IPS9 in R: Inference for Categorical Data (Chapter 9)

Shukry Zablah (szablah20@amherst.edu) and Nicholas Horton (nhorton@amherst.edu)

July 19, 2018

Introduction and background

These documents are intended to help describe how to undertake analyses introduced as examples in the Ninth Edition of *Introduction to the Practice of Statistics* (2017) by Moore, McCabe, and Craig.

More information about the book can be found here. The data used in these documents can be found under Data Sets in the Student Site. This file as well as the associated R Markdown reproducible analysis source file used to create it can be found at https://nhorton.people.amherst.edu/ips9/.

This work leverages initiatives undertaken by Project MOSAIC (http://www.mosaic-web.org), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignettes (http://cran.r-project.org/web/packages/mosaic). A paper describing the mosaic approach was published in the *R Journal*: https://journal.r-project.org/archive/2017/RJ-2017-024.

Chapter 9: Inference for Categorical Data

This file replicates the analyses from Chapter 9: Inference for Categorical Data.

First, load the packages that will be needed for this document:

```
library(mosaic)
library(readr)
```

Section 9.1: Inference for two-way tables

We will first recreate the dataset based on the table of counts that is provided in Example 9.1 in page 526.

```
Instag <- rbind(
  do(298) * data.frame(Sex = "Men",   User = "No"),
  do(209) * data.frame(Sex = "Women", User = "No"),
  do(234) * data.frame(Sex = "Men",   User = "Yes"),
  do(328) * data.frame(Sex = "Women", User = "Yes")
)
head(Instag)
```

```
##   Sex User .row .index
## 1 Men   No    1      1
## 2 Men   No    1      2
## 3 Men   No    1      3
## 4 Men   No    1      4
## 5 Men   No    1      5
## 6 Men   No    1      6
```

In the code chunk above we are adding the appropriate number of observations (based on the counts) with their respective attributes (whether they are men or women and a user or not) into the dataset. We take a small peek of the dataset with the `head()` function that returns the first few observations from a given dataset.

To recreate the table in Example 9.1 we first get the total count per sex.

```r
Combined_Sex <- Instag %>%
  group_by(Sex) %>%
  summarize(n = n())
Combined_Sex
```

```
## # A tibble: 2 x 2
##   Sex       n
##   <fct> <int>
## 1 Men     532
## 2 Women   537
```

Then we get a dataframe with only the counts of those who have a value of “Yes” for User.

```r
YesUsers <- Instag %>%
  group_by(User, Sex) %>%
  summarize(n = n()) %>%
  filter(User == "Yes")
YesUsers
```

```
## # A tibble: 2 x 3
## # Groups:   User [1]
##   User  Sex       n
##   <fct> <fct> <int>
## 1 Yes   Men     234
## 2 Yes   Women   328
```

And finally, we combine them and create the

$$\hat{p} = X/n$$

column.

```r
Ex8.11Table <- Combined_Sex %>%
  left_join(YesUsers, by = "Sex") %>%
  select(Sex, n = n.x, X = n.y) %>%
  mutate(`p_hat = X/n` = X/n)
Ex8.11Table
```

```
## # A tibble: 2 x 4
##   Sex       n     X `p_hat = X/n`
##   <fct> <int> <int>         <dbl>
## 1 Men     532   234         0.440
## 2 Women   537   328         0.611
```

And we then have the table with the percentages of yes over number of users!

Now take look at Example 9.2 in page 526. To recreate that table of counts we simply have to call the `tally()` function and it will make the 2-way table for us.

We call it like this:

```r
tally(~ User + Sex, data = Instag, margins = TRUE)
```

```
##        Sex
## User     Men Women Total
##   No     298   209   507
##   Yes    234   328   562
##   Total  532   537  1069
```

The `margins = TRUE` optio makes sure that `tally()` ouputs the convenient Total columns just like in page 527!

Turn your attention to Example 9.3 now. After creating the dataset from the counts, we can use a similar call to recreate the table and verify that our dataset is in fact accurate.

```
Vaccine <- rbind(
  do(729) * data.frame(Required = "Yes", Party = "Democratic"),
  do(479) * data.frame(Required = "Yes", Party = "Republican"),
  do(230) * data.frame(Required = "No",  Party = "Democratic"),
  do(258) * data.frame(Required = "No", Party = "Republican")
)

tally(~ Required + Party, data = Vaccine, margins = TRUE)
```

```
##         Party
## Required Democratic Republican Total
##    Yes          729        479  1208
##    No           230        258   488
##    Total        959        737  1696
```

Now we continue to explore our 2 way tables. In Example 9.5 we can see the marginal distribution of our Vaccine tables across political party preference. We recreate it with a call to `tally()`.

```
tally(Required ~ Party, data = Vaccine, margins = TRUE, format = "percent")
```

```
##         Party
## Required Democratic Republican
##    Yes      76.01668   64.99322
##    No       23.98332   35.00678
##    Total  100.00000  100.00000
```

In Example 9.7 we are interested in getting the expected counts of our Vaccine data. In R you can take advantage of the `chisq.test()` function and get the relevant output like this:

```
chiSq <- chisq.test(tally(Required ~ Party, data = Vaccine), correct = FALSE)
with(chiSq, expected)
```

```
##         Party
## Required Democratic Republican
##      Yes   683.0613   524.9387
##      No    275.9387   212.0613
```

We specify the `correct = FALSE` option to match the book's table. This option specifies that there should be no continuity correction applied to our test. You can see how the output changes by removing that option.

Similarly we could get the observed counts we calculated with `tally()` before. We just need to use the same test and get the relevant part of the returned object.

```
with(chiSq, observed)
```

```
##         Party
## Required Democratic Republican
##      Yes        729        479
##      No         230        258
```

To see the output of the Chi-Square test discussed in Example 9.8. Note that the book has a mistake. While it showed the correct output, it specified the wrong

$$\chi^2$$

squared value.

```
chiSq
```

```
##
##  Pearson's Chi-squared test
##
## data:  tally(Required ~ Party, data = Vaccine)
## X-squared = 24.709, df = 1, p-value = 6.666e-07
```

Section 9.2: Goodness of fit