# 022 - Contingency Tables and Difference of Proportions

**EPIB607 - Inferential Statistics**[a]

[a]Fall 2020, McGill University

This version was compiled on November 24, 2020

## 1. Inference for Binomial Proportions

**1.1. Hypothesis Testing with `prop.test()`.** The function `prop.test()` is used to conduct a hypothesis test for a single proportion or for the difference of two proportions, under the assumption that the sampling distribution for each sample proportion is approximately normal.

The **`prop.test()`** function has the following generic structure:

```
prop.test(x, n, alternative = "two.sided", p = 0.5, conf.level = 0.95, correct = TRUE)
```

where `x` is the count of successes, `n` is the number of trials, `alternative` specifies the form of the alternative hypothesis, `p` is $p_0$, and `conf.level` refers to the confidence level. The argument for `alternative` can be either `"two.sided"` ($H_A : p \neq p_0$), `"less"` ($H_A : p < p_0$), or `"greater"` ($H_A : p > p_0$. By default, confidence level is set to 95% and a two-sided alternative is tested. To conduct the two-sample test, enter `x` and `n` as vectors; i.e., enter the number of successes in each group and the number of trials in each group.

By default, Yates' continuity correction is applied where possible (`correct = TRUE`). The purpose of the continuity correction is to adjust for the error introduced by using a continuous distribution (the $\chi^2$ distribution) to model discrete probabilities; the correction is meant to protect from underestimating $p$-values when sample sizes are small. It has been shown, however, that the correction can be overly strict and contribute to Type II error. In modern practice, exact tests like the binomial test and Fisher's test are used when sample sizes are small.

The following example shows a hypothesis test for testing the one-sided hypothesis that the proportion of patients who respond to combined therapy with nivolumab and ipilimumab is greater than 0.30, using data that 21 out of 52 patients experienced a response.

```
prop.test(x = 21, n = 52, alternative = "greater", p = 0.30, conf.level = 0.95)
```

```
#
#   1-sample proportions test with continuity correction
#
#  data:  21 out of 52, null probability 0.3
#  X-squared = 2.1987, df = 1, p-value = 0.06906
#  alternative hypothesis: true p is greater than 0.3
#  95 percent confidence interval:
#   0.2906582 1.0000000
```

```
#  sample estimates:
#          p
#  0.4038462
```

The output of `prop.test()` is organized as a list object, and so specific pieces can be extracted using the dollar sign ($) and the name of the desired component. The following examples show the *p*-value and the confidence interval being selectively output from the example shown above.

```
prop.test(x = 21, n = 52, alternative = "greater", p = 0.30, conf.level = 0.95)$p.val
```

```
#  [1] 0.06906279
```

```
prop.test(x = 21, n = 52, alternative = "greater", p = 0.30, conf.level = 0.95)$conf.int
```

```
#  [1] 0.2906582 1.0000000
#  attr(,"conf.level")
#  [1] 0.95
```

The following example shows a hypothesis test for testing the one-sided hypothesis that the proportion of American adults who have sleep trouble is less than 0.40, using data from `nhanes.samp.adult`.

```
#load the data
library(oibiostat)
data("nhanes.samp.adult")

prop.test(sum(nhanes.samp.adult$SleepTrouble == "Yes"),
          length(nhanes.samp.adult$SleepTrouble), alternative = "less", p = 0.40)
```

```
#
#   1-sample proportions test with continuity correction
#
#  data:  sum(nhanes.samp.adult$SleepTrouble == "Yes") out of length(nhanes.samp.adult$Sle
#  X-squared = 9.4522, df = 1, p-value = 0.001055
#  alternative hypothesis: true p is less than 0.4
#  95 percent confidence interval:
#   0.0000000 0.3373014
#  sample estimates:
#          p
#  0.2666667
```

The following example shows a hypothesis test for the difference in population proportions of breast cancer deaths between women who received annual mammograms and women who received standard physical exams. Of the 44,925 women in the mammogram group, 500 died of breast cancer; of the 44,910 women in the control group, 505 died of breast cancer.

```
prop.test(x = c(500, 505), n = c(44925, 44910), alternative = "two.sided",
          conf.level = 0.95)
```

```
#
#   2-sample test for equality of proportions with continuity correction
#
#  data:  c(500, 505) out of c(44925, 44910)
#  X-squared = 0.01748, df = 1, p-value = 0.8948
#  alternative hypothesis: two.sided
#  95 percent confidence interval:
#   -0.001512853  0.001282751
#  sample estimates:
#      prop 1     prop 2
#  0.01112966 0.01124471
```

The following example shows a hypothesis test for the difference in population proportions of sleep trouble between American men and women, using data from `nhanes.samp.adult`.

```
x1 = sum(nhanes.samp.adult$Gender == "female" & nhanes.samp.adult$SleepTrouble == "Yes")
x2 = sum(nhanes.samp.adult$Gender == "male" & nhanes.samp.adult$SleepTrouble == "Yes")
n1 = length(nhanes.samp.adult$Gender == "female")
n2 = length(nhanes.samp.adult$Gender == "male")

prop.test(x = c(x1, x2), n = c(n1, n2), alternative = "two.sided", conf.level = 0.90)
```

```
#
#   2-sample test for equality of proportions with continuity correction
#
#  data:  c(x1, x2) out of c(n1, n2)
#  X-squared = 0.80128, df = 1, p-value = 0.3707
#  alternative hypothesis: two.sided
#  90 percent confidence interval:
#   -0.03087407  0.11976296
#  sample estimates:
#     prop 1    prop 2
#  0.1555556 0.1111111
```

**1.2. Hypothesis Testing with `binom.test()`.** The function `binom.test()` is used to conduct a hypothesis test for a single proportion based on exact binomial probabilities.

The **`binom.test()`** function has the following generic structure:

```
binom.test(x, n, alternative = "two.sided", p = 0.5, conf.level = 0.95)
```

where `x` is the count of successes, `n` is the number of trials, `alternative` specifies the form of the alternative hypothesis, `p` is $p_0$, and `conf.level` refers to the confidence level. The argument

for `alternative` can be either `"two.sided"` ($H_A : p \neq p_0$), `"less"` ($H_A : p < p_0$), or `"greater"` ($H_A : p > p_0$. By default, confidence level is set to 95% and a two-sided alternative is tested.

The following example shows a hypothesis test for testing the one-sided hypothesis that the proportion of glioblastoma patients who respond to Avastin is different from 0.05, using data that 24 out of 85 patients experienced a response.

```
binom.test(x = 24, n = 85, alternative = "two.sided", p = 0.05, conf.level = 0.95)
```

```
#
#   Exact binomial test
#
#  data:  24 and 85
#  number of successes = 24, number of trials = 85, p-value = 2.674e-12
#  alternative hypothesis: true probability of success is not equal to 0.05
#  95 percent confidence interval:
#   0.1900101 0.3904038
#  sample estimates:
#  probability of success
#              0.2823529
```

Note how an attempt to use `prop.test()` in this setting produces a warning that the $\chi^2$ (i.e., Normal) approximation may be incorrect.

```
prop.test(x = 24, n = 85, alternative = "two.sided", p = 0.05, conf.level = 0.95)
```

```
#  Warning in prop.test(x = 24, n = 85, alternative = "two.sided", p = 0.05, : Chi-
#  squared approximation may be incorrect
```

```
#
#   1-sample proportions test with continuity correction
#
#  data:  24 out of 85, null probability 0.05
#  X-squared = 91.78, df = 1, p-value < 2.2e-16
#  alternative hypothesis: true p is not equal to 0.05
#  95 percent confidence interval:
#   0.1926329 0.3920210
#  sample estimates:
#         p
#  0.2823529
```

## 2. Inference for Two-Way Tables

**2.1. $\chi^2$ Distribution Functions.** The function `pchisq()` used to calculate $P(X \leq k)$ or $P(X > k)$ has the generic structure

```
pchisq(q, df, lower.tail = TRUE)
```

where `q` is $k$ and `df` is the degrees of freedom. By default, R calculates $P(X \leq k)$. In order to compute $P(X > k)$, specify `lower.tail = FALSE`.

The following example shows how to calculate $P(X \leq 29.5)$ and $P(X > 29.5)$ for $X \sim \chi^2_{df=20}$.

```
#probability X is less than (or equal to) 1.20
pchisq(29.5, df = 20)
```

```
#  [1] 0.9216293
```

```
#probability X is greater than 1.20
pchisq(29.5, df = 20, lower.tail = FALSE)
```

```
#  [1] 0.07837072
```

The function `qchisq()` used to identify the observation that corresponds to a particular probability $p$ has the generic structure

```
qchisq(p, df, lower.tail = TRUE)
```

where `p` is $p$ and `df` is the degrees of freedom. By default, R identifies the observation that corresponds to area $p$ in the lower tail (i.e., to the left). To identify the observation with area $p$ in the upper tail, specify `lower.tail = FALSE`.

The following example shows how to calculate the value of the observation where there is 0.922 area to the left (and 0.078 area to the right) on a $\chi^2$ distribution with 20 degrees of freedom.

```
#identify X value
qchisq(0.922, df = 20)
```

```
#  [1] 29.52077
```

```
#probability X is greater than 1.20
qchisq(0.078, df = 20, lower.tail = FALSE)
```

```
#  [1] 29.52077
```

**2.2. Entering Data Tables.** The use of the function `matrix()` to construct matrices was previously introduced in the Chapter 1 Lab Notes. For clarity when displaying contingency tables, the matrix dimensions can be labeled using **`dimnames()`**. The first entry in `dimnames()` labels the rows and the second entry labels the columns.

The following example shows a matrix from the HIV study comparing nevirapine (NVP) and lopinarvir (LPV) with labeled dimensions.

```
#enter the data
hiv.table = matrix(c(60, 27, 87, 113),
                   nrow = 2, ncol = 2, byrow = T)

#add labels
dimnames(hiv.table) = list("Outcome" = c("Virologic Failure", "Stable Disease"),
                           "Drug" = c("NVP", "LPV"))

hiv.table
```

```
#                     Drug
#  Outcome             NVP LPV
#    Virologic Failure  60  27
#    Stable Disease     87 113
```

**2.3. Hypothesis Testing with `chisq.test()`.** The **`chisq.test()`** has the following generic structure

```
chisq.test(x, y, correct = TRUE)
```

where `x` is either a matrix or a vector, `y` is a vector, and Yates' continuity correction is applied by default. If `x` is a matrix, the argument `y` is ignored.

The following example shows a test of independence for treatment and outcome in the HIV data.

```
chisq.test(hiv.table)
```

```
#
#   Pearson's Chi-squared test with Yates' continuity correction
#
#  data:  hiv.table
#  X-squared = 14.733, df = 1, p-value = 0.0001238
```

The output of `chisq.test()` is organized as a list object, and so specific pieces can be extracted using the dollar sign ($) and the name of the desired component. The following examples show the residuals and expected values being selectively output from the test conducted above.

```
chisq.test(hiv.table)$residuals
```

```
#                      Drug
#  Outcome                    NVP       LPV
#    Virologic Failure  2.312824 -2.369939
#    Stable Disease    -1.525412  1.563082
```

```
chisq.test(hiv.table)$expected
```

```
#                      Drug
#  Outcome                    NVP      LPV
#    Virologic Failure  44.56098 42.43902
#    Stable Disease    102.43902 97.56098
```

The following example shows a test of independence for statin use and educational level from `prevend.samp`, using both entry options.

```
#load the data
data("prevend.samp")

#use x, y format
chisq.test(prevend.samp$Statin, prevend.samp$Education)
```

```
#
#   Pearson's Chi-squared test
#
#  data:  prevend.samp$Statin and prevend.samp$Education
#  X-squared = 19.054, df = 3, p-value = 0.0002665
```

```
#enter x as a table
statin.edu.table = table(prevend.samp$Statin, prevend.samp$Education)
chisq.test(statin.edu.table)
```

```
#
#   Pearson's Chi-squared test
#
#  data:  statin.edu.table
#  X-squared = 19.054, df = 3, p-value = 0.0002665
```

**2.4. Hypothesis Testing with `fisher.test( )`.** The **`fisher.test()`** has the following generic structure

```
fisher.test(x, y, correct = TRUE)
```

where x is either a matrix or a vector and y is a vector. If x is a matrix, the argument y is ignored.

The following example shows a test of independence for treatment and outcome in the *C. difficile* fecal infusion study.

```
#enter the data
infusion.table = matrix(c(13, 3, 4, 9), nrow = 2, ncol = 2, byrow = T)
dimnames(infusion.table) = list("Outcome" = c("Cured", "Uncured"),
                                "Drug" = c("Fecal Infusion", "Vancomycin"))

fisher.test(infusion.table)
```

```
#
#   Fisher's Exact Test for Count Data
#
#  data:  infusion.table
#  p-value = 0.00953
#  alternative hypothesis: true odds ratio is not equal to 1
#  95 percent confidence interval:
#    1.373866 78.811505
#  sample estimates:
#  odds ratio
#    8.848725
```

**2.5. Relative Risk and Odds Ratio with `epitools`.** The `epitools` package contains various useful calculators for epidemiology, including functions to calculate relative risk and odds ratios in two-way tables. First, install and load the package:

```
install.packages("epitools")
library(epitools)
```

The package requires that the rows of the table contain information on exposure (i.e., treatment) while the columns of the table contain information on outcome (i.e., disease), where the first row specifies the baseline treatment group and the second column specifies presence of the disease outcome.

The following example shows the HIV data re-entered to be in the preferred format, where the rows of the table specify the type of drug treatment and the columns specify the outcome. Note that the second column specifies virologic failure.

```
#enter the data
hiv.table = matrix(c(87, 113, 60, 27),
                   nrow = 2, ncol = 2, byrow = F)

#add labels
dimnames(hiv.table) = list("Drug" = c("NVP", "LPV"),
                           "Outcome" = c("Stable Disease", "Virologic Failure"))
```

```
hiv.table
```

```
#       Outcome
#  Drug  Stable Disease Virologic Failure
#    NVP              87                60
#    LPV             113                27
```

The function **riskratio()** calculates the relative risk and has the following generic structure

```
riskratio(x, y = NULL, rev = "neither")$measure
```

where x is either a matrix or a vector and y is a vector; if x is a matrix, then y is ignored. The argument rev can be either "neither", "rows", "columns", or "both", and will either leave the data as-is, reverse the ordering of the rows, reverse the ordering of the columns, or reverse the ordering of both. To specifically output the estimated relative risk, use $measure.

In the following example, the relative risk of virologic failure is calculated, first with nevirapine as the baseline then with lopinarvir as the baseline. The estimated RR for the baseline group appears as 1. The RR of virologic failure comparing NVP to LPV is 2.12; the risk of virologic failure for individuals treated with nevirapine is over twice that of the risk for those treated with lopinarvir.

The RR can also be calculated in terms of the risk of 'success' (ie., stable disease); for example, the RR of stable disease comparing LPV to NVP is 1.37; the risk of stable disease for individuals treated with lopinarvir is 1.37 times that of the risk for those treated with nevirapine.[1]

```
#calculate RR of failure with respect to NVP
riskratio(hiv.table)$measure
```

```
#       risk ratio with 95% C.I.
#  Drug  estimate     lower    upper
#    NVP   1.0000        NA       NA
#    LPV   0.4725 0.3196517 0.698436
```

```
#calculate RR of failure with respect to LPV
riskratio(hiv.table, rev = "rows")$measure
```

```
#       risk ratio with 95% C.I.
#  Drug  estimate   lower    upper
#    LPV 1.000000      NA       NA
#    NVP 2.116402 1.43177 3.128405
```

[1] Note that all combinations are only shown to illustrate use of rev; relative risks are generally interpreted in terms of presence of disease.

```r
#calculate RR of success with respect to NVP
riskratio(hiv.table, rev = "columns")$measure
```

```
#       risk ratio with 95% C.I.
#  Drug  estimate    lower    upper
#    NVP 1.000000       NA       NA
#    LPV 1.363793 1.165901 1.595274
```

```r
#calculate RR of success with respect to LPV
riskratio(hiv.table, rev = "both")$measure
```

```
#       risk ratio with 95% C.I.
#  Drug   estimate     lower     upper
#    LPV 1.0000000        NA        NA
#    NVP 0.7332491 0.6268517 0.8577056
```

The following example shows the use of `riskratio` with vectors; specifically, vectors from a larger dataframe. The relative risk of cardiovascular disease comparing smokers to non-smokers is 1.09; smokers have a 9% higher risk of cardiovascular disease compared to non-smokers, as estimated from `prevend.samp`.

```r
#load the data
data("prevend.samp")

#convert to factors
prevend.samp$Smoking = factor(prevend.samp$Smoking, levels = c(0, 1),
                              labels = c("SmokeNo", "SmokeYes"))
prevend.samp$CVD = factor(prevend.samp$CVD, levels = c(0, 1),
                          labels = c("CVDNo", "CVDYes"))

#calculate the relative risk of CVD with respect to not smoking
riskratio(prevend.samp$Smoking, prevend.samp$CVD)$measure
```

```
#           risk ratio with 95% C.I.
#  Predictor  estimate     lower  upper
#    SmokeNo  1.000000        NA     NA
#    SmokeYes 1.086207 0.5476699 2.1543
```

```r
#view the data table
riskratio(prevend.samp$Smoking, prevend.samp$CVD)$data
```

```
#            Outcome
#  Predictor  CVDNo CVDYes Total
#    SmokeNo    348     30   378
#    SmokeYes   106     10   116
#    Total      454     40   494
```

The function **oddsratio()** calculates the odds ratio and has the following generic structure

```
oddsratio(x, y = NULL, rev = "neither", method = "wald")$measure
```

where x is either a matrix or a vector and y is a vector; if x is a matrix, then y is ignored. The argument rev can be either "neither", "rows", "columns", or "both", and will either leave the data as-is, reverse the ordering of the rows, reverse the ordering of the columns, or reverse the ordering of both. The argument "wald" specifies that the odds ratio should be calculated using unconditional maximum likelihood estimation; this corresponds to the formula used in *OI Biostat*. To specifically output the estimated odds, use $measure.

In the following example, the odds ratio of virologic failure is calculated, first with nevirapine as the baseline then with lopinarvir as the baseline. The OR for the baseline group appears as 1. The OR of virologic failure comparing NVP to LPV is 2.87; the odds of virologic failure for individuals treated with NVP are over twice as large as the odds of failure for those treated with LPV.

```
#calculate OR of failure with respect to NVP
oddsratio(hiv.table, method = "wald")$measure
```

```
#       odds ratio with 95% C.I.
#  Drug   estimate     lower    upper
#    NVP 1.0000000        NA       NA
#    LPV 0.3464602 0.2032484 0.590581
```

```
#calculate OR of failure with respect to LPV
oddsratio(hiv.table, rev = "rows", method = "wald")$measure
```

```
#       odds ratio with 95% C.I.
#  Drug  estimate    lower    upper
#    LPV 1.000000       NA       NA
#    NVP 2.886335 1.693248 4.920088
```

The following example shows the use of oddsratio with vectors; specifically, vectors from a larger dataframe. The odds ratio of cardiovascular disease comparing smokers to non-smokers is 1.10, as estimated from prevend.samp; the odds of cardiovascular disease in smokers are 10% larger than in non-smokers.

```r
#calculate the odds ratio of CVD with respect to not smoking
oddsratio(prevend.samp$Smoking, prevend.samp$CVD, method = "wald")$measure
```

```
#            odds ratio with 95% C.I.
#  Predictor  estimate     lower    upper
#    SmokeNo   1.00000        NA       NA
#    SmokeYes  1.09434 0.5179749 2.312041
```

```r
#view the data table
oddsratio(prevend.samp$Smoking, prevend.samp$CVD)$data
```

```
#            Outcome
#  Predictor  CVDNo CVDYes Total
#    SmokeNo    348     30   378
#    SmokeYes   106     10   116
#    Total      454     40   494
```