Categorical Data Analysis

Avalon C.S. Owens, Eric R. Scott

11/02/2018

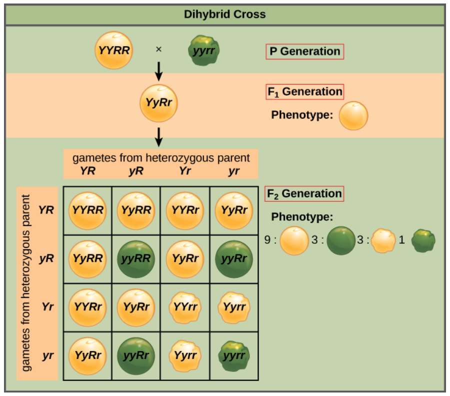
Overview

```
library(tidyverse)
# OR
library(dplyr)
library(ggplot2)
library(tidyr)
# For G-Test:
library(DescTools)
# New package to install:
library(ggmosaic)
```

- Data on Canvas!
- Goodness-of-fit tests (χ^2 and G-test AKA log-likelihood ratio)
- Contingency tables
- Contingency tests (Fisher's exact, χ^2, G-test)
- Mosaic plots in `ggplot2` with `ggmosaic`

Goodness-of-fit

Goodness-of-fit



	YR	yR	Yr	yr
YR	YYRR	YyRR	YYRr	YyRr
yR	YyRR	yyRR	YyRr	yyRr
Yr	YYRr	YyRr	YYrr	Yyrr
yr	YyRr	yyRr	Yyrr	yyrr

Chi-square with expected probabilities

Phenotypes:	Yellow & Round	Green & Round	Yellow & Wrinkled	Green & Wrinkled
Offspring:	93	31	28	8

Do offspring ratios support a diyhybrid cross model?

Observed numbers:

```
Obs <- c(yellowround = 93, greenround = 31, yelowwrinkled = 28,  greenwrinkled = 8)
```

Expected probabilities:

```
Exp.p <- c(9/16, 3/16, 3/16, 1/16)
```

Chi-square with expected probabilities

Do the test with `chisq.test()`

- Supply observed numbers and expected probabilities

```
chisq.test(Obs, p = Exp.p)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  Obs
## X-squared = 0.66667, df = 3, p-value = 0.881
```

- **Interpretation?**

Chi-squared with expected values

Use `rescale.p = TRUE` to use expected *values* instead of expected probabilities

```
Exp <- c(yellowround = 90, yelowwrinkled = 30, greenround = 30, greenwrinkled = 10)
sum(Obs) == sum(Exp) #the expected numbers of each type.
```

```
## [1] TRUE
```

```
chisq.test(Obs, p = Exp, rescale.p = TRUE)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  Obs
## X-squared = 0.66667, df = 3, p-value = 0.881
```

Chi-squared with expected ratios

You could also supply the 9:3:3:1 expected *ratio*

```
props <- c(9, 3, 3, 1)
chisq.test(Obs, p = props, rescale.p = TRUE)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  Obs
## X-squared = 0.66667, df = 3, p-value = 0.881
```

G test

- The package we installed for Dunnnett's Test also has a G-test function!
- Unlike `chisq.test()`, you *must* supply expected **probabilities**.

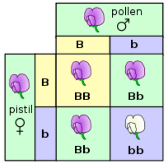
```
#library(DescTools)
GTest(Obs, p = Exp.p) #'p' must be probabilities
```

```
##
##  Log likelihood ratio (G-test) goodness of fit test
##
## data:  Obs
## G = 0.69798, X-squared df = 3, p-value = 0.8737
```

- **Interpretation?**

Exact binomial test

- If there are only two categories, you can use a binomial test.



- You cross two heterozygous corn plants and get 243 dwarf offspring and 682 giant offspring.
- Is plant size a Mendelian trait?

1. Define "success" (totally arbitrary)
2. Probability of "success" = 3/4 if you chose "giant", 1/4 if you chose "dwarf"

Exact binomial test in R

```
binom.test(c(<<#successes>>, <<#failures>>), p = <<probability of success>>)
```

If success is "giant":

```
binom.test(c(682, 243), p = 3/4)
```

```
##
##  Exact binomial test
##
## data:  c(682, 243)
## number of successes = 682, number of trials = 925, p-value =
## 0.3825
## alternative hypothesis: true probability of success is not equal to 0.75
## 95 percent confidence interval:
##  0.7076683 0.7654066
## sample estimates:
## probability of success
##              0.7372973
```

Interpretation

```
##
##  Exact binomial test
##
## data:  c(682, 243)
## number of successes = 682, number of trials = 925, p-value =
## 0.3825
## alternative hypothesis: true probability of success is not equal to 0.75
## 95 percent confidence interval:
##  0.7076683 0.7654066
## sample estimates:
## probability of success
##              0.7372973
```

- Accept or reject null?
- Is it Mendelian?

Contingency

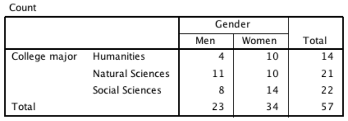
Contingency analysis

Context 1: Assign samples to levels of categorical variable, measure a categorical variable.

Context 2: Random sample individuals and measure *two* categorical variables.

Both have two categorical variables, but data entry and data visualization might differ.

Contingency tables



Count

		Gender		
		Men	Women	Total
College major	Humanities	4	10	14
	Natural Sciences	11	10	21
	Social Sciences	8	14	22
Total		23	34	57

- Inherently untidy!
- It's basically impossible to enter contingency tables *directly*.
- We'll cover two possible formats of data that we can convert to contingency tables in R:

1. Already tabulated data, like you might be likely to get from context 1 experiments
2. Two columns of categorical data, like you might be likely to get from context 2 experiments

Starting with frequncy data (context 1)

Angina treatment data from lecture:

- Every combination of treatment and symptoms and # of people in each group.
- You might record data in this format since you *assigned* the groups and simply counted individuals with or without symptoms.

```
angina <- read.csv("Angina.csv")
angina
```

```
#Try using `View(angina)`
```

Converting frequency data into a contingency table

- `"table"` is a special class in R.
- R knows how to do things like χ^2 tests automatically on `table`s.
- We need to make our `data.frame` into a `table`.
- In this case, we do that with `xtabs()`, which takes a formula.

```
class(angina)
```

```
## [1] "data.frame"
```

Making a `table` with `xtabs()`

- Uses formula interface.
- `Freq` as explained by `Treatment` and `Symptoms`

```
angina.table <- xtabs(Freq ~ Treatment + Symptoms, data = angina)
angina.table
```

```
##          Symptoms
## Treatment Angina No_Angina
##   Placebo    128        19
##   Timelol    116        44
```

```
class(angina.table)
```

```
## [1] "xtabs" "table"
```

Starting with tidy data (context 2)

- Example 9.4 from the text: Are fish infected by a trematode worm eaten or not?
- Two columns of factors: infection status and predation (eaten or not)
- What makes this tidier than the previous example?

```
worm <- read.csv("WormGetsBird.csv")
worm
```

Tabulating data

We *could* get this into the same format as the previous example (a frequency table) using `count()` and then use `xtabs()` on it…

```
worm %>% count(infection, bird_predation)
```

…But there is another way

Tabulating data

- `table()` is *another* function for making contingency tables
- Unlike `xtabs()` it takes two vectors of categorical data as input.

```
worm.table <- table(worm$infection, worm$bird_predation)
worm.table
```

```
##
##              eaten not eaten
##   highly        37         9
##   lightly       10        35
##   uninfected     1        49
```

```
class(worm.table)
```

```
## [1] "table"
```

Two ways to make tables!

- `xtabs()`: Use when you have a column of frequencies and two columns of factors. Uses the formula interface.

```
mytable <- xtabs(freq ~ factor1 + factor2, data = mydata)
```

- `table()`: Use when you have two columns of categorical data and each row is an observation. Needs vectors so you have to use the `$` operator.

```
mytable <- table(myotherdata$factorA, myotherdata$factorB)
```

Adding margins to a contingency table

```
angina.table %>% addmargins()
```

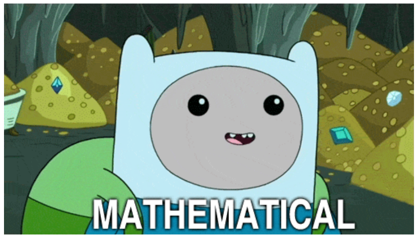
```
##          Symptoms
## Treatment Angina No_Angina Sum
##   Placebo    128        19 147
##   Timelol    116        44 160
##   Sum        244        63 307
```

```
worm.table %>% addmargins()
```

```
##
##              eaten not eaten Sum
##   highly        37         9  46
##   lightly       10        35  45
##   uninfected     1        49  50
##   Sum           48        93 141
```

Contingency tests

Once you have a contingency table of the class `table`, it's easy to do statistical tests



Fisher's Exact test for 2x2 tables

```
fisher.test(angina.table)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  angina.table
## p-value = 0.001785
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  1.365721 4.902015
## sample estimates:
## odds ratio
##   2.547687
```

- Accept or reject null?
- Did the drug work?

χ^2 test

```
chisq.test(worm.table)
```

```
##
##  Pearson's Chi-squared test
##
## data:  worm.table
## X-squared = 69.756, df = 2, p-value = 0.0000000000000007124
```

- Accept or reject null?
- Is fish predation contingent on infection status?

χ^2 test

```
chisq.test(angina.table)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  angina.table
## X-squared = 9.1046, df = 1, p-value = 0.00255
```

G-test

- `GTest()` function from the `DescTools` package

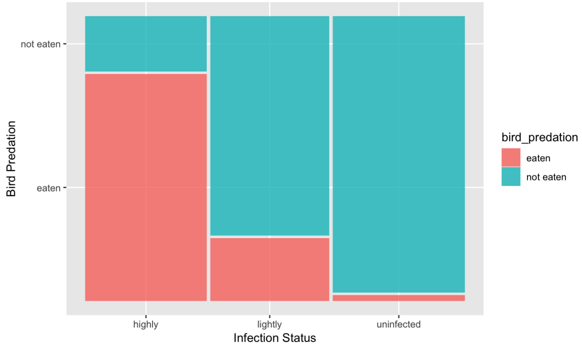
```
GTest(worm.table)
```

```
##
##  Log likelihood ratio (G-test) test of independence without
##  correction
##
## data:  worm.table
## G = 77.897, X-squared df = 2, p-value < 0.00000000000000022
```

Plotting Contingency Data

Mosaic plots

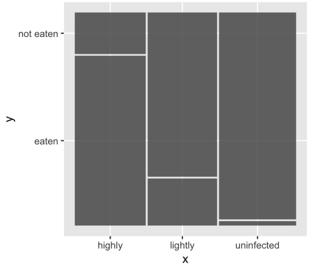
- `ggmosaic` adds `geom_mosaic()` for plotting contingency data
- Works on tidy data, **not** `table`s



Using geom_mosaic()

- `geom_mosaic()` is a little weird, because mosaic plots are a little weird
- `aes()` *MUST* go inside of `geom_mosaic()`, *NOT* inside of `ggplot()`.
- Rather than supplying an x and a y aesthetic, you supply only x as a `product()`. I know, weird.

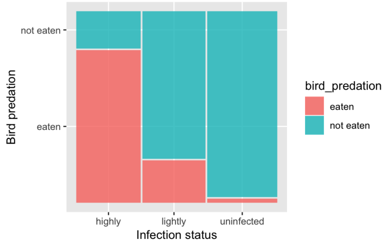
```
library(ggmosaic)
ggplot(worm) +
  geom_mosaic(aes(x = product(bird_predation, infection)))
```



Prettying up `geom_mosaic()`

- Add color with the `fill` aesthetic
- Add axis labels

```
ggplot(worm) +
  geom_mosaic(aes(x = product(bird_predation, infection), fill = bird_predation)) +
  labs(x = "Infection status", y = "Bird predation")
```



Mosaic plot for angina data

- `angina` data isn't tidy!
- Convert frequency table to tidy data with `uncount()` from `tidyr`

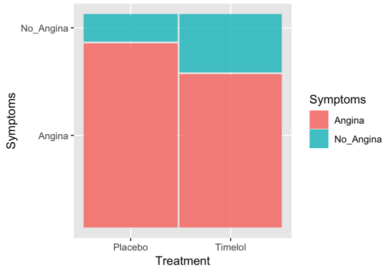
```
angina
```

```
angina.tidy <- angina %>% tidyr::uncount(weights = Freq)
angina.tidy
```

Mosaic plot for angina data

Now we can plot the tidy data with `geom_mosaic()`

```
ggplot(angina.tidy) +
  geom_mosaic(aes(x = product(Symptoms, Treatment), fill = Symptoms)) +
  labs(x = "Treatment", y = "Symptoms")
```



Try on your own!

`mtcars` is a dataset in `ggplot2` from *Motor Trends* magazine. `gear` is the number of gears a car has and `am` is whether a car has manual (1) or automatic (0) transmission.

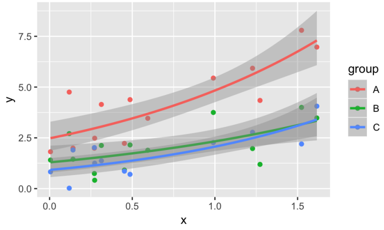
- Is # of gears contingent on transmission type?
- Make a mosaic plot
- Do a statistical test

```
mtcars
```

If you want to learn more…

More statistics!

- Ecological Models and Data (BIO0133)
  - If you have more than two variables
  - Instead of transforming data, use a test that assumes a different distribution besides normal
- Mixed Models Practical Guide
  - Fixed and random effects in the same regression



More tidyverse!

- R for Data Science: r4ds.had.co.nz
- Slack channel for R for Data Science: bit.ly/R4DSslack
- #TidyTuesday on Twitter

More R Markdown!

- Make customized web pages, PDFs, presentations, etc. in RStudio with R Markdown
- Make a website in RStudio with blogdown

Interactive plots!

- Make ggplots interactive with `ggplotly`

```
library(plotly)
p <- ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = Species)) +
  geom_point()
ggplotly(p)
```

Really fancy stuff with Shiny

- Shiny apps

Thank you, keep in touch!

Twitter:

- @LeafyEricScott
- @avalonceleste