```
import pandas as pd
Product_data =pd.read_csv("/content/statsfinal.csv", header=0, sep=",")
print(Product_data)
```

```
      Unnamed: 0        Date  Q-P1  Q-P2  Q-P3  Q-P4      S-P1      S-P2  \
0              0  13-06-2010  5422  3725   576   907  17187.74  23616.50
1              1  14-06-2010  7047   779  3578  1574  22338.99   4938.86
2              2  15-06-2010  1572  2082   595  1145   4983.24  13199.88
3              3  16-06-2010  5657  2399  3140  1672  17932.69  15209.66
4              4  17-06-2010  3668  3207  2184   708  11627.56  20332.38
...          ...         ...   ...   ...   ...   ...       ...       ...
4595        4595  30-01-2023  2476  3419   525  1359   7848.92  21676.46
4596        4596  31-01-2023  7446   841  4825  1311  23603.82   5331.94
4597        4597  01-02-2023  6289  3143  3588   474  19936.13  19926.62
4598        4598  02-02-2023  3122  1188  5899   517   9896.74   7531.92
4599        4599  03-02-2023  1234  3854  2321   406   3911.78  24434.36

          S-P3      S-P4
0      3121.92   6466.91
1     19392.76  11222.62
2      3224.90   8163.85
3     17018.80  11921.36
4     11837.28   5048.04
...        ...       ...
4595   2845.50   9689.67
4596  26151.50   9347.43
4597  19446.96   3379.62
4598  31972.58   3686.21
4599  12579.82   2894.78
```

```
import pandas as pd
Product_data =pd.read_csv("/content/statsfinal.csv", header=0, sep=",")
print(Product_data.head())
```

```
   Unnamed: 0        Date  Q-P1  Q-P2  Q-P3  Q-P4      S-P1      S-P2  \
0           0  13-06-2010  5422  3725   576   907  17187.74  23616.50
1           1  14-06-2010  7047   779  3578  1574  22338.99   4938.86
2           2  15-06-2010  1572  2082   595  1145   4983.24  13199.88
3           3  16-06-2010  5657  2399  3140  1672  17932.69  15209.66
4           4  17-06-2010  3668  3207  2184   708  11627.56  20332.38

       S-P3      S-P4
0   3121.92   6466.91
1  19392.76  11222.62
2   3224.90   8163.85
3  17018.80  11921.36
4  11837.28   5048.04
```

[5]
```
Product_data.dropna(axis=0,inplace=True)
print(Product_data)
```

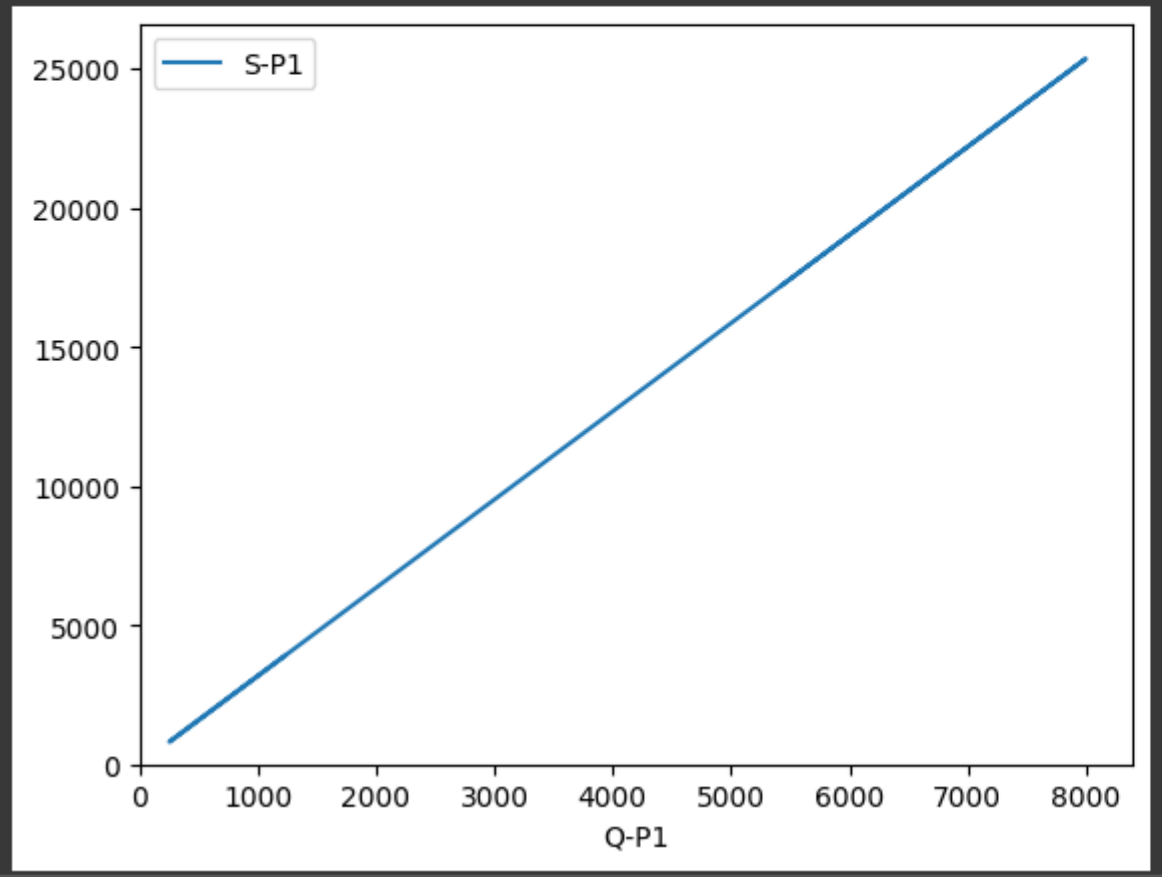
```
      Unnamed: 0        Date  Q-P1  Q-P2  Q-P3  Q-P4      S-P1      S-P2  \
0              0  13-06-2010  5422  3725   576   907  17187.74  23616.50
1              1  14-06-2010  7047   779  3578  1574  22338.99   4938.86
2              2  15-06-2010  1572  2082   595  1145   4983.24  13199.88
3              3  16-06-2010  5657  2399  3140  1672  17932.69  15209.66
4              4  17-06-2010  3668  3207  2184   708  11627.56  20332.38
...          ...         ...   ...   ...   ...   ...       ...       ...
4595        4595  30-01-2023  2476  3419   525  1359   7848.92  21676.46
4596        4596  31-01-2023  7446   841  4825  1311  23603.82   5331.94
4597        4597  01-02-2023  6289  3143  3588   474  19936.13  19926.62
```

```
print(Product_data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  4600 non-null   int64  
 1   Date        4600 non-null   object 
 2   Q-P1        4600 non-null   int64  
 3   Q-P2        4600 non-null   int64  
 4   Q-P3        4600 non-null   int64  
 5   Q-P4        4600 non-null   int64  
 6   S-P1        4600 non-null   float64
 7   S-P2        4600 non-null   float64
 8   S-P3        4600 non-null   float64
 9   S-P4        4600 non-null   float64
dtypes: float64(4), int64(5), object(1)
memory usage: 359.5+ KB
None
```

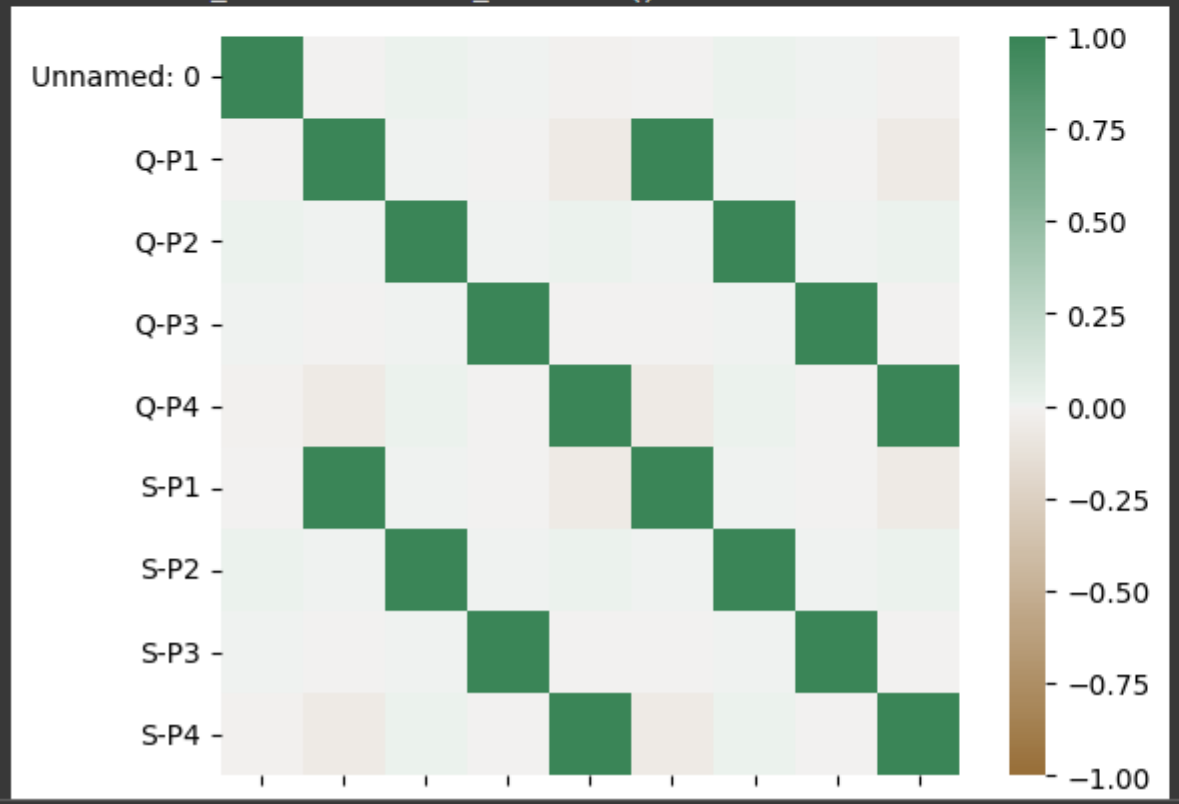
```
10] import matplotlib.pyplot as plt
```

```
import matplotlib.pyplot as plt

Product_data.plot(x ='Q-P1', y='S-P1', kind='line'),
plt.ylim(ymin=0)
plt.xlim(xmin=0)

plt.show()
```

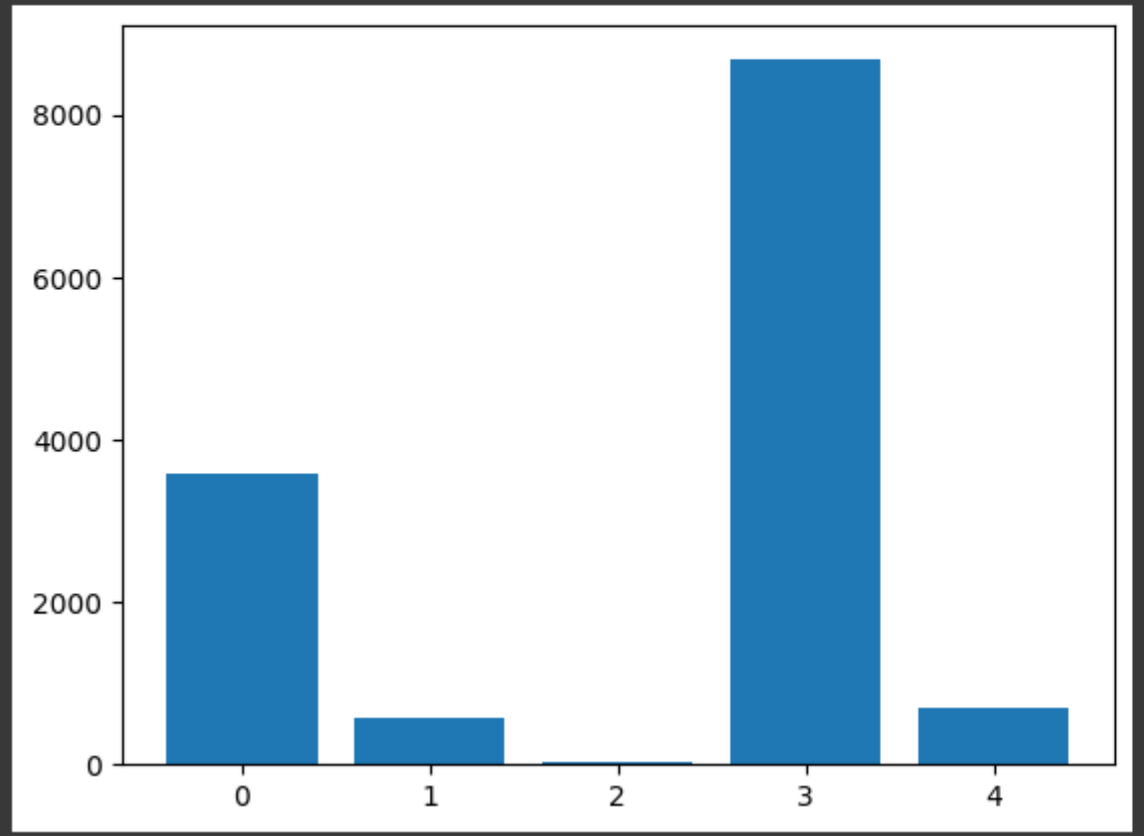


```
import matplotlib.pyplot as plt
import seaborn as sns

correlation_health = Product_data.corr()

axis_corr = sns.heatmap(
correlation_health,
vmin=-1, vmax=1, center=0,
cmap=sns.diverging_palette(50, 500, n=500),
square=True
)

plt.show()
```

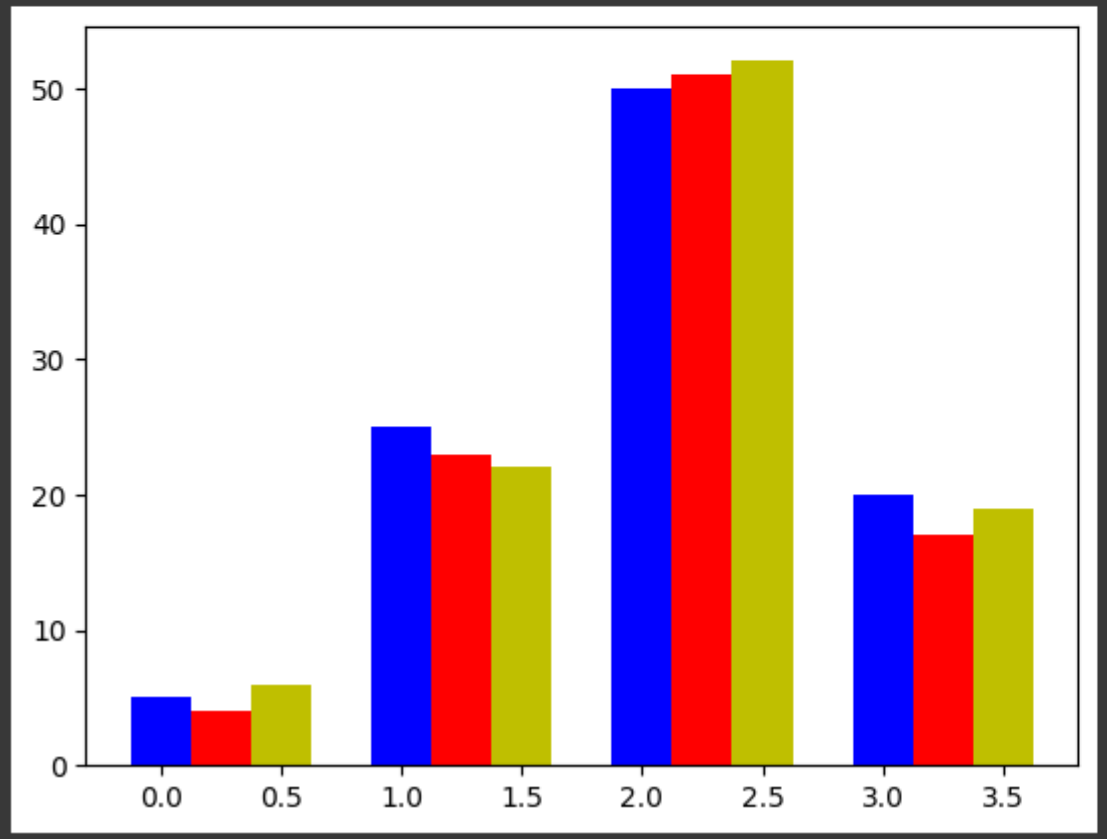
```
<ipython-input-14-8bba958e8c92>:4: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or
  correlation_health = Product_data.corr()
```



```
import pandas as pd
import matplotlib.pyplot as plt
data = [3578,567,23,8675,697]
plt.bar(range(len(data)),data)
plt.show()
```

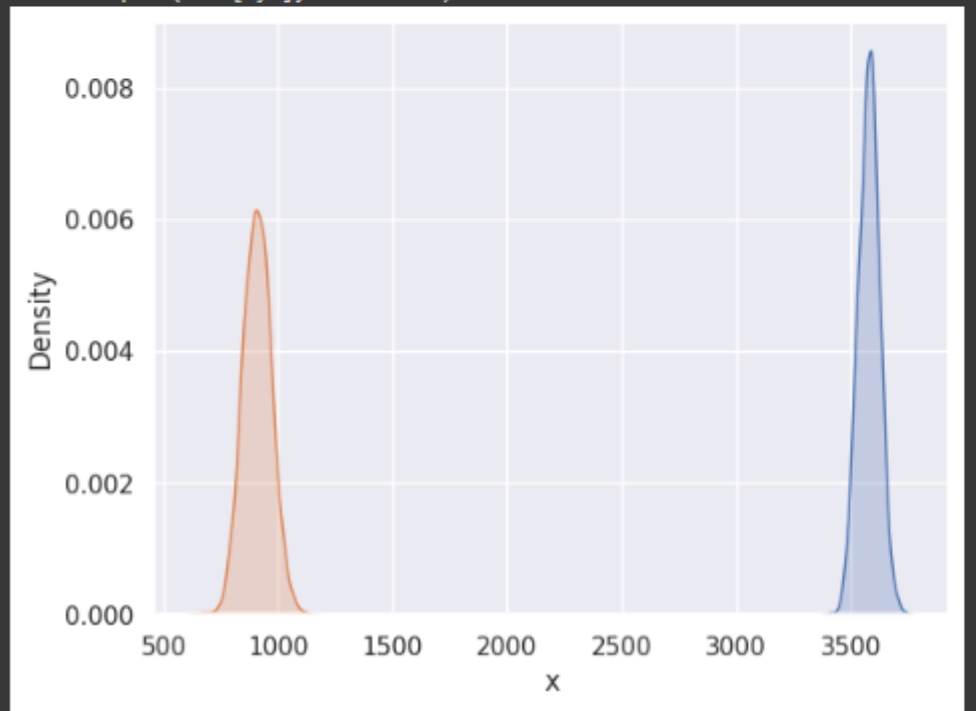


```
# Multiple Bar Plots

import numpy as np
data = [[5., 25., 50., 20.],
        [4., 23., 51., 17.],
        [6., 22., 52., 19.]]
x = np.arange(4)
plt.bar(x + 0.00, data[0], color='b', width=0.25)
plt.bar(x + 0.25, data[1], color='r', width=0.25)
plt.bar(x + 0.50, data[2], color='y', width=0.25)
plt.show()
```

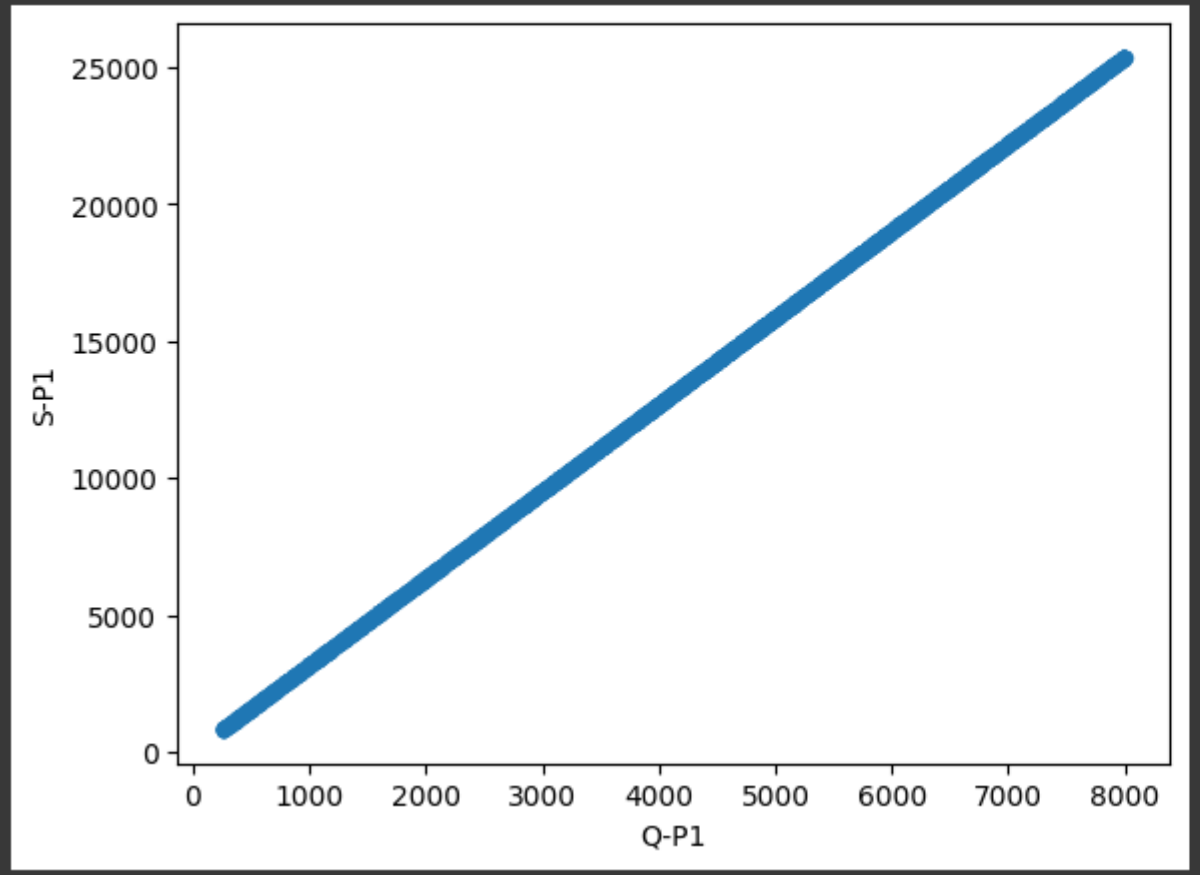


```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()

data = np.random.multivariate_normal([3578, 907], [[779, 3725], [2082,595]], size=2000)
data = pd.DataFrame(data, columns=['x', 'y'])
sns.kdeplot(data["x"], shade=True)
sns.kdeplot(data["y"], shade=True)
plt.show()
```

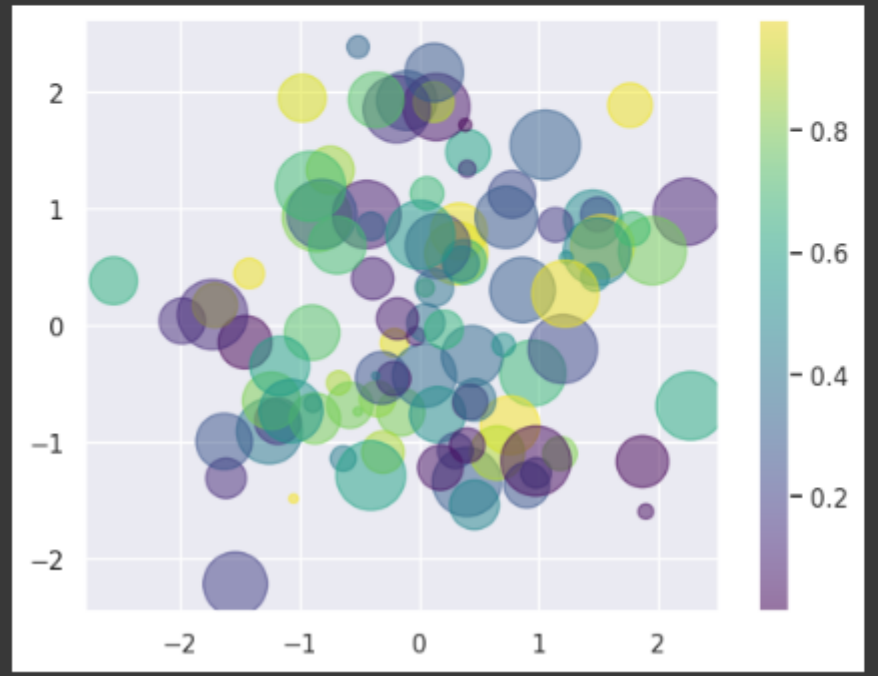
```
<ipython-input-20-94866236580d>:7: RuntimeWarning: covariance is not positive-semidefinite.
  data = np.random.multivariate_normal([3578, 907], [[779, 3725], [2082,595]], size=2000)
<ipython-input-20-94866236580d>:9: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data["x"], shade=True)
<ipython-input-20-94866236580d>:10: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data["y"], shade=True)
```



```
[13] import matplotlib.pyplot as plt
     Product_data.plot(x = 'Q-P1', y = 'S-P1', kind = 'scatter')
     plt.show()
```



```
import numpy as np
import matplotlib.pyplot as plt
rng = np.random.RandomState(0)
x = rng.randn(100)
y = rng.randn(100)
colors = rng.rand(100)
sizes = 1000 *rng.rand(100)

plt.scatter(x, y, c=colors, s=sizes, alpha=0.5, cmap='viridis')
plt.colorbar()
plt.show()
```



```
import matplotlib.pyplot as plt
data = [3725, 576, 907, 7047]
plt.pie(data)
plt.show()
```