**Data Mining - Assignment1**

**1930026143 薛劭杰**

**Q1**.

**(1). F**

**(2). F**

**(3). F**

**(4). F**

**(5). T**

**Q2.**

**(1). If a term occurs in only one document, , and the . So the** **inverse document frequency transformation has high result. If a term occurs in every document, , and the . So the frequency of the inverse document frequency transformation is zero.**

**(2). By the inverse document frequency transformation, we can search the articles with common ground including the same terms. It can avoid the high-frequency words only come from the same text, especially in academic journals. So this transformation can help us to recommend what we really need when looking for some literature.**

**Q3.**

**(1). There are 24 data tuples in increasing order.**

**Median: (, so the median of 12th and 13th number is the median. Median = .**

**Mode: The most frequent number is 35, frequency is 5. Mode = 35.**

**25th percentile:** **(, so Q1 = = 20.**

**75th percentile: , so Q3 = = 35.**

**(2). 1. The bin is site is 5, so we can divide into 5 groups.**

**bin: 13 15 16 16 19**

**bin: 20 20 21 22 22**

**bin: 25 25 25 25 30**

**bin: 33 33 35 35 35**

**bin: 35 35 40 45**

**2. Then we can get the median of each bins:**

**median1: 16; median2: 21; median3: 25; median4: 35 median5:**

**3. After smoothing by bin median:**

**bin: 16 16 16 16 16**

**bin: 21 21 21 21 21**

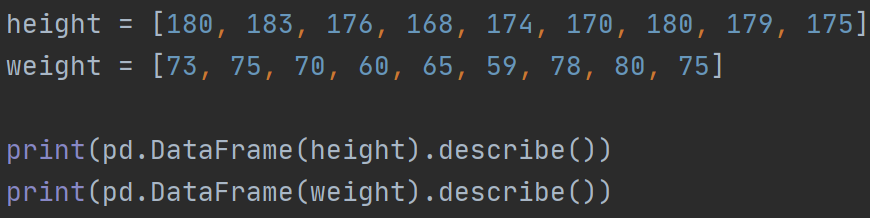
**bin: 25 25 25 25 25**

**bin: 35 35 35 35 35**

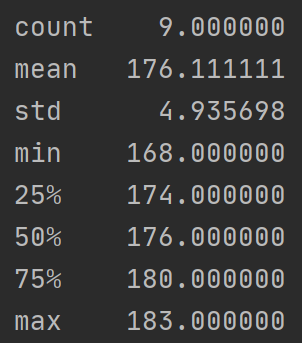
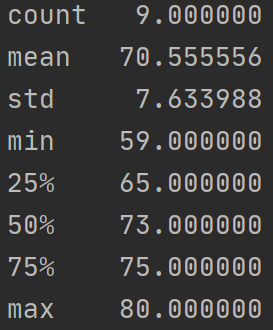
**bin: 37.5 37.5 37.5 37.5**

**Q4.**

**(1). Firstly, we can check the information like the median, mode, Q1 and Q3.**

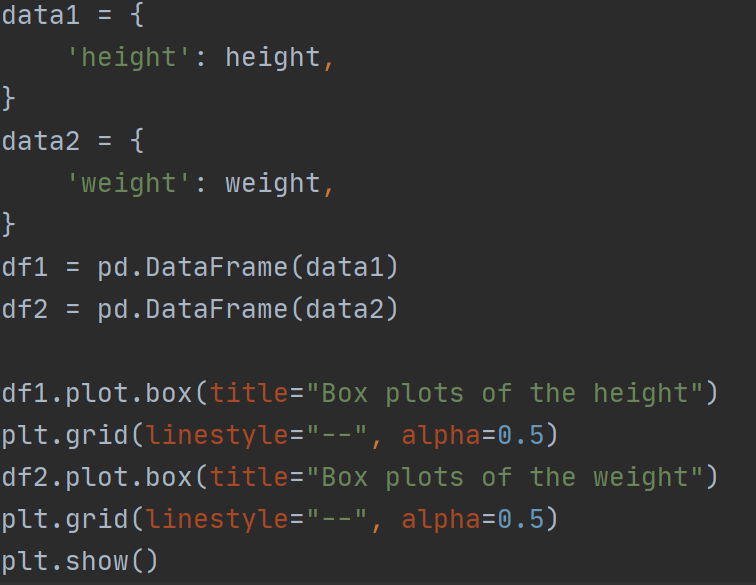
****

**Then we can get:**

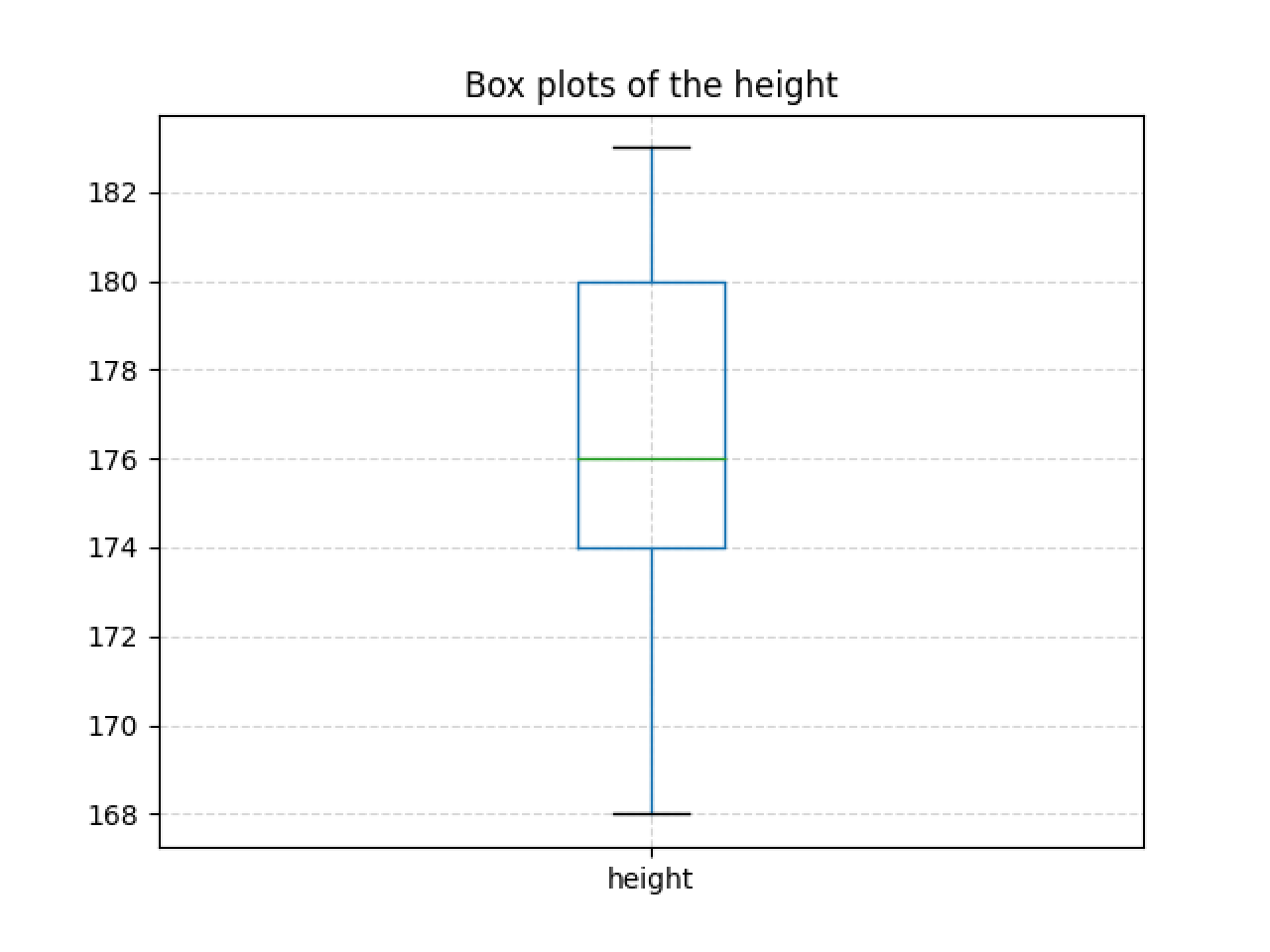
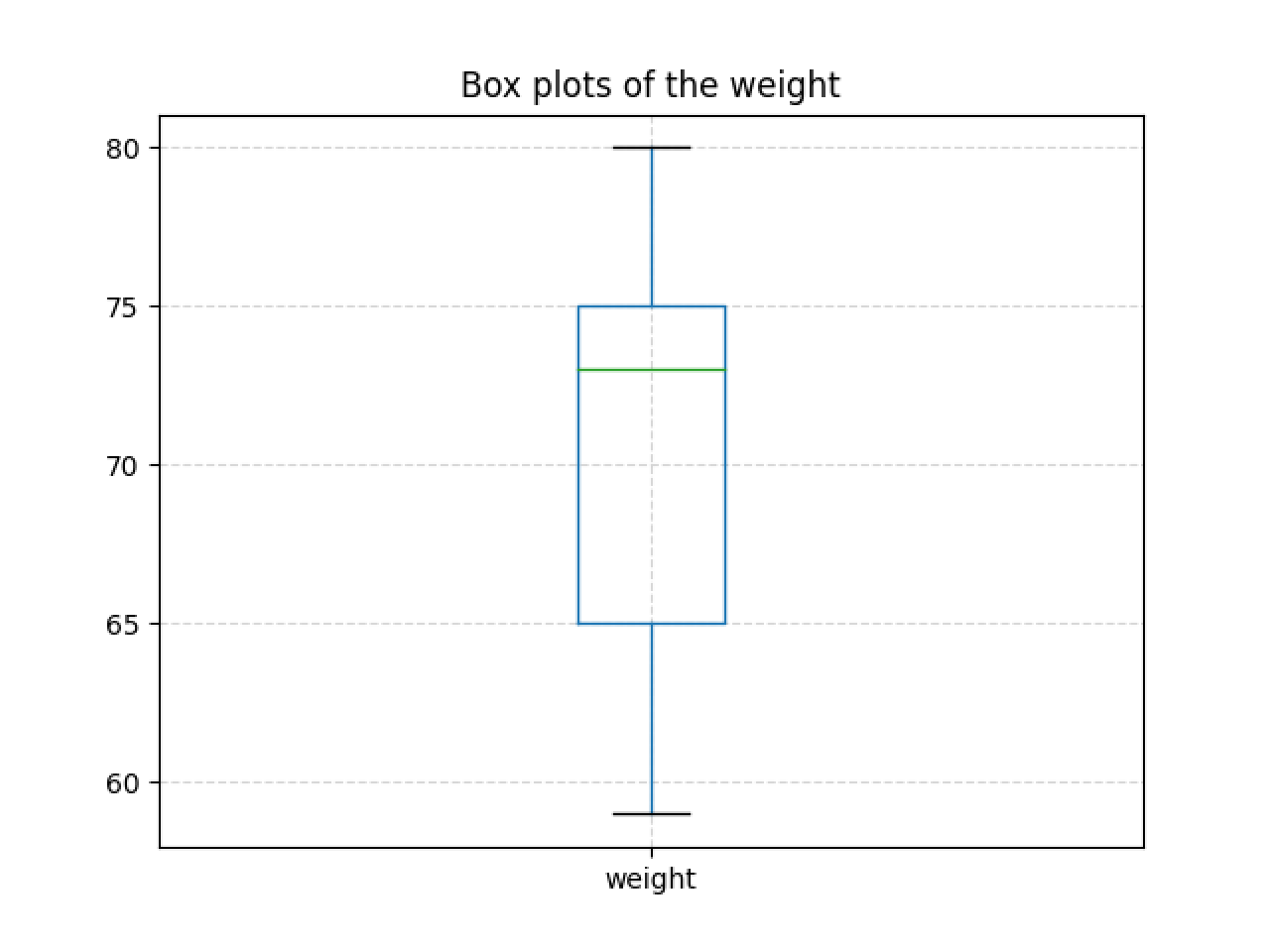
** **

**Figure 1 The describe of height Figure 2 The describe of weight**

**Secondly, drawing the box plots of weight and height.**

****

**Then we can get:**

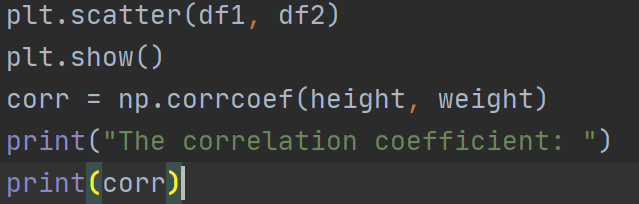
** **

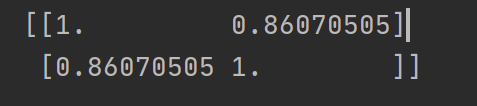
**(2).**

**Height(X) = [180, 183, 176, 168, 174, 170, 180, 179, 175]  
weight(Y) = [73, 75, 70, 60, 65, 59, 78, 80, 75]**

**= E(XY) –**

**We can check the result by Python:**

****

****

**(3). There are some steps to use PCA to reduce these to one dimensional attribute.**

**Combine two pieces of data into a matrix**

**[180 183 176 168 174 170 180 179 175**

**73 75 70 60 65 59 78 80 75]**

1. **Then subtract the average of each of them**

**E(X) = 176.111; E(Y) = 70.556**

**[ 3.889, 6.889, -0.1111, -8.111, -2.111, -6.111, 3.889, 2.889, -1.111**

**2.444, 4.444, -0.5556, -10.56, -5.556, -11.56, 7.444, 9.444, 4.444]**

1. **Then we should calculate the covariance matrix of the one.**

**According to the function: , we can get the covariance matrix**

**[ 21.654, 28.827**

**28.827, 51.802].**

1. **And then find out the eigenvalues and eigenvectors of the covariance matrix.**

**We can get . We can get 69.258, 4.198.**

**By the eigenvalues, we can find out , .**

1. **. so, we choose the as the basic.**

**[ 4.105, 7.37, -0.5328, -13.23, -5.846, -13.05, 8.382, 9.575, 3.226]**

**The data after transformation is [ 4.105, 7.37, -0.5328, -13.23, -5.846, -13.05, 8.382, 9.575, 3.226].**

**To conclude, the reduced 1-dimensional attribute value for Abel, Dale and Igor is**

**[4.105, -13.23, 3.226]**

**(5).**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Hazel** | **Brown** | **Blue** | **Total** |
| **Black** | **1** | **1** | **1** | **3** |
| **Brown** | **1** | **2** | **0** | **3** |
| **Blond** | **1** | **1** | **1** | **3** |
| **Total** | **3** | **4** | **2** | **9** |

**( are the expected values of this table).**

**, , ;**

**, , ;**

**, , ;**

**.**

**The degree of freedom: .**

**According of the Chi-Square distribution table, ，is much larger than 1.5. So there's no obvious difference between colors, which means that the attributes “eye-color” and “hair-color” are independent of each other.**