```
from google.colab import drive
drive.mount('/content/drive')

     Mounted at /content/drive
```

```
import pandas as pd
import numpy as np
```

```
# q1
arr=np.array([1,2,3,4,5])
print("minimum:",np.min(arr))
print("max:",np.max(arr))
print("sum:",np.sum(arr))
print("mean:",np.mean(arr))
print("std:",np.std(arr))

     minimum: 1
     max: 5
     sum: 15
     mean: 3.0
     std: 1.4142135623730951
```

Disk: 26.21 GB/107.72 GB

```
# q2
health_data=np.array([[160,70,30],
                      [165,65,35],
                      [175,75,40]])
#normalization
std_deviation=np.std(health_data,axis=1)
mean=np.mean(health_data,axis=1)
normalized=health_data-mean/std_deviation
print(std_deviation)
print(mean)
print("normalized:",normalized)


     [54.36502143 55.57777334 57.20334101]
     [86.66666667 88.33333333 96.66666667]
     normalized: [[158.40583772  68.41063562  28.31012201]
      [163.40583772  63.41063562  33.31012201]
      [173.40583772  73.41063562  38.31012201]]
```

```
student_data=np.array([[80,50,60,-1,20],
              [12,34,56,40,50],
              [78,67,-1,89,67]],dtype=float)
```

```
# q3
student_data1=np.array([[80,50,60,-1,20],
              [12,34,56,40,50],
              [78,67,-1,89,67]],dtype=float)
last_three1=student_data1[:,-3:]

nan_data1=np.where(last_three1!=-1,last_three1,np.nan)
print(np.nanmean(nan_data1,axis=1))
```

```
[40.         48.66666667 78.        ]
```

```
# q4
print(np.linspace(15,25,24))
```

```
[15.         15.43478261 15.86956522 16.30434783 16.73913043 17.17391304
 17.60869565 18.04347826 18.47826087 18.91304348 19.34782609 19.7826087
 20.2173913  20.65217391 21.08695652 21.52173913 21.95652174 22.39130435
 22.82608696 23.26086957 23.69565217 24.13043478 24.56521739 25.        ]
```

```
# q5
daily_closing_price=np.array([100,102,98,105,107,110,108,112,115,118,120])
my_dict={"data":daily_closing_price}
df=pd.DataFrame(my_dict)
df['rolling']=df['data'].rolling(window=5).mean()
print(df)
```

Disk: 26.21 GB/107.72 GB

```
    data  rolling
0    100      NaN
1    102      NaN
2     98      NaN
3    105      NaN
4    107    102.4
5    110    104.4
6    108    105.6
7    112    108.4
8    115    110.4
9    118    112.6
10   120    114.6
```

```
#q6
random_data=np.random.rand(100,2)
mean_vector=[0,0]
covariance_matrix=[[1,0.5],[0.5,2]]
print(random_data)
```

```
 [0.80987631 0.63759559]
 [0.49409889 0.41283576]
 [0.66026183 0.41121884]
 [0.28486119 0.70176316]
 [0.0031311  0.93318589]
 [0.7899246  0.3646131 ]
 [0.87065712 0.00110868]
 [0.04182206 0.16409703]
 [0.90925129 0.75758511]
 [0.12062814 0.37258731]
 [0.83065126 0.54691929]
 [0.48670202 0.89877236]
 [0.35219231 0.76848497]
 [0.94511335 0.79920845]
 [0.36802602 0.20632562]
 [0.84114294 0.66266356]
 [0.09499232 0.2008893 ]
 [0.39415933 0.08079925]
 [0.44805989 0.21456707]
 [0.8537898  0.58229292]
 [0.68262468 0.31821035]
 [0.13530544 0.77849893]
 [0.65138343 0.23690566]
 [0.96136893 0.8549535 ]
 [0.22485848 0.80526596]
 [0.06057603 0.36962867]
 [0.10051986 0.77856033]
 [0.55606407 0.33000731]
 [0.12670374 0.63030098]
 [0.19553557 0.
 [0.5990988  0.
 [0.54128275 0.40526175]
 [0.6352323  0.46591434]
 [0.34243164 0.82888049]
 [0.79508799 0.00176849]
 [0.54756418 0.65015247]
 [0.40006162 0.34000477]
 [0.51966069 0.8733626 ]
 [0.71318913 0.37821415]
 [0.48471814 0.9642781 ]
 [0.52059342 0.78454405]
 [0.71647272 0.46322238]
 [0.30202115 0.97388826]
 [0.35695171 0.56385331]
 [0.20152789 0.05605411]]
```

Disk: 26.21 GB/107.72 GB

```
#q7
properties_matrix=np.array([[1,2,3],
                            [4,5,6],
                             [7,8,9]
])
print(np.linalg.det(properties_matrix))
```

```
0.0
```

```
# q8
properties_matrix=np.array([[1,2,3],
                            [4,5,6],
                             [7,8,9]
])
boolean_matrix=np.array([
    [False,False,False],
    [False,True,True],
    [True,True,True]
    ])
print(properties_matrix[boolean_matrix])
```

```
[5 6 7 8 9]
```

```
#q9
data = {'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve', 'Frank', 'Grace'],
        'Age': [25, 30, 35, 40, 45, 50, 55],
        'City': ['New York', 'Los Angeles', 'Chicago', 'Houston', 'Phoenix', 'Miami', 'Boston'],
        'Department': ['HR', 'IT', 'Finance', 'Marketing', 'Sales', 'IT', 'HR']}

df=pd.DataFrame(data)
print(df[(df['Age']<45) &(df['Department']!='HR')][['Name','City']])
```

```
      Name         City
1      Bob  Los Angeles
2  Charlie      Chicago
3    David
```

Disk: 26.21 GB/107.72 GB

```
# q10
data = {'Department': ['Electronics', 'Electronics', 'Clothing', 'Clothing', 'Home Goods'],
        'Salesperson': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],
        'Sales': [70000, 50000, 30000, 40000, 60000]}
df=pd.DataFrame(data)
df2=pd.DataFrame(df.groupby(['Salesperson','Department'])['Sales'].aggregate(np.mean))
part_a=df2.sort_values(by='Sales',ascending=False)
df1=pd.DataFrame(df.groupby('Department')['Sales'].aggregate(np.mean))
part_b=df1.sort_values(by='Sales',ascending=False)
print("average sales per person",part_a)
print("---------")
print("rank by department on average sales",part_b)
```

```
average sales per person                            Sales
Salesperson Department
Alice       Electronics  70000.0
Eve         Home Goods   60000.0
Bob         Electronics  50000.0
David       Clothing     40000.0
Charlie     Clothing     30000.0
---------
rank by department on average sales                Sales
Department
Electronics  60000.0
Home Goods   60000.0
Clothing     35000.0
```

```
# q11
data = {
    'Product': ['Apples', 'Bananas', 'Cherries', 'Dates', 'Elderberries', 'Flour', 'Grapes'],
    'Category': ['Fruit', 'Fruit', 'Fruit', 'Fruit', 'Fruit', 'Bakery', 'Fruit'],
    'Price': [1.20, 0.50, 3.00, 2.50, 4.00, 1.50, 2.00],
    'Promotion': [True, False, True, True, False, True, False]
}
df=pd.DataFrame(data)
print(df[(df['Price']>df['Price'].mean()) & (df['Category']=='Fruit') &(df['Promotion']==False)])
```

```
        Product Category  Price  Promotion
4  Elderberries    Fruit    4.0      False
```

```
#q12
employee_data = {
    'Employee': ['Alice', 'Bob', 'Charlie', 'David'],
    'Department': ['HR', 'IT', 'Finance', 'IT'],
    'Manager': ['John', 'Rachel', 'Emily', 'Rachel']
}

# Dataset of employee project assignments
project_data = {
    'Employee': ['Alice', 'Charlie', 'Eve'],
    'Project': ['P1', 'P3', 'P2']
}
df1=pd.DataFrame(employee_data)
df2=pd.DataFrame(pro Disk: 26.21 GB/107.72 GB
print(pd.merge(df1,df2,on='Employee',how='left'))
```

```
  Employee Department Manager Project
0    Alice         HR    John      P1
1      Bob         IT  Rachel     NaN
2  Charlie    Finance   Emily      P3
3    David         IT  Rachel     NaN
```

```
# q13
df=pd.read_csv(r'/content/drive/MyDrive/dataset/Q13_sports_team_stats.csv')
df['win_Ratios']=df['Wins']/np.sum(df['Wins'])
df['average_scores']=(df['Wins']*3)/(df['GamesPlayed']*3)
df['strengths']=df['GamesPlayed']-df['Wins']
print(df['win_Ratios'])
print(df['average_scores'])
print(df['strengths'])
```

```
0    0.106061
1    0.090909
2    0.121212
3    0.075758
4    0.136364
5    0.090909
6    0.106061
7    0.060606
8    0.136364
9    0.075758
Name: win_Ratios, dtype: float64
0    0.7
1    0.6
2    0.8
3    0.5
4    0.9
```

```
5    0.6
6    0.7
7    0.4
8    0.9
9    0.5
Name: average_scores, dtype: float64
0    3
1    4
2    2
3    5
4    1
5    4
6    3
7    6
8    1
9    5
Name: strengths, dtype: int64
```

```
# q14
df=pd.read_csv(r'/content/drive/MyDrive/dataset/Q14_customer_purchases.csv')
k=df.groupby('CustomerID')['PurchaseAmount']
df1=pd.DataFrame(k)
print(df1)
```

```
    0                                            1
0   1  0    200
Name: PurchaseAmount, dtype: int64
1   2  1    150
Name: PurchaseA  Disk: 26.21 GB/107.72 GB
2   3  2    300
Name: PurchaseAmount, dtype: int64
3   4  3    250
Name: PurchaseAmount, dtype: int64
4   5  4    500
Name: PurchaseAmount, dtype: int64
5   6  5    350
Name: PurchaseAmount, dtype: int64
6   7  6    400
Name: PurchaseAmount, dtype: int64
7   8  7    450
Name: PurchaseAmount, dtype: int64
8   9  8    550
Name: PurchaseAmount, dtype: int64
9  10  9    600
Name: PurchaseAmount, dtype: int64
```

```
df.head()
```

	CustomerID	Date	PurchaseAmount	LoyaltyProgramSignUp
0	1	2021-01-31	200	2021-03-01
1	2	2021-02-28	150	2021-03-01
2	3	2021-03-31	300	2021-03-01
3	4	2021-04-30	250	2021-03-01
4	5	2021-05-31	500	2021-03-01

Next steps: View recommended plots

```
df.groupby('CustomerID')['PurchaseAmount']
```

```
# q15
df=pd.read_csv(r'/content/drive/MyDrive/dataset/Q15_student_grades.csv')
k=df.groupby('Subject')['Grade'].aggregate(np.mean)
df1=pd.DataFrame(k)
df1.sort_values(by='Grade',ascending=False)
```

```
              Grade
Subject
Science   91.000000
Math      87.250000
History   83.666667
```

```
df.head()
```

```
   StudentID  Subject  Grade
0          1     Math     88
1          2  Disk: 26.21 GB/107.72 GB
2          3  History     85
3          4     Math     87
4          5  Science     90
```

Next steps: View recommended plots

```
k=df.groupby('Subject')['Grade'].aggregate(np.mean)
```

```
df1=pd.DataFrame(k)
```

```
df1.sort_values(by='Grade',ascending=False)
```

```
              Grade
Subject
Science   91.000000
Math      87.250000
History   83.666667
```