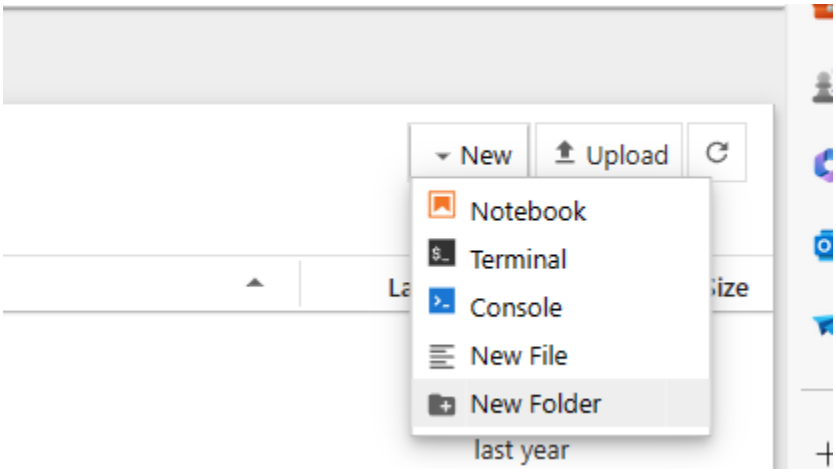
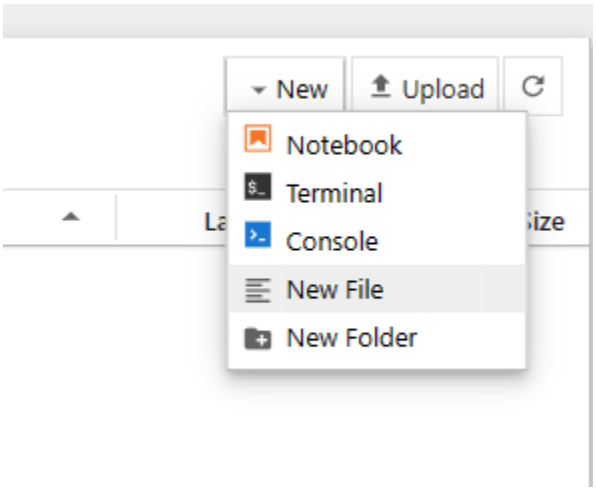
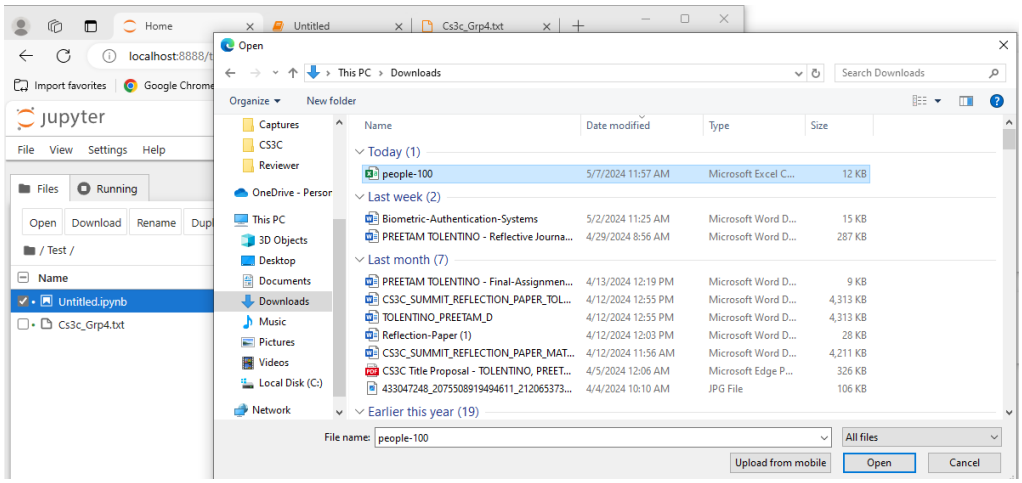
- Adding Folder



- Adding Text file

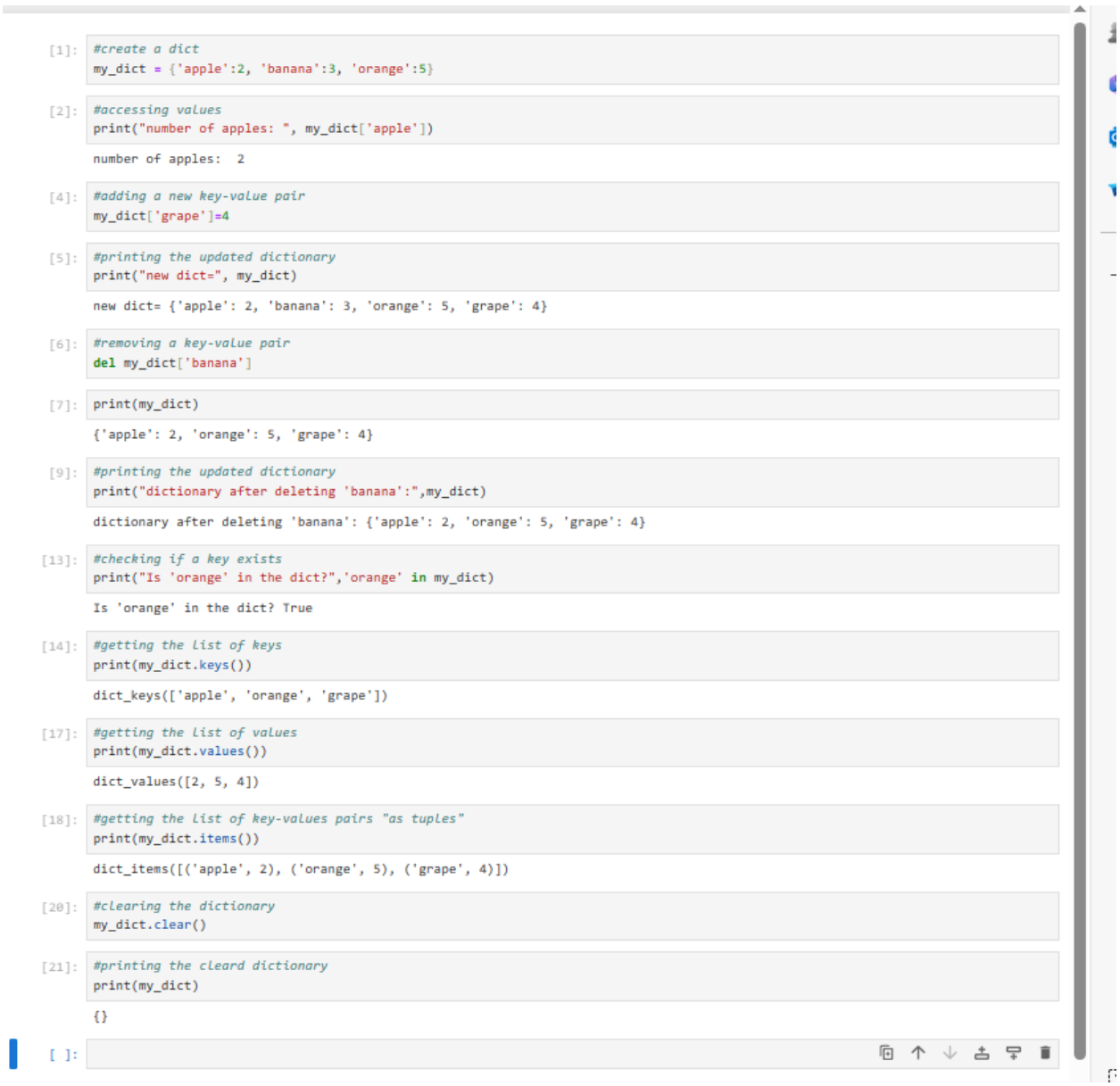


- CSV file for data analysis and visualization



●To write

and call dictionary methods.



```
[1]: #create a dict
     my_dict = {'apple':2, 'banana':3, 'orange':5}

[2]: #accessing values
     print("number of apples: ", my_dict['apple'])

     number of apples:  2

[4]: #adding a new key-value pair
     my_dict['grape']=4

[5]: #printing the updated dictionary
     print("new dict=", my_dict)

     new dict= {'apple': 2, 'banana': 3, 'orange': 5, 'grape': 4}

[6]: #removing a key-value pair
     del my_dict['banana']

[7]: print(my_dict)

     {'apple': 2, 'orange': 5, 'grape': 4}

[9]: #printing the updated dictionary
     print("dictionary after deleting 'banana':",my_dict)

     dictionary after deleting 'banana': {'apple': 2, 'orange': 5, 'grape': 4}

[13]: #checking if a key exists
      print("Is 'orange' in the dict?",'orange' in my_dict)

      Is 'orange' in the dict? True

[14]: #getting the list of keys
      print(my_dict.keys())

      dict_keys(['apple', 'orange', 'grape'])

[17]: #getting the list of values
      print(my_dict.values())

      dict_values([2, 5, 4])

[18]: #getting the list of key-values pairs "as tuples"
      print(my_dict.items())

      dict_items([('apple', 2), ('orange', 5), ('grape', 4)])

[20]: #clearing the dictionary
      my_dict.clear()

[21]: #printing the cleard dictionary
      print(my_dict)

      {}

[ ]:
```

- To create a directory using jupyter notebook.

[34]:
```
import os

# Define the directory name
directory_name = "my_directory"

# Create the directory
os.makedirs(directory_name)

# Check if the directory is created
if os.path.exists(directory_name):
    print(f"Directory '{directory_name}' created successfully.")
else:
    print(f"Failed to create directory '{directory_name}'.")
```

```
Directory 'my_directory' created successfully.
```

●To import libraries

[36]:
```
import pandas as pd

data = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})
print(data)
```

```
   A  B
0  1  4
1  2  5
2  3  6
```

[ ]:

- To use CSV file for data

[36]:
```
import pandas as pd

data = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})
print(data)
```

```
   A  B
0  1  4
1  2  5
2  3  6
```

[37]:
```
import pandas as pd

data = pd.read_csv("people-100.csv")

print (data.head())

print ("Summary statistics")
print (data.describe())
```

```
   Index          User Id First Name Last Name     Sex  \
0      1  88F7B33d2bcf9f5     Shelby   Terrell    Male
1      2  f90cD3E76f1A9b9    Phillip   Summers  Female
2      3  DbeAb8CcdfeFC2c   Kristine    Travis    Male
3      4  A31Bee3c201ef58    Yesenia  Martinez    Male
4      5  1bA7A3dc874da3c       Lori      Todd    Male

                        Email                    Phone Date of birth  \
0        elijah57@example.net  001-084-906-7849x73518    1945-10-26
1       bethany14@example.com       214.112.6044x4913    1910-03-24
2       bthompson@example.com            277.609.7938    1992-07-02
3   kaitlinkaiser@example.com            584.094.6111    2017-08-03
4  buchananmanuel@example.net       689-207-3558x7233    1938-12-01

            Job Title
0     Games developer
1      Phytotherapist
2           Homeopath
3   Market researcher
4  Veterinary surgeon
Summary statistics
            Index
count  100.000000
mean    50.500000
std     29.011492
min      1.000000
25%     25.750000
50%     50.500000
75%     75.250000
max    100.000000
```

[ ]:

- analysis and visualization

[41]:

	Index	User Id	First Name	Last Name	Sex	Email	Phone	Date of birth	Job Title
0	1	88F7B33d2bcf9f5	Shelby	Terrell	Male	elijah57@example.net	001-084-906-7849x73518	1945-10-26	Games developer
1	2	f90cD3E76f1A9b9	Phillip	Summers	Female	bethany14@example.com	214.112.6044x4913	1910-03-24	Phytotherapist
2	3	DbeAb8CcdfeFC2c	Kristine	Travis	Male	bthompson@example.com	277.609.7938	1992-07-02	Homeopath
3	4	A31Bee3c201ef58	Yesenia	Martinez	Male	kaitlinkaiser@example.com	584.094.6111	2017-08-03	Market researcher
4	5	1bA7A3dc874da3c	Lori	Todd	Male	buchananmanuel@example.net	689-207-3558x7233	1938-12-01	Veterinary surgeon
...	...	...	...	...	...	...	...	...	...
95	96	5eFda7caAeB260E	Dennis	Barnes	Female	bmartin@example.org	001-095-524-2112x257	1954-07-30	Software engineer
96	97	CCbFce93d3720bE	Steve	Patterson	Female	latasha46@example.net	001-865-478-5157	1932-04-29	Barrister
97	98	2fEc528aFAF0b69	Wesley	Bray	Male	regina11@example.org	995-542-3004x76800	1994-12-28	Police officer
98	99	Adc7ad986e4A1Fe	Summer	Oconnell	Female	alexiscantrell@example.org	001-273-685-6932x092	2012-04-12	Broadcast journalist
99	100	b8D0aD3490FC7e1	Mariah	Bernard	Male	pcopeland@example.org	(341)594-6554x44657	2016-11-15	IT sales professional

100 rows × 9 columns

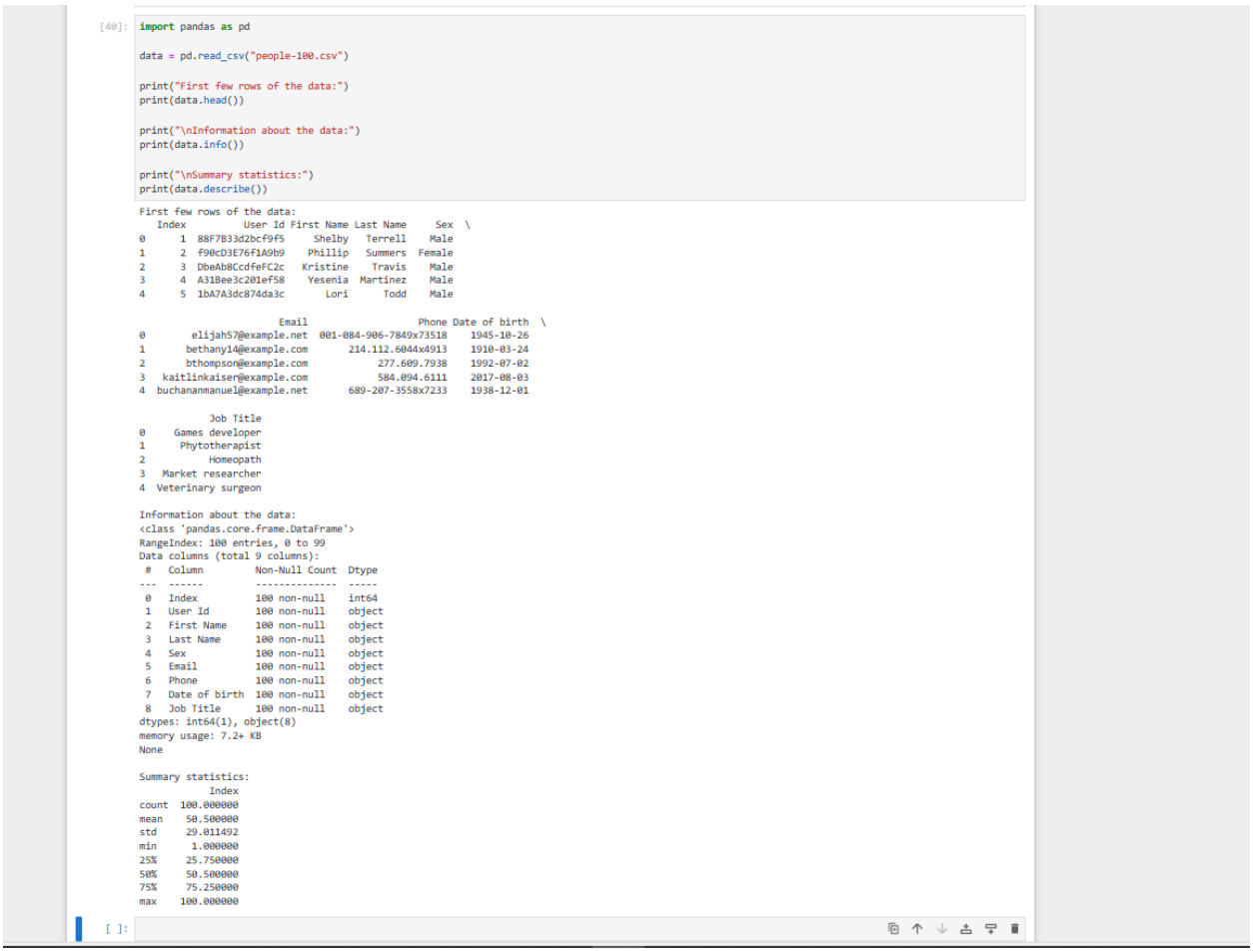
- Import libraries

```
0     Games developer
1      Phytotherapist
2            Homeopath
3   Market researcher
4  Veterinary surgeon
Summary statistics
             Index
count  100.000000
mean    50.500000
std     29.011492
min      1.000000
25%     25.750000
50%     50.500000
75%     75.250000
max    100.000000
```

```
[ ]: import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns
```

```
[ ]:
```

- Finding data



```
[40]: import pandas as pd

data = pd.read_csv("people-100.csv")

print("First few rows of the data:")
print(data.head())

print("\nInformation about the data:")
print(data.info())

print("\nSummary statistics:")
print(data.describe())
```

```
First few rows of the data:
   Index          User Id First Name Last Name     Sex  \
0      1  88F7B33d2bcf9f5     Shelby   Terrell    Male
1      2  f90cD3E76f1A9b9    Phillip   Summers  Female
2      3  DbeAb8CcdfeFC2c   Kristine    Travis    Male
3      4  A31Bee3c201ef58    Yesenia  Martinez    Male
4      5  1bA7A3dc874da3c       Lori      Todd    Male

                          Email                 Phone Date of birth  \
0       elijah57@example.net  001-084-906-7849x73518    1945-10-26
1      bethany14@example.com       214.112.6044x4913    1910-03-24
2      bthompson@example.com            277.609.7938    1992-07-02
3  kaitlinkaiser@example.com            584.094.6111    2017-08-03
4  buchananmanuel@example.net       689-207-3558x7233    1938-12-01

           Job Title
0    Games developer
1     Phytotherapist
2           Homeopath
3   Market researcher
4  Veterinary surgeon

Information about the data:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Index          100 non-null    int64
 1   User Id        100 non-null    object
 2   First Name     100 non-null    object
 3   Last Name      100 non-null    object
 4   Sex            100 non-null    object
 5   Email          100 non-null    object
 6   Phone          100 non-null    object
 7   Date of birth  100 non-null    object
 8   Job Title      100 non-null    object
dtypes: int64(1), object(8)
memory usage: 7.2+ KB
None

Summary statistics:
            Index
count  100.000000
mean    50.500000
std     29.011492
min      1.000000
25%     25.750000
50%     50.500000
75%     75.250000
max    100.000000
```

```
[ ]:
```

- Importing data

```
[43]: import pandas as pd

data = pd.read_csv("people-100.csv")

print(data.head())
```

```
   Index          User Id First Name Last Name     Sex  \
0      1  88F7B33d2bcf9f5     Shelby   Terrell    Male
1      2  f90cD3E76f1A9b9    Phillip   Summers  Female
2      3  DbeAb8CcdfeFC2c   Kristine    Travis    Male
3      4  A31Bee3c201ef58    Yesenia  Martinez    Male
4      5  1bA7A3dc874da3c       Lori      Todd    Male

                          Email                 Phone Date of birth  \
0       elijah57@example.net  001-084-906-7849x73518    1945-10-26
1      bethany14@example.com       214.112.6044x4913    1910-03-24
2      bthompson@example.com            277.609.7938    1992-07-02
3  kaitlinkaiser@example.com            584.094.6111    2017-08-03
4  buchananmanuel@example.net       689-207-3558x7233    1938-12-01

           Job Title
0    Games developer
1     Phytotherapist
2           Homeopath
3   Market researcher
4  Veterinary surgeon
```

- Data attributes

[45]:
```
import pandas as pd

data = pd.read_csv("people-100.csv")

print("Shape of the data:")
print(data.shape)

print("\nColumns of the data:")
print(data.columns)

print("\nData types of the columns:")
print(data.dtypes)

print("\nIndex of the data:")
print(data.index)

print("\nNumber of missing values in each column:")
print(data.isnull().sum())

print("\nSummary statistics:")
print(data.describe())

print("\nUnique values in column 'column_name':")
print(data[column_name].unique())
```

```
Shape of the data:
(100, 9)

Columns of the data:
Index(['Index', 'User Id', 'First Name', 'Last Name', 'Sex', 'Email', 'Phone',
       'Date of birth', 'Job Title'],
      dtype='object')

Data types of the columns:
Index             int64
User Id          object
First Name       object
Last Name        object
Sex              object
Email            object
Phone            object
Date of birth    object
Job Title        object
dtype: object

Index of the data:
RangeIndex(start=0, stop=100, step=1)

Number of missing values in each column:
Index            0
User Id          0
First Name       0
Last Name        0
Sex              0
Email            0
Phone            0
Date of birth    0
Job Title        0
dtype: int64

Summary statistics:
            Index
count  100.000000
mean    50.500000
std     29.011492
min      1.000000
25%     25.750000
50%     50.500000
75%     75.250000
max    100.000000
```