


# Welcome to the **Co**Grammar Counting, Probability and Statistics Lecture

The session will start shortly...

Questions? Drop them in the chat. We'll have dedicated moderators answering questions.



# Coding Interview Workshop Housekeeping

---

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.  
**(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: [Questions](#)

## Coding Interview Workshop Housekeeping cont.

---

- For all **non-academic questions**, please submit a query:  
[www.hyperiondev.com/support](https://www.hyperiondev.com/support)
- Report a **safeguarding** incident:  
[www.hyperiondev.com/safeguardreporting](https://www.hyperiondev.com/safeguardreporting)
- We would love your **feedback** on lectures: [Feedback on Lectures](#)

# Skills Bootcamp

## 8-Week Progression Overview

### Fulfil 4 Criteria to Graduation

#### ✓ Criterion 1: Initial Requirements

Timeframe: First 2 Weeks

Guided Learning Hours (GLH):

Minimum of 15 hours

Task Completion: First four tasks

**Due Date: 24 March 2024**

#### ✓ Criterion 2: Mid-Course Progress


**60** Guided Learning Hours

Data Science - **13 tasks**


Software Engineering - **13 tasks**

Web Development - **13 tasks**

**Due Date: 28 April 2024**



# Skills Bootcamp Progression Overview




## ✓ Criterion 3: Course Progress

Completion: All mandatory tasks,  
including Build Your Brand and  
resubmissions by study period end  
Interview Invitation: Within 4 weeks  
post-course  
Guided Learning Hours: Minimum of  
112 hours by support end date  
(10.5 hours average, each week)

## ✓ Criterion 4: Demonstrating Employability

Final Job or Apprenticeship  
Outcome: Document within 12  
weeks post-graduation  
Relevance: Progression to  
employment or related  
opportunity





**SKILLS  
FOR LIFE**

**SKILLS BOOTCAMPS**



Department  
for Education

# CoGrammar

## Counting, Probability and Statistics

March 2024



# Portfolio Assignment Reviews

[Submit your solutions here!](#)



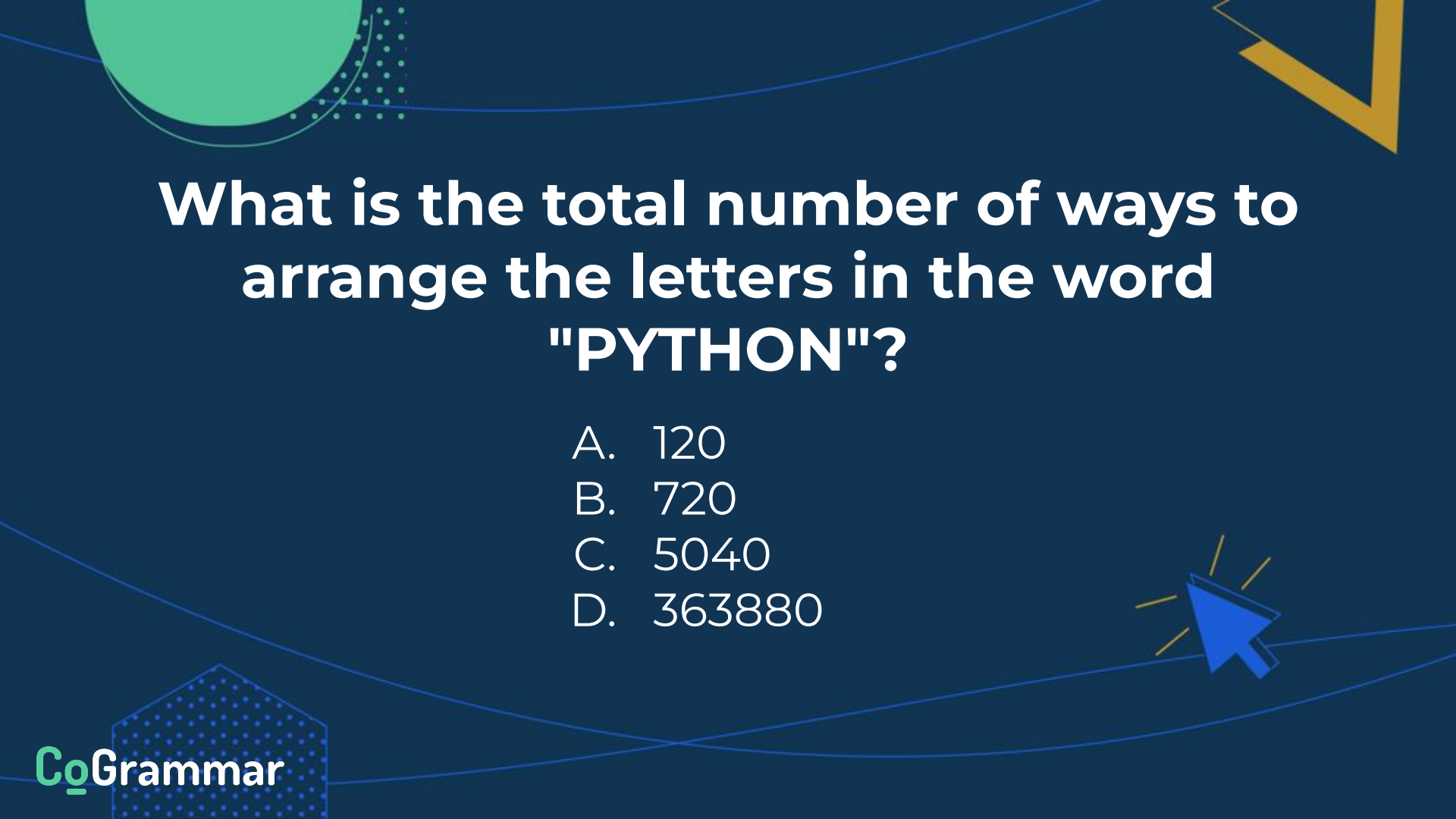
# Learning Objectives

- ❖ Apply **basic counting principles** and calculate **permutations** and **combinations** for solving practical problems.
- ❖ Utilise **fundamental probability concepts** and **Bayes' theorem** to model and solve real-world scenarios.
- ❖ Compute and interpret **descriptive statistics** to analyse and summarise data sets.



# Learning Objectives

- ❖ Identify and apply appropriate **distributions** and understand **sampling techniques** for data modelling.
- ❖ Introduce basic **inferential statistics**, including **hypothesis testing** and calculating **confidence intervals** to make data-driven decisions.



**What is the total number of ways to  
arrange the letters in the word  
"PYTHON"?**

- A. 120
- B. 720
- C. 5040
- D. 363880

## Correct Answer: B

- ❖ We use the following formula to calculate the number of ways to **arrange** a set of items:

$$P(n, r) = \frac{n!}{(n-r)!}$$

Where n is the number of items we have to **choose** from and r is the number of items we want to **arrange**. P is the number of **permutations**.

- ❖ In our case,  $n = r$ :

$$P(n, r) = n!$$
$$P(6, 6) = 6! = 720$$

## Correct Answer: B

- ❖ This should not be confused with the number of way to **choose** or **select** a set of items.
- ❖ This is known as the number of **combinations**.
- ❖ In this case, the order does not matter i.e. if you select 3 items 1, 2, 3 and then select 3 items 2, 3, 1; this would be considered to be the **same** selection.
- ❖ When order doesn't matter, we have less options compared to when order does matter.



**In a standard deck of playing cards,  
what is the probability of drawing a  
heart or a king?**

- A.  $\frac{1}{2}$
- B.  $\frac{1}{4}$
- C.  $\frac{1}{13}$
- D.  $\frac{4}{13}$

## Correct Answer: D

- ❖ There are **52 cards** in a standard playing card deck.
- ❖ The cards are divided into **4 suites** with **13 cards** in each.
- ❖ The 13 cards consist of numbered cards from 1-10, a jack, a queen and a king.
- ❖ The **probability of an outcome** is:

$$P(E) = \frac{\text{number of favourable outcomes}}{\text{total number of outcomes}}$$



## Correct Answer: D

- ❖ When we want to combine probabilities and determine the likelihood of Event A **or** Event B, we use:

$$P(A \text{ or } B) = P(A) + P(B)$$

- ❖ Thus to calculate the probability of drawing a heart or a king:

$$P(\text{heart or king}) = \frac{13}{52} + \frac{4}{52} - \frac{1}{52} = \frac{16}{52} = \frac{4}{13}$$

- ❖ Based on this answer, how many cards would you expect to draw before you draw a heart or king?



**What is the mean of the following set of numbers: [2, 4, 6, 8, 10]?**

- A. 4
- B. 6
- C. 8
- D. 10

## Correct Answer: B

- ❖ The **mean** of a set of numbers is also known as the **average**.
- ❖ The mean gives us an indication of the **central tendency** of a distribution.
- ❖ The other common measures of central tendency are: **mode** and **median**.

## Detour: What is a factorial?

- ❖ A factorial, denoted by an exclamation point (!), is **the product of all positive integers up to a given number**. So  $n! = n(n-1)!$

$$= n(n-1)(n-2)!$$

$$= n(n-1)(n-2)\dots(n-n)!$$

Where  $(n-n)! = 0! = 1$ .

- ❖ For example  $3! = 3 \times 2 \times 1 \times 0! = 3 \times 2 = 6$ . Usually  $0!$  is simply omitted.

## Detour: What is a factorial?

```
#Factorial
num = 5
fact = 1
for i in range(1, num+1):
    fact = fact * i

print("The factorial of "+str(num)+" is: ", fact)
#Output: The factorial of 5 is: 120
```

```
//Factorial
let num = 5;
let fact = 1;
for (let i = 1; i <= num; i++) {
    fact = fact * i;
}
console.log("The factorial of " + num + " is: ", fact);
//Output: The factorial of 5 is: 120
```

```
#Using math
import math
print("The factorial of "+str(num)+" is: ", math.factorial(num))
#Output: The factorial of 5 is: 120
```

# Permutations

**Arrangement of objects where order is important.**

- ❖ To calculate permutations, we use the following formula

$$P(n, r) = \frac{n!}{(n - r)!}$$

$n$  is the number of items to choose from  
 $r$  is the number of items chosen to be arranged

- ❖ For example, if we have 5 books and we want to arrange 3 on a bookshelf, we would use the following calculation to calculate the number of possible arrangements:

$$P(5, 3) = \frac{5!}{(5 - 3)!} = \frac{120}{2} = 60$$



# Permutations Application

In Software Engineering and Web Development, permutations are used in algorithms that require generating all possible arrangements of a set of items.

For example, permutations can be used to generate all possible combinations of a password.

Let's have a look at this in Python :)

# Combinations

**Selection where order doesn't matter.**

- ❖ To calculate combinations, we use the following formula

$$C(n, r) = \frac{n!}{r! (n - r)!}$$

$n$  is the number of items to choose from  
 $r$  is the number of items chosen to be chosen

- ❖ For example, if we have 10 flowers and we want to choose 3 for a bouquet, we would use the following calculation to calculate the number of possible bouquets:

$$C(10, 3) = \frac{10!}{3! (10 - 3)!} = \frac{10 \times 9 \times 8}{6} = 120$$

# Combinations Application

In Data Science, combinations are used in feature selection algorithms to choose the best subset of features for a machine learning model.

For example, when building a recommendation system, combinations can be used to select the most relevant features for predicting user preferences.

Let's have a look at this in Python :)

# Applications in JS and Python

- ❖ We can use the built-in Math library in Python and in JavaScript, the functions need to be created from scratch or they can be loaded from an external library like [mathjs](#) or [simple-statistics](#).

```
import math

# Calculate permutations of 3 out of 5 items
total_permutations = math.perm(5, 3)
print("Total permutations:", total_permutations)

# Calculate combinations of 3 out of 5 items
total_combinations = math.comb(5, 3)
print("Total combinations:", total_combinations)
```

```
// The number of ways to arrange items
function permutations(n, r) {
  |   return factorial(n) / factorial(n - r);
}

// The number of ways to select items
function combinations(n, r) {
  |   return factorial(n) / (factorial(r) * factorial(n - r));
}
```

# Probability

- ❖ **Sample Space:** The set of all possible outcomes.
- ❖ **Events:** Specific outcomes or sets of outcomes from the sample space.

For example, if  $\{2.2, 2.6, 2.8, 2.9\}$  is the **sample space**, then  $\{2.2, 2.6\}$  is one of the **events**.

In the case of flipping one coin,  $\{\text{Heads}, \text{Tails}\}$  is the **sample space** and  $\{\text{Heads}\}$  is one **event**.

# Probability

**The likelihood of the occurrence of a specific event in a sample space.**

- ❖ To calculate the probability of an event (E), we use the following formula

$$P(E) = \frac{\text{number of favourable outcomes}}{\text{total number of outcomes}}$$

- ❖ The total number of outcomes is the same as the number of outcomes in the **sample space**.
- ❖ We can use this definition to calculate the **probability of certain permutations or combinations**.



# Counting Probabilities

- ❖ We can calculate the **probability of a specific combination or permutation** using the probability formula.

```
import math

# Probability of a specific combination
comboProb = 1/(math.comb(10, 3))
print(comboProb)

# Probability of a specific permutation
permProb = 1/(math.perm(10, 3))
print(permProb)
```

```
var math = require('mathjs');

// Probability of a specific combination
let comboProb = 1/(math.combinations(10, 3));
console.log(comboProb);

// Probability of a specific permutation
let permProb = 1/(math.permutations(10, 3));
console.log(permProb);
```

# Probability Rules

- ❖ **Addition Rule:** For mutually exclusive events A and B,

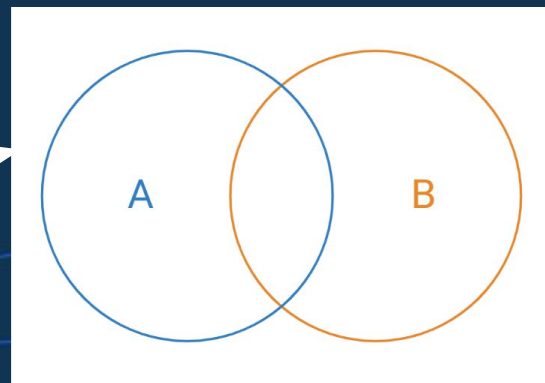
$$P(A \text{ or } B) = P(A) + P(B)$$

This cannot exceed 1.

- ❖ **Multiplication Rule:** For independent events A and B,

$$P(A \text{ and } B) = P(A) \times P(B)$$

This is a Venn Diagram,  
we can use it to  
visualise these rules :)



# Conditional Probability

- ❖ **Conditional Probability:** This is used to calculate the probability of an event A, given that another event B has already happened:

$$P(A|B) = \frac{P(A \text{ and } B)}{P(B)}$$

- ❖ **Independence:** Events A and B are independent if their occurrence do not influence each other i.e.

$$P(A|B) = P(A) \quad \text{and} \quad P(B|A) = P(B)$$

# Bayes Theorem

- ❖ **Bayes Theorem:** A formula for conditional probability which provides a way to revise existing predictions or theories (update probabilities) given new or additional evidence.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

- ❖ **Applications:**

- Calculation of Financial Risk
- Accuracy of Medical Tests
- Bayesian Inference and Statistics
- Machine Learning and Artificial Intelligence
- Weather Forecasting

# Bayes Theorem Example

- ❖ You are planning a barbeque (braai) today, but the morning is cloudy
  - Oh no! 50% of all rainy days start off cloudy! ->  $P(\text{Cloud} | \text{Rain})$
  - But cloudy mornings are common (about 40% of days start cloudy) ->  $P(\text{Cloud})$
  - And this is usually a dry month (only 3 of 30 days tend to be rainy, or  $P(\text{Rain}) = 10\%$ )

What is the chance of rain during the day?

$$P(\text{Rain}|\text{Cloud}) = \frac{P(\text{Rain}) P(\text{Cloud}|\text{Rain})}{P(\text{Cloud})} = \frac{0.1 \times 0.5}{0.4} = .125$$

Or a **12.5%** chance of rain. Not too bad, let's have a braai!



# Let's Breathe!

Let's take a small break  
before moving on to  
the next topic.



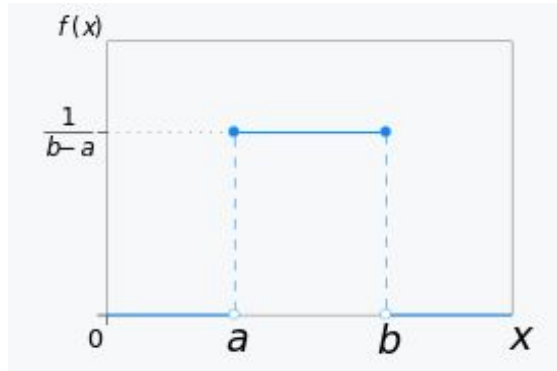


# Distributions

- ❖ **Random variables:** Numerical description of the outcome of an experiment.
- ❖ So far we have looked at experiments with a **finite** number of outcomes. These are known as **discrete random variables**.
- ❖ A random variable that can take on **any value** on an **interval** of a **real number line** is called a **continuous random variable**.
- ❖ **Probability Distributions:** Describes how the probabilities are distributed over the values of the random variable.
  - In the case of continuous random variables, this is a function  $f(x)$ .
  - This is a fun tool to help visualise distributions: [Probability distribution explorer](#).

# Uniform Distribution

- ❖ In a uniform distribution all outcomes are equally likely.

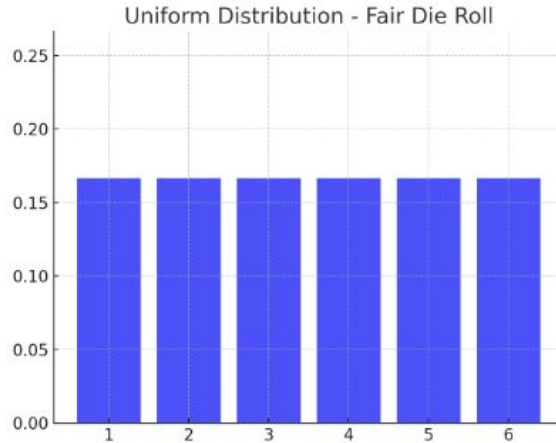


$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b, \\ 0 & \text{for } x < a \text{ or } x > b. \end{cases}$$

**Source:** [https://en.wikipedia.org/wiki/Continuous\\_uniform\\_distribution](https://en.wikipedia.org/wiki/Continuous_uniform_distribution)

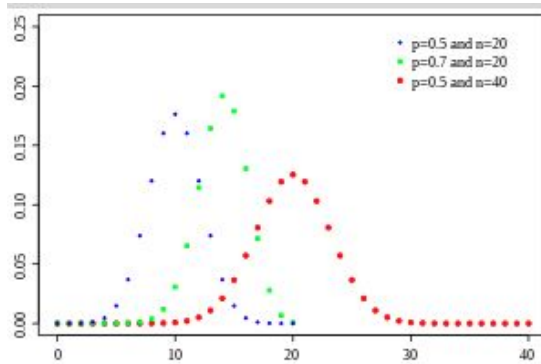
# Uniform Distribution

- ❖ An example is a fair 6-sided die, which has  $P(x) = \frac{1}{6}$  for all sides.



# Binomial Distribution

- ❖ Number of successes in a fixed number of trials



$$f(k, n, p) = \Pr(k; n, p) = \Pr(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

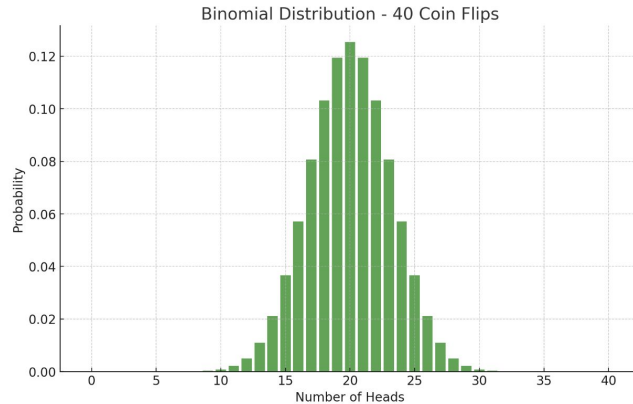
for  $k = 0, 1, 2, \dots, n$ , where

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Source: [https://en.wikipedia.org/wiki/Binomial\\_distribution](https://en.wikipedia.org/wiki/Binomial_distribution)

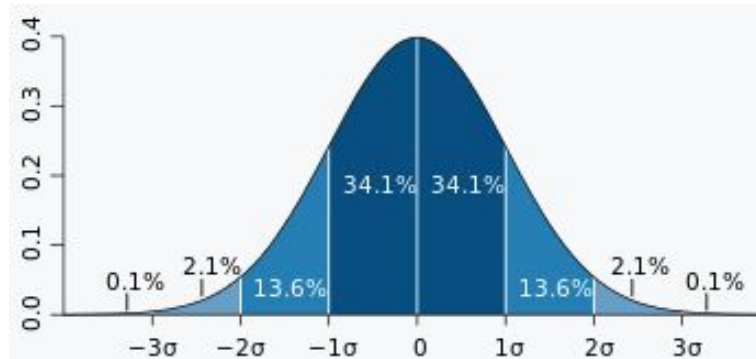
# Binomial Distribution

- ❖ To get the probability of getting 20 heads in a coin toss when doing 40 trials, substitute in  $p=\frac{1}{2}$  ,  $n=40$ ,  $k=20$ , to get  $P(40,20,\frac{1}{2}) = 0.125$



# Normal Distribution

- ❖ Describes data in clusters around a mean. It is the most common distribution in statistics since it tends to represent natural phenomena more accurately than most other distributions most of the time.



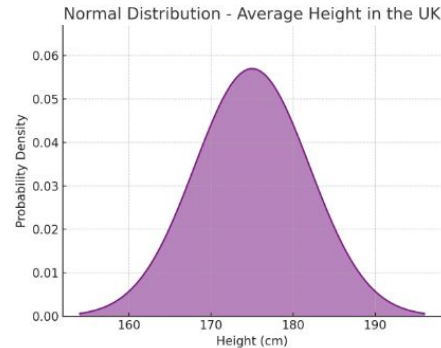
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

**Source:**

[https://en.wikipedia.org/wiki/Normal\\_distribution](https://en.wikipedia.org/wiki/Normal_distribution)

# Normal Distribution

- ❖ An example is the height of people. The probability of a male in the UK being between 168 cm (one standard deviation below the mean) and 182 cm (one standard deviation above the mean) is approximately 0.683.
- ❖ We get this by calculating the area underneath the curve with  $P(182) - P(168)$  where the mean is 175 cm and the standard deviation is 7 cm.





# Descriptive Statistics

A set of methods for organising, summarising and presenting data.

- ❖ These measures can help us **summarise the features** of a dataset.
- ❖ The most commonly used descriptive statistics are:
  - **Mean:** Measure of central tendency, also known as the **average**. It is equal to the sum of the dataset divided by the number of elements.
  - **Median:** Measure of central tendency. The middle value when the dataset is sorted in ascending or descending order.
  - **Mode:** Measure of central tendency. The value which has the highest frequency in the data.
  - **Variance:** Measure the spread of data around the mean.
  - **Standard Deviation:** Square root of the variance, measures dispersion about the mean.

# Descriptive Statistics

Calculation	Formula	Notes
Population Mean	$\mu = \frac{\sum X_i}{N}$	$\mu$ = population average $X$ = individual values of population $N$ = count of individual values
Sample Mean	$\bar{x} = \frac{\sum x_i}{n}$	$\bar{x}$ = sample average $x$ = individual values of population $n$ = count of individual values in sample
Weighted Mean	$\bar{x} = \frac{\sum w_i x_i}{\sum w_i}$	$\bar{x}$ = weighted sample average $w_i$ = weight of value $i$ $x_i$ = individual value to be weighted
Sample Mean of grouped data	$\bar{x} = \frac{\sum f_i x_i}{n}$	$f_i$ = number of observations in the $i$ th group $x_i$ = midpoint of the $i$ th class $n$ = count of all observations of $i$ th classes
Mean Deviation	$MD = \frac{\sum  x_i - \bar{x} }{n}$	$\bar{x}$ = sample average $x$ = individual values in sample $n$ = count of individual values in sample
Population Variance	$\sigma^2 = \frac{\sum (X_i - \mu)^2}{N}$	$\mu$ = population average $X$ = individual values in population $N$ = count of values in population

```
heights = [168, 170, 150, 160, 182,
           140, 175, 191, 152, 150]
```

```
# Mean
```

```
mean = np.mean(heights)
print(mean)
```

```
# Median
```

```
median = np.median(heights)
print(median)
```

```
# Mode
```

```
mode = sp.stats.mode(heights)
print(mode)
```

```
# Variance
```

```
var = np.var(heights)
print(var)
```

```
# Standard Deviation
```

```
std = np.std(heights)
print(std)
```

```
let heights = [168, 170, 150, 160, 182,
               140, 175, 191, 152, 150];
```

```
// Mean
```

```
let mean = math.mean(heights);
console.log(mean);
```

```
// Median
```

```
let median = math.median(heights);
console.log(median);
```

```
// Mode
```

```
let mode = math.mode(heights);
console.log(mode);
```

```
// Variance
```

```
let variance = math.variance(heights);
console.log(variance);
```

```
// Standard Deviation
```

```
let std = math.std(heights);
console.log(std);
```

# Inferential Statistics

Use of data from a sample to make inferences or predictions about a population

- ❖ Inferential statistics help us **make estimates and predictions** about our population and **test hypotheses** made about our population.
- ❖ Inferential Statistics allow us to use the sample to make reasonable estimates about the population as a whole.
- ❖ **Hypothesis testing:** Involves setting up a **null hypothesis** and an **alternative hypothesis** and then conducting a **statistical test of significance**. Conclusions are drawn based on the **test statistic**, the **critical value** and the **confidence interval**.
- ❖ **Confidence Interval:** Helps us estimate the parameters of a population. It allows us to **quantify the accuracy** of our estimates.

# Statistics in Code

- ❖ JavaScript is usually **not** used to perform extensive statistics. You can use external libraries like for these simple statistics [mathjs](#) or [simple-statistics](#).
- ❖ In Python, the **scipy** and **numpy** library provides extensive statistical functions and tools to help us analysis data sets and make predictions.

```
from scipy import stats
import numpy as np

data = np.array([23, 78, 789, 12, 90, 384, 12, 3759, 109, 45, 67])

# This function returns descriptive statistics for this dataset
# Including mean, variance, min and max
print(stats.describe(data))
```

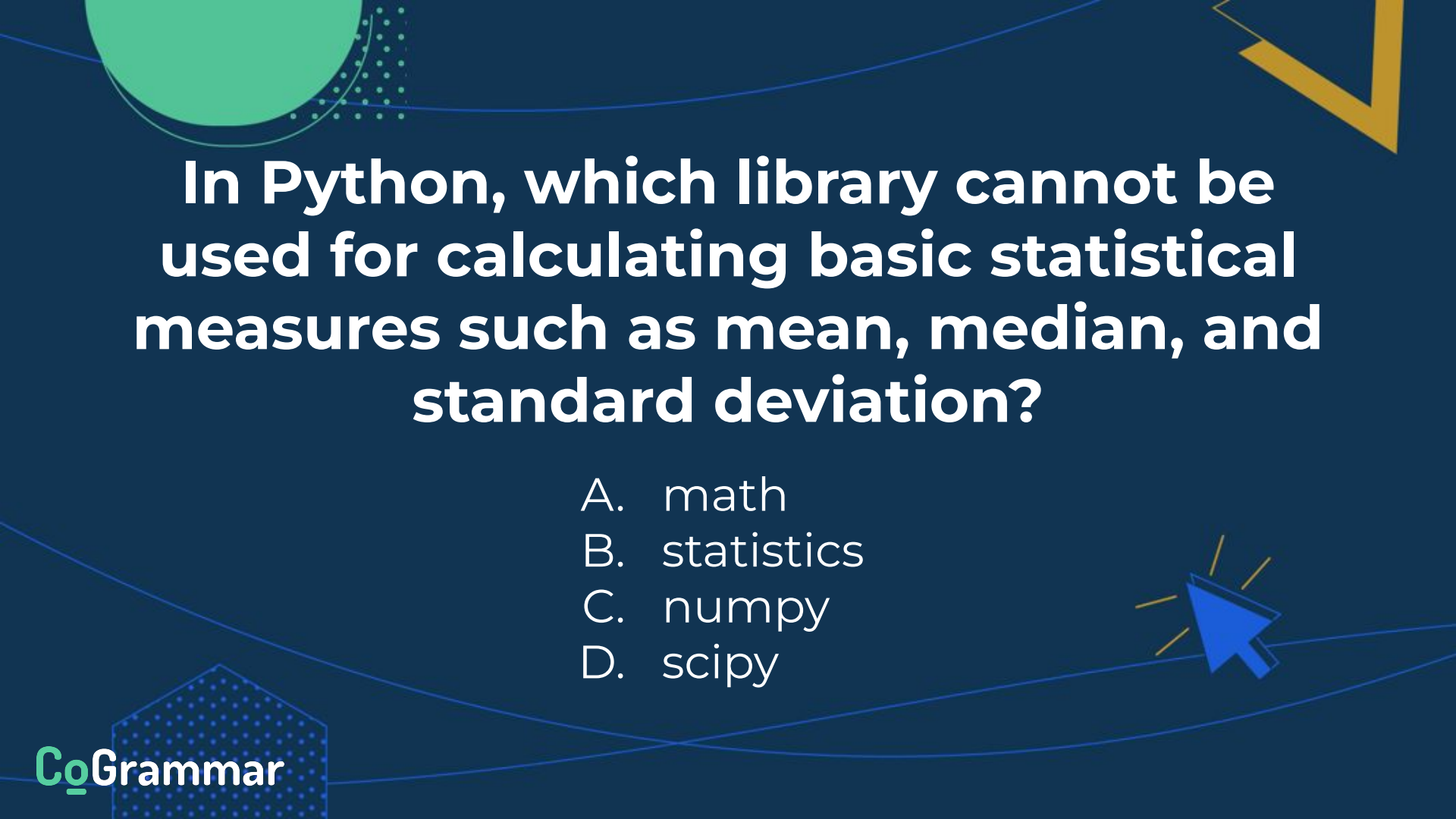


# Which of the following is an example of a continuous random variable?

- A. Number of heads in 10 coin flips
- B. Temperature in degrees Celsius
- C. Number of students in a class
- D. Number of cars passing a toll booth in an hour

## Correct Answer: B

- ❖ A continuous random variable **cannot be counted** and its value falls within a certain interval.
- ❖ It's value is dependent on a possible outcome of an experiment, thus *“Temperature in degrees Celsius”* is the best answer to this question.
- ❖ All the other options, despite being possible outcomes of an experiment, are all countable variables. Thus they would be **discrete random variables**.



**In Python, which library cannot be used for calculating basic statistical measures such as mean, median, and standard deviation?**

- A. math
- B. statistics
- C. numpy
- D. scipy



## Correct Answer: A

- ❖ The **scipy** Python module is the best choice for calculating statistics of a dataset.
- ❖ The built-in Python **statistics** module was not intended for use with large datasets and complex statistics, but can be used to calculate basic statistics.
- ❖ The **numpy** library also provides statistical functions but in it's documentation it is said to be less efficient than SciPy. It is best used for descriptive statistics an basic statistical functions.

# Portfolio Assignment: SE

## Event Probability Simulator

**Objective:** Build a Python application that simulates the probability of different outcomes for given scenarios. The tool should allow users to define a scenario, including the number of trials and the specific events to simulate, and then output the probability of each event occurring.

# Portfolio Assignment: SE

## Requirements:

- Develop the application in Python, using libraries like numpy or scipy for mathematical operations.
- Allow users to input parameters for simulations, such as the number of trials and event probabilities.
- Implement functionality to calculate permutations and combinations where needed to simulate scenarios (e.g. drawing cards from a deck, dice rolls).
- Utilise Bayes' theorem to update probabilities based on new information for certain scenarios.
- Include a README file that explains how to use the application, with examples of different scenarios and how to interpret output.

# Portfolio Assignment: DS

## Descriptive Statistics and Data Visualisation Tool

**Objective:** Create a Python script that computes descriptive statistics for a given dataset and visualizes these statistics through various charts and graphs.

The tool should help in understanding the distribution, central tendency, and variability of data.

# Portfolio Assignment: DS

## Requirements:

- Utilize pandas for data manipulation and matplotlib or seaborn for creating visualizations.
- Compute basic descriptive statistics (mean, median, mode, standard deviation, quartiles) and display them in a user-friendly format.
- Create visualizations such as histograms, box plots, and scatter plots to represent the data distribution and statistical summaries.
- Implement functionality to apply basic inferential statistics techniques, like calculating confidence intervals for sample means.
- Provide a README file detailing the tool's functionality, how to run it, and examples of output given sample datasets.

# Portfolio Assignment: WD

## Interactive Hypothesis Testing Tool

**Objective:** Develop a web application that allows users to perform hypothesis testing on data they input. The application should guide users through selecting a hypothesis, choosing a significance level, and then inputting or uploading data to test against the hypothesis.

# Portfolio Assignment: WD

## Requirements:

- Build the application using HTML/CSS for the frontend and JavaScript for the backend logic.
- Design an intuitive UI for users to input their hypothesis, select a significance level (e.g., 0.05, 0.01), and input or upload their data.
- Calculate and display the result of the hypothesis test, including the p-value and whether the hypothesis is accepted or rejected at the chosen significance level.
- Offer brief explanations or tooltips about the hypothesis testing process and the significance of the p-value and confidence intervals.
- Include a README file with instructions on setting up and using the application, along with a simple example to demonstration.



# Summary

## **Permutations**

- Arrangement of objects where order is important.

## **Combinations**

- Selection where order doesn't matter

## **Sample Space and Events**

- The set of all possible outcomes of an experiment.
- An event is a subset of the sample space that we are interested in.

# Summary

## Basic Probability

- The likelihood of an event occurring, calculated as favorable outcomes divided by total outcomes.

## Conditional Probability and Independence

- Probability of an event given another has occurred.
- Independence when one event does not influence another.

## Bayes Theorem

- A formula for conditional probability which provides a way to revise existing predictions or theories (update probabilities) given new or additional evidence.

# Summary

## Distributions

- Uniform: Equal probability for all outcomes.
- Binomial: Probability of 'success' in 'n' trials.
- Normal: Bell-curve distribution, common in natural data.

## Descriptive Statistics

- A set of methods for organising, summarising and presenting data.

## Inferential Statistics

- Use of data from a sample to make inferences or predictions about a population.

## Additional Resources

- ❖ [Better Explained](#) - Easy Permutations and Combinations
- ❖ [Math is Fun](#) - Combinations and Permutations
- ❖ [Khan Academy](#) - Counting, permutations, and combinations
- ❖ [Khan Academy](#) - Basic Probability
- ❖ [LibreTexts](#) - Basic Probability more examples
- ❖ [Coursera](#) - Basic to Advanced Probability (for the VERY curious)
- ❖ [ThinkStats](#) - Textbook covering basic and inferential statistics.
- ❖ [CueMath](#) - Inferential Statistics Theory and Examples

# CoGrammar

## Q & A SECTION

**Please use this time to ask  
any questions relating to the  
topic, should you have any.**

# Thank you for attending



Department  
for Education

CoGrammar

