Lesson 20 Companion

Research question:

I'm going to utilize the second nightlight example (three categories for both variables) for this document. Recall that the research question was whether or not the amount of light in a child's room is associated with refraction result. The hypothesis for this analysis are:

H_0 : There is no association between the level of light and refraction result.

H_a : There is an association between the level of light and refraction result.

You'll notice that this chapter discusses two different statistics for these sort of research questions: *mean absolute difference* (MAD) and the *chi-squared statistic* (χ^2). I believe the authors chose to do this because the MAD statistic is a bit easier to connect to what you intuitively want to measure in regard to the difference between groups. The book points out several strengths of the χ^2 statistic here so we will focus on that.

Our hypothesis test is a little different here because we aren't directly looking to make an inference about a population parameter. The p-values we get from our various tests should be used to judge the amount of evidence we have against the null hypothesis above.

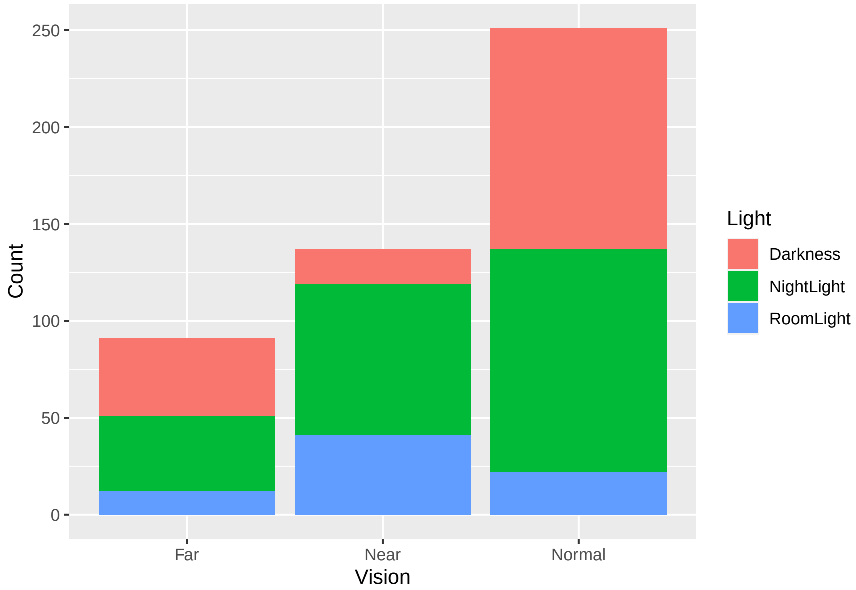
Data exploration

This data set is downloaded in a summary table form from the book's website.

```r
library(tidyverse)

night = read_table2("http://www.isi-stats.com/isi/data/chap8/NightLight2.txt",
                    col_names = c("Vision", "Darkness", "NightLight", "RoomLight"),
                    skip = 1)

head(night)
```

```
## # A tibble: 3 x 4
##   Vision Darkness NightLight RoomLight
##   <chr>     <dbl>      <dbl>     <dbl>
## 1 Near         18         78        41
## 2 Normal      114        115        22
## 3 Far          40         39        12
```

```r
night %>%
  pivot_longer(c(Darkness, NightLight, RoomLight),
               names_to = "Light", values_to = "Count") %>%
  ggplot(aes(x = Vision, y = Count, fill = Light)) +
  geom_bar(position="stack", stat="identity")
```



You can see that I had to do a **pivot_longer()** here in order to get the data into a format we could use for the bar chart. From this bar chart, it appears as if the majority of the participants in the study had normal refraction followed by nearsightedness and farsightedness. The *NightLight* category accounted for the majority of cases in each group with the *RoomLight* category being the least prevalent in each group except nearsightedness. Overall, it's difficult to make any firm conclusions from the bar chart alone because there is a lot going on. Good thing we have some quantitative methods to use!

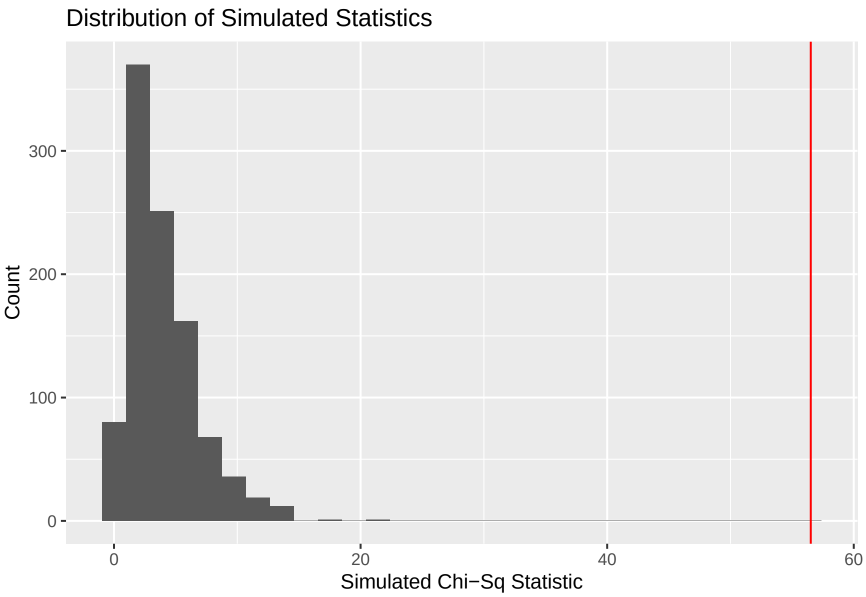
Simulation-based approach:

For the simulation-based approach, I am going to do some modifications to our downloaded data again. Once that is complete, I can "re-assign" a light type to each participant so I can build my null distribution. First I want to walk through the calculation of the sample statistic (χ^2).

```
sample_stat = night %>%
  #Create a total obs. column and a expected proportions column for each
  # refraction result
  mutate(Total = Darkness + NightLight + RoomLight,
         Expected = Total / (sum(Darkness, NightLight, RoomLight))) %>%
  #Calculate the expected # of obs. by multiplying each light type and
  # the expected proportion for that refraction result
  mutate(Darkness_Exp = sum(Darkness) * Expected,
         RoomLight_Exp = sum(RoomLight) * Expected,
         NightLight_Exp = sum(NightLight) * Expected) %>%
  #Calculate the parts of the chi^2 stat by light type... this step
  # isn't represented in your book because they do it by hand
  mutate(Darkness_chi = ((Darkness - Darkness_Exp) ^ 2) / Darkness_Exp,
         RoomLight_chi = ((RoomLight - RoomLight_Exp) ^ 2) / RoomLight_Exp,
         NightLight_chi = ((NightLight - NightLight_Exp) ^ 2) / NightLight_Exp) %>%
  #Sum up all the rows for the final stat
  summarise(chi_sq = sum(Darkness_chi, RoomLight_chi, NightLight_chi))
```

Now that we have the sample statistic we can start to build our null distribution.

```
#Pivot longer and then create a row for each participant
# that lists the light type and refraction result
longer_night = night %>%
  pivot_longer(c(Darkness, NightLight, RoomLight),
              names_to = "Light", values_to = "Count") %>%
  mutate(obs = map(Count, ~rep_len(1, .x))) %>%
  unnest() %>%
  select(Vision, Light)

replications_dataframe = NULL

num_reps = 1000

for (i in 1:num_reps){

  #Scramble the light type for each trial
  trial_night = longer_night %>%
    mutate(New_Light = sample(Light, size = n(), replace = FALSE)) %>%
    count(Vision, New_Light) %>%
    pivot_wider(names_from = New_Light, values_from = n)

  #Calculate the trial chi-squared stat using the same method as above
  trial_stat = trial_night %>%
    mutate(Total = Darkness + NightLight + RoomLight,
         Expected = Total / (sum(Darkness, NightLight, RoomLight))) %>%
    mutate(Darkness_Exp = sum(Darkness) * Expected,
         RoomLight_Exp = sum(RoomLight) * Expected,
         NightLight_Exp = sum(NightLight) * Expected) %>%
    mutate(Darkness_chi = ((Darkness - Darkness_Exp) ^ 2) / Darkness_Exp,
         RoomLight_chi = ((RoomLight - RoomLight_Exp) ^ 2) / RoomLight_Exp,
         NightLight_chi = ((NightLight - NightLight_Exp) ^ 2) / NightLight_Exp) %>%
    summarise(trial_stat = sum(Darkness_chi, RoomLight_chi, NightLight_chi))

  replications_dataframe = rbind(replications_dataframe, trial_stat)

}

replications_dataframe %>%
  ggplot(aes(x = trial_stat)) +
  geom_histogram() +
  labs(x = "Simulated Chi-Sq Statistic", y = "Count",
       title = "Distribution of Simulated Statistics") +
  geom_vline(xintercept = sample_stat[[1]], color = "red")
```



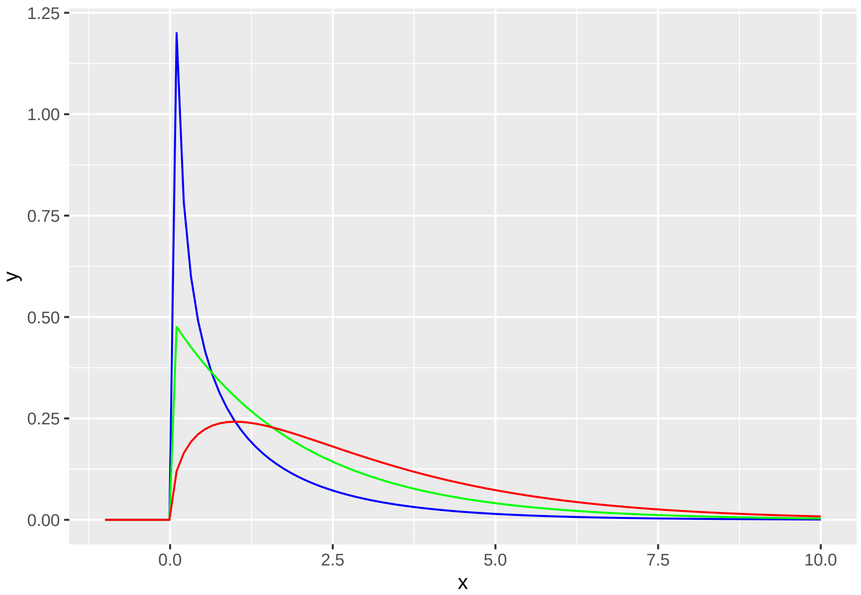
```
replications_dataframe %>%
  summarise(pvalue = sum(trial_stat >= sample_stat[[1]]) / n())
```

```
## # A tibble: 1 x 1
##   pvalue
##    <dbl>
## 1      0
```

Two things should be very clear from the histogram above: we're not in "normal-ville" anymore and our sample statistic is very extreme. This matches the p-value of 0 that we've calculated meaning it is very unlikely to see this result if there is no association between light type and refraction result. This sample statistic gives us very strong evidence against the null hypothesis.

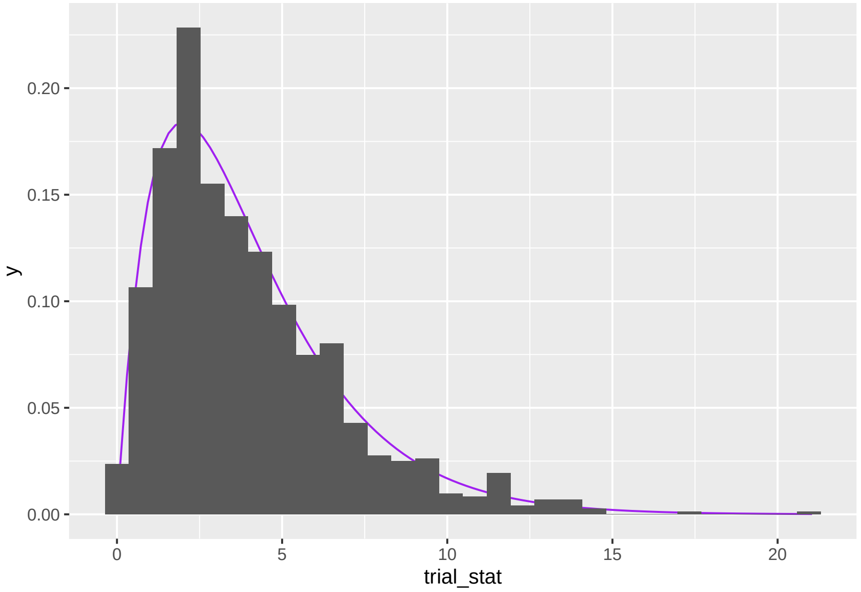
As mentioned, the χ^2 statistic is not distributed according to the normal distribution. Instead it's distributed according to the... χ^2 distribution. This distribution has a single parameter (degrees of freedom) and a couple examples look something like this:

```
ggplot(data.frame(x = c(-1,10)), aes(x = x)) +
  stat_function(fun = dchisq,
                args = list(df = 1),
                color = "blue") +
  stat_function(fun = dchisq,
                args = list(df = 2),
                color = "green") +
  stat_function(fun = dchisq,
                args = list(df = 3),
                color = "red")
```



You should notice that there is no such thing as a negative χ^2 value. The degrees of freedom are found by subtracting one for the number of categories for each variable and multiplying the results together.

```
df = (3 - 1) * (3 - 1)

replications_dataframe %>%
  ggplot(aes(x = trial_stat)) +
  stat_function(fun = dchisq,
                args = list(df = df),
                color = "purple") +
  geom_histogram(aes(y = ..density..))
```



Theory-based approach:

Now that I've kind of ruined the surprise for how we will represent the null distribution without simulating, let's discuss the validity condition for using the χ^2 distribution. It is that there are at least 10 observations in each category for each variable (at least 10 observations in each cell of the summary table) and clearly meet that condition in this example.

Once again, I will present two methods of computing the p-value for this hypothesis test. The first is finding the area under the χ^2 distribution curve that is beyond our sample statistic.

```
1 - pchisq(sample_stat[[1]], df = df)
```

```
## [1] 1.564893e-11
```

The second uses the **chisq.test()** function built in to *R*.

```
chisq.test(x = longer_night$Vision, y = longer_night$Light)
```

```
##
##  Pearson's Chi-squared test
##
## data:  longer_night$Vision and longer_night$Light
## X-squared = 56.513, df = 4, p-value = 1.565e-11
```

I'm sure you don't need a reminder at this point that you won't have p-values that are exactly zero when dealing with the theory-based tests because the tails of these distributions go on forever (AKA there is always area under the curve).

Interpretation of our results:

As you see from both the simulation-based and theory-based approaches, our p-values offer very strong evidence against the null hypothesis which implies that there appears to be an association between light type and refraction result. Now that we have a significant result, what's the next step in our analysis? What if we'd like try to narrow down our categories? I think the first step in further analysis should be completing confidence intervals for the difference in proportion parameters for pairwise comparisons.

These confidence intervals are little trickier than dealing with two groups, but are manageable. Unfortunately, our situation is even more complicated because we have three categories in both our explanatory and response variables. Because of this, I would instead complete three pairwise z-tests using two light type categories at a time.

The **pairwise.prop.test()** function here gives us a p-value for pairwise comparisons between two groups (two categories our of explanatory).

You can see that I've removed the *Darkness* observations for our first test. The **pairwise.prop.test()** function expects to see a list of "successes" in x and a list of the total number of observations in n.

As the table that's provided as output from this function uses numbers instead of categories, you can look at the variables created (*room_v_night*, *room_v_dark*, and *night_v_room*) to sort out what refraction result each number refers to.

```
room_v_night = night %>%
  select(-Darkness) %>%
  mutate(Total = NightLight + RoomLight) %>%
  pivot_longer(c(NightLight, RoomLight),
               names_to = "Light", values_to = "Count") %>%
  filter(Light == "RoomLight") %>%
  select(Vision, Count, Total)

pairwise.prop.test(x = room_v_night$Count, n = room_v_night$Total)
```

```
##
##  Pairwise comparisons using Pairwise comparison of proportions
##
## data:  room_v_night$Count out of room_v_night$Total
##
##   1      2
## 2 0.0033 -
## 3 0.4385 0.4385
##
## P value adjustment method: holm
```

```r
room_v_dark = night %>%
  select(-NightLight) %>%
  mutate(Total = Darkness + RoomLight) %>%
  pivot_longer(c(Darkness, RoomLight),
               names_to = "Light", values_to = "Count") %>%
  filter(Light == "RoomLight") %>%
  select(Vision, Count, Total)

pairwise.prop.test(x = room_v_dark$Count, n = room_v_dark$Total)
```

```
##
##  Pairwise comparisons using Pairwise comparison of proportions
##
## data:  room_v_dark$Count out of room_v_dark$Total
##
##   1       2
## 2 2.7e-12 -
## 3 5.3e-06 0.37
##
## P value adjustment method: holm
```

```r
night_v_dark = night %>%
  select(-RoomLight) %>%
  mutate(Total = Darkness + NightLight) %>%
  pivot_longer(c(Darkness, NightLight),
               names_to = "Light", values_to = "Count") %>%
  filter(Light == "NightLight") %>%
  select(Vision, Count, Total)

pairwise.prop.test(x = night_v_dark$Count, n = night_v_dark$Total)
```

```
##
##  Pairwise comparisons using Pairwise comparison of proportions
##
## data:  night_v_dark$Count out of night_v_dark$Total
##
##   1       2
## 2 1.2e-06 -
## 3 3.5e-05 1
##
## P value adjustment method: holm
```

Rather than using this **pairwise.prop.test()** you could also further reduce your analysis down to two groups and perform a set two-sample z-tests (which you should be familiar with). This just comes at the cost of having to do more tests.

Please note that continued analysis like this is only warranted if you get a significant result and if you are really trying to drill down into your data. As this is an introductory class, we will generally avoid doing analysis of this depth but at least you know that it is within your ability to do so.

But wait, can we still use the χ^2 statistic in the simulation-based approach if we don't meet the validity conditions?

This is a unique situation in that the statistic we are using in simulation-based approach is associated with the distribution we use in the theory-based approach. Usually we utilize the *central limit theorem* to assume that our sample statistic will be distributed according to the normal distribution as long as the validity conditions are met. Here we have used the $\chi 2$ statistic in both approaches. If we don't meet the conditions to utilize this distribution to represent our null distribution in the theory-based approach, does the associated statistic still do a good job in the simulation-based approach? Honestly, I don't know. When I called my "phone-a-statistician" they said that I was crazy and I should just use Fisher's exact test for this purpose so let's take a brief look at that.

This test has an interesting back story so if you'd like to know more look up the "lady tasting tea" experiment. While it seems feasible to compute the p-value for this test by hand for a 2 × 2 table, anything bigger than that is infeasible. We will simple use *R*'s **fisher.test()** to calculate our p-value for us. You can see that I need to make the columns that contain values into a matrix in order to use this function.

```r
night_mat = data.matrix(night[2:4])

fisher.test(night_mat,  workspace = 2e8)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  night_mat
## p-value = 3.068e-12
## alternative hypothesis: two.sided
```

So when should we use Fisher's Exact Test in place of the theory-based test offered by **chisq.test()**? Technically, Fisher's offers a more defensible result if you fail to meet the validity conditions of the theory-based test. Practically, however, the validity conditions for the theory-based test represent a very low bar to overcome and I would be more concerned about the generalizability of my results if I didn't have at least ten observations in each category.