Course Report
for
Programming Fundamentals

Institute of Economic and Social Research

Student ID: 2016050614

Student Name: 曾晓婷

Email: 496615431@qq.com

Mobile Phone Number: 18029265536

Table of Contents

Table of Figures

1. Introduction

1.1 Document Objectives

This report mainly focusses on illustrate the process of analyzing a given dataset *TCGA* using python. The whole process focuses on discovering some groups of genes similar to each other and appear similar change patterns on samples level. The dataset this project bases on is provided by Jinan University on website.

1.2 Overview

The dataset is given in excel form. It describes some kind of nature of 1881 genes shown in 255 samples given no units. I assume that as genes' activeness, and I will extend the analyzed result's practical significance under this assumption.

- **General nature of the data**

  Numbers of data in dataset ranges from zero to nearly 5000000, and most data of genes have normal distribution.

- **Intended use**

  This analysis can be a reference to more general analysis of the genes' correlation. This project mainly focusses on finding the relationship among relatively small group of genes rather than discover every relationship. And the analysis algorithm used in the project may be inspiring to others.

2. Data-wide Design Decisions

2.1 Interfaces

- **NumPy**

  NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. It is also the fundamental package for data analysis in this project.

- **Pandas**

  Pandas is a powerful data analysis toolkit constructed based on NumPy. It provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. In this project I only use it for reading excel file.

- **Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits. Most of plots in this project is drawn with its help.

- **Sklearn**

  Scikit-learn provides simple and efficient tools for data mining and data analysis building on NumPy, SciPy and matplotlib. The algorithm used in this project are mainly provided by it. I conduct mean shift clustering and linear regression analysis with it.

- **Seaborn**

  Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics. It provides great help in drawing scatter plots.

2.2 Analysis algorithms

- **Mean shift clustering**

  Among many approaches of clustering, I choose mean shift clustering for this project. It is a simple and flexible clustering with nice advantages over other approaches. Compared with others, such as k-means, mean shift clustering requires no number of clusters, which is suitable for the situation when the number of clusters is unknown. Building upon the concept of kernel density estimation, mean shift cleverly exploits the density of the points in an attempt to generate a reasonable number of clusters.

- **Cosine similarity**

  Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. Different from Euclidean distance, it measures how similar two documents are likely to be in terms of their subject matter. It ranges from 0 to 1, which is a simple and direct, and efficient measure of similarity.

- **Multiple linear regression**

  Multiple linear regression is a simple method to analysis the quantitative relationship between one explained variable and several explanatory variables. It uses linear function to model the relationships, with the least squares approach often.

2.3 Operations

My main goal is to find some genes that appear in the same trend and explain their correlation in quantitative level using linear models. I manage it by the following steps. First of all, I do data prepossessing, cleaning the zero values and do format arrangement. Secondly, I observe the distribution of data by generating the mean and max value of each gene's dataset, box plotting some of the most active gene's distribution. Then I do exploratory analysis by drawing scatter plots, intended to find out potential correlation between genes. After I get some idea of the data, I do clustering using mean shift and separate the data into different groups. Finally, I plan to do quantitative analysis using cosine similarity and multiple linear regression.

3. Detailed Data Analysis

3.1 Data prepossessing

After downloading the dataset from the website, I import it into python as an array named *odata*. In order to reduce the complexity of work, I cancel genes' dataset that with all values being zero. Number of genes reduce from 1881 to 1466. The dataset is clean and clear with no format error, so no operation is needed anymore. Then I generate an array named *data*. These operations are based on *NumPy* and *pandas*.

3.2 Descriptive statistics

Firstly, I plot the mean values and max values of genes' appearance on samples. The plots show similar results. From the plots we can see that several genes are very active. So, I decide to plot the distribution of these relatively active genes next.

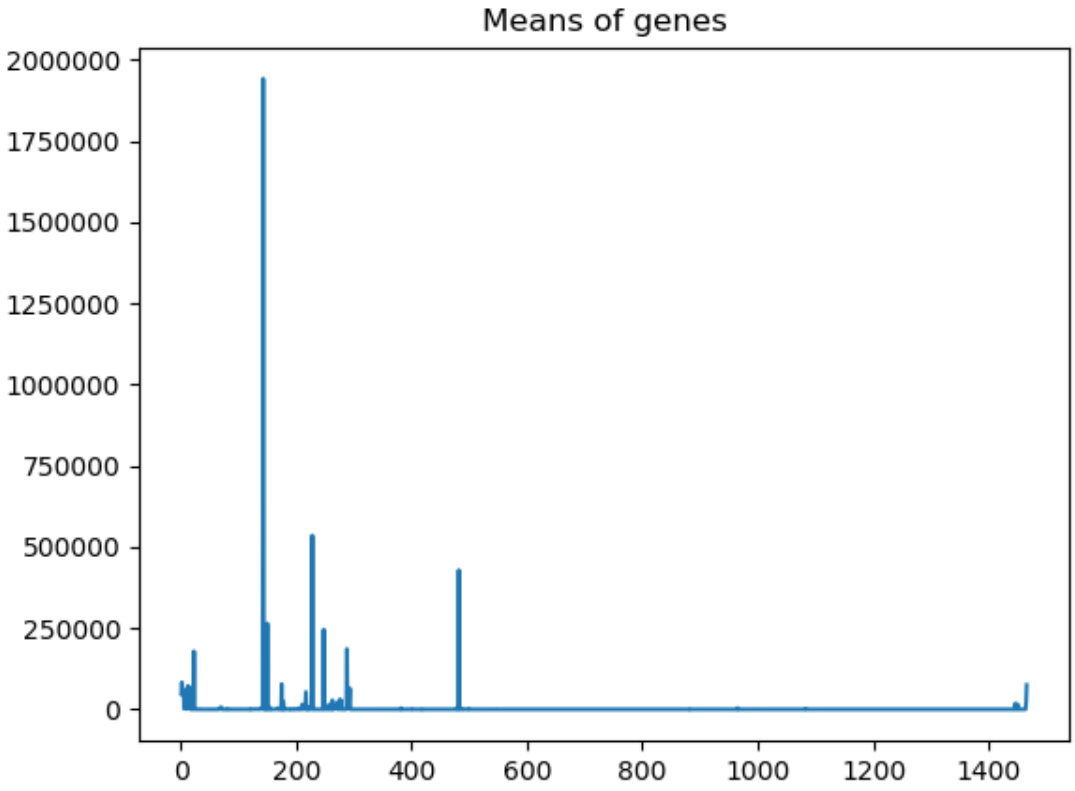

Figure 1 Mean values of genes

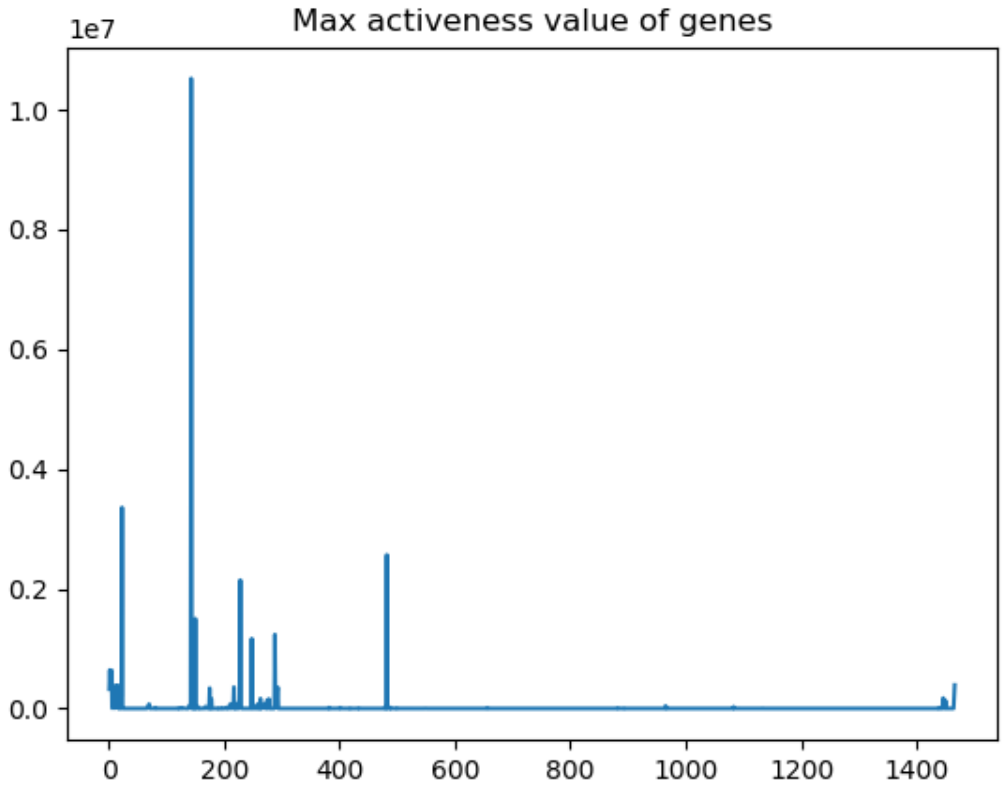

Figure 2 Max values of genes

I draw the most active 40 genes' distribution without outliers. The first boxplot, figure 3, shows that one gene has extremely large activeness value, with mean around 200000 and max over 400000, mostly distributed between 100000 and 250000. But with its existence, other genes' distribution is hard to observe. So, I remove it, "hsa-mir-143", and draw the second boxplot. According to the results shown in figure 4, the genes can be divided into 3 groups by their distribution, and genes in each group may have correlation with each other.

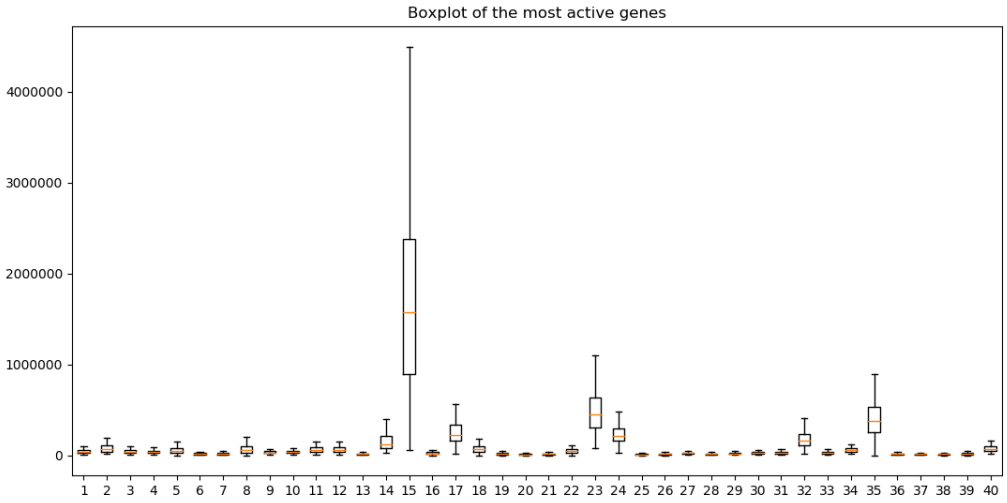

Figure 3 Boxplot of the most active genes

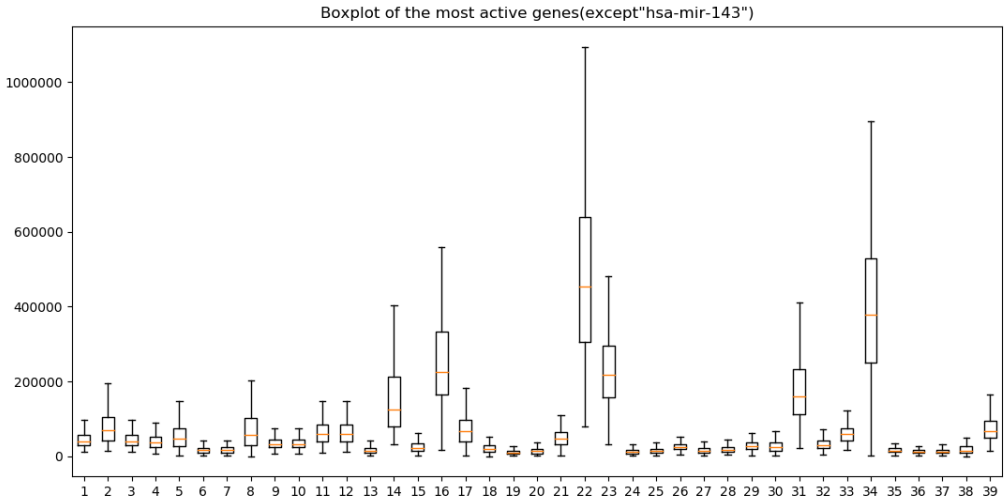

Figure 4 Boxplot of the most active genes(except"hsa-mir-143")

3.3 Exploratory analysis

I further explore the form relationship plotting scatter plots through the interface *seaborn*. Due to memory error, I cannot draw scatter plots of each two genes in dataset, but only five genes: “hsa-let-7a-1”, “hsa-let-7a-2”, “has-let-7a-3”,”hsa-let-7b” and “hsa-let-7c”. Figure 5 shows that there is clearly linear relationship between each two among gene “hsa-let-7a-1”, “hsa-let-7a-2” and “has-let-7a-3”. The rest ones show positive relativity, but the form is not obvious. Therefore, linear regression model can be used for quantitative analysis for genes.

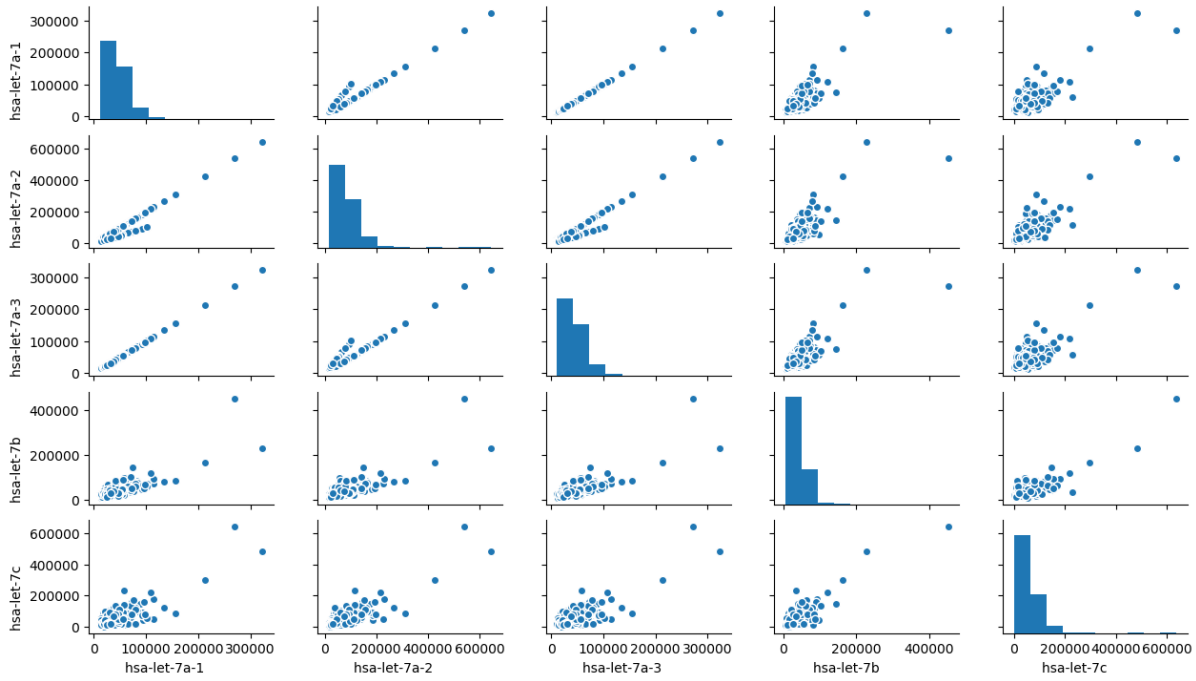

Figure 5 Scatter plots of 5 genes selected

3.4 Data analysis

In order to see how many groups of genes are in the same changing trend, I turn to clustering. Mean shift clustering is chosen to analysis this dataset. Based on *sklearn*, I implement that algorithm and find out that they can be clustered to 39 sets.

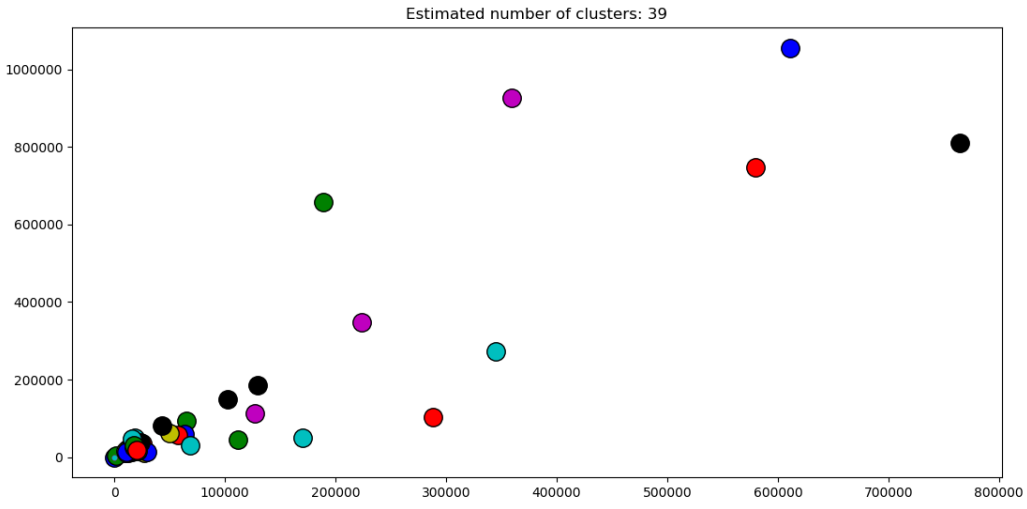

Figure 6 Mean shift clustering result

After getting the clustering result, I want to see which genes are in the same set. I use cosine similarity to measure the similarity of genes. I define “similar” as the cosine similarity no less than 0.85(the range is 0 to 1). It shows that gene “hsa-let-7a-1”, “hsa-let-7a-2” and other 94 genes are in the same group, to

say, group 1. “hsa-mir-105-1” and other two genes are in group2, etc. (Details can be seen in python code files.)

Finally, I use multiple linear regression to analyze the relationship in groups. Blue dots are the exact values of the specific gene and black lines are formed by the predicted values.

In group1, there are 95 correlated genes. For convenience, I only select first 15 genes in the group to do the regression. The estimated function is:

$\widehat{gene}_1 = 0.0019gene_2 + 0.9967gene_3 + 0.0019gene_4 + 0.0006gene_5 + 0.0853gene_6 + 0.0004gene_7 + 0.2181gene_8 + 0.2118gene_9 + 0.0010gene_{10} + 0.0804gene_{11} + 0.0003gene_{12} + 0.0620gene_{13} + 0.0618gene_{14} + 0.0526gene_{15} + 0.0528gene_{16}$

The predicted results are shown in figure 7, we can see that the exact and the predicted values overlap. The r-squared is a coefficient measures how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model. It ranges from 0 to 1. The high r-squared for the first estimated model also shows that the model has good estimated effect.

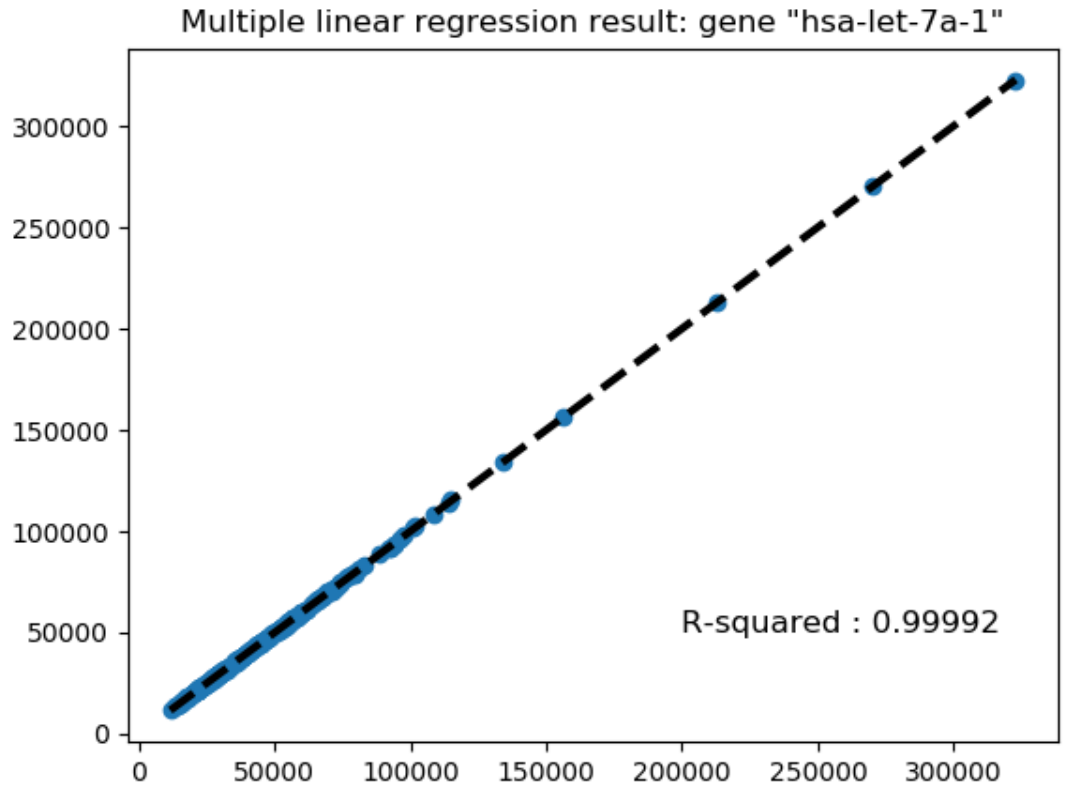

Figure 7 Linear regression result of gene"hsa-let-7a-1"

For group2, there are only 3 genes, so I use them all. The estimated function is:

$\widehat{gene}_{17} = 0.6169gene_{18} + 0.1980gene_{1393}$

R-squared for group2’ model is not so high as the model in group1, but it is also larger than 0.9. The dots and the predicted value line are basically fitted, so it is also a nice estimation.

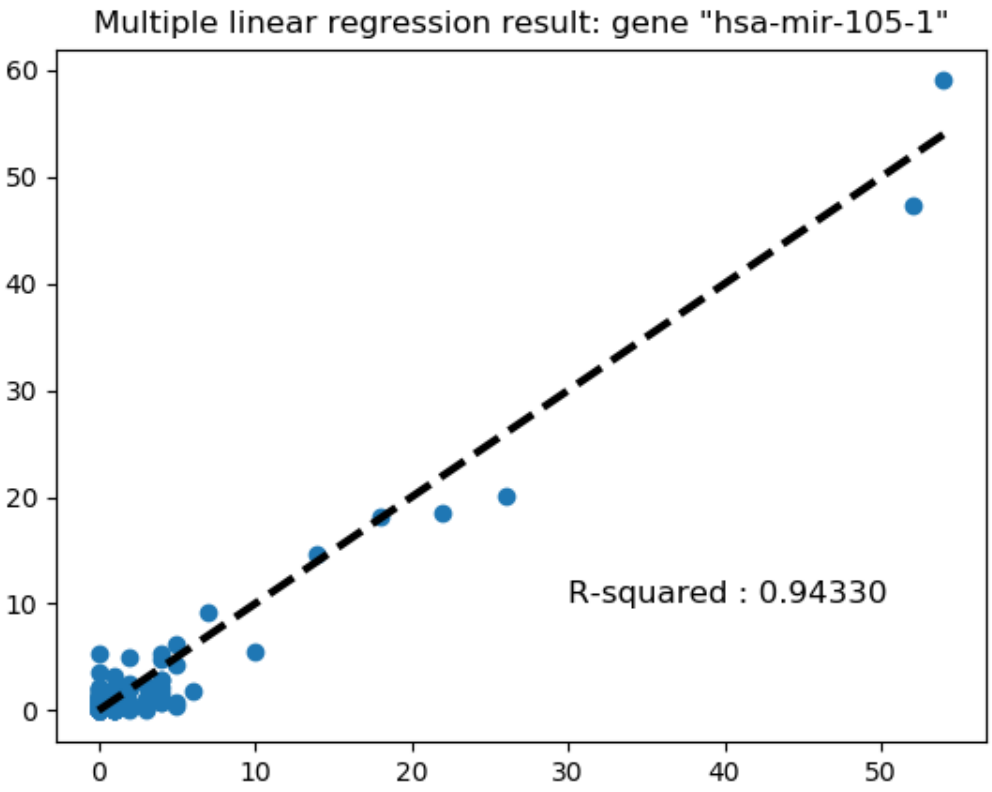

Figure 8 Linear regression result of gene"hsa-mir-105-1"

4. Conclusion

After data analysis, I find out the genes can be divided into 39 groups and there exist quantitative relationships within different groups of genes using linear regression estimation. I also list the linear estimation functions above.

- **Realistic significance**

Based on the assumption that the numbers in dataset are activeness of genes, the above discovered relationship shown above may reveal pathogenesis of some disease. Maybe a disease is related to abnormality in a specific gene group. For example, although I use gene1" hsa-let-7a-1" as explained variable, it can be replaced by any one among other 95 genes. Namely, they can explain each other's performance. But if, one exact value occurs, being very different from predicted value, then we may pay attention to that. If needs, we can generate the predicted value using this model at that time and see the difference.

- **Improvement**

Due to my limited abilities, I can only select part of the genes to do the analysis so far. I believe that there might be some method that enables us to analyze the whole dataset in very short time with efficient code.

Meanwhile, except for linear regression, there are other models we can use, such as logarithm-logarithm model and integral model. They might fit better for some groups.

5. Appendices

5.1 Data resource

https://ming.jnu.edu.cn/python.html

5.2 References

Pandas: http://pandas.pydata.org/pandas-docs/stable/

NumPy: https://docs.scipy.org/doc/numpy/user/whatisnumpy.html

Matplotlib: https://matplotlib.org/index.html

Sklearn: http://scikit-learn.org/stable/

Seaborn: http://seaborn.pydata.org/

[1]Pang-Ning Tan, Michael Steinbach, Vipin Kumar.数据挖掘导论[M].北京:人民邮电出版社,2011.