

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
df=pd.read_csv("/content/2_2015 - 2_2015.csv")
df
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom (C
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297
...
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453

158 rows × 12 columns



```
df=df.dropna()
data=df[["Country", "Happiness Score"]]
data
```

	Country	Happiness Score
0	Switzerland	7.587
1	Iceland	7.561
2	Denmark	7.527
3	Norway	7.522
4	Canada	7.427
...
153	Rwanda	3.465
154	Benin	3.340
155	Syria	3.006
156	Burundi	2.905
157	Togo	2.839



```
data.plot.line()
```

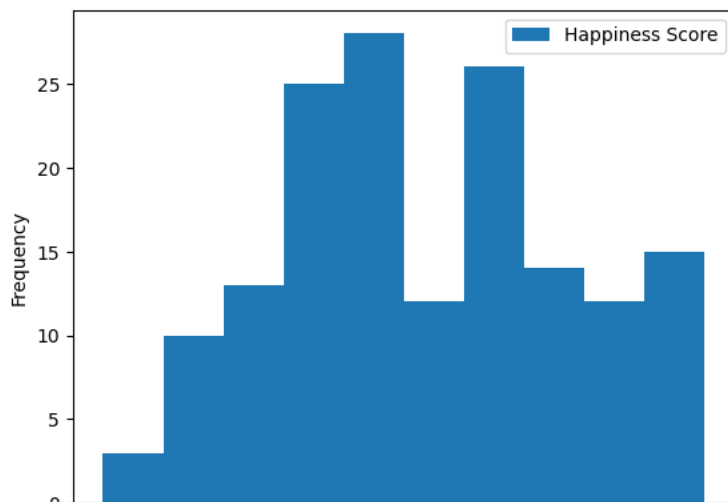
The graph displays a single data series labeled 'Happiness Score'. The y-axis represents the score, ranging from 3 to 7.5. The x-axis represents time in days, from 0 to 100. The score begins at approximately 7.5 on day 0 and shows a consistent downward trend. There are several small, temporary fluctuations, such as a slight increase around day 20 and a small dip around day 40. The most significant feature is a sharp decline starting around day 80, where the score drops from approximately 3.8 to 2.8 by day 100.

A boxplot showing the distribution of the number of children per family. The y-axis is labeled from 3 to 7. The box extends from approximately 4.5 to 6.2, with a green median line at 5.2. Whiskers extend from approximately 2.8 to 7.8.

A scatter plot showing a negative correlation between a variable on the x-axis and the Happiness Score on the y-axis. The y-axis is labeled 'Happiness Score' and ranges from 3 to 7. The data points are blue dots, showing a downward trend from a happiness score of approximately 7.5 at the left to approximately 2.8 at the right.

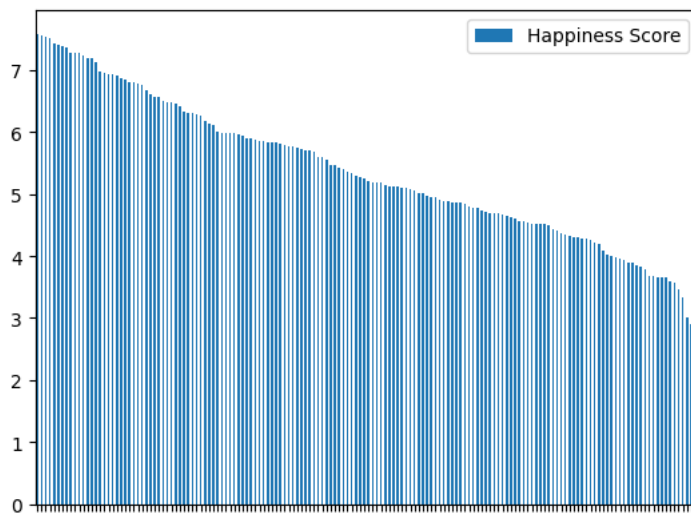
```
data.plot.hist()
```

<Axes: ylabel='Frequency'>



```
data.plot.bar()
```

<Axes: >



```
df2=pd.read_csv("/content/1_fiat500_VehicleSelection_Dataset - 1_fiat500_VehicleSelection_Dataset.csv")
df2
```

	ID	model	engine_power	age_in_days	km	previous_owners
0	1.0	lounge	51.0	882.0	25000.0	1.0 4
1	2.0	pop	51.0	1186.0	32500.0	1.0 4
2	3.0	sport	74.0	4658.0	142228.0	1.0 4
3	4.0	lounge	51.0	2739.0	160000.0	1.0 4
4	5.0	pop	73.0	3074.0	106880.0	1.0 4
...
1544	NaN	NaN	NaN	NaN	NaN	NaN
1545	NaN	NaN	NaN	NaN	NaN	NaN
1546	NaN	NaN	NaN	NaN	NaN	NaN
1547	NaN	NaN	NaN	NaN	NaN	NaN

```
data2=df2[["Unnamed: 9","Unnamed: 10"]]
data2=df2.drop(data2,axis=1)
data2=df2.drop(df2.index[1537:1549])
data2
```

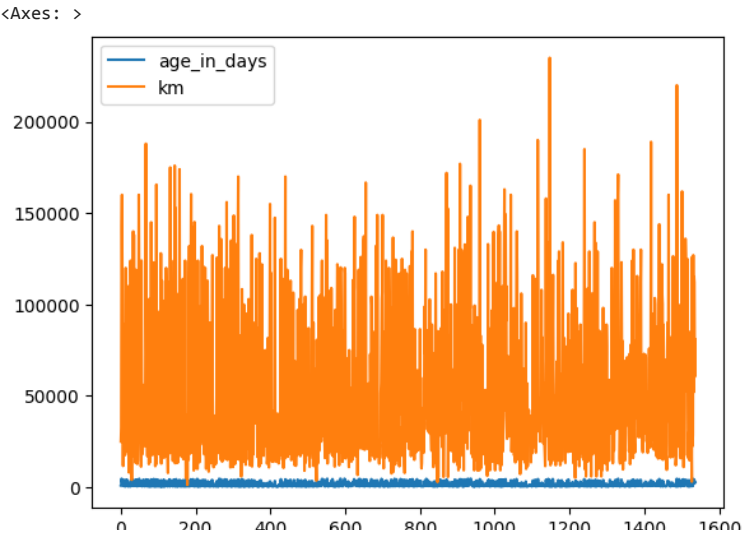
	ID	model	engine_power	age_in_days	km	previous_owners
0	1.0	lounge	51.0	882.0	25000.0	1.0
1	2.0	pop	51.0	1186.0	32500.0	1.0
2	3.0	sport	74.0	4658.0	142228.0	1.0
3	4.0	lounge	51.0	2739.0	160000.0	1.0
4	5.0	pop	73.0	3074.0	106880.0	1.0
...
1532	1533.0	pop	51.0	1917.0	52008.0	1.0
1533	1534.0	sport	51.0	3712.0	115280.0	1.0
1534	1535.0	lounge	74.0	3835.0	112000.0	1.0
1535	1536.0	pop	51.0	2223.0	60457.0	1.0

```
data3=data2[["age_in_days","km"]]  
data3
```

	age_in_days	km
0	882.0	25000.0
1	1186.0	32500.0
2	4658.0	142228.0
3	2739.0	160000.0
4	3074.0	106880.0
...
1532	1917.0	52008.0
1533	3712.0	115280.0
1534	3835.0	112000.0
1535	2223.0	60457.0
1536	2557.0	80750.0



```
data3.plot.line()
```



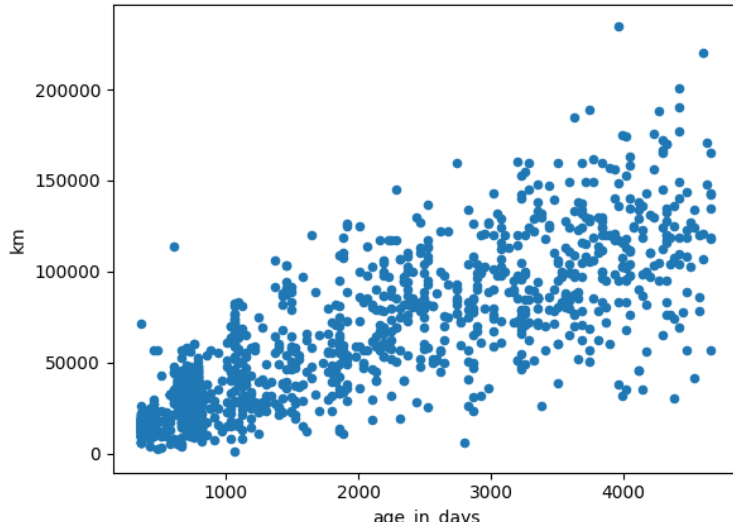
```
data3.plot.box()
```

<Axes: >



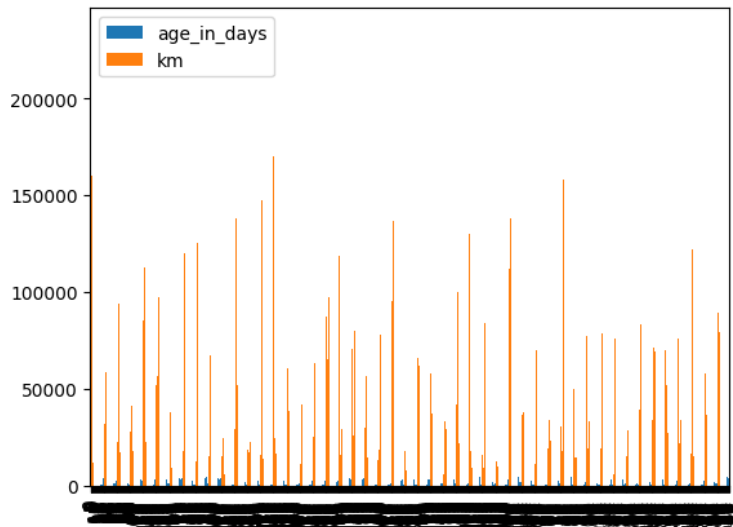
```
data3.plot.scatter("age_in_days","km")
```

<Axes: xlabel='age_in_days', ylabel='km'>



```
data3.plot.bar()
```

<Axes: >



```
data3.plot.hist()
```

<Axes: ylabel='Frequency'>



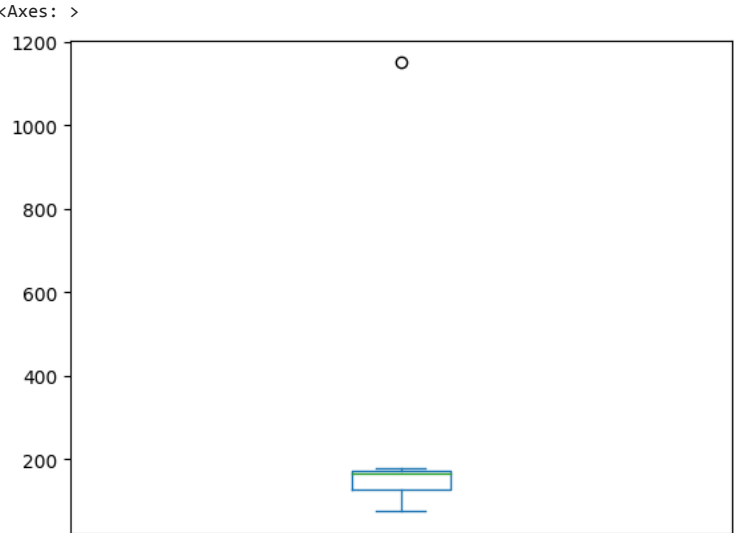
```
df3=pd.read_csv("3_Fitness-1 - 3_Fitness-1.csv")
df3
```

	Row Labels	Sum of Jan	Sum of Feb	Sum of Mar	Sum of Total Sales
0	A	5.62%	7.73%	6.16%	75
1	B	4.21%	17.27%	19.21%	160
2	C	9.83%	11.60%	5.17%	101
3	D	2.81%	21.91%	7.88%	127
4	E	25.28%	10.57%	11.82%	179
5	F	8.15%	16.24%	18.47%	167
6	G	18.54%	8.76%	17.49%	171
7	H	25.56%	5.93%	13.79%	170

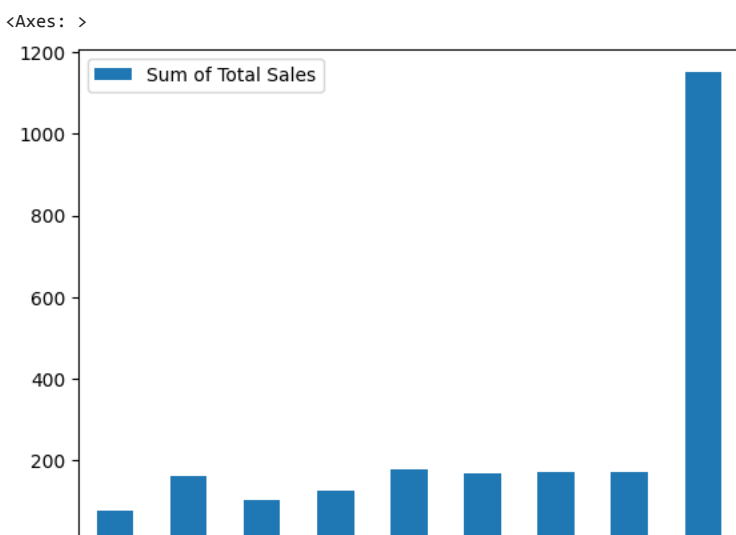
```
da=df3[["Row Labels","Sum of Total Sales"]]
da
```

	Row Labels	Sum of Total Sales
0	A	75
1	B	160
2	C	101
3	D	127
4	E	179
5	F	167
6	G	171
7	H	170

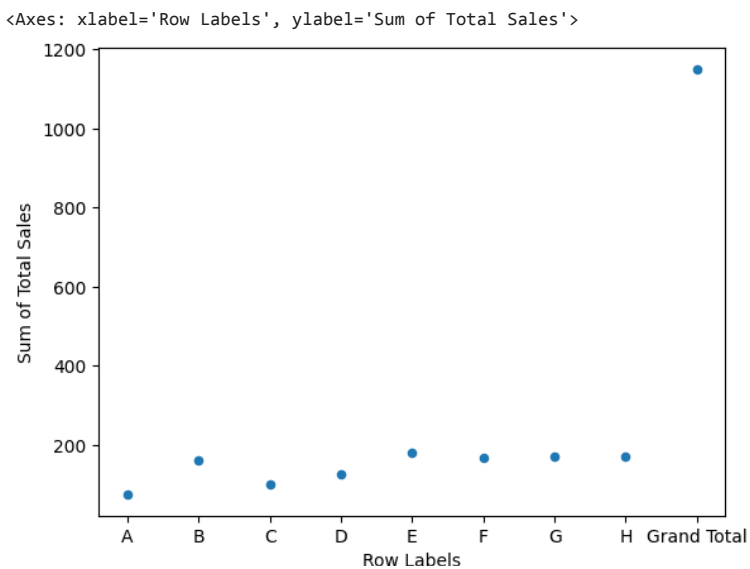
```
da.plot.box()
```



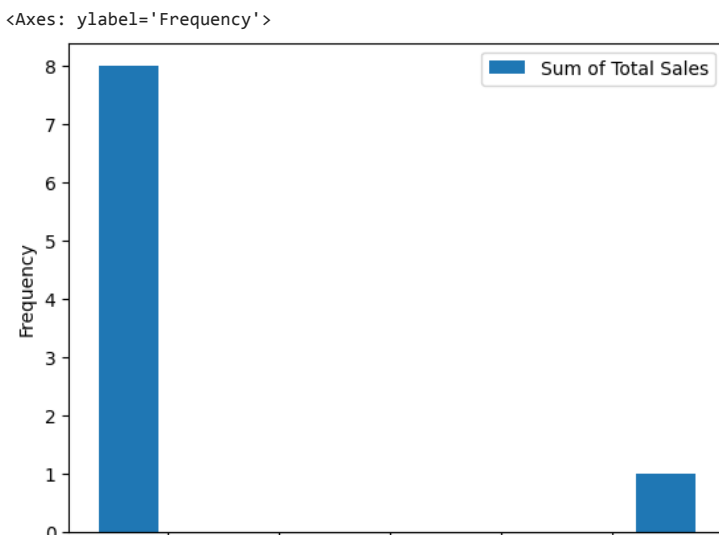
```
da.plot.bar()
```



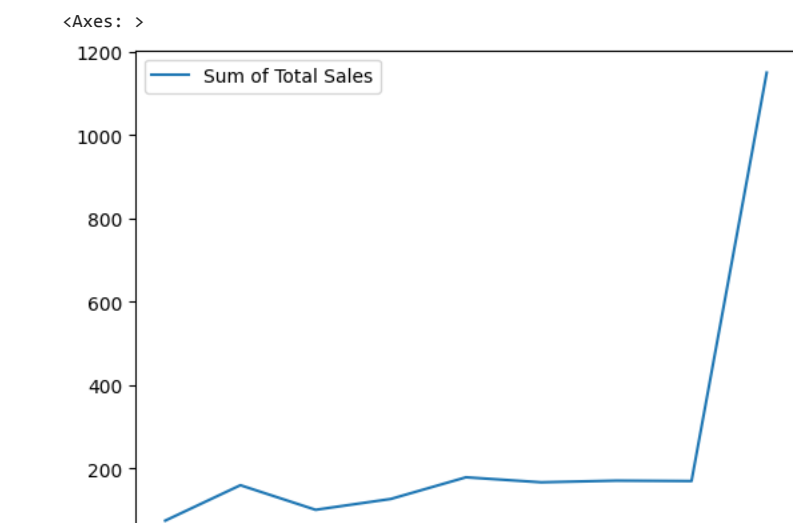
```
da.plot.scatter("Row Labels","Sum of Total Sales")
```



```
da.plot.hist()
```



```
da.plot.line()
```

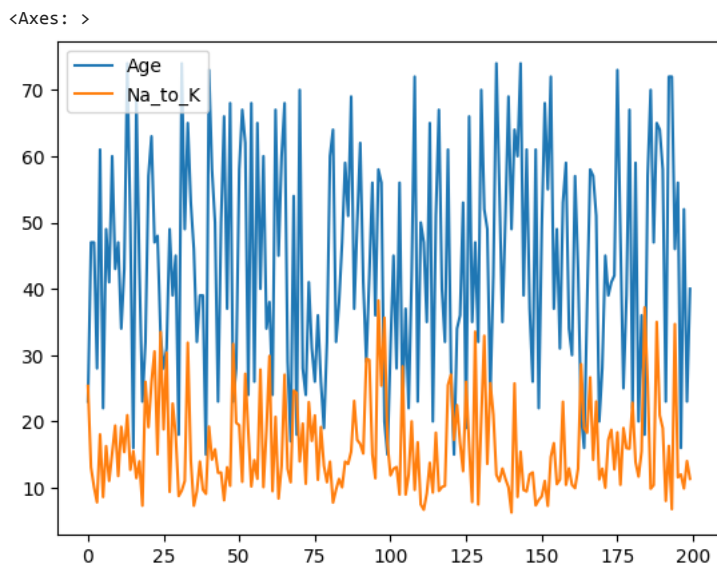


```
df4=pd.read_csv("/content/4_drug200 - 4_drug200.csv")
df4
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY
...
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

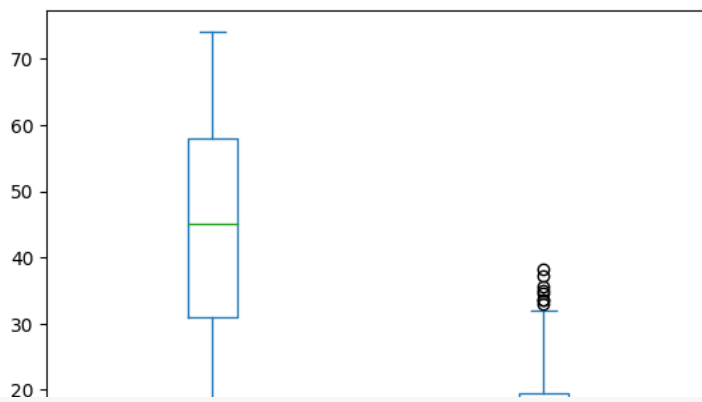
200 rows × 6 columns

```
data4=df4[["Age","Na_to_K"]]
data4.plot.line()
```



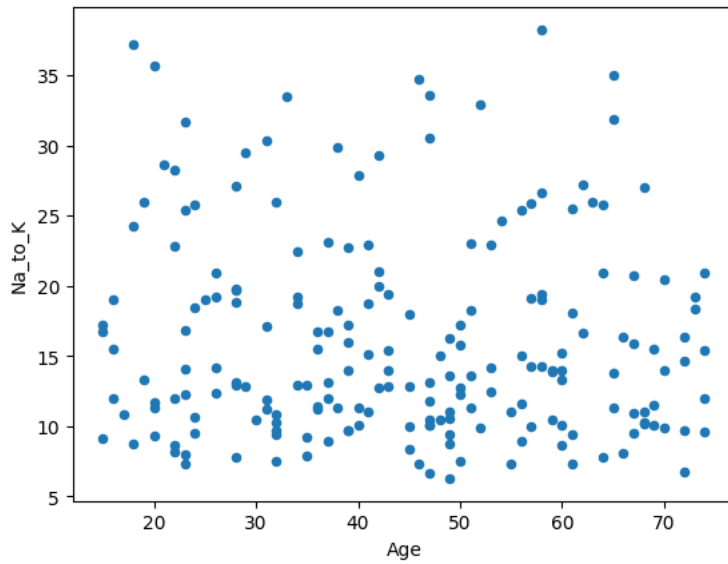
```
data4.plot.box()
```


<Axes: >



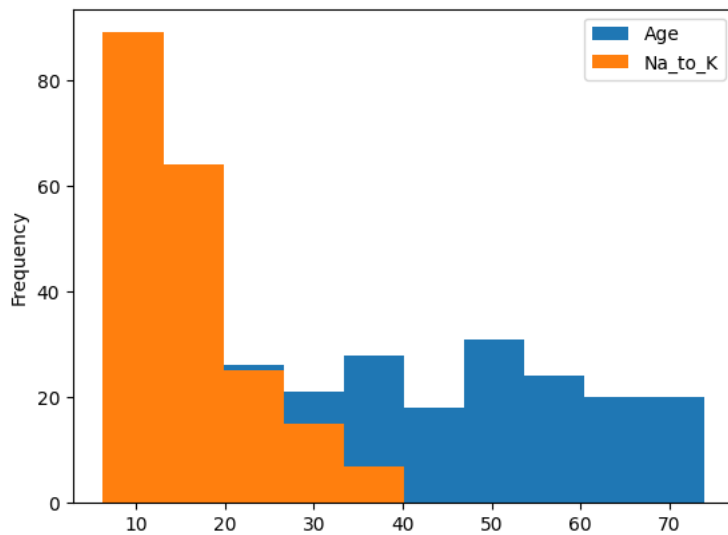
```
data4.plot.scatter("Age", "Na_to_K")
```

<Axes: xlabel='Age', ylabel='Na_to_K'>



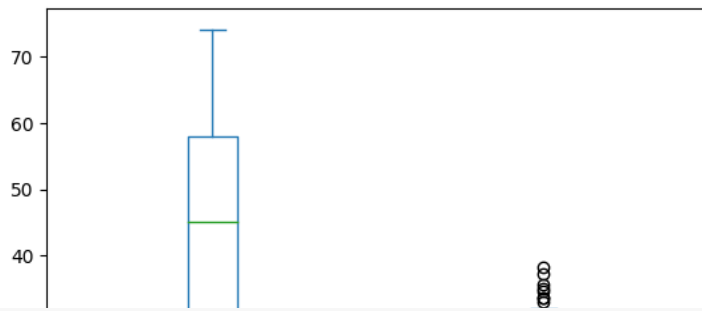
```
data4.plot.hist()
```

<Axes: ylabel='Frequency'>



```
data4.plot.box()
```

<Axes: >



```
df5=pd.read_csv("/content/5_Instagram data - 5_Instagram data.csv")
df5
```

Impressions

From Home

From Hashtags

From Explore

From Other

Saves

Comments

Shares

Likes

Profile Visits

Follows

Caption

Hash1

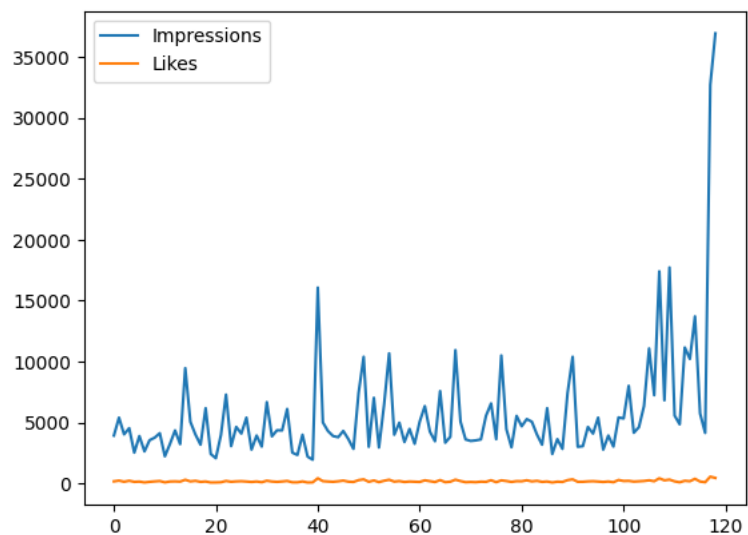
```
data5=df5[["Impressions","Likes"]]
data5
```

	Impressions	Likes
0	3920	162
1	5394	224
2	4021	131
3	4528	213
4	2518	123
...
114	13700	373
115	5731	148
116	4139	92
117	32695	549
118	36919	443

119 rows × 2 columns

```
data5.plot.line()
```

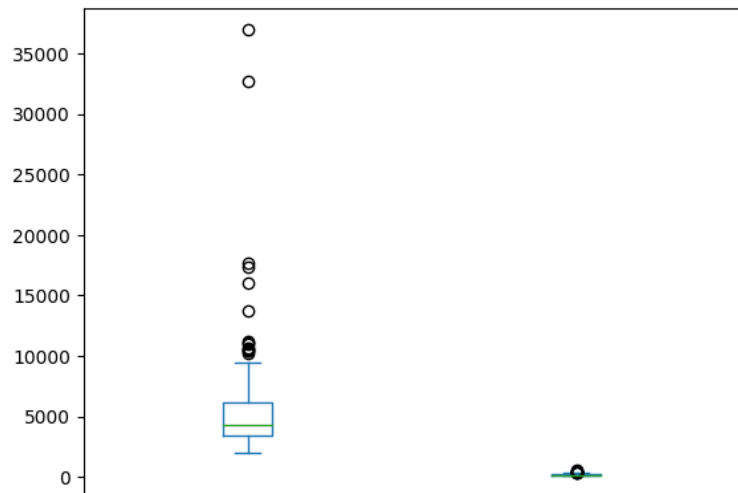
<Axes: >



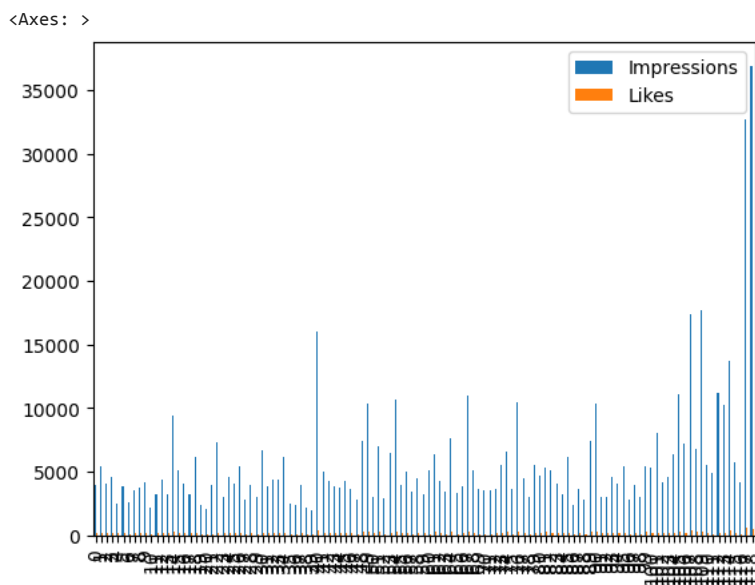
data

```
data5.plot.box()
```

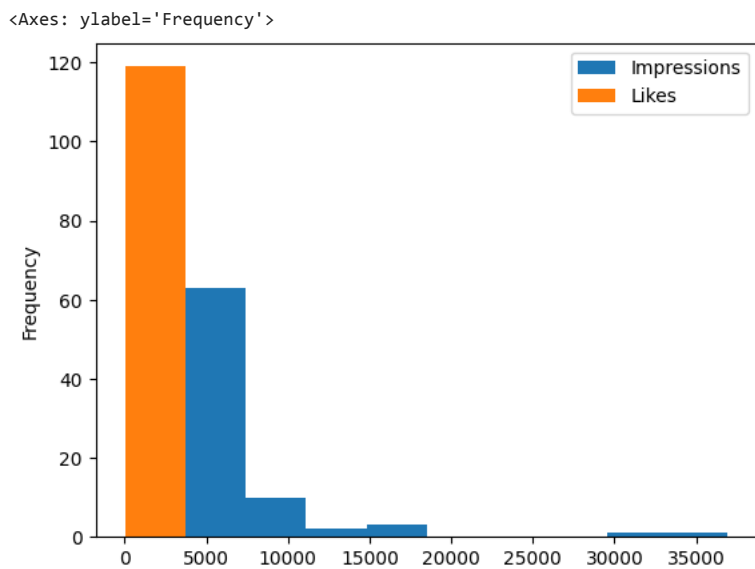
<Axes: >



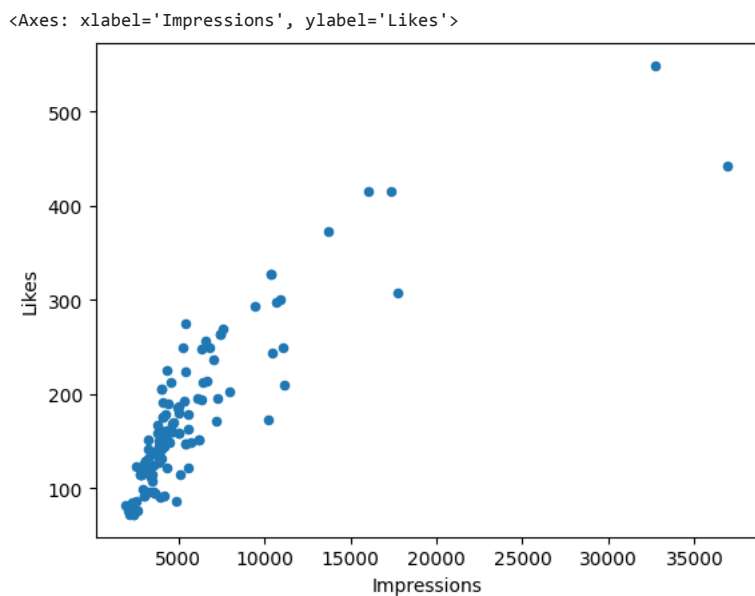
```
data5.plot.bar()
```



```
data5.plot.hist()
```



```
data5.plot.scatter("Impressions","Likes")
```



```
df6=pd.read_csv("/content/6_Salesworkload1 - 6_Salesworkload1.csv")  
df6
```

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLease	Sales units	Turnover	Customer	Area (m2)	Opening hours
0	10.2016	1.0	United Kingdom	88253.0	London (l)	1.0	Dry	3184.764	0.0	398560.0	1226244.0	NaN	953.04	Type A
1	10.2016	1.0	United Kingdom	88253.0	London (l)	2.0	Frozen	1582.941	0.0	82725.0	387810.0	NaN	720.48	Type A
2	10.2016	1.0	United Kingdom	88253.0	London (l)	3.0	other	47.205	0.0	438400.0	654657.0	NaN	966.72	Type A
3	10.2016	1.0	United Kingdom	88253.0	London (l)	4.0	Fish	1623.852	0.0	309425.0	499434.0	NaN	1053.36	Type A
4	10.2016	1.0	United Kingdom	88253.0	London (l)	5.0	Fruits & Vegetables	1759.173	0.0	165515.0	329397.0	NaN	1053.36	Type A
...
7653	6.2017	9.0	Sweden	29650.0	Gothenburg	12.0	Checkout	6322.323	0.0	3886530.0	14538825.0	NaN	#NV	Type A
7654	6.2017	9.0	Sweden	29650.0	Gothenburg	16.0	Customer Services	4270.479	0.0	245.0	0.0	NaN	#NV	Type A
7655	6.2017	9.0	Sweden	29650.0	Gothenburg	11.0	Delivery	0	0.0	0.0	0.0	NaN	#NV	Type A
7656	6.2017	9.0	Sweden	29650.0	Gothenburg	17.0	others	2224.929	0.0	245.0	0.0	NaN	#NV	Type A
7657	6.2017	9.0	Sweden	29650.0	Gothenburg	18.0	all	39652.2	0.0	3886530.0	15056214.0	NaN	#NV	Type A

7658 rows × 14 columns



```
data6=df6[["Country","Sales units"]]
data6=data6.dropna()
data6["Sales units"]=data6["Sales units"].astype(int)
data6
```

	Country	Sales units
0	United Kingdom	398560
1	United Kingdom	82725
2	United Kingdom	438400
3	United Kingdom	309425
4	United Kingdom	165515
...
7653	Sweden	3886530
7654	Sweden	245
7655	Sweden	0
7656	Sweden	245
7657	Sweden	3886530

7650 rows × 2 columns

```
data6.plot.line()
```

<Axes: >

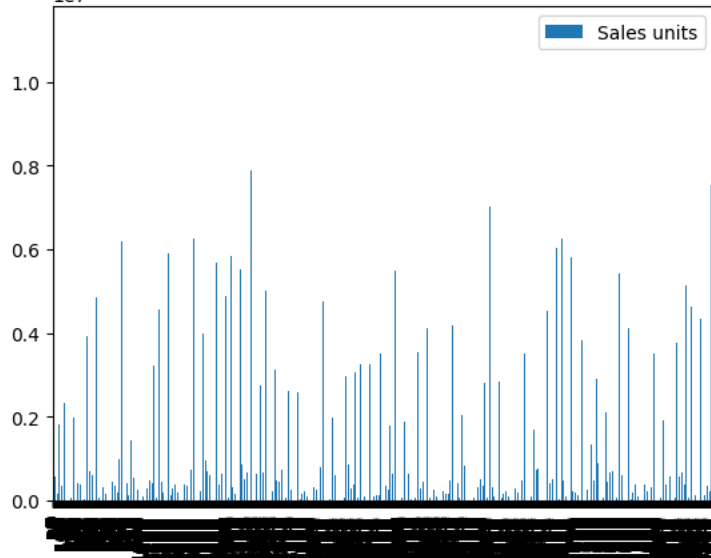
1e7



data6.plot.bar()

<Axes: >

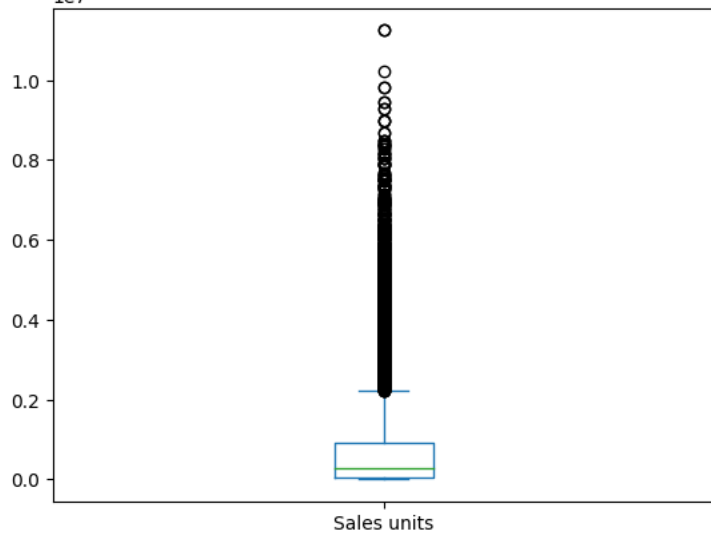
1e7



data6.plot.box()

<Axes: >

1e7



data6.plot.hist()

<Axes: ylabel='Frequency'>

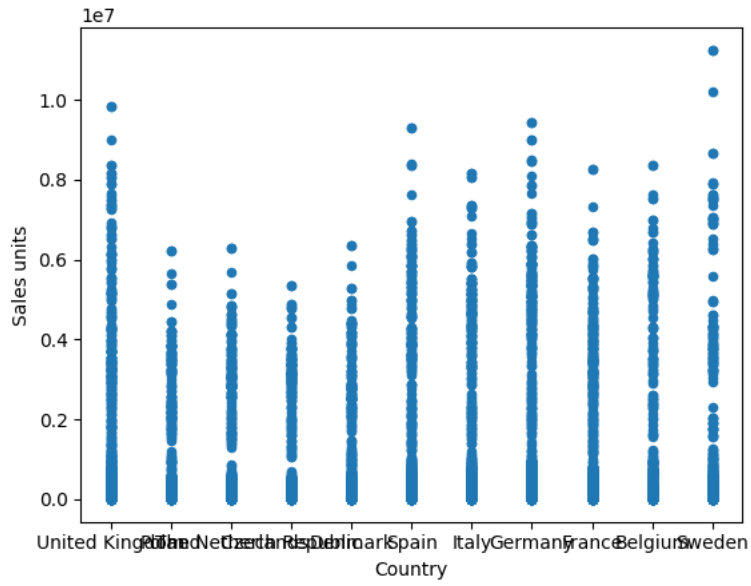
6000



Sales units

```
data6.plot.scatter("Country", "Sales units")
```

<Axes: xlabel='Country', ylabel='Sales units'>



Colab paid products - [Cancel contracts here](#)