Data Visualization: An Exploration of Dimension Reduction

MATH 298 Report

Xiaohong Liu

February 2, 2018

Contents

1 Introduction

The main goal of data visualization is to present data clearly and effectively, using plots and charts. Data visualization means "information that has been abstracted in some schematic form, including attributes or variables for the units of information" [1]. Effective visualization helps users analyze the data and gather useful information. It makes complex data more accessible, understandable and usable.

There are many varieties of low dimensional or high dimensional data, such as stock prices, images, text messages, videos, audios, etc. In this project, we will discuss low dimensional data visualization first, and explore how to extract and visualize the most important characteristics from high dimensional data.

Dimension reduction is a technique to extract hidden structures from high dimensional data. It transforms high dimensional data to meaningful lower dimensional data. Effective dimension reduction identifies the intrinsic dimension of data, captures the essential features, removes unimportant information and noise. Ideally, dimension reduction makes data processing, analyzing, modeling, and predicting much faster and much easier, which not only saves huge computer memory, but also saves plenty of money and time.

Dimension reduction techniques include supervised dimension reduction, such as Linear Discriminant Analysis(LDA), and unsupervised dimension reduction, such as Principal Components Analysis (PCA), Multidimensional Scaling (MDS), Locally Linear Embedding (LLE), Laplacian Eigenmap and ISOmap, etc.

In Section 1 of this report, we will introduce low dimensional (less than or equal to three dimensions) data visualization briefly; in Section 2, we will discuss high dimensional data visualization using different dimension reduction techniques, including unsupervised techniques and supervised techniques, with experiment introduced to compare the effects of each dimension reduction technique.

1.1 Related Definitions

To illustrate data set and variables more precisely, here we briefly introduce some related definitions.

• **Variable.** The word “variable” means a changing quantity. In math, it could be: a quantity that could be any of a set of values, or a symbol represents that quantity (like x or y).

• **Data**. Data is a set of values of qualitative or quantitative variables. As we know, data is a general concept which includes many varieties, such as continuous variable and categorical variable; or text, video and audio. Data can be stored with different formats: .txt, .csv, .jpg, or .png, etc. They can be image, binary data, video, audio or webpage. Data also can be treated as a set of variables, and the size of data set (number of observations, instances, or samples) can be from 1 to mega millions. Additionally, the dimension of data can be from 1 to tens of thousands, which makes it very difficult to figure out the most important features.

• **Data Visualization.** In general, data visualization is a technique which presents data values or related statistics information visually. Based on the complexity and diversity of data, data visualization has many branches and rich contents. In this report we'd like to introduce data visualization comprehensively.

• **Numerical Variable.** The values of numerical variable can be measured in numerical way with or without units. It can be count, height, weight, distance, etc.

• **Categorical Variable.** The values of categorical variable can be divided into categories. For example, race, zip code, nationality, species, gender are typical categorical variables. The distance between sub-categories are not measurable, as we cannot measure the quantitative distance between male and female although they have much qualitative difference. Categorical variable can be transformed into discrete numerical variable, making data analyzing more doable.

• **Discrete Variable and Continuous Variable.** Numerical variable includes discrete variable and continuous variable. Discrete variable has a countable number of values. It is also be named as counts. For instance, integer is discrete data. Continuous variable attains values in an intervals.

• **Low Dimension and High Dimension.** At the beginning of this chapter, we define "low dimension" as 1-dimension, 2-dimension or 3-dimension, and define "high dimension" as "4 or more dimensions". We define 1,2,3-dimension as "low dimension" because we live in a 3-dimensional world, hence we can analyze and visualize these data without losing information. We just need to present and interpret information of low dimensional data, which is drastically different from that of high dimensional data. For high dimensional data, we have to choose the most important dimensions to present.

2 Low dimensional Data Visualization

Low dimensional data visualization means to visualize data which has 3 or less dimensions. In most cases, low dimensional data visualization doesn't need complicated pre-processing or transformations. Based on the data characteristics, low dimensional data visualization displays the variable relationships, frequencies (distributions), or group comparisons. Here we mainly focus on statistics methods and will discuss it by the number of data dimensions.

2.1 Data Sets

In low dimensional data visualization, if not specially mentioned, we use the Iris data set as sample data set. Iris data set is a multivariate data set with 150 instances and 5 variables. The 5 variables are Sepal length, Sepal width, Petal length, Petal width and Species. (Source: https://archive.ics.uci.edu/ml)

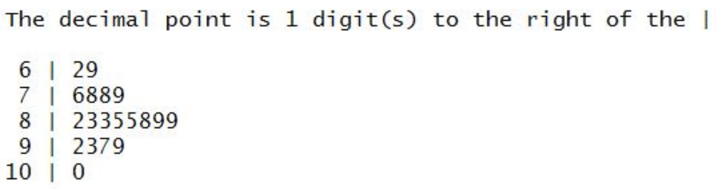
2.2 1D Data Visualization

2.2.1 Stem-and-leaf Plot

A stem-and-leaf plot is a simple and intuitive data visualization method. It shows the frequency of values in a plain and simple way: it sets a table which split each data value into 2 parts. The first part is the first digit or digits, representing the "stem" of this value, the second part is the last digit, representing the "leaf" of this value. Stem-and-leaf plot is a way of showing the frequency of 1-d numerical data. The input of the data will be rounded to the leaf unit. It provides information about the frequency of the 2nd (last) digit: the stem with longest leaf has highest frequency. Stem-and-leaf plot can be done easily and manually, which explains why it was widely used in last century.

- **Example 1**

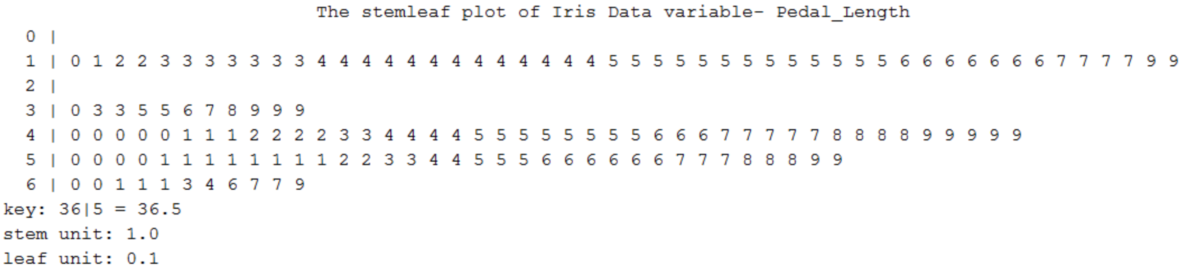
  ∗ Data Set: Midterm scores of a class
  (78, 62, 99, 69, 83, 85, 78, 88, 93, 97, 100, 76, 89, 85, 92, 89, 83, 79, 82)
  ∗ Stem and leaf plot



```
The decimal point is 1 digit(s) to the right of the |

 6 | 29
 7 | 6889
 8 | 23355899
 9 | 2379
10 | 0
```

Figure 1: Stem-and-leaf plot shows scores between 80 and 89 have most frequency.

- **Example 2:** Stem-and-leaf plot for Iris Data Set



```
                              The stemleaf plot of Iris Data variable- Pedal_Length
  0 |
  1 | 0 1 2 2 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 7 7 7 7 9 9
  2 |
  3 | 0 3 3 5 5 6 7 8 9 9 9
  4 | 0 0 0 0 0 1 1 1 2 2 2 2 3 3 4 4 4 4 5 5 5 5 5 5 5 5 6 6 6 7 7 7 7 7 8 8 8 8 9 9 9 9 9
  5 | 0 0 0 0 1 1 1 1 1 1 1 1 2 2 3 3 4 4 5 5 5 6 6 6 6 6 6 7 7 7 8 8 8 9 9
  6 | 0 0 1 1 1 3 4 6 7 7 9
key: 36|5 = 36.5
stem unit: 1.0
leaf unit: 0.1
```

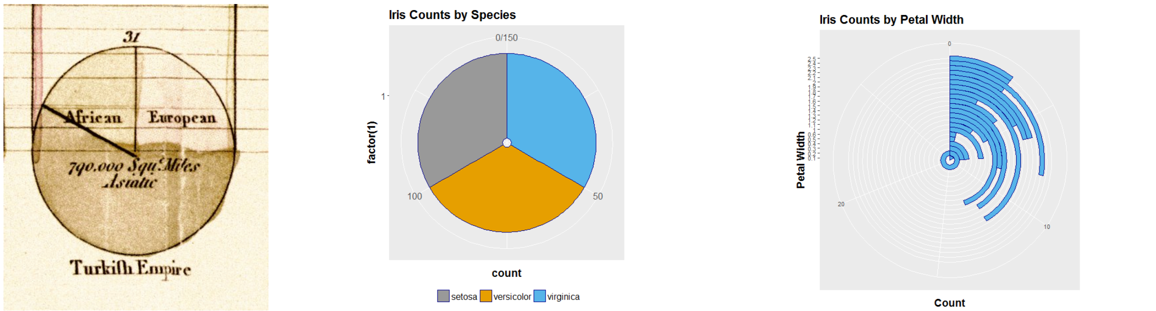
Figure 2: Stem-and-leaf Plot of Pedal Length

Figure 2 shows stem value 1 has the longest leaf length, which means most instances have pedal length in range [1,2), when compared to range [0,1), [2,3), [3,4) and [4,5).

2.2.2 Pie Chart

The earliest known pie chart is generally credited to William Playfair's Statistical Breviary of 1801. A pie chart is a circular statistical graph which is divided into slices to illustrate numerical proportion. In a pie chart, the central angle and area are proportional to the quantity it represents. Pie chart doesn't show the distributions, trend or values of a variable. It shows the frequency proportion of different values of a given variable, so it is not applicable to continuous data. Most often it is used in a variable which has a few distinct values, no matter those are discrete numerical or categorical.

Pie chart is not widely used in statistics, since its main use is to present frequency proportion. It can be easily replaced by other plotting methods such as bar chart, which can provide value and trend information besides frequency proportion.



(a) Hand Written Pie Chart, 1801

(b) Pie Chart of Iris Species

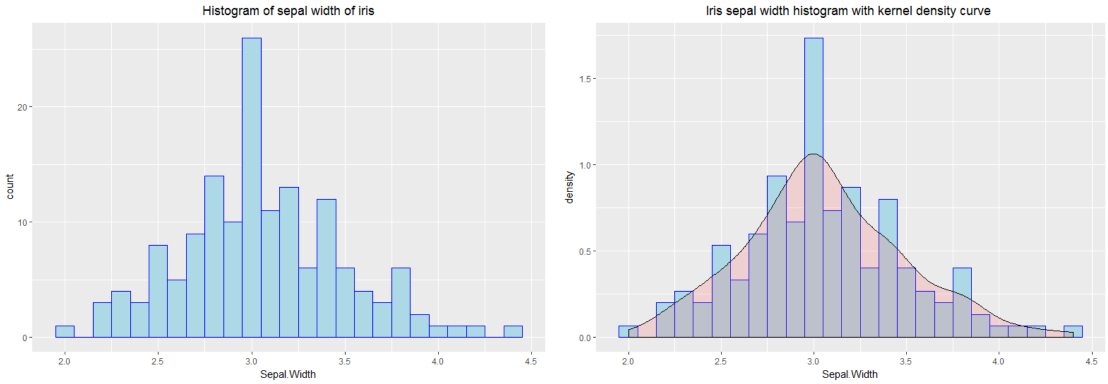
(c) A Pie Chart Alternative of Iris Species

Figure 3: Pie Chart Examples

Figure 3(b) shows the frequencies of categorical variable Iris Species; Figure 3(c) is an alternative form of pie chart, and it shows the frequencies of discrete variable Iris Petal width.

2.2.3 Histogram

A histogram is a widely used visualization method that shows the underlying frequency distribution (shape) of a continuous variable. This allows the inspection of the data for its underlying distribution, outliers, skewness, etc. To construct a histogram, the first step is to "bin" the range of values. That is, divide the entire range of values into a series of intervals and then count how many values fall into each interval. The bins are usually specified as consecutive and non-overlapping of a variable. The bins must be adjacent, and are often (but are not required to be) of equal size.



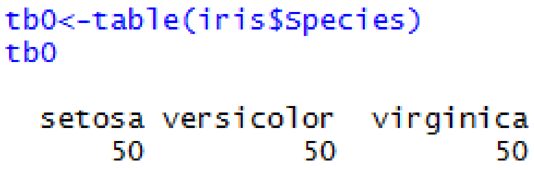
(a) Histogram with Count (b) Histogram with Kernel Density Curve

Figure 4: Histograms of Iris Sepal Width

In Figure 4(b), the erected rectangle area of histogram may also be defined to be proportional to the frequency of cases in the bin. The vertical axis is then not the frequency but frequency density. It can give us a rough sense of the density of the underlying distribution of the data. The total area of a histogram used for probability density is always normalized to 1.Figure 4(b) also shows when the histogram presents the density of data, it can be thought of as a kernel density approximation, which uses a kernel to smooth frequencies over the bins. This yields a smooth probability density curve.

2.2.4 1-way Table

The 1-way table is a tool to show the frequency of cases of a categorical variable. Figure 5 is the 1-way table of variable Species in the Iris data set.



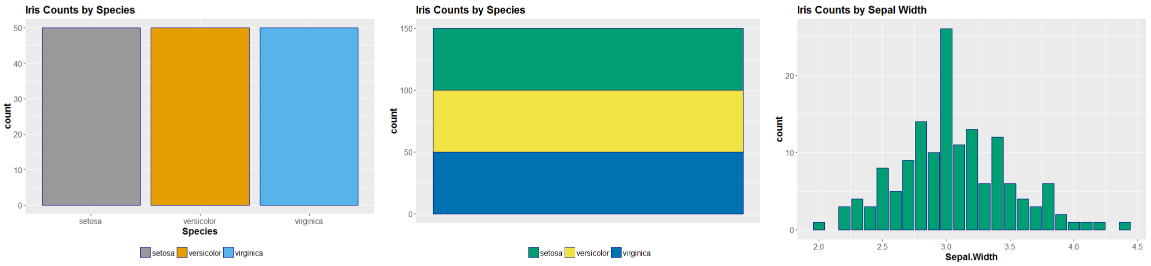
```
tb0<-table(iris$Species)
tb0

    setosa versicolor  virginica 
        50         50         50 
```

(a) 1-way Table

Figure 5: 1-way table shows the frequency of any 1 of 3 subspecies of Iris data set is 50.

2.2.5 1D Bar Chart

Bar chart is also called bar plot. The 1D bar chart is the basic edition of bar chart. It shows the frequency of a categorical variable or a discrete numerical variable. It is similar to a histogram, while the major difference between them is that bar chart shows the values or frequencies of non-continuous data while histogram shows the frequency of continuous data. The bar chart is also similar to the 1-way table in that it shows the frequency of a categorical or a discrete variable, while also being visually more straightforward. The bars can be plotted vertically or horizontally. A vertical bar plot is sometimes called a line graph or a stacked bar chart.



(a) 1D Bar Chart by Species (b) 1D Stacked Bar Chart (c) 1D Bar Chart by Sepal Width

Figure 6: 1D Bar Charts of the Iris Data Set

Figure 6(c) shows the bar chart of the sepal width of Iris data set. Here we treat the sepal width as a discrete variable and divide the whole range into equal-length intervals. Some bar charts are clustered in groups.

These bar charts can explain 2-dimensional information. We will discuss them later.

2.2.6 1D Box Plot

In descriptive statistics, a box plot is a convenient way of graphically depicting groups of numerical data through their quantiles. Box plots may also have lines extending vertically from the boxes (whiskers) indicating variability outside the upper and lower quantiles, hence the terms box-and-whisker plot. Box plots are non-parametric: they display variation in samples of a statistical population without making any assumptions of the underlying statistical distribution. The space between the different parts of the box indicate the degree of dispersion (spread) and skewness in the data, and show outliers. In addition to the points themselves, they allow one to visually estimate various estimators, such as inter-quantile range and and mean. 1D box plot is the basic form of box plot. It is composed of 1 box or 1 numerical variable. It gives us estimated sense of mean and quantiles about this variable. Box plots are usually used to do variable comparisons and the 1D box plot is seldom used by itself.

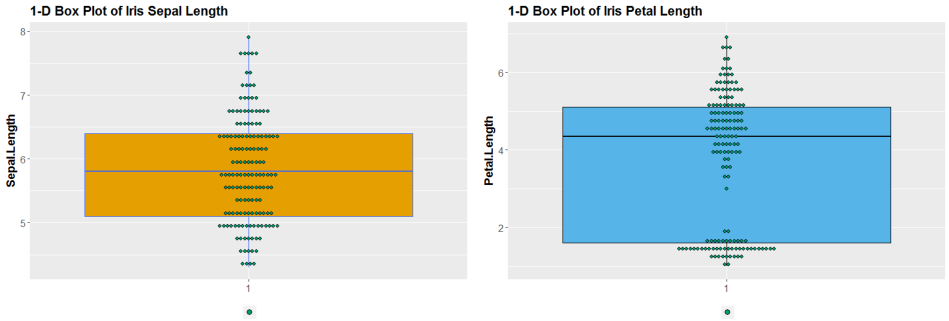
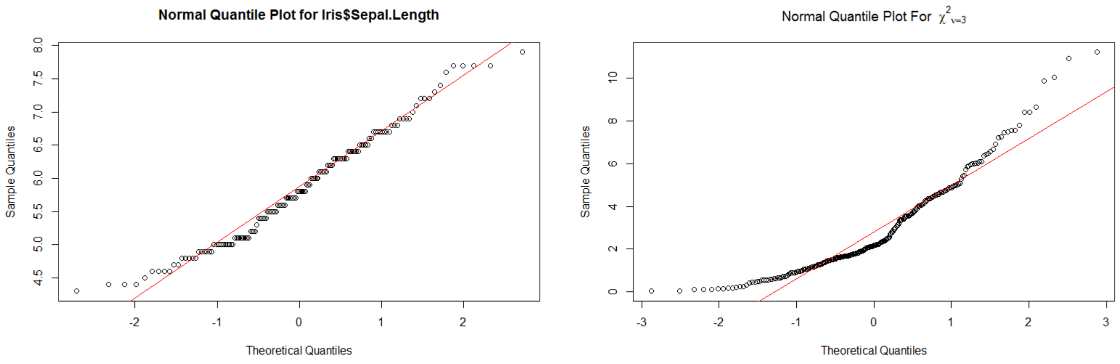


Figure 7: 1D Box Plots of Iris Data Set

2.2.7 1D Normal Quantile Plot

Normal Quantile Plot is also called Quantile-Quantile plot or Q-Q plot. It is a method of observing the distribution of a 1-dimensional quantitative continuous data and is widely used in descriptive statistics. By normal quantile plot we compare the distribution of sample data and the normal (Gaussian) distribution. The normal quantile plot from data extracted from a normal distribution is like a straight line composed of dots; the scatter plot from data which don't follow normal distribution is a curve composed of dots.

Normal quantile plot gives us the first instinct if sample data seems to follow a normal distribution. Quantile-quantile plot can be extended to compare sample data and other distributions.

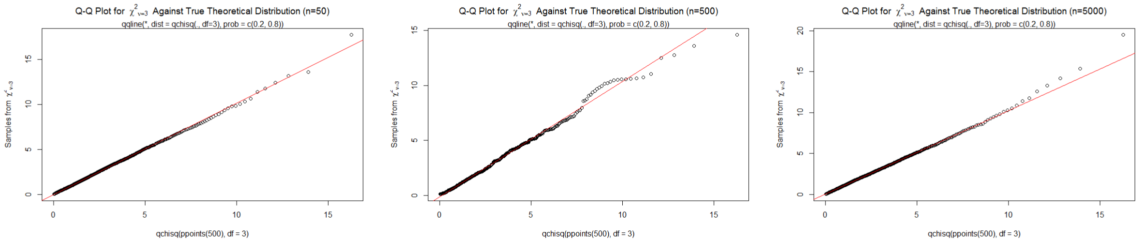


(a) Normal Quantile Plot of the Sepal Length

(b) The Sepal Length V.S. χ^2distribution

Figure 8: Normal Quantile Plots of the Iris Data Set

The plot in Figure 8(a) is like a straight line, which indicates that we may reasonably assume sepal length follows a normal distribution; the plot in Figure 8(b) is like a curve, which means sepal length doesnt́ follow a χ^2 distribution.
Sample size affects the comparison effect. We extracted given numbers (50, 500, 5000) of sample from χ^2 distribution and compare them with true theoretical distribution.



(a) Quantile-Quantile Plot for χ^2 distribution (n=50)

(b) Quantile-Quantile Plot for χ^2 distribution (n=500)

(c) Quantile-Quantile Plot for χ^2 distribution (n=5000)

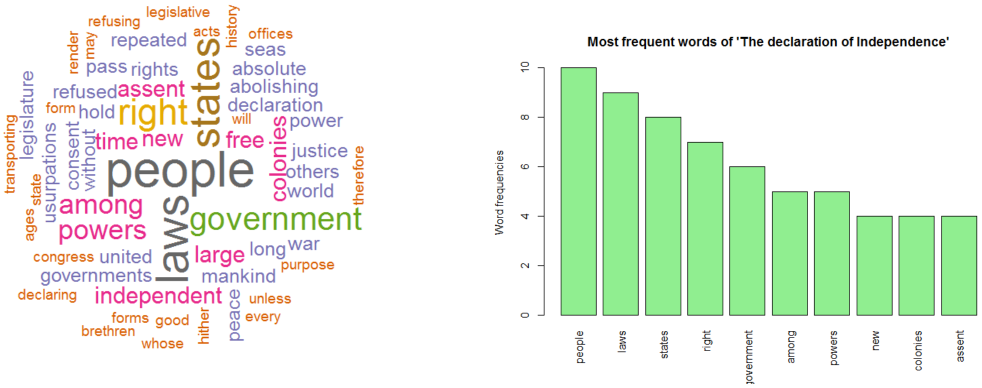
Figure 9: The Samples form χ^2 Distribution V.S. theoretical χ^2 Distribution

Figure 9 indicates that the more samples extracted, the higher similarity there is between the sample distribution and true theoretical distribution. Q-Q plot can also be used to compare any two continuous distributions. We will discuss this further in the 2D data visualization section.

2.2.8 Text Cloud

Text cloud is a method that uses different size for each word according to their frequencies. The size of the presented word is positively related to the frequency of that particular word used in a text document. It is a technique that shows the topic of the target document. We can easily get a sense of the topic or emphasize of the document since the most used words stand out. It is simple and clear.

Data set : "The Declaration of Independence" .
(Source: https://www.archives.gov/founding-docs/declaration-transcript)



(a) The Text Cloud of ”The Declaration of Independence”

(b) The Word Frequency of ”The Declaration of Independence”

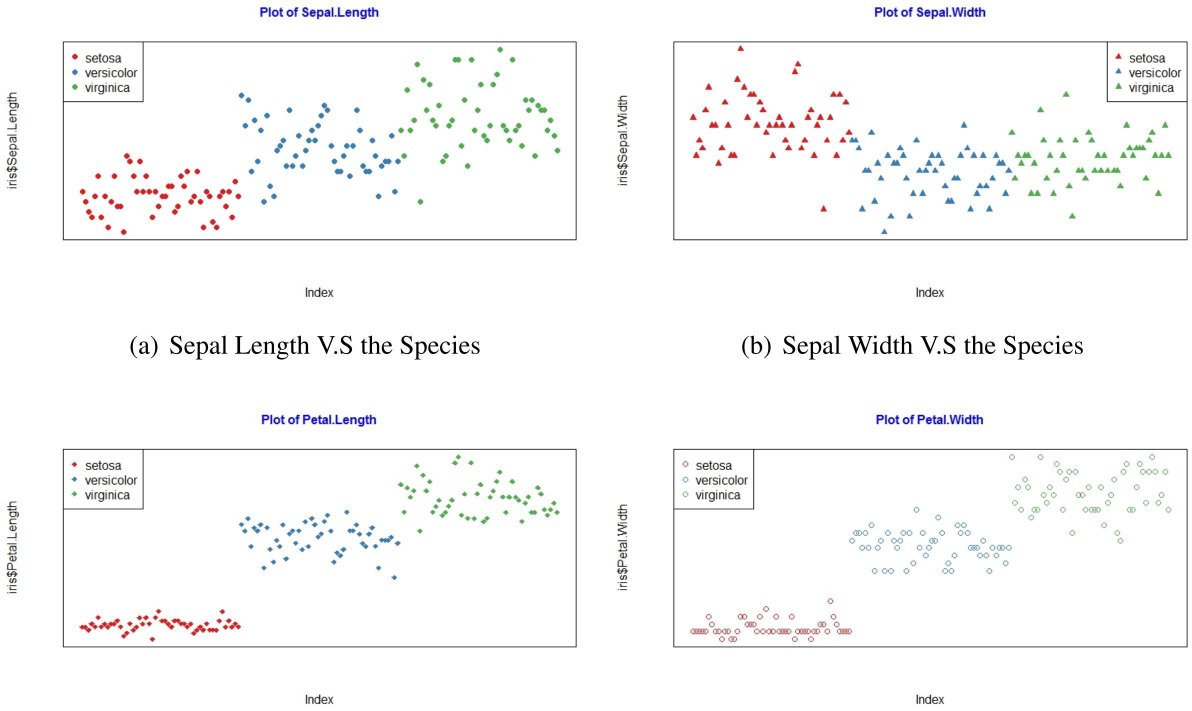
Figure 10: Text cloud gives us a quick impression of an article.

The most frequently used keywords include: people, law, states, right and government. Figuring out the topic via text cloud is much faster than reading through the target text file.

2.3 2D Data Visualization

2.3.1 2D Scatter Plot

2D scatter plot is a widely used graphic technique showing the relationships between two variables. In scatter plot, each dot or small circle represents an observation. All the points in the plot form a visual sample distribution, giving us statistical insight about the variable relationships.



(a) Sepal Length V.S the Species

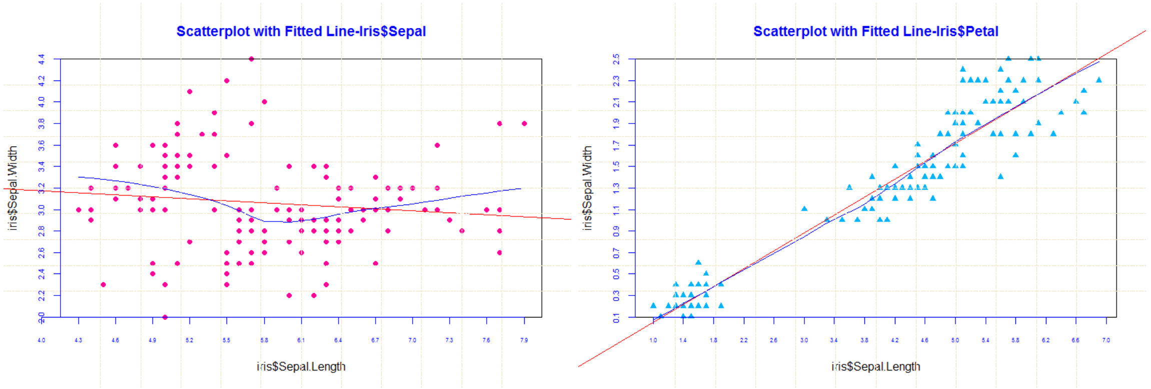
(b) Sepal Width V.S the Species

(c) Petal Length V.S Species

(d) Petal Length V.S Species

Figure 11: 2D Scatter Plots of the Iris Data Set

Figure 11 shows the scatter plots between categorical variable Species and numerical variables, respectively. Scatter plots with fitted lines see Figure 12.



(a) Sepal Length V.S. Sepal Width

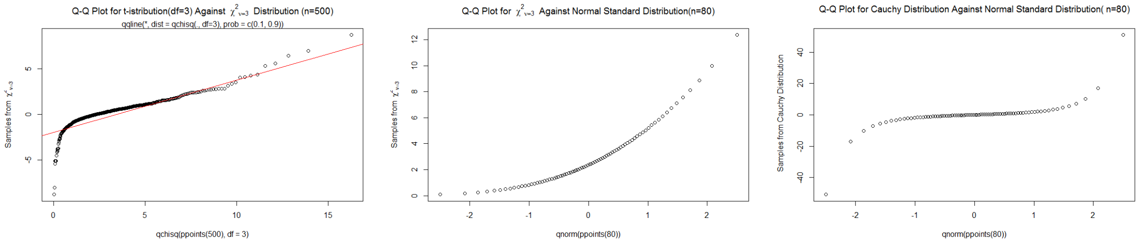
(b) Petal Length V.S. Petal Width

Figure 12: 2D Scatter Plots with Fitted Lines

From Figure 12 we reasonably assume the Iris$Sepal.Width and the Iris$Sepal.Length don't have apparent linear relationship; while the Iris$petal.Width and the Iris$Petal.Length are linear related.

2.3.2 2D Quantile-Quantile Plot

As we mentioned before, Quantile-Quantile plot (Q-Q plot) can also be used to compare any two continuous distributions. Instead of comparing targeted variable to standard normal distribution, we can compare it to other variable. By observing the scatter plot, we will have instinct sense about if the two variables may or may not follow same distribution. If they have approximately linear linear relationship, the Q-Q plot is like a straight line, or else it will be like a curve. Q-Q plot is very helpful in exploring the distributions of continuous variables.

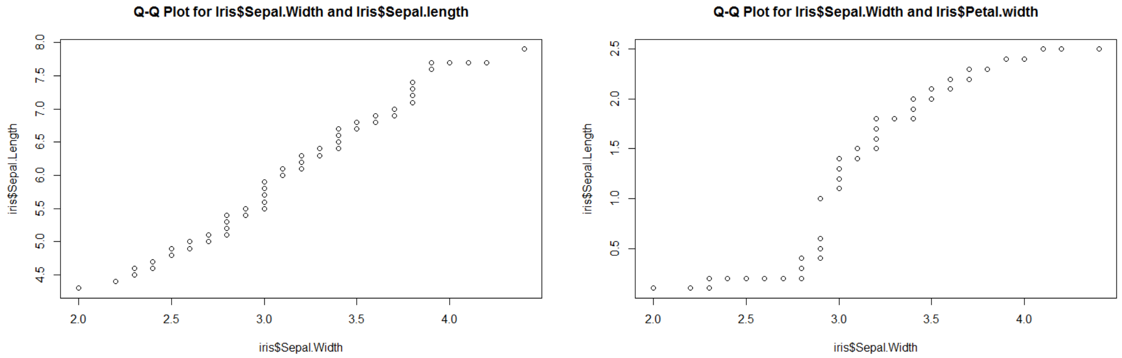


(a) t V.S. χ^2 Distribution

(b) χ^2 V.S. Normal Distribution

(c) Cauchy V.S. Normal Distribution

Figure 13: Q-Q plots from 2 different distributions are curves.



(a) Iris$Sepal.Length against Sepal.Width

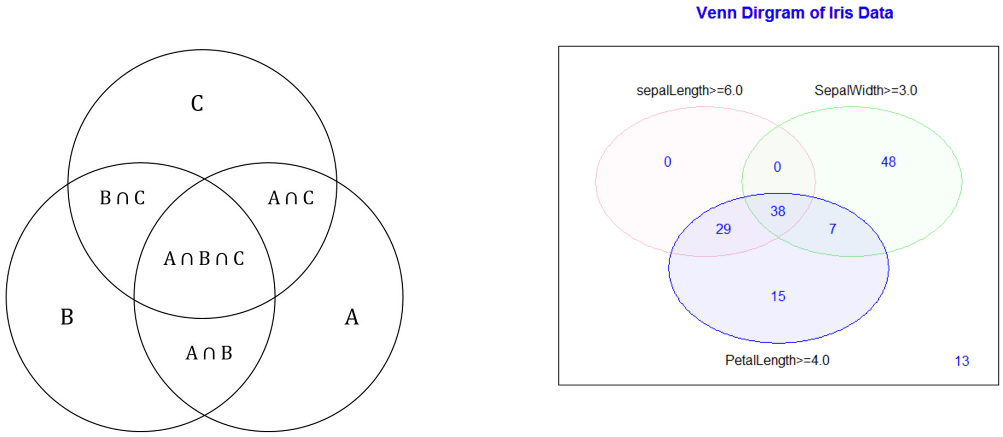
(b) Iris$Petal.Length against Petal.Width

Figure 14: Q-Q Plots for the Iris Data Set

Figure 14 shows the sepal length and the sepal width may follow the same distribution, but the petal length and the petal width may not.

2.3.3 2D Venn Diagram

A Venn diagram shows a qualitative relationship of a given variable or several given variables among different groups. It is often used to check if different groups have same categorical feature or features. This gives us a quick and straightforward presentation about the overlap on given features among different groups.



(a) General Diagram

(b) Venn Diagram of the Iris Data Set

Figure 15: Venn Diagram Examples

2.3.4 2D Box Plot

2D box plot is also called side-by-side box plot. It is a visual display comparing the distributions of different levels (the possible values) of a categorical variable, or different ranges of a numerical variable. Side-by-side boxplot is constructed by placing single box plots adjacent to one another on a single scale.

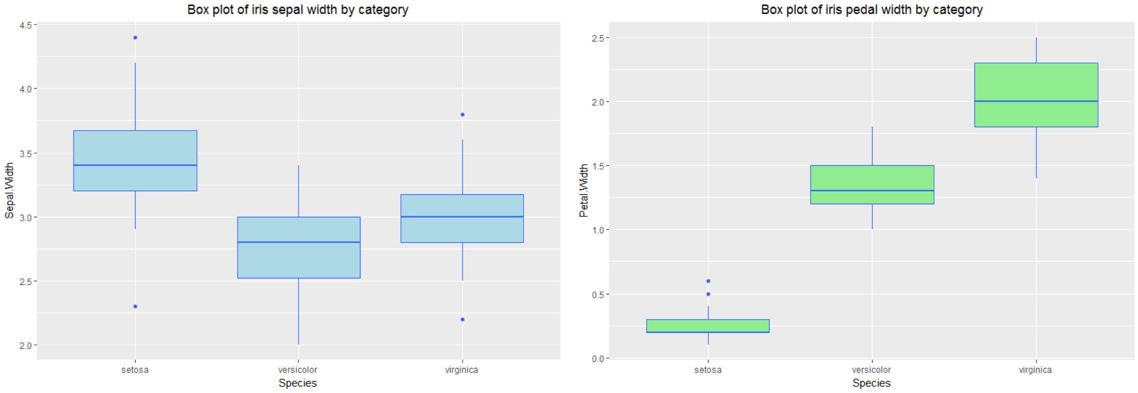
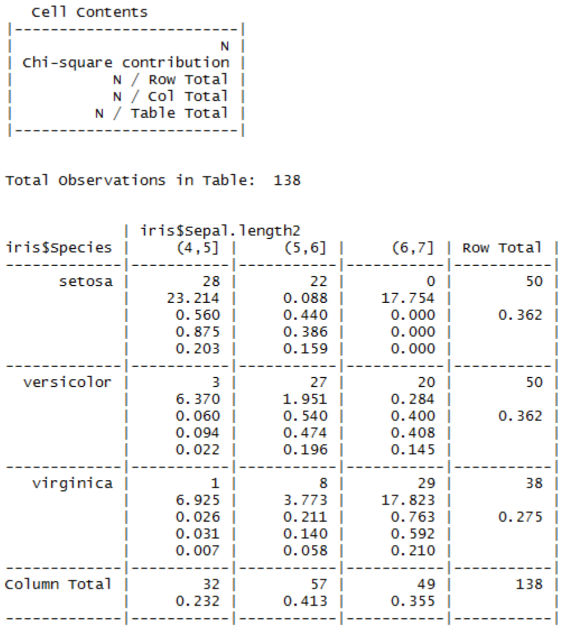


Figure 16: Box Plots by the Species

2.3.5 Two-way Table

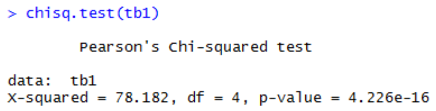
In Statistics, a two-way table (also called contingency table, or cross tabulation) is a useful tool for examining relationships between categorical variables. The entries in the cells of a two-way table can be frequency counts or relative frequencies (just like a one-way table). They are heavily used in survey research, business intelligence, engineering and scientific research. They provide a basic picture of the interrelation between two variables and can help find interactions between them. Iris only has one categorical variable, so we transformed continuous variable sepal length to categorical data with 3 length levels, then have the contingency table as following:



```
   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  138


             | iris$Sepal.length2
iris$Species |     (4,5] |     (5,6] |     (6,7] | Row Total |
-------------|-----------|-----------|-----------|-----------|
      setosa |        28 |        22 |         0 |        50 |
             |    23.214 |     0.088 |    17.754 |           |
             |     0.560 |     0.440 |     0.000 |     0.362 |
             |     0.875 |     0.386 |     0.000 |           |
             |     0.203 |     0.159 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
  versicolor |         3 |        27 |        20 |        50 |
             |     6.370 |     1.951 |     0.284 |           |
             |     0.060 |     0.540 |     0.400 |     0.362 |
             |     0.094 |     0.474 |     0.408 |           |
             |     0.022 |     0.196 |     0.145 |           |
-------------|-----------|-----------|-----------|-----------|
   virginica |         1 |         8 |        29 |        38 |
             |     6.925 |     3.773 |    17.823 |           |
             |     0.026 |     0.211 |     0.763 |     0.275 |
             |     0.031 |     0.140 |     0.592 |           |
             |     0.007 |     0.058 |     0.210 |           |
-------------|-----------|-----------|-----------|-----------|
Column Total |        32 |        57 |        49 |       138 |
             |     0.232 |     0.413 |     0.355 |           |
-------------|-----------|-----------|-----------|-----------|
```

Figure 17: The Contingency Table of the Iris Data Set

• **Pearson's Chi-squared Test.** Based on contingency table, χ^2 test is often used to test if the two categorical variables related.
Hypothesis: The species and the sepal length are independent at 0.05 significant level.



```
> chisq.test(tb1)

        Pearson's Chi-squared test

data:  tb1
X-squared = 78.182, df = 4, p-value = 4.226e-16
```

Figure 18: The χ^2 Test of Species and Sepal Length Level

The P-value is far more less than 0.05, so we reject the null hypothesis and think the sepal length is not independent from the species of the Iris at 0.05 significant level.

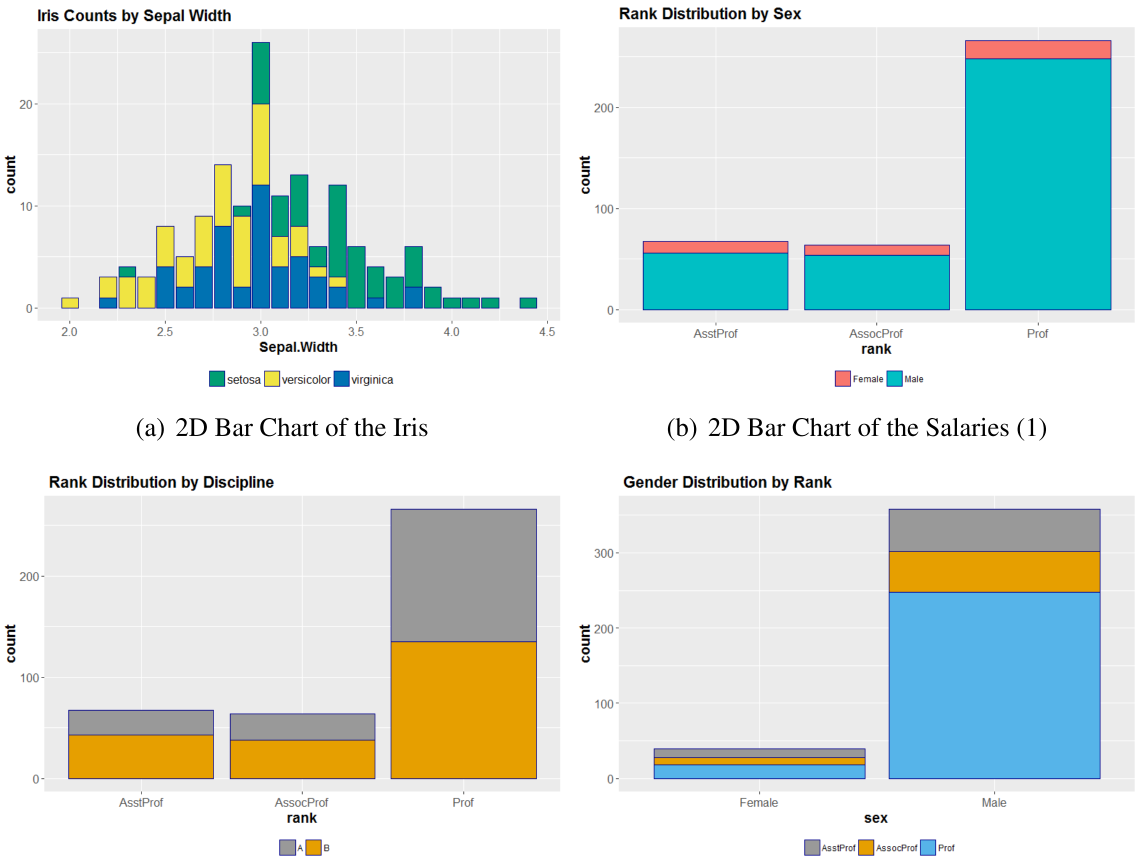
2.3.6 2D Bar Chart

Bar chart can display 2D information, where one variable should be categorical or numerical data range, the other one can be categorical or numerical. If one of the variables is numerical, each categorical value should have only one corresponding numerical value, so we usually use it to compare the mean value and the confidence interval of different categorical levels. The 2D bar chart includes stacked bar chart and grouped bar chart. They basically display the same information except stacked bar chart stacks grouped information in one bar, while grouped bar chart display grouped information horizontally.

•Data Sets.

1. Iris.
(Source: https://archive.ics.uci.edu/ml)
2. Salaries.
(Source: R data set from “car”package. Avaliable at https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/car/Salaries.csv)
3. $S\&P$ Case-Shiller Home Price Index.
(Source: Avaliable at https://www.econ.yale.edu/~shiller/data.htm)

• 2D Bar Chart with 2 Categorical Variables



(a) 2D Bar Chart of the Iris

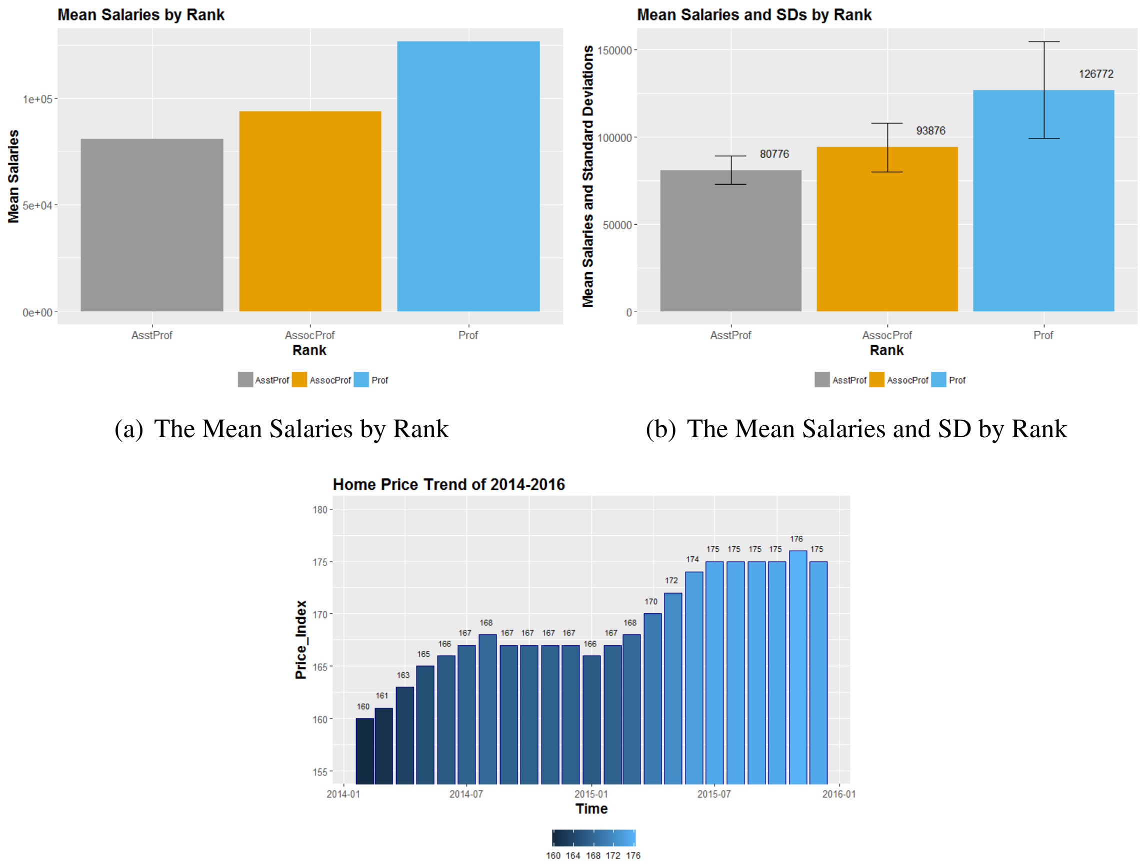
(b) 2D Bar Chart of the Salaries (1)

(c) 2D Bar Chart of the Salaries (2)

(d) 2D Bar Chart of the Salaries (3)

Figure 19: 2D bar charts with 2 categorical variables show the frequencies.

• 2D Bar Chart with 1 Categorical Variable and 1 Numerical Variable



(a) The Mean Salaries by Rank

(b) The Mean Salaries and SD by Rank

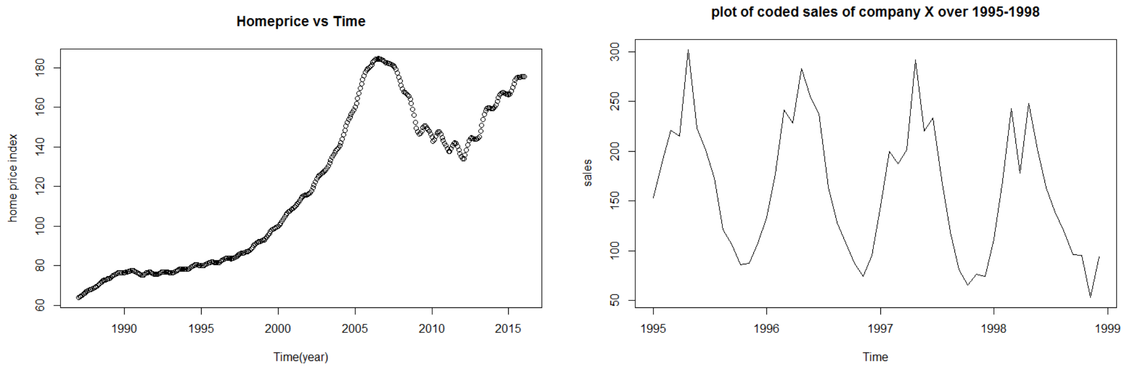
(c) The Home Price Index

Figure 20: 2D Bar Charts with 1 Categorical Variable and 1 Numerical Variable

In Figure 20 we observe the trend of time series data, or explore the mean and standard deviation by groups. With enhanced techniques, stacked and grouped bar chart can display 3D information. We will discuss this later.

2.3.7 Time Series Plot

A time series plot is a graph that shows the variable's value change over time. Based on the plot, we may have intrinsic sense of the value change pattern: Is this variable stationary or not ? Does the sample have seasonal fluctuation? A time series plot displays observations on the y-axis against equally spaced time intervals on the x-axis. Here 'time' is treated as a discrete quantitative variable.

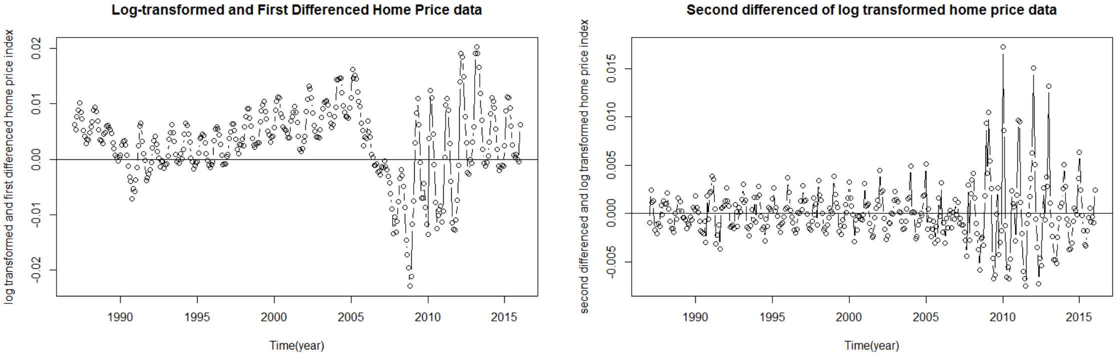


(a) The House Price Time Series Plot (b) The Sales Time Series Plot

Figure 21: Time Series Plots

Figure 21(a) contains monthly-measured average index home values throughout the USA from January 1987 to January 2016 (Source: Available at https://www.econ.yale.edu/~shiller/data.htm). Figure 21(b) is from a sales record and it shows apparent seasonal pattern.

Much of time series theory assumes that time series data is a stationary process, which means the parameters such as mean and variance, if they are present, do not change over time. In reality, this is not always the truth, hence we need transform non stationary data to stationary data, to make time series data analysis feasible. For example, we found that S&P/Case-Shiller Home Price Index is not stationary over time, so we make 2-step transformation.



t

(a) Logged S&P/Case-Shiller Home Price Index First Transformation (b) Logged S&P/Case-Shiller Home Price Index Second Transformation

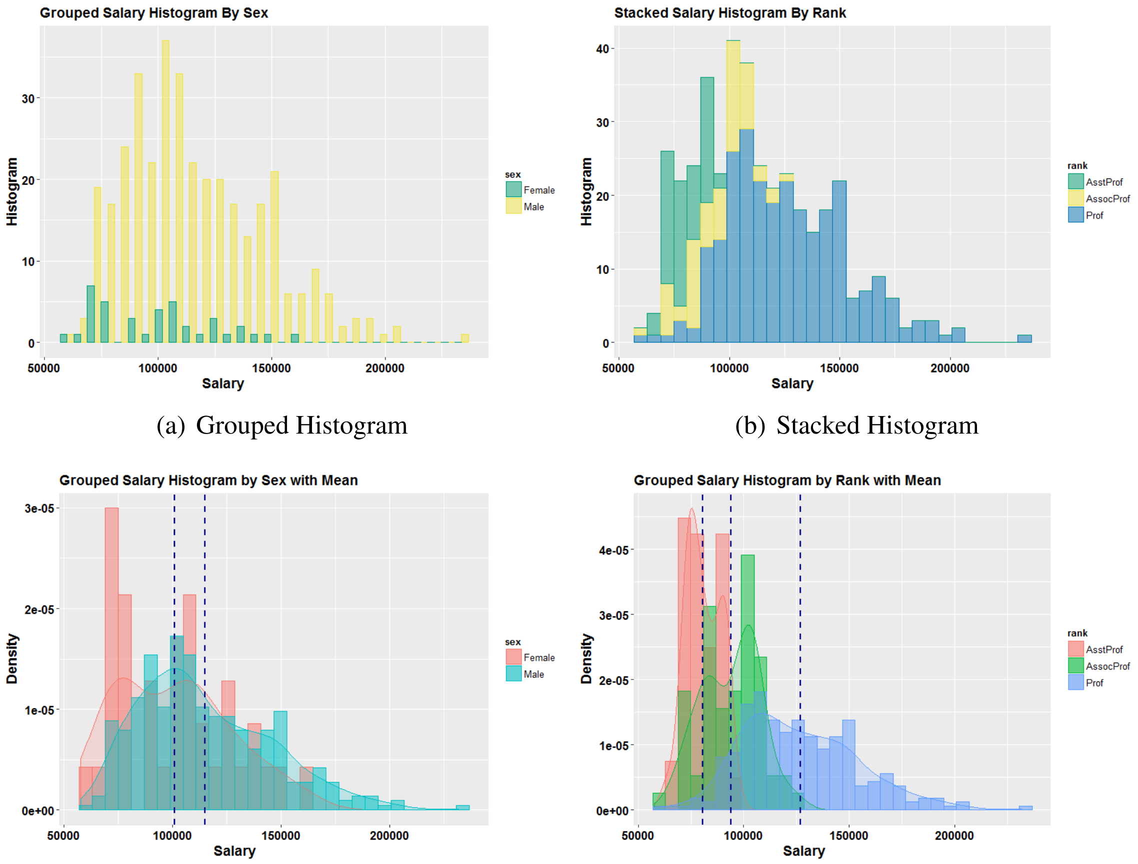
Figure 22: Time Series Data Transformations

After logged and the first difference, the data fluctuation is eased a lot but the estimated mean is still not steady enough, so we did the second difference, which makes the processed data trend keeps horizontal, in other words, makes the expected mean unchanged over time.

2.3.8 Grouped Histogram

Grouped histogram is extended from 1D histogram. It presents the frequency or density of a variable by group, extending the explained information from 1-d to 2-d. It is a intuitive tool to observe the distributions of different groups.

•**Data Set.** Salaries (Source: R data set from “car”package).



(a) Grouped Histogram

(b) Stacked Histogram

(c) Histogram with Means and Density

(d) Histogram with Means and Density

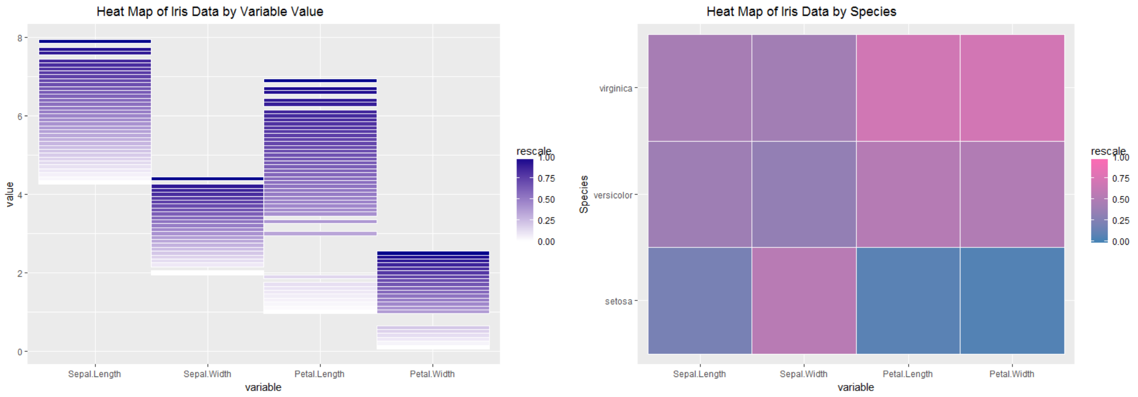
Figure 23: Grouped Histograms for the Salaries Data Set

Figure 23 shows the salary comparison between different groups. We can see that woman has a smaller sample size and overall lower salary compared to men. The salaries of assistant professors, associate professors, and professor are quite different, with professors having a much wider salary range and significantly higher earnings overall.

2.4 3D Data Visualization

2.4.1 Heat Map

A heat map is a data visualization technique which demonstrates data by colors. In most cases, the data are placed in a 2-dimensional coordinate system. To some extent, it is like a colorful map and different colors represent different values of data. It provides an immediate visual summary of information. It is a popular method to show dynamic data such as daily traffic flow in a given area, or the house price of a given district. If we observe the timeline of the heat maps in a given district, we may find out some superficial trend. For example, we can found out the daily traffic trend by watching the color change of a series of traffic heat map. Enhanced heat map can express multiple dimension information $(n \geq 3)$.The color of heat map represents the values of discrete data, an interval of continuous data or categorical data.



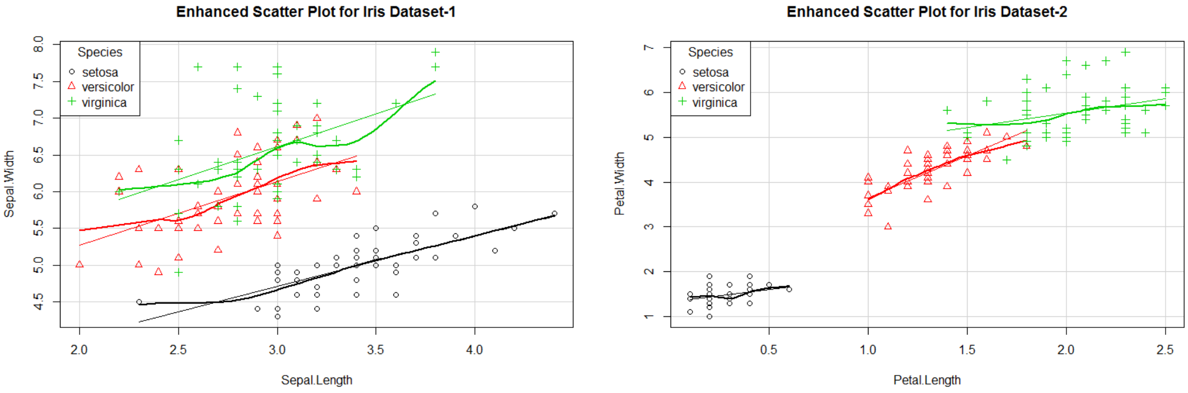
(a) Heat Map of the Iris by Variable Values

(b) Heat Map of the Iris by Species

Figure 24: Heat Maps of the Iris Data Set

2.4.2 Enhanced 2D Scatter Plot

Enhanced 2D scatter plot is also called grouped scatter plot. In general, 2D Scatter plot is a graph showing the joint distribution of data points in 2-dimensional space, but under some enhancement, 2-D Scatter plots may represent 3-dimensional or more dimensional information. For example, under each categorical level, we may have a 2D plot. If we combine all those plots in one graph, we may have a more well-rounded understanding of two numeric variables and categorical variable.



(a) Enhanced Scatterplot-1

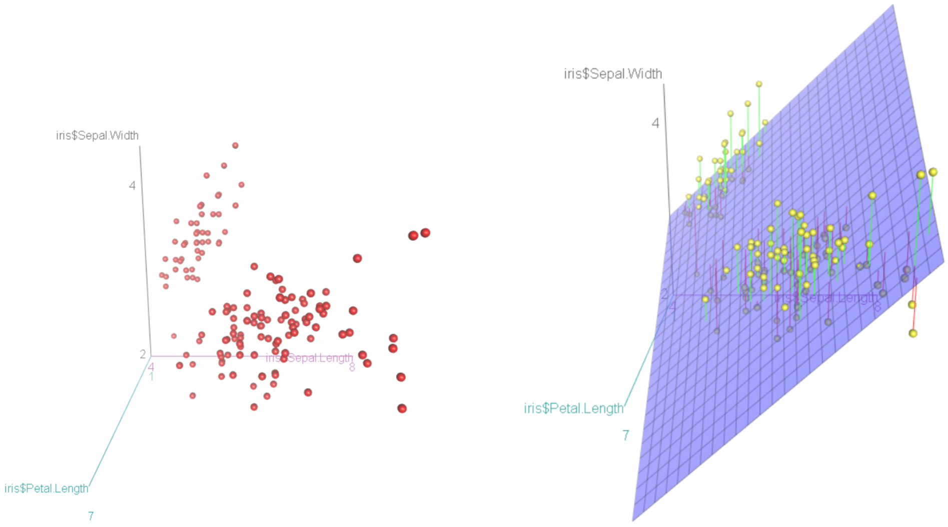
(b) Enhanced Scatterplot-2

Figure 25: Enhanced Scatter Plots of the Iris Data Set

From Figure 25 we may estimate that sepal length is independent with species, petal length is dependent with species, and that the petal length and petal width have positive correlation.

2.4.3 Scatter 3D plot

Scatter 3D plot is a visualization method to present the values of 3 variables and exploring the relationships between the 3 variables. Each variable is expressed by an axis, based on three axis in a 3D space. In general, the 3 variables are all numerical variables.



(a) 3D Scatter Plot of the Iris Data Set without Surface

(b) 3D Scatter Plot of the Iris Data Set with Surface

Figure 26: 3D Scatter Plot of the Iris Data Set without Group Information

Additionally, we can add colors to express the categorical information of those instants, which means we may actually explore 4D information through 3D scatter plots.

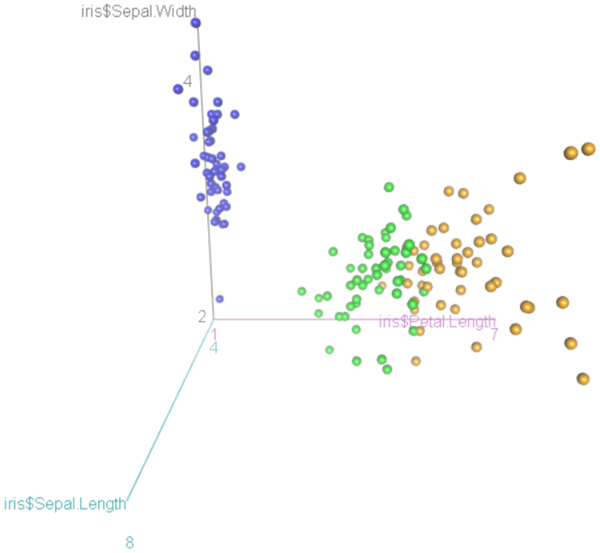
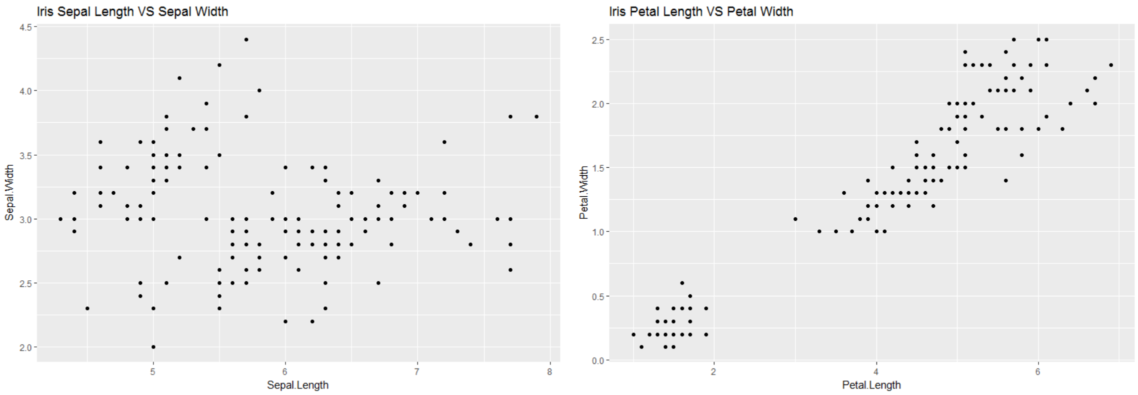


Figure 27: 3D Scatter Plot of the Iris Data Set with Group Information

2.4.4 Bubble Chart

A bubble chart is also called weighted scatter plot. It is a variation of a scatter plot. The difference between bubble chart and scatter plot is that in bubble chart points are represented by bubbles, not by same sized points or small circles; the size of bubbles represents the third dimension value: the frequency of this point, or the value of the third quantitative variable. The first dimension and the second dimension should be numerical. Under enhanced mode, it can display 4D information. We demonstrate the effect of bubble chart in the Iris data set.



(a) Original Scatterplot-1

(b) Original Scatterplot-2

Figure 28: Scatter Plots of the Iris Data Set

From Figure 28 we reasonably assume there is linear relationship between the petal length and the petal width, while the sepal length and the sepal width looks like no liner relationship, so we focus on exploring the relationships of the petal length, the sepal width and the sepal length.

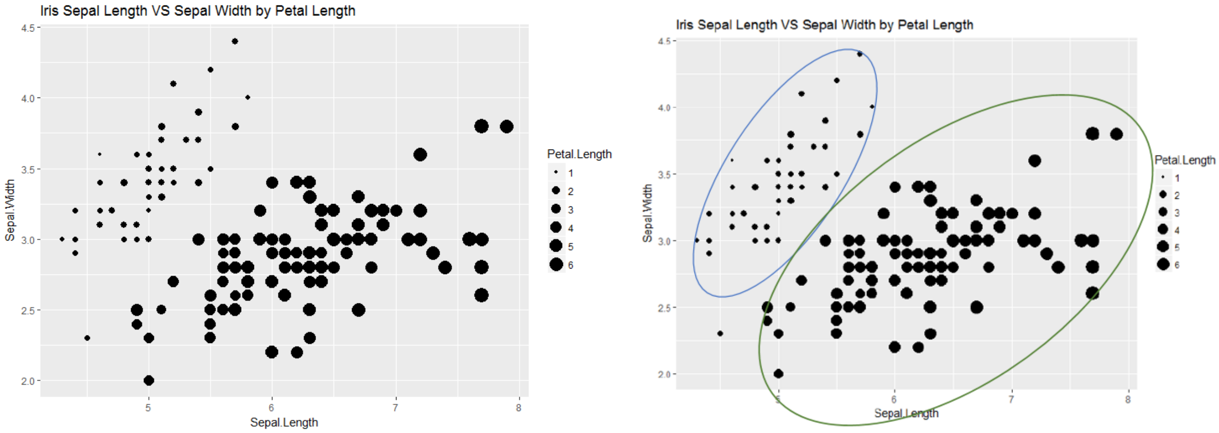
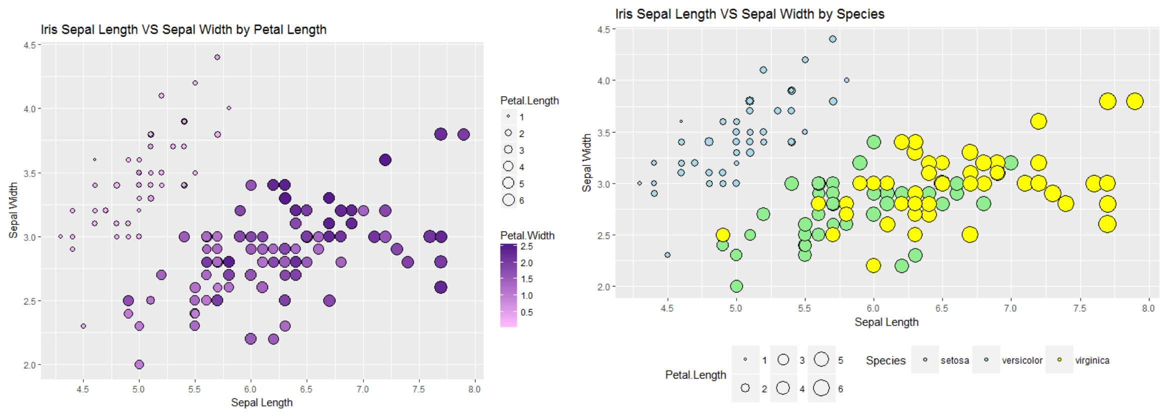


Figure 29: Bubble Charts of the Iris Data Set

Based on the size of points, the points can be naturally separated into 2 parts. Each top left point has small petal length; each bottom right point has large petal length, but this clustering looks does not depends on the sepal length or the sepal width. Let's try if it depends on other features.



(a) The Sepal Length vs the Sepal Width by the Petal Length (b) The Sepal Length vs the Sepal Width by the Species

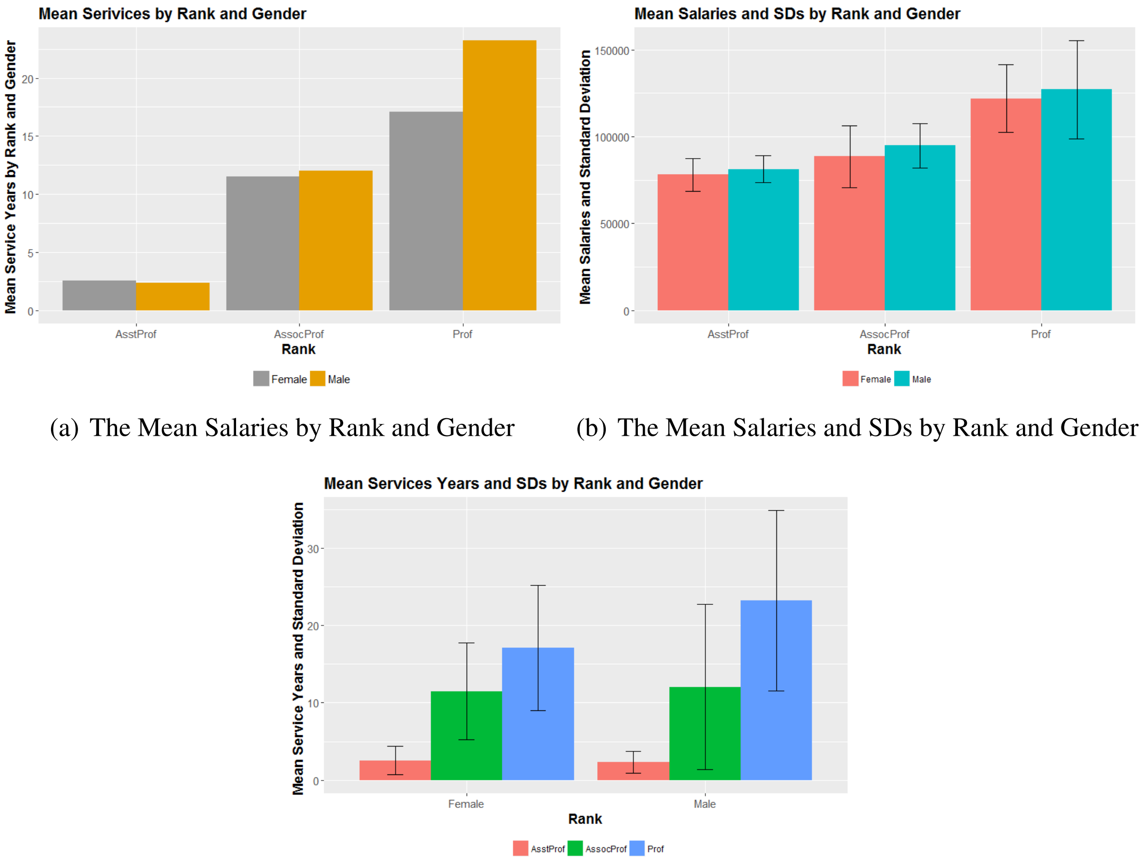
Figure 30: The Iris Clustering Exploration

Figure 30 demonstrates that the Iris clustering is not based on any numeric variables, but based on the categorical variable Species: Setosa has small petal length (and petal width), Vesicolor has large petal length and large petal width, while it doesn't hold large sepal length. Based on these characteristics, we can cluster the Iris into 3 groups.

2.4.5 Multi-set Bar Chart

Multi-set bar chart is also known as grouped bar chart or clustered bar chart. This variation of bar chart is used when two or more data sets are plotted side-by-side and grouped together under categories, all on the same axis. Like a bar chart, the length of each bar is used to show discrete, numerical comparisons among categories. Each data series is assigned an individual color or a varying shade of the same color, in order to distinguish them. Each group of bars are then spaced apart from each other. The use of multi-set bar charts is usually to compare grouped variables or categories. The downside of multi-set bar charts is that they become harder to read the more bars you have in one group.

•**Data Set:** Salaries. Data size: 397 observations with 3 numerical variables and 3 categorical variables. (Source: R data set in “car” package)



(a) The Mean Salaries by Rank and Gender

(b) The Mean Salaries and SDs by Rank and Gender

(c) The Mean Service Years by Gender and Rank

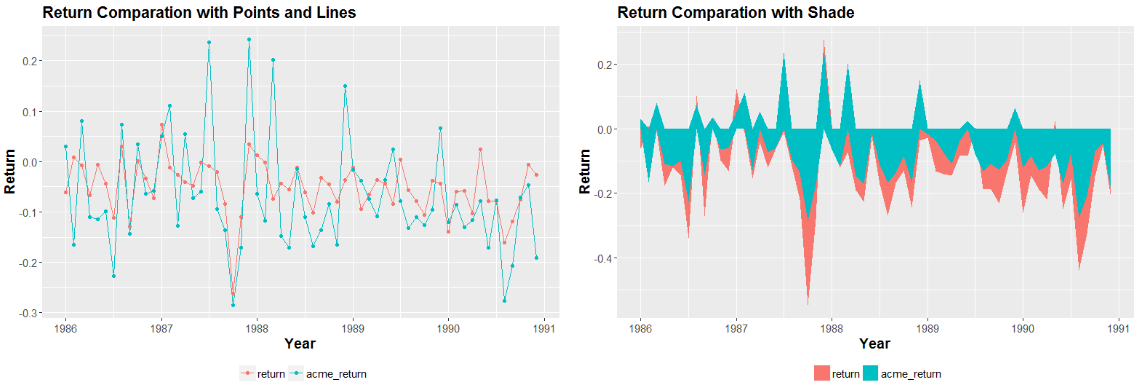
Figure 31: 3D Grouped Bar Charts

Figure(31) shows 3D information: the mean salaries of male professors are slightly higher than those of female professors based on the same rank; and it is apparent that the mean salaries of professor, associate professor and assistant professor are descending.

2.4.6 Grouped Line Chart

Line chart is an alternative to the scatter plot. The differences between line chart and scatter plot are line chart connects all points with line segments, and the points of line charts are ordered. Based on these characteristics, it is suitable for displaying points whose values change with time, such as monthly sales, stock prices, precipitations, etc. Grouped line chart shows the values trends by groups, so that we can compare the trends by groups. Grouped line charts present 3D information and require 2 numerical variables (at least one discrete variable) and 1 categorical variable. An area chart or area graph displays graphically quantitative data. It is based on line chart. The area between axis and line are commonly emphasized with colors or textures. Area charts are used to represent accumulated totals using numbers or percentages.

•**Data Set:** acme. Data size: 60 observations with 2 numerical variables and 1 categorical variable. (Source: “boot” package in R data set)



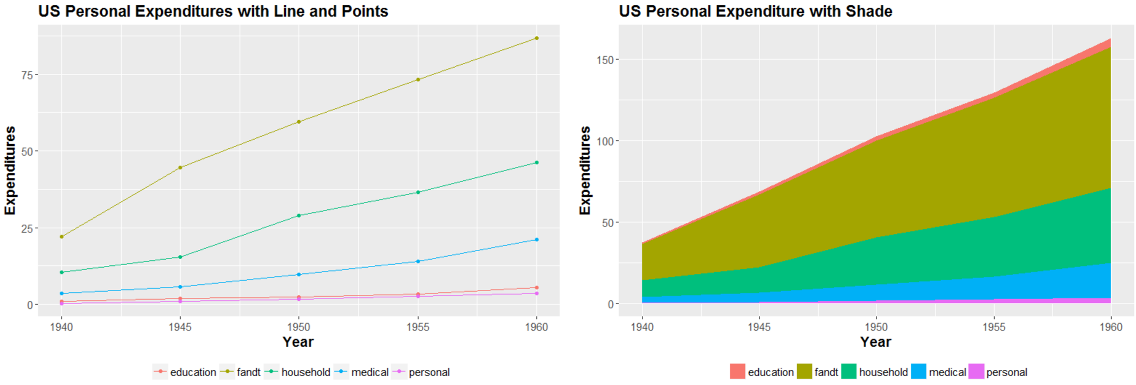
(a) The Return Comparison with Points

(b) The Return Comparison with Shade

Figure 32: The Return Comparisons

Figure 32 demonstrates that the fluctuation of acme is larger than that of market, leading us to the conclusion that acme is a more risky portfolio.

•**Data Set:** USPersonalExpenditure. Data size: 5 observations with 5 numerical variables and 1 categorical variable. (Source: “boot” package in R data set)



(a) The Personal Expenditures Structure

(b) The Personal Expenditures Area Graph

Figure 33: The Personal Expenditures Line Chart VS Area Chart

Figure 33 demonstrates people spend most money in food and tobacco, while spend least money in education and personal expenditure; and all subcategory expenditures increase a lot with time.

3 High Dimensional Data Visualization

In low dimensional data visualization, we use plot, line or bar, by grouping, coloring or bubbling to explore data intrinsic features. When the data dimensions go higher, the number of available methods reduces and the methods tend to be less effective.

3.1 High Dimensional Box Plot

We may use box plot to compare the quantiles of multiple numerical variables with same scales by putting them together.

•**Data Set:** ais. Data size: 202 observations. 11 numerical variables. 2 categorical variables. (Source: "DAAG" package in R data set.)

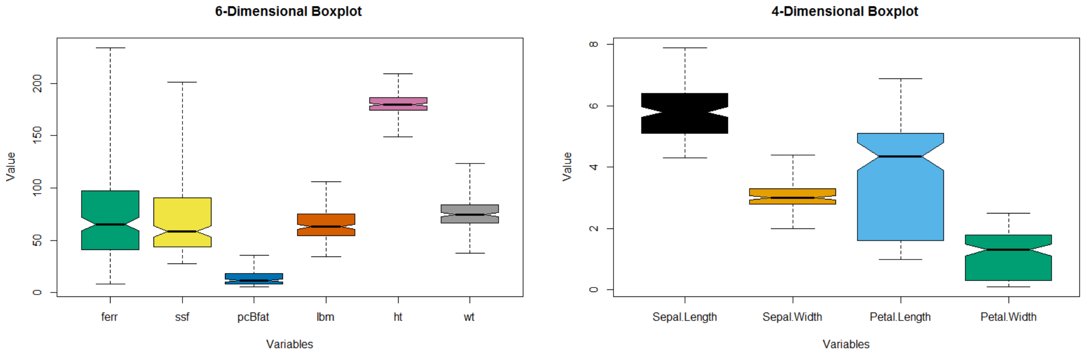


Figure 34: High Dimensional Box Plots

3.2 Matrix Scatter Plot

Matrix scatter plot is a combination of paired variable scatter plots. We made scatter plots between all numeric variables. Through this method, we can observe high dimensional data by putting the 2D plots of numerical variables together.

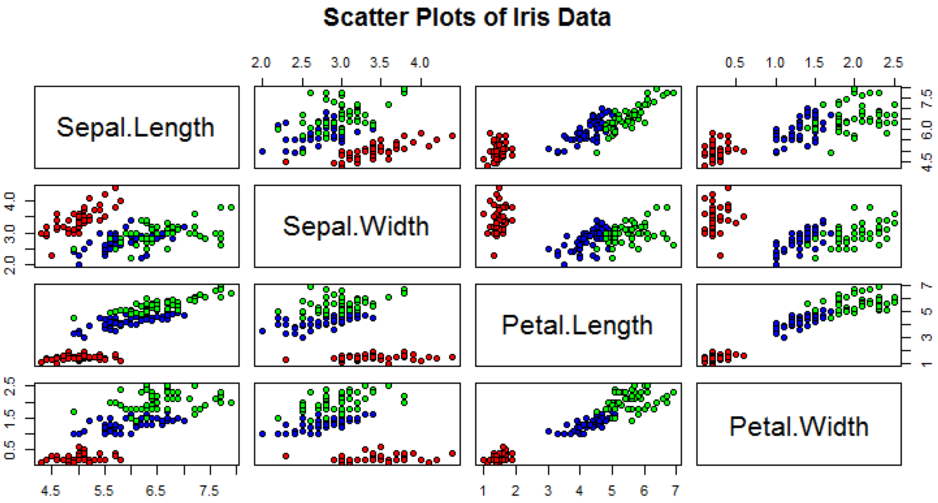


Figure 35: Scatter Plots of the Iris Data Set

3.3 Introduction of Dimension Reduction

•**Limited Data Visualization Methods on High Dimensional Data.** Basically there are few plot methods besides high dimensional box plot and matrix scatter plot in high dimensional data visualization. These 2 simple techniques cannot meet the huge and well rounded demand of high dimensional data visualization.

•**Curse of Dimensionality[3].** Its main idea is that the data points rapidly becomes sparse when the dimension of data set increases. It is hard to organize, storage and analysis high dimensional sparse data, also high dimensional data is lack of efficiency, since in many circumstance we are only interested in key features. This theme is widely accepted by current acdemia, and provides foundation for dimension reduction.

When the dimension of data is greater than 1000 or 10000, it is very hard, if not impossible, to present patterns within data points and variables by data visualization methods we introduced before. Intuitively we are interested in extracting important features from high dimensional data: reducing the dimension of data as much as possible, while keeping the useful information as much as possible. In following content we focus on exploring dimension reduction techniques: theories, effects, limitations and assessments.

3.4 Principle Components Analysis (PCA)

Principal Components Analysis (PCA)[9] computes the most meaningful axes to re-express a noisy, garbled data set. We hope the new axes filter out the noise and reveal hidden dynamics. As a widely used unsupervised dimension reduction technique , PCA is based on describing as much of the variance as possible. We reconstruct the data axes, which are orthogonal and are the linear combination of the original coordinates. PCA is different from the factor analysis method in that it does not require the analyst to subjectively determine factors or groups.

3.4.1 The Math

• **Eigenvalue Decomposition**. Assume A is a real square matrix whose eigenvalues are λ_1, $...\lambda_n$. Then we have:

$$det(A) = \prod_{i=1}^{n} \lambda_i \tag{1}$$

$$trace(A) = \sum_{i=1}^{n} \lambda_i \tag{2}$$

A square matrix A is diagonalizable if it is similar to a diagonal matrix, i.e., there exist an invertible matrix P and a diagonal matrix Λ such that

$$A = P\Lambda P^{-1} \tag{3}$$

Equation(1) and (2) imply a $n{\times}n$ matrix is diagonalizable if and only if it has n linearly independent eigenvectors, and also imply that $Ap_i = \lambda_i p_i$ for $1{\leq}i{\leq}n$, where p_i are the columns of P. This shows that the λ_i are the eigenvalues of A and p_i are the associated eigenvectors. For symmetric matrix A, there exists an orthogonal matrix Q and a diagonal matrix λ such that $A = Q\lambda Q^T$. Eigenvalue decomposition is also called the spectral decomposition.

• **Singular Value Decomposition (SVD).** Singular Value Decomposition (SVD) is defined for all matrices (rectangular or square) $A_{n\times d}$, unlike spectral decomposition only can be applied on square matrices. Suppose $A \in \mathbb{R}^{n\times d}$ and $C = A^T A$, which is a $d \times d$ matrix, the full singular value decomposition of matrix A is :

$$A_{n\times d} = U_{n\times n}\Sigma_{n\times d}V^T_{d\times d} \tag{4}$$

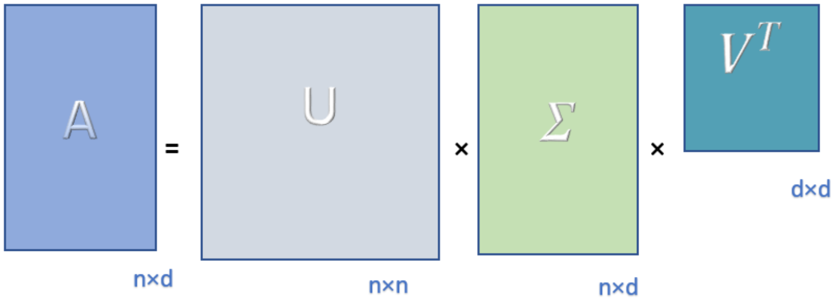


Figure 36: SVD. V, the right singular vectors of A, is the eigenvectors of $A^T A$; U, the left singular vectors of A, is the eigenvectors of AA^T; ${\sigma_i}^2$ $(1 \leq i \leq d)$ are the eigenvalues of $A^T A$.

We may use first k singular values to approximately explain matrix A. It is called Economic SVD.

$$A_{n\times d} \approx U_{n\times k}\Sigma_{k\times k}V^T_{k\times d} \qquad (k \ll d) \tag{5}$$

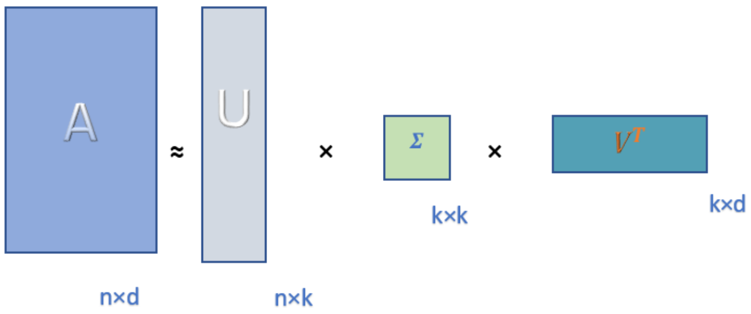


Figure 37: Economic SVD. The needed storage space is far less than that of Figure 36.

• **Principle Components Analysis (PCA).** PCA is an orthogonal linear transformation that transforms the data to a new coordinate system such that chosen coordinates (principal components) explained greatest variance by decreasing order. Generally if we centered data before doing dimension reduction, the means of all features are zero.

$$\tilde{A} = A - \bar{A} \tag{6}$$

The co-variance matrix C of matrix $\tilde{A}$ turns to:

$$C_{d\times d} = \frac{1}{n}\tilde{A}^T_{d\times n}\tilde{A}_{n\times d} \tag{7}$$

We are seeking the directions which maximize the variances. Based on equation (3), the eigenvectors V_i of matrix C corresponding to its eigenvalues λ_i in descending order are the directions on which the variances are greatest in descending order. Hence V is an orthonormal basis for restructured data and λ_i are the variances on the new basis.

$$C_{d\times d}V_{d\times n} = \Lambda_{d\times d}V_{d\times n} \tag{8}$$

$$C_{d\times d}V_{d\times k} \approx \Lambda_{d\times d}V_{d\times k} \qquad k \ll d \tag{9}$$

$A_{n\times d}V_{d\times k}$ is the reduced K dimensional coordinates of A.

3.4.2 The Algorithm

Algorithm 1 PCA

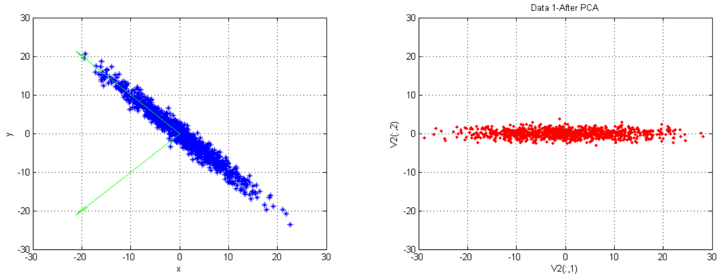
Input. Data set $A_{n \times d}$. n is the number of observations, d is the number of feature. K is the number of dimension to reduce.
Step 1. Organize data set x as an $n \times d$ matrix.
Step 2. Subtract off the mean for each feature, get $\tilde{A}$.
Step 3. Calculate the covariance matrix C, which is a d by d matrix.

$$C = \frac{1}{n} \tilde{A}^T \tilde{A} \tag{1}$$

Step 4. Calculate the eigenvalues Λ and eigenvectors V of matrix C. Sort the eigenvalues by descending order:$\lambda_1, \lambda_2, \ldots \lambda_n$,$\lambda_1 \geq \lambda_2 \ldots \geq \lambda_d$, and arrange the corresponding eigenvectors by column in the same order.
Step 5. Taking first K eigenvectors : K columns of matrix V.
Output. AV is the reduced K dimension coordinates.

3.4.3 Examples

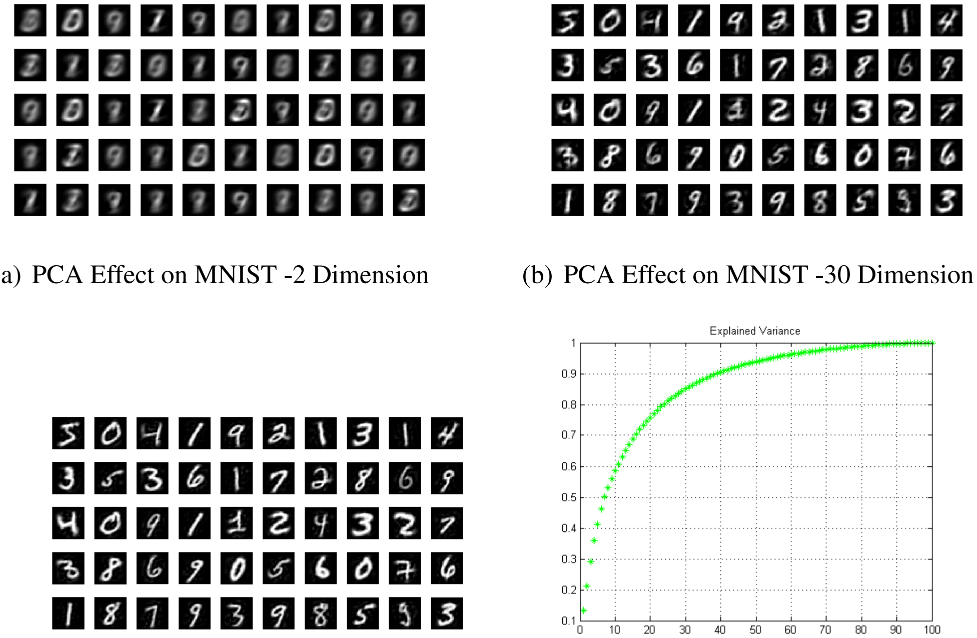
• 2D Gaussian Toy Data Example



(a) 2D Gaussian Data Before PCA (b) 1D Gaussian Data After PCA

Figure 38: PCA on 2D Gaussian Data: Under new basis, data makes $45°$ rotation, and has most variance in X axis direction.

• MNIST Handwritten Digit Example
Data Set Description. MNIST is a data set which has 60000 training samples and 10000 testing samples, with each sample a 28 × 28 pixel image . (Source: http://yann.lecun.com/exdb/mnist/)



(a) PCA Effect on MNIST -2 Dimension

(b) PCA Effect on MNIST -30 Dimension

(c) PCA Effect on MNIST -64 Dimension

(d) Explained Variance

Figure 39: PCA on MNIST

When dimension is 30, figure 37(b) shows pretty clear digits while saved more than 95% of storage space. Figure 37(d) shows the variance percentage PCA preserved when different K is chosen. When $K\geq 40$, PCA keeps more than 90% of variances.

3.5 Multidimensional Scaling (MDS)

MDS is a dimension reduction technique which attempts to preserve the pairwise distances in high dimensional space, and reconstruct this distance relationships in lower dimensional Euclidean space.

3.5.1 The Math

•**Problem**. For data set $\{x_1, x_2, ...x_n\}^T \in \mathbb{R}^d$, whose distances between x_i and x_j are $\ell_{i,j}$, $1 \leq i, j \leq n$. The distance matrix $D_{n\times n}$ of $X_{n\times d}$ is:

$$\|x_i - x_j\| = \ell_{i,j}, 1 \leqslant i, j \leqslant n \tag{10}$$

$$\begin{bmatrix} \ell_{1,1} & \ell_{1,2} & \cdots & \ell_{1,n} \\ \ell_{2,1} & \ell_{2,2} & \cdots & \ell_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n,1} & \ell_{n,2} & \cdots & \ell_{n,n} \end{bmatrix}$$

When we treat known distances as Euclidean distances, MDS tries to reconstruct data set $\{y_1, y_2, ...y_n\}^T \in \mathbb{R}^k (k \leq d)$, which meet

$$\|y_i - y_j\| \approx \ell_{i,j}, 1 \leqslant i, j \leqslant n \tag{11}$$

Since the solutions are not unique, we add a constraint $\sum y_i = 0$ and $\sum y_i = 0$.
•**Proof**. Squaring equation(10), we may expand them to get

$$\ell_{i,j^2} = \| x_i \|^2 + \| x_j \|^2 - 2\langle x_i, x_j \rangle$$

Summing over i an j separately to get

$$\sum_i \ell_{i,j^2} = \sum_i \| x_i \|^2 + n\| x_j \|^2$$
$$\sum_j \ell_{i,j^2} = \sum_j \| x_j \|^2 + n\| x_i \|^2$$

Denoting by

$$\ell_{.j}^2 = \frac{1}{n}\sum_i \ell_{ij}^2, \quad \ell_{i.}^2 = \frac{1}{n}\sum_j \ell_{ij}^2, \quad \ell_{..}^2 = \frac{1}{n^2}\sum_i\sum_j \ell_{ij}^2$$

Sum over i, j separately:

$$n^2\ell_{..}^2 = n\sum_i \| x_i \|^2 + n\sum_j \| x_j \|^2 = 2n\sum \| x_t \|^2$$

$$\frac{2}{n}\sum_t \| x_t \|^2 = \ell_{..}^2 \tag{12}$$

$$\| x_i \|^2 = \ell_{i.}^2 - \frac{1}{2}\ell_{..}^2 \tag{13}$$

$$\| x_j \|^2 = \ell_{.j}^2 - \frac{1}{2}\ell_{..}^2 \tag{14}$$

From equation(12),(13),(14)

$$\langle x_i, x_j \rangle = \frac{1}{2}\left(\ell_{i.}^2 + \ell_{.j}^2 - \ell_{..}^2 - \ell_{ij}^2\right) \qquad \forall i, j \tag{15}$$

Let L be a matrix whose entries are:

$$L_{i,j} = \frac{1}{2}\left(\ell_{i.}^2 + \ell_{.j}^2 - \ell_{..}^2 - \ell_{ij}^2\right) \tag{16}$$

From equations(15) and (16), $XX^T = L$ holds. The matrix L is symmetric with all row and column sums equal to zero, so we can find the eigenvalues and eigenvectors of L. $L = U\Lambda U^T$ is the spectral decomposition. Assuming L is positive definite, an exact solution of the above equation is

$$X = U_{n\times d}\Lambda^{\frac{1}{2}} = \left(\sqrt{\lambda_1}u_1..., \sqrt{\lambda_d}u_d\right)$$

And the reduced k dimensional coordinates of X is:

$$Y = U_{n\times k}\Lambda^{\frac{1}{2}} = \left(\sqrt{\lambda_1}u_1..., \sqrt{\lambda_k}u_k\right), k < d$$

3.5.2 The Algorithm

Algorithm 2 MDS

```
Input. Distance Matrix $D_{n\times n}$. $n$ is the number of observations. $K$ is the number of dimension to reduce.
Step 1. Center matrix $D$, making $\sum d_i = 0, (1 \leq i \leq n)$. get $\tilde{D}$.
Step 2. For every $i, j$, $(1 \leq i, j \leq n)$, calculate $\ell^2_{i.}, \ell^2_{.j}, \ell^2_{..}$.
Step 3. Construct $L$ symmetric matrix. $L_{i,j} = \frac{1}{2}\left(\ell^2_{i.} + \ell^2_{.j} - \ell^2_{..} - \ell^2_{ij}\right)$.
Step 4. Calculate the eigenvalue $\Lambda$ and eigenvector $U$ of $L$. Sort the eigenvalues by descending order:$\lambda_1, \lambda_2, \ldots \lambda_n$,$\lambda_1 \geq \lambda_2 \ldots \geq \lambda_n$, and take the corresponding $K$ eigenvectors in the same order.
Output. $Y = U_{n\times k}\Lambda^{\frac{1}{2}} = \left(\sqrt{\lambda_1}u_1..., \sqrt{\lambda_k}u_k\right), k < d$
```

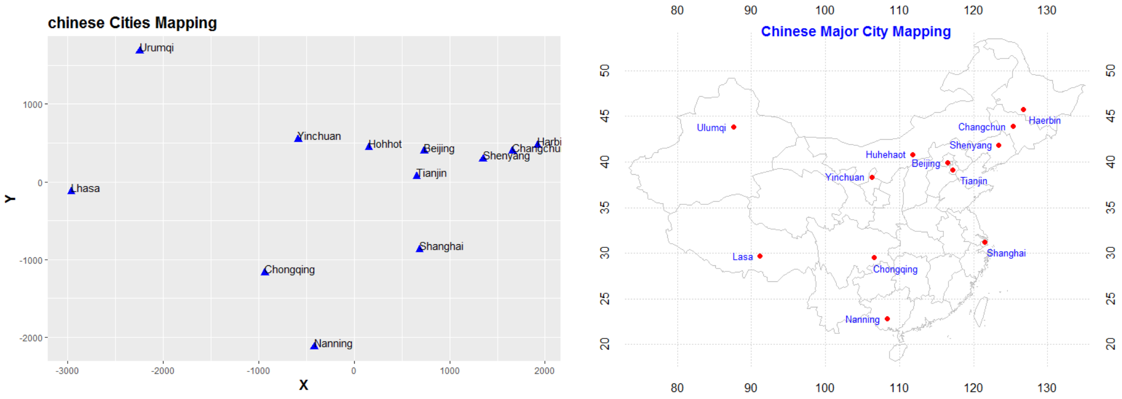
3.5.3 Chinese Cities Example

The table 1 is a Chinese major cities distances matrix.

City	Beijing	Tianjin	Shanghai	Chongqing	Hohhot	Urumqi	Lhasa	Yinchuan	Nanning	Harbin	Changchun	Shenyang
Beijing	0	125	1239	3026	480	3300	3736	1192	2373	1230	979	684
Tianjin	125	0	1150	1954	604	3330	3740	1316	2389	1207	955	661
Shanghai	1239	1150	0	1945	1717	3929	4157	2092	1892	2342	2090	1796
Chongqing	3026	1954	1945	0	1847	3202	2457	1570	993	3156	2905	2610
Hohhot	480	604	1717	1847	0	2825	3260	716	2657	1710	1458	1164
Urumqi	3300	3330	3929	3202	2825	0	2668	2111	4279	4531	4279	3985
Lhasa	3736	3740	4157	2457	3260	2668	0	2547	3431	4967	4715	4421
Yinchuan	1192	1316	2092	1570	716	2111	2547	0	2673	2422	2170	1876
Nanning	2373	2389	1892	993	2657	4279	3431	2673	0	3592	3340	3046
Harbin	1230	1207	2342	3156	1710	4531	4967	2422	3592	0	256	546
Changchun	979	955	2090	2905	1458	4279	4715	2170	3340	256	0	294
Shenyang	684	661	1796	2610	1164	3985	4421	1876	3046	546	294	0

Table 1: Chinese Major Cities Distances

After the MDS, the reconstructed maps for 12 cities is as Figure 40(a).



(a) Chinese Cities Mapping via MDS (b) Original Chinese Cities Map

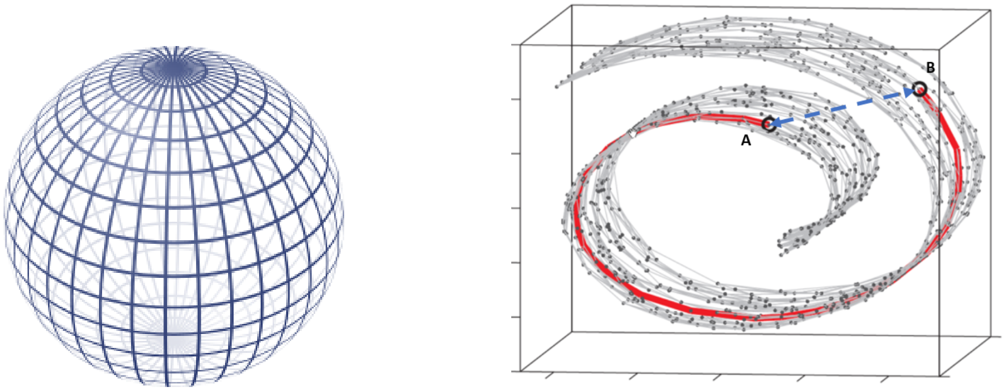
Figure 40: MDS Mapping for Chinese cities

Figures $40(a)$ and $40(b)$ show quite similar relative positions of major cities, while $40(a)$ still has a little bit of distortion. There may be 3 reasons leading to the distortion.

• The original map Figure 40(b) is not very accurate.
• The MDS only take first 2 dimensions for the position matrix, which means the result is just a approximate distance. Stress score tests how close the approximation is as for the distance rebuilding. Here the stress score is 0.0805, a very good mapping but still 8% distance unexplained.
• The surface of earth is manifold and the distances matrix D is not Euclidean but geodesic. In MDS we approximately treat the distance as Euclidean distance, as we treat 3-D distance as 2-D distance. This is the main reason of mapping distortion.

3.6 ISOmap

• **The Limitation of MDS**. MDS reduces dimensions while keep the dissimilarity matrix unchanged, if and only if we define the distance as Euclidean distance. What if the distances are not Euclidean?



(a) Will MDS work in sphere?

(b) Will MDS work in Swiss Roll?

Source of Figure 41(a): https://en.wikipedia.org/wiki/Sphere
Source of Figure 41(b): https://web.mit.edu/cocosci/Papers/sci_reprint.pdf

Figure 41: MDS doesn't work on sphere or manifold space.

If the distance between a given observation and its neighbor is not far, we can approximately treat the geodesic distance as Euclidean distance, but we can not treat the distance from North pole to South pole as Euclidean, also we can not treat the distance between point A and B in Swiss Roll as Euclidean.

• **ISOmap, inspired by MDS, extends MDS to preserving geodesic distance**, and maps high dimensional manifold data into low dimensional space.

3.6.1 The Math

The difference between MDS and ISOmap is the distances matrix construction. ISOmap distances matrix represent geodesic distances. The rules include:
• For neighboring points, Euclidean distance is a good approximation to the geodesic distance.
• For faraway points, estimate the distance by a series of short hops between neighboring points.
• Find shortest paths in a graph with edges connecting neighboring data points.
Once geodesic distances matrix D constructed, use classical metric MDS.

3.6.2 The Algorithm

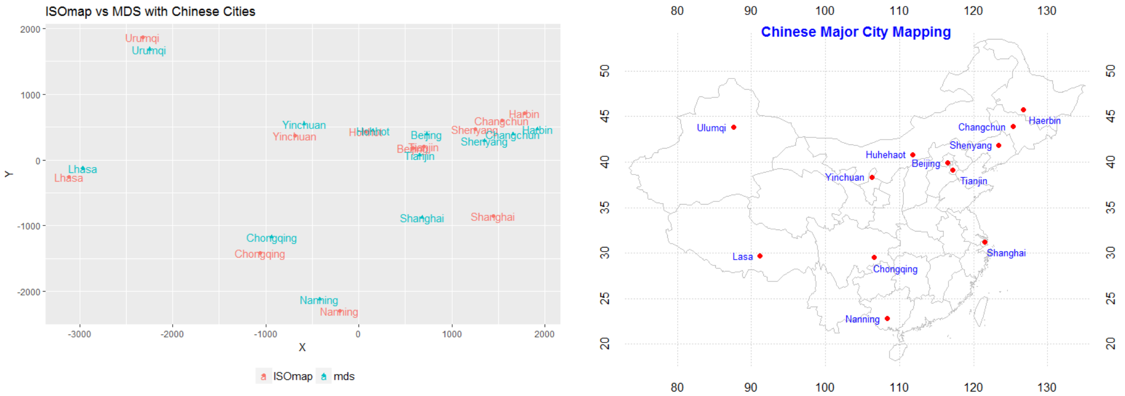
Widely used algorithms for constructing a geodesic distances matrix include Dijkstra's algorithm and FloydWarshall algorithm.

Algorithm 3 ISOmap

```
Input. Data set $X_{n\times m}$. $n$ is the number of observations, $m$ is the number of features. $k$ is the number of neighbors. $d$ is the reduced dimensions.
Step 1. Determine the neighbors of each point (In some fixed radius or KNN).
Step 2. Construct a neighborhood graph for each point. The weights of edges are the Euclidean distance from this point to neighboring points.
Step 3. Use Dijkstra's algorithm or Floyd−Warshall algorithm to calculate the shortest path for any 2 nodes. The geodesic distance matrix D constructed.
Step 4. Set the Geodesic distance matrix $D$ as the input of MDS algorithm.
Step 5. Apply MDS algorithm to do eigenvalue decomposition and get the new coordinates of nodes.
Step 6. Take $d$ columns of new coordinates as the dimension reduced coordinates.
Output. As MDS algorithm, $Y = U_{n\times d}\Lambda^{\frac{1}{2}} = \left(\sqrt{\lambda_1}u_1..., \sqrt{\lambda_d}u_d\right), d < m.$
```

3.6.3 Example: Chinese Cities ISOmap vs MDS

Based on the original Euclidean distance matrix (table 1), taking K=3 as the number of neighbors, using Floyd−Warshall algorithm, we rebuilt the geodesic distance matrix D and set D as the input of MDS algorithm.



(a) ISOmap on Chinese Cities , K=3, $d = 2$

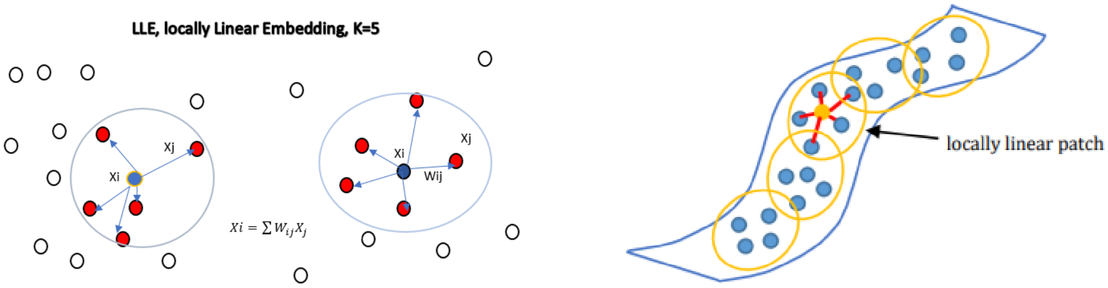
(b) Chinese Cities Map

Figure 42: For the Chinese Cities data set, the ISOmap method performed better than the MDS method.

Shanghai, Changchun, Shenyang, Harbin all have more accurate mapping. ISOmap has better performance on Chinese cities by better preserved geodesic distance matrix.

3.7 Locally Linear Embedding (LLE)

Unlike PCA trying to reconstruct data by linearly rearranging the space basis to get maximum variance globally, LLE is a dimension reduction method that keeps the local neighboring linear relationships when mapping high dimensional data into low dimensional space.



(a) Idea of LLE−1

(b) Idea of LLE−2

Source of Figure 43(b):
https://www.math.sjsu.edu/~gchen/Math285F15/285%20Final%20Project%20-%20LLE.pdf

Figure 43: LLE builds and keeps local linear relationships on manifold space.

3.7.1 The Math

Step 1: Construct weight matrix W . When the distance between two nodes is close enough, it is reasonable to assume the local distance is linear. Any node X_i can be denoted by:

$$X_i = \sum_{j=1}^{k} W_{ij} x_{ij} (1 \leq j \leq k)$$

Where x_j is k nearest neighbors of x_i. The weight W choose to minimize reconstruction error:

$$Min\ \varepsilon(W) = \sum_{i=1}^{n} \left(X_i - \sum_{j=1}^{k} W_{ij} x_{ij} \right)^2 \tag{17}$$

With three constrains:
- $W_{ij} = 0$ if point j is not the neighbor of point i.
- $\sum W_{ij} = 1$ for point i $(1 \leq i \leq n, and \quad 1 \leq j \leq k)$.
- $\sum x_i = 0$ for point i $(1 \leq i \leq n)$.

By derivation on Equation weight matrix W is found.
Step 2: Construct low dimensional projecting Y matrix.
$Y_{n \times d}$ matrix meet:

$$Min\ \phi(Y) = \sum_{i=1}^{n} \left(Y_i - \sum_{j=1}^{k} W_{ij} Y_{ij} \right)^2 \tag{18}$$

Find $d + 1$ bottom eigenvectors of B=$(I - W)^t (I - W)$ in corresponding eigenvalues ascending order. The second to the $(d + 1)$th eigenvetors of B form the low dimensional (d-dimension) reconstruction Y.

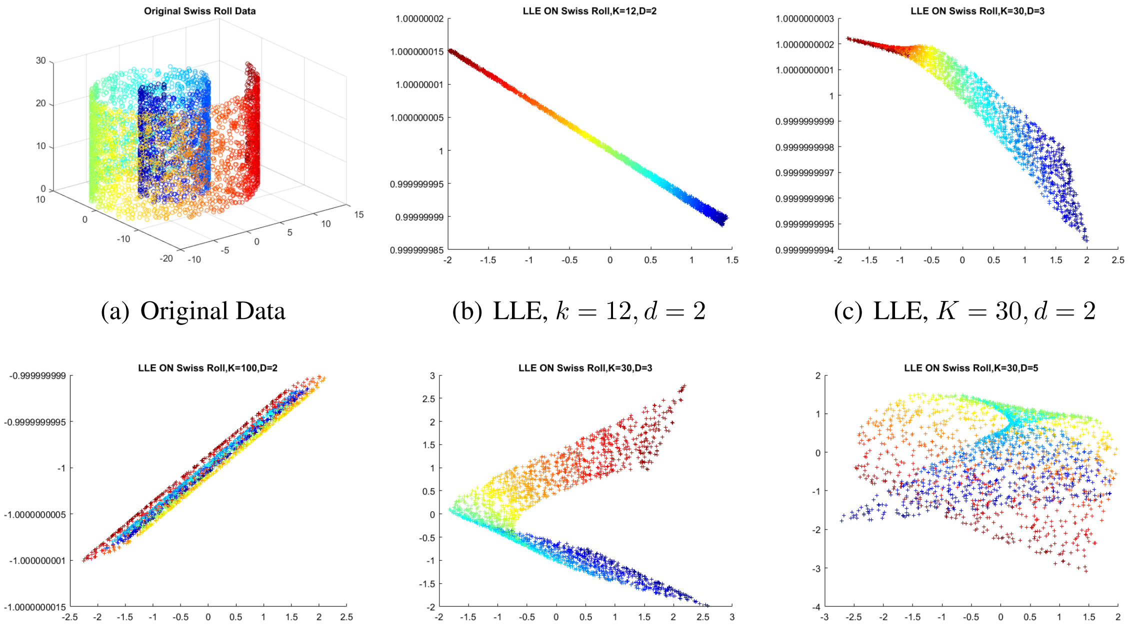
3.7.2 The Algorithm

Algorithm 4 Locally Linear Embedding(LLE)

```
Input . Data set $X_{n\times m}$. $n$ is the number of observations, $m$ is the number of features. $k$ is the number of neighbors. $d$ is the reduced dimensions.
Step 1. For $i = 1$ to $n$, determine the $K$ neighbors of each point .
Step 2. For $i = 1$ to $n$, calculate the weight matrix $W_i$.
Step 3. Calculate $B = (I - W)^T (I - W)$.
Step 4. Do spectral decomposition of $B$. Take $d$ eigenvectors $V$ by corresponding eigenvalues ascending order $v_2, v_3, \cdots, v_{d+1}$.
Output. $Y$= $(v_2, v_3, \cdots, v_{d+1})$
```

3.7.3 The Swiss Roll Example

• **Swiss Roll Data Set** was to create several points in 2D, and then map them to 3D with some smooth function, and then to see what the algorithm would find when it mapped the points back to 2D (Source: http://people.cs.uchicago.edu/~dinoj/manifold/swissroll.html). The Swiss Roll data set we used has 2048 observations.



(a) Original Data

(b) LLE, $k = 12, d = 2$

(c) LLE, $K = 30, d = 2$

(d) LLE, $K = 100, d = 2$

(e) LLE, $K = 30, d = 3$

(f) LLE, $K = 30, d = 5$

Figure 44: LLE Dimension Reduction with Different K and d

Figure 44 shows factor K and factor d have significant influences on the result of low dimensional projection.

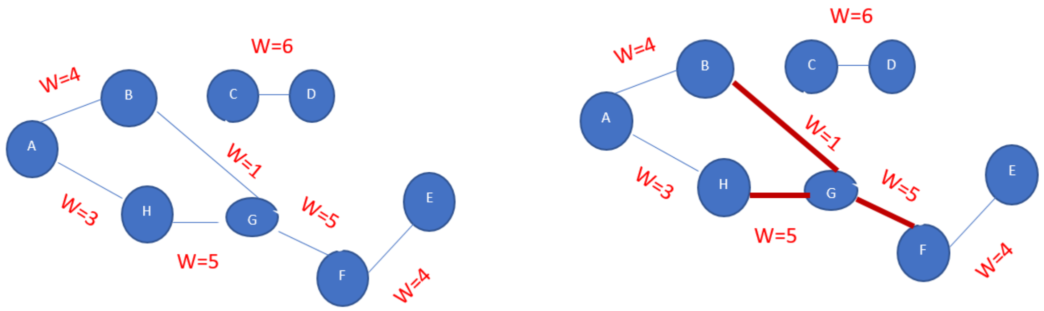
•Since data visualization is in 2D or 3D space, $d = 2$ or $d = 3$ is recommended. When $d > 3$, part of reconstruction information cannot be displayed, which may lead projection result not ideal.
•The choose of K is related to the density of data. In Swiss Roll data set, when $k > 30$, the weight matrix doesn't reflect locally linear relationship well, so $K > 30$ is not recommended.
•In Figure 44, best result is at $k = 12$, $d = 2$.

3.8 Laplacian Eigenmaps

To some extent, Laplacian Eigenmaps is like LLE, preserving local relationships while projecting high dimensional data into low dimensional space. The difference between them is the the definition of relationships between data points. Laplacian Eigenmaps constructs relationships by graph, which is denoted by G (V, E, W). V stands for vertices (points), E stands for edges connecting neighbor points, and W stands for weights to measure the distance (or dissimilarity) of two neighbor nodes.The closer the two nodes are, the higher the weight is. Weight between x_i and x_j can be set as 0 or 1; or as the Gaussian Kernel Function

$$W_{ij} = e^{-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}}$$

Laplacian Eigenmaps devotes to find embedding Y which minimize $\sum_{i=1}^{n} (y_i - y_j)^2 w_{ij}$.



(a) Graph Construction

(b) Weights of Neighbor Nodes of Node G

Figure 45: Graphs Construction In Laplacian Eigenmaps

3.8.1 The Math

• **Matrix Construction**. W matrix, D matrix and L matrix.

$$W_{ij} = \begin{cases} e^{-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}} & \text{if } x_i \text{ and } x_j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

$$D_{ii} = diag(d_1, d_2, ...d_n) \qquad d_i = \sum_j W_{ij} \, (1 \leq i \leq n) \tag{20}$$

$$L = D - W \tag{21}$$

- **Justification**. For any y $\in \mathbb{R}^n$, from equation(19), (20), (21):

$$y^T Ly = y^T Dy - y^T Wy$$

$$y^T Ly = \sum_{i=1}^{n} d_i {y_i}^2 - \sum_{i,j=1}^{n} w_{ij} y_i y_j$$

$$y^T Ly = \frac{1}{2}\left(\sum_{i=1}^{n} d_i {y_i}^2 - 2\sum_{i,j=1}^{n} w_{ij} y_i y_j + \sum_{i=1}^{n} d_i {y_i}^2\right)$$

$$y^T Ly = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(y_i - y_j)^2 \tag{22}$$

The low dimensional Laplacian Eigenmaps Embedding Y meets:

$$\min_{Y^T DY=1} \sum_{i=1}^{n} (y_i - y_j)^2 w_{ij} = Y^T LY \tag{23}$$

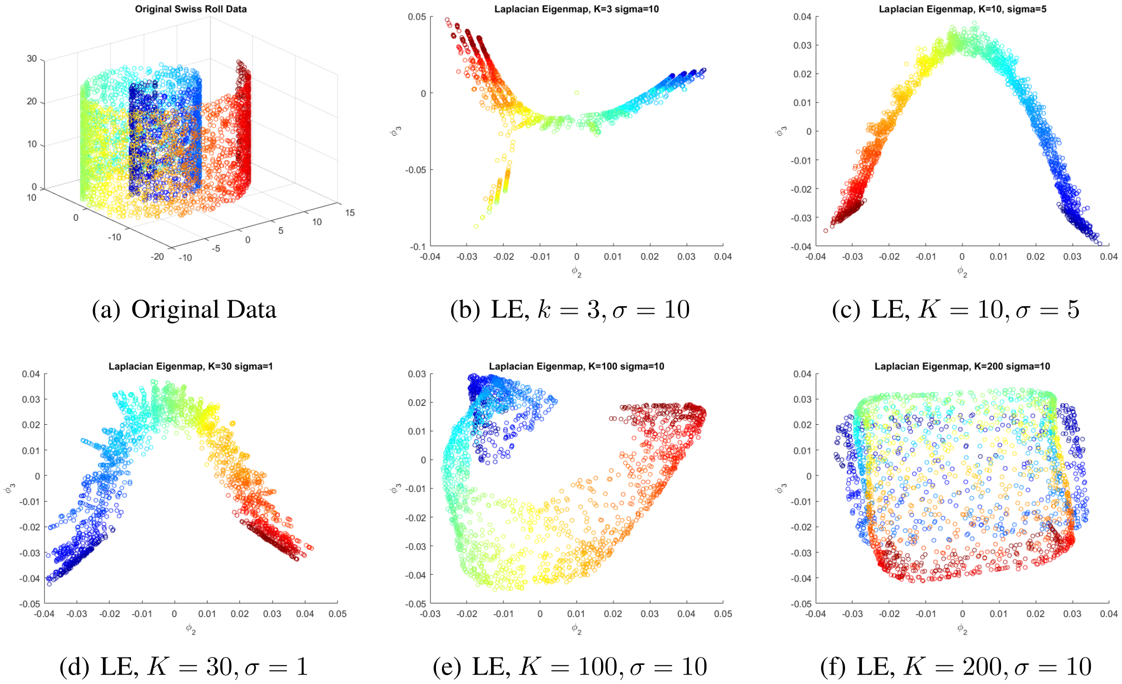
L is a semi-definite matrix and has n non-negative eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \cdots \lambda_n$. Based on equation (22) and (23), we do spectral decomposition of L, the d eigenvetors corresponding to $\lambda_2 \leq \lambda_3 \cdots \lambda_{d+1}$ is the low dimensional embedding Y.

3.8.2 The Algorithm

Algorithm 5 Laplacian Eigenmaps

```
Input. Data set $X_{n\times m}$. $n$ is the number of observations; $m$ is the number of features; $k$ is the number of neighbors (KNN); $d$ is the reduced dimensions; $\sigma$ is a weight matrix factor.
Step 1. For $i = 1$ to $n$, construct $K$ neighbors for each $x_i$.
Step 2. Calculate Weight Matrix. ( $\sigma$ is a input factor).$W_{ij} = e^{-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}}$.
Step 3. Compute $D$ matrix.$D_{ii} = diag(d_1, d_2, ...d_n) \quad and \quad d_i = \sum_j W_{ij}\,(1 \leq i \leq n)$.
Step 4. $L = D - W$.
Step 5. Compute eigenvalues of $L$. Sort them in ascending order, take the 2nd to $d+1$nd corresponding eigenvectors $(v_2, v_3, \cdots, v_{d+1})$.
Output. $Y$= $(v_2, v_3, \cdots, v_{d+1})$.
```

3.8.3 Swiss Roll Example



(a) Original Data (b) LE, $k = 3, \sigma = 10$ (c) LE, $K = 10, \sigma = 5$

(d) LE, $K = 30, \sigma = 1$ (e) LE, $K = 100, \sigma = 10$ (f) LE, $K = 200, \sigma = 10$

Figure 46: Laplacian Eigenmaps on Different K and σ

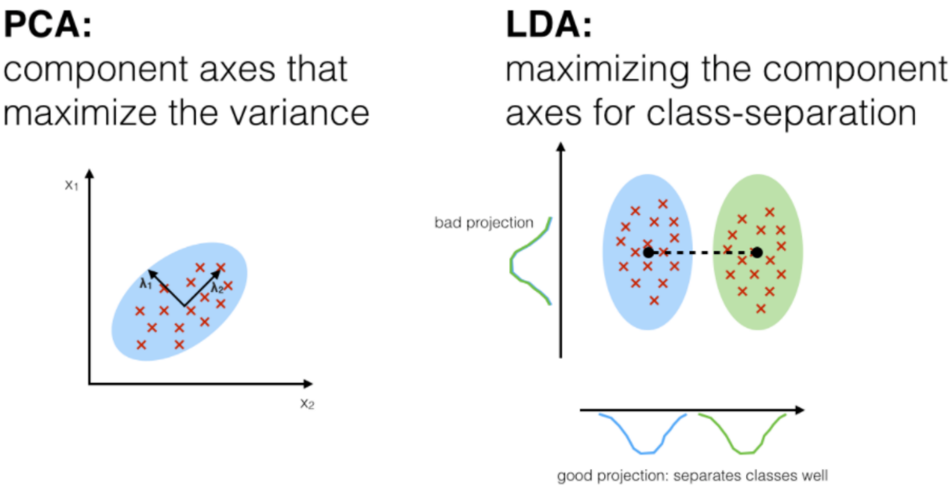
Figure 46 shows factor K has more influences on the result of low dimensional projection than factor σ does.
• Laplacian Eigenmaps projection is more non-linear than LLE, and more robust than LLE. • When $k > 100$, the weight matrix doesn't reflect dissimilarity well, so $K > 100$ is not recommended.

3.9 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a supervised linear dimension reduction technique. Unlike unsupervised technique PCA, which projects data to get variance as much as possible, LDA projects data to labeling data. LDA doesn't care much about variance. It focus on maximize distances between classes, while minimize variance within the same class.

3.9.1 LDA vs PCA

Both LDA and PCA are similar since both of them are seeking a new basis which is the linear transformation of original space; but they are different because PCA tries to find a direction on which all points spread most, while LDA wants to project points within the same class onto same line segment. See Figure 47 and Table 2.


PCA:
component axes that maximize the variance
x1
x2
λ1
λ2
LDA:
maximizing the component axes for class-separation
bad projection
good projection: separates classes well

(a) LDA V.S. PCA

Source of Figure 47(a): https://sebastianraschka.com/Articles/2014_python_lda.html

Figure 47: LDA V.S. PCA

Dimension Reduction Methods	PCA	LLA
Variance	Maximize variance	Minimize variance within class
Classes	Doesn't care classes	Maximize the distances between classes
Linear or not	Linear	Linear
Style	Unsupervised	Supervised
Label	Without label	With labels
Dimensions	$\leq$ Original Dimensions	Number of classes -1
Basis	Orthogonal Basis	Orthogonal not required

Table 2: LDA V.S. PCA

3.9.2 The Math

We have data set $X_{d\times n}$, n is the number of observations and d is the number of features, and label vector $(D_1, D_2, \cdots, D_c)$. The center of each class is :

$$\mu_i = \frac{1}{n_i}\sum_{x\in D_i} x$$

Where n_i is the number of samples in class i. We are trying to project data set in $k-$dimensional space $W = [w_1, w_2, \cdots, w_k]$, to make $y = W^T x$ is a k dimensional vector. We denote Within-class scatter matrix S_W and Between-class scatter matrix S_B.

• Justification

After projecting, the center of each class is:

$$\tilde{\mu_i} = W^T \mu_i \tag{24}$$

And the variance in each class is:

$$\tilde{s_i} = \sum_{x\in D_i} (W^T x - \tilde{\mu_i})^2 \tag{25}$$

From equation (24) and (25) we have:

$$\tilde{s_i} = \sum_{x\in D_i} \left(W^T\left(x-\mu_i\right)\right)^2 = \sum_{x\in D_i} W^T\left(x-\mu_i\right)\left(x-\mu_i\right)^T W$$

Denote $\sum_{x\in D_i}\left(x-\mu_i\right)\left(x-\mu_i\right)^T$ as Within-class scatter matrix S_{W_i} in class i, we have :

$$\tilde{S} = \sum_{\forall i} \tilde{s_i} = W^T\left(\sum_{\forall i} S_{W_i}\right)W$$

Now we are measuring the variance between classes. Denote the center of all classes as:

$$\tilde{\mu} = \frac{1}{C}\sum_i \tilde{\mu}_i$$

C is the number of classes, and we may define the distances between classes as:

$$\tilde{T} = \sum_i n_i\left(\tilde{\mu}_i - \tilde{\mu}\right)^2 \tag{26}$$

Where n_i is the number of observations in class i. Denote $\sum_i n_i\left(\tilde{\mu}_i-\tilde{\mu}\right)\left(\tilde{\mu}_i-\tilde{\mu}\right)$ as Between-class scatter matrix S_{W_i}, based on equation (26), we have:

$$\tilde{T} = \sum_i n_i\left(\tilde{\mu}_i - \tilde{\mu}\right)^2 = \sum_i n_i W^T\left(\tilde{\mu}_i - \tilde{\mu}\right)\left(\tilde{\mu}_i - \tilde{\mu}\right)^T W = W^T S_B W \tag{27}$$

The best projection W should maximize $\tilde{T}$ and minimize $\tilde{S}$, hence

$$\max J\left(W\right) = \frac{\tilde{T}}{\tilde{S}} = \frac{W^T S_B W}{W^T S_W W} \tag{28}$$

Equation(28) is known as generalized Reyleigh quotient[16], and $J\left(W\right)$ is maximized when W is the eigenvectors corresponding to the largest d eigenvalues of ${S_W}^{-1}S_B$, $1 \leq k \leq c-1$.

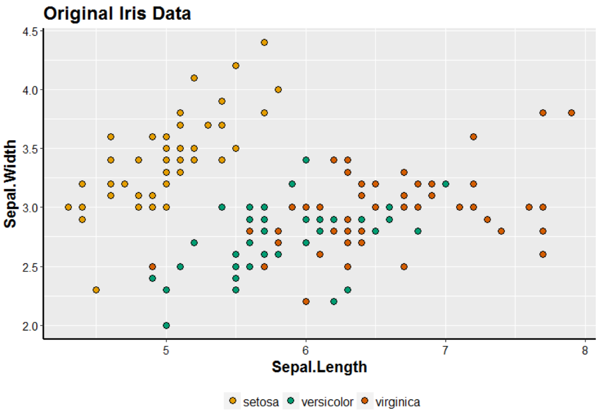
3.9.3 The Algorithm

Algorithm 6 LDA

```
Input. Data set $X$ with $n$ observations and $d$ dimensions. Label Vector $D$ with length $n$ marks the class of each point $x_i$. $1 \leq i \leq n$ . The number of classes is $C$.
Step 1. Calculate the center of each class j. $1 \leq j \leq C$.
Step 2. Compute the within scatter matrix $S_W$ and between scatter matrix $S_B$.
Step 3. Compute ${S_W}^{-1}S_B$.
Step 4. Do eigenvalue decomposition for ${S_W}^{-1}S_B$.
Step 5. Sort the eigenvalues by descending order. Choose k corresponding eigenvectors as column vectors to form matrix $W$. $1 \leq k \leq C-1$. $W$ is the reconstructed basis.
Output. $Y = W^T X$. $Y$ is the $k \times n$ dimension reduced coordinates of $X$.
```

3.9.4 Iris Example: LDA vs PCA

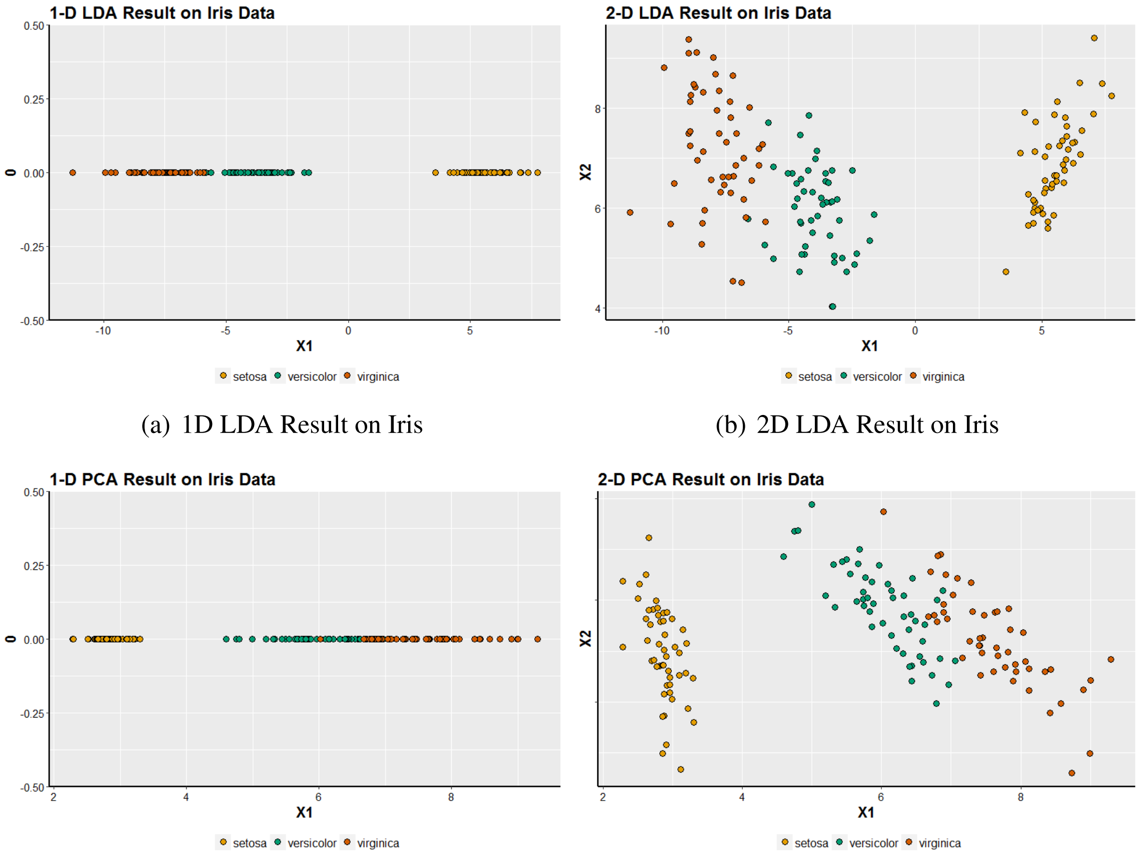
Data Set: Iris. Source: https://archive.ics.uci.edu/ml/datasets/iris



(a) Original Iris

Figure 48: Original Iris: 150 Points with 5 Dimensions Each

PCA and LDA results see Figure 49.



(a) 1D LDA Result on Iris

(b) 2D LDA Result on Iris

(c) 1D PCA Result on Iris

(d) 2D PCA Result on Iris

Figure 49: PCA vs LDA on Iris

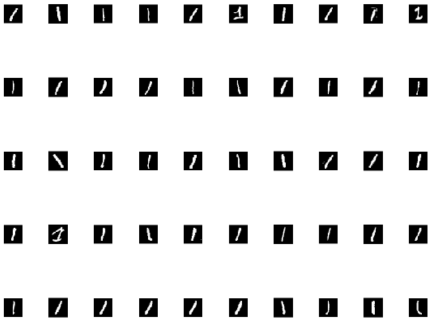
Comments based on Figure 49.

- LDA turns 4D Iris to 1D while group information keeps well.
- PCA spreads more widely than LDA since it keeps most variance.
- LDA does better grouping than PCA since it keeps points in different groups far away while points in same group as close as possible.

3.10 Experiments

3.10.1 Experiment 1. Dimension Reductions Comparison on MNIST Digit 1

- **Data Description**. Extract handwritten digit 1 from data set MNIST. (Source: http://yann.lecun.com/exdb/mnist/). Data size is 1000.



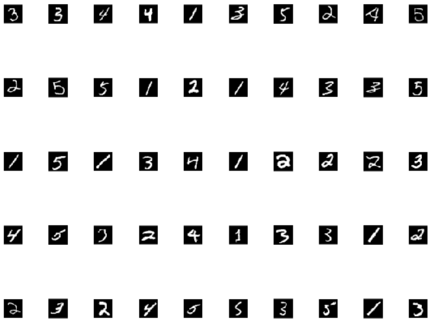
(a) Sample Data of Experiment 1

Figure 50: Sample Digit 1

- **Methodology**. PCA, MDS, ISOmap, LLE, Laplacian Eigenmaps were used to do dimension reduction on MNIST digit 1, then we visualize the original images in 2D reconstructed basis to observe and analyze the feature extraction effects of different dimension reduction techniques.
- **Exclusion**. LDA is excluded from this experiment since it need labeled(grouped) data while all digit 1 has same label "1", so LDA is not feasible for Digit 1 data set.

3.10.2 Experiment 2. Dimension Reductions Comparison on MNIST Digit 1 to 5

- **Data Description**. Extract handwritten digit 1, 2, 3, 4, 5 from data set MNIST. (Source: http://yann.lecun.com/exdb/mnist/). Data size is 3000.



(a) Sample Data of Experiment 2

Figure 51: Sample Digit 1 to 5

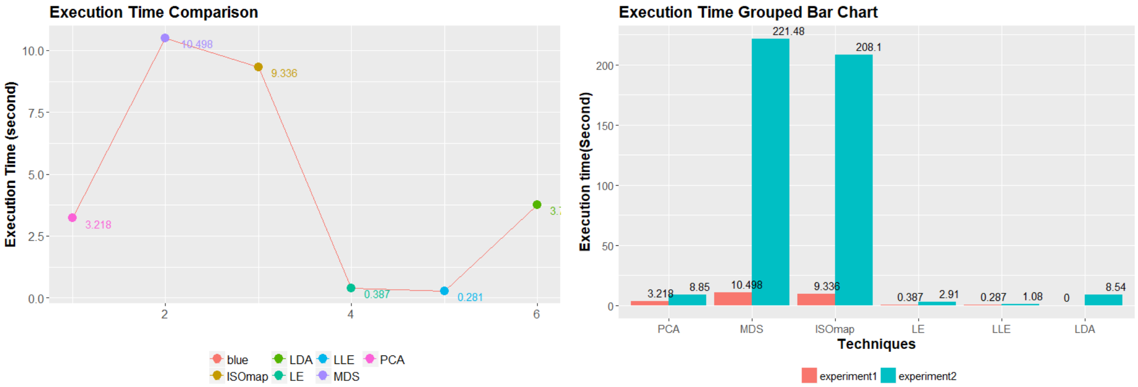
• **Methodology**. PCA, MDS, ISOmap, LLE, Laplacian Eigenmaps and LDA were used to do dimension reduction on MNIST digit 1 to 5, then we visualize the original digit images in 2D reconstructed basis. Since digit 1 to 5 is grouped naturally, we focus on observing the clustering effects of different dimension reduction techniques.

3.10.3 Experiment Results Analysis

• **Efficiency Comparison**
The theoretical time complexity of different dimension reduction techniques is as table 3.

Techniques	PCA	MDS	ISOmap	Laplacian Eigenmaps	LLE	LDA
Time Complexity	$O(nD) + O(D^3)$	$O(n^3)$	$O(Dnlog(n)) + O(n^3)$	$O(Dnlog(n)) + O(pn^2)$	$O(Dnlog(n)) + O(pn^2)$	$O(nD) + O(D^3)$
Experiment 1(Sec)	3.218	10.498	9.336	0.387	0.281	NA
Experiment 2(Sec)	8.85	221.48	208.10	2.91	1.08	8.54

Table 3: Time Complexity and Execution Time



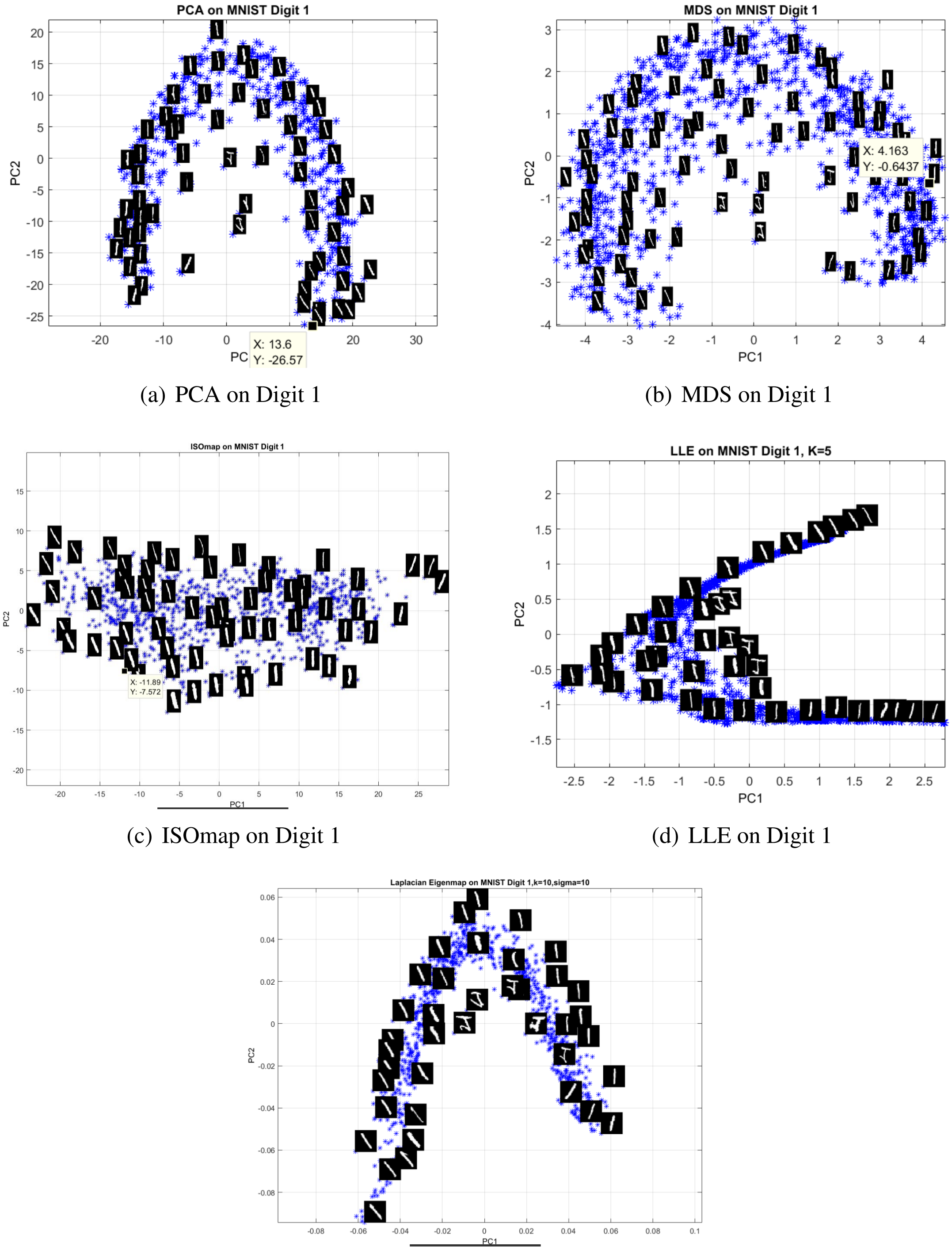
(a) Running Time Comparison-Digit 1

(b) Running Time Comparison

Figure 52: Efficiency Comparison

From Table 3 and Figure 52, based on the characteristics of data set, we found:
• LE and LLE are most efficient dimension reduction techniques among above 6 techniques.
• MDS and ISOmap are least efficient dimension reduction techniques since the construction of distance matrix is time consuming.
• MDS and ISOmap ‘s efficiency decrease extremely since the time complexity is related to $O(n^3)$.
• PCA, LE and LLE ‘s efficiency decrease steadily since the time complexity is related to $O(n^2)$ or $O(nD)$.

• **Effects Comparison**
The Digit 1 Visualizations on Reconstructed 2D Coordinate is as Figure 53.



(a) PCA on Digit 1

(b) MDS on Digit 1

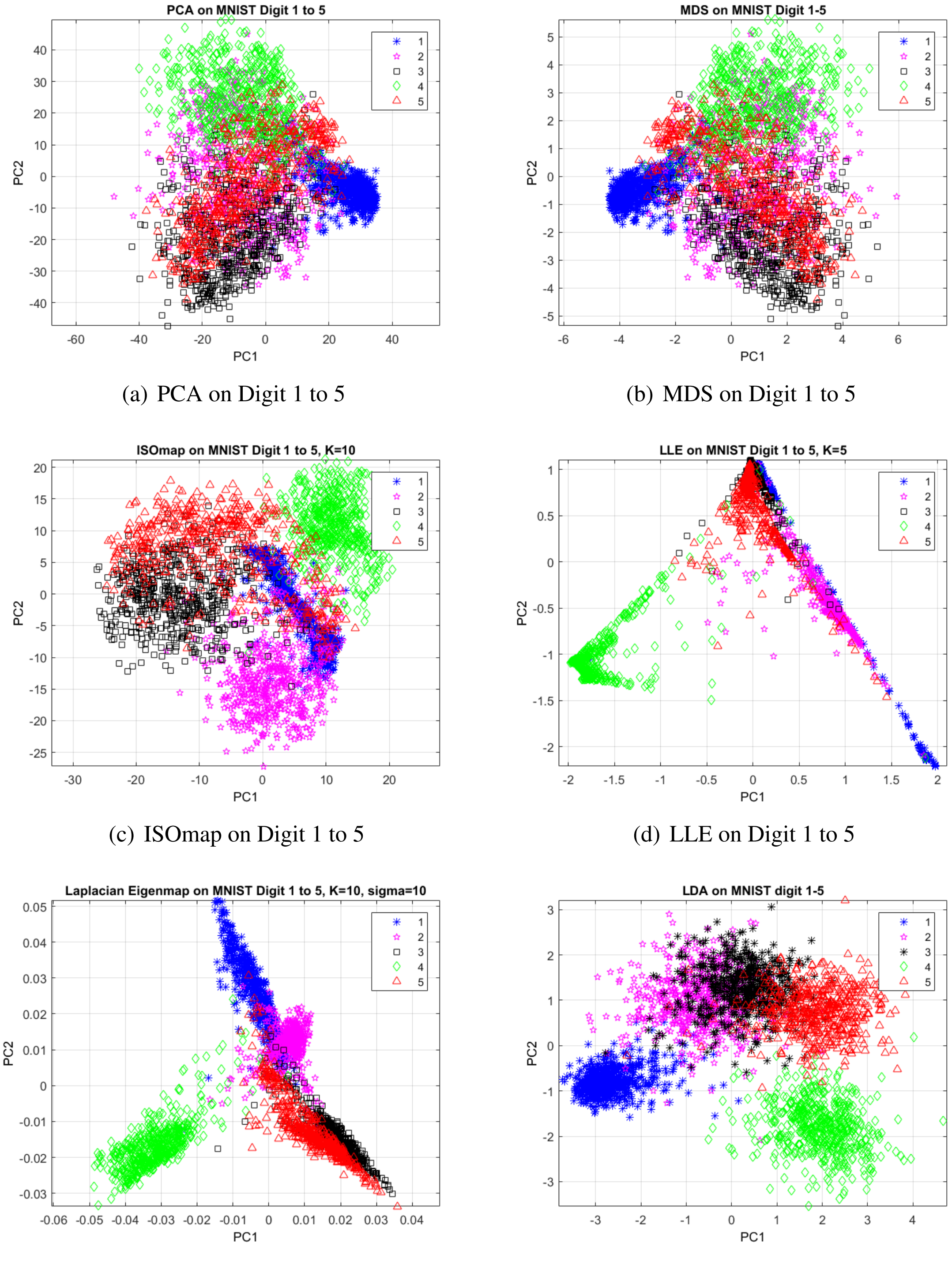
(c) ISOmap on Digit 1

(d) LLE on Digit 1

(e) Laplacian Eigenmaps on Digit 1

Figure 53: Feature Extraction Effects Comparison on MNIST Digit 1

The Digit 1-5 Visualizations on Reconstructed 2D Coordinate is as Figure 54.



(a) PCA on Digit 1 to 5

(b) MDS on Digit 1 to 5

(c) ISOmap on Digit 1 to 5

(d) LLE on Digit 1 to 5

(e) Laplacian Eigenmaps on Digit 1 to 5

(f) LDA on Digit 1 to 5

Figure 54: Dimension Reduction Effects on MNIST Digit 1 to 5

We may treat data set only include digit 1 as linear dominated and non-grouped data , and treat data set include digit 1 to 5 as nonlinear and grouped data.
From Figure 53, we noticed:

• PCA performs well in extracting key features in linear data. In Experiment 1, the direction which has most variance was the degree of tilt in handwritten digit 1.
• MDS performs well in extracting key features in linear data. It caught two features of digit 1, on the order of importance : the slope and the thickness.
• ISOmap worked best on digit 1 since it expressed both the slope and the thickness of digit 1 best. That is because it can extract both linear and nonlinear features.
• Overall nonlinear dimension reduction algorithm: ISOmap, LLE, Laplacian Eigenmaps worked better than linear algorithm : PCA and MDS. That is because they can catch the nonlinear features in digit 1, while PCA and MDA cannot.

From Figure 54, we realized:
• LDA performs best in digit 1-5 visualization. It made the same digit grouped together and different digits kept away.
• PCA and MDS performs poorly on nonlinear grouped data due to they cannot catch hidden nonlinear attributes.
• ISOmap performs OK but worse than Laplacian Eigenmaps and LLE do, showing Laplacian Eigenmaps and LLE can preserve the hidden nonlinear attribute better than ISOmap does.

3.10.4 Evaluations and Suggestions

• Dimension Reduction Techniques Evaluation

Techniques	Pros	Cons
PCA	Simple, convenient and stable. Extract linear attributes effectively	Performs poorly on manifold data.
MDS	Keeps the linear distance relationship well.	Time consuming especially when data size is large. Works poorly for nonlinear data
ISOmap	Keep the geodesic distance relationship well.	Time consuming. Sensitive to outliers. Manifold should be convex. Should choose K.
Laplacian Eigenmap	Performs well on linear and nonlinear data. Efficient	Have to choose factors K and σ. Convex needed.
LLE	Performs well on linear and nonlinear data. Efficient	Have to choose K.Sensitive to noises. Require locally linear structure.Convex needed.
LDA	Performs well on grouped data. Works for both linear data and nonlinear data.	Doesn't work for non labeled data. Doesn't work when group centers are the same. Largest dimension is the number of groups-1.

Table 4: Dimension Reduction Techniques Evaluation

• Suggestions

To choose appropriate dimension reduction method to get better visualization effects, based on the experiments results we have, we suggest:

• PCA algorithm is a good candidate for linear data dimension reduction since it is simple, easy to understand and effectively. Its efficiency will not decrease dramatically along with the increase of data size is another reason to consider it. MDS generally has the same effect with PCA but is much slower than PCA.

• If we don't know the data values but only know the data distance matrix, potential techniques include MDS, ISOmap and Laplacian Eigenmap. They can reconstruct the data without knowing the original data value. Among the 3 techniques, if we have known that the data set is linear and data size is less than 1000, we may try MDS; or else Laplacian Eigenmap is an appropriate choice, since it works for linear and nonlinear data, and it is efficient.

• ISOmap is more sensitive to outliers than other techniques, so if a data set has quite a few outliers, better to avoid ISOmap.

• ISOmap is extremely slow, so if the data size is large (as over 5000) , better to avoid ISOmap.

• If a data set is sparse, choose LLE or Laplacian Eigenmaps. Other techniques tends to generate distort results which interpret the hidden attribute misleadingly.

• If a data set has labels, we definitely should try LDA to get the best visualization effects since it is the only supervised dimension reduction method among the 6 methods. Other possible choices include LLE and Laplacian Eigenmaps. We need to be aware that LLE and Laplacian Eigenmaps may compress a cluster into a line, under extreme circumstance, it even can be several points.

3.10.5 Future Work

In the future, I would like to explore more data dimension techniques such as t−SNE, diffusion map and neural network. Also I would like to implement these dimension reduction techniques to larger size real data, as text file or image data. Combining dimension reduction techniques with prediction or clustering is also something that can be explored in the future.

A Acknowledgment

I would like to thank Professor Guangliang Chen for providing valuable advice and steady guidance throughout the writing of this paper, as well as Professor Andrea Gottileb and Professor David Anastasiu for their detailed suggestions. This report would have been impossible to complete without their wonderful support.

B References

References

[1] Michael Friendly (2008). Milestones in the history of thematic cartography, statistical graphics, and data visualization. Available at https://www.math.yorku.ca/SCS/Gallery/milestone/milestone.pdf/

[2] Pearson, K. (1895). "Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material". Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 186: 343-414.

[3] Richard Ernest Bellman; Rand Corporation (1957). Dynamic programming. Princeton University Press. ISBN 9780691079516.

[4] Brinton, Willard Cope(1914): Graphic methods for presenting facts. The Engineering magazine company. Available at https://archive.org/details/graphicmethodsfo00brinrich

[5] United States Patent and Trademark Office: registration No.75263259". 1993-09-01.

[6] Wilkinson, Leland; Friendly, Michael (May 2009): The History of the Cluster Heat Map. The American Statistician. 63 (2): 179-184. doi:10.1198tas.2009.0033

[7] DeAngelis, G. C.; Ohzawa, I.; Freeman, R. D. (October 1995). "Receptive-field dynamics in the central visual pathways". Trends Neurosci. 18 (10): 451-8. doi:10.10160166-2236(95)94496-R. PMID 8545912

[8] Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. Neural Computation 15, no. 6 (February 6, 2011): 13731396.

[9] Pearson, K. (1901). "On Lines and Planes of Closest Fit to Systems of Points in Space" (PDF). Philosophical Magazine. 2 (11): 559-572. doi:10.1080/1478644010946272

[10] Jonathon Shlens. A Tutorial on Principal Component Analysis. Available at https://arxiv.org/pdf/1404.1100.pdf

[11] Jolliffe I.T. Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., Springer, NY, 2002, XXIX, 487 p. 28 illus. ISBN 978-0-387-95442-4

[12] Joshua B. Tenenbaum. Vin de Silva, John C. Langford.A Global Geometric Framework for Nonlinear Dimensionality Reduction. Available at http://science.sciencemag.org/content/290/5500/2319/

[13] Quan Wang. Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models. Avaliable at https://arxiv.org/pdf/1207.3538v3.pdf/

[14] Fisher, R. A. (1936). "The Use of Multiple Measurements in Taxonomic Problems". Annals of Eugenics. 7 (2): 179-188. doi:10.1111/j.1469--1809.1936.tb02137.x. hdl:2440/1522

[15] McLachlan, G. J. (2004). Discriminant Analysis and Statistical Pattern Recognition. Wiley Interscience. ISBN 0-471-69115-1. MR 1190469

[16] Shi Yu, Lèon−Charles Tranchevent, Bart Moor, Yves Moreau, Kernel-based Data Fusion for Machine Learning. Methods and Applications in Bioinformatics and Text Mining, Ch. 2, Springer, 2011.

[17] L.J.P. van der Maaten. Accelerating t-SNE using Tree-Based Algorithms. Journal of Machine Learning Research 15(Oct):3221-3245, 2014.

[18] van der Maaten, L.J.P.; Hinton, G.E. (Nov 2008). Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research. 9: 2579-2605.

C Code

```
require(MASS)
library(ggplot2)

% Load data
data(iris)
head(iris, 3)

r <- lda(formula = Species ~ .,
         data = iris,
         prior = c(1,1,1)/3)

r$counts
r$means
r$scaling
r$svd

prop = r$svd^2/sum(r$svd^2)
prop

iris.matrix<-as.matrix(iris[,-5])
iris.new<-iris.matrix %*% r$scaling
iris.lda<-as.data.frame(cbind(iris.new,as.character(iris$Species)))
iris.lda$LD1<-as.numeric(as.character(iris.lda$LD1))
iris.lda$LD2<-as.numeric(as.character(iris.lda$LD2))

```

```

setTimeLimit(cpu = Inf, elapsed = Inf, transient = FALSE)
fill=c("#E69F00","#009E73",  "#D55E00", "#CC79A7")

p<-ggplot(iris.lda,  aes(x=LD1,y=LD2,fill=as.character(V3)))+
  geom_point(shape=21,size=3, colour="#000000")+
  ggtitle("2-D LDA Result on Iris Data")+
  xlab("X1")+ylab("X2")+
  scale_fill_manual(values = fill)+
  theme(legend.position="bottom", legend.direction="horizontal",
        legend.box = "horizontal", legend.title = element_blank(),
        legend.text=element_text(size=12),
        axis.line = element_line(size=1, colour = "black"),
        axis.text.x=element_text(colour="black", size = 11),
        axis.text.y=element_text(colour="black", size = 11)+
        title = element_text(size=15,face="bold"))
p

p<-ggplot(iris,  aes(x=Sepal.Length,y=Sepal.Width,fill=Species))+
  geom_point(shape=21,size=3, colour="#000000")+
  ggtitle("Original Iris Data")+
  xlab("Sepal.Length")+ylab("Sepal.Width")+
  scale_fill_manual(values = fill)+
  theme(legend.position="bottom", legend.direction="horizontal",
        legend.box = "horizontal", legend.title = element_blank(),
        legend.text=element_text(size=12),
        axis.line = element_line(size=1, colour = "black"),
        axis.text.x=element_text(colour="black", size = 11),
        axis.text.y=element_text(colour="black", size = 11)+
        title = element_text(size=15,face="bold"))
p

p<-ggplot(iris.lda,  aes(x=LD1,y=0,fill=as.character(V3)))+
  geom_point(shape=21,size=3, colour="#000000")+
  ggtitle("1-D LDA Result on Iris Data")+
  xlab("X1")+
  scale_fill_manual(values = fill)+
  theme(legend.position="bottom", legend.direction="horizontal",
        legend.box = "horizontal", legend.title = element_blank(),
        legend.text=element_text(size=12),
        axis.line = element_line(size=1, colour = "black"),
        axis.text.x=element_text(colour="black", size = 11),
        axis.text.y=element_text(colour="black", size = 11)+
        title = element_text(size=15,face="bold"))
p

%pca
s  <- prcomp(iris[,1:4], scale. = F, center = T)
iris.matrix<-as.matrix(iris[,-5])
iris.new<-iris.matrix %*% r=s$rotation                          % new
    coordinate

iris.pca<-as.data.frame(cbind(iris.new,as.character(iris$Species)))
iris.pca$LD1<-as.numeric(as.character(iris.pca$LD1))      % factor
```

```
    should be trans to char first then to numeric
iris.pca$LD2<-as.numeric(as.character(iris.lda$LD2))      % factor
    should be trans to char first then to numeric


p<-ggplot(iris.pca, aes(x=LD1,y=LD2,fill=as.character(V3)))+
  geom_point(shape=21,size=3, colour="#000000")+
  ggtitle("2-D PCA Result on Iris Data")+
  xlab("X1")+ylab("X2")+
  scale_fill_manual(values = fill)+
  theme(legend.position="bottom", legend.direction="horizontal",
        legend.box = "horizontal", legend.title = element_blank(),
        legend.text=element_text(size=12),
        axis.line = element_line(size=1, colour = "black"),
        axis.text.x=element_text(colour="black", size = 11),
        axis.text.y=element_text(colour="black", size = 11)+
        title = element_text(size=15,face="bold"))
p

p<-ggplot(iris, aes(x=Sepal.Length,y=Sepal.Width,fill=Species))+
  geom_point(shape=21,size=3, colour="#000000")+
  ggtitle("Original Iris Data")+
  xlab("Sepal.Length")+ylab("Sepal.Width")+
  scale_fill_manual(values = fill)+
  theme(legend.position="bottom", legend.direction="horizontal",
        legend.box = "horizontal", legend.title = element_blank(),
        legend.text=element_text(size=12),
        axis.line = element_line(size=1, colour = "black"),
        axis.text.x=element_text(colour="black", size = 11),
        axis.text.y=element_text(colour="black", size = 11)+
        title = element_text(size=15,face="bold"))
p

p<-ggplot(iris.lda, aes(x=LD1,y=0,fill=as.character(V3)))+
  geom_point(shape=21,size=3, colour="#000000")+
  ggtitle("1-D LDA Result on Iris Data")+
  xlab("X1")+
  scale_fill_manual(values = fill)+
  theme(legend.position="bottom", legend.direction="horizontal",
        legend.box = "horizontal", legend.title = element_blank(),
        legend.text=element_text(size=12),
        axis.line = element_line(size=1, colour = "black"),
        axis.text.x=element_text(colour="black", size = 11),
        axis.text.y=element_text(colour="black", size = 11)+
        title = element_text(size=15,face="bold"))
p

```

```
close all; clear; clc;
load MNISTDigit.mat
temp=horzcat(trainImages, trainLabels);
temp=temp(40001:60000,:);
whos trainImages;
group=temp(:,785);
```

```
whos group;
keyindex=find(group==1|group==2|group==3|group==4|group==5);
whos keyindex;
all1to5=temp(keyindex,1:785);
whos all1to5;
all1to5=all1to5(1:3000,:);
all1to5Ordered=sortrows(all1to5,785);

t1=clock;

g=unique(all1to5Ordered(:,785));
c=zeros(size(g));
for i=1:length(g)
c(i)=length(find(all1to5Ordered(:,785)==g(i)));
end

all1to5NoLabel = all1to5Ordered(:,1:784);
whos all1to5NoLabel;
data=all1to5NoLabel;
N = c;
reduced_dim=2;
C=length(N);
dim=size(data',1);

pos=zeros(C,2);
for i=1:C
START=1;
if i>1
START=START+sum(N(1:i-1));
end
END=sum(N(1:i));
pos(i,:)=[START END];
end

UI=[];
for i=1:C
if pos(i,1)==pos(i,2)
UI=[UI;data(pos(i,1),:)];
else
UI=[UI;mean(data(pos(i,1):pos(i,2),:))];
end
end

U=mean(data);
SB=zeros(dim,dim);
for i=1:C
SB=SB+N(i)*((UI(i,:)-U)'*(UI(i,:)-U));
end

SW=zeros(dim,dim);
for i=1:C
for j=pos(i,1):pos(i,2)
SW=SW+(data(j,:)-UI(i,:))'*(data(j,:)-UI(i,:));
end
```

```
end

SW=SW/sum(N);
SB=SB/sum(N);
matrix=pinv(SW)*SB;
[V,D]=eig(matrix);
[c,ind]=sort(diag(D),'descend');
V2=V(:,ind);
V2=V2(:,1:2);

reduced_data=data*V2;
t2=clock;
estiLDA=t2-t1

reduced_data(:,1)=1000*reduced_data(:,1);
reduced_data(:,2)=1000*reduced_data(:,2);
ReducedWithLabel=horzcat(reduced_data,all1to5Ordered(:,785));

figure;
plot(ReducedWithLabel(1:pos(1,2),1),ReducedWithLabel(1:pos(1,2),2), 'b*'); title ('LDA on MNIST digit 1-5')
xlabel('PC1');ylabel('PC2');axis('equal');
grid on;
hold on;
plot(ReducedWithLabel(pos(2,1):pos(2,2),1),ReducedWithLabel(pos(2,1):pos(2,2),2), 'mp');
grid on;
hold on;
plot(ReducedWithLabel(pos(3,1):pos(3,2),1),ReducedWithLabel(pos(3,1):pos(3,2),2), 'k*');
grid on;
hold on;
plot(ReducedWithLabel(pos(4,1):pos(4,2),1),ReducedWithLabel(pos(4,1):pos(4,2),2), 'gd');
grid on;
hold on;
plot(ReducedWithLabel(pos(5,1):pos(5,2),1),ReducedWithLabel(pos(5,1):pos(5,2),2), 'r^');
hold off;
legend ('1','2','3','4','5')
```