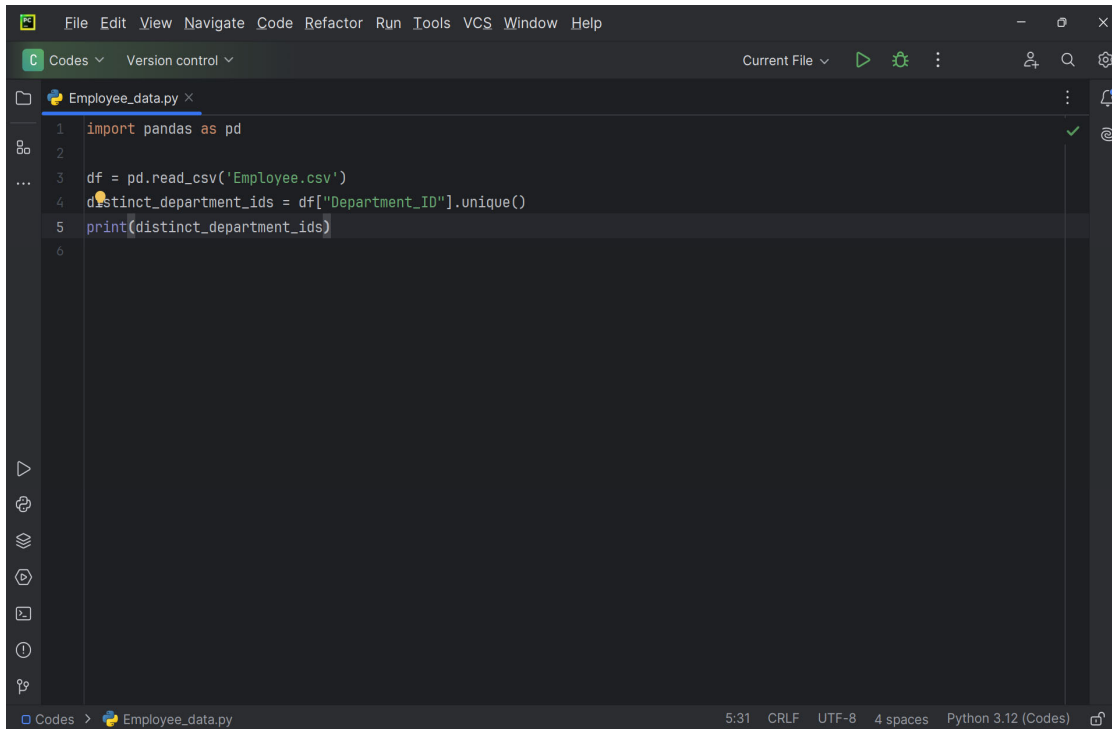


Experiment 1

Aim:

Pandas Program to select distinct department id from employees file.

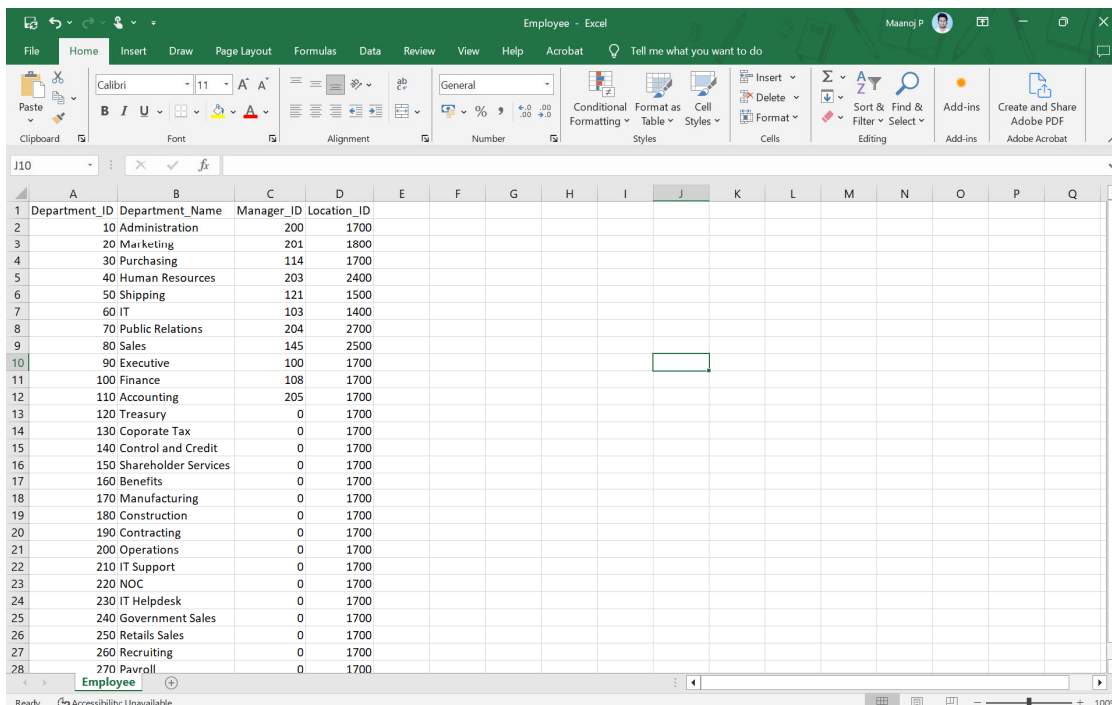
Code:



```
1 import pandas as pd
2
3 df = pd.read_csv('Employee.csv')
4 distinct_department_ids = df["Department_ID"].unique()
5 print(distinct_department_ids)
6
```

The screenshot shows a code editor with a dark theme. The file is named 'Employee_data.py'. The code reads a CSV file 'Employee.csv' and prints the unique values of the 'Department_ID' column. The status bar at the bottom indicates the file is encoded in UTF-8, uses 4 spaces for indentation, and is running on Python 3.12.

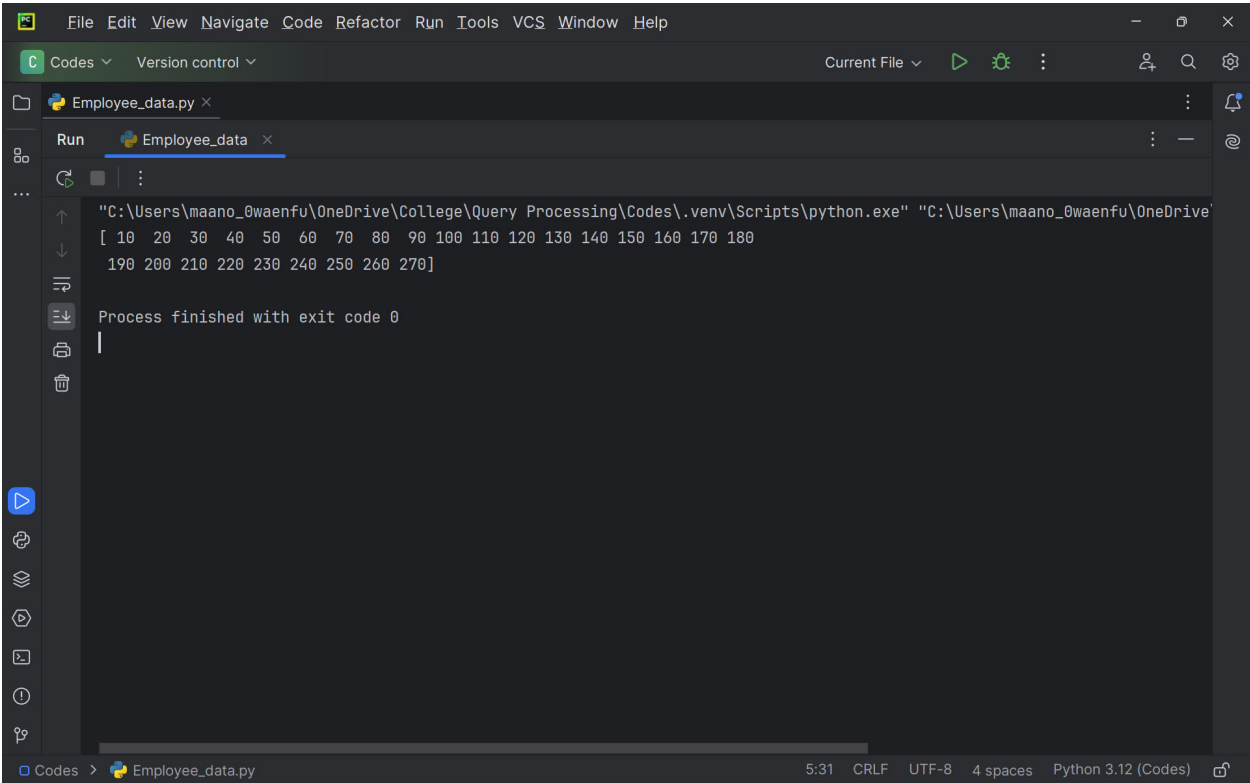
Input:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Department_ID	Department_Name	Manager_ID	Location_ID													
2	10	Administration	200	1700													
3	20	Marketing	201	1800													
4	30	Purchasing	114	1700													
5	40	Human Resources	203	2400													
6	50	Shipping	121	1500													
7	60	IT	103	1400													
8	70	Public Relations	204	2700													
9	80	Sales	145	2500													
10	90	Executive	100	1700													
11	100	Finance	108	1700													
12	110	Accounting	205	1700													
13	120	Treasury	0	1700													
14	130	Corporate Tax	0	1700													
15	140	Control and Credit	0	1700													
16	150	Shareholder Services	0	1700													
17	160	Benefits	0	1700													
18	170	Manufacturing	0	1700													
19	180	Construction	0	1700													
20	190	Contracting	0	1700													
21	200	Operations	0	1700													
22	210	IT Support	0	1700													
23	220	NOC	0	1700													
24	230	IT Helpdesk	0	1700													
25	240	Government Sales	0	1700													
26	250	Retails Sales	0	1700													
27	260	Recruiting	0	1700													
28	270	Pavroll	0	1700													

The screenshot shows an Excel spreadsheet titled 'Employee - Excel'. The data is organized into columns: Department_ID, Department_Name, Manager_ID, and Location_ID. The rows list various departments and their associated managers and locations. The status bar at the bottom indicates the file is named 'Employee' and is ready for use.

Output:



```
"C:\Users\maano_0waenfu\OneDrive\College\Query Processing\Codes\.venv\Scripts\python.exe" "C:\Users\maano_0waenfu\OneDrive\College\Query Processing\Codes\Employee_data.py"
[ 10  20  30  40  50  60  70  80  90 100 110 120 130 140 150 160 170 180
 190 200 210 220 230 240 250 260 270]

Process finished with exit code 0
```

Experiment 2

Aim:

To develop a Pandas program to display the ID for those employees who did two or more jobs in the past

Code:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Codes Version control Current File
Employee_data.py Experience.py x
1 import pandas as pd
2
3 df = pd.read_csv('Experience.csv')
4
5 employee_counts = df['Employee_id'].value_counts()
6 multiple_jobs = employee_counts[employee_counts > 1].index
7 print("Employee IDs: ",multiple_jobs.values)
8
```

Input:

Experience - Excel

File Home Insert Draw Page Layout Formulas Data Review View Help Tell me what you want to do

Clipboard Font Alignment Number Conditional Formatting Styles Cell Styles Cells Editing

Calibri 11 A A' B I U % 9 0.00 0.00 0.00

General

Sort & Find & Filter & Select Editing

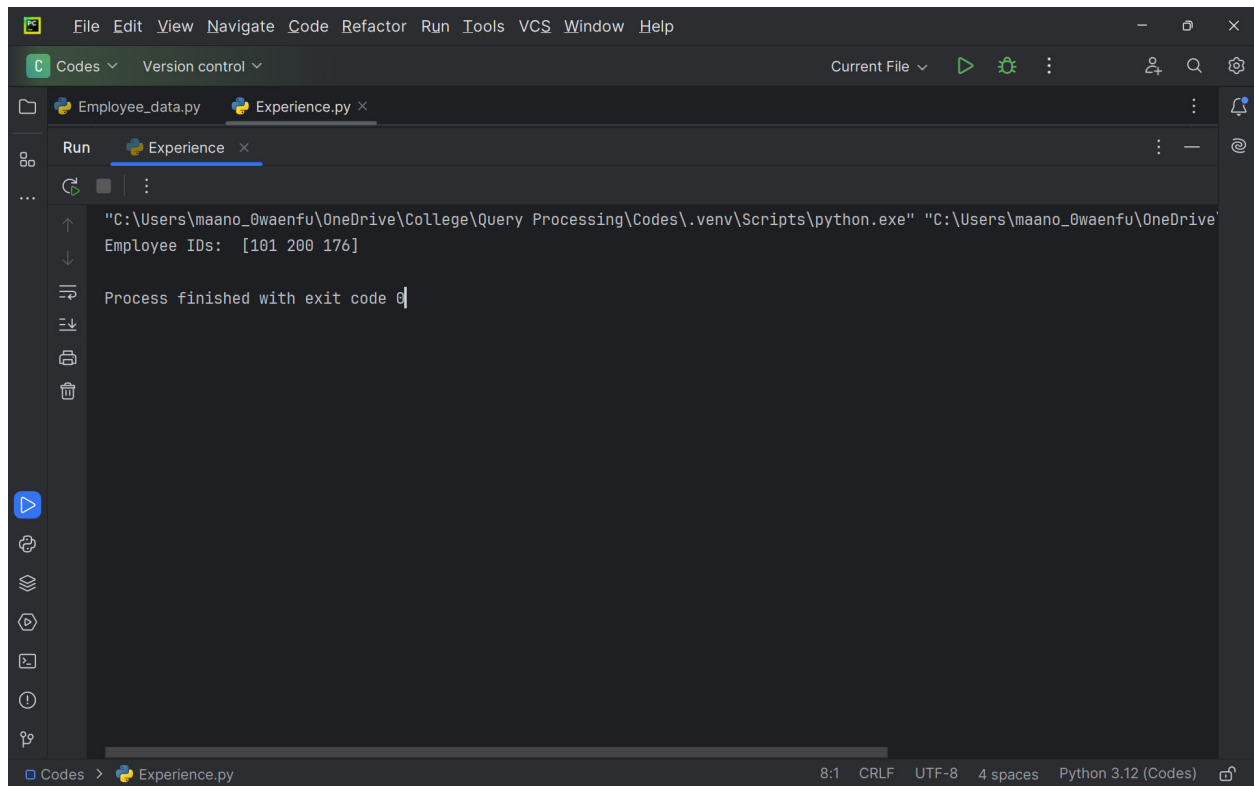
Add-ins Create and Share Adobe PDF

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	Employee_id	Start_date	End_date	Job_id	Department_id											
1	102	13-01-2001	24-07-2006	IT_PROG	60											
2	101	21-09-1997	27-10-2002	AC_ACCOUNT	110											
3	101	28-10-2001	15-03-2005	AC_MGR	110											
4	201	17-02-2004	19-12-2007	MK_REP	20											
5	114	24-03-2006	31-12-2007	ST_CLERK	50											
6	122	01-01-2007	31-12-2007	ST_CLERK	50											
7	200	17-09-1995	17-06-2001	AD_ASST	90											
8	176	24-03-2006	31-12-2006	SA_REP	80											
9	176	01-01-2007	31-12-2007	SA_MAN	80											
10	200	01-07-2002	31-12-2006	AC_ACCOUNT	90											
11																
12																
13																
14																
15																
16																
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																

Experience

Ready Accessibility: Unavailable

Output:



The screenshot shows a code editor with a dark theme. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The left sidebar has icons for Explorer, Search, and Run and Debug. The main editor area has two tabs: Employee_data.py and Experience.py. The Run and Debug console is open, showing the output of the Experience.py script. The output text is: "C:\Users\maano_0waenfu\OneDrive\College\Query Processing\Codes\.venv\Scripts\python.exe" "C:\Users\maano_0waenfu\OneDrive\Employee IDs: [101 200 176]". Below this, it says "Process finished with exit code 0". The status bar at the bottom shows the file is Experience.py, with 8:1 CRLF UTF-8 4 spaces Python 3.12 (Codes).

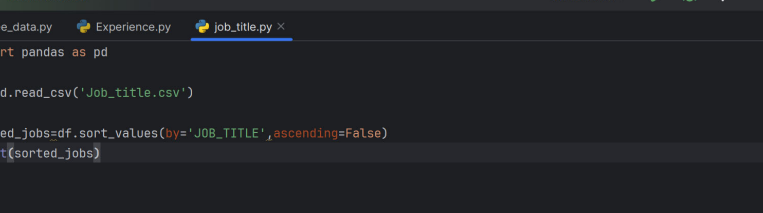
```
"C:\Users\maano_0waenfu\OneDrive\College\Query Processing\Codes\.venv\Scripts\python.exe" "C:\Users\maano_0waenfu\OneDrive\Employee IDs: [101 200 176]"
Process finished with exit code 0
```

Experiment 3

Aim:

To develop a Pandas Program to display the details of jobs in descending sequence on job title.

Code:



The screenshot shows a code editor with a dark theme. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. Below the menu bar, there are tabs for Codes, Version control, and Current File. The main editor area displays a Python script in a file named job_title.py. The script contains six lines of code: importing pandas as pd, reading a CSV file named 'Job_title.csv', sorting the data by 'JOB_TITLE' in ascending order, and printing the sorted jobs. The left sidebar shows a file explorer with three files: Employee_data.py, Experience.py, and job_title.py. The bottom status bar indicates the current file is job_title.py, the time is 6:19, and the encoding is CRLF, UTF-8, 4 spaces, Python 3.12 (Codes).

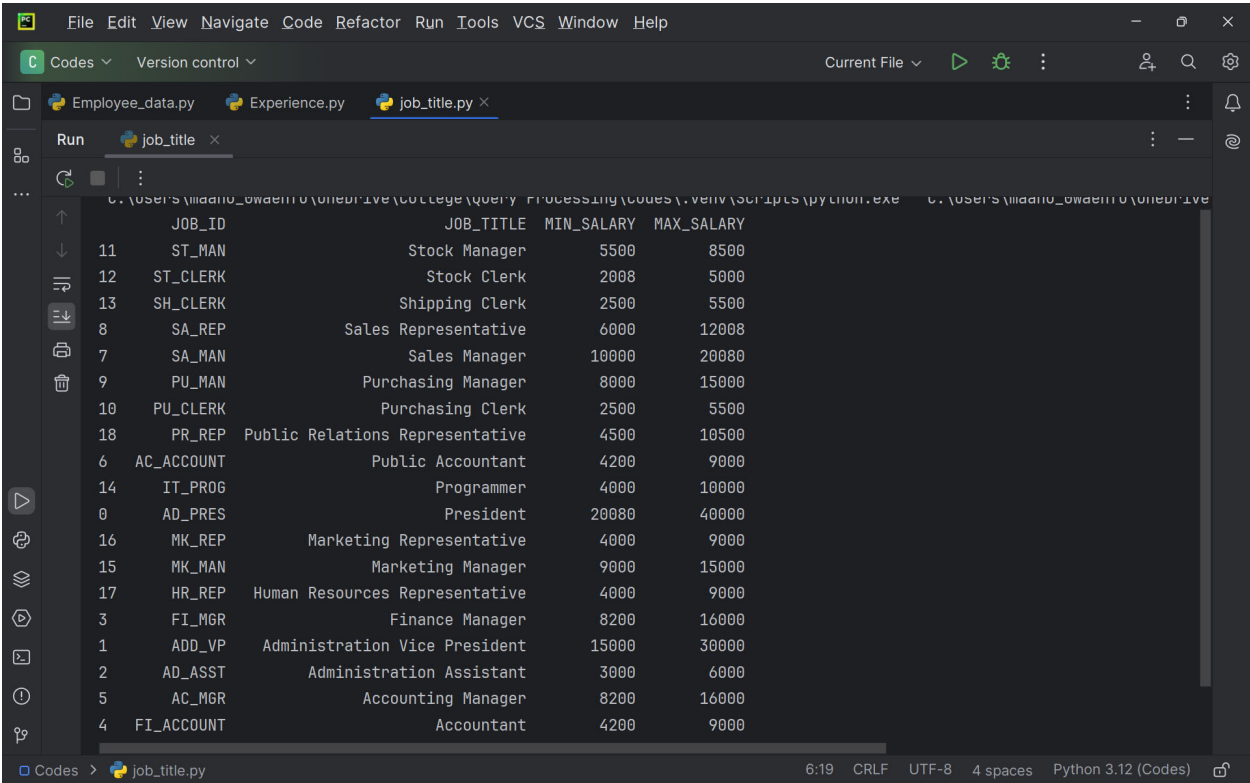
```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Codes Version control Current File
Employee_data.py Experience.py job_title.py x
1 import pandas as pd
2
3 df=pd.read_csv('Job_title.csv')
4
5 sorted_jobs=df.sort_values(by='JOB_TITLE',ascending=False)
6 print(sorted_jobs)
```

6:19 CRLF UTF-8 4 spaces Python 3.12 (Codes)

Input:

[illegible]

Output:



```
C:\Users\maano_owadent\OneDrive\College\query_processing\codes\.venv\scripts\python.exe C:\Users\maano_owadent\OneDrive\College\query_processing\codes\job_title.py
```

	JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
11	ST_MAN	Stock Manager	5500	8500
12	ST_CLERK	Stock Clerk	2008	5000
13	SH_CLERK	Shipping Clerk	2500	5500
8	SA_REP	Sales Representative	6000	12008
7	SA_MAN	Sales Manager	10000	20080
9	PU_MAN	Purchasing Manager	8000	15000
10	PU_CLERK	Purchasing Clerk	2500	5500
18	PR_REP	Public Relations Representative	4500	10500
6	AC_ACCOUNT	Public Accountant	4200	9000
14	IT_PROG	Programmer	4000	10000
0	AD PRES	President	20080	40000
16	MK_REP	Marketing Representative	4000	9000
15	MK_MAN	Marketing Manager	9000	15000
17	HR_REP	Human Resources Representative	4000	9000
3	FI_MGR	Finance Manager	8200	16000
1	ADD_VP	Administration Vice President	15000	30000
2	AD_ASST	Administration Assistant	3000	6000
5	AC_MGR	Accounting Manager	8200	16000
4	FI_ACCOUNT	Accountant	4200	9000

Codes > job_title.py 6:19 CRLF UTF-8 4 spaces Python 3.12 (Codes)

Experiment 4

Aim:

To develop a Pandas program to create a line plot of the historical stock prices of Alphabet Inc. between two specific dates

Code:

The image shows a VS Code editor window with a Python file named `line_plot.py`. The code is as follows:

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 df=pd.read_csv('Alphabet.csv')
5
6 start_date='01-04-2020'
7 end_date='15-04-2020'
8
9 mask=(df['Date'] >= start_date) & (df['Date'] <= end_date)
10 filtered_df=df.loc[mask]
11
12 plt.plot(*args: filtered_df['Date'],filtered_df['Close'])
13 plt.xlabel('Date')
14 plt.ylabel('Close Price')
15 plt.title('Historical Stock Prices')
16 plt.show()

```

The status bar at the bottom indicates the file is encoded in UTF-8, uses 4 spaces for indentation, and is running on Python 3.12 (Codes).

Input:

FileHomeInsertDrawPage LayoutFormulasDataReviewViewHelpAcrobatTell me what you want to do

Clipboard

Paste

Font

Calibri11A⁺⁺

B

I

U

Alignment

Number

General

%

0.00

0.00

Conditional Formatting

Format as Table

Cell Styles

Insert

Delete

Format

Sum

Sort & Filter

Find & Select

Add-ins

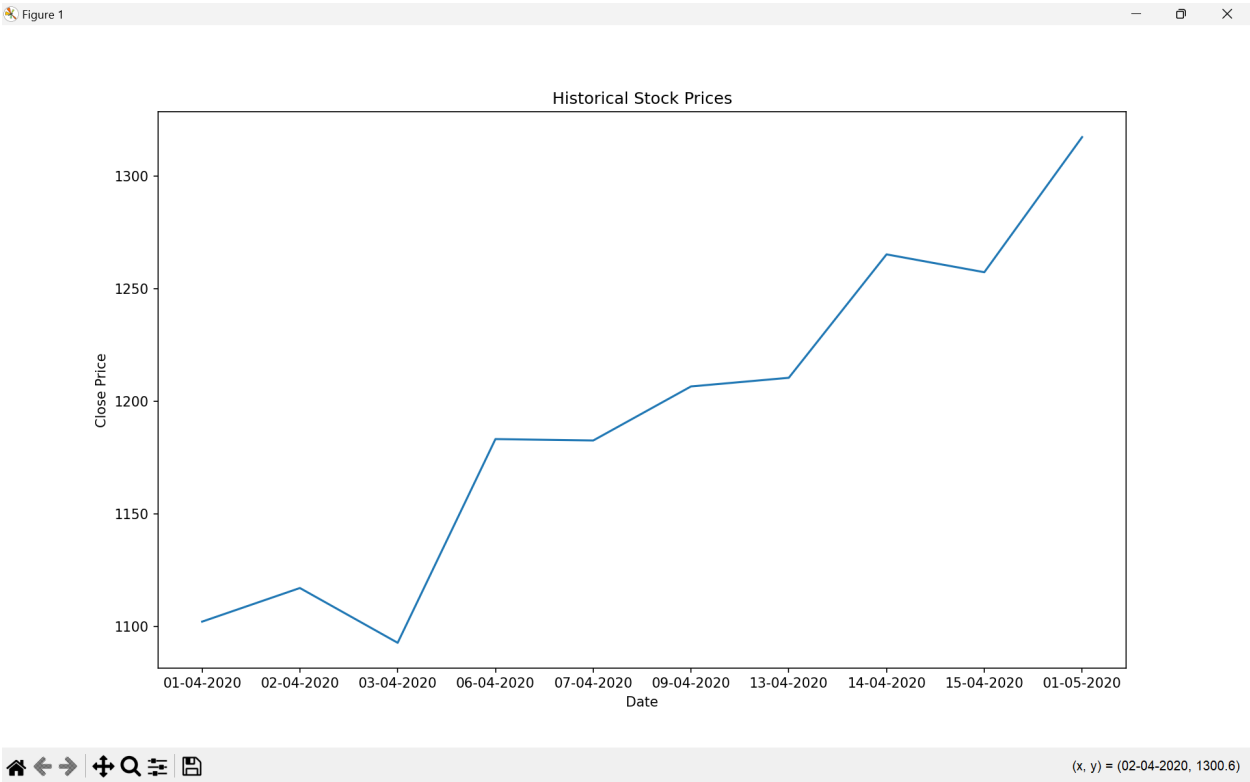
Create and Share Adobe PDF

Add-ins

K7

<

Output:

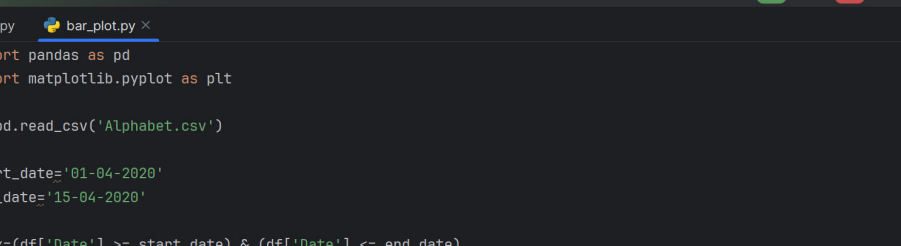


Experiment 5

Aim:

To develop a Pandas program to create a bar plot of the trading volume of Alphabet Inc. between two specific dates

Code:



The screenshot shows the VS Code editor interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The left sidebar contains icons for Explorer, Search, and Run and Debug. The main editor area displays a Python script in a dark theme. The script imports pandas and matplotlib, reads a CSV file, filters data by date, and creates a bar chart. The status bar at the bottom shows the current file is bar_plot.py, the time is 16:11, and the encoding is UTF-8.

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Codes Version control Current File
line_plot.py bar_plot.py x
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 df=pd.read_csv('Alphabet.csv')
5
6 start_date='01-04-2020'
7 end_date='15-04-2020'
8
9 mask=(df['Date'] >= start_date) & (df['Date'] <= end_date)
10 filtered_df=df.loc[mask]
11
12 plt.bar(filtered_df['Date'],filtered_df['Volume'])
13 plt.xlabel('Date')
14 plt.ylabel('Trading Volume')
15 plt.title('Trading Volume')
16 plt.show()
```

Codes > bar_plot.py 16:11 CRLF UTF-8 4 spaces Python 3.12 (Codes)

Input:

Alphabet - Excel

File Home Insert Draw Page Layout Formulas Data Review View Help Acrobat Tell me what you want to do

Paste Font Alignment Number Conditional Formatting Styles Cell Styles Cells Editing Add-ins Create and Share Adobe PDF

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
Date	Open	High	Low	Close	Adj Close	Volume												
01-04-2020	1124	1129.42	1093.49	1102.1	1102.1	2598500												
02-04-2020	1100	1122.77	1093.13	1117.03	1117.03	2820500												
03-04-2020	1114.71	1118.79	1075.08	1092.7	1092.7	2568700												
06-04-2020	1133	1190.75	1125	1183.19	1183.19	3166000												
07-04-2020	1217.01	1220.78	1177.25	1182.56	1182.56	3081000												
08-04-2020	1203.1	1214.9	1183.95	1207	1207	2016700												
09-04-2020	1218.18	1221.99	1192.42	1206.57	1206.57	2701400												
13-04-2020	1201.5	1214.52	1182.33	1210.41	1210.41	1935100												
14-04-2020	1239.97	1275.75	1228.54	1265.23	1265.23	3167900												
15-04-2020	1246.51	1275.11	1234	1257.3	1257.3	2111800												
16-04-2020	1267.14	1273.36	1238.2	1257.43	1257.43	2894800												
17-04-2020	1281.7	1290	1266	1279	1279	2552500												
20-04-2020	1269.89	1276.82	1256.44	1261.15	1261.15	1764600												
21-04-2020	1242.71	1250	1205	1212.16	1212.16	2482400												
22-04-2020	1241.11	1279.88	1237.2	1258.41	1258.41	2314800												
23-04-2020	1265.74	1288.15	1260.53	1271.17	1271.17	1710100												
24-04-2020	1255	1277.71	1244	1276.6	1276.6	1870100												
27-04-2020	1292	1294.1	1265.06	1270.86	1270.86	2209300												
28-04-2020	1283.2	1284.76	1230.38	1232.59	1232.59	4035000												
29-04-2020	1345	1360.15	1326.73	1342.18	1342.18	5417900												
30-04-2020	1331.36	1350	1321.5	1346.7	1346.7	2792100												
01-05-2020	1324.09	1351.43	1309.66	1317.32	1317.32	2443600												

Alphabet Accessible Unavailable

Output:

