homework_one

2018 年 10 月 22 日

0.1 1.(a) 答案:

In [2]:
```
import numpy as np
import pandas as pd
import seaborn as sns

# 价格、重量原始数据
price = np.array([5.89, 49.59, 59.98, 159, 17.99, 56.99, 82.75, 142.19, 31, 125.5,
         4.5, 22, 52.9, 61, 33.5, 328, 128, 142.19, 229, 189.4])

weight = np.array([1.4, 1.5, 2.2, 2.7, 3.2, 3.9, 4.1, 4.1, 4.6, 4.8,
          4.9, 5.3, 5.5, 5.8, 6.2, 8.9, 11.6, 18, 22.9, 38.2])
# 四分位数定义
quartile = np.array([25, 50, 75])



# 计算价格、重量的四分位数
price_q1, price_median, price_q3 = np.percentile(price, quartile)
weight_q1, weight_median, weight_q3 = np.percentile(weight, quartile)

print("price 的四分位数 Q1, MEDIAN, Q3 =", price_q1, price_median, price_q3)
print("weight 的四分位数 Q1, MEDIAN, Q3 =", weight_q1, weight_median, weight_q3)
```

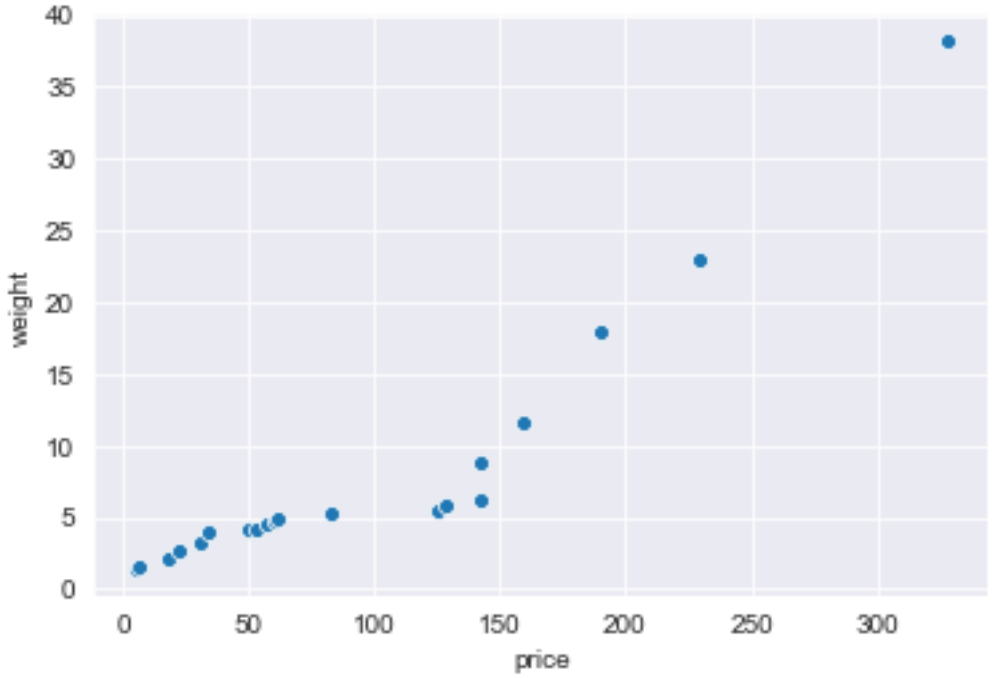
```
price 的四分位数 Q1, MEDIAN, Q3 = 32.875 60.489999999999995 142.19
weight 的四分位数 Q1, MEDIAN, Q3 = 3.7249999999999996 4.85 6.875
```

0.2 1. (b) 答案:

In [3]:
```
# 排序数据
data = np.column_stack((sorted(price), sorted(weight)))
df = pd.DataFrame(data, columns=["price", "weight"])

# 绘制 Q-Q 图
sns.set_style("darkgrid")
sns.scatterplot(x="price", y="weight", data=df, sizes=80)
```

Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x117d7c0f0>



0.3 1.(c) 答案:

In [4]:
```
p_min, p_max = min(price), max(price)
w_min, w_max = min(weight), max(weight)
new_min, new_max = 1, 10

price_normalized = [(n - p_min)/(p_max - p_min)*(new_max - new_min) + new_min for n in
weight_normalized = [(n - w_min)/(w_max - w_min)*(new_max - new_min) + new_min for n i
```

```
        print("规范化 price: ", price_normalized)
        print()
        print("规范化 weight: ", weight_normalized)
```

```
规范化 price:  [1.0386707882534776, 2.2544358578052552, 2.5434930448222564, 5.298299845440495,
```

```
规范化 weight:  [1.0, 1.0244565217391304, 1.1956521739130435, 1.3179347826086958, 1.4402173913
```

0.4 1.(d) 答案

```
In [5]: from scipy.stats import pearsonr
```

```
        print("pearson 互相关系数计算一: ", pearsonr(price, weight)[0])
```

```
pearson 互相关系数计算一:  0.5363070272140884
```

```
In [6]: print("pearson 互相关系数计算二: ", np.corrcoef(price, weight)[0, 1])
```

```
pearson 互相关系数计算二:  0.5363070272140884
```

0.5 2.(a) 答案

从小到大排序
欧式距离：x1, x4, x3, x5, x2;
曼哈顿距离：x1, x4, x3, x5, x2;
上确界距离：x1, x4, x3, x5, x2;
余弦相似度：x1, x3, x4, x2, x5;

```
In [7]: # 第二题
```

```
        x = np.array([1.4, 1.6])
        dataset = np.array([[1.5, 1.7],
                            [2, 1.9],
                            [1.6, 1.8],
                            [1.2, 1.5],
```

```
                [1.5, 1.0]])


        # 各数据点欧式距离
        euc_dist = [np.linalg.norm(x-data) for data in dataset]
        print("各数据点欧式距离:", euc_dist)

        # 各数据点曼哈顿距离
        man_dist = [np.linalg.norm(x-data, ord=1) for data in dataset]
        print("各数据点曼哈顿距离:", man_dist)

        # 各数据点上确界距离
        sup_dist = [np.linalg.norm(x-data, ord=np.inf) for data in dataset]
        print("各数据点上确界距离:", sup_dist)

        # 各数据点上的余弦相似度
        cos_dist = [float(np.dot(x, data))/(np.linalg.norm(x)*np.linalg.norm(data)) for data in
        print("各数据点余弦相似度:", cos_dist)
```

```
各数据点欧式距离: [0.14142135623730948, 0.6708203932499369, 0.28284271247461906, 0.223606797749
各数据点曼哈顿距离: [0.19999999999999996, 0.8999999999999999, 0.40000000000000013, 0.3000000000
各数据点上确界距离: [0.10000000000000009, 0.6000000000000001, 0.20000000000000018, 0.1999999999
各数据点余弦相似度: [0.999991391443956, 0.9957522612528874, 0.9999694838187877, 0.9990282349375
```

0.6 2.(b) 答案

欧式距离：x1, x3, x4, x2, x5

```
In [11]: from sklearn import preprocessing


        x_normalized = preprocessing.normalize([[1.4, 1.6]], norm="l2")
        dataset_normalized = preprocessing.normalize(dataset, norm="l2")

        print("归一化数据点 x: ", x_normalized)
        print()
        print("归一化数据集 x1, x2, x3, x4, x5: ", dataset_normalized, sep='\n')
```

```
        print()

        # 各数据点欧式距离
        euc_dist = [np.linalg.norm(x_normalized[0]-data) for data in dataset_normalized]
        print("各数据点欧式距离:", euc_dist)
```

```
归一化数据点 x:  [[0.65850461 0.75257669]]

归一化数据集 x1, x2, x3, x4, x5:
[[0.66162164 0.74983786]
 [0.72499943 0.68874946]
 [0.66436384 0.74740932]
 [0.62469505 0.78086881]
 [0.83205029 0.5547002 ]]

各数据点欧式距离: [0.004149350803200864, 0.09217091457843411, 0.007812321193114019, 0.044085486
```