An introduction to statistical data analysis (Summer 2014)
Lecture notes

Taught by Shravan Vasishth [vasishth@uni-potsdam.de]

Last edited: May 9, 2014

```
> sessionInfo()

R version 3.0.2 (2013-09-25)
Platform: x86_64-apple-darwin10.8.0 (64-bit)

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] datasets  utils     stats     graphics  grDevices methods   base

other attached packages:
[1] MASS_7.3-29

loaded via a namespace (and not attached):
[1] tools_3.0.2
```

Contents

Chapter 1

What this course is about

This is a graduate level course in linguistics that introduces statistical data analysis to people who have presumably never done any data analysis before. Only high school pre-calculus mathematics is presupposed, and even there not much is needed beyond basic math skills like addition, subtraction, multiplication, and division.

The goal of this course is to prepare students to understand and use the most commonly deployed statistical models in psycholinguistics. The course is designed to bring people to terms with the linear mixed model framework. We ignore ANOVA in this course because there is not enough time to cover it. We also limit the discussion to two commonly used distributions: the binomial and normal distributions.

The most frequent question people tend to have in this class is: **why do I need to study all this stuff**? The short answer is that linguistics is now a heavily experimental science, and one cannot function in linguistics any more without at least a basic knowledge of statistics. Because time is short in this course, I decided to drastically limit the scope of the course, so that we cover only a small number of topics; these will be the most frequently used tools in linguistics.

By the end of the course you should know the following:

- **Basic** usage of the R language for data analysis.
- Basic understanding of the logic of significance testing and hypothesis testing.
- The meaning of confidence intervals, p-values, z- and t-values, Type I and II error probability, Power.
- Linear models (including simple multiple regression), basic contrast coding.
- Basics of fitting linear mixed models and presenting results.

1.1 Quiz: Do you need this course?

You should take this quiz on your own to decide whether you need this course. If you can answer (almost) all the questions correctly, you are in pretty good shape. If you made more than one mistake or don't know the answer to more than one question, you should probably do this course. The solutions are at the end of the book.

Instructions: choose only one answer by circling the relevant letter. If you don't know the answer, just leave the answer blank.

1. Standard error is

   a the standard deviation of the sample scores
   b the standard deviation of the distribution of sample means
   c the square root of the sample variance
   d 2 times the standard deviation of sample scores

2. If we sum up the differences of each sample score from the sample's mean (average) we will always get

   a a large number
   b the number zero
   c a different number each time, sometimes large, sometimes small
   d the number one

3. As sample size increases, the standard error of the sample should

   a increase
   b decrease
   c remain unchanged

4. The 95% confidence interval tells you

   a that the probability is 95% that the population mean is equal to the sample mean
   b that the sample mean lies within this interval with probability 95%
   c that the population mean lies within this interval with probability 95%
   d none of the above

5. The 95% confidence interval is roughly equal to

   a 0.5 times the standard error
   b 1 times the standard error
   c 1.5 times the standard error
   d 2 times the standard error

6. The 95% confidence interval is — the 90% confidence interval

   a wider than
   b narrower than
   c same as

7. A p-value is

   a the probability of the null hypothesis being true
   b the probability of the null hypothesis being false
   c the probability of the alternative hypothesis being true
   d the probability of getting the sample mean that you got (or a value more extreme) assuming the null hypothesis is true
   e the probability of getting the sample mean that you got (or a value less extreme) assuming the null hypothesis is true

8. If Type I error probability, alpha, is 0.05 in a t-test, then

   a we have a 5% probability of rejecting the null hypothesis when it is actually true
   b we have a 95% probability of rejecting the null hypothesis when it is actually true
   c we necessarily have low power
   d we necessarily have high power

9. Type II error probability is

   a the probability of accepting the null when it's true
   b the probability of accepting the null when it's false
   c the probability of rejecting the null when it's true
   d the probability of rejecting the null when it's false

10. When power increases

   a Type II error probability decreases
   b Type II error probability increases
   c Type II error probability remains unchanged

11. If we compare two means from two samples, and the p>0.05 (p is greater than 0.05), we can conclude

   a that the two samples comes from two populations with different means
   b that the two samples comes from two populations with identical means
   c that we don't know whether two samples comes from two populations with identical means or not

1.2 Some basic knowledge before we start

In this course we are always going to be interested in estimating mean values and in quantifying our uncertainty about the accuracy of the estimate; an example is reading time: we are often interested in knowing if one kind of sentence is read faster than another kind of sentence. When we do an experiment, we obtain behavioral measures for each participant, and then we estimate means (e.g., mean reading time), and our certainty about these means. This leads us to the part

that affects the science: inference. From these estimates, we want to infer what is true or false about the world.

Given a sample of some dependent variable values $x_1, \ldots, x_n$, the mean, written $\bar{x}$ can be calculated using the formula:

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1.1}$$

Example:

```
> x<-1:10
> mean(x)

[1] 5.5
```

We can also quantify how much each individual value x_i deviates on average from the mean value $\bar{x}$. This is called the variance, and its square root is called standard deviation or SD. All this should be familiar from school.

$$s^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \cdots + (x_n - \bar{x})^2}{n-1} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{1.2}$$

Example:

```
> ## variance:
> var(x)

[1] 9.1667

> ## its square root:
> sd(x)

[1] 3.0277
```

Sometimes you will see the above formula with division by n rather than $n-1$. For our purposes, the distinction is uninteresting (but I can explain in class if there is interest). Note that for large n, it is not going to matter much whether you divide by n or $n-1$. The book by Kerns [4] is a good reference for the curious, and those of you who want to get deeper into this subject.

So what we will generally start with is a measure of central tendency and a measure of variability. These two numbers, mean and variance (or standard deviation), are useful for a particular case where the distribution of values that we have sampled has a particular shape. This is the bell-shaped curve, known as the Gaussian distribution or the normal distribution. Many measurements in the world have a roughly normal distribution, i.e., they can be accurately characterized if we know the mean and variance. I make this more precise in the next chapter.

In essence, we are in the business of figuring out what the world looks like; specifically, in this course we are going to be constantly trying to figure out what is the true mean and variance of some **unknown** distribution of values. The word unknown is key here. We are trying to **estimate** the **parameters** of some underlying distribution that is assumed to have generated the data, and we are trying to draw conclusions about the world from this estimate. *Basically, this is all this*

course is about. It might seem boring and irrelevant to your life, but basically this estimation and inference procedure runs pretty much every aspect of your existence, both in academia and outside. So, if you feel resistance against learning about these tools, think about the fact that without these tools you cannot do science (I am assuming that the desire to do research is what brings you to Potsdam).

Once we know the mean and variance from a sample, we are ready to so some inference. The theory of statistics is a vast and exciting area, and there is much, much more out there than I am willing to discuss in this course. But the little I will discuss will prepare you for the most common situations we encounter in linguistics.

1.3 How to survive and perhaps even enjoy this course

I have been teaching this course for several years now, and one reaction that I get quite often is fear, panic, and even anger. A common reaction is: Why do I have to learn all this? How can I do all this programming? And so on.

If you have such questions popping up in your mind, you have to stop and consider a few things before continuing with this course. Linguistics at Potsdam is a very empirically driven program. It is impossible to get through a master's degree in linguistics without coming into contact with data, even in formerly armchair disciplines like syntax. If you are at Potsdam, you are automatically committed to an empirically driven education.

More broadly, there is a widespread misunderstanding that statistics is something that can be outsourced to a statistician. It's true that if you have a non-standard statistical problem you probably need to talk to a professional. But for the kinds of methods used in linguistics, you are personally responsible for the analyses you do, and so you are going to have to learn something about the methods. The fundamental thing to understand is that the statistics *is* the science, it is not an add-on.

Now, regarding the panic issue. In order to pass this course, you have to understand that **you have to read the lecture notes**, and that it is not enough to just passively *read* these lecture notes. You have to play with the ideas by asking yourself questions like "what would happen if…", and then check the answer right there using R. That's the whole point of this approach to teaching statistics, that you can verify what happens under repeated sampling. There is no point in memorizing formulas; focus on developing understanding. The concepts presented here require nothing more than middle school mathematics. The ideas are not easy to understand, but simulation is a great way to develop a deeper understanding of the logic of statistical theory.

Many students come in expecting to get an A in this course. It's possible to get an A, if you read the lecture notes and understand them. However, in general, such students need to understand that they need to learn to make mistakes, and to use these mistakes as a learning tool. If you lose marks, it is not a personal insult; it is rather a very useful message telling you that you need to think about the material again.

1.4 Installing R and learning basic usage

You should google the word CRAN and RStudio and install R and RStudio on your machine. We will explain basic R uage in class; also see the introductory notes on using R released on Moodle. You should spend some time on the CRAN website looking at the information available on R there. Look especially at the section called Contributed (navigation panel on the left on the main page).

Chapter 2

Sampling distribution of sample mean

2.1 The normal distribution

A typical situation in linguistic research involves collecting reading times or reaction times in a particular experiment to answer a specific research question. For example, I might want to know whether subject relatives (SRs) are read faster than object relatives (ORs) in a population of speakers (say German speakers). To find this out, I would get randomly selected participants to read SRs and ORs; the details of experiment design will be discussed later. Right now, all that matters is that if I do such an experiment, I will get a difference in reading times between SRs and ORs for each participant. Suppose I have 100 participants, and I know the difference in OR vs SR reading times, in seconds. We can simulate this situation in R. We could have 100 data points, each representing a difference in means between subject and object relatives seen by each subject (synonymous here with participant).

```
> x<-rnorm(100)
> head(x)

[1]  0.75818 -0.17112  0.40004  1.19581 -0.80621 -0.13832
```

This is a (simulated) **sample**; we could have taken a different sample from the population of German speakers. You can simulate that by running the above code again.

The sample we did take comes from a **population** of speakers. Note that, for theoretical reasons we won't go into, it is important to take a **random sample**. In practice, this is not really the case—we just take the university students who are willing to come do the experiment. This is likely to introduce a bias in our results, in that the result is probably going to not be representative of the population. As far as I know, nobody worries about this problem (but maybe we should).

As mentioned above, each value in the above simulated sample represents one participant's response. A positive value means that the OR was read more slowly than the SR. If we plot the distribution of this sample's values, then we will see that it has roughly a "bell-shaped distribution":

Notice that most of the values are centered around 0, some are as big as 2, and some are as small as -3.

It turns out that a lot of very convenient theory can be built around this distribution. Since normal distribution theory is so fundamental to what we do in linguistics, I am going to focus on better understanding this one distribution.

```
> ## plot density histogram:
> hist(x,freq=F)
```

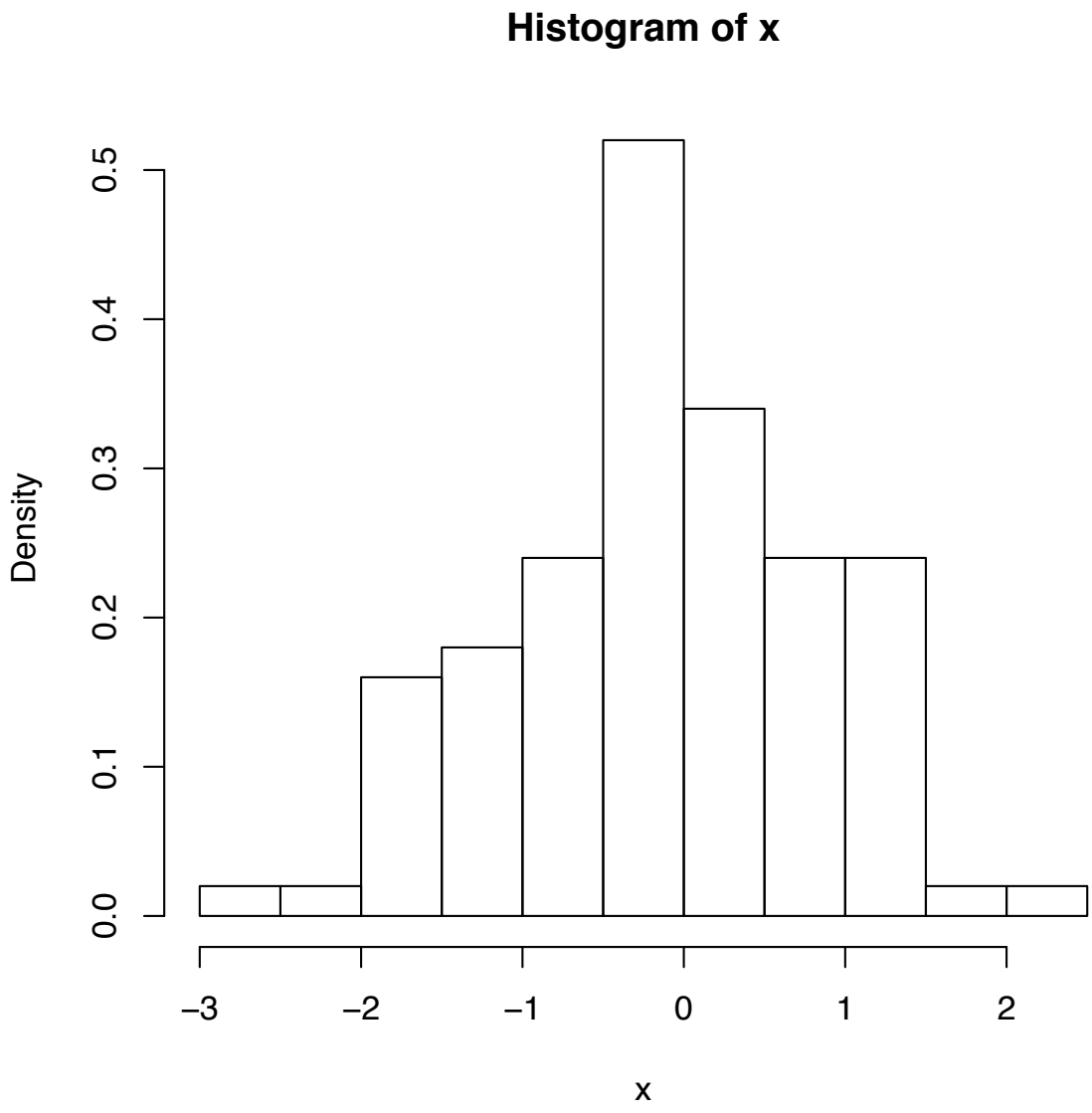


Figure 2.1: A histogram of the sample data.

The following function is defined as the normal density function:

$$f(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-((x-\mu)^2/2\sigma^2)} \tag{2.1}$$

Given a range of values for x, and specific values for μ, and σ, we can plot the result of applying this function. Since the function is defined by two values or **parameters**, we can write it in shorthand as follows: $N(\mu,\sigma)$, i.e., a normal distribution with some mean and some standard deviation. Statisticians usually write $N(\mu,\sigma^2)$, i.e., they use the variance rather than the standard deviation; but we will ignore that convention in this course, because in R we define the density function in terms of standard deviation. (But you should keep in mind that statisticians tend to define the normal distribution in terms of variance.)

We can define the density function in R as follows, setting mu and sigma to 0 and 1 respectively, for convenience (you could have set it to anything):

```
> ## mean and sigma set at 0 and 1 by default:
> normal.density.function <- function(x,mu=0,sigma=1){
    1/(sqrt(2*pi)*sigma)*exp(-((x - mu)^2/(2*sigma^2)))}
```

You can plot the shape of this distribution using the following command:

```
plot(function(x) normal.density.function(x), -3, 3,
      main = "Normal density function",ylim=c(0,.4),
             ylab="density",xlab="X")
```

R has a built-in function, `dnorm` that does the job of the function we defined above; we could just have used that built-in function:

```
> plot(function(x) dnorm(x), -3, 3,
         main = "Normal density",ylim=c(0,.4),
                ylab="density",xlab="X")
```

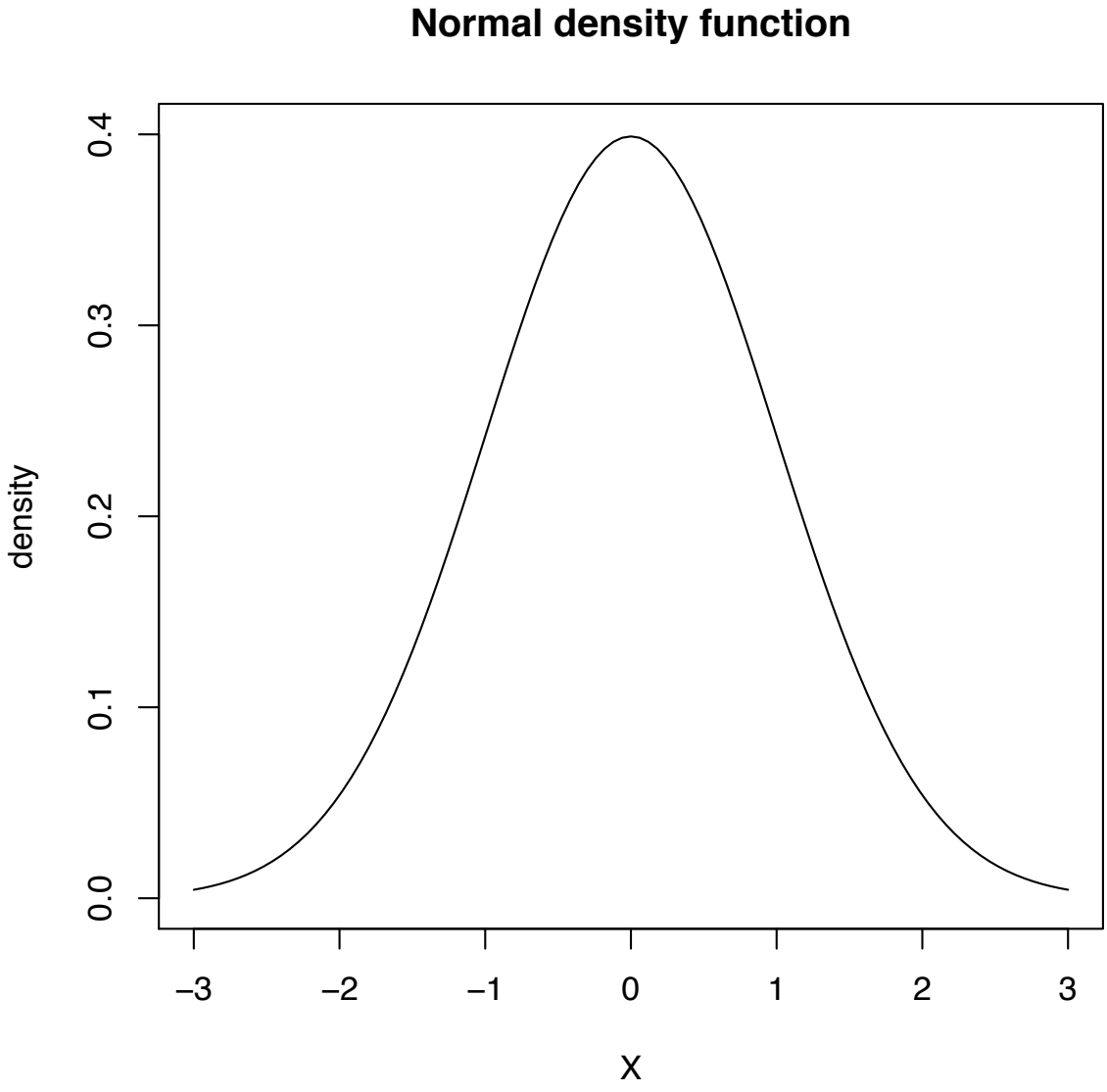
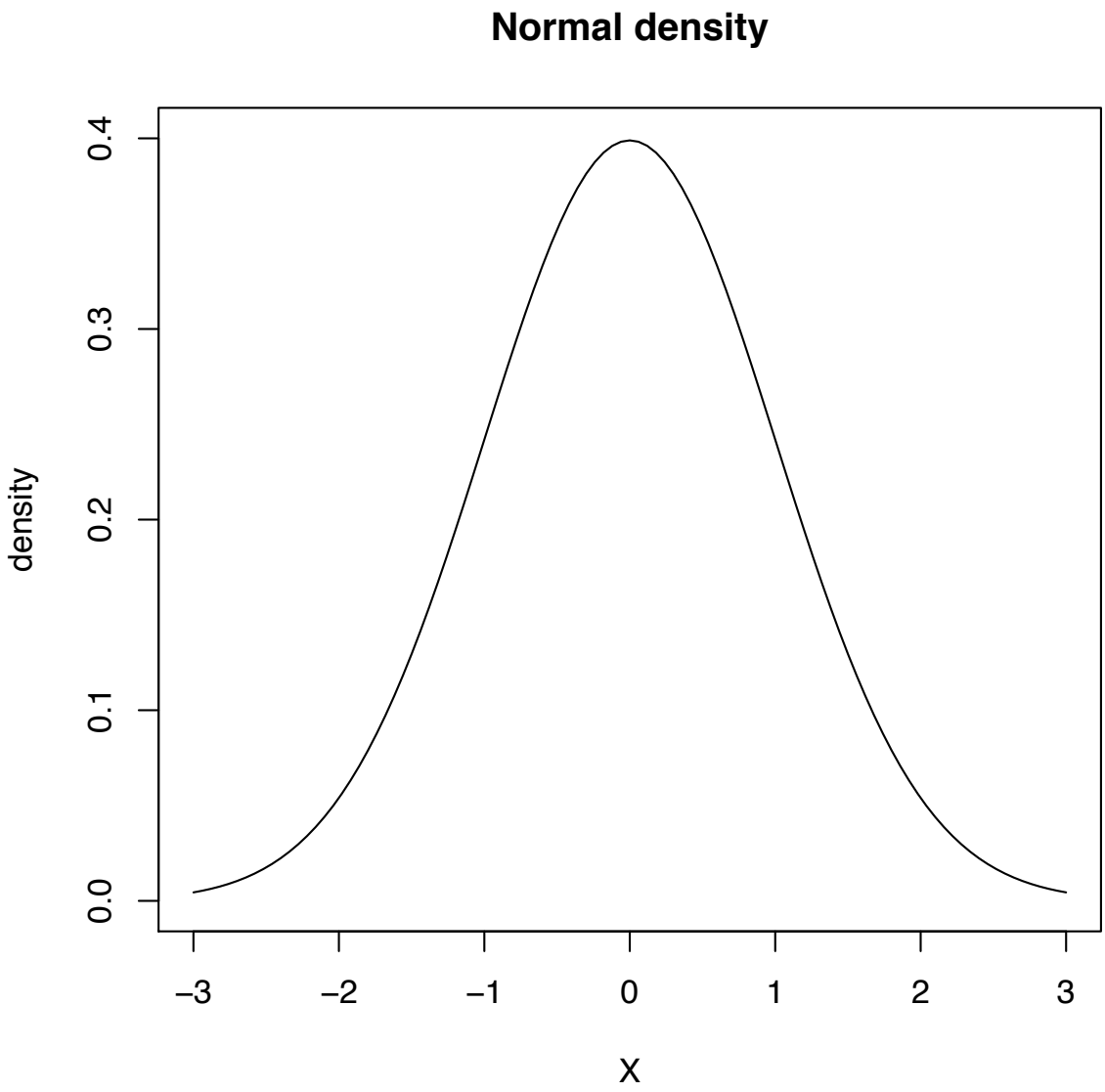


Figure 2.2: Plotting the normal distribution.



One important property of this function is that it stretches from -Infinity to +Infinity. We don't display this on the plot, hopefully it is obvious why not. Another important property is that it represents the probability of each of the x-axis values, and so the total probability of all possible values, stretching from all the way from -Infinity to +Infinity will be 1. You can calculate the total probability by summing up all the probabilities of all possible values. The function `integrate` does that summation for you:

```
> integrate(function(x) dnorm(x, mean = 0, sd = 1), -Inf, +Inf)

1 with absolute error < 9.4e-05
```

This will be our only brush with calculus in this course. The key point here is that it allows us to calculate the area under the curve given any lower and upper bound. For example, I could calculate the area under the curve between -2 and +2:

```
> integrate(function(x) dnorm(x, mean = 0, sd = 1), -2, +2)

0.9545 with absolute error < 1.8e-11
```

Why am I talking about calculating the area under the curve? It turns out we need this capability a lot in statistical data analysis, as you are about to discover in this lecture.

2.2 The area under the curve in a normal distribution

We begin by establishing a fundamental fact about any normal distribution: 95% of the probability lies within approximately 2 standard deviations (SDs) from the mean. If we sum the area under these curves, between 2 SD below the mean and 2 SD above the mean, we find the following areas, which correspond to the amount of probability within these bounds.

We can display this fact graphically (see Figure 2.3):

```
> ## plot multiple figures:
> ## replace ugly par... specification with
> ## something easier to remember:
> multiplot <- function(row,col){
        par(mfrow=c(row,col),pty="s")
      }
> main.title<-"Area within 2 SD of the mean"
> multiplot(1, 2)
> plot(function(x) dnorm(x, mean = 0, sd = 1),
   xlim=c(-3, 3),main="SD 1",xlab="x",ylab="",cex=2)
> segments(-2, 0, -2, 0.4)
> segments(2, 0, 2, 0.4)
> plot(function(x) dnorm(x, mean = 0, sd = 4),
   xlim=c(-12, 12),main="SD 4",xlab="x",ylab="",cex=2)
> segments(-8, 0, -8, 0.1)
> segments(8, 0, 8, 0.1)
```

There is a built-in R function, `pnorm`, for computing probabilities within a range; (so we don't really need the integrate function, I just used it initially to show that we are really doing a summation over continuous values). Here are some examples of `pnorm` in action. Here, we use the default values of `pnorm` for mean and standard deviation, but you could have had any mean or standard deviation:

```
> ## Prob. of getting 2 or less:
> pnorm(2)

[1] 0.97725

> ## Prob. of getting more than 2:
> 1-pnorm(2)

[1] 0.02275
```

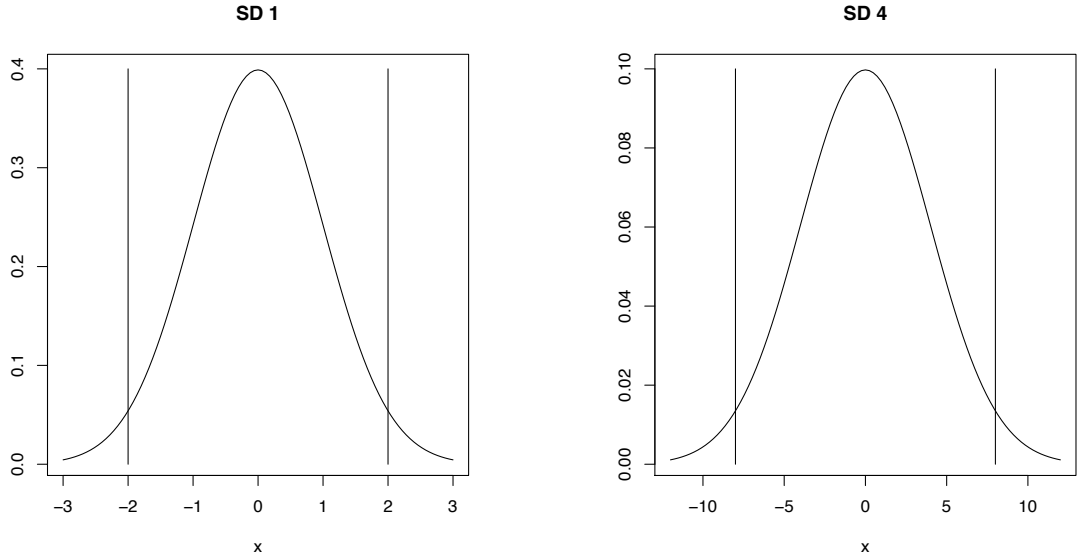


Figure 2.3: Two normal distributions with SD = 1 (left), SD = 4 (right). The lines delimit the region 2 SD from the mean in each case.

```
> ## Prob. of getting -2 or less:
> pnorm(-2)

[1] 0.02275

> ## Prob. of being between -2 and 2:
> pnorm(2)-pnorm(-2)

[1] 0.9545
```

You will sometimes need to know the following: given a normal distribution with a particular mean and standard deviation, what is the boundary marking x% of the area under the curve (usually centered around the mean value). For example, the command `pnorm(2)-pnorm(-2)` gives us the area between -2 and 2 in a normal distribution with mean 0 and sd=1, and the area is 0.9545 (I will freely switch between percentages and proportions for probability; don't get confused!). Suppose we only knew the area (probability) that we want to have under the curve, and want to know the bounds that mark that area. We actually know that the bounds are -2 and 2 here, but we will pretend we don't know and need to find this out. How to do this? Grade 3 arithmetic comes to the rescue: The total area under the curve is 1. We want the lower and upper bounds for the area 0.9545. This means that

$$1 - 0.9545 = 0.0455 \tag{2.2}$$

is the area outside the (as yet unknown) bounds. Since the normal density curve is symmetrical, that means that each of the two sides outside the boundaries we want to discover has area

$$\frac{0.0455}{2} = 0.02275 \tag{2.3}$$

So: there is some lower bound with area 0.02275 to the **left** of it (i.e., in the lower tail), and some upper bound with area 0.02275 to the **right** of it (i.e., in the upper tail). We are pretending right now that we don't know that lower=-2 and upper=2; we are engaging in this pretence because we will be in situations soon where we don't know these values and have to discover them given some probability range. R allows you to ask: "what is the bound, for a given distribution, such that the probability to the left of it is some value p1, or what is the bound such that the probability to the right of it is some value p2?" The function that gives this answer is called `qnorm`, and here is how we can use to answer our current question:

```
> ## figure out area between the unknown bounds:
> prob<-round(pnorm(2)-pnorm(-2),digits=4)
> ## figure out lower bound:
> (lower<-qnorm((1-prob)/2,mean=0,sd=1,lower.tail=T))
```

```
[1] -2
```

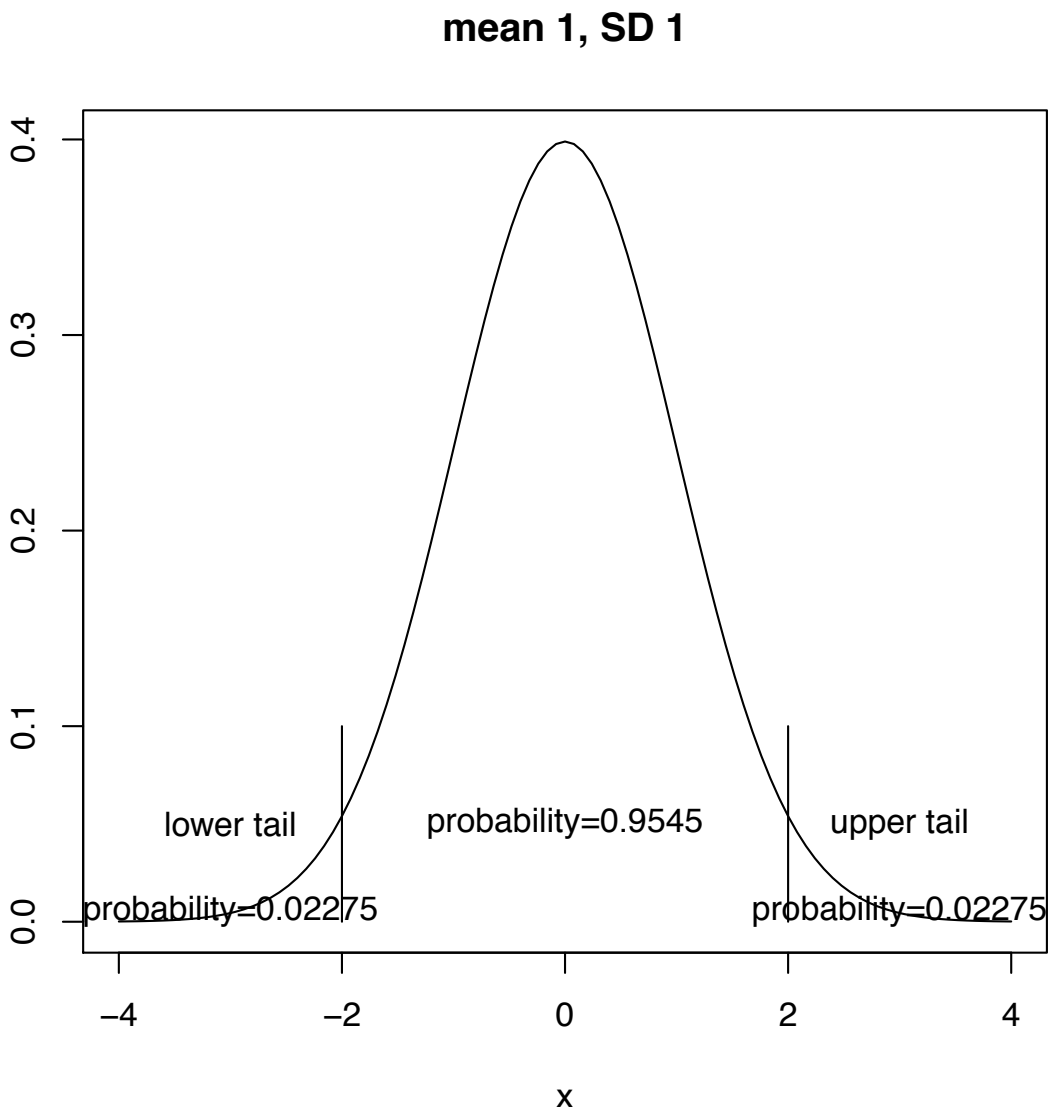
```
> ## figure out upper bound:
> (upper<-qnorm((1-prob)/2,mean=0,sd=1,lower.tail=F))
```

```
[1] 2
```

And so we discover what we expected: the lower bound is -2 and the upper bound is 2.

It **always** helps to visualize what we are doing:

```
> plot(function(x) dnorm(x, mean = 0, sd = 1),
   xlim=c(-4, 4),main="mean 1, SD 1",xlab="x",ylab="",cex=2)
> segments(-2, 0, -2, 0.1)
> segments(2, 0, 2, 0.1)
> text(0,.05,"probability=0.9545")
> text(-3,.05,"lower tail")
> text(-3,.005,"probability=0.02275")
> text(3,.05,"upper tail")
> text(3,.005,"probability=0.02275")
```



The skills we just learnt give us tremendous capability, as you will see in the rest of this chapter.

2.3 Repeated sampling

Suppose now that we have a population of people and that we know the age of each individual; let us assume also that distribution of the ages is approximately normal. Finally, let us also suppose that we know that mean age of the population is 60 years and the population SD is 4 years.

Now suppose that we **repeatedly** sample from this population: we take samples of 40, a total of 1000 times; and we calculate the mean $\bar{x}$ each time we take a sample. After taking 1000 samples, we have 1000 means; if we plot the distribution of these means, we have the **sampling distribution of the sample mean**.

```
> #1000 samples of 40 taken repeatedly:
> sample.means <- rep(NA,1000)
> for(i in 1:1000){
     sample.40 <- rnorm(40,mean=60,sd=4)
     sample.means[i] <- mean(sample.40)
   }
```

We can calculate the mean and standard deviation of this sampling distribution:

```
> means40<-mean(sample.means)

[1] 60.018

> sd40<-sd(sample.means)

[1] 0.62813
```

If we plot this distribution of means, we find that it is roughly normal.

```
> hist(sample.means)
```

We can characterize the distribution of means visually, as done in Figure 2.4 below, or in terms of the mean and standard deviation of the distribution. The mean value in the above simulation is 60.02 and the standard deviation of the distribution of means is 0.6281. Note that if you repeatedly run the above simulation code, these numbers will differ slightly in each run.

Consider now the situation where our sample size is 100. Note that the mean and standard deviation of the population ages is the same as above.

```
> sample.means <- rep(NA,1000)
> for(i in 1:1000){
     sample.100 <- rnorm(100,mean=60,sd=4)
     sample.means[i] <- mean(sample.100)
   }
>

> means100 <- mean(sample.means)

[1] 60.009

> sd100 <- sd(sample.means)

[1] 0.41301
```

In this particular simulation run, the mean of the means is 60 and the standard deviation of the distribution of means is 0.413. Let's plot the distribution of the means (Figure 2.5).

```
> hist(sample.means)
```

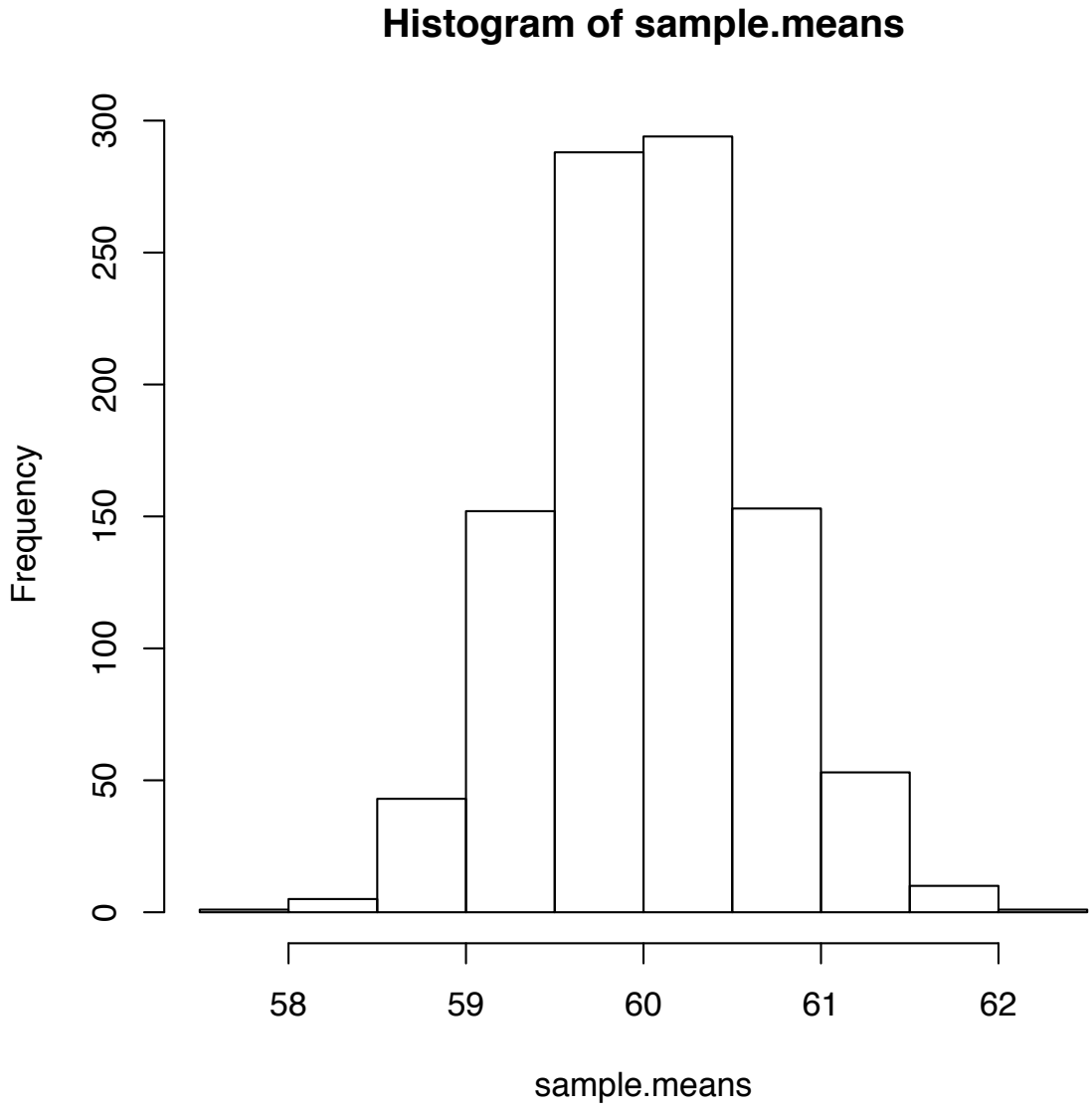


Figure 2.4: The sampling distribution of the sample mean with 1000 samples of size 40.

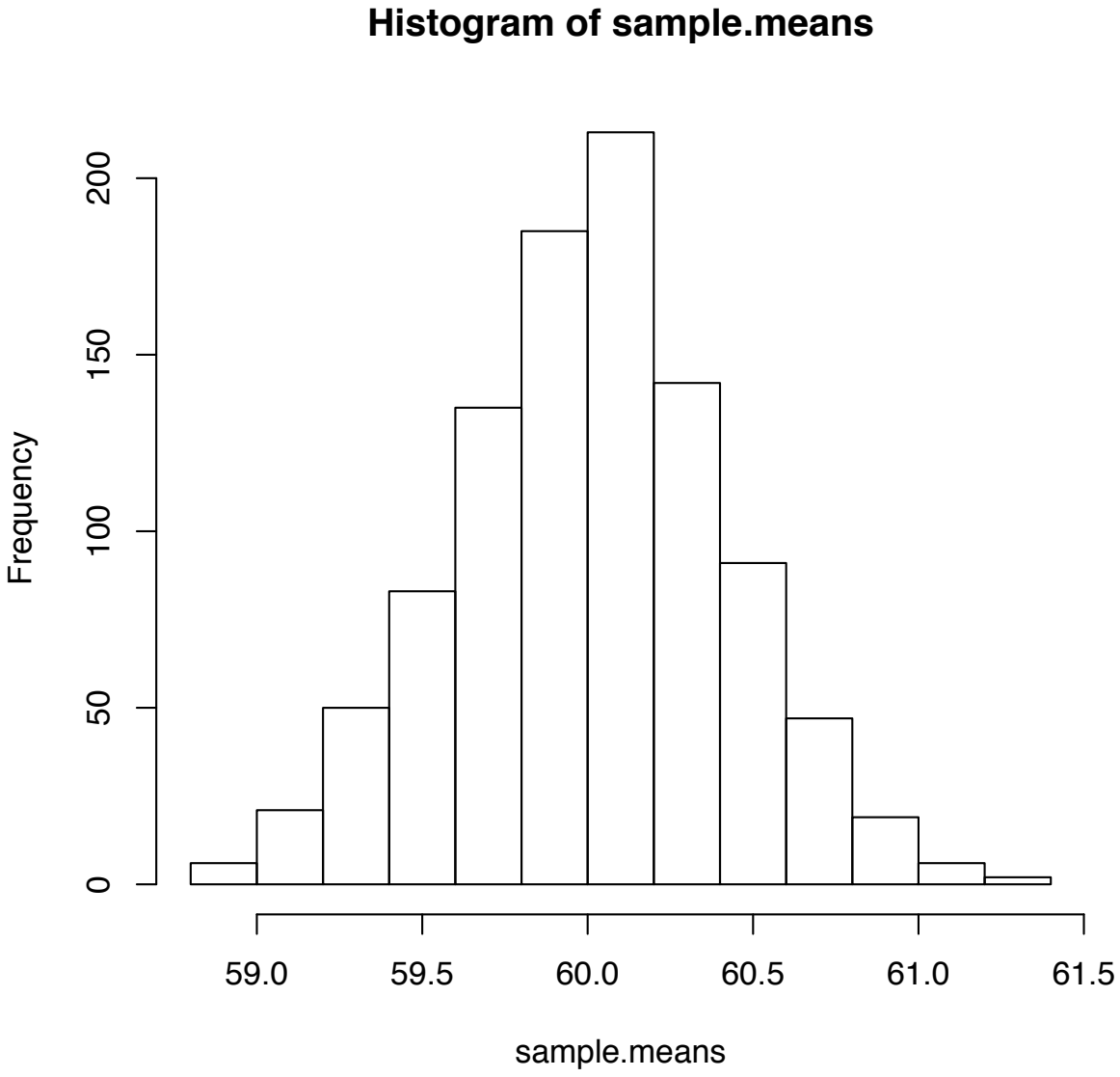


Figure 2.5: The sampling distribution of the sample mean with samples of size 100.

The above simulations show us several things. First, the standard deviation of the distribution of means gets smaller as we increase sample size. When the sample size is 40, the standard deviation is 0.6281; when it is 100, the standard deviation is 0.413. Second, as the sample size is increased, the mean of the sample means becomes a better and better estimate of the *population* mean $\mu_{\bar{x}}$. A third point (which is not obvious at the moment) is that there is a lawful relationship between the standard deviation σ of the population and the standard deviation of the *distribution of means*, which we will call $\sigma_{\bar{x}}$. This relationship is:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}} \tag{2.4}$$

Here, n is the sample size. It is possible to derive equation 2.4 from first principles, but for that we need a bit more theory, which won't cover in this course (see Kerns). Here, we simply note the important point that n is in the denominator in this equation, so there is an inverse relationship between the sample size and the standard deviation of the sample means. Let's take this equation on trust for the moment and use it to compute $\sigma_{\bar{x}}$ by using the population standard deviation (which we assume we know). Let's do this for a sample of size 40 and another of size 100:

```
> 4/sqrt(40)

[1] 0.63246

> 4/sqrt(100)

[1] 0.4
```

The above calculation is consistent with what we just saw: $\sigma_{\bar{x}}$ gets smaller and smaller as we increase sample size.

We have also introduced a notational convention that we will use throughout the notes: sample statistics are symbolized by Latin letters ($\bar{x}, s$); population parameters are symbolized by Greek letters (μ, σ).

2.4 The Central Limit Theorem

We will see now that the *sampling distribution of the sample mean* is also normally distributed. In the above example the means were drawn from a population with normally distributed scores. It turns out that the sampling distribution of the sample mean will be normal even if the population is not normally distributed, as long as the sample size is large enough and the distribution we are sampling from has a mean. This is known as the Central Limit Theorem:

> When sampling from a population that has a mean, provided the sample size is large enough, the sampling distribution of the sample mean will be close to normal regardless of the shape of the population distribution.

[Note: Note the caveat "When sampling from a population that has a mean". There are some distributions which do not have a mean; but in this course we will ignore these. More advanced textbooks on probability discuss these distributions.]

Let's check whether this theorem holds by testing it in a case where our population is not normally distributed. Let's take our samples from a population (Figure 2.6) whose values are distributed exponentially with the same mean of 60 (the mean of an EXPONENTIAL DISTRIBUTION is the reciprocal of the so-called 'rate' parameter).

```
> sample.100 <- rexp(100, 1/60)
> hist(sample.100)
```

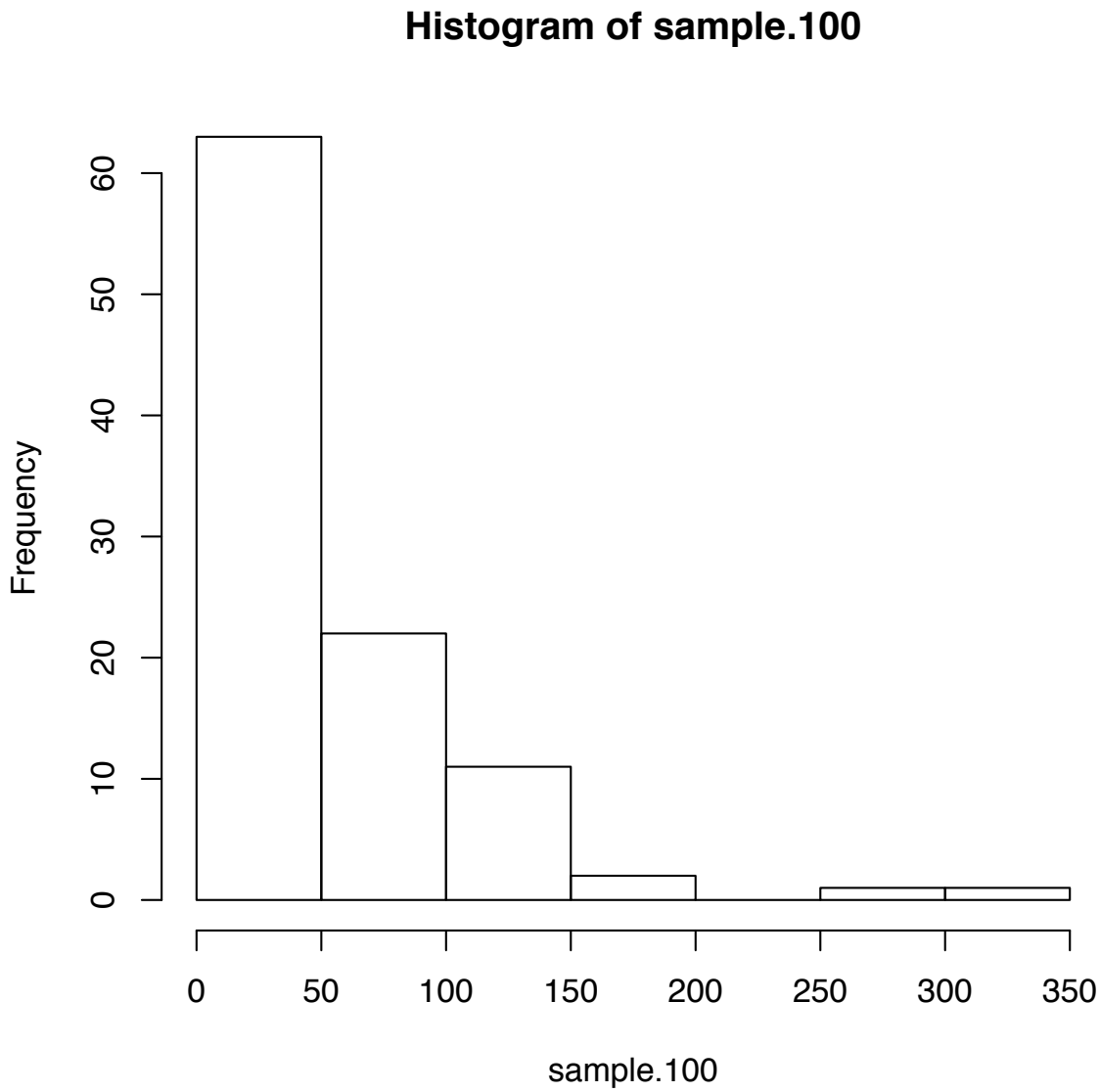


Figure 2.6: A sample from exponentially distributed population scores.

Now let us plot the sampling distribution of the sample mean. We take 1000 samples of size 100 each from this exponentially distributed population. As shown in Figure 2.7, the distribution of the means is again (more or less) normal.

```
> sample.means <- rep(NA,1000)
> for(i in 1:1000){
```

```
        sample.100 <- rexp(100, 1/60)
        sample.means[i] <- mean(sample.100)
  }
> hist(sample.means)
```

Recall that the mean of each sample is a point estimate of the true mean of the population. Some of these samples will have a mean slightly above the true mean, some slightly below, and the sampling distribution of *these* values is roughly normal. Try altering the sample size in this example to get a feel for what happens if the sample size is not 'large enough.'

To summarize:

1. The sampling distribution of the sample mean is normal for large sample sizes.
2. The mean of the sampling distribution of the sample mean is (in the limit) the same as the population mean.
3. It follows from the above two facts that the mean of a sample is a good estimate of the population mean.

2.5 σ and $\sigma_{\bar{x}}$

We saw earlier that the standard deviation of the sampling distribution of the sample mean $\sigma_{\bar{x}}$ gets smaller as we increase sample size. When the sample has size 40, this standard deviation is 0.6281; when it is 100, this standard deviation is 0.413.

Let's study the relationship between $\sigma_{\bar{x}}$ and σ. Recall that our population mean $\mu = 60$, $\sigma = 4$. The equation below summarizes the relationship; it shouldn't surprise you, since we just saw it above:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}} \tag{2.5}$$

But note also that the "tighter" the distribution, the lower the variance about the true population mean. So the $\sigma_{\bar{x}}$ is an indicator of how good our estimate of the population mean is. As we increase the size of a single sample, the smaller the standard deviation of its corresponding sampling distribution becomes. This brings us to the 95% confidence interval.

2.6 The 95% Confidence Interval for the Sample Mean

Let's take a sample of 11 ages from a normally distributed population with known mean age $\mu = 60$ years and SD $\sigma = 4$ years.

```
> sample.11 <- rnorm(11,mean=60,sd=4)

 [1] 62.931 72.455 59.282 67.740 60.397 58.549 59.109 60.032 64.563 57.480
[11] 51.247
```

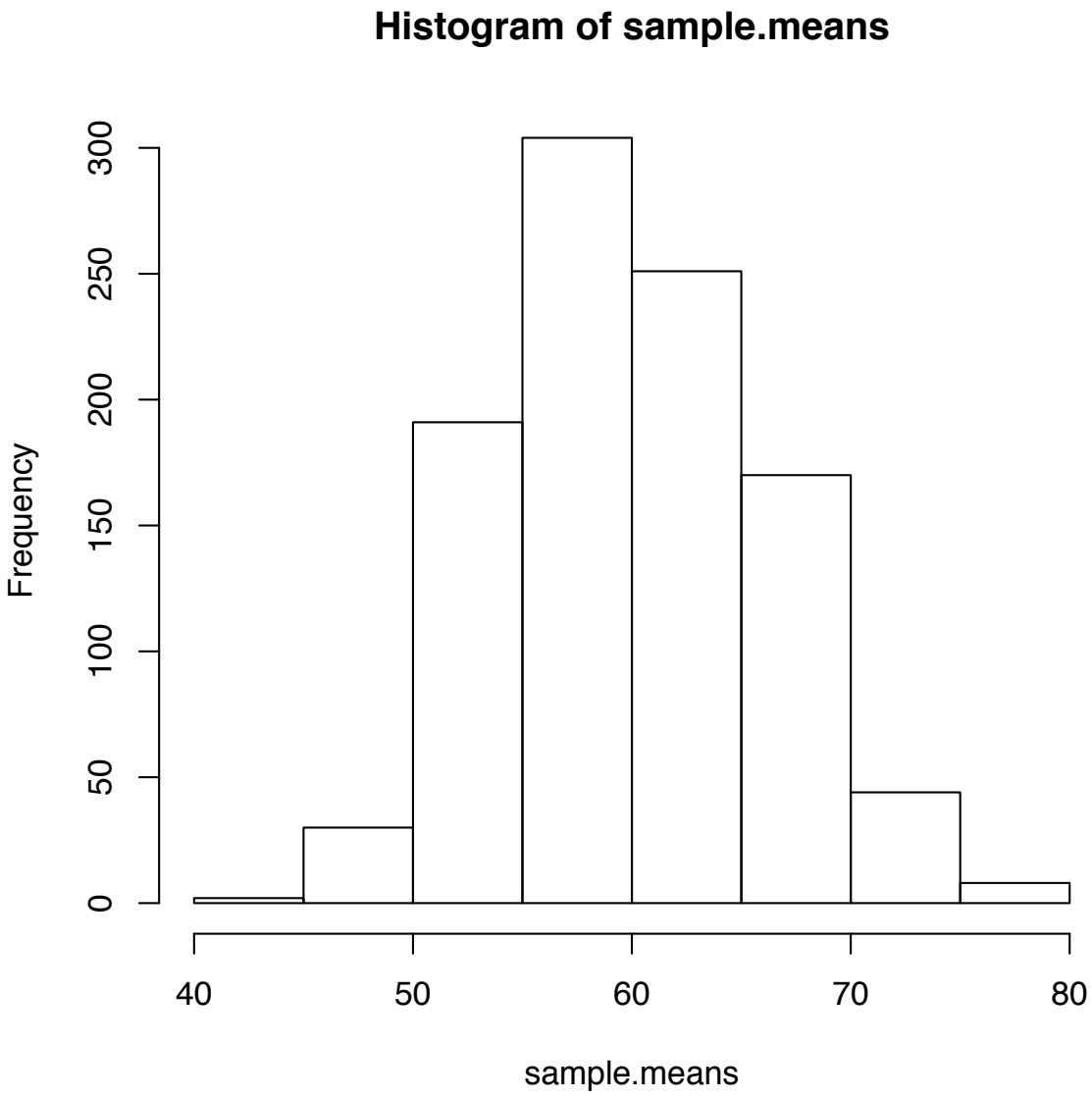


Figure 2.7: The sampling distribution of sample mean from an exponentially distributed population.

We know the mean here, but let's pretend we don't. Let's estimate a population mean from this sample using the sample mean $\bar{x}$, and compute the SD $\sigma_{\bar{x}}$ of the corresponding sampling distribution. Since we know the true population standard deviation we can get a precise value for $\sigma_{\bar{x}}$. We don't need to estimate the SD or the $\sigma_{\bar{x}}$. So, we have an estimate of the true mean, but we know the exact $\sigma_{\bar{x}}$.

```
> estimated.mean <- mean(sample.11)

[1] 61.253

> SD.population <- 4

[1] 4

> n <- length(sample.11)

[1] 11

> SD.distribution <- SD.population/sqrt(n)

[1] 1.206
```

We know from the Central Limit Theorem that the sampling distribution of the sample mean is roughly normal, and we know that in this case $\sigma_{\bar{x}} = 1.2$. Note that if we repeatedly sample from this population, our sample mean will change slightly each time, but the $\sigma_{\bar{x}}$ is not going to change. Why is that?

It turns out that the probability that the population mean is within $2 \times \sigma_{\bar{x}}$ of the sample mean is a bit over 0.95. Let's calculate this range, $2 \times \sigma_{\bar{x}}$:

$$\bar{x} \pm (2 \times \sigma_{\bar{x}}) = 61 \pm (2 \times 1.206) \tag{2.6}$$

The 0.95 probability region is between 58.8 and 63.7. The probability region we compute is centered around the **estimated** mean. If we repeatedly sample from this population, we will get different sample means each time, but the width of the interval would remain identical if we use the exact $\sigma_{\bar{x}}$ value we computed above. That means that under repeated sampling, the location of the mean and therefore the location of the probability region will vary.

The key thing to understand here is that probability region is centered around the **sample** mean, which will vary with each sample even if we sample from a population with a given distribution with a specific mean and standard deviation.

Suppose now that sample size was four times bigger (44). Let's again calculate the sample mean, the standard deviation of the corresponding sampling distribution, and from this information, compute the 95% confidence interval. First, we need to compute $\sigma_{\bar{x}}$:

```
> sample.44 <- rnorm(44,mean=60,sd=4)
> estimated.mean <- mean(sample.44)
> n <- length(sample.44)
> (SD.distribution <- SD.population/sqrt(n))
```

```
[1] 0.60302
```

Now we get a much tighter 95% confidence interval:

$$\bar{x} \pm 2 \times \sigma_{\bar{x}} = 60 \pm 2 \times 0.603 \tag{2.7}$$

The interval now is between 58.3 and 60.7, smaller than the one we got for the smaller sample size. In fact, it is exactly half as wide. Take a moment to make sure you understand why.

2.7 Realistic Statistical Inference

Until now we have been sampling from a population whose mean and standard deviation we know. However, we normally don't know the population parameters. In other words, although we know that:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}} \tag{2.8}$$

when we take samples in real life, we usually don't know σ. After all, it is based on an average of squared distances from the population mean μ, and that is usually the very thing we are trying to estimate!

What we *do* have, however, is the standard deviation *of the sample itself* (denoted s). This in turn means that we can only get an *estimate* of $\sigma_{\bar{x}}$. This is called the STANDARD ERROR (SE) or estimated standard error of the sample mean:

$$SE_{\bar{x}} = \frac{s}{\sqrt{n}} \tag{2.9}$$

Pay careful attention to the distinction between s (an estimate of the standard deviation of the population σ) and $SE_{\bar{x}}$ (an estimate of the standard deviation of the sampling distribution, which is in turn based on s).

We saw previously that the size of $\sigma_{\bar{x}}$—a measure of the spread of the sampling distribution—is crucial in determining the size of a 95% confidence interval for a particular sample. Now we only have an estimate of that spread. Moreover, the estimate will change from sample to sample, as the value of s changes. This introduces a new level of uncertainty into our task: the quantity $\sigma_{\bar{x}}$ has become an estimate based on an estimate! Intuitively, we would expect the confidence interval to increase in size, reflecting this increase in uncertainty. We will see how to quantify this intuition presently.

First, however, we should explore the pattern of variability in this new statistic we have introduced, s, which (like the sample mean) will vary randomly from sample to sample. Can we safely assume that s is a reliable estimate of σ?

2.8 s^2 provides a good estimate of σ^2

Earlier in this chapter we repeatedly sampled from a population of people with mean age 60 years and standard deviation 4 years; then we plotted the distribution of sample means that resulted from the repeated samples. One thing we noticed was that the sample means tended to be clustered around the value corresponding to the population mean (60). Let's repeat this experiment, but this time we plot the distribution of the samples' variances (Figure 2.8).

```
> sample.var <- rep(NA,1000)
> for(i in c(1:1000)){
     sample.40 <- rnorm(40,mean=60,sd=4)
     sample.var[i] <- var(sample.40)
   }
> hist(sample.var)
```

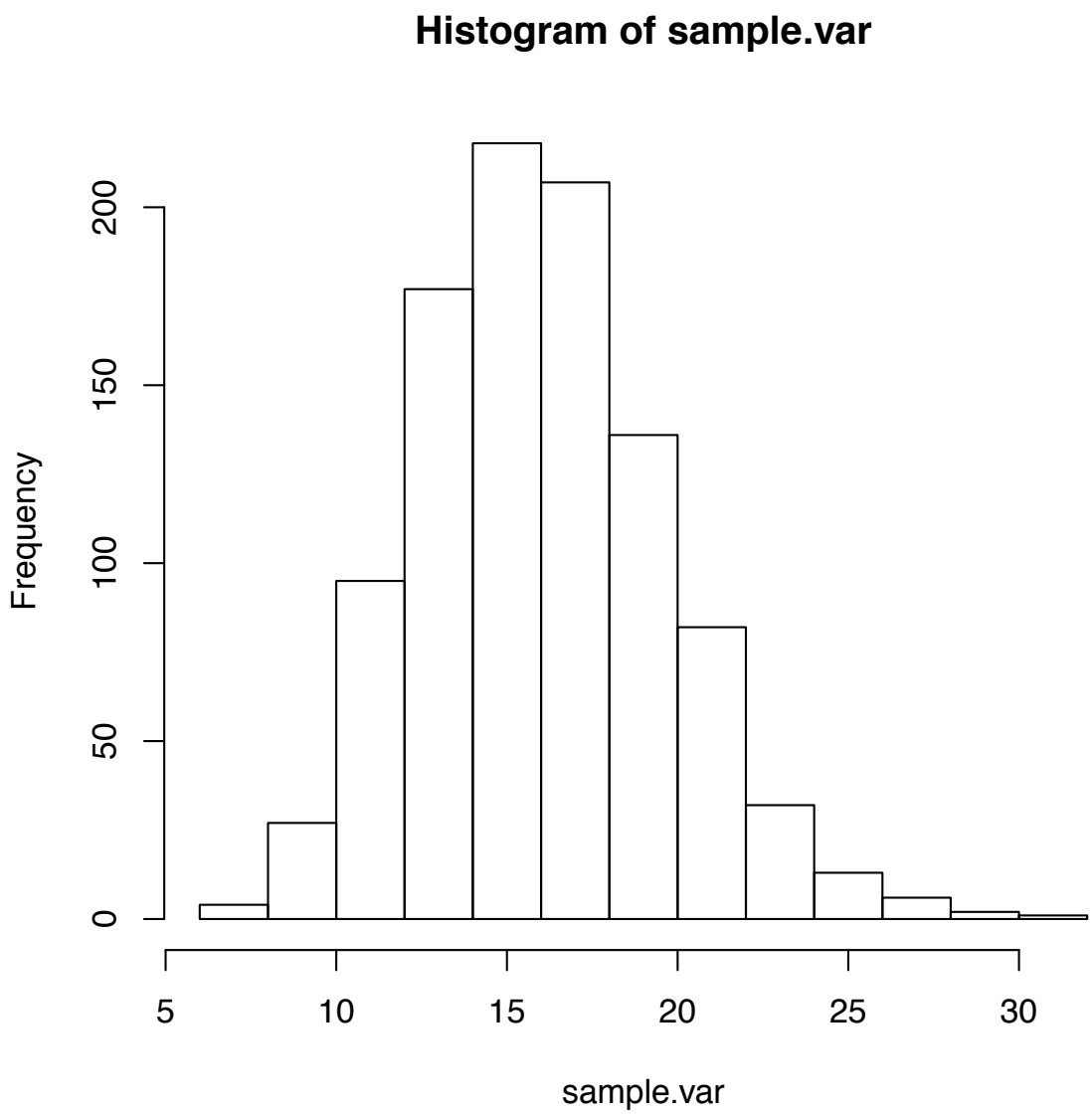


Figure 2.8: The distribution of the sample variance, sample size 40.

Figure 2.8 shows that the sample variances s^2 tend to cluster around the population variance (16). This is true even if we have an exponentially distributed population whose variance is 1 (Figure 2.9).

```
> sample.var <- rep(NA,1000)
> for(i in c(1:1000)){
     sample.var[i] <- var(rexp(40))
   }
> hist(sample.var)
```

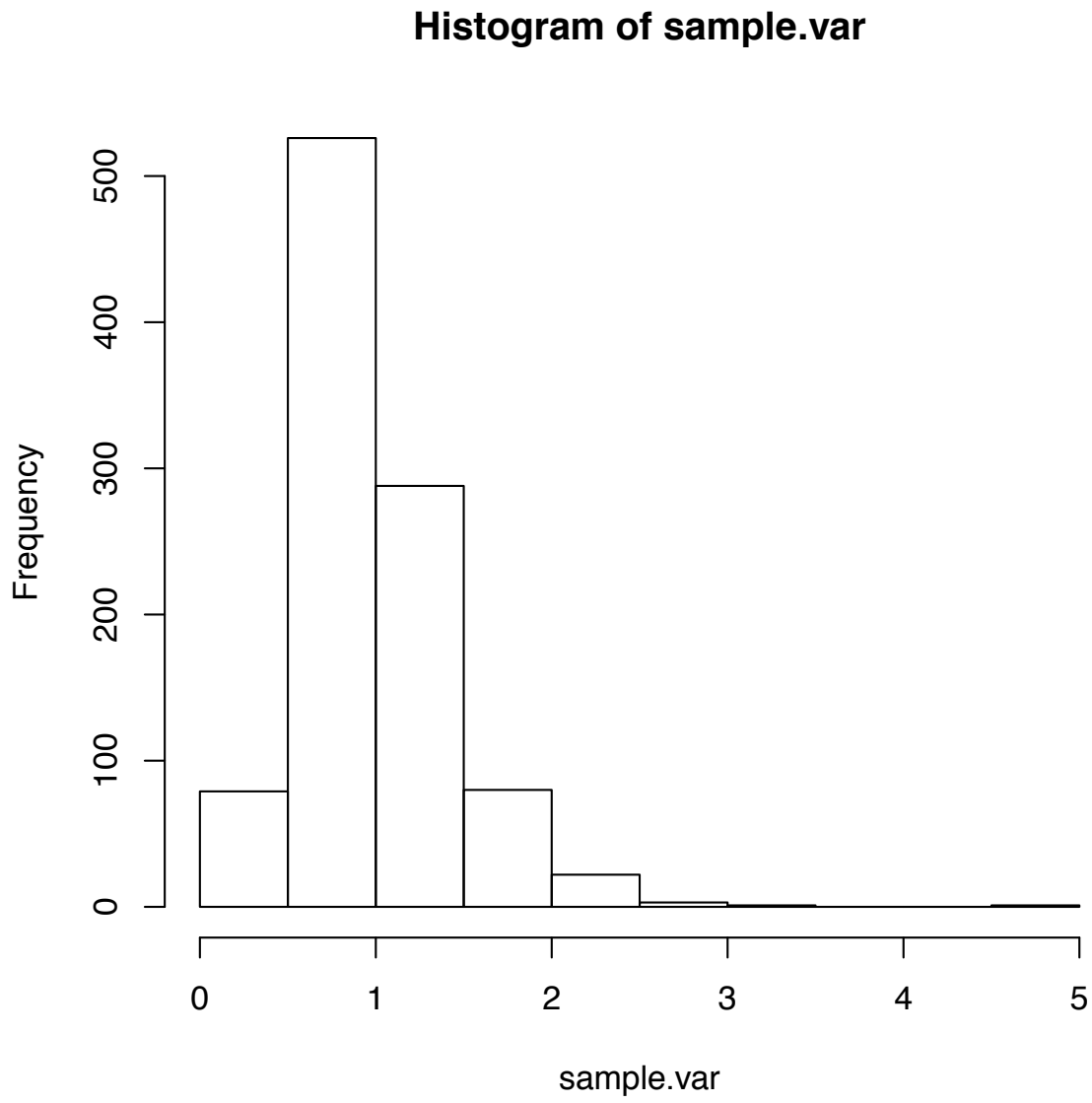


Figure 2.9: The sampling distribution of sample variances from an exponentially distributed population.

We use the square root of the sample variance s as an estimate of the unknown population standard deviation σ. This in turn allows us to estimate the standard deviation of the sampling distribution $\sigma_{\bar{x}}$ using the Standard Error $SE_{\bar{x}}$.

Notice that the Standard Error will vary from sample to sample, since the estimate s of the population parameter σ will vary from sample to sample. And of course, as the sample size increases the estimate s becomes more accurate, as does the SE, suggesting that the uncertainty introduced by this extra layer of estimation will be more of an issue for smaller sample sizes.

Our problem now is that the sampling distribution of the sample mean will take the estimate s from the sample, not σ, as a parameter. If we were to derive some value v for the SE, and simply plug this in to the normal distribution for the sample statistic, this would be equivalent to claiming that v *really was* the population parameter σ.

What we require is a distribution whose shape has greater uncertainty built into it than the normal distribution. This is the motivation for using the so-called t-DISTRIBUTION, which we turn to next.

2.9 The t-distribution

As discussed above, the distribution we use with an estimated s needs to reflect greater uncertainty at small sample sizes. There is in fact a family of t-distribution curves whose shapes vary with sample size. In the limit, if the sample were the size of the entire population, the t-distribution would *be* the normal distribution (since then s would *be* σ), so the t-curve becomes more like the normal distribution in shape as sample size increases. This t-distribution is formally defined by the DEGREES OF FREEDOM (which is simply sample size minus 1 in this case; we won't worry too much about what degrees of freedom means at this stage) and has more of the total probability located in the tails of the distribution. It follows that the probability of a sample mean being close to the true mean is slightly lower when measured by this distribution, reflecting our greater uncertainty. You can see this effect in Figure 2.10 at small sample sizes:

```
> range <- seq(-4,4,.01)
> multiplot(2,2)
>  for(i in c(2,5,15,20)){
      plot(range,dnorm(range),type="l",lty=1,
           xlab="",ylab="",
           cex.axis=1)
      lines(range,dt(range,df=i),lty=2,lwd=1)
      mtext(paste("df=",i),cex=1.2)
    }
```

But notice that with about 15 degrees of freedom, the t-distribution is already very close to normal.

The formal definition of the t-distribution is as follows: Suppose we have a random sample of size n, say of reading times (RTs), and these RTs come from a $N(\texttt{mean} = \mu, \texttt{sd} = \sigma)$ distribution. Then the quantity

$$T = \frac{\overline{X} - \mu}{S/\sqrt{n}} \tag{2.10}$$

has a $\texttt{t}(\texttt{df} = n-1)$ sampling distribution. The distribution is defined as (r is degrees of freedom; see below):

$$f_X(x,r) = \frac{\Gamma[(r+1)/2]}{\sqrt{r\pi}\,\Gamma(r/2)} \left(1 + \frac{x^2}{r}\right)^{-(r+1)/2}, \quad -\infty < x < \infty. \tag{2.11}$$

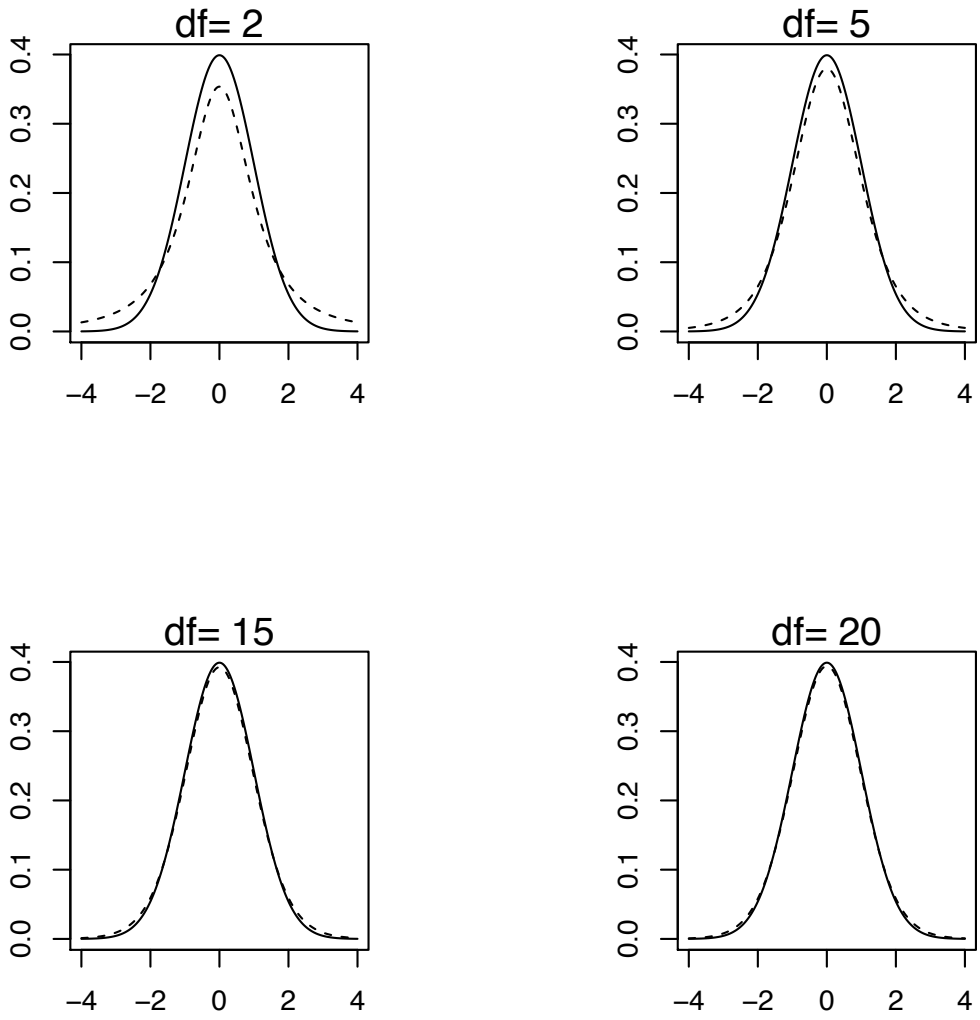


Figure 2.10: A comparison between the normal (solid line) and t-distribution (broken line) for different degrees of freedom.

[Γ refers to the gamma function; in this course we can ignore what this is, but read Kerns if you are interested.]

The t distribution is defined by its $r = n - 1$ degrees of freedom, and we will write that the sample scores are coming from $\mathtt{t}(\mathtt{df} = r)$. The shape of the function for the t distribution is similar to the normal, but the tails are considerably heavier for small sample sizes. As with the normal distribution, there are four functions in R associated with the t distribution, namely `dt`, `pt`, `qt`, and `rt`, which behave like `dnorm`, `pnorm`, `qnorm`, and `rnorm`, except for the t-distribution instead of the normal.

What do we have available to us to work with now? We have an estimate s of the population SD, and so an estimate $SE_{\bar{x}}$ of the SD of the sampling distribution:

$$SE_{\bar{x}} = \frac{s}{\sqrt{n}} \tag{2.12}$$

We also have a more spread-out distribution than the normal (at least for smaller sample sizes), the t-distribution, defined by the degrees of freedom (sample size minus 1). We are now ready to do realistic statistical inference.

2.10 The One-sample t-test

We start by taking a random sample of 11 peoples' ages from a population with mean age 60 years and standard deviation 4 years.

```
> sample <- rnorm(11,mean=60,sd=4)
```

We can ask for the 95% confidence interval, which (we saw this earlier) is *roughly* two times the Standard Error:

```
> t.test(sample)$conf.int
```

```
[1] 57.189 61.795
attr(,"conf.level")
[1] 0.95
```

Note that all of the information required to calculate this t-value is contained in the sample itself: the sample mean; the sample size and sample standard deviation s (from which we compute the SE), the degrees of freedom (the sample size minus 1, from which we reference the appropriate t-distribution). Sure enough, if our sample size had been larger, our CI would be narrower:

```
> sample <- rnorm(100,mean=60,sd=4)

> t.test(sample)$conf.int
```

```
[1] 59.215 60.873
attr(,"conf.level")
[1] 0.95
```

Given the specific sample values you get by running the above command that results in the object `sample`, try to reproduce this confidence interval by hand. Do this after the lecture for this chapter has been presented.

2.11 Some Observations on Confidence Intervals

There are some subtleties associated with confidence intervals that are often not brought up in elementary discussions, simply because the issues are just too daunting to tackle. However, we will use simulations to unpack some of these subtleties. The issues are in reality quite simple.

The first critical point to understand is the meaning of the confidence interval. Is the 95% confidence interval telling you the range within which we are 95% sure that the population mean lies? No!

Notice is that the range defined by the confidence interval will vary with each sample even if the sample size is kept constant. The reason is that the sample mean will vary each time, and the standard deviation will vary too. We can check this fact quite easily.

First we define a function for computing 95% CIs:[1]

```
> se <- function(x)
        {
          y <- x[!is.na(x)] # remove the missing values, if any
          sqrt(var(as.vector(y))/length(y))
   }
> ci <- function (scores){
   m <- mean(scores,na.rm=TRUE)
   stderr <- se(scores)
   len <- length(scores)
   upper <- m + qt(.975, df=len-1) * stderr
   lower <- m + qt(.025, df=len-1) * stderr
   return(data.frame(lower=lower,upper=upper))
   }
```

Next, we take 100 samples repeatedly from a population with mean 60 and SD 4, computing the 95% CI each time.

```
> lower <- rep(NA,100)
> upper <- rep(NA,100)
> for(i in 1:100){
     sample <- rnorm(100,mean=60,sd=4)
     lower[i] <- ci(sample)$lower
     upper[i] <- ci(sample)$upper
   }
> cis <- cbind(lower,upper)
> head(cis)

      lower  upper
[1,] 59.434 61.100
[2,] 58.515 60.294
[3,] 59.332 60.938
[4,] 58.865 60.380
[5,] 59.108 60.862
[6,] 59.912 61.405
```

[1] Here, we use the built-in R function called `qt(p,DF)` which, for a given confidence-interval range (say, 0.975), and a given degrees of freedom, DF, tells you the corresponding critical t-value.

Thus, the center and the size of any one confidence interval, based on a single sample, will depend on the particular sample mean and standard deviation you happen to observe for that sample. The sample mean and standard deviation are good estimates the population mean and standard deviation, but they are ultimately just estimates of these true parameters.

Importantly, because of the shapes of the distribution of sample means and the variances, if we repeatedly sample from a population and compute the confidence intervals each time, approximately 95% **of the confidence intervals** will contain the population mean. In the other 5% or so of the cases, the confidence intervals will not contain the population mean.

This is what 'the' 95% confidence interval means: it's a statement about confidence intervals computed with hypothetical repeated samples. More specifically, it's a statement about the probability that the hypothetical confidence intervals (that would be computed from the hypothetical repeated samples) will contain the population mean. I know that the meaning of the CI a very weird thing. But that's what it means, and our job right now is to understand this concept correctly.

So let's check the above statement. We can repeatedly build 95% CIs and determine whether the population mean lies within them. The claim is that the population mean will be in 95% of the CIs.

```
> store <- rep(NA,100)
> pop.mean<-60
> pop.sd<-4
> for(i in 1:100){
     sample <- rnorm(100,mean=pop.mean,sd=pop.sd)
     lower[i] <- ci(sample)$lower
     upper[i] <- ci(sample)$upper
     if(lower[i]<pop.mean & upper[i]>pop.mean){
       store[i] <- TRUE} else {
         store[i] <- FALSE}
   }
> ## need this for the plot below:
> cis <- cbind(lower,upper)
> ## convert store to factor:
> store<-factor(store)
> summary(store)

FALSE  TRUE
    2    98
```

So that's more or less true. To drive home the point, we can also plot the confidence intervals to visualize the proportion of cases where each CI contains the population mean (Figure 2.11).

```
> main.title<-"95% CIs in 100 repeated samples"
> line.width<-ifelse(store==FALSE,2,1)
> cis<-cbind(cis,line.width)
> x<-0:100
> y<-seq(55,65,by=1/10)
> plot(x,y,type="n",xlab="i-th repeated sample",ylab="Scores",main=main.title)
> abline(60,0,lwd=2)
```

```
> x0<-x
> x1<-x
> arrows(x0,y0=cis[,1],
         x1,y1=cis[,2],length=0,lwd=cis[,3])
```

In this figure, we control the width of the lines marking the CI using the information we extracted above (in the object `store`) to determine whether the population mean is contained in the CI or not: when a CI does not contain the population mean, the line is thicker than when it does contain the mean. You should try repeatedly sampling from the population as we did above, computing the lower and upper ranges of the 95% confidence interval, and then plotting the results as shown in Figure 2.11.

Note that when we compute a 95% confidence interval for a particular sample, we have only *one* interval. That *particular* interval does *not* have the interpretation that the probability that the population mean lies *within that interval* is 0.95. For that statement to be true, it would have to be the case that the population mean is a **random variable**, but it's not, it's a point value that we have to estimate.

Aside: Random variables

A random variable X is a function $X : S \to \mathbb{R}$ that associates to each outcome $\omega \in S$ exactly one number $X(\omega) = x$.

S_X is all the x's (all the possible values of X, the support of X). I.e., $x \in S_X$.

Good example: number of coin tosses till H

- $X : \omega \to x$
- ω: H, TH, TTH,... (infinite)
- $x = 0, 1, 2, \ldots; x \in S_X$

Every discrete random variable X has associated with it a **probability mass/distribution function (PDF)**, also called **distribution function**.

$$p_X : S_X \to [0, 1] \tag{2.13}$$

defined by

$$p_X(x) = P(X(\omega) = x), x \in S_X \tag{2.14}$$

[**Note**: Books sometimes abuse notation by overloading the meaning of X. They usually have: $p_X(x) = P(X = x), x \in S_X$]

The population mean is a single point value that cannot have a multitude of possible values and is therefore not one of many members in the set S of a random variable.

It's worth repeating the above point about confidence intervals. The meaning of the confidence interval depends crucially on hypothetical repeated samples: 95% of the confidence intervals in these repeated samples will contain the population mean. In essence, the confidence interval from a single sample is in the set S of a random variable, just like heads and tails in a coin toss are in the set S of a random variable. Just as a fair coin has a 0.5 chance of yielding a heads, a confidence interval has a 0.95 chance of containing the population mean.

The meaning of confidence intervals is confusing enough, but often (e.g., [5]), statisticians confuse the issue even further by writing, for a single sample: "We are 95% confident that the

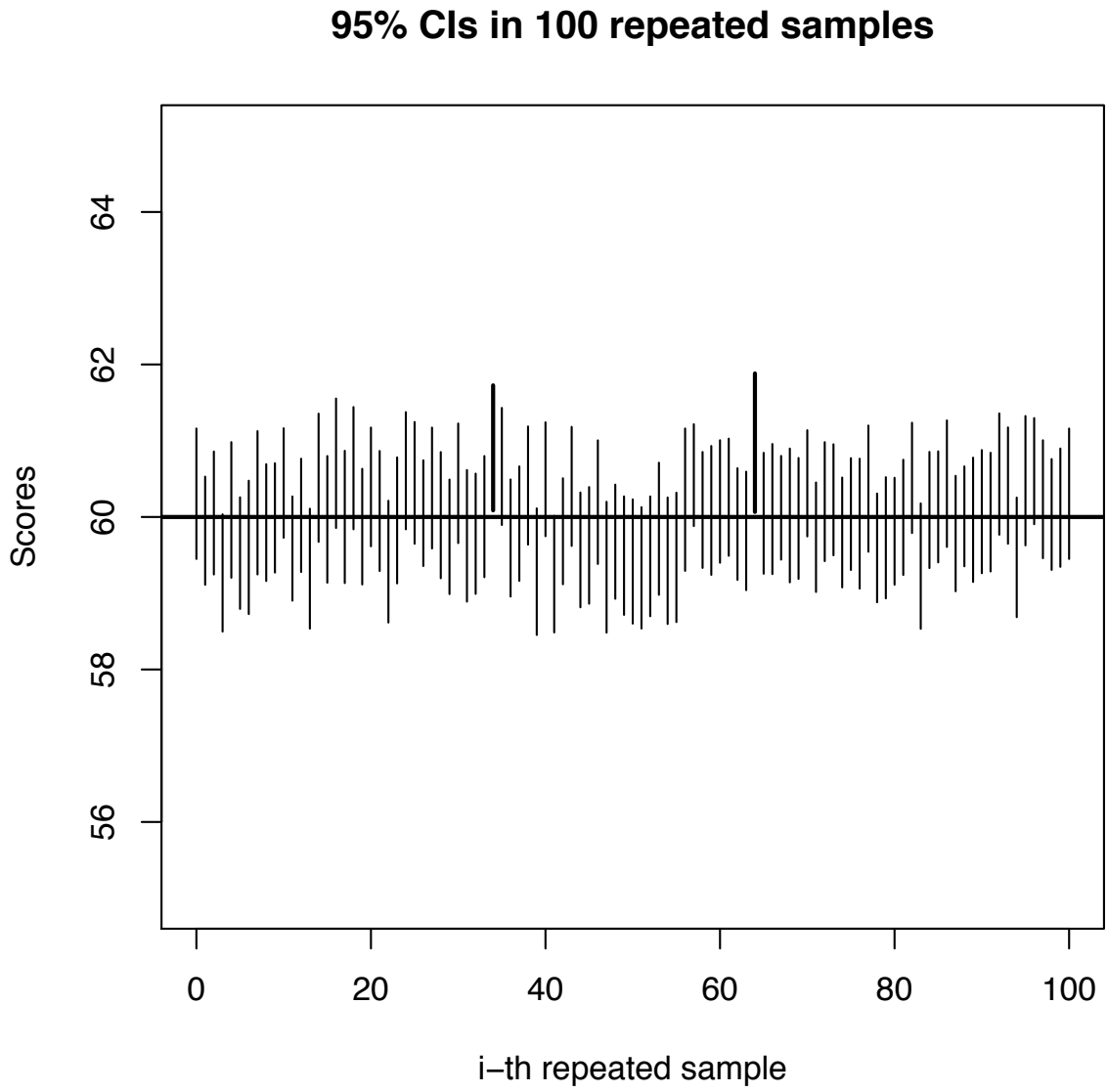


Figure 2.11: A visualization of the proportion of cases where the population mean is contained in the 95% CI, computed from repeated samples. The CIs that do not contain the population mean are marked with thicker lines.

population mean lies within this [a particular sample's] 95% CI." Here, they are using the word 'confidence' with a very specific meaning. Normally, when I say that I am 100% confident that it will rain today, I mean that the probability of it raining today is 100%. The above statement, "We are 95% confident that the population mean lies within this [a particular sample's] 95% CI.", uses 'confidence' differently; it even uses the word "this" in a very misleading way. Statistics textbooks do not mean that the probability of the population mean being in that one specific confidence interval is 95%, but rather that "95% of the confidence intervals will contain the population mean". Why this misleading wording? Either they were not paying attention to what they were writing, or they found it cumbersome to say the whole thing each time, so they (statisticians) came up with a short-cut formulation.

2.12 Sample SD and Degrees of Freedom

Let's revisit the question: Why do we use $n-1$ in the equation for standard deviation? Recall that the sample standard deviation s is just the square root of the variance: the average distance of the numbers in the list from the mean of the numbers:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{2.15}$$

We can explore the reason why we use $n-1$ in the context of estimation by considering what would happen if we simply used n instead. As we will see, if we use n, then s^2 (which is an estimate of the population variance σ^2) would be smaller than the true population variance. This smaller s^2 turns out to provide a poorer estimate than when we use $n-1$. Let's verify this using simulations.

We define new variance functions that use n, and simulate the sampling distribution of this new statistic from a population with known variance $\sigma^2 = 1$).

```
> # re-define variance to see whether it underestimates:
> new.var <- function(x){
            sum((x-mean(x))^2) / length(x)
   }
> correct <- rep(NA,1000)
> incorrect <- rep(NA,1000)
> for(i in 1:1000){
     sample.10 <- rnorm(10, mean=0, sd=1)
     correct[i] <- var(sample.10)
     incorrect[i] <- new.var(sample.10)
   }
```

As shown below (Figure 2.12), using n gives, on average, an underestimated value of the true variance:

```
> multiplot(1,2)
> hist(correct,main=paste("Mean ",round(mean(correct),digits=2),sep=" "))
> hist(incorrect,main=paste("Mean ",round(mean(incorrect),digits=2),sep=" "))
```

As we mentioned earlier, for large n it will not matter much whether we take n or $n-1$. Try it out yourself for large n to see if this is true. For more formal details on the n vs $n-1$ issue, read the book by Kerns.

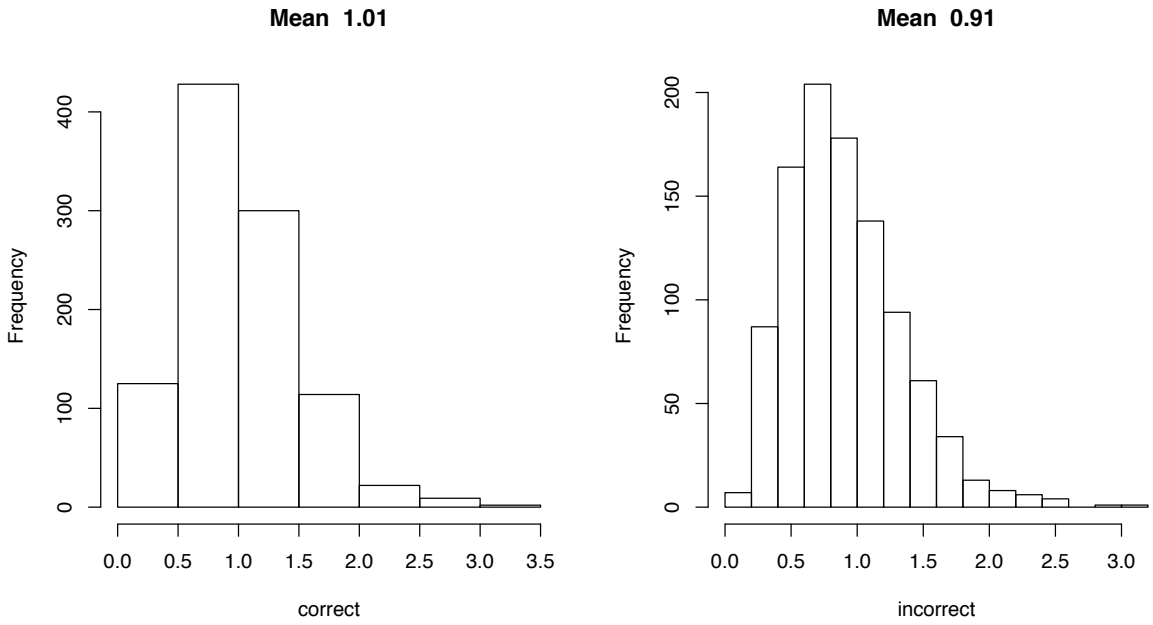


Figure 2.12: The consequence of taking $n-1$ versus n in the denominator for calculating variance, sample size 10.

2.13 Summary of the Sampling Process

It is useful at this point to summarize the terminology we have been using, and the logic of sampling. First, take a look at the concepts we have covered so far.

We provide a list of the different concepts in Table 2.1 below for easy reference. Here, $\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$ and $SE_{\bar{x}} = \frac{s}{\sqrt{n}}$.

Table 2.1: A summary of the notation used.

The sample statistic	is an estimate of
sample mean $\bar{x}$	population mean μ
sample SD s	population SD σ
standard error $SE_{\bar{x}}$	sampling distribution $\sigma_{\bar{x}}$

1. Statistical inference usually involves a single sample; due to the central limit theorem, we know that for large sample sizes, the sampling distribution is normal.

2. The statistic (e.g., sample mean) in a random sample is a good estimate of the population parameter (the population mean) than not. This follows from the normal distribution of the sample means.

3. The standard deviation of the sampling distribution $\sigma_{\bar{x}}$ is determined by two things: the inherent variability σ in the population, and the sample size. It tells us how "tight" the distribution is. If $\sigma_{\bar{x}}$ is small, then the distribution has a narrow shape: random samples are more likely to have means very close to the true population mean, and inference about the true mean is more certain. If $\sigma_{\bar{x}}$ is large, then the fall-off in probability from the center is gradual: means of random samples far from the true mean are more likely to occur, samples are not such good indicators of the population parameters, and inference is less certain.

4. We usually do not know $\sigma_{\bar{x}}$, but we can estimate it using $SE_{\bar{x}}$ and we can perform inference using the t-distribution.

2.14 Significance Tests

Recall the discussion of 95% confidence intervals: The sample gives us a mean $\bar{x}$. We compute $SE_{\bar{x}}$ (an estimate of $\sigma_{\bar{x}}$) using s (an estimate of σ) and sample size n. Then we calculate the (approximate) range $\bar{x} \pm 2 \times SE_{\bar{x}}$. That's the 95% CI. Make sure you understand why I am multiplying the standard error by 2; it's an approximation that I will presently make more precise.

We don't know the population mean—if we did, why bother sampling? But suppose we have a *hypothesis* that the population mean has a certain value. If we have a hypothesis about the population mean, then we also know what the corresponding sampling distribution would look like: we know the probability of any possible sample given that hypothesis. We then take an actual sample, measure the distance of our sample mean from the hypothesized population mean, and use the facts of the sampling distribution to determine the probability of obtaining such a sample *assuming the hypothesis is true*. This amounts to a *test* of the hypothesis. Intuitively, if the probability of our sample (given the hypothesis) is high, this provides evidence that the hypothesis

could be true. In a sense, this is what our hypothesis predicts. Conversely, if the probability of the sample is low (given the hypothesis), this is evidence against the hypothesis. The hypothesis being tested in this way is termed the NULL HYPOTHESIS. Let's do some simulations to better understand this concept.

Suppose our hypothesis, based perhaps on previous research, is that the population mean is 70, and let's assume for the moment the population $\sigma = 4$. This in turn means that the sampling distribution of the mean, given some sample size, say 11, would have a mean of 70, and a standard deviation $\sigma_{\bar{x}} = 1.2$:

```
> SD.distribution = 4/sqrt(11)

[1] 1.206
```

Figure 2.13 shows what we expect our sampling distribution to look like if our hypothesis *were in fact* true. This hypothesized distribution is going to be our reference distribution on which we base our test.

```
> range <- seq(55,85,0.01)
> plot(range,dnorm(range,mean=70,
        sd=SD.distribution),type="l",ylab="",main="The null hypothesis")
```

Suppose now that we take an actual sample of 11 from a population whose mean μ is in fact (contra the hypothesis) 60:

```
> sample <- rnorm(11,mean=60,sd=4)

> sample.mean <- mean(sample)

[1] 61.791
```

Inspection of (Figure 2.13) shows that, in a world in which the population mean was really 70, the probability of obtaining a sample whose mean is 62 is extremely low. Intuitively, this sample is "evidence against the null hypothesis".

A SIGNIFICANCE TEST provides a formal way of quantifying this reasoning. The result of such a test yields a probability that indicates exactly how well or poorly the data and the null hypothesis agree.

2.15 The Null Hypothesis

While this perfectly symmetrical, intuitive way of viewing things ('evidence for', 'evidence against') is on the right track, there is a further fact about the null hypothesis which gives rise to an asymmetry in the way we perform significance tests.

The statement being tested in a significance test— the NULL HYPOTHESIS, H_0— is usually formulated in such a way that the statement represents 'no effect,' 'pure chance' or 'no significant difference'. Scientists are usually not so interested in proving 'no effect.' This is where the asymmetry comes in: we are usually not interested in finding evidence *for* the null hypothesis, conceived in this way. Rather, we are interested in evidence *against* the null hypothesis, since this is evidence for some real statistically significant result. This is what a formal significance test does:

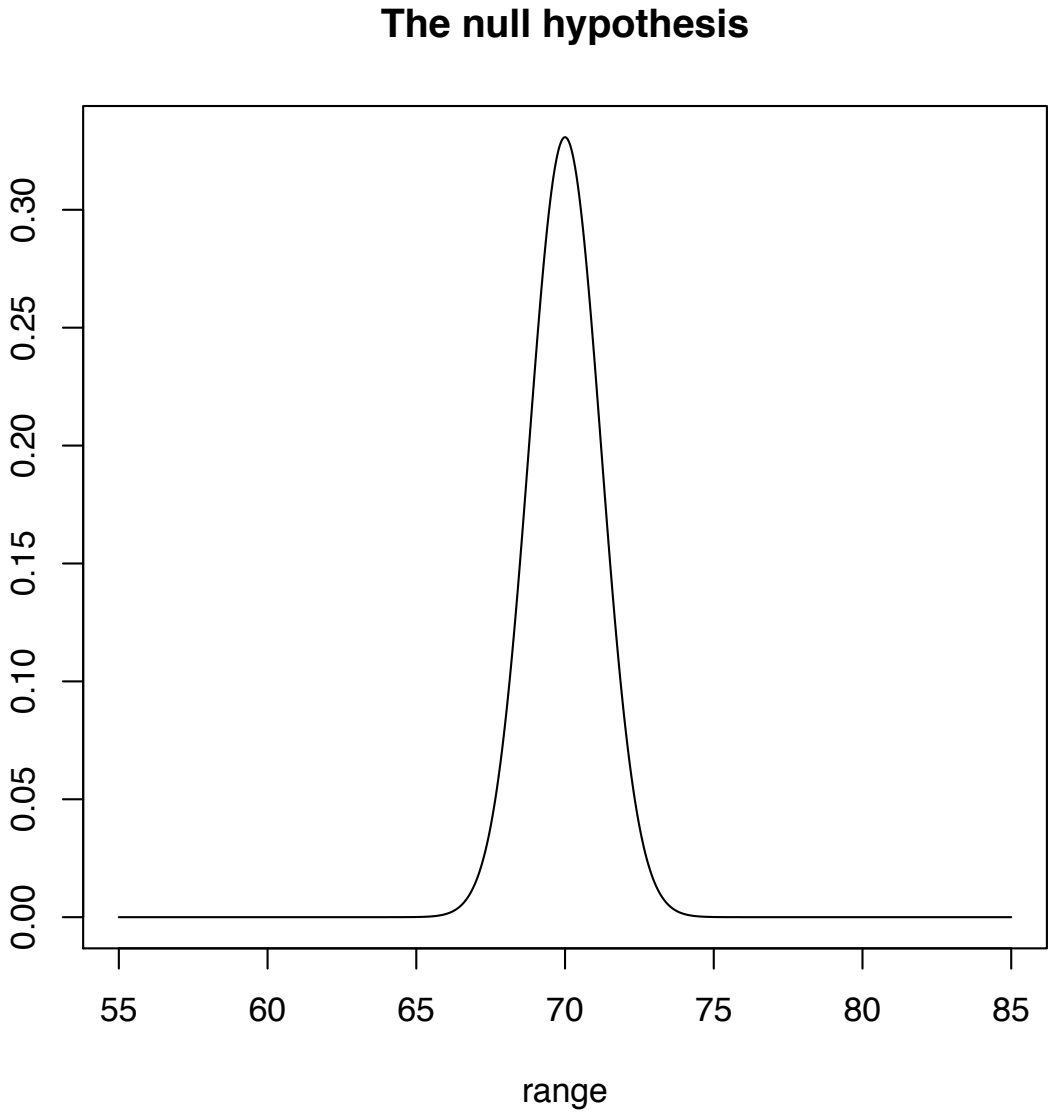


Figure 2.13: A sampling distribution with mean 70 and $\sigma_{\bar{x}} = 1.206$.

it determines whether the result provides sufficient evidence against the null hypothesis for us to reject it. Note that if it doesn't provide sufficient evidence against the null, we have *not* proved the contrary—we have not 'proved the null hypothesis.' We simply don't have enough evidence, *based on this single result*, to reject it. We come back to this in a later lecture.

In order to achieve a high degree of skepticism about the interpretation of the data, we require the evidence against the null hypothesis to be very great. In our current example, you might think the result we obtained, 62, was fairly compelling evidence against it. But how do we quantify this? Intuitively, the further away from the mean of the sampling distribution our data lies, the greater the evidence against it. Statistically significant results reside out in the tails of the distribution. How far out? The actual values and ranges of values we get will vary from experiment to experiment, and statistic to statistic. How can we determine a general rule?

2.16 z-scores

We have already seen that, in a normal distribution, about 95% of the total probability falls within 2 SD of the mean, and thus 5% of the probability falls far out in the tails. One way of setting a general rule then, is to say that if an observed value falls far out in the tail of the distribution, we will consider the result extreme enough to reject the null hypothesis (we can set this threshold anywhere we choose: 95% is a conventional setting).

Recall our model: we know the sampling distribution we would see in a world in which the null hypothesis is true, in which the population mean is really 70 (and whose population σ is known to be 4). We also know this distribution is normal. How many SDs from the mean is our observation? Is it more than 2 SDs?

We need to express the difference between our observation $\bar{x}$ and hypothesized mean of the distribution μ_0 in units of the standard deviation of the distribution: i.e., some number z times $\sigma_{\bar{x}}$. We want to know this number z.

$$\bar{x} - \mu_0 = z\sigma_{\bar{x}} \tag{2.16}$$

Solving for z:

$$z = \frac{\bar{x} - \mu_0}{\sigma_{\bar{x}}} \tag{2.17}$$

$$= \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}} \tag{2.18}$$

z is called the STANDARDIZED VALUE or the Z-SCORE. In addition, one could imagine computing this standardized version of the sample mean every time we take a sample, in which case we have effectively defined a new statistic. Viewed in this way, the score is also referred to as a TEST-STATISTIC.

Let's make this concrete. Suppose in our current simulation we draw a sample whose mean is precisely 60: then $\bar{x} = 60, \mu_0 = 70, \sigma = 4, n = 11$. So we get:

$$z = \frac{60 - 70}{4/\sqrt{11}} \tag{2.19}$$

$$= -8.291562 \tag{2.20}$$

We see that this observation is well beyond 2 SDs from the mean, and thus represents statistically significant evidence against the null hypothesis.

z-scores are a quick and accepted way of expressing 'how far away' from the hypothetical value an observation falls, and for determining if that observation is beyond some accepted threshold. Ultimately, however, they take their meaning from the probability corresponding to the value, which is traditionally expressed by rules-of-thumb (2 SD corresponds to 95%), or tables which translate particular scores to particular probabilities. It is this probability we turn to next.

2.17 P-values

We would like to reason as follows: 'If the probability of the obtaining the sample mean that we got is less than 0.05, given the null hypothesis, then we reject the null hypothesis.' However, in continuous distributions, the probability of getting exactly a particular value is (perhaps counter-intuitively) zero.

Although we cannot use the actual probability of the observed value, we can usefully ask *how much of the total probability lies beyond the observed value*, out into the tail of the distribution. In the discrete case this is a sum of probabilities, in the continuous case (normal distribution) an area under the curve. Call o_1 the observed value, o_2 the next value out, o_3 the next, and so on until we exhaust all the possibilities. The sum of these is the probability of a complex event, the probability of 'observing the value o_1 or a value more extreme.' (Once again, we couch our probability measure in terms of a range of values). This then is a measure, based directly on probability, of 'how far away' from the mean an observed value lies. The smaller this probability, the more extreme the value. We can now say, if this probability is less than 0.05, we reject the hypothesis. The technical name for this measure is the P-VALUE.

In short, the p-value of a statistical test is the probability, computed assuming that H_0 is true, that the test statistic would take a value as extreme or more extreme than that actually observed.

Note that this is a CONDITIONAL PROBABILITY: it is the probability of observing a particular sample mean (or something more extreme) conditional on the assumption that the null hypothesis is true. We can write this conditional probability as $P(\text{Observed mean} \mid H_0)$, or even more succinctly as $P(\text{Data} \mid H_0)$. The p-value does *not* measure the probability of the null hypothesis given the data, $P(H_0 \mid \text{Data})$. There is a widespread misunderstanding that the p-value tells you the probability that the null hypothesis is true (in light of some observation); it doesn't. You can confirm easily that we cannot "switch" conditional probabilities. The probability of the streets being wet given that rain has fallen $P(\text{Wet Streets} \mid \text{Rain})$ (presumably close to 1) is not at all the same as the probability of rain having fallen given that the streets are wet $P(\text{Rain} \mid \text{Wet Streets})$. There are many reasons why the streets may be wet (street cleaning, burst water pipes, etc.), rain is just one of the possibilities.

How do we determine this p-value? We simply sum up (integrate) the area under the normal curve, going out from our observed value. (Recall that, for the present, we are assuming we *know* the population parameter σ). We actually have two completely equivalent ways to do this, since we now have two values (the actual observed value and its z-score), and two corresponding curves (the sampling distribution where the statistic is the sample mean, and the sampling distribution where the statistic is the standardized mean, the 'z-statistic'). We have seen what the sampling distribution of the sample mean looks like, assuming the null hypothesis is true (i.e. $\mu_0 = 70$, Figure 2.13). What is the sampling distribution of the z-statistic under this hypothesis? Let's do a simulation to find out.

In Figure 2.14, we repeat the simulation of sample means that we carried out at the beginning of the chapter, but now using the parameters of our current null hypothesis $\mu_0 = 70$, $\sigma = 4$, sample size $= 11$. But in addition, for each sample we also compute the z-statistic, according to the formula provided above. We also include the corresponding normal curves for reference (recall these represent the limiting case of the simulations). As you can see, the distribution of the z-statistic is normal, with mean $= 0$, and SD $= 1$. A normal distribution with precisely these parameters is known as the STANDARDIZED NORMAL DISTRIBUTION.

```
> sample.means <- rep(NA, 1000)
> zs <- rep(NA, 1000)
> for(i in 1:1000){
     sample.11 <- rnorm(11,mean=70,sd=4)
     sample.means[i] <- mean(sample.11)
     zs[i] <- ( mean(sample.11) - 70 ) / (4/sqrt(11))
   }
> multiplot(2, 2)
> sd.dist <- 4/sqrt(11)
> plot(density(sample.means,kernel="gaussian"),xlim=range(70-(4*sd.dist),
   70+(4*sd.dist)),xlab="",ylab="",main="")
> plot(density(zs,kernel="gaussian"),xlim=range(-4, 4),xlab="",ylab="",main="")
> plot(function(x) dnorm(x, 70, 4/sqrt(11)),
   70-(4*sd.dist), 70+(4*sd.dist),xlab="",ylab="",main="")
> plot(function(x) dnorm(x, 0, 1), -4, 4,xlab="",ylab="",main="")
```

The crucial thing to note is that the area from either value out to the edge, which is the probability of interest, is precisely the same in the two cases, so we can use either. It is traditional to work with the standardized values, for reasons that will become clear.

Recall the z-score for our actual observation was -8.291562. This is an extreme value, well beyond 2 standard errors from the mean, so we would expect there to be very little probability between it and the left tail of the distribution. We can calculate it directly by integration:

```
> integrate(function(x) dnorm(x, mean = 0, sd = 1), -Inf, -8.291562)

5.5885e-17 with absolute error < 4.5e-24

> ## alternative, more standard way:
> pnorm(mean=0,sd=1,-8.291562)

[1] 5.5885e-17
```

This yields a vanishingly small probability. We also get precisely the same result using the actual observed sample mean with the original sampling distribution:

```
> integrate(function(x) dnorm(x, mean = 70, sd = 4/sqrt(11)), -Inf, 60)

5.5885e-17 with absolute error < 6.2e-20

> pnorm(60,mean=70,sd=4/sqrt(11))
```

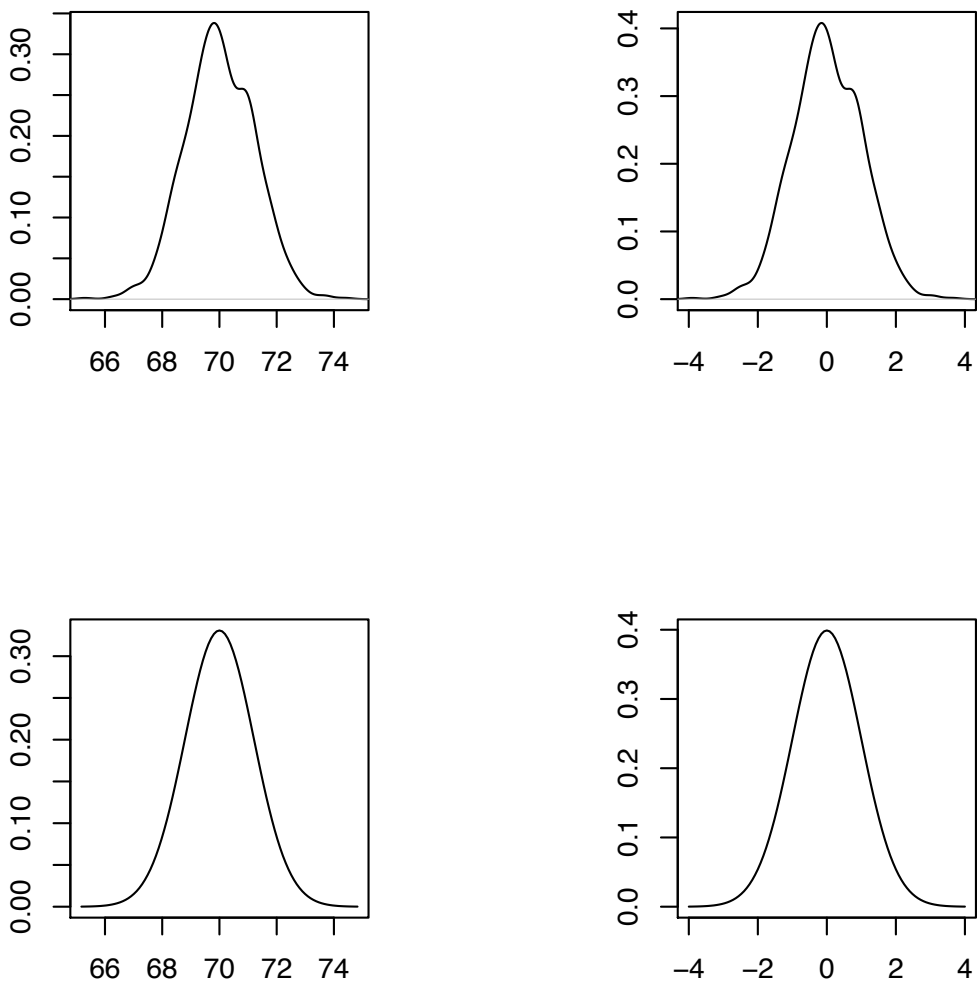


Figure 2.14: The sampling distribution of the sample mean (left) and its z-statistic (right).

```
[1] 5.5885e-17
```

Suppose now we had observed a sample mean of 67.58. This is much closer to the hypothetical mean of 70. The standardized value here is almost exactly -2.0:

```
> (67.58-70)/(4/sqrt(11))

[1] -2.0066
```

Integrating under the standardized normal curve we find the following probability:

```
> integrate(function(x) dnorm(x, 0, 1), -Inf, -2.0)

0.02275 with absolute error < 1.5e-05

> pnorm(-2,mean=0,sd=1)

[1] 0.02275
```

This accords well with our rule-of-thumb. About 95% of the probability is within 2 standard errors of the mean. The remainder is split into two, one at each end of the distribution, each representing a probability of about 0.025.

In the code above, we have used the `integrate` function, but the standard way to do this in R (and this is what we will do from now on) is to use `pnorm`. For example: we can compute the probability of getting a z-score like -8.291562 or smaller using:

```
> pnorm(-8.291562)

[1] 5.5885e-17
```

Note that I did not specify the mean and sd; this is because the default assumption in this function is that mean is 0 and sd=1.

2.18 Hypothesis Testing: A More Realistic Scenario

In the above example we were able to use the standard deviation of the sampling distribution $\sigma_{\bar{x}}$, because we were given the standard deviation of the population σ. As we remarked earlier, in the real world we usually do not know σ, it's just another unknown parameter of the population. Just as in the case of computing real world confidence intervals, instead of σ we use the estimate s; instead of $\sigma_{\bar{x}}$ we use the estimate $SE_{\bar{x}}$; instead of the normal distribution we use the t-distribution.

Recall the z-score:

$$z = \frac{\bar{x} - \mu_0}{\sigma_{\bar{x}}} \tag{2.21}$$

$$= \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}} \tag{2.22}$$

And recall our formal definition of a statistic: a number that describes some aspect of the sample. Using this definition, the z-score seems to fail as a statistic, since it makes reference to a

population *parameter* σ. But if we now replace that parameter with an estimate s derived from the sample itself, we get the so-called t-statistic:

$$t = \frac{\bar{x} - \mu_0}{SE_{\bar{x}}} \tag{2.23}$$

$$= \frac{\bar{x} - \mu_0}{s/\sqrt{n}} \tag{2.24}$$

This then can also be interpreted as yet another sampling statistic, with its own distribution.

Note that our null hypothesis H_0 was that the hypothesized mean has some specific value $\mu = \mu_0$. Thus, rejecting the null hypothesis amounts to accepting the alternative hypothesis, i.e., that the true value is less than the hypothesized mean *or* the true value is greater than the mean:

$$H_a : \mu \neq \mu_0 \Leftrightarrow \mu < \mu_0 \text{ or } \mu > \mu_0 \tag{2.25}$$

This means that as evidence for rejection of H_0 we will count extreme values on *both* sides of μ. For this reason, the above test is called a TWO-SIDED SIGNIFICANCE TEST (also known as the TWO-TAILED SIGNIFICANCE TEST). Note that if we simply reported the probability corresponding to the t-value t, we would *not* be reporting the probability of 'a value being more than t away' from the mean, but the probability in one direction only. For that reason, in a two-sided test, since the distributions are symmetrical, the p-value will be twice the value of the probability corresponding to the particular t-value we obtain. If the p-value is $\leq \alpha$, we say that the data are significant at level α. Purely by convention, $\alpha = 0.05$.

By contrast, if our null hypothesis were that $\mu \geq \mu_0$, then the alternative hypothesis would be:

$$H_a : \mu < \mu_0 \tag{2.26}$$

In this situation, we would use a one-sided significance test, reporting the probability in the relevant direction only.

R does everything required for a t-test of significance as follows, and you can specify (inter alia) what your μ_0 is (note that it need not be zero), whether it is two-sided or not (see the documentation for the `t.test` for details on how to specify this), and the confidence level (the α level) you desire, as follows:

```
> sample.11 <- rnorm(11,mean=60,sd=4)
> t.test(sample.11,
          alternative = "two.sided",
          mu = 70,
          conf.level = 0.95)

        One Sample t-test

data:  sample.11
t = -6.5608, df = 10, p-value = 6.385e-05
alternative hypothesis: true mean is not equal to 70
95 percent confidence interval:
 58.536 64.349
```

Table 2.2: Hypothetical data showing reading times for adults and children.

group	sample size n	Mean (secs)	SD
children	$n_1 = 10$	$\bar{x}_1 = 30$	$s_1 = 43$
adults	$n_2 = 20$	$\bar{x}_2 = 7$	$s_2 = 25$

```
sample estimates:
mean of x
   61.443
```

Experiment with the above code: change the hypothetical mean, change the mean of the sampled population and its SD, change the sample size, etc. In each case, see how the sample mean, the t-score, the p-value and the confidence interval differ. Make sure you understand what the output says—you have the relevant background at this point to do so.

It is also instructive to keep the parameters the same and simply repeat the experiment, taking different random samples each time (effectively, REPLICATING the experiment). Watch how the p-values change, watch how they change from replicate to replicate under different parameter settings. Do you ever find you would accept the null hypothesis when it is in fact false? How likely is it that you would make a mistake like that? This is an issue we will return to in more depth later.

The t-value we see above is indeed the t in equation 2.23; we can verify this by doing the calculation by hand:

```
> (mean(sample.11)-70)/se(sample.11)

[1] -6.5608
```

2.19 Comparing Two Samples

In one-sample situations our null hypothesis is that the population mean has some specific value μ_0:

$$H_0 : \mu = \mu_0 \tag{2.27}$$

When we compare samples from two (possibly) different populations, we ask the question: are the population means identical or not? Our goal now is to figure out some way to define our null hypothesis in this situation.

Consider this example of a common scenario in experimental research. Let us assume that the mean reading times and standard deviations are available for children and adults reading English sentences, and let us say that we want to know whether children are faster or slower than adults in terms of reading time. You probably don't need to do an experiment to answer this question, but it will do as an illustration of this type of experiment.

We know that, due to the nature of repeated sampling, there is bound to be *some* difference in sample means even if the population means are identical. We can reframe the research question as follows: is the difference observed between the two sample means consistent or inconsistent with our null hypothesis. The data are shown in Table 2.2.

Notice a few facts about the data. We have different sample sizes in each case. How will that affect our analysis? Notice too that we have different standard deviations in each case: this makes sense, since children exhibit a wider range of abilities than literate adults. But we now know how great an effect the variability of the data has on statistical inference. How will we cope with these different SD's? Finally, the mean reading times certainly 'look' different. We will quantify this difference with reference to the null hypothesis.

Such research problems have the properties that (i) the goal is to compare the responses in two groups; (ii) each group is considered a sample from a distinct population (a 'between-subjects' design); (iii) the responses in each group are independent of those in the other group; and (iv) the sample sizes of each group may or may not be different.

The question now is, given that we have a research question involving two means, how can we formulate the null hypothesis?

2.19.1 H_0 in Two-sample Problems

Let us start by saying that the unknown population mean of children is μ_1, and that of adults is μ_2. We can state our null hypothesis as follows:

$$H_0 : \mu_1 = \mu_2 \tag{2.28}$$

Equivalently, we can say that our null hypothesis is that the difference between the two means is zero:

$$H_0 : \mu_1 - \mu_2 = 0 = \delta \tag{2.29}$$

We have effectively created a new population parameter δ:

$$H_0 : \delta = 0 \tag{2.30}$$

We can now define a new statistic $d = \bar{x}_1 - \bar{x}_2$ and use that as an estimate of δ, which we've hypothesized to be equal to zero. But to do this we need a sampling distribution of the difference of the two sample means $\bar{x}_1$ and $\bar{x}_2$.

Let's do a simulation to get an understanding of this approach. For simplicity we will use the sample means and standard deviations from the example above as our population parameters in the simulation, and we will also use the sample sizes above for the repeated sampling. Assume a population with $\mu_1 = 30$, $\sigma_1 = 43$, and another with mean $\mu_2 = 7$, $\sigma_2 = 25$. So we already know in this case that the null hypothesis is false, since $\mu_1 \neq \mu_2$. But let's take 1000 sets of samples of each population, compute the differences in mean in each set of samples, and plot that distribution of *the differences of the sample mean*:

```
> d <- rep(NA,1000)
> for(i in 1:1000){
     sample1 <- rnorm(10,mean=30,sd=43)
     sample2 <- rnorm(20,mean=7,sd=25)
     d[i] <- mean(sample1) - mean(sample2)
   }
```

Note that the mean of the differences-vector `d` is close to the true difference:

```
> 30-7

[1] 23

> mean(d)

[1] 22.979
```

Then we plot the distribution of `d`; we see a normal distribution (Figure 2.15).

```
> hist(d)
```

So, the distribution of the differences between the two sample means is normally distributed, and centered around the true difference between the two populations. It is because of these properties that we can safely take d to be an estimate of δ. How accurate an estimate is it? In other words, what is the standard deviation of this new sampling distribution? It is clearly dependent on (a function of) the standard deviation of the two populations in some way:

$$\sigma_{\bar{x}_1-\bar{x}_2} = f(\sigma_1, \sigma_2) \tag{2.31}$$

(Try increasing one or other or both of the σ in the above simulation to see what happens). The precise relationship is fundamentally additive: instead of taking the root of the variance, we take the root of the sum of variances:

$$\sigma_{\bar{x}_1-\bar{x}_2} = \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}} = \sqrt{\frac{43^2}{10} + \frac{25^2}{20}} = 14.702. \tag{2.32}$$

```
> newsigma<-round(sqrt((43^2/10)+(25^2/20)),digits=4)
```

In our single sample, $\bar{x}_1 - \bar{x}_2 = 17$. The null hypothesis is $\mu_1 - \mu_2 = 0$. How should we proceed? Is this sample difference sufficiently far away from the hypothetical difference (0) to allow us to reject the null hypothesis? Let's first translate the observed difference 17 into a z-score. Recall how the z-score is calculated:

$$z = \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}} = \frac{\text{sample mean} - \text{pop. mean}}{\text{sd of sampling distribution}} \tag{2.33}$$

If we replace $\bar{x}$ with d, and the new standard deviation from the two populations' standard deviations, we are ready to work out the answer:

$$z = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{2.34}$$

$$= \frac{17 - 0}{14.702} \tag{2.35}$$

$$= 1.1563 \tag{2.36}$$

Using exactly the same logic as previously, because we don't know the population parameters in realistic settings, we replace the σ's with the sample standard deviations to get the t-statistic:

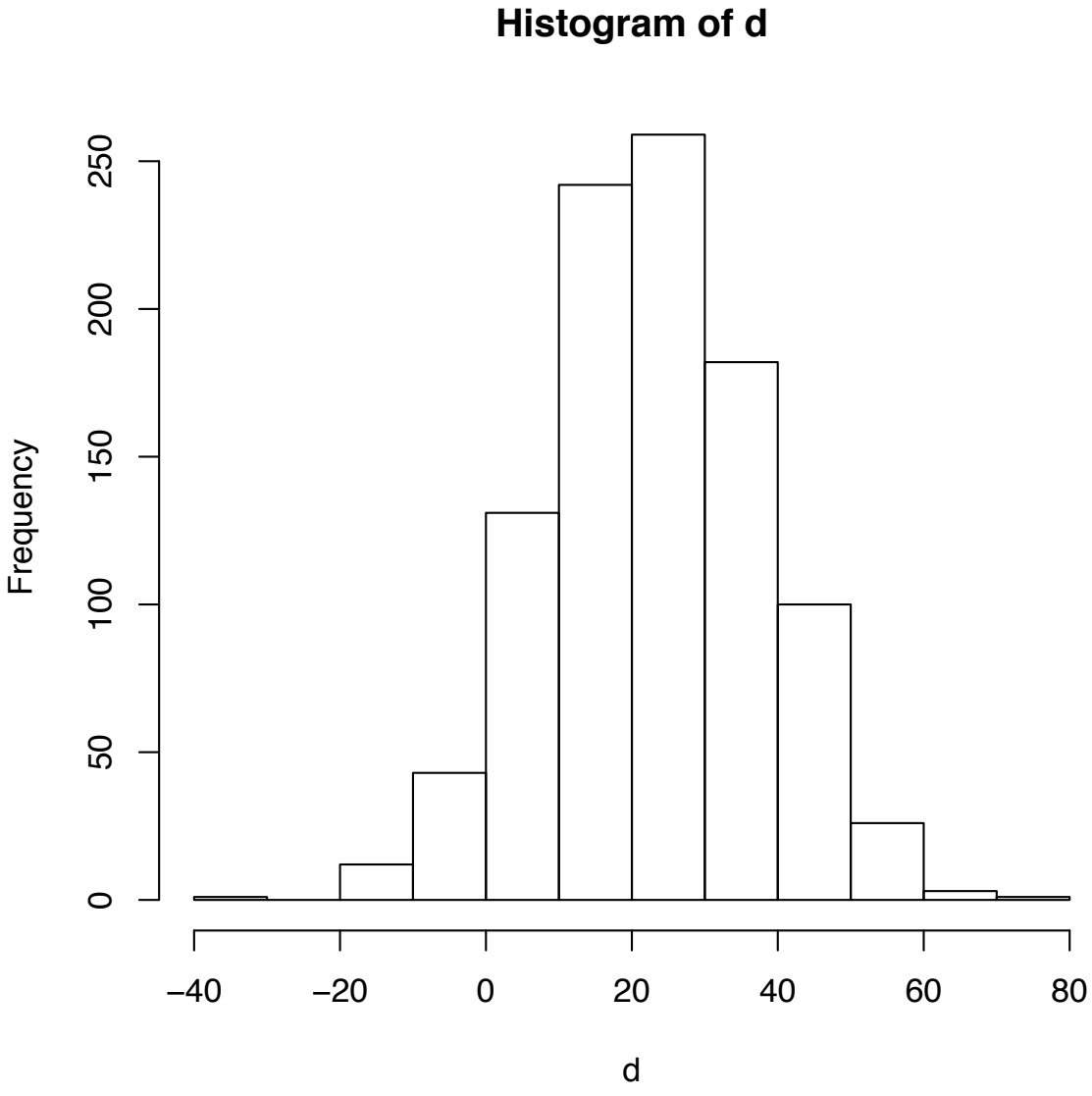


Figure 2.15: The distribution of the difference of sample means of two samples.

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{2.37}$$

This is the TWO-SAMPLE t-STATISTIC.

So far so good, but we want to now translate this into a p-value, for which we need the appropriate t-distribution. The problem we face here is that the degrees of freedom needed for the correct t-distribution are not obvious. The t-distribution assumes that only one s replaces a single σ; but we have two of these. If $\sigma_1 = \sigma_2$, we could just take a *weighted average* of the two sample SDs s_1 and s_2.

In our case the correct t-distribution has $n_1 - 1 + n_2 - 1$ degrees of freedom (the sum of the degrees of freedom of the two sample variances; see [6, 422] for a formal proof).

In real life we don't know whether $\sigma_1 = \sigma_2$. One response would be to err on the side of caution, and simply use degrees of freedom corresponding to the smaller sample size. Recall that smaller degrees of freedom reflect greater uncertainty, so the estimate we get from this simple approach will be a conservative one.

However, in a more sophisticated approach, something called Welch's correction corrects for possibly unequal variances in the t-curve. R does this correction for you if you specify that the variances are to be assumed to be unequal (`var.equal=FALSE`).

```
> t.test.result<-t.test(sample1,sample2,
          mu=0,
          alternative = "two.sided",
           conf.level = 0.95,var.equal=FALSE)
```

If you print out the contents of `t.test.result`, you will see detailed output. For our current discussion it is sufficient to note that the t-value is 2.67, the degrees of freedom are 10.45 (a value somewhere between the two sample sizes), and the p-value is 0.02. Recall that every time you run the t-test with newly sampled data (you should try this), your results will be slightly different; so do not be surprised if you occasionally fail to find a significant difference between the two groups even though you already know that in reality there is such a difference. We turn to this issue in the next lecture.

Chapter 3

Power

Let's assume we do an experiment, compute the t-value and p-value for our sample, and based on these values, reject the null hypothesis. As we mentioned in the previous chapter, and as you can prove to yourself through simulated replication of experiments, due to the very nature of random sampling it is always *possible* to stumble on a 'rogue sample', one whose statistic happens to be far from the population parameter. In this case it would, in fact, be an error to reject the hypothesis, though we wouldn't know it. The technical name for this is a TYPE I ERROR: the null hypothesis is true, but our sample leads us to reject it.

The converse may also happen. Suppose the null hypothesis is indeed false—there is a real difference between two population means, for example—but the sample values we have happen to be so close to each other that this difference is not detectable. Here, the null hypothesis is false, but we fail to reject it based on our sample. Again, we have been misled by the sampling process: this is known as a TYPE II ERROR.

In the first case, we would think our experiment had succeeded, publish our result, and move on, unaware of our mistake. Can we minimize this risk? In the second case, we don't get a significant difference, it appears our experiment has failed. Is there some way to minimize the risk?

Addressing these questions is of fundamental importance in experimental design. It turns out that a couple of the obvious things we *might* do to improve our experiments have unpleasant implications. For example, we might think that making the probability threshold more stringent—0.01 instead of 0.05, for example—will minimize the chance of error. We pay a price for that, however. The reason is that there is an intimate relationship between the two types of error, and a technique that simply aims to minimize one kind can unwittingly increase the chance of the other. This chapter uses simulation to explore this interaction.

3.1 Type I and Type II Errors

We fix some conventions first for the text below. Let: R = 'Reject the null hypothesis H_0'; $\neg R$ = 'Fail to reject the null hypothesis H_0.' These are **decisions** we make based on an experiment.

The decision R or $\neg R$ is based on the sample. Keep in mind that when we do an experiment we don't know whether the null hypothesis is true or not.

The first step in attempting to minimize error is to have some way to measure it. It turns out we can use probability for this as well: we will use conditional probabilities of the following kind: Let $P(R \mid H_0)$ = 'Probability of rejecting the null hypothesis *conditional on the assumption that the null hypothesis is in fact true.*'

Table 3.1: The logical possibilities given the two possible situations: null hypothesis true (H_0) or false ($\neg H_0$).

Reality:	H_0 TRUE	H_0 FALSE
Decision from sample is 'reject':	α	$1 - \beta$
	Type I error	**Power**
Decision from sample is 'accept':	$1 - \alpha$	β
		Type II error

Let's work through the logical possibilities that could hold: the null hypothesis could be in fact true or in fact false (but we don't know which), and in addition our decision could be to accept or reject the null hypothesis (see Table 3.1). In only two of these four possible cases do we make the right decision. In the table, think of α as the threshold probability we have been using all along, 0.05.

As shown in Table 3.1, the probability of a Type I error $P(R \mid H_0)$ is α, conventionally set at 0.05. We will see why this is so shortly. But it immediately follows that the probability of the logical complement $P(\neg R \mid H_0)$ is $1 - \alpha$. We define the probability of a Type II error $P(\neg R \mid \neg H_0)$ to be β (more on this below), but it immediately follows that the probability of the logical complement $P(\neg R \mid \neg H_0) = 1 - \beta$. We call this probability POWER. Thus, if we want to decrease the chance of a Type II error, we need to increase the power of the statistical test.

Let's do some simulations to get a better understanding of these various definitions. We focus on the case where the null hypothesis is in fact false: there is a real difference between population means.

Assume a population with mean $\mu_1 = 60$, $\sigma_1 = 1$, and another with mean $\mu_2 = 62$, $\sigma_2 = 1$. In this case we already *know* that the null hypothesis is false. The distribution corresponding to the null hypothesis is shown in Figure 3.1. It is centered around 0, consistent with the null hypothesis that the difference between the means is 0.

We define a function for easily shading the regions of the plot we are interested in. The function below, `shadenormal2`, is a modified version of the function `shadenormal.fnc` available from the package `languageR` (you do not need to load the library `languageR` to use the function below).

First, we define a function that will plot Type I error intervals. This function requires that several parameters be set (our use of this function will clarify how to use these parameters):

```
> ## function for plotting type 1 error.
> plot.type1.error<-function(x,
                             x.min,
                             x.max,
                             qnts,
                             mean,
                             sd,
                             gray.level,main,show.legend=TRUE){

          plot(x,dnorm(x,mean,sd),
                        type = "l",xlab="",ylab="",main=main)
```

```
          abline(h = 0)

   ## left side
       x1 = seq(x.min, qnorm(qnts[1]), qnts[1]/5)
       y1 = dnorm(x1, mean, sd)

       polygon(c(x1, rev(x1)),
               c(rep(0, length(x1)), rev(y1)),
               col = gray.level)

   ## right side
       x1 = seq(qnorm(qnts[2]), x.max, qnts[1]/5)
       y1 = dnorm(x1, mean, sd)
       polygon(c(x1, rev(x1)),
               c(rep(0, length(x1)), rev(y1)),
               col = gray.level)
   if(show.legend==TRUE){legend(2,0.3, legend="Type I error",fill=gray.level,cex=1)}
          }
```

Next, we define a function for plotting Type I and Type II errors; this function additionally allows us to specify the mean of the null hypothesis and the population mean that the sample is drawn from (`mean.true`):

```
> plot.type1type2.error<-function(x,
                                  x.min,
                                  x.max,
                                  qnts,
                                  mean.null,
                                  mean.true,
                                  sd,
                                  gray1,
                                  gray2,main,show.legend=TRUE){
               ## the reality:
               plot(x, dnorm(x,mean.true,sd), type = "l",ylab="",xlab="",main=main)
               ## null hypothesis distribution:
         lines(x,dnorm(x,mean.null,sd),col="black")
         abline(h = 0)

       ## plot Type II error region:

               x1 = seq(qnorm(qnts[1]), x.max, qnts[1]/5)
           y1 = dnorm(x1, mean.true, sd)

         polygon(c(x1, rev(x1)),
                 c(rep(0, length(x1)),
                 rev(y1)), col = gray2)
```

```
      ## plot Type I error region assuming alpha 0.05:

      x1 = seq(x.min, qnorm(qnts[1]), qnts[1]/5)
      y1 = dnorm(x1, mean.null, sd)
      polygon(c(x1, rev(x1)), c(rep(0, length(x1)), rev(y1)), col = gray1)

      x1 = seq(qnorm(qnts[2]), x.max, qnts[1]/5)
      y1 = dnorm(x1, mean.null, sd) ## changed
      polygon(c(x1, rev(x1)), c(rep(0, length(x1)), rev(y1)), col = gray1)

  if(show.legend==TRUE){
      legend(2,0.3, legend=c("Type I error","Type II error"),
      fill=c(gray1,gray2),cex=1)}
  }
```

The above two functions are then used within another function, `shadenormal2` (below), that plots either the Type I error probability region alone, or both Type I and Type II error probability regions. Playing with the parameter settings in this function allows us to examine the relationship between Type I and II errors.

```
> shadenormal2<-
  function (plot.only.type1=TRUE,
            alpha=0.05,
            gray1=gray(0.3), ## type I shading
            gray2=gray(0.7), ## type II shading
            x.min=-6,
            x.max=abs(x.min),
            x = seq(x.min, x.max, 0.01),
            mean.null=0,
            mean.true=-2,
            sd=1,main="",show.legend=TRUE)
  {

      qnt.lower<-alpha/2
      qnt.upper<-1-qnt.lower
      qnts<-c(qnt.lower,qnt.upper)

      if(plot.only.type1==TRUE){

       plot.type1.error(x,x.min,x.max,qnts,mean.null,sd,
       gray1,main,show.legend)

      } else { ## plot type I and type II error regions

     plot.type1type2.error(x,
                           x.min,
                           x.max,
```

```
                              qnts,
                              mean.null,
                              mean.true,
                              sd,
                              gray1,
                              gray2,main,show.legend)
        }
    }
```

```
> shadenormal2(plot.only.type1=TRUE)
```

The vertical lines in Figure 3.1 represent the 95% CI, and the shaded areas are the Type I error regions for a two-sided t-test (with probability in the two regions summing to $\alpha = 0.05$). A sample mean from one sample taken from a population with mean zero could possibly lie in this region (although it's unlikely, given the shape of the distribution), and based on that one sample, we would incorrectly decide that the null hypothesis is false when it is actually true.

In the present example we *know* there is a difference of -2 between the population means. Let's plot the *actual* (as opposed to hypothetical) sampling distribution of mean differences corresponding to this state of the world.

```
> shadenormal2(plot.only.type1=TRUE)
> xvals <- seq(-6,6,.1)
> lines(xvals,dnorm(xvals,mean=-2,sd=1),lwd=2)
```

Figure 3.2 shows the distribution corresponding to the null hypothesis overlaid with the *actual* distribution, which we *know* is centered around -2. The vertical lines are again the 95% CI, assuming the null hypothesis is true.

Now let's shade in the region that corresponds to Type II error; see Figure 3.3. Notice that the values in this region lie *within* the 95% CI of the null hypothesis. To take a specific example, given that the population means really differ by -2, if in our particular sample the difference happened to be -1, we would fail to reject H_0 even though it is false. This is true for any value in this Type II error range.

```
> shadenormal2(plot.only.type1=FALSE)
```

Some important insights emerge from Figure 3.3. First, if the true difference between the means had been not -2 but -4 (i.e., the EFFECT SIZE had been greater), then the Type II error probability (β) will go down, and therefore power ($1-\beta$) will go up. Let's confirm this visually (Figure 3.4).

```
> shadenormal2(plot.only.type1=FALSE,mean.true=-4)
```

The second insight is that if we reduce α, we also increase Type II error probability, which reduces power. Suppose α were 0.01; then the Type II error region would be as in Figure 3.5.

```
> shadenormal2(plot.only.type1=FALSE,alpha=0.01,main="alpha=.01")
```

The third insight is that as we increase sample size, the 95% confidence intervals become tighter. This decreases Type II error probability, and therefore increases power, as shown in Figure 3.6.

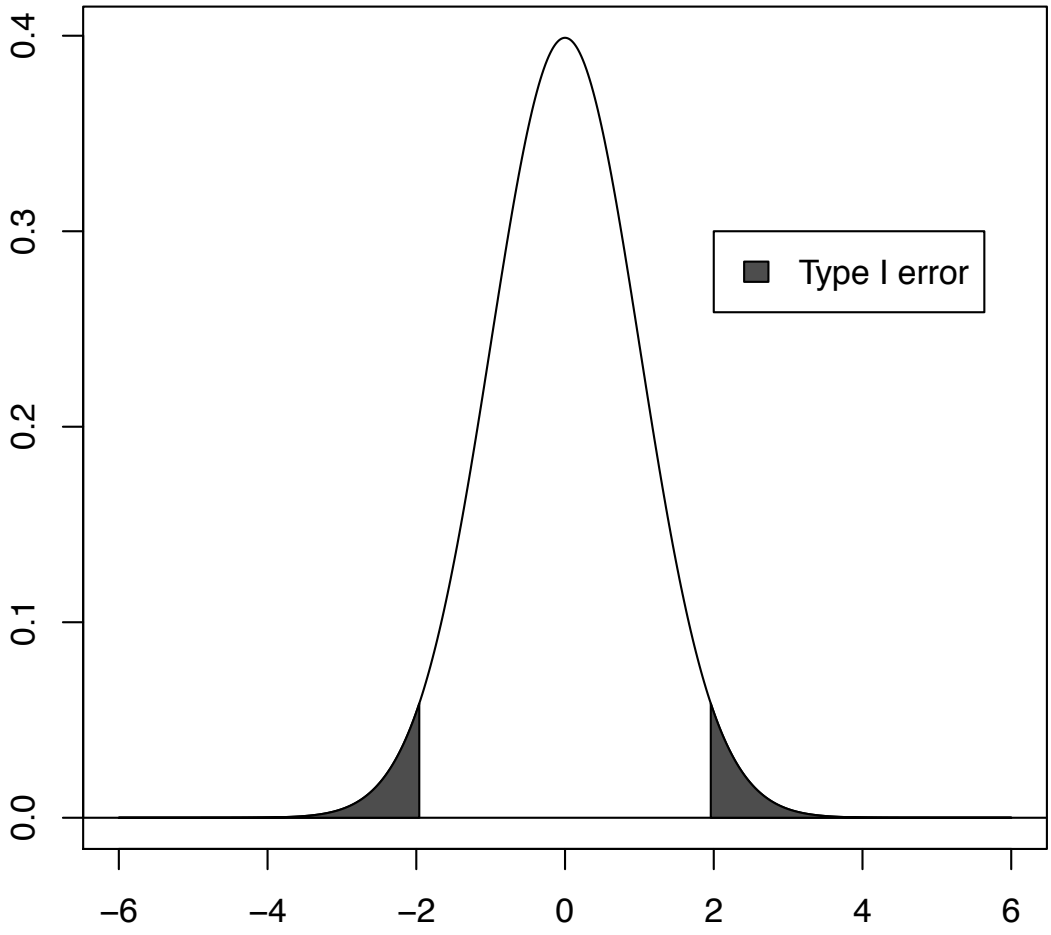


Figure 3.1: The distribution corresponding to the null hypothesis, along with rejection regions (the Type I error probability region α).

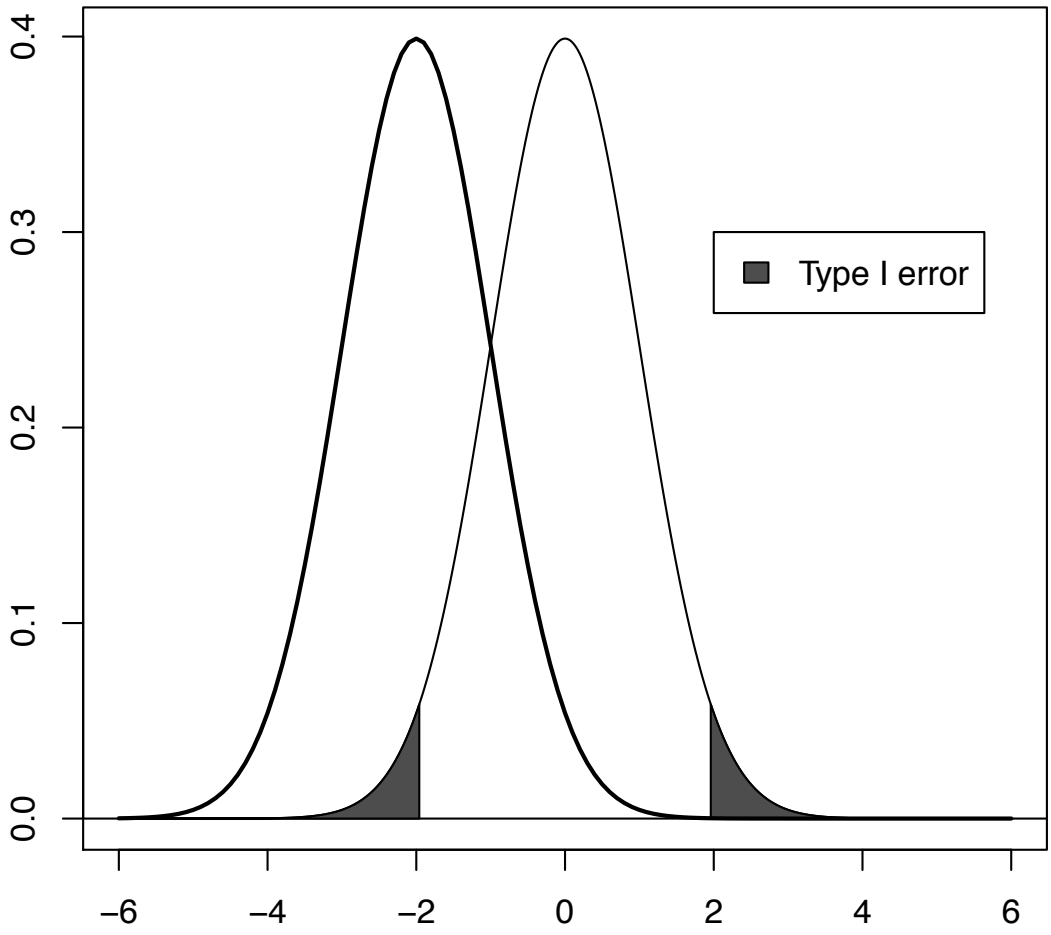


Figure 3.2: The distribution corresponding to the null hypothesis and the distribution corresponding to the true population scores.

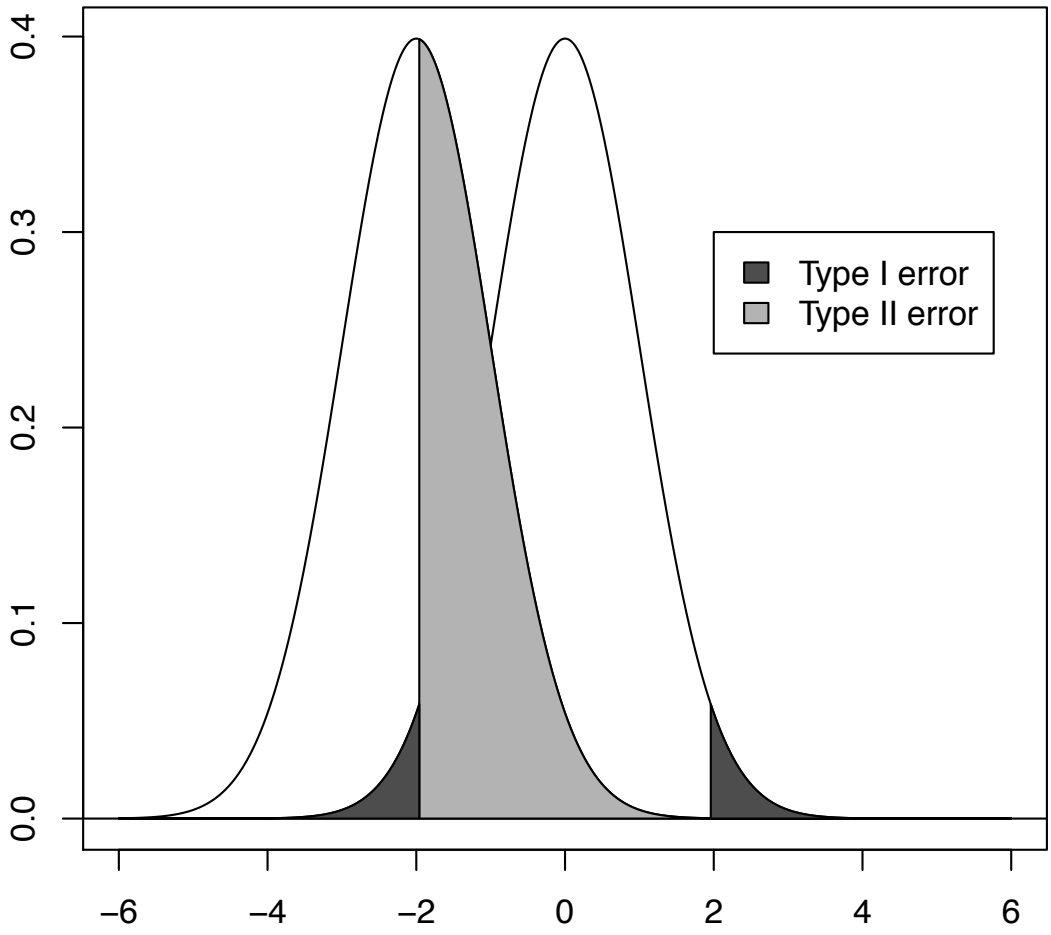


Figure 3.3: The distribution corresponding to the true population scores along with the confidence intervals from the distribution corresponding to the null hypothesis.

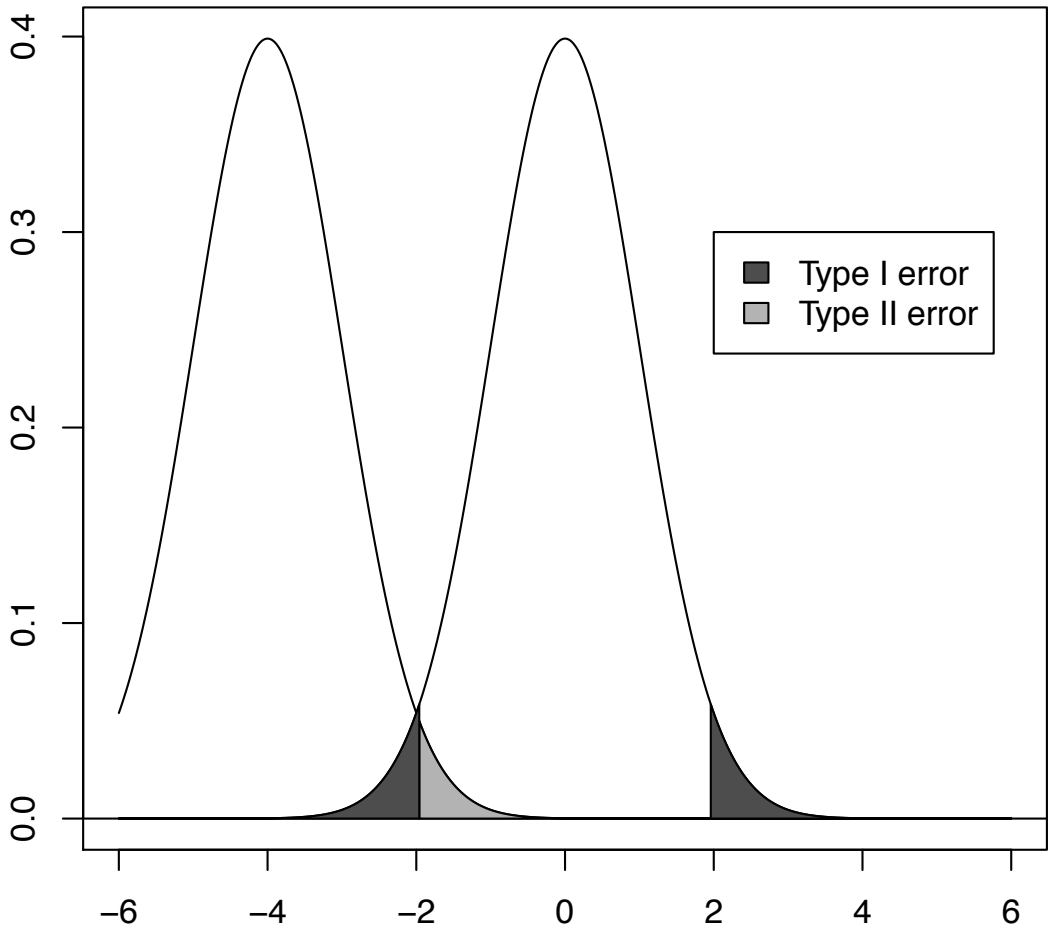


Figure 3.4: When the true difference, i.e., the effect size, increases from -2 to -4, Type II error probability decreases, and therefore power increases. Compare with Figure 3.3.

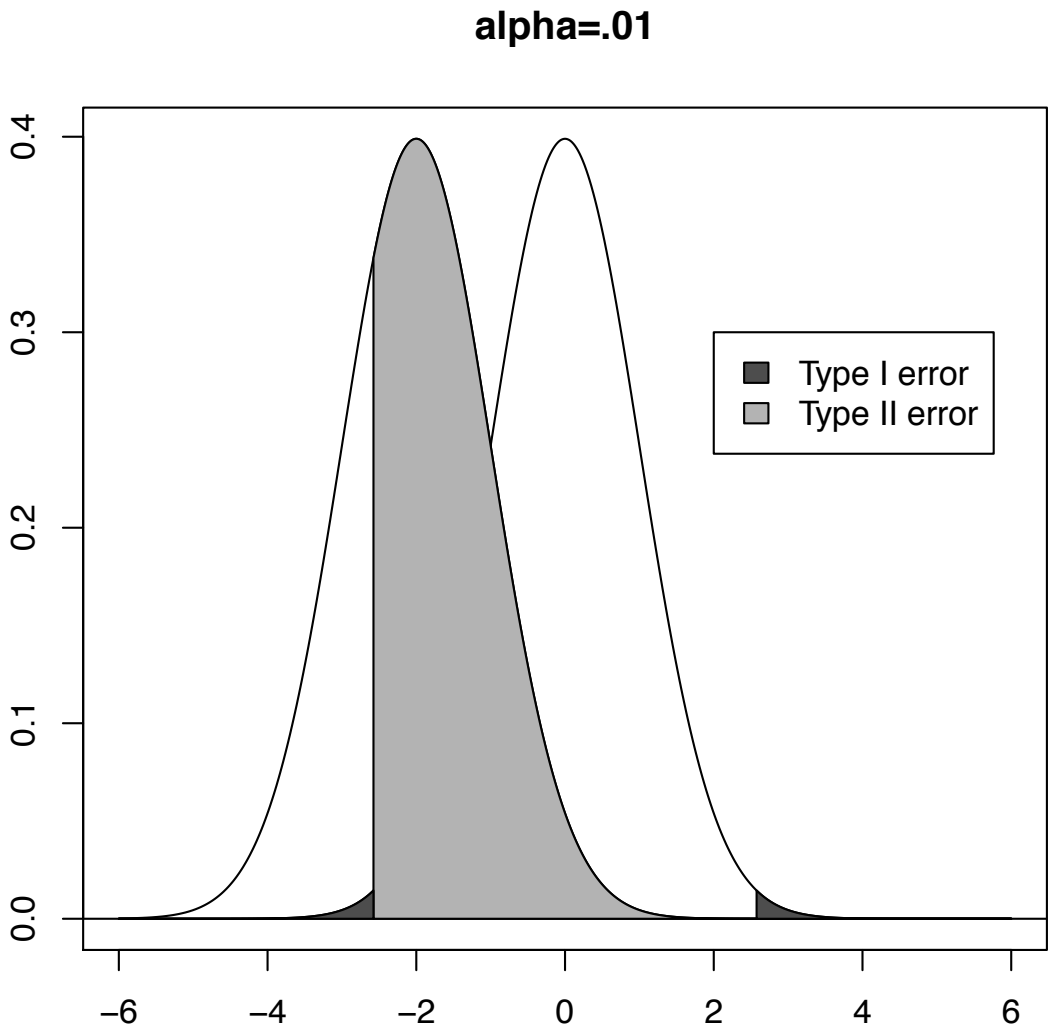


Figure 3.5: When we decrease α from 0.05 to 0.01, Type II error probability increases, and therefore power decreases (compare Figure 3.3).

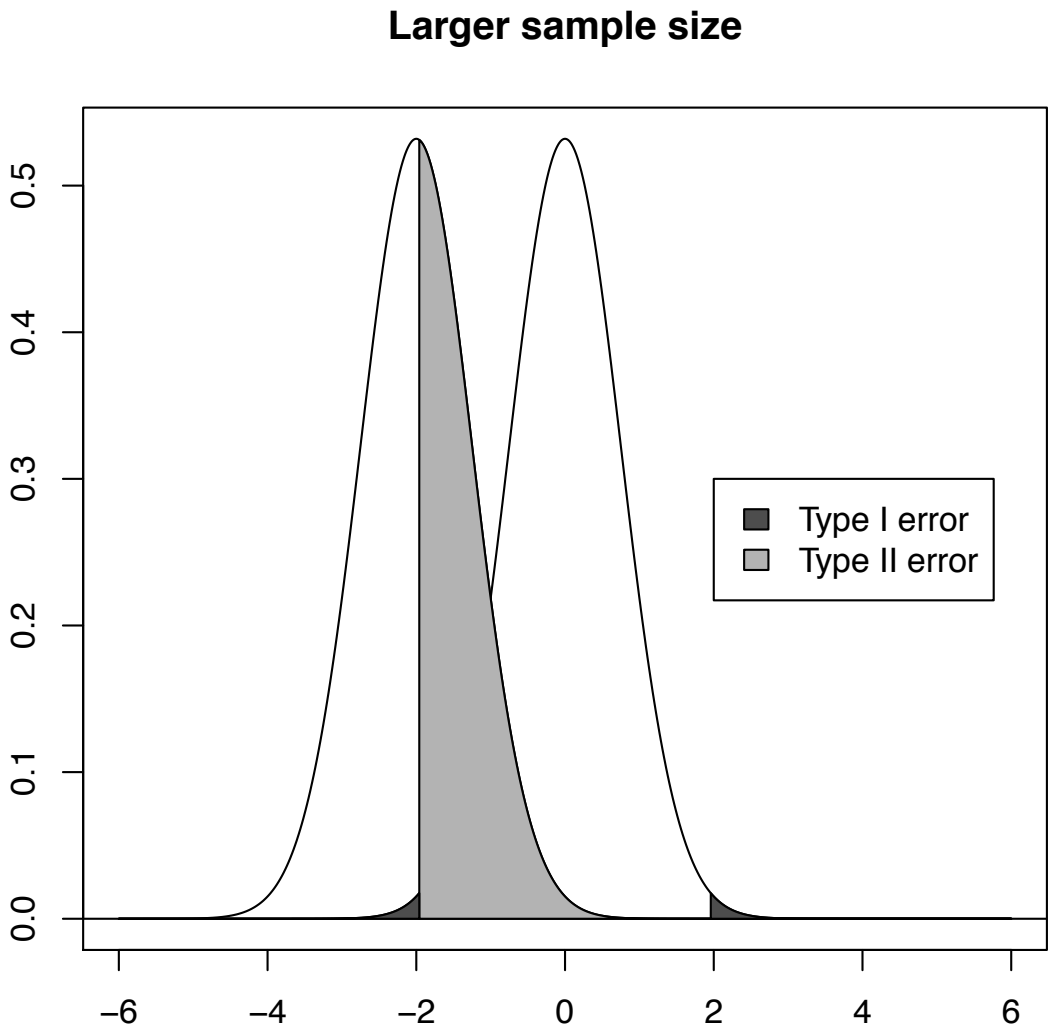


Figure 3.6: Increasing sample size will tighten 95% confidence intervals, decreasing Type II error probability, which increases power (compare with Figure 3.3).

```
> ## simulating larger sample size by decreasing SD to 0.75 from 1:
> shadenormal2(plot.only.type1=FALSE,sd=0.75,main="Larger sample size")
```

To summarize, the best situation is when we have relatively high power (low Type II error probability) and low Type I error probability (α). By convention, we keep α at 0.05. We usually do not want to change that: lowering α is costly in the sense that it reduces power as well, as we just saw. What we do want to ensure is that power is reasonably high; after all, why would you want to do an experiment where you have only 50% power or less? That would mean that you have an a priori chance of finding a true effect (i.e., an effect that is actually present in nature) only 50% of the time or less. As we just saw, we can increase power by increasing sample size, and/or by increasing the sensitivity of our experimental design so that we have larger effect sizes.

Researchers in psycholinguistics and other areas often do experiments with low power (for logistical or other reasons); it is not unheard of to publish reading studies (eyetracking or self-paced reading, etc.) or event-related potentials studies with 12 or 20 participants. This is not a serious problem if we succeed in getting the significant result that was predicted when the study was run. However, when we get a null (nonsignificant) result, it would be a mistake to conclude that no true effect exists (i.e., it would be a mistake to argue for the null hypothesis). If power is low, the chance of missing an effect that is actually present is high, so we should avoid concluding anything from a null result in this situation.

We would like to make four observations here:

1. At least in areas such as psycholinguistics, the null hypothesis is, strictly speaking, usually always false: When you compare reading times or any other dependent variable in two conditions, the a priori chance that the two means to be compared are *exactly* identical is low. The interesting question therefore is not whether the null hypothesis is false, but by how much (the effect size), and the sign (positive or negative) of the difference between the means being compared.

2. One can in principle nearly always get a statistically significant effect given a large enough sample size; the question is whether the effect is large enough to be theoretically important and whether the difference between means has the expected sign.

3. Especially in areas like psycholinguistics, replication is not given the importance it deserves. Note that we run a 5% risk of declaring a significant difference when in fact there is none (or effectively none, see point 1 above). Replication is an important method to convince oneself that an effect is truly present. High power, reasonably large effect sizes, and actually replicated results should be your goal in experimental science. A p-value of less than 0.05 does not tell you that you got a "true effect". Fisher himself pointed this out:

   > It is usual and convenient for experimenters to take-5 per cent. as a standard level of significance, in the sense that they are prepared to ignore all results which fail to reach this standard, and, by this means, to eliminate from further discussion the greater part of the fluctuations which chance causes have introduced into their experimental results. No such selection can eliminate the whole of the possible effects of chance. coincidence, and if we accept this convenient convention, and agree that an event which would occur by chance only once in 70 trials is decidedly "significant," in the statistical sense, we thereby admit that **no isolated**

> **experiment, however significant in itself, can suffice for the experimental demonstration of any natural phenomenon; for the "one chance in a million" will undoubtedly occur, with no less and no more than its appropriate frequency**, however surprised we may be that it should occur to us. In order to assert that a natural phenomenon is experimentally demonstrable we need, not an isolated record, but a reliable method of procedure. In relation to the test of significance, we may say that a phenomenon is experimentally demonstrable when we know how to conduct an experiment which will rarely fail to give us a statistically significant result.

4. Many researchers also believe that the lower the p-value, the lower the probability that the null hypothesis is true. However, as discussed earlier, this is a misunderstanding that stems from a failure to attend to the conditional probabilities involved.

It follows from the above discussion that if you have a relatively narrow CI, and a nonsignificant result ($p > .05$), you have relatively high power and a relatively low probability of making a Type II error (accepting the null hypothesis as true when it is in fact false). This is particularly important for interpreting null results (results where the p-value is greater than 0.05). [3] suggest a heuristic: if you have a narrow CI, and a nonsignificant result, you have some justification for concluding that the null hypothesis may in fact be effectively true. Conversely, if you have a wide CI and a nonsignificant result the result really is inconclusive.

3.2 Computing sample size for a t-test using R

Suppose you want to compute the sample size you need to have power 0.8 and alpha 0.05. I look at the paired sample case.

You need to decide on a few things:

- The magnitude of the effect you expect (based on previous work, or theory): delta (the difference between the means you're comparing)
- The standard deviation of the difference that between means that you expect (also based on previous data or theory)
- Your significance level (alpha)
- The power you want (APA recommends at least 0.80)
- Type of t-test (usually a paired t-test in psycholinguistics)
- The type of alternative hypothesis you have (we will always do the two-sided case)

```
> power.t.test(n = NULL, delta = 100, sd = 100, sig.level = 0.05,
               power = 0.80,
               type = c("paired"),
               alternative = c("two.sided"),
               strict = FALSE)
```

```
     Paired t test power calculation

              n = 9.9379
          delta = 100
             sd = 100
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE: n is number of *pairs*, sd is std.dev. of *differences* within pairs
```

The calculation suggests a sample size of 10 (rounding up). For other tests there are comparable functions in R, just look in the R help.

3.3 The form of the power function

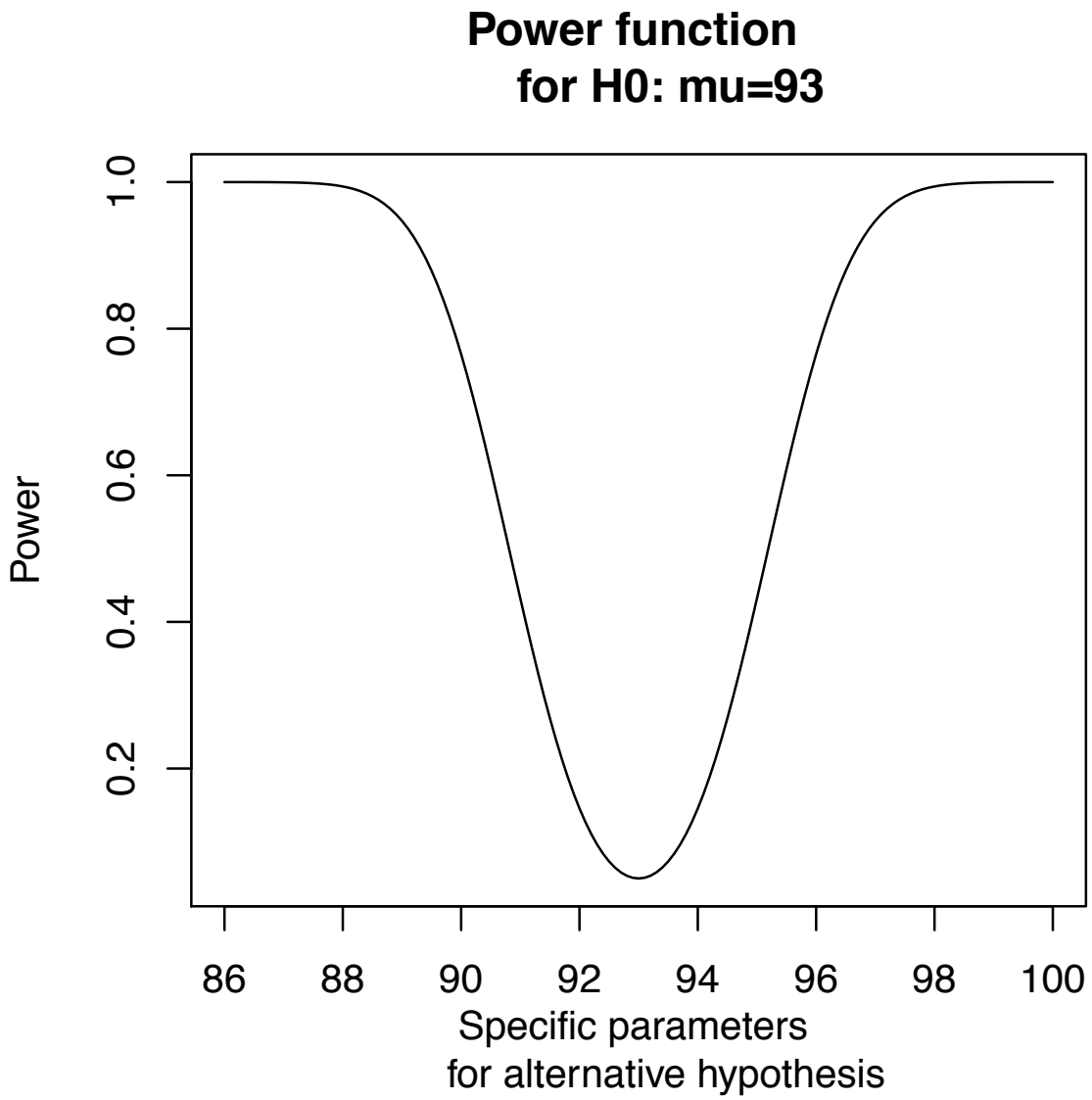
Assume that the null hypothesis has mean 93, true sd is 5, and sample size is 20. If the null were true, the rejection region would be bounded by

```
> qnorm(0.025,mean=93,sd=5/sqrt(20))

[1] 90.809

> qnorm(0.975,mean=93,sd=5/sqrt(20))

[1] 95.191
```

Intuitively, power will go up if the true value of the mean is far from the hypothesized mean, in either direction (greater than, or less than the hypothesized mean). We can look at how power changes as a function of how far away the true mean is from the hypothesized mean:

```
> sd<-5
> n<-20
> power.fn<-function(mu){
   ## lower and upper bounds of rejection regions
   ## defined by null hypothesis mean:
   lower<-qnorm(0.025,mean=93,sd=5/sqrt(20))
   upper<-qnorm(0.975,mean=93,sd=5/sqrt(20))

   ## lower rejection region:
   z.l<-(lower-mu)/(sd/sqrt(n))
   ## upper rejection region for given true mu:
   z.u<-(upper-mu)/(sd/sqrt(n))
   ## return rejection probability:
       return(pnorm(z.u,lower.tail=F)+
              pnorm(z.l,lower.tail=T))}
> ## a range of true values:
```

```
> alt<-seq(86,100,by=0.1)
> pow<-power.fn(alt)
> plot(alt,pow,type="l",
   xlab="Specific parameters
        for alternative hypothesis",
        ylab="Power",main="Power function
        for H0: mu=93")
```



What do you think would be the shape of this function if you increased sample size? What would be the shape if you increased standard deviation? First try to sketch the shapes on paper, and then use the above function to display the shape.

3.4 ADVANCED OPTIONAL SECTION: Computing the power function

[You don't need to know this material, but I added it in case anyone is interested in finding out how exactly the power calculation is done.]

As an example, let $H_0 : \mu = 93$, and let $H_1 : \mu \neq 93$. Assume that population sd σ and sample size n are given. Note that in realistic situations we don't know σ but we can estimate it using s.

We can get a sample mean that is greater than μ or one that is smaller than μ. Call these $\bar{x}_g$ and $\bar{x}_s$ respectively.

In the case where we know σ, the test **under the null hypothesis** is:

$$\frac{\bar{x}_g - 93}{\sigma/\sqrt{n}} > 1.96 \quad \text{or} \quad \frac{\bar{x}_s - 93}{\sigma/\sqrt{n}} > -1.96 \tag{3.1}$$

Solving for the two $\bar{x}$'s, we get:

$$\bar{x}_g > 1.96\frac{\sigma}{\sqrt{n}} + 93 \quad \text{or} \quad \bar{x}_s > -1.96\frac{\sigma}{\sqrt{n}} + 93 \tag{3.2}$$

Now, power is the probability of rejecting the null hypothesis when the mean is whatever the alternative hypothesis mean is (say some specific value μ).

That, the test **under the alternative hypothesis** is:

$$\frac{\bar{x}_g - \mu}{\sigma/\sqrt{n}} > 1.96 \quad or \quad \frac{\bar{x}_s - \mu}{\sigma/\sqrt{n}} < -1.96 \tag{3.3}$$

We can replace the $\bar{x}_g$ with its full form, and do the same with $\bar{x}_s$.

$$\frac{1.96\frac{\sigma}{\sqrt{n}} + 93 - \mu}{\sigma/\sqrt{n}} > 1.96 \quad or \quad \frac{-1.96\frac{\sigma}{\sqrt{n}} + 93 - \mu}{\sigma/\sqrt{n}} < -1.96 \tag{3.4}$$

I can rewrite the above as:

$$\frac{1.96\frac{\sigma}{\sqrt{n}} - (\mu - 93)}{\sigma/\sqrt{n}} > 1.96 \quad or \quad \frac{-1.96\frac{\sigma}{\sqrt{n}} - (\mu - 93)}{\sigma/\sqrt{n}} < -1.96 \tag{3.5}$$

Simplifying:

$$1.96 - \frac{(\mu - 93)}{\sigma/\sqrt{n}} > 1.96 \quad or \quad -1.96 - \frac{(\mu - 93)}{\sigma/\sqrt{n}} < -1.96 \tag{3.6}$$

This is now easy to solve! I will use R's pnorm function in the equation below, simply because we haven't introduced a symbol for pnorm in this course. We can rewrite the above expression as:

$$[1 - pnorm(1.96 - \frac{(\mu - 93)}{\sigma/\sqrt{n}})] + pnorm(-1.96 - \frac{(\mu - 93)}{\sigma/\sqrt{n}}) \tag{3.7}$$

The above equation allows us to

- compute sample size for any given null (here 93) and alternative hypotheses, provided I have the population standard deviation.
- compute power given a null and alternative hypothesis, population standard deviation, and sample size.

Example: suppose I need power of 0.99 for $H_0 : \mu = 93$ and $H_1 : \mu = 98$, $\sigma = 5$.
For this example, what sample size do I need? I take the above equation and fill in the values:

$$[1 - pnorm(1.96 - \frac{(98-93)}{5/\sqrt{n}})] + pnorm(-1.96 - \frac{(98-93)}{5/\sqrt{n}}) \tag{3.8}$$

Simplifying, this gives us:

$$[1 - pnorm(1.96 - \sqrt{n})] + pnorm(-1.96 - \sqrt{n}) \tag{3.9}$$

Note that the second term will be effectively zero for some reasonable n like 10:

```
> pnorm(-1.96-sqrt(10))

[1] 1.5093e-07
```

So we can concentrate on the first term:

$$[1 - pnorm(1.96 - \sqrt{n})] \tag{3.10}$$

If the above has to be equal to 0.99, then

$$pnorm(1.96 - \sqrt{n}) = 0.01 \tag{3.11}$$

So, we just need to find the value of the z-score that will give us a probability of approximately 0.01. You can do this analytically (exercise), but you could also play with some values of n to see what you get. The answer is $n = 18$.

```
> pnorm(1.96-sqrt(18))

[1] 0.011226
```

3.5 Stopping rules

Psycholinguists and psychologists often adopt the following type of data-gathering procedure. The experimenter gathers n data points, then checks for significance ($p < 0.05$ or not). If it's not significant, he gets more data (n more data points). Since time and money are limited, he might decide to stop anyway at sample size, say, some multiple of n. One can play with different scenarios here. A typical n might be 15.

This approach would give us a range of p-values under repeated sampling. Theoretically, under the standard assumptions of frequentist methods, we expect a Type I error to be 0.05. This is the case in standard analyses (I also track the t-statistic, in order to compare it with my stopping rule code below).

```
> ##Standard:
> pvals<-NULL
> tstat_standard<-NULL
> n<-10
> nsim<-1000
> ## assume a standard dev of 1:
```

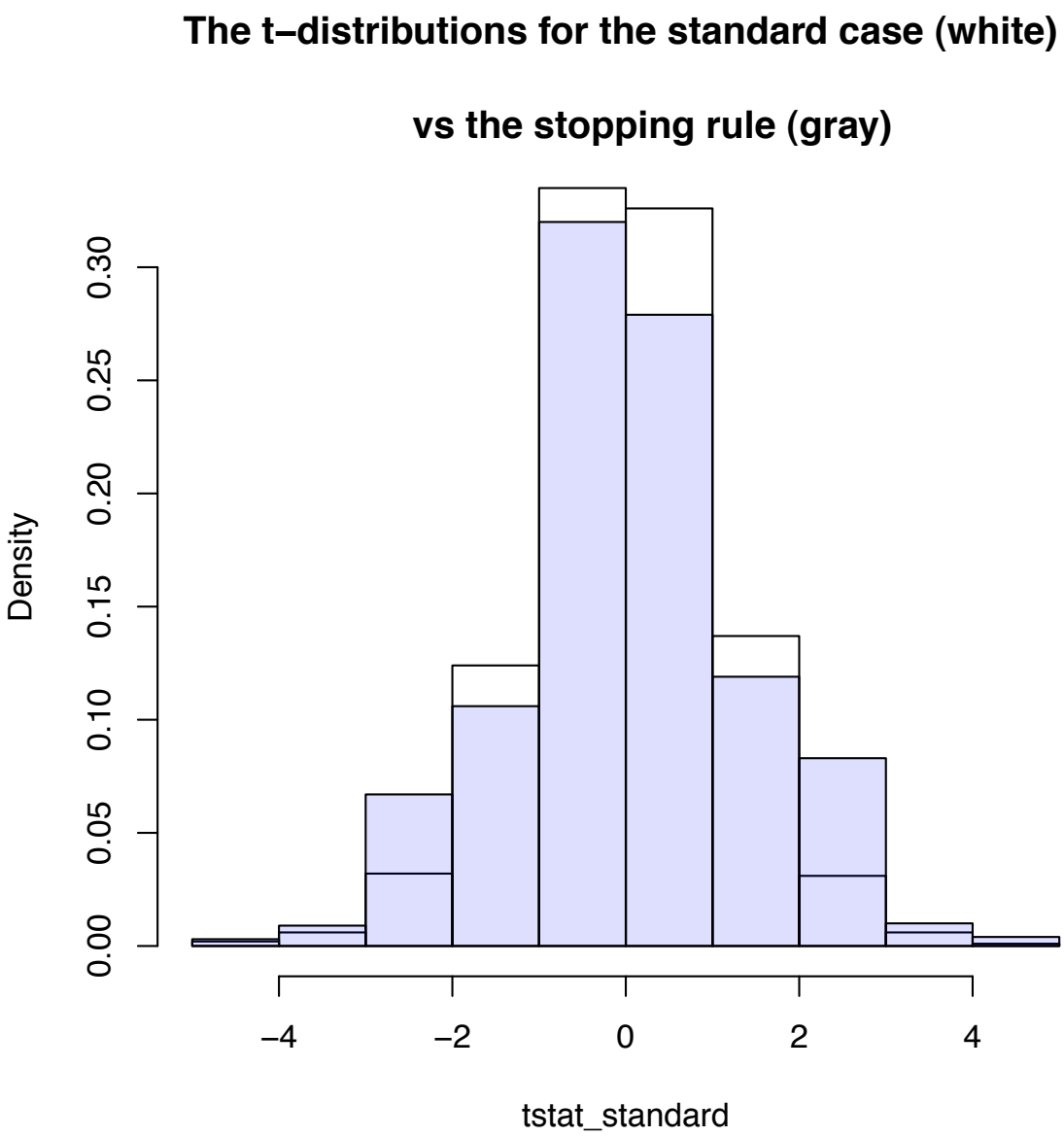
```
> stddev<-1
> mn<-0
> for(i in 1:nsim){
     samp<-rnorm(n,mean=mn,sd=stddev)
     pvals[i]<-t.test(samp)$p.value
     tstat_standard[i]<-t.test(samp)$statistic
   }
> ## Type I error rate: about 5% as theory says:
> table(pvals<0.05)[2]/nsim

 TRUE
0.048
```

But the situation quickly deteriorates as soon as adopt the strategy I outlined above. I will also track the distribution of the t-statistic below.

```
> pvals<-NULL
> tstat<-NULL
> ## how many subjects can I run?
> upper_bound<-n*6
> for(i in 1:nsim){
   ## at the outset we have no significant result:
     significant<-FALSE
   ## null hyp is going to be true,
   ## so any rejection is a mistake.
   ## take sample:
     x<-rnorm(n,mean=mn,sd=stddev)
   while(!significant & length(x)<upper_bound){
     ## if not significant:
   if(t.test(x)$p.value>0.05){
     ## get more data:
     x<-append(x,rnorm(n,mean=mn,sd=stddev))
     ## otherwise stop:
   } else {significant<-TRUE}
   }
   ## will be either significant or not:
   pvals[i]<-t.test(x)$p.value
   tstat[i]<-t.test(x)$statistic
   }
> ## Type I error rate:
> ## much higher than 5%:
> table(pvals<0.05)[2]/nsim

 TRUE
0.176
```

Now let's compare the distribution of the t-statistic in the standard case vs with the above stopping rule:

```
> hist(tstat_standard,main="The t-distributions for the standard case (white) \n
        vs the stopping rule (gray)",freq=F)
> hist(tstat,add=T,col="#0000ff22",freq=F)
```



We get fatter tails with the above stopping rule.

The point is that one should fix one's sample size in advance based on a power analysis, not deploy a stopping rule like the one above; if we used such a stopping rule, we are much more likely to incorrectly declare a result as statistically significant.

Chapter 4

Linear models

4.1 Introduction

Here's a data set from Maxwell and Delaney's book (this example was presented in the Baron and Li tutorial, available from the CRAN home page). This is within-subjects fake experimental data; it has a 2×3 design. Imagine that subjects are shown a stimulus (a picture) on the screen, and it's either shown with no noise (distortion, say) in the background, or with noise; in addition, the stimulus was either horizontal, tilted by 4 degrees, or tilted by 8 degrees. That it, the experiment has two levels of noise, and three levels of tilt.

Set up data first:

```
> MD497.dat <- matrix(c(
   420, 420, 480, 480, 600, 780,
   420, 480, 480, 360, 480, 600,
   480, 480, 540, 660, 780, 780,
   420, 540, 540, 480, 780, 900,
   540, 660, 540, 480, 660, 720,
   360, 420, 360, 360, 480, 540,
   480, 480, 600, 540, 720, 840,
   480, 600, 660, 540, 720, 900,
   540, 600, 540, 480, 720, 780,
   480, 420, 540, 540, 660, 780),
   ncol = 6, byrow = T)  # byrow=T so the matrix's layout is exactly like this
> MD497.df <- data.frame(
   rt    = as.vector(MD497.dat),
   subj  = factor(rep(paste("s", 1:10, sep=""), 6)),
   deg   = factor(rep(rep(c(0,4,8), c(10, 10, 10)), 2)),
   noise = factor(rep(c("no.noise", "noise"), c(30, 30))))
```

Let's compute the means by factor levels:

```
> means<-with(MD497.df,tapply(rt,IND=list(noise,deg),mean))
```

```
           0   4   8
no.noise 462 510 528
noise    492 660 762
```

This is how one would do a t-test with such data, to compare means across (sets of) conditions:

```
> no.noise<-subset(MD497.df,noise=="no.noise")
> no.noise.means<-with(no.noise,tapply(rt,subj,mean))
> noise<-subset(MD497.df,noise=="noise")
> noise.means<-with(noise,tapply(rt,subj,mean))
> t.test(noise.means,no.noise.means,paired=TRUE)

        Paired t-test

data:  noise.means and no.noise.means
t = 5.8108, df = 9, p-value = 0.000256
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  84.277 191.723
sample estimates:
mean of the differences
                    138
```

These are the means we are comparing in the noise case:

```
> ## means of noise levels:
> with(MD497.df,tapply(rt,noise,mean))

no.noise    noise
     500      638

> ## mean of no.noise=500
> ## mean of noise=500+138=638
```

And here are the means we would compare in the degree case if we were to do pairwise t-tests on the different levels (0 vs 4 deg, 0 vs 8 deg, for example):

```
> with(MD497.df,tapply(rt,deg,mean))

  0   4   8
477 585 645

> ## mean of 0 deg=477
> ## mean of 4 deg=477+108=585
> ## mean of 8 deg=477+168=645
```

Now, as for noise, we could fit t-tests repeatedly for degree as well. (As an exercise, you may want to stop reading and try doing this right now: compare 0 with 4 degrees, and 0 with 8 degrees using t-tests).

Now consider this **linear model**, which evaluates rt as a function of noise (**Note: this model is incorrect for the present dataset, but it's useful to understand how it work in order to explain how linear mixed models work**):

```
> m0<-lm(rt~noise,MD497.df)

> results.m0<-extractfit(m0,
                         coln=c("","coef.",
                                "SE","t-value"),
                         indices=1:2,
                         fac=c("Intercept","noise"))
> library(xtable)
> myxtable<-function(res,cap,lab){
     print(xtable(res,caption=cap,label=lab),
           include.rownames=F)}
```

	coef.	SE	t-value
Intercept	500.00	22.08	22.65
noise	138.00	31.23	4.42

Table 4.1: The effect of noise on reaction time.

It's important to understand what lies behind this output. We won't go through everything in the output, but only focus on some aspects that are immediately of relevance to us. First, we have the coefficients, which define the intercept and slope of the fitted line respectively:

```
> ## coefficients:
> coef(m0)

(Intercept)  noisenoise
        500         138
```

One instructive exercise is to compare these coefficients with the means we have for noise. The mean of no.noise is 500, and the mean of noise is 500+138=638:

```
> ## means of noise levels:
> with(MD497.df,tapply(rt,noise,mean))

no.noise    noise
     500      638

> ## mean of no.noise=500
> ## mean of noise=500+138=638
```

What do you think the above coefficients mean? Think about this before reading further.

Next, we have the residuals, whose distribution can be compared to a normal distribution (it's an assumption of the linear model that the residuals be normal; [7] has more details on linear models):

We can extract the residuals:

```
> ## residuals:
> res.m0<-residuals(m0)
```

```
> library(car)
> qqPlot(res.m0)
```

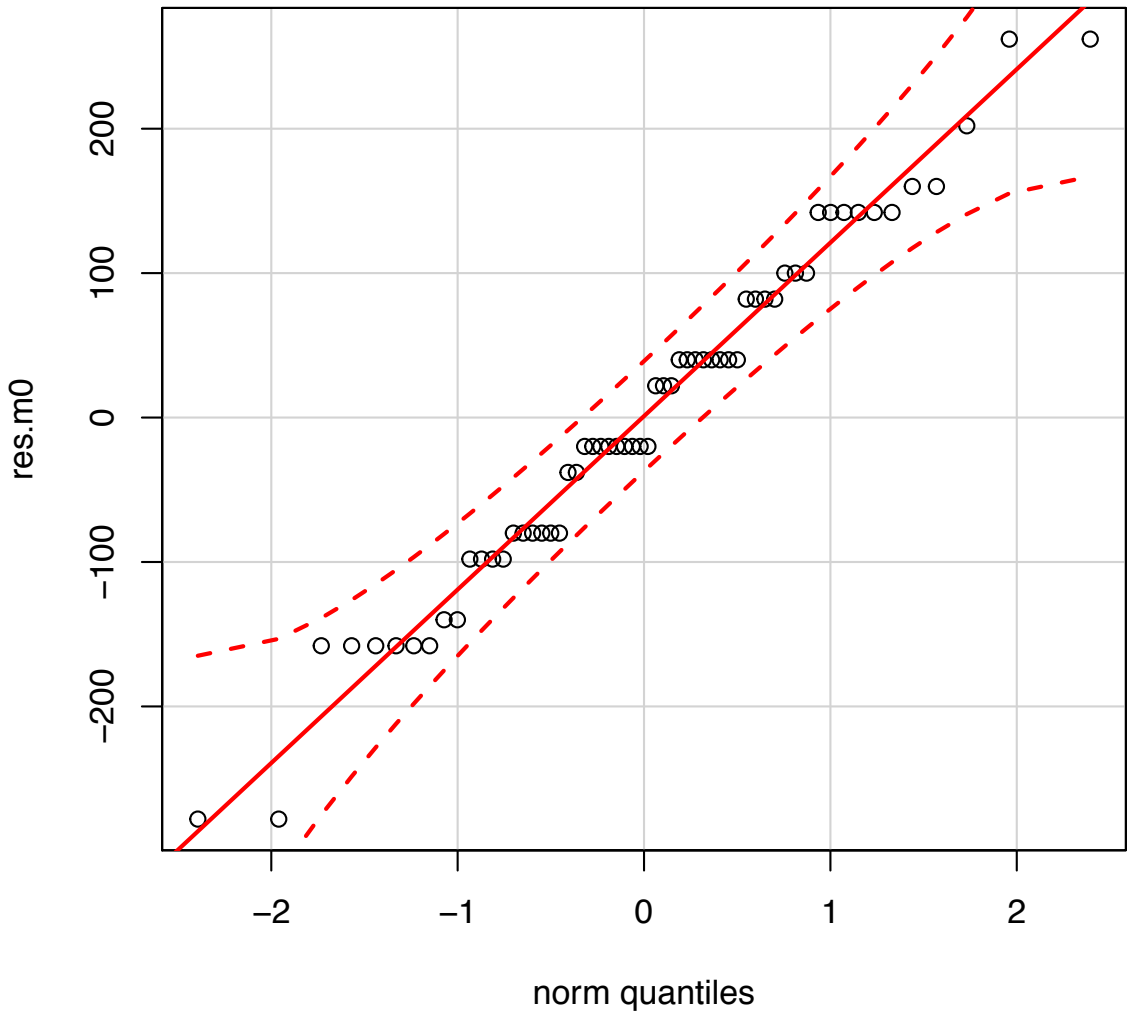


Figure 4.1: Residuals of the model m0.

and plot them by comparing them to a normal distribution:

And underlyingly, we have a design matrix or model matrix, which is being used by R to estimate the coefficients:

```
> ##
> head(model.matrix(m0),n=7)
```

```
  (Intercept) noisenoise
1           1          0
2           1          0
3           1          0
4           1          0
5           1          0
6           1          0
7           1          0
```

It's worth understanding a little bit about what these parts of the linear model are. Our linear model equation for the noise model m0 above is a system of equations. The single equation:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \tag{4.1}$$

can be expanded to:

$$\begin{array}{rcccccc} Y_1 & = & \beta_0 & + & X_1\beta_1 & + & \varepsilon_1 \\ Y_2 & = & \beta_0 & + & X_2\beta_1 & + & \varepsilon_2 \\ Y_3 & = & \beta_0 & + & X_3\beta_1 & + & \varepsilon_3 \\ Y_4 & = & \beta_0 & + & X_4\beta_1 & + & \varepsilon_4 \\ \vdots & & \vdots & & \vdots & & \vdots \\ Y_n & = & \beta_0 & + & X_n\beta_1 & + & \varepsilon_n \end{array} \tag{4.2}$$

And this system of linear equations can be restated in compact matrix form:

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon \tag{4.3}$$

where

Vector of responses:

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ \vdots \\ Y_n \end{pmatrix} \tag{4.4}$$

The design matrix (in R this is called the model matrix):

$$\mathbf{X} = \begin{pmatrix} 1 & X_1 \\ 1 & X_2 \\ 1 & X_3 \\ 1 & X_4 \\ \vdots & \vdots \\ 1 & X_n \end{pmatrix} \tag{4.5}$$

Vector of parameters to be estimated:

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \tag{4.6}$$

and
Vector of error terms (residuals):

$$\varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \vdots \\ \varepsilon_n \end{pmatrix} \tag{4.7}$$

We could write the whole equation as:

$$\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} 1 & X_1 \\ 1 & X_2 \\ 1 & X_3 \\ 1 & X_4 \\ \vdots & \vdots \\ 1 & X_n \end{pmatrix} \times \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \vdots \\ \varepsilon_n \end{pmatrix} \tag{4.8}$$

Our top goal when we fit a linear model is to find estimates of the parameters β_0 and β_1, the intercept and slope respectively; we will call the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$. This can be done by "solving" for the beta's. X is the model matrix, and X$'$ is the transpose of the model matrix. Y is the vector of dependent variables.

$$\beta = (X'X)^{-1}X'Y \tag{4.9}$$

You do not need to know how the above equation comes about; all you need to know is that given X, and Y, we can estimate the parameters. In case you are interested in more details, the Sen and Srivastava textbook will give you a fuller introduction.

Returning to our noise and deg(ree) example above, we can also fit a linear model where we examine the effect of deg on rt.

Here, a critical thing to attend to is the **contrast coding** for the factor degree:

```
> contrasts(MD497.df$deg)
```

```
  4 8
0 0 0
4 1 0
8 0 1
```

The above contrast coding says the following: compare 0 with deg 4, and deg 0 with deg 8 (i.e., deg 0 is the baseline). This kind of contrast is called treatment contrast coding: there's always a baseline that you compare another condition with.

Let's fit the model:

```
> ## evaluating effect of degree:
>
> summary(m1<-lm(rt~deg,MD497.df))

Call:
lm(formula = rt ~ deg, data = MD497.df)

Residuals:
   Min     1Q Median     3Q    Max
  -285   -105      3     75    255

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    477.0       27.2   17.53  < 2e-16
deg4           108.0       38.5    2.81   0.0068
deg8           168.0       38.5    4.37  5.4e-05

Residual standard error: 122 on 57 degrees of freedom
Multiple R-squared:  0.256,        Adjusted R-squared:  0.23
F-statistic: 9.79 on 2 and 57 DF,  p-value: 0.000221

> ## compare the coefficients with means computed earlier:
```

Compare the coefficients with these means for deg:

```
> with(MD497.df,tapply(rt,deg,mean))

  0   4   8
477 585 645

> ## mean of 0 deg=477
> ## mean of 4 deg=477+108=585
> ## mean of 8 deg=477+168=645
```

What do you think the coefficients mean? Stop and work this out before reading further.

We can also examine the effects of noise and degree together:

```
> ## evaluating effect of noise AND degree together:
> m2<-lm(rt~noise+deg,MD497.df)
> results.m2<-extractfit(m2,
                         coln=c("","coef.",
                                "SE","t-value"),
                         indices=1:4,
                         fac=c("Intercept","noise","deg 4 vs 0","deg 8 vs 0"))

> myxtable(results.m2,
           cap="The effect of noise and degree on reaction time.",
           lab="tab:m2")
```

	coef.	SE	t-value
Intercept	408.00	25.78	15.82
noise	138.00	25.78	5.35
deg 4 vs 0	108.00	31.58	3.42
deg 8 vs 0	168.00	31.58	5.32

Table 4.2: The effect of noise and degree on reaction time.

What do the coefficients mean?

In our current noise and deg example, the 'predictors' are categorial ones. What about when we have continuous predictors, such as instructors' beauty levels measured on a continuous scale as predictors of their teaching evaluations? Beauty levels are centered; this means that a beauty level of 0 means average beauty level. This is a data set from a paper by Hamermesh and Parker (Beauty in the Classroom: Instructors' Pulchritude and Putative Pedagogical Productivity," Economics of Education Review, August 2005). I got the data from [2].

```
> ## Example with a continuous predictor:
>
> ## teacher's evaluations as a function of their beauty score:
> bdata <- read.table("beauty.txt",header=T)
> head(bdata)

    beauty evaluation
1  0.20157        4.3
2 -0.82608        4.5
3 -0.66033        3.7
4 -0.76631        4.3
5  1.42145        4.4
6  0.50022        4.2

> m3<-lm(evaluation~beauty,bdata)

> results.m3<-extractfit(m3,
                         coln=c("","coef.",
                                "SE","t-value"),
                         indices=1:2,
                         fac=c("Intercept","beauty"))
```

	coef.	SE	t-value
Intercept	4.01	0.03	157.21
beauty	0.13	0.03	4.13

Table 4.3: The effect of beauty level on teaching evaluation score.

Think about what the coefficients mean. The point of this slight digression in looking at this beauty data is to realize that the linear model provides a general approach for evaluating the effect of variable X on dependent variable Y.

Returning to our noise and deg data, one important point we've neglected is that different subjects have different effects of noise and deg. In the linear models above we are ignoring this.

```
> ## returning to our noise data (MD497.df):
> ## here's an important fact about our data:
> # different subjects have different means for no.noise and noise
> # and different means for the three levels of deg
>
> means.noise<-with(MD497.df,tapply(rt,list(subj,noise),mean))

    no.noise noise
s1       440   620
s10      480   660
s2       460   480
s3       500   740
s4       500   720
s5       580   620
s6       380   460
s7       520   700
s8       580   720
s9       560   660

> means.deg<-with(MD497.df,tapply(rt,list(subj,deg),mean))

      0   4   8
s1  450 510 630
s10 510 540 660
s2  390 480 540
s3  570 630 660
s4  450 660 720
s5  510 660 630
s6  360 450 450
s7  510 600 720
s8  510 660 780
s9  510 660 660
```

We can view the differential behavior of subjects in a graph (Figures 4.2 and 4.3).

```
> ## We can visualize these differences graphically:
>
> library(lattice)
> ## noise by subject (data points):
> print(xyplot(rt~noise|subj,
           panel=function(x,y,...){panel.xyplot(x,y,type="r")},MD497.df))
```

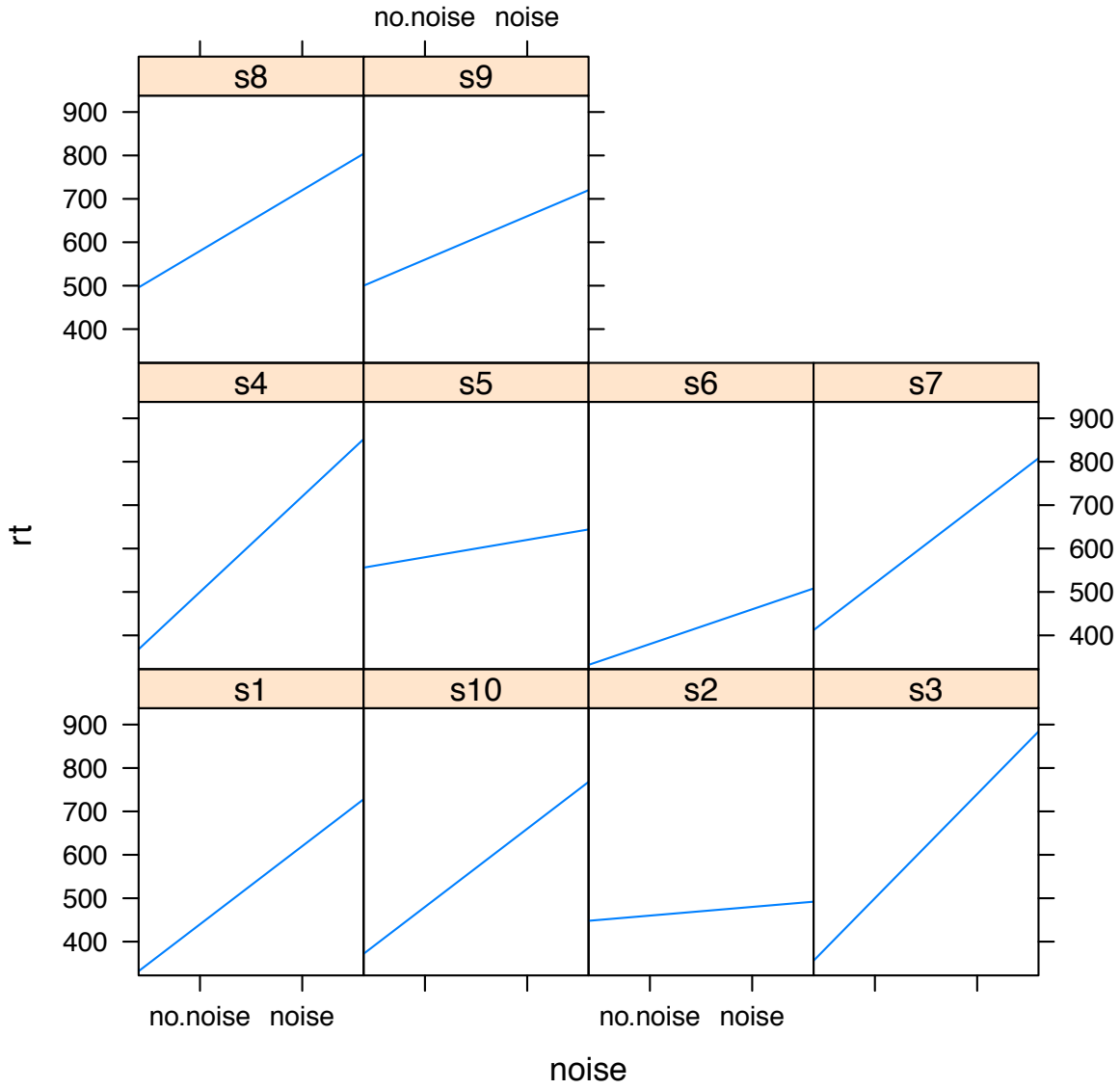


Figure 4.2: Noise effects by subject.

```
> ## same as above, but for deg:
> print(xyplot(rt~deg|subj,
           panel=function(x,y,...){panel.xyplot(x,y,type="r")},MD497.df))
```

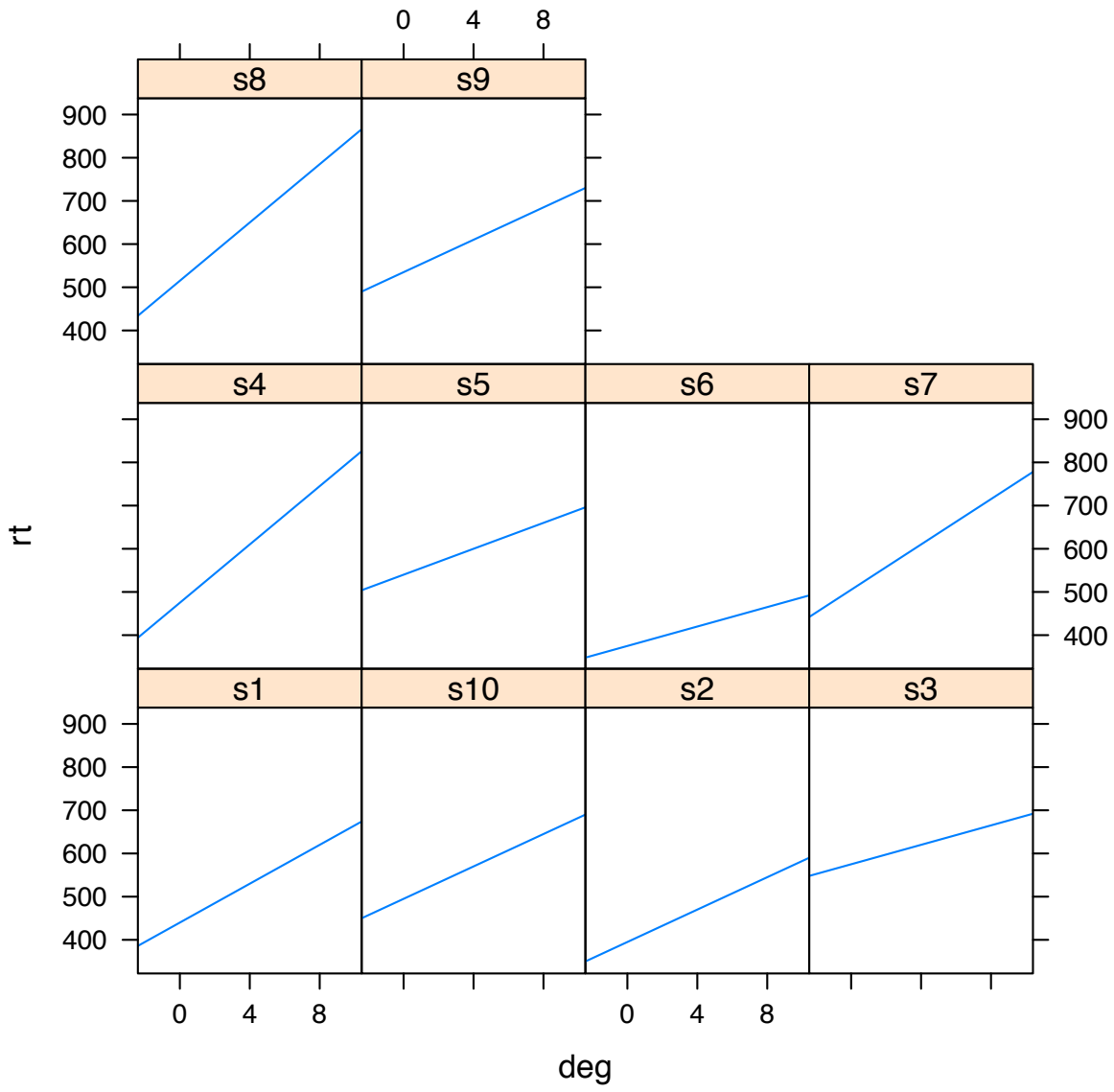


Figure 4.3: Noise effects by subject.

Given these differences between subjects, you could fit a separate linear model for each subject, collect together the intercepts and slopes for each subject, and then check if the intercepts and slopes are significantly different from zero.

Try this for one subject (s1):

```
> ## fit a separate linear model for subject s1:
> s1data<-subset(MD497.df,subj=="s1")
> lm(rt~noise,s1data)

Call:
lm(formula = rt ~ noise, data = s1data)

Coefficients:
(Intercept)   noisenoise
        440          180
```

Go back and look at the means for s1 for noise and compare them to the coefficients above.

Now we can do this for every one of our 10 subjects. I don't print this result out because it's consume a lot of pages.

```
> ## do the same for each subject using a for-loop
> subjects<-paste("s",rep(1:10),sep="")
> for(i in subjects){
     sdata<-subset(MD497.df,subj==i)
           lm(rt~noise,sdata)
   }
```

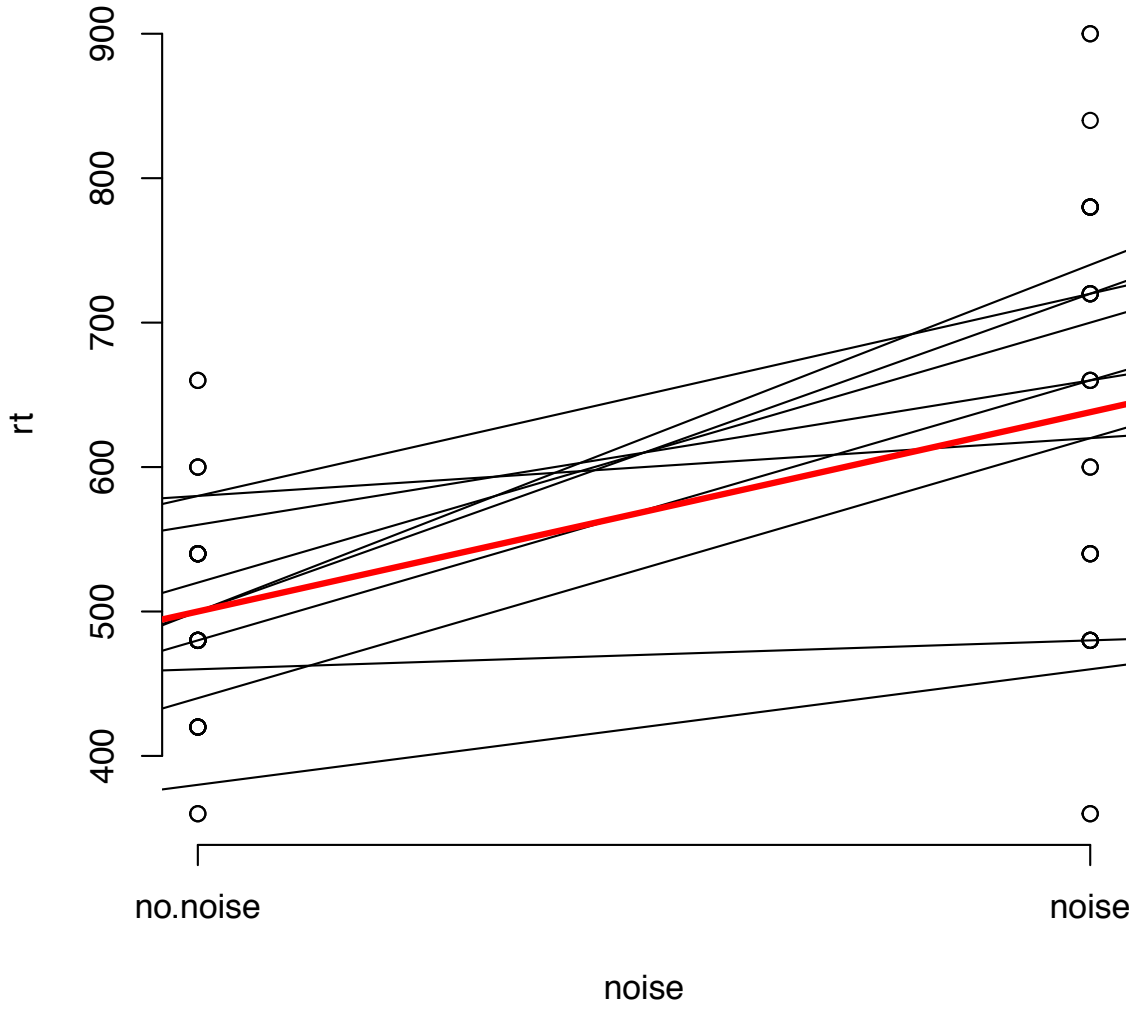
There is a function in the package `lme4` that does the above for you: `lmList`.

```
> ## do the same as the above for-loop for each subject in a single shot:
> library(lme4)
> lmlist.fm1<-lmList(rt~noise|subj,MD497.df)
> print(lmlist.fm1$s1)

Call:
lm(formula = formula, data = data)

Coefficients:
(Intercept)   noisenoise
        440          180
```

One can plot the individual lines for each subject, as well as the linear model m0's line (this shows how each subject deviates in intercept and slope from the model m0's intercept and slopes).

```
> plot(as.numeric(MD497.df$noise)-1,
        MD497.df$rt,axes=F,
        xlab="noise",ylab="rt")
> axis(1,at=c(0,1),
```

```
        labels=c("no.noise","noise"))
> axis(2)
> subjects<-paste("s",1:10,sep="")
> for(i in subjects){
   abline(lmlist.fm1[[i]])
   }
> abline(lm(rt~noise,MD497.df),lwd=3,col="red")
```



To find out if there is an effect of noise, you can simply check whether the slopes of the individual subjects' fitted lines taken together are significantly different from zero:

```
> ## now you can test with a t.test whether each coefficient is significantly different 1
> t.test(coef(lmlist.fm1)[2])
```

```
        One Sample t-test

data:  coef(lmlist.fm1)[2]
t = 5.8108, df = 9, p-value = 0.000256
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
  84.277 191.723
sample estimates:
mean of x
      138
```

The above is called repeated measures regression (see **??** for details). We now transition to the next stage of multiple regression: the linear mixed model.

4.2 Linear mixed model

The **linear mixed model** does something related to the above by-subject fits, but with some crucial twists, as we see below. In the model below, the the statement (1|subj) means that the variance associated with subject intercepts should be estimated, and from that variance the intercepts for each subject should be predicted.

```
> ## the following command fits a linear model, but in addition estimates between-subject
>
> summary(m0.lmer<-lmer(rt~noise+(1|subj),MD497.df))

Linear mixed model fit by REML ['lmerMod']
Formula: rt ~ noise + (1 | subj)
   Data: MD497.df

REML criterion at convergence: 722.4

Random effects:
 Groups   Name        Variance Std.Dev.
 subj     (Intercept)  3518     59.3
 Residual             11350    106.5
Number of obs: 60, groups: subj, 10

Fixed effects:
            Estimate Std. Error t value
(Intercept)    500.0       27.0   18.50
noisenoise     138.0       27.5    5.02

Correlation of Fixed Effects:
           (Intr)
noisenoise -0.509
```

One thing to notice is that the coefficients of the fixed effects of the above model are identical to those in the linear model m0 above. The predicted varying intercepts for each subject can be viewed by typing:

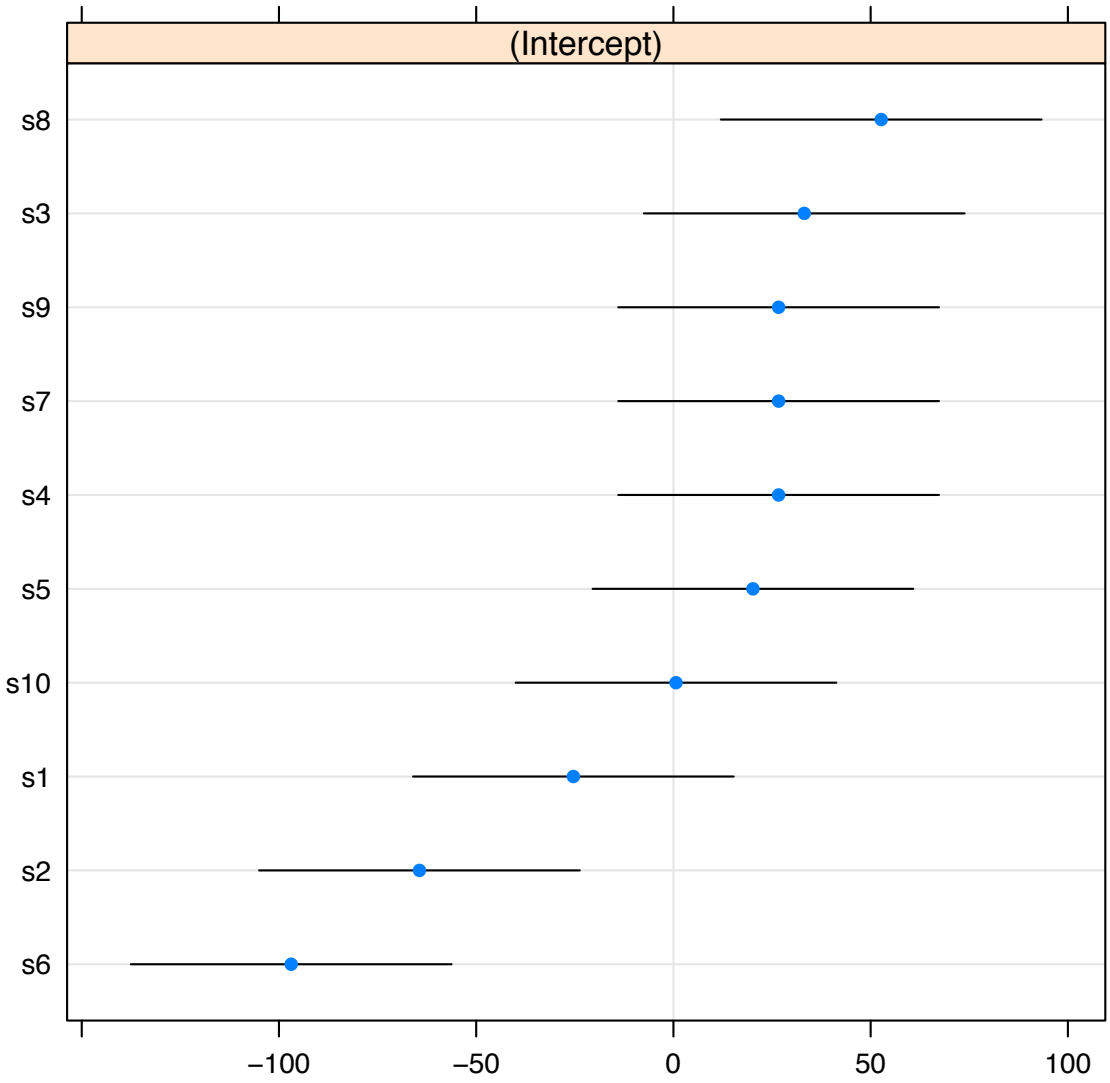
```
> ranef(m0.lmer)
```

```
$subj
    (Intercept)
s1    -25.36335
s10     0.65034
s2    -64.38390
s3     33.16746
s4     26.66404
s5     20.16061
s6    -96.90102
s7     26.66404
s8     52.67774
s9     26.66404

attr(,"class")
[1] "ranef.mer"
```

Or you can display them graphically.

```
> print(dotplot(ranef(m0.lmer,postVar=TRUE)))
```

```
$subj
```



The model m0.lmer above prints out the following type of linear model:

$$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_i + b_i + \varepsilon_i \tag{4.10}$$

It's just like our linear model except that there are different *predicted* (cf. the lmlist function above, where they are *estimated* for each subject) intercepts b_i for each subject. These are assumed by lmer to come from a normal distribution centered around 0; see [2] for more. The ordinary linear model m0 has one intercept β_0 for all subjects, whereas the linear mixed model with varying intercepts m0.lmer has a different intercept $(\beta_0 + b_i)$ for each subject.

We can visualize these different intercepts for each subject as shown below.

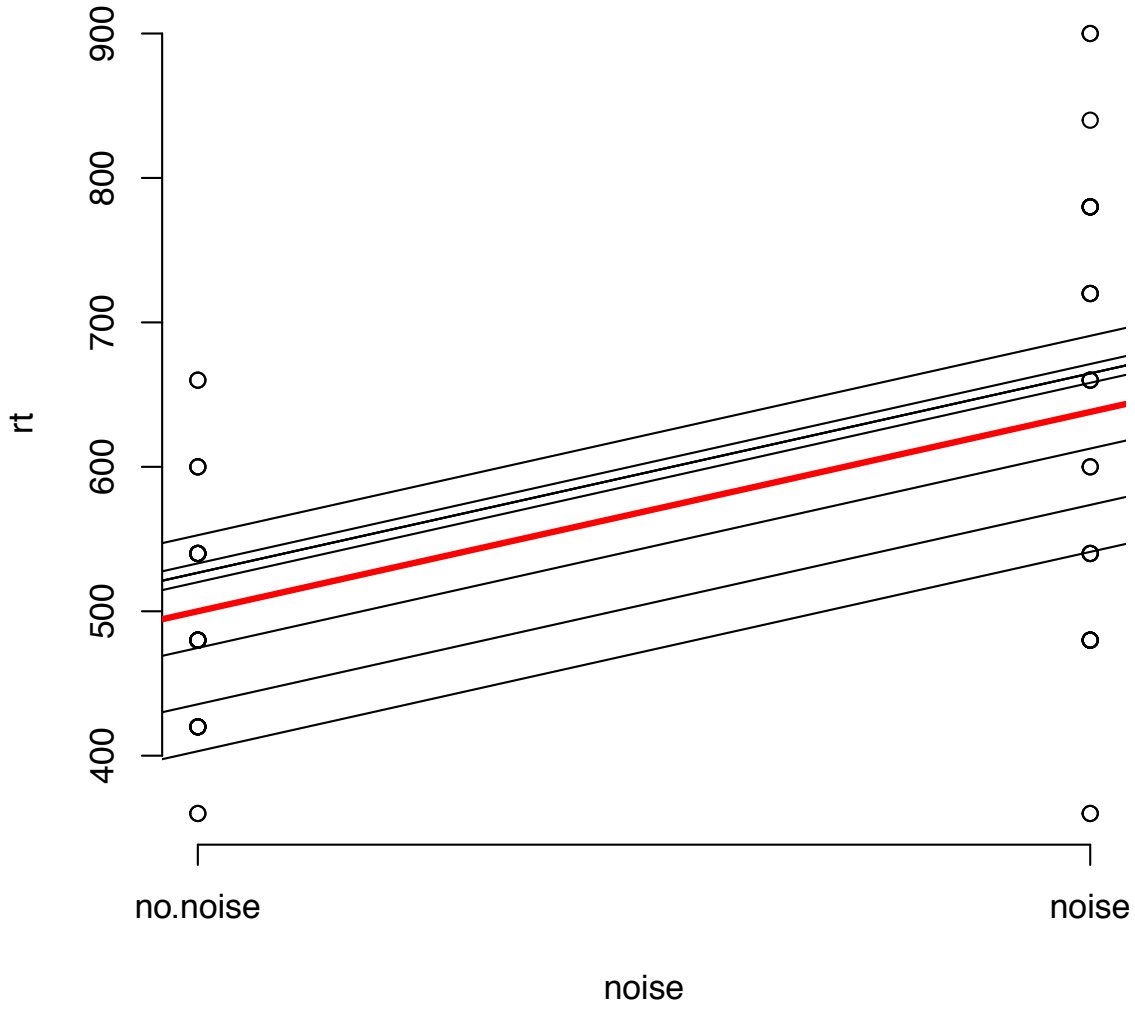
```
> (a<-fixef(m0.lmer)[1])
```

```
(Intercept)
        500
```

```
> (newa<-a+ranef(m0.lmer)$subj)
```

```
    (Intercept)
s1       474.64
s10      500.65
s2       435.62
s3       533.17
s4       526.66
s5       520.16
s6       403.10
s7       526.66
s8       552.68
s9       526.66
```

```
> ab<-data.frame(newa=newa,b=fixef(m0.lmer)[2])
> plot(as.numeric(MD497.df$noise)-1,MD497.df$rt,xlab="noise",ylab="rt",axes=F)
> axis(1,at=c(0,1),labels=c("no.noise","noise"))
> axis(2)
> for(i in 1:10){
   abline(a=ab[i,1],b=ab[i,2])
   }
> abline(lm(rt~noise,MD497.df),lwd=3,col="red")
```



Note that, unlike the figure associated with the lmlist.fm1 model above, which also involves fitting separate models for each subject, the model m0.lmer assumes different intercepts for each subject **but the same slope**. We can have lmer fit different intercepts AND slopes for each subject:

```
> summary(m1.lmer<-lmer(rt~noise+(1+noise|subj),MD497.df))

Linear mixed model fit by REML ['lmerMod']
Formula: rt ~ noise + (1 + noise | subj)
   Data: MD497.df

REML criterion at convergence: 721.03
```

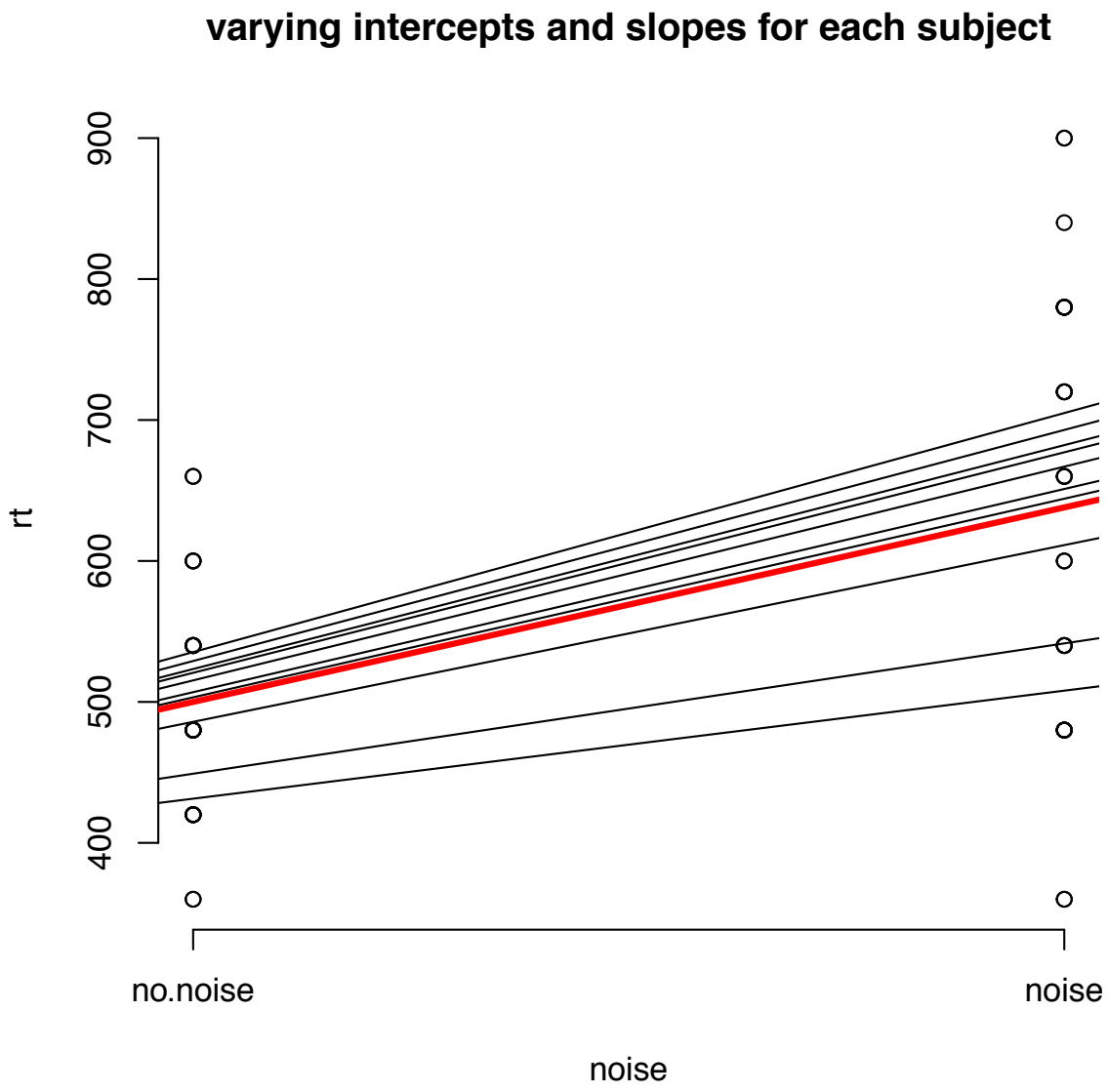
```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 subj     (Intercept)  1752     41.9
          noisenoise   1399     37.4    1.00
 Residual             10885    104.3
Number of obs: 60, groups: subj, 10

Fixed effects:
            Estimate Std. Error t value
(Intercept)    500.0       23.2   21.56
noisenoise     138.0       29.4    4.69

Correlation of Fixed Effects:
           (Intr)
noisenoise -0.302
```

These fits for each subject are visualized below (the red line shows the model with a single intercept and slope, i.e., our old model m0):

```
> (a<-fixef(m1.lmer)[1])

(Intercept)
        500

> (b<-fixef(m1.lmer)[2])

noisenoise
       138

> (newa<-a+ranef(m1.lmer)$subj[1])

    (Intercept)
s1       485.87
s10      503.25
s2       449.04
s3       529.02
s4       523.33
s5       506.90
s6       431.34
s7       520.64
s8       535.34
s9       515.27

> (newb<-b+ranef(m1.lmer)$subj[2])

    noisenoise
s1     125.369
s10    140.908
```

```
s2      92.464
s3     163.930
s4     158.846
s5     144.164
s6      76.640
s7     156.446
s8     169.585
s9     151.647
```

```
> ab<-data.frame(newa=newa,b=newb)
> plot(as.numeric(MD497.df$noise)-1,MD497.df$rt,xlab="noise",ylab="rt",axes=F,
   main="varying intercepts and slopes for each subject")
> axis(1,at=c(0,1),labels=c("no.noise","noise"))
> axis(2)
> for(i in 1:10){
   abline(a=ab[i,1],b=ab[i,2])
   }
> abline(lm(rt~noise,MD497.df),lwd=3,col="red")
```

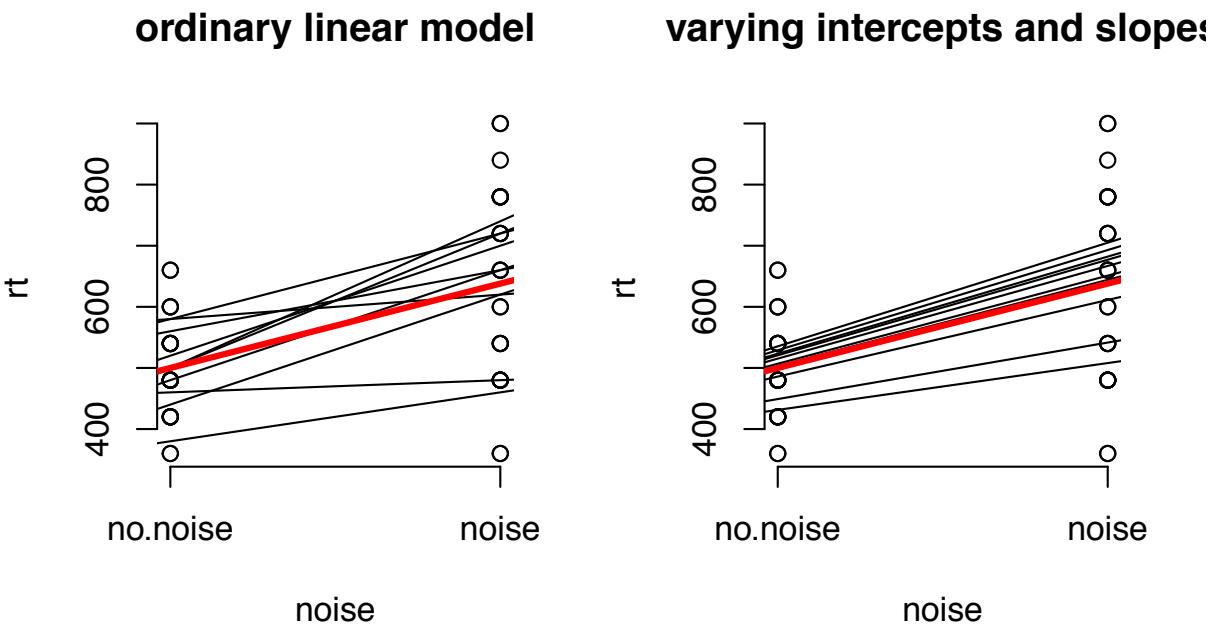


Compare this model with the lmlist.fm1 model we fitted earlier:

```
(Intercept)
        500

noisenoise
       138

    (Intercept)
s1       485.87
s10      503.25
s2       449.04
s3       529.02
```

```
s4       523.33
s5       506.90
s6       431.34
s7       520.64
s8       535.34
s9       515.27
```

```
    noisenoise
s1     125.369
s10    140.908
s2      92.464
s3     163.930
s4     158.846
s5     144.164
s6      76.640
s7     156.446
s8     169.585
s9     151.647
```



The above graphic shows some crucial difference between the lmlist (repeated measures) model and the lmer model. Note that the fitted line for each subject in the lmer model is much closer to the m0 model's fitted (red) line. This is because lmlist uses each subject's data separately (resulting in possibly wildly different models, depending on the variability between subjects), whereas lmer "borrows strength from the mean" and pushes (or "shrinks") the estimated intercepts and slopes of each subject closer to the mean intercepts and slopes (the model m0's intercepts and slopes). Because it shrinks the coefficients towards the means, this is called shrinkage. This is particularly useful when several data points are missing in a particular condition for a particular subject: in an ordinary linear model, estimating coefficients using lmList would lead to very poor estimates for that subject; by contrast, lmer assumes that the estimates for such a subject are not reliable and therefore shrinks that subject's estimate to the mean values.

To see an example of shrinkage, consider the case where we remove three of the data points from subject s8, resulting in exaggeratedly high means for that subject.

First, we read in a data frame which is just the same as MD497.df, except that subject 8 (s8) has only three data points, not six (I took out three of s8's low measures). This skews the subject's estimates for intercept and slope in the lmlist model fit.

```
> MD497.df2<-read.table("MD497df.txt",header=T)
```

Next, let's confirm that the new data frame has extreme means for s8:

```
> with(MD497.df,tapply(rt,list(subj,noise),mean,na.rm=TRUE))

    no.noise noise
s1       440   620
s10      480   660
s2       460   480
s3       500   740
s4       500   720
s5       580   620
s6       380   460
s7       520   700
s8       580   720
s9       560   660

> with(MD497.df2,tapply(rt,list(subj,noise),mean,na.rm=TRUE))

    no.noise noise
s1       440   620
s10      480   660
s2       460   480
s3       500   740
s4       500   720
s5       580   620
s6       380   460
s7       520   700
s8       660   810
s9       560   660
```

We now fit the lmlist model and the linear mixed model.

```
> lmlist.fm2<-lmList(rt~noise|subj,MD497.df2)
> summary(m2.lmer<-lmer(rt~noise+(1+noise|subj),MD497.df2))

Linear mixed model fit by REML ['lmerMod']
Formula: rt ~ noise + (1 + noise | subj)
   Data: MD497.df2

REML criterion at convergence: 683.17
```

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 subj     (Intercept)  2292     47.9
          noisenoise   1956     44.2    1.00
 Residual             10247    101.2
Number of obs: 57, groups: subj, 10

Fixed effects:
            Estimate Std. Error t value
(Intercept)    501.5       24.5   20.50
noisenoise     143.9       30.2    4.76

Correlation of Fixed Effects:
           (Intr)
noisenoise -0.209
```

Now if we plot the model for s8, we find that the lmlist model indeed estimates pretty extreme intercepts for s8. But the linear mixed model predicts an intercept that's much closer to the mean (the red line). Let's just plot s8's fitted line in both models relative to the linear model fitted line.

```
> multiplot <- function(row,col){
        par(mfrow=c(row,col),pty="s")
      }
> multiplot(2,2)
> ## reduced data:
> plot(as.numeric(MD497.df2$noise)-1,MD497.df2$rt,axes=F,xlab="noise",ylab="rt",main="orc
> axis(1,at=c(0,1),labels=c("no.noise","noise"))
> axis(2)
> abline(lmlist.fm2$s8)
> abline(lm(rt~noise,MD497.df2),lwd=3,col="red")
> (a<-fixef(m2.lmer)[1])

(Intercept)
     501.53

> (b<-fixef(m2.lmer)[2])

noisenoise
    143.92

> (newa<-a+ranef(m2.lmer)$subj[1])

    (Intercept)
s1       483.69
s10      502.57
s2       443.42
s3       530.66
```

```
s4       524.45
s5       506.30
s6       424.28
s7       521.46
s8       563.02
s9       515.50
```

```
> (newb<-b+ranef(m2.lmer)$subj[2])
```

```
    noisenoise
s1     127.435
s10    144.884
s2      90.236
s3     170.830
s4     165.090
s5     148.323
s6      72.559
s7     162.333
s8     200.723
s9     156.820
```

```
> ab<-data.frame(newa=newa,b=newb)
> plot(as.numeric(MD497.df2$noise)-1,MD497.df2$rt,axes=F,
   main="varying intercepts and slopes",
   sub="s8, missing data",
   xlab="noise",ylab="rt")
> axis(1,at=c(0,1),labels=c("no.noise","noise"))
> axis(2)
> abline(a=ab[9,1],b=ab[9,2])
> abline(lm(rt~noise,MD497.df2),lwd=3,col="red")
> ## unreduced
>
> plot(as.numeric(MD497.df$noise)-1,MD497.df$rt,axes=F,xlab="noise",ylab="rt",main="ordir
   ,sub="s8, no missing data")
> axis(1,at=c(0,1),labels=c("no.noise","noise"))
> axis(2)
> abline(lmlist.fm1$s8)
> abline(lm(rt~noise,MD497.df),lwd=3,col="red")
> (a<-fixef(m2.lmer)[1])
```

```
(Intercept)
     501.53
```

```
> (b<-fixef(m2.lmer)[2])
```

```
noisenoise
    143.92
```

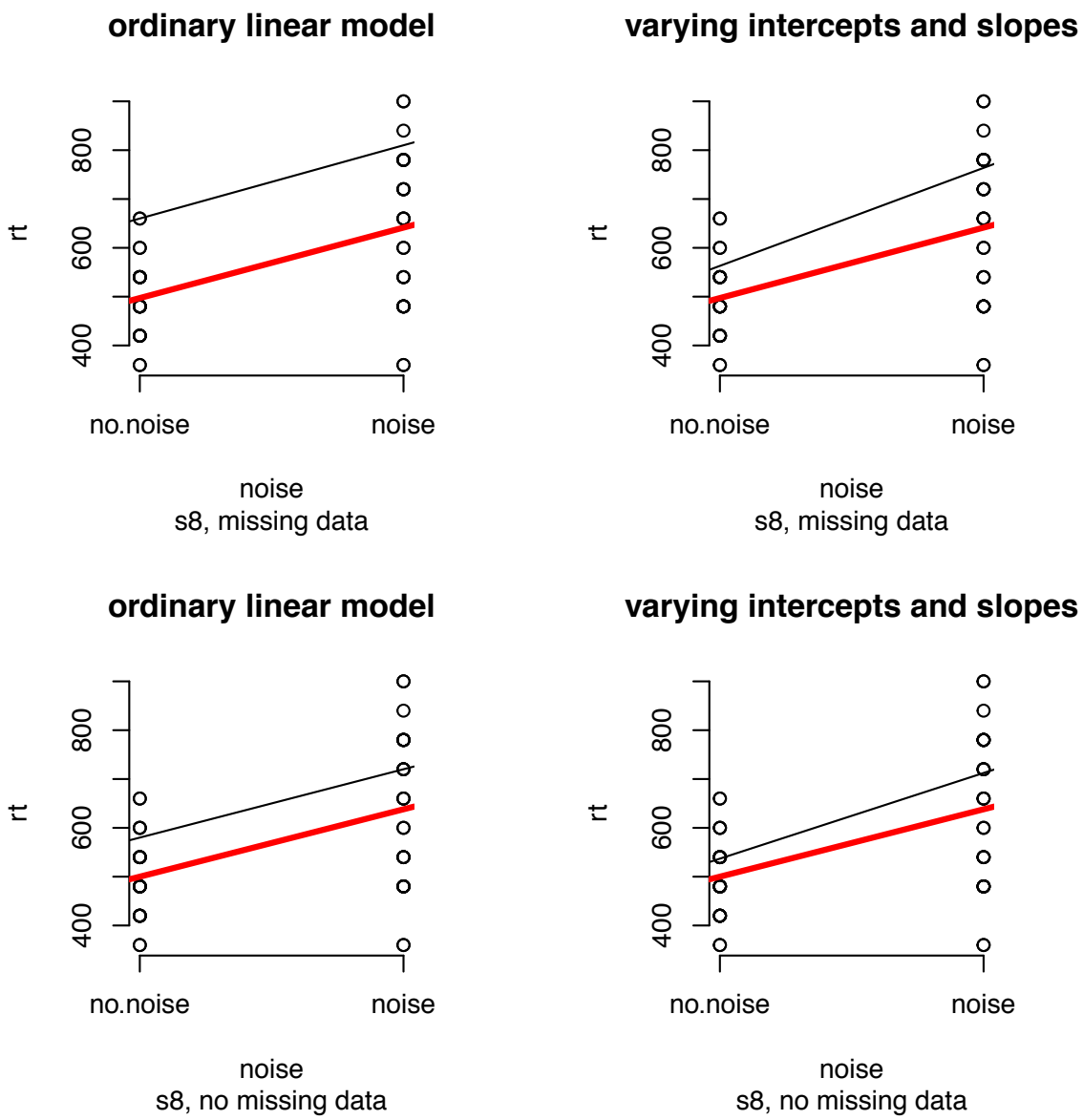
```
> (newa<-a+ranef(m1.lmer)$subj[1])
```

```
    (Intercept)
s1       487.40
s10      504.79
s2       450.58
s3       530.55
s4       524.86
s5       508.43
s6       432.87
s7       522.18
s8       536.88
s9       516.81
```

```
> (newb<-b+ranef(m1.lmer)$subj[2])
```

```
    noisenoise
s1     131.293
s10    146.831
s2      98.388
s3     169.853
s4     164.769
s5     150.087
s6      82.564
s7     162.370
s8     175.509
s9     157.571
```

```
> ab<-data.frame(newa=newa,b=newb)
> plot(as.numeric(MD497.df$noise)-1,MD497.df$rt,axes=F,
   main="varying intercepts and slopes",sub="s8, no missing data",xlab="noise",ylab="rt")
> axis(1,at=c(0,1),labels=c("no.noise","noise"))
> axis(2)
> abline(a=ab[9,1],b=ab[9,2])
> abline(lm(rt~noise,MD497.df),lwd=3,col="red")
```



One crucial difference between the lmlist model and the lmer model is that the former estimates the parameters for each subject separately, whereas the latter estimates the variance associated with subjects' intercepts (and slopes, if you specify in the model that one should do that) and then *predicts* each subjects intercepts and slopes based on that variance.

4.3 Contrast coding

Instead of working with the degree and noise data, we will work with the lexical decision data from the languageR package by Harald Baayen. His book [1] is an excellent one for psycholinguists.

4.3.1 Treatment contrasts

Consider the simplest case where we need to compare reaction times in an experiment involving two conditions. As mentioned above, we take the lexical decision dataset `lexdec` from the library `languageR` as an example.

```
> library(languageR)
> ## isolate relevant columns
> head(lexdec[,c(1,2,5)])
```

```
  Subject     RT NativeLanguage
1      A1 6.3404        English
2      A1 6.3081        English
3      A1 6.3491        English
4      A1 6.1862        English
5      A1 6.0259        English
6      A1 6.1800        English
```

This dataset shows log lexical decision times of participants to different words.

Suppose we want to know whether being a native speaker of English affects reaction time. Before even doing the experiment, it is clear that we would expect that native speakers to have shorter reaction times. We can verify that the means do have the expected difference; the question is whether this difference is statistically significant:

```
> means.lexdec<-with(lexdec,tapply(RT,NativeLanguage,mean))
```

```
English   Other
 6.3183  6.4741
```

The mean for English is 6.318, and the means for the other language is 6.474; the difference between the two is 0.156. These values become relevant in a moment (recall the discussion of the noise and deg data above, though; it should be clear to you why the means are relevant).

It is straightforward to carry out the comparison between these means using a linear model.

```
> summary(lm(RT~NativeLanguage,lexdec))
```

```
Call:
lm(formula = RT ~ NativeLanguage, data = lexdec)

Residuals:
    Min      1Q  Median      3Q     Max
-0.5688 -0.1529 -0.0323  0.1148  1.1132

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          6.31831    0.00744   849.8   <2e-16
NativeLanguageOther  0.15582    0.01136    13.7   <2e-16

Residual standard error: 0.229 on 1657 degrees of freedom
Multiple R-squared:  0.102,        Adjusted R-squared:  0.101
F-statistic:  188 on 1 and 1657 DF,  p-value: <2e-16
```

What is the interpretation of the coefficients? Comparing the means for each condition with the coefficients reveals that (i) the intercept's value is the mean for English; and (ii) the slope's value is the difference between the two conditions' mean.

But how does R deliver these particular values for the intercept and slope? This comes from the contrast coding specified for the predictor variable. By default, R assigns the so-called TREATMENT CONTRAST CODING to the predictors: the alphabetically earlier predictor level (here, English) is coded as 0 (the baseline), and the other level (here, Other) is coded as 1.

The interpretation for the intercept and slope derives from this numerical coding: when the predictor is 0 (i.e., when the participant is a native speaker of English), the predicted reaction time is the estimated intercept. When the predictor is coded as 1 (i.e., the participant is a non-native English speaker), then the predicted reaction time is the sum of the intercept and the slope. It is possible to examine the contrast coding using the `contrasts` command:

```
> contrasts(lexdec$NativeLanguage)

        Other
English     0
Other       1
```

As mentioned above, R alphabetically orders the factors and takes the first condition as the baseline. It is of course possible to take the other level as the baseline:

```
> lexdec$NativeLanguage<-factor(lexdec$NativeLanguage,levels=c("Other","English"))
> contrasts(lexdec$NativeLanguage)

        English
Other         0
English       1
```

Now, the intercept and slope will have a different interpretation:

```
> summary(lm(RT~NativeLanguage,lexdec))

Call:
lm(formula = RT ~ NativeLanguage, data = lexdec)

Residuals:
    Min      1Q  Median      3Q     Max
-0.5688 -0.1529 -0.0323  0.1148  1.1132

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           6.47413    0.00859   754.1   <2e-16
NativeLanguageEnglish -0.15582    0.01136   -13.7   <2e-16

Residual standard error: 0.229 on 1657 degrees of freedom
Multiple R-squared:  0.102,        Adjusted R-squared:  0.101
F-statistic:  188 on 1 and 1657 DF,  p-value: <2e-16
```

The intercept now represents the mean score of the level Other, and the slope the difference between the English and Other scores. The sign of the slope is negative because now the difference is computed by subtracting the mean English score from the mean Other score.

4.3.2 Sum contrasts

Treatment contrasts are only one option, however. It is possible to utilize the so-called SUM CONTRAST, which codes one of the two conditions as -1 and the other as 1, effectively 'centering' the predictor.

```
> c.sum<-contr.sum(2)

  [,1]
1    1
2   -1
```

In our example, we can assign the sum contrast so that Other is 1 and English is -1 (note that reordering the factors would give the opposite coding):

```
> contrasts(lexdec$NativeLanguage) <- c.sum

  [,1]
1    1
2   -1
```

The linear model's estimated coefficients now look different again:

```
> summary(lm(RT~NativeLanguage,lexdec))

Call:
lm(formula = RT ~ NativeLanguage, data = lexdec)

Residuals:
    Min      1Q  Median      3Q     Max
-0.5688 -0.1529 -0.0323  0.1148  1.1132

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      6.39622    0.00568  1126.3   <2e-16
NativeLanguage1  0.07791    0.00568    13.7   <2e-16

Residual standard error: 0.229 on 1657 degrees of freedom
Multiple R-squared:  0.102,        Adjusted R-squared:  0.101
F-statistic:  188 on 1 and 1657 DF,  p-value: <2e-16
```

The intercept is now the grand mean of the two conditions:

```
> mean(means.lexdec)

[1] 6.3962
```

When the predictor is coded as 1 (i.e., when the participant belongs to the group Other), the predicted RT is 6.39622+0.07791, and when the predictor is coded as -1 (i.e., when the participang belongs to the English group), the predicted RT is 6.39622-0.07791.

To summarize, treatment contrasts and sum contrasts are two possible ways to compare the two conditions, and they answer different research questions. Treatment contrasts compare one or more condition's mean against a baseline condition (we show an example below where more than two conditions are involved), whereas sum contrasts allow us to determine whether a condition's mean is significantly different from the grand mean.

Let us now look at some other contrast coding schemes.

4.3.3 Sliding contrasts

As an illustration, we take the same `lexdec` dataset and investigate the question: does word frequency affect reaction time? Here, we would expect that lower frequency would result in longer reaction time. In the `lexdec` dataset, frequency is provided as a continuous variable (each word has a frequency value associated with it). We could fit a linear model where we use continuous frequency values as a predictor of reaction times. Since our immediate focus is on qualitative predictors, we first convert this continuous predictor to a qualitative one: low, medium and high frequency:

```
> library(gtools)
> Freq <-cut(lexdec$Frequency,breaks=3,labels=c("low","med","high"))
> lexdec$Freq <- factor(Freq)
```

Let us first calculate the mean scores for each level of `Freq`:

```
> means.freq <- with(lexdec,tapply(RT,Freq,mean))
```

```
   low    med   high
6.4564 6.3867 6.3099
```

The default coding for such a three-condition case is the treatment contrast:

```
> contrasts(lexdec$Freq)
```

```
     med high
low    0    0
med    1    0
high   0    1
```

Suppose we want to know whether frequency level `low` leads to significantly longer reaction times than frequency level `medium`, and whether frequency level `medium` leads to significantly longer reaction times than frequency level `high`. R has a contrast coding for answering this question: SLIDING CONTRASTS or REPEATED CONTRASTS:

```
> library(MASS)
> c.sliding <- contr.sdif(3)
```

The two pairs of means being compared are:

```
> means.freq[2]-means.freq[1]

      med
-0.069744

> means.freq[3]-means.freq[2]

     high
-0.076788
```

The linear (mixed) model with sliding contrasts yields these means as coefficients:

```
> contrasts(lexdec$Freq) <- c.sliding
> summary(lm(RT~Freq,lexdec))

Call:
lm(formula = RT ~ Freq, data = lexdec)

Residuals:
    Min      1Q  Median      3Q     Max
-0.5456 -0.1607 -0.0341  0.1146  1.1309

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.38435    0.00631 1012.34  < 2e-16
Freq2-1     -0.06974    0.01449   -4.81  1.6e-06
Freq3-2     -0.07679    0.01449   -5.30  1.3e-07

Residual standard error: 0.237 on 1656 degrees of freedom
Multiple R-squared:  0.042,        Adjusted R-squared:  0.0409
F-statistic: 36.3 on 2 and 1656 DF,  p-value: 3.67e-16
```

This contrast coding answers the research question directly: each of the two differences is significantly different from 0.

Suppose now that our research question had been: is the mean of the last condition (`high`), significantly different from the average of the other two; and are the other two significantly different from each other? This question can be answered using Helmert contrasts:

```
> c.helmert <- contr.helmert(3)
> contrasts(lexdec$Freq) <- c.helmert
> contrasts(lexdec$Freq)

     [,1] [,2]
low    -1   -1
med     1   -1
high    0    2
```

Now we expect to see the following means being compared:

```
> c(means.freq[2]-means.freq[1], means.freq[3]-(means.freq[2]+means.freq[1])/2)

      med      high
-0.069744 -0.111660

> means.freq[3]-(means.freq[1]+means.freq[2])/2

    high
-0.11166

> means.freq[2]-means.freq[1]

      med
-0.069744
```

The linear model directly compares these means:

```
> summary(lm(RT~Freq,lexdec))

Call:
lm(formula = RT ~ Freq, data = lexdec)

Residuals:
    Min      1Q  Median      3Q     Max
-0.5456 -0.1607 -0.0341  0.1146  1.1309

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.38435    0.00631 1012.34  < 2e-16
Freq1       -0.03487    0.00724   -4.81  1.6e-06
Freq2       -0.03722    0.00472   -7.89  5.6e-15

Residual standard error: 0.237 on 1656 degrees of freedom
Multiple R-squared:  0.042,        Adjusted R-squared:  0.0409
F-statistic: 36.3 on 2 and 1656 DF,  p-value: 3.67e-16
```

However, note that the coefficients do not match the differences between means that we just explored. In order to get the comparisons of interest, we must take the generalized inverse of a normalized contrast specification:

```
> c.helmert2 <- matrix(c(  -1,    1,    0,
                       -1/2, -1/2,    1
                            ),  3,  2,
                          dimnames=list(c("low", "med", "high"),
                                        c(".low-med", ".med-high")))
> contrasts(lexdec$Freq) <- t(ginv(c.helmert2))
> summary(lm(RT~Freq,lexdec))
```

```
Call:
lm(formula = RT ~ Freq, data = lexdec)

Residuals:
    Min      1Q  Median      3Q     Max
-0.5456 -0.1607 -0.0341  0.1146  1.1309

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.38435    0.00631 1012.34  < 2e-16
Freq1       -0.06974    0.01449   -4.81  1.6e-06
Freq2       -0.11166    0.01416   -7.89  5.6e-15

Residual standard error: 0.237 on 1656 degrees of freedom
Multiple R-squared:  0.042,        Adjusted R-squared:  0.0409
F-statistic: 36.3 on 2 and 1656 DF,  p-value: 3.67e-16
```

Now the coefficients match the mean differences:

```
> means.freq[3]-(means.freq[1]+means.freq[2])/2

    high
-0.11166

> means.freq[2]-means.freq[1]

      med
-0.069744
```

The details regarding why we must take the inverse are not important right now, but [8] has more detail.

4.3.4 ANOVA contrast coding

One can also do a classical anova contrast coding (main effects and interaction). Consider a 2×2 design like this data (this is real EEG data from my lab):

```
> data <- read.table("mean_600_750.tab",header=T)
> head(xtabs(~subj+cond,data))

      cond
subj   101 102 103 104
  co01  23  23  23  23
  co02  23  23  23  23
  co03  23  23  23  23
  co04  23  23  23  23
  co05  23  23  23  23
  co06  23  23  23  23
```

```
> ## conditions:
> ## 101: P S gram
> ## 102: S P ungram with intruder
> ## 103: S S ungram w/o intruder
> ## 104: P P gram with intruder
>
> #        1  2  3  4
> #gram    1 -1 -1  1
> #intr.g -1  0  0  1
> #intr.u  0 -1  1  0
>
> head(data)

  subj cond chan        win   value    gmin tgmin     gmax tgmax
1 co01  101   F7 +600..+748 -2.8615 -6.3324 0.652 -0.10371 0.620
2 co01  101   F3 +600..+748 -4.8754 -7.5494 0.652 -3.16040 0.676
3 co01  101   FZ +600..+748 -4.2410 -6.4748 0.652 -2.41380 0.608
4 co01  101   F4 +600..+748 -1.9462 -5.0652 0.652  0.66873 0.700
5 co01  101   F8 +600..+748 -2.2612 -6.5130 0.620  1.44220 0.740
6 co01  101  FC5 +600..+748 -4.9317 -8.3295 0.652 -1.24640 0.624

> data$cond<-factor(data$cond,levels=c(101,102,104,103))
> ## critical channels
> critc <- c("F3","FZ","F4","C3","CZ","C4","P3","PZ","P4")
> ## frontals:
> frontc <- c("F3","FZ","F4")
> ## central:
> centralc <- c("C3","CZ","C4")
> # posterior:
> postc <- c("P3","PZ","P4")
> d <- subset(data,chan%in%critc)
> library(lme4)
> contrasts(d$cond)

    102 104 103
101   0   0   0
102   1   0   0
104   0   1   0
103   0   0   1

> anova.contrast <- matrix(c(  -1/2, -1/2, +1/2, +1/2,              # Main effect A
                             -1/2, +1/2, -1/2, +1/2,              # Main effect B
                             +1/2, -1/2, -1/2, +1/2),  4,  3,     # Interaction A x B
                              dimnames=list(c("101", "102", "104", "103"),
                                            c(".A", ".B", ".AxB")))
> contrasts(d$cond)<-anova.contrast
> contrasts(d$cond)
```

```
      .A   .B .AxB
101 -0.5 -0.5  0.5
102 -0.5  0.5 -0.5
104  0.5 -0.5 -0.5
103  0.5  0.5  0.5
```

```
> (fm1 <- lmer(value~cond+(1|subj), d ) )

Linear mixed model fit by REML ['lmerMod']
Formula: value ~ cond + (1 | subj)
   Data: d
REML criterion at convergence: 2792.2
Random effects:
 Groups   Name        Std.Dev.
 subj     (Intercept) 2.54
 Residual             2.60
Number of obs: 576, groups: subj, 16
Fixed Effects:
(Intercept)       cond.A       cond.B     cond.AxB
     -1.530        0.808        0.494        0.214
```

4.3.5 Steps in fitting a linear (mixed) model

Here is a checklist for fitting linear models:

1. First, check that your data have been correctly extracted. This step is often skipped, and it often leads to mistakes. Did all subjects deliver the expected number of data points? Do you have as many rows in your data frame as you'd expect? Are all items present in each subject's data? Are there any strange values for dependent measures? In other words, carefully check your assumptions about the data before you do anything else.

2. Next, define your contrast coding based on your predictions.

3. Having fit your model, check your assumptions, such as whether the residuals are approximately normally distributed. Although books like [2] say that the normality of residuals assumption in linear models is the "least important" of the assumptions in a linear model, it does not follow (and Gelman would agree) that you can simply ignore the normality of residuals assumption. This is especially important when, as is common in psycholinguistics, we want to do a hypothesis test. I explain this point next. The text below is taken almost verbatim from a comment I made on Andrew Gelman's blog.

   > Suppose we are interested in null hypothesis tests in linear models, e.g., $H_0 : \beta_1 = 0$, where β_0 is one of the parameters in the model. Suppose also that we have a "lot" of data. To make things concrete, assume that we have a 2ÃŰ2 within subjects design, with 100 subjects; each subject sees one of the four conditions in the 2ÃŰ2 design 24 times (the standard counterbalancing done in psychology). So, each subject will see each condition 24 times. Assume that the dependent measure is something like reading times. Linear mixed models are a standard way to analyze such data.

Here is the argument (it's a bit technical but I will elaborate on it in class) that suggests that checking the normality assumption of residuals is necessary. Note that $\hat{\beta} \sim N_p(\beta, \sigma^2(X^TX)^{-1})$, and that $\frac{\hat{\sigma}^2}{\sigma^2} \sim \frac{\chi^2_{n-p}}{n-p}$.
From distributional theory we know that $T = \frac{X}{\sqrt{Y/v}}$, when $X \sim N(0,1)$ and $Y \sim \chi^2_v$.

Let x_i be a column vector containing the values of the explanatory/regressor variables for a new observation i. Then if we define:

$$X = \frac{x_i^T\hat{\beta}x_i^T\beta}{\sqrt{\sigma^2 x_i^T(X^TX)^{-1}x_i}} \sim N(0,1) \tag{4.11}$$

and

$$Y = \frac{\hat{\sigma}^2}{\sigma^2} \sim \frac{\chi^2_{n-p}}{n-p} \tag{4.12}$$

it follows that $T = \frac{X}{\sqrt{Y/v}}$:

$$T = \frac{x_i^T\hat{\beta}x_i^T\beta}{\sqrt{\hat{\sigma}^2 x_i^T(X^TX)^{-1}x_i}} = \frac{\frac{x_i^T\hat{\beta}x_i^T\beta}{\sqrt{\sigma^2 x_i^T(X^TX)^{-1}x_i}}}{\sqrt{\frac{\hat{\sigma}^2}{\sigma^2}}} \sim t_{n-p} \tag{4.13}$$

I.e., a 95% CI:

$$x_i^T\hat{\beta} \pm t_{n-p,1-\alpha/2}\sqrt{\hat{\sigma}^2 x_i^T(X^TX)^{-1}x_i} \tag{4.14}$$

So, although we can estimate $\hat{\beta}$ without any distributional assumptions, we cannot calculate confidence intervals for parameters, and we can't do hypothesis testing relating to these parameters using F tests because we don't know that $\hat{\beta}$ is multivariate normal because the distribution of y might not be multivariate normal (because the distribution of ε might not be normal).

We can investigate the consequences of non-normality of residuals with a simulation.

```
> nsim<-100
> n<-100
> pred<-rep(c(0,1),each=n/2)
> store<-matrix(0,nsim,5)
> ## should the distribution of errors be non-normal?
> non.normal<-TRUE
> ## true effect:
> beta.1<-0.5
> for(i in 1:nsim){
   ## assume non-normality of residuals?
   ## yes:
   if(non.normal==TRUE){
   errors<-rchisq(n,df=1)
```

```
errors<-errors-mean(errors)} else {
## no:
errors<-rnorm(n)
}
## generate data:
y<-100 + beta.1*pred + errors
fm<-lm(y~pred)
## store coef., SE, t-value, p-value:
store[i,1:4]<-summary(fm)$coefficients[2,c(1,2,3,4)]
}
```

We can calculate the probability of finding a significant effect given that the null hypothesis is false:

```
> ## ``observed'' power for raw scores:
> table(store[,4]<0.05)[2]
```

```
TRUE
  45
```

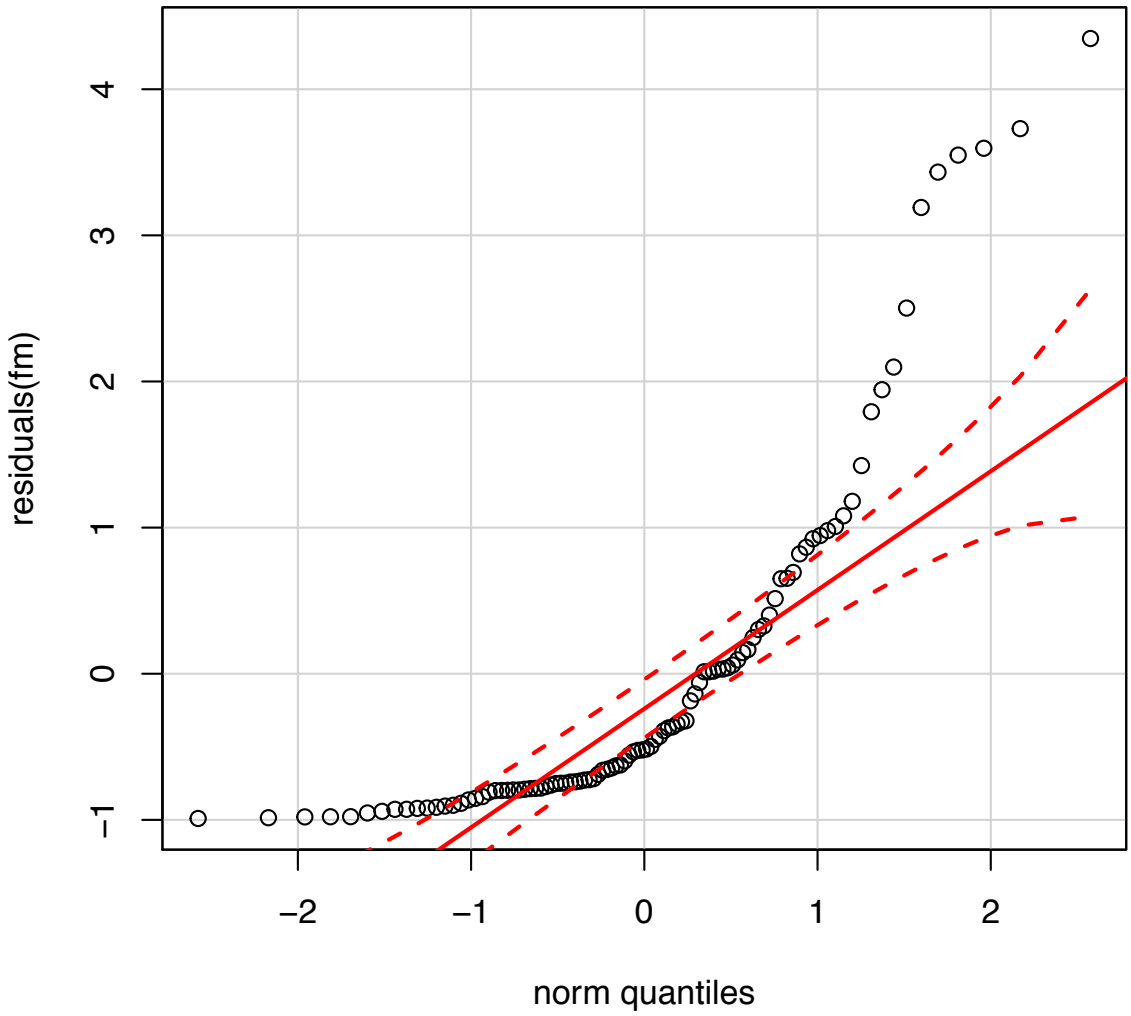
We see that there is a huge loss of power compared to the case where the residuals are normal (exercise).

Note that the coverage of the 95% CIs is unaffected, but this is not interesting for us when we are doing hypothesis testing!

```
> ## CIs:
> upper<-store[,1]+2*store[,2]
> lower<-store[,1]-2*store[,2]
> ## CIs' coverage is unaffected by skewness:
> table(lower<beta.1 & upper>beta.1)
```

```
FALSE  TRUE
    5    95
```

Here is the type of residual distribution we have in the above simulation; it is pretty typical for reading and reaction time studies.



Note also that if the residuals are non-normally distributed, your fitted model itself is no longer realistic for the data. You can establish this by doing what Gelman and Hill suggest we do for evaluating model quality: simulate new data and look at whether these simulated values fall in the right ball-park. (exercise)

4. Related to the above point, you should use the boxcox function in the MASS package in R to find out which transform you need to stabilize variance. Examples are provided in the case studies chapter.

5. After having fit the model, check whether there are influential values. Use the influence.ME package for this purpose.

6. Finally, displaying the results of a linear mixed model: usually we are not interested in the

random effects parameter estimates, only the fixed effects estimates. One can use something like the extractfit function and the myxtable function (below) that prints out the linear mixed model fit as a formatted LaTeX table (this is useful if you are working in LaTeX, which is often the case in linguistics).

```
> extractfit<-function(mod,indices=2:4,
                       coln=c("coef.","SE","t-value"),
                       fac,dig=2,model.type="LM"){
    if(model.type=="LM"){
      ##LM:
      fixefs<-coef(mod)[indices]} else {
      ##LMER:
      fixefs<-fixef(mod)[indices]
      }
    SEs<-sqrt(diag(vcov(mod)))[indices]
    torz<-fixefs/SEs
    results<-round(cbind(fixefs,SEs,torz),digits=dig)
    results<-data.frame(fac=fac,results)
    colnames(results)<-coln
    rownames(results)<-NULL
    results
  }
> myxtable<-function(res,cap,lab){
    print(xtable(res,caption=cap,label=lab),
          include.rownames=F)}
```

Here is an example:

```
> m0<-lm(rt~noise,MD497.df)
> results.m0<-extractfit(m0,
                         coln=c("","coef.",
                                "SE","t-value"),
                         indices=1:2,
                         fac=c("Intercept","noise"))
>
> #library(xtable)

> myxtable(results.m0,
           cap="The effect of noise on reaction time.",
           lab="tab:m0")
```

	coef.	SE	t-value
Intercept	500.00	22.08	22.65
noise	138.00	31.23	4.42

Table 4.4: The effect of noise on reaction time.

7. The final step is producing a good quality summary plot or plots of the results.

4.3.6 Where to look for more examples

See the case studies in the next chapter, and the website: http://openscience.uni-leipzig.de (the Mind Research Repository) for more examples.

Chapter 5

Review exercises 1

5.1 Computing a 95% confidence interval

Take a random sample of size 150 from a population with mean 100, and standard deviation 50.

Compute the mean and standard deviation of the sample. Next, compute the estimated standard error using the standard deviation you just estimated from the sample. Now you have an estimate of the mean and an estimate of the standard deviation of the sampling distribution of the sample means.

Using the approximation that 2 times the estimated standard deviation of a normal distribution covers 95% of the area under the curve, compute the lower and upper bounds around the sample mean such that 95% of the area under the curve of the estimated SDSM (the so-called 95% confidence interval) is .95.

5.2 The t-distribution

Just as we have pnorm and qnorm in the normal distribution, we also have the functions pt and qt. For example, I can ask for the probability to the left of -2 in a t-distribution with degrees of freedom 149:

```
> pt(-2,df=149)
[1] 0.023659
```

Compare this with the normal distribution with mean 0 and sd 1:

```
> pnorm(-2)
[1] 0.02275
```

Tasks:

1. In a t-distribution with degrees of freedom (df) 149, calculate the value t1 such that the probability to the right of it is .025.
2. Then, for a t-distribution with df=149, calculate the value t2 such that the probability to the left of it is 0.025.
3. What is the area between t1 and t2?
4. We will call the absolute value of t1 (or t2) the critical t value.

5.3 Confidence intervals revisited

For the data you generated in question 1, and using the calculations you did in question 2, recompute the 95% confidence interval using:

$\bar{x} \pm critical.t \times estimated.SE$

Now compare your lower and upper bounds with the output of the t.test. Assuming that the data was saved in a sample called x, you can do:

```
t.test(x)$conf.int
```

5.4 Your first data analysis

Read in the data `simdata1.txt` provided with these notes.

Then work out the 95% confidence intervals for condition a, and condition b (each one separately of course). Use the critical t-value for this calculation.

Without doing any more statistical analysis, can you say whether the values for condition a and b come from the same distribution?

Chapter 6

Review exercises 2

6.1 Your first linear model

Read in the data `simdata1.txt` from review exercises 1.

Fit a linear model to `simdata1.txt` to investigate whether condition a and b come from populations with different means.

Do the same with `simdata2.txt`.

6.2 Beauty data

Look at the beauty data; these list perceived "beauty" levels of professors along with their teaching evaluations. A beauty level of 0 means "average looking"; positive values signify above average beauty levels. In teaching evaluations a higher number signifies a better teaching score.

Load the data and plot it:

```
> beauty<-read.table("beauty.txt",header=T)
```

Fit a linear model of beauty as a predictor of teaching evaluation. Output a summary of the model fit. Discuss the interpretation of the estimated coefficients.

6.3 Mother and child IQ

Read in the data called kidiq.txt:

```
> kidiq<-read.table("kidiq.txt",header=T)
```

Here are the meanings of the terms used in this data set:

- kid score: the IQ of the child.
- mom hs: did mother go to high school? 1 if yes, else 0.
- mom iq: mother's IQ.
- mom age: mother's age

Answer the following questions:

1. Does the mother's IQ predict child's IQ?
2. Does the mother's high-school status (1=went to high school, 0=did not go to high school) predict child's IQ score?
3. Does mother's age predict child's IQ level?

6.4 2×2 factorial design

Read in the dataset noisdeg. Figure out what is in the dataset (you may need to look at the chapter on linear models).

Fit a model to look at the effect of degree of reaction time, and (in a separate model) the effect of noise on reaction time. Explain what the coefficients mean.

Chapter 7

Review exercises 3

For the `simdata3.txt` dataset (simulated data), fit a linear model of the form $rt \sim cond$, where cond represents two levels of a **categorical** predictor variable (i.e., cond is not a numerical predictor, such as beauty level).

Questions:

1. Explain what the coefficient estimates (the column marked Estimate in the output of the model) mean.

2. Are the residuals normally distributed?

3. What are the two null (and the respective alternative hypotheses) that the linear model is testing? Are the two null hypotheses rejected at $\alpha = 0.05$? Briefly explain your answer, referring to the standard error estimates, t-values, and p-values for explaining your decision.

4. For the second hypothesis test in the model (the one involving the cond factor), suppose the true distribution of δ, the difference in means, has mean 20. Sketch the distribution that represents the null hypothesis, and the true distribution, and then show the (a) Type I error region, (b) Type II error region, (c) the region representing power.

5. (This one requires some thought!) Still focusing on the second hypothesis test, given that the true distribution is centered around 20, and given (from the above model fit) that the estimated standard error is 4.2, what is the probability of correctly rejecting the null hypothesis (the power)? You can assume that the rejection region in the null hypothesis is bounded by -2 and 2.

Chapter 8

Course review questions 1

These questions cover the entire course. You should be able to solve all these questions (correctly!) within 2 hours.

8.1 Confidence intervals

The 95% confidence interval

a gives us a range that tells us that the population mean lies within this range with probability 95%.

b contains the sample mean with probability .95.

c tells us that we can be 95% sure that the sample mean is the true population mean.

d none of the above.

Briefly explain your answer (no more than one sentence!).

8.2 Type I error and p-value

Let us assume that you do an experiment and get a p-value from a t-test or whatever statistical test you do. What happens to the alpha value (Type I error probability) when the p-value is smaller than 0.05?

a alpha remains unchanged.

b alpha increases.

c alpha decreases.

d the answer depends on how much smaller the p-value is than 0.05.

Briefly explain your answer (no more than one sentence!).

8.3 Contrast coding

A factor with three levels (a, b, c) is used as a predictor for reading times (rt) in a linear model. Sample size is 20. You are given the following information:

```
> contrasts(dat1$cond)
  c a
b 0 0
c 1 0
a 0 1

> summary(mod)
Call:
lm(formula = rt ~ cond, data = dat1)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   490.46      17.24   28.44   <2e-16
condc          11.25      24.39    A      0.65
conda          -2.16      B      -0.089   0.93
```

1. Write down the sample means for each of the conditions a, b, c (to the nearest whole number is acceptable).

2. State what value should be in A; briefly explain how you got your answer. (it is enough to show a fraction).

3. State what value should be in B; briefly explain how you got your answer. (it is enough to show a fraction).

4. Write the three null hypotheses that are being tested in the rows labeled (Intercept), condc, and conda in the linear model.

5. For each of the three null hypotheses, state whether you would reject or fail to reject it.

6. For the hypothesis test associated with the row marked "conda" in the linear model output, repeated below:

```
            Estimate Std. Error t value Pr(>|t|)
conda         -2.16      B         -0.089     0.93
```

   what would the standard error have to be to reject the null hypothesis at $\alpha = 0.05$? Assume that the absolute critical t-value is 2.

8.4 Confidence intervals, power, Type I and II error probability

1. Given a sample with sample size 25, sample mean 50, and known standard deviation 5, specify a **90**% confidence interval for this sample (i.e., give the lower and upper bounds of the intervals). You may need this information:

```
> qnorm(0.05)
[1] -1.6449
```

2. Briefly explain what the qnorm function output above is telling you (do not hesitate to draw a sketch).

3. For the above problem, would you reject the null hypothesis that the true population mean is 45? Assume that you are willing to incorrectly reject the null hypothesis with probability 0.10. Carefully (but briefly!) explain your answer. Note: no p-value needs to be calculated.

4. For the above problem, sketch the distribution corresponding to the null hypothesis that the true mean is 45, and the distribution corresponding to the alternative that the true population mean is 60. Show the Type I error region, the Type II error region, and the region representing power.

8.5 More contrast coding

Consider the noise and degree data (we have seen this data set before in the lecture notes). We have 10 subjects, each of whom is shown a picture on the screen that is (a) masked by noise or no noise (i.e., there is one factor with level noise, an another with level no.noise), and is (b) angled at 0, 4, or 8 degrees. The dependent measure is recognition time in milliseconds. Here is subject 1's data to give you an idea of what the dataset looks like:

```
> MD497.df
rt subj deg    noise
420   s1   0 no.noise
420   s1   4 no.noise
480   s1   8 no.noise
480   s1   0    noise
600   s1   4    noise
780   s1   8    noise
...
```

We fit a linear mixed model given the following specifications:

```
> contrasts(MD497.df$deg)
  4 0
8 0 0
4 1 0
0 0 1
```

```
> summary(m0.lmer<-lmer(rt~deg+(1|subj),MD497.df))
Formula: rt ~ deg + (1 | subj)
Random effects:
 Groups   Name        Variance Std.Dev.
 subj     (Intercept)  3494     59.1
 Residual             11498    107.2
Number of obs: 60, groups: subj, 10

Fixed effects:
            Estimate Std. Error t value
(Intercept)    645.0       30.4   21.22
deg4           -60.0       33.9   -1.77
deg0          -168.0       33.9   -4.95
```

1. Briefly explain what the plot based on the following command would show us:

   ```
   qqPlot(residuals(m0.lmer))
   ```

2. Sketch a barplot (or any other kind of appropriate plot) summarizing the approximate mean reaction times for the three degree levels. It may help you to first work out the sample means for each level of the degree factor; this will help you in the subsequent parts as well.

3. If the contrast coding had instead been as below, what would the estimated coefficients in the linear model be? Give numbers for the letters C, D, and E below.

   ```
   > contrasts(MD497.df$deg)
     4 8
   0 0 0
   4 1 0
   8 0 1

   Fixed effects:
               Estimate  Std. Error t value
   (Intercept)     C       30.4   15.69
   deg4            D       33.9    3.19
   deg8            E       33.9    4.95
   ```

4. Briefly comment on whether the degree factor affects reaction time given the output in the immediately preceding part of this question.

Chapter 9

Course review questions 2

These questions cover the entire course. You should be able to solve all these questions (correctly!) within 2 hours.

9.1 Standard error

Standard error is

a the standard deviation of the sample scores.

b the standard deviation of the distribution of sample means.

c the square root of the sample variance.

d the 95% confidence interval.

9.2 Confidence interval

Given a particular sample, the 95% confidence interval is ——— the 90% confidence interval. (Fill in the blank above by circling one choice, and briefly explain your choice.)

a wider than

b narrower than

c the same width as

Explain your choice.

9.3 Power

When statistical power increases,

a Type II error probability decreases

b Type II error probability increases

c Type II error probability remains unchanged

9.4 Power, Type I and II error

Assume that the null hypothesis for a t-test is $H_0 : \mu = 0$, where μ is the population mean. Assume also that in reality $\mu = 4$.

1. Draw the distribution of the sample means under the null hypothesis (i.e., assuming that the null hypothesis is true). Directly below it, draw the distribution of the sample means that reflects the true state of affairs (i.e., that $\mu = 4$). Mark the Type I error region, the Type II error region, and the region representing statistical power (the plots are of course going to be approximate). No explanations are needed in words, just label the relevant regions asked for.

2. What would happen to Type II error probability if you reduce your Type I error probability? (No explanation needed).

3. What would happen to power if you reduce your Type I error probability? (No explanation needed).

9.5 Standard error

You are given a sample with mean $\bar{x}$, standard deviation s, and sample size n.

1. Write down the formula (just write the formula, no words needed!) for standard error, i.e., $SE_{\bar{x}}$.

2. What is the standard error an estimate of? (Now you need to use words and possibly also symbols to explain this.)

3. Let the null hypothesis H_0 be: $H_0 : \mu = 0$, where μ is the hypothesized population mean. Let the alternative hypothesis be $H_1 : \mu \neq 0$. We fix Type I error probability to 0.05. Let sample size be $n = 61$. R tells us what the critical t-value is for this sample size:

```
> qt(0.025,df=60)
[1] -2
```

[Recall that the function qt() tells you the critical t-value for a given Type I error probability and a given $n-1$ degrees of freedom.]

Suppose the standard error is $SE_{\bar{x}} = 20$, and the sample mean $\bar{x} = 40$. If one does a one-sample t-test, given all the above information, what would be the approximate p-value (to two decimal places) and t-value (to the nearest whole number) that the t.test function gives? Don't just write numbers, explain your answer.

9.6 Contrast coding

The `lexdec` data set in the library languageR has, among other things, log-transformed reaction times (i.e., in log milliseconds) of subjects for English words that they saw on a screen: they had to decide whether the word they saw was a word or not, and the reaction time represents the amount of time it took them to make the decision. Subjects were were either native speakers or not native speakers of English.

Our research question is: do native speakers make the lexical decision faster than non-native speakers?

Suppose we fit a linear model of log reaction time (RT) against native language status. Note that:

```
> contrasts(lexdec$NativeLanguage)
        Other
English     0
Other       1
```

I display the relevant output of the model below.

```
> summary(lm(RT~NativeLanguage,lexdec))

Coefficients:
                    Estimate   Std. Error t value
(Intercept)         6.32       0.01       849.78
NativeLanguageOther 0.16       0.01        13.72
```

1. Write down the null and alternative hypotheses for testing the research question above.
2. Explain what the coefficients (the intercept and slope) mean.
3. Given the above R output, and Type I error probability $\alpha = 0.05$, what would you conclude about the null hypothesis? Would you reject it? Explain your answer briefly.
4. Approximately how large would the standard error have to be to **fail** to reject the null hypothesis you specified above, at Type I error probability $\alpha = 0.05$? Briefly explain your answer.

Chapter 10

Solutions

10.1 Quiz 1 solutions

1. b
2. b
3. b
4. d
5. d
6. a
7. d
8. a
9. b
10. a
11. c

Bibliography

[1] R. H. Baayen. *Practical data analysis for the language sciences*. Cambridge University Press, 2008.

[2] A. Gelman and J. Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press, Cambridge, UK, 2007.

[3] John M. Hoenig and Dennis M. Heisey. The abuse of power: The pervasive fallacy of power calculations for data analysis. *The American Statistician*, 55(1):19–24, 2001.

[4] G. Jay Kerns. *Introduction to Probability and Statistics Using R*. 2010.

[5] D. S. Moore, G. P. McCabe, and Craig B. S. *Introduction to the Practice of Statistics*. W. H. Freeman, 2009.

[6] J.A. Rice. *Mathematical statistics and data analysis*. Duxbury press Belmont, CA, 1995.

[7] Ashish Sen and Muli Srivastava. *Regression Analysis: Theory, Methods and Applications*. Springer, New York, 1990.

[8] William N. Venables and Brian D. Ripley. *Modern Applied Statistics with S-PLUS*. Springer, New York, 2002.