Problem set 5: Worked solutions

Statistics and statistical programming
Northwestern University
MTS 525

Aaron Shaw (with Jeremy Foote)

October 26, 2020

Contents

Programming Challenges

PC1. Import

You had the option of downloading the dataset in several formats, including a Stata 13 ".dta" file. For demonstration purposes, I'll use that proprietary format here. I also uploaded a copy of the dataset to the course website for pedagogical purposes (so you can replicate my code here on your own machine).

There are a few ways to import data from Stata 13 files. One involves using the appropriately named `readstata13` package. Another uses the `haven` package (more of a general purpose tool for importing data from binary/proprietary formats). I'll go with the `read_dta()` command from `haven` here because it works better with importing from a URL.

```
library(haven)

df <- read_dta(url("https://communitydata.science/~ads/teaching/2020/stats/data/week_07/Halloween2012-20
```

PC2. Explore and cleanup

```
head(df)
```

```
## # A tibble: 6 x 7
##   obama fruit  year   age  male  neob treat_year
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>      <dbl>
## 1     0     0  2014     6     0     1          4
## 2     0     1  2014     5     0     1          4
## 3     0     0  2014     9     1     1          4
## 4     0     0  2014     5     1     1          4
## 5     0     0  2014     7     0     1          4
## 6     0     0  2014     9     0     1          4
```

```
summary(df)
```

```
##      obama            fruit             year           age
##  Min.   :0.0000   Min.   :0.0000   Min.   :2012   Min.   : 2.00
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:2014   1st Qu.: 6.00
##  Median :0.0000   Median :0.0000   Median :2015   Median : 8.00
##  Mean   :0.3639   Mean   :0.2512   Mean   :2014   Mean   : 8.52
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:2015   3rd Qu.:11.00
##  Max.   :1.0000   Max.   :1.0000   Max.   :2015   Max.   :19.00
##                   NA's   :1
##       male              neob          treat_year
##  Min.   :0.0000   Min.   :0.0000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:3.000
##  Median :1.0000   Median :1.0000   Median :5.000
##  Mean   :0.5262   Mean   :0.6361   Mean   :4.406
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:6.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :6.000
##  NA's   :1
```

There are a few noteable things about the dataset. One is the `neob`, which the codebook says means "not equal to obama"; in other words, it's the converse of the `obama` column. The `treat_year` column is a unique index of the `obama` column and the `year` column. See the codebook for more information. Happily, we only need to use the first two columns for now.

I'll drop everything else and convert those two columns into logical vectors here using a couple of tidyverse `dplyr` commands:

```
library(tidyverse)

df <- df %>%
  select(obama, fruit) %>%
  mutate(
    obama = as.logical(obama),
    fruit = as.logical(fruit)
  )

head(df)
```

```
## # A tibble: 6 x 2
##   obama fruit
##   <lgl> <lgl>
## 1 FALSE FALSE
## 2 FALSE TRUE
## 3 FALSE FALSE
## 4 FALSE FALSE
## 5 FALSE FALSE
## 6 FALSE FALSE
```

PC3. Summarize key variables

I'll run summary again and then make my contingency table.

```r
summary(df)
```

```
##    obama           fruit        
##  Mode :logical   Mode :logical  
##  FALSE:778       FALSE:915      
##  TRUE :445       TRUE :307      
##                  NA's :1        
```

```r
obama.tbl <- table(took_fruit = df$fruit, saw_flotus = df$obama)
obama.tbl
```

```
##           saw_flotus
## took_fruit FALSE TRUE
##      FALSE   593  322
##      TRUE    185  122
```

PC4. Test for differences between groups

So, the test we want to conduct here focuses on the difference between the proportion of the two groups (those shown a picture of Michelle Obama vs those not shown a picture of Michelle Obama) who took fruit vs. candy. Labeling the difference in proportions as Δ_{fruit}, the comparison can be constructed around the following (two-sided) hypothesis test

$$H_0: \ \Delta_{fruit} = 0$$
$$H_A: \ \Delta_{fruit} \neq 0$$

As discussed at length in *OpenIntro* Chapter 6, a great way to determine if two groups are independent (in terms of proportions or counts) is a χ^2 test.

Are the conditions for a valid test met? It seems so, since there are many observations in each cell of the table being compared and the observations appear to have been collected in a way that ensures independence.

The χ^2 test is implemented as `chisq.test()` in R. Since it's a 2x2 comparison, we can also test for a difference in proportions using the `prop.test()` function. Let's do both.

```r
chisq.test(obama.tbl)
```

```
## 
##  Pearson's Chi-squared test with Yates' continuity correction
## 
## data:  obama.tbl
## X-squared = 1.8637, df = 1, p-value = 0.1722
```

```r
prop.test(obama.tbl)
```

```
## 
##  2-sample test for equality of proportions with continuity correction
## 
## data:  obama.tbl
## X-squared = 1.8637, df = 1, p-value = 0.1722
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.01957437  0.11053751
## sample estimates:
##    prop 1    prop 2 
## 0.6480874 0.6026059
```

Notice that both functions report identical χ^2 test results and p-values. Did you expect this based on the *OpenIntro* reading?

Recall that if you want to double-check that p-value, you can also calculate it "by-hand" using the `pchisq()` function:

```
pchisq(1.8637, df = 1, lower.tail = FALSE)
```

```
## [1] 0.1721984
```

We'll discuss the interpretation of these results in our class session this week.

PC5. Replicate a figure

In order to replicate the top panel of Figure 1, we'll first want to calculate the proportion and standard error for fruit-takers in the treatment and control groups. These can be calculated individually or using a function (guess which one we'll document here). Also, note that I'm going to use the `complete.cases()` function to eliminate the missing items for the sake of simplicity.

```
df <- df[complete.cases(df), ]

prop.se <- function(values) { # Takes in a vector of T/F values
  N <- length(values)
  prop <- mean(as.numeric(values))
  se <- sqrt(prop * (1 - prop) / N) ## textbook formula for SE of a proportion
  return(c(prop, se))
}

prop.se(df$fruit[df$obama])
```

```
## [1] 0.27477477 0.02118524
```

```
prop.se(df$fruit[!df$obama])
```

```
## [1] 0.23778920 0.01526314
```

In order to graph that it will help to convert the results into a data frame with clearly-labeled variable names and values:

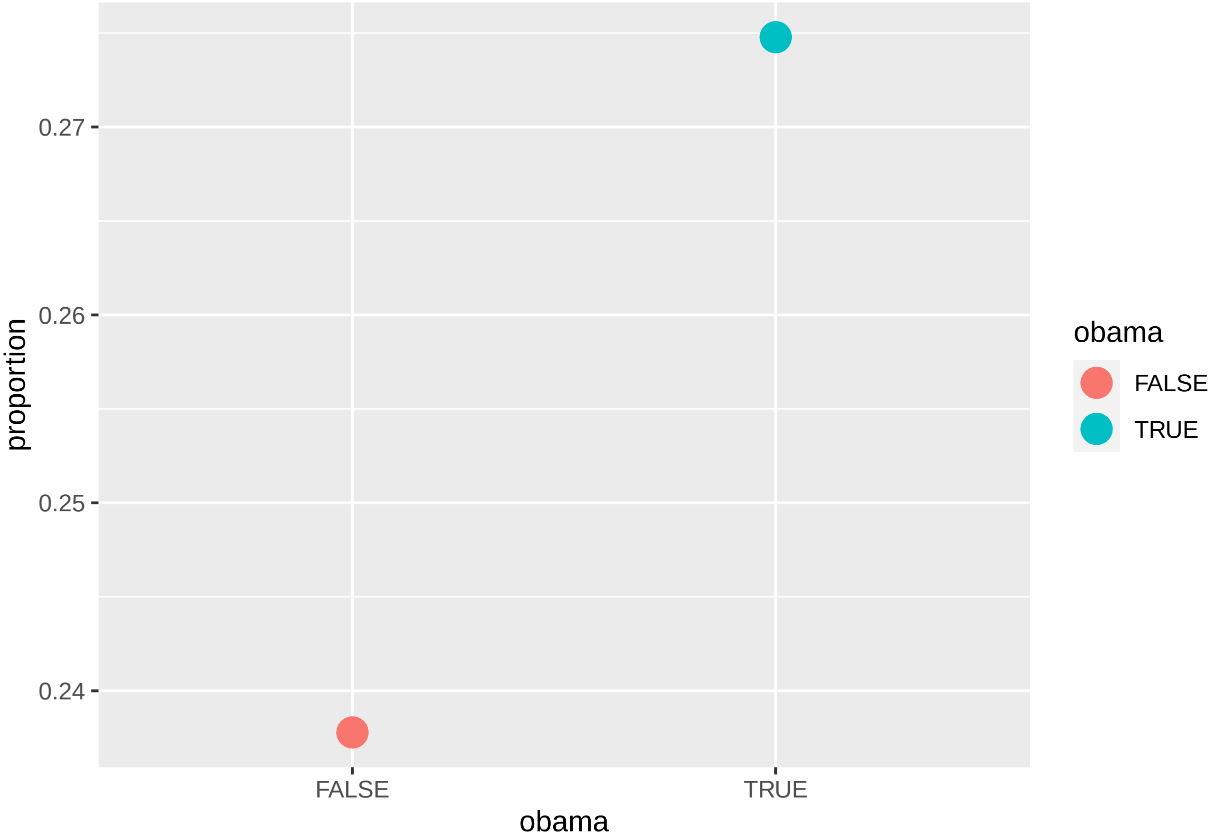
```
prop.and.se <- data.frame(
  rbind(
    prop.se(df$fruit[df$obama]),
    prop.se(df$fruit[!df$obama])
  )
)

names(prop.and.se) <- c("proportion", "se")
prop.and.se$obama <- c(TRUE, FALSE)

prop.and.se
```

```
##   proportion         se obama
## 1  0.2747748 0.02118524  TRUE
## 2  0.2377892 0.01526314 FALSE
```

Now we can start to build a visualization

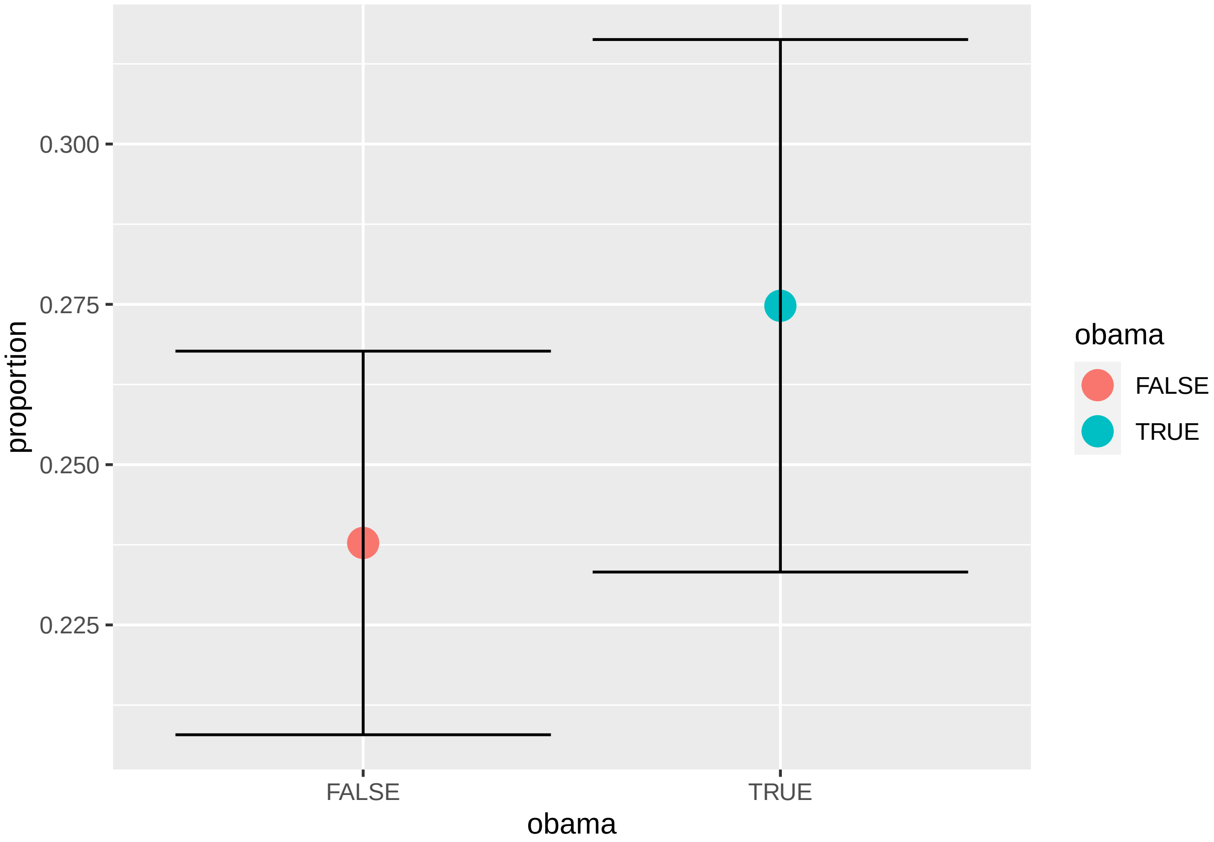
```
## library(ggplot2)  ## already imported w the tidyverse
```

```
p <- ggplot(prop.and.se, aes(x = obama, y = proportion)) +
  geom_point(aes(color = obama), size = 5)

p
```



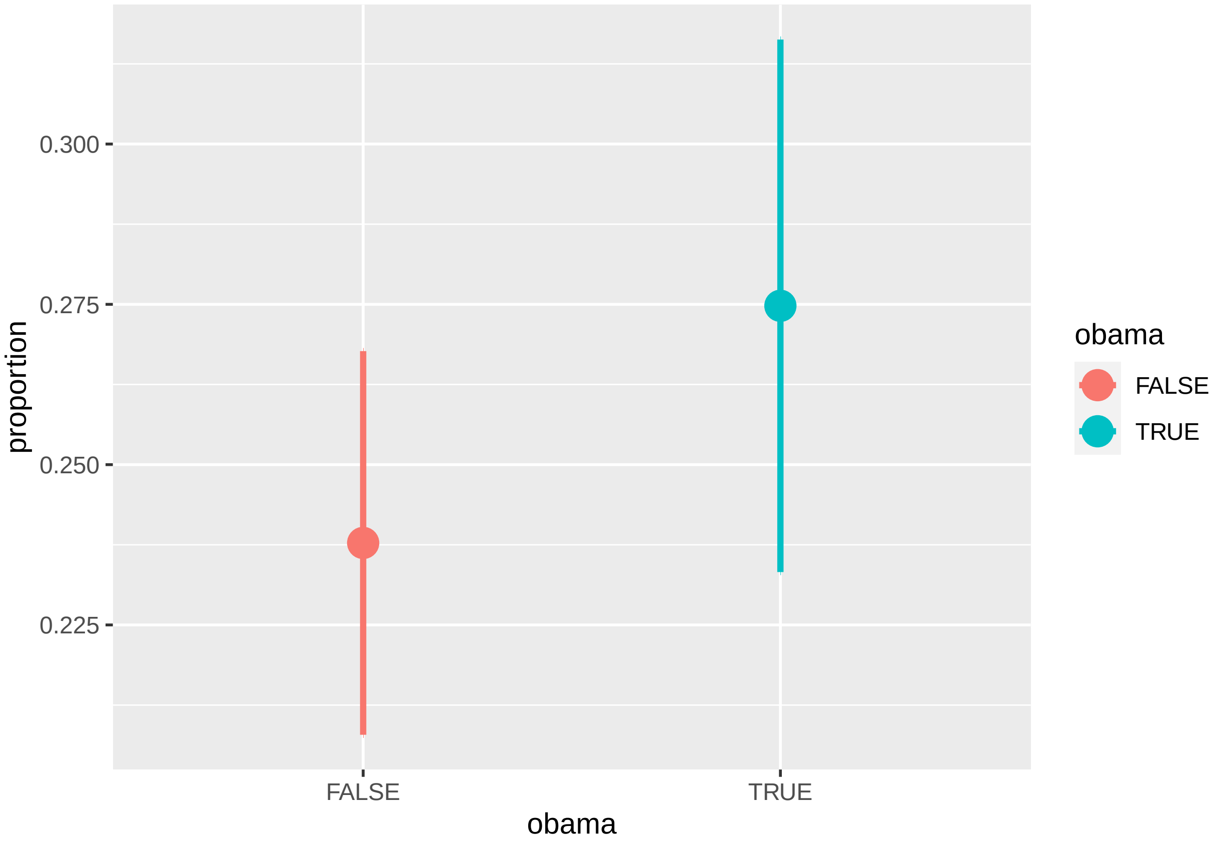
Looking pretty good. Let's go ahead and add some error bars:

```
p + geom_errorbar(aes(
  ymin = proportion - 1.96 * se, # Add error bars
  ymax = proportion + 1.96 * se
))
```



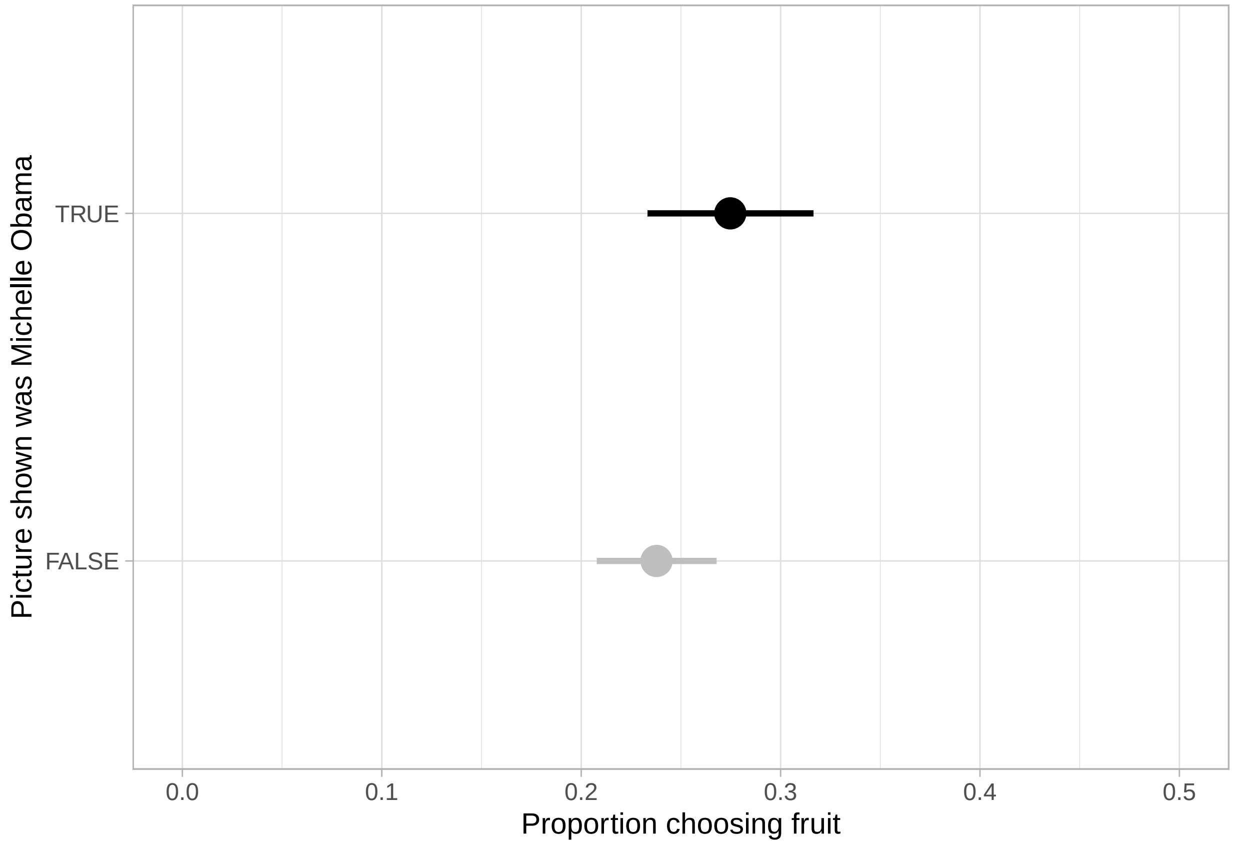
Now let's clean up those error bars a bit…

```
p1 <- p + geom_errorbar(aes(
  ymin = proportion - 1.96 * se,
  ymax = proportion + 1.96 * se,
  width = 0, # Remove the whiskers
  color = obama
), size = 1.1) # Make bars thicker and apply color

p1
```



Great! Now I can style it a bit more by flipping the plot on it's side with `coord_flip()`, adding a theme, converting it to grayscale, and fixing up my axes a bit:

```
p1 + coord_flip() + # Flip the chart
  theme_light() + # Change the theme (theme_minimal is also nice)
  scale_color_manual(values = c("gray", "black"), guide = F) + # Change the colors
  ylim(0, .5) + # Change the y axis to go from 0 to .5
  ylab("Proportion choosing fruit") + # Add labels
  xlab("Picture shown was Michelle Obama")
```



Pretty good!

PC6. Export a table

Here's one way to export our table, using `write.csv()` (and have commented it out, so you can run locally and determine where the exported file goes).

```
## uncomment to generate exported file
## write.csv(obama.tbl, file = 'crosstabs.csv')
```

We can make sure it worked by importing it (again, commented out here):

```
## read.csv('crosstabs.csv')
```

We lost some information, because the `table()` function doesn't save column names. Another way to do this would be to change it into a dataframe first, like this:

```
data.frame(obama.tbl)
```

```
##   took_fruit saw_flotus Freq
## 1      FALSE      FALSE  593
## 2       TRUE      FALSE  185
## 3      FALSE       TRUE  322
## 4       TRUE       TRUE  122
```

and then save that dataframe. Note that you can drop the rownames when importing (or exporting)

```
## write.csv(data.frame(obama.tbl), file = 'crosstabs.csv', row.names = FALSE)

## read.csv('crosstabs.csv')
```

That's formatted a little bit funny, but it's still usable.

You could also use the `xtable` package to do this. The package has many functions to customize table outputs in several formats. A relatively simple way to generate an html table looks like this:

```
library(xtable)
print(xtable(obama.tbl), type = "html")
```

```
## <!-- html table generated in R 4.0.3 by xtable 1.8-4 package -->
## <!-- Mon Oct 26 11:15:52 2020 -->
## <table border=1>
## <tr> <th>  </th> <th> FALSE </th> <th> TRUE </th>  </tr>
##   <tr> <td align="right"> FALSE </td> <td align="right"> 593 </td> <td align="right"> 322 </td> </tr
##   <tr> <td align="right"> TRUE </td> <td align="right"> 185 </td> <td align="right"> 122 </td> </tr>
##    </table>
```

You can export by assigning the html to an object and saving. I've commented it out here so you can choose whether/where to create the file:

```
## uncomment to generate file output

## print(xtable(obama.tbl), type="html", file="example_table.html")
```

You should be able to open that file in a web-browser.

There is a lot of documentation and examples online to help you customize as you see fit. If you're really excited about exporting tables, you might also take a look at the `tables` package, which has some nice export options.

Empirical paper questions

EQ1. LilyPad Arduino users

a) The unit of analysis is the customer. The dependent variable is the type of board purchased and the independent variable is gender. Males, females, and unknown gender customers are being compared. This is a two-way test.

b) For this type of comparison statistical tests help to give (or take away) confidence in any observed differences in counts or proportions across categories. Choosing a statistical test is based on the question that you want to answer and the type of data that you have available to answer it. For example, if this were continuous numeric data (e.g., the amount of money spent on electronics for men and women) then we would want a different to compare those distributions. Given that the test compares counts (or proportions) a χ^2 test for independence is appropriate.

c) The null hypothesis (H_0) is that the board purchased is independent of the gender of the customer. The alternative hypothesis (H_A) is that board purchase choice is dependent on gender.

d) A χ^2 test results indicate that board purchase behavior differs by gender ($p \leq 0.05$). This difference is convincing, but it does directly not tell us what the authors set out to understand, which is the difference between men and women (the test as-implemented might have identified a significant difference in the number of unknown gender customers across board types!). Many of these concerns are addressed in the text and with additional tests, giving increased confidence in the observed differences.

EQ2. Two blogospheres

a) The data are counts for two categorical variables and the procedure used was a χ^2 test. The null hypothesis is that blog governance (by one person or more than one person) is independent of whether the blog was on the left or the right ideologically.

b) One way to approach this focuses on the credibility of the null hypothesis. A null of equality/independence in this case seems like it might be a bit of a stretch since there are potentially so many factors involved in determining site governance. If we accept that the null hypothesis of no difference across the two groups is compelling, it seems like it could be surprising to see these results in a world where ideological orientation and blog governance have no relationship. In this respect, it makes sense to believe that there is some relationship of dependence. A closer reading of the paper suggests a different reason to be skeptical: the way that the measure of blog governance groups the data into categories. The authors could have grouped them differently (e.g., 1-2 people, 3-4 people, and 5+ people). If the decision on how to group was made after seeing the data or if the observed result depends on the choice of grouping, then we have good reason to be skeptical.

c) We can do this in R.

```r
## First we create the dataframe
df <- data.frame(
  Governance = c("Individual", "Multiple", "Individual", "Multiple"),
  Ideology = c("Left", "Left", "Right", "Right"),
  Count = c(13, 51, 27, 38)
)

df
```

```
##   Governance Ideology Count
## 1 Individual     Left    13
## 2   Multiple     Left    51
## 3 Individual    Right    27
## 4   Multiple    Right    38
```

```r
## We can make sure it's the same by testing the Chi-squared
chisq.test(matrix(df$Count, nrow = 2))
```

```
## 
##  Pearson's Chi-squared test with Yates' continuity correction
## 
## data:  matrix(df$Count, nrow = 2)
## X-squared = 5.8356, df = 1, p-value = 0.01571
```

```r
## We can convert that into proportions several ways.
proportions(matrix(df$Count, nrow = 2), margin = 2)
```

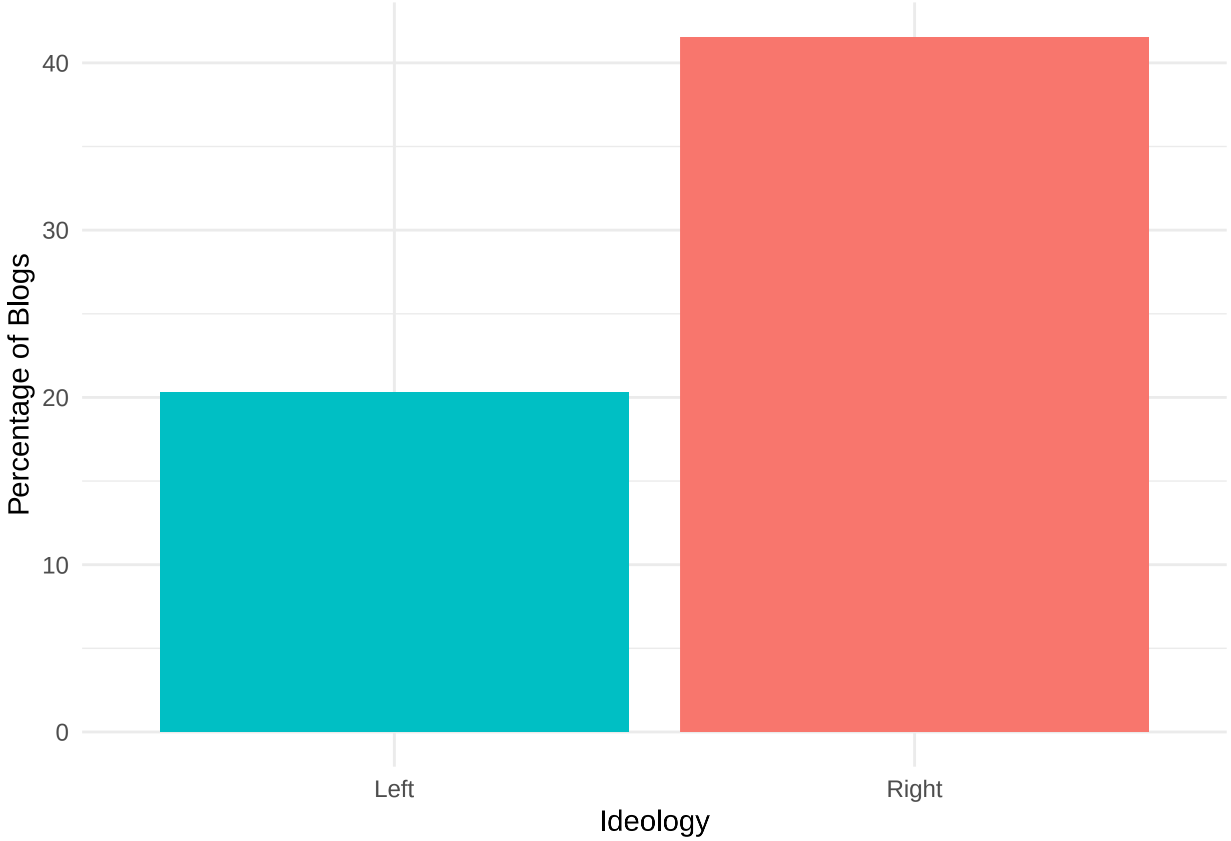
```
##           [,1]      [,2]
## [1,] 0.203125 0.4153846
## [2,] 0.796875 0.5846154
```

```r
## Here's one using tidyverse code that yields more readable results:
percentage_data <- df %>%
  group_by(Ideology) %>%
  summarize(
    individual_ratio = sum(Count[Governance == "Individual"]) / sum(Count),
    ideology_count = sum(Count)
  )

percentage_data
```

```
## # A tibble: 2 x 3
##   Ideology individual_ratio ideology_count
##   <chr>               <dbl>          <dbl>
## 1 Left                0.203             64
## 2 Right                0.415            65
```

And here we go with a figure using the `geom_bar()` layer in `ggplot2`. Note that by calling 'stat='identity", we tell ggplot to vizualize the provided counts (instead of trying to summarize the data further):

```r
shaw_benkler_plot <- percentage_data %>%
  ggplot(aes(x = Ideology, y = individual_ratio * 100)) +
  geom_bar(stat = "identity", aes(fill = c("red", "blue")), show.legend = F) +
  ylab("Percentage of Blogs") +
  theme_minimal()

shaw_benkler_plot
```

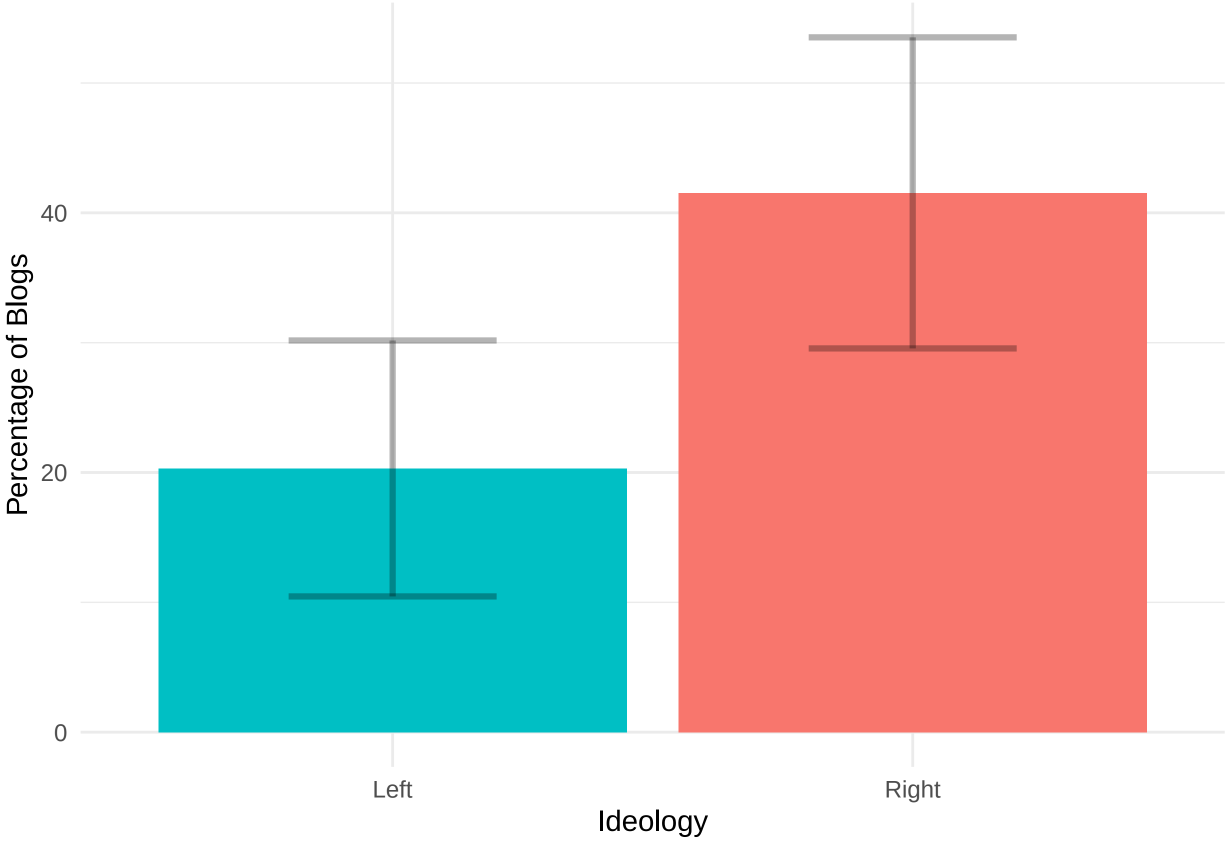


If we want to add error bars, we need to calculate them (Note that ggplot could do this for us if we had raw data - what a helpful reminder to always share your data!).

For our purposes here, you might decide to use confidence intervals or standard errors (both seem like reasonable choices as long as you label them). Either way, ggplot has a `geom_errorbar` layer that is very useful.

Remember that for a binomial distribution (we can consider individual/non-individual as binomial), confidence intervals are $\hat{p} \pm z^* \sqrt{\frac{\hat{p}\ (1-\hat{p})}{n}}$ and we can approximate $z^* \ = 1.96$ for a 95% confidence interval.

```r
ci_95 <- 1.96 * sqrt(percentage_data$individual_ratio * (1 - percentage_data$individual_ratio) / percen

shaw_benkler_plot + geom_errorbar(aes(ymin = (individual_ratio - ci_95) * 100, ymax = (individual_ratio
  alpha = .3,
  size = 1.1,
  width = .4
)
```



The 95% confidence intervals overlap in this case, indicating that the true population proportions may not be as far apart as our point estimates suggest. Note that this is not the same as the hypothesis test (illustrating one of Reinhart's points).

d) On the one hand, we don't need to worry about the base rate fallacy because the sizes of both groups are about the same and the paper does not abuse the evidence too egregiously. The base rate fallacy would likely come into play, however, in the ways that the results are (mis)represented. For example, you might imagine some news coverage looking at these results and claiming something (totally wrong!) like "study finds right wing blogs more than twice as likely to be solo affairs." This is taking a relationship between the sample proportions ($\hat{p}$ in the language of our textbook) and converting that into a statement about the relationship between population proportions (p). That would be a bit of a mess (and absolutely characterizes the news coverage that did follow the study).

   Another way in which the base rate fallacy could play a role in this paper, however, concerns the presence of multiple comparisons. The authors conducted numerous statistical tests (indeed, one of the authors seems to recall that some of the tests were not even reported in the paper <gasp!>) and they make no effort to address the baseline probability of false positives.

   In any case, the point here is that the statistical tests reported in the paper may not mean exactly what the authors said they did in the context of the publication. That may or may not change the validity of the results, but it should inspire us all to do better statistical analysis.