

# **Textbook English: A Multi-Dimensional Approach**

**Online Supplements**

Elen Le Foll

2024-03-06

# Table of contents

<b>Preface</b>	<b>4</b>
<b>1 Introduction</b>	<b>5</b>
1.1 Research objectives and methodological approach . . . . .	5
1.2 Outline of the book . . . . .	7
<b>2 Open Science statement</b>	<b>8</b>
<b>Appendix A: Literature review</b>	<b>10</b>
<b>Appendix B: Corpus data</b>	<b>11</b>
Textbook English Corpus (TEC) . . . . .	11
Reference corpora . . . . .	11
Spoken BNC2014 . . . . .	11
Informative Texts for Teens Corpus (Info Teens) . . . . .	11
Youth Fiction corpus . . . . .	12
<b>Appendix D: Data Preparation for the Model of Intra-Textbook Variation</b>	<b>13</b>
Packages required . . . . .	13
Data import from MFTE output . . . . .	13
Corpus size . . . . .	17
Data preparation for PCA . . . . .	17
Feature distributions . . . . .	18
Feature removal . . . . .	21
Identifying potential outlier texts . . . . .	22
Signed log transformation . . . . .	27
Feature correlations . . . . .	33
Composition of TEC texts/files . . . . .	35
<b>Appendix E: Data Analysis for the Model of Intra-Textbook Variation</b>	<b>40</b>
Packages required . . . . .	40
Preparing the data for PCA . . . . .	41
TEC data import . . . . .	41
Checking the factorability of data . . . . .	41
Removal of feature with MSAs of < 0.5 . . . . .	42
Choosing the number of principal components to retain . . . . .	42

Excluding features with low final communalities . . . . .	43
Testing the effect of rotating the components . . . . .	44
Principal Component Analysis (PCA) . . . . .	45
Using the full dataset . . . . .	45
Using random subsets of the data . . . . .	46
Using specific subsets of the data . . . . .	46
Performing the PCA . . . . .	46
Plotting PCA results . . . . .	47
3D plots . . . . .	47
Two-dimensional plots (biplots) . . . . .	48
Data wrangling for PCAtools . . . . .	48
Pairs plot . . . . .	49
Bi-plots . . . . .	50
Feature contributions (loadings) on each component . . . . .	55
Exploring the dimensions of the model . . . . .	61
Computing mixed-effects models of the dimension scores . . . . .	63
Dimension 1: ‘Overt instructions and explanations’ . . . . .	63
Dimension 2: ‘Involved vs. Informational Production’ . . . . .	67
Dimension 3: ‘Narrative vs. factual discourse’ . . . . .	75
Dimension 4: ‘Informational compression vs. elaboration’ . . . . .	81
Testing model assumptions . . . . .	87
<b>3 Summary</b>	<b>89</b>
<b>References</b>	<b>90</b>

# Preface

This Quarto book is **work in progress**. It will eventually contain the online supplements to:

Le Foll, Elen. to appear. *Textbook English: A Multi-Dimensional Approach* [Studies in Corpus Linguistics]. Amsterdam: John Benjamins.

The book is based on my PhD thesis, which is accessible in Open Access:

Le Foll, Elen. 2022. *Textbook English: A Corpus-Based Analysis of the Language of EFL textbooks used in Secondary Schools in France, Germany and Spain*. Osnabrück, Germany: Osnabrück University. PhD thesis. <https://doi.org/10.48693/278>.

# 1 Introduction

Asked “Where is Brian?”, French nationals of a certain generation will immediately reply: “Brian is in the kitchen”. Those with a particularly good memory may follow up with: “Where is Jenny, the sister of Brian?” – and, to those in the know, the correct answer is: “Jenny is in the bathroom”.<sup>1</sup> There is hardly any need for an in-depth linguistic analysis to conclude that this interaction is highly unlikely to have ever taken place in a real English-speaking family home. To most teachers and learners, it will be evident that it is the result of a none too inspired attempt to model WH-question forms in a textbook dialogue aimed at beginner learners of English as a Foreign Language (EFL). Together with dull gap-fill exercises and photos of out-of-date technology, for many adults, the very mention of the word textbook evokes vivid memories of such artificially sounding, contrived and sometimes even nonsensical dialogues.

This raises the question of the status and nature of textbook language as a specific ‘variety’ of language, which is at the heart of the present study. It focuses on contemporary EFL textbooks in use in European secondary schools. Situated at the interface between linguistics and foreign language teaching, this study examines the linguistic content of these textbooks and seeks empirical answers to the questions: What kind of English do school EFL textbooks portray? And how far removed is this variety of English from the kind of English that learners can be expected to encounter outside the EFL classroom?

## 1.1 Research objectives and methodological approach

The above questions are critical because, as many adults’ lingering memories of school foreign language lessons testify (see also, e.g., Freudenstein 2002: 55), textbooks play an absolutely central role in classroom-based foreign language learning. In the following, we will see that the dominance of textbooks in EFL school contexts persists to this day. According to Thornbury (2012 in a response to Chong 2012: n.p.), they “(more often [than] not) instantiate the curriculum, provide the texts, and - to a large extent - guide the methodology”. In lower secondary EFL instructional contexts, in particular, textbooks constitute a major vector of foreign language input. Yet, numerous studies have shown that “considerable mismatches

<sup>1</sup>Dialogue from *Speak English 6<sup>e</sup> série verte* (Benhamou & Dominique 1977: 167). It was made popular by stand-up comedian Gad Elmaleh. More information on the context of this textbook dialogue can be found [here](#). An extract of the comedy sketch by Gad Elmaleh that popularised the dialogue can be viewed here with English subtitles: <https://youtu.be/11jG7lkwDwU?t=50>.

between naturally occurring English and the English that is put forward as a model in pedagogical descriptions” (Römer 2006: 125-26) exist. These mismatches have been observed and sometimes extensively described in textbooks’ representations of numerous language features ranging from the use of individual words and phraseological patterns (e.g., Conrad 2004 on the preposition though; Gouverneur 2008 on the high-frequency verbs make and take), to tenses and aspects (e.g., Barbieri & Eckhardt 2007 on reported speech; Römer 2005 on the progressive). More rarely, textbook language studies have also ventured into the study of spoken grammar (e.g., Gilmore 2004) and pragmatics (e.g., Hyland 1994 on hedging in ESP/EAP textbooks).

However, as we will see in Chapter 2, previous EFL textbook studies have tended to focus on one or at most a handful of individual linguistic features. Taken together, they provide valuable insights into “the kind of synthetic English” (Römer 2004b: 185) that pupils are exposed to via their textbooks; yet, what is missing is a more comprehensive, broader understanding of what constitutes ‘Textbook English’ from a linguistic point of view. Although corpus-based<sup>2</sup> textbook analysis can be traced back to the pioneering work of Dieter Mindt in the 1980s, the language of secondary school EFL textbooks (as opposed to that of general adult EFL or English for Specific Purposes [ESP] coursebooks) remains an understudied area.

The present study therefore sets out to describe the linguistic content of secondary school EFL textbooks and to survey the similarities and most striking differences between ‘Textbook English’ and ‘naturally occurring English’ as used outside the EFL classroom, with respect to a wide range of lexico-grammatical features.

To this end, a corpus of nine series of secondary school EFL textbooks (43 textbook volumes) used at lower secondary level in France, Germany, and Spain was compiled (see 4.3.1). In addition, three reference corpora are used as baselines for comparisons between the language input EFL learners are confronted with via their school textbooks and the kind of naturally occurring English that they can be expected to encounter, engage with, and produce themselves on leaving school. Two of these have been built specifically for this project with the aim of representing comparable ‘authentic’ (for a discussion of this controversial term in ELT, see 2.2) and age-appropriate learner target language.

A bottom-up, corpus-based approach is adopted (e.g., Mindt 1992, 1995a; Biber & Quirk 2012; Biber & Gray 2015; Ronald Carter & McCarthy 2006a). A broad range of linguistic features are considered: ranging from tenses and aspects to negation and discourse markers. We will pay particular attention to the lexico-grammatical aspects of Textbook English that substantially diverge from the target learner language reference corpora and examine these with direct comparisons of textbook excerpts with comparable texts from the reference data.

---

<sup>2</sup>Here the adjectives ‘corpus-based’ and ‘corpus-driven’ are used synonymously (see, e.g., Meunier & Reppen 2015: 499 for further information as to how these terms are sometimes distinguished).

## **1.2 Outline of the book**

The following chapter outlines the background to and motivation behind the present study. Chapter 3 then provides a literature review of state-of-the-art research on the language of school EFL textbooks. It is divided in two parts. Part 1 is a methodological review in which the various methods employed so far to analyse, describe, and evaluate Textbook English are explained and illustrated with selected studies. Part 2 summarises the results of existing studies on various aspects of Textbook English, including lexical, grammatical and pragmatic aspects. Based on the methodological limitations and the gaps identified in the existing literature, Chapter 4 elaborates the specific research questions addressed in the present study. These research questions informed the decision-making processes involved in the compilation of the Textbook English Corpus (TEC) and the selection/compilation of three reference corpora designed to represent learners' target language. These processes and their motivations are explained in the remaining sections of Chapter 4.

Chapter 5 describes the multivariable statistical methods applied to describe the linguistic nature of Textbook English on multiple dimensions of linguistic variation. It begins by explaining the well-established multi-feature/dimensional analysis (MDA) method pioneered by Biber (1988, 1995; see also Berber Sardinha & Veirano Pinto 2014, 2019), before outlining the reasoning for the modified MDA framework applied in the present study. Chapter 6 presents the results of an MDA model of Textbook English which highlights the sources of linguistic variation within EFL textbooks across several dimensions of intra-textbook linguistic variation. Chapter 7 presents the results of a second MDA model that shows how Textbook English is both, in some respects, similar to and, in others, different from the kind of English that EFL learners are likely to encounter outside the classroom.

Chapter 8 explains how the two models contribute to a new understanding of the linguistic characteristics of Textbook English. This, in turn, has implications for teachers, textbook authors, editors, publishers, and policy-makers. These implications are discussed in Chapter 9. It first considers the potential impact of the substantial gaps between Textbook English and the target reference corpora before making suggestions as to how teachers, textbook authors, and editors may want to improve or supplement unnatural-sounding pedagogical texts using corpora and corpus tools. Chapter 10 focuses on the study's methodological strengths and limitations. It explains how the modified MDA framework presented and applied in this study may be of interest to corpus linguists working on a broad range of research questions. Chapter 11 concludes with a synthesis of the most important take-aways from the study. It also points to promising future research avenues.

## 2 Open Science statement

Another important insight from the methodological part of the literature review (see Section 3.1 in book publication) is that, to the author's best knowledge, no Textbook English study published so far has included (as an appendix or supplementary materials) the data and code necessary to reproduce or replicate the published results. As a result, it is very difficult to evaluate the reliability or robustness of the results reported (see also Le Foll 2024).

Though the terms are sometimes used interchangeably and different (at times incompatible) definitions abound, in computational sciences, 'reproducibility' usually refers to the ability to obtain the same results as an original study using the researchers' data and code, whilst 'replicability' refers to obtaining compatible results with the same method but different data (Association for Computing Machinery 2020; see also Berez-Kroeker et al. 2018).

A major barrier to the reproducibility of (corpus) linguistic research is that it is often not possible for copyright or, when participants are involved, data protection reasons to make linguistic data available to the wider public. However, both research practice and the impact of our research can already be greatly improved if we publish our code or, when using GUI software, methods sections detailed enough to be able to successfully replicate the full procedures. This step can enable others to conduct detailed reviews of our methodologies and conceptual replications of our results on different data.

Aside from data protection and copyright regulations, there are, of course, many reasons why researchers may be reluctant to share their data and code (Berez-Kroeker et al. 2018; McManus 2021). It is not within the scope of this monograph to discuss these; however, it is clear that, in many ways, such transparency makes us vulnerable. At the end of the day: to err is human. Yet, the risks involved in committing to Open Science practices are particularly tangible for researchers working on individual projects, like myself, who have had no formal training in data management or programming and have therefore had to learn "on the job". Nonetheless, I am convinced that the advantages outweigh the risks. Striving for transparency helps both the researchers themselves and others reviewing the work to spot and address problems. As a result, the research community can build on both the mishaps and successes of previous research, thus improving the efficiency of research processes and ultimately contributing to advancing scientific progress.

It is with this in mind that I have decided, whenever possible, to publish all the raw data and code necessary to reproduce the results reported in the present monograph following the FAIR principles (i.e., ensuring that research data are Findable, Accessible, Interoperable and Reusable, see Wilkinson et al. 2016). For copyright reasons, the corpora themselves and

annotated corpus data in the form of concordance lines cannot be made available. However, the outcome of both manual and automatic annotation processes is published in tabular formats in the Online Appendix. These tables allow for the reproduction of all the analyses reported on in the following chapters using the reproducible data analysis scripts also published in the [Online Supplements](#) and in the associated Open Science Framework (OSF) repository.

In all chapters of this monograph, full transparency is strived for by reporting on how each sample size was determined and on which grounds data points were excluded, manipulated and/or transformed. Most of these operations were conducted in the open-source programming language and environment R (R Core Team 2022). Most of the data processing and analysis scripts therefore consist of R markdown documents. These were rendered to HTML pages (viewable in the Online Supplements) thus allowing researchers to review the procedures followed without necessarily installing all the required packages and running the code themselves. These scripts also feature additional analyses, tables and plots that were made as part of this study but which, for reasons of space, were not reported on in detail here. Whenever additional software or open-source code from other researchers were used, links to these are also provided in the [Online Supplements](#) (in addition to the corresponding references in the bibliography).

## **Appendix A: Literature review**

This is a [tabular overview](#) of the Textbook English studies that I examined as part of my literature review. It presents the results of a non-exhaustive survey of Textbook English studies published over the past four decades, summarising some of the key information on each study, including its main language focus, methodological approach, information on the textbooks investigated, and, if applicable, on any reference corpora used. Empty cells represent fields that are either not applicable to this particular study or for which no information could be found. Intended as a dynamic resource, this interactive, searchable, and filterable table currently lists over 80 studies on the language content of English L2 textbooks, thereby demonstrating the breadth of Textbook English studies published as of early 2022.

# **Appendix B: Corpus data**

## **Textbook English Corpus (TEC)**

A detailed tabular overview of the composition of the Textbook English Corpus (TEC) together with the full bibliographic metadata is available at [doi.org/10.5281/zenodo.4922819](https://doi.org/10.5281/zenodo.4922819).

Note that, for copyright reasons, the corpus itself cannot be published. If you are interested in using the corpus for non-commercial research purposes and/or in a potential research collaboration, please get in touch with me via [e-mail](#).

## **Reference corpora**

### **Spoken BNC2014**

The original corpus files of the Spoken British National Corpus (BNC) 2014 (Love et al. 2017; Love et al. 2019) can be downloaded for free for research purposes from: <http://corpora.lancs.ac.uk/bnc2014/signup.php>. I used the untagged XML version.

The R script used to pre-process the untagged XML files as explained in Section 4.3.2.2 of the book can be found here: [https://github.com/elenlefoll/TextbookEnglish/blob/main/3\\_Data/BNCspoken\\_nomark-up\\_JackJill.R](https://github.com/elenlefoll/TextbookEnglish/blob/main/3_Data/BNCspoken_nomark-up_JackJill.R)

### **Informative Texts for Teens Corpus (Info Teens)**

For copyright reasons, the corpus itself cannot be made available. Details of its composition can be found in Section 4.3.2.5 of the book. If you are interested in using this corpus for non-commercial research purposes and/or in a potential research collaboration, please get in touch with me via [e-mail](#).

## **Youth Fiction corpus**

For copyright reasons, the corpus itself cannot be made available. The corresponding meta-data can be found here: [https://github.com/elenlefoll/TextbookEnglish/blob/main/3\\_Data/3\\_Youth\\_Fiction\\_Index.csv](https://github.com/elenlefoll/TextbookEnglish/blob/main/3_Data/3_Youth_Fiction_Index.csv). If you are interested in using this corpus for non-commercial research purposes and/or in a potential research collaboration, please get in touch with me via [e-mail](#).

# Appendix D: Data Preparation for the Model of Intra-Textbook Variation

This script documents the steps taken to pre-process the Textbook English Corpus (TEC) data that were entered in the multi-dimensional model of intra-textbook linguistic variation (Chapter 6).

## Packages required

The following packages must be installed and loaded to process the data.

```
#renv::restore() # Restore the project's dependencies from the lockfile to
← ensure that same package versions are used as in the original study

library(caret) # For its confusion matrix function
library(DT) # To display interactive HTML tables
library(here) # For dynamic file paths
library(knitr) # Loaded to display the tables using the kable() function
library(patchwork) # Needed to put together Fig. 1
library(PerformanceAnalytics) # For the correlation plot
library(psych) # For various useful, stats function
library(tidyverse) # For data wrangling
```

## Data import from MFTE output

The raw data used in this script is a tab-separated file that corresponds to the tabular output of mixed normalised frequencies as generated by the [MFTE Perl v. 3.1](#) (Le Foll 2021a).

```
# Read in Textbook Corpus data
TxBcounts <- read.delim(here("data", "MFTE",
← "Tx900MDA_3.1_normed_complex_counts.tsv"), header = TRUE,
← stringsAsFactors = TRUE)
TxBcounts <- TxBcounts |>
```

```
filter(Filename!=".DS_Store") |>
  droplevels()
#str(TxBcounts) # Check sanity of data
#nrow(TxBcounts) # Should be 2014 files
datatable(TxBcounts,
  filter = "top",
) |>
  formatRound(2:ncol(TxBcounts), digits=2)
```

Category	Sub-Category	Parameter	Value	Unit	Description
System A	Processor	Clock Speed	3.2 GHz	GHz	Processor clock speed
System A	Processor	Core Temperature	45°C	°C	Processor core temperature
System A	Processor	Power Consumption	120W	W	Processor power consumption
System A	Processor	Fan Speed	1500 RPM	RPM	Processor fan speed
System A	Memory	RAM Capacity	16GB	GB	RAM capacity
System A	Memory	RAM Speed	3200MHz	MHz	RAM speed
System A	Memory	RAM Latency	16ns	ns	RAM latency
System A	Storage	HDD Capacity	1TB	TB	HDD capacity
System A	Storage	HDD Transfer Rate	5400 RPM	RPM	HDD transfer rate
System A	Storage	Solid State Drive (SSD) Capacity	512GB	GB	SSD capacity
System A	Power Supply	Power Output	300W	W	Power supply output
System A	Power Supply	Efficiency	85%	%	Power supply efficiency
System A	Power Supply	Voltage Regulation	±1%	%	Power supply voltage regulation
System A	Power Supply	Fan Speed	1200 RPM	RPM	Power supply fan speed
System B	Processor	Clock Speed	3.5 GHz	GHz	Processor clock speed
System B	Processor	Core Temperature	48°C	°C	Processor core temperature
System B	Processor	Power Consumption	130W	W	Processor power consumption
System B	Processor	Fan Speed	1600 RPM	RPM	Processor fan speed
System B	Memory	RAM Capacity	16GB	GB	RAM capacity
System B	Memory	RAM Speed	3200MHz	MHz	RAM speed
System B	Memory	RAM Latency	16ns	ns	RAM latency
System B	Storage	HDD Capacity	1TB	TB	HDD capacity
System B	Storage	HDD Transfer Rate	5400 RPM	RPM	HDD transfer rate
System B	Storage	Solid State Drive (SSD) Capacity	512GB	GB	SSD capacity
System B	Power Supply	Power Output	300W	W	Power supply output
System B	Power Supply	Efficiency	85%	%	Power supply efficiency
System B	Power Supply	Voltage Regulation	±1%	%	Power supply voltage regulation
System B	Power Supply	Fan Speed	1200 RPM	RPM	Power supply fan speed

Metadata was added on the basis of the filenames.

```
# Adding a textbook proficiency level
TxBL

- levels <- read.delim(here("data", "metadata",
  "TxB900MDA_ProficiencyLevels.csv"), sep = ",")


TxBC

- ounts <- full_join(TxBcounts, TxBL
  - evels, by = "Filename") |>
  mutate(Level = as.factor(Level)) |>
  mutate(Filename = as.factor(Filename))



# Check distribution and that there are no NAs
summary(TxBcounts$Level) |>
  kable(col.names = c("Textbook Level", "# of texts"))
```

Textbook Level	# of texts
A	292
B	407
C	506
D	478
E	331

```
# Check matching on random sample  
# TxBcounts |>  
#   select(Filename, Level) |>  
#   sample n(20)
```

```

# Adding a register variable from the file names
TxBcounts$Register <- as.factor(stringr::str_extract(TxBcounts$Filename,
  ↵ "Spoken|Narrative|Other|Personal|Informative|Instructional|Poetry")) #
  ↵ Add a variable for Textbook Register
summary(TxBcounts$Register) |>
  kable(col.names = c("Textbook Register", "# of texts"))

```

Textbook Register	# of texts
Informative	364
Instructional	647
Narrative	285
Personal	88
Poetry	37
Spoken	593

```

TxBcounts$Register <- car::recode(TxBcounts$Register, "'Narrative' =
  ↵ 'Fiction'; 'Spoken' = 'Conversation'")
#colnames(TxBcounts) # Check all the variables make sense

# Adding a textbook series variable from the file names
TxBcounts$Filename <- stringr::str_replace(TxBcounts$Filename,
  ↵ "English_In_Mind|English_in_Mind", "EIM")
TxBcounts$Filename <- stringr::str_replace(TxBcounts$Filename,
  ↵ "New_GreenLine", "NGL") # Otherwise the regex for GreenLine will override
  ↵ New_GreenLine
TxBcounts$Filename <- stringr::str_replace(TxBcounts$Filename,
  ↵ "Piece_of_cake", "POC") # Shorten label for ease of plotting
TxBcounts$Series <- as.factor(stringr::str_extract(TxBcounts$Filename,
  ↵ "Access|Achievers|EIM|GreenLine|HT|NB|NM|POC|JTT|NGL|Solutions")) #
  ↵ Extract textbook series from (ammended) filenames
summary(TxBcounts$Series) |>
  kable(col.names = c("Textbook Name", "# of texts"))

```

Textbook Name	# of texts
Access	315
Achievers	240
EIM	180

Textbook Name	# of texts
GreenLine	209
HT	115
JTT	129
NB	44
NGL	298
NM	59
POC	98
Solutions	327

```
# Including the French textbooks for the first year of Lycée to their
# corresponding publisher series from collège
TxBcounts$Series <- car::recode(TxBcounts$Series, "c('NB', 'JTT') = 'JTT'";
                                 c('NM', 'HT') = 'HT'") # Recode final volumes of French series (see
# Section 4.3.1.1 on textbook selection for details)
summary(TxBcounts$Series) |>
  kable(col.names = c("Textbook Series", "# of texts"))
```

Textbook Series	# of texts
Access	315
Achievers	240
EIM	180
GreenLine	209
HT	174
JTT	173
NGL	298
POC	98
Solutions	327

```
# Adding a textbook country of use variable from the series variable
TxBcounts$Country <- TxBcounts$Series
TxBcounts$Country <- car::recode(TxBcounts$Series, "c('Access', 'GreenLine',
  'NGL') = 'Germany'; c('Achievers', 'EIM', 'Solutions') = 'Spain'; c('HT',
  'NB', 'NM', 'POC', 'JTT') = 'France'")"
summary(TxBcounts$Country) |>
  kable(col.names = c("Country of Use", "# of texts"))
```

Country of Use	# of texts
France	445
Germany	822
Spain	747

```
# Re-order variables
#colnames(TxBcounts)
TxBcounts <- select(TxBcounts, order(names(TxBcounts))) %>%
  select(Filename, Country, Series, Level, Register, Words, everything())
#colnames(TxBcounts)
```

## Corpus size

This table provides some summary statistics about the number of words included in the TEC texts originally tagged for this study.

```
TxBcounts |>
  group_by(Register) |>
  summarise(totaltexts = n(), totalwords = sum(Words), mean =
    as.integer(mean(Words)), sd = as.integer(sd(Words)), TTRmean =
    mean(TTR)) |>
  kable(digits = 2, format.args = list(big.mark = ","))
```

Register	totaltexts	totalwords	mean	sd	TTRmean
Conversation	593	505,147	851	301	0.44
Fiction	285	241,512	847	208	0.47
Informative	364	304,695	837	177	0.51
Instructional	647	585,049	904	94	0.42
Personal	88	69,570	790	177	0.48
Poetry	37	26,445	714	192	0.44

```
#TxBcounts <- saveRDS(TxBcounts, here("data", "processed", "TxBcounts.rds"))
```

## Data preparation for PCA

Poetry texts were removed for this analysis as there were too few compared to the other register categories.

```
summary(TxBcounts$Register) |>
  kable(col.names = c("Register", "# texts"))
```

Register	# texts
Conversation	593
Fiction	285
Informative	364
Instructional	647
Personal	88
Poetry	37

This led to the following distribution of texts across the five textbook English registers examined in the model of intra-textbook linguistic variation:

```
TxBcounts <- TxBcounts |>
  filter(Register!="Poetry") |>
  droplevels()

summary(TxBcounts$Register) |>
  kable(col.names = c("Register", "# texts"))
```

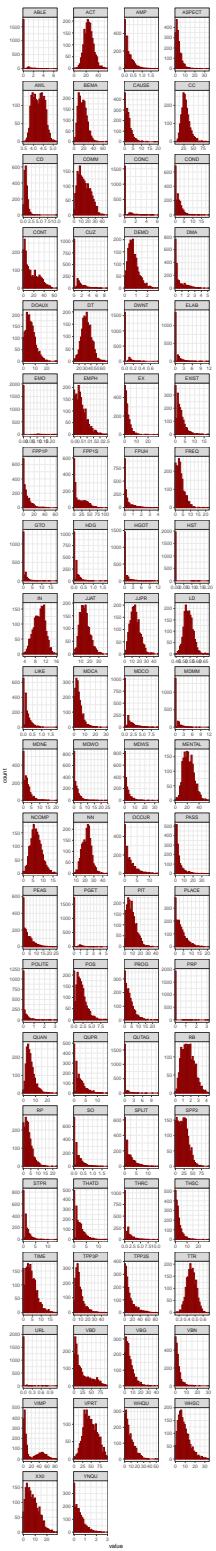
Register	# texts
Conversation	593
Fiction	285
Informative	364
Instructional	647
Personal	88

## Feature distributions

The distributions of each linguistic features were examined by means of visualisation. As shown below, before transformation, many of the features displayed highly skewed distributions.

```
TxBcounts |>
  select(-Words) |>
  keep(is.numeric) |>
  tidyrr::gather() |> # This function from tidyrr converts a selection of
  ↵ variables into two variables: a key and a value. The key contains the
  ↵ names of the original variable and the value the data. This means we can
  ↵ then use the facet_wrap function from ggplot2
```

```
ggplot(aes(value)) +
  theme_bw() +
  facet_wrap(~ key, scales = "free", ncol = 4) +
  scale_x_continuous(expand=c(0,0)) +
  geom_histogram(bins = 30, colour= "darkred", fill = "darkred", alpha =
    0.5)
```



```
#ggsave(here("plots", "TEC-HistogramPlotsAllVariablesTEC-only.svg"), width =
  ↵  20, height = 45)
```

## Feature removal

A number of features were removed from the dataset as they are not linguistically interpretable. In the case of the TEC, this included the variable CD because numbers spellt out as digits were removed from the textbooks before these were tagged with the MFTE. In addition, the variables LIKE and SO because these are “bin” features included in the output of the MFTE to ensure that the counts for these polysemous words do not inflate other categories due to mistags (Le Foll 2021b).

Whenever linguistically meaningful, very low-frequency features were merged. Finally, features absent from more than third of texts were also excluded. For the analysis intra-textbook register variation, the following linguistic features were excluded from the analysis due to low dispersion:

```
# Removal of meaningless features:
TxBcounts <- TxBcounts |>
  select(-c(CD, LIKE, SO))

# Function to compute percentage of texts with occurrences meeting a
  ↵ condition
compute_percentage <- function(data, condition, threshold) {
  numeric_data <- Filter(is.numeric, data)
  percentage <- round(colSums(condition[, sapply(numeric_data,
  ↵ is.numeric)])/nrow(data) * 100, 2)
  percentage <- as.data.frame(percentage)
  colnames(percentage) <- "Percentage"
  percentage <- percentage |>
    filter(!is.na(Percentage)) |>
    rownames_to_column() |>
    arrange(Percentage)
  if (!missing(threshold)) {
    percentage <- percentage |>
      filter(Percentage > threshold)
  }
  return(percentage)
}

# Calculate percentage of texts with 0 occurrences of each feature
```

```

zero_features <- compute_percentage(TxBcounts, TxBcounts == 0, 66.6)
# zero_features |>
#   kable(col.names = c("Feature", "% texts with zero occurrences"))

# Combine low frequency features into meaningful groups whenever this makes
#   ↵ linguistic sense
TxBcounts <- TxBcounts |>
  mutate(JJPR = ABLE + JJPR, ABLE = NULL) |>
  mutate(PASS = PGET + PASS, PGET = NULL)

# Re-calculate percentage of texts with 0 occurrences of each feature
zero_features2 <- compute_percentage(TxBcounts, TxBcounts == 0, 66.6)
zero_features2 |>
  kable(col.names = c("Feature", "% texts with zero occurrences"))

```

Feature	% texts with zero occurrences
GTO	67.07
ELAB	69.30
MDMM	70.81
HGOT	73.75
CONC	80.48
DWNT	81.44
QUTAG	85.99
URL	96.51
EMO	97.82
PRP	98.33
HST	99.44

```

# Drop variables with low document frequency
TxBcounts <- select(TxBcounts, -one_of(zero_features2$rownames))
#ncol(TxBcounts)-8 # Number of linguistic features remaining

# List of features
#colnames(TxBcounts)

```

These feature removal operations resulted in a feature set of 64 linguistic variables.

## Identifying potential outlier texts

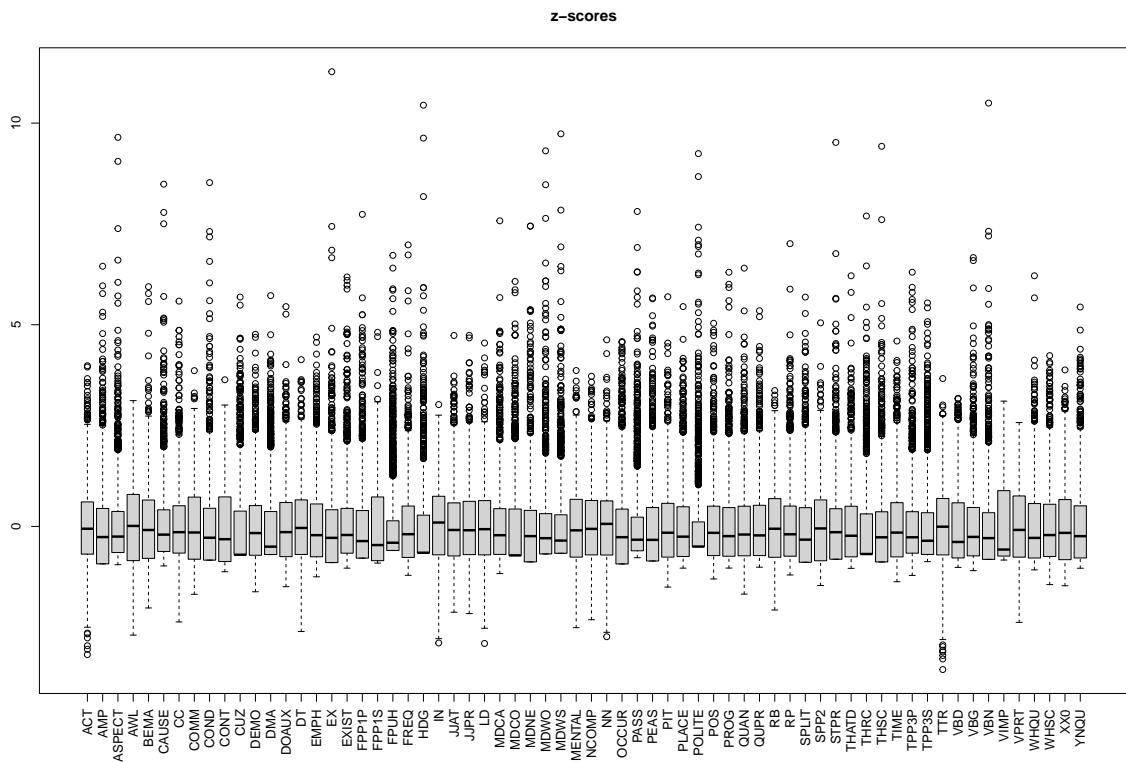
All normalised frequencies were normalised to identify any potential outlier texts.

```

# First scale the normalised counts (z-standardisation) to be able to compare
# the various features
TxBzcounts <- TxBcounts |>
  select(-Words) |>
  keep(is.numeric) |>
  scale()

boxplot(TxBzcounts, las = 3, main = "z-scores") # Slow to open!

```



```

# If necessary, remove any outliers at this stage.
TxBdata <- cbind(TxBcounts[,1:6], as.data.frame(TxBzcounts))

outliers <- TxBdata |>
  select(-c(Words, LD, TTR)) |>
  filter(if_any(where(is.numeric), ~ .x > 8)) |>
  select(Filename)

```

The following outlier texts were identified and excluded in subsequent analyses.

```
outliers
```

```
                                Filename
1                  POC_4e_Spoken_0007.txt
2      Solutions_Elementary_Personal_0001.txt
3                  NGL_5_Instructional_0018.txt
4                  Access_1_Spoken_0011.txt
5                  EIM_1_Spoken_0012.txt
6                  NGL_4_Spoken_0011.txt
7      Solutions_Intermediate_Plus_Personal_0001.txt
8      Solutions_Elementary_ELF_Spoken_0021.txt
9                  NB_2_Informative_0009.txt
10     Solutions_Intermediate_Plus_Spoken_0022.txt
11     Solutions_Intermediate_Instructional_0025.txt
12 Solutions_Pre-Intermediate_Instructional_0024.txt
13                  POC_4e_Spoken_0010.txt
14     Solutions_Intermediate_Spoken_0019.txt
15                  Access_1_Spoken_0019.txt
16 Solutions_Pre-Intermediate_ELF_Spoken_0005.txt
```

```
TxBcounts <- TxBcounts |>
  filter(!Filename %in% outliers$Filename)

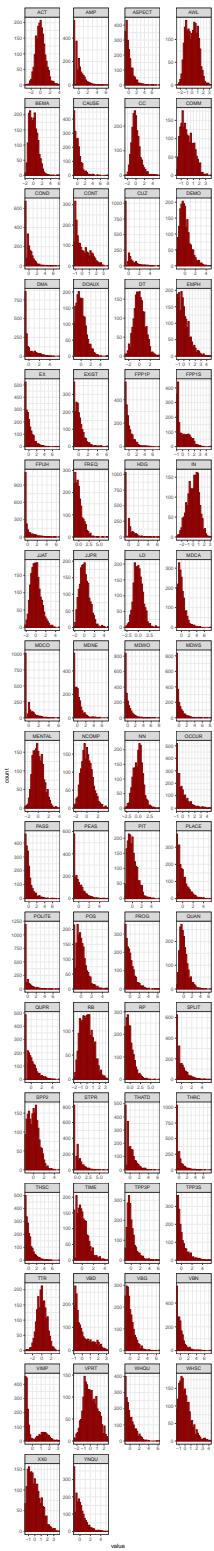
#saveRDS(TxBcounts, here("data", "processed", "TxBcounts3.rds")) # Last saved
  ↵ 6 March 2024

TxBzcounts <- TxBcounts |>
  select(-Words) |>
  keep(is.numeric) |>
  scale()
```

This resulted in 1,961 TEC texts being included in the model of intra-textbook linguistic variation with the following normalised feature distributions.

```
TxBzcounts |>
  as.data.frame() |>
  gather() |> # This function from tidyverse converts a selection of variables
  ↵  into two variables: a key and a value. The key contains the names of
  ↵  the original variable and the value the data. This means we can then
  ↵  use the facet_wrap function from ggplot2
  ggplot(aes(value)) +
```

```
theme_bw() +  
facet_wrap(~ key, scales = "free", ncol = 4) +  
scale_x_continuous(expand=c(0,0)) +  
geom_histogram(bins = 30, colour= "darkred", fill = "darkred", alpha =  
 0.5)
```



```
#ggsave(here("plots", "TEC-zscores-HistogramsAllVariablesTEC-only.svg"),
  width = 20, height = 45)
```

## Signed log transformation

A signed logarithmic transformation was applied to (further) deskew the feature distributions (Diwersy, Evert, and Neumann 2014; Neumann and Evert 2021).

The signed log transformation function was inspired by the SignedLog function proposed in <https://cran.r-project.org/web/packages/DataVisualizations/DataVisualizations.pdf>

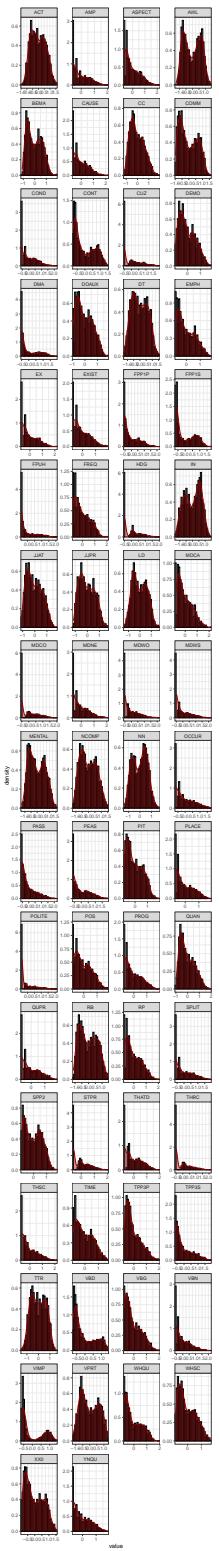
```
# All features are signed log-transformed (note that this is also what
  ↵  Neumann & Evert 2021 propose)
signed.log <- function(x) {
  sign(x) * log(abs(x) + 1)
}

TxBzlogcounts <- signed.log(TxBzcounts) # Standardise first, then signed log
  ↵  transform

#saveRDS(TxBzlogcounts, here("data", "processed", "TxBzlogcounts.rds")) #
  ↵  Last saved 6 March 2024
```

The new feature distributions are visualised below.

```
TxBzlogcounts |>
  as.data.frame() |>
  gather() |> # This function from tidyverse converts a selection of variables
    ↵  into two variables: a key and a value. The key contains the names of
    ↵  the original variable and the value the data. This means we can then
    ↵  use the facet_wrap function from ggplot2
  ggplot(aes(value, after_stat(density))) +
  theme_bw() +
  facet_wrap(~ key, scales = "free", ncol = 4) +
  scale_x_continuous(expand=c(0,0)) +
  scale_y_continuous(limits = c(0,NA)) +
  geom_histogram(bins = 30, colour= "black", fill = "grey") +
  geom_density(colour = "darkred", weight = 2, fill="darkred", alpha = .4)
```



```
#ggsave(here("plots", "DensityPlotsAllVariablesSignedLog-TEC-only.svg"),
  width = 15, height = 49)
```

The following correlation plots serve to illustrate the effect of the variable transformations performed in the above chunks.

Example feature distributions before transformations:

```
# This is a slightly amended version of the
# PerformanceAnalytics::chart.Correlation() function. It simply removes the
# significance stars that are meaningless with this many data points (see
# commented out lines below)

chart.Correlation.nostars <- function (R, histogram = TRUE, method =
  c("pearson", "kendall", "spearman"), ...) {
  x = checkData(R, method = "matrix")
  if (missing(method))
    method = method[1]
  panel.cor <- function(x, y, digits = 2, prefix = "", use =
    "pairwise.complete.obs", method = "pearson", cex.cor, ...) {
    usr <- par("usr")
    on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- cor(x, y, use = use, method = method)
    txt <- format(c(r, 0.123456789), digits = digits)[1]
    txt <- paste(prefix, txt, sep = "")
    if (missing(cex.cor))
      cex <- 0.8/strwidth(txt)
    test <- cor.test(as.numeric(x), as.numeric(y), method = method)
    # Signif <- symnum(test$p.value, corr = FALSE, na = FALSE,
    #                   cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1), symbols =
    #                   c("***",
    #                   **", "*",
    #                   ".",
    #                   ""))
    text(0.5, 0.5, txt, cex = cex * (abs(r) + 0.3)/1.3)
    # text(0.8, 0.8, Signif, cex = cex, col = 2)
  }
  f <- function(t) {
    dnorm(t, mean = mean(x), sd = sd.xts(x))
  }
  dotargs <- list(...)
  dotargs$method <- NULL
```

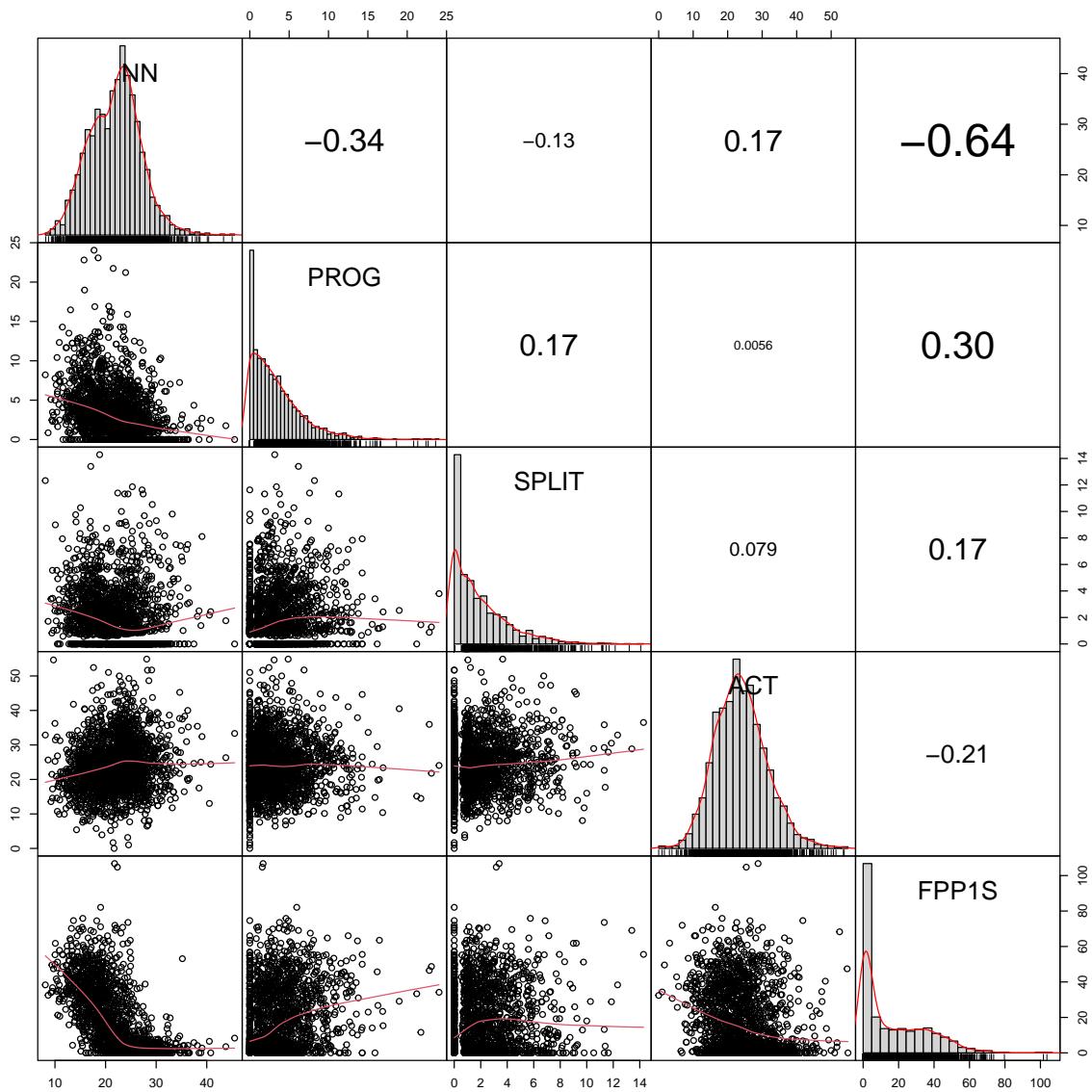
```

rm(method)
hist.panel = function(x, ... = NULL) {
  par(new = TRUE)
  hist(x, col = "light gray", probability = TRUE,
    axes = FALSE, main = "", breaks = "FD")
  lines(density(x, na.rm = TRUE), col = "red", lwd = 1)
  rug(x)
}
if (histogram)
  pairs(x, gap = 0, lower.panel = panel.smooth, upper.panel = panel.cor,
    diag.panel = hist.panel)
else pairs(x, gap = 0, lower.panel = panel.smooth, upper.panel = panel.cor)
}

# Example plot without any variable transformation
example1 <- TxBcounts |>
  select(NN,PROG,SPLIT,ACT,FPP1S)

#png(here("plots", "CorrChart-TEC-examples-normedcounts.png"), width = 20,
#  height = 20, units = "cm", res = 300)
chart.Correlation.nostars(example1, histogram=TRUE, pch=19)

```

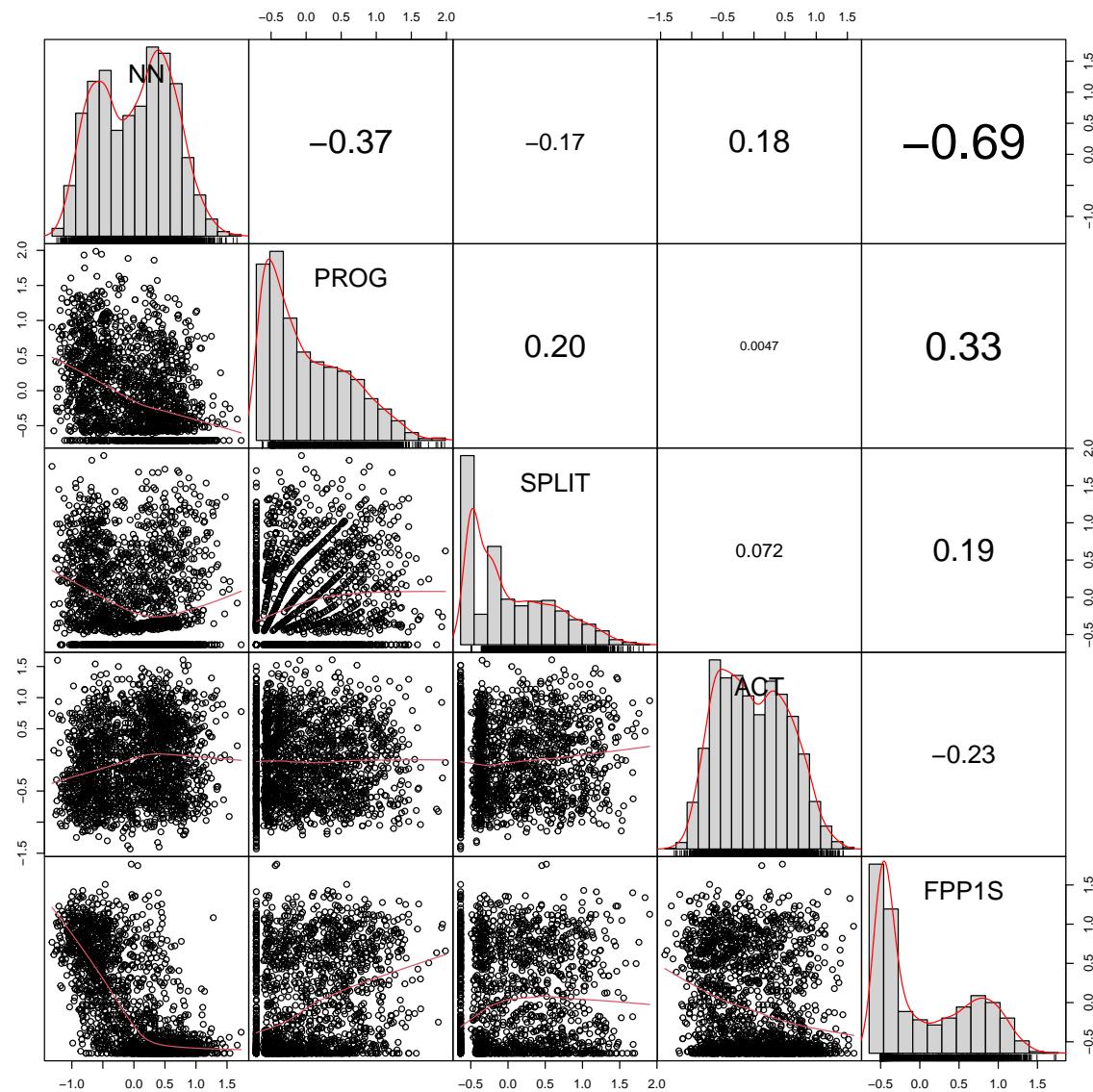


```
#dev.off()
```

Example feature distributions after transformations:

```
# Example plot with transformed variables
example2 <- TxBzlogcounts |>
  as.data.frame() |>
  select(NN,PROG,SPLIT,ACT,FPP1S)
```

```
#png(here("plots", "CorrChart-TEC-examples-zsignedlogcounts.png"), width =
  ↵ 20, height = 20, units = "cm", res = 300)
chart.Correlation.nostars(example2, histogram=TRUE, pch=19)
```



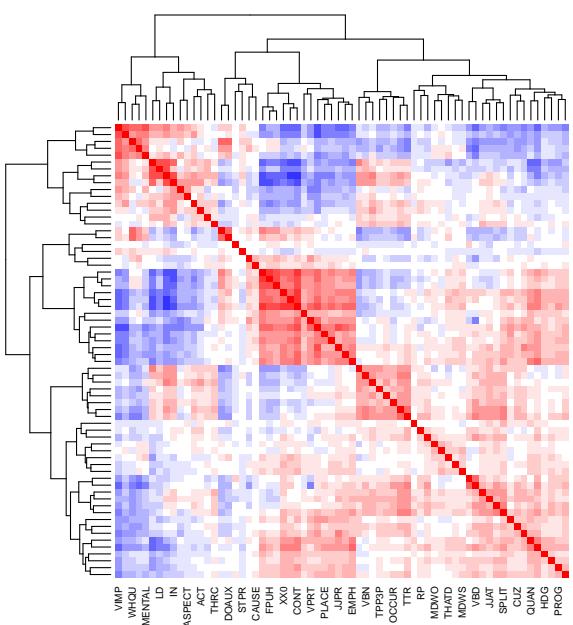
```
#dev.off()
```

## Feature correlations

The correlations of the transformed feature frequencies can be visualised in the form of a heatmap. Negative correlations are rendered in blue, whereas positive ones are in red.

```
# Simple heatmap in base R (inspired by Stephanie Evert's SIGIL code)
cor.colours <- c(
  hsv(h=2/3, v=1, s=(10:1)/10), # blue = negative correlation
  rgb(1,1,1), # white = no correlation
  hsv(h=0, v=1, s=(1:10/10))) # red = positive correlation

#png(here("plots", "heatmapzlogcounts-TEC-only.png"), width = 30, height= 30,
#  units = "cm", res = 300)
heatmap(cor(TxBzlogcounts),
        symm=TRUE,
        zlim=c(-1,1),
        col=cor.colours,
        margins=c(0,0))
```



```

#dev.off()

# Calculate the sum of all the words in the tagged texts of the TEC
totalwords <- TxBcounts |>
  select(Words) |>
  sum() |>
  format(big.mark=",")

```

## Composition of TEC texts/files

These figures and tables provide summary statistics on the texts/files of the TEC that were entered in the multi-dimensional model of intra-textbook linguistic variation. In total, the TEC texts entered amounted to 1,693,650 words.

```

metadata <- TxBcounts |>
  select(Filename, Country, Series, Level, Register, Words) |>
  mutate(Volume = paste(Series, Level)) |>
  mutate(Volume = fct_rev(Volume)) |>
  mutate(Volume = fct_reorder(Volume, as.numeric(Level))) |>
  group_by(Volume) |>
  mutate(wordcount = sum(Words)) |>
  ungroup() |>
  distinct(Volume, .keep_all = TRUE)

# Plot for book
metadata2 <- TxBcounts |>
  select(Country, Series, Level, Register, Words) |>
  mutate(Volume = paste(Series, Level)) |>
  mutate(Volume = fct_rev(Volume)) |>
  #mutate(Volume = fct_reorder(Volume, as.numeric(Level))) |>
  group_by(Volume, Register) |>
  mutate(wordcount = sum(Words)) |>
  ungroup() |>
  distinct(Volume, Register, .keep_all = TRUE)

# This is the palette created above on the basis of the suffrager pakcage
# (but without needed to install the package)
palette <- c("#BD241E", "#A18A33", "#15274D", "#D54E1E", "#EA7E1E",
             "#4C4C4C", "#722672", "#F9B921", "#267226")

```

```

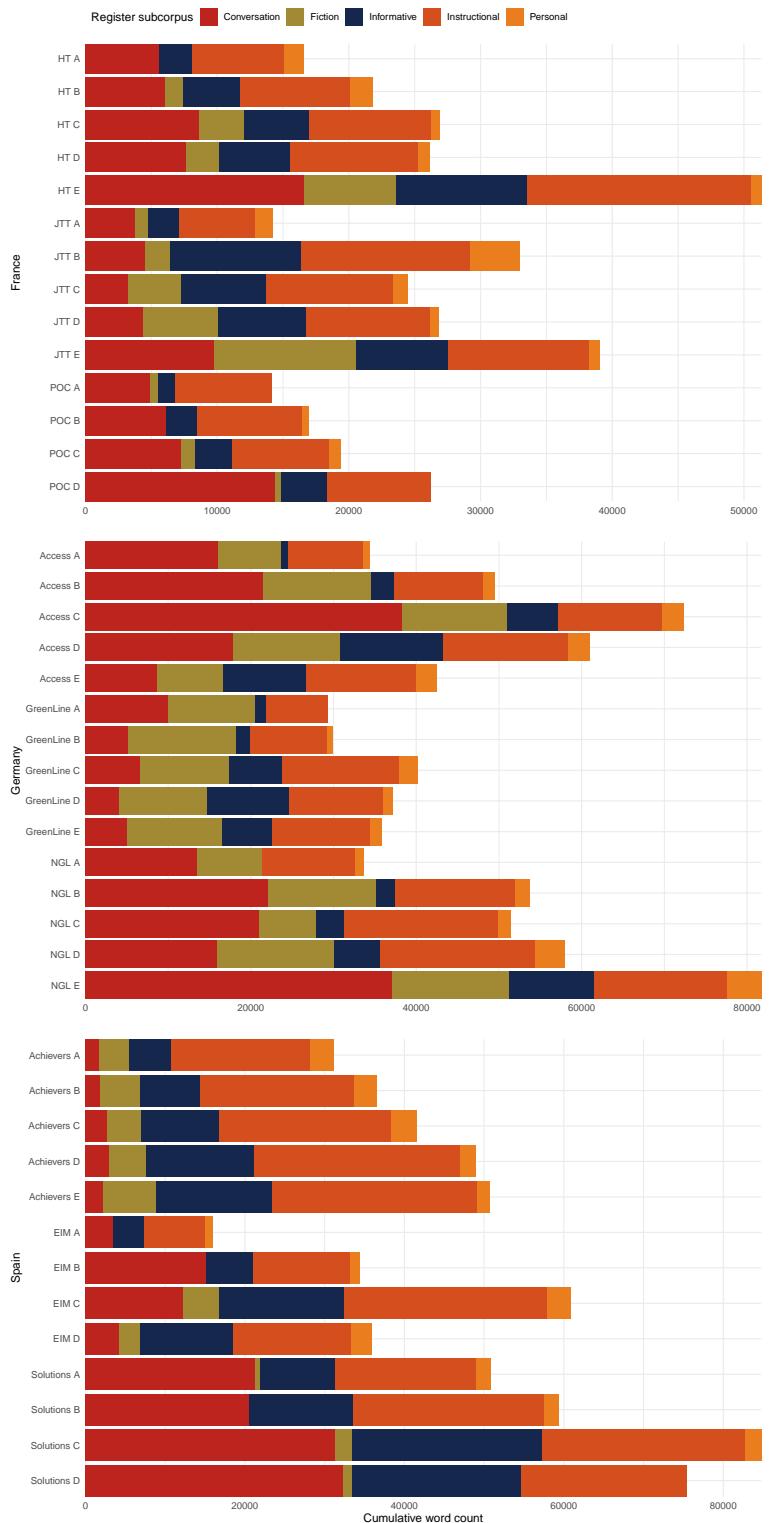
PlotSp <- metadata2 |>
  filter(Country=="Spain") |>
  #arrange(Volume) |>
  ggplot(aes(x = Volume, y = wordcount, fill = fct_rev(Register))) +
    geom_bar(stat = "identity", position = "stack") +
    coord_flip(expand = FALSE) + # Removes those annoying ticks before each
    ↵ bar label
    theme_minimal() + theme(legend.position = "none") +
    labs(x = "Spain", y = "Cumulative word count") +
    scale_fill_manual(values = palette[c(5,4,3,2,1)],
                      guide = guide_legend(reverse = TRUE))

PlotGer <- metadata2 |>
  filter(Country=="Germany") |>
  #arrange(Volume) |>
  ggplot(aes(x = Volume, y = wordcount, fill = fct_rev(Register))) +
    geom_bar(stat = "identity", position = "stack") +
    coord_flip(expand = FALSE) +
    labs(x = "Germany", y = "") +
    scale_fill_manual(values = palette[c(5,4,3,2,1)], guide =
    ↵ guide_legend(reverse = TRUE)) +
    theme_minimal() + theme(legend.position = "none")

PlotFr <- metadata2 |>
  filter(Country=="France") |>
  #arrange(Volume) |>
  ggplot(aes(x = Volume, y = wordcount, fill = fct_rev(Register))) +
    geom_bar(stat = "identity", position = "stack") +
    coord_flip(expand = FALSE) +
    labs(x = "France", y = "", fill = "Register subcorpus") +
    scale_fill_manual(values = palette[c(5,4,3,2,1)], guide =
    ↵ guide_legend(reverse = TRUE, legend.hjust = 0)) +
    theme_minimal() + theme(legend.position = "top", legend.justification =
    ↵ "left")

PlotFr /
PlotGer /
PlotSp

```



```
#ggsave(here("plots", "TEC-T_wordcounts_book.svg"), width = 8, height = 12)
```

The following table provides information about the proportion of instructional language featured in each textbook series.

```
metadataInstr <- TxBcounts |>
  select(Country, Series, Level, Register, Words) |>
  filter(Register=="Instructional") |>
  mutate(Volume = paste(Series, Register)) |>
  mutate(Volume = fct_rev(Volume)) |>
  mutate(Volume = fct_reorder(Volume, as.numeric(Level))) |>
  group_by(Volume, Register) |>
  mutate(InstrWordcount = sum(Words)) |>
  ungroup() |>
  distinct(Volume, .keep_all = TRUE) |>
  select(Series, InstrWordcount)

metaWordcount <- TxBcounts |>
  select(Country, Series, Level, Register, Words) |>
  group_by(Series) |>
  mutate(TECwordcount = sum(Words)) |>
  ungroup() |>
  distinct(Series, .keep_all = TRUE) |>
  select(Series, TECwordcount)

wordcount <- merge(metaWordcount, metadataInstr, by = "Series")

wordcount |>
  mutate(InstrucPercent = InstrWordcount/TECwordcount*100) |>
  arrange(InstrucPercent) |>
  mutate(InstrucPercent = round(InstrucPercent, 2)) |>
  kable(col.names = c("Textbook Series", "Total words", "Instructional
  ↵ words", "% of textbook content"),
        digits = 2,
        format.args = list(big.mark = ","))
```

Textbook Series	Total words	Instructional words	% of textbook content
Access	259,679	60,938	23.47
NGL	278,316	79,312	28.50
GreenLine	172,267	54,263	31.50

Textbook Series	Total words	Instructional words	% of textbook content
Solutions	270,278	87,829	32.50
JTT	137,557	48,375	35.17
HT	142,676	51,550	36.13
POC	76,714	30,548	39.82
EIM	147,185	59,928	40.72
Achievers	208,978	109,886	52.58

# Appendix E: Data Analysis for the Model of Intra-Textbook Variation

This script documents the analysis of the pre-processed data from the Textbook English Corpus (TEC) to arrive at the multi-dimensional model of intra-textbook linguistic variation (Chapter 6). It generates all of the statistics and plots included in the book, as well as many others that were used in the analysis, but not included in the book for reasons of space.

## Packages required

The following packages must be installed and loaded to carry out the following analyses.

```
#renv::restore() # Restore the project's dependencies from the lockfile to
#                ensure that same package versions are used as in the original study

library(caret) # For its confusion matrix function
library(cowplot)
library(DescTools) # For 95% CI
library(emmeans)
library(factoextra) # For circular graphs of variables
library(forcats) # For data manipulation
library(ggthemes) # For theme of factoextra plots
library(here) # For dynamic file paths
library(knitr) # Loaded to display the tables using the kable() function
library(lme4) # For linear regression modelling
library(patchwork) # To create figures with more than one plot
library(PCAtools) # For nice biplots of PCA results
library(psych) # For various useful stats function
library(sjPlot) # For model plots and tables
library(tidyverse) # For data wrangling
library(visreg) # For plots of interaction effects

# From https://github.com/RainCloudPlots/RainCloudPlots:
source(here("R_rainclouds.R")) # For geom_flat_violin rainplots
```

## Preparing the data for PCA

### TEC data import

```
TxBcounts <- readRDS(here("data", "processed", "TxBcounts3.rds"))
# colnames(TxBcounts)
# nrow(TxBcounts)

TxBzlogcounts <- readRDS(here("data", "processed", "TxBzlogcounts.rds"))
# nrow(TxBzlogcounts)
# colnames(TxBzlogcounts)

TxBdata <- cbind(TxBcounts[,1:6], as.data.frame(TxBzlogcounts))
# str(TxBdata)
```

First, the TEC data as processed in Appendix D is imported. It consists of 1,961 texts/files, each with logged standardised normalised frequencies for 66 linguistic features.

### Checking the factorability of data

```
kmo <- KMO(TxBdata[,7:ncol(TxBdata)])
```

The overall MSA value of the dataset is 0.86. The features have the following individual MSA values (ordered from lowest to largest):

```
kmo$MSAi[order(kmo$MSAi)] |> round(2)
```

MDWO	MDWS	MDNE	MDCA	VBD	VPRT	POS	ACT	FREQ	TPP3S	LD
0.34	0.46	0.52	0.53	0.59	0.60	0.64	0.65	0.65	0.66	0.68
CAUSE	COND	MDCO	VIMP	NCOMP	DT	TPP3P	STPR	RP	SPP2	MENTAL
0.69	0.75	0.77	0.78	0.79	0.80	0.80	0.81	0.81	0.83	0.84
DOAUX	WHSC	VBG	EXIST	THATD	COMM	FPP1S	IN	NN	WHQU	JJAT
0.84	0.85	0.86	0.86	0.86	0.87	0.87	0.88	0.88	0.89	0.89
DEMO	THRC	ASPECT	CC	EX	OCCUR	PEAS	TTR	YNQU	AWL	QUAN
0.89	0.89	0.90	0.90	0.90	0.90	0.91	0.91	0.91	0.92	0.92
FPP1P	PROG	XX0	CONT	TIME	BEMA	SPLIT	PASS	JJPR	AMP	QUPR
0.92	0.92	0.92	0.92	0.93	0.93	0.93	0.94	0.94	0.95	0.95

THSC	RB	FPUH	CUZ	VBN	PIT	DMA	POLITE	EMPH	HDG	PLACE
0.95	0.95	0.95	0.95	0.95	0.96	0.96	0.96	0.96	0.96	0.97

### Removal of feature with MSAs of < 0.5

We first remove the first feature with an individual MSA < 0.5, then check the MSA values again and continue removing features one by one if necessary.

```
TxBdata <- TxBdata |>
  select(-c(MDW0))

kmo2 <- KMO(TxBdata[, 7:ncol(TxBdata)])
```

The overall MSA value of the dataset is now 0.87. None of the remaining features have individual MSA values below 0.5:

```
kmo2$MSAi [order(kmo2$MSAi)] |> round(2)
```

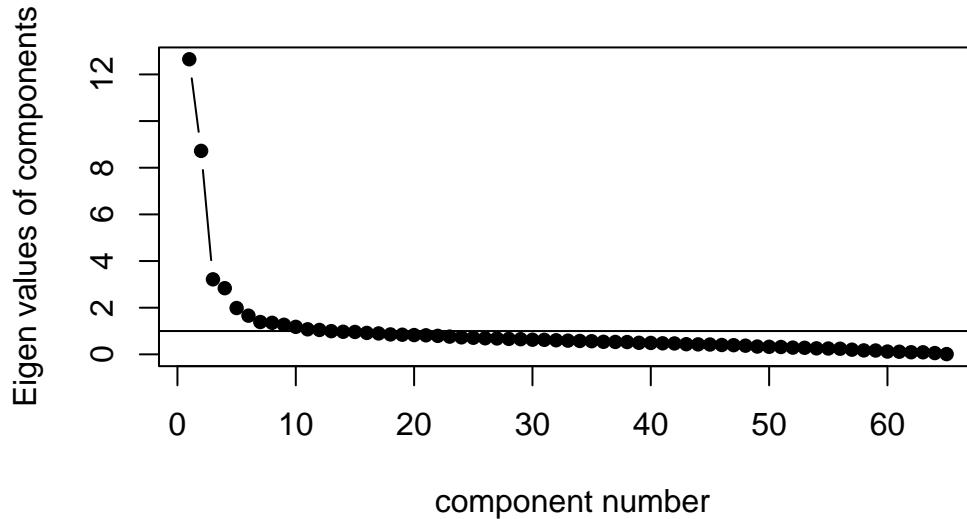
MDWS	MDNE	MDCA	VBD	POS	VPRT	FREQ	ACT	TPP3S	LD	CAUSE
0.55	0.58	0.61	0.63	0.64	0.65	0.65	0.66	0.66	0.69	0.70
MDC0	COND	DT	TPP3P	VIMP	NCOMP	RP	STPR	SPP2	DOAUX	MENTAL
0.77	0.80	0.80	0.80	0.81	0.81	0.81	0.82	0.83	0.84	0.85
VBG	WHSC	EXIST	THATD	FPP1S	COMM	IN	NN	DEMO	WHQU	THRC
0.85	0.85	0.86	0.87	0.87	0.87	0.88	0.89	0.89	0.89	0.89
JJAT	ASPECT	PEAS	EX	OCCUR	CC	TTR	YNQU	AWL	QUAN	FPP1P
0.89	0.89	0.90	0.90	0.91	0.91	0.91	0.91	0.92	0.92	0.92
TIME	XX0	CONT	PROG	BEMA	SPLIT	PASS	JJPR	THSC	AMP	RB
0.92	0.92	0.93	0.93	0.93	0.93	0.94	0.94	0.95	0.95	0.95
QUPR	FPUH	PIT	VBN	DMA	CUZ	POLITE	EMPH	HDG	PLACE	
0.95	0.95	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.97	

### Choosing the number of principal components to retain

On the basis of this scree plot, six principal components were initially retained.

```
# Plot screen plot
#png(here("plots", "screeplot-TEC-only.png"), width = 20, height= 12, units =
#  "cm", res = 300)
scree(TxBdata[, 7:ncol(TxBdata)], factors = FALSE, pc = TRUE) # Retain six
# components
```

## Scree plot



```
#dev.off()

# Perform PCA
pca1 <- psych::principal(TxBdata[, 7:ncol(TxBdata)],
                           nfactors = 6,
                           rotate = "none")
#pca1$loadings
```

### Excluding features with low final communalities

We first check whether some feature have extremely low communalities (see <https://rdrr.io/cran/FactorAssumpt/>)

	STPR	MDNE	HDG	CAUSE	FREQ	THRC	POS	PROG	ACT	DEMO	MDWS
	0.09	0.17	0.17	0.19	0.22	0.22	0.24	0.26	0.26	0.27	0.28
CUZ	COND	QUPR	EXIST	MDC0	NCOMP	OCCUR	TIME	ASPECT	TPP3P	AMP	
	0.28	0.28	0.29	0.29	0.31	0.31	0.32	0.32	0.33	0.33	0.34
RP	THATD	THSC	EX	FPP1P	PLACE	PIT	VBG	PEAS	MDCA	DOAUX	
	0.34	0.36	0.39	0.43	0.43	0.43	0.45	0.46	0.47	0.48	0.48
VBN	JJPR	JJAT	WHSC	SPLIT	EMPH	QUAN	MENTAL	TPP3S	PASS	YNQU	
	0.48	0.49	0.49	0.50	0.51	0.53	0.55	0.56	0.56	0.57	0.58
POLITE	RB	CC	XX0	DT	COMM	WHQU	TTR	FPP1S	IN	LD	

0.58	0.58	0.58	0.59	0.62	0.62	0.64	0.65	0.67	0.68	0.68
FPUH	VPRT	SPP2	BEMA	DMA	VBD	AWL	CONT	NN	VIMP	
0.70	0.71	0.72	0.74	0.74	0.80	0.85	0.85	0.87	0.90	

As we chose to exclude features with communalities of < 0.2, we remove STPR, HDG, MDNE and CAUSE from the dataset to be analysed.

```
TxBdataforPCA <- TxBdata |>
  select(-c(STPR, MDNE, HDG, CAUSE))
```

The overall MSA value of the dataset is now 0.88. None of the remaining features have individual MSA values below 0.5:

```
kmo3$MSAi [order(kmo3$MSAi)] |>round(2)
```

MDWS	MDCA	POS	FREQ	VBD	TPP3S	VPRT	ACT	LD	COND	DT
0.54	0.64	0.64	0.65	0.65	0.66	0.66	0.67	0.69	0.78	0.79
MDCO	TPP3P	RP	NCOMP	VIMP	SPP2	DOAUX	MENTAL	WHSC	VBG	THATD
0.79	0.81	0.82	0.82	0.82	0.82	0.84	0.85	0.86	0.86	0.86
EXIST	FPP1S	COMM	NN	IN	WHQU	DEMO	ASPECT	JJAT	THRC	EX
0.86	0.87	0.88	0.89	0.89	0.89	0.89	0.89	0.89	0.90	0.90
OCCUR	PEAS	CC	YNQU	QUAN	AWL	TIME	XX0	FPP1P	TTR	CONT
0.90	0.91	0.91	0.91	0.91	0.92	0.92	0.92	0.92	0.92	0.93
PROG	BEMA	SPLIT	PASS	JJPR	THSC	RB	QUPR	AMP	FPUH	PIT
0.93	0.93	0.93	0.94	0.94	0.95	0.95	0.95	0.95	0.95	0.95
VBN	DMA	POLITE	CUZ	EMPH	PLACE					
0.95	0.96	0.96	0.96	0.96	0.97					

The final number of linguistic features entered in the intra-textbook model of linguistic variation is 61.

## Testing the effect of rotating the components

This chunk was used when considering whether or not to rotate the components (see methods section). Ultimately, the components were not rotated.

```

# Comparing a rotated vs. a non-rotated solution

#TxBdata <- readRDS(here("data", "processed", "TxBdataforPCA.rds"))

# No rotation
pca2 <- psych::principal(TxBdata[,7:ncol(TxBdata)],
                           nfactors = 6,
                           rotate = "none")

pca2$loadings

biplot.psych(pca2,
              vars = TRUE,
              choose=c(1,2),
              )

# Promax rotation
pca2.rotated <- psych::principal(TxBdata[,7:ncol(TxBdata)],
                                   nfactors = 6,
                                   rotate = "promax")

# This summary shows the component correlations which is particularly
→ interesting
pca2.rotated

pca2.rotated$loadings

biplot.psych(pca2.rotated, vars = TRUE, choose=c(1,2))

```

## Principal Component Analysis (PCA)

### Using the full dataset

Except outliers removed as part of the data preparation (see Appendix D).

```

# Perform PCA on full data
TxBdata <- readRDS(here("data", "processed", "TxBdataforPCA.rds"))

```

## Using random subsets of the data

Alternatively, it is possible to conduct the PCA on random subsets of the data to test the stability of the solution. Re-running this line will generate a new subset of the TEC texts containing 2/3 of the texts randomly sampled.

```
TxBdata <- readRDS(here("data", "processed", "TxBdataforPCA.rds")) |>
  slice_sample(n = round(1961*0.6), replace = FALSE)

nrow(TxBdata)
TxBdata$Filename[1:10]
nrow(TxBdata) / (ncol(TxBdata)-6) # Check that there is enough data to
  ↵ conduct a PCA. This ratio should be at least 5 (see Friginal & Hardy
  ↵ 2014: 303-304).
```

## Using specific subsets of the data

The following chunk can be used to perform the PCA on a country subset of the data to test the stability of the solution. See (Le Foll, n.d.) for a detailed analysis of the subcorpus of textbooks used in Germany.

```
TxBdata <- readRDS(here("data", "processed", "TxBdataforPCA.rds")) |>
  #filter(Country == "France")
  #filter(Country == "Germany")
  filter(Country == "Spain")

nrow(TxBdata)
TxBdata$Filename[1:10] # Check data
nrow(TxBdata) / (ncol(TxBdata)-6) # Check that there is enough data to
  ↵ conduct a PCA. This should be > 5 (see Friginal & Hardy 2014: 303-304).
```

## Performing the PCA

We perform the PCA using the `prcomp` function and print a summary of the results.

```
pca <- prcomp(TxBdata[, 7:ncol(TxBdata)], scale.=FALSE, rank. = 6) # All
  ↵ quantitative variables for all TxB files except outliers
register <- factor(TxBdata[, "Register"]) # Register
level <- factor(TxBdata[, "Level"]) # Textbook proficiency level
```

```
# summary(register)
# summary(level)
summary(pca)
```

```
Importance of first k=6 (out of 61) components:
          PC1      PC2      PC3      PC4      PC5      PC6
Standard deviation    2.1693 1.7776 1.08902 1.00207 0.84288 0.76792
Proportion of Variance 0.2108 0.1416 0.05313 0.04499 0.03183 0.02642
Cumulative Proportion 0.2108 0.3524 0.40553 0.45051 0.48234 0.50876
```

## Plotting PCA results

### 3D plots

The following chunk can be used to create projections of TEC texts on three dimensions of the model. These plots cannot be rendered in two dimensions and are therefore not generated in the present document. For more information on the `pca3d` library, see: <https://cran.r-project.org/web/packages/pca3d/vignettes/pca3d.pdf>.

```
library(pca3d) # For 3-D plots

col <- palette[c(1:3,8,7)] # without poetry
names(col) <- c("Conversation", "Fiction", "Informative", "Instructional",
  ↵ "Personal")
scales::show_col(col) # Check colours

pca3d(pca,
  group = register,
  components = 1:3,
  #components = 4:6,
  show.ellipses=FALSE,
  ellipse.ci=0.75,
  show.plane=FALSE,
  col = col,
  shape = "sphere",
  radius = 1,
  legend = "right")

snapshotPCA3d(here("plots", "PCA_TxB_3Dsnapshot.png"))
```

```

names(col) <- c("C", "B", "E", "A", "D") # To colour the dots according to
  ↴ the proficiency level of the textbooks
pca3d(pca,
       components = 4:6,
       group = level,
       show.ellipses=FALSE,
       ellipse.ci=0.75,
       show.plane=FALSE,
       col = col,
       shape = "sphere",
       radius = 0.8,
       legend = "right")

```

## Two-dimensional plots (biplots)

These plots were generated using the `PCAtools` package, which requires the data to be formatted in a rather unconventional way so it needs to wrangled first.

### Data wrangling for `PCAtools`

```

#TxBdata <- readRDS(here("data", "processed", "TxBdataforPCA.rds"))

TxBdata2meta <- TxBdata[,1:6]
rownames(TxBdata2meta) <- TxBdata2meta$Filename
TxBdata2meta <- TxBdata2meta |> select(-Filename)
#head(TxBdata2meta)

TxBdata2 = TxBdata
rownames(TxBdata2) <- TxBdata2$Filename
TxBdata2num <- as.data.frame(base::t(TxBdata2[,7:ncol(TxBdata2)]))
#TxBdata2num[1:12,1:3] # Check sanity of data

p <- PCAtools::pca(TxBdata2num,
                    metadata = TxBdata2meta,
                    scale = FALSE)

```

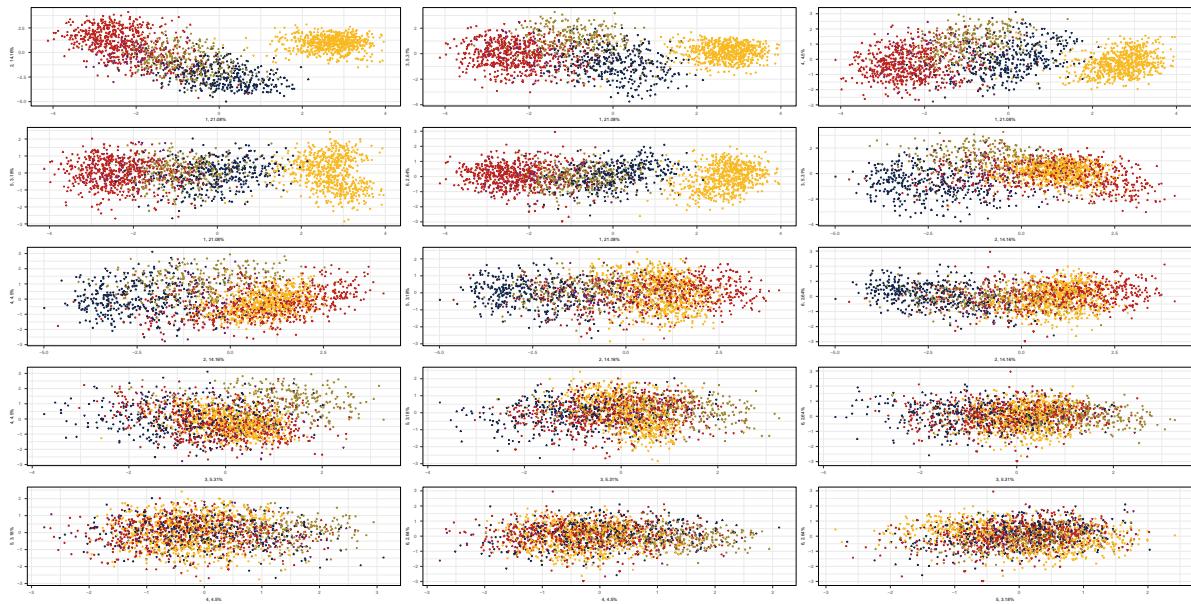
## Pairs plot

We first produce a scatterplot matrix of all the combinations of the first six dimensions of the model of intra-textbook variation. Note that the number before the comma on each axis label shows which principal component is plotted on that axis; this is followed by the percentage of the total variance explained by that particular component. The colours correspond to the text registers.

```
## Colour and shape scheme for all biplots
colkey = c(Conversation="#BD241E", Fiction="#A18A33", Informative="#15274D",
          ↵ Instructional="#F9B921", Personal="#722672")
shapekey = c(A=1, B=2, C=6, D=0, E=5)

## Very slow, open in zoomed out window!
# Add legend manually? Yes (take it from the biplot code below), sadly really
  ↵ the simplest solution, here. Or use Evert's mvar.pairs plot function
  ↵ (though that also requires manual axis annotation).

# png(here("plots", "PCA_TxB_pairsplot.png"), width = 12, height= 19, units =
  ↵ "cm", res = 300)
PCAtools::pairsplot(p,
                     triangle = FALSE,
                     components = 1:6,
                     ncol = 3,
                     nrow = 5,
                     pointSize = 0.8,
                     lab = NULL, # Otherwise will try to label each data point!
                     colby = "Register",
                     colkey = colkey,
                     shape = "Level",
                     shapekey = shapekey,
                     margin gaps = unit(c(0.2, 0.2, 0.2, 0.2), "cm"),
                     legendPosition = "none")
```



## Bi-plots

Then, biplots of the most important dimensions are generated to examine components more carefully.

```

colkey = c(Conversation="#BD241E", Fiction="#A18A33", Informative="#15274D",
         ↵   Instructional="#F9B921", Personal="#722672")
shapekey = c(A=1, B=2, C=6, D=0, E=5)

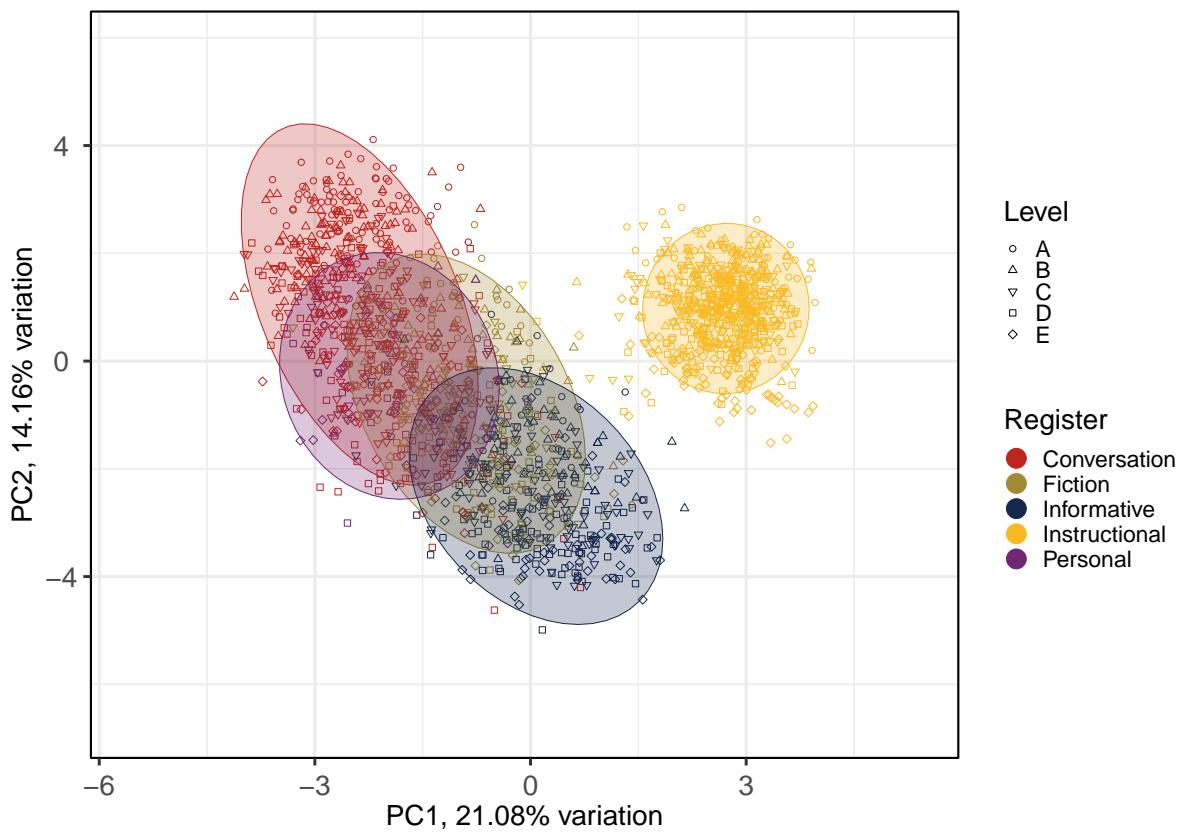
#png(here("plots", "PCA_TxB_Biplot_PC1_PC2.png"), width = 40, height= 25,
     ↵   units = "cm", res = 300)
PCAtools::biplot(p,
                  x = "PC1",
                  y = "PC2",
                  lab = NULL, # Otherwise will try to label each data point!
#xlim = c(min(p$rotated$PC1)-0.5, max(p$rotated$PC1)+0.5),
#ylim = c(min(p$rotated$PC2)-0.5, max(p$rotated$PC2)+0.5),
                  colby = "Register",
                  pointSize = 2,
                  colkey = colkey,
                  shape = "Level",
                  shapekey = shapekey,

```

```

showLoadings = FALSE,
ellipse = TRUE,
axisLabSize = 22,
legendPosition = 'right',
legendTitleSize = 22,
legendLabSize = 18,
legendIconSize = 7) +
theme(plot.margin = unit(c(0,0,0,0.2), "cm"))

```



```

#dev.off()
#ggsave(here("plots", "PCA_TxB_BiplotsPC1_PC2.svg"), width = 12, height = 10)

# Biplots to examine components more carefully
pRegisters <- PCAtools::biplot(p,
                                x = "PC3",

```

```

y = "PC4",
lab = NULL, # Otherwise will try to label each data point!
colby = "Register",
pointSize = 2,
colkey = colkey,
shape = "Level",
shapekey = shapekey,
showLoadings = FALSE,
ellipse = TRUE,
legendPosition = 'right',
legendTitleSize = 22,
legendLabSize = 18,
legendIconSize = 7) +
theme(plot.margin = unit(c(0,0,0,0.2), "cm"))

#ggsave(here("plots", "PCA_TxB_BiplotPC3_PC4.svg"), width = 12, height = 10)

# Biplots to examine components more carefully
pRegisters2 <- PCAtools::biplot(p,
x = "PC5",
y = "PC6",
lab = NULL, # Otherwise will try to label each data point!
colby = "Register",
pointSize = 2,
colkey = colkey,
shape = "Level",
shapekey = shapekey,
showLoadings = FALSE,
ellipse = TRUE,
legendPosition = 'right',
legendTitleSize = 22,
legendLabSize = 18,
legendIconSize = 7) +
theme(plot.margin = unit(c(0,0,0,0.2), "cm"))

#ggsave(here("plots", "PCA_TxB_BiplotPC5_PC6.svg"), width = 12, height = 10)

```

Changing the colour of the points and the ellipses to represent the texts' target proficiency levels instead of the register allows for a different interpretation of the model.

```

# Inverted keys for the biplots with ellipses for Level rather than Register
colkeyLevels = c(A="#F9B921", B="#A18A33", C="#BD241E", D="#722672",
                 E="#15274D")
shapekeyLevels = c(Conversation=1, Fiction=2, Informative=6, Instructional=0,
                  Personal=5)

pLevels <- PCAtools::biplot(p,
                            x = "PC3",
                            y = "PC4",
                            lab = NULL, # Otherwise will try to label each data point!
                            #xlim = c(min(p$rotated$PC1)-0.5, max(p$rotated$PC1)+0.5),
                            #ylim = c(min(p$rotated$PC2)-0.5, max(p$rotated$PC2)+0.5),
                            colby = "Level",
                            pointSize = 2,
                            colkey = colkeyLevels,
                            shape = "Register",
                            shapekey = shapekeyLevels,
                            showLoadings = FALSE,
                            ellipse = TRUE,
                            legendPosition = 'right',
                            legendTitleSize = 22,
                            legendLabSize = 18,
                            legendIconSize = 7) +
  theme(plot.margin = unit(c(0,0,0,0.2), "cm"))
#ggsave(here("plots", "PCA_TxB_BiplotPC3_PC4_Level.svg"), width = 12, height
#       = 10)

pLevels2 <- PCAtools::biplot(p,
                             x = "PC5",
                             y = "PC6",
                             lab = NULL, # Otherwise will try to label each data point!
                             #xlim = c(min(p$rotated$PC1)-0.5, max(p$rotated$PC1)+0.5),
                             #ylim = c(min(p$rotated$PC2)-0.5, max(p$rotated$PC2)+0.5),
                             colby = "Level",
                             pointSize = 2,
                             colkey = colkeyLevels,
                             shape = "Register",
                             shapekey = shapekeyLevels,
                             showLoadings = FALSE,
                             ellipse = TRUE,
                             legendPosition = 'right',
                             legendTitleSize = 22,

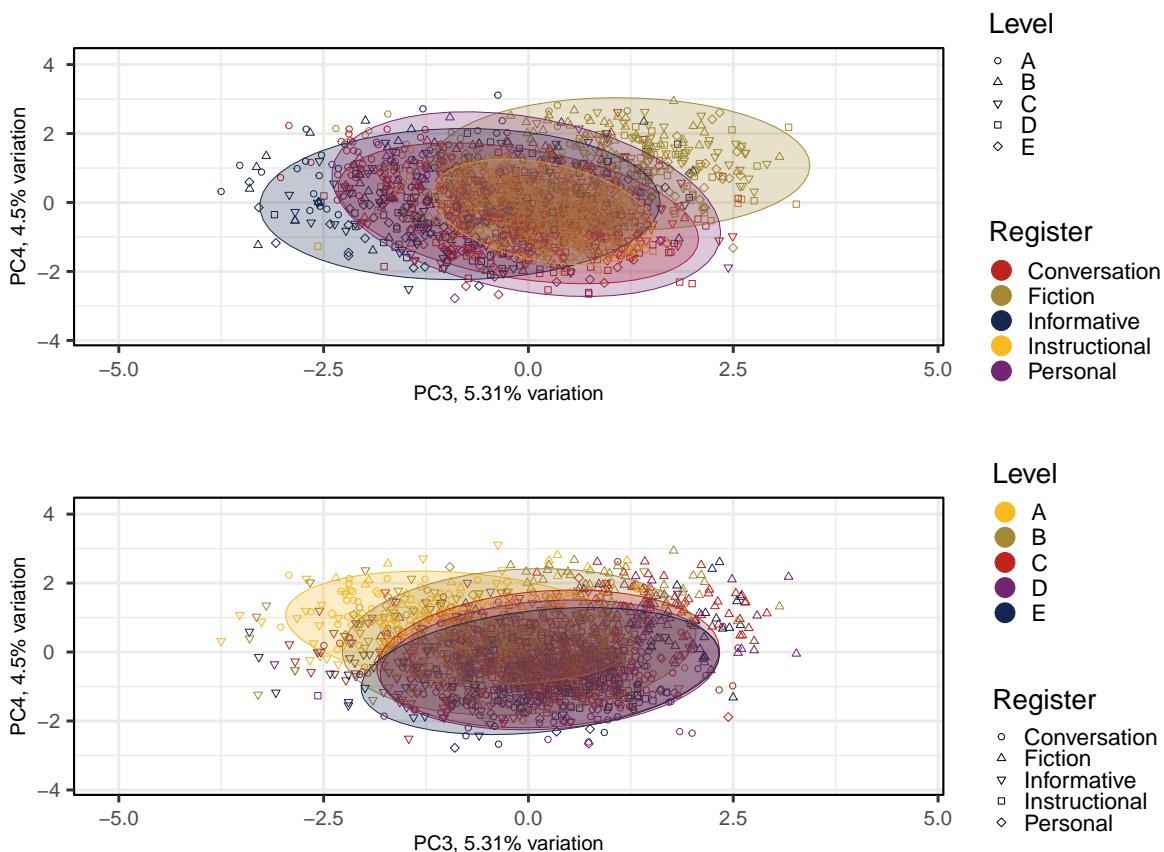
```

```

    legendLabSize = 18,
    legendIconSize = 7) +
theme(plot.margin = unit(c(0,0,0,0.2), "cm"))
#ggsave(here("plots", "PCA_TxB_BiplotPC5_PC6_Level.svg"), width = 12, height
        = 10)

# Display and save the two different representations of data points on PC2
# and PC3 using the {patchwork} package
pRegisters / pLevels

```



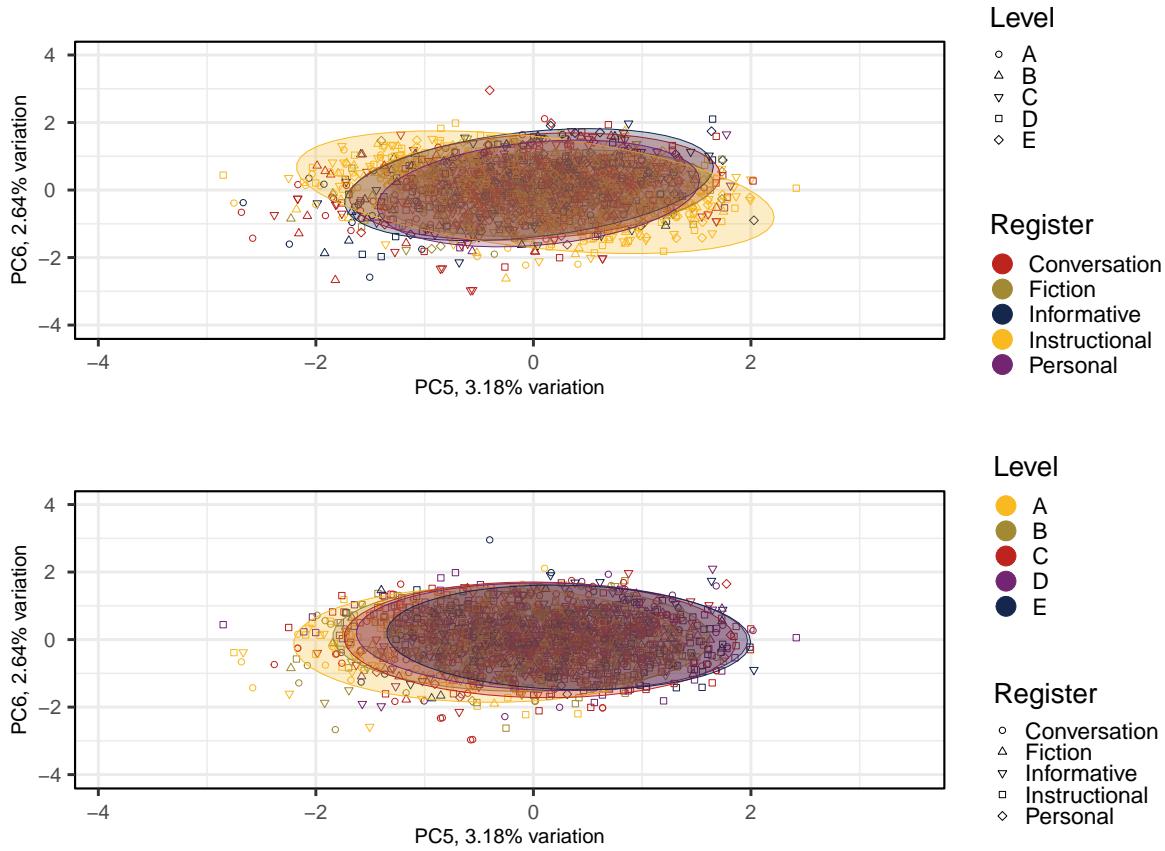
```

#ggsave(here("plots", "PCA_TxB_BiplotPC3_PC4_Register_vs_Level.svg"), width =
        = 14, height = 20)

# Display and save the two different representations of data points on PC5
# and PC6 using the {patchwork} package

```

```
pRegisters2 / pLevels2
```



```
#ggsave(here("plots", "PCA_TxB_BiplotPC5_PC6_Register_vs_Level.svg"), width =  
  ↵  14, height = 20)
```

## Feature contributions (loadings) on each component

```
#TxBdata <- readRDS(here("data", "processed", "TxBdataforPCA.rds"))  
  
pca <- prcomp(TxBdata[, 7:ncol(TxBdata)], scale.=FALSE) # All quantitative  
  ↵  variables for all TEC files
```

```

# The rotated data that represents the observations / samples is stored in
# ↵ rotated, while the variable loadings are stored in loadings
loadings <- as.data.frame(pca$rotation[,1:4])
loadings |>
  round(2) |>
  kable()

```

	PC1	PC2	PC3	PC4
ACT	0.08	-0.11	0.04	-0.10
AMP	-0.12	-0.10	-0.11	0.01
ASPECT	0.10	-0.05	0.14	-0.01
AWL	0.22	-0.16	-0.12	-0.13
BEMA	-0.22	0.01	-0.21	0.02
CC	0.05	-0.21	-0.19	0.00
COMM	0.20	0.09	0.14	-0.04
COND	-0.01	-0.02	0.11	-0.24
CONT	-0.25	0.11	-0.03	-0.06
CUZ	-0.09	-0.13	-0.06	-0.02
DEMO	-0.12	0.08	0.03	-0.09
DMA	-0.20	0.14	-0.02	0.00
DOAUX	-0.01	0.20	0.05	-0.15
DT	0.12	0.00	0.31	-0.02
EMPH	-0.19	-0.02	0.06	-0.14
EX	-0.10	-0.05	-0.11	0.05
EXIST	-0.02	-0.15	-0.09	-0.09
FPP1P	-0.17	0.01	-0.07	0.00
FPP1S	-0.23	0.07	0.08	-0.01
FPUH	-0.16	0.15	-0.09	0.07
FREQ	-0.03	-0.05	0.01	-0.10
IN	0.17	-0.18	0.02	-0.08
JJAT	-0.06	-0.18	0.04	-0.21
JJPR	-0.17	-0.06	-0.11	-0.11
LD	0.16	-0.03	-0.26	-0.01
MDCA	-0.04	0.10	-0.18	-0.09
MDCO	-0.05	-0.10	0.22	0.01
MDWS	-0.07	-0.01	0.05	-0.16
MENTAL	0.14	0.13	0.12	-0.25
NCOMP	0.04	-0.05	-0.24	-0.15
NN	0.20	-0.09	-0.29	0.11
OCCUR	0.02	-0.18	0.03	0.02
PASS	-0.01	-0.22	-0.06	-0.05

	PC1	PC2	PC3	PC4
PEAS	-0.06	-0.17	0.13	-0.13
PIT	-0.19	-0.04	-0.06	-0.06
PLACE	-0.16	-0.01	-0.07	0.09
POLITE	-0.14	0.13	-0.07	0.02
POS	-0.01	0.03	-0.04	0.16
PROG	-0.11	-0.02	0.11	0.00
QUAN	-0.15	-0.03	0.12	-0.19
QUPR	-0.10	-0.05	0.16	-0.11
RB	-0.19	-0.08	0.20	0.00
RP	0.00	-0.09	0.14	0.02
SPLIT	-0.11	-0.18	0.02	-0.16
SPP2	0.10	0.22	-0.01	-0.25
THATD	-0.05	0.04	0.16	-0.24
THRC	0.02	-0.11	-0.02	-0.18
THSC	-0.06	-0.17	0.07	-0.14
TIME	-0.12	-0.08	-0.01	0.06
TPP3P	-0.01	-0.16	-0.09	-0.02
TPP3S	-0.06	-0.11	0.13	0.30
TTR	-0.04	-0.26	-0.05	-0.01
VBD	-0.08	-0.20	0.23	0.30
VBG	0.04	-0.18	0.00	-0.22
VBN	0.03	-0.18	-0.07	-0.04
VIMP	0.25	0.15	0.04	-0.08
VPRT	-0.15	0.05	-0.32	-0.22
WHQU	0.11	0.23	0.00	-0.09
WHSC	0.11	-0.11	0.03	-0.15
XX0	-0.22	0.03	0.06	-0.06
YNQU	-0.03	0.23	0.00	-0.08

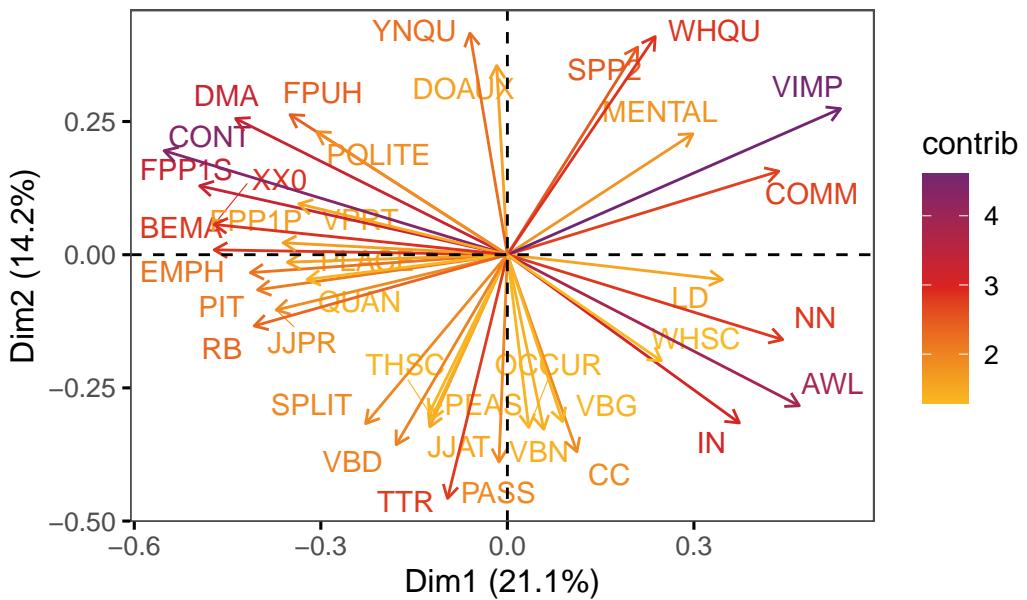
We can go back to the normalised frequencies of the individual features to compare them across different registers and levels, e.g.:

```
TxBcounts |>
  group_by(Register, Level) |>
  summarise(median(NCOMP), MAD(NCOMP)) |>
  select(1:4) |>
  kable(digits=2)
```

Register	Level	median(NCOMP)	MAD(NCOMP)
Conversation	A	5.69	2.79
Conversation	B	5.48	2.66
Conversation	C	5.32	2.58
Conversation	D	6.18	2.91
Conversation	E	6.21	2.62
Fiction	A	4.14	2.34
Fiction	B	3.96	2.17
Fiction	C	4.05	1.86
Fiction	D	5.05	2.34
Fiction	E	5.05	2.16
Informative	A	8.07	2.48
Informative	B	7.62	2.40
Informative	C	7.49	3.16
Informative	D	7.56	2.46
Informative	E	8.77	2.45
Instructional	A	6.84	2.54
Instructional	B	6.80	2.65
Instructional	C	6.14	2.35
Instructional	D	6.22	2.29
Instructional	E	6.75	2.69
Personal	A	6.72	1.42
Personal	B	4.92	2.33
Personal	C	5.75	1.45
Personal	D	6.46	3.19
Personal	E	8.22	3.09

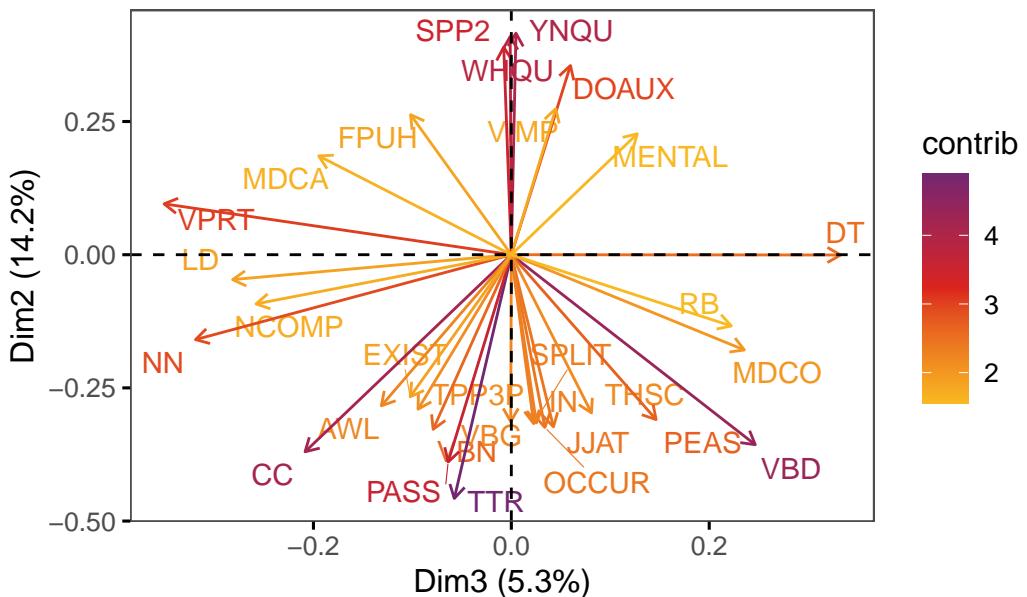
Graphs of features display the features with the strongest contributions to any two dimensions of the model of intra-textbook variation. They are created using the `factoextra::fviz_pca_var` function.

```
factoextra::fviz_pca_var(pca,
  axes = c(1,2),
  select.var = list(cos2 = 0.1),
  col.var = "contrib", # Colour by contributions to the PC
  gradient.cols = c("#F9B921", "#DB241E", "#722672"),
  title = "",
  repel = TRUE, # Try to avoid too much text overlapping
  ggtheme = ggthemes::theme_few())
```



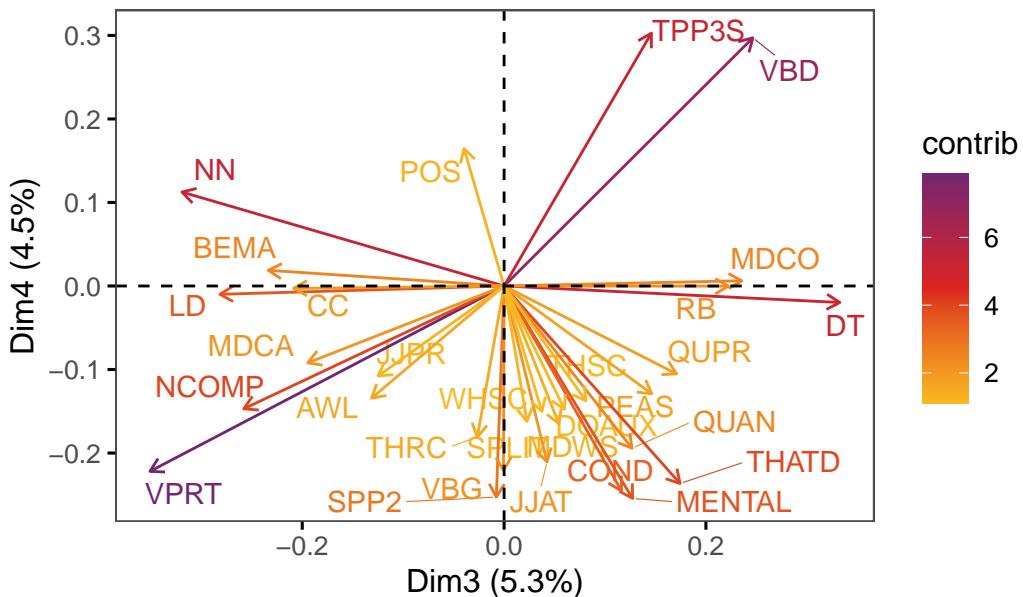
```
#ggsave(here("plots", "fviz_pca_var_PC1_PC2.svg"), width = 11, height = 9)

factoextra::fviz_pca_var(pca,
  axes = c(3,2),
  select.var = list(contrib = 30),
  col.var = "contrib", # Colour by contributions to the PC
  gradient.cols = c("#F9B921", "#DB241E", "#722672"),
  title = "",
  repel = TRUE, # Try to avoid too much text overlapping
  ggtheme = ggthemes::theme_few())
```



```
#ggsave(here("plots", "fviz_pca_var_PC3_PC2.svg"), width = 9, height = 8)

factoextra::fviz_pca_var(pca,
  axes = c(3,4),
  select.var = list(contrib = 30),
  col.var = "contrib", # Colour by contributions to the PC
  gradient.cols = c("#F9B921", "#DB241E", "#722672"),
  title = "",
  repel = TRUE, # Try to avoid too much text overlapping
  ggtheme = ggthemes::theme_few())
```



```
#ggsave(here("plots", "fviz_pca_var_PC3_PC4.svg"), width = 9, height = 8)
```

## Exploring the dimensions of the model

We begin with some descriptive statistics of the dimension scores.

```
#  
#<-- http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide  
  
#TxBdata <- readRDS(here("data", "processed", "TxBdataforPCA.rds"))  
  
pca <- prcomp(TxBdata[, 7:ncol(TxBdata)], scale.=FALSE) # All quantitative  
#<-- variables for all TxB files  
register <- factor(TxBdata[, "Register"]) # Register  
level <- factor(TxBdata[, "Level"]) # Textbook proficiency level  
  
# summary(register)  
# summary(level)  
# summary(pca)  
  
## Access to the PCA results for individual PC
```

```
#pca$rotation[,1]

res.ind <- cbind(TxBdata[,1:5], as.data.frame(pca$x)[,1:6])

res.ind |>
  group_by(Register) |>
  summarise_if(is.numeric, mean) |>
  kable(digits = 2)
```

Register	PC1	PC2	PC3	PC4	PC5	PC6
Conversation	-2.29	0.93	-0.14	-0.27	-0.02	0.06
Fiction	-0.85	-0.81	1.02	1.09	0.11	-0.10
Informative	0.06	-2.45	-0.83	0.01	-0.08	0.12
Instructional	2.68	0.93	0.15	-0.24	0.01	-0.07
Personal	-1.92	-0.29	-0.05	-0.02	0.07	-0.09

```
res.ind |>
  group_by(Register, Level) |>
  summarise_if(is.numeric, mean) |>
  kable(digits = 2)
```

Register	Level	PC1	PC2	PC3	PC4	PC5	PC6
Conversation	A	-2.39	2.39	-1.23	0.71	-0.45	-0.01
Conversation	B	-2.54	1.72	-0.25	0.04	-0.14	0.13
Conversation	C	-2.25	0.70	0.18	-0.41	0.09	-0.02
Conversation	D	-2.10	-0.08	0.28	-0.73	0.17	0.09
Conversation	E	-2.13	-0.14	0.07	-0.98	0.16	0.17
Fiction	A	-0.95	0.85	-0.54	1.48	-0.31	-0.46
Fiction	B	-0.89	-0.14	0.95	1.78	-0.06	-0.03
Fiction	C	-0.98	-0.81	1.62	1.23	0.26	-0.16
Fiction	D	-0.71	-1.57	1.27	0.72	0.21	-0.01
Fiction	E	-0.80	-1.45	1.16	0.56	0.25	-0.01
Informative	A	-0.09	-1.11	-1.94	0.87	-0.88	-0.15
Informative	B	0.15	-1.67	-1.19	0.46	-0.38	0.13
Informative	C	-0.02	-2.37	-0.68	-0.03	-0.06	-0.01
Informative	D	0.06	-2.89	-0.45	-0.19	0.06	0.10
Informative	E	0.15	-3.13	-0.79	-0.38	0.30	0.43
Instructional	A	2.89	1.55	-0.20	0.46	-0.34	-0.24

Register	Level	PC1	PC2	PC3	PC4	PC5	PC6
Instructional	B	2.68	1.27	0.09	0.00	-0.12	-0.12
Instructional	C	2.59	0.99	0.28	-0.32	-0.07	0.01
Instructional	D	2.63	0.70	0.28	-0.49	0.12	0.07
Instructional	E	2.64	0.09	0.20	-0.80	0.49	-0.16
Personal	A	-1.84	0.53	-1.11	1.21	-0.31	0.12
Personal	B	-1.85	0.40	-0.58	0.59	0.21	-0.07
Personal	C	-2.05	-0.46	0.52	-0.17	0.06	-0.03
Personal	D	-1.89	-1.05	0.45	-0.63	0.39	-0.06
Personal	E	-1.96	-0.92	0.21	-1.10	-0.19	-0.45

```
# res.ind |>
#   select(Register, Level, PC2) |>
#   group_by(Register, Level) |>
#   summarise_if(is.numeric, c(Median = median, MAD = mad)) |>
#   mutate(across(where(is.numeric), round, 2)) |>
#   as.data.frame()
```

The following chunk can be used to search for example texts that are located in specific areas of the biplots. For example, we can search for texts for have high scores on Dim3 and low ones on Dim2 to proceed with a qualitative comparison and analysis of these texts.

```
res.ind |>
  filter(PC3 > 2.5 & PC2 < -2) |>
  select(Filename, PC2, PC3) |>
  kable(digits = 2)
```

Filename	PC2	PC3
Achievers_B1_plus_Narrative_0005.txt	-3.88	2.60
Solutions_Intermediate_Plus_Spoken_0018.txt	-2.08	2.56
JTT_3_Narrative_0005.txt	-2.85	2.76
Achievers_B2_Narrative_00031.txt	-2.61	2.59
Access_4_Narrative_0006.txt	-2.19	3.18

## Computing mixed-effects models of the dimension scores

### Dimension 1: ‘Overt instructions and explanations’

Having compared various models, the following model is chosen as the best-fitting one.

```
# Models with Textbook series as random intercepts
md1 <- lmer(PC1 ~ Register*Level + (1|Series), data = res.ind, REML = FALSE)
md1Register <- lmer(PC1 ~ Register + (1|Series), data = res.ind, REML =
  FALSE)
md1Level <- lmer(PC1 ~ Level + (1|Series), data = res.ind, REML = FALSE)

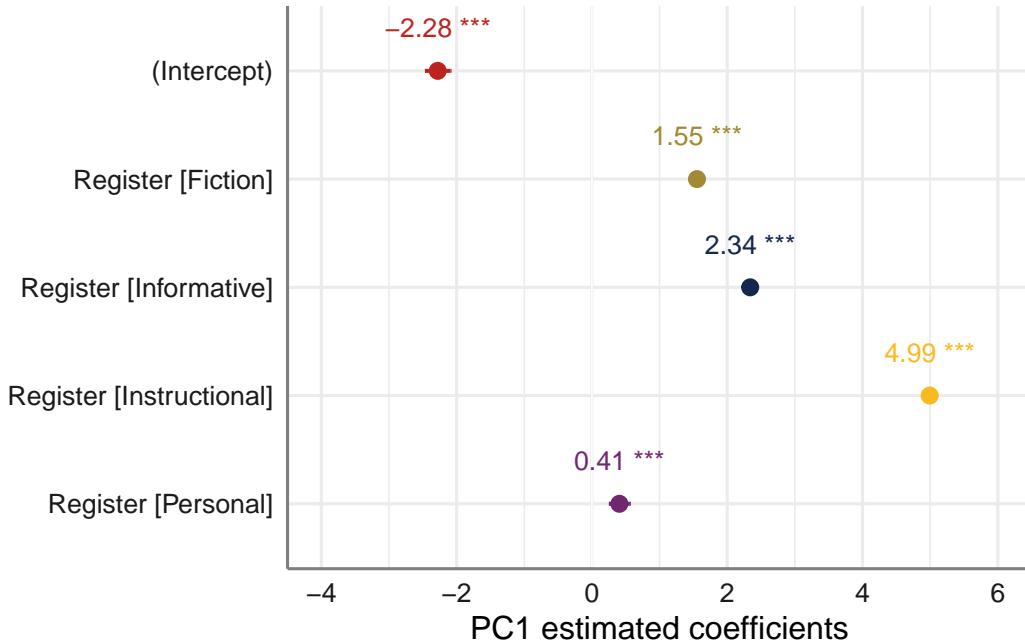
anova(md1, md1Register, md1Level)
```

```
Data: res.ind
Models:
md1Register: PC1 ~ Register + (1 | Series)
md1Level: PC1 ~ Level + (1 | Series)
md1: PC1 ~ Register * Level + (1 | Series)
      npar    AIC    BIC  logLik deviance   Chisq Df Pr(>Chisq)
md1Register     7 4080.4 4119.4 -2033.2    4066.4
md1Level        7 8533.0 8572.0 -4259.5    8519.0    0.0    0
md1            27 4068.3 4219.0 -2007.2    4014.3  4504.6 20 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
tab_model(md1, wrap.labels = 300) # Marginal R2 = 0.890
```

Its estimated coefficients are visualised in the plot below.

```
# Plot of fixed effects:
plot_model(md1Register,
           type = "est",
           show.intercept = TRUE,
           show.values=TRUE,
           show.p=TRUE,
           value.offset = .4,
           value.size = 3.5,
           colors = palette[c(1:3,8,7)],
           group.terms = c(1:5),
           title = "",
           wrap.labels = 40,
           axis.title = "PC1 estimated coefficients") +
theme_sjplot2()
```



```
#ggsave(here("plots", "TxB_PCA1_lmer_fixedeffects_Register.svg"), height = 3,
        width = 8)
```

The `emmeans` and `pairs` functions are used to compare the estimated Dim1 scores for each register and to compare these to one another.

```
Register_results <- emmeans(md1Register, "Register")
summary(Register_results)
```

Register	emmean	SE	df	lower.CL	upper.CL
Conversation	-2.2793	0.102	11.6	-2.502	-2.056
Fiction	-0.7267	0.106	13.9	-0.955	-0.498
Informative	0.0603	0.104	12.7	-0.165	0.286
Instructional	2.7141	0.101	11.3	2.492	2.937
Personal	-1.8734	0.122	25.5	-2.125	-1.622

Degrees-of-freedom method: kenward-roger  
 Confidence level used: 0.95

```
comparisons <- pairs(Register_results, adjust = "tukey")
comparisons
```

contrast	estimate	SE	df	t.ratio	p.value
Conversation - Fiction	-1.553	0.0508	1963	-30.535	<.0001
Conversation - Informative	-2.340	0.0465	1961	-50.341	<.0001
Conversation - Instructional	-4.993	0.0399	1961	-125.141	<.0001
Conversation - Personal	-0.406	0.0791	1958	-5.134	<.0001
Fiction - Informative	-0.787	0.0557	1962	-14.135	<.0001
Fiction - Instructional	-3.441	0.0497	1962	-69.168	<.0001
Fiction - Personal	1.147	0.0840	1958	13.645	<.0001
Informative - Instructional	-2.654	0.0447	1957	-59.399	<.0001
Informative - Personal	1.934	0.0816	1957	23.692	<.0001
Instructional - Personal	4.587	0.0780	1957	58.820	<.0001

Degrees-of-freedom method: kenward-roger

P value adjustment: tukey method for comparing a family of 5 estimates

```
#write_last_clip()
confint(comparisons)
```

contrast	estimate	SE	df	lower.CL	upper.CL
Conversation - Fiction	-1.553	0.0508	1963	-1.691	-1.414
Conversation - Informative	-2.340	0.0465	1961	-2.466	-2.213
Conversation - Instructional	-4.993	0.0399	1961	-5.102	-4.884
Conversation - Personal	-0.406	0.0791	1958	-0.622	-0.190
Fiction - Informative	-0.787	0.0557	1962	-0.939	-0.635
Fiction - Instructional	-3.441	0.0497	1962	-3.577	-3.305
Fiction - Personal	1.147	0.0840	1958	0.917	1.376
Informative - Instructional	-2.654	0.0447	1957	-2.776	-2.532
Informative - Personal	1.934	0.0816	1957	1.711	2.156
Instructional - Personal	4.587	0.0780	1957	4.374	4.800

Degrees-of-freedom method: kenward-roger

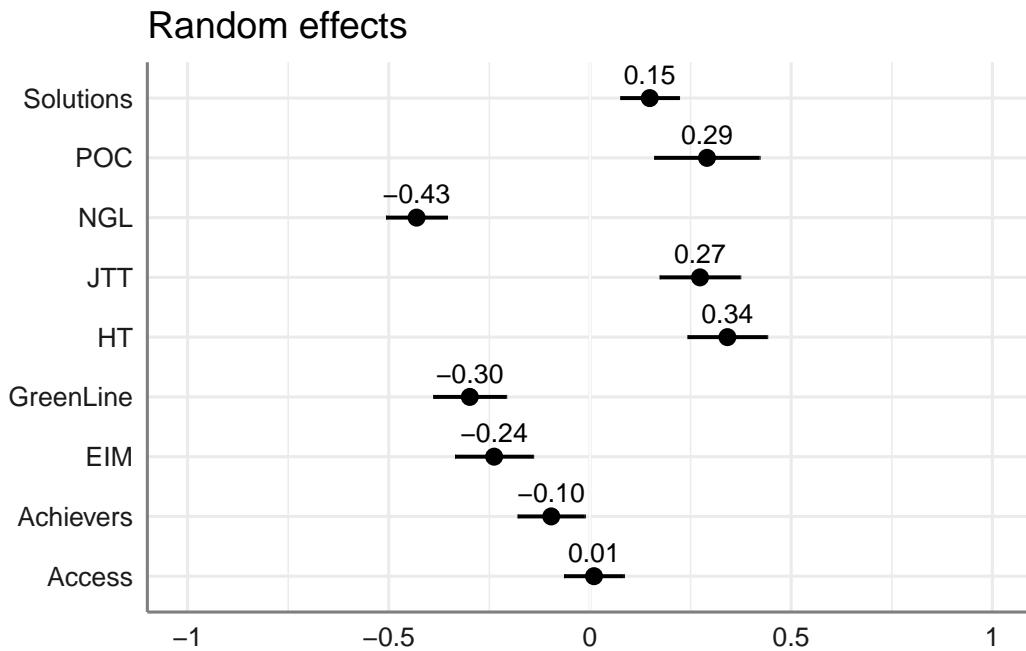
Confidence level used: 0.95

Conf-level adjustment: tukey method for comparing a family of 5 estimates

```
#write_last_clip()
```

We can also visualise the estimated coefficients for the textbook series, which is modelled here as a random effect.

```
plot_model(mdl,
            type = "re", # Option to visualise random effects
            show.values=TRUE,
            show.p=TRUE,
            value.offset = .4,
            value.size = 3.5,
            colors = "bw",
            wrap.labels = 40,
            axis.title = "PC1 estimated coefficients") +
theme_sjplot2()
```



```
#ggsave(here("plots", "TxB_PCA1_lmer_randomeffects.svg"), height = 3, width =
     8)
```

## Dimension 2: 'Involved vs. Informational Production'

```

md2 <- lmer(PC2 ~ Register*Level + (1|Series), data = res.ind, REML = FALSE)
md2Register <- lmer(PC2 ~ Register + (1|Series), data = res.ind, REML =
  FALSE)
md2Level <- lmer(PC2 ~ Level + (1|Series), data = res.ind, REML = FALSE)
anova(md2, md2Register, md2Level)

```

```

Data: res.ind
Models:
  md2Register: PC2 ~ Register + (1 | Series)
  md2Level: PC2 ~ Level + (1 | Series)
  md2: PC2 ~ Register * Level + (1 | Series)
    npar      AIC      BIC  logLik deviance   Chisq Df Pr(>Chisq)
  md2Register     7 6155.2 6194.3 -3070.6    6141.2
  md2Level        7 7290.1 7329.2 -3638.1    7276.1    0.0    0
  md2            27 5200.9 5351.6 -2573.4    5146.9  2129.2  20 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
tab_model(md2) # Marginal R2 = 0.723
```

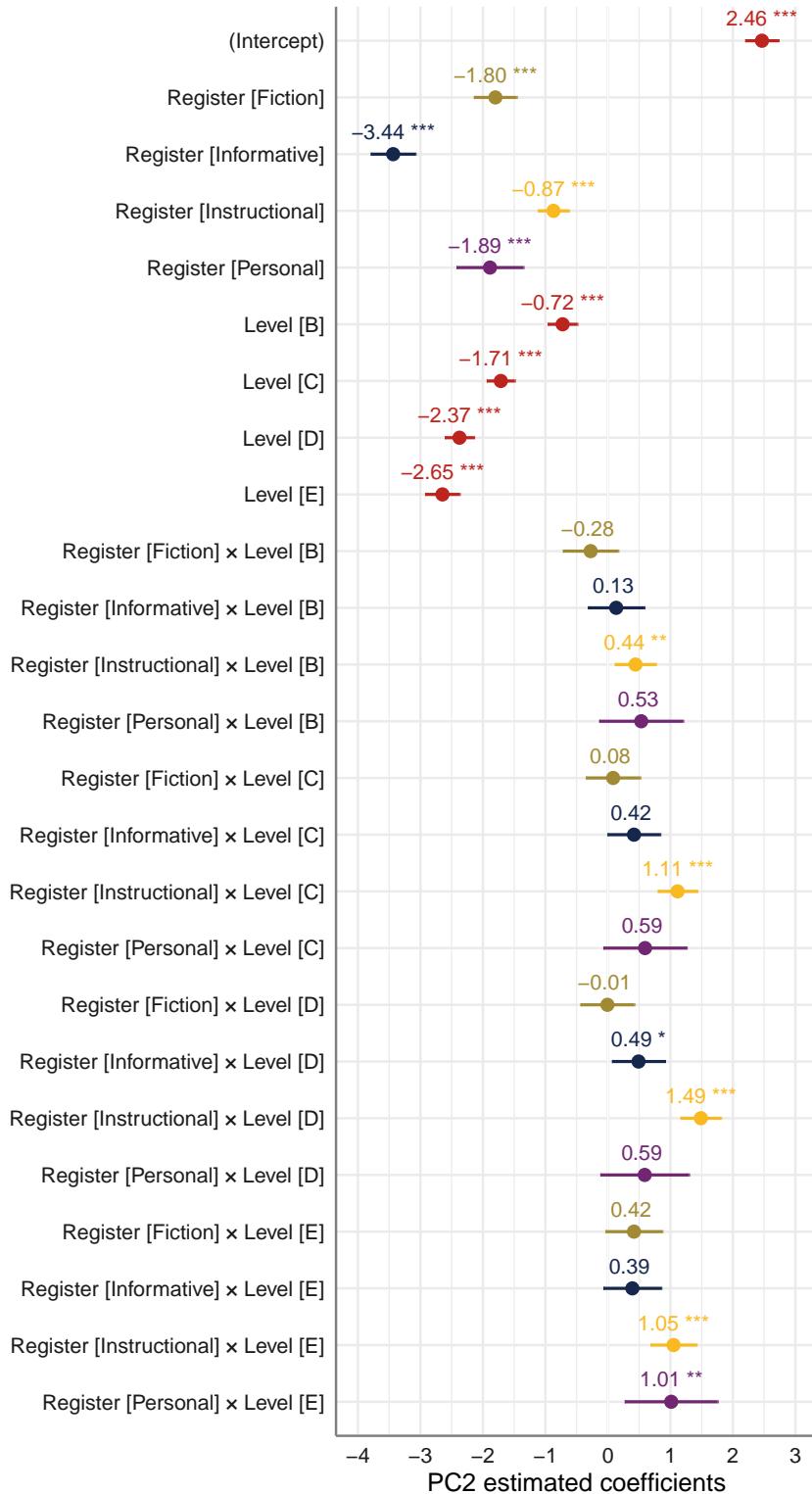
```
# tab_model(md2Register) # Marginal R2 = 0.558
# tab_model(md2Level) # Marginal R2 = 0.228
```

Estimated coefficients of fixed effects on Dim2 scores:

```

plot_model(md2,
  type = "est",
  show.intercept = TRUE,
  show.values=TRUE,
  show.p=TRUE,
  value.offset = .4,
  value.size = 3.5,
  colors = palette[c(1:3,8,7)],
  group.terms = c(1:5,1,1,1,1,2:5,2:5,2:5,2:5),
  title = "",
  wrap.labels = 40,
  axis.title = "PC2 estimated coefficients") +
theme_sjplot2()

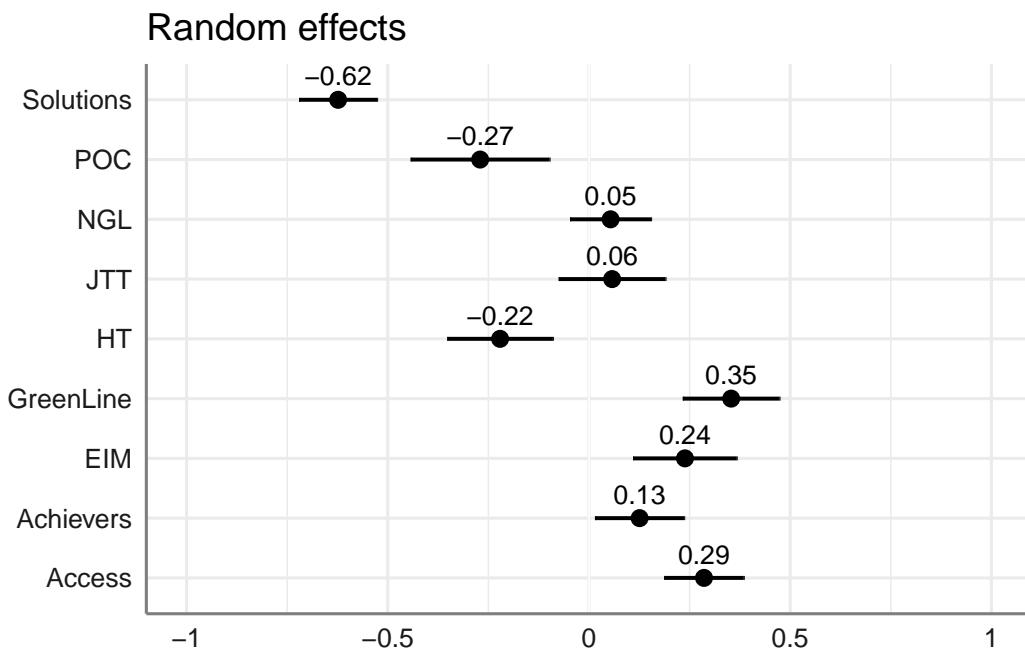
```



```
#ggsave(here("plots", "TxB_PCA2_lmer_fixedeffects.svg"), height = 8, width =
  8)
```

Estimated coefficients of random effects on Dim2 scores:

```
## Random intercepts
plot_model(md2,
            type = "re", # Option to visualise random effects
            show.values=TRUE,
            show.p=TRUE,
            value.offset = .4,
            value.size = 3.5,
            colors = "bw",
            wrap.labels = 40,
            axis.title = "PC2 estimated coefficients") +
  theme_sjplot2()
```



```
#ggsave(here("plots", "TxB_PCA2_lmer_randomeffects.svg"), height = 3, width =
  8)

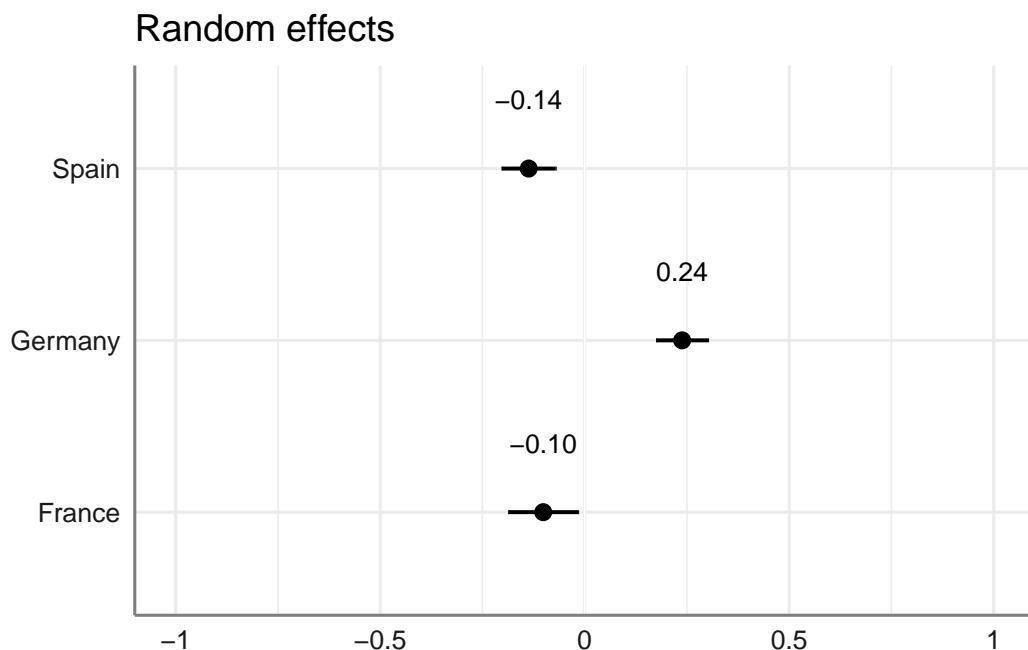
# Textbook Country as a random effect variable
```

```

md2country <- lmer(PC2 ~ Register*Level + (1|Country), data = res.ind, REML =
  FALSE)

plot_model(md2country,
  type = "re", # Option to visualise random effects
  show.values=TRUE,
  show.p=TRUE,
  value.offset = .4,
  value.size = 3.5,
  colors = "bw",
  wrap.labels = 40,
  axis.title = "PC2 estimated coefficients") +
theme_sjplot2()

```



```

#ggsave(here("plots", "TxB_PCA2_lmer_randomeffects_country.svg"), height = 3,
  width = 8)

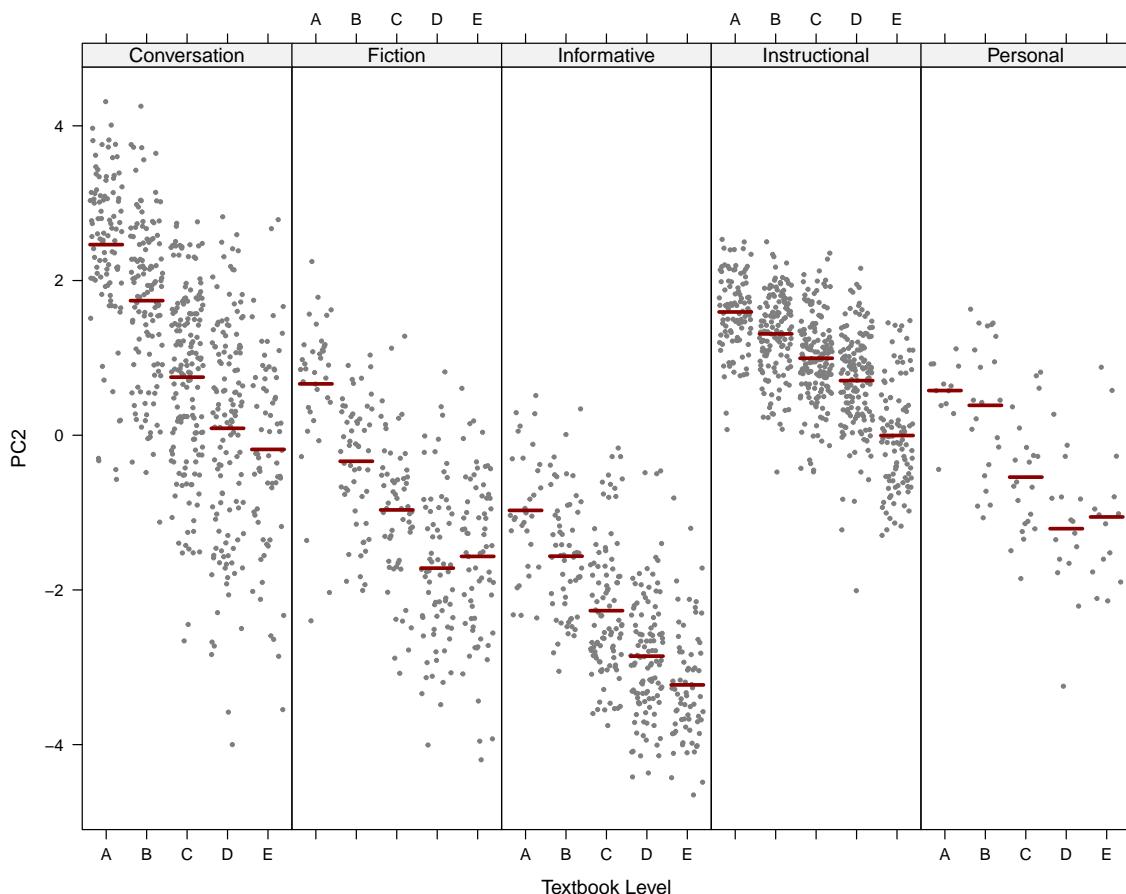
```

The `visreg` function is used to visualise the distributions of the modelled Dim2 scores:

```

# svg(here("plots", "TxB_predicted_PC2_scores_interactions.svg"), height = 5,
#      width = 8)
visreg(md2, xvar = "Level", by="Register", type = "conditional",
       line=list(col="darkred"),
       xlab = "Textbook Level", ylab = "PC2"
#,gg = TRUE
,layout=c(5,1)
)

```

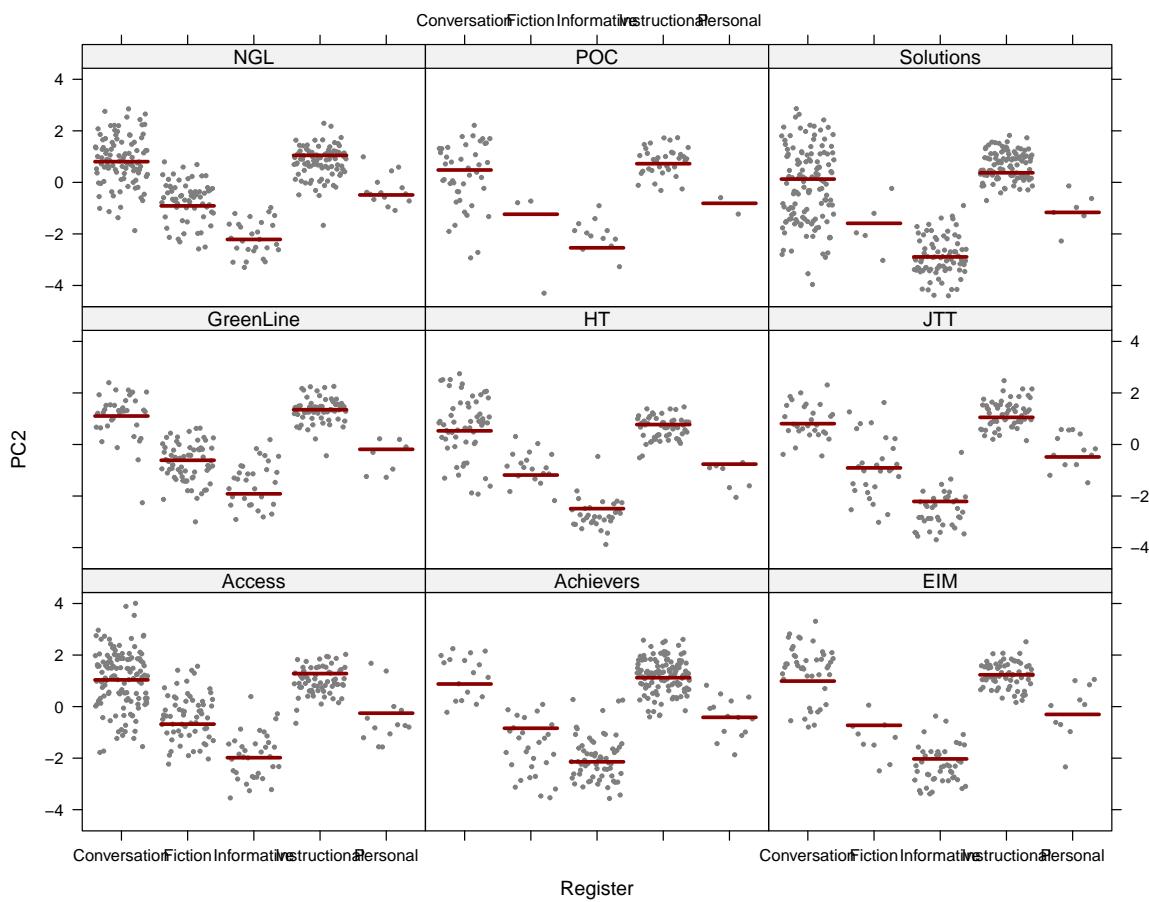


```

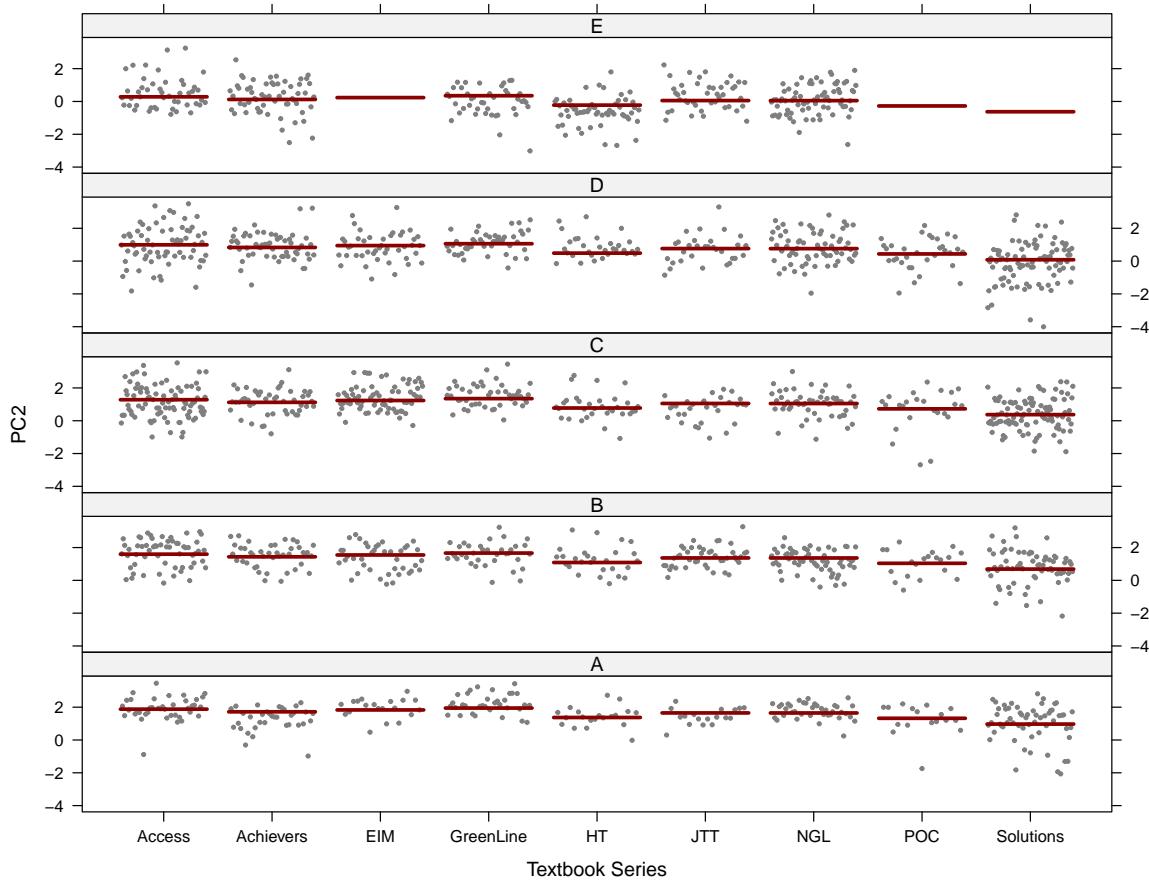
# dev.off()

# Textbook Series-Register interactions
visreg::visreg(md2, "Register", by="Series", re.form=~(1|Series),
               ylab="PC2", line=list(col="darkred"))

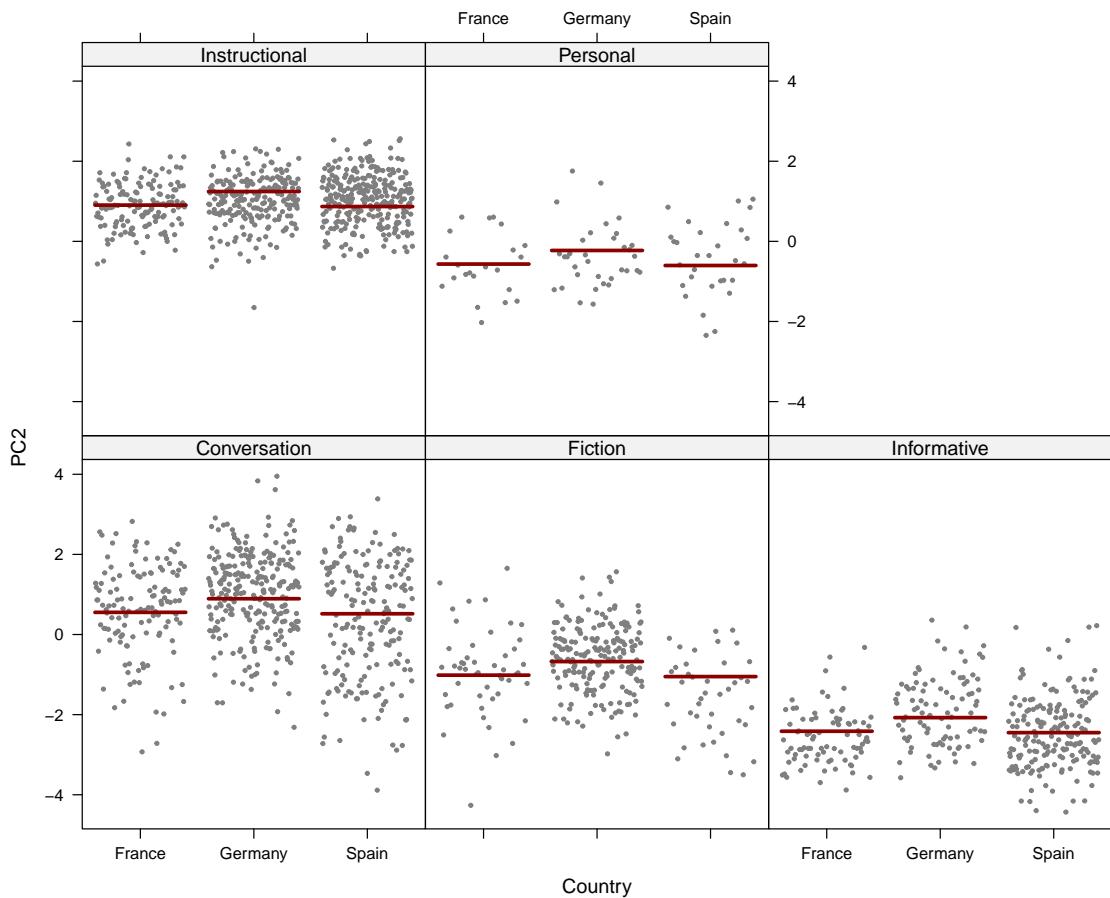
```



```
visreg(md2, xvar = "Series", by="Level", type = "conditional",
  re.form=~(1|Series),
  line=list(col="darkred"), xlab = "Textbook Series", ylab = "PC2",
  layout=c(1,5))
```



```
# Textbook Series-Register interactions
# svg(here("plots", "TxB_PCA2_lmer_randomeffects_country_register.svg"),
#      height = 5, width = 8)
visreg::visreg(mdl2country, "Country", by="Register", re.form=~(1|Country),
               ylab="PC2", line=list(col="darkred"))
```



```
# dev.off()
```

### Dimension 3: 'Narrative vs. factual discourse'

```
md3 <- lmer(PC3 ~ Register*Level + (1|Series), data = res.ind, REML = FALSE)
md3Register <- lmer(PC3 ~ Register + (1|Series), data = res.ind, REML =
  FALSE)
md3Level <- lmer(PC3 ~ Level + (1|Series), data = res.ind, REML = FALSE)

anova(md3, md3Register, md3Level)
```

Data: res.ind

```

Models:
md3Register: PC3 ~ Register + (1 | Series)
md3Level: PC3 ~ Level + (1 | Series)
md3: PC3 ~ Register * Level + (1 | Series)
      npar     AIC     BIC logLik deviance Chisq Df Pr(>Chisq)
md3Register    7 5139.9 5179.0 -2563.0    5125.9
md3Level       7 5528.8 5567.9 -2757.4    5514.8    0.00   0
md3          27 4582.6 4733.3 -2264.3    4528.6  986.21  20 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

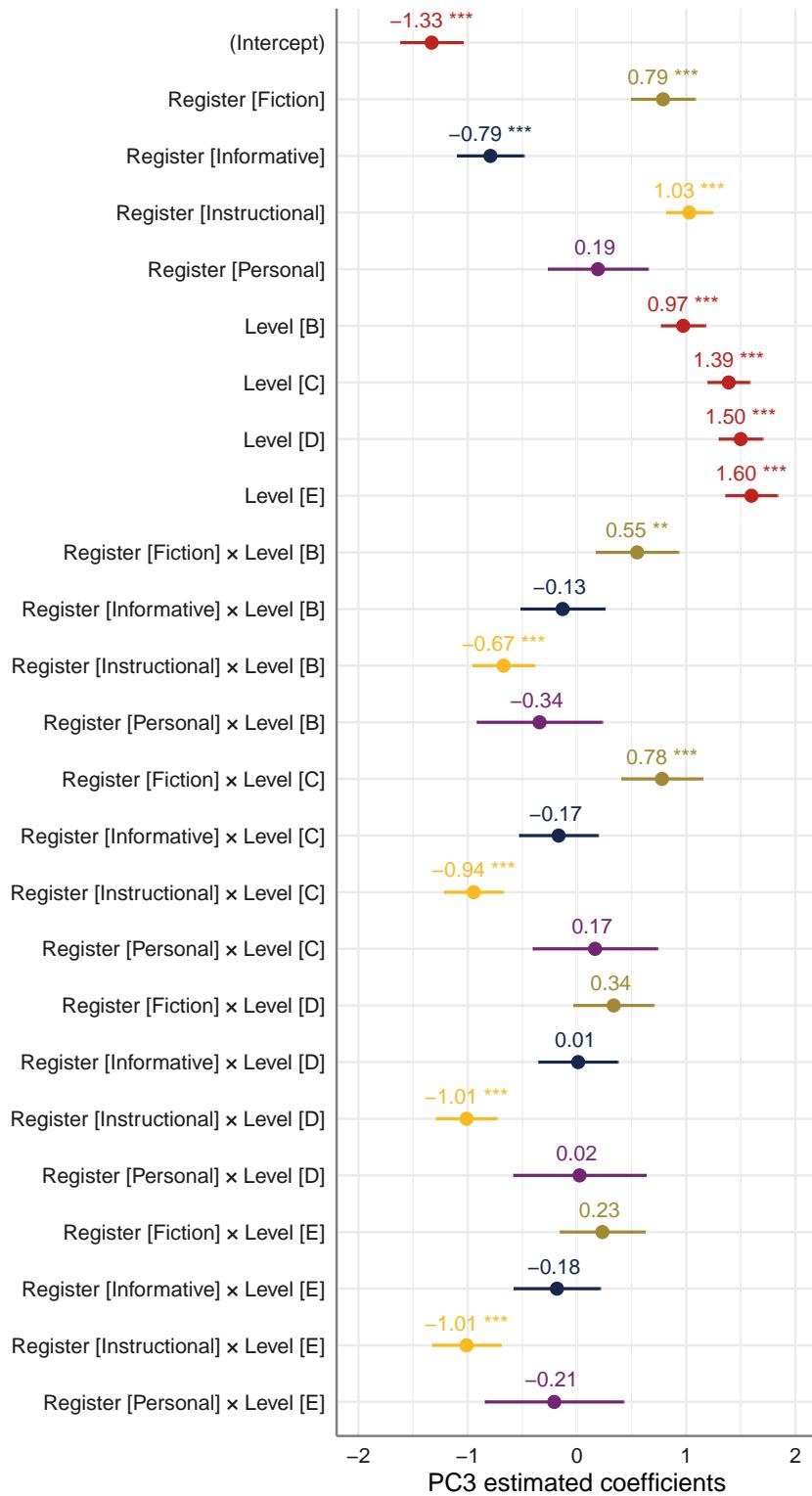
```
tab_model(md3) # Marginal R2 = 0.436
```

```

# tab_model(md3Register) # Marginal R2 = 0.272
# tab_model(md3Level) # Marginal R2 = 0.119

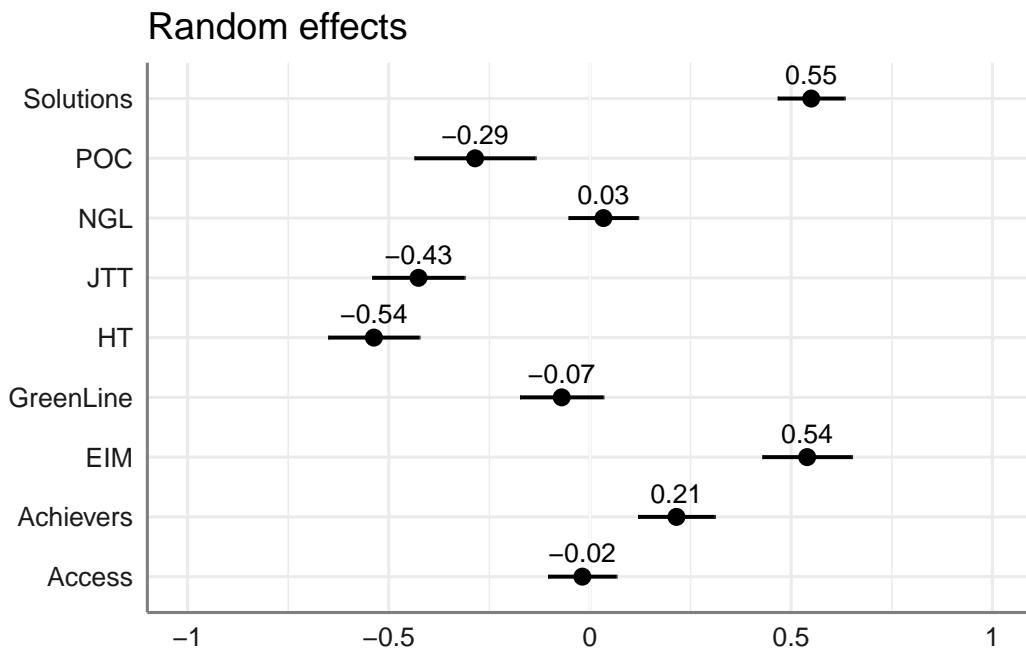
# Plot of fixed effects:
plot_model(md3,
            type = "est",
            show.intercept = TRUE,
            show.values=TRUE,
            show.p=TRUE,
            value.offset = .4,
            value.size = 3.5,
            colors = palette[c(1:3,8,7)],
            group.terms = c(1:5,1,1,1,1,2:5,2:5,2:5,2:5),
            title = "",
            wrap.labels = 40,
            axis.title = "PC3 estimated coefficients") +
theme_sjplot2()

```



```
#ggsave(here("plots", "TxB_PCA3_lmer_fixedeffects.svg"), height = 8, width =
  8)
```

```
# Plot of random effects:
plot_model(md3,
            type = "re", # Option to visualise random effects
            show.values=TRUE,
            show.p=TRUE,
            value.offset = .4,
            value.size = 3.5,
            color = "bw",
            wrap.labels = 40,
            axis.title = "PC3 estimated coefficients") +
theme_sjplot2()
```



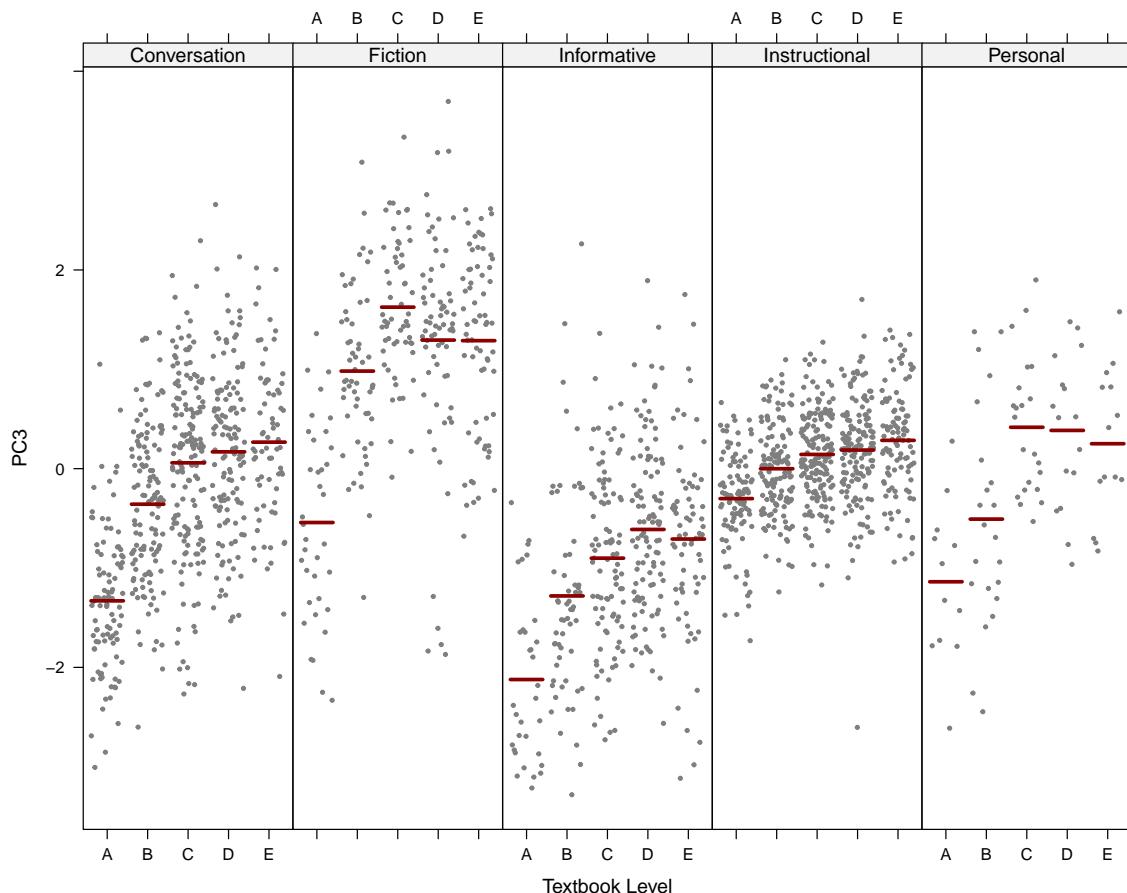
```
#ggsave(here("plots", "TxB_PCA3_lmer_randomeffects.svg"), height = 3, width =
  8)
```

```
# svg(here("plots", "TxB_predicted_PC3_scores_interactions.svg"), height = 5,
  width = 8)
visreg(md3, xvar = "Level", by="Register", type = "conditional",
```

```

    line=list(col="darkred"),
    xlab = "Textbook Level", ylab = "PC3"
  #,gg = TRUE
  ,layout=c(5,1)
)

```

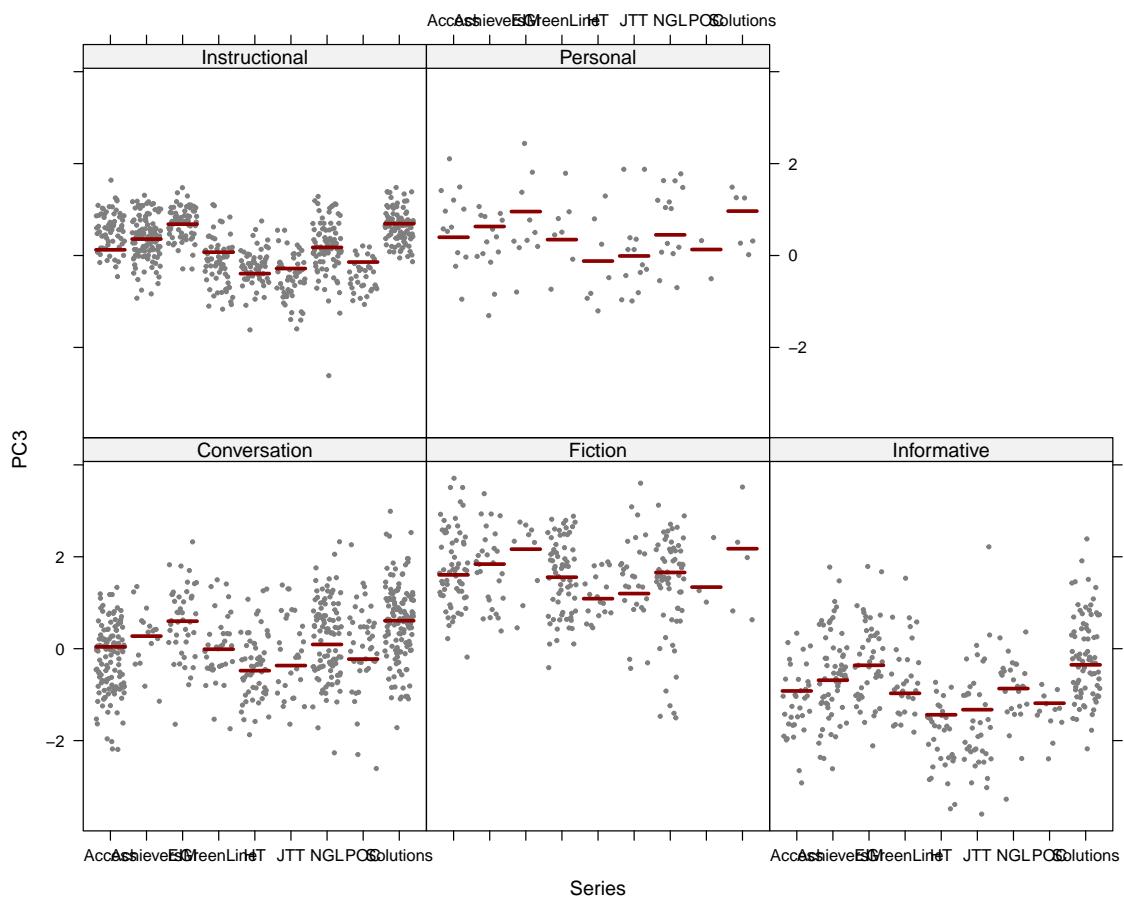


```

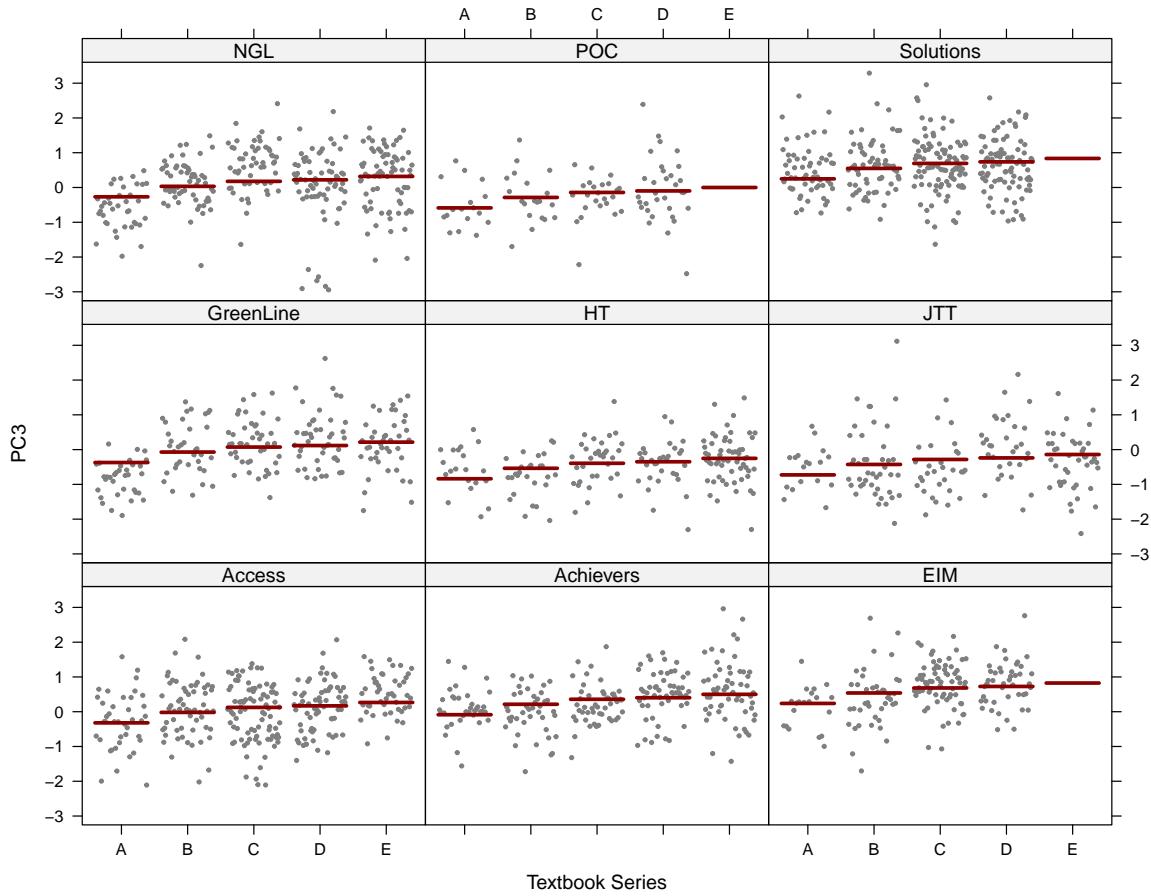
# dev.off()

# Textbook Series-Register interactions
visreg::visreg(mdl3, "Series", by="Register", re.form=~(1|Series),
               ylab="PC3", line=list(col="darkred"))

```



```
visreg(md3, xvar = "Level", by="Series", type = "conditional",
      re.form=~(1|Series),
      line=list(col="darkred"), xlab = "Textbook Series", ylab = "PC3")
```



#### Dimension 4: 'Informational compression vs. elaboration'

```
md4 <- lmer(PC4 ~ Register*Level + (1|Series), data = res.ind, REML = FALSE)
md4Register <- lmer(PC4 ~ Register + (1|Series), data = res.ind, REML =
  FALSE)
md4Level <- lmer(PC4 ~ Level + (1|Series), data = res.ind, REML = FALSE)

anova(md4, md4Register, md4Level)
```

```
Data: res.ind
Models:
md4Register: PC4 ~ Register + (1 | Series)
md4Level: PC4 ~ Level + (1 | Series)
```

```

md4: PC4 ~ Register * Level + (1 | Series)
      npar     AIC     BIC  logLik deviance  Chisq Df Pr(>Chisq)
md4Register    7 5034.0 5073.0 -2510.0    5020.0
md4Level       7 5043.6 5082.7 -2514.8    5029.6   0.00   0
md4          27 4372.1 4522.8 -2159.1    4318.1 711.52 20 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

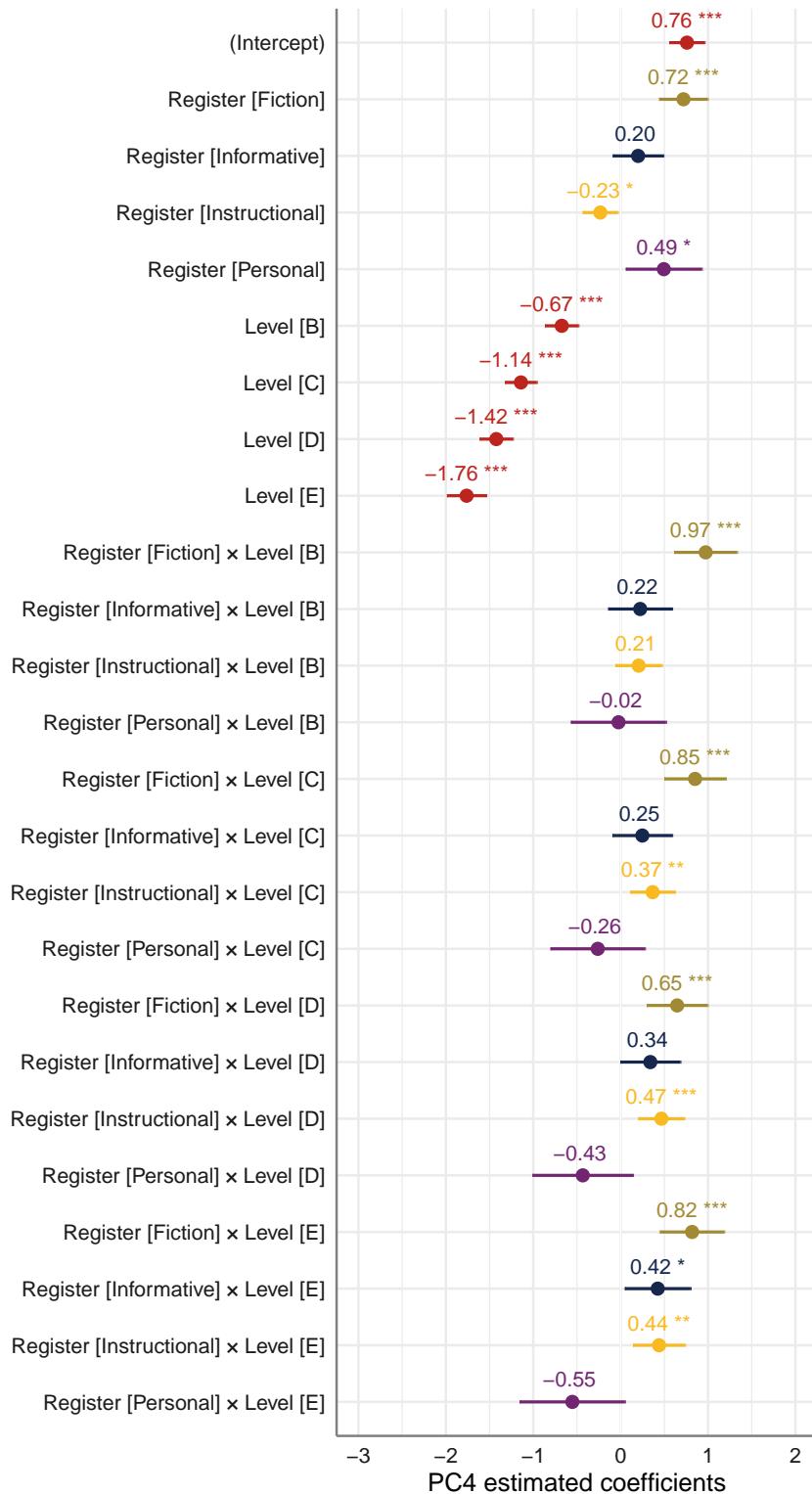
```
tab_model(md4) # Marginal R2 = 0.426
```

```

# tab_model(md4Register) # Marginal R2 = 0.203
# tab_model(md4Level) # Marginal R2 = 0.187

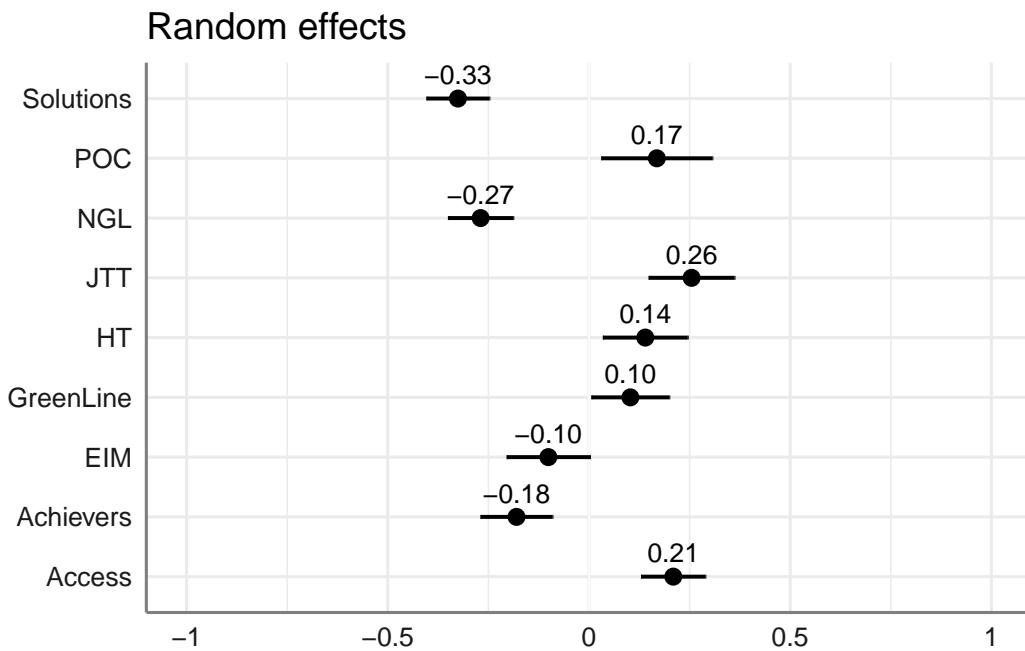
# Plot of fixed effects:
plot_model(md4,
            type = "est",
            show.intercept = TRUE,
            show.values=TRUE,
            show.p=TRUE,
            value.offset = .4,
            value.size = 3.5,
            colors = palette[c(1:3,8,7)],
            group.terms = c(1:5,1,1,1,1,2:5,2:5,2:5,2:5),
            title = "",
            wrap.labels = 40,
            axis.title = "PC4 estimated coefficients") +
theme_sjplot2()

```



```
#ggsave(here("plots", "TxB_PCA4_lmer_fixedeffects.svg"), height = 8, width =
  8)
```

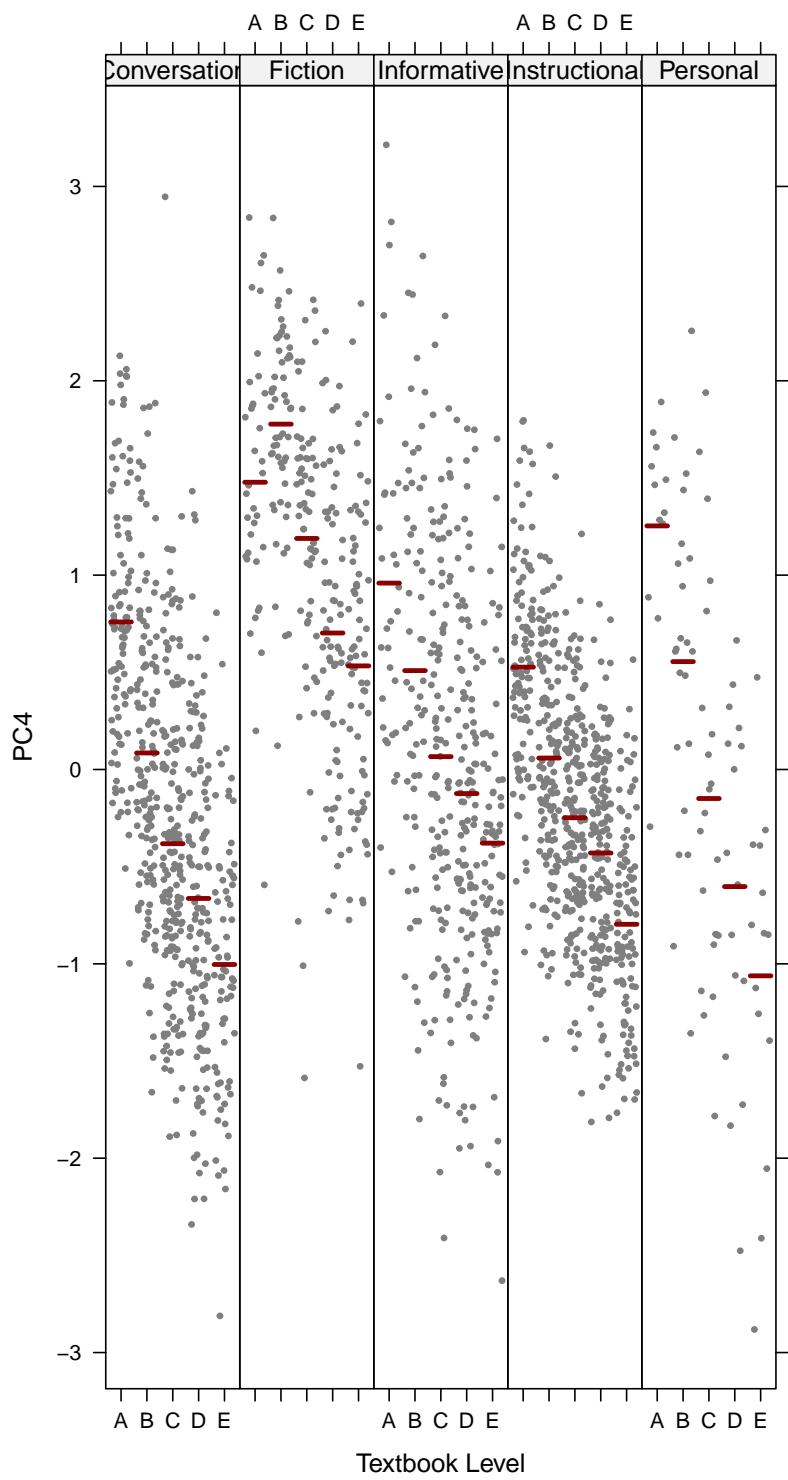
```
# Plot of random effects:
plot_model(md4,
            type = "re", # Option to visualise random effects
            show.values=TRUE,
            show.p=TRUE,
            value.offset = .4,
            value.size = 3.5,
            color = "bw",
            wrap.labels = 40,
            axis.title = "PC4 estimated coefficients") +
theme_sjplot2()
```



```
#ggsave(here("plots", "TxB_PCA4_lmer_randomeffects.svg"), height = 3, width =
  8)
```

```
# svg(here("plots", "TxB_predicted_PC4_scores_interactions.svg"), height = 5,
  width = 8)
visreg(md4, xvar = "Level", by="Register", type = "conditional",
```

```
line=list(col="darkred"),
xlab = "Textbook Level", ylab = "PC4"
#,gg = TRUE
,layout=c(5,1)
)
```



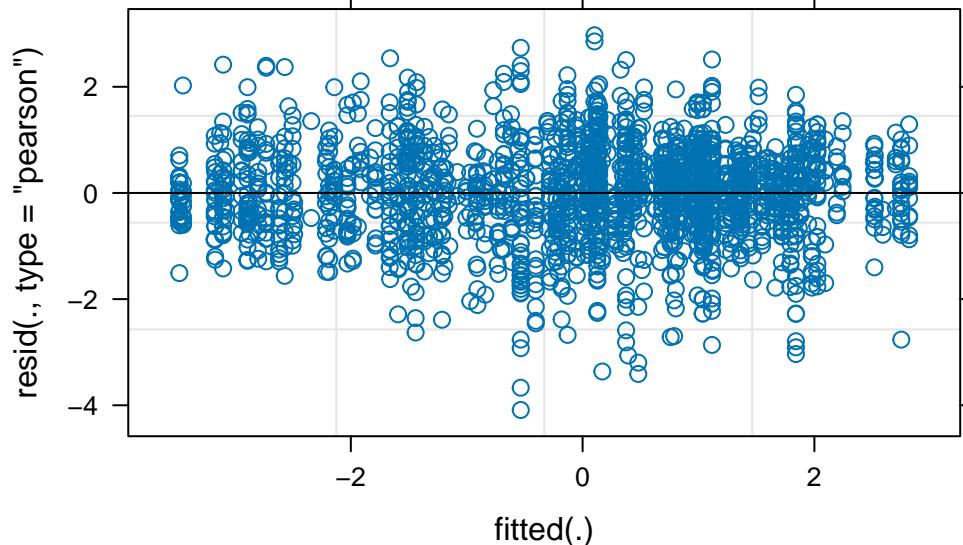
```
# dev.off()
```

## Testing model assumptions

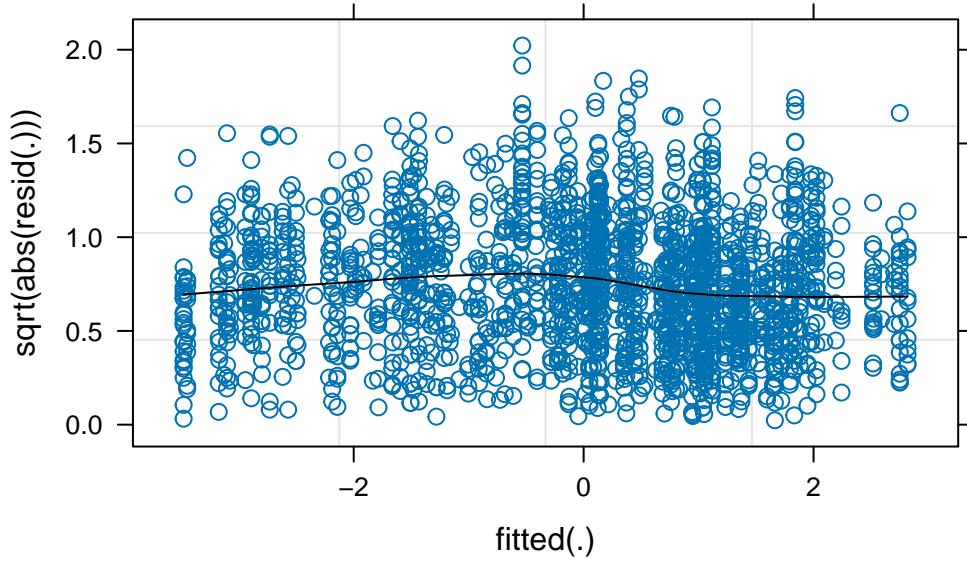
This chunk can be used to check the assumptions of all of the models computed above. In the following example, we examine the final model selected to predict Dim2 scores.

```
model2test <- md2

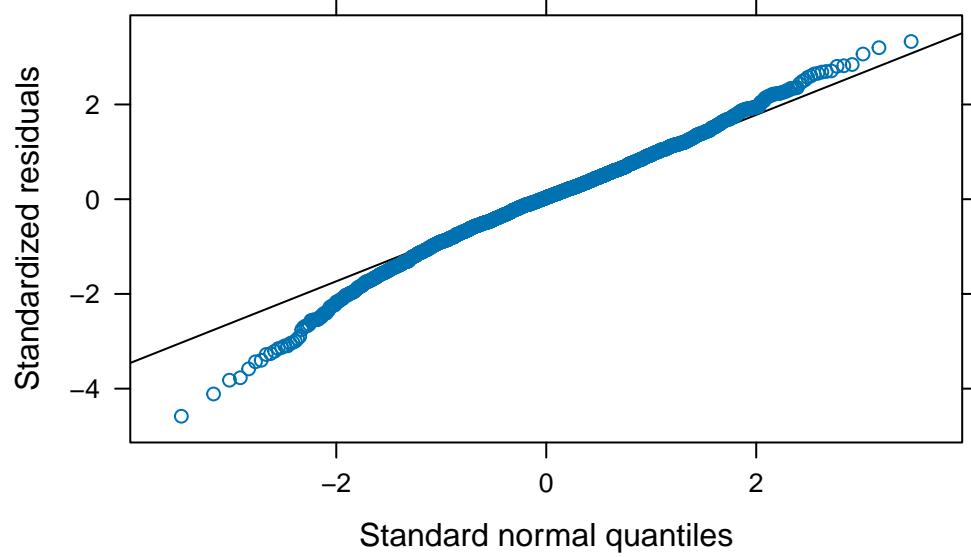
# check distribution of residuals
plot(model2test)
```



```
# scale-location plot
plot(model2test,
      sqrt(abs(resid(.)))~fitted(.),
      type=c("p","smooth"), col.line=1)
```



```
# Q-Q plot  
lattice::qqmath(model2test)
```



## **3 Summary**

In summary, this book has no content whatsoever.

# References

- Association for Computing Machinery, (ACM). 2020. “Artifact Review and Badging Version 1.1.” <https://www.acm.org/publications/policies/artifact-review-and-badging-current>.
- Berez-Kroeker, Andrea L., Lauren Gawne, Susan Smythe Kung, Barbara F. Kelly, Tyler Heston, Gary Holton, Peter Pulsifer, et al. 2018. “Reproducible Research in Linguistics: A Position Statement on Data Citation and Attribution in Our Field.” *Linguistics* 56 (1): 1–18. <https://doi.org/10.1515/ling-2017-0032>.
- Diwersy, Sascha, Stephanie Evert, and Stella Neumann. 2014. “A Weakly Supervised Multivariate Approach to the Study of Language Variation.” In, edited by Benedikt Szmrecsanyi and Bernhard Wälchli, 174–204. Berlin: De Gruyter.
- Le Foll, Elen. 2021a. *Introducing the Multi-Feature Tagger of English (MFTE)*. Osnabrück University. <https://github.com/elenlefoll/MultiFeatureTaggerEnglish>.
- . 2021b. *Introducing the Multi-Feature Tagger of English (MFTE)*. Osnabrück University. <https://github.com/elenlefoll/MultiFeatureTaggerEnglish>.
- . 2024. “Why We Need Open Science and Open Education to Bridge the Corpus Research-practice Gap.” In, edited by Peter Crosthwaite, 142–56. London: Routledge.
- . n.d. “Schulenglisch: A Multi-Dimensional Model of the Variety of English Taught in German Secondary Schools.” *AAA: Arbeiten Aus Anglistik Und Amerikanistik* 49.
- Love, Robbie, Vaclav Brezina, Tony McEnery, Abi Hawtin, Andrew Hardie, and Claire Dembry. 2019. “Functional Variation in the Spoken BNC2014 and the Potential for Register Analysis.” *Register Studies* 1 (2): 296–317. <https://doi.org/10.1075/rs.18013.lov>.
- Love, Robbie, Claire Dembry, Andrew Hardie, Vaclav Brezina, and Tony McEnery. 2017. “The Spoken BNC2014.” *International Journal of Corpus Linguistics* 22 (3): 319–44. <https://doi.org/https://doi.org/10.1075/ijcl.22.3.02lov>.
- McManus, Kevin. 2021. “Are Replication Studies Infrequent Because of Negative Attitudes? Insights from a Survey of Attitudes and Practices in Second Language Research.” *Studies in Second Language Acquisition*, December, 1–14. <https://doi.org/10.1017/S0272263121000838>.
- Neumann, Stella, and Stephanie Evert. 2021. “A Register Variation Perspective on Varieties of English.” In, edited by Elena Seoane and Douglas Biber, 144178. *Studies in Corpus Linguistics* 103. Amsterdam: Benjamins.
- R Core Team. 2022. “R: A Language and Environment for Statistical Computing.” Vienna, Austria. <https://www.R-project.org/>.
- Wilkinson, Mark D., Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, et al. 2016. “The FAIR Guiding Principles for

Scientific Data Management and Stewardship.” *Scientific Data* 3 (1): 160018. <https://doi.org/10.1038/sdata.2016.18>.