Data Analysis for a Fictional Trial of Dietary Supplements

Tim Liptrot

Contents

Introduction

This code is meant as a brief demonstration of my R coding skills in hypothesis testing, with some data cleaning and visualization. It is based on data provided in an employment tests by a development econometrics organization as an employment test.

Problem Statement

A company marketing dietary supplement has carried out an experiment on the relationship between taking their supplements and formal reasoning. Study participants took one of four supplements (or none) and executed a series of mental math problems. The company would like to know if their data shows that the supplements make people smarter. Unfortunately, their data is not neatly organized. For example, the time is mm.ss format.

Packages and data loading

In this demonstration, I will use the Tidyverse library. To learn more about the packages, I recommend Wickham and Grolemund 2016, "R for Data Science".

```
library(tidyverse)
library(ggthemes)
library(kableExtra)
library(tinytex)

setwd("C:\\Users\\liptr\\Documents\\R\\vitamins")
vit <- read_csv("vitamins\\vitamins.csv")
```

Data cleaning

First, let us take a look at this data.

```
vit
```

```
## # A tibble: 500 x 3
##    treat  time supplement
##    <dbl> <dbl> <chr>
##  1     1  2.35 B
##  2     1  4.13 B
##  3     0  5.26 <NA>
##  4     1  3.45 B
##  5     1  3.29 A
##  6     1  5.36 C
##  7     0  2.41 <NA>
##  8     1  3.43 B
##  9     0  2.32 <NA>
## 10     1  3.4  c
## # ... with 490 more rows
```

I have two glaring problems in the data set. One is that the time isn in mm.ss, rather than number of seconds, which would be easier to analyze. Secondly, there are several misnamed treatments (with names like b, n, r, 1 and 2). Additionally, it is reduntant to have one variable labeled treatment and one variable for the type of treatment. I will later collapse them into one file with control, A, B, C and D as the variables.

The Time Variable

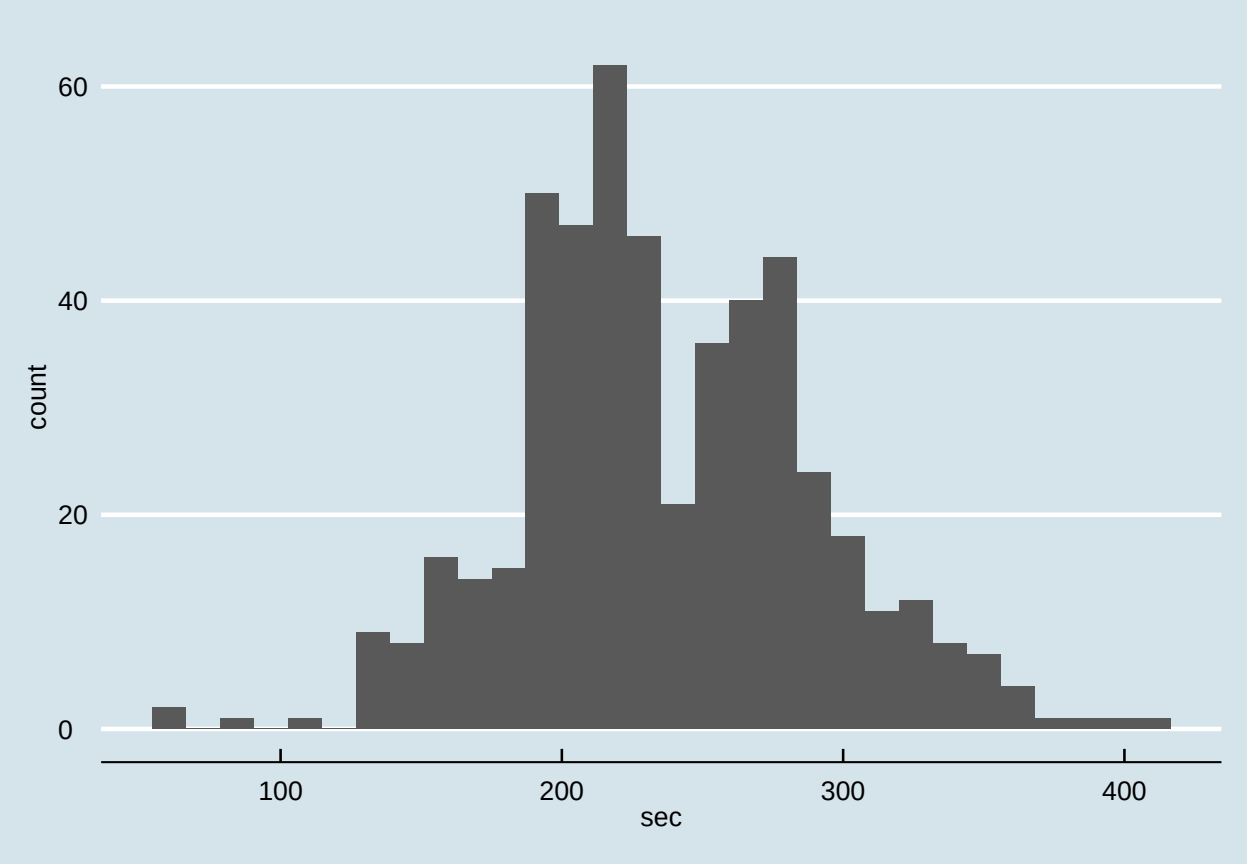
First I calculate the new time variable.

```
vit <- mutate(vit,
       min = time%/%1,
       sec = (time - min)*100 + 60*min,
       position = rownames(vit))

#print(filter(vit, position=))
```

It's important to check that this formula is working correctly. The easiest way to do so is to make a quick histogram of the times.

```
p <- ggplot(vit, mapping = aes(sec))
p + geom_histogram() + theme_economist() + scale_fill_economist()
```



Since there are no suspicious gaps or shapes in this histogram, it looks like my equation is working just fine.

Cleaning the Treatment Categories

Now let us look at our labeling of the subjects. Firstly, there is a redundancy between the treatment column, which shows a binary for whether the subject is in the control group, and the supplement column. I simplify this by simply changing the sup category to "Control" for all subjects in the control group. I also rename that variable sup, to shorten the code later.

Then I count all of the different values in the sup variable.

```
vit <- rename(vit, sup = supplement)

vit <- mutate(vit, sup = replace(sup, treat ==0, "CONTROL"))

count(vit, sup)
```

```
## # A tibble: 13 x 2
##    sup         n
##    <chr>   <int>
##  1 2           1
##  2 3           3
##  3 a           6
##  4 A          34
##  5 b          35
##  6 B          47
```

```
##  7 c          15
##  8 C          60
##  9 CONTROL   260
## 10 d          10
## 11 D          25
## 12 n           3
## 13 r           1
```

From this count we can see there are many poorly labeled treatments. Most treaments are labeled with a single capital letter, but some are lower case, some are numbers, and some are even other random letters (n and r). To solve this, I remove all spaces and uppercase all the letters, then remove any unclear values. Then I rename them Control, and Treatment_X, to make the final outputs easier to read.

```
vit <- mutate(vit,
           sup = str_to_upper(sup),
           sup = str_trim(sup)
           )

vit <- filter(vit, sup == "A" | sup == "B" | sup == "C" | sup == "D" | sup == "CONTROL")

vit <- mutate(vit,
              sup = replace(sup, treat ==0, "Control"),
              sup = replace(sup, sup == "A", "Treatment_A"),
              sup = replace(sup, sup == "B", "Treatment_B"),
              sup = replace(sup, sup == "C", "Treatment_C"),
              sup = replace(sup, sup == "D", "Treatment_D")
              )
```

Checking for Duplicates

I always produce a unique observation id for each entry in a survey, often a housheold ID or respondent ID. Unfortunately, this experimenter did not create an id for every observation (respondent in this case). This makes it impossible to be sure I have found all duplicate observations, since two subjects could have received the same treatment and used the same number of seconds by chance. However, I can check for consecutive duplicates with the following code that finds consecutive duplicates then prints those observations.

```
vit<- mutate(vit,
             dup = FALSE,
             dup = replace(dup, sup==lead(sup) & sec==lead(sec), TRUE))

print(filter(vit, dup == TRUE | lag(dup==TRUE)))
```

```
## # A tibble: 2 x 7
##   treat  time sup       min   sec position dup
##   <dbl> <dbl> <chr>   <dbl> <dbl> <chr>    <lgl>
## 1     0  3.49 Control     3  229. 337      TRUE
## 2     0  3.49 Control     3  229. 338      FALSE
```

GOod, only one value is a duplicate. But is this value a duplicate? Let's check the original values.

```
vit2 <- read_csv("vitamins\\vitamins.csv")
```

```
## Parsed with column specification:
## cols(
##   treat = col_double(),
##   time = col_double(),
##   supplement = col_character()
## )
```

```
print(filter(vit2, rownames(vit2) == 337 | rownames(vit2) == 338))
```

```
## # A tibble: 2 x 3
##   treat  time supplement
##   <dbl> <dbl> <chr>
## 1     0  3.49 <NA>
## 2     0  3.49 <NA>
```

These two observations are suspicious, as it woudl be easy for an enumerator to accidentally enter the same value twice. But since these enumerators are fictional, I cannot call them up and ask about the observations. Therefore, I will keep it in the data.

After cleaning, out cleaned data looks like this.

```
vit
```

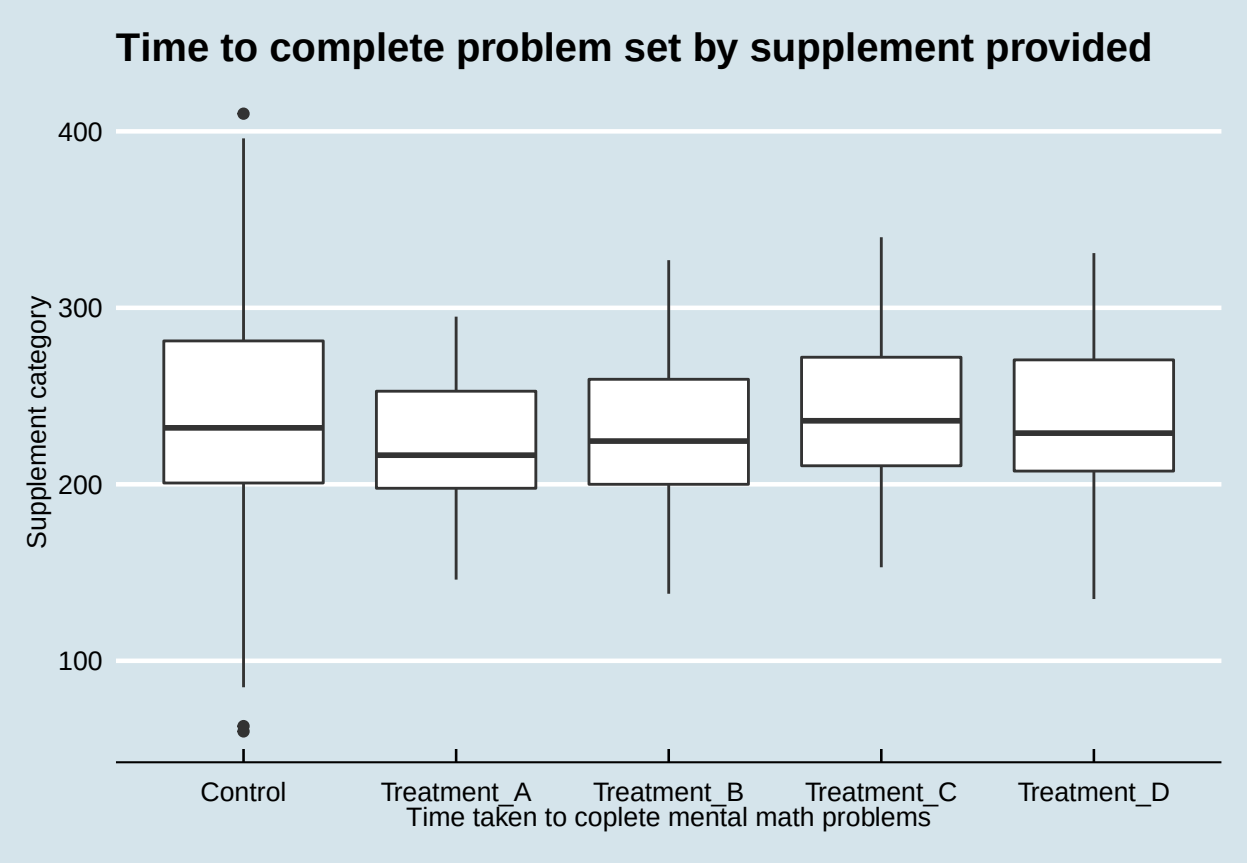
```
## # A tibble: 492 x 7
##    treat  time sup           min   sec position dup
##    <dbl> <dbl> <chr>       <dbl> <dbl> <chr>    <lgl>
##  1     1  2.35 Treatment_B     2   155 1        FALSE
##  2     1  4.13 Treatment_B     4   253 2        FALSE
##  3     0  5.26 Control         5   326 3        FALSE
##  4     1  3.45 Treatment_B     3   225 4        FALSE
##  5     1  3.29 Treatment_A     3   209 5        FALSE
##  6     1  5.36 Treatment_C     5   336 6        FALSE
##  7     0  2.41 Control         2   161 7        FALSE
##  8     1  3.43 Treatment_B     3   223 8        FALSE
##  9     0  2.32 Control         2   152 9        FALSE
## 10     1  3.4  Treatment_C     3   220 10       FALSE
## # ... with 482 more rows
```

Data Analysis

Visualizing the Results

Now that the data is clean, I can begin analysis. But, before begining any formal tests of significance, it is valuable to visualize our results. This will help us to check that everything is running smoothly and to contextualize any findings. Because this data set is quite narrow, a simple boxplot provides the most important information.

```
g <- ggplot(vit, mapping = aes(sup, sec))
g + geom_boxplot() +
  labs(title="Time to complete problem set by supplement provided", x="Time taken to coplete mental mat
  theme_economist() + scale_fill_economist()
```



Just from reading the boxplot, I can make a few preliminary conclusions.

1. Subjects who took supplements A, B, and D all performed the test faster than subjects who took the control. In the next section, I will determine if this result is significant.
2. Subjects who took supplement C actually performed the mental math more slowly. I know that our test will not present evidence that it is effective.
3. The variance in the control group seems to higher than the treatment groups. Several participants in the control group completed the trial in less than 50 seconds. This may suggest problems in the experimental design, and would normally be reason to contact the enumerators.

Testing for Signficance

To test the effectiveness of the supplements, I use an inedependent two-sample t-test. By convention, I set my signficance threshold a to $\alpha = .05$. Each t-test compares a supplement, independent of the others, to our control group.

```
attach(vit)
```

```
t.test(sec[sup=="Control"], sec[sup=="Treatment_A"], alternative = "less", data = vit)
##
##  Welch Two Sample t-test
##
## data:  sec[sup == "Control"] and sec[sup == "Treatment_A"]
## t = 2.4718, df = 74.33, p-value = 0.9921
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 29.55953
## sample estimates:
## mean of x mean of y
##  240.7346  223.0750
t.test(sec[sup=="Control"], sec[sup=="Treatment_B"], alternative = "less", data = vit)
##
##  Welch Two Sample t-test
##
## data:  sec[sup == "Control"] and sec[sup == "Treatment_B"]
## t = 1.874, df = 204.54, p-value = 0.9688
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 20.84217
## sample estimates:
## mean of x mean of y
##  240.7346  229.6585
t.test(sec[sup=="Control"], sec[sup=="Treatment_C"], alternative = "less", data = vit)
##
##  Welch Two Sample t-test
##
## data:  sec[sup == "Control"] and sec[sup == "Treatment_C"]
## t = -0.0019088, df = 167.2, p-value = 0.4992
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 10.43069
## sample estimates:
## mean of x mean of y
##  240.7346  240.7467
t.test(sec[sup=="Control"], sec[sup=="Treatment_D"], alternative = "less", data = vit)
##
##  Welch Two Sample t-test
##
## data:  sec[sup == "Control"] and sec[sup == "Treatment_D"]
## t = 0.50729, df = 52.518, p-value = 0.693
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 18.64161
## sample estimates:
## mean of x mean of y
##  240.7346  236.4000
```

For each t-test there is a hypothesis and a null hypothesis. The hypothesis is that the supplements improved mental math speed (reduced the number of seconds) and the null hypothesis is that there was no effect. The quickest number to look at here is the p-value, which is that chance of these results occuring if the null value is true.

Supplements A and B both have p-values above .95. This means these results would be unlikely (occuring in only one int twenty trials), if the supplement was not improving math ability. It does not prove the supplements are effective, but it suggests that they are. That these results are also dependent n the fairness fo the trial and soundness of the design.

For C and D, I cannot eliminate the null hypothesis that the supplements have no effect. It is unlikely, but not impossible, that this result was an aberration of our study and future, more detailed work would find a different result.