FINAL ASSIGNMENT

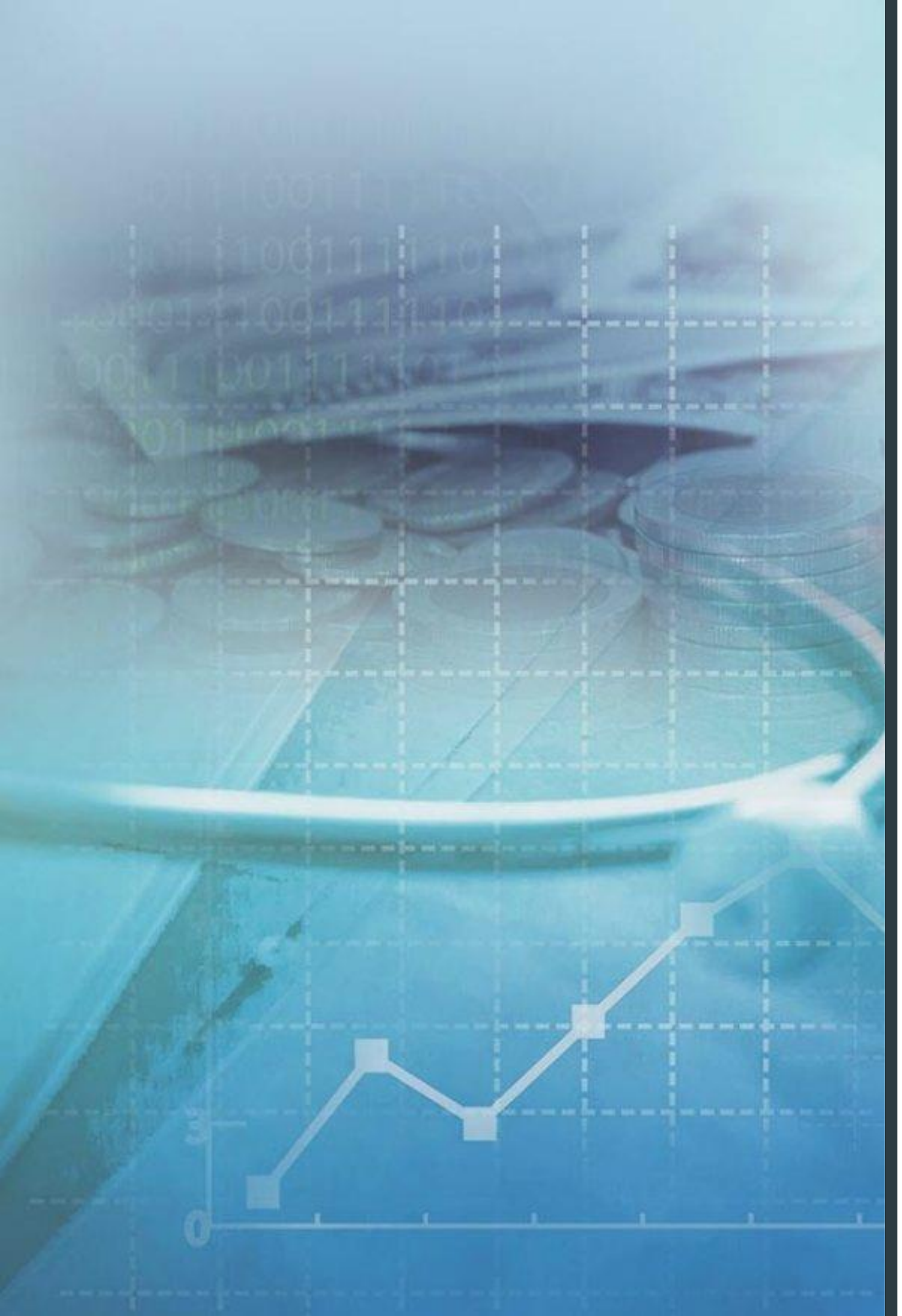
MODULE:
PROBABILITY AND STATISTICS
PROGRAM:
DATA RESEARCH ANALYST

- Student: Rina Rafalski
- Mentors: Rony-Reuven Nir, MD, PhD;
  Guy Uziel

NAYA College

Table of contents

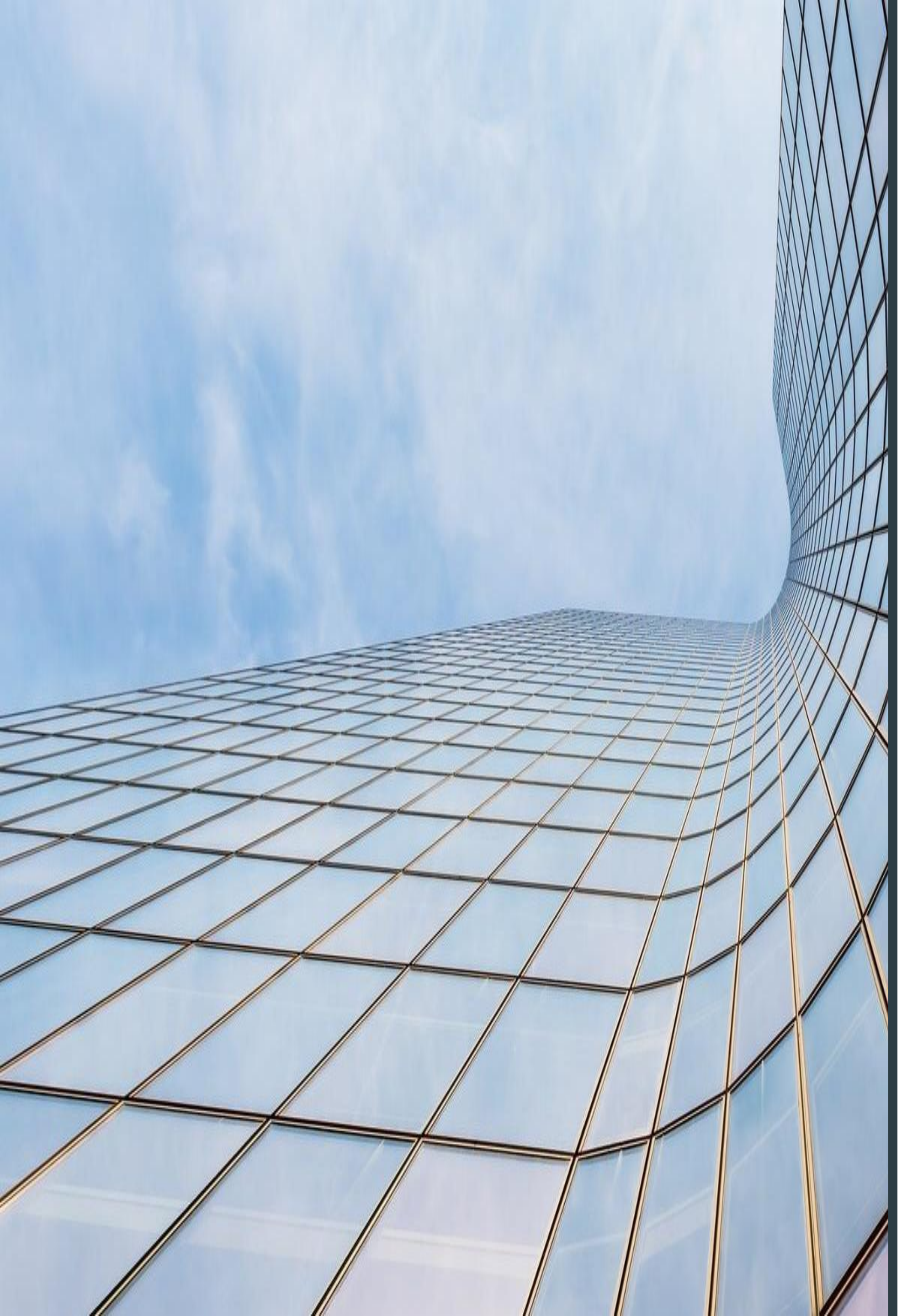
Table of contents



Assignment Description

A company has recently launched a new health insurance product. The product has been sold for a few months now. The marketing director wishes to examine the company's sales performance thus far. You work as a data research analyst in the marketing department. The director asked you to analyze data from 10,000 sales leads. You have gathered the data in the "SALES DATA" sheet, which contains the following parameters:

- ID – a unique identifier of each customer
- AGE GROUP – the age group of the customer
- CHANNEL OF SALE – either "internet" or "phone", depending on where the customers left their contact details
- SALES – an indicator variable (1 – successful sale, 0 – unsuccessful sale)



Question 1

Question 1
1.01 Probability: task 1

Calculate the probability distribution function (PDF) of the 10,000 leads by AGE GROUP and by CHANNEL OF SALE separately (present your findings as the percent of each group out of 10,000)

Guidance:

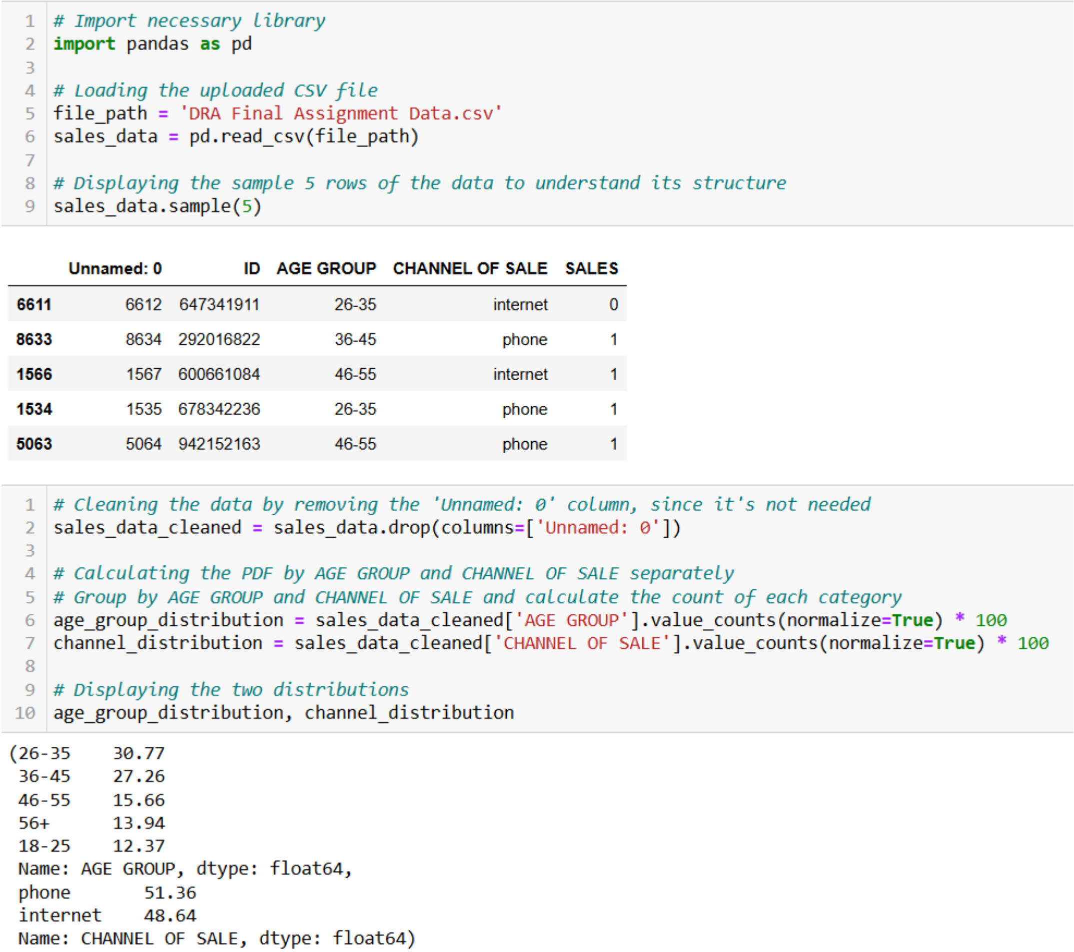
- You are calculating the probability distribution function of your customers belonging to each sub-category of a variable
- The probability distribution function should sum up to 100% (or 1)
- The probability of each sub-category should be calculated as the number of “wanted” outcomes (e.g. the number of people who have an “internet” sales channel) divided by the number of “possible” outcomes (the total number of leads) - מצוי חלקי רצוי

Answer:

Probability distribution functions (PDFs) for the 10,000 leads, presented as percentages:

By AGE GROUP:

26-35:	30.77%
36-45:	27.26%
46-55:	15.66%
56+:	13.94%
18-25:	12.37%



```python
# Import necessary library
import pandas as pd

# Loading the uploaded CSV file
file_path = 'DRA Final Assignment Data.csv'
sales_data = pd.read_csv(file_path)

# Displaying the sample 5 rows of the data to understand its structure
sales_data.sample(5)
```

	Unnamed: 0	ID	AGE GROUP	CHANNEL OF SALE	SALES
6611	6612	647341911	26-35	internet	0
8633	8634	292016822	36-45	phone	1
1566	1567	600661084	46-55	internet	1
1534	1535	678342236	26-35	phone	1
5063	5064	942152163	46-55	phone	1

```python
# Cleaning the data by removing the 'Unnamed: 0' column, since it's not needed
sales_data_cleaned = sales_data.drop(columns=['Unnamed: 0'])

# Calculating the PDF by AGE GROUP and CHANNEL OF SALE separately
# Group by AGE GROUP and CHANNEL OF SALE and calculate the count of each category
age_group_distribution = sales_data_cleaned['AGE GROUP'].value_counts(normalize=True) * 100
channel_distribution = sales_data_cleaned['CHANNEL OF SALE'].value_counts(normalize=True) * 100

# Displaying the two distributions
age_group_distribution, channel_distribution
```

```
(26-35    30.77
 36-45    27.26
 46-55    15.66
 56+      13.94
 18-25    12.37
 Name: AGE GROUP, dtype: float64,
 phone       51.36
 internet    48.64
 Name: CHANNEL OF SALE, dtype: float64)
```

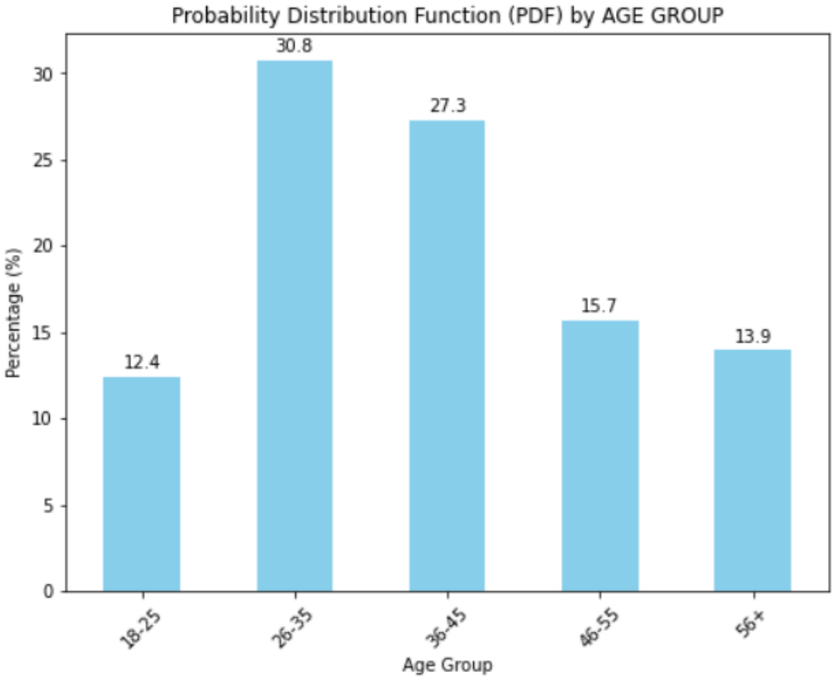
Question 1
1.02 Probability: task 2

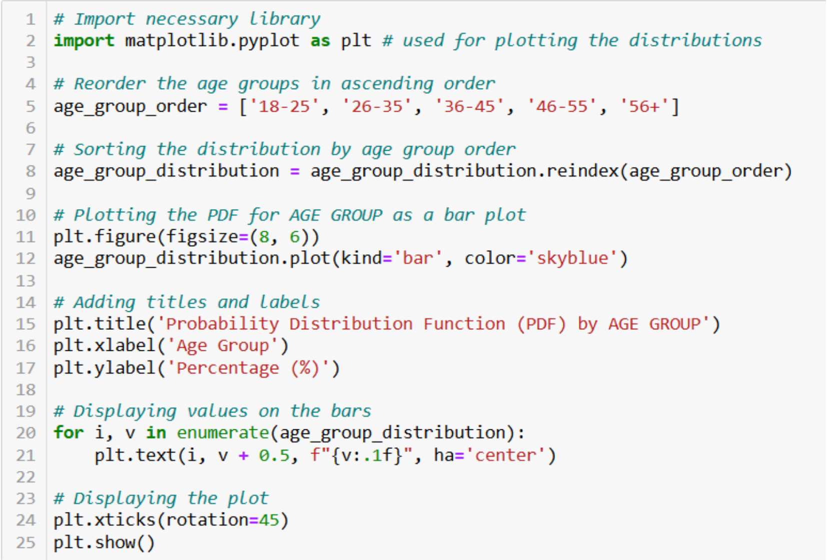
Plot the probability distribution function (PDF) for AGE GROUP

Guidance:

- Create a bar-plot for each of the 5 sub-categories of AGE GROUP

Answer:





```python
# Import necessary library
import matplotlib.pyplot as plt # used for plotting the distributions

# Reorder the age groups in ascending order
age_group_order = ['18-25', '26-35', '36-45', '46-55', '56+']

# Sorting the distribution by age group order
age_group_distribution = age_group_distribution.reindex(age_group_order)

# Plotting the PDF for AGE GROUP as a bar plot
plt.figure(figsize=(8, 6))
age_group_distribution.plot(kind='bar', color='skyblue')

# Adding titles and labels
plt.title('Probability Distribution Function (PDF) by AGE GROUP')
plt.xlabel('Age Group')
plt.ylabel('Percentage (%)')

# Displaying values on the bars
for i, v in enumerate(age_group_distribution):
    plt.text(i, v + 0.5, f"{v:.1f}", ha='center')

# Displaying the plot
plt.xticks(rotation=45)
plt.show()
```

Insights

- The age group 26-35 has the highest percentage of leads, making up 30.8% of the total leads. This indicates that the majority of people showing interest in the new health insurance product belong to this age range.
- The 36-45 age group is the second largest segment, comprising 27.3% of the total leads.

Conclusion

The plot suggests that middle-aged individuals (between 26-45 years old) are the primary focus for the company, either due to marketing targeting, product relevance, or interest in health insurance products in general.

Question 1

1.03 Combinatorics & Joint distribution: task 1

How many combinations of AGE GROUP and CHANNEL OF SALE do you expect to see?

Guidance:

- Use combinatorics, think of a multi-stage experiment and the number of options at each stage

Answer:

Determining the number of combinations of AGE GROUP and CHANNEL OF SALE using combinatorics:

This is a multi-stage experiment where:

- There are 5 possible AGE GROUPS:

  18-25, 26-35, 36-45, 46-55, 56+

- There are 2 possible CHANNELS OF SALE:

  Internet, Phone

- Since each AGE GROUP can pair with each CHANNEL OF SALE, the total number of combinations is:

  $Total\ combinations = 5\times2 = 10$

We expect to see **10 different combinations of AGE GROUP and CHANNEL OF SALE.**

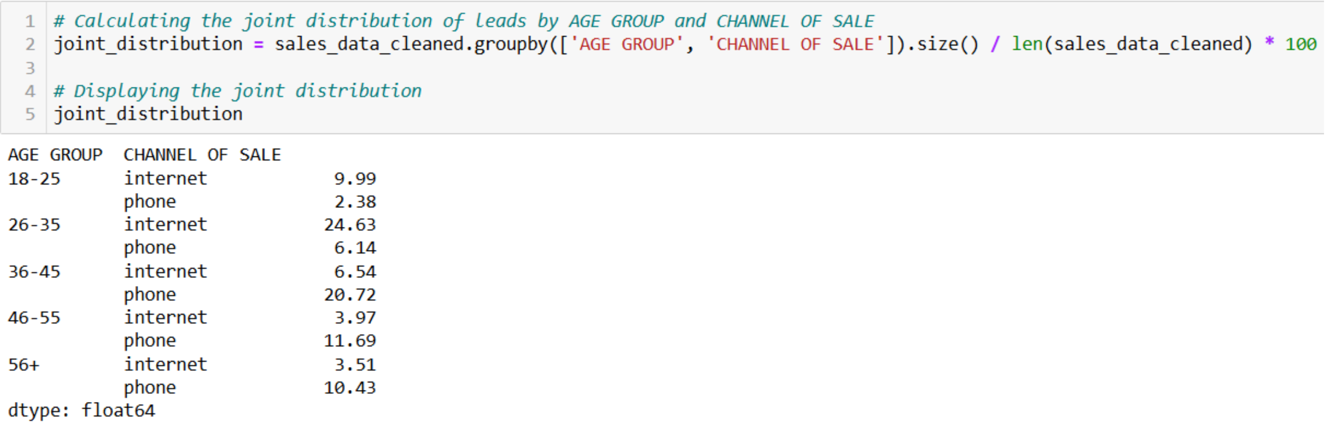
Question 1

1.04 Combinatorics & Joint distribution: task 2

Calculate the joint distribution of leads by AGE GROUP and CHANNEL OF SALE

Guidance:

- The joint distribution is the intersection between AGE GROUP and CHANNEL OF SALE
- The number of different combinations for which you will calculate the joint distribution is determined by your answer to the previous section
  For example, to calculate the joint probability of AGE GROUP = 18-25 AND CHANNEL OF SALE = “phone”, you will count the number of customers who are both 18-25 and contacted the company via “phone”, then divide by the total number of leads
- The joint probability should also add up to 100% (or 1)



```python
# Calculating the joint distribution of leads by AGE GROUP and CHANNEL OF SALE
joint_distribution = sales_data_cleaned.groupby(['AGE GROUP', 'CHANNEL OF SALE']).size() / len(sales_data_cleaned) * 100

# Displaying the joint distribution
joint_distribution
```

```
AGE GROUP  CHANNEL OF SALE
18-25      internet            9.99
           phone               2.38
26-35      internet           24.63
           phone               6.14
36-45      internet            6.54
           phone              20.72
46-55      internet            3.97
           phone              11.69
56+        internet            3.51
           phone              10.43
dtype: float64
```

Answer:

Here is the joint distribution of leads by AGE GROUP and CHANNEL OF SALE:

18-25:

- Internet: 9.99%
- Phone: 2.38%

26-35:

- Internet: 24.63%
- Phone: 6.14%

36-45:

- Internet: 6.54%
- Phone: 20.72%

46-55:

- Internet: 3.97%
- Phone: 11.69%

56+:

- Internet: 3.51%
- Phone: 10.43%

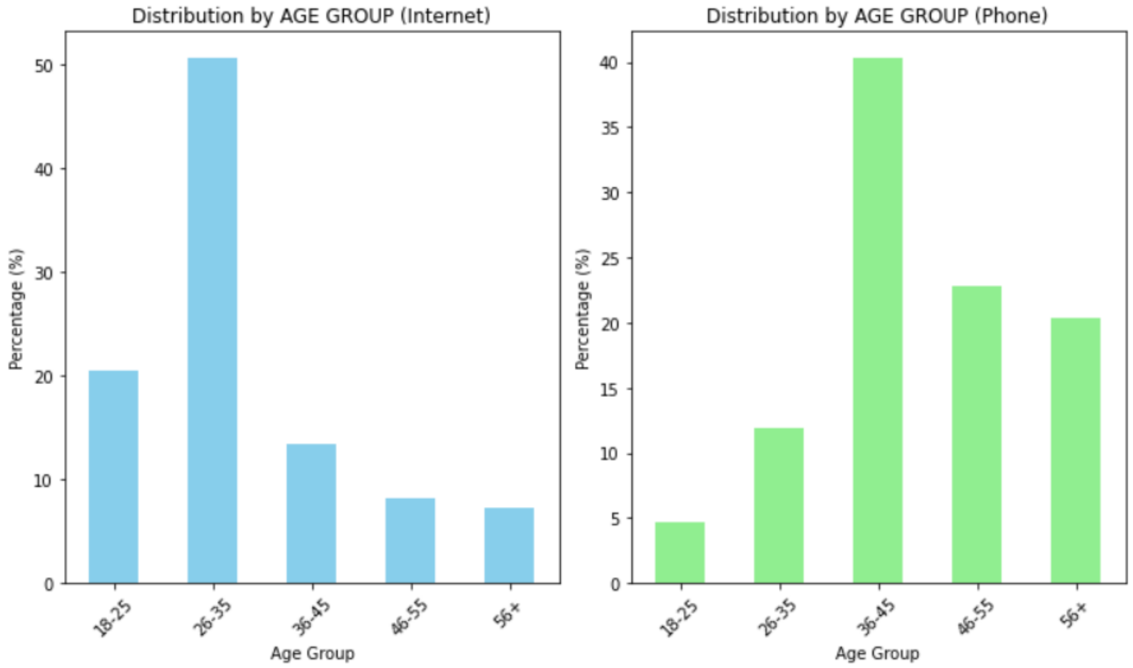
Question 1
1.05 Combinatorics & Joint distribution: task 3

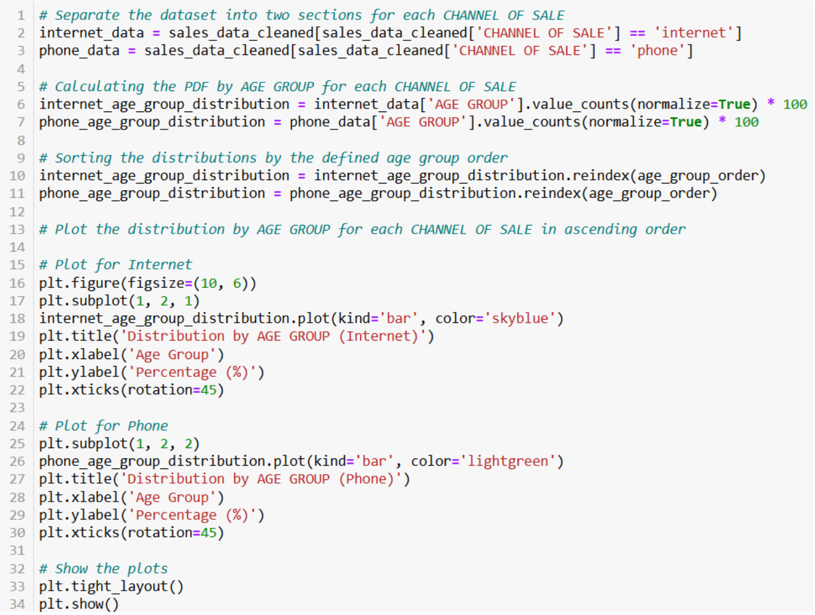
Plot the distribution by AGE GROUP for each CHANNEL OF SALE separately

Guidance:

- Separate your dataset into two sections, for each CHANNEL OF SALE, then create a bar-plot of the probability distribution by AGE GROUP for each of the sections

Answer:





```python
# Separate the dataset into two sections for each CHANNEL OF SALE
internet_data = sales_data_cleaned[sales_data_cleaned['CHANNEL OF SALE'] == 'internet']
phone_data = sales_data_cleaned[sales_data_cleaned['CHANNEL OF SALE'] == 'phone']

# Calculating the PDF by AGE GROUP for each CHANNEL OF SALE
internet_age_group_distribution = internet_data['AGE GROUP'].value_counts(normalize=True) * 100
phone_age_group_distribution = phone_data['AGE GROUP'].value_counts(normalize=True) * 100

# Sorting the distributions by the defined age group order
internet_age_group_distribution = internet_age_group_distribution.reindex(age_group_order)
phone_age_group_distribution = phone_age_group_distribution.reindex(age_group_order)

# Plot the distribution by AGE GROUP for each CHANNEL OF SALE in ascending order

# Plot for Internet
plt.figure(figsize=(10, 6))
plt.subplot(1, 2, 1)
internet_age_group_distribution.plot(kind='bar', color='skyblue')
plt.title('Distribution by AGE GROUP (Internet)')
plt.xlabel('Age Group')
plt.ylabel('Percentage (%)')
plt.xticks(rotation=45)

# Plot for Phone
plt.subplot(1, 2, 2)
phone_age_group_distribution.plot(kind='bar', color='lightgreen')
plt.title('Distribution by AGE GROUP (Phone)')
plt.xlabel('Age Group')
plt.ylabel('Percentage (%)')
plt.xticks(rotation=45)

# Show the plots
plt.tight_layout()
plt.show()
```

Insights

Internet Sales:

- The majority of internet sales leads come from the 26-35 age group, which makes up over 50%.

Phone Sales:

- Phone sales are more evenly distributed among several age groups.
- The largest group is 36-45, with over 40%, followed by the 46-55 and 56+ groups, which together account for a significant portion of phone sales.

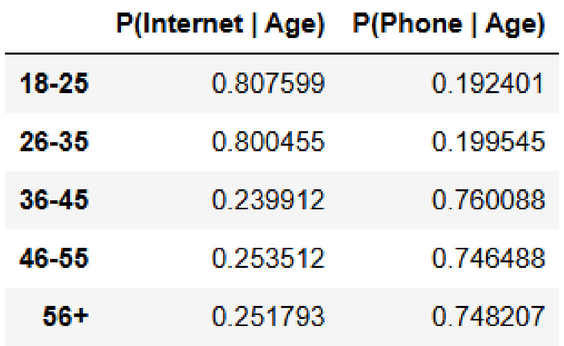
Question 1

1.06 Conditional probability: task 1

Given that a person belongs to a specific age group, calculate the probability that they will choose to use each CHANNEL OF SALE

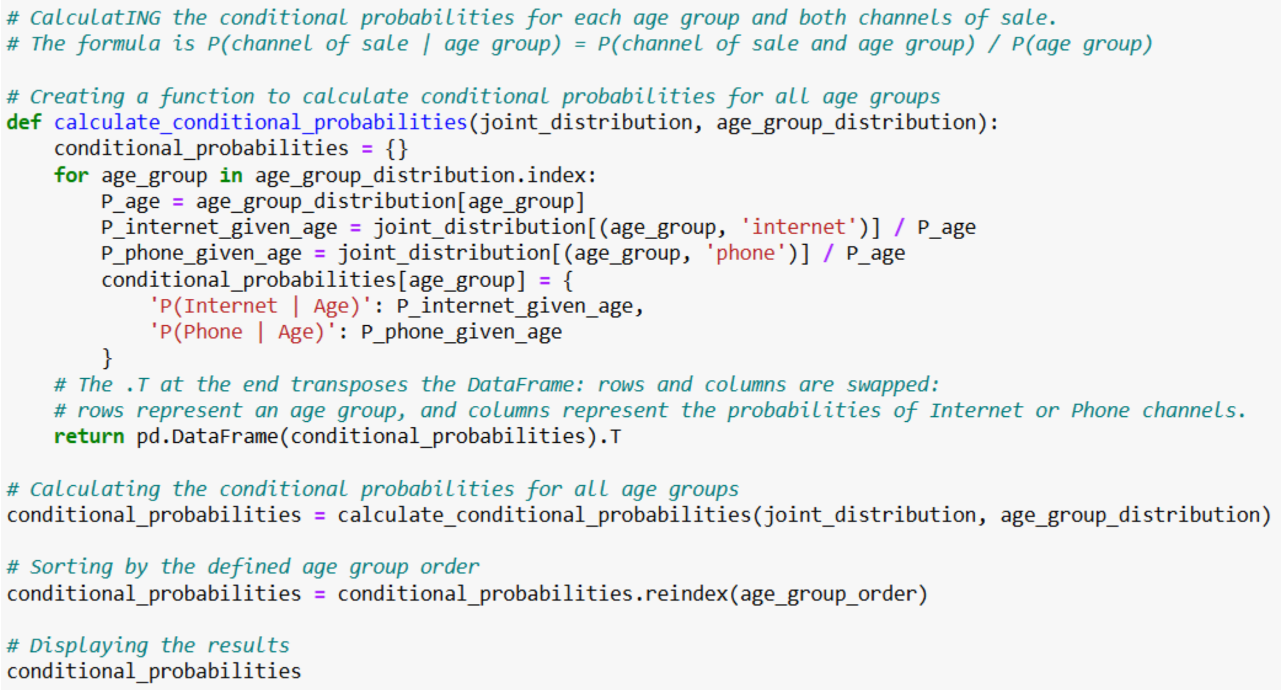
Guidance:

- A conditional probability is calculated as the joint probability divided by the probability of the (הסתברות משותפת חלקי ההסתברות של התנאי) condition
- You can use your previous calculations of the joint probability function of AGE GROUP and CHANNEL OF SALE combined with the probability function for AGE GROUP



Answer:

	P(Internet | Age)	P(Phone | Age)
18-25	0.807599	0.192401
26-35	0.800455	0.199545
36-45	0.239912	0.760088
46-55	0.253512	0.746488
56+	0.251793	0.748207



```python
# CalculatING the conditional probabilities for each age group and both channels of sale.
# The formula is P(channel of sale | age group) = P(channel of sale and age group) / P(age group)

# Creating a function to calculate conditional probabilities for all age groups
def calculate_conditional_probabilities(joint_distribution, age_group_distribution):
    conditional_probabilities = {}
    for age_group in age_group_distribution.index:
        P_age = age_group_distribution[age_group]
        P_internet_given_age = joint_distribution[(age_group, 'internet')] / P_age
        P_phone_given_age = joint_distribution[(age_group, 'phone')] / P_age
        conditional_probabilities[age_group] = {
            'P(Internet | Age)': P_internet_given_age,
            'P(Phone | Age)': P_phone_given_age
        }
    # The .T at the end transposes the DataFrame: rows and columns are swapped:
    # rows represent an age group, and columns represent the probabilities of Internet or Phone channels.
    return pd.DataFrame(conditional_probabilities).T

# Calculating the conditional probabilities for all age groups
conditional_probabilities = calculate_conditional_probabilities(joint_distribution, age_group_distribution)

# Sorting by the defined age group order
conditional_probabilities = conditional_probabilities.reindex(age_group_order)

# Displaying the results
conditional_probabilities
```

Question 1

1.07 Conditional probability: task 2

Comment on your results - is there a recommendation regarding the CHANNEL OF SALE preference of each group that you can give?

Guidance:

- Comment on the distribution of channel of sale for each age group - are there differences or similarities that the marketing department should be aware of?
- Contrast it with the probability distribution function of CHANNEL OF SALE

Answer:

The conditional probabilities highlight some important differences and similarities across age groups when it comes to the choice of the CHANNEL OF SALE (Internet vs. Phone):

Differences and Similarities by Age Group:

- 18-25 and 26-35 Age Groups:
  - Both age groups show a strong preference for the Internet channel, with over 80%. This indicates that younger customers are significantly more likely to use the Internet, making digital marketing an effective strategy for targeting them.
- 36-45, 46-55, and 56+ Age Groups:
  - These groups show a clear preference for the Phone channel with probability over 74% for customers aged 36 and older, with the highest being for the 36-45 age group (76%). This suggests that older customers tend to prefer phone calls to interacts with the company.

Contrast with the Overall Probability Distribution of Channel of Sale:

- In the overall distribution, the Phone channel accounts for 51.36% of leads, and the Internet channel makes up 48.64%. However, there is significant difference in preferences based on age: the younger customers (18-35) overwhelmingly prefer the Internet, while older customers (36+) prefer the Phone.

Key Marketing Takeaway:

The marketing department should use tailored approaches to engage different age groups effectively.

- **Digital Focus:** For younger demographics (under 35), online channels, such as email campaigns, social media ads, and website optimization, should be the main focus.
- **Traditional Outreach:** For older demographics, telemarketing or direct phone outreach should be emphasized, as they are much more likely to prefer this method of communication.

Question 1

1.08 Statistical tests – Chi square for independence: task 1

The company wishes to examine the dependence/independence between the choice of CHANNEL OF SALE and the SALES indicator

Perform a Chi square test of independence between CHANNEL OF SALE and the SALES indicator

- **What are the null and alternative hypotheses of this test?**
- **Calculate the P-value of the test and state your conclusion at a 5% significance level and at a 1% significance level**

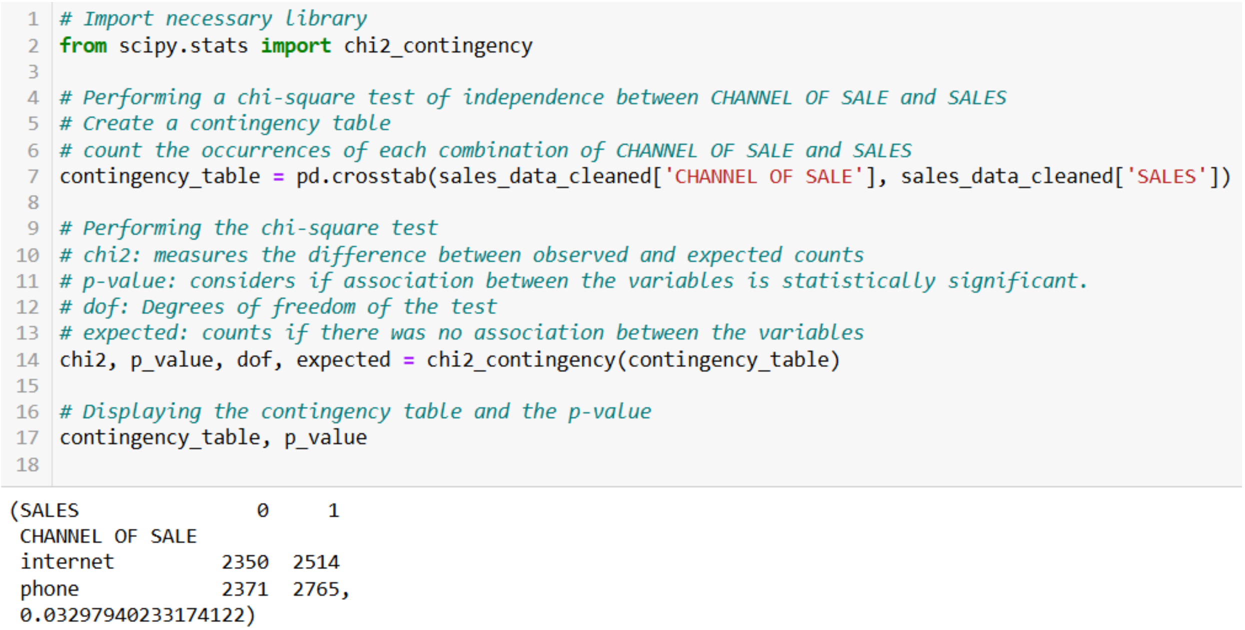
Guidance:

- Create a 2x2 table depicting CHANNEL OF SALE in rows and SALES in columns - the table should present COUNTS of people belonging to each combination (these are your observed values)
- Calculate total counts for rows and columns
- Calculate your expected values (total in row * total in column / overall total)
- Tip: use the CHITEST function

Answer:

Null Hypothesis (H0):
There is no association between the CHANNEL OF SALE and the SALES outcome (they are independent).

Alternative Hypothesis (H1):
There is an association between the CHANNEL OF SALE and the SALES outcome (they are not independent).



```python
# Import necessary library
from scipy.stats import chi2_contingency

# Performing a chi-square test of independence between CHANNEL OF SALE and SALES
# Create a contingency table
# count the occurrences of each combination of CHANNEL OF SALE and SALES
contingency_table = pd.crosstab(sales_data_cleaned['CHANNEL OF SALE'], sales_data_cleaned['SALES'])

# Performing the chi-square test
# chi2: measures the difference between observed and expected counts
# p-value: considers if association between the variables is statistically significant.
# dof: Degrees of freedom of the test
# expected: counts if there was no association between the variables
chi2, p_value, dof, expected = chi2_contingency(contingency_table)

# Displaying the contingency table and the p-value
contingency_table, p_value

```

```
(SALES               0     1
 CHANNEL OF SALE
 internet         2350  2514
 phone            2371  2765,
 0.03297940233174122)
```

At a 5% significance level, since the p-value (0.03298) is less than 0.05, we reject the null hypothesis. This suggests that there is a statistically significant association between the CHANNEL OF SALE and the SALES outcome.

At a 1% significance level, however, the p-value is greater than 0.01, so we fail to reject the null hypothesis. Thus, at this level, there is not enough evidence to conclude a significant association.

Question 1

1.09 Random variables & CLT: task 1

The company wishes to calculate its expected profit:

- It spends 45$ on acquiring each sales lead
- Price of each sale is 86$

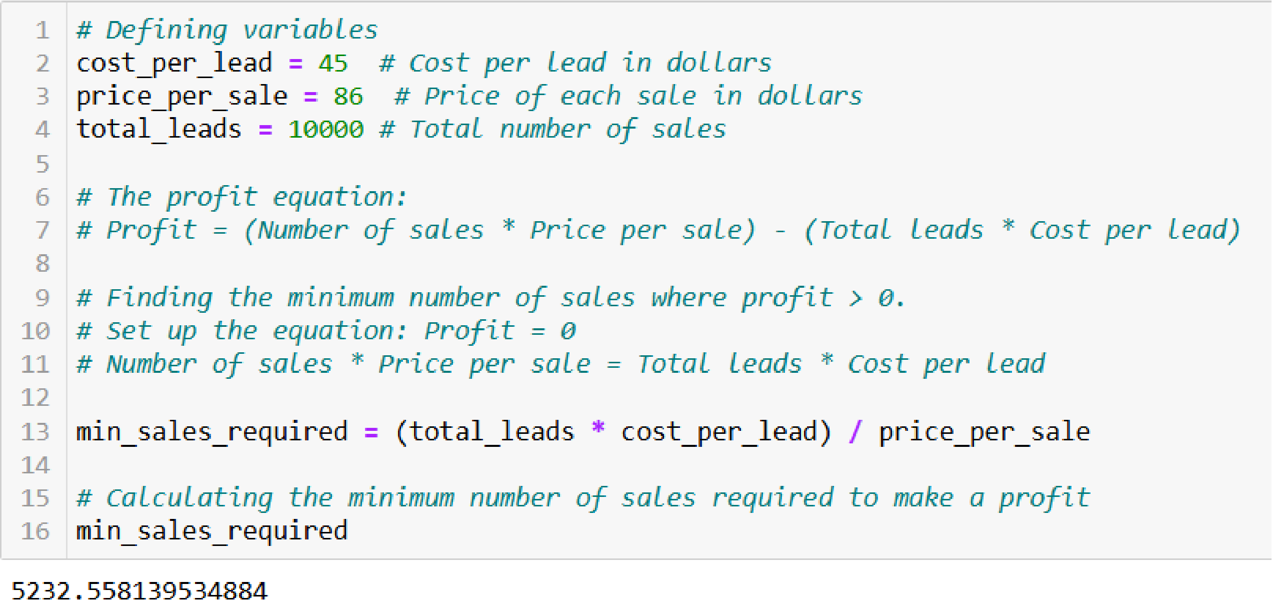
Calculate the minimal number of sales (out of 10,000 leads) required to make a profit

Guidance:

- You may solve this as an equation:
- PROFIT = NUMBER OF SALES x PRICE OF SALE - TOTAL LEADS x COST PER LEAD > 0
- Where only the number of sales is unknown
- Alternatively, you may calculate the profit for each possible number of sales and locate the first positive value

Answer:

The minimum number of sales required to make a profit is approximately 5233 sales (out of 10,000 leads).
At this point, the company's revenue from sales will exceed the total cost of acquiring the leads.



```
# Defining variables
cost_per_lead = 45  # Cost per lead in dollars
price_per_sale = 86  # Price of each sale in dollars
total_leads = 10000 # Total number of sales

# The profit equation:
# Profit = (Number of sales * Price per sale) - (Total leads * Cost per lead)

# Finding the minimum number of sales where profit > 0.
# Set up the equation: Profit = 0
# Number of sales * Price per sale = Total leads * Cost per lead

min_sales_required = (total_leads * cost_per_lead) / price_per_sale

# Calculating the minimum number of sales required to make a profit
min_sales_required
```

```
5232.558139534884
```

Question 1

1.10 Random variables & CLT: task 2

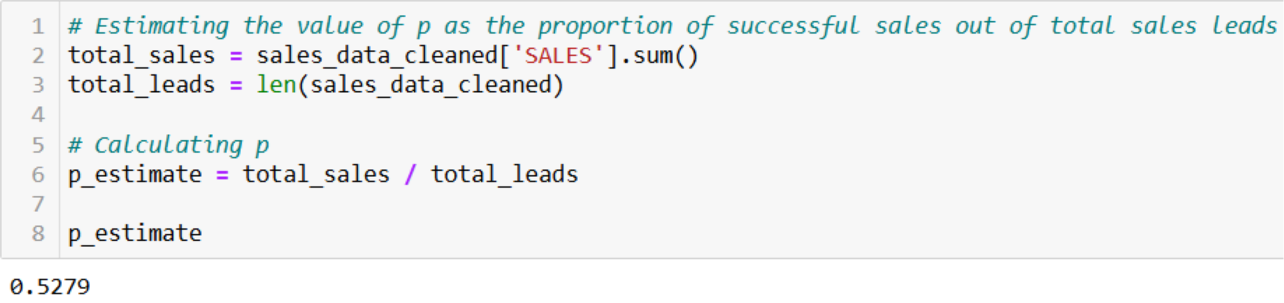
The company decides to model the total amount of sales using a Binomial random variable

- Where n=10,000
- And p is estimated from the SALES column

Estimate the value of p

Guidance:

- Estimate p as the proportion of sales out of total sales leads
- Count how many sales there were, and divide by the total number of leads



```
# Estimating the value of p as the proportion of successful sales out of total sales leads
total_sales = sales_data_cleaned['SALES'].sum()
total_leads = len(sales_data_cleaned)

# Calculating p
p_estimate = total_sales / total_leads

p_estimate
```

```
0.5279
```

Answer:

The estimated value of p, which represents the proportion of successful sales out of the total leads, is 0.5279.
This means that approximately 52.79% of the leads result in successful sales.

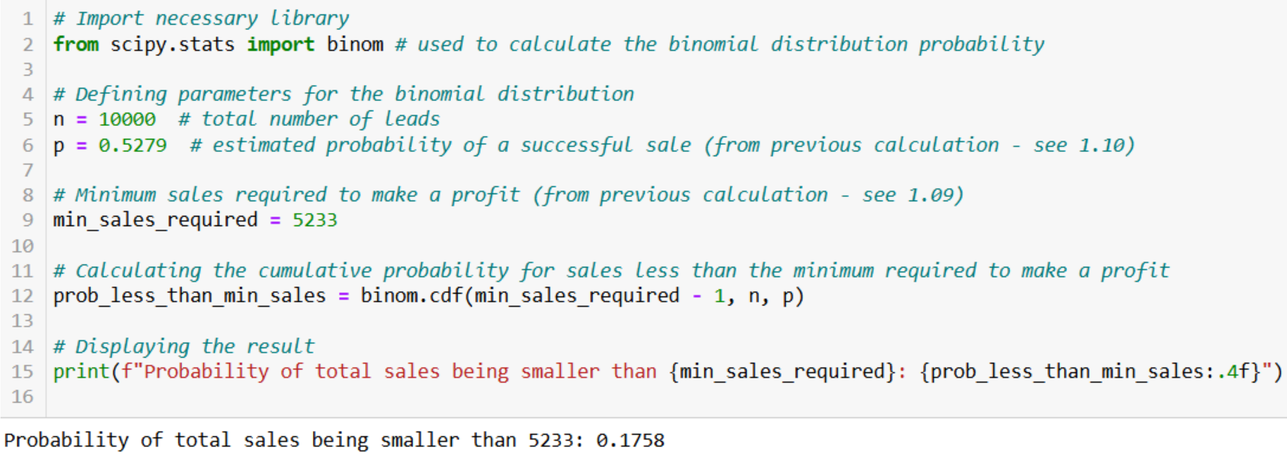
Question 1

1.11 Random variables & CLT: task 3

Using the Binomial random variable, calculate the probability of the total amount of sales being smaller than the number required to make a profit

Guidance:

- Use the Binom.dist function
- For each possible level of sales (0-10,000) - denoted by x, calculate the probability function: Binom.dist(x,n,p,0)
- Make sure the column adds up to 1, as a sanity check
- Sum over the probabilities where the profit value is negative (those smaller than the minimal number required to make a profit)



```python
# Import necessary library
from scipy.stats import binom # used to calculate the binomial distribution probability

# Defining parameters for the binomial distribution
n = 10000  # total number of leads
p = 0.5279  # estimated probability of a successful sale (from previous calculation - see 1.10)

# Minimum sales required to make a profit (from previous calculation - see 1.09)
min_sales_required = 5233

# Calculating the cumulative probability for sales less than the minimum required to make a profit
prob_less_than_min_sales = binom.cdf(min_sales_required - 1, n, p)

# Displaying the result
print(f"Probability of total sales being smaller than {min_sales_required}: {prob_less_than_min_sales:.4f}")

```

```
Probability of total sales being smaller than 5233: 0.1758
```

Answer:

- There is a 17.58% chance that the company will fail to make a profit from the 10,000 leads based on the estimated probability of a successful sale (52.79%).
- In other words, while the company has a relatively good chance of turning a profit (around 82.42%), there is still a non-negligible risk (17.58%) that the sales will fall short of the profit threshold.

Question 1

1.12 Random variables & CLT: task 4

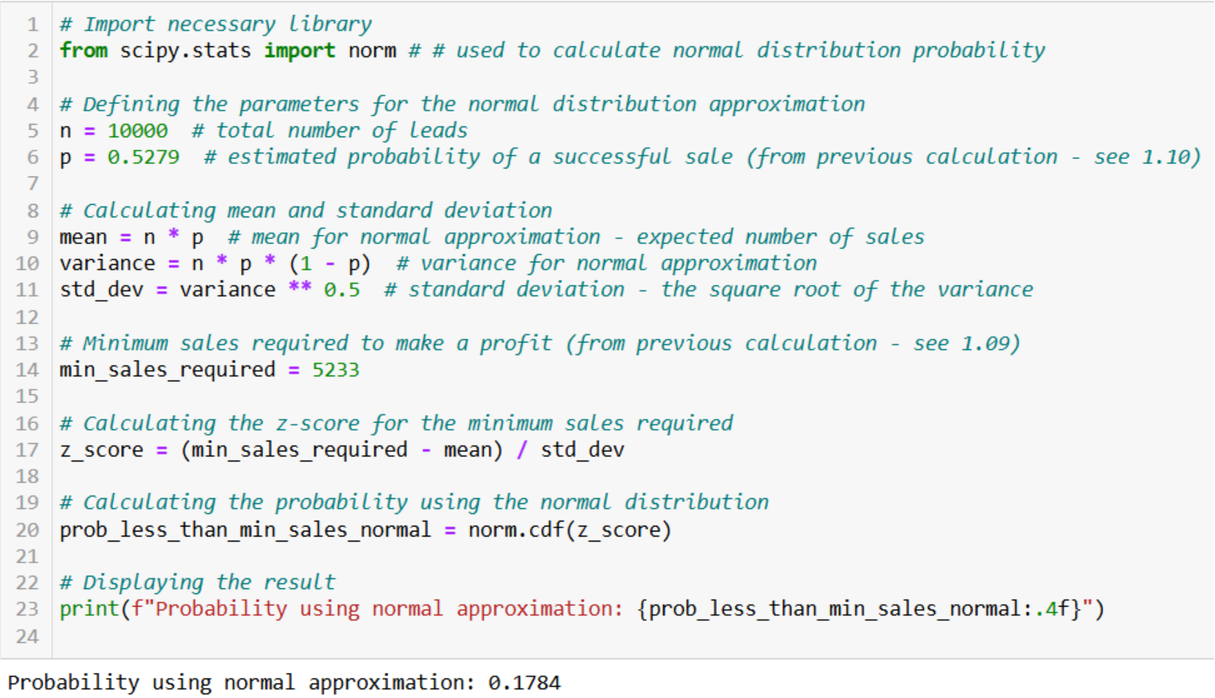
Using the Normal approximation of the Binomial random variable, calculate the same probability

Guidance:

- The CLT allows us to approximate the distribution of a SUM of random variables as a normal distribution
- The Binomial distribution function is a SUM of independent & identically distributed Bernoulli random variables - therefore is eligible for use of the CLT
- In this case you can approximate the Binomial distribution as a Normal distribution with:
  - Mean = np
  - Variance = np(1-p)
- Use the Norms.dist function: Norms.dist(x, mean, standard deviation, 1)
- Notice that the function required standard deviation and not variance

Answer:

- Using the normal approximation (based on the Central Limit Theorem), there is a **17.84% chance** that the total number of sales will fall short of the 5233 threshold (see 1.09) needed to make a profit.
- This result is very close to the 17.58% probability calculated using the exact binomial distribution, showing that the normal approximation provides a reasonable estimate for this scenario.
- This further supports the idea that the company faces about a 17-18% risk of not reaching the break-even point in sales, based on the current success rate (52.79%).



```python
# Import necessary library
from scipy.stats import norm # # used to calculate normal distribution probability

# Defining the parameters for the normal distribution approximation
n = 10000  # total number of leads
p = 0.5279  # estimated probability of a successful sale (from previous calculation - see 1.10)

# Calculating mean and standard deviation
mean = n * p  # mean for normal approximation - expected number of sales
variance = n * p * (1 - p)  # variance for normal approximation
std_dev = variance ** 0.5  # standard deviation - the square root of the variance

# Minimum sales required to make a profit (from previous calculation - see 1.09)
min_sales_required = 5233

# Calculating the z-score for the minimum sales required
z_score = (min_sales_required - mean) / std_dev

# Calculating the probability using the normal distribution
prob_less_than_min_sales_normal = norm.cdf(z_score)

# Displaying the result
print(f"Probability using normal approximation: {prob_less_than_min_sales_normal:.4f}")

```

```
Probability using normal approximation: 0.1784
```

Question 1

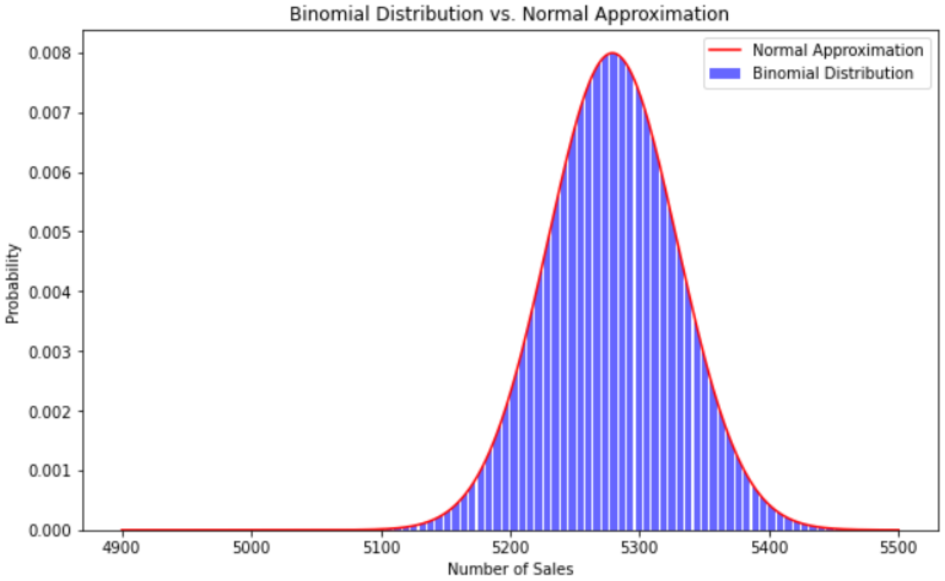
1.13 Random variables & CLT: task 5

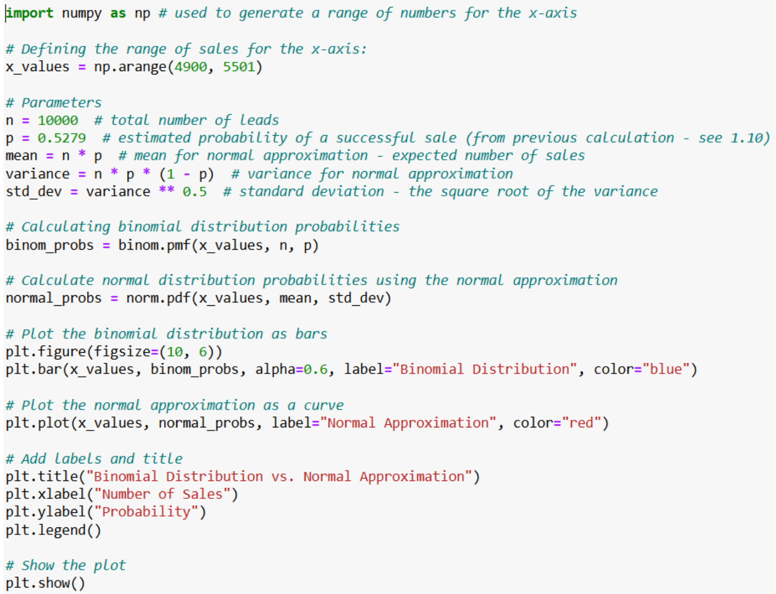
Plot the two distributions together - Binomial as bars, Normal as line (you may choose the X-axis range which is best suitable for your plot). Comment on the appropriateness of the approximation

Guidance:

- The Binomial distribution is discrete and should be presented using a bar-plot
- The Normal distribution is continuous and should be presented using a curve
- Calculate Binom.dist(x,n,p,0) and Norms.dist(x, mean, standard deviation, 0) for each possible value of sales (make sure the probabilities in each column add up to 1)
- You may plot them together for ease of comparison or separately

Answer:





```python
import numpy as np # used to generate a range of numbers for the x-axis

# Defining the range of sales for the x-axis:
x_values = np.arange(4900, 5501)

# Parameters
n = 10000  # total number of leads
p = 0.5279  # estimated probability of a successful sale (from previous calculation - see 1.10)
mean = n * p  # mean for normal approximation - expected number of sales
variance = n * p * (1 - p)  # variance for normal approximation
std_dev = variance ** 0.5  # standard deviation - the square root of the variance

# Calculating binomial distribution probabilities
binom_probs = binom.pmf(x_values, n, p)

# Calculate normal distribution probabilities using the normal approximation
normal_probs = norm.pdf(x_values, mean, std_dev)

# Plot the binomial distribution as bars
plt.figure(figsize=(10, 6))
plt.bar(x_values, binom_probs, alpha=0.6, label="Binomial Distribution", color="blue")

# Plot the normal approximation as a curve
plt.plot(x_values, normal_probs, label="Normal Approximation", color="red")

# Add labels and title
plt.title("Binomial Distribution vs. Normal Approximation")
plt.xlabel("Number of Sales")
plt.ylabel("Probability")
plt.legend()

# Show the plot
plt.show()
```

- **Binomial Distribution:** This is plotted as blue bars, representing the discrete probabilities for the number of sales.

- **Normal Approximation:** This is plotted as a red curve, representing the continuous normal approximation to the binomial distribution.

- The plot help to compare how well the normal approximation fits the binomial distribution. As typically, the approximation is quite accurate when the number of trials is large, as is the case here with 10,000 leads.

Question 1

1.14 Random variables & CLT: task 6

The risk manager said: “if we have 0 sales, we will lose $450,000 which is very risky for a new product!”. Comment on his statement

Guidance:

- What is the risk manager saying is correct? Is it **possible** to reach such a value? Is it **probable**?

Answer:

To evaluate the risk manager’s statement regarding the potential loss of $450,000 if the company makes zero sales, it should be breaken down into the following points:

1. **Is the Statement Correct?**

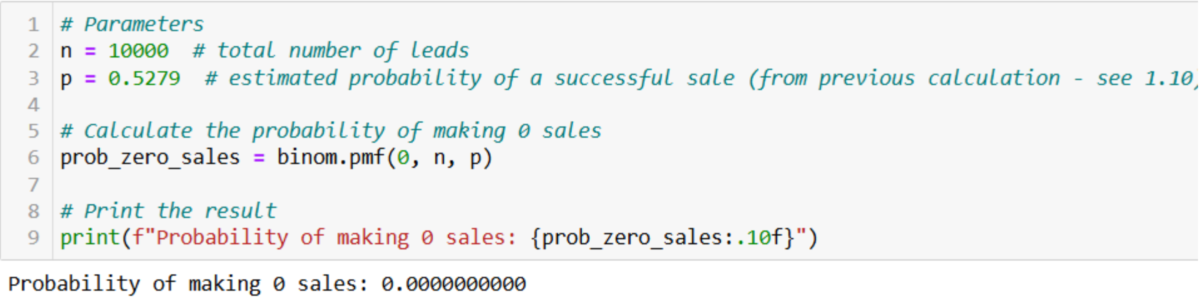
   Cost Calculation: The company spends 45$ per lead, and with 10,000 leads, the total cost would indeed be:

   $\mathit{Total\ Cost} = 10{,}000 \times 45 = 450{,}000$

   If no sales are made, the company would lose the entire 450,000$ spent on acquiring the leads. Therefore, **the calculation is correct.**

2. **Is it Possible to Make 0 Sales?**

   Theoretically, it is possible, though highly unlikely. In a binomial distribution, even the extreme case of 0 sales has some non-zero probability (because the binomial distribution accounts for all possibilities from 0 to 10,000 sales).

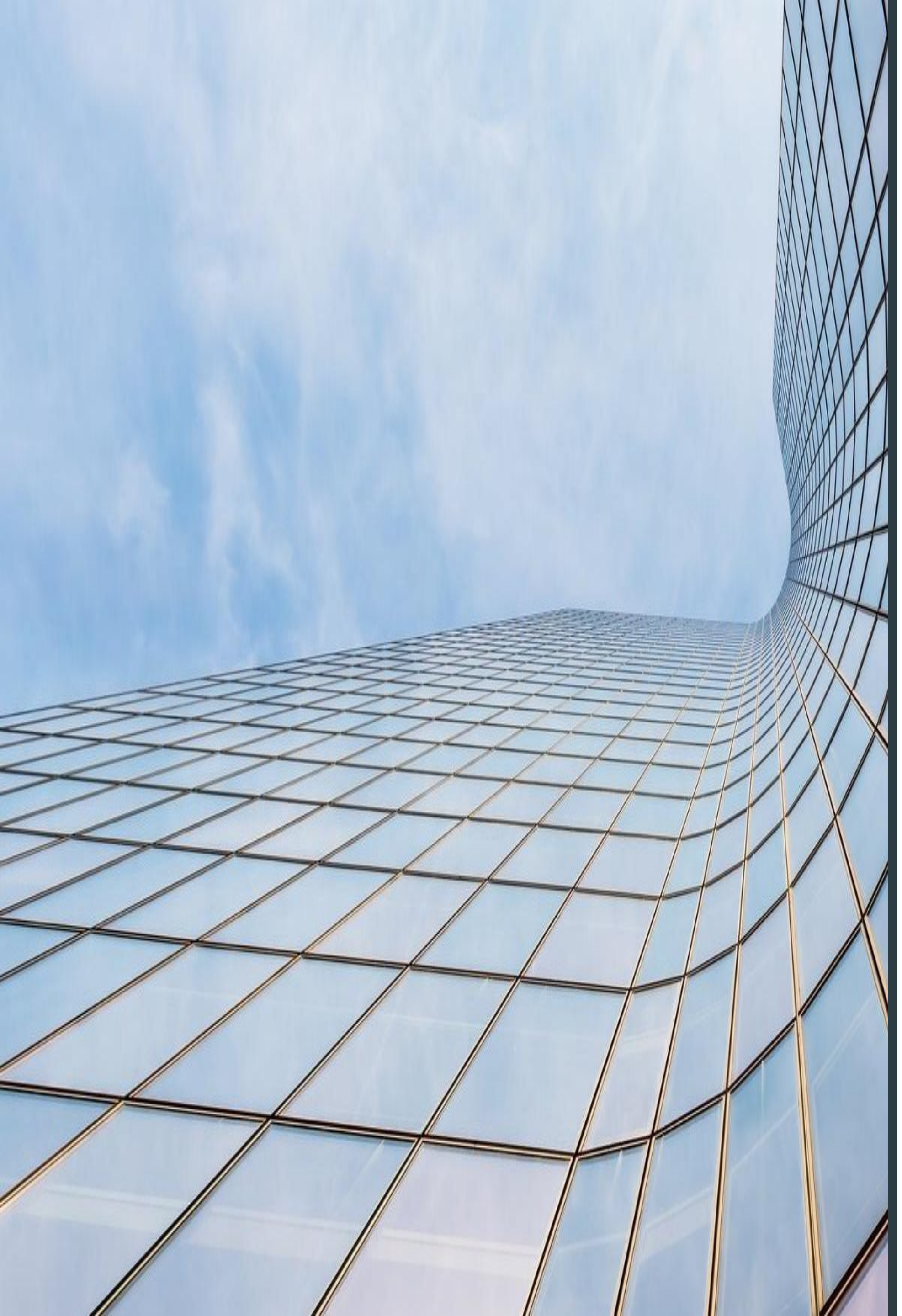


```
# Parameters
n = 10000  # total number of leads
p = 0.5279  # estimated probability of a successful sale (from previous calculation - see 1.10,

# Calculate the probability of making 0 sales
prob_zero_sales = binom.pmf(0, n, p)

# Print the result
print(f"Probability of making 0 sales: {prob_zero_sales:.10f}")
```

```
Probability of making 0 sales: 0.0000000000
```

3. **Is it Probable?**

   - Calculating the probability of making zero sales using the binomial distribution, where:
     - n = 10000 - total number of leads
     - p = 0.5279 - estimated probability of a successful sale (from previous calculation - see 1.10)
   - Using the binomial probability mass function (PMF) to calculate the probability of making 0 sales:
     The probability of making zero sales will be extremely close to zero, given the large number of leads (10,000) and the relatively high probability of success (52.79%). Therefore, while theoretically possible, it is **highly improbable.**



Question 2

Use the sheet named “RAND” for the following questions:

Simulation

- The company decided to perform 100 simulations of its potential profit using a Binomial random variable (10,000 Bernoulli variables)
- In the sheet “RAND” you will find 10000x100 random variables from a Uniform(0,1) distribution

Question 2
2.01 Simulation: task 1

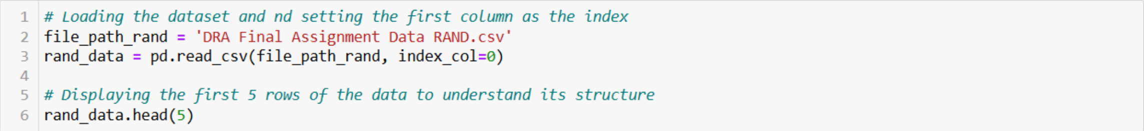
Perform 100 simulations of the possible total profit under these 100 simulations

- **Any number smaller than the estimator for p should be 1 (considered a sale), otherwise it should be 0 (considered no sale)**

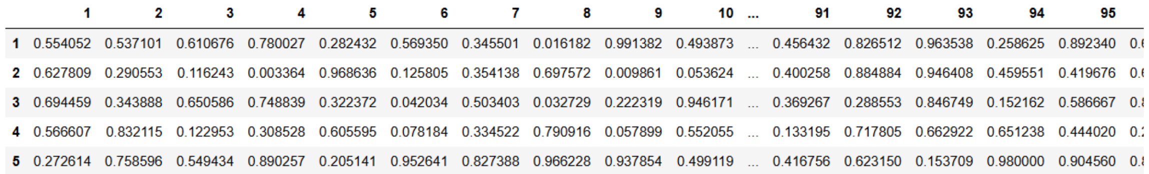
Guidance:

- We want to simulate values in similar proportions to our original sample, where the sales proportion was p.
- If we have these uniform (0,1) random variables, and for example p is 60%, if we decide that any number simulated which is smaller than 0.6 is considered as a sale, we are likely to get very similar proportions of sales
- Tip: use IF conditions

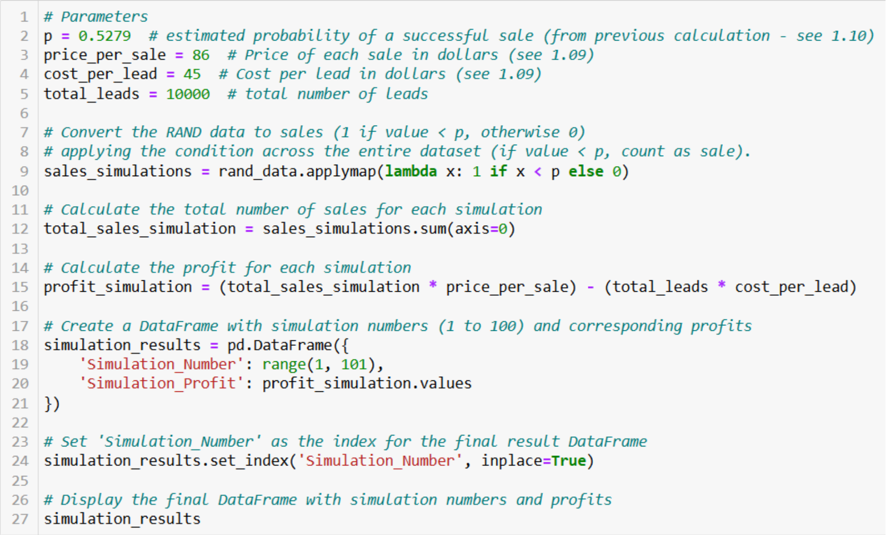
Answer:



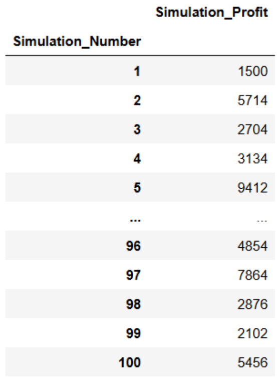
```python
# Loading the dataset and nd setting the first column as the index
file_path_rand = 'DRA Final Assignment Data RAND.csv'
rand_data = pd.read_csv(file_path_rand, index_col=0)

# Displaying the first 5 rows of the data to understand its structure
rand_data.head(5)
```



	1	2	3	4	5	6	7	8	9	10	...	91	92	93	94	95
1	0.554052	0.537101	0.610676	0.780027	0.282432	0.569350	0.345501	0.016182	0.991382	0.493873	...	0.456432	0.826512	0.963538	0.258625	0.892340
2	0.627809	0.290553	0.116243	0.003364	0.968636	0.125805	0.354138	0.697572	0.009861	0.053624	...	0.400258	0.884884	0.946408	0.459551	0.419676
3	0.694459	0.343888	0.650586	0.748839	0.322372	0.042034	0.503403	0.032729	0.222319	0.946171	...	0.369267	0.288553	0.846749	0.152162	0.586667
4	0.566607	0.832115	0.122953	0.308528	0.605595	0.078184	0.334522	0.790916	0.057899	0.552055	...	0.133195	0.717805	0.662922	0.651238	0.444020
5	0.272614	0.758596	0.549434	0.890257	0.205141	0.952641	0.827388	0.966228	0.937854	0.499119	...	0.416756	0.623150	0.153709	0.980000	0.904560



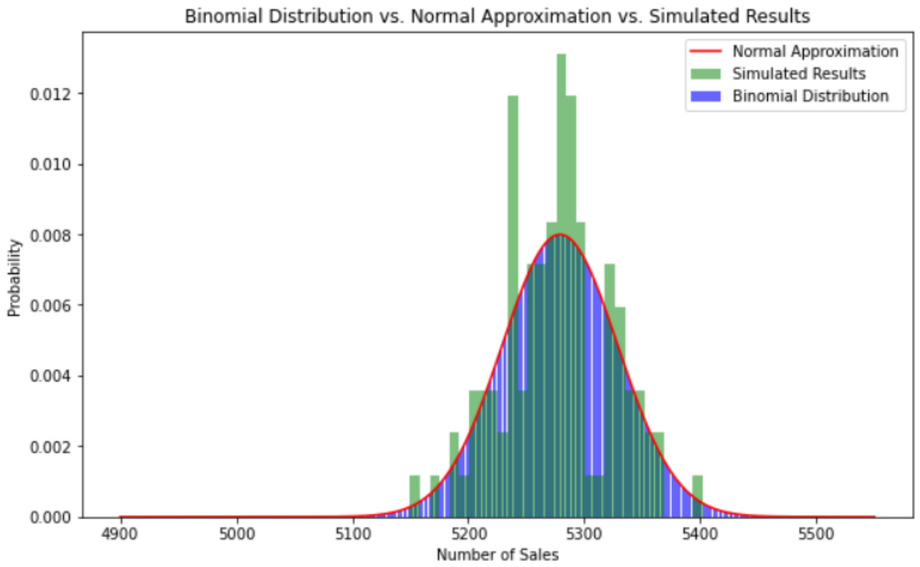
```python
# Parameters
p = 0.5279  # estimated probability of a successful sale (from previous calculation - see 1.10)
price_per_sale = 86  # Price of each sale in dollars (see 1.09)
cost_per_lead = 45  # Cost per lead in dollars (see 1.09)
total_leads = 10000  # total number of leads

# Convert the RAND data to sales (1 if value < p, otherwise 0)
# applying the condition across the entire dataset (if value < p, count as sale).
sales_simulations = rand_data.applymap(lambda x: 1 if x < p else 0)

# Calculate the total number of sales for each simulation
total_sales_simulation = sales_simulations.sum(axis=0)

# Calculate the profit for each simulation
profit_simulation = (total_sales_simulation * price_per_sale) - (total_leads * cost_per_lead)

# Create a DataFrame with simulation numbers (1 to 100) and corresponding profits
simulation_results = pd.DataFrame({
    'Simulation_Number': range(1, 101),
    'Simulation_Profit': profit_simulation.values
})

# Set 'Simulation_Number' as the index for the final result DataFrame
simulation_results.set_index('Simulation_Number', inplace=True)

# Display the final DataFrame with simulation numbers and profits
simulation_results
```

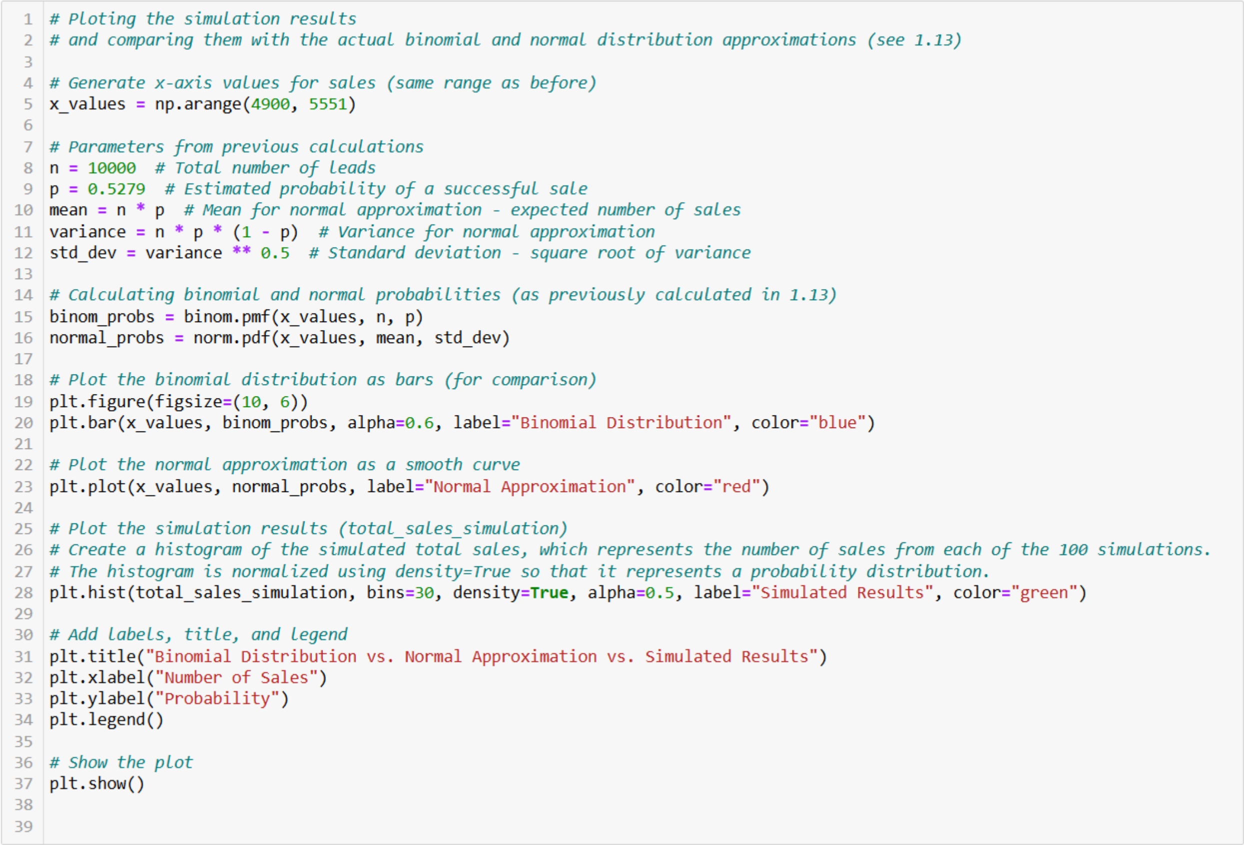


	Simulation_Profit
Simulation_Number	
1	1500
2	5714
3	2704
4	3134
5	9412
...	...
96	4854
97	7864
98	2876
99	2102
100	5456

Question 2

2.01 Simulation: task 1





```
# Ploting the simulation results
# and comparing them with the actual binomial and normal distribution approximations (see 1.13)

# Generate x-axis values for sales (same range as before)
x_values = np.arange(4900, 5551)

# Parameters from previous calculations
n = 10000  # Total number of leads
p = 0.5279  # Estimated probability of a successful sale
mean = n * p  # Mean for normal approximation - expected number of sales
variance = n * p * (1 - p)  # Variance for normal approximation
std_dev = variance ** 0.5  # Standard deviation - square root of variance

# Calculating binomial and normal probabilities (as previously calculated in 1.13)
binom_probs = binom.pmf(x_values, n, p)
normal_probs = norm.pdf(x_values, mean, std_dev)

# Plot the binomial distribution as bars (for comparison)
plt.figure(figsize=(10, 6))
plt.bar(x_values, binom_probs, alpha=0.6, label="Binomial Distribution", color="blue")

# Plot the normal approximation as a smooth curve
plt.plot(x_values, normal_probs, label="Normal Approximation", color="red")

# Plot the simulation results (total_sales_simulation)
# Create a histogram of the simulated total sales, which represents the number of sales from each of the 100 simulations.
# The histogram is normalized using density=True so that it represents a probability distribution.
plt.hist(total_sales_simulation, bins=30, density=True, alpha=0.5, label="Simulated Results", color="green")

# Add labels, title, and legend
plt.title("Binomial Distribution vs. Normal Approximation vs. Simulated Results")
plt.xlabel("Number of Sales")
plt.ylabel("Probability")
plt.legend()

# Show the plot
plt.show()
```

Key Observations:

- The green bars represent the distribution of the simulated sales based on the 100 given simulations.
- These bars follow the general shape of the binomial distribution and the normal approximation of actual results. However, there are some peaks and gaps, which are due to the limited number of simulations (100 simulations), resulting in some variability.
- The slight difference between the Normal Approximation(Red Line) and the green bars, especially at the peaks and tails, highlights where the normal approximation smooths over the discrete nature of the binomial distribution.

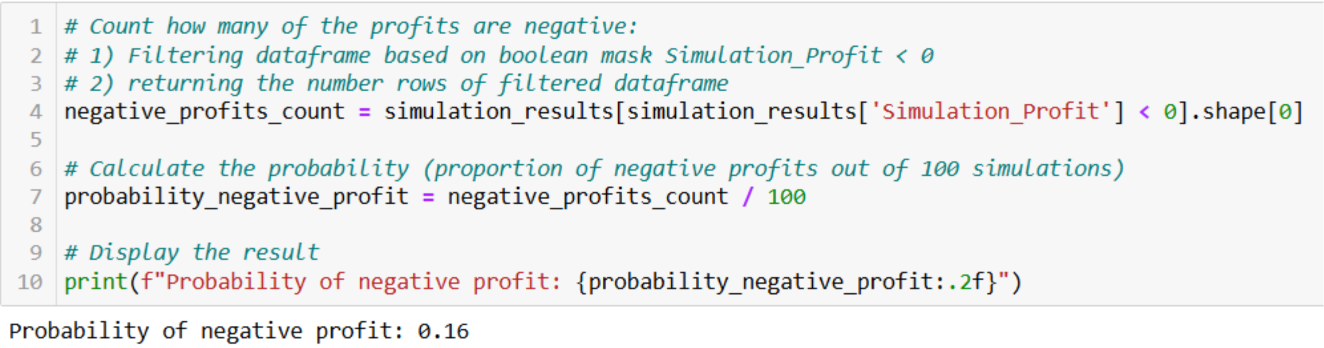
Question 2

2.02 Simulation: task 2

Estimate the probability of the profit being negative

Guidance:

- You are performing 100 simulations
- This means that you are generating 100 possible scenarios of profits
- For each simulation, count the number of sales (out of 10,000), and calculate the profit for this amount of sales (using the profit formula from before)
- Once you have calculated 100 profits, count the proportion of negative profits (out of 100)



```python
# Count how many of the profits are negative:
# 1) Filtering dataframe based on boolean mask Simulation_Profit < 0
# 2) returning the number rows of filtered dataframe
negative_profits_count = simulation_results[simulation_results['Simulation_Profit'] < 0].shape[0]

# Calculate the probability (proportion of negative profits out of 100 simulations)
probability_negative_profit = negative_profits_count / 100

# Display the result
print(f"Probability of negative profit: {probability_negative_profit:.2f}")
```

```
Probability of negative profit: 0.16
```

Answer:

- Based on the result from the 100 simulations, the profit was negative in 16% of those simulations.

- The simulated probability of negative profit (16%) is slightly lower than the exact binomial probability (17.58%, see 1.11)

- This small difference (1.58%) is likely due to the limited number of simulations (100), which introduces variability and may not fully capture the exact binomial distribution.

Question 2

2.03 Simulation: task 3

Comment on your result in relation to your answer using the Binomial and Normal probabilities from the previous section

Guidance:

- Are your answers close? If not, what could you recommend that would make the simulation more accurate?

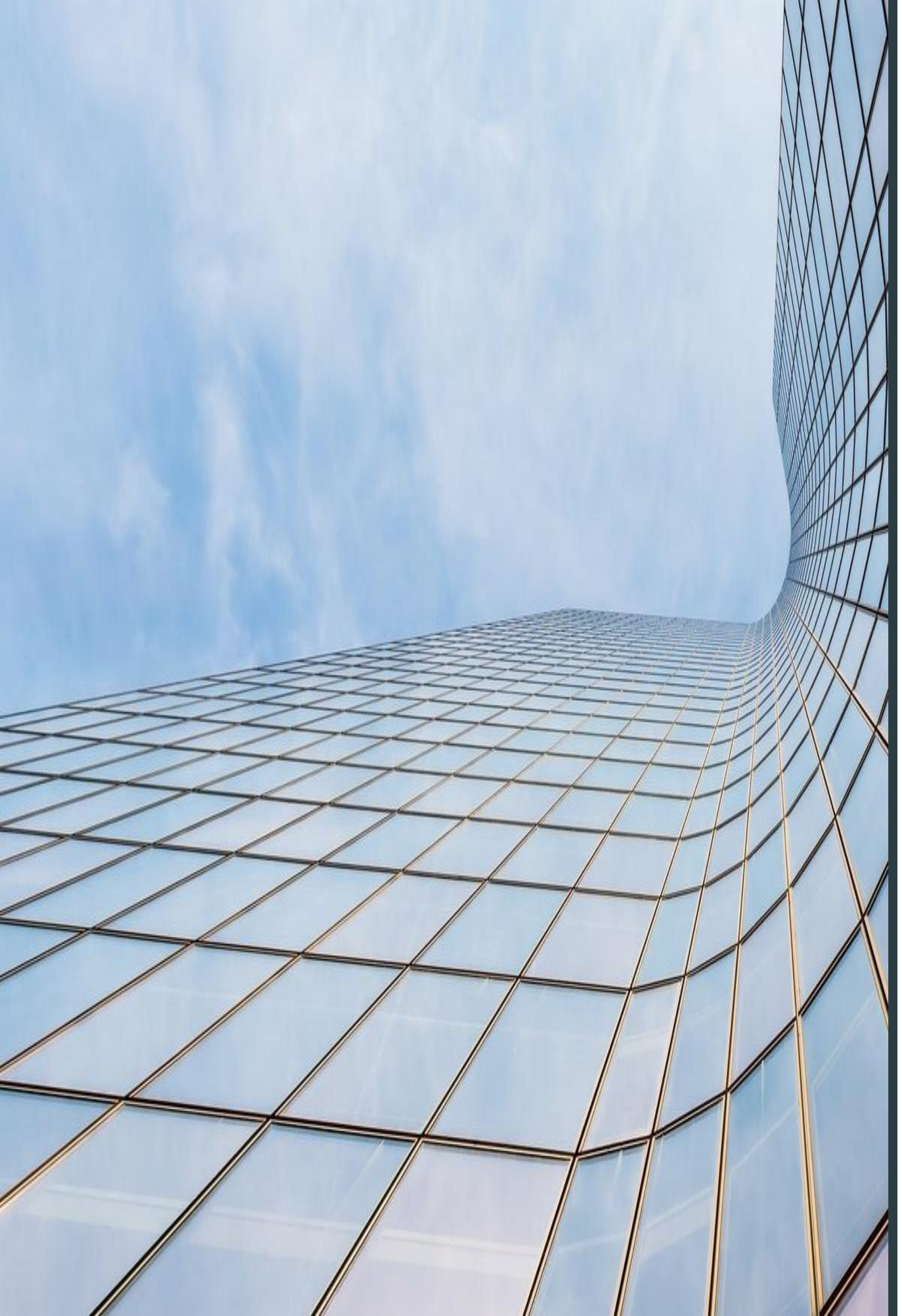
Answer:

- **Accuracy:**
  The simulation results are reasonably accurate, as they align closely with the binomial distribution.
  The slight discrepancies are due to the number of simulations (100), which introduces some variability. Running more simulations (e.g., 1,000 or more) would likely result in an even closer match.

- **Recommendation:**
  To increase the accuracy of the simulation and reduce the variability seen in the simulation results could be recommended following:

  - Increasing the number of simulations to better capture the underlying binomial distribution.

  - Considering using a continuity correction for the normal approximation to refine the comparison further.



Question 3

Use the sheet named “AB TESTING” for the following questions:

Time series & A/B Testing

The company decided to launch a media campaign to increase leads:

- It launched an extensive media campaign during January and February of 2022
- It tracked daily sales leads for 4 months, to assess the prolonging effect of the campaign

Question 3

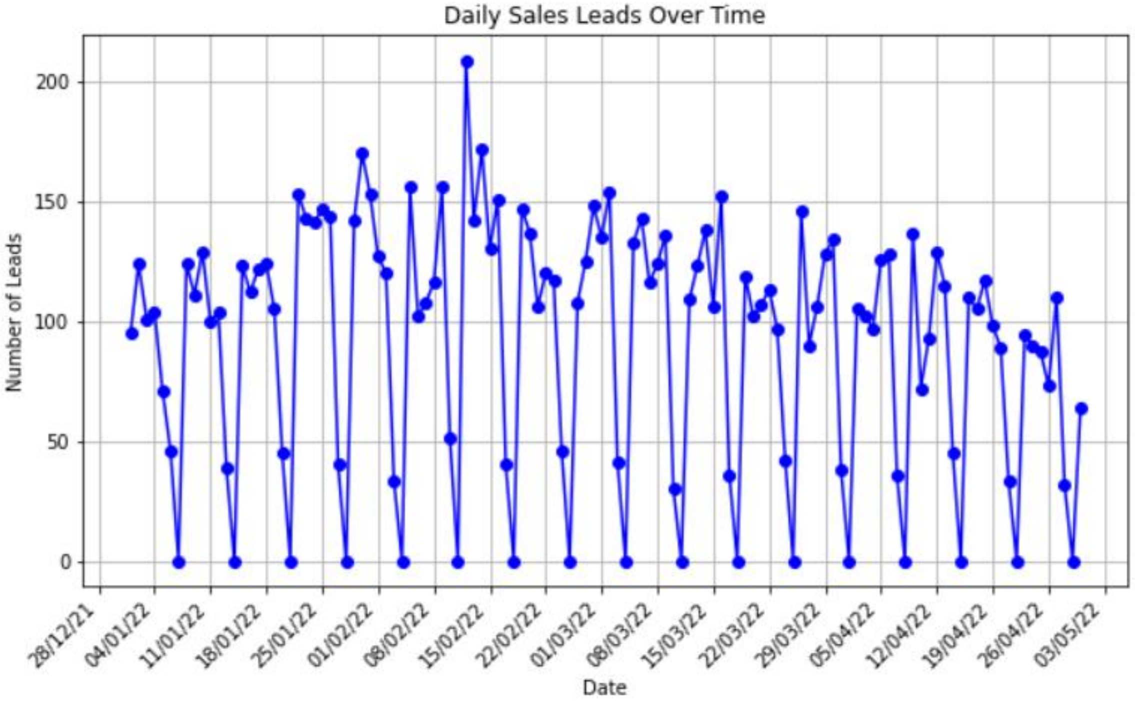
3.01 Time series & A/B Testing: task 1

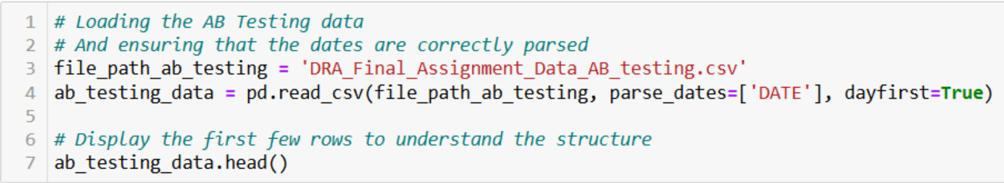
Plot the daily sales leads over time

Guidance:

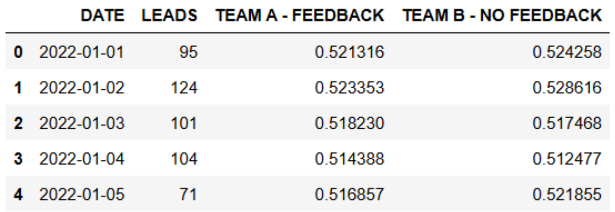
- We X-axis should be days
- Y-axis should be count of daily leads

Answer:

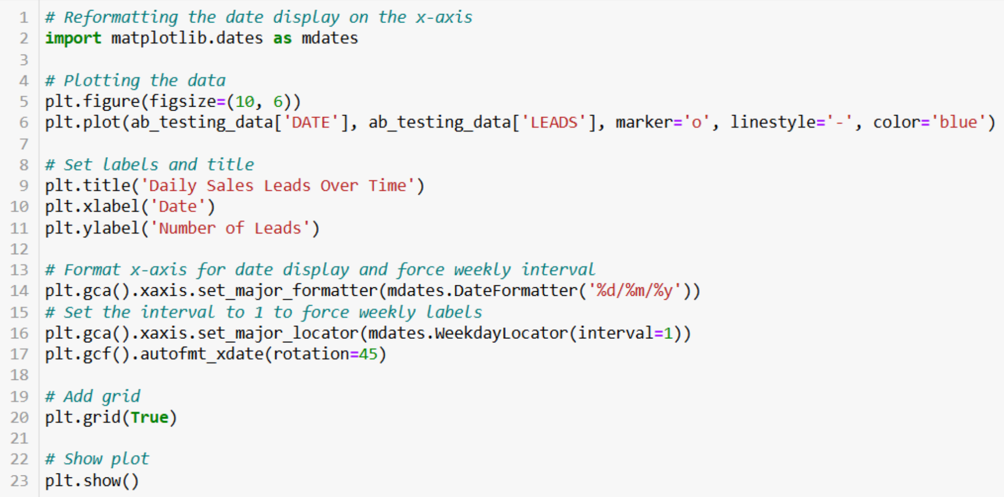




```python
# Loading the AB Testing data
# And ensuring that the dates are correctly parsed
file_path_ab_testing = 'DRA_Final_Assignment_Data_AB_testing.csv'
ab_testing_data = pd.read_csv(file_path_ab_testing, parse_dates=['DATE'], dayfirst=True)

# Display the first few rows to understand the structure
ab_testing_data.head()
```



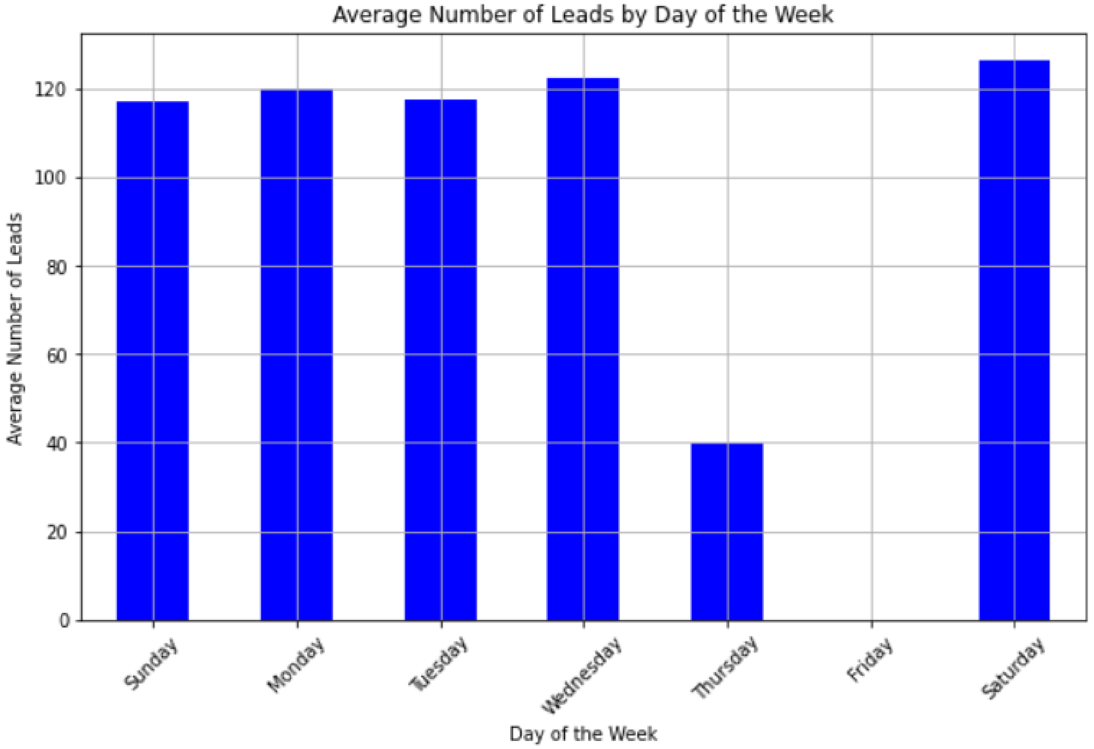
	DATE	LEADS	TEAM A - FEEDBACK	TEAM B - NO FEEDBACK
0	2022-01-01	95	0.521316	0.524258
1	2022-01-02	124	0.523353	0.528616
2	2022-01-03	101	0.518230	0.517468
3	2022-01-04	104	0.514388	0.512477
4	2022-01-05	71	0.516857	0.521855

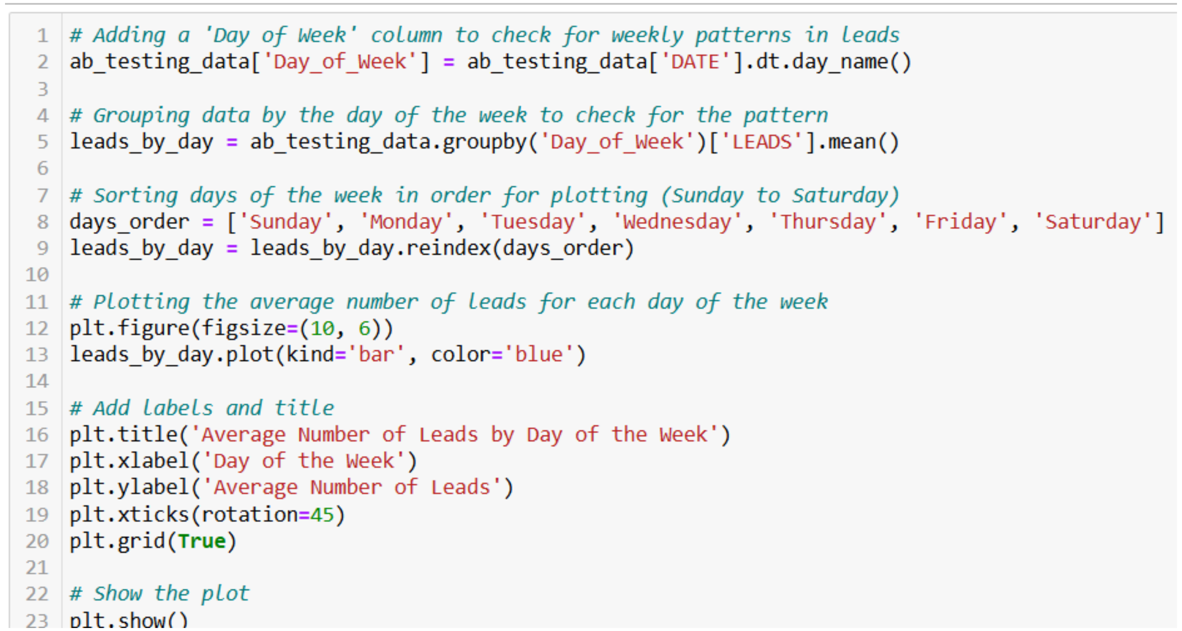


```python
# Reformatting the date display on the x-axis
import matplotlib.dates as mdates

# Plotting the data
plt.figure(figsize=(10, 6))
plt.plot(ab_testing_data['DATE'], ab_testing_data['LEADS'], marker='o', linestyle='-', color='blue')

# Set labels and title
plt.title('Daily Sales Leads Over Time')
plt.xlabel('Date')
plt.ylabel('Number of Leads')

# Format x-axis for date display and force weekly interval
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%d/%m/%y'))
# Set the interval to 1 to force weekly labels
plt.gca().xaxis.set_major_locator(mdates.WeekdayLocator(interval=1))
plt.gcf().autofmt_xdate(rotation=45)

# Add grid
plt.grid(True)

# Show plot
plt.show()
```

Question 3

3.01 Time series & A/B Testing: task 1





```python
# Adding a 'Day of Week' column to check for weekly patterns in leads
ab_testing_data['Day_of_Week'] = ab_testing_data['DATE'].dt.day_name()

# Grouping data by the day of the week to check for the pattern
leads_by_day = ab_testing_data.groupby('Day_of_Week')['LEADS'].mean()

# Sorting days of the week in order for plotting (Sunday to Saturday)
days_order = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday']
leads_by_day = leads_by_day.reindex(days_order)

# Plotting the average number of leads for each day of the week
plt.figure(figsize=(10, 6))
leads_by_day.plot(kind='bar', color='blue')

# Add labels and title
plt.title('Average Number of Leads by Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Average Number of Leads')
plt.xticks(rotation=45)
plt.grid(True)

# Show the plot
plt.show()
```

Question 3

3.02 Time series & A/B Testing: task 2

Comment on the existence of:

- **Trends**
- **Seasonality**
- **Random variability**
- **Irregularities & Outliers**

Guidance:

- Trend - overall direction of the data over time (could have more than one trend)
- Seasonality - a recurring pattern of predictable length
- Random variability - random deviations from the "perfect" model
- Irregularities & outliers - extreme or erroneous observations

Answer:

- **Trends:**
  - The Daily Sales Leads Over Time plot shows some fluctuations, but there is a slight downward trend from the peak leads in February through to the end of April:
    the general level of daily sales leads appears to be decreasing over time.
  - There are multiple smaller trends, such as spikes in January and February, possibly due to the media campaign, followed by a gradual decline.
- **Seasonality:**
  - The Average Number of Leads by Day of the Week plot clearly shows a weekly seasonality pattern: There is a significant drop in leads every Thursday and Friday, followed by recovery on Saturday. Leads are relatively stable from Monday to Wednesday. This suggests that the business or consumer behavior may be influenced by the day of the week, with potential drops in sales towards the end of the workweek.
- **Random Variability:**
  - Random deviations are visible in both plots. The daily sales leads have day-to-day fluctuations that do not follow a clear predictable pattern, especially within weeks. These variations may be due to external factors or random noise in the data. The second plot also shows some level of random variability in the average number of leads for each day of the week, though the pattern is more consistent.
- **Irregularities & Outliers:**
  - There are some clear outliers where the number of leads spikes significantly (e.g., peaks around beginning and mid-February), as well as sudden drops to low numbers. These outliers might be due to special promotions, system issues, or other events not regularly occurring.

Question 3

3.03 Time series & A/B Testing: task 3

Summarize your conclusions regarding the success of the media campaign

Guidance:

- Has the campaign been successful over time? Are there seasonal effects that should be disregarded or taken into account?

Answer:

- The **media campaign was initially successful**, as it generated significant spikes in leads during January and February.
- However, the **sustained impact was limited**, and the number of leads gradually declined after the campaign period.
- The **weekly seasonality** (Thursday and Friday drops) should be considered in future evaluations to avoid misinterpreting natural fluctuations as campaign-related issues.
- To maintain or improve lead generation, the company may need to **supplement the campaign with additional strategies** or new promotions to extend its effects over a longer period.

Question 3
3.04 Time series & A/B Testing: task 4

The company has also decided to improve sales rates by having sales managers review the sales team's performance every 2 weeks, starting January 2022.

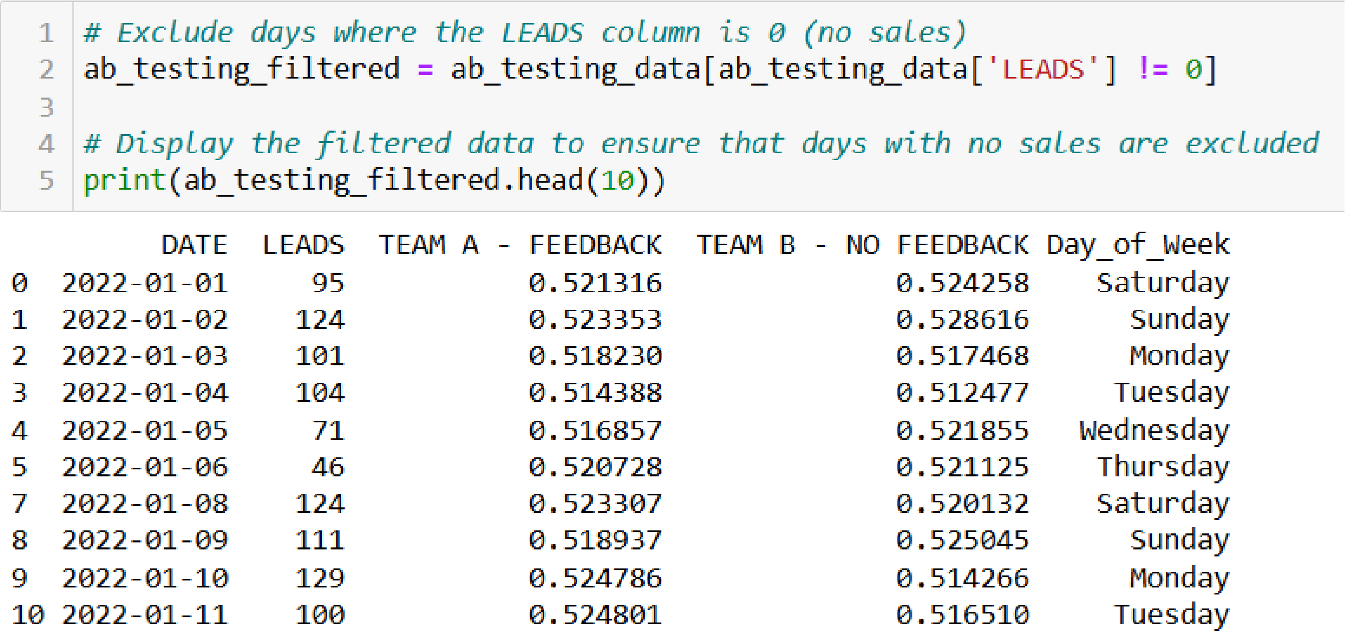
The managers will listen to past recorded calls with potential customers, give feedback and present their conclusions regarding possible improvements in the sales process to the staff every 2 weeks.

In order to test the effect of the potential improvement in sales practices the company has decided to conduct an A/B test, where half of the staff will receive feedback, and the other half will not.

Exclude days where the company is not working and there are no sales

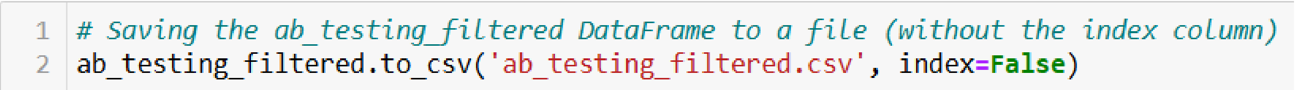
Guidance:

- Days with no sales are the one where the LEADS column is 0



```python
# Exclude days where the LEADS column is 0 (no sales)
ab_testing_filtered = ab_testing_data[ab_testing_data['LEADS'] != 0]

# Display the filtered data to ensure that days with no sales are excluded
print(ab_testing_filtered.head(10))
```

```
          DATE  LEADS  TEAM A - FEEDBACK  TEAM B - NO FEEDBACK Day_of_Week
0   2022-01-01     95           0.521316              0.524258    Saturday
1   2022-01-02    124           0.523353              0.528616      Sunday
2   2022-01-03    101           0.518230              0.517468      Monday
3   2022-01-04    104           0.514388              0.512477     Tuesday
4   2022-01-05     71           0.516857              0.521855   Wednesday
5   2022-01-06     46           0.520728              0.521125    Thursday
7   2022-01-08    124           0.523307              0.520132    Saturday
8   2022-01-09    111           0.518937              0.525045      Sunday
9   2022-01-10    129           0.524786              0.514266      Monday
10  2022-01-11    100           0.524801              0.516510     Tuesday
```



```python
# Saving the ab_testing_filtered DataFrame to a file (without the index column)
ab_testing_filtered.to_csv('ab_testing_filtered.csv', index=False)
```

Question 3

3.05 Time series & A/B Testing: task 5

Suggest a way of selecting which sales representative goes in which group. What biases is the company trying to prevent?

Guidance:

- Think what makes a good experiment, you may give examples of “bad” ways of performing this experiment to illustrate your point further

Answer

- **Recommended Approach:**
  - To prevent biases and ensure a valid A/B test, randomly assigning sales representatives is the best approach. This will help ensure that the groups are comparable, and any differences observed in sales performance are more likely due to the feedback intervention rather than external factors.
- **Biases the Company Is Trying to Prevent:**
  - **Performance Bias:** Assigning the most experienced or highest-performing salespeople to the group receiving feedback would lead to skewed results. In this case, the improvement in sales could be attributed to the fact that the more skilled representatives were already better, regardless of the feedback intervention.
  - **Experience Bias:** Placing only new hires in the feedback group and experienced hires in the no-feedback group would not allow the company to properly compare the effects of feedback. The differences in sales results would likely be driven by experience rather than the feedback intervention.
  - **Geographic Bias:** Placing all sales reps from high-sales regions in the feedback group and others in less active regions in the no-feedback group would confound the results, as regional market conditions, rather than the feedback intervention, could drive differences.

Question 3

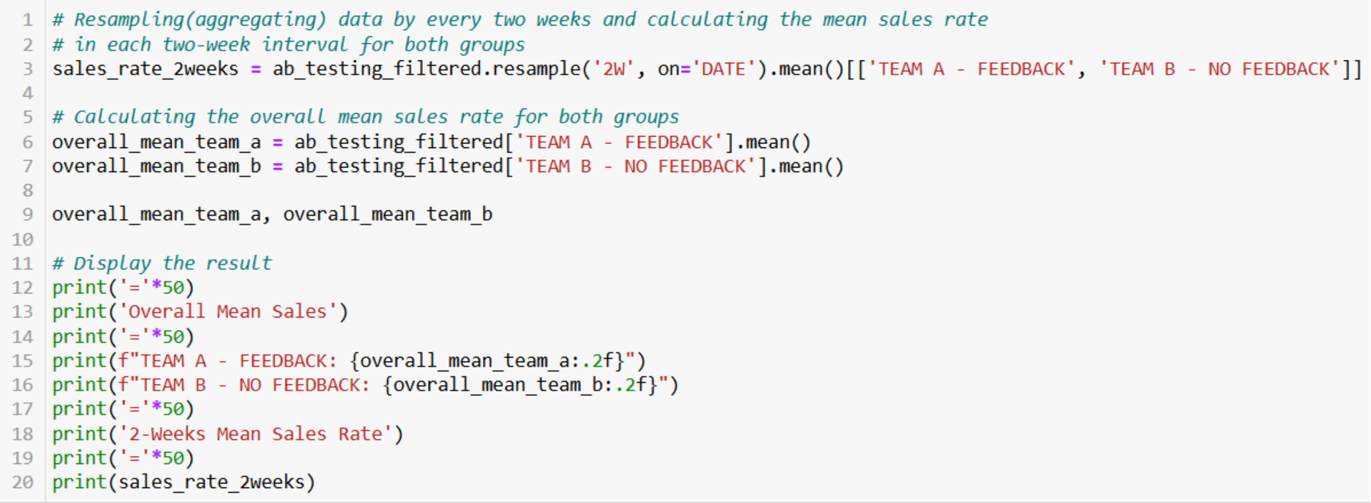
3.06 Time series & A/B Testing: task 6

The Measure the differences in sales performance during the A/B test:

- By calculating means for both groups
- By plotting the sales rates over time
- By suggesting a suitable statistical test

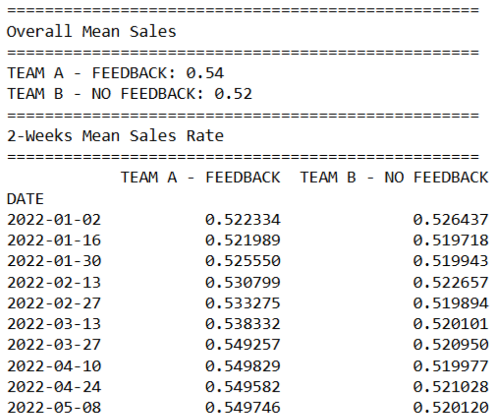
Guidance:

- Calculate the overall mean sales rate or the mean sales rate every two weeks (whichever you prefer)
- Plot the sales rate over time and visually analyse differences
- Comment which statistical test would be appropriate to measure the differences in mean sales rate between the two groups (you do not have to perform the test)



```python
# Resampling(aggregating) data by every two weeks and calculating the mean sales rate
# in each two-week interval for both groups
sales_rate_2weeks = ab_testing_filtered.resample('2W', on='DATE').mean()[['TEAM A - FEEDBACK', 'TEAM B - NO FEEDBACK']]

# Calculating the overall mean sales rate for both groups
overall_mean_team_a = ab_testing_filtered['TEAM A - FEEDBACK'].mean()
overall_mean_team_b = ab_testing_filtered['TEAM B - NO FEEDBACK'].mean()

overall_mean_team_a, overall_mean_team_b

# Display the result
print('='*50)
print('Overall Mean Sales')
print('='*50)
print(f"TEAM A - FEEDBACK: {overall_mean_team_a:.2f}")
print(f"TEAM B - NO FEEDBACK: {overall_mean_team_b:.2f}")
print('='*50)
print('2-Weeks Mean Sales Rate')
print('='*50)
print(sales_rate_2weeks)
```

Answer



```
==================================================
Overall Mean Sales
==================================================
TEAM A - FEEDBACK: 0.54
TEAM B - NO FEEDBACK: 0.52
==================================================
2-Weeks Mean Sales Rate
==================================================
            TEAM A - FEEDBACK  TEAM B - NO FEEDBACK
DATE
2022-01-02           0.522334              0.526437
2022-01-16           0.521989              0.519718
2022-01-30           0.525550              0.519943
2022-02-13           0.530799              0.522657
2022-02-27           0.533275              0.519894
2022-03-13           0.538332              0.520101
2022-03-27           0.549257              0.520950
2022-04-10           0.549829              0.519977
2022-04-24           0.549582              0.521028
2022-05-08           0.549746              0.520120
```

Question 3

3.06 Time series & A/B Testing: task 6

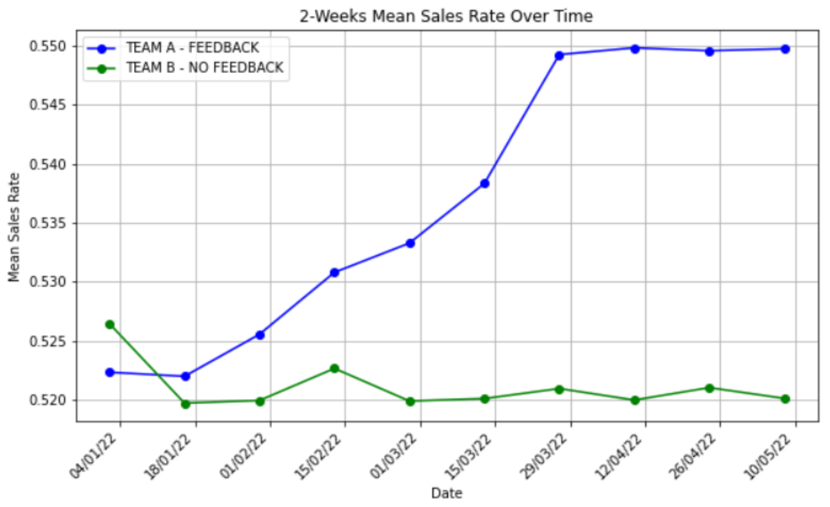
The Measure the differences in sales performance during the A/B test:

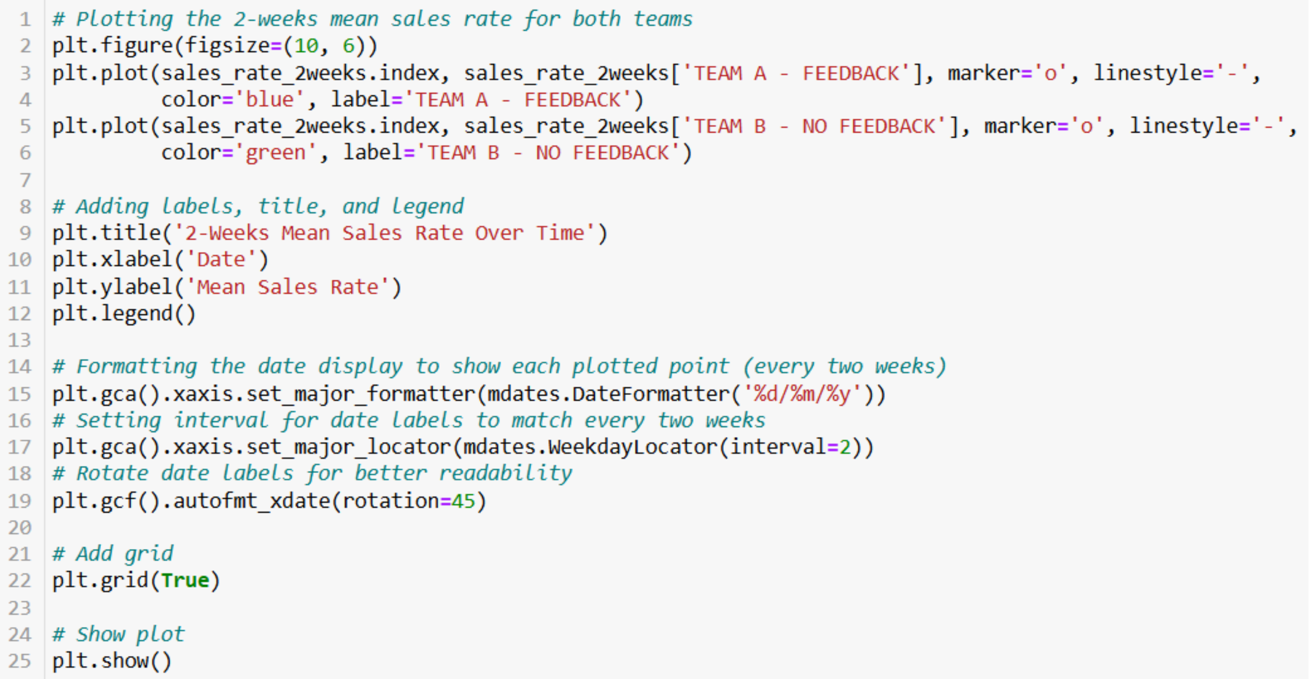
- **By calculating means for both groups**
- **By plotting the sales rates over time**
- **By suggesting a suitable statistical test**

Guidance:

- Calculate the overall mean sales rate or the mean sales rate every two weeks (whichever you prefer)
- Plot the sales rate over time and visually analyse differences
- Comment which statistical test would be appropriate to measure the differences in mean sales rate between the two groups (you do not have to perform the test)

Answer





```python
# Plotting the 2-weeks mean sales rate for both teams
plt.figure(figsize=(10, 6))
plt.plot(sales_rate_2weeks.index, sales_rate_2weeks['TEAM A - FEEDBACK'], marker='o', linestyle='-',
         color='blue', label='TEAM A - FEEDBACK')
plt.plot(sales_rate_2weeks.index, sales_rate_2weeks['TEAM B - NO FEEDBACK'], marker='o', linestyle='-',
         color='green', label='TEAM B - NO FEEDBACK')

# Adding labels, title, and legend
plt.title('2-Weeks Mean Sales Rate Over Time')
plt.xlabel('Date')
plt.ylabel('Mean Sales Rate')
plt.legend()

# Formatting the date display to show each plotted point (every two weeks)
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%d/%m/%y'))
# Setting interval for date labels to match every two weeks
plt.gca().xaxis.set_major_locator(mdates.WeekdayLocator(interval=2))
# Rotate date labels for better readability
plt.gcf().autofmt_xdate(rotation=45)

# Add grid
plt.grid(True)

# Show plot
plt.show()
```

Question 3

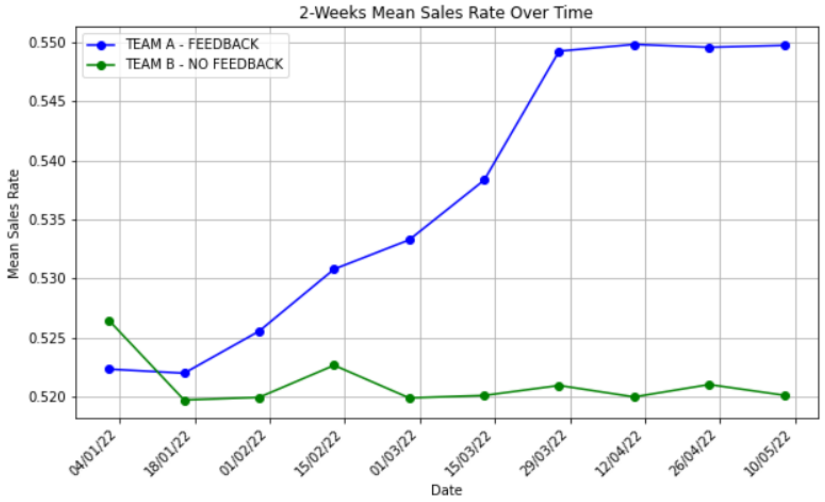
3.06 Time series & A/B Testing: task 6

The plot shows a clear difference in performance between the two groups over time:

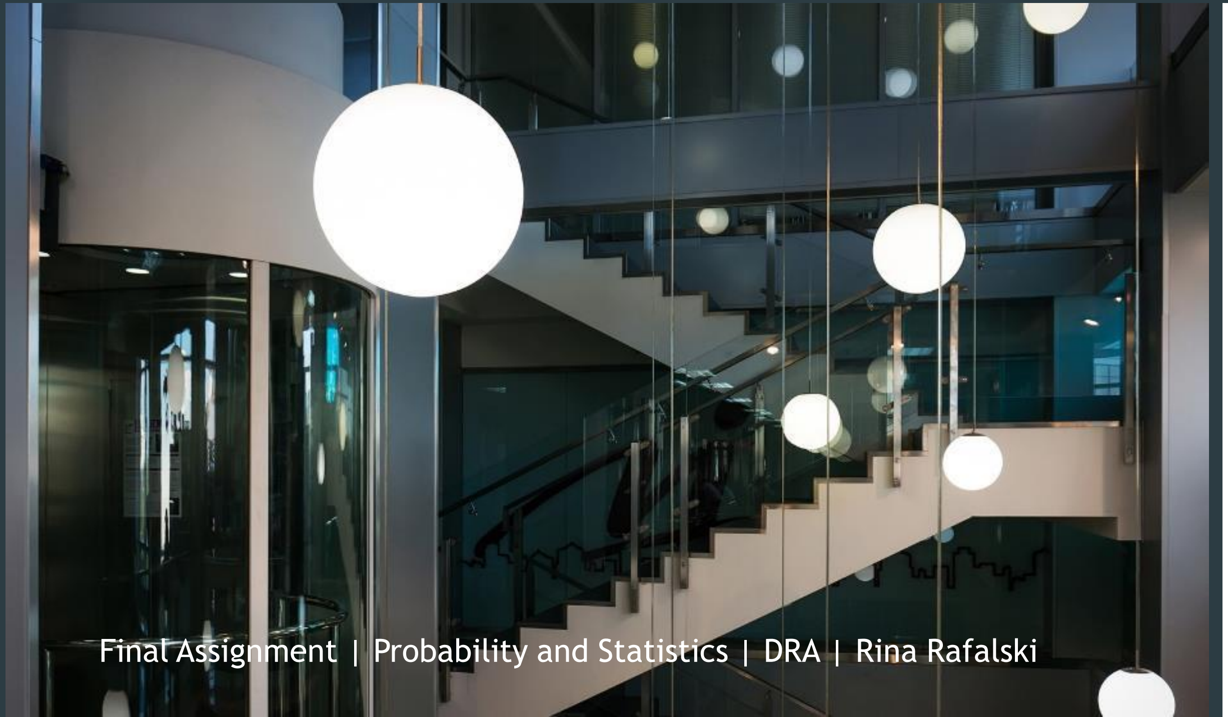
- **Team A - Feedback** shows a consistent upward trend in mean sales rate, indicating that the feedback process has had a positive impact on their performance.
- **Team B - No Feedback**, on the other hand, has a relatively flat trend, indicating no significant improvement in performance over the same period.

Appropriate Statistical Test:

- To measure the differences in the mean sales rates between the two groups, could be performed a **two-sample t-test (independent t-test)** to compare the means of two independent groups (Team A and Team B).
- **Assumptions:** The sales rates in both groups should approximately **follow a normal distribution** (this assumption could be tested using normality tests like the Shapiro-Wilk test). **Variances in both groups should be equal** (this assumption could be tested with Levene’s test or Bartlett’s test).
- **If these assumptions are violated**, a non-parametric alternative such as the **Mann-Whitney U test (Wilcoxon rank-sum test)** would be more appropriate, as it doesn't assume normality or equal variances.



THANK YOU



CONTACT

in Rina Irene Rafalski

@ rinaraf@gmail.com

GitHub Rina-Irene-arch