Statistics 305/605: Introduction to Biostatistical Methods for Health Sciences

Demo for Chapter 15, part 4: Inference for Odds Ratios

Jinko Graham, Brad McNeney

Testing whether $OR = 1$

- The chi-square test assesses the null hypothesis that $OR = 1$ (no association between exposure and disease) against the alternative hypothesis that $OR \neq 1$ (an association).

```
mydf <- data.frame(case=c(1350,7),control=c(1296,61)) # Doll and Hill's data
rownames(mydf) <- c("smoker","non-smoker")
mydf
```

```
##            case control
## smoker     1350    1296
## non-smoker    7      61
```

```
chisq.test(mydf)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  mydf
## X-squared = 42.37, df = 1, p-value = 7.552e-11
```

Using R to get point and interval estimates of the OR.

- For a 2×2 table, the calculations are easy enough to do with a hand calculator or spreadsheet.
- Below we present some R code that illustrates data frame manipulation in R.
  - This will give us code that will generalize to the case of multiple exposures, as in the Doll and Hill dataset where smoking status has four levels.

Data Frame for Confidence Intervals

```r
library(dplyr) # for the mutate() function
mydf <- data.frame(a=1350,b=1296,c=7,d=61)
zstar <- qnorm((1-.95)/2,lower.tail=FALSE)
mydf <- mutate(mydf,
               OR=a*d/(b*c),
               logOR=log(OR),
               SE=sqrt(1/a+1/b+1/c+1/d),
               logci.lower=logOR-zstar*SE,
               logci.upper=logOR+zstar*SE,
               ci.lower=exp(logci.lower),
               ci.upper=exp(logci.upper))
round(mydf,2) # Round all numbers to 2 decimals when printing
```

```
##      a    b c  d   OR logOR  SE logci.lower logci.upper ci.lower ci.upper
## 1 1350 1296 7 61 9.08  2.21 0.4        1.42        2.99     4.14    19.92
```

- `mutate()` is used to create new variables from existing ones and add them to our data frame.
- In this example, variables such as `OR` and `logOR` are created and added to `mydf`.
  - Notice that the calculation of `logOR` can use the newly-created variable `OR`.

More Than Two Exposure Levels

- Doll and Hill's data with smokers classified by the average number of cigarettes per day:

		case	control
Number of	25+	340	182
cigarettes	15-24	445	408
per day	1-14	565	706
	0	7	61

- Can use the last row with 0 cigs per day (unexposed) as a baseline group, and calculate our ORs for each level of exposure.
- Here is where the R code we wrote can pay off. We essentially repeat the code, but with different definitions of `a`, `b`, `c` and `d`.

```r
mydf <- data.frame(a=c(340,445,566),
                   b=c(182,408,706),
                   c=c(7,7,7),
                   d=c(61,61,61))
mydf <- mutate(mydf,
               OR=a*d/(b*c),
               logOR=log(OR),
               SE=sqrt(1/a+1/b+1/c+1/d),
               logci.lower=logOR-zstar*SE,
               logci.upper=logOR+zstar*SE,
               ci.lower=exp(logci.lower),
               ci.upper=exp(logci.upper))
round(mydf,2) #
```

```
##     a   b c  d    OR logOR   SE logci.lower logci.upper ci.lower ci.upper
## 1 340 182 7 61 16.28  2.79 0.41        1.99        3.59     7.30    36.32
## 2 445 408 7 61  9.50  2.25 0.40        1.46        3.05     4.30    21.02
## 3 566 706 7 61  6.99  1.94 0.40        1.15        2.73     3.17    15.39
```

Software Notes

- Simple arithmetic in R such as `a*d` generalizes to the case where `a` and `d` are comprised of more than one number.
- In the example on the previous page, where `a=c(340,445,566)` and `d=c(61,61,61)`, the result is `c(340*61,445*61,566*61)`; i.e., the multiplication is done element-wise.
- Many other functions, such as `log()`, `exp()` and `sqrt()` also act element-wise.
- This allows use to calculate multiple ORs, SEs and CIs simultaneously.