```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data = pd.read_excel('/content/drive/MyDrive/Classroom/Sample - Superstore.xls')
```

```
data
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Categor
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-BO-10001798	Furnitu
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-CH-10000454	Furnitu
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-LA-10000240	Offic Supplie
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	FUR-TA-10000577	Furnitu
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	OFF-ST-10000760	Offic Supplie
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9989	9990	CA-2014-110422	2014-01-21	2014-01-23	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	...	33180	South	FUR-FU-10001889	Furnitu
9990	9991	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	FUR-FU-10000747	Furnitu
9991	9992	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	TEC-PH-10003645	Technolo
9992	9993	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	OFF-PA-10004041	Offic Supplie
9993	9994	CA-2017-119914	2017-05-04	2017-05-09	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	...	92683	West	OFF-AP-10002684	Offic Supplie

9994 rows × 21 columns

```
data.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Category	Su Catego
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-BO-10001798	Furniture	Bookcas
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-CH-10000454	Furniture	Cha
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-LA-10000240	Office Supplies	Labe
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	FUR-TA-10000577	Furniture	Tabl
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	OFF-ST-10000760	Office Supplies	Stora

5 rows × 21 columns

```
data.tail()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Categor
9989	9990	CA-2014-110422	2014-01-21	2014-01-23	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	...	33180	South	FUR-FU-10001889	Furnitur
9990	9991	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	FUR-FU-10000747	Furnitur
9991	9992	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	TEC-PH-10003645	Technolog
9992	9993	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	OFF-PA-10004041	Offic Supplie
9993	9994	CA-2017-119914	2017-05-04	2017-05-09	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	...	92683	West	OFF-AP-10002684	Offic Supplie

5 rows × 21 columns

```
data.shape
```

```
(9994, 21)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
```

```
 1   Order ID       9994 non-null   object
 2   Order Date     9994 non-null   datetime64[ns]
 3   Ship Date      9994 non-null   datetime64[ns]
 4   Ship Mode      9994 non-null   object
 5   Customer ID    9994 non-null   object
 6   Customer Name  9994 non-null   object
 7   Segment        9994 non-null   object
 8   Country        9994 non-null   object
 9   City           9994 non-null   object
 10  State          9994 non-null   object
 11  Postal Code    9994 non-null   int64
 12  Region         9994 non-null   object
 13  Product ID     9994 non-null   object
 14  Category       9994 non-null   object
 15  Sub-Category   9994 non-null   object
 16  Product Name   9994 non-null   object
 17  Sales          9994 non-null   float64
 18  Quantity       9994 non-null   int64
 19  Discount       9994 non-null   float64
 20  Profit         9994 non-null   float64
dtypes: datetime64[ns](2), float64(3), int64(3), object(13)
memory usage: 1.6+ MB
```

```
data.isnull().sum()
```

	0
Row ID	0
Order ID	0
Order Date	0
Ship Date	0
Ship Mode	0
Customer ID	0
Customer Name	0
Segment	0
Country	0
City	0
State	0
Postal Code	0
Region	0
Product ID	0
Category	0
Sub-Category	0
Product Name	0
Sales	0
Quantity	0
Discount	0
Profit	0

dtype: int64

```
data.duplicated()
```

	0
0	False
1	False
2	False
3	False
4	False
...	...
9989	False
9990	False
9991	False
9992	False
9993	False

9994 rows × 1 columns

dtype: bool

Double-click (or enter) to edit

```
data.describe()
```

	Row ID	Order Date	Ship Date	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994	9994	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	4997.500000	2016-04-30 00:07:12.259355648	2016-05-03 23:06:58.571142912	55190.379428	229.858001	3.789574	0.156203	28.656896
min	1.000000	2014-01-03 00:00:00	2014-01-07 00:00:00	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	2499.250000	2015-05-23 00:00:00	2015-05-27 00:00:00	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	4997.500000	2016-06-26 00:00:00	2016-06-29 00:00:00	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	7495.750000	2017-05-14 00:00:00	2017-05-18 00:00:00	90008.000000	209.940000	5.000000	0.200000	29.364000
max	9994.000000	2017-12-30 00:00:00	2018-01-05 00:00:00	99301.000000	22638.480000	14.000000	0.800000	8399.976000

Exploratory Data Analysis

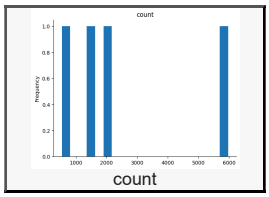
```
No_Of_Cities = data['City'].nunique()
print('there is %d cities'%No_Of_Cities)
```

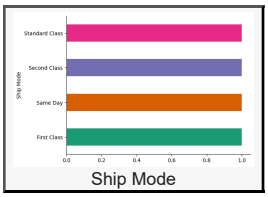
```
there is 531 cities
```

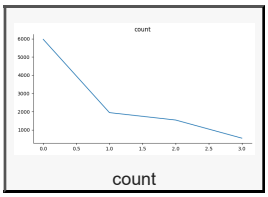
```
ship_mode_count = data['Ship Mode'].value_counts().reset_index()
ship_mode_count
```

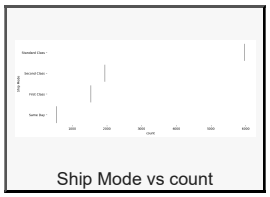
	Ship Mode	count
0	Standard Class	5968
1	Second Class	1945
2	First Class	1538
3	Same Day	543

Next steps: Generate code with ship_mode_count View recommended plots New interactive sheet

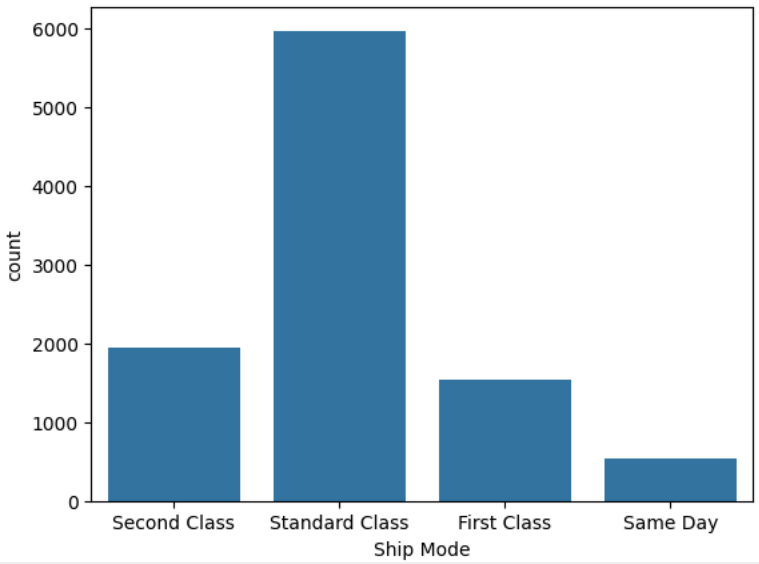






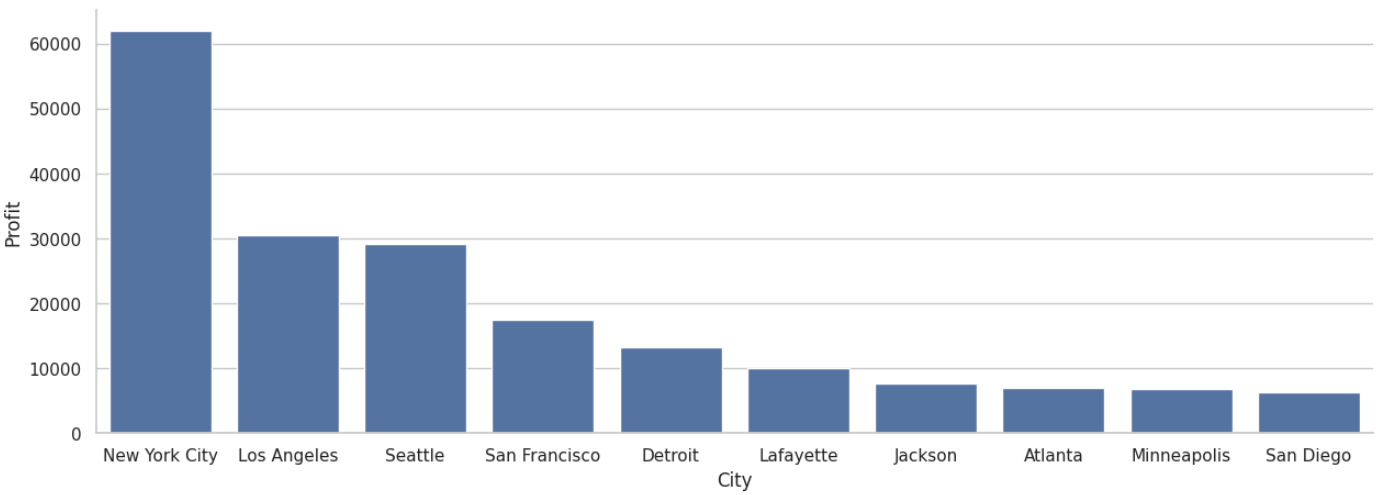



```
sns.countplot(x=data['Ship Mode'])
sns.set_style('ticks')
```




```
sales_city = data.groupby(['City'], as_index=False)['Profit'].sum().sort_values(by='Profit', ascending=False).head(10)

sns.barplot(data = sales_city, x = 'City',y= 'Profit')
sns.despine()
```




```
data.rename(columns = {'Product Name':'product_name'},inplace=True)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
```

```
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 non-null   datetime64[ns]
 3   Ship Date      9994 non-null   datetime64[ns]
 4   Ship Mode      9994 non-null   object
 5   Customer ID    9994 non-null   object
 6   Customer Name  9994 non-null   object
 7   Segment        9994 non-null   object
 8   Country        9994 non-null   object
 9   City           9994 non-null   object
 10  State          9994 non-null   object
 11  Postal Code    9994 non-null   int64
 12  Region         9994 non-null   object
 13  Product ID     9994 non-null   object
 14  Category       9994 non-null   object
 15  Sub-Category   9994 non-null   object
 16  product_name   9994 non-null   object
 17  Sales          9994 non-null   float64
 18  Quantity       9994 non-null   int64
 19  Discount       9994 non-null   float64
 20  Profit         9994 non-null   float64
dtypes: datetime64[ns](2), float64(3), int64(3), object(13)
memory usage: 1.6+ MB
```

```
data.columns
```

```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
       'product_name', 'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

Filtration

Double-click (or enter) to edit

```
data['Profit'].max()
```

```
8399.975999999999
```

```
data.groupby('State')['Sales'].describe()
```

State	count	mean	std	min	25%	50%	75%	max
Alabama	61.0	319.846557	545.761807	3.620	19.56000	70.980	358.58000	3040.000
Arizona	224.0	157.508933	249.710692	1.408	15.16400	61.512	201.98400	1879.960
Arkansas	60.0	194.635500	316.405669	4.300	19.77750	54.420	209.83750	1793.980
California	2001.0	228.729451	491.005264	0.990	19.76000	61.020	225.29600	8187.650
Colorado	182.0	176.418231	324.415072	1.080	15.16000	51.016	175.41800	2549.985
Connecticut	82.0	163.223866	251.732268	3.520	14.71250	50.000	181.91175	1133.350
Delaware	96.0	285.948635	1112.818485	3.380	19.95000	67.005	202.86700	10499.970
District of Columbia	10.0	286.502000	547.419707	9.640	20.26500	35.800	41.04750	1379.920
Florida	383.0	233.612815	1205.490630	1.167	11.95200	41.472	182.83200	22638.480
Georgia	184.0	266.825217	637.796228	1.780	19.23000	70.955	250.44000	6354.950
Idaho	21.0	208.689810	282.061093	3.304	21.31200	89.970	304.77600	1128.390
Illinois	492.0	162.939230	317.122940	0.836	11.20100	36.568	180.25200	2799.960
Indiana	149.0	359.431946	1481.538652	1.980	24.56000	70.080	207.24000	17499.950
Iowa	30.0	152.658667	326.637475	5.400	13.83750	33.465	103.45250	1408.100
Kansas	24.0	121.429583	124.606356	5.760	20.65500	63.980	233.05750	360.380
Kentucky	139.0	263.250000	472.818538	2.610	24.10000	76.300	281.89000	3080.000
Louisiana	42.0	219.453095	365.264153	3.890	21.06000	64.140	232.79750	1665.620
Maine	8.0	158.816250	146.676370	8.260	58.44500	105.720	256.50000	437.850
Maryland	105.0	225.766886	396.914182	1.640	25.02000	89.820	219.90000	2541.980
Massachusetts	135.0	212.106919	336.924698	3.150	22.32500	63.200	245.24750	1737.180
Michigan	255.0	299.096525	826.126521	2.200	22.69000	85.520	278.55000	9892.740
Minnesota	89.0	335.541011	1070.678744	3.750	22.00000	50.400	221.16000	9449.950
Mississippi	53.0	203.232830	365.834352	6.160	30.44000	77.560	245.94000	2430.080
Missouri	66.0	336.441667	830.866752	7.640	25.15500	57.685	235.44000	4899.930
Montana	15.0	372.623467	784.412796	6.096	21.83000	63.980	278.70000	2999.950
Nebraska	38.0	196.445526	459.444553	5.040	17.61750	34.200	113.42250	2479.960
Nevada	39.0	428.951333	881.267531	3.640	31.74000	79.140	183.71000	4535.976
New Hampshire	27.0	270.093481	475.181739	14.820	29.94500	68.620	258.67000	2249.910
New Jersey	130.0	275.110092	890.520072	3.760	19.89250	66.730	208.43750	9099.930
New Mexico	37.0	129.284378	190.093868	4.170	16.68000	45.360	159.99000	883.840
New York	1128.0	275.599531	691.716475	1.240	20.02750	60.045	239.97000	11199.968
North Carolina	249.0	223.305880	648.754454	1.752	15.98400	45.216	189.58800	7999.980
North Dakota	7.0	131.415714	256.602857	2.480	14.45500	25.900	78.93000	704.760
Ohio	469.0	166.861697	353.261659	1.448	14.48000	44.376	155.37200	4499.985
Oklahoma	66.0	298.233182	431.874240	3.520	20.61000	79.550	362.21750	1805.880
Oregon	124.0	140.573790	215.028430	1.080	16.59975	46.596	164.28225	1487.040
Pennsylvania	587.0	198.487077	531.058157	0.852	12.67200	41.472	175.09300	8399.976
Rhode Island	56.0	404.070643	853.179916	2.220	34.77000	71.200	352.89150	5399.910
South Carolina	42.0	201.945476	318.462333	6.460	23.96500	69.970	250.08750	1690.040
South Dakota	12.0	109.630000	148.257140	2.970	13.63500	34.250	152.20500	416.320
Tennessee	183.0	167.551219	360.678518	1.584	12.22800	42.048	122.02400	2314.116
Texas	985.0	172.779742	424.447771	0.444	10.82400	36.288	158.37600	8159.952
Utah	53.0	211.699170	378.723382	4.960	21.36000	60.120	158.90000	1499.950
Vermont	11.0	811.760909	1309.310745	2.040	46.10000	205.030	1004.97500	4404.900
Virginia	224.0	315.342500	792.001296	2.840	27.49500	65.250	251.63250	8749.950

Washington	506.0	273.994605	777.244483	1.344	20.66350	65.940	209.83500	13999.960
West Virginia	4.0	302.456000	313.508627	6.240	63.36000	265.120	504.21600	673.344
Wisconsin	110.0	291.951000	461.590111	1.810	25.63500	93.860	381.60750	2807.840
Wyoming	1.0	1603.136000	NaN	1603.136	1603.13600	1603.136	1603.13600	1603.136

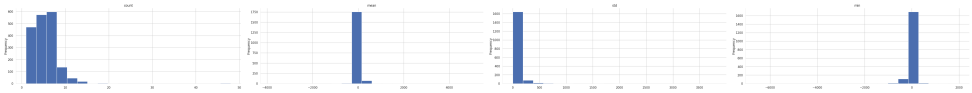
Generate print hello world using rot13 Close

```
data.groupby('product_name')['Profit'].describe()
```

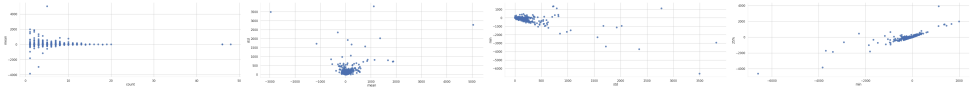
product_name	count	mean	std	min	25%	50%	75%	max
"While you Were Out" Message Book, One Form per Page	3.0	3.462667	0.214197	3.3390	3.33900	3.33900	3.5245	3.7100
#10 Gummed Flap White Envelopes, 100/Box	4.0	4.191950	2.372385	2.1476	2.95295	3.51050	4.7495	7.5992
#10 Self-Seal White Envelopes	4.0	13.030750	13.439354	3.2161	4.87960	8.15115	16.3023	32.6046
#10 White Business Envelopes,4 1/8 x 9 1/2	7.0	31.877257	19.981115	4.2309	17.39370	36.82450	44.1894	58.9192
#10- 4 1/8" x 9 1/2" Recycled Envelopes	10.0	11.528060	5.326773	4.7196	8.21560	8.82740	15.4698	20.5390
...	...	...	...	...	...	...	...	...
iKross Bluetooth Portable Keyboard + Cell Phone Stand Holder + Brush for Apple iPhone 5S 5C 5, 4S 4	5.0	23.128800	23.446036	3.3520	6.70400	11.73200	35.1960	58.6600
iOttie HLCRIO102 Car Mount	5.0	-2.398800	7.499581	-13.9930	-2.99850	-2.99850	1.9990	5.9970
iOttie XL Car Mount	2.0	-25.187400	10.177246	-32.3838	-28.78560	-25.18740	-21.5892	-17.9910
invisibleSHIELD by ZAGG Smudge-Free Screen Protector	7.0	24.466400	14.618097	4.3176	15.11160	25.18600	34.1810	43.1760
netTALK DUO VoIP Telephone Service	7.0	61.488286	37.694775	31.4940	39.36750	52.49000	62.9880	141.7230

1850 rows × 8 columns

Distributions



2-d distributions



Values

```
from matplotlib import pyplot as plt
_df_10['std'].plot(kind='line', figsize=(8, 4), title='std')
plt.gca().spines[['top', 'right']].set_visible(False)
```

Generate a slider using jupyter widgets Close

```
data[(data['Ship Mode'].str.contains('Second Class'))]
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Category
1		CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-BO-10001798	Furniture
2		CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-CH-10000454	Furniture
3		CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-LA-10000240	Office Supplies
18		CA-2014-167164	2014-05-13	2014-05-15	Second Class	AG-10270	Alejandro Grove	Consumer	United States	West Jordan	...	84084	West	OFF-ST-10000107	Office Supplies
		CA-	2014	2014	Second	ZD	Zuschuss		United	San				OFF-AR	Office