DATA VISUALIZATION IN EXPLORATORY DATA ANALYSIS: AN OVERVIEW OF METHODS AND TECHNOLOGIES

by

YINGSEN MAO

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2015

Acknowledgements

First I would like to express my deepest gratitude to my thesis advisor, Dr. Sikora. I came to know Dr. Sikora in his data mining course in Fall 2013. I knew little about data mining before taking this course and it is this course that aroused my great interest in data mining, which made me continue to pursue knowledge in related field. In Spring 2015, I transferred to thesis track and I appreciated Dr. Sikora's acceptance to be my thesis adviser, which made my academic research possible. Because this is my first time diving myself into academic research, I'm thankful for Dr. Sikora for his valuable guidance and patience throughout the thesis. Without his precious support this thesis would not have been possible.

I also would like to thank the members of my Master committee, Dr. Wang and Dr. Zhang, for their insightful advices, and the kind acceptance for being my committee members at the last moment.

November 17, 2015

Abstract

DATA VISUALIZATION IN EXPLORATORY DATA ANALYSIS: AN OVERVIEW OF METHODS AND TECHNOLOGIES

Yingsen Mao, MS

The University of Texas at Arlington, 2015

Supervising Professor: Riyaz Sikora

Exploratory data analysis (EDA) refers to an iterative process through which analysts constantly ‘ask questions’ and extract knowledge from data. EDA is becoming more and more important for modern data analysis, such as business analytics and business intelligence, as it greatly relaxes the statistical assumption required by its counterpart—confirmation data analysis (CDA), and involves analysts directly in the data mining process. However, exploratory visual analysis, as the central part of EDA, requires heavy data manipulations and tedious visual specifications, which might impede the EDA process if the analyst has no guidelines to follow. In this paper, we present a framework of visual data exploration in terms of the type of variable given, using the effectiveness and expressiveness rules of visual encoding design developed by Munzner [1] as guidelines, in order to facilitate the EDA process. A classification problem of the Titanic data is also provided to demonstrate how the visual exploratory analysis facilitates the data mining process by increasing the accuracy rate of prediction.

In addition, we classify prevailing data visualization technologies, including the layered grammar of ggplot2 [2], the VizQL of Tableau [3], d3 [4] and Shiny [5], as grammar-based and web-based, and review their adaptability for EDA, as EDA is discovery-oriented and analysts must be able to quickly change both what they are viewing and how they are viewing the data.

Table of Contents

List of Illustrations

List of Tables

Chapter 1 An Introduction of Exploratory data analysis

Confirmation data analysis (CDA), which is well known as statistical hypothesis testing, has dominated a long history of statistical data analysis and became the hallmark of the first half of the twentieth century. Exploratory data analysis (EDA), on the contrary, had not been widely used until the groundbreaking work of Tukey [6] [7]. Tukey [7] stated that exploratory data analysis is about looking at data to see what it seems to say. He argued that data analysis is not just testing a pre-defined hypothesis and cannot be reduced to a single set of isolated calculations. EDA is especially important for today's data analysis tasks for several reasons. Perhaps the most important one is that EDA does not assume pre-specified hypothesis. Traditional data analysis assumes the analyst has basic knowledge of data and have well-defined question to ask from the data. For example, CDA, namely statistical hypothesis testing, requires the analyst to know what hypothesis to test beforehand. However, as data emerges exponentially in various fields in different kind of forms, we may not know what we are looking for until we iteratively extract knowledge from data and update our understanding of data as we go through EDA. In addition, Data science platforms, such as Kaggle and KDD cup which host data prediction competitions, are becoming more and more popular in recent years. Mu [8] stated that these prediction contests are changing the landscape of how research is traditionally conducted in predictive analytics. 'Rubbish in, rubbish out' is a consensus among these communities, and proper data cleaning and feature engineering is the key to success [9]. As Xavier Conort [10], winner of Flight Quest challenge on Kaggle, said that the prediction algorithms that most Kaggle participants use are very common and standard. It is the appropriate feature engineering that boosts the prediction accuracy. Basically, feature engineering is a trial-and-error process of uncovering hidden feature that better represent underlying problems from raw data. According to our experience, EDA facilitates feature engineering because of its discovery-oriented nature. The goal of EDA is to discover the structure, find patterns and identify relationships, which are central for the feature engineering. Besides, EDA helps to prevent common statistical problems. Most statistical techniques requires special assumptions before they could be employed, and EDA can investigate these assumptions. Zuur [11] argued that EDA can avoid type 1 and type 2 statistical errors and he developed a protocol for the EDA process.

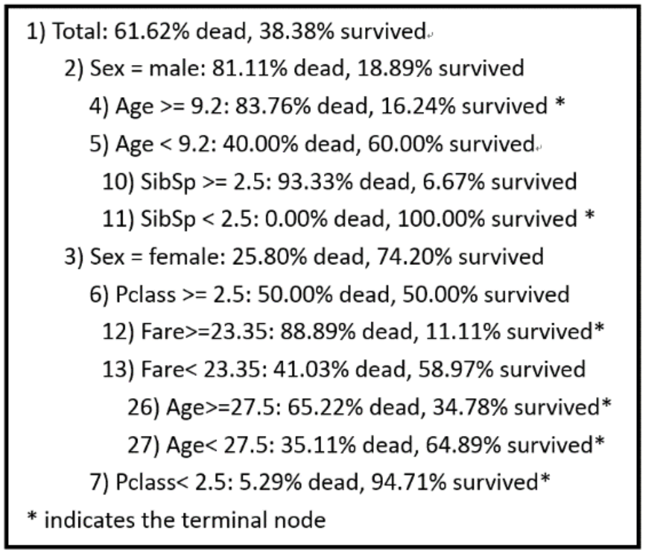
Visual data exploratory (VDE) forms the core of EDA. Basically, the process of VDE is like this: the analyst approach the data and presents the data in visual forms, from which the analyst gains expected or unexpected insights, based on which the analyst generates hypotheses and adjusts the visual representation, then the analyst comes up with new hypotheses and the cycle continues. Thus, VDE is a process of hypothesis, experiment and discovery, with data presenting in visual forms. Shneiderman [12] summarized the process as: overview first, zoom and filter, and then details-on-demand. VDE aims at integrating perceptual ability of human and computation power of today's computer in the knowledge-discovery process, and the advantage is that the perceptual system of human outperforms computer power in terms of detecting patterns and anomalies, and the computer is far more efficient in large amount of automated computations than human. However, the major challenge is that most accessible visualization technologies requires the analyst to manually encode visual specifications. The tedious encoding design of visualization might hinder the exploration process as analysts are assumed to have the visualization design knowledge. The goal of this paper is to solve this problem.

We organize this paper as two main parts corresponding two indispensable components of VDE—human and computer. In first part, we present two-dimensional (2D) visualization of all the possible combinations of continuous variables and categorical variables, as continuous variables and categorical variables are the most common data types in data analysis tasks and 2D display is much more preferable and expressive than high-dimensional display (3D) [1]. We illustrate the ‘behind the scene’ principle of each visual representation such as what the data table looks like, what descriptive statistics are needed, and what visual encodings are applied. The classification problem is also provided to demonstrate how VDE facilitates feature engineering, which increases the accuracy rate of decision tree and random forest algorithms. The second part of paper explores cutting-edge visualization technologies and reviews their compatibilities with VDE. This is because not all visualization technologies are suitable for VDE, as VDE requires analysts to quickly change what they are viewing and how they are viewing the data in the hypothesis and experimentation process. Some visualization technologies, especially web-based visualization, are designed for presentation purposes, that is, they allow sophisticated and highly interactive designs but required tedious specification designs, which greatly prolongs the development process. However, some emerging web-based technologies that are developed with declarative domain specific languages, such as Shiny [5], greatly facilitate the exploratory process.

Chapter 2 Why Visualization is Important

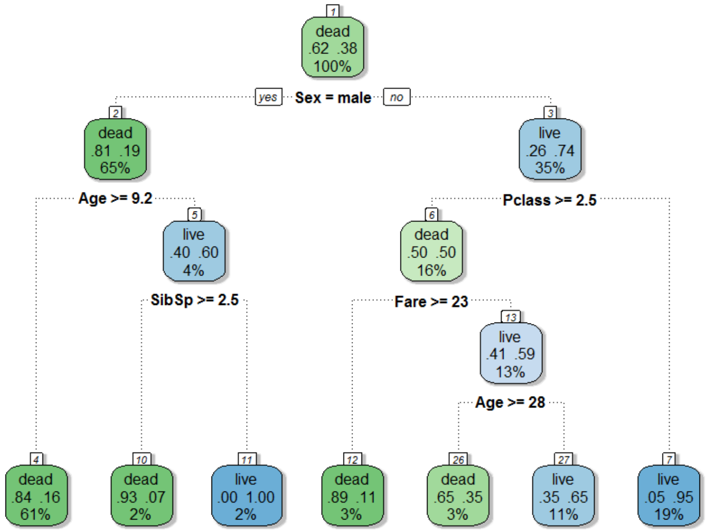
It is important to know what the advantage of visualization techniques are and why we should use it for EDA. In this chapter, two examples are given to illustrate that the visualization conveys information in a much more efficient and accurate manner than text description and data table do.

Larkin and Simon [13] argued that graphs arouse human’s perceptual system to make perceptual inference. They stated that graphs group together all relevant information and avoid a large number of search of non-relevant information. For example, both Figure 2-1 and Figure 2-2 demonstrate the rule of decision tree algorithm. The only difference is that Figure 2-1 uses the text description, whereas Figure 2-2 employs the visual form of decision tree. It is clear that the graph representation conveys information in more effective and efficient way than pure text description. For example, if we want to explore the survival rate of a male passenger with more than 2 siblings or parents, it is difficult to locate this piece of rule among all the rules but we can find it easily in the tree graph.



```
1) Total: 61.62% dead, 38.38% survived
   2) Sex = male: 81.11% dead, 18.89% survived
      4) Age >= 9.2: 83.76% dead, 16.24% survived *
      5) Age < 9.2: 40.00% dead, 60.00% survived
        10) SibSp >= 2.5: 93.33% dead, 6.67% survived
        11) SibSp < 2.5: 0.00% dead, 100.00% survived *
   3) Sex = female: 25.80% dead, 74.20% survived
      6) Pclass >= 2.5: 50.00% dead, 50.00% survived
        12) Fare>=23.35: 88.89% dead, 11.11% survived*
        13) Fare< 23.35: 41.03% dead, 58.97% survived
           26) Age>=27.5: 65.22% dead, 34.78% survived*
           27) Age< 27.5: 35.11% dead, 64.89% survived*
      7) Pclass< 2.5: 5.29% dead, 94.71% survived*
* indicates the terminal node
```

Figure 2-1 Text representation for decision rules



Rattle 2015-10月-04 14:02:49 Yingsen

Figure 2-2 Visual representation of decision rules

Besides, visualization is able to show data in details, which is considered as a valuable quality that not possessed by other analysis techniques. As Cleveland [14] put that a small number of numerical values [summary statistics] do not retain wealth of information in the data. A good example is Anscombe’s Quartet [15]. The raw data is shown in Table 2-1, and the corresponding plot is shown in Figure 2-3. It illustrates how datasets that have identical statistical summary can have very different structure that are immediately obvious when the dataset is shown graphically

	1		2		3		4	
	x1	y1	x2	y2	x3	y3	x4	y4
	10	8.04	10	9.14	10	7.46	8	6.58
	8	6.95	8	8.14	8	6.77	8	5.76
	13	7.58	13	8.74	13	12.74	8	7.71
	9	8.81	9	8.77	9	7.11	8	8.84
	11	8.33	11	9.26	11	7.81	8	8.47
	14	9.96	14	8.1	14	8.84	8	7.04
	6	7.24	6	6.13	6	6.08	8	5.25
	4	4.26	4	3.1	4	5.39	19	12.5
	12	10.84	12	9.13	12	8.15	8	5.56
	7	4.82	7	7.26	7	6.42	8	7.91
	5	5.68	5	4.74	5	5.73	8	6.89
Mean	9	7.501	9	7.501	9	7.5	9	7.501
Variance	10	3.752	10	3.752	10	3.748	10	3.748
Correlation	0.816		0.816		0.816		0.816	

Table 2-1 Anscombe's Quartet: raw data

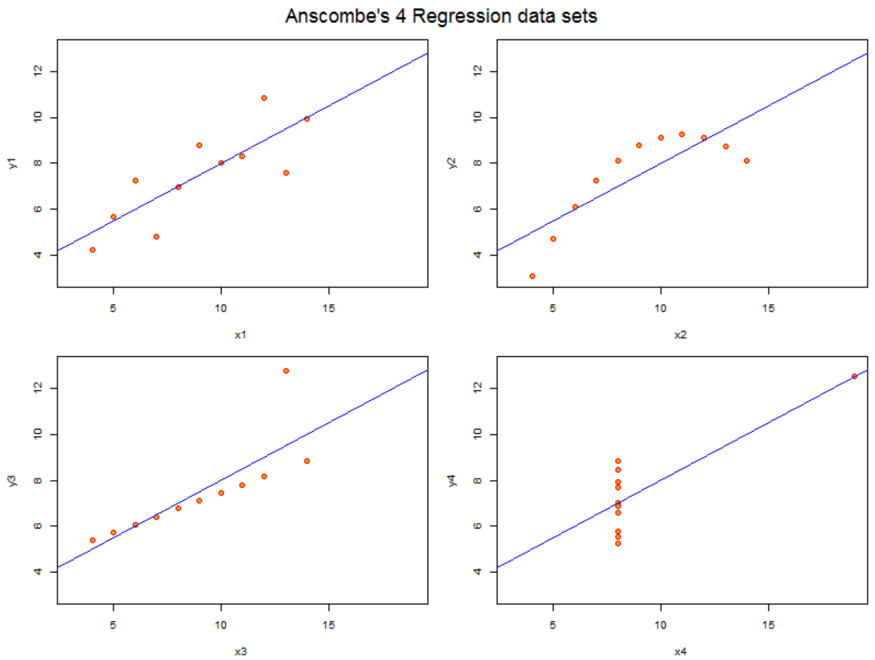


Figure 2-3 Anscombe's Quartet

In this chapter we use two examples to prove that graphs convey information in an effective and detailed way. Analysts can identify outliers, trend and patterns hidden in data by using appropriate visualization techniques. In the next chapter, we will explore the design rules analyst should follow in order to make effective and expressive graphs.

Chapter 3 Expressiveness and Effectiveness for Visualization Specification

Before diving into specific visualization techniques, it is necessary to introduce the principle of visual encoding, as an effective visual encoding conveys accurate information and inappropriate visualizations district users. This is critical for visualizing high dimensional variables, which is discussed in chapter 6.

The visual encoding applied in this paper is heavily affected by the expressiveness and effectiveness rules developed by Munzner [1]. According to Munzner [1], marks and channels are two fundamental components of visualization representations. Marks are basic geometric elements that depict data observations. Figure 3-1 is an example for marks that include points, lines and areas.

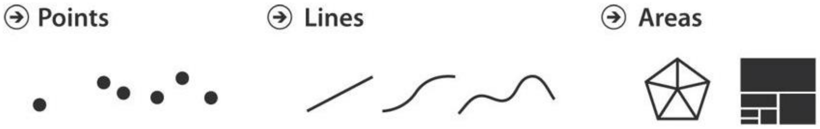


Figure 3-1 Marks represent the basic geometric object. From [1] Fig 5.2

Channels, on the other hand, control appearance of marks. It includes position, color, size, etc. which is shown as Figure 3-2.

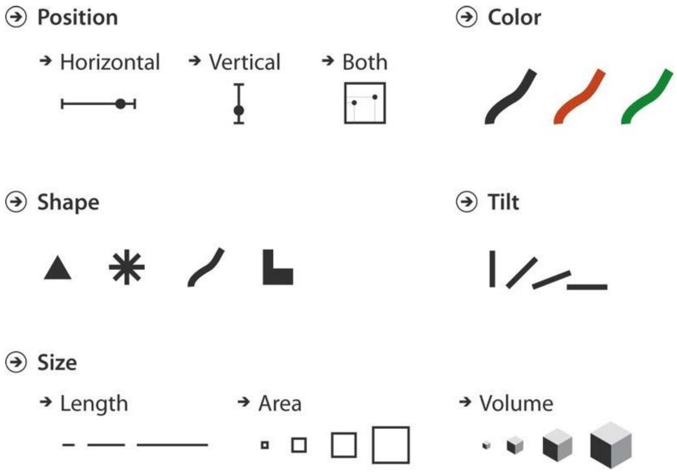


Figure 3-2 Channels controls the appearance of marks. From [1] Fig 5.3

Tamara [1] developed a guideline for how to use visual channels in an effective manner in terms of type of variables and importance of variables. The guideline is shown in Figure 3-3. In Figure 3-3 channels is categorized into identity channels and magnitude channels. Identity channels show what something is or where it is and is designed for categorical variables. In contrast, the magnitude channels display how much of something there is and is designed for quantitative variables. Besides, the rankings range from the most effective at the top and the least effective at the bottom. Two things should be noted are that, first, channels related to the position are the most effective for both continuous variables and categorical variables. Second, color channels are more effective than area/shape channels in conveying the information. Thus, the importance of the variable should match the effectiveness level of channels it employs, that is, the most important variable should be encoded as position channels, then color channels, and then area or shape channels.

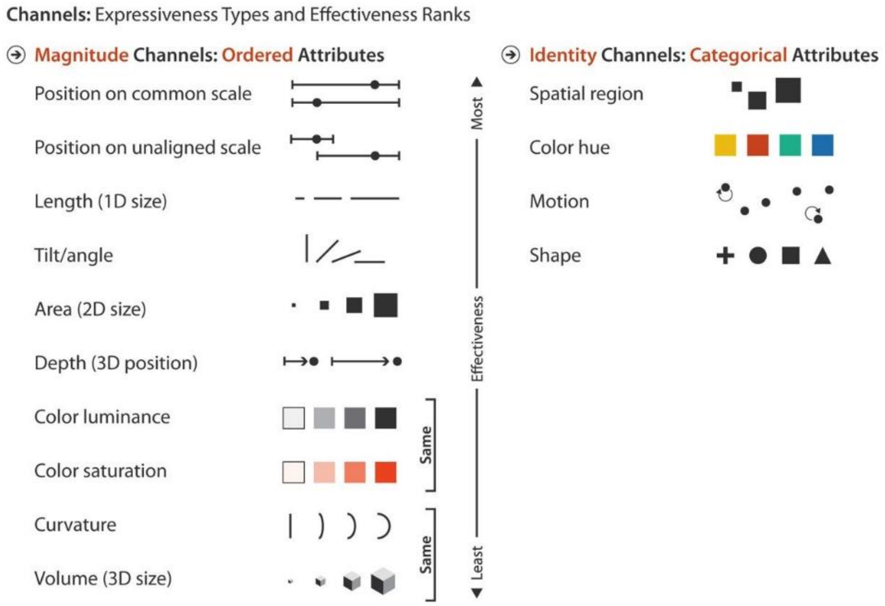


Figure 3-3 Channels ranked by effectiveness according to data and channel types.

From [1] Fig 5.6

In this paper, we will use Tamara's rule as guideline for illustrating visualization techniques.

Chapter 4 Visualizing Categorical Variables

The confirmatory statistics analysis of categorical variables, such as Chi-square test of independence, has been intensively studied early in history, but it was not until late twentieth century that most of its advanced visualization techniques were invented. These techniques include mosaic plot [16], association plot [17], double-decker plots [18], spine plots [19] and spinograms [20], etc, most of which are designed for visualizing multi-dimensional categorical variables (n-way contingency table) and shaded with the result of independence tests.

Mosaic plot was originally invented by Hartigan and Kleiner [16]. It is composed of rectangular tiles (cells) whose area is proportional to cell frequency, aiming to visualize N-way contingency tables and models of associations among its variables [21] [22] [23]. In 1984, Hartigan and Kleiner [16], inventors of the mosaic plot, constructed a mosaic plot of four-way contingency table, which is shown in Figure 4-1, to demonstrate the strength of the mosaic plot, using Nielsen television ratings data in 1977 and 1978. Nielsen television ratings data include four categorical variables, network companies, time of day, day of week, and week of month. The area of each rectangle is proportional to the number of Nielson households watching television on a particular day, viewing a particular network during a particular prime-time half hour. Hartigan and Kleiner [16] suggested that the maximum number of six categorical variables can be visualized by the mosaic plot.

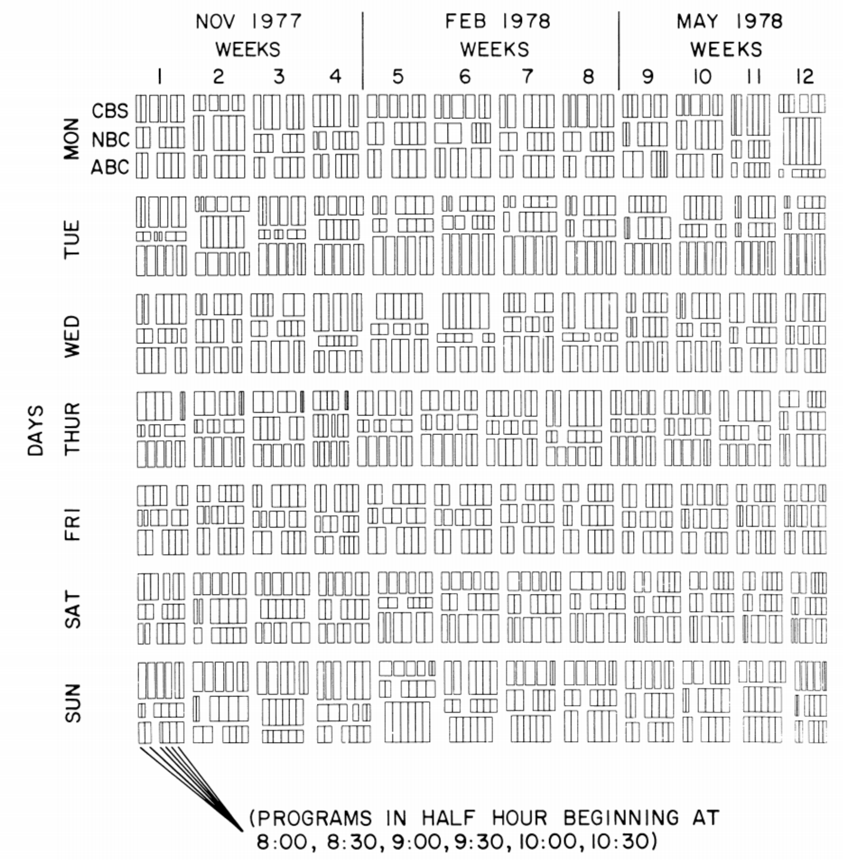


Figure 4-1 Mosaic plot for Nielsen television ratings data – A four way contingency table

An association plot [17](C visualizes the standardized deviations of observed frequencies from those expected under a certain independence hypothesis, with the area of each cell representing the difference between observed and expected frequency. Spine plot [19] is a variation of bar plot for visualizing two categorical variables where the height of bar is rescaled to be constant across levels of conditioning categorical variables.

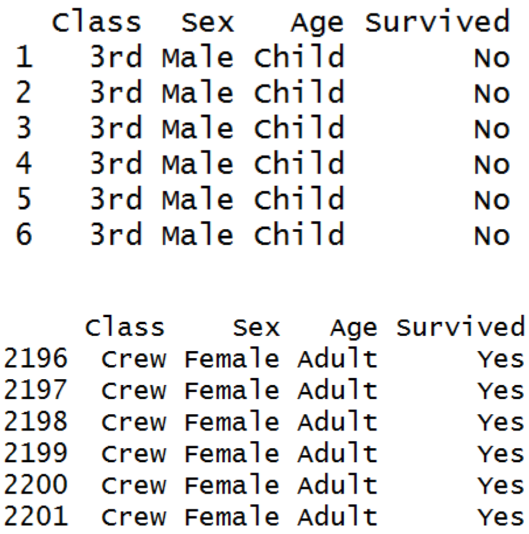
To give you a clear illustration of visualization techniques, the titanic data set consisted of four categorical variables is used across this section. Table 4-1 shows the description of the four categorical variables.

Categorical Variables	Levels
Economic status (class)	1st, 2nd 3rd, Crew
Gender (sex)	Male, Female
age	Adult, Child
Survival	Yes, No

Table 4-1 Titanic data set description

Before the visualization techniques are introduced, several representations of categorical data generating the visualizations are presented. We use the norms defined by Friendly [23]. The three most common representation forms of categorical variables are case form, frequency form, and table form.

Case form is the most common form of data, and most raw data frame exist as case form, with individual observations existing in each row and categorical variables in columns. Table 4-2 is the case form of Titanic data. Case form is unaggregated raw data and gives you a sense of what the data table looks like, such as what variables are included and how many observations in total, but no additional visualization statistics are given to create graphs.

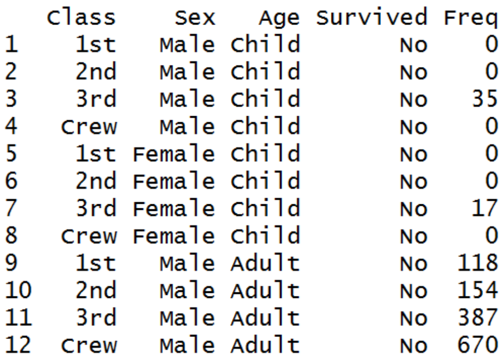


```
  Class  Sex   Age Survived
1   3rd Male Child       No
2   3rd Male Child       No
3   3rd Male Child       No
4   3rd Male Child       No
5   3rd Male Child       No
6   3rd Male Child       No
```

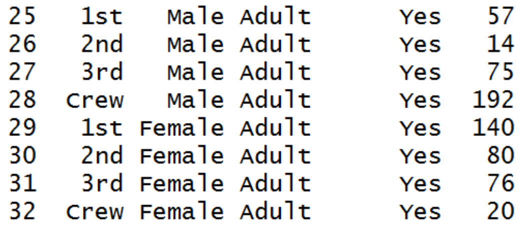
```
     Class    Sex   Age Survived
2196  Crew Female Adult      Yes
2197  Crew Female Adult      Yes
2198  Crew Female Adult      Yes
2199  Crew Female Adult      Yes
2200  Crew Female Adult      Yes
2201  Crew Female Adult      Yes
```

Table 4-2 Case form of Titanic data: Head and tail example

The second form of categorical data, frequency data table, has been tabulated, existing in an aggregated form. Frequency table gives the summary of frequency for all possible combinations of categorical variables. Table 4-3 is an example of frequency data. There are 32 possible combinations of various levels of categorical variables.



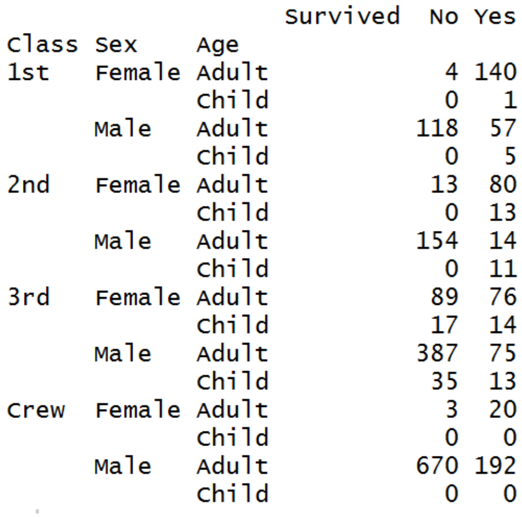
```
   Class    Sex   Age Survived Freq
1    1st   Male Child       No    0
2    2nd   Male Child       No    0
3    3rd   Male Child       No   35
4   Crew   Male Child       No    0
5    1st Female Child       No    0
6    2nd Female Child       No    0
7    3rd Female Child       No   17
8   Crew Female Child       No    0
9    1st   Male Adult       No  118
10   2nd   Male Adult       No  154
11   3rd   Male Adult       No  387
12  Crew   Male Adult       No  670
```



```
25   1st   Male Adult      Yes   57
26   2nd   Male Adult      Yes   14
27   3rd   Male Adult      Yes   75
28  Crew   Male Adult      Yes  192
29   1st Female Adult      Yes  140
30   2nd Female Adult      Yes   80
31   3rd Female Adult      Yes   76
32  Crew Female Adult      Yes   20
```

Table 4-3 Frequency table of Titanic data: head and tail example

Contingency table is another form to display frequency distribution of categorical variables. It is prevalent in the statistics domain as it is the form specialized to conduct the test of independence. Unlike frequency table, which embeds frequencies as an individual column, contingency table treats frequencies as the element of the table and all possible categorical variable combinations as rows and columns of the table. Table 4-4 is an example of the four-way contingency table.

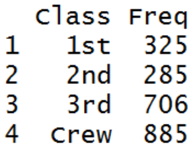


```
                    Survived  No Yes
Class Sex    Age
1st   Female Adult             4 140
             Child             0   1
      Male   Adult           118  57
             Child             0   5
2nd   Female Adult            13  80
             Child             0  13
      Male   Adult           154  14
             Child             0  11
3rd   Female Adult            89  76
             Child            17  14
      Male   Adult           387  75
             Child            35  13
Crew  Female Adult             3  20
             Child             0   0
      Male   Adult           670 192
             Child             0   0
```

Table 4-4 Contingency(four-way) table of Titanic data

The number of categorical variables to be visualized determines what kind of visualization techniques are to be used. The bar chart is best for visualizing individual categorical variable, and it is advantageous that categorical variable takes the form of frequency table, which gives all the statistics needed to create the simple bar chart. Table

4-5 is the frequency table of Class of Titanic data, and Figure 4-2 is the corresponding bar plot with the height of each bar representing the frequency for each class.



```
  Class Freq
1   1st  325
2   2nd  285
3   3rd  706
4  Crew  885
```

Table 4-5 Frequency data of Class of Titanic data

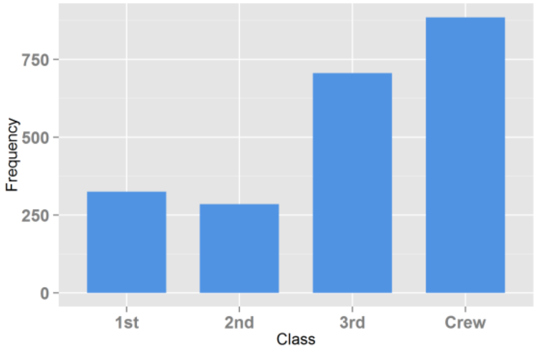


Figure 4-2 Bar Chart of Class of Titanic data

The bar chart has high scalability for the number of levels of categorical variables, and the number of levels can range from two to hundreds. For example, Figure 4-3 visualizes the frequency of the first letter of baby names in 2013. Here the first letter is a categorical variable with 26 levels. The bar chart can be used to look up and compare values. If the bar char is designed specifically for comparison purposes, it is better in the vertical sorted form, which is shown in Figure 4-4.

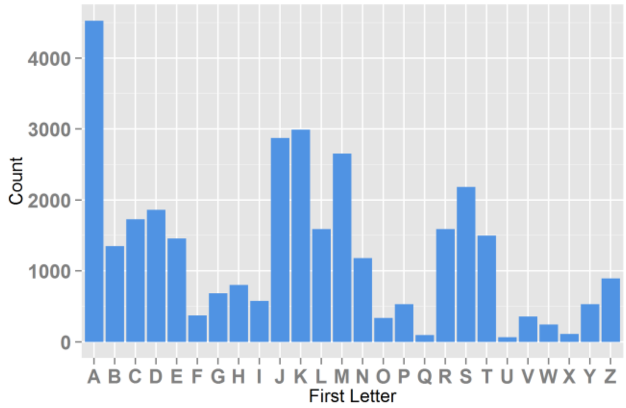


Figure 4-3 Bar Chart of First Letters of Baby Names in 2013

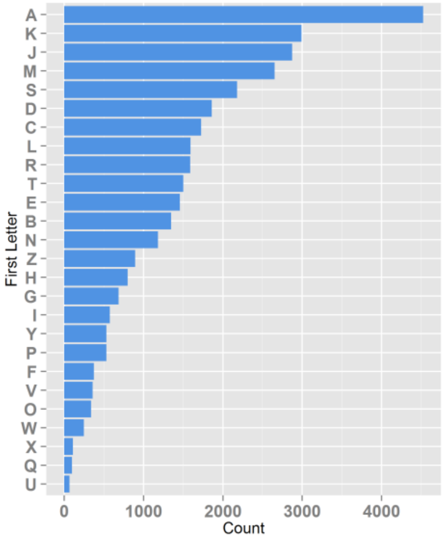
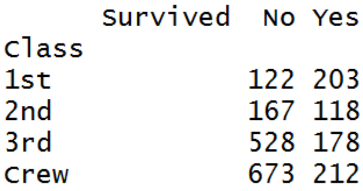


Figure 4-4 Sorted Bar Chart of First Letters of Baby Names in 2013

For visualizing more than one categorical variables, the stacked bar chart, the grouped bar chart, and the spine plot can be used. And N-way contingency table is the preferable form as it gives all the necessary statistics to generate the graph. Figure 4-5 is the stacked bar chart for visualizing two categorical variables in the form of two-way contingency table, which is shown in Table 4-6. In Figure 4-5, the Class is the conditioning variable which is used first for splitting and the Survived is the conditioned variable. Conditioning variables are often used for highlighting and it is one of the most important concepts for mosaic plot. The height of each full bar represents the margin of corresponding rows, and the height of sub-bars in the full bar represents the weight of the conditioned variable within each level of conditioning variables. Thus the stacked bar is used for comparing the frequency of conditioning variable, and the weight of conditioned variable in each level of conditioned variables.



```
      Survived  No Yes
Class
1st            122 203
2nd            167 118
3rd            528 178
Crew           673 212
```

Table 4-6 Two way contingency table of Class and Survived of Titanic Data

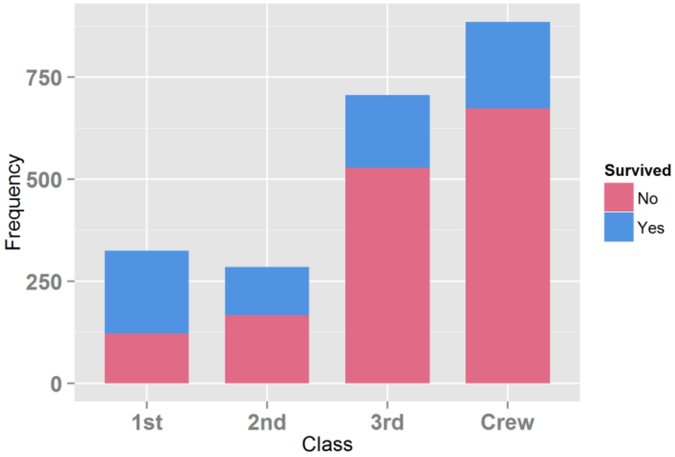


Figure 4-5 Stacked bar chart of Class and Survived of Titanic Data

However, the purpose for comparing the weight of the conditioned variable within each level of conditioning variables becomes vague when there are huge differences among the margin conditioning variables. In Figure 4-6, for example, it is hard to show the distribution of Sex in each levels of the First Letter, because the frequency of the first letter varies dramatically. One solution is to rescale the margin of conditioning variable to 1, as shown in Figure 4-7.

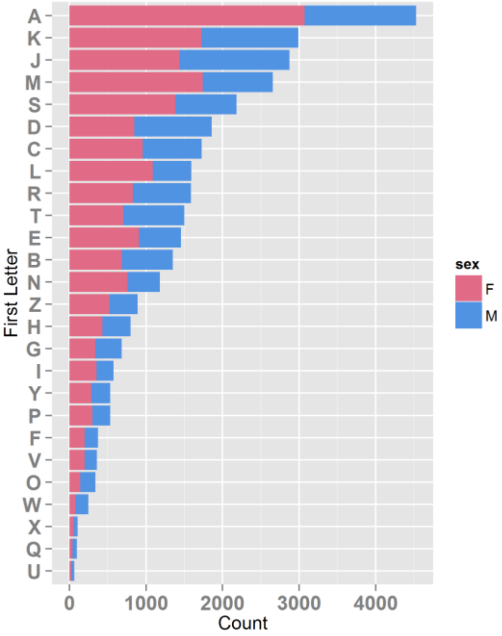


Figure 4-6 Stacked bar chart of first letter of baby names and gender in 2013

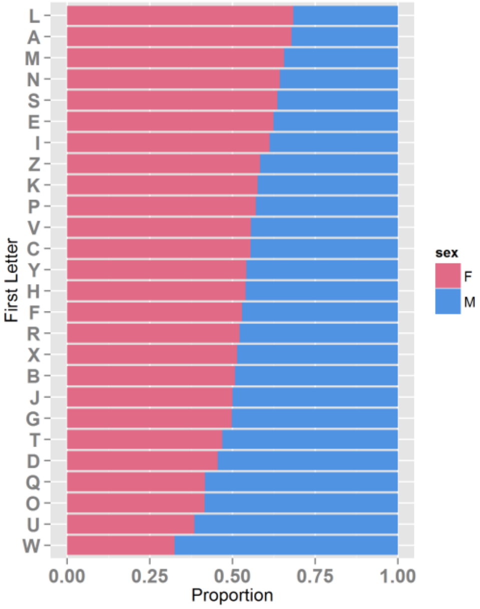


Figure 4-7 Rescaled Stacked bar chart of first letter of baby names and gender in 2013

In Figure 4-7, however, the ability to show the frequency distribution is suppressed due to the height of each bars is rescaled to be the same. This can be solved by using the spine plot as shown in Figure 4-8. The spine plot is a special form of mosaic plot for the two-way contingency table. It is an extension of stacked bar chart with the width of each bar indicating the margin of conditioning variable. However, the spine plot compromises the ability to show the margin distribution of conditioning variable, as the difference of width of bars is not as perceptible as the length of bars for human to make the comparison. Thus as we mentioned at the beginning, there is always a trade-off in visualizations. The 'ideal visualization' always depends on what to be analyzed and which part needs to be focused.

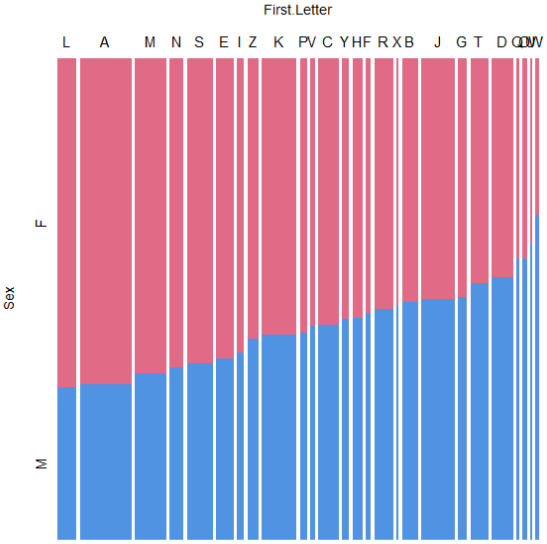


Figure 4-8 Spine plot of the sex and the first letter of the baby names in 2013

The focus of analysis can be shifted from the conditioning variable to the conditioned variable by using the grouped bar chart, as shown in Figure 4-9. The grouped bar chart first split the conditioning variable and then split each level based on the conditioned variable. The absolute frequency distribution of conditioned variable can be compared to each other within each level or across the levels of conditioning variable.

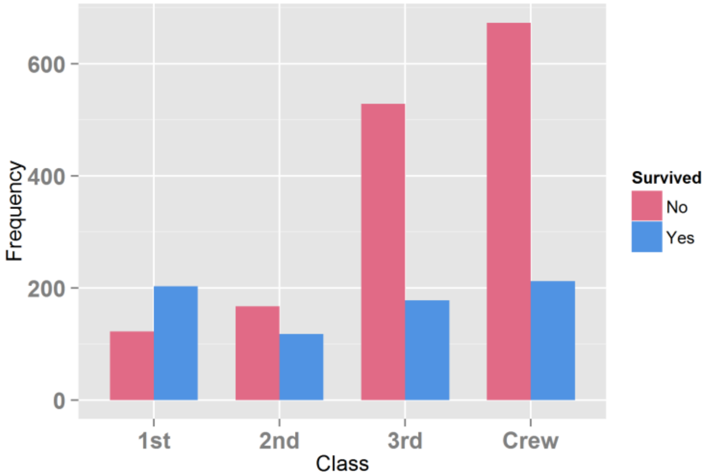


Figure 4-9 Grouped bar chart of class and survived of titanic data

For visualizing more than two categorical variables, the mosaic plot, which is generated from the N-way contingency table, is mostly used. Frendly [21] [22] [23] modified the original version of mosaic plot. In his modified version, the result of test of independence generated by Chi-square test or model-based test is encoded as color channel. Therefore, the relationships of categorical variables, such as how they are related with each other, can also be visualized.

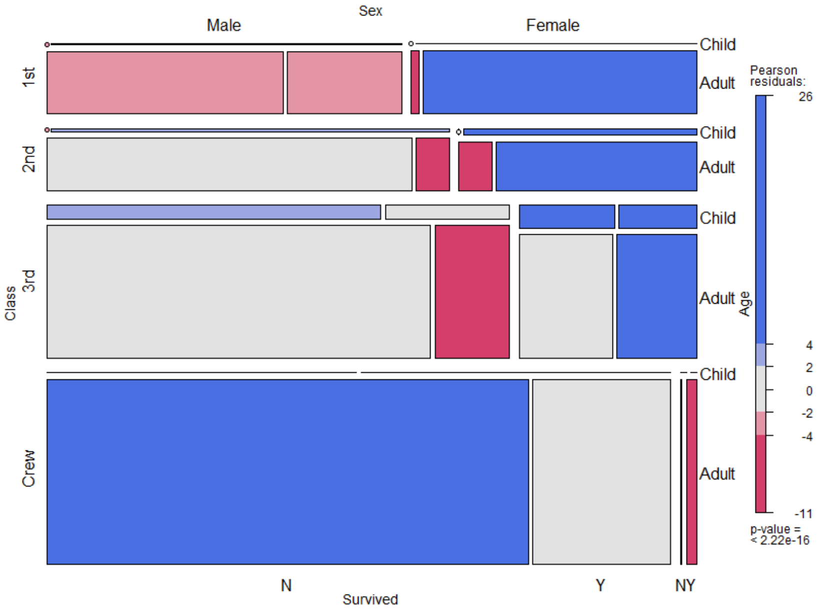


Figure 4-10 Mosaic plot of class, sex, age and survived of Titanic data

Figure 4-10 shows the mosaic plot of a four-way contingency table shown in Table 4-4. The plot is easier to understand if the order of split is given. In Figure 4-10, the order of the split is clock-wised and starts with the Class and ends with the Survived, thus the Class is the first variable to split and the Survived is the last one. Each rectangular cell is shaded based on the Pearson residual, which is the standardized difference between the expected frequency and the observed frequency in that cell. Thus, conclusions such as whether the frequency distribution of each cell is higher or lower than expected can be reached. In figure 4-10, it can be found that the survival rate of female adult from 1st, 2nd and 3rd are much higher than expected, whereas the death rate of 2nd, 3rd and Crew adult are much higher than expected.

Frendly [22] highlighted the unexpected frequency distribution shown in Figure 4-11 in his original work.

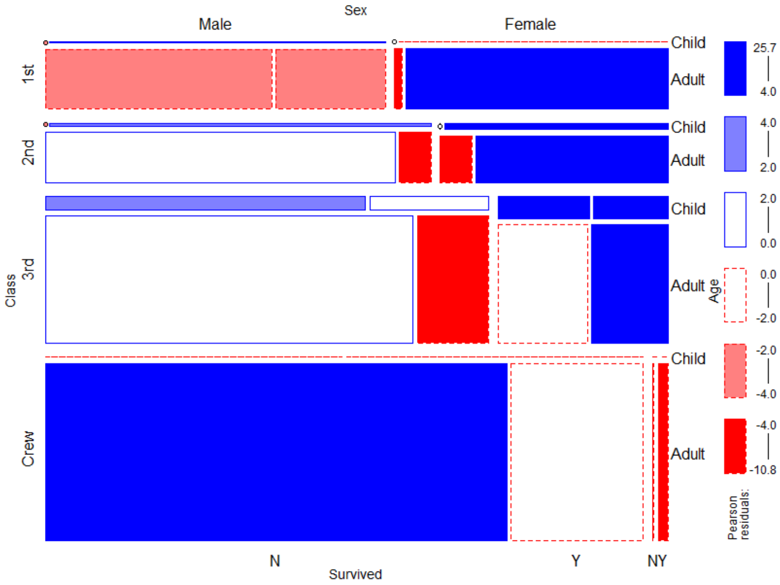


Figure 4-11 Mosaic plot with unexpected frequency distribution highlighted

For visualizing the inference result, mosaic plot can visualize the null model by investigating log-linear models for a contingency table [23]. It is not discussed here as it is not part of exploratory analysis.

Double decker plot is another variation of mosaic plot, which highlights the influence of conditioned variables on conditioning variables. Figure 4-12 shows an example.

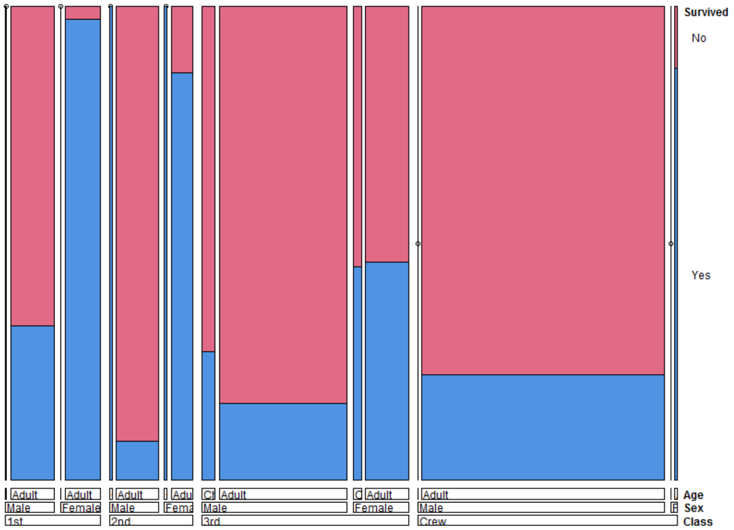


Figure 4-12 Double decker plot of class, sex, age and survived of Titanic data

Chapter 5 Visualizing Continuous Variable

Visualizing quantitative variables is much more straightforward compared to visualize categorical variables, as the attribute can be directly mapped to the spatial position along an axis. Additional attributes can be mapped to non-spatial channels such as shape and color. In addition, visualizing quantitative variables is more meaningful for the statistical inference, as it can be used to verify assumptions required by statistical inference methods. Zuur [11] provided a protocol for EDA to avoid common statistical inference problems. We will start with the univariate variable, i.e. one numerical variable.

Though visualizing the univariate variable tells us nothing about how it is related to any of other variables, it is imperative to do so as it helps update our understanding about the data, and based on what we can adopt different statistical techniques. During the visualization process, a few questions should be asked which are listed as follows.

Are there any outliers? An outlier is an exceptional observation that falls far away from the distribution of majority observations, and it has be dealt with before proceeding. Outliers can be caused by different reasons that can be summarized as two main reasons. The first is that the outlier is an erroneous value that can be simply deleted or replaced with the correct one. The second is that the outlier is produced by a legitimate cause and a special treatment regarding the outlier is needed. However there is no framework to follow to deal with such outliers as the legitimate cause is different case by case. Here we focus on techniques that visualize and detect outliers. For people who are interested in statistical method to handle outliers, see [24] for a complete reference.

How many modes? The mode of empirical distribution refers to the most frequent or densely concentrated segment of the distribution. The modicity indicates the number of modes of variable’s empirical distribution. Visualizing the modicity become important as it has no strict mathematical definitions, and many statistical methods require unimodal variables. Multimodal variables can be an indication that data come from a mixture of distributions.

What are specific characteristics of shape? Here we define specific characteristics of shape as location, spread, skewness and kurtosis. The location measures the overall tendency of the variable and the spread measures the amount of variation around the overall tendency. The two most popular measurements are the mean and the variance. The skewness is a measure of symmetry, i.e. the lack of symmetry. The kurtosis measures whether the data are peaked or flat relative to a normal distribution. The skewness of a normal distribution is zero, and any symmetric data should have a skewness that is near zero. All of these specific characteristics have the mathematics definition.

Is the shape a symmetrical distribution? A variable is distributed symmetrically if the empirical distribution of a variable is divided in the center, each half is a mirror image of the other. The symmetrical distribution cannot be skewed, but it does not have to be unimodal and it can have any type of kurtosis. As Chambers [25] pointed out, the symmetry is important for data analysis for several reasons, the highlighted two are that it is easier to understand the variable with symmetrical distribution, and most of the robust statistic techniques assume that the variable is symmetrical.

Is the shape a normal distribution? The test of normality is probably the main goal of the univariate variable visualization. Because of its inference character, the normal distribution is one of the most important distribution required for the statistical analysis. Samples drawn from the normally distributed variable have a mean and standard deviation that are the most reasonable indicator for the corresponding population mean and standard deviation. Most of the statistical inferences and significant tests require the assumption of normality, and the result would be incorrect if the assumption is violated. Normally distributed variable has the following characteristics:

- It has no outliers and only one mode
- It is not skewed and has no or slight kurtosis.
- It is a symmetrical distribution.

All of these questions should be answered through the visualization. In the rest of the chapter, several visualization techniques are illustrated by using continuous variables from a data table that include mpg (miles/gallon), hp (gross horsepower), wt (weight), and disp (Displacement).

Dotplot. Dotplot is the simplest representation of the distribution of the univariate variable with the lowest level of raw data points showing on the graph. Though the dotplot gives the most detailed information of the numerical variable, its representation becomes obscure as the number of observation increase. One solution to relieve this problem is the jittering. Jittering move the point by a random or systematic amount along the axis that is not the axis displaying the variable value. Figure 5-1 is an example of the dotplot. Each point is ‘jittered’ along the horizontal axis by adding a random variable. From Figure 5-1, one of the findings are that the range of the MPG is from 10 to 35, with the majority of data distributed between 12.5 and 27.5. The distribution is slightly skewed towards the upper value of MPG.

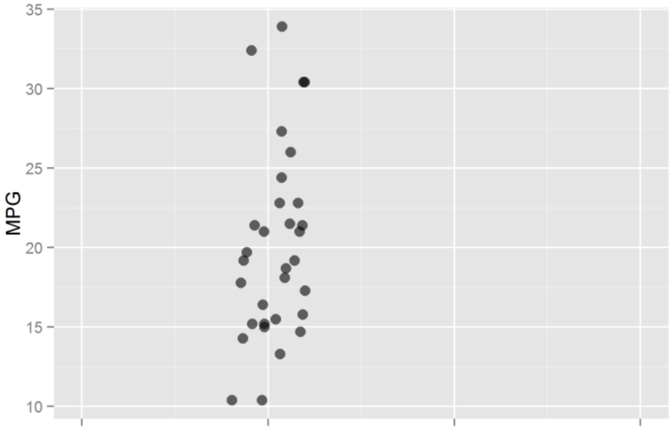


Figure 5-1 Jittered dotplot of MPG

Boxplot. The boxplot can be viewed as an extension of the dotplot, where the dotplot is enhanced by adding statistical information of the variable. The elements of the boxplot include a box with the upper line representing the third quartile, the middle line representing the median, and the lower line representing the first quartile. The two vertical lines extending out of the box reach the highest value and the lowest value that is within 1.5 * IQR of the hinge, where the IQR is the inter-quartile range. Thus, half of the data are inside the box, and the other half outside the box. One quarter of data above the box, and one quarter below the box. Figure 5-2 is the boxplot of MPG variable.

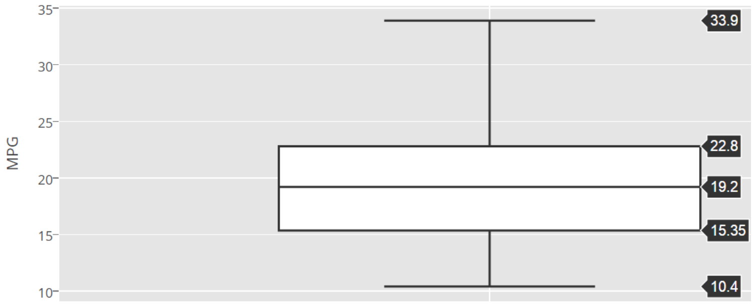
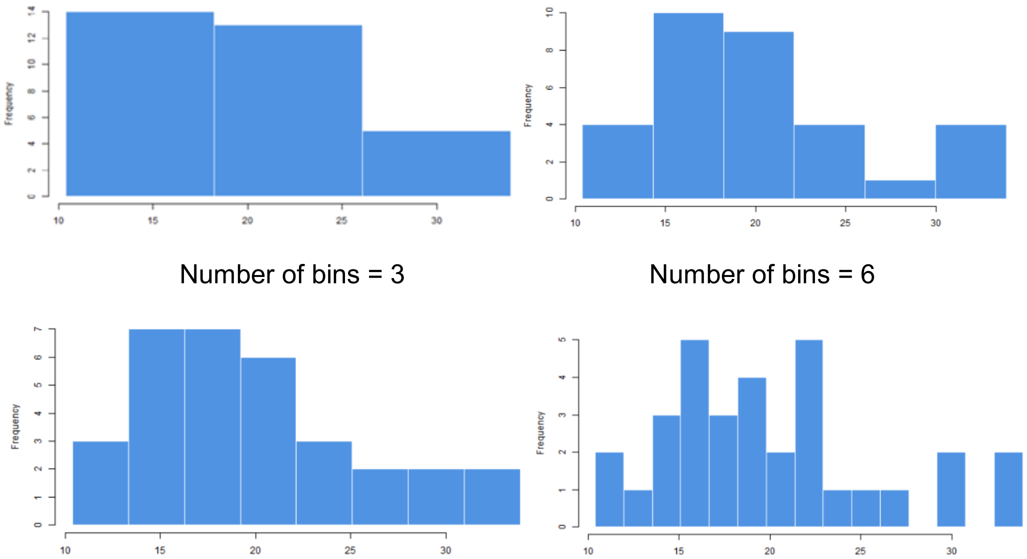


Figure 5-2 Boxplot of MPG

Histogram. A histogram is a graphical method for displaying the shape (distribution) of a univariate variable. It helps identify the center of distribution, estimate the spread of distribution, recognize number of modes, detect outliers and estimate the skewness and kurtosis. The histogram is probably one of the most popular visualization techniques. However, one of the problems the histogram suffer are the selection of bin width. Wand [26] stated that the bin width has an enormous effect on the shape of a histogram. Unsatisfied bin width may lead under-smoothing or over-smoothing histograms and distort the real variable distribution. For example, in Figure 5-3, histograms are created using different number of bins. Each of them seems to tell a different story about the MPG distribution. The histogram produced by six bins suggests an existence of outliers in the right tail, whereas the histogram generated by eight bins indicate no outliers except the distribution is right-skewed.



Number of bins = 3

Number of bins = 6

Number of bins = 8

Number of bins = 15

Figure 5-3 Histogram with different number of bins

Several statisticians provided rules for selecting the bin width. Sturges [27] suggested the bin width as

$$\text{Bin width} = \frac{\text{range}}{1+\log_2 \text{n}}$$

, where n is the number of observation. A lot of statistical software use this rule as the default to determine the bin width [26]. Scott [28] suggest that for the normally distributed variable or symmetric variable that is similar to normal distribution, the bin width is

$$\text{Bin width} = 3.49 * \alpha * n^{-1/3}$$

, where α is the standard deviation.

According to our experience, it is highly recommended to use the interactive visualization technology to determine the number of bins and its major advantages are as follows. First, it greatly accelerates the updating of the histogram as analysts change number of bins. Analysts can constantly pose hypotheses by changing the number of bins and test the hypotheses rapidly by viewing the updated histogram. Second, the exploration of the number of bins is a knowledge updating process for analysts, as analysts are detecting the change of the variable distributions with the different number of bins. For example, as we randomly select three, six, eight, and fifteen as the number of bins, we had a better understanding of the distribution of MPG. We found that the distribution is slightly right-skewed with several outliers existing between 30 and 35, and the histogram produced by six bins seems an optimal choice because it describes every aspect of MPG distribution.

Instead of plotting the frequency of each bin along the vertical axis, we can plot the kernel density curve. The advantage of using the kernel density curve is that it is relatively robust to the number of bins. An example is shown in Figure 5-4.

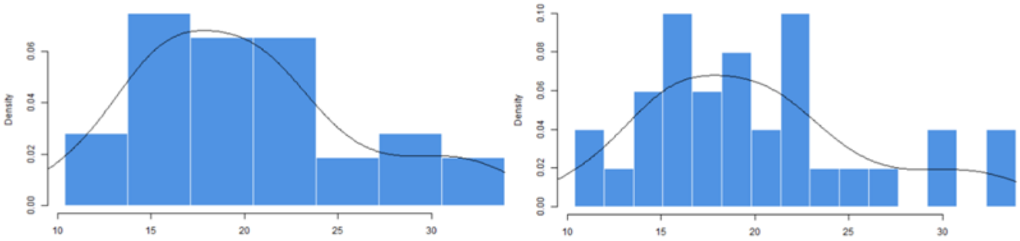


Figure 5-4 Histogram with Kernel density curve

Cumulative Distribution Plots. The cumulative distribution plot, which is commonly called quantile plot or probability plot, is a family of graph techniques that identify underlying deviations from a particular theoretical distribution, such as normal process, Chi-square process. Normal probability plot is probably the most popular cumulative distribution plot, since it helps identify whether a numerical variable is normally distributed. It plots the observed data along vertical axis against the normal score that would obtained if the data is generated from the normal process. Deviations from the straight line suggest departures from the normal process. The normal probability plot of MPG is shown in Figure 5-5.

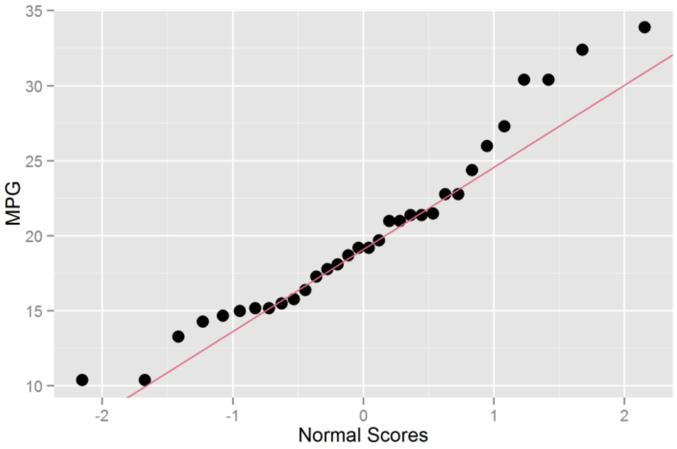


Figure 5-5 Normal probability plot of MPG

For visualizing two quantitative variables, each quantitative variable can be encoded as the spatial channel along the axis, and scatterplot is most common visualization techniques. Scatterplot shows us the information about the shape, direction, strength, density of the relationship between two variables, as well as existence of outliers and skedasticity. The shape can be characterized by the formal mathematical definition including linear, quadratic, cubic, exponential and sinusoidal. The strength of relationship can be captured by coefficient of correlation. Existence of outliers has to be investigated as it may distort the analysis result. The guidelines, such as regression lines and smoother lines, can be added onto scatterplot in order to facilitate the discovery of shape, direction and strength. Figure 5-6 shows the scatterplot of the MPG and Horsepower combined with a linear regression. It shows that a negative relationship exists between MPG and Horsepower, and the r-square that measures the strength of the relationship is 0.602, which means 60.2% of variation of the MPG is predicted by the horsepower, assuming linear relationship exist between the MPG and Horsepower.

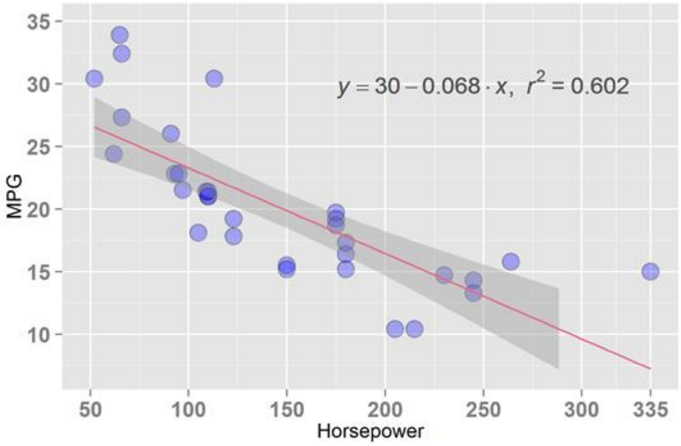


Figure 5-6 Scatterplot of MPG and Horsepower with linear regression line

Another useful technique is called lowess (LOcally WEighted Scatter-plot Smoother). The major advantages of the lowess over Linear Least Squares Regression are that the lowess relaxes the linearity assumption of the linear regression, and the Lowess is a non-parametic regression method that does not assume the data fits some types of distribution (i.e. normal distribution). Loosely speaking, the lowess uses weighted regression method, both linear regression and non-linear regression, to fit the localized subset of data for each part of the scatterplot. Figure 5-7 shows an example, we can see that the lowess suggests that a quadratic regression might be a better fitting compared to the linear regression.

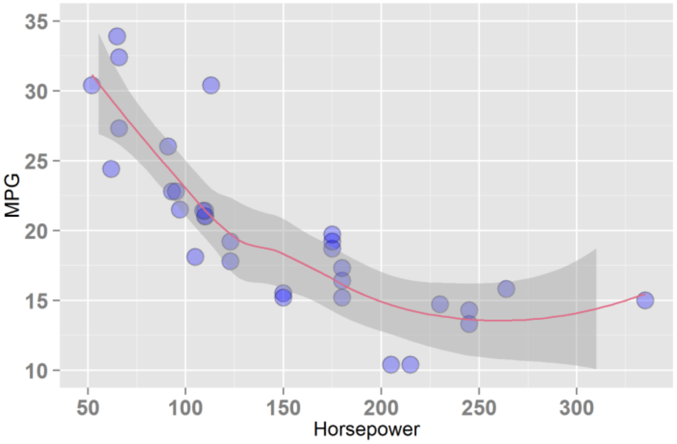


Figure 5-7 Scatterplot of MPG and Horsepower with linear regression line with lowess

However, the disadvantage of Lowess is that, unlike the linear regression, the relationship between the two variables is hard to be defined by mathematic equations. For measuring the relationship of the MPG and Horsepower, one solution is using the quadratic regression suggested by Lowess in figure 5-7. In Figure 5-8, quadratic relationship between MPG and Horsepower is stronger than linear relationship as 75.6% of the variation in Horsepower is explained by the MPG, assuming the quadratic relationship exists.

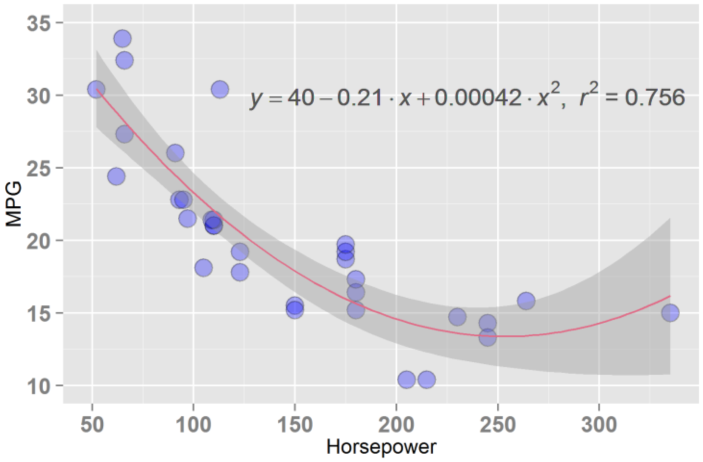


Figure 5-8 Scatterplot of MPG and Horsepower with linear regression line with quadratic regression line

The scatterplot matrix is a matrix of paired scatterplot with a row and column for each variable being plotted. Figure 5-9 is an example, which shows the relationship of paired quantitative variables in the dataset.

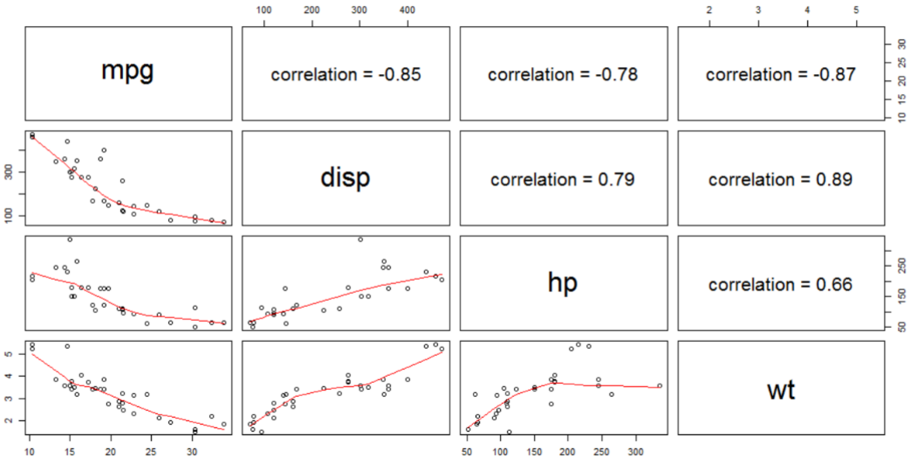
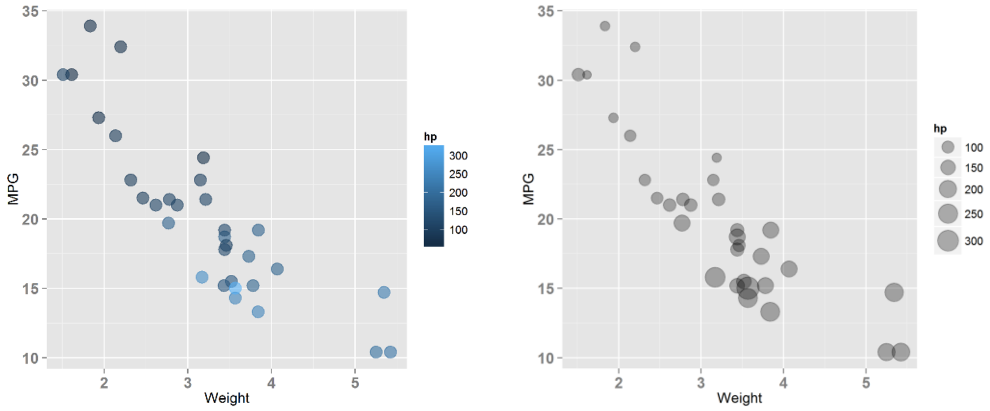


Figure 5-9 Scatterplot Matrix of mpg, disp (displacement), hp (horsepower) and wt (weight)

One principle of visualizing more than two continuous variables is that analyst first make a scatterplot of two of the most important continuous variables, which attract most interests. The dependent variable is put on the Y-axis and the independent variable is put on the X-axis. Additional continuous variables are encoded as color or area. Figure 5-10, for example, in addition to MPG and Weight, we add Horsepower encoded as color (a) and area (b) to scatterplot.



(a)

(b)

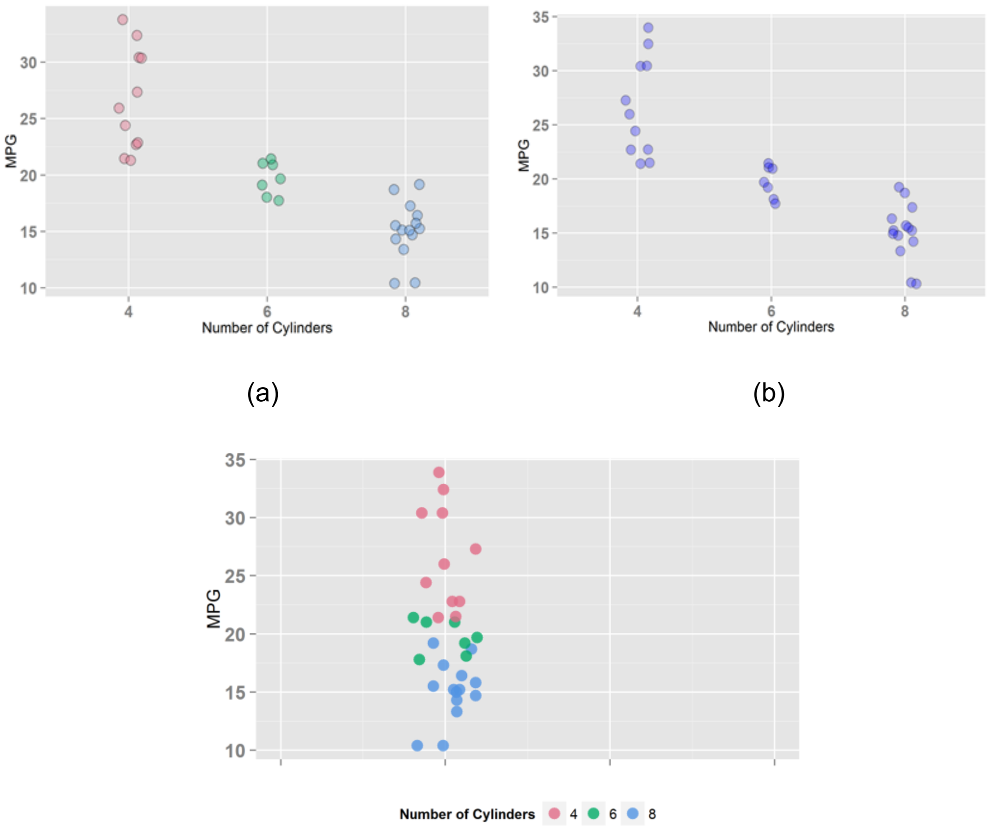
Figure 5-10 Scatterplot of MPG and Weight with Horsepower encoded as color (a) and area (b)

Although we can visualize four continuous variables by using both color and area channels, the plot become messy and distorted quickly as the number of observation increase.

Chapter 6 Visualizing Categorical Variable and Continuous Variable

Using visual channels in an effect way, as we mentioned in Chapter 3, is very important for complicated visualization tasks, such as visualizing high-dimensional categorical variable and quantitative variable. We will follow the custom that the visualization techniques is introduced according to the number of quantitative and categorical variables given.

The visualization of one continuous variable and one categorical variable is subject to the visualization of individual continuous variable, with categorical variable encoded as spatial position and color hue. Figure 6-1 shows three graphs that the continuous variable MPG is shown as dotplot along the vertical axis. However, the categorical variable, number of cylinders, is encoded with spatial position and color hue (a), spatial position (b), and color hue (c). As illustrated in Munzner's effectiveness encoding rule, which is shown as Figure 3-3, removing spatial channel of categorical variable in plot (c) shifts the attention of comparing different levels of categorical variable to investigating the overall distribution of continuous variable.



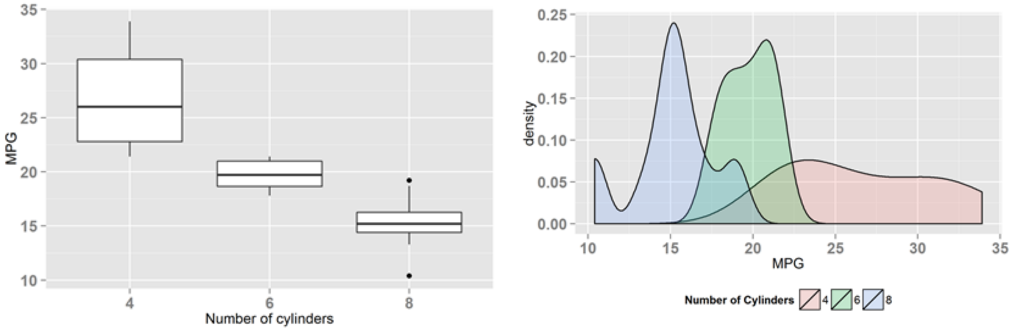
(a)

(b)

(c)

Figure 6-1 Dotplot of MPG with Number of Cylinders encoded as (a) color and position (b) position and (c) color

Figure 6-2 is another two examples that tell the same story about the distribution of the MPG in terms of the number of cylinders. Figure 6-2 (a) is a side-by-side boxplot with a spatial position channel added, while Figure 6-2 (b) is a density curve plot that adds another color hue channel to emphasize the comparison among level of categorical variable, as the spatial position is overlapped along the horizontal axis.



(a) (b)

Figure 6-2 (a) Boxplot of MPG with Number of Cylinders encoded as position and (b) Density plot with Number of Cylinders encoded as position and color

For visualizing one continuous variable and two categorical variables, the side-by-side boxplot can be used. According to Munzner [1], the spatial position is more effective for inducing the viewer to tell the difference than other encodings such color and shape. Figure 6-3 and Figure 6-4, for example, tell the same story with different parts of focuses. In Figure 6-3, the transmission type (automatic or manual) is encoded as the spatial position and the number of cylinders is encoded as color hue. Thus the transmission type has higher priority than the number of cylinders. It induces the viewer to compare the MPG distribution for different transmission type (automatic or manual) with different number of cylinders. Figure 6-4, in contrast, takes the opposite way as it intends to compare the MPG for different number of cylinders with the different transmission type.

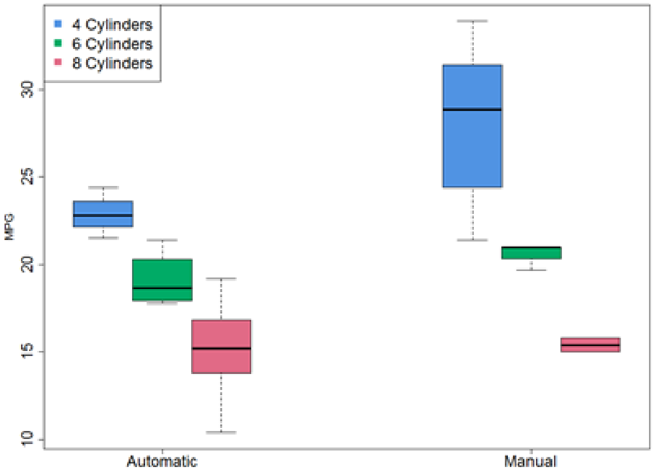


Figure 6-3 Boxplot of MPG with transmission type encoded as position and number of cylinders encoded as color

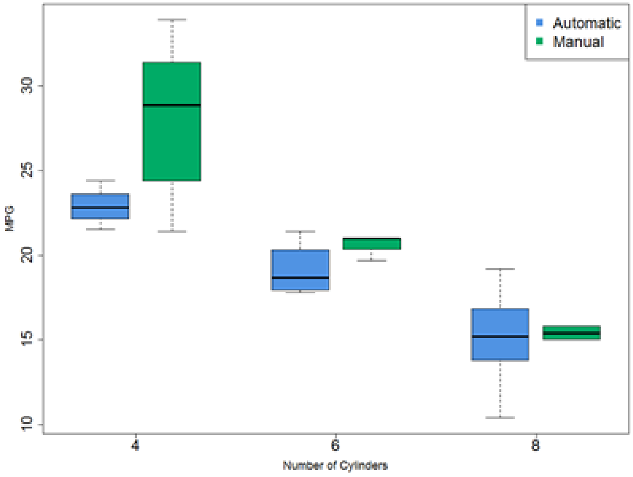


Figure 6-4 Boxplot of MPG with number of cylinders encoded as position and transmission type encoded as color

Faceting is another way to encode the categorical variable. It is a kind of conditional plots that partition the entire dataset based on levels of specified categorical variable, and useful to investigate whether the patterns are same or different across different conditions. Figure 6-5 shows an example that designates the Weight as the conditional variable, and based on which the dataset is divided. The Weight is categorized into three levels that include the light weight, medium weight and heavy weight. In Figure 6-5, instead of the boxplot, the jittered dotplot is used, since the partitioned subsets are too small to be presented. For example, there are only three observations in the subset of heavy cars with automatic transmission and 8 cylinders.

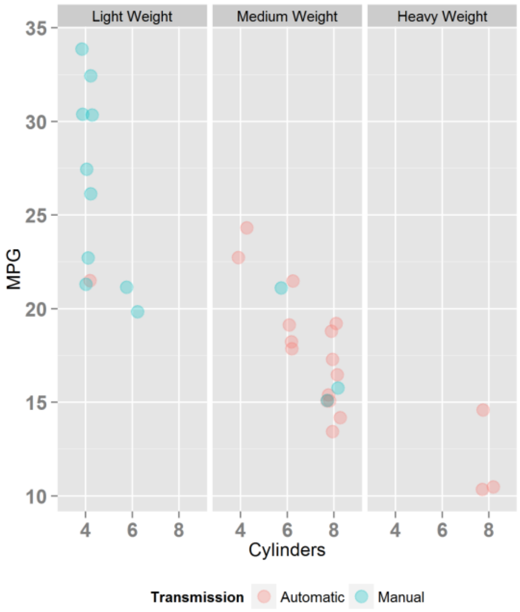


Figure 6-5 Scatterplot of MPG and number of cylinders conditioned on Weight (facet)

Whenever two continuous variables are part of the visualization tasks, the scatterplot is always the first choice as it aims to find the trend, shape and outliers. The categorical variables can be encoded by using identity channels such as color and area. Here in Figure 6-6, the categorical variable is encoded by adding color to the scatterplot.

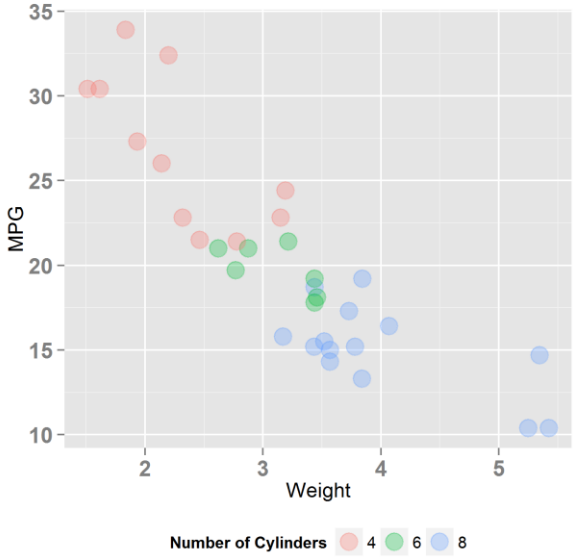


Figure 6-6 Scatterplot of MPG and Weight with Number of Cylinders encoded as Color

Finally, the visualization of three continuous variable and one categorical variable is straightforward. Two of continuous that attract the most interests are first plotted as the scatterplot. The categorical variable is encoded as the color channel and the third continuous variable is encoded as the size channel. One example is shown in Figure 6-7.

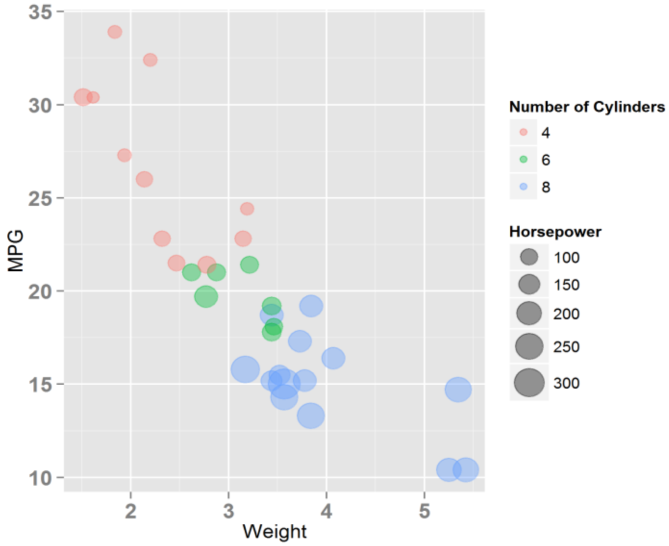
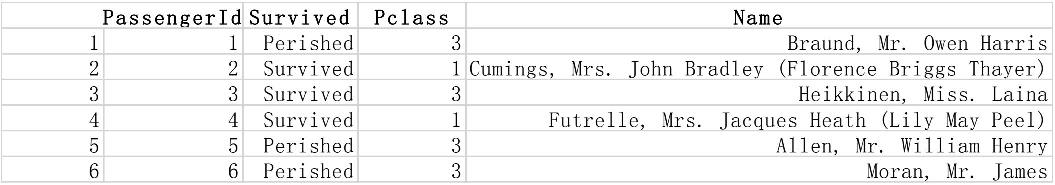


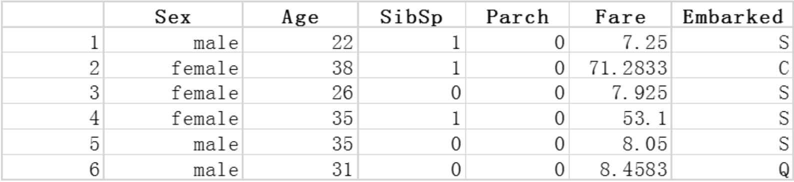
Figure 6-7 Scatterplot of MPG and Weight with Number of Cylinders encoded as color and Horsepower encoded as area

Chapter 7 Case Study

The following case study illustrates how the visual exploratory analysis facilitates the feature engineering in the data mining process. The case study is a classification problem with the goal of predicting the chance of survival of each passenger on the Titanic. The Titanic data used here is different from the one used for visualizing the categorical data in chapter four, which consists of solely categorical variables. The first six rows are shown as below:



	PassengerId	Survived	Pclass	Name
1	1	Perished	3	Braund, Mr. Owen Harris
2	2	Survived	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)
3	3	Survived	3	Heikkinen, Miss. Laina
4	4	Survived	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)
5	5	Perished	3	Allen, Mr. William Henry
6	6	Perished	3	Moran, Mr. James



	Sex	Age	SibSp	Parch	Fare	Embarked
1	male	22	1	0	7.25	S
2	female	38	1	0	71.2833	C
3	female	26	0	0	7.925	S
4	female	35	1	0	53.1	S
5	male	35	0	0	8.05	S
6	male	31	0	0	8.4583	Q

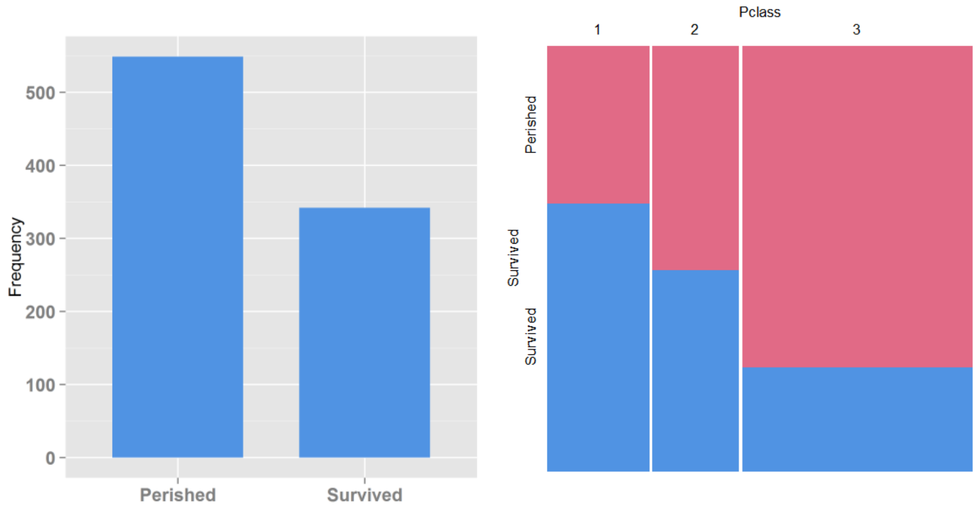
Table 7-1 First six rows of Titanic Data

The following table is a description of the variables that have vague variable names.

Variables Name	Description
Pclass	Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
sibsp	Number of Siblings/Spouses Aboard
parch	Number of Parents/Children Aboard
embarked	Port of Embarkation

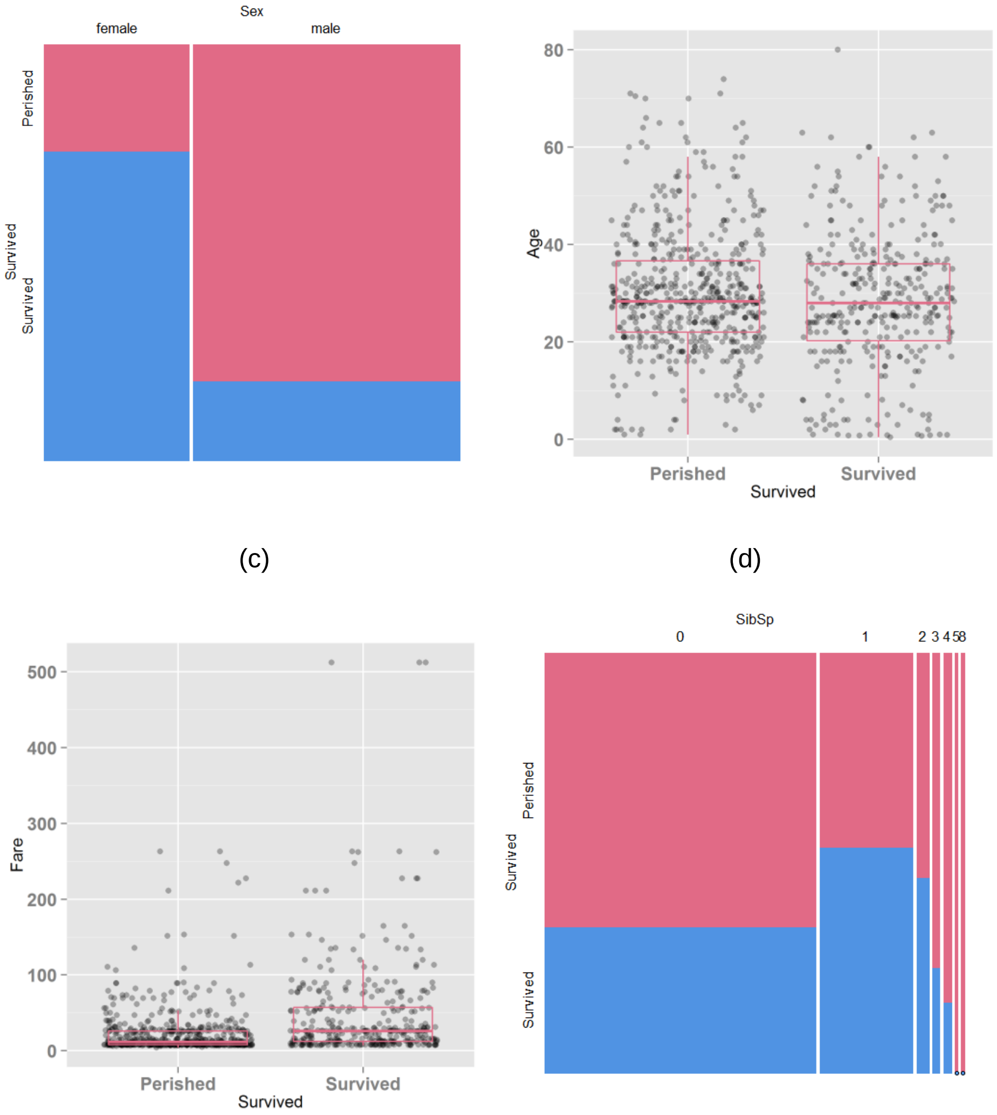
Table 7-2 Description of Titanic variables

Figure 7-1 (a) – (h) list outcomes of the visual exploratory analysis. Figure 7-1 (a) shows the bar chart of the frequency of survival. For a classification problem, it is necessary to investigate the class distribution as a highly skewed class distribution may distort the interpretation of the predictive models performance of. From Figure 7-1 (a), it is concluded that more passenger perished than survived on the Titanic. In fact, on average, 62% of passenger perished, meaning that if we classify all passenger as survivals we will get an accuracy rate of around 62%. This is called the random guessing rate and predictive models should not be considered unless its performance outperforms the random guessing. Figure 7-1 (b)-(h) inspect the distribution between the Survival and Passenger Class, Sex, Age, Fare, Number of Siblings and Spouses aboard, Number of Parents and Children aboard, and Port of Embarkation, respectively. One conclusion might be reached is that the gender is the most prominent variable to predict the odds of survival. The reason is that the proportion of female survivals far exceeds the proportion of male survivals, and the proportion of male and female is close to each other and divides the data evenly. The Passenger Class is probably the second most important variables as passengers in the 3rd class has much lower odds of survival than passengers in the 1st and 2nd. However, pattern of survival in other variables is either vague or erratic. Figure 7-1 (d), for example, does not indicate a relationship between the odds of survival and the Age as the two boxplots overlap with each other. Figure 7-1 (e) seems to tell that a higher fare is an indication of a higher survival rates since the mean fare of survived passengers is almost same as the first quartile of fare of perished passengers. However the result might be misinterpreted as the mean value is exaggerated by the three outliers with an extremely high value over 500.



(a)

(b)

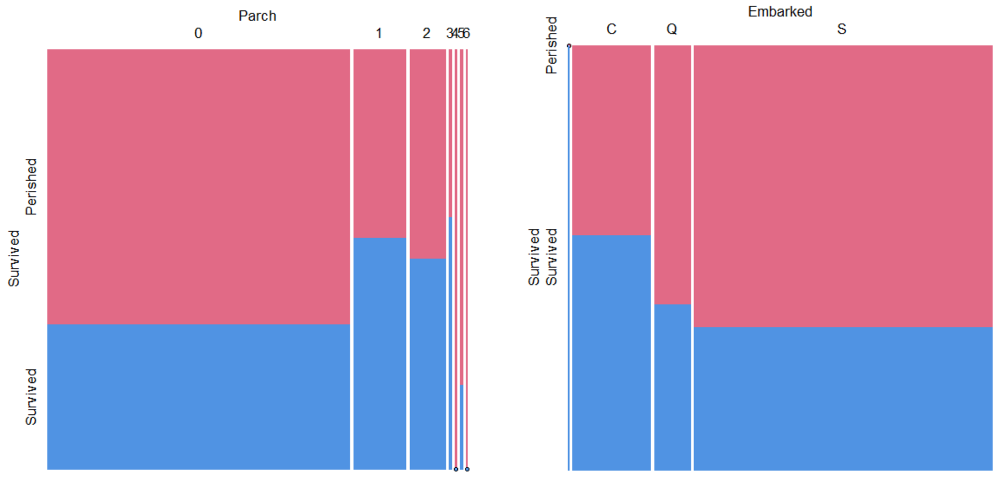

Sex
female
male
Perished
Survived
Survived
80
60
40
20
0
Age
Perished
Survived
Survived
500
400
300
200
100
0
Fare
Perished
Survived
Survived
SibSp
0
1
2 3 4 5 8
Perished
Survived
Survived

(c)

(d)

(e)

(f)



(g) (h)

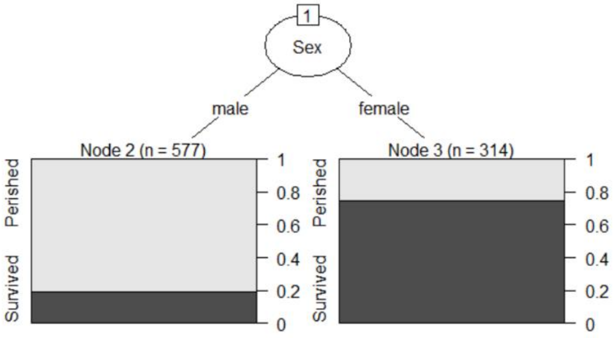
Figure 7-1 Visual exploratory analysis of Titanic data

One hypothesis generated from the visual exploratory analysis that Sex is the most important feature to predict the chance of survival can be tested by using the decision tree algorithm. Four experiments are implemented by using four different inputs. The 10 folds cross-validation is used throughout the experiment. The result is shown as follows:

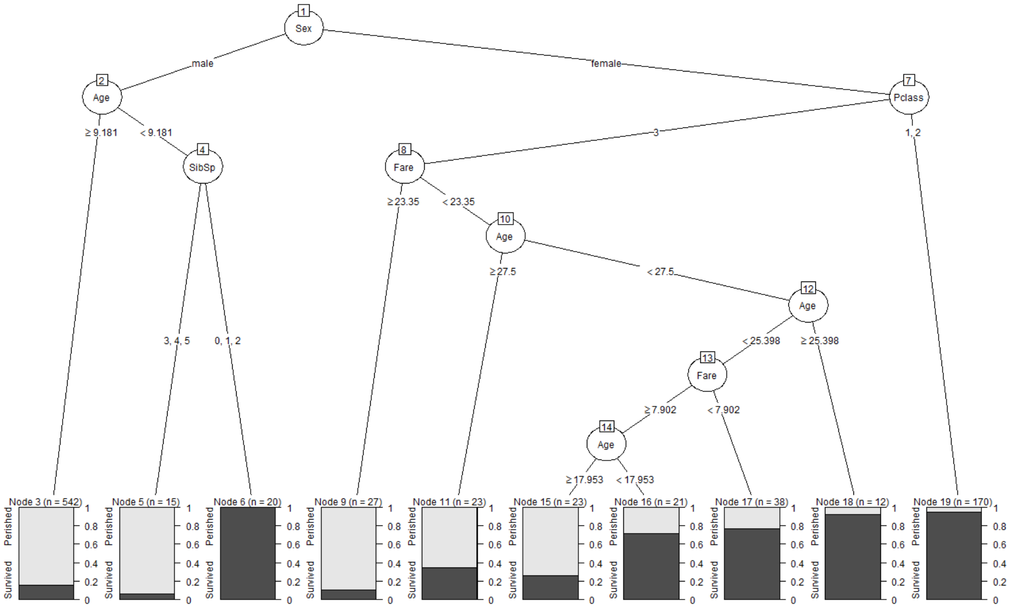
Exp.	Feature	Accuracy Rate
A	Sex	78.68%
B	Pclass	67.90%
C	Age	63.20
D	Pclass + Sex + Age + SibSp + Parch + Fare + Embarked	80.58%

Table 7-3 Result of Classification Tree Algorithm with different input

In experiment A, the Sex is selected as the only input for the decision tree algorithm, and the accuracy rate is 78.68%, which is only 1.9% lower than the accuracy rate produced from the experiment D where all the features are used. This means that the feature Sex contributes the most prediction accuracies among all other features. In addition, the tree generated from the experiment A outperforms the tree generated from the experiment D in terms of simplicity, as the former is as simple as predicting all males perished and all females survived. In order to avoid the overfiting and over complicated representation, the decision tree that uses all features is pruned and it has only 9 splits. The representation of decision tree is shown in Figure 7-2. In fact, the random forest, a sophisticated ensemble method, with a tuning parameter of 2000 trees and all seven features included, produces an accuracy rate of 81.47%, which is only 0.89% higher than the individual tree that employs all seven features and 2.79% higher than individual tree with only the Sex feature.



(a)



(b)

Figure 7-2 Visual representation of Classification tree algorithms with (a) single variable and (b) all 7 variables

The goal of feature engineering is not only to select the most prominent features from existing variables, but it is also to create new features by using the domain knowledge and EDA. For example, each name record has a consistent format of surname, title and first name. We can extract the title and generate a bar chart as shown in Figure 7-3. The data contains 17 different titles and most of which are rarely happened. Using features such as this one is subject to overfittings. One solution is to merge the levels within categorical variables by using the domain knowledge. First, the title indicates the gender of passengers as it contains titles such as Mr., Mrs., Miss., and it can be confirmed by Figure 7-4, which is a plot of gender across titles. Besides, titles suggest age of passengers. For example, English honorific dictates that, by the late 19th century, men are addressed as Mister and boys as Master. Miss can be used to differentiate younger and older females. The boxplots of Age in terms of Titles in Figure 7-5 demonstrates the relationship between the age and titles. In order to merge the titles into groups that evenly divide the data, a dummy variable ‘Nobel’ is created, and all ‘rarely-happened’ titles with an indication of nobility such as Capt, Col, and Sir are assigned to this dummy variable. Other assignments can be found in Figure 7-6.

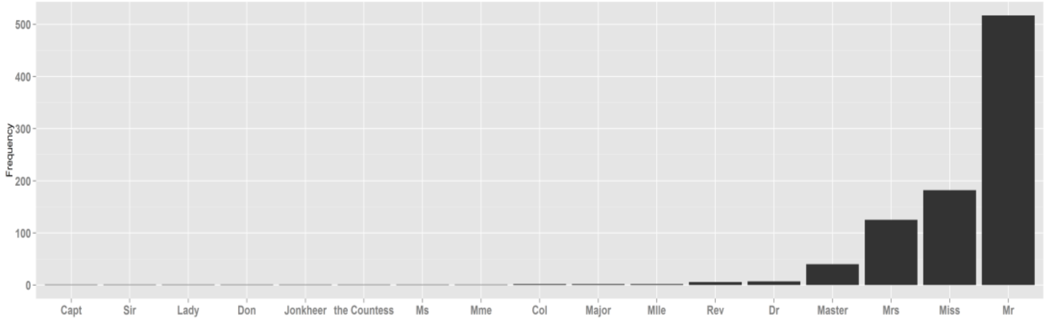


Figure 7-3 Bar chart of titles

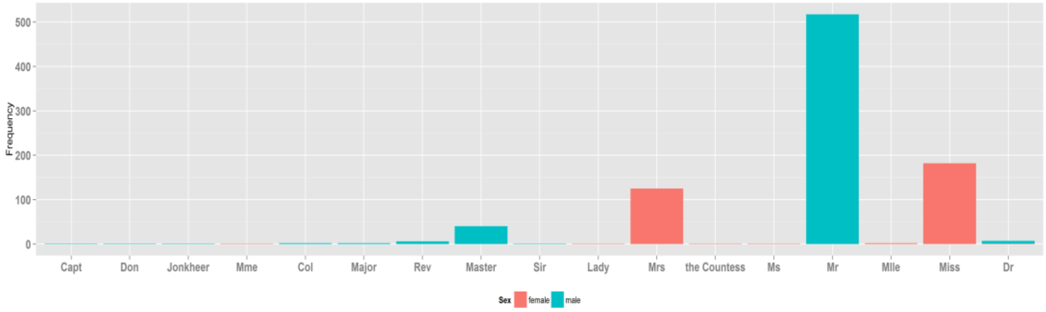


Figure 7-4 Bar chart of Titles with gender encoded as color

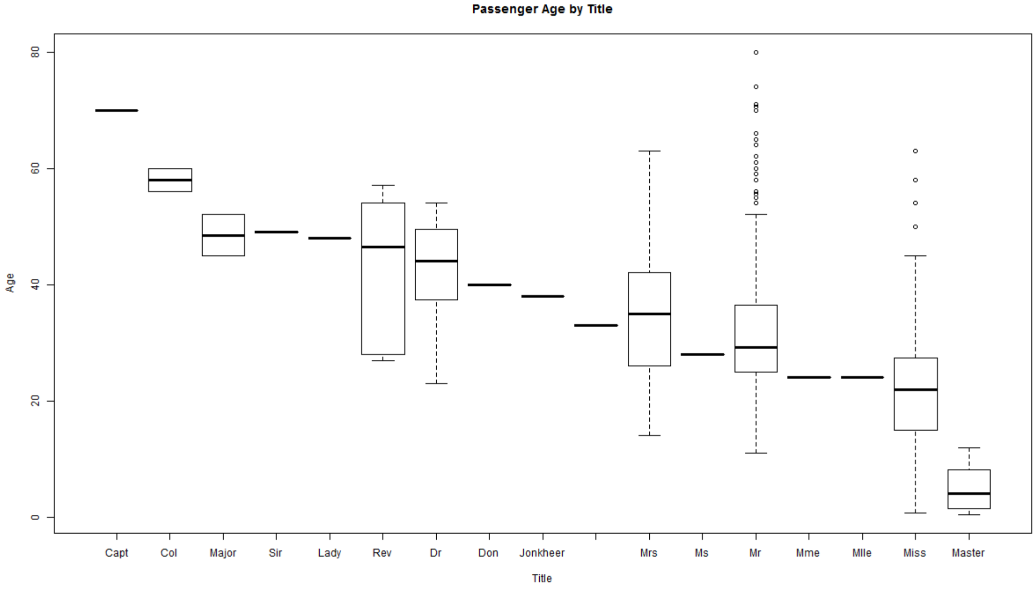
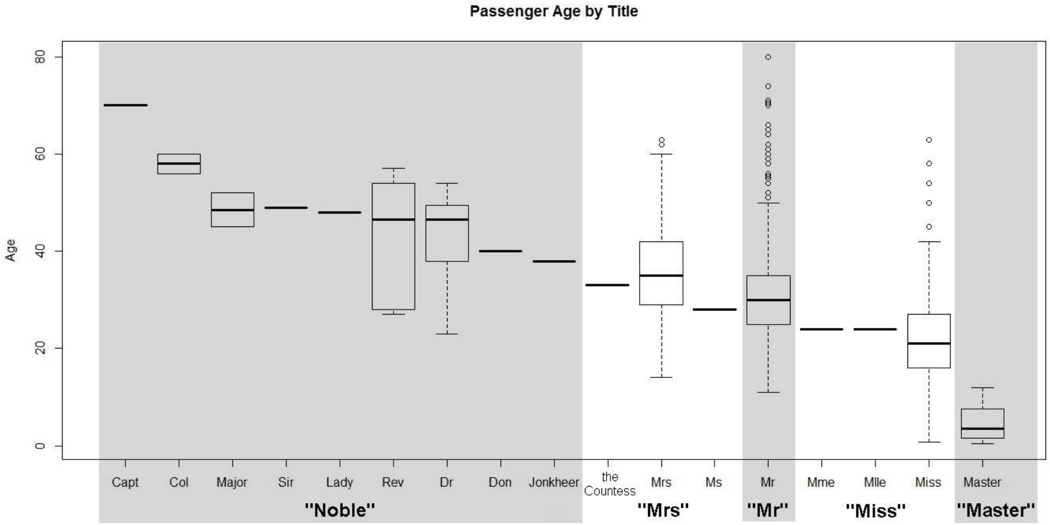


Figure 7-5 Boxplot of Age with Titles encoded as position



(b)

Figure 7-6 Boxplot of Age with Titles and assignment encoded as position

The title variable is an indication of the age, nobility and gender. In order to investigate the importance of the title variable, the Title is used instead of the Sex in both decision tree and random forest algorithm. The following table shows the result. The column of improvement, which is the rightmost column of table, shows the increase of the accuracy rate of using Title instead of Sex in the specified model.

Algorithm	Feature	Accuracy Rate	Improvement
Decision Tree	Title	79.14%	0.46%
Decision Tree	All	82.04%	1.46%
Random Forest	All	83.60%	2.13%

Table 7-4 Evaluation of Title feature

This case study demonstrate how visual exploratory analysis facilitates the feature engineering, which in turn is critical in improving the prediction accuracy in data mining problems.

Chapter 8 An Overview of Exploratory Data Visualization Technologies

Exploratory visual analysis is a highly iterative process in which analysts propose and test hypotheses constantly by using visualizations techniques. It is desirable for the visualization technologies to generate graphical representations rapidly according to the analyst's needs without arduous manual specifications of the visual encoding. Although a number of visualization technologies such as Tableau, ggplot2, d3 are available, their philosophies are different and thus have different concentrations. In this chapter, cutting edge visualization technologies are categorized into grammar-based and web (JavaScript)-based, and their representatives that include ggplot2 [2], Tableau [3], Shiny [29], and d3 [4] are reviewed, with a focus on the compatibility for the exploratory visual analysis.

8.1 Grammar-Based Visualization Tools

Perhaps the most important modern work in graphical grammars is 'The Grammar of Graphics' by Wilkinson [30], since it has largely affected the grammar of several of the most popular visualization tools such as ggplot2's layered grammar [2] [31] and Tableau's VizQL [3]. Wilkinson's grammar provides concise and comprehensive rules of describing objects of statistical graphs that include the data, transformation, scale, coordinates, elements, guides and displays, and how to render these objects by using a plotting system. ggplot2 is a plotting system for R [32] whose layered grammar is heavily affected by Wilkinson's grammar. As Wickham [31] , the inventor of ggplot2, put that the layered grammar used by ggplot2 is a variation of Wilkinson's grammar, differing in its arrangement of the components, the development of defaults, and that it is embedded inside another programming language. Layers are fundamental components of the layered grammar, and a plot may have multiple layers. Layers include four parts:

- Data and aesthetic mapping
- A statistical transformation (stat)
- A geometric object (geom)
- A position adjustment

Examples of aesthetics are the coordinate, shape, size and color. Aesthetic mapping controls which variable maps to which aesthetics. The statistical transformation transforms the input data into new variables that are needed for the creating target graphical representation. For example, in order to make a boxplot, first and third quartiles, maximum and minimum values, and the median are generated from the input data by statistical transformations. The geometric object represents the type of plot such as scatterplots, line plots etc. Position adjustment is rarely used as it tweaks the position of geometric elements on the plot when the geometric elements tend overlap with each other. In addition to the layer component, there are three other components that include scales, the coordinate system and the faceting. The scale controls the properties of aesthetics that each variable maps to. The coordinate system maps the position of each objects onto the plane of the plot, and the Cartesian coordinate system is the most common coordinate system. Faceting is only useful for specific situations since it splits the graph based on specified conditions. We illustrate layered grammar by demonstrating how to make the plot shown in Figure 8-1.

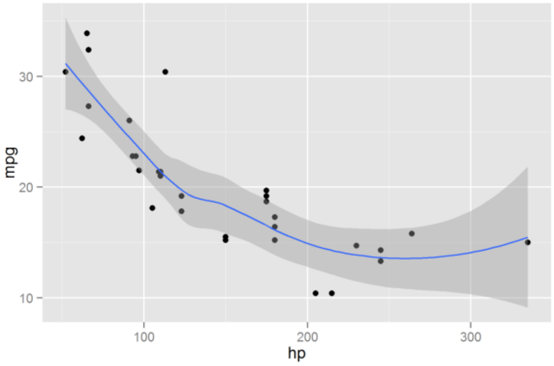
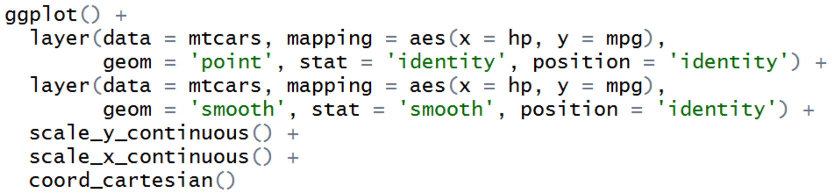


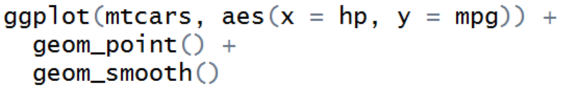
Figure 8-1 Simple plot generated by layered grammar

The basic components of Figure 8-1 include two layers – the scatter point and the loess curve, and its specification of layered grammar is:



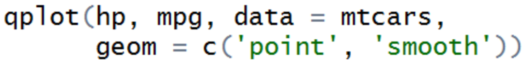
```
ggplot() +
  layer(data = mtcars, mapping = aes(x = hp, y = mpg),
        geom = 'point', stat = 'identity', position = 'identity') +
  layer(data = mtcars, mapping = aes(x = hp, y = mpg),
        geom = 'smooth', stat = 'smooth', position = 'identity') +
  scale_y_continuous() +
  scale_x_continuous() +
  coord_cartesian()
```

As we mentioned at the beginning, the layered grammar has its unique default settings that greatly simplify the grammar specification in a number of ways. For example, the default settings of objects are intertwined with each other, and each geometric object has a default stat. The intentional specification of the stat is only necessary when the default stat needs to be suppressed. For example, the statistic values required for making a boxplot, such as quartile and median, are already given as input data, then we override the default stat by setting it to 'identity'. The shorthand version making Figure 8-1 is:



```
ggplot(mtcars, aes(x = hp, y = mpg)) +
  geom_point() +
  geom_smooth()
```

The layered grammar has a relatively steep learning curve as it enables a full control of the visual specification. However, once analysts master the layered grammar, its development of hierarchical default settings enables remarkable flexibilities that greatly speed the visual specification process. For example the specification of Figure 8-1 can be further shortened as:



```
qplot(hp, mpg, data = mtcars,
      geom = c('point', 'smooth'))
```

The simplicity of visual specification is the core value of exploratory data analysis as analysts are able to rapidly change what data they are viewing and how they are viewing that data. In addition, R possesses multiple libraries that are designed for data manipulation such as dplyr (), reshape2 (), and data.table() greatly facilitates the EDA.

Thus, we consider the R's plot system and database solutions make it a suitable choice for EDA.

Although Tableau does not require the manual specification of graph grammar in its implementation, we classify it as grammar-based because its philosophy is an application of Wilkinson's grammar for multidimensional tables with the interactive interface. As Stolte and Hanrahan [3], the inventor of Tableau as well as the co-founder of Tableau cooperation, stated that "We have extended Wilkinson's ideas to develop a specification that can be directly mapped to an interactive interface and that is tightly integrated with the relational data model". They even use an entire section to discuss the distinctions between their systems and Wilkinson's system.

From a broader view, Polaris systems [3], as the blueprint of Tableau, enables to 'visualize' the Pivot Table by using the grammar of graph. The Pivot Table provides the interactive interface for aggregating and summarizing multi-dimensional database, but the result is limited to text-based display. Polaris makes it possible to visualize the text-based result through automatic or semi-automatic visual specification and mapping. Visual query language (VizQL) forms the core of the Polaris as well as its commercialized version—Tablueau. VizQL is a declarative language for describing tables, charts, graphs, maps, time series and tables of visualizations. Since most of the queries performed in Tableau are drags and drops of target variables, VizQL is generated automatically as analysts drag and drop variables on the shelves as shown in Figure 8-2. In Figure 8-2, the Market (categorical) and the Quantity (continuous) are put on the columns shelf, and Category (categorical) and Segment (categorical) are put on the rows shelf. Market (categorical) is also put on the color marks shelf.

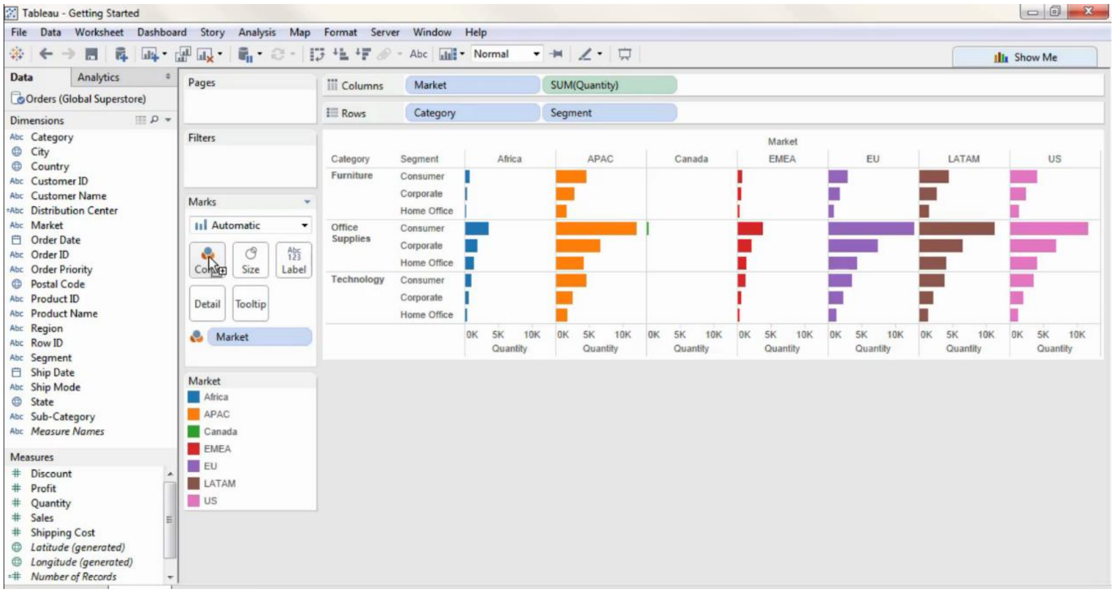


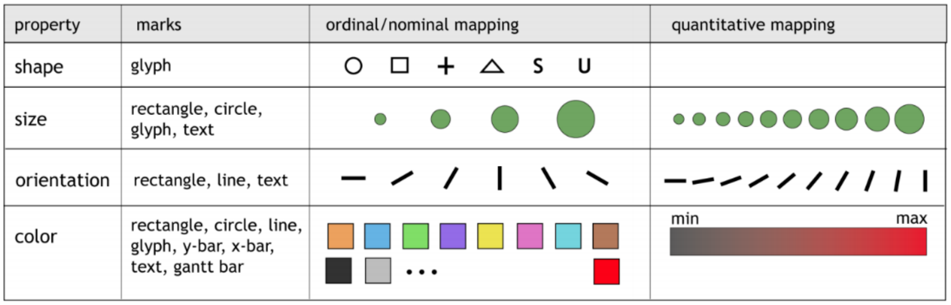
Figure 8-2 Tableau Interface: an illustration of VizQL

Implementation of VizQL can be divided into two tasks –generation of graphics and generation of database queries. Generation of graphics happens before generation of database queries and can be further divided into the three following components.

- Table Algebra: when analyst put variables on the columns and rows shelf, the table algebra is implicitly generated. The table algebra fully controls forms of the table such as how many cells (panes) in the table. The principle is that categorical variables always partition the table display into rows and columns, whereas continuous variables are spatially encoded as axes within the panes. In Figure 8-3, for example, the three categorical variable on rows and columns shelf divide the table into rows and columns, and the continuous variable Quantity is encoded as the axes within each pane.
- Types of Graphics: Similar to the table algebra, the type of variable in the columns and rows determines the types of graphics generated. Stolte and Hanrahan [3] came

up with three combinations of variable types that include categorical variables, continuous variables, and categorical and continuous variables, and based on what the type of graphs are suggested. For example, when two continuous variables are put on the column shelf and the row shelf respectively, a scatterplot is suggested. This is similar to Chapters four to six where we provided an extensive investigation of the visual design and representation in terms of the types of data given.

- Visual Mappings: Visual mappings involve in the visual specification for selected marks on the Marks shelf. For example in Figure 8-4 the categorical variable Market is put on the Mark shelf. VizQL specifies the visual property according to the Bertin's retinal variable [33], which similar to the expressiveness and effectiveness rule mentioned in chapter 3. The default mapping used for visual specifications by VizQL is shown in Figure 8-5.



property	marks	ordinal/nominal mapping	quantitative mapping
shape	glyph	○ □ + △ S U	
size	rectangle, circle, glyph, text	● ● ● ●	● ● ● ● ● ● ● ● ● ●
orientation	rectangle, line, text	— ⁄ / | \ ⟍	— — ⁄ ⁄ ⁄ / / / / | |
color	rectangle, circle, line, glyph, y-bar, x-bar, text, gantt bar	🟧 🟦 🟩 🟪 🟨 🟪 🟦 🟫 ⬛ ⬜ • • • 🟥	min max

Figure 8-5 The Default Mapping of Marks in VizQL. From [3] Fig.3.2

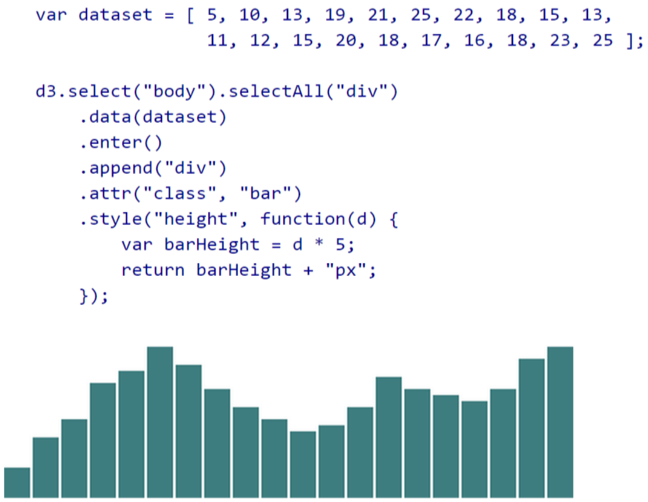
After the visual specification is completed, the visual specification produces database SQL queries for:

- Selecting the target variable for analysis
- Filtering, sorting and grouping the retrieved record into panes in the table display

Tableau is a suitable tool for EDA as the VizQL it employs integrates the grammar of graphics with the manipulation of relational database through an interactive interface. The data manipulation is implicitly implemented and the visual encoding is automatically specified. This highly matches the core of EDA that is a highly rapid discovery-oriented process.

8.2 Web-Based Visualization Tools

One of the important characteristics of web-based visualization technologies are that the visual representation is specified by the HTML, CSS and JavaScript, with input data bound correspondingly. The major advantages of the web-based visualization technology are that it significantly expands the possibilities of visualization representations and provides highly interactive and dynamic graphs. However, because of its intricate designs, it hinders rapid development and thus not fits the ‘quick and dirty’ exploration. Among all the web-based visualization tools, d3.js [4] is probably the most novel and popular technology because of its efficient manipulation of document objects based on data. Data-Driven Documents (D3) is nothing but a JavaScript library, but it solves the problem of how to bind input data to the native representation of web page – document object model (DOM) in an efficient and succinct manner. The core development of D3 is selecting individual or a group of target elements of web page by using .select() or .selectAll(), and binding input data with these elements by using .data() function. The specification of making a simple bar chart, which is shown in Figure 8-6, using D3 is as follows:



```
var dataset = [ 5, 10, 13, 19, 21, 25, 22, 18, 15, 13,
                11, 12, 15, 20, 18, 17, 16, 18, 23, 25 ];

d3.select("body").selectAll("div")
    .data(dataset)
    .enter()
    .append("div")
    .attr("class", "bar")
    .style("height", function(d) {
        var barHeight = d * 5;
        return barHeight + "px";
    });
```

Figure 8-6 Bar Chart created by d3

.enter() and .exit() (not used here) are also two common functions that define how to response when there are more data or fewer data in the array than elements in your selection. Here division (“div”) is used for representing the bar and there are fewer divisions in body section than the value in array. .enter() function tells D3 to .append() new division until the number of division reaches the number of value in array.

D3 is not an ideal tool for EDA since it requires tedious visual specification, which can be seen in the specification for making Figure 8-6. It focuses on the visual representation of data rather than the hypotheses, experimentation, and discovery process.

As the advancement of web (JavaScript)-based technologies, the interactive visualization technologies are much more accessible than ever before. A number of open-source tools are trying to facilitate the development process by developing their own declarative language, that is, analysts specify ‘what to do’ rather than ‘how to do’. Shiny [5] is an open source package for R [32] and is developed by RStudio. Its main advantage is that it provides an incredibly easier way to build interactive web applications in R environment without requiring HTML, CSS, or JavaScript knowledge. All the Shiny applications consist of two components that are:

user-interface (ui) script – controls the layout and appearance of your app

server script – contains the instructions that your computer needs to build your app.

The histogram with interactive bin size in Figure 8-7 is created by using Shiny. The script specification is given as below:

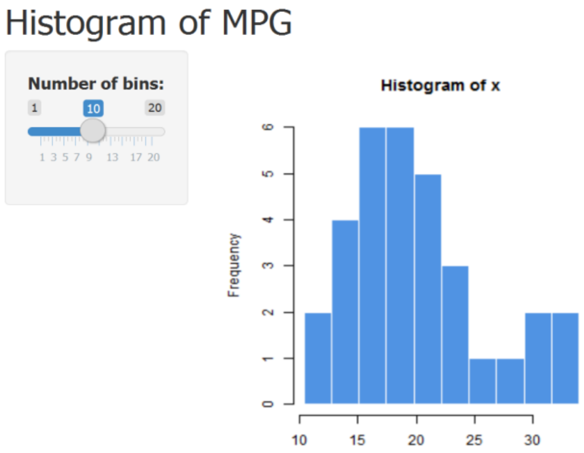
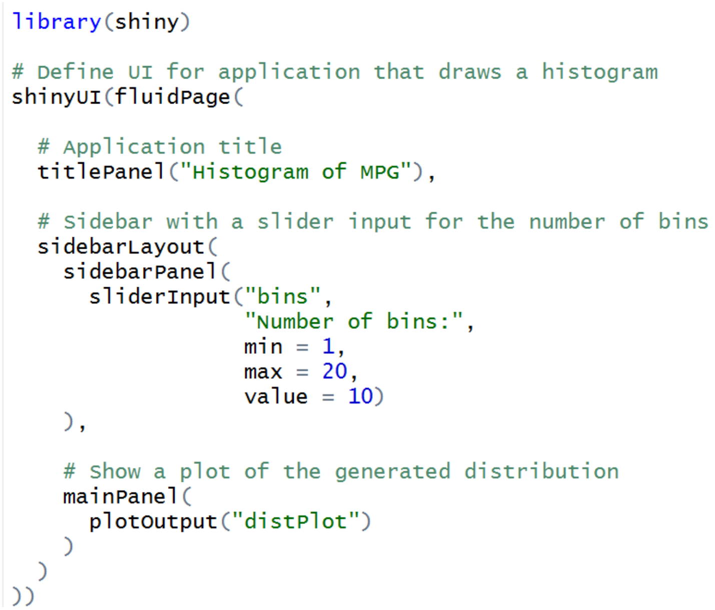


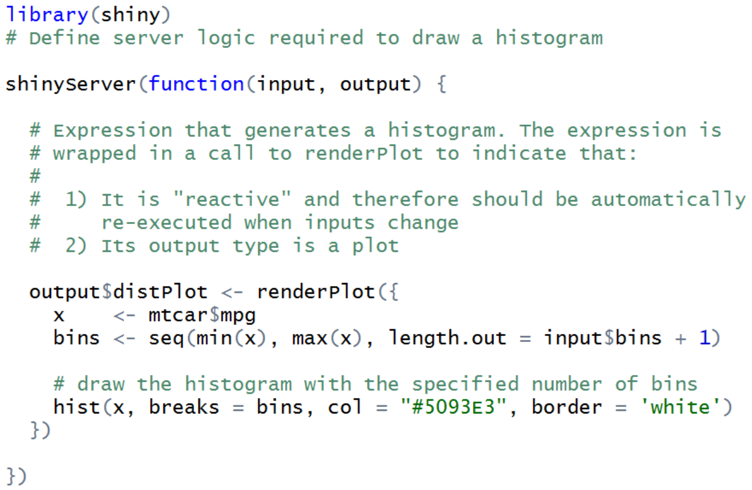
Figure 8-7 Interactive histogram created by using Shiny

ui.r



```
library(shiny)

# Define UI for application that draws a histogram
shinyUI(fluidPage(

  # Application title
  titlePanel("Histogram of MPG"),

  # Sidebar with a slider input for the number of bins
  sidebarLayout(
    sidebarPanel(
      sliderInput("bins",
                  "Number of bins:",
                  min = 1,
                  max = 20,
                  value = 10)
    ),

    # Show a plot of the generated distribution
    mainPanel(
      plotOutput("distPlot")
    )
  )
))
```

server.r



```
library(shiny)
# Define server logic required to draw a histogram

shinyServer(function(input, output) {

  # Expression that generates a histogram. The expression is
  # wrapped in a call to renderPlot to indicate that:
  #
  #  1) It is "reactive" and therefore should be automatically
  #     re-executed when inputs change
  #  2) Its output type is a plot

  output$distPlot <- renderPlot({
    x    <- mtcar$mpg
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

    # draw the histogram with the specified number of bins
    hist(x, breaks = bins, col = "#5093E3", border = 'white')
  })

})
```

The interactive feature of Shiny relies on the reactive programming model whose implementation is similar to a spreadsheet. Several spreadsheet cells can be related with other though specified formulas. When value of one cell changes, the value of another cell that based on the first cell is automatically updated. Here the ‘input’ and ‘output’ value in the shinyServer() function of server.r is a reactive ‘cells’. Every time analysts send a command through interactive feature such as changing number of bins, shinyServer() get its ‘input’ value from sliderInput() function in ui.r script. The value got updated in the shinyServer() function and then send it back to the plotOutput() in the ui.r script in order to render the plot.

Shiny is the favorable tool for EDA since the specification of interactive interface is greatly simplified. Analysts are able to build highly interactive visualizations in the shortest time and put more effort on testing their hypothesis. Besides, Shiny is sitting on the R environment that possesses numerous data wrangling packages that can greatly ease the exploration process.

Chapter 9 Summary

Two main reasons make the exploratory data analysis (EDA) essential in the modern data analysis. First, combined with the computer’s computational power, EDA immerses humans in the data mining process, and the integration of the human greatly amplifies the human’s perceptual system in the pattern discovering. Second, unlike sophisticated computational techniques, such as machine learning and predictive analytics, exploratory data analysis does not assume pre-defined hypothesis and greatly relax the mathematical assumptions. Visual exploratory analysis is the key ingredient of EDA and requires high-level of visual encodings. This paper have reviewed the methods and technologies for visualizing continuous variable and categorical variable, both of which are very common data types in data analysis tasks. We illustrated how each variable is encoded and what descriptive statistics are needed for each visualization techniques. A case study of the Titanic data is provided in order to demonstrate how the visualization techniques introduced in this paper can be implemented to facilitate the data mining process. In addition, several prevailing visualization technologies are reviewed, both grammar-based and web-based, and their adaptabilities for EDA is explored. We found grammar-based visualization tools are highly compatible for EDA, whereas web-based requires tedious visual specification thus does not fit the philosophy of EDA. However, some emerging web-based technologies such as Shiny [5] is trying to fit the gap.

Reference

[1] T. Munzner, Visualization Analysis & Design, Boca Raton, FL: CRC Press, 2014.

[2] H. Wickham, ggplot2: Elegant Graphics for Data Analysis, New York: Springer-Verlag, 2009.

[3] C. Stolte, D. Tang and P. Hanrahan, “Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases,” *IEEE Transactions on Visualization and Computer Graphics,* pp. 52-65, 2002.

[4] M. Bostock, V. Ogievetsky and J. Heer, “D3: Data-Driven Documents,” *IEEE Transactions on Visualization and Computer Graphics,* pp. 2301-2309, Dec. 2011.

[5] RStudio Core Team, *Shiny-Web Application Framework for R,* Boston MA, 2015.

[6] W. J. Tukey, “The future of data analysis,” *Annals of Mathematical Statistics,* pp. 1-67, 1962.

[7] W. J. Tukey, Exploratory Data Analysis, Reading, MA: Addison-Wesley, 1977.

[8] M. Zhu, “The impact of prediction contests,” *Contribution to the 2012 Long Range Plan for Mathematical and Statistical Sciences.,* 2011.

[9] P. Domingos , “A Few Useful Things to Know About Machine Learning,” *Communications of the ACM,* pp. 78-87, 2012.

[10] X. Conort, Interviewee, *Tutorials and Winners' Interviews*. 10 Apr. 2013.

[11] A. F. Zuur, E. N. Leno and C. S. Elphick, “A protocol for data exploration to avoid common statistical problems,” *Methods in Ecology and Evolution,* pp. 3-14, Mar. 2010.

[12] B. Shneiderman, “The eye have it: A task by data type taxonomy for information visualizations,” *Visual Languages,* 1996.

[13] J. H. Larkin and H. A. Simon, “Why a Diagram is (Sometimes) Worth Ten Thousand Words,” *Cognitive Science,* pp. 65-100, Jan. 1987.

[14] W. S. Cleveland, The Element of Graphing Data, Lafayette, IN: Hobart Press, 1994.

[15] F. J. Ansombe, “Graphs in Statistical Analysis,” *American Statistician ,* pp. 17-21, 1973.

[16] J. A. Hartigan and B. Kleiner, “A Mosaic of Television Ratings,” *The American Statistician,* pp. 32-35, 1984.

[17] A. Cohen, “On the Graphical Display of the Significant Components in a Two-Way,” *Communications in Statistics—Theory and Methods,* pp. 1025-1041, 1980.

[18] H. Hofmann, “Generalized Odds Ratios for Visual Modelling,” *Journal of Computa-,* pp. 1-13, 2001.

[19] J. Hummel, “Linked Bar Charts: Analysing Categorical Data Graphically,” *Computational Statistics,* pp. 23-33, 1996.

[20] H. Hofmann and M. Theus, “Interactive Graphics for Visualizing Conditional Distributions,” Unpublished Manuscript, 2005.

[21] M. Friendly, “Mosaic Displays for Loglinear Models,” *Proceedings of the Statistical Graphics Section,* pp. 61-68, Aug 1992.

[22] M. Friendly, “Mosaic Displays for Multi-Way Contingency Tables,” *Journal of the American Statistical Association,* pp. 190-200, Mar. 1994.

[23] M. Friendly , Visualizing Categorical Data, SAS Institute, 2001.

[24] V. Barnett and T. Lewis, Outliers in Statistical Data, New York: Wiley, 1995.

[25] J. M. Chamber, W. S. Cleveland and P. A. Tukey, Graphical Methods for Data Analysis, Belmont, CA: Wadsworth International Group, 1983.

[26] M. P. Wand, “Data-based choice of histogram bin width,” *Statistical Computing and Graphics,* pp. 59-64, 1996.

[27] A. H. Sturges, “The choice of a class interval,” *Journal of the American Statistical Association,* pp. 65-66, 1926.

[28] W. D. Scott, Multivariate Density Estimation, New York: Wiley, 1992.

[29] R. D. C. Team, *R: A language and environment for statistical computing,* Vienna, Austria: R Foundation for Statistical Computing, 2012.

[30] L. Wilkinson, The Grammar of Graphics, New York: Springer, 1999.

[31] H. Wickham, “A Layered Grammar of Graphics,” *Journal of Computational and,* pp. 3-28, 2010.

[32] R Development Core Team, *R: A language and environment for statistical computing,* Vienna, Austria: R Foundation for Statistical Computing, 2012.

[33] J. Bertin, Semiology of Graphics, Madison, Wisconsin: The University of Wisconsin Press, 1983.

[34] F. M. Young and M. Frendly, Visual Statistics: Seeing Data with Dynamic Interactive Graphics, Hoboken, New Jersey: John Wiley & Sons, Inc., 2006.

[35] A. J. Hartigan and B. Kleiner, “Mosaics for Contingency Tables,” *Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface,* pp. 268-273, Aug. 1981.

[36] L. Wilkinson and R. Grossman, “Graph-Theoretic Scagnostics,” *IEEE Symposium on Information Visualization,* pp. 157-164, 2005.

Biographical Information

Yingsen Mao received the Bachelor of Business Administration in Accounting from Hong Kong Baptist University in 2011. Yingsen joined The University of Texas at Arlington in 2012 and in the second year he received the Master of Science in Quantitative Finance. Yingsen continued his study in Information Systems as he aims to combine the information technology with his knowledge from business domain. In December 2015, Yingsen will graduate with Master of Information Systems, with a focus in business analytics. His research interests include database, predictive analysis and data visualization. He is also active in participating in practical prediction projects hosted by Kaggle, which is a platform for data prediction competitions.

Yingsen's previous professional experience include accountant and analyst. Currently he is trying to follow a career in data science and becoming a professional data scientist in the near future.