Lessons in Statistical Thinking

Daniel Kaplan

11/16/2022

Table of contents

Preface

Note to students in Math 300

Up to now, you have been using the *OpenIntro* textbook. We will not be continuing into Block III of *OpenIntro*, but will replace it with the lessons in this little book.

Many of the topics in *OpenIntro* Block III are covered in the following chapters. But they are introduced in a fundamentally different way with a fundamentally different orientation. *OpenIntro* Block III is entitled "Statistical inference with `infer`" and shows how to compute various traditional statistical summaries. Those statistical summaries were developed during a specific era, roughly 1900 to 1950, and oriented toward the interpretation of bench-top lab experiments with a handful of observations. The purpose of the summaries was to indicate whether the experiment collected enough data to draw a definite conclusion and, later, as a bit of quality control for scientific journals.

Because of the orientation to laboratory experiments, the statistical summaries never had to deal with the common settings faced by today's data scientists. Today, it is common for data to be collected in large masses from *observations* rather than *experiments*. The analysis of data is often done for utterly different purposes. One common purpose is "prediction," which might be as simple as the uses of medical screening tests or as breathtaking as machine-learning techniques of "artificial intelligence." Another important purpose of data analysis is to understand possible causal connections between variables.

The work of today's data scientists is often to discover novel connections and to guide decision-making. That is far cry from the analysis of small, laboratory experiments.

> It turns out that some of the methods designed for the interpretation of experimental data are also useful in data science. But some of them are not so useful, such as classical "hypothesis testing." And every statistics book devoted to the traditional methods carries the warnings, "Correlation is not causation," and "No causation without experimentation." But in today's world, such a dogmatic attitude toward establishing causal connections does not reflect modern developments in statistical methods which have been designed to meet the broader needs of guiding decision-making and intervention in the world.

Instead of focusing exclusively on statistical inference, we are going to work with a broader idea called "statistical thinking." Statistical inference is a small part of statistical thinking, and hardly the most important part. Indeed, many statisticians and statistically-savvy scientists believe that statistical inference can be harmful and misleading. We will discuss the good reasons behind this belief in Lesson 38. If you can't wait, take a look at this article in the prestigious science journal *Nature*. Figure 1 reproduces a cartoon from that article that puts the shortcomings of "statistical significance" in a historical context.

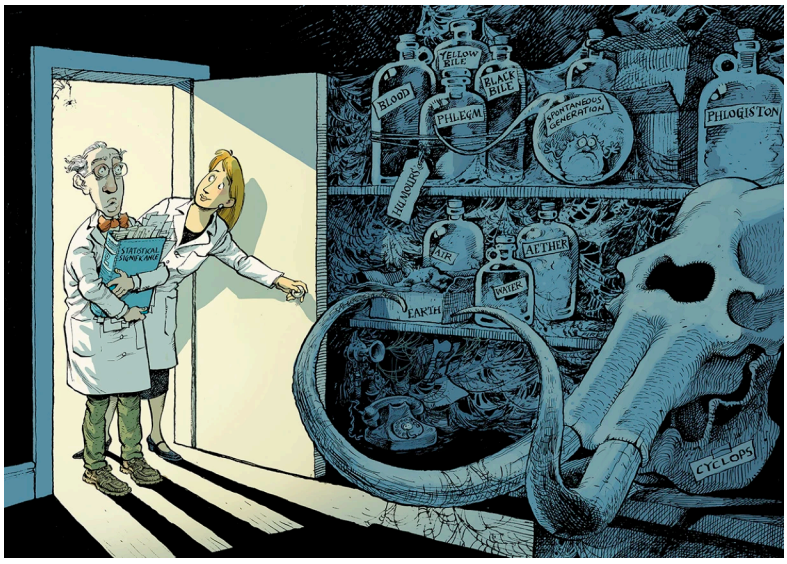


Figure 1: A cartoon published along with an article in *Nature*, "Retire statistical significance", showing this once-respected idea being relegated to the graveyard for outdated and misleading "scientific" concepts such as phlogiston and aether.

Statistical thinking

Over the next dozen lessons, you are going to be learning a way of thinking that is historically novel, unfamiliar to most otherwise well-educated people, and incredibly useful for making sense of the world and what data can tell us about the world. Learning a new way of thinking is genuinely hard. One reason is that you will have to suspend some of the familiar, go-to concepts that you've learned in school or through your reading.

To get you started with statistical thinking, it will help to have a concise definition of "statistical thinking." Here's one I like:

> *Statistic thinking is the explanation or description of measured variation* in the context of *what remains unexplained or undescribed.*

Implicit in this definition is a pathway for learning to think statistically: first, you need to learn how to use data to describe variation; second, you need to know how to measure "what remains undescribed" and to use that as a context for interpretation; third, you'll need to understand how "explanation" differs from "description." The lessons that follow will take you down this path.

1 Preliminaries

Prof. Danny Kaplan
November 17, 2022

This lesson introduces a few basic tools that you will be using throughout the remaining lessons.

1. A standard, unified format for data graphics that simplifies both the construction and the interpretation of graphics and permits layers of descriptions to be laid on top of a data layer.
2. The presentation of descriptions using **intervals** rather than a number like the mean or proportion.
3. A modern mode of displaying one type of description—the "**density**" (also called the "**distribution**") of data—that is compatible with the unified data-graphics format.
4. How extend regression modeling, which in Chapters 11-17 of *OpenIntro* always required a *quantitative* response variable, to be useful for modeling *categorical* response variables.

Since this is an introductory course, we will treat only categorical response variables that have two levels, for instance, Alive/Dead, Promoted/Not, Win/Loss, and so on. We will call these types of categorical response variables as "binomial" variables (that is, bi (two) nomial (names)) or "yes/no" variables, or zero/one variables. All of these terms refer to the same idea: a categorical variable with two levels.

Statistical techniques for handling categorical response variables with three or many more levels require more book-keeping and more intricate computer programming. The models used by the machine-learning community are called "classifiers"

rather than "regression models." But limiting ourselves in this course to binomial response variables means that classifiers are indeed regression models.

The core descriptive technique we will be using is based on regression models. And, as you know, a key paradigm for building regression models is the choice of a response variable and the choice of one or more explanatory variables. (Actually, the previous sentence would be more complete if it said, "the choice of **zero** or more explanatory variables. You'll see why a *zero explanatory variable* model is a useful concept as we move through the rest of the course. It is one of the main ways of"establishing context" for "what remains unexplained or undescribed. But we will cross that bridge when we come to it.)

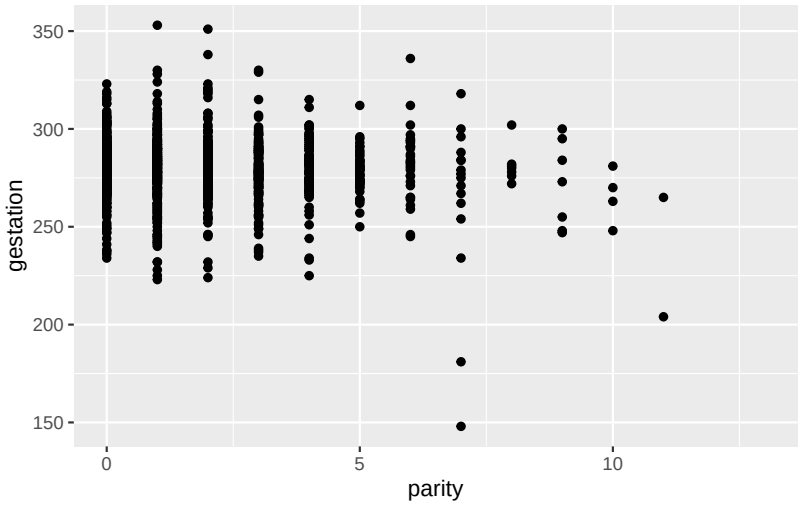
Since our descriptions will be grounded in regression models, and since we want to be able to generate graphics that show in different layers in the same graphics frame both the raw data and the description, it makes sense to structure data graphics so that there is a response variable displayed as well as one or more explanatory variables. Following convention, we will *always* display the response variable on the vertical (y) axis, and an explanatory variable on the horizontal (x) axis. If there are other explanatory variables to be displayed, we will use color and faceting.

Another aspect of our unified data graphic format is that it will *always* be a point plot or, closely related, a jitter plot.

To illustrate the construction of standard-format data graphics, consider the `mosaicData::Gestation` data frame. You can read about this data frame with the R command `?Gestation`. Suppose we want to address the question, "Do experienced mothers have systematically different gestation periods than inexperienced mothers?" For this question, an appropriate response variable is the length of `gestation`. The explanatory variable needs to measure "experience," which is a vague idea. We will make it concrete by taking it to mean the number of the mother's pregnancies prior to the one reported in the data. This is the variable `parity` and ranges from zero to thirteen.

Now that we know the response and explanatory variable, we can generate the data graphic simply enough:

```
Gestation %>% ggplot(aes(x=parity, y=gestation)) + geom_point()
```



This graph tells you some things at a glance. A typical gestation period is about 275 days (about 9 months). And you can see that it's much more common to have a low parity than a very high one. But perhaps there is some overplotting that's hiding the number of low-parity cases. We can easily resolve this by using `geom_jitter()`, perhaps with some transparency. At the same time, noting that there are very few cases with, say, parity greater than 5, we will focus on the part of the data with parity of zero to five:

```
Gestation %>%
  filter(parity <= 5) %>%
  #mutate(parity = as.character(parity)) %>%
  ggplot(aes(x=parity, y=gestation)) +
  geom_jitter(alpha=0.2, width=0.2, height=0)
```

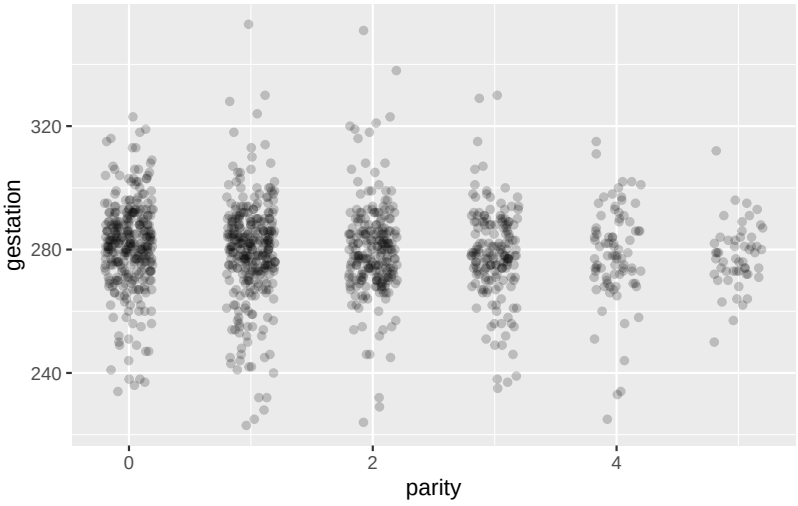


Figure 1.1: A jitter plot showing gestational period for pregnancies where the mother had five or fewer previous pregnancies. The `width=0.2` controls the amount of horizontal jittering. We chose it to make the columns of data clear. Also, there's no need to jitter in the vertical direction, so we set `height=0`

1.1 Displaying density

It is easy to see a pattern in Figure 1.1: It looks like mothers with high parity tend to have gestation periods that are more reliably close to 280 days than for mothers with low parity. Or, maybe this pattern is an illusion. There are so few pregnancies with parity 3, 4, or 5 that we don't expect to see as many uncommonly short or long gestational periods as for the parities with lots of cases.

One way to explore this idea is to plot the density of the dots as a function of gestation for each of the parity levels individually. A "violin" layer will make it easier to compare the distributions in the different columns, despite the unevenness in the case count. Figure 1.2 gives an example.

```
Gestation %>%
  filter(parity <= 5) %>%
  ggplot(aes(x=parity, y=gestation)) +
  geom_jitter(alpha=0.2, width=0.2, height=0) +
  geom_violin(aes(group=parity), fill="blue", alpha=0.2, color=NA)
```

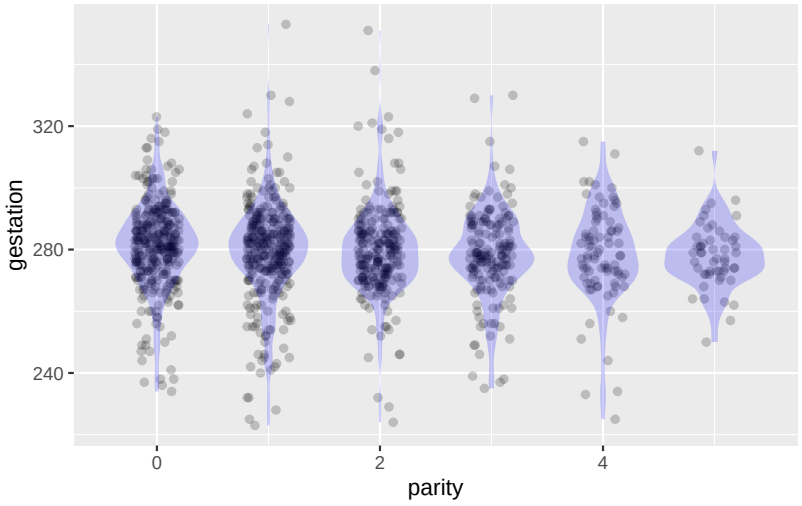


Figure 1.2: A violin plot. The long axis of the violin-like shape is oriented along the response-variable axis (that is, the vertical axis in our standard format). The width of the violin for each possible value of the response variable is proportional to the density of data near that value.

The violin plot is a more flexible display of the distribution of gestation period that would be a histogram. The histogram has all those bars that clutter up the display. Even worse, one of the axes in the frame of a histogram plot is "count" or maybe "density." Such a frame is not consistent with the unified response/explanatory format we will be using. The violin is drawn in the no-mans-land between the different levels of parity, just as the jittering moves data away from a single vertical line into that same no-mans-land.

This idea of using the graphical no-mans-land between levels of a categorical explanatory variable is not new. You encountered it earlier when you drew box plots. **?@fig-density-box** adds a box-plot annotation layer on top of the violin-plot layer.

```
Gestation %>%
  filter(parity <= 5) %>%
  ggplot(aes(x=parity, y=gestation)) +
  geom_jitter(alpha=0.2, width=0.2, height=0) +
  geom_violin(aes(group=parity), fill="blue", alpha=0.2, color=NA) +
  geom_boxplot(aes(group=parity), color="blue", fill=NA, alpha=.5)
```

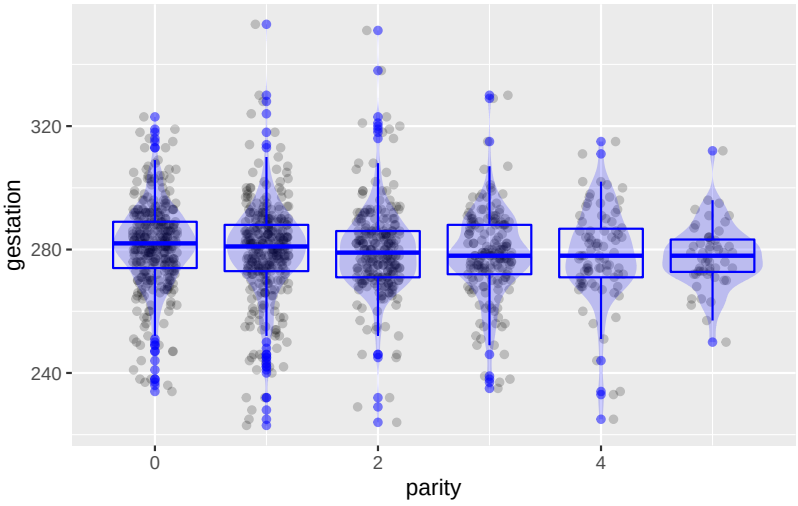


Figure 1.3: A box and whisker plot uses the no-mans-land between levels of a categorical explanatory variable.

Violins versus boxes

All of the graphical statistical annotations are human inventions. Each invention attempts to meet a need, but usually the invention is a compromise between the statistical objective and the computational and graphical resources available. The **box plot** format is a case in point. The statistical goal of a box plot is to display the distribution of values of a variable. It was invented in a time when graphics were mostly drawn by hand and computers were not widely available. The computations behind a box plot produce a five-number summary: min, first quartile, median, third quartile, max. It's straightforward (but tedious!) to do these by hand since they are based on sorting and counting. The drawing itself uses only straight lines, which are easy to draw by hand with only a pencil and a straightedge.

A cigar-box guitar

A violin plot requires hundreds or thousands of evaluations of the gaussian function along with post-processing.

They are not feasible for a human; a computer is required. Similarly, drawing the detailed shape of the violin (Figure 1.2) requires a computer.
The box plot has important deficiencies. It is appropriate only for uni-modal distributions and doesn't give even a hint of possible bi-modality. The sharp boundaries of the box and endpoints of the whiskers suggest that even smooth density shapes have abrupt transitions. Points are marked as "outliers" in order to keep the whiskers from becoming absurdly long, but box-plots of data with a normal (gaussian) distribution will produce such "outliers" whenever the sample size is large.
When it comes to computing power, we are today unimaginably rich compared to the generation that introduced box plots. In a sense, we are so rich we can use expensive, well made products such as a violin. The box-plot generation was living in computational poverty. Not having the (computational) funds to buy a violin, they had to make do with primitive instruments they had to make do with the materials at hand, just as early blues mucisians, coming out of poverty, often had to build instruments such as a cigar-box guitar.

1.2 Describing with intervals

Statistical thinking often involves quantifying uncertainty. One manifestation of this is moving away from single-number "**point**" summaries such as the mean or median to "**interval**" summaries. As you will see as we progress through the future lessons, there are many kinds of such intervals, each of which is designed to deal to address a specific question. So you'll see **prediction intervals**, **confidence intervals**, **confidence bands**, and so on.

For this lesson, we're concerned only with what interval summaries look like. So let's generate some, *without worrying* yet about the computer commands and mathematical underpinnings involved.

⚠ Ground rules for "demonstrations"

There will be many occasions in these lessons where we want you illustrate a statistical technique or phenomenon, but we don't expect the reader to master the commands involved. We will call these **demonstrations**: something we don't expect you to do at home. A good way to think about these demonstrations is that you should focus on the *outputs* from the calculations, rather than the calculation steps themselves. We'll show you the calculations since some readers might be interested, but focus your attention on the output.

ℹ Demonstration: Food at Starbucks

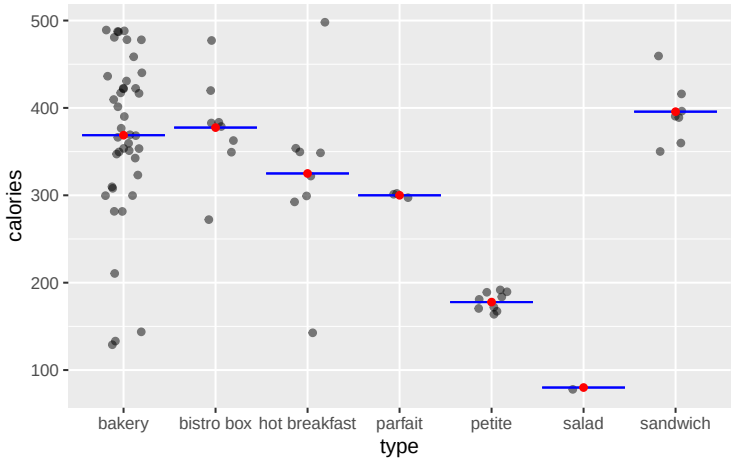
Starbucks is a famous coffee-shop franchise, with more than 30,000 branches (as of 2021) across the world. People go to Starbucks for the coffee, but they often buy something to eat as well. Let's look at the calorie content of Starbucks' food offerings. As always, when conducting a statistical analysis, it's helpful to have in mind the purpose for the task. We'll imagine, tongue in cheek, that we want to make food recommendations for the calorie conscious consumer.
First, a **point summary** of the calories in the different types of food products available at Starbucks:

```
point_summary <-
  df_stats(calories ~ type,
           data = openintro::starbucks, mean)
point_summary
```

```
  response          type     mean
1 calories        bakery 368.7805
2 calories    bistro box 377.5000
3 calories hot breakfast 325.0000
4 calories       parfait 300.0000
5 calories        petite 177.7778
6 calories         salad  80.0000
7 calories      sandwich 395.7143
```

This summary supports the sensible advice that to avoid calories, focus your choices on salads or on smaller portions (type "petite"). You might be tempted to go further, for example concluding that a sandwich is a bad choice (in terms of calorie content) so lean toward parfaits or hot breakfasts. You can even imagine someone concluding from this summary that a bistro box is a better calorie-conscious choice than a sandwich.
A graphic showing both the point summary and the raw data can put things in a useful context.

```
openintro::starbucks %>%
  ggplot(aes(x=type, y=calories)) +
  geom_jitter(width=0.2, alpha=0.5) +
  geom_errorbar(data=point_summary, aes(ymin=mean, ymax=mean),
                y=NA, color="blue") +
  geom_point(data=point_summary, aes(y=mean), color="red")
```



We've shown the point summary as red dots, one for each food type. A somewhat stronger visual impression is given by drawing the point summary not as points, but as lines that extend into the no-mans-land between food types. These are drawn in blue and they make the red dots superfluous; you don't need both.
Plotting the point summary in the context of the raw data

shows at a glance that the point summary is not of any use beyond the common sense advice to eat salads and small portions if you are trying to avoid calories. With the point summary on its own, we were tempted to conclude that, say, hot breakfasts are a better choice than sandwiches, but the data display suggests otherwise; there's just one low-calorie breakfast. The others are much like sandwiches.

A point summary is compact, but it fails to take into account the *variation* within each food type.

An interval summary does take into account this variation. This is an important aspect of statistical thinking. Recall the definition of statistical thinking given earlier:

> *The explanation or description of measured variation* in the context of *what remains unexplained or undescribed.*

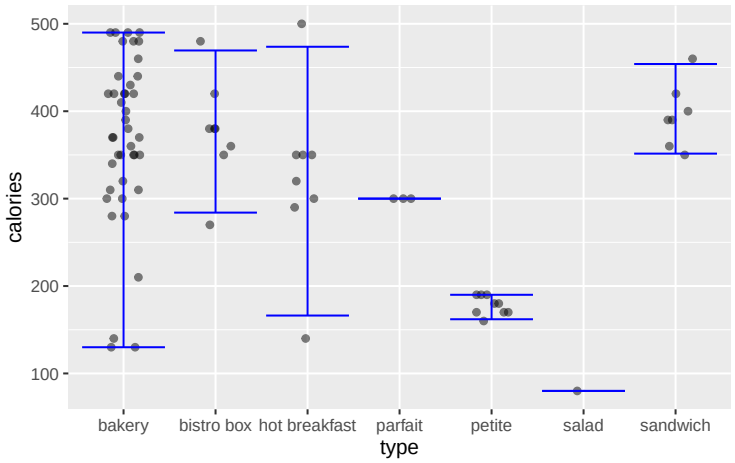
There are several kinds of interval summaries. You're not yet in a position to know which kind is the appropriate one for the task at hand—giving advice about food choices based on food type—so we'll tell you: a **prediction interval**.

```
interval_summary <- df_stats(calories ~ type,
                             data = openintro::starbucks, coverage(.95))
interval_summary
```

response	type	lower	upper
calories	bakery	130.00	490.00
calories	bistro box	284.00	469.50
calories	hot breakfast	166.25	473.75
calories	parfait	300.00	300.00
calories	petite	162.00	190.00
calories	salad	80.00	80.00
calories	sandwich	351.50	454.00

Or in graphical form:

```
openintro::starbucks %>%
  ggplot(aes(x=type, y=calories)) +
  geom_jitter(width=0.2, alpha=0.5, height=0) +
  geom_errorbar(data=interval_summary,
                aes(ymin=lower, ymax=upper), y=NA, color="blue")
```



Unlike point summaries, interval summaries can overlap. Such overlap is an indication that the groups being summarized are not all that different. Here, an appropriate conclusion indicated by the interval summary is, "Don't make your diet choices based on food type. Look at the calorie content of individual items before making your choice."

Admittedly, in this simple setting the data themselves would lead to the conclusion. But as we move into more complicated settings, it will become infeasible to quickly see patterns straight from the data. In these complicated settings, summaries are an important tool for displaying and quantifying patterns. *The statistical thinker knows to prefer interval summaries.*

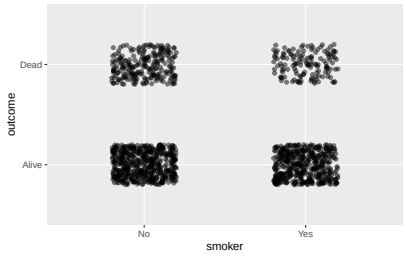
1.3 Categorical response variables.

Our last topic in this lesson is relatively simple: the zero-one transformation of categorical variables which allows regression and related techniques to be used for categorical response variables.

To illustrate, let's use data collected in the 1970s to study the relationship between smoking and mortality. The data we'll use, `mosaicData::Whickham`, recorded for one-thousand nurses whether or not they smoked at the time of the initial interview and whether or not they were still alive twenty years after the initial interview.

Here's a graph of the data in our standard response-vs-explanatory graphic frame:

```
Whickham %>%
  ggplot(aes(x = smoker, y = outcome)) +
  geom_jitter(width=0.2, height=0.2, alpha=0.5)
```



The graph suggests that non-smokers were more likely than smokers to be dead at the follow-up interview. But it's hard to calculate proportions from such a graph. It's reasonable to argue that for the purpose of showing the fraction of smokers and of non-smokers who died, a bar chart would be better.

The left barplot, showing counts, suggests that a higher proportion of non-smokers died than of smokers. But its easy to instruct the `geom_bar()` to graph proportions rather than counts, as done in the left plot. This makes it easy to conclude at a glance that a higher proportion of non-smokers have died.

The important question here, "Does smoking affect mortality?" translates well into the response/explanatory paradigm: `outcome` is the response variable while `smoker` is the explanatory variable. In the jitter-plot presentation of the data, these assignments are clearly indicated in the computer commands, which set `x=smoker, y=outcome`. In the barplot, a different notation is used: `x=smoker, fill=outcome`.

```
Whickham %>%
  ggplot(aes(x=smoker, fill=outcome)) +
  geom_bar()
Whickham %>%
  ggplot(aes(x=smoker, fill=outcome)) +
  geom_bar(position = "fill") +
  ylab("Proportions")
```

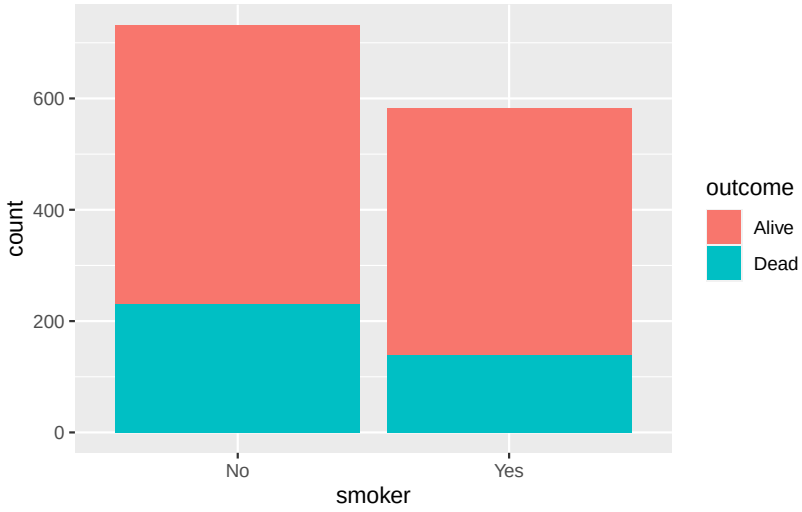


Figure 1.4: counts

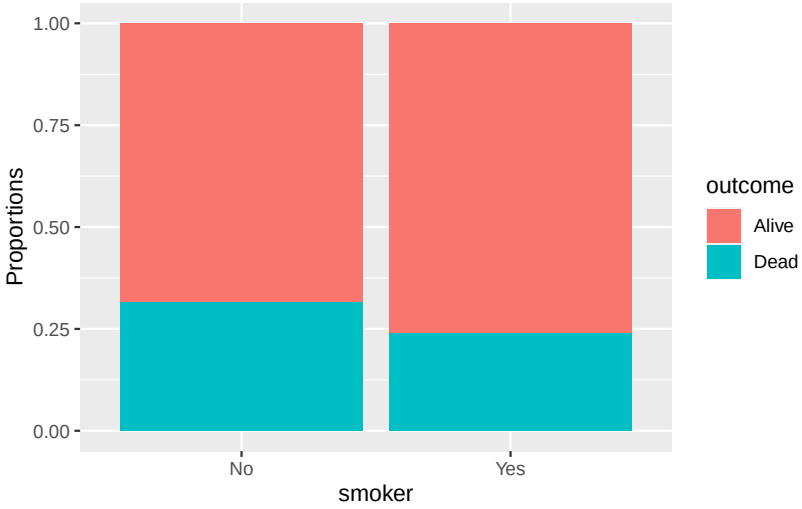


Figure 1.5: proportions

Figure 1.6: Barplots of the Whickham smoking and survival data.

Unfortunately, neither of the graphic styles—jitter or boxplot—answers the important question. At best they provide a description of the nurses in the `Whickham` data frame.

To answer the important question, we need to invoke statistical thinking. In particular, we need an *interval summary* of the proportion who died, not the point summary produced by the barplot.

This doesn't mean that we can't easily calculate the proportions from the categorical response variable: we just have to use the right commands. for instance:

```
Whickham %>%
  df_stats(outcome ~ smoker, prop, ci.prop)
```

```
  response smoker prop_Alive     lower    center     upper
1  outcome     No  0.6857923 0.6507824 0.6857923 0.7192969
2  outcome    Yes  0.7611684 0.7243939 0.7611684 0.7952677
```

The point summary—the `prop_Alive` column—suggests an obvious difference between the smokers and non-smokers. The interval summary—columns `lower` and `upper`—tempers this conclusion a little: the intervals almost touch.

Although regression is our go-to technique for modeling relationships between variables, we can't use it directly on a categorical response variable.

> **Warning**
>
> Here's what happens if we try:
>
> ```
> lm(outcome ~ smoker, data = Whickham) %>% confint()
> ```
>
> ```
> Warning in model.response(mf, "numeric"): using type = "numeric" with a factor
> response will be ignored
>
> Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors
>
> Warning in Ops.factor(r, 2): '^' not meaningful for factors
> ```

```
            2.5 % 97.5 %
(Intercept)    NA     NA
smokerYes      NA     NA
```

The computer's warning message is a reminder that the response variable is categorical. (The message uses the phrase "factor response," which is just computerese for "categorical response.")

To use regression with a two-level categorical response variable, transform the it into a zero-one encoding. In the following, we'll use 1 to represent `"Alive"` and 0 to represent `"Dead"`, although we can equally well do things the other way around.

```
lm(zero_one(outcome, one="Alive") ~ smoker, data = Whickham) %>%
  confint()
```

```
                2.5 %    97.5 %
(Intercept) 0.65329520 0.7182895
smokerYes   0.02654662 0.1242054
```

You don't yet know enough to interpret this interval summary. That will have to wait until Lesson 24. The significant[1] feature of the interval on `smokerYes` is that it does not include zero. In everyday terms, the interval means, "Smokers are 3 to 12 percentage points more likely to survive for 20 years than are non-smokers."

Using interval summaries instead of point summaries is an important aspect of statistical thinking, but there are other aspects that need to be taken into account. A simple, but important, question is whether the nurses recorded in the `Whickham` data frame are good representatives of all smokers. (It turns out that the nurses in `Whickham` are all women interviewed in the 1970s. At that moment of history, women were very different than men when it comes to smoking, and the Whickham

[1] In lesson 38 you'll learn to be wary whenever a statistician uses the word "significant."

smokers were also very different from today's female smokers. We'll say more about this in the demonstration below.)

Statistical thinking also leads one to ask another sort of question: What else might be going on other than smoking? In technical language, the other-goings-on are called "**covariates**," the topic of Lessons 28 & 29.

For instance, you might wonder about the overall result from our brief examination of the `Whickham` data. Is it really the case that the smokers were more likely to survive than the non-smokers? The answer is "yes," as we have demonstrated from the previous analysis. But this answer is completely misleading. Tobacco companies worked hard to mislead people into thinking that smoking was not dangerous. They knew full well the negative health consequence of smoking, but they used statistical-sounding claims to hide this knowledge from the public.

In the following demonstration, we'll look at the `Whickham` data again using the power of regression models to incorporate covariates.

i Demonstration: Smoking with covariates

Remember that you are not expected to master the calculations in these demonstrations. Focus your attention on the output from the calculations.

It goes without saying that smoking is not the only thing that kills people. There are other risky behaviors such as heavy drinking, there's environmental exposure to pollutants, and there's disease (other than the smoking induced ones of lung cancer, emphasema, and high blood pressure). But there's one risk factor for death that everyone knows about but nobody is doing anything about: getting old.

In virtually every public health or clinical study, the participants' age is taken into account. Not doing so can produce a completely misleading view of the situation. This is also the case with smoking and mortality in the `Whickham` study.

Regression techniques enable us to take multiple explanatory variables into account. In this demonstration, we'll

use regression to study `outcome` as a function of `smoker` and, importantly, `age`.
To get started, we need to remember to convert the categorical `outcome` variable into a zero-one encoding. After that, building the model is not so hard.

```
survival_model <- Whickham %>%
  mutate(survived = zero_one(outcome, one="Alive")) %>%
  model_train(survived ~ age + smoker, data=.)
```

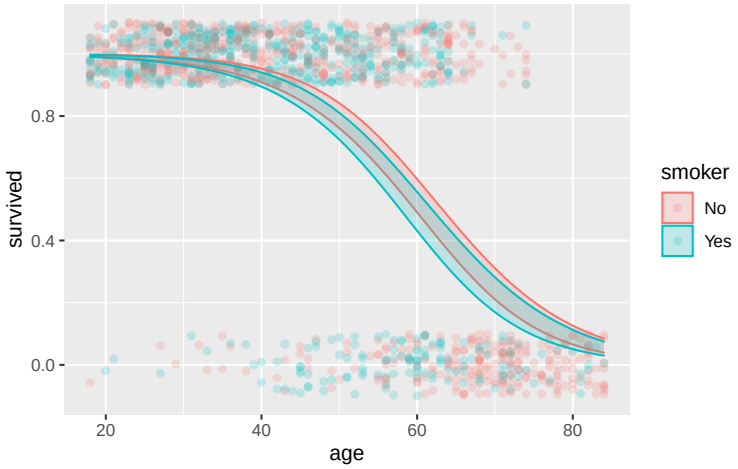
From this model, we can read off an interval summary of the effect of smoking on survival:

```
survival_model %>% confint()
```

```
                2.5 %     97.5 %
(Intercept)  6.7686824  8.5002535
age         -0.1382922 -0.1101260
smokerYes   -0.5369777  0.1238805
```

A full understanding of this interval summary will need to wait until Lessons 22 through 24. For the present, we'll simply point out that the summary interval on `smokerYes` includes zero, so `Whickham` provides no support for the mistaken conclusion that smoking improves survival. But seeing this requires taking into account `age`. A graphic may help explain why:

```
Model_output <- model_eval(survival_model, interval="confidence")
Model_output %>%
  ggplot(aes(y = survived, ymin=.lwr, ymax=.upr, x=age, color=smoker, fill=smoker)) +
  geom_jitter(height=.1, width=0, alpha=0.2) +
  geom_ribbon(alpha=0.2)
```



The interval summary in the graph shows how the probability of survival changes for different ages. The intervals for non-smokers and smokers entirely overlap. For both groups, 20-year survival goes down with greater initial age. So why did the model `outcome ~ smoker` suggest that smokers have a higher survival? The reason relates to the proportion of smokers with initial age 70+. In the 1970s, life expectancy was such that people 70+ were unlikely to survive 20 years. This pulls down the survival rate at that age. Notice that the 70+ nurses were unlikely to have been smokers compared to younger nurses. The 70+ nurses grew up in an era when social conventions caused smoking to be uncommon for women (even though it was very common for men).

1.4 Wrangling versus modeling

The first half of this course emphasized data wrangling and visualization. When using data wrangling commands, summaries of data frames were computed using, naturally enough, the `summarize()` function. Typical summaries for quantitative variables include means, medians, standard deviations, etc., each of which applies to one variable at a time. For instance,

this command calculates four summary statistics on the `net` running time recorded in the `TenMileRace` data frame:

Constructing such summaries groupwise is a matter of using the `group_by()` modifier. Here, we calculate summaries broken down by the state of residence of the participant and arranged from fastest (average) running time downward.

```
TenMileRace %>%
  group_by(state) %>%
  summarize(ave = mean(net), middle = median(net), sd = sd(net), n = n()) %>%
  arrange(ave) %>%
  head(10)
```

```
# A tibble: 10 x 5
   state       ave middle    sd     n
   <fct>     <dbl>  <dbl> <dbl> <int>
 1 Australia 2872   2872    NA       1
 2 Kenya     2934.  2874.  141.     14
 3 Lithuania 2961   2961    NA       1
 4 Japan     2992   2992   132.      2
 5 Colombia  2998   2998    NA       1
 6 Ethiopia  3185   3185   356.      2
 7 EN        3251   3251    NA       1
 8 Ukraine   3256   3256    NA       1
 9 Russia    3267   3197   272.      3
10 Romania   3287.  3297   162.      3
```

Wrangling is essential for many statistical purposes, including making graphical displays, cleaning data, and assembling data that comes from multiple sources.

In Lessons 11-17, you were introduced to regression modeling. The computing tasks for regression, such as fitting a model with `lm()`, don't fit into the wrangling framework. To illustrate:

```
lm(net ~ age, TenMileRace)
```

```
Call:
lm(formula = net ~ age, data = TenMileRace)

Coefficients:
(Intercept)          age
    5297.22         8.19
```

Regression produces a summary of a data set, somewhat like `summarize()`. The output of `summarize()` is always a data frame. The output of `lm()` is a different kind of thing, as you can see above. In computing lingo, the "kind of thing" is often called the "object class" or "object type."

Perhaps the first thing that's confusing about the object produced by `lm()` is that they contain much more than what's printed out when you display them directly on the screen. The printed output is just a glimpse at the object and, almost always, you use another function to present the content of the object in a way that suits your current need. Let's informally call such functions "extractors." Here are examples of a few of the extractors that you will be using in the coming lessons; you are not expected at this point to know what each is doing.

```
lm(net ~ age, TenMileRace) %>% coefficients()
```

```
(Intercept)         age
5297.219248    8.189886
```

```
lm(net ~ age, TenMileRace) %>% rsquared()
```

```
[1] 0.008014284
```

```
lm(net ~ age, TenMileRace) %>% regression_summary()
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  5297.      37.6      141.   0
2 age             8.19     0.981      8.35 7.76e-17
```

```
lm(net ~ age, TenMileRace) %>% model_eval(skeleton=TRUE)
```

```
  age  .output     .lwr     .upr
1  10 5379.118 3485.069 7273.167
2  20 5461.017 3567.394 7354.640
3  30 5542.916 3649.525 7436.307
4  40 5624.815 3731.460 7518.170
5  50 5706.714 3813.200 7600.227
6  60 5788.612 3894.744 7682.480
7  70 5870.511 3976.094 7764.928
8  80 5952.410 4057.249 7847.571
9  90 6034.309 4138.210 7930.408
```

Much of what you will be learning in the following Lessons concerns such regression extractors: why what they calculate is useful and how to use it. But let's return to the command patterns you will be seeing.

Regression and wrangling are allies. You will use wrangling, especially `mutate()` and `filter()` to pre-process data before carrying out the regression. For instance, in studying the relationship between `age` and `net` running time, you might want to focus on older people.

```
TenMileRace %>%
  filter(age > 40) %>%
  lm(net ~ age, data=.) %>% confint()
```

```
                2.5 %     97.5 %
(Intercept) 4014.7081 4541.71744
age           22.8315   33.43884
```

In this example, we've used `confint()` as the extractor, but, depending on our purpose, we might have used any of the others.

Looking closely at the above command you will notice something new: the `data=.` argument being used inside `lm()` . The simple `.` is doing something important, carrying the output of the earlier stages of the pipeline into the `data=` argument of `lm()`.

The dot (`.`) has always been available for use when wrangling, but we haven't needed to use it. For instance, here is an earlier example of a wrangling command translated to use the `.` notation:

```
TenMileRace %>%
  group_by(., state) %>%
  summarize(., ave = mean(net), middle = median(net), sd = sd(net), n = n()) %>%
  arrange(., ave) %>%
  head(., 10)
```

```
# A tibble: 10 x 5
   state       ave middle    sd     n
   <fct>     <dbl>  <dbl> <dbl> <int>
 1 Australia 2872   2872    NA       1
 2 Kenya     2934.  2874.  141.     14
 3 Lithuania 2961   2961    NA       1
 4 Japan     2992   2992   132.      2
 5 Colombia  2998   2998    NA       1
 6 Ethiopia  3185   3185   356.      2
 7 EN        3251   3251    NA       1
 8 Ukraine   3256   3256    NA       1
 9 Russia    3267   3197   272.      3
10 Romania   3287.  3297   162.      3
```

The `.` always means, use the output of the preceding stages of the pipeline as *this argument* to the function. All of the data wrangling commands were designed so that the first argument is always a data frame. The phrase `%>% group_by(., state) %>%` is an explicit direction to place the data coming from the pipeline as the first argument. The pipeline connector, `%>%`,

is arranged by default to put the content it is transmitting as the first argument to the following function. For this reason, `.` is not needed if the pipeline pumps results into the first argument.

But not all functions are designed with this convention in mind. In particular, in `lm()` the first argument should be the model formula, e.g. `net ~ age`. The data frame that will be used in fitting the model is the second argument. So, in `lm(net ~ age, data=.)` the dot is directing the pipeline to empty its data frame into the second argument. Otherwise, by default, the pipeline would be plumbed to force the first argument to be the data frame and demote the `net ~ age` to the second position.

1.5 Grouping

The `group_by()` wrangling function is used whenever you want to treat a data frame in a group-by-group manner. It's natural to assume that `group_by()` is slicing up a data frame according to groups. The `mutate()` and `summarize()` functions were specifically designed to make it appear that `group_by()` is doing the slicing. But in reality, `group_by()` is just adding a kind of tag to the data frame. `mutate()` and `summarize()` know to interpret that tag as an instruction to slice up the data when calculating on it.

Most R functions simply ignore the tag added by `group_by()`. Consider, for instance, the pipeline

```
TenMileRace %>%
  group_by(sex) %>%
  lm(net ~ age, data=.) %>%
  coefficients()
```

```
(Intercept)         age
5297.219248    8.189886
```

The output is not what you might expect from your earlier experiences using `group_by()`. Particularly, there is *not* a separate row for each of the groups defined by `sex`. Nor is there a column listing the sex. The `group_by()` has had no effect.

What do you do if you really want to compare groups using regression models? For instance, it might be that you believe that the `F` group ages differently than the `M` group. How do you reveal this with regression?

Regression models have their own, internal system for comparing groups. You express your wish to compare groups by using the model formula. The simple formula `net ~ age` does not instruct `lm()` to compare the sexes. Instead, you would use the formula `net ~ age*sex`. For instance,

```
TenMileRace %>%
  lm(net ~ age * sex, data = .) %>%
  coefficients()
```

```
(Intercept)         age        sexM    age:sexM
5370.999953   15.961661 -785.145163    1.606559
```

The output of `lm()` already contains the comparison information. You don't yet know how to read and interpret that comparison information, but you will learn.

There are extremely good reasons why `lm()` does things the way it does. It is not at all a matter of software incompatabiility between the wrangling family of commands and the regression family. The `lm()` paradigm can make much more efficient use of data than `group_by()`. It also offers much more flexibility. `lm()` can handle multiple "grouping" variables together and even lets you "group" by quantitative variables. These capabilities are extremely important for extracting relevant information from data, as you will see in the following lessons.

1.6 Learning challenges

1. One of these pipeline commands will work and the other won't. Which one will work? Explain why the other one doesn't work.

```
lm(net ~ age, data = TenMileRace)
TenMileRace %>% lm(net ~ age)
```

2. An example from the *OpenIntro* book uses data on promotions. Some data wrangling commands that might be relevant are these:

```
promotions %>% tally()
```

```
# A tibble: 1 x 1
      n
  <int>
1    48
```

```
promotions %>% group_by(decision) %>% tally()
```

```
# A tibble: 2 x 2
  decision     n
  <fct>    <int>
1 not         13
2 promoted    35
```

```
promotions %>% group_by(gender) %>% tally()
```

```
# A tibble: 2 x 2
  gender     n
  <fct>  <int>
1 male      24
2 female    24
```

```
promotions %>% group_by(gender, decision) %>% tally()
```

```
# A tibble: 4 x 3
# Groups:   gender [2]
  gender decision     n
  <fct>  <fct>    <int>
1 male   not          3
2 male   promoted    21
3 female not         10
4 female promoted    14
```

You could use such wrangling to compare groups. For instance, you can use the results of the last command to calculate separately the proportion of males who were promoted and, similarly, the proportion of females.

a. **What are those proportions?**

The following wrangling command will calculate the proportions for you, but it is a bit complicated:

```
promotions %>%
  group_by(gender) %>%
  summarize(prop_promoted = sum(decision=="promoted") / n())
```

b. **Use the above command to check your calculations in (a).**

c. In the regression paradigm, the comparison of proportions between the two groups is done directly in `lm()`, like this:

```
promotions %>%
  mutate(promoted = zero_one(decision, one="promoted")) %>%
  lm(promoted ~ gender, data = .) %>%
  coefficients()
```

```
 (Intercept) genderfemale
   0.8750000   -0.2916667
```

We'll explain the purpose of `zero_one()` in Lesson 19, but putting that matter aside for a moment, compare the two coefficients in the regression model to the proportion results you got from wrangling.

i. **What does the value of the intercept coefficient correspond to in the wrangling results?**

ii. **What does the `genderfemale` coefficient correspond to in the wrangling results?** (Hint: you will have to do a bit of arithmetic on the wrangling results.)

2 Measuring and simulating variation

Prof. Danny Kaplan
November 17, 2022

This Lesson introduces two ideas. The first is how to measure variation. This is important, as you can see from the definition of statistical thinking given in the previous Lesson:

> *Statistic thinking is the explanation or description of measured variation* in the context of *what remains unexplained or undescribed.*

Variation is what we're trying to explain/describe. To do this, it helps to be able to measure variation.

The second idea is also fundamental to statistical thinking. Often, but not always, our interest in studying data is to reveal the causal connections between variables. This is important, for instance, if we are planning to make an intervention in the world and want to anticipate the consequences. Interventions are things like "increase the dose of medicine," "stop smoking!", "lower the budget," "add more cargo to a plane (which will increase fuel consumption and reduce the range)."

Historically, statisticians were hostile to the idea of using data to explore causal relationships. The one exception was **experiment**, where the data come from an actual intervention in the world. (See Lesson 32.) Statistics teachers encouraged students to use phrases like "associated with" or "correlated with" and reminded them that "correlation is not causation."

Regretably, this attitude made statistics irrelevant to that part of the real world where intervention was the matter of interest and experiment was not feasible. A tragic episode of this sort likely caused millions of unnecessary deaths. Starting in the 1940s, doctors and epidemiologists were seeing evidence that smoking causes lung cancer. In stepped the most famous statistician of the age, Ronald Fisher, to insist that the statement should be, "smoking is associated with lung cancer." He speculated that smoking and lung cancer might have a common cause, perhaps genetic. To establish causation, it would be necessary to run an experiment where people are randomly assigned to smoke or not smoke and then observed for decades to see if they developed lung cancer. Such an experiment is unfeasible and unethical, to say nothing of the need to wait decades to get a result.

Fortunately, around 1960 a researcher at the US National Institutes of Health, Jerome Cornfield, was able to show mathematically that the strength of the association between smoking and cancer ruled out any genetic mechanism. This was one of the first developments in a field called "causal inference"

> *Variation itself is nature's only irreducible essence. Variation is the hard reality, not a set of imperfect measures for a central tendency. Means and medians are the abstractions.* —— Stephen Jay Gould (1941- 2002), paleontologist and historian of science

A common task in statistical modeling is to break down a variable into components. For instance, a person's height or intelligence or charm is presumably a combination of genetics and environment. In doing this breaking down, it's convenient to be able to characterize the **size** of each component.

There are many possible ways to measure "size." In this course, we will emphasize two, intimately related measures:

i. *variance*
ii. *standard deviation*, which is simply the square root of variance.

The *OpenIntro* text introduced the standard deviation in Chapter 3 where it was described as a measure of "spread." In

Chapter 6, *OpenIntro* introduced the variance as the square of the variance. All this is right, so far as it goes, but it dramatically understates the importance of the two measures. These measures are as important to statistical thinking as the Pythagorean Theorem is to geometry.

You remember the Pythagorean Theorem: $A^2+B^2 = C^2$, where C is the length of the hypothenuse of a right triangle, and A and B are the lengths of the other two sides of the triangle. Surprisingly, the Pythagorean Theorem is highly relevant to statistical models.

Recall from *OpenIntro* chapters 5 & 6 that the linear modeling technique produces two columns of numbers: the **fitted values** and the **residuals**. These columns have the same number of rows as the data frame used for training. The fitted values are the output from the model when the explanatory variables from the training data are given as inputs to the model. The residuals are the row-by-row numerical *difference* betwen the response variable and the fitted values.

These three columns of numbers—the response variable, the fitted model values, and the residuals—are exactly analogous to the three sides of a right triangle. (This is not an obvious fact, but it is an important one to keep in mind.) In particular, the following numerical relationship is as true for linear models as it is for triangles:

$$\text{sd(fitted)}^2 + \text{sd(residuals)}^2 = \text{sd(response)}^2$$

where sd() refers to the standard deviation. Consequently, $\text{sd}()^2$ is the variance.

The variance and the standard deviation are defined mathematically in a special way that makes the Pythagorean relationship always describe models constructed the the `lm()` technique, that is "least squares" models.

> **Why "standard deviation"**
>
> "Standard deviation" is an antique term and is misleading to people who think about "deviation" in the everyday

sense. The term is so widely used that we can hardly avoid it, but it is helpful to have in mind a modernization of "standard deviation."
Step 1 in the modernization is to make clear what "standard" means:

> standard deviation = accepted *measure of* deviation.

Step 2 in the modernization replaces the archaic word "deviation" with something more descriptive:

> standard deviation = accepted *measure of* variation in the variable.

We won't explain here *why* the standard deviation became the go-to, accepted, standard measure of variation, but it did and for excellent reasons.

Both variance and standard deviation are **quantities**, that is, a single number with associated units. The standard deviation of any variable has units that are exactly the same as the variable itself. For instance, the measured heights of a group of people is often measured in cm. So the units of the standard deviation of height will also be in cm.

In contrast, the variance, being the square of standard deviation, has units of the square of the units of the variable. The variance of height, for instance, will be measured in cm^2. This will seem odd at first glance, but you have to get used to it: the variance of a variable has units that are the square of the units of the variable itself.

Keep in mind also that variance and standard deviation are *summaries* of a variable. A variable in a data frame consists of multiple values, one for each row of the data frame. The variance or standard deviation of that variable will be just a single number (and units), summarizing all of the values in the variable.

Calculating variance

Almost always, people use software to do the calculations. The relevant R functions are `sd()` and `var()`. You can use these functions in a `summarize()` statement, for instance

```
mtcars %>% summarize(v = var(hp))
```

```
          v
1 4700.867
```

```
mtcars %>% summarize(s = sd(hp))
```

```
          s
1 68.56287
```

Regrettably, the software does not indicate the units of the quantity. For that, you need to determine the units of the variable itself, typically by reading the documentation for the data frame.

To understand *what* is being calculated by `var()`, we will describe an algorithm. There are more efficient algorithms than the one described here, but this one is easy to understand.

Starting material:

- A single column of numbers creating by pulling out from the data frame the variable whose variance is to be calculated.
- A long roll of paper on which you can write numbers, one after the other.

Basic calculation: You are going to repeat a calculation for each and every row in the column of numbers. To illustrate, suppose you are doing the calculation for the k^{th} row. Take the data value from the k^{th} row, and call it the "reference value." Then subtract the reference value from each and every other value in the column and square the results. Write those numbers, all of them, on the roll.

Using the same roll of paper for all, carry out the basic calculation starting at each of the rows in the single column of data. Now your roll of paper has many numbers on each, each of which is the square difference between the values from two rows of the table. If you are mathematically inclined, you might like to know that there will be exactly $n(n-1)$ numbers written on the roll. If you are a statistical instructor, your ears might perk up when you notice the $n-1$ in that count.

The final result—the variance—will be the mean of the numbers on the roll. Since each of the numbers on the roll is the square difference between two values of the variable, the mean will be the average square difference.

⚠ For the statistically experienced reader …

Warning! This box contains mathematical formulas that are **not needed for the course**. The formulas might be interested to mathematically inclined statistics instructors. If that's not you, skip this material.
I realize that the algorithm described above is probably not used by any statistical software package. It's really inefficient numerically.
You can see this by comparing the traditional formula for the variance to the formula version of the above algorithm:

$$\underbrace{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2}_{\text{traditional}} \quad \text{versus} \quad \underbrace{\frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{n-1}\sum_{j\neq i}(x_i-x_j)^2\right]}_{\text{our algorithm}}$$

The inefficiency of the algorithm stems from the double sum. The advantage of the algorithm is conceptual and two-fold:

i. You can see where the $n-1$ in the formula for the variance comes from: the inner sum involves $(n-1)$ numbers. No hand waving needed to explain the $n-1$. (What might need explaining is the $j \neq i$ in the inner sum. Why not $\sum_j = 1^n$? Because that would put n zeros on the roll and bias the result

downward. We want to average the square distance between each value and *every other value*.)

ii. There is no need to introduce the mean $\bar{x}$ of the values. Of course, $\bar{x}$ is easy and fast to calculate so there is no numerical reason to avoid using it in the calculation. There is, however, a philosophical reason based on Stephen J. Gould's observation, quoted at the start of this lesson: "Variation is the hard reality. Means … are the abstractions."

Here's a traditional-minded definition: "Variance is equal to the average squared deviations from the mean." This definition makes it seem like the mean has some special status and that variations from it are "deviations."

2.1 Causality

The introduction to this chapter contained a very brief mention of a causal relationship: changing the dose of a medication to, say, lower a patient's blood pressure. Assuming that the drug is effective, it's common sense that the change in dose had a causal influence on the patients' condition. A natural belief that one thing can cause another is the entire basis for medical and other interventions.

The historical statisticians who insisted that data alone cannot establish a causal connection would also, nonetheless, go to the doctor for treatment. Without an experiment, professional pride would lead them to stipulate that data can only establish "correlation" or "association" and that it's impossible to say from data what causes what. But in their everyday lives they believed that medication has a causal influence. How could they justify this belief given their professional attitudes toward causality? Because they have common sense and know something about how the world works.

This section is about formal ways to say "something about how the world works" that can be used, along with data, to make responsible conclusions about causal relationships.

2.2 Directed acyclic graphs

The title of this section is a mouthful, but the mathematical structure of a "directed acyclic graph" (DAG, for short) is one of the most popular ways for statistical thinkers to express their ideas about what might be going on in the real world. Despite the long name, DAGs are very accessible to a broad audience. You may even have constructed one without knowing the formal name.

Statistical graph**ics** are so common and so often used in this course, that you may think that the "graph" in DAG refers to these. Not really, even though statistical thinkers often draw pictures of DAGs. The "graph" in DAG is a mathematical term of art; a good synonym is "network." Mathematical graphs consist of a set of "nodes" and a set of "edges" connecting the nodes. **?@fig-graphs** shows three different graphs, each one having five nodes labelled A, B, C, D, and E.

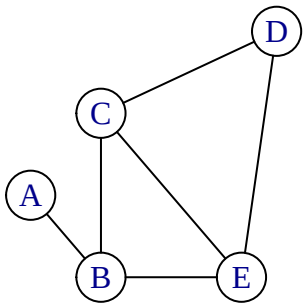


Figure 2.1: undirected graph

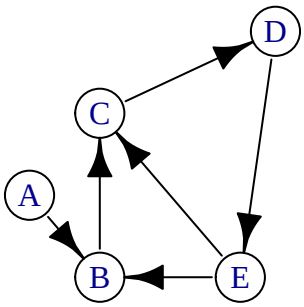


Figure 2.2: directed but cyclic

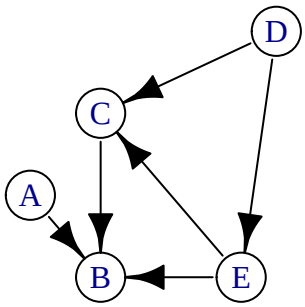


Figure 2.3: directed acyclic graph

The nodes are the same in all three graphs of **?@fig-graphs**, but each of the graphs is different from the others. That's because it is not just the nodes that define a graph; the edges (drawn as lines) are part of the definition as well.

The left-most graph in **?@fig-graphs** is an "**undirected**" graph; there is no suggestion that the edges run one way or another. In contrast, the middle graph as the same nodes and edges, but the edges are **directed**. A nice way to think about a directed graph is that each node is a pool of water and each directed edge shows how the water flows between pools. This

analogy is also helpful in thinking about causality: the causal influence flow like water.

Look more carefully at the middle graph. There is a couple of loops; the graph is **cyclic**. In one loop, water is flowing from E to C to D and back again to E. The other loop runs B, C, D, E, and back to B. Such a flow pattern could not exist unless some of the edges involved pumps to push the water back uphill.

In the right-most graph, some of the edges have had their direction reversed. This graph has no cycles; it is **acyclic**. To use the flowing and pumped water analogy, in an acyclic graph no pumps are needed. You could arrange the pools of water at different heights to create the flow through gravity. The node-D pool would be the highest, E and C a little lower. But C has to be lower than E in order for gravity to pull water along the edge from E to C. The node-B pool has to be the lowest of all so that water can flow to it from E, C, and A.

Directed acyclic graphs are used to represent causal influences; think of "A causes B" as meaning that causal "water" flows naturally from A to B.

In a DAG, a node can have multiple outputs, like D and E, and it might have multiple inputs, like B and C. In terms of causality, when a node—like B—has multiple inputs it means merely that more than one factor is responsible for the value of that node. A real-world example: the rising sun causes a rooster to crow, but so can another intruder to the coop.

Often, nodes do not have any inputs. These are called "**exogenous factors**"—at least by economists. The "genous" beens "originates from," the "exo" means "outside." The value of an exogenous node is determined by something, just not something that we are interested in (or perhaps capable of) modeling. But we can be sure that the exogenous node is not determined by any of the other nodes in the DAG. Otherwise there would need to be an arrow drawn into the node. But then it wouldn't be exogenous.

2.3 Using DAGs

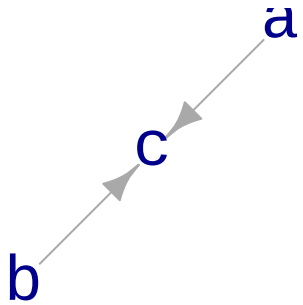
The point of a DAG is to make a clear statement of a hypothesis about causation. Drawing a DAG doesn't mean that the hypothesis is right, just that we believe the hypothesis is in some sense a possibility. Different people might have different beliefs about what causes what in real world systems. Comparing their different DAGs can help, sometimes, to discuss and resolve the disagreement.

We are going to use DAGs for two distinctive purposes. One purpose is to inform responsible conclusions from data about what causes what. The data on its own is not sufficient to establish what are the causal connections. But data *combined with* a DAG can tell us something. Sometimes a DAG includes a causal connection that should create an association between variables. Not seeing that association in the data is evidence that the hypothesis behind the DAG is wrong. DAGs are also useful for building models; they can tell us which variables to include and which to exclude from a model in order to capture the hypothetical causal connections.

The second purpose is for learning modeling technique. We are going to generate simulated data from various DAGs. You can see exactly what is going on the the DAG and then see whether your statistical analysis of data is capturing the known mechanism of the DAG. This is useful for learning what can go right or wrong in building a model, just as a aircraft simulator is useful for training pilots to handle real-world situations in real aircraft. The DAGs that we will use for this second purpose consist of formulas that relate the value of each variable to the values of its inputs. The value of exogenous nodes is usually set randomly, without any input from the other nodes in the DAG.

Here's a simple example: a DAG with two exogenous nodes (`a` and `b`) and one another node, `c`, that gets input from both `a` and `b`.

```
dag_draw(dag09)
```



```
print(dag09)
```

```
[[1]]
a ~ eps()

[[2]]
b ~ eps()

[[3]]
c ~ binom(2 * a + 3 * b)

attr(,"class")
[1] "list"      "dagsystem"
```

The `dag_draw()` command draws a picture of the graph. Printing the value of the dag gives the formulas that set the values of the nodes. In `dag09`, the nodes `a` and `b` are both set at random and independently of one another. That's what the `eps()` function does: set the value at random. In contrast, the formula for node `c` says that the value of `c` will be a linear combination of the values of `a` and `b`, translated into a zero-one format.

You will be generating simulated data from such dags using the `sample()` function. For instance,

```
sample(dag09, size=5)
```

```
# A tibble: 5 x 3
       a      b     c
   <dbl>  <dbl> <dbl>
1 -0.326  1.17      1
2  0.552  0.619     0
3 -0.675 -0.113     0
4  0.214  0.917     1
5  0.311 -0.223     0
```

Each of the rows in the sample is one trial in which the values of the nodes have been assigned, either by random numbers

for exogenous nodes, or by a formula for nodes that receive inputs.

> ⚠ Reality check: DAGs and data
>
> DAGs are a way to lay down on paper your beliefs about what is connected to what in the real world. They are also a kind of scratchpad for constructing alternative scenarios and, as you'll see starting in Lesson 28, thinking about how your models might go wrong in the face of a plausible alternative.
> In this book, we extend the use of DAGs beyond their scope in professional statistics; we use them as simulations from which we can generate data. This can be a good way to learn about statistical methodology.
> DAGs are aides to reasoning, scratchpads that help us play out the consequences of our hypotheses about possible real-world mechanisms. But data from DAG simulations shouldn't be confused with data from reality.
> If you want to know about the real world, you need to collect data from the real world. The proper role of DAGs in real work is to guide model building **from real data**.
> In this course, we sample from DAGs to learn statistical technique. But never to make claims about real-world phenomena.

As the name suggests, `sample()` collects a sample of data. Typically, you will then summarize the data, often by fitting a model to the data and then summarizing the model. To illustrate, here we generate a sample of size $n = 10,000$, then fit the model `c ~ a + b`, and summarize by taking the coefficients.

```
sample(dag09, size=10000) %>%
  lm(c ~ a + b, data = .) %>%
  coefficients()
```

```
(Intercept)           a           b
  0.5064368   0.1994570   0.3013939
```

You might notice that the coefficients on `a` and `b` are not the same as the coefficients in the `dag09` formulas. That's because the `lm()` technique is not adequate to reveal the coefficients.

The simulated data from DAGs, along with a knowledge of the actual formulas used in the simulation, will help you learn about model building.

⚠ Demonstration: Modeling binomial variables

Keep in mind that this is just a demonstration. You're not expected to master (or even understand) the calculations done in this box.
Recall from the printed version of `dag09` that the value of node `c` was set by a linear combination of `a` and `b` converted into a zero-one, binomial value. The linear modeling, `lm()`, technique is not well tuned to work with binomial data. Instead, another technique called "generalized linear modeling" (implemented by `glm`) is appropriate. When we use the right technique we can, in this case, recover the coefficients in the DAG formula: 2 for `a` and 3 for `b`.

```
sample(dag09, size=10000) %>%
  glm(c ~ a + b, data = ., family="binomial") %>%
  coefficients()
```

```
(Intercept)           a           b
 0.03286538  1.96356232  3.10024718
```

2.4 Samples, summaries, and samples of summaries

A "**sample**" (for the purposes of this course) is a set of rows in a data frame. The "sample size" is the number or rows. "Sampling" is the process of collecting the data to be put into the data frame.

A "**summary of a sample**" is exactly that: a summary, not the sample itself. In Chapter 3 of the *OpenIntro* textbook, you were introduced to a data wrangling operator, `summarize()` and used it to construct some summaries of data frames, that is, "summaries of a sample." For instance, you might decide to summarize the `mtcars` data frame by finding the mean and standard deviation of the `mpg` variable. In the following command, `mtcars` is the sample and the summary is produced by `summarize()`.

```
mtcars %>%
  summarize(m = mean(mpg), s = sd(mpg))
```

```
         m        s
1 20.09062 6.026948
```

It can be very good style for the summary to be contained within a single row. The `dplyr` package for data wrangling is popular because it makes this happen automatically.

When we use DAGs and sometimes even with real data (see Lesson 22), we may want to see whether the summary is always the same or whether it varies from trial to trial.

The following command is one trial of sampling and summarizing data from `dag09`.

```
sample(dag09, size=10000) %>%
  glm(c ~ a + b, data = ., family="binomial") %>%
  coefficients()
```

```
(Intercept)           a           b
-0.01222425  1.99487924  2.93153502
```

To generate a sample of summaries, run many trials of the summary. The `do()` function is handy for this. The following command runs five trials of the `dag09` summary. (Note that the command for the trial is placed inside curly braces.)

```
do(5) * {
  sample(dag09, size=10000) %>%
    glm(c ~ a + b, data = ., family="binomial") %>%
    coefficients()
}
```

```
  Intercept        a        b
1 -0.01315189 2.041331 3.049575
2 -0.01027258 1.987569 3.074509
3  0.02787421 1.934953 3.044001
4 -0.03509540 1.973983 2.969853
5 -0.09887196 1.978861 2.971170
```

Each trial produces one row of the data frame. The five trials are collected together by `do()` into the five rows of a single data frame. Such a data frame can be considered a "**sample of summaries**."

One of the things we will do with a "sample of summaries" is to ... wait for it ... summarize it. For instance, in the following code chunk, a sample of 40 summaries is stored under the name `Trials`. Then we will summarize `Trials`, in this case to see how much the the values of the `a` and `b` coefficients vary from trial to trial.

```
Trials <- do(40) * {
  sample(dag09, size=10000) %>%
    glm(c ~ a + b, data = ., family="binomial") %>%
    coefficients()
}
Trials %>%
  summarize(mean_a = mean(a), spread_a = sd(a),
            mean_b = mean(b), spread_b = sd(b))
```

```
  mean_a   spread_a   mean_b   spread_b
1 2.004514 0.04364563 3.012044 0.06410626
```

The result of summarizing the trials is a "summary of a sample of summaries." This phrase is admittedly awkward, but we will

be using this technique often: summarizing trials, where each trial is a summary of a sample. Often the clue will be the use of `do()`, which repeats trials as many times as you ask.

3 Signal and noise

Prof. Danny Kaplan
November 17, 2022

Imagine being transported back to June 1940 . You and your family might be sitting around a radio receiver, having just switched on set and waited for it to warm up in time to hear a news broadcast. I've selected a newscast for you, recording 103. The recording covers the surrender of the French in the face of the German invasion. Press the play button in the box below and listen.

There are many other recordings on the site which are worth listening to. I'm directing you to #103 as an example of a radio phenomenon: **noise** (or, in slang, "static"). You can clearly make out the spoken words from the recording. But there is also a background sound, something like the sound made by the act of crumpling paper.

Modern radio engineering has more-or-less eliminated noise, mainly by the use of digital technology. (Many of the recordings on the radio archive site have been "cleaned" so the noise is not so evident.) But if you have ever talked to a friend at a sporting event, you have probably had to shout to get your message over the noise of the crowd. At the receiving end, your friend intuitively filters out the noise (unless it is too loud) and recovers your words.

Engineers and others make a distinction between **signal** and noise. The signal is the spoken words of the 1940 broadcast, the noise is the hiss and clicks. You can intuitively separate the signal and the noise in this recording, focusing attention on the signal and ignoring the noise.

Separating signal from noise—or, at least trying to reduce the noise—is a common problem in all sorts of settings. Historically, statistics emerged from a confluence of two needs: i) the need to summarize the resources and activities of countries and **states** (whence comes the "stat" in "statistics") and ii) the need to filter out noise so that the signal becomes clearer.

To illustrate the statistical problem of signal and noise, let's turn to a DAG simulation: `dag01`. Here's a sample from `dag01`:

```
sample(dag01, size=2)
```

```
# A tibble: 2 x 2
      x     y
  <dbl> <dbl>
1 -0.326  2.84
2  0.552  5.04
```

The DAG simulation implements a relationship between `x` and `y`. In statistics, this *relationship* is the signal.

> Look at the 2-row sample from the simulation and make a guess about what the relationship is.

Your guess will be exactly that: a guess. Any of an infinite number of possible relationships could account for the `x` and `y` data. The noise reduction problem of statistics is to make the guess as good as possible. For a sample of size $n = 2$, as good as possible is not very good!

To have a better shot at revealing the relationship hidden by the noise, we need more data, a bigger sample. Here's a sample of size $n = 10$:

```
sample(dag01, size=10)
```

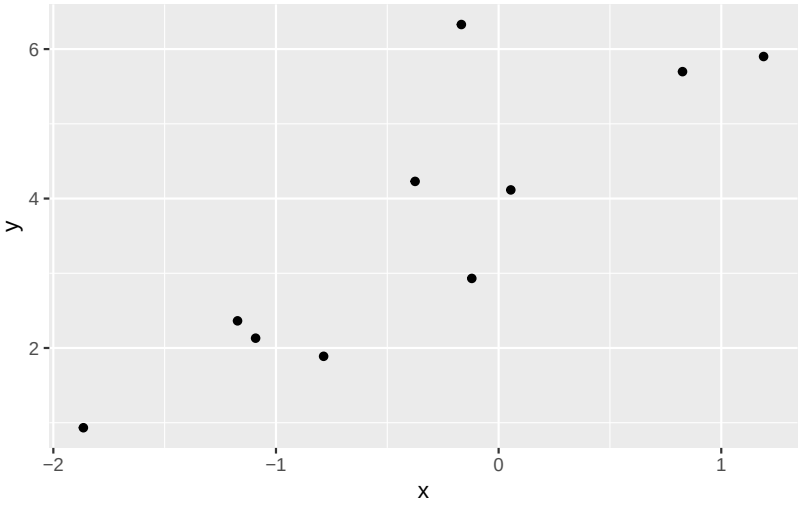
```
# A tibble: 10 x 2
        x     y
    <dbl> <dbl>
```

```
 1 -0.786  1.89
 2  0.0547 4.12
 3 -1.17   2.36
 4 -0.167  6.33
 5 -1.87   0.933
 6 -0.120  2.93
 7  0.826  5.70
 8  1.19   5.90
 9 -1.09   2.13
10 -0.375  4.23
```

Looking carefully at the two rows of data you may be able to see some patterns. `x` is never larger than, say, 2 in magnitude and can be positive or negative. `y` is always positive. And notice that when `x` is negative, the corresponding `y` value is relatively small compared to the `y` values for positive `x`.

With the bigger sample size, $n = 10$ versus $n = 2$, we can make a more informed guess about the relationship between variables `x` and `y`.

Human cognition is not well suited to looking a long columns of numbers. Often, we can make better use of our natural human talents by translating the sample into a graphic, like this:



Collecting more data can make the relationship clearer. Here's a graph of a sample of size $n = 10,000$ with the smaller $n = 10$ sample shown in orange:

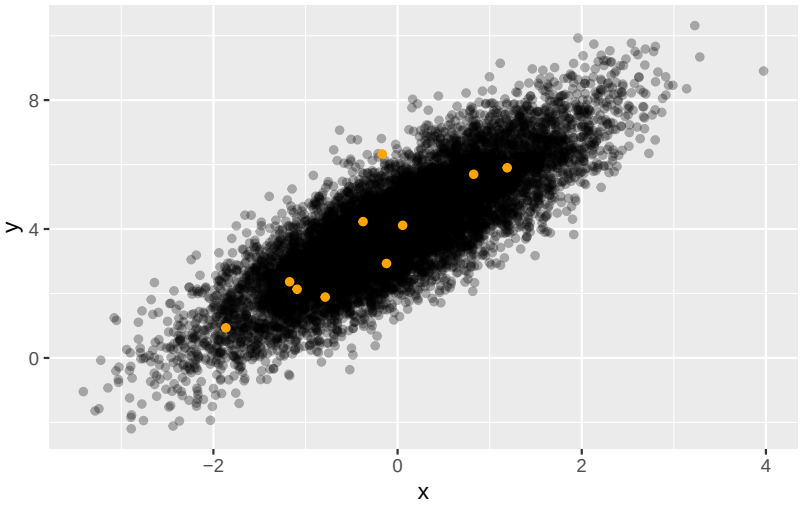


Figure 3.1: With $n = 10,000$ rows, the relationship between `x` and `y` is evident graphically.

There are many ways to describe the relationship between `x` and `y` indicated by the graph of the large sample. For instance, we can see that when `x` is positive, `y` is almost always greater than 4, but for negative `x` the value of `y` tends to be less than 4.

In this course, we prefer to make quantitative summaries of relationships. We do this by fitting models to the data. Here's the relationship that's shown by the original sample of 10:

```
lm(y ~ x, data = original) %>% coefficients() # size 10 sample
```

```
(Intercept)           x
   4.262846    1.741758
```

The coefficients in this model correspond to the mathematical relationship $y = 4.26 + 1.74x$. On it's own, this formula doesn't tell us the extent to which we have filtered out the noise in the simulation.

With more data, say, the larger sample of $n = 10,000$, the relationship becomes more evident:

```
lm(y ~ x, data = larger) %>% coefficients() # size 10 sample
```

```
(Intercept)           x
   4.008928    1.495904
```

Because these data come from a DAG simulation, we can look at the formulas to see exactly what relationship is behind the data:

```
print(dag01)
```

```
[[1]]
x ~ eps()

[[2]]
y ~ 1.5 * x + 4 + eps()

attr(,"class")
[1] "list"      "dagsystem"
```

Comparing the two models to the DAG formula for `y`, we can see that the larger sample produced coefficients that are much closer to the formula than did the smaller sample. Closer, but not exactly the same. Even in the coefficients calculated from the large sample, there is still a legacy of the noise in the original relationship.

The lesson here is simple: *More data can give a better view of the relationships.*

The challenge we face when working with data generated in the real world is that it is not often possible to open the black box that generated the data; all we have is the data! So how can we tell whether the data we have at hand are sufficient for giving a faithful description of the actual relationships?

The general idea is to use the variation within the sample to accomplish two things at once: i. make a description of the relationship, and ii. estimate how much inherited noise there is

in the description. The result of (ii) is important, since it can tell us whether or not our description (i) is good enough for the purpose we seek to serve.

To get started, let's explore how to measure the amount of variation in the data. This can give us an idea of the size of the overall signal+noise, which we will do in the next section of this Lesson. In Lesson 22 we will use DAG simulations to get an idea of how to estimate the amount of inherited noise in the description of the relationship. The DAG simulation is useful because we have access to the internal mechanism of the DAG, so it is easy to see how close the description is to the actual relationship.

In Lesson 23, we will take off the DAG training wheels and learn how to estimate the size of the inherited noise in the description directly from data, without having to open the black box of the mechanism that generated the data. If you think about it, it is an amazing claim that we can estimate how close our data-driven description is to the actual mechanism, without having to know the actual mechanism!

3.1 Measuring variation

We already know the standard way to measure variation in a single variable: the **variance** or, equivalently, the **standard deviation**, which is simply the square root of the variance.

Perhaps you are wondering why there are *two* standard ways to measure variation, when each can be calculated from the other? The variance can be found by squaring the standard deviation, the standard deviation by taking the square root of the variance. Either will do, so why both?

The answer to this question is illustrated by a bit of geometry: the mathematics of right triangles and the corresponding Pythagorean relationship among the sides of the triangle: $A^2 + B^2 = C^2$. The quantities A, B, and C are the lengths of the edges of the right triangle. The quantities A^2, B^2 and C^2 are the lengths-*squared* of those edges. In order to calculate the length of one edge given the lengths of the others, we need

first to square the lengths. Having squared them, we can easily do the calculation of the length-squared of the unknown edge. Then, we take the square root of the length-squared to find the length of the edge.

Lengths are like standard deviations, lengths-squared are like the variance. Where does the right triangle fit in? The overall variation in the response variable is like the hypothenuse of a right triangle. One of the other two edges represents the *noise* in the relationship. The other edge represents the *signal*: the relationship itself. It's easy to measure the overall variation in the response variable. We can also measure the noise, but indirectly. First, we fit a model connecting the response variable to the explanatory variable(s). Then the variation of the **residuals** for that model are the estimate for the noise.

Our first illustration will use data from `dag01`. We will arbitrarily set the sample size to $n = 10,000$. (In Lesson 22, we will look at the impact of sample size on the results.)

```
Dag_data <- sample(dag01, size=10000)
```

Now measure the variation in `x` and `y` in the standard way:

```
Dag_data %>%
  summarize(sx = sd(x), sy = sd(y), vx = var(x), vy = var(y))
```

```
# A tibble: 1 x 4
     sx    sy    vx    vy
  <dbl> <dbl> <dbl> <dbl>
1 0.998  1.81 0.995  3.29
```

The size of the `x` variation is about 1. The size of the `y` variation is about 1.7. (We're using the standard deviation to measure the size of the variation.

Look again at the formulas that compose `dag01`:

```
print(dag01)
```

```
[[1]]
x ~ eps()

[[2]]
y ~ 1.5 * x + 4 + eps()

attr(,"class")
[1] "list"      "dagsystem"
```

From the formula for `x` we can see that `x` comes from a random number generator, `eps()`. The `eps()` generator is designed to generate noise of size 1 by default.

As for `y`, the formula includes two sources of variation:

1. The part of `y` determined by `x`, that is $y = \mathbf{1.5x} + 4 + \text{eps}()$
2. The noise added directly into `y`, that is $y = 1.5\mathbf{x} + 4 + \mathbf{eps}()$

The 4 in the formula doesn't add any *variation* to `y`; it's just a number.

Let's measure variation using the standard deviation: We already know that `eps()` generates variation of size 1. So the amount of variation contributed by the `+ eps()` term in the DAG formula is 1. The remaining variation is contributed by `1.5 * x`. The amount of variation in `x` is 1, coming from the `eps()` in the formula for `x`. A reasonable guess is that `1.5 * x` will have 1.5 times the variation in `x`. So, the variation contributed by the `1.5 * x` component is 1.5. The overall variation in `y` is the sum of the variations contributed by the individual components. This suggests that the variation in `y` should be

$$\underbrace{1}_{\text{from eps()}} + \underbrace{1.5}_{\text{from 1.5 x}} = \underbrace{2.5}_{\text{overall variation in y}} .$$

Simple addition! Unfortunately, the result is wrong. In the previous `summarize()` of the `Dag_data`, we measured the overall variation in `y` as about 1.72.

Let's try again, this time using the *variance* as our measure of variation.

Since `eps()` generates variation whose standard deviation is 1, the variance is simply $1^2 = 1$. The variance of `x` is therefore 1, as is the variance of the `eps()` component of `y`.

What's the variance of `1.5 * x`? It turns out to be $1.5^2 \text{var}(x) = 2.25$. Adding up the variances from the two components gives

$$\text{var}(y) = \underbrace{2.25}_{\text{from 1.5 eps()}} + \underbrace{1}_{\text{from eps()}} = 3.25$$

This result, that the variance of `y` is 3.25, is a close match to what we found in summarizing the `y` data generated by the DAG. And, of course, $\sqrt{3.25} = 1.80$, which is what we found by calculating the standard deviation of the `y` directly from the data.

The lesson here: When adding two sources of variation, the variances of the individual sources add to produce the overall variance of the sum. Just like $A^2 + B^2 = C^2$.

3.2 DAGs from data

In modeling data from `dag01` we could recover the DAGs formula for `y`.

```
sample(dag01, size=10000) %>%
  lm(y ~ x, data = .) %>%
  coefficients()
```

```
(Intercept)           x
   3.996846    1.494939
```

It is wrong to think that from data we can determine the DAG that generated the data. It's only if we know the structure of the data-generation DAG that we can recover the mechanism inside that DAG. But another statistical thinker might claim that what's behind the data is `y` causing `x`. Based on

this assumption, she also can find the mechanism inside her hypothesized DAG:

```
sample(dag01, size=10000) %>%
  lm(x ~ y, data = .) %>%
  coefficients()
```

```
(Intercept)           y
 -1.8485902   0.4635813
```

A DAG is a **hypothesis**, a statement that might or might not be true. DAGs are part of the statistical apparatus for thinking responsibly about **causality**. You use a DAG—or, potentially, multiple DAGs—when the issue of what causes what is relevant to your work.

When there are only two variables involved in the system under consideration—we'll call them X and Y for simplicity—there are only two possible DAGs:

$$X \to Y \quad \text{and} \quad X \leftarrow Y$$

Often, but not always, our understanding of the world allows us to focus on one of these and not the other. Example: Does the rooster crowing cause the sun to rise, or does the rising sun cause the rooster to crow? That's a pretty easy question if you know how things work.

But there are additional DAG possibilities that can account for the relationship between `x` and `y`. For instance, if we introduce another quantity, `c` in between `x` and `y`, four other DAGs need to be considered:

$$X \to C \to Y \quad \text{and} \quad X \leftarrow C \leftarrow Y \quad \text{and} \quad X \leftarrow C \to Y \quad \text{and} \quad X \to C \leftarrow Y$$

Actually, there are many other configurations of DAGs involving three variables. To keep things simple, we'll restrict things to DAGs where X might or might not cause Y, but Y never

causes X. (We don't lose anything from this restriction because you get to make the choice of what real-world variable correspond to X and which one to Y.) Figure 3.2 shows the 10 configurations of 3-variable DAGs where Y doesn't cause X.

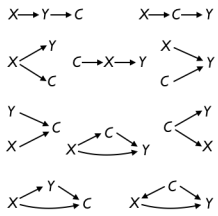


Figure 3.2: Ten DAG configurations involving three variables X, Y, and C and where there is no causal path going from Y to X.

The statistical thinker, with the conceptual tool of DAGs at, can consider multiple possibilities for what might cause what. Sometimes she will be able to discard some of the possibilities based on common sense. (Think: roosters and the sun.) But other times there may be possibilities that she doesn't favor but which nonetheless might be plausible to other people. In Lesson 28 we will explore how each configuration of DAG has implications for which model formulas can or cannot reveal the hypothesized causal mechanism.

4 Sampling variation

Prof. Danny Kaplan
November 17, 2022

There are many sources of noise in data; every variable has its own story, part of which is noise from measurement error, recording blunders, etc. Economists use national statistics, like GDP, even though the definition is arbitrary (a Hurricane can raise GDP!) and early reports are invariably corrected a few months later. Historians go back to original documents, but inevitably many of the documents have been lost or destroyed: a source of noise. Even in elections, where you would think counting is straightforward, the voters' intentions are measured imperfectly due to "hanging chads," "butterfly ballots," broken voting machines, spoiled ballots, and so on.

The statistical thinker is well advised to know about the sources of noise in the system she is studying. Your analysis of data will be better the more you know about how measurements are made and data collected.

> **Note**
>
> The author has on several occasions testified in legal hearings as a statistical expert. In one case, the US Department of Labor had audited the records of a contractor with several hundred employees and high turnover. The records led the Department to bring suit against the contractor for discriminating against Hispanics, and open-and-shut case. How so? The hiring records showed that many Hispanics applied for jobs but none of them was hired.

As the statistical expert, I was asked to review the findings from the Department of Labor. The lawyers thought they were asking me to check the arithmetic. As a statistical thinker, I know that the arithmetic is only part of the story. You also have to investigate the data. Although I had the spreadsheets to work from, I asked to be given the complete files on all applicants and hires the previous year.
Here's what happened. There were indeed many Hispanic applicants, as shown both by the spreadsheet files and the paper job applications. And the spreadsheets showed no hires of Hispanics. But often the data on the job application form wasn't consistent with the data on hires. It turned out that whenever an applicant was hired, the contractor (per regulation) got a report on that person from the state police. But the report returned by the state police had only two available race/ethnicities: white and Black. The personnel office in the contractor filled in the hired-worker spreadsheet based on the state police report. So all the Hispanic applicants who were hired had been transformed into white or Black by the state police. Noise.

4.1 Sampling variation

There is one source of noise that is so common across many settings that every statistical thinker needs to be intimately familiar with. This is called "**sampling variation**."

We've been using "sample" as a near synonym for "data frame." But that's not completely fair. Often, data frames contain a row for each and every "item" of relevance. For instance, the Department of Labor never suggested that the contractor in the previous example had left out some of the applicants. Such a complete enumeration—the inventory records of a merchant, the records kept of student grades by the school registrar—has a technical name: a "**census**." Famously, many countries have a regular census of the population—every 10 years in the US or the UK or China—in which (they try to) reach out to every resident.

But there are many settings where it is unfeasible to collect data in the form of a census. The records will be incomplete and therefore constitute a "**sample**." Sampling is called for when we want to find out about a large group but don't have the resources—time, energy, money—to contact every member of the group. France, in order to collect up-to-date data while staying within a budget, runs a "rolling census" where samples are made at short time intervals. It's estimated that the French rolling census ultimately reaches 70% of the population.

Sometimes, as in quality control in manufacturing, the measurement process is destructive: the item is consumed in the process of measurement. Then, of course, it would be pointless to make a measurement of every single item. A sample will have to do.

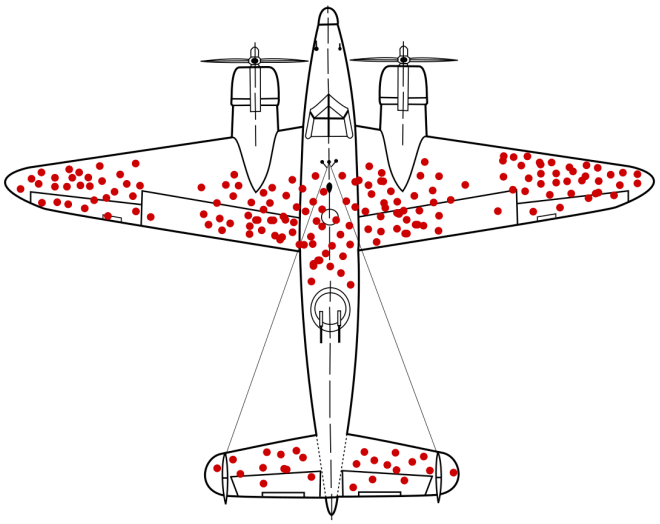
Collecting a reliable sample is usually a lot of work. One idealization is called a "simple random sample" (SRS) where all of the items are available, but only some are selected, at random, to be recorded as data. The work comes from having to assemble all of the items. Making a SRS calls for first assembling a "sampling frame," which is essentially a census. If a census is unfeasible, the construction of a perfect sampling frame is hardly less so.

Professional work, such as the collection of unemployment data, often requires government-level resources and draws on specialized statistical techniques such as stratified sampling and weighting of the results. We won't cover the specialized techniques in this introductory course, even though they are very important in creating representative samples. If you're interested in seeing what's involved, you can get an idea by scrolling through the table of contents of a classic text, William Cochran's *Sampling techniques*

All statistical thinkers, whether expert in sampling techniques or not, should be aware of factors that can bias a sample away from being representative. **Non-response bias** can be significant, even overwhelming, in surveys. In political polls, many (most?) people will not respond to the questions. If this non-response stems from, for example, an expectation that the response will be unpopular, then the poll sample won't adequately reflect unpopular opinions.

Survival bias is an important consideration in many settings. An example is given by the `mosaicData::TenMileRace` data. This records the running times of 8636 participants in a 10-mile road race held in 2005 and includes information about the runner, such as his or her age. You might think that such data could tell you about changes in running performance as people age: the data frame includes runners from age 10 to 87. But a model of running time as a function of age from this data frame is seriously biased. The reason? As people age, casual runners tend to drop out of such races. So the older runners are skewed toward higher performance. (We can see this by taking a different approach to the sample: collecting data over multiple years and tracking individual runners as they age.

> **i** Examples: Returned to base
>
> An inspiring story about dealing with survival bias comes from a World War II study of the damage sustained by bombers due to enemy guns. The sample, by necessity, included only those bombers that survived the mission and returned to base. The shell holes in those surviving bombers were not representative of where shells hit the planes, they were only representative of shell hits that did not prevent the plane from returning. The study report, available here, is a tribute the the work and ingenuity needed to deal with issues such as survival bias. The report itself doesn't contain any diagram showing where shells it the bombers, but a hypothetical diagram on Wikipedia conveys the idea.



The shell holes on the surviving planes were clustered in certain areas. The clustering stems from survivor bias. Planes that were it in the areas, such as the middle of the wings, the cockpit, the engines, and the back of the fusalage did not return to base. Consequently, those shell hits were never recorded.

4.2 Measuring sampling variation

We start the process of learning about sampling variation on the training ground. That is, we'll use simulations from DAGs even though our ultimate goal is to work with real data. DAGs are a convenient training tool because the data generated is always a simple random sample and we can generate any number of samples of any size we wish. In the spirit of starting simply, we'll return to `dag01` which, you may remember, has the structure `x` $\longrightarrow$ `y` and the causal formula `y ~ 4 + 1.5 * x + eps(x)`.

It's crucial to remember that sampling variation is not about the row-to-row variation in a single sample, it is about the variation in the summary from one sample to another. So our initial process for exploring sampling variation will be to carry out many trials, each of which is a summary of a sample.

::: {.callout-warning} ## Samples and specimens

To illustrate, here is one trial using a sample of size $n = 25$. There are many ways to summarize a sample, here we will use `y ~ 1`.

```
Sample <- sample(dag01, size=25)
Sample %>%
  lm(y ~ 1, data = .) %>%
  coefficients()
```

```
(Intercept)
   3.267232
```

We can't see sampling variation directly in the above result because there is only one trial. To see sampling variation directly, we need to run *many* trials. In each trial, a new sample (of size $n = 25$ is taken and summarized.)

```
Trials <- do(100) * {
  Sample <- sample(dag01, size=25)
  Sample %>%
    lm(y ~ 1, data = .) %>%
    coefficients()
}
Trials
```

```
    Intercept
1    3.889744
2    4.431278
3    4.164084
4    3.481357
5    4.388745
6    4.501441
7    4.282840
8    3.800674
9    4.428752
10   4.423763
11   4.216165
```

```
12   3.825953
13   3.726697
14   3.899564
15   3.846793
16   4.452258
17   4.450224
18   3.827420
19   3.855896
20   3.878339
21   3.779224
22   4.209077
23   3.653399
24   3.822603
25   3.679266
26   4.023805
27   3.749925
28   4.176551
29   4.362977
30   3.465763
31   3.880549
32   4.058939
33   4.701485
34   3.451283
35   3.794117
36   3.761613
37   3.704658
38   3.544211
39   3.612334
40   4.178411
41   3.677431
42   4.209276
43   3.576305
44   4.174389
45   3.936319
46   3.745918
47   3.807143
48   3.346535
49   2.455597
50   3.580130
51   3.939518
52   3.893458
```

53	4.235055
54	4.225746
55	3.868170
56	3.469867
57	3.636248
58	3.968085
59	3.744367
60	4.051956
61	4.046258
62	3.282494
63	4.423492
64	3.108257
65	4.337208
66	3.827530
67	4.002049
68	3.698438
69	4.210539
70	3.593579
71	3.543198
72	4.109773
73	3.952732
74	3.587537
75	3.514824
76	4.308436
77	3.529206
78	4.164861
79	4.018987
80	3.984119
81	3.854006
82	4.819582
83	4.121940
84	3.997939
85	3.583546
86	3.501273
87	4.651216
88	4.062853
89	4.155898
90	4.379576
91	4.142334
92	4.185428
93	3.933915

```
94   4.688621
95   3.236602
96   3.664758
97   4.245515
98   3.158921
99   3.442100
100  4.142309
```

`Trials` is a *sample of summaries*. In `Trials`, the sampling variation can indeed be seen in the row-to-row variation in the data frame, but only because the data frame is a summary of samples. Since it is hard to read columns of numbers, we will *summarize* the variation in the *sample of summaries*.

As always, our standard measure of variation is the standard deviation (or, equivalently, variance):

```
Trials %>%
  summarize(sIntercept = sd(Intercept))
```

```
  sIntercept
1  0.3853414
```

This quantity, which is the standard deviation of a sample of summaries, has a technical name in statistics: the **standard error**. The words **standard error** should properly be followed by a description of the summary and the size of the individual samples involved. Here it would be, "0.348 is the standard error of the Intercept coefficient from a sample of size 25.

The standard error is an ordinary standard deviation, but in a particular context: the standard deviation of a sample of summaries. This can be confusing, since "error" and "deviation" are somewhat synonomous in everyday language. It can be hard to remember when to use "error" and when to use "deviation." Fortunately, it's more common to use another way to present the information about sampling variation.

4.3 The SE depends on sample size

We found an SE of 0.348 on the Intercept in a sample of size $n = 25$. Does the SE depend on the sample size. We can find out by trying it for several different sample sizes, say, 25, 100, 400, 1600, 6400, 25600, 102400. We picked these particular numbers to be multiples of 4 times the previous sample size.

Here's the SE for a sample of size 400:

```
Trials <- do(100) * {
  Sample <- sample(dag01, size=25)
  Sample %>%
    lm(y ~ 1, data = .) %>%
    coefficients()
}
Trials %>% summarize(se = sd(Intercept))
```

```
        se
1 0.3500593
```

You can try it yourself for the other sample sizes. Here's what we got, running 1000 trials in each instance:

```
SE  %>% knitr::kable()
```

n	se
25	0.3600
100	0.1900
400	0.0910
1600	0.0430
6400	0.0230
25600	0.0110
102400	0.0056

There's a pattern here. Every time we double n, the standard error goes down by a factor of 2, that is, $\sqrt{4}$. (The pattern isn't exact because there is sampling variation in the trials themselves.)

Lesson: The standard error gets smaller the larger the sample size. For a sample size of n, the SE will be proportional to $1/\sqrt{n}$.

4.4 The confidence interval

The "confidence interval" is a more user-friendly format for describing the amount of sampling variation. As an interval, it commonly written either as [lower, upper] or center±half-width. These styles are completely equivalent and either style can be used. The preferred style can depend on the field or the journal in which a report is being published. Some journals like a different style, center (half-width).

> **i** Technical vocabulary
>
> There is a technical name for the half-width: the "**margin of error**." We will leave the confidence interval calculation to software, so we won't have much need to refer to the margin of error, but it is a term commonly used by statisticians and scientists.

The margin of error is defined to be twice the SE. A lot of early statistical theory was given over to defining "twice." For our purposes, twice means "multiply by 2." Some people prefer the theoretically more precise "multiply by 1.96" which is appropriate for very large sample sizes. For small sample sizes "twice" is larger than 1.96 and depends on how many model coefficients there are. For instance, consider the simplest model `y ~ 1`. There is one coefficient and for a sample size of $n = 20$ twice would be 2.09 while a for a sample size of $n = 5$ "twice" would be 2.8.

> **⚠** Demonstration: Confidence interval on the Intercept coefficient for $n = 25$
>
> The following command computes the confidence interval (that is, the two numbers [`lower`,`upper`]) for the trials we ran on samples of size $n = 25$ from `dag01` and summarized

by the intercept coefficient from `y ~ 1`. We show this just to make clear what that the margin of error is twice the standard error.

```
Trials %>%
  summarize(m=mean(Intercept), se=sd(Intercept)) %>%
  mutate(lower = m - 2*se, upper = m + 2*se)
```

```
        m         se    lower    upper
1 3.95587 0.3500593 3.255751 4.655988
```

Notice that we have used 2 for "twice." But best to leave the detailed calculations to the software.

Statistical software is written to use the correct value of "twice" for any given sample size and number of coefficients. But for everyday purposes, and samples larger than, say, $n = 10$, "twice" is roughly 2.

In calculations, finding the half-width of the confidence interval requires first finding the standard error, then multiplying by "twice." In practice, it's far easier to use software. In R, the `confint()` function reports the confidence interval for model coefficients:

```
Hill_racing %>%
  lm(time ~ distance + climb, data=.) %>%
  confint()
```

```
                2.5 %      97.5 %
(Intercept) -533.432471 -406.521402
distance     246.387096  261.229494
climb          2.493307    2.726209
```

Demonstration: How many digits?

In the calculation of confidence intervals on the model `time ~ distance + climb`, the results were reported to many digits. Such a report is appropriate for whatever

further calculations might need to be done on the results, but it is usually not appropriate for a human reader.
To know how many digits are worth reporting to humans, you can look at the standard error. The standard error is a part of a different kind of summary of a model: the "regression report." We won't need to look at regression reports until the end of the course. We show one here just to make the point about how many digits are worth reporting to humans.
Here's the regression report on the `Hill_racing` model

```
Hill_racing %>%
  lm(time ~ distance + climb, data=.) %>%
  regression_summary()
```

```
# A tibble: 3 x 5
  term        estimate std.error statistic   p.value
  <chr>          <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept)  -470.     32.4        -14.5 9.92e- 46
2 distance      254.      3.78        67.1 0
3 climb           2.61    0.0594      43.9 4.08e-304
```

The "standard error" for each coefficient is reported in the column labelled `std.error`.
For the human reader, only the first two significant digits of the standard error are worth reporting. In this case, that is 32 for the Intercept, 3.8 for the distance coefficient, and 0.059 for the climb coefficient. The confidence interval will be the coefficient itself (column labelled `estimate`) plus-or-minus "twice" the `std.error`. The report of the confidence interval (for a human reader) should be rounded to the place of the first two significant digits of the standard error.
For example, the confidence interval on the distance coefficient will be $253.808295 \pm 2 \times 3.78433220$. Keep only the digits that come before the first two significant digits of the SE, so the reported interval can be 253.8 ± 3.8.

5 Estimating sampling variation from a single sample

Prof. Danny Kaplan
November 17, 2022

Lesson 21 introduced the idea of separating data into separate components: signal and noise. The *signal* is a summary of the data that tells us something we want to know. Often, the signal will be one or more coefficients from a regression report, but it might be something as simple as the mean or median or standard deviation of a variable in a data frame.

The *noise* comes into the data from any of a variety of sources: e.g. error in measurement or a data-entry blunder. Another source of noise is omnipresent (except in a perfect census): sampling variation. The idea behind sampling variation is that the particular data at hand is just one sample and is contingent on the time and situation in which the data were collected. Data collection at a different time or situation would presumably be somewhat different.

The Greek philosopher Heraclitus (c. 500 BC) said, "You can't step into the same river twice." Each time you step into a river, you might be at the same place on the bank but the water around you will be different. A data sample is like collecting water from a river or lake using a dipper. Imagine ten people standing side by side on the shore of a lake, each person dipping into the water acquire a specimen and making one or more measurements from the specimen, for instance the temperature, pH, and bacteria count. Each person collects a sample—that

is, a series of specimens. These might be taken one right after the other or by some protocol, say a weekly tracking of lake conditions over time.

The ten people are each doing the same thing in approximately the same place and same time, but each person's sample—say the collection of 52 weekly specimens over the course of a year—will be different. Perhaps only a little bit different. That sample-to-sample variation will be noise.

If the ten people were fishing, each specimen would be the result of one cast of the rod. Typically this is just an empty hook, lake weeds, or a stick, but sometimes it will be a fish. At the end of the fishing day, each fisherman will have a sample. With fishing luck (or skill), some of the specimens in the sample will be fish. Presumably, the luck (or skill) of the fishermen will differ one from the other. That's the noise of sampling variation. To fishermen the question of interest, the signal they want to measure, might be, "How good is the fishing today?" Each answers that question by looking at how many fish he or she caught. The ten fishermen's catch will differ: sampling variation. Consequently, each fisherman's answer will be contaminated with some noise. The fisherman's summary (the count of fish caught) will have some noise stemming from sampling variation.

In Lesson 21, to gain some feeling for sampling variation, we repeated trials over and over gain. Each trial consisted of collecting a sample (that is, multiple specimens) and summarizing it. The individual trial is a summary of a sample. We then summarized the whole set of trials with the standard deviation. This is how we quantified sampling variation.

Now it is time to take off the DAG training wheels and measure sampling variation from actual data, from a *single* sample. This sounds like an impossible task. Sampling variation is about the variation in summaries *between* samples. With a *single* sample there is no "between" to be had.

What we will need to estimate sampling variation from a *single* sample is a way to simulate drawing new samples from the single sample.

Authors of statistics books tend to choose examples based on their own interests rather than their students', so in this sec-

tion let's look at athletic performance as people age. The `TenMileRace` data are readily to hand, so we'll look at net race time (from start line to finish line) as a function of age. We'll limit the study to people over 40.

```
TenMileRace %>% filter(age > 40) %>%
  lm(net ~ age, data = .) %>% coefficients()
```

```
(Intercept)         age
 4278.21279    28.13517
```

The units of `net` are seconds, the units of `age` are years (as conventional). The model coefficient on `age` tells us how the `net` time changes for each additional year of `age`. This summary of the data tells us that the time to run the race gets longer by about 28 seconds per year. So a 45-year-old runner who completed this year's 10-mile race in 3900 seconds (that's about 9.2 mph, a pretty good pace!) might expect that, in ten years, when she is 55 years old her time will be longer by 280 seconds.

It would be asinine to report the ten-year change as 281.3517 seconds. The runner's time ten years from now will be influenced by the weather, crowding, the course conditions, whether she finds a good pace runner, the training regime, improvements in shoe technology, injuries, illnesses, etc. There's little or nothing we can say from the `TenMileRace` data about such factors.

But we should not forget sampling variation. `TenMileRace` has 2898 includes 2898. The way the data was collected (radio-frequency interogation of a dongle on the runner's shoe) suggests that the data is a census of finishers, but really it is a sample of the kind of people who run such races. People might have been interested in running but had a schedule conflict, or lived too far away, or missed their train into the start line in the city. So we'll treat the data as a sample.

This sample of 2898 runners is the only sample we have, so there is no option to compare multiple samples in order to look at sampling variation. That's no excuse for not making an

interval estimate on the age coefficient. We have to use some ingenuity.

Here's an idea. Instead of using samples of size 2898, let's use sub-samples one-tenth the size: $n = 290$. We'll select the subsamples at random:

```
Over40 <- TenMileRace %>% filter(age > 40)
lm(time ~ age, data = Over40 %>% sample(size=290)) %>% coefficients()
```

```
(Intercept)         age
  4429.0069     28.4424
```

```
lm(time ~ age, data = Over40 %>% sample(size=290)) %>% coefficients()
```

```
(Intercept)         age
 4129.74194    34.21546
```

The age coefficients differ one from the other by about 0.5 seconds. Better, let's select many subsamples of size 1449 at random, and find the age coefficient for each of them. We will run 100 trials

```
# a sample of summaries
Trials <- do(1000) * {
  lm(time ~ age, data = sample(Over40, size=290)) %>% coefficients()
}
# a summary of the sample of summaries
Trials %>%
  summarize(se = sd(age))
```

```
       se
1 9.042024
```

We used the name `se` for the summary of samples of summaries because what we have calculated is the standard error of the age coefficient in a sample of size $n = 290$.

In Lesson 22 we saw that the standard error is proportional to $1/\sqrt{n}$, where n is the sample size. From the subsamples, know that the SE for $n = 290$ is about 9.0 seconds. This tells us that the SE for the full $n = 2898$ samples would be about $9.0\frac{\sqrt{290}}{\sqrt{2898}} = 2.85$.

So the interval summary of the `age` coefficient—the so-called "confidence interval" is

$$\underbrace{28.1}_{\text{age coef.}} \pm 2\times \underbrace{2.85}_{\text{standard error}} = 28.1 \pm \underbrace{5.6}_{\text{margin of error}} \quad \text{or, equivalently, 22.6 to 33.6}$$

5.1 Bootstrapping

There is a trick to generating a random subsample of a data frame with the same n as the data frame: draw the subsample from the original sample **with replacement**. An example will suffice to show what the "with replacement" does:

```
example <- c(1,2,3,4,5)
# without replacement
sample(example)
```

```
[1] 1 4 3 5 2
```

```
# now, with replacement
sample(example, replace=TRUE)
```

```
[1] 2 4 3 3 5
```

```
sample(example, replace=TRUE)
```

```
[1] 3 5 4 4 4
```

```
sample(example, replace=TRUE)
```

```
[1] 1 1 2 2 3
```

```
sample(example, replace=TRUE)
```

```
[1] 4 3 1 4 5
```

The "with replacement" leads to the possibility that some of the values will appear two or more times in the subsample and others of the values will be left out.

The calculation of the SE using sampling with replacement looks like this: rset.seed(207)‘

```
# run many trials
Trials <- do(1000) * {
  lm(time ~ age, data = sample(Over40, replace=TRUE)) %>%
       coefficients()
}
# summarize the trials to find the SE
Trials %>% summarize(se = sd(age))
```

```
       se
1 2.948786
```

```
# or let the computer do the work of converting to a confidence interval
Trials %>% confint()
```

```
  name    lower   upper level     method estimate
1  age 21.58031 33.3163  0.95 percentile  27.2782
```

This method is called "**bootstrapping** a confidence interval." The same word, "bootstrapping" is used to describe how a computer turns itself on. It comes from the idea of a person raising herself from the ground by pulling upward on her own boots. An impossible task. And a suitable metaphor for generating many samples from a single sample.

5.2 Using the residuals

Lesson 21 pointed to the idea of data consisting of two parts: signal plus noise. In Lesson 22 and thusfar in this lesson, we've tried to estimate the signal by summarizing the data. But we still had to account for sampling variation, which we did by generating many subsamples.

Another route to measuring sampling variation takes more literally the division of data into signal and noise. The idea is still to estimate the signal by a regression model summary. But now, we take the **residuals** from the model as the evidence for how much noise there is. We quantify the variation in the residuals in the same way that we have always done: their standard deviation. This quantity, the standard deviation of the residuals from a model, has its own technical name: the "residual standard error." For some types of models, it's possible to push the residual standard error through the model-fitting apparatus in order to construct standard errors and confidence intervals. The mathematics of this is a matter for specialists, but computers handle the calculations well.

The `confint()` function knows how to take `lm()`-fitted models and translate the residuals into confidence intervals. Like this:

```
lm(time ~ age, data = Over40) %>% confint()
```

```
                2.5 %     97.5 %
(Intercept) 4239.50891 4804.89396
age           21.58833   32.96807
```

Contemporary statistics uses many different model types depending on the situation. In this course we will only use two: linear models and generalized linear models. The field of machine learning has introduced many other kinds of models, often with evocative names like "regression trees," "random forests," and "vector support machines." Usually there is not a way to push the residual standard error through such calculations. But when a confidence interval is needed, the boostrapping method can always be used.

5.3 Margin of error

```
one_trial <- function(n=2) {
  vals <- rnorm(n)
  tibble(m = mean(vals), s = sd(vals))
}
```

The confidence interval from each trial will be $m \pm \beta s$, where β is a number yet to be determined. How to do so, we want to select β so that, across all trials, 95% will include the mean of the distribution from which the data values were drawn.

```
# vary beta until 95% of the trials have a left value smaller than zero.
n <- 10000
beta <- 0.02
Trials <- do(1000) * one_trial(n=n) %>%
  mutate(left = m - beta*s, right = m + beta*s)
Trials %>%
  summarize(coverage = sum(sign(left*right) < 0)/n())
```

```
# A tibble: 1 x 1
  coverage
     <dbl>
1    0.967
```

For sample size $n = 10$, β needs to be 0.72, while for a sample size $n = 100$, β needs to be 0.20. For $n = 1000$, the multiplier needs to be 0.062, and so on. For $n = 10000$, the multiplier needs to be 0.02

n	β	$t = \beta/\sqrt{n}$
10	0.72	2.26
15	0.55	2.14
20	0.47	2.09
50	0.28	2.01
100	0.20	1.98
500	0.088	1.96

n	β	$t = \beta/\sqrt{n}$
1000	0.062	1.96
10000	0.20	1.96

Notice that as n gets bigger, the size of β to cover 95% of the trials gets smaller. More than a century ago, it was known that the multiplier for any sample size n is effectively $2/\sqrt{n}$. Consequently, the confidence interval for the mean of n values is approximately

$$\texttt{CI} = \texttt{mean(x)} \pm \underbrace{\frac{2}{\sqrt{n}}\texttt{sd(x)}}_{\text{margin of error}}$$

The quantity following the $\pm$ is called the "**margin of error**." Because of the $\pm$, he overall length of the confidence interval is twice the margin of error.

It's much easier to remember $2/\sqrt{n}$ than a list of β values that change from one n to the next. Another ubiquitous memory aid involves another technical term, the **standard error**. This involves a simple re-arrangement of the equation for the confidence interval:

$$\texttt{CI} = \texttt{mean(x)} \pm 2 \underbrace{\frac{\texttt{sd(x)}}{\sqrt{n}}}_{\text{standard error}}$$

It's standard in statistical software to report the standard error of a coefficient. Usually abbreviated `se` or `std.error` or something similar. The software is doing the divide-by-$\sqrt{n}$ for you, so all you need to construct the margin of error is multiply the standard error by 2. That's convenient, but it comes at the cost of yet another use of the words "standard" and "error," which can be confusing.

Here's an example of a typical software output summarizing a model in the format called a "**regression report**." Here's an example, looking at the fuel economy of cars (`mpg`) as a function the car's weight (`wt`) and horsepower (`hp`).

```
lm(mpg ~ wt + hp, data = mtcars) %>%
  regression_summary()
```

```
# A tibble: 3 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  37.2      1.60        23.3  2.57e-20
2 wt           -3.88     0.633       -6.13 1.12e- 6
3 hp           -0.0318   0.00903     -3.52 1.45e- 3
```

According to this report, each additional 1000 lbs of weight decreases fuel economy by an estimated 3.9 miles per gallon. But since the model is based on a sample of data, it's important to report the *precision* of that number in the face of sampling variation. The confidence interval is the standard format for that precision. It will be the estimate plus-or-minus two times the standard error, that is: $-3.88 \pm 2 \times 0.633$, that is, -5.15 to -2.61 mpg per 1000 lbs. Similarly, each addition horsepower (`hp`) lowers fuel economy by $-0.032 \pm 2 \times 0.009$, that is, -0.05 to 0.013 mpg per horsepower.

Even more convenient is to calculate the confidence interval with `confint()` which handles all the computations, including the ones for tiny n described in **?@sec-tiny-n**.

> **i** How many digits?
>
> Notice that the estimate of the `wt` coefficient in the above regression report is -3.87783074. That seems like an awful lot of digits to report when the confidence interval is -5.15 to -2.61. Or, rather, an awful lot of digits for the human reader.
> It is of course easy for the human to ignore the last several digits of the number. This makes reading more reliable; there are not as many digits to confuse. Even worse, the many digits suggest a level of precision that is belied by the width of the confidence interval. (When the number is going to be part of a continuing computation, that is, the "reader" is a computer, mis-interpretion or faulty reading is not an issue, which is why the software calculates so

many digits .)
So how many digits ought to be reported for a human reader? There is an easy procedure to determine this.

1. Look at the *standard error* in the regression report and multiply by 2 to get the **margin** of error. For example, for the `hp` coefficient, the margin of error is $2 \times 0.63273349 = 1.265467$.

2. It is always the case that no more than two digits of the margin of error have any meaning. (Even the second digit would suffer sampling variation.) So round the margin of error to two digits, that is 1.3 for the `hp` standard error.

3. Notice the location of the second digit of the rounded standard error. For `hp`, the second digit is 3 and it's located in the one-tenths place. Round the coefficient to this place. So, the `hp` coefficient -3.87783074 will round to -3.9.

4. The confidence interval, formatted for the human reader, will be the rounded coefficient plus-or-minus the rounded standard error. For `hp`, the confidence interval will be -3.9 ± 1.3 or -5.2 to -2.6.

5.4 Tiny n (optional)

When you have a very small sample size—say, $n = 2$—the values may coincidentally be very close together. Around 1907, William Gosset, a scientist at Guinness, discovered that such coincidences force β to be much larger than $2/\sqrt{n}$ in order to produce confidence intervals that cover the mean of the data-generating process. Gosset's particular interest was in making sense of Guinness's standard testing protocols, which involve averaging the results from three small batches of beer ingredients. Contacting the leading statisticians of the day, Gosset was told that such small n is "brewing, not statistics." Nonetheless, Gosset had to work within Guinness's testing protocols, which were

indeed brewing but still needed statistical interpretation.

Gosset carried out trials by hand, a large number of measurements from a study of criminals' hand sizes. (They did this kind of thing in 1900.) Each measurement was written on a card. A trial consisted of drawing n cards from the deck and calculating the mean and standard deviation of the measurements. Using computers, we can simulate the calculation of results from a Gosset-like trials using a simple function that calculates the mean and standard deviation of data from a Gaussian distribution.

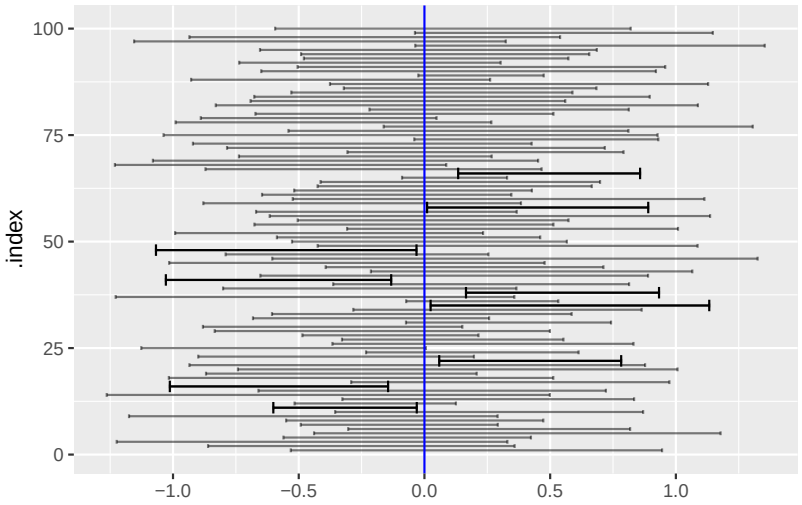
```
one_trial <- function(n=2) {
  vals <- rnorm(n)
  tibble(m = mean(vals), s = sd(vals))
}
```

We can pick a small n and running many trials using a candidate value for "twice."

::: {.callout-warning} ## IN DRAFT

CONVERT the `beta` to something named `twice`.

```
n=10
beta <- 2 / sqrt(n)
Trials <- do(100) * one_trial(n=n) %>%
  mutate(left = m - beta*s, right = m + beta*s)
gf_errorbarh(.index ~ left + right, data = Trials, alpha=0.5) %>%
  gf_errorbarh(.index ~ left + right,
               data = Trials %>% filter(left > 0 | right < 0)) %>%
  gf_vline(xintercept = ~ 0, color="blue", inherit=FALSE)
```



Gosset effectively tabulated the β multipliers

n	β	$t = \beta/\sqrt{n}$
2	8.98	12.7
3	2.48	4.30
4	1.59	3.18
5	1.24	2.78
6	1.04	2.57
7	0.92	2.44
⋮		
10	0.72	2.26
15	0.55	2.14
20	0.47	2.09
50	0.28	2.01
100	0.20	1.98
500	0.088	1.96
1000	0.062	1.96

You can see that for n bigger than 10 or 20, the t multiplier is 2. But for very small n, the t-multiplier can be considerably larger.

You can see the wisdom of brewers here. They made tests by averaging measurements from three small batches of beer. If

they had used only two batches, the confidence interval would be almost three times larger than for $n = 3$, making it very hard to conclude anything about whether the tests show the ingredients to be within the quality-control standards.

Gosset's work was published under the pseudonym "Student," since Guinness forbade employees to publish under their own names. Statisticians, recognizing the value of the work (and knowing the name behind the pseudonym), came to use the name t, perhaps because tea was considered more refined than "beer." In many statistics texts, you will see the phrase "Student t" to refer to how Gosset's work is used.

6 Effect size

Prof. Danny Kaplan
November 17, 2022

You now have a substantial toolbox for for summarizing data in ways that support statistical thinnking. Time to move to the next step: extracting actionable information from such summarizing. Why the word "actionable" in the previous sentence? Because much of the time the goal of summarizing data is to guide **decision making**. The setting is that you or your organization have to make a decision: administer a medicine, change a budget, raise or lower a price, respond to an evolving situation, and so on. Decisions ought to be made on an informed basis. Often, the information needed is hidden in tables of data. The statistical thinker knows how to extract information in a form that is as useful as possible to the decision maker.

Setting for decisions vary widely, but a useful simplification splits support for decision making into two broad categories.

1. **Making a prediction** for an individual choice. The need for predictions arises in both mundane and in critical settings. For instance, an airline needs to set prices. They want to maximize revenue. Higher prices will bring in more money per seat, but the seats may not be filled. To make the decision, the airline needs a prediction about what the demand will be for those seats, which may vary based on day of the week, time of day, time of year, origin and destination of the flight, and so on. Another example: Merchants and social media sites have to make choices about what products or posts to display to a viewer. Merchants have many products, social media has many news feeds, tweets, blog entries to choose from. They want

to predict which ones are most likely to cause you to respond, either by buying a particular product or watching a video, "news" report, and so on.

Less mundane: A patient comes to an urgent-care clinic with symptoms. A decision needs to be made about what disease or illness the patient has in order to guide choices of tests and, in turn, possible treatment. The inputs to the prediction are the symptoms—neck stiffness, a tremor, and so on—as well as facts about the person—age, sex, occupation, etc. The output of the prediction will assign a probability to each of medical conditions that could cause the symptom. As new tests or measurements are done—temperature, blood pressure, white-blood-cell count, blood oxygenation, and others—they become new inputs for the prediction and the probabilities change accordingly. The television drama *House* provides in every episode an example of such evolving predictions, which clinicians call "differential diagnosis." The word "prediction" suggests the future, but many predictions have to do with the current or past state that is as yet unknown to greater or lesser extent. Synonyms for "prediction" include "classification" (Lessons 34 and 35), "conjecture", "guess", "bet", …. The phrase "informed guess" points to the idea: using information to support decision making.

2. **Intervening** in a system. Such interventions occur on both grand scales and small: changes in government policies such as funding for pre-school education or subsidies for renewable energy, closing a road to redirect traffic or opening a new highway or bus line, changing the minimum wage, etc. Before making such interventions, it is wise to know what the consequences are likely to be. Figuring this out often requires understanding how the system works: what causes what. Without knowing this, how can you antipate the influence of the intervention on other components of the system? Also, interventions often affect many individuals: influencing the overall trend of the effect across individuals might be the goal, as opposed to a prediction for each individual affected.

This lesson is focuses on two ideas that are useful for building and summarizing models of a system for the purposes of *intervening* in that system: effect size and interaction. We will need

some additional concepts and tools in order to bring causality into the picture. This will have to wait until Lessons 28 through 31.

In an intervention you change something about the world. That might be the budget for a program, the dose of a medicine, the fuel input to an engine. The thing you change is the input. In response, something else in the world changes: reading ability of students, the patient's seratonin levels (a neurotransmitter), the power output from the engine. The thing that changes in response to the change in input is called the "output." Systems such as education, mental state, or aircraft have many components. The context in which the modeler works dictates which of these components ought to be considered the input and which the output. Usually the input is something that you can directly change; the output is something that changes in response.

The **effect size** is merely a statement of the amount of change in the output with respect to the input. There are two fundamental types of *inputs*, just as there are two fundamental types of variables:

- categorical: e.g., whether or not a person smokes.
- quantitative: e.g., how many cigarettes per day a person smokes

Similarly, there are two fundamental types of *outputs*: categorical or quantitative.

- categorical: e.g. whether the person develops cancer
- quantitative: e.g. the lung capacity of the person

How you properly describe an effect size depends on types of both the input and the output.

input	output	effect size
categorical	quantitative	the **amount* by which the output changes when the input changes category

input	output	effect size
quantitative	quantitative	the **rate** of change in the output with respect to the input. Calculus students will recognize this rate as the partial derivative of the output with respect to the input.
categorical	categorical	the *probability* of being in each of the output categories when the input category is changed
quantitative	categorical	the *rate of probability* of being in each of the output categories per unit of change in the input.

Terms like "rate of probability" can be confusingly abstract. It helps to have some examples in mind to keep your thinking clear.

Examples:

- System: an automobile
  - Selected input: Gallons of gasoline put in a car's tank. Quantitative.
  - Selected output: How far the car can be driven.
  - Effect size will be a *rate*: miles per gallon.
- System: an autombile
  - Selected input: Whether to use a fuel additive the promises high fuel efficiency. Categorical.
  - Selected output: Money spent on fuel (or, perhaps, tons of CO_2 emitted). Quantitative.
  - Effect size is an *amount*: Dollars spent (or, tons of CO_2 emitted)
- System:
  - input categorical

  - output quantitative

- System:
  - input categorical
  - output quantitative

6.1 Calculating an effect size

So long as you keep track of which of the four combinations of input and out are applicable to your case, calculating an effect size is easy. You evaluate the model at two values for the input then collect the two corresponding output values. For instance, you can use the `model_eval()` function. It takes as arguments the model whose effect size you're interested in and, optionally, values for some or all of the inputs.

```
Mod <- lm(mpg ~ hp, data=mtcars)
model_eval(Mod, hp=c(100, 150))
```

```
   hp   .output     .lwr     .upr
1 100 23.27603 15.20660 31.34547
2 150 19.86462 11.85278 27.87645
```

The column labeled `.output` shows the model output for the corresponding input values for `hp`. Here, both the input and the output are quantitative, so the effect size will be a ratio: change in output divided by change in input. In this case:

$$\text{effect size: } \frac{23.28 - 19.86}{100 - 150} = -0.0684$$

It is wise to pay attention to the *units* of the effect size. Here, the output is `mpg`, which has units miles-per-gallon. The input has units horsepower, so the units of the effect size are miles gallon^{-1} horsepower^{-1}. Admittedly, that's a mouthful of units. But it tells us something simple: A car with 100 additional horsepower will get worse fuel economy, down by 6.8 miles per gallon.

Notice that the report from `model_eval()` has additional columns: `.lwr` and `.upr`. That s a glue that it is giving both a single-number, "point" estimate (`.output.`) and a two number interval estimate. We'll talk about the meaning of the interval in the following section and in Lesson 26.

⚠ Is horsepower the cause?

It might seem from the negative sign on the effect size of engine horsepower on fuel economy that a more powerful engine is not as efficient than a less powerful engine at moving the car a given number of miles. That's a reasonable conclusion. But the statistical thinker always keeps in mind other possibilities. For instance, another factor in fuel economy is the overall weight of the vehicle. A van designed to haul many passengers weighs more than a 2-passenger sporty vehicle. The van needs more horsepower because it is accelerating more weight.
?@fig-four-hp-mpg-dags shows four DAGs, each of which describe a plausible scenario.

⚠ Warning

Manually insert the four part figure as a single png

In DAG A, the vehicle's design weight determines that an engine with high horsepower will be part of the design. The weight is also responsible for the lower fuel economy. The other DAGs describe other scenarios. In DAG C, for instance, the car designers decided to build a muscle car and put in a big engine. The engine itself adds to the vehicle's weight, and the higher weight determines lower miles per gallon. DAG D expresses a slightly different belief: again the choice to build a muscle car (high `hp`) influences the weight. But in DAG B, the big engine also directly influences the fuel economy, perhaps because the fuel-to-air ratio of the car, in normal use, is not optimal. As we will see in Lesson 28, to reveal the direct causal link between engine power and fuel economy requires different choices for the model formula depending on which DAG

you think might be relevant.

⚠ Example for LCs: Price of book versus its page count.

Another example: Are longer books more expensive? Intuition suggests so, because more editing, paper, printing and shipping goes into making a longer book. We have some data that might be informative, `moderndive::amazon_books`. We can build a model of, say, `list_price` versus `num_pages`. To look at the effect size, let's compare a 200-page book to a 400-page book.

```
Mod <- lm(list_price ~ num_pages, data = moderndive::amazon_books)
model_eval(Mod, num_pages = c(200, 400))
```

```
  num_pages  .output       .lwr     .upr
1       200 15.82014 -11.636987 43.27726
2       400 19.79643  -7.637503 47.23037
```

The longer book costs about 4 dollars more. So the effect size, to judge from this model, is $4 dollars divided by 200 more pages, which comes to 2 cents per page.

Another example: Are hardcovers more expensive than paperbacks? The output is a quantitative variable: price. The input is categorical. In the `moderndive::amazon_books` data frame the variable `hard_paper` has levels "P" and "H." A possible model:

```
Mod <- lm(list_price ~ hard_paper, data = amazon_books)
model_eval(Mod, hard_paper = c("P", "H"))
```

```
  hard_paper  .output      .lwr     .upr
1          P 17.13523 -10.62291 44.89338
2          H 22.39393  -5.46052 50.24839
```

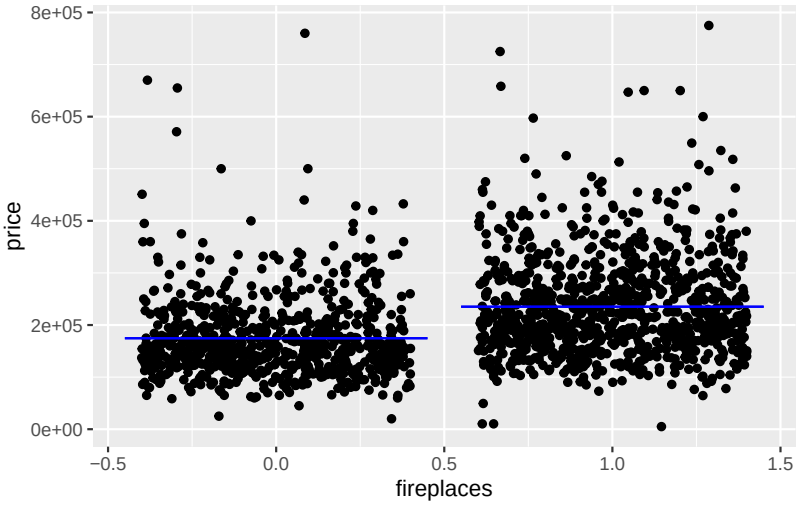
6.2 Multiple explanatory variables

When a model has more than one explanatory variable, there is a separate effect size for each. To illustrate, let's consider prices of houses as recorded in the `mosaicData::SaratogaHouses` data frame, based on house sales in Saratoga County, NY, USA in 2006. We'll follow a question asked by then-student Candice Corvetti in her Stat 101 class at Williams College: "How much is a fireplace worth?" Response variable: `price`. Explanatory variable: `fireplaces`. Since a handful of the houses has multiple fireplaces, we will simplify by filtering out those houses to retain only the ones with a single fireplace or none.

```
Simplified <- SaratogaHouses %>%
  filter(fireplaces <= 1)
Mod <- lm(price ~ fireplaces, data = Simplified)
Mod_values <- model_eval(Mod, fireplaces = c(0,1))
Mod_values
```

```
  fireplaces  .output       .lwr     .upr
1          0 174653.3  -751.4332 350058.1
2          1 235162.9 59783.5404 410542.3
```

```
Simplified %>%
  ggplot(aes(x=fireplaces, y = price)) +
  geom_jitter() +
  geom_errorbar(data=Mod_values, aes(ymin=.output, ymax=.output, x = fireplaces), y=NA,
                color="blue")
```



From the graphic, you can see that houses with a fireplace tend to have higher prices. From the report of the evaluated model, you can calculate the effect size: \$235K for a house with a fireplace, \$175K for a house without one. This suggests the value of a fireplace is \$60K.

There are, of course, many other things that determine the price of a house. Real-estate agents famously list the three most important factors as "location, location, and location." Common sense brings in other explanatory variables: how big the house is, how luxurious, how many bathrooms, and so on. The statistical thinker knows to put any one explanatory variable into the context of other plausable factors.

For simplicity, let's collect all the factors other than `fireplaces` into a hypothetical variable which we will call "`fancy`." Here are three plausible DAGs that plausibly describe an affect of fireplace on `price` in the context of `fancy`.

> **Warning**
>
> Manually insert the three-part figure as a single png

In DAG A, `fancy` and `fireplace` both contribute to `price`, but independently. In DAG B, `fireplace` directly contributes

to `price`, but whether or not a house has a fireplace depends on the level of `fancy`. In DAG D, `fireplace` has no direct affect on `price`, which is set entirely by `fancy`. The `fireplace` variable is just an indicator of `fancy`.

We can't say from the data alone which of these three DAGs is the closest description of the situation. In Lessons 28, 30, and 31 we will consider how the choice of explanatory variables in a model leads to a faithful or misleading picture of the connections. There you will find out that DAGS A & B both imply that `fancy` should be an explanatory variable if we want the effect size from the model to represent the **direct** effect of a fireplace on price. Easy enough to fit that model, ... except that we don't have an actual variable `fancy` in the `SaratogaHouses` data frame. To keep things simple for the moment, we will use `livingArea`—the size of the house—as a rough approximation to the hypothetical `fancy`.

The effect size of `fireplaces` on `price` is found by comparing the model output for houses with and without a fireplace, *holding the values of all the other explanatory variables* **constant**.

```
Mod2 <- lm(price ~ fireplaces + livingArea, data = Simplified)
model_eval(Mod2, fireplaces = c(0,1), livingArea = 2000)
```

```
  fireplaces livingArea  .output     .lwr     .upr
1          0       2000 234706.4 101212.9 368199.9
2          1       2000 240420.8 106988.7 373852.8
```

For a house with living area 2000 feet^2, the model output is \$235K with no fireplace and \$240K with a fireplace, putting the effect size of `fireplace` on `price` at \$5K. That's much smaller than the previous model, `price ~ fireplace`, gave for the effect size. The reason for the difference in results from the two models is that houses with fireplaces tend to be larger in area.

6.3 Confidence intervals

Statistical thinkers know that any estimate they make, including estimates of effect sizes, are subject to sampling variation. Consequently, an *interval* estimate should be given. This communicates to the decision maker the uncertainty in the quantity being estimated. Sophisticated decision makes take this uncertainty into account, considering the range of outcomes likely from whatever use they make of an effect size. Statistically naive decision makers—even highly educated descision makers can be statistically naive—look at the interval and will sometimes ask the modeler, "Just give me a number. I don't know what to do with two numbers." Such a request might elicit a frank response: "If you don't know what to do with two numbers, you also won't know what to do with one number." That kind of frankness is not often well received; a reasonable alternative is: "The interval indicates the amount of uncertainty in the result. If you would like to reduce the uncertainty, we'll need to collect more data." (In Lesson 29 you'll meet a not-always-available alternative to collecting more data: building a better model!)

The appropriate interval estimate for an effect size is called a "**confidence interval**." It's extremely important to keep this name in mind, since there is another kind of interval to quantify uncertainty, called a "prediction interval," which will be introduced in Lesson's 25 and 26. Confusing the two kinds of intervals is a serious blunder.

Confidence interval can be constructed using the same sorts of techniques introduced in Lesson 23. For models that are constructed by adding together different terms, like the `price ~ fireplaces + livingArea` model of the previous example, the estimated effect size for a given term is the corresponding model coefficient. The confidence interval on that effect size is simply the confidence interval on the coefficient. For example, for fireplaces:

```
lm(price ~ fireplaces + livingArea, data = Simplified) %>% confint()
```

```
                2.5 %    97.5 %
```

```
(Intercept)  6979.0960 27188.993
fireplaces  -1521.3683 12950.131
livingArea    102.7093   114.913
```

Thus, the confidence interval for the effect of a fireplace ranges from negative $1500 to positive $13,000. Broad though this may seem at first, it does carry genuine information. You can be confident that a fireplace alone will not add as much as $50,000 to the price of the house, nor will it cause the house's value to fall by $10,000.

The confidence interval on the `livingArea` is pretty narrow $103 to $115 per square foot. If you're looking to save a bit of money by shopping for a slighly smaller house, say 200 square-feet smaller, you can adjust your budget downwards by something in the range of $206,000 to $230,000. The units here come from multiplying the area units (square feet) by the effect size units (dollars per square feet), producing a quantity denominated in dollars.

It's important always to keep in mind that an estimate of an effect size will likely be misleading if your choice of model seriously misrepresents reality. For instance, a salesperson hawking add-on fireplaces might show you results from the "obvious" model `price ~ fireplace`, leading to an effect size of $52,000 to $69,000, calculated this way.

```
lm(price ~ fireplaces, data = Simplified) %>% confint()
```

```
                2.5 %    97.5 %
(Intercept) 168209.69 181097.01
fireplaces   51899.26  69119.92
```

It would be unfair to say that the $52,000 to $69,000 claim is a lie; it's entirely consistent with the data. But it relies on a grossly implausible description of the factors that determine house price.

⚠ Note in draft: For the confounding section

An idea ...
Suppose the DAG is that fireplaces cause living area (`fancy`) and that both of these cause price. That's distinct from DAG C in the above, because the causal arrow from `fancy` to `fireplace` is reversed. Could we decide between DAG C and this new DAG. How about the models `fireplace ~ livingArea` versus `fireplace` versus `fancy` plus `price`.

6.4 Interaction

⚠ Note in draft:

Not all effects are additive.

7 Mechanics of prediction

Prof. Danny Kaplan
November 17, 2022

We make a prediction when we have known values for some aspects of the system, but do not yet know the values of other aspects of the system but wish to infer what they might be by a calculation on the known values. An example: We wish to predict whether a patient will develop cancer or not. The patient is not known to have cancer at present, but we do know other relevant aspects of the patient: family history (e.g. which relatives developed cancer already), genetic markers (such as the BRCA marker of risk for breast cancer), exposure to environmental or workplace carcinogens, habits such as smoking, etc.

The prediction itself is the output of a kind of special-purpose machine. The inputs given to the machine are values for what we already know, the output is a value (or interval) for the as-yet-unknown aspects of the system. In the cancer prediction example, the output would take the form of a probability or probability rate: the probability of developing cancer in the upcoming 10-year period or the odds of developing cancer stated as a rate of odds per time unit. (For the present, just treat "odds" as a synonym for probability. In Lesson 33 we'll get more specific about how odds differ from probability and why they are used to quantify rates of probability.)

There are always two phases involved in making a prediction. The first is building the prediction machine. This is often done once in preparation for making a batch of predictions. The

second phase is providing the machine with inputs for the individual case, turning the machine crank, and receiving the prediction as output.

These two phases require different sorts of data. Building the machine requires a "historical" data set that includes records for many instances where we already know both the inputs that will be used as well as the observed output. The word "historical" emphasizes that the machine-building data must already have known values for each of the inputs and outputs of interest.

The evaluation phase—turning the crank of the machine—is simple: take values for the inputs for the case you want to predict, put them into the machine, and receive a predicted value as output. Those input values may come from pure speculation, or they might be the measured values from a case of interest.

This is practically the same as statistical modeling: the machine is the model function and has a specific format, e.g. a linear equation or some other function. Training data are used to adjust parameters of the function to find ones that do a good job matching the data. In the linear models `lm()` is the model trainer and the best parameters found become the coefficients of the model.

It's a mistake, however, to think that the model function is everything. Appropriate use of prediction requires additional information that stems from the training data but is not part of the formula. Let's look at this a little more closely using computer commands and the house `price ~ fireplaces + livingArea` model. You are not expected to master the commands introduced in the following demonstration.

> **Warning**
>
> With the `SaratogaHouses` data frame, we started by eliminating the handful of houses (mansions or historical structures?) that have multiple fireplaces, to create a new data frame which we called `Simplified`.

```
Simplified <- SaratogaHouses %>% filter(fireplaces <= 1)
```

From your work with data wrangling in the first half of the course, you know that wrangling functions like `filter()` return a data frame as output. The next R command—not one you need to know—interrogates the object `Simplified` to find out what "type" of thing it is. This is much the same as someone holding up some object from your kitchen and asking you what kind of thing it is. The thing might be a toaster or a glass or a pan.

```
class(Simplified)
```

```
[1] "data.frame"
```

Another class of R object you have used is `price ~ fireplaces + livingArea`. We have been calling this a "tilde expression" in honor of the tilde character that makes such things special. The R name for this is:

```
class(price~fireplaces + livingArea)
```

```
[1] "formula"
```

The official name for "tilde expression" is "formula." This is fine for people whose business is R programming, but for people who use mathematics in other ways a "formula" can be something different, e.g.
$\$5000 \times$ `fireplaces` $+ \$110 \times$ `livingArea`.
Another kind of R object that you make a lot of use of is called a "function."

```
class(lm)
```

```
[1] "function"
```

```
class(filter)
```

```
[1] "function"
```

```
class(model_eval)
```

```
[1] "function"
```

Knowing that something has class `"function"` tells you that you can put a pair of parentheses after it and some arguments in the parentheses, and R will know what to do (so long as your arguments are suited to the function.) For instance, we're often building models by evaluating the `lm()` function with two arguments: a tilde expression (officially, `"formula"`) and a data frame:

```
thing <- lm(price ~ fireplaces + livingArea, data = Simplified)
```

The word `thing` is a terrible name for an R object; the name doesn't serve to remind the human reader what the purpose of the `thing` is. As for R, it can sort out what type of thing any object is just by asking:

```
class(thing)
```

```
[1] "lm"
```

This `"lm"`-class object can be used in certain ways and not others. For example, we can't treat it as a function by providing arguments in paretheses:

```
thing(fireplaces = 1, livingArea = 2000)
```

```
Error in thing(fireplaces = 1, livingArea = 2000): could not find function "thing"
```

One of the operations you can apply to an object of class `"lm"` is `makeFun()`, which will create a function:

```
f <- makeFun(thing)
class(f)
```

```
[1] "function"
```

This particular function, named `f`, can have arguments applied to it:

```
f(fireplaces=1, livingArea=2000)
```

```
       1
240420.8
```

Even if you can't evaluate a `"lm"` object using parentheses, there are other things you can do with it by applying a suitable function. For instance:

```
coefficients(thing)
```

```
(Intercept)  fireplaces  livingArea
 17084.0446   5714.3813    108.8112
```

```
confint(thing)
```

```
                2.5 %    97.5 %
(Intercept)  6979.0960 27188.993
fireplaces  -1521.3683 12950.131
livingArea    102.7093   114.913
```

```
rsquared(thing)
```

```
[1] 0.480277
```

```
residuals(thing) %>% head()
```

```
        1          2          3          4          5          6
 11118.66  -48477.26 -125327.34  -79327.34  -22425.43  -28148.89
```

But you can't do all these same things to the *function* extracted from `thing` using `makeFun()`:

```
confint(f)
```

```
Error in UseMethod("vcov"): no applicable method for 'vcov' applied to an object of class "fu
```

```
rsquared(f)
```

```
NULL
```

```
residuals(f)
```

```
Error in object$na.action: object of type 'closure' is not subsettable
```

The function `model_eval()` is set up to take a `"lm"` object and evaluate it on inputs. First build the model, then you can evaluate it as many times as you want:

```
house_mod1 <- lm(price ~ fireplaces, data = Simplified)
model_eval(house_mod1, fireplaces=1, livingArea=2000)
```

```
  fireplaces livingArea  .output     .lwr     .upr
1          1       2000 235162.9 59783.54 410542.3
```

```
house_mod2 <- lm(price ~ fireplaces + livingArea, data = Simplified)
model_eval(house_mod2, fireplaces=1, livingArea=2000)
```

```
  fireplaces livingArea  .output     .lwr     .upr
1          1       2000 240420.8 106988.7 373852.8
```

The output of `model_eval()` is a data frame containing one column for each of the inputs to the model. After those columns, comes the point output from the model, sensibly labeled `.output`. `model_eval()` uses the model coefficients to transform the inputs into the `.output`. But it goes further. The model object itself—`house_mod1` or `house_mod2` here—contains additional information about the results from training the model, for example the residuals from the trained model and the number of rows in the training data frame. `model_eval()` takes this information to produce an interval

estimate for the prediction. The lower and upper ends of that interval are reported in the columns `.lwr` and `.upr`, respectively. This interval is called, naturally, a "**prediction interval**."

Statistical thinkers know that a prediction should always contain information about how uncertain the prediction is. That indication of uncertainty is provided by the prediction interval. To see why this is useful, look at the prediction interval for the rather silly model `price ~ fireplaces`; it spans a huge range from $60K to $410K. On the other hand, the model `price ~ fireplaces + livingArea` is covers a smaller range: $107K to $374K.

In Lesson 26 we'll look at the components that make up the prediction interval and some of the ways to use it.

8 Constructing a prediction interval

Prof. Danny Kaplan
November 17, 2022

In Lesson 25 you encountered the **prediction interval**. A model is trained on historical data. When a prediction is needed, values for the known inputs are provided and the model function is used to produce the output from the model. For instance, here's a model which relates the running times in Scottish hill races to the race distance and climb.

```
time_mod <- lm(time ~ distance + climb, data = Hill_racing)
```

Every model has a "model function" which calculates the value of the output when given values for the inputs. We won't often have to work with the model function because there is additional information, not provided by the model function that lets us make better interpretations of the model. But, for the sake of illustration, here we will extract the model function and evaluate it for a 10-km-long race that climbs 500m.

> **Warning**

```
time_fun <- makeFun(time_mod) # Extract model function
time_fun(distance=10, climb=500) #Evaluate it
```

```
       1
3372.985
```

The output of the model function is a **point prediction**:

a single value. Here, that value is 3373 seconds, or about 56 minutes.

Many people like a point prediction, possibly because the single number suggests a single, correct answer, which is somehow emotionally comforting. *But the comfort is unjustified.*

A proper form for a prediction is a **prediction interval**; two numbers setting the lower and upper limits for likely outcome once the new 10-km/500m race is actually run. To construct a prediction interval, use the `model_eval()` function, giving as arguments the prediction model as well as the inputs:

```
model_eval(time_mod, distance=10, climb=500)
```

```
  distance climb  .output    .lwr    .upr
1       10   500 3372.985 1664.41 5081.56
```

From this report, you can read off the same model output as provided by the model function. But added to it are the lower and upper bounds of the prediction interval, here 1660 s to 5082 d. That's a big interval! The low end is less than one-third the time of the high end.

This Lesson is about what goes into finding a prediction interval. Evidently, there are a lot of other factors in Scottish hill racing than distance and climb.

8.1 Where does the prediction interval come from

The prediction interval has two distinct components:

1. The uncertainty in the model function and hence in the output of the model function.
2. The size of the residuals found when training the model.

Consider first the model function. For the running-time model, we can construct the model function from the coefficients of the linear model. These are:

```
time_mod %>% coefficients()
```

```
(Intercept)    distance       climb
-469.976937  253.808295    2.609758
```

The model function is therefore

$$t(d, c) \equiv -470 + 254d + 2.61c$$

To get the point prediction, simply calculate $t(d = 10, c = 500)$.

The statistical thinker knows that rather than a point estimate for the coefficients, it is better to state the model in terms of the *confidence interval* on the coefficients. Here, that is:

```
time_mod %>% confint()
```

```
                 2.5 %      97.5 %
(Intercept) -533.432471 -406.521402
distance     246.387096  261.229494
climb          2.493307    2.726209
```

Since we cannot legitimately claim to know the values of the coefficients any better than indicated by these confidence intervals, we ought to temper our claims about the model function so that it reflects the uncertainty in the coefficients. For instance, we might provide an interval for the model output, using in an "upper" function the high ends of the confidence intervals on the coefficients and another, "lower" function that uses the low ends of the confidence interval. Like this:

$t_{upr}(d, c) \equiv -407 + 261d + 2.72ct_{lwr}(d, c) \equiv -533 + 246d + 2.49c$

To get an *interval* on the model output, we can evaluate both the lower and upper functions. That would give us $t_{lwr}(10, 500) = 3172$ and $t_{upr}(10, 500) = 3569$.

This particular idea for generating the "lower" and "upper" functions has the right spirit, but is not on target mathematically. The reason is that using the low end of the confidence interval for all coefficients is overly pessimistic; usually the uncertainty in the different coefficients cancels out to some extent.

This is not the place to go into the mathematics of making the "lower" and "upper" functions. Without going into any mathematical details we'll just stipulate that `model_eval()` knows how to do the calculations correctly. To see the results of the correct calculation, ask `model_eval()` for a confidence interval on the model output.

```
model_eval(time_mod, distance=10, climb=500, interval="confidence")
```

```
  distance climb  .output     .lwr     .upr
1       10   500 3372.985 3335.264 3410.706
```

Reading this report, you can see that the confidence interval on the model output for a 10km/500m race is pretty narrow: 3335 seconds to 3411 seconds, or, in plus-or-minus format, 3373 ± 38 seconds.

Let's compare the **confidence** interval on the model output to the **prediction interval** as calculated by `model_eval()`. To repeat a calculation that we did at the start of the Lesson, the prediction interval is

```
model_eval(time_mod, distance=10, climb=500, interval="prediction")
```

```
  distance climb  .output    .lwr    .upr
1       10   500 3372.985 1664.41 5081.56
```

The prediction interval is huge compared to the confidence interval: 1664 to 5082 seconds, or, in plus-or-minus format, 3373 ± 1709 seconds.

Why is the prediction interval so much wider than the confidence interval? The confidence interval reports on the sampling variation of a model constructed as a kind of average over all the data, the $n = 2236$ participants recorded in the `Hill_racing` data frame. But each individual runner in `Hill_racing` has their own individual time: not an average but just for the individual. The individual value might be larger or smaller than the average. How much larger or smaller? This is recorded in the residuals for the model. As always, we can measure the variation from individual to individual in their residuals with the standard deviation.

```
time_mod %>% residuals() %>% sd()
```

```
[1] 870.6588
```

Keeping in mind that the overall spread of the residuals is plus-or-minus "twice" the standard deviation of the residuals, we can say that the residuals indicate an additional uncertainty in the prediction for a runner of about ± 1700 seconds. This ± 1700 seconds is our estimate of the **noise** in the measurements. The **confidence interval**, is about the sampling variation in the signal. This sampling variation is the portion of noise that is inherited by the averaging process that is involved in calculating the coefficient.

The **prediction interval** is, in this case, completely dominated by noise; the sampling variability contributes only a tiny amount of addition uncertainty.

⚠ Demonstration: Simple-minded data analysis shows why

We constructed our model of running time using the "linear least-squares" modeling methodology implemented by the `lm()` model-training function. For the purpose of demonstration, we'll show you a simple-minded method

to make a prediction. This simple-minded method would give results more or less equivalent to the least squares method if we had an almost infinite amount of data. Since we don't, the simple-minded method is not as reliable as the least-squares method.
Remember our goal: to predict the running time for a 10km race with a 500m climb. All that we have to inform the prediction is the historical data contained in the `Hill_racing` data frame. The simple-minded method is … well … simple to understand. We will pull out from the `Hill_racing` data frame those rows where the race distance is close to 10km and the race climb is close to 500m. For example:

```
close_rows <- Hill_racing %>%
  filter(9 <= distance, distance <= 11,
         450 <= climb, climb <= 550)
## the prediction
close_rows %>% summarize(sample_size=n(), pred = mean(time))
```

```
# A tibble: 1 x 2
  sample_size  pred
        <int> <dbl>
1          52 3523.
```

You might not agree with our definition of "close to 10km" as "between 9 and 11", and similarly for `climb`.
To get the confidence interval on this simple-minded prediction, we point out that the value of the mean time is the same as the coefficient from the model `time ~ 1`. Let's fit that model formula to the `close_rows` and look at the coefficient and confidence interval.

```
simple_mod <- lm(time ~ 1, data = close_rows)
simple_mod %>% coefficients()
```

```
(Intercept)
   3523.481
```

```
simple_mod %>% confint()
```

```
                2.5 %   97.5 %
(Intercept) 3363.901 3683.061
```

To interpret this confidence interval, we can plot the actual running times and compare them to the interval, as in Figure 8.1

```
ggplot(close_rows, aes(y=time, x=1)) +
  geom_jitter(width=0.2) + xlim(0,2) +
  geom_violin(fill="blue", alpha=0.2, color=NA) +
  geom_errorbar(aes(ymin=3364, ymax=3683), color="red") +
  geom_errorbar(aes(ymin=1665, ymax=5082, x=1.5), color="blue")
```

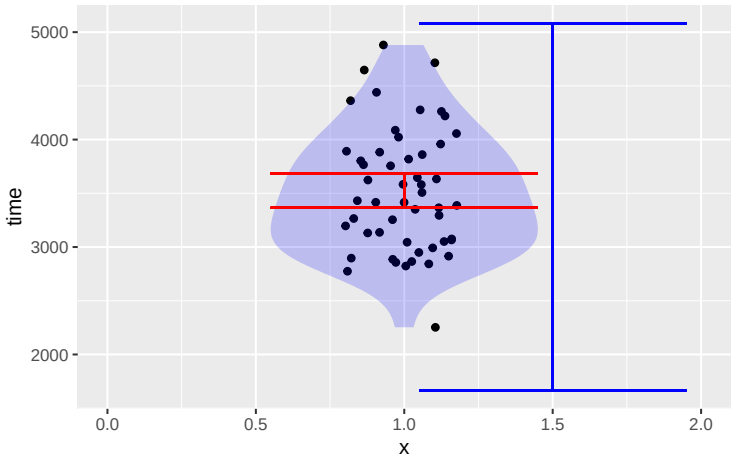


Figure 8.1: The rows from `Hill_racing` with a distance close to 10km and climb close to 500m. The confidence interval on the mean time (shown in red) is narrow compared to the prediction interval (blue) calculated from the whole data frame and the model `time ~ distance + climb`.

8.2 Example: Predicting running time from age

We shift the running scene from Scotland to Washington, DC. The race now is a single 10-miler with almost 9000 registered participants. We wish to predict the running time based on age. Since we're looking at the prediction as a function of an input variable, what we formerly showed using the "errorbar" glyph is now shown using a ribbon or **band**. As you'll see, the prediction band is so large as to be useless. But you'll also see why the confidence band on the model output is completely misleading about the uncertainty in the prediction.

The following commands train the model `net ~ age` (where `net` is the net running time, start line to finish line) and plot the confidence band on the model output along with the actual data.

```
age_mod <- lm(net ~ age, data = TenMileRace)
model_plot(age_mod, x=age, interval="confidence", data_alpha=0.05)
```

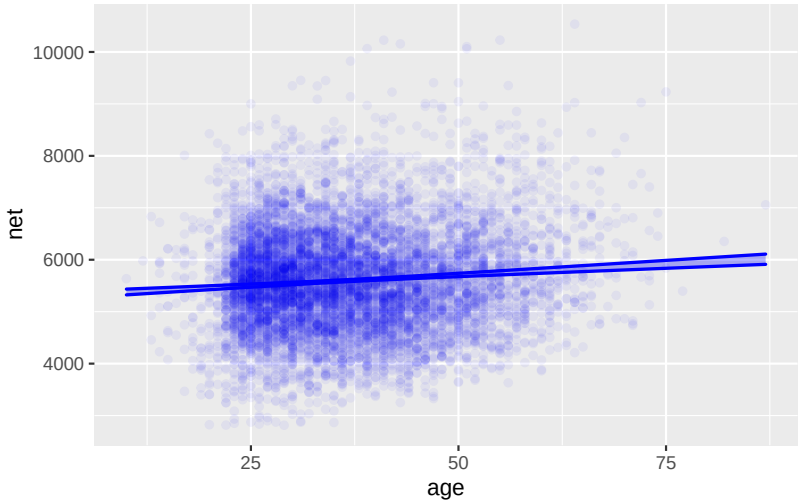


Figure 8.2: The confidence interval on the output from the model `net ~ age`. This includes only a tiny fraction of the actual data points.

You can see that the confidence band on the model output includes only a minute fraction of the actual running times recorded in `TenMileRace`. On the other hand, the prediction band—Figure 8.3 —includes the large majority of the actual running times.

```
model_plot(age_mod, x=age, interval="prediction", data_alpha=0.05)
```

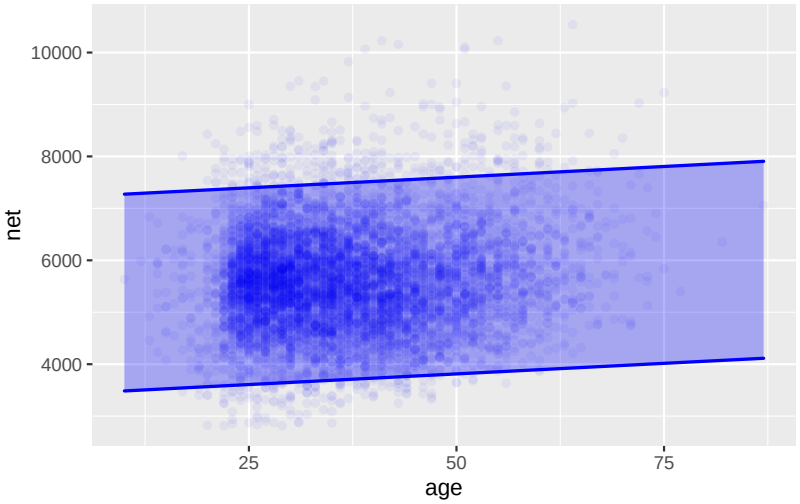


Figure 8.3: The prediction interval from the model `net ~ age`. This includes the vast majority of the actual data points.

The prediction interval doesn't include *all* the actual running times, but it is not meant to. The prediction interval shown is a 95% prediction interval, and so ought to include only about 95% of the actual running times.

To summarize:

1. When making a prediction, report a prediction interval.
2. The prediction interval is always larger than the confidence interval and often *much* larger.

The confidence interval is not useful for predictions. Use a confidence interval when looking at an effect size. Graphically, the confidence interval is useful to indicate whether there is

an overall trend in the model. For instance, Figure 8.2 shows a clear upward trend in running time with age. There is no level or negatively sloping line compatible with the confidence interval.

9 Review of Lessons 1-8

> **Warning**
>
> I'll put learning challenges here. The class day will be given over to the QR.

10 Covariates

Prof. Danny Kaplan
November 17, 2022

Dr. Mary Meyer is a statistics professor at Colorado State University. In 2006, she published an article recounting an episode from family life:

> *When my daughter was in fourth grade, I took her shopping for dress shoes. I was disappointed in the quality of girls' shoes at every store in the mall. The shoes for boys were sturdy and had plenty of room in the toes. On the other hand, shoes for girls were flimsy, narrow, and had pointed toes. In spite of the better construction for boys, the costs of the shoes were similar! For children the same age, boys had shoes they could run around in, while girls' shoes were clearly for style and not comfort.*
>
> *Upon complaining about this state of affairs, I was told by sales representatives in two stores that boys actually had wider feet than girls, so needed wider shoes. Being very skeptical, I thought I would test this claim.*

We'll return to Dr. Meyer's project in a little bit. But now let's imagine how this situation might be addressed by someone who has not yet developed good statistical thinking skills. We'll call this imagined protagonist "Mr. Shoebuyer." Since the salesmen's claim was that girl's feet are narrower than boys, Mr. Shoebuyer heads out to measure the widths of girls' and boys' shoes.

A shoe store provides convenient place to measure the widths of lots of different shoe styles. Mr. Shoebuyer gets to the shoe store, heads to the children's section and starts measuring. For each shoe on display he records the shoe width and whether the shoe is for girls or boys. Here are his data:

sex	width
G	9.0
G	8.5
G	9.0
G	9.5
B	8.6
B	8.4
B	8.8
B	9.4

Once back home, Mr. Shoebuyer uses his calculator to find the mean width of the shoes in each group. His results surprise him:

sex	mean width
Girls	9.0 cm
Boys	8.8 cm

Mr. Shoebuyer happens to be your uncle. He knows that you are taking a statistics course and writes you with a request to recheck his arithmetic. Putting on your statistical thinking hat, you see immediately that what's missing is a confidence interval on the effect size of sex on shoe width.

```
Shoebuyer_data %>% lm(width ~ sex, data=.) %>% confint()
```

```
                2.5 %    97.5 %
(Intercept)  8.2857603 9.3142397
sexG        -0.5272448 0.9272448
```

At the Thanksgiving break, you see your uncle. You say, "Sorry, Uncle, but you don't have nearly enough data to conclude that girls' feet are wider than boys'." Translating your confidence

interval into plus-or-minus format, you point out that the difference between the sexes is 0.2 ± 0.8 cm. "You'll need enough data to get that 0.8 margin of error down to something like 0.2." You also point out that a shoe store might not be an appropriate place to collect data. "It's the feet, not the shoes, that you want to look at."

Dr. Meyer worked worked with the third- and fourth-grade teachers at her daughter's school to collect data to illuminate the matter. Being a statistical thinker, before carrying out the data collection, she thought about what data would illuminate the matter. Her data, a sample of size $n = 39$, are recorded in the `KidsFeet` data frame.

```
lm(width ~ sex, data = KidsFeet) %>% confint()
```

```
                2.5 %      97.5 %
(Intercept)  8.9758882  9.40411181
sexG        -0.7125476 -0.09903131
```

Translated to plus-or-minus format, this confidence interval is -0.4 ± 0.3. Whatever the format, Dr. Meyer's data provides some evidence that girls' feet are narrower than boys'.

As a statistical thinker, Dr. Meyer knows that even though the foot width is the original quantity of interest, other factors might be playing a role in the system. For example, boys' feet might trend longer or, perhaps, shorter than girls' feet. This should be taken into account. What we want is the effect size of `sex` on width, holding length constant. After all, when buying shoes you tell the salesperson your foot length (or "size") and they bring you shoes of that size to choose among.

```
lm(width ~ sex + length, data=KidsFeet) %>% confint()
```

```
                2.5 %     97.5 %
(Intercept)  1.1048182 6.17751841
sexG        -0.4947759 0.02974084
length       0.1202348 0.32181513
```

Although `sex` is the explanatory variable of primary interest to Dr. Meyer's question, she knows to include other explanatory variables that might be playing a role. Such explanatory variables, not of direct interest, are called "**covariates**." Dr. Meyer's expertise led her to think about possible covariates *before* collecting her data. That's why she went to the trouble of measuring foot length in addition to foot width.

The confidence interval on the `sexG` coefficient includes zero when `length` is taken into account. Dr. Meyer's little study provides evidence that, even if girls' shoes tend narrower than boys', the feet inside them have about the same shape for both sexes.

The common phrase "all other things being equal" is an important qualifier in describing relationships. To illustrate: A simple claim in economics is that a high price for a commodity reduces the demand. For example increasing the price of heating fuel will reduce demand as people turn down thermostats in order to save money. But the claim can be considered obvious only with the qualifier *all other things being equal*. For instance, the fuel price might have increased because winter weather has increased the demand for heating compared to summer. Thus, higher prices may be associated with higher demand. Unless you hold other variables constant – e.g., weather conditions – increased price may not in fact be associated with lower demand.

In fields such as economics, the Latin equivalent of "all other things being equal" is sometimes used: "**ceteris paribus**". So, the economics claim would be, "higher prices are associated with lower demand, *ceteris paribus*."

Although the phrase "all other things being equal" has a logical simplicity, it's impractical to implement "all." Instead of the blanket "all other things," it's helpful to be able to consider just "some other things" to be held constant, being explicit about what those things are. Other phrases along these lines are "taking into account …" and "controlling for …." Such phrases apply when you want to examine the relationship between two variables, but there are additional variables that may be coming into play. The additional variables are called "**covariates**" or "**confounders**".

Example: Covariates and Death

This news report appeared in 2007:

> **Heart Surgery Drug Carries High Risk, Study Says.** A drug widely used to prevent excessive bleeding during heart surgery appears to raise the risk of dying in the five years afterward by nearly 50 percent, an international study found. The researchers said replacing the drug—aprotinin, sold by Bayer under the brand name Trasylol—with other, cheaper drugs for a year would prevent 10,000 deaths worldwide over the next five years.
> Bayer said in a statement that the findings are unreliable because Trasylol tends to be used in more complex operations, and the researchers' statistical analysis did not fully account for the complexity of the surgery cases. The study followed 3,876 patients who had heart bypass surgery at 62 medical centers in 16 nations. Researchers compared patients who received aprotinin to patients who got other drugs or no antibleeding drugs. Over five years, 20.8 percent of the aprotinin patients died, versus 12.7 percent of the patients who received no antibleeding drug. [This is a 64% increase in the death rate.] When researchers adjusted for other factors, they found that patients who got Trasylol ran a 48 percent higher risk of dying in the five years afterward. The other drugs, both cheaper generics, did not raise the risk of death significantly. The study was not a randomized trial, meaning that it did not randomly assign patients to get aprotinin or not. In their analysis, the researchers took into account how sick patients were before surgery, but they acknowledged that some factors they did not account for may have contributed to the extra deaths. - Carla K. Johnson, Associ-

ated Press, 7 Feb. 2007

The report involves several variables. Of primary interest is the relationship between (1) the risk of dying after surgery and (2) the drug used to prevent excessive bleeding during surgery. Also potentially important are (3) the complexity of the surgical operation and (4) how sick the patients were before surgery. Bayer disputes the published results of the relationship between (1) and (2) holding (4) constant, saying that it's also important to hold variable (3) constant.
With aprotinin, the total relationship involves a death rate of 20.8 percent of patients who got aprotinin, versus 12.7 percent for others. This implies an increase in the death rate by a factor of 1.64. When the researchers looked at a partial relationship (holding constant the patient sickness before the operation), the death rate was seen to increase by less: a factor of 1.48. In evaluating the drug, it's best to examine its effects holding other factors constant. So, even though the data directly show a 64% increase in the death rate, 48% is a more meaningful number since it adjusts for covariates such as patient sickness. The difference between the two estimates reflect that sicker patients tended to be given aprotinin. As the last paragraph of the story indicates, however, the researchers did not take into account all covariates. Consequently, it's hard to know whether the 48% number is a reliable guide for decision making.

10.1 "*Mutatis mutandis*"

Using covariates in models enable the relationship between a response and an explanatory variable to be described "all other things being equal." Another phrase used in news stories is "after adjusting for …", since the *all* in "all other things" is more properly restricted just to those factors represented by the covariates actually used in a model. So, Dr. Meyer's foot width results might be stated in everyday language as, "After

adjusting for foot width, she found no difference in the widths of girls' and boys' feet."

Not to include covariates in a model amounts to "letting other things change as they will." In Latin this is "*mutatis mutandis*." In the foot-width example, the model `width ~ sex` looks at the differences in foot width for the two sexes. But sex is not the only thing "changed" when comparing foot width. Since `width ~ sex` ignores all other factors than sex, it is comparing boys and girls letting other things change as they will. In this case, comparing boys and girls involves not just the possible differences in foot width but the differences as well in other factors: foot length, body weight, etc.

i Example: One change can bring another

I was once involved in a budget committee that recommended employee health benefits for the college at which I work. At the time, college employees who belonged to the college's insurance plan received a generous subsidy for their health insurance costs. Employees who did not belong to the plan received no subsidy but were instead given a modest monthly cash payment. After the stock-market crashed in year 2000, the college needed to cut budgets. As part of this, it was proposed to eliminate the cash payment to the employees who did not belong to the insurance plan. This proposal was supported by a claim that this would save money without reducing health benefits. I argued that this claim was based on an "all other things being equal" analysis: how expenditures would change assuming that the number of people belonging to the insurance plan remained constant. But in reality, the policy change would play out *mutatis matandis*; the loss of the cash payment would cause some employees, who currently received health benefits through their spouse's health plan, to switch to the college's health plan. That's what happened, contributing to an increase of health-care expenses.

11 Covariates eat variance

Prof. Danny Kaplan
November 17, 2022

> ⚠ Warning
>
> - Covariates reduce residuals in-sample.
> - Out of sample, they may or may not reduce residuals, depending on whether the covariate is informative.
> - Covariates can lead to better predictions since prediction intervals are substantially shaped by the size of the residuals
> - They can lead to better or worse estimates of effect size (colinearity)

In model building, we create a function to link a response variable to one or more explanatory variables. Let's imagine that the response variable is named `y` and the explanatory variable named `x`, `a`, `b`, and so on. By *training* a model on data, we create a function, let's call it $f(x, a, b)$. This function that results from training can be used in two two distinct decision-making settings:

i. Prediction mode: Having measured values for the explanatory variables `x`, `a`, `b` and so on, figure out what's a likely value for the as-yet-unmeasured response variable.

To illustrate, suppose we know the length `x` of a road trip, the speed driven `a`, and the horsepower of the car's engine. To plan for the trip, we want to predict how much fuel `y` will be used.

ii. Intervention mode: We propose to intervene in the world to change the value of `x` by an amount `dx`. For instance, suppose `x` is the dose of a drug taken by a patient and `y` is the patient's blood pressure. If we increase the dose by `dx`, what will be the corresponding change in the blood pressure `dy`. The ratio `dy/dx` is called the **effect size**.

In intervention mode, as we saw in Lesson 28, it's important that the model formula reflect accurate the *causal connections* among the variables. The simple model `y ~ x`, without covariates, can sometimes give a misleading view of the effect of `x` on `y`. Including a covariate in the model might improve or worsen the estimate of effect size, depending on the causal connections in the real-world mechanism of the system.

In prediction mode, capturing the real-world causal connections in the model formula is not essential. For example, even if `y` is the cause of `x`, the model formula `y ~ x` might do a good job of predicting `y` from a measured value of `x`.

In this Lesson, we'll examine the use of covariates in constructing prediction models.

The output of a prediction model is typically somewhat different from what happens in the real world. The difference between the real-world value of the response variable and the output of the prediction model is called the *prediction error*. As we saw in Lesson 26, in stating the output of a prediction model, it is helpful to also be able to state a typical size for the prediction error, usually in the form of a prediction interval.

11.0.1 Alternative accountings

We've been using RMS prediction error to quantify how well the response variable has been accounted for by the explanatory variable(s). RMS prediction error is a convenient summary of the size of the typical prediction error because 1) it is an average

over all the cases in the testing data and 2) it has the same units as the response variable. But RMS is not the only the only such accounting. In this section, we'll look at two others that are widely used in statistical reports: R^2 and the "**sum of squares**".

We'll start with the *sum of squares* accounting. Recall that the letters in RMS each stand for a specific step:

- **S**: square the each of the values
- **M**: average over the (squared) values
- **R**: take the square root of the (average squared) values.

The *order* of the steps is important; S first, M next, then finally R.

The *sum of squares*, often written SS, is a two-step process.

- **S**: square each of the values
- **S**: sum (not average) the (squared) values.

Again, the *order* of the steps is important: square first then finally sum. (The notation SS doesn't make the order clear, but the name "sum of squares" does. So remember that SS stands for "sum of squares".)

> ⚠ Note in draft
>
> WHEN YOU GET TO THE ANOVA REPORT, make sure to point out that the so-called "mean square" is a confusing name because it wrongly brings to mind the MS in RMS. The quantity is really the sum of squares divided by the degrees of freedom.

12 Confounding

Prof. Danny Kaplan
November 17, 2022

Suppose you are concerned that the chemicals used by lawn-greening companies are a source of cancer or other illness. You propose to find out by collecting and modeling data; sampling many households that have used lawn-greening chemicals for at least a decade and other households that have never used lawn-greening chemicals. You'll record both chemical use and a measure of health outcome: whether anyone in that household has developed cancer in the last five years.

Here are a few rows from the data (which we have simulated for this example):

grass	cancer
organic	no
chemicals	no
chemicals	no
chemicals	no
organic	no
organic	no
organic	no
organic	no
chemicals	yes
organic	no

Analyzing such simple data is straightforward, since we are interested in the possible role of grass-greening chemicals in increasing risk of cancer. First, check the overall cancer rate:

```
lm(zero_one(cancer, one="yes") ~ 1, data = Cancer_data) %>% coefficients()
```

```
(Intercept)
      0.026
```

In these data, 2.6% of the sampled households had a cancer in the last five years. But how does the grass treatment affect that rate?

```
mod <- lm(zero_one(cancer, one="yes") ~ grass, data = Cancer_data)
coefficients(mod)
```

```
 (Intercept) grassorganic
  0.01246883   0.02258960
```

For households whose lawn treatment is "organic," the risk of cancer is higher by 2.3 percentage points compared to households that treat their grass with chemicals. This is certainly not what we were expecting, but it is what the data show. On the other hand, there is sampling variability to take into account. Let's look at the confidence intervals:

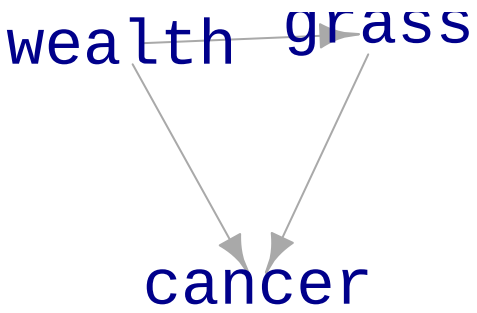
```
confint(mod)
```

```
                    2.5 %     97.5 %
(Intercept)  -0.003103381 0.02804104
grassorganic  0.002469193 0.04271001
```

The confidence interval on `grassorganic` does not include zero, but it comes pretty close. We are not sure what to conclude: Might the chemical treatment of grass be protective against cancer? This seems implausible. Might we have causality backwards? Hard to imagine that the appropriate DAG is cancer $\rightarrow$ chemical treatment; causation must be the other way around:

$$\text{cancer} \leftarrow \text{chemical treatment}\ .$$

The statistical thinker knows to consider the possible role of other factors. To form reasonable hypotheses, you need some knowledge of how the system under study works. For instance, green grass is not a necessity, so the households who treat their lawn with chemicals tend to have money to spare. It's also the case that health outcomes are somewhat better for wealthier people. In part this is because of better access to health care. Another factor is that wealthier people can live in less polluted neighborhoods and are less likely to work in dangerous conditions, such as exposure to toxic chemicals. This suggests a DAG hypothesis where "`wealth`" influences how the household's `grass` is treated and `wealth` similarly influences the risk of developing `cancer`. Like this:



A description of this structure of causality is, "The effect of grass treatment on cancer is **confounded** by wealth." The Oxford Dictionary has two definitions of "confound."

1. *Cause surprise or confusion in someone, especially by acting against their expectations.*
2. *Mix up something with something else so that the individual elements become difficult to distinguish.*

It is this second definition that describes the statistical meaning of "confound."

To be sure, the first definition seems relevant to our story, since the protagonist expected that chemical use would be associated with higher cancer rates and was surprised to find otherwise. But the statistical thinker doesn't throw up her hands when faced with the mixing up of causal factors. Instead, she uses modeling techniques to untangle the influences of the various factors.

Using covariates in models is one such technique. For instance, in generating the simulated data shown above, we used a DAG which associated greater wealth with chemical treatment of grass and also, separately, with better health outcomes. `Wealth` is one of the variables included in the data, even if we didn't show it earlier in the example:

wealth	grass	cancer
1.4283990	organic	no
0.0628559	chemicals	no
0.4382804	chemicals	no
0.6084487	chemicals	no
0.8033695	organic	no
-0.9367287	organic	no
0.6664468	organic	no
-1.2445977	organic	no
-1.3194594	chemicals	yes
-1.6162391	organic	no

Including `wealth` as a covariate in this case untangles the system so that we can see the *direct* link between chemicals and health.

```
lm(zero_one(cancer, one="yes") ~ grass + wealth, data = Cancer_data) %>%
  confint()
```

```
                  2.5 %        97.5 %
(Intercept)   0.02468113  0.0574819325
grassorganic -0.04508107 -0.0009698601
wealth       -0.05680934 -0.0356454288
```

Same data, but the opposite conclusion. With `wealth` as a covariate, "organic" lawn treatment (that is, leaving things be!) reduces the risk of cancer. But the bigger factor in shaping cancer risk is represented by `wealth`.

Keep in mind that this is simulated data. So don't draw any conclusions from the data in this example about the safety of the chemicals used by lawn-greening companies.

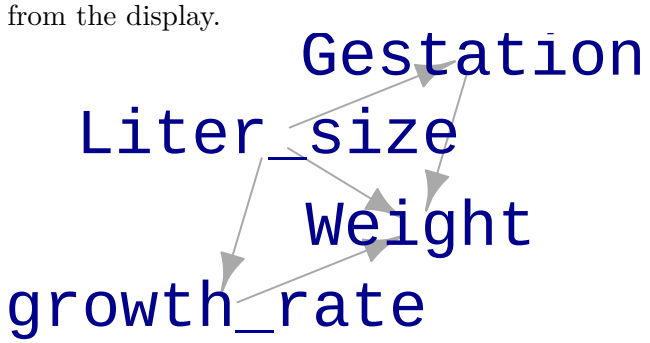
Example: A missed opportunity

Sewall Wright (1889-1988) was an American geneticist and statistical pioneer. One of his statistical inventions is the "coefficient of determination" now universally called R^2 and a widely used, basic summary of statistical models. In 1921, he invented "path analysis." One of his "path diagrams" is shown in **?@fig-path-guinea-pig**.

```
[1] "GET THIS FROM p.75 of the *Book of Why*"
```

His path diagrams are directed acyclic graphs, DAGs, augmented with coefficients representing the relative strength of each contributor to a node. He worked out the algebra of the correlation induced by the graph between any two nodes. Then, by measuring the R^2 between pairs of nodes, he was able in some cases to work backwards to numerical values for the coefficients.

Wright's path diagrams are the historically earliest form of our DAGs. In his honor, we've constructed a DAG to represent one of his calculations, how much the body weight at birth of a guinea pig increases due to one day longer in the womb. The path diagram Wright imagined is drawn below, though we have left out the coefficients from the display.



We can't measure the growth rate directly, but we can measure liter size, gestation length, and birth weight. How can we estimate the direct effect of growth rate when it is confounded with the other causal pathways.

Sewall's breeding experiments would have provided data like this:

liter_size	growth_rate	weight
5	4	86
5	5	109
5	6	110
5	5	96
6	4	80
5	5	99

You might think that weight gain per day of gestation can be simply calculated as `weight/gestation`, but this ignores the fact that weight gain is slow early in gestation and faster as the cubs develop. Instead, using a model `weight ~ gestation` lets us look at the marginal impact of an extra day of gestation. The coefficients from this model indicates that weight increases by 6.8 grams per extra day of gestation.

```
lm(weight ~ gestation, data = Pigs) %>% coefficients()
```

```
(Intercept)   gestation
 -32.444194    6.840467
```

But Wright knew that this number was misleading. Larger liters tend to have shorter gestation times. And larger liters produce cubs that weigh less. With more computational power available to us, we can use a simpler calculation to incorporate these facts into the estimation of weight gain per day of gestation:

```
lm(weight ~ gestation + liter_size, data = Pigs) %>% coefficients()
```

```
(Intercept)   gestation  liter_size
  104.19844     4.50268   -17.92767
```

This model pegs the growth rate at about 4.5 grams per day.
Since we generated the data from a DAG, we have the luxury of measuring the actual growth rate used for each liter.

```
Pigs %>% summarize(rate = mean(growth_rate))
```

```
# A tibble: 1 x 1
   rate
  <dbl>
1  4.84
```

Covariates help us deal with confounding!

$X \xrightarrow{a} Y \xleftarrow{b} C$

$\text{resid}_y^2 = \sigma_{yy} - (a^2\sigma_{xx} + b^2\sigma_{cc})$

$\sigma_{xy} = \sqrt{\sigma_{xx}\left[\sigma_{yy} - \left(b^2\sigma_{cc} + \text{resid}_y^2\right)\right]}$

```
wright_example <- dag_make(
  x ~ eps(4),
  c ~ eps(1),
  y ~ 2*x + 3*c + eps(1)
)
Sample <- sample(wright_example, size=1000)
Stats <- Sample %>%
  summarize(xv = var(x), yv=var(y), cv=var(c), xy = cov(x,y), cy = cov(y,c), xc = cov(x,c))
Normalize <- Sample %>% mutate(x = x/sqrt(Stats$xv),
                               y = y/sqrt(Stats$yv),
                               c = c/sqrt(Stats$cv))
Stats
```

```
# A tibble: 1 x 6
     xv    yv    cv    xy    cy    xc
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  16.6  76.8  1.00  33.3  3.17 0.100
```

```
with(Stats, yv - (4*xv + 9*cv)) # variance of y
```

```
[1] 1.462633
```

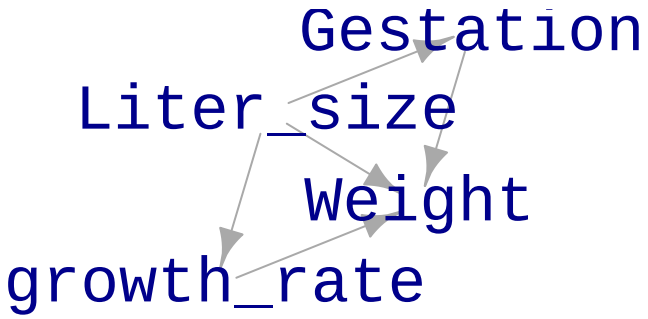
```
with(Stats, sqrt(xv*(yv - (9*cv +1)))) # covariance of x with y
```

```
[1] 33.26253
```

```
with(Stats, sqrt(cv*(yv - (4*xv + 1)))) # covariance of c with y
```

```
[1] 3.077456
```

```
dag_pigs <- dag_make(
  liter_size ~ 5 + as.numeric(eps() > 1),
  .gestation ~ 24 - liter_size + eps(),
  .growth_rate ~ 10 - liter_size + eps(.3),
  weight ~ .growth_rate*.gestation + liter_size + eps(),
  gestation ~ round(.gestation)
)
dag_pigs_for_drawing <- dag_make(
  Liter_size ~ 5 + as.numeric(eps() > 1),
  Gestation ~ 24 - Liter_size + eps(),
  growth_rate ~ 10 - Liter_size + eps(.3),
  Weight ~ growth_rate*Gestation + Liter_size + eps()
)
set.seed(103); dag_draw(dag_pigs_for_drawing)
```




```
set.seed(103); Dat <- sample(dag_pigs, size=1000)
lm(weight ~ liter_size + gestation, data = Dat) %>% confint()
```

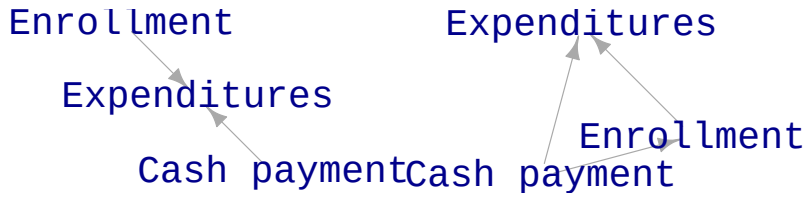
```
                2.5 %     97.5 %
(Intercept)  91.885081 111.143461
liter_size  -18.810932 -16.710462
gestation     4.267084   4.939893
```

```
lm(weight ~ gestation, data = Dat) %>% confint()
```

```
                2.5 %     97.5 %
(Intercept) -34.762796 -17.414430
gestation     6.060804   6.979343
```

12.1 DAGs and covariates

The argument, "reduce spending by reducing spending" is very compelling, common sense even. It's harder to see how reducing spending in one area—the cash payment to people not on the insurance plan—can increase spending overall. I might have been more successful convincing the college budget committee not to eliminate the cash payment if they had understood the language of DAGs. Figure 12.1 shows two competing DAGs for the situation:



(a) One path from cash to expenditures

(b) Two paths from cash to expenditures

Figure 12.1: Two different DAGs relevant to the debate about eliminating the cash payment to employees not on the college's health care plan.

The people on the budget committee saw clearly the direct link between the cash payment and total expenditures and likely would not have disputed a direct link between enrollment and expenditures. But they didn't imagine a link between the cash payment and enrollment. I did, because I knew of several colleagues who used their spouse's companies insurance plan, even though it was identical to the college's plan.

The situation with the drug aprotinin is similar.

(a) One path from cash to expenditures

(b) Two paths from cash to expenditures

Figure 12.2: Two different DAGs relevant to the link between aprontinin and mortality.

LOOK AT SOME DAGs to show what happens when you include a covariate: which links you study.

13 Non-causal correlation

Prof. Danny Kaplan
November 17, 2022

Note

As you know, people are encouraged to get vaccinated before flu season. This recommendation is particularly emphasized for older adults, say, 60 and over.
The benefits of the flu vaccine have been extensively studied. It's been found based on medical records that older adults who are vaccinated have a lower mortality rate than unvaccinated older adults. This is certainly a **correlation** between vaccination and (lower) mortality, but is there necessarily a causal connection.
In 2012, the *Lancet*, a leading medical journal, published a systematic examination and comparison of many previous studies. (Such a study of earlier studies is called a *meta-analysis*.) The *Lancet* article describes a hypothesis that existing flu vaccines may not be as effective as was originally found.

> *A series of observational studies undertaken between 1980 and 2001 attempted to estimate the effect of seasonal influenza vaccine on rates of hospital admission and mortality in [adults 65 and older]. Reduction in all-cause mortality after vaccination in these studies ranged from 27% to 75%. In 2005, these results were questioned after reports that increasing vaccination in people aged 65 years or older did not result in a significant decline in mortal-*

> *ity. Five different research groups in three countries have shown that these early observational studies had substantially overestimated the mortality benefits in this age group because of unrecognized confounding. This error has been attributed to a healthy vaccine recipient effect: reasonably healthy older adults are more likely to be vaccinated, and a small group of frail, undervaccinated elderly people contribute disproportionately to deaths, including during periods when influenza activity is low or absent.*

13.1 Causality & Correlation

Causality is about relationships among entities in the world, e.g. the immunological properties of the drug acetaminophen lead to a reduction in fever. Correlation is about relationships that are evident in data, which might or might not be due to direct causal connections. For example, people who take acetaminophen tend to have fever, but this is not because acetaminophen causes fever. Instead, people who are unwell, and perhaps have fever, are more likely to take acetaminophen than those who are asymptomatic.

Correlations are properly part of the evidence to support a claim or quantification of causation. Indeed, whenever there is a correlation between two variables, it's likely that there is some chain of causal connections that links the two variables, even if that chain is not directly from one variable to the other. For instance, taking the flu vaccine is correlated with reduced mortality. Some of this correlation is due to the immunological properties of the vaccine itself. But some of the correlation results from healthy people being more likely to take the vaccine than sick people, and healthy people having a lower mortality than sick people.

Seen as a pessimist, this chapter can help you understand some of the ways that correlations can be present without a direct causal pathway, and how you can be badly mislead if you rely purely on data without any causal theory of the way your system works in the real world.

Seen as an optimist, this chapter is about ways of calculating effect sizes from data that allow you to incorporate knowledge of the causal connections amongst the variables in your data.

The field of statistics comprises both optimists and pessimists. Perhaps to oversimplify, the pessimists think the proper domain of statistics is data and stylized mathematical models, and ought not include speculative notions of causal connections in the real world. The only sort of causal connection that the pessimists will accept is that of the experimenter who *sets* the values of inputs, for example by giving one "**treatment**" group of patients a drug and another "**control**" group a "**placebo**". This has been a highly productive attitude in statistics, resulting in the development of clever designs for experiments that give the most information with the least laboratory effort. Unfortunately, the no-causation-without-experimentation philosophy leaves us without recourse when working with a system where a controlled experiment is not feasible.

Perhaps the outstanding historical example of the limits of the no-causation-without-experimentation philosophy relates to the health effects of smoking. Nowadays, the morbidity and mortality caused by tobacco smoking is mainstream knowledge. Among the other proofs of the causal relationship is the decline in mortality due to lung cancer amoung populations where smoking became much less popular. Until the mid 1960s, however, some statisticians were in the vanguard of challenging the idea of a causal connection between smoking and, e.g., lung cancer. Notably, Ronald Fisher, generally considered to be the leading statistical figure of the 20th century, vehemently and influentially criticized the evidence for the causal connection.

The optimists, again to oversimplify, believe it is possible to make useful statements (e.g. "the class helped" in Figure 13.1) about the causal connections that underlie data. They emphasize that statistics can support decision making even when knowledge of causation is incomplete and uncertain.

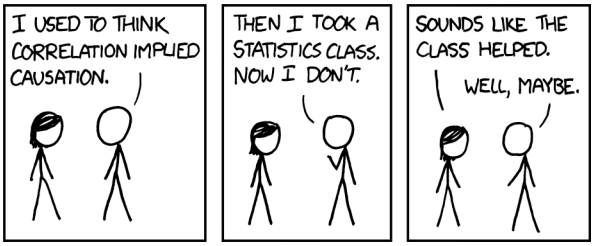


Figure 13.1: Insisting that "correlation is not causation" can interfere with making useful judgements, as interpreted by Randall Munroe in his XKCD cartoon series.

The optimists and the pessimists use the same set of mathematical and statistical tools for data analysis, particularly the calculation of effect sizes. The difference between them is the range of legitimate conclusions that can be drawn. The pessimists place in the center the idea that "correlation is not causation" and that *only* controlled experiment can be a justification for making causal conclusions. (We'll study experiment in Lesson 32.) The optimists also see the difference between correlation and causation: correlation is a mathematical property, causation is a physical one. And the optimists accept that controlled experiment is an excellent way to form strong conclusions. But they accept other sources of knowledge or theoretical speculations as potentially useful, and use effect-size calculations in a way that, contingent on that knowledge or speculation, creates through the process of data analysis situations analogous to those created in the laboratory by careful experimentation.

13.2 Old stuff

Interest in data often stems from a desire to anticipate the consequences of an intervention. Is a new polio vaccine effective? Will increasing the consumption of organic food improve health generally? Does giving bed nets to poor people in malaria-prone regions reduce the incidence of malaria?

The previous chapters introduced techniques for modeling a response variable as a function of explanatory variables. Each model is a machine for turning inputs into outputs. Change the input and the output will change correspondingly. But this does not mean that nature works in the same way. Changing an input in the real world – administering polio vaccine, eating organic food, providing bed nets to the poor – may not *cause* the same change in the response variable as happens when you change the input to a model.

*The key word here is "**cause**."*

Statisticians are careful to distinguish between two different interpretations of relationship: **"correlation"** and **"causal."** Every successful prediction model `Y ~ X` is a demonstration that there is a correlation between the response Y and the explanatory variable X.[1] But the performance of the model does not itself tell us that X *causes* Y in the real world. There are other possible configurations that will produce a correlation between X and Y. For instance, both X and Y may themselves have a common cause C without X being otherwise related to Y. In such a circumstance, a real-world intervention to change X will have no effect on Y. To put this in the form of a story, consider that the start of the school year and leaves changing color are correlated. But an intervention to start the school year in mid-winter will not result in leaves changing color. There's a common cause for the school year and colorful folliage that produces the relationship: the end of summer.

This chapter considers simple networks of causality involving three variables, generically called X, Y, and C. Always, we'll imagine that the modeler's interest is in anticipating how an intervention to change X will create to a change in Y. To accomplish this, the modeler has two basic choices for structuring a model, either

1. `Y ~ X`, or
2. `Y ~ X + C`.

It's surprising to many people that models (1) and (2) can have utterly different, even contradictory implications for how

[1] "Successful" means that the prediction performance of the model is better than the performance of a no-input model.

a change in model input X will produce a change in the model output Y. To the modeler trying to capture how the real world works, there's a fundamental choice to be made between using model (1) or model (2) to anticipate the consequences of a real-world intervention on X.

Consider this hypothesis: "It's harder to learn to drive as you move into your 20s." The hypothesis might or might not be true. The way such hypotheses are formed is often by anecdote. Say, you're having dinner when the conversation turns to a friend who has been learning to drive in her 30s. She explains that even after taking many lessons last year, she failed her driving test twice. Others at the table, who started driving in their teens, learned in a much shorter amount of time.

The hypothesis suggests a practical recommendation: It will be easier to learn to drive when younger, so better to start young. Such recommendations to take an action are always rooted in causality: starting young will *cause* you to have less difficulty learning to drive. A diagram, or *graphical causal model*, representing the causal hypothesis is seen in Figure 13.2.

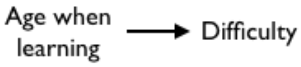


Figure 13.2: A simple graphical causal model expressing the hypothesis that age when learning to drive is the causal factor for the difficulty of learning to drive.

The direction of the arrow in fig-learn-to-drive-1 is a crucial feature. The diagram asserts that a person's age when learning to drive *causes* difficulty, as opposed to the other way around.

So, does learning to drive when young make it easier to succeed? You might collect some data, perhaps a survey asking people at what age (if ever) they learned to drive (X) and how difficult it was (Y). Suppose you build a successful model Y ~ X. This establishes a correlation X and Y (and vice versa), confirming the dinner table anecdote.

If you were undertaking serious study of the hypothesis, you should consider how other factors might influence the situation. For instance, it might be that people who learn to drive in

their 30s were more anxious about driving when in their teens. That anxiety is why they didn't learn in their teens. The anxiety might also influence the drivers perception of the difficulty learning. Such a situation is expressed in the graphical causal models in Figure 13.3.

Figure 13.3: Two possible graphical causal models of the hypothesis, "Age causes difficulty learning to drive," which do not incorporate a direct link from age to difficulty learning. Left: Anxiety itself leads to people deferring learning to drive, and to increased difficulty when learning. Right: Age is causally linked to difficulty, but the issue is really the diminished support available to older learners.

Another possibility is that older learners have busier lives and less support for learning to drive. (Figure 13.3(right)) It's harder for older learners to schedule opportunities to practice or to find car owners who can help them learn.

There is nothing inevitable about graphical causal networks. We can, and often do, intervene in ways that alter the causal flow. For example, Figure 13.4 shows the network when a later-in-life friend steps in to support a mature student.

Figure 13.4: Intervening in a system can change the structure of the graphical causal network. Here, a friend has stepped in to provide the support needed to learn to drive, severing the link that previously connected age when learning to support. (That link – now severed – reflects the kind of learning support often available to teenage students of driving, but not to older learners.)

Causal Caution

Just because you've calculated an effect size doesn't mean that you have captured any sort of causal relationship between the variables. To illustrate, use `dag01` and fit two different models: `y ~ x` and `x ~ y`.

```
Sample <- sample(dag01, size=500)
lm(y ~ x, data = Sample)
```

```
Call:
lm(formula = y ~ x, data = Sample)

Coefficients:
(Intercept)            x
      4.064        1.480
```

```
lm(x ~ y, data = Sample)
```

```
Call:
lm(formula = x ~ y, data = Sample)

Coefficients:
(Intercept)            y
    -1.9163       0.4709
```

You can't tell from these coefficients whether `x` causes `y` or vice versa (or something entirely different). The words "correlation" or "association" are used when we don't want to claim that there is a causal connection. Many statisticians will only use those words unless the data come from an **experiment**.
We're going to use causal language ("relationship", "effect," etc.) because that is often the matter of concern to decision making. But using language doesn't doesn't make the connection causal.

14 Experiment and random assignment

Prof. Danny Kaplan
November 17, 2022

In its everyday meaning, the word "experiment" is similar in meaning to the word "experience." As a verb, to experiment means to "try out new concepts or ways of doing things." As a noun, an experiment is a "course of action tentatively adopted without being sure of the outcome: the farm is an ongoing experiment in sustainable living." Both quotes are from the Oxford Dictionaries, which provides examples of each: "the designers experimented with new ideas in lighting" or "the farm is an ongoing experiment in sustainable living."

From movies and other experiences, people associate experiments with science. Indeed, one of the dictionary definitions of "experiment" is: "a scientific procedure undertaken to make a discovery, test a hypothesis, or demonstrate a known fact."

Almost all the knowledge needed to perform a scientific experiment relates to the science itself: what reagents to use, how to measure, say, the concentration of a neurotransmitter, how to administer a drug safely, and so on. This is why people who carry out scientific procedures are trained primarily in their area of science.

> **Note**
>
> In many parts of the world, malaria is a major cause of disability and death. Economists who study ways to relieve poverty have a simple, plausible theory: reducing the

effect of illnesses such as malaria will have an impact on poverty rates, since healthier people are more productive and reduced uncertainty can help them amass capital to invest to increase production further.
There are many possible ways to reduce the burden of malaria. Vaccination (although effective vaccines have been hard to develop), insect control using pesticides (which can cause environmental problems), etc. One simple intervention is the use of bed nets; screen nets deployed at night by draping over the bed and its occupant. Still, there are reasons why distributing bed nets may not be effective; people might use them incorrectly or for other purposes such as fishing. People might not be able to afford them, but giving them away might signal that they have no value.
To find out, try it: do an experiment. For instance, run a trial program where nets are given away to everyone in an area and observed whether and to what extent rates of malarial illness go down.
Such a trial is certainly and experiment. But it may not be the best way to get meaningful information.

To understand the contribution that statistical thinking can make to experiment, recall our earlier definition:

> *Statistic thinking is the explanation/description of measured variation* in the context of *what remains unexplained/undescribed.*

A key concept that statistical thinking brings to experiment is the idea of **variation**. Simply put, a good experiment should involve some variation. The simplest way to create variation is to repeat each experimental trial multiple times. This is called "**replication**."

i Example: Replicated bed net trials

One way to improve the simple experiment bed net described above is to carry out many trials. One reason is that the results from any single trial might be shaped by

accidental or particular circumstances: the weather in the trial area was less favorable to mosquito reproduction; another government agency decided to help out by spraying pesticides broadly, and so on. Setting up trials in different areas can help to balance out these influences.

Replicated trials also allow us to estimate the size of the variability caused by the accidental or particular factors. To illustrate, suppose a single trial is done and the rate of malarial illness goes down by 5 percentage points. What can we conclude? The result is promising but we can't rule out that it occurred because of accidental factors other than bed nets. Why not? Because we have no idea how much unexplained variation is in play.

In contrast, suppose four trials at different sites are done, showing reductions by 5, 8, 2, and -1 percentage points. (Reduction by a negative number, like -1, is an *increase*.) Now we know something about the amount of variation due to accidental, site-to-site factors. The replication introduces *observed* variation in results, the observed variation can be quantified and used to place the overall trend in context.

Common sense correctly suggests summarizing the results of the four trials by their mean: a reduction in the rate of malarial diseases of $\frac{5+8+2-1}{4} = 3.5$ percentage points. Statistical thinking tells us that this exact number, 3.5, is somewhat of an accident. This is because the four numbers tell us that there is site-to-site variation that remains unexplained by the bed nets: the residual variation. The residual variation gives us a handle on the amount of sampling variation to expect, the variation in results if we had collected a different sample of trials from different sites or at different times.

As usual, we quantify the sampling variation by a "standard error," which in turn gets translated into a "margin of error," and transformed yet again into an interval estimate. Here, that interval is 3.5 ± 4.5. Generations of statistics students have learned how to carry out the standard-error/margin-of-error/confidence-interval calculations and how to interpret them. 3.5 ± 4.5 means that

> the results from our trials are entirely consistent with the bed nets having zero effect on rates of malarial illness. Put another way, the replication of trials—the $n = 4$ trials in this example—provides us a way to quantify the amount of noise in our results. Here, the observed 3.5 percentage point reduction cannot be distinquished from noise so we have no confidence that we have seen a signal.

Replication is a comparatively modern idea. Experiments go back into pre-history. For instance, a biblical example is [DAVID's experiment as described by Judea Pearl.]

Galileo, taken by many historians to mark the appearance of "science," didn't replicate his trials. He had no clear knowledge of the idea of "signal" versus "noise," let alone ……

GALILEO didn't do replication.

Statistics contributes to experiment in two, very different ways. The first is that statistical methods are used to summarize the measurements made during the experiment. For instance, if your experiment involves

Go back to definition of statistics: explanation/description of variation in the context of what remains unexplained.

Purpose of a "scientif

14.1 From SM2

One of the most important ideas in science is "**experiment**". In a simple, ideal form of an experiment, you cause one explanatory factor to vary, hold all the other conditions constant, and observe the response. A famous story of such an experiment involves Galileo Galilei (1564-1642) dropping balls of different masses but equal diameter from the Leaning Tower of Pisa.[1] Would a heavy ball fall faster than a light ball, as theorized by Aristotle 2000 years previously? The quantity that Galileo

[1] The picturesque story of balls dropped from the Tower of Pisa may not be true. Galileo did record experiments done by rolling balls down ramps.

varied was the weight of the ball, the quantity he observed was how fast the balls fell, the conditions he held constant were the height of the fall and the diameter of the balls. The experimental method of dropping balls side by side also holds constant the atmospheric conditions: temperature, humidity, wind, air density, etc.

Of course, Galileo had no control over the atmospheric conditions. By carrying out the experiment in a short period, while atmospheric conditions were steady, he effectively held them constant.

Today, Galileo's experiment seems obvious. But not at the time. In the history of science, Galileo's work was a landmark: he put *observation* at the fore, rather than the beliefs passed down from authority. Aristotle's ancient theory, still considered authoritative in Galileo's time, was that heavier objects fall faster.

The ideal of "holding all other conditions constant" is not always so simple as with dropping balls from a tower in steady weather. Consider an experiment to test the effect of a blood-pressure drug. Take two groups of people, give the people in one group the drug and give nothing to the other group. Observe how blood pressure changes in the two groups. The factor being caused to vary is whether or not a person gets the drug. But what is being held constant? Presumably the researcher took care to make the two groups as similar as possible: similar medical conditions and histories, similar weights, similar ages. But "similar" is not "constant."

14.2 DAG interpretation of experiment

Albert Einstein is reputed to have said:

> *A theory is something nobody believes, except the person who made it. An experiment is something everybody believes, except the person who made it.*

A graphical causal network is a kind of theory. As a theory, it's natural for people to be skeptical about results stem from

the theory. Experiments are more persuasive. Let's consider what an experiment looks like when represented by a graphical causal networks.

In an experiment, you have some real-world system and a means to intervene physically on at least one of the variables in that system and to read out the response of the system to the intervention. You don't necessarily know much about the actual structure of the real world system. In Figure 14.1 the real-world system is shown in the rounded box. The intervention is on X and the output is Y.

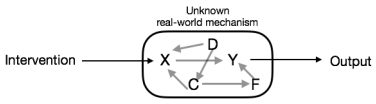


Figure 14.1: An experiment is a system in which there is an intervention and an output.

Note that in Figure 14.1 X is, potentially, affected by other variables in the system.

Ideally, the experiment is set up to eliminate all other effects on X except the intervention as in Figure 14.2. And the intervention is done in a way that none of the variables in the system can have any effect on it, for instance by assigning the intervention using a **computer random-number generator**. The lovely thing about this configuration is that the correct model to capture the effect of `X` on `Y` is simply `Y ~ X`. Whatever different people might believe about the real-world mechanism doesn't matter. The correct model is always `Y ~ X`. This is why Einstein's statement, "An experiment is something everybody believes," is justified.

But there is another part to Einstein's statement: "... except the person who made it." Why shouldn't the experimenter believe her own experiment? The experimenter might know that she didn't or couldn't conduct an ideal experiment. She wasn't actually able to eliminate the arrows D → X and C → X. The other variables in the system might also be influencing X as in Figure 14.1. In this situation, the right model may not be Y ~ X. In fact, for the particular system shown in Figure 14.1 the correct model would be Y ~ X + C + D. But how could the

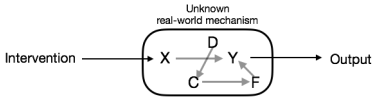


Figure 14.2: An ideal experiment is one where the *only* influence on X is the intervention. Any effect on X or the intervention of the other variables in the system has been eliminated. The input paths to X from C and D that appear in Figure 14.1 have been deleted by the experimenter. This is not always possible in practice.

experimenter know this for sure if she didn't know all about the real-world mechanism?

It turns out that for either of the causal systems in Figures 14.1 there is always a correct model to show the link between the intervention and output: Output ~ Intervention. Rather than modeling the output by the physical quantity X, model the output by the random numbers generated by the computer that were used to set the intervention. This modeling approach is called **intent to treat**.

Typically, experiments are done using a specially constructed system that is thought to resemble the system on which the intervention will actually be done. Insofar as the experimental system does resemble the real-world system, the experimental results will anticipate the effect of the real-world intervention. But often it's hard to establish that the experimental system is a match to the system on which the real-world intervention will be applied. As such, subjective belief is still a factor in accepting that the experiment will be informative about the real-world systems we work with.

It's appropriate to show some humility about models and recognize that they can be no better than the assumptions that go into them. Useful object lessons are given by the episodes where conclusions from modeling (with careful adjustment for covariates) can be compared to experimental results. Some examples (from (**freedman-editorial-2008?**)):

- Does it help to use telephone canvassing to get out the vote? Models suggest it does, but experiments indicate otherwise.
- Is a diet rich in vitamins, fruits, vegetables and low in fat protective against cancer, heart disease or cognitive decline? Models suggest yes, but experiments generally do not.

The divergence between models and experiment suggests that an important covariate has been left out of the models.

15 Measuring and accumulating risk

Prof. Danny Kaplan
November 17, 2022

A probability—a number between 0 and 1—is the most used measure of the chances that something will happen, but it is not the only way nor the best for all purposes.

Also part of everyday language is the word "odds," as in, "What are the odds?" to express surprise at an unexpected event.

Odds are usually expressed in terms of two numbers, as in "3 to 2" or "100 to 1", written more compactly as 3:2 and 100:1 respectively. The setting for odds is an even that might happen or not: the horse Fortune's Chance might win the race, otherwise not; it might rain today, otherwise not; the Red Sox might win the World Series, otherwise not.

The format of a *probability* assigns a number between 0 and 1 to the chances that Fortune's Chance will win, or that it will rain, or that the Red Sox will come out on top. If that number is called p, then the chances of the "otherwise outcome" must be $1 - p$. The event with probability p would be reformatted into odds as $p : (1 - p)$. No information is lost if we treat the odds as a single number, the result of the division $p/(1 - p)$. Thus, when $p = 0.25$ the corresponding odds will be $0.25/0.75$, in other words, $1/3$.

A big mathematical advantage to using odds is that the odds number can be anything from zero to infinity; it's not bounded

within 0 to 1. Even more advantageous for the purposes of accumulating risk is the logarithm of the odds, called "**log odds**." We will come back to this later.

15.1 Staying in bounds

> ⚠ Note in draft
>
> Maybe move this to a earlier lesson. Not clear.

The *linear models* (`lm()`) we have mostly been using up until now accumulate the model output as a linear combination of model inputs. Consider, for instance, a simple model of fuel economy based on the horsepower and weight of a car:

```
mpg_mod <- lm(mpg ~ hp + wt, data = mtcars)
mpg_mod %>% coefficients()
```

```
(Intercept)          hp          wt
37.22727012 -0.03177295 -3.87783074
```

These coefficients mean that the model output is a **sum**. For instance, a 100 horsepower car weighting 2500 pounds has a predicted fuel economy of `37.2 - 0.032*100 - 3.88*2.5`=24.3 miles per gallon.[1] If we're interested in making a prediction, we often hide the arithmetic behind a computer function, but it is exactly this arithmetic:

```
mod_eval(mpg_mod, hp = 100, wt = 2.5)
```

```
  hp  wt model_output
1 100 2.5      24.3554
```

[1] The `wt` variable is measured in units of 1000 lbs, so a 2500 pound vehicle has a `wt` value of 2.5.

The arithmetic, in principle, let's us evaluate the model for any inputs, even ridiculous ones like a 10,000 hp car weighing 50,000 lbs. There is no such car, but there is a model output.[2]

```
mod_eval(mpg_mod, hp=10000, wt = 50)
```

```
     hp wt model_output
1 10000 50    -474.3937
```

The prediction reported here means that such a car goes *negative* 474 miles on a gallon of gas. That's silly. One way to deal with such silliness is to restrict the inputs to "reasonable" values.

Often, a better way to avoid the silliness is to structure the model so that unreasonable outputs—such as negative miles per gallon—cannot happen. Figuring out how to do this draws on mathematical experience. In this case, modeling the *logarithm* of mpg means that a numerically negative output still corresponds to a positive mpg. If we want the model output denominated in miles-per-gallon rather than logarithmic units, we just need to exponentiate (`exp()`) the logarithmic output to return to the world of miles-per-gallon:

```
mod_logmpg <- lm(log(mpg) ~ hp + wt, data = mtcars)
mod_eval(mod_logmpg, hp=10000, wt=50) %>%
  mutate(model_mpg = exp(model_output))
```

```
     hp wt model_output   model_mpg
1 10000 50     -21.6327 4.02753e-10
```

This "trick" of modeling the logarithm of output keeps the model output **in bounds** so far as `mpg` is concerned. There will never be a negative mpg output.[3] ## Modeling log odds

[2] A 10,000 hp, 50,000 lbs ground vehicle does have a name: a "tank." Common sense dictates that one not put too much stake in a calculate of a tank's fuel economy based on data from cars!

[3] That does not fix the absurdity of modeling tanks based on the fuel economy of cars.

When a model output is intended to be interpreted as a probability, we have a similar problem. THATS what the LOG-ODDS transformation DOES. WITH LOG-ODDS we can model probability using linear combinations of inputs.

15.2 Probability as prediction?

DOES IT MAKE SENSE TO FRAME a PREDICTION IN TERMS OF A PROBABILITY? So long as the probability is not exactly zero or one, observing either of the two kinds of events—e.g., yes/no, alive/dead, diseased/healthy—does not contradict the model output. So how can we judge if a model is on-target or not. Or, equivalently, how can we decide which of two models is better.

In Lesson 26 we introduced the idea of a prediction interval OUTPUT SHOULD ALMOST ALWAYS (95% of the time) be within the interval.

The problem for us now is to create something like the PREDICTION INTERVAL when the model output is a probability.

INTRODUCE the variance as a measure of uncertainty. The variance of a probability is $p(1-p)$.

i Example: A bookies' calculations [NEEDS FIXING]

The most familiar use of "odds" is in gambling. For instance, a famous song lyric puts the odds of Valentine winning the horse rate "at five to nine." Less musically, this odds is $5/9 = 0.5555$, but the two-number format makes particular sense for keeping track of bets. Five-to-nine describes a bet of one unit. The second number, 9, specifies the amount the gambler is staking on the outcome. On a loss, the gambler loses that stake. On a win, the gambler gets back the stake and, in addition, gets the amount specified by the first number. So a winner at five to nine would leave the racetrack with an extra \$5. But on a loss, the gambler leaves \$9 behind.
A "bookie" is someone who provides a service. You can

go to a bookie to lay a bit. In drama, this might be done by telephone: “Lay $90 on Valentine” is all the gambler needs to communicate. No money has to change hands. On a win, the bookie will return $50 to the gambler. On a loss, the gambler has a debt of $90.

A bookie is not a gambler; he’s an accountant who records numbers. The bookie arranges these numbers so that he makes money. To see this, imagine a horse race including Valentine, Paul Revere, and Epitaph. To start, the bookie specifies odds on each possible outcome, say 5:9 for Valentine, 1:3 for Paul Revere (a favorite!), and 1:2 on Epitaph.

If the bookie has a good nose, about a third of the stakes will be bet on each outcome. If not, as new bets come in the bookie raises or lowers the odds to encourage or discourage bets so that the roughly one-third of stakes are placed on each outcome. Suppose at the end of the day that $500 is staked on each of the three outcomes.

WRONG WRONG WRONG. It needs to work that the winning returned for Valentine has to be less than the stakes on the other horses, and similarly for all horses. So if $100 is bet on Valentine we need $100 staked on the other horses.

Added up, these odds are $5 + 1 + 1 = 7$ on the top and $9 + 2 + 1 = 12$ on the bottom. It’s important—for the bookie—that the odds are arranged so that the bottom number is larger than the top number: 12 is larger than 11. Note that this method of adding is simpler than combining fractions. To add the fractions 1/2 and 1/3 gives 5/6. But to combine the odds $1 : 2$ and $1 : 3$ gives $2 : 5$. One more detail is needed for a real-life bookies, taking into account the size of each bet. For instance, a $5 bet at 5:9 would be recorded as 25:45.

Now the race is run. The winner is … well … from the bookie’s point of view it doesn’t matter who wins.

15.3 "Irrationality"

[From *The Model Thinker*, p. 52]

Gain Framing: You have two options

Option A) Win $400 for certain

Option B) Win $1000 if a fair coin comes up heads and $0 if tails

Loss Framing: You are given $1000 and have two options:

Option a) Lose $600 for certain

Option b) Lose $0 if a fair coin comes up heads and lose $1000 if tails.

Hyperbolic discounting: see pp 52-43

"Prospect theory", Kahneman and Tversky (1979) "Prospect theory: an analysis of decisions under risk," *Econometrica* 47(2):263-291 link to paper

> **i** Example
>
> A subtle modification to the linear model architecture allows the modeller to guarantee that the output will be between zero and one. The modified architecture, called "**logistic regression**", is therefore well suited to modeling categorical response variables, where the model output will be interpreted as a probability.
> **?@fig-w-logistic** shows a logistic model of survival as a function of age and smoking status. Notice that in the logistic model, the effect of smoking on survival is negative, particularly for people around age 50. The logistic architecture provides an intrinsic flexibility which avoids the undue influence of the very young and very old, for whom survival is close to 100% or 0 respectively *regardless* of smoking status.
> :::: {.callout-warning} The figure fig-w-logistic is not compiling in PDF mode

Example: Fraction attributable

US Federal law forbids employment discrimination based on age. (There are some exceptions, such as air-traffic controllers, whose mandatory retirement age is 56). In a discrimination lawsuit, data on who was and who was not laid-off was used to construct a model of the probability of layoff. The effect size is, as usual for a probability model, expressed in log odds.

- baseline: risk of 20%, so log odds of of -1.4.
- age over 50, add log odds of 1 $\pm$ 0.3
- software engineer, subtract log odds of 0.5 $\pm$ 0.25
- paid different from company average, subtract log odds of 0.2 $\pm$ 0.1 per $10,000 high than company average.

These estimates come from a logistic regression model `laid_off ~ over50 + software_engineer + pay_above_average`.

1. For a laid-off employee over 50, what is the fraction attributable to age?

Solution

The baseline risk of being laid off is 20%. For the employee aged over 50 years, the log odds of the risk is -1.4 + 1 $\pm$ 0.3, or -0.7 to -0.1. Translating these log odds into probabilities gives a risk of 33% to 47%, with the range reflecting the uncertainty in the effect size from the model. The estimated relative risk (risk ratio) for the employees over 50 ranges from 33/20 to 47/20, that is, from 1.65 to 2.35. The attributable fraction is $(RR - 1)/RR$ and therefore ranges from (1.65 - 1)/1.65 to (2.35 - 1) / 2.35 or 40% to 57%.

2. What fraction of all layoffs can be attributed to age over 50? (Population attributable fraction.) Assume that one-third of the employees are over 50.

Solution

16 Constructing a classifier

Prof. Danny Kaplan
November 17, 2022

We all face many yes/no situations. A patient has a disease or does not. A credit card transaction is genuine or fraudulent. A **classifier** is a statistical model designed to *predict* the unknown outcome of a yes/no situation from information that is already available.

Consider this news report:

> Higher vitamin D intake has been associated with a significantly reduced risk of pancreatic cancer, according to a study released last week. Researchers combined data from two prospective studies that included 46,771 men ages 40 to 75 and 75,427 women ages 38 to 65. They identified 365 cases of pancreatic cancer over 16 years. Before their cancer was detected, subjects filled out dietary questionnaires, including information on vitamin supplements, and researchers calculated vitamin D intake. After statistically adjusting for [that is, holding constant] age, smoking, level of physical activity, intake of calcium and retinol and other factors, the association between vitamin D intake and reduced risk of pancreatic cancer was still significant. Compared with people who consumed less than 150 units of vitamin D a day, those who consumed more than 600 units reduced their risk by 41 percent. - New York Times, 19 Sept. 2006, p. D6.

There are more than 125,000 cases in this study, but only 365 of them developed pancreatic cancer. If those 365 cases had been scattered around dozens or hundreds of groups and analyzed separately, there would be so little data in each group that no pattern would be discernible.

16.1 School spending example

16.2 Example: Covariates and context in educational outcomes

To illustrate how covariates set context, consider an issue of interest to public policy-makers in many societies: How much money to spend on children's education? In the United States, for instance, educational budget policy is set mainly on a state-by-state level. State lawmakers are understandably concerned with the quality of the public education provided, but they also have other concerns and constraints and constituencies who give budget priority to other matters.

In evaluating the various trade-offs they face, lawmakers would be helped by knowing how increased educational spending will shape educational outcomes. What can available data tell us? Unfortunately, there are various political constraints that work against states adopting and publishing data on a common measure of genuine educational outcome. Instead, we have high-school graduation rates, student grades, etc. These have some genuine meaning but also can reflect the way the system is gamed by administrators and teachers and which cannot be easily compared across states. At a national level, we have college admissions tests such as the ACT and SAT. Perhaps because these tests are administered by private organizations and not state governments, it's possible to gather data on test-score outcomes on a state-by-state basis and collate these with public spending information.

Figure 16.1 shows average SAT score in 2010 in each state versus expenditures per pupil in public elementary and secondary schools. Laid on top of the data is a flexible linear model (and

its confidence band) of SAT score versus expenditure. The overall impression given by the model is that the relationship is negative, with lower expenditures corresponding to higher SAT scores. But the confidence bands are broad and it is possible to find a smooth path through the confidence band that has almost zero slope. Either way, the conventional wisdom that higher spending produces better school outcomes is not supported by this graph.

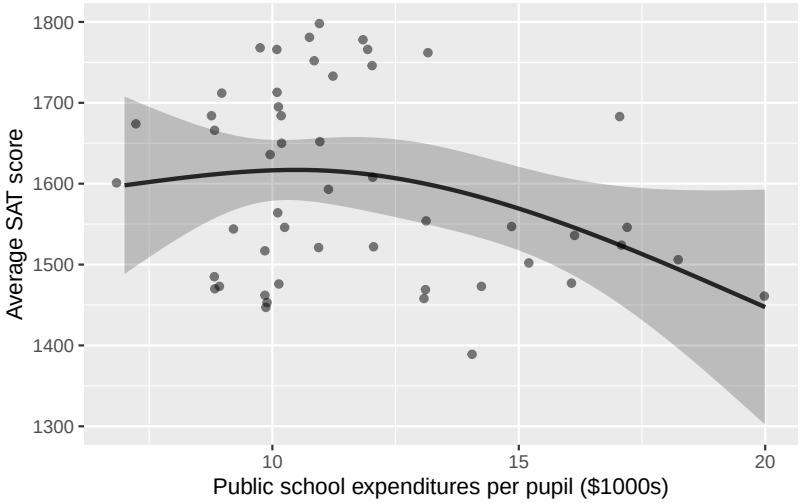


Figure 16.1: State by state data (from 2010) on average score on the SAT college admissions test and expenditures for public education.

There are other factors that play a role in shaping education outcomes: poverty levels, parental education, how the educational money is spent (higher pay for teachers or smaller class sizes? administrative bloat?), and so on. Modeling educational outcomes solely by expenditures ignores these other factors.

At first glance, it's tempting to ignore these additional factors. We may not have data on them. And insofar as our interest is in understanding the relationship between expenditures and education outcomes, we are not directly concerned with the additional factors. This lack of direct concern, however, doesn't imply that we should totally ignore them but that we should do what we can to "hold them constant".

To illustrate, let's consider a factor on which we do have data:

the fraction of eligible students (those in their last year of high school) who actually take the test. This varies widely from state to state. In a poor state where few students go to college the fraction can be very small (Alabama 8%, Arkansas 5%, Mississippi 4%, Louisiana 8%). In some states, the large majority of students take the SAT (Maine 93%, Massachusetts 89%, New York 89%). In states with low SAT participation rates, the students who do take the test are applying to schools with competitive admissions. Such strong students can be expected to be get high scores. In contrast, the scores in states with high participation rates reflect both strong and weak students; they will be lower on average than in the low-participation states.

Putting the relationship between expenditure and SAT scores in the context of the fraction taking the SAT can be done by using fraction as a co-variate, that is, building the model `SAT ~ expenditure + fraction` rather than just `SAT ~ expenditure`. Figure 16.2) shows a model with fraction taken into account.

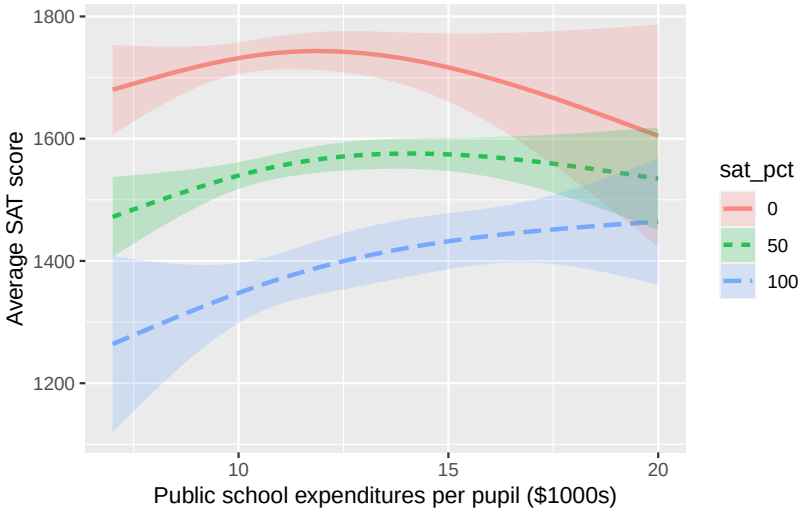


Figure 16.2: The model of SAT score versus expenditures, including as a covariate the fraction of eligible students in the state who take the SAT.

Note that the effect size of spending on SAT scores is positive when the expenditure level is less than $10,000 per pupil. And notice that when the fraction taking the SAT is near 0, the

average scores don't depend on expenditure. This suggests that among elite students, expenditure doesn't make a discernable difference: it's the students, not the schools that matter.

The relationship shown in Figure 16.1 is genuine. So is the very different relationship seen in Figure 16.2. How can the same data be consistent with two utterly different displays? The answer, perhaps unexpectedly, has to do with the connections among the explanatory variables. Whatever the relationship between each individual explanatory variable and the response variable, the *appearance* of that relationship will depend on how explanatory variables are connected to each other.

16.3 Connections among explanatory variables

To demonstrate that the apparent relationship between an explanatory variable and a response variable – for instance, school expenditures and education outcomes – depends on the connections of the explanatory variable with other explanatory variables, let's move away from the controversies of political issues and study some systems where everyone can agree exactly how the variables are connected. We'll look at data produced by simulations where we specify exactly what the connections are.

A simulation implements a hypothesis: a statement about that might or might not be true about the real world. As a starting point for our simulation, let's imagine that education outcomes increase with school expenditures in a very simple way: each $1000 increase in school expenditures per pupil results in an average increase of 10 points in the SAT score: an effect size of 0.01 points per dollar. Thus, the imagined relationship is:

$$\text{sat} = 1100 + 0.01 * \text{dollar expenditure}$$

Let's also imagine that the fraction of students taking the SAT test also influences the average test score with an effect size of -4 sat points per percentage point. Adding this effect into the simulation leads to an imagined relationship of

sat = 1100+0.01∗dollar expenditure−4∗participation percentage.

And, of course, there are other factors, but we'll treat their effect as random with a typical size of ± 50 points.

To complete the simulation, we'll need to set values for dollar expenditures and participation percentage. We'll let the dollar expenditures vary randomly from \$7000 to \$18,000 from one state to another and the participation percentage vary randomly from 1 to 100 percentage points.

Notice that in this simulation, both participation percentage and expenditures affect education outcomes, but there is no connection at all between the two explanatory variables. That is, the graphical causal network is that shown in Figure @ref(fig:school-sim-1).

```
dag_school1
```

```
[[1]]
expenditure ~ unif(7000, 18000)

[[2]]
participation ~ unif(1, 100)

[[3]]
outcome ~ 1100 + 0.01 * expenditure - 4 * participation + eps(50)

attr(,"class")
[1] "list"      "dagsystem"
```

```
dag_draw(dag_school1)
```

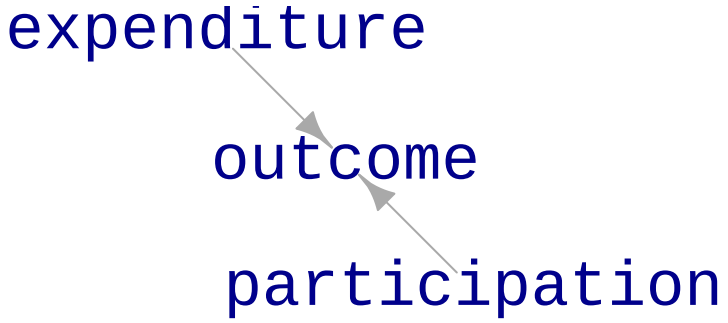


Figure 16.3: A graphical causal network relating expenditures, participation percentage, and education outcome, where there is no connection between expenditures and participation.

We can generate simulated data and use the data to train models. **?@fig-school-data-1** shows the data and two different models.

```
Dat1 <- sample(dag_school1, size=500)
mod1_1 <- lm(outcome ~ ns(expenditure,2), data = Dat1)
mod1_2 <- lm(outcome ~ ns(expenditure,2) * participation, data = Dat1)
mod_plot(mod1_1, interval="prediction") %>%
  gf_point(outcome ~ expenditure, data = Dat1)
mod_plot(mod1_2, interval="prediction") %>%
  gf_point(outcome ~ expenditure, alpha=~participation, data = Dat1, inherit=FALSE)
```

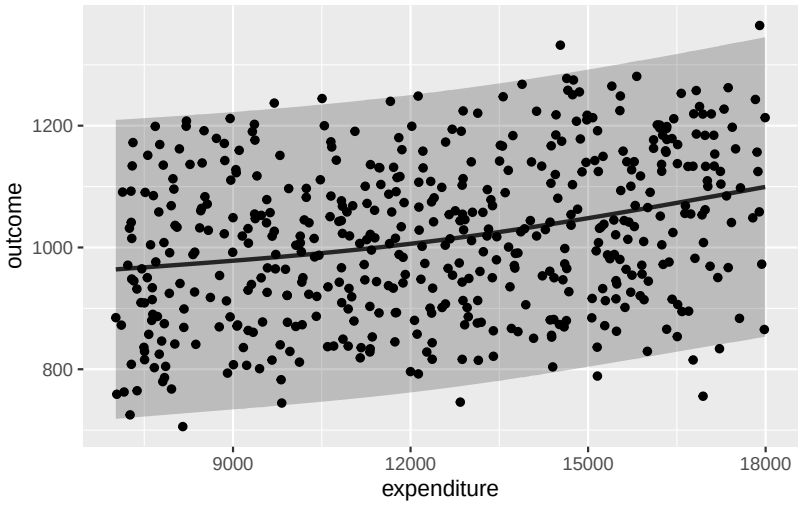


Figure 16.4: Data and models of the relationship between expenditures and education outcomes from a simulation in which expenditures and participation rate are unconnected as in Figure 16.3. - (a) The model `outcome ~ expenditure` - (b) The model with participation as a covariate: `outcome ~ expenditure + participation` Both models (a) and (b) show the same effect size for outcome with respect to expenditure.

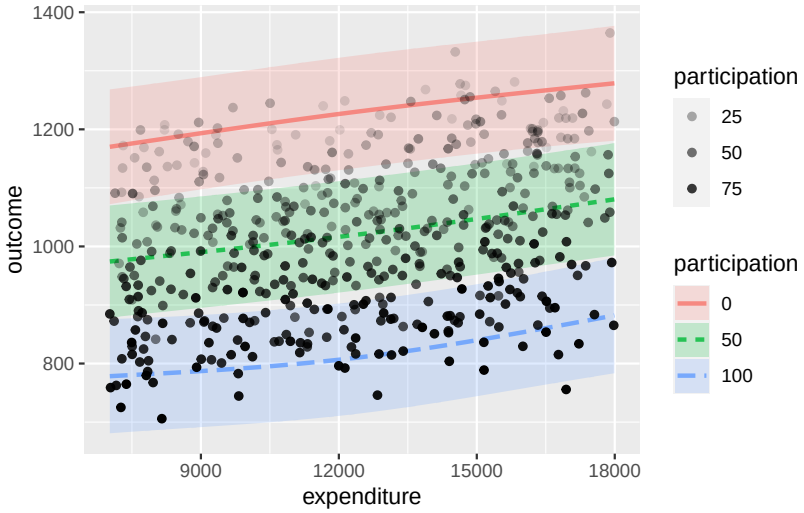


Figure 16.5: Data and models of the relationship between expenditures and education outcomes from a simulation in which expenditures and participation rate are unconnected as in Figure 16.3. - (a) The model `outcome ~ expenditure` - (b) The model with participation as a covariate: `outcome ~ expenditure + participation` Both models (a) and (b) show the same effect size for outcome with respect to expenditure.

The relationship between outcome and expenditure can be quantified by the effect size, which appears as the slope of the function. You can see that when the explanatory variables are unconnected, as in Figure 16.3, the functions have the same slope.

Now consider a somewhat different simulation. Rather than expenditures and participation being unconnected (as in the causal diagram shown in Figure 16.3), in this new situation we will posit a connection between the two explanatory variables. We'll image that there is some broad factor, labeled "culture" in **?@fig-school-sim-2**, that influences both the amount of expenditure and the participation in the tests used to measure education outcome. For instance, "culture" might be the importance that the community places on education or the wealth of the community.

```
dag_school2
```

```
[[1]]
culture ~ unif(-1, 1)

[[2]]
expenditure ~ 12000 + 4000 * culture + eps(1000)

[[3]]
participation ~ (50 + 30 * culture + eps(15)) %>% pmax(0) %>%
    pmin(100)

[[4]]
outcome ~ 1100 + 0.01 * expenditure - 4 * participation + eps(50)

attr(,"class")
[1] "list"      "dagsystem"
```

```
dag_draw(dag_school2)
```

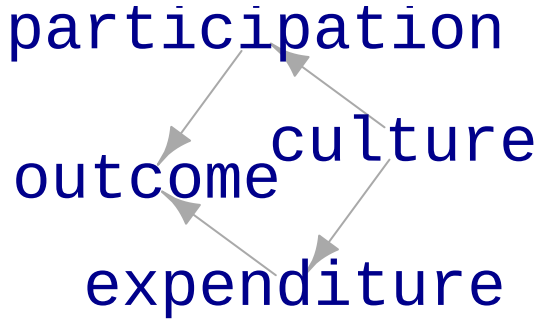


Figure 16.6: A DAG for school outcomes that links `participation` and `expenditure` as a function of `culture`.

Again, using data from this simulation, we can train models:

- (a) `outcome ~ expenditures`, which has no covariates.

- (b) `outcome ~ expenditures + participation`, which includes participation as a covariate.

?@fig-school-data-2 shows the data from the new simulation (which is the same in both subplots) and the form of the function trained on the data. Now model (a) shows a very different relationship between expenditures and outcome than model (b).

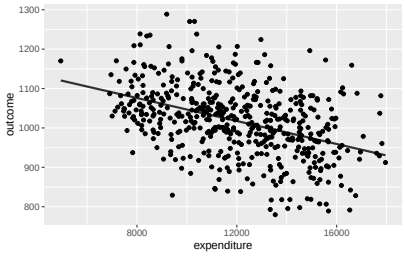


Figure 16.7: Similar to **?@fig-school-data-1** but using the simulation in which the explanatory variables – expenditure and participation – are connected by a common cause. The two models show very different relationships between outcomes and expenditures. Model (b) matches the mechanism used in the simulation, while that mechanism is obscured in model (a).

Since we know the exact mechanism in the simulation—`outcome` increases with `expenditure`—we know that model (b) matches the workings of the simulation while model (a) does not.

For the simulation where expenditure and participation share a common cause, failing to stratify on `participation` – that is, looking at the points in (**fig?**):-school-data-2 (a) but ignoring color – gives an utterly different result than if the stratification includes `participation`.

16.4 Other stuff

Consider a credit-card company might building a classifier to predict at the time of the transaction whether a purchase of

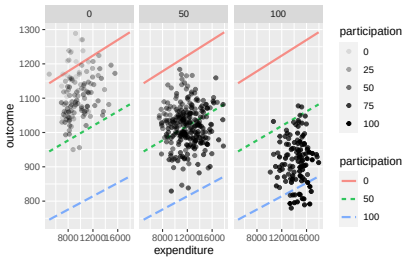


Figure 16.8: Similar to **?@fig-school-data-1** but using the simulation in which the explanatory variables – expenditure and participation – are connected by a common cause. The two models show very different relationships between outcomes and expenditures. Model (b) matches the mechanism used in the simulation, while that mechanism is obscured in model (a).

gasoline is fraudulent. The company knows how often and how much gasoline the individual cardholders buys, where the cardholder lives, whether the cardholder travels extensively, typical times of day for a purchase, and so on. **Feature engineering** is the process of using existing data—including, in our example, whether the purchase turned out to be fraudulent—to develop potential markers or signals of the outcome. For simplicity, imagine the features selected are the number of days since the last gasoline purchase and the distance from the last place of purchase.

Once potential features have been proposed, the engineers building the classifier assemble training and testing data sets. Suppose, for the purpose of illustration, that the training data has 2000 fraudulent transactions and 4000 non-fraudulent ones, and the testing set is about the same.

The word "assemble" was used intentionally to describe how the testing and training data were collected: a **case-control study**. Since the objective is to detect fraud, it is reasonable to have a lot of "yes" cases in the data. The "no" cases serve as a kind of control; they were included specifically to have balance in the data. If data had been collected as a simple random sample of credit card transactions, there would have

been many, many more "no" cases than "yes."

With such training data it is easy to build a statistical model with `Fraud` as the response variable. That model can then be evaluated on the testing data to produce a model output for each row:

Fraud	Feature 1	Feature 2	Model output
no	6 days	5 miles	-
no	1 day	250 miles	+
yes	120 days	75 miles	-
no	5 days	0 miles	-
yes	0.2 days	90 miles	+

It's understandable that a classifier may not have perfect performance. After all, it iss trying to make a prediction based on limited data, and randomness may play a role.

There are different ways of making a mistake, and these different ways have very different consequences. One kind of mistake, called a "**false positive**", involves a classifier output that's positive (i.e. the classifier indicates fraud) but which is wrong. The consequence of this sort of mistake in the present example is a customer who has to find another way to pay for gasoline.

The other kind of mistake is called a "**false negative**". Here, the classifer output is that the transaction is not fraudulent, but in actuality it was. The consequence of this kind of mistake is different: a successful theft.

The nomenclature signals that a mistake has been made with the word "false." The kind of mistake is either "positive" or "negative", corresponding to the output of the classifier.

When the classifier gets things right, that is a "true" result. As with the false results, a true result is possible both for a "positive" and a "negative" classifier output. So the two ways of getting things right are called "**true positive**" and "**true negative**".

Tabulating all 6000 rows of the testing data might produce something like this:

Fraud	test +	test -
yes	1900	100
no	50	3950

16.5 Incidence

16.6 Sensitivity and specificity

i Example: Accuracy of airport security screening

Airplane passengers have, for decades, gone through a security screening process involving identity checks, "no fly" lists, metal detection, imaging of baggage, random patdowns, and such. How accurate is such screening? Almost certainly, the accuracy is not as good as an extremely simple, no-input, alternative process: automatically identify every passenger as "not a security problem." We can estimate the accuracy of the "not a security problem" classifier by guessing what fraction of airplane passengers are indeed a threat to aircraft. In the US alone, there are about 2.5 million airplane passengers each day and security problems of any sort rarely happen. So the accuracy of the no-input classifier is something like 99.999%.
The actual screening system, using metal detectors, baggage x-rays, etc. will have a lower accuracy. We know this since it regularly mis-identifies innocent people as security problems.
The problem here is not with airport security screening, but with the flawed use of *accuracy* as a measure of performance. Indeed, achieving super-high accuracy is not the objective of the security screening process. Instead, the objective is to *deter* security problems by convincing potential terrorists that they are likely to get caught before they can get on a plane. This has to do with the *sensitivity* of the system. The *specificity* of the system, although important to the everyday traveller, is not what deters the terrorist.

17 Accounting for prevalence

Prof. Danny Kaplan
November 17, 2022

Make sure this connects to the likelihood comparison material in Lesson 36.

18 Hypothesis testing

Prof. Danny Kaplan
November 17, 2022

It's important, first of all, to know that a sensible non-technical interpretation of the phrase "hypothesis testing" describes something that is very different from its actual technical meaning. Consider the hypothesis expressed by this sentence: "Birds are the evolutionary descendants of dinosaurs." Like all hypotheses, this one is a statement that might or might not be true. Unlike many hypotheses, it is an interesting and surprising statement. For many people, the reference examples of dinosaurs are heavy, armor plated and bespiked, land-dwelling fighting giants, so different from sweetly singing, gently dashing, lightweight, feathered airborne creatures found in our backyards. Even the names suggest an irreconcilable gap: compare Tyrannosaurus Rex, Stegosaurus, and Triceratops, to jay, finch, lark, swallow, and wren. In general, we test a hypothesis by making a list of things that should be true if the hypothesis were right, for instance, similar bone layout (check), reproduction via eggs (check), nest-building (check), hollow bones (check). A hypothesis test might involve also similar lists of possible facts that contradict the hypothesis; observing such facts is evidence against the hypothesis. Framing and testing hypotheses is a central part of scientific method.

In statistics, a "hypothesis test" has a completely different flavor. First of all, the hypothesis being tested is almost always *uninteresting*, something that we would *not* be surprised at. Second, we don't look for evidence to support the hypothesis or establish its truth. Instead, the conclusion we reach from statistical hypothesis testing can never be that the hypothesis

is true or even likely. The allowed conclusions come in only two possibilities: *rejecting* the hypothesis or *failing to reject* the hypothesis. Also strange about a statistical hypothesis test: one person might collect data that lead correctly to the conclusion of *failing to reject* the hypothesis; another person might collect data that are completely compatible with the first person's, yet correctly lead to the opposite conclusion.

With such a strange and counter-intuitive structure, it's understandable that the consensus among statisticians is that few users of statistical hypothesis tests understand the correct interpretation of them. Many statisticians argue that the employment of statistical hypothesis testing in the usual ways has led to a crisis in science and a justifiable lack of trust in published scientific findings.

The three previous paragraphs might suggest that what's called "statistical hypothesis testing" is an odd topic that shouldn't be included in statistics textbooks. Yet, statistical hypothesis testing has for generations been at the center of statistics courses. Using it—even if it's not understood—is a practical necessity for scientists who want to publish their results or apply for research grants based on preliminary research.

Hypothesis testing is a difficult topic for many introductory students. Partly that's because of the large amount of terminology that's involved: Null hypothesis, test statistic, p-value, type-I and type-II errors, significance level. Partly that's because everyday language is being used to mean something that is nothing like the everyday meaning. The word "significant" is particularly abused by statistical hypothesis testing. In contemporary usage, "significant" is synonymous with "important," "consequential," "meaningful," "substantial," "momentous," and so on. Your "significant other" is a person of great importance to you. It's very good news when your doctor reports a "significant improvement" in your friend or relative's condition. In statistical hypothesis testing, "significant" can be entirely consistent with "trivial," "useless," "of no practical importance." It's unclear whether the choice of "significant" in statistics was intended to deceive, but it very often has that effect.

18.1 What people want to know

Another major reason why statistical hypothesis testing can be difficult to get your head around is that many people have an intuitive idea about what they want to know when testing a hypothesis: whether the stated hypothesis is likely to be true. But statistical hypothesis is expressly designed to avoid making any statement about the probability that a hypothesis might be true or false. There is good reason for this since "truth" is a shakey concept philosophically.

Consider this example of a hypothesis: "Drug X lowers blood pressure." People being so different one from another, a hypothesis like this would not be refuted just because X raised the blood pressure of a person. A better statement might be, "Drug X typically lowers blood pressure." Still better would be a more definite statement, "Drug X typically lowers blood pressure by around 10 mmHg." Whether such a statement is true or not depends on the meaning of "typically" and "by around."

Rather than looking for the truth or falsity of a hypotheses, statistical thinkers focus on a question in this form, "Is the statement that 'Drug X typically lowers blood pressure by around 10 mmHg' consistent with the observed facts?" Suppose, to illustrate, that the facts are the recorded change in blood pressure in 10 patients given drug X.

Consider these measurements of change in blood pressure before and after administration of the drug: -5, -1, +7, -15, -3, -6, +1, -8, +2, 0

We're seeing a reduction (a negative number) in most of the patients. The numbers are near -10, even if they are not exactly -10 all the time. When there's an increase in blood pressure, it's small.

In contrast, suppose the numbers were 5, 1, -7, 15, 3, 6, -1, 8, -2, 0. These number are inconsistent with the claim that the typical change is -10. Most of them are positive, sometimes by a lot. Of the negative numbers, none of them even reaches -10.

It would be better to have a quantitative way to measure "consistency with the observed facts." Two changes to the way we frame hypotheses will help.

1. Be more specific about "typical" and "by around." For instance, here's a very definite hypothesis: "In a group of patients with such-and-such condition, drug X lowers blood pressure by an average of -10 mmHg with a standard deviation of 7mmHg."

i Demonstration: Likelihood of the facts given the hypothesis

With such a statement we can actually do some arithmetic to find a numeric measure of consistency: something very much like the probability of seeing the observed data under the assumption that the stated hypothesis is true. For instance, the relative probability of seeing -5 from a normal distribution with mean -10 and standard deviation 7 is easily calculated in R:

```
dnorm(-5, mean = -10, sd = 7)
```

```
[1] 0.04415934
```

We can do a similar calculation for each of the "facts."

```
facts <- c(-5, -1, +7, -15, -3, -6, +1, -8, +2, 0)
dnorm(facts, mean = -10, sd = 7)
```

```
 [1] 0.044159344 0.024937582 0.002985977 0.044159344 0.034567246 0.048406848
 [7] 0.016580258 0.054712394 0.013111882 0.020542552
```

But what we really want is the probability not of each of the facts but of all of them put together. This is the product of the individual relative probabilities:

```
dnorm(facts, mean = -10, sd = 7) %>% prod()
```

```
[1] 5.936822e-17
```

This number, 5.9×10^{-17}, is called the "likelihood" of the observed facts. Obviously it is very small, but that is not an indication that the facts are unlikely under the assumption that the hypothesis is true. The smallness is a consequence of the fact that *any* particular number is an unlikely outcome of a draw from a continuous probability density. (For those familiar with calculus, the numbers calculated by `dnorm()` are not actually probabilities, they are probability densities, which we've casually called "relative probabilities.")

How to interpret a likelihood number like 5.9×10^{-17}? In order to make sense of it, we need to add another component to our measurement of "consistency with the observed facts."

2. Have one or more additional hypotheses so that we can compare likelihoods one to another. For instance, we might use the hypothesis, "In a group of patients with such-and-such condition, drug X lowers blood pressure by an average of 0 mmHg with a standard deviation of 7mmHg."

   A hypothesis like this, that stipulates zero change (on average), is called a "null hypothesis," the word "null" meaning "nothing" or "zero."

   The null hypothesis may not be of direct interest, but it provides a way for us to interpret likelihoods from the hypotheses of actual interest. We do this by comparing the numerical value of the likelihoods, often as a ratio.

i The likelihood of the null hypothesis

Calculating the likelihood of the observed facts under the assumption that the null is true is done in the same way as we did for our original hypotheses. The only change is to use a mean of zero with `dnorm()`.

```
dnorm(facts, mean = 0, sd = 7) %>% prod()
```

```
[1] 5.289902e-15
```

The likelihoods of the two hypotheses are:

- Hypothesis: Mean change of -10 mmHg: 5.9×10^{-17}
- Null hypothesis: Mean change of 0 mmHg: 5.3×10^{-15}

Taking the ratio of likelihoods gives 0.01. From this, we conclude that the original hypothesis is only 1% as likely as the null hypotheses. In other words, the original hypothesis is not very likely given the observed facts.

18.2 Digression: Likelihood, sensitivity, and specificity

Recall our discussion of classifiers in Lessons 34 and 35. There we defined two terms:

- Sensitivity: The probability of a + test given that the patient has the disease.
- Specificity: The probability of a - test given that the patient does not have the disease.

Both of these are actually likelihoods: the probability of a possible observation for a given condition of the world.

$p(D|+) = p(+|D)p(D)$ Probability is sensitivity times prevalence

Or do probability ratio: $\frac{p(D|+)}{p(H|+)} = \frac{p(+|D)p(D)}{p(+|H)p(H)}$ the probability of the state of the world given the observed data (a + or - test). Here, $p(H) = 1 - p(D)$ and $p(+|H) = 1 - p(-|H)$

We could turn the likelihood ratio into a probability ratio, but we need a "prior" to do so.

18.3 What makes hypothesis testing different?

1. No prior is allowed. Your subjective beliefs shouldn't influence your result.
2. There's no definite hypothesis other than the null hypothesis. This means that all calculations need to be done based on the likelihood of the null hypothesis. But with nothing to compare the likelihood to, we have a problem. Established solution: Compare the total likelihood over all facts that are "more extreme" than the observed facts.

18.4 The Null hypothesis as a DAG

Zero out any inputs to the node in question.

A DAG is a kind of hypothesis, a statement about the world that might or might not be true.

So far, we've used only the first three columns of the regression report: the name of the term to which the remaining entries belong, the estimate of the coefficient on that term, and the standard error of that estimate.

```
lm(mpg ~ wt + hp, data = mtcars) %>%
  regression_summary()
```

```
# A tibble: 3 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  37.2      1.60        23.3  2.57e-20
2 wt           -3.88     0.633       -6.13 1.12e- 6
3 hp           -0.0318   0.00903     -3.52 1.45e- 3
```

There are two more columns to go. The fourth column is labelled "**statistic**" (short for "**test statistic**") and the fifth column is the "**p-value**."

It's the p-value that concerns us here, the "statistic" is just an intermediate on the way to calculating the p-value. Both are reported because, in some fields people are accustomed to reading the statistic to draw quick conclusions. But in every field, the p-value is used.

The p-value is at once incredibly simple to interpret and impossibly difficult to make sense of. This contradiction is the reason many statisticians have called for moving away from the p-value as a summary of a result. We will discuss the reasons for the controversy in Lesson 38. In Lesson 37, we'll show how the p-value is computed from the test statistic and introduce another report summarizing a model, the "**ANOVA report**," which is useful for many purposes.

In this Lesson, we'll explain the "incredibly simple" interpretation of the p-value and the subtle logic behind it. This is important because frequently (pretty close to "always") people mistake the p-value as addressing a completely different question than the question it actually pertains to. It's useful to know about this misconception, because it points to a different question that often more directly addresses the needs of researchers and decision makers.

18.5 "Incredibly simple" interpretation

As you will see, the p-value is always a number between zero and one. When the p-value is small, the conclusion is that the corresponding explanatory variable is contributing to explaining the variation in the response variable. That is, a p-value that's small is justification for believing that there is a connection of some sort between the explanatory variable and the response variable.

"Small" in the phrase "when the p-value is small" is most usually taken to mean $p < 0.05$. But different fields have different standards for defining small. For instance, it's common in psychology to consider $p < 0.10$ as fairly small, while in physics, "small" means perhaps $p < 0.001$ or even $p < 0.000001$.

It may seem odd that there is no universal agreement about "small." The reason is that p-values are part of a *standard operating procedure* for evaluating research results to know if they are worthy of publication.

In physics, laws and models are meant to be exact or close to exact. Lord Rutherford (1871-1935), an important physicist who won the Nobel prize in 1908, famously disparaged the use of statistics, reportedly saying, "If your experiment needs statistics, you should have done a better experiment." This was in an era where the p-value *standard operating procedure* had not yet been invented. Today, when p-values are common in most fields, Rutherford's distaste for statistical method is reflected in p-value thresholds like $p < 0.000001$.

In other fields such as economics or psychology or clinical medicine, models are sought that are *useful* but without any expectation that they be exact. (In the 19th and early 20th century, psychologists and economists sometimes used the vocabulary of "law" to describe their findings, but "model" is more appropriate, because, unlike physics, the laws are not strictly enforced!) Often, in economics or psychology or medicine, the size of a sample used to train a model is less than, say, $n = 100$. And the units of observation—people or countries, for instance—are different one from the other, quite unlike, say, electrons, which are all the same. Consequently, sampling variation is often an important source of noise, obscuring relationships or even suggesting relationships that are not really there. (See Lesson 31.) This situation—small sample size, variation in observational units, and large sampling variation—would cause many useful findings to go unreported, as would happen if $p < 0.000001$ were the standard. So a less stringent threshold for publication is used, most commonly 0.05.

18.6 What is a p-value?

A p-value is the result of a calculation based on data, but also involving a special hypothesis, called the "Null hypothesis." The Null hypothesis is almost always a statement in line with the

claim that "there is no relationship between these variables" or "nothing is going on." For example, in a study about the effectiveness of a new drug, the Null hypothesis will be that the drug has no effect at all. Another example: In an economics study about the possible relationship between a country's "corruption index" and interest rates, the Null hypothesis would be "corruption is unrelated to interest rates."

Perhaps it is helpful to envision the Null hypothesis as the belief of a skeptical devil, standing on the researcher's shoulder and constantly whispering to the research that, "this study is useless, a waste of time, the result purely of sampling variation." Note that in a world where researchers always took the devil's advice, no study would be done. What motivates a researcher is a belief that the study will indeed produce results that are useful and represent something about the real world other than sampling variation, e.g. a relationship between two variables.

The calculation that results in a p-value is done under the assumption that the devil is right. The point is to see if the data are consistent with the devil's skeptical position. If they are consistent, then the p-value will be large. If not, the p-value will be "small."

The format of a p-value is that of a **conditional probability**. The condition is that the devil is right. The probability is that of seeing what the data analysis shows—typically summarized as a model coefficient—if the devil were right.

Actually, the probability reported in the p-value is not that of seeing the exact value of the model coefficient shown in the regression report. The probability also includes the events where the coefficient was larger in magnitude than the coefficient. Why? Because larger coefficients are stronger evidence that the devil is not right.

18.7 The world of the Null hypothesis

Recall that the Null hypothesis is the claim that "nothing is going on." For a regression model, this amounts to saying that "there is no relationship between an explanatory variable and

the response variable." In order to help clarify the description in the previous section, let's do an example calculation of a p-value. We will use for the example the possible relationship between a car's fuel economy (`mpg`) and the maximum horsepower (`hp`) of the engine.

A skeptic, such as the imaginary devil from the previous section, might argue this way: "The maximum horsepower is hardly ever used by a car. Instead, the driver throttles the engine so that it generates only that power needed to move the car along under the current conditions: acceleration, speed, wind, slope of the road. The maximum horsepower just affects the range of conditions under which the car can operate. But the fuel economy is based on a standard set of conditions which is the same for every car, regardless of the horsepower."

We will pick up the action at the point where the study has been designed and the design implemented to produce data. For the example, we have the `mtcars` data frame in hand. As should be familiar at this point, the data are modeled and the model coefficient on the explanatory variable of interest is recorded. Looking at the regression report presented at the beginning of this Lesson, that coefficient is -0.0318 mpg/horsepower.

The data were collected in the real world, but that is not the world that's relevant to the Null hypothesis. The world of the Null hypothesis is one where fuel economy is utterly unrelated to horsepower. To calculate the p-value, we construct a simulation of the Null-hypothesis world. But it is not sufficient for the simulation to generate Null-hypothesis data out of the blue. We want the simulation to be as much like the actual data as possible, except that there is no relationship between `mpg` and `hp`.

Perhaps surprisingly, there is a very simple device for accomplishing this. It involves creating a new variable to use in place of `hp` in the model, but which is unrelated to `mpg`. Let's call this new variable `hp_null`. We can generate `hp_null` by taking the values in `hp` and shuffling them. This randomized version of `hp` has no relationship to `mpg` because it is being dealt out to each row of the data frame at random.

Here's what the shuffling looks like, pretending for readability that there wee only ten rows in `mtcars`.

```
Samp <- mtcars %>%
  select(mpg, wt, hp) %>%
  sample_n(size=10) %>%
  mutate(hp_null = shuffle(hp))
Samp
```

```
                   mpg    wt  hp hp_null
Merc 450SLC       15.2 3.780 180     230
Porsche 914-2     26.0 2.140  91      52
Toyota Corolla    33.9 1.835  65     175
Honda Civic       30.4 1.615  52     175
Cadillac Fleetwood 10.4 5.250 205      91
Pontiac Firebird  19.2 3.845 175     180
Hornet Sportabout 18.7 3.440 175      62
Datsun 710        22.8 2.320  93      65
Merc 240D         24.4 3.190  62      93
Chrysler Imperial 14.7 5.345 230     205
```

We'll use such data, replacing the actual `hp` with the shuffled `hp`, to find the model coefficient on `hp`. This can be done concisely:

```
set.seed(112)
lm(mpg ~ wt + shuffle(hp), data = mtcars) %>%
  regression_summary()
```

```
# A tibble: 3 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  34.2      2.93        11.7  1.65e-12
2 wt           -4.95     0.626       -7.90 1.03e- 8
3 shuffle(hp)   0.0120   0.00894      1.34 1.90e- 1
```

In this trial, the coefficient on the shuffled `hp` is 0.0120. Of course the coefficient might well be different if the trial were

repeated. Let's run 1000 trials, from each of which we'll extract the coefficient on the shuffled `hp`.

```
Trials <- do(1000) * {
  lm(mpg ~ wt + shuffle(hp), data = mtcars) %>%
  regression_summary() %>%
  filter(term == "shuffle(hp)") %>%
  select(estimate)
}
```

Figure 18.1 shows the distribution of the shuffled `hp` coefficient, compared to the coefficient we found from the original, unshuffled data.

```
gf_jitter(estimate ~ 1, data = Trials, alpha=0.3, width=0.3) %>%
  gf_violin(color=NA, fill="blue", alpha=0.5, width=0.1) %>%
  gf_lims(x=c(0,2)) %>%
  gf_hline(yintercept = ~ -0.03177295, color="red")
```

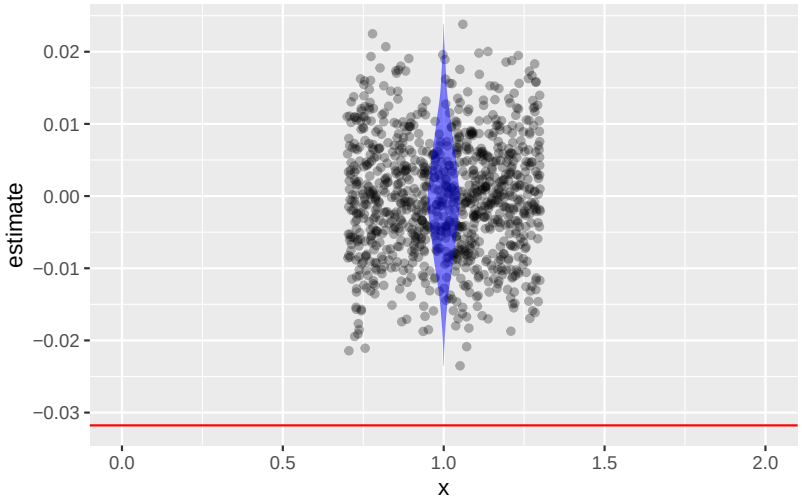


Figure 18.1: The sampling distribution of the shuffled `hp` coefficient

You can see in Figure 18.1 that the coefficient on the shuffled `hp` is near zero, as would be expected since we enforced `hp_null` to be unrelated to `mpg`. Almost always, the coefficients on `hp_null`

are in the interval ± 0.02. That is to say, even though `hp_null` is unrelated to `mpg`, sampling variation will spread out the estimated coefficient away from the ideal of zero. The amount of spread due to sampling variation is ± 0.02.

The estimated coefficient on `hp` in the original, unshuffled data is shown as a red horizontal line. You can see that this is farther from zero than any of the null-hypothesis trials. Since there were 1000 trials, the extreme nature of the coefficient from the original data let's us eyeball the probability of that coefficient (or larger) coming out of the Null hypothesis simulation is something on the order of one-in-a-thousand. A detailed calculation—refer to the regression table at the start of this Lesson—puts that probability at $p = 0.0015$.

18.8 What to conclude?

Remember always that the p-value is a probability calculated in a **hypothetical world** the world of the Null hypothesis. In the calculation, we are able to place the data in this hypothetical world by shuffling the explanatory variable.

No calculation done in the Null hypothesis world is going to tell us whether that hypothesis is correct or not. Nonetheless, that the Null hypothesis simulation did not generate a coefficient as large as that in the actual data suggests that the data are inconsistent with the Null hypothesis, that we can in the case of the `hp` coefficient regard the Null hypothesis as an implausible candidate to account for the data.

Almost always, newcomers to this p-value based scheme of hypothesis testing misinterpret the p-value to be the probability that the Null hypothesis is right. Small p-value would thus mean a small probability that the Null is right.

But suppose we want to do a calculation to produce something in the format "the probability that the Null is right?" The probability that decision-makers are usually interested in is the relative conditional probabilities for each of a set of hypotheses of interest. The "condition" under which these probabilities are

calculated is, "*given the data at hand*." Returning to the notation of Lesson 34, this is $p(H|\text{data})$, where H stands for each of the hypotheses of interest, say, that a drug has a large effect, a medium effect, no effect at all, or even a negative effect. The framework for calculation is called "**Bayesian**" statistics, the ideas of which date from the very beginnings of the emergence of statistical method.

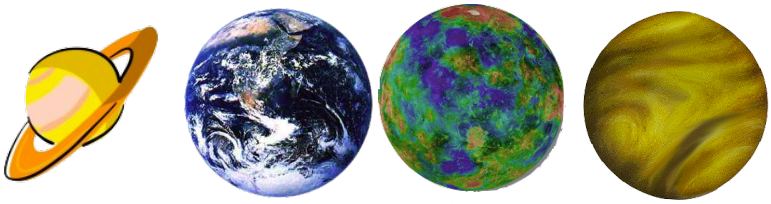
To illustrate the Bayesian approach, return to Lesson 35 when we were evaluating the performance of classifiers. There, we had two hypotheses that were relevant. In the context of health, these might be H_{sick} and H_{healthy}. The quantity of ultimate interest to the patient is p(sick given the test result). To calculate this probability we need to take a round-about route. We first find two completely different probabilities: p(test result given sick) and p(test result given healthy). In practice, these two probabilities are accessible: take a group of sick patients and see what fraction of them have positive tests, and take a different group of people who are healthy and see what fraction of that group have positive tests. With those two probabilities in hand, we take an estimate of the **prevalence** of sickness: p(sick). Then the probability of clinical interest, p(sick given test result) can be calculated using the Bayesian formula, just as we did in Lesson 35.

In contrast, the p-value is a probability in a different format: $p(\text{summary(data)}|H_0)$. Here, H_0, the Null hypothesis, is indeed a specific hypothesis, but not any hypothesis that motivates the research. The quantity "summary(data)" is a particular summary computed from the data, say the sample mean or a regression coefficient.

The p-value probability is bound to be confusing on first sight (and, for most people, on second, third, and later sightings). After all, we know exactly what is the "summary(data)"; we just calculated it from the data! The probability of "summary(data)" is therefore 1, at least until you understand what is the event being summarized by the p-value probability.

For the p-value, the random event that lies behind the probability is a number generated by a process: Go to the world of the Null hypothesis, that boring world of "nothing happening" or "no relationship between variables." Figure 18.2 lays out

the different worlds involved in statistical inference using the metaphor of planets.



(a) Planet Alt (b) The real world (c) Planet Sample (d) Planet Null

Figure 18.2: The four planets of the statistical solar system.

What motivates the work of collecting and modeling data is a hypothesis about the world. Typically, such hypotheses are simplistic, cartoon-like ideas about the shape of things.

Naturally, our ultimate interest is in the real world. But we don't have the whole Earth at hand; we have only a sample from it. The sample is something like the real world, but being a sample it is somewhat patchy, assembled from the n cases in our sample. Planet Sample lacks the detail of the real world, but each point on Planet Sample comes from a genuine place on Planet Earth.

The p-value is a probability computed on Planet Null, that boring world where nothing is going on and any perceived patterns are illusions, the appearance produced by random and shifting gusts of the winds of chance.

Almost all the work of calculating a p-value takes place on Planet Null. That work consists of simulation trials. Each trial involves taking a sample from Planet Null, summarizing it, and recording the result for later comparison the the summary calculated from Planet Sample.

It may seem perverse to base conclusions for real-world data on an imagined planet of no direct interest. And it is! At a minimum, we should put into competition at least two hypotheses: for instance Planet Alt and Planet Null. But in the world of the first half of the 20th century, when statistical analysis of data was just coming into the mainstream, it was impractical to compute the competing probabilities of the Bayesian style of

reasoning. The reason: the computers and algorithms we use now had not been invented.

In addition, those early statisticians put a big premium on what they called "objectivity." They did not think the subjective beliefs of researchers—the cartoon alternative hypothesis—should play any role in data analysis. The method they ended up inventing, p-values, was based only on a hypothesis that everyone could agree might be in play: the Null hypothesis. Unfortunately, the only valid conclusions that can be drawn from p-values are 1) "reject the Null hypothesis" and 2) "fail to reject the Null hypothesis." These conclusions don't guide us to favor any other particular hypothesis and so are inadequate to support decision-making in the real world. But the p-value conclusions can be the basis for a standard operating procedure: If the conclusion is "fail to reject the Null hypothesis," don't allow the work to be published.

So, standard operating procedures were based on the tools at hand. We will return to the mismatch between hypothesis testing and the contemporary world in Lesson 38.

18.9 More metaphors?

Use this from Section 19.4 of *Computational Probability and Statistics*?

> A US court considers two possible claims about a defendant: she is either innocent or guilty. Imagine you are the prosecutor. If we set these claims up in a hypothesis framework, the null hypothesis is that the defendant is innocent and the alternative hypothesis is that the defendant is guilty. Your job as the prosecutor is to use evidence to demonstrate to the jury that the alternative hypothesis is the reasonable conclusion.
>
> The jury considers whether the evidence under the null hypothesis, innocence, is so convincing (strong) that there is no reasonable doubt regarding the person's guilt. That is, the skeptical perspective (null

hypothesis) is that the person is innocent until evidence is presented that convinces the jury that the person is guilty (alternative hypothesis).

Jurors examine the evidence under the assumption of innocence to see whether the evidence is so unlikely that it convincingly shows a defendant is guilty. Notice that if a jury finds a defendant not guilty, this does not necessarily mean the jury is confident in the person's innocence. They are simply not convinced of the alternative that the person is guilty.

This is also the case with hypothesis testing: even if we fail to reject the null hypothesis, we typically do not accept the null hypothesis as truth. Failing to find strong evidence for the alternative hypothesis is not equivalent to providing evidence that the null hypothesis is true.

There are two types of mistakes possible in this scenario, letting a guilty person go free and sending an innocent person to jail. The criteria for making the decision, reasonable doubt, establishes the likelihood of those errors.

Hypothesis tests are not flawless. Just think of the court system: innocent people are sometimes wrongly convicted and the guilty sometimes walk free. Similarly, data can point to the wrong conclusion. However, what distinguishes statistical hypothesis tests from a court system is that our framework allows us to quantify and control how often the data lead us to the incorrect conclusion.

There are two competing hypotheses: the null and the alternative. In a hypothesis test, we make a statement about which one might be true, but we might choose incorrectly. There are four possible scenarios in a hypothesis test, which are summarized below.

Make this table nicer by constructing it in some other system.

.	do not reject H_0_	reject H_0_ in favor of H_A_
H_0_ true	Correct decision	Type 1 error
H_A_ true	Type 2 error	Correct decision

A **Type 1 error**, also called a **false positive**, is rejecting the null hypothesis when H_0 is actually true. Since we rejected the null hypothesis in the gender discrimination (from the Case Study) and the commercial length studies, it is possible that we made a Type 1 error in one or both of those studies. A **Type 2 error**, also called a **false negative**, is failing to reject the null hypothesis when the alternative is actually true. A Type 2 error was not possible in the gender discrimination or commercial length studies because we rejected the null hypothesis.

⚠ In DRAFT

Recast the previous paragraph to tie it to classifiers. Point out that in a hypothesis test, unlike a court, we never "accept the Null hypothesis." Neither is there any definite notion of "true," since neither the Null nor the Alternative are strictly speaking correct: they are both models of the world.

19 Calculating a p-value

Prof. Danny Kaplan
November 17, 2022

20 False discovery

Prof. Danny Kaplan
November 17, 2022

> *Ask, and it shall be given you; seek, and ye shall find; knock, and it shall be opened unto you: For every one that asketh receiveth; and he that seeketh findeth; and to him that knocketh it shall be opened.*
> – Matthew 7:7-8

The modeling techniques we've covered are surprisingly powerful at identifying patterns in data. With power comes responsibility. This chapter is about how spurious patterns can arise in data and processes you can use to help ensure that the patterns your models identify are genuine.

It's well known that people are particularly adept at finding patterns. To see this, spend a minute or two with Figure 20.1, which shows x-y pairs generated by a complex mathematical procedure called the Mersenne-Twister algorithm. How many of the structures created by Mersenne-Twister algorithm can you identify by eye? Take five of the stronger-looking patterns: clusters of points, large empty areas, strings of dots, etc. Write down a list of the patterns you spotted, including the coordinate location of each, a short description (e.g. "arc of dots"), and your subjective sense of how strong or convincing that pattern is.

With your list in hand, look at Figure 20.2 at the end of this section, which displays another n = 1000 x-y pairs generated by the same mathematical procedure. You're going to check which of the patterns you found in the testing data are confirmed by the training data. Go through your list, looking at each location

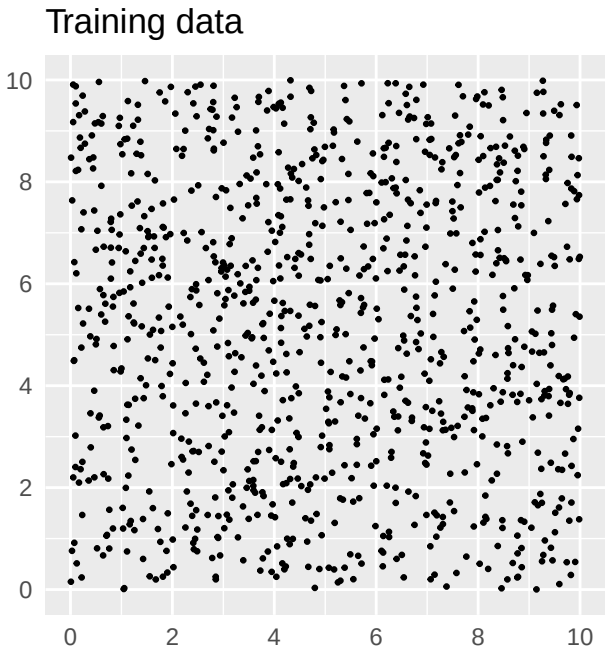


Figure 20.1: Training data (n = 1000)

where you found a pattern in the training data and checking whether a similar pattern appears at that location in the testing data.

Were any of the patterns you saw in the training data confirmed by the testing data?

There's no denying that the patterns you saw were in the data. But the Mersenne-Twister algorithm is specifically designed *not* to produce regular patterns. Any that you saw were accidental alignments in the particular sample of data from the algorithm.

The "patterns" abstractly referred to in the previous paragraphs appear in data. In data used for modeling, a pattern might be a relationship or correlation between two or more variables, or a cluster of rows in a data frame that have similar values for a response variable and explanatory variables.

Training models on data can encode the underlying patterns. For instance, a pattern in the data might result in a model

generating detailed predictions or demonstrating a strong effect size of one variable on another.

A *valid* pattern is one that steadily appears from one sample of data to another (so long as the sample is big enough). Such consistency suggests that the pattern reflects some genuine aspect of the system generating the data. A *false* or *accidental* pattern is one that appears in a sample of data, but is unlikely to show up in another sample. This inconsistency indicates that conclusions based on this pattern are unlikely to be applicable in the future or in new situations.

The obvious, direct way to check the validity of a pattern encoded by a model is to see if the same pattern occurs in new data, data that was not used in building the initial model. Lesson 22 took this approach by constructing a *sampling distribution* of a statistic such as an effect size. To create a sampling distribution, we train many models on different subsets of a data set.

When working with prediction models, the sign of a valid pattern is that the quality of the predictions – perhaps measured with a root-mean-square-error or a sensitivity/specificity – remains consistent when we calculate it on new data. A prediction that shows very small error on the data used to train the model but large error on new data is not a prediction that we can rely on in new settings.

The historical rapid growth in data analysis activity and the construction of data sets with large numbers of explanatory variables has made it easier to capture with models both valid patterns and false patterns. This makes it important to recognize that the false detection of patterns is possible whenever you train a model, to be aware of the characteristics of models and data that make false detection more likely, and to adopt procedures to mitigate the risk that the results of your work may not generalize beyond the particular sample of data you have in hand.

You are a data scientist for an internet retailer, Potomac.com, which has just bought a national grocery chain, Austin Foods. You're part of the team that is connecting the customer loyalty card data from Austin Foods with Potomac's own large record

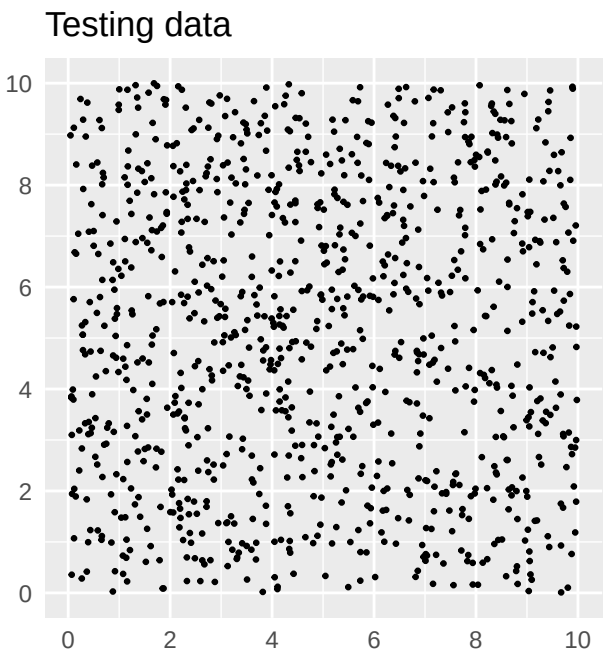


Figure 20.2: Testing data (n = 1000).

of purchases. This is accomplished by offering a 10% Xdiscount for an item on Potomac to people who enter their Austin loyalty card number.

Potomac's management wants to create a cross-marketing program in which a customer shopping at Potomac will be offered coupons for Austin products. The hope is that the coupon discount will attract new customers to start shopping at Austin's. In order for this to work, it's best if the coupons are for products that the customer finds attractive.

Your job is to build the *coupon assignment system*, that is, to figure out how to choose which products a customer is most likely to find attractive. To do this, you'll create a set of classifiers that indicates the interest of a Potomac customer in an Austin product.

You've got data on 10,000 Potomac/Austin customers, that is, people whose records from Potomac and from Austin you can bring together. There are ten popular Austin products for which coupons can be offered. Among the 10,000 customers,

about 16% have actually bought any given Austin product. You have built ten classifiers, one for each of the ten products. The input to the classifiers is 100 standard measures of a customer's Potomac activity. The output of each classifier is the probability that the customer actually bought the corresponding Austin product.

The no-input classifier gives a probability of about 16% that the customer will buy the product. Management hopes that you will be able to segment the market to identify the products that a given person is much more likely to buy.

It's a lot to ask of a person to sort through 100 potential explanatory variables to identify those that are predictive of buying a product. But it's straightforward to use a model family that can *learn* on its own which variables are informative. You train the ten classifiers using a tree family of models.

> Heads up! The "data" has been created using random numbers, so that there are no actual relationships between the explanatory variables and the purchase outcomes. That is, no actual relationships aside from the accidental ones, such as the patterns encountered in Figure 20.1.

To illustrate how the coupon assignment system works, Table @ref(tab:some-results) shows an intermediate step in the calculation, where a probability for each of the ten products is calculated for each customer.

Table @ref(tab:some-results) shows the output of the classifiers for just the first fifteen customers out of the 10,000 used to build the coupon selection system. For each person, all ten classifiers have been applied to estimate the probability that the person would buy each of the ten products. Highlighted in green are those products with a purchase probability greater than 40%.

The final output of the coupon assignment system is, for each customer, the identification of the specific products for which the probability is large. Reading Table @ref(tab:some-results), you'll see that for person 1, product 9 merits a coupon. For person 2, products 2 and 10 merit a coupon. A winning product

Table 20.1: (ref:some-results-cap)

	Customer ID														
product	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	11	**71**	7	9	**63**	14	9	8	0	13	7	11	9	4	11
2	6	14	20	6	14	14	14	20	14	14	14	14	14	14	14
3	15	11	11	6	15	15	12	11	11	15	10	9	8	8	5
4	6	14	11	8	9	11	6	9	13	13	13	13	6	10	11
5	11	11	11	11	11	0	13	**92**	7	11	13	13	**43**	8	9
6	13	13	13	7	9	13	10	12	10	13	13	13	10	7	13
7	15	12	6	9	9	15	11	**75**	9	11	9	11	4	9	16
8	30	10	10	8	6	10	12	7	6	**78**	**86**	6	6	13	8
9	**67**	14	8	9	10	10	10	**75**	11	9	0	11	15	2	9
10	19	**46**	6	9	10	7	**42**	9	10	6	16	6	9	16	14

has not been identified for every customer, but you can't please everyone.

```
Attaching package: 'formattable'

The following objects are masked from 'package:scales':

    comma, percent, scientific
```

(ref:some-results-cap) The output of the ten classifiers for the first 15 customers. Green highlighting is used for those products which a given customer is likely to buy.

To test the performance of the system, we can look at the product/customer combinations for which a coupon was merited, and check how many of them actually corresponded to a purchase: it's 74%. But for the combinations with no coupon, the purchase rate was only 11%.

The results are impressive. For about half of the customers, the coupon assignment system has identified customers/product combinations with a purchase probability of more than 40%. Often, the probability of purchase is considerably higher than 40%. Targeting each customer with a coupon for the right product is likely to generate a lot of new sales!

> Since data was generated using random numbers, we know that the "success" of the coupon assignment system is illusory. Later, we'll see how the process was able to uncover so many accidental patterns from random data and list some things to look out for when modeling. But first, let's provide a reliable method for you to identify when your results are based in accidental patterns: using testing data.

A true measure of the performance of a model should be based not on the data on which the model was trained, but data which have been held back for use in testing and not used in training. For this example, we'll use testing data consisting of 10,000 customers for whom we have the same 100 explanatory variables from the Potomac database and for whom we know if each customer purchased any of the ten products from Austin Foods. Only about 1 in 6 customers bought any single product from Austin. We want to see if the classifier assigns a high probability to those customers who did buy the product. If so, it means we can use just the 100 explanatory variables to find winning products for customers for whom we have no Austin purchasing data.

```
Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if
`.name_repair` is omitted as of tibble 2.0.0.
i Using compatibility `.name_repair`.
```

(ref:purchase-test-cap) Similar to Table @ref(tab:some-results) but for the *testing* data.

A valid evaluation of the performance of the system involves using the *testing* data rather than the *training* data. Figure @ref(fig:purchase-test) shows the assignment of coupons for the customers in the test data. Although coupons are assigned to these customers, the purchase rate for these items is only 16%, no different than the probability of purchase for no-coupon items. In other words, the coupon assignment system doesn't work at all!

Table 20.2: (ref:purchase-test-cap)

	Customer ID														
product	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	*17*	*8*	**75**	*16*	**62**	*15*	*2*	*11*	*8*	*24*	*7*	*15*	*7*	*13*	*10*
2	*14*	*20*	*14*	*14*	*14*	*14*	*14*	*6*	*14*	*13*	*6*	*14*	*14*	*13*	*14*
3	*14*	*22*	*6*	*8*	*14*	*10*	*14*	*13*	**63**	*11*	*9*	*11*	*17*	*13*	*15*
4	*13*	*13*	*13*	*13*	*13*	*13*	*6*	*9*	*0*	*13*	*13*	*13*	*12*	*0*	**62**
5	*9*	*13*	*9*	**85**	*11*	*11*	**59**	*9*	*12*	*7*	*11*	*11*	*11*	*13*	*13*
6	*13*	*13*	*13*	*14*	*7*	*7*	*12*	*9*	*13*	*11*	**80**	*13*	*7*	*7*	*13*
7	*15*	*11*	*12*	*6*	*9*	*0*	*9*	*15*	*5*	*7*	*33*	*12*	*9*	*12*	*15*
8	*7*	*13*	*6*	*7*	*10*	*6*	*13*	*14*	*10*	**100**	*9*	*8*	*10*	*13*	*6*
9	*9*	*12*	**75**	**58**	*8*	*11*	*11*	*9*	*10*	*8*	**60**	*11*	*10*	*10*	*12*
10	*2*	*10*	*20*	*2*	*16*	*12*	*7*	**75**	*15*	*16*	**83**	*15*	*16*	**56**	*14*

20.1 Sources of false discovery

How did the coupon classifier system identify so many accidental patterns, patterns that existed in the training data but not in the testing data?

One source of false discovery stems from having multiple potential response variables. In the Potomac/Austin example, there were ten different classifiers at work, one for each of the ten Austin products. Even if the probability of finding an accidental pattern in one classifier is small, looking in ten different places dramatically increases the odds of finding something.

Similarly, having a large number of explanatory variables – we had 100 in the coupon classifier – provides many opportunities for false discovery. The probability of an accidental pattern between one outcome and one explanatory variable is small, but with many explanatory variables each being considered it's much more likely to find something.

A third source of false discovery at work in the coupon classifier relates to the family of models selected to implement the classifier. We used a tree model classifier capable of searching through the (many) explanatory variables to find ones that are associated with the response outcome. Unbridled, the tree

model is capable of very fine stratification. Each coupon classifiers stratified the customers into about 200 levels. On average, then, there were about 50 customers in each strata. But there is variation, so many of the strata are much smaller, with ten or fewer customers. The small groups were constructed by the tree-building algorithm to have similar outcomes among the members, so it's not surprising to see a very strong pattern in each group. For each classifier, about 15% of all customers fall into a strata with 20 or fewer customers.

To illustrate, Figure 20.3 shows the shape of the tree model for a typical coupon classifier. Each of the splits reflects an accidental alignment of the response variable with one of the explanatory variables. As more splits are made, the group of customers contained in the split becomes smaller. Many of the leaves on the tree contain just a handful of customers who accidentally had similar values for the several explanatory variables used in the splits.

The tree is too complex to be plausible as a real-world mechanism. None of the details in Figure 20.3 have any validity beyond the training data itself.

20.2 Identifying false discovery

We use data to build statistical models and systems such as the coupon-assignment machine. False discovery occurs when a pattern or model performance seen with one set of data does not generalize to other potential data sets.

The basic technique to avoid false discovery is called **cross validation**. One simple approach to cross validation splits the data frame into two randomly selected non-overlapping sets of rows: one for training and the other for testing. Use the training data to build the system. Use the *testing* data to evaluate the system's performance.

Most often, cross validation is used to test model prediction performance such as the root-mean-square error or the sensitivity and specificity of a classifier. This can be accomplished by

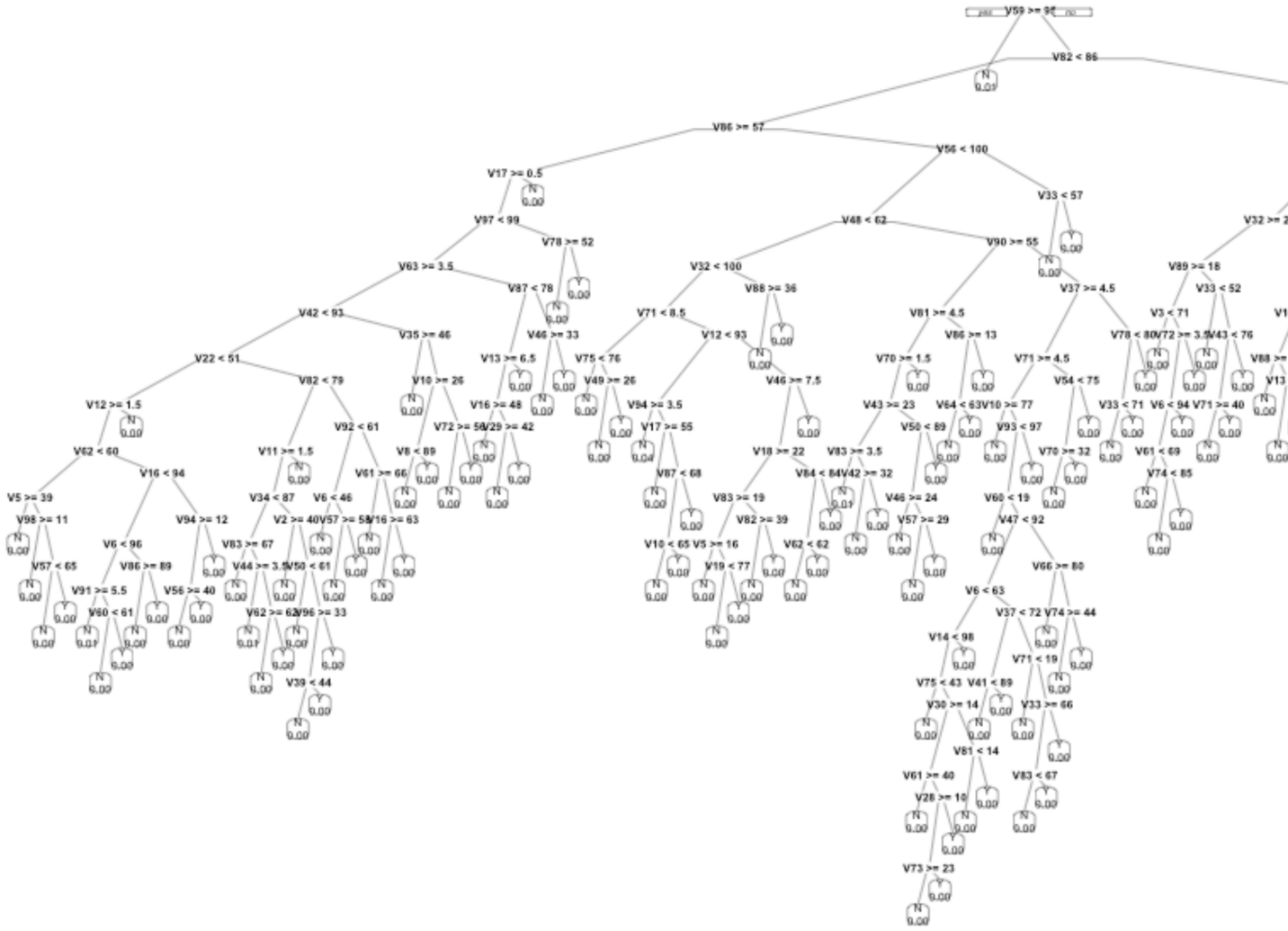


Figure 20.3: A sketch of one of the classifiers constructed for the coupon selection system. The tree-growing algorithm was allowed to keep going until the customer data was split up into very small strata.

taking the trained model and providing as input the explanatory variables from the testing data, then comparing the model output to the actual response variable values in the testing data. Note that using testing data in this way does not involve retraining the model on the testing data.

How big should the training set be compared to the testing set? For now, we'll keep things simple and encourage use of a 50:50 split or something very close to that.

This is a simple and reliable approach that should always be used.

20.3 False discovery and multiple testing

When the main interest is in an effect size, standard procedure calls for calculating a confidence interval on the effect. For example, a 2008 study examined the possible relationship between a woman's diet before conception and the sex of the conceived child. The popular press was particularly taken by this result from the study:

> Women producing male infants consumed more breakfast cereal than those with female infants. The odds ratio for a male infant was 1.87 (95% CI 1.31, 2.65) for women who consumed at least one bowl of breakfast cereal daily compared with those who ate less than or equal to one bowlful per week. (**fetal-sex-2008?**)

The model here is a classifier of the sex of the baby based on the amount of breakfast cereal eaten. The effect size tells the change in the odds of a male when the explanatory variable changes from one bowlful of cereal per week to one bowl per day (or more). This effect size is sensibly reported as a ratio of the two odds. A ratio bigger than one means that boys are more likely outcomes for the one-bowl-a-day potential mother than the one-bowl-a-week potential mother. The 95% confidence interval is given as 1.31 to 2.65. This confidence interval

Table 20.3: (ref:sex-consumption-1-cap)

	high	low
B	165	182
G	211	182

does not contain 1. In a conventional interpretation, this provides compelling evidence that the relationship between cereal consumption and sex is not a false pattern.

But the confidence interval is not the complete story. The authors are clear in stating their methodology: "Data of the 133 food items from our food frequency questionnaire were analysed, and we also performed additional analyses using broader food groups." In other words, the authors had available more than 133 potential explanatory variables. For each of these explanatory variables, the study's authors constructed a confidence interval on the odds ratio. Most of the confidence intervals included 1, providing no compelling evidence of a relationship between that food item and the sex of the conceived child. As it happens, breakfast cereal produced the confidence interval that was the most distant from an odds ratio of 1.

Let's look at the range of confidence intervals that can be found from studying 100 potential random variables that are each unrelated to the response variable. We'll simulate a response randomly generated "sex" G and B where the odds of G is 1. Similarly, each explanatory variable will be a randomly generated "consumption" high or low where the odds of high is 1. A simple stratification of sex by consumption will generate the odds of G for those cases with consumption Y and also the odds of G for those cases with consumption N. Taking the ratio of these odds gives, naturally enough, the odds ratio. We can also calculate from the stratified data a 95% confidence interval on the odds ratio.

So that the results will be somewhat comparable to the results in (**fetal-sex-2008?**), we'll use a similar sample size, that is, n = 740. Table @ref(tab:sex-consumption-1) shows one trial of the simulation.

(ref:sex-consumption-1-cap) A stratification of sex outcome (B

or G) on consumption (high or low) for one trial of the simulation described in the text.

Referring to Table @ref(tab:sex-consumption-1), you can see that the odds of G when consumption is low is 182 / 182 = 1. The odds of G when consumption is high is 211/165 = 1.28. The 95% confidence interval on the odds ratio can be calculated. It is 0.95 to 1.73. Since that includes 1, the data underlying Table @ref(tab:sex-consumption-1) provide little or no evidence for a relationship between sex and consumption. This is exactly what we expect, since the simulation involves entirely random data.

Figure 20.4 shows the 95% confidence interval on the odds ratio for 133 trials like that in Table @ref(tab:sex-consumption-1). The confidence interval from each trial is shown as a horizontal line. The large majority of them include 1. That's to be expected because the data have been generated so that sex and consumption have no relationship except those arising by chance.

```
Warning: geom_vline(): Ignoring `mapping` because `xintercept` was provided.
```

Nonetheless, out of 133 simulations there are six where the confidence interval does not include 1. These are shown in red. By necessity, one of the intervals will be the most extreme. If instead of numbering the simulations, we had labelled them with food items – e.g. grapefruit, breakfast cereal, toast – we would have a situation very similar to what seems to have happened in the sex-vs-food study. (For a more detailed analysis of the impact of multiple testing in (**fetal-sex-2008?**), see (**young-2009?**).)

Suppose now that half of the data used in (**fetal-sex-2008?**) had been held back as testing data. Using the training data, it would be an entirely legitimate practice to generate hypotheses about which specific food items might be related to the sex of the baby. The validity of any one selected hypothesis could then be established using the testing data without the ambiguity introduced by multiple testing. The testing data confidence interval can be taken at face value; the training data confidence interval cannot.

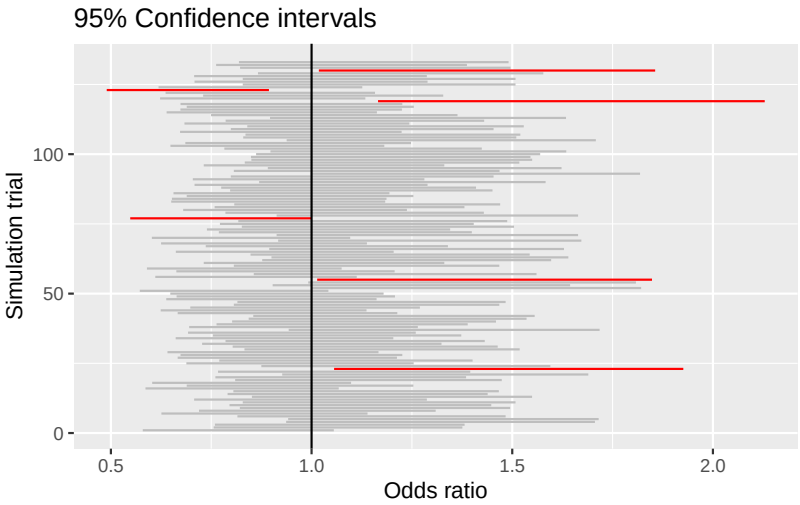


Figure 20.4: Confidence intervals on the odds ratio comparing female and male birth rates for many trials of simulated data with no genuine relationship between the explanatory and response variables.

20.4 Example: Organic discovery?

It's easy to find organic foods in many large grocery stores. Advocates of an organic diet are attracted by a view that it is sustainable, promotes small farms, and helps avoid contact with pesticides. There are also nay-sayers who make valid points, but that is not our purpose here. Informally, I find that many people and news reports point to the health benefits of an organic diet. Usually they believe that these benefits are an established fact.

A 2018 New York *Times* article observed:

> *People who buy organic food are usually convinced it's better for their health, and they're willing to pay dearly for it. But until now, evidence of the benefits of eating organic has been lacking.* (**NYT-2018-10-23-Rabin?**)

The new evidence of health benefits is reported in an article in the *Journal of the American Medical Association: Internal Medicine* (**baudry-2018?**)

Describing the findings of the research, the *Times* article continued:

> *Even after these adjustments [for covariates], the most frequent consumers of organic food had 76 percent fewer lymphomas, with 86 percent fewer non-Hodgkin's lymphomas, and a 34 percent reduction in breast cancers that develop after menopause.*

The study warrants being taken seriously: it involved about 70,000 French adults among whom 1340 cancers were noted. The summary of organic foot consumption was a scale from 0 to 32 and included 16 labeled products including dairy, meat and fish, eggs, coffee and tea, wine, vegetable oils, and sweets such as chocolate. Adjustment was made for a substantial number of covariates: age, sex, educational level, marital status, income, physical activity, smoking, alcohol intake, family history of cancer, body mass index, hormonal treatment for menopause, and others.

Yet … the reseach displays many of the features that can lead to false discovery. For instance, results were reported for four different types of cancer: breast, prostate, skin, lymphomas. The study reports p-values and hazard ratios[1] comparing cancer rates among the four quartiles of the organic consumption index.

Comparing the most organic (average organic index 19.36/32) and the least organic (average index 0.72/32) groups the 95% confidence interval on the relative risk and p-values given in the study's Table 4 are:

- Breast cancer: 0.66 - 1.16 ($p = 0.38$)
- Prostate cancer: 0.61- 1.73 ($p = 0.39$)
- Skin cancer: 0.49 - 1.28 ($p = 0.11$)
- Lymphomas: 0.07 - 0.69 ($p = 0.05$)

You might be surprised to see that the confidence interval on the relative risk for breast cancer includes 1.0, which suggests no evidence for an effect. As clearly stated in the report, the risk reduction for breast cancer is seen only in a subgroup of

[1] Hazard ratios are analogous to risk ratios.

study participants: those who are postmenopausal. And even then, the confidence intervals continue to include 1.0:

- Breast cancer pre-menopausal: 0.67 - 1.52 ($p = 0.85$)
- Breast cancer post-menopausal: 0.53 - 1.18 ($p = 0.18$)

So where is the claimed 34% reduction in breast cancer cited in the New York Times article. It turns out the the study used two different indices of organic food consumption. The 0 to 32 scale which includes many items for which the amount consumed is very small (e.g., coffee, chocolate) and a "simplified, plant derived organic food score." It's only when you look at the full 0 to 32 scale that you see the reduction in post-menopausal breast cancer: the confidence interval is 0.45 to 0.96 ($p = 0.03$).

What about cancer rates overall? For the 0 to 32 scale the risk ratio was 0.58 - 1.01 ($p = 0.10$). To see the claimed reduction clearly you need to look at the simplified food score which gives 0.63 - 0.89 ($p < 0.005$). And it's only in comparing the highest-index quarter of participants with the lowerest quarter participants that any difference at all is seen in any type of cancer: the middle-half of participants show no difference in relative risk from the lowest-organic quarter of participants. (Because of this, had the study compared the highest quarter to the next highest quarter, they would have seen basically the same relative risks reported in the highest-to-lowest quarter comparison. Then the conclusion would have had a different flavor, perhaps to be reported as "Typical organic food consumption levels show no cancer benefits.")

A further source of potential false discovery stems from the study's starting and stop times. It's not clear that these were pre-defined; the reported results are intermediate to a longer follow up. The choice to report intermediate results is another way that the number of opportunities for false discovery is increased. And the choice is important: for the follow-up time used, about 2% of the participants developed cancer. In an earlier study of more than 600,000 middle-aged UK women (average age 59), the incidence of cancer was four times larger: 8.6%. (**bradbury-2014?**) That study did not find any relationship between organic food consumption and overall cancer rates, and found no relationship for 15 out of 16 different types

of cancer. The exception is extremely interesting: non-Hodgkin lymphoma for which a similar result was found in the French study.

So is the study reported in the New York *Times* a matter of false discovery? It's emotionally unsatisfying to discount a result about organic food and non-Hodgkin lymphoma simply because it's part of a larger study that looked at many different combinations of cancer types and organic food indices. What if the researchers had only studied non-Hodgkin lymphoma – they would have gotten the same result and it wouldn't have the deficiencies of being the strongest result of many possibilities. It would have stood on its own. But it doesn't and we are left in a state of doubt.

20.5 p-values and "significance"

False discovery is not a new problem. The traditional logic can be traced back to 1710, when John Arbuthnot was examining London birth records from 1629 to 1710. Arbuthnot was surprised to find that for each year males were more common than females. In interpreting this finding, Arbuthnot refered to the conventional wisdom that births of males and females are equally likely. If this were the case, in any one year there might, by chance, be more females than males or the other way around. While it's theoretically possible that chance might produce the string of 82 years with more males, it's very unlikely. "From whence it follows, that it is Art, not Chance, that governs," Arburthnot wrote. In more modern language, Arburthnot concluded that the hypothesis of equal rates of male and female births was not consistent with the data. Arbuthnot's "Chance" corresponds to false discovery, while "Art" is a valid discovery.

Arburthnot's logic became a standard component of statistical method.

1. Summarize the data into a single number called a "**test statistic**". For Arburthnot the test statistic was the number of years where male births predominated, out of the

82 years being examined. The observed value of the test statistic was 82.

2. State a "**null hypothesis**", typically something that is the conventional wisdom. For Arburthnot, the null hypothesis was that male and female births are equally likely.
3. Calculate a hypothetical quantity based on the null hypothesis: the probability that the test statistic produced in a world in which the null hypothesis holds true would be at least as large as the test statistic.
4. If the probability in (3) is small, one is entitled to "reject the null hypothesis." Typically, "small" is defined as 0.05 or less.

In the 1890s, statistical pioneer Karl Pearson invented a test statistic he called χ^2 ("chi"-squared, with "chi" pronounced "ki" as in "kite") that can be applied in a variety of settings. In 1900, Pearson published a table (**pearson-1900?**) that makes it an easy matter to look up the probability in step (3) above. He called this theoretical probability "P", a name that has stuck but is conventionally written as lower-case "p".

Data scientists tend to work with "big data", but for many applications of statistics, data is so scarce that use of separate training and testing data is impractical. For such small data, the calculation of a p-value can be a sensible guard against false discovery. Still, a p-value does not address any of the sources of false discovery outlined in the previous sections of this chapter. When used with small data and simple modeling methods, those sources of false discovery are not so much of a problem. In small data there won't be multiple explanatory variables that can be searched and there won't be a choice of response variables. This doesn't eliminate all problems, since in small data results can depend critically on the inclusion or exclusion of a single row of data. The name "**p hacking**" has been given to the various ways that researchers can manipulate p-values to get them below 0.05.

Another problem with p-values stems from misinterpretation of the admittedly difficult logic that underlies them. The misinterpretations are encouraged by the use of the term "**tests of significance**" to the p-value method. Particularly galling

is the use of the description "**statistically significant**" to describe a result where $p < 0.05$. The everyday meaning of "significant" as something of importance is in no way justified by $p < 0.05$. Instead, the practical importance or not is more clearly signaled by examining an effect size. (It's extremely disappointing that journalists, who are writing for an audience that for the most part has no understanding of p-value methodology, use "significant" when reporting on the statistics of research findings. It would be more honest to use a neutral term such as "null-validated" or "p-validated" which does not confuse the statistical result with actual practical importance.)

The p-value methodology has little or nothing to contribute to data science practice. When data is big there is a much more straightforward method to guard against false discovery: cross validation. And when data is big there is another, more fundamental problem with p-values. They are calculated with reference to a specific null hypothesis of "no effect" or "no relationship." More realistically, they should be calculated with respect to a hypothesis of "trivial (but potentially non-zero) effect". There are all sorts of mechanisms in the world (such as common causes) that can create the appearance of some effect or relationship. No matter how trivial in size this is, with sufficient data the p-value will become small. To illustrate, Figure 20.5 shows the p-value as a function of the sample size n in a system with an R-squared of 0.001, which in most settings would be of no practical signficance.

```
Warning: geom_hline(): Ignoring `mapping` because `yintercept` was provided.
```

20.6 NOTES IN DRAFT

"Statistical crisis" in science

https://www.americanscientist.org/article/the-statistical-crisis-in-science

Garden of the Forking Paths

Ionedes

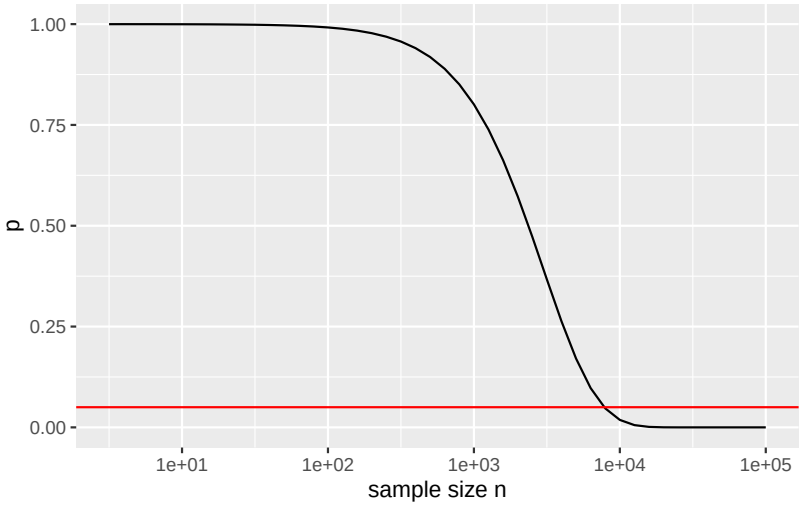


Figure 20.5: The p-value as a function of sample size n when the test statistic R-squared has the trivial value 0.001. The horizontal line shows the usual threshold for "significance" of $p < 0.05$.

21 Review of Lessons 9-19

> **Warning**
>
> I'll put learning challenges here. The class day will be given over to the QR.