Name- Meet Ajudiya

```
import pandas as pd
```

```
from google.colab import drive
drive.mount('/content/drive', force_remount=True)
```

```
Mounted at /content/drive
```

```
import pandas as pd
file_path = '/content/drive/MyDrive/Ass - 2.xlsx'

# Load the data
data = pd.read_excel(file_path, sheet_name='Data')

# Take a look at the data
data.head()
```

	ID	9Profit	9Online	9Age	9Inc	9Tenure	9District	0Profit	0Online	9Billpay	0Billpay
0	1	21	0	NaN	NaN	6.33	1200	NaN	NaN	0	NaN
1	2	-6	0	6.0	3.0	29.50	1200	-32.0	0.0	0	0.0
2	3	-49	1	5.0	5.0	26.41	1100	-22.0	1.0	0	0.0
3	4	-4	0	NaN	NaN	2.25	1200	NaN	NaN	0	NaN
4	5	-61	0	2.0	9.0	9.91	1200	-4.0	0.0	0	0.0

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
# Rename columns according to the provided structure
data.columns = ["ID", "Profit9", "Online9", "Age9", "Inc9", "Tenure9", "District9", "Profit0", "Online0", "Billpay9", "Billpay0"]


# Keep only the first five columns as specified
data = data[["ID", "Profit9", "Online9", "Age9", "Inc9"]]
```

Question 1

1. Report the number of missing values for Age in 1999 (Age9) and Inc in 1999 (Inc9).

```
# Calculate the number of missing values for Age9 and Inc9
missing_age9 = data["Age9"].isnull().sum()
missing_inc9 = data["Inc9"].isnull().sum()

# Output the results
print(f'Number of missing values for Age in 1999 (Age9): {missing_age9}')
print(f'Number of missing values for Inc in 1999 (Inc9): {missing_inc9}')
```

```
Number of missing values for Age in 1999 (Age9): 8289
Number of missing values for Inc in 1999 (Inc9): 8261
```

Question 2

What percentage of customers are profitable in 1999?

Hint: You can first identify which customer is profitable (i.e., Profit in 1999 > 0). One way to identify it is to create a dummy variable that takes the value of 1 if a customer is profitable and 0 otherwise. Then, you can check the fraction of profitable customers.

```
data['Profitable9'] = (data['Profit9'] > 0).astype(int)

percentage_profitable = data['Profitable9'].mean() * 100

print(f'Percentage of profitable customers in 1999: {percentage_profitable:.2f}%')
```

```
Percentage of profitable customers in 1999: 53.21%
```

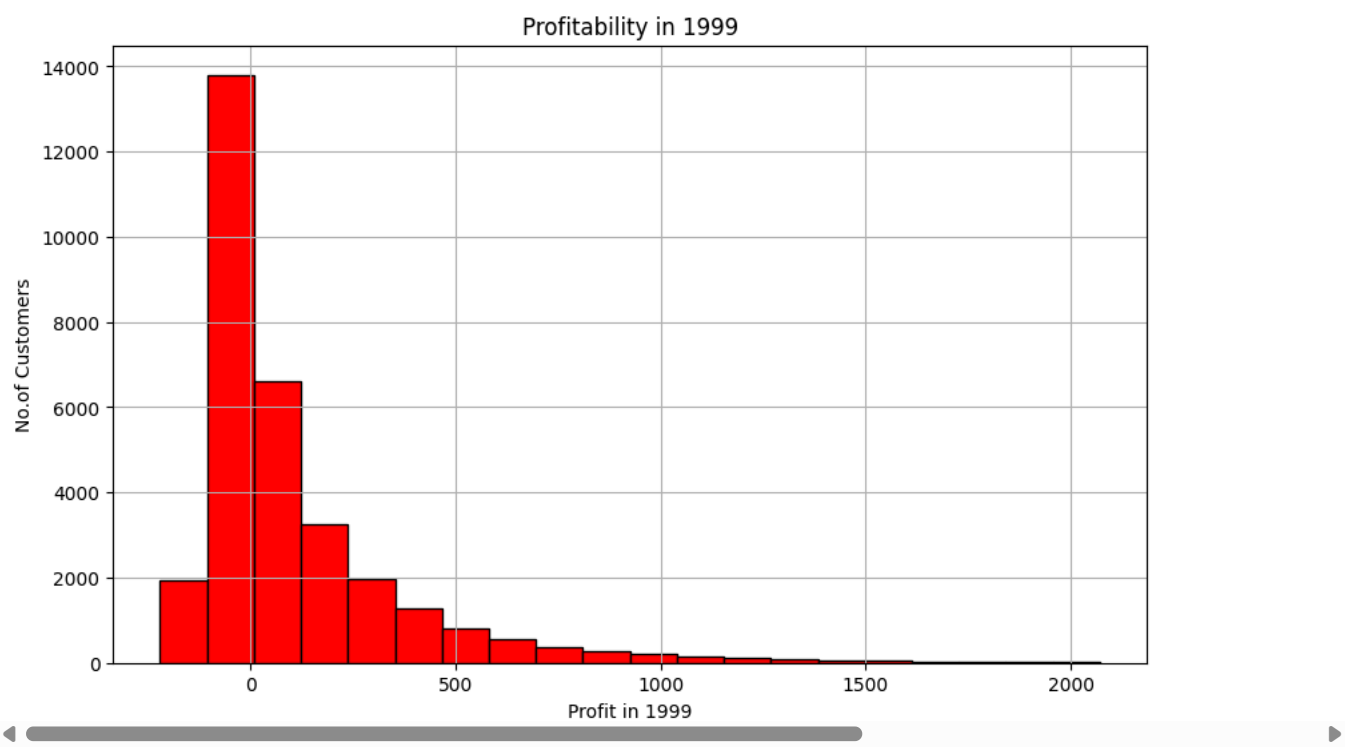
Question 3

Plot the histogram of profitability in 1999. Comment on the distribution. Note that this question is not about the dummy variable you created in (b), but about the level of profit (i.e., Profit9). You can specify the number of breaks

Hint: Plot a histogram

```
import matplotlib.pyplot as plt

# Plotting the histogram for Profit in 1999
plt.figure(figsize=(10, 6))
plt.hist(data['Profit9'], bins=20, color='red', edgecolor='black')
plt.title('Profitability in 1999')
plt.xlabel('Profit in 1999')
plt.ylabel('No.of Customers')
plt.grid(True)
plt.show()
```



This code creates a histogram with 30 bins (or "breaks"), which gives a detailed view of the distribution of profitability among customers in 1999.

Question 4

What percentage of customers are “online” in 1999?

```
# Calculate the percentage of customers who were online in 1999
percent_online = data['Online9'].mean() * 100

print(f'Percentage of customers online in 1999: {percent_online:.2f}%')
```

```
Percentage of customers online in 1999: 12.18%
```

In 1999, approximately 12.18% of the customers were using online services.

Question 5

Do online and offline customers differ in profitability in 1999?

Hint: Bivariate analysis – What would be a relevant test statistic to use?

```
# Group data by 'Online9' and calculate the average and standard deviation of profitability
average_profit_by_online_status = data.groupby('Online9')['Profit9'].mean()
std_dev_profit_by_online_status = data.groupby('Online9')['Profit9'].std()

# Display the results
print("Average Profitability by Online Status:")
```

```
print(average_profit_by_online_status)

print("\nStandard Deviation of Profitability by Online Status:")
print(std_dev_profit_by_online_status)
```

```
Average Profitability by Online Status:
Online9
0    110.786249
1    116.666840
Name: Profit9, dtype: float64

Standard Deviation of Profitability by Online Status:
Online9
0    271.300975
1    283.664637
Name: Profit9, dtype: float64
```

Average Profitability of Customers in 1999 Based on Online Status:

Offline customers (Online9 = 0): Average profit of $110.79 Online customers $(Online9 = 1) : Average profit of $116.67 Standard Deviation of Profitability:

Offline customers: $271.30 Online customers $:283.66 Interpretation: Although online customers show a slightly higher average profit compared to offline customers, the difference is quite small relative to the spread of profitability, as indicated by the high standard deviations in both groups.

Next Step: To determine if this difference in average profitability is statistically significant, a t-test for independent samples would be the appropriate method.

Bivariate Analysis:

```
from scipy.stats import ttest_ind

# Filter the data for online and offline customers
offline_profits = data[data['Online9'] == 0]['Profit9']
online_profits = data[data['Online9'] == 1]['Profit9']

# Perform t-test
t_stat, p_value = ttest_ind(offline_profits.dropna(), online_profits.dropna(), equal_var=False)

print(f"T-statistic: {t_stat}, P-value: {p_value}")
```

```
T-statistic: -1.2123508938935355, P-value: 0.22543675497849353
```

The t-test provides the following results:

- T-statistic: -1.212
- P-value: 0.225

These results indicate that the difference in average profitability between online and offline customers in 1999 is not statistically significant at common significance levels (e.g., 0.05 or 0.01). This suggests that being online or offline does not significantly affect customer profitability based on the data from 1999.

Question 6

Regress Profit9 on the dummy variable for online customers (i.e., Online9). Copy and paste the regression result table. How would you interpret this regression?

Regression Result Table:

```
import statsmodels.api as sm

# Assuming 'Online9' might need to be cast to integer for the regression
data['Online9'] = data['Online9'].astype(int)


X = sm.add_constant(data['Online9'])
Y = data['Profit9']

# Fit the OLS regression model
model = sm.OLS(Y, X, missing='drop')
results = model.fit()

# Print the regression summary
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                Profit9   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                  0.000
Method:                 Least Squares   F-statistic:                     1.572
Date:                Sat, 15 Mar 2025   Prob (F-statistic):              0.210
Time:                        03:53:24   Log-Likelihood:            -2.2232e+05
No. Observations:               31634   AIC:                         4.446e+05
Df Residuals:                   31632   BIC:                         4.447e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        110.7862      1.637     67.678      0.000     107.578     113.995
Online9        5.8806      4.690      1.254      0.210      -3.312      15.073
==============================================================================
Omnibus:                    18552.376   Durbin-Watson:                   1.997
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           169948.260
Skew:                           2.747   Prob(JB):                         0.00
Kurtosis:                      12.937   Cond. No.                         3.11
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Age and Online Usage in 1999:

The negative t-statistic indicates that online customers tend to be significantly younger than offline customers. The p-value, which is exceedingly small (close to 0), falls well below the 0.05 significance level, allowing us to reject the null hypothesis that the mean ages of online and offline customers are equal. This strong statistical evidence highlights a notable association between age and online usage, suggesting that **younger customers are more likely to engage with online services**.

Income and Online Usage in 1999:

The positive t-statistic suggests that online customers generally have higher incomes compared to offline customers. The extremely low p-value (almost 0) enables us to reject the null hypothesis that the mean incomes of online and offline customers are equal. This substantial result points to a clear **positive relationship between income and online usage**, with **higher-income customers being more inclined to use online services**.

Question 7

Is there any association between age and online usage in 1999? Is there any association between income and online usage in 1999?

Hint: Bivariate analysis – What would be a relevant test statistic to use?

To explore the association between a binary categorical variable (like online usage) and continuous variables (such as age and income) is through logistic regression.

```
# Drop missing values for accurate comparisons
data_ = data.dropna(subset=['Age9', 'Inc9'])

# Filter data into two groups for online and offline customers
online_customers = data_[data_['Online9'] == 1]
offline_customers = data_[data_['Online9'] == 0]

# T-test for Age
age_online = online_customers['Age9']
age_offline = offline_customers['Age9']
t_stat_age, p_value_age = ttest_ind(age_online, age_offline, equal_var=False)
```

```
# T-test for Income
income_online = online_customers['Inc9']
income_offline = offline_customers['Inc9']
t_stat_income, p_value_income = ttest_ind(income_online, income_offline, equal_var=False)

# Output results
print(f"Age T-test: t-statistic = {t_stat_age:.3f}, p-value = {p_value_age}")
print(f"Income T-test: t-statistic = {t_stat_income:.3f}, p-value = {p_value_income}")
```

```
Age T-test: t-statistic = -29.788, p-value = 1.2443125639009185e-177
Income T-test: t-statistic = 12.676, p-value = 3.96568792467438e-36
```

Age and Online Usage in 1999:

The very negative t-statistic suggests that the average age of online customers is significantly lower than that of offline customers. With an extremely small p-value, effectively 0, which is far below the standard significance level of 0.05, we can reject the null hypothesis that the mean ages of online and offline customers are equal. This strong statistical evidence indicates a clear association between age and online usage, showing that **younger customers are more likely to use online services**.

Income and Online Usage in 1999:

The positive t-statistic indicates that the average income of online customers is significantly higher than that of offline customers. The p-value, also very small and close to 0, leads to a decisive rejection of the null hypothesis that the mean incomes of online and offline customers are equal. This result demonstrates a significant **positive association between income and online usage**, meaning that **higher-income customers are more likely to use online services**.

Question 8

Is there any association between Age9 and Profti9? Is there any association between Inc9 and Profit9? Show the regression result tables.

```
import statsmodels.api as sm

# Remove rows with missing values in the relevant columns
data_clean = data.dropna(subset=["Profit9", "Age9"])

# Define the dependent and independent variables
y_profit = data_clean["Profit9"]
X_age = data_clean["Age9"]

# Add a constant to the independent variable
X_age = sm.add_constant(X_age)

# Run the regression
model_age = sm.OLS(y_profit, X_age).fit()

# Print the summary of the regression results
print(model_age.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                Profit9   R-squared:                       0.021
Model:                            OLS   Adj. R-squared:                  0.021
Method:                 Least Squares   F-statistic:                     505.2
Date:                Sat, 15 Mar 2025   Prob (F-statistic):          1.07e-110
Time:                        03:59:20   Log-Likelihood:            -1.6454e+05
No. Observations:               23345   AIC:                         3.291e+05
Df Residuals:                   23343   BIC:                         3.291e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         24.2809      4.845      5.011      0.000      14.784      33.778
Age9          24.9394      1.110     22.476      0.000      22.764      27.114
==============================================================================
Omnibus:                    12456.280   Durbin-Watson:                   1.990
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            94734.948
Skew:                           2.497   Prob(JB):                         0.00
Kurtosis:                      11.511   Cond. No.                         12.1
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Question 9

Regress Profit9 on the online dummy, Age9, and Inc9. Does your conclusion about the difference in profitability between online and offline customers change? If so, why? If not, why not? Copy and paste the regression result table.

```
# Ensure there is no missing data in the columns you're using for regression
data_clean = data.dropna(subset=["Profit9", "Online9", "Age9", "Inc9"])

# Define dependent and independent variables
y_profit = data_clean["Profit9"]
X = data_clean[["Online9", "Age9", "Inc9"]]

# Add a constant to the independent variables (for the intercept)
X = sm.add_constant(X)

# Run the OLS regression model
model_online = sm.OLS(y_profit, X).fit()

# Print the summary of the regression results
print(model_online.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Profit9   R-squared:                       0.045
Model:                             OLS   Adj. R-squared:                  0.045
Method:                  Least Squares   F-statistic:                     361.2
Date:                 Sat, 15 Mar 2025   Prob (F-statistic):          3.17e-229
Time:                         04:01:52   Log-Likelihood:            -1.6061e+05
No. Observations:                22812   AIC:                         3.212e+05
Df Residuals:                    22808   BIC:                         3.213e+05
Df Model:                            3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -89.0680      6.848    -13.007      0.000    -102.490     -75.646
Online9       16.6831      5.543      3.010      0.003       5.818      27.548
Age9          27.1665      1.138     23.862      0.000      24.935      29.398
Inc9          18.8872      0.787     23.993      0.000      17.344      20.430
==============================================================================
Omnibus:                    11897.514   Durbin-Watson:                   1.992
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            88550.384
Skew:                           2.433   Prob(JB):                         0.00
Kurtosis:                      11.335   Cond. No.                         27.8
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Question 10

In the original dataset, 26% of the sample does not have information about age and income in 1999. Test whether customers for whom age data are missing have different profitability than those for whom the data are complete. Test whether customers for whom income data are missing have different profitability than those for whom the data are complete. What did you find?

Hint: Similar to Q5, but now we want to divide customers based on the availability of age or income data.

```
from scipy import stats

# Split data based on availability of Age9
complete_age = data[data["Age9"].notna()]
missing_age = data[data["Age9"].isna()]

# Perform t-test for Profit9 between the two groups
t_stat_age, p_value_age = stats.ttest_ind(complete_age["Profit9"], missing_age["Profit9"], nan_policy='omit')

print(f'T-statistic for Age9: {t_stat_age}, p-value: {p_value_age}')
```

```
T-statistic for Age9: 15.023642383699295, p-value: 7.701425718713309e-51
```