Week Five

Last Week

- Contingency Tables
  - Contingency Table Overview: multiple categorical variables
  - joint, marginal, and conditional probabilities
  - Comparing Proportions: Relative Risk & Odds Ratios
  - Chi-Squared Tests for Independence

This Week: Contingency Tables, Part II

Today:

- Activity
- Thursday: Lab

Next Week: Generalized Linear Models

Ordinal Data

Up to this point, we have largely considered nominal categorical data. However, we will also see ordinal data.

Table 1: Ordinal contingency table

	low	medium	high
group 1	$n_{1,l}$	$n_{1,m}$	$n_{1,h}$
group 2	$n_{2,l}$	$n_{2,m}$	$n_{2,h}$

All procedures still apply in this setting. However, we might be interested in a different question rather than just independence we might wish to think more about the trends across the ordinal categories.

Consider simulating data with this structure.

1. Simulate ordinal responses with three levels across two different groups.

2. Visualize the data

3. Create a contingency table

4. Run a χ^2 test and interpret the results. Clearly articulate what you are testing here (avoid statistical lingo).

5. Create another table that includes the observed values and expected values (from the χ^2 test). How do these values contribute to the test statistic in the previous part?

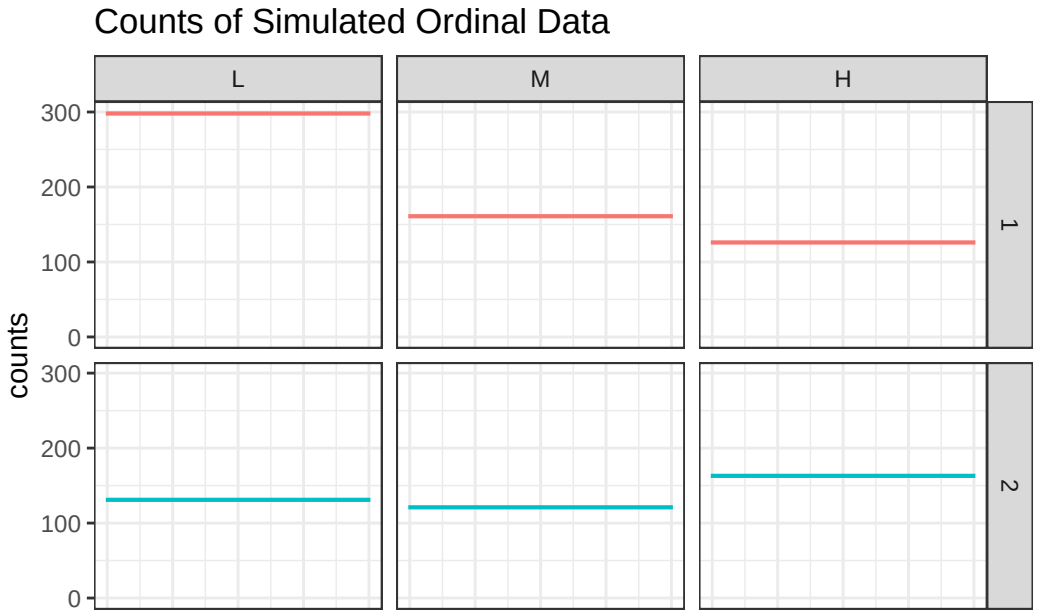
1. Simulate ordinal responses with three levels across two different groups.

The observed data will come from a multinomial distribution. There are many ways we could construct the multinomial probabilites, but we will keep it simple for now - more later when we see ordinal regression methods.

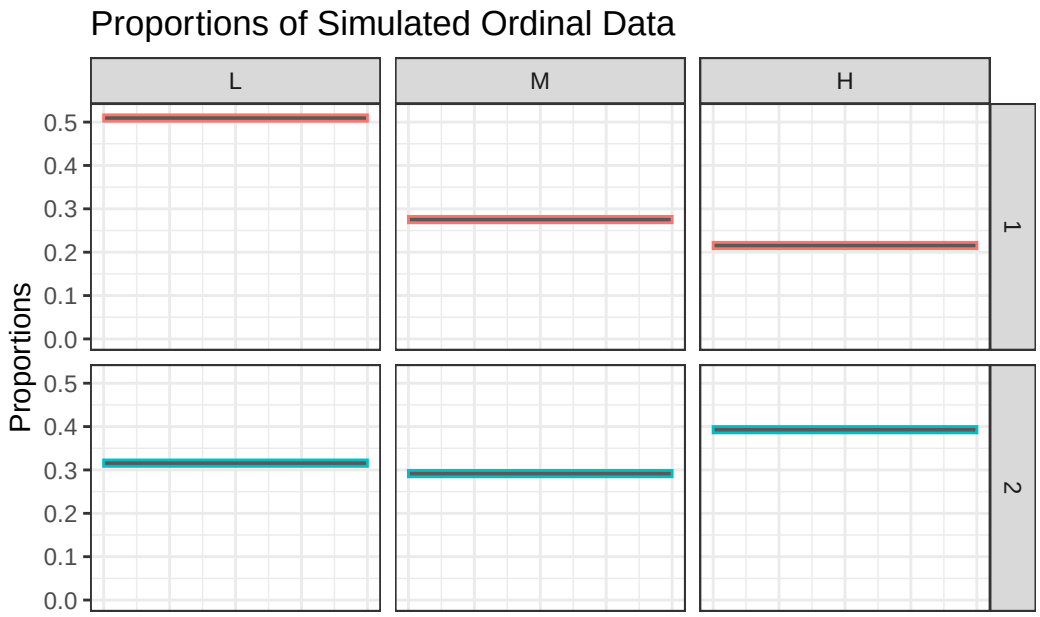
```
set.seed(09152025)
pi_1 <- c(.5, .3, .2)
pi_2 <- c(.3, .3, .4)

n <- 1000
pi_g1 <- .6
n1 <- rbinom(1, n, pi_g1)
n2 <- n - n1
y1 <- rmultinom(1, n1, pi_1)
y2 <- rmultinom(1, n2, pi_2)
```

2. Visualize the data

```
library(tidyverse)
tibble(vals = c(y1, y2),
       group = rep(c('1','2'), each =3),
       level = ordered(rep(c('L','M','H'), 2), levels = c('L','M','H'))) |>
  ggplot(aes(y = vals, color = group)) +
          geom_bar() +
  facet_grid(group~ level) +
  ylim(0, NA) +
  theme_bw() +
  xlab('') +
  ylab('counts') +
  theme(legend.position = 'none',
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  ggtitle('Counts of Simulated Ordinal Data')
```




```
tibble(vals = c(y1, y2),
       group = rep(c('1','2'), each =3),
       level = ordered(rep(c('L','M','H'), 2), levels = c('L','M','H'))) |>
  group_by(group) |>
  mutate(proportion = vals / sum(vals)) |>
  ggplot(aes(y = proportion, color = group)) +
          geom_bar() +
  facet_grid(group~ level) +
  ylim(0, NA) +
  theme_bw() +
  xlab('') +
  ylab('Proportions') +
  theme(legend.position = 'none',
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  ggtitle('Proportions of Simulated Ordinal Data')
```



3. Create Contingency Table

```
library(knitr)
tibble(group = c('1','2'),
       L = c(y1[1],y2[1]),
       M = c(y1[2],y2[2]),
       H = c(y1[3],y2[3])) |>
  kable()
```

group	L	M	H
1	298	161	126
2	131	121	163

4. Run a χ^2 test and interpret the results. Clearly articulate what you are testing here (avoid statistical lingo).

We are testing to see whether there is any association between our two categorical variables. In other words, if we know we are group 1 does that change our belief about the probability of L - M - H response.

```r
dat.mat <- as.matrix(tibble(L = c(y1[1],y2[1]),
       M = c(y1[2],y2[2]),
       H = c(y1[3],y2[3])))

chisq.test(dat.mat)
```

```
    Pearson's Chi-squared test

data:  dat.mat
X-squared = 47.905, df = 2, p-value = 3.96e-11
```

We see that there is some clear relationship between the two variables.

5. Create another table that includes the residual values (observed values - expected values) from the χ^2 test. How do these values contribute to the test statistic in the previous part?

Recall the expected values (under the no association hypothesis) can be generated as a product of the marginal values (and overall sample size.)

```r
group_marg <- tibble(vals = c(y1, y2),
       group = rep(c('1','2'), each =3),
       level = ordered(rep(c('L','M','H'), 2), levels = c('L','M','H'))) |>
  group_by(group)|>
  summarize(vals = sum(vals)) |>
  mutate(marg_prop = vals / sum(vals) )

level_marg <- tibble(vals = c(y1, y2),
       group = rep(c('1','2'), each =3),
       level = ordered(rep(c('L','M','H'), 2), levels = c('L','M','H'))) |>
  group_by(level)|>
  summarize(vals = sum(vals)) |>
  mutate(marg_prop = vals / sum(vals) )



tibble(group = c('1','2'),
       L = c(n * level_marg[1,3] |> pull() * group_marg[1,3] |> pull(),
             n * level_marg[1,3] |> pull() * group_marg[2,3] |> pull()),
```

```
      M = c( n * level_marg[2,3] |> pull() * group_marg[1,3] |> pull(),
             n * level_marg[2,3] |> pull() * group_marg[2,3] |> pull()),
      H = c( n * level_marg[3,3] |> pull() * group_marg[1,3] |> pull(),
             n * level_marg[3,3] |> pull() * group_marg[2,3] |> pull())) |>
  kable(caption = 'Expected Counts')
```

Table 3: Expected Counts

group	L	M	H
1	250.965	164.97	169.065
2	178.035	117.03	119.935

```
tibble(group = c('1','2'),
       L = c(y1[1] - n * level_marg[1,3] |> pull() * group_marg[1,3] |> pull(),
             y2[1] - n * level_marg[1,3] |> pull() * group_marg[2,3] |> pull()),
       M = c(y1[2] - n * level_marg[2,3] |> pull() * group_marg[1,3] |> pull(),
             y2[2] - n * level_marg[2,3] |> pull() * group_marg[2,3] |> pull()),
       H = c(y1[3] - n * level_marg[3,3] |> pull() * group_marg[1,3] |> pull(),
             y2[3] - n * level_marg[3,3] |> pull() * group_marg[2,3] |> pull())) |>
  kable(caption = 'Residual Counts')
```

Table 4: Residual Counts

group	L	M	H
1	47.035	-3.97	-43.065
2	-47.035	3.97	43.065

The group 1 has more low counts than expected; whereas, group 2 has more high counts than expected.

The previous analysis treats the data as nominal, and is still valid, but doesn't explicitly account for the ordinal structure of the data. We can do something that mimics correlation with continuous data.

- First, we need to define scores for the each of the responses. An example would be low = 1, medium = 2, high = 3

- Then define $\bar{u} = \sum_i u_i p_{i+}$ to denote the sample mean of the row scores and $\bar{v} = \sum_j v_i p_{+j}$ to be the sample mean of the column scores.

$$r = \frac{\sum_{ij}(u_i - \bar{u})(v_j - \bar{v})p_{ij}}{\sqrt{\left[\sum_i (u_i - \bar{u})^2 p_{i+}\right]\left[\sum_j (v_j - \bar{v})^2 p_{+j}\right]}}$$

- $M^2 = (n-1)r^2$ and M^2 has an approximate χ^2 distribution with 1 degree of freedom (with large n).

Consider the following table.

group	L	M	H
1	298	161	126
2	131	163	121

calculate r and M^2.

```
r1 <- c(298, 161, 126)
r2 <- c(131, 163, 121)
n <- sum(r1 + r2)
pi_iplus <- c(sum(r1) / n, sum(r2) / n)
u_bar <- sum(pi_iplus * c(1,2))

c1 <- c(298, 131)
c2 <- c(161, 163)
c3 <- c(126, 121)

pi_plusj <- c(sum(c1) / n , sum(c2) / n, sum(c3) / n)
v_bar <- sum(pi_plusj * c(1, 2, 3))

pij <- matrix(c(r1, r2), nrow = 2, byrow = T) / n

numerator <- (1 - u_bar) * (1 -v_bar) * pij[1,1] +
  (2 - u_bar) * (1 - v_bar) * pij[2,1] +
  (1 - u_bar) * (2 - v_bar) * pij[1,2] +
  (2 - u_bar) * (2 - v_bar) * pij[2,2] +
  (1 - u_bar) * (3 - v_bar) * pij[1,3] +
  (2 - u_bar) * (3 - v_bar) * pij[2,3]

u_denom <- (1 - u_bar)^2 * pi_iplus[1] +
  (2 - u_bar)^2 * pi_iplus[2]
v_denom <- (1 - v_bar)^2 * pi_plusj[1] +
  (2 - v_bar)^2 * pi_plusj[2]+
  (3 - v_bar)^2 * pi_plusj[3]

r_val  <- numerator / sqrt(u_denom * v_denom)
```

- r is 0.1658726
- $M^2 = 27.4861966$ which gives a p-value of $(1.3452611 \times 10^{-7})$ implies there is an association between the variables.

Fisher's Exact Test

The story behind Fisher's exact test is that a lab mate claimed they could taste the difference between cups of tea that had milk added first or last. So an experiment was designed with 4 coups of each. We will do something similar on Thursday.

This setting requires the use of the hypergeometric distribution. Let n_{ij} be the cell counts for the i^{th} row and j^{th} column.

	Milk(guess)	Tea (guess)	Total
Milk(actual)	n_{11}	n_{12}	n_{1+}
Tea (guess)	n_{21}	n_{22}	n_{2+}
Total	n_{+1}	n_{+2}	

Given we know the marginal counts, the guesses, n_{11} completely determines the other values (a bit like sudoku).

Using a hypergeometric distribution we can estimate the probability of the possible values for n_{11} as

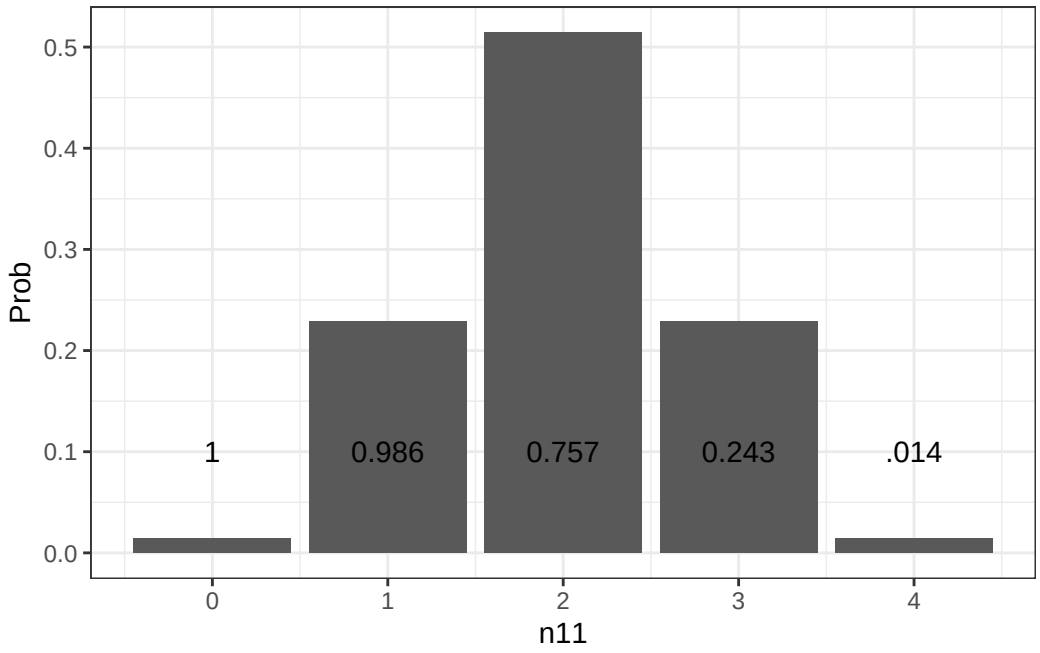
$$Pr(n_{11}) = \frac{\binom{n_{1+}}{n_{11}}\binom{n_{2+}}{n_{+1}-n_{11}}}{\binom{n}{n_{+1}}}$$

```
tibble(n11 = c(0, 1, 2, 3, 4),
       Prob = c(choose(4,0) * choose(4,4) / choose(8,4),
                choose(4,1) * choose(4,3) / choose(8,4),
                choose(4,2) * choose(4,2) / choose(8,4),
                choose(4,3) * choose(4,1) / choose(8,4),
                choose(4,4) * choose(4,0) / choose(8,4))
       ) |> kable()
```

n11	Prob
0	0.0142857
1	0.2285714
2	0.5142857
3	0.2285714
4	0.0142857

How do we get p-values from this table?

```
tibble(n11 = c(0, 1, 2, 3, 4),
       Prob = c(choose(4,0) * choose(4,4) / choose(8,4),
                choose(4,1) * choose(4,3) / choose(8,4),
                choose(4,2) * choose(4,2) / choose(8,4),
                choose(4,3) * choose(4,1) / choose(8,4),
                choose(4,4) * choose(4,0) / choose(8,4))
       ) |>
 ggplot(aes(y = Prob, x = n11)) +
 geom_col() + theme_bw() +
 annotate('text', x = 4, y = .1, label = '.014') +
 annotate('text', x = 3, y = .1, label = as.character(.229 + .014)) +
 annotate('text', x = 2, y = .1, label = as.character(.514 + .229 + .014)) +
 annotate('text', x = 1, y = .1, label = as.character(.229 + .514 + .229 + .014)) +
 annotate('text', x = 0, y = .1, label = as.character(.014 + .229 + .514 + .229 + .014))
```


0.5
0.4
0.3
0.2
0.1
0.0
Prob
1
0.986
0.757
0.243
.014
0
1
2
3
4
n11