post01-ranzhi-xue-Hypothesis_testing_in_R

Ranzhi Xue

October 30, 2017

Introduction to Hypothesis Testing Using R

Introduction

Earlier in the course, we learnt about and conducted some statistical analysis for our NBA data set, including calculating key parameters, plotting diagrams, and looking for correlation between different variables. During the process of learning, I wondered how these features we obtained of the data set (or in other words, the NBA player population) can have real applications. For example, I computed the average height of all NBA players, but what can I do with it? That reminded me of the hypothesis testing for populations I learnt in Stats 20. Therefore, in this post, I would like to explore the steps of doing hypothesis testing using R. Instead of going into various situations, I would focus on hypothesis testing of population mean for a single population. This whole process would also include plotting of some distributions and introduction of some new functions.

Some Background Knowledge

To start off, what is hypothesis testing? According to R Tutorial, it is basically the retainment or rejection of a hypothesis based on measurements of observed samples using a statistical mechanism. Let me help you better understand or refresh your memory with the following example: Suppose a light bulb brand claims its bulbs have a lifetime of on average 10,000 hours with a standard deviation of 120 hours. However, a random sample of 100 consumers reflect that the light bulbs they have purchased last on average only 9,900 hours. At a 0.05 significance level, a hypothesis test can give us hint on whether the brand is lying about the durability of its light bulbs.

Data and Package Preparation

```
# the data frame of a random sample of 40 nba players and their weight
df1 <- data.frame(read.csv(file = "../data/weight-sample-table-for-z-test", stringsAsFactors = FALSE))

# the data frame of a random sample of 20 nba players and their weight
df2 <- data.frame(read.csv(file = "../data/weight-sample-table-for-t-test", stringsAsFactors = FALSE))

library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Note: For this post's purpose, we would assume we don't have the population data - weight of all nba players. The only data we have and will use is the two samples above.

Getting started

There are two types of testing we are going to discuss today: z-test and t-test. When we know the standard deviation of our population and when the sample size is larger than 30, we would use the z-test for the population mean. Otherwise, we would use t-test.

z-test

z-test includes 3 different situations: lower tail test, upper tail test, and two-tailed test. They lead to 3 different alternative hypotheses:

1. mu < mu0: true population mean is lower than hypothesized population mean
2. mu > mu0: true population mean is higher than hypothesized population mean
3. mu != mu0: true population mean differs from hypothesized population mean

Note: In all three cases below, we assume our population is approximately normally distributed.

1) mu < mu0:

Suppose one day, you and your friend were talking about nba after Stat 133 class. Your friend claimed that all nba players on average weigh about 226 pounds (mu0 = 226) with a standard deviation of 26 pounds (sigma = 26). You doubted it, so after going home, you randomly checked 40 players' weight data and they are shown as below:

```
df1
```

```
##       X                  player weight
## 1   127           E'Twaun Moore    191
## 2   347          Raymond Felton    205
## 3   180    James Michael McAdoo    230
## 4   387            Steven Adams    255
## 5   411           Trevor Booker    228
## 6    20         Andrew Harrison    213
## 7   230         Justise Winslow    225
## 8   388          T.J. McConnell    200
## 9   239           Kent Bazemore    201
## 10 198            Jimmy Butler    220
## 11 413              Trey Lyles    234
## 12 195              Jeremy Lin    200
## 13 291        Maurice Harkless    215
## 14 246           Klay Thompson    215
## 15  44          Brandon Ingram    190
## 16 384         Stanley Johnson    245
## 17 105        Denzel Valentine    212
## 18  18          Andre Iguodala    215
## 19 139          Frank Kaminsky    242
## 20 403 Timothe Luwawu-Cabarrot    205
## 21 375          Shabazz Napier    175
## 22 429         Wayne Ellington    200
## 23 269            Luke Babbitt    225
## 24 416           Troy Williams    218
## 25 274           Malik Beasley    196
## 26 295         Michael Beasley    235
## 27 226           Julius Randle    250
## 28 428            Wade Baldwin    202
## 29 120     Dorian Finney-Smith    220
## 30  61        Chandler Parsons    230
## 31 396         Thomas Robinson    237
## 32 370          Semaj Christon    190
## 33 283         Markieff Morris    245
## 34 325             Otto Porter    198
## 35  11           Alex Abrines    190
## 36 194            Jeremy Lamb    185
## 37 308           Myles Turner    243
## 38  88          Darrell Arthur    235
## 39 129             Edy Tavares    260
## 40  94           Davis Bertans    210
```

```r
# compute the sample mean
x_bar <- mean(df1$weight)
x_bar
```

```
## [1] 217.125
```

After browsing through the sample data table, your instinct may tell you that your friend very likely overestimated the population mean, but how can you prove his/her claim is wrong? Here, with known population sd and sample size over 30, a z-test can give you a direct test of your instinct. Since R does not have a direct function for z-test, we will be making our own function:

H0: mu = 226 (Null hypothesis)

H1: mu < 226 (Alternative hypothesis)

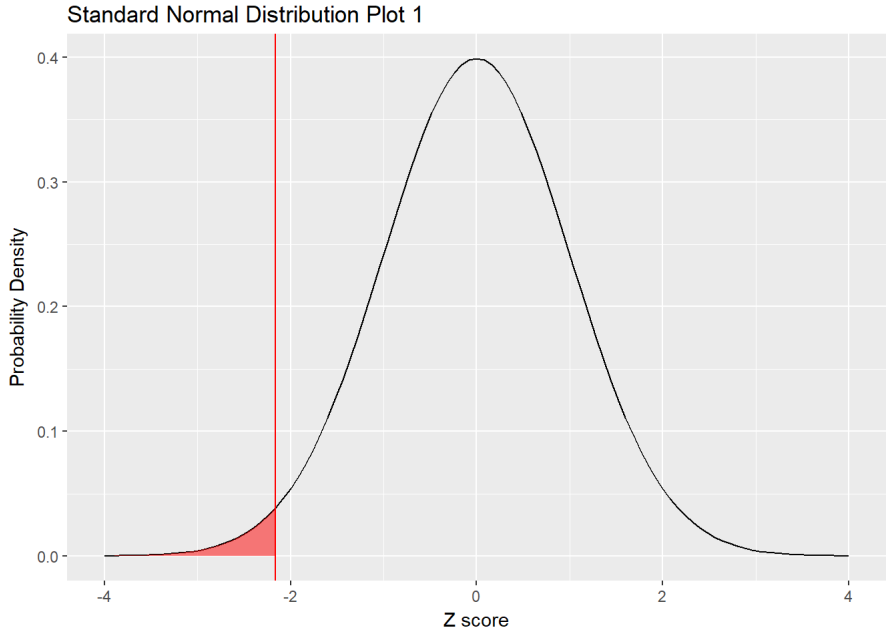
```r
# create the z-test function
n <- 40
mu0 <- 226
sigma <- 26

z.test <- function(x_bar, mu0, sigma) {
  z.score = round((x_bar - mu0) / (sigma / sqrt(n)), 3)
  p_value = round(pnorm(abs(z.score), lower.tail = FALSE), 3)
  cat("z.score =", z.score, "\n",
      "p_value =", p_value)
}

z.test(x_bar, mu0, sigma)
```

```
## z.score = -2.159
##  p_value = 0.015
```

```r
# The pnorm() function with lower.tail = FALSE calculates P[X > x],
# in this case P[z > 2.159] (which is equivalent to P[z < -2.159] given the normal distribution)
# The cat() function prints the results in an ideal way with formatting
```

The z-test function we just created allows us to apply z-test to other populations with different parameters directly in the future, which saves effort since we don't have to compute the parameters every time by hand anymore. Let's see a more direct visualization of the result:

```
# create a standard normal distribution plot
zplot1 <- ggplot(
  data = data.frame(x = c(-4, 4)), aes(x)) +
  stat_function(fun = dnorm,
                args = list(mean = 0, sd = 1)) +
  ylab("Probability Density") +
  xlab("Z score") +
  ggtitle("Standard Normal Distribution Plot 1") +
  # exhibit the z score
  geom_vline(xintercept = -2.159, colour = "red") +
  # exhibit the p-value
  stat_function(fun = dnorm,
                xlim = c(-4, -2.159),
                geom = "area",
                fill = "red",
                alpha = 0.5)
zplot1
```



Explanation and Interpretation of the graph: We plot a standard normal distribution by plotting the path of a dnorm() function with mean = 0 and sd = 1, using stat_function() in ggplot. The vertical red line intercepts the x axis with the z score -2.159, and the shaded area in red represents the probability of having a z score of no higher than -2.159 assuming the null hypothesis is true, aka. the p-value. In this case, from previous calculation, we know the probability is 0.015, which is quite low.

Conclusion: At a 0.05 significance level, since 0.015 < 0.05, we would reject the null hypothesis that the population mean of weight of nba players is 226 pounds. In other words, it is very unlikely to get a sample mean of 217.125 pounds in a random sample of 40 if the claimed the population mean is true. Therefore, there may be something wrong with your friend's claim. It's time to go back and ask your friend where he/she got the number.

2) mu > mu0:

Similarly, if we change the claimed population mean to 208 pounds while everything else stays the same, you might be wondering if your friend is underestimating the mean weight of all nba players. In this case, we would be using the alternative hypothesis: mu > mu0.

H0: mu = 208 (Null hypothesis)

H1: mu > 208 (Alternative hypothesis)

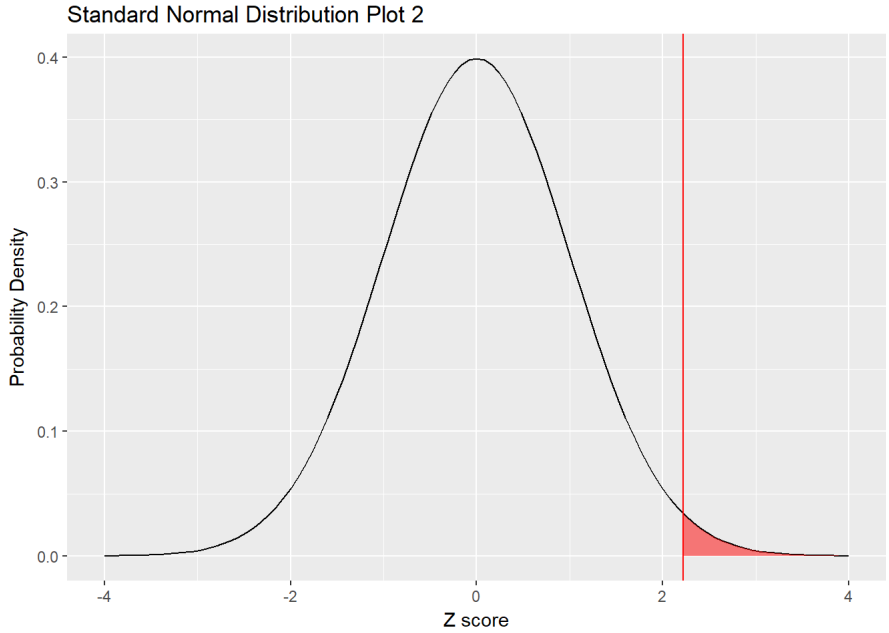
The z-test function created earlier can now be applied directly:

```
# applying the z-test
n <- 40
mu0 <- 208
sigma <- 26

z.test(x_bar, mu0, sigma)
```

```
## z.score = 2.22
##  p_value = 0.013
```

Again, here is the visualization of the result:

```
# create a standard normal distribution plot
zplot2 <- ggplot(
  data = data.frame(x = c(-4, 4)), aes(x)) +
  stat_function(fun = dnorm,
                args = list(mean = 0, sd = 1)) +
  ylab("Probability Density") +
  xlab("Z score") +
  ggtitle("Standard Normal Distribution Plot 2") +
  # exhibit the z score
  geom_vline(xintercept = 2.22, colour = "red") +
  # exhibit the p-value
  stat_function(fun = dnorm,
                xlim = c(2.22, 4),
                geom = "area",
                fill = "red",
                alpha = 0.5)
zplot2
```



Explanation and Interpretation of the graph: Similarly, the vertical red line intercepts the x axis with the z score 2.22, and the shaded area in red represents the probability of having a z score of no lower than 2.22 assuming the null hypothesis is true, aka. the p-value. In this case, from previous calculation, we know the probability is 0.013, which is quite low.

Conclusion: At a 0.05 significance level, since 0.013 < 0.05, we would reject the null hypothesis that the population mean of weight of nba players is 208 pounds. In other words, it is very unlikely to get a sample mean of 217.125 pounds in a random sample of 40 if the claimed the population mean is true.

3) mu != mu0:

The two-tailed test deals with situations of inequality. Instead of telling whether the true population mean is lower or higher than the hypothesized population mean, we simply wonder if we can reject the null hypothesis that the actual mean weight does not differ from the hypothesized mean weight. Assume everything stays the same, and we still have a hypothesized population mean of 208 pounds. This time, the z-test function will be a bit different given the two-tailed feature:

H0: mu = 208 (Null hypothesis)

H1: mu != 208 (Alternative hypothesis)

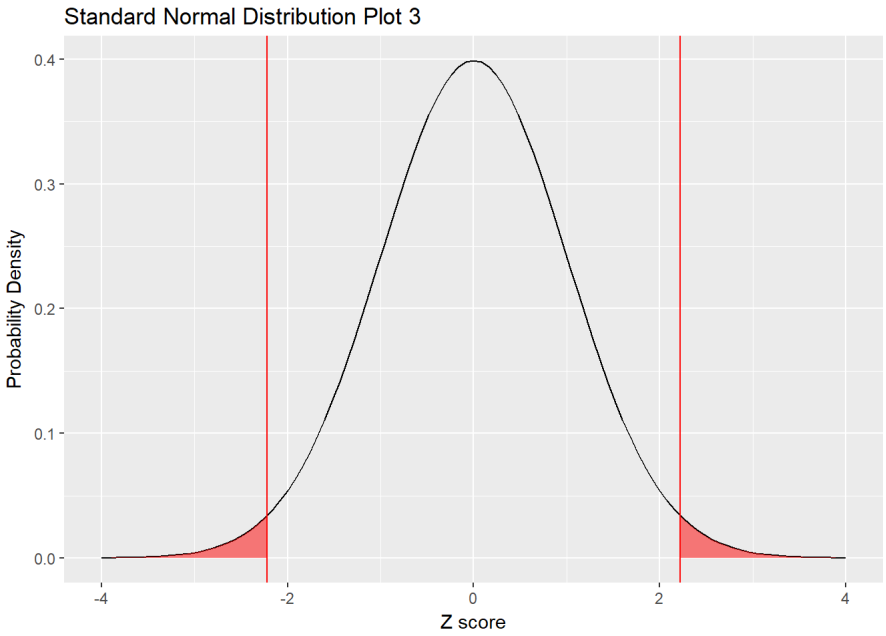
```
n <- 40
mu0 <- 208
sigma <- 26

z.test <- function(x_bar, mu0, sigma) {
  z.score = round((x_bar - mu0) / (sigma / sqrt(n)), 3)
  p_value = round(2 * pnorm(abs(z.score), lower.tail = FALSE), 3)
  cat("z.score =", z.score, "\n",
      "p_value =", p_value)
}

z.test(x_bar, mu0, sigma)
```

```
## z.score = 2.22
##  p_value = 0.026
```

```
# We have to add together the probabilities on each side of the distribution to get the p-value.
# In other words, we have to multiply our previous p-value by 2, given that it is a two-tailed test.
```

Here is the visualization of the result:

```
# create a standard normal distribution plot
zplot3 <- ggplot(
  data = data.frame(x = c(-4, 4)), aes(x)) +
  stat_function(fun = dnorm,
                args = list(mean = 0, sd = 1)) +
  ylab("Probability Density") +
  xlab("Z score") +
  ggtitle("Standard Normal Distribution Plot 3") +
  # exhibit the z score
  geom_vline(xintercept = c(-2.22, 2.22), colour = "red") +
  # exhibit the p-value
  stat_function(fun = dnorm,
                xlim = c(2.22, 4),
                geom = "area",
                fill = "red",
                alpha = 0.5) +
  stat_function(fun = dnorm,
                xlim = c(-4, -2.22),
                geom = "area",
                fill = "red",
                alpha = 0.5)
zplot3
```



Explanation and Interpretation of the graph: The vertical red lines intercept the x axis with the z scores -2.22 and 2.22, and the total shaded area in red represents the probability of having a z score of no higher than -2.22 or no lower than 2.22 assuming the null hypothesis is true, aka. the p-value. In this case, from previous calculation, we know the probability is 0.026.

Conclusion: At a 0.05 significance level, since 0.026 < 0.05, we still reject the null hypothesis that the true population mean of weight of nba players does not differ from 208 pounds. In other words, it is very unlikely to get a sample mean of 217.125 pounds in a random sample of 40 if the true population mean does not differ from the hypothesized value.

t-test

Previously, we have dealt with cases where the population standard deviation is known and the sample size is larger than 30. What should we do in cases where either one of them is not fulfilled, or neither of them is fulfilled? That's when we use the t-test.

Similar to the z-test, the t-test also includes lower tail test, upper tail test, and two-tailed test, and the processes of analysis are very similar. Here, I will use the lower tail test as an example:

Suppose on another day, you and another friend were talking about nba after Stat 133 class. Your friend, coincidentally, also claimed that all nba players on average weigh about 226 pounds (mu0 = 226), but didn't give any information on the standard deviation. You doubted it, so after going home, you randomly checked 20 players' weight data and they are shown as below:

```
df2
```

```
##       X                  player weight
## 1   329          Patricio Garino    210
## 2   344               Randy Foye    213
## 3     9             Alan Williams    260
## 4   340                Quinn Cook    184
## 5    30             Avery Bradley    180
## 6   282             Mario Hezonja    215
## 7   405          Tomas Satoransky    210
## 8   312                Nick Young    210
## 9   402            Timofey Mozgov    275
## 10  123           Draymond Green    230
## 11  240 Kentavious Caldwell-Pope    205
## 12  236              Kelly Olynyk    238
## 13  251             Kyle Anderson    230
## 14  250        Kristaps Porzingis    240
## 15    1              A.J. Hammons    260
## 16  188              JaVale McGee    270
## 17  134              Eric Bledsoe    190
## 18  314          Nicolas Brussino    195
## 19   59              Caris LeVert    203
## 20  368             Sasha Vujacic    195
```

```r
# compute the sample mean
x_bar <- mean(df2$weight)
x_bar
```

```
## [1] 220.65
```

Again, you want to check if you should reject the null hypothesis that the true population mean of weight is equal to the hypothesized mean of 226, as you doubt your friend is overestimating the population mean. In this case, we would use a t-test.

H0: mu = 226 (Null hypothesis)

H1: mu < 226 (Alternative hypothesis)

Unlike z-test, R has a built-in function for t-test, which allows us to compute it directly:

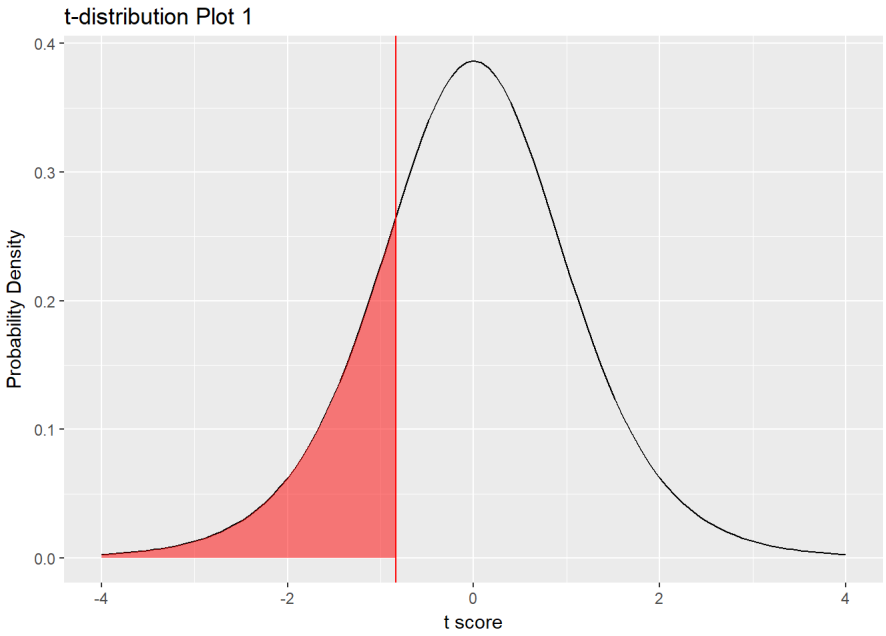
```r
t.test(df2$weight, alternative = "less", mu = 226, conf.level = 0.95)
```

```
##
##  One Sample t-test
##
## data:  df2$weight
## t = -0.83568, df = 19, p-value = 0.2069
## alternative hypothesis: true mean is less than 226
## 95 percent confidence interval:
##      -Inf 231.7199
## sample estimates:
## mean of x
##    220.65
```

As we can see from the result, the t score is -0.83568, and the p value is 0.2069.

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

```r
# create a t-distribution plot
tplot1 <- ggplot(
  data = data.frame(x = c(-4, 4)), aes(x)) +
  stat_function(fun = dt,
                args = list(df = 8)) +
  ylab("Probability Density") +
  xlab("t score") +
  ggtitle("t-distribution Plot 1") +
  # exhibit the t score
  geom_vline(xintercept = -0.83568, colour = "red") +
  # exhibit the p-value
  stat_function(fun = dt,
                args = list(df = 8),
                xlim = c(-4, -0.83568),
                geom = "area",
                fill = "red",
                alpha = 0.5)
tplot1
```



Explanation and Interpretation of the graph: Similarly, the vertical red line intercepts the x axis with the t score -0.83568, and the shaded area in red represents the probability of having a t score of no higher than -0.83568 assuming the null hypothesis is true, aka. the p-value. In this case, from previous calculation, we know the probability is 0.2069, which is relatively high.

Conclusion: At a 0.05 significance level, since 0.2069 > 0.05, we do not reject the null hypothesis that the population mean of weight of nba players is 226 pounds. In other words, it is not unlikely to get a sample mean of 220.65 pounds in a random sample of 20 if the claimed the population mean is true.

Conclusion and Some Take Home Messages

In this post, I explored hypothesis testing using R, specifically z-test and t-test. 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 ideas and hypotheses. It is also meaningful to explore how to use R specifically to do these testings, 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 hypothesis testing), 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 hypothesis testing, 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 functions, ggplots, and data manipulation learnt in Stat 133, which is relevant to the course but also explores something new.

References

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

1. http://www.r-tutor.com/elementary-statistics/hypothesis-testing
2. http://web.science.mq.edu.au/~mjohnson/papers/Johnson14-02HT-talk.pdf
3. http://www.dummies.com/education/math/statistics/z-testing-r/
4. https://en.wikibooks.org/wiki/Statistics/Testing_Data/z-tests
5. https://stackoverflow.com/questions/10488988/making-a-standard-normal-distribution-in-r
6. https://sebastiansauer.github.io/normal_curve_ggplot2/
7. https://stackoverflow.com/questions/33244629/filling-under-the-a-curve-with-ggplot-graphs
8. https://sebastiansauer.github.io/shade_Normal_curve/
9. https://www.rdocumentation.org/packages/stats/versions/3.4.1/topics/t.test
10. http://t-redactyl.io/blog/2016/03/creating-plots-in-r-using-ggplot2-part-9-function-plots.html