O'REILLY

Vibing with Statistics

Thomas Nield

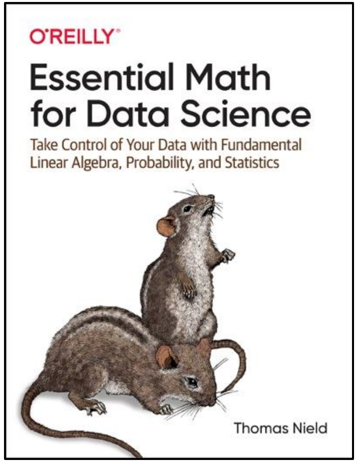
Meet your Instructor

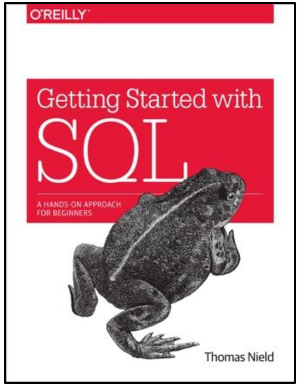
Thomas Nield

Instructor, Author, Inventor

- AI Safety and Security, University of Southern California
- Airline veteran w/ 10 years experience in operations research.
- Inventor of handheld flight control system, the Yawman Arrow.

https://www.linkedin.com/in/thomasnield/

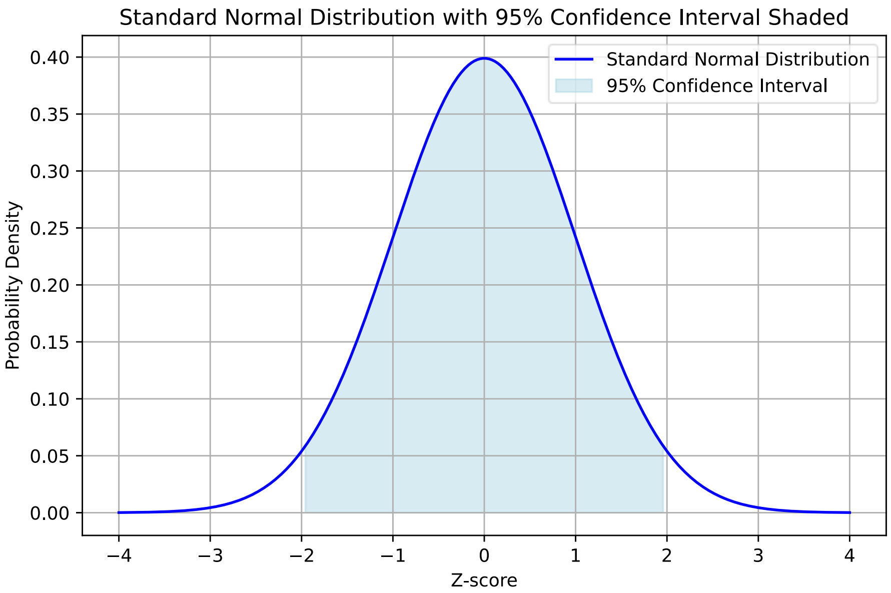

O'REILLY
Essential Math for Data Science
Take Control of Your Data with Fundamental Linear Algebra, Probability, and Statistics
Thomas Nield



Course schedule

What to Expect

1) **What is Statistics?**
2) **Measures of Central Tendency**
3) **Data Distributions**
4) **Normal Distribution and CLT**
5) **Hypothesis Testing**
6) **Ethics and p-hacking**



https://github.com/thomasnield/oreilly_vibing_with_statistics

O'REILLY

What is Statistics?

Thomas Nield

What is Statistics?

Making sense of data

- **Statistics** describes and infers truths from data, often taking the form of analyzing a sample to represent a larger population.

- It is an area of mathematics that collects, analyzes, interprets, and quantifies numerical data.

- Statistics can be applicable to any area that touches data, from software engineering to machine learning

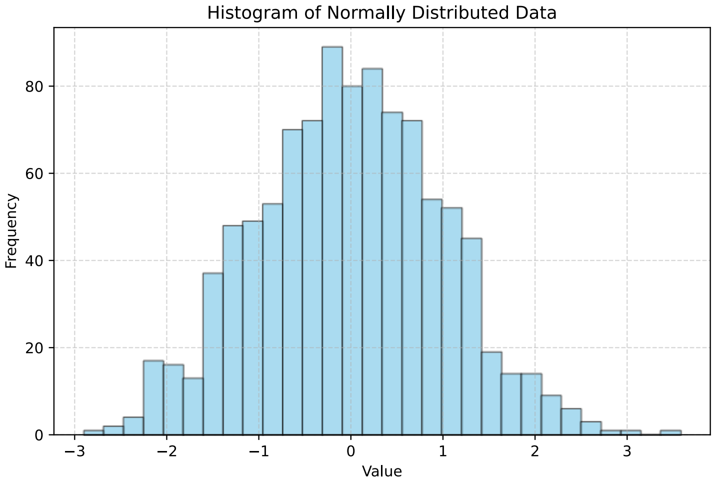
What Are the Benefits?

Making sense of data

Data is never going away, so there will always be a need for professionals who can get insight from it.

Benefits of learning statistics:

- Employability – Using domain knowledge with data proficiency, you can find signals others might miss.
- Data utilization – Create value from data that is otherwise unused.
- Machine learning/AI – Statistical automation is more productive with a statistics mindset.
- Better data collection – By understanding the relationship between a sample and a population, data projects can be more productive.



A Brief History

How did we get here?

Ancient Civilization

- Babylonians would tabulate records and written numbers.
- Roman Empire would take a census of all its territories.

17th Century

- Carl Friedrich Gauss and other mathematicians proposed formal methods.
- Gauss discovered the normal distribution, method of least squares, and maximum likelihood.
- Laplace proved the central limit theorem (CLT).

Carl Friedrich Gauss

By Gottlieb Biermann / After Christian Albrecht Jensen - Gauß-Gesellschaft Göttingen e.V. (Foto: A. Wittmann)., Public Domain, https://commons.wikimedia.org/w/index.php?curid=57629

A Brief History

How did we get here?

Early 20th Century

- Statistics began to find application in industry and economic policy.
- Modern statistics began to take shape.

Later 20th Century

- Giant mainframes processed data on punch cards.
- By the 1980’s, magnetic and digital storage streamlined data collection, and computers became smaller.
- In the 1990’s, the Internet created a medium of rapid data collection.
- Statisticians, researchers, and quants used MATLAB, SaaS, and SQL.

Apple IIe

By All About Apple museum official web site - http://www.allaboutapple.com/, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=1049963

A Brief History

How did we get here?

2000 - 2009

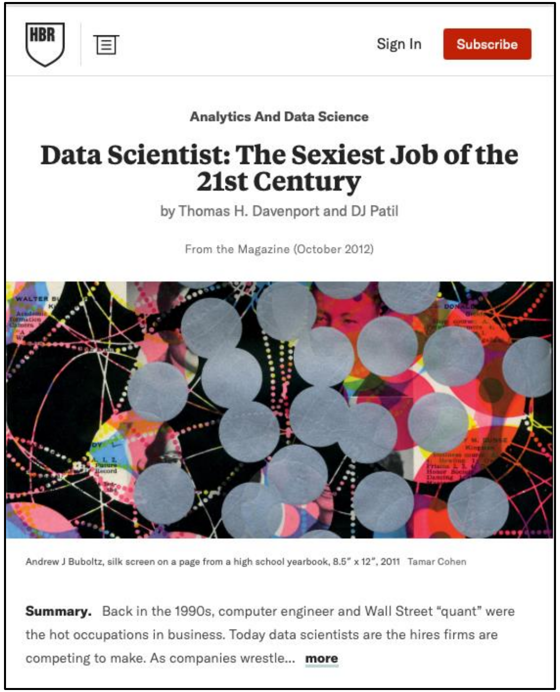
- Dotcom bubble went bust, a handful of winning companies like Amazon, Google, Facebook, and Yahoo emerged alongside Apple and Microsoft.
- Mobile devices created a powerful new data collection point that people carried with them.
- Google executives insisted that statisticians would have the “sexy” job for the next 10 years.

A Brief History

How did we get here?

2010 - 2020

- Data collection continued to happen at an unprecedented scale, and *cloud computing* made data infrastructure more accessible.
- Harvard Business Review declared data science *The Sexiest Job of the 21st Century*[1].
- The boundaries between data science, statistics, machine learning, AI, and computer science became blurred.
- *Big data*, *machine learning*, *deep learning,* and *analytics* went through boom and bust cycles.



Analytics And Data Science

Data Scientist: The Sexiest Job of the 21st Century

by Thomas H. Davenport and DJ Patil

From the Magazine (October 2012)

Andrew J Buboltz, silk screen on a page from a high school yearbook, 8.5" x 12", 2011 Tamar Cohen

Summary. Back in the 1990s, computer engineer and Wall Street "quant" were the hot occupations in business. Today data scientists are the hires firms are competing to make. As companies wrestle... **more**

[1] https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

A Brief History

How did we get here?

Post-2020

- Every digital interaction in our daily lives becomes a data point, and data proficiency is needed to find signals in the noise.
- Data is streamed into a vastly connected infrastructure trying to automate insights, predictions, and alerts.
- From quantitative trading to airline operations, the applications for statistics are universal wherever there is data.
- Contemporary machine learning and artificial intelligence are amplifications of statistics, scaling both the benefits and challenges.

What is Data?

A Thought Experiment

Imagine you were provided a photo of a family:

- Do you know this family’s story based on just one photo?
- What if you had 20 photos? 200 photos? 2000 photos? How many photos do you need to know their story?
- Do you need photos of them in different situations? Home, work, school, and vacation? Alone and together? With relatives and friends?

How much can we know about this family just through photos? And how many/what photos do we need?

Data is Snapshots!

Just like photographs

Data is just like photographs; it provides snapshots of a story at discrete points in time.

The continuous reality and contexts of the story are not captured, as well as the infinite number of variables in that story.

Today there is a great deal of emphasis on data collection, promoting the idea it can intelligently tell a story, even predict what happens next, and thus create artificial intelligence.

Data is Snapshots!

The Problem of Scope

But how realistic is this objective? How narrow must our scope be to make it feasible?

A few strategic photos of the father playing golf can easily tell us whether he is good at golf.

- A photo of him captured mid-swing
- A photo of him cheerful or lamenting at the 18th hole
- A photo of his scorecard!

But trying to decipher his entire life story through photos… that is a much harder problem.

Data is Snapshots!

The Problem of Scope

Even with a narrow objective, it can still be hard to determine what is **ground truth**, or what is *actually* true given the data.

Let’s say we are trying to evaluate whether the father is good at golf based off those few chosen photos.

- If we catch him fist-pumping at the hole, is he cheering for himself or someone else?
- If we take a picture of the scorecard, how can we be sure it was not forged?

This just goes to show that data can be taken out of context or even forged, and it becomes even more important to realize that data provides clues to the truth and is not actually the source of truth.

Data is Everywhere!

It is collected in so many ways

Connected devices like the Apple Watch

A.I. Is Learning From Humans. Many Humans.

Artificial intelligence is being taught by thousands of office workers around the world. It is not exactly futuristic work.

Manual data entry

Environment sensors like LIDAR

Drones broadcasting GPS location

Aircraft sensors

Apps and websites

Surveys and paperwork

We Can Trust Data, Right?

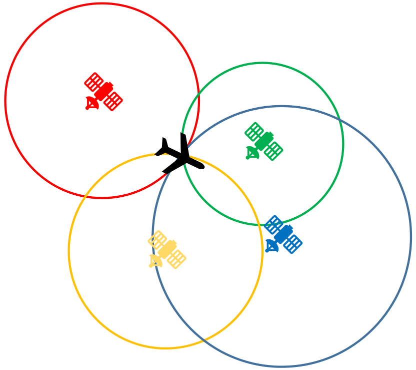
WRONG!

Let’s take getting a “precise” location of a vehicle using GPS. Simple right? *WRONG!!!!*

GPS approximates with 4 satellites at a time to determine location, and the faster the vehicle is going the more error it will accumulate.

Vehicles can supplement for this error using dead reckoning and radio ground stations, but faster speeds still will create errors.

This error may not matter for a food delivery service, but it does matter for a vehicle that can right-turn too early or an airplane that self-lands!

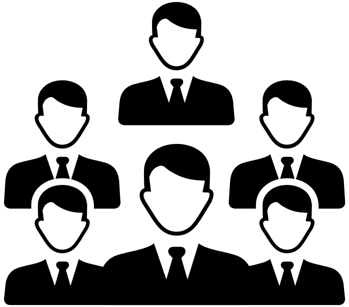


Samples versus Populations

The heart of statistics

A population is a particular group of interest we want to study:

- All MacBook Pro M4 owners in Texas
- All registrants in the 7pm cycling class at a local gym
- All possible road events a self-driving car could encounter
- All adult collies in Scotland



Note how we have well-defined boundaries for our population, allowing us to clearly identify who/what we are trying to measure.

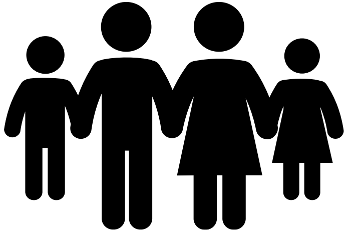
Some populations are accessible, but most in practice are not.

Samples versus Populations

The heart of statistics

A sample is a subset of the population that we use to infer attributes about the population.

- Because it’s almost always impractical to get a hold of a population, we use samples to infer information about the population instead.
- In a perfect world, a sample is random and unbiased.
- In machine learning, we treat the training data as a sample from a bigger domain that we are trying to model.



Measures of Central Tendency and Dispersion

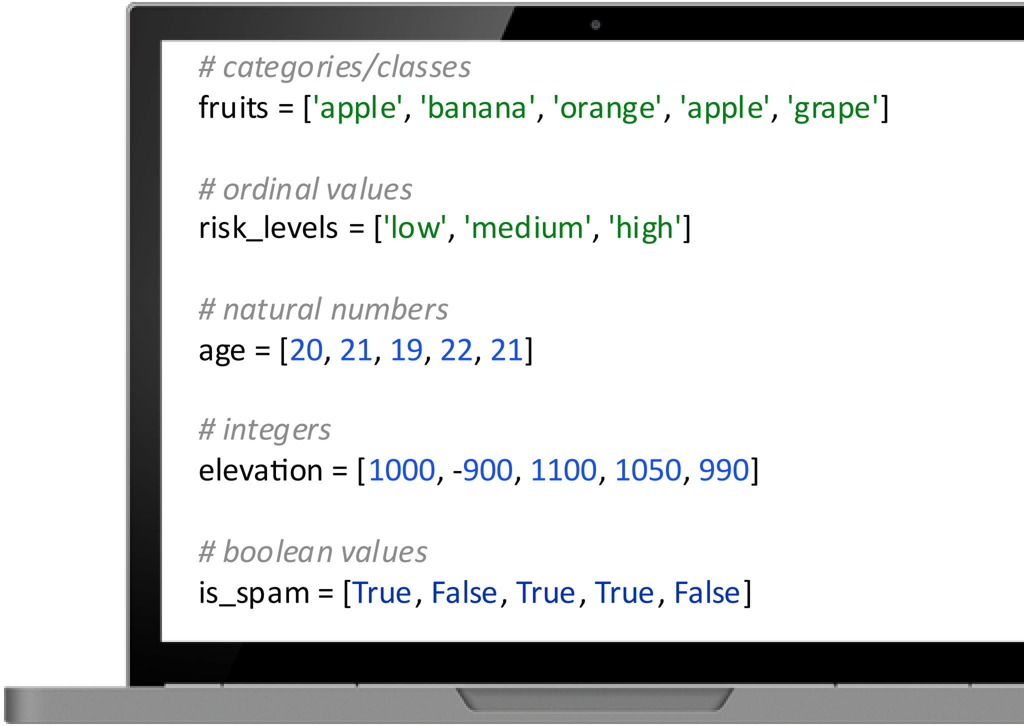
Thomas Nield

Discrete Values

Classes, Integers, Booleans, etc.

Discrete values are countable values that cannot be subdivided meaningfully.

To the right are examples of discrete values in Python.



```
# categories/classes
fruits = ['apple', 'banana', 'orange', 'apple', 'grape']

# ordinal values
risk_levels = ['low', 'medium', 'high']

# natural numbers
age = [20, 21, 19, 22, 21]

# integers
elevation = [1000, -900, 1100, 1050, 990]

# boolean values
is_spam = [True, False, True, True, False]
```

SOURCE: 2_measures_of_central_tendency/1_categorical_data.py

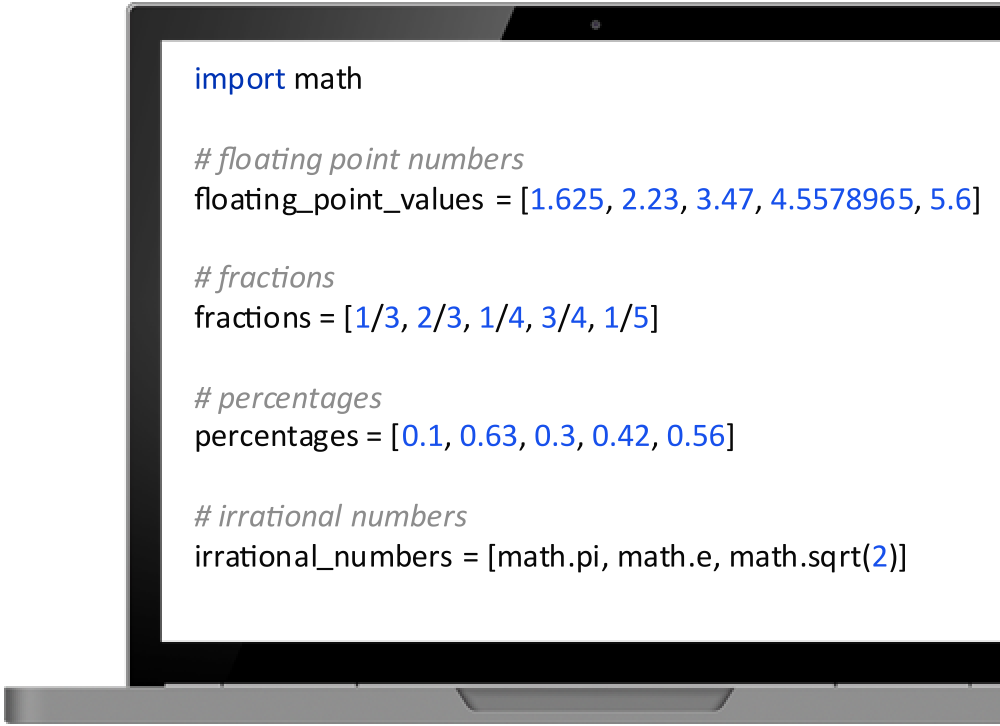
Continuous Values

Fractions, decimals, floating point values

Continuous values can contain *any* value in a range, including fractions and decimals.

This means a finite range, like 0 through 1, can contain an infinite number of values.

This is because decimals can have an infinite number of digits after the point.



```
import math

# floating point numbers
floating_point_values = [1.625, 2.23, 3.47, 4.5578965, 5.6]

# fractions
fractions = [1/3, 2/3, 1/4, 3/4, 1/5]

# percentages
percentages = [0.1, 0.63, 0.3, 0.42, 0.56]

# irrational numbers
irrational_numbers = [math.pi, math.e, math.sqrt(2)]
```

SOURCE: 2_measures_of_central_tendency/2_continuous_data.py

Mean

a.k.a. the “average”

The **mean** is the average of numerical values, taking the sum and dividing by the number of values in the sample/population.

It shows where the center of gravity is, or the central location is.

Sample

$$\bar{x} = \frac{\sum x_i}{n}$$

Population

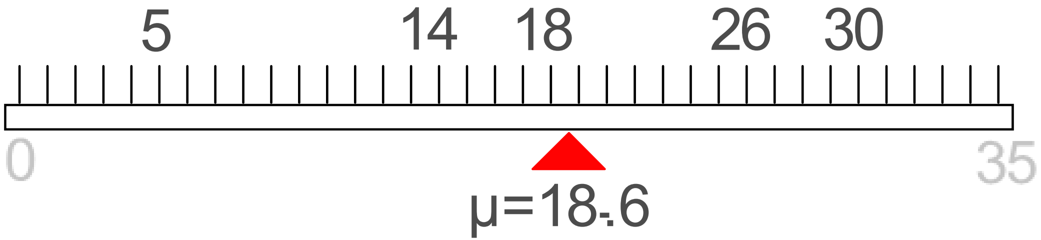
$$\mu = \frac{\sum x_i}{N}$$

Mean

a.k.a. the "average"

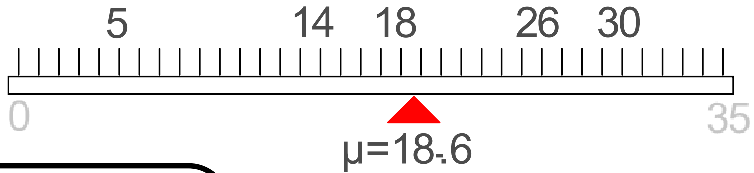
Think of the mean as the fulcrum on a seesaw holding the number line of values.

That fulcrum's position to balance the numbers would be the mean (and each number weighs the same).



Mean

Calculating the mean in NumPy



```
import numpy as np

iphone_case_weights = np.array(
 [7,14,15,21,27,
  33,34,37.5,39,40]
)
iphone_case_weight_mean = np.mean(iphone_case_weights)

# iPhone 15 Case Weight Mean (grams): 26.75
print("iPhone 15 Case Weight Mean (grams):", iphone_case_weight_mean)
```

Median

The ”middlemost” value

The **median** is the middlemost value in an ordered sequence of values.

Odd number of elements

$$1, 3, 5, \boxed{9,} 11, 13, 105$$

Even number of elements
(take mean of two centermost values)

$$1, 3, 5, \boxed{9, 11}, 13, 105, 199$$

$$\frac{9+11}{2} = 10$$

Median

The ”middlemost” value

The median is useful because it is less sensitive to outliers than the mean.

To the right we have some restaurant tabs, but one of them is extremely high compared to the rest.

When the median is far from the mean, you know the data is skewed.

```
import numpy as np

dining_tabs = [35, 47, 97, 103, 109, 135,
               172, 194, 205, 225, 2100]

mean = np.mean(dining_tabs)
median = np.median(dining_tabs)

print(f"Mean: {mean:.4f}")     # Mean: 311.0909
print(f"Median: {median:.4f}") # Median:  135.0000
```

SOURCE: 2_measures_of_central_tendency/4_median.py

Mode

The most common value

The mode is the most frequently occurring value.

It is primarily useful for repetitive, discrete data.

If multiple values are tied in frequency count, then

```
from scipy.stats import mode

dining_party_size = [1,2,2,2,2,2,3,3,
                      4,4,4,4,5,7,9,11]

mode, ct = mode(dining_party_size)

# Mode: 2, Count: 5
print(f"Mode: {mode}, Count: {ct}")
```

SOURCE: 2_measures_of_central_tendency/5_mode.py

Percentiles

The median is a 50 percentile

Percentiles reflect data points below which a specified percentage of ordered data falls under.

The median is the 50th percentile.

The **quartiles** are the 25, 50, and 75th percentiles.

A **quantile** is like a percentile, but allows fractional values

```
import numpy as np

dining_tabs = [35, 47, 97, 103, 109, 135, 172, 194, 205, 225, 2100]
quartiles = np.percentile(dining_tabs, [25, 50, 75])

print("Quartiles:", quartiles)
# Quartiles: [100.  135.  199.5]

tertiles = np.percentile(dining_tabs, [33, 66])
print("Tertiles:" , tertiles)
# Tertiles: [104.8 185.2]
```

SOURCE: 2_measures_of_central_tendency/6_percentile.py

Range and Interquartile Range

Getting a sense of spread

We can extend the percentile/quartile concept to get a sense of how spread out the values are.

The **range** is the max minus the min value, showing how spread the data is.

The **interquartile range (IQR)** subtracts the 25th from the 75th percentile, to get a range removing outliers.

```python
import numpy as np

dining_tabs = [35, 47, 97, 103, 109, 135, 172, 194, 205, 225, 2100]
quartiles = np.percentile(dining_tabs, [25, 50, 75])
min, max = np.min(dining_tabs), np.max(dining_tabs)

print(f"Range: {max-min}")
# Range: 2065

print(f"Interquartile Range: {quartiles[2]-quartiles[0]}")
# Interquartile Range: 99.5
```

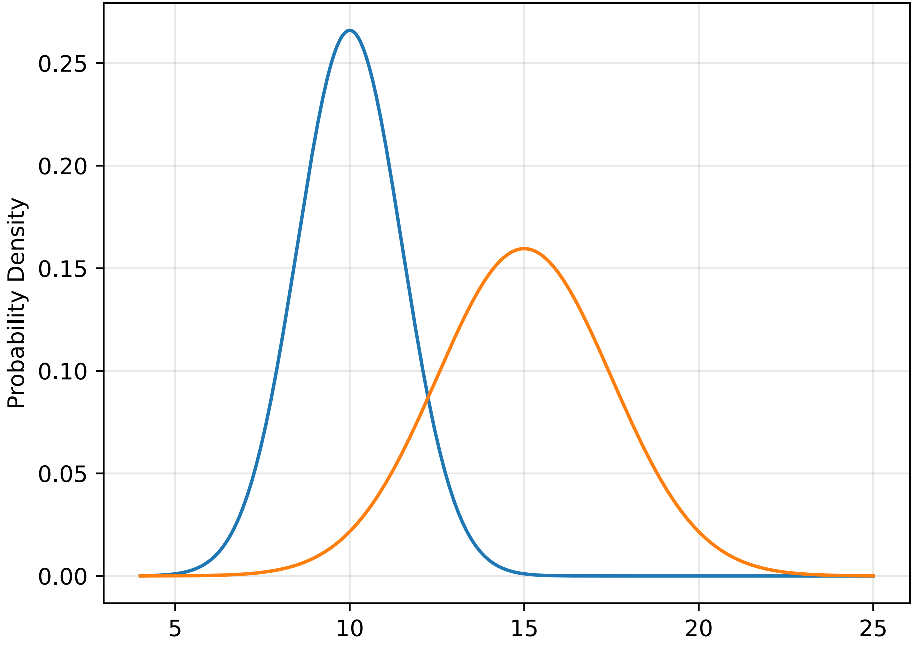
SOURCE: 2_measures_of_central_tendency/7_range_iqr.py

Variance

The most common value

We want data to be predictable, but sometimes it is **dispersed**, or spread out.

Measures like interquartile range (IQR), variance, and standard deviation measure how tightly or loosely data clusters around a central value.



SOURCE: 2_measures_of_central_tendency/8_visualizing_spread.py

Variance

Measuring spread

Variance is a measure of spread that takes the squared differences between the mean and each data point, then averages them.

A sample is going to slightly increase the variance by dividing $n - 1$ unlike a population which uses N.

Sample

$$s^2 = \frac{\sum (x_i - \bar{x})^2}{n - 1}$$

Population

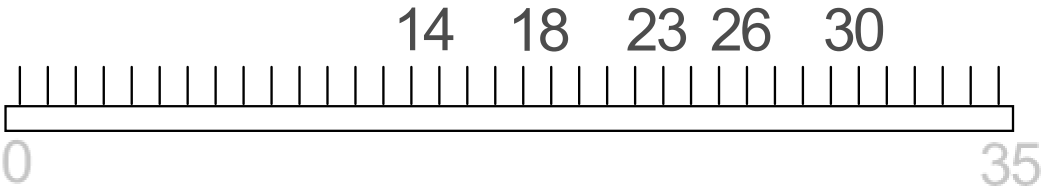
$$\sigma^2 = \frac{\sum (x_i - \mu)^2}{N}$$

Variance

Measuring Spread

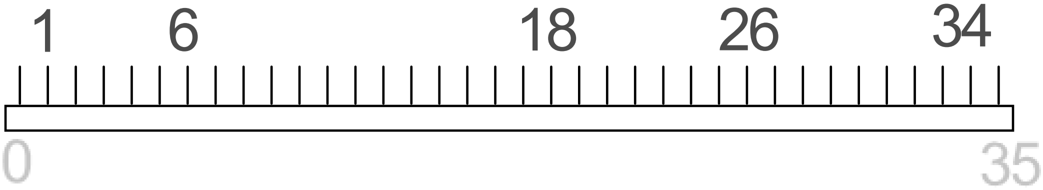
LOW VARIANCE

$$s^2 = 40.2$$



HIGH VARIANCE

$$s^2 = 187$$



Standard Deviation

Measuring Spread

The **standard deviation** is nothing more than the square root of the variance.

It’s a bit more meaningful to interpret because it “undoes” all the squaring that happened in the variance operation.

Sample

$$s^2 = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}}$$

Population

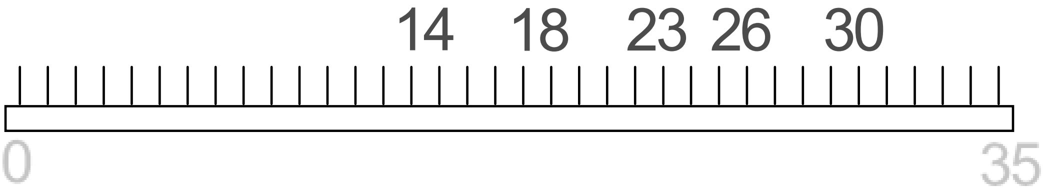
$$\sigma^2 = \sqrt{\frac{\sum (x_i - \mu)^2}{N}}$$

Standard Deviation

Measuring Spread

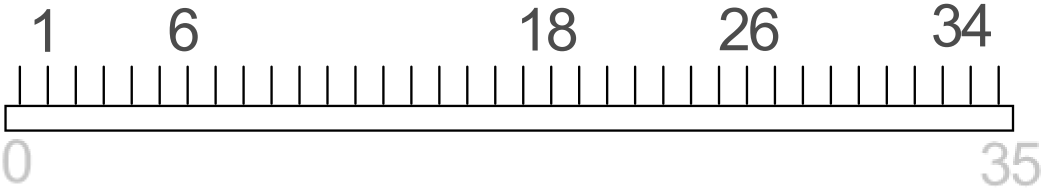
$s = 6.34$

LOW VARIANCE



$s = 13.67$

HIGH VARIANCE



Calculating Variance and Standard Deviation

```python
from math import sqrt

temps = [79,75,68,85,60,65,81,71,75,73,
    79,80,61,84,72,69,82,81,58,63,
    77,83,80,80,88,76,67,73,80,81,82]

def variance(data: [int]):
  # calculate mean
  mean = sum(data) / len(data)

  # sum all the squared differences
  squared_diffs = 0
  for x in data:
    squared_diffs += (x - mean) ** 2

  # take the average of the squared differences
  return squared_diffs / (len(data) - 1)

# just take the square root of variance
def standard_deviation(data: [int]):
  return sqrt(variance(data))

# Standard Deviation:  7.942941683069641
print("Standard Deviation: ", standard_deviation(temps))
```

"From scratch" in Python

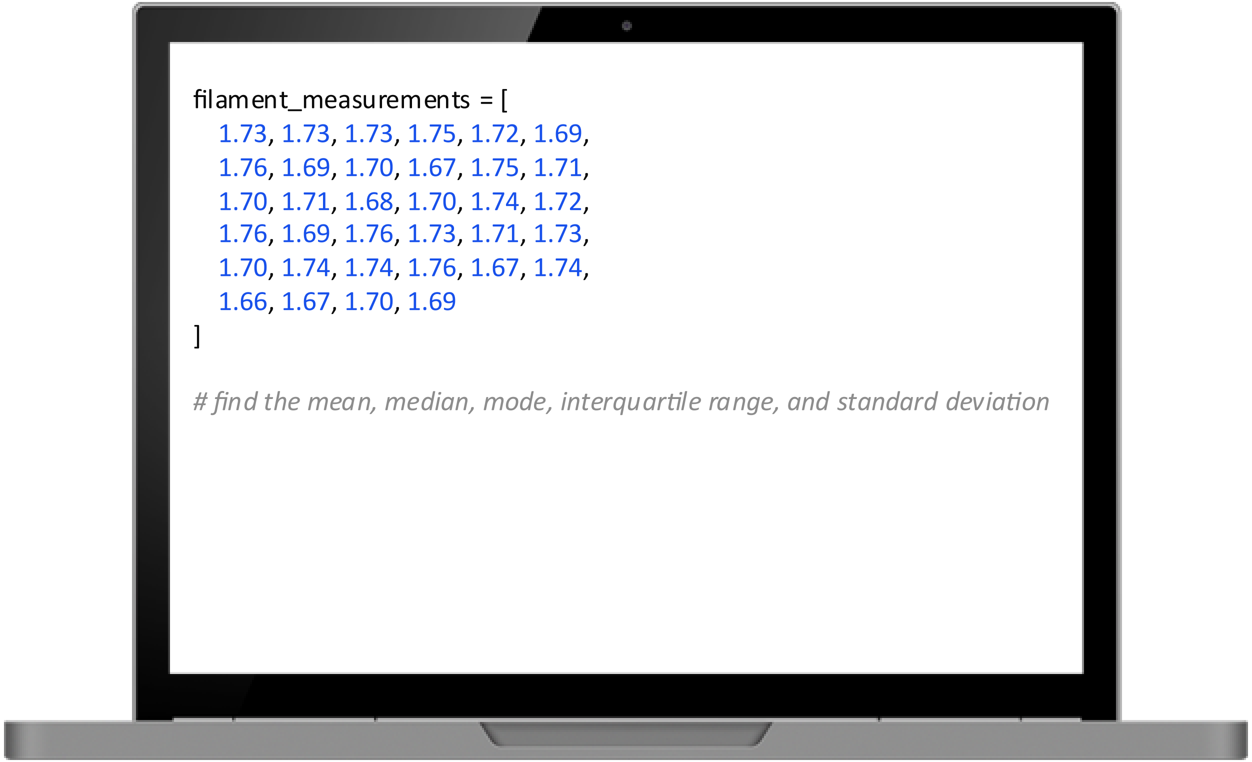
```python
import numpy as np

temps = [79,75,68,85,60,65,81,71,75,73,
    79,80,61,84,72,69,82,81,58,63,
    77,83,80,80,88,76,67,73,80,81,82]

# From NumPy: 7.942941683069641
print("From NumPy: ", np.std(temps, ddof=1))
```

NumPy

SOURCE: 2_measures_of_central_tendency/9_calculating_variance_std.py

Exercise

2_measures_of_central_tendency/10_exercise.py



```
filament_measurements = [
    1.73, 1.73, 1.73, 1.75, 1.72, 1.69,
    1.76, 1.69, 1.70, 1.67, 1.75, 1.71,
    1.70, 1.71, 1.68, 1.70, 1.74, 1.72,
    1.76, 1.69, 1.76, 1.73, 1.71, 1.73,
    1.70, 1.74, 1.74, 1.76, 1.67, 1.74,
    1.66, 1.67, 1.70, 1.69
]

# find the mean, median, mode, interquartile range, and standard deviation
```

Data Distribution

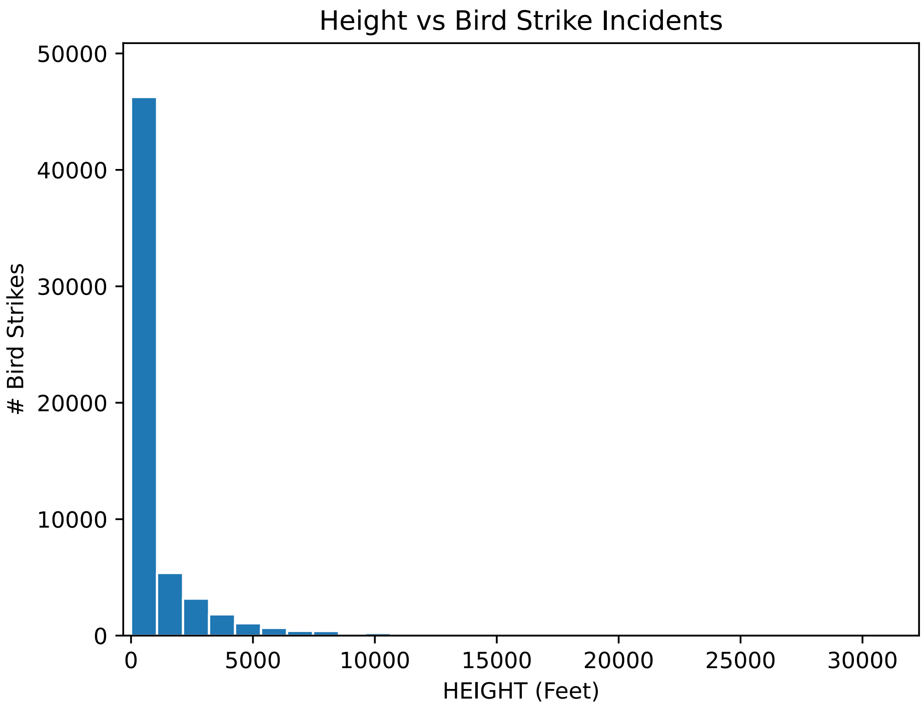
Thomas Nield

Histograms

What shape is it?

Histograms are a type of bar chart that counts the number of data points (continuous values) in equally ranged bins.

Those resulting bars reflecting the counts provide a sense of which values occur most frequently.



SOURCE: 3_data_distribution/1_histogram.py

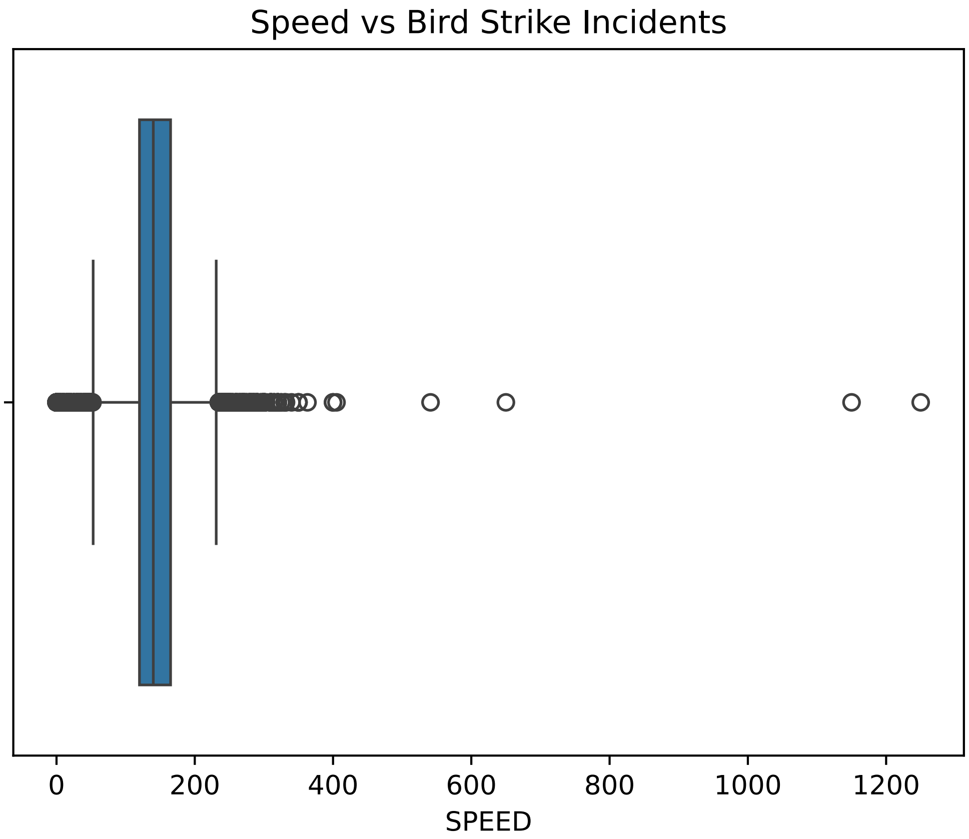
Box Plots

Visualizing the Interquartile Range

Box plots are a visualization of the interquartile range.

A box captures the 25th and 75th percentile, and the whiskers capture everything that is not within $1.5 \times IQR$ of the box.

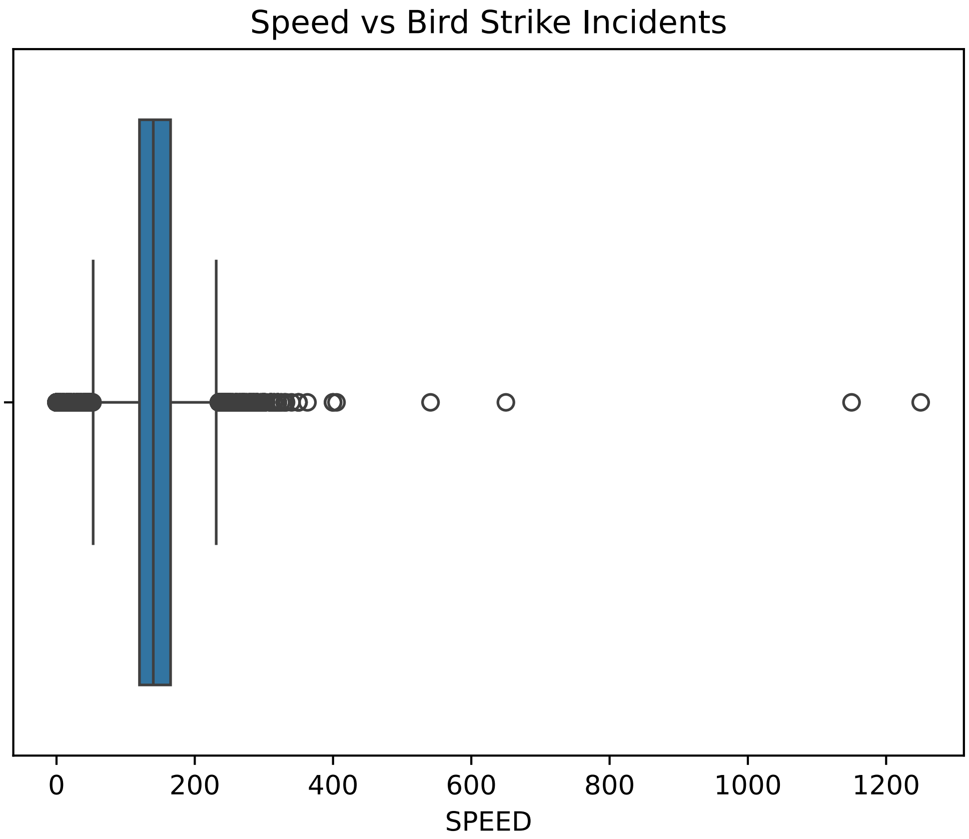
The dots outside the whiskers are extreme outliers that should be studied (e.g., for removal, concern, etc.)



SOURCE: 3_data_distribution/2_box_plot.py

Deconstructing the Box Plot

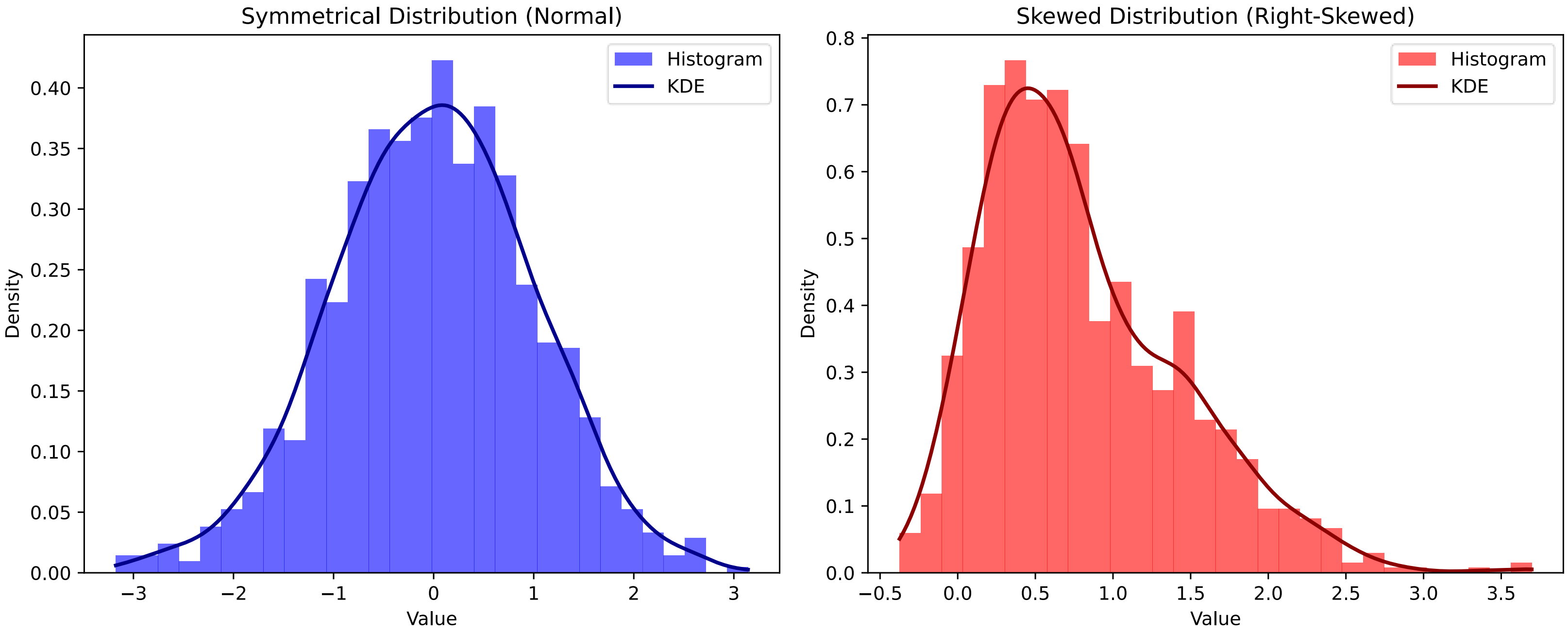
```
from numpy import percentile
import pandas as pd

# Get SPEED column, remove missing values
X = pd.read_csv('birdstrike.csv', usecols=["SPEED"]).dropna()

# these would bound the box
q25 = percentile(X, 25)
q75 = percentile(X, 75)

iqr = q75 - q25

# Box range: 120.00 - 165.00
print(f"Box range: {q25:.2f} - {q75:.2f}")

# these would bound the whiskers
k = 1.5
cut_off = iqr * k
lower = q25 - cut_off
upper = q75 + cut_off

# Whisker range: 52.50 - 232.50
print(f"Whisker range: {lower:.2f} - {upper:.2f}")
```



SOURCE: 3_data_distribution/3_iqr_cutoff.py

Symmetry vs Skewness

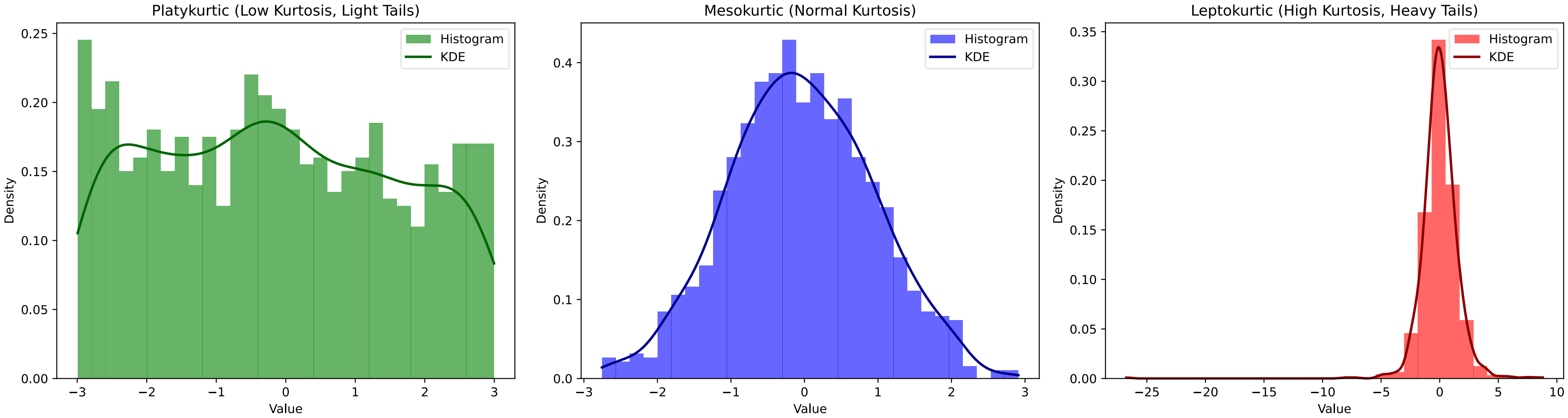
Mirrored or lop-sided?



SOURCE: 3_data_distribution/assets/5_symmetry_skewness.svg

Kurtosis

How distinctly present are any “tails” in the distribution?



SOURCE: 3_data_distribution/assets/5_symmetry_skewness.svg

Hands-On Time!

NOAA Tornado Dataset

Let’s walk through this notebook and do some hands-on analysis with what we’ve learned so far.

3_data_distribution/7_tornado_noaa.ipynb

We will do an exercise at the end of the notebook.

Normal Distribution and Central Limit Theorem

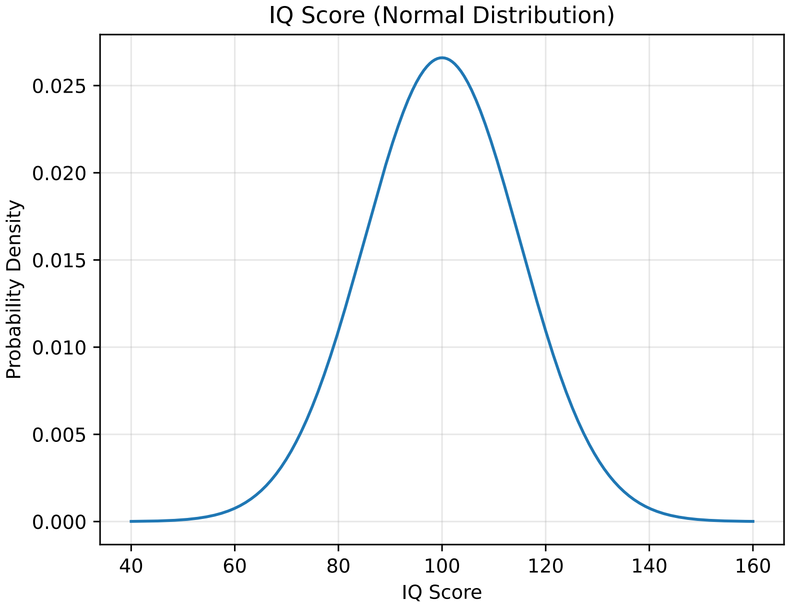
Thomas Nield

The Normal Distribution

a.k.a., the bell curve, the Gaussian distribution

The **normal distribution** is the most famous probability distribution, forming a bell-shaped curve with symmetrical properties.

Even non-normal data can apply the normal distribution through the central limit theorem (CLT) under the right conditions.



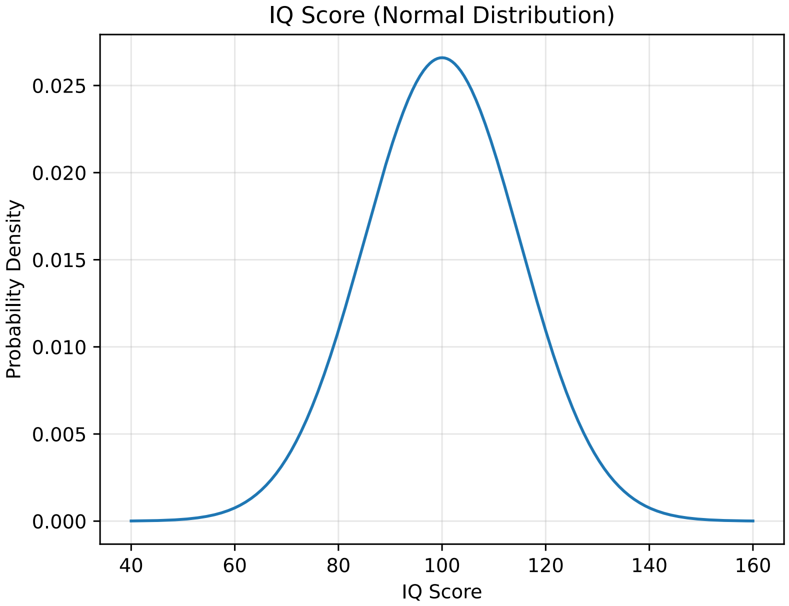
SOURCE: 4_normal_distribution_clt/1_normal_distribution.py

The Normal Distribution

a.k.a., the bell curve, the Gaussian distribution

Examples of real-world datasets that approximate normal distributions:

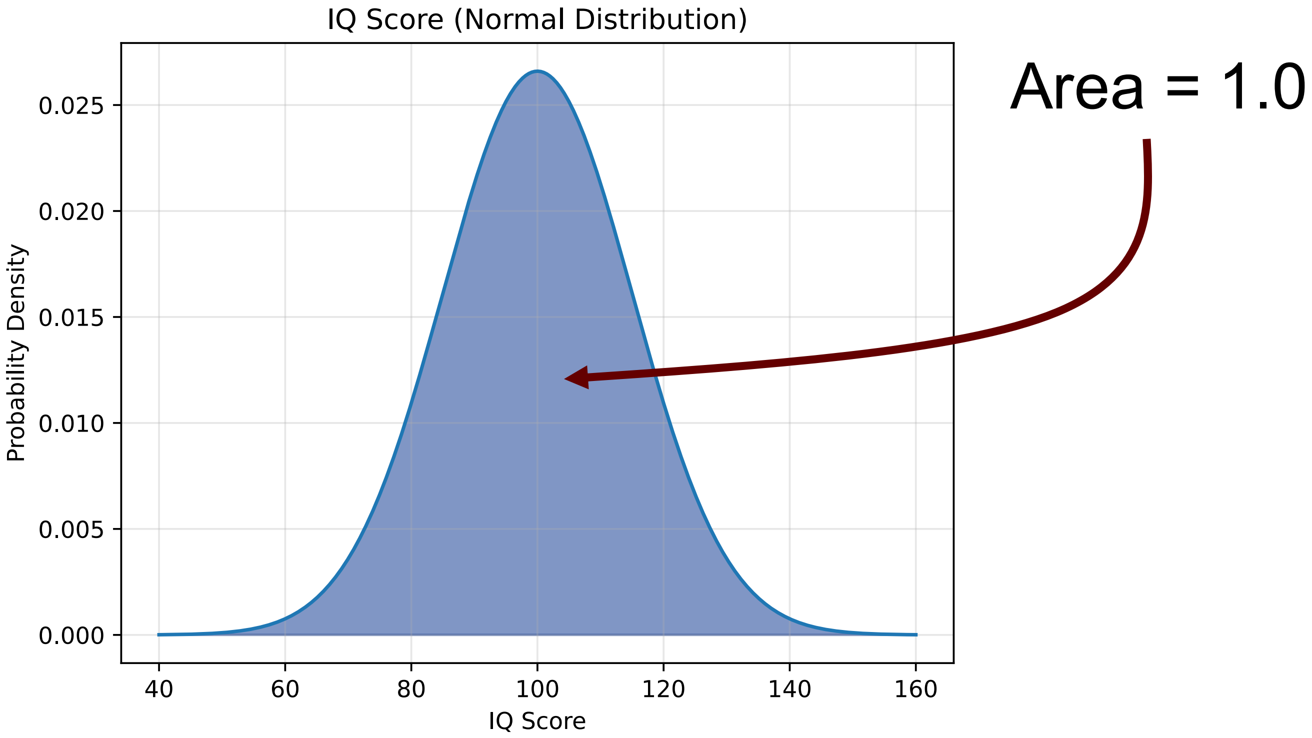
- **Engineering** –e.g., measurement errors)
- **Nature** – e.g., heights or weights
- **Medicine** – e.g., blood pressure or cholesterol levels
- **Social sciences** – e.g., IQ scores or standardized tests



SOURCE: 4_normal_distribution_clt/1_normal_distribution.py

The Normal Distribution

Properties – The area under the whole curve is 1.0

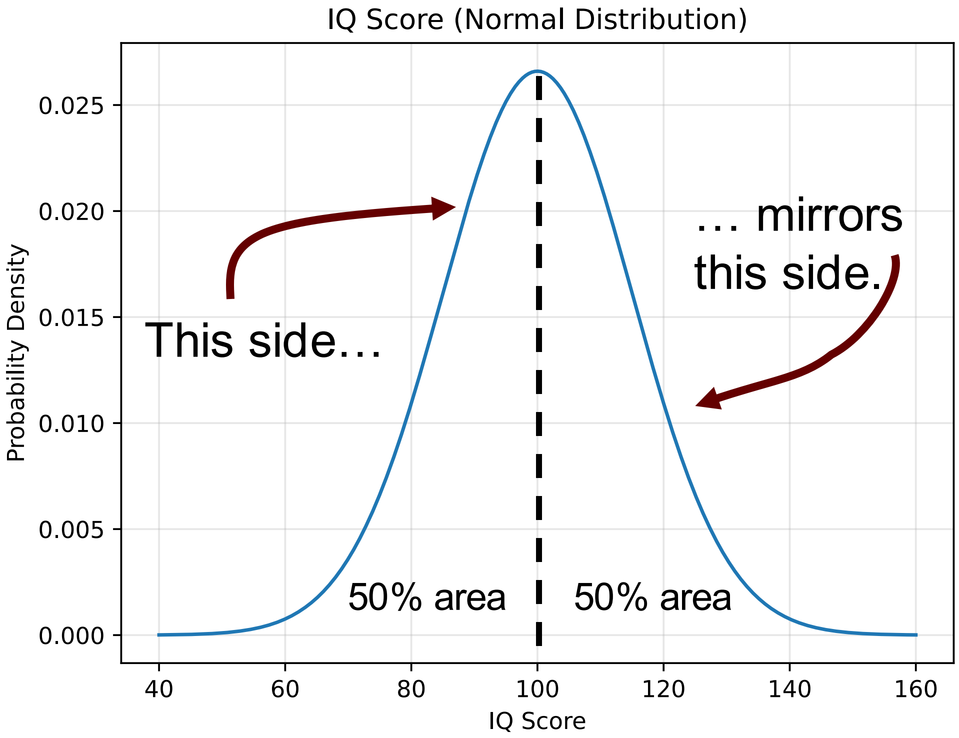


Area = 1.0

SOURCE: 4_normal_distribution_clt/2_area_one.py

The Normal Distribution

Properties – It’s symmetrical

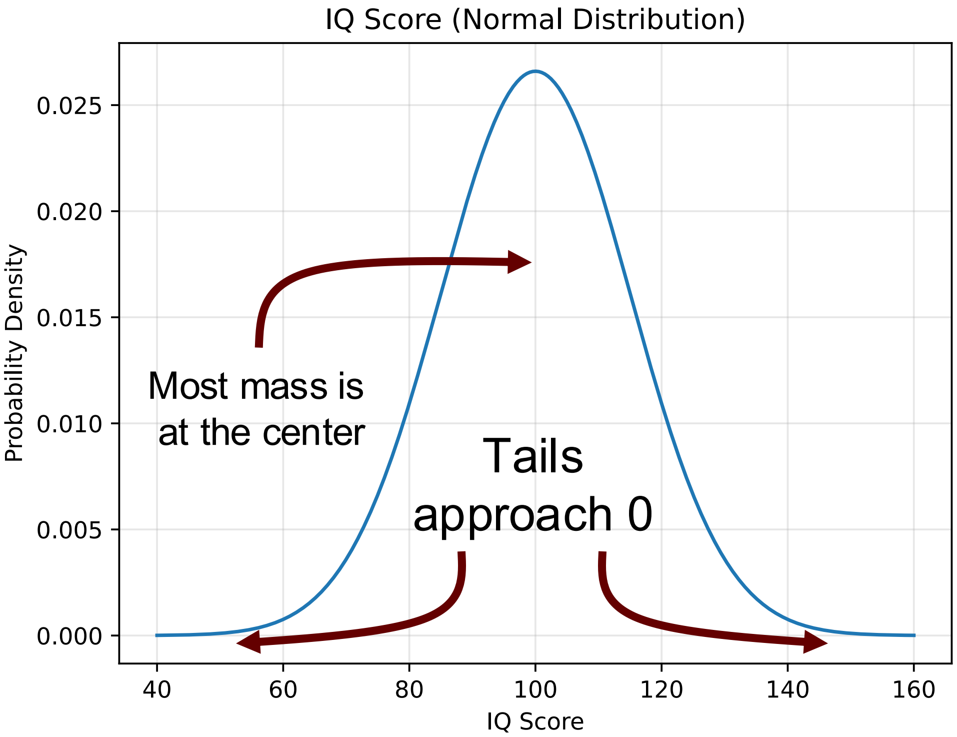


This side…

… mirrors this side.

The Normal Distribution

Properties – Tails approach 0

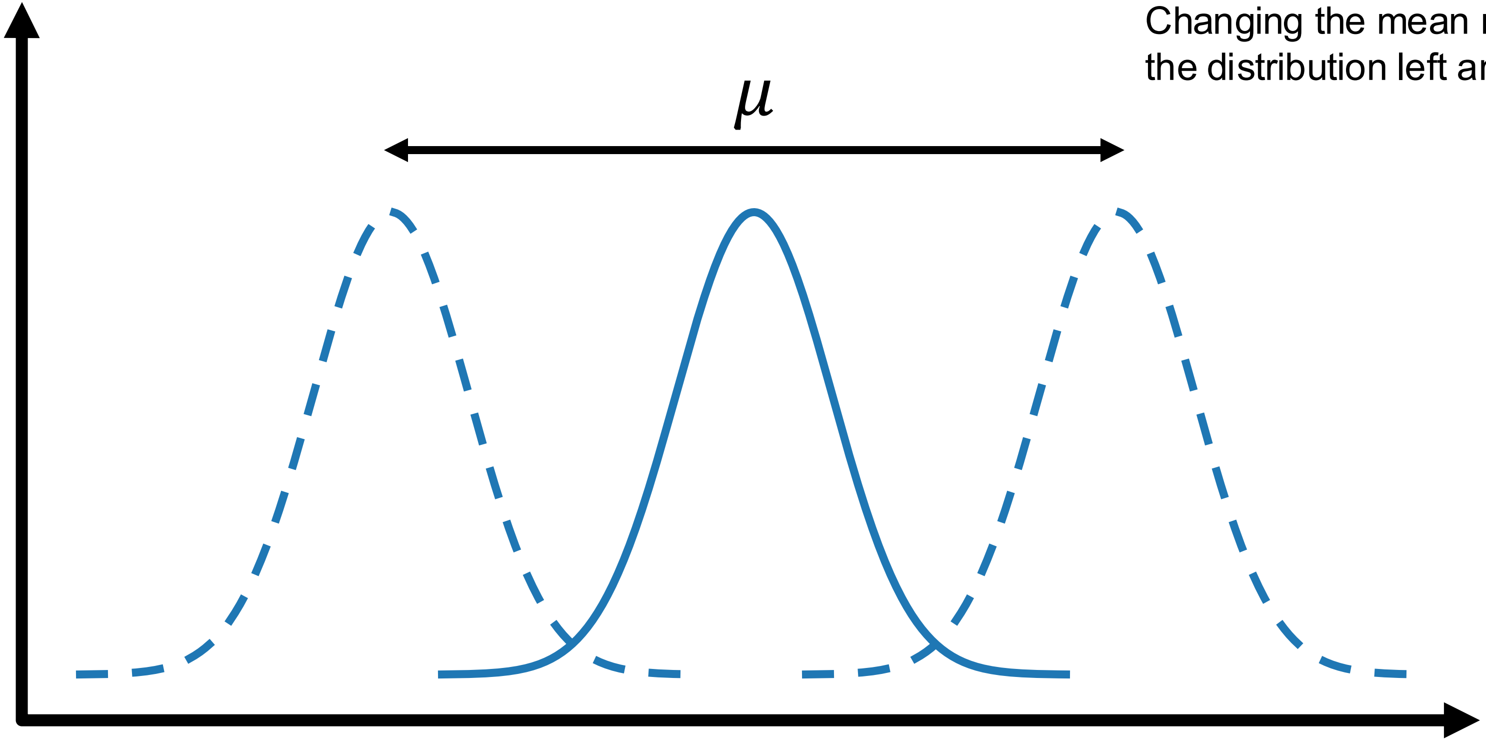


Most mass is at the center

Tails approach 0

The Normal Distribution

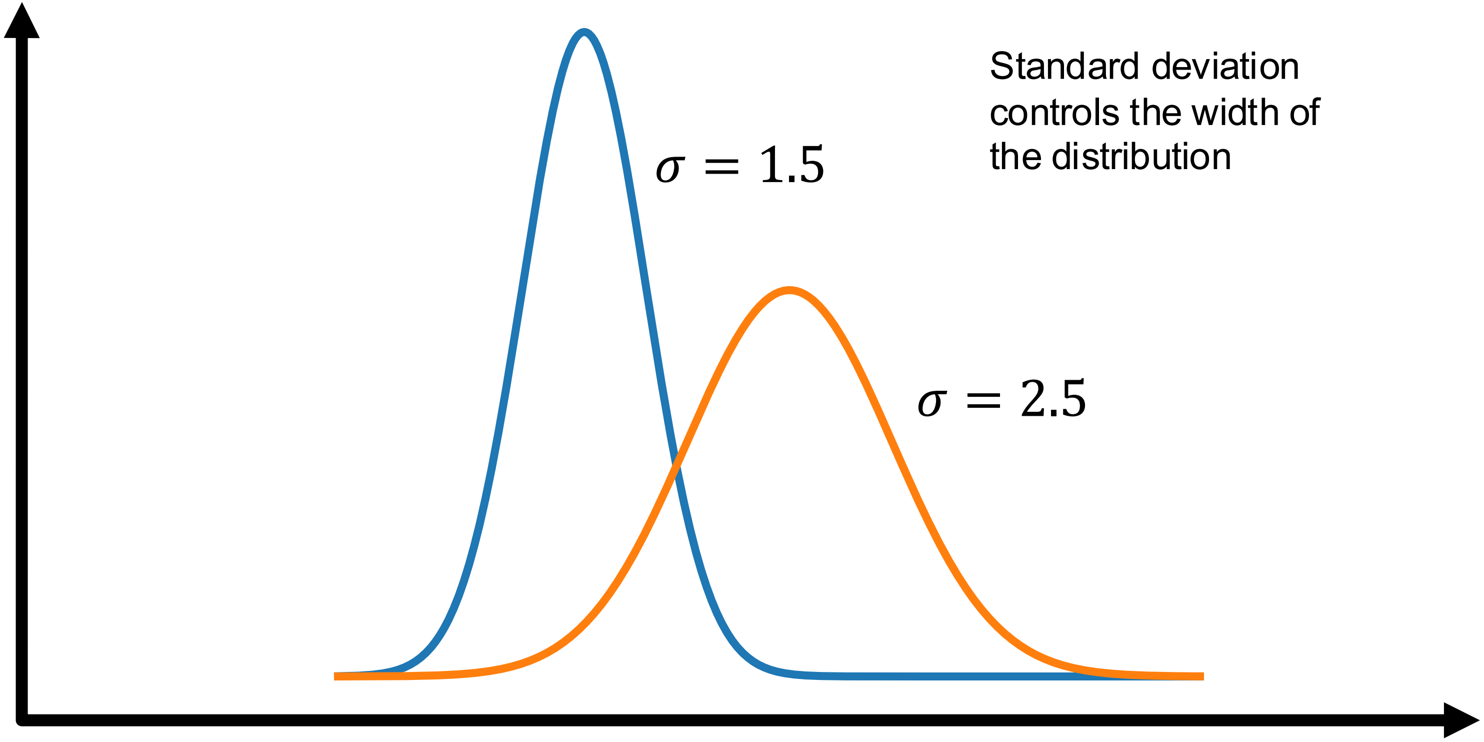
Properties – Mean moves the bell curve



Changing the mean moves the distribution left and right.

The Normal Distribution

Properties – Standard deviation controls spread



Standard deviation controls the width of the distribution

The Normal Distribution

Probability density function (PDF)

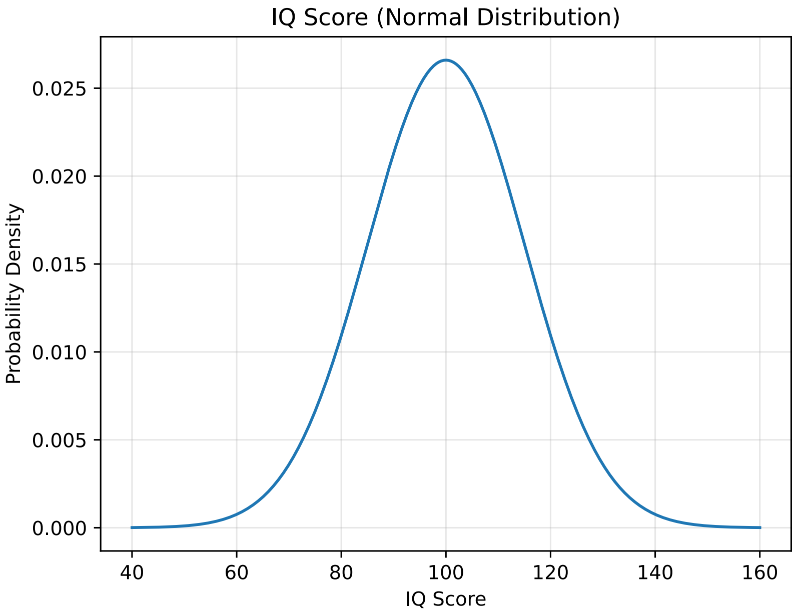
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(\frac{x-\mu}{2\sigma})^2}$$

Probability Density Function (PDF)

“From scratch” implementation

```
import numpy as np
import matplotlib.pyplot as plt
from math import sqrt, exp, pi
from scipy.stats import norm
import os

mu, sigma = 100, 15

def norm_pdf(x, mu, sigma):
    return (1 / (sigma * sqrt(2 * pi))) * exp(-0.5 * ((x - mu) / sigma) ** 2)

x = np.linspace(mu-sigma*4, mu+sigma*4, 1000)

# y = norm.pdf(x, mu, sigma)
# make from scratch
y = [norm_pdf(_x, mu, sigma) for _x in x]

# Plot the distribution
plt.plot(x, y)
plt.title("IQ Score (Normal Distribution)")
plt.xlabel("IQ Score")
plt.ylabel("Probability Density")
plt.grid(True, alpha=0.3)
plt.show()
```

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(\frac{x-\mu}{2\sigma})^2}$$



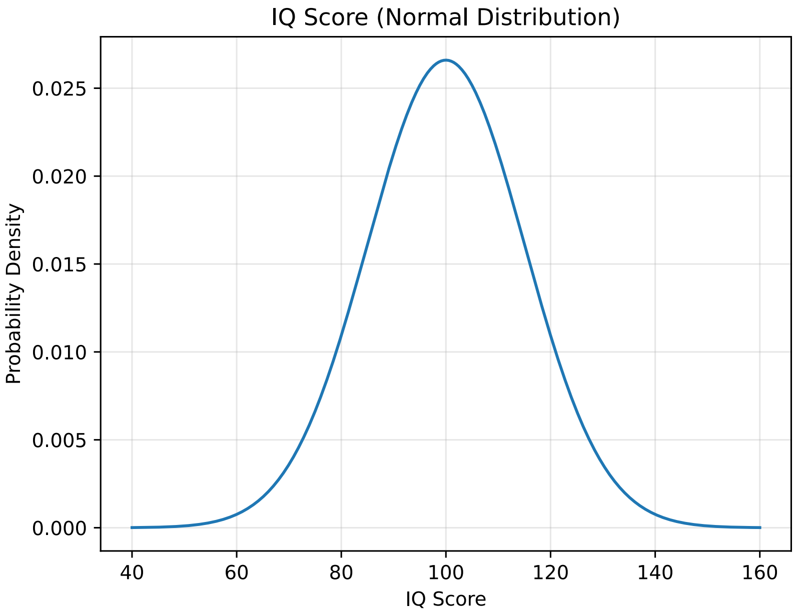
SOURCE: 4_normal_distribution_clt/3_normal_pdf_from_scratch.py

Probability Density Function (PDF)

SciPy implementation

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
import os

mu, sigma = 100, 15

x = np.linspace(mu-sigma*4, mu+sigma*4, 1000)
y = norm.pdf(x, mu, sigma)

# Plot the distribution
plt.plot(x, y)
plt.title("IQ Score (Normal Distribution)")
plt.xlabel("IQ Score")
plt.ylabel("Probability Density")
plt.grid(True, alpha=0.3)
plt.show()
```

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(\frac{x-\mu}{2\sigma})^2}$$



SOURCE: 4_normal_distribution_clt/4_normal_pdf_scipy.py

Cumulative Density Function (CDF)

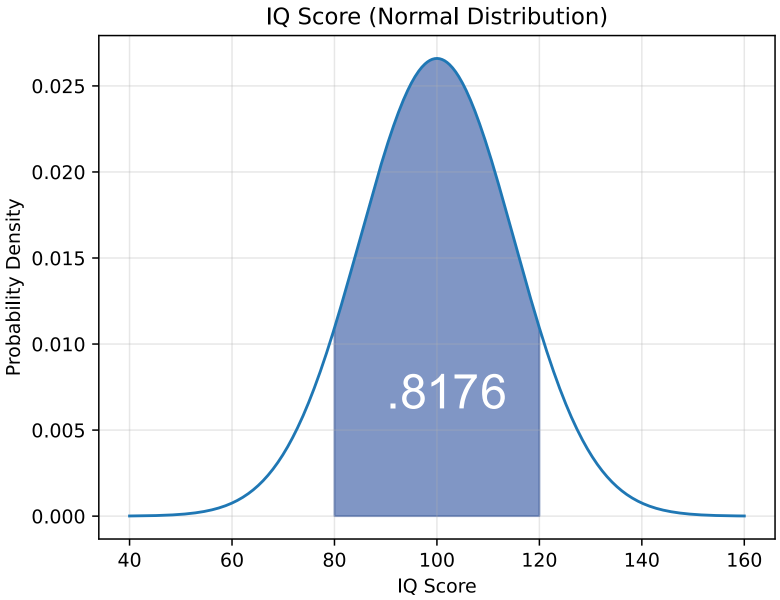
What is the probability?

Let’s say we want to find the probability a randomly selected person has an IQ between 80 and 120.

IQ follows a normal distribution with a mean of 100 and standard deviation of 15.

How do we solve this? We find this area shaded in the center, which will be the probability.

The cumulative density function (CDF) will return areas for us.



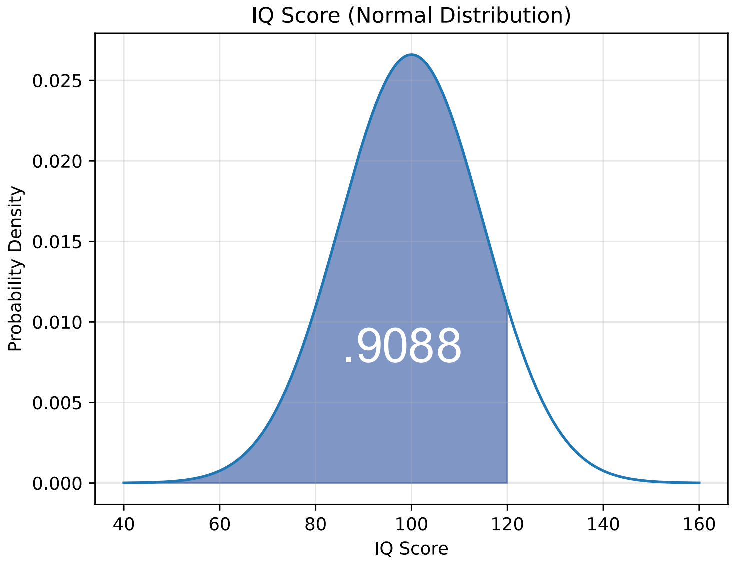
SOURCE: 4_normal_distribution_clt/5_range_80_to_120.py

Cumulative Density Function (CDF)

What is the probability?

Let’s find the area up to 120 using `norm.cdf()`.

```
from scipy.stats import norm

mu, sigma = 100, 15
upper = 120

# AREA LESS THAN 120: 0.9088
print(f"AREA LESS THAN {upper}: {norm.cdf(upper, mu, sigma):.4f}")
```



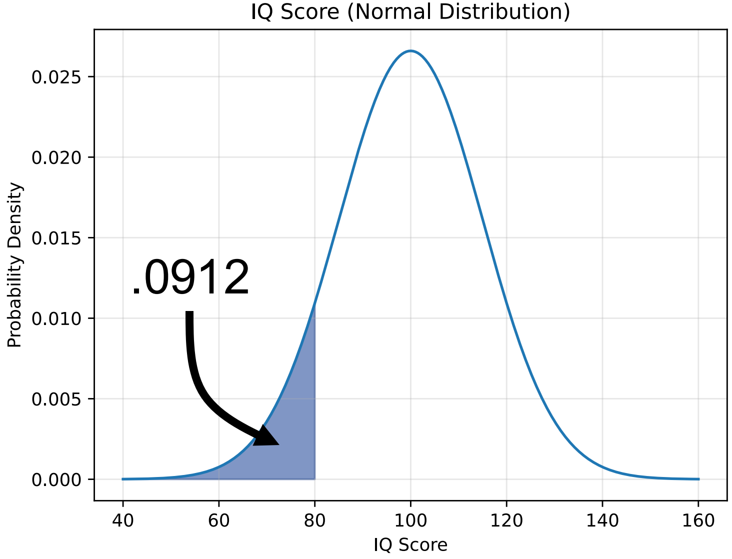
SOURCE: 4_normal_distribution_clt/6_less_than_120.py

Cumulative Density Function (CDF)

What is the probability?

Let’s find the area up to 80 using `norm.cdf()`.

```
from scipy.stats import norm

mu, sigma = 100, 15
upper = 80

# AREA LESS THAN 80: 0.0912
print(f"AREA LESS THAN {upper}: {norm.cdf(upper, mu, sigma):.4f}")
```

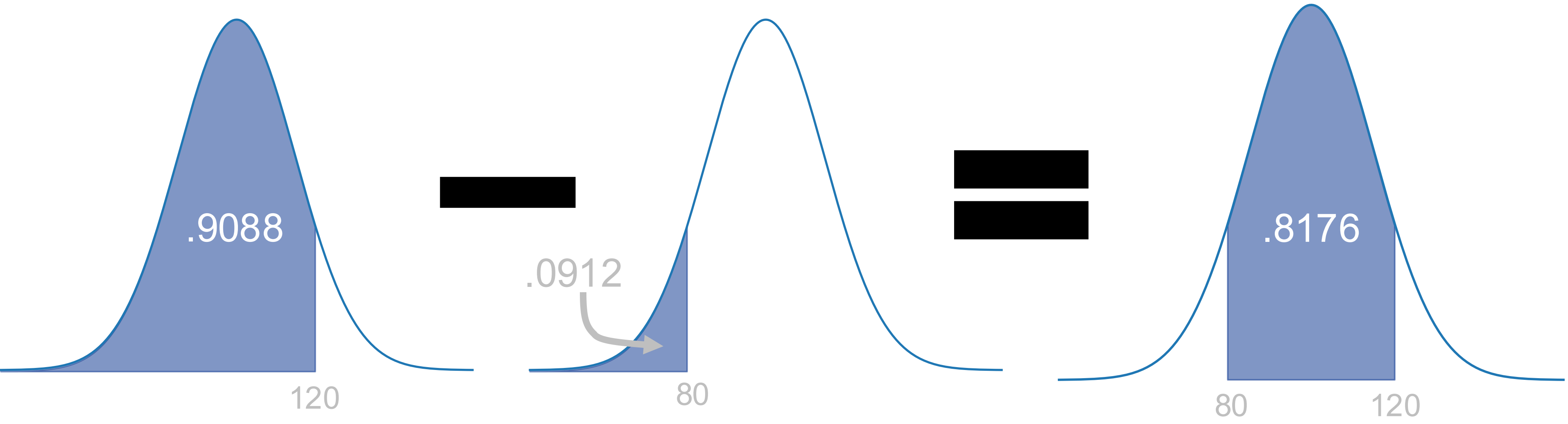


SOURCE: 4_normal_distribution_clt/7_less_than_80.py

Cumulative Density Function (CDF)

The probability of an IQ score between 80 and 120

```
from scipy.stats import norm

mu, sigma = 100, 15

middle_area = norm.cdf(120, mu, sigma) - norm.cdf(80, mu, sigma)

# AREA: 0.8176
print(f"AREA: {middle_area:.4f}")
```

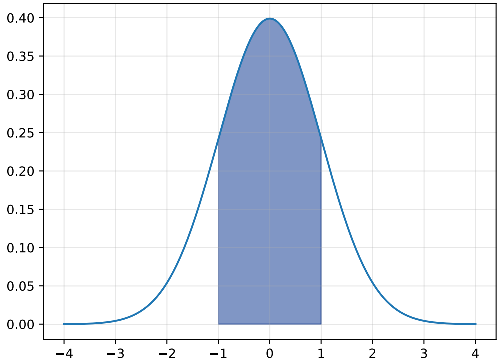


SOURCE: 4_normal_distribution_clt/8_final_middle_range.py

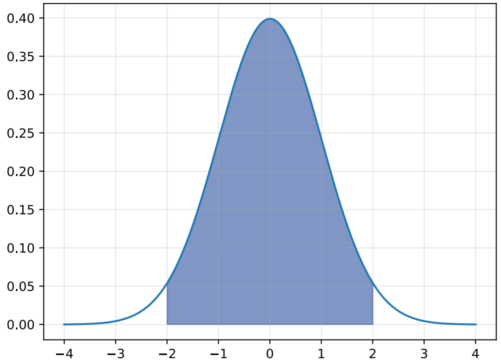
The Empirical Rule

A helpful rule of thumb

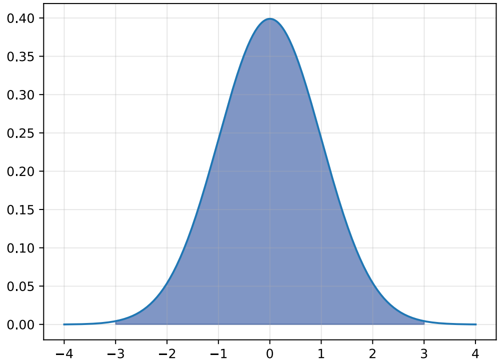
Within 1, 2, and 3 standard deviations of the mean, you capture 68%, 95%, and 99.7% of the area, respectively.



μ ± σ
A = 0.6827



μ ± 2σ
A = 0.9545



μ ± 3σ
A = 0.9973

SOURCE: 4_normal_distribution_clt/9_empirical_rule.py

The Empirical Rule

A helpful rule of thumb

Within 1, 2, and 3 standard deviations of the mean, you capture 68%, 95%, and 99.7% of the area, respectively.

```
from scipy.stats import norm

mu, sigma = 100, 15

for i in range (1,4):
    lower, upper = mu-i*sigma, mu+i*sigma
    area = norm.cdf(upper, mu, sigma) - norm.cdf(lower, mu, sigma)
    print(f"{area*100:.2f}%: of people have an IQ between {lower:.1f} and {upper:.1f} points.")

# 68.27%: of people have an IQ between 85.0 and 115.0 points.
# 95.45%: of people have an IQ between 70.0 and 130.0 points.
# 99.73%: of people have an IQ between 55.0 and 145.0 points.
```

SOURCE: 4_normal_distribution_clt/9_empirical_rule.py

Central Limit Theorem

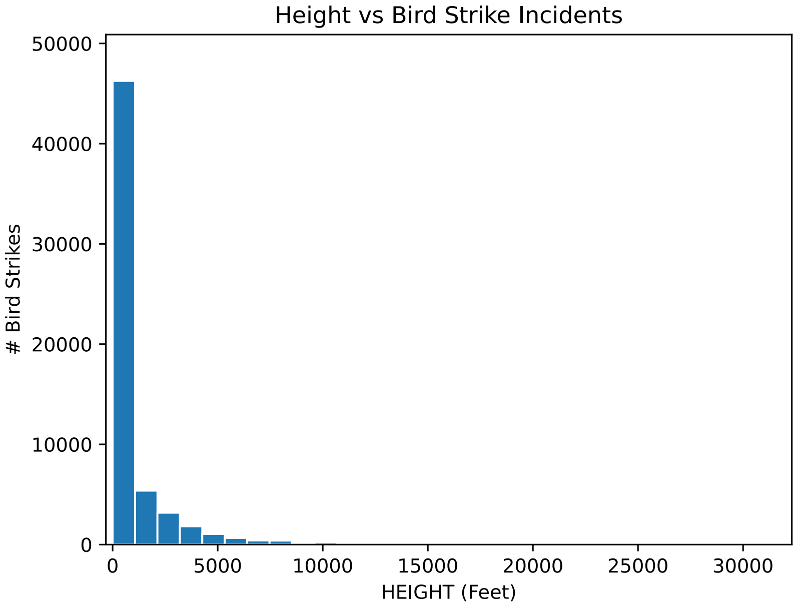
Simulating Sample Means Distribution

Let's take our data of heights where bird strikes occurred.

Recall the data is highly skewed to the right, and this make sense as birds tend to be closer to the ground.

Let's repeatedly take random samples of 60, calculate that sample mean, and plot this **distribution of sample means**.

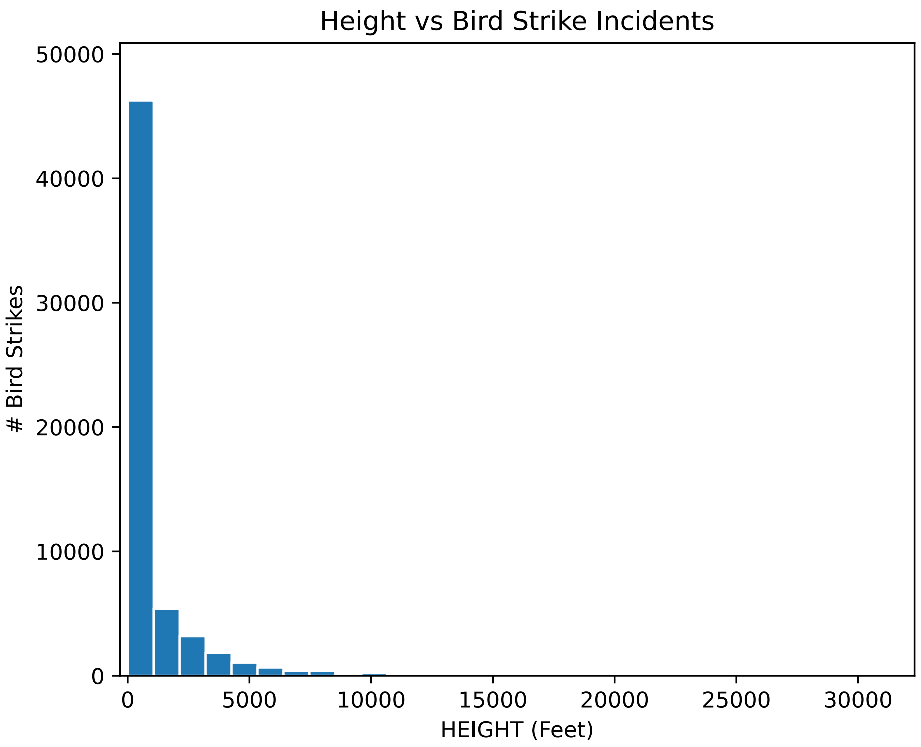
Something remarkable happens.



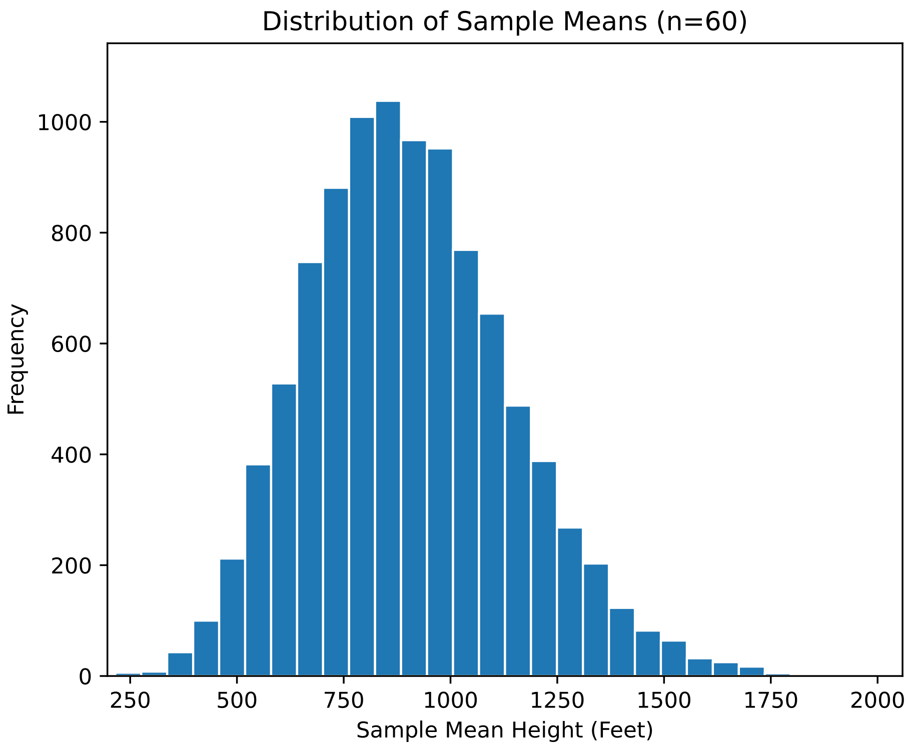
SOURCE: 4_normal_distribution_clt/10_central_limit_theorem.py

Central Limit Theorem

Simulating Sample Means Distribution



Height vs Bird Strike Incidents



Distribution of Sample Means (n=60)

SOURCE: 4_normal_distribution_clt/10_central_limit_theorem.py

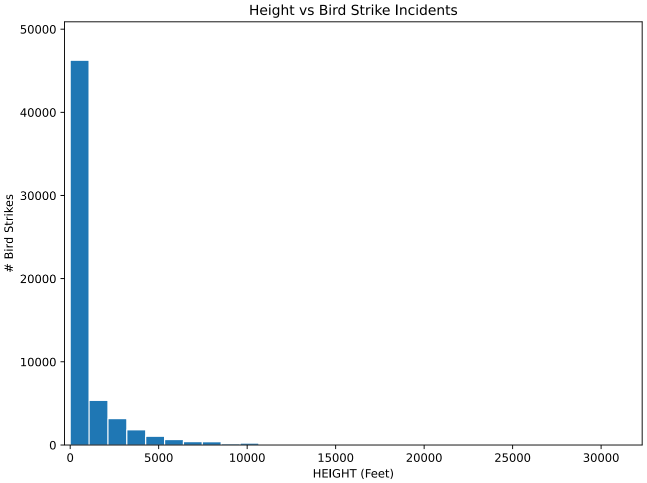
Central Limit Theorem

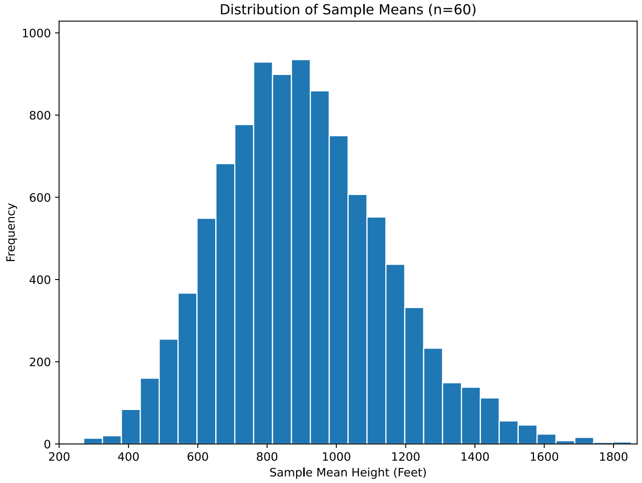
Properties

The **central limit theorem** demonstrates interesting things occur when we take sizable samples and create distributions based on their means.

1. **The mean of the sample means equals the population mean.**

$$\mu_{\bar{x}} = \mu$$



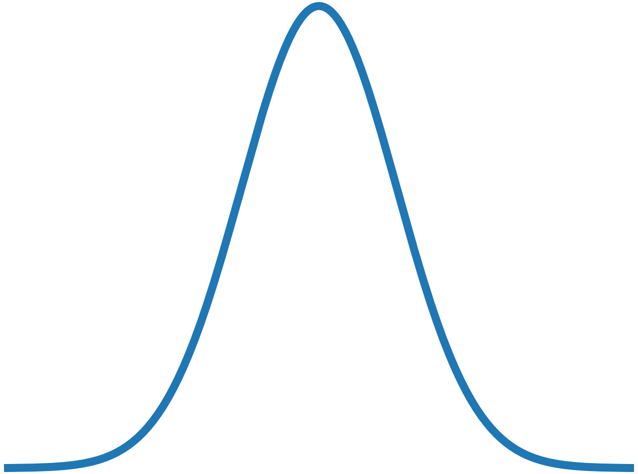


Central Limit Theorem

Properties

2. The standard deviation of the sample means has this relationship to the standard deviation of the population.

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$



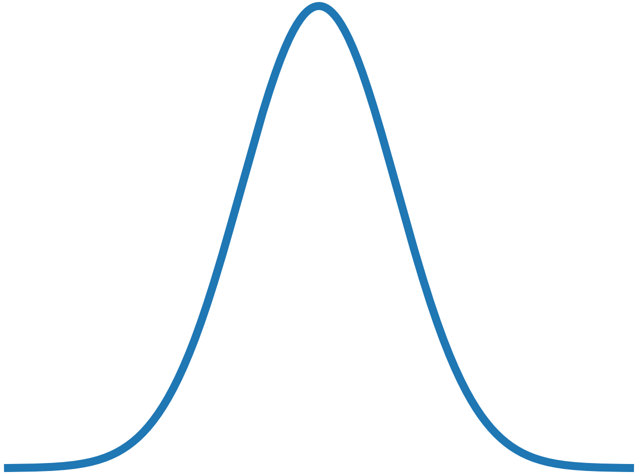
Central Limit Theorem

Properties

3. **If the population is normal, then the sample means will have a normal distribution (regardless of the sample size).**

4. **If the population is not normal, but the sample size is 31 or more, the distribution of the sample means will *usually* approximate a normal distribution.**

Because of the central limit theorem, we can infer a lot from a population, including hypothesis testing.



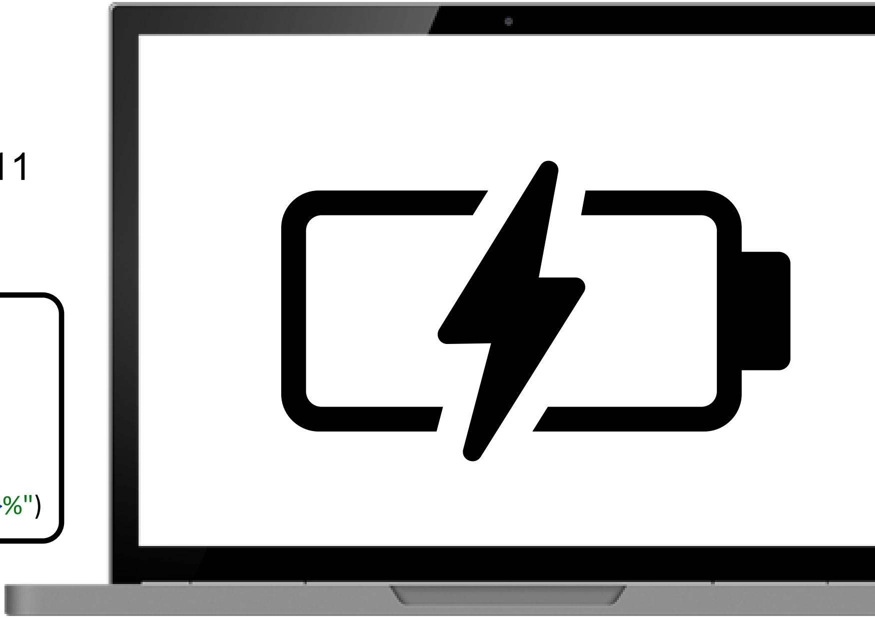
Exercise

Calculating Battery Life

A study suggests that a laptop's battery life is 10 hours with a standard deviation of 1 hour for typical usage and is normally distributed.

Find the probability the laptop would last between 8 and 11 hours.

```
from scipy.stats import norm

mu, sigma = 10, 1
area = norm.cdf(11, mu, sigma) - norm.cdf(8, mu, sigma)

print(f"The probability the battery lasts between 8 and 11 hours is {100*area:.2f}%")
```



SOURCE: 4_normal_distribution_clt/11_exercise_answer.py

Hypothesis Testing

Thomas Nield

Lady Drinking Tea

The invention of the p-value

The best way to introduce hypothesis testing is go back to its invention.

In the 1920’s, Ronald Fisher was at a party with his colleague Muriel Bristol.

As a tea connoisseur, she made a claim: she can detect when tea is poured before the milk.

Ronald made 8 cups of tea on the spot, 4 had milk poured first, the other had tea poured first.

Remarkably, she got them all correct.

Hands-On Time!

Lady Drinking Tea Simulation

Let’s walk through this simulation and do some hands-on analysis on what’s happening here.

5_hypothesis_testing/1_lady_drinking_tea_simulation.py

Lady Drinking Tea

The invention of the p-value

The simulation assumed she was guessing at random, and the probability of getting all 4 cups correct was about ~1.4%.

That 1.4% is what we call the **p-value**, the probability of this outcome given she was guessing.

How do we interpret this? Was she guessing? Does she have a gift? Do we know for sure?

Lady Drinking Tea

The invention of the p-value

We frame experiments as a null hypothesis H_0 and an alternative hypothesis H_A.

$H_0 = \text{“}She\ was\ guessing!\text{”}$
$H_A = \text{“}She\ has\ a\ gift!\text{”}$
$p = .0147$

How do we interpret this? Was she guessing? Does she have a gift? Do we know for sure?

If we set our threshold to .05, that would reject the null hypothesis since the p-value is less.

If we set our threshold to .01, that would tail to reject the null hypothesis since the p-value is more.

Hypothesis Testing

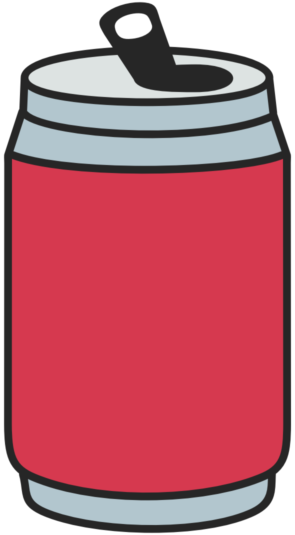
Running an experiment

A **hypothesis** is a premise or claim we want to test.

Let’s say I am wondering if my favorite soda brand is advertising its amount in a can accurately.

It advertises a typical can has 355 ml (H_0, the null hypothesis) but we suspect it is not 355 ml (H_A, the alternative hypothesis).

Here’s the real-world problem in statistics: we often do not have access to an entire population, but we can rely on the central limit theorem to get some insights from a sample.



Hypothesis Testing

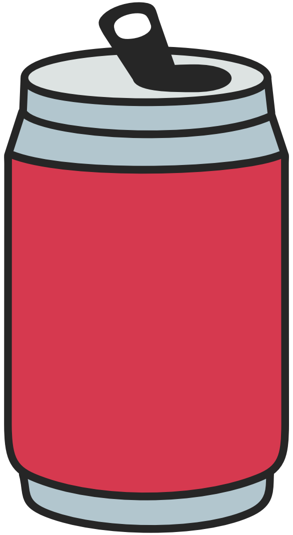
Running an experiment

We declare a **null hypothesis**, which is the status quo or “nothing special happened” claim.

$$H_0 : \mu = 355$$

We then declare the **alternative hypothesis**, which is the claim challenging the status quo.

$$H_A : \mu \neq 355$$



Hypothesis Testing

Running an experiment

We sample 40 cans, measure the liquid of each one on a kitchen scale, and get this sample mean and standard deviation.

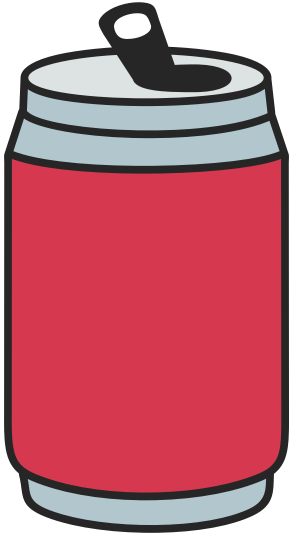
$$\bar{x} = 353$$
$$s = 6$$
$$n = 40$$

Given these parameters, do we reject our null hypothesis and entertain our alternative hypothesis?

$$H_0 : \mu = 355$$
$$H_A : \mu \neq 355$$

We must prove *<u>the sample mean is not just different, but very different from the population mean</u>*.

Because of the $= \ \ and \ \neq$ structure of the hypotheses, this is a **two-tailed test.**



Level of Significance

Defining a threshold

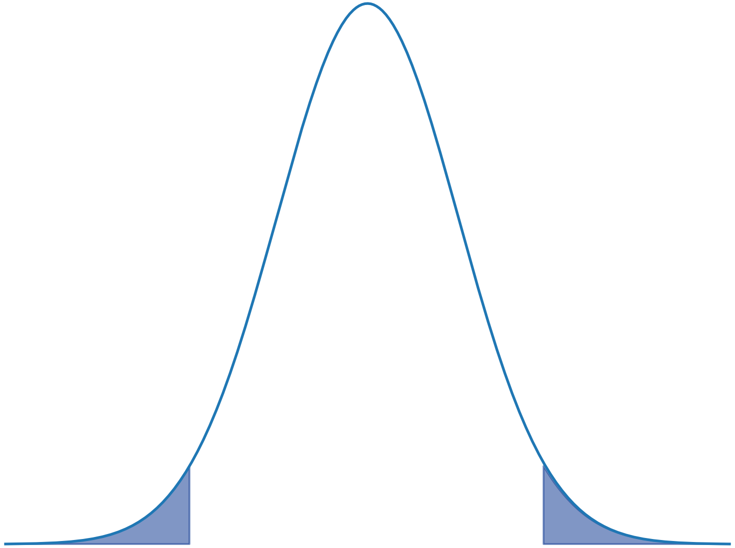
To decide whether to reject our null hypothesis, we need to define a threshold of confidence/significance.

Do we want a 95% **level of confidence C**, which would mean a 5% **level of significance α**?

Do we want a 99% **level of confidence C**, which would mean a 1% **level of significance α**?

This directly affects how confidence we are.

Let's use a 95% level of confidence / 5% level of significance.



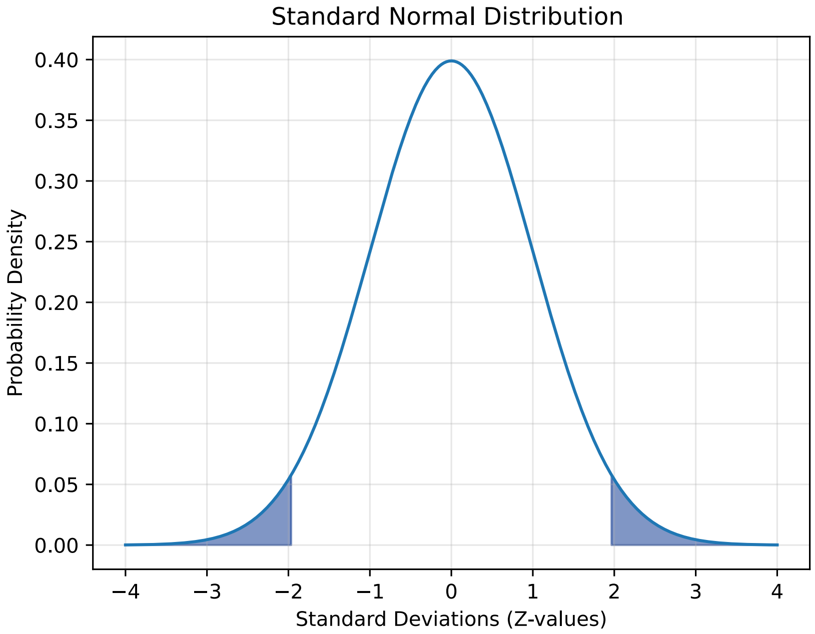
We Need Some New Tools!

The standard normal distribution

The **standard normal distribution** is a normal distribution with a mean of 0 and a standard deviation of 1.

We refer to the x-axis values as ***Z*-scores** or ***Z*-values**.

The standard normal distribution is helpful to standardize problems, including this hypothesis test we are about to do.



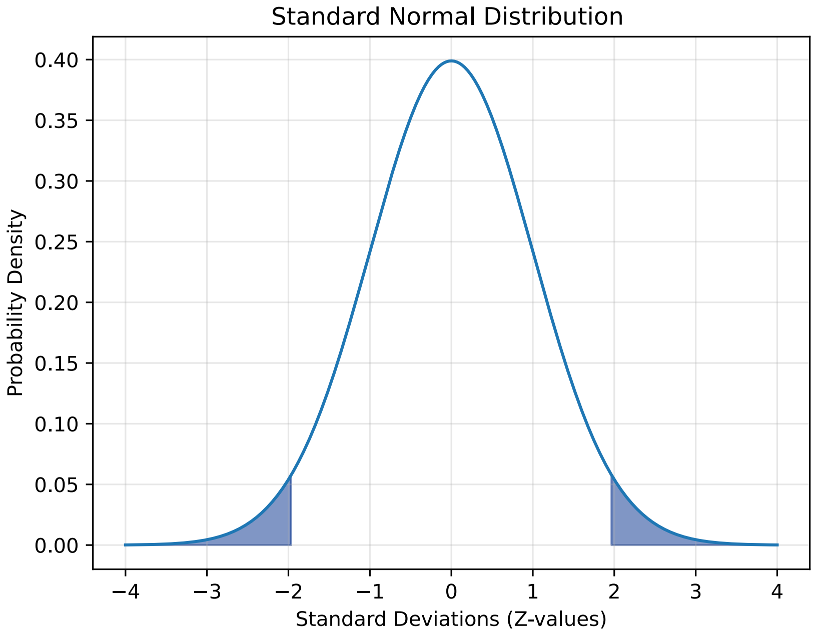
We Need Some New Tools!

The z-score/z-value

We will convert our experiment to a single Z-value using this formula.

$$Z = \frac{\bar{x} - \mu}{\left(\frac{s}{\sqrt{n}}\right)}$$

This effectively converts our test value to the standard normal distribution.



We Need Some New Tools!

The z-score/z-value

$$Z = \frac{\bar{x} - \mu}{\left(\frac{s}{\sqrt{n}}\right)}$$

We find our Z-value to be -2.11 for our soda can experiment.

```python
from math import sqrt

# Given data
mu = 355.0 # Alleged population mean
n = 40  # Sample size
x_bar = 353  # Sample mean
s = 6 # Sample standard deviation

# Calculate z test value
z_stat = (x_bar - mu) / (s / sqrt(n))

print(f"z-statistic: {z_stat:.2f}")
# z-statistic: -2.11
```

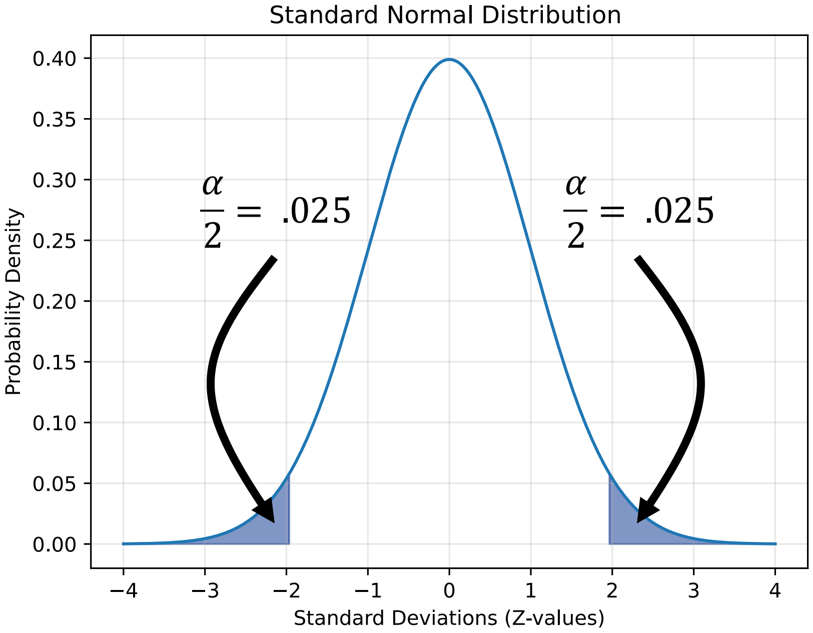
We Need Some New Tools!

Level of significance

The shaded areas to the right reflect a $.05$ level of significance α, where each tail has an area of…

$$\frac{\alpha}{2} = .025$$

The unshaded area in the middle is .95.

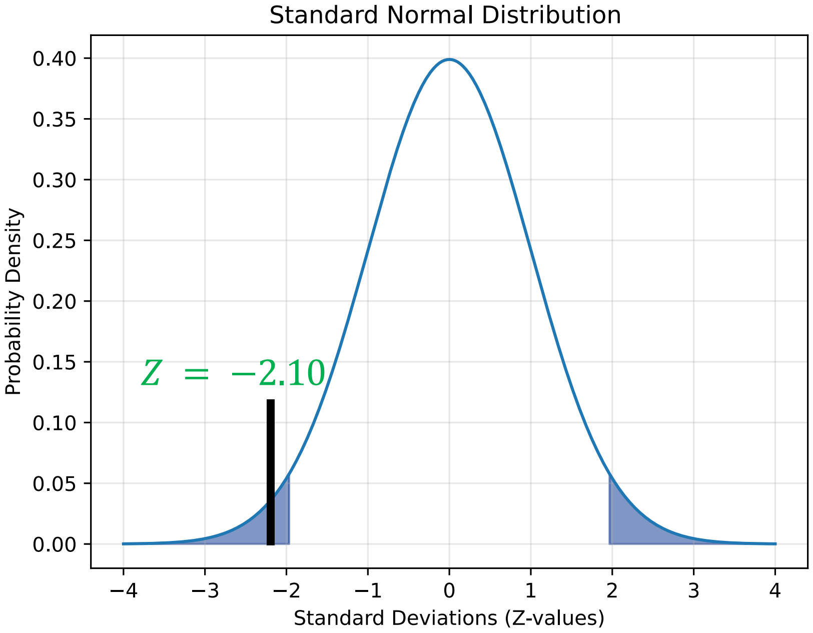


We Need Some New Tools!

Level of significance

If our Z-value lands in either significant region in either tail, **we reject our null hypothesis H_0.**

The outcome is considered unlikely enough that we entertain the alternative hypothesis H_A.



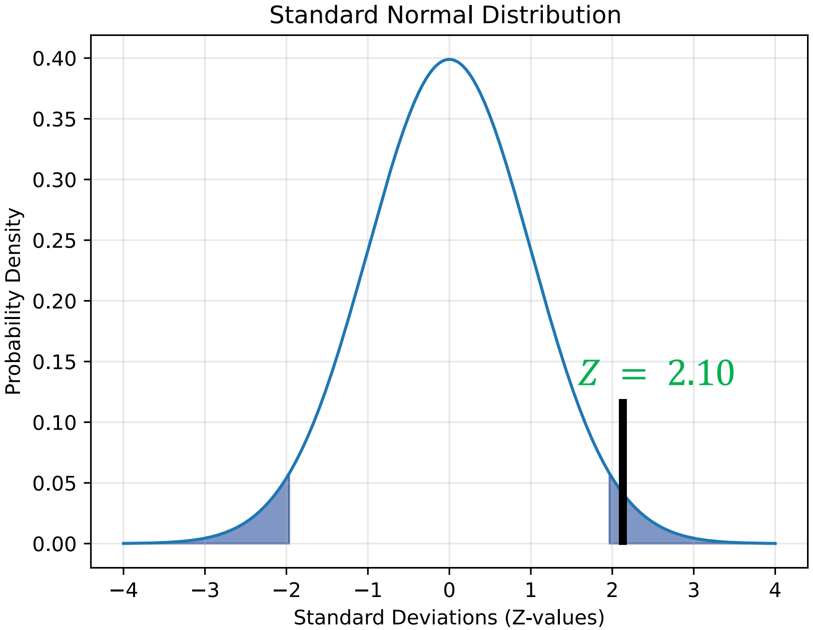
We Need Some New Tools!

Level of significance

If our Z-value lands in either significant region in either tail, **we reject our null hypothesis $\boldsymbol{H_0}$.**

The outcome is considered unlikely enough that we entertain the alternative hypothesis H_A.

This also applies if the Z-value is on the other tail too.

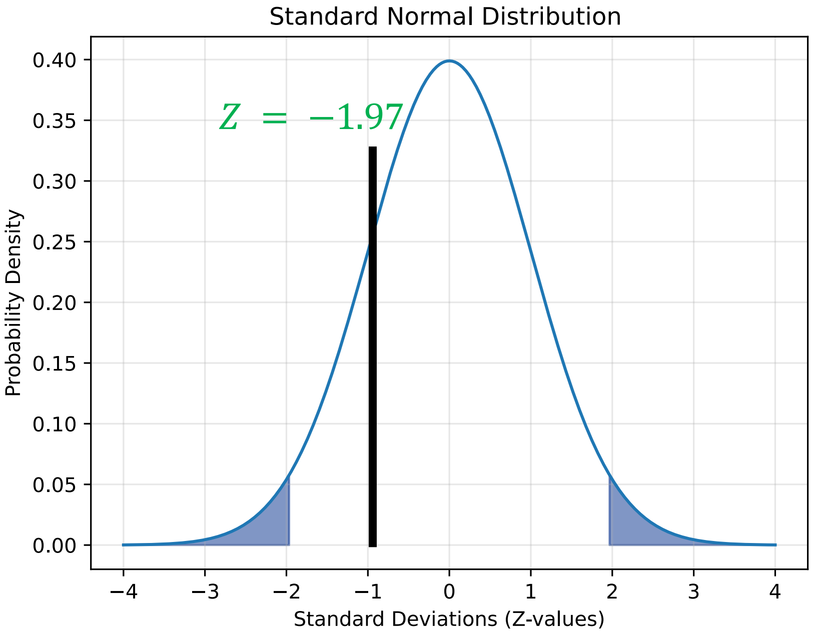


We Need Some New Tools!

Level of significance

However, if our Z-value lands inside the distribution (and not the tails reflecting significant area), **we fail to reject our null hypothesis.**

The outcome is considered too likely that H_0 holds and our sample is not special.

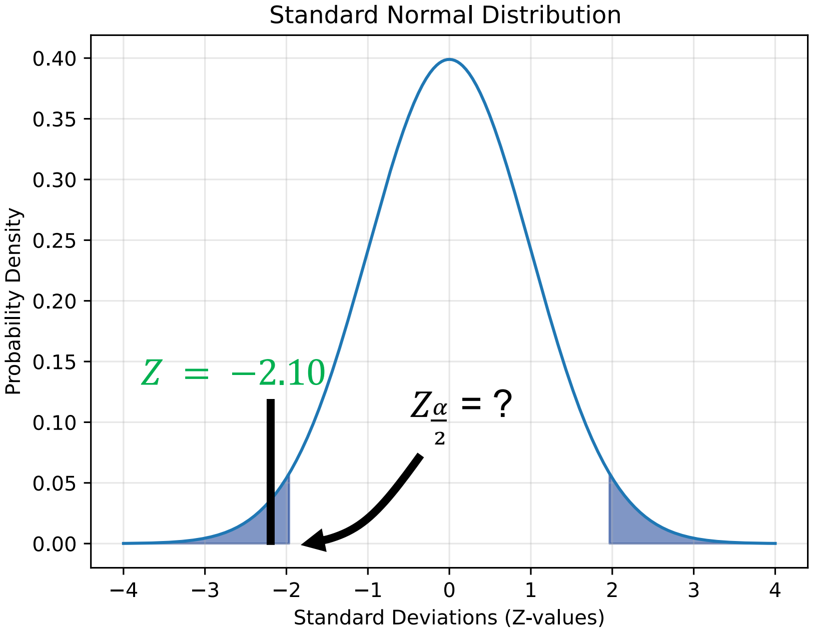


We Need Some New Tools!

Percentile point function (PPF)

So we need to find the Z-value ($Z_{\frac{\alpha}{2}}$) that gives me the area of $\frac{\alpha}{2} = .025$ to its left.

If our test Z is lesser than $Z_{\frac{\alpha}{2}}$, **we reject our null hypothesis.**



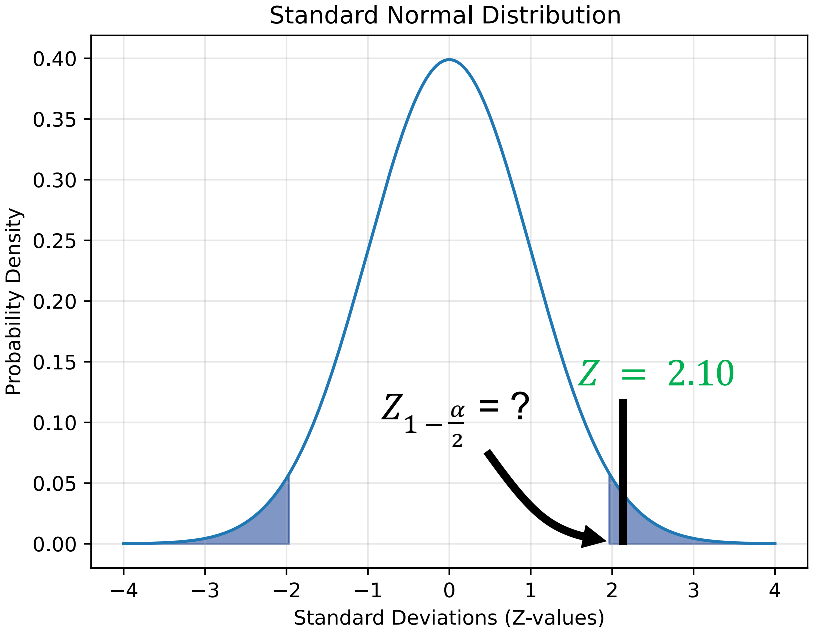
We Need Some New Tools!

Level of significance

So we need to find the Z-value ($Z_{\frac{\alpha}{2}}$) that gives me the area of $\frac{\alpha}{2} = .025$ to its left.

If our test Z is lesser than $Z_{\frac{\alpha}{2}}$, **we reject our null hypothesis.**

This also applies if our Z is greater than $Z_{1-\frac{\alpha}{2}}$, which is opposite of $Z_{\frac{\alpha}{2}}$.



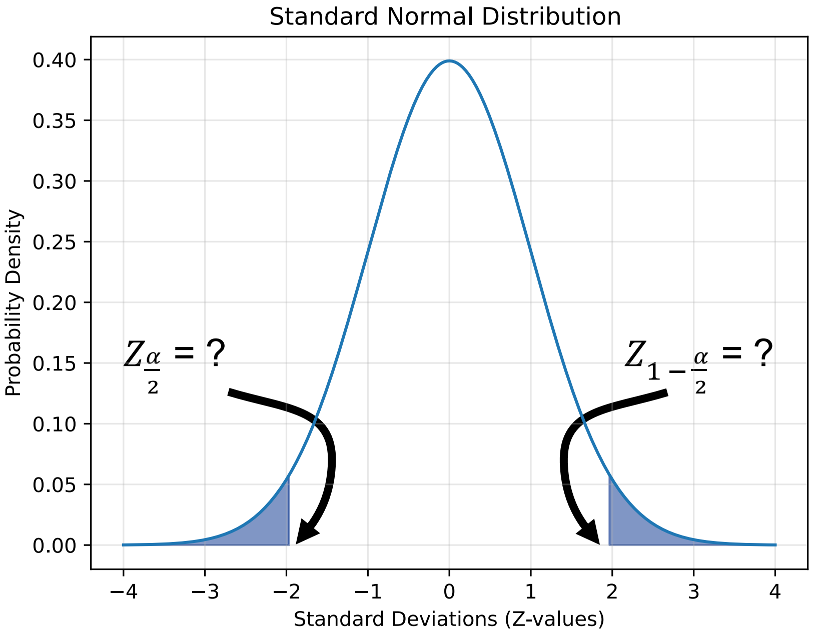
We Need Some New Tools!

Percentile point function (PPF)

So how do we solve for these bounds of the tails $Z_{\frac{\alpha}{2}}$ and $Z_{1-\frac{\alpha}{2}}$?

We need to do a reverse lookup of the CDF, finding the Z values that give .025 and .975 (.025 + .95) areas respectively.

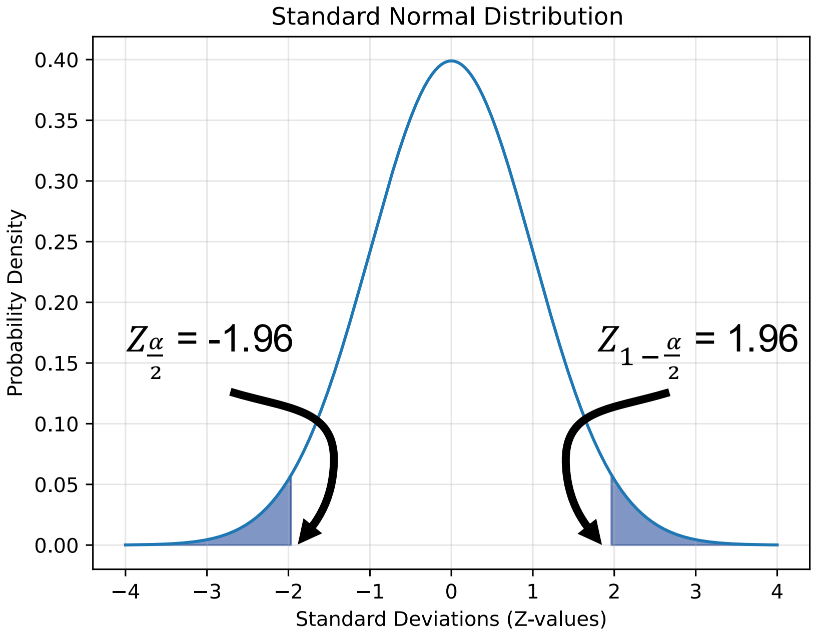
This is what the **percentile point function (PPF)** is for, which returns the value that captures an area to its left.



We Need Some New Tools!

Percentile point function (PPF)

```python
from scipy.stats import norm

alpha = .05

Z_left = norm.ppf(alpha / 2)
Z_right = norm.ppf(1 - (alpha / 2))

print(f"Z_left: {Z_left:.2f}")
print(f"Z_right: {Z_right:.2f}")

# Z_left: -1.96
# Z_right: 1.96
```



SOURCE: 5_hypothesis_testing/4_ppf_lookup.py

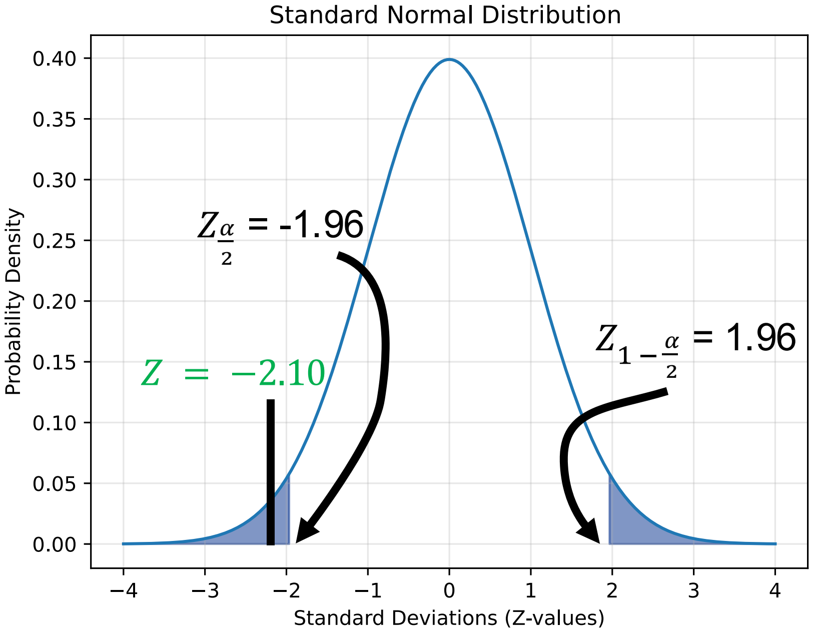
Two-Tailed Test

Final Outcome

Since $Z = -2.10$ is less than $Z_{\frac{\alpha}{2}}$ = -1.96, **we can reject our null hypothesis.**

With 95% confidence, **we reject the null hypothesis that a soda can is 355 ml on average.**

We would have also rejected if Z was greater than 1.96.



Two-Tailed p-value

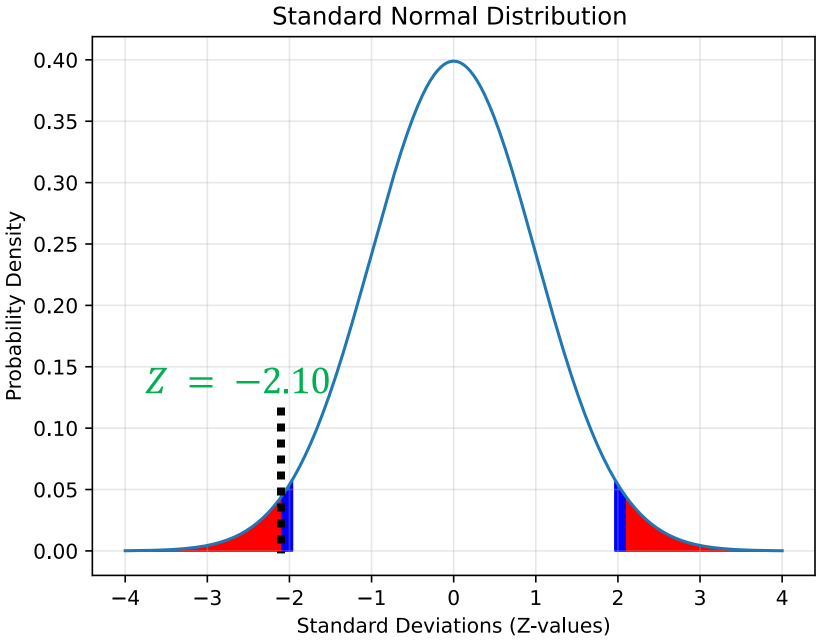
The “coincidence?!” factor

The **p-value** is the probability of getting a sample measure at least as extreme as the sample you just observed, assuming the null hypothesis to be true.

In plain English, it helps answer: **“Was this just a coincidence?”**

Note how it is bound by our test value $\pm Z$, as this captures all values more extreme than $\pm Z$.

That red area is the p-value, and if it is less than our level of significance that will reject H_0.



One-Tailed Testing

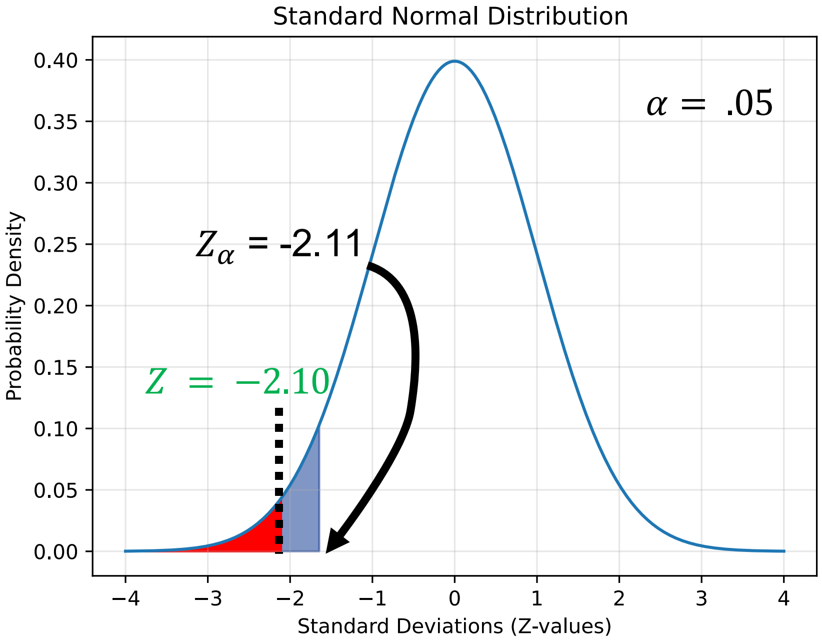
Testing for one direction

While not as robust, we could also test with our hypotheses in a **one-tailed testing** format:

$$H_0 : \mu < 355$$
$$H_A : \mu \geq 355$$

This biases our test in one direction and only works if we *know* the value should be less than the null hypothesis.

Level of significance α is all in one tail, and so is the p-value.

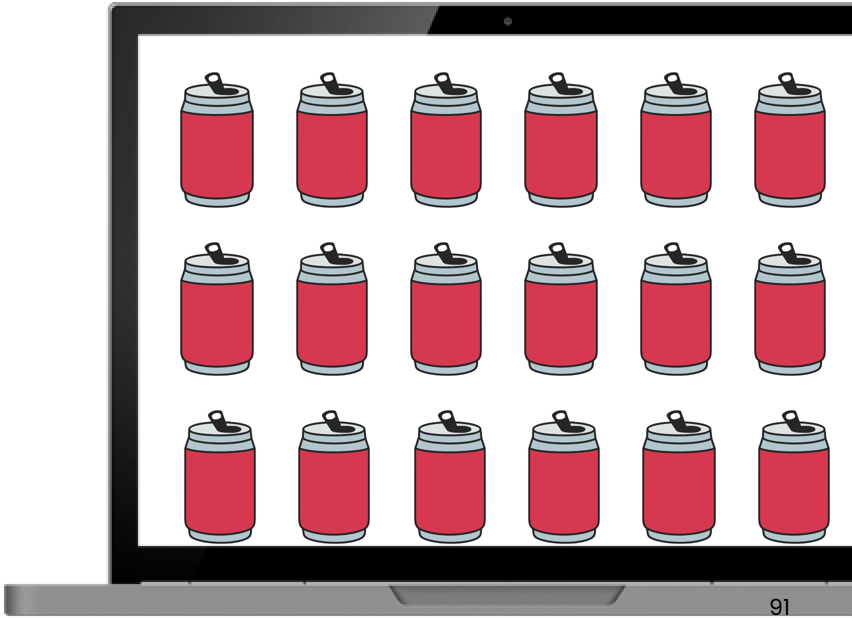


Hands-On Time!

Hypothesis Testing

Let’s walk through two-tailed and one-tailed testing.

5_hypothesis_testing/1_lady_drinking_tea_simulation.py



O'REILLY

Ethics and p-hacking

Thomas Nield

Unbiased Sampling is Hard!

Really, really hard.

Getting an unbiased sample is difficult.

Let’s say you want to study college students on their sleeping habits.

- You sample 13 students with Apple watches at your dorm
- You record the hours slept each night over 2 weeks.

What are the problems with this? *Think!*

Unbiased Sampling is Hard!

Really, really hard.

You sample 13 students with Apple watches at your dorm

You record the hours slept each night over 2 weeks.

There are few problems with this.

- The students in your dorm may not be representative of all students in the country.
- Students with smart watches may be more health-conscious.
- Students with smart watches may be “plugged in” to electronic devices and exposed to more blue light.
- The sample size is very small (more than 30 would be ideal)

Unbiased Sampling is Hard!

Really, really hard.

Ideally, we would randomly sample students across the whole country, not just our dorm.

- Students cannot elect into or out of the study, which would create self-selection bias.
- Because this is impractical, we often have to disclose the possibility of biases.
- We also sometimes combat bias with bias, sampling based on assumed demographic representations.

Unbiased Sampling is Hard!

Really, really hard.

These problems also extend to machine learning and AI.

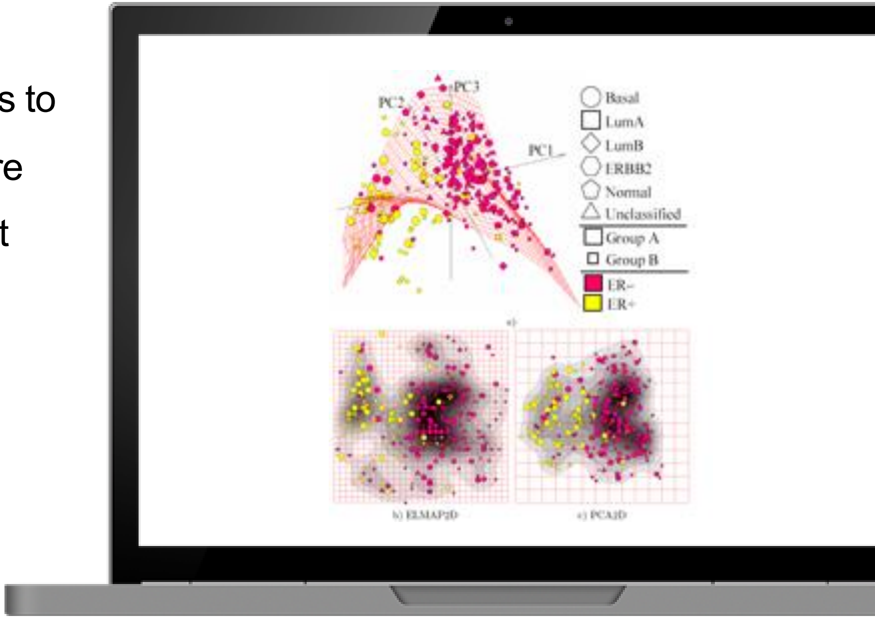
- Self-driving cars are trained in ideal conditions.
- Outliers and wrong assumptions derail the self-driving car in public.
- Large language models struggle to extrapolate outside its training data.

Exercise

Missing the Forest for the Trees

A vendor has approached your military aircraft operation. **They have an AI model that uses data collected from aircraft returning from combat and predicts where lightweight armor needs to be.** It uses statistical models to measure and correlate where bullet holes and damage are likely to be found. It then recommends armoring those hot spots.

What questions do you have for the vendor? Is their data pipeline and methods sound? Why or why not?



SOLUTION: Missing the Forest for the Trees

This is a modernized version of a real data problem back in WWII.

https://apps.dtic.mil/docs/citations/ADA091073

The Center for Naval Analyses conducted a study on mitigating the loss of bombers. After analyzing fleets of bombers returned from missions, they conclude surfaces that statistically show the most damage should be prioritized for more armor.

But a Hungarian mathematician named Abraham Wald pointed out a fatal flaw with this heuristic.

SC

SOURCE: Wikimedia Commons

SOLUTION: Missing the Forest for the Trees

The flaw: the data only captured survived aircraft, and therefore the heuristic was wrong.

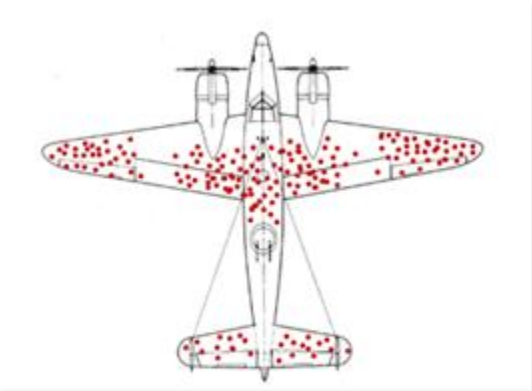
This is an example of **survival bias**, a type of selection bias where we make faulty inferences on the survived population while the deceased population is never accounted for.

While many would cynically say the data is incomplete, the data still provides a valuable clue to solve our objective.

The question we should be asking: why did the aircraft return safely despite the observed damage?

With success, Abraham flipped the theory by armoring the undamaged parts of the aircraft, inferring these were likely the critical areas causing a plane to go down and never returning to base.

This not only saved aircraft and lives but was a pivotal moment for the war effort.

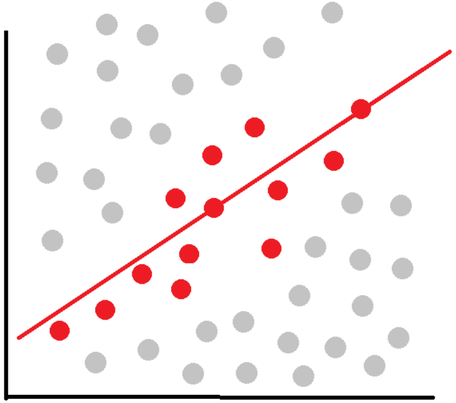


P-hacking

Remaining ethical under pressure

A statistician can be put under pressure.

- “Our client wants to see a model that predicts 10% savings in transportation costs”
- “Our VC investors want a demonstration, so find a dataset that will produce favorable results.”



P-hacking can manifest in many ways:

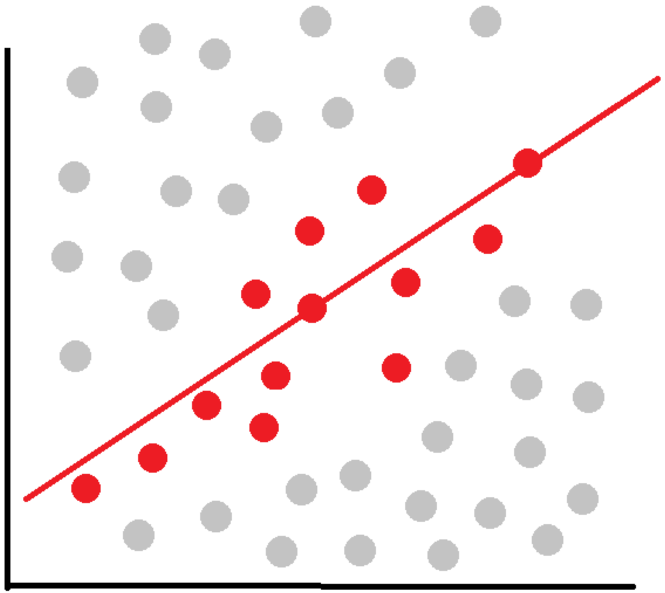
- Collecting just enough data to get a desired result.
- Removing inconvenient data as “outliers” or “noise”
- Shopping for variables that give a desired result
- Dividing data into sub-groups, and focusing on one group
- Shopping model parameters that give the right result
- Using random seeds that produce desired outcomes

P-hacking

Remaining ethical under pressure

“If you torture the data long enough, it will confess to anything.”

— Ronald H. Coase, [Essays on Economics and Economists]



“Look! I found a positive correlation!”

Politically Sensitive Topics

Doing your job *too* well

If you do your job, you may find information that can cause upset.

- Underperforming parts of the organization
- Baffling and unproductive activities
- Liabilities that others overlook
- Misaligned incentives causing problems
- Problems in the organization as a whole

Organizations full of people can do confusing things that make little sense.

Sometimes you should “feel things out” before pointing out concerns publicly.

Politically Sensitive Topics

Doing your job *too* well

“It is difficult to get a man to understand something, when his salary depends on his not understanding it.” – Upton Sinclair

This can be precarious so tread carefully.

- Privately ask questions assuming you’re missing something.
- Read reactions and gauge if this is a taboo subject.
- If you get pushback, a hand slap, or adversity for inexplicable reasons… back off and go to next steps.

Politically Sensitive Topics

Doing your job *too* well

You found a political landmine. Now what?

- If you can “stay in your lane” without ethical or professional concerns, then do so.
- If you sense the matter signals a troubled workplace, then consider leaving.
- If the matter is brazenly unethical or illegal, run for the hills.

Sharing your findings may be more challenging in this kind of environment, and if you stay there are tactics to stay ethical.

Practicing Diplomacy

You can be savvy and ethical

When you know inconvenient truths, you can still work ethically by practicing diplomacy.

- Seek to understand, not judge.
- Focus on the problem, not the people.
- Give the benefit of the doubt.
- Recognize the right time and place to share information.
- Know when something is out of your control.
- Discern an annoyance versus a threat.

Tom’s Rules for Ethical Career Survival

“I’ve seen things, folks.”

1. Share information, not judgements

2. Share hypotheses, not facts

Thank you for coming to my TED Talk!

Rules for Ethical Career Survival

Rule #1: Share information, not judgements

As a statistician, what should you do when you feel political pressures?

- **Share information, not judgements.**
- Dryly share information and do not editorialize it: "*This demonstration was only done in X,Y,Z conditions.*"
- Do not say *"This project is going to fail and should be cancelled"* but rather *"Here are facts A, B, and C about how this project was done."* – let the audience make the conclusions.

This tactic can work remarkably well.

Rules for Ethical Career Survival

Rule #2: Share hypotheses, not facts

Another tactic is to label a dubious claim as a "hypothesis" instead of a fact.

- **"The current hypothesis is that product X will create a 10% savings in transportation cost."**
- **"A current hypothesis is AI will take white collar jobs."**
- Notice how labeling a claim as a *hypothesis* immediately negates it from being a fact?
- You also did not say whose hypothesis it was! (You never said it was *yours*)
- You also insinuated it could change, because it is the *current* hypothesis.

This is helpful especially when marketing is confused as information.