```
import pandas as pd
import numpy as np
df = pd.read_csv("/content/Mall_Customers.csv")
df
                                                                                1
                      Genre Age Annual Income (k$) Spending Score (1-100)
          CustomerID
      0
                               19
                                                   15
                                                                           39
                   1
                        Male
                   2
                        Male
                               21
                                                   15
                                                                           81
       2
                   3 Female
                               20
                                                   16
                                                                            6
       3
                   4 Female
                               23
                                                   16
                                                                           77
       4
                   5 Female
                               31
                                                   17
                                                                           40
      195
                  196 Female
                               35
                                                  120
                                                                           79
                  197 Female
                               45
                                                  126
                                                                           28
      196
      197
                  198
                        Male
                               32
                                                  126
                                                                           74
      198
                  199
                        Male
                               32
                                                  137
                                                                           18
     199
                 200
                              30
                                                                           83
                        Male
                                                  137
    200 rows × 5 columns
df.mean()
     <ipython-input-33-c61f0c8f89b5>:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a futur
       df.mean()
    CustomerID
                              100.50
    Age
                                38.85
    Annual Income (k$)
                                60.56
     Spending Score (1-100)
                                50.20
     dtype: float64
df.loc[:,'Age'].mean()
     38.85
df.mean(axis = 1)[0:4]
     <ipython-input-5-5a17804ef008>:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None'
       df.mean(axis = 1)[0:4]
         18.50
    A
    1
         29.75
    2
         11.25
     3
         30.00
    dtype: float64
df.median()
     <ipython-input-6-6d467abf240d>:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a futu
      df.median()
                               100.5
    CustomerID
                               36.0
    Age
    Annual Income (k$)
                                61.5
    Spending Score (1-100)
                                50.0
     dtype: float64
df.loc[:,'Age'].median()
     36.0
df.median(axis = 1)[0:4]
     <ipython-input-8-b75ab1b1d07a>:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None'
       df.median(axis = 1)[0:4]
    0
         17.0
    1
         18.0
    2
         11.0
         19.5
     dtype: float64
```

8.057088

32.300671 dtype: float64

3

CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object .loc[:,'Age'].max(skipna = False) 70 .std() <ipython-input-15-ce97bb7eaef8:1: (1-100)="" (k\$)="" .loc[:,'age'].std()="" 13.969007="" 13.96900733155888<="" 25.823522="" 26.264721="" 57.879185="" a="" age="" annual="" customerid="" dataframe.std="" default="" deprecated.="" df.std()="" dtype:="" float64="" futurewarning:="" in="" income="" is="" numeric_only="" of="" score="" spending="" th="" the="" value=""><th>1 2 NaN NaN 78.0 NaN NaN 18.0 NaN 18.0</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></ipython-input-15-ce97bb7eaef8:1:>	1 2 NaN NaN 78.0 NaN NaN 18.0							
2 3 NoN NoN NoN NoN NoN NoN NoN NoN NoN N	2 3 NaN NaN NaN NaN NaN NaN NaN NaN NaN N	0	1 Fema	ale 32.0	54.0	42.0		
3	3	1	2 Na	aN NaN	78.0	NaN		
4 5 NaN NaN NaN NaN NaN NaN NaN NaN NaN N	4 5 NaN NaN NaN NaN NaN NaN NaN NaN NaN N	2	3 Na	aN NaN	NaN	NaN		
195	195	3	4 Na	aN NaN	NaN	NaN		
195	195	4	5 N	aN NaN	NaN	NaN		
196	196							
197	197	195	196 Na	aN NaN	NaN	NaN		
198	198	196	197 Na	aN NaN	NaN	NaN		
198	198	197	198 Na	aN NaN	NaN	NaN		
199	199	198			NaN	NaN		
200 rows × 5 columns .loc[:, 'Age'].mode() 8	200 rows × 5 columns loc[;,'Age'].mode() 8							
loc[;,'Age'].mode() 8 32 Name: Age, dtype: int64	loc[:,'Age'].mode() 8 32							
0 32 Name: Age, dtype: int64 min() CustomerID 1 Genre Female Age 18 Annual Income (k\$) 15 Spending Score (1-100) 1 dtype: object loc[:,'Age'].min(skipna = False) 18 max() CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object loc[:,'Age'].max(skipna = False) 70 std() cipython-input-15-ce97bb7eaef83:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() customerID 57.879185 Age 13.9690087 Annual Income (k\$) 25.823522 dtype: float64 loc[:,'Age'].std() 13.96900733155888	0 32 Name: Age, dtype: int64 min() CustomerID							
Name: Age, dtype: int64 .min() CustomerID	Mane: Age, dtype: int64 min() CustomerID	.loc[:,'Age']	.mode()					
Name: Age, dtype: int64 .min() CustomerID	Mane: Age, dtype: int64 min() CustomerID	0 32						
CustomerID 1 Genre Female Age 18 Annual Income (k\$) 15 Spending Score (1-180) 1 Cloc[:,'Age'].min(skipna = False) 18 .max() CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-180) 99 dtype: object .loc[:,'Age'].max(skipna = False) 70 .std() <ipre> <ipre> <ipre> <ipre> <pre></pre></ipre></ipre></ipre></ipre>	CustomerID 1 Genre Female Age 18 Annual Income (k\$) 15 Spending Score (1-100) 1 dtype: object 18 max() CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object 10c[:,'Age'].max(skipna = False) 18 max() CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object 10c[:,'Age'].max(skipna = False) 70 std() cipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID 57.879185 Age 13.969007 Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 Clc[:,'Age'].std() 13.96900733155888		dtype: inte	54				
CustomerID 1 Genre Female Age 18 Annual Income (k\$) 15 Spending Score (1-180) 1 dtype: object CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-180) 99 dtype: object cloc[:,'Age'].max(skipna = False) 18 max() CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-180) 99 dtype: object .loc[:,'Age'].max(skipna = False) 70 .std() cipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID 57.879185 Age 13.969007 Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64	CustomerID 1 Genre Female Age 18 Annual Income (k\$) 15 Spending Score (1-100) 1 dtype: object 18 max() CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object 10c[:,'Age'].max(skipna = False) 18 max() CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object 10c[:,'Age'].max(skipna = False) 70 std() cipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID 57.879185 Age 13.969007 Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 Clc[:,'Age'].std() 13.96900733155888							
Genre	Genre Female Age 18 Annual Income (k\$) 15 Spending Score (1-100) 1 dtype: object loc[:,'Age'].min(skipna = False) 18 max() LustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object loc[:,'Age'].max(skipna = False) 70 std() <pre></pre>	.min()						
Genre	Genre Female Age 18 Annual Income (k\$) 15 Spending Score (1-100) 1 dtype: object loc[:,'Age'].min(skipna = False) 18 max() LustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object loc[:,'Age'].max(skipna = False) 70 std() <pre></pre>	CustomerID	,	1				
Annual Income (k\$) 15 Spending Score (1-100) 1 CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object 70 .loc[:,'Age'].max(skipna = False) 70 .std() <ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID 99 std() <ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID 57.879185 Age 13.969007 Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 .loc[:,'Age'].std() 13.96900733155888</ipython-input-15-ce97bb7eaef8></ipython-input-15-ce97bb7eaef8>	Annual Income (k\$) 15 Spending Score (1-100) 1 loc[:,'Age'].min(skipna = False) 18 max() CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object loc[:,'Age'].max(skipna = False) 70 std() <pre></pre>			Female				
<pre>dtype: object .loc[:,'Age'].min(skipna = False) 18 .max() CustomerID</pre>	dtype: object loc[:,'Age'].min(skipna = False) 18 max() CustomerID	Annual Inc		15				
18	loc[:,'Age'].min(skipna = False) 18 max() CustomerID) 1				
.max() CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object .loc[:,'Age'].max(skipna = False) 70 .std() <ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID 57.879185 Age 13.969007 Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 .loc[:,'Age'].std() 13.96900733155888</ipython-input-15-ce97bb7eaef8>	max() CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object loc[:,'Age'].max(skipna = False) 70 std() <pre></pre>	utype. obj	ect					
CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object Cloc[:,'Age'].max(skipna = False) 70 .std() <ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID 57.879185 Age 13.969007 Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 .loc[:,'Age'].std() 13.96900733155888</ipython-input-15-ce97bb7eaef8>	CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object loc[:,'Age'].max(skipna = False) 70 std() <ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID 57.879185 Age 13.969007 Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 </ipython-input-15-ce97bb7eaef8>	.loc[:,'Age']	.min(skipna	= False)				
CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object .loc[:,'Age'].max(skipna = False) 70 .std() <ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID 57.879185 Age 13.969007 Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 .loc[:,'Age'].std() 13.96900733155888</ipython-input-15-ce97bb7eaef8>	CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object loc[:,'Age'].max(skipna = False) 70 std() <pre></pre>	18						
CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object .loc[:,'Age'].max(skipna = False) 70 .std() <ipython-input-15-ce97bb7eaef8:1: (1-100)="" (k\$)="" .loc[:,'age'].std()="" 13.969007="" 13.96900733155888<="" 25.823522="" 26.264721="" 57.879185="" a="" age="" annual="" customerid="" dataframe.std="" default="" deprecated.="" df.std()="" dtype:="" float64="" futurewarning:="" in="" income="" is="" numeric_only="" of="" score="" spending="" td="" the="" value=""><td>CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object loc[:,'Age'].max(skipna = False) 70 std() <pre></pre></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></ipython-input-15-ce97bb7eaef8:1:>	CustomerID 200 Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object loc[:,'Age'].max(skipna = False) 70 std() <pre></pre>							
Genre	Genre Male Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object loc[:,'Age'].max(skipna = False) 70 std() <pre></pre>	.max()						
Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object .loc[:, 'Age'].max(skipna = False) 70 .std() <ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID 57.879185 Age 13.969007 Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 .loc[:, 'Age'].std() 13.96900733155888</ipython-input-15-ce97bb7eaef8>	Age 70 Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object 10c[:,'Age'].max(skipna = False) 70 std() <ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID 57.879185 Age 13.969007 Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 10c[:,'Age'].std() 13.96900733155888</ipython-input-15-ce97bb7eaef8>)					
Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object loc[:,'Age'].max(skipna = False) 70 .std() <pre></pre>	Annual Income (k\$) 137 Spending Score (1-100) 99 dtype: object loc[:,'Age'].max(skipna = False) 70 std() <pre></pre>							
<pre>dtype: object .loc[:,'Age'].max(skipna = False) 70 .std()</pre>	<pre>dtype: object loc[:,'Age'].max(skipna = False) 70 std() <ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID</ipython-input-15-ce97bb7eaef8></pre>	Annual Inc		137				
.std() <pre></pre>	std() <ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID</ipython-input-15-ce97bb7eaef8>) 99				
.std() <pre></pre>	std() <ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID</ipython-input-15-ce97bb7eaef8>							
<pre>.std() <ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID</ipython-input-15-ce97bb7eaef8></pre>	<pre>std()</pre>	.loc[:,'Age']	.max(skipna	= False)				
<pre><ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a</ipython-input-15-ce97bb7eaef8></pre>	<pre><ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID</ipython-input-15-ce97bb7eaef8></pre>	70						
<pre><ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a</ipython-input-15-ce97bb7eaef8></pre>	<pre><ipython-input-15-ce97bb7eaef8>:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a df.std() CustomerID</ipython-input-15-ce97bb7eaef8></pre>							
df.std() CustomerID 57.879185 Age 13.969007 Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 .loc[:,'Age'].std() 13.96900733155888	df.std() CustomerID 57.879185 Age 13.969007 Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 loc[:,'Age'].std() 13.96900733155888							
CustomerID 57.879185 Age 13.969007 Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 Lloc[:,'Age'].std() 13.96900733155888	CustomerID 57.879185 Age 13.969007 Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 loc[:,'Age'].std() 13.96900733155888			7bb7eaef8>:1: Fut	ureWarning: The defau	ilt value of numeric_only	in DataFrame.std is deprecated.	In a
Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 .loc[:,'Age'].std() 13.96900733155888	Annual Income (k\$) 26.264721 Spending Score (1-100) 25.823522 dtype: float64 loc[:,'Age'].std() 13.96900733155888	CustomerID						
Spending Score (1-100) 25.823522 dtype: float64 .loc[:,'Age'].std() 13.96900733155888	Spending Score (1-100) 25.823522 dtype: float64 loc[:,'Age'].std() 13.96900733155888		ome (k\$)					
.loc[:,'Age'].std() 13.96900733155888	loc[:,'Age'].std() 13.96900733155888	Spending S	core (1-100)					
13.96900733155888	13.96900733155888	d type. P10	-a c 0+					
13.96900733155888	13.96900733155888							
		.loc[:,'Age']	.std()					
	std(axis=1)[0:4]	13.9690073	3155888					
	std(axis=1)[0:4]							

<ipython-input-17-18b37211c5c6>:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None
 df.std(axis=1)[0:4]
0 15.695010
1 35.074920

CustomerID Genre Age Annual Income (k\$) Spending Score (1-100)

df.groupby(['Genre'])['Age'].mean()

Genre

Female 38.098214 Male 39.806818 Name: Age, dtype: float64

 $\begin{tabular}{ll} df_u=df.rename(columns= {'Annual Income (k$)':'Income'}, inplace=False) \\ df u \\ \end{tabular}$

	CustomerID	Genre	Age	Income	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

200 rows × 5 columns

df_u.groupby(['Genre'])['Income'].mean()

Genre

Female 59.250000 Male 62.227273

Name: Income, dtype: float64

from sklearn import preprocessing
enc = preprocessing.OneHotEncoder()
enc_df = pd.DataFrame(enc.fit_transform(df[['Genre']]).toarray())
enc_df

	0	1	7 .
0	0.0	1.0	
1	0.0	1.0	
2	1.0	0.0	
3	1.0	0.0	
4	1.0	0.0	
195	1.0	0.0	
196	1.0	0.0	
197	0.0	1.0	
198	0.0	1.0	
199	0.0	1.0	

200 rows × 2 columns

df_encode =df_u.join(enc_df)
df_encode

```
0
                         Male
                                 19
                                         15
                                                                  39 0.0 1.0
       1
                    2
                          Male
                                21
                                         15
                                                                  81 0.0 1.0
       2
                    3 Female
                                20
                                         16
                                                                   6
                                                                     1.0 0.0
                    4 Female
       3
                                 23
                                         16
                                                                  77
                                                                     1.0
                                                                          0.0
       4
                    5
                       Female
                                31
                                         17
                                                                  40
                                                                      1.0
                                                                          0.0
                  400 F----I-
      405
                                                                     40 00
iris = pd.read csv("/content/IRIS.csv")
col_names = ['Sepal_Length','Sepal_Width','Petal_Length','Petal_Width','Species']
      198
                  199
                          Male
                                32
                                        137
                                                                  18 0.0 1.0
iris
           sepal_length sepal_width petal_length petal_width
                                                                      species
       0
                     5.1
                                  3.5
                                                 1.4
                                                              0.2
                                                                    Iris-setosa
                                                 1.4
       1
                     4.9
                                  3.0
                                                              0.2
                                                                    Iris-setosa
       2
                     4.7
                                  3.2
                                                 1.3
                                                              0.2
                                                                    Iris-setosa
       3
                     4.6
                                  3.1
                                                 1.5
                                                              0.2
                                                                    Iris-setosa
       4
                     5.0
                                  3.6
                                                 1.4
                                                              0.2
                                                                    Iris-setosa
                                                  ...
                     6.7
                                  3.0
                                                 5.2
                                                              2.3 Iris-virginica
      145
      146
                     6.3
                                  2.5
                                                 5.0
                                                              1.9 Iris-virginica
      147
                     6.5
                                  3.0
                                                 5.2
                                                              2.0 Iris-virginica
      148
                     6.2
                                                              2.3 Iris-virginica
                                  3.4
                                                 5.4
      149
                     5.9
                                  3.0
                                                 5.1
                                                              1.8 Iris-virginica
     150 rows × 5 columns
irisSet = (iris['species'] == 'Iris-setosa')
print('Iris-setosa')
print(iris[irisSet].describe())
     Iris-setosa
            sepal_length sepal_width petal_length petal_width
                             50.000000
                                                           50.00000
                50.00000
                                            50.000000
     count
                 5.00600
                              3.418000
                                             1.464000
                                                            0.24400
     mean
                 0.35249
                              0.381024
                                             0.173511
                                                            0.10721
     std
     min
                 4.30000
                              2.300000
                                             1.000000
                                                            0.10000
     25%
                  4.80000
                              3.125000
                                             1.400000
                                                            0.20000
     50%
                  5.00000
                              3.400000
                                             1.500000
                                                            0.20000
     75%
                  5.20000
                              3.675000
                                             1.575000
                                                            0.30000
     max
                  5.80000
                              4.400000
                                             1.900000
                                                            0.60000
irisVer = (iris['species'] == 'Iris-versicolor')
print('Iris-versicolor')
print(iris[irisVer].describe())
     Iris-versicolor
            sepal_length sepal_width petal_length petal_width
     count
               50.000000
                             50.000000
                                            50.000000
                                                         50.000000
     mean
                5.936000
                              2.770000
                                             4.260000
                                                          1.326000
                0.516171
                              0.313798
                                             0.469911
                                                           0.197753
     std
                4.900000
                              2.000000
                                             3.000000
                                                           1.000000
     min
     25%
                5.600000
                              2.525000
                                             4.000000
                                                           1.200000
                                             4.350000
                                                           1.300000
     50%
                5.900000
                              2.800000
                              3.000000
                                             4.600000
     75%
                6.300000
                                                           1.500000
                7.000000
                                             5.100000
     max
                              3.400000
                                                           1.800000
irisVir = (iris['species'] == 'Iris-virginica')
print('Iris-virginica')
print(iris[irisVir].describe())
     Iris-virginica
            sepal_length sepal_width petal_length petal_width
     count
                50.00000
                             50.000000
                                            50.000000
                                                           50.00000
                                             5.552000
```

2.02600

2.974000

mean

1

Genre Age Income Spending Score (1-100) 0 1

CustomerID

	0 63500	0 222407	0 554005	0 07465
std	0.63588	0.322497	0.551895	0.27465
min	4.90000	2.200000	4.500000	1.40000
25%	6.22500	2.800000	5.100000	1.80000
50%	6.50000	3.000000	5.550000	2.00000
75%	6.90000	3.175000	5.875000	2.30000
max	7.90000	3.800000	6.900000	2.50000

iris[irisVir]

	sepal_length	sepal_width	petal_length	petal_width	species	1
100	6.3	3.3	6.0	2.5	Iris-virginica	
101	5.8	2.7	5.1	1.9	Iris-virginica	
102	7.1	3.0	5.9	2.1	Iris-virginica	
103	6.3	2.9	5.6	1.8	Iris-virginica	
104	6.5	3.0	5.8	2.2	Iris-virginica	
105	7.6	3.0	6.6	2.1	Iris-virginica	
106	4.9	2.5	4.5	1.7	Iris-virginica	
107	7,3	2,9	6.3	1,8	Iris-virginica	
108	6.7	2.5	5.8	1.8	Iris-virginica	
109	7.2	3.6	6.1	2.5	Iris-virginica	
110	6.5	3.2	5.1	2.0	Iris-virginica	
111	6.4	2.7	5.3	1.9	Iris-virginica	
112	6.8	3.0	5.5	2.1	Iris-virginica	
113	5.7	2.5	5.0	2.0	Iris-virginica	
114	5.8	2.8	5.1	2.4	Iris-virginica	
115	6.4	3.2	5.3	2.3	Iris-virginica	
116	6.5	3.0	5.5	1.8	Iris-virginica	
117	7.7	3.8	6.7	2.2	Iris-virginica	

✓ 0s completed at 12:34 AM