

Introduction to mixed-effects models

Rob Davies (r.davies1@lancaster.ac.uk)

Contents

1	Introduction to linear mixed-effects models	2
1.1	Motivations: repeated measures designs and crossed random effects	2
1.2	The key idea to get us started	2
1.3	Targets	2
1.4	Study guide	2
1.5	The data we will work with: CP reading study	3
1.5.1	Our research question	3
1.5.2	The challenges of working with real (untidy) experimental data	3
1.5.3	Locate and download the data files	5
1.6	Tidy the data	6
1.6.1	Read in the data files by using the read_csv and read_tsv functions	6
1.6.2	Reshape the data from wide to long using the gather() function	8
1.6.3	Merging data from different data-sets using _join()	11
1.6.4	Select or transform the variables	13
1.6.5	Filter observations	14
1.6.6	Now we have some tidy data	16
1.6.7	We can output the data as a .csv file	16
1.6.8	Data tidying – conclusions	16
1.7	Repeated measures designs and crossed random effects	17
1.8	Working with mixed-effects models	17
1.8.1	Load the data if you need to	17
1.8.2	Linear model for multilevel data – ignoring the hierarchical structure	18
1.8.3	Can we ignore the hierarchical structure?	20
1.8.4	Multilevel – here, more appropriately known as – mixed-effects models	22
1.8.5	Is there a difference between linear model and linear mixed-effects model results? . . .	25
1.8.6	What we estimate when we estimate random effects	26
1.9	Variation between stimuli: the “language as fixed-effect fallacy”	29
1.9.1	Include the random effect of stimulus	30

1.10	Variances and covariances of random effects	32
1.10.1	But remember we excluded random effects covariance	33
1.11	Reporting the results of a mixed-effects model	33
1.12	Conclusions	34
1.12.1	Summary	34
1.12.2	Useful functions	35
1.13	R code and data file access for the class	35
1.14	References	36
1.14.1	Recommended reading	36
1.14.2	References list	36

1 Introduction to linear mixed-effects models

1.1 Motivations: repeated measures designs and crossed random effects

In the **Introduction to multilevel data**, we looked at a multilevel structured dataset in which there are observations about children's grades, and it is evident that those children can be grouped by or under classes. As we discussed, this kind of data structure will come from studies with a very common design in which the researcher recorded observations about a sample of children who are members of a sample of classes. In working with these kind of data, it is common to say that the observations of children's grades are *nested* within classes in a hierarchy.

Many Psychologists conduct studies where observations are properly understood to be structured in groups of some form but where, nevertheless, it is inappropriate to think of the observations as being nested (Baayen et al., 2008). We are talking, here, about **repeated-measures designs** where the experimenter presents a sample of multiple stimuli for response to each participant in a sample of for multiple participants. This is another *very* common experimental design in psychological science. Studies with this kind of design will produce data with a structure that, also, requires the use of mixed-effects models but, as we shall see, the way we think about the structure will be a bit more complicated. We could say that observations of the responses made by participants to each stimulus can be grouped by participant: each person will tend to respond in similar ways to different stimuli. Or, we could say that observations of responses can be grouped by stimulus because each stimulus will tend to evoke similar kinds of responses in different people. Or, we could say that both forms of grouping should be taken into account at the same time.

We shall take the third position and this chapter will concern why, and how we will adapt our thinking and practice.

1.2 The key idea to get us started

Linear mixed-effects models and multilevel models are basically the same.

This week, we again look at data with multilevel structure. But we are looking at data where participants were asked to respond to a set of stimuli (words) so that our observations consist of recordings made of the response made by each child to each stimulus. We use the same procedure we did for multilevel data but with one significant change which we shall identify and explain.

1.3 Targets

Our learning objectives again include the development of both understanding and practical skills.

skills Practice how to tidy experimental data for mixed-effects analysis

concepts Begin to develop an understanding of crossed random effects of subjects and stimuli

skills and concepts Practice fitting linear mixed-effects models incorporating random effects of subjects and stimuli

1.4 Study guide

1. Read in the example CP study data
2. Identify how the data are structured by both participant and stimulus differences
3. Use visualizations to explore the impact of the structure

4. Run analyses using linear mixed-effects models involving multiple random effects
5. Review readings provided

1.5 The data we will work with: CP reading study

This week, we will be working with the **CP reading study** dataset. CP tested 62 children (aged 116-151 months) on reading aloud in English. In the experimental reading task, she presented 160 words as stimuli. The same 160 words were presented to all children. The words were presented one at a time on a computer screen. Each time a word was shown, the children had to read the word out loud and their response was recorded. Thus, the CP reading study dataset comprised observations about the responses made by 62 children to 160 words.

In addition to the reading task, CP administered tests of reading skill (TOWRE sight word and phonemic tests, Torgesen et al., 1999), reading experience (CART, Stainthorp, 1997), the Spoonerisms sub-test of the Phonological Awareness test Battery (Frederickson et al., 1997), and an orthographic choice test measure of orthographic knowledge (based on Olson et al., 1985). She also recorded the gender and the handedness of the children.

Ultimately, the CP dataset were incorporated in an analysis of the impact of age on reading skills over the life-span, reported by Davies, Arnell, Birchenough, Grimmond and Houlson (2017). You can find more details on the data and the methods in that paper.

The CP study resembles many studies in psychological science. The critical features of the study are that we have an outcome measure – the reading response – observed multiple times (for each stimulus) for each participant. We have 160 responses recorded for each participant, one response for each stimulus word. And we have 62 responses recorded for each word, one response for each participant. The presence of these features is the reason why we need to use mixed-effects models in our analysis. These features are common across a range of study designs so the lessons we learn will apply frequently in psychological research. This is the reason why it is important we teach and learn how to use mixed-effects models.

1.5.1 Our research question

We are going to use these data to examine the answers to the following question:

RQ.1. What word properties influence responses to words in a test of reading aloud?

We can look at the answers to this question while also taking into account the impacts of random differences – between sampled participants or between sampled words – using mixed-effects models. But, first, we are going to look at how we get the data ready for analysis.

1.5.2 The challenges of working with real (untidy) experimental data

Ordinarily, textbooks and guides to data analysis give you the data ready for analysis but this situation will never be true for your professional practice (at least, not at first). Instead of pretending that data arrive ready for analysis, we are going to look at the process of **data tidying**, step-by-step. This will help you to get ready for the same process when you have to develop and use it in your own research.

We are going to spend a bit of time looking at the data tidying process. This process involves identifying and resolving a series of challenges, in order. Looking at the tidying process will give you a concrete sense of the structure in the data. You should also take this opportunity to reflect on the nature of the process itself – what we have to do and why, in what order and why – so that you can develop a sense of the process you might need to build when the time comes for you to prepare your own data for analysis.

The time that we spend looking at data tidying is an investment in learning that will save you time later, in your professional work. If, however, you want to skip it, go to section 1.7.

1.5.2.1 The data we need to use for analysis are not all in the same file In analyzing psychological data, the first step is usually to collect the data together. In psychological research, the data may exist, at first, in separate files. For the CP study, we have *separate files* for each of the pieces of information we need to use in our analyses:

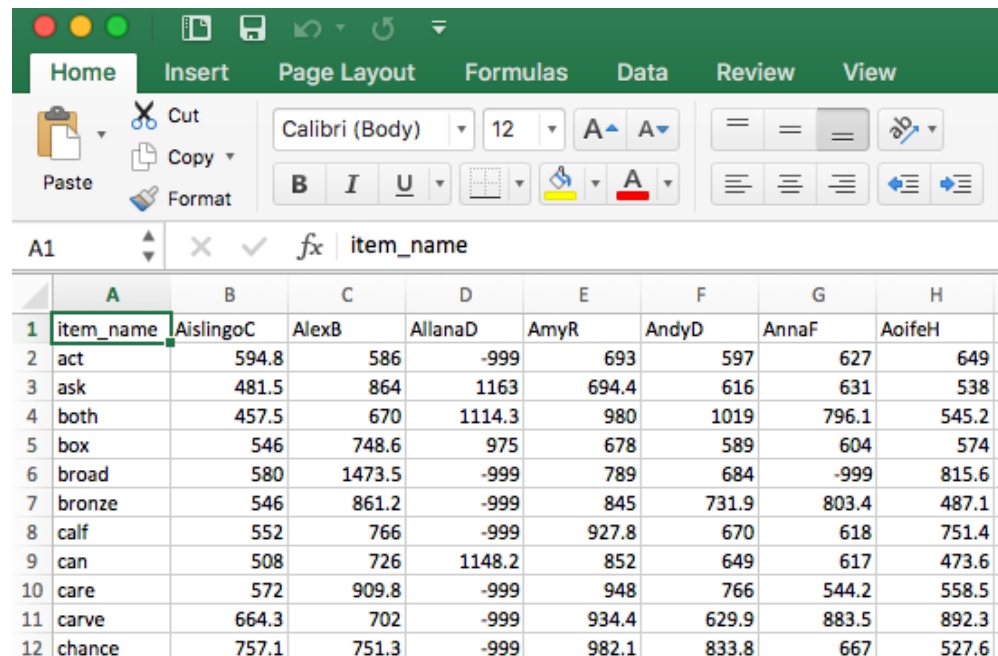
Participant attributes – information about participants’ age, gender, identifier code, and abilities on various measures.

Stimulus attributes – information about stimulus words, e.g., the word, its item number, its value on each variable in a set of psycholinguistic properties (like word length, frequency).

Behaviour – behavioural observations e.g. reaction time or accuracy of responses made by each participant to each stimulus word.

Often, we need all these kinds of information for our analyses but different pieces of information are produced in separate ways and come to us in separate files. For example, we may collect experimental response data using software like PsychoPy, E-Prime, Qualtrics or DMDX. We may collect information about participant characteristics using standardized measures, or by asking participants to complete a set of questions on their age, gender, and so on.

1.5.2.2 The data we need to use are untidy Often, the files we get are untidy: not in a useful or *tidy* format. For example, if you open the file CP_study_word_naming_rt_180211.dat (a .dat or tab delimited file) in Excel, you will see a spreadsheet that looks like Figure 1.



	A	B	C	D	E	F	G	H	
1	item_name	AislingoC	AlexB	AllanaD	AmyR	AndyD	AnnaF	AoifeH	C
2	act	594.8	586	-999	693	597	627	649	
3	ask	481.5	864	1163	694.4	616	631	538	
4	both	457.5	670	1114.3	980	1019	796.1	545.2	
5	box	546	748.6	975	678	589	604	574	
6	broad	580	1473.5	-999	789	684	-999	815.6	
7	bronze	546	861.2	-999	845	731.9	803.4	487.1	
8	calf	552	766	-999	927.8	670	618	751.4	
9	can	508	726	1148.2	852	649	617	473.6	
10	care	572	909.8	-999	948	766	544.2	558.5	
11	carve	664.3	702	-999	934.4	629.9	883.5	892.3	
12	chance	757.1	751.3	-999	982.1	833.8	667	527.6	

Figure 1: CP study RTs .dat file

Typical of the output from data collection software, we can see a data table with:

1. in the top row, column header labels `item_name`, `AislingoC`, `AllanaD` ...;
2. in the first (leftmost) column, row labels `item_name`, `act`, `ask`, `both` ...;
3. for each row, we see values equal to the reaction time (RT) observed for the response made to each stimulus (listed in the row labels);

4. for each column, we see values equal to the RTs observed for each person (listed in the column labels);
5. and at each intersection of row and column (for each cell), we see the RT observed for a response made by a participant to a stimulus.

Data laid out like this are sometimes said to be in *wide* format. You can see that the data are *wide* because at least one variable – here, reading reaction time – is held not in one column but spread out over several columns, side-by-side. Thus, the dataset is wide with fewer rows and many columns.

We want the data in what is called the *tidy* format.

1.5.2.3 How tidy data are tidy There are three inter-related rules which make data *tidy* (Grolemund & Wickham, 2019):

1. Each variable must have its own column.
2. Each observation must have its own row.
3. Each value must have its own cell.

You can read more about tidy data here:

<http://r4ds.had.co.nz/tidy-data.html>

For our purposes, the reason we want the data in *tidy* format is that it is required for the functions we are going to use for mixed-effects modelling. However, in general, *tidy* format is maximally flexible, and convenient, for use with different R functions.

1.5.3 Locate and download the data files

Go to the 402 Moodle folder for week 18, and download the .zip (compressed) folder labeled **PSYC402-01-multilevel-resources**

Or, download the same folder by clicking on the link:

<https://modules.lancaster.ac.uk/mod/resource/view.php?id=1795341>

In this folder, we have got four files that we will need to import or read in to R:

- CP study word naming rt 180211.dat
- CP study word naming acc 180211.dat
- words.items.5 120714 150916.csv
- all.subjects 110614-050316-290518.csv

The `words.items` file holds information about the 160 stimulus words presented in the experimental reading (word naming) task. The `all.subjects` file holds information about the 62 participants who volunteered to take part in the experiment. The `.csv` files are *comma separated values* files. The `.dat` files are *tab delimited* files holding behavioural data: the latency or reaction time `rt` (in milliseconds) and the accuracy `acc` of response made by each participant to each stimulus.

1.6 Tidy the data

To answer our research question, we will need to combine the behavioural data with information about the participants (age, gender ...) and about the words (word, frequency ...) We will need to ensure that the data-set we construct will be in *tidy* format. We will need to *select* variables (columns) to get just those required for our later analyses. And we will need to *filter* cases (rows), excluding errors or outliers.

We shall need to do this work in a series of processing steps:

1. Import the data or read the data into R, see Section 1.6.1
2. Restructure the data, see Section 1.6.2
3. Select or transform variables, see Section 1.6.4
4. Filter observations, see Section 1.6.5

We will use `tidyverse` library functions from the beginning, starting with the import stage.

```
library(tidyverse)
```

(Every step can also be done in alternative processing steps with the same result using *base R* code.)

1.6.1 Read in the data files by using the `read_csv` and `read_tsv` functions

I am going to assume you have downloaded the data files, that they are all in the same folder, and that you know where they are on your computer or server. We need to use different versions of the `read_` function to read all four files into R.

```
behaviour.rt <- read_tsv("CP study word naming rt 180211.dat", na = "-999")
behaviour.acc <- read_tsv("CP study word naming acc 180211.dat", na = "-999")
subjects <- read_csv("all.subjects 110614-050316-290518.csv", na = "-999")
words <- read_csv("words.items.5 120714 150916.csv", na = "-999")
```

These different versions respect the different ways in which the `.dat` and `.csv` file formats work. We need `read_tsv()` when data files consist of tab separated values. We need `read_csv()` when data files consist of comma separated values.

You can read more about the **tidyverse readr** library of helpful functions here:

<https://readr.tidyverse.org/>

It is *very* common to get experimental data in all sorts of different formats. Learning to use **tidyverse** functions will make it easier to cope with this when you do research.

1.6.1.1 Code tip Notice, here, that we use the `read_` function to read in the data, entering two arguments inside the brackets after the function name. For example, we write the code as:

```
behaviour.rt <- read_tsv("CP study word naming rt 180211.dat", na = "-999")
```

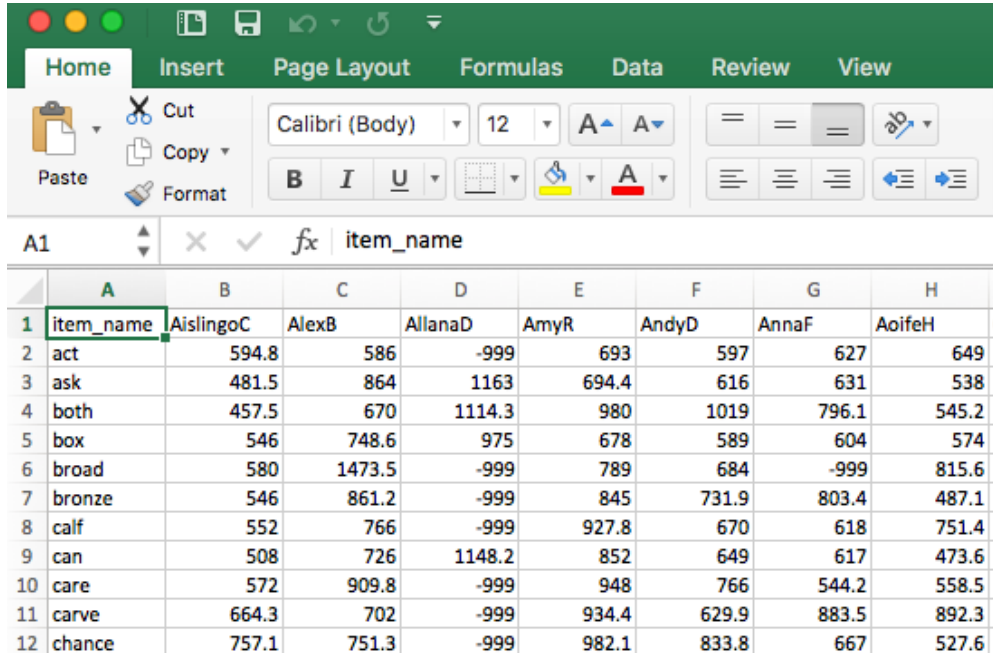
Take a look at what this line of code includes, element by element.

1. We write `behaviour.rt <- read_tsv(...)` to create an object in the R environment, which we call `behaviour.rt` – the object with this name is the dataset we read into R using `read_tsv(...)`.
2. When we write the function `read_tsv(...)` we include two arguments inside it.
3. `read_tsv("CP study word naming rt 180211.dat", ...` first, the name of the file, given in quotes `"` and then a comma.
4. `read_tsv(..., na = "-999")` second, we tell R that there are some missing values `na` which are coded with the value `"-999"`.

1.6.1.2 A quick lesson about missing value codes

In R, a missing value is said to be “not available”: NA.

In the datasets – typically, the spreadsheets – we create in our research, we will have values missing for different reasons. Take another look at the data spreadsheet you saw earlier, Figure 2.



	A	B	C	D	E	F	G	H
1	item_name	AislingC	AlexB	AllanaD	AmyR	AndyD	AnnaF	AoifeH
2	act	594.8	586	-999	693	597	627	649
3	ask	481.5	864	1163	694.4	616	631	538
4	both	457.5	670	1114.3	980	1019	796.1	545.2
5	box	546	748.6	975	678	589	604	574
6	broad	580	1473.5	-999	789	684	-999	815.6
7	bronze	546	861.2	-999	845	731.9	803.4	487.1
8	calf	552	766	-999	927.8	670	618	751.4
9	can	508	726	1148.2	852	649	617	473.6
10	care	572	909.8	-999	948	766	544.2	558.5
11	carve	664.3	702	-999	934.4	629.9	883.5	892.3
12	chance	757.1	751.3	-999	982.1	833.8	667	527.6

Figure 2: CP study RTs .dat file

You should be able to see that the spreadsheet holds information, as explained, about the RTs of the responses made by each child to each stimulus word. Each of the cells in the spreadsheet (i.e. the box where a column intersects with a row) includes a number value. Most of the values are positive numbers like 751.3: the reaction time of a response, recorded in milliseconds. The values have to be positive because they represent the length of time between the moment the stimulus word is presented on the test computer screen and the moment the child’s spoken word response has begun to be registered by the computer microphone and sound recording software.

Some of the cells hold the value -999, however. Obviously, we cannot have negative RT. The value represents the fact that we have no data. Take a look at Figure 2: we have a -999 where we should have a RT for the response made by participant AllanaD to the word broad. This -999 is there because, for some reason, we did not record an RT or a response for that combination of participant and stimulus.

We can choose any value we like, as researchers, to code for missing data like this. Some researchers choose not to code for the absence of a response recording or leave the cell in a spreadsheet blank or empty where data are missing. This is **bad practice** though it is common.

There are a number of reasons why it is bad practice to just leave a cell empty when it is empty because no observation is to be recorded.

1. Data may be missing for different reasons: maybe a child did not make any response to a stimulus (often called a “null response”); or maybe a child made a response but there was a microphone or other technical fault; or maybe a child made a response but it was an error and (here) the corresponding performance measure (RT) cannot be counted.
2. If you do not code for missingness in the data then the software you use will do it for you, but you may not know how it does so, or where.
3. If you have missing data, you ought to be able to identify where the data are missing.

I use -999 to code for missing values because you should never see a value like that in real reading RT data. You can use whatever value you like but you should make sure you *do* code for missing data somehow.

1.6.2 Reshape the data from wide to long using the `gather()` function

We are going to need to restructure these data from a wide format to a longer format. We need to restructure both behavioural data-sets, accuracy and RT. We do this using the `pivot_longer()` function.

```
rt.long <- behaviour.rt %>%
  pivot_longer(2:62, names_to = "subjectID", values_to = "RT")

acc.long <- behaviour.acc %>%
  pivot_longer(2:62, names_to = "subjectID", values_to = "accuracy")
```

Doing data-set construction programmatically, using R functions, is generally more reliable, and faster, than doing it by hand. Researchers used to have to do this sort of thing by hand, using copying and pasting, in Excel or SPSS. Doing the process by hand takes many hours or days. And you *always* make errors.

1.6.2.1 Code tip Here, we use a function you may not have seen before: `pivot_longer()`.

```
rt.long <- behaviour.rt %>%
  pivot_longer(2:62, names_to = "subjectID", values_to = "RT")
```

The name of the function comes from the fact that we are starting with data in wide format e.g. `behaviour.rt` where we have what should be a single variable of observations (RTs) arranged in a wide series of multiple columns, side-by-side (one column for each participant). But we want to take those wide data and *lengthen* the dataset, increasing the number of rows and decreasing the number of columns.

Let's look at this line of code bit by bit. It includes a powerful function that accomplishes a lot of tasks, so it is worth explaining this function in some detail.

1. `rt.long <- behaviour.rt %>%`

- At the start, I tell R that I am going to create a new longer dataset (more rows, fewer columns) that I shall call `rt.long`.
- I will create this longer dataset from `<-` the original wide dataset `behaviour.rt`.
- and I will create the new longer dataset by taking the original wide dataset and piping it `%>%` to the pivot function coded on the next line:

2. `pivot_longer(2:62, names_to = "subjectID", values_to = "RT")`

- On this next line, I tell R how to do the pivoting by using three arguments.

a. `pivot_longer(2:62...)`

- First, I tell R that I want to re-arrange all the columns that can be found in the dataset from the second column to the sixty-second column.
- In a spreadsheet, we have a number of columns.
- Columns can be identified by their position in the spreadsheet.
- The position of a column in a spreadsheet can be identified by number, from the leftmost column (column number 1) to the rightmost column (here, column number 62) in our dataset.

- So this argument tells R exactly which columns I want to pivot.

b. `pivot_longer(..., names_to = "subjectID", ...)`

- Second, I tell R that I want it to take the column labels and put them into a new column, called `subjectID`.
- In the wide dataset `behaviour.rt`, each column holds a list of numbers (RTs) but begins with a word in the topmost cell, the name code for a participant, in the column label position.
- We want to keep the information about which participant produces which response when we pivot the wide data to a longer structure.
- We do this by asking R to take the column labels (the participant names) and listing them in a new column, called `subjectID` which now holds the names as participant ID codes.

c. `pivot_longer(...values_to = "RT")`

- Third, we tell R that all the RT values should be put in a single column.
- We can understand that this new column RT will hold RT observations in a vertical stack, one cell for each response by a person to a word, with rows ordered by `subjectID`.

There are 61 columns of data listed by participant though 62 children were tested because we lost one child's data through an administrative error. As a result, in the wide data sets there are 62 columns, with the first column holding `item_name` data.

You can find more information about pivoting data here:

<https://tidyr.tidyverse.org/articles/pivot.html>

And you can find more information specifically about the `pivot_longer()` operation here:

<https://tidyr.tidyverse.org/articles/pivot.html>

1.6.2.1.1 Why we restructure the data As I noted, one problem with the wide format is that the data are structured so that the column names are not names of variables. In our example wide format dataset `behaviour.rt`, the columns are headed by a participant identity code or name but a participant code is not the name of a variable, it is a value of the variable I call `subjectID`. In the design of the CP reading study, we want to take into account the impact of differences between participants on response RT (so, we need to identify which participant makes which response). But we do not see the responses made by a participant as a predictor variable.

A second problem is that, in a wide format file like `behaviour.rt`, information about the responses made to each stimulus word is all on the same row (that seems good) but in different columns. Each person responded to all the words. But the response made to a word e.g. `act` made by one participant is in a different column (e.g., 594.8ms, for `AislingoC`) from the response made to the same word by a different participant (e.g., 586ms, for `AlexB`). This means that information about the responses made to each stimulus word are spread out as values across multiple columns.

You can see this for yourself if you inspect the source data using `head()`.

```
head(behaviour.rt)
```

item_name	AislingoC	AlexB	AllanaD	AmyR	AndyD	AnnaF	AoifeH	ChloeBergin	ChloeF	ChloeS	CianR
act	594.8	586.0	NA	693.0	597.0	627.0	649.0	1081.0	642.0	622.7	701.0
ask	481.5	864.0	1163.0	694.4	616.0	631.0	538.0	799.3	603.0	526.0	591.5
both	457.5	670.0	1114.3	980.0	1019.0	796.1	545.2	NA	581.0	568.4	665.0
box	546.0	748.6	975.0	678.0	589.0	604.0	574.0	658.0	688.7	492.0	641.0
broad	580.0	1473.5	NA	789.0	684.0	NA	815.6	NA	NA	798.0	1473.6
bronze	546.0	861.2	NA	845.0	731.9	803.4	487.1	1701.0	871.0	574.0	753.0

```
head(behaviour.acc)
```

item_name	AislingoC	AlexB	AllanaD	AmyR	AndyD	AnnaF	AoifeH	ChloeBergin	ChloeF	ChloeS	CianR
have	1	1	1	1	1	1	1	1	1	1	1
cheer	1	1	0	1	1	1	1	1	1	1	1
ask	1	1	1	1	1	1	1	1	1	1	1
care	1	1	0	1	1	1	1	1	1	1	1
with	1	1	1	1	1	1	1	1	1	1	1
false	1	1	0	1	1	1	1	1	1	1	1

This structure is a problem for visualization and for analysis because the functions we will use require us to specify *single* columns for an outcome variable like reaction time.

We are looking at the process of tidying data because untidiness is very common. Learning how to deal with it will save you a lot of time and grief later.

You should check for yourself how **subjectID** and **RT** or **accuracy** scores get transposed from the old structure to the new structure.

```
head(rt.long)
```

item_name	subjectID	RT
act	AislingoC	594.8
act	AlexB	586.0
act	AllanaD	NA
act	AmyR	693.0
act	AndyD	597.0
act	AnnaF	627.0

```
head(acc.long)
```

item_name	subjectID	accuracy
have	AislingoC	1
have	AlexB	1
have	AllanaD	1
have	AmyR	1
have	AndyD	1
have	AnnaF	1

If you compare the **rt.long** or **acc.long** data with what you see in when the data are in the original wide format then you can see how – in going from wide – we have re-arranged the data to a longer and narrower set of columns, one column listing each word, one column for **subjectID** and one column for **RT** or **accuracy**. What a check will show you is that we have multiple rows for responses to each item so that the item is repeated multiple times in different rows.

These data are *now tidy*.

- Each column has information about one variable
- And each row has information about one observation, here, the response made by a participant to a word

But these data are *incomplete*. Next we shall combine behavioural observations with data about stimulus words and about participants.

1.6.2.1.2 The tidyverse evolves Over the years, different ways of reshaping data have evolved. This reflects how important and common the task is. An older way to do the same operation uses the function `gather()`.

You can read more about `gather()` here:

<http://r4ds.had.co.nz/tidy-data.html#spreading-and-gathering>

In **tidyverse** the functions designed to enable you to restructure data have evolved through a series of different forms. This change is one of the real benefits of using open software like R. In my experience, the newer functions can be useful for *really* untidy data. I expect things will continue to evolve and improve over time.

1.6.3 Merging data from different data-sets using `__join()`

To answer our research question, we next need to combine the **RT** with the **accuracy** data, and then the combined behavioural data with **participant** information and **stimulus** information. This is because, as we have seen, information about behavioural responses, about participant attributes or stimulus word properties, are located in separate files.

Many researchers have completed this kind of operation by hand. This involves copying and pasting bits of data in a spreadsheet. It can take hours or days. I know because I have done it, and I have seen others do it. **Please don't.** There are better ways to spend your time. And you will make mistakes that you will not then be able to identify.

We can combine the datasets, in the way that we need, using the **tidyverse** `full_join()` function. This gets the job done quickly, and accurately.

First, we join RT and accuracy data together.

```
long <- rt.long %>%  
  full_join(acc.long)
```

Then, we join subject and item information to the behavioural data.

```
long.subjects <- long %>%  
  full_join(subjects, by = "subjectID")  
  
long.all <- long.subjects %>%  
  full_join(words, by = "item_name")
```

Notice, we can let R figure out how to join the pieces of data together. If we were doing this by hand then we would need to check *very carefully* the correspondences between observations in different datasets.

1.6.3.1 Code tip Here, in a series of steps, we take one dataset and join it (merge it) with the second dataset. Let's look at an example.

```
long <- rt.long %>%  
  full_join(acc.long)
```

The code work as follows.

```
1. long <- rt.long %>%
```

- We create a new dataset we call `long`.
- We do this by taking one original dataset `rt.long` and `%>%` piping it to the operation defined in the second step.

2. `full_join(acc.long)`

- In this second step, we use the function `full_join()` to add observations from a second original dataset `acc.long` to those already from `rt.long`

The addition of observations from one database joining to those from another happens through a matching process.

- R looks at the datasets being merged.
- It identifies if the two datasets have columns in common. Here, the datasets have `subjectID` and `item_name` in common).
- R can use these common columns to identify rows of data. Here, each row of data will be identified by both `subjectID` and `item_name` i.e. as data about the response made by a participant to a word.
- R will then do a series of identity checks, comparing one dataset with the other and, row by row, looking for matching values in the common columns.
- If there is a match then R joins the corresponding rows of data together.
- If there isn't a match then it creates `NAs` where there are missing entries in one row for one dataset which cannot be matched to a row from the joining dataset.

Note that in one example, the example of code I discuss here, I did not specify identifying columns in common, allowing the function to do the work. In the other code chunks I did: `long.all <- long.subjects %>% full_join(words, by = "item_name")` using the `by = ...` argument.

1.6.3.2 Relational data In the `tidyverse` family of `dplyr` functions, when you work with multiple datasets (tables of data), we call the datasets **relational** data.

<http://r4ds.had.co.nz/relational-data.html#relational-data>

There are three families of verbs designed to work with relational data:

- Mutating joins, which add new variables to one data frame from matching observations in another.
- Filtering joins, which filter observations from one data frame based on whether or not they match an observation in the other table.
- Set operations, which treat observations as if they were set elements.

We can connect datasets – relate them – according to shared variables like `subjectID`, `item_name` (for our data). In `tidyverse`, the variables that connect pairs of tables are called keys where, and this is what counts, *key(-s) are variable(-s) that uniquely identify an observation*.

For the experimental reading data, we have observations about each response made by a participant (one of 61 subjects) to an item (one of 160 words). For these data, we can match up a pair of RT and accuracy observations for each (unique) `subjectID-item_name` combination.

If you reflect, we could not combine the RT and accuracy data correctly:

1. If we did not have both identifying variables for both datasets, both required to uniquely identify each observation.

2. If there were mismatches in values of the identifying variable.

Sometimes, I have done this operation and it has gone wrong because a `subjectID` has been spelled one way in one dataset e.g. `hugh` and another way in the other dataset e.g. `HughH`. This means I am careful about spelling identifiers and I always check my work after merger operations, calculating dataset lengths to ensure the number of rows in the new dataset matches my expectations.

1.6.3.3 `__join` functions

We used the `full_join()` function.

There are three kinds of joins.

- A left join keeps all observations in `x`.
- A right join keeps all observations in `y`.
- A full join keeps all observations in `x` and `y`.

I used `full_join()` because I wanted to retain all observations from both datasets, whether there was a match (as assumed) or not, in the identifying variables, between observations in each dataset.

1.6.3.4 Exercise

Break the join You could examine how the `full_join()` works by experimenting with stopping it from working

As I discuss, you need to have matches in values on key (common) variables. If the `subjectID` is different on different datasets, you will lose data that would otherwise be merged to form the merged or composite dataset. So, check what happens if you deliberately mis-spell one of the `subjectID` values in one of the original source wide behavioural data files.

To be safe, you might want to do this exercise with copies of the source files kept in a folder you create for this purpose. If it goes wrong, you can always re-access the source files and read them in again.

You can check what happens before and after you break the match by counting the number of rows in the dataset that results from the merger. We can count the number of rows in a dataset with:

```
length(long.all$RT)
```

```
## [1] 9762
```

This bit of code takes the length of the vector (i.e. variable column `RT` in dataset `long.all`), thus counting the number of rows in the dataset.

1.6.4 Select or transform the variables

OK, now we have all the data about everything all in one big, long and wide, dataset. But we do not actually need all this stuff. We next need to do two things. First, we need to get rid of variables we will not use: we do that by using `select()`. Then, we need to remove errors and outlying short `RT` observations: we do that by using `filter()` in Section 1.6.5.

We are going to select just the variables we need using the `select()` function.

```
long.all.select <- long.all %>%
  select(item_name, subjectID, RT, accuracy,
         Lg.UK.CDcount, brookesIMG, AoA_Kup_lem,
         Ortho_N, regularity, Length, BG_Mean,
         Voice, Nasal, Fricative, Liquid_SV,
         Bilabials, Labiodentals, Alveolars,
         Palatals, Velars, Glottals, age.months,
         TOWREW_skill, TOWRENW_skill, spoonerisms, CART_score)
```

Notice that these variables do not have *reader-friendly* names. Naming things well is important, as Jenny Bryan teaches:

<https://speakerdeck.com/jennybc/how-to-name-files>

I would say that this true for variables as much as for files. The names we have in the CP study data were fine for internal use within my research group but we should be careful to ensure that variables have names that make sense to others and to our future selves. We can adjust variable names using the `rename()` function but I will leave that as an exercise for you to do.

1.6.4.1 Exercise

Select different variables You could analyze the CP study data for a research report. What if you wanted to analyze a different set of variables, could you select different variables?

1.6.5 Filter observations

We now have a tidy dataset `long.all.select` with 26 columns and 9762 rows.

The dataset includes missing values, designated NA for *not available* (to you). Here, every error (coded 0, in `accuracy`) corresponds to an NA in the RT column.

The dataset also includes outlier data. In this context, $RT < 200$ are probably response errors or equipment failures. We will want to analyse `accuracy` later, so we shall need to be careful about getting rid of NAs.

At this point, I am going to exclude two sets of observations only.

- observations corresponding to correct response reaction times that are too short: $RT < 200$.
- plus observations corresponding to the word *false* which (because of stupid Excel auto-formatting) dropped item attribute data.

We can do this using the `filter()` function, setting conditions on rows, as arguments.

```
# step 1
long.all.select.filter <- long.all.select %>%
  filter(item_name != 'FALSE')

# step 2
long.all.select.filter <- long.all.select.filter %>%
  filter(RT >= 200)
```

1.6.5.1 Code tip Here, I am using the function `filter()` to ...

- Create a new dataset `long.all.select.filter <- ...` by
- Using functions to work on the data named immediately to the right of the assignment arrow: `long.all.select`
 - An observation is included in the new dataset if it matches the condition specified as an argument in the `filter()` function call, thus:
 1. `filter(item_name != 'FALSE')` means: include in the new dataset `long.all.select.filter` all observations from the old dataset `long.all.select` that are not `!=` (! not = equal to) the value `FALSE` in the variable `item_name`
 2. then recreate the `long.all.select.filter` as a version of itself (with no name change) by including in the new version only those observations where RT was greater than or equal to 200ms using `RT >= 200`

1.6.5.2 The difference between = and == You need to be careful to distinguish these signs.

- `=` assigns a value, so `x = 2` means "x equals 2"
- `==` tests a match so `x == 2` means: "is x equal to 2?"

1.6.5.3 Using multiple arguments in filtering You can supply multiple arguments to `filter()` and this may be helpful if (1.) you want to filter observations according to a match on condition-A **and** condition-B (logical “and” is coded with `&`) or (2.) you want to filter observations according to a match on condition-A or condition-B (logical “or” is coded `|`).

You can read more about using multiple arguments to filter observations here:

<https://dplyr.tidyverse.org/reference/filter.html>

1.6.5.4 Exercise

Vary the filter conditions in different ways

1. Change the threshold for including RTs from `RT >= 200` to something else
2. Can you assess what impact the change has? Note that you can count the number of observations (rows) in a dataset using e.g. `length(data.set.name$variable.name)`

Filtering or re-coding observations is an important element of the research workflow in psychological science. How we do or do not remove observations from original data may have an impact on our results (as explored by, e.g., Steegen et al., 2014). It is important, therefore, that we learn how to do this reproducibly using R scripts that we can share with our research reports.

You can read further information about filtering here:

<https://r4ds.had.co.nz/transform.html?q=filter#filter-rows-with-filter>

1.6.5.5 Remove missing values We will be working with the `long.all.select.filter.csv` dataset collated from the experimental, subject ability scores, and item property data collected for the CP word naming study.

For convenience, I am going to remove missing values before we go any further, using the `na.omit()` function.


```
long.all.noNAs <- na.omit(long.all.select.filter)
```

1.6.5.6 Code tip The `na.omit()` function is powerful. In using this function, I am asking R to create a new dataset `long.all.noNAs` from the old dataset `long.all.select.filter` in a process in which the new dataset will have *no* rows in which there is a missing value `NA` in *any* column. You need to be reasonably sure, when you use this function, where your `NAs` may be because, otherwise, you may end the process with a new filtered dataset that has many fewer rows in it than you expected.

1.6.6 Now we have some tidy data

```
head(long.all.noNAs, n = 10)
```

item_name	subjectID	RT	accuracy	Lg.UK.CDcount	brookesIMG	AoA_Kup_lem	Ortho_N	regularity	Le
act	AislingC	594.8	1	4.034067	4	6.42	5	1	
act	AlexB	586.0	1	4.034067	4	6.42	5	1	
act	AmyR	693.0	1	4.034067	4	6.42	5	1	
act	AndyD	597.0	1	4.034067	4	6.42	5	1	
act	AnnaF	627.0	1	4.034067	4	6.42	5	1	
act	AoifeH	649.0	1	4.034067	4	6.42	5	1	
act	ChloeBergin	1081.0	1	4.034067	4	6.42	5	1	
act	ChloeF	642.0	1	4.034067	4	6.42	5	1	
act	ChloeS	622.7	1	4.034067	4	6.42	5	1	
act	CianR	701.0	1	4.034067	4	6.42	5	1	

If we inspect the `long.all.noNAs` data-set, we can see that we have now got a tidy data-set with all the data we need for our analyses:

- One observation per row, corresponding to data about a response made by a participant to a stimulus in an experimental trial
- One variable per column
- We have information about the speed and accuracy of responses
- And we have information about the children and about the words.

We have removed the missing values and we have filtered outliers.

1.6.7 We can output the data as a .csv file

Having produced the tidy dataset, we may wish to share it, or save ourselves the trouble of going through the process again. We can do this by creating a .csv file.

```
write_csv(long.all.noNAs, "long.all.noNAs.csv")
```

This function will create a .csv file from the dataset you name `long.all.noNAs` which R will put in your working directory.

1.6.8 Data tidying – conclusions

Most research work involving quantitative evidence requires a *big* chunk of data tidying or other processing before you get to the statistics. Most of the time, this is work *you* will have to do. The lessons you can learn about the process will generalize to many future research scenarios.

1.7 Repeated measures designs and crossed random effects

Our focus this week is on analyzing data that come from studies with **repeated-measures designs** where the experimenter presents multiple stimuli for response to each participant. In our working example, the **CP reading study**, CP asked all participants in her study to read a selection of words. All participants read the same selection of words, and every person read every word. For each participant, we have multiple observations and these (within-participant) observations will not be independent of each other. One participant will tend to be slower or less accurate compared to another participant, on average. Likewise, one participant's responses will reveal a stronger (or weaker) impact of the effect of an experimental variable than another participant. These between-participant differences will tend to be apparent for each set of observations we have for each participant, across the sample of participants.

You could say that the lowest trial-level observations can be grouped with respect to participants, that observations are nested within participant. But the data can also be grouped by stimuli. Remember that in the CP study, all participants read the same selection of words, and every person read every word. This means that for each stimulus word, there are multiple observations because all participants responded to each word, and these (within-item) observations will not be independent of each other. One word may prove to be more challenging compared to another, eliciting slower or less accurate responses, on average. Likewise, participants' responses to a word will reveal a stronger (or weaker) impact of the effect of an experimental variable than the responses to another word. Again, these between-stimulus differences will tend to be apparent for each set of observations we have for each stimulus word, across the sample of words.

Under these circumstances, are observations about the responses made by different participants nested under words or are observations about the responses to different words nested under participants? We do not have to make a decision.

Given this common **repeated-measures** design, we can analyze the outcome variable in relation to:

fixed effects the impact of independent variables like participant reading skill or word frequency

random effects the impact of random or unexplained differences between participants and also between stimuli

In this situation, we can say that the random effects are crossed (Baayen et al., 2008). When multilevel models require the specification of crossed random effects, they tend to be called **mixed-effects models**.

1.8 Working with mixed-effects models

To illustrate the approach, we examine observations from the CP study. We begin, as we did previously, by ignoring differences due to grouping variables (like participant or stimulus). We pretend that all observations are independent. In this fantasy situation, we address our research question.

RQ.1. What word properties influence responses to words in a test of reading aloud?

1.8.1 Load the data if you need to

If you did not go through the process of tidying the CP study data from the component source data files, then you can import the pre-tidied data here.

```
long.all.noNAs <- read_csv("long.all.noNAs.csv",
  col_types = cols(
    subjectID = col_factor(),
    item_name = col_factor()
  )
)
```

1.8.1.1 Code tip Notice that I am using `read_csv()` with an additional argument `col_types = cols(...)` through which I control how `read_csv()` processes specific column variables in the data. Here, I am requesting that `read_csv()` treats `subjectID` and `item_name` as factors.

This is a very useful capacity, and a more efficient way to work than, say, first reading in the data and then using *coercion* to ensure that variables are assigned appropriate types. You can read more about it here.

<https://readr.tidyverse.org/articles/readr.html>

1.8.2 Linear model for multilevel data – ignoring the hierarchical structure

We begin by asking if reading reaction time (RT) varies in association with word frequency. A scatterplot shows that response latencies decrease with increasing word frequency (Figure 3).

```
long.all.noNAs %>%
  ggplot(aes(x = Lg.UK.CDcount, y = RT)) +
    geom_point(alpha = .2) +
    geom_smooth(method = "lm", se = FALSE, size = 1.5, colour="red") +
    theme_bw() +
    xlab("Word frequency: log context distinctiveness (CD) count")
```

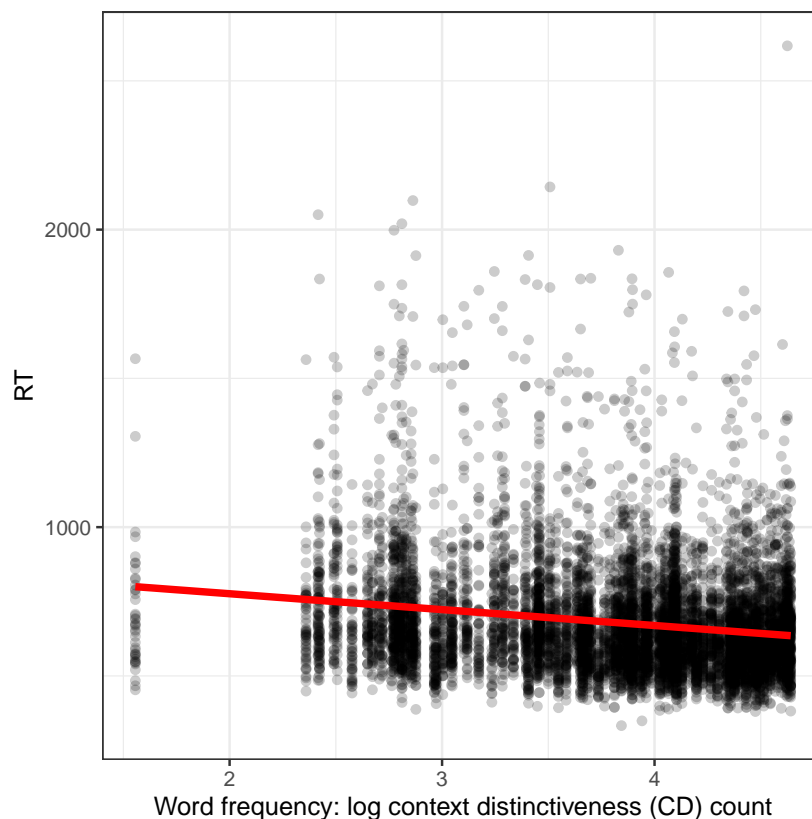


Figure 3: Reading reaction time compared to word frequency, all data

In the plot, we see that the best fit line drawn with `geom_smooth()` trends downward for higher values of word frequency. This means that Figure 3 suggests that RT decreases with increasing word frequency. (I know there is a weird looking line of points around 0 but we can ignore that here.)

We can estimate the relationship between RT and word frequency using a linear model in which we ignore the possibility that there may be differences (between subjects, or between items) in the intercept or (between subjects) in the slope of the frequency effect:

$$Y_{ij} = \beta_0 + \beta_1 X_j + e_{ij} \quad (1)$$

- where Y_{ij} is the value of the observed outcome variable, the RT of the response made by the i participant to the j word;
- $\beta_1 X_j$ refers to the fixed effect of the explanatory variable (here, word frequency), where the frequency value X_j is different for different words j , and β_1 is the estimated coefficient of the effect due to the relationship between response RT and word frequency;
- e_{ij} is the residual error term, representing the differences between observed Y_{ij} and predicted values (given the model).

The linear model can be fit in R using the `lm()` function, as we have done previously.

```
lm.all.1 <- lm(RT ~ Lg.UK.CDcount,
               data = long.all.noNAs)

summary(lm.all.1)

##
## Call:
## lm(formula = RT ~ Lg.UK.CDcount, data = long.all.noNAs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -346.62 -116.03  -38.37   62.05 1981.58
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    882.983     11.901   74.19  <2e-16 ***
## Lg.UK.CDcount  -53.375       3.067  -17.40  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 185.9 on 9083 degrees of freedom
## Multiple R-squared:  0.03227,    Adjusted R-squared:  0.03216
## F-statistic: 302.8 on 1 and 9083 DF,  p-value: < 2.2e-16
```

We can see that, in this first analysis, the estimated effect of word frequency is $\beta = -53.375$ (here, word frequency information is in the `Lg.UK.CDcount` variable). This means that, in the linear model, RT decreases by about 54 milliseconds for each unit increase in log word frequency. (In our analyses, in common with many in the reading literature, we transformed the frequency estimate to the Log base 10 of the frequency of occurrence estimated for each word.) The model does not explain much variance, as $R^2 = .03$ but, no doubt due to the large sample, the regression model is overall significant $F(1, 9083) = 302.8, p < .001$.

1.8.2.1 Exercise

Vary the linear model using different outcomes or predictors

The CP study dataset is rich with possibility. It would be useful to experiment with it.

1. Change the predictor from frequency to something else: what do you see when you visualize the relationship between variables using scatterplots?
2. Specify linear models with different predictors: do the relationships you see in plots match the coefficients you see in the model estimates?

1.8.3 Can we ignore the hierarchical structure?

In this linear model, the observations are assumed to be independent but the assumption of independence is questionable given the expectation that participants will differ, with one participant's responses perhaps slower or less accurate than another, perhaps more or less affected by word frequency than another. We can examine that variation by estimating the intercept and the slope of the frequency effect separately using the data for each participant alone.

We can start by examining the frequency effect for each child in a grid of plots, with each plot representing the $RT \sim frequency$ relationship for the data for a child (Figure 4).

We discussed how the plotting code functions in the previous chapter.

```
long.all.noNAs %>%
  ggplot(aes(x = Lg.UK.CDcount, y = RT)) +
    geom_point(alpha = .2) +
    geom_smooth(method = "lm", se = FALSE, size = 1.25, colour = "red") +
    theme_bw() +
    xlab("Word frequency (log10 UK SUBTLEX CD count)") +
    facet_wrap(~ subjectID)
```

Figure 4 shows how, on average, more frequent words are associated with shorter reaction time, faster responses. The plot further shows, however, that the effect of frequency varies considerably between children. Some children show little or no effect; the best fit line is practically level. Other children show a marked effect, with a steep fit line indicating a strong frequency effect.

We can get more insight into the differences between children, however, if we plot the estimated intercept and frequency effect coefficients for each child directly. This allows more insight because it focuses the eye on the differences between children in the estimates.

Figure 5 presents a plot showing the estimates of the intercept and the coefficient of the effect of word frequency on reading RT, calculated separately for each child. The estimate for each child is shown as a black dot. The standard error of the estimate is shown as a black vertical line, shown above and below a point. You can say that where there is a longer line there we have more uncertainty about the location of the estimate.

Figure 5 presents the estimates of intercept and the frequency coefficient, calculated for each child, ordered by the size of the estimate. Drawn this way, we can see how the estimates of both the intercept and the slope of the frequency effect vary substantially between children. We can see also how the standard errors vary greatly between children.

Notice that if there is an average intercept for everyone in the sample or, better, an intercept we could estimate for everyone in the population, then the different intercepts we have estimated for each child would be distributed around that population-level average. Some children will have slower (here, larger) intercepts and other children will have faster (shorter) intercepts. (Here, the intercept can be taken to be the average RT when all other effects in the model are set to zero. RT varies for this sample around somewhere like $\beta_0 = 883ms$ so a slower larger intercept might be e.g. $\beta_0 = 1000ms$.)

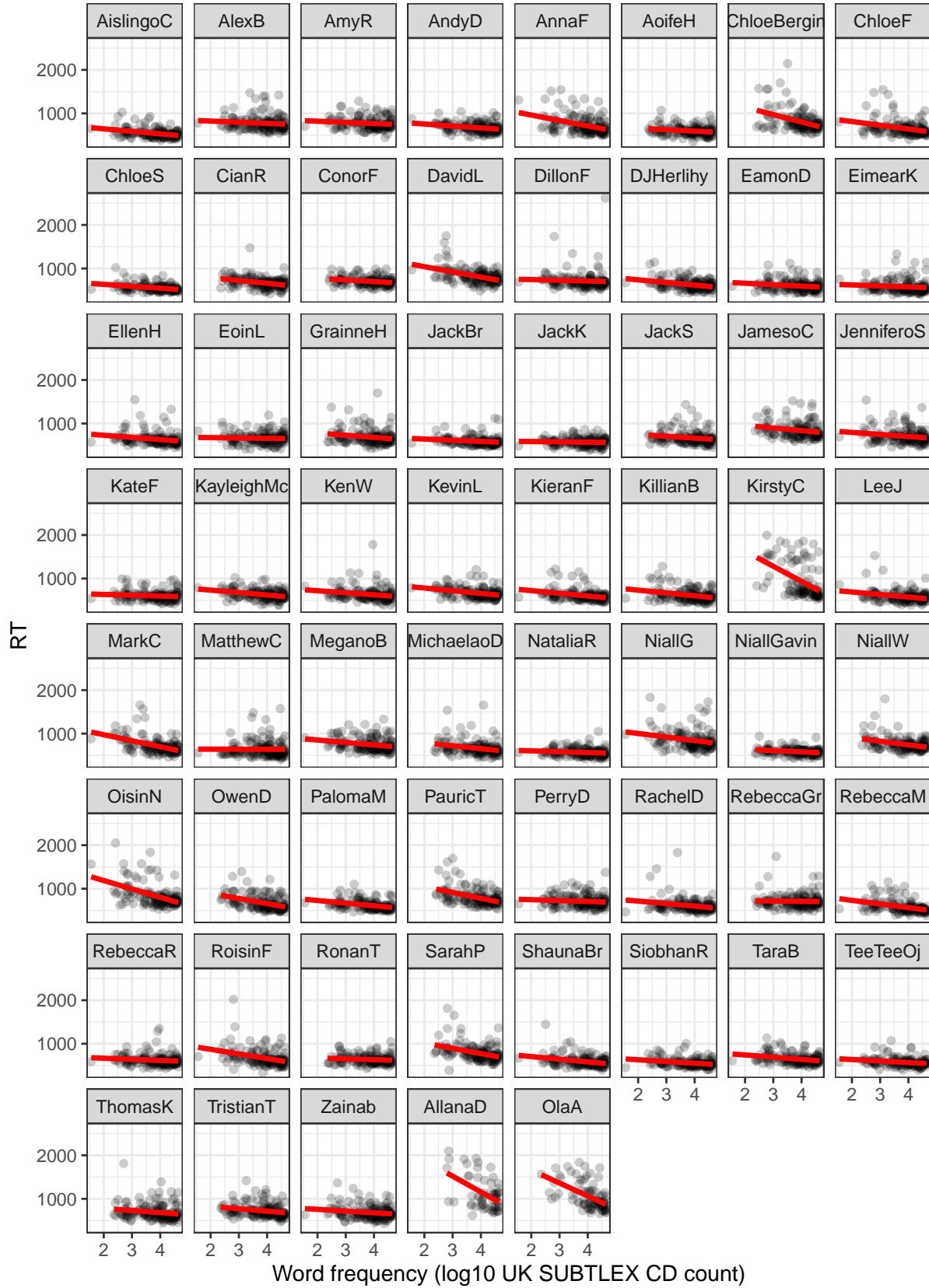


Figure 4: RT vs. word frequency, considered separately for data for each child

In the same way, if there is an average slope for the frequency effect, an effect of frequency on reading RT, averaged across everyone in the population, then, again, the different slopes we have estimated for each child would be distributed around that population-level effect. Some children will have larger (here, more negative) frequency effects and other children will have smaller (less negative) frequency effects. (Here, the frequency effect is associated with a negative coefficient e.g. $\beta_1 = -53$ so a larger frequency effect will be a bigger negative number e.g. $\beta_1 = -100$.)

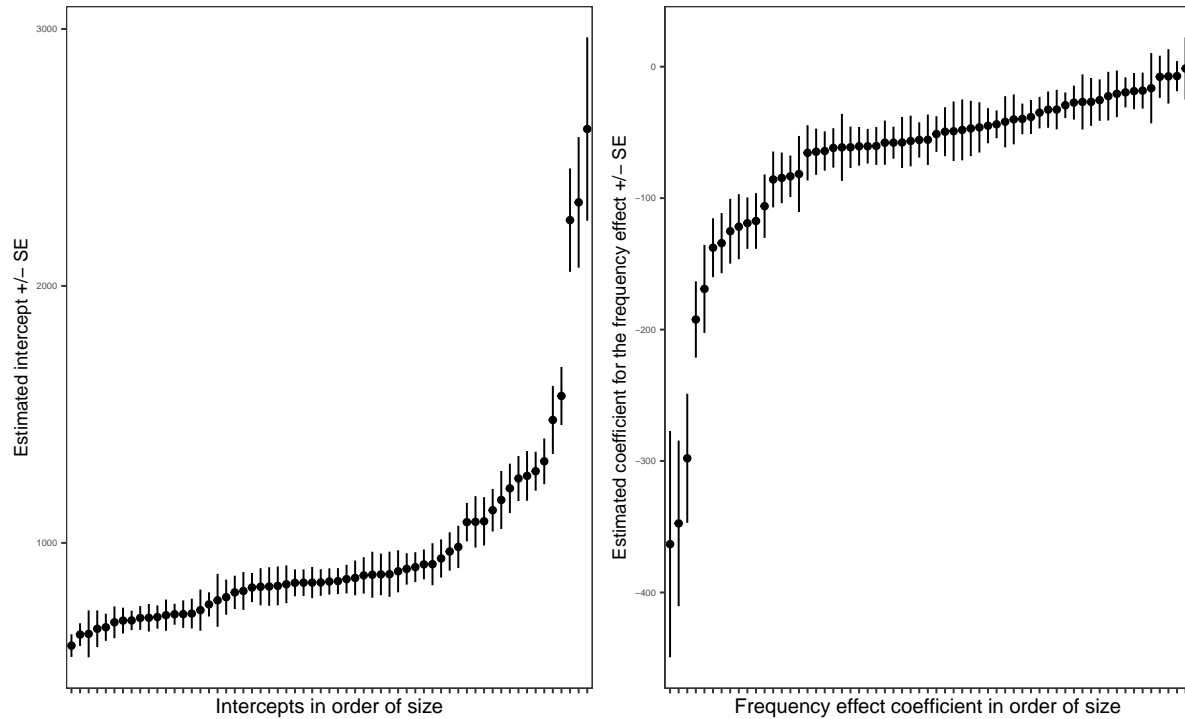


Figure 5: Estimated intercepts and frequency effect slopes (with SEs) calculated for each child analysed separately, with point estimates presented in order of size

1.8.4 Multilevel – here, more appropriately known as – mixed-effects models

In a mixed-effects model, we account for this variation: the differences between participants in intercepts and slopes. We do this by modeling the intercept as two terms:

$$\beta_{0i} = \gamma_0 + U_{0i} \quad (2)$$

- where γ_0 is the average intercept and U_{0i} is the difference for each i child between their intercept and the average intercept.

We model the frequency effect as two terms:

$$\beta_{1i} = \gamma_1 + U_{1i} \quad (3)$$

- where γ_1 is the average slope and U_{1i} represents the difference for each i child between the slope of their frequency effect and the average slope.

We can then incorporate in a single model the **fixed effects** due to the average intercept and the average frequency effect, as well as the **random effects**, error variance due to unexplained differences between participants in intercepts and frequency effects:

$$Y_{ij} = \gamma_0 + \gamma_1 X_j + U_{0i} + U_{1i} X_j + e_{ij} \quad (4)$$

- where the outcome Y_{ij} is related to ...
- the average intercept γ_0 and differences between i children in the intercept U_{0i} ;
- the average effect of the explanatory variable frequency $\gamma_1 X_j$ and differences between i participants in the slope $U_{1i} X_j$;
- in addition to residual error variance e_{ij} .

1.8.4.1 What are we doing with these random effects terms? Note that in sections 1.8.6 and 1.10, we look at what *exactly* is captured in these random effects terms U_{0i}, U_{1i} . Let's first look at the practicalities of analysis then come back to deepen our understanding a bit more.

Right now, it is important to understand that in our analysis we do not care about the differences between *specific* children. We care that there are differences. And we care how widely spread are the differences between child A and the average intercept (or slope), or between child B and the average intercept (or slope), or between child C ... (you get the idea). Therefore, in our analysis, we estimate the spread of the differences as a *variance term*. We can see this when we look at the results of the mixed-effects model we specify, next.

1.8.4.2 Fitting a mixed-effect model using the lmer() function We can fit a mixed-effects model of the $RT \sim frequency$ relationship, taking into account the random differences between participants. I first go through the model fitting code bit by bit. (I then go through the output, the results.)

```
lmer.all.1 <- lmer(RT ~ Lg.UK.CDcount + (Lg.UK.CDcount + 1 || subjectID),
                  data = long.all.noNAs)

summary(lmer.all.1)
```

You have seen the `lmer()` function code before but *practice makes perfect* so we shall go through the code step by step, as we did previously. This time, notice what is different versus what stays the same.

First, we have a chunk of code mostly similar to what we do when we do a regression analysis.

1. `lmer.all.1 <- lmer(...)` creates a *linear mixed-effects model* object using the `lmer()` function.
2. `RT ~ Lg.UK.CDcount` is a formula expressing the model in which we estimate the fixed effect on the outcome or dependent variable `RT` (reaction time, in milliseconds) predicted \sim by the independent or predictor variable `Lg.UK.CDcount` (word frequency).
3. `...(data = long.all.noNAs)` specifies the dataset in which you can find the variables named in the model fitting code.
4. `summary(lmer.all.1)` gets a summary of the fitted model object, showing you the results.

Second, we have the bit that is specific to multilevel or mixed-effects models.

- We add `(... || subjectID)` to tell R about the random effects corresponding to random differences between sample groups (here, observations grouped by child) that are coded by the `subjectID` variable.
- `(... 1 || subjectID)` says that we want to estimate random differences between sample groups (observations by child) in intercepts, where the intercept is coded by 1.
- `(Lg.UK.CDcount ... || subjectID)` adds random differences between sample groups (observations by child) in slopes of the frequency effect coded using the `Lg.UK.CDcount` variable name.

1.8.4.2.1 What does || mean? I want you to notice something that looks like nothing much: ||. We are going to need to defer until later a (necessary) discussion of exactly why we need the two double lines. In short, the use of || asks R to fit a model in which we estimate random effects associated with

- variance due to differences in intercepts
- variance due to differences in slopes
- but *not* covariance between the two sets of differences

I do this because otherwise the model I specify will not converge. We shall need to discuss these things: **convergence**, and failures to converge; as well as **random effects specification** and simplification. We will discuss random effects covariance in Section 1.10. For now, the most important lesson is learnt by seeing how the analysis approach we saw last week can be extended to examining the effects of experimental variables in data from repeated measures design studies.

1.8.4.3 Reading the lmer() results The `lmer()` model code we discussed in Section 1.8.4.2 gives us the following output.

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: RT ~ Lg.UK.CDcount + ((1 | subjectID) + (0 + Lg.UK.CDcount |
##   subjectID))
##   Data: long.all.noNAs
##
## REML criterion at convergence: 117805.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.7839 -0.5568 -0.1659  0.3040 12.4850
##
## Random effects:
##   Groups      Name                Variance Std.Dev.
##   subjectID   (Intercept)         87575    295.93
##   subjectID.1 Lg.UK.CDcount       2657      51.55
##   Residual                                23734    154.06
## Number of obs: 9085, groups:  subjectID, 61
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)   950.913    39.216  24.248
## Lg.UK.CDcount  -67.980     7.092  -9.586
##
## Correlation of Fixed Effects:
##              (Intr)
## Lg.UK.CDcnt -0.093
```

We discussed the major elements of the results output last week. We expand on that discussion, a little, here.

The output from the model summary first gives us information about the model.

- First, we see information about the function used to fit the model, and the model object created by the `lmer()` function call

- Then, we see the model formula `RT ~ Lg.UK.CDcount + (Lg.UK.CDcount + 1|subjectID)`
- Then, we see **REML criterion at convergence** about the model fitting process, which we can usually ignore
- Then, we see information about the distribution of the model residuals.

We then see information listed under **Random effects**.

This is where you can see information about the error variance terms estimated by the model.

The information is listed in four columns: 1. **Groups**; 2. **Name**; 3. **Variance**; and 4. **Std.Dev.** You will recall that we have talked about how observations can be grouped by participant (because we have multiple response observations for each person in the study) just as previously we talked about how observations could be grouped by class (because we saw that children were nested under class). That is what we mean when we refer to **Groups**, we are identifying the grouping variables that give hierarchical structure to the data. The **Name** lists whether the estimate we are looking at corresponds to, here, random differences between participants in intercepts (listed as **(Intercept)**), or in slopes (listed as **Lg.UK.CDcount**). As we discuss later, in Section 1.10, mixed-effects models estimate the spread in random differences. We are not interested in the specific differences in intercept or slope between specific individuals. What we want is to be able to take into account the variance associated with those differences.

Thus, we see in the **Random Effects** section, the variances associated with:

- **subjectID Intercept** 87575, differences between participants in the intercepts;
- **subjectID.1 Lg.UK.CDcount** 2657, differences between participants in the slopes of the frequency effect;
- Alongside **Residual** 23734, residuals where, just like a linear model, we have variance associated with differences between model estimates and observed RT, here, at the trial level.

We do not usually discuss the specific variance estimates in research reports. However, the relative size of the variances does provide useful information (see also Meteyard & Davies, 2020), as we shall see when we discuss the different estimates we get when we include a random effect due to differences between items (Section 1.9.1.1).

Lastly, we see estimates of the coefficients (of the slopes) of the fixed effects.

In this model, we see estimates of the fixed effects of the intercept and the slope of the `RT ~ Lg.UK.CDcount` model. We discuss these estimates next.

1.8.5 Is there a difference between linear model and linear mixed-effects model results?

Recall that the linear model yields the estimate for the frequency effect on reading RT such that RT decreases by about 53 ms for unit increase in log word frequency ($\beta = -53.375$). Now, when we have taken random differences between participants into account, we see that the estimate of the effect **for the mixed-effects model** is $\beta = -67.980$. This is a noteworthy difference in our estimate for the effect. As we saw last, we see, again, that taking into account random differences has an impact on results.

Which coefficient estimate should you trust? Well, it is obvious that the linear model and the linear mixed-effects model estimate are relatively similar. However, it is also obvious that the linear model makes an assumption – the *assumption of independence of observations* – that does not make sense theoretically (we can readily expect that reading responses will be similar within a child) and does not make sense empirically (responses clearly differ between children, Figure 5). Thus, I think we have good grounds for supposing that

the linear mixed-effects model estimate for the frequency effect is likely to be closer to the true underlying population effect (whatever that might be).

That being said, it is important to remember, in this discussion, that whatever estimate we can produce is the estimate we can produce *given* the sample of words we used, the measurement of reading RT we were able to make, and the estimate of word frequency we were able to collect. How far our estimate actually generalizes to the wider population is not something we can settle in the context of a single study.

Further, we have not finished in our consideration of the random effects that the account should include. We need to do more work by thinking about the differences between stimuli (Section 1.9).

1.8.5.1 Why aren't there p-values? We will come back to this but note that if $t > 2$ we can suppose that an effect is significant at the .05 significance level.

1.8.6 What we estimate when we estimate random effects

We have said that we can incorporate, in a mixed-effects model, **fixed effects** (e.g., the average frequency effect) and **random effects**, error variance due to unexplained differences between participants in intercepts and in frequency effects:

$$Y_{ij} = \gamma_0 + \gamma_1 X_j + U_{0i} + U_{1i} X_j + e_{ij} \quad (5)$$

So we distinguish:

- the average intercept γ_0 and differences between i children in the intercept U_{0i} ;
- the average effect of the explanatory variable frequency $\gamma_1 X_j$ and differences between i participants in the slope $U_{1i} X_j$.

When we think about the differences between participants (or between the units of any grouping variable), in intercepts or in slopes, we should understand that for the mixed-effects model, the differences are:

- random;
- should be normally distributed;
- and are distributed around the population or average fixed effects.

We should understand that the mixed-effects model sees the differences between participants **relative to the fixed effect intercept or slope**, that is, relative to the population level or average effects. We can illustrate this by plotting, in Figure 6, the differences as estimated (technically, predicted) by the mixed-effects model that we discussed in sections 1.8.4.2 and 1.8.4.3.

What you can see in Figure 6 are distributions. The centers of the distributions are on zero (shown by a red line). For each distribution (a. and b.), that is where the model estimate of the intercept or the slope of the frequency effect is located. Spread around that central point, you see the adjustments the model makes to account for differences between participants.

You can see how in Figure 6 (a.), some children have intercepts that are smaller than the population-level or average intercept – so their adjustments are negative (to decrease their intercepts). In comparison, some children have intercepts that are larger than the population-level or average intercept – so their adjustments are positive (to increase their intercepts). Strikingly, you can see that a few children have intercepts that are as much as 1000ms larger than the population-level or average intercept.

You can see also how in Figure 6 (b.), some children have frequency effects (coefficients) that are smaller than the population-level or average frequency effect – so their adjustments are positive (to decrease their frequency effect, by making it *less* negative). (Remember the estimated β coefficient for the frequency effect

is negative because higher word frequency is associated with smaller RT.) In comparison, some children have frequency effects that are larger than the population-level or average frequency effect – so their adjustments are negative (to increase their frequency effect, by making it *more* negative). Strikingly, you can see that a few children have frequency effects that are as much as 200ms larger (see plot (b.) around $x = -200$) than the population-level or average effect.

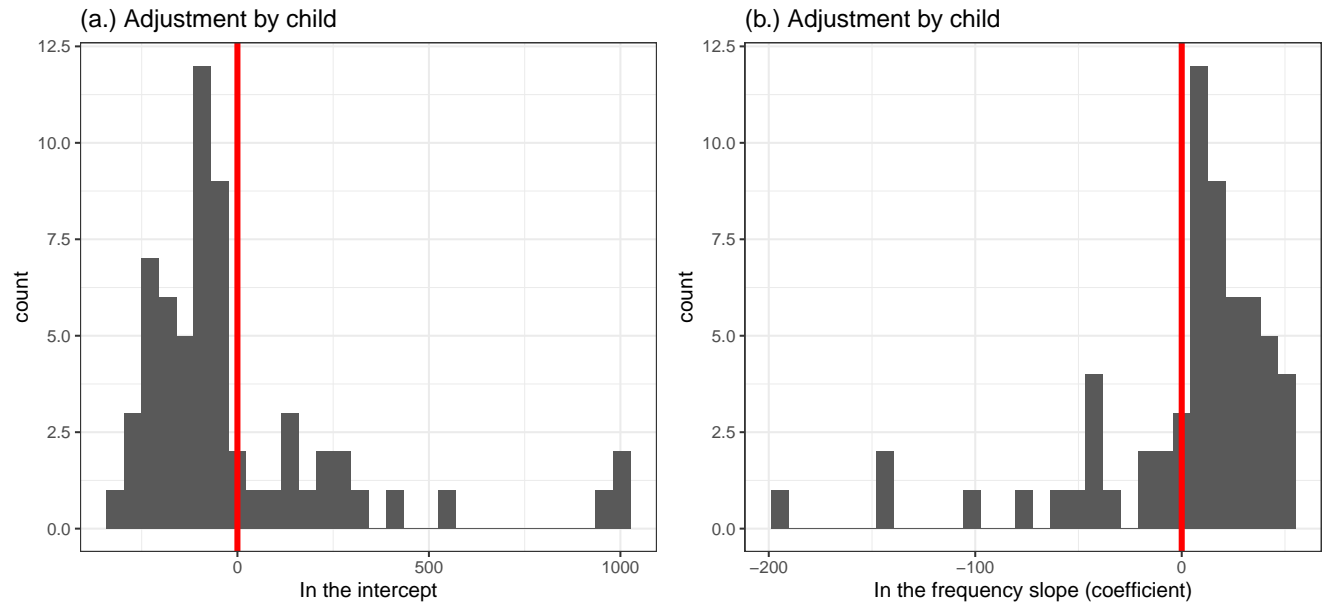


Figure 6: Plot showing histograms indicating the distribution of participant adjustments to account for between-child differences in intercept or slope (known as Best Linear Unbiased Predictionss)

When a mixed-effects model is fitted to a dataset, its set of estimated parameters includes the coefficients for the fixed effects as well as the standard deviations for the random effects (Baayen, 2008). The individual values of the adjustments made to intercepts and slopes are calculated once the random effects have been estimated. If you read the literature on mixed-effects models, you will see that the adjustments are called Best Linear Unbiased Predictors (or BLUPs).

1.8.6.1 Exercise Mixed-effects modeling is hard to get used to *at first*. A bit more practice helps to show you how the different parts of the model work. We again focus on the random effects.

- In the model we have seen so far, we specify `(Lg.UK.CDcount + 1||subjectID)`
- We can change this part – and only this part – to see what happens to the results, do this:
 1. `lmer(RT ~ Lg.UK.CDcount + (1|subjectID)...)` gives us a *random intercepts* model accounting for just random differences between participants in the intercept
 2. `lmer(RT ~ Lg.UK.CDcount + (Lg.UK.CDcount + 0|subjectID)...)` gives us a *random slope* model accounting for just random differences between participants in the slope of the frequency effect
 3. `lmer(RT ~ Lg.UK.CDcount + (Lg.UK.CDcount + 1|subjectID)...)` gives us a *random intercepts and slopes* model accounting for both random differences between participants in the intercept and in the slope, as well as covariance in these differences.

Try out these variations and *look carefully* at the different results. Look, especially, at what happens to the **Random effects** part of the summary.

We can *visualize* the differences between the models in a plot showing the different predictions that the different models give us. Figure 7 shows what a mixed-effects model would predict should be the effect of frequency on RT for different children in the CP study. The predictions vary depending on the nature of the random effects we specify in the model.

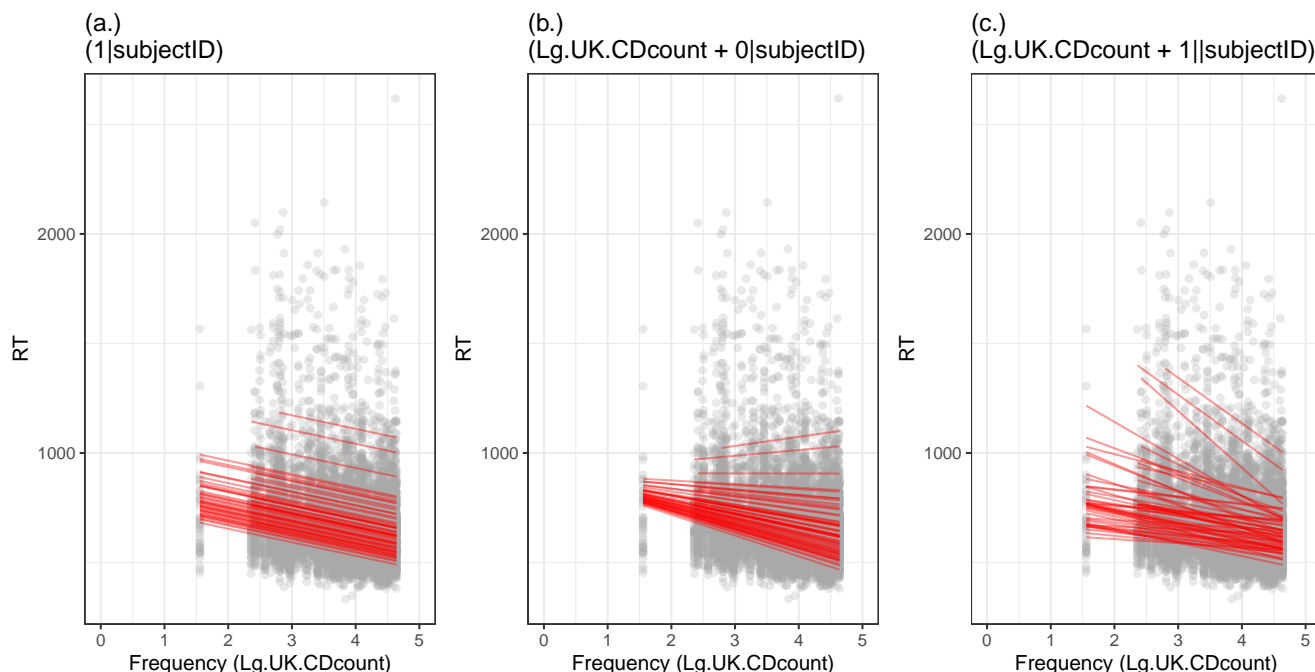


Figure 7: Plot showing model predictions of the effect, for each individual, of word frequency on reading reaction time – predictions vary between models incorporating (a.) random effect of participants on intercepts only; (b.) random effect of participants on slopes only and (c.) random effect of participants on intercepts and on slopes

We can see that:

1. If the model includes the random effect of *participants on intercepts only* then all the slopes are the same (the lines in the figure are parallel) because this model assumes that the only differences between participants are differences in the intercepts.
2. If the model includes the random effect of *participants on slopes only* then the slopes vary but they all have the same intercept. The plot does not show this but you can see how all the slopes are converging on one point somewhere on the left. This happens because this model assumes that the only differences between participants are differences in the slopes.
3. If the model includes the random effect of *participants on intercepts and on slopes* then we can see how the intercepts and the slopes vary. Given what we saw when we looked at the relation between frequency and RT for each participant considered separately we might argue that this model is much more realistic about the data.

1.8.6.2 Code tip It is very important to learn to make effective use of the warnings and error messages R can produce.

Note: you do not have to just believe me when I say that `||` is in the model code to stop a problem appearing. Experiment – and see what happens when you change the code. Try this.

```
lmer.all.1 <- lmer(RT ~ Lg.UK.CDcount + (Lg.UK.CDcount + 1|subjectID),
  data = long.all.noNAs)
summary(lmer.all.1)
```

Do you get an error message?

A very useful trick is to learn to copy the error message you get into a search engine on your web browser. Do this and you will find useful help, as here.

https://rstudio-pubs-static.s3.amazonaws.com/33653_57fc7b8e5d484c909b615d8633c01d51.html

1.9 Variation between stimuli: the “language as fixed-effect fallacy”

Experimental psychologists will often collect data in studies where they present some stimuli to a sample of participants (as CP did in her study). Clark (1973) showed that the appropriate analysis of experimental effects for such data requires the researcher to take into account the error variance due to unexplained or random differences between sampled participants *and* to random differences between sampled stimuli. This is true in the context of psycholinguistics but it is also true in the context of work in any field where the presented stimuli can be understood to constitute a sample from a wider population of potential stimuli (Judd, Westfall, & Kenny, 2012).

If we were to estimate the average latency of the responses made by different children to each word, we would see that there is considerable variation between words (Figure 8). Some words elicit slower and some elicit faster responses on average. We can also see that there is, again, variation in the uncertainty of estimates, as reflected in differences in the lengths of the error bars corresponding to the standard errors of the estimates.

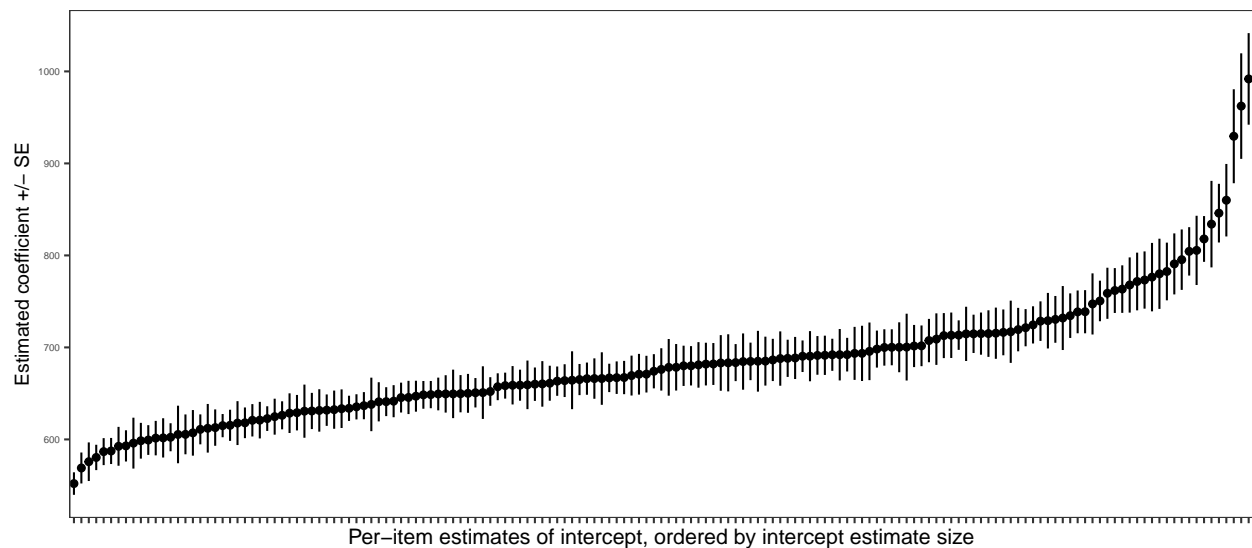


Figure 8: Estimated intercepts (with SEs) calculated for each stimulus word, with coefficients ordered by average latency for each word

In general, psychologists have been aware since Clark (1973, if not earlier) that responses to experimental stimuli can vary because of random or unexplained differences between the stimuli: whether the stimuli are words, pictures or stories, etc. And researchers have been aware that if we did not take such variation into account, we might mistakenly detect an experimental effect, for example, as a significant difference between

mean response in different conditions, simply because different stimuli presented in different conditions varied in some unknown way, randomly, in relative difficulty.

For many years, psychologists tried to take random differences between stimuli into account, alongside random differences between participants, using a variety of strategies with important limitations (see Baayen et al., 2008, for discussion). Clark (1973) suggested that researchers could calculate $minF'$ (not F) when doing Analyses of Variance of experimental data

This involves a series of steps.

1. You start by *aggregating* your data
 - By-subjects data – for each subject, take the average of their responses to all the items
 - By-items data – for each item, take the average of all subjects' responses to that item
2. You do separate ANOVAs, one for by-subjects (F_1) data and one for by-items (F_2) data
3. You put F_1 and F_2 together, calculating $minF'$

Averaging data by-subjects or by-items is relatively simple. And you will often see, in the literature, psychological reports in which F_1 and F_2 analysis results are presented.

Calculating $minF'$ is also relatively simple:

$$minF' = \frac{MS_{effect}}{MS_{random-subject-effects} + MS_{random-word-differences}} = \frac{F_1 F_2}{F_1 + F_2} \quad (6)$$

However, after a while, psychologists stopped doing the extra step of the $minF'$ calculation (Raaijmakers et al., 1999). They carried on calculating and reporting F_1 and F_2 ANOVA results but, as Baayen et al. (2008) discuss, this approach risks a high potential false positive error rate.

Psychologists also found that while the $minF'$ approach allowed them to take into account between-participant and between-stimulus differences it could not be applied where ANOVA could not be used. This stopped researchers from taking a comprehensive approach to error variance where they wanted to conduct multiple regression analyses. You will often see multiple regression analyses of by-items data, where a sample of participants has been asked to respond to a sample of stimuli, and the analysis is of the effects of stimulus properties on outcomes averaged (over participants' responses) to the mean outcome by item. But analyzing data only by-items ensures that we lose track of participant differences. Lorch and Myers (1990) warn that analyzing only by-items mean RTs just assumes wrongly that *subjects are a fixed effect*. This approach, again, risks a higher rate of false positive errors.

1.9.1 Include the random effect of stimulus

We now no longer need to tolerate these problems.

In the context of our working example, with our analysis of the CP study data, we can build up our mixed-effects model by adding a random effect to capture the impact of unexplained differences between stimuli. We model the random effect of items on intercepts by modeling the intercept as two terms:

$$\beta_{0j} = \gamma_0 + W_{0j} \quad (7)$$

- where γ_0 is the average intercept and W_{0j} represents the deviation, for each word, between the average intercept and the per-word intercept.

Our model can now incorporate the additional random effect of items on intercepts:

$$Y_{ij} = \gamma_0 + \gamma_1 X_j + U_{0i} + U_{1i} X_j + W_{0j} + e_{ij} \quad (8)$$

In this model, the outcome Y_{ij} is related to the average intercept γ_0 and the word frequency effect $\gamma_1 X_j$ plus random effects due to unexplained differences between participants in intercepts U_{0i} and the slope of the frequency effect $U_{1i} X_j$ as well as random differences between items in intercepts W_{0j} , in addition to the residual term e_{ij} .

1.9.1.1 Fitting a mixed-effect model – now with random effects of subjects and items We can fit a mixed-effects model of the $RT \sim frequency$ relationship, taking into account the random differences between participants *and now also* the random differences between stimulus words.

```
lmer.all.2 <- lmer(RT ~ Lg.UK.CDcount +
                  (Lg.UK.CDcount + 1||subjectID) +
                  (1|item_name),
                  data = long.all.noNAs)

summary(lmer.all.2)

## Linear mixed model fit by REML ['lmerMod']
## Formula: RT ~ Lg.UK.CDcount + ((1 | subjectID) + (0 + Lg.UK.CDcount |
##      subjectID)) + (1 | item_name)
##      Data: long.all.noNAs
##
## REML criterion at convergence: 116976.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.1795 -0.5474 -0.1646  0.3058 12.9485
##
## Random effects:
##      Groups      Name      Variance Std.Dev.
## item_name      (Intercept)    3397    58.29
## subjectID      Lg.UK.CDcount   3624    60.20
## subjectID.1    (Intercept)  112314   335.13
## Residual                        20704   143.89
## Number of obs: 9085, groups: item_name, 159; subjectID, 61
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)    971.07     51.87  18.723
## Lg.UK.CDcount  -72.33     10.79  -6.703
##
## Correlation of Fixed Effects:
##              (Intr)
## Lg.UK.CDcnt -0.388
```

This is the same mixed-effects model as the one we discussed in sections 1.8.4.2 and 1.8.4.3 but with one important addition.

- We add (1|item_name) to take into account random differences between between words in intercepts

1.9.1.1.1 Reading the results Take a look at the model results. You should notice three changes.

1. You can see that the estimate for the effect of word frequency on reading reaction time has changed again, it is now $\beta = -72.33$
2. `item_name (Intercept)` 3397 there is now an additional term in the list of random effects, giving the model estimate for variance associated with random differences between words in intercepts
3. And you can see that the residual variance has changed. In the first model `lmer.all.1` it was 23734, now it is 20704

The reduction in residual variance is one way in which we can judge how good a job the model is doing in accounting for the variance in the outcome, observed response reaction time. We can see that by adding a term to account for differences between items we can reduce the amount by which the model estimates deviate from observed outcomes. This difference in error variance is, essentially, one basis for estimating how well the model fits the data, and a basis for estimating the *variance explained* by a model in terms of the R^2 statistic you have seen before. We will come back to this.

1.10 Variances and covariances of random effects

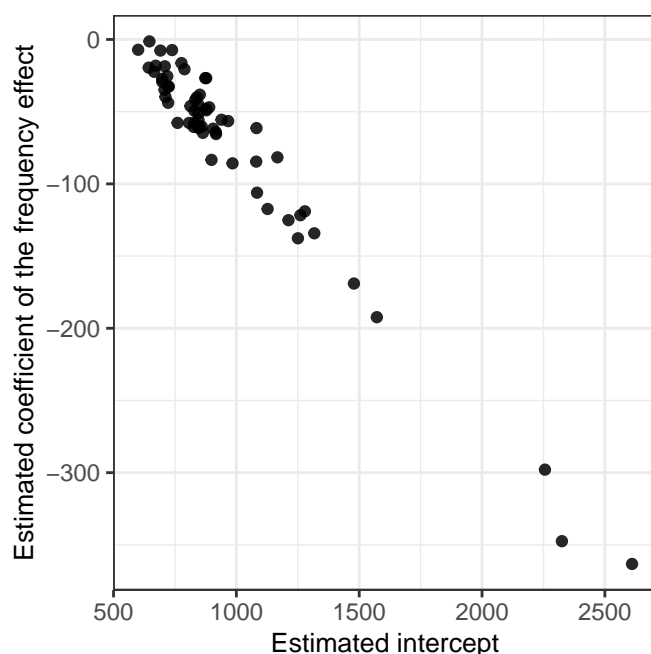


Figure 9: Scatterplot showing the relationship between estimated coefficients for the intercept and for the frequency effect, for each child analysed separately

As I have said, we usually do not aim to examine the specific deviation from the average intercept or the average fixed effect slope for a participant or stimulus. We estimate just the spread of deviations by-participants or by-items. A mixed-effects model like our final model includes fixed effects corresponding to the intercept and the slope of the word frequency effect plus the variances:

- $var(U_{0i})$ variance of deviations by-participants from the average intercept;
- $var(U_{1i}X_j)$ variance of deviations by-participants from the average slope of the frequency effect;
- $var(W_{0j})$ variance of deviations by-items from the average intercept;

- $var(e_{ij})$ residuals, at the response level, after taking into account all other terms.

We may expect the random effects of participants or items to covary, e.g., participants who are slow to respond may also be more susceptible to the frequency effect, as can be seen in Figure 9. Thus, we could specify the random effects of the model can incorporate terms corresponding to the covariance of random effects:

- $covar(U_{0i}, U_{1i}X_j)$

1.10.1 But remember we excluded random effects covariance

In Section 1.8.4.2.1, I noted how we used the `||` notation to stop the model estimating the covariance between differences between participants in intercepts and in slopes. The reason I did this is that if I had requested that the model estimate the covariance the model would have failed to converge. What this means depends on understanding how mixed-effects models are estimated. We shall have to return to a development of that understanding later. For now, it is enough to note that mixed-effects models fitted with `lmer()` often have more difficulty with random effects covariance estimates.

1.11 Reporting the results of a mixed-effects model

There is no official convention on what or how to report the results of a mixed-effects model. Lotte Meteyard and I suggest what psychologists should report in an article (Meteyard & Davies, 2020) that has been downloaded a few thousand times so, maybe, our advice will help to influence practice.

We would argue that researchers should explain what analysis they have done and, where space allows, should report both the estimates of the **fixed effects** and the estimates of the **random effects**. We think you can report the model code (maybe in an appendix, maybe in a note under a tabled summary of results).

Coefficients	Estimate	SE	t
(Intercept)	971.1	51.9	18.7
Frequency effect	-72.3	10.8	-6.7

Groups	Name	Variance	SD
item	(Intercept)	3397	58.3
participant	(Intercept)	112314	335.1
participant	Frequency	3624	60.2
residual		20704	143.9

Note: `lmer(RT ~ Lg.UK.CDcount + (Lg.UK.CDcount + 1||subjectID) + (1|item_name)`

Researchers should report their modelling in sufficient detail that their results can be reproduced by others. Barr et al. (2013) argued that choices about random effects structure affect the generalizability of the estimates of fixed effects. In particular, it seems sensible to examine the possibility that the slope of the effect of an explanatory variable may vary at random between participants or between stimuli. Correspondingly, researchers should report and explain their decisions about the inclusion of random effects.

It is normal practice in psychology to report the p-values associated with null hypothesis significance tests of effects when reporting analysis. Performing hypothesis tests using t- or F-distributions depends on the calculation of degrees of freedom yet it is uncertain how degrees of freedom should be counted when analysing

multilevel data (Baayen et al., 2008). In most software applications, however, p-values associated with fixed effects may be calculated using an approximation for denominator degrees of freedom.

We will come back to how we should report the results of mixed-effects models because, here, too, we can benefit by developing our approach, in depth, step by step.

1.12 Conclusions

A large proportion of psychological studies involves scenarios in which the researcher samples both participants and some kind of stimuli. Often, the researcher will present the stimuli to the participants for response in some version of a range of possible designs: all participants see and respond to all stimuli; participants respond to different sub-sets of stimuli in different conditions (or in different groups) but they see and respond to all stimuli in a sub-set; participants are allocated to respond to stimulus sub-sets according to a counter balancing scheme (e.g., through the use of Latin squares). Whatever version of this scenario, *if* participants are responding to multiple stimuli and *if* multiple participants respond to each stimulus, then the data will have a multilevel structure such that each observation can be grouped both by participant and by stimulus. We are interested in taking into account the random effects associated with unexplained or random differences between participants or between stimuli. We often discuss the accounting of these effects in terms of the estimation of error variances associated with the random differences, calling the effects of the differences *random effects*. Where we have to deal with both samples of participants and samples of stimuli, we can talk about *crossed random effects*.

The terms are not that important. The insight is: in general, in experimental psychological science, when we do data analysis, if we want to estimate effects of experimental variables more accurately then our models need to incorporate terms to capture the impact on observed outcomes of sampled participants and sampled stimuli. Historically, we have, as a field, learned to take into account these sampling effects. Now, and most likely, more and more commonly in the future, we are learning to use multilevel or mixed-effects models to do this.

1.12.1 Summary

We discussed the way that data are structured when they come from studies with repeated measures designs. Critically, we examined data from a common study design where a sample of stimulus items are presented for response to members of a participant sample. This means that each observation can be grouped by participant and, also, by stimulus. The possibility that observations can be grouped means that the data have a multilevel structure. The multilevel structure requires the use of linear mixed-effects models when we seek to estimate the effects of experimental variables. The fact that data can be grouped both by participant and by stimulus means that the model can incorporate random effects to capture random between-participant differences as well as between-stimulus differences. The use of mixed-effects models has meant that psychologists no longer need to adopt compromise solutions which have important limitations, like by-items and by-subjects analyses.

We reviewed the ways that experimental data can be untidy. And we outlined the steps that may be required to process untidy data into a tidy format suitable for analysis. As is typical for the data analysis we need to do for experimental psychological science, getting data ready for analysis requires a series of steps including: access; import; restructure; select variables; and filter observations.

We then developed a mixed-effects model to answer the research question:

RQ.1. What word properties influence responses to words in a test of reading aloud?

Our analysis focused on the relationship between reading response reaction time (RT, in ms) and the predictor word frequency. We examined how the effect of word frequency was estimated in a linear model ignoring the multilevel structure and then in mixed-effects models which incorporated terms to capture variance

associated with random differences between participants in intercepts or in the slope of the frequency effect, and between items in intercepts.

We saw that estimates of the frequency effect differed between different models.

We considered the possibility of within-items effects.

1.12.2 Useful functions

We used a number of functions to tidy, visualize and analyze the CP study data.

- `read_csv()` and `read_csv()` to load source data files into the R workspace
- `pivot_longer()` to restructure data from wide to long
- `full_join()` to put together data from separate datasets; in our example, from datasets holding information about participant attributes, stimulus word properties, and participant behaviours
- `select()` to select the variables we need
- `filter()` to filter observations based on conditions
- `na.omit()` to remove missing values
- For visualisation, we used `facet_wrap()` to show plots of the relationship between outcome and predictor variables separately for different groups (by participant, or by item)
- We used `lmer()` to fit a multilevel model

We used the `summary()` function to get model results for both linear models and for the mulilevel or liner mixed-effects model.

1.13 R code and data file access for the class

Activities in the class that goes with this chapter are associated with the following data file and .R code file:

- 402-02-mixed-effects-workbook.R
- CP study word naming rt 180211.dat
- CP study word naming acc 180211.dat
- words.items.5 120714 150916.csv
- all.subjects 110614-050316-290518.csv

A pre-tidied version of the CP study data is available as:

- long.all.noNAs.csv

You can get these materials by going to the 402 Moodle folder for week 18, and downloading the .zip (compressed) folder labeled **PSYC402-01-multilevel-resources**

Or, you can download the same folder by clicking on the link:

<https://modules.lancaster.ac.uk/mod/resource/view.php?id=1795341>

Run the code in the .R file to reproduce the results presented in this chapter and in the slides.

1.14 References

1.14.1 Recommended reading

Snijders and Bosker (2012) present a helpful overview of multilevel modelling. Baayen et al. (2008; see, also, Barr et al., 2013; Judd et al., 2012) discuss mixed-effects models with crossed random effects. Readers familiar with the book will see that I rely on it to construct the formal presentation of the models.

I wrote a tutorial article on mixed-effects models with Lotte Meteyard. We discuss how important the approach now is for psychological science, what researchers worry about when they use it, and what they should do and report when they use the method/

Meteyard, L., & Davies, R.A.I. (2020). Best practice guidance for linear mixed-effects models in psychological science, *Journal of Memory and Language*, 112, 104092, <https://doi.org/10.1016/j.jml.2020.104092>

1.14.2 References list

Baayen, R. H., Davidson, D. J., & Bates, D. M. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59, 390-412.

Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, 68, 255-278.

Davies, R. A., Arnell, R., Birchenough, J. M., Grimmond, D., & Houlson, S. (2017). Reading through the life span: Individual differences in psycholinguistic effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 43, 1298.

Judd, C. M., Westfall, J., & Kenny, D. A. (2012). Treating stimuli as a random factor in social psychology: a new and comprehensive solution to a pervasive but largely ignored problem. *Journal of Personality and Social Psychology*, 103, 54.

Lorch, R. F., Jr., & Myers, J. L. (1990). Regression analyses of repeated measures data in cognitive research. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16, 149-157. <http://dx.doi.org/10.1037/0278-7393.16.1.149>

Raaijmakers, J. G., Schrijnemakers, J. M., & Gremmen, F. (1999). How to deal with “the language-as-fixed-effect fallacy”: Common misconceptions and alternative solutions. *Journal of Memory and Language*, 41, 416-426.

Snijders, T.A., & Bosker, R.J. (2012). *Multilevel analysis (2nd Edition)*. London, UK: Sage.

Steegen, S., Tuerlinckx, F., Gelman, A., & Vanpaemel, W. (2016). Increasing transparency through a multiverse analysis. *Perspectives on Psychological Science*, 11, 702-712.