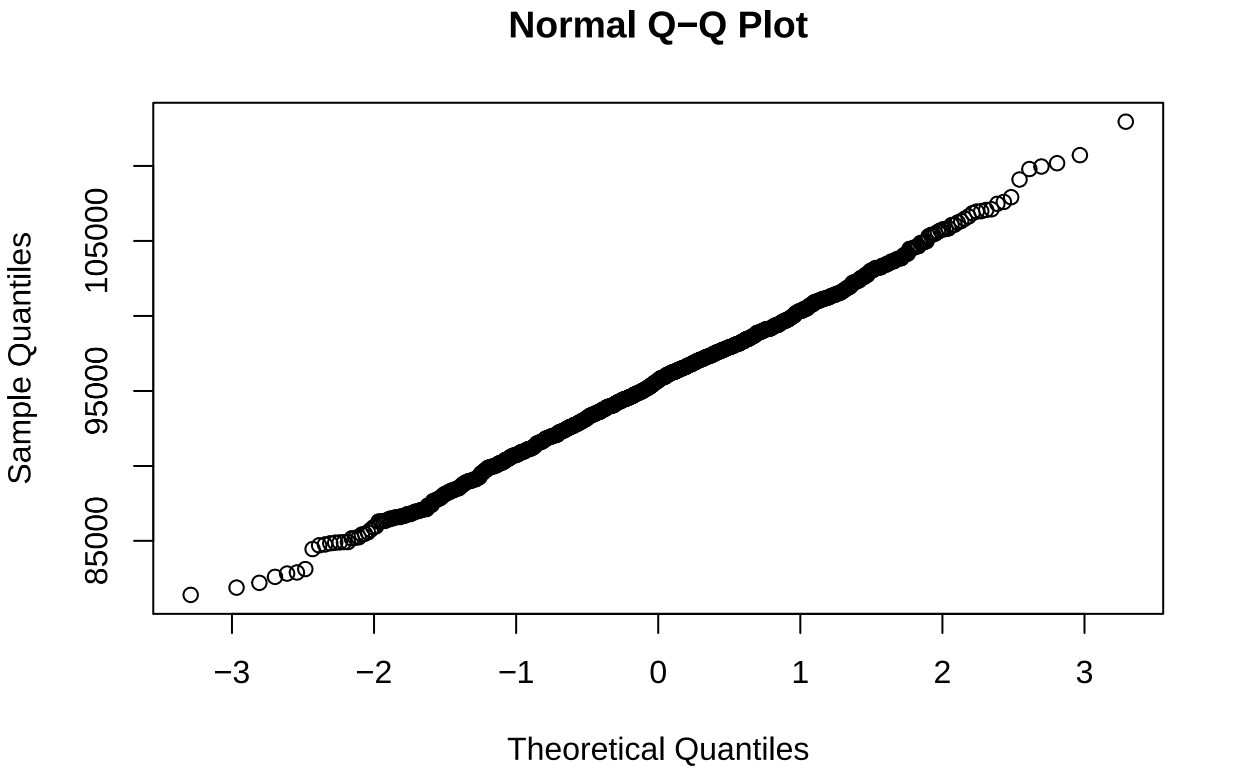
Try $n = 10$:

```
sample.means.10 = replicate(1000, mean(sample(salaries.all, size=10)))
qqnorm(sample.means.10)
```



This is better, but if you look, you'll notice the line curves upward a little bit. What if we bump the sample size to 100?

```
sample.means.100 = replicate(1000, mean(sample(salaries.all, size=100)))
qqnorm(sample.means.100)
```



It's not absolutely perfectly normal: there's still a bit of weird stuff going on at the tails. But for practical purposes, the normal is a good enough approximation.

For an even more extreme case, consider games of roulette where you bet on one number (out of 38): if your number comes up, you make a profit of 35 times your original stakes; otherwise, you lose. The random variable representing your winnings has two non-nice features: firstly, it's extremely skewed, and secondly, it's discrete, with only two possible outcomes. Neither of these features is present in the normal. It thus takes an *extremely* large n before the distribution of the sample mean converges to the normal: in the thousands.

So our answer to the question "how large is large n?" is "it depends" – not only on the shape of the population distribution, but also on what we want to use the normal distribution for the sample mean. In particular, getting the tails of the distribution right can take a much larger sample size than just getting the middle of the distribution to resemble a bell-shaped curve. A common (if theoretically unjustified) rule of thumb is that when $n \geq 30$, the normal distribution can be used to approximate the distribution of the sample mean. This rule is probably acceptable if the fate of the world doesn't depend on the exact probabilities you obtain. On the other hand, improper use of the normal distribution to estimate tail probabilities contributed to the 2008 financial crisis. So a better rule of thumb might be: if using the normal inappropriately might trigger a financial crisis, then don't use the normal.

Example. *The number of points scored in a NFL game (by both teams combined) in the 2013 regular season had a non-normal distribution with mean 46.8 points and standard deviation 14.4 points.*

I select a simple random sample of 30 NFL games from the 2013 season. What is the probability that the average number of points scored in these 30 games is at least 50 points?

There are some drawbacks to using the normal approximation here:

- The distribution is probably skewed: sometimes teams score a lot of points.
- The distribution is discrete.
- 30 isn't a huge sample size.
- The games aren't quite independent: there are only 256 games in an NFL season, so sampling 30 games without replacement won't be iid.

Apply our rule of thumb: No, a slightly inaccurate answer here is unlikely to cause a financial crisis. So we will use the normal approxmation.

The sample mean thus has an approximately normal distribution with mean 46.8 and SD $14.4/\sqrt{30}$. Now we just need one line of R:

```r
1 - pnorm(50, 46.8, 14.4/sqrt(30))
```

```
## [1] 0.1117714
```

By the normal approximation, the probability is about 11%. If we had all the data (and you can easily find it on the internet), we could find a more accurate answer by simulation. It's actually more like 10%. That's okay: a 1% difference in probability is unlikely to matter much when it comes to NFL games, unless you're a gambler or you're a coach down 3 in the fourth quarter of the Super Bowl and you're trying to decide whether to pass or run.

Example. A **random walk** is a popular statistical model for the evolution of "unpredictable" values over time, such as stock prices. In its simplest form, the daily change in the value is taken to be a sequence of iid random variables.

One use of random walks is to model elections. Let W be the support of a major candidate (in terms of proportion of the two-party vote) on a day 0, n days before the election. Let X_i be the change in the support of the candidate, and suppose the X_i's are iid with expected value 0 and standard deviation σ. Then the candidate's support on election day is $Y = W + \sum_{i=1}^{n} X_i$. Further, suppose that the election is conducted in a country with a sensible electoral system, where the candidate with the most votes wins. So the candidate wins if and only if $Y > 0.5$.

Suppose that 30 days before the election, a candidate's has 52% support, and that past data suggests that $\sigma = 0.5\%$ or 0.005. Is the candidate's lead safe?

Answer. Firstly, will an inaccurate answer cause a financial crisis? Let's hope not.

Now, the change each day is iid with mean 0 and standard deviation 0.005. The change over 30 days will be the sum of these 30 iid random variables, so it should be approximately normal. The mean of this sum will be $30 \times 0 = 0$, and the standard deviation will be $0.005 \times \sqrt{30} = 0.027$. Now we add on the candidate's initial support of 0.52. So Y is approximately normal with mean 0.52 and standard deviation 0.027. We need to find $P(Y > 0.5)$:

```
1 - pnorm(0.5, 0.52, 0.005*sqrt(30))
```

```
## [1] 0.7673956
```

So the candidate has a 77% chance of winning. He or she is the favorite to win, but it's far from certain.

See Trosset pp. 190–193 for more examples.

Errors and standard errors

So far we know:

1. The expected value of the sample mean equals the population mean.
2. The distribution of the sample mean is approximately normal for a large sample.

Let the **error** of the sample mean be defined as sample mean minus population mean ($\bar{X} - \mu$). From 1 and 2, it follows that:

3. For a large sample, the error of the sample mean is approximately normal with mean 0.

Now, to use the normal, we need to know one more thing: the standard deviation of the distribution of the error. This is sometimes called the **standard error** of the mean for short. If we knew the population standard deviation was σ, then from the CLT, the standard error would be $\text{SD(error)} = \sigma/\sqrt{n}$, where n is the sample size.

Well, if our sample size is indeed large, a good estimate of the population standard deviation is the sample standard deviation, s, which is just the estimate provided by the `sd()` function in R. This adds another layer of approximation, so we restate:

4. For a very large sample, the error of the sample mean is approximately normal with mean 0 and estimated standard error $s/\sqrt{n}$.

We can often do better than this approximation if we know something about the population (e.g. what kind of distribution it follows.) But this result always holds, provided we take a large enough sample.

Sneak preview: Confidence intervals

If we set up a interval of the form

$$\bar{X} \pm q\frac{s}{\sqrt{n}}$$

then the (approximate) chance that the interval contains the true population mean μ is `pnorm(q) - pnorm(-q)`.

Example. Let $q = 1$. Then the interval $\bar{X} \pm q\frac{s}{\sqrt{n}}$ contains μ with probability `pnorm(1) - pnorm(-1)` $\approx 68\%$.

Now, what if instead of starting with a q, we start with a probability of getting an interval that contains the true value? Let's say we want such a probability $1 - \alpha$. Then we need to find a q such that `pnorm(q) - pnorm(-q)` $= 1 - \alpha$. From the definition of quantiles and the symmetry of the normal, this is just

$q =$ `qnorm(1 - alpha/2)`

Why? Then `pnorm(q)` is $1 - \alpha/2$ by definition, `pnorm(-q)` is $\alpha/2$ by symmetry, and `pnorm(q) - pnorm(-q)` is $1 - \alpha$ as required.

Example. What does q have to be to get a 95% chance of including the population mean?

```
qnorm(0.975)
```

```
## [1] 1.959964
```

Round this to 1.96 and memorize it.

We now know how to construct an **interval estimate** (rather than a single-number **point estimate**) for the mean faculty salary based on our original random sample of 100. Take the sample mean and add and subtract 1.96 estimated standard errors.

```
mean(salaries) - qnorm(0.975) * sd(salaries) / sqrt(length(salaries))
```

```
## [1] 81643.45
```

```
mean(salaries) + qnorm(0.975) * sd(salaries) / sqrt(length(salaries))
```

```
## [1] 100572.7
```

We get the interval estimate $81,643 to $100,573. As we said earlier, we don't care about a few dollars (and couldn't estimate to that accuracy even if we did), so round to $82,000 to $101,000.

The last catch is that it's not longer straightforward to talk about probability. You can talk about probability before you take the sample, or if you're talking about a theoretical sample. But once you get a particular sample, that'll determine a particular interval, and it'll either contain the true value or it won't. So we fudge and call it an (approximate) **95% confidence interval**. A 95% confidence interval is an interval estimate constructed in such a way that 95% of the time, you'll get an interval that contains the true value. That means 5% of the time you'll be wrong, but that's statistics.

Do confidence intervals work?

Let's do a simulation. (The exact R syntax here is beyond the scope of the course, but copy-paste if you feel like it.)

```
faculty.interval = function(n){
  salaries.sample = sample(salaries.all, size=n)
  x.bar = mean(salaries.sample)
  s = sd(salaries.sample)
  lower = x.bar - qnorm(0.975) * s / sqrt(n)
  upper = x.bar + qnorm(0.975) * s / sqrt(n)
  return(c(lower, upper))
}
simulations = replicate(1000, faculty.interval(100))
lower.list = simulations[1,]
upper.list = simulations[2,]
too.low = sum(upper.list < mean(salaries.all))
too.high = sum(lower.list > mean(salaries.all))
just.right = 1000 - too.low - too.high
print(c(too.low, too.high, just.right))
```

```
## [1]  37  17 946
```

It's a bit lower than 95%, but close. In chapter 10, we'll learn a method that has closer to the right level of coverage.