GLM – Lecture 1
Nuts and bolts

Jan Vanhove
https://janhove.github.io

Ghent, 12–14 July 2023

Goal and overview

This module is devoted to the workhorse of quantitative data analysis: the **general linear model**. 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. 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.
- 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 generate simulated 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

I used R version 4.3.0 when writing this script. 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 (File > New Project…). In your project directory, create a subdirectory named `data`. The datasets we'll be working with are available from the course module's website as well as from `https://github.com/janhove/GeneralLinearModel`. Put them in the `data` subdirectory. Next, create a subdirectory named `functions` and put the `scatterplot_matrix.R` file in it.

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 shortly. 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: Don't type along or work in R as you're attending this lecture. Focus on the logic; it's best to work through the R code at your own pace.

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

Figure 1 shows a basic dot plot of the participants' GJT scores.

```
> 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 you 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, we would have the following equation:

$$\text{a specific outcome} = \text{communality between all outcomes} + \text{some discrepancy}.$$

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 **residuals** or the **error term**. 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, the values of ε that we'll end up with will be estimates:

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

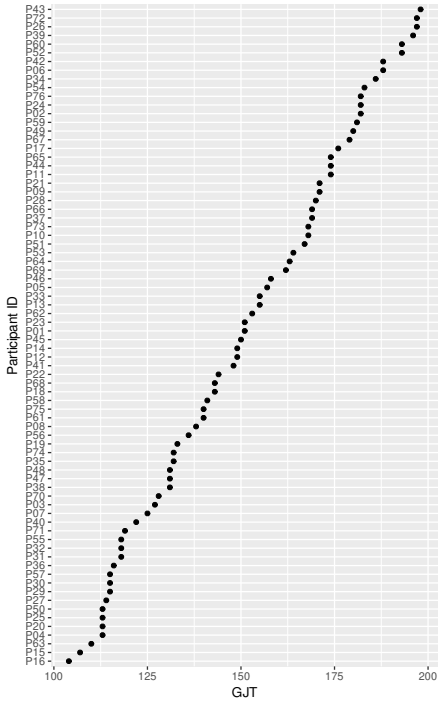

Figure 1: A dot plot of the participants' grammaticality judgement scores.

This is equivalent to

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

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., $\sum_{i=1}^{n} |\widehat{\varepsilon}_i|$) 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:

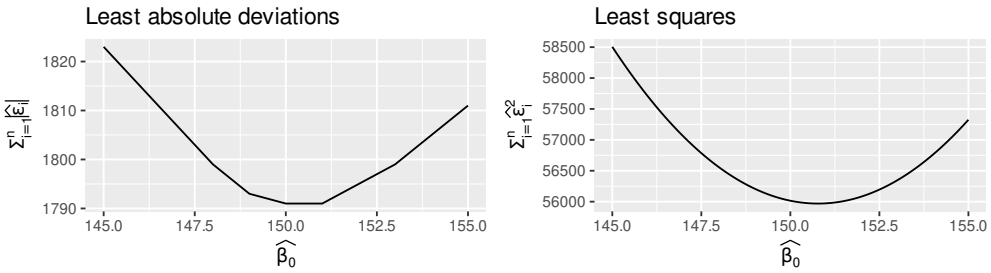

Figure 2: *Left:* If we estimate the parameter so as to minimise the sum of the absolute deviations, we obtain the sample median. *Right:* If we estimate the parameter so as to minimise the sum of the squared deviations, we obtain the sample mean.

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

[1] 1799
```

So by our 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 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 median of the GJT values is 150.5: if we estimate $\widehat{\beta}_0$ by minimising the sum of the absolute residuals, then we obtain the sample median (Schwertman et al., 1990)!

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., $\sum_{i=1}^{n} \widehat{\varepsilon}_i^2$) is as small as possible. This is the **method of least squares**. As shown in the right graph of Figure 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 estimate β_0 using least squares, we obtain the sample mean! You can prove this by finding the value for $\widehat{\beta}_0$ for which the derivative of the sum of squares, seen as a function of $\widehat{\beta}_0$, equals 0:

$$\frac{\mathrm{d}}{\mathrm{d}\widehat{\beta}_0} \sum_{i=1}^{n} (y_i - \widehat{\beta}_0)^2 \stackrel{!}{=} 0.$$

This only requires secondary school mathematics.

Computationally, the method of least squares is the easiest to work with. Unless otherwise mentioned, the parameter estimates in the general linear model are obtained using this method ('ordinary least squares', or OLS).

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)

       1        2        3        4        5        6
 0.22368 31.22368 -23.77632 -37.77632  6.22368 37.22368
       7        8        9       10       11       12
-25.77632 -12.77632  20.22368  17.22368  23.22368  -1.77632
      13       14       15       16       17       18
 4.22368  -1.77632 -43.77632 -46.77632  25.22368  -7.77632
      19       20       21       22       23       24
-17.77632 -37.77632  20.22368  -6.77632  0.22368  31.22368
      25       26       27       28       29       30
-37.77632  46.22368 -36.77632  19.22368 -35.77632 -35.77632
      31       32       33       34       35       36
-32.77632 -32.77632   4.22368  35.22368 -18.77632 -34.77632
      37       38       39       40       41       42
 18.22368 -19.77632  45.22368 -28.77632  -2.77632  37.22368
      43       44       45       46       47       48
 47.22368  23.22368  -0.77632   7.22368 -19.77632 -19.77632
      49       50       51       52       53       54
 29.22368 -37.77632  16.22368  42.22368  13.22368  32.22368
      55       56       57       58       59       60
-32.77632 -14.77632 -35.77632  -9.77632  30.22368  42.22368
      61       62       63       64       65       66
-10.77632   2.22368 -40.77632  12.22368  23.22368  18.22368
      67       68       69       70       71       72
 28.22368  -7.77632  11.22368 -22.77632 -31.77632  46.22368
      73       74       75       76
```

```
 17.22368 -18.77632 -10.77632  31.22368
```

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 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 a 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. By making some assumptions and leveraging the information in the data, we can, however, estimate the extent of this uncertainty.

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

The 'identical' part of this assumption is also referred to as the homoskedasticity assumption. It means that we assume that there exists a single distribution from which all residuals were drawn, that is,

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

for $i = 1, \ldots, n$. We don't make any further assumptions about this distribution yet; we're merely assuming that it is not the case that some residuals are drawn from, say, a uniform distribution spanning from -6 to 6 whereas some other residuals 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 residual, then this doesn't provide us with any more information about any of the remaining residuals.[1] 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 residual for a single production from a single speaker, chances are the other productions from this speaker will also have positive residuals. That is, once you've learnt something about the value of one specific residual, you can make more informed guesses about the values of some other residuals, too. Similarly, it is conceivable that [u] sounds tend to be longer in some words or phonological contexts than others. If so, the residual for a single word produced by a single speaker may

[1] The formal definition of stochastic independence is more involved, but essentially boils down to this.

give you some clues as to the residuals 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 using the method discussed above.

Problems with dependencies between residuals can be tackled in a number of ways, with one popular approach since 2008 or so being the use of mixed-effects models. These are beyond the scope of this module, but see Module 9.

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]

2.1 The semi-parametric bootstrap

We're currently assuming that the residuals be identically and independently distributed. To make some progress, we need to add a further assumption about the shape of the distribution from which the residuals were drawn. In the present subsection, we'll assume that the estimated residuals we have give us a reasonable approximation of the distribution of the residuals. Then, we can use the bootstrap technique (Efron, 1979; Efron & Tibshirani, 1993; Hesterberg, 2015) 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 goes 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.
3. Create a new vector of outcome values: $y_i^* = \widehat{\beta}_0 + \widehat{\varepsilon}_i^*$.
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.

In R:

```
> # Step 1 (model 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 + Step 3
+   d <- d |>
+     mutate(bs_outcome = Prediction + sample(Residual, replace = TRUE))
+   # Step 4
+   bs_mod <- lm(bs_outcome ~ 1, data = d)
+   bs_b0[[i]] <- coef(bs_mod)[[1]]
+ }
```

[2] We *could* go on and estimate the uncertainty about our uncertainty estimates, but no-one does so and let's not go there.

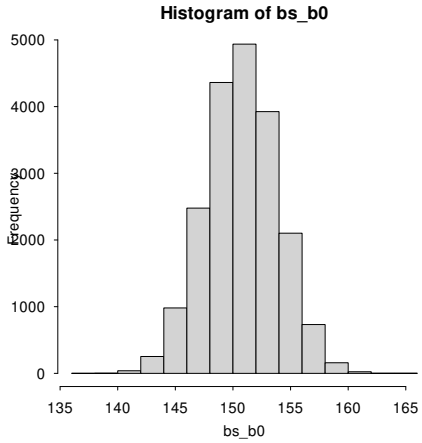

Figure 3: A quick histogram of the bootstrapped $\widehat{\beta}_0$ estimates.

A quick no-frills histogram of the $\widehat{\beta}_0^*$ values shows that these approximately form a normal distribution (Figure 3).

```
> hist(bs_b0)
```

At this point, let's go over some **basic facts of probability theory**. First, if n observations are drawn randomly from a distribution with mean μ, then the expected value of the mean of these n observations equals exactly μ. Put differently, if we draw lots of random samples from some distribution, then the mean of the sample means will equal the mean of the distribution from which the samples were drawn.

Second, if these n observations were not only drawn randomly but also independently from a distribution and this distribution has variance $\sigma^2 < \infty$, then the mean of the n observations will fall in a distribution with variance σ^2/n. The square root of this variance, i.e., $\sigma/\sqrt{n}$, is referred to as the **standard error** of the mean.

Third, if the observations are drawn randomly and independently from a distribution with finite variance and the sample size n tends to infinity, then the distribution of the sample means will converge to a normal distribution. As per the first and second point, this normal distribution has the same mean as the distribution from which the observations were drawn and variance σ^2/n. This fact is known as the **central limit theorem** (CLT).

What Figure 3 essentially shows us is an estimate of what the sampling distribution of the $\widehat{\beta}_0$ estimate looks like, subject to the assumptions we've made in the semi-parametric bootstrap. The fact that this sampling distribution looks approximately normal shows the central limit theorem at work. The standard deviation of the bootstrapped $\widehat{\beta}_0^*$ values provides us with an estimate of the **standard error**, i.e., the standard deviation of the sampling distribution of the parameter estimate. (Note that if you run the same commands, the outcome will be slightly different due to the randomness in step 2.)

```
> sd(bs_b0)

[1] 3.1178
```

We could also use the second fact discussed above to estimate the standard error, plugging in

the sample standard deviation for the unknown population standard deviation ($\widehat{SE} = s/\sqrt{n}$). Both approaches yield essentially the same outcome:

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

[1] 3.1336
```

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

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

  2.5%  97.5%
144.67 156.83
```

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:

```
> 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 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.

2.2 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 same mean, the mean of the residuals is always equal to 0.[3] We can write down our assumptions like so:

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

for $i = 1, 2, \ldots, n$. The standard deviation of residual distribution can be estimated from the

[3] The mean of the residuals is always equal to zero:

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

estimated residuals like so:

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

where p is the number of estimated β parameters (currently: $p = 1$). 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 $\widehat{\sigma}_\varepsilon$:

```
> sigma_mod.lm <- sigma(mod.lm)
>
> n_bootstrap <- 20000
> bs_b0 <- vector(length = n_bootstrap)
>
> for (i in 1:n_bootstrap) {
+   # Step 2 + Step 3
+   d <- d |>
+     mutate(bs_outcome = Prediction + rnorm(n = nrow(d), sd = sigma_mod.lm))
+   # 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] 3.1206

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

  2.5%  97.5%
144.64 156.93
```

While we assumed that the residual distribution was normal here, we can also apply the parametric bootstrap when making different assumptions about the residual distribution. We'd just have to plug in the corresponding function in step 2.

2.3 t-distributions

If we're happy to assume that the residuals are drawn from a normal distribution, we can compute the standard error estimates and confidence intervals directly, without the bootstrap. 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 estimated standard errors; the `t value` and `Pr(>|t|)` values correspond to the t- and p-values you know from Module 1.

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, the one-sample t-test is an instantiation of the general linear model.

2.4 Excursion: Maximum likelihood estimation

There's an alternative method for obtaining parameter estimates: maximum likelihood estimation (MLE). 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:

$$\begin{aligned}
\widehat{\mu} &= \arg\max_{\mu} \left(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right) \right) \\
&= \arg\max_{\mu} \left(\prod_{i=1}^{n} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right) \right) \\
&= \arg\max_{\mu} \log \left(\prod_{i=1}^{n} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right) \right) \\
&= \arg\max_{\mu} \sum_{i=1}^{n} \log \left(\exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right) \right) \\
&= \arg\max_{\mu} - \sum_{i=1}^{n} \frac{(y_i - \mu)^2}{2\sigma^2} \\
&= \arg\max_{\mu} - \sum_{i=1}^{n} (y_i - \mu)^2 \\
&= \arg\min_{\mu} \sum_{i=1}^{n} (y_i - \mu)^2.
\end{aligned}$$

(Line 1 to 2: The scaling factor doesn't matter, so drop it. Line 2 to 3: Take the logarithm. Line 3 to 4: $\log ab = \log a + \log b$. Line 4 to 5: $\log(\exp(x)) = x$. Line 5 to 6: The factor $1/(2\sigma^2)$ is just a scaling factor and doesn't matter. Line 6 to 7: Maximising $-x$ is the same as minimising x.)

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

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

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. For some generalisations of the general linear model (creatively called the *generalised* linear model), the method of least squares isn't available or attractive, and parameter estimates are usually obtained using some version of maximum likelihood estimation.

3 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, homoskedasticity 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 homoskedasticity assumption 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 homoskedasticity 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 light of your data. Consider the data in Figure 4. It seems unlikely that the residuals 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 – and 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.

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. This can be estimated using the discrepancies (residuals) and some optimisation criterion.

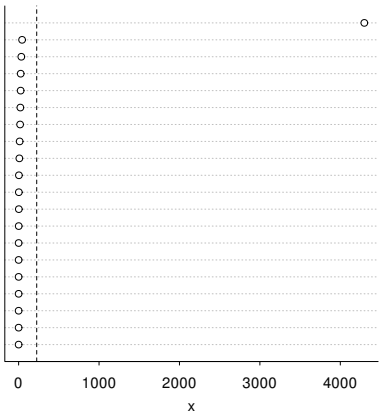

Figure 4: You probably *could* fit these data in a general linear model. But the mean (dashed vertical line) just isn't an informative measure of these data's central tendency. So why bother?

- 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 about normality, ask yourself whether what you want to compute is actually relevant in light of your research questions and your data.

Exercises

1. (Paper and pencil.) The distribution generated by throws of a fair six-sided dice has mean $\mu = 3.5$ and variance of $\sigma^2 = 2.91\overline{6}$. You throw the dice 9 times and you jot down the average number of pips observed. The throws can be assumed to be independent. What do you know about the sampling distribution of the average number of pips observed?
2. (With R.) Read in the DeKeyser et al. data as explained in Section 1, fit the model as we did above, and run the semi-parametric bootstrap using the code provided in Section 2.1; you can set the `n_bootstrap` value to 2000 instead of 20000 to speed up the computation. Compute 90% confidence intervals for the $\widehat{\beta}_0$ estimate using both the bootstrapped sampling distribution and the central limit theorem.

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.

Hesterberg, Tim C. 2015. What teachers should know about the bootstrap: Resampling in the undergraduate statistics curriculum. *The American Statistician* 69(4). 371–386. doi:10.1080/00031305.2015.1089789.

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.