Assignment - Statistics [Major]

Name – Amit Kumar

E-mail- amitkumar209727@gmail.com

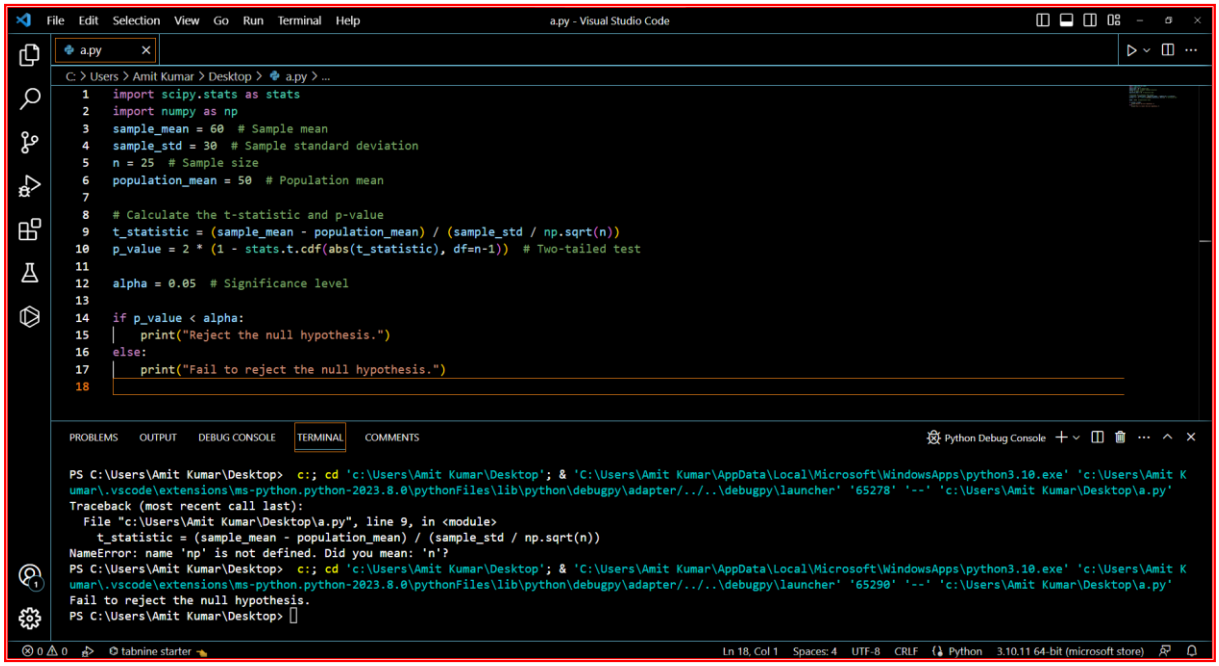
M0- 7881178181

Batch No- 7

1. According to a study, the daily average time spent by a user on a social media website is 50 minutes. To test the claim of this study, Ramesh, a researcher, takes a sample of 25 website users and finds out that the mean time spent by the sample users is 60 minutes and the sample standard deviation is 30 minutes. Based on this information, the null and the alternative hypotheses will be: Ho = The average time spent by the users is 50 minutes H1 = The average time spent by the users is not 50 minutes Use a 5% significance level to test this hypothesis.

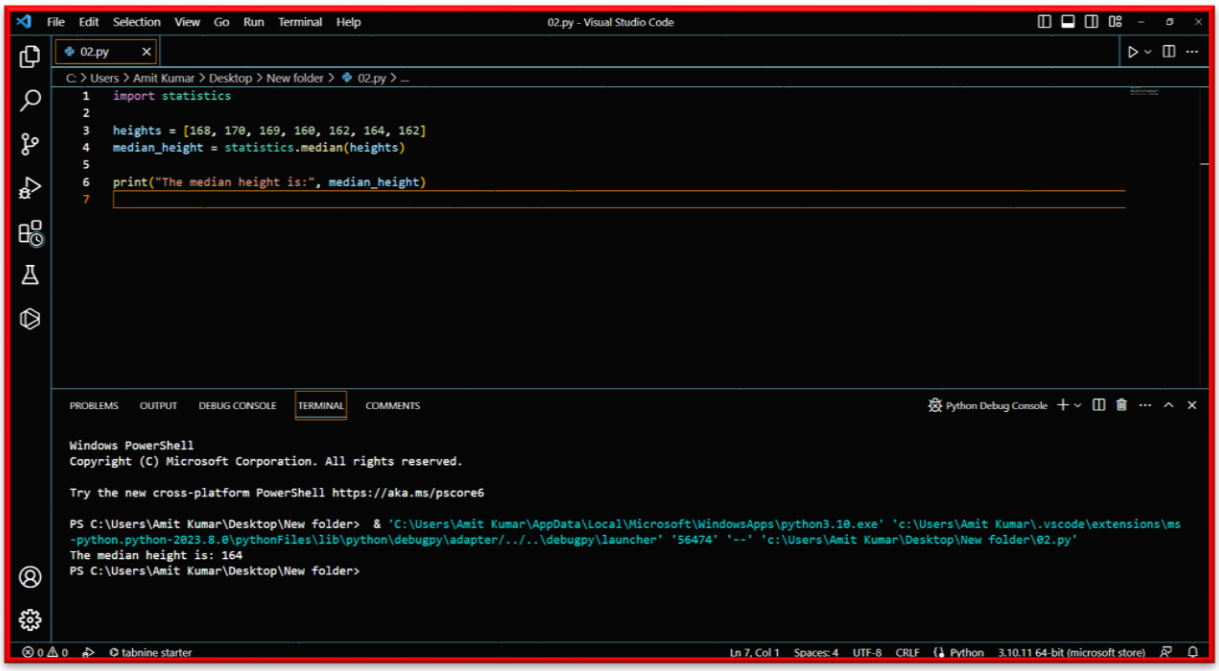
```
import scipy.stats as stats
import numpy as np
sample_mean = 60  # Sample mean
sample_std = 30  # Sample standard deviation
n = 25  # Sample size
population_mean = 50  # Population mean

# Calculate the t-statistic and p-value
```

```
t_statistic = (sample_mean - population_mean) / (sample_std / np.sqrt(n))
p_value = 2 * (1 - stats.t.cdf(abs(t_statistic), df=n-1))  # Two-tailed test

alpha = 0.05  # Significance level

if p_value < alpha:
    print("Reject the null hypothesis.")
else:
    print("Fail to reject the null hypothesis.")
```



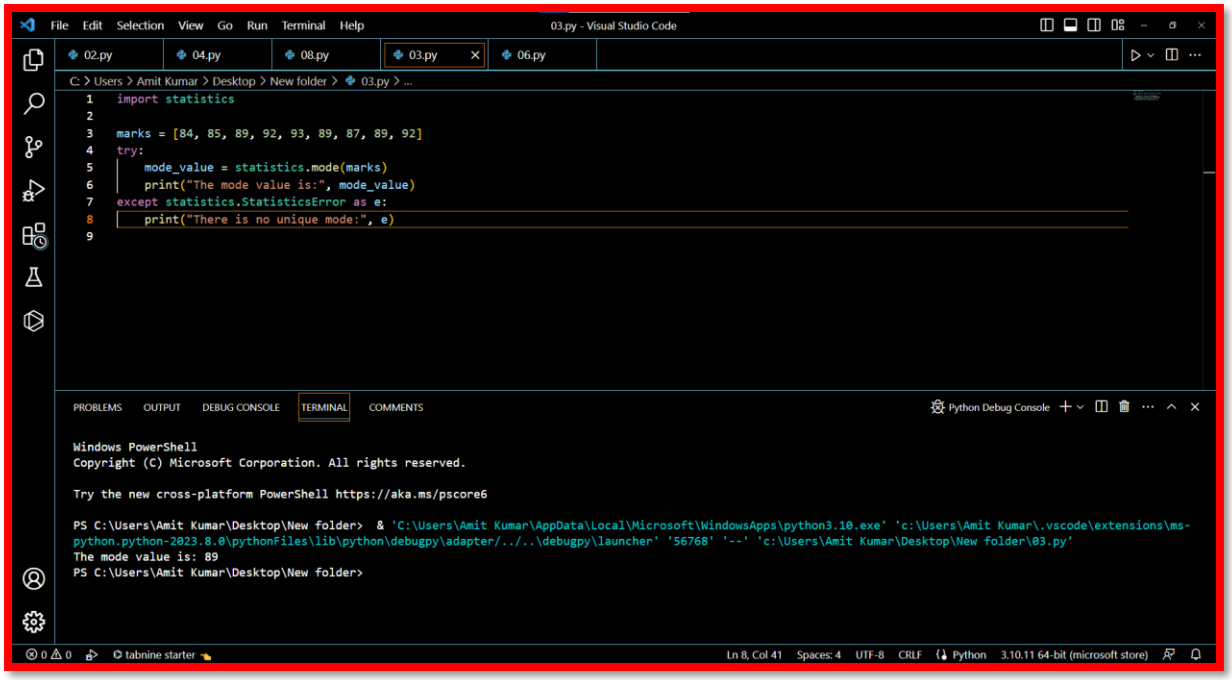
2. Height of 7 students (in cm) is given below. What is the median? 168 170 169 160 162 164 162.

```
import statistics

heights = [168, 170, 169, 160, 162, 164, 162]
median_height = statistics.median(heights)

print("The median height is:", median_height)
```



3. Below are the observations of the marks of a student. Find the value of mode. 84 85 89 92 93 89 87 89 92.

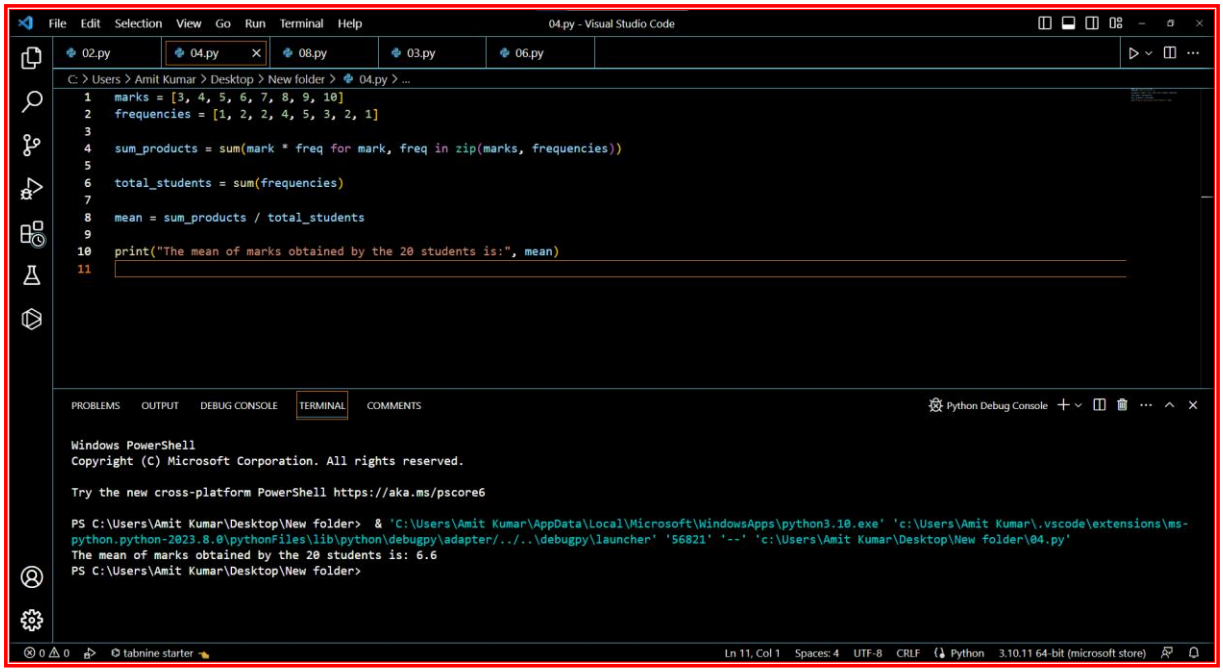
```python
import statistics

marks = [84, 85, 89, 92, 93, 89, 87, 89, 92]
try:
    mode_value = statistics.mode(marks)
    print("The mode value is:", mode_value)
except statistics.StatisticsError as e:
    print("There is no unique mode:", e)
```



4. From the table given below, what is the mean of marks obtained by 20 students?

Marks Xi	No. of students fi
3	1
4	2
5	2
6	4
7	5
8	3
9	2
10	1
Total	20

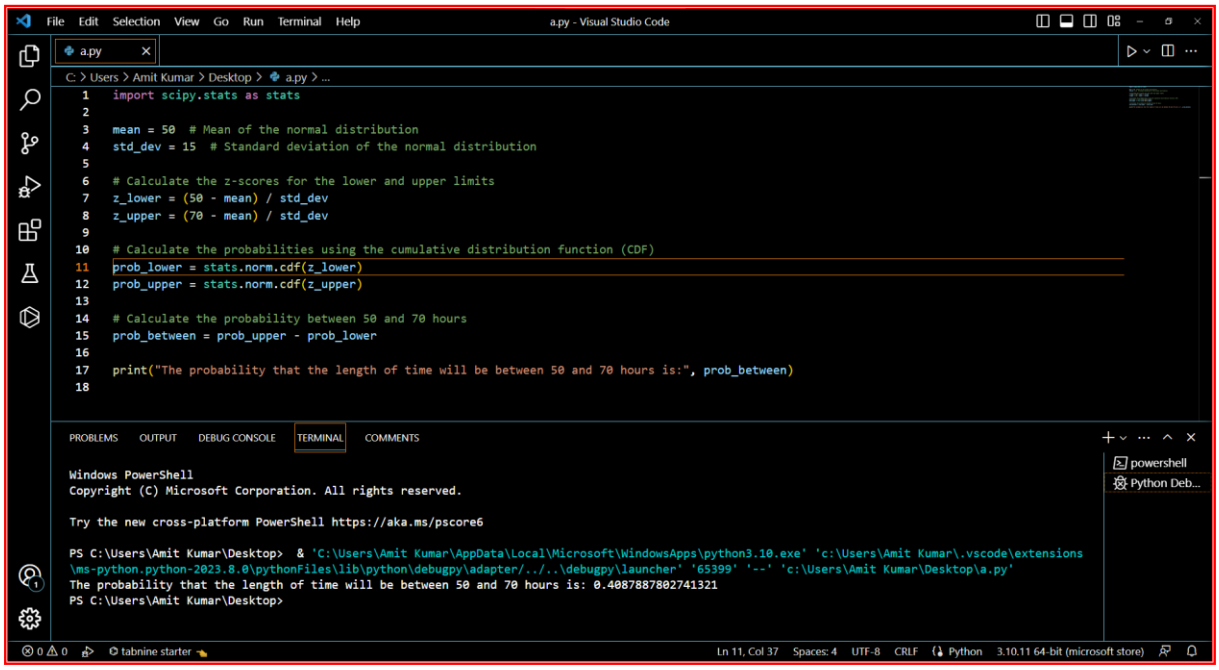
```
marks = [3, 4, 5, 6, 7, 8, 9, 10]
frequencies = [1, 2, 2, 4, 5, 3, 2, 1]

sum_products = sum(mark * freq for mark, freq in zip(marks, frequencies))

total_students = sum(frequencies)

mean = sum_products / total_students

print("The mean of marks obtained by the 20 students is:", mean)
```



5. For a certain type of computer, the length of time between charges of the battery is normally distributed with a mean of 50 hours and a standard deviation of 15 hours. John owns one of these computers and wants to know the probability that the length of time will be between 50 and 70 hours.

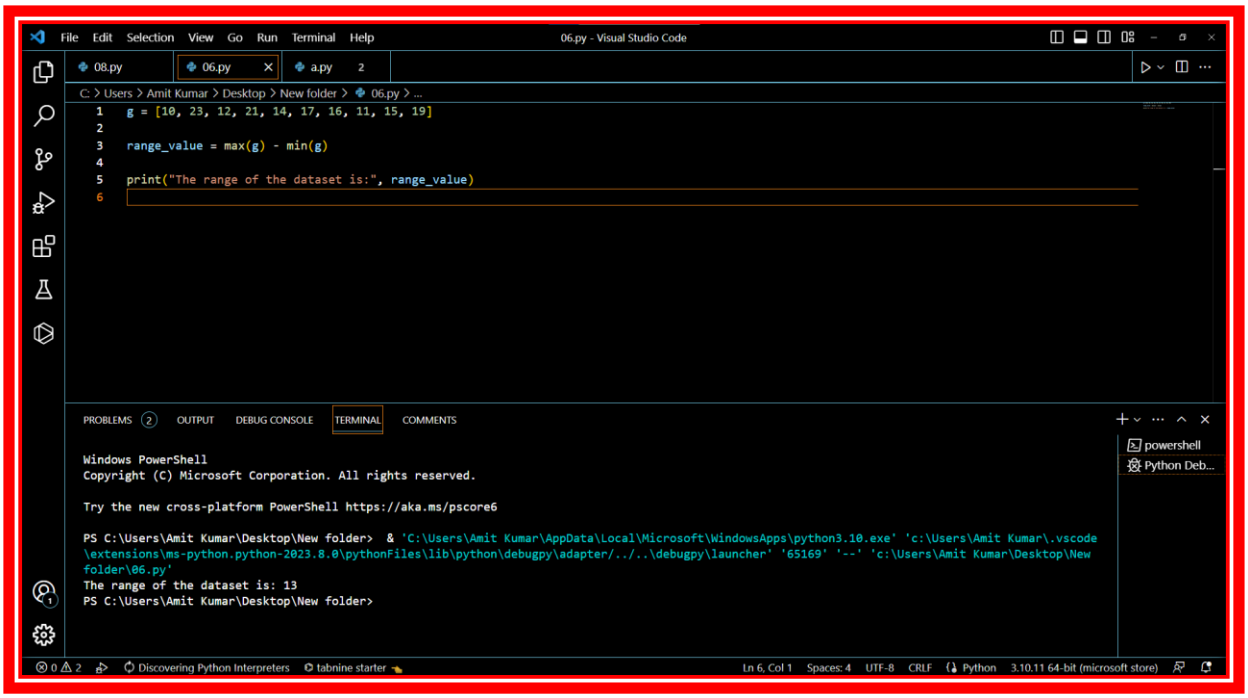
```
import scipy.stats as stats

mean = 50  # Mean of the normal distribution
std_dev = 15  # Standard deviation of the normal distribution

# Calculate the z-scores for the lower and upper limits
z_lower = (50 - mean) / std_dev
z_upper = (70 - mean) / std_dev

# Calculate the probabilities using the cumulative distribution function (CDF)
prob_lower = stats.norm.cdf(z_lower)
prob_upper = stats.norm.cdf(z_upper)

# Calculate the probability between 50 and 70 hours
```

```
prob_between = prob_upper - prob_lower

print("The probability that the length of time will be between 50 and 70 hours is:", prob_between)
```



6. Find the range of the following. g = [10, 23, 12, 21, 14, 17, 16, 11, 15, 19]

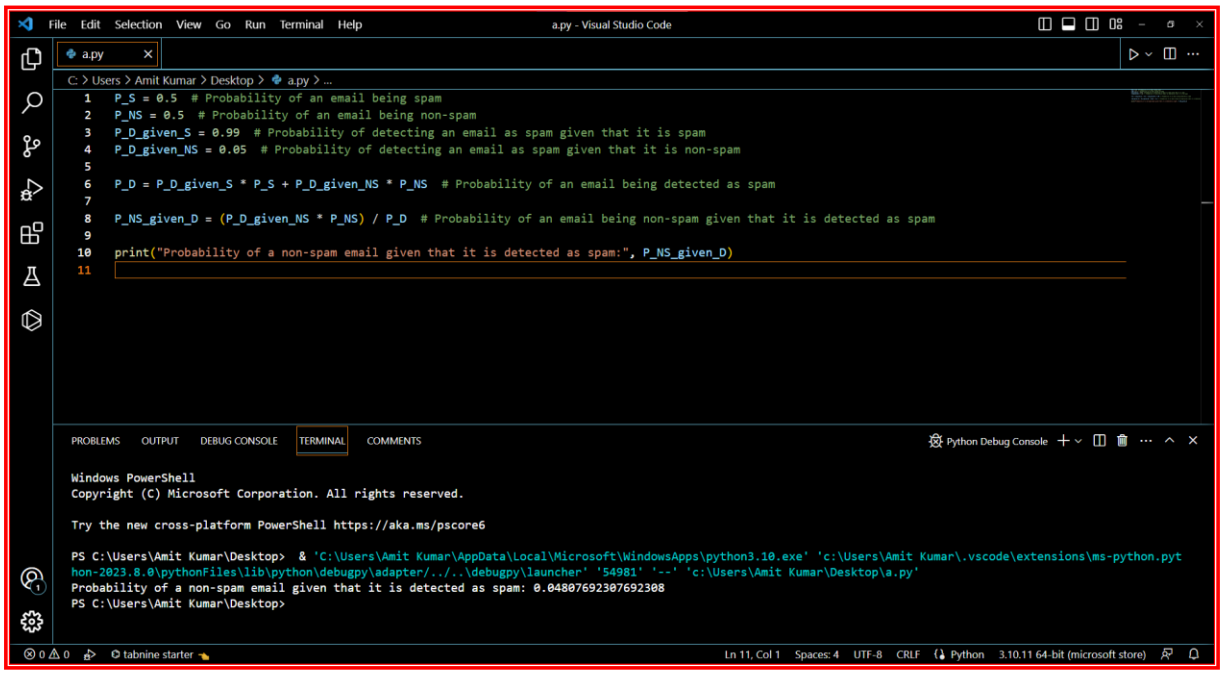
```
g = [10, 23, 12, 21, 14, 17, 16, 11, 15, 19]

range_value = max(g) - min(g)

print("The range of the dataset is:", range_value)
```



7. It is estimated that 50% of emails are spam emails. Some software has been applied to filter these spam emails before they reach your inbox. A certain brand of software claims that it can detect 99% of spam emails, and the probability for a false positive (a non-spam 2 email detected as spam) is 5%. Now if an email is detected as spam, then what is the probability that it is in fact a non-spam email?

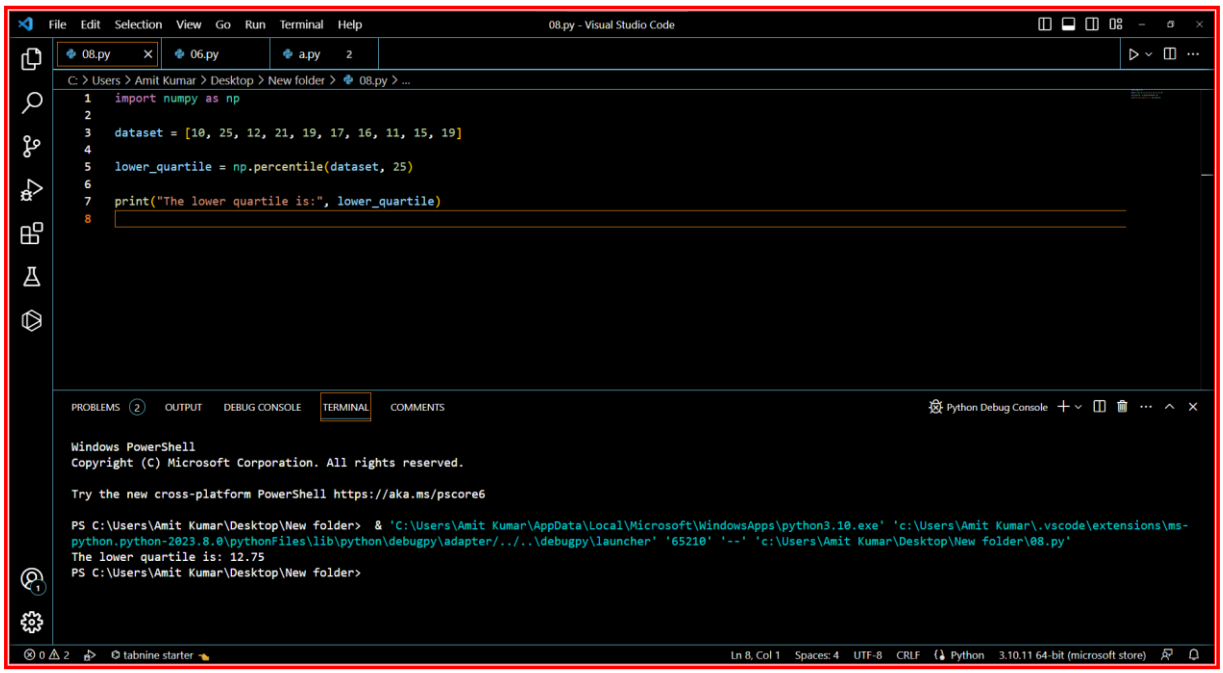
```
P_S = 0.5  # Probability of an email being spam

P_NS = 0.5  # Probability of an email being non-spam

P_D_given_S = 0.99  # Probability of detecting an email as spam given that it is spam

P_D_given_NS = 0.05  # Probability of detecting an email as spam given that it is non-spam
```

```
P_D = P_D_given_S * P_S + P_D_given_NS * P_NS  # Probability of an email being detected as spam


P_NS_given_D = (P_D_given_NS * P_NS) / P_D  # Probability of an email being non-spam given that it is detected as spam


print("Probability of a non-spam email given that it is detected as spam:", P_NS_given_D)
```



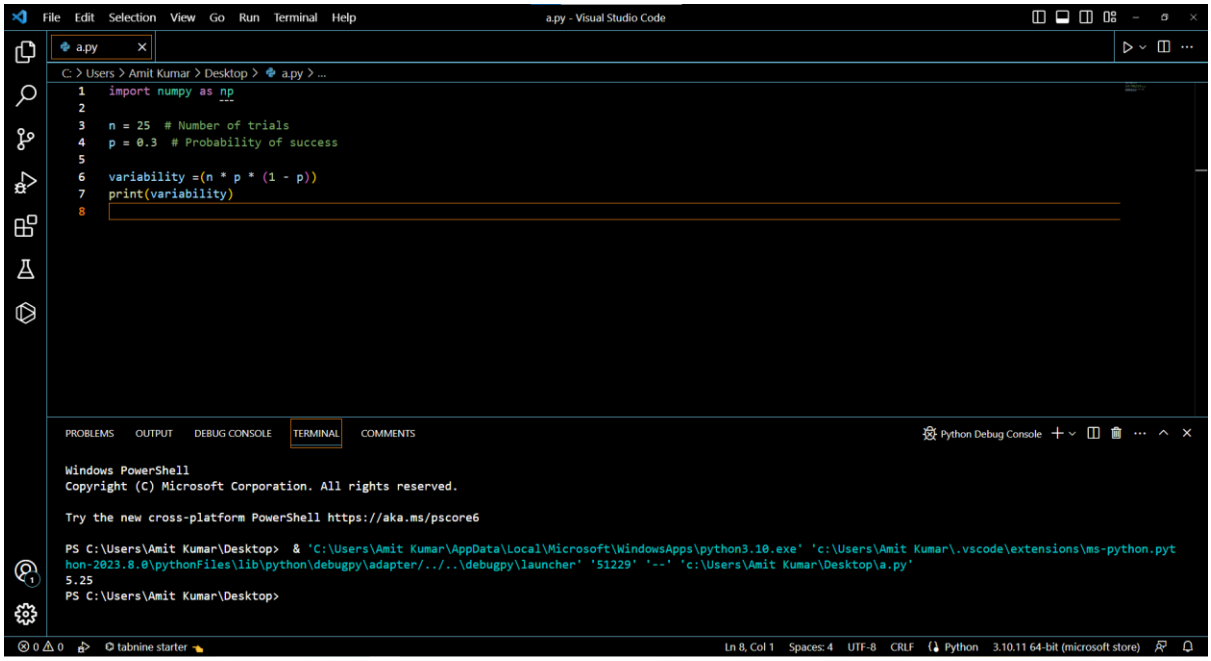
8. Given the following distribution of returns, determine the lower quartile: {10 25 12 21 19 17 16 11 15 19}

```
import numpy as np

dataset = [10, 25, 12, 21, 19, 17, 16, 11, 15, 19]

lower_quartile = np.percentile(dataset, 25)

print("The lower quartile is:", lower_quartile)
```



9. For a Binomial distribution, the number of trials(n) is 25, and the probability of success is 0.3. What's the variability of the distribution?

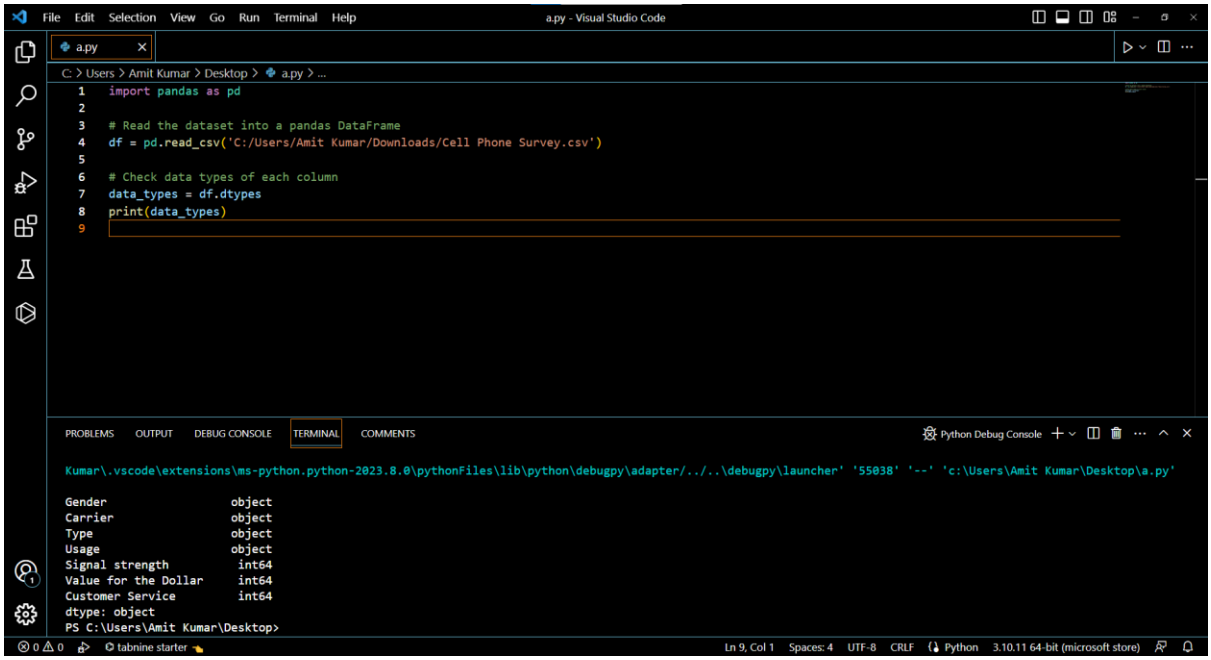
```
import numpy as np

n = 25  # Number of trials
p = 0.3  # Probability of success

variability =(n * p * (1 - p))
print(variability)
```



10. Download the Cell Phone Survey Dataset and perform the below mentioned operations on the dataset:-

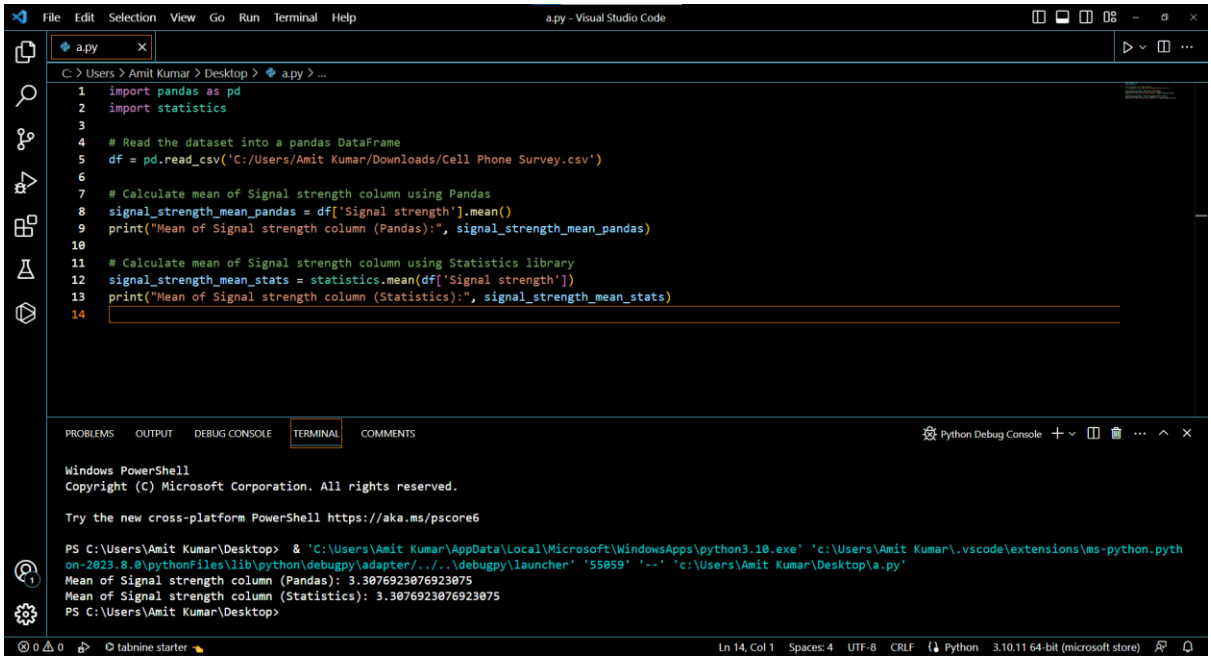
- **Checking datatypes of each column in the dataset.**

```
import pandas as pd

# Read the dataset into a pandas DataFrame
df = pd.read_csv('C:/Users/Amit Kumar/Downloads/Cell Phone Survey.csv')

# Check data types of each column
data_types = df.dtypes
print(data_types)
```



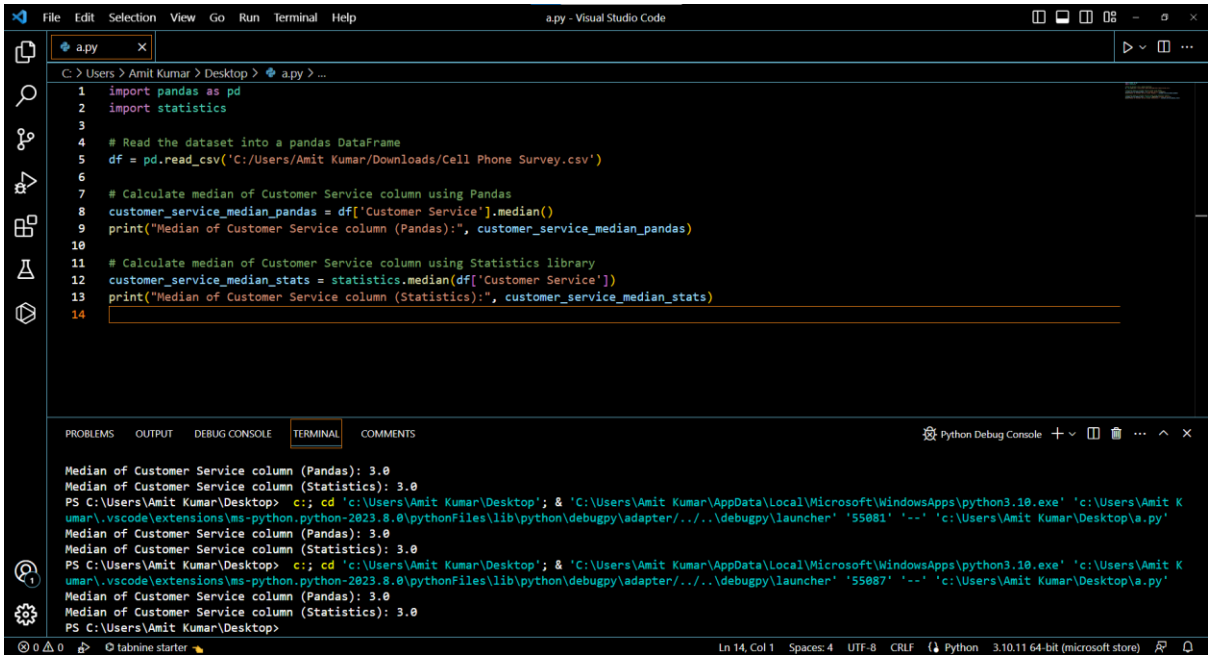
- **Find Mean of Signal strength column using Pandas and Statistics library.**

```
import pandas as pd
import statistics

# Read the dataset into a pandas DataFrame
df = pd.read_csv('C:/Users/Amit Kumar/Downloads/Cell Phone Survey.csv')

# Calculate mean of Signal strength column using Pandas
signal_strength_mean_pandas = df['Signal strength'].mean()
print("Mean of Signal strength column (Pandas):", signal_strength_mean_pandas)

# Calculate mean of Signal strength column using Statistics library
signal_strength_mean_stats = statistics.mean(df['Signal strength'])
print("Mean of Signal strength column (Statistics):",
signal_strength_mean_stats)
```



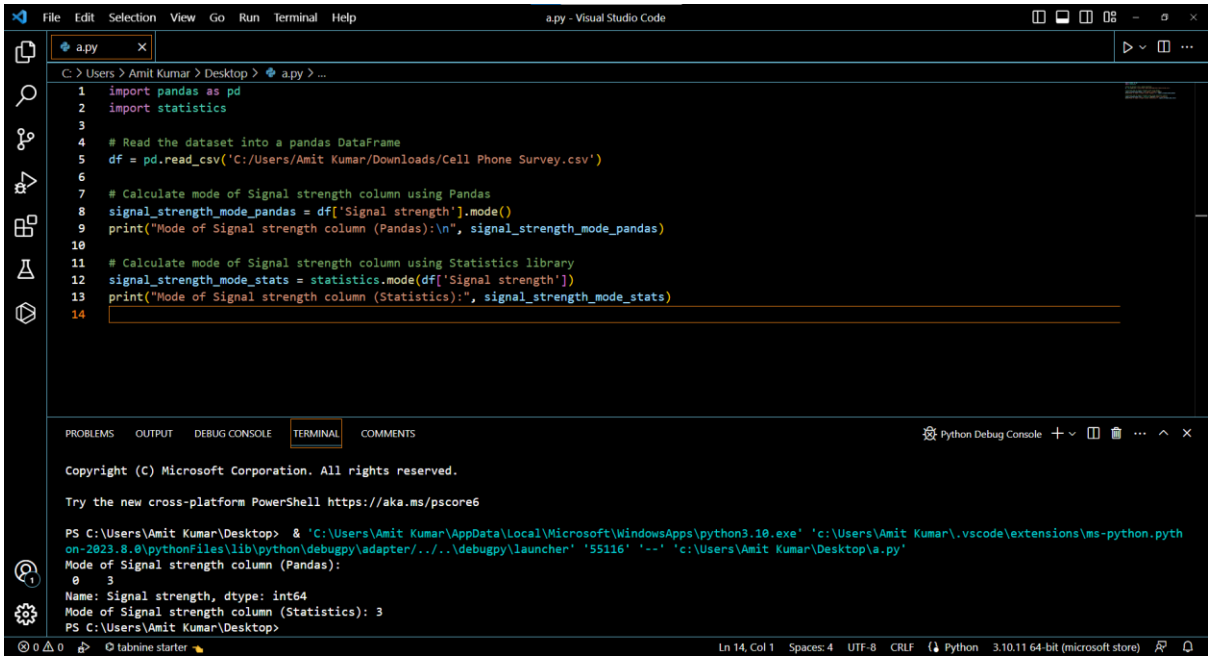
- **Find the Median of Customer Service column using Pandas and Statistics library.**

```
import pandas as pd
import statistics

# Read the dataset into a pandas DataFrame
df = pd.read_csv('C:/Users/Amit Kumar/Downloads/Cell Phone Survey.csv')

# Calculate median of Customer Service column using Pandas
customer_service_median_pandas = df['Customer Service'].median()
print("Median of Customer Service column (Pandas):",
customer_service_median_pandas)

# Calculate median of Customer Service column using Statistics library
customer_service_median_stats = statistics.median(df['Customer Service'])
print("Median of Customer Service column (Statistics):",
customer_service_median_stats)
```



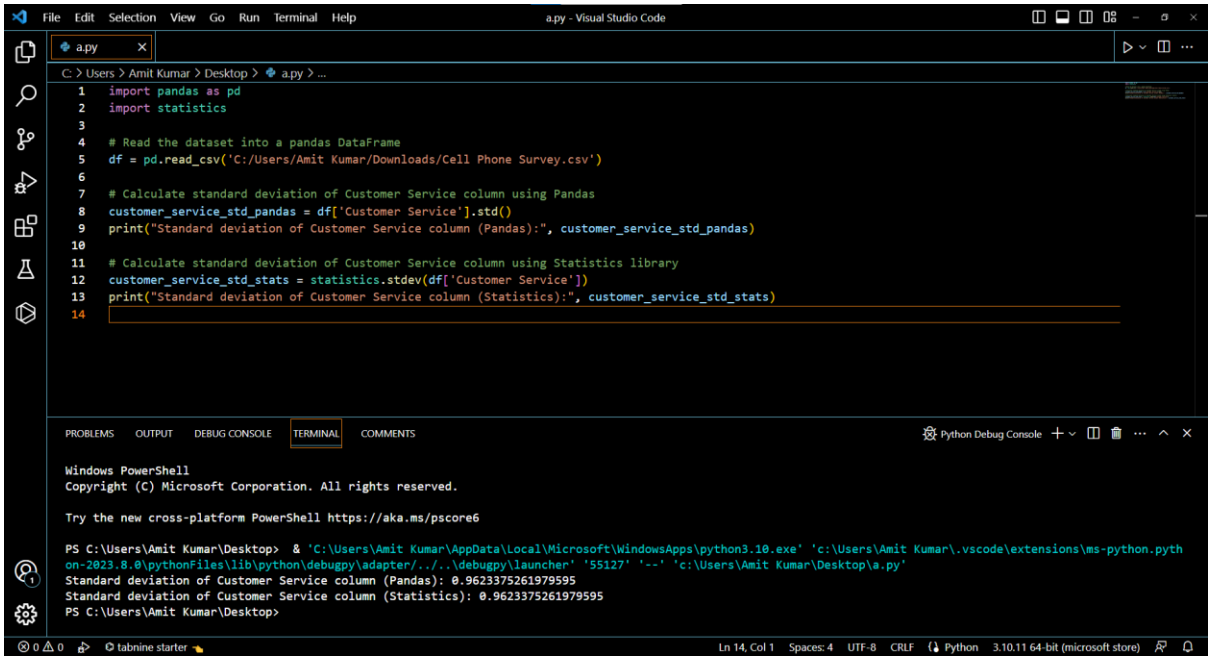
- **Find Mode of Signal strength column using Pandas and Statistics library.**

```
import pandas as pd
import statistics

# Read the dataset into a pandas DataFrame
df = pd.read_csv('C:/Users/Amit Kumar/Downloads/Cell Phone Survey.csv')

# Calculate mode of Signal strength column using Pandas
signal_strength_mode_pandas = df['Signal strength'].mode()
print("Mode of Signal strength column (Pandas):\n",
signal_strength_mode_pandas)

# Calculate mode of Signal strength column using Statistics library
signal_strength_mode_stats = statistics.mode(df['Signal strength'])
print("Mode of Signal strength column (Statistics):",
signal_strength_mode_stats)
```



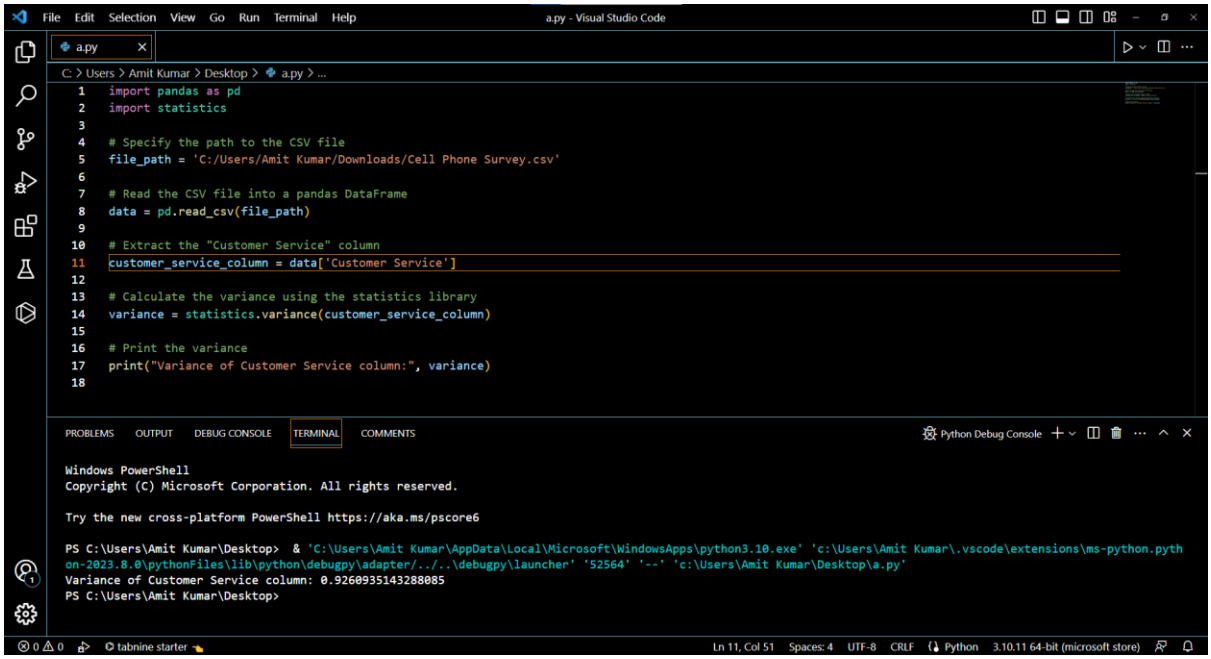
- **Find Standard deviation of Customer Service column using Pandas and Statistics library.**

```
import pandas as pd
import statistics

# Read the dataset into a pandas DataFrame
df = pd.read_csv('C:/Users/Amit Kumar/Downloads/Cell Phone Survey.csv')

# Calculate standard deviation of Customer Service column using Pandas
customer_service_std_pandas = df['Customer Service'].std()
print("Standard deviation of Customer Service column (Pandas):",
customer_service_std_pandas)

# Calculate standard deviation of Customer Service column using Statistics
library
customer_service_std_stats = statistics.stdev(df['Customer Service'])
print("Standard deviation of Customer Service column (Statistics):",
customer_service_std_stats)
```



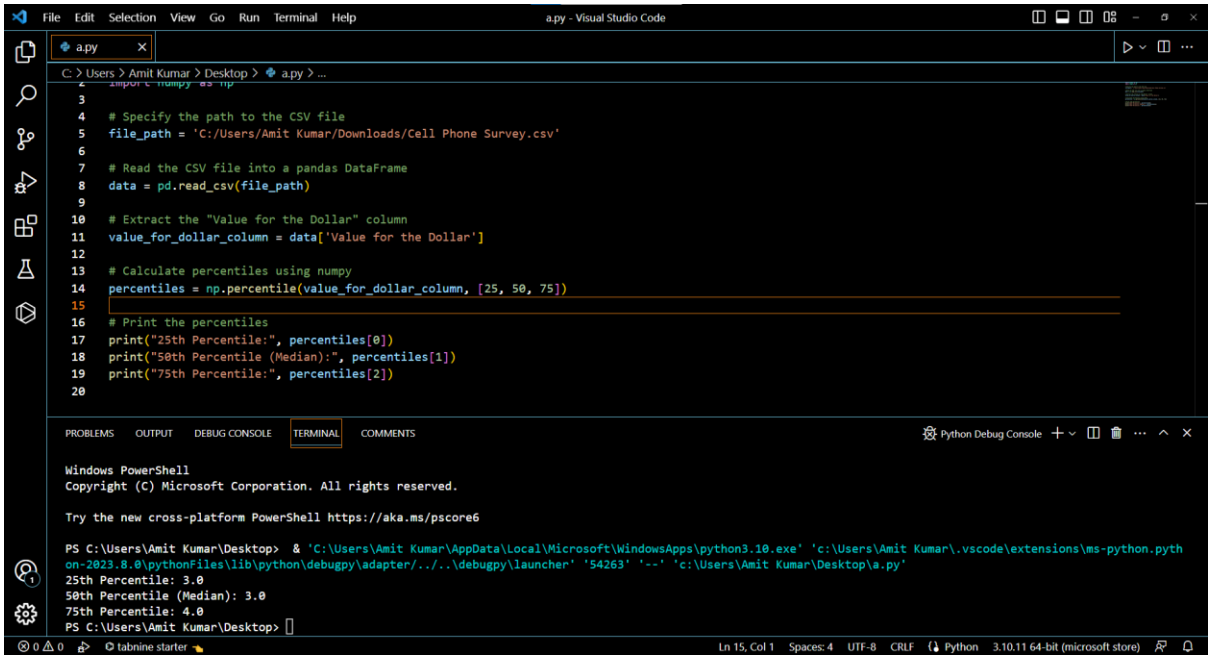
- **Find Variance of Customer Service column using Pandas and Statistics library.**

```
import pandas as pd
import statistics

# Specify the path to the CSV file
file_path = 'C:/Users/Amit Kumar/Downloads/Cell Phone Survey.csv'

# Read the CSV file into a pandas DataFrame
data = pd.read_csv(file_path)

# Extract the "Customer Service" column
customer_service_column = data['Customer Service']

# Calculate the variance using the statistics library
variance = statistics.variance(customer_service_column)

# Print the variance
print("Variance of Customer Service column:", variance)
```



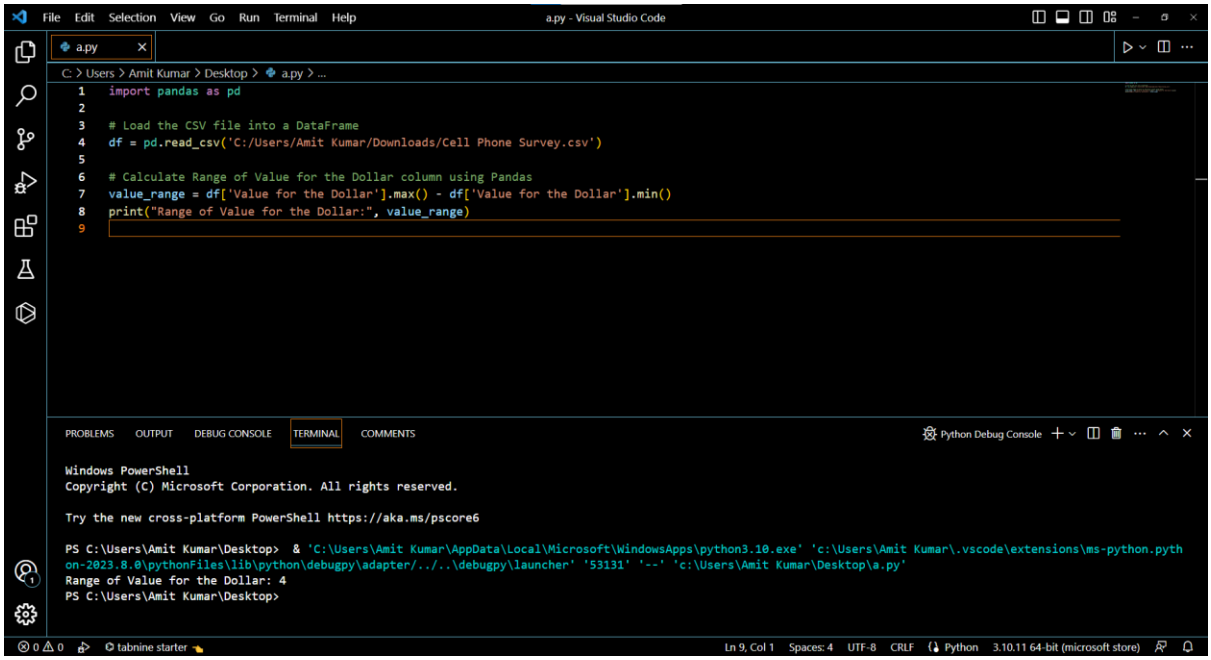
● Calculate Percentiles of Value for the Dollar column using Numpy.

```
import pandas as pd
import numpy as np

# Specify the path to the CSV file
file_path = 'C:/Users/Amit Kumar/Downloads/Cell Phone Survey.csv'

# Read the CSV file into a pandas DataFrame
data = pd.read_csv(file_path)

# Extract the "Value for the Dollar" column
value_for_dollar_column = data['Value for the Dollar']

# Calculate percentiles using numpy
percentiles = np.percentile(value_for_dollar_column, [25, 50, 75])

# Print the percentiles
print("25th Percentile:", percentiles[0])
print("50th Percentile (Median):", percentiles[1])
print("75th Percentile:", percentiles[2])
```



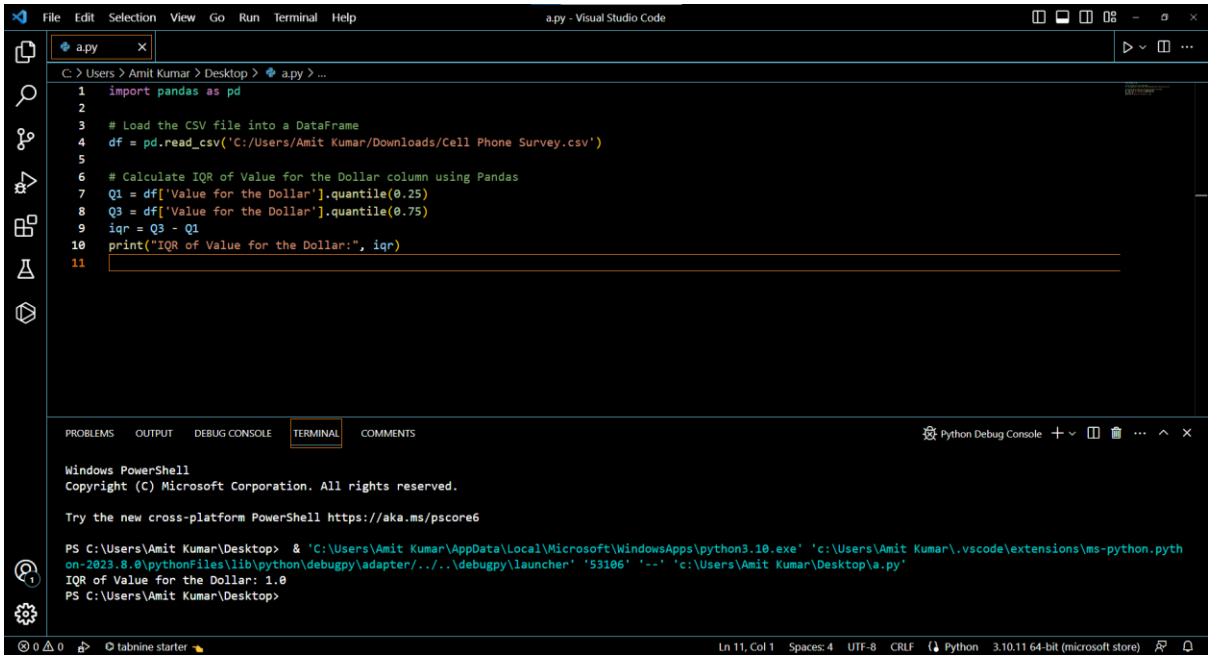
- **Calculate Range of Value for the Dollar column using Pandas.**

```
import pandas as pd

# Load the CSV file into a DataFrame
df = pd.read_csv('C:/Users/Amit Kumar/Downloads/Cell Phone Survey.csv')

# Calculate Range of Value for the Dollar column using Pandas
value_range = df['Value for the Dollar'].max() - df['Value for the
Dollar'].min()
print("Range of Value for the Dollar:", value_range)
```



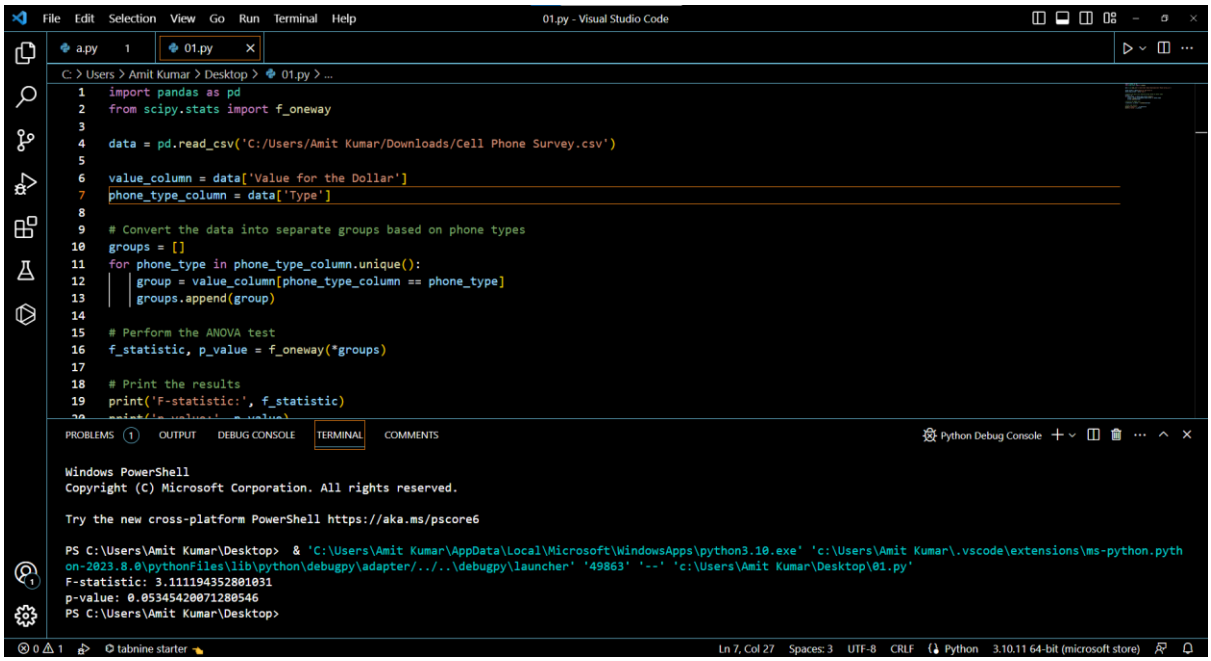
- **Calculate IQR of Value for the Dollar column using Pandas.**

```
import pandas as pd

# Load the CSV file into a DataFrame
df = pd.read_csv('C:/Users/Amit Kumar/Downloads/Cell Phone Survey.csv')

# Calculate IQR of Value for the Dollar column using Pandas
Q1 = df['Value for the Dollar'].quantile(0.25)
Q3 = df['Value for the Dollar'].quantile(0.75)
iqr = Q3 - Q1
print("IQR of Value for the Dollar:", iqr)
```



- **Hypothesis Testing - Using the data in the Cell Phone Survey dataset, apply ANOVA to determine if the mean response for Value for dollar is the same for different types of cell phones.**

```
import pandas as pd
from scipy.stats import f_oneway

data = pd.read_csv('C:/Users/Amit Kumar/Downloads/Cell Phone Survey.csv')

value_column = data['Value for the Dollar']
phone_type_column = data['Type']

# Convert the data into separate groups based on phone types
groups = []
for phone_type in phone_type_column.unique():
    group = value_column[phone_type_column == phone_type]
    groups.append(group)

# Perform the ANOVA test
f_statistic, p_value = f_oneway(*groups)

# Print the results
print('F-statistic:', f_statistic)
print('p-value:', p_value)
```


File Edit Selection View Go Run Terminal Help
01.py - Visual Studio Code
a.py 1
01.py
C: > Users > Amit Kumar > Desktop > 01.py > ...
1 import pandas as pd
2 from scipy.stats import f_oneway
3
4 data = pd.read_csv('C:/Users/Amit Kumar/Downloads/Cell Phone Survey.csv')
5
6 value_column = data['Value for the Dollar']
7 phone_type_column = data['Type']
8
9 # Convert the data into separate groups based on phone types
10 groups = []
11 for phone_type in phone_type_column.unique():
12 group = value_column[phone_type_column == phone_type]
13 groups.append(group)
14
15 # Perform the ANOVA test
16 f_statistic, p_value = f_oneway(*groups)
17
18 # Print the results
19 print('F-statistic:', f_statistic)
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL COMMENTS
Python Debug Console
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\Amit Kumar\Desktop> & 'C:\Users\Amit Kumar\AppData\Local\Microsoft\WindowsApps\python3.10.exe' 'c:\Users\Amit Kumar\.vscode\extensions\ms-python.python-2023.8.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '49863' '--' 'c:\Users\Amit Kumar\Desktop\01.py'
F-statistic: 3.111194352801031
p-value: 0.05345420071280546
PS C:\Users\Amit Kumar\Desktop>
0 1 tabnine starter
Ln 7, Col 27 Spaces: 3 UTF-8 CRLF Python 3.10.11 64-bit (microsoft store)