230701156-1

November 21, 2024

[18]:

```
#unstructured data
information1="veena violin flute 20 25 26 15 17 18"
print("unstructured data:\n")
print("\n",information1)
print("\n")

#structured data
import pandas as pd
data={'instruments':['veena','violin','flute'],
      'no_registrations':[20,25,26],
       'no_eligible_students':[15,17,18]}

data_frame=pd.DataFrame(data)
print("structured data:\n")
print(data_frame)

#semistructured data
data=[
    {"instrument":"veena",
     "no_of_students":{"no_of_registration":20,"no_of_eligible_students":15}},

      {
     "no_of_students":{"no_of_registration":25,"no_of_eligible_students":17}
      },

       {"instrument":"flute",
     "no_of_students":{"no_of_registration":26}}
     ]

pd.json_normalize(data, max_level=1)
```

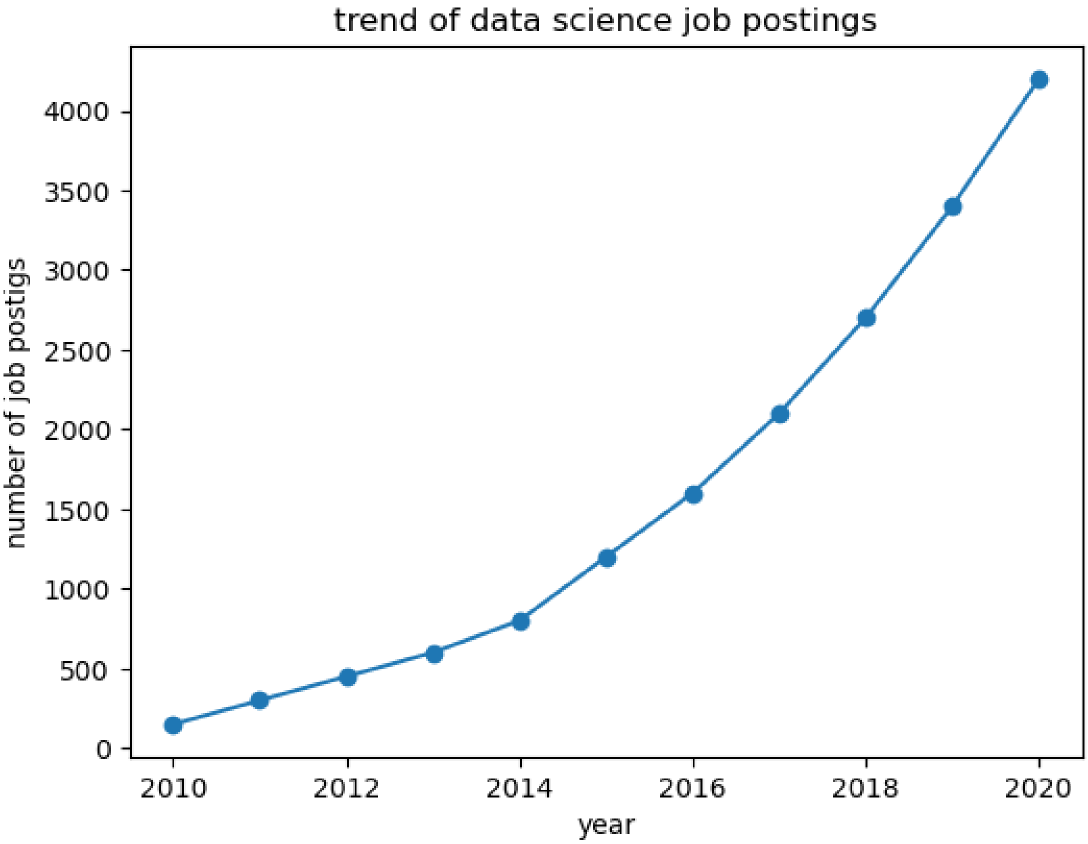
```
unstructured data:



 veena violin flute 20 25 26 15 17 18



structured data:
  instruments  no_registrations  no_eligible_students
0       veena                20                    15
1      violin                25                    17
2       flute                26                    18
```

[18]:

```
  instrument  no_of_students.no_of_registration  \
0      veena                                 20
1        NaN                                 25
2      flute                                 26

   no_of_students.no_of_eligible_students
0                                    15.0
1                                    17.0
2                                     NaN
```

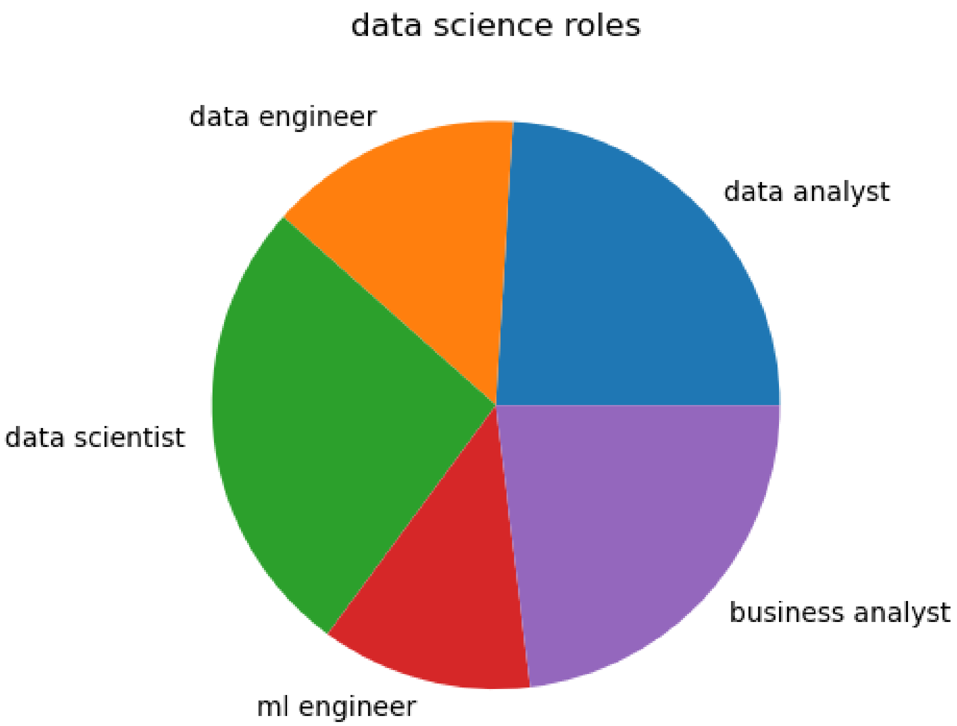
[26]:

```
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]}
df=pd.DataFrame(data)
plt.plot(df['year'],df['job postings'],marker='o')
plt.title('trend of data science job postings')
plt.xlabel('year')
plt.ylabel('number of job postigs')
plt.show()
```



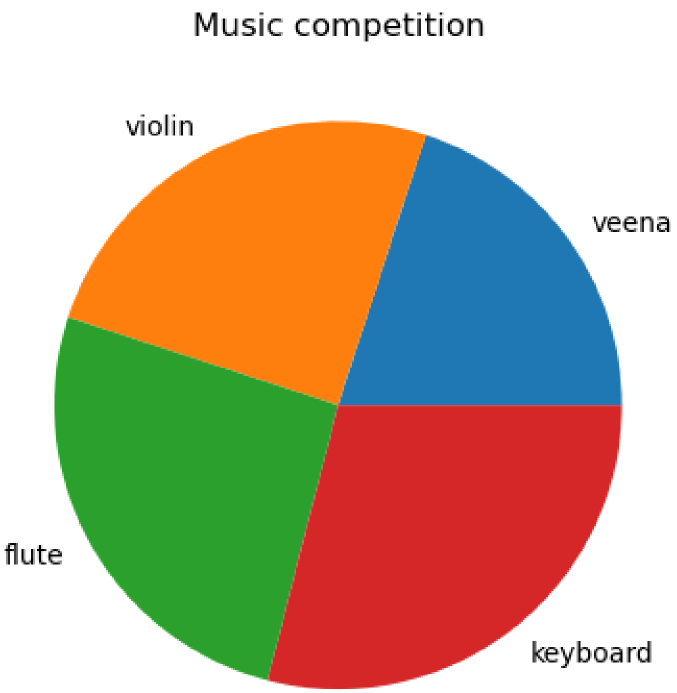
[25]:

```
import matplotlib.pyplot as plt
roles=['data analyst','data engineer','data scientist','ml engineer','business␣
 ↪analyst']
data=[500,300,550,250,480]
plt.pie(data,labels=roles)
plt.title('data science roles')

plt.show()
```



```
[4]: import matplotlib.pyplot as plt
     instruments=['veena','violin','flute','keyboard']
     data=[20,25,26,29]
     plt.pie(data,labels=instruments)
     plt.title('Music competition')
```

```
[4]: Text(0.5, 1.0, 'Music competition')
```



[7]:

```
from cryptography.fernet import Fernet
key=Fernet.generate_key()
f=Fernet(key)
token=f.encrypt(b"I love Music")
token
b'...'
f.decrypt(token)
b'I love Music'
key=Fernet.generate_key()
cipher_suite=Fernet(key)
plain_text=b"I love Music"
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'I love Music'
Encrypted data: b'gAAAAABmtEP4DuOPE3OrJK_3H_eaYPotPhwtHLm-
eRzKCkkxTJbdJQOQhbirgPz4Qy4dUNJpd6qI0ucJz4yKqlU9A_NFaAI_hA=='
Decrypted data: b'I love Music'
```

[21]:

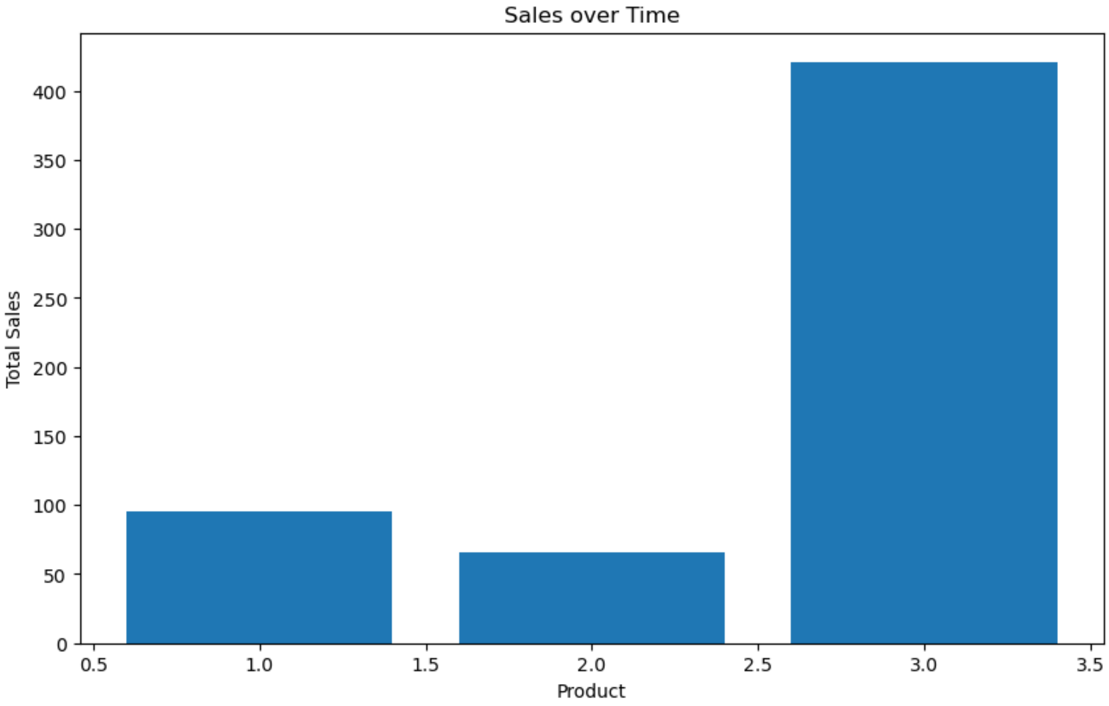
```
import pandas as pd
db=pd.read_csv("C:/Users/DELL/Downloads/archive/diabetes.csv")
print(db.head())
list(db.columns)
i=db[db['BMI']<30.0].index.tolist()
```

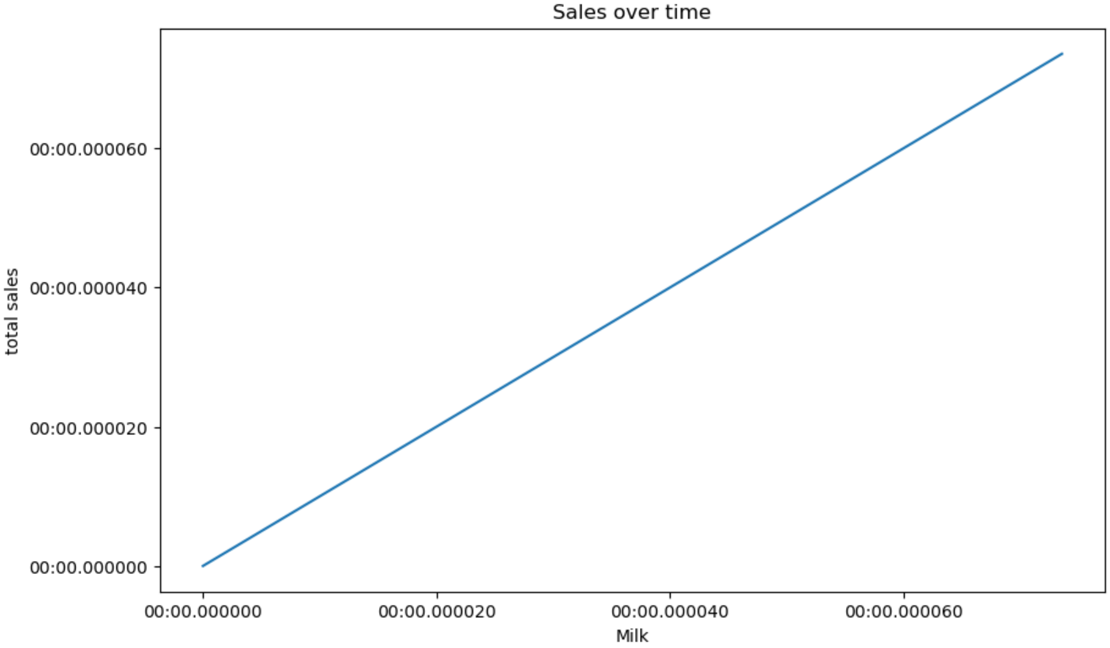
```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       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
```

[20]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df=pd.read_csv('C:/Users/DELL/Downloads/wholesale+customers/Wholesale customers␣
  ↪data.csv')
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','Delicassen'],
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()
```

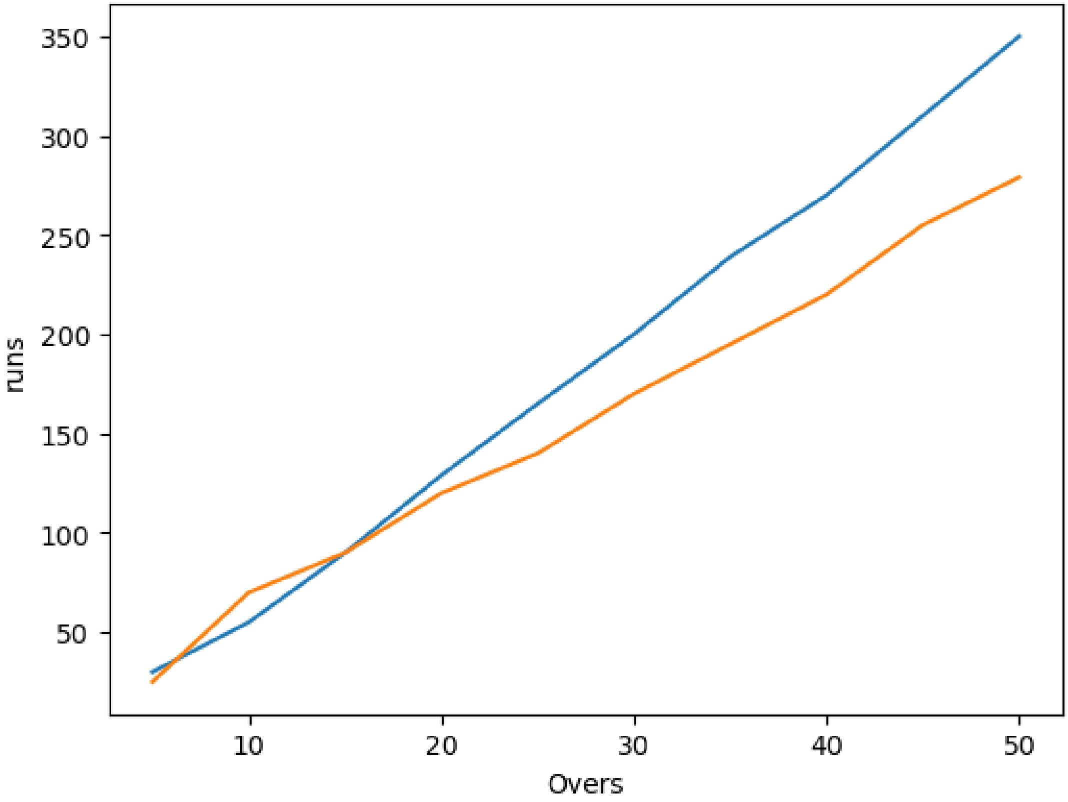
```
   Channel  Region  Fresh  Milk  Grocery  Frozen  Detergents_Paper  Delicassen
0        2       3  12669  9656     7561     214              2674        1338
1        2       3   7057  9810     9568    1762              3293        1776
2        2       3   6353  8808     7684    2405           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



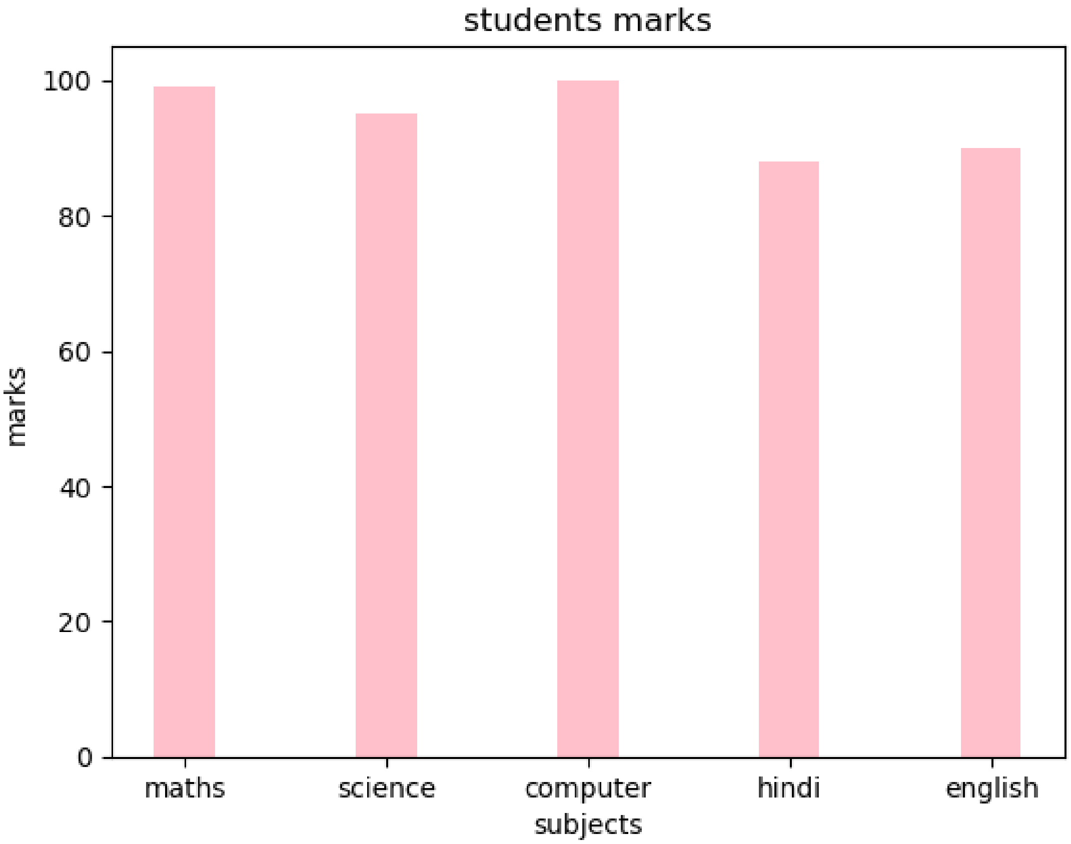


[5]:
```
import matplotlib.pyplot as plt
Overs=list(range(5,51,5))
Indian_Score=[30,55,90,129,165,200,239,270,310,350]
```

```
Srilankan_Score=[25,70,90,120,140,170,195,220,255,279]
plt.plot(Overs,Indian_Score)
plt.plot(Overs,Srilankan_Score)
plt.xlabel('Overs')
plt.ylabel('runs')
plt.show()
```

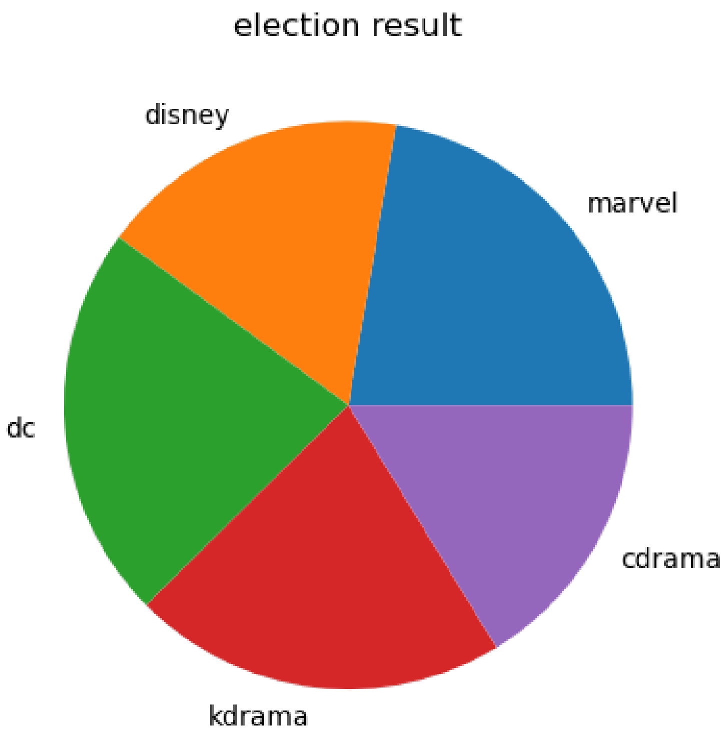


[2]:
```
import pandas as pd
import matplotlib.pyplot as plt
sub=['maths','science','computer','hindi','english']
marks=[99,95,100,88,90]

plt.bar(sub,marks,width=0.3,color='pink')
plt.title('students marks')
plt.xlabel('subjects')
plt.ylabel('marks')
plt.show()
```



[3]:

```
import matplotlib.pyplot as plt
parties=['marvel','disney','dc','kdrama','cdrama']
views=[678,535,683,647,494]
plt.pie(views,labels=parties)
plt.title('election result')
plt.show()
```

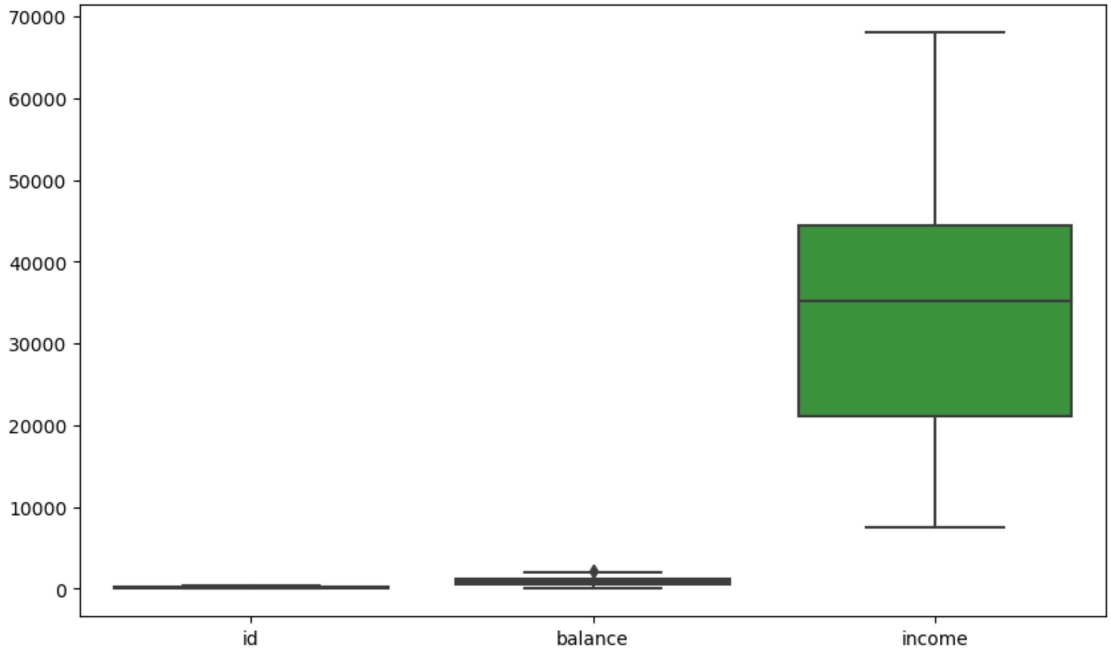


[18]:
```
import pandas as pd
db=pd.read_csv("C:/Users/DELL/Downloads/data.csv")
print(db.isnull().sum())
a=db['balance'].fillna(db['balance'].mean(),inplace=True)
print(db.isnull().sum())
```

```
id          0
default     0
student     0
balance    33
income     21
dtype: int64
id          0
default     0
student     0
balance     0
income     21
dtype: int64
```

[21]:
```
import pandas as pd
import seaborn as sns
import numpy as np
```

```
db=pd.read_csv("C:/Users/DELL/Downloads/data.csv")
plt.figure(figsize=(10,6))
sns.boxplot(data=db)
plt.show()
from scipy import stats
data_db=db[(np.abs(stats.zscore(db))<3).all(axis=1)]
```



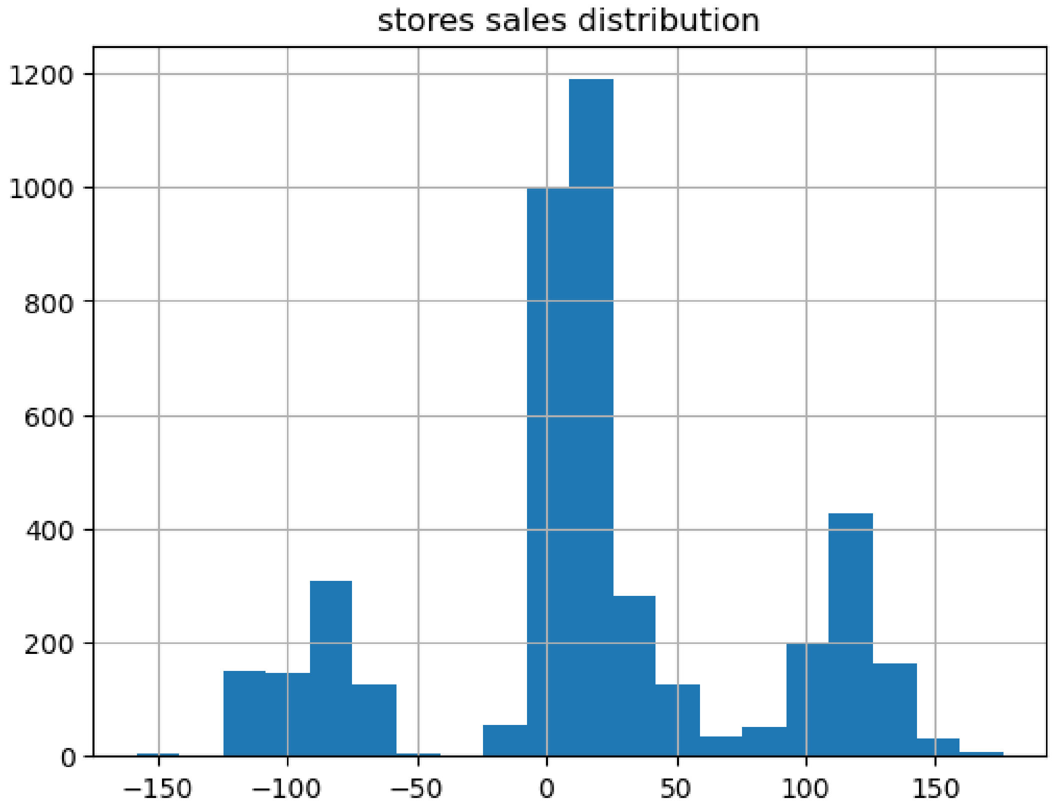
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[21], line 9
      7 plt.show()
      8 from scipy import stats
----> 9 data_db=db[(np.abs(stats.zscore(db))<3).all(axis=1)]

File ~\anaconda3\lib\site-packages\scipy\stats\_stats_py.py:2713, in zscore(a,␣
 ↪axis, ddof, nan_policy)
   2644 def zscore(a, axis=0, ddof=0, nan_policy='propagate'):
   2645     """
   2646     Compute the z score.
   2647
   (…)
   2711            [-0.91611681, -0.89090508,  1.4983032 ,  0.88731639, -0.
 ↪5785977 ]])
   2712     """
-> 2713     return zmap(a, a, axis=axis, ddof=ddof, nan_policy=nan_policy)
```

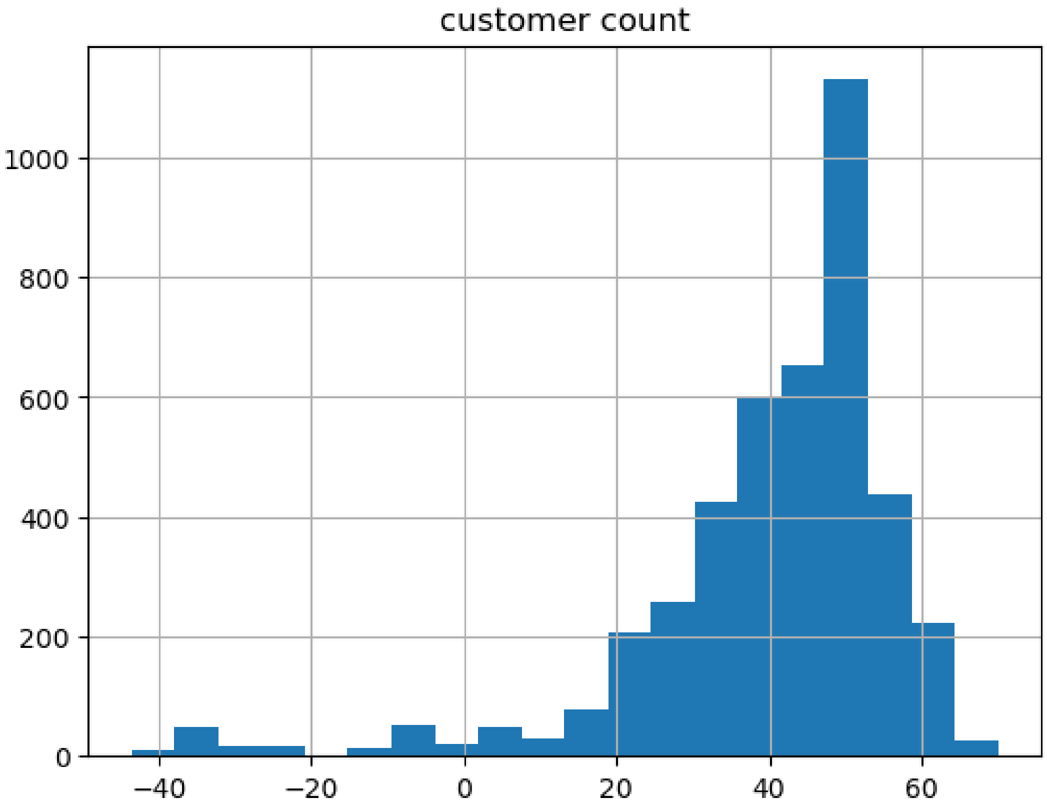
```
File ~\anaconda3\lib\site-packages\scipy\stats\_stats_py.py:2872, in␣
  ↪zmap(scores, compare, axis, ddof, nan_policy)
   2870         isconst = np.apply_along_axis(_isconst, axis, a)
   2871 else:
-> 2872     mn = a.mean(axis=axis, keepdims=True)
   2873     std = a.std(axis=axis, ddof=ddof, keepdims=True)
   2874     if axis is None:

File ~\anaconda3\lib\site-packages\numpy\core\_methods.py:182, in _mean(a, axis ␣
  ↪dtype, out, keepdims, where)
    180 ret = umr_sum(arr, axis, dtype, out, keepdims, where=where)
    181 if isinstance(ret, mu.ndarray):
--> 182     ret = um.true_divide(
    183             ret, rcount, out=ret, casting='unsafe', subok=False)
    184     if is_float16_result and out is None:
    185         ret = arr.dtype.type(ret)

TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

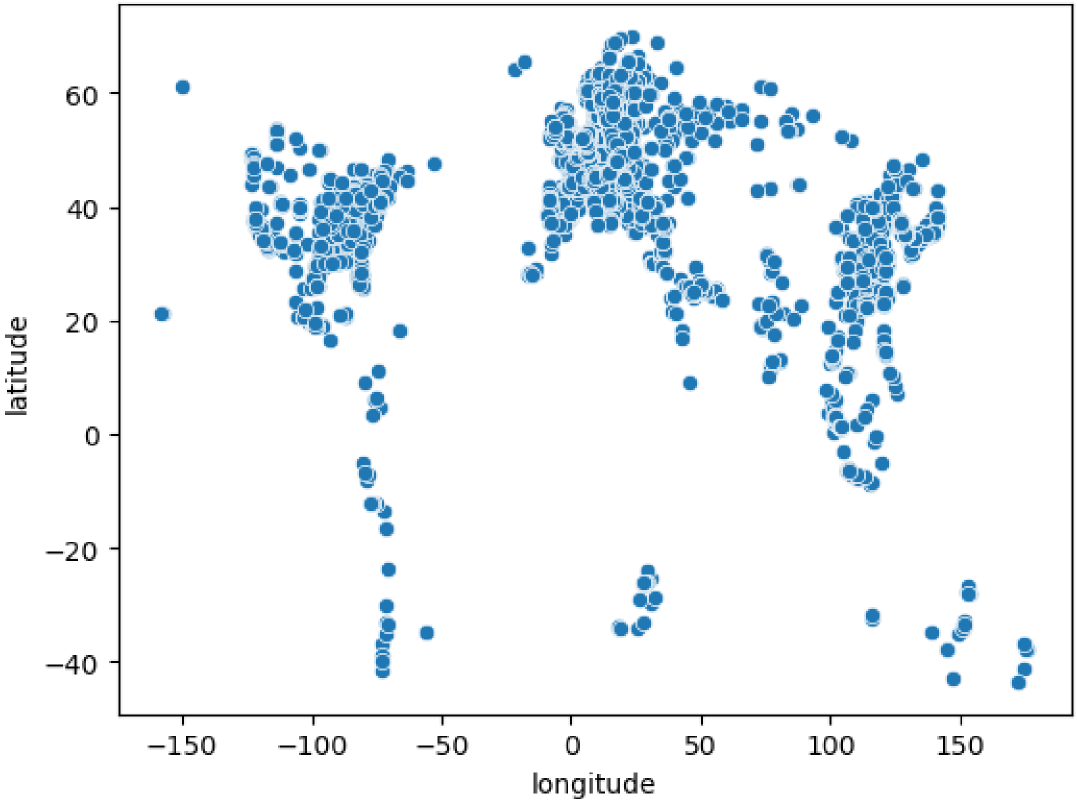
[2]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
data=pd.read_csv("C:/Users/DELL/Downloads/HM_all_stores.csv")
print(data.describe())
data['longitude'].hist(bins=20)
plt.title('stores sales distribution')
plt.show()
data['latitude'].hist(bins=20)
plt.title('customer count')
plt.show()
sns.scatterplot(x='longitude',y='latitude',data=data)
plt.show()
sns.heatmap(data.corr(),annot=True,cmap='coolwarm')
plt.title('correlation Matrix')
plt.show()
```

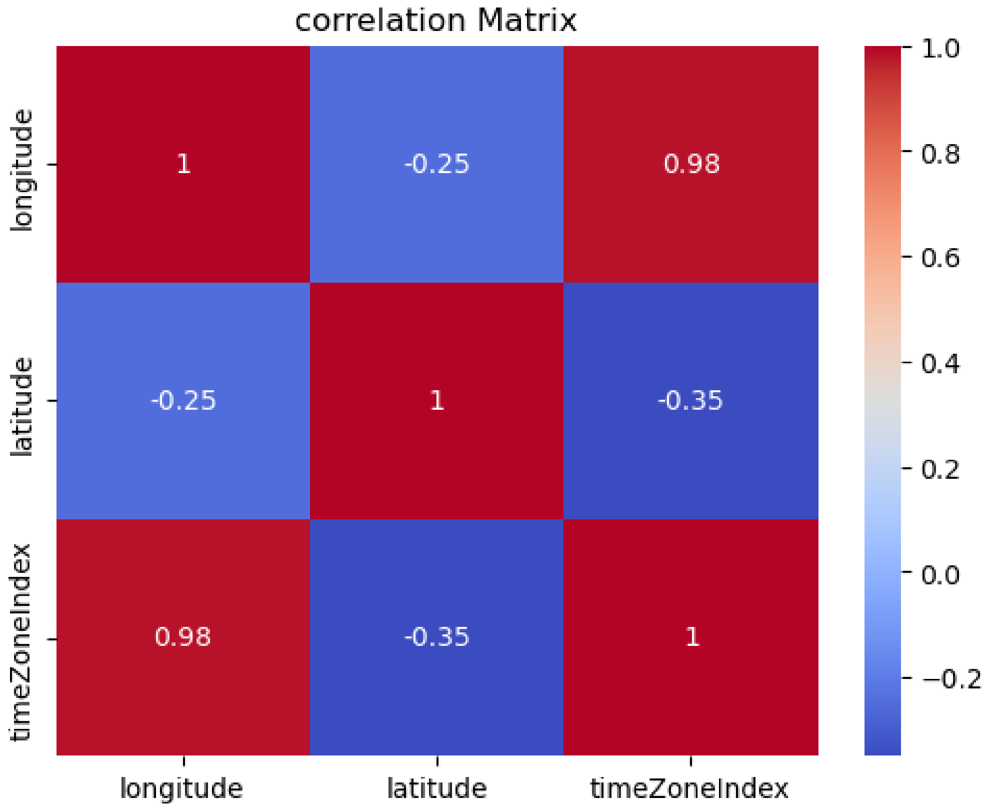
```
         longitude     latitude  timeZoneIndex
count  4292.000000  4292.000000    4289.000000
mean     17.298810    40.167707     115.934017
std      65.487986    17.449655      61.721152
min    -158.051943   -43.532996       2.000000
25%      -0.362545    34.180576     100.000000
50%      12.198888    44.803550     100.000000
75%      37.964040    51.348067     145.000000
max     176.104461    69.965270     290.000000
```


stores sales distribution
1200
1000
800
600
400
200
0
−150
−100
−50
0
50
100
150


customer count
1000
800
600
400
200
0
−40
−20
0
20
40
60



```
C:\Users\DELL\AppData\Local\Temp\ipykernel_2248\916600988.py:14: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future
version, it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  sns.heatmap(data.corr(),annot=True,cmap='coolwarm')
```



```
[10]: import numpy as np
      import pandas as pd
      df=pd.read_csv('C:/Users/krith/Downloads/Social_Network_Ads -␣
       ↪Social_Network_Ads.csv')
      df
```

```
[10]:       User ID  Gender  Age  EstimatedSalary  Purchased
      0    15624510    Male   19            19000          0
      1    15810944    Male   35            20000          0
      2    15668575  Female   26            43000          0
      3    15603246  Female   27            57000          0
      4    15804002    Male   19            76000          0
      ..        …       …   …              …          …
      395  15691863  Female   46            41000          1
      396  15706071    Male   51            23000          1
      397  15654296  Female   50            20000          1
      398  15755018    Male   36            33000          0
      399  15594041  Female   49            36000          1

      [400 rows x 5 columns]
```

```
[11]: df.head()
```

```
[11]:     User ID  Gender  Age  EstimatedSalary  Purchased
      0  15624510    Male   19            19000          0
      1  15810944    Male   35            20000          0
      2  15668575  Female   26            43000          0
      3  15603246  Female   27            57000          0
      4  15804002    Male   19            76000          0
```

```
[12]: features=df.iloc[:,[2,3]].values
      label=df.iloc[:,4].values
      features
```

```
[12]: array([[    19,  19000],
              [    35,  20000],
              [    26,  43000],
              [    27,  57000],
              [    19,  76000],
              [    27,  58000],
              [    27,  84000],
              [    32, 150000],
              [    25,  33000],
              [    35,  65000],
              [    26,  80000],
              [    26,  52000],
              [    20,  86000],
              [    32,  18000],
              [    18,  82000],
              [    29,  80000],
              [    47,  25000],
              [    45,  26000],
              [    46,  28000],
              [    48,  29000],
              [    45,  22000],
              [    47,  49000],
              [    48,  41000],
              [    45,  22000],
              [    46,  23000],
              [    47,  20000],
              [    49,  28000],
              [    47,  30000],
              [    29,  43000],
              [    31,  18000],
              [    31,  74000],
              [    27, 137000],
              [    21,  16000],
              [    28,  44000],
              [    27,  90000],
              [    35,  27000],
              [    33,  28000],
              [    30,  49000],
              [    26,  72000],
              [    27,  31000],
              [    27,  17000],
              [    33,  51000],
              [    35, 108000],
              [    30,  15000],
              [    28,  84000],
              [    23,  20000],
              [    25,  79000],
              [    27,  54000],
              [    30, 135000],
              [    31,  89000],
              [    24,  32000],
              [    18,  44000],
              [    29,  83000],
              [    35,  23000],
              [    27,  58000],
              [    24,  55000],
              [    23,  48000],
              [    28,  79000],
              [    22,  18000],
              [    32, 117000],
              [    27,  20000],
              [    25,  87000],
              [    23,  66000],
              [    32, 120000],
              [    59,  83000],
              [    24,  58000],
              [    24,  19000],
              [    23,  82000],
              [    22,  63000],
              [    31,  68000],
              [    25,  80000],
              [    24,  27000],
              [    20,  23000],
              [    33, 113000],
              [    32,  18000],
              [    34, 112000],
              [    18,  52000],
              [    22,  27000],
              [    28,  87000],
              [    26,  17000],
              [    30,  80000],
              [    39,  42000],
              [    20,  49000],
              [    35,  88000],
              [    30,  62000],
              [    31, 118000],
              [    24,  55000],
              [    28,  85000],
              [    26,  81000],
              [    35,  50000],
              [    22,  81000],
              [    30, 116000],
              [    26,  15000],
              [    29,  28000],
              [    29,  83000],
              [    35,  44000],
              [    35,  25000],
              [    28, 123000],
              [    35,  73000],
              [    28,  37000],
              [    27,  88000],
              [    28,  59000],
              [    32,  86000],
              [    33, 149000],
              [    19,  21000],
              [    21,  72000],
              [    26,  35000],
              [    27,  89000],
              [    26,  86000],
              [    38,  80000],
              [    39,  71000],
              [    37,  71000],
              [    38,  61000],
              [    37,  55000],
              [    42,  80000],
              [    40,  57000],
              [    35,  75000],
              [    36,  52000],
              [    40,  59000],
              [    41,  59000],
              [    36,  75000],
              [    37,  72000],
              [    40,  75000],
              [    35,  53000],
              [    41,  51000],
              [    39,  61000],
              [    42,  65000],
              [    26,  32000],
              [    30,  17000],
              [    26,  84000],
              [    31,  58000],
              [    33,  31000],
              [    30,  87000],
              [    21,  68000],
              [    28,  55000],
              [    23,  63000],
              [    20,  82000],
              [    30, 107000],
              [    28,  59000],
              [    19,  25000],
              [    19,  85000],
              [    18,  68000],
              [    35,  59000],
              [    30,  89000],
              [    34,  25000],
              [    24,  89000],
              [    27,  96000],
              [    41,  30000],
              [    29,  61000],
              [    20,  74000],
              [    26,  15000],
              [    41,  45000],
              [    31,  76000],
              [    36,  50000],
              [    40,  47000],
              [    31,  15000],
              [    46,  59000],
              [    29,  75000],
              [    26,  30000],
              [    32, 135000],
              [    32, 100000],
              [    25,  90000],
              [    37,  33000],
              [    35,  38000],
              [    33,  69000],
              [    18,  86000],
              [    22,  55000],
              [    35,  71000],
              [    29, 148000],
              [    29,  47000],
              [    21,  88000],
              [    34, 115000],
              [    26, 118000],
              [    34,  43000],
              [    34,  72000],
              [   23,  28000],
              [   35,  47000],
              [   25,  22000],
              [   24,  23000],
              [   31,  34000],
              [   26,  16000],
              [   31,  71000],
              [   32, 117000],
              [   33,  43000],
              [   33,  60000],
              [   31,  66000],
              [   20,  82000],
              [   33,  41000],
              [   35,  72000],
              [   28,  32000],
              [   24,  84000],
              [   19,  26000],
              [   29,  43000],
              [   19,  70000],
              [   28,  89000],
              [   34,  43000],
              [   30,  79000],
              [   20,  36000],
              [   26,  80000],
              [   35,  22000],
              [   35,  39000],
              [   49,  74000],
              [   39, 134000],
              [   41,  71000],
              [   58, 101000],
              [   47,  47000],
              [   55, 130000],
              [   52, 114000],
              [   40, 142000],
              [   46,  22000],
              [   48,  96000],
              [   52, 150000],
              [   59,  42000],
              [   35,  58000],
              [   47,  43000],
              [   60, 108000],
              [   49,  65000],
              [   40,  78000],
              [   46,  96000],
              [   59, 143000],
              [   41,  80000],
              [   35,  91000],
              [    37, 144000],
              [    60, 102000],
              [    35,  60000],
              [    37,  53000],
              [    36, 126000],
              [    56, 133000],
              [    40,  72000],
              [    42,  80000],
              [    35, 147000],
              [    39,  42000],
              [    40, 107000],
              [    49,  86000],
              [    38, 112000],
              [    46,  79000],
              [    40,  57000],
              [    37,  80000],
              [    46,  82000],
              [    53, 143000],
              [    42, 149000],
              [    38,  59000],
              [    50,  88000],
              [    56, 104000],
              [    41,  72000],
              [    51, 146000],
              [    35,  50000],
              [    57, 122000],
              [    41,  52000],
              [    35,  97000],
              [    44,  39000],
              [    37,  52000],
              [    48, 134000],
              [    37, 146000],
              [    50,  44000],
              [    52,  90000],
              [    41,  72000],
              [    40,  57000],
              [    58,  95000],
              [    45, 131000],
              [    35,  77000],
              [    36, 144000],
              [    55, 125000],
              [    35,  72000],
              [    48,  90000],
              [    42, 108000],
              [    40,  75000],
              [    37,  74000],
              [    47, 144000],
              [    40,  61000],
              [    43, 133000],
              [    59,  76000],
              [    60,  42000],
              [    39, 106000],
              [    57,  26000],
              [    57,  74000],
              [    38,  71000],
              [    49,  88000],
              [    52,  38000],
              [    50,  36000],
              [    59,  88000],
              [    35,  61000],
              [    37,  70000],
              [    52,  21000],
              [    48, 141000],
              [    37,  93000],
              [    37,  62000],
              [    48, 138000],
              [    41,  79000],
              [    37,  78000],
              [    39, 134000],
              [    49,  89000],
              [    55,  39000],
              [    37,  77000],
              [    35,  57000],
              [    36,  63000],
              [    42,  73000],
              [    43, 112000],
              [    45,  79000],
              [    46, 117000],
              [    58,  38000],
              [    48,  74000],
              [    37, 137000],
              [    37,  79000],
              [    40,  60000],
              [    42,  54000],
              [    51, 134000],
              [    47, 113000],
              [    36, 125000],
              [    38,  50000],
              [    42,  70000],
              [    39,  96000],
              [    38,  50000],
              [    49, 141000],
              [    39,  79000],
              [    39,  75000],
              [    54, 104000],
              [    35,  55000],
              [    45,  32000],
              [    36,  60000],
              [    52, 138000],
              [    53,  82000],
              [    41,  52000],
              [    48,  30000],
              [    48, 131000],
              [    41,  60000],
              [    41,  72000],
              [    42,  75000],
              [    36, 118000],
              [    47, 107000],
              [    38,  51000],
              [    48, 119000],
              [    42,  65000],
              [    40,  65000],
              [    57,  60000],
              [    36,  54000],
              [    58, 144000],
              [    35,  79000],
              [    38,  55000],
              [    39, 122000],
              [    53, 104000],
              [    35,  75000],
              [    38,  65000],
              [    47,  51000],
              [    47, 105000],
              [    41,  63000],
              [    53,  72000],
              [    54, 108000],
              [    39,  77000],
              [    38,  61000],
              [    38, 113000],
              [    37,  75000],
              [    42,  90000],
              [    37,  57000],
              [    36,  99000],
              [    60,  34000],
              [    54,  70000],
              [    41,  72000],
              [    40,  71000],
              [    42,  54000],
              [    43, 129000],
              [    53,  34000],
              [    47,  50000],
              [    42,  79000],
              [    42, 104000],
              [    59,  29000],
              [    58,  47000],
              [    46,  88000],
              [    38,  71000],
              [    54,  26000],
              [    60,  46000],
              [    60,  83000],
              [    39,  73000],
              [    59, 130000],
              [    37,  80000],
              [    46,  32000],
              [    46,  74000],
              [    42,  53000],
              [    41,  87000],
              [    58,  23000],
              [    42,  64000],
              [    48,  33000],
              [    44, 139000],
              [    49,  28000],
              [    57,  33000],
              [    56,  60000],
              [    49,  39000],
              [    39,  71000],
              [    47,  34000],
              [    48,  35000],
              [    48,  33000],
              [    47,  23000],
              [    45,  45000],
              [    60,  42000],
              [    39,  59000],
              [    46,  41000],
              [    51,  23000],
              [    50,  20000],
              [    36,  33000],
              [    49,  36000]])
```

```
[13]: label
```

```
[13]: array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
             0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0,
             1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0,
             1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
             0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
             1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1,
             0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0,
             1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1,
             0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1,
             1, 1, 0, 1])
```

[1]:

```python
import numpy as np
import scipy.stats as stats
growth_A = np.array([10.2, 9.8, 10.6, 10.1, 9.8, 10.3, 10.8, 9.9, 10.0, 9.6,
                    10.4, 10.7, 9.8, 9.5, 10.3, 9.9, 10.6, 10.2, 10.0, 9.7,
                    10.1, 9.8, 9.9, 10.4, 9.6, 10.5])

growth_B = np.array([12.2, 12.5, 12.0, 12.4, 12.4, 12.3, 12.6, 12.0, 12.2, 12.1,
                    12.3, 12.5, 12.4, 12.2, 12.9, 12.1, 12.2, 12.3, 12.5, 12.4,
                    12.1, 12.9, 12.2, 12.5, 12.4, 12.2])

growth_C = np.array([14.2, 14.7, 14.1, 14.4, 14.3, 14.7, 14.2, 14.0, 14.3, 14.5,
                    14.1, 14.2, 14.4, 14.6, 14.8, 14.3, 14.4, 14.2, 14.5, 14.4,
                    14.3, 14.6, 14.2, 14.1, 14.7, 14.3])
data=[growth_A,growth_B,growth_C]
f_stat, p_value = stats.f_oneway(*data)
print(f"F-statistic: {f_stat}")
print(f"P-value: {p_value}")
alpha=0.05
if(p_value<alpha):
    print("There is a significance difference between the mean growth of the␣
 ↪plants")
else:
    print("There is no significance difference between the mean growth of the␣
 ↪plants")
```

```
F-statistic: 1525.3644939965586
P-value: 1.7952414278872804e-61
There is a significance difference between the mean growth of the plants
```

[ ]:

```

```