Stat 411/511

Assumptions of the T-tools

Oct 19 2015

Charlotte Wickham

stat511.cwick.co.nz

Announcements

Quiz #2 this weekend.

Same format, same timing.

Study guide posted.

Participation Task

You have been assigned a **Participation Number** in canvas grades: 1, 2, 3 or 4.

Form a study group with at **least 3 numbers represented.**

Primary task: organize a time to meet for at least an hour and plan a topic/quiz/chapter/concept to study. Meet at the scheduled time and study together!

Required steps to get credit:

Make/join your group in canvas

Use your group discussion board to schedule your time and post a plan of action before meeting.

After meeting post a summary of who attended, what you studied and any other comments you have.

Each member should also post this summary in the “Participation” Assignment

Grading: Pass or Fail, all group members get the same grade (unless you failed to attend the study session).

Graded Thursday Dec 10th 2015

Today

Assumptions of the t-tools

Robustness

Normality

Mathematical assumptions of the two sample t-tools

1. Independent random samples from

2. Normal populations, with

3. equal standard deviations

never see the population

examine samples for evidence

In practice these assumptions are never strictly met, but luckily they don't have to be.

A test is **robust** to an assumption if the test is valid even if the assumption is not met.

This is going to quite a high level summary. Read Section 3.2 in Sleuth for all the details.

What does it mean for a procedure to be valid?

A **test** is valid if 5% of the time,
the test rejects the null hypothesis (at the 5% level)
when the null hypothesis is true.

A **confidence interval** procedure is valid if 95% of the time,
the 95% confidence interval covers the true parameter.

If a confidence interval procedure is valid
it's corresponding test is also valid.

How do we check if a procedure is valid under some violation of the assumptions?

Use mathematical theory

Use simulation

Checking validity of CIs using simulation

1. Decide the type of violation you are interested in.
2. Simulate data according to the violation in step 1, with a known parameter.
3. Calculate a confidence interval for the simulated data.
4. Repeat 2 & 3 many times and count how often the confidence intervals capture the true parameter.

An example with no violations

Say, we are interested in the two sample t confidence intervals.

For now let's not violate any assumptions.

We will simulate two independent samples from Normal populations.

We need to pick n_1, n_2, μ_1, μ_2 and σ.

Say, $n_1 = n_2 = 30$, $\mu_1 = 1$, $\mu_2 = 2$ and $\sigma = 1$,

The parameter of interest is $\mu_2 - \mu_1$,

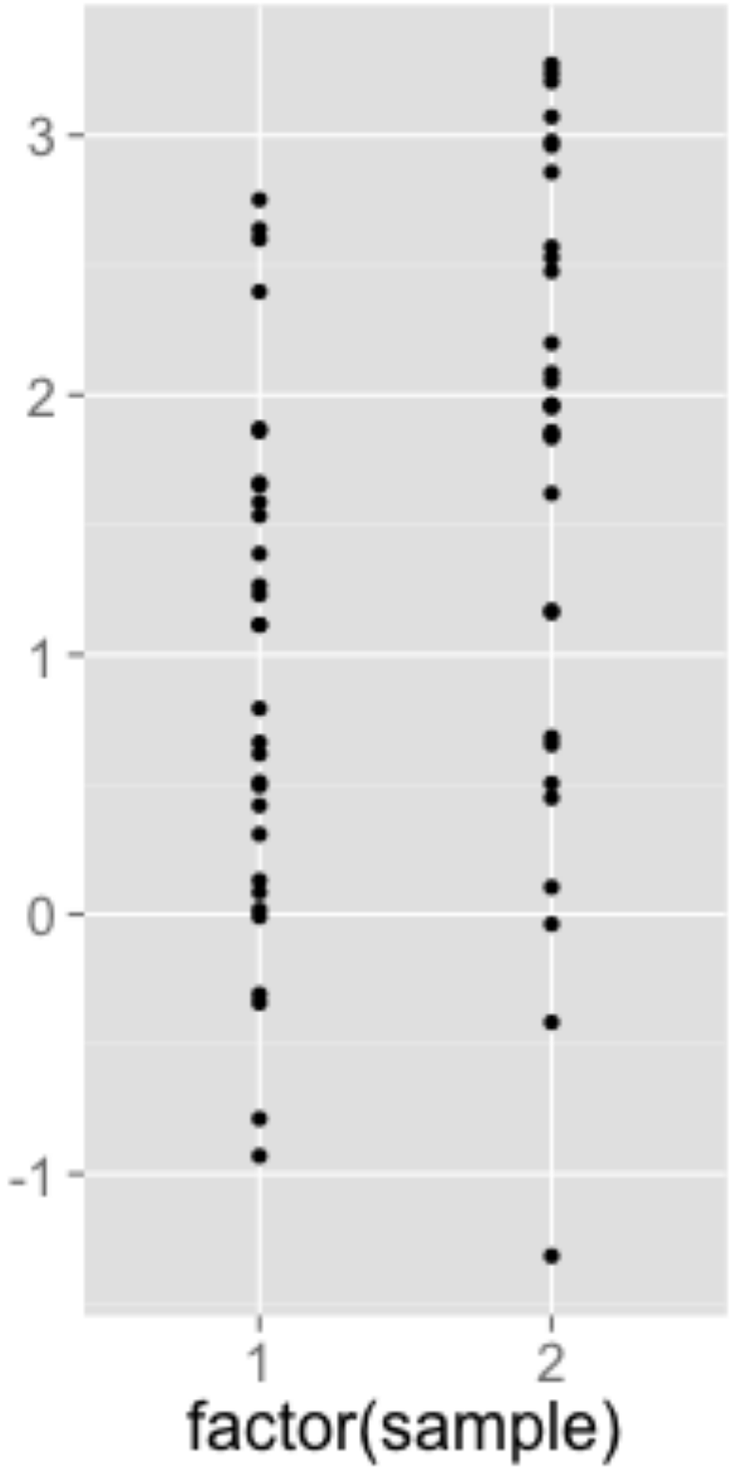
we know it, it is 1.

One simulation

```
sample1 <- rnorm(30, mean = 1, sd = 1)
sample2 <- rnorm(30, mean = 2, sd = 1)
```

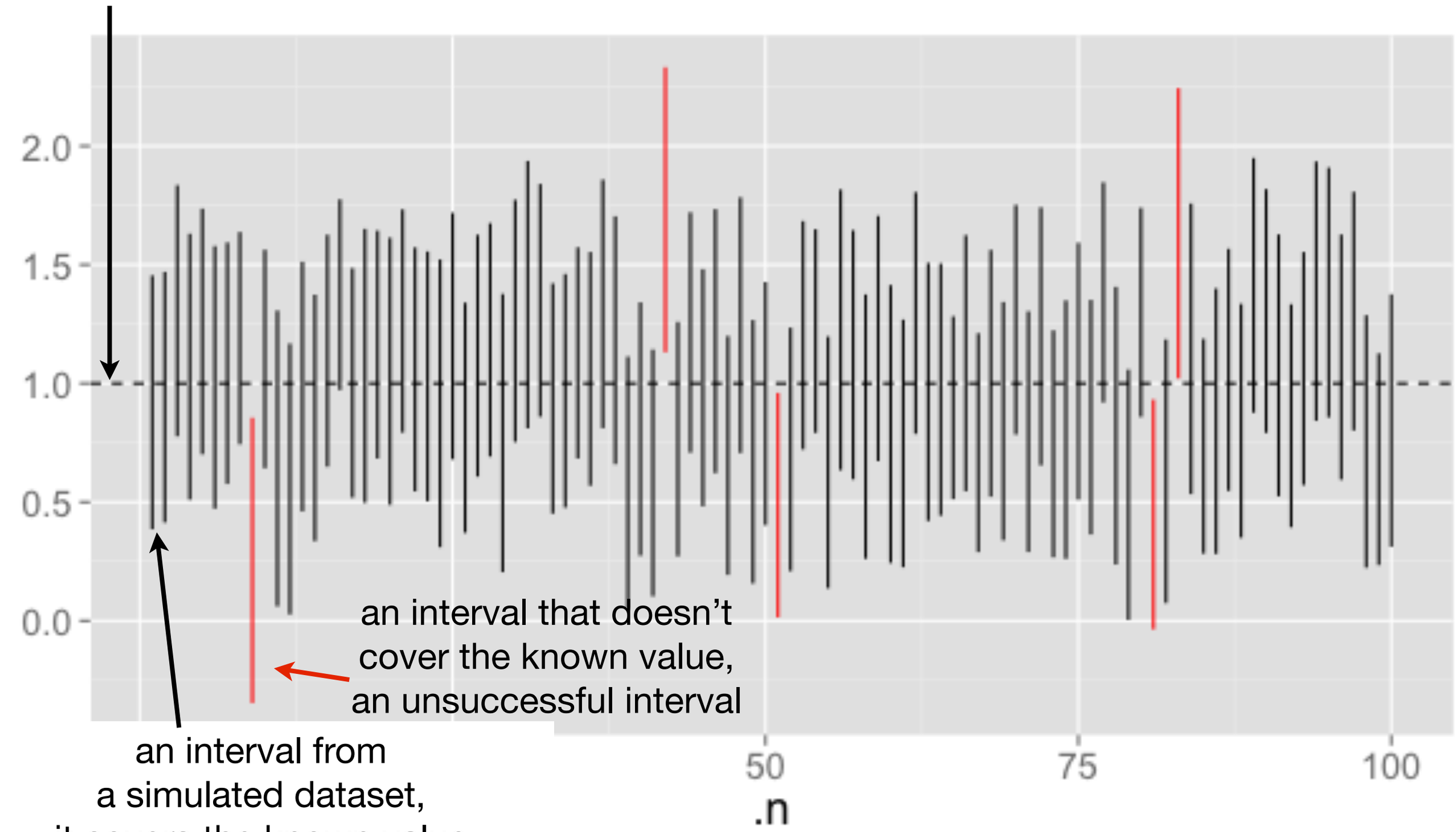
```
> t.test(sample2, sample1, var.equal = TRUE)$conf.int
[1] 0.2029862 1.3411881
attr(,"conf.level")
[1] 0.95
```

Known value of $\mu_2 - \mu_1 = 1$,
this interval covers the true value



100 simulations…

known value

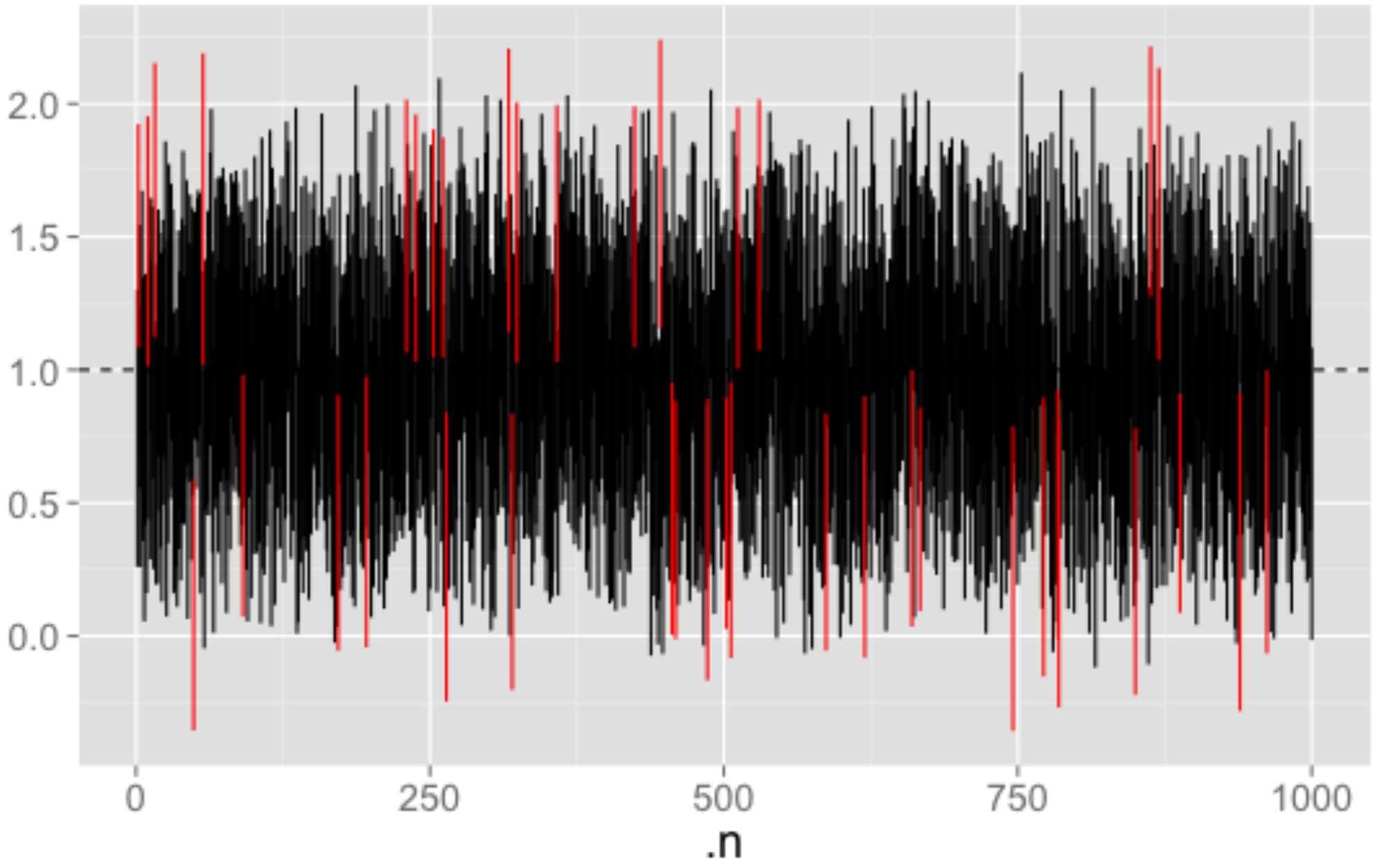


an interval from
a simulated dataset,
it covers the known value,
a successful confidence interval

an interval that doesn't
cover the known value,
an unsuccessful interval

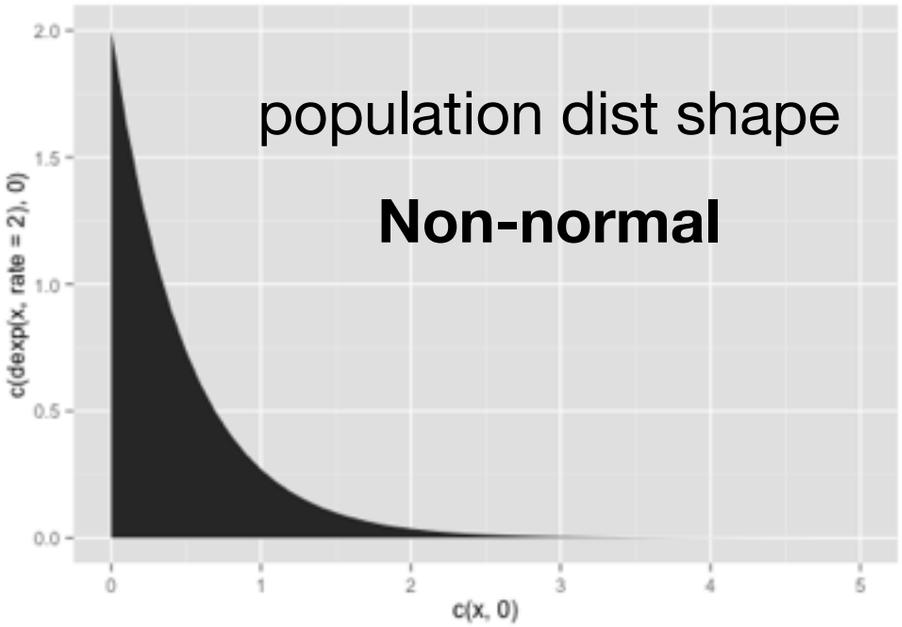
95/100 cover known value

1000 simulations...

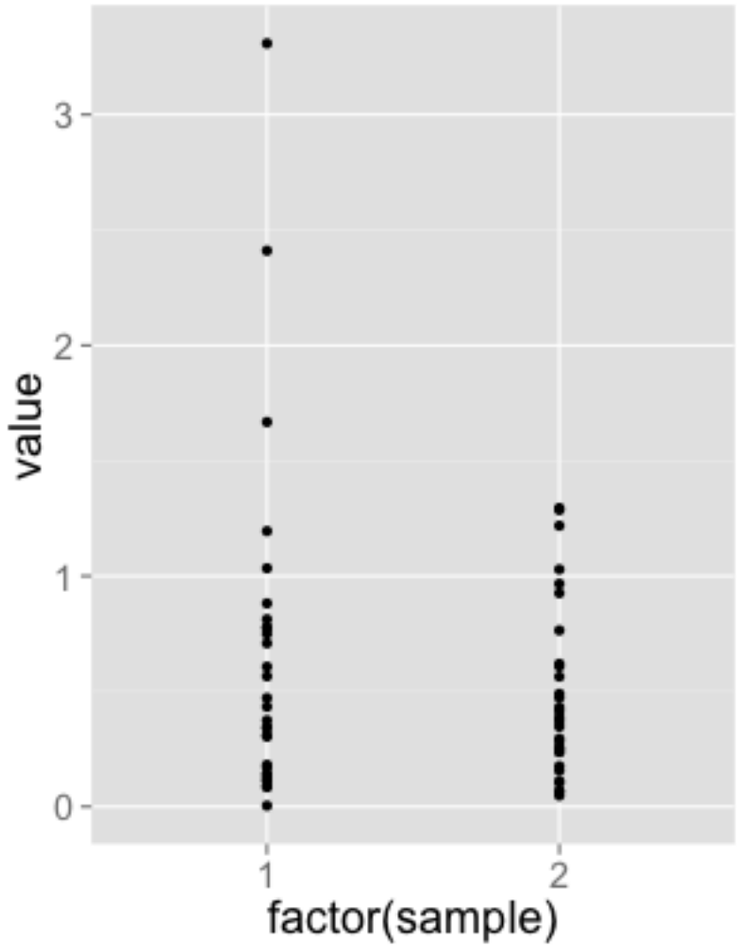


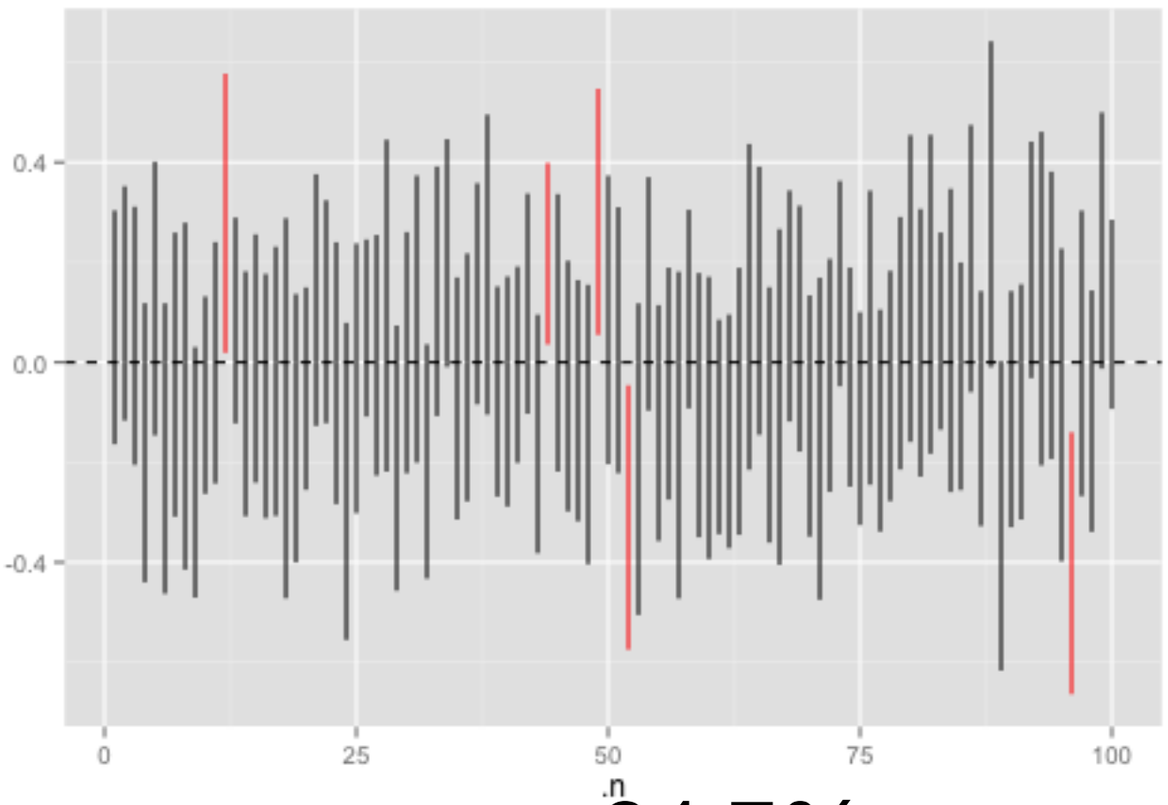
96% cover the known value

±1% based on 1000 simulated data sets



Known value of $\mu_2 - \mu_1 = 0$





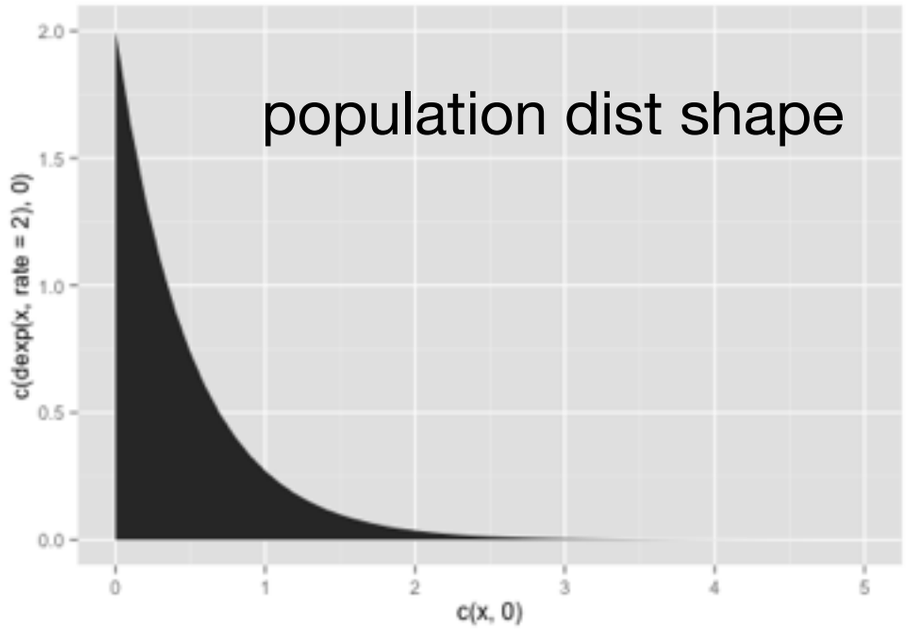
95/100 cover

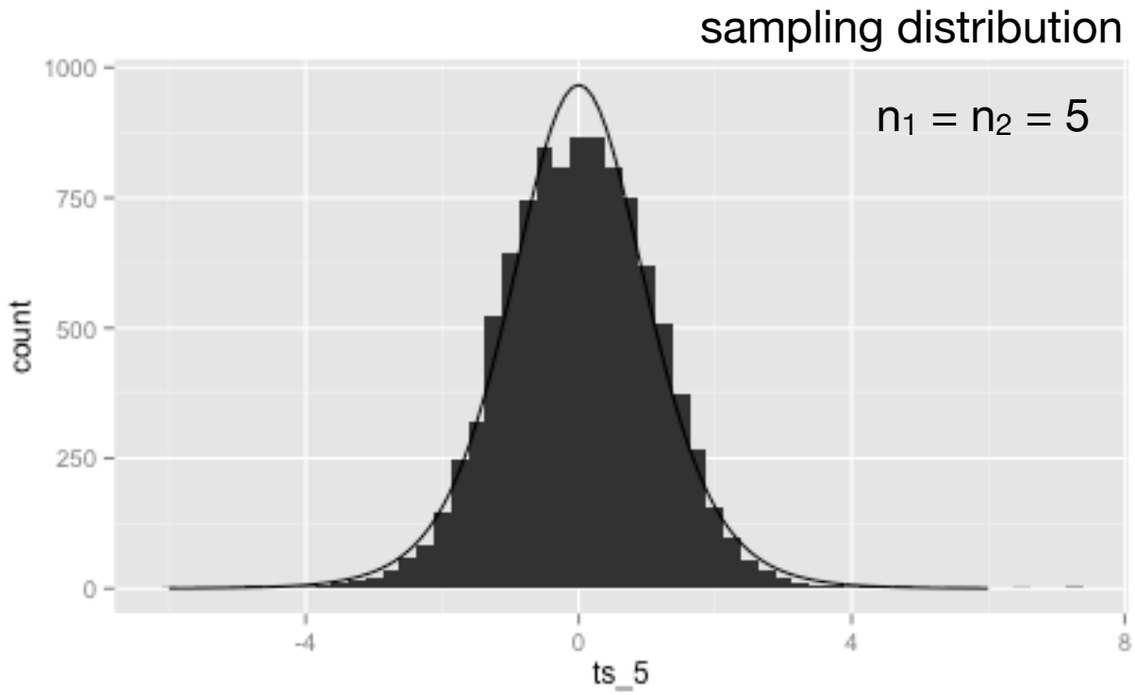
94.7% coverage in 1000 simulations

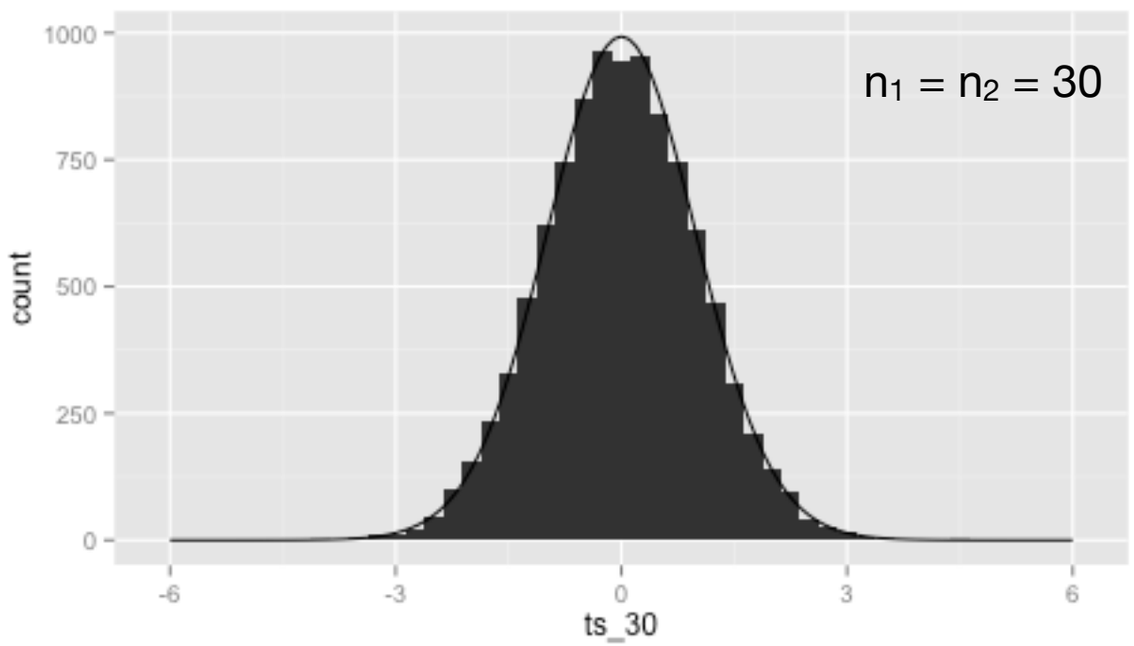
The two sample t 95% confidence intervals appear robust when the populations are exponential (with rate = 2) and sample size is 30

For the t procedures to be valid we need:

the sampling distribution of the t-statistic to be close to a t-distribution.







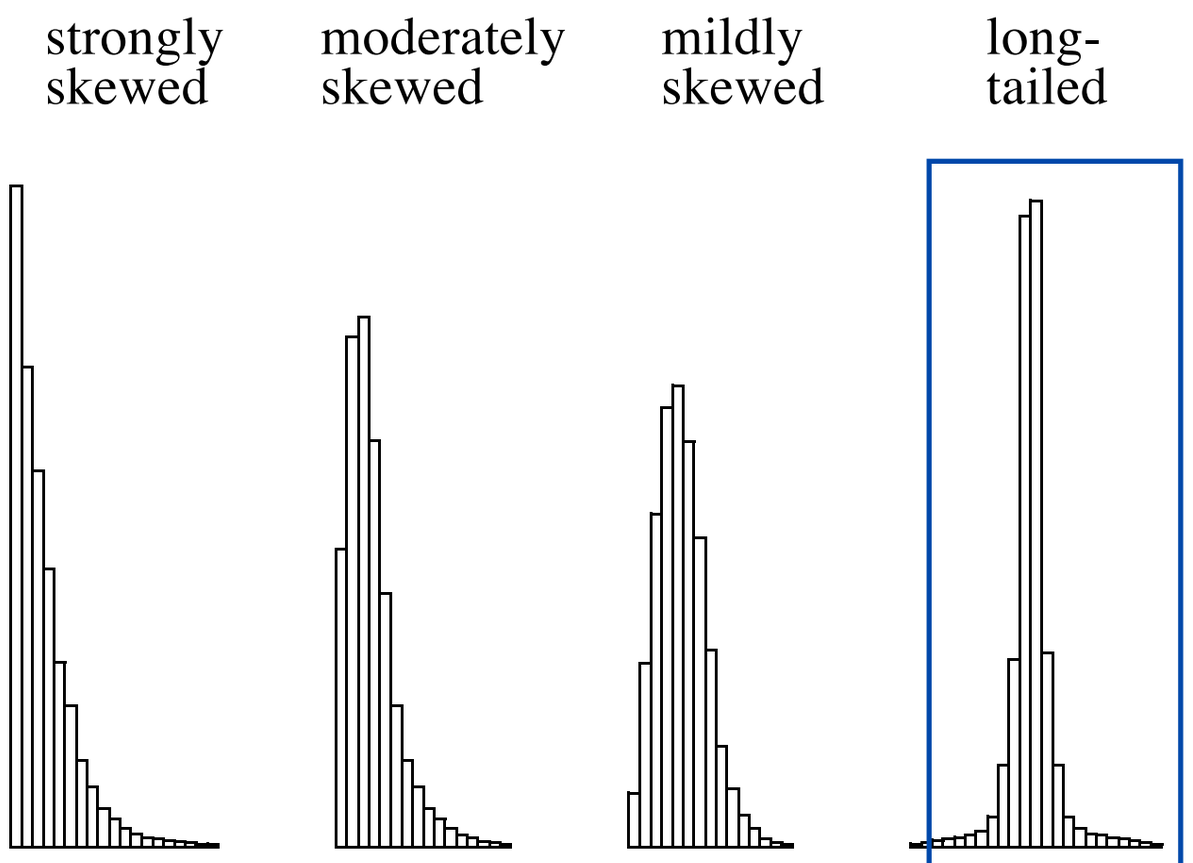
Normality of population

Normality of the population

The **larger** the sample size, the **less you have to worry**, thanks to the Central Limit Theorem.

What is a **large** sample? It depends...

Percentage of 95% confidence intervals that are successful when the two populations are non-normal (but same shape and SD, and equal sample sizes); each percentage is based on 1,000 computer simulations



Not good, but this is a pathological pop.

n_1, n_2	strongly skewed	moderately skewed	mildly skewed	long-tailed	short-tailed
5	95.5	95.4	95.2	98.3	94.5
10	95.5	95.4	95.2	98.3	94.6
25	95.3	95.3	95.1	98.2	94.9
50	95.1	95.3	95.1	98.1	95.2
100	94.8	95.3	95.0	98.0	95.6

Generalizations

If the **sample sizes** are roughly the same, the **spreads** are the same and population **shapes** are the same:

- Skew is **ok**
- Long tailed can be a problem

If the samples sizes are very different and/or the two populations have different spreads or shapes:

- Skew can be a problem
- Long tailed can be a problem

Things get better with larger samples

Normality is generally the most forgiving assumption

Normality of population

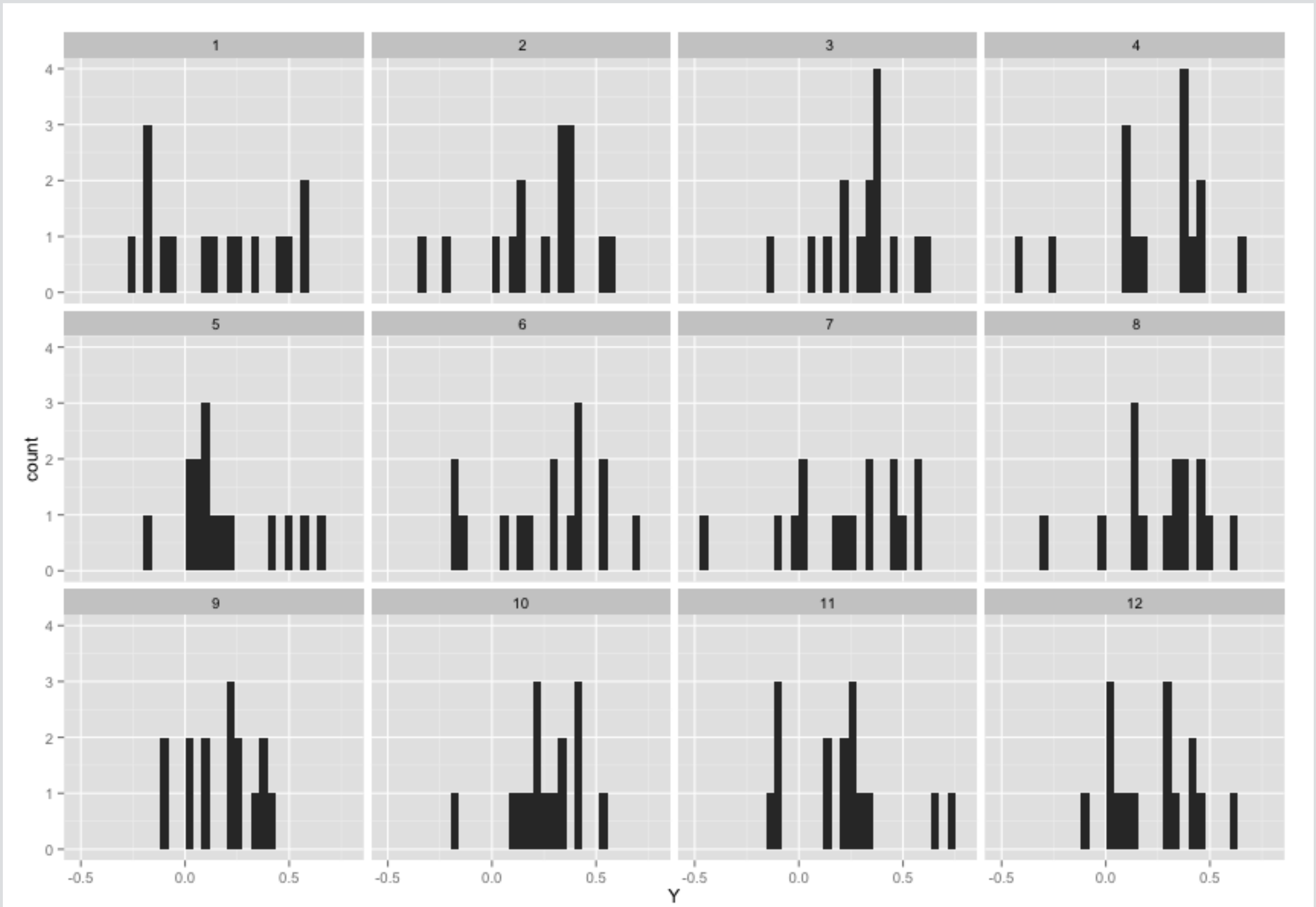
Check: by looking at histograms of samples or normal probability plots.

Remedy:

use a transformation (next week) **or**

use a non-parametric test (next week)

Your Turn: Do these samples look like they come from Normal populations?



Sampling variability

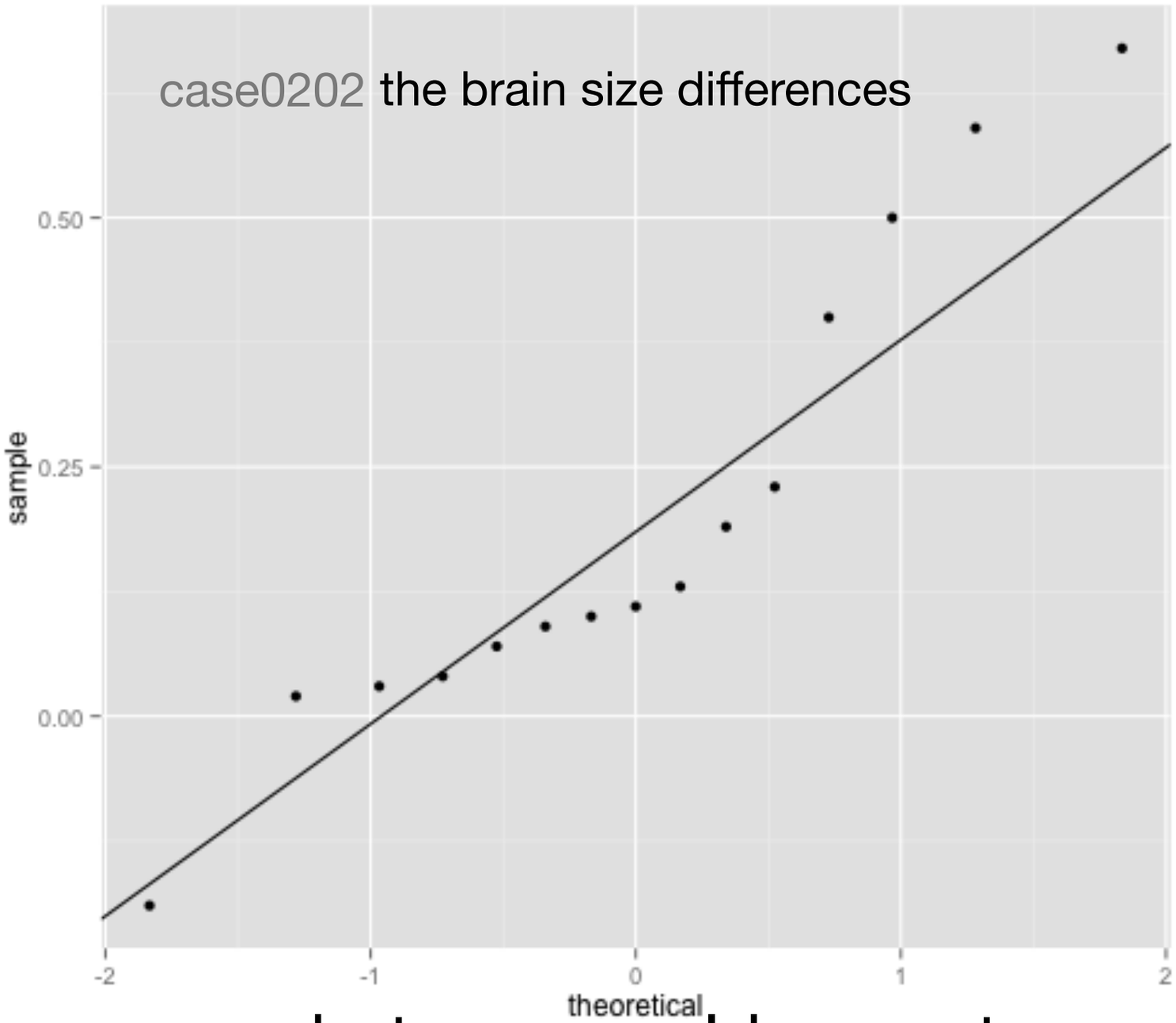
Very large samples look like the population they come from.

Small samples rarely do, just because they are small.

E.g. a sample of size 15 will never look like a “bell shaped” curve, even if the 15 numbers are drawn from a Normal distribution.

Be careful: does this sample look unusual because it comes from a non-Normal population, or just by chance?

Normal Probability Plot



the data

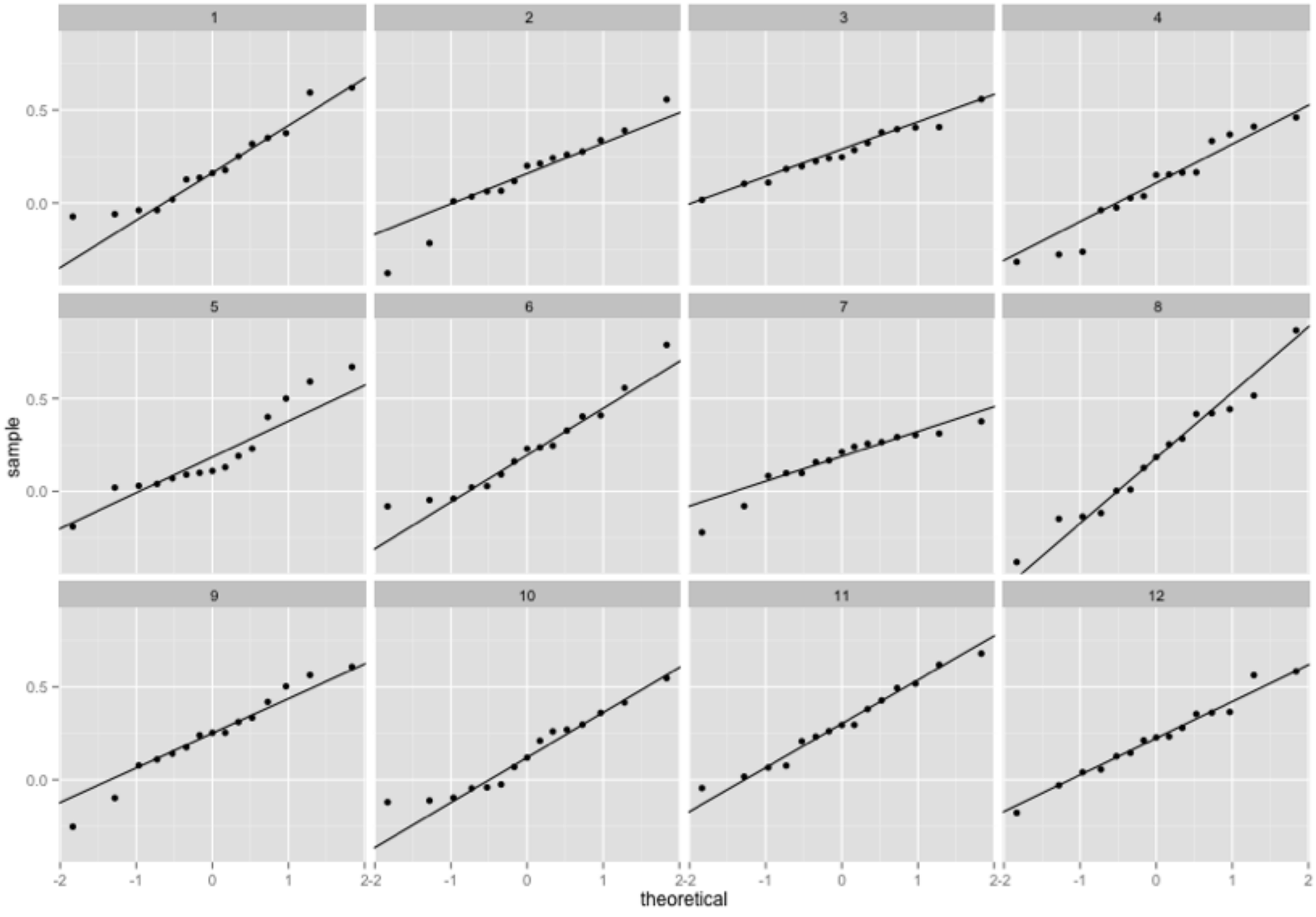
what you would expect the values to be if the data was Normal

Normal data should lie on a straight line

(but there is always sampling variability)

Calibrating your eyes

11 datasets with 15 observations randomly drawn from a Normal population + the real data



Does the real data stand out?

Normal Probability Plot

```
source(url("http://stat511.cwick.co.nz/code/stat_qqline.r"))
qplot(sample = Depth, data = case0201) +
  stat_qqline() +
  facet_wrap(~ Year, scales = "free")
```