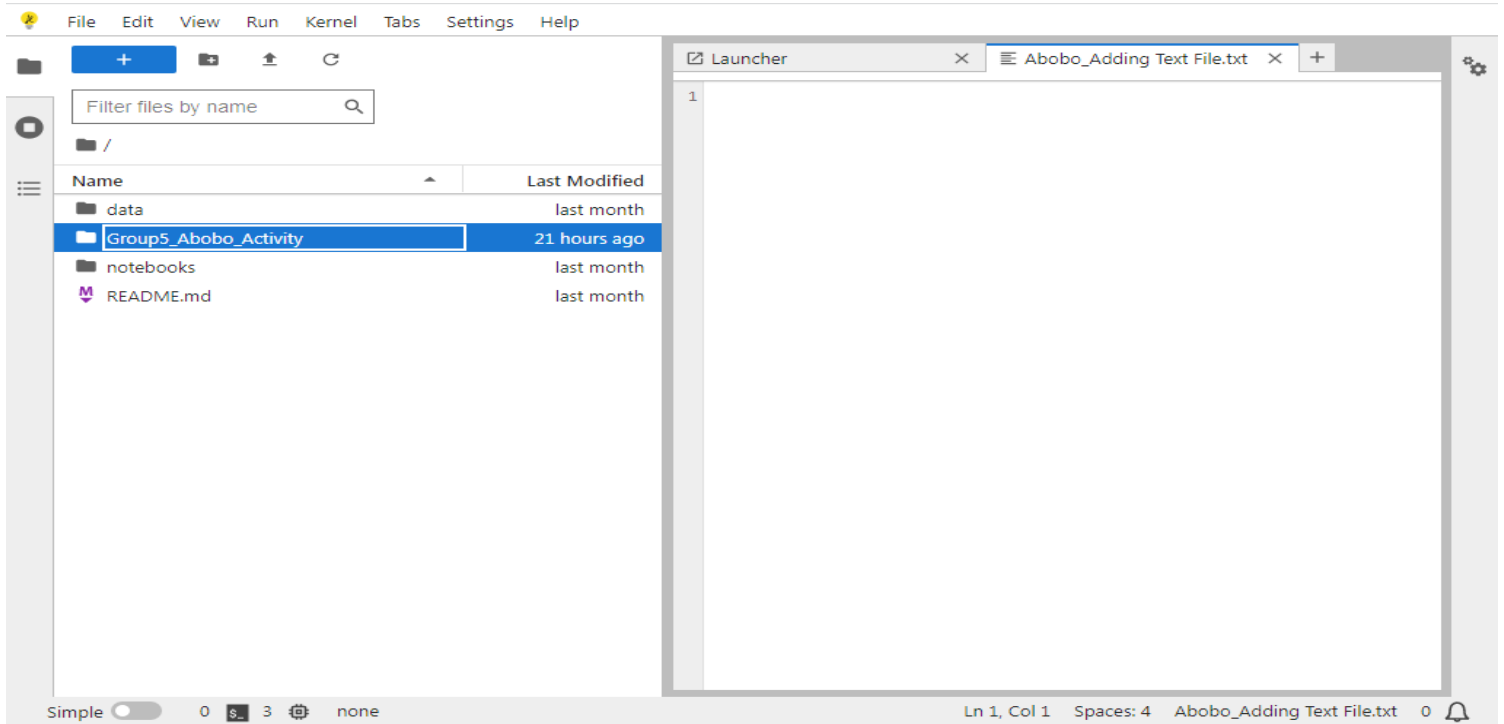
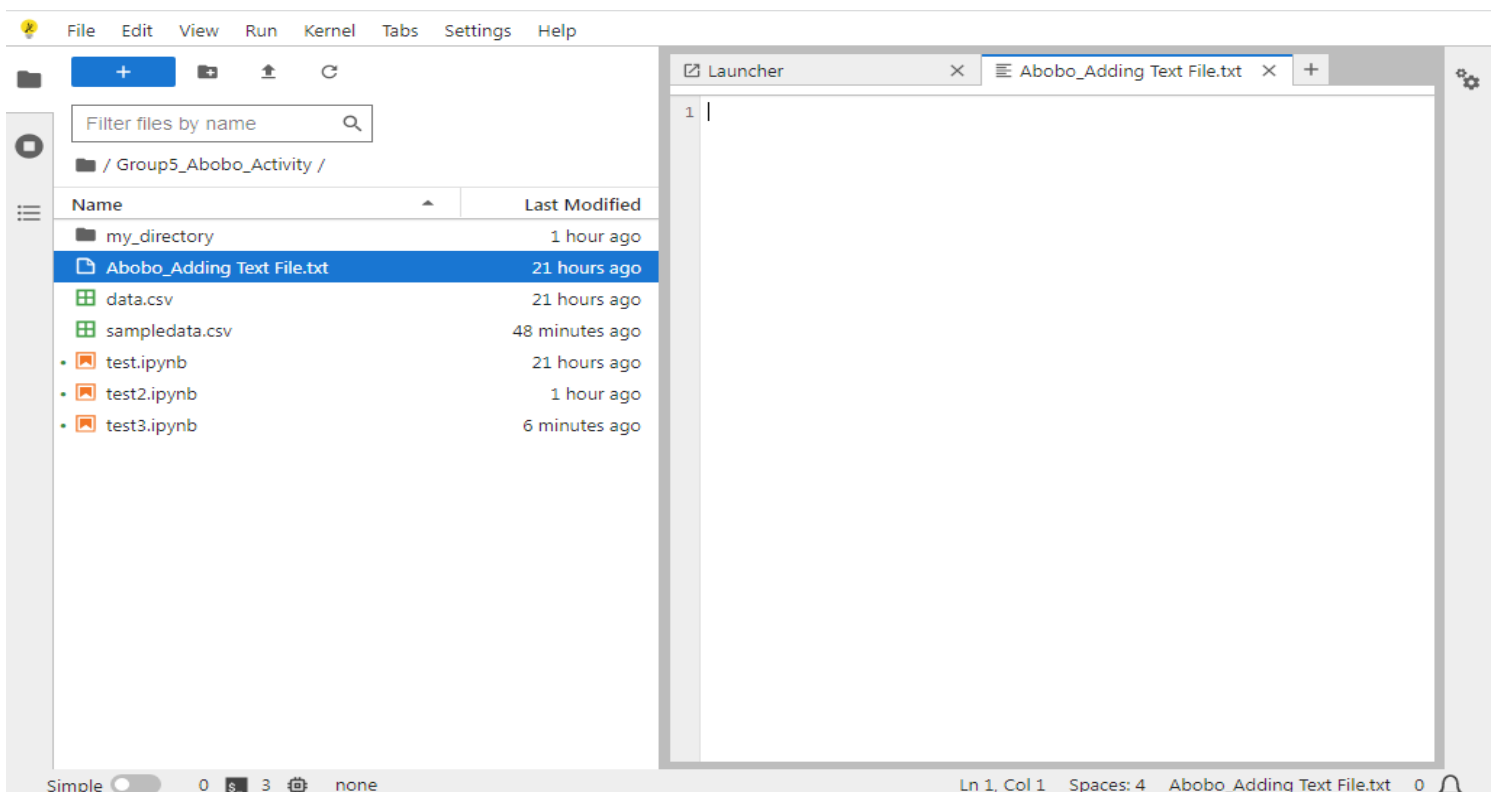


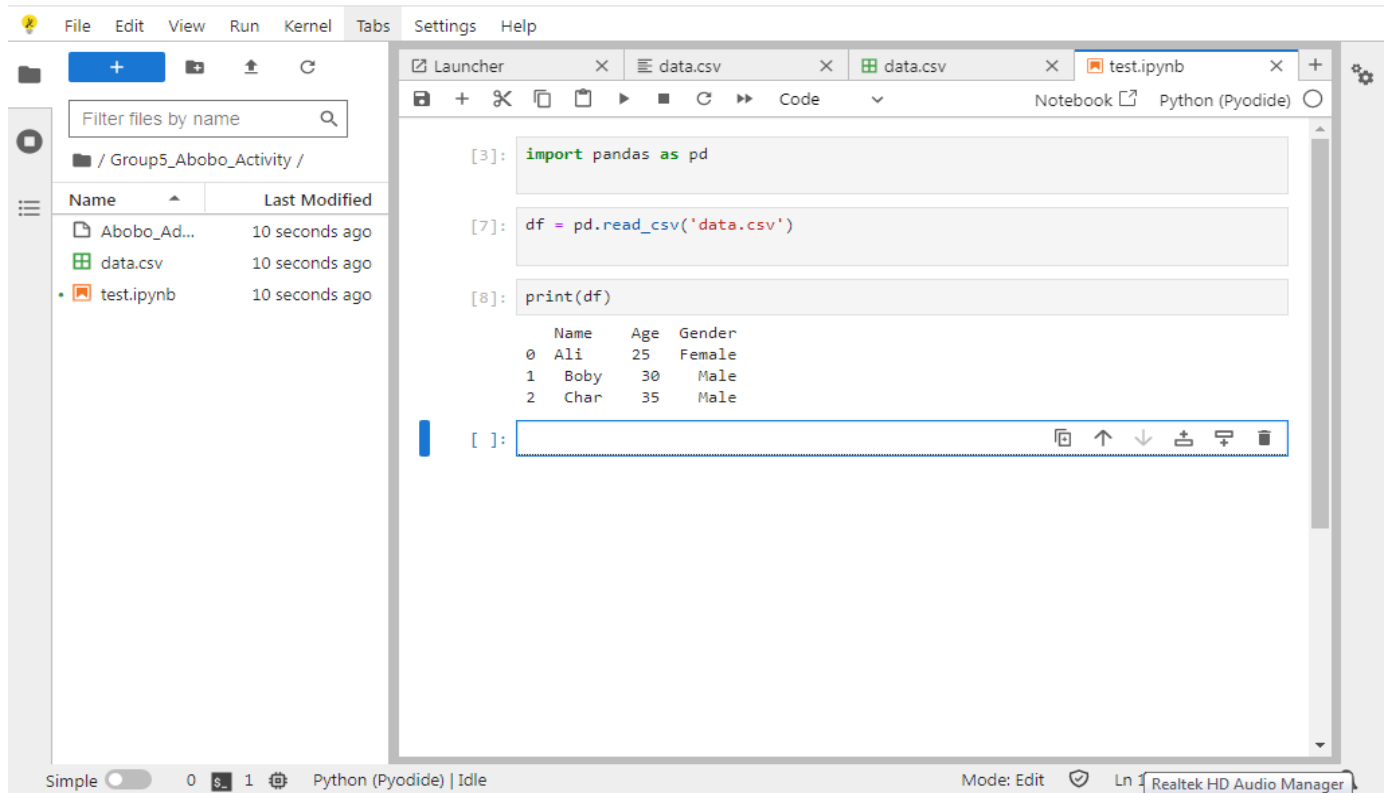
- Adding Folder



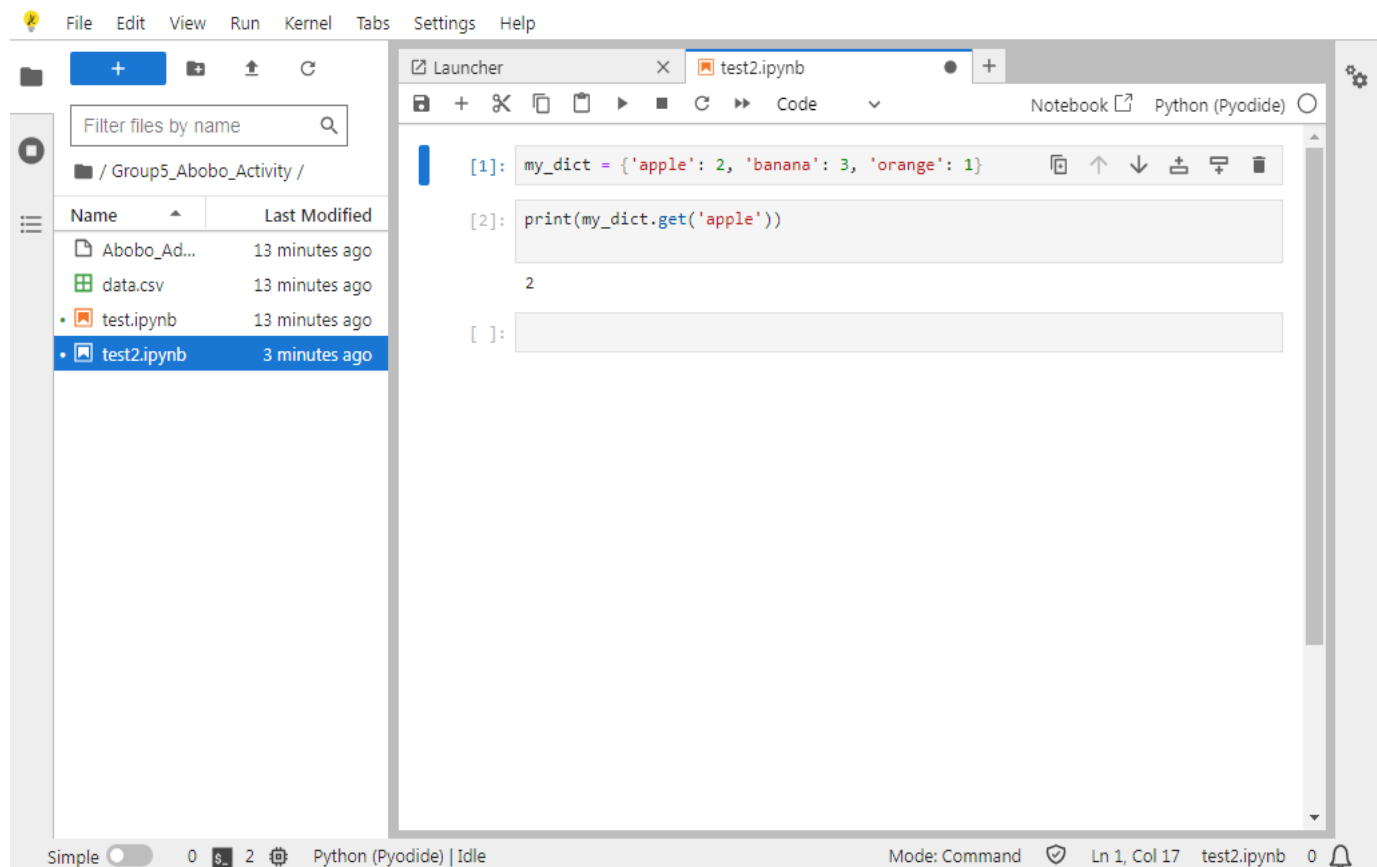
- Adding Text File



- CSV file for data analysis and visualization



- To write and call dictionary methods.



- To create a directory using Jupyter Notebook.

The screenshot shows the Jupyter Notebook interface. On the left, the file browser displays the directory structure under '/ Group5_Abobo_Activity /'. A table lists files and their last modified times:

Name	Last Modified
my_directory	4 seconds ago
Abobo_Ad...	19 hours ago
data.csv	19 hours ago
test.ipynb	19 hours ago
test2.ipynb	2 minutes ago

The main notebook area shows the following code cells:

```
[1]: my_dict = {'apple': 2, 'banana': 3, 'orange': 1}

[2]: print(my_dict.get('apple'))

2

[1]: import os

os.makedirs('my_directory')

[ ]:
```

The status bar at the bottom indicates 'Python (Pyodide) | Idle' and 'Mode: Command'.

- To import libraries

The screenshot shows the Jupyter Notebook interface with the file browser on the left. The table of files is updated:

Name	Last Modified
my_directory	12 minutes ago
Abobo_Ad...	20 hours ago
data.csv	20 hours ago
test.ipynb	20 hours ago
test2.ipynb	1 minute ago

The main notebook area shows the following code cells:

```
[1]: my_dict = {'apple': 2, 'banana': 3, 'orange': 1}

[2]: print(my_dict.get('apple'))

2

[1]: import os

os.makedirs('my_directory')

[6]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

[12]: import pandas as pd
my_array = np.array([1, 2, 3, 4, 5])
print(my_array)

[1 2 3 4 5]

[ ]:
```

The status bar at the bottom indicates 'Python (Pyodide) | Idle' and 'Mode: Command'.

- To use CSV file for data

The screenshot shows the JupyterLab interface. On the left, the file browser displays the directory structure under `/ Group5_Abobo_Activity /`. The files listed are `my_directory`, `Abobo_Adding Text Fi...`, `data.csv`, `sampledata.csv` (selected), and three `test.ipynb` files. On the right, the `sampledata.csv` file is open, showing a table with columns `Name`, `Age`, and `Salary`. The table contains five rows of data.

	Name	Age	Salary
1	Alice	25	50000
2	Bob	30	60000
3	Charlie	35	70000
4	David	40	80000
5	Eve	45	90000

The screenshot shows the JupyterLab interface with the `test3.ipynb` file open. The code editor displays the following Python code:

```
[15]: import pandas as pd
import matplotlib.pyplot as plt

[3]: df = pd.read_csv('sampledata.csv')

[4]: print(df)
```

The output of the `print(df)` command is displayed as a table:

	Name	Age	Salary
0	Alice	25	50000
1	Bob	30	60000
2	Charlie	35	70000
3	David	40	80000
4	Eve	45	90000

- Analysis and Visualization

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ Group5_Abobo_Activity /

Name	Last Modified
my_directory	49 minutes ago
Abobo_Ad...	20 hours ago
data.csv	20 hours ago
sampledata...	17 minutes ago
test.ipynb	20 hours ago
test2.ipynb	35 minutes ago
test3.ipynb	1 minute ago

Launcher X test2.ipynb X sampledatt X test3.ipynb X sampledatt X test.ipynb X

Code Notebook Python (Pyodide)

```
[18]: print("First 5 rows of the DataFrame:")
      print(df.head())
```

First 5 rows of the DataFrame:

	Name	Age	Salary
0	Alice	25	50000
1	Bob	30	60000
2	Charlie	35	70000
3	David	40	80000
4	Eve	45	90000

```
[19]: print("\nMissing Values:")
      print(df.isnull().sum())
```

Missing Values:

Name	0
Age	0
Salary	0
dtype:	int64

```
[20]: print("\nSummary Statistics:")
      print(df.describe())
```

Summary Statistics:

	Age	Salary
count	5.000000	5.000000
mean	35.000000	70000.000000
std	7.905694	15811.388301

Simple 0 3 Python (Pyodide) | Idle Mode: Command Ln 1, Col 1 test3.ipynb 0

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ Group5_Abobo_Activity /

Name	Last Modified
my_directory	49 minutes ago
Abobo_Ad...	20 hours ago
data.csv	20 hours ago
sampledata...	18 minutes ago
test.ipynb	20 hours ago
test2.ipynb	36 minutes ago
test3.ipynb	now

Launcher X test2.ipynb X sampledatt X test3.ipynb X sampledatt X test.ipynb X

Code Notebook Python (Pyodide)

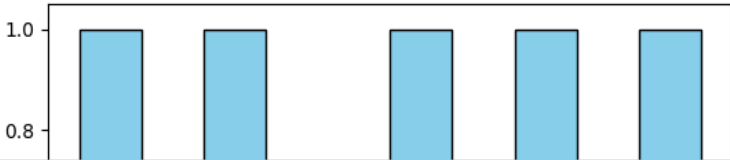
```
[20]: print("\nSummary Statistics:")
      print(df.describe())
```

Summary Statistics:

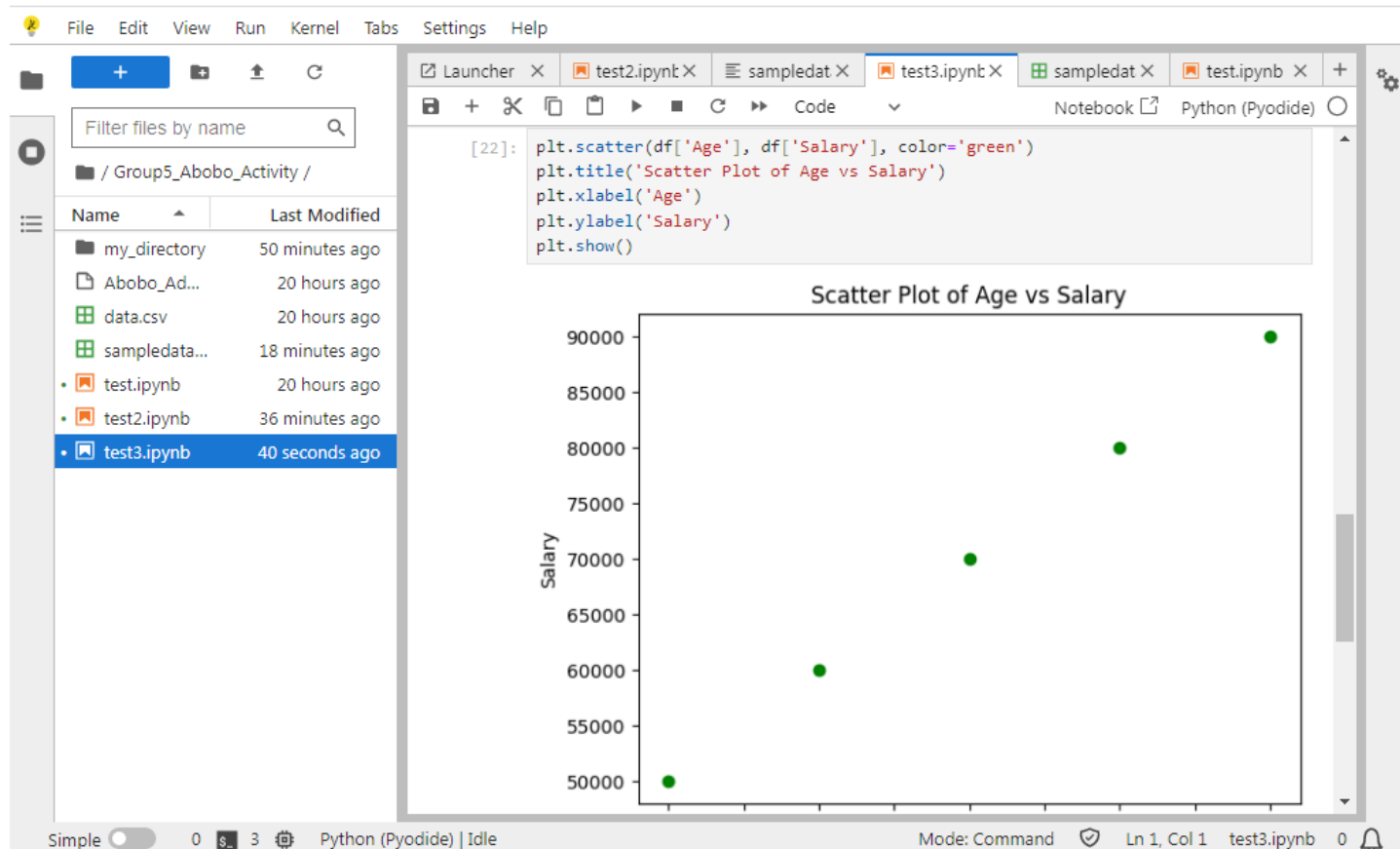
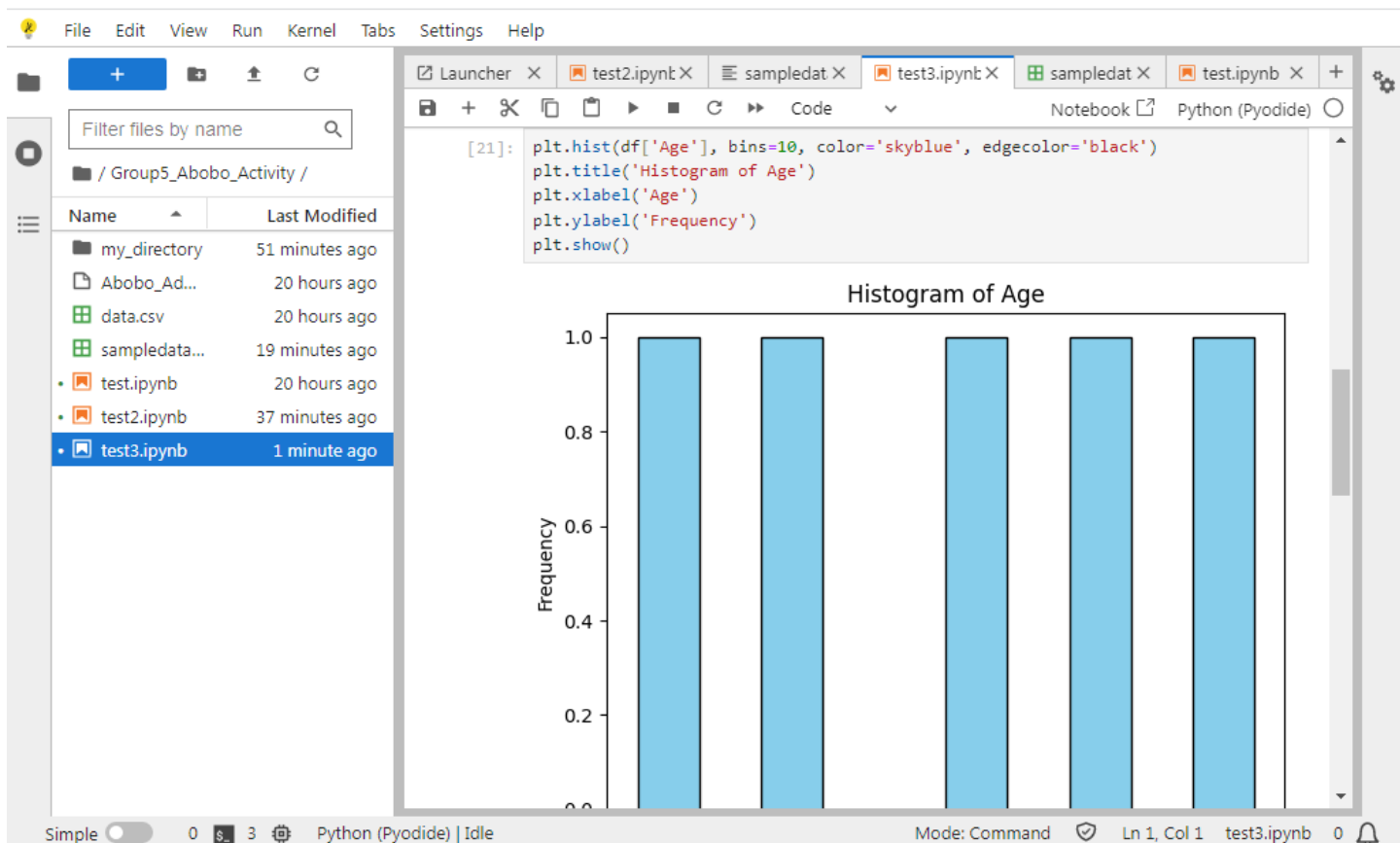
	Age	Salary
count	5.000000	5.000000
mean	35.000000	70000.000000
std	7.905694	15811.388301
min	25.000000	50000.000000
25%	30.000000	60000.000000
50%	35.000000	70000.000000
75%	40.000000	80000.000000
max	45.000000	90000.000000

```
[21]: plt.hist(df['Age'], bins=10, color='skyblue', edgecolor='black')
      plt.title('Histogram of Age')
      plt.xlabel('Age')
      plt.ylabel('Frequency')
      plt.show()
```

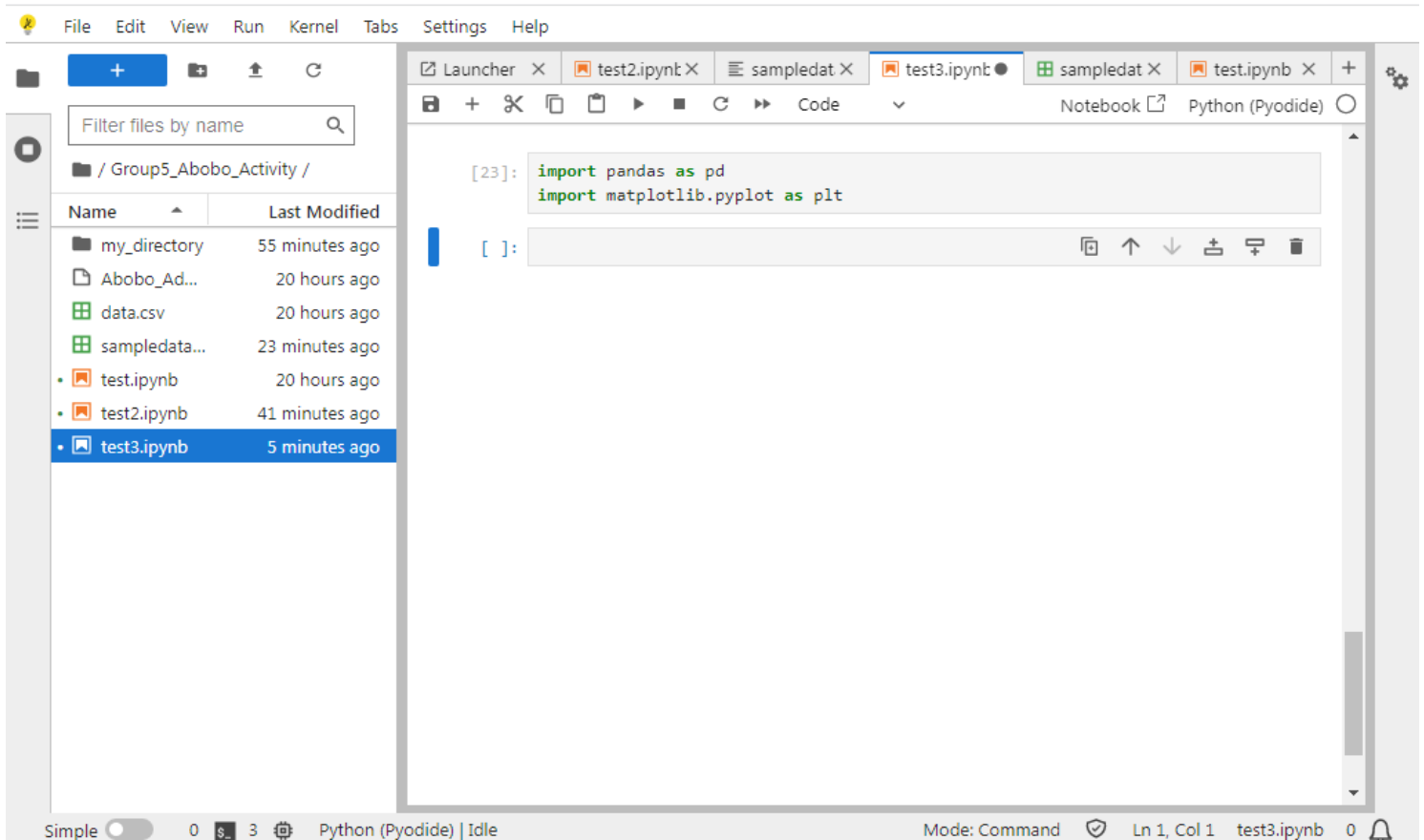
Histogram of Age



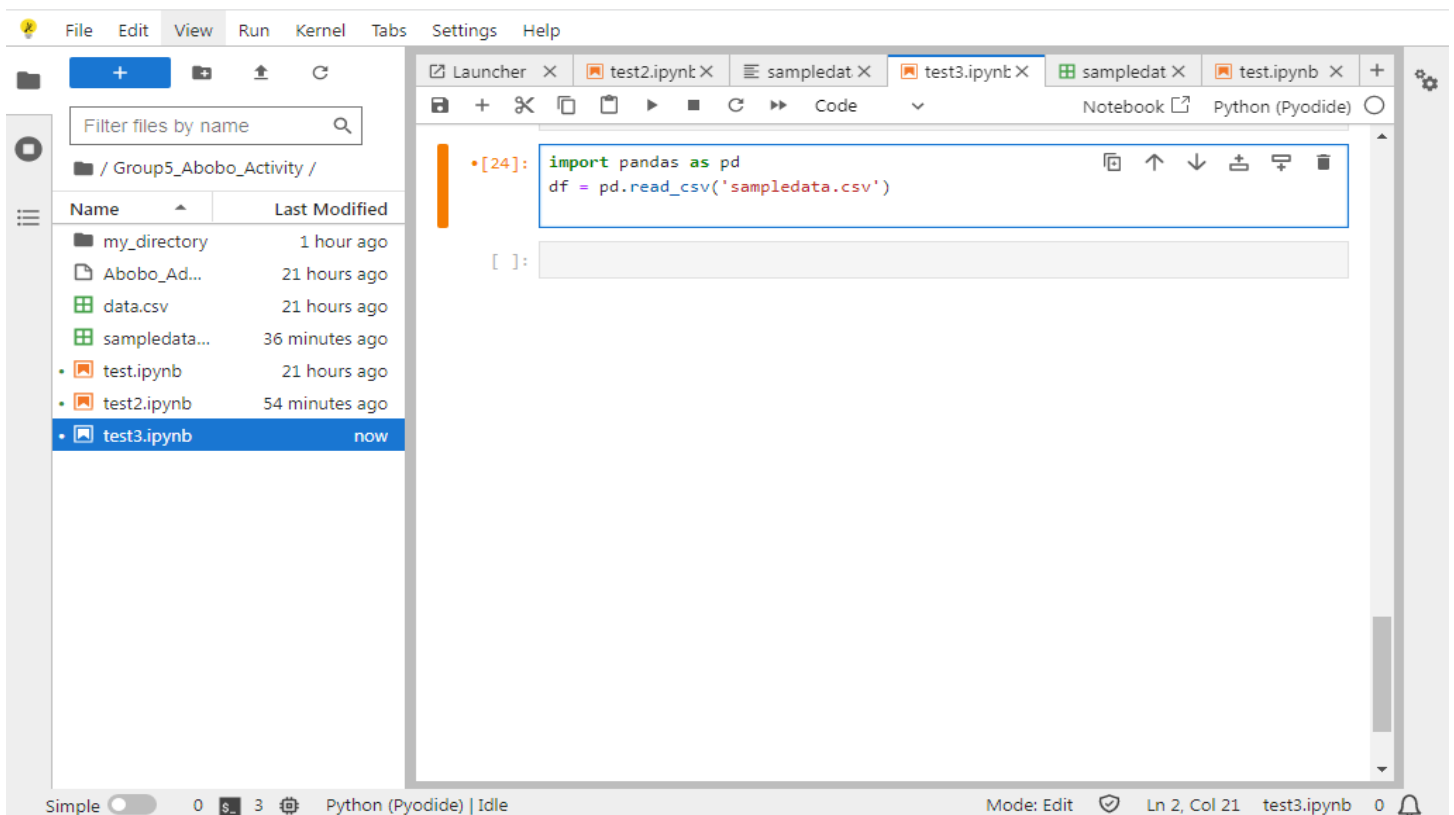
Simple 0 3 Python (Pyodide) | Idle Mode: Command Ln 1, Col 1 test3.ipynb 0



- Import libraries



- Finding data



- Importing data

The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The file explorer displays a directory structure with files like 'my_directory', 'Abobo_Ad...', 'data.csv', 'sampledata...', 'test.ipynb', 'test2.ipynb', and 'test3.ipynb'. The code editor shows a single cell with the following code:

```
[31]: import pandas as pd
df = pd.read_csv('sampledata.csv')
print(df)
```

The output of the code is a DataFrame with 5 rows and 3 columns (Name, Age, Salary):

	Name	Age	Salary
0	Alice	25	50000
1	Bob	30	60000
2	Charlie	35	70000
3	David	40	80000
4	Eve	45	90000

The interface includes a menu bar (File, Edit, View, Run, Kernel, Tabs, Settings, Help) and a status bar at the bottom showing 'Mode: Command' and 'Ln 1, Col 1'.

- Data attributes

The screenshot shows a Jupyter Notebook interface with multiple tabs open. The active tab is 'test3.ipynb', which contains three code cells. The first cell prints the data types of the DataFrame:

```
[27]: print("Data Types:")
print(df.dtypes)
```

The output is:

```
Data Types:
Name      object
Age       int64
Salary    int64
dtype: object
```

The second cell prints the column names:

```
[29]: print("\nColumn Names:")
print(df.columns)
```

The output is:

```
Column Names:
Index(['Name', 'Age', 'Salary'], dtype='object')
```

The third cell prints the missing values:

```
[30]: print("\nMissing Values:")
print(df.isnull().sum())
```

The output is:

```
Missing Values:
Name      0
Age       0
Salary    0
dtype: int64
```

The interface includes a menu bar (File, Edit, View, Run, Kernel, Tabs, Settings, Help) and a status bar at the bottom showing 'Mode: Command' and 'Ln 1, Col 1'.