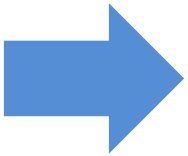
Session 3: Sampling and Confidence Intervals

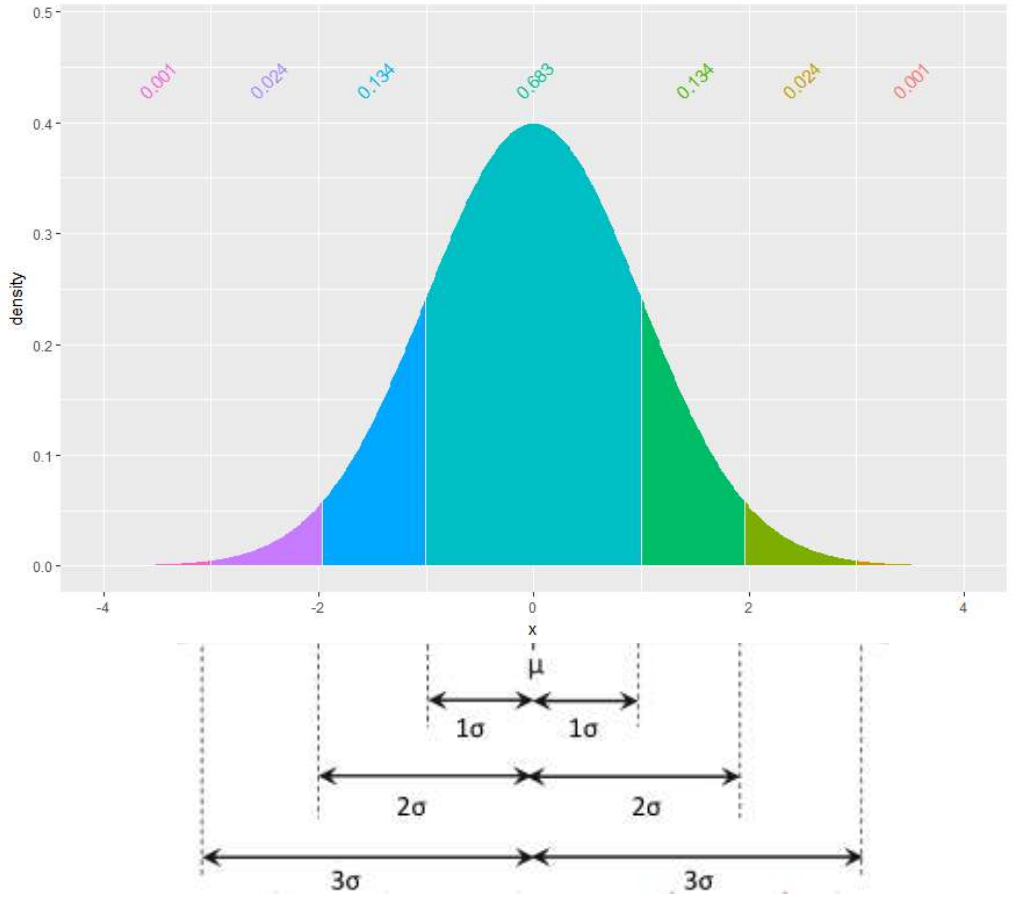
Kostis Christodoulou
London Business School

Contents



- Review of Sessions 1-2
- Sampling and Inferential Statistics
- Confidence interval for the mean
- Binomial Distribution and confidence intervals for proportions
- Bootstrap estimation using **infer**

Properties of the Normal Distribution



95% chance of being within (+/-) 1.96 standard-deviations of the mean

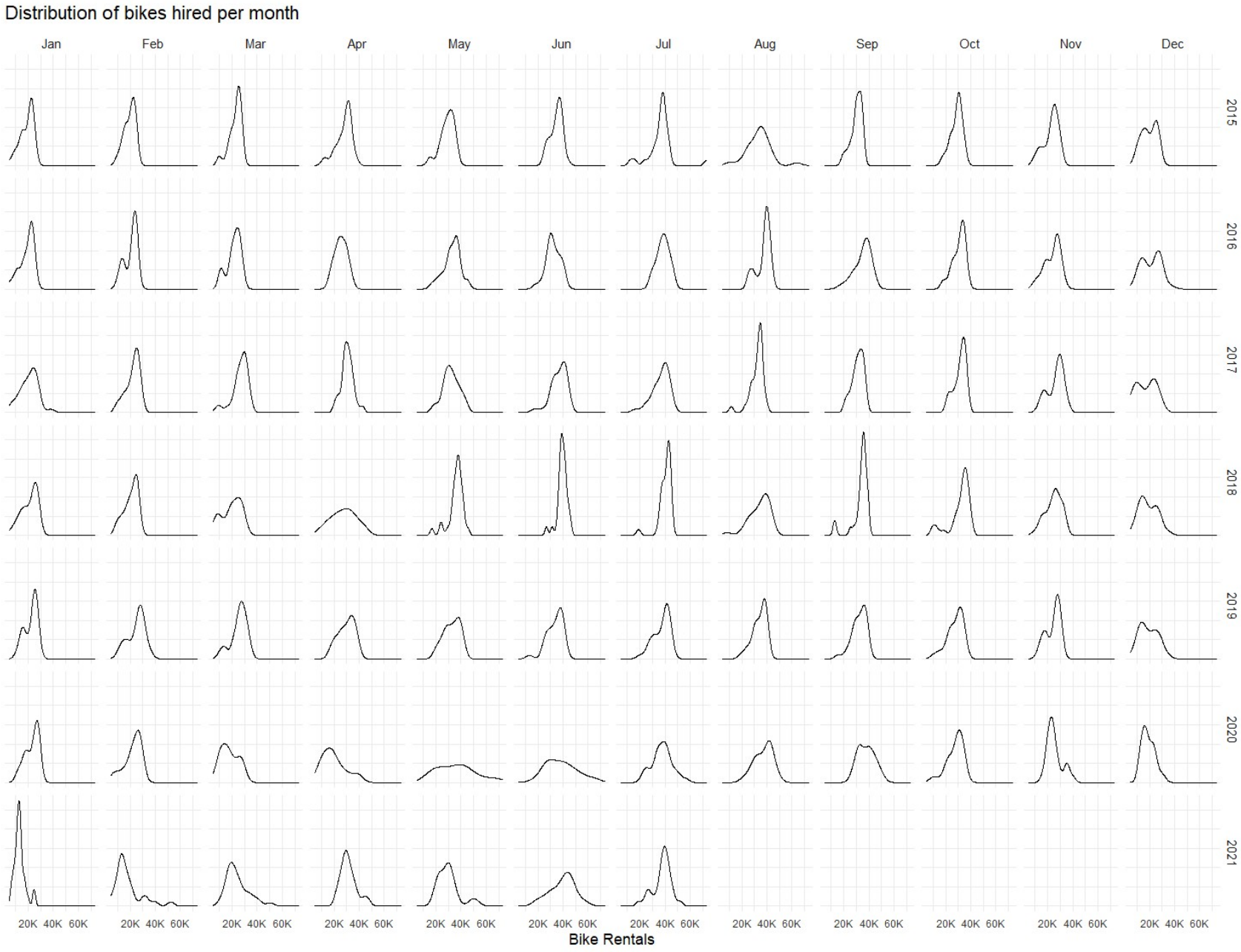
A ***Z score,*** or the number of standard deviations the observation falls above or below the mean, creates a common scale so you can assess data without worrying about the specific units in which it was measured.

$$Z = \frac{(observation - mean)}{SD}$$

Z distribution (also called the standardi**Z**ed normal distribution, is a special case of the normal distribution where μ = 0 and σ = 1
$Z \sim N(\mu = 0, \sigma = 1)$

Observations with a Z score > 2 or < -2, are usually considered unusual.

Santander Bikes: Hirings per month

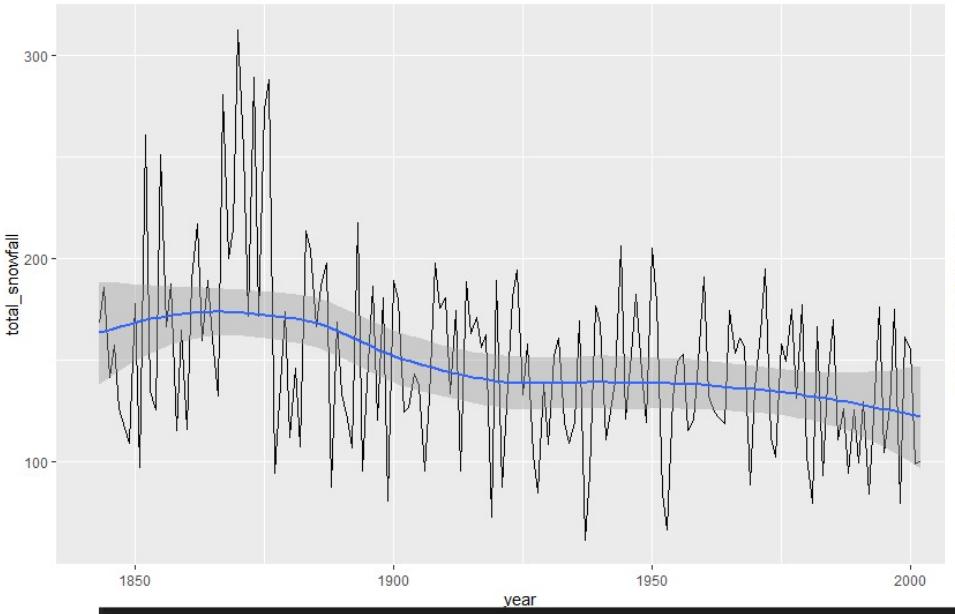


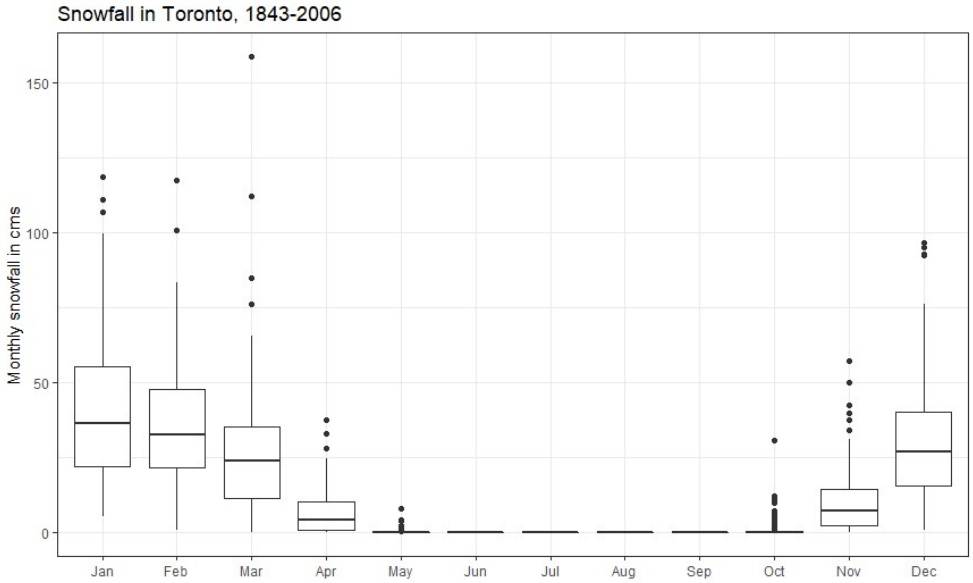
Snowfall in Toronto, 1843-2006

- Snowfall < 20% of average --> 100% refund
- Snowfall < 30% of average --> 75% refund
- Snowfall < 40% of average --> 50% refund
- Snowfall > 40% of average --> no refund

Refund_Levels	Snowfall_Below	Snowfall_cm	Probability
100	20% of Average	29.9	0.005
75	30% of Average	44.9	0.012
50	40% of Average	59.8	0.026

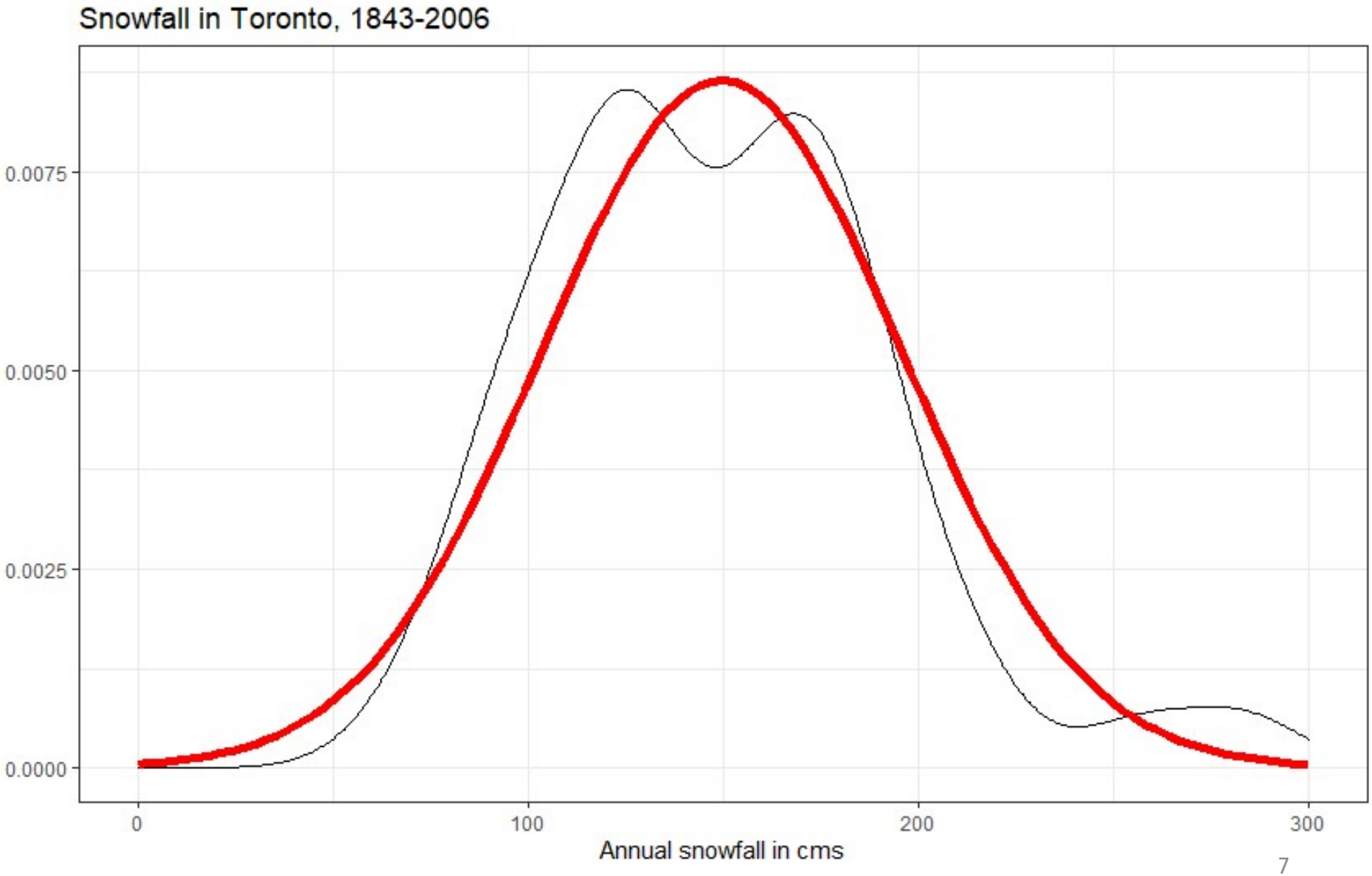
Snowfall in Toronto, 1843-2006



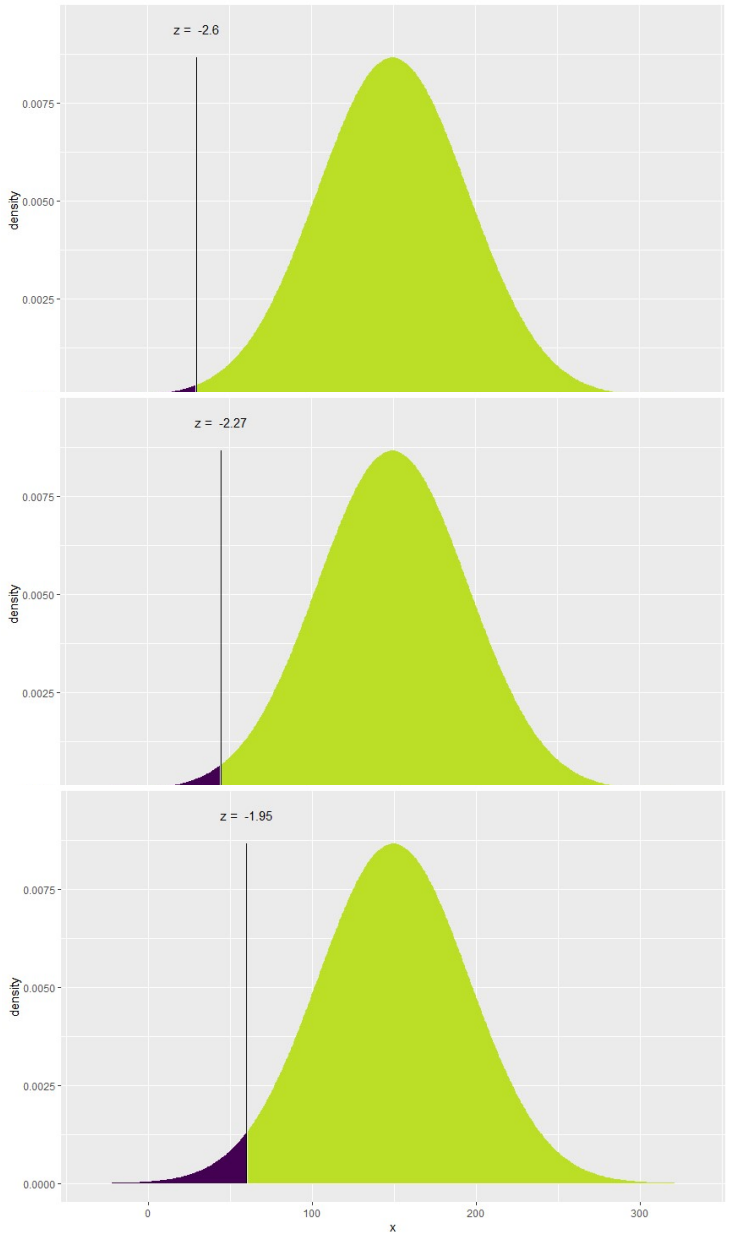



```
-- Data Summary ------------------------
                           Values
Name                       snow_totals
Number of rows             164
Number of columns          2
___________________________
Column type frequency:
  numeric                  2
___________________________
Group variables            None

-- Variable type: numeric --------------------------------------------------------------------------
-----
# A tibble: 2 x 11
  skim_variable n_missing complete_rate  mean    sd     p0   p25   p50   p75  p100 hist
* <chr>              <int>         <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
1 year                   0             1 1924.  47.5 1843   1884. 1924. 1965. 2006 ▇▇▇▇▇
2 total                  4             0.976  150.  46.1   61.8  119.  145.  176.  312. ▅▇▃▁▁
```

Is actual data close to a Normal distribution?

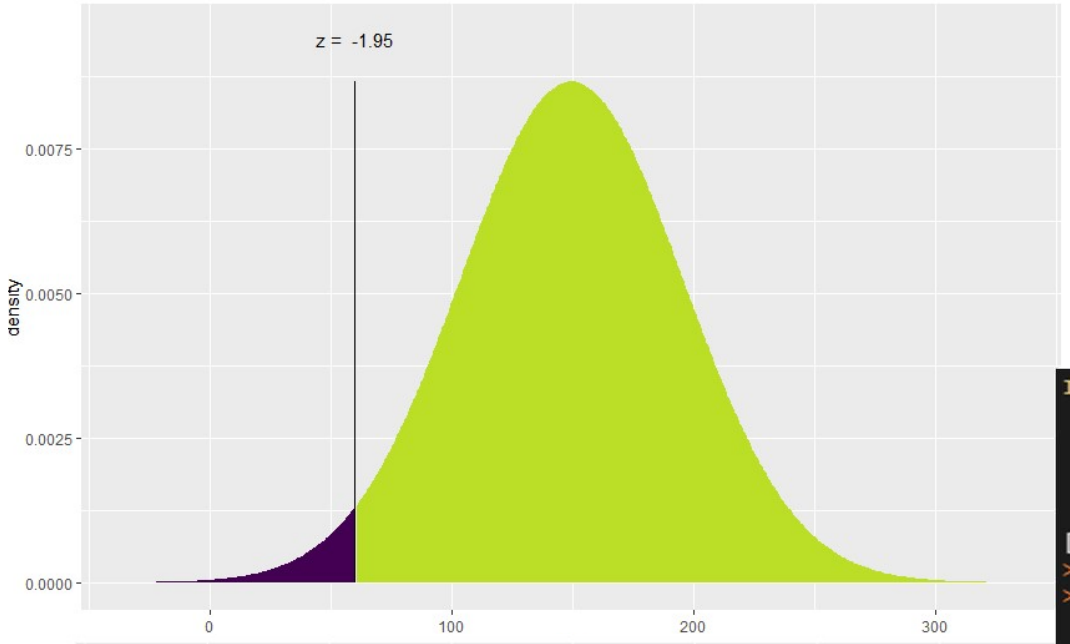


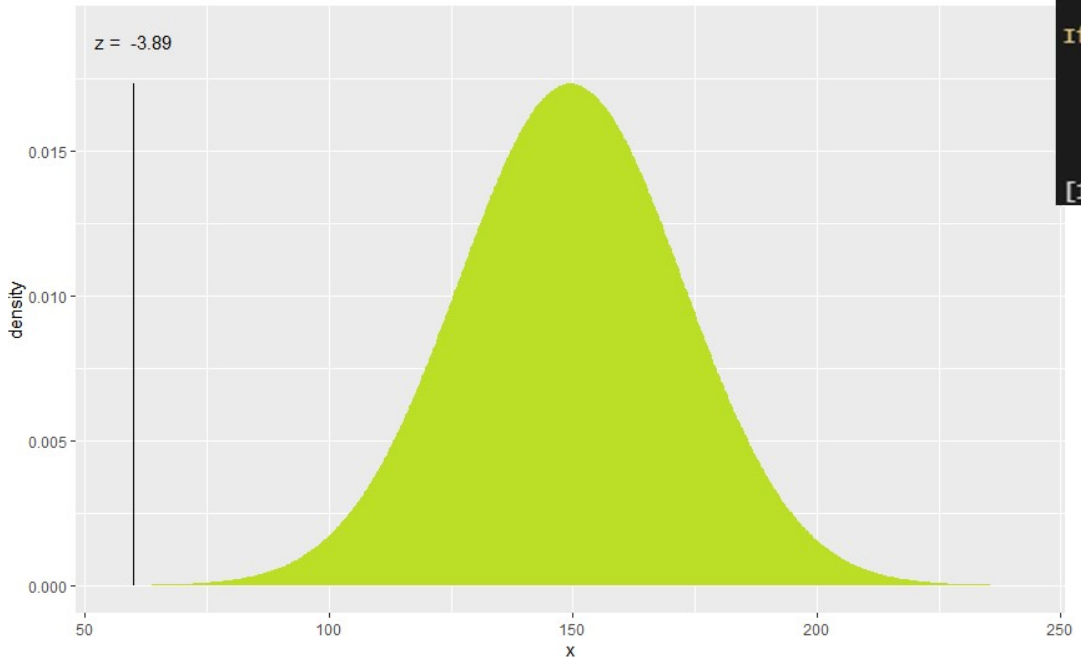
Calculate probabilities of refund




```
If X ~ N(149.5, 46.09), then

        P(X <= 29.91) = P(Z <= -2.596) = 0.004722
        P(X >  29.91) = P(Z >  -2.596) = 0.9953


[1] 0.00472


If X ~ N(149.5, 46.09), then

        P(X <= 44.86) = P(Z <= -2.271) = 0.01157
        P(X >  44.86) = P(Z >  -2.271) = 0.9884


[1] 0.0116


If X ~ N(149.5, 46.09), then

        P(X <= 59.82) = P(Z <= -1.947) = 0.02579
        P(X >  59.82) = P(Z >  -1.947) = 0.9742


[1] 0.0258
```

What happens if variability changed?





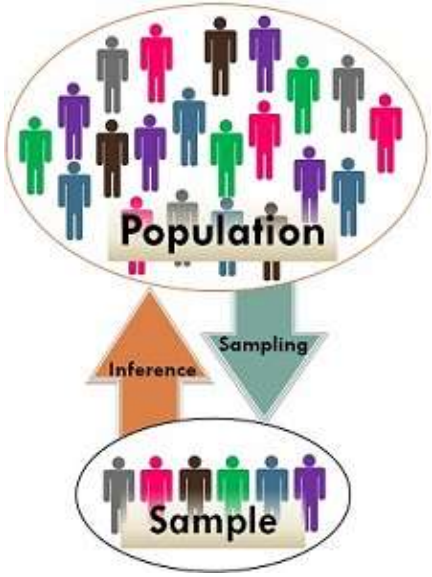

```
If X ~ N(149.5, 46.09), then

        P(X <= 59.82) = P(Z <= -1.947) = 0.02579
        P(X >  59.82) = P(Z >  -1.947) = 0.9742


[1] 0.0258
> #what happens is SD (46) is hald as large (32)?
> xpnorm(0.4*mean_snow, mean = mean_snow, sd = 0.5*sd_snow)


If X ~ N(149.5, 23.05), then

        P(X <= 59.82) = P(Z <= -3.893) = 4.943e-05
        P(X >  59.82) = P(Z >  -3.893) = 1


[1] 4.94e-05
```

Contents

- Review of Sessions 1-2
- Sampling and Inferential Statistics
- Confidence interval for the mean
- Binomial Distribution and confidence intervals for proportions
- Bootstrap estimation using **infer**

Statistics is a Science of Inference

• Statistical Inference: ✓ Predict and forecast values of ***population parameters…*** ✓ Test hypotheses about values of population parameters… ✓ Make decisions…	On basis of ***sample statistics*** derived from limited and incomplete sample information



Make generalizations about the characteristics of a ***population…***

On the basis of observations of a ***sample***, a part of a population

Inferential Statistics

- **Descriptive statistics**
  - describe the whole population (everyone we are interested in)
  - properties calculated (mean, std. dev.) are called parameters (the TRUTH)
  - Population parameters is perfect knowledge; they are so special, we reserve Greek letters (μ, σ) for them

- **Inferential statistics**
  - use a (random) sample to infer something about the population
  - properties calculated are called sample statistics (estimators), which are used to estimate parameters of the population
  - Different samples will typically have different statistics ($\bar{x}$, s)

Population Parameters

Proportion	p	% of supporters
Mean	μ	Median commute time
Difference between proportions	$p_1 - p_2$	Difference in loan default rates between North-South
Difference between means	$\mu_1 - \mu_2$	Difference in test scores between large and small classes
Intercept	α or β_0	Relationship between test scores and class size
Slope	β_1	
Standard deviation	σ	

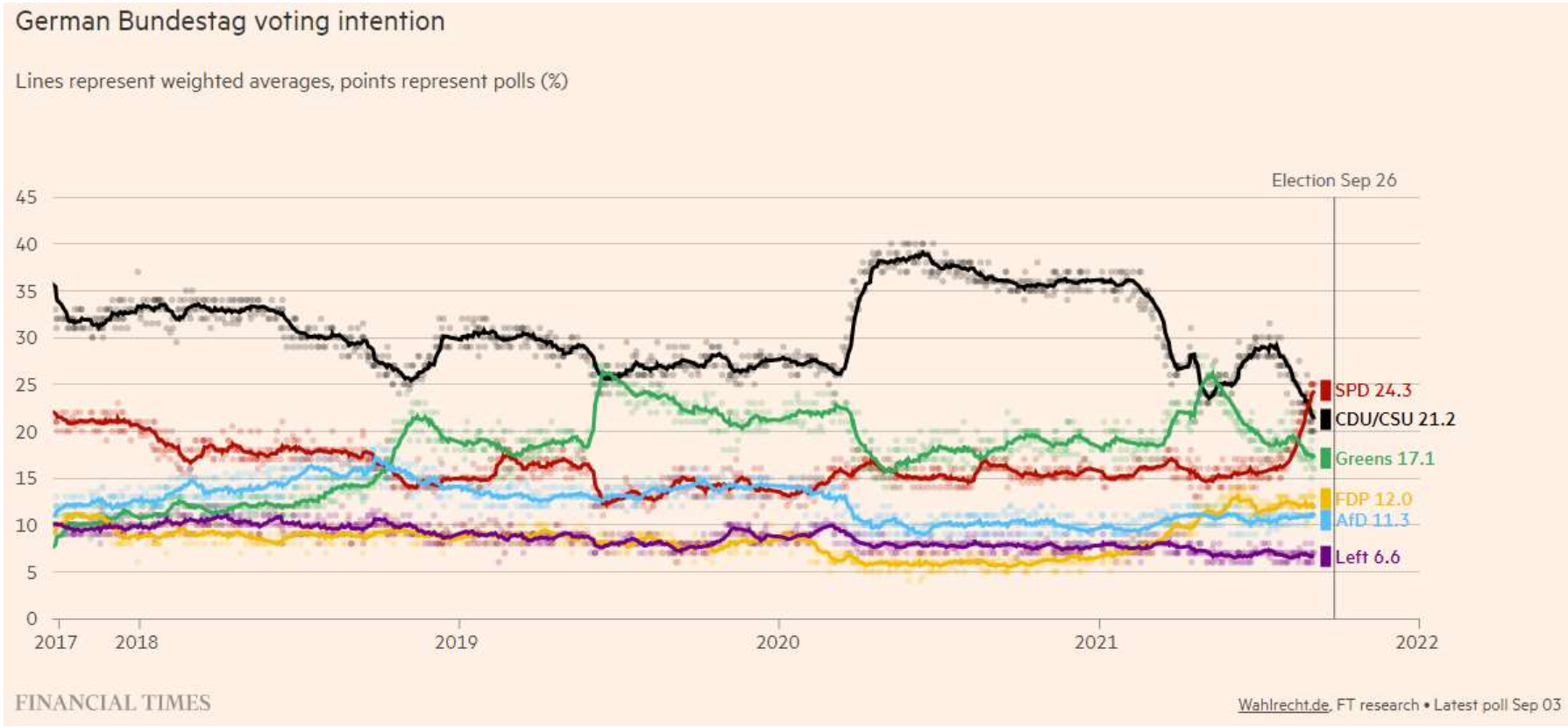
There are 'true', fixed population parameters out in the world that we would get if we measured everything and everyone. We are seldom able to measure population parameters directly, so we use INFERENCE

Examples of sampling

1. In business, a company wants
    - to find what customers think about redesigned shops (customer satisfaction survey)
    - how many employees would be prepared to work on Saturdays for an extra fee
    - what percentage of the buying public are aware of the existence of a new banking product (market survey on new product/service)
    - The proportion of time that customers are served on time (quality control)
2. In politics,
    - who is going to win the upcoming election?
    - What is the approval rate of presidents, PMs, etc.

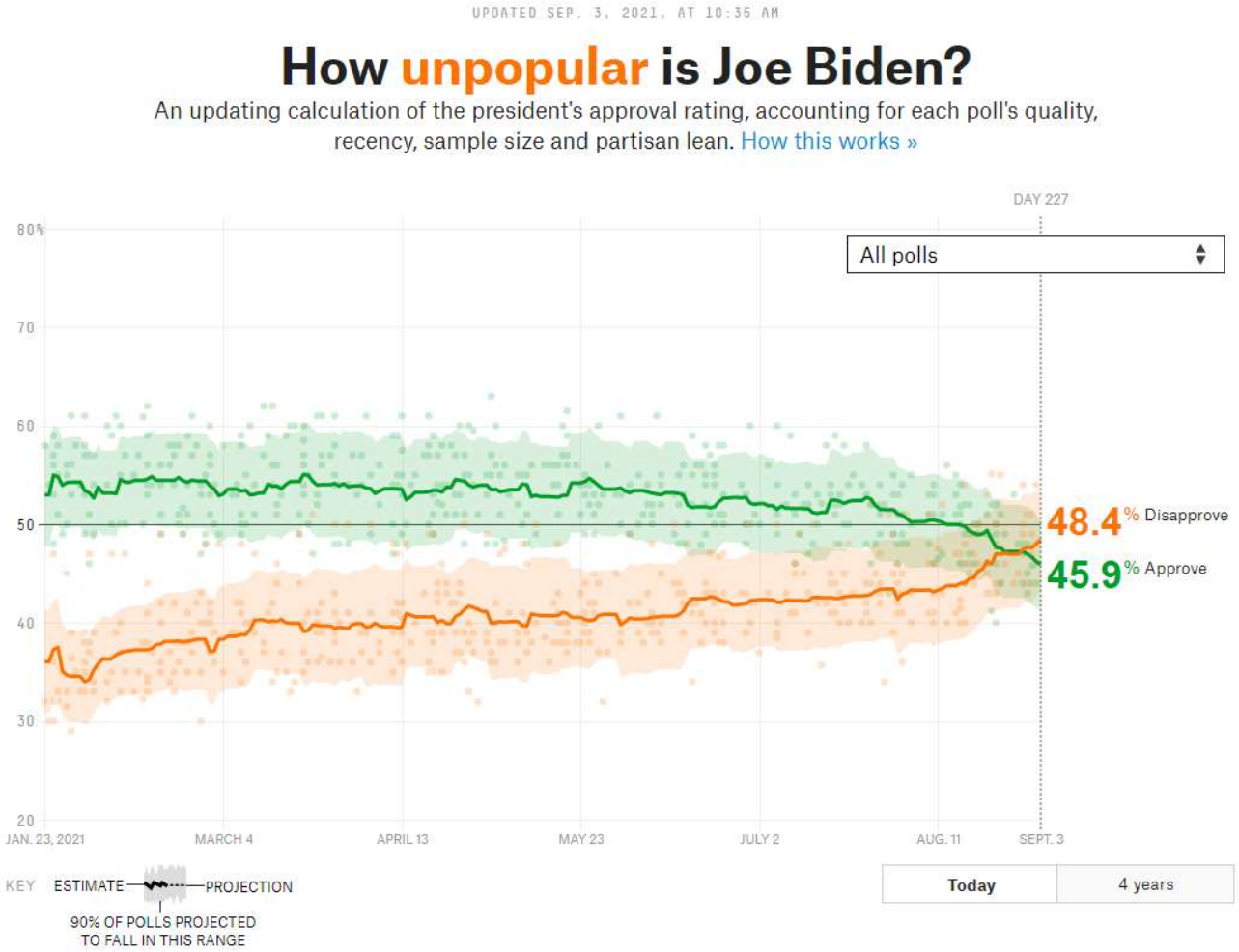
- In all of this cases dealing with the whole population is either impractical or too expensive and sampling is the only option.

German Federal Election



https://www.ft.com/content/5885e964-6d54-46ba-be63-8fb7009075f2

Approval rate of US President



DATES	POLLSTER	GRADE	SAMPLE	WEIGHT	APPROVE	DISAPPROVE	ADJUSTED	
SEP. 1-2	Ipsos	B-	1,005 A	0.69	46%	48%	45%	49%
AUG. 31-SEP. 2	Rasmussen Reports/Pulse Opinion Research	B	1,500 LV	1.30	44%	54%	47%	48%
AUG. 30-SEP. 1	Emerson College	A-	1,200 RV	1.85	46%	47%	47%	47%

Source: https://projects.fivethirtyeight.com/biden-approval-rating/

Inference

	Population Parameter	Sample statistic/ point estimate
Proportion	p	$\hat{p}$
Mean	μ	$\bar{x}$
Difference between proportions	$p_1 - p_2$	$\hat{p}_1 - \hat{p}_2$
Difference between means	$\mu_1 - \mu_2$	$\bar{x}_1 - \bar{x}_2$
Intercept	β_0	$\hat{\beta}_0$
Slope	β_1	$\hat{\beta}_1$
Standard deviation	σ	s

Use sample statistics to ***infer***, or make conclusions, about the underlying population parameters

Inferential Statistics Overview

- We are often interested in **population parameters**.
- Since complete populations are difficult (or impossible) to collect data on, we use **sample statistics** (sample mean, sample median) as **point estimates** for the unknown population parameters of interest.
- **Sample statistics** vary from sample to sample. If we take two samples, say those to the left or those to the right of the class, we will not get exactly the same sample statistics
- Quantifying how sample statistics vary provides a way to estimate **the margin of error** associated with our point estimate.

What do you want to do?

- Estimation -> Confidence intervals
- Decision -> Hypothesis test

First step: Ask the following questions

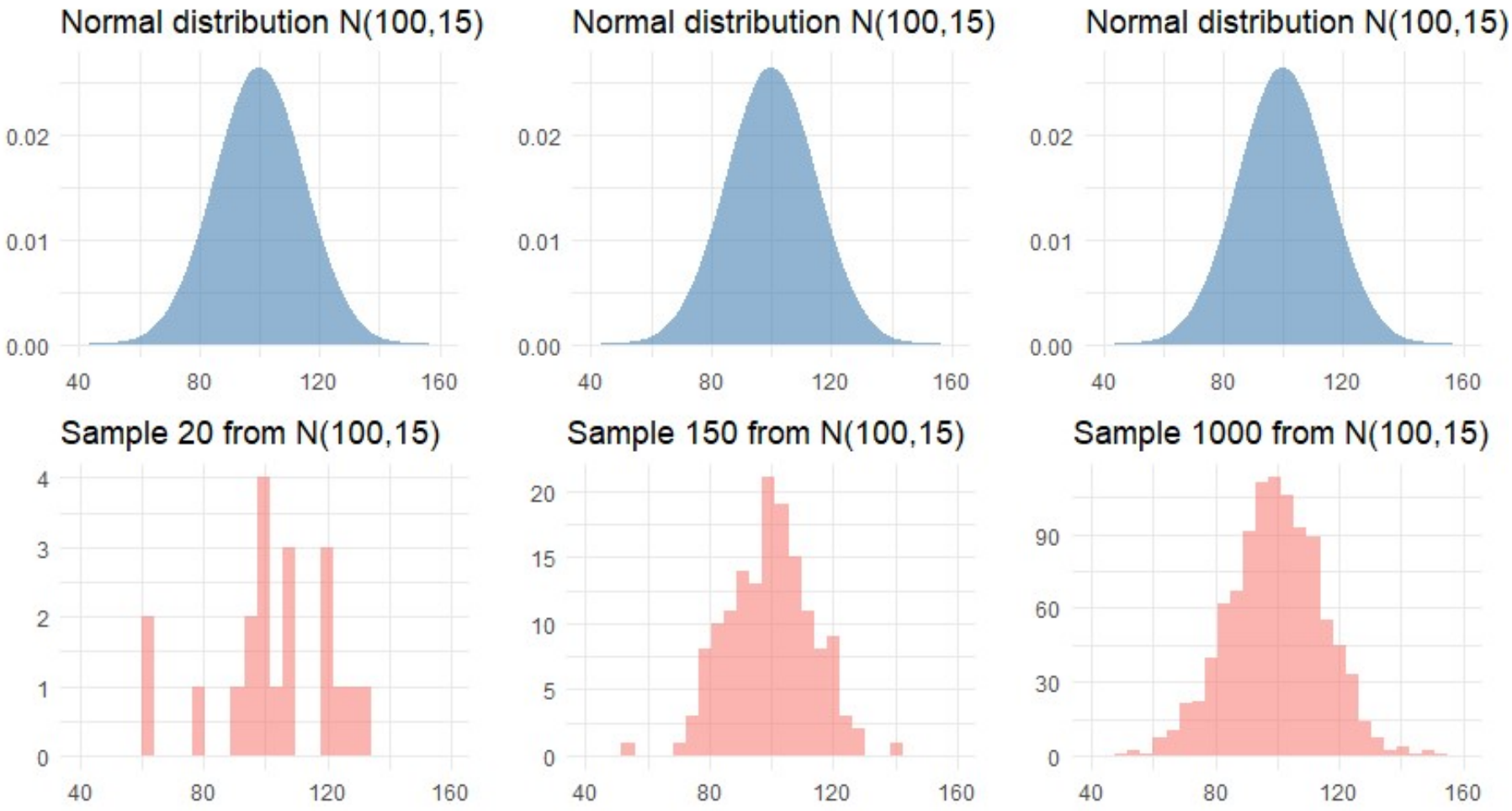
- What is the (research) question you want to answer?
- How many and what types of variables?
- What happens to your sample statistic/point estimate as you increase the size of the sample?

Contents

- Review of Sessions 1-2
- Sampling and Inferential Statistics
- Confidence interval for the mean
- Binomial Distribution and confidence intervals for proportions
- Bootstrap estimation using **infer**

Variability of a Sample

- **X** follows a Normal distribution with **μ = 100, σ = 15.**
- If we randomly sample from the population, what does our sample look like if we change sample size?



Variability of a Sample Statistic

- Every sample statistic (e.g., mean, proportion) has some variability.
- You have an average, but how different might that average be if you take another sample?

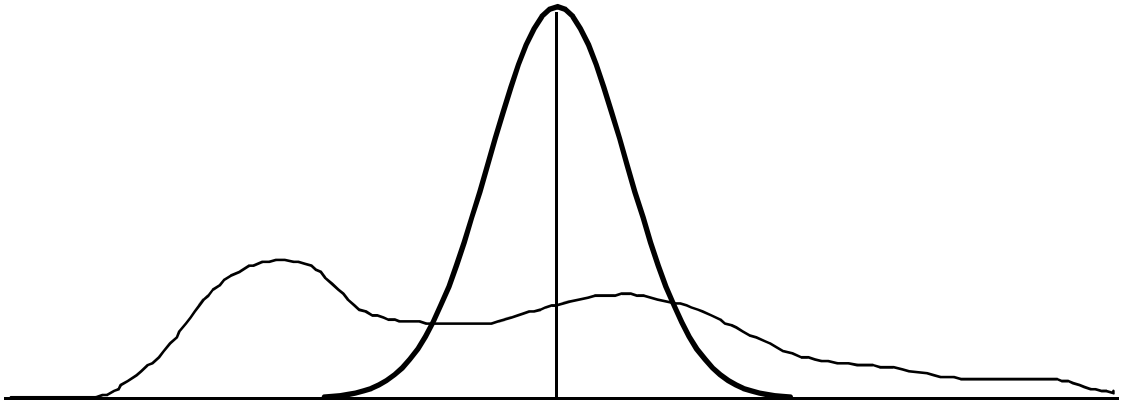
You take a random sample of 50 students and 5 are left-handed.

- If you take a different random sample of 50 students, how many would you expect to be left-handed?
  - 3 out of are left handed. Is this sample statistic (3/50) surprising?
  - 40 out of 50 are left handed. Is this sample statistic (40/50) surprising?

Two ways to estimate variability of sample statistic

1. Theory and math formulas
2. Bootstrap simulation

Central Limit Theorem

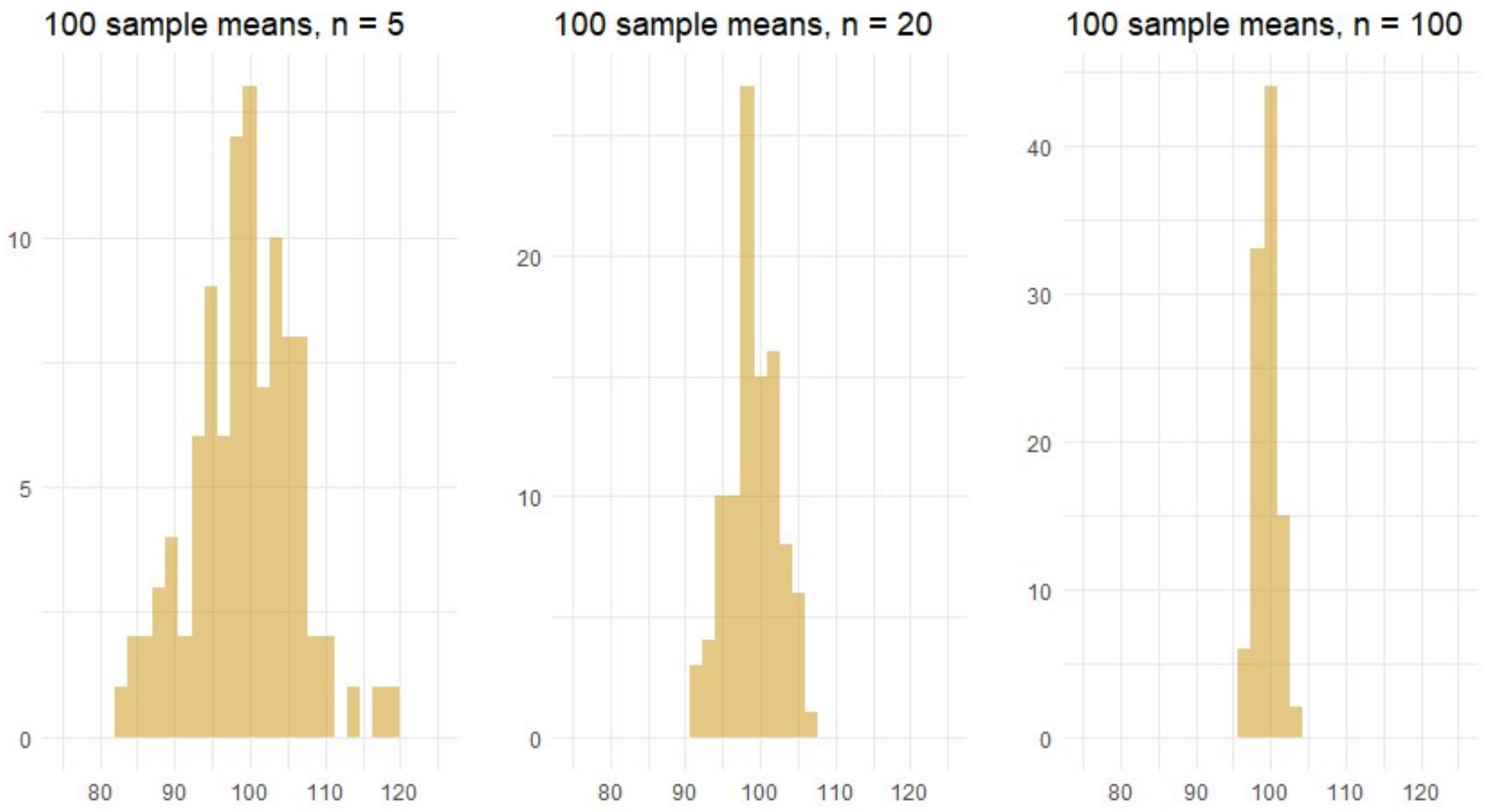


- Consider any population of observations with mean μ and standard dev. σ
- Distribution of the sample mean will approach a normal distribution as sample size *n* increases
- As *n* increases, distribution of the sample mean approaches $N(\mu, \sigma/\sqrt{n})$, a Normal distribution with mean μ and standard deviation of the sample mean, or Standard Error SE = SD/SQRT(n)

- A very powerful theorem:
  - Enables us to construct confidence intervals and do hypothesis tests for the **sample mean**
  - It is safe to apply Central Limit Theorem if the sample is reasonably large (25-30 or more)

http://onlinestatbook.com/stat_sim/sampling_dist/index.html

Central Limit Theorem



http://onlinestatbook.com/stat_sim/sampling_dist/index.html

Confidence Interval Approximation

If the parameter of interest is the population mean, and the point estimate is the sample mean, then the general formula for a **Confidence Interval (CI)** is

CI : point estimate ± margin of error = point estimate ± Z * StandardError

For a population that is Normally distributed or a large sample (n > 25), we construct a 95% confidence interval, namely a range in which we are 95% certain that the **true value**, μ, lies

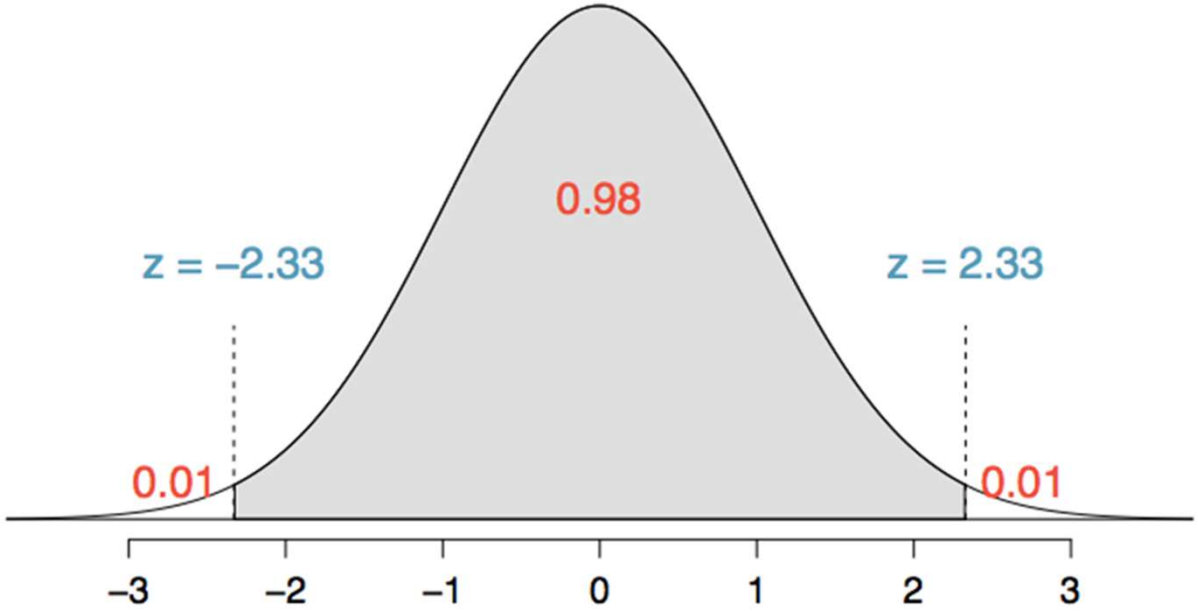
$$\bar{x} \pm 1.96 \frac{\sigma}{\sqrt{n}}$$

- 90% Confidence Interval: $Z = 1.64$
- 99% Confidence Interval: $Z = 2.58$

Communicating uncertainty

Which of the Z scores below is the appropriate z* when calculating a 98% confidence interval?

(a) Z = 2.05
(b) Z = 1.96
(c) Z = 2.33
(d) Z = -2.33
(e) Z = -1.65




```
> qnorm(0.99)
[1] 2.326348
```

Communicating uncertainty

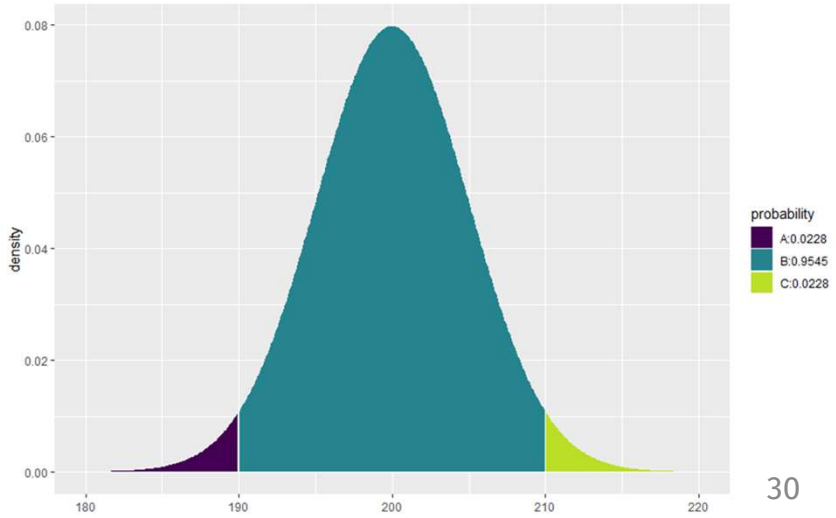
You have recently joined the marketing department of Apricot, a company that produces a mobile phone that will compete with the iPhone. Your task is to find out how much people are willing to pay on average for this new mobile phone, which is thinner and lighter than the iPhone and has 256GB

- A survey of n=100 people yields an average amount of £200, or $\bar{x} = 200$
- The standard deviation of the amount is known to be £50, $\sigma = 50$
- You know your answer is not "the truth". How can you **communicate the uncertainty** behind your number?

The CLT tells us the sample mean follows a Normal distribution.
The approximate 95% confidence interval is defined as **point estimate ± 2 x SE**
In this case, the point estimate is $\bar{x} = 200$

The SE is $\frac{\sigma}{\sqrt{n}} = \frac{50}{\sqrt{100}} = 5$

The CI is 200 ± 2*5= [190,210]

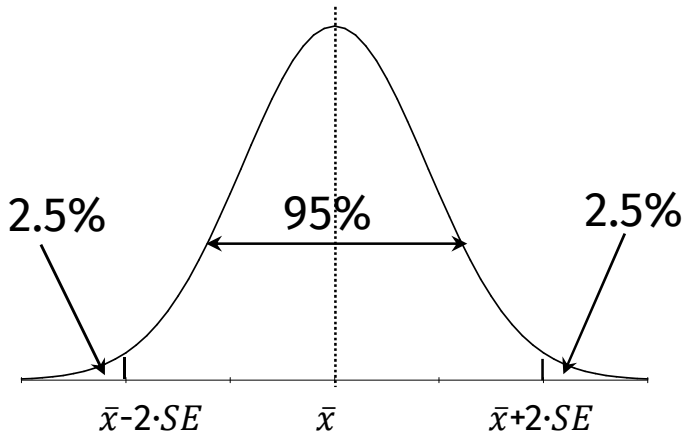


Effect of sample size

Sample of size 100

Estimated mean: $\bar{x} = 200$

$SE = \frac{50}{\sqrt{100}} = 5$

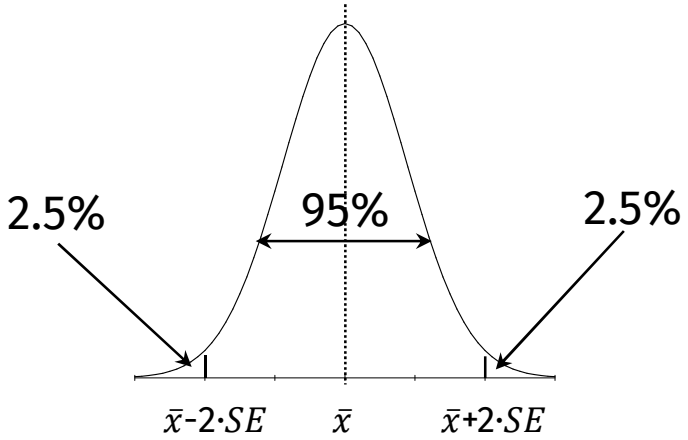


You can be 95% confident that the real value is in the range:
[200 – 2 · 5, 200 + 2 · 5]
≈ [190, 210]

Sample of size 400

Estimated mean: $\bar{x} = 200$

$SE = \frac{50}{\sqrt{400}} = 2.5$



You can be 95% confident that the real value is in the range:
[200 – 2 · 2.5, 200 + 2 · 2.5]
≈ [195, 205]

CI for a large sample

For a Normal population or a large sample (n > 30)

- **95% confidence interval**
  - a range in which we are 95% certain that the true value, μ, lies

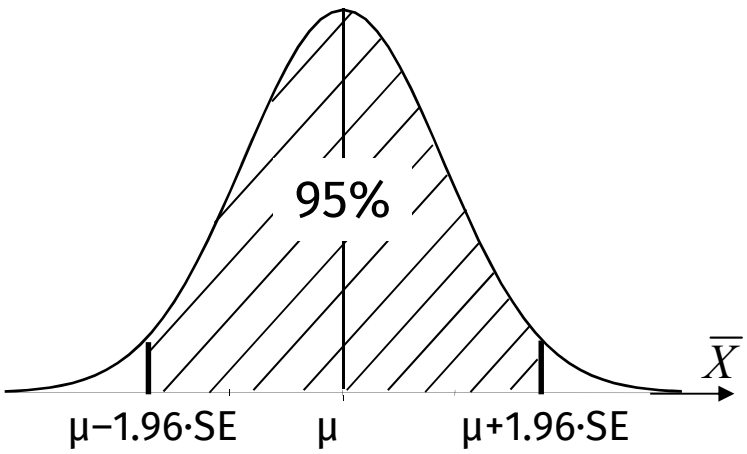
(Critical) Z - value

$$\bar{x} \pm \overbrace{\mathbf{1.96}}^{} \cdot SE$$

where

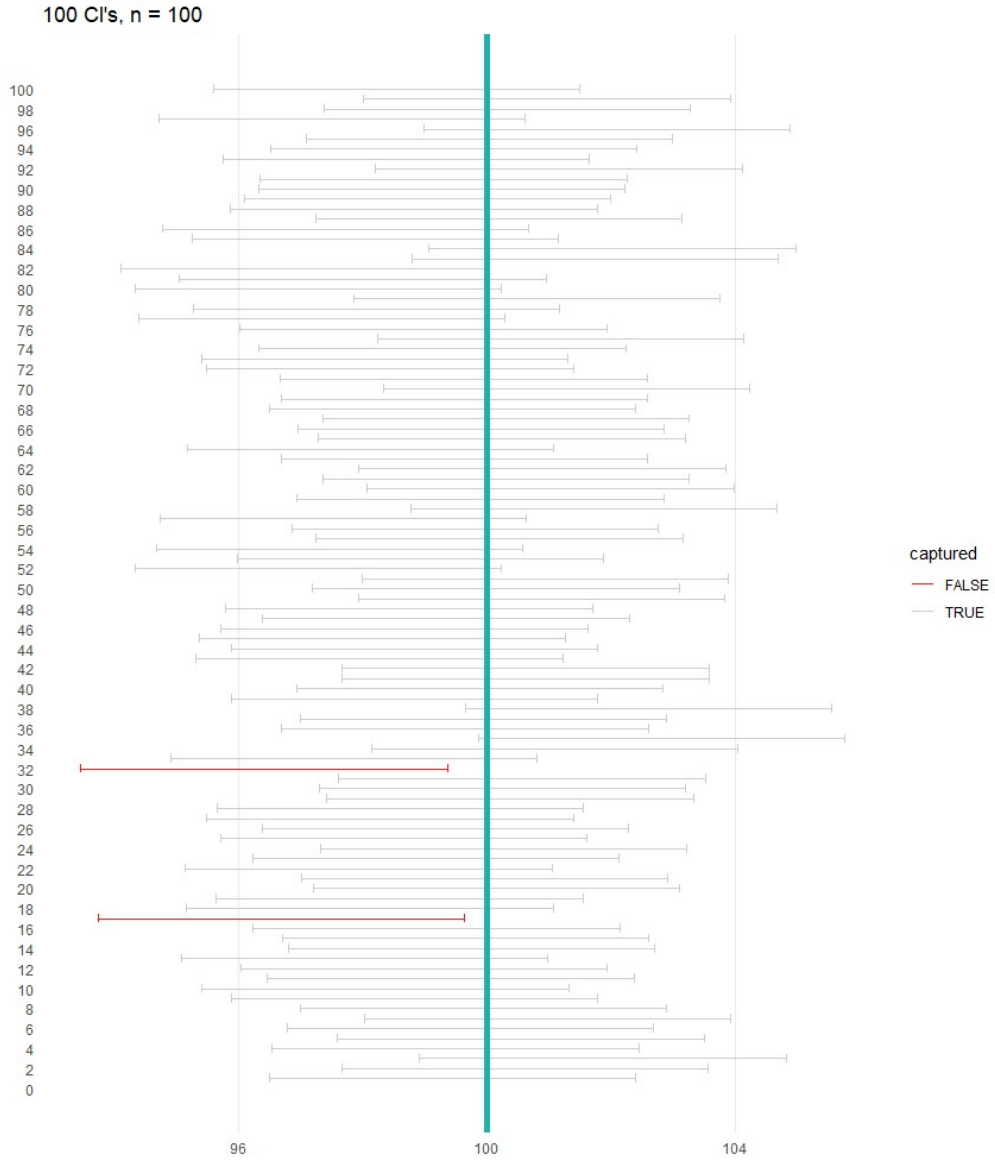
$\bar{x}$ is the sample average

$SE \cong s/\sqrt{n}$ is the standard error

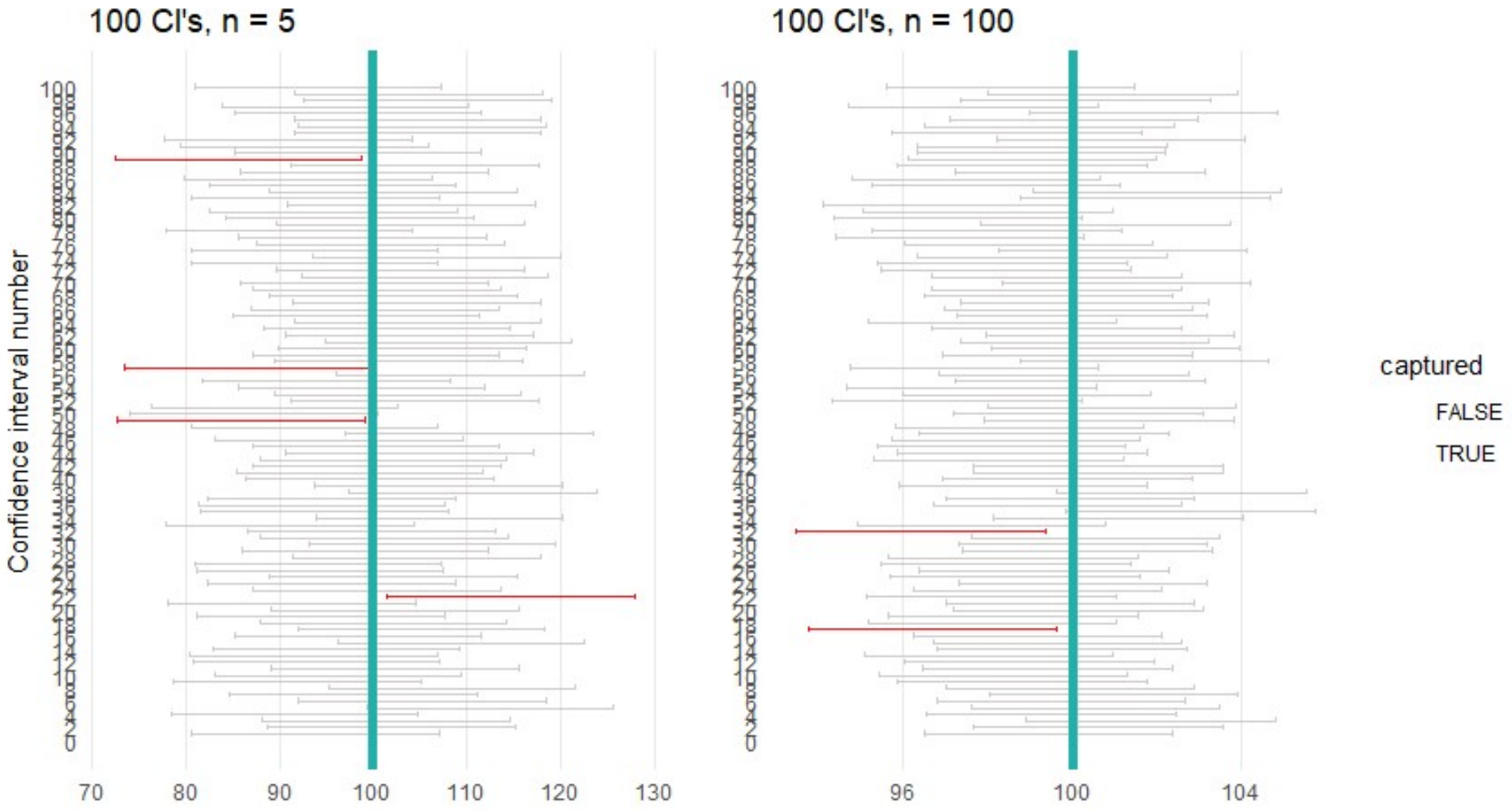


What does 95% confidence mean?

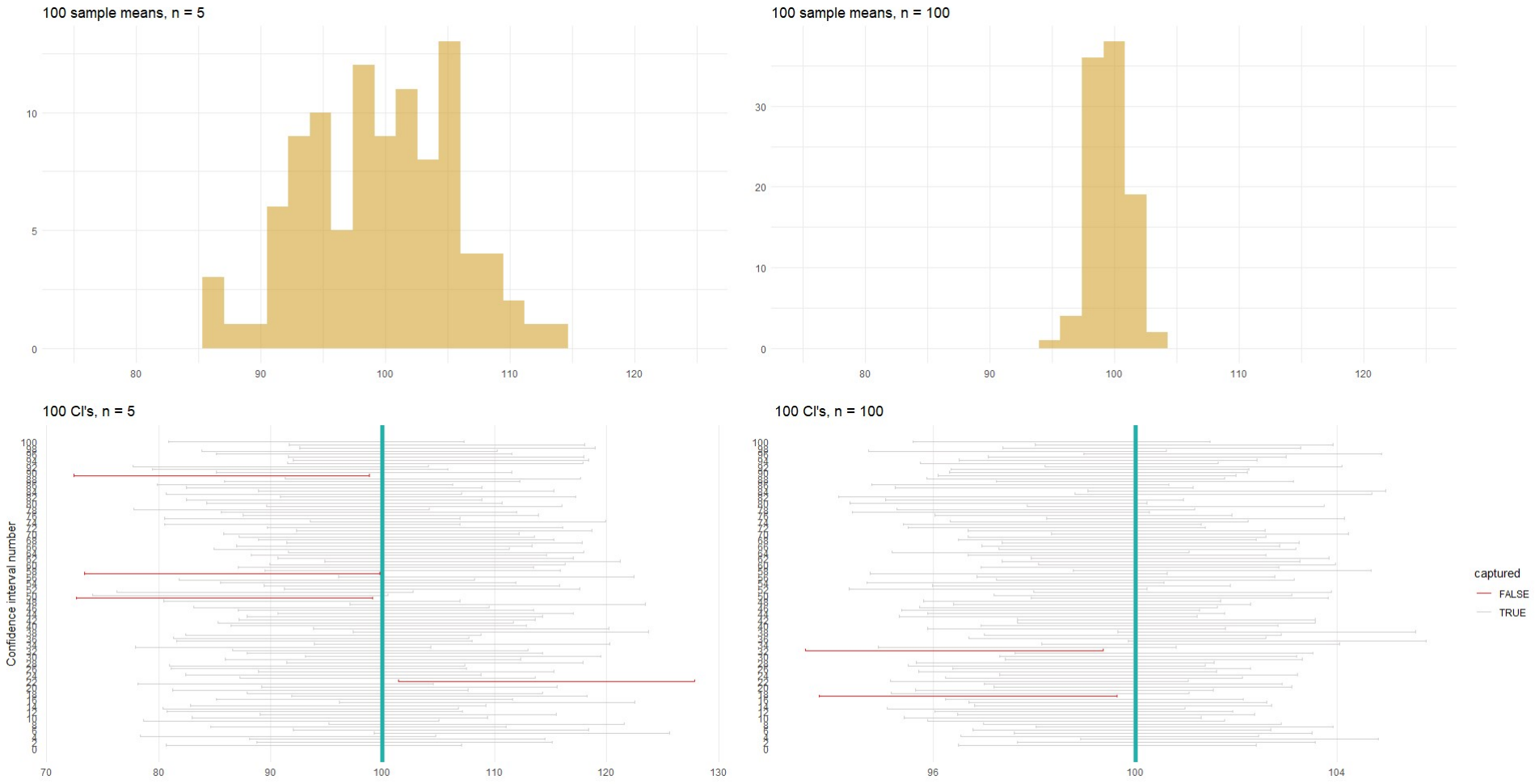
- Confidence Intervals are like a net: If we took 100 samples, about 95 of those intervals would contain the true population **parameter**.
- Confidence Intervals allow us to draw inferences from data. Given a sample, what can we infer the 'true' population parameter?
- By using a 95% interval rule, we hope that **in the long run**, we will not be wrong too often. We allow ourselves a 5% probability of making a mistake.
- This 95% interval rule is a rule to govern behaviour in the long run. It tells us nothing about the one, current, specific interval we have constructed which may, or may not, contain the true, population value
- Tempting, but wrong, to say "We're 95% sure that the population parameter is X"



CI width depends on confidence level & sample size



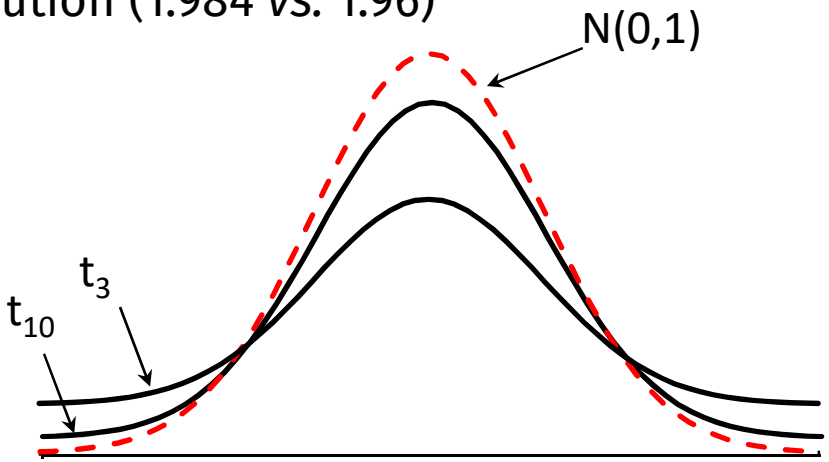
Central Limit Theorem



http://onlinestatbook.com/stat_sim/sampling_dist/index.html

When σ is uknown, use *t-distribution*

- We use t-distribution instead of Z when we don't know the standard deviation of the population. Most of the time, this is the case, so z-tests are rarely calculated in social research.
- The sample mean follows a t-Distribution
- Exact when the population is normally distributed, approximate if not
- The t-distribution has thicker tails than the Normal
- The shape depends on the “degrees of freedom” n – 1
- When n > 100, close to Normal distribution (1.984 *vs.* 1.96)



C.I. with t-distribution

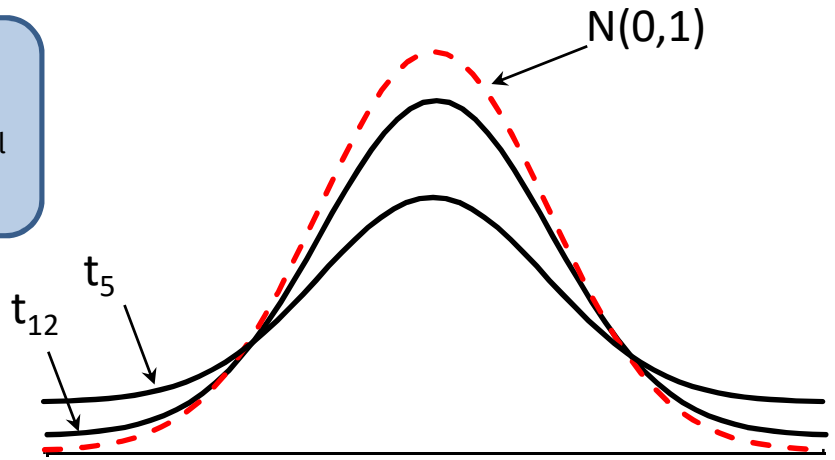
sample_size	t_critical
5	2.7764
6	2.5706
7	2.4469
8	2.3646
9	2.3060
10	2.2622
11	2.2281
12	2.2010
13	2.1788
14	2.1604
15	2.1448
16	2.1314
17	2.1199
18	2.1098
19	2.1009
20	2.0930
21	2.0860
22	2.0796
23	2.0739
24	2.0687
25	2.0639
26	2.0595
27	2.0555
28	2.0518
29	2.0484
30	2.0452

$$\left[\bar{x} - t\frac{s}{\sqrt{n}}, \bar{x} + t\frac{s}{\sqrt{n}}\right]$$

t (critical t-value): similar to z-value, but uses the t-distribution

```
t_critical <- tibble(
  sample_size = c(5:100),
  t_critical = qt(0.975, sample_size-1)
)
```

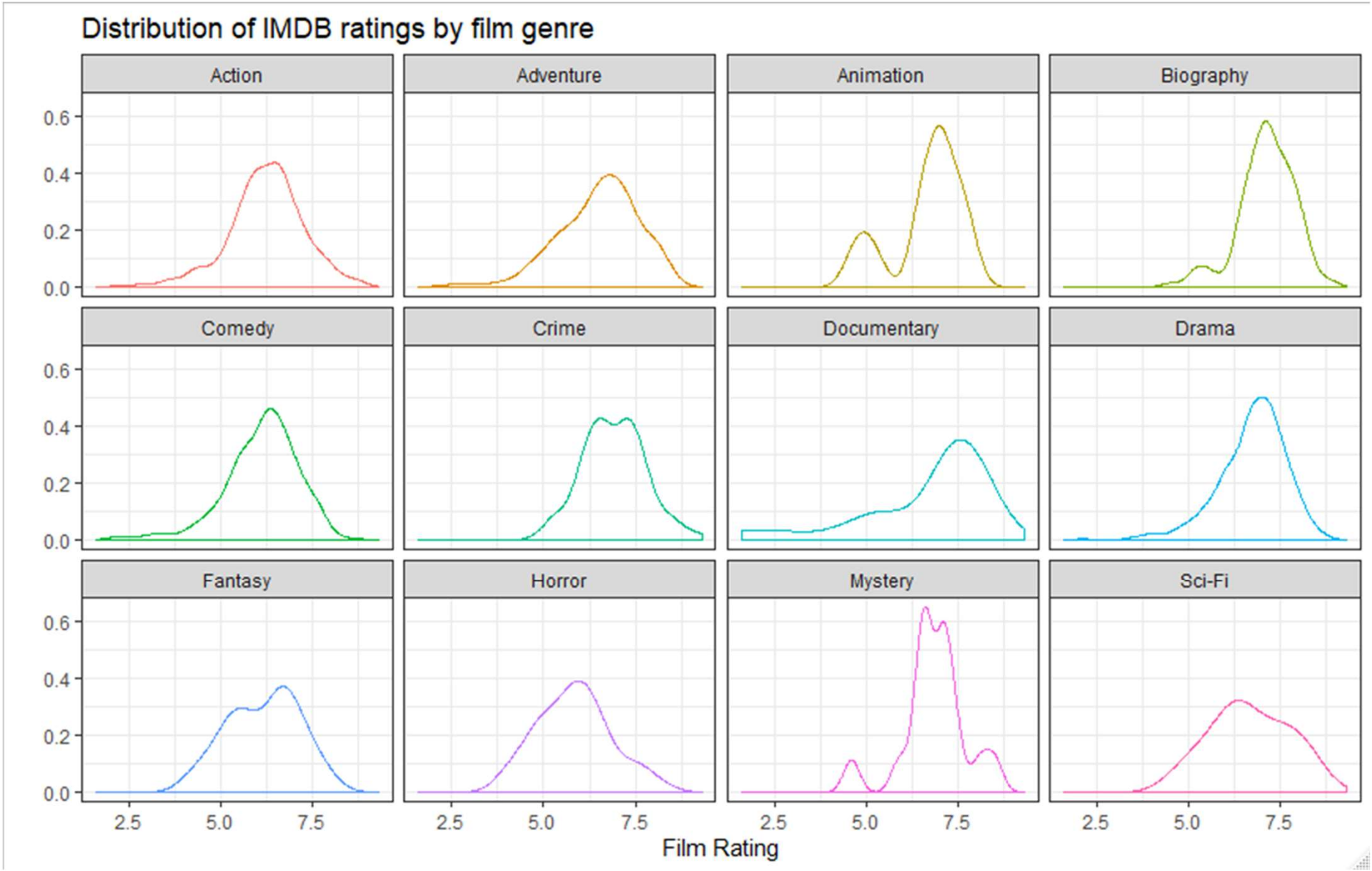
- Use 97.5% cumulative of the ***t-distribution*** with (n-1) degrees of freedom
- This leaves 2.5% outside on one tail
- Both tails leave out 2.5+2.5=5% and thus we get our 95% CI



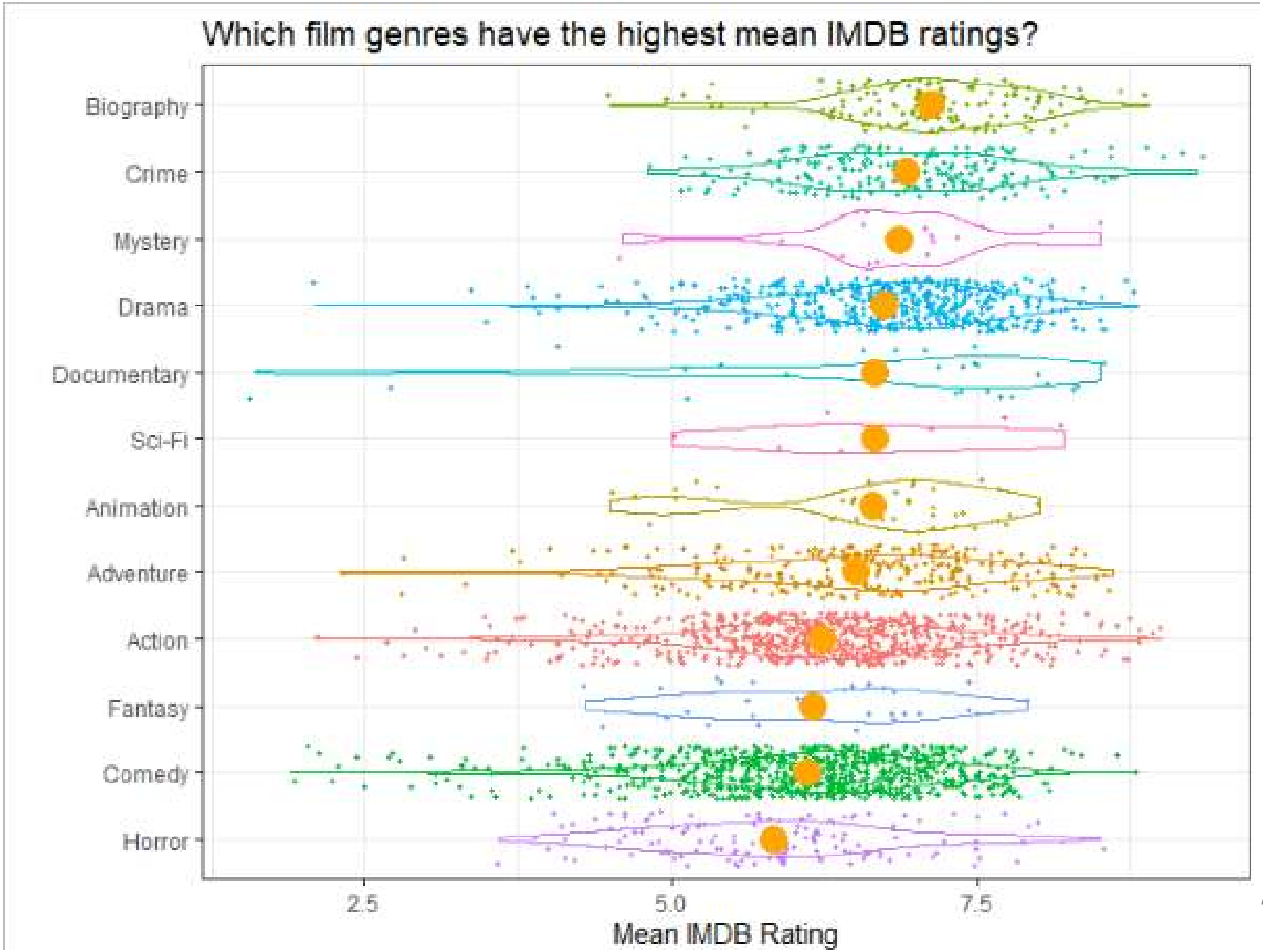
Approximation formulas for Inference

Parameter	Distribution	Conditions	Standard Error
Proportion	Normal	All counts at least 10 $np \geq 10$, $n(1-p) \geq 10$	$\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$
Difference in Proportions	Normal	All counts at least 10 $n_1p_1 \geq 10$, $n_1(1-p_1) \geq 10$, $n_2p_2 \geq 10$, $n_2(1-p_2) \geq 10$	$\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1}+\frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}$
Mean	t, df = $n-1$	$n \geq 30$ or data normal	$\sqrt{\frac{s^2}{n}}=\frac{s}{\sqrt{n}}$
Difference in Means	t, df = smaller of n_1-1, n_2-1	$n_1 \geq 30$ or data normal, $n_2 \geq 30$ or data normal	$\sqrt{\frac{s_1^{\,2}}{n_1}+\frac{s_2^{\,2}}{n_2}}$

IMDB ratings by film genre



Visualisation of ratings by genre



On average, does one genre have better ratings?

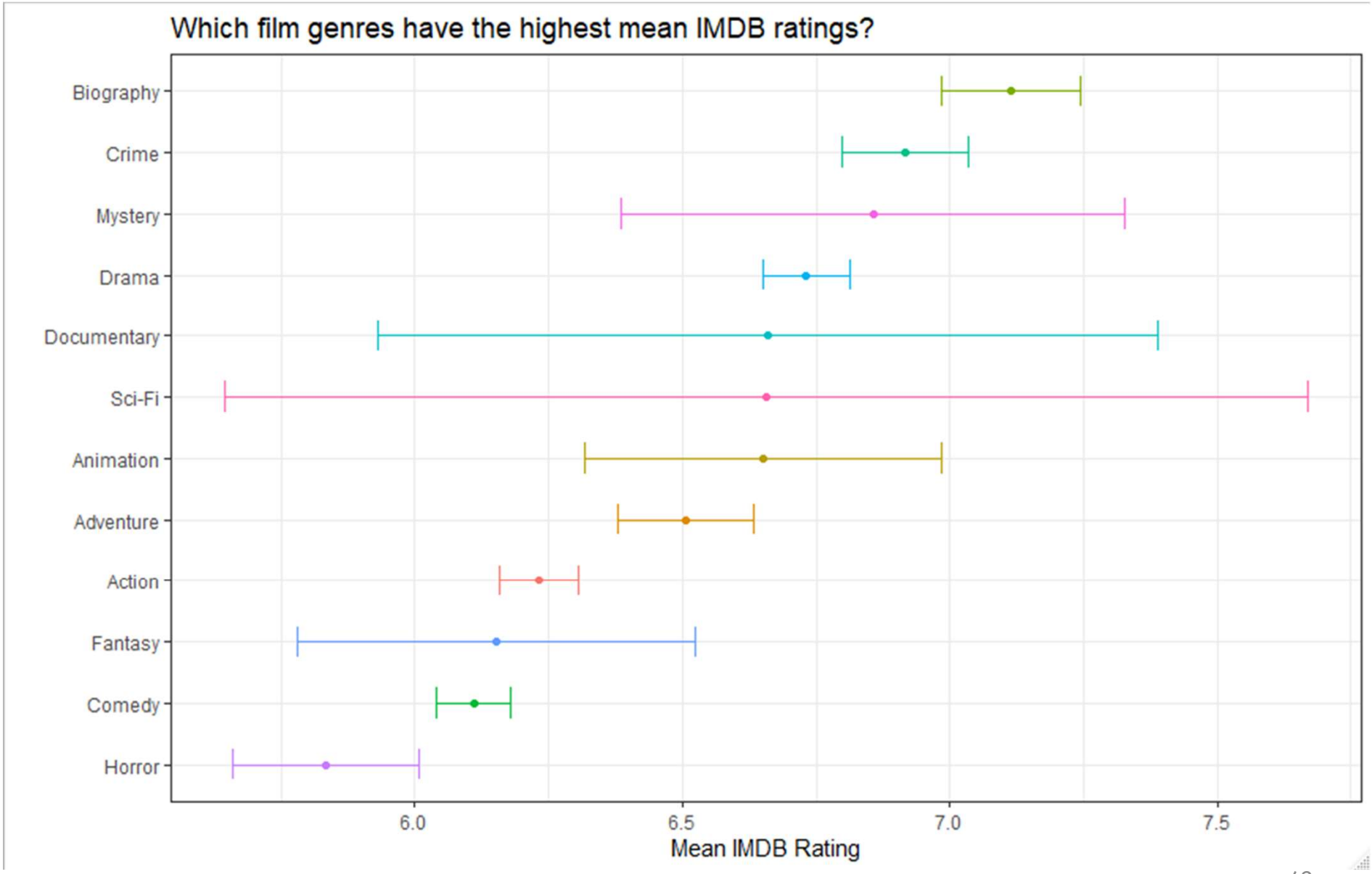
```
genre_formula_ci <- movies %>%
  filter(genre %in% genres_to_choose) %>%
  group_by(genre) %>%
  summarise(mean_rating = mean(rating),
            sd_rating = sd(rating),
            count = n(),
            # get t-critical value with (n-1) degrees of freedom
            t_critical = qt(0.975, count-1),
            se_rating = sd(rating)/sqrt(count),
            margin_of_error = t_critical * se_rating,
            rating_low = mean_rating - margin_of_error,
            rating_high = mean_rating + margin_of_error
  ) %>%
  arrange(desc(mean_rating))
```

```
> genre_formula_ci
# A tibble: 12 x 9
```

	genre	mean_rating	sd_rating	count	t_critical	se_rating	margin_of_error	rating_low	rating_high
	<chr>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Biography	7.11	0.760	135	1.98	0.0654	0.129	6.98	7.24
2	Crime	6.92	0.849	202	1.97	0.0598	0.118	6.80	7.03
3	Mystery	6.86	0.882	16	2.13	0.220	0.470	6.39	7.33
4	Drama	6.73	0.917	498	1.96	0.0411	0.0807	6.65	6.81
5	Documentary	6.66	1.77	25	2.06	0.353	0.729	5.93	7.39
6	Sci-Fi	6.66	1.09	7	2.45	0.413	1.01	5.65	7.67
7	Animation	6.65	0.968	35	2.03	0.164	0.333	6.32	6.98
8	Adventure	6.51	1.09	288	1.97	0.0645	0.127	6.38	6.63
9	Action	6.23	1.03	738	1.96	0.0379	0.0745	6.16	6.31
10	Fantasy	6.15	0.959	28	2.05	0.181	0.372	5.78	6.53
11	Comedy	6.11	1.02	848	1.96	0.0351	0.0690	6.04	6.18
12	Horror	5.83	1.01	131	1.98	0.0886	0.175	5.66	6.01

The bigger the sample (***count***), the faster does ***t-critical*** approach 1.96

Visualisation of CIs derived using formula



Contents

- Review of Sessions 1-2
- Sampling and Inferential Statistics
- Confidence interval for the mean
- Binomial Distribution and confidence intervals for proportions
- Bootstrap estimation using **infer**

Binomial Distribution

- A discrete distribution
- Applies to measurements/events which have only two outcomes
  - "Success" with probability *p*, or "Failure" with probability *(1-p)*

Examples:

- Flipping a coin
- Reliability inspection schemes (good / defective)
- Opinion surveys (agree/ disagree, yes/ no)
- Selling (sale / no sale)

The terms *success* and *failure* are simply statistical terms, and do not have positive or negative implications.
In a production setting, finding a defective product may be termed a "success," although it is not a positive result.

Binomial Distribution

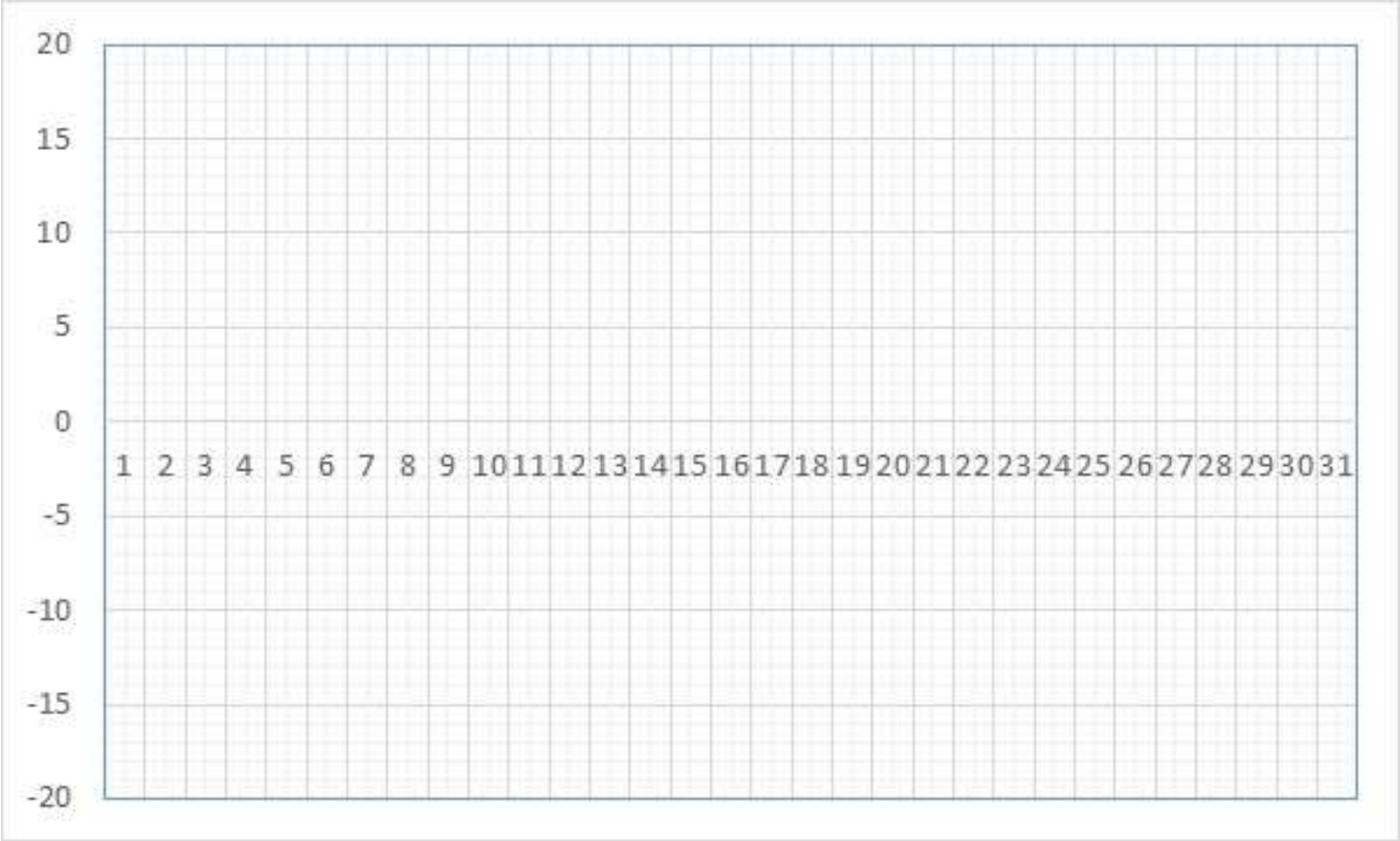
- *n* is the number of trials (*e.g., the number of times we flip the coin*)
- *p* is the probability of success (*the probability of getting a heads*)- 50% in a fair coin


```
> rflip(30)

Flipping 30 coins [ Prob(Heads) = 0.5 ] ...

H H T H H H H T T T H H H T H H T H T H T H T H T T T T T H

Number of Heads: 16 [Proportion Heads: 0.533333333333333]

> rflip(30)

Flipping 30 coins [ Prob(Heads) = 0.5 ] ...

T T T T T T H H H T T T T T H H T T H H T T H H H H T H T H

Number of Heads: 13 [Proportion Heads: 0.433333333333333]

> rflip(30)

Flipping 30 coins [ Prob(Heads) = 0.5 ] ...

H T T T H H T H T H H T H H H T T T H H H H H T T H H H T T

Number of Heads: 17 [Proportion Heads: 0.566666666666667]
```

Flipping coins and Random walks

How many heads would you expect after 10, 20, 30, 40, or 50 trials?
Suppose I start at position 0 and I flip a coin N times. If I get a head I go to whatever my position is +1 , if I get a tail, I go to my current position -1. 3 heads in a row would take me to 3, if I get a tail, I go to 3-1=2, etc.

If we plot a graph and X is the number of times we flip a coin, and Y your expected position, what does graph look like? Break in groups. Flip a coin for 30 times and plot your position below. Do you see any patterns?



Dealing with proportions: the Binomial Distribution

- The appropriate distribution to use when dealing with proportions (or *percentages*, %) of positive/negative responses is the binomial distribution
- Generally we are interested in the **number (n)** or the **sample proportion $\hat{p}$ (X/n)** within the overall sample who give a particular response:
- this number is subject to sampling error. If we take another sample, we will not get exactly the same proportion
- treat the result as a random variable, say 'X' and say it is *binomially distributed* with certain values for 'p' (the chance of the outcome we are interested in) and 'n' the sample size
- for short-hand, write this as X ~ B(n, $\hat{p}$)
- As long as the number of "successes" (*n* $\hat{p}$) and the number of "failures" (*n(1-* $\hat{p}$*)*) are both greater than 5, we can *approximate* the binomial distribution using the Normal distribution.

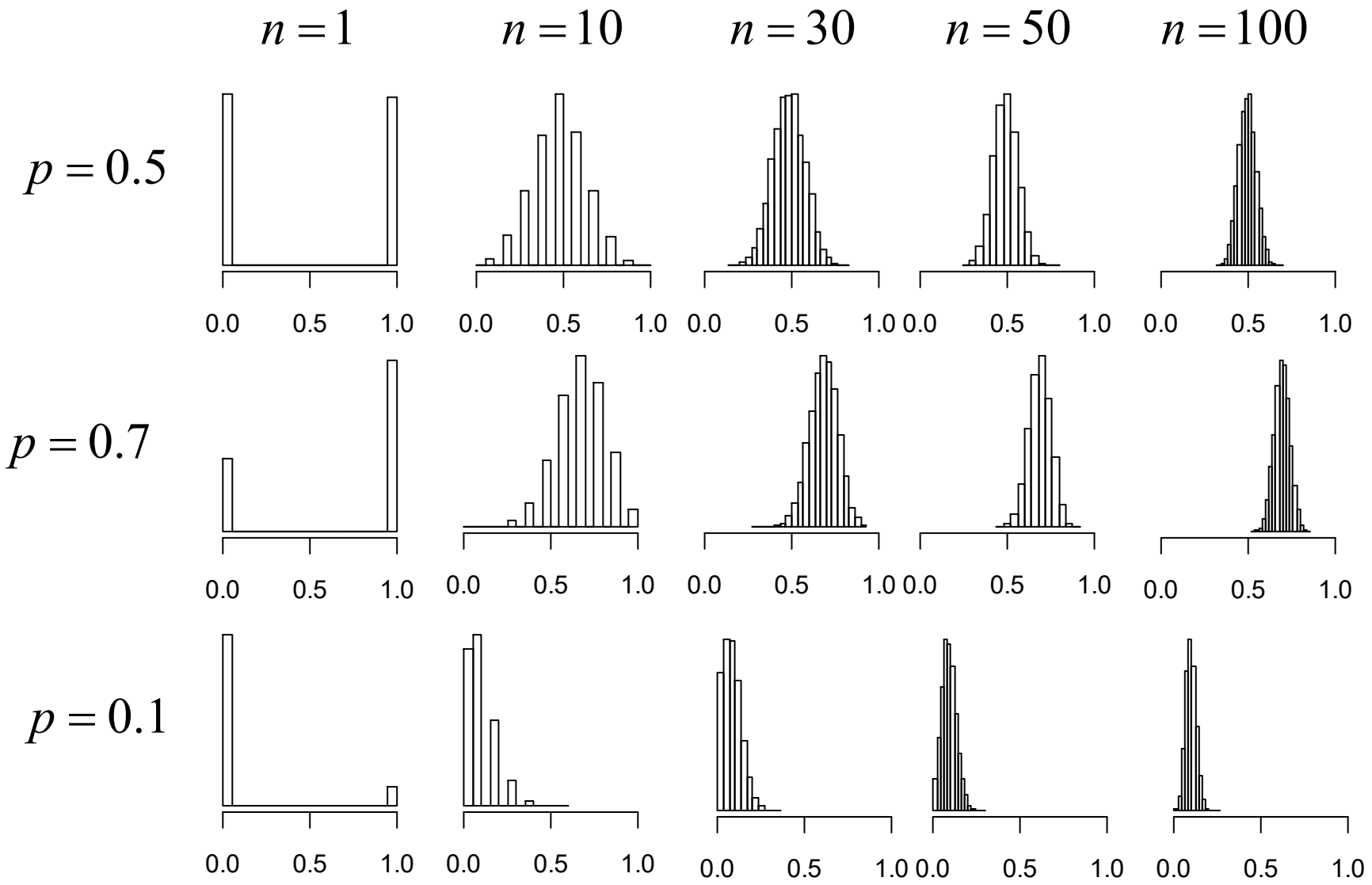
Binomial Distribution: independent events with two outcomes

Can approximate for large samples using normal distribution

Rule of thumb: *n* $\hat{p}$ *> 10* and *n(1-* $\hat{p}$*)> 10*

Proportion of occurrences: $$\hat{p} = \frac{X}{n} \sim N\left(\hat{p}, \sqrt{\frac{\hat{p}.(1-\hat{p})}{n}}\right)$$

Distribution of $\hat{p}$



Approximation formulas for Inference

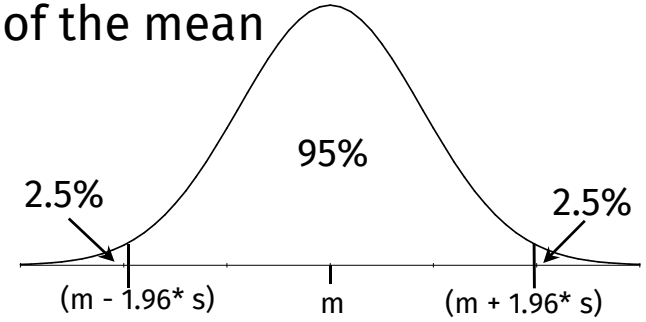
Parameter	Distribution	Conditions	Standard Error
Proportion	Normal	All counts at least 10 $np \geq 10$, $n(1-p) \geq 10$	$\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$
Difference in Proportions	Normal	All counts at least 10 $n_1p_1 \geq 10$, $n_1(1-p_1) \geq 10$, $n_2p_2 \geq 10$, $n_2(1-p_2) \geq 10$	$\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1}+\frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}$
Mean	t, df = $n-1$	$n \geq 30$ or data normal	$\sqrt{\frac{s^2}{n}}=\frac{s}{\sqrt{n}}$
Difference in Means	t, df = smaller of n_1-1, n_2-1	$n_1 \geq 30$ or data normal, $n_2 \geq 30$ or data normal	$\sqrt{\frac{s_1^{\,2}}{n_1}+\frac{s_2^{\,2}}{n_2}}$

Confidence Interval for Proportions

- We want to create a "margin of error" around our sample estimate
  - idea is that we will construct a range which will contain the true, population value *most of the time*
- Sample proportion $\hat{p}$ can be approximated with a Normal distribution

$$\hat{p} = \frac{X}{n} \sim N\left(\hat{p}, \sqrt{\frac{\hat{p}.(1-\hat{p})}{n}}\right)$$

- for Normal distribution we know there is a 95% probability of being within 1.96 standard deviations of the mean



- thus 95% Confidence Interval is simply [m - z * s, m + z * s] where z = 1.96
  - for 90% CI use same approach with z = 1.64
  - for 98% CI use z = 2.33, etc.
- larger sample size 'n' has effect of reducing standard deviation 's'

Confidence Interval for proportions

G4 | =SQRT(F4*(1-F4)/D4)

	A	B	C	D	E	F	G	H	I
1	***homeopathyWorks***	No	Yes	Total		**Pr(Yes)**	**SE**	**Lower 95**	**Upper 95**
4	Total	107	91	198		0.46	0.04	0.39	0.53
5									
6	***liedAboutAge***	No	Yes	Total		**Pr(Yes)**	**SE**	**Lower 95**	**Upper 95**
9	Total	73	125	198		0.63	0.03	0.56	0.70
10									
11	***marijuana***	No	Yes	Total		**Pr(Yes)**	**SE**	**Lower 95**	**Upper 95**
14	Total	86	109	195		0.56	0.04	0.49	0.63

$$p^* = \frac{91}{198} \approx 0.46$$

$$SE = \sqrt{\frac{p^* \cdot (1 - p^*)}{n}} = \sqrt{\frac{0.46 \cdot 0.52}{198}} = \sqrt{\frac{0.2392}{198}} \approx 0.035$$

$$CI = 0.46 \pm 1.96 * 0.035 \approx 0.46 \pm 0.07 \approx [0.39, 0.53]$$

> prop.test(k,n) creates a CI when the sample proportion is k out of n

```
> prop.test(91,198)

        1-sample proportions test with continuity correction

data:  91 out of 198, null probability 0.5
X-squared = 1.1364, df = 1, p-value = 0.2864
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.3891592 0.5316260
sample estimates:
       p
0.459596
```

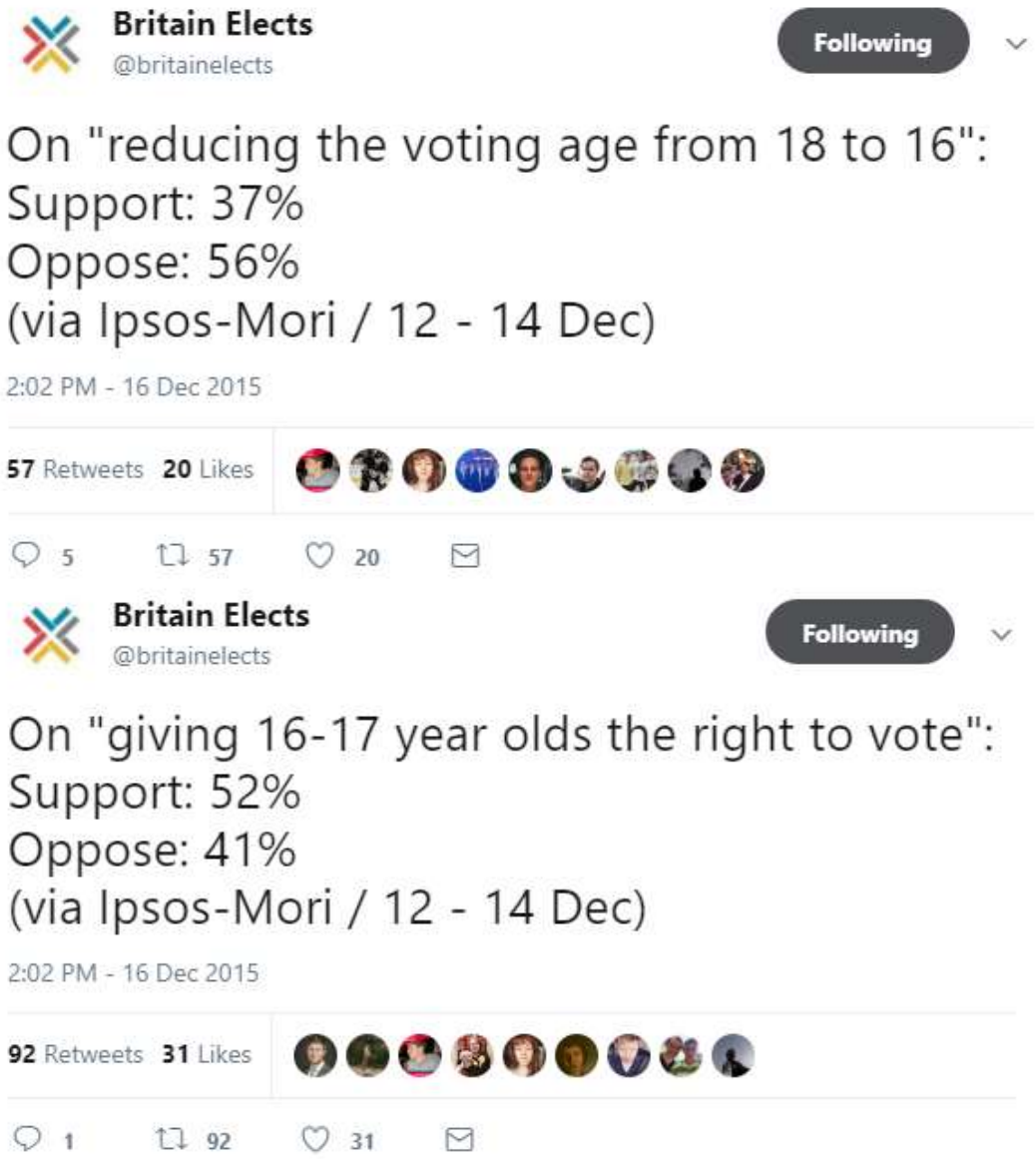
Power of question wording (1/2)

Consider the following two questions, on whether you support

1. *Reducing the voting age from 18 to 16*
2. *Giving 16-17 year olds the right to vote*

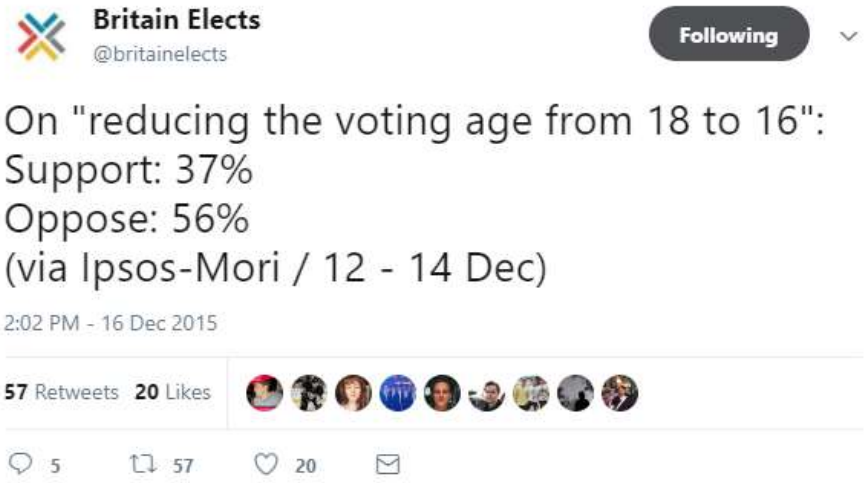
Should you expect any difference in the proportion (%) of people supporting (1) or (2)?

Power of question wording (2/2)



Source: https://twitter.com/britainelects/status/677096408424542208, https://twitter.com/britainelects/status/677096380884779009

Confidence intervals for % *supporting* assuming *n = 1,000*



$$Support = \frac{0.37}{0.37 + 0.56} = 0.398$$

$$SE = \sqrt{\frac{p*(1-p)}{n}} = \sqrt{\frac{0.398*(1-0.398)}{1{,}000}} = 0.015$$

$$CI = 0.398 \pm 1.96 * 0.015 = [0.367, 0.427]$$

$$Support = \frac{0.52}{0.52 + 0.41} = 0.559$$

$$SE = \sqrt{\frac{p*(1-p)}{n}} = \sqrt{\frac{0.559*(1-0.559)}{1{,}000}} = 0.016$$

$$CI = 0.559 \pm 1.96 * 0.016 = [0.527, 0.590]$$

Source: https://twitter.com/britainelects/status/677096408424542208, https://twitter.com/britainelects/status/677096380884779009

Contents

- Review of Sessions 1-2
- Sampling and Inferential Statistics
- Confidence interval for the mean
- Binomial Distribution and confidence intervals for proportions
- Bootstrap estimation using **infer**

The idea behind bootstrapping

Suppose we want to estimate the mean IMDB rating difference using our dataset. We can estimate one sample mean, but this does not capture how precise this estimate is if we want to make a claim about the population mean

The key idea of the bootstrap is to treat our sample as an approximate representation of the population, and to generate an approximate sampling distribution by sampling (with replacement) from our sample. The shape of the bootstrap distribution indicates how precise our estimate is.

1. Resampling does not provide a better estimate.
Resampling is only used to estimate the sample-to-sample variability in our estimate, not in an attempt to improve the estimate itself. If we attempted to improve our estimate using our bootstrap samples, we would just make things worse by producing an estimate of our estimate and essentially doubling any bias in the estimation.
2. Resampling works better with large samples than with small samples.
Small samples are unlikely to represent the population well. While resampling can provide methods that work as well as the traditional methods in standard situations and which can be applied in a wider range of situations without degraded performance, they do not fundamentally alter the need to have a sufficient sample size.

Bootstrap simulation

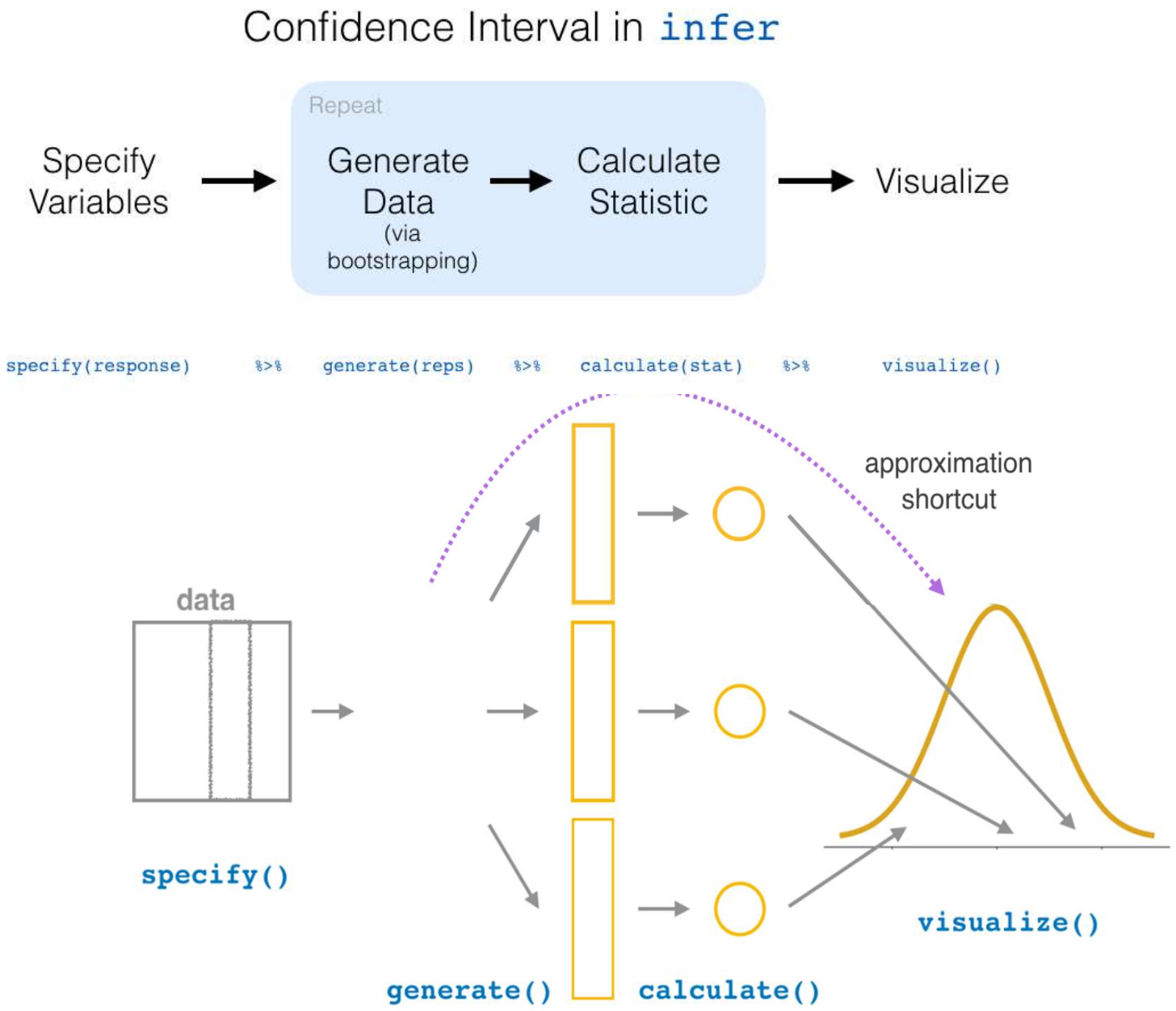
With infinite resources, we could take thousands of simultaneous samples (or even conduct a census) and get exact population parameter and know its exact value and variability.

1. **Take a bootstrap sample** — Sample with replacement; same size as original
2. **Calculate a bootstrap statistic** — Mean, median, proportion, difference, etc.

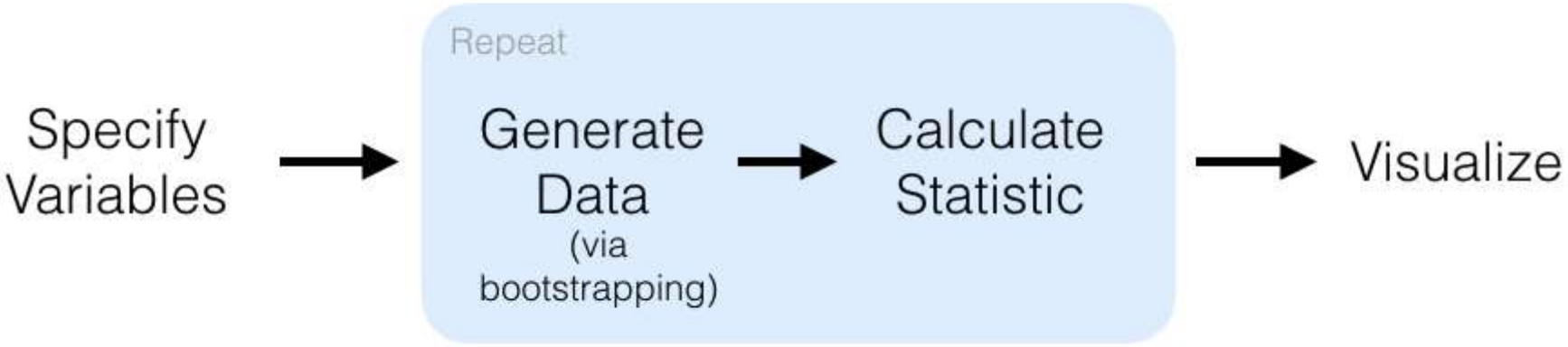
Repeat steps 1 -2 many times to get a distribution of sample statistics

Calculate the bounds of an X% confidence interval as the middle X% of the bootstrap distribution

Bootstrapping with **infer**



Confidence Interval in `infer`




```
specify(response)  %>%  generate(reps)  %>%  calculate(stat)  %>%  visualize()
```

With **infer** we can construct confidence intervals not just for the mean, but for any sample statistic we wanted, e.g. the median.

Random number seeds

A random number seed ensures your (pseudo)random numbers are the same every time. It's a good practice to set a seed, as this helps with reproducibility.

```
set.seed(1234)
```

```
sample(1:100, 5)
```

```
> set.seed(1234)
> sample(1:100,5)
[1] 28 80 22  9  5
> sample(1:100,5)
[1] 38 16  4 86 90
> sample(1:100,5)
[1] 70 79 78 14 56
>
> set.seed(1234)
> sample(1:100,5)
[1] 28 80 22  9  5
```

Bootstrapping with the **infer** package

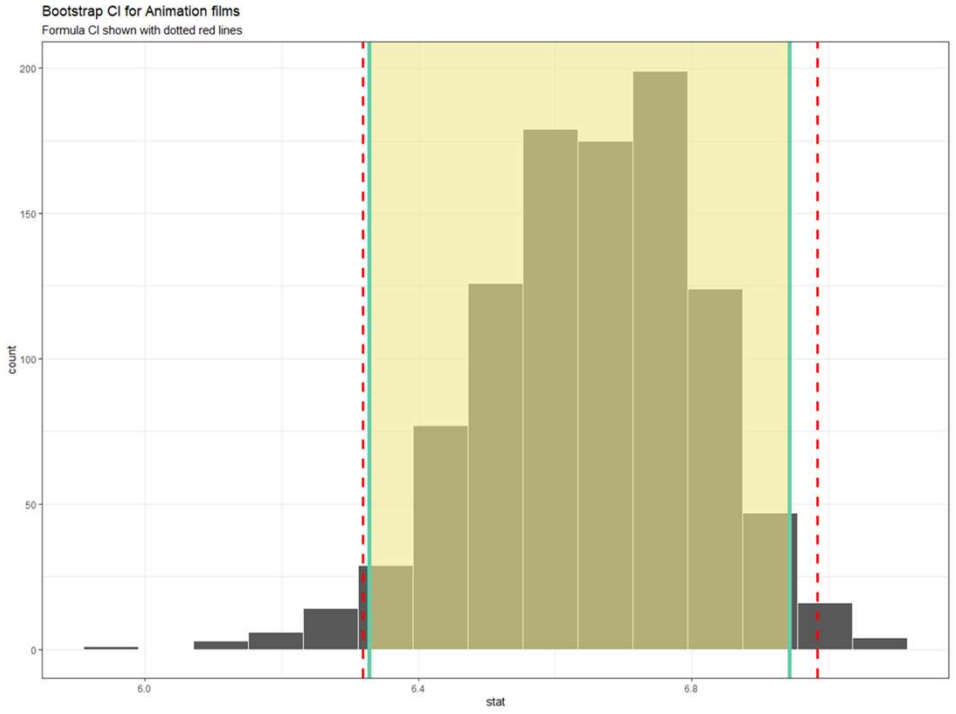
```
set.seed(1234)

boot_ratings <- movies %>%
  # Select Animation films
  filter(genre == "Animation") %>%

  # Specify the variable of interest
  specify(response = rating) %>%

  # Generate a bunch of bootstrap samples
  generate(reps = 1000, type = "bootstrap") %>%

  # Find the median of each sample
  calculate(stat = "mean")
```

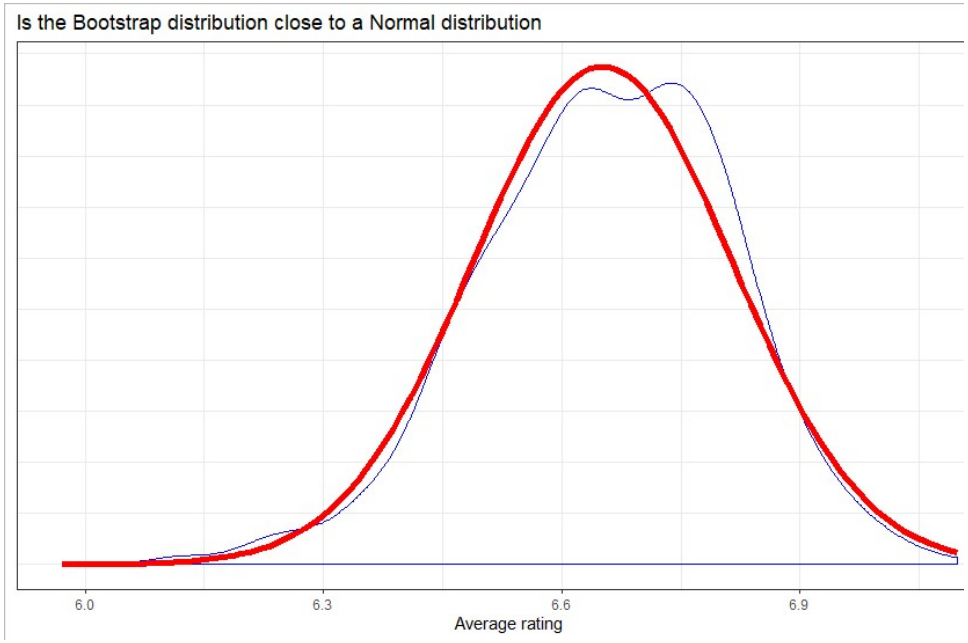
Bootstrapping with **infer**

95% CI is the middle 95% of bootstrap distribution

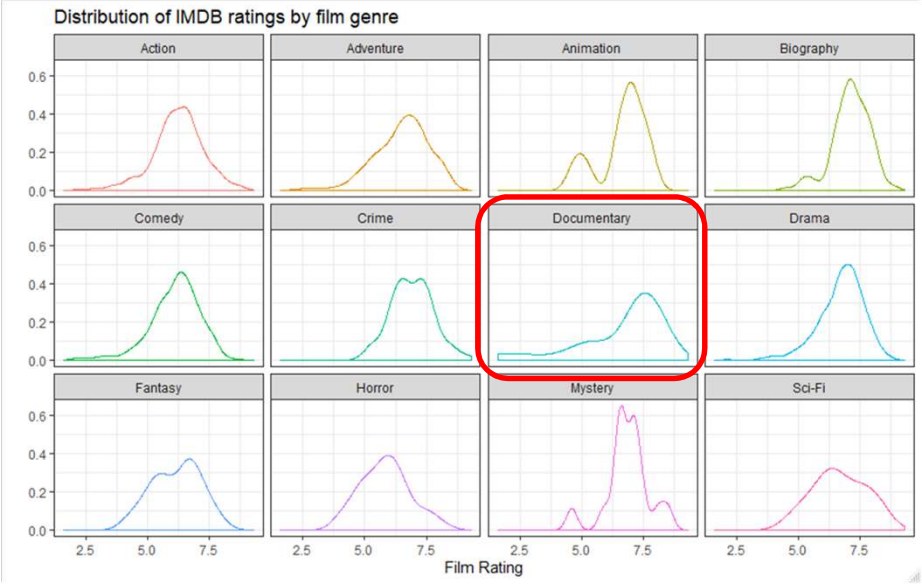
```
> percentile_ci <- boot_ratings %>%
    get_confidence_interval(level = 0.95, type = "percentile")
> percentile_ci
# A tibble: 1 x 2
  `2.5%` `97.5%`
   <dbl>   <dbl>
1   6.32    6.94
> formula_ci %>%
+   select(rating_low, rating_high)
# A tibble: 1 x 2
  rating_low rating_high
       <dbl>       <dbl>
1       6.32        6.98
```

95% CI using math formula





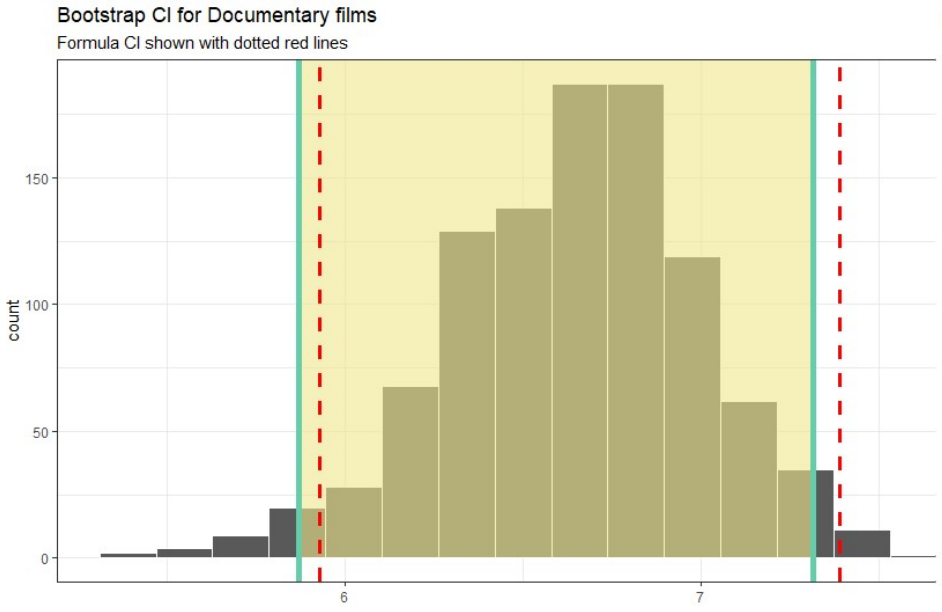
What about *genre==Documentary*

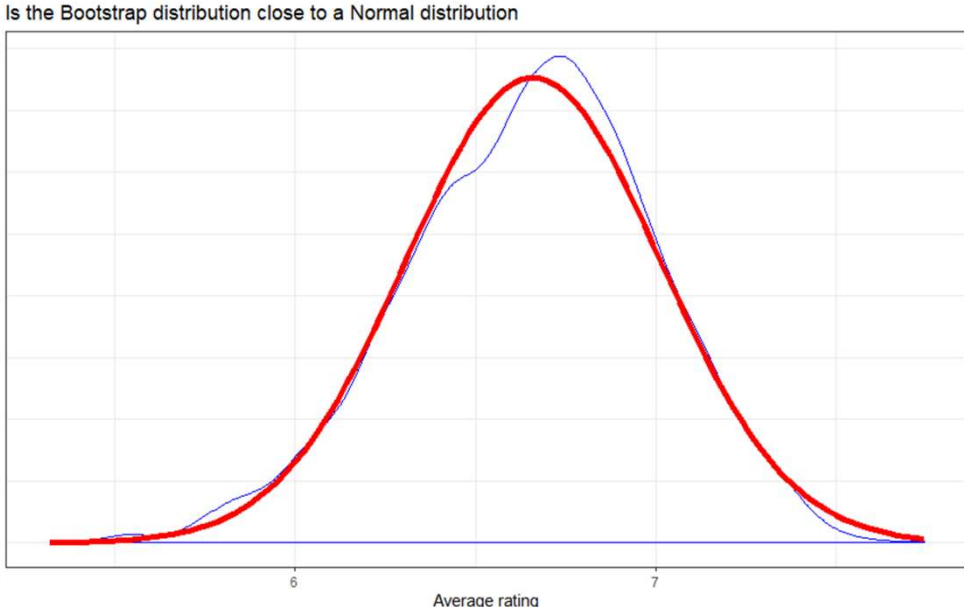


Distribution for ***genre==Documentary*** is heavily left skewed.

Will the bootstrap and formula CIs be similar?

```
> percentile_ci
# A tibble: 1 x 2
  `2.5%` `97.5%`
   <dbl>   <dbl>
1   5.87    7.32
> formula_ci %>%
    select(rating_low, rating_high)
# A tibble: 1 x 2
  rating_low rating_high
       <dbl>       <dbl>
1       5.93        7.39
```





Confidence Interval Interpretation (1/2)

```
> percentile_ci
# A tibble: 1 x 2
  `2.5%` `97.5%`
   <dbl>   <dbl>
1   6.32    6.94
```

The 95% confidence interval for the mean rating of margin of animation movies was calculated as (6.32, 6.94). Which of the following is the correct interpretation of this interval?

95% of the time the ratings for animation movies in this sample is between 6.32 and 6.94

95% of all ratings for the animation movies are between 6.32 and 6.94

We are 95% confident that the average rating for animation movies is between 6.32 and 6.94.

We are 95% confident that the average rating for animation movies in this sample between 6.32 and 6.94.

Confidence Interval Interpretation (1/2)

```
> percentile_ci
# A tibble: 1 x 2
  `2.5%` `97.5%`
   <dbl>   <dbl>
1   6.32    6.94
```

The 95% confidence interval for the mean rating of margin of animation movies was calculated as (6.32, 6.94). Which of the following is the correct interpretation of this interval?

95% of the time the ratings for animation movies in this sample is between 6.32 and 6.94

95% of all ratings for the animation movies are between 6.32 and 6.94

We are 95% confident that the average rating for animation movies is between 6.32 and 6.94.

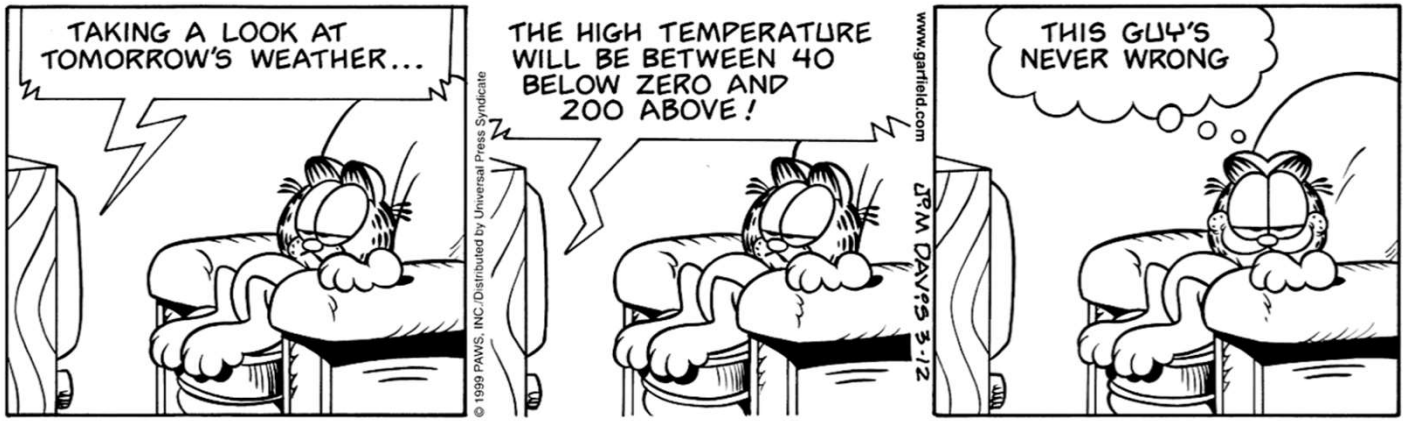
We are 95% confident that the average rating for animation movies in this sample between 6.32 and 6.94.

Common misconceptions about confidence intervals

1. *The confidence level of a confidence interval is the probability that the true population parameter is in the confidence interval you construct for a single sample.*
The confidence level is equal to the proportion of random samples that result in confidence intervals that contain the true population parameter.

2. *A narrower confidence interval is always better.*
This is incorrect since the width is a function of both the confidence level and the standard error.

3. *A wider interval means less confidence.*
This is incorrect since it is possible to make very precise statements with very little confidence.



CI Examples (1/2)

CNN conducts a poll among a random sample of 800 voters about whether they approve of the president's performance. CNN analysts create a 90% confidence interval for the true proportion of all voters in the US who approve of the president's performance.

If CNN conducts many identical polls on the same night, about 90% of the intervals produced will capture the true proportion of voters who approve of the president

About 90% of people who support the president will respond to the poll

If CNN repeats this poll 20 times on the same night and calculates 90% confidence intervals for each poll, we can expect that around 18 of those intervals will contain the true proportion of voters who approve of the president.

There's a 90% chance that the actual population proportion is in the confidence interval

CI Examples (1/2)

CNN conducts a poll among a random sample of 800 voters about whether they approve of the president's performance. CNN analysts create a 90% confidence interval for the true proportion of all voters in the US who approve of the president's performance.

If CNN conducts many identical polls on the same night, about 90% of the intervals produced will capture the true proportion of voters who approve of the president

About 90% of people who support the president will respond to the poll

If CNN repeats this poll 20 times on the same night and calculates 90% confidence intervals for each poll, we can expect that around 18 of those intervals will contain the true proportion of voters who approve of the president.

There's a 90% chance that the actual population proportion is in the confidence interval

CI Examples (2/2)

A city manager wants to know the true average property value of house prices in her city. She takes a random sample of 200 houses and builds a 95% confidence interval through bootstrapping. The interval is ($180,000, $300,000).

- If the city manager took another random sample of 200 houses, there's a 95% chance *that* sample mean would be between $180,000 and $300,000
- About 95% of houses in the sample are valued between $180,000 and $300,000
- We're 95% confident that the interval ($180,000, $300,000) captured the true mean value
- There's a 95% chance that the true mean is between $180,000 and $300,000

The city manager was to be more confident about home prices in her city, so she uses the same sample data and uses bootstrapping techniques to calculate a 99% confidence interval. What will happen to the interval when she changes the confidence level from 95% to 99%?

- It's impossible to say without seeing the sample data
- Increasing the confidence to 99% will increase the margin of error and result in a wider interval
- Increasing the confidence to 99% will decrease the margin of error and result in a narrower interval

CI Examples (2/2)

A city manager wants to know the true average property value of house prices in her city. She takes a random sample of 200 houses and builds a 95% confidence interval through bootstrapping. The interval is ($180,000, $300,000).

- If the city manager took another random sample of 200 houses, there's a 95% chance *that* sample mean would be between $180,000 and $300,000
- About 95% of houses in the sample are valued between $180,000 and $300,000
- ★ We're 95% confident that the interval ($180,000, $300,000) captured the true mean value
- ☆ There's a 95% chance that the true mean is between $180,000 and $300,000

The city manager was to be more confident about home prices in her city, so she uses the same sample data and uses bootstrapping techniques to calculate a 99% confidence interval. What will happen to the interval when she changes the confidence level from 95% to 99%?

- It's impossible to say without seeing the sample data
- ★ Increasing the confidence to 99% will increase the margin of error and result in a wider interval
- Increasing the confidence to 99% will decrease the margin of error and result in a narrower interval

Summary: bootstrap and CIs

Sample statistic ≠ population parameter

But if the sample is good, it can be a good estimate

Report estimate with confidence interval

Width of interval depends on how variable sample statistics would be from different samples

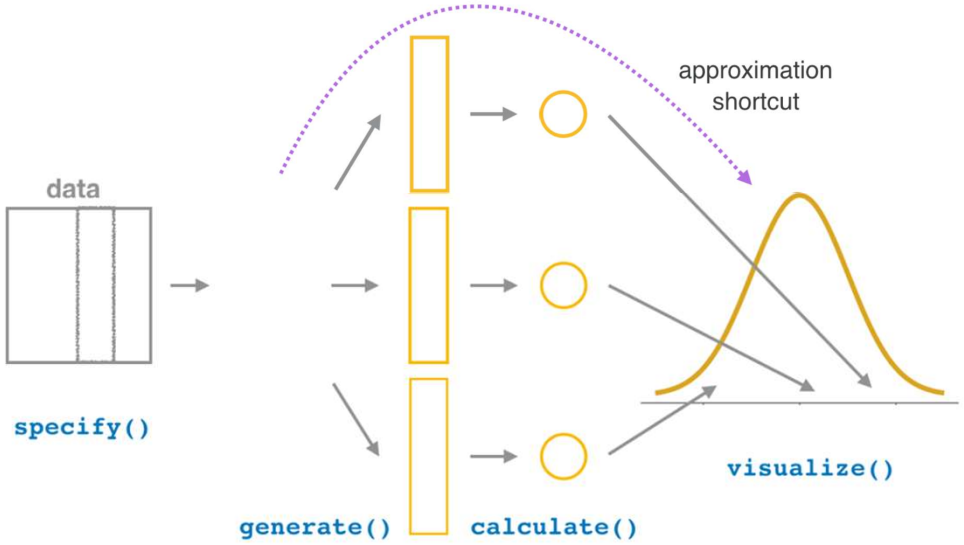
We can't keep sampling from the population, so bootstrap

This allows us to measure the variability

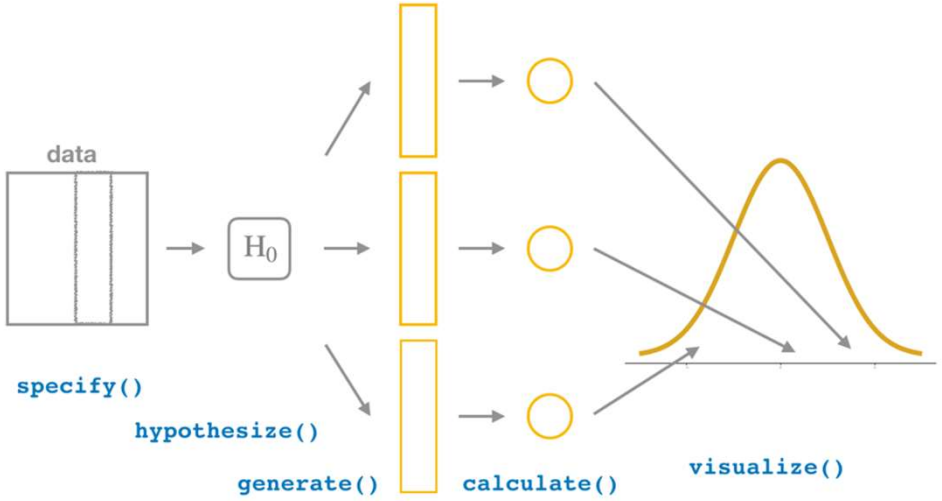
Summary: bootstrap and CIs

bootstrap

1. `specify()` the variables of interest in your data frame
2. `generate()` replicates of bootstrap resamples with replacement
3. `calculate()` the summary statistic of interest
4. `visualize()` the resulting bootstrap distribution and the confidence interval.



Hypothesis testing



Appendix

lubridate - Dealing with dates in tidyverse

Date-Time format in R Uses ISO 8601 Universal format: 2021-12-25
ordered from the largest to smallest unit of time:
year, month (or week), day, hour, minute, second, and fraction of second

lubridate() handles dates; Part of the tidyverse, but must be explicitly loaded
ymd() – stands for year-month-day

```
> date <- ymd("2021-12-25")
> today <- Sys.Date() #reads system's current date
>
> year(date)
[1] 2021
> month(date) # Equal to 12
[1] 12
> month(date,label = TRUE) # Equal to "Dec"
[1] Dec
Levels: Jan < Feb < Mar < Apr < May < Jun < Jul < Aug < Sep < Oct < Nov < Dec
>
> day(date)
[1] 25
> wday(date,label = TRUE) # Equal to "Wed"
[1] Sat
Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
>
> isoweek(date)
[1] 51
>
> date - today
Time difference of 111 days
```

Dates and times with lubridate : : **CHEAT SHEET**

Date-times

`2017-11-28 12:00:00`

`2017-11-28 12:00:00`
A **date-time** is a point on the timeline, stored as the number of seconds since 1970-01-01 00:00:00 UTC

dt <- ***as_datetime****(1511870400)*
"2017-11-28 12:00:00 UTC"

`2017-11-28`
A **date** is a day stored as the number of days since 1970-01-01

d <- ***as_date****(17498)*
"2017-11-28"

`12:00:00`
An hms is a **time** stored as the number of seconds since 00:00:00

*t <- hms::****as.hms****(85)*
00:01:25

PARSE DATE-TIMES (Convert strings or numbers to date-times)

1. Identify the order of the year (**y**), month (**m**), day (**d**), hour (**h**), minute (**m**) and second (**s**) elements in your data
2. Use the function below whose name replicates the order. Each accepts a wide variety of input formats.

`2017-11-28T14:02:00` **ymd_hms**(), **ymd_hm**(), **ymd_h**(). *ymd_hms("2017-11-28T14:02:00")*

`2017-22-12 10:00:00` **ydm_hms**(), **ydm_hm**(), **ydm_h**(). *ydm_hms("2017-22-12 10:00:00")*

`11/28/2017 1:02:03` **mdy_hms**(), **mdy_hm**(), **mdy_h**(). *mdy_hms("11/28/2017 1:02:03")*

`1 Jan 2017 23:59:59` **dmy_hms**(), **dmy_hm**(), **dmy_h**(). *dmy_hms("1 Jan 2017 23:59:59")*

`20170131` **ymd**(), **ydm**(). *ymd(20170131)*

`July 4th, 2000` **mdy**(), **myd**(). *mdy("July 4th, 2000")*

`4th of July '99` **dmy**(), **dym**(). *dmy("4th of July '99")*

`2001: Q3` **yq**() Q for quarter. *yq("2001: Q3")*

`2:01` hms::**hms**() Also lubridate::**hms**(), **hm**() and **ms**(), which return periods.* *hms::hms(sec = 0, min= 1, hours = 2)*

`2017.5` **date_decimal**(decimal, tz = "UTC") Q for quarter. *date_decimal(2017.5)*

now(tzone = "") Current time in tz (defaults to system tz). *now()*

today(tzone = "") Current date in a tz (defaults to system tz). *today()*

fast_strptime() Faster strptime. *fast_strptime('9/1/01', '%y/%m/%d')*

parse_date_time() Easier strptime. *parse_date_time("9/1/01", "ymd")*

GET AND SET COMPONENTS

Use an accessor function to get a component. *d ## "2017-11-28"* *day(d) ## 28*

Assign into an accessor function to change a component in place. *day(d) <- 1* *d ## "2017-11-01"*

`2018-01-31 11:59:59` **date**(x) Date component. *date(dt)*

`2018-01-31 11:59:59` **year**(x) Year. *year(dt)*
isoyear(x) The ISO 8601 year.
epiyear(x) Epidemiological year.

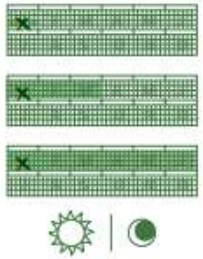
`2018-01-31 11:59:59` **month**(x, label, abbr) Month. *month(dt)*

`2018-01-31 11:59:59` **day**(x) Day of month. *day(dt)*
wday(x,label,abbr) Day of week.
qday(x) Day of quarter.

`2018-01-31 11:59:59` **hour**(x) Hour. *hour(dt)*

`2018-01-31 11:59:59` **minute**(x) Minutes. *minute(dt)*

`2018-01-31 11:59:59` **second**(x) Seconds. *second(dt)*



week(x) Week of the year. *week(dt)*
isoweek() ISO 8601 week.
epiweek() Epidemiological week.

quarter(x, with_year = FALSE) Quarter. *quarter(dt)*

semester(x, with_year = FALSE) Semester. *semester(dt)*

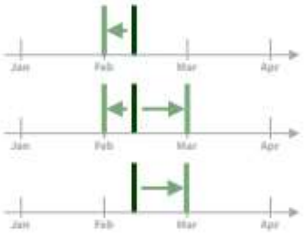
am(x) Is it in the am? *am(dt)*
pm(x) Is it in the pm? *pm(dt)*

dst(x) Is it daylight savings? *dst(d)*

leap_year(x) Is it a leap year? *leap_year(d)*

update(object, ..., simple = FALSE) *update(dt, mday = 2, hour = 1)*

Round Date-times



floor_date(x, unit = "second") Round down to nearest unit. *floor_date(dt, unit = "month")*

round_date(x, unit = "second") Round to nearest unit. *round_date(dt, unit = "month")*

ceiling_date(x, unit = "second", change_on_boundary = NULL) Round up to nearest unit. *ceiling_date(dt, unit = "month")*

rollback(dates, roll_to_first = FALSE, preserve_hms = TRUE) Roll back to last day of previous month. *rollback(dt)*

Stamp Date-times

stamp() Derive a template from an example string and return a new function that will apply the template to date-times. Also **stamp_date**() and **stamp_time**().

1. Derive a template, create a function
   sf <- stamp("Created Sunday, Jan 17, 1999 3:34")
2. Apply the template to dates
   sf(ymd("2010-04-05"))
   ## [1] "Created Monday, Apr 05, 2010 00:00"

Tip: use a date with day > 12

Time Zones

R recognizes ~600 time zones. Each encodes the time zone, Daylight Savings Time, and historical calendar variations for an area. R assigns *one* time zone per vector.

Use the **UTC** time zone to avoid Daylight Savings.

OlsonNames() Returns a list of valid time zone names. *OlsonNames()*

with_tz(time, tzone = "") Get the **same date-time** in a new time zone (a new clock time). *with_tz(dt, "US/Pacific")*

force_tz(time, tzone = "") Get the **same clock time** in a new time zone (a new date-time). *force_tz(dt, "US/Pacific")*

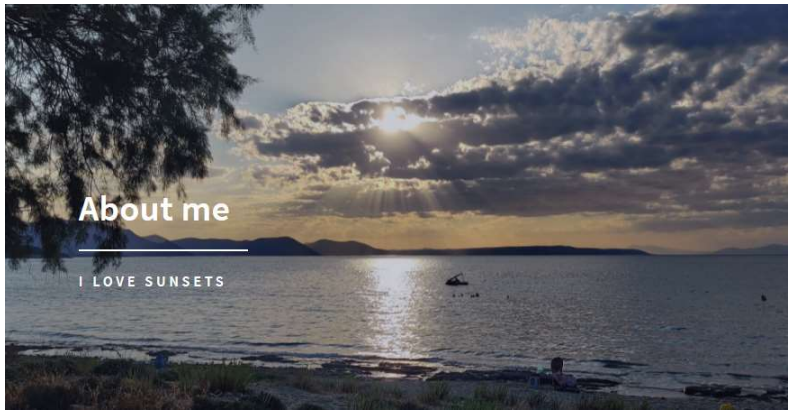
Portfolio website – change tile image, title, subtitle

config.yaml

\themes\forty\static\img

```
tiles:
  enable: yes
  showcase:
    - image: pic01.jpg
      subtitle: Ipsum Dolor Sit Amet
      title: Aliquam
      url: blogs/aliquam
    - image: pic02.jpg
      subtitle: Feugiat Amet Tempus
      title: Tempus
      url: blogs/tempus
```


```
  showcase:
    - image: backround_sunset.jpg
      subtitle: I love sunsets
      title: About me
      url: blogs/aliquam
    - image: pic02.jpg
      subtitle: Feugiat Amet Tempus
      title: Tempus
      url: blogs/tempus
```

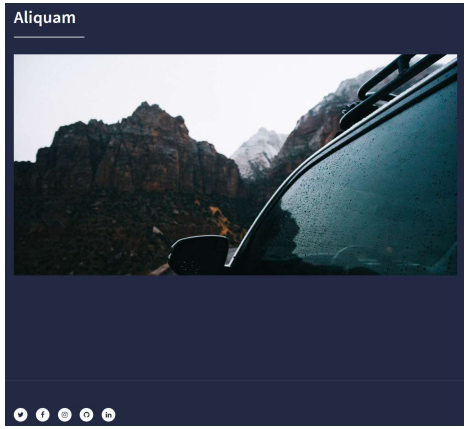


Change image inside blog
slug = blog address

config.yaml

\static\img\blogs

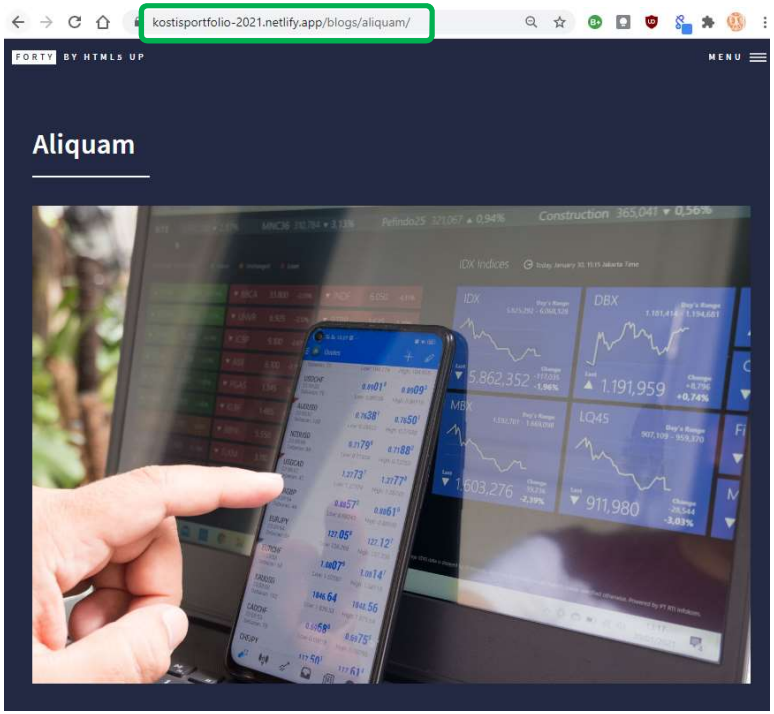
```yaml
  tiles:
    enable: yes
    showcase:
      - image: pic01.jpg
        subtitle: Ipsum Dolor Sit Amet
        title: Aliquam
        url: blogs/aliquam
      - image: pic02.jpg
        subtitle: Feugiat Amet Tempus
        title: Tempus
        url: blogs/tempus
```



\content\blogs\blog4.md

```yaml
---
categories:
- ""
- ""
date: "2017-10-31T22:42:51-05:00"
description: Nullam et orci eu lorem consequat tincidunt vivamus et
  sed nunc rhoncus condimentum sem. In efficitur ligula tate urna.
  sed magna lacinia magna pellentesque lorem ipsum dolor. Nullam et
  consequat tincidunt. Vivamus et sagittis tempus.
draft: false
image: forex_prices.jpg
keywords: ""
slug: aliquam
title: Aliquam
---

```

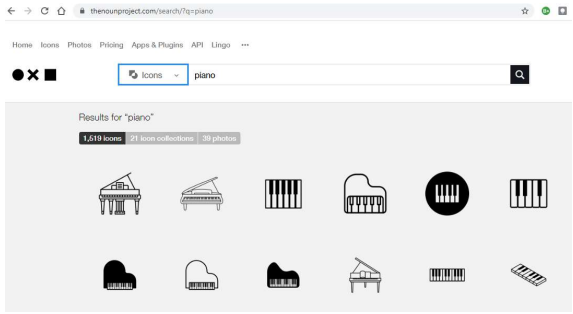


Where to find royalty-free photos-icons?

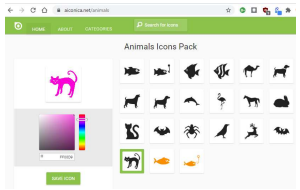
- Use the Creative Commons filters on *Google Images* or *Flickr*
- Unsplash https://unsplash.com/
- freephotos.cc https://freephotos.cc/en
- Pexels https://www.pexels.com/
- Pixabay https://pixabay.com/
- StockSnap.io https://stocksnap.io/
- Burst https://burst.shopify.com/

Icons and Vectors

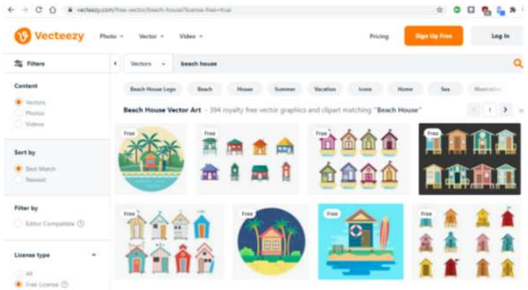
- Noun Project https://thenounproject.com/



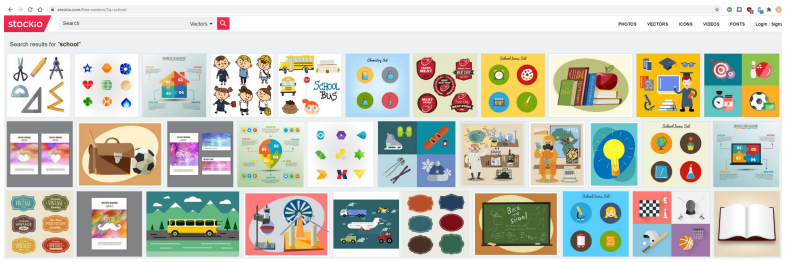
- Aiconica https://aiconica.net/



- Vecteezy https://www.vecteezy.com/



- Stockio https://www.stockio.com/



Take your pre-programme Rmd save it in \content\blogs\

\content\blogs\blog4.md

```
---
categories:
- ""
- ""
date: "2017-10-31T22:42:51-05:00"
description: Nullam et orci eu lorem consequat tincidunt vivamus et
  sed nunc rhoncus condimentum sem. In efficitur ligula tate urna.
  sed magna lacinia magna pellentesque lorem ipsum dolor. Nullam et
  consequat tincidunt. Vivamus et sagittis tempus.
draft: false
image: forex_prices.jpg
keywords: ""
slug: aliquam
title: Aliquam
---

```

Change the YAML in your Rmd to be like blog4.md

```
---
categories:
- ""
- ""
date: "2017-10-31T22:42:51-05:00"
description: Nullam et orci eu lorem consequat tincidunt vivamus et sagittis magna
  sed nunc rhoncus condimentum sem. In efficitur ligula tate urna. Maecenas massa
  sed magna lacinia magna pellentesque lorem ipsum dolor. Nullam et orci eu lorem
  consequat tincidunt. Vivamus et sagittis tempus.
draft: false
image: forex_prices.jpg
keywords: ""
slug: aliquam
title: Aliquam
---


```{r load-libraries, warning=FALSE, message=FALSE, echo=FALSE}
library(tidyverse)  # Load ggplot2, dplyr, and all the other tidyverse packages
library(gapminder)  # gapminder dataset
library(here)
library(janitor)
```

The goal is to test your software installation, to demonstrate competency in Markdown,
```

```
---
title: 'Pre-programme Assignment'
author: "Kostis"
date: "What date?"
output:
  html_document:
    theme: flatly
    highlight: zenburn
    toc: yes
    toc_float: yes
---

```{r load-libraries, warning=FALSE, message=FALSE, echo=FALSE}
library(tidyverse)  # Load ggplot2, dplyr, and all the other tidyverse packages
library(gapminder)  # gapminder dataset
library(here)
library(janitor)
```
```

blogdown::serve_site() will knit the Rmd

```
Quitting from lines 144-148 (kostis_pre_programme.Rmd)
Error: 'C:/Users/kchristodoulou/Desktop/my_gorgeous_website/data/brexit_results.csv' does not exist.
Execution halted
Error: Failed to render content/blogs/kostis_pre_programme.Rmd
```

1. Create a folder \data\ and save brexit_results.csv
2. Don't knit. You may have to restart R (Cmd + Shift + F10)
3. Delete `blog4.md`, because it has the same slug (shortcut address)
4. Run `blogdown::serve_site()`

5. Once it has rendered, you need to
   - `git add -A`
   - `git commit -m "a useful message"`
   - `git pull`
   - `git push`