The general linear model
Lecture 1 – Nuts and bolts

Jan Vanhove
https://janhove.github.io

Ghent, 14–16 July 2025

Goal and overview

This module is devoted to the workhorse of quantitative data analysis: the **general linear model** (GLM). This is a tool with which we can express how one or several **predictors** (or, indeed, none) are related to a single **outcome**. Commonly used techniques such as *t*-tests, analysis of variance (ANOVA) as well as regression models are all instantiations of the GLM, so a firm grasp of its underpinnings will stand you in good stead when learning how to analyse quantitative data. Moreover, if you want to master more advanced techniques such as general*ised* linear models (e.g., logistic regression) or mixed-effects modelling, you need to be comfortable with the general linear model first. The expected outcomes for this module (allowing for some time to digest the module's contents after the summer school) are the following:

- Students are able to explain the **literal meaning** of the parameter estimates in the output of general linear models. If you don't know the literal meaning of these numbers, interpreting them in terms of the subject matter (e.g., what these results mean for second-language learning) is fraught with danger.
- They know how to obtain **uncertainty estimates** about these parameter estimates under different sets of assumptions.
- They will always ask themselves whether the models they fit are at all **relevant** to the task at hand.
- They know how to **simulate data** in R in order to help them pick between alternative modelling strategies.

Time willing, we'll also take a look at the logistic model.

Prerequisites

The code in these lecture notes was last run using R version 4.5.1. I use the `tidyverse` package throughout (version 2.0.0) in order to draw graphs and work comfortably with data sets. Additionally, I use the `here()` function from the `here` package (version 1.0.1).

If you want to use the exact same commands as in this script, you'll need to create a project in RStudio. Please follow the instructions on `https://github.com/janhove/GeneralLinearModel`.

1 Another look at the mean

DeKeyser et al. (2010) administered a 204-item Hebrew grammaticality judgement task (GJT) to 76 Russian speakers who had moved to Israel. The authors were interested in gauging the relationship between the participants' performance on the GJT and the age at which they had started to learn Hebrew (age of acquisition, AOA). We'll look at this relationship in the next lecture. But first, we'll use their data set in order to introduce a few core concepts that we will make use of throughout this lecture series.

Tip 1.1. Don't type along or work in R as you're attending this lecture. You're bound to make some typos or forget some commas and brackets, which will cause your code not to work and you to lose track of the lecture. Instead, focus on the code's logic and take notes. Work through the R code at your own pace – after the lecture. ♢

Let's load the `tidyverse` and `here` packages and read in the data set. Incidentally, when you run these commands, you may see a few messages, but I opted not to print these in order to reduce the clutter.

```
library(tidyverse)
library(here)

# Override default plotting theme (black and white instead of grey)
theme_set(theme_bw())

d <- read_csv(here("data", "dekeyser2010.csv"))
```

Tip 1.2. As you're working through the R code after the lecture, don't copy-paste the commands from the PDF, but *type* them, character by character. You'll appreciate their structure better and you'll get more proficient at weeding out syntax errors. And if you don't know what a certain parameter is supposed to achieve, change its value and see what happens! ♢

Figure 1.1 shows a basic Cleveland dot plot of the participants' GJT scores.

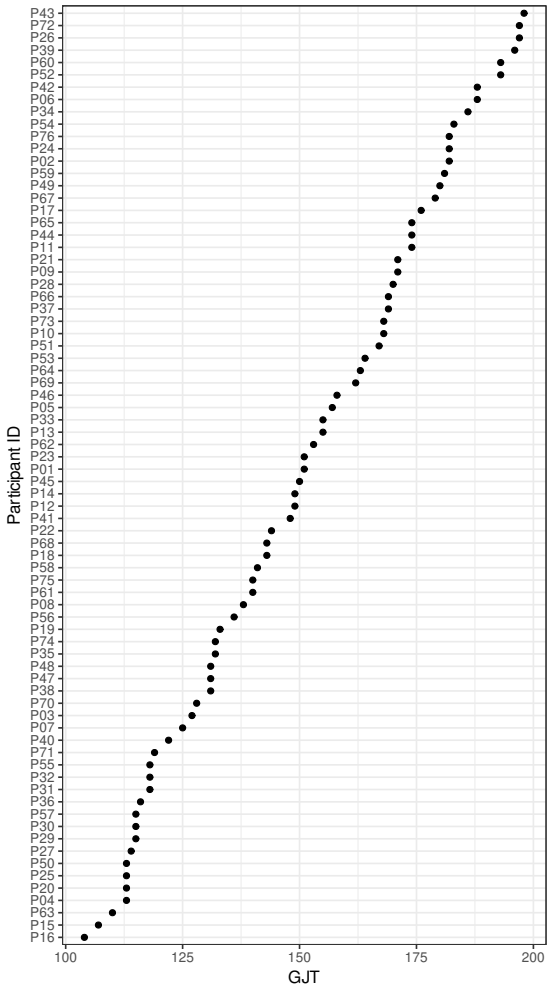

Figure 1.1: A Cleveland dot plot of the participants' grammaticality judgement scores. The plot nicely shows that there are no extreme outlying data points.

```
ggplot(data = d,
       aes(x = GJT,
           y = reorder(Participant, GJT))) +
  geom_point() +
  ylab("Participant ID")
```

Our first goal in this section is to describe these 76 GJT values using a single number. While different kinds of averages exist (arithmetic mean, median, trimmed mean, winsorised mean, etc.), let's pretend we don't know anything about those. Then, a promising plan of attack would be to partition the information in the data points into two parts: a systematic part that captures the communalities between the data points, and an unsystematic part that captures

the individual discrepancies between those communalities and the data points. That is, for each observation ($i = 1, \ldots, n$), we would have the following equation:

$$\text{the } i\text{-th outcome} = \text{communality between all } n \text{ outcomes} + \text{the } i\text{-th discrepancy.}$$

This notation is a bit cumbersome, so let's use some shorthand:

$$y_i = \beta_0 + \varepsilon_i,$$

where y_i is the i-th observation in the data set (here: $i = 1, 2, \ldots, 76$), β_0 represents the communality between all 76 outcome values, and ε_i expresses by how much the i-th observation deviates from this communality. The ε_i values are referred to as the equation's **errors**. We write β_0 instead of just β because we'll be using more than one β parameter shortly. We're usually more interested in the βs than in the εs.

Since we don't have the entire population at our disposal, we can only **estimate** the βs based on our data set. We'll let $\widehat{\beta}_0$ denote our estimate of β_0. Since $\widehat{\beta}_0$ is merely an estimate, so, too, will the values of ε that we'll end up with be estimates:

$$y_i = \widehat{\beta}_0 + \widehat{\varepsilon}_i. \tag{1}$$

This is equivalent to

$$\widehat{\varepsilon}_i = y_i - \widehat{\beta}_0.$$

These estimated errors are called the **residuals**.

1.1 Obtaining parameter estimates

But how to compute $\widehat{\beta}_0$? Or put differently: How to estimate β_0? In principle, we could pick any value for $\widehat{\beta}_0$ and choose the $\widehat{\varepsilon}_i$ values accordingly to make Equation 1 hold. For instance, we could arbitrarily pick $\widehat{\beta}_0 = 1823$. Since $y_1 = 151$, we could then pick $\widehat{\varepsilon}_1 = -1672$ and we'd have $y_1 = 151 = 1823 - 1672$. Similarly, we could pick $\widehat{\varepsilon}_2, \ldots, \widehat{\varepsilon}_{76}$ to accommodate our choice of $\widehat{\beta}_0$.

Clearly, we need a more principled method for obtaining our parameter estimates. So what's the optimal principled way for obtaining parameter estimates? The hardly surprising answer is that this depends on what you mean by 'optimal'.

One sensible way of defining 'optimal' is that β_0 needs to be estimated in such a way that the sum of the absolute residuals, i.e.,

$$\sum_{i=1}^{n} |\widehat{\varepsilon}_i| = |\widehat{\varepsilon}_1| + \cdots + |\widehat{\varepsilon}_n|,$$

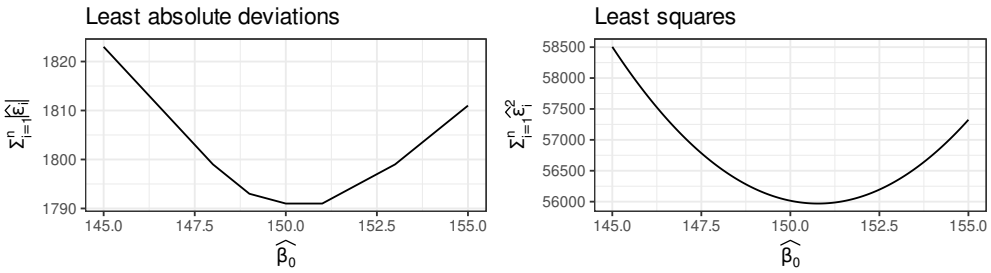

Figure 1.2: *Left:* If we estimate the parameter so as to minimise the sum of the absolute deviations, we obtain the sample median. *Right:* If we estimate the parameter so as to minimise the sum of the squared deviations, we obtain the sample mean.

is as small as possible. This is known as the **method of least absolute deviations**. If we pick $\widehat{\beta}_0 = 135$, then the sum of the absolute residuals equals 1993:

```
sum(abs(d$GJT - 135))

[1] 1993
```

If we pick $\widehat{\beta}_0 = 148$, the sum of the absolute residuals equals 1799:

```
sum(abs(d$GJT - 148))

[1] 1799
```

So by this definition of 'optimal', $\widehat{\beta}_0 = 148$ is a better choice than $\widehat{\beta}_0 = 135$. We can try out a bunch of possible values for $\widehat{\beta}_0$ and see which one minimises the sum of the absolute residuals. The left graph of Figure 1.2 shows exactly this. Note that for the GJT data, the sum of absolute residuals is minimised for $\widehat{\beta}_0$ values between 150 and 151. It's not a coincidence that the medians of the GJT comprise the interval $[150, 151]$: if we estimate $\widehat{\beta}_0$ by minimising the sum of the absolute residuals, then we obtain the sample medians! For a proof, see Schwertman et al. (1990).[1]

Another sensible definition of 'optimal' is that β_0 needs to be estimated in such a way that the sum of the squared residuals, i.e.,

$$\sum_{i=1}^{n} \widehat{\varepsilon}_i^2 = \widehat{\varepsilon}_1^2 + \cdots + \widehat{\varepsilon}_n^2,$$

is as small as possible. This is the **method of least squares**. As shown in the right graph of Figure 1.2, the optimal $\widehat{\beta}_0$ value when using least squares is about 150.8 for the GJT data. The mean of the GJT data is about 150.8, too, and of course, this isn't a coincidence: When we

[1] *A* median of a list of values $(x_1, x_2, \ldots, x_n)$ is any value r such that at least half of the x_i values are at least as large as r and at least half of them are at most as large as r. When talking about *the* median, we refer to the average of the smallest such value r and the largest such value.

estimate β_0 using least squares, we obtain the sample mean! This can be proved using a bit of secondary school mathematics; the footnote contains two such proofs.[2]

Both computationally and mathematically, the method of least squares is the easiest to work with. Unless mentioned otherwise, the parameter estimates in the general linear model are obtained using this method ('**ordinary least squares**', or OLS). If you want to use the method of least absolute deviations, median (or quantile) regression models are available. Further, in several applications in machine learning, other optimality criteria are quite common.

[2]Consider the expression

$$\sum_{i=1}^{n}(y_i - \widehat{\beta}_0)^2$$

as a function of $\widehat{\beta}_0$. This function is the sum of n quadratic functions, so it is itself a quadratic function. We're interested in the value of $\widehat{\beta}$ that minimises this function. As you may recall from secondary school, we can find this value by differentiating the function, setting the derivative to zero, and solving the resulting expression for $\widehat{\beta}_0$. So let's do this.

First compute the derivative of $\sum_{i=1}^{n}(y_i - \widehat{\beta}_0)^2$ with respect to $\widehat{\beta}_0$:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\widehat{\beta}_0}\sum_{i=1}^{n}(y_i - \widehat{\beta}_0)^2 &= \frac{\mathrm{d}}{\mathrm{d}\widehat{\beta}_0}\sum_{i=1}^{n}\left(y_i^2 - 2y_i\widehat{\beta}_0 + \widehat{\beta}_0^2\right) && [(a-b)^2 = a^2 - 2ab + b^2] \\
&= \sum_{i=1}^{n}\left(\frac{\mathrm{d}}{\mathrm{d}\widehat{\beta}_0}y_i^2 - \frac{\mathrm{d}}{\mathrm{d}\widehat{\beta}_0}2y_i\widehat{\beta}_0 + \frac{\mathrm{d}}{\mathrm{d}\widehat{\beta}_0}\widehat{\beta}_0^2\right) && [\text{The derivative of a sum is the sum of the derivatives.}] \\
&= \sum_{i=1}^{n}\left(0 - 2y_i + 2\widehat{\beta}_0\right) && [\text{Basic derivatives.}] \\
&= -\left(2\sum_{i=1}^{n}y_i\right) + 2n\widehat{\beta}_0.
\end{aligned}$$

The above is equal to zero if and only if $2\sum_{i=1}^{n}y_i = 2n\widehat{\beta}_0$. Dividing both sides by $2n$, we obtain

$$\widehat{\beta}_0 = \frac{1}{n}\sum_{i=1}^{n}y_i,$$

that is, the arithmetic mean of the y_i values. This completes the first proof.

An alternative proof that doesn't rely on taking derivatives is this. Write $\overline{y} := \left(\sum_{i=1}^{n}y_i\right)/n$, i.e., the sample mean. Note that

$$\begin{aligned}
\sum_{i=1}^{n}(y_i - \widehat{\beta}_0)^2 &= \sum_{i=1}^{n}\left((y_i - \overline{y}) + (\overline{y} - \widehat{\beta}_0)\right)^2 && [\text{Add 0.}] \\
&= \sum_{i=1}^{n}\left((y_i - \overline{y})^2 + 2(y_i - \overline{y})(\overline{y} - \widehat{\beta}_0) + (\overline{y} - \widehat{\beta}_0)^2\right) && [(a+b)^2 = a^2 + 2ab + b^2] \\
&= \sum_{i=1}^{n}(y_i - \overline{y})^2 + 2(\overline{y} - \widehat{\beta}_0)\left(\sum_{i=1}^{n}y_i - n\overline{y}\right) + n(\overline{y} - \widehat{\beta}_0)^2 && [\text{Write out sum.}] \\
&= \sum_{i=1}^{n}(y_i - \overline{y})^2 + n(\overline{y} - \widehat{\beta}_0)^2 && [\sum_{i=1}^{n}y_i = n\overline{y}] \\
&\geq \sum_{i=1}^{n}(y_i - \overline{y})^2,
\end{aligned}$$

with equality if and only iff $\widehat{\beta}_0 = \overline{y}$. This completes the second proof.

1.2 General linear models in R

We can use the `lm()` function to build linear models whose parameters are estimated using least squares. This function takes a formula in which the outcome is listed in front of the tilde and the predictors are listed after it. In the present case, we don't have any predictors, so we just write a 1 instead:

```
mod.lm <- lm(GJT ~ 1, data = d)
```

The estimated parameters can be printed by typing the name of the model:

```
mod.lm


Call:
lm(formula = GJT ~ 1, data = d)

Coefficients:
(Intercept)
        151
```

A more compact overview can be obtained using `coef()`:

```
coef(mod.lm)

(Intercept)
     150.78
```

Don't be confused by differences in the formatting (151 vs. 150.78). This is merely a matter of rounding the output; the underlying representation of these estimates is the same.

We can obtain the residuals, that is, the $\widehat{\varepsilon}_i$ values as follows:

```
resid(mod.lm) # output not shown
```

If we subtract the residuals from the observed values, we obtain the (so-called) **predicted values**. (A term I'm not entirely chuffed with.) Since

$$y_i - \widehat{\varepsilon}_i = y_i - (y_i - \widehat{\beta}_0) = \widehat{\beta}_0,$$

this results in 76 repetitions of $\widehat{\beta}_0$:

```
# output not shown:
d$GJT - resid(mod.lm)
# alternatively (output not shown):
predict(mod.lm)
```

2 Quantifying uncertainty

What we've done so far is *estimate* the β_0 parameter that we're interested in. While we now have a principled approach for obtaining this estimate, the resulting value will depend on the data that we've collected. Usually, the estimate is based only on a sample from some larger population. As a result, our estimate of β_0 won't exactly coincide with the true but unknown value of β_0, that is, there is some inherent uncertainty about our estimate.

2.1 The sampling distribution of the sample mean

Before continuing with our discussion of linear models, let's consider by how much means of different samples taken from the same distribution tend to differ from one another. To make matters more concrete, let us assume that we're interested in the GJT variable and that this variable has the exact same distribution in the population of interest as it does in DeKeyser et al.'s sample (Figure 1.3, top row). This distribution has expectation (mean) $\mu = 150.78$ and population variance $\sigma^2 = 746.26$:[3]

```
(pop_mean <- mean(d$GJT)) # expectation

[1] 150.78

(pop_var <- mean(d$GJT^2) - mean(d$GJT)^2) # variance

[1] 736.44
```

A **simple random sample** of size n from a population consists of n independently drawn values following this population's distribution. In order to study the behaviour of the sample mean, we now draw a large number (here: 20,000) of independent simple random samples (i.e., with replacement) of size 20 from this population (Figure 1.3, middle row), compute their means, and inspect the distribution of these sample means (bottom row).

Observe, first, that the distribution of sample means has itself an expected value (i.e., a mean), $\mu_{\bar{x}}$, that is identical to mean μ of the population from which the samples were drawn. (The mean of the empirical distributions of these 20,000 sample means won't exactly equal $\mu_{\bar{x}} = \mu$, but any slight discrepancy is due solely to our working with a limited number of sample means.) If the original population has a mean, then this is always true – regardless of the shape of the distribution, and even for samples with observations that are not independent.[4]

Second, if the original distribution has finite variance, the distribution of the sample means

[3] I don't use R's `var()` function for computing the population variance since it computes the sample variance rather than the population variance. The formulae are slightly different.

[4] Not all distributions have means. The textbook example of a distribution without a mean is the Cauchy distribution. That said, we mustn't worry about such theoretical counterexamples.

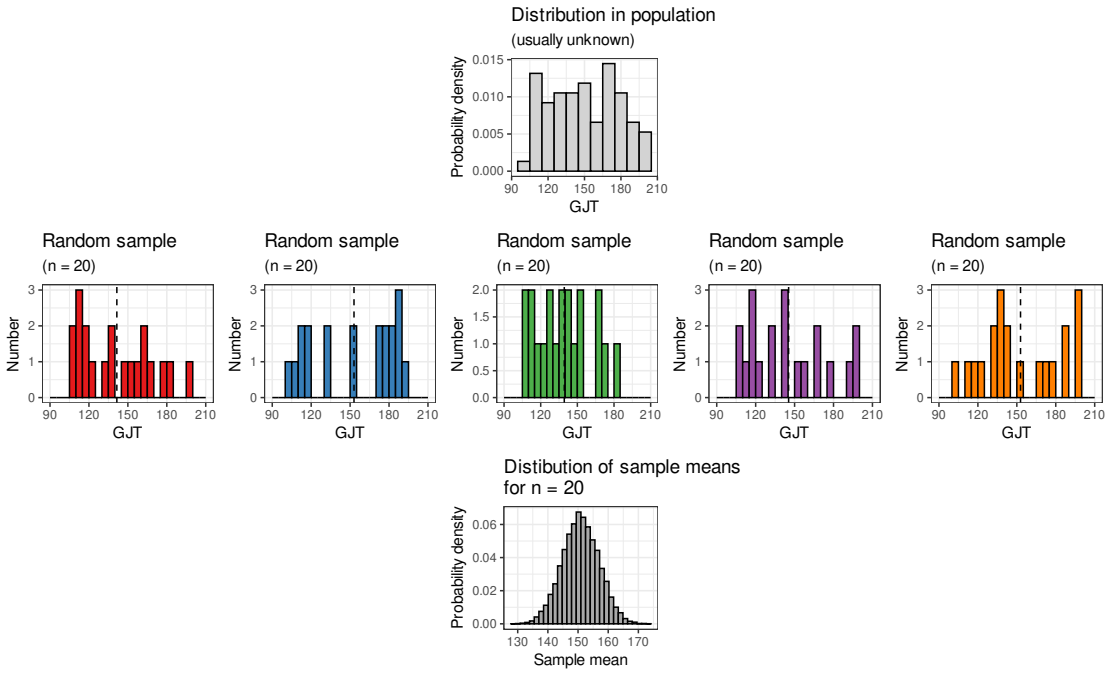

Figure 1.3: The top row shows the distribution that we're (for the sake of the illustration) interested in. Imagine that we draw a large number of independent simple random samples with the same size (here: $n = 20$) from this distribution and compute their means. The middle row shows five such random samples, and their means are highlighted by the dashed vertical line. The bottom row shows the distribution of these sample means. In this example, the latter distribution seems to be more or less approximately distributed, but this isn't necessarily the case.

has itself finite variance, $\sigma^2_{\overline{x}}$.[5] As the next exercise will show by means of a simulation, we have

$$\sigma^2_{\overline{x}} = \frac{\sigma^2}{n},$$

where n is the size of the samples. Hence, the standard deviation of the distribution of sample means is given by $\sigma/\sqrt{n}$. This again is true regardless of the shape of the distribution that the data were drawn from.

Exercise 1.3 (Expectation and variance of the sample mean)**.** Throughout this lecture series, we'll rely heavily on simulations and related methods. As a warm-up, consider the following code snippet. It draws a random sample (with replacement) of size 20 from the GJT data and compute its mean. It repeats this process 20,000 times and returns a vector of the 20,000 sample means so computed. Then it computes the mean and the variance of these means.

```
sample_means <- replicate(20000, {
  mean(sample(d$GJT, 20, replace = TRUE))
})
mean(sample_means)
var(sample_means)
```

Adapt this code snippet in order to verify the identities $\mu_{\overline{x}} = \mu$ and $\sigma^2_{\overline{x}} = \frac{\sigma^2}{n}$ for sample sizes $n = 5, 10, 20, 100$. Also briefly explain why the results you obtain aren't *exactly* what the formulas predict. ♢

A third observation is that the distribution of the sample means seems to be more or less normally distributed. This is the celebrated **central limit theorem** at play.

Theorem 1.4 (Central limit theorem)**.** Let $X_1, X_2, \ldots, X_n$ be independent observations drawn at random from a distribution with expectation μ and finite variance σ^2 (i.e., finite standard deviation σ). Then the distribution of the statistic

$$\sqrt{n}\left(\overline{X} - \mu\right), \tag{2}$$

where $\overline{X} := 1/n \sum_{i=1}^{n} X_i$, converges to a normal distribution with mean 0 and variance σ^2 (written as $\mathcal{N}(0, \sigma^2)$) as $n \to \infty$. ♢

The central limit theorem (or CLT among friends) does not state that the distribution of the statistic in (2) mean *is* normal for any one sample size n. Rather, it offers the guarantee that the difference between the distribution of this statistic and $\mathcal{N}(0, \sigma^2)$ becomes ever smaller for

[5] Not all distribution have a variance. This includes all distributions without a mean (such as the Cauchy) and some other cases such as the Student's t distribution with two degrees of freedom. Again, we needn't worry about such cases here.

larger sample sizes. That is, if we know the mean and the variance of the distribution from which we draw a random sample, we may use the properties of the normal distribution to give *approximate* answers as to where this statistic will end up. The standard proof of the CLT is too technical for the present lectures.

Remark 1.5. Let a, b be real numbers. If Z is $\mathcal{N}(0,1)$-distributed, then $a + bZ$ is $\mathcal{N}(a, b^2)$-distributed. Hence, if the statistic in (2) is approximately $\mathcal{N}(0, \sigma^2)$-distributed for a given sample size n, then $\overline{X}$ is approximately approximately $\mathcal{N}(\mu, \sigma^2/n)$-distributed, confirming our earlier observations about the mean and variance of the distribution of the sample mean. ♢

The following exercises may help you develop your intuition about the workings of the central limit theorem. They are entirely optional.

Example 1.6 (CLT for discrete uniform distribution)**.** A dice throw with a standard six-sided dice can be modelled using a discrete uniform distribution on $\{1, 2, 3, 4, 5, 6\}$, i.e., a distribution that assigns equal probability to all of these values. This distribution has mean $\mu = 3.5$ and variance $\sigma^2 = (6^2 - 1^2)/12 = 35/12$. (You can look up this information on Wikipedia.)

If we draw a random sample of $n = 3$ observations and compute their mean, then the sampling distribution of this mean has expectation $\mu_{\overline{x}} = \mu = 3.5$ and variance $\sigma^2_{\overline{x}} = \sigma^2/n = 35/(12 \cdot 3)$. The code below generates 20,000 such sample means and plots their distribution as a histogram (Figure 1.4, left). Additionally, it plots the empirical cumulative distribution function (ecdf) of these sample means, that is, the proportion of the 20,000 sample means that fall below some number r (Figure 1.4, right, in red). The blue line shows what proportion of sample means would fall below r if these perfectly followed a $\mathcal{N}(3.5, 35/(12 \cdot 3))$ distribution. (In the R code, note that the normal distribution in R is parametrised by its mean and standard deviation rather than by its mean and variance. Hence, we need to take the square root of the variance.)

```
n <- 3 # set sample size
mu <- 3.5
sd_n <- sqrt(35/(12 * n))
sample_means <- replicate(20000, {
  mean(sample(1:6, n, replace = TRUE)) # generate n random values from 1, ..., 6
})

par(mfrow = c(1, 2)) # two plots side by side
hist(sample_means, xlab = expression(bar(x)), breaks = 30,
     main = paste0("Distribution of sample means for n = ", n))

plot(ecdf(sample_means), pch = ".", col = "red",
     xlab = "r",
```

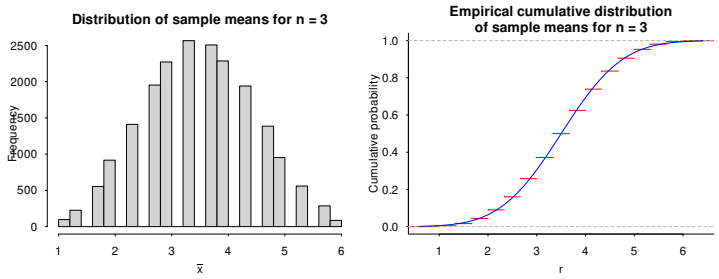

Figure 1.4: *Left:* Histogram of 20,000 means of samples of size $n = 3$ from a discrete uniform distribution on $1, \ldots, 6$. *Right:* The empirical cumulative distribution (in red) shows the proportion of sample means below a value r (red). The blue line shows the proportions expected according to the CLT-based normal approximation.

```
     main = paste0("Empirical cumulative distribution\nof sample means for n = ", n),
     ylab = "Cumulative probability")
curve(pnorm(x, mean = mu, sd = sd_n), # add cdf of normal
      add = TRUE, col = "blue")

par(mfrow = c(1, 1)) # back to plotting one plot per row
```

Clearly, the sampling distribution of the sample mean is far from normal for $n = 3$. The following commands compute the proportion of sample means lie in the interval $[3.45, 3.55]$, and also estimate this proportion under the CLT-based normal approximation:

```
mean(sample_means >= 3.45 & sample_means <= 3.55)
```

```
[1] 0
```

```
pnorm(3.55, mean = mu, sd = sd_n) - pnorm(3.45, mean = mu, sd = sd_n)
```

```
[1] 0.040443
```

Due to the discreteness of the population, it's completely impossible to find a sample mean in the interval $[3.45, 3.55]$. The normal approximation doesn't capture this fact.

For a given distribution with distribution function F and a number $\gamma \in (0, 1)$, the **quantile function** $F^{-1}(\gamma)$ outputs the smallest number r such that $F(r) \geq \gamma$. The quantile functions for normal distributions are available in R as `qnorm()`, whereas the `quantile()` function can be applied to an empirical distribution. The following commands can be used to look up between which two values 95% of the sample means lie – be it in the simulation (`quantile()`) or according to the normal approximation (`qnorm()`).

```
quantile(sample_means, probs = c(0.025, 0.975))
```

```
  2.5%  97.5%
1.6667 5.3333
```

```
qnorm(p = c(0.025, 0.975), mean = mu, sd = sd_n)
```

```
[1] 1.5674 5.4326
```

That is, in this case, the normal approximation would indicate that the sample means are more spread out than they actually are.

Finally, we compute the proportion of sample means that lie below 0.9999 as well as the normality-based estimate; note that the true proportion is necessarily 0 since the lowest value that could be observed was 1:

```
mean(sample_means <= 0.99999)
```

```
[1] 0
```

```
pnorm(0.99999, mean = mu, sd = sd_n)
```

```
[1] 0.0056148
```

Use and adapt the code above to inspect how well the sampling distribution of the mean of $n = 2, 8, 32, 128$ dice throws can be approximated by a suitable normal distribution. ♢

Exercise 1.7 (CLT for log-normal distribution)**.** A random variable $X > 0$ follows a log-normal distribution with parameters a, b (LogNorm(a, b^2)) if the random variable $\log(X)$ follows a $\mathcal{N}(a, b^2)$ distribution. Log-normally distributed variables have a right-skewed distribution, unlike the uniform distributions from the previous examples/exercises, which have no skew. Run the following code to inspect the sampling distribution of the mean of a sample log-normally distributed observations and see how well the normal approximation works for such skewed data. Play around with the `n`, `a` and `b` values. (To get an idea of the shape of the log-normal distribution, use `n <- 1`.)

```
# Set simulation parameters
n <- 20
a <- 1
b <- 1

# Mean and standard deviation of normal approximation
mu <- exp(a + b^2/2)
sd_n <- sqrt((exp(b^2) - 1)*exp(2*a + b^2)/n)

# Sample from distribution
```

```
sample_means <- replicate(20000, {
  mean(rlnorm(n, a, b))
})
```

```
quantile(sample_means, probs = c(0.025, 0.975))
```

```
  2.5%  97.5%
2.5660 7.6615
```

```
qnorm(c(0.025, 0.975), mean = mu, sd = sd_n)
```

```
[1] 1.9070 7.0564
```

Note that even for pretty large sample sizes, the normal approximation isn't very good. The CLT offers the guarantee that distribution of sample means becomes eventually indistinguishable from a normal distribution, as $n \to \infty$. However, the rate of convergence depends heavily on the shape of the distribution from which the samples are drawn. ♢

In the GJT example, however, the CLT-based normal approximation is excellent even for a sample size of $n = 20$; see the output below as well as Figure 1.5.

```
n <- 20
sample_means <- replicate(20000, {
  mean(sample(d$GJT, n, replace = TRUE))
})

# Cumulative proportions
plot(ecdf(sample_means), pch = ".", col = "red",
     xlab = "r",
     main = paste0("Empirical cumulative distribution\nof sample means for n = ", n),
     ylab = "Cumulative probability")
curve(pnorm(x, mean = pop_mean, sd = sqrt(pop_var / n)), add = TRUE, col = "blue")
```

```
mean(sample_means >= 149 & sample_means <= 151)
```

```
[1] 0.1302
```

```
pnorm(151, mean = pop_mean, sd = sqrt(pop_var / n)) -
  pnorm(149, mean = pop_mean, sd = sqrt(pop_var / n))
```

```
[1] 0.12984
```

```
quantile(sample_means, probs = c(0.025, 0.975))
```

```
  2.5%  97.5%
```

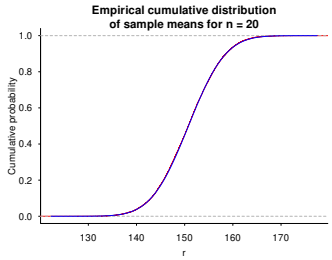

Figure 1.5: The empirical cumulative distribution (in red) shows the proportion of sample means below a value r (red). The blue line shows the proportions expected according to the CLT-based normal approximation. The lines can hardly be distinguished as the normal approximation is so good in this case.

```
138.85 162.55
```

```
qnorm(c(0.025, 0.975), mean = pop_mean, sd = sqrt(pop_var / n))
```

```
[1] 138.88 162.67
```

Now imagine that you were told just the difference between, say, the 97.5th and the 2.5th percentile of the sampling distribution of the sample mean, but you didn't know anything else about it. If this difference is small, then this implies that any given individual sample mean stands a good chance of being pretty close to the actual population mean; if this difference is large, individual sample means can't be relied on to be good approximations of the actual population mean. That is, a measure of the width of the sampling distribution would be a useful indication of the degree of uncertainty we have when estimating the population mean. Instead of using differences between percentiles, other measures can be used, for instance, the standard deviation of the sampling distribution.

2.2 Why we need assumptions

It's not immediately obvious why the insights from the previous subsection are of any use to us: they assume that we know the properties of the distribution we sample from. But if we knew these properties, we wouldn't be doing any sampling! In order to make some headway, we need to make a few assumptions about our data.

The first assumption we'll make is that the errors, i.e., the ε_i values, are distributed **identically** and **independently**. This assumption is often abbreviated as the i.i.d. assumption (for 'identically and independently distributed').

The 'identical' bit means that we assume that there exists a single distribution from which all errors were drawn, that is,

$$\varepsilon_i \sim \text{some distribution}$$

for $i = 1, \ldots, n$. The only assumption about this distribution we'll make for now is that it has an expected value (i.e., a mean) 0 and finite variance. In the present setting, this is equivalent to assuming that the y_i values are drawn from a distribution with expectation $\mu = \beta_0$ and finite variance. We don't make any further assumptions about this distribution yet; we're merely assuming that it is not the case that some errors are drawn from, say, a uniform distribution spanning from -6 to 6 whereas some other errors are drawn from a normal distribution with mean 0 and a standard deviation of 3.

The independence assumption entails that if we know the value of one error, then this doesn't provide us with any more information about any of the remaining errors. (The formal definition of stochastic independence is more involved, but essentially boils down to this.) To make this more concrete, consider the following example. Let's say you wanted to estimate the average length of the [u] vowel in the Ghent vernacular. To this end, you have 25 informants from Ghent read out 50 words containing [u]. It is conceivable that some informants tend to produce longer [u] sounds than others, that is, once you obtain a positive error (corresponding to longer than average vowel length) for a single production from a single speaker, chances are the other productions from this speaker will also have positive errors. That is, once you've learnt something about the value of one specific error, you can make more informed guesses about the values of some other error, too. (Recall that 'error' merely refers to the discrepancy between the observation y_i and the shared communality $\mu = \beta_0$.) Similarly, it is conceivable that [u] sounds tend to be longer in some words or phonological contexts than others. If so, the error for a single word produced by a single speaker may give you some clues as to the error for other productions of the same word. In sum, the $25 \times 50 = 1250$ data points that this study would produce would violate the independence assumption if we were to analyse them with the tools we're going to discuss.

Problems with dependencies between errors can be tackled in a number of ways, with one popular approach since 2008 or so being the use of mixed-effects models. Problems having to do with different errors stemming from different distributions usually aren't as detrimental, but, if necessary, they can be tackled in a few ways (e.g., generalised least squares models, nonparametric bootstrapping; see Zuur et al., 2009, and Hesterberg, 2014, respectively, for accessible introductions).

We'll have more to say about modelling assumptions later. Let's now turn to the matter of estimating the extent of the uncertainty in our parameter estimates.

2.3 The bootstrap

We're currently assuming that the errors are identically and independently distributed according to some non-weird distribution. To make some headway, we need to add a further assumption about the shape of the distribution from which the errors were drawn. One possible

approach is to assume that the (estimated) residuals we have give us a reasonable approximation of the distribution of the errors. Then, we can use the **bootstrap** technique (Efron, 1979; Efron & Tibshirani, 1993) to quantify the uncertainty in our parameter estimates. While a couple of flavours of the bootstrap exist, we'll consider the **semi-parametric** version, which works as follows:

1. Compute $\widehat{\beta}_0$ and in doing so obtain a vector $\widehat{\varepsilon}$ containing values $\widehat{\varepsilon}_1$ through $\widehat{\varepsilon}_n$.
2. Sample with replacement an n-valued vector from $\widehat{\varepsilon}$ and label it $\widehat{\varepsilon}^*$. The vector $\widehat{\varepsilon}^*$ may contain some values from $\widehat{\varepsilon}$ several times and others not at all.[6]
3. Create a new vector of outcome values: $y_i^* = \widehat{\beta}_0 + \widehat{\varepsilon}_i^*$. This new vector is referred to as a bootstrap replicate.
4. Estimate β_0 again but this time using y^*. Call the new estimate $\widehat{\beta}_0^*$.
5. Do steps 2–4 a couple of thousand times and in doing so obtain a distribution of bootstrapped β_0 estimates. Always use the *original* $\widehat{\beta}_0$ estimate in step 3, not the previously computed $\widehat{\beta}_0^*$ estimate.

Figure 1.6 illustrates this procedure. Hopefully, this figure helps to bring across the key idea behind the bootstrap, which is this: In an ideal world, you'd know the distribution from which the observations (or at least the errors) were sampled. Then, you could approximate the sampling distribution of the estimated parameter to arbitrary precision by repeatedly generating samples from this distribution and reestimating the parameter (like we did in Figure 1.3). Then, you'd compute, say, the standard deviation of the sampling distribution or some other measure of this distribution's width. Lacking such knowledge about the original distribution, you instead use the observed (empirical) distribution of the residuals instead. (This bootstrap variant is called semi-parametric since we use a parametrised model to generate the predicted values but not to generate the new errors.)

What Figure 1.7 shows is that none of the distributions of the bootstrapped means is identical to the true sampling distribution of the sample mean. In particular, the bootstrap distributions are centred are the means of the samples they are based on, and samples with less empirical variance give rise to narrower bootstrap distributions. But *on average*, the bootstrap distributions seem to give decent approximation of the shape and width of the true sampling distribution. This is also underscored by the numerical summary in Table 1.1.

[6] For large n, about 36.7% entries in $\widehat{\varepsilon}$ won't occur in $\widehat{\varepsilon}^*$.

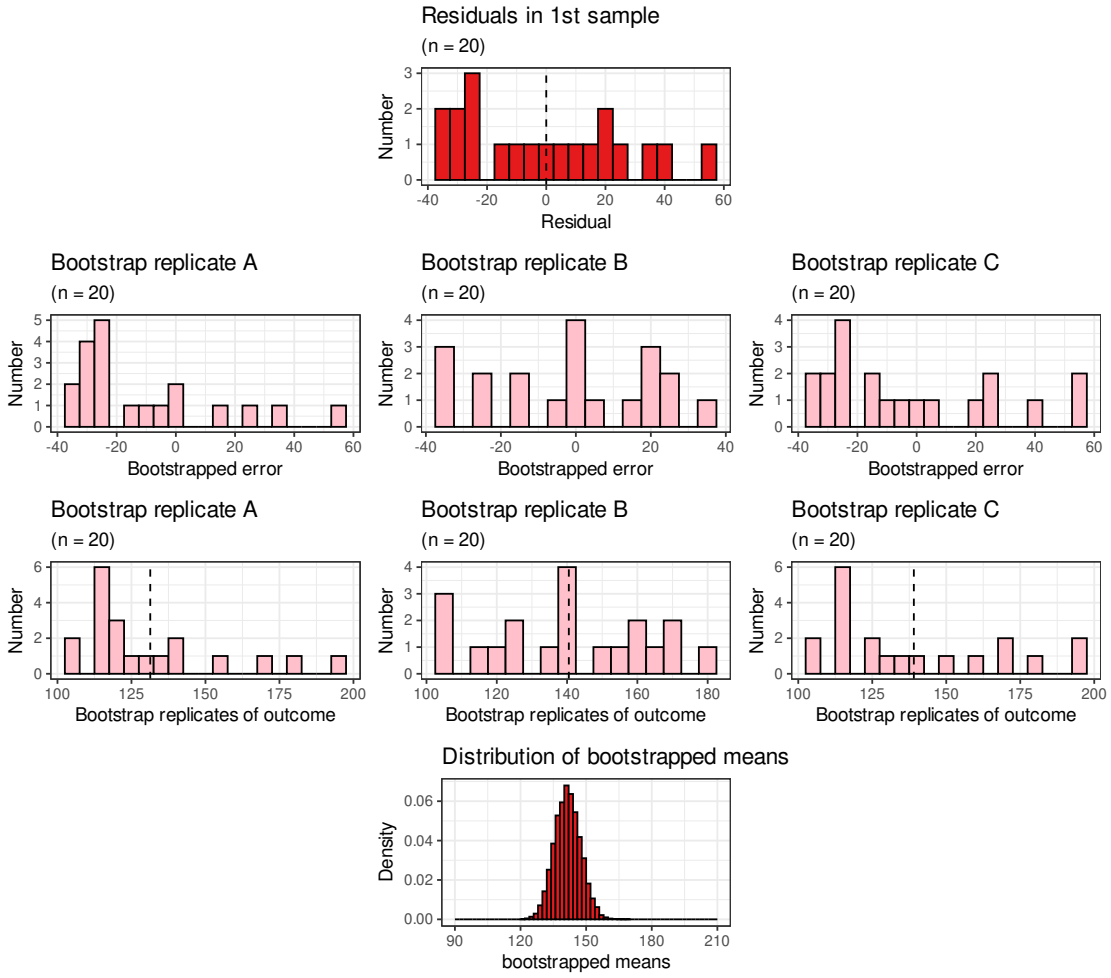

Figure 1.6: *Top row:* The first (red) sample from Figure 1.3 serves as our point of departure. We've estimated β_0 using this sample and have obtained 20 residuals. *Second row:* We generate a large number of new samples ('bootstrap replicates') containing 20 values by resampling with replacement from these 20 residual values. By way of illustration, three such new samples are shown. *Third row:* For each newly created sample, we combine the resampled residuals with the $\widehat{\beta}_0$ value and reestimate β_0 using these new values (highlighted with a dashed vertical line). *Bottom row:* We consider the distribution of all these new estimates of β_0 to get a sense of how strongly such estimates vary from sample to sample for samples of size $n = 20$.

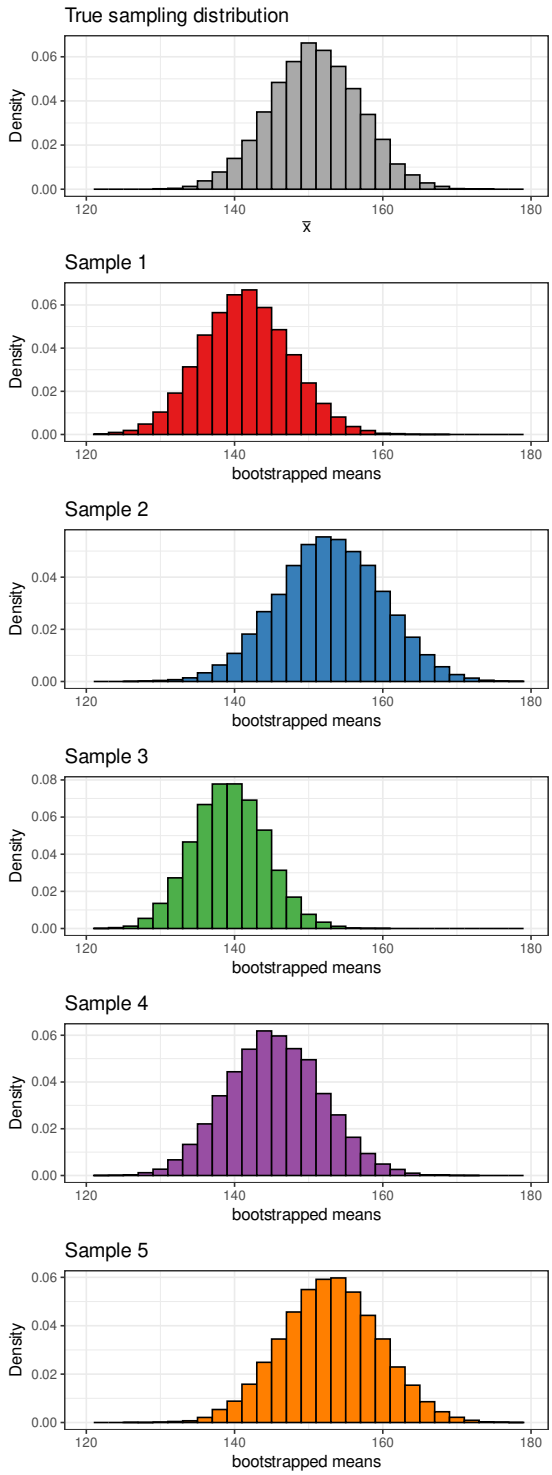

Figure 1.7: The true sampling distribution of the sample mean of 20 GJT observations as well as the distributions of bootstrapped means based on the five samples from Figure 1.3.

Table 1.1: Standard deviation (SD), 0.025 and 0.975 quantiles (i.e., 2.5th and 97.5th percentiles) and the difference between them for the true sampling distribution as well as for the five bootstrapped distributions in Figure 1.7. Slight inconsistencies between the percentiles and the difference between them are due to rounding.

Distribution of sample means	SD	2.5th percentile	97.5th percentile	Interval width
Actual (unknown)	6.1	139	163	24
Bootstrap based on sample 1	6.0	130	154	23
Bootstrap based on sample 2	7.0	139	167	28
Bootstrap based on sample 3	4.9	130	149	19
Bootstrap based on sample 4	6.5	133	158	25
Bootstrap based on sample 5	6.6	140	166	26

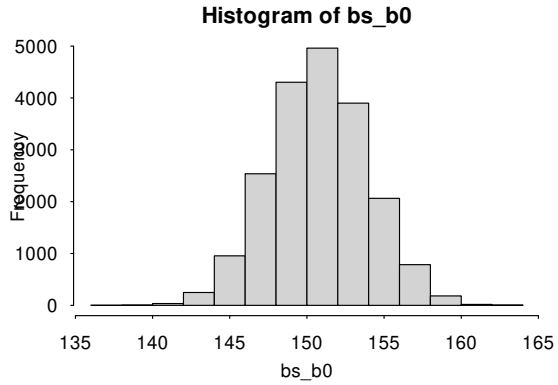

Figure 1.8: A quick histogram of the bootstrapped $\widehat{\beta}_0$ estimates.

Let's return to the original problem: We want to estimate the uncertainty we ought to have the mean of the GJT values based on a sample mean of 76 such observations. Let's apply the bootstrap procedure:

```
# Step 1 (model was already fitted)
d$Prediction <- predict(mod.lm)
d$Residual <- resid(mod.lm)

n_bootstrap <- 20000
bs_b0 <- vector(length = n_bootstrap)

for (i in 1:n_bootstrap) {
  # Step 2
  bs_resid <- sample(d$Residual, replace = TRUE)
  # Step 3
  d$bs_outcome <- d$Prediction + bs_resid
  # Step 4
  bs_mod <- lm(bs_outcome ~ 1, data = d)
  bs_b0[[i]] <- coef(bs_mod)[[1]]
}
```

A quick no-frills histogram of the $\widehat{\beta}_0^*$ values shows that these approximately form a normal distribution (Figure 1.8).

```
hist(bs_b0)
```

The standard deviation of the bootstrapped $\widehat{\beta}_0^*$ values provides us with an estimate of the standard deviation of the sampling distribution. An estimate of the standard deviation of a sampling distribution is referred to as a **standard error**, though this term is sometimes also

used more loosely to refer to the actual standard deviation of a sampling distribution:

```
sd(bs_b0)
```

```
[1] 3.1286
```

Since $\sigma_{\bar{x}} = \sigma/\sqrt{n}$, an alternative standard error can be obtained by plugging in the sample standard deviation s for σ:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}} \approx \frac{s}{\sqrt{n}} :$$

```
sd(d$GJT) / sqrt(nrow(d))
```

```
[1] 3.1336
```

Both of these formulae tend to slightly underestimate the true standard deviation of the sampling distribution, more so for small samples. For the sample mean, the latter formula is more accurate. The advantage of the bootstrap for obtaining standard errors, though, is that it can also be used when there aren't any neat formulae for computing standard errors like there is for the sample mean.

An *approximate* 95% **confidence interval** around the parameter estimate can be constructed by looking up the 2.5th and 97.5th percentile of the distribution of the $\widehat{\beta}_0^*$ values:

```
quantile(bs_b0, probs = c(0.025, 0.975))
```

```
  2.5%  97.5%
144.68 156.95
```

This method for obtaining confidence intervals is known as the **percentile method**; other methods exist.

Alternatively, we could use the central limit theorem: look up the 2.5th and 97.5th percentiles of the standard normal distribution (with `qnorm()`), multiply these by the estimated standard error, and translate the results by the parameter estimate. Here, we use the more accurate standard error formula, but in cases where it isn't available, the bootstrap-based estimate will do.

```
coef(mod.lm)[[1]] + qnorm(c(0.025, 0.975)) * sd(d$GJT) / sqrt(nrow(d))
```

```
[1] 144.63 156.92
```

In this example, the standard error estimates and confidence intervals produced by the bootstrap and those produced by the central limit theorem are essentially the same. This is a consequence of the distribution of the $\widehat{\beta}_0^*$ values being normal. But this distribution doesn't *have* to be normal. Indeed, what's nice about the bootstrap is that it can be used even when it

is doubtful that the sampling distribution of the parameter estimate is approximately normal.

Example 1.8 (Standard error for the sample median)**.** We've discussed the bootstrap in a GLM setting, but it can be applied much more broadly. Let's say we're interested in the median of the GJT values and we want to obtain a standard error for it. The code above only needs to be adapted slightly.

```
# Step 1
gjt_median <- median(d$GJT)
d$Prediction <- gjt_median
d$Residual <- d$GJT - gjt_median # difference between observed and predicted

n_bootstrap <- 20000
bs_b0 <- vector(length = n_bootstrap)

for (i in 1:n_bootstrap) {
  # Step 2
  bs_resid <- sample(d$Residual, replace = TRUE)
  # Step 3
  d$bs_outcome <- d$Prediction + bs_resid
  # Step 4
  bs_b0[[i]] <- median(d$bs_outcome)
}
hist(bs_b0)
# plot(ecdf(bs_b0)) # ecdf of bootstrapped medians; not shown
sd(bs_b0) # standard error for median
```

```
[1] 5.3153
```

Figure 1.9 shows the distribution of the bootstrapped medians. Note that it is not at all normal. While no nice general formula for computing a standard error for the median is available, the bootstrap provides us with a useful approximation. A (conservative) procedure for computing confidence intervals is available for the median, but the following also constitutes an approximate 95% confidence interval:

```
quantile(bs_b0, probs = c(0.025, 0.975))
```

```
 2.5% 97.5%
  140   163
```

♢

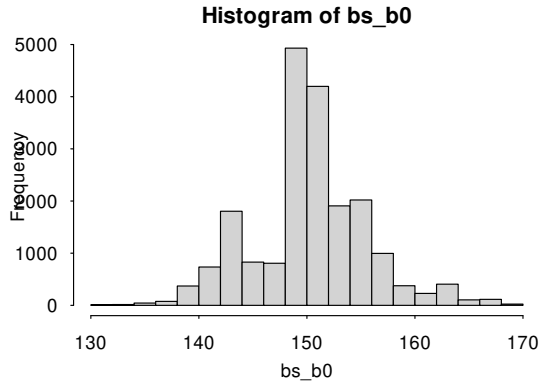

Figure 1.9: A quick histogram of the bootstrapped estimates for the median. Note that the distribution is far from normal!

2.4 The parametric bootstrap

In the previous section, we assumed that the distribution of the estimated residuals gives us a reasonable approximation of the distribution from which the residuals were drawn. Alternatively, we could make a different assumption about the residual distribution, for instance, that this distribution is normal. Note the difference between this assumption and the assumption we made earlier. Now we're assuming that the residuals are randomly and independently drawn from a normal distribution; earlier, we assumed that the residuals were randomly and independently sampled from *some* distribution, and it was the (estimated) sampling distribution of $\widehat{\beta}_0$ that *happened* to be roughly normally distributed.

Normal distributions are defined by two parameters: their mean μ and their variance σ_ε^2 (or their standard deviation σ_ε). Since $\widehat{\beta}_0$ is equal to the sample mean, the mean of the residuals is always equal to 0.[7] We can write down our assumptions like so:

$$
\begin{aligned}
y_i &= \beta_0 + \varepsilon_i, \\
\varepsilon_i &\overset{\text{i.i.d.}}{\sim} \text{Normal}(0, \sigma_\varepsilon^2),
\end{aligned}
$$

for $i = 1, 2, \ldots, n$. The standard deviation of the error distribution can be estimated from the estimated residuals like so:

$$
\widehat{\sigma}_\varepsilon = \sqrt{\frac{1}{n-d} \sum_{i=1}^{n} \widehat{\varepsilon}_i^2},
$$

where d is the number of estimated β parameters (currently: $d = 1$, so $\widehat{\sigma}_\varepsilon$ is just the sample

[7] The mean of the deviations from the sample mean is always equal to zero:

$$
\frac{1}{n} \sum_{i=1}^{n} \widehat{\varepsilon}_i = \frac{1}{n} \sum_{i=1}^{n} (y_i - \widehat{\beta}_0) = \frac{1}{n} \left(\sum_{i=1}^{n} y_i \right) - n \cdot \frac{1}{n} \widehat{\beta}_0 = \widehat{\beta}_0 - \widehat{\beta}_0 = 0.
$$

standard deviation). This estimate can be obtained using

```
sigma(mod.lm)
```

```
[1] 27.318
```

The parametric bootstrap now proceeds similarly to the semi-parametric bootstrap. But instead of sampling the $\widehat{\varepsilon}^*$ values with replacement from $\widehat{\varepsilon}$, we sample them from a normal distribution with mean 0 and standard deviation $\widehat{\sigma}_\varepsilon$:

```
# Step 1
d$Prediction <- predict(mod.lm)
d$Residual <- resid(mod.lm)
sigma_mod.lm <- sigma(mod.lm)

n_bootstrap <- 20000
bs_b0 <- vector(length = n_bootstrap)

for (i in 1:n_bootstrap) {
  # Step 2
  bs_resid <- rnorm(n = nrow(d), sd = sigma_mod.lm)
  # Step 3
  d$bs_outcome <- d$Prediction + bs_resid
  # Step 4
  bs_mod <- lm(bs_outcome ~ 1, data = d)
  bs_b0[[i]] <- coef(bs_mod)[[1]]
}
```

We can again draw a histogram, which would again show that the $\widehat{\beta}_0^*$ values are normally distributed:

```
# not shown
hist(bs_b0)
```

The standard error estimate and confidence intervals are essentially the same as previously:

```
sd(bs_b0)
```

```
[1] 3.1361
```

```
quantile(bs_b0, c(0.025, 0.975))
```

```
  2.5%  97.5%
144.64 156.91
```

While we assumed that the residual distribution was normal here, we could in principle also apply the parametric bootstrap when making different assumptions about the residual distribution. We'd just have to plug in the corresponding sampling function in step 2. Further, an intermediate form between the semi-parametric bootstrap and the parametric bootstrap is to apply a smoothing procedure to the empirical distribution of the residuals (e.g., kernel density estimation) and sample from the smoothed distribution. But we won't concern ourselves with such alternative tools.

2.5 t-distributions

If we're happy to assume that the errors were drawn from a normal distribution, we can compute the standard error estimates and confidence intervals directly, without the bootstrap by relying on t-distributions. The formulae aren't too informative; let's skip straight to how to obtain the results in R:

```
summary(mod.lm)


Call:
lm(formula = GJT ~ 1, data = d)

Residuals:
   Min     1Q Median     3Q    Max
-46.78 -23.03  -0.28  23.22  47.22

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   150.78       3.13    48.1   <2e-16

Residual standard error: 27.3 on 75 degrees of freedom
```

For reasons that'll become clear shortly, the β_0 estimate is referred to as `(Intercept)`. The column with the `Std. Error` lists the standard error; the `t value` and `Pr(>|t|)` values correspond to the t- and p-values you may know from introductory statistics.

95% confidence intervals can easily be obtained using `confint()`:

```
confint(mod.lm)

             2.5 % 97.5 %
(Intercept) 144.53 157.02
```

Note that all these results correspond exactly to those obtained when running a one-sample

t-test on the data:

```
t.test(d$GJT)


 One Sample t-test

data:  d$GJT
t = 48.1, df = 75, p-value <2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 144.53 157.02
sample estimates:
mean of x
   150.78
```

Indeed, you can think of the one-sample t-test as but one instantiation of the general linear model.

If the assumption of normality isn't met, the confidence intervals and p-values may still be approximately valid, especially for large n. When in doubt, it doesn't hurt to also use a semi-parametric bootstrap to verify the confidence intervals obtained using t-distributions. While bootstrap procedures for computing p-values exist, they'd take us too far. Instead, you may want to look into permutation tests (see Ernst, 2004; Hesterberg, 2014, for overviews).

3 Excursion: Maximum likelihood estimation

There's an alternative method for obtaining parameter estimates: **maximum likelihood estimation** (MLE). You'll encounter this estimation method when you start to dabble in more complex methods such as generalised linear models. Feel free to skip this section and come back to it when you encounter this term on your statistical journey.

Let us assume that the data $y_1, \ldots, y_n$ are i.i.d. from a normal distribution with mean μ and variance σ^2. The probability density of such a normal distribution is given by

$$p(y_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right).$$

Since the data are assumed to be i.i.d., the density of their joint distribution is given by the product of their individual densities. (This, technically, is what it means for data to be independent.) That is,

$$p(y_1, \ldots, y_n) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right).$$

When this expression is viewed as a function of μ, it is known as the **likelihood function** of the data. A sensible way of estimating μ from the data is to pick the value for $\widehat{\mu}$ that maximises this likelihood function. The product makes this a bit cumbersome, but we can exploit the properties of logarithms to rewrite this product as a sum: You may remember from secondary school that $\log ab = \log a + \log b$. Since the logarithm is strictly monotonously increasing, the choice of $\widehat{\mu}$ that maximises the log-likelihood also maximises the likelihood. In sum, we choose $\widehat{\mu}$ as follows:

$$
\begin{aligned}
\widehat{\mu} &= \arg\max_{\mu} \left(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2} \right) \right) \\
&= \arg\max_{\mu} \left(\prod_{i=1}^{n} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2} \right) \right) && \text{[Scaling factor doesn't affect argmax.]} \\
&= \arg\max_{\mu} \log \left(\prod_{i=1}^{n} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2} \right) \right) && \text{[Take log.]} \\
&= \arg\max_{\mu} \sum_{i=1}^{n} \log \left(\exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2} \right) \right) && [\log ab = \log a + \log b] \\
&= \arg\max_{\mu} -\sum_{i=1}^{n} \frac{(y_i - \mu)^2}{2\sigma^2} && [\log(\exp x) = x] \\
&= \arg\max_{\mu} -\sum_{i=1}^{n} (y_i - \mu)^2 && [1/(2\sigma^2) \text{ is also a scaling factor.}] \\
&= \arg\min_{\mu} \sum_{i=1}^{n} (y_i - \mu)^2. && \text{[Maximising } -x \text{ is same as minimising } x.]
\end{aligned}
$$

We end up with exactly the same optimisation problem as when minimising the sum of squared residuals! So the solution is the same, too,

$$\widehat{\mu} = \frac{1}{n} \sum_{i=1}^{n} y_i.$$

In conclusion, when working with the general linear model, we obtain the same β estimates regardless of whether we're using the method of least squares (without making any assumptions about the data) or using maximum likelihood estimation under the assumption of i.i.d. normal data.[8] In the generalised linear model, the method of least squares isn't available or attractive, and parameter estimates are usually obtained using maximum likelihood estimation or some variation thereof.

Maximum likelihood theory provides methods for computing approximate standard errors

[8]Other assumptions about the distribution of the data will result in different maximum likelihood estimates. For instance, if we assumed that the data follow a Laplace distribution, then the maximum likelihood estimate for μ would be the sample median.

and for constructing approximate confidence intervals. We won't discuss these in this lecture series, though.

4 Assumptions and relevance, Episode 1

Model assumptions enter into the data analysis at some point, be it when estimating the model parameters (using maximum likelihood) or when estimating the uncertainty about these estimates. But how important are these assumptions about independence, equality of distributions and – when using t-distributions or maximum likelihood – normality?

The independence assumption is essential. If it is violated, the uncertainty about the parameter estimates may be massively underestimated using the methods covered in this lecture series. Deciding whether this assumption is justified typically requires knowledge about how the data were collected.

The assumption that the distributions are the same for different error terms isn't too relevant just yet, but we will come back to it in the next lectures.

The normality assumption is typically the least important of the three. In fact, in our running example, we *know* that this assumption is violated: Normal distributions theoretically range from $-\infty$ to $+\infty$, but our data are known to be constrained to the interval $[0, 204]$. Moreover, normal data are infinitely fine-grained, whereas our data consists of integers. That said, we observed that we obtained pretty much the same uncertainty measures when assuming normality as when using the semi-parametric bootstrap, which does not assume normality. This is quite common and can be attributed to the workings of the central limit theorem.

In my view, you shouldn't worry too much about the normality and equality of distributions assumptions. What you should ask yourself instead, however, is whether the model you want to fit and the statistics you want to compute are at all **relevant** in the context of your study and in the light of your data. Consider the data in Figure 1.10. It seems unlikely that the errors were drawn from a normal distribution. But rather than fit these data in a general linear model and then worry about the normality violation, you ought to ask yourself whether whatever research question you have can sensibly be answered in terms of the mean of these data. For this example, this seems unlikely, and a more pressing question presents itself: Where does the outlier come from? If, by contrast, you do think that the mean *is* a sensible statistic to compute in light of your research question and your data, then normality violations are unlikely to affect your results – but you can always verify those results using the semi-parametric bootstrap. Drawing plots of your data, both for yourself and for your readership, is essential in ensuring that your numeric analyses are relevant to the research questions at hand. Also see Vanhove (2021).

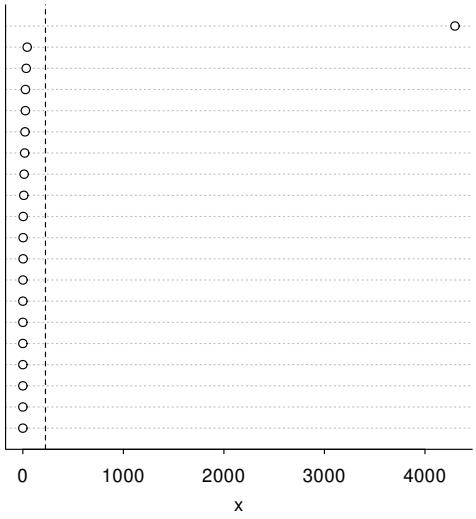

Figure 1.10: You probably *could* fit these data in a general linear model. But the mean (dashed vertical line) just isn't an informative measure of these data's central tendency. So why bother?

Summary

- We can think of our individual outcome data as being composed of some communality between all our outcome data and individual discrepancies.
- It is usually the communality that is of interest. It can be estimated using the discrepancies (residuals) and some optimisation criterion.
- Unless otherwise mentioned, the method of least squares is the optimisation criterion used. When working with a single outcome and no predictors, this yields the sample mean. But other optimisation criteria do exist.
- We can estimate the uncertainty in the parameter estimates using some version of the bootstrap or by making further assumptions.
- Rather than worry too much about normality, ask yourself whether what you want to compute is actually relevant in the light of your research questions and your data. If it is and you still have nagging doubts, use the semi-parametric bootstrap (or some similar procedure) to construct confidence bounds or to verify the results obtained using t-distributions.

Suggested reading

Hesterberg (2014) is a trove of accessibly presented information on the bootstrap.

References

DeKeyser, Robert, Iris Alfi-Shabtay & Dorit Ravid. 2010. Cross-linguistic evidence for the nature of age effects in second language acquisition. *Applied Psycholinguistics* 31. 413–438. doi:10.1017/S0142716410000056.

Efron, Bradley. 1979. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics* 7(1). 1–26. doi:10.1214/aos/1176344552.

Efron, Bradley & Robert J. Tibshirani. 1993. *An introduction to the bootstrap*. Boca Raton, FL: Chapman & Hall/CRC.

Ernst, Michael D. 2004. Permutation methods: A basis for exact inference. *Statistical Science* 19. 676–685.

Hesterberg, Tim C. 2014. What teachers should know about the bootstrap: Resampling in the undergraduate statistics curriculum. arXiv. doi:10.48550/arXiv.14115279.

Schwertman, Neil C., A. J. Gilks & J. Cameron. 1990. A simple noncalculus proof that the median minimizes the sum of the absolute deviations. *The American Statistician* 44(1). 38–39. doi:10.1080/00031305.1990.10475690.

Vanhove, Jan. 2021. Towards simpler and more transparent quantitative research reports. *ITL - International Journal of Applied Linguistics* 172(1). 3–25. doi:10.1075/itl.20010.van.

Zuur, Alain F., Elena N. Ieno, Neil J. Walker, Anatoly A. Saveliev & Graham M. Smith. 2009. *Mixed effects models and extensions in ecology with R*. New York, NY: Springer.