431 Class 18

github.com/THOMASELOVE/2019-431

2019-10-31

Today's Setup and Data

```r
library(exact2x2); library(PropCIs) # new today

library(Epi)
library(magrittr); library(janitor)
library(here); library(tidyverse)

source(here("R", "Love-boost.R"))

dm431 <- readRDS(here("data", "dm431.Rds"))
```

Example B: Statin use in Medicaid vs. Uninsured

In the dm431 data, suppose we want to know whether statin prescriptions are more common among Medicaid patients than Uninsured subjects. So, we want a two-way table with "Medicaid", "Statin" in the top left.

```
dm431 %>%
  filter(insurance %in% c("Medicaid", "Uninsured")) %>%
  tabyl(insurance, statin)
```

```
  insurance  0  1
 Commercial  0  0
   Medicaid 17 83
   Medicare  0  0
  Uninsured 15 29
```

But we want the tabyl just to show the levels of insurance we're studying. . .

Obtaining a 2x2 Table from a data frame

We want to know whether statin prescriptions are more common among Medicaid patients than Uninsured subjects.. So, we want a two-way table with "Medicaid", "Uninsured" in the top left.

```
dm431 %>%
  filter(insurance %in% c("Medicaid", "Uninsured")) %>%
  droplevels() %>%
  tabyl(insurance, statin)
```

```
 insurance  0  1
  Medicaid 17 83
 Uninsured 15 29
```

But we want Medicaid in the top row (ok) and "statin = yes" in the left column (must fix)...

Building and Releveling Factors in the data frame

```
exampleB <- dm431 %>%
  filter(insurance %in% c("Medicaid", "Uninsured")) %>%
  droplevels() %>%
  mutate(insur_f = fct_relevel(insurance, "Medicaid"),
         statin_f = fct_recode(factor(statin),
                               on_statin = "1", no_statin = "0"),
         statin_f = fct_relevel(statin_f, "on_statin"))

exampleB %>% tabyl(insur_f, statin_f)
```

```
   insur_f on_statin no_statin
  Medicaid        83        17
 Uninsured        29        15
```

Since Medicaid was already on top, we didn't *have to* set `insur_f`.

Adorning the tabyl with % using row as denominator

```
exampleB %>% tabyl(insur_f, statin_f) %>%
  adorn_totals(where = c("row", "col")) %>%
  adorn_percentages(denom = "row") %>%
  adorn_pct_formatting(digits = 1) %>%
  adorn_ns(position = "front") %>%
  adorn_title(row = "Insurance", col = "Statin Status")
```

```
           Statin Status
 Insurance     on_statin  no_statin        Total
  Medicaid    83 (83.0%) 17 (17.0%) 100 (100.0%)
 Uninsured    29 (65.9%) 15 (34.1%)  44 (100.0%)
     Total   112 (77.8%) 32 (22.2%) 144 (100.0%)
```

Running `twoby2` against a data set

The `twoby2` function from the `Epi` package can operate with tables (but not, alas, `tabyls`) generated from data.

Original Data

```
twoby2(exampleB %$% table(insur_f, statin_f))
```

(output on next slide)

With Bayesian Augmentation

```
twoby2(exampleB %$% table(insur_f, statin_f) + 1)
```

(output on the slide after that)

Complete `twoby2` for Example B

```
2 by 2 table analysis:
------------------------------------------------------
Outcome   : on_statin
Comparing : Medicaid vs. Uninsured

          on_statin no_statin    P(on_statin) 95% conf. interval
Medicaid         83        17          0.8300    0.7434   0.8916
Uninsured        29        15          0.6591    0.5090   0.7829

                                    95% conf. interval
             Relative Risk: 1.2593    1.0003   1.5854
         Sample Odds Ratio: 2.5254    1.1202   5.6933
Conditional MLE Odds Ratio: 2.5074    1.0252   6.1298
    Probability difference: 0.1709    0.0218   0.3307

             Exact P-value: 0.0299
        Asymptotic P-value: 0.0255
------------------------------------------------------
```

twoby2 for Example B (with Bayesian augmentation)

```
2 by 2 table analysis:
------------------------------------------------------
Outcome   : on_statin
Comparing : Medicaid vs. Uninsured

          on_statin no_statin    P(on_statin) 95% conf. interval
Medicaid         84        18          0.8235    0.7372   0.8859
Uninsured        30        16          0.6522    0.5055   0.7748

                                    95% conf. interval
             Relative Risk: 1.2627    1.0039   1.5883
         Sample Odds Ratio: 2.4889    1.1273   5.4951
Conditional MLE Odds Ratio: 2.4721    1.0368   5.8963
    Probability difference: 0.1714    0.0233   0.3285

             Exact P-value: 0.0336
        Asymptotic P-value: 0.0240
------------------------------------------------------
```

Comparing Proportions using Paired Samples (Course Notes Chapter 24)

`dm431` Example C.

Among the current Commercially insured subjects, compare the proportion with A1c below 8 to the proportion for the same patients two years ago.

```
dm431 %>% filter(insurance == "Commercial") %>%
    count(a1c_old < 8, a1c < 8)
```

```
# A tibble: 7 x 3
  `a1c_old < 8` `a1c < 8`     n
  <lgl>         <lgl>     <int>
1 FALSE         FALSE        31
2 FALSE         TRUE         15
3 TRUE          FALSE        30
4 TRUE          TRUE         82
5 NA            FALSE         1
6 NA            TRUE          3
7 NA            NA            2
```

- How might we rearrange this information? Exposure? Outcome?

How many subjects do we have?

- How many commercial subjects provide us with A1c values at each time point?

```
dm431 %>% filter(complete.cases(a1c_old, a1c)) %>%
  filter(insurance == "Commercial") %>% nrow()
```

```
[1] 158
```

- How many A1c values did we obtain from those subjects?

What is our design here?

- Here are four of the subjects in this group:

```
# A tibble: 4 x 4
  subject insurance    a1c a1c_old
  <chr>   <fct>      <dbl>   <dbl>
1 S-001   Commercial   6.3    11.4
2 S-004   Commercial   6.5     5.8
3 S-012   Commercial  12.2    11.3
4 S-013   Commercial   8.1     6.7
```

- What is our outcome?
- What are the two exposure groups?
- Are these samples paired or independent?

dm431 Example C, rearranged

```
dm431 %>% filter(insurance == "Commercial") %>%
    mutate(now_stat = ifelse(a1c < 8,
                         "below_8_now", "high_now"),
           old_stat = ifelse(a1c_old < 8,
                         "old_below_8", "old_high")) %>%
    tabyl(now_stat, old_stat)
```

```
   now_stat old_below_8 old_high NA_
 below_8_now          82       15   3
    high_now          30       31   1
        <NA>           0        0   2
```

- What should we do about the missingness?

dm431 Example C (dropping the missing data)

```
tableC <- dm431 %>% filter(insurance == "Commercial") %>%
    filter(complete.cases(a1c, a1c_old)) %>%
    mutate(now_stat = ifelse(a1c < 8,
                             "below_8_now", "high_now"),
           old_stat = ifelse(a1c_old < 8,
                             "old_below_8", "old_high")) %>%
    tabyl(old_stat, now_stat)

tableC %>%
    adorn_totals(where = c("row", "col"))
```

```
    old_stat below_8_now high_now Total
 old_below_8          82       30   112
    old_high          15       31    46
       Total          97       61   158
```

Concordant and Discordant Pairs

```
tableC
```

```
   old_stat below_8_now high_now
old_below_8          82       30
   old_high          15       31
```

When the same result is observed in the old and new data, we call that *concordant*. When there's a change, we call that *discordant*.

We have 82 + 31 = 113 subjects with concordant results here, and 15 + 30 = 45 subjects with discordant results. Each subject provides a pair of A1c results.

It turns out that the discordant pairs, generally, will be of maximum interest to us, as they give us an indication of the relatively likelihood of A1c increasing vs. A1c decreasing, while the concordant results don't allow us to make any meaningful progress in building our comparison.

The McNemar Odds Ratio

```
   old_stat below_8_now high_now
old_below_8          82       30
   old_high          15       31
```

The general paired data 2x2 table is:

```
a  b
c  d
```

- We have b = 30 subjects with good results two years ago but high ones (A1c >= 8) now.
- We have c = 15 subjects with high results two years ago but good ones (A1c < 8) now.

The McNemar odds ratio is the larger of the two ratios (either c/b or b/c) that we can form with these data.

So in our case, it is $30/15 = 2.0$

Cohen's g statistic

```
   old_stat below_8_now high_now
old_below_8          82       30
   old_high          15       31
```

Cohen's g statistic is also measured using the discordant counts. First, we identify the larger of $\frac{b}{b+c}$ and $\frac{c}{b+c}$. Cohen's g is that value minus 0.5. In our case,

- b = 30 subjects with good results two years ago but high ones (A1c $>=$ 8) now, and
- c = 15 subjects with high results two years ago but good ones (A1c $<$ 8) now.

$$g = \frac{30}{45} - 0.5 = 0.167$$

Cohen's g is just a simple function of the McNemar odds ratio, so we'll focus on that.

Estimating the CI for the McNemar Odds Ratio

To estimate the CI for the McNemar odds ratio, we use the exact2x2 function from the exact2x2 package.

```
dm431 %>% filter(insurance == "Commercial") %>%
    filter(complete.cases(a1c, a1c_old)) %>%
    mutate(now_stat = ifelse(a1c < 8,
                        "below_8_now", "high_now"),
           old_stat = ifelse(a1c_old < 8,
                        "old_below_8", "old_high")) %$%
    exact2x2(old_stat, now_stat, paired = TRUE,
             conf.int = TRUE, conf.level = 0.95)
```

Results on the next slide...

95% CI for the McNemar Odds Ratio

```
    Exact McNemar test (with central confidence
    intervals)

data:  old_stat and now_stat
b = 30, c = 15, p-value = 0.0357
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.042886 3.999858
sample estimates:
odds ratio
         2
```

Estimating the Difference in Proportions

Among current Commercial subjects, compare the proportion with A1c below 8 to the proportion for the same patients two years ago.

old_stat	below_8_now	high_now	Total
old_below_8	82	30	112
old_high	15	31	46
Total	97	61	158

- Now, 97/158 (0.614) have A1c below 8.
- Two years ago, 112/158 (0.709) had A1c below 8.
- The sample difference is -0.095

Can we build a confidence interval for the difference of those two proportions that takes the pairing into account? **Yes**, using some tools from the `PropCIs` package.

Wald confidence interval approach

```
diffpropci.Wald.mp(b = 30, c = 15, n = 158, conf.level = 0.95)
```

```
data:

95 percent confidence interval:
 -0.17682361 -0.01304981
sample estimates:
[1] -0.09493671
```

Be careful to compare the right things. This is the difference between the rate of success (A1c < 8) now, and the rate of success (A1c < 8) two years ago. The current rate appears to be a bit lower.

Agresti-Min confidence interval approach

It's also possible to run an Agresti-Min approach, although I usually stick with the Wald method.

```
diffpropci.mp(b = 30, c = 15, n = 158, conf.level = 0.95)
```

```
data:

95 percent confidence interval:
 -0.17555222 -0.01194778
sample estimates:
[1] -0.09375
```

The two intervals produce slightly different point and interval estimates, because they are making different sorts of approximations.

What if we looked at all subjects?

This table includes all subjects, not just those with commercial insurance.

```
              now_stat
   old_stat below_8_now high_now Total
old_below_8         227       56   283
   old_high          47       86   133
      Total         274      142   416
```

McNemar Odds Ratio 95% Confidence Interval

```r
dm431 %>% filter(complete.cases(a1c, a1c_old)) %>%
    mutate(now_stat = ifelse(a1c < 8,
                             "below_8_now", "high_now"),
           old_stat = ifelse(a1c_old < 8,
                             "old_below_8", "old_high")) %$%
    exact2x2(old_stat, now_stat, paired = TRUE,
             conf.int = TRUE, conf.level = 0.95)
```

```

    Exact McNemar test (with central confidence
    intervals)

data:  old_stat and now_stat
b = 56, c = 47, p-value = 0.4307
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.7940648 1.7948142
```

Comparing % meeting A1c < 8 then and now

old_stat	below_8_now	high_now	Total
old_below_8	227	56	283
old_high	47	86	133
Total	274	142	416

Across all insurance groups,

- Now, 274/416 (0.659) have A1c below 8.
- Two years ago, 283/416 (0.680) had A1c below 8.
- The sample difference is -0.022.
- The Wald 95% CI for that difference is (-0.069, 0.026)

Coming Soon

- Comparing More than 2 Means with Independent Samples: Analysis of Variance
- Power and Sample Size Ideas
- Working with Larger Contingency Tables (Chi-Square Tests of Independence)
- Mantel-Haenszel Procedures for Three-Way Tables