13
Multiple Testing

Thus far, this textbook has mostly focused on *estimation* and its close cousin, *prediction*. In this chapter, we instead focus on hypothesis testing, which is key to conducting *inference*. We remind the reader that inference was briefly discussed in Chapter 2.

While Section 13.1 provides a brief review of null hypotheses, p-values, test statistics, and other key ideas in hypothesis testing, this chapter assumes that the reader has had previous exposure to these topics. In particular, we will not focus on *why* or *how* to conduct a hypothesis test — a topic on which entire books can be (and have been) written! Instead, we will assume that the reader is interested in testing some particular set of null hypotheses, and has a specific plan in mind for how to conduct the tests and obtain p-values.

Much of the emphasis in classical statistics focuses on testing a single null hypothesis, such as H_0*: the expected blood pressure of mice in the control group equals the expected blood pressure of mice in the treatment group*. Of course, we would probably like to discover that there *is* a difference between the mean blood pressure in the two groups. But for reasons that will become clear, we construct a null hypothesis corresponding to no difference.

In contemporary settings, we are often faced with huge amounts of data, and consequently may wish to test a great many null hypotheses. For instance, rather than simply testing H_0, we might want to test m null hypotheses, $H_{01}, \ldots, H_{0m}$, where H_{0j}*: the expected value of the* j^{th} *biomarker among mice in the control group equals the expected value of the* j^{th} *biomarker among mice in the treatment group*. When conducting *multiple testing*, we need to be very careful about how we interpret the results, in order to avoid erroneously rejecting far too many null hypotheses.

This chapter discusses classical as well as more contemporary ways to conduct multiple testing in a big-data setting. In Section 13.2, we highlight the challenges associated with multiple testing. Classical solutions to these

G. James et al., *An Introduction to Statistical Learning*, Springer Texts in Statistics,
https://doi.org/10.1007/978-3-031-38747-0_13

challenges are presented in Section 13.3, and more contemporary solutions in Sections 13.4 and 13.5.

In particular, Section 13.4 focuses on the false discovery rate. The notion of the false discovery rate dates back to the 1990s. It quickly rose in popularity in the early 2000s, when large-scale data sets began to come out of genomics. These datasets were unique not only because of their large size,[1] but also because they were typically collected for *exploratory* purposes: researchers collected these datasets in order to test a huge number of null hypotheses, rather than just a very small number of pre-specified null hypotheses. Today, of course, huge datasets are collected without a pre-specified null hypothesis across virtually all fields. As we will see, the false discovery rate is perfectly-suited for this modern-day reality.

This chapter naturally centers upon the classical statistical technique of p-values, used to quantify the results of hypothesis tests. At the time of writing of this book (2020), p-values have recently been the topic of extensive commentary in the social science research community, to the extent that some social science journals have gone so far as to ban the use of p-values altogether! We will simply comment that when properly understood and applied, p-values provide a powerful tool for drawing inferential conclusions from our data.

13.1 A Quick Review of Hypothesis Testing

Hypothesis tests provide a rigorous statistical framework for answering simple "yes-or-no" questions about data, such as the following:

1. Is the true coefficient β_j in a linear regression of Y onto $X_1, \ldots, X_p$ equal to zero?[2]

2. Is there a difference in the expected blood pressure of laboratory mice in the control group and laboratory mice in the treatment group?[3]

In Section 13.1.1, we briefly review the steps involved in hypothesis testing. Section 13.1.2 discusses the different types of mistakes, or errors, that can occur in hypothesis testing.

13.1.1 Testing a Hypothesis

Conducting a hypothesis test typically proceeds in four steps. First, we define the null and alternative hypotheses. Next, we construct a test statistic that summarizes the strength of evidence against the null hypothesis. We then compute a p-value that quantifies the probability of having obtained

[1] Microarray data was viewed as "big data" at the time, although by today's standards, this label seems quaint: a microarray dataset can be (and typically was) stored in a Microsoft Excel spreadsheet!

[2] This hypothesis test was discussed on page 76 of Chapter 3.

[3] The "treatment group" refers to the set of mice that receive an experimental treatment, and the "control group" refers to those that do not.

a comparable or more extreme value of the test statistic under the null hypothesis. Finally, based on the p-value, we decide whether to reject the null hypothesis. We now briefly discuss each of these steps in turn.

Step 1: Define the Null and Alternative Hypotheses

In hypothesis testing, we divide the world into two possibilities: the *null hypothesis* and the *alternative hypothesis*. The null hypothesis, denoted H_0, is the default state of belief about the world.[4] For instance, null hypotheses associated with the two questions posed earlier in this chapter are as follows:

null hypothesis
alternative hypothesis

1. The true coefficient β_j in a linear regression of Y onto $X_1, \ldots, X_p$ equals zero.

2. There is no difference between the expected blood pressure of mice in the control and treatment groups.

The null hypothesis is boring by construction: it may well be true, but we might hope that our data will tell us otherwise.

The alternative hypothesis, denoted H_a, represents something different and unexpected: for instance, that there *is* a difference between the expected blood pressure of the mice in the two groups. Typically, the alternative hypothesis simply posits that the null hypothesis does not hold: if the null hypothesis states that *there is no difference between A and B*, then the alternative hypothesis states that *there is a difference between A and B*.

It is important to note that the treatment of H_0 and H_a is asymmetric. H_0 is treated as the default state of the world, and we focus on using data to reject H_0. If we reject H_0, then this provides evidence in favor of H_a. We can think of rejecting H_0 as making a *discovery* about our data: namely, we are discovering that H_0 does not hold! By contrast, if we fail to reject H_0, then our findings are more nebulous: we will not know whether we failed to reject H_0 because our sample size was too small (in which case testing H_0 again on a larger or higher-quality dataset might lead to rejection), or whether we failed to reject H_0 because H_0 really holds.

Step 2: Construct the Test Statistic

Next, we wish to use our data in order to find evidence for or against the null hypothesis. In order to do this, we must compute a *test statistic*, denoted T, which summarizes the extent to which our data are consistent with H_0. The way in which we construct T depends on the nature of the null hypothesis that we are testing.

test statistic

To make things concrete, let $x_1^t, \ldots, x_{n_t}^t$ denote the blood pressure measurements for the n_t mice in the treatment group, and let $x_1^c, \ldots, x_{n_c}^c$ denote the blood pressure measurements for the n_c mice in the control group, and $\mu_t = \mathrm{E}(X^t)$, $\mu_c = \mathrm{E}(X^c)$. To test $H_0 : \mu_t = \mu_c$, we make use of a *two-sample t-statistic*,[5] defined as

two-sample t-statistic

[4] H_0 is pronounced "H naught" or "H zero".

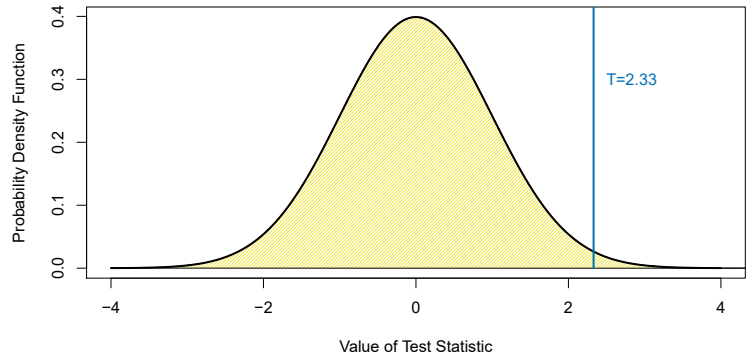

FIGURE 13.1. *The density function for the $N(0,1)$ distribution, with the vertical line indicating a value of 2.33. 1% of the area under the curve falls to the right of the vertical line, so there is only a 2% chance of observing a $N(0,1)$ value that is greater than 2.33 or less than -2.33. Therefore, if a test statistic has a $N(0,1)$ null distribution, then an observed test statistic of $T = 2.33$ leads to a p-value of 0.02.*

$$T = \frac{\hat{\mu}_t - \hat{\mu}_c}{s\sqrt{\frac{1}{n_t} + \frac{1}{n_c}}} \tag{13.1}$$

where $\hat{\mu}_t = \frac{1}{n_t}\sum_{i=1}^{n_t} x_i^t$, $\hat{\mu}_c = \frac{1}{n_c}\sum_{i=1}^{n_c} x_i^c$, and

$$s = \sqrt{\frac{(n_t - 1)s_t^2 + (n_c - 1)s_c^2}{n_t + n_c - 2}} \tag{13.2}$$

is an estimator of the pooled standard deviation of the two samples.[6] Here, s_t^2 and s_c^2 are unbiased estimators of the variance of the blood pressure in the treatment and control groups, respectively. A large (absolute) value of T provides evidence against $H_0 : \mu_t = \mu_c$, and hence evidence in support of $H_a : \mu_t \neq \mu_c$.

Step 3: Compute the *p*-Value

In the previous section, we noted that a large (absolute) value of a two-sample t-statistic provides evidence against H_0. This begs the question: *how large is large?* In other words, how much evidence against H_0 is provided by a given value of the test statistic?

The notion of a *p-value* provides us with a way to formalize as well as answer this question. The p-value is defined as the probability of observing a test statistic equal to or more extreme than the observed statistic, *under the assumption that H_0 is in fact true*. Therefore, a small p-value provides evidence *against* H_0.

p-value

[5]The t-statistic derives its name from the fact that, under H_0, it follows a t-distribution.

[6]Note that (13.2) assumes that the control and treatment groups have equal variance. Without this assumption, (13.2) would take a slightly different form.

To make this concrete, suppose that $T = 2.33$ for the test statistic in (13.1). Then, we can ask: what is the probability of having observed such a large value of T, if indeed H_0 holds? It turns out that under H_0, the distribution of T in (13.1) follows approximately a $N(0, 1)$ distribution[7] — that is, a normal distribution with mean 0 and variance 1. This distribution is displayed in Figure 13.1. We see that the vast majority — 98% — of the $N(0, 1)$ distribution falls between -2.33 and 2.33. This means that under H_0, we would expect to see such a large value of $|T|$ only 2% of the time. Therefore, the p-value corresponding to $T = 2.33$ is 0.02.

The distribution of the test statistic under H_0 (also known as the test statistic's *null distribution*) will depend on the details of what type of null hypothesis is being tested, and what type of test statistic is used. In general, most commonly-used test statistics follow a well-known statistical distribution under the null hypothesis — such as a normal distribution, a t-distribution, a χ^2-distribution, or an F-distribution — provided that the sample size is sufficiently large and that some other assumptions hold. Typically, the R function that is used to compute a test statistic will make use of this null distribution in order to output a p-value. In Section 13.5, we will see an approach to estimate the null distribution of a test statistic using re-sampling; in many contemporary settings, this is a very attractive option, as it exploits the availability of fast computers in order to avoid having to make potentially problematic assumptions about the data.

null distribution

The p-value is perhaps one of the most used and abused notions in all of statistics. In particular, it is sometimes said that the p-value is the probability that H_0 holds, i.e., that the null hypothesis is true. This is not correct! The one and only correct interpretation of the p-value is as the fraction of the time that we would expect to see such an extreme value of the test statistic[8] if we repeated the experiment many many times, *provided* H_0 *holds*.

In Step 2 we computed a test statistic, and noted that a large (absolute) value of the test statistic provides evidence against H_0. In Step 3 the test statistic was converted to a p-value, with small p-values providing evidence against H_0. What, then, did we accomplish by converting the test statistic from Step 2 into a p-value in Step 3? To answer this question, suppose a data analyst conducts a statistical test, and reports a test statistic of $T = 17.3$. Does this provide strong evidence against H_0? It's impossible to know, without more information: in particular, we would need to know

[7] More precisely, assuming that the observations are drawn from a normal distribution, then T follows a t-distribution with $n_t + n_c - 2$ degrees of freedom. Provided that $n_t + n_c - 2$ is larger than around 40, this is very well-approximated by a $N(0, 1)$ distribution. In Section 13.5, we will see an alternative and often more attractive way to approximate the null distribution of T, which avoids making stringent assumptions about the data.

[8] A *one-sided* p-value is the probability of seeing such an extreme value of the test statistic; e.g. the probability of seeing a test statistic greater than or equal to $T = 2.33$. A *two-sided* p-value is the probability of seeing such an extreme value of the *absolute* test statistic; e.g. the probability of seeing a test statistic greater than or equal to 2.33 or less than or equal to -2.33. The default recommendation is to report a two-sided p-value rather than a one-sided p-value, unless there is a clear and compelling reason that only one direction of the test statistic is of scientific interest.

		Truth	
		H_0	H_a
Decision	Reject H_0	Type I Error	Correct
	Do Not Reject H_0	Correct	Type II Error

TABLE 13.1. *A summary of the possible scenarios associated with testing the null hypothesis H_0. Type I errors are also known as false positives, and Type II errors as false negatives.*

what value of the test statistic should be expected, under H_0. This is exactly what a p-value gives us. In other words, a p-value allows us to transform our test statistic, which is measured on some arbitrary and uninterpretable scale, into a number between 0 and 1 that can be more easily interpreted.

Step 4: Decide Whether to Reject the Null Hypothesis

Once we have computed a p-value corresponding to H_0, it remains for us to decide whether or not to reject H_0. (We do not usually talk about "accepting" H_0: instead, we talk about "failing to reject" H_0.) A small p-value indicates that such a large value of the test statistic is unlikely to occur under H_0, and thereby provides evidence against H_0. If the p-value is sufficiently small, then we will want to reject H_0 (and, therefore, make a "discovery"). But how small is small enough to reject H_0?

It turns out that the answer to this question is very much in the eyes of the beholder, or more specifically, the data analyst. The smaller the p-value, the stronger the evidence against H_0. In some fields, it is typical to reject H_0 if the p-value is below 0.05; this means that, if H_0 holds, we would expect to see such a small p-value no more than 5% of the time.[9] However, in other fields, a much higher burden of proof is required: for example, in some areas of physics, it is typical to reject H_0 only if the p-value is below 10^{-9}!

In the example displayed in Figure 13.1, if we use a threshold of 0.05 as our cut-off for rejecting the null hypothesis, then we will reject the null. By contrast, if we use a threshold of 0.01, then we will fail to reject the null. These ideas are formalized in the next section.

13.1.2 Type I and Type II Errors

If the null hypothesis holds, then we say that it is a *true null hypothesis*; otherwise, it is a *false null hypothesis*. For instance, if we test $H_0 : \mu_t = \mu_c$ as in Section 13.1.1, and there is indeed no difference in the *population* mean blood pressure for mice in the treatment group and mice in the control group, then H_0 is true; otherwise, it is false. Of course, we do not know *a priori* whether H_0 is true or whether it is false: this is why we need to conduct a hypothesis test!

true null hypothesis

false null hypothesis

[9] Though a threshold of 0.05 to reject H_0 is ubiquitous in some areas of science, we advise against blind adherence to this arbitrary choice. Furthermore, a data analyst should typically report the p-value itself, rather than just whether or not it exceeds a specified threshold value.

Table 13.1 summarizes the possible scenarios associated with testing the null hypothesis H_0.[10] Once the hypothesis test is performed, the *row* of the table is known (based on whether or not we have rejected H_0); however, it is impossible for us to know which *column* we are in. If we reject H_0 when H_0 is false (i.e., when H_a is true), or if we do not reject H_0 when it is true, then we arrived at the correct result. However, if we erroneously reject H_0 when H_0 is in fact true, then we have committed a *Type I error*. The *Type I error rate* is defined as the probability of making a Type I error given that H_0 holds, i.e., the probability of incorrectly rejecting H_0. Alternatively, if we do not reject H_0 when H_0 is in fact false, then we have committed a *Type II error*. The *power* of the hypothesis test is defined as the probability of not making a Type II error given that H_a holds, i.e., the probability of correctly rejecting H_0.

Type I error
Type I error rate
Type II error
power

Ideally we would like both the Type I and Type II error rates to be small. But in practice, this is hard to achieve! There typically is a trade-off: we can make the Type I error small by only rejecting H_0 if we are quite sure that it doesn't hold; however, this will result in an increase in the Type II error. Alternatively, we can make the Type II error small by rejecting H_0 in the presence of even modest evidence that it does not hold, but this will cause the Type I error to be large. In practice, we typically view Type I errors as more "serious" than Type II errors, because the former involves declaring a scientific finding that is not correct. Hence, when we perform hypothesis testing, we typically require a low Type I error rate — e.g., at most $\alpha = 0.05$ — while trying to make the Type II error small (or, equivalently, the power large).

It turns out that there is a direct correspondence between the p-value threshold that causes us to reject H_0, and the Type I error rate. By only rejecting H_0 when the p-value is below α, we ensure that the Type I error rate will be less than or equal to α.

13.2 The Challenge of Multiple Testing

In the previous section, we saw that rejecting H_0 if the p-value is below (say) 0.01 provides us with a simple way to control the Type I error for H_0 at level 0.01: if H_0 is true, then there is no more than a 1% probability that we will reject it. But now suppose that we wish to test m null hypotheses, $H_{01}, \ldots, H_{0m}$. Will it do to simply reject all null hypotheses for which the corresponding p-value falls below (say) 0.01? Stated another way, if we reject all null hypotheses for which the p-value falls below 0.01, then how many Type I errors should we expect to make?

As a first step towards answering this question, consider a stockbroker who wishes to drum up new clients by convincing them of her trading

[10] There are parallels between Table 13.1 and Table 4.6, which has to do with the output of a binary classifier. In particular, recall from Table 4.6 that a false positive results from predicting a positive (non-null) label when the true label is in fact negative (null). This is closely related to a Type I error, which results from rejecting the null hypothesis when in fact the null hypothesis holds.

acumen. She tells 1,024 ($1{,}024 = 2^{10}$) potential new clients that she can correctly predict whether Apple's stock price will increase or decrease for 10 days running. There are 2^{10} possibilities for how Apple's stock price might change over the course of these 10 days. Therefore, she emails each client one of these 2^{10} possibilities. The vast majority of her potential clients will find that the stockbroker's predictions are no better than chance (and many will find them to be even worse than chance). But a broken clock is right twice a day, and one of her potential clients will be really impressed to find that her predictions were correct for all 10 of the days! And so the stockbroker gains a new client.

What happened here? Does the stockbroker have any actual insight into whether Apple's stock price will increase or decrease? No. How, then, did she manage to predict Apple's stock price perfectly for 10 days running? The answer is that she made a lot of guesses, and one of them happened to be exactly right.

How does this relate to multiple testing? Suppose that we flip 1,024 fair coins[11] ten times each. Then we would expect (on average) one coin to come up all tails. (There's a $1/2^{10} = 1/1{,}024$ chance that any single coin will come up all tails. So if we flip 1,024 coins, then we expect one coin to come up all tails, on average.) If one of our coins comes up all tails, then we might therefore conclude that this particular coin is not fair. In fact, a standard hypothesis test for the null hypothesis that this particular coin is fair would lead to a p-value below 0.002![12] But it would be incorrect to conclude that the coin is not fair: in fact, the null hypothesis holds, and we just happen to have gotten ten tails in a row by chance.

These examples illustrate the main challenge of *multiple testing*: when testing a huge number of null hypotheses, we are bound to get some very small p-values by chance. If we make a decision about whether to reject each null hypothesis without accounting for the fact that we have performed a very large number of tests, then we may end up rejecting a great number of true null hypotheses — that is, making a large number of Type I errors.

multiple testing

How severe is the problem? Recall from the previous section that if we reject a single null hypothesis, H_0, if its p-value is less than, say, $\alpha = 0.01$, then there is a 1% chance of making a false rejection if H_0 is in fact true. Now what if we test m null hypotheses, $H_{01}, \ldots, H_{0m}$, all of which are true? There's a 1% chance of rejecting any individual null hypothesis; therefore, we expect to falsely reject approximately $0.01 \times m$ null hypotheses. If $m = 10{,}000$, then that means that we expect to falsely reject 100 null hypotheses by chance! That is a *lot* of Type I errors.

The crux of the issue is as follows: rejecting a null hypothesis if the p-value is below α controls the probability of falsely rejecting *that null hypothesis* at level α. However, if we do this for m null hypotheses, then the chance of falsely rejecting *at least one of the m null hypotheses* is quite a bit higher!

---

[11] A *fair coin* is one that has an equal chance of landing heads or tails.

[12] Recall that the p-value is the probability of observing data at least this extreme, under the null hypothesis. If the coin is fair, then the probability of observing at least ten tails is $(1/2)^{10} = 1/1{,}024 < 0.001$. The p-value is therefore $2/1{,}024 < 0.002$, since this is the probability of observing ten heads or ten tails.

	H_0 is True	H_0 is False	Total
Reject H_0	V	S	R
Do Not Reject H_0	U	W	$m-R$
Total	m_0	$m-m_0$	m

TABLE 13.2. *A summary of the results of testing m null hypotheses. A given null hypothesis is either true or false, and a test of that null hypothesis can either reject or fail to reject it. In practice, the individual values of V, S, U, and W are unknown. However, we do have access to $V+S=R$ and $U+W=m-R$, which are the numbers of null hypotheses rejected and not rejected, respectively.*

We will investigate this issue in greater detail, and pose a solution to it, in Section 13.3.

13.3 The Family-Wise Error Rate

In the following sections, we will discuss testing multiple hypotheses while controlling the probability of making at least one Type I error.

13.3.1 What is the Family-Wise Error Rate?

Recall that the Type I error rate is the probability of rejecting H_0 if H_0 is true. The *family-wise error rate* (FWER) generalizes this notion to the setting of m null hypotheses, $H_{01},\ldots,H_{0m}$, and is defined as the probability of making *at least one* Type I error. To state this idea more formally, consider Table 13.2, which summarizes the possible outcomes when performing m hypothesis tests. Here, V represents the number of Type I errors (also known as false positives or false discoveries), S the number of true positives, U the number of true negatives, and W the number of Type II errors (also known as false negatives). Then the family-wise error rate is given by

family-wise error rate

$$\text{FWER} = \Pr(V \geq 1). \tag{13.3}$$

A strategy of rejecting any null hypothesis for which the p-value is below α (i.e. controlling the Type I error for each null hypothesis at level α) leads to a FWER of

$$\begin{aligned}\text{FWER}(\alpha) &= 1-\Pr(V=0)\\ &= 1-\Pr(\text{do not falsely reject any null hypotheses})\\ &= 1-\Pr\left(\textstyle\bigcap_{j=1}^{m}\{\text{do not falsely reject } H_{0j}\}\right).\end{aligned} \tag{13.4}$$

Recall from basic probability that if two events A and B are independent, then $\Pr(A\cap B)=\Pr(A)\Pr(B)$. Therefore, if we make the additional rather strong assumptions that the m tests are independent and that all m null hypotheses are true, then

$$\text{FWER}(\alpha) = 1-\prod_{j=1}^{m}(1-\alpha) = 1-(1-\alpha)^m. \tag{13.5}$$

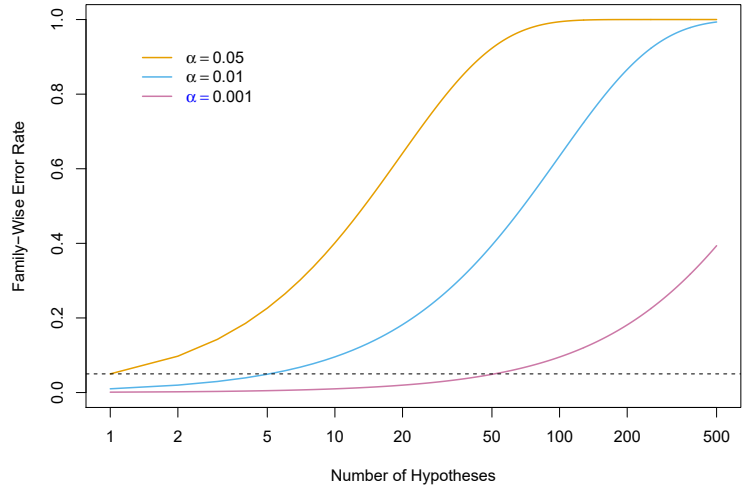

FIGURE 13.2. *The family-wise error rate, as a function of the number of hypotheses tested (displayed on the log scale), for three values of* α*:* $\alpha = 0.05$ *(orange),* $\alpha = 0.01$ *(blue), and* $\alpha = 0.001$ *(purple). The dashed line indicates* 0.05*. For example, in order to control the FWER at* 0.05 *when testing* $m = 50$ *null hypotheses, we must control the Type I error for each null hypothesis at level* $\alpha = 0.001$*.*

Hence, if we test only one null hypothesis, then $\text{FWER}(\alpha) = 1-(1-\alpha)^1 = \alpha$, so the Type I error rate and the FWER are equal. However, if we perform $m = 100$ independent tests, then $\text{FWER}(\alpha) = 1-(1-\alpha)^{100}$. For instance, taking $\alpha = 0.05$ leads to a FWER of $1-(1-0.05)^{100} = 0.994$. In other words, we are virtually guaranteed to make at least one Type I error!

Figure 13.2 displays (13.5) for various values of m, the number of hypotheses, and α, the Type I error. We see that setting $\alpha = 0.05$ results in a high FWER even for moderate m. With $\alpha = 0.01$, we can test no more than five null hypotheses before the FWER exceeds 0.05. Only for very small values, such as $\alpha = 0.001$, do we manage to ensure a small FWER, at least for moderately-sized m.

We now briefly return to the example in Section 13.1.1, in which we consider testing a single null hypothesis of the form $H_0 : \mu_t = \mu_c$ using a two-sample t-statistic. Recall from Figure 13.1 that in order to guarantee that the Type I error does not exceed 0.02, we decide whether or not to reject H_0 using a cutpoint of 2.33 (i.e. we reject H_0 if $|T| \geq 2.33$). Now, what if we wish to test 10 null hypotheses using two-sample t-statistics, instead of just one? We will see in Section 13.3.2 that we can guarantee that the FWER does not exceed 0.02 by rejecting only null hypotheses for which the p-value falls below 0.002. This corresponds to a much more stringent cutpoint of 3.09 (i.e. we should reject H_{0j} only if its test statistic $|T_j| \geq 3.09$, for $j = 1, \ldots, 10$). In other words, controlling the FWER at level α amounts to a much higher bar, in terms of evidence required to reject any given null hypothesis, than simply controlling the Type I error for each null hypothesis at level α.

Manager	Mean, $\bar{x}$	Standard Deviation, s	t-statistic	p-value
One	3.0	7.4	2.86	0.006
Two	-0.1	6.9	-0.10	0.918
Three	2.8	7.5	2.62	0.012
Four	0.5	6.7	0.53	0.601
Five	0.3	6.8	0.31	0.756

TABLE 13.3. *The first two columns correspond to the sample mean and sample standard deviation of the percentage excess return, over $n = 50$ months, for the first five managers in the* `Fund` *dataset. The last two columns provide the t-statistic $(\sqrt{n} \cdot \bar{X}/S)$ and associated p-value for testing $H_{0j} : \mu_j = 0$, the null hypothesis that the (population) mean return for the jth hedge fund manager equals zero.*

13.3.2 Approaches to Control the Family-Wise Error Rate

In this section, we briefly survey some approaches to control the FWER. We will illustrate these approaches on the `Fund` dataset, which records the monthly percentage excess returns for 2,000 fund managers over $n = 50$ months.[13] Table 13.3 provides relevant summary statistics for the first five managers.

We first present the Bonferroni method and Holm's step-down procedure, which are very general-purpose approaches for controlling the FWER that can be applied whenever m p-values have been computed, regardless of the form of the null hypotheses, the choice of test statistics, or the (in)dependence of the p-values. We then briefly discuss Tukey's method and Scheffé's method in order to illustrate the fact that, in certain situations, more specialized approaches for controlling the FWER may be preferable.

The Bonferroni Method

As in the previous section, suppose we wish to test $H_{01}, \ldots, H_{0m}$. Let A_j denote the event that we make a Type I error for the jth null hypothesis, for $j = 1, \ldots, m$. Then

$$\begin{aligned} \text{FWER} &= \Pr(\text{falsely reject at least one null hypothesis}) \\ &= \Pr(\cup_{j=1}^{m} A_j) \\ &\leq \sum_{j=1}^{m} \Pr(A_j). \end{aligned} \tag{13.6}$$

In (13.6), the inequality results from the fact that for any two events A and B, $\Pr(A \cup B) \leq \Pr(A) + \Pr(B)$, regardless of whether A and B are independent. The *Bonferroni method*, or *Bonferroni correction*, sets the threshold for rejecting each hypothesis test to α/m, so that $\Pr(A_j) \leq \alpha/m$. Equation 13.6 implies that

$$\text{FWER}(\alpha/m) \leq m \times \frac{\alpha}{m} = \alpha,$$

[13] Excess returns correspond to the additional return the fund manager achieves beyond the market's overall return. So if the market increases by 5% during a given period and the fund manager achieves a 7% return, their *excess return* would be $7\% - 5\% = 2\%$.

so this procedure controls the FWER at level α. For instance, in order to control the FWER at level 0.1 while testing $m = 100$ null hypotheses, the Bonferroni procedure requires us to control the Type I error for each null hypothesis at level $0.1/100 = 0.001$, i.e. to reject all null hypotheses for which the p-value is below 0.001.

We now consider the `Fund` dataset in Table 13.3. If we control the Type I error at level $\alpha = 0.05$ for each fund manager separately, then we will conclude that the first and third managers have significantly non-zero excess returns; in other words, we will reject $H_{01} : \mu_1 = 0$ and $H_{03} : \mu_3 = 0$. However, as discussed in previous sections, this procedure does not account for the fact that we have tested multiple hypotheses, and therefore it will lead to a FWER greater than 0.05. If we instead wish to control the FWER at level 0.05, then, using a Bonferroni correction, we must control the Type I error for each individual manager at level $\alpha/m = 0.05/5 = 0.01$. Consequently, we will reject the null hypothesis only for the first manager, since the p-values for all other managers exceed 0.01. The Bonferroni correction gives us peace of mind that we have not falsely rejected too many null hypotheses, but for a price: we reject few null hypotheses, and thus will typically make quite a few Type II errors.

The Bonferroni correction is by far the best-known and most commonly-used multiplicity correction in all of statistics. Its ubiquity is due in large part to the fact that it is very easy to understand and simple to implement, and also from the fact that it successfully controls Type I error regardless of whether the m hypothesis tests are independent. However, as we will see, it is typically neither the most powerful nor the best approach for multiple testing correction. In particular, the Bonferroni correction can be quite conservative, in the sense that the true FWER is often quite a bit lower than the nominal (or target) FWER; this results from the inequality in (13.6). By contrast, a less conservative procedure might allow us to control the FWER while rejecting more null hypotheses, and therefore making fewer Type II errors.

Holm's Step-Down Procedure

Holm's method, also known as Holm's step-down procedure or the Holm–Bonferroni method, is an alternative to the Bonferroni procedure. Holm's method controls the FWER, but it is less conservative than Bonferroni, in the sense that it will reject more null hypotheses, typically resulting in fewer Type II errors and hence greater power. The procedure is summarized in Algorithm 13.1. The proof that this method controls the FWER is similar to, but slightly more complicated than, the argument in (13.6) that the Bonferroni method controls the FWER. It is worth noting that in Holm's procedure, the threshold that we use to reject each null hypothesis — $p_{(L)}$ in Step 5 — actually depends on the values of *all* m of the p-values. (See the definition of L in (13.7).) This is in contrast to the Bonferroni procedure, in which to control the FWER at level α, we reject any null hypotheses for which the p-value is below α/m, regardless of the other p-values. Holm's method makes no independence assumptions about the m hypothesis tests, and is uniformly more powerful than the Bonferroni method — it will

Holm's method

Algorithm 13.1 *Holm's Step-Down Procedure to Control the FWER*

1. Specify α, the level at which to control the FWER.
2. Compute p-values, $p_1, \ldots, p_m$, for the m null hypotheses $H_{01}, \ldots, H_{0m}$.
3. Order the m p-values so that $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$.
4. Define
$$L = \min \left\{ j : p_{(j)} > \frac{\alpha}{m+1-j} \right\}. \tag{13.7}$$
5. Reject all null hypotheses H_{0j} for which $p_j < p_{(L)}$.

always reject at least as many null hypotheses as Bonferroni — and so it should always be preferred.

We now consider applying Holm's method to the first five fund managers in the `Fund` dataset in Table 13.3, while controlling the FWER at level 0.05. The ordered p-values are $p_{(1)} = 0.006, p_{(2)} = 0.012, p_{(3)} = 0.601, p_{(4)} = 0.756$ and $p_{(5)} = 0.918$. The Holm procedure rejects the first two null hypotheses, because $p_{(1)} = 0.006 < 0.05/(5+1-1) = 0.01$ and $p_{(2)} = 0.012 < 0.05/(5+1-2) = 0.0125$, but $p_{(3)} = 0.601 > 0.05/(5+1-3) = 0.0167$, which implies that $L = 3$. We note that, in this setting, Holm is more powerful than Bonferroni: the former rejects the null hypotheses for the first and third managers, whereas the latter rejects the null hypothesis only for the first manager.

Figure 13.3 provides an illustration of the Bonferroni and Holm methods on three simulated data sets in a setting involving $m = 10$ hypothesis tests, of which $m_0 = 2$ of the null hypotheses are true. Each panel displays the ten corresponding p-values, ordered from smallest to largest, and plotted on a log scale. The eight red points represent the false null hypotheses, and the two black points represent the true null hypotheses. We wish to control the FWER at level 0.05. The Bonferroni procedure requires us to reject all null hypotheses for which the p-value is below 0.005; this is represented by the black horizontal line. The Holm procedure requires us to reject all null hypotheses that fall below the blue line. The blue line always lies above the black line, so Holm will always reject more tests than Bonferroni; the region between the two lines corresponds to the hypotheses that are only rejected by Holm. In the left-hand panel, both Bonferroni and Holm successfully reject seven of the eight false null hypotheses. In the center panel, Holm successfully rejects all eight of the false null hypotheses, while Bonferroni fails to reject one. In the right-hand panel, Bonferroni only rejects three of the false null hypotheses, while Holm rejects all eight. Neither Bonferroni nor Holm makes any Type I errors in these examples.

Two Special Cases: Tukey's Method and Scheffé's Method

Bonferroni's method and Holm's method can be used in virtually any setting in which we wish to control the FWER for m null hypotheses: they

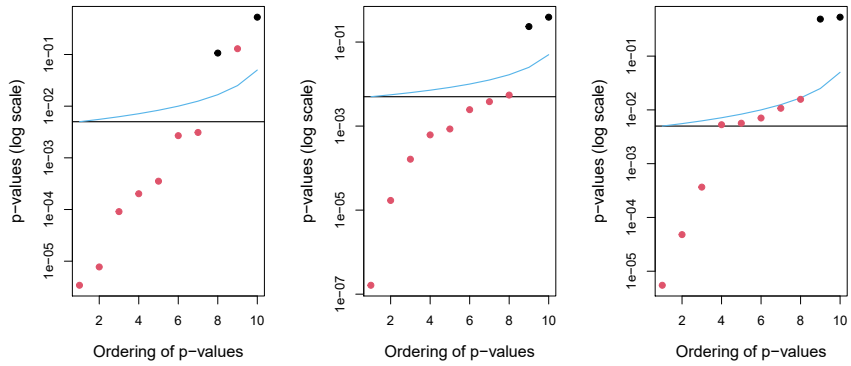

FIGURE 13.3. *Each panel displays, for a separate simulation, the sorted p-values for tests of* $m = 10$ *null hypotheses. The p-values corresponding to the* $m_0 = 2$ *true null hypotheses are displayed in black, and the rest are in red. When controlling the FWER at level* 0.05*, the Bonferroni procedure rejects all null hypotheses that fall below the black line, and the Holm procedure rejects all null hypotheses that fall below the blue line. The region between the blue and black lines indicates null hypotheses that are rejected using the Holm procedure but not using the Bonferroni procedure. In the center panel, the Holm procedure rejects one more null hypothesis than the Bonferroni procedure. In the right-hand panel, it rejects five more null hypotheses.*

make no assumptions about the nature of the null hypotheses, the type of test statistic used, or the (in)dependence of the p-values. However, in certain very specific settings, we can achieve higher power by controlling the FWER using approaches that are more tailored to the task at hand. Tukey's method and Scheffé's method provide two such examples.

Table 13.3 indicates that for the `Fund` dataset, Managers One and Two have the greatest difference in their sample mean returns. This finding might motivate us to test the null hypothesis $H_0 : \mu_1 = \mu_2$, where μ_j is the (population) mean return for the jth fund manager. A two-sample t-test (13.1) for H_0 yields a p-value of 0.0349, suggesting modest evidence against H_0. However, this p-value is misleading, since we decided to compare the average returns of Managers One and Two only after having examined the returns for all five managers; this essentially amounts to having performed $m = 5 \times (5 - 1)/2 = 10$ hypothesis tests, and selecting the one with the smallest p-value. This suggests that in order to control the FWER at level 0.05, we should make a Bonferroni correction for $m = 10$ hypothesis tests, and therefore should only reject a null hypothesis for which the p-value is below 0.005. If we do this, then we will be unable to reject the null hypothesis that Managers One and Two have identical performance.

However, in this setting, a Bonferroni correction is actually a bit too stringent, since it fails to consider the fact that the $m = 10$ hypothesis tests are all somewhat related: for instance, Managers Two and Five have similar mean returns, as do Managers Two and Four; this guarantees that the mean returns of Managers Four and Five are similar. Stated another way, the m p-values for the m pairwise comparisons are *not* independent. Therefore, it should be possible to control the FWER in a way that is

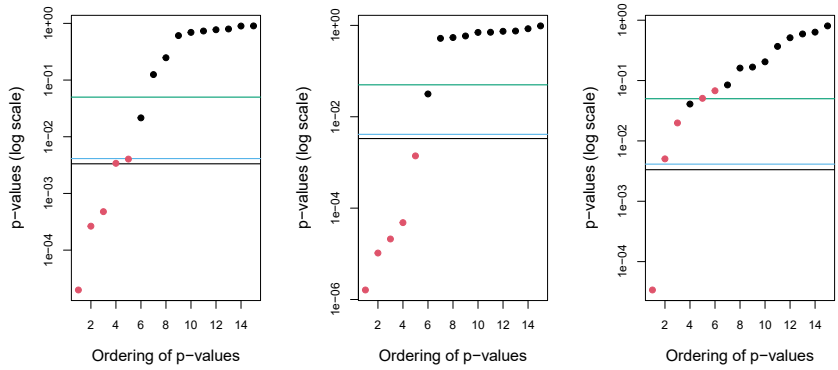

FIGURE 13.4. *Each panel displays, for a separate simulation, the sorted p-values for tests of* $m = 15$ *hypotheses, corresponding to pairwise tests for the equality of* $G = 6$ *means. The* $m_0 = 10$ *true null hypotheses are displayed in black, and the rest are in red. When controlling the FWER at level* 0.05*, the Bonferroni procedure rejects all null hypotheses that fall below the black line, whereas Tukey rejects all those that fall below the blue line. Thus, Tukey's method has slightly higher power than Bonferroni's method. Controlling the Type I error* without *adjusting for multiple testing involves rejecting all those that fall below the green line.*

less conservative. This is exactly the idea behind *Tukey's method*: when performing $m = G(G-1)/2$ pairwise comparisons of G means, it allows us to control the FWER at level α while rejecting all null hypotheses for which the p-value falls below α_T, for some $\alpha_T > \alpha/m$.

Tukey's method

Figure 13.4 illustrates Tukey's method on three simulated data sets in a setting with $G = 6$ means, with $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 \neq \mu_6$. Therefore, of the $m = G(G-1)/2 = 15$ null hypotheses of the form $H_0 : \mu_j = \mu_k$, ten are true and five are false. In each panel, the true null hypotheses are displayed in black, and the false ones are in red. The horizontal lines indicate that Tukey's method always results in at least as many rejections as Bonferroni's method. In the left-hand panel, Tukey correctly rejects two more null hypotheses than Bonferroni.

Now, suppose that we once again examine the data in Table 13.3, and notice that Managers One and Three have higher mean returns than Managers Two, Four, and Five. This might motivate us to test the null hypothesis

$$H_0 : \frac{1}{2}(\mu_1 + \mu_3) = \frac{1}{3}(\mu_2 + \mu_4 + \mu_5). \tag{13.8}$$

(Recall that μ_j is the population mean return for the jth hedge fund manager.) It turns out that we could test (13.8) using a variant of the two-sample t-test presented in (13.1), leading to a p-value of 0.004. This suggests strong evidence of a difference between Managers One and Three compared to Managers Two, Four, and Five. However, there is a problem: we decided to test the null hypothesis in (13.8) only after peeking at the data in Table 13.3. In a sense, this means that we have conducted multiple testing. In this setting, using Bonferroni to control the FWER at level α

would require a p-value threshold of α/m, for an extremely large value of m[14].

Scheffé's method is designed for exactly this setting. It allows us to compute a value α_S such that rejecting the null hypothesis H_0 in (13.8) if the p-value is below α_S will control the Type I error at level α. It turns out that for the `Fund` example, in order to control the Type I error at level $\alpha = 0.05$, we must set $\alpha_S = 0.002$. Therefore, we are unable to reject H_0 in (13.8), despite the apparently very small p-value of 0.004. An important advantage of Scheffé's method is that we can use this same threshold of $\alpha_S = 0.002$ in order to perform a pairwise comparison of any split of the managers into two groups: for instance, we could also test $H_0 : \frac{1}{3}(\mu_1 + \mu_2 + \mu_3) = \frac{1}{2}(\mu_4 + \mu_5)$ and $H_0 : \frac{1}{4}(\mu_1 + \mu_2 + \mu_3 + \mu_4) = \mu_5$ using the same threshold of 0.002, without needing to further adjust for multiple testing.

Scheffé's method

To summarize, Holm's procedure and Bonferroni's procedure are very general approaches for multiple testing correction that can be applied under all circumstances. However, in certain special cases, more powerful procedures for multiple testing correction may be available, in order to control the FWER while achieving higher power (i.e. committing fewer Type II errors) than would be possible using Holm or Bonferroni. In this section, we have illustrated two such examples.

13.3.3 Trade-Off Between the FWER and Power

In general, there is a trade-off between the FWER threshold that we choose, and our *power* to reject the null hypotheses. Recall that power is defined as the number of false null hypotheses that we reject divided by the total number of false null hypotheses, i.e. $S/(m - m_0)$ using the notation of Table 13.2. Figure 13.5 illustrates the results of a simulation setting involving m null hypotheses, of which 90% are true and the remaining 10% are false; power is displayed as a function of the FWER. In this particular simulation setting, when $m = 10$, a FWER of 0.05 corresponds to power of approximately 60%. However, as m increases, the power decreases. With $m = 500$, the power is below 0.2 at a FWER of 0.05, so that we successfully reject only 20% of the false null hypotheses.

Figure 13.5 indicates that it is reasonable to control the FWER when m takes on a small value, like 5 or 10. However, for $m = 100$ or $m = 1{,}000$, attempting to control the FWER will make it almost impossible to reject any of the false null hypotheses. In other words, the power will be extremely low.

Why is this the case? Recall that, using the notation in Table 13.2, the FWER is defined as $\Pr(V \geq 1)$ (13.3). In other other words, controlling the FWER at level α guarantees that the data analyst is *very unlikely* (with probability no more than α) to reject *any* true null hypotheses, i.e. to have any false positives. In order to make good on this guarantee when m is large, the data analyst may be forced to reject very few null hypotheses, or perhaps even none at all (since if $R = 0$ then also $V = 0$; see Table 13.2).

[14] In fact, calculating the "correct" value of m is quite technical, and outside the scope of this book.

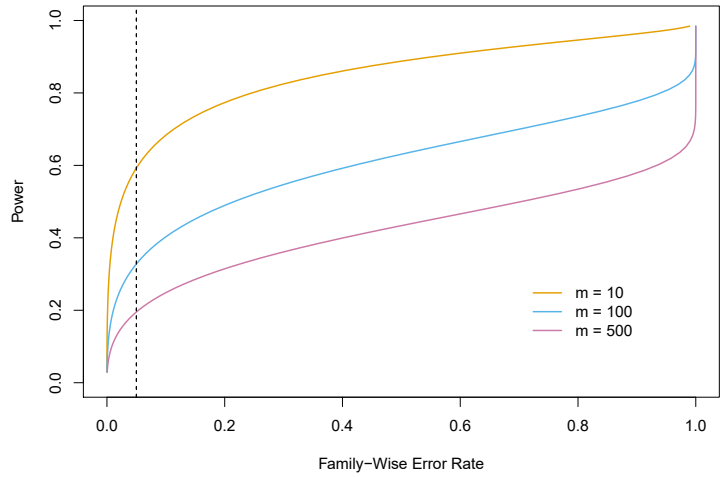

FIGURE 13.5. *In a simulation setting in which 90% of the m null hypotheses are true, we display the power (the fraction of false null hypotheses that we successfully reject) as a function of the family-wise error rate. The curves correspond to $m = 10$ (orange), $m = 100$ (blue), and $m = 500$ (purple). As the value of m increases, the power decreases. The vertical dashed line indicates a FWER of* 0.05.

This is scientifically uninteresting, and typically results in very low power, as in Figure 13.5.

In practice, when m is large, we may be willing to tolerate a few false positives, in the interest of making more discoveries, i.e. more rejections of the null hypothesis. This is the motivation behind the false discovery rate, which we present next.

13.4 The False Discovery Rate

13.4.1 Intuition for the False Discovery Rate

As we just discussed, when m is large, then trying to prevent *any* false positives (as in FWER control) is simply too stringent. Instead, we might try to make sure that the ratio of false positives (V) to total positives ($V + S = R$) is sufficiently low, so that most of the rejected null hypotheses are not false positives. The ratio V/R is known as the *false discovery proportion* (FDP).

false discovery proportion

It might be tempting to ask the data analyst to control the FDP: to make sure that no more than, say, 20% of the rejected null hypotheses are false positives. However, in practice, controlling the FDP is an impossible task for the data analyst, since she has no way to be certain, on any particular dataset, which hypotheses are true and which are false. This is very similar to the fact that the data analyst can control the FWER, i.e. she can guarantee that $\Pr(V \geq 1) \leq \alpha$ for any pre-specified α, but she cannot guarantee that $V = 0$ on any particular dataset (short of failing to reject any null hypotheses, i.e. setting $R = 0$).

Therefore, we instead control the *false discovery rate* (FDR)[15], defined as

false discovery rate

$$\text{FDR} = \text{E}(\text{FDP}) = \text{E}(V/R). \tag{13.9}$$

When we control the FDR at (say) level $q = 20\%$, we are rejecting as many null hypotheses as possible while guaranteeing that no more than 20% of those rejected null hypotheses are false positives, *on average*.

In the definition of the FDR in (13.9), the expectation is taken over the population from which the data are generated. For instance, suppose we control the FDR for m null hypotheses at $q = 0.2$. This means that if we repeat this experiment a huge number of times, and each time control the FDR at $q = 0.2$, then we should expect that, on average, 20% of the rejected null hypotheses will be false positives. On a given dataset, the fraction of false positives among the rejected hypotheses may be greater than or less than 20%.

Thus far, we have motivated the use of the FDR from a pragmatic perspective, by arguing that when m is large, controlling the FWER is simply too stringent, and will not lead to "enough" discoveries. An additional motivation for the use of the FDR is that it aligns well with the way that data are often collected in contemporary applications. As datasets continue to grow in size across a variety of fields, it is increasingly common to conduct a huge number of hypothesis tests for exploratory, rather than confirmatory, purposes. For instance, a genomic researcher might sequence the genomes of individuals with and without some particular medical condition, and then, for each of 20,000 genes, test whether sequence variants in that gene are associated with the medical condition of interest. This amounts to performing $m = 20{,}000$ hypothesis tests. The analysis is exploratory in nature, in the sense that the researcher does not have any particular hypothesis in mind; instead she wishes to see whether there is modest evidence for the association between each gene and the disease, with a plan to further investigate any genes for which there is such evidence. She is likely willing to tolerate some number of false positives in the set of genes that she will investigate further; thus, the FWER is not an appropriate choice. However, some correction for multiple testing is required: it would not be a good idea for her to simply investigate *all* genes with p-values less than (say) 0.05, since we would expect 1,000 genes to have such small p-values simply by chance, even if no genes are associated with the disease (since $0.05 \times 20{,}000 = 1{,}000$). Controlling the FDR for her exploratory analysis at 20% guarantees that — on average — no more than 20% of the genes that she investigates further are false positives.

It is worth noting that unlike p-values, for which a threshold of 0.05 is typically viewed as the minimum standard of evidence for a "positive" result, and a threshold of 0.01 or even 0.001 is viewed as much more compelling, there is no standard accepted threshold for FDR control. Instead, the choice of FDR threshold is typically context-dependent, or even dataset-dependent. For instance, the genomic researcher in the previous example might seek to control the FDR at a threshold of 10% if the planned follow-

[15] If $R = 0$, then we replace the ratio V/R with 0, to avoid computing 0/0. Formally, $\text{FDR} = \text{E}(V/R|R > 0)\,\Pr(R > 0)$.

up analysis is time-consuming or expensive. Alternatively, a much larger threshold of 30% might be suitable if she plans an inexpensive follow-up analysis.

13.4.2 The Benjamini–Hochberg Procedure

We now focus on the task of controlling the FDR: that is, deciding which null hypotheses to reject while guaranteeing that the FDR, $\text{E}(V/R)$, is less than or equal to some pre-specified value q. In order to do this, we need some way to connect the p-values, $p_1, \ldots, p_m$, from the m null hypotheses to the desired FDR value, q. It turns out that a very simple procedure, outlined in Algorithm 13.2, can be used to control the FDR.

Algorithm 13.2 *Benjamini–Hochberg Procedure to Control the FDR*

1. Specify q, the level at which to control the FDR.
2. Compute p-values, $p_1, \ldots, p_m$, for the m null hypotheses $H_{01}, \ldots, H_{0m}$.
3. Order the m p-values so that $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$.
4. Define
$$L = \max\{j : p_{(j)} < qj/m\}. \tag{13.10}$$
5. Reject all null hypotheses H_{0j} for which $p_j \leq p_{(L)}$.

Algorithm 13.2 is known as the *Benjamini–Hochberg procedure*. The crux of this procedure lies in (13.10). For example, consider again the first five managers in the `Fund` dataset, presented in Table 13.3. (In this example, $m = 5$, although typically we control the FDR in settings involving a much greater number of null hypotheses.) We see that $p_{(1)} = 0.006 < 0.05 \times 1/5$, $p_{(2)} = 0.012 < 0.05 \times 2/5$, $p_{(3)} = 0.601 > 0.05 \times 3/5$, $p_{(4)} = 0.756 > 0.05 \times 4/5$, and $p_{(5)} = 0.918 > 0.05 \times 5/5$. Therefore, to control the FDR at 5%, we reject the null hypotheses that the first and third fund managers perform no better than chance.

Benjamini–Hochberg procedure

As long as the m p-values are independent or only mildly dependent, then the Benjamini–Hochberg procedure guarantees[16] that

$$\text{FDR} \leq q.$$

In other words, this procedure ensures that, on average, no more than a fraction q of the rejected null hypotheses are false positives. Remarkably, this holds regardless of how many null hypotheses are true, and regardless of the distribution of the p-values for the null hypotheses that are false. Therefore, the Benjamini–Hochberg procedure gives us a very easy way to determine, given a set of m p-values, which null hypotheses to reject in order to control the FDR at any pre-specified level q.

[16] However, the proof is well beyond the scope of this book.

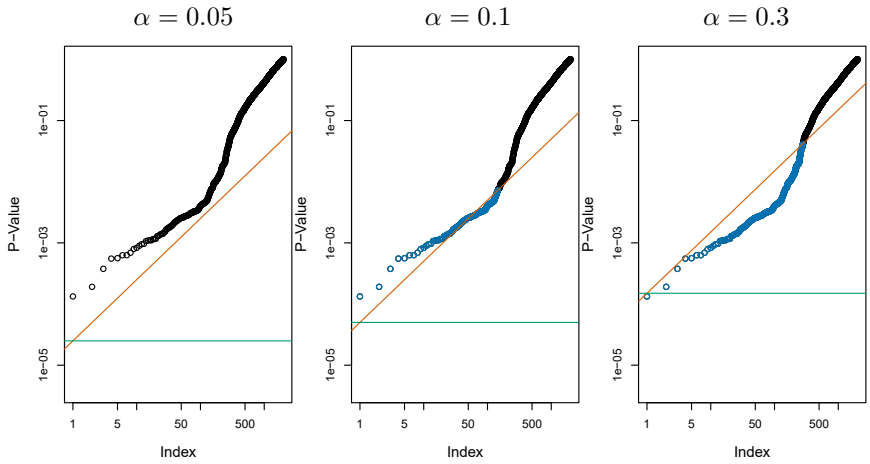

FIGURE 13.6. *Each panel displays the same set of* $m = 2{,}000$ *ordered p-values for the* `Fund` *data. The green lines indicate the p-value thresholds corresponding to FWER control, via the Bonferroni procedure, at levels* $\alpha = 0.05$ (left), $\alpha = 0.1$ (center), *and* $\alpha = 0.3$ (right). *The orange lines indicate the p-value thresholds corresponding to FDR control, via Benjamini–Hochberg, at levels* $q = 0.05$ (left), $q = 0.1$ (center), *and* $q = 0.3$ (right). *When the FDR is controlled at level* $q = 0.1$, *146 null hypotheses are rejected* (center)*; the corresponding p-values are shown in blue. When the FDR is controlled at level* $q = 0.3$, *279 null hypotheses are rejected* (right)*; the corresponding p-values are shown in blue.*

There is a fundamental difference between the Bonferroni procedure of Section 13.3.2 and the Benjamini–Hochberg procedure. In the Bonferroni procedure, in order to control the FWER for m null hypotheses at level α, we must simply reject null hypotheses for which the p-value is below α/m. This threshold of α/m does not depend on anything about the data (beyond the value of m), and certainly does not depend on the p-values themselves. By contrast, the rejection threshold used in the Benjamini–Hochberg procedure is more complicated: we reject all null hypotheses for which the p-value is less than or equal to the Lth smallest p-value, where L is itself a function of all m p-values, as in (13.10). Therefore, when conducting the Benjamini–Hochberg procedure, we cannot plan out in advance what threshold we will use to reject p-values; we need to first see our data. For instance, in the abstract, there is no way to know whether we will reject a null hypothesis corresponding to a p-value of 0.01 when using an FDR threshold of 0.1 with $m = 100$; the answer depends on the values of the other $m - 1$ p-values. This property of the Benjamini–Hochberg procedure is shared by the Holm procedure, which also involves a data-dependent p-value threshold.

Figure 13.6 displays the results of applying the Bonferroni and Benjamini–Hochberg procedures on the `Fund` data set, using the full set of $m = 2{,}000$ fund managers, of which the first five were displayed in Table 13.3. When the FWER is controlled at level 0.3 using Bonferroni, only one null hypothesis is rejected; that is, we can conclude only that a single fund manager is beating the market. This is despite the fact that a substantial portion of

the $m = 2{,}000$ fund managers appear to have beaten the market without performing correction for multiple testing — for instance, 13 of them have p-values below 0.001. By contrast, when the FDR is controlled at level 0.3, we can conclude that 279 fund managers are beating the market: we expect that no more than around $279 \times 0.3 = 83.7$ of these fund managers had good performance only due to chance. Thus, we see that FDR control is much milder — and more powerful — than FWER control, in the sense that it allows us to reject many more null hypotheses, with a cost of substantially more false positives.

The Benjamini–Hochberg procedure has been around since the mid-1990s. While a great many papers have been published since then proposing alternative approaches for FDR control that can perform better in particular scenarios, the Benjamini–Hochberg procedure remains a very useful and widely-applicable approach.

13.5 A Re-Sampling Approach to p-Values and False Discovery Rates

Thus far, the discussion in this chapter has assumed that we are interested in testing a particular null hypothesis H_0 using a test statistic T, which has some known (or assumed) distribution under H_0, such as a normal distribution, a t-distribution, a χ^2-distribution, or an F-distribution. This is referred to as the *theoretical null distribution*. We typically rely upon the availability of a theoretical null distribution in order to obtain a p-value associated with our test statistic. Indeed, for most of the types of null hypotheses that we might be interested in testing, a theoretical null distribution is available, provided that we are willing to make stringent assumptions about our data.

theoretical null distribution

However, if our null hypothesis H_0 or test statistic T is somewhat unusual, then it may be the case that no theoretical null distribution is available. Alternatively, even if a theoretical null distribution exists, then we may be wary of relying upon it, perhaps because some assumption that is required for it to hold is violated. For instance, maybe the sample size is too small.

In this section, we present a framework for performing inference in this setting, which exploits the availability of fast computers in order to approximate the null distribution of T, and thereby to obtain a p-value. While this framework is very general, it must be carefully instantiated for a specific problem of interest. Therefore, in what follows, we consider a specific example in which we wish to test whether the means of two random variables are equal, using a two-sample t-test.

The discussion in this section is more challenging than the preceding sections in this chapter, and can be safely skipped by a reader who is content to use the theoretical null distribution to compute p-values for his or her test statistics.

13.5.1 A Re-Sampling Approach to the p-Value

We return to the example of Section 13.1.1, in which we wish to test whether the mean of a random variable X equals the mean of a random variable Y, i.e. $H_0 : \mathrm{E}(X) = \mathrm{E}(Y)$, against the alternative $H_a : \mathrm{E}(X) \neq \mathrm{E}(Y)$. Given n_X independent observations from X and n_Y independent observations from Y, the two-sample t-statistic takes the form

$$T = \frac{\hat{\mu}_X - \hat{\mu}_Y}{s\sqrt{\frac{1}{n_X} + \frac{1}{n_Y}}} \tag{13.11}$$

where $\hat{\mu}_X = \frac{1}{n_X}\sum_{i=1}^{n_X} x_i$, $\hat{\mu}_Y = \frac{1}{n_Y}\sum_{i=1}^{n_Y} y_i$, $s = \sqrt{\frac{(n_X-1)s_X^2+(n_Y-1)s_Y^2}{n_X+n_Y-2}}$, and s_X^2 and s_Y^2 are unbiased estimators of the variances in the two groups. A large (absolute) value of T provides evidence against H_0.

If n_X and n_Y are large, then T in (13.11) approximately follows a $N(0, 1)$ distribution. But if n_X and n_Y are small, then in the absence of a strong assumption about the distribution of X and Y, we do not know the theoretical null distribution of T.[17] In this case, it turns out that we can approximate the null distribution of T using a *re-sampling* approach, or more specifically, a *permutation* approach.

re-sampling
permutation

To do this, we conduct a thought experiment. If H_0 holds, so that $\mathrm{E}(X) = \mathrm{E}(Y)$, and we make the stronger assumption that the distributions of X and Y are the same, then the distribution of T is invariant under swapping observations of X with observations of Y. That is, if we randomly swap some of the observations in X with the observations in Y, then *the test statistic T in (13.11) computed based on this swapped data has the same distribution as T based on the original data.* This is true only if H_0 holds, and the distributions of X and Y are the same.

This suggests that in order to approximate the null distribution of T, we can take the following approach. We randomly permute the $n_X + n_Y$ observations B times, for some large value of B, and each time we compute (13.11). We let $T^{*1}, \ldots, T^{*B}$ denote the values of (13.11) on the permuted data. These can be viewed as an approximation of the null distribution of T under H_0. Recall that by definition, a p-value is the probability of observing a test statistic at least this extreme under H_0. Therefore, to compute a p-value for T, we can simply compute

$$p\text{-value} = \frac{\sum_{b=1}^{B} 1_{(|T^{*b}| \geq |T|)}}{B}, \tag{13.12}$$

the fraction of permuted datasets for which the value of the test statistic is at least as extreme as the value observed on the original data. This procedure is summarized in Algorithm 13.3.

[17] If we assume that X and Y are normally distributed, then T in (13.11) follows a t-distribution with $n_X + n_Y - 2$ degrees of freedom under H_0. However, in practice, the distribution of random variables is rarely known, and so it can be preferable to perform a re-sampling approach instead of making strong and unjustified assumptions. If the results of the re-sampling approach disagree with the results of assuming a theoretical null distribution, then the results of the re-sampling approach are more trustworthy.

Algorithm 13.3 *Re-Sampling p-Value for a Two-Sample t-Test*

1. Compute T, defined in (13.11), on the original data $x_1, \ldots, x_{n_X}$ and $y_1, \ldots, y_{n_Y}$.
2. For $b = 1, \ldots, B$, where B is a large number (e.g. $B = 10{,}000$):
   (a) Permute the $n_X + n_Y$ observations at random. Call the first n_X permuted observations $x^*_1, \ldots, x^*_{n_X}$, and call the remaining n_Y observations $y^*_1, \ldots, y^*_{n_Y}$.
   (b) Compute (13.11) on the permuted data $x^*_1, \ldots, x^*_{n_X}$ and $y^*_1, \ldots, y^*_{n_Y}$, and call the result T^{*b}.
3. The p-value is given by $\frac{\sum_{b=1}^{B} 1_{(|T^{*b}| \geq |T|)}}{B}$.

We try out this procedure on the `Khan` dataset, which consists of expression measurements for 2,308 genes in four sub-types of small round blood cell tumors, a type of cancer typically seen in children. This dataset is part of the `ISLR2` package. We restrict our attention to the two sub-types for which the most observations are available: rhabdomyosarcoma ($n_X = 29$) and Burkitt's lymphoma ($n_Y = 25$).

A two-sample t-test for the null hypothesis that the 11th gene's mean expression values are equal in the two groups yields $T = -2.09$. Using the theoretical null distribution, which is a t_{52} distribution (since $n_X + n_Y - 2 = 52$), we obtain a p-value of 0.041. (Note that a t_{52} distribution is virtually indistinguishable from a $N(0, 1)$ distribution.) If we instead apply Algorithm 13.3 with $B = 10{,}000$, then we obtain a p-value of 0.042. Figure 13.7 displays the theoretical null distribution, the re-sampling null distribution, and the actual value of the test statistic ($T = -2.09$) for this gene. In this example, we see very little difference between the p-values obtained using the theoretical null distribution and the re-sampling null distribution.

By contrast, Figure 13.8 shows an analogous set of results for the 877th gene. In this case, there is a substantial difference between the theoretical and re-sampling null distributions, which results in a difference between their p-values.

In general, in settings with a smaller sample size or a more skewed data distribution (so that the theoretical null distribution is less accurate), the difference between the re-sampling and theoretical p-values will tend to be more pronounced. In fact, the substantial difference between the re-sampling and theoretical null distributions in Figure 13.8 is due to the fact that a single observation in the 877th gene is very far from the other observations, leading to a very skewed distribution.

13.5.2 A Re-Sampling Approach to the False Discovery Rate

Now, suppose that we wish to control the FDR for m null hypotheses, $H_{01}, \ldots, H_{0m}$, in a setting in which either no theoretical null distribution is available, or else we simply prefer to avoid the use of a theoretical null

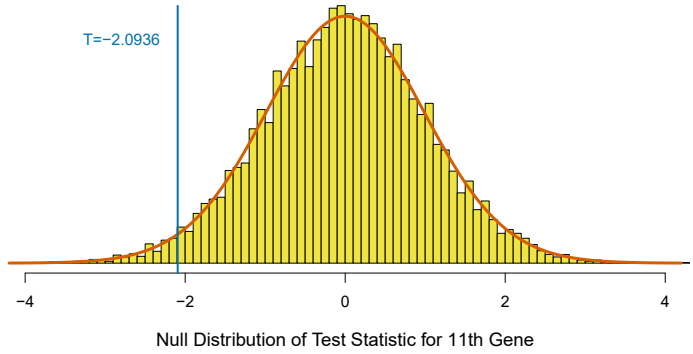

FIGURE 13.7. *The* 11*th gene in the* Khan *dataset has a test statistic of* $T = -2.09$*. Its theoretical and re-sampling null distributions are almost identical. The theoretical p-value equals* 0.041 *and the re-sampling p-value equals* 0.042*.*

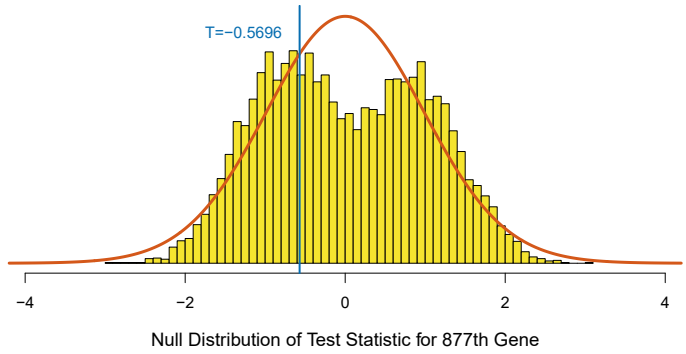

FIGURE 13.8. *The* 877*th gene in the* Khan *dataset has a test statistic of* $T = -0.57$*. Its theoretical and re-sampling null distributions are quite different. The theoretical p-value equals* 0.571*, and the re-sampling p-value equals* 0.673*.*

distribution. As in Section 13.5.1, we make use of a two-sample t-statistic for each hypothesis, leading to the test statistics $T_1, \ldots, T_m$. We could simply compute a p-value for each of the m null hypotheses, as in Section 13.5.1, and then apply the Benjamini–Hochberg procedure of Section 13.4.2 to these p-values. However, it turns out that we can do this in a more direct way, without even needing to compute p-values.

Recall from Section 13.4 that the FDR is defined as $\mathrm{E}(V/R)$, using the notation in Table 13.2. In order to estimate the FDR via re-sampling, we first make the following approximation:

$$\mathrm{FDR} = E\left(\frac{V}{R}\right) \approx \frac{\mathrm{E}(V)}{R}. \tag{13.13}$$

Now suppose we reject any null hypothesis for which the test statistic exceeds c in absolute value. Then computing R in the denominator on the right-hand side of (13.13) is straightforward: $R = \sum_{j=1}^{m} 1_{(|T_j| \geq c)}$.

However, the numerator $\mathrm{E}(V)$ on the right-hand side of (13.13) is more challenging. This is the expected number of false positives associated with rejecting any null hypothesis for which the test statistic exceeds c in absolute value. At the risk of stating the obvious, estimating V is challenging because we do not know which of $H_{01}, \ldots, H_{0m}$ are really true, and so we do not know which rejected hypotheses are false positives. To overcome this problem, we take a re-sampling approach, in which we simulate data under $H_{01}, \ldots, H_{0m}$, and then compute the resulting test statistics. The number of re-sampled test statistics that exceed c provides an estimate of V.

In greater detail, in the case of a two-sample t-statistic (13.11) for each of the null hypotheses $H_{01}, \ldots, H_{0m}$, we can estimate $\mathrm{E}(V)$ as follows. Let $x_1^{(j)}, \ldots, x_{n_X}^{(j)}$ and $y_1^{(j)}, \ldots, y_{n_Y}^{(j)}$ denote the data associated with the jth null hypothesis, $j = 1, \ldots, m$. We permute these $n_X + n_Y$ observations at random, and then compute the t-statistic on the permuted data. For this permuted data, we know that all of the null hypotheses $H_{01}, \ldots, H_{0m}$ hold; therefore, the number of permuted t-statistics that exceed the threshold c in absolute value provides an estimate for $\mathrm{E}(V)$. This estimate can be further improved by repeating the permutation process B times, for a large value of B, and averaging the results.

Algorithm 13.4 details this procedure.[18] It provides what is known as a *plug-in estimate* of the FDR, because the approximation in (13.13) allows us to estimate the FDR by plugging R into the denominator and an estimate for $\mathrm{E}(V)$ into the numerator.

We apply the re-sampling approach to the FDR from Algorithm 13.4, as well as the Benjamini–Hochberg approach from Algorithm 13.2 using theoretical p-values, to the $m = 2{,}308$ genes in the `Khan` dataset. Results are shown in Figure 13.9. We see that for a given number of rejected hypotheses, the estimated FDRs are almost identical for the two methods.

We began this section by noting that in order to control the FDR for m hypothesis tests using a re-sampling approach, we could simply compute m re-sampling p-values as in Section 13.5.1, and then apply the Benjamini–Hochberg procedure of Section 13.4.2 to these p-values. It turns out that if we define the jth re-sampling p-value as

$$p_j = \frac{\sum_{j'=1}^{m} \sum_{b=1}^{B} 1_{(|T_{j'}^{*b}| \geq |T_j|)}}{Bm} \tag{13.14}$$

for $j = 1, \ldots, m$, instead of as in (13.12), then applying the Benjamini–Hochberg procedure to these re-sampled p-values is *exactly* equivalent to Algorithm 13.4. Note that (13.14) is an alternative to (13.12) that pools the information across all m hypothesis tests in approximating the null distribution.

13.5.3 When Are Re-Sampling Approaches Useful?

In Sections 13.5.1 and 13.5.2, we considered testing null hypotheses of the form $H_0 : \mathrm{E}(X) = \mathrm{E}(Y)$ using a two-sample t-statistic (13.11), for which we

[18] To implement Algorithm 13.4 efficiently, the same set of permutations in Step 2(b)i. should be used for all m null hypotheses.

Algorithm 13.4 *Plug-In FDR for a Two-Sample T-Test*

1. Select a threshold c, where $c > 0$.
2. For $j = 1, \ldots, m$:
   (a) Compute $T^{(j)}$, the two-sample t-statistic (13.11) for the null hypothesis H_{0j} on the basis of the original data, $x_1^{(j)}, \ldots, x_{n_X}^{(j)}$ and $y_1^{(j)}, \ldots, y_{n_Y}^{(j)}$.
   (b) For $b = 1, \ldots, B$, where B is a large number (e.g. $B = 10{,}000$):
      i. Permute the $n_X + n_Y$ observations at random. Call the first n_X observations $x_1^{*(j)}, \ldots, x_{n_X}^{*(j)}$, and call the remaining observations $y_1^{*(j)}, \ldots, y_{n_Y}^{*(j)}$.
      ii. Compute (13.11) on the permuted data $x_1^{*(j)}, \ldots, x_{n_X}^{*(j)}$ and $y_1^{*(j)}, \ldots, y_{n_Y}^{*(j)}$, and call the result $T^{(j),*b}$.
3. Compute $R = \sum_{j=1}^{m} 1_{(|T^{(j)}| \geq c)}$.
4. Compute $\widehat{V} = \frac{\sum_{b=1}^{B} \sum_{j=1}^{m} 1_{(|T^{(j),*b}| \geq c)}}{B}$.
5. The estimated FDR associated with the threshold c is $\widehat{V}/R$.

approximated the null distribution via a re-sampling approach. We saw that using the re-sampling approach gave us substantially different results from using the theoretical p-value approach in Figure 13.8, but not in Figure 13.7.

In general, there are two settings in which a re-sampling approach is particularly useful:

1. Perhaps no theoretical null distribution is available. This may be the case if you are testing an unusual null hypothesis H_0, or using an unsual test statistic T.

2. Perhaps a theoretical null distribution *is* available, but the assumptions required for its validity do not hold. For instance, the two-sample t-statistic in (13.11) follows a $t_{n_X+n_Y-2}$ distribution only if the observations are normally distributed. Furthermore, it follows a $N(0,1)$ distribution only if n_X and n_Y are quite large. If the data are non-normal and n_X and n_Y are small, then p-values that make use of the theoretical null distribution will not be valid (i.e. they will not properly control the Type I error).

In general, if you can come up with a way to re-sample or permute your observations in order to generate data that follow the null distribution, then you can compute p-values or estimate the FDR using variants of Algorithms 13.3 and 13.4. In many real-world settings, this provides a powerful tool for hypothesis testing when no out-of-box hypothesis tests are available, or when the key assumptions underlying those out-of-box tests are violated.

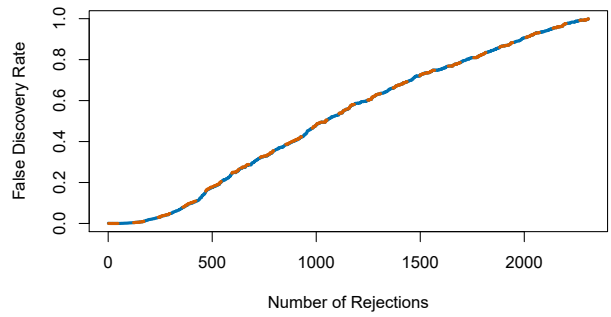

FIGURE 13.9. *For $j = 1, \ldots, m = 2{,}308$, we tested the null hypothesis that for the jth gene in the* `Khan` *dataset, the mean expression in Burkitt's lymphoma equals the mean expression in rhabdomyosarcoma. For each value of k from 1 to 2,308, the y-axis displays the estimated FDR associated with rejecting the null hypotheses corresponding to the k smallest p-values. The orange dashed curve shows the FDR obtained using the Benjamini–Hochberg procedure, whereas the blue solid curve shows the FDR obtained using the re-sampling approach of Algorithm 13.4, with $B = 10{,}000$. There is very little difference between the two FDR estimates. According to either estimate, rejecting the null hypothesis for the 500 genes with the smallest p-values corresponds to an FDR of around 17.7%.*

13.6 Lab: Multiple Testing

We include our usual imports seen in earlier labs.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from ISLP import load_data
```

We also collect the new imports needed for this lab.

In [2]:
```
from scipy.stats import \
    (ttest_1samp,
     ttest_rel,
     ttest_ind,
     t as t_dbn)
from statsmodels.stats.multicomp import \
     pairwise_tukeyhsd
from statsmodels.stats.multitest import \
     multipletests as mult_test
```

13.6.1 Review of Hypothesis Tests

We begin by performing some one-sample t-tests. First we create 100 variables, each consisting of 10 observations. The first 50 variables have mean 0.5 and variance 1, while the others have mean 0 and variance 1.

In [3]:
```
rng = np.random.default_rng(12)
X = rng.standard_normal((10, 100))
true_mean = np.array([0.5]*50 + [0]*50)
X += true_mean[None,:]
```

To begin, we use `ttest_1samp()` from the `scipy.stats` module to test $H_0 : \mu_1 = 0$, the null hypothesis that the first variable has mean zero. `ttest_1samp()`

In [4]:
```
result = ttest_1samp(X[:,0], 0)
result.pvalue
```

Out[4]: 0.931

The p-value comes out to 0.931, which is not low enough to reject the null hypothesis at level $\alpha = 0.05$. In this case, $\mu_1 = 0.5$, so the null hypothesis is false. Therefore, we have made a Type II error by failing to reject the null hypothesis when the null hypothesis is false.

We now test $H_{0,j} : \mu_j = 0$ for $j = 1, \ldots, 100$. We compute the 100 p-values, and then construct a vector recording whether the jth p-value is less than or equal to 0.05, in which case we reject H_{0j}, or greater than 0.05, in which case we do not reject H_{0j}, for $j = 1, \ldots, 100$.

In [5]:
```
p_values = np.empty(100)
for i in range(100):
   p_values[i] = ttest_1samp(X[:,i], 0).pvalue
decision = pd.cut(p_values,
                  [0, 0.05, 1],
                  labels=['Reject H0',
                          'Do not reject H0'])
truth = pd.Categorical(true_mean == 0,
                       categories=[True, False],
                       ordered=True)
```

Since this is a simulated data set, we can create a 2×2 table similar to Table 13.2.

In [6]:
```
pd.crosstab(decision,
            truth,
     rownames=['Decision'],
     colnames=['H0'])
```

Out[6]:

H0	True	False
Decision		
Reject H0	5	15
Do not reject H0	45	35

Therefore, at level $\alpha = 0.05$, we reject 15 of the 50 false null hypotheses, and we incorrectly reject 5 of the true null hypotheses. Using the notation from Section 13.3, we have $V = 5$, $S = 15$, $U = 45$ and $W = 35$. We have set $\alpha = 0.05$, which means that we expect to reject around 5% of the true null hypotheses. This is in line with the 2×2 table above, which indicates that we rejected $V = 5$ of the 50 true null hypotheses.

In the simulation above, for the false null hypotheses, the ratio of the mean to the standard deviation was only $0.5/1 = 0.5$. This amounts to quite a weak signal, and it resulted in a high number of Type II errors. Let's instead simulate data with a stronger signal, so that the ratio of the mean to the standard deviation for the false null hypotheses equals 1. We make only 10 Type II errors.

In [7]:
```
true_mean = np.array([1]*50 + [0]*50)
X = rng.standard_normal((10, 100))
X += true_mean[None,:]
for i in range(100):
   p_values[i] = ttest_1samp(X[:,i], 0).pvalue
decision = pd.cut(p_values,
                  [0, 0.05, 1],
                  labels=['Reject H0',
                          'Do not reject H0'])
truth = pd.Categorical(true_mean == 0,
                       categories=[True, False],
                       ordered=True)
pd.crosstab(decision,
            truth,
            rownames=['Decision'],
            colnames=['H0'])
```

Out[7]:

Decision \ H0	True	False
Reject H0	2	40
Do not reject H0	48	10

13.6.2 Family-Wise Error Rate

Recall from (13.5) that if the null hypothesis is true for each of m independent hypothesis tests, then the FWER is equal to $1 - (1 - \alpha)^m$. We can use this expression to compute the FWER for $m = 1, \ldots, 500$ and $\alpha = 0.05$, 0.01, and 0.001. We plot the FWER for these values of α in order to reproduce Figure 13.2.

In [8]:
```
m = np.linspace(1, 501)
fig, ax = plt.subplots()
[ax.plot(m,
         1 - (1 - alpha)**m,
         label=r'$\alpha=%s$' % str(alpha))
         for alpha in [0.05, 0.01, 0.001]]
ax.set_xscale('log')
ax.set_xlabel('Number of Hypotheses')
ax.set_ylabel('Family-Wise Error Rate')
ax.legend()
ax.axhline(0.05, c='k', ls='--');
```

As discussed previously, even for moderate values of m such as 50, the FWER exceeds 0.05 unless α is set to a very low value, such as 0.001. Of course, the problem with setting α to such a low value is that we are likely to make a number of Type II errors: in other words, our power is very low.

We now conduct a one-sample t-test for each of the first five managers in the `Fund` dataset, in order to test the null hypothesis that the jth fund manager's mean return equals zero, $H_{0,j} : \mu_j = 0$.

In [9]:
```
Fund = load_data('Fund')
fund_mini = Fund.iloc[:,:5]
fund_mini_pvals = np.empty(5)
for i in range(5):
```

```
    fund_mini_pvals[i] = ttest_1samp(fund_mini.iloc[:,i], 0).pvalue
fund_mini_pvals
```

```
Out[9]: array([0.006, 0.918, 0.012, 0.601, 0.756])
```

The p-values are low for Managers One and Three, and high for the other three managers. However, we cannot simply reject $H_{0,1}$ and $H_{0,3}$, since this would fail to account for the multiple testing that we have performed. Instead, we will conduct Bonferroni's method and Holm's method to control the FWER.

multiple-tests()

adjusted p-values

To do this, we use the `multipletests()` function from the `statsmodels` module (abbreviated to `mult_test()`). Given the p-values, for methods like Holm and Bonferroni the function outputs *adjusted p-values*, which can be thought of as a new set of p-values that have been corrected for multiple testing. If the adjusted p-value for a given hypothesis is less than or equal to α, then that hypothesis can be rejected while maintaining a FWER of no more than α. In other words, for such methods, the adjusted p-values resulting from the `multipletests()` function can simply be compared to the desired FWER in order to determine whether or not to reject each hypothesis. We will later see that we can use the same function to control FDR as well.

The `mult_test()` function takes p-values and a `method` argument, as well as an optional `alpha` argument. It returns the decisions (`reject` below) as well as the adjusted p-values (`bonf`).

```
In [10]: reject, bonf = mult_test(fund_mini_pvals, method = "bonferroni")[:2]
reject
```

```
Out[10]: array([ True, False, False, False, False])
```

The p-values `bonf` are simply the `fund_mini_pvalues` multiplied by 5 and truncated to be less than or equal to 1.

```
In [11]: bonf, np.minimum(fund_mini_pvals * 5, 1)
```

```
Out[11]: (array([0.03, 1.  , 0.06, 1.  , 1.  ]),
          array([0.03, 1.  , 0.06, 1.  , 1.  ]))
```

Therefore, using Bonferroni's method, we are able to reject the null hypothesis only for Manager One while controlling FWER at 0.05.

By contrast, using Holm's method, the adjusted p-values indicate that we can reject the null hypotheses for Managers One and Three at a FWER of 0.05.

```
In [12]: mult_test(fund_mini_pvals, method = "holm", alpha=0.05)[:2]
```

```
Out[12]: (array([ True, False,  True, False, False]),
          array([0.03, 1.  , 0.05, 1.  , 1.  ]))
```

As discussed previously, Manager One seems to perform particularly well, whereas Manager Two has poor performance.

```
In [13]: fund_mini.mean()
```

```
Out[13]: Manager1    3.0
         Manager2   -0.1
         Manager3    2.8
         Manager4    0.5
         Manager5    0.3
         dtype: float64
```

Is there evidence of a meaningful difference in performance between these two managers? We can check this by performing a *paired t-test* using the `ttest_rel()` function from `scipy.stats`:

paired t-test

`ttest_rel()`

```
In [14]: ttest_rel(fund_mini['Manager1'],
                   fund_mini['Manager2']).pvalue
```

```
Out[14]: 0.038
```

The test results in a p-value of 0.038, suggesting a statistically significant difference.

However, we decided to perform this test only after examining the data and noting that Managers One and Two had the highest and lowest mean performances. In a sense, this means that we have implicitly performed $\binom{5}{2} = 5(5-1)/2 = 10$ hypothesis tests, rather than just one, as discussed in Section 13.3.2. Hence, we use the `pairwise_tukeyhsd()` function from `statsmodels.stats.multicomp` to apply Tukey's method in order to adjust for multiple testing. This function takes as input a fitted *ANOVA* regression model, which is essentially just a linear regression in which all of the predictors are qualitative. In this case, the response consists of the monthly excess returns achieved by each manager, and the predictor indicates the manager to which each return corresponds.

`pairwise_tukeyhsd()`

ANOVA

```
In [15]: returns = np.hstack([fund_mini.iloc[:,i] for i in range(5)])
         managers = np.hstack([[i+1]*50 for i in range(5)])
         tukey = pairwise_tukeyhsd(returns, managers)
         print(tukey.summary())
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====================================================
group1 group2 meandiff p-adj   lower  upper  reject
-----------------------------------------------------
     1      2     -3.1 0.1862 -6.9865 0.7865  False
     1      3     -0.2 0.9999 -4.0865 3.6865  False
     1      4     -2.5 0.3948 -6.3865 1.3865  False
     1      5     -2.7 0.3152 -6.5865 1.1865  False
     2      3      2.9 0.2453 -0.9865 6.7865  False
     2      4      0.6 0.9932 -3.2865 4.4865  False
     2      5      0.4 0.9986 -3.4865 4.2865  False
     3      4     -2.3  0.482 -6.1865 1.5865  False
     3      5     -2.5 0.3948 -6.3865 1.3865  False
     4      5     -0.2 0.9999 -4.0865 3.6865  False
-----------------------------------------------------
```

The `pairwise_tukeyhsd()` function provides confidence intervals for the difference between each pair of managers (`lower` and `upper`), as well as a

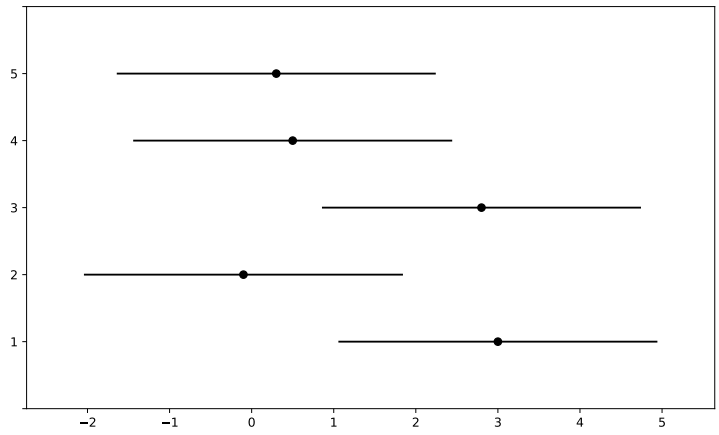

FIGURE 13.10. *95% confidence intervals for each manager on the* `Fund` *data, using Tukey's method to adjust for multiple testing. All of the confidence intervals overlap, so none of the differences among managers are statistically significant when controlling FWER at level* 0.05.

p-value. All of these quantities have been adjusted for multiple testing. Notice that the p-value for the difference between Managers One and Two has increased from 0.038 to 0.186, so there is no longer clear evidence of a difference between the managers' performances. We can plot the confidence intervals for the pairwise comparisons using the `plot_simultaneous()` method of `tukey`. Any pair of intervals that don't overlap indicates a significant difference at the nominal level of 0.05. In this case, no differences are considered significant as reported in the table above.

In [16]:
```
fig, ax = plt.subplots(figsize=(8,8))
tukey.plot_simultaneous(ax=ax);
```

The result can be seen[19] in Figure 13.10.

13.6.3 False Discovery Rate

Now we perform hypothesis tests for all 2,000 fund managers in the `Fund` dataset. We perform a one-sample t-test of $H_{0,j} : \mu_j = 0$, which states that the jth fund manager's mean return is zero.

In [17]:
```
fund_pvalues = np.empty(2000)
for i, manager in enumerate(Fund.columns):
    fund_pvalues[i] = ttest_1samp(Fund[manager], 0).pvalue
```

There are far too many managers to consider trying to control the FWER. Instead, we focus on controlling the FDR: that is, the expected fraction of rejected null hypotheses that are actually false positives. The

[19] Traditionally this plot shows intervals for each paired difference. With many groups it is more convenient and equivalent to display one interval per group, as is done here. By "differencing" all pairs of intervals displayed here you recover the traditional plot.

`multipletests()` function (abbreviated `mult_test()`) can be used to carry out the Benjamini–Hochberg procedure.

In [18]:
```
fund_qvalues = mult_test(fund_pvalues, method = "fdr_bh")[1]
fund_qvalues[:10]
```

Out[18]:
```
array([0.09, 0.99, 0.12, 0.92, 0.96, 0.08, 0.08, 0.08, 0.08,
       0.08])
```

The *q-values* output by the Benjamini–Hochberg procedure can be interpreted as the smallest FDR threshold at which we would reject a particular null hypothesis. For instance, a q-value of 0.1 indicates that we can reject the corresponding null hypothesis at an FDR of 10% or greater, but that we cannot reject the null hypothesis at an FDR below 10%. q-values

If we control the FDR at 10%, then for how many of the fund managers can we reject $H_{0,j} : \mu_j = 0$?

In [19]:
```
(fund_qvalues <= 0.1).sum()
```

Out[19]:
```
146
```

We find that 146 of the 2,000 fund managers have a q-value below 0.1; therefore, we are able to conclude that 146 of the fund managers beat the market at an FDR of 10%. Only about 15 (10% of 146) of these fund managers are likely to be false discoveries.

By contrast, if we had instead used Bonferroni's method to control the FWER at level $\alpha = 0.1$, then we would have failed to reject any null hypotheses!

In [20]:
```
(fund_pvalues <= 0.1 / 2000).sum()
```

Out[20]:
```
0
```

Figure 13.6 displays the ordered p-values, $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(2000)}$, for the `Fund` dataset, as well as the threshold for rejection by the Benjamini–Hochberg procedure. Recall that the Benjamini–Hochberg procedure identifies the largest p-value such that $p_{(j)} < qj/m$, and rejects all hypotheses for which the p-value is less than or equal to $p_{(j)}$. In the code below, we implement the Benjamini–Hochberg procedure ourselves, in order to illustrate how it works. We first order the p-values. We then identify all p-values that satisfy $p_{(j)} < qj/m$ (`sorted_set_`). Finally, `selected_` is a boolean array indicating which p-values are less than or equal to the largest p-value in `sorted_[sorted_set_]`. Therefore, `selected_` indexes the p-values rejected by the Benjamini–Hochberg procedure.

In [21]:
```
sorted_ = np.sort(fund_pvalues)
m = fund_pvalues.shape[0]
q = 0.1
sorted_set_ = np.where(sorted_ < q * np.linspace(1, m, m) / m)[0]
if sorted_set_.shape[0] > 0:
    selected_ = fund_pvalues < sorted_[sorted_set_].max()
    sorted_set_ = np.arange(sorted_set_.max())
else:
    selected_ = []
    sorted_set_ = []
```

We now reproduce the middle panel of Figure 13.6.

In [22]:
```
fig, ax = plt.subplots()
ax.scatter(np.arange(0, sorted_.shape[0]) + 1,
           sorted_, s=10)
ax.set_yscale('log')
ax.set_xscale('log')
ax.set_ylabel('P-Value')
ax.set_xlabel('Index')
ax.scatter(sorted_set_+1, sorted_[sorted_set_], c='r', s=20)
ax.axline((0, 0), (1,q/m), c='k', ls='--', linewidth=3);
```

13.6.4 A Re-Sampling Approach

Here, we implement the re-sampling approach to hypothesis testing using the `Khan` dataset, which we investigated in Section 13.5. First, we merge the training and testing data, which results in observations on 83 patients for 2,308 genes.

In [23]:
```
Khan = load_data('Khan')
D = pd.concat([Khan['xtrain'], Khan['xtest']])
D['Y'] = pd.concat([Khan['ytrain'], Khan['ytest']])
D['Y'].value_counts()
```

Out[23]:
```
2    29
4    25
3    18
1    11
Name: Y, dtype: int64
```

There are four classes of cancer. For each gene, we compare the mean expression in the second class (rhabdomyosarcoma) to the mean expression in the fourth class (Burkitt's lymphoma). Performing a standard two-sample t-test using `ttest_ind()` from `scipy.stats` on the 11th gene produces a test-statistic of -2.09 and an associated p-value of 0.0412, suggesting modest evidence of a difference in mean expression levels between the two cancer types.

`ttest_ind()`

In [24]:
```
D2 = D[lambda df:df['Y'] == 2]
D4 = D[lambda df:df['Y'] == 4]
gene_11 = 'G0011'
observedT, pvalue = ttest_ind(D2[gene_11],
                              D4[gene_11],
                              equal_var=True)
observedT, pvalue
```

Out[24]:
```
(-2.094, 0.041)
```

However, this p-value relies on the assumption that under the null hypothesis of no difference between the two groups, the test statistic follows a t-distribution with $29 + 25 - 2 = 52$ degrees of freedom. Instead of using this theoretical null distribution, we can randomly split the 54 patients into two groups of 29 and 25, and compute a new test statistic. Under the null hypothesis of no difference between the groups, this new test statistic should have the same distribution as our original one. Repeating this

process 10,000 times allows us to approximate the null distribution of the test statistic. We compute the fraction of the time that our observed test statistic exceeds the test statistics obtained via re-sampling.

In [25]:
```
B = 10000
Tnull = np.empty(B)
D_ = np.hstack([D2[gene_11], D4[gene_11]])
n_ = D2[gene_11].shape[0]
D_null = D_.copy()
for b in range(B):
    rng.shuffle(D_null)
    ttest_ = ttest_ind(D_null[:n_],
                       D_null[n_:],
                       equal_var=True)
    Tnull[b] = ttest_.statistic
(np.abs(Tnull) > np.abs(observedT)).mean()
```

Out[25]: 0.0398

This fraction, 0.0398, is our re-sampling-based p-value. It is almost identical to the p-value of 0.0412 obtained using the theoretical null distribution. We can plot a histogram of the re-sampling-based test statistics in order to reproduce Figure 13.7.

In [26]:
```
fig, ax = plt.subplots(figsize=(8,8))
ax.hist(Tnull,
        bins=100,
        density=True,
        facecolor='y',
        label='Null')
xval = np.linspace(-4.2, 4.2, 1001)
ax.plot(xval,
        t_dbn.pdf(xval, D_.shape[0]-2),
        c='r')
ax.axvline(observedT,
           c='b',
           label='Observed')
ax.legend()
ax.set_xlabel("Null Distribution of Test Statistic");
```

The re-sampling-based null distribution is almost identical to the theoretical null distribution, which is displayed in red.

Finally, we implement the plug-in re-sampling FDR approach outlined in Algorithm 13.4. Depending on the speed of your computer, calculating the FDR for all 2,308 genes in the `Khan` dataset may take a while. Hence, we will illustrate the approach on a random subset of 100 genes. For each gene, we first compute the observed test statistic, and then produce 10,000 re-sampled test statistics. This may take a few minutes to run. If you are in a rush, then you could set `B` equal to a smaller value (e.g. `B=500`).

In [27]:
```
m, B = 100, 10000
idx = rng.choice(Khan['xtest'].columns, m, replace=False)
T_vals = np.empty(m)
Tnull_vals = np.empty((m, B))

for j in range(m):
    col = idx[j]
```

```
    T_vals[j] = ttest_ind(D2[col],
                          D4[col],
                          equal_var=True).statistic
    D_ = np.hstack([D2[col], D4[col]])
    D_null = D_.copy()
    for b in range(B):
        rng.shuffle(D_null)
        ttest_ = ttest_ind(D_null[:n_],
                           D_null[n_:],
                           equal_var=True)
        Tnull_vals[j,b] = ttest_.statistic
```

Next, we compute the number of rejected null hypotheses R, the estimated number of false positives $\widehat{V}$, and the estimated FDR, for a range of threshold values c in Algorithm 13.4. The threshold values are chosen using the absolute values of the test statistics from the 100 genes.

In [28]:
```
cutoffs = np.sort(np.abs(T_vals))
FDRs, Rs, Vs = np.empty((3, m))
for j in range(m):
   R = np.sum(np.abs(T_vals) >= cutoffs[j])
   V = np.sum(np.abs(Tnull_vals) >= cutoffs[j]) / B
   Rs[j] = R
   Vs[j] = V
   FDRs[j] = V / R
```

Now, for any given FDR, we can find the genes that will be rejected. For example, with FDR controlled at 0.1, we reject 15 of the 100 null hypotheses. On average, we would expect about one or two of these genes (i.e. 10% of 15) to be false discoveries. At an FDR of 0.2, we can reject the null hypothesis for 28 genes, of which we expect around six to be false discoveries.

The variable `idx` stores which genes were included in our 100 randomly-selected genes. Let's look at the genes whose estimated FDR is less than 0.1.

In [29]:
```
sorted(idx[np.abs(T_vals) >= cutoffs[FDRs < 0.1].min()])
```

At an FDR threshold of 0.2, more genes are selected, at the cost of having a higher expected proportion of false discoveries.

In [30]:
```
sorted(idx[np.abs(T_vals) >= cutoffs[FDRs < 0.2].min()])
```

The next line generates Figure 13.11, which is similar to Figure 13.9, except that it is based on only a subset of the genes.

In [31]:
```
fig, ax = plt.subplots()
ax.plot(Rs, FDRs, 'b', linewidth=3)
ax.set_xlabel("Number of Rejections")
ax.set_ylabel("False Discovery Rate");
```

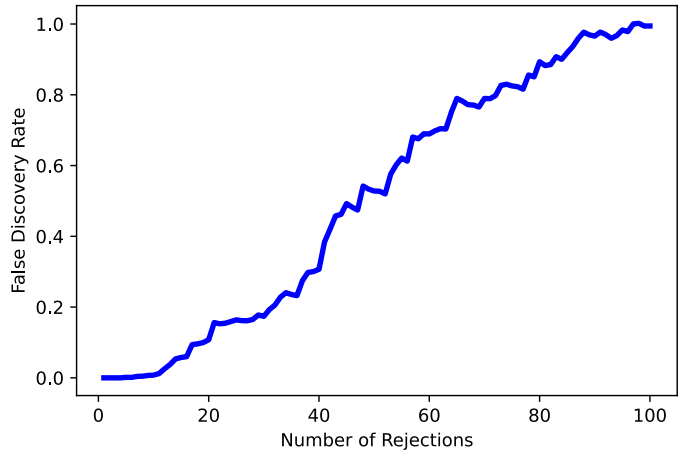

FIGURE 13.11. *The estimated false discovery rate versus the number of rejected null hypotheses, for 100 genes randomly selected from the* `Khan` *dataset.*

13.7 Exercises

Conceptual

1. Suppose we test m null hypotheses, all of which are true. We control the Type I error for each null hypothesis at level α. For each subproblem, justify your answer.

   (a) In total, how many Type I errors do we expect to make?

   (b) Suppose that the m tests that we perform are independent. What is the family-wise error rate associated with these m tests?

   Hint: If two events A and B are independent, then $\Pr(A \cap B) = \Pr(A)\Pr(B)$.

   (c) Suppose that $m = 2$, and that the p-values for the two tests are positively correlated, so that if one is small then the other will tend to be small as well, and if one is large then the other will tend to be large. How does the family-wise error rate associated with these $m = 2$ tests qualitatively compare to the answer in (b) with $m = 2$?

   Hint: First, suppose that the two p-values are perfectly correlated.

   (d) Suppose again that $m = 2$, but that now the p-values for the two tests are negatively correlated, so that if one is large then the other will tend to be small. How does the family-wise error rate associated with these $m = 2$ tests qualitatively compare to the answer in (b) with $m = 2$?

   Hint: First, suppose that whenever one p-value is less than α, then the other will be greater than α. In other words, we can never reject both null hypotheses.

2. Suppose that we test m hypotheses, and control the Type I error for each hypothesis at level α. Assume that all m p-values are independent, and that all null hypotheses are true.

   (a) Let the random variable A_j equal 1 if the jth null hypothesis is rejected, and 0 otherwise. What is the distribution of A_j?

   (b) What is the distribution of $\sum_{j=1}^{m} A_j$?

   (c) What is the standard deviation of the number of Type I errors that we will make?

3. Suppose we test m null hypotheses, and control the Type I error for the jth null hypothesis at level α_j, for $j = 1, \ldots, m$. Argue that the family-wise error rate is no greater than $\sum_{j=1}^{m} \alpha_j$.

Null Hypothesis	p-value
H_{01}	0.0011
H_{02}	0.031
H_{03}	0.017
H_{04}	0.32
H_{05}	0.11
H_{06}	0.90
H_{07}	0.07
H_{08}	0.006
H_{09}	0.004
H_{10}	0.0009

TABLE 13.4. *p-values for Exercise 4.*

4. Suppose we test $m = 10$ hypotheses, and obtain the p-values shown in Table 13.4.

   (a) Suppose that we wish to control the Type I error for each null hypothesis at level $\alpha = 0.05$. Which null hypotheses will we reject?

   (b) Now suppose that we wish to control the FWER at level $\alpha = 0.05$. Which null hypotheses will we reject? Justify your answer.

   (c) Now suppose that we wish to control the FDR at level $q = 0.05$. Which null hypotheses will we reject? Justify your answer.

   (d) Now suppose that we wish to control the FDR at level $q = 0.2$. Which null hypotheses will we reject? Justify your answer.

   (e) Of the null hypotheses rejected at FDR level $q = 0.2$, approximately how many are false positives? Justify your answer.

5. For this problem, you will make up p-values that lead to a certain number of rejections using the Bonferroni and Holm procedures.

   (a) Give an example of five p-values (i.e. five numbers between 0 and 1 which, for the purpose of this problem, we will interpret as p-values) for which both Bonferroni's method and Holm's method

reject exactly one null hypothesis when controlling the FWER at level 0.1.

(b) Now give an example of five p-values for which Bonferroni rejects one null hypothesis and Holm rejects more than one null hypothesis at level 0.1.

6. For each of the three panels in Figure 13.3, answer the following questions:

(a) How many false positives, false negatives, true positives, true negatives, Type I errors, and Type II errors result from applying the Bonferroni procedure to control the FWER at level $\alpha = 0.05$?

(b) How many false positives, false negatives, true positives, true negatives, Type I errors, and Type II errors result from applying the Holm procedure to control the FWER at level $\alpha = 0.05$?

(c) What is the false discovery proportion associated with using the Bonferroni procedure to control the FWER at level $\alpha = 0.05$?

(d) What is the false discovery proportion associated with using the Holm procedure to control the FWER at level $\alpha = 0.05$?

(e) How would the answers to (a) and (c) change if we instead used the Bonferroni procedure to control the FWER at level $\alpha = 0.001$?

Applied

7. This problem makes use of the `Carseats` dataset in the `ISLP` package.

(a) For each quantitative variable in the dataset besides `Sales`, fit a linear model to predict `Sales` using that quantitative variable. Report the p-values associated with the coefficients for the variables. That is, for each model of the form $Y = \beta_0 + \beta_1 X + \epsilon$, report the p-value associated with the coefficient β_1. Here, Y represents `Sales` and X represents one of the other quantitative variables.

(b) Suppose we control the Type I error at level $\alpha = 0.05$ for the p-values obtained in (a). Which null hypotheses do we reject?

(c) Now suppose we control the FWER at level 0.05 for the p-values. Which null hypotheses do we reject?

(d) Finally, suppose we control the FDR at level 0.2 for the p-values. Which null hypotheses do we reject?

8. In this problem, we will simulate data from $m = 100$ fund managers.

```
rng = np.random.default_rng(1)
n, m = 20, 100
X = rng.normal(size=(n, m))
```

These data represent each fund manager's percentage returns for each of $n = 20$ months. We wish to test the null hypothesis that each fund manager's percentage returns have population mean equal to zero. Notice that we simulated the data in such a way that each fund manager's percentage returns do have population mean zero; in other words, all m null hypotheses are true.

(a) Conduct a one-sample t-test for each fund manager, and plot a histogram of the p-values obtained.

(b) If we control Type I error for each null hypothesis at level $\alpha = 0.05$, then how many null hypotheses do we reject?

(c) If we control the FWER at level 0.05, then how many null hypotheses do we reject?

(d) If we control the FDR at level 0.05, then how many null hypotheses do we reject?

(e) Now suppose we "cherry-pick" the 10 fund managers who perform the best in our data. If we control the FWER for just these 10 fund managers at level 0.05, then how many null hypotheses do we reject? If we control the FDR for just these 10 fund managers at level 0.05, then how many null hypotheses do we reject?

(f) Explain why the analysis in (e) is misleading.
Hint: The standard approaches for controlling the FWER and FDR assume that all *tested null hypotheses are adjusted for multiplicity, and that no "cherry-picking" of the smallest p-values has occurred. What goes wrong if we cherry-pick?*