Week 5 Notes Part 2 Contingency Table

Code

Author: Brendan Gongol

Last update: 07 January, 2023

1 Overview

We start this Note with one more example of ANOVA analysis, this time for (continuous) linear regression $Y \sim X$. Next we switch the topic and discuss the analysis of association between a pair of categorical random variables. To this end, we introduce contingency tables and show how chi-square and exact Fisher test can be applied to test for such two-way associations.

2 ANOVA Example: Continuous Variables

Before moving to the main topic of this Note, let us consider one more example of ANOVA calculation (moved here mainly to make the Notes a little bit more balanced in size). This time we consider ANOVA for a linear model on continuous variables:

Hide

```
library(ALL); data(ALL)
ALL.pdat <- pData(ALL)
date.cr.chr <- as.character(ALL.pdat$date.cr)
diag.chr <- as.character(ALL.pdat$diagnosis)
date.cr.t <- strptime(date.cr.chr,"%m/%d/%Y")
diag.t <- strptime(diag.chr,"%m/%d/%Y")
days2remiss <- as.numeric(date.cr.t - diag.t)
##############################################
d2r.df.34852 <- data.frame(
 G=exprs(ALL)["34852_g_at",],
 D2R=as.numeric(days2remiss))
lm.34852 <- lm(D2R~G,d2r.df.34852)
anova(lm.34852)
```

vector of remission dates

vector of diagnosis dates

convert to format that allows arithmetic.

vector of integers, the index of each corresponds to index of patient/sample in exprs(ALL).

df w 2 columns - (gene exprs of 34852_g_at, days until remission). rows - n of patients/samples.

dependent

independent.

Q: is gene expression of 34852_g_at predictive of number of days to remission.

```
## Analysis of Variance Table
##
## Response: D2R
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## G          1  6000.1  6000.1  30.498 2.959e-07 ***
## Residuals 94 18493.6   196.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

SS between = explained variation

low p-value → significant relationship b/w variables.

Hide

```
y <- d2r.df.34852$D2R; y <- y[!is.na(y)] # y=days-to-remission
y.hat <- predict(lm.34852)               # fitted (predicted) values
# predicted values with intercept-only model:
y.hat.wo.G <- predict(lm(D2R~1,d2r.df.34852))
range(y.hat.wo.G-mean(y))
```

y.hat.wo.G - mean(y) computes diff b/w each predicted value + the mean of y.

then range returns mins + max. of the differences

min + max are same suggesting differences are 0.

predicted values of linear model with intercept only is same as the mean of the dependent variable

```
## [1] 1.421085e-14 1.421085e-14
```

Hide

```
eps <- resid(lm.34852)
range(y.hat+eps-y) # just a sanity check. Y IS y_predicted + residual
```

y.hat eps

y.hat + eps = reconstructs y from the model

y.hat + eps - y : computes the difference b/w the reconstructed y values + original y values. range then finds the min and max of the differences.

```
## [1] 0.000000e+00 3.126388e-13
```

indicates that the difference is very close to zero.

Hide

```
sum((y.hat-y.hat.wo.G)^2)                # SSbetween
```

sum of sq of full model - intercept only model (ie: mean of y)

```
## [1] 6000.107
```

SS between 2 models (w/ independ. var + w/o).

Hide

```
sum(resid(lm.34852)^2)                   # residuals=(remaining noise), SSwithin
```

```
## [1] 18493.61
```

SS of the residuals

Hide

```
sum((y.hat-y.hat.wo.G)^2)/(sum(resid(lm.34852)^2)/94) # F
```

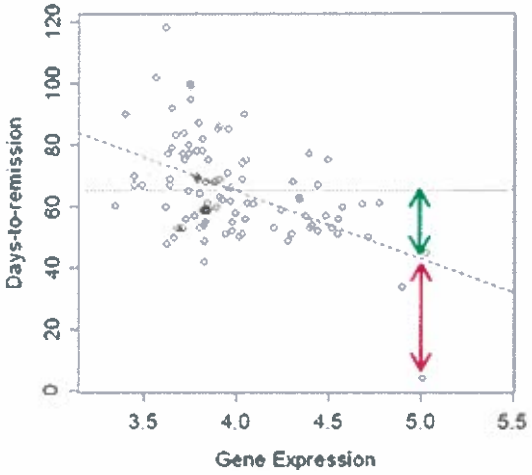
$$\frac{\text{SS between}}{\text{SS within : df}}$$

```
## [1] 30.49756
```

In the code fragment above, we first fit a linear model (days-to-remission vs. gene expression level, just to peruse a familiar example). Let us make a note of the `anova()` output on that model, and try to reproduce the numbers. For convenience, we discard NA's and save the days-to-remission values into variable y. We also save values predicted by our fitted model into `y.hat`. Next we fit a model with intercept-only. Since this

latter model has only intercept term and no dependence on gene expression (slope is 0, $y = b + \varepsilon$), it obviously predicts the same value for y regardless of the value of the independent variable x (gene expression level), and this predicted value is simply the mean of the whole corpus of observations for y (as we confirm by looking at difference between the model's prediction and the mean). According to Eq.(11) from Part1, the "between-group" variance (from data point to data point, i.e. the variance explained by our fitted `model`) is $\sum_i (\bar{y}_i - \bar{y})^2$, i.e. sum of squares of differences between predicted value in each "group" (i.e. at each "level", or value of independent variable x) and the total mean of all) and the total mean of all observations of y. The corresponding calculation in our code returns exactly the number reported as `Sum.Sq` . for the G term (i.e. slope) in the output of the `anova()` test above. The sum of squared residuals is the "within-group" variability (noise) – something that is not explained by our model. As we can see, this sum is again the same as reported in anova() output in the 'Residuals' line. Finally, we calculate the F statistics (taking into account the number of degrees of freedom – we have 96 datapoints total), and it matches exactly the number reported by the `anova()` .

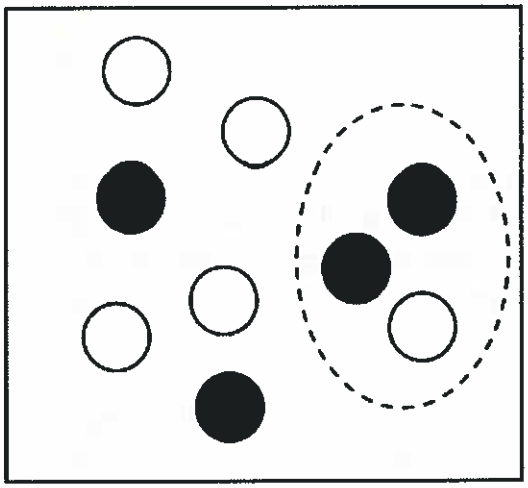
The data and the fitted regression line are shown in the figure below. The arrows indicate, for just one data point, the contribution to the between-group variance (or "effect" – that's how much of a difference between this measurement and the overall mean of all measurements is captured by our model, green arrow), and the contribution to the within-group variance ("noise", red arrow: that's the difference between observed and predicted value, the residual; we could not do any better and this part of variability remains "unexplained").



3 Contingency Tables

Let us now consider the case when we have two categorical variables, each with its own levels (two or more). Since both variables have discreet (categorical) values, all we can observe are the counts of objects in different categories, and we can ask questions such as: "is any combination of our categorical factors more likely than others" or "is there any association between the factors" (e.g. if we select people randomly and count males/females and football players/non-players, we may observe that football players tend to be males).

Contingency table is simply a table that counts occurrences of different combinations of our categorical factors. Let us consider a simple example: we have a box filled with balls, and we know that the balls come in two colors (black and white) and in two weights (light and heavy). We select 9 balls, 4 black and 5 white, and we discover that 3 of them, 2 black and 1 white, are heavy. The figure below illustrates what happens in our example (the dashed outline is drawn around the heavier balls). Looking at the counts of the balls with different characteristics, we can try inferring if there is any association between color and weight, i.e whether black balls tend to be heavier (or lighter).



The contingency table in our case is going to have dimensions 2x2 (two categorical variables with two levels each), and it looks as shown below; we have also added column totals and row totals to our contingency table, which are known as marginal totals (total number of white balls, total number of black balls, total number of heavy, total number of light); the total number of observations in the whole table is called a grand total.

	Light	Heavy	Row Total
Black	2	2	4
White	4	1	5
Column Total	6	3	

The "classical" test that can be applied to a contingency table is Pearson's chi-square test. Namely, we set up a null hypothesis, under which we assume that there is no association between the categorical variables under study. Under this null, we calculate the expected counts in each cell of the table and evaluate whether the differences between the observed and expected values are significant or, on the contrary, can be quite expected simply due to random sampling (this generic description of hypothesis testing procedure should sound very familiar to you by now). Let us first calculate the expected values. Since we do not know the underlying proportions, we can only estimate them from the data, there is no way to do better. For the fraction of the black and white balls (or, equivalently, the probability to draw a black or white ball), we thus get:

P(black)=4/9; P(white)=5/9; For the probability being heavy/light, we get P(heavy)=3/9; P(light)=6/9.

Note that we used marginal totals here, e.g. P(black)=4/9, where 4 is (obviously) the total number of black balls, or the marginal total of the black row in the table above (regardless of the 'weight' status). 9 is the total number of observations we have in our experiment.

Under the null hypothesis, there is no association between the two categorical variables we are studying, i.e. they are independent. But, by definition of the independence, we have then, for instance, P(black|heavy)=P(black), i.e. the knowledge that the ball is heavy does not help us better predict if it's black

– the conditional probability of the selected ball to be black (i.e. probability given the ball is heavy) is the same as "unconditional" marginal probability. Hence, the joint probability for the ball to be black and heavy is

P(black,heavy)=P(black|heavy)P(heavy)=P(black)P(heavy)=(4/9)*(3/9),

And hence the expected number of heavy black balls (top-right cell in the table above) is N*P(black,heavy), i.e. 9*(4/9)*(3/9)=(4*3)/9=12/9=1.3333.

We can easily generalize our derivation: let us assume that we have two categorical variables X, Y, with P and Q levels, respectively, and we perform grand total of N measurements and fill a P x Q contingency table similar to the table above. We thus have marginal row totals

$$M_k^{(r)} = \sum_{i=1}^{Q} O_{ki}, \quad k = 1...P$$

and marginal column totals

$$M_l^{(c)} = \sum_{j=1}^{P} O_{ji}, \quad l = 1...Q$$

where O_{ij} is the count of observation in the cell of the contingency table located in the i-th row and j-th column.

The fractions (probabilities) of each value (level) of our categorical variables are

$$P(X = k) = M_k^{(r)}/N$$

$$P(Y = l) = M_l^{(c)}/N$$

And under the condition of independence of X and Y, which is our null hypothesis, the expected value in the cell i, j of our contingency table is

$$E_{kl} = N \cdot P(X = k) \cdot P(Y = l) = M_k^{(r)} M_l^{(c)}/N = (\sum_{i=1}^{Q} O_{ki})(\sum_{j=1}^{P} O_{jl})/N$$

Now that we calculated the expected values under the null hypothesis assumptions, we need to construct a statistic to evaluate. In Pearson's chi-square test, this statistic is sum of squared deviations of observations in each cell from the expected values (normalized to the expected values themselves):

$$X^2 = \sum_{i=l}^{P} \sum_{j=1}^{Q} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

It can be shown that asymptotically (at large number of observations) the distribution of this test statistics approaches chi-square distribution with $(P-1)(Q-1)$ degrees of freedom (hence the name of the test).

Let us consider a numerical example that uses the same data as shown above:

Hide

```
x.df<-data.frame(
 Col=c("B","B","B","B","W","W","W","W","W"),
 Weight=c("H","H","L","L","L","L","L","L","H"))
table(x.df)
```

```
##    Weight
## Col H L
##   B 2 2
##   W 1 4
```

Hide

```
x.chi <- chisq.test(table(x.df),correct=F)
print(x.chi)
```

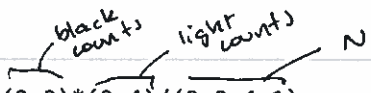
```
##
##  Pearson's Chi-squared test
##
## data:  table(x.df)
## X-squared = 0.9, df = 1, p-value = 0.3428
```

Hide

```
x.chi$expected
```

```
##    Weight
## Col        H        L
##   B 1.333333 2.666667
##   W 1.666667 3.333333
```

Hide




```
(2+2)*(2+4)/(2+2+4+1)
```

```
## [1] 2.666667
```

Hide

```
sum((table(x.df)-x.chi$expected)^2/x.chi$expected)
```

```
## [1] 0.9
```

In the code shown above, we first create a data frame that holds our data. Each row is an observation, or case, and the values of the two categorical variables Col (color) and Weight are provided for each case.

The familiar command `table()`, as we can see works not only with single categorical variables (when counts of observations in each category are represented by a vector), but also with multiple variables. In our case, when applied to a dataframe with two categorical variables, `table()` generates just the contingency

table that we need: counts of cases for each combination of the levels of our variables. Of course you can manually create a matrix and fill it with the counts of observations for each combination of factor levels: the test implementations discussed below will happily work with such a matrix – it does not matter where the matrix came from.

Next we run pearson's chi-square test (`chisq.test()`) on the contingency table and observe that the resulting p-value is ~0.3, i.e. there is no significant bias of the black/white proportion among the light and heavy balls. The object returned by `chisq.test()` function contains additional useful information, such as, e.g.the expected counts, and in our code example we print these expected counts out. Make sure you understand the expected values in all cells -they can be obtained with the same logic as discussed above (it is left for the homework).Next line of code calculates the expected value in the top left cell of the table using the marginal counts, just like it was explained above. The last line demonstrates that sum of squared deviations from expected, normalized by the expected values, indeed returns the value of the statistics reported by `chisq.test()` . This concludes the example, as we know completely understand the calculations behind chi-square test.

3.1 Fisher Exact Test

The chi-square test we just considered is very important historically (and methodologically), however it has its drawbacks. First, the distribution of the test statistics becomes what we expect it to be only at moderately large N. There is no rigorous threshold of course, but many sources suggest "rules of thumb" of having at least 5-10 observations in each cell of the contingency table before you can reliably use Pearson's test; it is also suggested that the counts should be relatively uniform across the cells, as in very extreme cases the distribution of the test statistics may deviate significantly from the expected chi-square distribution.

There is another test that can be applied to the same problem; this test is known as Fisher's exact test. As the name suggests, this test makes no assumptions at all about the underlying distributions (and thus is an example of non-parametric test). We will not go into the derivation of analytical expressions behind the Fisher test, but the idea is as follows: consider a contingency table like the one we have examined above. While keeping the total number of observations (grand total) and under the constraint of fixed marginals, consider all possible reassignments of our cases into the cells of the table. Out of all such possible tables, count the ones that have "more extreme" arrangement than the one we actually observed (i.e. have even larger differences between the cells). The fraction of such "more extreme" contingency tables out of all possible contingency tables, obviously, gives some kind of p-value. The analytical expression for the probability used in Fisher's test is given by:

$$p = \frac{\binom{a+b}{a}\binom{c+d}{c}}{\binom{n}{a+c}} = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{a!b!c!d!n!}$$

where $C_n^k = \binom{n}{k}$ are binomial coefficients ("choose k from n" – see our earlier discussion of combinatorics), and the values in the contingency table are encoded as shown below (perusing our example):

	Light	Heavy	Total
Black	a 2	b 2	$a+b$ 4
White	c 4	d 1	$c+d$ 5
Totals	$a+c$ 6	$b+d$ 3	$a+b+c+d\ (=N)$ 9

The expression for the probability above is a hypergeometric distribution (and for this reason Fisher's test is sometimes also referred to as hypergeometric test). The expression shown above is hard to compute for large numbers and this difficulty alone used to be an important reason for choosing chi-square test. Today, however, with extensive computing power and efficient algorithms available, applying Fisher's test is perfectly feasible even with large numbers of observations (even though it maybe just an unnecessary overkill, as the statistics does converge to chi-square at large N).

Let us work through an example, where we will analyze the same contingency table using Fisher's test:

Hide

```
fisher.test(table(x.df),alternative="greater")$p.value
```

```
## [1] 0.4047619
```

Hide

a+b a c+d c n a+c

```
choose(4,2)*choose(5,4)/choose(9,6)
```

```
## [1] 0.3571429
```

dhyper(x,m,n,k)

x = a (observed black in the sample) = 2
m = a+b (total black in population) = 4
n = c+d (total white in pop) = 5
k = number items sampled = 3 (we took out 3).

Hide

```
dhyper(2,4,5,3)                    #P{Black=2}
```

```
## [1] 0.3571429
```

Hide

```
dhyper(3,4,5,3)                    #P{Black=3}
```

```
## [1] 0.04761905
```

Hide

```
dhyper(4,4,5,3)                    #P{Black=4}
```

```
## [1] 0
```
← we only pulled out 3, can't have 4 black.

Hide

```
dhyper(2,4,5,3)+dhyper(3,4,5,3) #P{Black>=2}
```
more extreme cases than pulling black=2
add what we have + more extreme case.

```
## [1] 0.4047619
```

calc the probability of having more than 1.

Hide

```
phyper(1,4,5,3,lower.tail=F)    #P{Black>1}
```
CDF.
tells R to compute the prob of seeing values greater than q=1, rather than upto q=1

```
## [1] 0.4047619
```

The first line is all you normally need to do: we simply apply fisher test to the contingency table, and we are done (reported p-value of 0.4 suggests that our data do not support significant association between color and weight, same conclusion as we have achieved with chi-square test; statistically speaking, maybe there is indeed no association, or maybe we do not have enough data to discern it – we never know!).

Next we try evaluating by hand the expressions that go into the Fisher test calculations. First, using the expression for the probability above, we compute the probability to observe by random chance the contingency table that we have in hand. Next line computes exactly the same expression, but using R's built-in function for hypergeometric distribution. This probability is not a p-value yet: we also need probabilities for "more extreme" tables (with even larger counts in a particular cell – we chose top-left, but it does not matter, the expressions are symmetric). So next we compute those probabilities, and summing them up we arrive at exactly the same value as reported by the Fisher test (0.4047...). We can also obtain the same value without employing manual summation if we use directly the CDF of the hypergeometric distribution instead of probability densities.

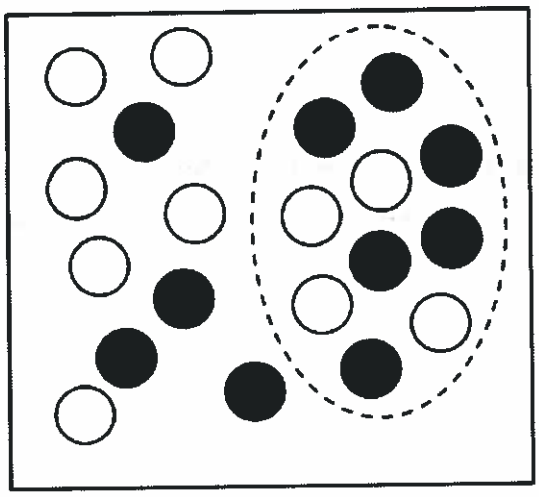
Note that both chi-square and Fisher test can deal with contingency tables built on categorical variables with more than two levels, i.e. on general M x N contingency tables. However, for Fisher test the calculation may become prohibitive even with modern computers (the implementation is likely to invoke some Monte Carlo simulation!).

To conclude this section, let us compare Fisher test and Chi-square test. Let us consider the following scenario: we have a box with 20 balls, 10 black and 10 white, and we are drawing 10 balls from the box (one example of such selection is shown with dashed outline in the diagram). We observe, e.g. 4 white balls among the selected ones and we are asking if the selection is biased towards (or against) white balls, i.e. if there is an association between the color variable and 'selection status' variable.

dhyper(): prob mass function (PMF)
phyper(): cumulative dist function (CDF).

gives prob of exactly 1 draw – used when you need prob of a specific event, like getting exactly 2 blacks. ∴ need to add them together if wanting to know prob of equal or more extreame.

computes prob of getting more than _ successful draws.



Note that 1) the problem is actually the same as the one we have considered before, we are just using "selection status" as a variable, instead of "weight"; 2) such interpretation of the selection process as a random variable (selected/not selected) is very useful in many applications. Suppose, you run a high-throughput experiment, for instance a microarray, where you test N genes at once. This experiment selects K genes for you (for instance, you found them to be differentially expressed between the two conditions). You also have a biological annotation for each of N genes: whether the gene belongs ("black") or does not belong ("white") to a particular biological process, for the sake of certainty let it be "apoptosis". Among your K selected genes, some are annotated as being related to the apoptosis process, some are not (i.e. some are "black" and some are "white").

A very important biological question you can (and should) ask in this situation is: "do the genes differentially expressed between the two conditions under study tend to be associated with apoptosis" (or with any other biological process for that matter). When we use the mapping onto the problem of looking for an association between the two categorical random variables, as described above, we immediately see that the answer to this important question can be found by simply running Fisher (or chi-square) test.

But let us get to the example code now:

Hide

```
Nw <- 10
Nb <- 10
Nd <- 10
p.chi <- numeric()
p.hyp <- numeric()
p.fish <- numeric()
p.fish.2 <- numeric()
for ( Nwd in 0:10 ) {
 x.df <- data.frame(C=c(rep("W",Nw),
 rep("B",Nb)),
 D=c(rep("Y",Nwd),
 rep("N",Nw-Nwd),
 rep("Y",Nd-Nwd),
 rep("N",Nb-Nd+Nwd)))
 x.tbl <- table(x.df)
 p.chi[Nwd+1] <- chisq.test(x.tbl,correct=F)$p.value
 p.hyp[Nwd+1] <- phyper(Nwd-1,Nw,Nb,Nd,lower.tail=F)
 p.fish[Nwd+1] <- fisher.test(x.tbl,alternative='greater')$p.value
 p.fish.2[Nwd+1] <-
fisher.test(x.tbl,alternative='two.sided')$p.value
}
old.par <- par(lwd=2,ps=20)
plot(c(0,10),c(0,1),type="n",xlab="Nwd",ylab="p-val")
points(0:10,p.chi,type="l",col="black")
points(0:10,p.hyp,type="l",col="blue")
points(0:10,p.fish,type="p",col="green")
points(0:10,p.fish.2,type="p",col="magenta")
text(6,0.9,"chisq",pos=4)
text(6,0.8,"phyper",col="blue",pos=4)
text(6,0.7,"Fisher greater",col="green",pos=4)
text(6,0.6,"Fisher 2-sided",col="magenta",pos=4)
```

array to store p-vals from chi-sq.

from hypergeometric test (fisher.test)

from fishers 1-sided

from fishers 2-sided.

x.df represents the selection status of balls

number of white balls drawn from 0-10.

rep() ← replicates elements in vectors.

creates a combined vector of 10 W's followed by 10 B's.

(Nw=10)

column C: represents color.

D: represents selection status. (Y-selected, N-not selected)

*[Nwd+1] because range is 0-10, but once stored in a vector the first index is 1.

eg: for the third iteration of the loop Nwd = 3

x.df ← data.frame(C = c(rep("W", Nw(10)), rep("B", Nb(10))),

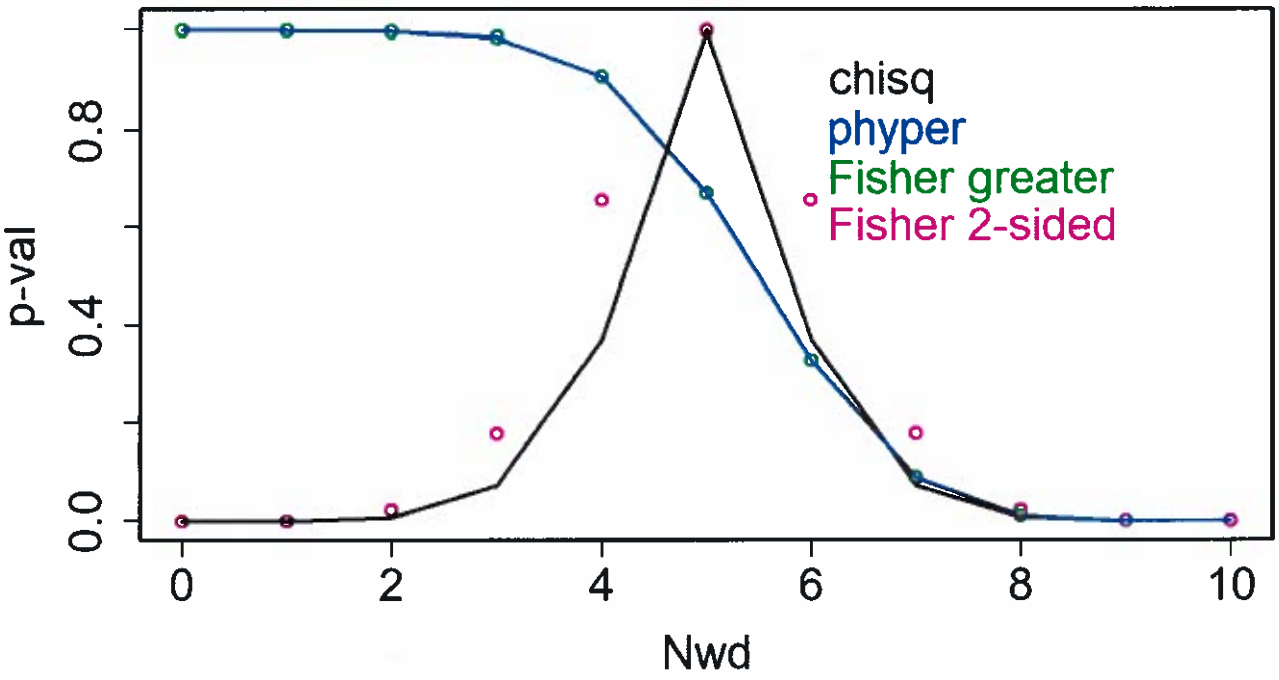
D = c(rep("Y", Nwd(3)), rep("N", Nw - Nwd(7)),

rep("Y", Nd - Nwd(7)), rep("N", Nb - Nd + Nwd(10-10+3=3))))

∴ D = c(Y, Y, Y, N, N, N, N, N, N, N, Y, Y, Y, Y, Y, Y, Y, N, N, N)

So the df looks like:

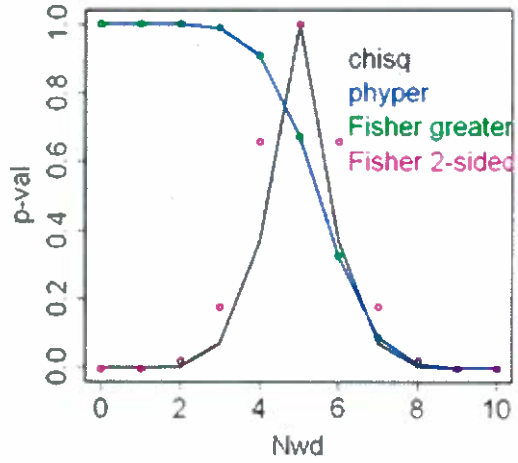
C	D
W	Y
W	Y
W	Y
W	N
B	N
⋮	⋮



Hide

```
par(old.par)
```

The figure resulting from running this code is shown below. In the code fragment, we simulate different selections of 10 balls from a set of 10 white + 10 black balls. Each selection that we make differs by the number Nwd of white balls that end up being selected. For each such selection we calculate p-values using chi-square, one sided Fisher, two-sided Fisher, and also directly using CDF of hypergeometric distribution (which we know is supposed to be exactly what Fisher test returns, so we are just checking this statement here). As we can see, selection of 5 white balls out of total of 10 selected is the least significant (which makes perfect sense, of course). The small difference between chi-square and Fisher is due to approximations made by chi-square test; you can see that those differences are not too dramatic.



Let us consider one last and more practical example using real-life data from the ALL dataset. In this example we will assess the association (or lack thereof) between the patient's sex and remission status (CR=complete remission; REF=refractory disease). You should appreciate how fast and easy it can be done in R (assuming all the data are already packed into the ready-to use data frame!!):

Hide

```
table(pData(ALL)$sex,pData(ALL)$remission)
```

```
##
##     CR REF
##   F 27   7
##   M 71   8
```

Hide

```
chisq.test(table(pData(ALL)$sex,pData(ALL)$remission))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(pData(ALL)$sex, pData(ALL)$remission)
## X-squared = 1.4424, df = 1, p-value = 0.2298
```

Hide

```
chisq.test(table(pData(ALL)$sex,pData(ALL)$remission),correct=F)
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(pData(ALL)$sex, pData(ALL)$remission)
## X-squared = 2.2598, df = 1, p-value = 0.1328
```

Hide

```
fisher.test(table(pData(ALL)$sex,pData(ALL)$remission))
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  table(pData(ALL)$sex, pData(ALL)$remission)
## p-value = 0.1429
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.1248219 1.5680294
## sample estimates:
## odds ratio
##  0.4381879
```

All that the code above does is 1) compute the contingency table (how many males and females are in remission or are refractory): we do not actually need this line to run the test, but it is always a good practice to review the data first!; 2) apply chi-square or Fisher test to the contingency table. All the tests return insignificant p-value, i.e. our data do not support the existence of association between the sex and remission status.

Note the two flavors of the chi-square test that we used here. The first invocation (default) uses so-called Yates' continuity correction. The rationale for its existence is the fact that contingency tables contain counts, i.e. discrete numbers. We approximate distributions of those counts (or statistics computed from them, anyway) by continuous chi-square distribution, so, intuitively, it is natural to expect that our probability estimates can be a bit off. Yates' suggested an adjustment to the χ^2 statistics we have introduced above, which amounts to subtracting 0.5 from the (integer) cell counts. This ad-hoc correction is not perfect and the question whether it has to be used at all is debated. In any case, the correction is more justified when the number of observations and/or counts in at least some cells are extremely small (as you can imagine, when counts are very large, subtracting 0.5 from them is not going to have strong relative effect anyway).

In the second attempt to run chi-square test in the code above, we invoke it with Yates' correction turned off. Note that it results in smaller p-value AND that this p-value is much closer to the exact p-value computed by the Fisher test. This suggests that the second answer is correct, and that Yates' correction is way too conservative for our data.