Post02: Chi-squared Test Using R

Ranzhi Xue

December 2, 2017

Chi-squared Test Using R

Introduction: Motivation and Content

In the previous post, I explored how we may use R to compute hypothesis testing (including t-test and z-test), which can be applied to various fields or industries for data analysis. While this is primarily focused on quantitative variables of a population, I realized it is also important to explore the relationship between qualitative variables, which sometimes receive less attention but can end up being confounding factors or have influences on the data we are analyzing. For example, is survival on Titanic dependent on the cruise class? Is survival on Titanic dependent on gender? In order to test hypotheses like these, I will introduce the Chi-squared test of independence in this post, and go through the steps of testing and analysis. This whole process would also include plotting of some distributions and introduction of some new functions.

Background

To start off, what does it mean to have two variables that are independent? Accroding to R Tutorial, two random variables x and y are called independent if the probability distribution of one variable is not affected by the presence of another. To give you better sense or help refresh your memory, I will explain how the Chi-squared test works to determine independence with our previous Titanic example.

To start off, suppose we are wondering if survival on Titanic depends on the cruise class (e.g. you may guess first class had a higher chance of survival). Chi-squared, our key indicator in this case, is equal to the sum of the ratio between square of observed frequencies minus expected frequencies over expected frequencies. Although we have data on observed frequencies, we don't actually know the expected frequencies. The Chi-squared test estimates expected frequencies with the equation: row total times column total over sample size. After we obtain the two tables containing data for observed and expected frequencies, we can then start applying these data to the equation of Chi-squared above. Finally, we will compare our Chi-squared value to the critical value of Chi-squared at 5% significance level, which can be found in Reference 1. If it is larger than the critical value, that means at 5% significance level (or we are 95% confident that) survival on Titanic is dependent on the cruise class. If it is smaller than the critical value, then we know survival on Titanic and cruise class are independent of each other.

Data and Package Preparation

Given the steps of computing a Chi-squared test, let's have an actual analysis of our Titanic example. Let's first start with preparation for data, packages and any other information required for the reproducibility of this analysis.

```r
# R version: 3.4.1 (2017-06-30) -- "Single Candle"

# RStudio version: 1.0.153

# Create a data frame for the built-in R dataset 'Titanic'
Titanic <- data.frame(Titanic)

# Installation of Packages:
# ggplot2 (version: 2.2.1) - to plot Chi-squared distribution and rejection area
# stats (version: 3.4.1) - this should be built-in
# base (version: 3.4.1) - this should be built-in

library(ggplot2)
```

Getting started

Now, let's start with creating the hypotheses for our Chi-squared test.

H0: Survival and Class on Titanic are independent. (Null hypothesis)

H1: Survival and Class on Titanic are dependent. (Alternative hypothesis)

The next step is to create the two tables for observed and expected frequencies by manipulating the Titanic data frame.

```r
# Create a contingency table for the observed frequencies of Survival and Class
obs_tbl <- xtabs(Freq ~ Class + Survived, data = Titanic)
obs_tbl
```

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

```
# Create a contingency table for the expected frequencies of Survival and Class

# Create an empty 4 by 2 matrix
exp_tbl <- matrix(nrow = 4, ncol = 2)

# Fill in the matrix with values
# according to the equation for expected frequencies
for (i in 1:4) {
  for (j in 1:2) {
    exp_tbl[i,j] <- sum(obs_tbl[i,]) * sum(obs_tbl[,j]) / margin.table(obs_tbl)
  }
}

# Change the names of columns and rows
colnames(exp_tbl) <- c("No", "Yes")
row.names(exp_tbl) <- c("1st", "2nd", "3rd", "Crew")

exp_tbl
```

```
##            No       Yes
## 1st  220.0136 104.98637
## 2nd  192.9350  92.06497
## 3rd  477.9373 228.06270
## Crew 599.1140 285.88596
```

Chi-squared Test

With the two tables for observed frequencies and expected frequencies of survival and class on Titanic, let’s now proceed to create a function for computing Chi-squared.

```
# Create a function to compute Chi-squared
chi_squared <- function(observed, expected) {
  chi_squared <- 0
  for (i in 1 : nrow(observed)) {
    for(j in 1 : ncol(observed)) {
      chi_squared <- chi_squared +
                    (observed[i,j] - expected[i,j]) ^ 2 / expected[i,j]
    }
  }

  # Print the output
  print(chi_squared, digits = 4)
}

chi_squared(obs_tbl, exp_tbl)
```

```
## [1] 190.4
```

Compare with the Critical Value and Draw a Conclusion

To compare the Chi-squared value we got with the critical value, we first need to find the corresponding critical value. In order to do that, besides the significance level we have (a = 5%), we also need to find the degree of freedom according to the function:

df = (nrow-1) * (ncol-1)

In this case, df = (4-1)(2-1) = 3.

Now, find the corresponding critical value on the table shown in Reference 1. When df = 3 and significance level = 0.05, the critical value for Chi-squared is about 7.82. Since chi_squared is way larger than the critical value (190.4011 > 7.82), we should reject the null hypothesis.

Alternative Approach

The Chi-squared test function we just created allows us to directly apply Chi-squared test to other categorical variables in other populations in the future as long as we have the two contingency tables of observed and expected frequencies, which can be created according to the dataset. If we want, we can also eliminate the process of computing the two tables by simply incorporating the process into a single function to compute the Chi-squared value and the degree of freedom.

```
# Create a function to compute the Chi-squared value and degree of freedom
# according to a dataset with frequencies

chi_squared <- function(dataset, Var1, Var2) {
  # Create the table for observed frequencies
  obs_tbl <- xtabs(Freq ~ Var1 + Var2, data = dataset)

  # Create the table for observed frequencies
  exp_tbl <- matrix(nrow = nrow(obs_tbl), ncol = ncol(obs_tbl))
  for (i in 1:nrow(obs_tbl)) {
    for (j in 1:ncol(obs_tbl)) {
      exp_tbl[i,j] <- sum(obs_tbl[i,]) * sum(obs_tbl[,j]) / margin.table(obs_tbl)
    }
  }

  # Compute the Chi-squared value
  chi_squared <- 0
  for (i in 1 : nrow(obs_tbl)) {
    for(j in 1 : ncol(obs_tbl)) {
      chi_squared <- chi_squared +
                    (obs_tbl[i,j] - exp_tbl[i,j]) ^ 2 / exp_tbl[i,j]
    }
  }

  # Compute the degree of freedom
  df =  (nrow(obs_tbl)-1) * (ncol(obs_tbl)-1)

  # Print the output
  print(c(chi_squared = chi_squared, df = df), digits = 4)
}

chi_squared(Titanic, Titanic$Class, Titanic$Survived)
```

```
## chi_squared          df
##       190.4         3.0
```

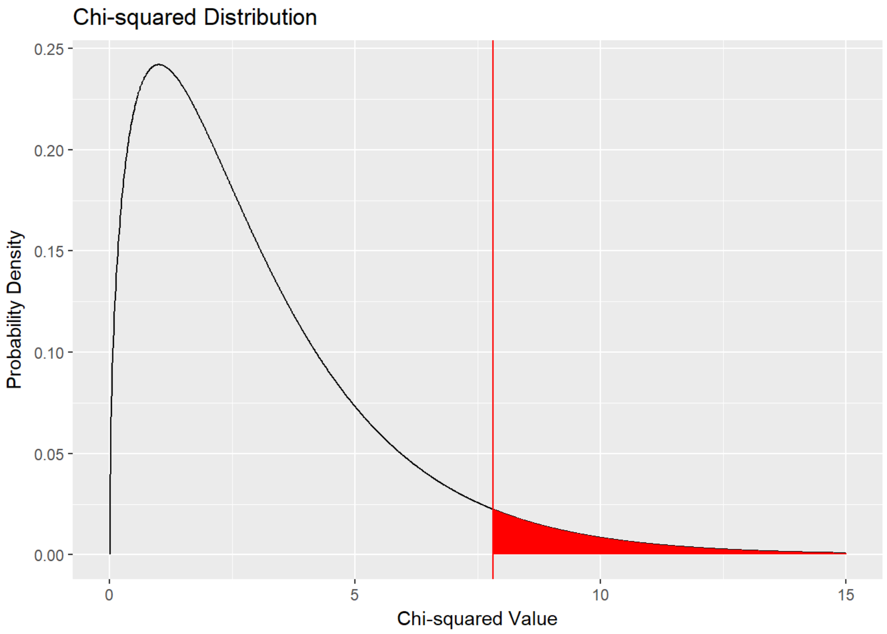
This saves effort since we don’t have to calculate the Chi-squared value every time by hand anymore, especially in cases of huge datasets. With the dataset and the two categorical variables we want to test, we can get the Chi-squared value and the degree of freedom we need to find the critical value and compare the two values to reach a conclusion.

Let’s see a more direct visualization of the result:

```
# Create a plot for the Chi-squared distribution

# Create the data for the distribution
df <- 3
p <- 0.05
distr_data <- data.frame(x=seq(0,15,0.01))
distr_data$y <- dchisq(distr_data$x,df)

# Plot the distribution
ggplot(distr_data) +
  geom_path(aes(x,y)) +
  # exhibit the rejection area
  geom_linerange(data=distr_data[distr_data$x>qchisq(p,df,lower.tail=F), ],
                 aes(x, ymin=0, ymax=y),
                 colour="red") +
  # exhibit the Chi-squared critical value
  geom_vline(xintercept = 7.82, colour = "red") +
  xlab("Chi-squared Value") +
  ylab("Probability Density") +
  ggtitle("Chi-squared Distribution")
```



Note: It is true that the Chi-squared distribution, unlike the normal distribution, is right-skewed since it is always no less than 0 (given the equation of Chi-squared value and the fact that frequencies can’t be negative). The skewness will decrease as the degree of freedom increases.

Interpretation and Discussion

We just plotted a Chi-squared distribution with the path of a qchisq() function with significance level = 0.05 and df = 1. The vertical red line intercepts the x axis with the Chi-squared critical value of 7.82 in this specific case, and the shaded area in red represents the probability of having a Chi-squared value no lower than 7.82, aka. the rejection area. In this case, as we concluded from our previous analysis, since the actual Chi-squared value (190.4) is way larger than the critical value, it falls in the rejection area, and we should reject the null hypothesis.

This means survival on Titanic is actually dependent on the cruise class.

R’s Built-in Approach to Chi-squared Test

In fact, R has a built-in function for Chi-squared test, but it requires us to have a contingency table ready as the obs_tbl we had in the previous example. It works like this:

```
# I will still use the example of survival and class on Titanic

# Built-in Chi-squared test
chisq.test(obs_tbl)
```

```
##
##  Pearson's Chi-squared test
##
## data:  obs_tbl
## X-squared = 190.4, df = 3, p-value < 2.2e-16
```

Discussion

As we can see, from this built-in test, we have gained the same Chi-squared value as we got from the function we created.

An advantage of this built-in test is that it gives you the p-value - the probability of getting a Chi-squared value no smaller than 190.4 assuming the null hypothesis is true, which is extremely small in this case and thus supports the conclusion that we should reject the null hypothesis. In this case, we don’t have to check the Chi-squared table to determine if the Chi-squared value falls in the rejection area as we did previously. The p-value gives us a direct sense of whether we should reject the null hypothesis or not at the significance level of 0.05 and the degree of freedom of 3.

The disadvantage of this built-in test is that we have to convert a dataset into a contingency table first. However, in our second self-created function, we can avoid this step and directly compute the Chi-squared value with the original dataset.

With different functions available, it is really up to our own judgement of which approach to use, depending on the specific situation.

Conclusion and Some Take Home Messages

In this post, I explored Chi-squared test using R.

It is important and can be useful for students of all majors since it allows us to conclude patterns and extract useful information about large sets of data, as well as to test independence among categorical variables. It is also meaningful to explore how to use R specifically to do the testing, since R is a powerful tool to deal with large data sets, which is an advantage that hand-calculation and calculators do not have.

In this post, considering the likelihood that some students may not have taken STAT 20 or AP stats before (and thus may know little about the basics of the Chi-squared test), I briefly introduced the concepts whenever necessary. For those who have already had the knowledge, thanks for bearing with me. If you hope to learn more about Chi-squared test, there are tons of tutorials and materials online, which are quite helpful.

Overall, the post expands into a new field based on the knowledge of building new functions, using new functions, and plotting with ggplots learnt in Stat 133, which is relevant to the course but also explores something new beyond what was covered in the course.

With detailed information on all packages, explanation of code for all functions, tables, graphs and concepts, and interpretation of all the outputs, the post should be computationally reproducible.

References

To make it easier to check the validity of the sources, I will provide a list of the URLs as references.

1. https://www.ma.utexas.edu/users/davis/375/popecol/tables/chisq.html
2. https://onlinecourses.science.psu.edu/statprogram/node/158
3. https://www.rdocumentation.org/packages/datasets/versions/3.4.1/topics/Titanic
4. http://www.cookbook-r.com/Manipulating_data/Converting_between_data_frames_and_contingency_tables/
5. https://stackoverflow.com/questions/21434709/visualize-the-rejection-region-in-a-probability-distribution-curve
6. http://www.r-tutor.com/elementary-statistics/goodness-fit/chi-squared-test-independence
7. http://www.r-tutor.com/elementary-statistics/probability-distributions/chi-squared-distribution
8. https://stat.ethz.ch/R-manual/R-devel/library/base/html/margin.table.html
9. https://stackoverflow.com/questions/29787850/how-do-i-add-a-url-to-r-markdown