Lab 3

Public Health 241: Statistical Analysis of Categorical Data

YOUR NAME / YOUR STUDENT ID HERE

TODAY'S DATE

In this lab, we'll cover the following topics:

- Homework 2, in R with the `epiR` and `epitools` packages
- How to carry out a χ^2 test and obtain a p-value

Many of these topics and code smippets will be useful for Homework 4.

1. Homework 2, Revisited

1.1 *Problem 1.*

You worked with this data in `lab02`, but let's briefly run through how to solve a problem like this in R. In order to recreate the table that appears on the assignment, we must learn how to create a `data.frame`, or a table.

Pay close attention to the following code:

```
population <- data.frame("disease"=c(140, 180), "no.disease"=c(60, 620))
population
```

```
##   disease no.disease
## 1     140         60
## 2     180        620
```

Later in this lab, you will learn how to calculate all of these values (odds ratio, relative risk, etc.) through a statistical package called `epiR`, but for now, we can calculate these manually quite easily using dataframe access methods:

- The odds of disease among men:

```
prob.disease.given.man <- population$disease[1]/(population$disease[1]+population$no.disease[1])
prob.disease.given.man / (1 - prob.disease.given.man)
```

```
## [1] 2.333333
```

and so on. Practice doing these calculations on the rest of problem 1 (b,c,d) and problem 2.

2. Cohort and Case-Control Studies using `epi.2by2`

From lecture, we have a foundational understanding of the difference between the cohort and case-control studies, which can be easily simulated in R using a function from the `epiR` package called `epi.2by2`. You should be aware of the limitations of each study design, as described in Chapter 5 of this course's textbook. Let's try this out now.

IMPORTANT: The `data` argument in both studies must be a `table` object in order for the function to run correctly. If you have a dataframe representation of your data, you must turn it into matrix then table. An example is shown below.

2.1 Cohort Study:

Generic: `epi.2by2(data, method = "cohort.count", conf.level = 0.95, units = 100, homogeneity = "breslow.day", outcome = "as.columns")`

- Note that if you're dealing with a cohort study's data, the value given for Attributable Risk is bogus.

Example from Problem 1 Assignment 2:

```r
pop.matrix <- data.matrix(population)
pop.table <- as.table(pop.matrix)
pop.table
```

```
##   disease no.disease
## A     140         60
## B     180        620
```

```r
epi.2by2(pop.table, method = "cohort.count")
```

```
##              Outcome +    Outcome -      Total        Inc risk *
## Exposed +          140           60        200              70.0
## Exposed -          180          620        800              22.5
## Total              320          680       1000              32.0
##                  Odds
## Exposed +       2.333
## Exposed -       0.290
## Total           0.471
##
## Point estimates and 95 % CIs:
## -------------------------------------------------------------------
## Inc risk ratio                               3.11 (2.66, 3.64)
## Odds ratio                                   8.04 (5.69, 11.35)
## Attrib risk *                                47.50 (40.52, 54.48)
## Attrib risk in population *                  9.50 (5.41, 13.59)
## Attrib fraction in exposed (%)               67.86 (62.38, 72.54)
## Attrib fraction in population (%)            29.69 (23.98, 34.97)
## -------------------------------------------------------------------
##  X2 test statistic: 165.901 p-value: < 0.001
##  Wald confidence limits
##  * Outcomes per 100 population units
```

2.2 Case-Control Study:

Generic: `epi.2by2(data, method = "case.control", conf.level = 0.95, units = 100, homogeneity = "breslow.day", outcome = "as.columns")`

- Note that if you're dealing with a case-control study's data, the value given for Attributable Risk is an approximation that is only valid if your disease is rare (p. 50)

Another Example from Problem 1 Assignment 2:

```
pop.matrix <- data.matrix(population)
pop.table <- as.table(pop.matrix)
pop.table
```

```
##   disease no.disease
## A     140         60
## B     180        620
```

```
epi.2by2(pop.table, method = "case.control") # Notice the difference from above?
```

```
##              Outcome +    Outcome -      Total        Prevalence *
## Exposed +          140           60        200                70.0
## Exposed -          180          620        800                22.5
## Total              320          680       1000                32.0
##                  Odds
## Exposed +       2.333
## Exposed -       0.290
## Total           0.471
##
## Point estimates and 95 % CIs:
## -------------------------------------------------------------------
## Odds ratio (W)                               8.04 (5.69, 11.35)
## Attrib prevalence *                          47.50 (40.52, 54.48)
## Attrib prevalence in population *            9.50 (5.41, 13.59)
## Attrib fraction (est) in exposed  (%)        87.52 (82.20, 91.34)
## Attrib fraction (est) in population (%)      38.31 (31.86, 44.14)
## -------------------------------------------------------------------
##  X2 test statistic: 165.901 p-value: < 0.001
##  Wald confidence limits
##  * Outcomes per 100 population units
```

3. Chi-Squared Test from Function Calls

The output of both studies shows the chi-squared value near the bottom: `165.901 p-value: < 0.001`, which is what we expect. Just to see a more reasonable example, let's pretend that we have a χ^2 test statistic of 7. and we wanted to calculate its corresponding p-valie. Another option (rather than running `epi.2by2`) for finding the p-value associated with a value in a χ^2 distribution is to type `pchisq()`. More specifically:

```
pchisq(7, df=1, lower.tail=FALSE)
```

```
## [1] 0.008150972
```

The `pchisq` function expects the test statistic as its first entry, then a comma, then the degrees of freedom. If `lower.tail=FALSE`, then it gives us the probability to the **left** of the test statistic. If we need the **right** side, we can assign the argument to `lower.tail=TRUE`.

```
pchisq(7, df=1, lower.tail=TRUE)
```

```
## [1] 0.991849
```

To lear more about the built-in statistical distributions and how to use them to get p-values, you can go to the documentation linked here: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/Distributions.html