# Question 1

**Question Content**
A company is analyzing customer purchase amounts over a year. They want to determine if these amounts follow a uniform distribution. How can they identify this?

**Question Type**: MCQ

**Options**:
A) By checking if the data forms a bell curve when graphed.
B) By ensuring all outcomes have equal frequency across their range.
C) By observing clustering at certain values in the dataset.
D) By calculating skewness and kurtosis values.

**Approach**:

1. Recall the definition of a uniform distribution (all values occur with the same frequency).
2. Consider how the data should be distributed (flat/constant frequency across the range) if it is truly uniform.
3. Compare frequencies of different intervals within your data.
4. If all intervals have roughly the same frequency, that points toward uniformity.

# Question 2

**Question Content**
A manufacturing plant experiences an average of 3 equipment failures per hour. What is the probability of observing exactly 5 failures in an hour? Round your answer to the nearest whole percentage.

$P(X = x) = \frac{e^{-\lambda} \cdot \lambda^x}{x!}$

where:

- $x$ = Actual number of occurrences
- $e$ = Euler's number (≈ 2.718)

**Question Type**: Integer Type

**Approach**:

1. Identify that the number of failures follows a Poisson distribution with $\lambda = 3$.
2. Use the Poisson probability mass function $P(X = x) = \frac{e^{-\lambda}\lambda^x}{x!}$.
3. Plug in $x = 5$ and $\lambda = 3$.
4. Perform the factorial calculation for 5!, the power calculation for $\lambda^5$, and multiply by $e^{-3}$.
5. Convert the resulting decimal into a percentage and round.

# Question 3

### Question Content

A marketing firm conducts a survey with 15 questions, each having a probability of being answered correctly as 0.4. What is the probability of exactly 6 correct answers? Round your answer to the nearest whole percentage.

$$P(X = r) = C(n, r)\, p^r\, (1 - p)^{(n-r)}$$

where:

- $n$ = Number of trials
- $r$ = Number of successes
- $p$ = Probability of success in a single trial
- $C(n, r)$ = Binomial coefficient = $\frac{n!}{r!(n-r)!}$

**Question Type**: Integer Type

**Approach**:

1. Recognize that this follows a binomial distribution with $n = 15$ and $p = 0.4$.
2. Use the formula for binomial probability $P(X = r)$.
3. Calculate the binomial coefficient $C(15, 6)$.
4. Raise $p$ to the power of 6 and $(1 - p)$ to the power of 9.
5. Multiply all terms and then convert to a percentage and round.

# Question 4

### Question Content

A researcher conducts a study to determine if a new medication is effective. The null hypothesis ($H_0$) states that the medication has no effect. If the researcher fails to reject $H_0$, what does this imply?

**Question Type**: MSQ

**Options**:
A) There's strong evidence against "medication has no effect".
B) $H_0$ must be true.
C) There's insufficient evidence to support "medication has any effect".
D) $H_0$ has been proven correct.

**Approach**:

1. Understand the hypothesis testing framework (null hypothesis vs. alternative hypothesis).
2. Recall the meaning of "failing to reject $H_0$": it does not prove $H_0$ true, but rather indicates insufficient evidence to support the alternative.
3. Differentiate between "failing to reject" and "confirming" a hypothesis.

# Question 5

**Question Content**
A rare genetic disorder occurs in 1 out of every 10,000 people. In a city of 1,000,000 people, what is the expected number of people with this disorder?

**Question Type**: Integer Type

**Approach**:

1. Recognize this as a binomial setting (each individual either has or does not have the disorder).
2. Use the concept of expected value in a binomial distribution $E(X) = n \times p$.
3. Identify $n = 1,000,000$ and $p = \frac{1}{10,000}$.
4. Multiply to get the expectation (no rounding needed here, it should be an integer by the setup).

# Question 6

**Question Content**
What does it mean if data follows a normal distribution?

**Question Type**: MSQ

**Options**:
A) Data points are uniformly distributed across all values.
B) Most values cluster around the mean with symmetrical tails on both sides.
C) There are no outliers present in the data set.
D) Data can only take integer values.

**Approach**:

1. Review the shape and properties of a normal (Gaussian) distribution: bell-shaped, symmetric around the mean.
2. Contrast this with other distributions (uniform vs. normal, discrete vs. continuous).
3. Identify key features of normality: central clustering around the mean and symmetric tails.

# Question 7

**Question Content**
A researcher conducts a hypothesis test to determine if a new fertilizer increases crop yield. The p-value obtained is 0.05. What does this p-value represent?

**Question Type**: MSQ

**Options**:
A) The probability that the null hypothesis is true.
B) The likelihood of observing data as extreme as what was observed, assuming the null hypothesis is true.
C) The threshold for rejecting the null hypothesis.
D) The confidence level of the results.

**Approach**:

1. Recall the definition of a p-value: probability of obtaining results at least as extreme as the observed data if the null hypothesis is correct.
2. Understand that p-value $\neq$ probability that $H_0$ is true.
3. Connect p-value interpretation to standard significance testing procedures (e.g., compare with alpha level).

# Question 8

**Question Content**
Which of the following statements about NumPy are true?

**Question Type**: MSQ

**Options**:
A) NumPy stands for Numeric Python
B) NumPy arrays can only store elements of the same data type
C) NumPy is slower than standard Python lists for large datasets
D) NumPy provides powerful tools for working with arrays

**Approach**:

1. Recall what NumPy is and its common uses in Python.
2. Note that NumPy arrays are homogeneous (same data type), while standard Python lists can be heterogeneous.
3. Know that NumPy is usually faster than plain Python lists for vectorized operations.
4. Recognize that NumPy includes many array manipulation and numerical computation features.

# Question 9

**Question Content**
Which of the following NumPy functions can be used to create arrays?

**Question Type**: MSQ

**Options**:
A) np.array()
B) np.arange()
C) np.ones()
D) np.zeros()

**Approach**:

1. Recall the various ways to create NumPy arrays: converting lists, creating ranges, creating arrays of ones/zeros, etc.
2. Identify what each function does:
   - `np.array()` converts a Python list or other sequence into a NumPy array.

- `np.arange()` generates a range of values in an array.
- `np.ones()` creates an array of all ones.
- `np.zeros()` creates an array of all zeros.

# Question 10

### Question Content

What will be the output of the following code?

```
import numpy as np
arr = np.array([[1,3,-1],[2,4,-2]])
print(f"{arr[1,0]} and {arr[0][-1]}")
```

**Question Type**: MCQ

**Options**:

A) 1 and -1
B) 2 and 2
C) 2 and -1
D) -1 and -2

**Approach**:

1. Understand Python's zero-based indexing for 2D arrays ( `arr[row_index, column_index]` ).
2. Check `arr[1,0]` to find the element in the second row, first column.
3. Check `arr[0][-1]` to find the last element in the first row ( `-1` indicates the last column).
4. Format the string accordingly.

# Question 11

### Question Content

You have a NumPy array `arr = np.array([2,3,4,3*3,5//2])` . Which of the following will create a new array that contains only the odd numbers from `arr` ?

**Question Type**: MSQ

**Options**:

A) `arr % 2 == 0`

B) `arr[arr % 2 == 0]`

C) `arr[arr == 2]`

D) `arr[arr % 2 != 0]`

**Approach**:

1. Recall how array filtering works in NumPy (boolean indexing).
2. Understand the difference between creating a boolean mask vs. using it to index the array.
3. Identify which expression uses a condition that checks for odd values.
4. Check each option carefully to see whether it yields odd or even numbers, or if it's just a mask without returning new elements.

# Question 12

### Question Content

What is the shape of the resulting array after the following operation?

```
import numpy as np
a = np.ones((2,  3,  4))
b = np.ones((3,  4))
result = a + b
```

**Question Type**: MCQ

**Options**:

A) (2, 3, 4)

B) (3, 4)

C) (2, 3)

D) Error: shapes not aligned

**Approach**:

1. Recognize how NumPy broadcasting works.
2. Compare the shapes of `a` (2×3×4) and `b` (3×4).
3. Determine how `b` is "expanded" so that each dimension lines up with `a`.
4. Check whether the resulting shape matches that of `a` or leads to an error.

## Question 13

**Question Content**
Which of the following are valid ways to perform matrix multiplication in NumPy?

**Question Type**: MSQ

**Options**:
A) `np.dot(a, b)`
B) `a @ b`
C) `np.matmul(a, b)`
D) `a * b`

**Approach**:

1. Distinguish between element-wise multiplication vs. matrix multiplication.
2. Recall the functions/methods that explicitly perform matrix multiplication in NumPy.
3. Note the newer Python operator `@` for matrix multiplication.
4. Identify which function or operator gives a true matrix product.

## Question 14

**Question Content**
Which of the following statements about broadcasting in NumPy are true?

**Question Type**: MSQ

**Options**:
A) It always creates a copy of the smaller array
B) It can lead to implicit dimension creation
C) It only works with arrays of the same dimension
D) It follows a set of rules to determine compatibility

**Approach**:

1. Recall the broadcasting rules (shape alignment from right to left, expanding dimensions of size 1, etc.).

2. Understand that broadcasting does not necessarily create a copy, and arrays of different dimensions can be broadcast if they are compatible.
3. Identify which statements align with these rules.

# Question 15

**Question Content**
Given a 2D NumPy array `arr`, which of the following operations will correctly compute the column-wise sum?

**Question Type**: MSQ

**Options**:
A) `np.sum(arr, axis=0)`
B) `arr.sum(axis=1)`
C) `np.sum(arr, axis=1)`
D) `arr.sum(axis=0)`

**Approach**:

1. Recall how axes work in NumPy (axis=0 typically refers to the "rows" dimension when summing, resulting in a sum across rows → a column-wise sum).
2. Compare axis=0 vs. axis=1 to see which corresponds to summing columns vs. summing rows.
3. Identify the difference between `np.sum(arr, axis=...)` and `arr.sum(axis=...)` —they do the same thing but are just different syntaxes.