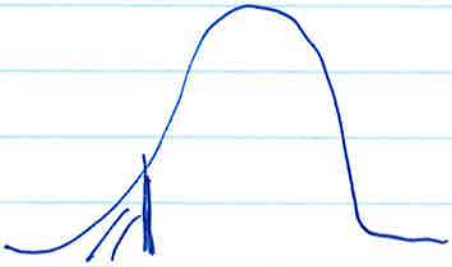
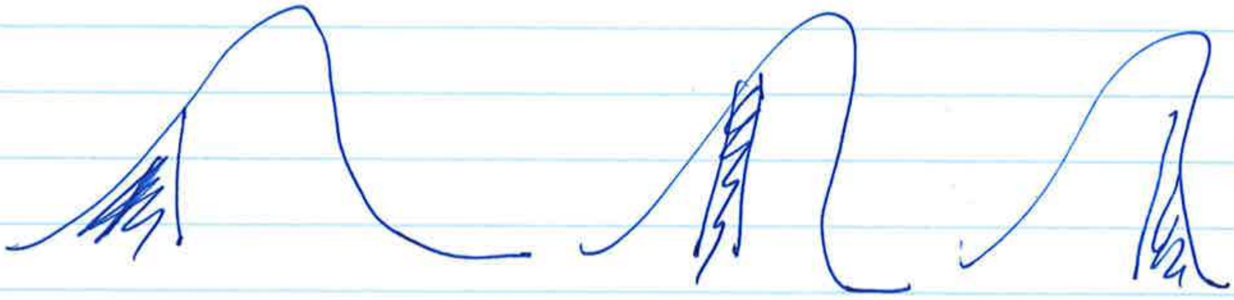
Stat 202 2015S-W4-Tues Pg 1

Review - not a lot of new material last class



Cumulative proportion - proportion of obs (or area of density curve) at or below a given value

We showed with StatCrunch how to compute



Area of these wedges with Stat Crunch for Normal density curves.
Also to compute value given cumulative proportion.

Any Questions?

New Normal Quantile Plot

QQ-plot in statcruch)

Question What transformation makes the data N(0,1) distributed? In otherwords what transformation of the data makes the new variable N(0,1)?

The transformation that makes the data N(0,1) distributed will be linear if and only if the original data is Normally distributed.

So here's a way to tell if data are normally distributed.

Compute and Plot the transformation that makes it N(0,1). If the plot is a line the original data is Normal, If not it isn't. More sensitive to the eye than juding if density curve is bell shaped.

The plut is the Normal - Quantile plot or QQ plot.

Stat202-2015S-W4-Tuesday pg3

To get an intution about what a QQ plot looks like with Normal data we want to get some Normal data and plot it.

Hint: It won't be a perfect line even if data are Normal because the data are random.

We need to see plots to interpret plots.

Where are we going to get the data?

There are computer programs that output numbers that are Normally distributed.

In all other situations you are never sure its normal.

So we are going to use StatCrunch→data
→ simulate to make normal data.

(More will be said about using computers to generate random numbers).

↗ in a few minutes

Pg 4

Stat 202-2015 S-W4-Tuesday

StatCrunch

Simulate 200 N(0,1) dynamic seed

Plot QQ plot

Do it again

Plot QQ Plot

mean

Simulate N(15, 1)

Plot QQ plot

Note: Old variable on vertical axis.

~~Uniform~~ — mention because on homework

Try with t-distribution if time permits and judging attention of class.

Show link from website

Stat 202 2015 S W4 - Tues

Lets talk about using a computer to generate random numbers

Computers are not random. Every time you run a program with the same input you get the same results

The solution is to program a a pseudorandom number generator

All random numbers in a computer are actually pseudorandom numbers

A Pseudorandom number generator will give you a stream of numbers that are indistinguisable from a random sequence (with the specified distribution) at least in theory.

There are different algorithms for doing this, some are better than others, none are perfect. Actually they are really really good. The only people who worry about their imperfections are mathematicians who study the algorithms

I said computers always get the same result every time they run a program with the same input.

That's true of random number generators as well; you get the same stream of numbers every time you run the program

But you have the option of selecting a seed. Every seed gives a different stream of numbers. And there a trillions of possible seeds (any integer that can be represented by the computer will do).

In stat crunch if you pick a fixed seed you will get the same stream of numbers

Try it

3, 4

The nice thing about this is that with a fixed seed you can always go back and get the same seed,

Helpful for reproducing errors — debugging
eg if you were programming a game.

If I tell you what seed to use, I'll know what answer you should get
(good for exams)

If you choose a dynamic seed StatCrunch will pick the seed for you based on the system clock so that every time you run it you get a different seed.

§4.1 So we have ~~[illegible]~~ talked about pseudorandomness. It is time to talk about randomness

quoted from book: We call a phenomenon random if individual outcomes are uncertain but there is nonetheless a regular distribution of outcomes in a large number of repetitions

Coin Toss: $\frac{1}{2}$ heads $\frac{1}{2}$ tails
Die toss: $\frac{1}{6}$ for each face
SAT Scores of ~~Heights~~ of students: Normal (bell curve)
~~female or male (might not be~~

The probability of any outcome of a random phenomenon is the proportion of times the outcome would occur in a very long series of repetitions

$\frac{1}{2}$, ~~@~~ for Heads
$\frac{1}{6}$ for 1 on dice / harder for SAT Scores
we'll get to it.