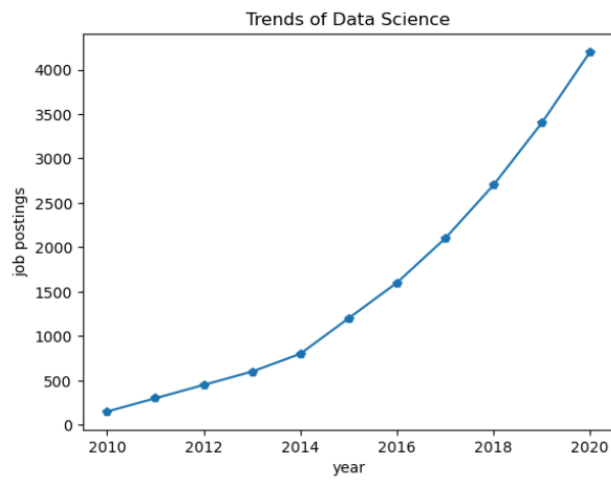
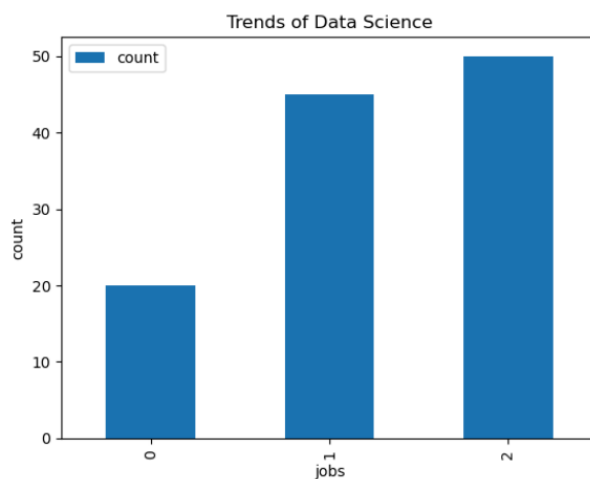


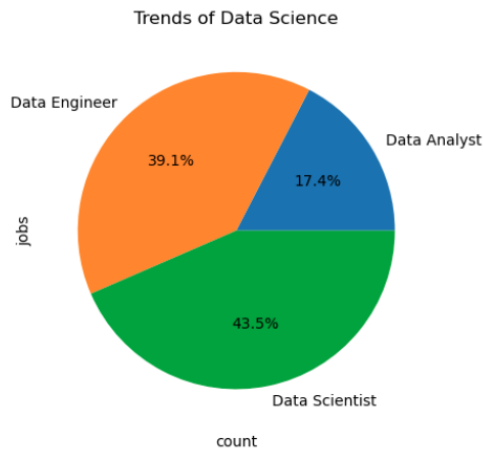
```
In [11]: import pandas as pd
import matplotlib.pyplot as plt
data={"Year": list(range(2010,2021)), "Job Postings": [150,300,450,600,800,1200,1600,2100,2700,3400,4200]}
#print(data)
df=pd.DataFrame(data)
plt.plot(df['Year'],df['Job Postings'],marker='p')
plt.title('Trends of Data Science')
plt.xlabel('year')
plt.ylabel('job postings')
plt.show()
```



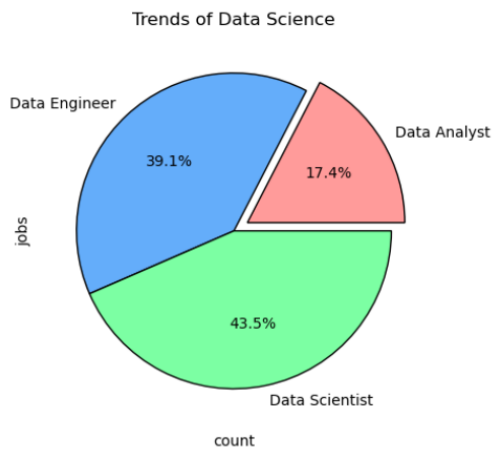
```
In [17]: import pandas as pd
import matplotlib.pyplot as plt
data={"count": [20,45,50], "jobs": ['Data Analyst', 'Data Engineer', 'Data Scientist']}
df=pd.DataFrame(data)
df.plot.bar()
plt.title('Trends of Data Science')
plt.ylabel('count')
plt.xlabel('jobs')
plt.show()
```



```
In [30]: import pandas as pd
import matplotlib.pyplot as plt
count=[20,45,50]
jobs=['Data Analyst','Data Engineer','Data Scientist']
plt.pie(count,labels=jobs,autopct='%1.1f%%')
plt.title('Trends of Data Science')
plt.xlabel('count')
plt.ylabel('jobs')
plt.show()
```



```
In [32]: import pandas as pd
import matplotlib.pyplot as plt
count=[20,45,50]
jobs=['Data Analyst','Data Engineer','Data Scientist']
plt.pie(count,labels=jobs,autopct='%1.1f%%', colors=colors, explode=explode, wedgeprops=dict(edgecolor='black'))
plt.title('Trends of Data Science')
plt.xlabel('count')
plt.ylabel('jobs')
colors = ['#ff9999','#66b3ff','#99ff99']
explode = (0.1, 0, 0)
plt.show()
```



```
In [46]: import pandas as pd
unstructured_data={"Ram 21 Chennai","Aadhi 22 Coimbatore","Leela 23 Trichy"}
lst=list(unstructured_data)
#df=pd.DataFrame(lst,columns=['unstructured_data'])
print(lst)

['Ram 21 Chennai', 'Leela 23 Trichy', 'Aadhi 22 Coimbatore']
```

```
In [45]: import pandas as pd
semistructured_data=["Ram 21 Chennai","Aadhi 22 Coimbatore","Leela 23 Trichy"]
df=pd.DataFrame(semistructured_data,columns=['Semistructured_Data'])
print(df)

   Semistructured_Data
0      Ram 21 Chennai
1  Aadhi 22 Coimbatore
2    Leela 23 Trichy
```

```
In [50]: import pandas as pd
structured_data=pd.DataFrame({
    'ID':[1,2,3],
    'Name':['Lakshya','Lakshetha','Lavanya'],
    'Age':[18,19,20]})
print("Structured Data\n",structured_data)

Structured Data
   ID  Name  Age
0   1  Lakshya  18
1   2  Lakshetha  19
2   3   Lavanya  20
```

```
In [52]: from cryptography.fernet import Fernet
key=Fernet.generate_key()
f=Fernet(key)
token=f.encrypt(b"Save Wayanad")
token
b'...'
f.decrypt(token)
b'Save Wayanad'
key=Fernet.generate_key()
cipher_suite=Fernet(key)
plain_text=b"Save Wayanad"
cipher_text=cipher_suite.encrypt(plain_text)
decrypted_text=cipher_suite.decrypt(cipher_text)
print("Original Data: ",plain_text)
print("Encrypted Data: ",cipher_text)
print("Decrypted Data: ",decrypted_text)
```

Original Data: b'Save Wayanad'
 Encrypted Data: b'gAAAAABmvr88_5TTcXTstZ8PdRyis4iUUaIZ1eIOiFZzUcT1qbZcAsVJJpiCU6U1VaCAHB0KyAjpg1Mz3uuz0eUJhWyHXhedfwQ=='
 Decrypted Data: b'Save Wayanad'

```
In [5]: import pandas as pd
db=pd.read_csv("C:/Users/DELL/Downloads/archive.zip")
print(db.head())
```

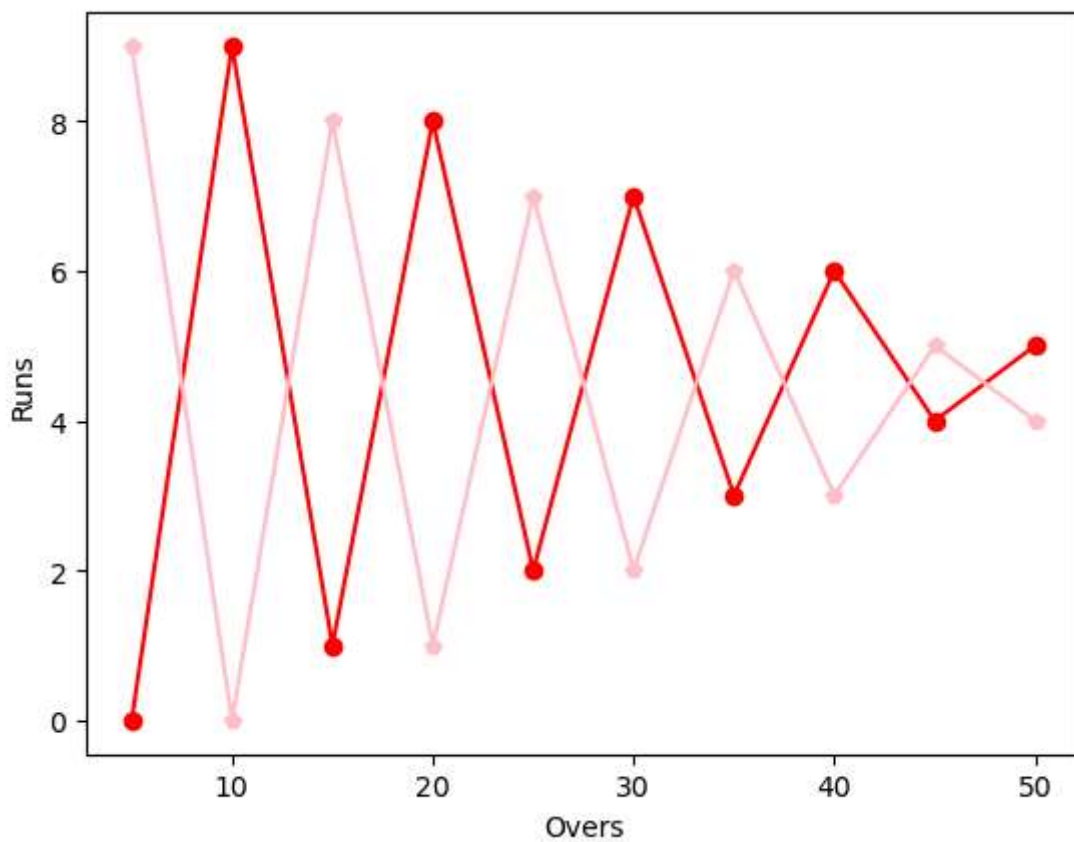
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1

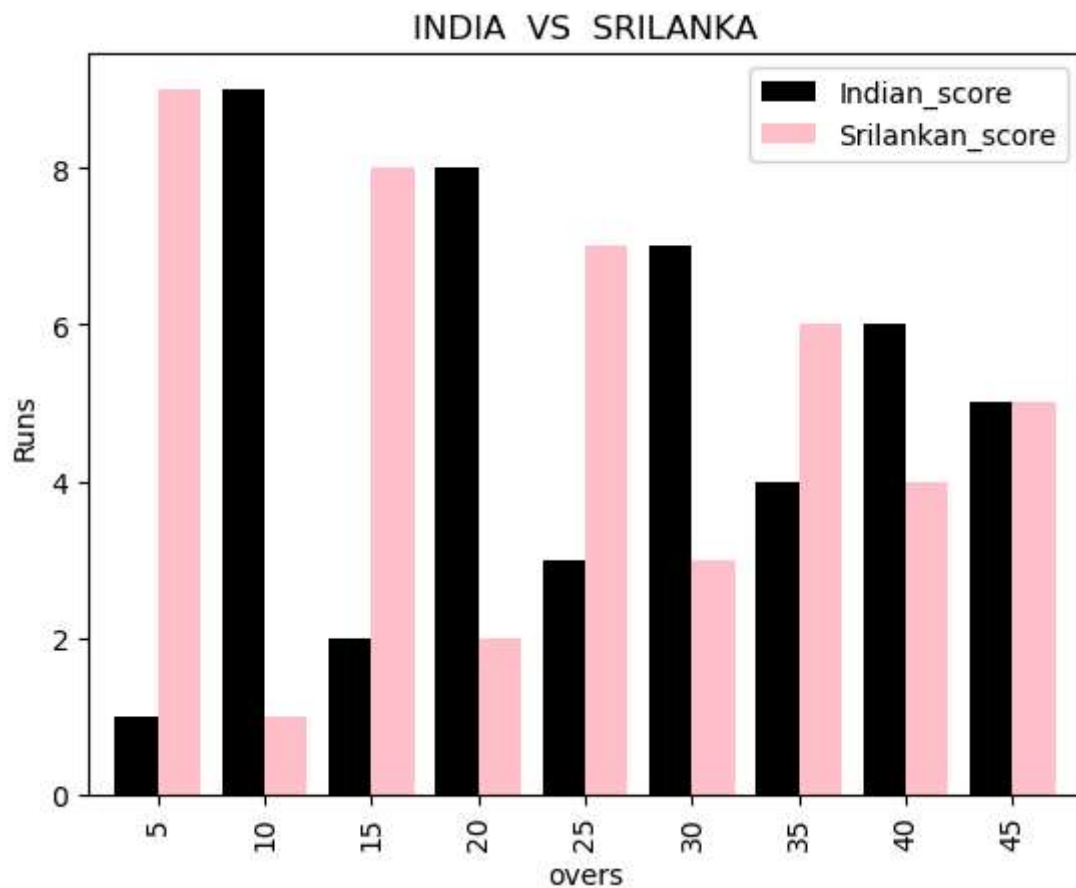
In []:

```
In [16]: import matplotlib.pyplot as cricket
overs=list(range(5,55,5))
Indian_score=[int(input())for _ in range(10)]
Srilankan_score=[int(input())for _ in range(10)]
cricket.plot(overs,Indian_score,marker='o',color='red')
cricket.plot(overs,Srilankan_score,marker='p',color='pink')
cricket.xlabel('Overs')
cricket.ylabel('Runs')
cricket.show()
```

0
9
1
8
2
7
3
6
4
5
9
0
8
1
7
2
6
3
5
4



```
In [38]: import matplotlib.pyplot as cricket
import pandas as pd
data1={'overs':list(range(5,50,5)), 'Indian_score':[1,9,2,8,3,7,4,6,5], 'Srilanka_score':[0,9,1,8,2,7,3,6,4,5]}
#data2={'overs':list(range(5,55,5)), 'Srilankan_score':[0,9,1,8,2,7,3,6,4,5]}
df1=pd.DataFrame(data1)
#df2=pd.DataFrame(data2)
#df=pd.merge(df1,df2)
#df1.plot.bar()
#cricket.plot(overs,Indian_score)
#cricket.plot(overs,Srilankan_score)
df1.plot(kind='bar',x='overs',y=['Indian_score','Srilankan_score'],color=['black','pink'])
cricket.xlabel('overs')
cricket.ylabel('Runs')
cricket.title('INDIA VS SRILANKA')
cricket.show()
```

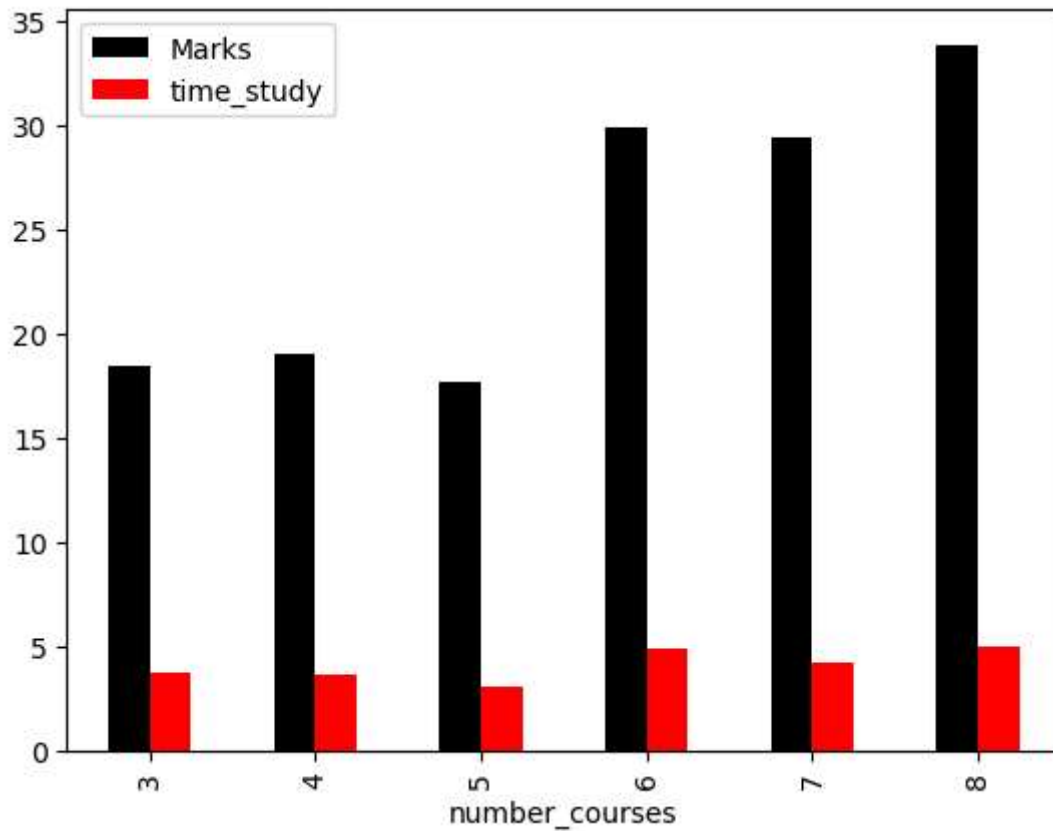


```
In [69]: import matplotlib.pyplot as plt
import pandas as pd
file="C:/Users/DELL/Downloads/Student_Marks.csv"
df=pd.read_csv(file)
grouped_df = df.groupby('number_courses').mean()
grouped_df[['Marks', 'time_study']].plot(kind='bar', color=['black', 'red'])

print(df)
plt.show()
```

	number_courses	time_study	Marks
0	3	4.508	19.202
1	4	0.096	7.734
2	4	3.133	13.811
3	6	7.909	53.018
4	8	7.811	55.299
..
95	6	3.561	19.128
96	3	0.301	5.609
97	4	7.163	41.444
98	7	0.309	12.027
99	3	6.335	32.357

[100 rows x 3 columns]

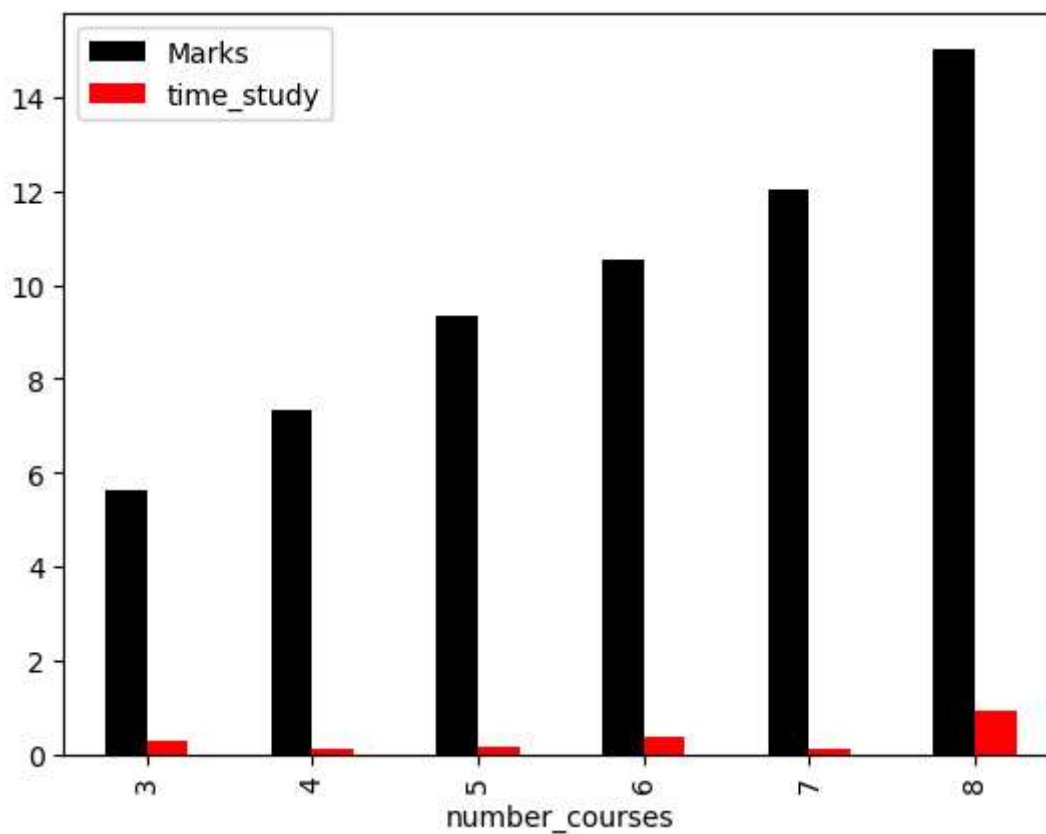


```
In [72]: import matplotlib.pyplot as plt
import pandas as pd
file="C:/Users/DELL/Downloads/Student_Marks.csv"
df=pd.read_csv(file)
grouped_df = df.groupby('number_courses').min()
grouped_df[['Marks', 'time_study']].plot(kind='bar', color=['black', 'red'])

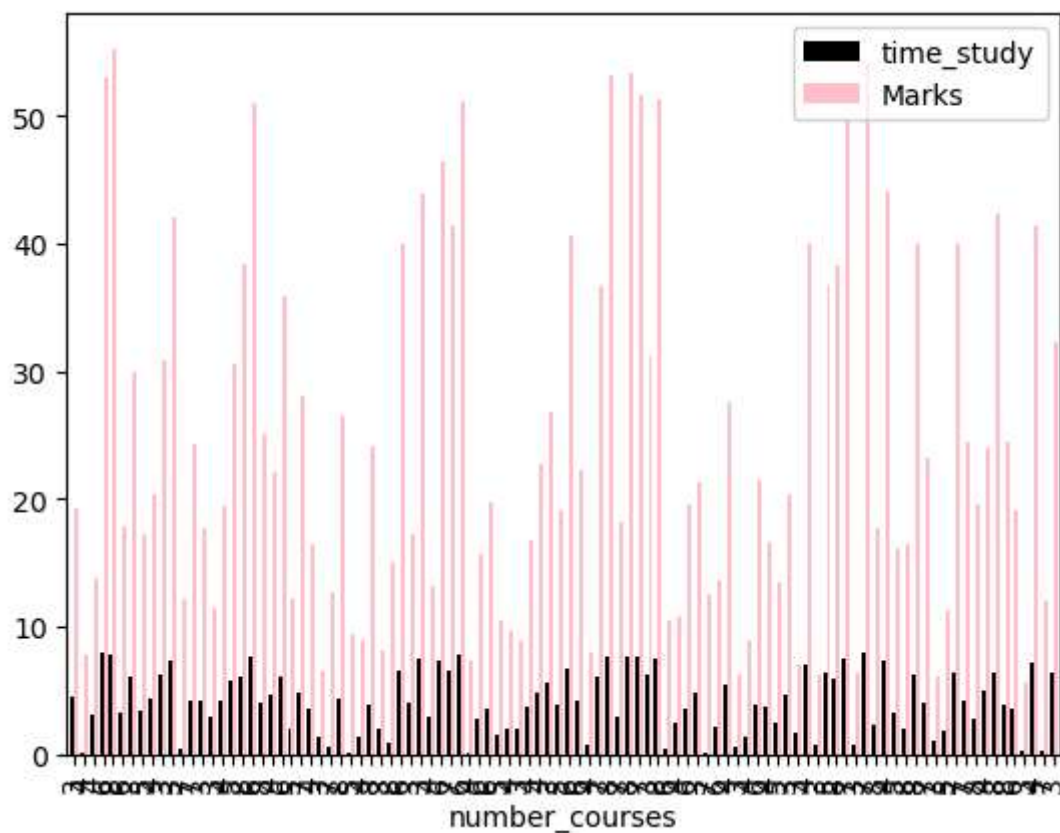
print(df)
plt.show()
```

	number_courses	time_study	Marks
0	3	4.508	19.202
1	4	0.096	7.734
2	4	3.133	13.811
3	6	7.909	53.018
4	8	7.811	55.299
..
95	6	3.561	19.128
96	3	0.301	5.609
97	4	7.163	41.444
98	7	0.309	12.027
99	3	6.335	32.357

[100 rows x 3 columns]




```
In [78]: import matplotlib.pyplot as plt
import pandas as pd
file="C:/Users/DELL/Downloads/Student_Marks.csv"
df=pd.read_csv(file)
df.plot(kind='bar',x='number_courses',y=['time_study','Marks'],color=['black',
plt.show()
```



```
In [ ]:
```

```
In [1]: import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
file="C:/Users/DELL/Documents/AI_job_edit.csv"
df=pd.read_csv(file)
print(df.isnull().sum())
df.fillna(df.mean(numeric_only=True),inplace=True)
df.fillna('NA',inplace =True)
print(df)
```

```

Job_Title      2
Industry       0
Company_Size   0
Location       0
AI_Adoption_Level  0
Automation_Risk  0
Required_Skills  0
Salary_USD     2
Remote_Friendly  0
Job_Growth_Projection  0
dtype: int64

```

	Job_Title	Industry	Company_Size	Location
0	NA	Entertainment	Small	Dubai
1	Marketing Specialist	Technology	Large	Singapore
2	AI Researcher	Technology	Large	Singapore
3	Sales Manager	Retail	Small	Berlin
4	Cybersecurity Analyst	Entertainment	Small	Tokyo
..
495	Data Scientist	Telecommunications	Medium	Berlin
496	Cybersecurity Analyst	Telecommunications	Small	London
497	Cybersecurity Analyst	Energy	Large	Dubai
498	Operations Manager	Healthcare	Large	Paris
499	HR Manager	Entertainment	Medium	Berlin

	AI_Adoption_Level	Automation_Risk	Required_Skills	Salary_USD
0	Medium	High	UX/UI Design	91176.728431
1	Medium	High	Marketing	91176.728431
2	Medium	High	UX/UI Design	107170.263100
3	Low	High	Project Management	93027.953760
4	Low	Low	JavaScript	87752.922170
..
495	Low	Medium	Machine Learning	105821.394000
496	Low	High	UX/UI Design	119794.992100
497	High	Low	UX/UI Design	79644.933100
498	High	Low	Python	77642.150630
499	Medium	High	Project Management	68764.378920

	Remote_Friendly	Job_Growth_Projection
0	Yes	Growth
1	No	Decline
2	Yes	Growth
3	No	Growth
4	Yes	Decline
..
495	Yes	Stable
496	No	Decline
497	Yes	Stable
498	Yes	Stable
499	Yes	Decline

[500 rows x 10 columns]

```
In [14]: import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from scipy import stats
import seaborn as sns
file="C:/Users/DELL/Documents/AI_job_edit.csv"
df=pd.read_csv(file)
print(df.isnull().sum())
df.fillna(df.mean(numeric_only=True),inplace=True)
df.dropna(inplace =True)
print(df)
sns.pairplot(df.select_dtypes(include=[np.number]))
plt.show()
sns.boxplot(y=df[ 'Salary_USD' ])
plt.show()
df.dropna(inplace=True)
z_scores = np.abs(stats.zscore(df.select_dtypes(include=[np.number])))
threshold = 3
outliers = (z_scores > threshold)
outlier_rows = df.index[np.any(outliers, axis=1)]
print(outlier_rows)
print("Outlier rows:")
print(df.loc[outlier_rows])
```

```

Job_Title      2
Industry       0
Company_Size   0
Location       0
AI_Adoption_Level  0
Automation_Risk  0
Required_Skills  0
Salary_USD     2
Remote_Friendly  0
Job_Growth_Projection  0
dtype: int64

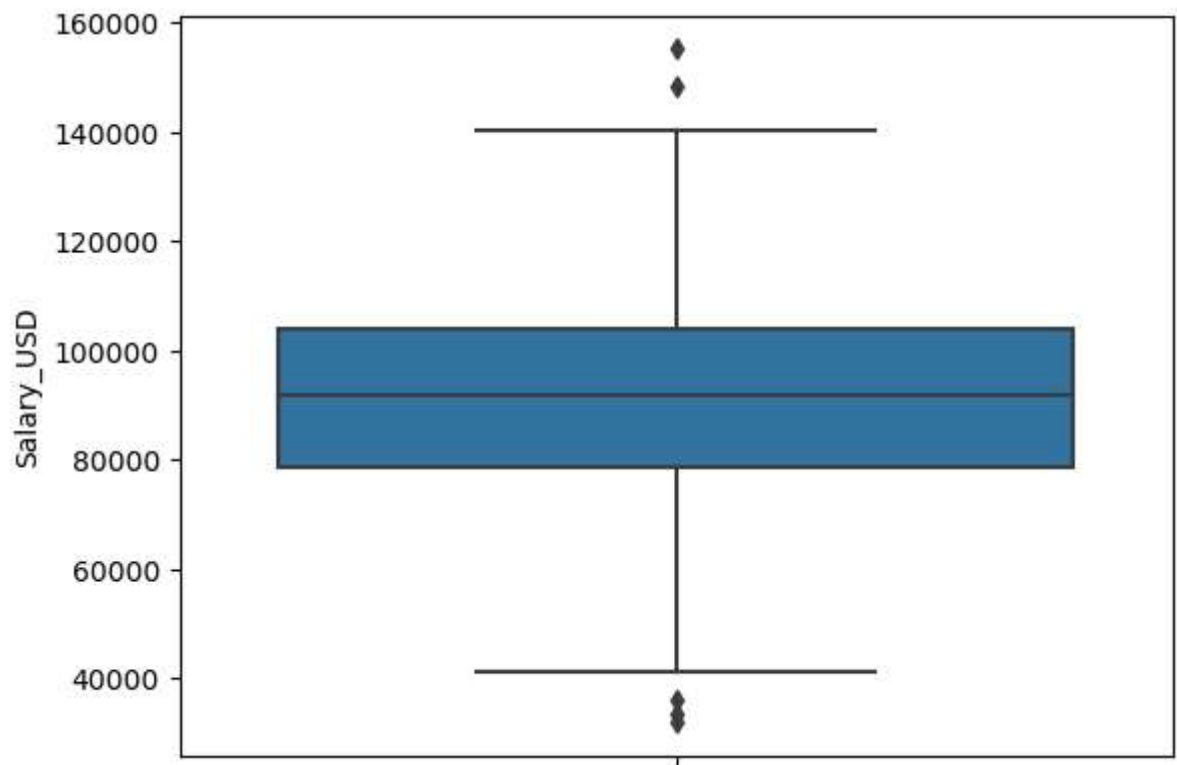
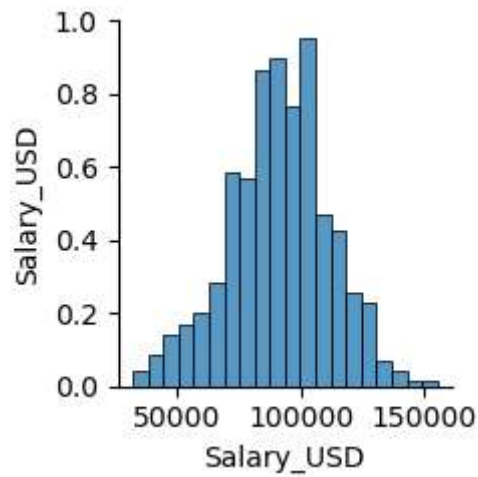
```

	Job_Title	Industry	Company_Size	Location	\
1	Marketing Specialist	Technology	Large	Singapore	
2	AI Researcher	Technology	Large	Singapore	
3	Sales Manager	Retail	Small	Berlin	
4	Cybersecurity Analyst	Entertainment	Small	Tokyo	
5	UX Designer	Education	Large	San Francisco	
..	
495	Data Scientist	Telecommunications	Medium	Berlin	
496	Cybersecurity Analyst	Telecommunications	Small	London	
497	Cybersecurity Analyst	Energy	Large	Dubai	
498	Operations Manager	Healthcare	Large	Paris	
499	HR Manager	Entertainment	Medium	Berlin	

	AI_Adoption_Level	Automation_Risk	Required_Skills	Salary_USD	\
1	Medium	High	Marketing	91176.728431	
2	Medium	High	UX/UI Design	107170.263100	
3	Low	High	Project Management	93027.953760	
4	Low	Low	JavaScript	87752.922170	
5	Medium	Medium	Cybersecurity	102825.007900	
..	
495	Low	Medium	Machine Learning	105821.394000	
496	Low	High	UX/UI Design	119794.992100	
497	High	Low	UX/UI Design	79644.933100	
498	High	Low	Python	77642.150630	
499	Medium	High	Project Management	68764.378920	

	Remote_Friendly	Job_Growth_Projection
1	No	Decline
2	Yes	Growth
3	No	Growth
4	Yes	Decline
5	No	Growth
..
495	Yes	Stable
496	No	Decline
497	Yes	Stable
498	Yes	Stable
499	Yes	Decline

[498 rows x 10 columns]



```
Int64Index([420], dtype='int64')
```

Outlier rows:

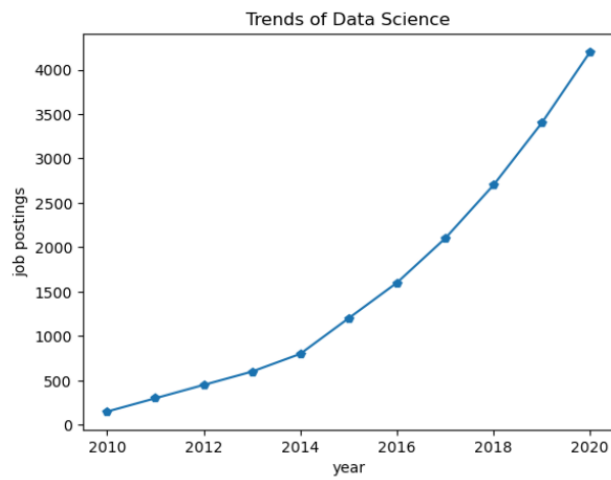
	Job_Title	Industry	Company_Size	Location \
420	Marketing Specialist	Finance	Medium	San Francisco

	AI_Adoption_Level	Automation_Risk	Required_Skills	Salary_USD \
420	High	High	Sales	155209.8216

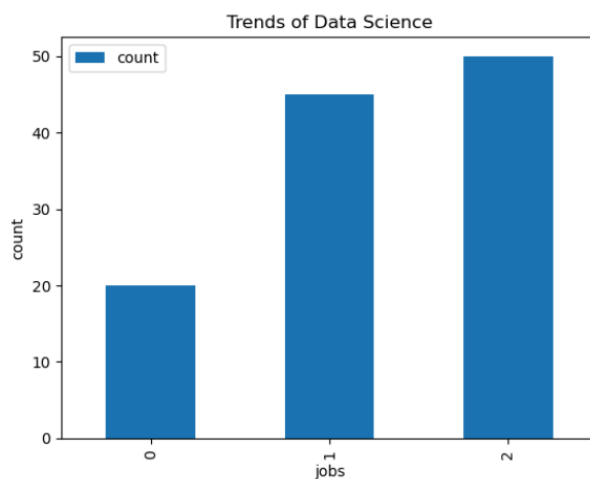
	Remote_Friendly	Job_Growth_Projection
420	Yes	Decline

```
In [ ]: import numpy as np
```

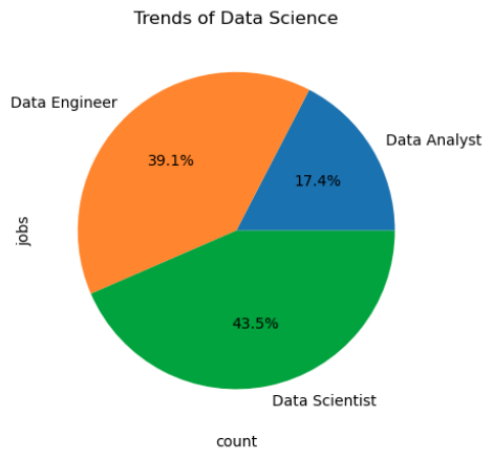
```
In [11]: import pandas as pd
import matplotlib.pyplot as plt
data={"Year": list(range(2010,2021)), "Job Postings": [150,300,450,600,800,1200,1600,2100,2700,3400,4200]}
#print(data)
df=pd.DataFrame(data)
plt.plot(df['Year'],df['Job Postings'],marker='p')
plt.title('Trends of Data Science')
plt.xlabel('year')
plt.ylabel('job postings')
plt.show()
```



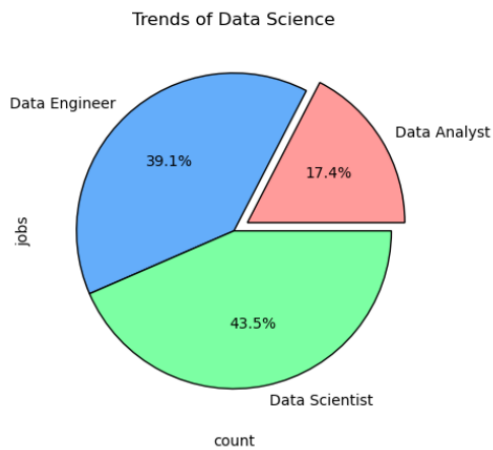
```
In [17]: import pandas as pd
import matplotlib.pyplot as plt
data={"count": [20,45,50], "jobs": ['Data Analyst', 'Data Engineer', 'Data Scientist']}
df=pd.DataFrame(data)
df.plot.bar()
plt.title('Trends of Data Science')
plt.ylabel('count')
plt.xlabel('jobs')
plt.show()
```



```
In [30]: import pandas as pd
import matplotlib.pyplot as plt
count=[20,45,50]
jobs=['Data Analyst','Data Engineer','Data Scientist']
plt.pie(count,labels=jobs,autopct='%1.1f%%')
plt.title('Trends of Data Science')
plt.xlabel('count')
plt.ylabel('jobs')
plt.show()
```



```
In [32]: import pandas as pd
import matplotlib.pyplot as plt
count=[20,45,50]
jobs=['Data Analyst','Data Engineer','Data Scientist']
plt.pie(count,labels=jobs,autopct='%1.1f%%', colors=colors, explode=explode, wedgeprops=dict(edgecolor='black'))
plt.title('Trends of Data Science')
plt.xlabel('count')
plt.ylabel('jobs')
colors = ['#ff9999','#66b3ff','#99ff99']
explode = (0.1, 0, 0)
plt.show()
```




```
In [46]: import pandas as pd
unstructured_data={"Ram 21 Chennai","Aadhi 22 Coimbatore","Leela 23 Trichy"}
lst=list(unstructured_data)
#df=pd.DataFrame(lst,columns=['unstructured_data'])
print(lst)

['Ram 21 Chennai', 'Leela 23 Trichy', 'Aadhi 22 Coimbatore']
```

```
In [45]: import pandas as pd
semistructured_data=["Ram 21 Chennai","Aadhi 22 Coimbatore","Leela 23 Trichy"]
df=pd.DataFrame(semistructured_data,columns=['Semistructured_Data'])
print(df)

   Semistructured_Data
0      Ram 21 Chennai
1  Aadhi 22 Coimbatore
2    Leela 23 Trichy
```

```
In [50]: import pandas as pd
structured_data=pd.DataFrame({
    'ID':[1,2,3],
    'Name':['Lakshya','Lakshetha','Lavanya'],
    'Age':[18,19,20]})
print("Structured Data\n",structured_data)

Structured Data
   ID  Name  Age
0   1  Lakshya  18
1   2  Lakshetha  19
2   3   Lavanya  20
```

```
In [52]: from cryptography.fernet import Fernet
key=Fernet.generate_key()
f=Fernet(key)
token=f.encrypt(b"Save Wayanad")
token
b'...'
f.decrypt(token)
b'Save Wayanad'
key=Fernet.generate_key()
cipher_suite=Fernet(key)
plain_text=b"Save Wayanad"
cipher_text=cipher_suite.encrypt(plain_text)
decrypted_text=cipher_suite.decrypt(cipher_text)
print("Original Data: ",plain_text)
print("Encrypted Data: ",cipher_text)
print("Decrypted Data: ",decrypted_text)
```

Original Data: b'Save Wayanad'
Encrypted Data: b'gAAAAABmvr88_5TTcXTstZ8PdRyis4iUUaIZ1eIOiFZzUcT1qbZcAsVJJpiCU6U1VaCAHB0KyAjpg1Mz3uuz0eUJhWyHXhedfwQ=='
Decrypted Data: b'Save Wayanad'

```
In [5]: import pandas as pd
db=pd.read_csv("C:/Users/DELL/Downloads/archive.zip")
print(db.head())
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1

In []:

```
In [1]: import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
file="C:/Users/DELL/Documents/AI_job_edit.csv"
df=pd.read_csv(file)
print(df.isnull().sum())
df.fillna(df.mean(numeric_only=True),inplace=True)
df.fillna('NA',inplace =True)
print(df)
```

```

Job_Title      2
Industry       0
Company_Size   0
Location       0
AI_Adoption_Level  0
Automation_Risk  0
Required_Skills  0
Salary_USD     2
Remote_Friendly  0
Job_Growth_Projection  0
dtype: int64

```

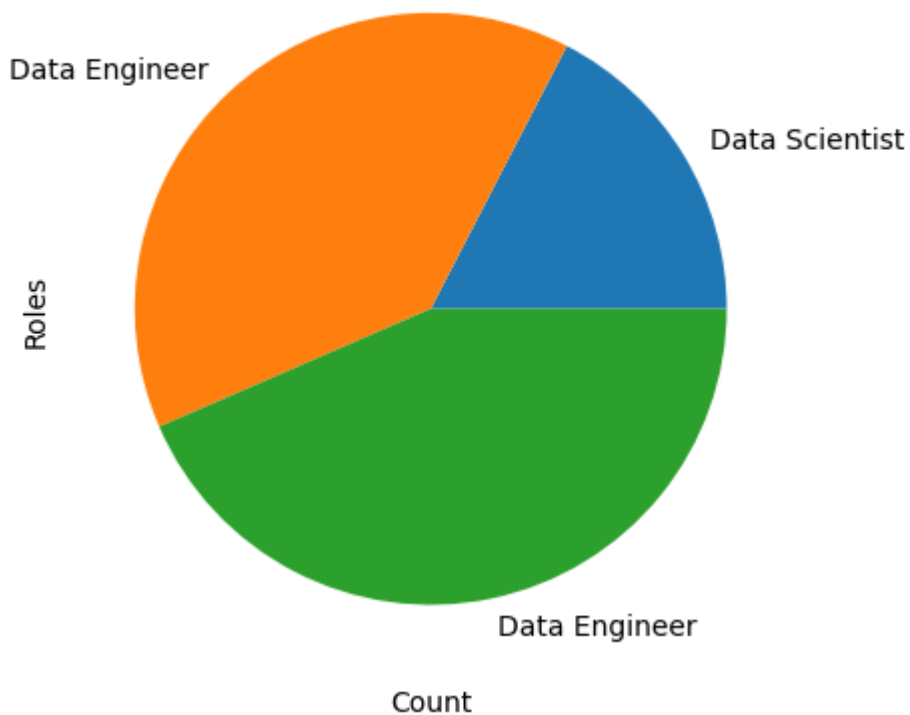
	Job_Title	Industry	Company_Size	Location
0	NA	Entertainment	Small	Dubai
1	Marketing Specialist	Technology	Large	Singapore
2	AI Researcher	Technology	Large	Singapore
3	Sales Manager	Retail	Small	Berlin
4	Cybersecurity Analyst	Entertainment	Small	Tokyo
..
495	Data Scientist	Telecommunications	Medium	Berlin
496	Cybersecurity Analyst	Telecommunications	Small	London
497	Cybersecurity Analyst	Energy	Large	Dubai
498	Operations Manager	Healthcare	Large	Paris
499	HR Manager	Entertainment	Medium	Berlin

	AI_Adoption_Level	Automation_Risk	Required_Skills	Salary_USD
0	Medium	High	UX/UI Design	91176.728431
1	Medium	High	Marketing	91176.728431
2	Medium	High	UX/UI Design	107170.263100
3	Low	High	Project Management	93027.953760
4	Low	Low	JavaScript	87752.922170
..
495	Low	Medium	Machine Learning	105821.394000
496	Low	High	UX/UI Design	119794.992100
497	High	Low	UX/UI Design	79644.933100
498	High	Low	Python	77642.150630
499	Medium	High	Project Management	68764.378920

	Remote_Friendly	Job_Growth_Projection
0	Yes	Growth
1	No	Decline
2	Yes	Growth
3	No	Growth
4	Yes	Decline
..
495	Yes	Stable
496	No	Decline
497	Yes	Stable
498	Yes	Stable
499	Yes	Decline

[500 rows x 10 columns]

Distribution of data science positions



```
In [14]: import pandas as pd
structured_d=pd.DataFrame(
{
    'ID':[1,2,3],
    'Name':['Alice',"Bob","Charlie"],
    'Age':[25,30,35]
})
print("StructuredData\n",structured_d)
```

```
StructuredData
   ID  Name  Age
0   1  Alice  25
1   2   Bob  30
2   3 Charlie  35
```

```
In [15]: import pandas as pd
unstructured_data={"Lavanaya 25 chennai","Lakshaya 19 chennai","Karthik 13 chennai"}
print("Unstructured Data\n",unstructured_data)
```

```
Unstructured Data
{'Lakshaya 19 chennai', 'Karthik 13 chennai', 'Lavanaya 25 chennai'}
```

```
In [17]: import pandas as pd
emp={"employee":[{"Lavana 25 chennai"}, {"Lakshya 19 chennai"}, {"Karthik 13 chennai"}]}
df=pd.DataFrame(emp)
print("Semistructured\n",df)
```

```
Semistructured
      employee
0  {Lavana 25 chennai}
1  {Lakshya 19 chennai}
2  {Karthik 13 chennai}
```

```
In [19]: from cryptography.fernet import Fernet
key=Fernet.generate_key()
f=Fernet(key)
token=f.encrypt(b"Kumaran S")
```

```

token
b'...'
f.decrypt(token)
b'Computer Science'
key=Fernet.generate_key()
cipher_suite=Fernet(key)
plain_text=b'Computer science'
cipher_text=cipher_suite.encrypt(plain_text)
decrypted_text=cipher_suite.decrypt(cipher_text)
print("Original Data:",plain_text)
print("Encrypted Data:",cipher_text)
print("Decrypted Data:",decrypted_text)

```

Original Data: b'Computer science'

Encrypted Data: b'gAAAAABmwr3Ey_XoLiP93u8x9Pful9Qac390SwMK5ud7NpjdH4Wiso95GqRBrf2SjSwIJXWR9UP8Pb_pTUNiRryzqkjyrLTqCASEQocHfG0XZdJki56Mitg='

Decrypted Data: b'Computer science'

```

In [8]: import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("C:/Users/DELL/Downloads/archive (2).zip")
print(df.head())
print(df.isnull().sum())
df['Channel'].fillna(df['Channel'].mean(),inplace=True)
df.dropna(subset=['Region','Fresh','Milk','Grocery','Frozen','Detergents_Paper','De
print(df.describe())
product_summary=df.groupby('Region').agg({
    'Channel':'sum',
    'Fresh':'sum'
}).reset_index()
print(product_summary)
plt.figure(figsize=(10,6))
plt.bar(product_summary['Region'],product_summary['Channel'])
plt.xlabel('Product')
plt.ylabel('Total Sales')
plt.title('Sales over Time')
plt.show()
df['Milk']=pd.to_datetime(df['Milk'])
sales_over_time=df.groupby('Milk').agg({'Channel':'sum'}).reset_index()
plt.figure(figsize=(10,6))
plt.plot(sales_over_time['Milk'],sales_over_time['Milk'])
plt.xlabel('Milk')
plt.ylabel('total sales')
plt.title('Sales over time')
plt.show()

```

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
0	2	3	12669	9656	7561	214	2674	1338
1	2	3	7057	9810	9568	1762	3293	1776
2	2	3	6353	8808	7684	2405	3516	7844
3	1	3	13265	1196	4221	6404	507	1788
4	2	3	22615	5410	7198	3915	1777	5185

Channel
Region
Fresh
Milk
Grocery
Frozen
Detergents_Paper
Delicassen

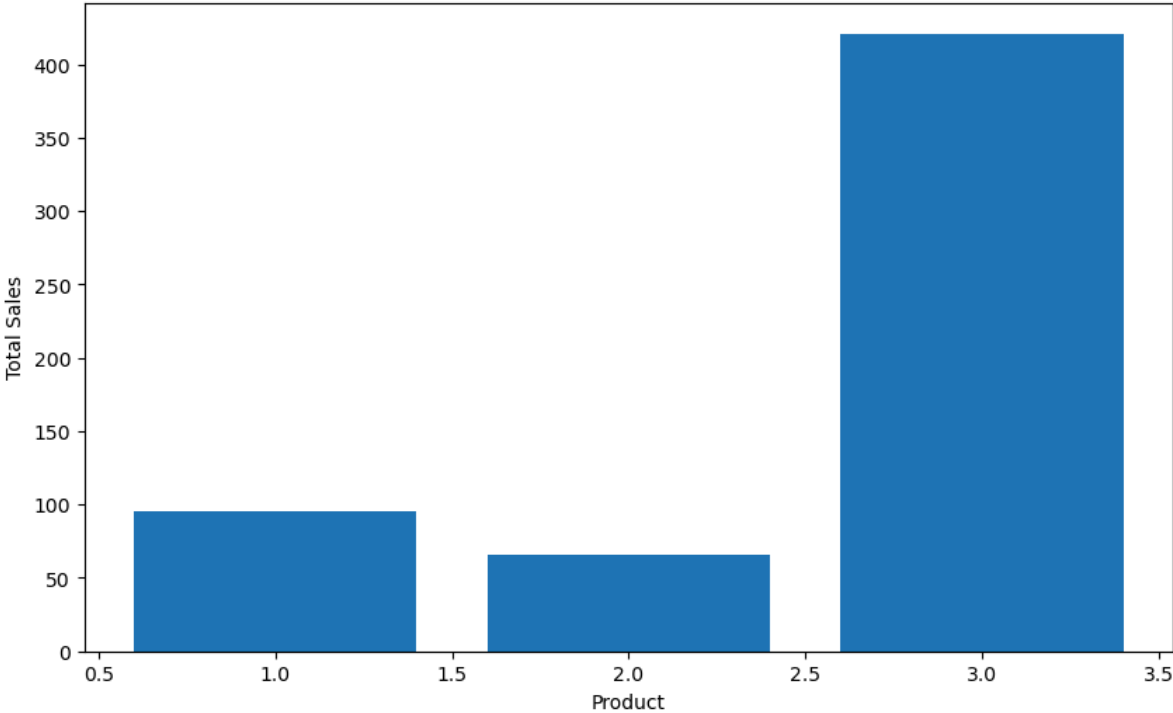
dtype: int64

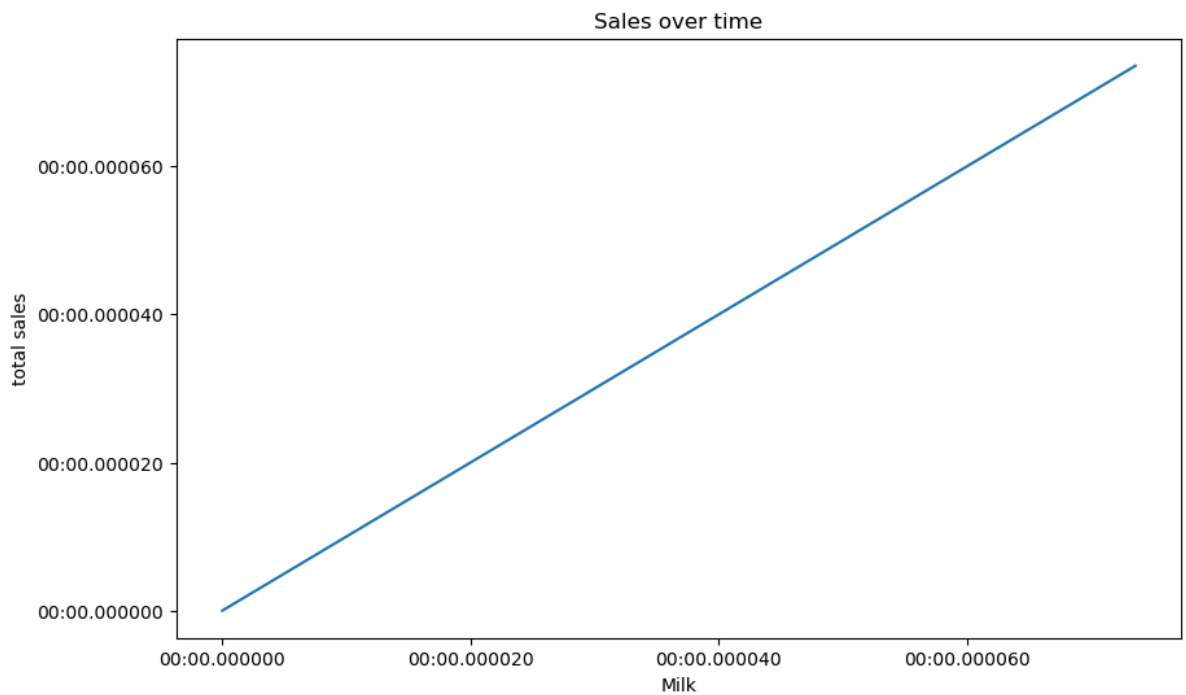
	Channel	Region	Fresh	Milk	Grocery \
count	440.000000	440.000000	440.000000	440.000000	440.000000
mean	1.322727	2.543182	12000.297727	5796.265909	7951.277273
std	0.468052	0.774272	12647.328865	7380.377175	9503.162829
min	1.000000	1.000000	3.000000	55.000000	3.000000
25%	1.000000	2.000000	3127.750000	1533.000000	2153.000000
50%	1.000000	3.000000	8504.000000	3627.000000	4755.500000
75%	2.000000	3.000000	16933.750000	7190.250000	10655.750000
max	2.000000	3.000000	112151.000000	73498.000000	92780.000000

	Frozen	Detergents_Paper	Delicassen
count	440.000000	440.000000	440.000000
mean	3071.931818	2881.493182	1524.870455
std	4854.673333	4767.854448	2820.105937
min	25.000000	3.000000	3.000000
25%	742.250000	256.750000	408.250000
50%	1526.000000	816.500000	965.500000
75%	3554.250000	3922.000000	1820.250000
max	60869.000000	40827.000000	47943.000000

	Region	Channel	Fresh
0	1	95	854833
1	2	66	464721
2	3	421	3960577

Sales over Time





In [29]: `db.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Pregnancies            768 non-null   int64
1   Glucose                768 non-null   int64
2   BloodPressure          768 non-null   int64
3   SkinThickness          768 non-null   int64
4   Insulin                768 non-null   int64
5   BMI                   768 non-null   float64
6   DiabetesPedigreeFunction 768 non-null   float64
7   Age                   768 non-null   int64
8   Outcome               768 non-null   int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [30]: `db.describe()`

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPer
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	

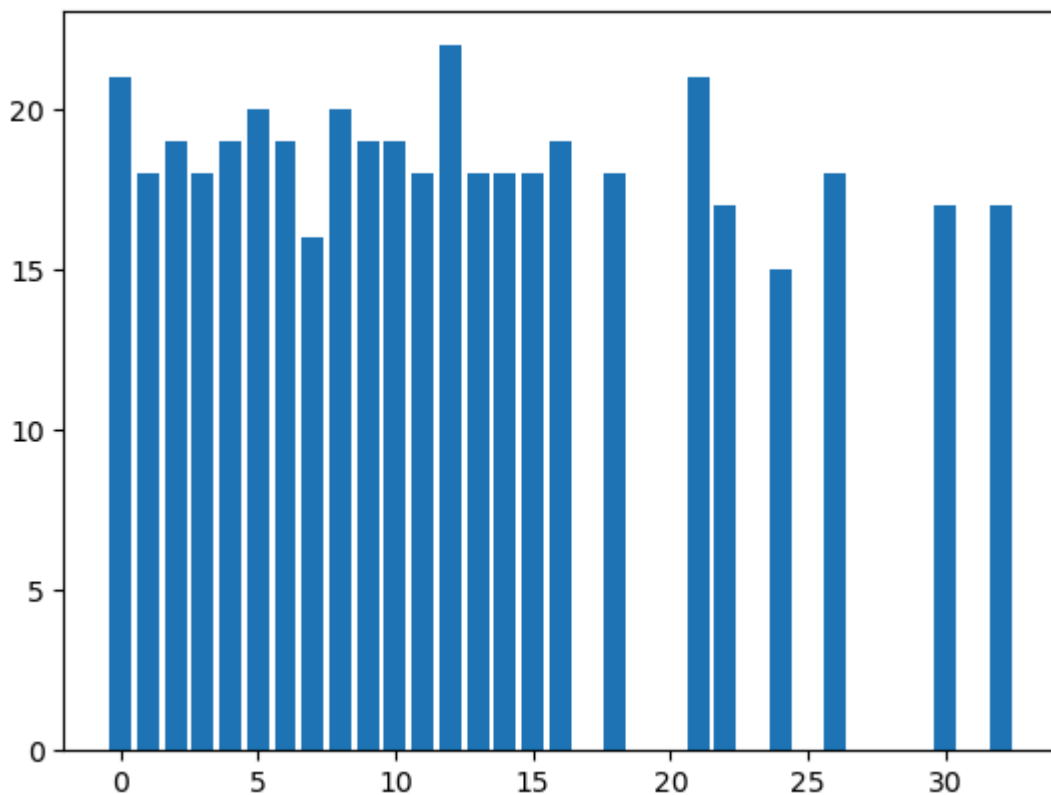
In [18]: `import pandas as pd`
`import matplotlib.pyplot as plt`


```
df=pd.read_csv("C:/Users/DELL/Downloads/archive (3).zip")
print(df)
plt.bar(df['absences'],df['age'])
plt.show()
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	\
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	
1	GP	F	17	U	GT3	T	1	1	at_home	other	
2	GP	F	15	U	LE3	T	1	1	at_home	other	
3	GP	F	15	U	GT3	T	4	2	health	services	
4	GP	F	16	U	GT3	T	3	3	other	other	
..	
644	MS	F	19	R	GT3	T	2	3	services	other	
645	MS	F	18	U	LE3	T	3	1	teacher	services	
646	MS	F	18	U	GT3	T	1	1	other	other	
647	MS	M	17	U	LE3	T	3	1	services	services	
648	MS	M	18	R	LE3	T	3	2	services	other	

	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	...	4	3	4	1	1	3	4	0	11	11
1	...	5	3	3	1	1	3	2	9	11	11
2	...	4	3	2	2	3	3	6	12	13	12
3	...	3	2	2	1	1	5	0	14	14	14
4	...	4	3	2	1	2	5	0	11	13	13
..
644	...	5	4	2	1	2	5	4	10	11	10
645	...	4	3	4	1	1	1	4	15	15	16
646	...	1	1	1	1	1	5	6	11	12	9
647	...	2	4	5	3	4	2	6	10	10	10
648	...	4	4	1	3	4	5	4	10	11	11

[649 rows x 33 columns]



```
In [21]: import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("C:/Users/DELL/Downloads/archive (6).zip")
print(df)
```

	st_name	year	ac_no	ac_name	ac_type	\
0	Andhra Pradesh	1978.0	1	Ichapuram	GEN	
1	Andhra Pradesh	1978.0	1	Ichapuram	GEN	
2	Andhra Pradesh	1978.0	1	Ichapuram	GEN	
3	Andhra Pradesh	1978.0	1	Ichapuram	GEN	
4	Andhra Pradesh	1978.0	1	Ichapuram	GEN	
...
327289	West Bengal	2011.0	294	Murarai	GEN	
327290	West Bengal	2011.0	294	Murarai	GEN	
327291	West Bengal	2011.0	294	Murarai	GEN	
327292	West Bengal	2011.0	294	Murarai	GEN	
327293	West Bengal	2011.0	294	Murarai	GEN	

	cand_name	cand_sex	partyname	\
0	Manabala Ramarao	M	Independent	
1	Appadu Sahu	M	Independent	
2	Uppada Rangababu	M	Indian National Congress	
3	Kalla Balarama Swamy	M	Indian National Congress (I)	
4	Bendalam Venkatesam Sarma	M	Janata Party	
...
327289	Ajrail Md	M	IND	
327290	Shiblal Sardar	M	BSP	
327291	Sudhiranjan Das Goswami	M	BJP	
327292	Elahi Kamre Dr	M	CPM	
327293	Nure Alam Chowdhury	M	AITC	

	partyabbre	totvotpoll	electors
0	IND	813.0	83247
1	IND	1743.0	83247
2	INC	4427.0	83247
3	INC(I)	19805.0	83247
4	JNP	34251.0	83247
...
327289	IND	2471.0	189320
327290	BSP	3074.0	189320
327291	BJP	4961.0	189320
327292	CPM	73414.0	189320
327293	AITC	77817.0	189320

[327294 rows x 11 columns]

In []:

In []:

In []:

```
In [35]: import pandas as pd
import matplotlib.pyplot as plt
import zipfile

# Define the path to your ZIP file
zip_path = "C:/Users/DELL/Downloads/archive (6).zip"

# Read the CSV file from the ZIP archive
with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    # List the files in the ZIP archive
    csv_files = zip_ref.namelist()
    # Assume there's only one CSV file, use the first one
    csv_file = csv_files[0]
    # Read the CSV file into a DataFrame
    df = pd.read_csv(zip_ref.open(csv_file))
```

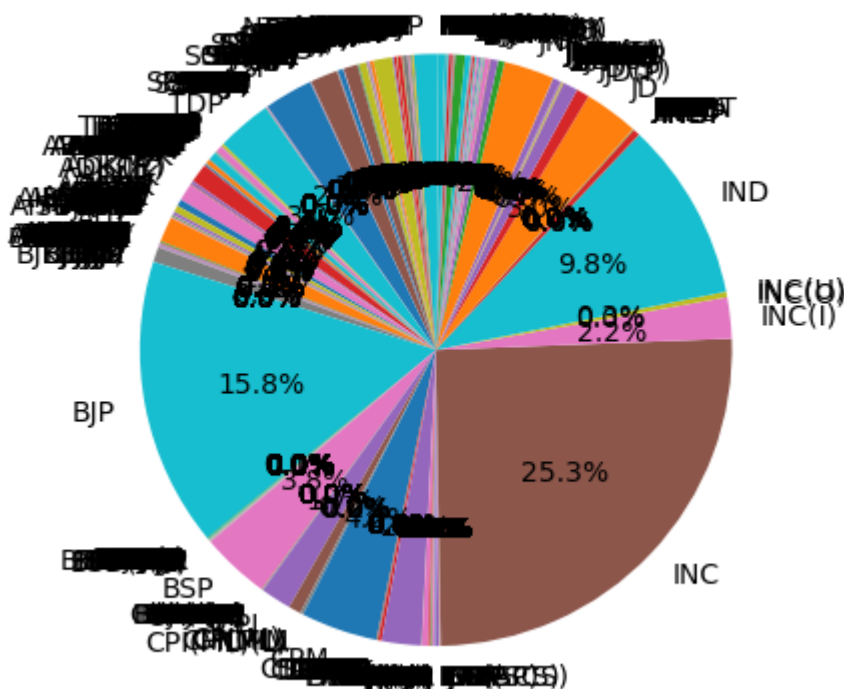
```
# Aggregate data by 'partyabbre'
agg_df = df.groupby('partyabbre')['totvotpoll'].sum().reset_index()

# Verify aggregated data
print(agg_df.head())

# Plot pie chart
plt.pie(agg_df['totvotpoll'], labels=agg_df['partyabbre'], autopct='%1.1f%%', startangle=90)
plt.title('Pie Chart of Total Votes Poll by Party Abbreviation')
plt.show()
```

	partyabbre	totvotpoll
0	A S P	3112.0
1	AAAP	7419281.0
2	AACP	3024.0
3	AAMP	2548.0
4	AAP	8008.0

Pie Chart of Total Votes Poll by Party Abbreviation



```
In [17]: import pandas as pd
df=pd.read_csv("C:/Users/DELL/Downloads/archive (7).zip")
print(df.isnull().sum())
df.fillna(df.mean(numeric_only=True),inplace=True)
print(df.isnull().sum())
print(df.head())
```

school	0
sex	0
age	0
address	0
famsize	0
Pstatus	0
Medu	0
Fedu	0
Mjob	0
Fjob	0
reason	0
guardian	0
traveltime	0
studytime	0
failures	0
schoolsup	0
famsup	0
paid	0
activities	0
nursery	0
higher	0
internet	0
romantic	0
famrel	0
freetime	0
goout	0
Dalc	0
Walc	0
health	0
absences	0
G1	0
G2	0
G3	0
dtype: int64	
school	0
sex	0
age	0
address	0
famsize	0
Pstatus	0
Medu	0
Fedu	0
Mjob	0
Fjob	0
reason	0
guardian	0
traveltime	0
studytime	0
failures	0
schoolsup	0
famsup	0
paid	0
activities	0
nursery	0
higher	0
internet	0
romantic	0
famrel	0
freetime	0
goout	0
Dalc	0
Walc	0
health	0
absences	0

```

G1          0
G2          0
G3          0
dtype: int64

```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	\
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	
3	GP	F	15	U	GT3	T	4	2	health	services	...	
4	GP	F	16	U	GT3	T	3	3	other	other	...	

	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	4	3	4	1	1	3	4	0	11	11
1	5	3	3	1	1	3	2	9	11	11
2	4	3	2	2	3	3	6	12	13	12
3	3	2	2	1	1	5	0	14	14	14
4	4	3	2	1	2	5	0	11	13	13

[5 rows x 33 columns]

```

In [28]: import pandas as pd
df=pd.read_csv("C:/Users/DELL/Downloads/archive (7).zip")
print(df.isnull().sum())
df.fillna(df.median(numeric_only=True),inplace=True)
print(df.isnull().sum())
print(df)

```

school	0
sex	0
age	0
address	0
famsize	0
Pstatus	0
Medu	0
Fedu	0
Mjob	0
Fjob	0
reason	0
guardian	0
traveltime	0
studytime	0
failures	0
schoolsup	0
famsup	0
paid	0
activities	0
nursery	0
higher	0
internet	0
romantic	0
famrel	0
freetime	0
goout	0
Dalc	0
Walc	0
health	0
absences	0
G1	0
G2	0
G3	0
dtype: int64	
school	0
sex	0
age	0
address	0
famsize	0
Pstatus	0
Medu	0
Fedu	0
Mjob	0
Fjob	0
reason	0
guardian	0
traveltime	0
studytime	0
failures	0
schoolsup	0
famsup	0
paid	0
activities	0
nursery	0
higher	0
internet	0
romantic	0
famrel	0
freetime	0
goout	0
Dalc	0
Walc	0
health	0
absences	0

```

G1          0
G2          0
G3          0
dtype: int64

```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	\
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	
1	GP	F	17	U	GT3	T	1	1	at_home	other	
2	GP	F	15	U	LE3	T	1	1	at_home	other	
3	GP	F	15	U	GT3	T	4	2	health	services	
4	GP	F	16	U	GT3	T	3	3	other	other	
..	
644	MS	F	19	R	GT3	T	2	3	services	other	
645	MS	F	18	U	LE3	T	3	1	teacher	services	
646	MS	F	18	U	GT3	T	1	1	other	other	
647	MS	M	17	U	LE3	T	3	1	services	services	
648	MS	M	18	R	LE3	T	3	2	services	other	

	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	...	4	3	4	1	1	3	4	0	11	11
1	...	5	3	3	1	1	3	2	9	11	11
2	...	4	3	2	2	3	3	6	12	13	12
3	...	3	2	2	1	1	5	0	14	14	14
4	...	4	3	2	1	2	5	0	11	13	13
..
644	...	5	4	2	1	2	5	4	10	11	10
645	...	4	3	4	1	1	1	4	15	15	16
646	...	1	1	1	1	1	5	6	11	12	9
647	...	2	4	5	3	4	2	6	10	10	10
648	...	4	4	1	3	4	5	4	10	11	11

[649 rows x 33 columns]

```

In [32]: import pandas as pd
df=pd.read_csv("C:/Users/DELL/Downloads/archive (7).zip")
print(df.isnull().sum())
df.fillna(df.mode(numeric_only=True),inplace=True)
print(df.isnull().sum())
print(df)

```

school	0
sex	0
age	0
address	0
famsize	0
Pstatus	0
Medu	0
Fedu	0
Mjob	0
Fjob	0
reason	0
guardian	0
traveltime	0
studytime	0
failures	0
schoolsup	0
famsup	0
paid	0
activities	0
nursery	0
higher	0
internet	0
romantic	0
famrel	0
freetime	0
goout	0
Dalc	0
Walc	0
health	0
absences	0
G1	0
G2	0
G3	0
dtype: int64	
school	0
sex	0
age	0
address	0
famsize	0
Pstatus	0
Medu	0
Fedu	0
Mjob	0
Fjob	0
reason	0
guardian	0
traveltime	0
studytime	0
failures	0
schoolsup	0
famsup	0
paid	0
activities	0
nursery	0
higher	0
internet	0
romantic	0
famrel	0
freetime	0
goout	0
Dalc	0
Walc	0
health	0
absences	0


```

G1      0
G2      0
G3      0
dtype: int64

```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	\
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	
1	GP	F	17	U	GT3	T	1	1	at_home	other	
2	GP	F	15	U	LE3	T	1	1	at_home	other	
3	GP	F	15	U	GT3	T	4	2	health	services	
4	GP	F	16	U	GT3	T	3	3	other	other	
..	
644	MS	F	19	R	GT3	T	2	3	services	other	
645	MS	F	18	U	LE3	T	3	1	teacher	services	
646	MS	F	18	U	GT3	T	1	1	other	other	
647	MS	M	17	U	LE3	T	3	1	services	services	
648	MS	M	18	R	LE3	T	3	2	services	other	

	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	...	4	3	4	1	1	3	4	0	11	11
1	...	5	3	3	1	1	3	2	9	11	11
2	...	4	3	2	2	3	3	6	12	13	12
3	...	3	2	2	1	1	5	0	14	14	14
4	...	4	3	2	1	2	5	0	11	13	13
..
644	...	5	4	2	1	2	5	4	10	11	10
645	...	4	3	4	1	1	1	4	15	15	16
646	...	1	1	1	1	1	5	6	11	12	9
647	...	2	4	5	3	4	2	6	10	10	10
648	...	4	4	1	3	4	5	4	10	11	11

[649 rows x 33 columns]

```

In [30]: import pandas as pd
df=pd.read_csv("C:/Users/DELL/Downloads/archive (2).zip")
print(df.isnull().sum())
df.fillna(df.mean(),inplace=True)
print(df)

```

```

Channel      0
Region       0
Fresh        0
Milk         0
Grocery      0
Frozen       0
Detergents_Paper 0
Delicassen   0
dtype: int64

```

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper \
0	2	3	12669	9656	7561	214	2674
1	2	3	7057	9810	9568	1762	3293
2	2	3	6353	8808	7684	2405	3516
3	1	3	13265	1196	4221	6404	507
4	2	3	22615	5410	7198	3915	1777
..
435	1	3	29703	12051	16027	13135	182
436	1	3	39228	1431	764	4510	93
437	2	3	14531	15488	30243	437	14841
438	1	3	10290	1981	2232	1038	168
439	1	3	2787	1698	2510	65	477

	Delicassen
0	1338
1	1776
2	7844
3	1788
4	5185
..	...
435	2204
436	2346
437	1867
438	2125
439	52

[440 rows x 8 columns]

```

In [31]: import pandas as pd
df=pd.read_csv("C:/Users/DELL/Downloads/archive (2).zip")
print(df.isnull().sum())
df.fillna(df.mode(),inplace=True)
print(df)

```

```

Channel      0
Region       0
Fresh        0
Milk         0
Grocery      0
Frozen       0
Detergents_Paper 0
Delicassen   0
dtype: int64

```

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper \
0	2	3	12669	9656	7561	214	2674
1	2	3	7057	9810	9568	1762	3293
2	2	3	6353	8808	7684	2405	3516
3	1	3	13265	1196	4221	6404	507
4	2	3	22615	5410	7198	3915	1777
..
435	1	3	29703	12051	16027	13135	182
436	1	3	39228	1431	764	4510	93
437	2	3	14531	15488	30243	437	14841
438	1	3	10290	1981	2232	1038	168
439	1	3	2787	1698	2510	65	477

	Delicassen
0	1338
1	1776
2	7844
3	1788
4	5185
..	...
435	2204
436	2346
437	1867
438	2125
439	52

[440 rows x 8 columns]

```

In [23]: import pandas as pd
df=pd.read_csv("C:/Users/DELL/Downloads/archive (2).zip")
print(df.isnull().sum())
df.fillna(df.median(),inplace=True)
print(df)

```

```

Channel      0
Region       0
Fresh        0
Milk         0
Grocery      0
Frozen       0
Detergents_Paper 0
Delicassen   0
dtype: int64

```

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper \
0	2	3	12669	9656	7561	214	2674
1	2	3	7057	9810	9568	1762	3293
2	2	3	6353	8808	7684	2405	3516
3	1	3	13265	1196	4221	6404	507
4	2	3	22615	5410	7198	3915	1777
..
435	1	3	29703	12051	16027	13135	182
436	1	3	39228	1431	764	4510	93
437	2	3	14531	15488	30243	437	14841
438	1	3	10290	1981	2232	1038	168
439	1	3	2787	1698	2510	65	477

	Delicassen
0	1338
1	1776
2	7844
3	1788
4	5185
..	...
435	2204
436	2346
437	1867
438	2125
439	52

[440 rows x 8 columns]

```

In [22]: import pandas as pd
df=pd.read_csv("C:/Users/DELL/Downloads/sales.csv")
print(df.isnull().sum())
df.fillna(df.mean(numeric_only=True),inplace=True)
print(df)

```

```

Date      0
Product    0
Sales      0
Quantity   0
Region     0
dtype: int64

```

	Date	Product	Sales	Quantity	Region
0	01-01-2023	Product A	200	4	North
1	02-01-2023	Product B	150	3	South
2	03-01-2023	Product A	220	5	North
3	04-01-2023	Product C	300	6	East
4	05-01-2023	Product B	180	4	West
5	06-01-2023	Product A	210	5	North
6	07-01-2023	Product C	320	7	East
7	08-01-2023	Product B	160	3	South
8	09-01-2023	Product A	230	6	North
9	10-01-2023	Product C	310	7	East
10	11-01-2023	Product B	190	4	West
11	12-01-2023	Product A	240	6	North
12	13-01-2023	Product C	330	8	East
13	14-01-2023	Product B	170	3	South
14	15-01-2023	Product A	250	7	North
15	16-01-2023	Product C	340	8	East

```

In [21]: import pandas as pd
df=pd.read_csv("C:/Users/DELL/Downloads/sales.csv")
print(df.isnull().sum())
df.fillna(df.mode(numeric_only=True),inplace=True)
print(df)

```

```

Date      0
Product    0
Sales      0
Quantity   0
Region     0
dtype: int64

```

	Date	Product	Sales	Quantity	Region
0	01-01-2023	Product A	200	4	North
1	02-01-2023	Product B	150	3	South
2	03-01-2023	Product A	220	5	North
3	04-01-2023	Product C	300	6	East
4	05-01-2023	Product B	180	4	West
5	06-01-2023	Product A	210	5	North
6	07-01-2023	Product C	320	7	East
7	08-01-2023	Product B	160	3	South
8	09-01-2023	Product A	230	6	North
9	10-01-2023	Product C	310	7	East
10	11-01-2023	Product B	190	4	West
11	12-01-2023	Product A	240	6	North
12	13-01-2023	Product C	330	8	East
13	14-01-2023	Product B	170	3	South
14	15-01-2023	Product A	250	7	North
15	16-01-2023	Product C	340	8	East

```

In [40]: import pandas as pd
import numpy as np
df=pd.read_csv("C:/Users/DELL/Downloads/sales.csv")
print(df.isnull().sum())
df.fillna(df.median(numeric_only=True),inplace=True)
df.dropna(inplace=True)
print(df)

```

```

Date      0
Product   1
Sales     0
Quantity  0
Region    0
dtype: int64

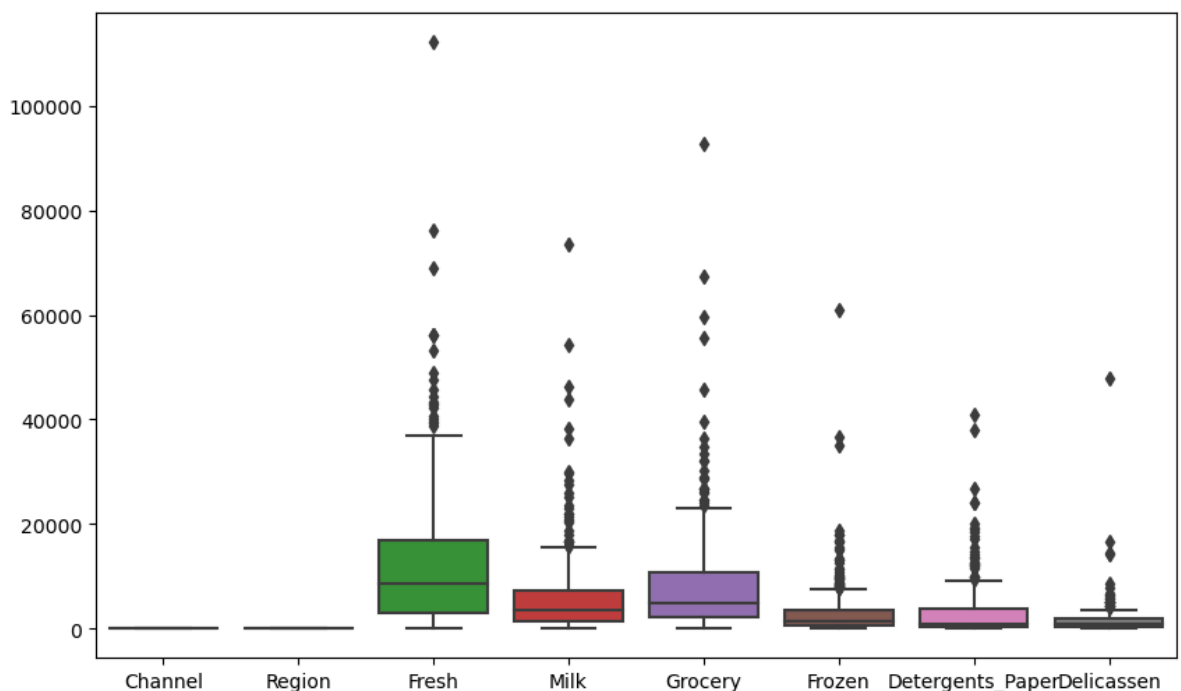
```

	Date	Product	Sales	Quantity	Region
1	02-01-2023	Product B	150	3	South
2	03-01-2023	Product A	220	5	North
3	04-01-2023	Product C	300	6	East
4	05-01-2023	Product B	180	4	West
5	06-01-2023	Product A	210	5	North
6	07-01-2023	Product C	320	7	East
7	08-01-2023	Product B	160	3	South
8	09-01-2023	Product A	230	6	North
9	10-01-2023	Product C	310	7	East
10	11-01-2023	Product B	190	4	West
11	12-01-2023	Product A	240	6	North
12	13-01-2023	Product C	330	8	East
13	14-01-2023	Product B	170	3	South
14	15-01-2023	Product A	250	7	North
15	16-01-2023	Product C	340	8	East

```

In [43]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
db=pd.read_csv("C:/Users/DELL/Downloads/archive (2).zip")
plt.figure(figsize=(10,6))
sns.boxplot(data=db)
plt.show()
ddf=db[(np.abs(stats.zscore(db)<3)).all(axis=1)]]

```



```

In [ ]:

```

```
In [ ]: import numpy as np
import pandas as pd
df=pd.read_csv('Salary_data.csv')
df
```

```
In [19]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   YearsExperience  30 non-null    float64
 1   Salary          30 non-null    int64
dtypes: float64(1), int64(1)
memory usage: 612.0 bytes
```

```
In [3]: df.dropna(inplace=True)
```

```
In [4]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   YearsExperience  30 non-null    float64
 1   Salary          30 non-null    int64
dtypes: float64(1), int64(1)
memory usage: 612.0 bytes
```

```
In [5]: df.describe()
```

```
Out[5]:
```

	YearsExperience	Salary
count	30.000000	30.000000
mean	5.313333	76003.000000
std	2.837888	27414.429785
min	1.100000	37731.000000
25%	3.200000	56720.750000
50%	4.700000	65237.000000
75%	7.700000	100544.750000
max	10.500000	122391.000000

```
In [6]: features=df.iloc[:,[0]].values
label=df.iloc[:,[1]].values
```

```
In [7]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(features,label,test_size=0.2,random_st
```

```
In [20]: from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train,y_train)
```

```
Out[20]:
```

LinearRegression

LinearRegression()

```
In [21]: model.score(x_train,y_train)
```

```
Out[21]: 0.9603182547438908
```

```
In [23]: model.score(x_test,y_test)
```

```
Out[23]: 0.9184170849214232
```

```
In [24]: model.coef_
```

```
Out[24]: array([[9281.30847068]])
```

```
In [25]: model.intercept_
```

```
Out[25]: array([27166.73682891])
```

```
In [26]: import pickle  
pickle.dump(model,open('SalaryPred.model','wb'))
```

```
In [27]: model=pickle.load(open('SalaryPred.model','rb'))
```

```
In [28]: yr_of_exp=float(input("Enter Years of Experience: "))  
yr_of_exp_NP=np.array([[yr_of_exp]])  
Salary=model.predict(yr_of_exp_NP)
```

Enter Years of Experience: 44

```
In [ ]:
```

```
In [29]: print("Estimated Salary for {} years of experience is {}: " .format(yr_of_exp,Salary))
```

Estimated Salary for 44.0 years of experience is [[435544.30953887]]:

```
In [ ]:
```



```
In [ ]: import numpy as np
import pandas as pd
df=pd.read_csv('Salary_data.csv')
df
```

```
In [19]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   YearsExperience  30 non-null     float64
 1   Salary          30 non-null     int64
dtypes: float64(1), int64(1)
memory usage: 612.0 bytes
```

```
In [3]: df.dropna(inplace=True)
```

```
In [4]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   YearsExperience  30 non-null     float64
 1   Salary          30 non-null     int64
dtypes: float64(1), int64(1)
memory usage: 612.0 bytes
```

```
In [5]: df.describe()
```

```
Out[5]:
```

	YearsExperience	Salary
count	30.000000	30.000000
mean	5.313333	76003.000000
std	2.837888	27414.429785
min	1.100000	37731.000000
25%	3.200000	56720.750000
50%	4.700000	65237.000000
75%	7.700000	100544.750000
max	10.500000	122391.000000

```
In [6]: features=df.iloc[:,[0]].values
label=df.iloc[:,[1]].values
```

```
In [7]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(features,label,test_size=0.2,random_st
```

```
In [20]: from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train,y_train)
```

```
Out[20]:
```

LinearRegression

LinearRegression()

```
In [21]: model.score(x_train,y_train)
```

```
Out[21]: 0.9603182547438908
```

```
In [23]: model.score(x_test,y_test)
```

```
Out[23]: 0.9184170849214232
```

```
In [24]: model.coef_
```

```
Out[24]: array([[9281.30847068]])
```

```
In [25]: model.intercept_
```

```
Out[25]: array([27166.73682891])
```

```
In [26]: import pickle  
pickle.dump(model,open('SalaryPred.model','wb'))
```

```
In [27]: model=pickle.load(open('SalaryPred.model','rb'))
```

```
In [28]: yr_of_exp=float(input("Enter Years of Experience: "))  
yr_of_exp_NP=np.array([[yr_of_exp]])  
Salary=model.predict(yr_of_exp_NP)
```

Enter Years of Experience: 44

```
In [ ]:
```

```
In [29]: print("Estimated Salary for {} years of experience is {}: " .format(yr_of_exp,Salary))
```

Estimated Salary for 44.0 years of experience is [[435544.30953887]]:

```
In [ ]:
```