# Contents

# Chapter 6
# Multiple Testing

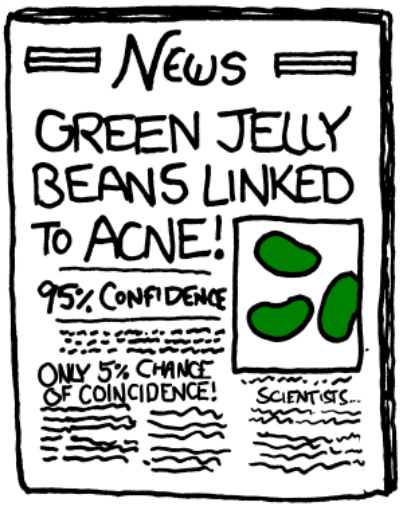

Figure 6.1: From http://xkcd.com/882

Hypothesis testing is one of the workhorses of science. It is how we draw conclusions or make decisions based on finite samples of data. For instance, new drugs are usually approved on the basis of clinical trials that aim to decide whether the drug has better efficacy (and an acceptable trade-off of side effects) compared to the other available options. Such trials are expensive and can take a long time. Therefore, the number of patients we can enroll is limited, and we need to base our inference on a limited sample of observed patient responses. The sample needs to be big enough allow us to make a reliable conclusion, but small enough not to waste precious resources or time. The machinery of hypothesis testing was developed largely with this application in mind, although today it is used much more widely.

## 6.1 Goals for this Chapter

- Familiarize ourselves with the machinery of hypothesis testing, its vocabulary, its purpose, and its strengths and limitations.
- Understand what multiple testing means.
- See that multiple testing is not a problem — but rather, an opportunity, as it fixes many of the limitations of single testing.
- Understand the false discovery rate.
- Learn how to make diagnostic plots.
- Use hypothesis weighting to increase the power of our analyses.

### 6.1.1 Drinking from the firehose

If statistical testing —reasoning with uncertainty— seems a hard task if you do it for one single decision (or test), then brace yourself: in genomics, or more generally with "big data", we need to accomplish it not once, but thousands or millions of times. You've already seen in this Chapter 7, where we analysed RNA-Seq data for differential expression. We applied a hypothesis test to each of the genes, that is, we did several thousand tests. Similarly, in whole genome sequencing, we scan every position in the genome for a difference between the sample at hand and a reference (or, another

sample): that's on the order of 3 billion tests if we are looking at human data! In RNAi or chemical compound screening, we test each of the reagents for an effect in the assay, compared to control: that's again tens of thousands, if not millions of tests.

Yet, in many ways, the task becomes simpler, not harder. Since we have so much data, and so many tests, we can ask questions like: are the assumptions of the tests actually met by the data? What are the prior probabilities that we should assign to the possible outcomes of the tests? Answers to these questions can be incredibly helpful, and we can address them **because** of the multiplicity. So we should think about it not as a "multiple testing problem", but as an opportunity!

Figure 6.2: Modern biology often involes navigating a deluge of data. Source

There is a powerful premise in data-driven sciences: we usually expect that most tests will not be rejected. Out of the thousands or millions of tests (genes, positions in the genome, RNAi reagents), we expect that only a small fraction will be interesting, or "significant". In fact, if that is not the case, if the hits are not rare, then arguably our analysis method –serially univariate screening of each variable for association with the outcome– is not suitable for the dataset. Either we need better data (a more specific assay), or a different analysis method, e. g., a multivariate model.

Since most nulls are true, we can use the behaviour of the many test statistics and p-values to empirically understand their null distributions, their correlations, and so on. Rather than having to rely on **assumptions** we can check them empirically!

### 6.1.2 Testing vs classification

There are many methodological similarities between hypothesis testing and classification, but the differences are good to keep in mind. In both cases, we aim to use data to choose between several possible decisions. For instance, we might use the measured expression level of a marker gene to decide whether the cells we're studying are from cell type A or B. If we have no prior assumption, and if we're equally worried about mistaking an A for a B, or vice versa, then we're best served by the machinery of classification as covered briefly in Chapter **??** and in detail in [1]. On the other hand, if –before seeing the data– we have a preference for A, and need evidence to be convinced otherwise, then the machinery of hypothesis testing is right for us. For instance, if a disease is currently treated with some established medication, and someone comes up with a proposal to treat it with a different treatment instead, the burden of proof should be with them, and the data should prove the benefit of the new treatment with high certainty. We can also think of this as an application of **Occam's razor**[2] – don't come up with a more complicated solution if a simpler one does the job.

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. **The Elements of Statistical Learning**. Springer, 2008

[2] See also `https://en.wikipedia.org/wiki/Occam%27s_razor`

## 6.2 An Example: Coin Tossing

To understand multiple tests, let's first review the mechanics of single hypothesis testing. For example, suppose we are flipping a coin to see if it is a fair coin[3]. We flip

[3] The same kind of reasoning, just with more details, applies to any kind of gambling. Here we stick to coin tossing since everything can be worked out easily, and it shows all the important concepts.

the coin 100 times and each time record whether it came up heads or tails. So, we have a record that could look something like this:

```
H H T T H T H T T ...
```

Which we can simulate in R. We set `probHead` different from 1/2, so we are sampling from a biased coin:

```
set.seed(0xdada)
numFlips <- 100
probHead <- 0.6
coinFlips <- sample(c("H", "T"), size = numFlips,
  replace = TRUE, prob = c(probHead, 1 - probHead))
head(coinFlips)
## [1] "T" "T" "H" "T" "H" "H"
```

Now, if the coin were fair, we expect half of the time to get heads. Let's see.

```
table(coinFlips)
## coinFlips
##  H  T
## 59 41
```

So that is different from 50/50. Suppose we didn't know whether the coin is fair or not – but our prior assumption is that coins are, by and large, fair: would these observed data be strong enough to make us conclude that this coin isn't fair? We know that random sampling differences are to be expected. To decide, let's look at the sampling distribution of our test statistic –the total number of heads seen in 100 coin tosses– for a fair coin[4]. This is really easy to work out with elementary combinatorics:

$$P(K = k \,|\, n, p) = \left( \begin{array}{c} n \\ k \end{array} \right) p^k \, (1-p)^{n-k} \tag{6.1}$$

Let's parse the notation: $n$ is the number of coin tosses (100) and $p$ is the probability of head (0.5 if we assume a fair coin). $k$ is the number of heads. Statisticians like to make a difference between all the possible values of a statistic and the one that was observed, and we use the lower case $k$ for the possible values (so $k$ can be anything between 0 and 100), and the upper case $K$ for the observed value. We pronounce the left hand side of the above equation as "the probability that the observed number takes the value $k$, given that $n$ is what it is and $p$ is what it is".

Let's plot Equation (6.1); for good measure, we also mark the observed value `numHeads` with a vertical blue line.

```
k <- 0:numFlips
numHeads <- sum(coinFlips == "H")
binomDensity <- data.frame(k = k,
     p = dbinom(k, size = numFlips, prob = 0.5))
```

```
library("ggplot2")
ggplot(binomDensity) +
  geom_bar(aes(x = k, y = p), stat = "identity") +
  geom_vline(xintercept = numHeads, col="blue")
```

Suppose we didn't know about Equation (6.1). We could still manoeuvre our way out by simulating a reasonably good **approximation** of the distribution.

[4] We haven't really defined what we mean be fair – a reasonable definition would be that head and tail are equally likely, and that the outcome each coin toss is completely independent of the previous ones. For more complex applications, nailing down the exact null hypothesis can take a bit more thought.

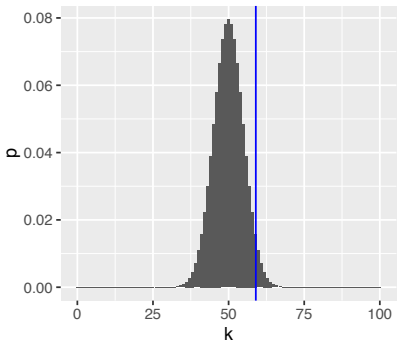


Figure 6.3: The binomial distribution for the parameters $n = 100$ and $p = 0.5$, according to Equation (6.1).

```
numSimulations <- 10000
outcome <- replicate(numSimulations, {
  coinFlips <- sample(c("H", "T"), size = numFlips,
    replace = TRUE, prob = c(0.5, 0.5))
  sum(coinFlips == "H")
})
ggplot(data.frame(outcome)) + xlim(0, 100) +
  geom_histogram(aes(x = outcome), binwidth = 1, center = 0) +
  geom_vline(xintercept = numHeads, col="blue")
```

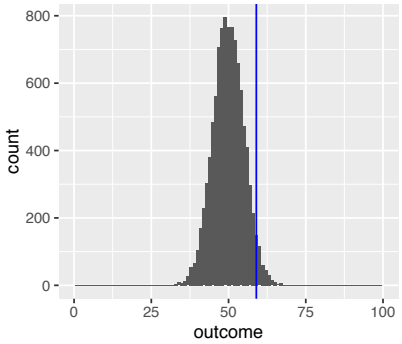


Figure 6.4: An approximation of the binomial distribution from $10^4$ simulations (same parameters as Figure 6.3).

As expected, the most likely number of heads is 50, that is, half the number of coin flips. But we see that other numbers near 50 are also not unlikely. How do we quantify whether the observed value, 59, is among those values that we are likely to see from a fair coin, or whether its deviation from the expected value is already big enough for us to conclude with enough confidence that the coin is biased? We divide the set of all possible $k$'s (0 to 100) in two complementary subsets, the **acceptance region** and the **rejection region**. A natural choice[5] is to fill up the rejection region with as many $k$ as possible while keeping the total probability below some threshold $\alpha$ (say, 0.05). So the rejection set consists of the values of $k$ with the smallest probabilities (6.1), so that their sum remains $\leq \alpha$.

[5] More on this below.

```
library("dplyr")
alpha <- 0.05
binomDensity <- arrange(binomDensity, p) %>%
        mutate(reject = (cumsum(p) <= alpha))

ggplot(binomDensity) +
  geom_bar(aes(x = k, y = p, col = reject), stat = "identity") +
  scale_colour_manual(
    values = c(`TRUE` = "red", `FALSE` = "darkgrey")) +
  geom_vline(xintercept = numHeads, col="blue") +
  theme(legend.position = "none")
```

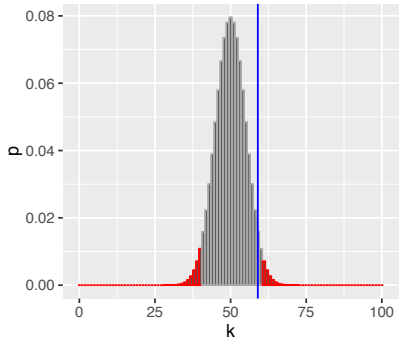


Figure 6.5: As Figure 6.3, with rejection region (red) that has been chosen such that it contains the maximum number of bins whose total area is at most $\alpha = 0.05$.

In the code above, we used the functions `arrange` and `mutate` from the *dplyr* package to sort the the p-values from lowest to highest, compute the cumulative sum (`cumsum`), and stop rejecting once it exceeds `alpha`.

The explicit summation over the probabilities is clumsy, we did it here for pedagogic value. For one-dimensional distributions, R provides not only functions for the densities (e. g., `dbinom`) but also for the cumulative distribution functions (`pbinom`), which are more precise and faster than `cumsum` over the probabilities. These should be used in practice.

We see in Figure 6.5 that the observed value, 59, lies in the grey shaded area, so we would **not** reject the null hypothesis of a fair coin from these data at a significance level of $\alpha = 0.05$.

Question **6.2.1** *Does the fact that we don't reject the null hypothesis mean that the coin is fair?*

Question **6.2.2** *Would we have a better chance of detecting that the coin is not fair if we did more coin tosses? How many?*

Question **6.2.3** *If we repeated the whole procedure and again tossed the coin 100 times, might we **then** reject the null hypothesis?*

Question **6.2.4** *The rejection region in Figure 6.5 is asymmetric - its left part ends with $k = 40$, while its right part starts with $k = 61$. Why is that? Which other ways of defining the rejection region might be useful?*

The binomial test is such a frequent activity that it has been wrapped into a single function, and we can compare its output to our results

```
binom.test(x = numHeads, n = numFlips, p = 0.5)
##
##  Exact binomial test
##
## data:  numHeads and numFlips
## number of successes = 59, number of trials = 100, p-value =
## 0.08863
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.4871442 0.6873800
## sample estimates:
## probability of success
##                   0.59
```

## 6.3 The Five Steps of Hypothesis Testing

Let's summarise the general principles[6] of hypothesis testing:

[6] These are idealised; for a reality check, see below, Section 6.6.

1. Choose an experimental design and a data summary function for the effect that you are interested in, the **test statistic**.
2. Set up a **null hypothesis**, which is a simple, computationally tractable model of reality that lets you compute the **null distribution**, i. e., the possible outcomes of the test statistic and their probabilities.
3. Decide on the **rejection region**, i. e., a subset of possible outcomes whose total probability is small.
4. Do the experiment, collect data, compute the test statistic.
5. Make a decision: reject the null hypothesis if the test statistic is in the rejection region.

Null hypothesis

The null hypothesis we used in the coin tossing example was that heads and tails are equally likely, and that the outcome of each coin toss is independent of the previous ones. This is idealized: a real coin might have some, if ever so slight irregularities, so that the probability of head might be 0.500001; but here we don't worry about that, nor about any possible effects of air drag, elasticity of the material on which the coin falls, and so on. It is also computationally tractable, namely, with the binomial distribution.

Test statistic

The test statistic in our example was the total number of heads. Suppose we observed 50 tails in a row, and then 50 heads in a row. Our test statistic ignores the order of the outcomes, and we would conclude that this is a perfectly fair coin. However, if we used a different test statistic (say, the number of times we see two tails

in a row), we might notice that there is something funny about this coin.

Question **6.3.1** *What is the null distribution of this different test statistic?*

Question **6.3.2** *Would a test based on that statistic be generally preferable?*

What we have just done is that we looked at two different classes of **alternative hypotheses**. The first class of alternatives was that subsequent coin tosses are still independent of each other, but that the probability of heads differed from 0.5 The second one was that the overall probability of heads may still be 0.5, but that subsequent coin tosses were correlated.

Alternative hypotheses

Question **6.3.3** *Recall the concept of sufficient statistics from Chap**??**. Is the total number of heads a sufficient statistic for the binomial distribution? Why might be it be a good test statistic for our first class of alternatives, but not for the second?*

Question **6.3.4** *Does a test statistic always have to be sufficient?*

So let's remember that we typically have multiple possible choices of test statistic (in principle it could be any numerical summary of the data). Making the right choice is important for getting a test with good power[7]. What the right choice is will depend on what kind of alternatives we expect. This is not always easy to know in advance.

[7] See Section 6.4.

Once we have chosen the test statistic we need to compute its null distribution. You can do this either with pencil and paper or by computer simulations. A pencil and paper solution that leads to a closed form mathematical expression (like Equation (6.1)) has the advantage that it holds for a range of model parameters of the null hypothesis (such as $n, p$). And it can be quickly computed for any specific set of parameters. But it is not always as easy as in the coin tossing example. Sometimes a pencil and paper solution is impossibly difficult to compute. At other times, it may require simplifying assumptions. An example is a null distribution for the $t$-statistic (which we will see later in this chapter). We can compute one if we assume that the data are independent and Normal distributed, the result is called the $t$-distribution. Such modelling assumptions may be more or less realistic. Simulating the null distribution offers a potentially more accurate, more realistic and perhaps even more intuitive approach. The drawback of simulating is that it can take a rather long time, and we have to work extra to get a systematic understanding of how varying parameters influence the result. Generally, it is more elegant to use the parametric theory when it applies[8]. When you are in doubt, simulate – or do both.

Parametric theory versus simulation

[8] The assumptions don't need to be **exactly** true – it is sufficient if the theory's predictions are an acceptable approximation of the truth.

As for the rejection region: how small is small enough? That is your choice of the **significance level** $\alpha$, which is the total probability of the test statistic falling into this region if the null hypothesis is true[9]. Even when $\alpha$ is given, the choice of the rejection region is not unique. A further condition that we require from a good rejection region is that the probability of the test statistic falling into it is as large possible if the null hypothesis is indeed false. In other words, we want our test to have high **power**.

Rejection region

[9] Some people at one point in time for a particular set of questions colluded on $\alpha = 0.05$ as being "small". But there is nothing special about this number.

In Figure 6.5, the rejection region is split between the two tails of the distribution. This is because we anticipate that unfair coins could have a bias either towards head or toward tail; we don't know. If we did know, we could instead concentrate our rejection region all on the appropriate side, e. g., the right tail if we think the bias would be towards head. Such choices are also refered to as **two-sided** and **one-sided** tests.

## 6.4 Types of Error

Having set out the mechanics of testing, we can assess how well we are doing. Table 6.2 compares reality (whether or not the null hypothesis is in fact true) with the decision whether or not to reject it.

Table 6.2: Types of error in a statistical test.

| Test vs reality | **Null hypothesis is true** | **...is false** |
|---|---|---|
| **Reject null hypothesis** | Type I error (false positive) | True positive |
| **Do not reject** | True negative | Type II error (false negative) |

The two types of error we can make are in the lower left and upper right cells of the table. It's always possible to reduce one of the two error types on the cost of increasing the other one. The real challenge is to find an acceptable trade-off between both of them. This is exemplified in Figure 6.6. We can always decrease the **false positive rate** (FPR) by shifting the threshold to the right. We can become more "conservative". But this happens at the price of higher **false negative rate** (FNR). Analogously, we can decrease the FNR by shifting the threshold to the left. But then again, this happens at the price of higher FPR. A bit on terminology: the FPR is the same as the probability $\alpha$ that we mentioned above. $1-\alpha$ is also called the **specificity** of a test. The FNR is sometimes also called $\beta$, and $1-\beta$ the **power**, **sensitivity** or **true positive rate** of a test.

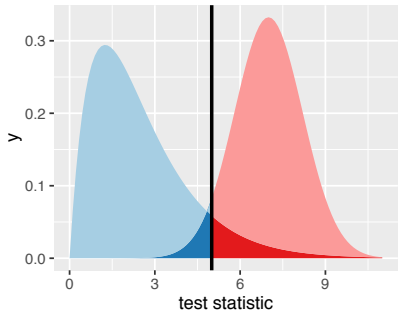


Figure 6.6: The trade-off between type I and II errors. The densities represent the distributions of a hypothetical test statistic either under the null or the alternative. The peak on the left (light and dark blue plus dark red) represents the test statistic's distribution under the null. It integrates to 1. Suppose the decision boundary is the black line and the hypothesis is rejected if the statistic falls to the left. The probability of a false positive (the FPR) is then simply the dark red area. Similarly, if the peak on the right (light and dark red plus dark blue area) is the test statistic's distribution under the alternative, the probability of a false negative (the FNR) is the dark blue area.

Question **6.4.1**

*At the end of Section 6.3 we learned about one- and two-sided tests. Why does this distinction exist – why don't we alway just use the two-sided test, which is sensitive to a larger class of alternatives?*

## 6.5 The t-test

Many experimental measurements are reported as real numbers, and the simplest comparison we can make is between two groups, say, cells treated with a substance compared to cells that are not. The basic test for such situations is the $t$-test. The test statistic is defined as

$$t = c\,\frac{m_1 - m_2}{s}, \tag{6.2}$$

where $m_1$ and $m_2$ are the mean of the values in the two groups, $s$ is the pooled standard deviation and $c$ is a constant that depends on the sample sizes, i. e., the numbers

of samples $n_1$ and $n_2$ in the two groups. To be totally explicit,

$$\begin{aligned} m_g &= \frac{1}{n_g}\sum_{i=1}^{n_g} x_{g,i} \qquad g = 1, 2 \\ s^2 &= \frac{1}{n_1+n_2-2}\left(\sum_{i=1}^{n_1}\left(x_{1,i}-m_1\right)^2 + \sum_{j=1}^{n_2}\left(x_{2,j}-m_2\right)^2\right) \\ c &= \sqrt{\frac{n_1 n_2}{n_1+n_2}}. \end{aligned} \tag{6.3}$$

where $x_{g,i}$ is the $i^{\text{th}}$ data point in the $g^{\text{th}}$ group. Let's try this out with the `PlantGrowth` data from R's *datasets* package.

```
data("PlantGrowth")
ggplot(PlantGrowth, aes(y = weight, x = group, col = group)) +
  geom_jitter(height = 0, width = 0.4) +
  theme(legend.position = "none")
tt <- with(PlantGrowth,
   t.test(weight[group =="ctrl"],
          weight[group =="trt2"],
          var.equal = TRUE))
tt
##
##  Two Sample t-test
##
## data:  weight[group == "ctrl"] and weight[group == "trt2"]
## t = -2.134, df = 18, p-value = 0.04685
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.980338117 -0.007661883
## sample estimates:
## mean of x mean of y
##     5.032     5.526
```

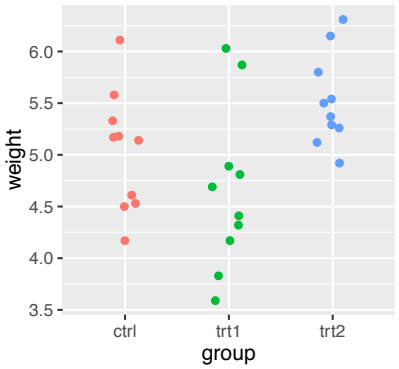


Figure 6.7: The `PlantGrowth` data.

Question **6.5.1** *What do you get from the comparison with* `trt1`*? What for* `trt1` *versus* `trt2`*?*

Question **6.5.2** *What is the significance of the* `var.equal = TRUE` *in the above call to* `t.test`*?*

Question **6.5.3** *Rewrite the above call to* `t.test` *using the formula interface, i.e., by using the notation* `weight ~ group`*.*

To compute the p-value, the `t.test` function uses the asymptotic theory for the $t$-statistic (6.2); this theory states that under the null hypothesis of equal means in both groups, this quantity follows a known, mathematical distribution, the so-called $t$-distribution with $n_1 + n_2$ degrees of freedom. The theory uses additional technical assumptions, namely that the data are independent and come from a Normal distribution with the same standard deviation. We could be worried about these assumptions. Clearly they do not hold: weights are always positive, while the Normal distribution extends over the whole real axis. The question is whether this deviation from the theoretical assumption makes a real difference. We can use sample permutations to figure this out.

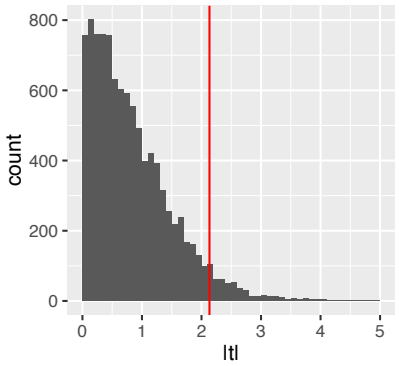


Figure 6.8: The null distribution of the (absolute) $t$-statistic determined by simulations – namely, by random permutations of the group labels.

```
abs_t_null <- with(
  filter(PlantGrowth, group %in% c("ctrl", "trt2")),
    replicate(10000,
      abs(t.test(weight ~ sample(group))$statistic)))

ggplot(data_frame(`|t|` = abs_t_null), aes(x = `|t|`)) +
  geom_histogram(binwidth = 0.1, boundary = 0) +
  geom_vline(xintercept = abs(tt$statistic), col="red")
mean(abs(tt$statistic) <= abs_t_null)
## [1] 0.0471
```

Question **6.5.4** *Why did we use the absolute value function (`abs`) in the above code?*

Question **6.5.5** *Plot the (parametric) t-distribution with the appropriate degrees of freedom?*

Different flavors of *t*-test

The *t*-test comes in multiple flavors, all of which can be chosen through parameters of the `t.test` function. What we did above was a two-sided two-sample unpaired test with equal variance. **Two-sided** refers to the fact that we were open to reject the null hypothesis if the weight of the treated plants was either larger or smaller than that of the untreated ones. **Two-sample** indicates that we compared the means of two groups to each other; another option would be to compare the mean of one group against a given, fixed number. **Unpaired** means that there was no direct 1:1 mapping between the measurements in the two groups. If, on the other hand, the data had been measured on the same plants before and other treatment, then a paired test would be more appropriate, as it looks at the change of weight within each plant, rather than their absolute weights. **Equal variance** refers to the way the statistic (6.2) is calculated. That expression is most appropriate if the variances within each group are about the same. If they are much different, an alternative form[10] and associated asymptotic theory exist.

[10] Welch's *t*-test

The independence assumption

Now let's try something peculiar: duplicate the data.

```
with(rbind(PlantGrowth, PlantGrowth),
   t.test(weight[group =="ctrl"],
          weight[group =="trt2"],
          var.equal = TRUE))
##
##  Two Sample t-test
##
## data:  weight[group == "ctrl"] and weight[group == "trt2"]
## t = -3.1007, df = 38, p-value = 0.003629
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.8165284 -0.1714716
## sample estimates:
## mean of x mean of y
##     5.032     5.526
```

Note how the estimates of the group means (and thus, of the difference) are unchanged, but the p-value is now much smaller! We can conclude two things from this:

- The power of the $t$-test depends on the sample size. Even if the underlying biological differences are the same, a dataset with more samples tends to give more significant results[11].
- The assumption of independence between the measurements is really important. Blatant duplication of the same data is an extreme form of dependence, but to some extent the same thing happens if you mix up different levels of replication. For instance, suppose you had data from 8 plants, but measured the same thing twice on each plant (technical replicates), then pretending that these are now 16 independent measurements to a downstream analysis, such as the $t$-test, is wrong.

[11] You can already see this from Equation 6.3.

## 6.6 P-value Hacking

Let's go back to the coin tossing example. We could not reject the null hypothesis (that the coin is fair) at a level of 5% – even though we "knew" that it is unfair. After all, `probHead` was 0.6 on page 5. Let's suppose we now start looking at different test statistics. Perhaps the number of consecutive series of 3 or more heads. Or the number of heads in the first 50 coin flips. And so on. At some point we will find a test that happens to result in a small p-value, even if just by chance (after all, the probability for the p-value to be less than 5% under the null is 0.05, not an infinitesimally small number). We just did what is called **p-value hacking**[12] [13]. You see what the problem is: in our zeal to prove our point we tortured the data until some statistic did what we wanted. A related tactic is **hypothesis switching** or **HARKing** – hypothesizing after the results are known: we have a dataset, maybe we have invested a lot of time and money into assembling it, so we need results. We come up with lots of different null hypotheses, test them, and iterate, until we can report something interesting.

All these tactics are not according to the rule book, as described in Section 6.3, with a linear and non-iterative sequence of choosing the hypothesis and the test, and then seeing the data. But, of course, they are often more close to reality. With biological data, we tend to have so many different choices for "normalising" the data, transforming the data, add corrections for apparent batch effects, removing outliers, .... The topic is complex and open-ended. Wasserstein and Lazar (2016) give a very readable short summary of the problems with how p-values are used in science, and of some of the misconceptions. They also highlight how p-values can be fruitfully used. The essential message is: be completely transparent about your data, what analyses were tried, and how they were done. Provide the analysis code. Only with such contextual information can a p-value be useful.

[12] `http://fivethirtyeight.com/features/science-isnt-broken`

[13] Megan L Head, Luke Holman, Rob Lanfear, Andrew T Kahn, and Michael D Jennions. The extent and consequences of p-hacking in science. **PLoS Biol**, 13(3):e1002106, 2015

**Avoid fallacy**. Keep in mind that our statistical test is never attempting to prove our null hypothesis is true - we are simply saying whether or not there is evidence for it to be false. If a high p-value **were** indicative of the truth of the null hypothesis, we could formulate a completely crazy null hypothesis, do an utterly irrelevant experiment, collect a small amount of inconclusive data, find a p-value that would just be a random number between 0 and 1 (and so with some high probability above our threshold $\alpha$) and, whoosh, our hypothesis would be demonstrated!

## 6.7 Multiple Testing

Question **6.7.1** *Look up xkcd comic 882. Why didn't the newspaper report the results for the other colors?*

The same quandary occurs with high-throughput data in biology. And with force! You will be dealing not only with 20 colors of jellybeans, but, say, with 20,000 genes

that were tested for differential expression between two conditions, or with 3 billion positions in the genome where a DNA mutation might have happened. So how do we deal with this? Let's look again at our table relating statistical test results with reality (Table 6.2), this time framing everything in terms of many null hypotheses.

Table 6.4: Types of error in multiple testing. The letters designate the number of times each type of error occurs.

| Test vs Reality | **Null Hypothesis is true** | **...is false** | **Total** |
|---|---|---|---|
| **Rejected** | V | S | $R$ |
| **Not rejected** | U | T | $m - R$ |
| **Total** | $m_0$ | $m - m_0$ | $m$ |

- $m$: total number of hypotheses
- $m_0$: number of null hypotheses
- $V$: number of false positives (a measure of type I error)
- $T$: number of false negatives (a measure of type II error)
- $S, U$: number of true positives and true negatives
- $R$: number of rejections

## 6.8 The Family Wise Error Rate

The **family wise error rate** (FWER) is the probability that $V > 0$, i. e., that we make one or more false positive errors. We can compute it as the complement of making no false positive errors at all[14].

[14] Assuming independence.

$$1 - P(\text{no rejection of any of } m_0 \text{ nulls}) = 1 - (1-\alpha)^{m_0} \to 1 \quad \text{as } m \to \infty \tag{6.4}$$

For any fixed $\alpha$, this probability is appreciable as soon as $m$ is in the order of $1/\alpha$, and tends towards 1 as $m$ becomes larger. This relationship can have big consequences for experiments like DNA matching, where a large database of potential matches is searched. For example, if there is a one in a million chance that the DNA profiles of two people match by random error, and your DNA is tested against a database of 800000 profiles, then the probability of a random hit with the database (i. e., without you being in it) is:

```
1 - (1 - 1/1e6)^8e5
## [1] 0.5506712
```

That's pretty high. And once the database contains a few million profiles, a false hit is virtually unavoidable.

Question **6.8.1** *Prove that the probability* (6.4) *does indeed become very close to 1 when m is large.*

### 6.8.1 Bonferroni correction

How are we to choose the per-hypothesis $\alpha$ if we want FWER control? The above computations give us an intuition that the product of $\alpha$ with $m$ gives us a ballpark

estimate, and this guess is in fact true. The Bonferroni correction is simply that if we want FWER control at level $\alpha_{\text{FWER}}$, we should choose the per hypothesis threshold $\alpha = \alpha_{\text{FWER}}/m$. Let's check this out on an example.

```
m <- 10000

ggplot(data_frame(
  alpha = seq(0, 7e-6, length.out = 100),
  p     = 1 - (1 - alpha)^m),
  aes(x = alpha, y = p)) +  geom_line() +
  xlab(expression(alpha)) +
  ylab("Prob( no false rejection )") +
  geom_hline(yintercept = 0.05, col="red")
```

In Figure 6.9, the black line intersects the red line (which corresponds to a value of 0.05) at $\alpha = 5.13 \times 10^{-6}$, which is just a little bit more than the value of $0.05/m$ implied by the Bonferroni correction.

Question **6.8.2** *Why are the two values not exactly the same?*

A potential drawback of this method, however, is that when $m$ is large, the rejection threshold is very small. This means that the individual tests need to be very powerful if we want to have any chance to detect something. Often this is not possible, or would not be an effective use of our time and money. We'll see that there are more nuanced methods of controlling our type I error.

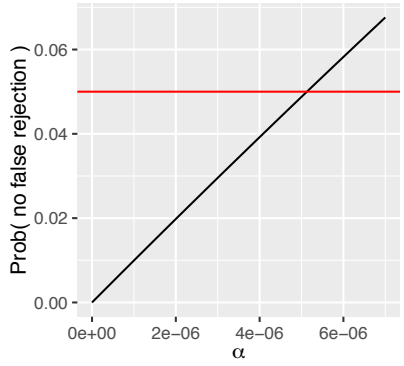


Figure 6.9: Bonferroni correction. The plot shows the graph of (6.4) for $m = 10^4$ as a function of $\alpha$.

## 6.9 The False Discovery Rate

Let's look at some real data. We load up the RNA-Seq dataset `airway`, which contains gene expression measurements (gene-level counts) of four primary human airway smooth muscle cell lines with and without treatment with dexamethasone, a synthetic glucocorticoid. We'll use the *DESeq2* method that we'll discuss in more detail in Chapter **??**. For now it suffices to say that it performs a test for differential expression for each gene. Conceptually, the tested null hypothesis is very similar to that of the $t$-test, although the test statistic and the null distribution are slightly more involved since we are dealing with count data.

```
library("DESeq2")
library("airway")
data("airway")
aw <- DESeqDataSet(se = airway, design = ~ cell + dex)
aw <- aw[ rowMeans(counts(aw)) > 1, ]
dim(aw)
## [1] 22724     8
counts(aw)[1:2, 1:3]
##                 SRR1039508 SRR1039509 SRR1039512
## ENSG00000000003        679        448        873
## ENSG00000000419        467        515        621
colData(aw)[, 2:4]
## DataFrame with 8 rows and 3 columns
##                cell      dex    albut
##            <factor> <factor> <factor>
## SRR1039508   N61311    untrt    untrt
```

```
## SRR1039509   N61311      trt    untrt
## SRR1039512  N052611    untrt    untrt
## SRR1039513  N052611      trt    untrt
## SRR1039516  N080611    untrt    untrt
## SRR1039517  N080611      trt    untrt
## SRR1039520  N061011    untrt    untrt
## SRR1039521  N061011      trt    untrt
awfit <- DESeq(aw)
awde  <- as.data.frame(results(awfit))
```

Question **6.9.1** *Why did we (in the 5$^{th}$ line of the above code chunk) remove genes that have a very small number of counts on average across all samples?*

Question **6.9.2** *Have a look at the content of* **awde**.

Question **6.9.3** *(Optional) Consult the DESeq2 vignette and/or Chapter* **??** *for more information on what the above code chunk does.*

### 6.9.1 The p-value histogram

Let's plot the histogram of p-values.

```
ggplot(awde, aes(x = pvalue)) +
  geom_histogram(binwidth = 0.025, boundary = 0)
```

The histogram (Figure 6.10) is an important sanity check for any analysis that involves multiple tests. We expected it to be composed of two components:

- A uniform background, which corresponds to the null hypotheses. Remember that under the null, the p-value is distributed uniformly in $[0, 1]$.
- A peak at the left, from small p-values that were emitted by the alternatives.

The relative size of these two components depends on the fraction of true nulls and true alternatives in the data. The shape of the peak towards the left depends on the power of the tests: if the experiment was underpowered, we can still expect that the p-values from the alternatives tend towards being small, but some of them will scatter up into the middle of the range.

Suppose we reject all tests with a p-value less than $\alpha$. We could visually determine an estimate of null hypotheses among these with a plot like in Figure 6.11

```
alpha <- binw <- 0.025
pi0 <- 2 * mean(awde$pvalue > 0.5)
ggplot(awde,
  aes(x = pvalue)) + geom_histogram(binwidth = binw, boundary = 0) +
  geom_hline(yintercept = pi0 * binw * nrow(awde), col = "blue") +
  geom_vline(xintercept = alpha, col = "red")
```

We see that there are 4783 p-values in the first bin ($[0, \alpha]$), among which we expect around 439 to be nulls (as indicated by the blue line). Thus we can estimate the fraction of false rejections as

```
pi0 * alpha / mean(awde$pvalue <= alpha)
## [1] 0.09168932
```

Coming back to our terminology of Table 6.4, the **false discovery rate** (FDR) is

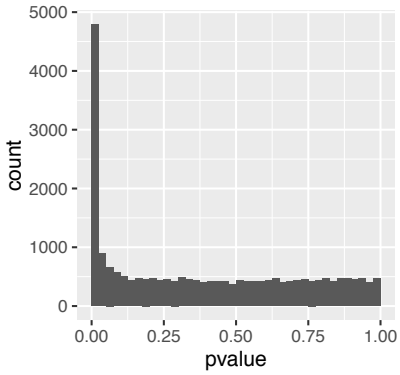


Figure 6.10: p-value histogram of for the `airway` data.

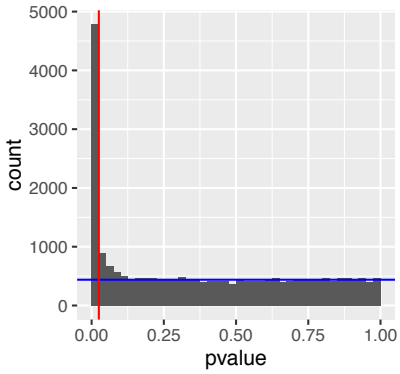


Figure 6.11: Visual estimation of the FDR with the p-value histogram.

defined as

$$\mathrm{FDR} = \mathrm{E}\left[\frac{V}{\max(R,1)}\right], \tag{6.5}$$

The expression in the denominator makes sure that the maths are well-defined even when $R = 0$[15]. E[ ] stands for the **expectation value**. That means that the FDR is not a quantity associated with a specific outcome of $V$ and $R$ for one particular experiment. Rather, given our choice of tests and associated rejection rules for them, it is the average[16] proportion of type I errors out of the rejections made, where the average is taken (at least conceptually) over many replicate instances of the experiment.

[15] ...and thus by implication $V = 0$.

[16] Since the FDR is an expectation value, it does not provide worst case control: in any single experiment, the so-called false discovery proportion (FDP), that is $V/R$ without the E[ ], could be much higher (or lower). Just as knowing the mean of a population does not tell you the values of the extremes.

### 6.9.2 The Benjamini-Hochberg algorithm for controlling the FDR

There is a more elegant alternative to the "visual FDR" method of the last section. The procedure, introduced by Y. Benjamini and Y. Hochberg[17] has these steps:

- First, order the p-values in increasing order, $p_1 \ldots p_m$
- Then for some choice of $\varphi$ (our target FDR), find the largest value of $k$ that satisfies: $p_k \leq \varphi\, k/m$
- Finally reject the hypotheses $1 \ldots k$

We can see how this procedure works when applied to our RNA-seq p-values through a simple graphical illustration:

```
phi  <- 0.10
awde <- mutate(awde, rank = rank(pvalue))
m    <- nrow(awde)

ggplot(filter(awde, rank <= 7000), aes(x = rank, y = pvalue)) +
  geom_line() + geom_abline(slope = phi / m, col="red")
```

The method now simply finds the rightmost point where the black (our p-values) and red lines (slope $\varphi/m$) intersect. Then it rejects all tests to the left.

```
kmax <- with(arrange(awde, rank),
         last(which(pvalue <= phi * rank / m)))
kmax
## [1] 4563
```

Question **6.9.4** *Compare the value of* `kmax` *with the number of 4783 from above (Figure 6.11). Why are they different?*

Question **6.9.5** *Look at the code associated with the option* `method="BH"` *of the* `p.adjust` *function that comes with R. Compare it to what we did above.*

[17] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. **Journal of the Royal Statistical Society B**, 57:289–300, 1995

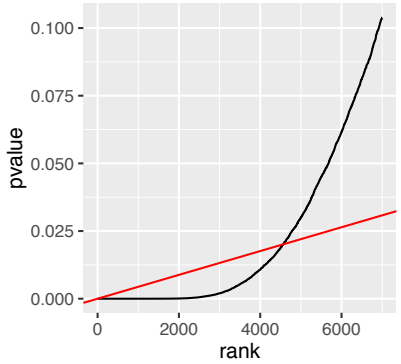


Figure 6.12: Visualisation of the Benjamini-Hochberg procedure. Shown is a zoom-in to the 7000 lowest p-values.

## 6.10 The Local FDR

While the xkcd comic mentioned in Figure 6.1 ends with a rather sinister intepretation of the multiple testing problem as a way to accumulate errors, Figure 6.13 highlights the multiple testing opportunity: when we do many tests, we can use the data to increase our understanding beyond what's possible with a single test.

Let's get back to the histogram in Figure 6.11. Conceptually, we can think of it in terms of the two-groups model [18]:

[18] Bradley Efron. **Large-scale inference: empirical Bayes methods for estimation, testing, and prediction**, volume 1. Cambridge University Press, 2010

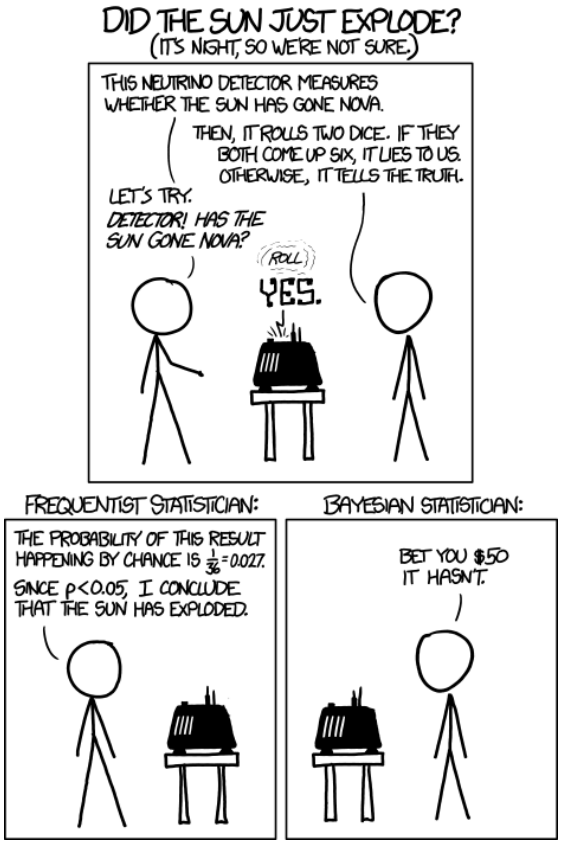


Figure 6.13: From http://xkcd.com/1132 – While the frequentist only has the currently available data, the Bayesian can draw on mechanistic insight or on previous experience. As a Bayesian, she would know enough about physics to understand that our sun's mass is too small to become a nova. And if she does not know physics, she might be an **empirical Bayesian**, and draw her prior from countless previous days where the sun did not go nova.

$$f(p) = \pi_0 + (1 - \pi_0) f_{\text{alt}}(p), \tag{6.6}$$

Here, $f(p)$ is the density of the distribution (what the histogram would look like with infinitely much data and infinitely small bins), $\pi_0$ is a number between 0 and 1 that represents the size of the uniform component, and $f_{\text{alt}}$ is the alternative component. These functions are visualised in the upper panel of Figure 6.14: the blue areas together correspond to the graph of $f_{\text{alt}}(p)$, the grey areas to that of $f_{\text{null}}(p) = \pi_0$. If we now consider one particular cutoff $p$ (say, $p = 0.1$ as in Figure 6.14), then we can decompose the value of $f$ at the cutoff (red line) into the contribution from the nulls (light red, $\pi_0$) and from the alternatives (darker red, $(1 - \pi_0) f_{\text{alt}}(p)$). So we have the **local false disovery rate**

$$\text{fdr}(p) = \frac{\pi_0}{f(p)}, \tag{6.7}$$

and this quantity, which by definition is between 0 and 1, tells us the probability that a hypothesis which we rejected at some cutoff $p$ would be a false positive. Note how the fdr in Figure 6.14 is a monotonically decreasing function of $p$, and this goes with our intuition that the fdr should be lowest for the smallest $p$ and then gradually get larger, until it reaches 1 at the very right end. We can make a similar decomposition not only for the red line, but also for the area under the curve. This is

$$F(p) = \int_0^p f(t)\, dt, \tag{6.8}$$

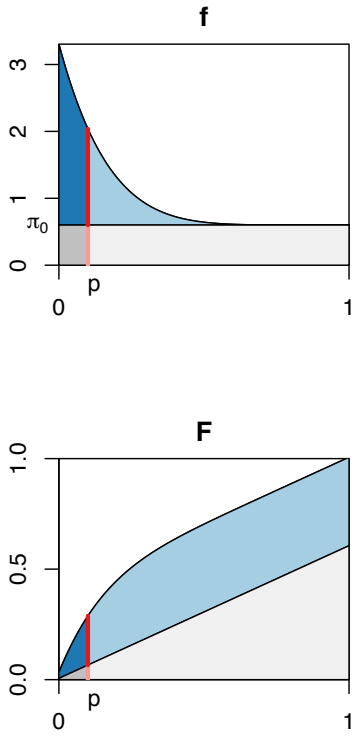


Figure 6.14: Local false discovery rate and the two-group model, with some choice of $f_{\text{alt}}(p)$, and $\pi_0 = 0.6$. Top: densities, bottom: distribution functions.

and the ratio of the dark grey area (that is, $\pi_0$ times $p$) to that is the **tail area false disovery rate** (Fdr[19]).

$$\mathrm{Fdr}(p) = \frac{\pi_0\, p}{F(p)}, \tag{6.9}$$

We'll use the data version of $F$ for diagnostics in Figure 6.18.

The packages *qvalue* and *fdrtool* offer facilities to fit these models to data.

```
library("fdrtool")
ft <- fdrtool(awde$pvalue, statistic = "pvalue")
```

In *fdrtool*, what we called $\pi_0$ above is called `eta0`:

```
ft$param[,"eta0"]
##      eta0
## 0.7605948
```

Question **6.10.1** *What do the plots show that are produced by the above call to* `fdrtool`*?*

Question **6.10.2** *Explore the other elements of the list* `ft`*.*

Question **6.10.3** *What does the* ***empirical*** *in empirical Bayes methods stand for?*

### 6.10.1 Local versus total

The FDR (or the Fdr) is a set property - it is a single number that applies to a whole set of rejections made in the course of a multiple testing analysis. In contrast, the fdr is a local property - it applies to individual additional hypothesis. Recall Figure 6.14, where the fdr was computed for each point along the $x$-axis of the density plot, whereas the Fdr depends on the areas to the left of the red line.

Question **6.10.4** *Check out the concepts of* ***total cost*** *and* ***marginal cost*** *in economics. Can you seen an analogy with Fdr and fdr?*

## 6.11 Independent Filtering and Hypothesis Weighting

The Benjamini-Hochberg method and the two-groups model, as we have seen them so far, implicitly assume **exchangeability** of the hypotheses: all we use are the p-values. Beyond these, we do not take into account any additional information. This is not always optimal.

Let's look at an example. Intuitively, the signal-to-noise ratio for genes with larger numbers of reads mapped to them should be better than for genes with few reads, and that should affect the power of our tests. We look at the mean of normalized counts across samples. In the *DESeq2* software this quantity is called the `baseMean`.

```
awde$baseMean[1]
## [1] 708.6022
cts <- counts(awfit, normalized = TRUE)[1, ]
cts
## SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516 SRR1039517
##   663.3142   499.9070   740.1528   608.9063   966.3137   748.3722
## SRR1039520 SRR1039521
##   836.2487   605.6024
mean(cts)
## [1] 708.6022
```

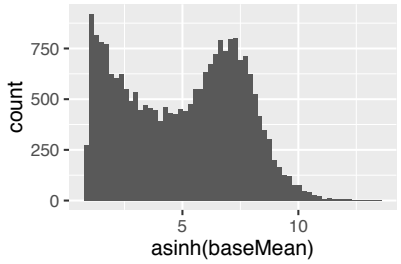


Figure 6.15: Histogram of `baseMean`. We see that it covers a large dynamic range, from close to 0 to around $3.3 \times 10^5$.

[19] The convention is to use the lower case abbreviation fdr for the local, and the abbreviation Fdr for the tail-area false discovery rate in the context of the two-groups model (6.6). The abbreviation FDR is used for the original definition (6.5), which is a bit more general.

Next we produce its histogram across genes, and a scatterplot between it and the p-values.

```
ggplot(awde, aes(x = asinh(baseMean))) +
  geom_histogram(bins = 60)
```

```
ggplot(awde, aes(x = rank(baseMean), y = -log10(pvalue))) +
  geom_hex(bins = 60) +
  theme(legend.position = "none")
```

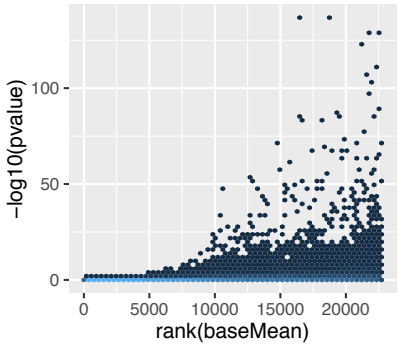


Figure 6.16: Scatterplot of the rank of `baseMean` versus the negative logarithm of the p-value. For small values of `baseMean`, no small p-values occur. Only for genes whose read counts across all samples have a certain size, the test for differential expression has power to come out with a small p-value.

Question **6.11.1** *Why did we use the asinh transformation for the histogram? How does it look like with no transformation, the logarithm, the shifted logarithm, i. e., $\log(x + const.)$?*

Question **6.11.2** *In the scatterplot, why did we use $-\log_{10}$ for the p-values? Why the rank transformation for the `baseMean`?*

For convenience, we discretize `baseMean` into a factor variable `group`, which corresponds to six equal-sized groups.

```
awde <- mutate(awde, stratum = cut(baseMean,
   breaks = quantile(baseMean, probs =
              seq(0, 1, length.out = 7)),
   include.lowest = TRUE))
```

In Figures 6.17 and 6.18 we see the histograms of p-values and the ECDFs stratified by `stratum`.

```
ggplot(awde, aes(x = pvalue)) +
  geom_histogram(binwidth = 0.025, boundary = 0) +
  facet_wrap( ~ stratum, nrow = 4)
```

```
ggplot(awde, aes(x = pvalue, col = stratum)) +
  stat_ecdf(geom = "step")
```

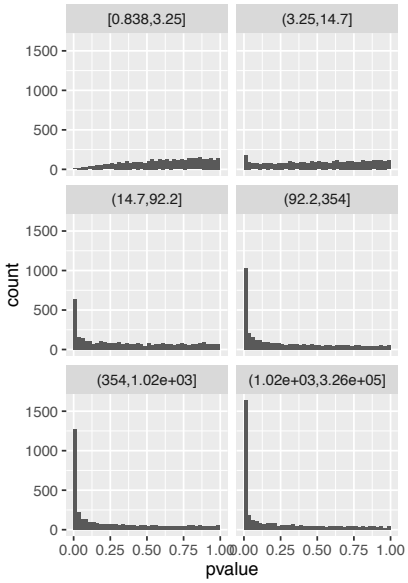


Figure 6.17: p-value histograms of the airway data, stratified into 6equally sized groups defined by increasing value of `baseMean`.

If we were to fit the two-group model to these strata separately, we would get quite different parameters (i. e., $\pi_0$, $f_{\text{alt}}$). For the most lowly expressed genes (those in the first `baseMean`-bin), the power of the *DESeq2*-test is low, and the p-values essentially all come from the null component. As we go higher in average expression, the height of the small-p-values peak in the histograms increases, reflecting the increasing power of the test.

Can we use that for a better multiple testing correction? It turns out that this is possible. We can use either **independent filtering** [20] or **independent hypothesis weighting** (IHW) [21].

```
library("IHW")
ihw_res <- ihw(awde$pvalue, awde$baseMean, alpha = 0.1)
rejections(ihw_res)
## [1] 4915
```

Let's compare this to what we get from the ordinary (unweighted) Benjamini-Hochberg method:

```
padj_BH <- p.adjust(awde$pvalue, method = "BH")
sum(padj_BH < 0.1)
## [1] 4563
```

With hypothesis weighting, we get more rejections. For these data, the difference

[20] Richard Bourgon, Robert Gentleman, and Wolfgang Huber. Independent filtering increases detection power for high-throughput experiments. **PNAS**, 107(21):9546–9551, 2010. URL http://www.pnas.org/content/107/21/9546.long

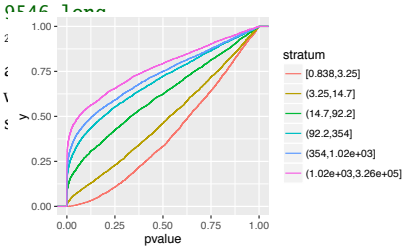


Figure 6.18: Same data as in Figure 6.17, shown with ECDFs.

is notable though not spectacular, this is because their signal-to-noise is already quite high. In other situations (e. g., when there are fewer replicates or they are more noisy, or when the effect of the treatment is less drastic), the difference from using IHW can be more pronounced.

We can have a look at the weights determined by the `ihw` function.

```
plot(ihw_res)
```

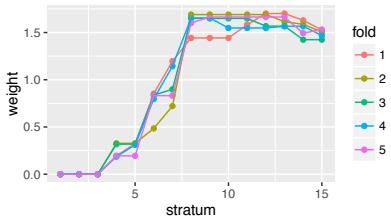


Figure 6.19: Hypothesis weights determined by the `ihw` function. Here the function's default settings chose 15 strata, while in our manual exploration above (Figures 6.17, 6.18) we had used 6; in practice, this is a minor detail.

Intuitively, what happens here is that IHW chooses to put more weight on the hypothesis strata with higher `baseMean`, and low weight on those with very low counts. The Benjamini-Hochberg method has a certain type-I error budget, and rather than spreading it equally among all hypotheses, here we take it away from those strata that have little change of small fdr anyway, and "invest" it in strata where many hypotheses can be rejected at small fdr.

Question **6.11.3** *Why does Figure 6.19 show 5 curves, rather than only one?*

Such possibilities for stratification by a covariate (in our case: `baseMean`) exist in many multiple testing situations. Informally, we need the covariate to be

- statistically independent from our p-values under the null, but
- informative of the prior probability $\pi_0$ and/or the power of the test (the shape of the alternative density, $f_{\text{alt}}$) in the two-groups model.

These requirements can be assessed through diagnostic plots as in Figures 6.15–6.18.

## 6.12 Summary of this Chapter

To summarize what we hope you've learned from this chapter:

- Understand the principal steps of a hypothesis test.
- Know the different types of errors we are about to commit when doing hypothesis testing.
- Understand the challenges and opportunities of doing thousands or millions of tests.
- Know your different between the family wise error rate and the false discovery rate.
- Be familiar with the false discovery rate, and understand the difference between its local and total (tail-area) definitions.
- Understand that often not all hypotheses are exchangeable, and that taking into account informative covariates can improve your analyses.
- Be familiar with diagnostic plots, and know to always look at the p-value histogram when encountering a multiple testing analysis.

## 6.13 Exercises

Exercise **6.1** *What is a data type or an analysis method from your scientific field of expertise that relies on multiple testing? Do you focus on FWER or FDR? Are the hypotheses all exchangeable, or are there any informative covariates?*

Exercise **6.2** *Why do statisticians often focus so much on the null hypothesis of a test, compared to the alternative hypothesis?*

Exercise **6.3** *How can we ever prove that the null hypothesis is true? Or that the alternative is true?*

Exercise **6.4** *Make a less extreme example of correlated test statistics than the data duplication at the end of Section 6.5. Simulate data with true null hypotheses only, so that the data morph from being completely independent to totally correlated as a function of some continuous-valued control parameter. Check type-I error control (e. g., with the p-value histogram) as a function of this control parameter.*

Exercise **6.5** *Find an example in the published literature that looks like p-value hacking, outcome switching, HARKing played a role.*

Exercise **6.6** *What other type-I and type-II error concepts are there for multiple testing?*

Exercise **6.7** *The FDR is an expectation value, i. e., aims to control average behavior of a procedure. Are there methods for worst case control?*

## 6.14 Further Reading

- A comprehensive text book treatment of multiple testing is given by [22].
- Outcome switching in clinical trials: `http://compare-trials.org`
- For hypothesis weighting, the *IHW* vignette, the IHW paper [23] and the references therein.

[22] Bradley Efron. **Large-scale inference: empirical Bayes methods for estimation, testing, and prediction**, volume 1. Cambridge University Press, 2010

[23] Nikolaos Ignatiadis, Bernd Klaus, Judith Zaugg, and Wolfgang Huber. Data-driven hypothesis weighting increases detection power in genome-scale multiple testing. **Nature Methods**, 2016

# Bibliography


Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. **Journal of the Royal Statistical Society B**, 57:289–300, 1995.

Richard Bourgon, Robert Gentleman, and Wolfgang Huber. Independent filtering increases detection power for high-throughput experiments. **PNAS**, 107(21):9546–9551, 2010. URL http://www.pnas.org/content/107/21/9546.long.

Bradley Efron. **Large-scale inference: empirical Bayes methods for estimation, testing, and prediction**, volume 1. Cambridge University Press, 2010.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. **The Elements of Statistical Learning**. Springer, 2008.

Megan L Head, Luke Holman, Rob Lanfear, Andrew T Kahn, and Michael D Jennions. The extent and consequences of p-hacking in science. **PLoS Biol**, 13(3):e1002106, 2015.

Nikolaos Ignatiadis, Bernd Klaus, Judith Zaugg, and Wolfgang Huber. Data-driven hypothesis weighting increases detection power in genome-scale multiple testing. **Nature Methods**, 2016.

Ronald L Wasserstein and Nicole A Lazar. The asa's statement on p-values: context, process, and purpose. **The American Statistician**, 2016.