Ve401 Probabilistic Method in Engineering

Summer 2016 Term Project

Submitted by

Roll No	Names of Students
5133709006	Zhen Ruchen
5133709030	Wu Xinyi
5133709047	Gu Zuguang
5133709101	Ni Pengwei
5133709185	Jiang Lu

Under the guidance of
Dr. Horst Hohberger

UMJI-SJTU Joint Institute, Shanghai Jiao Tong University
Dong Chuan Road 800, Minhang, Shanghai, China

UMJI-SJTU Joint Institute

Shanghai Jiao Tong University

Certificate

Group 22 hereby claims that all of our members have acknowledged the **JI Honor Code** Statements and follows them. If any texts in this report violate any statement of JI-Honor Codes, our groups will take full responsibilities. On the other hand, our work is original, therefore any citations from the report must proceed under the permissions of the group members.

Roll No	Names of Students
5133709006	Zhen Ruchen
5133709030	Wu Xinyi
5133709047	Gu Zuguang
5133709101	Ni Pengwei
5133709185	Jiang Lu

Dr. Horst Hohberger
(Project Guide)

Yuan Peng
(Course Coordinator)

Date: 2016 July 31st

Abstract

This term project comprises of three topics. All of them are general researches that closely relate to the methods of probabilities. We will apply our knowledges leant in the course to solve these problems.

The first topic is *Analysis of Package Contents.* We apply basic test studies learnt in Ve401 to this topic. According to an official file about food weight, we are able to choose certain packaged food and test if the specified weight equal to the real weight.Finally we made some observations.

The second topic *Mass Shootings in the United States* shows some diagrams and tables for the statistic result of the mass shooting database provided by the *Gun Violence Archive* [1]. Intuitively,a mass shooting is an incident involving multiple victims of gun violence[3].However we find several different definitions about a ”mass shooting”. To make the following research more accurate, we take that of the GVA as standard. Following the thought of the article *London murders: a predicable pattern?*, we then extract the number of death in a homicide and the corresponding occurrence date from the database of GVA and categorize these data by date , 7 days in a week, and by number of successive days. Observing that the occurrence of homicide vaguely follows Poison distribution, we finally make a linear regression model indicating the number of death from 2013 Jan. 1st to a certain day afterwards.

The third topic is about *The German Bundesliga.* We perform forward selection , backward elimination, stepwise method, Max R Square Method and Mallow Cp to estimate the data. Finally some astonishing result comes.

Contents

List of Figures

List of Tables

Chapter 1

Analysis of Package Contents

In our daily life, when we buy something containing in a package, we often wonder that whether the net weight of the packaged product is less than the nominal net weight labeled on the package. In order to maintain the order of the market, the Chinese government has made Rules of Metrological Testing for Net Quantity of Products in Prepackages with Fixed Content [6] to protect the interest of consumers. In this rule, the product producers are required to make their products net weight almost a little bit heavier than the label. In the section 4.3.2 of this rule, the sample size of different population is noted. Moreover, equation for relationship between sample mean and nominal net quantity is also given. In addition, tolerance limit is also a factor to be considered. In the end, we will have one product checked to see whether it satisfy the rule or not.

1.1 Metrology and Inspection Sampling Plan

From the Table 1.1, we obtain a requirement for the sample mean.

$$\bar{q} \geq (Q_n - \lambda s)$$

where

$$\bar{q} = \frac{1}{n}\sum_{i=1}^{n} q_i$$

denotes the sample mean, Q_n denotes the labelled net quantity, λ denotes the revision factor.

$$\lambda = t_{0.995} \times \frac{1}{\sqrt{n}}$$

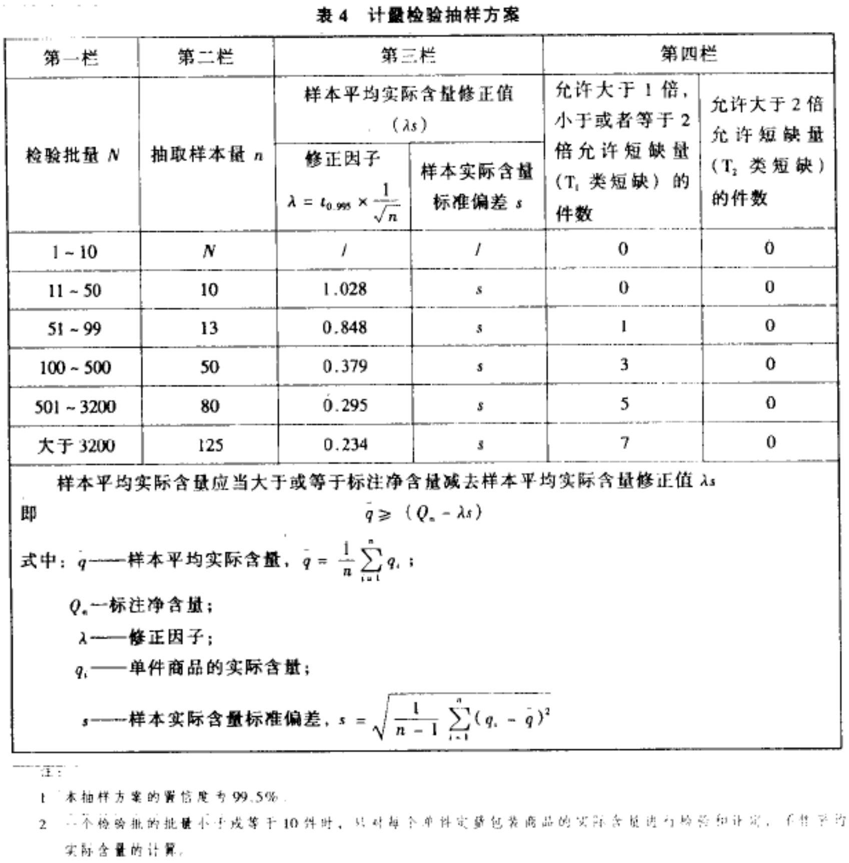


表 4 计量检验抽样方案

第一栏	第二栏	第三栏		第四栏	
检验批量 N	抽取样本量 n	样本平均实际含量修正值（λs）		允许大于 1 倍，小于或者等于 2 倍允许短缺量（T_1 类短缺）的件数	允许大于 2 倍允许短缺量（T_2 类短缺）的件数
		修正因子 $\lambda = t_{0.995} \times \frac{1}{\sqrt{n}}$	样本实际含量标准偏差 s		
1 ~ 10	N	/	/	0	0
11 ~ 50	10	1.028	s	0	0
51 ~ 99	13	0.848	s	1	0
100 ~ 500	50	0.379	s	3	0
501 ~ 3200	80	0.295	s	5	0
大于 3200	125	0.234	s	7	0

样本平均实际含量应当大于或等于标注净含量减去样本平均实际含量修正值 λs

即 $\bar{q} \geqslant (Q_n - \lambda s)$

式中：$\bar{q}$——样本平均实际含量，$\bar{q} = \frac{1}{n}\sum_{i=1}^{n} q_i$；

Q_n—标注净含量；

λ——修正因子；

q_i——单件商品的实际含量；

s——样本实际含量标准偏差，$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(q_i - \bar{q})^2}$

注：

1 本抽样方案的置信度为 99.5%.

2 一个检验批的批量小于或等于 10 件时，只对每个单件定量包装商品的实际含量进行检验和评定，不作平均实际含量的计算。

Figure 1.1: By-variable type sampling plan

q_i denotes for the quantity for a single thing in the sample, s denotes the sample standard deviation.

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(q_i - \bar{q})^2}$$

The equation $\bar{q} \geq (Q_n - \lambda s)$ is equivalent to:

$$\bar{q} \geq (Q_n - t_{0.995} \times \frac{1}{\sqrt{n}} s) \Rightarrow Q_n \leq \bar{q} + t_{0.995} s/\sqrt{n}$$

This implies that Q_n is in a 0.5% one-sided confidence interval: $(-\infty, \bar{q} + t_{0.995} s/\sqrt{n})$ So we will perform a left-tailed hypothesis test to it, with

$$H_0 : \mu \geq Q_n \qquad H_1 : \mu < Q_n$$

where μ is the population mean. We will apply a T-test with:

$$T_{n-1} = \frac{\bar{q} - Q_n}{S/\sqrt{n}}$$

if $T_{n-1} \geq t_{0.5\%}$, we will reject H_0 with 0.5% significance, and conclude that the net weight of the package fits the requirement of [6].

The condition we used to reject H_0 is equal to $Q_n \leq \bar{q} + t_{0.995}s/\sqrt{n}$, is actually the confidence interval we got previously. Therefore, the result using confidence interval and hypothesis test should be exactly the same.

1.2 Additional restrictions to the actual weight

In addition to the restriction on the mean we derived in the previous section, there is another restriction on the numbers of samples that deviates a lot with the net weight labelled on the package, i.e., the existence of an outlier. That means, there should not be a sample with content much less than labelled. Let n_i denotes the number of the samples with insufficient content and let T denote the acceptable shortage in terms of the content. Then we have:

- [6] defined two types of shortage.
- T1-type shortage means the shortage is between two times and one time the permissible shortage. We denote the allowable number in a sample by n_i.
- T2-type shortage means the shortage is more than two times the permissible shortage, which is not allowed in a product, so the allowable number is 0.

In this section we will derive the allowing number of n_i.

With the definition of T1-type shortage, we obtained the allowing region:

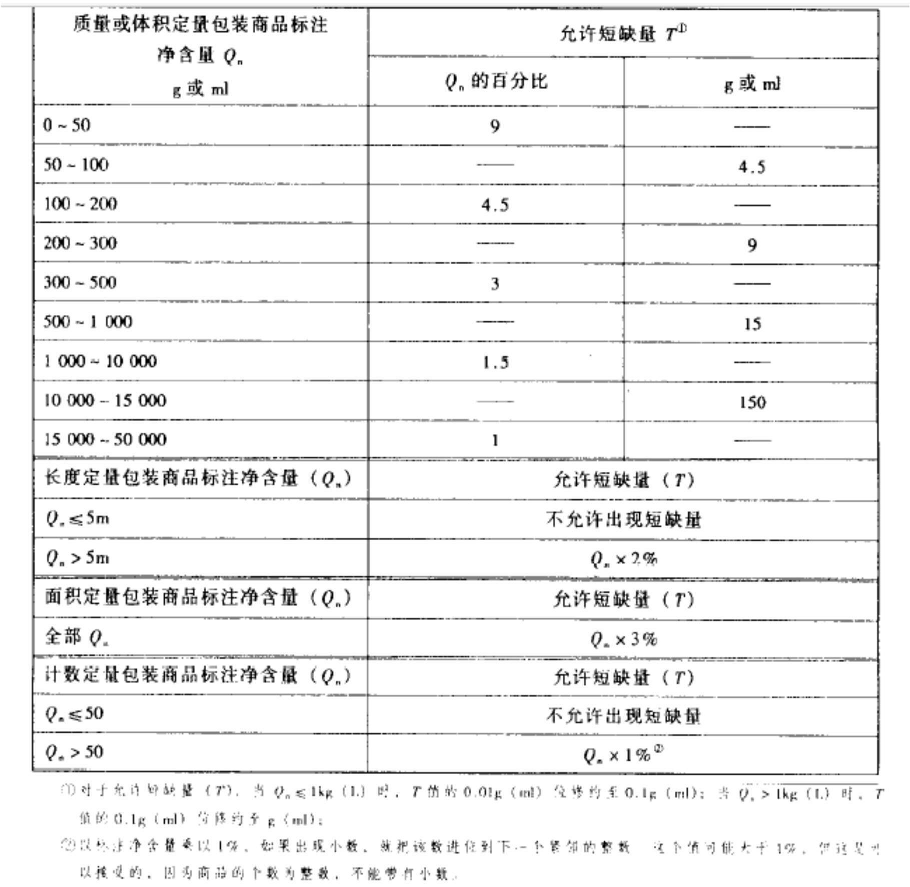
$$P[q \leq Q_n - T] - P[q \leq Q_n - 2T] \leq \frac{n_i}{n}$$

We assume that q follows a normal distribution, and then we have:

$$\Phi\left(\frac{Q_n - T - \bar{q}}{s}\right) - \Phi\left(\frac{Q_n - 2T - \bar{q}}{s}\right) \leq \frac{n_i}{n}$$

We will calculate the critical situation that:

$$\Phi\left(\frac{Q_n - T - \bar{q}}{s}\right) - \Phi\left(\frac{Q_n - 2T - \bar{q}}{s}\right) = \frac{n_i}{n}$$



质量或体积定量包装商品标注 净含量 Q_n g 或 ml	允许短缺量 T①	
	Q_n 的百分比	g 或 ml
0 ~ 50	9	——
50 ~ 100	——	4.5
100 ~ 200	4.5	——
200 ~ 300	——	9
300 ~ 500	3	——
500 ~ 1 000	——	15
1 000 ~ 10 000	1.5	——
10 000 ~ 15 000	——	150
15 000 ~ 50 000	1	——
长度定量包装商品标注净含量（Q_n）	允许短缺量（T）	
$Q_n \leqslant 5$m	不允许出现短缺量	
$Q_n > 5$m	$Q_n \times 2\%$	
面积定量包装商品标注净含量（Q_n）	允许短缺量（T）	
全部 Q_n	$Q_n \times 3\%$	
计数定量包装商品标注净含量（Q_n）	允许短缺量（T）	
$Q_n \leqslant 50$	不允许出现短缺量	
$Q_n > 50$	$Q_n \times 1\%$②	

① 对于允许短缺量（T），当 $Q_n \leqslant 1$kg（L）时，T 值的 0.01g（ml）位修约至 0.1g（ml）；当 $Q_n > 1$kg（L）时，T 值的 0.1g（ml）位修约至 g（ml）；

② 以标注净含量乘以 1%，如果出现小数，就把该数进位到下一个紧邻的整数。这个值可能大于 1%，但这是可以接受的，因为商品的个数为整数，不能带有小数。

Figure 1.2: Allowing Shortage Value
]

We take $Q_n = 50g$ as an example to calculate. From Figure 1.2 , we obtain that $T = 4.5g$. Then plug into $\bar{q} = Q_n - \lambda s$ and check the accumulative distribution table of normal distribution to find $\Phi((Q_n - T - \bar{q_0})/s)$ and $\Phi((Q_n - 2T - \bar{q_0})/s)$. And we will calculate n_i based on:

$$n_i = n \left[\Phi\left(\frac{Q_n - T - \bar{q}}{s} \right) - \Phi\left(\frac{Q_n - 2T - \bar{q}}{s} \right) \right]$$

we could find that n_i is equal to 0.013, 0.489, 1.512, 3.477, 5.269, 7.101 for Q_n equals 5, 10, 13, 50, 80, 125. Thus we round it down to get 0, 0, 1, 3, 5, 7. It is the same with the fifth column of Figure 1.1.

Since that the T2-type shortage is not allowed, so the sixth column of Figure 1.1 are all zeroes.

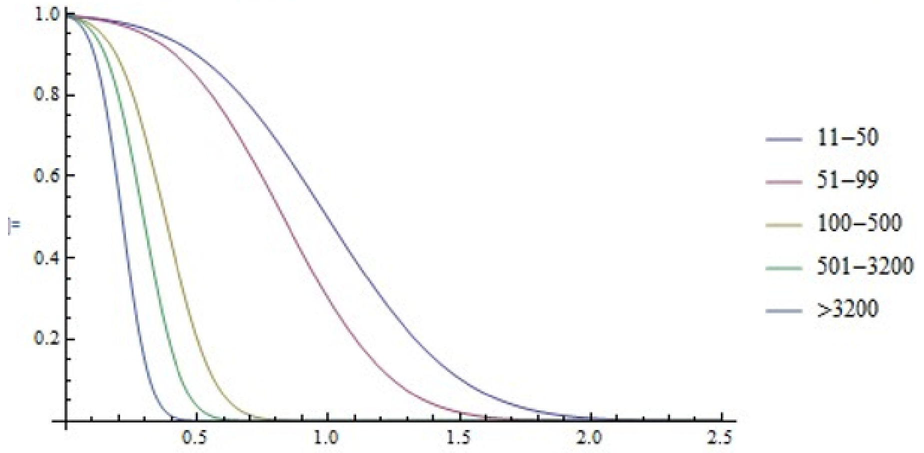


Figure 1.3: OC Curve for the test

1.3 Find the OC Curve

In this section we will find out the probability of type II error and power. We will also derive the probability density function and plot the OC curve of non-central T-distribution. The probability of the Type II error can be found by calculating the probability of a random variable follows a non-central Student T-distribution falls in acceptance region. The non-central T-distribution has the form

$$T_n = \frac{Z+\lambda}{\sqrt{\chi^2/n}}$$

where where n is the degrees of freedom and λ is the non-centrality parameter.

We then normalized it with $d = \frac{Q_n-\mu}{\sigma}$, and obtain

$$\frac{\bar{q}-Q_n}{s/\sqrt{n}} = \frac{\bar{q}-\mu-d\sigma}{s/\sqrt{n}} = \frac{(\bar{q}-\mu)/(\sigma/\sqrt{n})-d\sqrt{n}}{\sqrt{((n-1)s^2/\sigma^2)/(n-1)}} = \frac{Z-d\sqrt{n}}{\sqrt{\chi^2_{n-1}/(n-1)}} \sim T_{n-1,-d\sqrt{n}}$$

Since that the hypothesis test we form is

$$H_0 : \mu > Q_n \qquad H_1 : \mu < Q_n$$

it is a one-sided hypothesis test. Hence, the probability of type II error:

$$\beta = 1 - \int_{\infty}^{t_{0.005}} T_{n-1,-d\sqrt{n}}(x)dx$$

The power of this test is 0.504, 0.701, 0.999, 1, 1, respectively.

1.4 Discussions on Section 5.3.2 of the Regulation

The first part of the section discussed the type I error, which is the maximum possibility of rejecting H_0 when it is actually true. where H_0 : $= Q_n$. According to the rule by the government, the type one error is fixed in a quite low level: $\alpha\mu = 0.5\%$.

The second part of this section talks about the tolerance limit, it shows that we would be 95% confident that 97.5% packaged product of population should be in the tolerance limit. The one-sided tolerance limit is: $(-\infty, \bar{q} + K(n, \alpha_\mu, \delta) \cdot S)$.

1.5 Verification of nominal net quantity for Cui Cui Sha

We assume the N belongs to 51 - 90, therefore, we took 13 samples to do this experiment.

Figure 1.4: Samples of packaged Cui Cui Sha

We want to find out whether the *Cui Cui Sha*® we bought fits the requirement of [6]. We will first test whether the sample has a T1-type shortage, and then test whether it has a T2-type shortage.

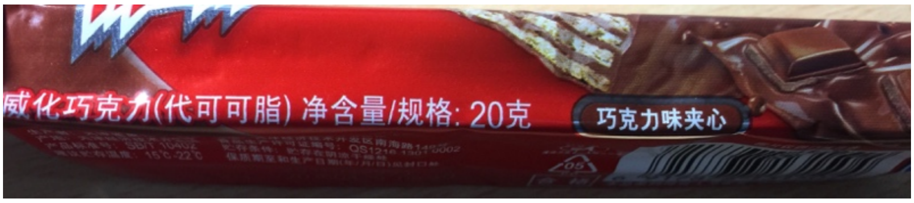


Figure 1.5: The nominal net quantity labelled on the package of *Cui Cui Sha*®

The data we measure are shown below:

Sample Number	1	2	3	4	5	6	7
Real Weight(g)	20.6	20.5	20.8	20.4	20.5	20.4	20.8
Sample Number	8	9	10	11	12	13	
Real Weight(g)	20.6	20.6	20.6	20.5	20.9	20.5	

Table 1.1: Sample weight for *Cui Cui Sha*®

We then calculate the sample mean and sample variance:

$$\bar{q} = \frac{1}{3}\sum_{i=1}^{13} q_i = 20.59$$

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{13}(\bar{q} - q_i)^2} = 0.155$$

We use the following two methods to test whether the sample has a T1-type shortage.

In the first method, we form a one-sided confidence interval

$$\left(-\infty, \bar{q} + \frac{t_{12,0.995}s}{\sqrt{n}}\right] \Rightarrow -\infty, 20.59 + \frac{3.055 \times 0.155}{\sqrt{13}} \Rightarrow (-\infty, 20.72]$$

and we find that the nominal net quantity 20 is in this confidence interval, so it we can say that with 99.5% level of confidence that the sample does not have a T1-type shortage.

In the second method, we will do a hypothesis test to check whether $\bar{q} \geq (Q_n - \lambda s)$

$$H_0 : \mu = 23g \qquad H_1 : \mu > 23g$$

where μ is the unknown population mean. We use T-test:

$$T_{n-1} = \frac{\bar{q} - Q_n}{S/\sqrt{n}}$$

$$T_{n-1} = \frac{20.59 - 20}{0.155/\sqrt{13}} = 13.724 > t_{12,0.995} = 3.055$$

then we will reject H_0 and then prefer $\mu > 23g$ with 0.5% level of significance.

For the sample number between 50 and 99, the permitted shortage $T = 4.5g$. Since that this is no sample in the T2-type shortage region:

$$(Q_n - T, Q_n) \Rightarrow (20 - 4.5, 20) \Rightarrow (15.5, 20)$$

we can conclude that this sample do not have a T2-type shortage. Since that this is no sample in the T2-type shortage region:

$$(Q_n - 2T, Q_n - T) \Rightarrow (20 - 9, 20 - 4.5) \Rightarrow (11, 15.5)$$

we can conclude that this sample do not have a T2-type shortage.

Chapter 2

Mass Shootings in the United States

This is a serious topic. Intuitively "Mass Shooting" is a kind of homicide that cause multiple deaths and injuries. For statistician, these recorded data produce a certain feeling of sacredness and respect. Thus every results, either tables or graphics, that obtained from the GVA(*Gun Violence Archive*) must be granted seriously and carefully.

The analyse of data regarding murders and shootings show us a strong sense of responsibility. Thus in order to make accurate predictions, we categorized the data into several different parts and applied corresponding statistic methods.

Since this project focus on the number of mass shootings, at first our group made the definition of "Mass Shooting" and "Mass Murder" clear. To visualize the mass shooting numbers from 2013 to 2015. We categorized the data into dates and made a bar chart with dates on x-axis. Since the occurrence of the homicide are independent of each other and the probability of occurrence within some large region in a short time is assumed to be stable, we made the assumption that the process of mass shooting follows a Poisson process. With adequate analysis, the Poisson distribution fitting is successful.

Another noticing point is the distinguished difference among the seven days in a week in terms of the number of mass shooting.

To make the prediction more accurate our group further investigate on the number of successive days of free mass shooting from 2013 to 2015 and made a corresponding prediction in 2016 or later years. There are some differences between a model of continuous interval and discrete interval and we will explain the differences and compare them

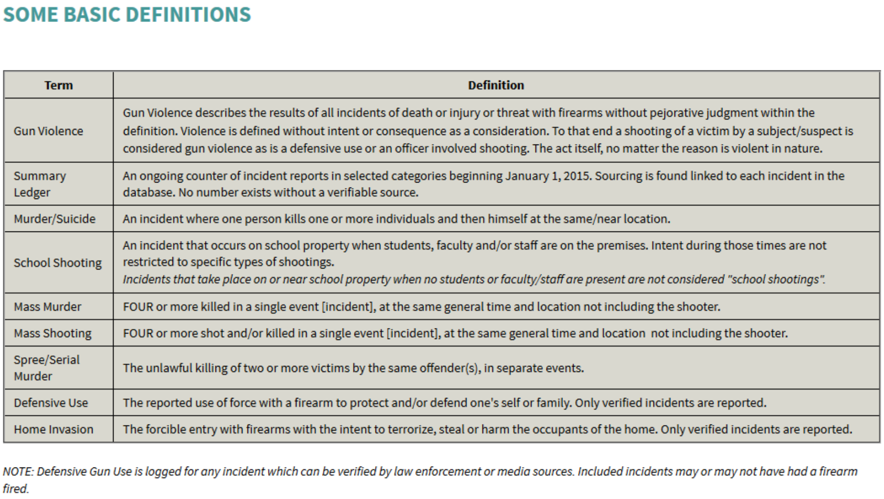
Finally a diagram indicating the cumulative number of mass shootings during the period 2013 $\sim$ 2015 is sketched. This diagram is important and meaningful since we made certain the process of mass shooting is a Poisson process, and a linear regression model is fitted. With the prediction, a prediction band is also added.

Reference to these models all come from GVA and our models may provide some defects. However, our group are trying to do our best in visualizing data and making predictions.

2.1 Definition Formalizing

By searching on the Internet as well as several e-books, our group found several versions of definition on **"Mass Shooting"** and **"Mass murder"**.

2.1.1 GVA (Gun Violence Archive)



SOME BASIC DEFINITIONS

Term	Definition
Gun Violence	Gun Violence describes the results of all incidents of death or injury or threat with firearms without pejorative judgment within the definition. Violence is defined without intent or consequence as a consideration. To that end a shooting of a victim by a subject/suspect is considered gun violence as is a defensive use or an officer involved shooting. The act itself, no matter the reason is violent in nature.
Summary Ledger	An ongoing counter of incident reports in selected categories beginning January 1, 2015. Sourcing is found linked to each incident in the database. No number exists without a verifiable source.
Murder/Suicide	An incident where one person kills one or more individuals and then himself at the same/near location.
School Shooting	An incident that occurs on school property when students, faculty and/or staff are on the premises. Intent during those times are not restricted to specific types of shootings. *Incidents that take place on or near school property when no students or faculty/staff are present are not considered "school shootings".*
Mass Murder	FOUR or more killed in a single event [incident], at the same general time and location not including the shooter.
Mass Shooting	FOUR or more shot and/or killed in a single event [incident], at the same general time and location not including the shooter.
Spree/Serial Murder	The unlawful killing of two or more victims by the same offender(s), in separate events.
Defensive Use	The reported use of force with a firearm to protect and/or defend one's self or family. Only verified incidents are reported.
Home Invasion	The forcible entry with firearms with the intent to terrorize, steal or harm the occupants of the home. Only verified incidents are reported.

NOTE: Defensive Gun Use is logged for any incident which can be verified by law enforcement or media sources. Included incidents may or may not have had a firearm fired.

Figure 2.1: Definitions of "Mass Shooting" from GVA[2]

Mass Shooting: FOUR or more shot and/or killed in a single event [incident], at the same general time and location not including the shooter.

Mass murder: FOUR or more killed in a single event [incident], at the same general time and location not including the shooter.

2.1.2 FBI (Federal Bureau of Investigation)

Mass murder: Murdering four or more persons during an event with no "cooling-off period" between the murders. A mass murder typically occurs in a single location where one or more people kill several others.[5]

2.1.3 The United States' Congressional Research Service

A "**public mass shooting**" is one in which four or more people selected indiscriminately, not including the perpetrator, are killed, echoing the **FBI definition** of the term "mass murder".

2.1.4 Mass Shooting Tracker (Unofficial)

Another unofficial definition of a **mass shooting** is an event involving the shooting (not necessarily resulting in death) of 10 or more people with no cooling-off period[4]

2.1.5 Contrast between GVA and other Sources

"This difference is that we do not count the shooter among the victims when determining if a shooting reaches the threshold of Mass Shooting. It insures a clear separation between victims of a shooting and those who perpetrate the crime. GVA also does not parse the definition to exclude any type of gun violence such as gang shooting or domestic violence. The definition is purely numerical and reflects ALL shootings which reach that statistical threshold." [1]

Another difference, between the official sources and the unofficial ones, are the threshold number (4 to 10), which is trivial.

2.2 Daily Number of Mass Shootings

Citizens strive for decent public security. If it is possible to predict the pattern of mass shootings, the government can reassure the social masses that the society order is not getting worse. Figure 2.2 shows the number of mass shootings recorded each day in America between January 2013 and December 2015 provided by Gun Violence Archive. From the figure, we can find that the mass shootings maybe can be treated as random incidents, however, we still need more evidence to come up with the conclusion.

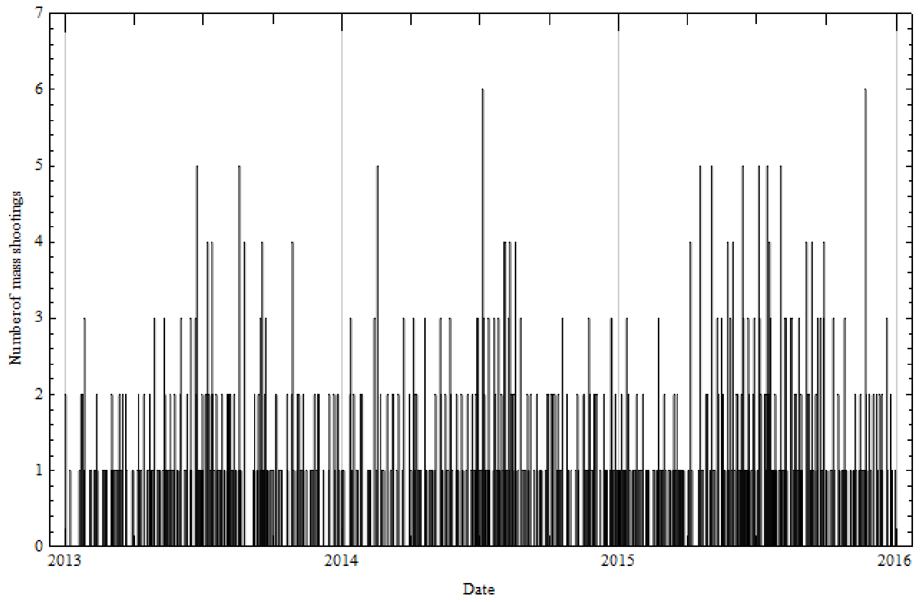


Figure 2.2: Number of mass shootings recorded each day in America between January 2013 and December 2015

2.3 Hypothesis Test for Poisson Distribution

2.3.1 From 2013 to 2015

We want to use Pearsons Chi-Squared Goodness-of-Fit Test to test whether the occurrence of mass shootings in the US follows a Poisson distribution in the three years 2013-2015. We use the sample mean as the estimator for Poisson parameter $\hat{k} = \frac{863}{1095} = 0.79$.

Therefore

$$\begin{aligned} P[X = 0] =& 0.454 \\ P[X = 1] =& 0.358 \\ P[X = 2] =& 0.142 \\ P[X = 3] =& 0.0373 \\ P[X = 4] =& 0.00737 \\ P[X = 5] =& 0.00116 \\ P[X \geq 6] =& 0.00017 \end{aligned}$$

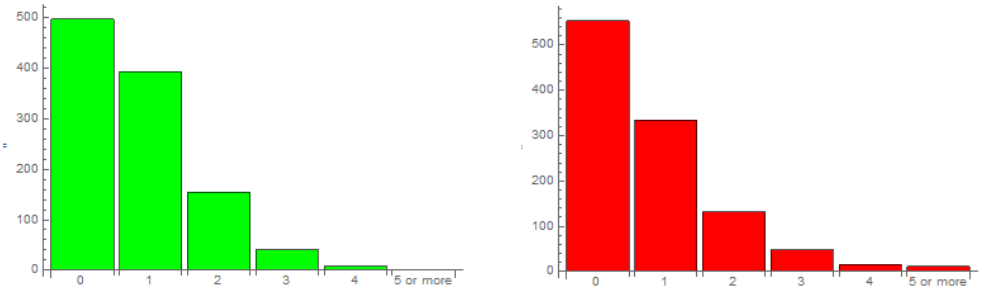


Figure 2.3: (a) Expected

Figure 2.4: (b) Observed

Figure 2.5: Frequency of occurrence of days with different numbers of mass shooting in 3 years

Then we calculate the expected frequencies $E_i = np_i$.

$$\begin{aligned} E_0 =& 496.96 \\ E_1 =& 392.60 \\ E_2 =& 155.08 \\ E_3 =& 40.48 \\ E_4 =& 8.06 \\ E_5 =& 1.27 \\ E_6 =& 0.19 \end{aligned}$$

Since $E_5, E_6 < 5$, we should make sure 80% of the expected values are greater than 5. Hence we combine the last two categories.

Number of mass shooting per day	0	1	2	3	4	5 or more
Expected	496.96	392.6	155.08	40.84	8.06	1.46
Observed	554	334	132	48	16	11

Table 2.1: Observed and expected number of days with mass shootings occurring in US on 1095 days between 2013 and 2015

The graphic description are figure 2.3 and figure 2.4.

H_0: Number of days between 3 years follows a Poisson distribution with parameter k=0.79

$$\chi^2_4 = \sum_{i=0}^{5} \frac{(O_i - E_i)^2}{E_i} = 90.14$$

The P-value $P < 0.005$ is quite small, we can reject H_0. Hence the occurrence of mass shootings in the US does not follow a Poisson distribution in the three years with $k = 0.79$.

2.3.2 In 2013

Then we will test the individual years.

For the year 2013, we recalculate the estimator for Poisson parameter $\hat{k} = 0.696$.

Number of mass shooting per day	0	1	2	3	4 or more
Expected	181.98	126.66	44.08	10.22	2.06
Observed	197	106	47	8	7

Table 2.2: Observed and expected number of days of mass shooting occurring in US on 365 days in 2013

H_0: Number of days in 2013 follows a Poisson distribution with parameter $k = 0.696$.

$$\chi_3^2 = \sum_{i=0}^{4} \frac{(O_i - E_i)^2}{E_i} = 17.13$$

The P-value $P < 0.005$ is quite small, we can reject H_0. Hence the occurrence of mass shootings in the US does not follow a Poisson distribution in 2013 with $k = 0.696$.

2.3.3 In 2014

For the year 2014, we recalculate the estimator for Poisson parameter $\hat{k} = 0.76$.

Number of mass shooting per day	0	1	2	3	4 or more
Expected	170.7	129.73	49.3	12.49	2.78
Observed	194	99	47	19	6

Table 2.3: Observed and expected number of days of mass shooting occurring in US on 365 days in 2014

H_0: Number of days in 2014 follows a Poisson distribution with parameter $k = 0.76$.

$$\chi_3^2 = \sum_{i=0}^{4} \frac{(O_i - E_i)^2}{E_i} = 17.69$$

The P-value $P < 0.005$ is quite small, we can reject H_0. Hence the occurrence of mass shootings in the US does not follow a Poisson distribution in 2014 with $k = 0.76$.

2.3.4 In 2015

For the year 2015, we recalculate the estimator for Poisson parameter $\hat{k} = 0.91$.

Number of mass shooting per day	0	1	2	3	4 or more
Expected	146.92	133.7	60.83	18.45	5.1
Observed	163	129	38	21	14

Table 2.4: Observed and expected number of days of mass shooting occurring in US on 365 days in 2015

H_0: Number of days in 2015 follows a Poisson distribution with parameter $k = 0.91$.

$$\chi_3^2 = \sum_{i=0}^{4} \frac{(O_i - E_i)^2}{E_i} = 26.38$$

The P-value $P < 0.005$ is quite small, we can reject H_0. Hence the occurrence of mass shootings in the US does not follow a Poisson distribution in 2015 with $k = 0.91$.

2.4 Test Dependence on Weekdays and Month-s

2.4.1 On Weekdays

First we examine the possible dependence on the weekday. We first display the statistic result with a table.

Weekday	Mon.	Tue.	Wed.	Thur.	Fri.	Sat.	Sun.
Observed	87	73	79	66	95	199	264
Expected	123.3	123.3	123.3	123.3	123.3	123.3	123.2

Table 2.5: Observed and expected number of mass shooting occurring in weekdays

The display of diagram is figure 2.6

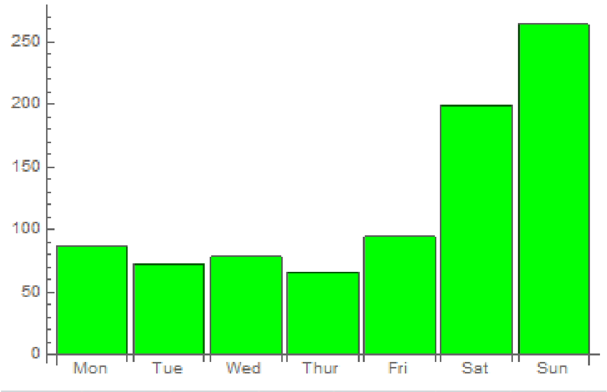


Figure 2.6: Frequency of occurrence of mass shooting in weekdays

H_0: number of mass shooting occurring in weekdays follows a uniform distribution.

$$\chi_6^2 = \sum_{i=0}^{7} \frac{(O_i - E_i)^2}{E_i} = 287.64$$

The P-value $P < 0.005$ is quite small, we can reject H_0. Hence the number of mass shooting depends on weekdays. From figure 2.6 we found that the number of mass shooting in weekend is much larger than other days.

2.4.2 On Months

Next we examine the possible dependence on the months. We first display the statistic result with a table.

Month	Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sep.	Oct.	Nov.	Dec.
Observed	52	46	55	56	80	95	52	46	55	56	80	95
Expected	71.9	71.9	71.9	71.9	71.9	71.9	71.9	71.9	71.9	71.9	71.9	72.1

Table 2.6: Observed and expected number of mass shooting occurring in different months

The display of diagram is figure 2.7

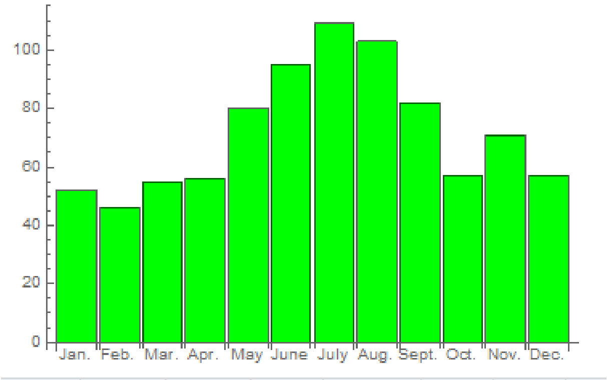


Figure 2.7: Frequency of occurrence of mass shooting in different months

H_0: number of mass shooting occurring in weekdays follows a uniform distribution.

$$\chi^2_{11} = \sum_{i=0}^{12} \frac{(O_i - E_i)^2}{E_i} = 70.94$$

The P-value $P < 0.005$ is quite small, we can reject H_0. Hence the number of mass shooting depends on months. From figure 2.7 we found that the number of mass shooting in the middle of the year is larger than other months.

2.5 Test the gaps between mass shootings

In this part we would use two different approach to find out the pattern of the gaps between mass shootings.

1. Consider that the time is continuous. For this approach, "1 day" means 24 hours.

2. Consider that the time is discrete. For this approach, ”1 day” means 1 natural day.

2.5.1 Assume Time is Continuous

To begin with, we consider the time is continuous. Since if the time is a continuum, the Poisson distribution implies an exponentially distributed time interval between mass shootings (cite project). To use an exponentially distribution, firstly we need to calculate the parameter λ of the Poisson distribution:

$$\lambda = \frac{total\ number\ of\ mass\ shootings}{total\ number\ of\ days} = \frac{863}{1095} = 0.7881$$

Since we know that $\beta = 1/\lambda$, we now have the probability distribution function and the cumulative distribution function of our exponential distribution:

$$\begin{aligned} f_X(x) &= \lambda e^{-\lambda x}, & \lambda &= 0.7881 \\ F_X(x) &= 1 - e^{-\lambda x}, & x &> 0 \end{aligned}$$

Now we can calculate the probability of having complete n days between mass shootings using (cite project):

$$P(n) = F_x(n+1) - F_X(n), \quad n > 0$$

After calculating, we can compare the result with observed values.

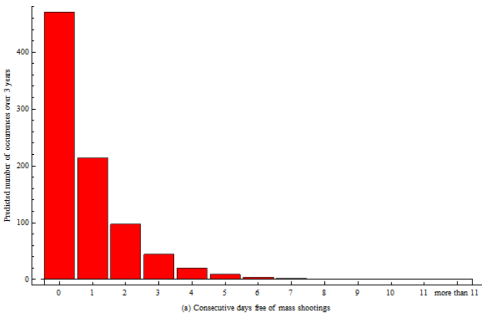


Figure 2.8: (a) Predicted

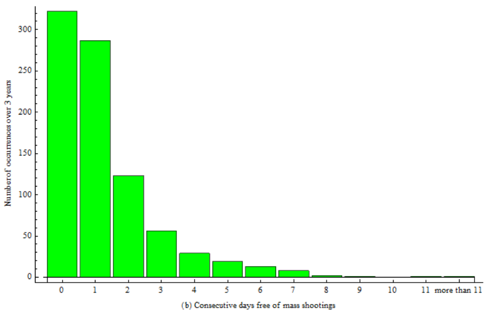


Figure 2.9: (b) Actual

Figure 2.10: The length of gaps between mass shooting (continuous)

Consecutive Days Free of Mass Shootings	Expected Days	Observed Days
0	470.05	322
1	213.73	287
2	97.18	123
3	44.19	56
4	20.09	29
5	9.14	19
6	4.15	13
7	1.89	8
8	0.86	2
9	0.39	1
10	0.18	0
11	0.08	1
More than 11	0.07	1

Table 2.7: Expected and observed length of gaps between mass shootings (continuous)

Figure 2.10 shows that the observed pattern does not follow our predicted pattern well, so we do not have evidence to say that the time interval between mass shootings is exponential distributed.

2.5.2 Assume Time is Discrete

Next, we consider the time is discrete. Since if the time is discrete, every day would have the same probability of having mass shootings (maybe more

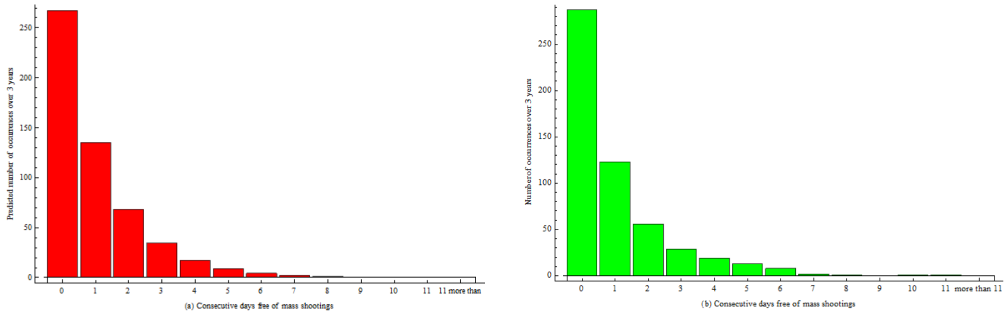


Figure 2.11: (a) Predicted

Figure 2.12: (b) Actual

Figure 2.13: The length of gaps between mass shooting (discrete)

than 1). To predict the pattern of no mass shootings on n successive days, we use a geometric distributed model. First, we can calculate the probability p by:

$$p = \frac{total\ days\ of\ mass\ shootings}{total\ number\ of\ days} = \frac{541}{1095} = 0.4941$$

Now we can calculate the probability of having n days free of mass shootings between two days having mass shootings by:

$$f(n) = (1-p)^n p, \qquad p = 0.4941$$

After calculating, we can compare the result with observed values.

Consecutive Days Free of Mass Shootings	Expected Days	Observed Days
0	267.29	287
1	135.23	123
2	68.42	56
3	34.62	29
4	17.51	19
5	8.86	13
6	4.48	8
7	2.27	2
8	1.15	1
9	0.58	0
10	0.29	1
11	0.15	1
More than 11	0.15	0

Table 2.8: Expected and observed length of gaps between mass shootings (discrete)

Figure 2.13 shows that the observed pattern follows our predicted pattern well, so we have evidence to say that the time interval of no mass shootings on n successive days is geometric distributed.

To conclude with, the second approach is much better than the first one. We should consider that the time is discrete when dealing with this kind of question.

2.6 Research on cumulative number of mass shootings

In the previous chapter, we observed that it's not appropriate to model the number of mass shootings with Poisson Distribution. However, we hypothesized that there is somewhat linear with the cumulative number of mass shootings with x axis to be time period.

First, according to the demand, I merged three line charts into a single graph. The line chats indicate the increase tendency of cumulative number in 2013 to 2015.

Secondly, we make a linear regression model for all of the three models. With these three models, the level of growth will be apparent.

Finally we are interested in the prediction band of each model.

2.6.1 Line Chart Representation

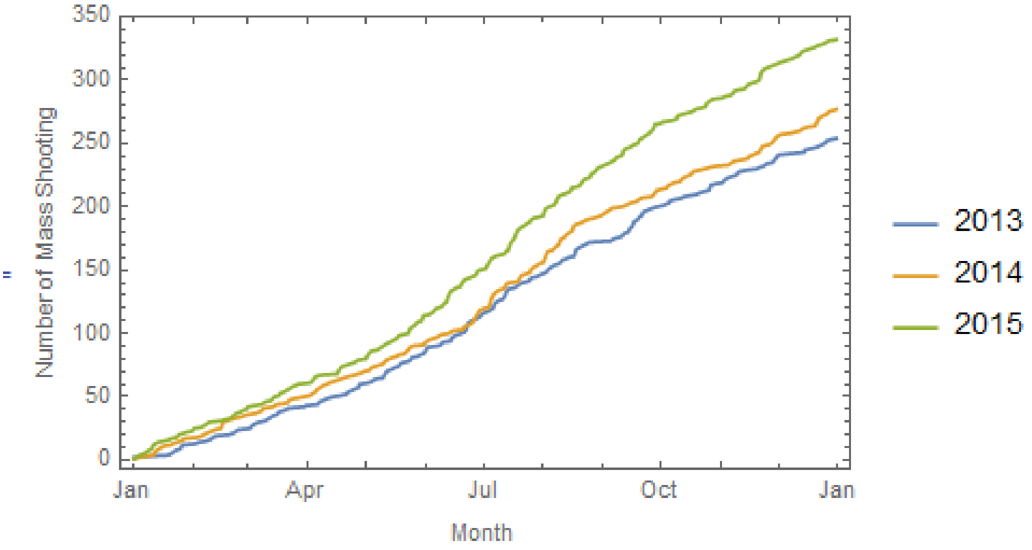


Figure 2.14: Line chart representation

Figure 2.14 represent the cumulative number of mass shootings during 2013 to 2015.

2.6.2 Linear Model Fitting

First we would like to derive the linear fitting model for all of the three years,

In 2013, the linear model is

$$C_1(x) = -23.6611 + 0.786404x$$

where x is the xth day in a year, e.g. the x value for *Dec* 31*st* is 365.

Similarly, in 2014, the linear model is

$$C_2(x) = -17.1043 + 0.812581x$$

In 2015, the linear model is

$$C_3(x) = -19.3567 + 0.989794x$$

Therefore the synthesized graph is figure 2.15 .

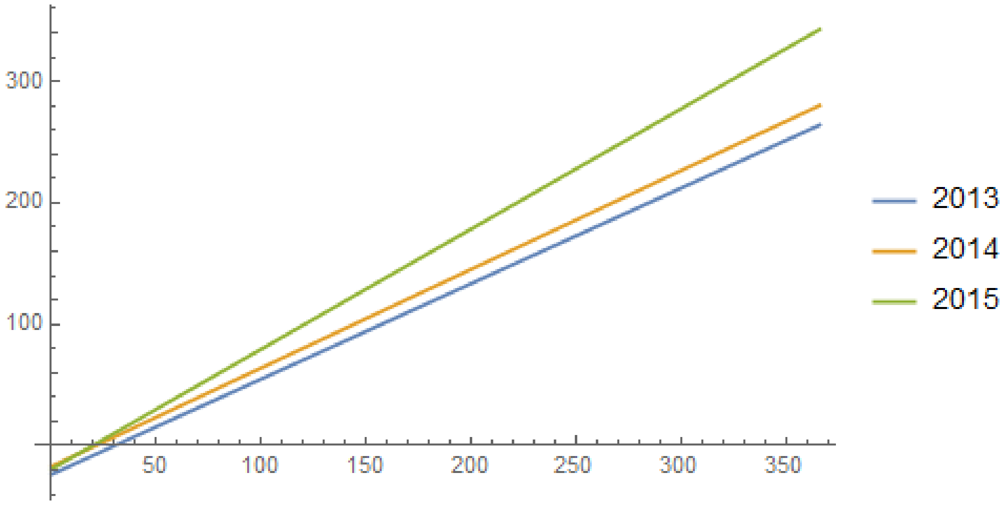


Figure 2.15: Regression model

2.6.3 Prediction Band

To visualize relative prediction of the mass shootings, we would like to provide the prediction band.

Figure 2.16 refer to the band in 2013, note that the function of prediction band is given by:

$$\pm 1.97436\sqrt{0.0000392089x^2 - 0.0147267x + 61.8686} + 0.786404x - 23.6611$$

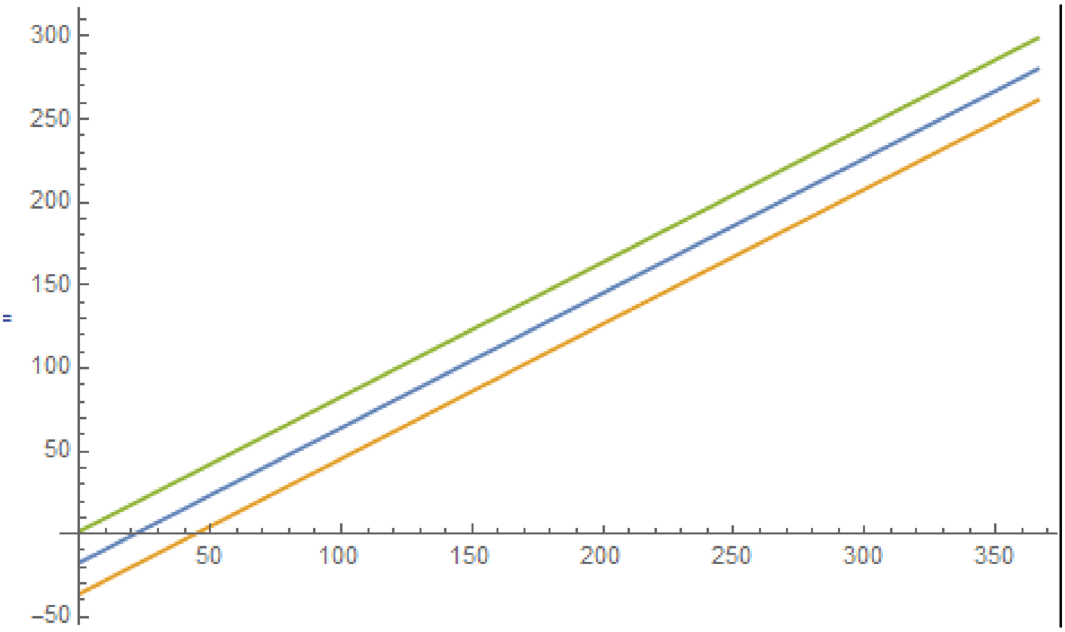


Figure 2.16: Regression model

The prediction band of 2014 is given by:

$$\pm 1.9741\sqrt{0.0000517194x^2 - 0.0201179x + 90.7843} + 0.812581x - 17.1043$$

The prediction band of 2015 is given by:

$$\pm 1.9719\sqrt{0.0000577928x^2 - 0.0211899x + 124.066} + 0.989794x - 19.3567$$

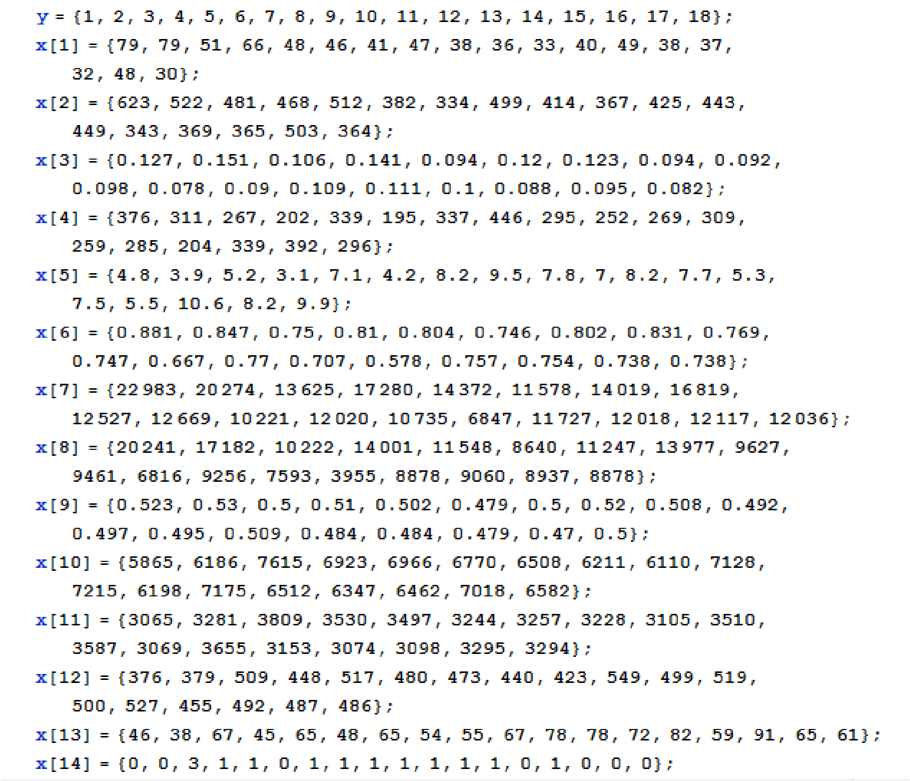
Since the band is similar to that of 2013 we omit the plot. However the plot will occur in the appendix.

Chapter 3

The German Bundesliga 2015-16

Football is the most popular sports in the world, and The German Bundesliga is one of the most famous football matches. Different football teams compete with each other and get their standing according to their performance. As we all know, there are many criterions used to evaluate one teams strength and ability, such as goal success rate, passes success rate, and fouls they get. We are interested in which criterions will make a difference to the teams standings, how these criterions affect the standings, and which of them counts most. Therefore, we will use methods of model selection to analyze this problem, based on the data of The German Bundesliga from 2015 to 2016.

3.1 Choose and input data

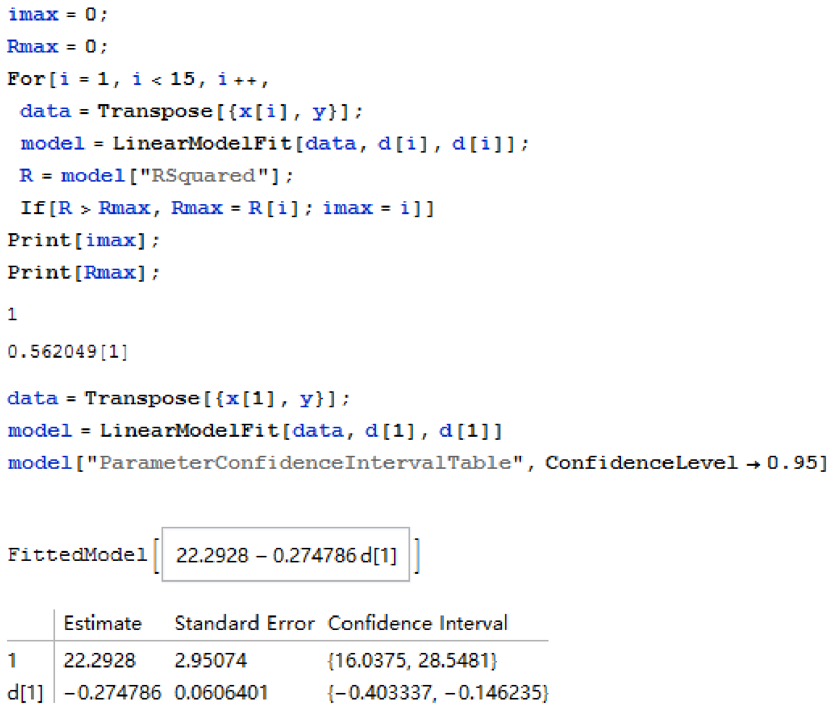


```
y = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18};
x[1] = {79, 79, 51, 66, 48, 46, 41, 47, 38, 36, 33, 40, 49, 38, 37,
   32, 48, 30};
x[2] = {623, 522, 481, 468, 512, 382, 334, 499, 414, 367, 425, 443,
   449, 343, 369, 365, 503, 364};
x[3] = {0.127, 0.151, 0.106, 0.141, 0.094, 0.12, 0.123, 0.094, 0.092,
   0.098, 0.078, 0.09, 0.109, 0.111, 0.1, 0.088, 0.095, 0.082};
x[4] = {376, 311, 267, 202, 339, 195, 337, 446, 295, 252, 269, 309,
   259, 285, 204, 339, 392, 296};
x[5] = {4.8, 3.9, 5.2, 3.1, 7.1, 4.2, 8.2, 9.5, 7.8, 7, 8.2, 7.7, 5.3,
   7.5, 5.5, 10.6, 8.2, 9.9};
x[6] = {0.881, 0.847, 0.75, 0.81, 0.804, 0.746, 0.802, 0.831, 0.769,
   0.747, 0.667, 0.77, 0.707, 0.578, 0.757, 0.754, 0.738, 0.738};
x[7] = {22983, 20274, 13625, 17280, 14372, 11578, 14019, 16819,
   12527, 12669, 10221, 12020, 10735, 6847, 11727, 12018, 12117, 12036};
x[8] = {20241, 17182, 10222, 14001, 11548, 8640, 11247, 13977, 9627,
   9461, 6816, 9256, 7593, 3955, 8878, 9060, 8937, 8878};
x[9] = {0.523, 0.53, 0.5, 0.51, 0.502, 0.479, 0.5, 0.52, 0.508, 0.492,
   0.497, 0.495, 0.509, 0.484, 0.484, 0.479, 0.47, 0.5};
x[10] = {5865, 6186, 7615, 6923, 6966, 6770, 6508, 6211, 6110, 7128,
   7215, 6198, 7175, 6512, 6347, 6462, 7018, 6582};
x[11] = {3065, 3281, 3809, 3530, 3497, 3244, 3257, 3228, 3105, 3510,
   3587, 3069, 3655, 3153, 3074, 3098, 3295, 3294};
x[12] = {376, 379, 509, 448, 517, 480, 473, 440, 423, 549, 499, 519,
   500, 527, 455, 492, 487, 486};
x[13] = {46, 38, 67, 45, 65, 48, 65, 54, 55, 67, 78, 78, 72, 82, 59, 91, 65, 61};
x[14] = {0, 0, 3, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0};
```

Figure 3.1: Input data table
]

In the data table(Figure 3.1), there are 16 sets of data about teams performance in total. However, we find that the success rate and the failure rate have a linear relationship (success rate = 1 failure rate), so when we build a linear model, there is no need to take the failure rate into account. Therefore, we finally choose 14 sets of data about performance as following: x[1] represents goals, x[2] represents shots on, x[3] represents goal success rate, x[4] represents crosses, x[5] represents crosses/goal, x[6] represents passes success rate, x[7] represents passes(total), x[8] represents passes(successful), x[9] represents tackles success rate, x[10] represents tackles(total) , x[11] represents tackles(successful), x[12] represents fouls, x[13] represents yellow cards, x[14] represents red cards. Besides, we use y to denote teams standing.

3.2 Forward Selection Method

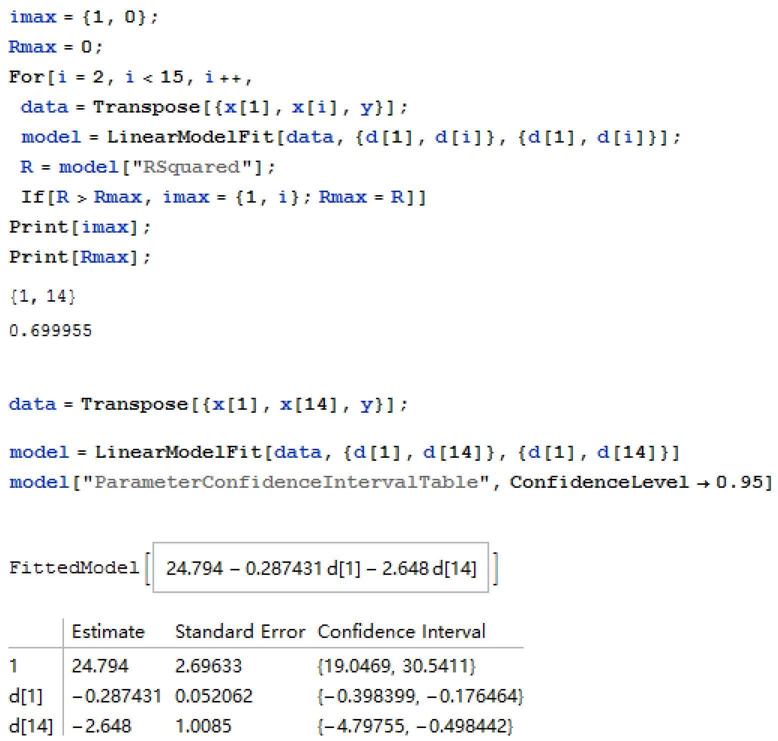


```
imax = 0;
Rmax = 0;
For[i = 1, i < 15, i ++,
 data = Transpose[{x[i], y}];
 model = LinearModelFit[data, d[i], d[i]];
 R = model["RSquared"];
 If[R > Rmax, Rmax = R[i]; imax = i]]
Print[imax];
Print[Rmax];

1

0.562049[1]

data = Transpose[{x[1], y}];
model = LinearModelFit[data, d[1], d[1]]
model["ParameterConfidenceIntervalTable", ConfidenceLevel → 0.95]

FittedModel[ 22.2928 − 0.274786 d[1] ]
```

	Estimate	Standard Error	Confidence Interval
1	22.2928	2.95074	{16.0375, 28.5481}
d[1]	−0.274786	0.0606401	{−0.403337, −0.146235}

Figure 3.2: Forward Selection Method (Step 1)
]

Step 1: Refer to 3.2 We build 14 single-variable models, and find the one with the maximum R^2. The result is x[1] has the maximum R^2. Then we test whether the coefficient of x[1] equals to zero, and find that 0 is not in the confidence interval. So, we need to take x[1] into account.



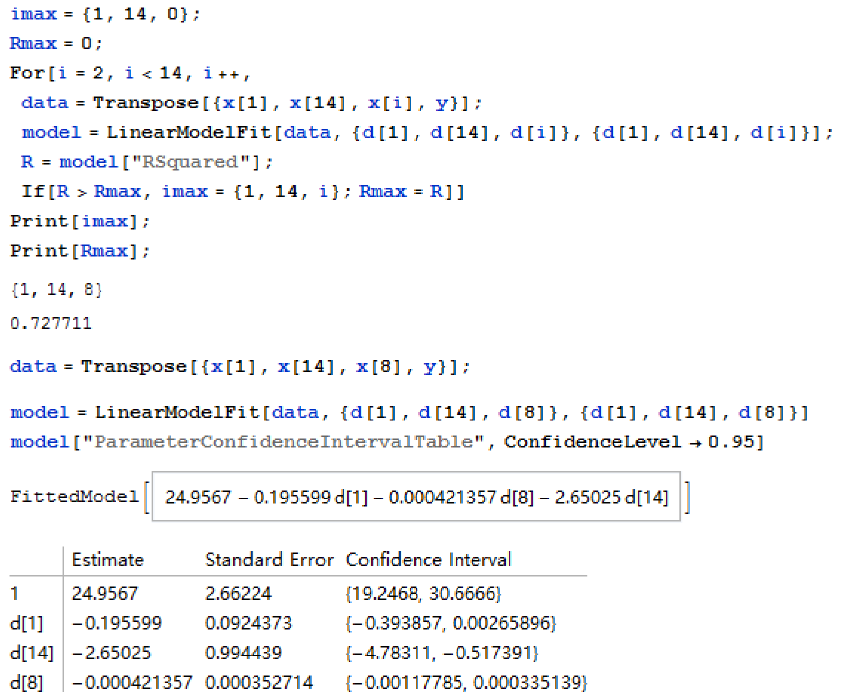
```
imax = {1, 0};
Rmax = 0;
For[i = 2, i < 15, i ++,
 data = Transpose[{x[1], x[i], y}];
 model = LinearModelFit[data, {d[1], d[i]}, {d[1], d[i]}];
 R = model["RSquared"];
 If[R > Rmax, imax = {1, i}; Rmax = R]]
Print[imax];
Print[Rmax];

{1, 14}

0.699955


data = Transpose[{x[1], x[14], y}];

model = LinearModelFit[data, {d[1], d[14]}, {d[1], d[14]}]
model["ParameterConfidenceIntervalTable", ConfidenceLevel → 0.95]


FittedModel[ 24.794 − 0.287431 d[1] − 2.648 d[14] ]
```

	Estimate	Standard Error	Confidence Interval
1	24.794	2.69633	{19.0469, 30.5411}
d[1]	−0.287431	0.052062	{−0.398399, −0.176464}
d[14]	−2.648	1.0085	{−4.79755, −0.498442}

Figure 3.3: Forward Selection Method (Step 2)
]

Step 2: With x[1] fitted, we build 13 two-variable models, and find the one with the max R^2. The result is x[14]. Then we test whether the coefficient of x[14] equals to zero, and find that 0 is not in the confidence interval. So, we need to take x[14] into account.

Step 3: Refer to figure 3.4. With x[1] and x[14] fitted, we build 12 three-variable models, and find the one with the max R^2. The result is x[8]. Then we test whether the coefficient of x[8] equals to zero, and find that 0 is in the confidence interval. So, we dont need to take x[8] into account.



```
imax = {1, 14, 0};
Rmax = 0;
For[i = 2, i < 14, i ++,
 data = Transpose[{x[1], x[14], x[i], y}];
 model = LinearModelFit[data, {d[1], d[14], d[i]}, {d[1], d[14], d[i]}];
 R = model["RSquared"];
 If[R > Rmax, imax = {1, 14, i}; Rmax = R]]
Print[imax];
Print[Rmax];
```

```
{1, 14, 8}
0.727711
```

```
data = Transpose[{x[1], x[14], x[8], y}];

model = LinearModelFit[data, {d[1], d[14], d[8]}, {d[1], d[14], d[8]}]
model["ParameterConfidenceIntervalTable", ConfidenceLevel → 0.95]
```

```
FittedModel[ 24.9567 – 0.195599 d[1] – 0.000421357 d[8] – 2.65025 d[14] ]
```

	Estimate	Standard Error	Confidence Interval
1	24.9567	2.66224	{19.2468, 30.6666}
d[1]	−0.195599	0.0924373	{−0.393857, 0.00265896}
d[14]	−2.65025	0.994439	{−4.78311, −0.517391}
d[8]	−0.000421357	0.000352714	{−0.00117785, 0.000335139}

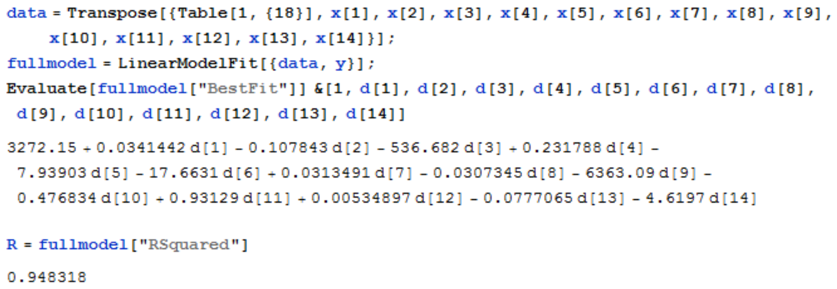
Figure 3.4: Forward Selection Method (Step 3)

]

Therefore, we only need to consider x[1] and x[14], and the model we get is: y=24.79395902847831-0.287431132148677x[1]-2.6479979824629463x[14]

3.3 Backward Elimination

Step 1: Refer to figure 3.5.At first we build a full model and find the R^2.



```
data = Transpose[{Table[1, {18}], x[1], x[2], x[3], x[4], x[5], x[6], x[7], x[8], x[9],
    x[10], x[11], x[12], x[13], x[14]}];
fullmodel = LinearModelFit[{data, y}];
Evaluate[fullmodel["BestFit"]] &[1, d[1], d[2], d[3], d[4], d[5], d[6], d[7], d[8],
 d[9], d[10], d[11], d[12], d[13], d[14]]
```

```
3272.15 + 0.0341442 d[1] - 0.107843 d[2] - 536.682 d[3] + 0.231788 d[4] -
 7.93903 d[5] - 17.6631 d[6] + 0.0313491 d[7] - 0.0307345 d[8] - 6363.09 d[9] -
 0.476834 d[10] + 0.93129 d[11] + 0.00534897 d[12] - 0.0777065 d[13] - 4.6197 d[14]
```

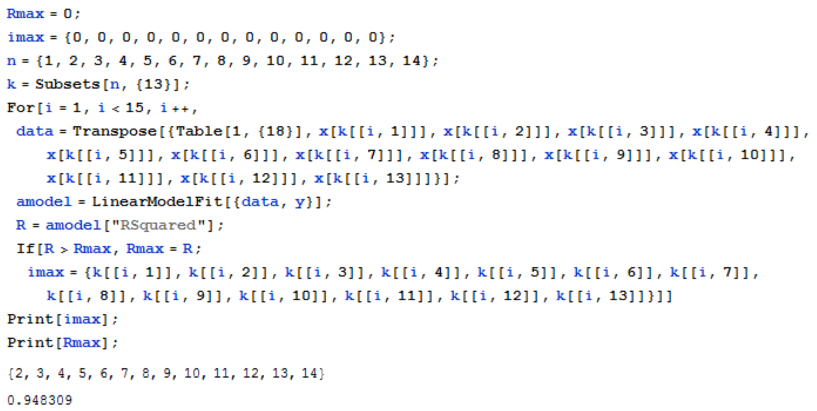
```
R = fullmodel["RSquared"]
```

```
0.948318
```

Figure 3.5: Backward Elimination (Step 1)

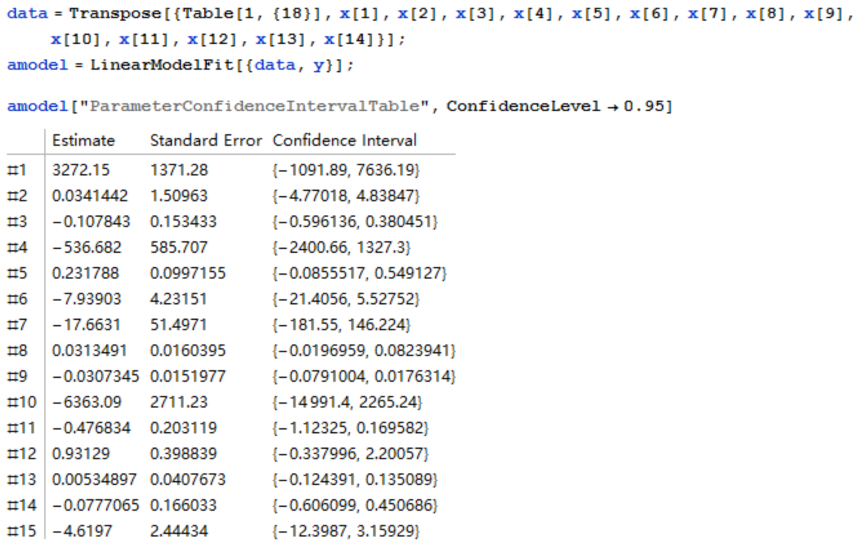
]

Step 2: Refer to figure 3.6, We need to find the one with max R^2 from the 14 thirteen- variable models. The result is the model without x[1]. Then we compare this model with the full model, and test whether the coefficient of x[1] equals to zero. Refer to Figure 3.7, We find that 0 is in the confidence interval, so that we can delete x[1] from the full model.



```
Rmax = 0;
imax = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {13}];
For[i = 1, i < 15, i ++,
 data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]], x[k[[i, 3]]], x[k[[i, 4]]],
    x[k[[i, 5]]], x[k[[i, 6]]], x[k[[i, 7]]], x[k[[i, 8]]], x[k[[i, 9]]], x[k[[i, 10]]],
    x[k[[i, 11]]], x[k[[i, 12]]], x[k[[i, 13]]]}];
 amodel = LinearModelFit[{data, y}];
 R = amodel["RSquared"];
 If[R > Rmax, Rmax = R;
  imax = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]], k[[i, 6]], k[[i, 7]],
    k[[i, 8]], k[[i, 9]], k[[i, 10]], k[[i, 11]], k[[i, 12]], k[[i, 13]]}]]
Print[imax];
Print[Rmax];

{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}
0.948309
```

Figure 3.6: Backward Elimination Modeling)

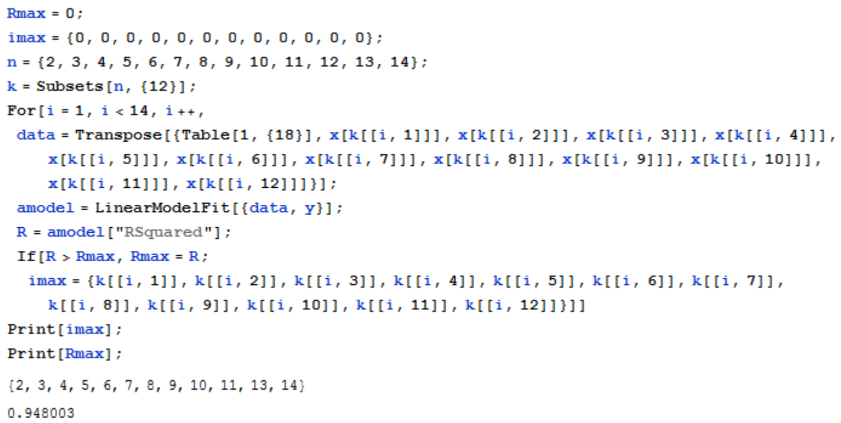


```
data = Transpose[{Table[1, {18}], x[1], x[2], x[3], x[4], x[5], x[6], x[7], x[8], x[9],
    x[10], x[11], x[12], x[13], x[14]}];
amodel = LinearModelFit[{data, y}];

amodel["ParameterConfidenceIntervalTable", ConfidenceLevel → 0.95]
```

	Estimate	Standard Error	Confidence Interval
#1	3272.15	1371.28	{−1091.89, 7636.19}
#2	0.0341442	1.50963	{−4.77018, 4.83847}
#3	−0.107843	0.153433	{−0.596136, 0.380451}
#4	−536.682	585.707	{−2400.66, 1327.3}
#5	0.231788	0.0997155	{−0.0855517, 0.549127}
#6	−7.93903	4.23151	{−21.4056, 5.52752}
#7	−17.6631	51.4971	{−181.55, 146.224}
#8	0.0313491	0.0160395	{−0.0196959, 0.0823941}
#9	−0.0307345	0.0151977	{−0.0791004, 0.0176314}
#10	−6363.09	2711.23	{−14 991.4, 2265.24}
#11	−0.476834	0.203119	{−1.12325, 0.169582}
#12	0.93129	0.398839	{−0.337996, 2.20057}
#13	0.00534897	0.0407673	{−0.124391, 0.135089}
#14	−0.0777065	0.166033	{−0.606099, 0.450686}
#15	−4.6197	2.44434	{−12.3987, 3.15929}

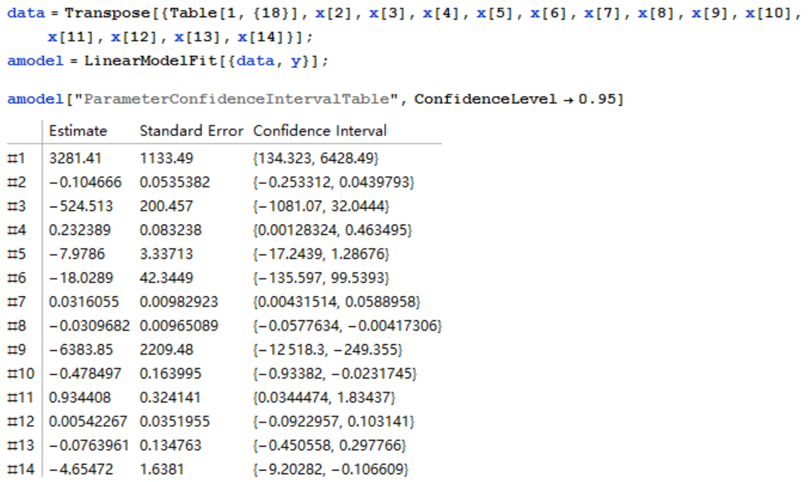
Figure 3.7: Backward Elimination Database

Step 3: Refer to figure 3.8 and figure 3.9. We need to find the one with max R^2 from the 13 twelve variable(without x[1]) models. The result is the model without x[12]. Then we compare this model with the full model, and test whether the coefficient of x[12] equals to zero. We find that 0 is in the confidence interval, so that we can delete x[12] from the full model.



```
Rmax = 0;
imax = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
n = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {12}];
For[i = 1, i < 14, i ++,
 data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]], x[k[[i, 3]]], x[k[[i, 4]]],
    x[k[[i, 5]]], x[k[[i, 6]]], x[k[[i, 7]]], x[k[[i, 8]]], x[k[[i, 9]]], x[k[[i, 10]]],
    x[k[[i, 11]]], x[k[[i, 12]]]}];
 amodel = LinearModelFit[{data, y}];
 R = amodel["RSquared"];
 If[R > Rmax, Rmax = R;
  imax = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]], k[[i, 6]], k[[i, 7]],
    k[[i, 8]], k[[i, 9]], k[[i, 10]], k[[i, 11]], k[[i, 12]]}]]
Print[imax];
Print[Rmax];

{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14}

0.948003
```

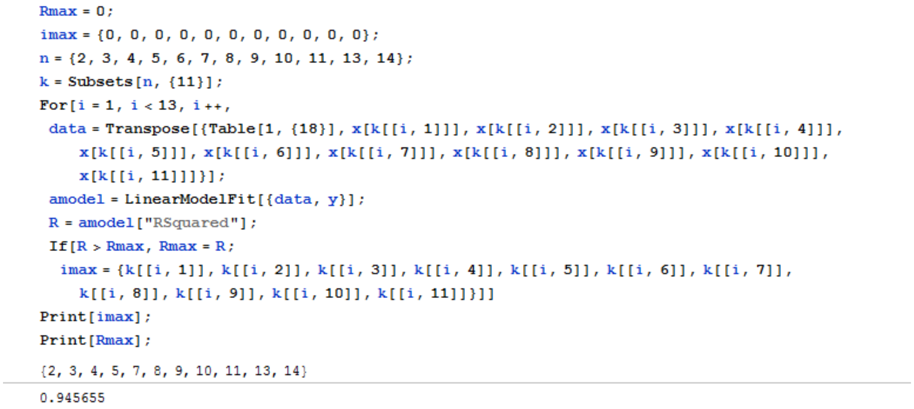
Figure 3.8: Backward Elimination Step 3 One)



```
data = Transpose[{Table[1, {18}], x[2], x[3], x[4], x[5], x[6], x[7], x[8], x[9], x[10],
    x[11], x[12], x[13], x[14]}];
amodel = LinearModelFit[{data, y}];

amodel["ParameterConfidenceIntervalTable", ConfidenceLevel → 0.95]

    | Estimate    Standard Error  Confidence Interval
----+------------------------------------------------
#1  | 3281.41     1133.49         {134.323, 6428.49}
#2  | −0.104666   0.0535382       {−0.253312, 0.0439793}
#3  | −524.513    200.457         {−1081.07, 32.0444}
#4  | 0.232389    0.083238        {0.00128324, 0.463495}
#5  | −7.9786     3.33713         {−17.2439, 1.28676}
#6  | −18.0289    42.3449         {−135.597, 99.5393}
#7  | 0.0316055   0.00982923      {0.00431514, 0.0588958}
#8  | −0.0309682  0.00965089      {−0.0577634, −0.00417306}
#9  | −6383.85    2209.48         {−12518.3, −249.355}
#10 | −0.478497   0.163995        {−0.93382, −0.0231745}
#11 | 0.934408    0.324141        {0.0344474, 1.83437}
#12 | 0.00542267  0.0351955       {−0.0922957, 0.103141}
#13 | −0.0763961  0.134763        {−0.450558, 0.297766}
#14 | −4.65472    1.6381          {−9.20282, −0.106609}
```

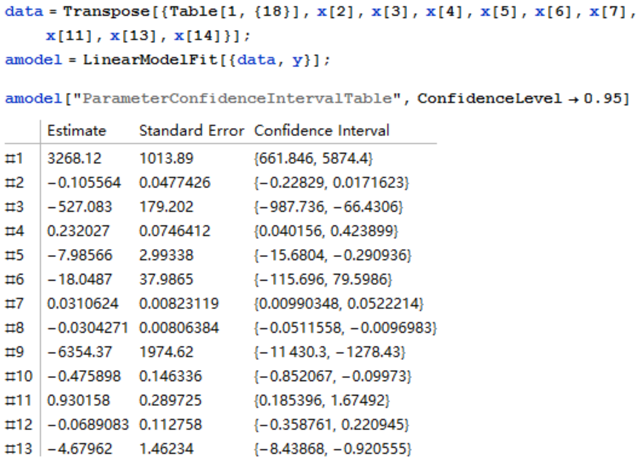
Figure 3.9: Backward Elimination Step 3 Two

Step 4: Refer to figure 3.10 and figure 3.11. We need to find the one with max R^2 from the 12 eleven variables(without x[1] and x[12]) models. The result is the model without x[6]. Then we compare this model with the full model, and test whether the coefficient of x[6] equals to zero. We find that 0 is in the confidence interval, so that we can delete x[6] from the full model.



```
Rmax = 0;
imax = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
n = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14};
k = Subsets[n, {11}];
For[i = 1, i < 13, i ++,
 data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]], x[k[[i, 3]]], x[k[[i, 4]]],
    x[k[[i, 5]]], x[k[[i, 6]]], x[k[[i, 7]]], x[k[[i, 8]]], x[k[[i, 9]]], x[k[[i, 10]]],
    x[k[[i, 11]]]}];
 amodel = LinearModelFit[{data, y}];
 R = amodel["RSquared"];
 If[R > Rmax, Rmax = R;
  imax = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]], k[[i, 6]], k[[i, 7]],
    k[[i, 8]], k[[i, 9]], k[[i, 10]], k[[i, 11]]}]]
Print[imax];
Print[Rmax];

{2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14}

0.945655
```

Figure 3.10: Backward Elimination Step 4 One

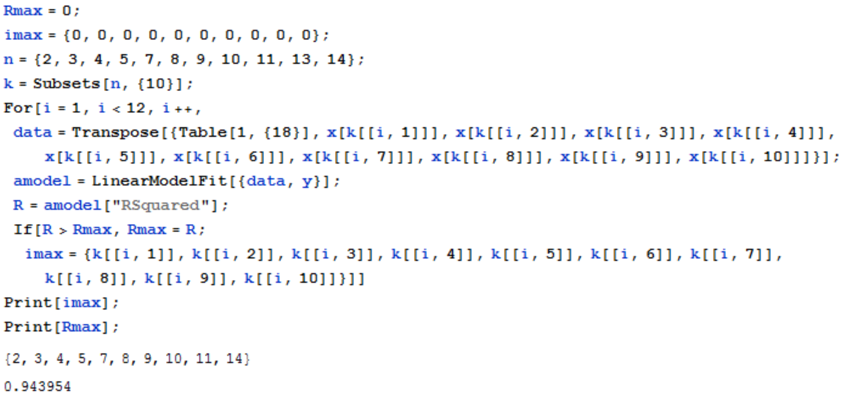


```
data = Transpose[{Table[1, {18}], x[2], x[3], x[4], x[5], x[6], x[7], x[8], x[9], x[10],
    x[11], x[13], x[14]}];
amodel = LinearModelFit[{data, y}];

amodel["ParameterConfidenceIntervalTable", ConfidenceLevel → 0.95]
```

	Estimate	Standard Error	Confidence Interval
#1	3268.12	1013.89	{661.846, 5874.4}
#2	−0.105564	0.0477426	{−0.22829, 0.0171623}
#3	−527.083	179.202	{−987.736, −66.4306}
#4	0.232027	0.0746412	{0.040156, 0.423899}
#5	−7.98566	2.99338	{−15.6804, −0.290936}
#6	−18.0487	37.9865	{−115.696, 79.5986}
#7	0.0310624	0.00823119	{0.00990348, 0.0522214}
#8	−0.0304271	0.00806384	{−0.0511558, −0.0096983}
#9	−6354.37	1974.62	{−11 430.3, −1278.43}
#10	−0.475898	0.146336	{−0.852067, −0.09973}
#11	0.930158	0.289725	{0.185396, 1.67492}
#12	−0.0689083	0.112758	{−0.358761, 0.220945}
#13	−4.67962	1.46234	{−8.43868, −0.920555}

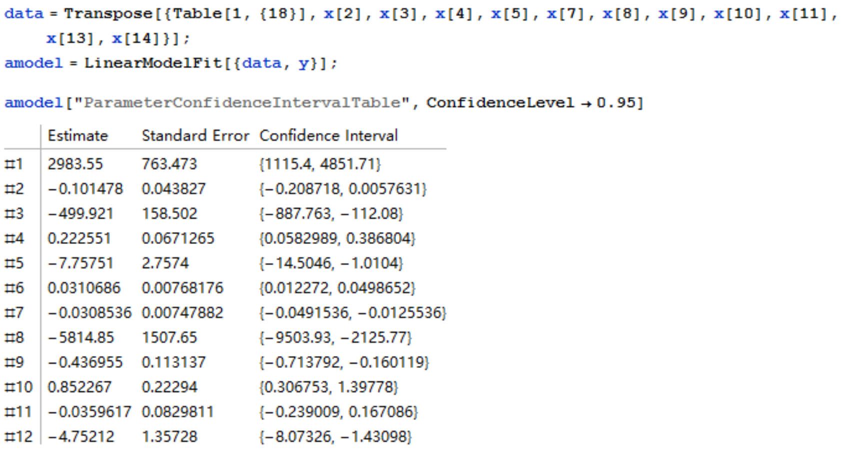
Figure 3.11: Backward Elimination Step 4 Two

Step 5: Refer to figure 3.12 and figure 3.13. We need to find the one with max R^2 from the 11 ten variables(without x[1] x[6] and x[12]) models. The result is the model without x[13]. Then we compare this model with the full model, and test whether the coefficient of x[13] equals to zero. We find that 0 is in the confidence interval, so that we can delete x[13] from the full model.



```
Rmax = 0;
imax = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
n = {2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14};
k = Subsets[n, {10}];
For[i = 1, i < 12, i ++,
 data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]], x[k[[i, 3]]], x[k[[i, 4]]],
    x[k[[i, 5]]], x[k[[i, 6]]], x[k[[i, 7]]], x[k[[i, 8]]], x[k[[i, 9]]], x[k[[i, 10]]]}];
 amodel = LinearModelFit[{data, y}];
 R = amodel["RSquared"];
 If[R > Rmax, Rmax = R;
  imax = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]], k[[i, 6]], k[[i, 7]],
    k[[i, 8]], k[[i, 9]], k[[i, 10]]}]]
Print[imax];
Print[Rmax];

{2, 3, 4, 5, 7, 8, 9, 10, 11, 14}

0.943954
```

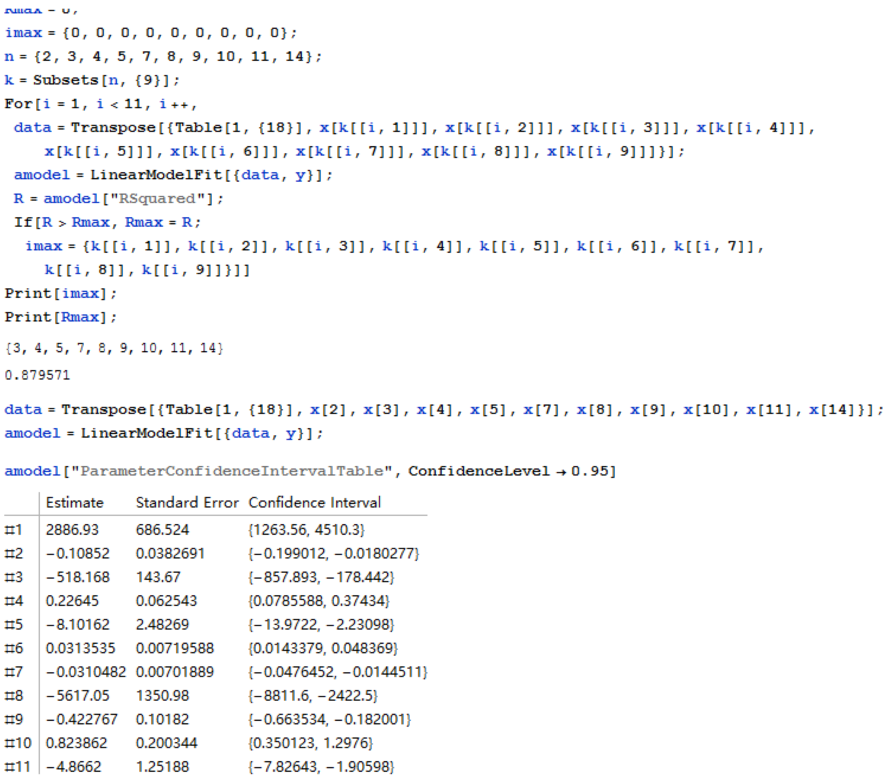
Figure 3.12: Backward Elimination Step 5 One



```
data = Transpose[{Table[1, {18}], x[2], x[3], x[4], x[5], x[7], x[8], x[9], x[10], x[11],
    x[13], x[14]}];
amodel = LinearModelFit[{data, y}];

amodel["ParameterConfidenceIntervalTable", ConfidenceLevel → 0.95]

     Estimate   Standard Error Confidence Interval
#1   2983.55    763.473        {1115.4, 4851.71}
#2   −0.101478  0.043827       {−0.208718, 0.0057631}
#3   −499.921   158.502        {−887.763, −112.08}
#4   0.222551   0.0671265      {0.0582989, 0.386804}
#5   −7.75751   2.7574         {−14.5046, −1.0104}
#6   0.0310686  0.00768176     {0.012272, 0.0498652}
#7   −0.0308536 0.00747882     {−0.0491536, −0.0125536}
#8   −5814.85   1507.65        {−9503.93, −2125.77}
#9   −0.436955  0.113137       {−0.713792, −0.160119}
#10  0.852267   0.22294        {0.306753, 1.39778}
#11  −0.0359617 0.0829811      {−0.239009, 0.167086}
#12  −4.75212   1.35728        {−8.07326, −1.43098}
```

Figure 3.13: Backward Elimination Step 5 Two

Step 6: Refer to figure 3.14. We need to find the one with max R^2 from the 10 nine variables(without x[1] x[6] x[12] and x[13]) models. The result is the model without x[2]. Then we compare this model with the full model, and test whether the coefficient of x[2] equals to zero. We find that 0 is not in the confidence interval, so that we can not delete x[2] from the full model.



```
Rmax = 0;
imax = {0, 0, 0, 0, 0, 0, 0, 0, 0};
n = {2, 3, 4, 5, 7, 8, 9, 10, 11, 14};
k = Subsets[n, {9}];
For[i = 1, i < 11, i ++,
 data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]], x[k[[i, 3]]], x[k[[i, 4]]],
    x[k[[i, 5]]], x[k[[i, 6]]], x[k[[i, 7]]], x[k[[i, 8]]], x[k[[i, 9]]]}];
 amodel = LinearModelFit[{data, y}];
 R = amodel["RSquared"];
 If[R > Rmax, Rmax = R;
  imax = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]], k[[i, 6]], k[[i, 7]],
    k[[i, 8]], k[[i, 9]]}]]
Print[imax];
Print[Rmax];

{3, 4, 5, 7, 8, 9, 10, 11, 14}

0.879571

data = Transpose[{Table[1, {18}], x[2], x[3], x[4], x[5], x[7], x[8], x[9], x[10], x[11], x[14]}];
amodel = LinearModelFit[{data, y}];

amodel["ParameterConfidenceIntervalTable", ConfidenceLevel → 0.95]

    | Estimate   Standard Error Confidence Interval
#1  | 2886.93    686.524        {1263.56, 4510.3}
#2  | −0.10852   0.0382691      {−0.199012, −0.0180277}
#3  | −518.168   143.67         {−857.893, −178.442}
#4  | 0.22645    0.062543       {0.0785588, 0.37434}
#5  | −8.10162   2.48269        {−13.9722, −2.23098}
#6  | 0.0313535  0.00719588     {0.0143379, 0.048369}
#7  | −0.0310482 0.00701889     {−0.0476452, −0.0144511}
#8  | −5617.05   1350.98        {−8811.6, −2422.5}
#9  | −0.422767  0.10182        {−0.663534, −0.182001}
#10 | 0.823862   0.200344       {0.350123, 1.2976}
#11 | −4.8662    1.25188        {−7.82643, −1.90598}
```

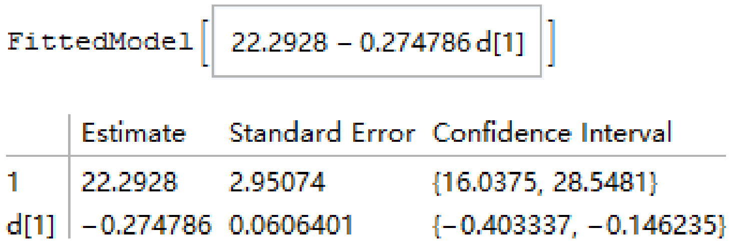
Figure 3.14: Backward Elimination Step 6

Therefore, we need to take x[2],x[3],x[4],x[5],x[7],x[8],x[9],x[10],x[11],x[14] into account. And then we can get the final model : y=2886.9260593535923-0.10851971780360774x[2]-518.16779843268x[3]+ 0.2264495651930377x[4] -8.101615010030407x[5]+ 0.031353491504968536x[7]-0.031048151833223094x[8] -5617.054232054676x[9]-0.4227672260484368x[10] +0.8238617212529299x[11]-4.866204565741854x[14]

3.4 Stepwise Selection Method

Compared with the stepwise selection method, we need to do one more thing, which is to check the former coefficients.

Step 1: Refer to figure 3.15. Only consider x[1]

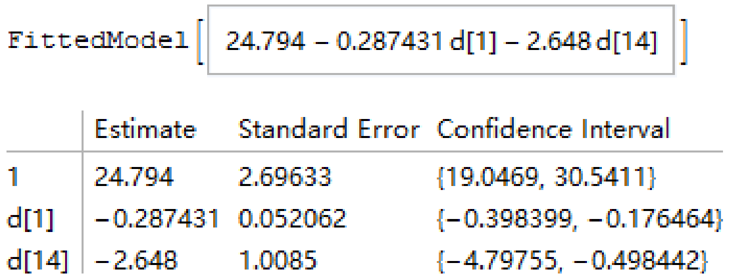


```
FittedModel[ 22.2928 − 0.274786 d[1] ]
```

	Estimate	Standard Error	Confidence Interval
1	22.2928	2.95074	{16.0375, 28.5481}
d[1]	−0.274786	0.0606401	{−0.403337, −0.146235}

Figure 3.15: Stepwise Selection Method Step 1

Step 2: Refer to figure 3.16. With x[1] fitted, consider x[14]. We find that when we take x[14] into account, the coefficient of x[1] is not zero. So, we cant delete x[1] from the model.

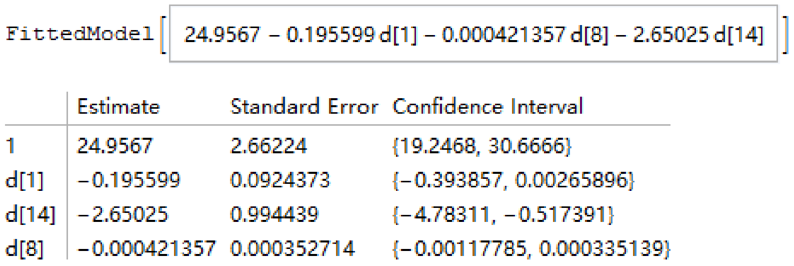


```
FittedModel[ 24.794 − 0.287431 d[1] − 2.648 d[14] ]
```

	Estimate	Standard Error	Confidence Interval
1	24.794	2.69633	{19.0469, 30.5411}
d[1]	−0.287431	0.052062	{−0.398399, −0.176464}
d[14]	−2.648	1.0085	{−4.79755, −0.498442}

Figure 3.16: Stepwise Selection Method Step 2

Step 3: Refer to figure 3.17. With x[1] and x[14] fitted, consider x[8]. We find that when we take x[8] into account, the coefficient of x[8] is zero. So, we need to delete x[8] from the model, and x[1] and x[14] should be hold.



```
FittedModel[ 24.9567 − 0.195599 d[1] − 0.000421357 d[8] − 2.65025 d[14] ]
```

	Estimate	Standard Error	Confidence Interval
1	24.9567	2.66224	{19.2468, 30.6666}
d[1]	−0.195599	0.0924373	{−0.393857, 0.00265896}
d[14]	−2.65025	0.994439	{−4.78311, −0.517391}
d[8]	−0.000421357	0.000352714	{−0.00117785, 0.000335139}

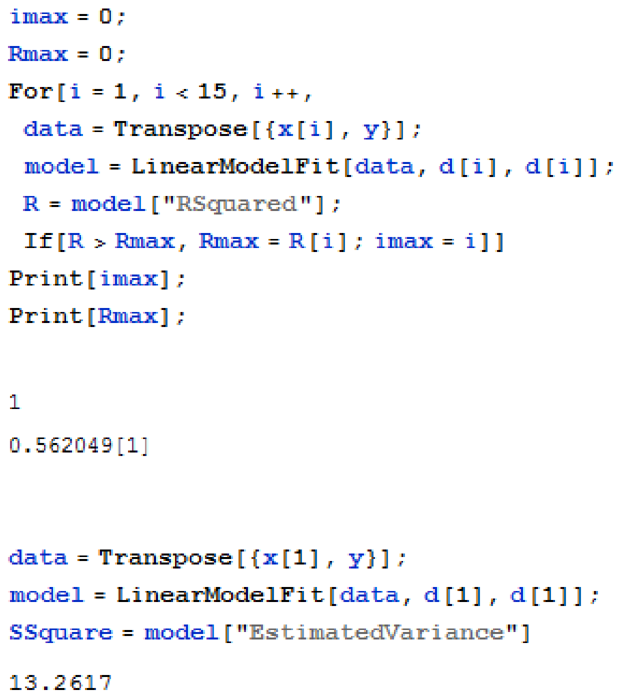
Figure 3.17: Stepwise Selection Method Step 3

Therefore, the final model we choose is same with the forward selection method: y=24.79395902847831-0.287431132148677x[1]-2.6479979824629463x[14]

3.5 Max R square selection method

We need to use this method to test all the possible models. All the possible models including the models with 1 parameter, the models with 2 parameters, the models with 3 parameters.... the models with 14 parameters. To make life easy, we use "for" loops to find the best fitted model.

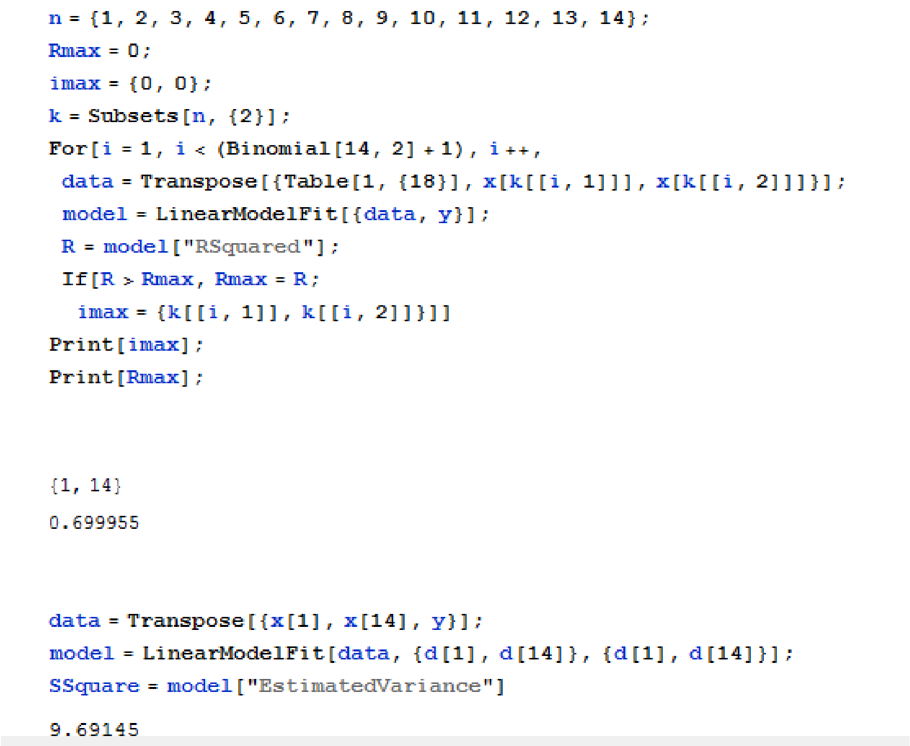
Step 1: Refer to figure 3.18. Consider one parameter and find the model with maximum R^2. The result is x[1]. Then, find the corresponding S^2;



```
imax = 0;
Rmax = 0;
For[i = 1, i < 15, i ++,
 data = Transpose[{x[i], y}];
 model = LinearModelFit[data, d[i], d[i]];
 R = model["RSquared"];
 If[R > Rmax, Rmax = R[i]; imax = i]]
Print[imax];
Print[Rmax];


1
0.562049[1]


data = Transpose[{x[1], y}];
model = LinearModelFit[data, d[1], d[1]];
SSquare = model["EstimatedVariance"]

13.2617
```

Figure 3.18: Max R square selection Step 1

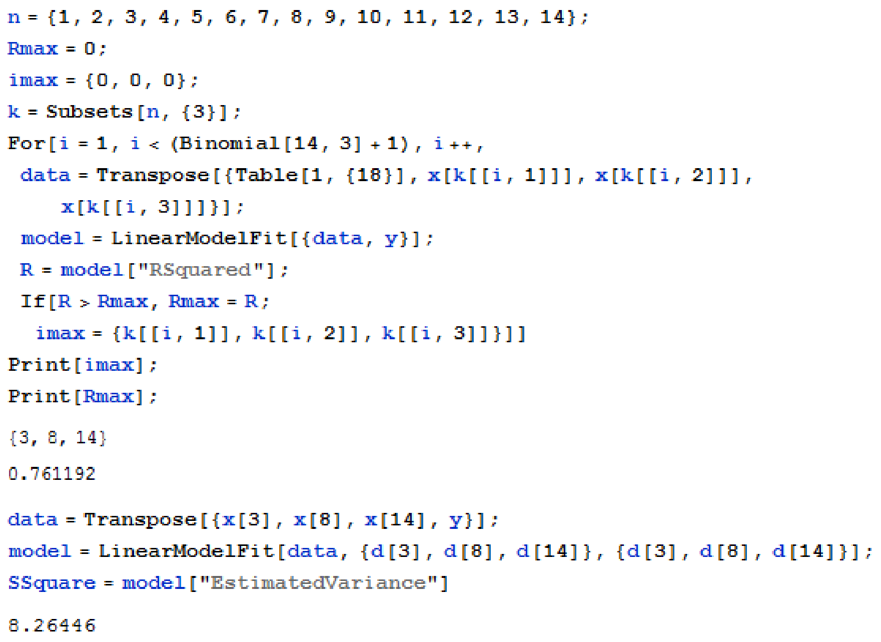
Step 2: Refer to figure 3.19. Consider two parameters and find the model with maximum R^2. The result is x[1] and x[14]. Then, find the corresponding S^2;



```
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
Rmax = 0;
imax = {0, 0};
k = Subsets[n, {2}];
For[i = 1, i < (Binomial[14, 2] + 1), i ++,
 data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]]}];
 model = LinearModelFit[{data, y}];
 R = model["RSquared"];
 If[R > Rmax, Rmax = R;
  imax = {k[[i, 1]], k[[i, 2]]}]]
Print[imax];
Print[Rmax];



{1, 14}
0.699955


data = Transpose[{x[1], x[14], y}];
model = LinearModelFit[data, {d[1], d[14]}, {d[1], d[14]}];
SSquare = model["EstimatedVariance"]

9.69145
```

Figure 3.19: Max R square selection Step 2

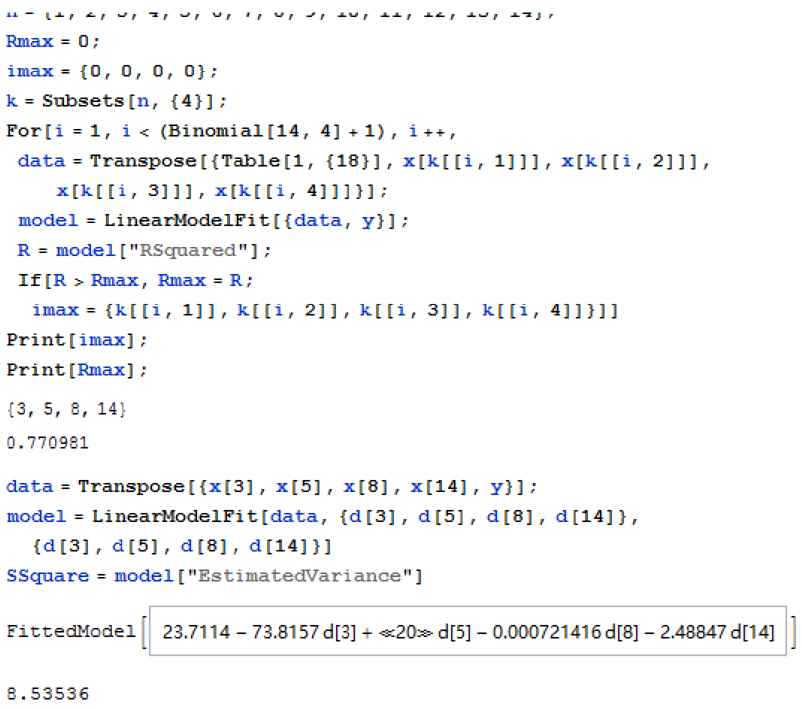
Step 3: Refer to figure 3.20. Consider three parameters and find the model with maximum R^2. The result is x[3] x[8] and x[14]. Then, find the corresponding S^2



```
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
Rmax = 0;
imax = {0, 0, 0};
k = Subsets[n, {3}];
For[i = 1, i < (Binomial[14, 3] + 1), i ++,
 data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]],
    x[k[[i, 3]]]}];
 model = LinearModelFit[{data, y}];
 R = model["RSquared"];
 If[R > Rmax, Rmax = R;
  imax = {k[[i, 1]], k[[i, 2]], k[[i, 3]]}]]
Print[imax];
Print[Rmax];

{3, 8, 14}

0.761192

data = Transpose[{x[3], x[8], x[14], y}];
model = LinearModelFit[data, {d[3], d[8], d[14]}, {d[3], d[8], d[14]}];
SSquare = model["EstimatedVariance"]

8.26446
```

Figure 3.20: Max R square selection Step 3

Step 4: Refer to figure 3.21. Consider four parameters and find the model with maximum R^2. The result is x[3] x[5] x[8] and x[14]. Then, find the corresponding S^2



```
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
Rmax = 0;
imax = {0, 0, 0, 0};
k = Subsets[n, {4}];
For[i = 1, i < (Binomial[14, 4] + 1), i ++,
 data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]],
    x[k[[i, 3]]], x[k[[i, 4]]]}];
 model = LinearModelFit[{data, y}];
 R = model["RSquared"];
 If[R > Rmax, Rmax = R;
  imax = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]]}]]
Print[imax];
Print[Rmax];

{3, 5, 8, 14}

0.770981

data = Transpose[{x[3], x[5], x[8], x[14], y}];
model = LinearModelFit[data, {d[3], d[5], d[8], d[14]},
  {d[3], d[5], d[8], d[14]}]
SSquare = model["EstimatedVariance"]

FittedModel[ 23.7114 - 73.8157 d[3] + ≪20≫ d[5] - 0.000721416 d[8] - 2.48847 d[14] ]

8.53536
```

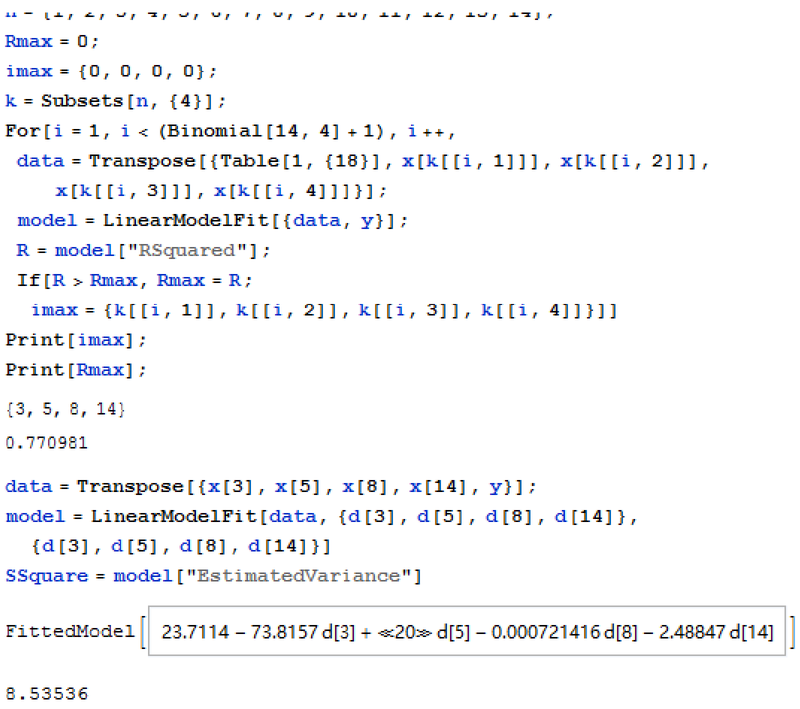
Figure 3.21: Max R square selection Step 4

We know that with the increasing of the number of parameters, the variance of the model S^2 will decrease at first, and then increase. We find that from step 1 to step 3, S^2 decreases and from step 3 to step 4, S^2 increases. Therefore, the model with 3 parameters has the minimum S^2. The final model we choose is: y=30.915798884248527-114.02441235366341x[3]-0.0006981351708096522x[8] -2.818626263091707x[14]

3.6 Mallows Cp

As the method of Max R^2, in this part, we also use "for" loops to test all the models and find the most suitable model. There are two ways to find the value of Cp. The first way is to find the minimum value of Cp, and we use Cpa to denote this situation. The second way is to find the Cp which is closest to the value of p(p is the number of parameters), and we use Cpb to denote this situation.

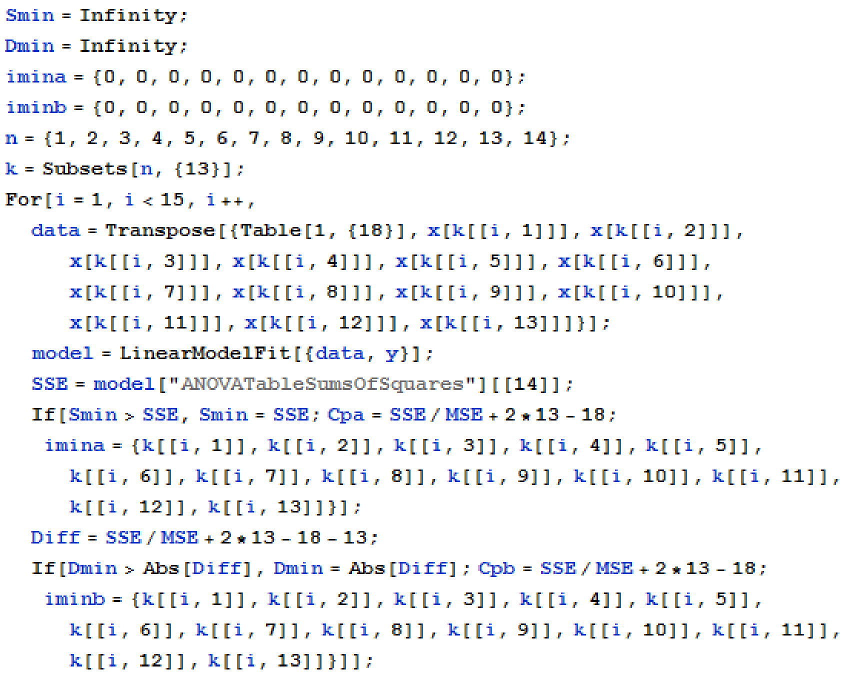
Step 1: Refer to figure 3.22. Consider 14 parameters



```
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
Rmax = 0;
imax = {0, 0, 0, 0};
k = Subsets[n, {4}];
For[i = 1, i < (Binomial[14, 4] + 1), i++,
 data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]],
    x[k[[i, 3]]], x[k[[i, 4]]]}];
 model = LinearModelFit[{data, y}];
 R = model["RSquared"];
 If[R > Rmax, Rmax = R;
  imax = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]]}]]
Print[imax];
Print[Rmax];

{3, 5, 8, 14}

0.770981

data = Transpose[{x[3], x[5], x[8], x[14], y}];
model = LinearModelFit[data, {d[3], d[5], d[8], d[14]},
  {d[3], d[5], d[8], d[14]}]
SSquare = model["EstimatedVariance"]

FittedModel[ 23.7114 − 73.8157 d[3] + ≪20≫ d[5] − 0.000721416 d[8] − 2.48847 d[14] ]

8.53536
```

Figure 3.22: Mallows Cp Step 1

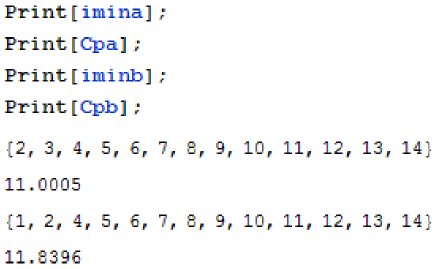
Step 2: Refer to figure 3.23. Consider 13 parameters



```
Smin = Infinity;
Dmin = Infinity;
imina = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
iminb = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {13}];
For[i = 1, i < 15, i ++,
  data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]],
     x[k[[i, 3]]], x[k[[i, 4]]], x[k[[i, 5]]], x[k[[i, 6]]],
     x[k[[i, 7]]], x[k[[i, 8]]], x[k[[i, 9]]], x[k[[i, 10]]],
     x[k[[i, 11]]], x[k[[i, 12]]], x[k[[i, 13]]]}];
  model = LinearModelFit[{data, y}];
  SSE = model["ANOVATableSumsOfSquares"][[14]];
  If[Smin > SSE, Smin = SSE; Cpa = SSE / MSE + 2 * 13 - 18;
   imina = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]], k[[i, 7]], k[[i, 8]], k[[i, 9]], k[[i, 10]], k[[i, 11]],
     k[[i, 12]], k[[i, 13]]}];
  Diff = SSE / MSE + 2 * 13 - 18 - 13;
  If[Dmin > Abs[Diff], Dmin = Abs[Diff]; Cpb = SSE / MSE + 2 * 13 - 18;
   iminb = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]], k[[i, 7]], k[[i, 8]], k[[i, 9]], k[[i, 10]], k[[i, 11]],
     k[[i, 12]], k[[i, 13]]}]];
```

Figure 3.23: Mallows Cp Step 2 (Procedure)

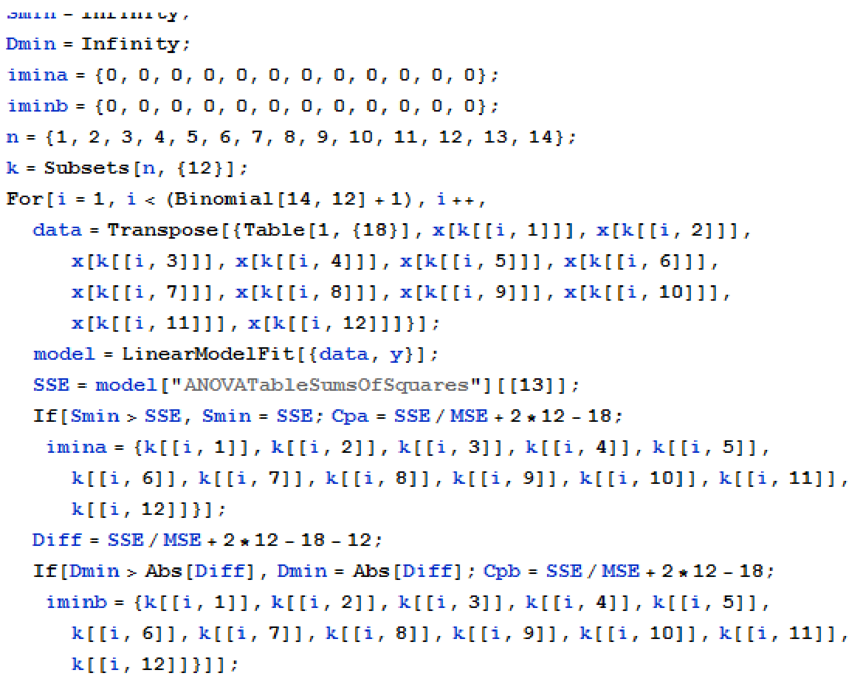
The result refers to figure 3.24.



```
Print[imina];
Print[Cpa];
Print[iminb];
Print[Cpb];

{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}

11.0005

{1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}

11.8396
```

Figure 3.24: Mallows Cp Step 2 (Result)

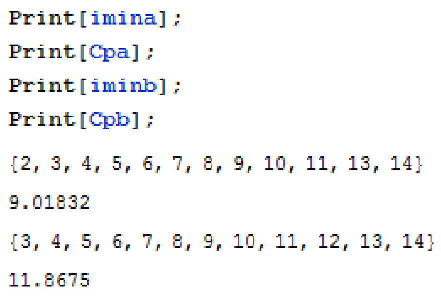
Step 3: Refer to figure 3.25. Consider 12 parameters



```
Smin = Infinity;
Dmin = Infinity;
imina = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
iminb = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {12}];
For[i = 1, i < (Binomial[14, 12] + 1), i++,
  data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]],
     x[k[[i, 3]]], x[k[[i, 4]]], x[k[[i, 5]]], x[k[[i, 6]]],
     x[k[[i, 7]]], x[k[[i, 8]]], x[k[[i, 9]]], x[k[[i, 10]]],
     x[k[[i, 11]]], x[k[[i, 12]]]}];
  model = LinearModelFit[{data, y}];
  SSE = model["ANOVATableSumsOfSquares"][[13]];
  If[Smin > SSE, Smin = SSE; Cpa = SSE / MSE + 2 * 12 - 18;
   imina = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]], k[[i, 7]], k[[i, 8]], k[[i, 9]], k[[i, 10]], k[[i, 11]],
     k[[i, 12]]}];
  Diff = SSE / MSE + 2 * 12 - 18 - 12;
  If[Dmin > Abs[Diff], Dmin = Abs[Diff]; Cpb = SSE / MSE + 2 * 12 - 18;
   iminb = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]], k[[i, 7]], k[[i, 8]], k[[i, 9]], k[[i, 10]], k[[i, 11]],
     k[[i, 12]]}]];
```

Figure 3.25: Mallows Cp Step 3 (Procedure)

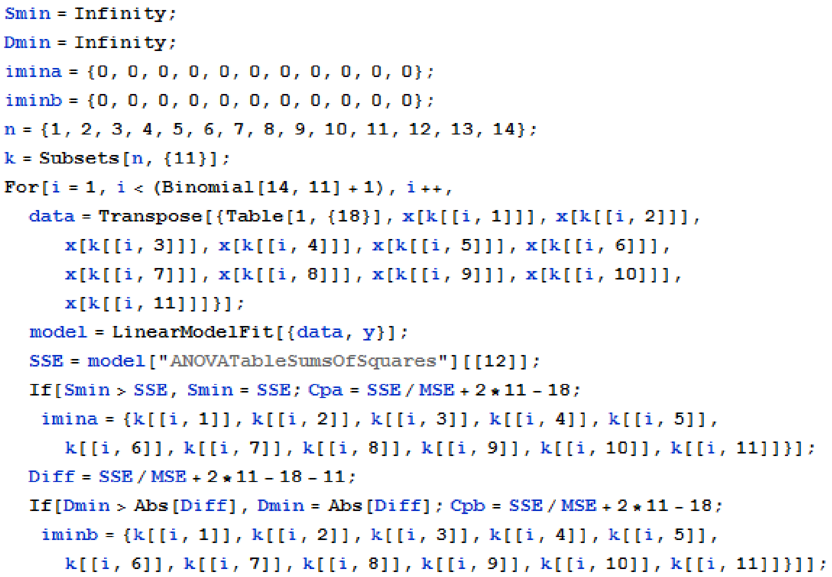
The result refers to figure 3.26.



```
Print[imina];
Print[Cpa];
Print[iminb];
Print[Cpb];

{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14}

9.01832

{3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}

11.8675
```

Figure 3.26: Mallows Cp Step 3 (Result)

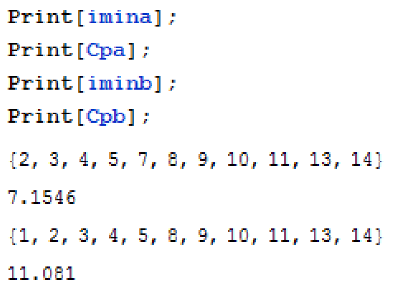
Step 4: Refer to figure 3.27. Consider 11 parameters



```
Smin = Infinity;
Dmin = Infinity;
imina = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
iminb = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {11}];
For[i = 1, i < (Binomial[14, 11] + 1), i ++,
  data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]],
     x[k[[i, 3]]], x[k[[i, 4]]], x[k[[i, 5]]], x[k[[i, 6]]],
     x[k[[i, 7]]], x[k[[i, 8]]], x[k[[i, 9]]], x[k[[i, 10]]],
     x[k[[i, 11]]]}];
  model = LinearModelFit[{data, y}];
  SSE = model["ANOVATableSumsOfSquares"][[12]];
  If[Smin > SSE, Smin = SSE; Cpa = SSE / MSE + 2 * 11 - 18;
   imina = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]], k[[i, 7]], k[[i, 8]], k[[i, 9]], k[[i, 10]], k[[i, 11]]}];
  Diff = SSE / MSE + 2 * 11 - 18 - 11;
  If[Dmin > Abs[Diff], Dmin = Abs[Diff]; Cpb = SSE / MSE + 2 * 11 - 18;
   iminb = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]], k[[i, 7]], k[[i, 8]], k[[i, 9]], k[[i, 10]], k[[i, 11]]}]];
```

Figure 3.27: Mallows Cp Step 4 (Procedure)

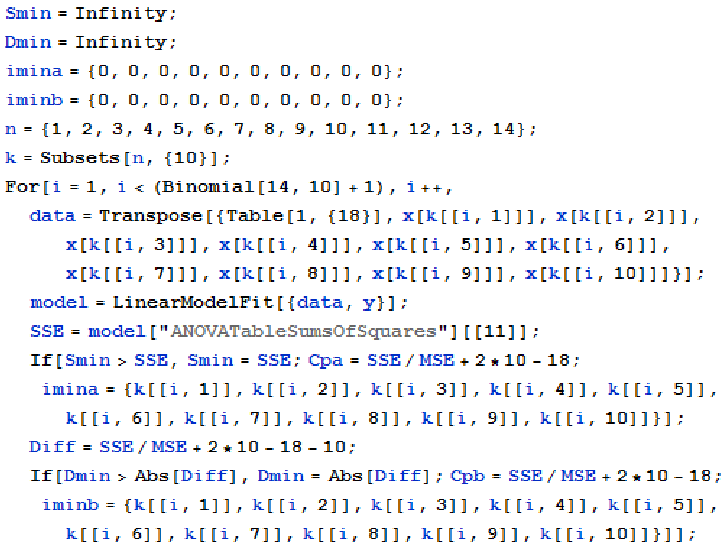
The result refers to figure 3.28.



```
Print[imina];
Print[Cpa];
Print[iminb];
Print[Cpb];

{2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14}

7.1546

{1, 2, 3, 4, 5, 8, 9, 10, 11, 13, 14}

11.081
```

Figure 3.28: Mallows Cp Step 4 (Result)

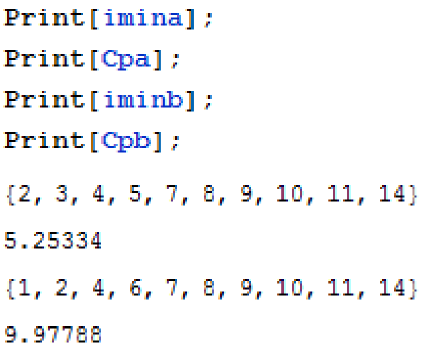
Step 5: Refer to figure 3.29. Consider 10 parameters



```
Smin = Infinity;
Dmin = Infinity;
imina = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
iminb = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {10}];
For[i = 1, i < (Binomial[14, 10] + 1), i ++,
  data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]],
     x[k[[i, 3]]], x[k[[i, 4]]], x[k[[i, 5]]], x[k[[i, 6]]],
     x[k[[i, 7]]], x[k[[i, 8]]], x[k[[i, 9]]], x[k[[i, 10]]]}];
  model = LinearModelFit[{data, y}];
  SSE = model["ANOVATableSumsOfSquares"][[11]];
  If[Smin > SSE, Smin = SSE; Cpa = SSE / MSE + 2 * 10 - 18;
   imina = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]], k[[i, 7]], k[[i, 8]], k[[i, 9]], k[[i, 10]]}];
  Diff = SSE / MSE + 2 * 10 - 18 - 10;
  If[Dmin > Abs[Diff], Dmin = Abs[Diff]; Cpb = SSE / MSE + 2 * 10 - 18;
   iminb = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]], k[[i, 7]], k[[i, 8]], k[[i, 9]], k[[i, 10]]}]];
```

Figure 3.29: Mallows Cp Step 5 (Procedure)

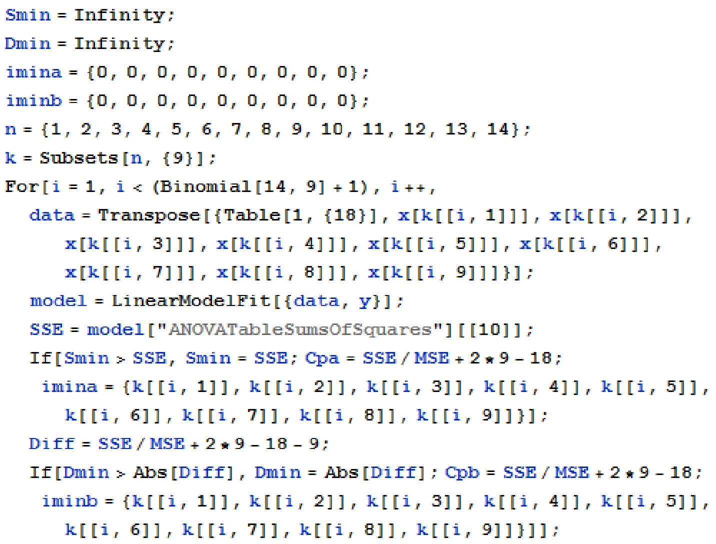
The result refers to figure 3.30.



```
Print[imina];
Print[Cpa];
Print[iminb];
Print[Cpb];

{2, 3, 4, 5, 7, 8, 9, 10, 11, 14}

5.25334

{1, 2, 4, 6, 7, 8, 9, 10, 11, 14}

9.97788
```

Figure 3.30: Mallows Cp Step 5 (Result)

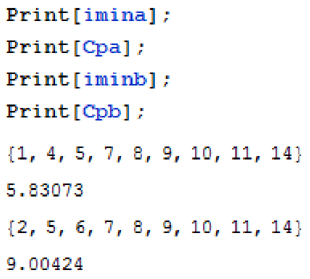
Step 6: Refer to figure 3.31. Consider 9 parameters



```
Smin = Infinity;
Dmin = Infinity;
imina = {0, 0, 0, 0, 0, 0, 0, 0, 0};
iminb = {0, 0, 0, 0, 0, 0, 0, 0, 0};
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {9}];
For[i = 1, i < (Binomial[14, 9] + 1), i ++,
  data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]],
     x[k[[i, 3]]], x[k[[i, 4]]], x[k[[i, 5]]], x[k[[i, 6]]],
     x[k[[i, 7]]], x[k[[i, 8]]], x[k[[i, 9]]]}];
  model = LinearModelFit[{data, y}];
  SSE = model["ANOVATableSumsOfSquares"][[10]];
  If[Smin > SSE, Smin = SSE; Cpa = SSE / MSE + 2 * 9 - 18;
   imina = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]], k[[i, 7]], k[[i, 8]], k[[i, 9]]}];
  Diff = SSE / MSE + 2 * 9 - 18 - 9;
  If[Dmin > Abs[Diff], Dmin = Abs[Diff]; Cpb = SSE / MSE + 2 * 9 - 18;
   iminb = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]], k[[i, 7]], k[[i, 8]], k[[i, 9]]}]];
```

Figure 3.31: Mallows Cp Step 6 (Procedure)

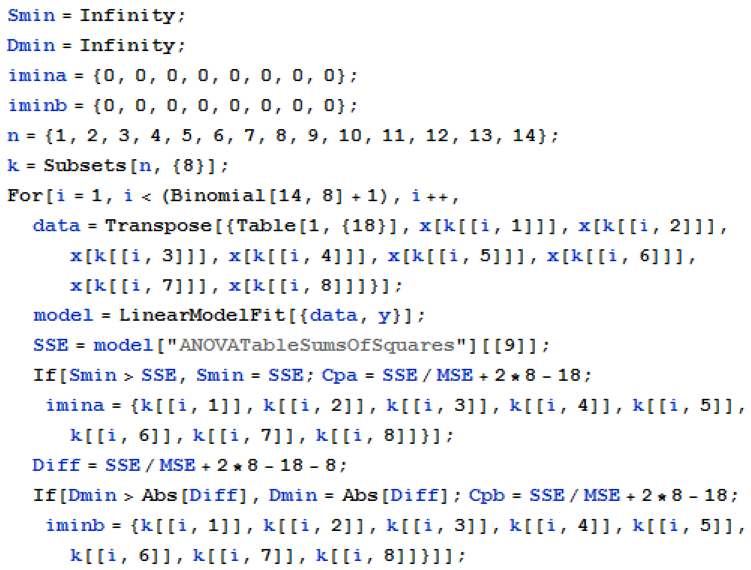
The result refers to figure 3.32.



```
Print[imina];
Print[Cpa];
Print[iminb];
Print[Cpb];

{1, 4, 5, 7, 8, 9, 10, 11, 14}

5.83073

{2, 5, 6, 7, 8, 9, 10, 11, 14}

9.00424
```

Figure 3.32: Mallows Cp Step 6 (Result)

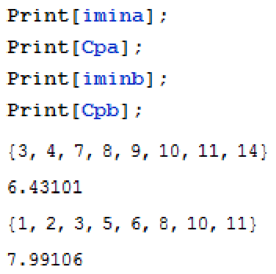
Step 7: Refer to figure 3.33. Consider 8 parameters



```
Smin = Infinity;
Dmin = Infinity;
imina = {0, 0, 0, 0, 0, 0, 0, 0};
iminb = {0, 0, 0, 0, 0, 0, 0, 0};
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {8}];
For[i = 1, i < (Binomial[14, 8] + 1), i ++,
  data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]],
     x[k[[i, 3]]], x[k[[i, 4]]], x[k[[i, 5]]], x[k[[i, 6]]],
     x[k[[i, 7]]], x[k[[i, 8]]]}];
  model = LinearModelFit[{data, y}];
  SSE = model["ANOVATableSumsOfSquares"][[9]];
  If[Smin > SSE, Smin = SSE; Cpa = SSE / MSE + 2 * 8 - 18;
   imina = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]], k[[i, 7]], k[[i, 8]]}];
  Diff = SSE / MSE + 2 * 8 - 18 - 8;
  If[Dmin > Abs[Diff], Dmin = Abs[Diff]; Cpb = SSE / MSE + 2 * 8 - 18;
   iminb = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]], k[[i, 7]], k[[i, 8]]}]];
```

Figure 3.33: Mallows Cp Step 7 (Procedure)

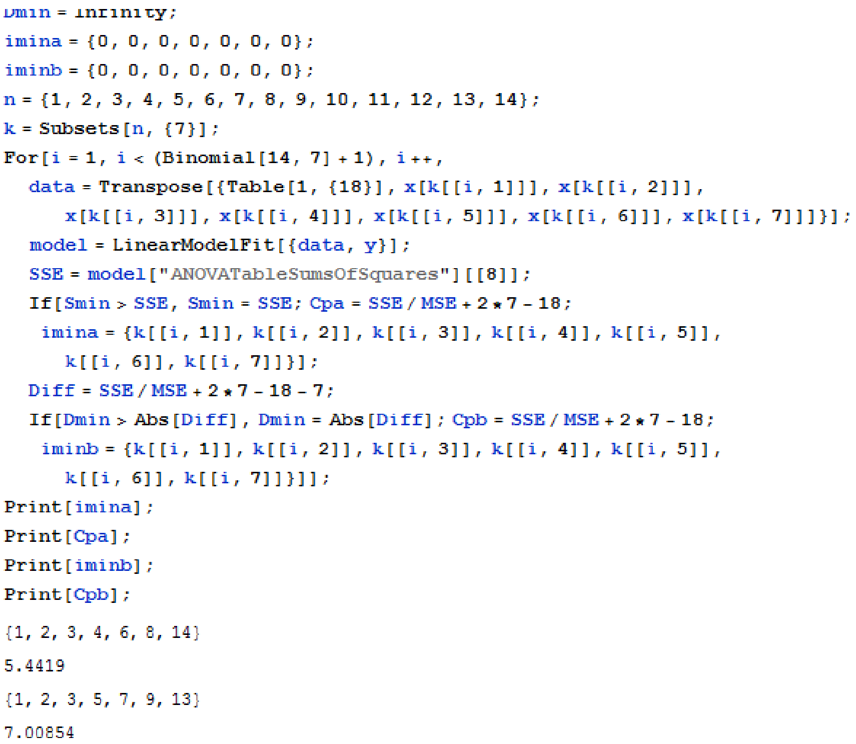
The result refers to figure 3.34.



```
Print[imina];
Print[Cpa];
Print[iminb];
Print[Cpb];

{3, 4, 7, 8, 9, 10, 11, 14}

6.43101

{1, 2, 3, 5, 6, 8, 10, 11}

7.99106
```

Figure 3.34: Mallows Cp Step 7 (Result)

Step 8: Refer to figure 3.35. Consider 7 parameters

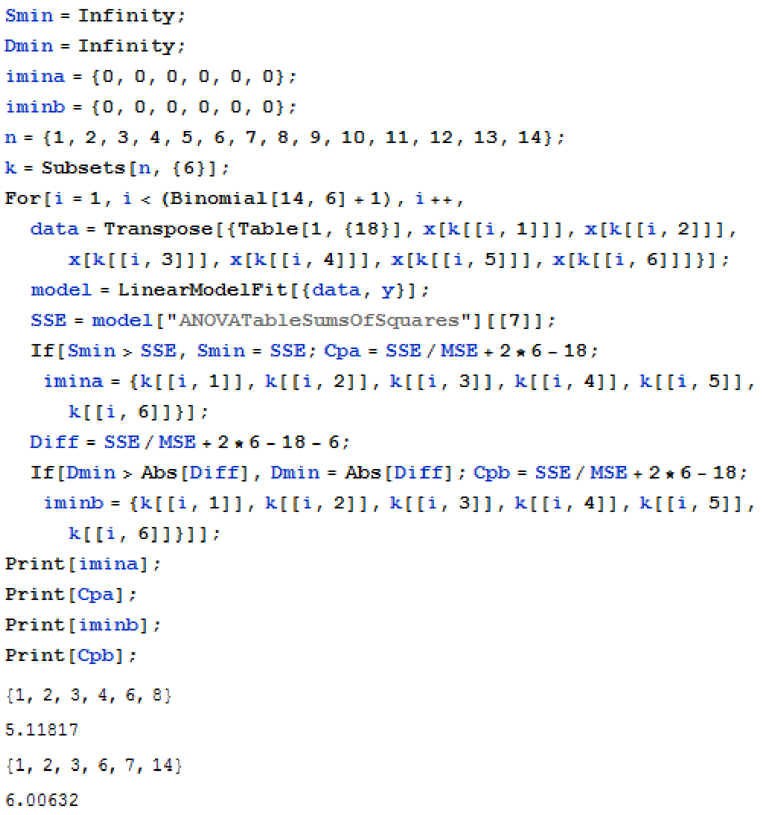


```
Dmin = Infinity;
imina = {0, 0, 0, 0, 0, 0, 0};
iminb = {0, 0, 0, 0, 0, 0, 0};
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {7}];
For[i = 1, i < (Binomial[14, 7] + 1), i++,
  data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]],
     x[k[[i, 3]]], x[k[[i, 4]]], x[k[[i, 5]]], x[k[[i, 6]]], x[k[[i, 7]]]}];
  model = LinearModelFit[{data, y}];
  SSE = model["ANOVATableSumsOfSquares"][[8]];
  If[Smin > SSE, Smin = SSE; Cpa = SSE / MSE + 2 * 7 - 18;
   imina = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]], k[[i, 7]]}];
  Diff = SSE / MSE + 2 * 7 - 18 - 7;
  If[Dmin > Abs[Diff], Dmin = Abs[Diff]; Cpb = SSE / MSE + 2 * 7 - 18;
   iminb = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]], k[[i, 7]]}]];
Print[imina];
Print[Cpa];
Print[iminb];
Print[Cpb];

{1, 2, 3, 4, 6, 8, 14}

5.4419

{1, 2, 3, 5, 7, 9, 13}

7.00854
```

Figure 3.35: Mallows Cp Step 8

The result refers to figure 3.35.

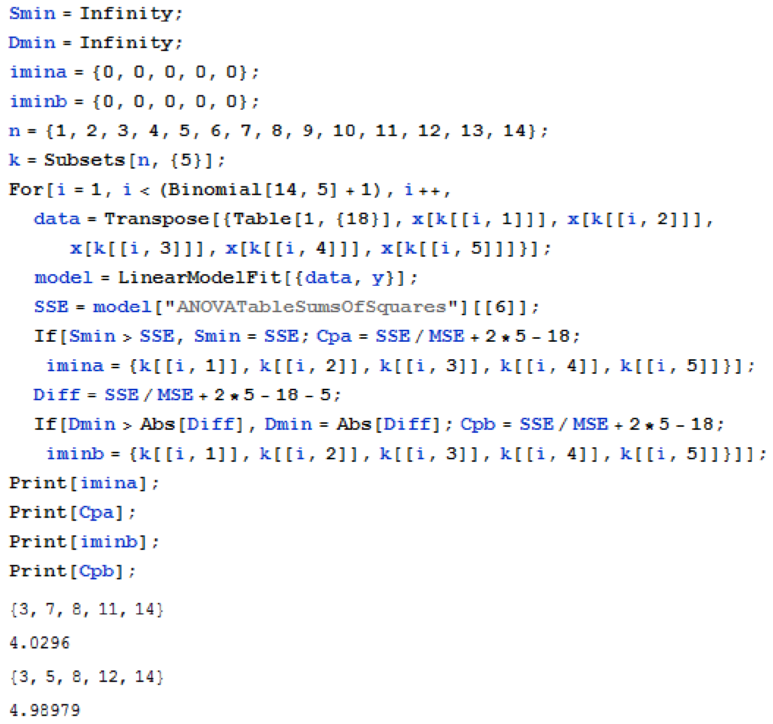
Step 9: Refer to figure 3.36. Consider 6 parameters



```
Smin = Infinity;
Dmin = Infinity;
imina = {0, 0, 0, 0, 0, 0};
iminb = {0, 0, 0, 0, 0, 0};
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {6}];
For[i = 1, i < (Binomial[14, 6] + 1), i ++,
  data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]],
     x[k[[i, 3]]], x[k[[i, 4]]], x[k[[i, 5]]], x[k[[i, 6]]]}];
  model = LinearModelFit[{data, y}];
  SSE = model["ANOVATableSumsOfSquares"][[7]];
  If[Smin > SSE, Smin = SSE; Cpa = SSE / MSE + 2 * 6 - 18;
   imina = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]]}];
  Diff = SSE / MSE + 2 * 6 - 18 - 6;
  If[Dmin > Abs[Diff], Dmin = Abs[Diff]; Cpb = SSE / MSE + 2 * 6 - 18;
   iminb = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]],
     k[[i, 6]]}]];
Print[imina];
Print[Cpa];
Print[iminb];
Print[Cpb];

{1, 2, 3, 4, 6, 8}

5.11817

{1, 2, 3, 6, 7, 14}

6.00632
```

Figure 3.36: Mallows Cp Step 9

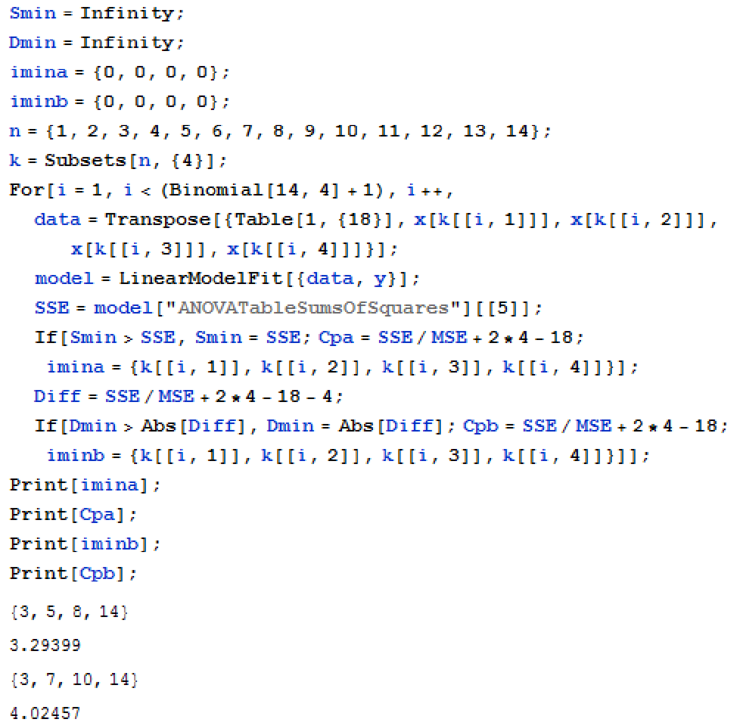
The result refers to figure 3.36.
Step 10: Refer to figure 3.37. Consider 5 parameters



```
Smin = Infinity;
Dmin = Infinity;
imina = {0, 0, 0, 0, 0};
iminb = {0, 0, 0, 0, 0};
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {5}];
For[i = 1, i < (Binomial[14, 5] + 1), i ++,
  data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]],
     x[k[[i, 3]]], x[k[[i, 4]]], x[k[[i, 5]]]}];
  model = LinearModelFit[{data, y}];
  SSE = model["ANOVATableSumsOfSquares"][[6]];
  If[Smin > SSE, Smin = SSE; Cpa = SSE / MSE + 2 * 5 - 18;
   imina = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]]}];
  Diff = SSE / MSE + 2 * 5 - 18 - 5;
  If[Dmin > Abs[Diff], Dmin = Abs[Diff]; Cpb = SSE / MSE + 2 * 5 - 18;
   iminb = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]], k[[i, 5]]}]];
Print[imina];
Print[Cpa];
Print[iminb];
Print[Cpb];
{3, 7, 8, 11, 14}
4.0296
{3, 5, 8, 12, 14}
4.98979
```

Figure 3.37: Mallows Cp Step 10

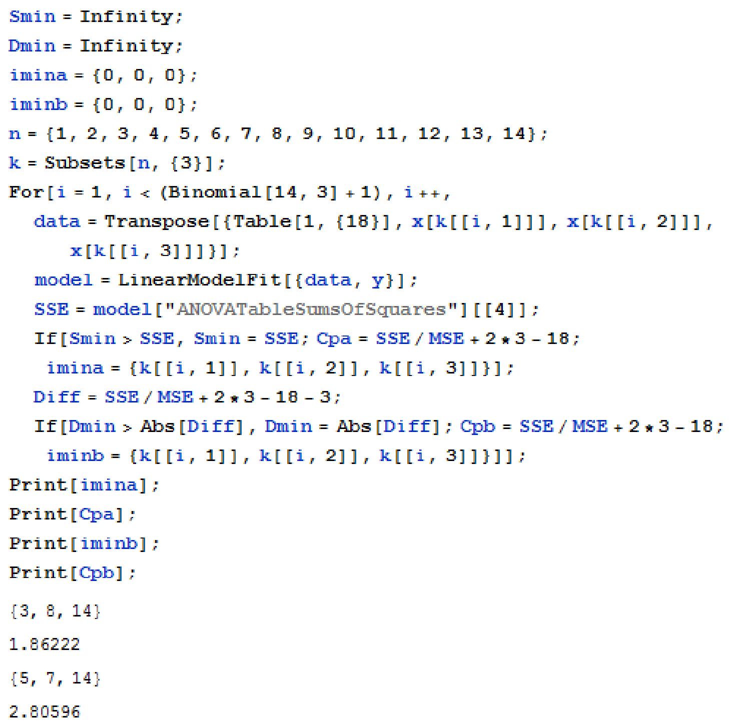
The result refers to figure 3.37.
Step 11: Refer to figure 3.38. Consider 4 parameters



```
Smin = Infinity;
Dmin = Infinity;
imina = {0, 0, 0, 0};
iminb = {0, 0, 0, 0};
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {4}];
For[i = 1, i < (Binomial[14, 4] + 1), i++,
  data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]],
     x[k[[i, 3]]], x[k[[i, 4]]]}];
  model = LinearModelFit[{data, y}];
  SSE = model["ANOVATableSumsOfSquares"][[5]];
  If[Smin > SSE, Smin = SSE; Cpa = SSE / MSE + 2 * 4 - 18;
   imina = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]]}];
  Diff = SSE / MSE + 2 * 4 - 18 - 4;
  If[Dmin > Abs[Diff], Dmin = Abs[Diff]; Cpb = SSE / MSE + 2 * 4 - 18;
   iminb = {k[[i, 1]], k[[i, 2]], k[[i, 3]], k[[i, 4]]}]];
Print[imina];
Print[Cpa];
Print[iminb];
Print[Cpb];

{3, 5, 8, 14}

3.29399

{3, 7, 10, 14}

4.02457
```

Figure 3.38: Mallows Cp Step 11

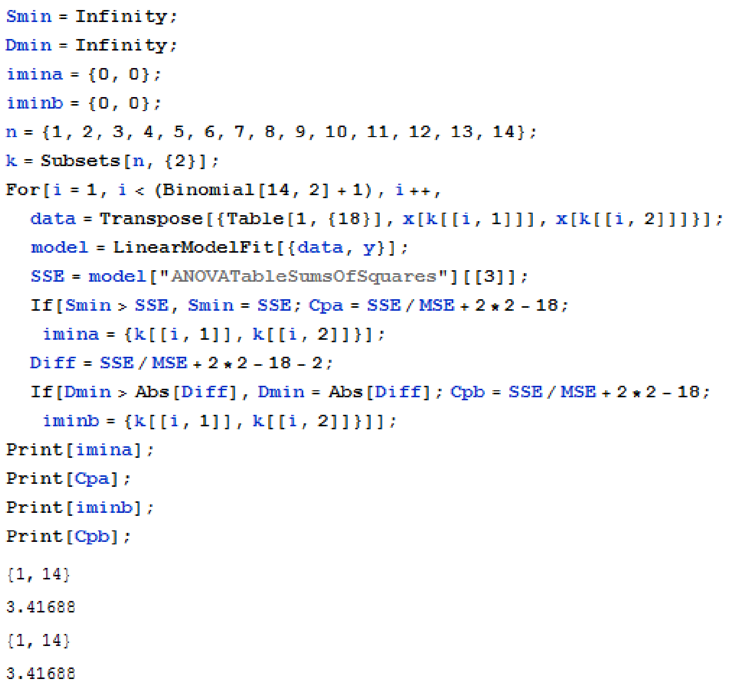
The result refers to figure 3.38.
Step 12: Refer to figure 3.39. Consider 3 parameters



```
Smin = Infinity;
Dmin = Infinity;
imina = {0, 0, 0};
iminb = {0, 0, 0};
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {3}];
For[i = 1, i < (Binomial[14, 3] + 1), i ++,
  data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]],
     x[k[[i, 3]]]}];
  model = LinearModelFit[{data, y}];
  SSE = model["ANOVATableSumsOfSquares"][[4]];
  If[Smin > SSE, Smin = SSE; Cpa = SSE / MSE + 2 * 3 - 18;
   imina = {k[[i, 1]], k[[i, 2]], k[[i, 3]]}];
  Diff = SSE / MSE + 2 * 3 - 18 - 3;
  If[Dmin > Abs[Diff], Dmin = Abs[Diff]; Cpb = SSE / MSE + 2 * 3 - 18;
   iminb = {k[[i, 1]], k[[i, 2]], k[[i, 3]]}]];
Print[imina];
Print[Cpa];
Print[iminb];
Print[Cpb];

{3, 8, 14}

1.86222

{5, 7, 14}

2.80596
```

Figure 3.39: Mallows Cp Step 12

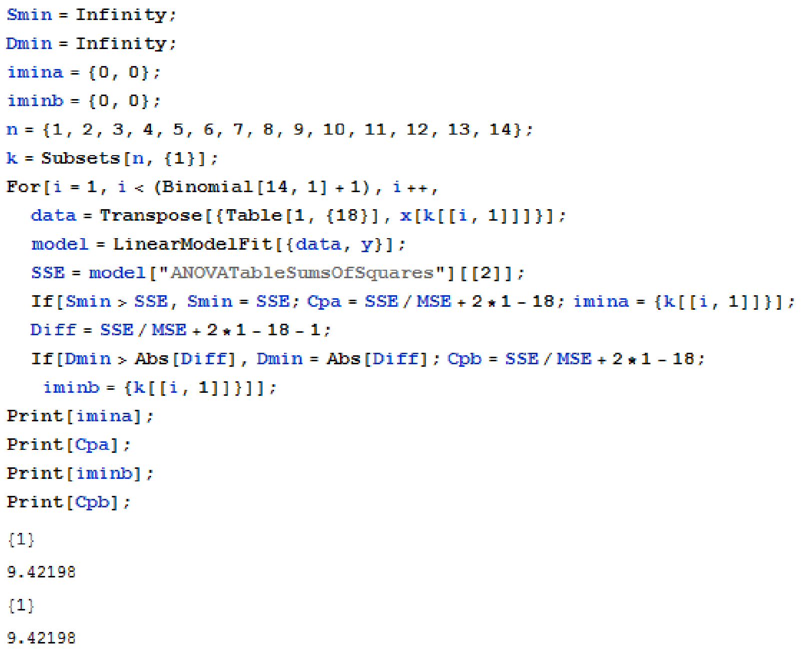
The result refers to figure 3.39.
Step 13: Refer to figure 3.40. Consider 2 parameters



```
Smin = Infinity;
Dmin = Infinity;
imina = {0, 0};
iminb = {0, 0};
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {2}];
For[i = 1, i < (Binomial[14, 2] + 1), i ++,
  data = Transpose[{Table[1, {18}], x[k[[i, 1]]], x[k[[i, 2]]]}];
  model = LinearModelFit[{data, y}];
  SSE = model["ANOVATableSumsOfSquares"][[3]];
  If[Smin > SSE, Smin = SSE; Cpa = SSE / MSE + 2 * 2 - 18;
   imina = {k[[i, 1]], k[[i, 2]]}];
  Diff = SSE / MSE + 2 * 2 - 18 - 2;
  If[Dmin > Abs[Diff], Dmin = Abs[Diff]; Cpb = SSE / MSE + 2 * 2 - 18;
   iminb = {k[[i, 1]], k[[i, 2]]}]];
Print[imina];
Print[Cpa];
Print[iminb];
Print[Cpb];

{1, 14}

3.41688

{1, 14}

3.41688
```

Figure 3.40: Mallows Cp Step 13

The result refers to figure 3.40.
Step 14: Refer to figure 3.41. Consider 1 parameter



```
Smin = Infinity;
Dmin = Infinity;
imina = {0, 0};
iminb = {0, 0};
n = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
k = Subsets[n, {1}];
For[i = 1, i < (Binomial[14, 1] + 1), i ++,
  data = Transpose[{Table[1, {18}], x[k[[i, 1]]]}];
  model = LinearModelFit[{data, y}];
  SSE = model["ANOVATableSumsOfSquares"][[2]];
  If[Smin > SSE, Smin = SSE; Cpa = SSE / MSE + 2 * 1 - 18; imina = {k[[i, 1]]}];
  Diff = SSE / MSE + 2 * 1 - 18 - 1;
  If[Dmin > Abs[Diff], Dmin = Abs[Diff]; Cpb = SSE / MSE + 2 * 1 - 18;
   iminb = {k[[i, 1]]}]];
Print[imina];
Print[Cpa];
Print[iminb];
Print[Cpb];
{1}
9.42198
{1}
9.42198
```

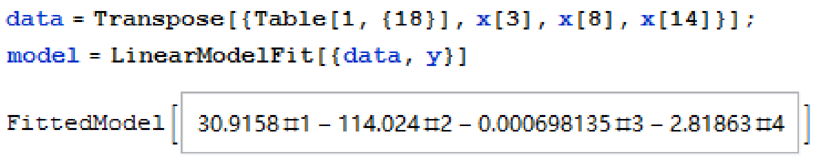
Figure 3.41: Mallows Cp Step 14

The result refers to figure 3.41.

At last, we compare the results of all the steps and find the most suitable Cpa and Cpb.

For Cpa: y=30.915798884248527-114.02441235366341x[3] -0.0006981351708096522x[8]-2.818626263091707x[14]

For Cpb: y=788.4777686468177+0.02598776710981143x[2]+ 0.8907132969606881x[5] +25.99881418102408x[6]+ 0.011485960175095x[7]-0.012717844039662746x[8] -1593.5060142160467x[9]-0.11874516418809196x[10] +0.22561954081731345x[11]-2.038185244815709x[14]



```
data = Transpose[{Table[1, {18}], x[3], x[8], x[14]}];
model = LinearModelFit[{data, y}]

FittedModel[ 30.9158 #1 - 114.024 #2 - 0.000698135 #3 - 2.81863 #4 ]
```

Figure 3.42: Mallows Cp Final Result

The result refers to figure 3.42.

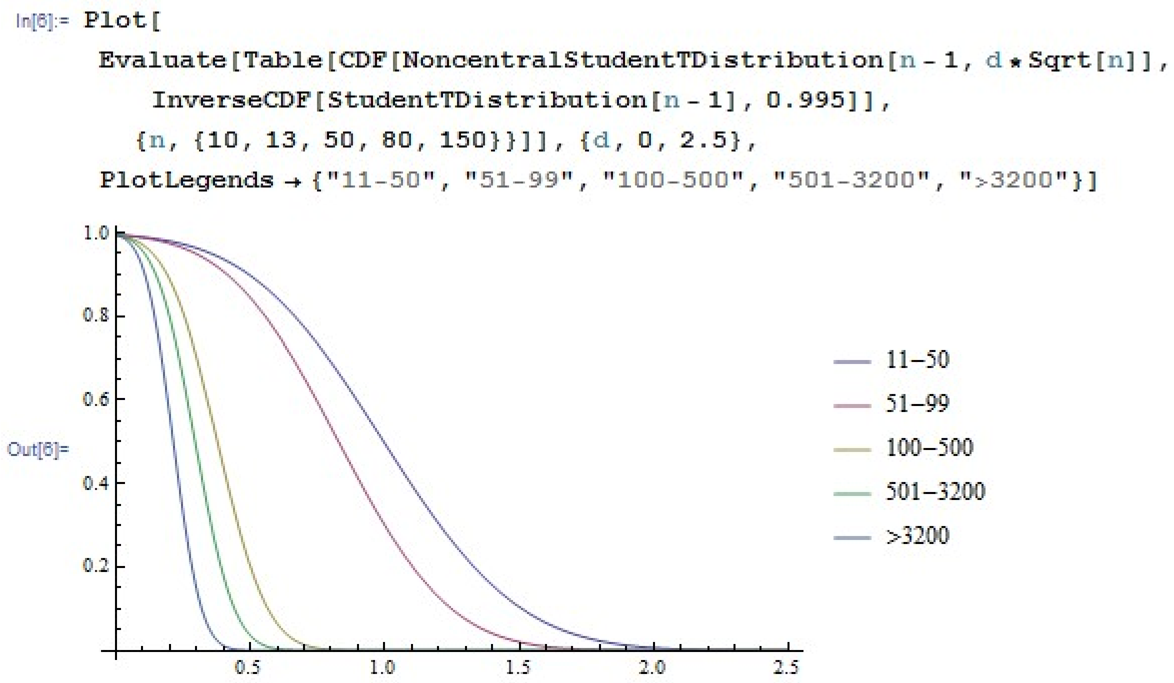
3.7 Conclusion

Comparing all the models we get through different methods, we find that every model has it advantages and disadvantages. For the forward and stepwise model, it needs the least steps, but has big R^2. Besides, it only takes 2 parameters into account, so that the model can't evaluate a team properly. For the backward model, it is better than the forward and stepwise model, since it has smaller R^2 and considers more parameters. For Max R^2 and Mallow' Cp, we think they are much better because they test all the possible models and find the best one, so they are more accurate. Mallow's Cp can take more parameters into account, so that it is better than Max R^2. Therefore, I think the model we get through Mallow's Cp method is best. However, we think all the models have a serious problem, which is that the selection of y is not proper. We try to find a linear relationship between the standings and the performances. Standings can only be "1,2,3,4,5,6$\cdots$", which makes it very difficult for the model to fit well. So, we think replace standings with accumulate points will be better.

Appendix A

Mathematical Codes

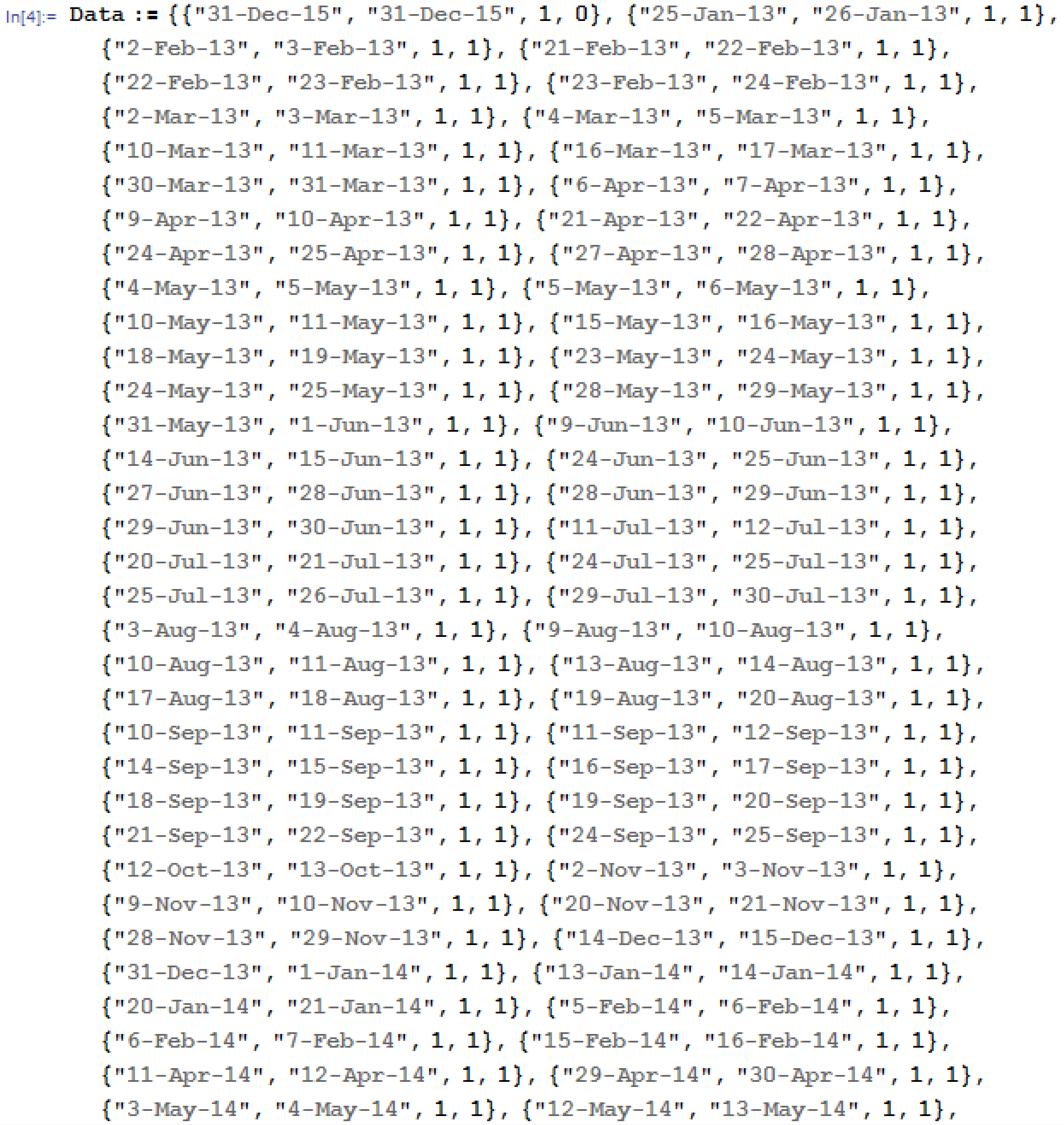
A.1 Analysis of Package Contents



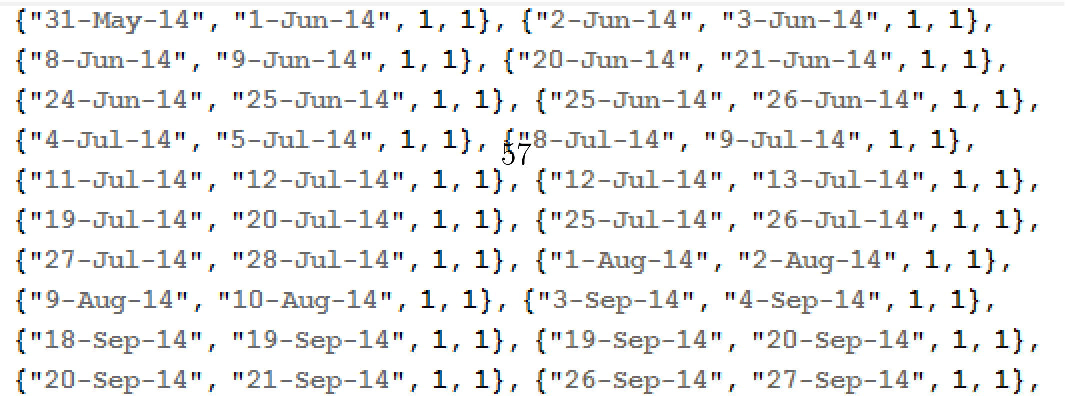
```
In[6]:= Plot[
  Evaluate[Table[CDF[NoncentralStudentTDistribution[n - 1, d * Sqrt[n]],
     InverseCDF[StudentTDistribution[n - 1], 0.995]],
    {n, {10, 13, 50, 80, 150}}]], {d, 0, 2.5},
  PlotLegends → {"11-50", "51-99", "100-500", "501-3200", ">3200"}]
```

A.2 Mass Shootings in the United States

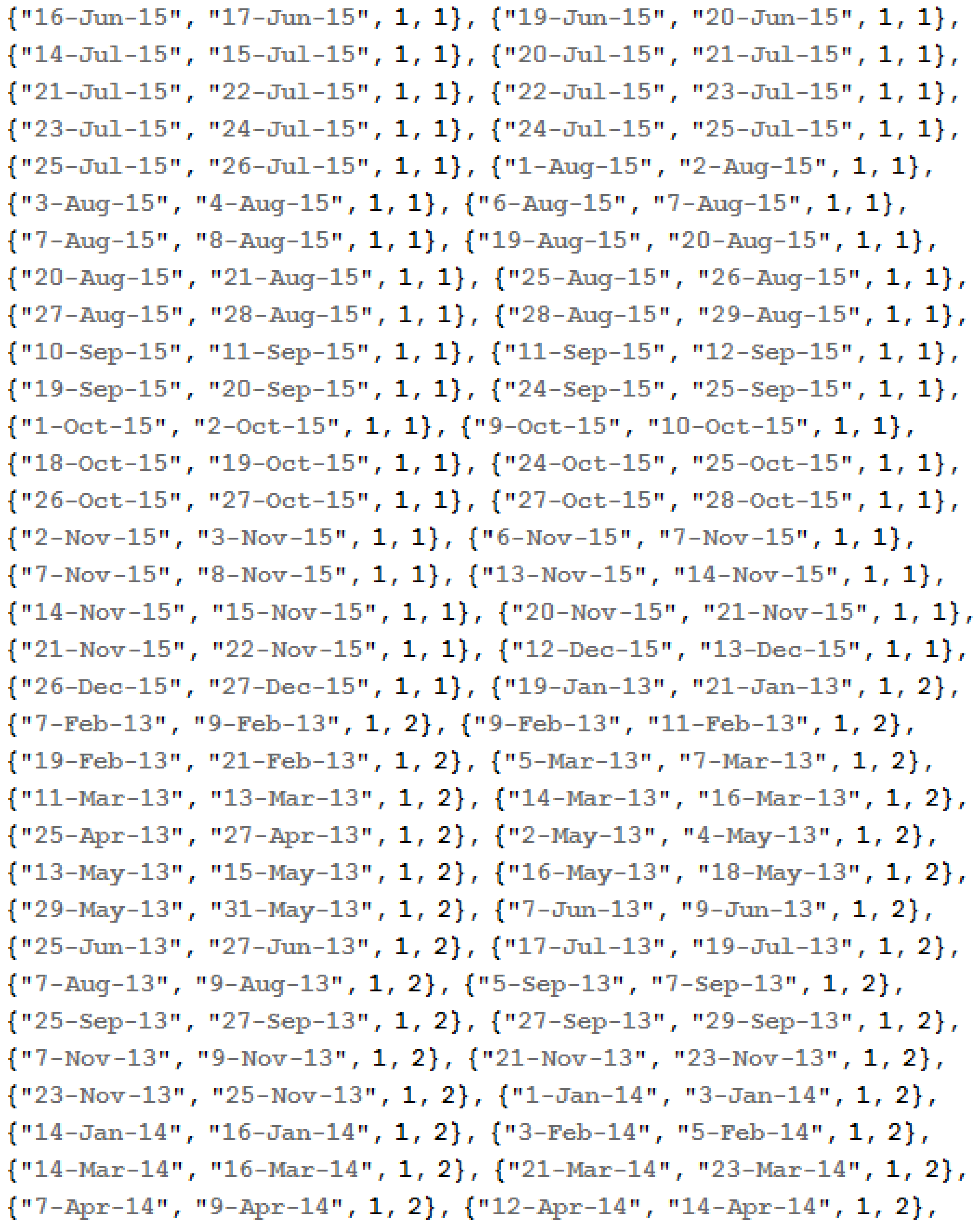
A.2.1 Daily Number of Mass Shootings



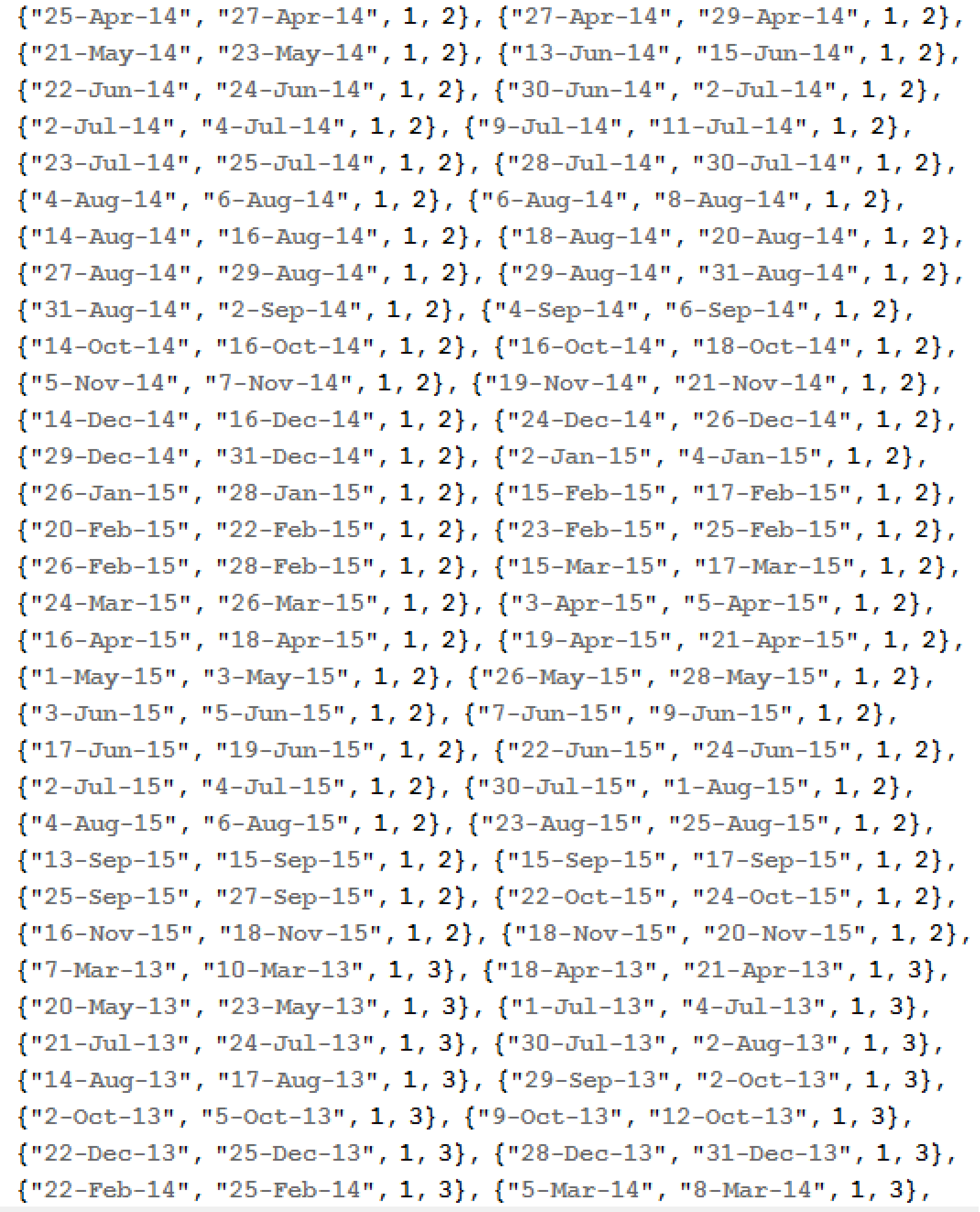
```
In[4]:= Data := {{"31-Dec-15", "31-Dec-15", 1, 0}, {"25-Jan-13", "26-Jan-13", 1, 1},
      {"2-Feb-13", "3-Feb-13", 1, 1}, {"21-Feb-13", "22-Feb-13", 1, 1},
      {"22-Feb-13", "23-Feb-13", 1, 1}, {"23-Feb-13", "24-Feb-13", 1, 1},
      {"2-Mar-13", "3-Mar-13", 1, 1}, {"4-Mar-13", "5-Mar-13", 1, 1},
      {"10-Mar-13", "11-Mar-13", 1, 1}, {"16-Mar-13", "17-Mar-13", 1, 1},
      {"30-Mar-13", "31-Mar-13", 1, 1}, {"6-Apr-13", "7-Apr-13", 1, 1},
      {"9-Apr-13", "10-Apr-13", 1, 1}, {"21-Apr-13", "22-Apr-13", 1, 1},
      {"24-Apr-13", "25-Apr-13", 1, 1}, {"27-Apr-13", "28-Apr-13", 1, 1},
      {"4-May-13", "5-May-13", 1, 1}, {"5-May-13", "6-May-13", 1, 1},
      {"10-May-13", "11-May-13", 1, 1}, {"15-May-13", "16-May-13", 1, 1},
      {"18-May-13", "19-May-13", 1, 1}, {"23-May-13", "24-May-13", 1, 1},
      {"24-May-13", "25-May-13", 1, 1}, {"28-May-13", "29-May-13", 1, 1},
      {"31-May-13", "1-Jun-13", 1, 1}, {"9-Jun-13", "10-Jun-13", 1, 1},
      {"14-Jun-13", "15-Jun-13", 1, 1}, {"24-Jun-13", "25-Jun-13", 1, 1},
      {"27-Jun-13", "28-Jun-13", 1, 1}, {"28-Jun-13", "29-Jun-13", 1, 1},
      {"29-Jun-13", "30-Jun-13", 1, 1}, {"11-Jul-13", "12-Jul-13", 1, 1},
      {"20-Jul-13", "21-Jul-13", 1, 1}, {"24-Jul-13", "25-Jul-13", 1, 1},
      {"25-Jul-13", "26-Jul-13", 1, 1}, {"29-Jul-13", "30-Jul-13", 1, 1},
      {"3-Aug-13", "4-Aug-13", 1, 1}, {"9-Aug-13", "10-Aug-13", 1, 1},
      {"10-Aug-13", "11-Aug-13", 1, 1}, {"13-Aug-13", "14-Aug-13", 1, 1},
      {"17-Aug-13", "18-Aug-13", 1, 1}, {"19-Aug-13", "20-Aug-13", 1, 1},
      {"10-Sep-13", "11-Sep-13", 1, 1}, {"11-Sep-13", "12-Sep-13", 1, 1},
      {"14-Sep-13", "15-Sep-13", 1, 1}, {"16-Sep-13", "17-Sep-13", 1, 1},
      {"18-Sep-13", "19-Sep-13", 1, 1}, {"19-Sep-13", "20-Sep-13", 1, 1},
      {"21-Sep-13", "22-Sep-13", 1, 1}, {"24-Sep-13", "25-Sep-13", 1, 1},
      {"12-Oct-13", "13-Oct-13", 1, 1}, {"2-Nov-13", "3-Nov-13", 1, 1},
      {"9-Nov-13", "10-Nov-13", 1, 1}, {"20-Nov-13", "21-Nov-13", 1, 1},
      {"28-Nov-13", "29-Nov-13", 1, 1}, {"14-Dec-13", "15-Dec-13", 1, 1},
      {"31-Dec-13", "1-Jan-14", 1, 1}, {"13-Jan-14", "14-Jan-14", 1, 1},
      {"20-Jan-14", "21-Jan-14", 1, 1}, {"5-Feb-14", "6-Feb-14", 1, 1},
      {"6-Feb-14", "7-Feb-14", 1, 1}, {"15-Feb-14", "16-Feb-14", 1, 1},
      {"11-Apr-14", "12-Apr-14", 1, 1}, {"29-Apr-14", "30-Apr-14", 1, 1},
      {"3-May-14", "4-May-14", 1, 1}, {"12-May-14", "13-May-14", 1, 1},
```



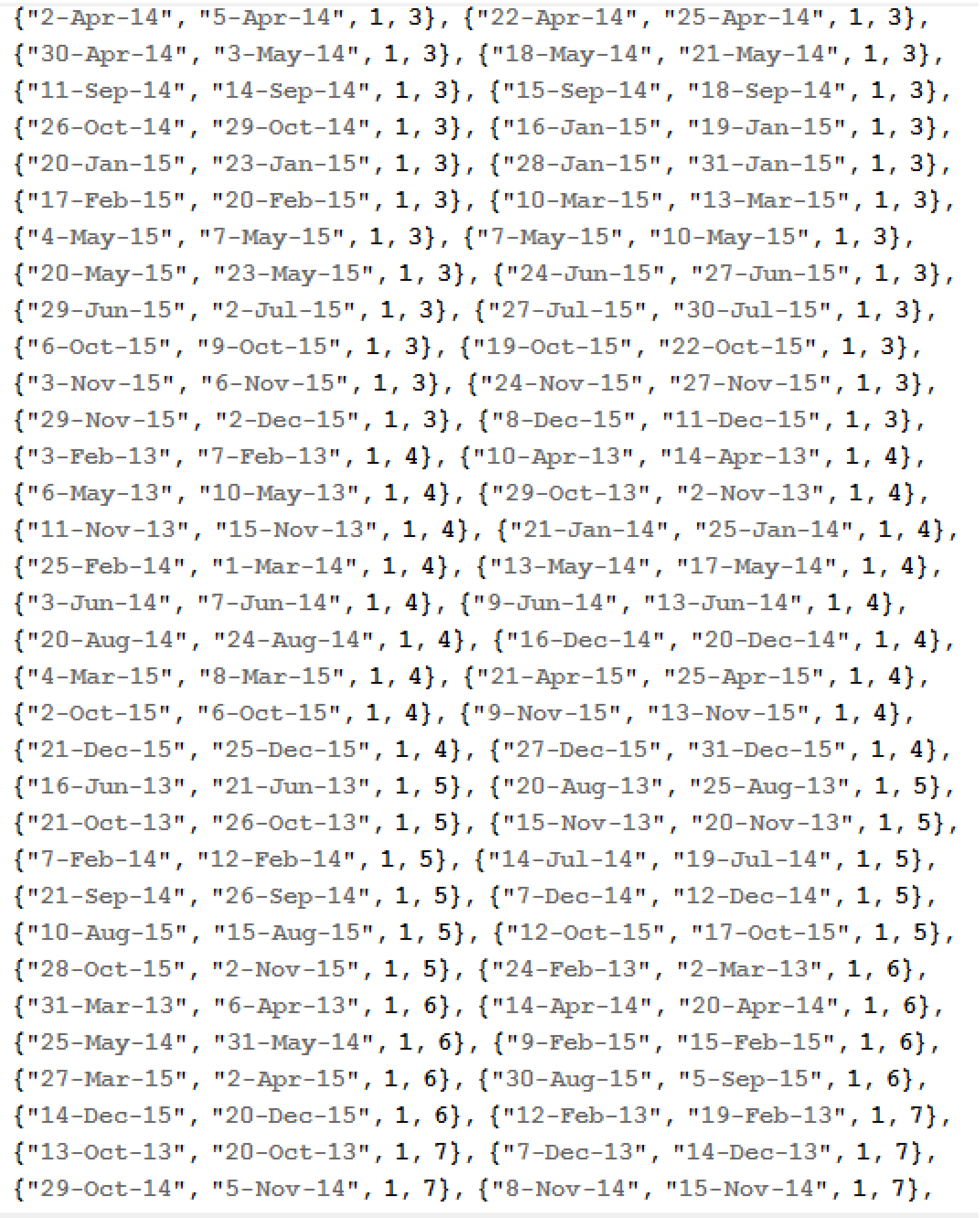
```
{"31-May-14", "1-Jun-14", 1, 1}, {"2-Jun-14", "3-Jun-14", 1, 1},
{"8-Jun-14", "9-Jun-14", 1, 1}, {"20-Jun-14", "21-Jun-14", 1, 1},
{"24-Jun-14", "25-Jun-14", 1, 1}, {"25-Jun-14", "26-Jun-14", 1, 1},
{"4-Jul-14", "5-Jul-14", 1, 1}, {"8-Jul-14", "9-Jul-14", 1, 1},
{"11-Jul-14", "12-Jul-14", 1, 1}, {"12-Jul-14", "13-Jul-14", 1, 1},
{"19-Jul-14", "20-Jul-14", 1, 1}, {"25-Jul-14", "26-Jul-14", 1, 1},
{"27-Jul-14", "28-Jul-14", 1, 1}, {"1-Aug-14", "2-Aug-14", 1, 1},
{"9-Aug-14", "10-Aug-14", 1, 1}, {"3-Sep-14", "4-Sep-14", 1, 1},
{"18-Sep-14", "19-Sep-14", 1, 1}, {"19-Sep-14", "20-Sep-14", 1, 1},
{"20-Sep-14", "21-Sep-14", 1, 1}, {"26-Sep-14", "27-Sep-14", 1, 1},
```



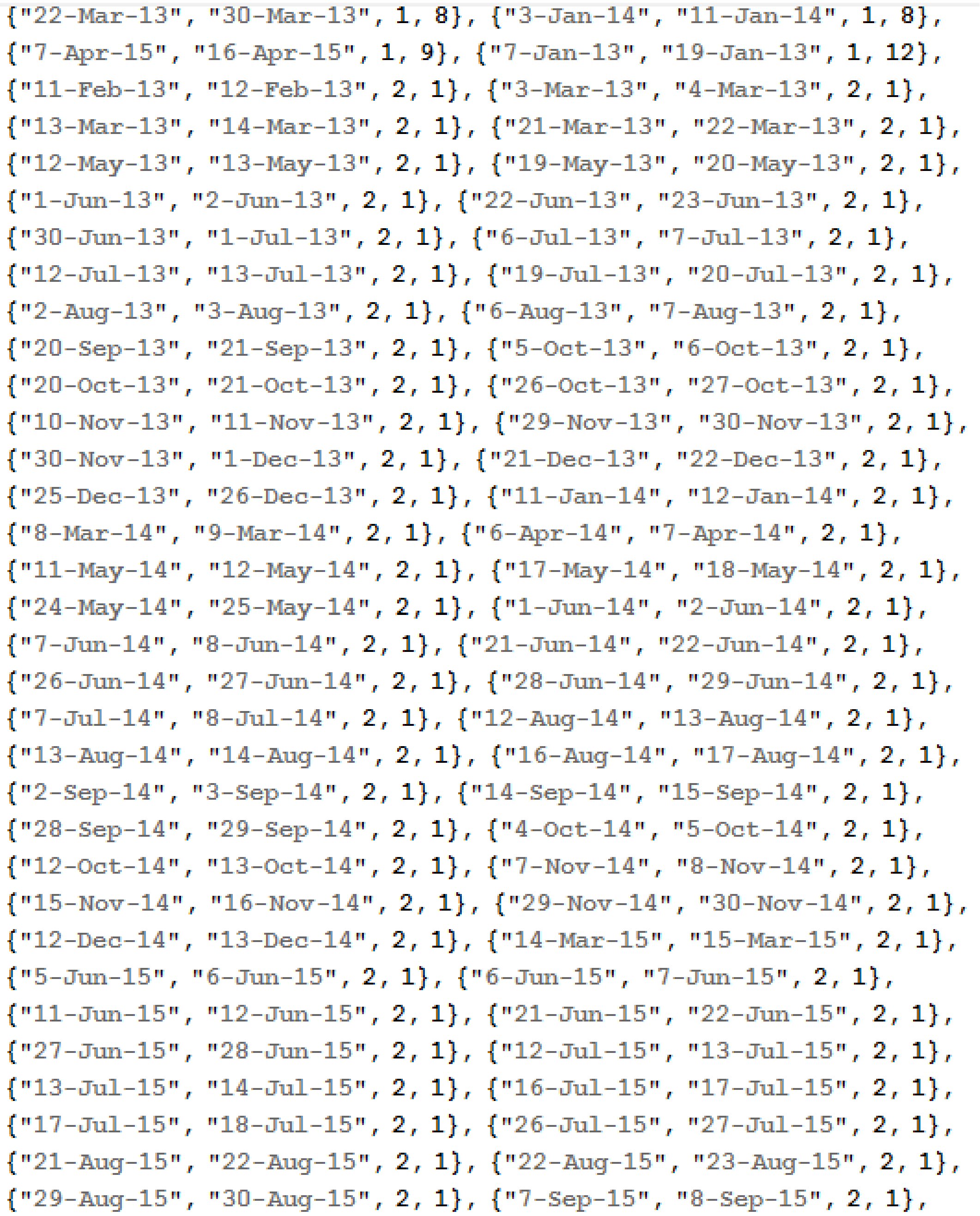
```
{"16-Jun-15", "17-Jun-15", 1, 1}, {"19-Jun-15", "20-Jun-15", 1, 1},
{"14-Jul-15", "15-Jul-15", 1, 1}, {"20-Jul-15", "21-Jul-15", 1, 1},
{"21-Jul-15", "22-Jul-15", 1, 1}, {"22-Jul-15", "23-Jul-15", 1, 1},
{"23-Jul-15", "24-Jul-15", 1, 1}, {"24-Jul-15", "25-Jul-15", 1, 1},
{"25-Jul-15", "26-Jul-15", 1, 1}, {"1-Aug-15", "2-Aug-15", 1, 1},
{"3-Aug-15", "4-Aug-15", 1, 1}, {"6-Aug-15", "7-Aug-15", 1, 1},
{"7-Aug-15", "8-Aug-15", 1, 1}, {"19-Aug-15", "20-Aug-15", 1, 1},
{"20-Aug-15", "21-Aug-15", 1, 1}, {"25-Aug-15", "26-Aug-15", 1, 1},
{"27-Aug-15", "28-Aug-15", 1, 1}, {"28-Aug-15", "29-Aug-15", 1, 1},
{"10-Sep-15", "11-Sep-15", 1, 1}, {"11-Sep-15", "12-Sep-15", 1, 1},
{"19-Sep-15", "20-Sep-15", 1, 1}, {"24-Sep-15", "25-Sep-15", 1, 1},
{"1-Oct-15", "2-Oct-15", 1, 1}, {"9-Oct-15", "10-Oct-15", 1, 1},
{"18-Oct-15", "19-Oct-15", 1, 1}, {"24-Oct-15", "25-Oct-15", 1, 1},
{"26-Oct-15", "27-Oct-15", 1, 1}, {"27-Oct-15", "28-Oct-15", 1, 1},
{"2-Nov-15", "3-Nov-15", 1, 1}, {"6-Nov-15", "7-Nov-15", 1, 1},
{"7-Nov-15", "8-Nov-15", 1, 1}, {"13-Nov-15", "14-Nov-15", 1, 1},
{"14-Nov-15", "15-Nov-15", 1, 1}, {"20-Nov-15", "21-Nov-15", 1, 1},
{"21-Nov-15", "22-Nov-15", 1, 1}, {"12-Dec-15", "13-Dec-15", 1, 1},
{"26-Dec-15", "27-Dec-15", 1, 1}, {"19-Jan-13", "21-Jan-13", 1, 2},
{"7-Feb-13", "9-Feb-13", 1, 2}, {"9-Feb-13", "11-Feb-13", 1, 2},
{"19-Feb-13", "21-Feb-13", 1, 2}, {"5-Mar-13", "7-Mar-13", 1, 2},
{"11-Mar-13", "13-Mar-13", 1, 2}, {"14-Mar-13", "16-Mar-13", 1, 2},
{"25-Apr-13", "27-Apr-13", 1, 2}, {"2-May-13", "4-May-13", 1, 2},
{"13-May-13", "15-May-13", 1, 2}, {"16-May-13", "18-May-13", 1, 2},
{"29-May-13", "31-May-13", 1, 2}, {"7-Jun-13", "9-Jun-13", 1, 2},
{"25-Jun-13", "27-Jun-13", 1, 2}, {"17-Jul-13", "19-Jul-13", 1, 2},
{"7-Aug-13", "9-Aug-13", 1, 2}, {"5-Sep-13", "7-Sep-13", 1, 2},
{"25-Sep-13", "27-Sep-13", 1, 2}, {"27-Sep-13", "29-Sep-13", 1, 2},
{"7-Nov-13", "9-Nov-13", 1, 2}, {"21-Nov-13", "23-Nov-13", 1, 2},
{"23-Nov-13", "25-Nov-13", 1, 2}, {"1-Jan-14", "3-Jan-14", 1, 2},
{"14-Jan-14", "16-Jan-14", 1, 2}, {"3-Feb-14", "5-Feb-14", 1, 2},
{"14-Mar-14", "16-Mar-14", 1, 2}, {"21-Mar-14", "23-Mar-14", 1, 2},
{"7-Apr-14", "9-Apr-14", 1, 2}, {"12-Apr-14", "14-Apr-14", 1, 2},
```



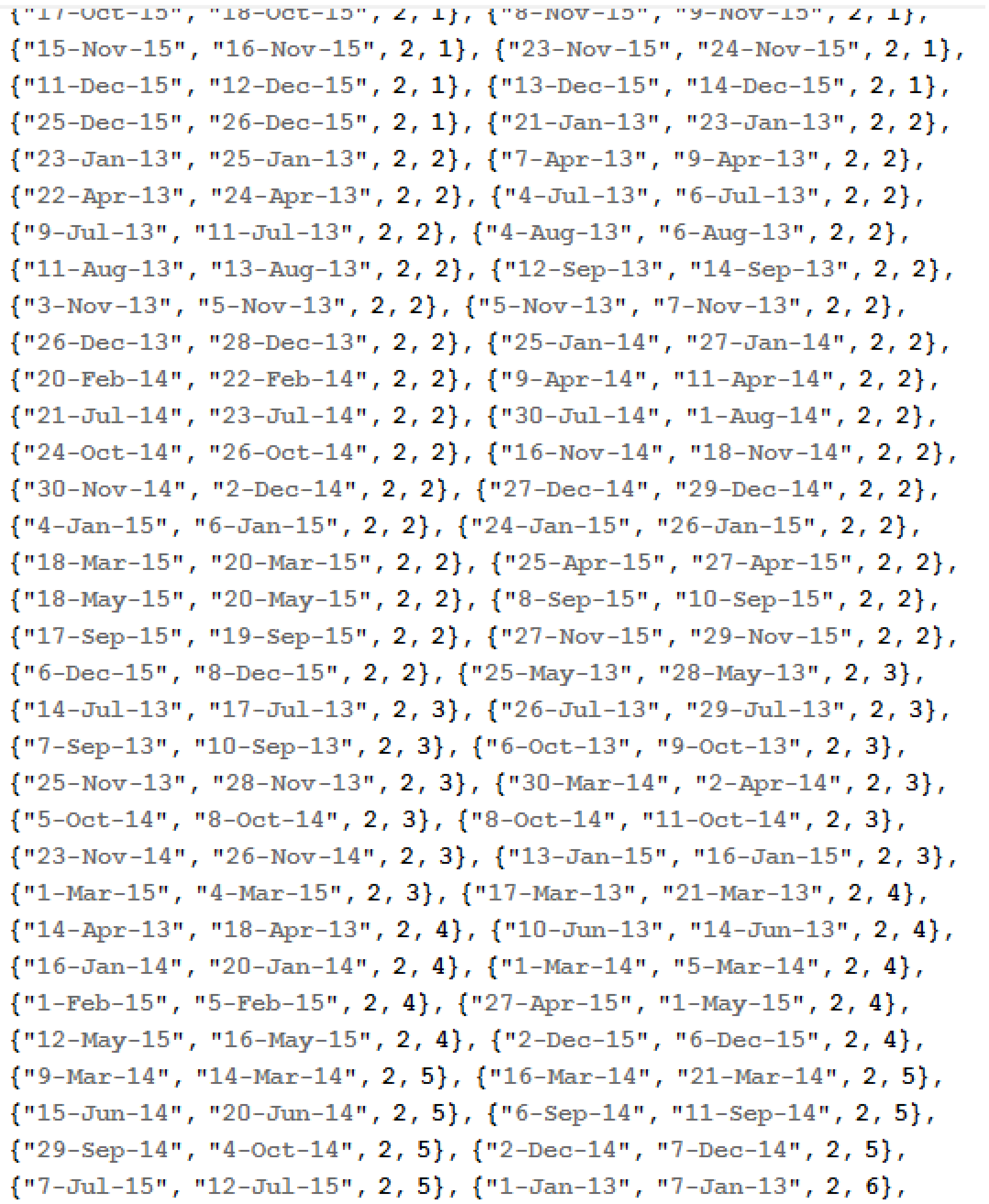
```
{"25-Apr-14", "27-Apr-14", 1, 2}, {"27-Apr-14", "29-Apr-14", 1, 2},
{"21-May-14", "23-May-14", 1, 2}, {"13-Jun-14", "15-Jun-14", 1, 2},
{"22-Jun-14", "24-Jun-14", 1, 2}, {"30-Jun-14", "2-Jul-14", 1, 2},
{"2-Jul-14", "4-Jul-14", 1, 2}, {"9-Jul-14", "11-Jul-14", 1, 2},
{"23-Jul-14", "25-Jul-14", 1, 2}, {"28-Jul-14", "30-Jul-14", 1, 2},
{"4-Aug-14", "6-Aug-14", 1, 2}, {"6-Aug-14", "8-Aug-14", 1, 2},
{"14-Aug-14", "16-Aug-14", 1, 2}, {"18-Aug-14", "20-Aug-14", 1, 2},
{"27-Aug-14", "29-Aug-14", 1, 2}, {"29-Aug-14", "31-Aug-14", 1, 2},
{"31-Aug-14", "2-Sep-14", 1, 2}, {"4-Sep-14", "6-Sep-14", 1, 2},
{"14-Oct-14", "16-Oct-14", 1, 2}, {"16-Oct-14", "18-Oct-14", 1, 2},
{"5-Nov-14", "7-Nov-14", 1, 2}, {"19-Nov-14", "21-Nov-14", 1, 2},
{"14-Dec-14", "16-Dec-14", 1, 2}, {"24-Dec-14", "26-Dec-14", 1, 2},
{"29-Dec-14", "31-Dec-14", 1, 2}, {"2-Jan-15", "4-Jan-15", 1, 2},
{"26-Jan-15", "28-Jan-15", 1, 2}, {"15-Feb-15", "17-Feb-15", 1, 2},
{"20-Feb-15", "22-Feb-15", 1, 2}, {"23-Feb-15", "25-Feb-15", 1, 2},
{"26-Feb-15", "28-Feb-15", 1, 2}, {"15-Mar-15", "17-Mar-15", 1, 2},
{"24-Mar-15", "26-Mar-15", 1, 2}, {"3-Apr-15", "5-Apr-15", 1, 2},
{"16-Apr-15", "18-Apr-15", 1, 2}, {"19-Apr-15", "21-Apr-15", 1, 2},
{"1-May-15", "3-May-15", 1, 2}, {"26-May-15", "28-May-15", 1, 2},
{"3-Jun-15", "5-Jun-15", 1, 2}, {"7-Jun-15", "9-Jun-15", 1, 2},
{"17-Jun-15", "19-Jun-15", 1, 2}, {"22-Jun-15", "24-Jun-15", 1, 2},
{"2-Jul-15", "4-Jul-15", 1, 2}, {"30-Jul-15", "1-Aug-15", 1, 2},
{"4-Aug-15", "6-Aug-15", 1, 2}, {"23-Aug-15", "25-Aug-15", 1, 2},
{"13-Sep-15", "15-Sep-15", 1, 2}, {"15-Sep-15", "17-Sep-15", 1, 2},
{"25-Sep-15", "27-Sep-15", 1, 2}, {"22-Oct-15", "24-Oct-15", 1, 2},
{"16-Nov-15", "18-Nov-15", 1, 2}, {"18-Nov-15", "20-Nov-15", 1, 2},
{"7-Mar-13", "10-Mar-13", 1, 3}, {"18-Apr-13", "21-Apr-13", 1, 3},
{"20-May-13", "23-May-13", 1, 3}, {"1-Jul-13", "4-Jul-13", 1, 3},
{"21-Jul-13", "24-Jul-13", 1, 3}, {"30-Jul-13", "2-Aug-13", 1, 3},
{"14-Aug-13", "17-Aug-13", 1, 3}, {"29-Sep-13", "2-Oct-13", 1, 3},
{"2-Oct-13", "5-Oct-13", 1, 3}, {"9-Oct-13", "12-Oct-13", 1, 3},
{"22-Dec-13", "25-Dec-13", 1, 3}, {"28-Dec-13", "31-Dec-13", 1, 3},
{"22-Feb-14", "25-Feb-14", 1, 3}, {"5-Mar-14", "8-Mar-14", 1, 3},
```



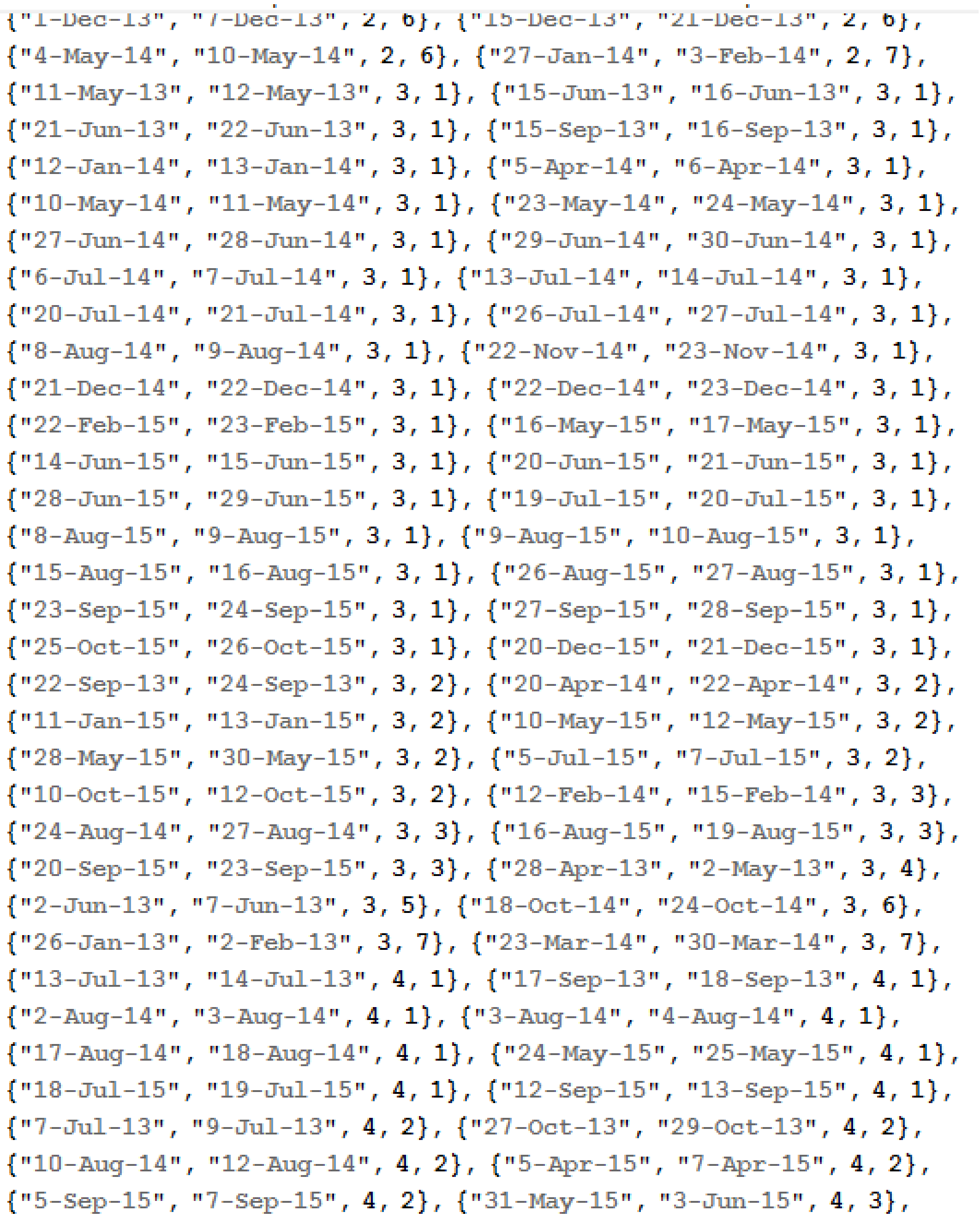
```
{"2-Apr-14", "5-Apr-14", 1, 3}, {"22-Apr-14", "25-Apr-14", 1, 3},
{"30-Apr-14", "3-May-14", 1, 3}, {"18-May-14", "21-May-14", 1, 3},
{"11-Sep-14", "14-Sep-14", 1, 3}, {"15-Sep-14", "18-Sep-14", 1, 3},
{"26-Oct-14", "29-Oct-14", 1, 3}, {"16-Jan-15", "19-Jan-15", 1, 3},
{"20-Jan-15", "23-Jan-15", 1, 3}, {"28-Jan-15", "31-Jan-15", 1, 3},
{"17-Feb-15", "20-Feb-15", 1, 3}, {"10-Mar-15", "13-Mar-15", 1, 3},
{"4-May-15", "7-May-15", 1, 3}, {"7-May-15", "10-May-15", 1, 3},
{"20-May-15", "23-May-15", 1, 3}, {"24-Jun-15", "27-Jun-15", 1, 3},
{"29-Jun-15", "2-Jul-15", 1, 3}, {"27-Jul-15", "30-Jul-15", 1, 3},
{"6-Oct-15", "9-Oct-15", 1, 3}, {"19-Oct-15", "22-Oct-15", 1, 3},
{"3-Nov-15", "6-Nov-15", 1, 3}, {"24-Nov-15", "27-Nov-15", 1, 3},
{"29-Nov-15", "2-Dec-15", 1, 3}, {"8-Dec-15", "11-Dec-15", 1, 3},
{"3-Feb-13", "7-Feb-13", 1, 4}, {"10-Apr-13", "14-Apr-13", 1, 4},
{"6-May-13", "10-May-13", 1, 4}, {"29-Oct-13", "2-Nov-13", 1, 4},
{"11-Nov-13", "15-Nov-13", 1, 4}, {"21-Jan-14", "25-Jan-14", 1, 4},
{"25-Feb-14", "1-Mar-14", 1, 4}, {"13-May-14", "17-May-14", 1, 4},
{"3-Jun-14", "7-Jun-14", 1, 4}, {"9-Jun-14", "13-Jun-14", 1, 4},
{"20-Aug-14", "24-Aug-14", 1, 4}, {"16-Dec-14", "20-Dec-14", 1, 4},
{"4-Mar-15", "8-Mar-15", 1, 4}, {"21-Apr-15", "25-Apr-15", 1, 4},
{"2-Oct-15", "6-Oct-15", 1, 4}, {"9-Nov-15", "13-Nov-15", 1, 4},
{"21-Dec-15", "25-Dec-15", 1, 4}, {"27-Dec-15", "31-Dec-15", 1, 4},
{"16-Jun-13", "21-Jun-13", 1, 5}, {"20-Aug-13", "25-Aug-13", 1, 5},
{"21-Oct-13", "26-Oct-13", 1, 5}, {"15-Nov-13", "20-Nov-13", 1, 5},
{"7-Feb-14", "12-Feb-14", 1, 5}, {"14-Jul-14", "19-Jul-14", 1, 5},
{"21-Sep-14", "26-Sep-14", 1, 5}, {"7-Dec-14", "12-Dec-14", 1, 5},
{"10-Aug-15", "15-Aug-15", 1, 5}, {"12-Oct-15", "17-Oct-15", 1, 5},
{"28-Oct-15", "2-Nov-15", 1, 5}, {"24-Feb-13", "2-Mar-13", 1, 6},
{"31-Mar-13", "6-Apr-13", 1, 6}, {"14-Apr-14", "20-Apr-14", 1, 6},
{"25-May-14", "31-May-14", 1, 6}, {"9-Feb-15", "15-Feb-15", 1, 6},
{"27-Mar-15", "2-Apr-15", 1, 6}, {"30-Aug-15", "5-Sep-15", 1, 6},
{"14-Dec-15", "20-Dec-15", 1, 6}, {"12-Feb-13", "19-Feb-13", 1, 7},
{"13-Oct-13", "20-Oct-13", 1, 7}, {"7-Dec-13", "14-Dec-13", 1, 7},
{"29-Oct-14", "5-Nov-14", 1, 7}, {"8-Nov-14", "15-Nov-14", 1, 7},
```



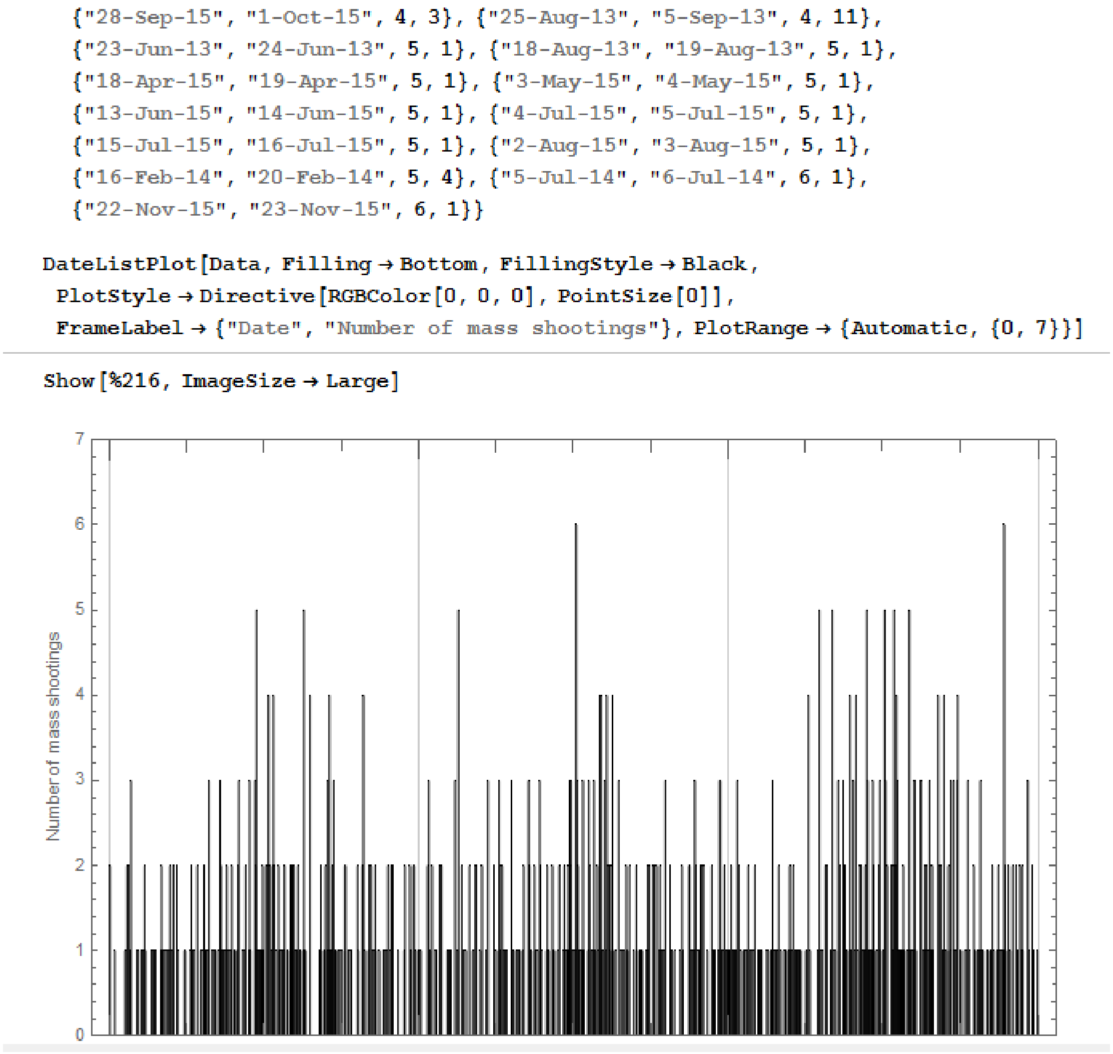
```
{"22-Mar-13", "30-Mar-13", 1, 8}, {"3-Jan-14", "11-Jan-14", 1, 8},
{"7-Apr-15", "16-Apr-15", 1, 9}, {"7-Jan-13", "19-Jan-13", 1, 12},
{"11-Feb-13", "12-Feb-13", 2, 1}, {"3-Mar-13", "4-Mar-13", 2, 1},
{"13-Mar-13", "14-Mar-13", 2, 1}, {"21-Mar-13", "22-Mar-13", 2, 1},
{"12-May-13", "13-May-13", 2, 1}, {"19-May-13", "20-May-13", 2, 1},
{"1-Jun-13", "2-Jun-13", 2, 1}, {"22-Jun-13", "23-Jun-13", 2, 1},
{"30-Jun-13", "1-Jul-13", 2, 1}, {"6-Jul-13", "7-Jul-13", 2, 1},
{"12-Jul-13", "13-Jul-13", 2, 1}, {"19-Jul-13", "20-Jul-13", 2, 1},
{"2-Aug-13", "3-Aug-13", 2, 1}, {"6-Aug-13", "7-Aug-13", 2, 1},
{"20-Sep-13", "21-Sep-13", 2, 1}, {"5-Oct-13", "6-Oct-13", 2, 1},
{"20-Oct-13", "21-Oct-13", 2, 1}, {"26-Oct-13", "27-Oct-13", 2, 1},
{"10-Nov-13", "11-Nov-13", 2, 1}, {"29-Nov-13", "30-Nov-13", 2, 1},
{"30-Nov-13", "1-Dec-13", 2, 1}, {"21-Dec-13", "22-Dec-13", 2, 1},
{"25-Dec-13", "26-Dec-13", 2, 1}, {"11-Jan-14", "12-Jan-14", 2, 1},
{"8-Mar-14", "9-Mar-14", 2, 1}, {"6-Apr-14", "7-Apr-14", 2, 1},
{"11-May-14", "12-May-14", 2, 1}, {"17-May-14", "18-May-14", 2, 1},
{"24-May-14", "25-May-14", 2, 1}, {"1-Jun-14", "2-Jun-14", 2, 1},
{"7-Jun-14", "8-Jun-14", 2, 1}, {"21-Jun-14", "22-Jun-14", 2, 1},
{"26-Jun-14", "27-Jun-14", 2, 1}, {"28-Jun-14", "29-Jun-14", 2, 1},
{"7-Jul-14", "8-Jul-14", 2, 1}, {"12-Aug-14", "13-Aug-14", 2, 1},
{"13-Aug-14", "14-Aug-14", 2, 1}, {"16-Aug-14", "17-Aug-14", 2, 1},
{"2-Sep-14", "3-Sep-14", 2, 1}, {"14-Sep-14", "15-Sep-14", 2, 1},
{"28-Sep-14", "29-Sep-14", 2, 1}, {"4-Oct-14", "5-Oct-14", 2, 1},
{"12-Oct-14", "13-Oct-14", 2, 1}, {"7-Nov-14", "8-Nov-14", 2, 1},
{"15-Nov-14", "16-Nov-14", 2, 1}, {"29-Nov-14", "30-Nov-14", 2, 1},
{"12-Dec-14", "13-Dec-14", 2, 1}, {"14-Mar-15", "15-Mar-15", 2, 1},
{"5-Jun-15", "6-Jun-15", 2, 1}, {"6-Jun-15", "7-Jun-15", 2, 1},
{"11-Jun-15", "12-Jun-15", 2, 1}, {"21-Jun-15", "22-Jun-15", 2, 1},
{"27-Jun-15", "28-Jun-15", 2, 1}, {"12-Jul-15", "13-Jul-15", 2, 1},
{"13-Jul-15", "14-Jul-15", 2, 1}, {"16-Jul-15", "17-Jul-15", 2, 1},
{"17-Jul-15", "18-Jul-15", 2, 1}, {"26-Jul-15", "27-Jul-15", 2, 1},
{"21-Aug-15", "22-Aug-15", 2, 1}, {"22-Aug-15", "23-Aug-15", 2, 1},
{"29-Aug-15", "30-Aug-15", 2, 1}, {"7-Sep-15", "8-Sep-15", 2, 1},
```



```
{"17-Oct-15", "18-Oct-15", 2, 1}, {"8-Nov-15", "9-Nov-15", 2, 1},
{"15-Nov-15", "16-Nov-15", 2, 1}, {"23-Nov-15", "24-Nov-15", 2, 1},
{"11-Dec-15", "12-Dec-15", 2, 1}, {"13-Dec-15", "14-Dec-15", 2, 1},
{"25-Dec-15", "26-Dec-15", 2, 1}, {"21-Jan-13", "23-Jan-13", 2, 2},
{"23-Jan-13", "25-Jan-13", 2, 2}, {"7-Apr-13", "9-Apr-13", 2, 2},
{"22-Apr-13", "24-Apr-13", 2, 2}, {"4-Jul-13", "6-Jul-13", 2, 2},
{"9-Jul-13", "11-Jul-13", 2, 2}, {"4-Aug-13", "6-Aug-13", 2, 2},
{"11-Aug-13", "13-Aug-13", 2, 2}, {"12-Sep-13", "14-Sep-13", 2, 2},
{"3-Nov-13", "5-Nov-13", 2, 2}, {"5-Nov-13", "7-Nov-13", 2, 2},
{"26-Dec-13", "28-Dec-13", 2, 2}, {"25-Jan-14", "27-Jan-14", 2, 2},
{"20-Feb-14", "22-Feb-14", 2, 2}, {"9-Apr-14", "11-Apr-14", 2, 2},
{"21-Jul-14", "23-Jul-14", 2, 2}, {"30-Jul-14", "1-Aug-14", 2, 2},
{"24-Oct-14", "26-Oct-14", 2, 2}, {"16-Nov-14", "18-Nov-14", 2, 2},
{"30-Nov-14", "2-Dec-14", 2, 2}, {"27-Dec-14", "29-Dec-14", 2, 2},
{"4-Jan-15", "6-Jan-15", 2, 2}, {"24-Jan-15", "26-Jan-15", 2, 2},
{"18-Mar-15", "20-Mar-15", 2, 2}, {"25-Apr-15", "27-Apr-15", 2, 2},
{"18-May-15", "20-May-15", 2, 2}, {"8-Sep-15", "10-Sep-15", 2, 2},
{"17-Sep-15", "19-Sep-15", 2, 2}, {"27-Nov-15", "29-Nov-15", 2, 2},
{"6-Dec-15", "8-Dec-15", 2, 2}, {"25-May-13", "28-May-13", 2, 3},
{"14-Jul-13", "17-Jul-13", 2, 3}, {"26-Jul-13", "29-Jul-13", 2, 3},
{"7-Sep-13", "10-Sep-13", 2, 3}, {"6-Oct-13", "9-Oct-13", 2, 3},
{"25-Nov-13", "28-Nov-13", 2, 3}, {"30-Mar-14", "2-Apr-14", 2, 3},
{"5-Oct-14", "8-Oct-14", 2, 3}, {"8-Oct-14", "11-Oct-14", 2, 3},
{"23-Nov-14", "26-Nov-14", 2, 3}, {"13-Jan-15", "16-Jan-15", 2, 3},
{"1-Mar-15", "4-Mar-15", 2, 3}, {"17-Mar-13", "21-Mar-13", 2, 4},
{"14-Apr-13", "18-Apr-13", 2, 4}, {"10-Jun-13", "14-Jun-13", 2, 4},
{"16-Jan-14", "20-Jan-14", 2, 4}, {"1-Mar-14", "5-Mar-14", 2, 4},
{"1-Feb-15", "5-Feb-15", 2, 4}, {"27-Apr-15", "1-May-15", 2, 4},
{"12-May-15", "16-May-15", 2, 4}, {"2-Dec-15", "6-Dec-15", 2, 4},
{"9-Mar-14", "14-Mar-14", 2, 5}, {"16-Mar-14", "21-Mar-14", 2, 5},
{"15-Jun-14", "20-Jun-14", 2, 5}, {"6-Sep-14", "11-Sep-14", 2, 5},
{"29-Sep-14", "4-Oct-14", 2, 5}, {"2-Dec-14", "7-Dec-14", 2, 5},
{"7-Jul-15", "12-Jul-15", 2, 5}, {"1-Jan-13", "7-Jan-13", 2, 6},
```

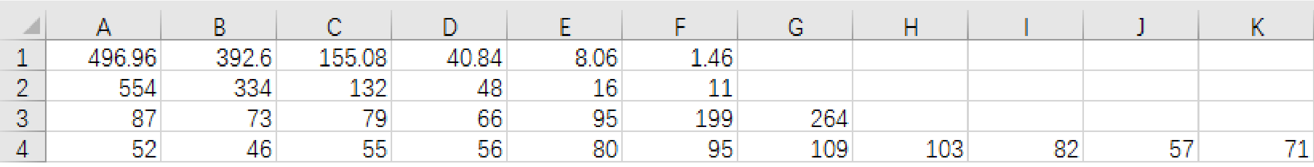


```
{"1-Dec-13", "7-Dec-13", 2, 6}, {"15-Dec-13", "21-Dec-13", 2, 6},
{"4-May-14", "10-May-14", 2, 6}, {"27-Jan-14", "3-Feb-14", 2, 7},
{"11-May-13", "12-May-13", 3, 1}, {"15-Jun-13", "16-Jun-13", 3, 1},
{"21-Jun-13", "22-Jun-13", 3, 1}, {"15-Sep-13", "16-Sep-13", 3, 1},
{"12-Jan-14", "13-Jan-14", 3, 1}, {"5-Apr-14", "6-Apr-14", 3, 1},
{"10-May-14", "11-May-14", 3, 1}, {"23-May-14", "24-May-14", 3, 1},
{"27-Jun-14", "28-Jun-14", 3, 1}, {"29-Jun-14", "30-Jun-14", 3, 1},
{"6-Jul-14", "7-Jul-14", 3, 1}, {"13-Jul-14", "14-Jul-14", 3, 1},
{"20-Jul-14", "21-Jul-14", 3, 1}, {"26-Jul-14", "27-Jul-14", 3, 1},
{"8-Aug-14", "9-Aug-14", 3, 1}, {"22-Nov-14", "23-Nov-14", 3, 1},
{"21-Dec-14", "22-Dec-14", 3, 1}, {"22-Dec-14", "23-Dec-14", 3, 1},
{"22-Feb-15", "23-Feb-15", 3, 1}, {"16-May-15", "17-May-15", 3, 1},
{"14-Jun-15", "15-Jun-15", 3, 1}, {"20-Jun-15", "21-Jun-15", 3, 1},
{"28-Jun-15", "29-Jun-15", 3, 1}, {"19-Jul-15", "20-Jul-15", 3, 1},
{"8-Aug-15", "9-Aug-15", 3, 1}, {"9-Aug-15", "10-Aug-15", 3, 1},
{"15-Aug-15", "16-Aug-15", 3, 1}, {"26-Aug-15", "27-Aug-15", 3, 1},
{"23-Sep-15", "24-Sep-15", 3, 1}, {"27-Sep-15", "28-Sep-15", 3, 1},
{"25-Oct-15", "26-Oct-15", 3, 1}, {"20-Dec-15", "21-Dec-15", 3, 1},
{"22-Sep-13", "24-Sep-13", 3, 2}, {"20-Apr-14", "22-Apr-14", 3, 2},
{"11-Jan-15", "13-Jan-15", 3, 2}, {"10-May-15", "12-May-15", 3, 2},
{"28-May-15", "30-May-15", 3, 2}, {"5-Jul-15", "7-Jul-15", 3, 2},
{"10-Oct-15", "12-Oct-15", 3, 2}, {"12-Feb-14", "15-Feb-14", 3, 3},
{"24-Aug-14", "27-Aug-14", 3, 3}, {"16-Aug-15", "19-Aug-15", 3, 3},
{"20-Sep-15", "23-Sep-15", 3, 3}, {"28-Apr-13", "2-May-13", 3, 4},
{"2-Jun-13", "7-Jun-13", 3, 5}, {"18-Oct-14", "24-Oct-14", 3, 6},
{"26-Jan-13", "2-Feb-13", 3, 7}, {"23-Mar-14", "30-Mar-14", 3, 7},
{"13-Jul-13", "14-Jul-13", 4, 1}, {"17-Sep-13", "18-Sep-13", 4, 1},
{"2-Aug-14", "3-Aug-14", 4, 1}, {"3-Aug-14", "4-Aug-14", 4, 1},
{"17-Aug-14", "18-Aug-14", 4, 1}, {"24-May-15", "25-May-15", 4, 1},
{"18-Jul-15", "19-Jul-15", 4, 1}, {"12-Sep-15", "13-Sep-15", 4, 1},
{"7-Jul-13", "9-Jul-13", 4, 2}, {"27-Oct-13", "29-Oct-13", 4, 2},
{"10-Aug-14", "12-Aug-14", 4, 2}, {"5-Apr-15", "7-Apr-15", 4, 2},
{"5-Sep-15", "7-Sep-15", 4, 2}, {"31-May-15", "3-Jun-15", 4, 3},
```

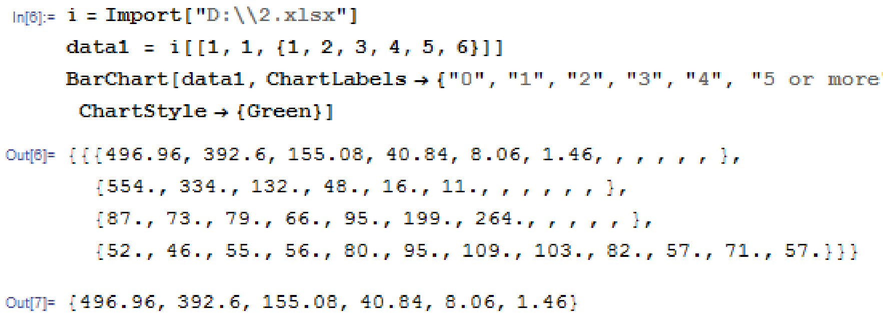


```
  {"28-Sep-15", "1-Oct-15", 4, 3}, {"25-Aug-13", "5-Sep-13", 4, 11},
  {"23-Jun-13", "24-Jun-13", 5, 1}, {"18-Aug-13", "19-Aug-13", 5, 1},
  {"18-Apr-15", "19-Apr-15", 5, 1}, {"3-May-15", "4-May-15", 5, 1},
  {"13-Jun-15", "14-Jun-15", 5, 1}, {"4-Jul-15", "5-Jul-15", 5, 1},
  {"15-Jul-15", "16-Jul-15", 5, 1}, {"2-Aug-15", "3-Aug-15", 5, 1},
  {"16-Feb-14", "20-Feb-14", 5, 4}, {"5-Jul-14", "6-Jul-14", 6, 1},
  {"22-Nov-15", "23-Nov-15", 6, 1}}

DateListPlot[Data, Filling → Bottom, FillingStyle → Black,
 PlotStyle → Directive[RGBColor[0, 0, 0], PointSize[0]],
 FrameLabel → {"Date", "Number of mass shootings"}, PlotRange → {Automatic, {0, 7}}]

Show[%216, ImageSize → Large]
```

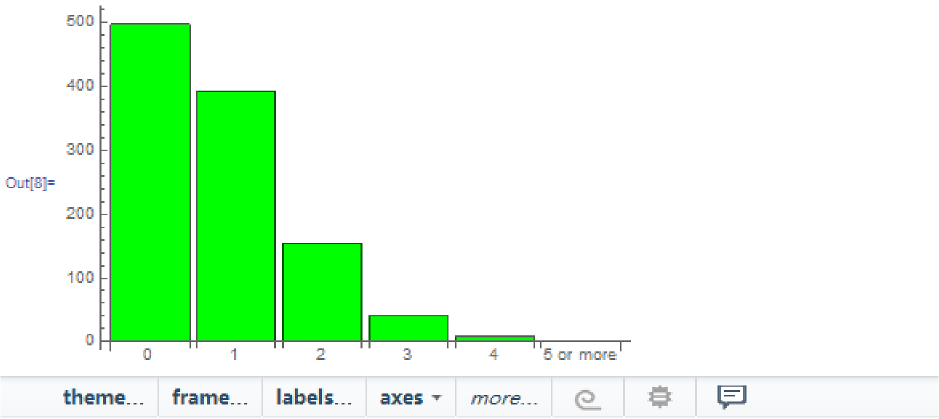
A.2.2 Hypothesis Test for Poisson Distribution

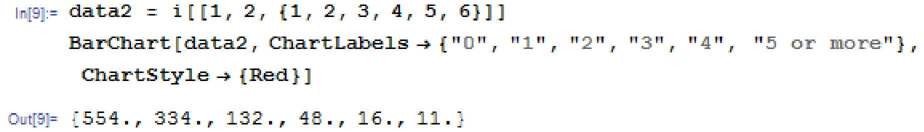


	A	B	C	D	E	F	G	H	I	J	K
1	496.96	392.6	155.08	40.84	8.06	1.46					
2	554	334	132	48	16	11					
3	87	73	79	66	95	199	264				
4	52	46	55	56	80	95	109	103	82	57	71

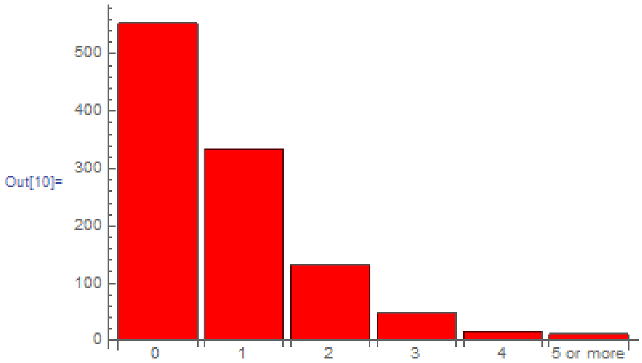


```
In[6]:= i = Import["D:\\2.xlsx"]
        data1 = i[[1, 1, {1, 2, 3, 4, 5, 6}]]
        BarChart[data1, ChartLabels → {"0", "1", "2", "3", "4", "5 or more"},
         ChartStyle → {Green}]

Out[6]= {{{496.96, 392.6, 155.08, 40.84, 8.06, 1.46, , , , , , },
          {554., 334., 132., 48., 16., 11., , , , , , },
          {87., 73., 79., 66., 95., 199., 264., , , , , },
          {52., 46., 55., 56., 80., 95., 109., 103., 82., 57., 71., 57.}}}

Out[7]= {496.96, 392.6, 155.08, 40.84, 8.06, 1.46}
```

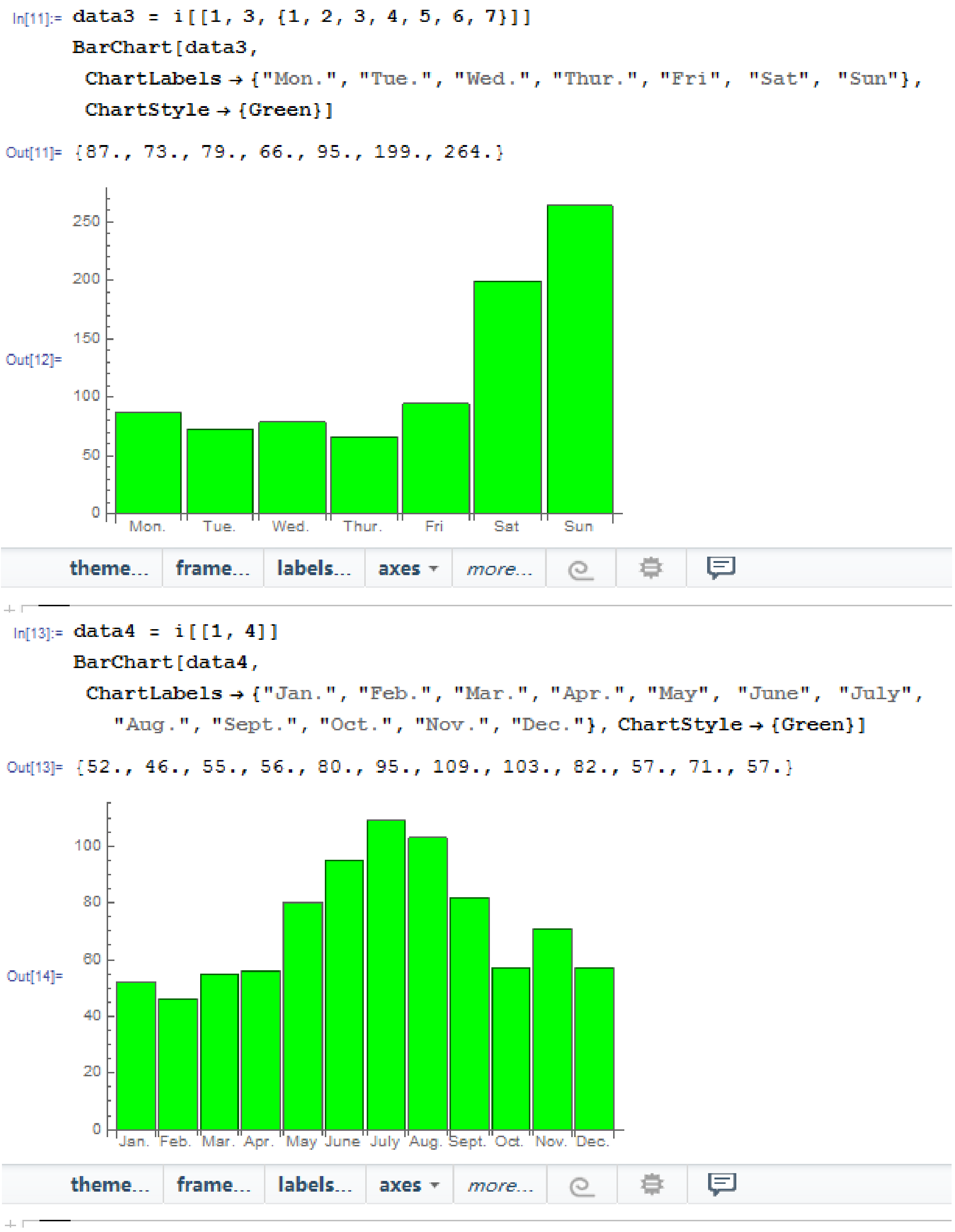




```
In[9]:= data2 = i[[1, 2, {1, 2, 3, 4, 5, 6}]]
        BarChart[data2, ChartLabels → {"0", "1", "2", "3", "4", "5 or more"},
         ChartStyle → {Red}]

Out[9]= {554., 334., 132., 48., 16., 11.}
```



A.2.3 Test Dependence on Weekdays and Months



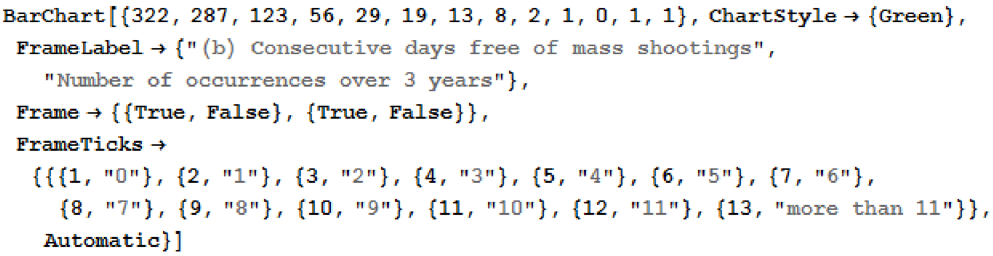
```
In[11]:= data3 = i[[1, 3, {1, 2, 3, 4, 5, 6, 7}]]
         BarChart[data3,
          ChartLabels → {"Mon.", "Tue.", "Wed.", "Thur.", "Fri", "Sat", "Sun"},
          ChartStyle → {Green}]

Out[11]= {87., 73., 79., 66., 95., 199., 264.}
```

Out[12]=

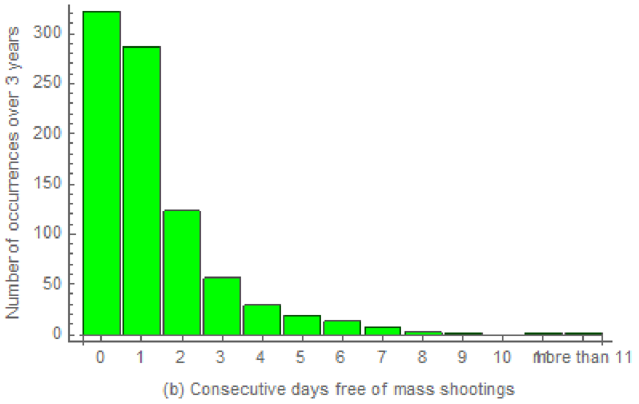
```
In[13]:= data4 = i[[1, 4]]
         BarChart[data4,
          ChartLabels → {"Jan.", "Feb.", "Mar.", "Apr.", "May", "June", "July",
            "Aug.", "Sept.", "Oct.", "Nov.", "Dec."}, ChartStyle → {Green}]

Out[13]= {52., 46., 55., 56., 80., 95., 109., 103., 82., 57., 71., 57.}
```

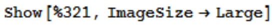
Out[14]=

A.2.4 Test the gaps between mass shootings

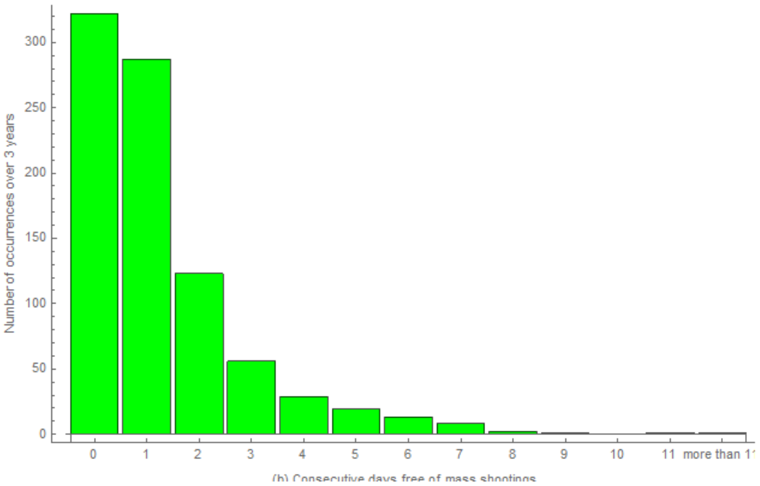


```
BarChart[{322, 287, 123, 56, 29, 19, 13, 8, 2, 1, 0, 1, 1}, ChartStyle → {Green},
 FrameLabel → {"(b) Consecutive days free of mass shootings",
   "Number of occurrences over 3 years"},
 Frame → {{True, False}, {True, False}},
 FrameTicks →
  {{{1, "0"}, {2, "1"}, {3, "2"}, {4, "3"}, {5, "4"}, {6, "5"}, {7, "6"},
    {8, "7"}, {9, "8"}, {10, "9"}, {11, "10"}, {12, "11"}, {13, "more than 11"}},
   Automatic}]
```

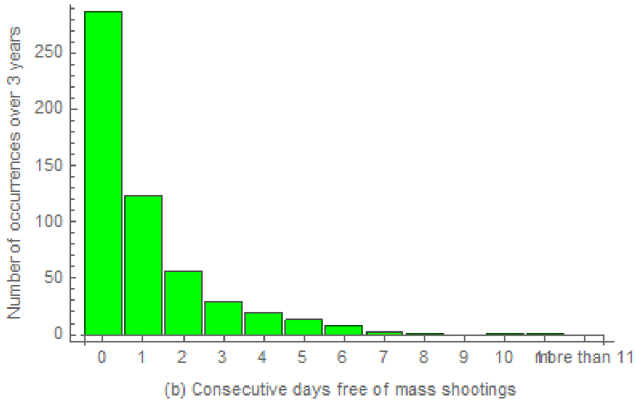




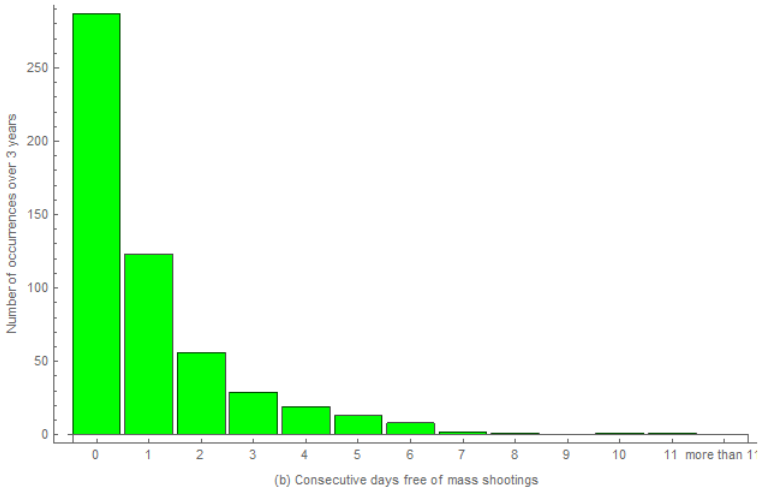
```
Show[%321, ImageSize → Large]
```

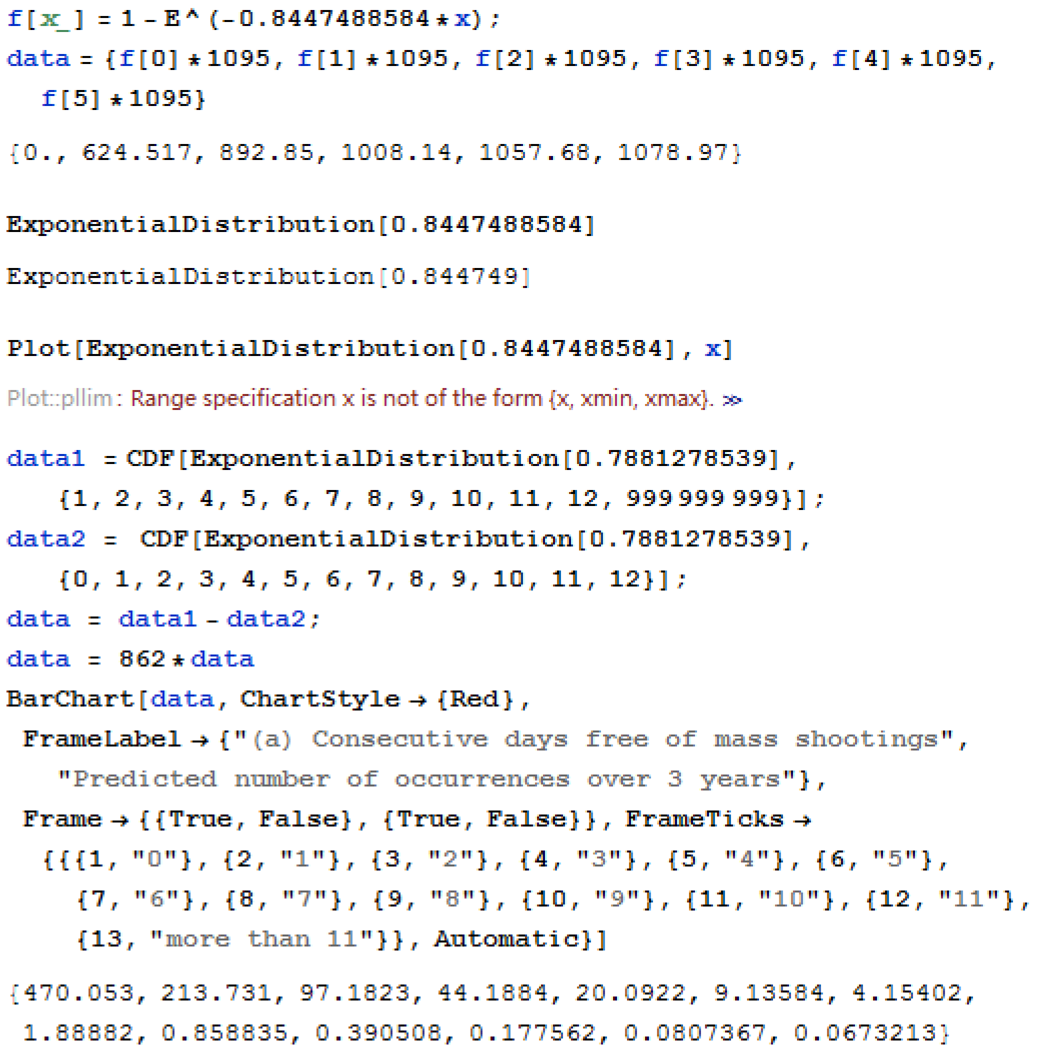


```
BarChart[{287, 123, 56, 29, 19, 13, 8, 2, 1, 0, 1, 1, 0}, ChartStyle → {Green},
 FrameLabel → {"(b) Consecutive days free of mass shootings",
   "Number of occurrences over 3 years"},
 Frame → {{True, False}, {True, False}},
 FrameTicks →
  {{{1, "0"}, {2, "1"}, {3, "2"}, {4, "3"}, {5, "4"}, {6, "5"}, {7, "6"},
    {8, "7"}, {9, "8"}, {10, "9"}, {11, "10"}, {12, "11"}, {13, "more than 11"}},
   Automatic}]
```

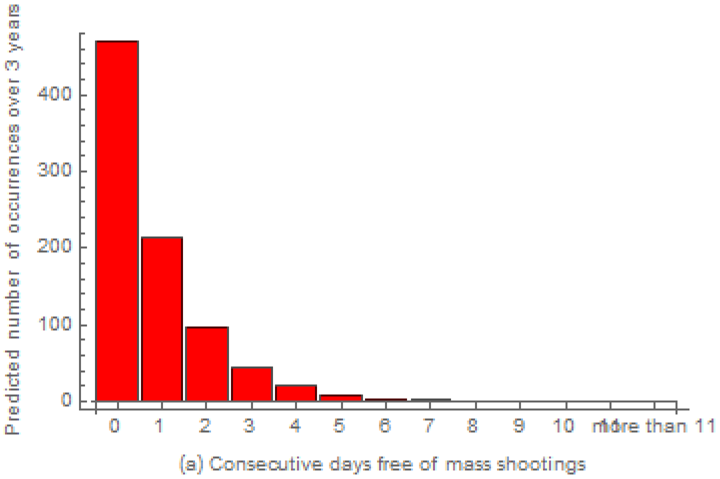


```
Show[%319, ImageSize → Large]
```

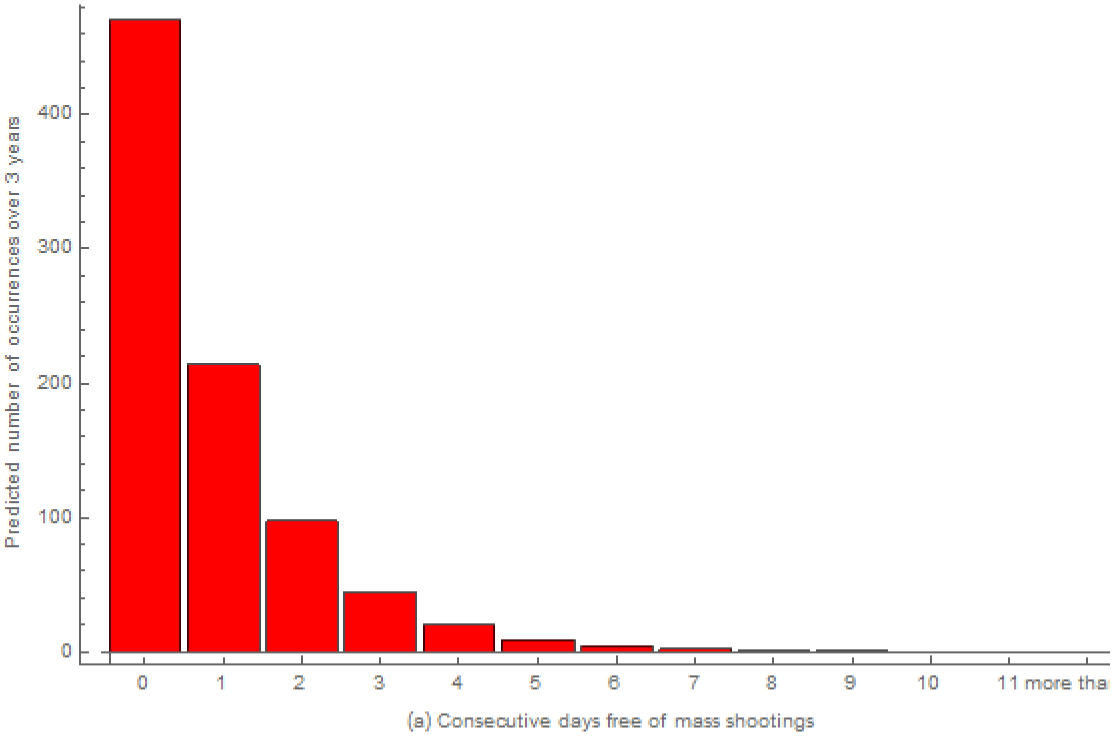




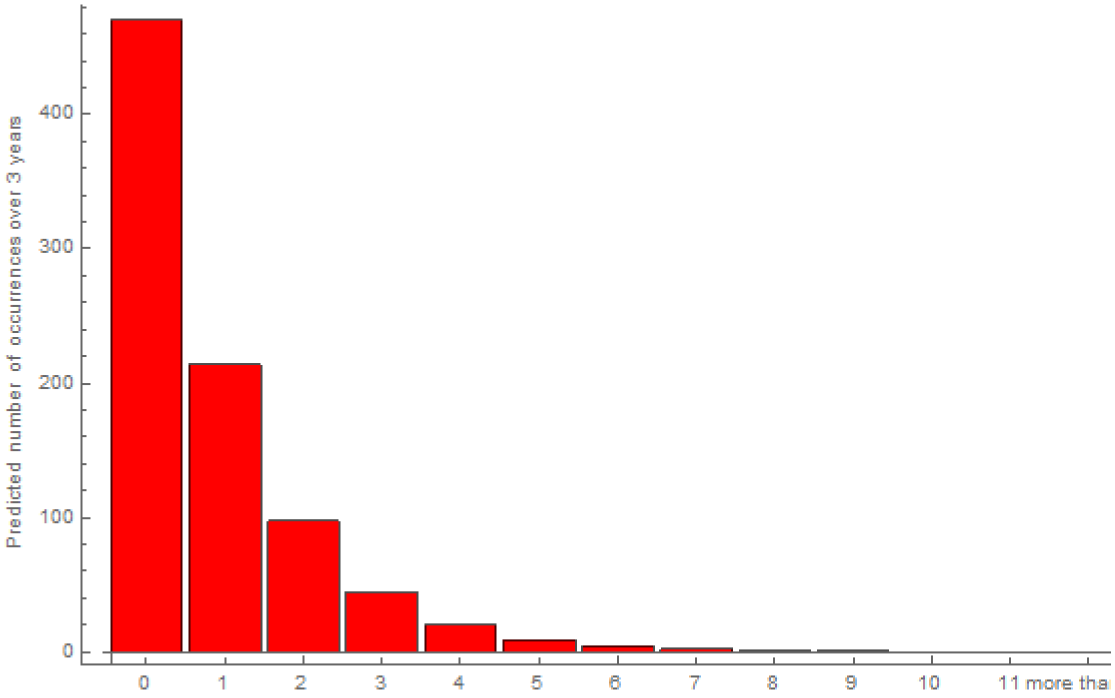
```
f[x_] = 1 - E^(-0.8447488584 * x);
data = {f[0] * 1095, f[1] * 1095, f[2] * 1095, f[3] * 1095, f[4] * 1095,
  f[5] * 1095}

{0., 624.517, 892.85, 1008.14, 1057.68, 1078.97}

ExponentialDistribution[0.8447488584]

ExponentialDistribution[0.844749]

Plot[ExponentialDistribution[0.8447488584], x]

Plot::pllim : Range specification x is not of the form {x, xmin, xmax}. >>

data1  = CDF[ExponentialDistribution[0.7881278539],
   {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 999 999 999}];
data2  =  CDF[ExponentialDistribution[0.7881278539],
   {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}];
data  =  data1 - data2;
data  =  862 * data
BarChart[data, ChartStyle → {Red},
 FrameLabel → {"(a) Consecutive days free of mass shootings",
   "Predicted number of occurrences over 3 years"},
 Frame → {{True, False}, {True, False}}, FrameTicks →
  {{{1, "0"}, {2, "1"}, {3, "2"}, {4, "3"}, {5, "4"}, {6, "5"},
    {7, "6"}, {8, "7"}, {9, "8"}, {10, "9"}, {11, "10"}, {12, "11"},
    {13, "more than 11"}}, Automatic}]

{470.053, 213.731, 97.1823, 44.1884, 20.0922, 9.13584, 4.15402,
 1.88882, 0.858835, 0.390508, 0.177562, 0.0807367, 0.0673213}
```

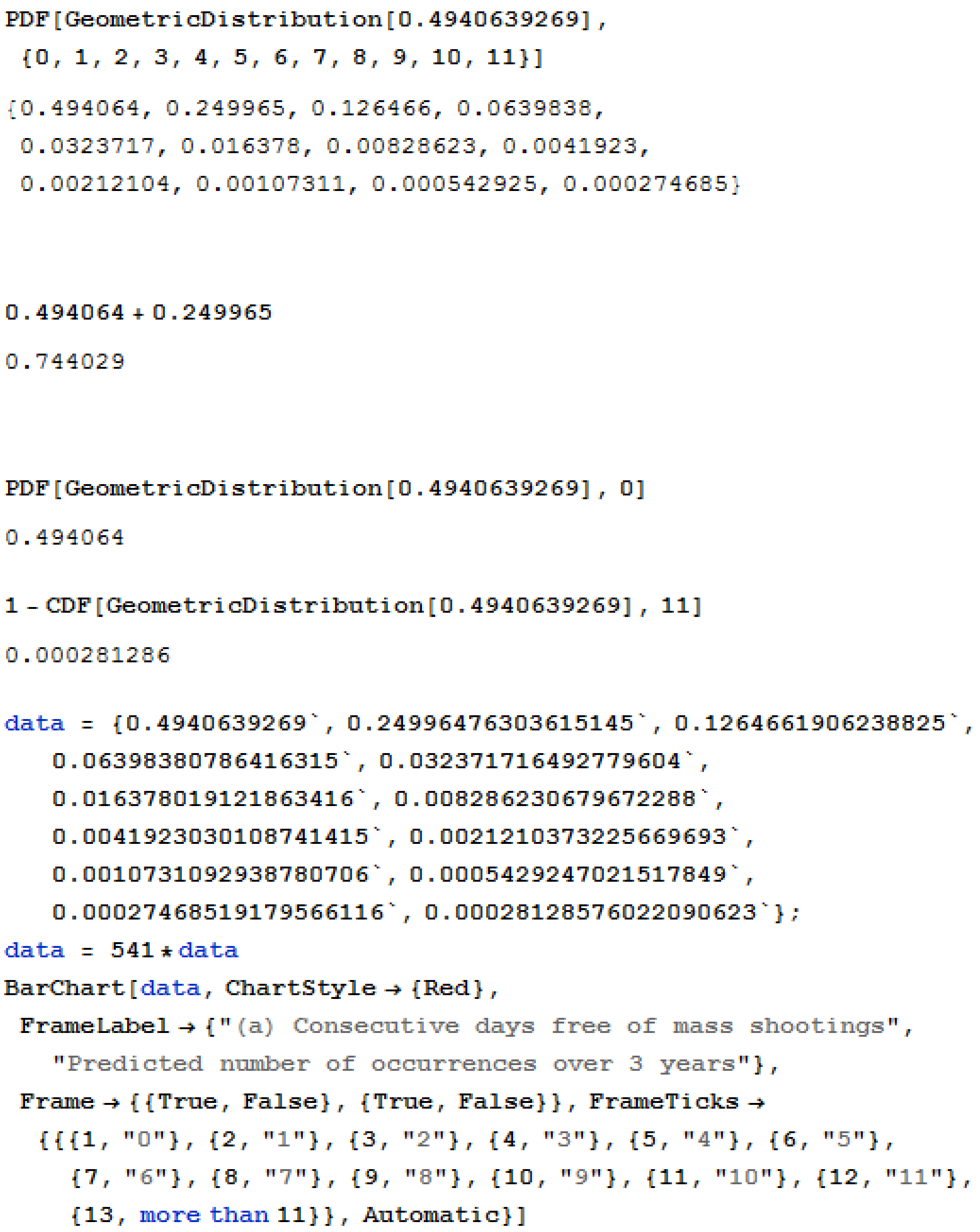


```
Show[%328, ImageSize → Large]
```

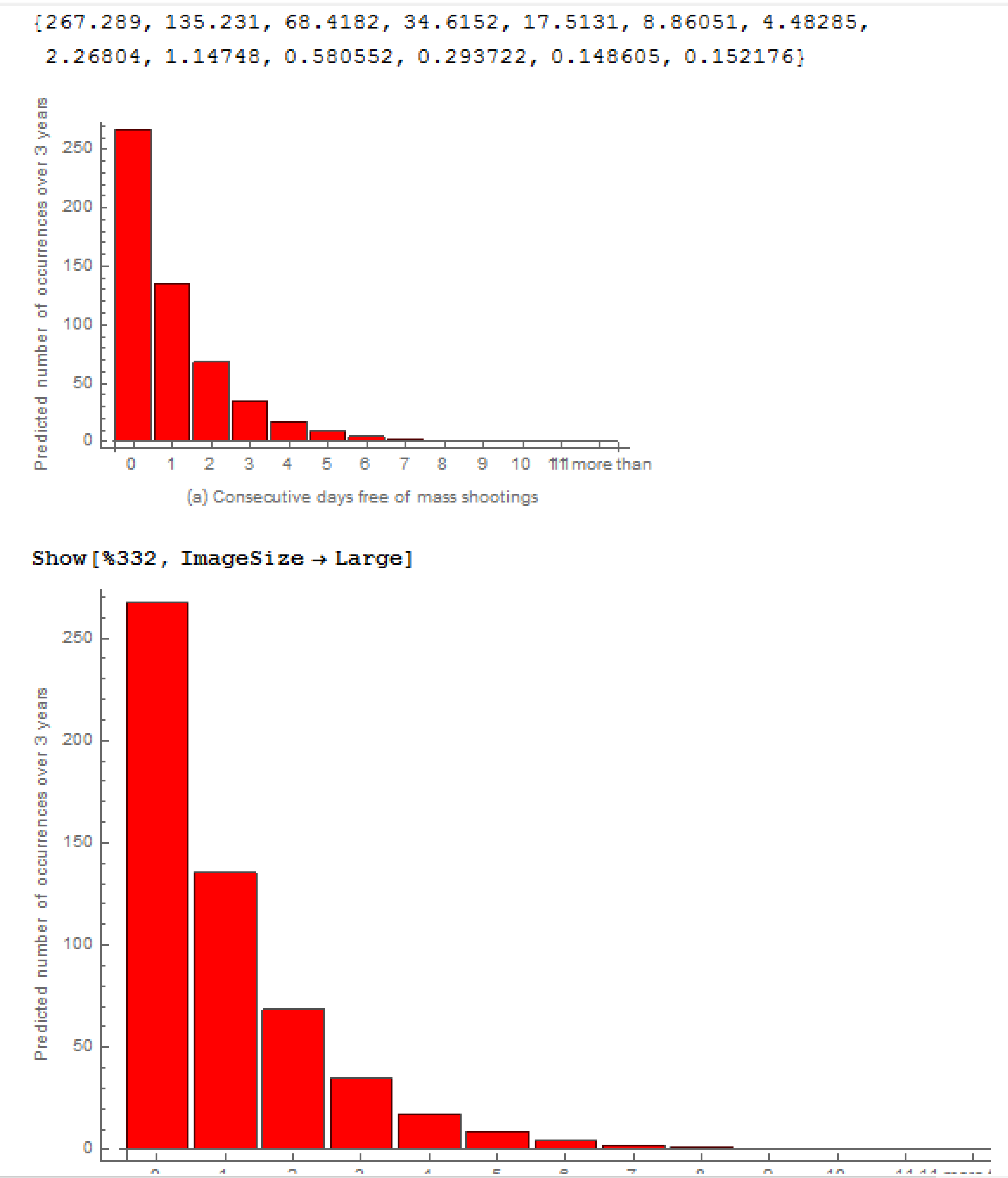


```
Show[%314, ImageSize → Large]
```





```
PDF[GeometricDistribution[0.4940639269],
 {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}]

{0.494064, 0.249965, 0.126466, 0.0639838,
 0.0323717, 0.016378, 0.00828623, 0.0041923,
 0.00212104, 0.00107311, 0.000542925, 0.000274685}



0.494064 + 0.249965

0.744029



PDF[GeometricDistribution[0.4940639269], 0]

0.494064

1 - CDF[GeometricDistribution[0.4940639269], 11]

0.000281286

data = {0.4940639269`, 0.24996476303615145`, 0.1264661906238825`,
   0.06398380786416315`, 0.032371716492779604`,
   0.016378019121863416`, 0.008286230679672288`,
   0.004192303010874141`, 0.0021210373225669693`,
   0.0010731092938780706`, 0.0005429247021517849`,
   0.00027468519179566116`, 0.00028128576022090623`};
data = 541 * data
BarChart[data, ChartStyle → {Red},
 FrameLabel → {"(a) Consecutive days free of mass shootings",
   "Predicted number of occurrences over 3 years"},
 Frame → {{True, False}, {True, False}}, FrameTicks →
  {{{1, "0"}, {2, "1"}, {3, "2"}, {4, "3"}, {5, "4"}, {6, "5"},
    {7, "6"}, {8, "7"}, {9, "8"}, {10, "9"}, {11, "10"}, {12, "11"},
    {13, more than 11}}, Automatic}]
```



```
{267.289, 135.231, 68.4182, 34.6152, 17.5131, 8.86051, 4.48285,
 2.26804, 1.14748, 0.580552, 0.293722, 0.148605, 0.152176}
```

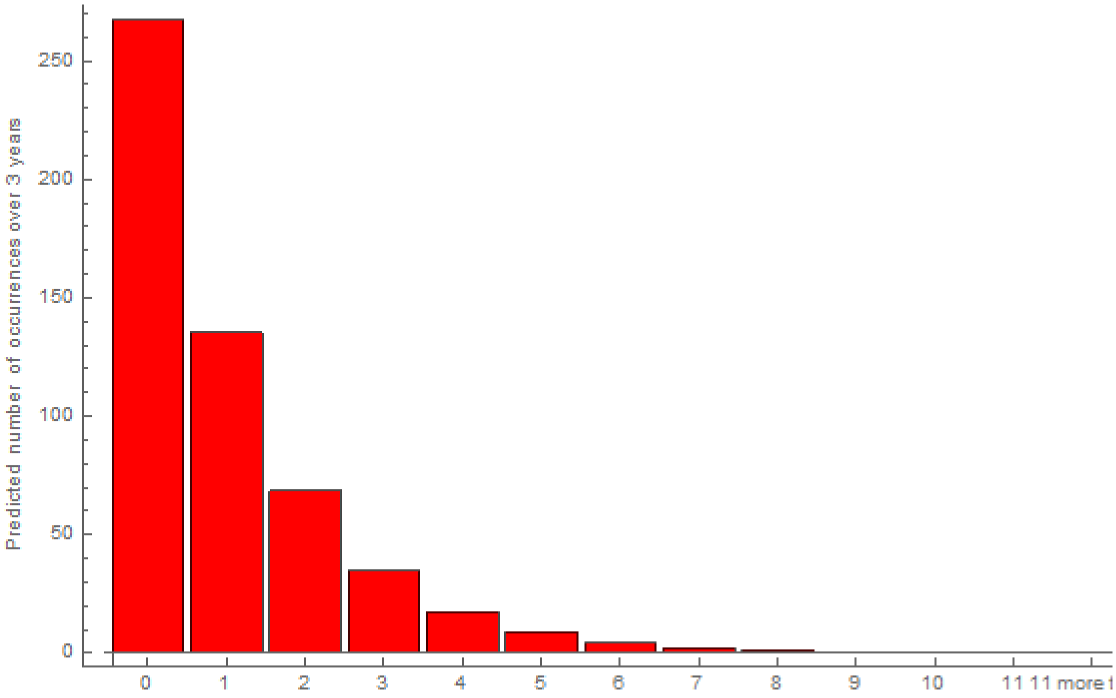
Predicted number of occurrences over 3 years
250
200
150
100
50
0
0
1
2
3
4
5
6
7
8
9
10
11
11 more than

(a) Consecutive days free of mass shootings

```
Show[%332, ImageSize → Large]
```

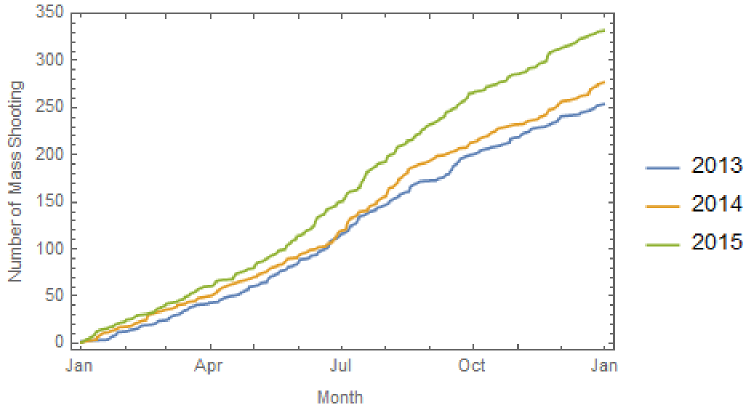
Predicted number of occurrences over 3 years
250
200
150
100
50
0

```
Show[%307, ImageSize → Large]
```



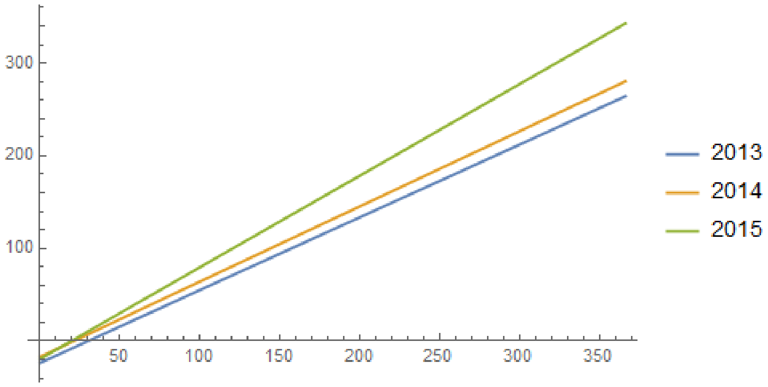
A.2.5 Research on cumulative numbers of mass shootings

```
DateListPlot[{Data2013, Data2014, Data2015},
 PlotLegends → {"2013", "2014", "2015"},
 FrameLabel → {"Month", "Number of Mass Shooting"}]
```

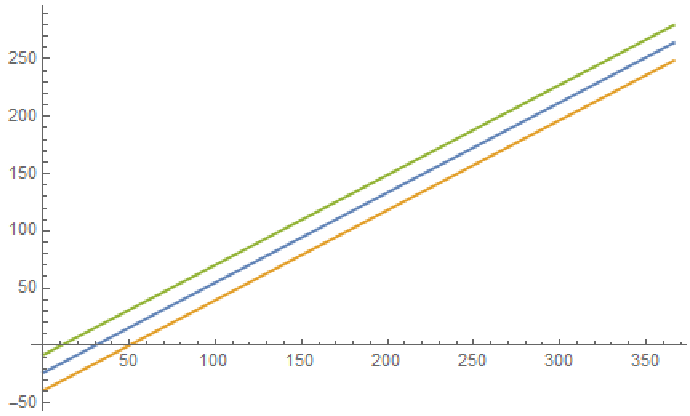


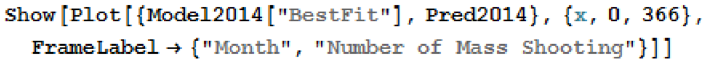
```
Model2013 = LinearModelFit[Data2013linear, x, x]

FittedModel[ -23.6611 + 0.786404 x ]

Model2014 = LinearModelFit[Data2014linear, x, x]

FittedModel[ -17.1043 + 0.812581 x ]

Model2015 = LinearModelFit[Data2015linear, x, x]

FittedModel[ -19.3567 + 0.989794 x ]
```

```
Plot[{Model2013["BestFit"], Model2014["BestFit"],
  Model2015["BestFit"]}, {x, 0, 366},
 PlotLegends → {"2013", "2014", "2015"},
 FrameLabel → {"Month", "Number of Mass Shooting"}]
```

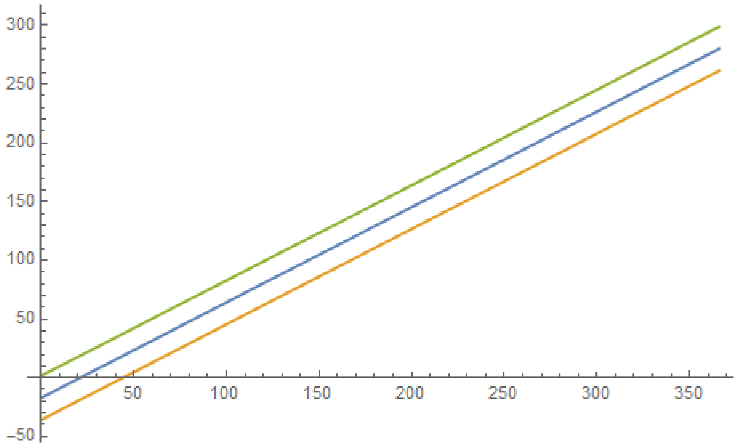


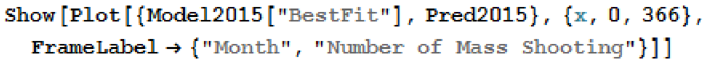
```
Show[Plot[{Model2013["BestFit"], Pred2013}, {x, 0, 366},
  FrameLabel → {"Month", "Number of Mass Shooting"}]]
```



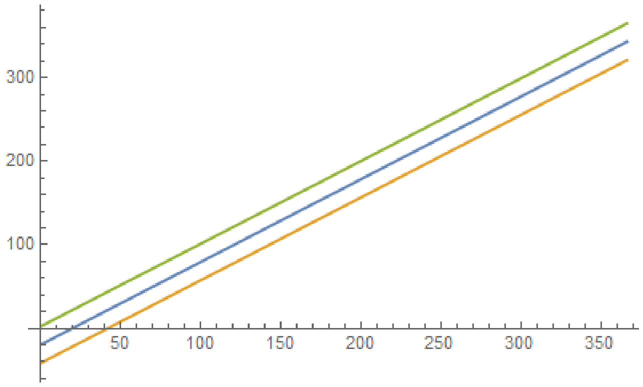


```
Show[Plot[{Model2014["BestFit"], Pred2014}, {x, 0, 366},
  FrameLabel → {"Month", "Number of Mass Shooting"}]]
```





```
Show[Plot[{Model2015["BestFit"], Pred2015}, {x, 0, 366},
  FrameLabel → {"Month", "Number of Mass Shooting"}]]
```



A.3 The German Bundesliga 2015-2016

Acknowledgments

I would like to say thank you for all of the team members. Your great work really contribute a lot to this essay. Also I would say thank you to Professor Horst Hohberger for teaching us so many useful knowledge on statistics. All of us improved a lot during the whole project and this will be an unforgettable experience for me. To be specific, Section 1 is done independently by Ruchen Zhen. Section 2.1 and 2.6 is done by Xinyi WU, 2.3 and 2.4 is done by Pengwei Ni, and 2.2 and 2.5 are done by Pengwei Ni. Section 3 is done by Jiang Lu.

Xinyi Wu

2016.8.6
UMJI-Joint Institute, Shanghai Jiao Tong University

References

[1] http://www.shootingtracker.com/

[2] http://www.gunviolencearchive.org/methodology

[3] Weiss, Jeffrey (December 6, 2015). ”Mass shootings in the U.S. this year? 353 or 4, depending on your definition”. Dallas Morning News. Retrieved December 6, 2015.

[4] ”About the Mass Shooting Tracker”. Mass Shooting Tracker. Retrieved 13 June 2016.

[5] ”Serial Murder - Multi-Disciplinary Perspectives for Investigators”. Federal Bureau of Investigation. 2005. Retrieved March 17, 2016.

[6] 全国法制计量管理计量技术委员会/定量包装商品净含量工作组. 定量包装商品净含量计量检测规则(Rules of metrological testing for net quantities of products in prepackages with fixed content). Technical Report JJF 1070-2005, 中国计量出版社, 2005. http://www.fsqtsc.gov.cn/fsqtsc/FSQTSCBook/1607.aspx [Online; accessed 5-July-2015].