DATA2002

Bootstrap

Garth Tarr

THE UNIVERSITY OF SYDNEY

Speed of light

Confidence intervals

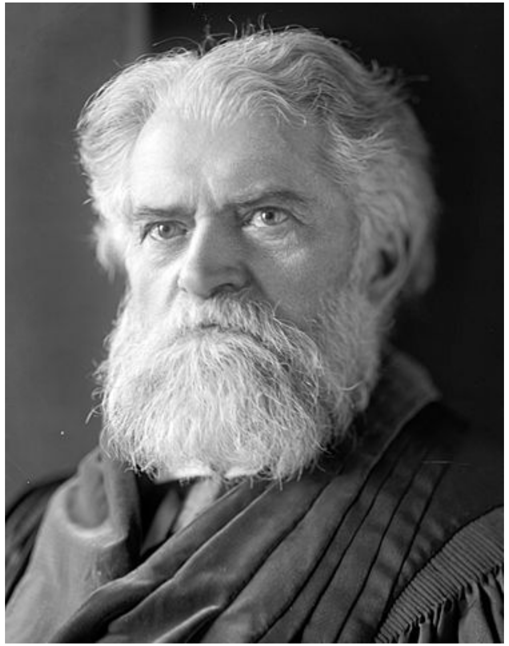
Bootstrap

Flight delays

Speed of light

Speed of light

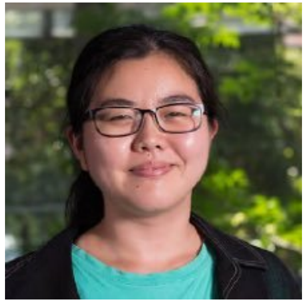
- Simon Newcomb measured the time required for light to travel from his laboratory on the Potomac River to a mirror at the base of the Washington Monument and back, a total distance of about 7400 meters.
- He performed this experiment 66 times.
- These measurements were used to estimate the speed of light.



Simon Newcomb

Emi Tanaka

- Was a lecturer here at the University of Sydney, now at Monash.
- Statistics for food security - designing and analysing experiments looking at improving the genetics of crops.
- ▶ Sydney Data Stories - Masterclass Series - Dr Emi Tanaka
- Creates fun R packages and makes super nice presentations.
- 🐦 @statsgen
- Website: https://emitanaka.github.io/
- Her data science showcase: https://emitanaka.github.io/showcase/ (Absolutely worth a look!)



Emi Tanaka

About 40,000 grain research plots at Narrabri.

Speed of light

Newcomb's measurements of the passage time of light, made July 24 1882 to September 5 1882. The values $\times 10^{-3} + 24.8$ are Newcomb's measurements recorded in millionths of a second for observations of light passing over a distance of 3721 m and back, from Fort Myer on the west bank of the Potomac to a fixed mirror at the base of the Washington monument. The "true" value is 33.02. (Stigler, 1977; Table 5)

```r
library(readr)
speed_file = read_csv("https://raw.githubusercontent.com/DATA2002/data/master/speed_of_light.txt")
speed = speed_file$Speed_of_Light
```

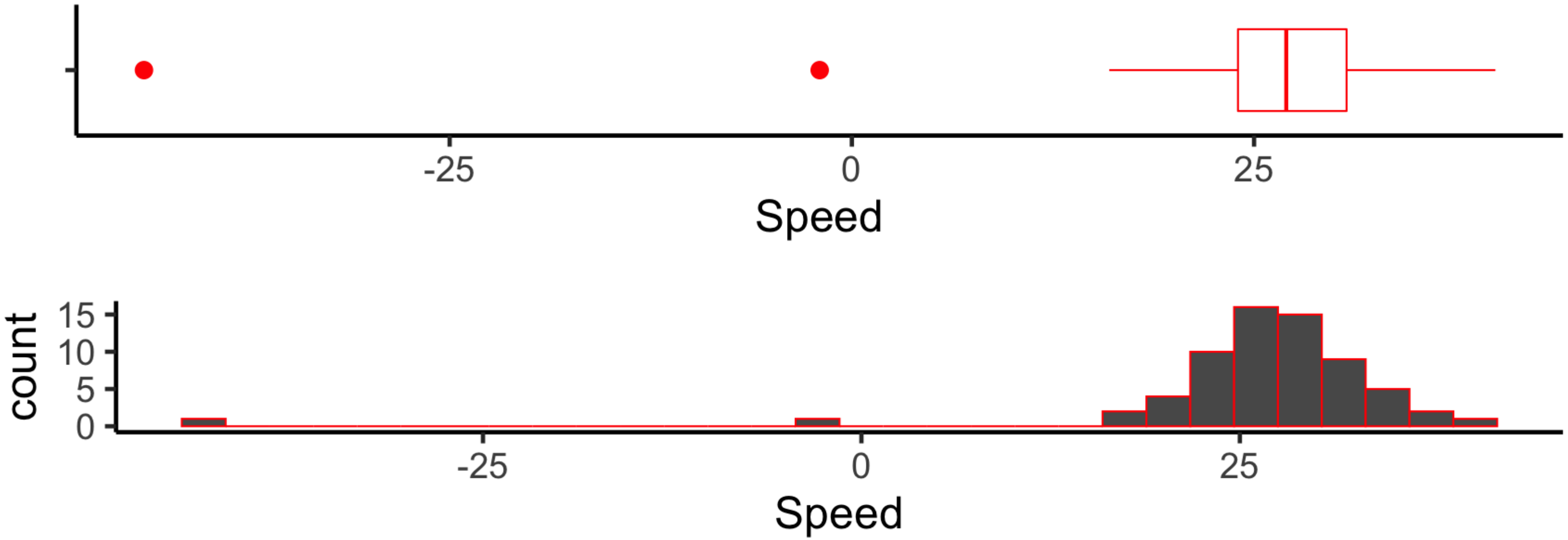
```r
mean(speed)
```

```
## [1] 26.21212
```

```r
median(speed)
```

```
## [1] 27
```

```r
library(ggplot2)
ggplot(speed_file, aes(x="", y = Speed_of_Light)) +
  geom_boxplot(colour = "red", outlier.size = 4) +
  theme_classic(base_size = 24) +
  labs(x = "", y = "Speed") + coord_flip()
ggplot(speed_file, aes(x = Speed_of_Light)) +
  geom_histogram(colour = "red") +
  theme_classic(base_size = 24) +
  labs(x = "Speed")
```



Confidence intervals

Estimation vs hypothesis testing

Estimation

- A population parameter is unknown.
- Use the sample statistics to generate estimates of the population parameter.

Hypothesis testing

- Explicit statement (or hypothesis) regarding the population parameter.
- Test statistics are generated which will either support or reject the null hypothesis.

Confidence intervals

- We should avoid reporting just a point estimate for a sample.
- We should always include a measure of variability:

$$\hat{\theta} \pm \text{margin of error}$$

where $\hat{\theta}$ is the point estimate (e.g. sample mean, $\bar{X}$).

The margin of error usually takes the form

$$\text{margin of error} = \text{critical value} \times \text{SE}(\hat{\theta})$$

where the critical value is some quantile from an appropriate distribution (e.g. $z_{\alpha/2}$ or $t_\nu(\alpha/2)$) and $\text{SE}(\theta)$ is the standard error of the point estimate (e.g. $\text{SE}(\bar{X}) = \sigma/\sqrt{n}$).

Confidence intervals

- The *point* estimate $\hat{\theta}$ (say $\bar{x}$) of a parameter θ (say μ) does not show its variability across samples.
- To show such estimation precision, we should find an *interval* estimate.

Definition: Let $\hat{\theta}_L$ and $\hat{\theta}_R$ be two statistics. If

$$P(\hat{\theta}_L \leq \theta \leq \hat{\theta}_R) = 1 - \alpha,$$

then the random interval $[\hat{\theta}_L, \hat{\theta}_R]$ is called a $100(1-\alpha)\%$ *confidence interval* (CI) for θ, and $100(1-\alpha)\%$ is called the *confidence level* of the interval.

- In general, the α may be chosen to be 0.01, 0.05, 0.10, etc, and then we get 99%, 95%, 90% confidence interval accordingly.

Confidence intervals for the mean

Let $X_1, X_2, \ldots, X_n$ be a random sample from normal population and $X_i \sim \mathcal{N}(\mu, \sigma^2)$, where σ^2 is unknown.

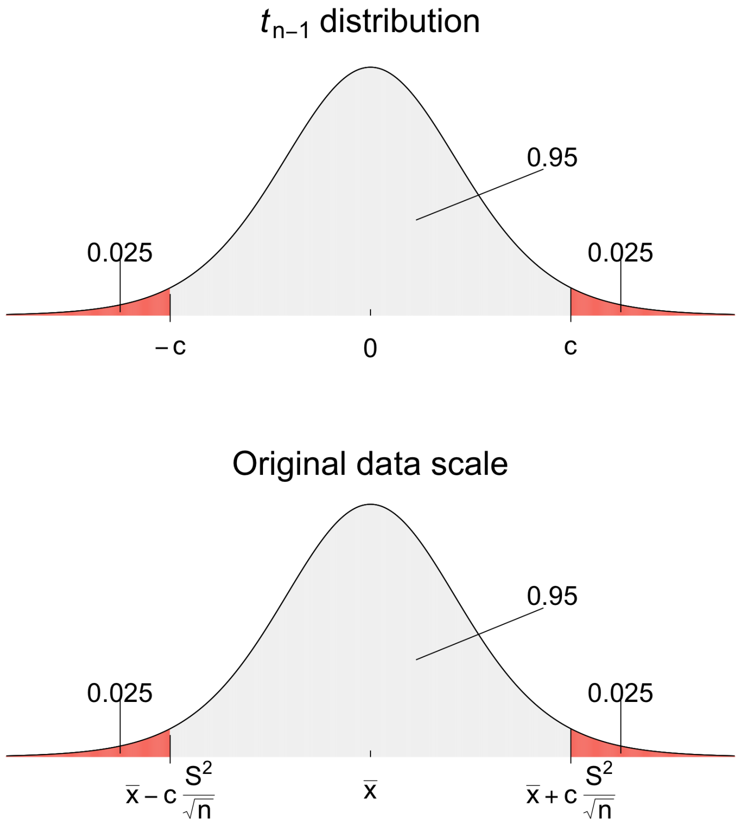
Then

$$\frac{\bar{X} - \mu}{S/\sqrt{n}} \sim t_{n-1}$$

and

$$P\left(-c < \frac{\bar{X} - \mu}{S/\sqrt{n}} < c\right) = 1 - \alpha$$
$$P\left(-cS/\sqrt{n} < \mu - \bar{X} < cS/\sqrt{n}\right) = 1 - \alpha$$
$$P\left(\bar{X} - cS/\sqrt{n} < \mu < \bar{X} + cS/\sqrt{n}\right) = 1 - \alpha$$

In the plots on the right, $\alpha = 0.05$.



Meaning of the confidence interval

Suppose a 95% confidence interval for the mean μ is (a, b).

- This does **not** mean that 95% of the means μ are in (a, b), that is $P(a < \mu < b) = 0.95$ since μ is a **fixed** but unknown parameter.
- It also does **not** mean $P(a < \bar{X} < b) = 0.95$, where $\bar{X}$ is the sample mean since the CI is for the true mean μ not the sample mean $\bar{X}$.

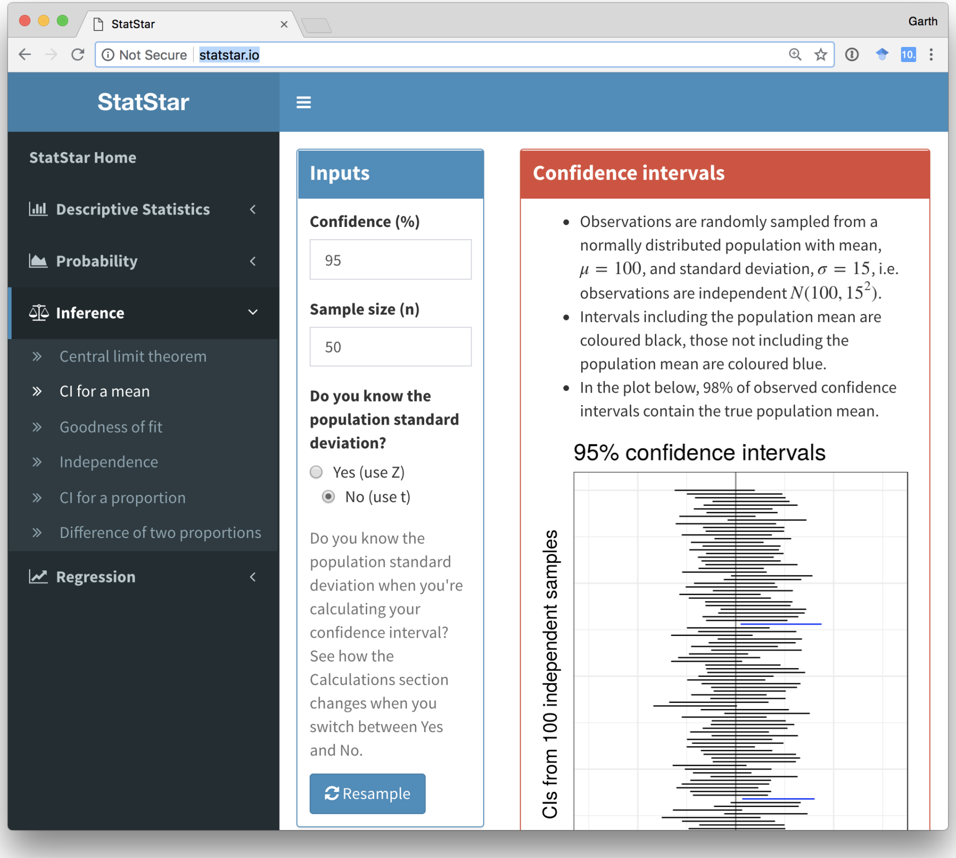
It **does** mean that if we draw a large number of random samples and compute for each sample a 95% CI, about 95% of these CIs will contain μ.

It **can** be described as a **range of plausible values** for the population parameter.

Try for yourself

http://statstar.io

Inference > CI for a mean



Inputs

Confidence (%): 95

Sample size (n): 50

Do you know the population standard deviation?

○ Yes (use Z)

● No (use t)

Do you know the population standard deviation when you're calculating your confidence interval? See how the Calculations section changes when you switch between Yes and No.

Confidence intervals

- Observations are randomly sampled from a normally distributed population with mean, $\mu = 100$, and standard deviation, $\sigma = 15$, i.e. observations are independent $N(100, 15^2)$.
- Intervals including the population mean are coloured black, those not including the population mean are coloured blue.
- In the plot below, 98% of observed confidence intervals contain the true population mean.

Confidence intervals

A confidence interval is a statement about the underlying population parameter.

- Formally, for a 95% confidence interval, 95% of all possible random samples will contain the population mean, leading to 95% **confidence** that a single interval contains the true mean.
- In reality, a specific interval either contains the mean or it doesn't. We just do not know which one is true; but we have 95% **confidence** that it does.

In this context **confidence** isn't the same as **probability**.

Distributional assumptions

> 💬
>
> What happens if your data does not follow a normal distribution?

1. Guess the distribution of the data and use this distribution to calculate critical values for confidence levels. **Risky.**
2. Use **bootstrap resampling** to empirically model the distribution of the data.

Bootstrapping

Bootstrap resampling

Bootstrapping is a computational process that allows us to as make inferences about the population where no information is available about the population.

Bootstrap methods take their name from the idea of "lifting yourself up by your bootstraps" - moving up without any additional outside help. The name was introduced by Efron (1979).

> "in the absence of any other knowledge about a population, the distribution of values found in a sample of size n from the population is the best guide to the distribution in the population. Therefore, to approximate what would happen if the population was resampled it is sensible to resample the sample."
>
> Manly (2007, p. 41)

The classic approach to bootstrapping is to **repeatedly resample** from the sample (with replacement).

Speed of light

- Simon Newcomb measured the time required for light to travel from his laboratory on the Potomac River to a mirror at the base of the Washington Monument and back, a total distance of about 7400 meters.
- He performed this experiment 66 times (66 observations).
- These measurements were used to estimate the speed of light.
- What if we approximated **sampling from the population** by **sampling from this sample**?

Bootstrapping speed of light measurements

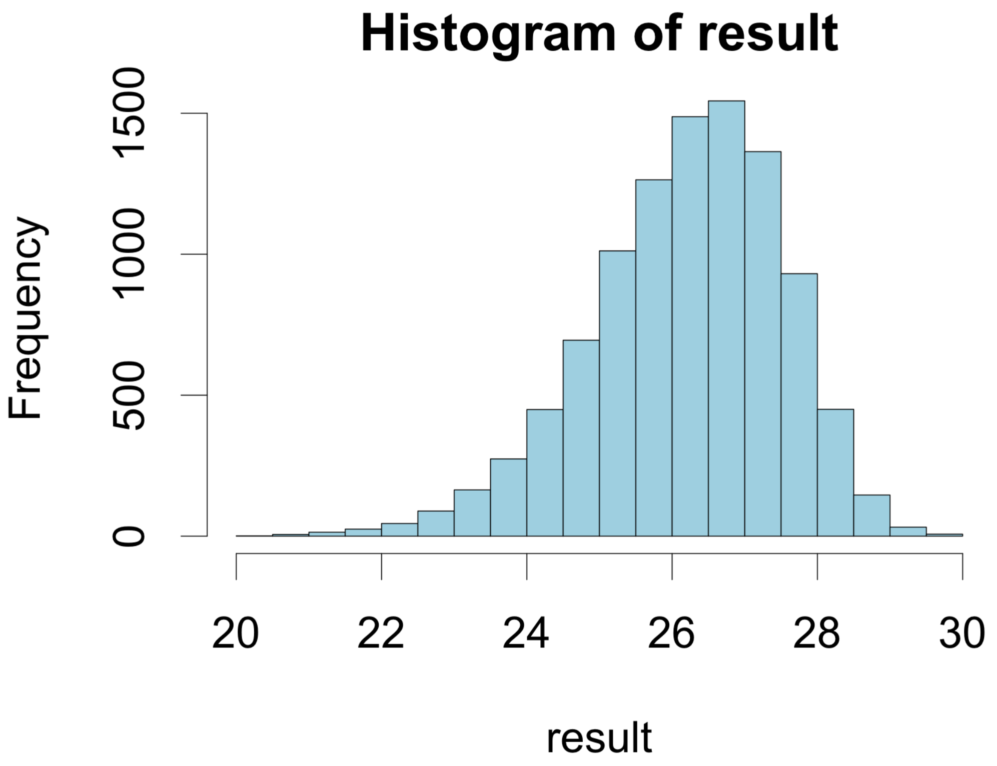
```
mean(speed)
```

```
## [1] 26.21212
```

```
set.seed(123)
B = 10000
result = vector("numeric", length = B)
for(i in 1:B){
  newData = sample(speed, replace = TRUE)
  result[i] = mean(newData)
}
round(head(result), 2)
```

```
## [1] 24.17 26.92 25.38 25.41 24.85 26.24
```

```
hist(result, col = "lightblue")
```



Bootstrap confidence intervals

Efron (1979) proposed that the bootstrap confidence interval be the quantiles from the bootstrap distribution.

In general, (θ^*_L, θ^*_U) are the bounds of the $100(1{-}\alpha)$ bootstrap CI where θ^*_L is the $\alpha/2$ quantile from the bootstrap distribution and θ^*_U is the $1-\alpha/2$ quantile from the bootstrap distribution.

If `result` has our bootstrap estimates then we can get a 95% confidence interval using:

```
quantile(result, c(0.025, 0.975))
```

There are other ways of constructing bootstrap, e.g. the centred percentile method of Hall (1989) which can be used when the bootstrap distribution isn't symmetric.

```r
CI = quantile(result, c(0.025, 0.975))
CI
```

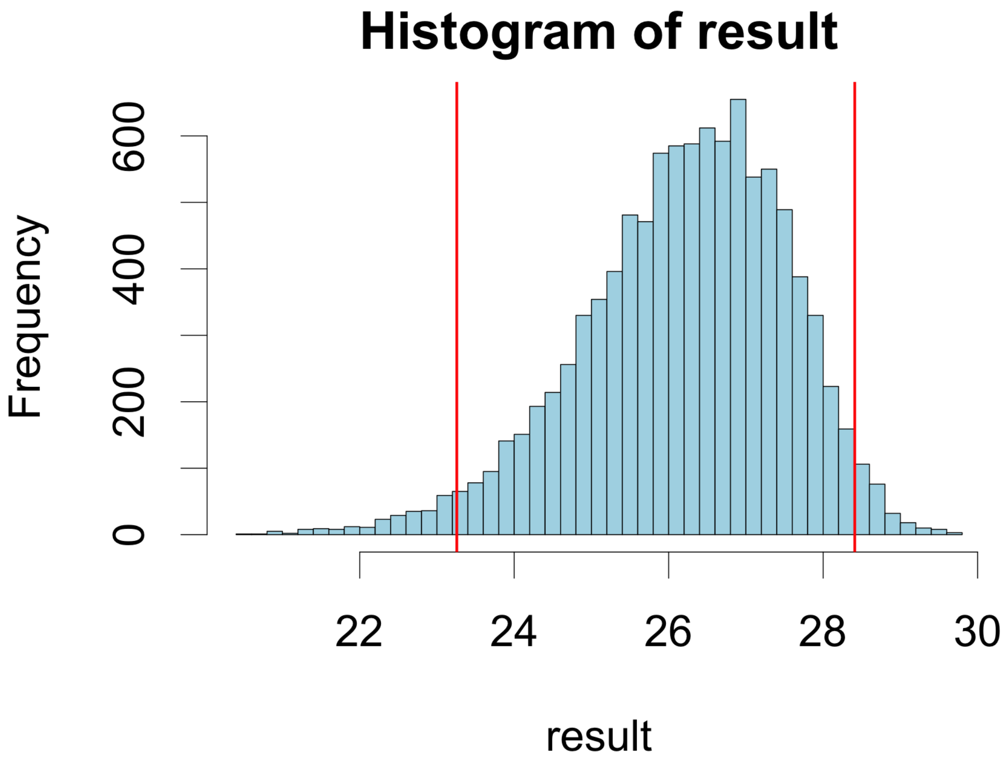
```
##     2.5%    97.5%
## 23.25720 28.40909
```

```r
CI - mean(speed)
```

```
##      2.5%     97.5%
## -2.954924  2.196970
```

The bootstrap confidence interval is not symmetric about the mean!

```r
hist(result, breaks = 50,
     col = "lightblue")
abline(v = CI, col = "red", lwd = 3)
```



Compare with the confidence interval using the t distribution.

```
xbar = mean(speed)
n = length(speed)
se = sd(speed)/sqrt(n)
c(xbar, n, se)
```

```
## [1] 26.212121 66.000000  1.322658
```

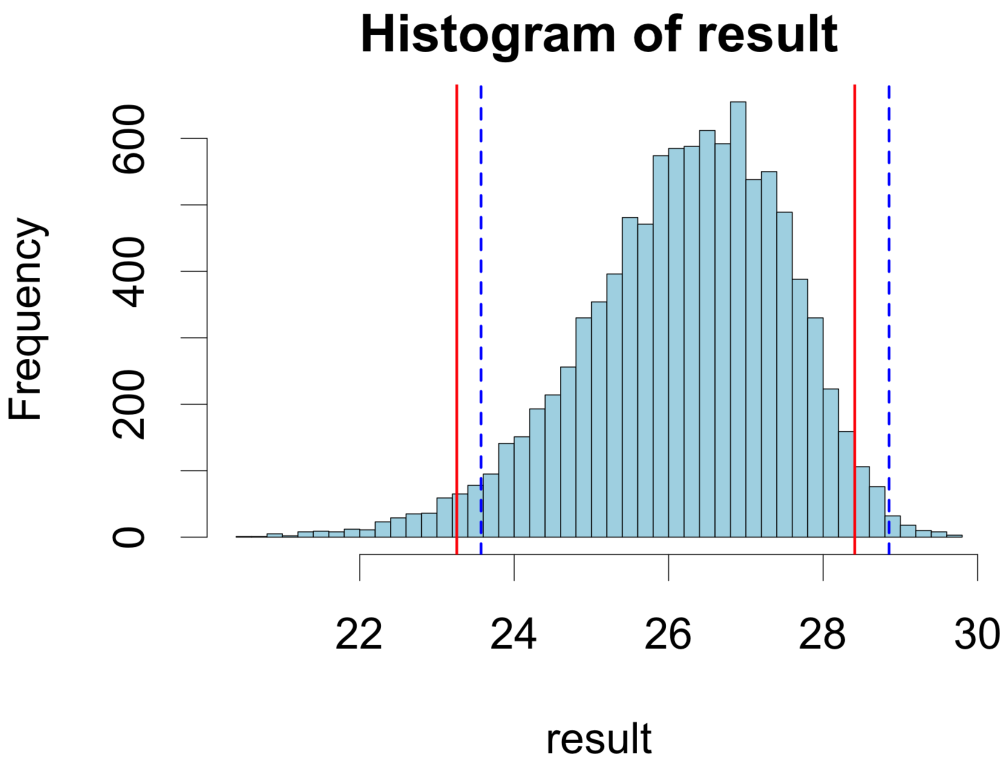
```
critical_values = qt(c(0.025,0.975), df = n-1)
critical_values
```

```
## [1] -1.997138  1.997138
```

```
CI_t = xbar + critical_values*se
CI_t
```

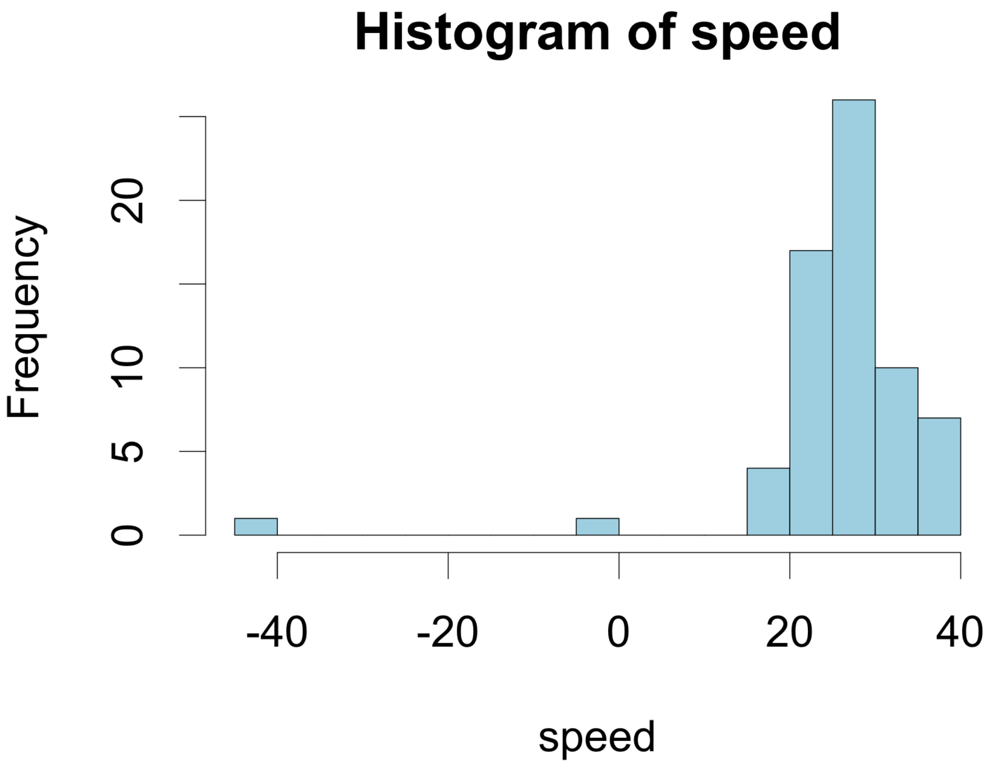
```
## [1] 23.57059 28.85365
```

```
hist(result, breaks = 50,
     col = "lightblue")
abline(v = CI, col = "red", lwd = 3)
abline(v = CI_t, col = "blue",
       lwd = 3, lty = 2)
```



What if we trimmed the data?

```
hist(speed, col = "lightblue",
     breaks = 15)
```



Keep only the positive speeds.

```
speed1 = speed[speed>0]
mean(speed)
```

```
## [1] 26.21212
```

```
mean(speed1)
```

```
## [1] 27.75
```

```r
B = 10000
result = vector("numeric", length = B)
for(i in 1:B){
  newData = sample(speed1, replace = TRUE)
  result[i] = mean(newData)
}
CI = quantile(result, c(0.025, 0.975))
CI
```

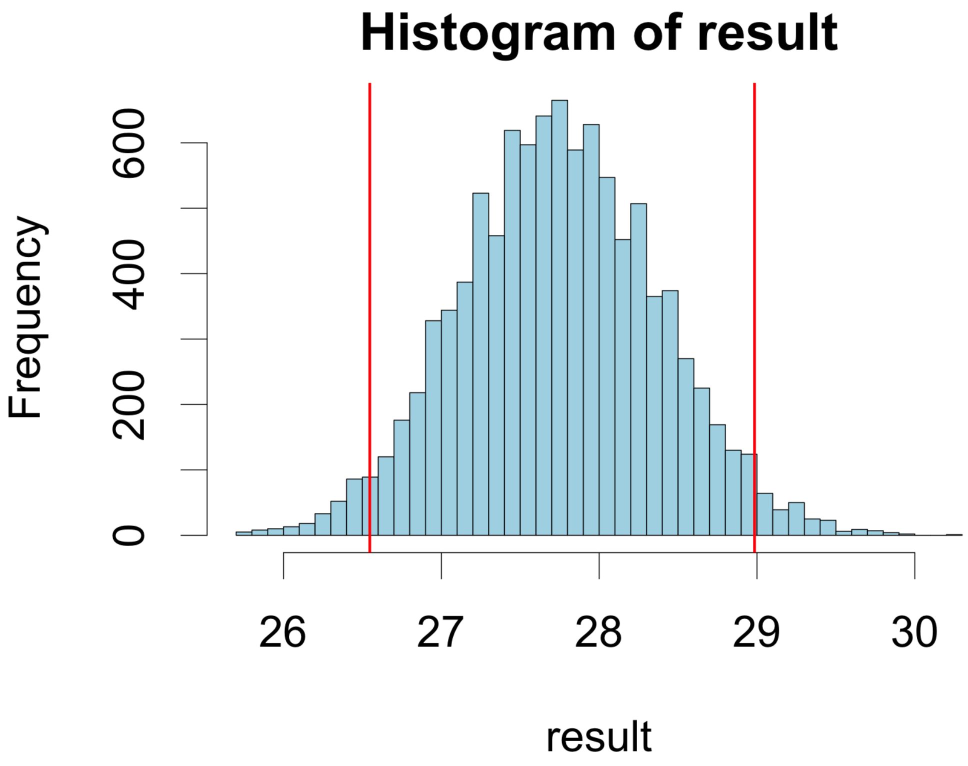
```
##     2.5%    97.5% 
## 26.54688 28.98438
```

```r
CI - mean(speed1)
```

```
##      2.5%     97.5% 
## -1.203125  1.234375
```

Much more symmetric.

```r
hist(result, breaks = 50, col = "lightblue")
abline(v = CI, col = "red", lwd = 3)
```



Compare with the confidence interval using the t distribution.

```
xbar = mean(speed1)
n = length(speed1)
se = sd(speed1)/sqrt(n)
c(xbar, n, se)
```

```
## [1] 27.7500000 64.0000000  0.6354289
```

```
critical_values = qt(c(0.025,0.975), df = n-1)
critical_values
```

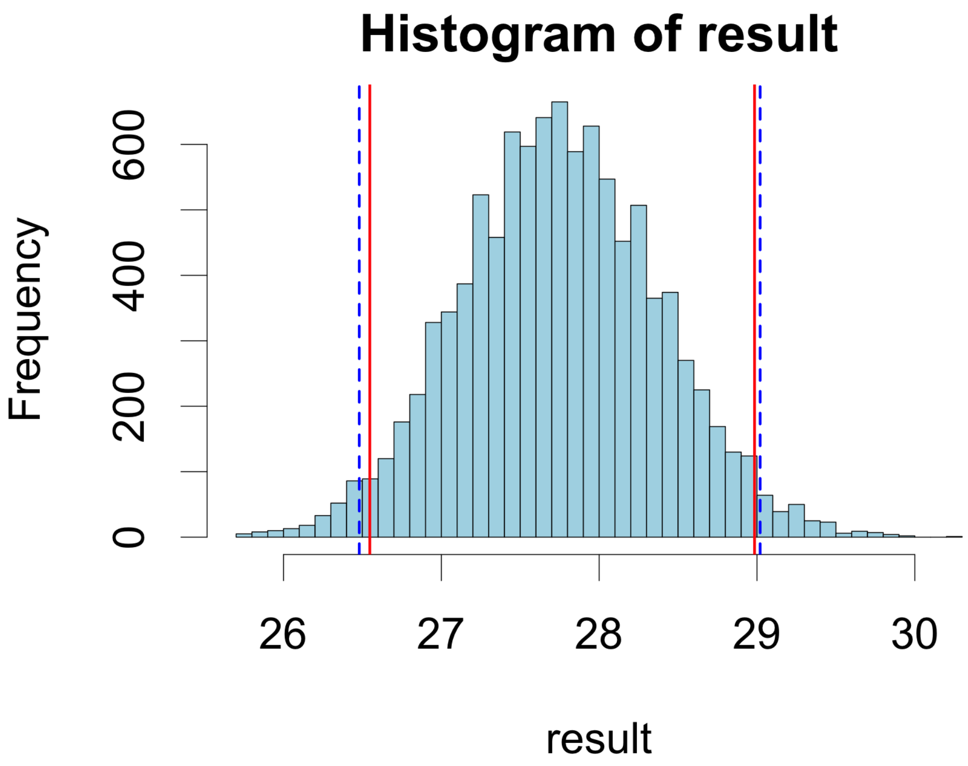
```
## [1] -1.998341  1.998341
```

```
CI_t = xbar + critical_values*se
CI_t
```

```
## [1] 26.4802 29.0198
```

The bootstrap and the t confidence intervals are now very similar.

```
hist(result, breaks = 50,
     col = "lightblue")
abline(v = CI, col = "red", lwd = 3)
abline(v = CI_t, col = "blue",
       lwd = 3, lty = 2)
```



Flight departure delays

Flights data set

```r
# install.packages("nycflights13")
library(nycflights13)
glimpse(flights)
```

```
## Rows: 336,776
## Columns: 19
## $ year           <int> 2013, 2013, 2013, 2013, 2013, 2013,…
## $ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…
## $ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…
## $ dep_time       <int> 517, 533, 542, 544, 554, 554, 555, …
## $ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, …
## $ dep_delay      <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2…
## $ arr_time       <int> 830, 850, 923, 1004, 812, 740, 913,…
## $ sched_arr_time <int> 819, 830, 850, 1022, 837, 728, 854,…
## $ arr_delay      <dbl> 11, 20, 33, -18, -25, 12, 19, -14, …
## $ carrier        <chr> "UA", "UA", "AA", "B6", "DL", "UA",…
## $ flight         <int> 1545, 1714, 1141, 725, 461, 1696, 5…
## $ tailnum        <chr> "N14228", "N24211", "N619AA", "N804…
## $ origin         <chr> "EWR", "LGA", "JFK", "JFK", "LGA", …
## $ dest           <chr> "IAH", "IAH", "MIA", "BQN", "ATL", …
## $ air_time       <dbl> 227, 227, 160, 183, 116, 150, 158, …
## $ distance       <dbl> 1400, 1416, 1089, 1576, 762, 719, 1…
## $ hour           <dbl> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6,…
```

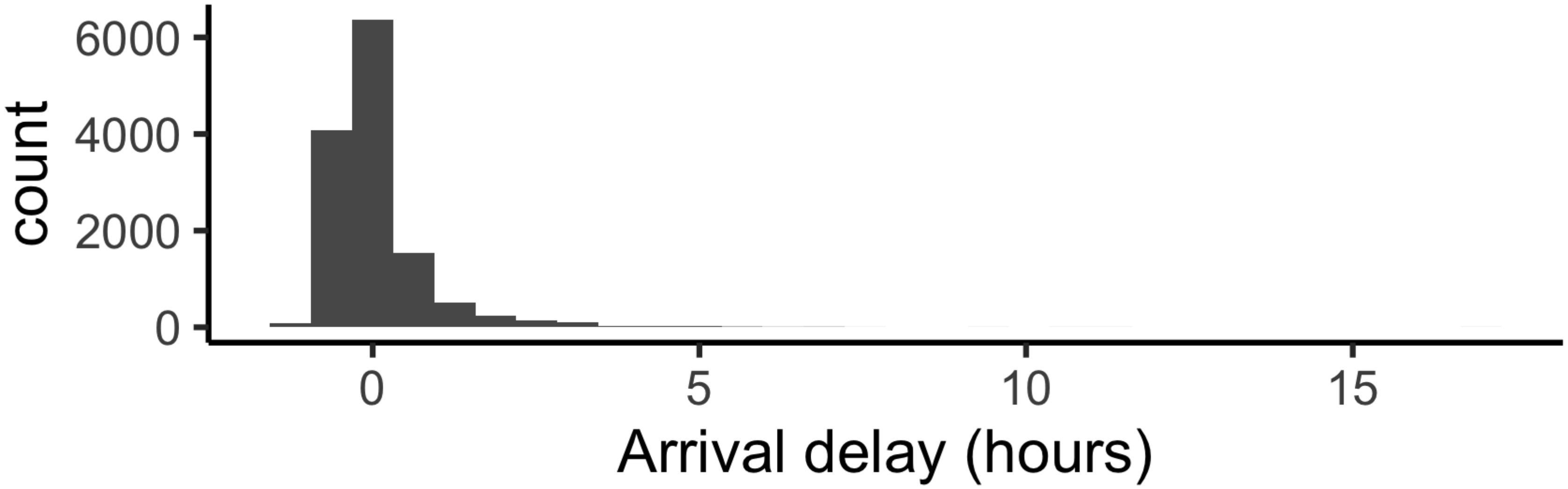
New York City to San Fransisco

Let's restrict attention to flights between NYC and San Francisco (SFO).

```
sfo = flights %>% filter(dest == "SFO")
```

This is the **population** of flights in 2013. Let's look at the distribution of arrival delays:

```
sfo %>% ggplot(aes(x = arr_delay/60)) + geom_histogram() + labs(x = "Arrival delay (hours)")
```



Travel policy

An organisation regularly flies staff from NYC to SFO. It decides that it is acceptable for staff to be late 2% of the time. How early should they book their flights to ensure that staff arrive on time?

```r
quantile(sfo$arr_delay, p = 0.98, na.rm = TRUE)
```

```
## 98%
## 153
```

The 98th percentile of the arrival delay distribution is about 2.5 hours, so we should send them on a flight about 2.5 hours early.

> What if we didn't have the population data?

Sample of flights

If all we had access to was a sample of 100 flights from 2013, this is our point estimate of the 98th percentile.

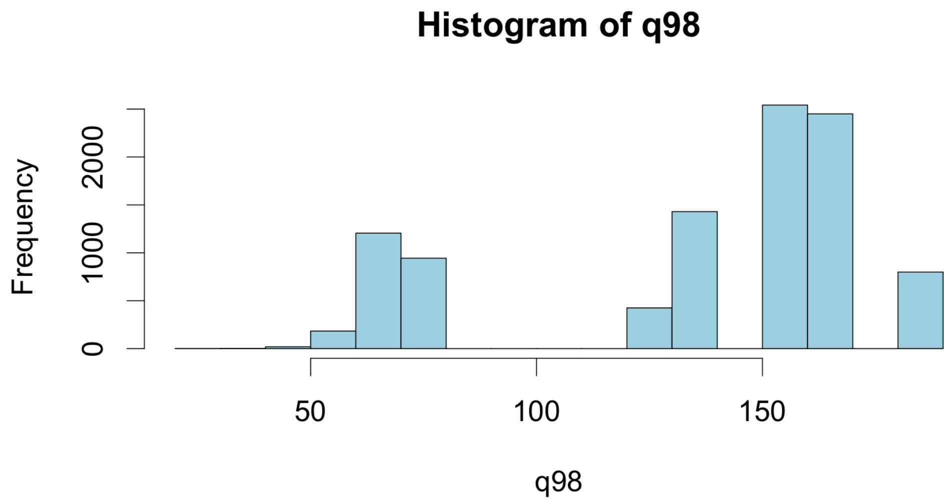
```
set.seed(2)
sfo_sample = sfo %>% filter(!is.na(arr_delay)) %>% sample_n(size = 100, replace = FALSE)
quantile(sfo_sample$arr_delay, p = 0.98)
```

```
##   98%
## 154.2
```

How reliable is that point estimate?

Bootstrap CI for quantiles

```r
B = 10000
q98 = vector("numeric", length = B)
for(i in 1:B) {
  resample = sample(sfo_sample$arr_delay,
                    replace = TRUE)
  q98[i] = quantile(resample, probs = 0.98)
}
par(cex = 2)
hist(q98, col = "lightblue")
```



Histogram of q98

A 95% confidence interval for this quantile:

```r
quantile(q98, c(0.025,0.975))
```

```
##     2.5%   97.5% 
##   61.451 182.000
```

Based on our sample our (bootstrap) 95% confidence interval is between 1 hour and 3 hours.

Final remarks

Bootstrapping is useful when

- the theoretical distribution of a statistic is complicated or unknown (e.g. coefficient of variation, quantile regression parameter estimates, etc.)
- the sample size is too small to make any sensible parametric inferences about the parameter

Advantages

- Bootstrapping frees us from making parametric assumptions to carry out inferences
- Provides answers to problems for which analytic solutions are impossible
- Can be used to verify, or check the stability of results
- Asymptotically consistent

References

Efron, B. (1979). "Bootstrap Methods: Another Look at the Jackknife". In: *The Annals of Statistics* 7.1, pp. 1-26. DOI: 10.1214/aos/1176344552.

Hall, P. (1989). "On efficient bootstrap simulation". In: *Biometrika* 76.3, pp. 613-617. DOI: 10.1093/biomet/76.3.613.

Manly, B. (2007). *Randomization, bootstrap and Monte Carlo methods in biology*. Boca Raton, FL: Chapman & Hall/CRC. ISBN: 9781584885412.

Stigler, S. M. (1977). "Do Robust Estimators Work with Real Data?". In: *The Annals of Statistics* 5.6, pp. 1055-1098. DOI: 10.1214/aos/1176343997.

Wickham, H. (2018a). *nycflights13: Flights that Departed NYC in 2013*. R package version 1.0.0. URL: https://CRAN.R-project.org/package=nycflights13.