IS5 in R: Comparing Counts (Chapter 19)

Nicholas Horton (nhorton@amherst.edu)

December 17, 2020

Introduction and background

Chapter 19: Comparing Counts

```
library(mosaic)
library(readr)
library(janitor)
Zodiac <- read_csv("http://nhorton.people.amherst.edu/is5/data/Zodiac.csv")
```

```
##
## -- Column specification --------------------------------------------------------------------------
## cols(
##   Month = col_character(),
##   Births = col_double(),
##   Expected = col_double(),
##   Residual = col_double()
## )
```

By default, `read_csv()` prints the variable names. These messages can be suppressed using the `message=FALSE` code chunk option to save space and improve readability.

```
Zodiac %>%
  select(Month, Births)
```

```
## # A tibble: 12 x 2
##    Month       Births
##    <chr>        <dbl>
##  1 Pisces          29
##  2 Aquarius        24
##  3 Aries           23
##  4 Cancer          23
##  5 Capricorn       22
##  6 Scorpio         21
##  7 Taurus          20
##  8 Leo             20
##  9 Saggitarius     19
## 10 Virgo           19
## 11 Libra           18
## 12 Gemini          18
```

Section 19.1: Goodness-of-Fit Tests

```
# page 611
```

```
BaseballBirths <- read_csv("http://nhorton.people.amherst.edu/is5/data/Ballplayer_births.csv") %>%
  janitor::clean_names() # doesn't contain national birth %
```

Example 19.1: Finding Expected Counts Here we use the `clean_names()` function from the `janitor` package to sanitize the names of the columns (which would otherwise contain special characters or whitespace).

```
natbirth <- c(.08, .07, .08, .08, .08, .08, .09, .09, .09, .09, .08, .09)
BaseballBirths <- cbind(BaseballBirths, natbirth) # adding a column for national birth %
totaln <- sum(~ballplayer_count, data = BaseballBirths)
totaln
```

```
## [1] 1478
```

```
BaseballBirths <- BaseballBirths %>%
  mutate(
    expected = totaln * natbirth,
    observed = ballplayer_count,
    contrib = (observed - expected)^2 / expected
  )
sum(~contrib, data = BaseballBirths)
```

```
## [1] 26.48442
```

Assumptions and Conditions

Calculations

```
# Examples of chisq p-values
qchisq(df = 2, p = .1, lower.tail = FALSE)
```

Chi-Square P-values

```
## [1] 4.60517
```

```
qchisq(df = 10, p = .05, lower.tail = FALSE)
```

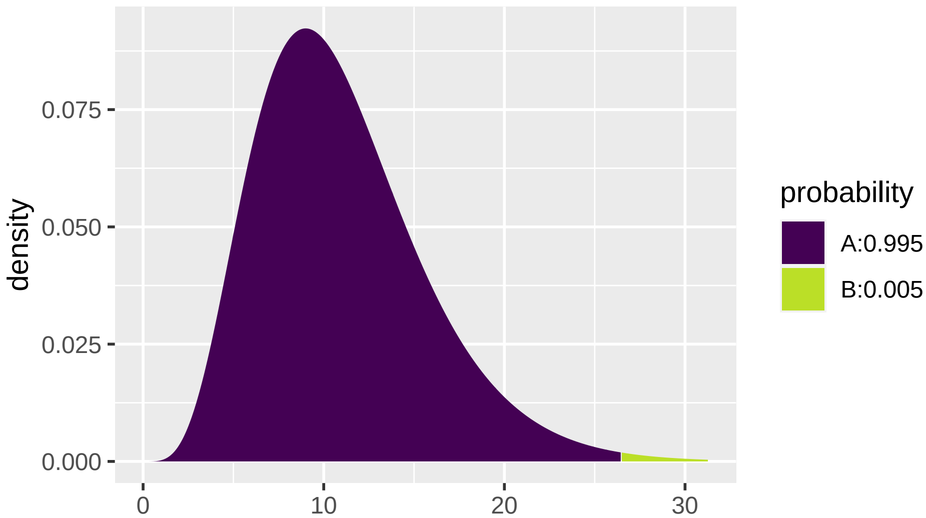
```
## [1] 18.30704
```

```
# page 614
df <- nrow(BaseballBirths) - 1
df
```

Example 19.3: Doing a Goodness-of-Fit Test

```
## [1] 11
```

```
chisq <- sum(~contrib, data = BaseballBirths)
xpchisq(q = chisq, df = df, lower.tail = FALSE)
```



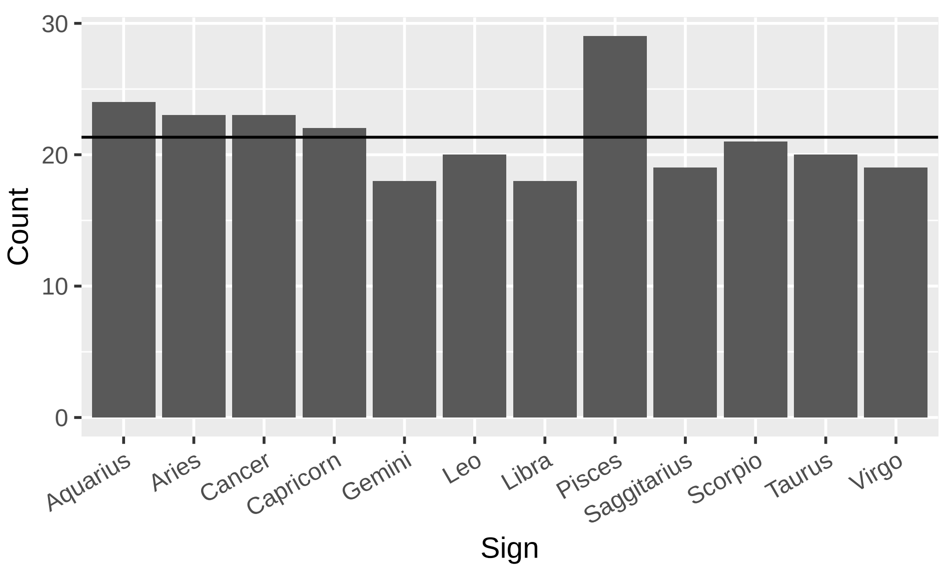

```
## [1] 0.005494028
```

```
expected <- mean(~Births, data = Zodiac)
expected
```

Step-By-Step Example: A Chi-Square Test for Goodness-of-Fit

```
## [1] 21.33333
```

```
gf_col(Births ~ Month, data = Zodiac) %>%
  gf_hline(yintercept = expected) %>%
  gf_labs(x = "Sign", y = "Count") +
  theme(axis.text.x = element_text(angle = 30, hjust = 1)) # to adjust the angle of the x axis labels
```



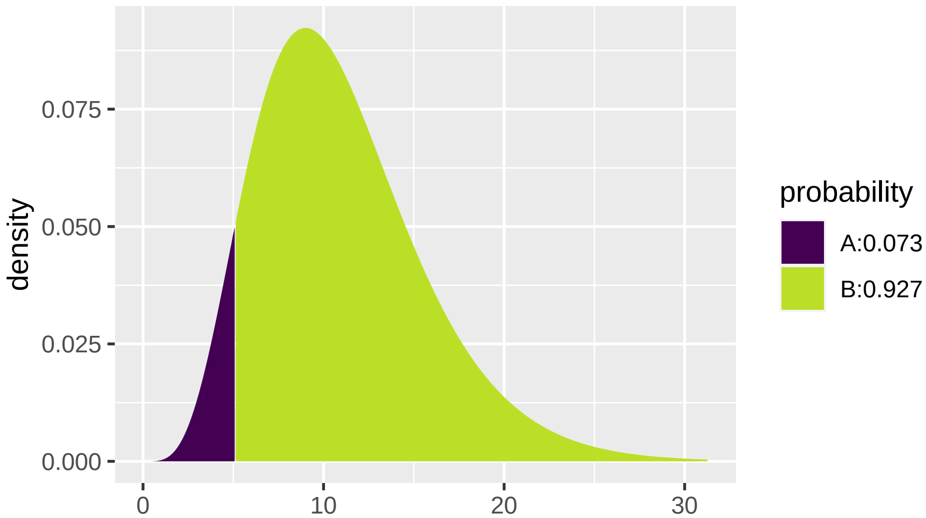

```
# Mechanics
df <- nrow(Zodiac) - 1
df
```

```
## [1] 11
```

```r
Zodiac <- Zodiac %>%
  mutate(chisq = ((Births - Expected)^2) / Expected)
chisq <- sum(~chisq, data = Zodiac)
chisq
```

```
## [1] 5.09383
```

```r
xpchisq(q = chisq, df = df, lower.tail = FALSE)
```




```
## [1] 0.9265374
```

```r
Zodiac %>%
  mutate(residsq = Residual^2) %>%
  mutate(component = residsq / Expected)
```

The Chi-Square Calculation

```
## # A tibble: 12 x 7
##    Month       Births Expected Residual   chisq residsq component
##    <chr>        <dbl>    <dbl>    <dbl>   <dbl>   <dbl>     <dbl>
##  1 Pisces          29     21.3    7.67  2.76     58.8     2.76
##  2 Aquarius        24     21.3    2.67  0.333     7.11    0.333
##  3 Aries           23     21.3    1.67  0.130     2.78    0.130
##  4 Cancer          23     21.3    1.67  0.130     2.78    0.130
##  5 Capricorn       22     21.3    0.667 0.0209    0.445   0.0209
##  6 Scorpio         21     21.3   -0.333 0.00520   0.111   0.00520
##  7 Taurus          20     21.3   -1.33  0.0833    1.78    0.0833
##  8 Leo             20     21.3   -1.33  0.0833    1.78    0.0833
##  9 Saggitarius     19     21.3   -2.33  0.255     5.44    0.255
## 10 Virgo           19     21.3   -2.33  0.255     5.44    0.255
## 11 Libra           18     21.3   -3.33  0.521    11.1     0.521
## 12 Gemini          18     21.3   -3.33  0.521    11.1     0.521
```

The Trouble with Goodness-of-Fit Tests: What's the Alternative?

Section 19.2: Chi-Square Test of Homogeneity

```
# Create the data set
Postgrad <- rbind(
  do(209) * data.frame(activity = "Employed", school = "Agriculture"),
  do(198) * data.frame(activity = "Employed", school = "Arts & Sciences"),
  do(177) * data.frame(activity = "Employed", school = "Engineering"),
  do(101) * data.frame(activity = "Employed", school = "ILR"),
  do(104) * data.frame(activity = "Grad School", school = "Agriculture"),
  do(171) * data.frame(activity = "Grad School", school = "Arts & Sciences"),
  do(158) * data.frame(activity = "Grad School", school = "Engineering"),
  do(33) * data.frame(activity = "Grad School", school = "ILR"),
  do(135) * data.frame(activity = "Other", school = "Agriculture"),
  do(115) * data.frame(activity = "Other", school = "Arts & Sciences"),
  do(39) * data.frame(activity = "Other", school = "Engineering"),
  do(16) * data.frame(activity = "Other", school = "ILR")
)
```

```
# Table 19.1, page 618
tally(activity ~ school, data = Postgrad, margins = TRUE)
```

```
##              school
## activity      Agriculture Arts & Sciences Engineering ILR
##   Employed            209             198         177 101
##   Grad School         104             171         158  33
##   Other               135             115          39  16
##   Total               448             484         374 150
```

```
# Table 19.2
tally(activity ~ school, format = "percent", data = Postgrad, margins = TRUE)
```

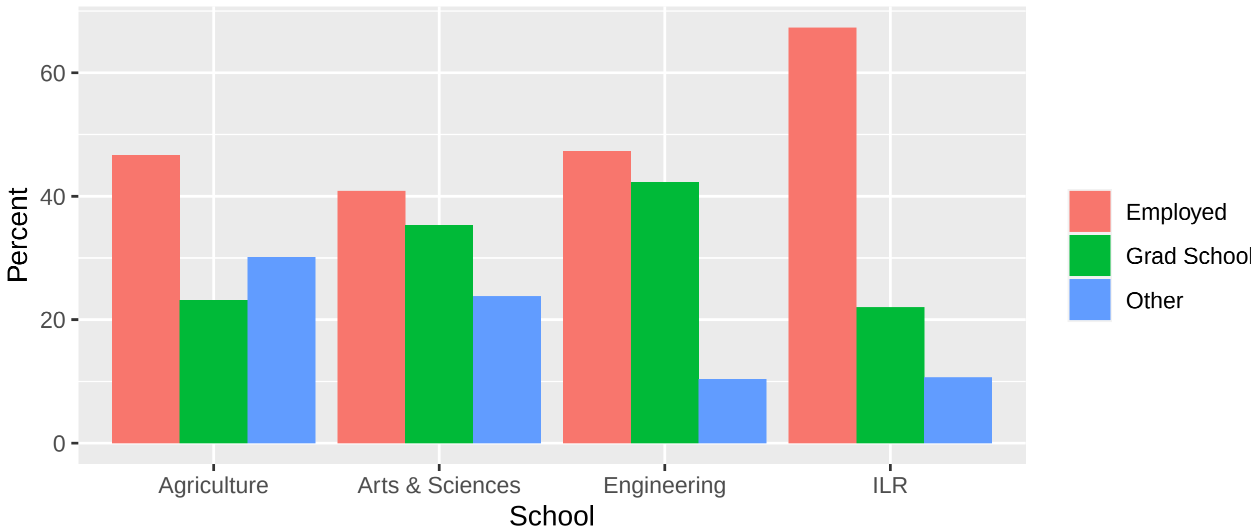
```
##              school
## activity      Agriculture Arts & Sciences Engineering       ILR
##   Employed       46.65179        40.90909    47.32620  67.33333
##   Grad School    23.21429        35.33058    42.24599  22.00000
##   Other          30.13393        23.76033    10.42781  10.66667
##   Total         100.00000       100.00000   100.00000 100.00000
```

```
# Table 19.3
with(chisq.test(tally(activity ~ school, data = Postgrad, margins = TRUE)), expected)
```

```
##              school
## activity      Agriculture Arts & Sciences Engineering       ILR
##   Employed      210.76923        227.7060   175.95467  70.57005
##   Grad School   143.38462        154.9066   119.70055  48.00824
##   Other          93.84615        101.3874    78.34478  31.42170
##   Total         448.00000        484.0000   374.00000 150.00000
```

Step-By-Step Example: A Chi-Square Test for Homogeneity We can undertake a chi-square test for homogeneity. First let's display the data.

```
tally(activity ~ school, format = "percent", data = Postgrad) %>%
  data.frame() %>%
  gf_col(Freq ~ school, fill = ~activity, position = "dodge") %>%
  gf_labs(x = "School", y = "Percent", fill = "")
```




```r
# Mechanics
tally(activity ~ school, data = Postgrad, margins = TRUE)
```

```
##              school
## activity      Agriculture Arts & Sciences Engineering ILR
##   Employed            209             198         177 101
##   Grad School         104             171         158  33
##   Other               135             115          39  16
##   Total               448             484         374 150
```

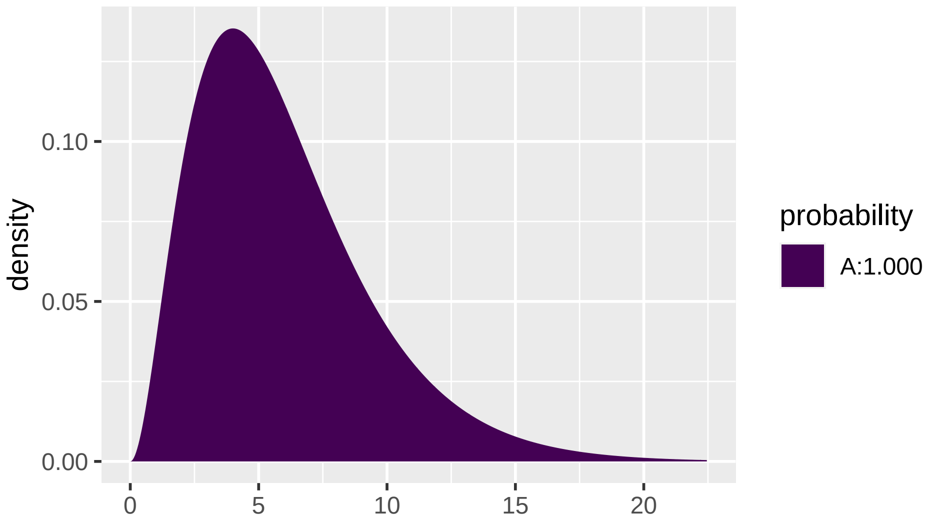
```r
with(chisq.test(tally(activity ~ school, data = Postgrad, margins = TRUE)), expected)
```

```
##              school
## activity      Agriculture Arts & Sciences Engineering       ILR
##   Employed      210.76923        227.7060   175.95467  70.57005
##   Grad School   143.38462        154.9066   119.70055  48.00824
##   Other          93.84615        101.3874    78.34478  31.42170
##   Total         448.00000        484.0000   374.00000 150.00000
```

```r
with(chisq.test(tally(activity ~ school, data = Postgrad)), statistic)
```

```
## X-squared
##  93.65667
```

```r
xpchisq(q = 93.7, df = 6, lower.tail = FALSE)
```




```
## [1] 5.154981e-18
```

Section 19.3: Examining the Residuals

```
# Table 19.4, page 622
with(chisq.test(tally(activity ~ school, data = Postgrad, margins = TRUE)), residuals)
```

```
##              school
## activity      Agriculture Arts & Sciences Engineering         ILR
##   Employed    -0.12186553     -1.96860027  0.07880484  3.62235442
##   Grad School -3.28908677      1.29304319  3.50061599 -2.16606715
##   Other        4.24817296      1.35191804 -4.44510568 -2.75117035
##   Total        0.00000000      0.00000000  0.00000000  0.00000000
```

```
BaseballBirths %>%
  mutate(residuals = (ballplayer_count - expected) / (expected^.5)) %>%
  select(month, residuals)
```

Example 19.4: Looking at χ^2, Residuals

```
##    month   residuals
## 1      1  1.72524439
## 2      2  1.72442119
## 3      3 -0.20599933
## 4      4  0.25382060
## 5      5  0.71364054
## 6      6 -0.38992730
## 7      7 -2.68957291
## 8      8  2.77280921
## 9      9  0.08497039
## 10    10 -1.56241469
## 11    11 -1.21760318
## 12    12 -0.95548335
```

Section 19.4: Chi-Square Test of Independence

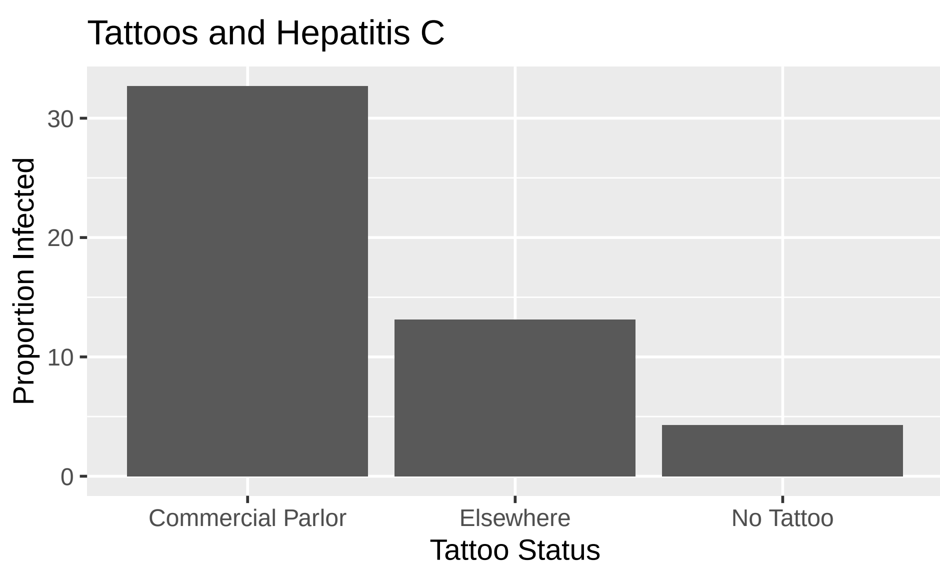
```
Tattoos <- read_csv("http://nhorton.people.amherst.edu/is5/data/Tattoos.csv", skip = 1) %>%
  janitor::clean_names() # skip = 1 because first row is "Col1", "Col2"
# Table 19.5, page 623
tally(location ~ has_hepatitis_c, data = Tattoos, margins = TRUE)
```

```
##                    has_hepatitis_c
## location             No Yes
##   Commercial Parlor  35  17
##   Elsewhere          53   8
##   No Tattoo         491  22
##   Total             579  47
```

Assumptions and Conditions

Step-By-Step Example: A Chi-Square Test for Independence We use the `mosaic::tally()` function to prepare the data for the graphical display.

```
tally(has_hepatitis_c ~ location, format = "percent", data = Tattoos) %>%
  data.frame() %>%
  filter(has_hepatitis_c == "Yes") %>%
  gf_col(Freq ~ location) %>%
  gf_labs(x = "Tattoo Status", y = "Proportion Infected", title = "Tattoos and Hepatitis C")
```




```
# Observed
tally(location ~ has_hepatitis_c, data = Tattoos, margins = TRUE)
```

```
##                    has_hepatitis_c
## location             No Yes
##   Commercial Parlor  35  17
##   Elsewhere          53   8
##   No Tattoo         491  22
##   Total             579  47
```

```
# Expected
with(chisq.test(tally(location ~ has_hepatitis_c, data = Tattoos, margins = TRUE)), expected)
```

```
## Warning in stats::chisq.test(x, y, ...): Chi-squared approximation may be
## incorrect
```

```
##                    has_hepatitis_c
## location                   No       Yes
##   Commercial Parlor  48.09585  3.904153
##   Elsewhere          56.42013  4.579872
##   No Tattoo         474.48403 38.515974
##   Total             579.00000 47.000000
```

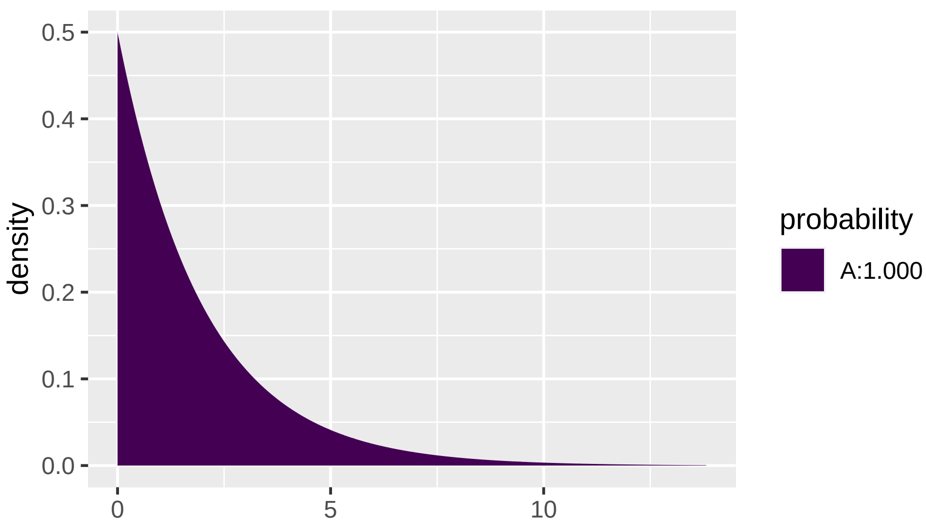
We note the warning that several of the expected cell counts are less than 5, which raises concerns about the accuracy of the test.

```
# Mechanics
with(chisq.test(tally(location ~ has_hepatitis_c, data = Tattoos)), statistic)
```

```
## Warning in stats::chisq.test(x, y, ...): Chi-squared approximation may be
## incorrect
```

```
## X-squared
##  57.91217
```

```
xpchisq(q = 57.9, df = 2, lower.tail = FALSE)
```




```
## [1] 2.674082e-13
```

```
# Table 19.6, page 627
with(chisq.test(tally(location ~ has_hepatitis_c, data = Tattoos)), residuals)
```

Examine the Residuals

```
## Warning in stats::chisq.test(x, y, ...): Chi-squared approximation may be
## incorrect
```

```
##                    has_hepatitis_c
## location                    No        Yes
##   Commercial Parlor -1.8883383  6.6278115
##   Elsewhere         -0.4553290  1.5981431
##   No Tattoo          0.7582168 -2.6612383
```

```r
# Table 19.7, page 628
Tattoos <- Tattoos %>%
  mutate(tattoo = ifelse(location == "No Tattoo", "None", "Tattoo"))
tally(tattoo ~ has_hepatitis_c, margins = TRUE, data = Tattoos)
```

```
##         has_hepatitis_c
## tattoo    No Yes
##   None   491  22
##   Tattoo  88  25
##   Total  579  47
```

Chi-Square and Causation