# Tabular Statistics

Manpreet S. Katari / Kris Gunsalus

October 21, 2019

## Contents

### Background Reading

- Aho, Chapter 11 – Tabular Analyses (through Section 11.6)
- Rosner, Chapter 10 – Hypothesis Testing: Categorical Data

### Example: Two-way comparison

An international study was done to test the hypothesis that women who postpone childbirth until later in life have a higher risk of breast cancer, which they examined by asking whether there is an association between the incidence of breast cancer and the age at which women first gave birth (Rosner, Example 10.4).

The study found that 683 out of 3,220 women WITH breast cancer first gave birth above the age of 30 (21.2%), whereas 1,498 out of 10,245 women WITHOUT breast cancer first gave birth at an age above 30 (14.6%):

```r
# knitr is a library for dynamic report generation that makes pretty tables
library(knitr)

Case    = c(683,2537)   # total =  3220
Control = c(1498,8747)  # total = 10245
data_matrix = rbind(Case, Control)
colnames(data_matrix) = c("Above30","Below30")

# kable is a table generator from the knitr package
kable(data_matrix)
```

| | Above30 | Below30 |
|---|---:|---:|
| Case | 683 | 2537 |
| Control | 1498 | 8747 |

Is this difference significant? If there is no association, then the probabilities of these two factors should be independent, and the expected proportion of individuals with both characteristics would just equal the product of the independent proportions. For example:

$$\hat{p}_{(Cancer\ \cap Above30)} = \hat{p}_{Cancer} * \hat{p}_{Above30}$$

This is our **null hypothesis**. Below we will consider three different approaches to determine the significance of the differences.

## Binomial proportions

Specifically, we want to know if the **proportion** of women who gave birth at an older age is significantly higher in the group of women WITH breast cancer (call this $\hat{p}_1$) than it is in the group of women WITHOUT breast cancer (call this $\hat{p}_2$). Under the null hypothesis $H_0$, the two proportions should be equal, so their difference should be zero:

$$H_0 : \hat{p}_{(Cancer\ \cap Above30)} - \hat{p}_{(Normal \cap Above30)} = \hat{p}_1 - \hat{p}_2 = 0$$

### Normal-theory method

When the samples are large enough, we can use the **normal approximation** for the binomial distribution to answer this question, even though we don't have information about the variation in the data. The population mean and variance are then given by: $\mu = p$ and $\sigma^2 = pq/n$, where $q = 1 - p$. We don't know the overall population proportion, but we can estimate it as the the weighted average of the sample proportions:

$$\hat{p} = \frac{Over30}{Total} = \frac{x_1 + x_2}{n_1 + n_2}$$

where $x_1$ and $x_2$ are the number of women above 30 in each sample (Case = cancer, Control = normal) and $n_1$ and $n_2$ are the total number of women in each sample.

Assuming the **proportions** in the two samples follow a normal distribution, then the **difference in proportions** should also follow a normal distribution. This is similar to looking at the difference of the means between two sample groups, which we learned about previously. We can calculate a $z$-score for the difference in proportions in the usual way:

$$z = \frac{(\hat{p}_1 - \hat{p}_2)}{SE(\hat{p}_1 - \hat{p}_2)} \sim \mathcal{N}(0, 1)$$

As before, the $z$-score follows a normal distribution with $mean = 0$ and $SD = 1$. The critical test statistic, $z*$, will be determined as usual by the choice of $\alpha$. If $z* > z_{1-\alpha/2}$, then we will consider the difference to be significant.

Fortunately, we can use the `prop.test` function to calculate if the difference in the proportions is significant using this model. Since we expect the Case group to have a higher proportion of women who first gave birth above the age of 30, our alternative hypothesis is $H_A : \hat{p_1} > \hat{p_2}$, so we will choose a one-tailed test:

```
## two-sample test for equal proportions
# prop.test(x,n,p=NULL,correct=TRUE)
#   look at the help section on how to use this command:
#     x = vector or 1D table with 2 entries giving # of "successes" in each sample,
#         or (SEE BELOW) a 2x2 table or matrix with # of sucesses and failures
#     n = vector giving total number of samples
#     correct = correction for continuous approximation of the binomial (default)

## Data --
```

```
# Case group (with cancer):         683/3220 are in "Above30" group
# Control group (without cancer): 1498/10245 are in "Above30" group

## Syntax method 1: using raw data
# "successes" are "Above30", so            x = c( 683, 1498)
# totals in each group (Case, Control) are n = c(3220,10245)

prop.test(c(683,1498),c(3220,10245), alternative="greater")
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(683, 1498) out of c(3220, 10245)
## X-squared = 77.885, df = 1, p-value < 2.2e-16
## alternative hypothesis: greater
## 95 percent confidence interval:
##  0.05252238 1.00000000
## sample estimates:
##    prop 1    prop 2
## 0.2121118 0.1462177
```

```
## Syntax method 2: using a matrix
# Notes:
#  1) If a matrix is given instead of vector x, then n is ignored.
#  2) A two-dimensional table (or matrix) is expected with 2 COLUMNS giving
#     the number of "successes" (Above30) and "failures" (Under30), respectively.
#     Thus, the ROWS are the samples being compared (Case, Control).

data_matrix
```

```
##         Above30 Below30
## Case        683    2537
## Control    1498    8747
```

```
prop.test(data_matrix, alternative="greater")
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  data_matrix
## X-squared = 77.885, df = 1, p-value < 2.2e-16
## alternative hypothesis: greater
## 95 percent confidence interval:
##  0.05252238 1.00000000
## sample estimates:
##    prop 1    prop 2
## 0.2121118 0.1462177
```

```
# The function returns a data structure with a lot of different pieces of information.
# You can recover the p-value directly from the test result:
prop.test(data_matrix, alternative="greater")$p.value
```

```
## [1] 5.460478e-19
```

```
# NOTE: if table orientation is flipped, we change our frame of reference,
# so the reported proportions and confidence interval are different ...
prop.test(t(data_matrix), alternative="greater")
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  t(data_matrix)
## X-squared = 77.885, df = 1, p-value < 2.2e-16
## alternative hypothesis: greater
## 95 percent confidence interval:
##  0.07048667 1.00000000
## sample estimates:
##    prop 1    prop 2
## 0.3131591 0.2248316
```

```
# ... but the p-value is the same!
# This is because the test statistic itself is agnostic to table orientation.
prop.test(t(data_matrix), alternative="greater")$p.value
```

```
## [1] 5.460478e-19
```

```
# Note that `prop.test` reports the Chi-squared statistic ("X-squared") with df = 1.
# Though the conceptual framework is different, the z-score and X-squared statistics
# are mathematically equivalent for two-way comparisons.
# (In fact, for a two-sided test, `prop.test` calls `chisq.test` underneath.)
# You can examine the code for this (or any) function using `getAnywhere(prop.test)`.
```

The extremely low $p$-value suggests that there is indeed an association between age of first child birth and breast cancer.

The equations for the distribution of $\hat{p}_1 - \hat{p}_2$ and the $z$-score are explained in Rosner, Section 10.2 and summarized in Eq. 10.3. The default for the `prop.test` is to use Yates' correction for continuity, since the (continuous) normal approximation underestimates the true $p$-values based on the binomial distribution.

As a rule of thumb, **the normal approximation is considered valid whenever** $n_1\hat{p}\hat{q} \geq 5$ **and** $n_2\hat{p}\hat{q} \geq 5$. When this does ***not*** hold, **Fisher's Exact Test** is recommended (see below).

## Contigency tables

Tabular data are typically organized into a 2x2 **contigency table** to represent data for two variables with two possible outcomes each. Larger tables can be used for multinomial vs. binary outcomes.

**Note:** ***By convention***, the two samples to be compared are presented in the ROWs (e.g. Case, Control), and the different groups within each sample are presented in the COLUMNs (e.g. Above30, Below30). However, ***table orientation is arbitrary for the purpose of the statistical tests***, which will give the same result either way.

Let's make a contingency table with two rows and two columns to represent our cancer data.

- **Row 1:** women WITH cancer ("Case")
- **Row 2:** women WITHOUT cancer ("Control")
- **Col 1:** women first giving birth above 30 ("Above30")
- **Col 2:** women first giving birth below 30 ("Below30")

A common practice is to designate rows with the subscript $i$ and columns with the subscript $j$, so the counts in each cell will have an index $x_{ij}$.

Since this can get confusing, we can alternatively refer to the quadrants as $a$ (top left), $b$ (top right), $c$ (bottom left), and $d$ (bottom right).

- $a = x_{11} = \#$ of women with cancer & first birth ABOVE 30

- $b = x_{12}$ = # of women with cancer & first birth BELOW 30
- $c = x_{21}$ = # of women without cancer & first birth ABOVE 30
- $d = x_{22}$ = # of women without cancer & first birth BELOW 30

Individual row and column totals are called ***marginal*** totals (since the sum of the cells is written in the margins). The row margins will be $m_1 = (a+b)$ and $m_2 = (c+d)$, and the column margins $n_1 = (a+c)$ and $n_2 = (b+d)$:

- **Row 1 margin:** $m_1 = x_{1+} = x_{11} + x_{12}$ = all women with cancer
- **Row 2 margin:** $m_2 = x_{2+} = x_{21} + x_{22}$ = all women without cancer
- **Col 1 margin:** $n_1 = x_{+1} = x_{11} + x_{21}$ = # of women who first gave birth over 30
- **Col 2 margin:** $n_2 = x_{+2} = x_{12} + x_{22}$ = # of women who first gave birth under 30

Finally, the ***grand total*** is:

- **Grand total:** $N = (a+b+c+d) = x_{11} + x_{12} + x_{21} + x_{22} = x_{1+} + x_{2+} = x_{+1} + x_{+2}$

Now let's generate the contingency table:

```r
# from above
Case    = c(683,2537)    # total =  3220
Control= c(1498,8747)  # total = 10245

# contingency table with row and col margins, plus grand total
CaseT     = c(Case,    sum(Case))     # = c( 683, 2537,  3220)
ControlT  = c(Control, sum(Control))  # = c(1498, 8747, 10245)
Totals    = CaseT + ControlT

data_table = rbind(CaseT, ControlT, Totals)
colnames(data_table) = c("Above30", "Below30", "Total")
rownames(data_table) = c("Case", "Control", "Total")
kable(data_table)
```

| | Above30 | Below30 | Total |
|---|---:|---:|---:|
| Case | 683 | 2537 | 3220 |
| Control | 1498 | 8747 | 10245 |
| Total | 2181 | 11284 | 13465 |

## Chi-squared Test

A second method we can use to find the probability of an association is the **Chi-squared Test**. The test statistic, $X^2$, compares individual proportions in each group to the **expected proportion** based on the **population mean estimate**. The resulting $p$-value will be the same as for a ***two-tailed*** `prop.test`.

Returning to our example, under the null hypothesis $H_0$, the proportion of women with first birth over 30 in each group (Case and Control) should be the same, and the joint probability should follow the product rule under independence. We can restate our null hypothesis above more generally as:

$$H_0 : \hat{p}_{ij} = \hat{p}_{i+} * \hat{p}_{+j}$$

We first determine the *Expected* values for each cell, which will be the total table count multipled by the the joint probability under independence. This can also be calculated as the product of the *row margins* and the *column margins*, divided by the grand total:

$$H_0 : E_{ij} = N * \hat{p}_{i+} * \hat{p}_{+j} = \frac{x_{i+} x_{+j}}{N} = \frac{m_i n_j}{N}$$

So for example,

$$E(Case \cap > 30) = \frac{(TotalCase) * (Total > 30)}{Total} = \frac{m_1 * n_1}{N} = \frac{3220 * 2181}{13465}$$

```
data_table
```

```
##          Above30 Below30 Total
## Case         683    2537  3220
## Control     1498    8747 10245
## Total       2181   11284 13465
```

```
N = data_table[3,3]
Case_tot    = data_table[1,3]
Ctl_tot     = data_table[2,3]
Over30_tot  = data_table[3,1]
Under30_tot = data_table[3,2]

Expected = c( Case_tot * Over30_tot / N,
              Ctl_tot  * Over30_tot / N,
              Case_tot * Under30_tot / N,
              Ctl_tot  * Under30_tot / N)

ExpectedValues = matrix(Expected,nrow=2,ncol=2)
ExpectedValues
```

```
##           [,1]     [,2]
## [1,]  521.5611 2698.439
## [2,] 1659.4389 8585.561
```

The test statistic, $X^2$, is:

$$X^2 = \sum \frac{(Obs - Exp)^2}{Exp}$$

So for each cell, we take the square of the difference between the Observed and Expected and divide by Expected, then sum these to get $X^2$. Under $H_0$, $X^2$ will have an asymptotic $\chi^2$ distribution with $(r-1)(c-1) = 1$ degree of freedom.

We can use the `pchisq` function to get a $p$-value for the cumulative Chi-squared distribution with one degree of freedom, $P(\chi^2_{1,1-\alpha} \geq X^2)$:

```
data_matrix
```

```
##          Above30 Below30
## Case         683    2537
## Control     1498    8747
```

```
chisq = sum( (data_matrix - ExpectedValues)^2 / ExpectedValues )
chisq
```

```
## [1] 78.36984
```

```
pchisq(chisq, df = 1, lower.tail = F)
```

```
## [1] 8.544684e-19
```

The R command for the Chi-squared test is `chisq.test`:

```
chisq.test(data_matrix)     # default is correct = TRUE
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  data_matrix
## X-squared = 77.885, df = 1, p-value < 2.2e-16
```

```
chisq.test(data_matrix)$p.value
```

```
## [1] 1.092096e-18
```

```
# result is the same regardless of table orientation
chisq.test(t(data_matrix))$p.value
```

```
## [1] 1.092096e-18
```

Notice that the manual calculation using `pchisq` does not give exactly the same *p*-value as `chisq.test`. This is because the approximation we made by hand does not correct for the fact that $X^2$ is a continuous function, and thus underestimates the true *p*-values based on the binomial distribution. In contrast, `chisq.test` uses Yates' correction by default.

We can apply Yates' correction manually by subtracting 1/2 from each of the squared differences above, which will give us the same result at `chisq.test`:

```
# Yates' correction for continuity
chisq = sum( (abs(data_matrix - ExpectedValues) - 0.5)^2 / ExpectedValues )
chisq
```

```
## [1] 77.88515
```

```
pchisq(chisq, df = 1, lower.tail = F)
```

```
## [1] 1.092096e-18
```

We can confirm that the *p*-value returned by `chisq.test` is the same as for a two-tailed `prop.test`:

```
chisq.test(data_matrix)$p.value
```

```
## [1] 1.092096e-18
```

```
# two-sided prop.test
prop.test(c(683,1498),c(3220,10245))$p.value
```

```
## [1] 1.092096e-18
```

**Note:** Similar to the normal approximation above, **the $\chi^2$ test is not recommended for cases where expected value in any cell is less than 5**; instead, **Fisher's Exact Test** is recommended.

## Fisher's Exact Test

For Fisher's exact test, we don't need the data to be normally distributed and we can look at all permutations and combinations to determine the *p*-value.

Fisher's exact test calculates a *p*-value using the non-central hypergeometric distribution, which gives the probability of the observed data **given fixed row margins**.

With fixed marginal counts, the count in one cell, $x_{ij}$, will determine the counts in the other three cells (see Aho, Section 11.6.3 for details).

The probability of the observed data is simply the product of factorials of all the marginals, divided by the product of factorials of each cell and factorial of total number.

$$P(a,b,c,d) = \frac{(a+b)!\,(a+c)!\,(b+d)!\,(c+d)!}{a!b!c!d!N!}$$

where $N = a + b + c + d$. This turns out to be a special case of the hypergeometric distribution.

The $p$-value will be the proportion of all possible ways to get a value **equal to or more extreme** than the observed count in one of the cells (typically $x_{11}$).

We can perform a Fisher's exact test using the `fisher.test` function. We will choose a one-tailed test, since we expect the Case group to have a higher proportion of women who first gave birth over 30 ($H_A : \Theta > 1$).

```
data_matrix
```

```
##         Above30 Below30
## Case        683    2537
## Control    1498    8747
```

```
# full result from Fisher's Exact Test
fisher.test(data_matrix, alternative='greater')
```

```
## 
##  Fisher's Exact Test for Count Data
## 
## data:  data_matrix
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is greater than 1
## 95 percent confidence interval:
##  1.442384      Inf
## sample estimates:
## odds ratio 
##   1.571925
```

```
# p-value from Fisher's Exact Test
ftest_pval = fisher.test(data_matrix, alternative='greater')$p.value
ftest_pval
```

```
## [1] 3.526441e-18
```

```
# table orientation is arbitrary; both orientations give the same result
fisher.test(t(data_matrix), alternative='greater')$p.value
```

```
## [1] 3.526441e-18
```

The $p$-value computed in this way is not exactly the same as we get from the other two methods we used above, but they are all extremely significant.

Notice that we have a reference to the **odds ratio**, which is something we have discussed in the past. The OR here is:

$$\hat{\Theta}_{1,2} = \frac{\hat{p}_{11}/\hat{p}_{12}}{\hat{p}_{21}/\hat{p}_{22}} = \frac{\hat{p}_{11}\hat{p}_{22}}{\hat{p}_{12}\hat{p}_{21}} = \frac{x_{11}x_{22}}{x_{12}x_{21}}$$

Under independence, this ratio should equal 1; if one row has higher or lower counts than expected, then the OR will be greater than or less than 1.

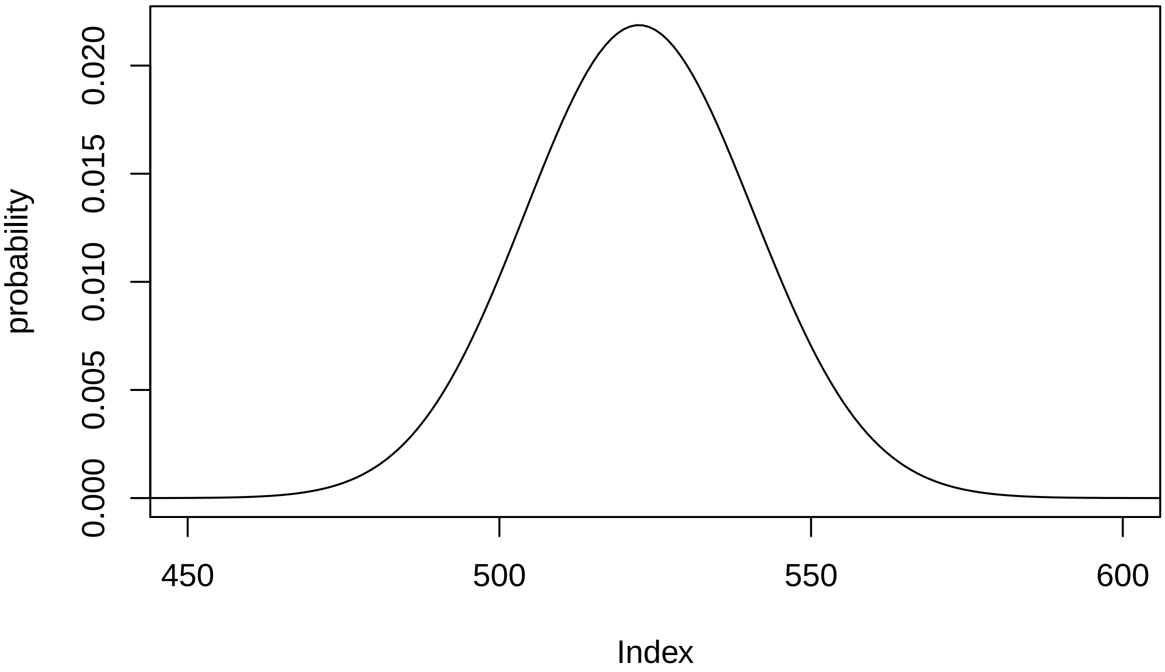
## Hypergeometric Distribution

Fisher's Exact Test uses a non-central hypergeometric distribution, so we should get the same result if we compute a $p$-value empirically using the `hyper` function family.

Let's check this. Keeping the marginals the same, we will compute the probabilities of all possible combinations of counts in each quadrant and then look at the distribution of all the $p$-values we obtain.

To find the $p$-value for our observed data, we will just add up all the probabilities for each simulated table where the count in quadrant $a$ is at least as large as the observed value, 683.

```r
# initialize variables
probability = numeric()
hyper_pval  = 0

Case_Above30 = data_table[1,1]
Case_tot     = data_table[1,3]
Above30_tot  = data_table[3,1]
N            = data_table[3,3]

# Iterate over total number of cancer samples, starting at 0.
# Each time set observed number of (cancer AND Over30 = iterator), and
# compute the rest of the cells assuming fixed margins.
# This will produce the full hypergeometric distribution.
for (i in 0:Case_tot) {
  a = i
  b = Case_tot    - a
  c = Above30_tot - a
  d = N - a - b - c

  tempdata = matrix(c(a,b,c,d), nrow=2, ncol=2)
  # use index = i + 1 since R matrices start with index 1 not 0
  probability[i+1]=dhyper(a, a+b, c+d, a+c)

  if (a >= Case_Above30) {
    hyper_pval = hyper_pval + probability[i+1]
  }
}

plot(probability, type="l", xlim=c(450,600))
```



```
# cumulative probabilities for test counts >= observed for Case_Over30
hyper_pval
```

```
## [1] 3.526441e-18
```

```
ftest_pval # Fisher's p-value
```

```
## [1] 3.526441e-18
```

The $p$-value is the same as that from Fisher's exact test! ;-)

We can also use the `phyper` function directly to get a $p$-value for the observed data from our breast cancer example above. Since this is a discrete distribution, we perform a one-tailed test using $(683 - 1)$:

```
# phyper(q, m, n, k, lower.tail = FALSE)
# phyper(a, a+b, c+d, a+c, lower.tail = FALSE)
#  q = observation (a   = cancer and above 30)
#  m = white balls (a+b = cancer, a.k.a. "success")
#  n = black balls (c+d = normal, a.k.a. "failure")
#  k = total draws (a+c = above 30)
phyper(683 - 1, 3220, 10245, 2181, lower.tail = FALSE)
```

```
## [1] 3.526441e-18
```

Below I have rewritten the same computations for the Fisher's Exact and hypergeometric tests using R's terminology for the `hyper` family of functions:

```
# data (variable names chosen to match dhyper() argument names)
x = 683    # Case_Above30 = a
m = 3220   # Case         = row 1 margin = a+b
n = 10245  # Control      = row 2 margin = c+d
k = 2181   # Above30      = col 1 margin = a+c

# same as 'data_matrix' used above
# (although table orientation doesn't matter for the Fisher exact test)
matrix(c(x,   m-x,      # a, b
         k-x, n-(k-x)), # c, d
       2,2,
       byrow = TRUE) # matrices fill by columns by default
```

```
##      [,1] [,2]
## [1,]  683 2537
## [2,] 1498 8747
```

```
# Fisher test, alternative = 'greater'
fisher.test(matrix(c(x, m-x, k-x, n-(k-x)),2,2), alternative='greater')$p.value
```

```
## [1] 3.526441e-18
```

```
q = x-1  # upper tail for discrete distribution
phyper(q, m, n, k, lower.tail = FALSE)
```

```
## [1] 3.526441e-18
```

## Barnard's Exact Test for equality of proportions

It turns out that experimental study design governs whether the true distribution of the data is hypergeometric or multinomial, and therefore whether Fisher's is necessarily the right choice.

Barnard's test is a non-parametric alternative to Fisher's exact test. Because it does not assume (is not conditioned on) fixed margins, Barnard's exact test is reported to have greater power than Fisher's exact test for 2x2 contingency tables. You can read about it on Wikipedia: Barnard's test

An implementation of Barnard's test is available in R: Blog on Barnard's exact test

## Exercise

22,071 physicians served as subjects to study effect of aspirin on incidence of heart attacks. There are two groups: those who took aspirin regularly for 5 years and others (11,043 of the 22,071) received placebo instead of aspirin. 189 of the group that took placebo suffered a heart attack and 104 of those who took aspirin suffered a heart attack. Does aspirin have a significant association with physicians suffering from a heart attack ?

1. Create a contingency table for these data.

```
# aspirin
# heart attach
N = 22071
a = 189 # took placebo and had heart attack
b = 104 # took aspirin and had heart attach
c =
d =

placebo_total = 11043
placebo_heart = 189
placebo_healthy = placebo_total - placebo_heart

aspirin_total = N - placebo_total
aspirin_heart = 104
aspirin_healthy = aspirin_total - aspirin_heart


aspirindata = rbind(c(placebo_heart,aspirin_heart),
                    c(placebo_healthy, aspirin_healthy)
                    )
aspirindata
```

```
##        [,1]  [,2]
## [1,]   189   104
## [2,] 10854 10924
```

```
heart_total = placebo_heart + aspirin_heart
healthy_total = placebo_healthy + aspirin_healthy
```

2. Calculate the expected values.

```
placebo_heart_expected = (placebo_total/N) * (heart_total/N) * N
placebo_heart_expected
```

```
## [1] 146.5996
```

```
placebo_healthy_expected = (placebo_total/N) * (healthy_total/N) * N
placebo_healthy_expected
```

```
## [1] 10896.4
```

```
aspirin_healthy_expected = (aspirin_total/N) * (healthy_total/N) * N
aspirin_healthy_expected
```

```
## [1] 10881.6
```

```
aspirin_heart_expected = (aspirin_total/N) * (heart_total/N) * N
aspirin_heart_expected
```

```
## [1] 146.4004
```

3. Calculate the $X^2$ value using your observed and expected counts.

```
chisqvalue = sum( ((placebo_healthy - placebo_healthy_expected)^2)/placebo_healthy_expected +
                  ((placebo_heart - placebo_heart_expected)^2)/placebo_heart_expected +
                  ((aspirin_healthy - aspirin_healthy_expected)^2)/aspirin_healthy_expected +
                  ((aspirin_heart - aspirin_heart_expected)^2)/aspirin_heart_expected
                  )

chisqvalue
```

```
## [1] 24.87352
```

4. What is the $p$-value of the test statistic?

```
pchisq(chisqvalue, df = 1, lower.tail = F)
```

```
## [1] 6.121763e-07
```

```
chisq.test(aspirindata)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  aspirindata
## X-squared = 24.29, df = 1, p-value = 8.285e-07
```

5. Calculate the $p$-value using Fisher's exact test.

```
fisher.test(aspirindata)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  aspirindata
## p-value = 6.596e-07
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  1.430030 2.350038
## sample estimates:
## odds ratio
##   1.828967
```

6. Calculate the $p$-value using the `phyper` function.

phyper(q, m, n, k, lower.tail = TRUE, log.p = FALSE)

x = , q vector of quantiles representing the number of white balls drawn without replacement from an urn which contains both black and white balls.

m
the number of white balls in the urn.

n
the number of black balls in the urn.

k
the number of balls drawn from the urn.

p
probability, it must be between 0 and 1.

```
x = aspirin_heart
m = aspirin_total
n = placebo_total
k = heart_total

phyper(x,m,n,k,lower.tail = T )
```

```
## [1] 3.495616e-07
```