In [2]:
```
# Loading the library
import pandas as pd
```

In [4]:
```
# Loading the dataset
tips_df = pd.read_csv("tips.csv")
```

In [6]:
```
tips_df.head()
```

Out[6]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [37]:
```
# filtering df where total_bill greater than 30
df = tips_df[tips_df['total_bill'] >30]
```

In [39]:
```
df
```

Out[39]:

	total_bill	tip	sex	smoker	day	time	size
11	35.26	5.00	Female	No	Sun	Dinner	4
23	39.42	7.58	Male	No	Sat	Dinner	4
39	31.27	5.00	Male	No	Sat	Dinner	3
44	30.40	5.60	Male	No	Sun	Dinner	4
47	32.40	6.00	Male	No	Sun	Dinner	4
52	34.81	5.20	Female	No	Sun	Dinner	4
56	38.01	3.00	Male	Yes	Sat	Dinner	4
59	48.27	6.73	Male	No	Sat	Dinner	4
83	32.68	5.00	Male	Yes	Thur	Lunch	2
85	34.83	5.17	Female	No	Thur	Lunch	4
95	40.17	4.73	Male	Yes	Fri	Dinner	4
102	44.30	2.50	Female	Yes	Sat	Dinner	3
112	38.07	4.00	Male	No	Sun	Dinner	3
141	34.30	6.70	Male	No	Thur	Lunch	6
142	41.19	5.00	Male	No	Thur	Lunch	5
156	48.17	5.00	Male	No	Sun	Dinner	6
167	31.71	4.50	Male	No	Sun	Dinner	4
170	50.81	10.00	Male	Yes	Sat	Dinner	3
173	31.85	3.18	Male	Yes	Sun	Dinner	2
175	32.90	3.11	Male	Yes	Sun	Dinner	2
179	34.63	3.55	Male	Yes	Sun	Dinner	2

180	34.65	3.68	Male	Yes	Sun	Dinner	4
182	45.35	3.50	Male	Yes	Sun	Dinner	3
184	40.55	3.00	Male	Yes	Sun	Dinner	2
187	30.46	2.00	Male	Yes	Sun	Dinner	5
197	43.11	5.00	Female	Yes	Thur	Lunch	4
207	38.73	3.00	Male	Yes	Sat	Dinner	4
210	30.06	2.00	Male	Yes	Sat	Dinner	3
212	48.33	9.00	Male	No	Sat	Dinner	4
219	30.14	3.09	Female	Yes	Sat	Dinner	4
237	32.83	1.17	Male	Yes	Sat	Dinner	2
238	35.83	4.67	Female	No	Sat	Dinner	3

groupby() functionality

In [42]:
```
# Applying the groupby() fuction on the column - time

G1 = tips_df.groupby('time')
```

In [44]:
```
# Printing the first entries of each group
G1.first()
```

Out[44]:

	total_bill	tip	sex	smoker	day	size
time						
Dinner	16.99	1.01	Female	No	Sun	2
Lunch	27.20	4.00	Male	No	Thur	4

In [46]:
```
# Let's find the values present in 'Dinner' group

G1.get_group('Dinner')
```

Out[46]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

176 rows × 7 columns

In [48]:
```
# The values present in 'Lunch' group
G1.get_group('Lunch')
```

Out[48]:

	total_bill	tip	sex	smoker	day	time	size
77	27.20	4.00	Male	No	Thur	Lunch	4
78	22.76	3.00	Male	No	Thur	Lunch	2
79	17.29	2.71	Male	No	Thur	Lunch	2
80	19.44	3.00	Male	Yes	Thur	Lunch	2
81	16.66	3.40	Male	No	Thur	Lunch	2
...	...	...	...	...	...	...	...
222	8.58	1.92	Male	Yes	Fri	Lunch	1
223	15.98	3.00	Female	No	Fri	Lunch	3
224	13.42	1.58	Male	Yes	Fri	Lunch	2
225	16.27	2.50	Female	Yes	Fri	Lunch	2
226	10.09	2.00	Female	Yes	Fri	Lunch	2

68 rows × 7 columns

Aggregate funtions - sum, min, avg etc

In [51]:
```
# Let's apply 'sum' and 'min' aggregate function.
# 'sum' will give the sum of the individual column and
# 'min' will give the minimum values present in each columns
```

In [53]:
```
tips_df[['total_bill','tip','size']].aggregate(['sum','min'])
```

Out[53]:

	total_bill	tip	size
sum	4827.77	731.58	627
min	3.07	1.00	1

In [55]:
```
# Note : Above operation will only work for the numerical columns
```

In [57]:
```
# applying the 'average' operation on the column 'total_bill'

tips_df['total_bill'].aggregate(['average'])
```

Out[57]:
```
average    19.785943
Name: total_bill, dtype: float64
```

In [59]:
```
# applying the 'max' operation on the column 'tip'

tips_df['tip'].aggregate(['max'])
```

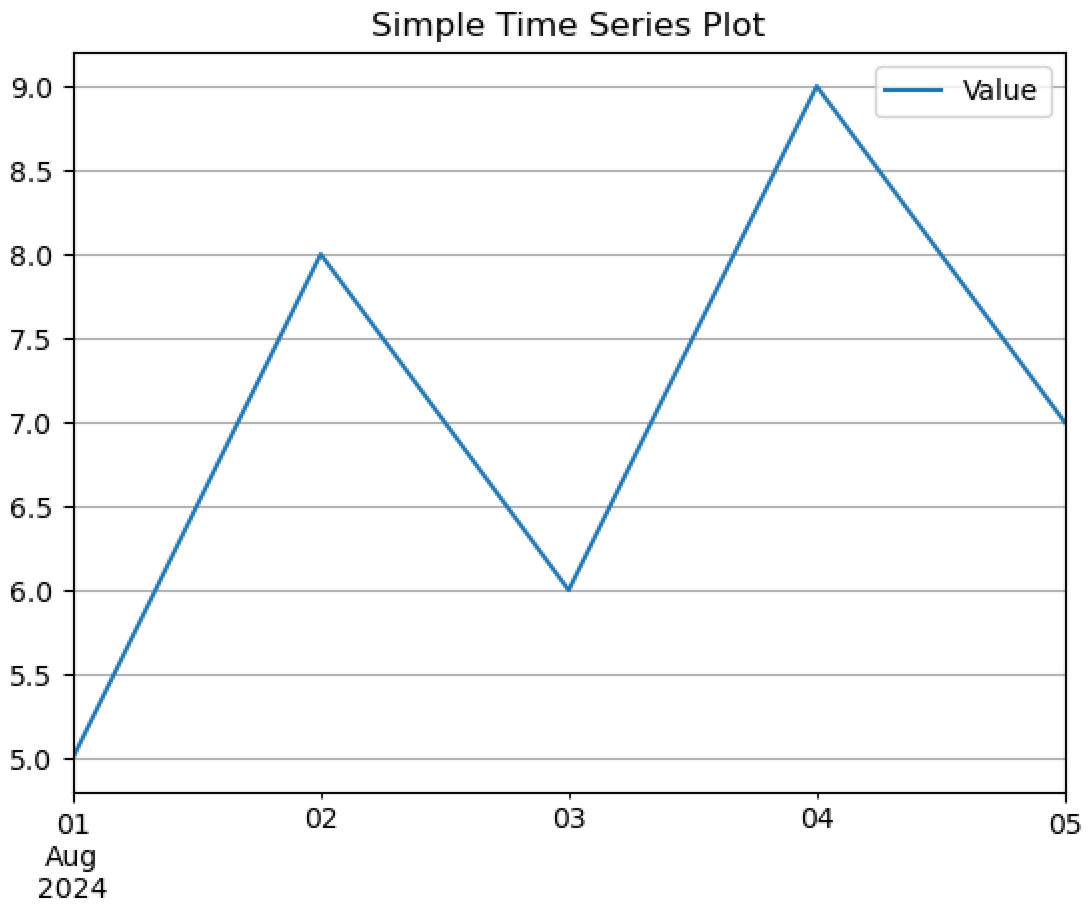
Out[59]:
```
max    10.0
Name: tip, dtype: float64
```

Time Series Operation

In [62]:
```
# Import matplotlib for plotting
import matplotlib.pyplot as plt

# Create a time series DataFrame with dates over five days
dates = pd.date_range(start='2024-08-01', periods=5, freq='D')
data = [5, 8, 6, 9, 7]
df = pd.DataFrame(data, columns=['Value'], index=dates)

# Plot the data
df.plot(title="Simple Time Series Plot", grid=True)
plt.show()
```



In [64]:
```
!jupyter nbconvert --to webpdf --allow-chromium-download Week4_Lab.ipynb
```

```
[NbConvertApp] Converting notebook Week4_Lab.ipynb to webpdf
[NbConvertApp] Building PDF
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 363171 bytes to Week4_Lab.pdf
```

In [ ]:
```

```