

Textbook English: A Multi-Dimensional Approach

Online Supplements

Elen Le Foll

2024-03-10

Table of contents

About	5
1 Introduction	6
1.1 Research objectives and methodological approach	6
1.2 Outline of the book	8
2 Open Science statement	9
3 Summary	11
References	12
Appendices	14
A Literature Review Data	14
B Corpus Data	16
B.1 Textbook English Corpus (TEC)	16
B.2 Reference corpora	16
B.2.1 Spoken BNC2014	16
B.2.2 Informative Texts for Teens Corpus (Info Teens)	16
B.2.3 Youth Fiction corpus	17
C Evaluation of the Multi-Feature Tagger of English (MFTE)	18
C.1 Packages required	18
C.2 Data import from evaluation files	18
C.3 Estimating MFTE accuracy for Textbook English	24
C.4 MFTE accuracy for reference corpora (or comparable corpora)	27
C.4.1 Conversation	27
C.4.2 Fiction	28
C.4.3 Informative	29
C.5 Estimating the overall MFTE accuracy for corpora used in the study	30
C.6 Exploring tagger errors	36
D Data Preparation for the Model of Intra-Textbook Variation	40
D.1 Packages required	40

D.2	Data import from MFTE output	40
D.2.1	Corpus size	44
D.3	Data preparation for PCA	44
D.3.1	Feature distributions	45
D.3.2	Feature removal	48
D.3.3	Identifying potential outlier texts	49
D.3.4	Signed log transformation	54
D.3.5	Feature correlations	60
D.4	Composition of TEC texts/files	62
E	Data Analysis for the Model of Intra-Textbook Variation	67
E.1	Packages required	67
E.2	Preparing the data for PCA	68
E.2.1	TEC data import	68
E.3	Checking the factorability of data	68
E.3.1	Removal of feature with MSAs of < 0.5	69
E.3.2	Choosing the number of principal components to retain	69
E.3.3	Excluding features with low final communalites	70
E.4	Testing the effect of rotating the components	71
E.5	Principal Component Analysis (PCA)	72
E.5.1	Using the full dataset	72
E.5.2	Using random subsets of the data	73
E.5.3	Using specific subsets of the data	73
E.5.4	Performing the PCA	73
E.6	Plotting PCA results	74
E.6.1	3D plots	74
E.7	Two-dimensional plots (biplots)	75
E.7.1	Data wrangling for PCAtools	75
E.7.2	Pairs plot	76
E.7.3	Bi-plots	78
E.8	Feature contributions (loadings) on each component	83
E.9	Exploring the dimensions of the model	89
E.10	Computing mixed-effects models of the dimension scores	91
E.10.1	Dimension 1: ‘Overt instructions and explanations’	91
E.10.2	Dimension 2: ‘Involved vs. Informational Production’	95
E.10.3	Dimension 3: ‘Narrative vs. factual discourse’	103
E.10.4	Dimension 4: ‘Informational compression vs. elaboration’	109
E.10.5	Testing model assumptions	115
F	Data Preparation for the Model of Textbook English vs. ‘real-world’ English	117
F.1	Packages required	117
F.2	Data import from MFTE outputs	118
F.2.1	Spoken BNC2014	118

F.2.2	Youth Fiction corpus	118
F.2.3	Informative Texts for Teens (InfoTeens) corpus	119
F.3	Merging TEC and reference corpora data	119
F.3.1	Corpus size	119
F.4	Data preparation for PCA	120
F.4.1	Feature distributions	120
F.4.2	Feature removal	123
F.4.3	Identifying outlier texts	124
F.4.4	Signed log transformation	131
F.4.5	Merging of data for MDA	134
F.5	Testing factorability of data	134
F.5.1	Visualisation of feature correlations	134
F.5.2	Collinearity	135
F.5.3	MSA	136
F.5.4	Scree plot	136
F.5.5	Communalities	137
G	Data Analysis for the Model of Textbook English vs. ‘real-world’ English	139
G.1	Packages required	139
G.2	Conducting the PCA	140
G.3	Plotting PCA results	141
G.3.1	3D plots	141
G.4	Two-dimensional plots (biplots)	142
G.4.1	Data wrangling for PCAtools	142
G.4.2	Pairs plot	142
G.4.3	Bi-plots	145
G.5	Feature contributions (loadings) on each component	149
G.6	Graphs of features of that contribute most to each component/dimension	152
G.7	Exploring feature contributions in terms of normalised frequencies	154
G.8	Exploring the dimensions of the model	157
G.9	Raincloud plots visualising dimension scores	160
G.10	Computing mixed-effects models of the dimension scores	163
G.10.1	Data preparation	163
G.10.2	Dimension 1: ‘Spontaneous interactional vs. Edited informational’	164
G.10.3	Dimension 2: ‘Narrative vs. Non-narrative’	167
G.10.4	Dimension 3: ‘Pedagogically adapted vs. Natural’	172
G.10.5	Dimension 4: ‘Factual vs. Speculative’ / ‘Simple vs. complex verb forms’?	178
G.10.6	Testing model assumptions	184

About



Warning

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”.¹ 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

¹Dialogue from *Speak English 6^e 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-based² 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.

²Here 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).

3 Summary



Warning

This website is a work in progress.

This page will soon feature a summary of the book.

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.
- Le Foll, Elen, and Muhammad Shakir. 2023. “Introducing a New Open-Source Corpus-Linguistic Tool: The Multi-Feature Tagger of English (MFTE),” May.
- 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>.

A Literature Review Data

This table provides a tabular overview of the studies surveyed as part of this study's literature review on textbook research examining describing and/or evaluating the language of English textbooks designed for English L2 learners in various instructional settings.

It presents the results of this 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 for which no information was published. The table is fully searchable and filterable. You can adjust the widths of the individual columns to best fit your screen size.

This list is intended to be a dynamic resource that will grow over time. If you would like to contribute any studies to the table, please either fork [the corresponding CSV file in the repository](#) or send me an [e-mail](#) with the corresponding details of the studies that you would like to add.

This page was last updated on 10 March 2024.

Search													
Author	Year	Title	Focus	Nb.TxE	Settin	TxB.level	TxB.market	TxB.p	TxB.content	Method	Ref.corpus	Reference	
Al Khateen & Almuajjewel	2018	Communicative Activities in Saudi EFL Textbooks: A Corpus-driven Analysis	speaking tasks	57	EFL	primary and secondary	local, Saudi Arabia	2016–8		corpus-driven		Al Khateeb, Ahmad, & Almuajjewel, Sultan. 2018. Communicative Activities in Saudi EFL Textbooks: A Corpus-driven Analysis. <i>Journal of Language Teaching and Research</i> , 9(6), 1301. https://doi.org/10.17507/jltr.0906.20	
Alejo González et al.	2010	Phrasal verbs in EFL course books	phrasal verbs	8	EFL	ESO3 and Bac1	local: Spain, secondary schools	2002–8		page-by-page survey, comparison with BNC data	BNC1994	Alejo González, Rafael, Ana Piquer Piriz & Guadalupe Reviriego Sierra. 2010. Phrasal verbs in EFL course books. In Sabine De Knop, Frank & Antoon De Rycker (eds.), <i>Filtering Language Teaching Materials: A Corpus Linguistic Approach</i> (Applications of Cognitive Linguistics 17), 59–77. Berlin, New York: De Gruyter Mouton. https://doi.org/10.1515/9783110245837.89	
Barbieri & Eckhardt	2007	Applying corpus-based findings to form-focused instruction: The case of reported speech	reported speech	7	ESL /EFL		international grammar textbooks	1989–2001		page-by-page survey of textbooks, corpus analysis of reference corpus		Barbieri, Federica & Suzanne EB Eckhardt. 2007. Applying Corpus-Based Findings to Form-Focused Instruction: The Case of Reported Speech. <i>Lar</i> . <i>Teaching Research</i> 11(3), 319–346. http://dx.doi.org/10.1177/1362168807077563	
Bardovi-Harlig et al.	1991	Developing pragmatic awareness: Closing the conversation	conversation closings	20	ESL	beginner-intermediate	international	"current"		page-by-page survey		Bardovi-Harlig, Kathleen, Beverly A. S. Harto, Rebecca Mahan-Taylor, Mary J. Morgan & Duke Reynolds. 1991. Developing pragmatic awareness: closing the conversation. <i>ELT Journal</i> 45(1), 4–14. https://doi.org/10.1093/elt/45.1.4	
Biber	2002	What does frequency have to do with grammar teaching?	noun premodifiers, present progressive vs. simple present	6	EFL/ ESL	intermediate-advanced	international grammar books	1986–2000		comparative corpus-driven	LGSWE (20 million words)	Biber, Douglas & Randi Reppen. 2002. What Does Frequency Have to Do with Grammar Teaching? <i>Studies in Second Language Acquisition</i> 24(02), 208. https://doi.org/10.1017/S027226310200208	
Biber	2006	University language: a corpus-based study of spoken and written registers	lexical bundles	at least 18	EAP /ESL	university	university textbooks, from a total of six broad disciplines	87 texts from the textbooks		MDA	TOEFL 2000 Spoken and Written Academic Language	Biber, Douglas. 2006. University language: a corpus-based study of spoken and written registers (S in Corpus Linguistics v. 23). Amsterdam, Philadelphia: John Benjamins.	
Biber et al.	2002	Speaking and Writing in the University: A Multidimensional Comparison	wide range of lexicogrammatical features	at least 18	EAP /ESL	university	university textbooks, from a total of six broad disciplines	87 texts from the textbooks		MDA	TOEFL 2000 Spoken and Written Academic Language	Biber, Douglas, Susan Conrad, Randi Reppen, Byrd & Marie Helt. 2002. Speaking and Writing in the University: A Multidimensional Comparison. <i>TE Quarterly</i> 36(1), 9. https://doi.org/10.2307/373586	
Bouhlal Horst, & Martini	2018	Modality in ESL Textbooks: Insights from a Contrastive Corpus-Based Analysis.	central modals	9 (3 series)	ESL	intermediate	local: Québec, upper secondary school	2007–2009		comparative corpus-based	BNC1994 and COCA	Bouhlal, Fatma, Horst, Marlise, & Martini, Julia. 2018. Modality in ESL Textbooks: Insights from Contrastive Corpus-Based Analysis. <i>The Cana Modern Language Review</i> , 74(2), 227–252. https://doi.org/10.3138/cmlr.3073	
Boxer & Pickering	1995	Problems in the presentation of speech acts in ELT materials: the case of complaints	speech acts of complaints	7	EFL/ ESL	intermediate-advanced	international	1981–1991		page-by-page analysis		Boxer, Diana & Lucy Pickering. 1995. Problems in the presentation of Speech Acts in ELT Materials: the Case of Complaints. <i>ELT Journal</i> 49(1), 44–58. https://doi.org/10.1093/elt/49.1.44	
Cane	1998	Teaching conversation skills more effectively	lexico-grammatical features of conversation		EFL		international	"recent"		page-by-page survey		Cane, Graeme. 1998. Teaching Conversation More Effectively. <i>The Korea TESOL Journal</i> 51, 31–37.	

1–10 of 86 rows Show 10 ▾

Previous 1 2 3 4 5 ... 9 Next

The raw data can be downloaded as a comma-separated file from [the project GitHub repository](#).

B Corpus Data

B.1 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.

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](#).

B.2 Reference corpora

B.2.1 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 into the format used in this study (the “John and Jill in Ivybridge” version with added full stops at speaker turns, as explained in Section 4.3.2.2 of the book) can be found here: https://github.com/elenlefol/TextbookEnglish/blob/main/3_Data/BNCspoken_nomark-up_JackJill.R

B.2.2 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](#).

B.2.3 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. 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](#).

C Evaluation of the Multi-Feature Tagger of English (MFTE)

For more information on the tagger itself, as well as the evaluation data and methods, see Le Foll (2021a) and <https://github.com/elenlefoll/MultiFeatureTaggerEnglish>.

💡 Using the MFTE

The Multi-Feature Tagger of English (MFTE) is free to use and was released under an Open Source licence. If you are interested in using the MFTE for your own project, I recommend using the latest version of the MFTE Python, which is much easier to use, can tag many more features, and also underwent a thorough evaluation. To find out more, see Le Foll & Shakir (2023) and <https://github.com/mshakirDr/MFTE>.

C.1 Packages required

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

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

library(caret) # For computing confusion matrices
library(harrypotter) # Only for colour scheme
library(here) # For path management
library(knitr) # Loaded to display the tables using the kable() function
library(paletteer) # For nice colours
library(readxl) # For the direct import of Excel files
library(tidyverse) # For everything else!
```

C.2 Data import from evaluation files

The data is imported directly from the Excel files in which the manual tag check and corrections was performed. A number of data wrangling steps need to be made for the data to be converted

to a tidy format.

```
# Function to import and wrangle the evaluation data from the Excel files in
# which the manual evaluation was conducted
importEval3 <- function(file, fileID, register, corpus) {
  Tag1 <- file |>
    add_column(FileID = fileID, Register = register, Corpus = corpus) |>
    select(FileID, Corpus, Register, Output, Tokens, Tag1, Tag1Gold) |>
    rename(Tag = Tag1, TagGold = Tag1Gold, Token = Tokens) |>
    mutate(Evaluation = ifelse(is.na(TagGold), TRUE, FALSE)) |>
    mutate(TagGold = ifelse(is.na(TagGold), as.character(Tag),
                           as.character(TagGold))) |>
    filter(!is.na(Tag)) |>
    mutate_if(is.character, as.factor)

  Tag2 <- file |>
    add_column(FileID = fileID, Register = register, Corpus = corpus) |>
    select(FileID, Corpus, Register, Output, Tokens, Tag2, Tag2Gold) |>
    rename(Tag = Tag2, TagGold = Tag2Gold, Token = Tokens) |>
    mutate(Evaluation = ifelse(is.na(TagGold), TRUE, FALSE)) |>
    mutate(TagGold = ifelse(is.na(TagGold), as.character(Tag),
                           as.character(TagGold))) |>
    filter(!is.na(Tag)) |>
    mutate_if(is.character, as.factor)

  Tag3 <- file |>
    add_column(FileID = fileID, Register = register, Corpus = corpus) |>
    select(FileID, Corpus, Register, Output, Tokens, Tag3, Tag3Gold) |>
    rename(Tag = Tag3, TagGold = Tag3Gold, Token = Tokens) |>
    mutate(Evaluation = ifelse(is.na(TagGold), TRUE, FALSE)) |>
    mutate(TagGold = ifelse(is.na(TagGold), as.character(Tag),
                           as.character(TagGold))) |>
    filter(!is.na(Tag)) |>
    mutate_if(is.character, as.factor)

  output <- rbind(Tag1, Tag2, Tag3) |>
    mutate(across(where(is.factor), str_remove_all, pattern = fixed(" "))) |> # 
      # Removes all white spaces which are found in the excel files
    filter(!is.na(Output)) |>
    mutate_if(is.character, as.factor)
}
```

```

# Second function to import and wrangle the evaluation data for Excel files
  ↵ with four tag columns as opposed to three
importEval4 <- function(file, fileID, register, corpus) {
  Tag1 <- file |>
    add_column(FileID = fileID, Register = register, Corpus = corpus) |>
    select(FileID, Corpus, Register, Output, Tokens, Tag1, Tag1Gold) |>
    rename(Tag = Tag1, TagGold = Tag1Gold, Token = Tokens) |>
    mutate(Evaluation = ifelse(is.na(TagGold), TRUE, FALSE)) |>
    mutate(TagGold = ifelse(is.na(TagGold), as.character(Tag),
      ↵ as.character(TagGold))) |>
    filter(!is.na(Tag)) |>
    mutate_if(is.character, as.factor)

  Tag2 <- file |>
    add_column(FileID = fileID, Register = register, Corpus = corpus) |>
    select(FileID, Corpus, Register, Output, Tokens, Tag2, Tag2Gold) |>
    rename(Tag = Tag2, TagGold = Tag2Gold, Token = Tokens) |>
    mutate(Evaluation = ifelse(is.na(TagGold), TRUE, FALSE)) |>
    mutate(TagGold = ifelse(is.na(TagGold), as.character(Tag),
      ↵ as.character(TagGold))) |>
    filter(!is.na(Tag)) |>
    mutate_if(is.character, as.factor)

  Tag3 <- file |>
    add_column(FileID = fileID, Register = register, Corpus = corpus) |>
    select(FileID, Corpus, Register, Output, Tokens, Tag3, Tag3Gold) |>
    rename(Tag = Tag3, TagGold = Tag3Gold, Token = Tokens) |>
    mutate(Evaluation = ifelse(is.na(TagGold), TRUE, FALSE)) |>
    mutate(TagGold = ifelse(is.na(TagGold), as.character(Tag),
      ↵ as.character(TagGold))) |>
    filter(!is.na(Tag)) |>
    mutate_if(is.character, as.factor)

  Tag4 <- file |>
    add_column(FileID = fileID, Register = register, Corpus = corpus) |>
    select(FileID, Corpus, Register, Output, Tokens, Tag4, Tag4Gold) |>
    rename(Tag = Tag4, TagGold = Tag4Gold, Token = Tokens) |>
    mutate(Evaluation = ifelse(is.na(TagGold), TRUE, FALSE)) |>
    mutate(TagGold = ifelse(is.na(TagGold), as.character(Tag),
      ↵ as.character(TagGold))) |>
    filter(!is.na(Tag)) |>
    mutate_if(is.character, as.factor)

```

```

output <- rbind(Tag1, Tag2, Tag3, Tag4) |>
  mutate(across(where(is.factor), str_remove_all, pattern = fixed(" "))) |> #
  ↵ Removes all white spaces which are found in the excel files
  filter(!is.na(Tag)) |>
  mutate_if(is.character, as.factor)

}

# Function to decide which of the two above functions should be used
importEval <- function(file, fileID, register, corpus) {
  if(sum(!is.na(file$Tag4)) > 0) {
    output = importEval4(file = file, fileID = fileID, register = register,
    ↵ corpus = corpus)
  }
  else{
    output = importEval3(file = file, fileID = fileID, register = register,
    ↵ corpus = corpus)
  }
}

Solutions_Intermediate_Spoken_0032 <- importEval(file =
  ↵ read_excel(here("data", "MFTE", "evaluation",
  ↵ "Solutions_Intermediate_Spoken_0032_Evaluation.xlsx")), fileID =
  ↵ "Solutions_Intermediate_Spoken_0032", register = "Conversation", corpus =
  ↵ "TEC-Sp")

HT_5_Poetry_0001 <- importEval(file = read_excel(here("data", "MFTE",
  ↵ "evaluation", "HT_5_Poetry_0001_Evaluation.xlsx")), fileID =
  ↵ "HT_5_Poetry_0001", register = "Poetry", corpus = "TEC-Fr")

Achievers_A1_Informative_0006 <- importEval(file = read_excel(here("data",
  ↵ "MFTE", "evaluation", "Achievers_A1_Informative_0006_Evaluation.xlsx")),
  ↵ fileID = "Achievers_A1_Informative_0006", register = "Informative",
  ↵ corpus = "TEC-Sp")

New_GreenLine_5_Personal_0003 <- importEval(file = read_excel(here("data",
  ↵ "MFTE", "evaluation", "New_GreenLine_5_Personal_0003_Evaluation.xlsx")),
  ↵ fileID = "New_GreenLine_5_Personal_0003", register = "Personal
  ↵ communication", corpus = "TEC-Ger")

Piece_of_cake_3e_Instructional_0006 <- importEval(file =
  ↵ read_excel(here("data", "MFTE", "evaluation",
  ↵ "Piece_of_cake_3e_Instructional_0006_Evaluation.xlsx")), fileID =
  ↵ "Piece_of_cake_3e_Instructional_0006", register = "Instructional", corpus
  ↵ = "TEC-Fr")

```

```

Access_4_Narrative_0006 <- importEval(file = read_excel(here("data", "MFTE",
  ↵ "evaluation", "Access_4_Narrative_0006_Evaluation.xlsx")), fileID =
  ↵ "Access_4_Narrative_0006", register = "Fiction", corpus = "TEC-Ger")

BNCBFict_b2 <- importEval(file = read_excel(here("data", "MFTE",
  ↵ "evaluation", "BNCBFict_b2.xlsx")), fileID = "BNCBFict_b2", register =
  ↵ "fiction", corpus = "BNC2014")

BNCBFict_m54 <- importEval(file = read_excel(here("data", "MFTE",
  ↵ "evaluation", "BNCBFict_m54.xlsx")), fileID = "BNCBFict_m54", register =
  ↵ "fiction", corpus = "BNC2014")

BNCBFict_e27 <- importEval(file = read_excel(here("data", "MFTE",
  ↵ "evaluation", "BNCBFict_e27.xlsx")), fileID = "BNCBFict_e27", register =
  ↵ "fiction", corpus = "BNC2014")

BNCBMass16 <- importEval(file = read_excel(here("data", "MFTE", "evaluation",
  ↵ "BNCBMass16.xlsx")), fileID = "BNCBMass16", register = "news", corpus =
  ↵ "BNC2014")

BNCBMass23 <- importEval(file = read_excel(here("data", "MFTE", "evaluation",
  ↵ "BNCBMass23.xlsx")), fileID = "BNCBMass23", register = "news", corpus =
  ↵ "BNC2014")

BNCBReg111 <- importEval(file = read_excel(here("data", "MFTE", "evaluation",
  ↵ "BNCBReg111.xlsx")), fileID = "BNCBReg111", register = "news", corpus =
  ↵ "BNC2014")

BNCBReg750 <- importEval(file = read_excel(here("data", "MFTE", "evaluation",
  ↵ "BNCBReg750.xlsx")), fileID = "BNCBReg750", register = "news", corpus =
  ↵ "BNC2014")

BNCBSer486 <- importEval(file = read_excel(here("data", "MFTE", "evaluation",
  ↵ "BNCBSer486.xlsx")), fileID = "BNCBSer486", register = "news", corpus =
  ↵ "BNC2014")

BNCBSer562 <- importEval(file = read_excel(here("data", "MFTE", "evaluation",
  ↵ "BNCBSer562.xlsx")), fileID = "BNCBSer562", register = "news", corpus =
  ↵ "BNC2014")

BNCBEB18 <- importEval(file = read_excel(here("data", "MFTE", "evaluation",
  ↵ "BNCBEB18.xlsx")), fileID = "BNCBEB18", register = "internet", corpus =
  ↵ "BNC2014")

```

```

BNCBEFor32 <- importEval(file = read_excel(here("data", "MFTE", "evaluation",
  ↵  "BNCBEFor32.xlsx")), fileID = "BNCBEFor32", register = "internet", corpus
  ↵  = "BNC2014")

S2DD <- importEval(file = read_excel(here("data", "MFTE", "evaluation",
  ↵  "S2DD.xlsx")), fileID = "S2DD", register = "spoken", corpus = "BNC2014")

S3AV <- importEval(file = read_excel(here("data", "MFTE", "evaluation",
  ↵  "S3AV.xlsx")), fileID = "S3AV", register = "spoken", corpus = "BNC2014")

SEL5 <- importEval(file = read_excel(here("data", "MFTE", "evaluation",
  ↵  "SEL5.xlsx")), fileID = "SEL5", register = "spoken", corpus = "BNC2014")

SVLK <- importEval(file = read_excel(here("data", "MFTE", "evaluation",
  ↵  "SVLK.xlsx")), fileID = "SVLK", register = "spoken", corpus = "BNC2014")

SZXQ <- importEval(file = read_excel(here("data", "MFTE", "evaluation",
  ↵  "SZXQ.xlsx")), fileID = "SZXQ", register = "spoken", corpus = "BNC2014")

TaggerEval <- rbind(Solutions_Intermediate_Spoken_0032, HT_5_Poetry_0001,
  ↵  Achievers_A1_Informative_0006, New_GreenLine_5_Personal_0003,
  ↵  Piece_of_cake_3e_Instructional_0006, Access_4_Narrative_0006, BNCBEB18,
  ↵  BNCFict_b2, BNCFict_m54, BNCFict_e27, BNCBEFor32, BNCBMass16,
  ↵  BNCBMass23, BNCBReg111, BNCBReg750, BNCBSer486, BNCBSer562, S2DD, S3AV,
  ↵  SEL5, SVLK, SZXQ)

```

Some tags had to be merged to account for changes made to the MFTE between the evaluation and the tagging of the corpora included in the present study.

```

TaggerEval <- TaggerEval |>
  mutate(Tag = ifelse(Tag == "PHC", "CC", as.character(Tag))) |>
  mutate(TagGold = ifelse(TagGold == "PHC", "CC", as.character(TagGold))) |>
  mutate(Tag = ifelse(Tag == "QLIKE", "LIKE", as.character(Tag))) |>
  mutate(TagGold = ifelse(TagGold == "QLIKE", "LIKE", as.character(TagGold))) |
    |>
  mutate(Tag = ifelse(Tag == "TO", "IN", as.character(Tag))) |>
  mutate(TagGold = ifelse(TagGold == "TO", "IN", as.character(TagGold))) |>
  mutate_if(is.character, as.factor) |>
  mutate(Evaluation = ifelse(as.character(Tag) == as.character(TagGold),
    ↵  TRUE, FALSE))

```

```

# head(TaggerEval) # Check sanity of data
# summary(TaggerEval) # Check sanity of data

# saveRDS(TaggerEval, here("data", "processed",
#   "MFTE_Evaluation_Results.rds"))

# write.csv(TaggerEval, here("data", "processed",
#   "MFTE_Evaluation_Results.csv"))

```

This table provides a summary of the complete evaluation dataset. It comprises 25,233 tags that were checked (and, if needs be, corrected) by at least one human annotator. This number includes tags for punctuation marks, which make up a considerable proportion of the tags.

FileID	Corpus	Register	Output
BNCBFict_b2 : 2621	TEC-Sp : 1042	fiction :6500	._. : 1156
BNCBFict_e27: 2104	TEC-Fr : 2058	news :6312	the_DT : 820
BNCBFict_m54: 1775	TEC-Ger: 1415	spoken :6047	,_,: 720
BNCBMass16 : 1619	BNC2014:20718	internet :1859	a_DT : 466
SEL5 : 1463		Instructional:1048	of_IN : 328
BNCBEFor32 : 1305		Poetry :1010	(Other):21742
(Other) :14346		(Other) :2457	NA's : 1
Token	Tag	TagGold	Evaluation
.	: 1156	NN : 4415	NN : 4328 Mode :logical
the	: 820	IN : 2145	IN : 2113 FALSE:832
,	: 720	DT : 1454	DT : 1457 TRUE :24401
to	: 495	.	: 1367 : 1367
's	: 493	VPRT : 1044	VPRT : 1054
(Other):21547	VBD : 899	VBD : 895	
NA's	: 2	(Other):13909	(Other):14019

C.3 Estimating MFTE accuracy for Textbook English

In total, 4,515 tags from the TEC were manually checked. This chunk calculates the recall and precision rates of each feature, ignoring all punctuation and symbols.

```

data <- TaggerEval |>
  filter(Corpus %in% c("TEC-Fr", "TEC-Ger", "TEC-Sp")) |>
  filter(TagGold != "UNCLEAR") |>
  filter(Tag %in% c(str_extract(Tag, "[A-Z0-9]+")))) |> # Remove punctuation
  ↵ tags which are uninteresting here.

```

```

filter(Tag != "SYM" & Tag != "``") |>
droplevels() |>
mutate(Tag = factor(Tag, levels = union(levels(Tag), levels(TagGold)))) |>
  # Ensure that the factor levels are the same for the next caret
  # operation
mutate(TagGold = factor(TagGold, levels = union(levels(Tag),
  levels(TagGold)))) |>

# Spot gold tag corrections that are not actually errors (should return zero
# rows if all is well)
# data[data$Tag==data$TagGold & data$Evaluation == FALSE,] |> as.data.frame()

```

The breakdown of inaccurate vs. accurate tags in this TEC evaluation sample is:

Mode	FALSE	TRUE
logical	114	3831

Note that the following accuracy metrics calculated using the `caret::confusionMatrix` are not very representative because they include tags, which were not entered in the study, e.g., LS and FW.

Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull
0.97	0.97	0.97	0.98	0.20
AccuracyPValue	McnemarPValue			
0.00	NaN			

Accuracy metrics per feature are more interesting and relevant.

	Precision	Recall	F1
Class: ABLE	1.00	1.00	1.00
Class: ACT	0.97	0.98	0.98
Class: AMP	1.00	1.00	1.00
Class: ASPECT	1.00	1.00	1.00
Class: BEMA	1.00	1.00	1.00
Class: CAUSE	1.00	1.00	1.00
Class: CC	1.00	0.99	1.00
Class: CD	0.95	0.95	0.95
Class: COMM	1.00	0.98	0.99
Class: COND	1.00	1.00	1.00

	Precision	Recall	F1
Class: CONT	1.00	1.00	1.00
Class: CUZ	1.00	1.00	1.00
Class: DEMO	0.97	0.97	0.97
Class: DMA	1.00	1.00	1.00
Class: DOAUX	0.86	1.00	0.92
Class: DT	1.00	1.00	1.00
Class: DWNT	0.67	1.00	0.80
Class: ELAB	1.00	1.00	1.00
Class: EMPH	0.83	1.00	0.91
Class: EX	1.00	1.00	1.00
Class: EXIST	1.00	1.00	1.00
Class: FPP1P	1.00	1.00	1.00
Class: FPP1S	1.00	1.00	1.00
Class: FPUH	1.00	1.00	1.00
Class: FREQ	1.00	1.00	1.00
Class: FW	0.10	1.00	0.18
Class: GTO	1.00	1.00	1.00
Class: HDG	1.00	1.00	1.00
Class: HGOT	1.00	1.00	1.00
Class: IN	1.00	1.00	1.00
Class: JJ	0.96	0.98	0.97
Class: JPRED	0.97	0.90	0.94
Class: LIKE	0.83	1.00	0.91
Class: MDCA	1.00	1.00	1.00
Class: MDCO	1.00	1.00	1.00
Class: MDMM	1.00	0.67	0.80
Class: MDNE	1.00	0.80	0.89
Class: MDWO	1.00	1.00	1.00
Class: MDWS	1.00	1.00	1.00
Class: MENTAL	0.99	0.99	0.99
Class: NCOMP	0.88	1.00	0.94
Class: NN	0.95	0.99	0.97
Class: NULL	1.00	0.08	0.14
Class: OCCUR	0.94	1.00	0.97
Class: PASS	0.89	0.89	0.89
Class: PEAS	1.00	0.87	0.93
Class: PGET	1.00	1.00	1.00
Class: PIT	1.00	1.00	1.00
Class: PLACE	1.00	0.83	0.91
Class: POLITE	1.00	1.00	1.00
Class: POS	1.00	1.00	1.00

	Precision	Recall	F1
Class: PROG	1.00	0.89	0.94
Class: QUAN	0.96	0.98	0.97
Class: QUPR	1.00	1.00	1.00
Class: RB	1.00	0.99	0.99
Class: RP	1.00	1.00	1.00
Class: SO	1.00	0.64	0.78
Class: SPLIT	1.00	1.00	1.00
Class: SPP2	1.00	1.00	1.00
Class: STPR	0.60	1.00	0.75
Class: THATD	0.86	1.00	0.92
Class: THRC	1.00	0.71	0.83
Class: THSC	0.69	1.00	0.82
Class: TIME	1.00	0.97	0.98
Class: TPP3P	1.00	1.00	1.00
Class: TPP3S	1.00	1.00	1.00
Class: VB	0.94	0.94	0.94
Class: VBD	0.97	0.99	0.98
Class: VBG	0.96	1.00	0.98
Class: VBN	0.85	0.92	0.88
Class: VIMP	0.99	0.88	0.93
Class: VPRT	0.98	0.98	0.98
Class: WHQU	0.97	1.00	0.98
Class: WHSC	1.00	0.97	0.99
Class: XX0	1.00	1.00	1.00
Class: YNQU	1.00	1.00	1.00
Class: OCR	NA	0.00	NA

C.4 MFTE accuracy for reference corpora (or comparable corpora)

C.4.1 Conversation

These are extracts from the Spoken BNC2014 (as entered in the study). The evaluation data for this sample excludes 7 tokens deemed *unclear* by at least one human annotator.

```
data <- TaggerEval |>
  filter(Register == "spoken") |>
  filter(TagGold != "UNCLEAR") |>
  filter(Tag %in% c(str_extract(Tag, "[A-Z0-9]+")))) |> # Remove all
  ↵ punctuation tags which are uninteresting here.
```

```

droplevels() |>
  mutate(Tag = factor(Tag, levels = union(levels(Tag), levels(TagGold)))) |>
    # Ensure that the factor levels are the same for the next caret
    # operation
  mutate(TagGold = factor(TagGold, levels = union(levels(Tag),
    levels(TagGold)))) |>

# Spot gold tag corrections that are not actually errors (should return zero
# rows if all is well)
# data[data$Tag==data$TagGold & data$Evaluation == FALSE,] |> as.data.frame()

```

The breakdown of inaccurate vs. accurate tags in this evaluation sample is:

Mode	FALSE	TRUE
logical	224	5388

Note that the following accuracy metrics calculated using the `caret::confusionMatrix` are not very representative because they include tags, which were not entered in the study, e.g., LS and FW.

Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull
0.96	0.96	0.95	0.97	0.12
AccuracyPValue	McNemarPValue			
0.00	NaN			

C.4.2 Fiction

The evaluation data for this sample excludes 0 tokens deemed *unclear* by at least one human annotator.

```

data <- TaggerEval |>
  filter(Register == "fiction") |>
  filter(TagGold != "UNCLEAR") |>
  filter(Tag %in% c(str_extract(Tag, "[A-Z0-9]+")))) |> # Remove all
    # punctuation tags which are uninteresting here.
  filter(Tag != "SYM" & Tag != "``") |>
  droplevels() |>
  mutate(Tag = factor(Tag, levels = union(levels(Tag), levels(TagGold)))) |>
    # Ensure that the factor levels are the same for the next caret
    # operation

```

```

  mutate(TagGold = factor(TagGold, levels = union(levels(Tag),
    ↵  levels(TagGold)))))

# Spot gold tag corrections that are not actually errors (should return zero
←  rows if all is well)
# data[data$Tag==data$TagGold & data$Evaluation == FALSE,] |> as.data.frame()

```

The breakdown of inaccurate vs. accurate tags in this evaluation sample is:

Mode	FALSE	TRUE
logical	168	5346

Note that the following accuracy metrics calculated using the `caret::confusionMatrix` are not very representative because they include tags, which were not entered in the study, e.g., LS and FW.

Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull
0.97	0.97	0.96	0.97	0.19
AccuracyPValue	McnemarPValue			
0.00	NaN			

C.4.3 Informative

The evaluation data for this sample excludes 8 tokens deemed *unclear* by at least one human annotator.

```

data <- TaggerEval |>
  filter(Register == "news" | FileID %in% c("BNCBEFor32", "BNCBEB18")) |>
  filter(TagGold != "UNCLEAR") |>
  filter(Tag %in% str_extract(Tag, "[A-Z0-9]+")) |> # Remove all
  ← punctuation tags which are uninteresting here.
  filter(Tag != "SYM" & Tag != "``") |>
  droplevels() |>
  mutate(Tag = factor(Tag, levels = union(levels(Tag), levels(TagGold)))) |>
  ← # Ensure that the factor levels are the same for the next caret
  ← operation
  mutate(TagGold = factor(TagGold, levels = union(levels(Tag),
  ← levels(TagGold))))

```

```
# Spot gold tag corrections that are not actually errors (should return zero
→ rows if all is well)
# data[data$Tag==data$TagGold & data$Evaluation == FALSE,] |> as.data.frame()
```

The breakdown of inaccurate vs. accurate tags in this evaluation sample is:

Mode	FALSE	TRUE
logical	309	7113

Note that the following accuracy metrics calculated using the `caret::confusionMatrix` are not very representative because they include tags, which were not entered in the study, e.g., LS and FW.

Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull
0.96	0.95	0.95	0.96	0.24
AccuracyPValue	McnemarPValue			
0.00	NaN			

C.5 Estimating the overall MFTE accuracy for corpora used in the study

```
data <- TaggerEval |>
  filter(TagGold != "UNCLEAR") |>
  filter(Tag %in% c(str_extract(Tag, "[A-Z0-9]+")))) |> # Remove all
  → punctuation tags which are uninteresting here.
  filter(Tag != "SYM" & Tag != "``") |>
  filter(TagGold != "SYM" & TagGold != "``") |>
  droplevels() |>
  mutate(Tag = factor(Tag, levels = union(levels(Tag), levels(TagGold)))) |>
  → # Ensure that the factor levels are the same for the next caret
  → operation
  mutate(TagGold = factor(TagGold, levels = union(levels(Tag),
  → levels(TagGold)))) |>

# Generate a better formatted results table for export: recall, precision and
→ f1
confusion_matrix <- cm$table
```

```

total <- sum(confusion_matrix)
number_of_classes <- nrow(confusion_matrix)
correct <- diag(confusion_matrix)
# sum all columns
total_actual_class <- apply(confusion_matrix, 2, sum)
# sum all rows
total_pred_class <- apply(confusion_matrix, 1, sum)
# Precision = TP / all that were predicted as positive
precision <- correct / total_pred_class
# Recall = TP / all that were actually positive
recall <- correct / total_actual_class
# F1
f1 <- (2 * precision * recall) / (precision + recall)
# create data frame to output results
results <- data.frame(precision, recall, f1, total_actual_class)

results |>
  kable(digits = 2)

```

	precision	recall	f1	total_actual_class
ACT	0.92	0.99	0.95	177
AMP	1.00	0.94	0.97	16
ASPECT	1.00	1.00	1.00	23
BEMA	0.99	0.99	0.99	111
CAUSE	1.00	1.00	1.00	18
CC	1.00	0.99	0.99	254
CD	0.99	0.98	0.98	134
COMM	1.00	1.00	1.00	88
CONC	0.90	0.82	0.86	11
COND	1.00	1.00	1.00	17
CONT	0.96	1.00	0.98	54
CUZ	1.00	0.90	0.95	10
DEMO	1.00	0.96	0.98	51
DMA	0.50	0.40	0.44	5
DOAUX	0.92	0.92	0.92	25
DT	1.00	1.00	1.00	490
DWNT	1.00	1.00	1.00	5
ELAB	1.00	1.00	1.00	3
EMPH	0.98	0.95	0.96	43
EX	1.00	1.00	1.00	15
EXIST	0.96	1.00	0.98	27

	precision	recall	f1	total_actual_class
FPP1P	1.00	1.00	1.00	49
FPP1S	1.00	1.00	1.00	59
FPUH	1.00	0.67	0.80	3
FREQ	1.00	1.00	1.00	15
FW	0.29	0.40	0.33	5
GTO	1.00	1.00	1.00	4
HDG	1.00	1.00	1.00	5
IN	0.99	1.00	0.99	836
JJAT	0.94	0.87	0.90	360
JJPR	0.92	0.74	0.82	108
LIKE	1.00	1.00	1.00	9
MDCA	1.00	1.00	1.00	12
MDCO	1.00	1.00	1.00	12
MDMM	1.00	1.00	1.00	1
MDNE	1.00	0.95	0.98	22
MDWO	1.00	1.00	1.00	20
MDWS	1.00	1.00	1.00	31
MENTAL	0.98	1.00	0.99	106
NCOMP	0.92	0.99	0.96	171
NN	0.96	0.98	0.97	1805
OCCUR	1.00	1.00	1.00	11
PASS	0.92	0.92	0.92	79
PEAS	1.00	0.91	0.96	70
PGET	1.00	0.67	0.80	6
PIT	1.00	0.96	0.98	78
PLACE	0.86	1.00	0.93	19
POLITE	1.00	1.00	1.00	7
POS	0.98	0.96	0.97	46
PROG	0.92	0.88	0.90	40
PRP	0.00	0.00	NaN	1
QUAN	0.96	1.00	0.98	80
QUPR	1.00	1.00	1.00	21
RB	0.96	0.95	0.96	137
RP	1.00	0.82	0.90	44
SO	1.00	0.89	0.94	9
SPLIT	1.00	1.00	1.00	40
SPP2	1.00	1.00	1.00	53
STPR	0.50	1.00	0.67	2
THATD	0.85	1.00	0.92	11
THRC	1.00	0.50	0.67	8
THSC	0.85	1.00	0.92	34

	precision	recall	f1	total_actual_class
TIME	0.95	0.98	0.96	40
TPP3P	1.00	1.00	1.00	61
TPP3S	1.00	1.00	1.00	108
URL	1.00	1.00	1.00	1
USEDTO	0.00	NaN	NaN	0
VB	0.90	0.93	0.91	258
VBD	0.96	0.97	0.97	215
VBG	0.91	0.91	0.91	111
VBN	0.42	1.00	0.59	22
VIMP	0.71	0.34	0.47	29
VPRT	0.95	0.95	0.95	351
WHQU	1.00	0.44	0.62	9
WHSC	0.95	1.00	0.97	95
XX0	1.00	0.97	0.99	76
YNQU	0.00	NaN	NaN	0
NULL	NaN	0.00	NaN	38
SYM	NaN	0.00	NaN	1
"	NaN	0.00	NaN	1

```

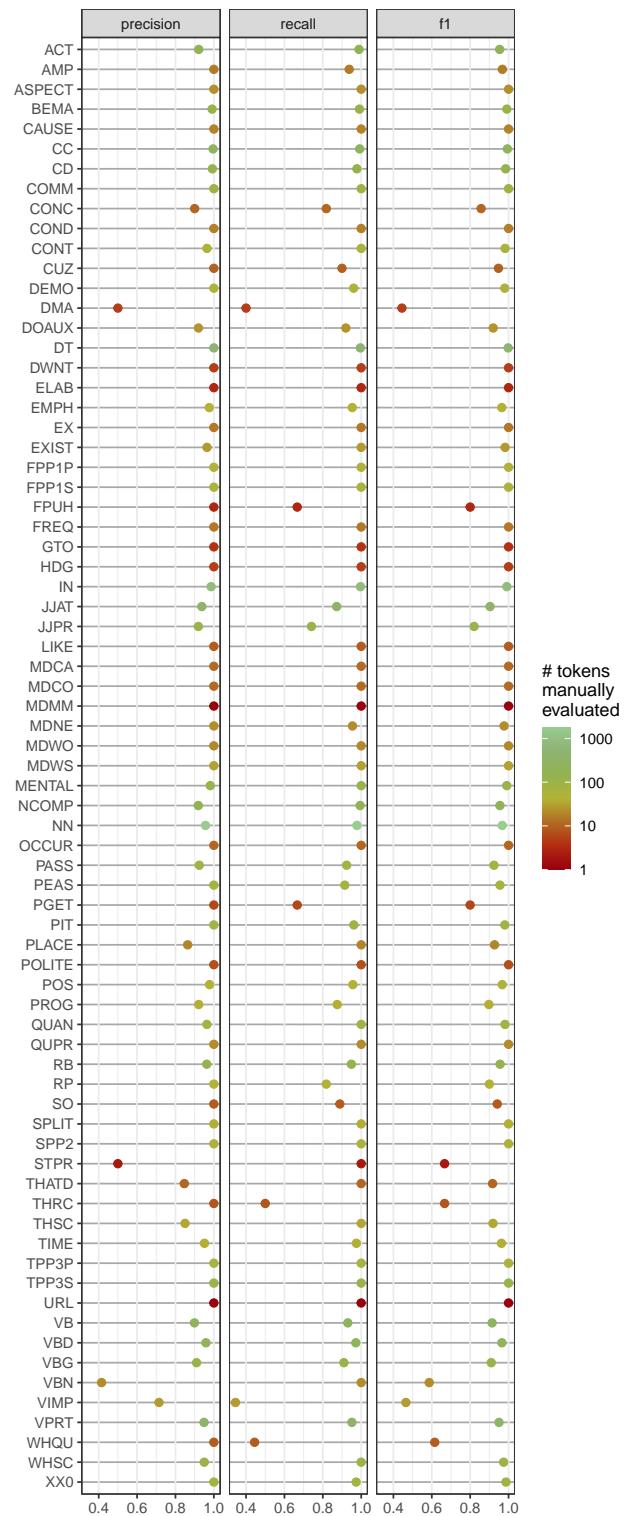
resultslong <- results |>
  drop_na() %>%
  mutate(tag = row.names(.)) |>
  filter(tag != "NULL" & tag != "SYM" & tag != "OCR" & tag != "FW" & tag !=
    "USEDTO") |>
  rename(n = total_actual_class) |>
  pivot_longer(cols = c("precision", "recall", "f1"), names_to = "metric",
    values_to = "value") |>
  mutate(metric = factor(metric, levels = c("precision", "recall", "f1")))

# summary(resultslong$n)

ggplot(resultslong, aes(y = reorder(tag, desc(tag)), x = value, group =
  metric, colour = n)) +
  geom_point(size = 2) +
  ylab("") +
  xlab("") +
  facet_wrap(~ metric) +
  scale_color_paletteer_c("harrypotter::harrypotter", trans = "log", breaks =
    c(1,10, 100, 1000), labels = c(1,10, 100, 1000), name = "# tokens
    \nmanually\nevaluated") +

```

```
theme_bw() +  
  theme(panel.grid.major.y = element_line(colour = "darkgrey")) +  
  theme(legend.position = "right")
```



```
#ggsave(here("plots", "TaggerAccuracyPlot.svg"), width = 7, height = 12)
```

C.6 Exploring tagger errors

To inspect regular/systematic tagger errors, we add an error tag with the incorrectly assigned tag and underscore and then the correct “gold” label.

```
errors <- TaggerEval |>
  filter(Evaluation=="FALSE") |>
  filter(TagGold != "UNCLEAR") |>
  mutate(Error = paste(Tag, TagGold, sep = " -> "))

FreqErrors <- errors |>
  #filter(Corpus %in% c("TEC-Fr", "TEC-Ger", "TEC-Sp")) |>
  count(Error) |>
  arrange(desc(n))

# Number of error types that only occur once
once <- FreqErrors |>
  filter(n == 1) |>
  nrow()
```

The total number of errors is 817. Of those, 94 occur just once. In total, there are 198 different types of errors. The most frequent 10 are:

```
FreqErrors |>
  filter(n > 10) |>
  kable(digits = 2)
```

Error	n
NCOMP -> NULL	37
NN -> JJAT	35
JJAT -> NN	27
NN -> VB	27
IN -> RP	25
NN -> VPRT	24
VB -> NN	22
THSC -> DEMO	19

Error	n
VB -> VIMP	19
NN -> OCR	16
VBN -> JJAT	16
ACT -> NULL	15
THATD -> NULL	15
CD -> NN	12
MENTAL -> NULL	12
NN -> VBG	11
NN -> VIMP	11
THSC -> THRC	11
VBG -> PROG	11
VBN -> JJPR	11

The code in the following chunk can be used to take a closer look at specific types of frequent errors.

```
errors |>
  filter(Error == "NN -> JJAT") |>
  select(-Output, -Corpus, -Tag, -TagGold) |>
  filter(grepl(x = Token, pattern = "[A-Z]+.")) |>
  kable(digits = 2)
```

FileID	Register	Token	Evaluation	Error
BNCBEFor32	internet	Intermediate	FALSE	NN -> JJAT
BNCBMass16	news	FINAL	FALSE	NN -> JJAT
BNCBMass16	news	Big	FALSE	NN -> JJAT
BNCBReg111	news	Scottish	FALSE	NN -> JJAT
BNCBReg111	news	Scottish	FALSE	NN -> JJAT
BNCBReg111	news	Mental	FALSE	NN -> JJAT
BNCBReg111	news	Scottish	FALSE	NN -> JJAT
BNCBReg111	news	Central	FALSE	NN -> JJAT
BNCBReg750	news	English	FALSE	NN -> JJAT
BNCBReg750	news	Natural	FALSE	NN -> JJAT
BNCBReg750	news	European	FALSE	NN -> JJAT
BNCBReg750	news	Christian	FALSE	NN -> JJAT
BNCBReg750	news	Social	FALSE	NN -> JJAT
BNCBReg750	news	Common	FALSE	NN -> JJAT
BNCBSer486	news	Northern	FALSE	NN -> JJAT

FileID	Register	Token	Evaluation	Error
BNCSer486	news	Northern	FALSE	NN -> JJAT
BNCSer486	news	Northern	FALSE	NN -> JJAT
BNCSer562	news	United	FALSE	NN -> JJAT
BNCSer562	news	White	FALSE	NN -> JJAT
BNCSer562	news	Untold	FALSE	NN -> JJAT
BNCSer562	news	New	FALSE	NN -> JJAT
SEL5	spoken	Black	FALSE	NN -> JJAT

```
errors |>
  filter(Error %in% c("NN -> VB", "VB -> NN", "NN -> VPRT", "VPRT -> NN")) |>
  count(Token) |>
  arrange(desc(n)) |>
  filter(n > 1) |>
  kable(digits = 2)
```

Token	n
mince	5
build	4
win	4
hunt	3
wags	3
throw	2
look	2
swamp	2
stop	2
defeats	2

```
errors |>
  filter(Error == "ACT -> NULL") |>
  count(Token) |>
  arrange(desc(n)) |>
  kable(digits = 2)
```

Token	n
win	3
throw	2

Token	n
lost	2
left	1
waiting	1
working	1
running	1
done	1
fixed	1
Play	1
reached	1

For more information on the MFTE evaluation, see (Le Foll 2021a) and <https://github.com/elenlefoll/MultiFeatureTaggerEnglish>.

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).

D.1 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
```

D.2 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",
← "TxB900MDA_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	3.0 GHz	GHz	Processor clock speed
System B	Processor	Core Temperature	42°C	°C	Processor core temperature
System B	Processor	Power Consumption	110W	W	Processor power consumption
System B	Processor	Fan Speed	1400 RPM	RPM	Processor fan speed
System B	Memory	RAM Capacity	14GB	GB	RAM capacity
System B	Memory	RAM Speed	3000MHz	MHz	RAM speed
System B	Memory	RAM Latency	18ns	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	280W	W	Power supply output
System B	Power Supply	Efficiency	83%	%	Power supply efficiency
System B	Power Supply	Voltage Regulation	±1%	%	Power supply voltage regulation
System B	Power Supply	Fan Speed	1100 RPM	RPM	Power supply fan speed

Metadata was added on the basis of the filenames.

```
# Adding a textbook proficiency level
TxBLevels <- read.delim(here("data", "metadata",
  "TxB900MDA_ProficiencyLevels.csv"), sep = ",")
TxBcounts <- full_join(TxBcounts, TxBLevels, 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)
```

D.2.1 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"))
```

D.3 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"))
```

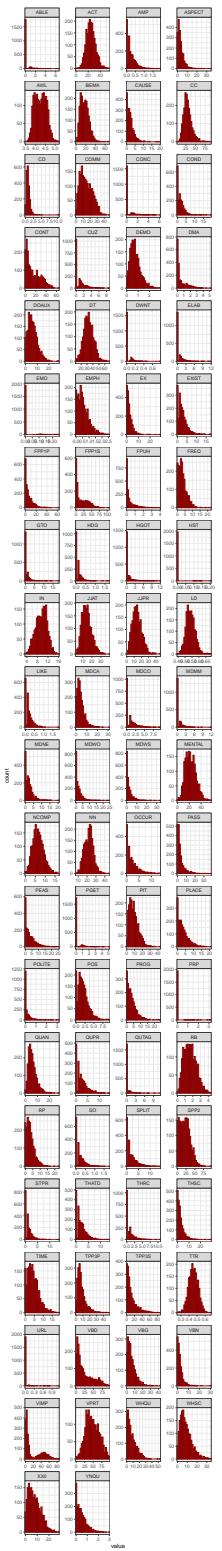
Register	# texts
Conversation	593
Fiction	285
Informative	364
Instructional	647
Personal	88

D.3.1 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)
```

D.3.2 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 spelt 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.

D.3.3 Identifying potential outlier texts

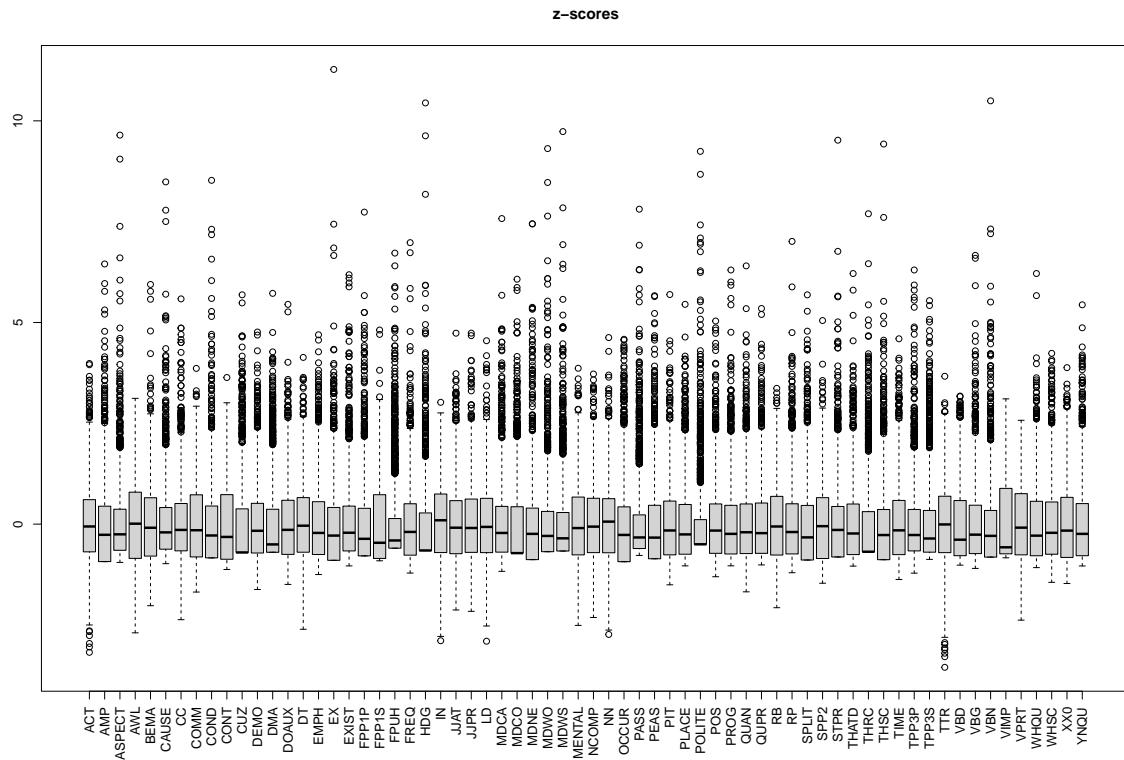
All normalised frequencies were normalised to identify any potential outlier texts.

```

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(Word, 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)

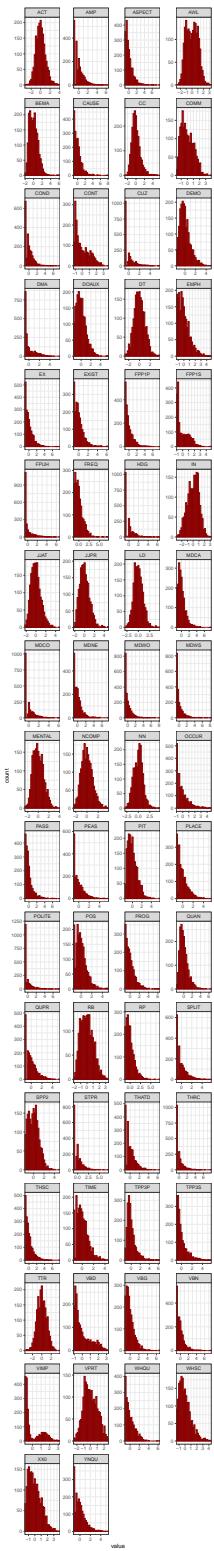
#saveRDS(TxBcounts, here("data", "processed", "TxBcounts3.rds")) # Last saved
  ↵ 6 March 2024

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 standardised 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)
```

D.3.4 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>

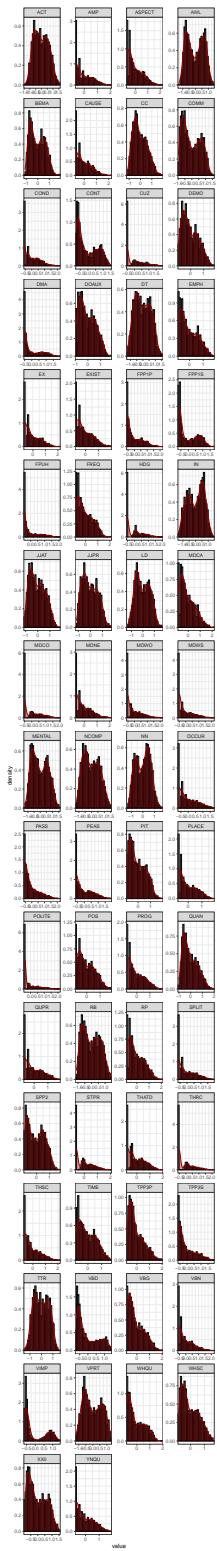
```
# All features are signed log-transformed (note that this is also what
  ↵ Neumann & Evert 2021 propose)
signed.log <- function(x) {
  sign(x) * log(abs(x) + 1)
}

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
```

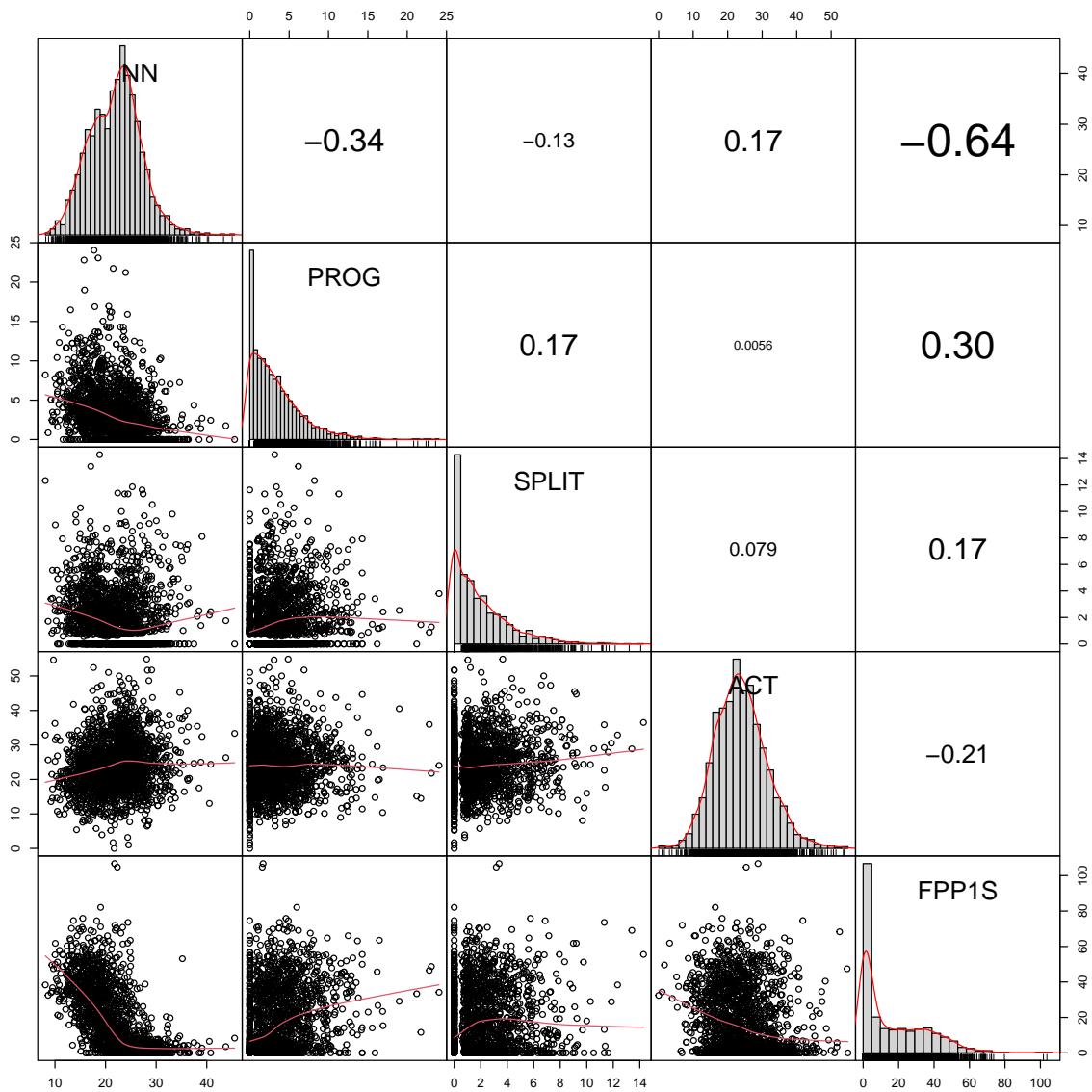
```

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)

```

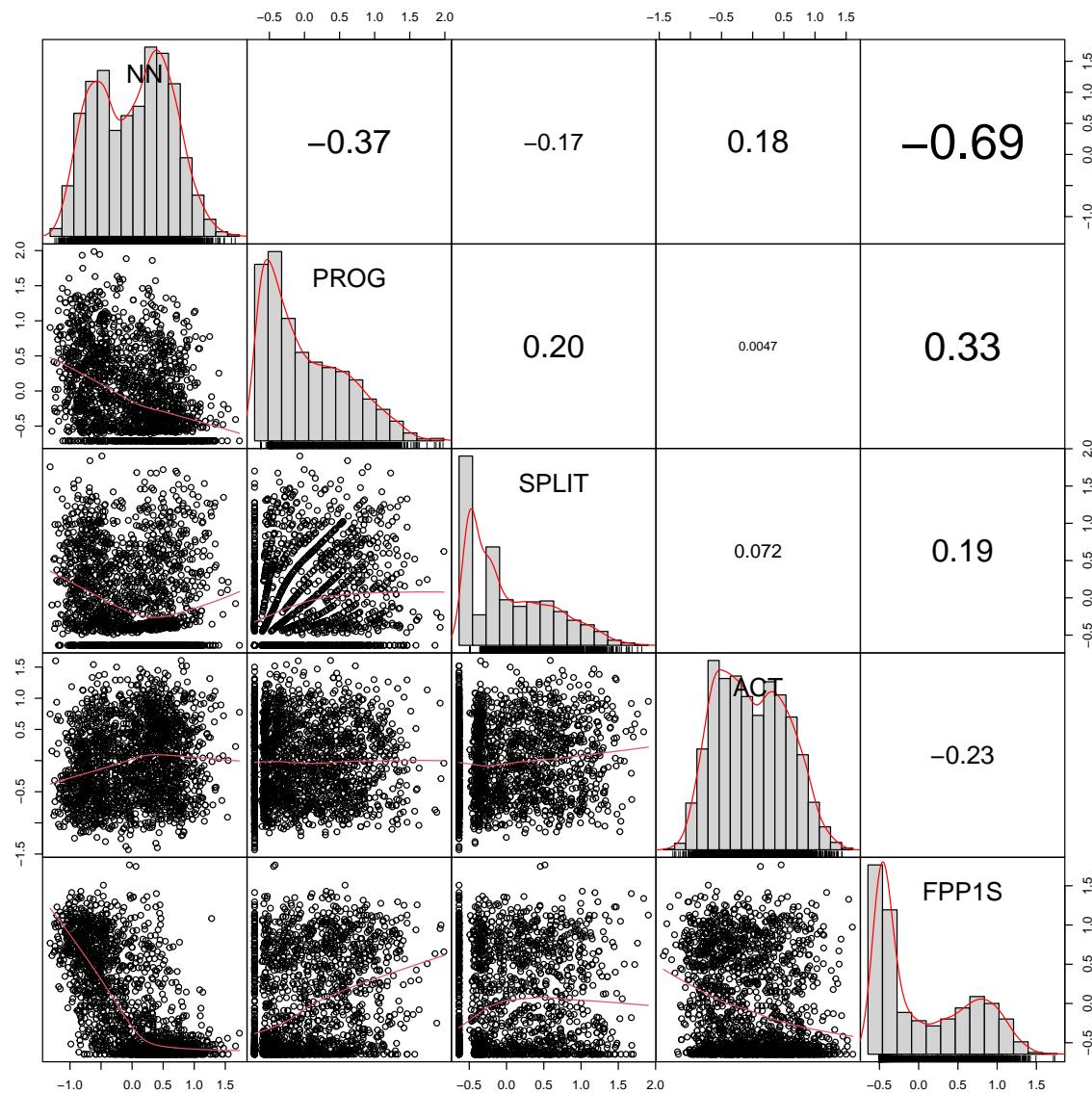


```
#dev.off()
```

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)
```



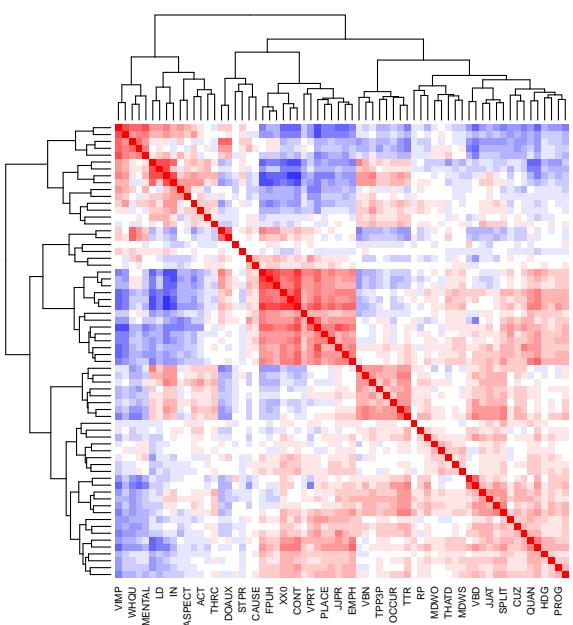
```
#dev.off()
```

D.3.5 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=",")
```

D.4 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")
```

```

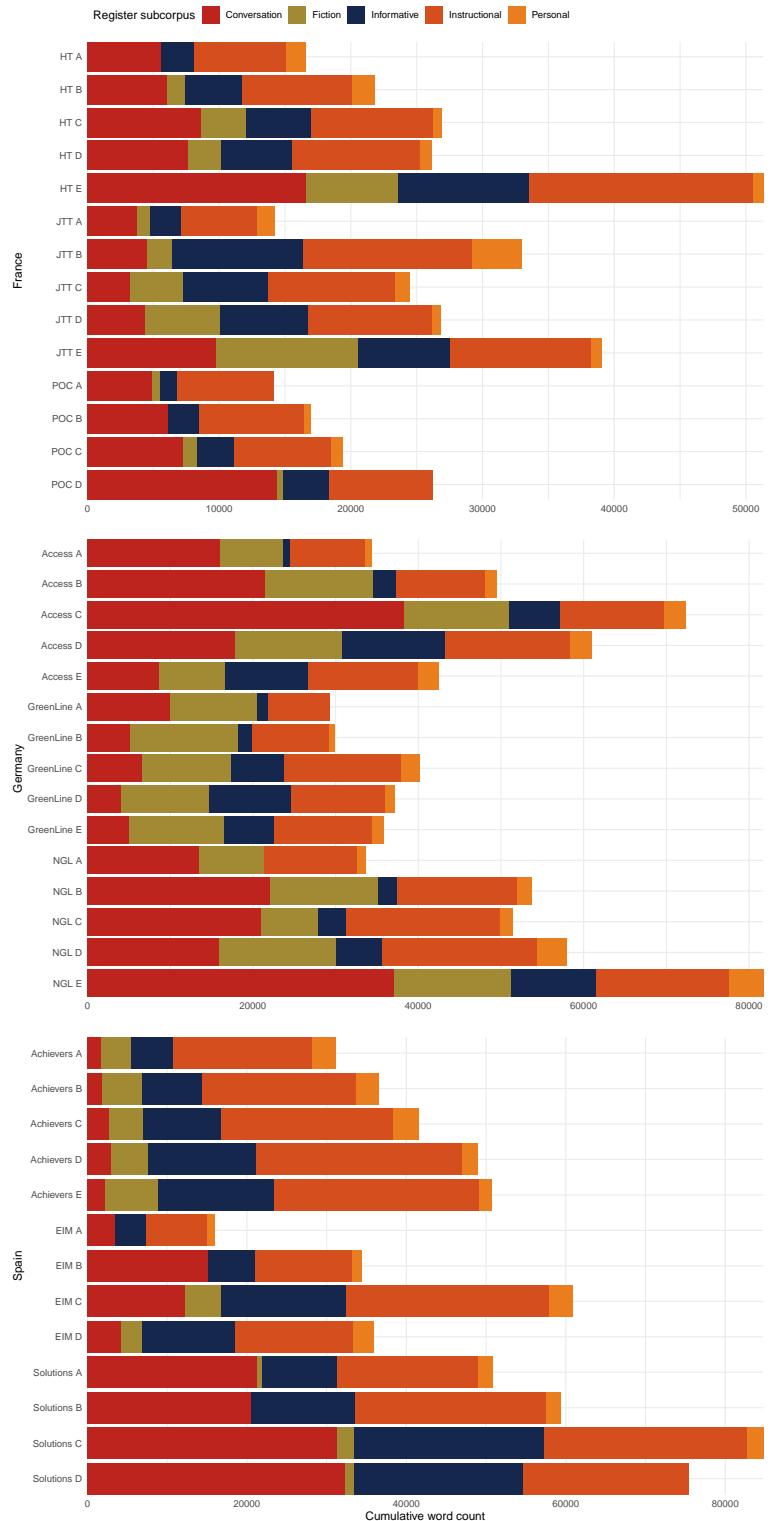
PlotSp <- metadata2 |>
  filter(Country=="Spain") |>
  #arrange(Volume) |>
  ggplot(aes(x = Volume, y = wordcount, fill = fct_rev(Register))) +
    geom_bar(stat = "identity", position = "stack") +
    coord_flip(expand = FALSE) + # Removes those annoying ticks before each
    ↵ bar label
    theme_minimal() + 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

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.

E.1 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
```

E.2 Preparing the data for PCA

E.2.1 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 comprises 1,961 texts/files, each with logged standardised normalised frequencies for 66 linguistic features.

E.3 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

E.3.1 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:

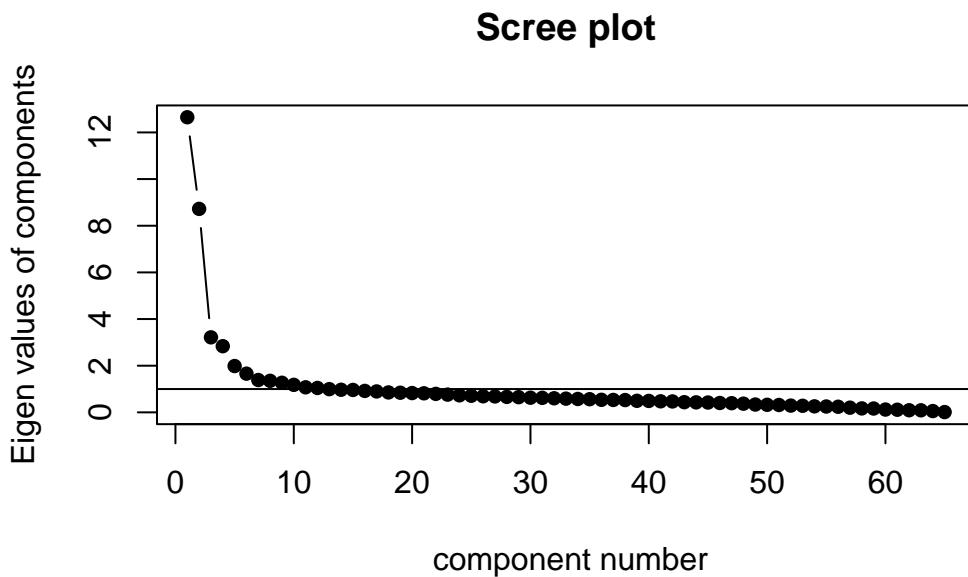
```
kmo2$MSAi [order(kmo2$MSAi)] |> round(2)
```

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	

E.3.2 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
```



```
#dev.off()

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

E.3.3 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.

E.4 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))

```

E.5 Principal Component Analysis (PCA)

E.5.1 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"))

```

E.5.2 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).
```

E.5.3 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).
```

E.5.4 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

```

E.6 Plotting PCA results

E.6.1 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")

```

E.7 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.

E.7.1 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)

```

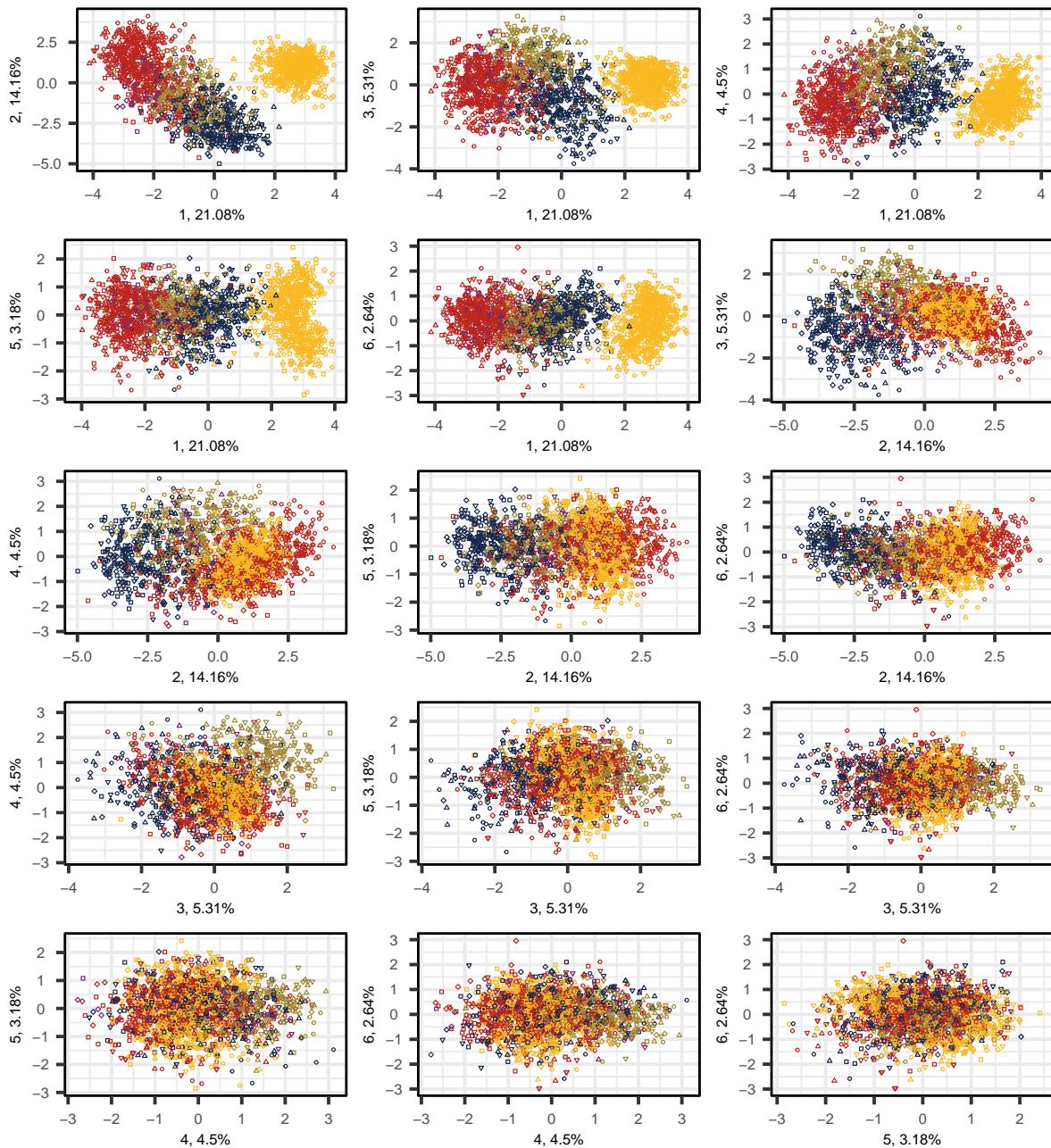
E.7.2 Pairs plot

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")
```

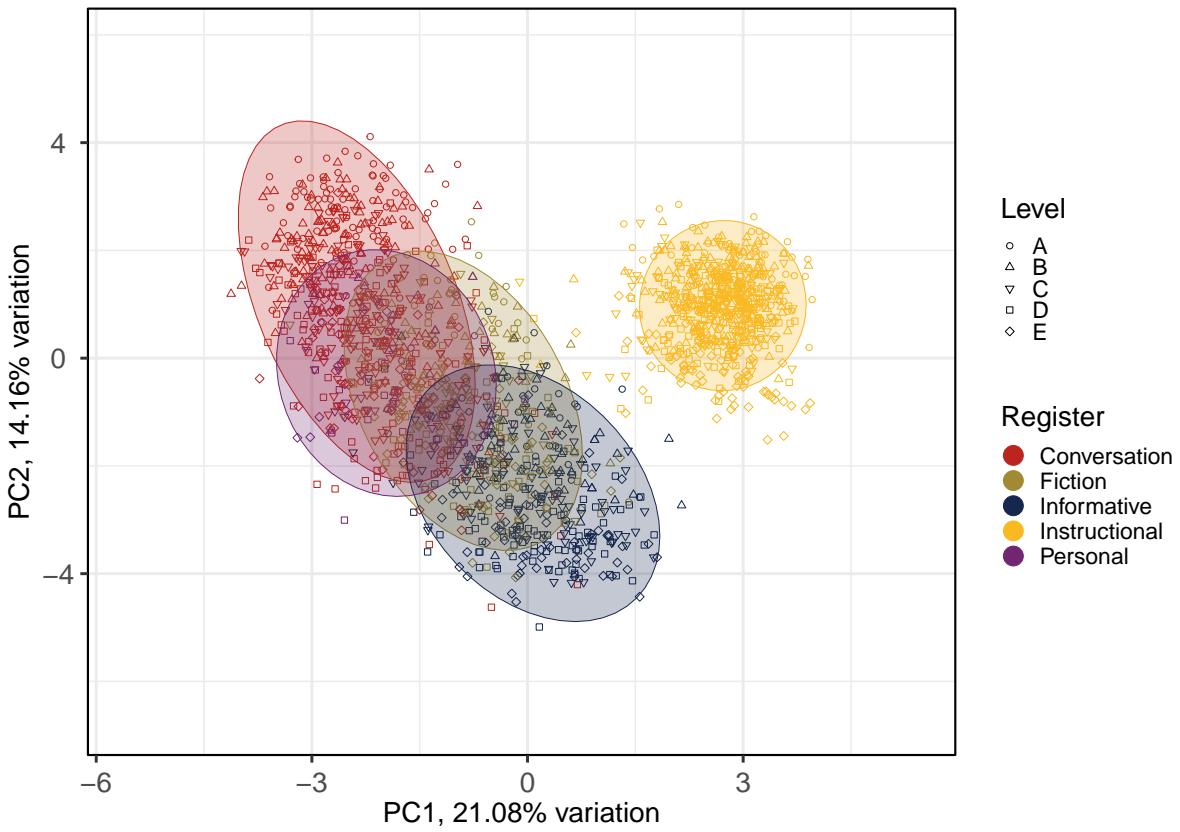


E.7.3 Bi-plots

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_BiplotPC1_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),

```

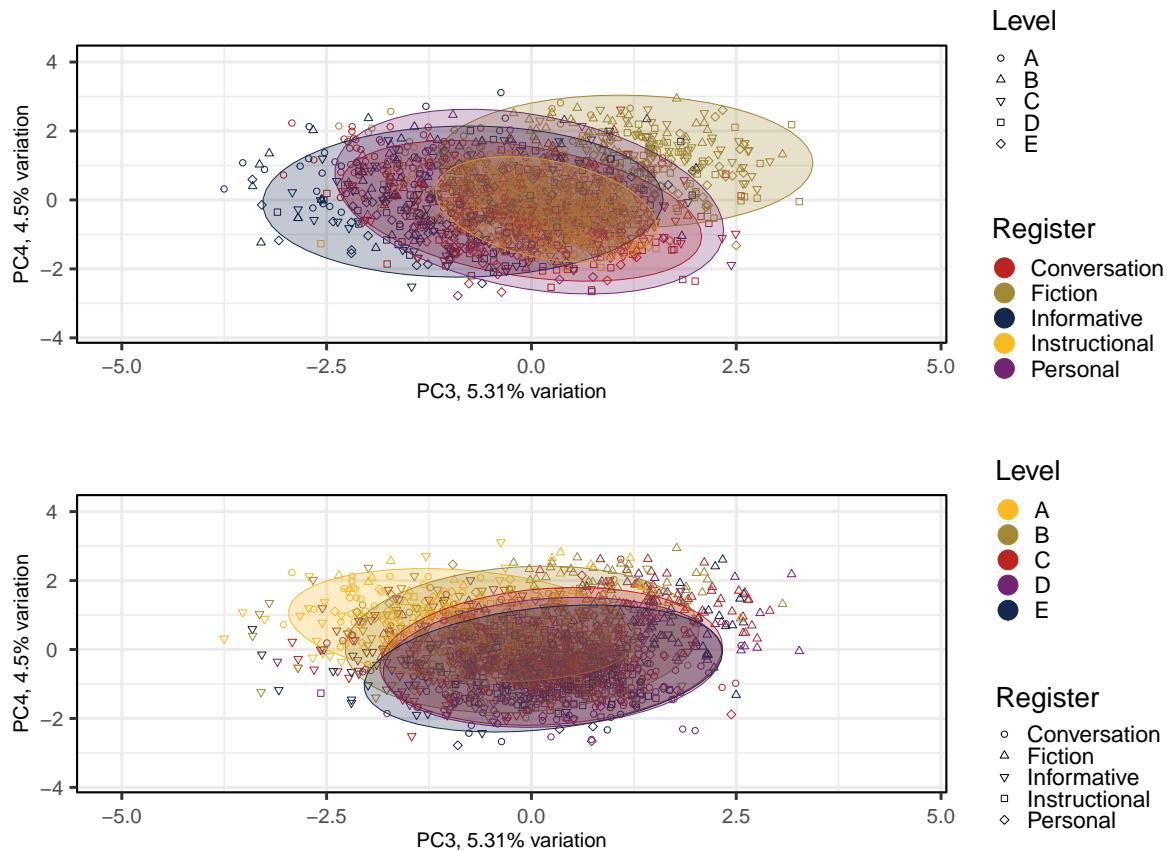
```

    #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)

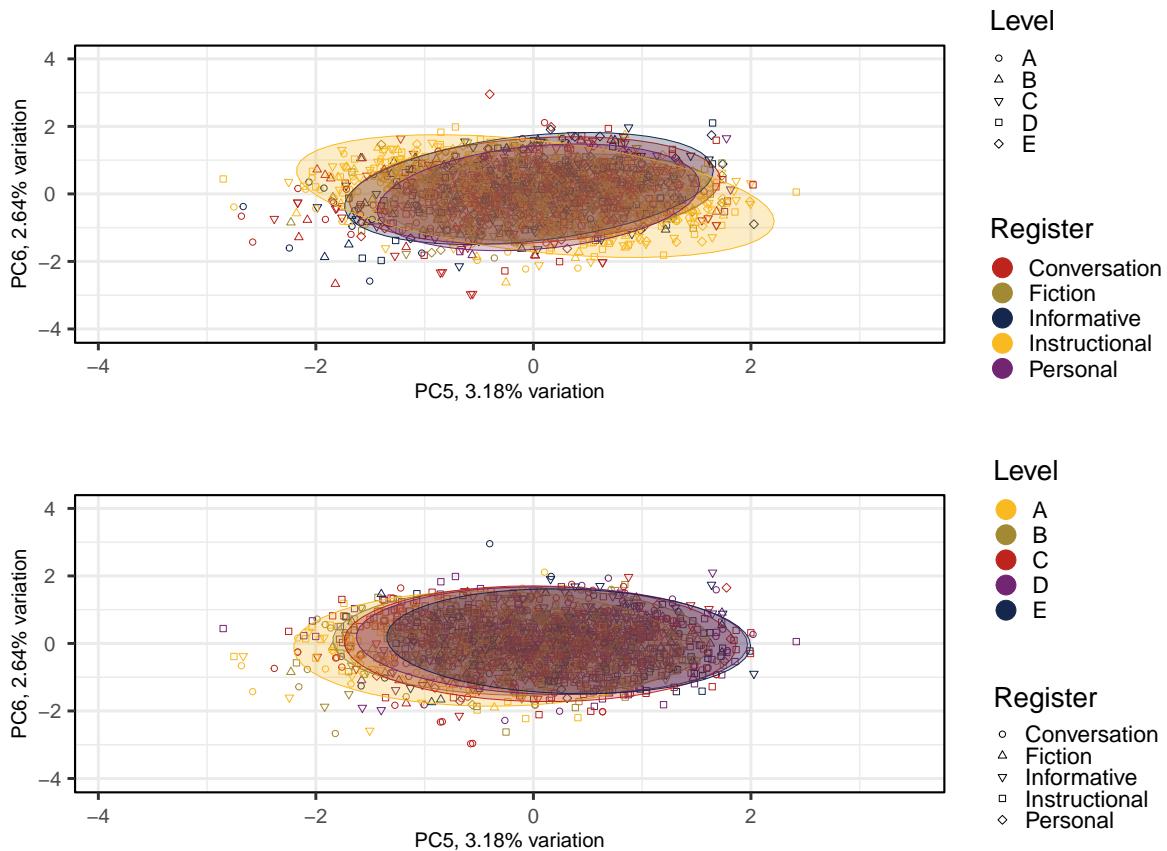
# Display and save the two different representations of data points on PC2
  ← and PC3 using the {patchwork} package
pRegisters / pLevels

```



```
#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)
```

E.8 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
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

	PC1	PC2	PC3	PC4
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.:

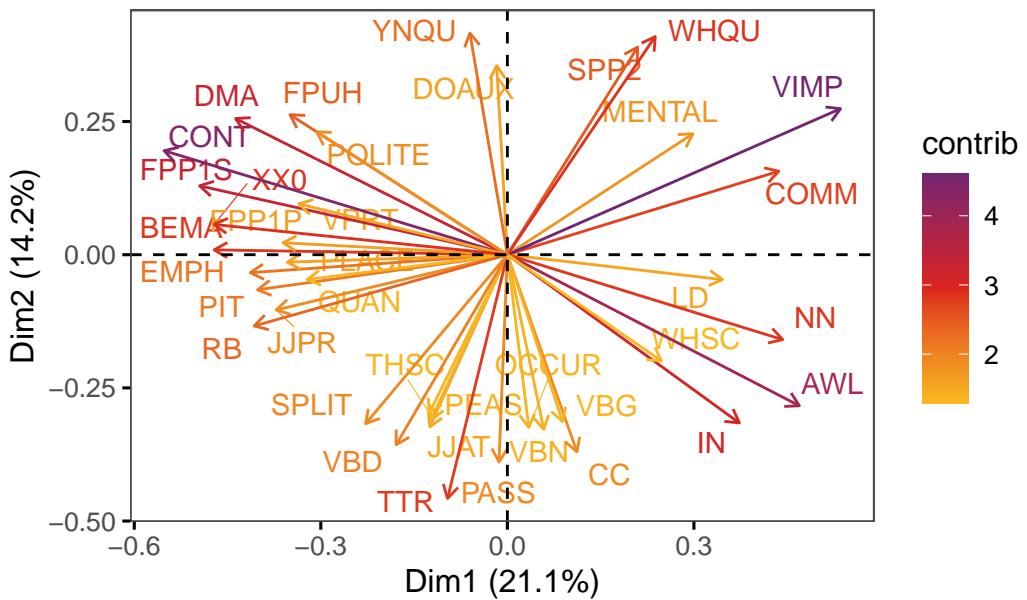
```
TxBcounts |>
  group_by(Register, Level) |>
  summarise(median(NCOMP), MAD(NCOMP)) |>
  select(1:4) |>
  kable(digits=2)
```

Register	Level	median(NCOMP)	MAD(NCOMP)
Conversation	A	5.69	2.79
Conversation	B	5.48	2.66

Register	Level	median(NCOMP)	MAD(NCOMP)
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

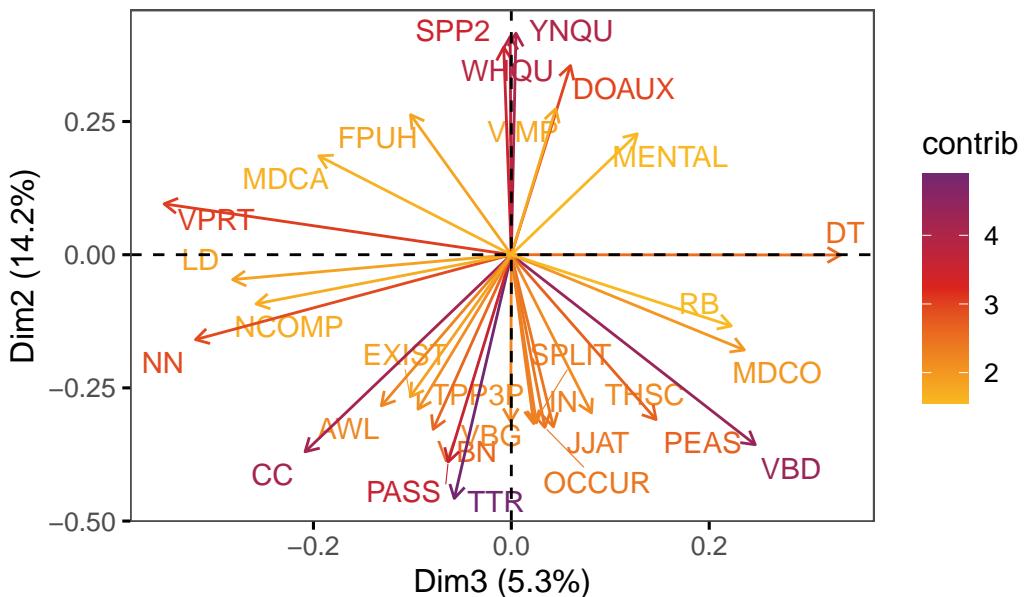
Graphs of features display the features with the strongest contributions to any two dimensions of the model of intra-textbook variation. They are created using the `factoextra::fviz_pca_var` function.

```
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())
```



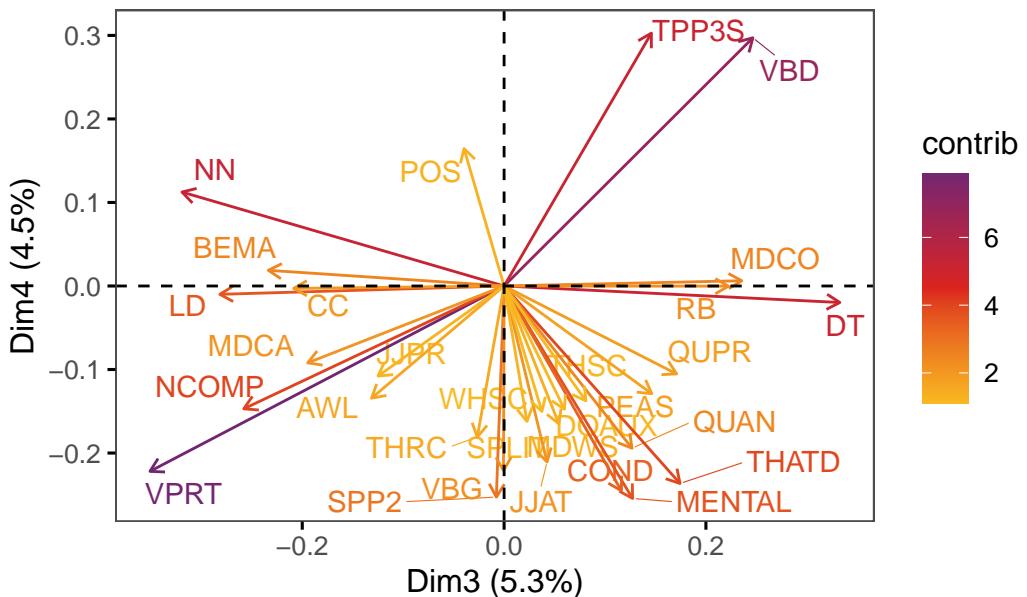
```
#ggsave(here("plots", "fviz_pca_var_PC1_PC2.svg"), width = 11, height = 9)

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)
```

E.9 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

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 that 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

E.10 Computing mixed-effects models of the dimension scores

E.10.1 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

```

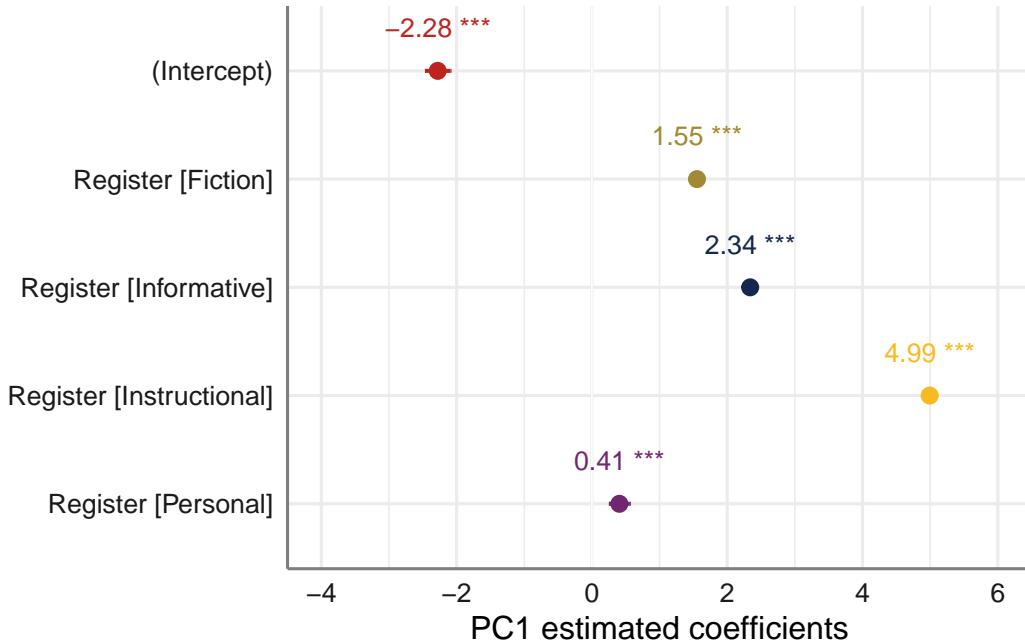
```
tab_model(md1, wrap.labels = 300) # Marginal R2 = 0.890
```

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

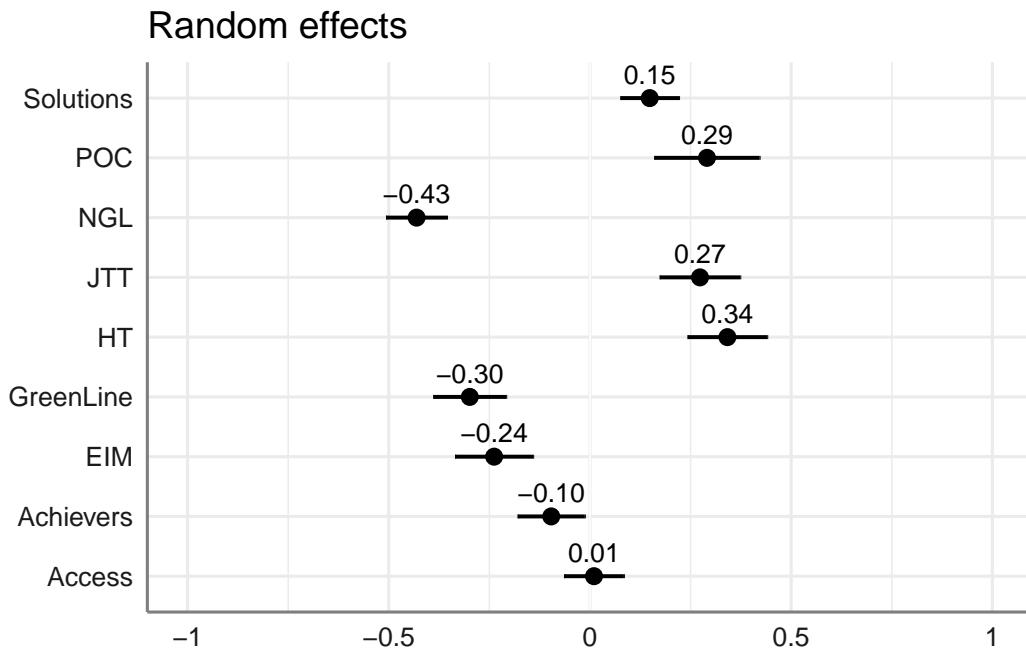
Confidence level used: 0.95

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)
```

E.10.2 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
```

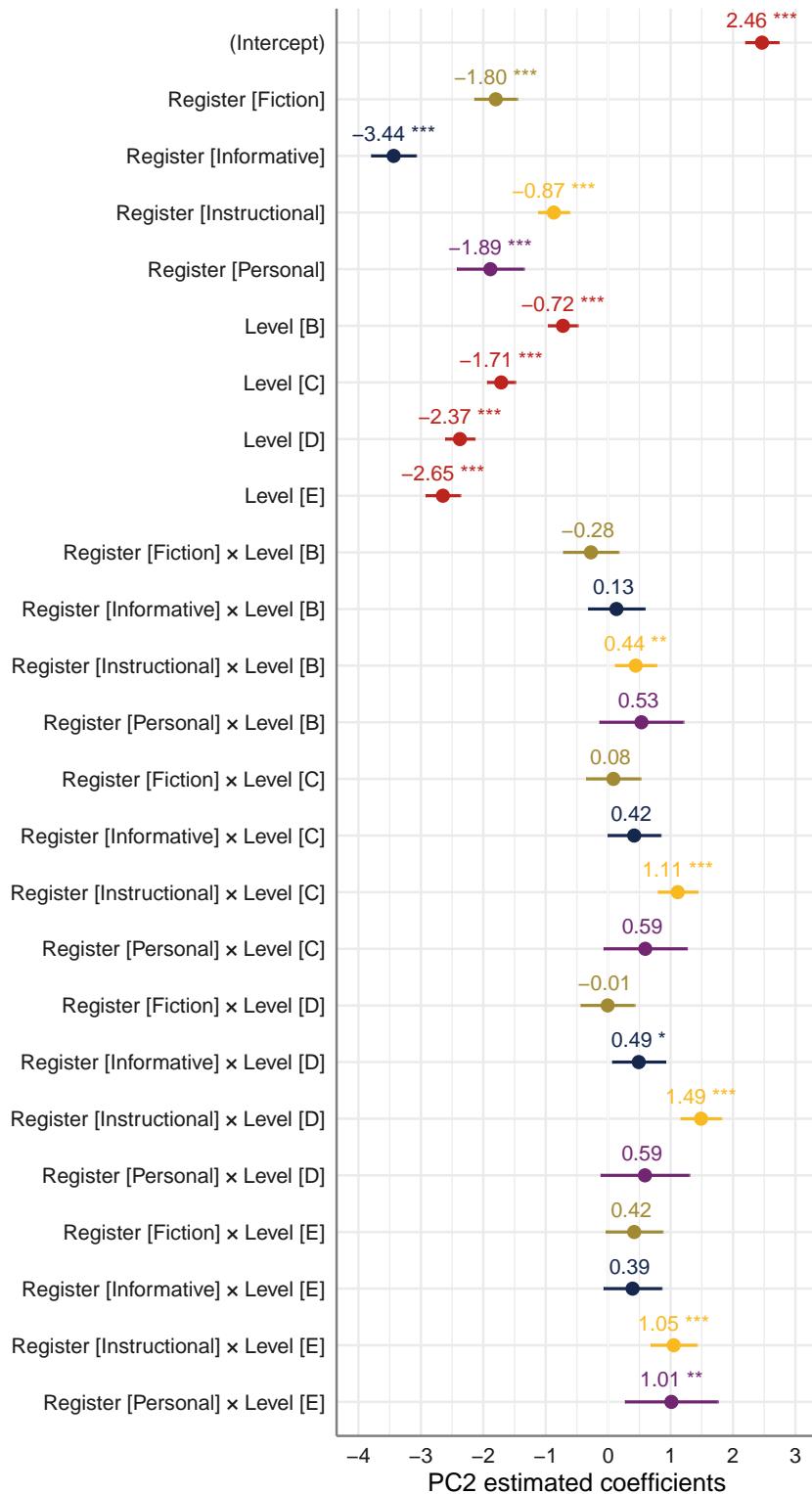
```
# tab_model(md2Register) # Marginal R2 = 0.558
# tab_model(md2Level) # Marginal R2 = 0.228
```

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()

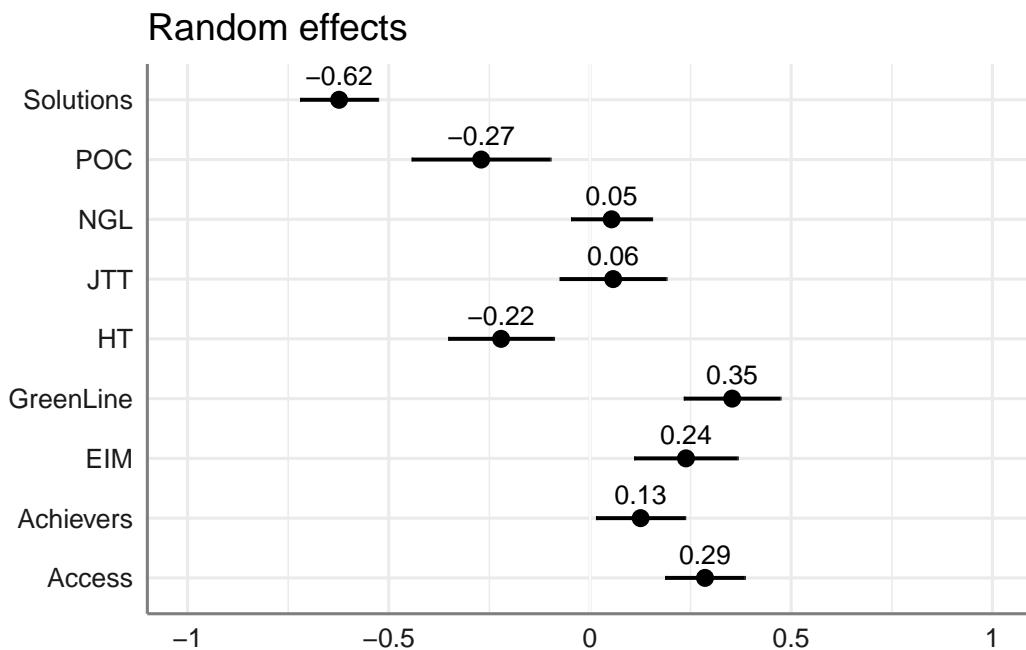
```



```
#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)

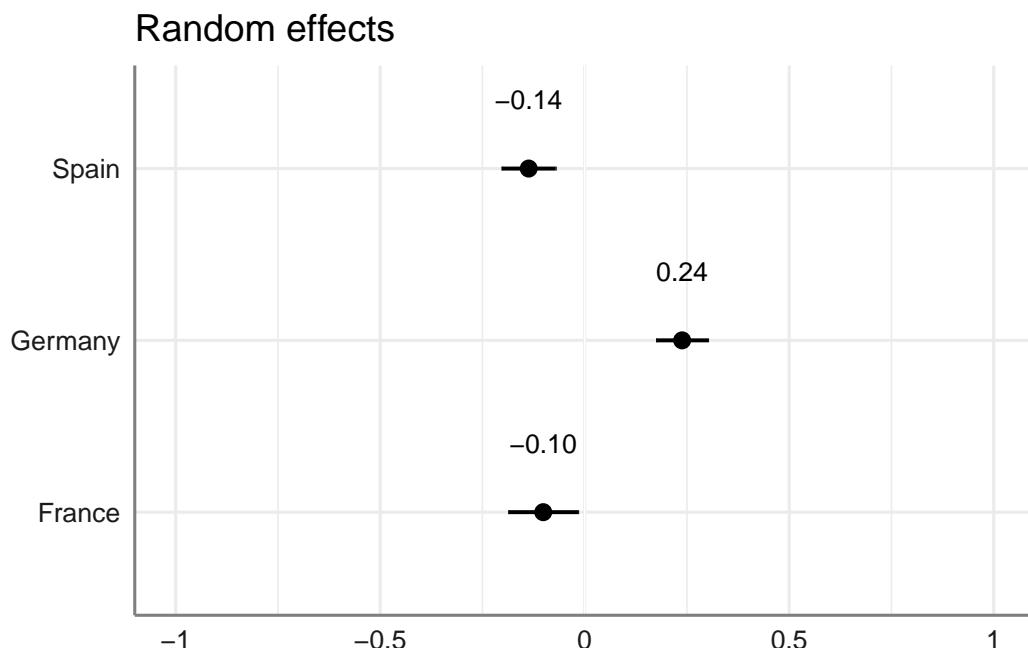
# Textbook Country as a random effect variable
```

```

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)

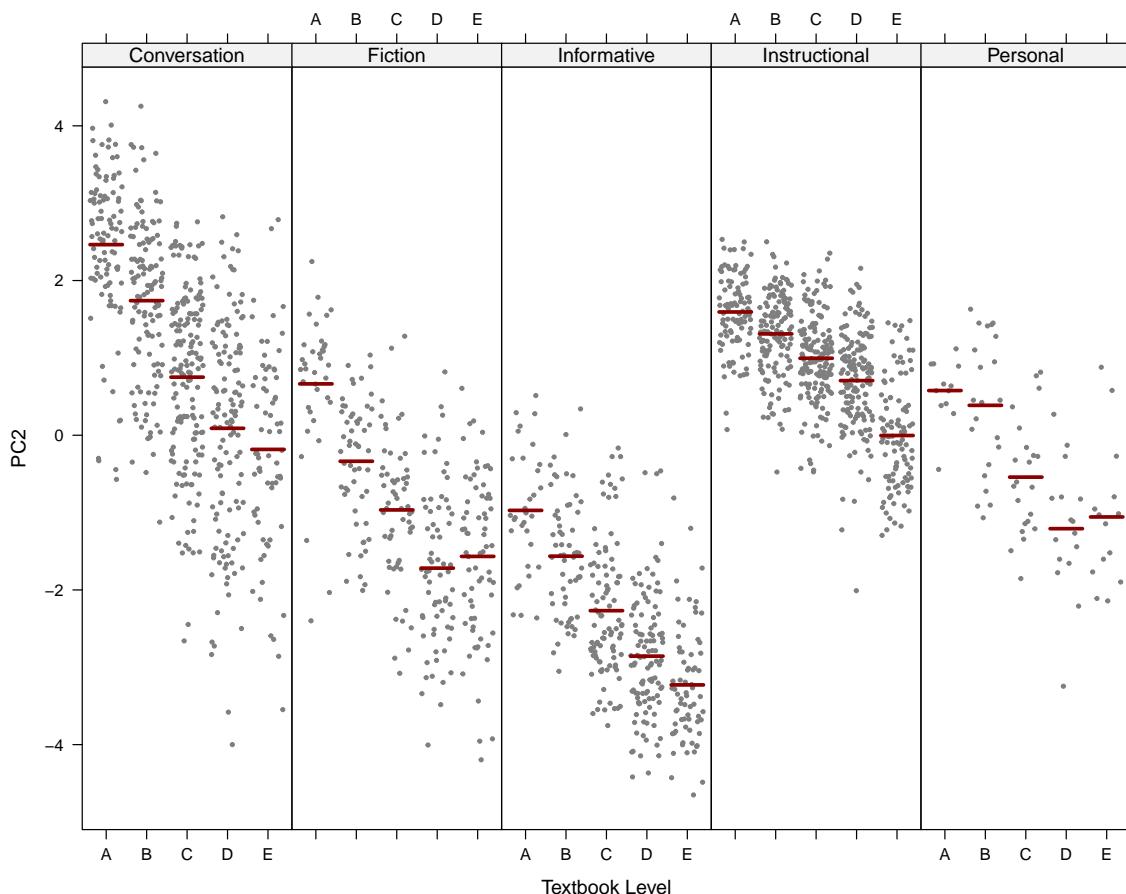
```

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)
)

```

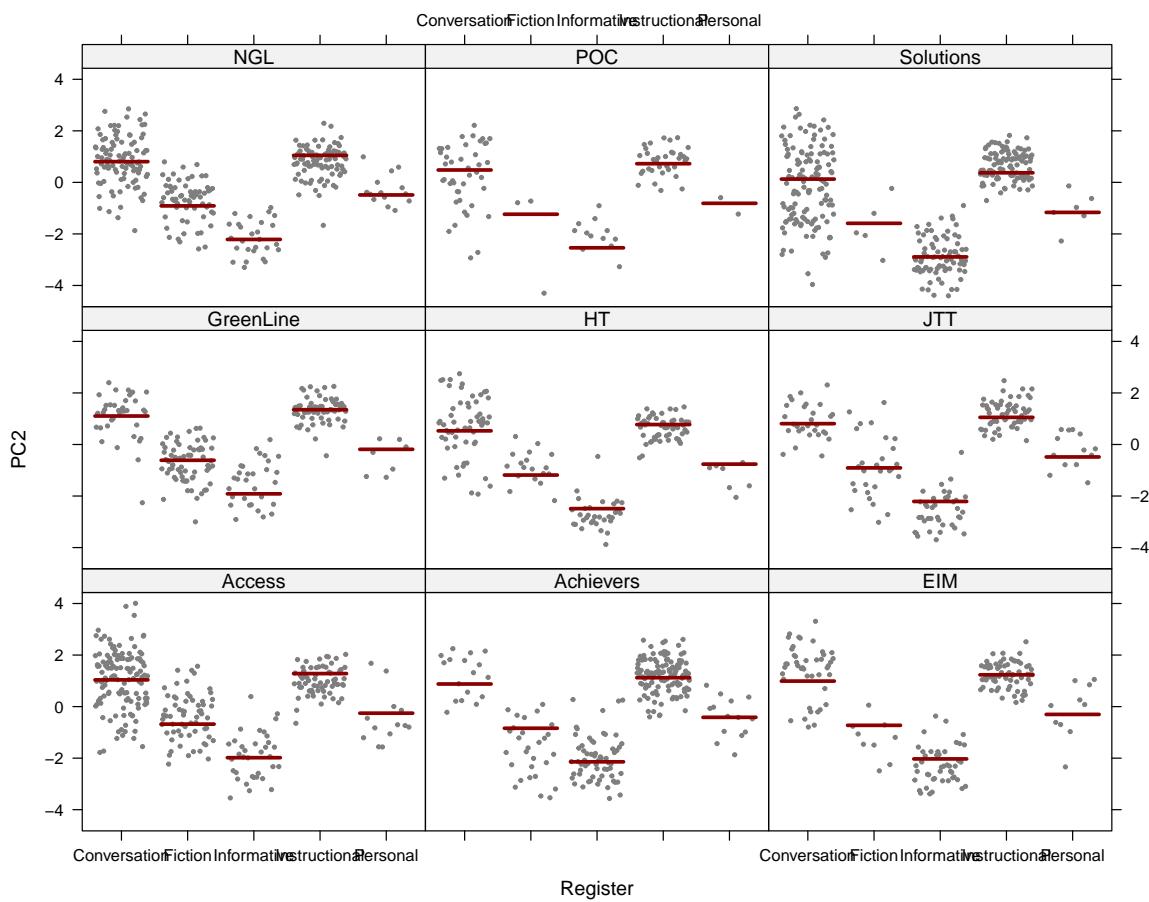


```

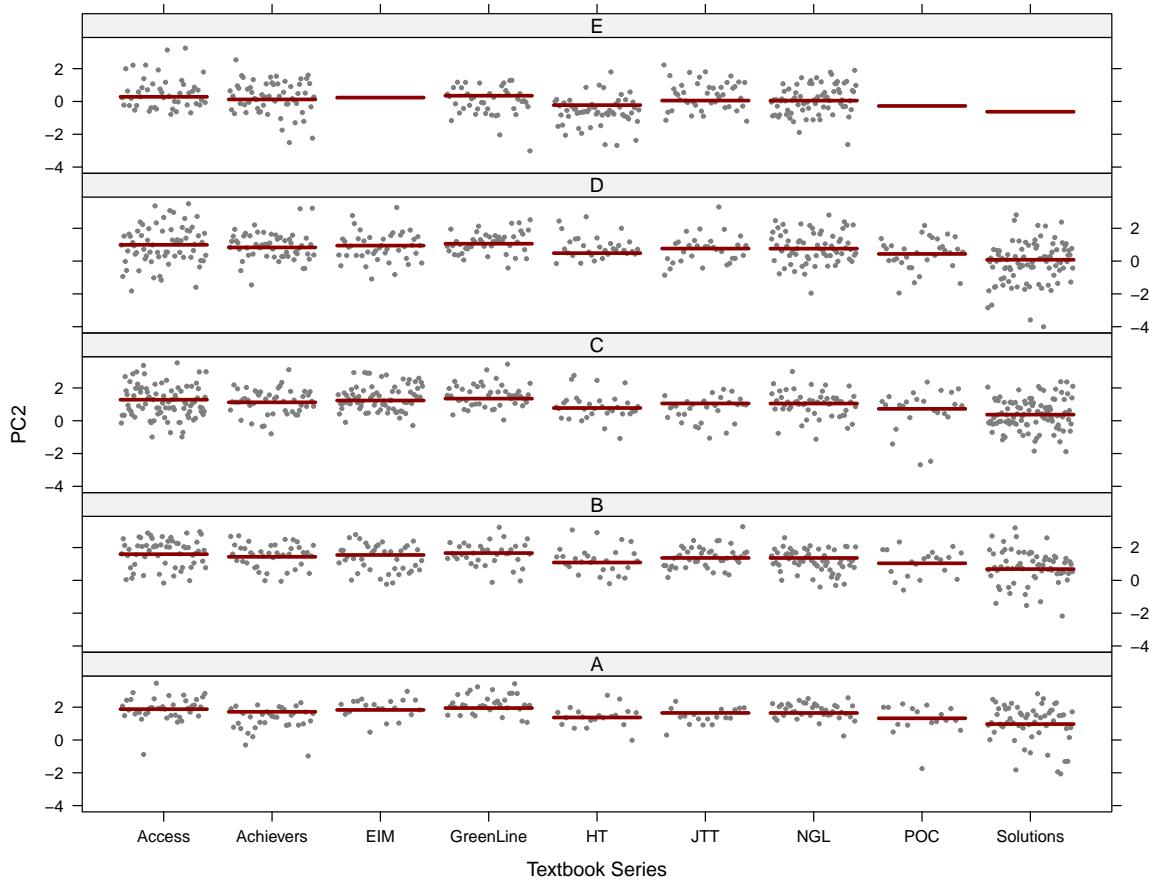
# dev.off()

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

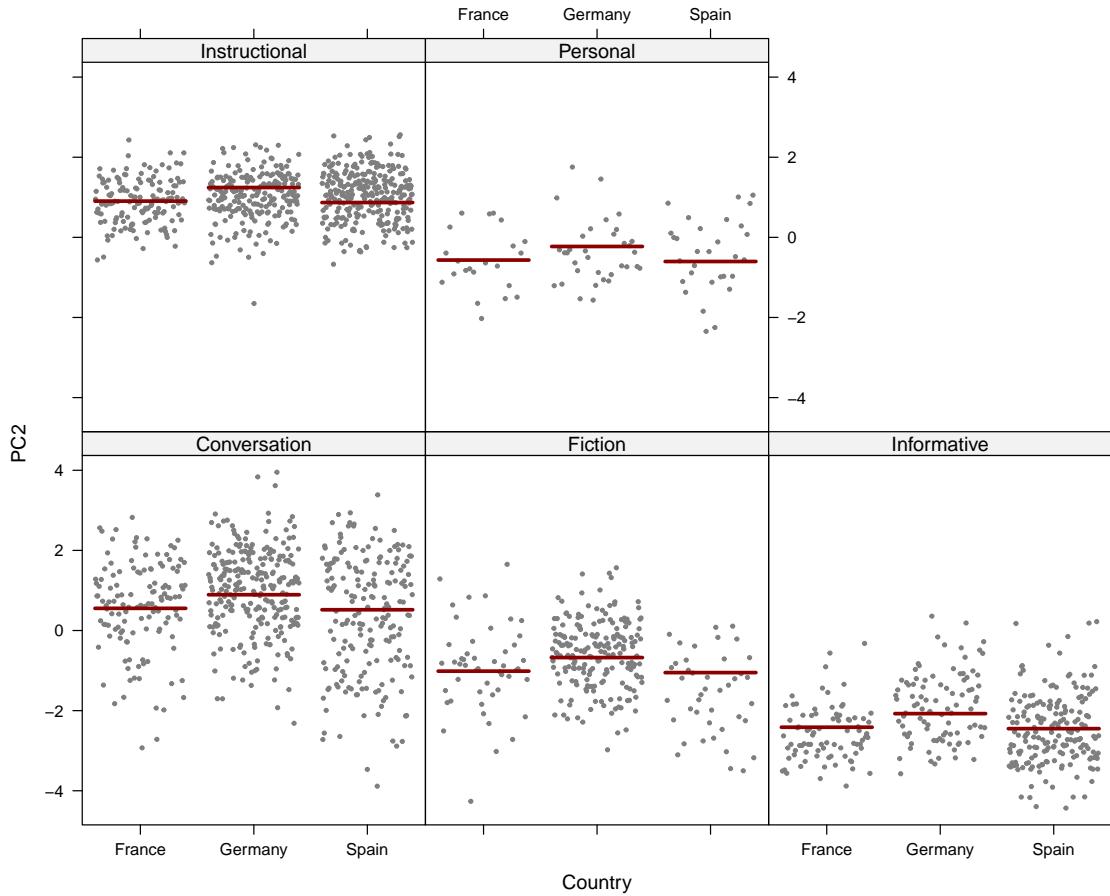
```



```
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()
```

E.10.3 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

```

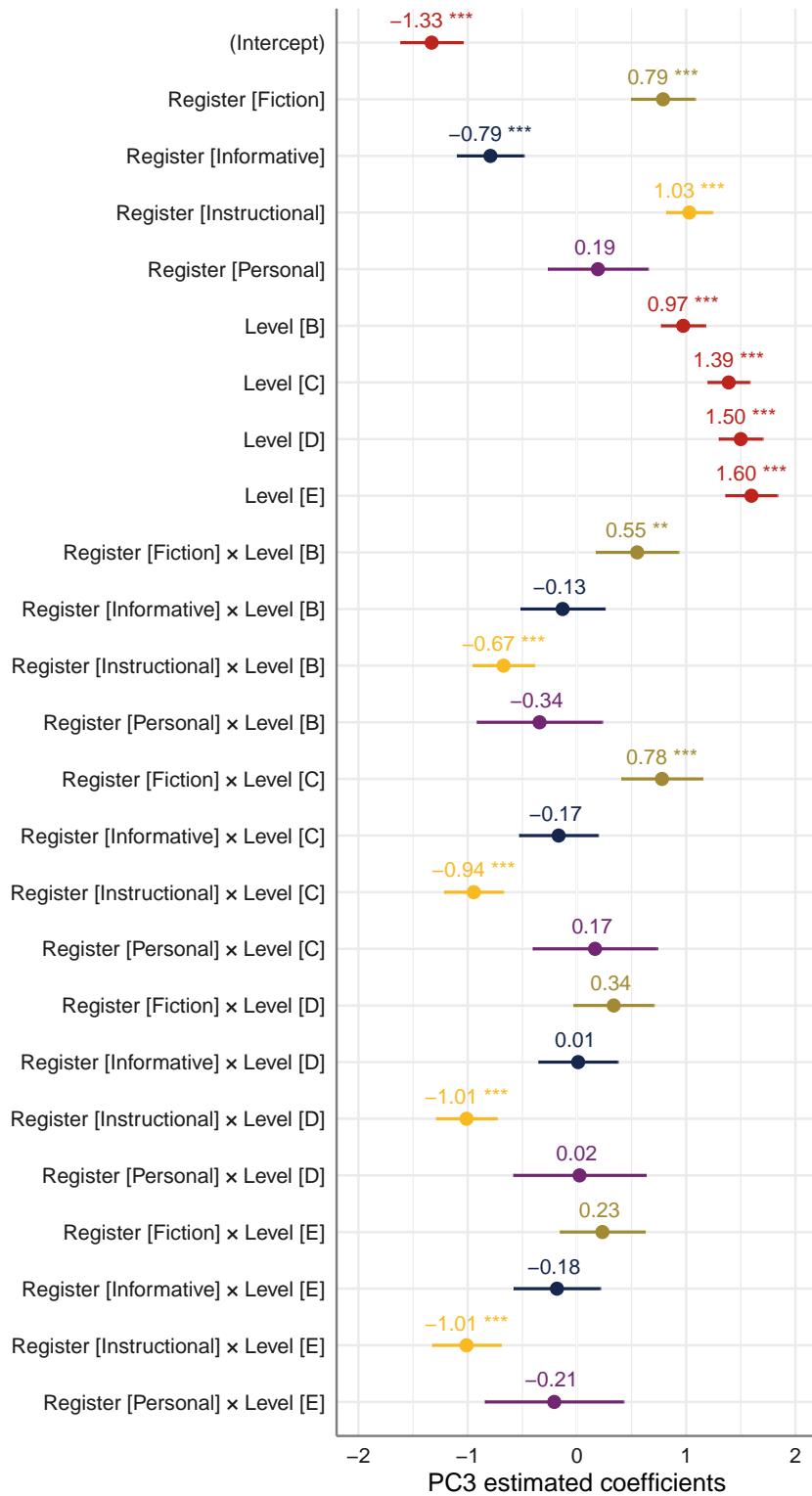
```
tab_model(md3) # Marginal R2 = 0.436
```

```

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

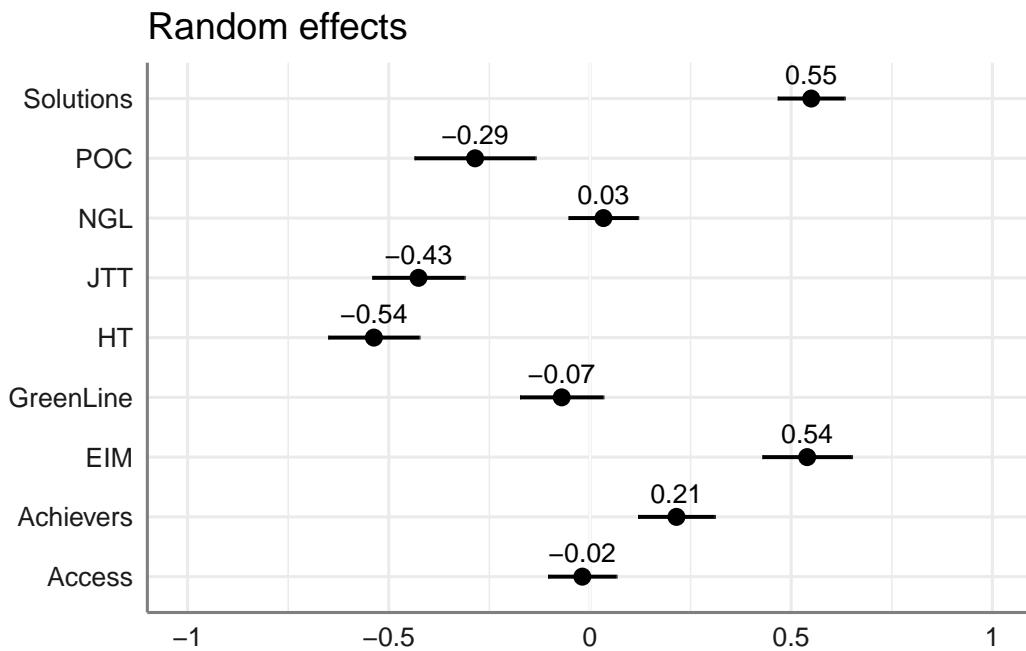
# Plot of fixed effects:
plot_model(md3,
            type = "est",
            show.intercept = TRUE,
            show.values=TRUE,
            show.p=TRUE,
            value.offset = .4,
            value.size = 3.5,
            colors = palette[c(1:3,8,7)],
            group.terms = c(1:5,1,1,1,1,2:5,2:5,2:5,2:5),
            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()
```



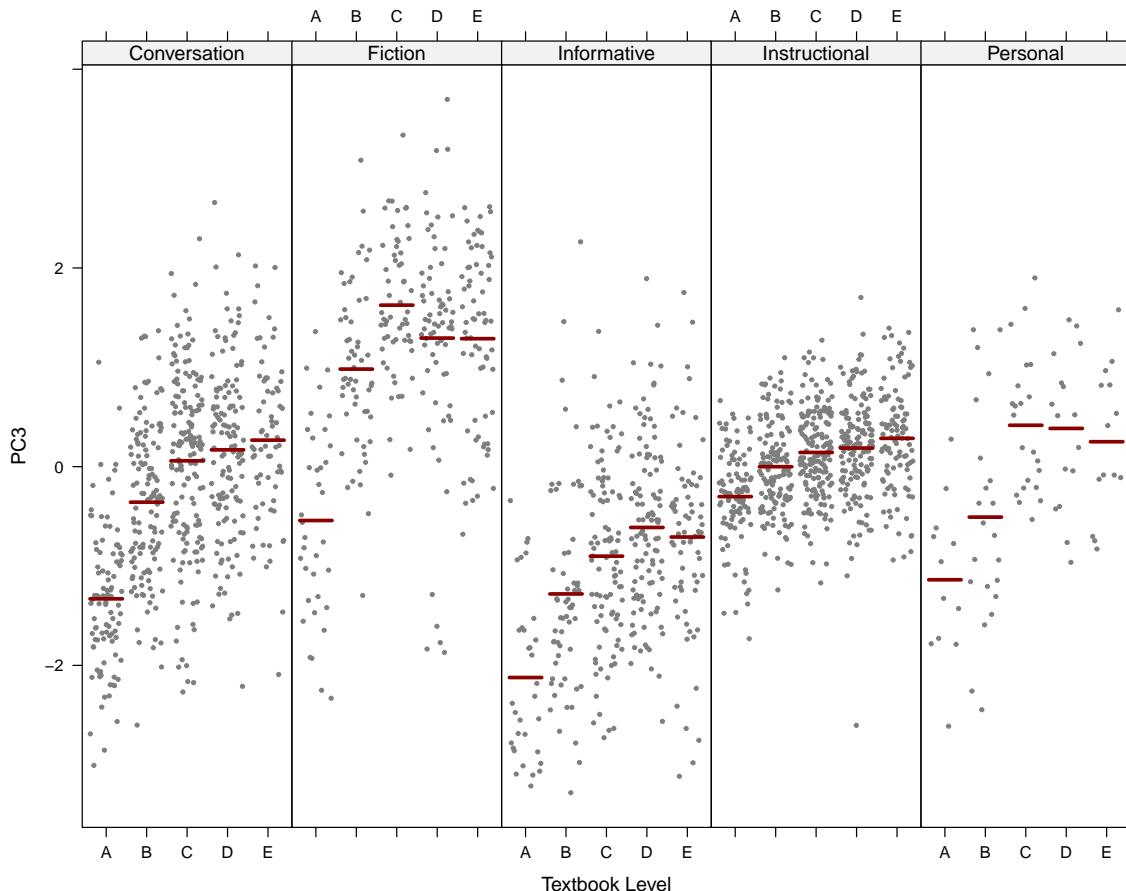
```
#ggsave(here("plots", "TxB_PCA3_lmer_randomeffects.svg"), height = 3, width =
  8)
```

```
# 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)
)

```

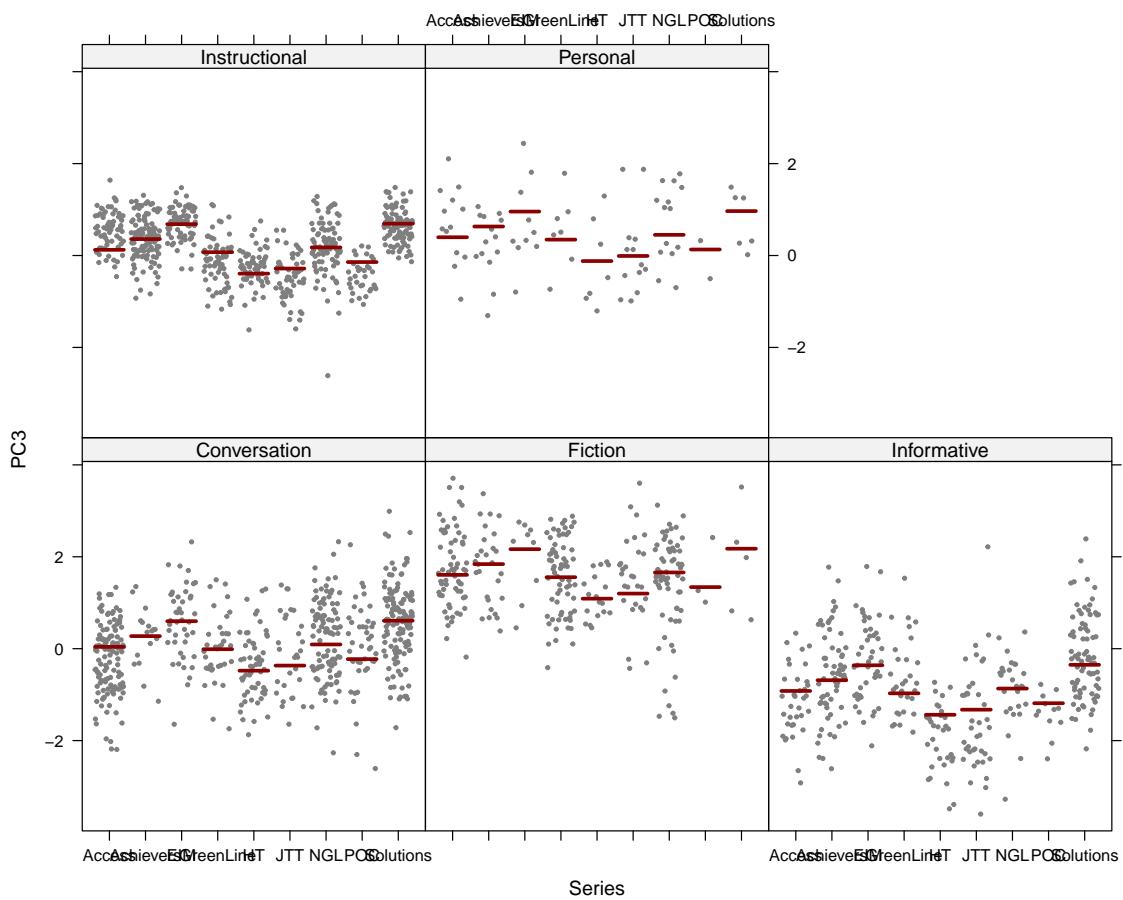


```

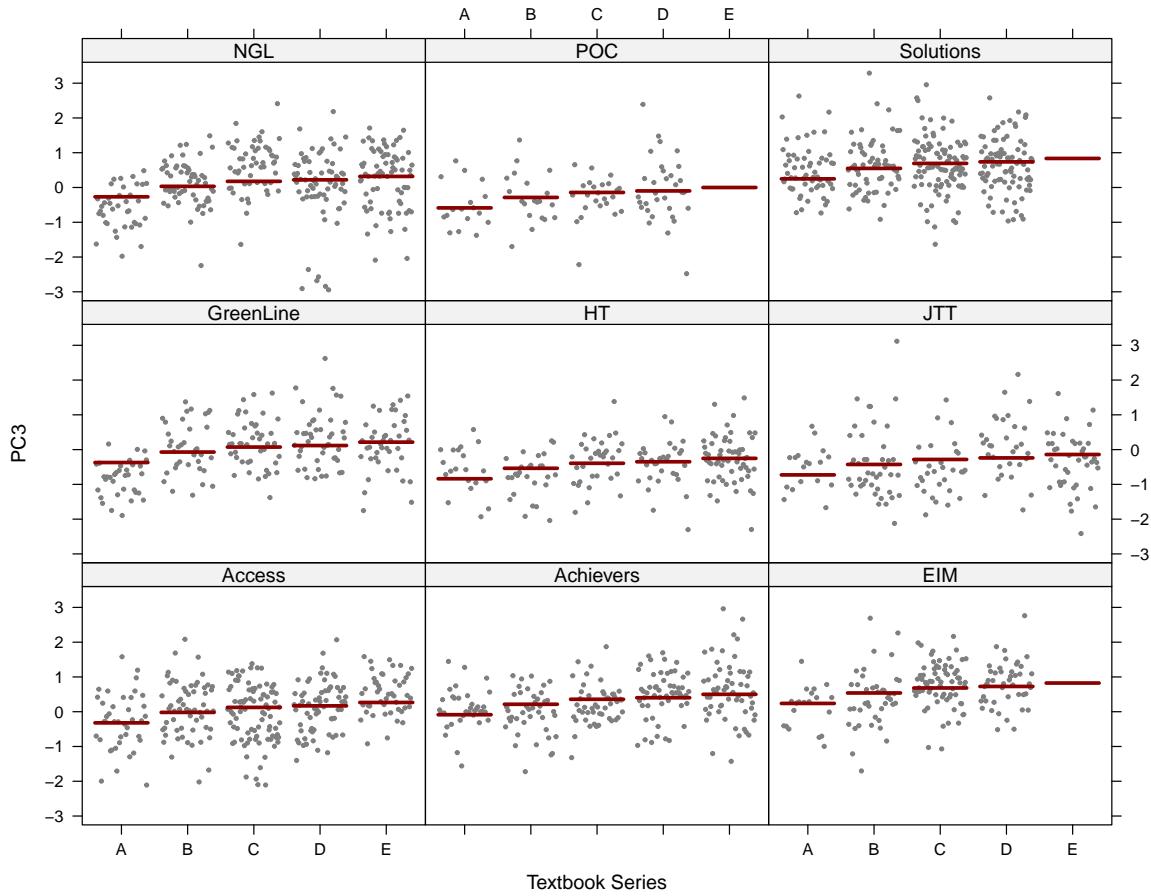
# dev.off()

# 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")
```



E.10.4 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

```

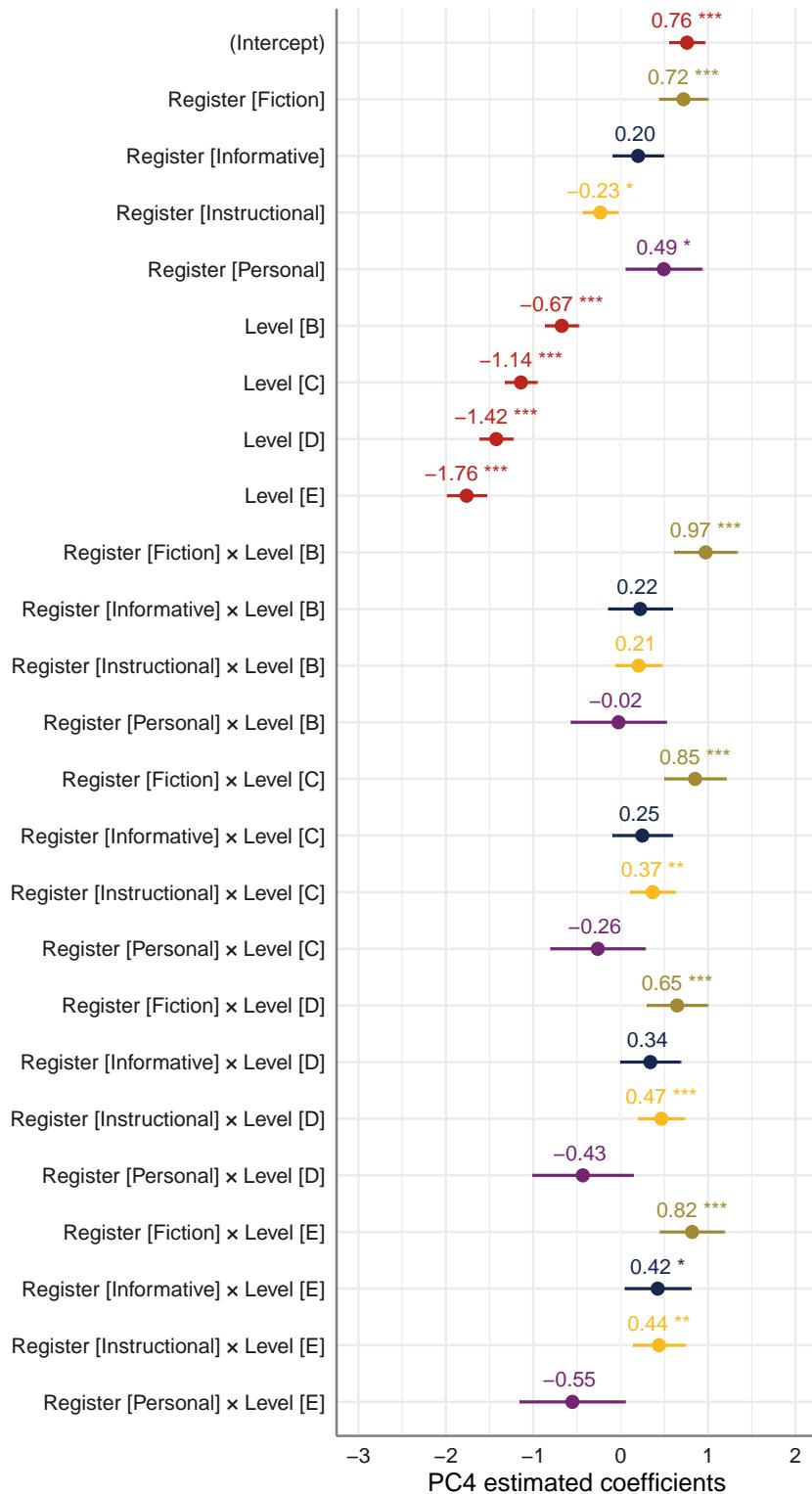
```
tab_model(md4) # Marginal R2 = 0.426
```

```

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

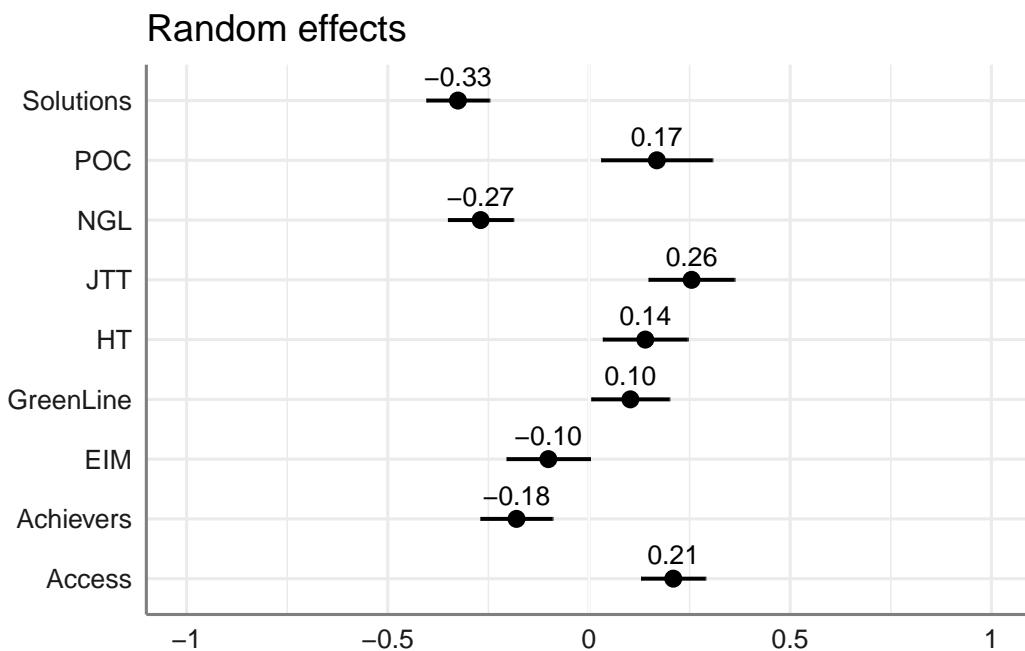
# Plot of fixed effects:
plot_model(md4,
            type = "est",
            show.intercept = TRUE,
            show.values=TRUE,
            show.p=TRUE,
            value.offset = .4,
            value.size = 3.5,
            colors = palette[c(1:3,8,7)],
            group.terms = c(1:5,1,1,1,1,2:5,2:5,2:5,2:5),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

```



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

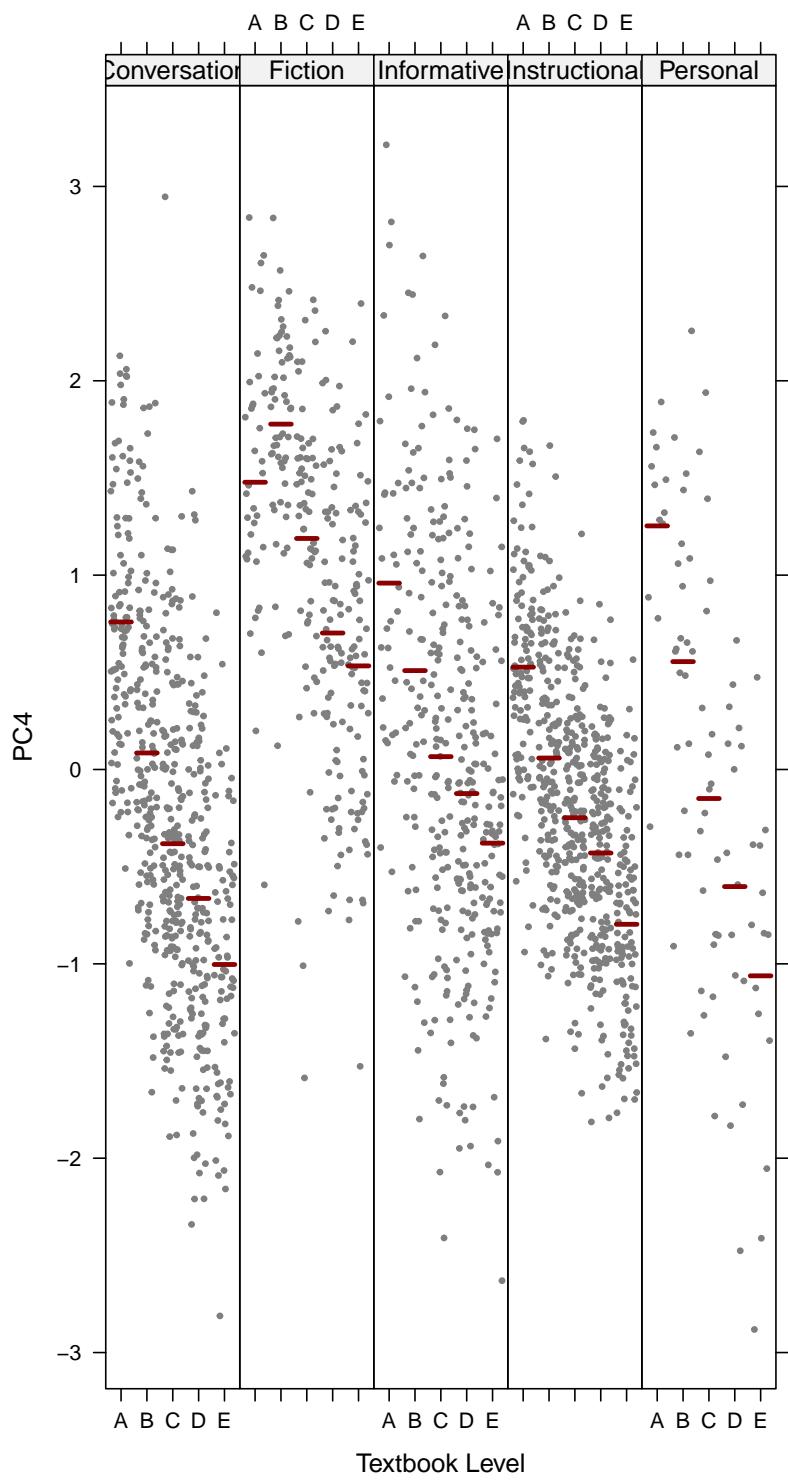
# Plot of random effects:
plot_model(md4,
  type = "re", # Option to visualise random effects
  show.values=TRUE,
  show.p=TRUE,
  value.offset = .4,
  value.size = 3.5,
  color = "bw",
  wrap.labels = 40,
  axis.title = "PC4 estimated coefficients") +
theme_sjplot2()
```



```
#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)
)
```



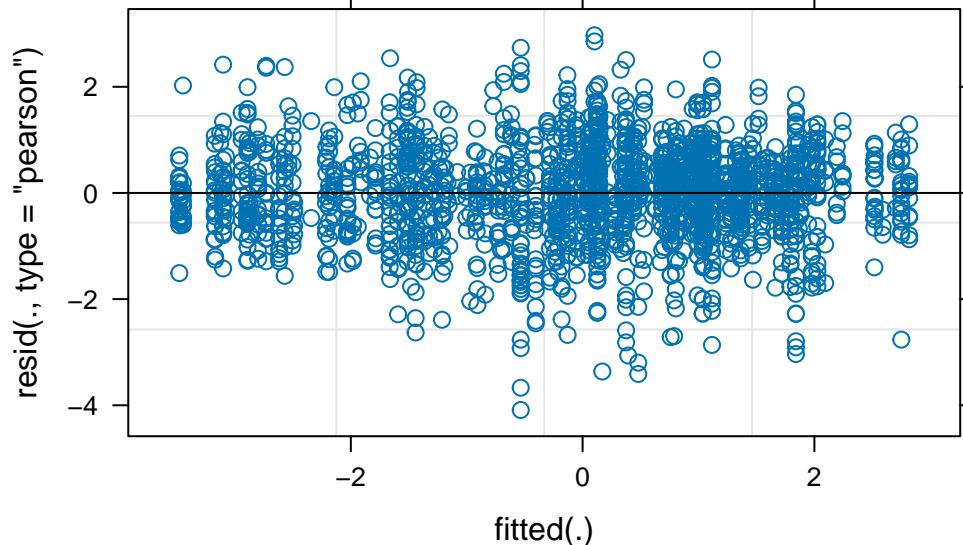
```
# dev.off()
```

E.10.5 Testing model assumptions

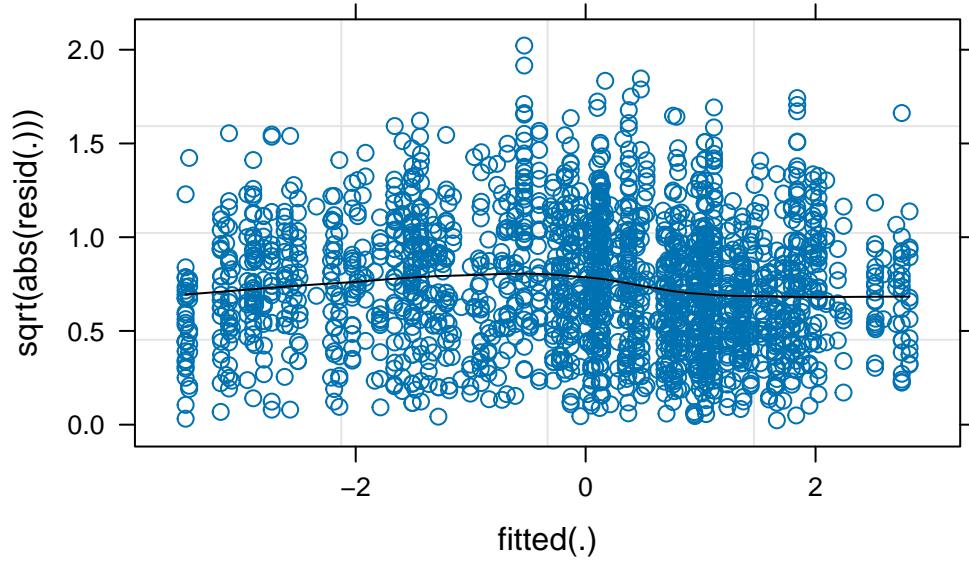
This chunk can be used to check the assumptions of all of the models computed above. In the following example, we examine the final model selected to predict Dim2 scores.

```
model2test <- md2

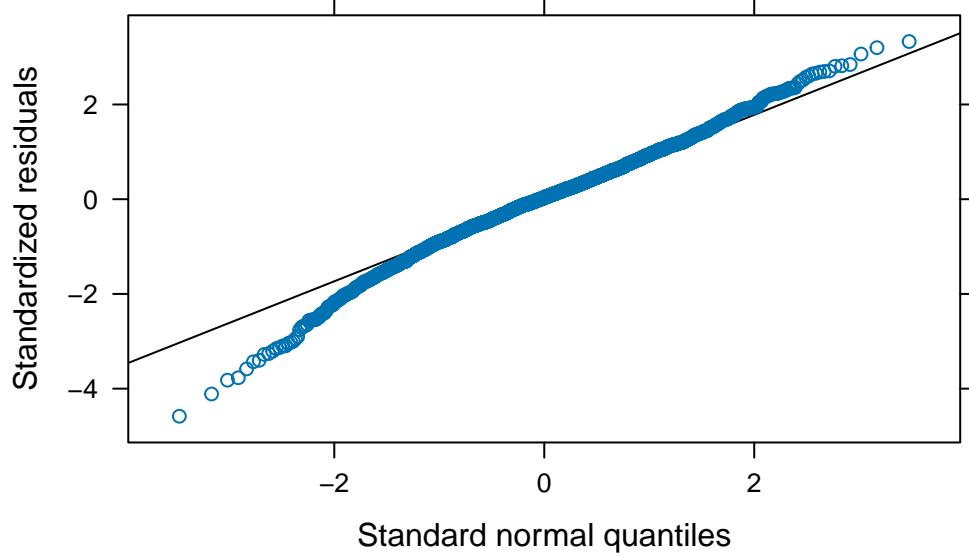
# check distribution of residuals
plot(model2test)
```



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



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



F Data Preparation for the Model of Textbook English vs. ‘real-world’ English

This script documents the steps taken to pre-process the data extracted from the Textbook English Corpus (TEC) and the three reference corpora that were ultimately entered in the comparative multi-dimensional model of Textbook English as compared to English outside the EFL classroom (Chapter 7).

F.1 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 thesis.

library(broom.mixed) # For checking singularity issues
library(car) # For recoding data
library(corrplot) # For the feature correlation matrix
library(cowplot) # For nice plots
library(emmeans) # Comparing group means of predicted values
library(GGally) # For ggpairs
library(gridExtra) # For making large faceted plots
library(here) # For ease of sharing
library(knitr) # Loaded to display the tables using the kable() function
library(lme4) # For mixed effects modelling
library(psych) # For various useful stats function, including KMO()
library(scales) # For working with colours
library(sjPlot) # For nice tabular display of regression models
library(tidyverse) # For data wrangling and plotting
library(visreg) # For nice visualisations of model results
select <- dplyr::select
filter <- dplyr::filter
```

F.2 Data import from MFTE outputs

The raw data used in this script comes from the matrices of mixed normalised frequencies as output by the [MFTE Perl v. 3.1](#) (Le Foll 2021a).

F.2.1 Spoken BNC2014

```
SpokenBNC2014 <- read.delim(here("data", "MFTE",
  "SpokenBNC2014_3.1_normed_complex_counts.tsv"), header = TRUE,
  stringsAsFactors = TRUE)

SpokenBNC2014$Series <- "Spoken BNC2014"
SpokenBNC2014$Level <- "Ref."
SpokenBNC2014$Country <- "Spoken BNC2014"
SpokenBNC2014$Register <- "Spoken BNC2014"
```

These normalised frequencies were computed on the basis of my own “John and Jill in Ivybridge” version of the Spoken BNC2014 with added full stops at speaker turns (see Appendix B for details). This corpus comprises of 1,251 texts, all of which were used in the following analyses.

F.2.2 Youth Fiction corpus

```
YouthFiction <- read.delim(here("data", "MFTE",
  "YF_sampled_500_3.1_normed_complex_counts.tsv"), header = TRUE,
  stringsAsFactors = TRUE)

YouthFiction$Series <- "Youth Fiction"
YouthFiction$Level <- "Ref."
YouthFiction$Country <- "Youth Fiction"
YouthFiction$Register <- "Youth Fiction"
```

These normalised frequencies were computed on the basis of the random samples of approximately 5,000 words of the books of the Youth Fiction corpus (for details of the works included in this corpus, see Appendix B). The sampling procedure is described in Section 4.3.2.4 of the book. This dataset consists of 1,191 files.

F.2.3 Informative Texts for Teens (InfoTeens) corpus

```
InfoTeen <- read.delim(here("data", "MFTE",
  "InfoTeen_3.1_normed_complex_counts.tsv"), header = TRUE,
  stringsAsFactors = TRUE)

# Removes three outlier files which should not have been included in the
# corpus as they contain exam papers only
InfoTeen <- InfoTeen |>
  filter(Filename!=".DS_Store" &
    Filename!="Revision_World_GCSE_10529068_wjec-level-law-past-papers.txt"
    &
    Filename!="Revision_World_GCSE_10528474_wjec-level-history-past-papers.txt"
    &
    Filename!="Revision_World_GCSE_10528472_edexcel-level-history-past-papers.txt")

InfoTeen$Series <- "Info Teens"
InfoTeen$Level <- "Ref."
InfoTeen$Country <- "Info Teens"
InfoTeen$Register <- "Info Teens"
```

Details of the composition of the Info Teens corpus can be found in Section 4.3.2.5 of the book. The version used in the present study comprises 1,411 texts.

F.3 Merging TEC and reference corpora data

F.3.1 Corpus size

These tables provide some summary statistics about the texts/files whose normalised feature frequencies were entered in the model of Textbook English vs. real-world English described in Chapter 7.

```
summary(ncounts$Subcorpus) |>
  kable(col.names = c("(Sub)corpus", "# texts"),
    format.args = list(big.mark = ","))
```

(Sub)corpus	# texts
Textbook Conversation	593
Textbook Fiction	285
Info Teens Ref.	1,411
Textbook Informative	364
Spoken BNC2014 Ref.	1,251
Youth Fiction Ref.	1,191

```
ncounts |>
  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 = ","),
        col.names = c("Register", "# texts/files", "# words", "mean # words
                     per text", "SD", "mean TTR"))
```

Register	# texts/files	# words	mean # words per text	SD	mean TTR
Conversation	1,844	13,804,196	7,486	8,690	0.40
Fiction	1,476	7,321,747	4,960	2,022	0.49
Informative	1,775	1,436,732	809	188	0.51

F.4 Data preparation for PCA

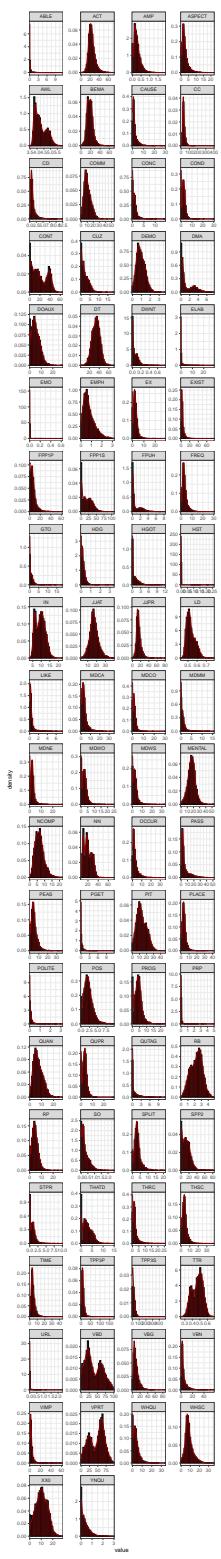
F.4.1 Feature distributions

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.

```
#ncounts <- readRDS(here("data", "processed", "counts3Reg.rds"))

ncounts |>
  select(-Words) |>
  keep(is.numeric) |>
```

```
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", "DensityPlotsAllVariables.svg"), width = 15, height =
  ↵ 49)
```

F.4.2 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 spelt 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, features with low MSA or communalities (see chunks below) were merged. Finally, features absent from more than third of texts were also excluded. For the comparative analysis of TEC and the reference corpora, the following linguistic features were excluded from the analysis due to low dispersion:

```
# Removal of meaningless feature: CD because numbers as digits were mostly
  ↵ removed from the textbooks, LIKE and SO because they are dustbin
  ↵ categories
ncounts <- ncounts |>
  select(-c(CD, LIKE, SO))

# Combine problematic features into meaningful groups whenever this makes
  ↵ linguistic sense
ncounts <- ncounts |>
  mutate(JJPR = JJPR + ABLE, ABLE = NULL) |>
  mutate(PASS = PGET + PASS, PGET = NULL) |>
  mutate(TPP3 = TPP3S + TPP3P, TPP3P = NULL, TPP3S = NULL) |> # Merged due to
  ↵ TPP3P having an individual MSA < 0.5
  mutate(FQTI = FREQ + TIME, FREQ = NULL, TIME = NULL) # Merged due to TIME
  ↵ communality < 0.2 (see below)

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

```

Feature	% texts with zero occurrences
PRP	85.34
URL	93.03
EMO	98.98
HST	99.55

```

# Drop variables with low document frequency
ncounts2 <- select(ncounts, -one_of(zero_features$rowname))

```

These feature removal operations resulted in a feature set of 71 linguistic variables.

F.4.3 Identifying 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
zcounts <- ncounts2 |>
  select(-Words) |>
  keep(is.numeric) |>
  scale()

```

```

# If necessary, remove any outliers at this stage.
data <- cbind(ncounts2[,1:8], as.data.frame(zcounts))
outliers <- data |>
  filter(if_any(where(is.numeric) & !Words, .fns = function(x){x > 8})) |>
  select(Filename, Corpus, Register, Words)

```

The following outlier texts were identified according to the above conditions and excluded in subsequent analyses.

```

# These are potential outlier texts :
outliers |>
  kable(col.names = c("Filename", "Corpus", "Register", "# words"))

```

Filename	Corpus	Register	# words
POC_4e_Spoken_0007.txt	Textbook	Eng	130
Solutions_Elementary_ELF_Spoken_0013.txt	Textbook	Eng	131
EIM_Starter_Informative_0004.txt	Textbook	Eng	133
GreenLine_1_Spoken_0003.txt	Textbook	Eng	130
Access_1_Spoken_0011.txt	Textbook	Eng	130
Achievers_B1_Informative_0003.txt	Textbook	Eng	130
EIM_Starter_Spoken_0002.txt	Textbook	Eng	130
GreenLine_1_Spoken_0008.txt	Textbook	Eng	130
JTT_3_Informative_0003.txt	Textbook	Eng	130
GreenLine_1_Spoken_0010.txt	Textbook	Eng	130
EIM_1_Spoken_0012.txt	Textbook	Eng	130
NGL_1_Spoken_0013.txt	Textbook	Eng	130
NGL_3_Spoken_0018.txt	Textbook	Eng	130
Solutions_Intermediate_Spoken_0029.txt	Textbook	Eng	130
NGL_1_Spoken_0012.txt	Textbook	Eng	130
GreenLine_1_Spoken_0006.txt	Textbook	Eng	130
GreenLine_2_Spoken_0004.txt	Textbook	Eng	130
Access_2_Spoken_0023.txt	Textbook	Eng	130
HT_4_Informative_0006.txt	Textbook	Eng	130
Solutions_Intermediate_Informative_0017.txt	Textbook	Eng	130
EIM_1_Spoken_0013.txt	Textbook	Eng	130
Solutions_Elementary_ELF_Spoken_0021.txt	Textbook	Eng	130
Solutions_Intermediate_Plus_Spoken_0022.txt	Textbook	Eng	130
Access_2_Spoken_0028.txt	Textbook	Eng	130
NGL_1_Spoken_0005.txt	Textbook	Eng	130

Filename	Corpus	Register	# words
Solutions_Elementary_ELF_Spoken_0016.txt	Textbook.Eng	Kid	Conversation871
Solutions_Pre-Intermediate_ELF_Spoken_0007.txt	Textbook.Eng	Kid	Conversation630
Solutions_Intermediate_Informative_0013.txt	Textbook.Eng	Informative	770
GreenLine_2_Spoken_0003.txt	Textbook.Eng	Kid	Conversation850
HT_4_Spoken_0010.txt	Textbook.Eng	Kid	Conversation727
Solutions_Elementary_Informative_0003.txt	Textbook.Eng	Informative	1051
Access_2_Informative_0001.txt	Textbook.Eng	Informative	655
Solutions_Elementary_Informative_0010.txt	Textbook.Eng	Informative	708
GreenLine_1_Informative_0001.txt	Textbook.Eng	Informative	731
Access_2_Spoken_0002.txt	Textbook.Eng	Kid	Conversation72
Solutions_Intermediate_Spoken_0019.txt	Textbook.Eng	Kid	Conversation1024
Access_3_Informative_0003.txt	Textbook.Eng	Informative	1000
Access_1_Spoken_0019.txt	Textbook.Eng	Kid	Conversation701
Access_2_Spoken_0013.txt	Textbook.Eng	Kid	Conversation981
Solutions_Intermediate_Plus_Informative_0014.txt	Textbook.Eng	Informative	537
Revision_World_GCSE_10525362_literary-terms.txt	Informative.T	Informative	790
Revision_World_GCSE_10528697_p6-physics-radioactive-materials.txt	Informative.T	Informative	1015
Science_Tech_Kinds_NZ_10382383_math.txt	Informative.T	Informative	522
Science_for_students_10064820_scientists-say-metabolism.txt	Informative.T	Informative	895
Science_Tech_Kinds_NZ_10382388_recycling.txt	Informative.T	Informative	666
History_Kids_BBC_10404337_go_further.txt	Informative.T	Informative	620
Science_Tech_Kinds_NZ_10382391_sports.txt	Informative.T	Informative	657
Teen_Kids_News_10402607_so-you-want-to-be-an-archivist.txt	Informative.T	Informative	763
Science_Tech_Kinds_NZ_10382234_biology.txt	Informative.T	Informative	843
Science_Tech_Kinds_NZ_10382372_astronomy.txt	Informative.T	Informative	900
Dogo_News_file10060404_banana-plant-extract-may-be-the-key-to-slower-melting-ice-cream.txt	Informative.T	Informative	611
Science_Tech_Kinds_NZ_10382667_countries.txt	Informative.T	Informative	717
Quatr_us_file10390777_quick-summary-geological-erashtm.txt	Informative.T	Informative	643
Science_Tech_Kinds_NZ_10382873_physics.txt	Informative.T	Informative	722
Science_Tech_Kinds_NZ_10382382_light.txt	Informative.T	Informative	639
Factmonster_10053687_august-13.txt	Informative.T	Informative	523
Revision_World_GCSE_10526703_limited-companies.txt	Informative.T	Informative	714
Revision_World_GCSE_10529637_transition-metals.txt	Informative.T	Informative	787
Quatr_us_10390856_early-african-history.htm	Informative.T	Informative	1136
History_Kids_BBC_10401873_ff6_sicilylandingss.txt	Informative.T	Informative	813

Filename	Corpus	Register	# words
Quatr_us_10394250_harappan.txt	Informative.T	Informative	651
Ducksters_10398301_iraqphp.txt	Informative.T	Informative	657
History_Kids_BBC_10403171_death_sakkara_gallery_04s.txt	Informative.T	Informative	844
Revision_World_GCSE_10528246_agricultural-change.txt	Informative.T	Informative	789
Revision_World_GCSE_10528086_uk-government-judiciary.txt	Informative.T	Informative	1019
Revision_World_GCSE_10529794_definitions.txt	Informative.T	Informative	904
Encyclopedia_Kinds_au_10085347_Nobel_Prize_in_Chemis	Informative.T	Informative	598
Science_for_students_10064875_questions-big-melt-earth-ice-sheets-are-under-attack.txt	Informative.T	Informative	685
Teen_Kids_News_10403301_golden-globe-winners-2019-the-complete-list.txt	Informative.T	Informative	800
Science_Tech_Kinds_NZ_10382201_projects.txt	Informative.T	Informative	947
Revision_World_GCSE_10529753_probability.txt	Informative.T	Informative	816
Encyclopedia_Kinds_au_10085531_Complex_analysis.txt	Informative.T	Informative	735
History_Kids_BBC_10401890_ff7_ddays.txt	Informative.T	Informative	759
History_Kids_BBC_10403434s.txt	Informative.T	Informative	732
History_Kids_BBC_10401872_ff6_italys.txt	Informative.T	Informative	786
Science_Tech_Kinds_NZ_10382371_amazing.txt	Informative.T	Informative	629
Quatr_us_10391129_athabascan.txt	Informative.T	Informative	637
Encyclopedia_Kinds_au_10085355_20th_century.txt	Informative.T	Informative	864
Dogo_News_10060755_luxury-space-hotel-promises-guests-a-truly-out-of-this-world-vacation.txt	Informative.T	Informative	722
Revision_World_GCSE_10528072_nationalism-practice.txt	Informative.T	Informative	776
Quatr_us_10390861_quatr-us-privacy-policyhtm.txt	Informative.T	Informative	960
History_Kids_BBC_10401909_ff7_bulges.txt	Informative.T	Informative	732
History_kids_10381259_timeline-of-mesopotamia.txt	Informative.T	Informative	768
Revision_World_GCSE_10528123_gender-written-textual-analysis-framework.txt	Informative.T	Informative	905
Science_Tech_Kinds_NZ_10386406_floods.txt	Informative.T	Informative	580
Revision_World_GCSE_10529693_advantages.txt	Informative.T	Informative	782
Science_Tech_Kinds_NZ_10382378_geography.txt	Informative.T	Informative	761
Science_Tech_Kinds_NZ_10382374_earth.txt	Informative.T	Informative	726
Science_for_students_10066286_watering-plants-wastewater-can-spread-germs.txt	Informative.T	Informative	836
Science_Tech_Kinds_NZ_10382393_water.txt	Informative.T	Informative	856
World_Dteen_10406069_website_policies.txt	Informative.T	Informative	995
Science_Tech_Kinds_NZ_10382384_metals.txt	Informative.T	Informative	669

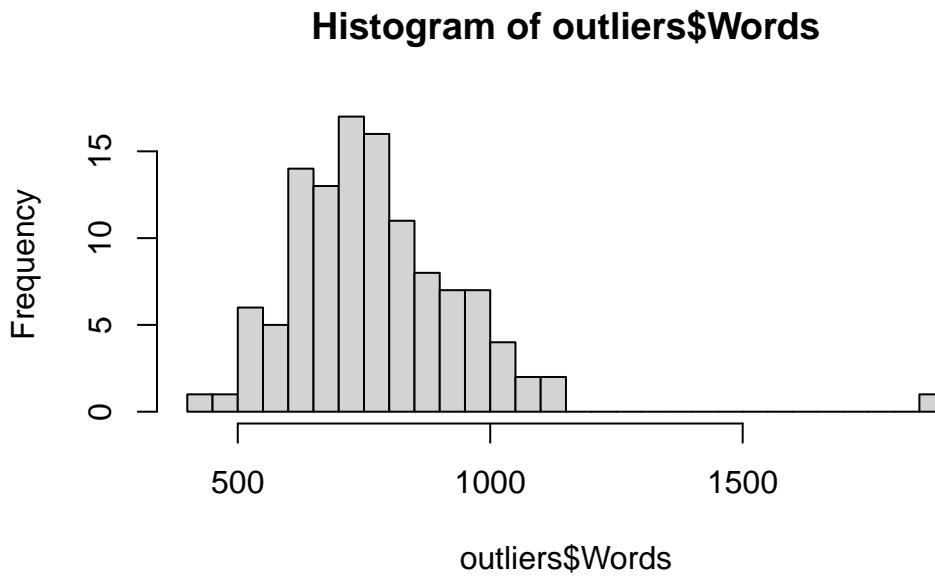
Filename	Corpus	Register	# words
Dogo_News_10062028_puppy-bowl-14-promises-viewers-a-paw-some-time-on-super-bowl-sunday.txt	Informative.T	Informative	581
History_Kids_BBC_10404730_go_further.txt	Informative.T	Informative	611
Science_Tech_Kinds_NZ_10382385_nature.txt	Informative.T	Informative	722
Science_for_students_10065015_scientists-say-dna-sequencing.txt	Informative.T	Informative	953
Quatr_us_file10390817_conifers-pine-trees-gymnosperms.htm.txt	Informative.T	Informative	533
TweenTribute_10051509_it-true-elephants-cant-jump.txt	Informative.T	Informative	790
Revision_World_GCSE_10528494_application-software.txt	Informative.T	Informative	855
Revision_World_GCSE_10529581_different-types-questions-examinations.txt	Informative.T	Informative	742
Dogo_News_10061669_the-chinese-city-of-chengdu-may-soon-be-home-to-multiple-moons.txt	Informative.T	Informative	614
Ducksters_10398306_geography_of_ancient_chinaphp.txt	Informative.T	Informative	638
Science_for_students_10065144_scientists-say-multiverse.txt	Informative.T	Informative	712
Science_Tech_Kinds_NZ_10382211_images.txt	Informative.T	Informative	793
Factmonster_10053754_may-18.txt	Informative.T	Informative	497
World_Dteen_10406047_AboutWORLDteen.txt	Informative.T	Informative	1053
Ducksters_10398078_first_new_dealphp.txt	Informative.T	Informative	649
Revision_World_GCSE_10526926_economies-scale.txt	Informative.T	Informative	621
Factmonster_10053201_september-03.txt	Informative.T	Informative	445
Science_Tech_Kinds_NZ_10387183_calciumcarbonates.txt	Informative.T	Informative	804
Science_Tech_Kinds_NZ_10382380_health.txt	Informative.T	Informative	694
Revision_World_GCSE_10529587_sources-finance.txt	Informative.T	Informative	665
Quatr_us_10393444_fishing.txt	Informative.T	Informative	656
Ducksters_10398315_glossary_and_termsphp.txt	Informative.T	Informative	684
S5AA.txt	Spoken.BNC2014	Conversation	869

We check that that outlier texts are not particularly long or short texts by looking at the distribution of text/file length of the outliers.

```
summary(outliers$Words)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
445.0	655.5	751.0	773.6	860.0	1869.0

```
hist(outliers$Words, breaks = 30)
```



We also check the distribution of outlier texts across the four corpora. The majority come from the Info Teens corpus, though quite a few are also from the TEC.

```
summary(outliers$Corpus) |>  
kable(col.names = c("(Sub)corpus", "# outlier texts"))
```

(Sub)corpus	# outlier texts
Textbook.English	40
Informative.Teens	74
Spoken.BNC2014	1
Youth.Fiction	0

```
# Report on the manual check of a sample of these outliers:  
  
# Encyclopedia_Kinds_au_10085347_Nobel_Prize_in_Chemistry.txt is essentially  
# a list of Nobel prize winners but with some additional information. In  
# other words, not a bad representative of the type of texts of the Info  
# Teen corpus.  
# Solutions_Elementary_ELF_Spoken_0013 --> Has a lot of "going to"  
# constructions because they are learnt in this chapter but is otherwise a  
# well-formed text.
```

```

# Teen_Kids_News_10403972_a-brief-history-of-white-house-weddings --> No
↳ issues
# Teen_Kids_News_10403301_golden-globe-winners-2019-the-complete-list -->
↳ Similar to the Nobel prize laureates text.
# Revision_World_GCSE_10528123_gender-written-textual-analysis-framework -->
↳ Text includes bullet points tokenised as the letter "o" but otherwise a
↳ fairly typical informative text.

# Removing the outliers at the request of the reviewers (but comparisons of
↳ models including the outliers showed that the results are very similar):
ncounts3 <- ncounts2 |>
  filter(!Filename %in% outliers$Filename)

#saveRDS(ncounts3, here("data", "processed", "ncounts3_3Reg.rds")) # Last
↳ saved 6 March 2024

```

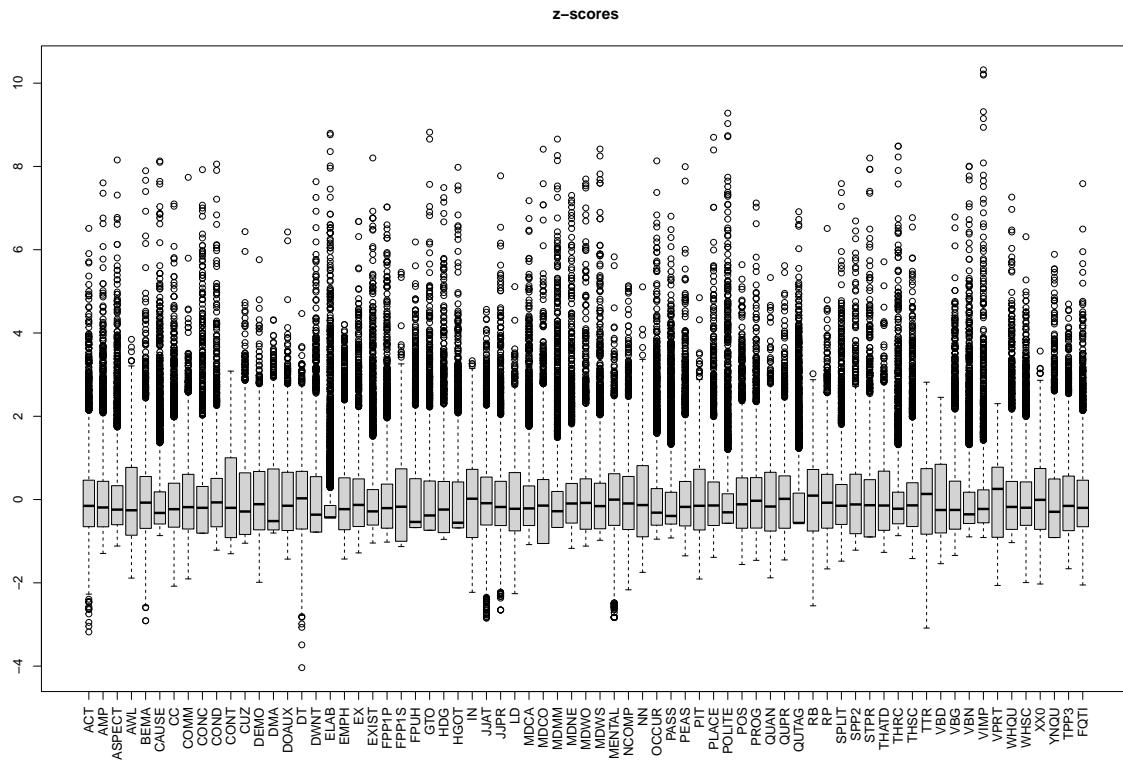
This resulted in 4,980 texts/files being included in the comparative model of Textbook English vs. ‘real-world’ English. These standardised feature frequencies were distributed as follows:

```

zcounts3 <- ncounts3 |>
  select(-Words) |>
  keep(is.numeric) |>
  scale()

boxplot(zcounts3, las = 3, main = "z-scores") # Slow

```



F.4.4 Signed log transformation

A signed logarithmic transformation was applied to (further) deskew the feature distributions (see 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>.

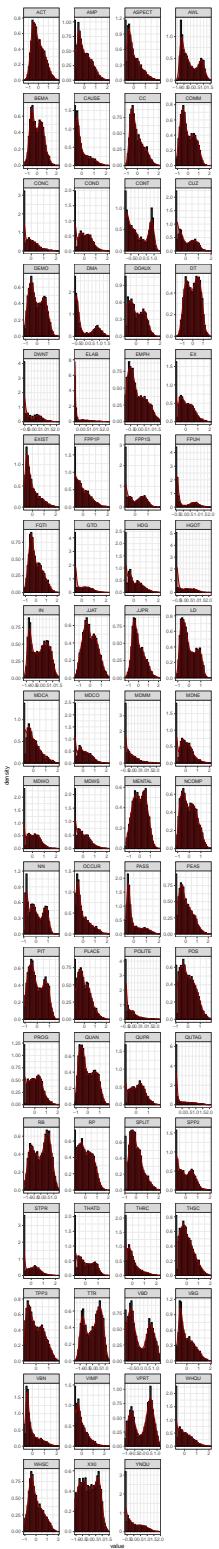
```
signed.log <- function(x) {sign(x)*log(abs(x)+1)}

# Standardise first, then sign log transform
zlogcounts <- signed.log(zcounts3)
```

The new feature distributions are visualised below.

```
zlogcounts |>
  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.svg"), width = 15,
  ↴ height = 49)
```

F.4.5 Merging of data for MDA

```
zlogcounts <- readRDS(here("data", "processed", "zlogcounts_3Reg.rds"))
#nrow(zlogcounts)
#colnames(zlogcounts)

ncounts3 <- readRDS(here("data", "processed", "ncounts3_3Reg.rds"))
#nrow(ncounts3)
#colnames(ncounts3)

data <- cbind(ncounts3[,1:8], as.data.frame(zlogcounts))
#saveRDS(data, here("data", "processed", "datazlogcounts_3Reg.rds")) # Last
  ↴ saved 16 March 2024
```

The final dataset comprises of 4,980 texts/files, divided as follows:

(Sub)corpus	# texts/files
Textbook Conversation	565
Textbook Fiction	285
Info Teens Ref.	1337
Textbook Informative	352
Spoken BNC2014 Ref.	1250
Youth Fiction Ref.	1191

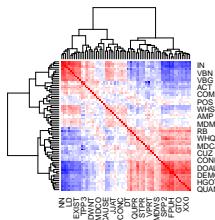
F.5 Testing factorability of data

F.5.1 Visualisation of feature correlations

We begin by visualising the correlations of the transformed feature frequencies using the `heatmap` function of the `stats` library. Negative correlations are rendered in blue; 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.png"), width = 30, height= 30, units =
#  "cm", res = 300)
heatmap(cor(zlogcounts),
        symm=TRUE,
        zlim=c(-1,1),
        col=cor.colours,
        margins=c(7,7))
```



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

F.5.2 Collinearity

As a result of the normalisation unit of finite verb phrases for verb-based features, the present tense (VPRT) and past tense (VBD) variables are correlated to a very high degree:

```
cor(data$VPRT, data$VBD) |> round(2)
```

```
[1] -0.97
```

We therefore remove the least marked of the pair of collinear variables: VPRT.

```
data <- data |>
  select(-c(VPRT))
```

F.5.3 MSA

```
kmo <- KMO(data[,9:ncol(data)]) # The first eight columns contain metadata.
```

The overall MSA value of the dataset is 0.95. The features have the following individual MSA values (ordered from lowest to largest):

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

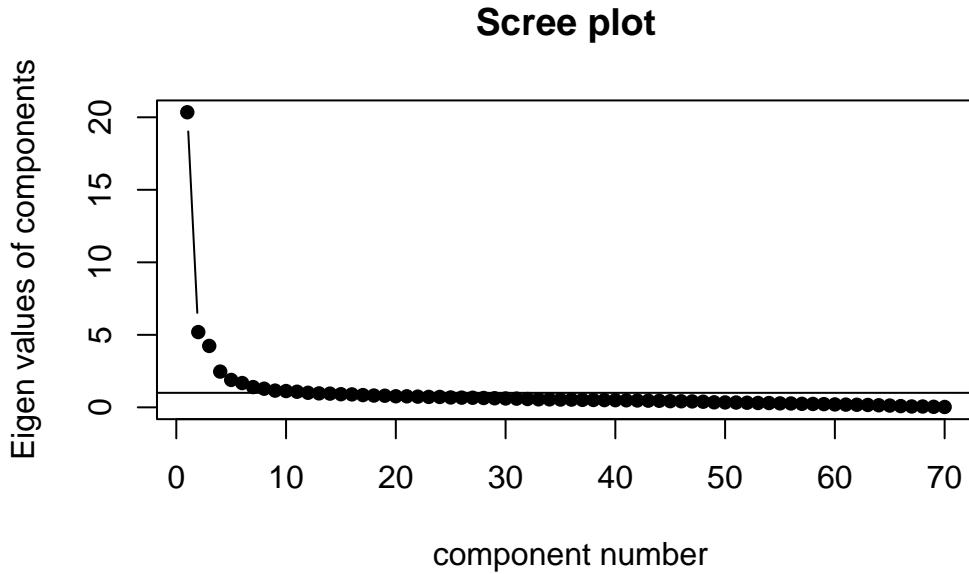
	AMP	COMM	POS	TPP3	JJPR	PLACE	SPLIT	DT	JJAT	VIMP	MDC0
0.67	0.69	0.70	0.74	0.76	0.82	0.83	0.83	0.84	0.84	0.84	0.85
RP	EX	THSC	LD	NCOMP	BEMA	MDWS	FQTI	FPP1P	MDCA	ACT	
0.85	0.85	0.86	0.87	0.88	0.88	0.89	0.89	0.89	0.89	0.89	0.89
MENTAL	VBD	FPP1S	MDMM	PEAS	CONC	MDW0	THRC	NN	COND	PROG	
0.91	0.91	0.91	0.91	0.91	0.93	0.93	0.94	0.94	0.95	0.95	0.95
CC	SPP2	RB	DWNT	MDNE	WHSC	CONT	QUPR	XX0	CAUSE	WHQU	
0.95	0.95	0.95	0.95	0.95	0.96	0.96	0.96	0.96	0.96	0.96	0.96
VBG	AWL	POLITE	PASS	PIT	DOAUX	ELAB	ASPECT	DMA	DEMO	HDG	
0.96	0.96	0.96	0.96	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
IN	FPUH	OCCUR	CUZ	EMPH	YNQU	QUAN	TTR	QUTAG	THATD	VBN	
0.97	0.97	0.97	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
EXIST	STPR	GTO	HGOT								
0.98	0.99	0.99	0.99								

We aim to remove features with an individual MSA < 0.5. All features have individual MSAs of > 0.5 (but only because TPP3P was merged into a broader category in an earlier chunk).

F.5.4 Scree plot

Six components were originally retained on the basis of the following screeplot, though only the first four were found to be interpretable and were therefore included in the model.

```
# png(here("plots", "screeplot-TEC-Ref_3Reg.png"), width = 20, height= 12,
  ↴ units = "cm", res = 300)
scree(data[,9:ncol(data)], factors = FALSE, pc = TRUE) #
```



```
# dev.off()

# Perform PCA
pca1 <- psych::principal(data[9:ncol(data)],
                           nfactors = 6)
```

F.5.5 Communalities

If features with final communalities of < 0.2 are removed, TIME would have to be removed. TIME was therefore merged with FREQ in an earlier chunk so that now all features have final communalities of > 0.2 (note that this is a very generous threshold!).

```
pca1$communality |> sort() |> round(2)
```

DWNT	STPR	CONC	FQTI	POS	ASPECT	MDNE	FPP1P	PROG	MDCO	MDMM
0.22	0.23	0.23	0.23	0.24	0.25	0.27	0.28	0.29	0.32	0.32
MDWO	SPLIT	MDWS	PEAS	QUPR	AMP	PLACE	HDG	COMM	CAUSE	EX
0.32	0.33	0.34	0.35	0.35	0.35	0.37	0.38	0.38	0.38	0.38
THSC	OCCUR	WHSC	THRC	JJAT	COND	MENTAL	ACT	VIMP	ELAB	EXIST
0.40	0.40	0.42	0.43	0.44	0.44	0.45	0.45	0.46	0.46	0.46
JJPR	NCOMP	RP	GTO	DEMO	MDCA	POLITE	CUZ	CC	WHQU	TPP3

0.46	0.48	0.49	0.50	0.50	0.52	0.52	0.53	0.57	0.58	0.58
VBG	THATD	PIT	BEMA	FPP1S	DT	HGOT	RB	VBN	QUTAG	EMPH
0.60	0.60	0.61	0.61	0.61	0.61	0.62	0.62	0.64	0.64	0.64
PASS	XX0	QUAN	SPP2	DOAUX	TTR	YNQU	VBD	LD	FPUH	IN
0.65	0.65	0.67	0.68	0.69	0.71	0.74	0.78	0.81	0.83	0.86
CONT	DMA	AWL	NN							
0.89	0.89	0.91	0.93							

```
#saveRDS(data, here("data", "processed", "dataforPCA.rds")) # Last saved on 6
← March 2024
```

The final dataset entered in the analysis described in Chapter 7 therefore comprises 4,980 texts/files, each with logged standardised normalised frequencies for 70 linguistic features.

G Data Analysis for the Model of Textbook English vs. ‘real-world’ English

This script documents the analysis of data from the TEC and reference corpus data (as pre-processed in Appendix F) to arrive at the multi-dimensional model of Textbook Englisch vs. ‘real-world’ English described in Chapter 7. It generates all of the statistics and plots included in the book, as well as many others that were used in the analysis, but were not included in the book for reasons of space.

G.1 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 thesis.

library(caret) # For its confusion matrix function
library(cowplot) # For its plot themes
library(DescTools) # For 95% CI
library(emmeans) # For the emmeans function
library(factoextra) # For circular graphs of variables
library(gtsummary) # For nice table of summary statistics (optional)
library(gridExtra) # For Fig. 35
library(here) # For dynamic file paths
library(ggthemes) # For theme of factoextra plots
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(pca3d) # For 3-D plots (not rendered in exports)
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
```

```
source(here("R_rainclouds.R")) # For geom_flat_violin rainplots
```

G.2 Conducting the PCA

We first import the full dataset (see Appendix F for data preparation steps).

The following chunks can be used to perform the MDA on various subsets of the data (see also Section 10.1.1 in the book).

- i. Subset of the data that excludes the lower-level textbooks:

```
data <- readRDS(here("processed_data", "dataforPCA.rds")) |>  
  filter(Level != "A" & Level != "B") |>  
  droplevels()  
summary(data$Level)
```

- i. Subset of the data that includes only one **Country**‘ subcorpus of the TEC (note that a detailed analysis of the German subcorpus can be found in (Le Foll, n.d.)):

```
data <- readRDS(here("processed_data", "dataforPCA.rds")) |>  
#filter(Country != "France" & Country != "Germany") |> # Spain only  
#filter(Country != "France" & Country != "Spain") |> # Germany only  
filter(Country != "Spain" & Country != "Germany") |> # France only  
droplevels()  
summary(data$Country)
```

- i. Random subsets of the data to test the stability of the model proposed in Chapter 7. Re-running this line will generate a new subset of 2/3 of the texts randomly sampled. `set.seed(13)` was used for the analyses reported on in Section 10.1.1.

```
set.seed(13)  
data <- readRDS(here("processed_data", "dataforPCA.rds")) |>  
  slice_sample(n = 4980*0.6, replace = FALSE)  
nrow(data)  
data$Filename[1:4]  
#Using the set.seed(13), these should be:  
#[1] HT_4_Spoken_0009.txt  
#[2] Solutions_Intermediate_Plus_Spoken_0020.txt  
#[3] 141_PRATCHETT1992DW13GODS_4.txt  
#[4] Achievers_B2_Informative_0004.txt
```

G.3 Plotting PCA results

G.3.1 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>.

```
# Data preparation for 3D plots
colnames(data) # Checking that the features start at the 9th column
pca <- prcomp(data[,9:ncol(data)], scale.=FALSE) # All quantitative variables
# that contribute to the model
register <- factor(data[,"Register"])
corpus <- factor(data[,"Corpus"])
subcorpus <- factor(data[,"Subcorpus"])

library(pca3d)

pca3d(pca, group = subcorpus,
       components = 1:3,
       components = 4:6,
       show.plane=FALSE,
       col = col6,
       shape = shapes6,
       radius = 0.7,
       legend = "right")

snapshotPCA3d(here("plots", "PCA_TxB_3Ref_3Dsnapshot.png"))

# Alternative visualisation, looking at all three Textbook English registers
# in one colour

pca3d(pca, group = corpus,
       show.plane=FALSE,
       components = 1:3,
       col = col4,
       shape = shapes4,
       radius = 0.7,
       legend = "right")
```

G.4 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 be wrangled first.

G.4.1 Data wrangling for PCAtools

```
# Data wrangling
data2 <- data |>
  mutate(Source = recode_factor(Corpus, Textbook.English = "Textbook English
    ↵ (TEC)", Informative.Teens = "Reference corpora", Spoken.BNC2014 =
    ↵ "Reference corpora", Youth.Fiction = "Reference corpora")) |>
  mutate(Corpus = fct_relevel(Subcorpus, "Info Teens Ref.", after = 9)) |>
  relocate(Source, .after = "Corpus") |>
  droplevels()

# colnames(data2)
data2meta <- data2[,1:9]
rownames(data2meta) <- data2meta$Filename
data2meta <- data2meta |> select(-Filename)
# head(data2meta)
rownames(data2) <- data2$Filename
data2num <- as.data.frame(base::t(data2[,10:ncol(data2)]))
# data2num[1:5,1:5] # Check data frame format is correct by comparing to
  ↵ output of head(data2meta) above

p <- PCAtools::pca(data2num,
  metadata = data2meta,
  scale = FALSE)
```

The cumulative proportion of variance in the dataset explained the first four components explain 47.15%.

G.4.2 Pairs plot

This chunk produces a scatterplot matrix of combinations of the four dimensions of the model of Textbook English vs. ‘real-world’ English. 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.

```

# For five TEC registers
# colkey = c(`Spoken BNC2014 Ref.`="#BD241E", `Info Teens Ref.`="#15274D",
#   `Youth Fiction Ref.`="#267226", `Textbook Fiction`="#A18A33", `Textbook
#   Conversation`="#F9B921", `Textbook Informative` = "#722672", `Textbook
#   Instructional` = "grey", `Textbook Personal` = "black")

# For three TEC registers
# summary(data2$Corpus)
colkey = c(`Spoken BNC2014 Ref.`="#BD241E", `Info Teens Ref.`="#15274D",
  `Youth Fiction Ref.`="#267226", `Textbook Fiction`="#A18A33", `Textbook
  Conversation`="#F9B921", `Textbook Informative` = "#722672")

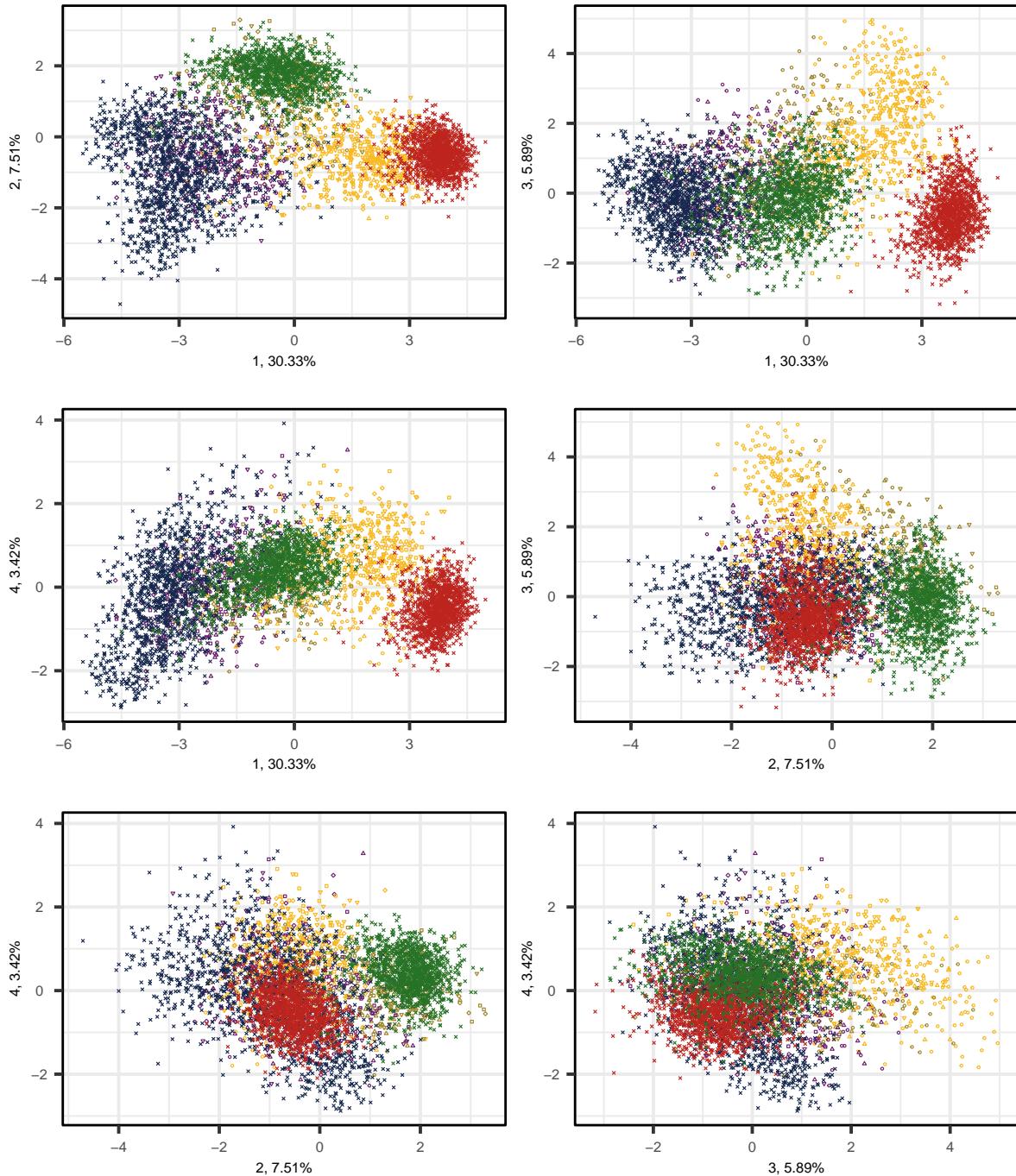
#summary(data2$Source)
#shapekey = c(`Textbook English (TEC)`=6, `Reference corpora`=1)

# summary(data2$Level)
shapekey = c(A=1, B=2, C=6, D=0, E=5, `Ref.`=4)

## Warning: this can be very slow! Open in extra zoomed out window!

#png(here("plots", "PCA_3Ref_pairsplot.png"), width = 12, height= 19, units =
#  "cm", res = 300)
PCAtools::pairsplot(p,
  triangle = FALSE,
  components = 1:4,
  ncol = 2,
  nrow = 3,
  pointSize = 0.6,
  shape = "Level",
  shapekey = shapekey,
  lab = NULL, # Otherwise will try to label each data point!
  colby = "Corpus",
  legendPosition = "none",
  marginingaps = unit(c(0.2, 0.2, 0.8, 0.2), "cm"),
  colkey = colkey)

```



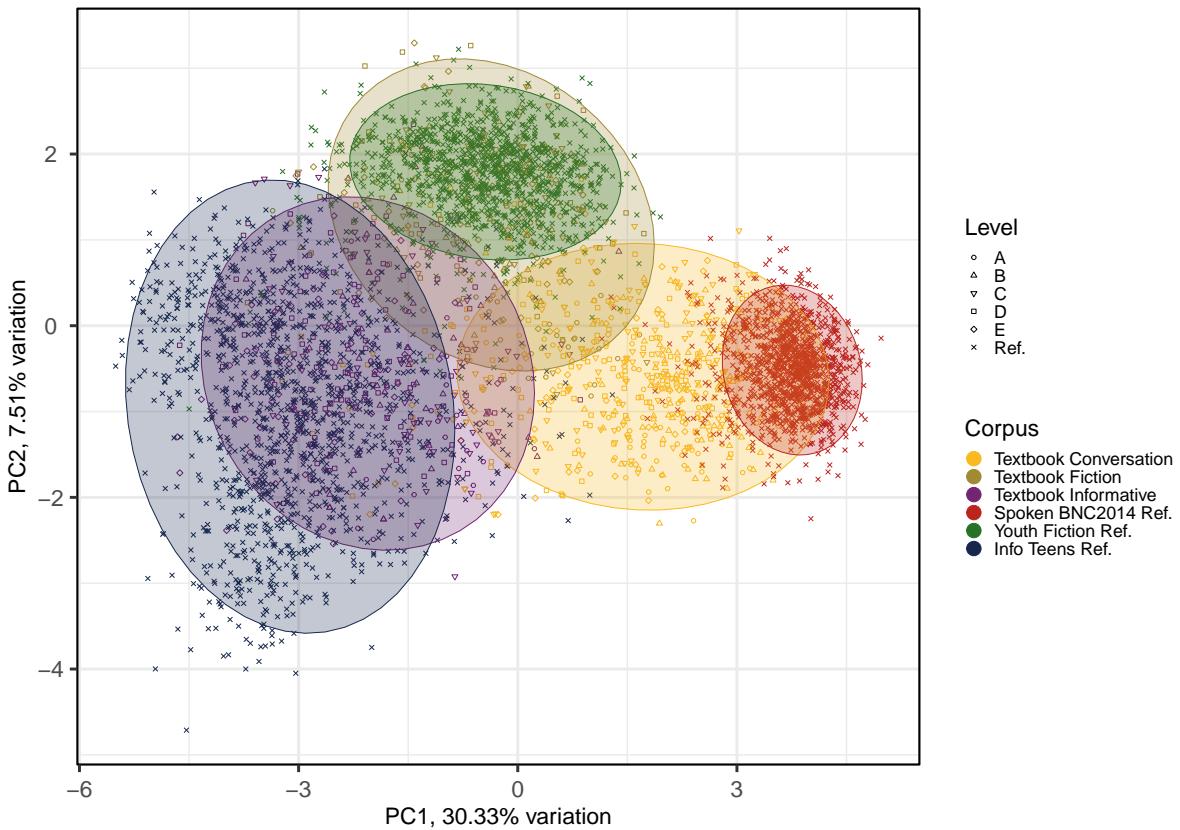
```
#dev.off()
#ggsave(here("plots", "PCA_TxB_pairsplot.svg"), width = 6, height = 10)
# Note that the legend has to be added manually (it was taken from the biplot
# code below).
```

G.4.3 Bi-plots

Biplots are used to more closely examine the position of texts on just two dimensions.

```
# These settings (with legendPosition = "top") were used to generate the
# legend for the scatterplot matrix above:
#png(here("plots", "PCA_3Ref_Biplot_PC1_PC2test.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!
  colby = "Corpus",
  pointSize = 1.3,
  colkey = colkey,
  shape = "Level",
  shapekey = shapekey,
  xlim = c(min(p$rotated[, "PC1"]), max(p$rotated[, "PC1"])),
  ylim = c(min(p$rotated[, "PC2"]), max(p$rotated[, "PC2"])),
  showLoadings = FALSE,
  ellipse = TRUE,
  axisLabSize = 18,
  legendPosition = 'right',
  legendTitleSize = 18,
  legendLabSize = 14,
  legendIconSize = 5) +
  theme(plot.margin = unit(c(0,0,0,0.2), "cm"))
```



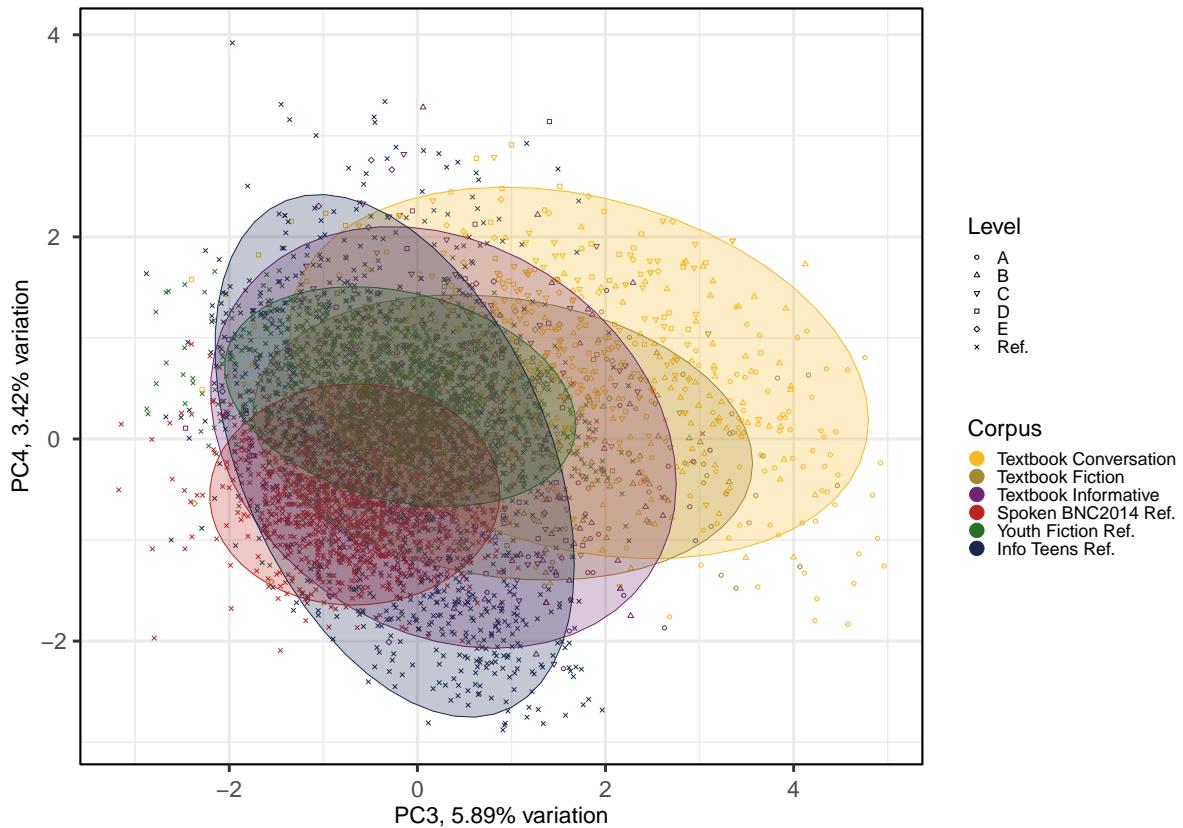
```
#ggsave(here("plots", "PCA_Ref3TxB_BiplotPC1_PC2.svg"), width = 12, height =
  8)

# Biplots to examine components more carefully
PCAtools::biplot(p,
  x = "PC3",
  y = "PC4",
  lab = NULL, # Otherwise will try to label each data point!
  colby = "Corpus",
  pointSize = 1.2,
  colkey = colkey,
  shape = "Level",
  shapekey = shapekey,
  xlim = c(min(p$rotated[, "PC3"]), max(p$rotated[, "PC3"])),
  ylim = c(min(p$rotated[, "PC4"]), max(p$rotated[, "PC4"])),
  showLoadings = FALSE,
```

```

ellipse = TRUE,
axisLabSize = 18,
legendPosition = 'right',
legendTitleSize = 18,
legendLabSize = 14,
legendIconSize = 5) +
theme(plot.margin = unit(c(0,0,0,0.2), "cm"))

```



```
#ggsave(here("plots", "PCA_Ref3Tx_B_BiplotPC3_PC4.svg"), width = 12, height =
  8)
```

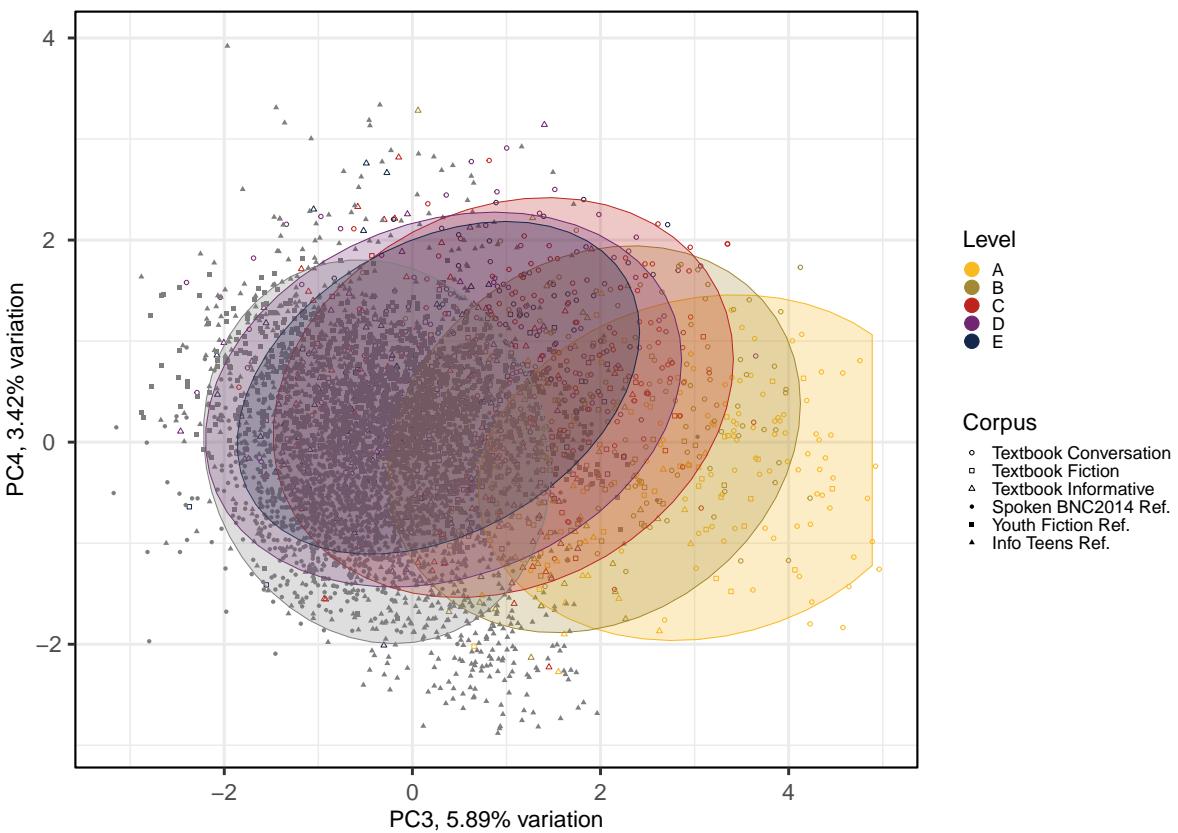
The colours and corresponding ellipses can be used to visualise different clusters and patterns. In the following, we change the colour of the points and the ellipses to represent the texts' target proficiency levels instead of the register, allowing for a different interpretation of the model.

```

# Biplot with ellipses for Level rather than Register
colkeyLevels = c(A="#F9B921", B="#A18A33", C="#BD241E", D="#722672",
                 E="#15274D", `Ref. data` = "darkgrey")
shapekeyLevels = c(`Spoken BNC2014 Ref.`=16, `Info Teens Ref.`=17, `Youth
                   Fiction Ref.`=15, `Textbook Fiction`=0, `Textbook Conversation`=1,
                   `Textbook Informative`=2)

PCAtools::biplot(p,
                  x = "PC3",
                  y = "PC4",
                  lab = NULL, # Otherwise will try to label each data point!
                  colby = "Level",
                  pointSize = 1.3,
                  colkey = colkeyLevels,
                  shape = "Corpus",
                  shapekey = shapekeyLevels,
                  xlim = c(min(p$rotated[, "PC3"]), max(p$rotated[, "PC3"])),
                  ylim = c(min(p$rotated[, "PC4"]), max(p$rotated[, "PC4"])),
                  showLoadings = FALSE,
                  ellipse = TRUE,
                  axisLabSize = 18,
                  legendPosition = 'right',
                  legendTitleSize = 18,
                  legendLabSize = 14,
                  legendIconSize = 5) +
theme(plot.margin = unit(c(0,0,0,0.2), "cm"))

```



```
#ggsave(here("plots", "PCA_Ref3TxB_BiplotPC3_PC4_levels.svg"), width = 12,
        height = 8)
```

G.5 Feature contributions (loadings) on each component

```
#data <- readRDS(here("processed_data", "dataforPCA.rds"))
pca <- prcomp(data[,9:ncol(data)], scale.=FALSE) # All quantitative variables
# to be included in the model

# 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])
```

```
# Table of loadings with no minimum threshold applied
loadings |>
  round(2) |>
  kable()
```

	PC1	PC2	PC3	PC4
ACT	-0.10	0.01	-0.01	0.12
AMP	0.00	-0.05	-0.05	0.09
ASPECT	-0.08	0.10	-0.01	0.00
AWL	-0.21	-0.13	-0.06	-0.01
BEMA	0.08	-0.24	0.06	-0.02
CAUSE	-0.08	-0.12	0.06	0.14
CC	-0.14	-0.13	-0.09	-0.09
COMM	-0.03	0.19	0.03	0.20
CONC	-0.03	-0.03	-0.18	-0.06
COND	0.08	-0.02	-0.17	0.23
CONT	0.22	-0.05	0.03	0.00
CUZ	0.10	-0.14	-0.20	-0.18
DEMO	0.15	-0.12	-0.08	-0.04
DMA	0.20	-0.09	-0.04	-0.16
DOAUX	0.18	-0.02	0.06	0.00
DT	0.08	0.16	-0.24	-0.03
DWNT	-0.04	0.10	-0.12	0.09
ELAB	-0.07	-0.17	-0.04	0.07
EMPH	0.17	-0.07	-0.09	-0.03
EX	0.06	-0.02	-0.04	0.00
EXIST	-0.13	-0.09	-0.05	0.00
FPP1P	0.08	-0.02	0.09	0.19
FPP1S	0.17	0.06	0.06	0.08
FPUH	0.18	-0.10	-0.05	-0.19
GTO	0.14	0.00	-0.04	0.00
HDG	0.10	-0.07	-0.14	-0.12
HGOT	0.16	-0.05	-0.01	-0.11
IN	-0.21	-0.01	-0.08	0.01
JJAT	-0.05	-0.13	-0.25	0.07
JJPR	-0.03	-0.17	-0.03	0.21
LD	-0.17	-0.16	0.13	-0.04
MDCA	0.05	-0.21	0.11	0.22
MDCO	0.00	0.19	-0.15	0.10
MDMM	-0.02	-0.10	-0.14	0.17
MDNE	0.06	-0.02	-0.06	0.22

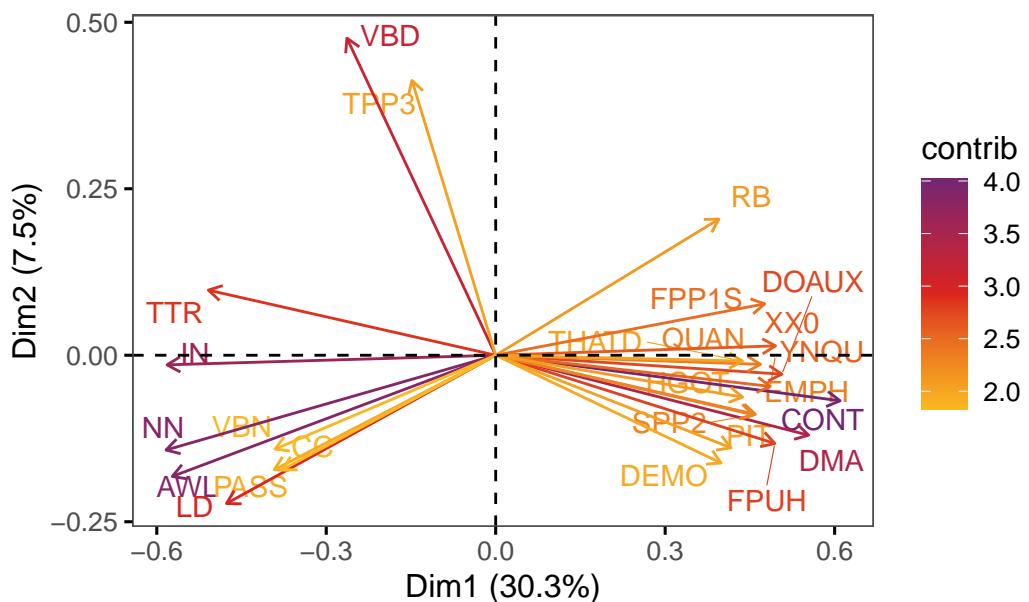
	PC1	PC2	PC3	PC4
MDWO	0.07	0.11	-0.18	0.04
MDWS	0.06	-0.02	-0.01	0.25
MENTAL	0.11	-0.02	-0.05	0.16
NCOMP	0.00	-0.27	-0.05	0.03
NN	-0.21	-0.10	0.10	-0.07
OCCUR	-0.13	-0.02	-0.05	-0.06
PASS	-0.14	-0.13	-0.09	-0.10
PEAS	-0.06	0.12	-0.19	0.12
PIT	0.15	-0.10	-0.15	-0.07
PLACE	0.02	0.09	0.09	0.07
POLITE	0.09	0.00	0.20	0.11
POS	0.02	0.09	0.04	-0.05
PROG	0.09	0.08	-0.04	0.15
QUAN	0.17	-0.01	-0.16	0.01
QUPR	0.08	0.11	-0.12	0.21
QUTAG	0.15	-0.04	-0.07	-0.15
RB	0.14	0.15	-0.18	0.07
RP	-0.01	0.22	-0.09	0.15
SPLIT	-0.03	-0.11	-0.21	0.08
SPP2	0.17	-0.07	0.10	0.16
STPR	0.10	0.01	0.01	-0.04
THATD	0.16	-0.01	-0.14	-0.02
THRC	-0.05	-0.17	-0.15	-0.02
THSC	-0.02	-0.08	-0.27	0.07
TTR	-0.19	0.07	-0.02	0.16
VBD	-0.10	0.35	-0.05	-0.20
VBG	-0.14	-0.02	-0.14	0.12
VBN	-0.14	-0.10	-0.08	-0.07
VIMP	0.01	-0.07	0.21	0.21
WHQU	0.13	-0.02	0.20	0.07
WHSC	-0.09	-0.10	-0.20	0.05
XX0	0.18	0.01	-0.06	0.06
YNQU	0.18	-0.03	0.14	-0.02
TPP3	-0.05	0.30	-0.04	-0.15
FQTI	-0.07	0.03	0.01	0.14

```
#clipr::write_last_clip()
```

G.6 Graphs of features of that contribute most to each component/dimension

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(contrib = 25),
  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())
```



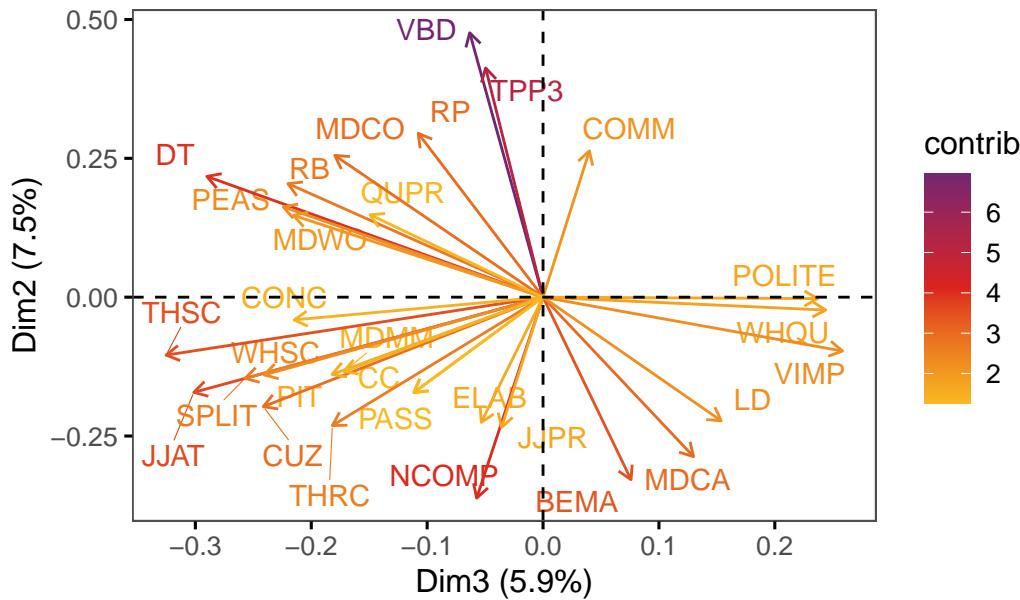
```
#ggsave(here("plots", "fviz_pca_var_PC1_PC2_Ref3Reg.svg"), width = 9, height
        = 7)

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()

```

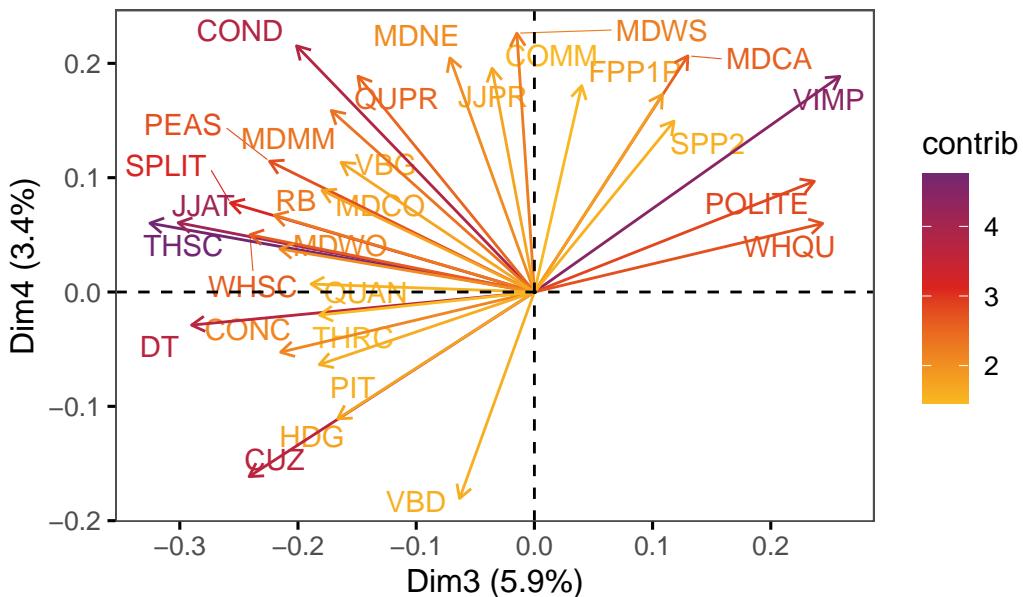


```

#ggsave(here("plots", "fviz_pca_var_PC3_PC2_Ref3Reg.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_Ref3Reg.svg"), width = 9, height
       = 8)
```

G.7 Exploring feature contributions in terms of normalised frequencies

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

```
ncounts <- readRDS(here("data", "processed", "ncounts3_3Reg.rds"))

ncounts |>
  filter(Register=="Informative") |>
  #filter(Level %in% c("C", "D", "E")) |>
  select(Level, VBD, PEAS) |>
  group_by(Level) |>
  summarise_if(is.numeric, mean) |>
  kable(digits=2)
```

Level	VBD	PEAS
A	28.11	0.21
B	34.11	2.49
C	35.21	5.36
D	39.83	7.63
E	35.17	6.91
Ref.	39.73	4.94

The following chunk produces Figure 35 which shows normalised counts of selected features with salient loadings on PC1 in the Textbook Informative subcorpus (Levels A to E) and the reference Info Teens corpus (Ref.). This plots visualises the observed normalised frequencies as they were extracted using the MFTE Perl (see Appendices C and F).

```

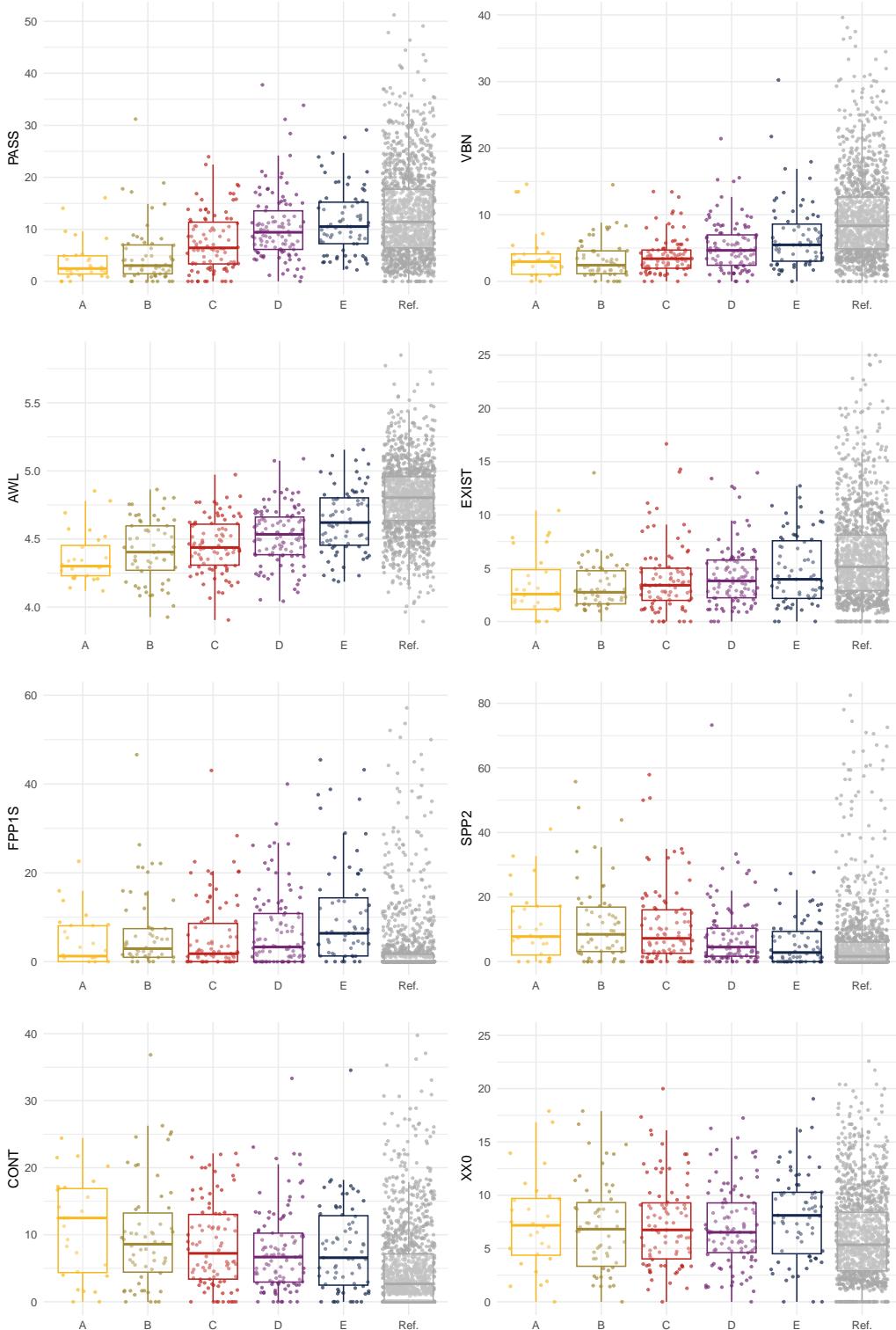
cols = c("#F9B921", "#A18A33", "#BD241E", "#722672", "#15274D", "darkgrey")

boxfeature <- ncounts |>
  filter(Register=="Informative") |>
  #filter(Level %in% c("C", "D", "E")) |>
  select(Level, FPP1S, SPP2, CONT, EXIST, AWL, XX0, PASS, VBN) |>
  ggplot(aes(x = Level, y = CONT, colour = Level, fill = Level)) +
  geom_jitter(size=0.7, alpha=.7) +
  geom_boxplot(outlier.shape = NA, fatten = 2, fill = "white", alpha = 0.3) +
  scale_colour_manual(values = cols) +
  theme_minimal() +
  theme(legend.position = "none") +
  xlab("")

CONT = boxfeature
SPP2 <- boxfeature + aes(y = SPP2)
EXIST <- boxfeature + aes(y = EXIST) + ylim(c(0,25)) # These y-axis limits
  ↳ remove individual outliers that overextend the scales and make the
  ↳ existing differences invisible to the naked eye. They can be removed to
  ↳ visualise all data points.
FFP1 <- boxfeature + aes(y = FPP1S) + ylim(c(0,60))
AWL <- boxfeature + aes(y = AWL)
XX0 <- boxfeature + aes(y = XX0) + ylim(c(0,25))
PASS <- boxfeature + aes(y = PASS)
VBN <- boxfeature + aes(y = VBN) + ylim(c(0,40))

boxplots <- gridExtra::grid.arrange(PASS, VBN, AWL, EXIST, FFP1, SPP2, CONT,
  ↳ XX0, ncol=2, nrow=4)

```



```
#ggsave(here("plots", "BoxplotsInformativeFeatures.svg"), plot = boxplots,
  ↴ dpi = 300, width = 9, height = 11)
```

G.8 Exploring the dimensions of the model

We begin with some descriptive statistics of the dimension scores.

```
#data <- readRDS(here("data", "processed", "dataforPCA.rds"))
#colnames(data)
pca <- prcomp(data[,9:ncol(data)], scale.=FALSE) # All quantitative variables

## Access to the PCA results
#colnames(data)
res.ind <- cbind(data[,1:8], as.data.frame(pca$x) [,1:4])

## Summary statistics
res.ind |>
  group_by(Subcorpus, Level) |>
  summarise_if(is.numeric, c(mean = mean, sd = sd)) |>
  kable(digits = 2)
```

Subcorpus	Level	Words	mPG1	mPG2	mPG3	mPG4	mWords	BC1	sHC2	sHC3	sHC4	sd
Textbook Conversa- tion	A	813.02	1.91	-1.02	3.49	-0.17	154.57	0.89	0.60	1.03	0.73	
Textbook Conversa- tion	B	819.48	2.11	-0.64	2.33	0.39	218.88	0.91	0.69	1.04	0.79	
Textbook Conversa- tion	C	823.69	1.59	-0.38	1.39	0.75	279.14	1.20	0.73	1.04	0.79	
Textbook Conversa- tion	D	797.85	1.09	-0.43	0.79	0.91	187.98	1.45	0.72	1.26	0.79	
Textbook Conversa- tion	E	1132.79	1.03	-0.61	0.86	1.09	569.85	1.30	0.78	0.88	0.62	
Textbook Fiction	A	886.00	0.13	0.22	2.74	-0.27	242.78	0.97	0.69	0.90	0.81	

Subcorpus	Level	Words	mPC1	mPC2	mPC3	mPC4	mWords	PC1	sPC2	sPC3	sPC4	sd
Textbook Fiction	B	864.16	-0.21	1.34	1.73	-0.29	222.44	0.79	0.62	0.73	0.53	
Textbook Fiction	C	854.21	-0.27	1.63	0.76	0.12	198.21	0.89	0.78	0.96	0.63	
Textbook Fiction	D	801.78	-0.84	1.38	0.26	0.15	196.33	1.15	0.79	0.83	0.59	
Textbook Fiction	E	853.26	-0.65	1.27	0.26	0.20	195.97	1.10	0.75	0.92	0.57	
Textbook Informative	A	851.71	-1.46	-0.96	1.86	-0.70	199.77	0.82	0.86	0.69	0.85	
Textbook Informative	B	844.03	-1.63	-0.50	1.23	-0.27	182.83	0.92	1.03	0.80	1.08	
Textbook Informative	C	838.90	-1.87	-0.54	0.25	0.14	160.21	1.09	0.98	0.93	1.06	
Textbook Informative	D	847.65	-2.24	-0.32	-0.15	0.13	179.25	0.95	0.92	0.95	0.83	
Textbook Informative	E	823.23	-2.57	-0.66	-0.28	0.37	180.39	0.99	0.90	0.79	0.82	
Spoken BNC2014 Ref.	Ref.	10637.74	3.71	-0.52	-0.64	-0.53	8974.14	0.49	0.47	0.76	0.52	
Youth Fiction Ref.	Ref.	5944.78	-0.49	1.75	-0.21	0.41	198.64	0.90	0.52	0.88	0.52	
Info Teens Ref.	Ref.	805.41	-3.06	-0.96	-0.23	-0.15	193.31	1.09	1.19	0.88	1.19	

```

res.ind <- res.ind |>
  mutate(Subsubcorpus = paste(Corpus, Register, sep = "_")) |>
  mutate(Subsubcorpus = as.factor(Subsubcorpus))

res.ind |>
  select(PC1, PC2, PC3, PC4, Subsubcorpus) |>
 tbl_summary(by = Subsubcorpus,
  digits = list(all_continuous() ~ c(2, 2)),
  statistic = all_continuous() ~ "{mean} ({sd})")

```

Characteristic	Informative, T	Spoken, ING, G	Written, C	English Fiction	French Fiction	Diction Month	Fictionat	Fiction, F
	N = 337	N = 1,250	N = 565	N = 285	N = 352	N = 1,191		
PC1	-3.06 (1.09)	3.71 (0.49)	1.56 (1.25)	-0.43 (1.05)	-2.05 (1.04)	-0.49 (0.90)		
PC2	-0.96 (1.19)	-0.52 (0.47)	-0.57 (0.74)	1.25 (0.84)	-0.52 (0.96)	1.75 (0.52)		
PC3	-0.23 (0.88)	-0.64 (0.76)	1.71 (1.43)	0.96 (1.22)	0.31 (1.10)	-0.21 (0.88)		
PC4	-0.15 (1.19)	-0.53 (0.52)	0.61 (0.86)	0.02 (0.64)	0.05 (0.98)	0.41 (0.52)		

```
res.ind |>
  select(Register, Level, PC4) |>
  group_by(Register, Level) |>
  summarise_if(is.numeric, c(Median = median, MAD = mad)) |>
  kable(digits = 2)
```

Register	Level	Median	MAD
Conversation	A	-0.07	0.68
Conversation	B	0.42	0.80
Conversation	C	0.73	0.80
Conversation	D	0.86	0.70
Conversation	E	1.09	0.61
Conversation	Ref.	-0.54	0.52
Fiction	A	-0.38	0.75
Fiction	B	-0.39	0.53
Fiction	C	0.13	0.67
Fiction	D	0.01	0.54
Fiction	E	0.12	0.67
Fiction	Ref.	0.40	0.50
Informative	A	-0.71	0.82
Informative	B	-0.37	1.10
Informative	C	0.07	1.09
Informative	D	0.04	0.79
Informative	E	0.37	0.57
Informative	Ref.	-0.08	1.21

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 that have high scores on Dim3 and low ones on Dim2 to proceed with a qualitative comparison and analysis of these texts.

```
# Search for example texts to illustrate results
res.ind |>
  filter(PC3 > 2 & PC2 < -2) |>
  #filter(Register=="Conversation") |>
  #filter(Level == "B") |>
  #filter(PC1 > 4.7) |>
  select(Filename, PC1, PC2, PC3) |>
  kable(digits=2)
```

Filename	PC1	PC2	PC3
NGL_1_Spoken_0002.txt	2.41	-2.27	4.36
HT_5_ELF_Spoken_0003.txt	1.94	-2.30	3.49
HT_6_Informative_0001.txt	-2.19	-2.36	3.11
Science_Tech_Kinds_NZ_10383721_typesofrobots.txt	-3.09	-2.62	2.24
History_Kids_BBC_10402894_go_further.txt	-4.26	-2.07	2.08

G.9 Raincloud plots visualising dimension scores

```
res.ind$Subcorpus <- fct_relevel(res.ind$Subcorpus, "Spoken BNC2014 Ref.",
  ↵ "Textbook Conversation", "Youth Fiction Ref.", "Textbook Fiction", "Info
  ↵ Teens Ref.", "Textbook Informative")

# colours <- suf_palette(name = "london", n = 6, type = "continuous")
# colours2 <- suf_palette(name = "classic", n = 5, type = "continuous")
# colours <- c(colours, colours2[c(2:4)]) # Nine colours range
# palette <- colours[c(1,5,6,2,3,8,7,4,9)] # Good order for PCA
# colours <- palette[c(1,8,9,2,7,3)]

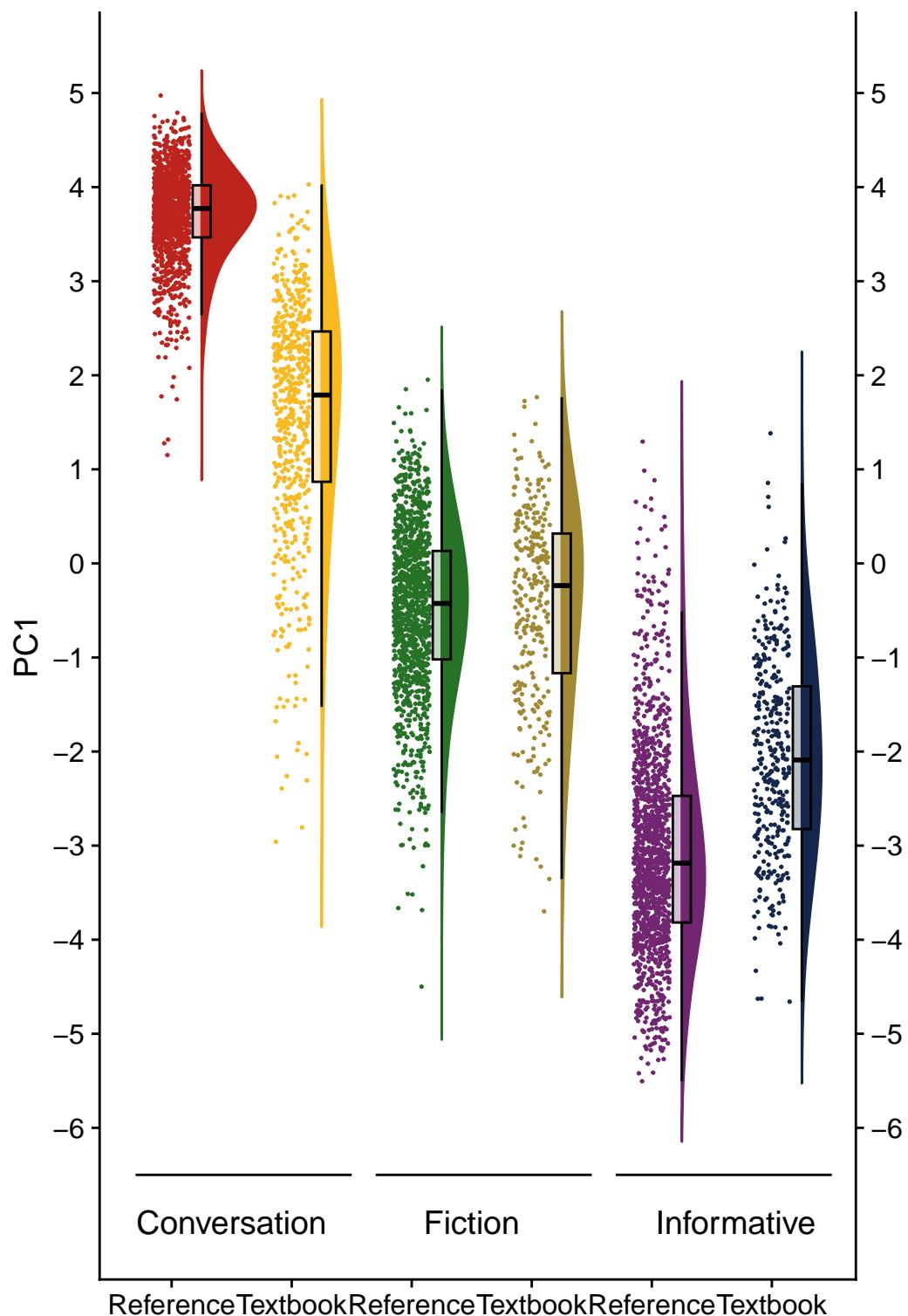
# This translates as:
palette <- c("#BD241E", "#A18A33", "#15274D", "#D54E1E", "#EA7E1E",
  ↵ "#4C4C4C", "#722672", "#F9B921", "#267226")
colours <- c("#BD241E", "#F9B921", "#267226", "#A18A33", "#722672", "#15274D")

ggplot(res.ind, aes(x=Subcorpus, y=PC1, fill = Subcorpus, colour =
  ↵ Subcorpus))+ # Or leave out "colour = Register" to keep the dots in black
  geom_flat_violin(position = position_nudge(x = .25, y = 0), adjust = 2, trim
  ↵ = FALSE) +
  geom_point(position = position_jitter(width = .15), size = .25) +
```

```

# note that here we need to set the x-variable to a numeric variable and bump
# it to get the boxplots to line up with the rainclouds.
geom_boxplot(aes(x = as.numeric(Subcorpus)+0.25, y = PC1), outlier.shape =
  NA, alpha = 0.3, width = .15, colour = "BLACK") +
ylab('PC1')+
theme_cowplot()+
theme(axis.title.x=element_blank())+
guides(fill = "none", colour = "none") +
scale_colour_manual(values = colours) +
scale_fill_manual(values = colours) +
annotate(geom = "text", x = 1.5, y = -7, label = "Conversation", size = 5)
  +
annotate(geom = "segment", x = 0.7, xend = 2.5, y = -6.5, yend = -6.5) +
annotate(geom = "text", x = 3.5, y = -7, label = "Fiction", size = 5) +
annotate(geom = "segment", x = 2.7, xend = 4.5, y = -6.5, yend = -6.5) +
annotate(geom = "text", x = 5.7, y = -7, label = "Informative", size = 5) +
annotate(geom = "segment", x = 4.7, xend = 6.5, y = -6.5, yend = -6.5) +
scale_x_discrete(labels=rep(c("Reference", "Textbook"), 3))+
scale_y_continuous(sec.axis = dup_axis(name=NULL), breaks = seq(from = -6,
  to = 5, by = 1))

```



```
#ggsave(here("plots", "PC1_3RegComparison.svg"), width = 13, height = 8)
#ggsave(here("plots", "PC1_3RegComparison.png"), width = 20, height = 15,
#  units = "cm", dpi = 300)
```

G.10 Computing mixed-effects models of the dimension scores

G.10.1 Data preparation

In this chunk, we add a `Source` variable to be used as a random effect variable in the following mixed-effects models (see 5.3.8 for details).

```
res.ind <- res.ind |>
  mutate(Source = case_when(
    Corpus=="Youth.Fiction" ~ paste("Book", str_extract(Filename,
      "[0-9]{1,3}"), sep = ""),
    Corpus=="Spoken.BNC2014" ~ "Spoken.BNC2014",
    Corpus=="Textbook.English" ~ as.character(Series),
    Corpus=="Informative.Teens" ~ str_extract(Filename, "BBC|Science_Tech"),
    TRUE ~ "NA")) |>
  mutate(Source = case_when(
    Corpus=="Informative.Teens" & is.na(Source) ~ str_remove(Filename, "_.*"),
    TRUE ~ as.character(Source))) |>
  mutate(Source = as.factor(Source)) |>
  mutate(Corpus = case_when(
    Corpus=="Textbook.English" ~ "Textbook",
    Corpus=="Informative.Teens" ~ "Reference",
    Corpus=="Spoken.BNC2014" ~ "Reference",
    Corpus=="Youth.Fiction" ~ "Reference"
  )) |>
  mutate(Corpus = as.factor(Corpus))

# Change the reference levels to theoretically more meaningful levels and one
# that is better populated (see, e.g.,
# https://stats.stackexchange.com/questions/430770/in-a-multilevel-linear-regression-how-do
# summary(res.ind$Corpus)
res.ind$Corpus <- relevel(res.ind$Corpus, "Reference")

# summary(res.ind$Subcorpus)
res.ind$Subcorpus <- factor(res.ind$Subcorpus, levels = c("Spoken BNC2014
# Ref.", "Textbook Conversation", "Youth Fiction Ref.", "Textbook Fiction",
# Info Teens Ref.", "Textbook Informative"))
```

```
# summary(res.ind$Level)
res.ind$Level <- relevel(res.ind$Level, "Ref.")
```

G.10.2 Dimension 1: ‘Spontaneous interactional vs. Edited informational’

We first compare various models and then present a tabular summary of the best-fitting one.

```
md_source <- lmer(PC1 ~ 1 + (Register|Source), res.ind, REML = FALSE)
md_corpus <- update(md_source, .~. + Level) # Failed to converge
md_register <- update(md_source, . ~ . + Register)
md_both <- update(md_corpus, .~. + Register)
md_interaction <- update(md_both, . ~ . + Level:Register)

anova(md_source, md_corpus, md_register, md_both, md_interaction)
```

```
Data: res.ind
Models:
  md_source: PC1 ~ 1 + (Register | Source)
  md_register: PC1 ~ (Register | Source) + Register
  md_corpus: PC1 ~ (Register | Source) + Level
  md_both: PC1 ~ (Register | Source) + Level + Register
  md_interaction: PC1 ~ (Register | Source) + Level + Register + Level:Register
      npar   AIC   BIC logLik deviance   Chisq Df Pr(>Chisq)
  md_source     8 12421 12473 -6202.5    12405
  md_register   10 12357 12422 -6168.4    12337  68.106  2  1.625e-15 ***
  md_corpus     13 12181 12266 -6077.5    12155 181.996  3 < 2.2e-16 ***
  md_both       15 12117 12215 -6043.5    12087  67.843  2  1.854e-15 ***
  md_interaction 25 12098 12261 -6024.0    12048 39.104 10  2.435e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
md_interaction <- lmer(PC1 ~ Level + Register + Level*Register +
  ~ (Register|Source), res.ind, REML = FALSE)

tab_model(md_interaction, wrap.labels = 200) # R2 = 0.870 / 0.923
```

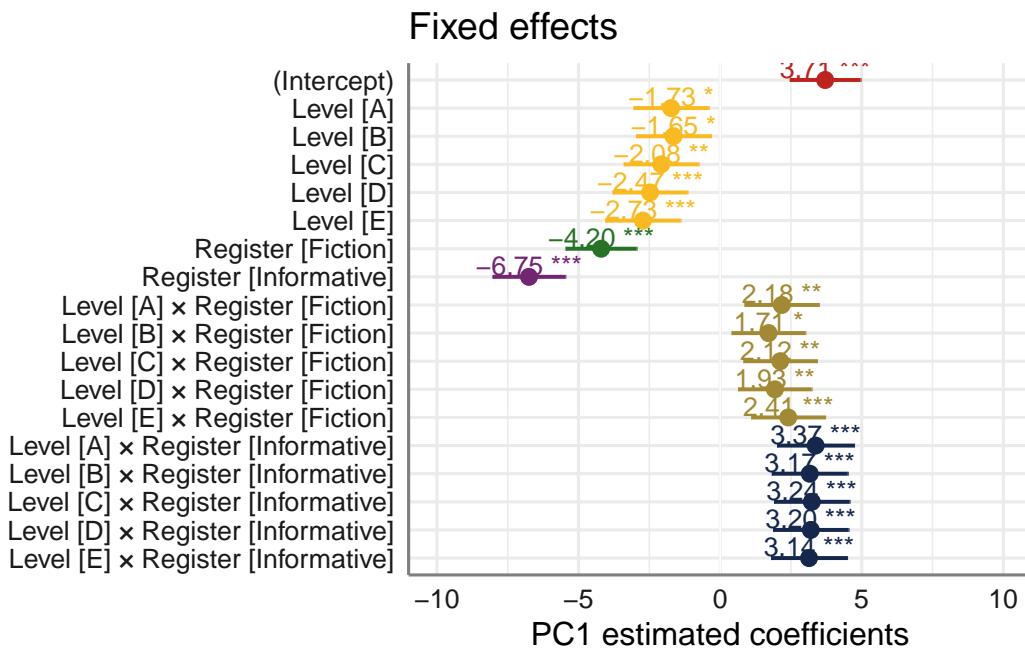
Its estimated coefficients are visualised in the plot below.

```

# Tweak plot aesthetics with:
#   ↳ https://cran.r-project.org/web/packages/sjPlot/vignettes/custplot.html
# Colour customisation trick from:
#   ↳ https://stackoverflow.com/questions/55598920/different-line-colors-in-forest-plot-output

plot_model(md_interaction,
            #type = "re", # Option to visualise random effects
            show.intercept = TRUE,
            show.values=TRUE,
            show.p=TRUE,
            value.offset = .4,
            value.size = 3.5,
            colors = palette[c(1:3,7:9)],
            group.terms = c(1,5,5,5,5,6,4,2,2,2,2,2,3,3,3,3,3),
            title="Fixed effects",
            wrap.labels = 40,
            axis.title = "PC1 estimated coefficients") +
            theme_sjplot2()

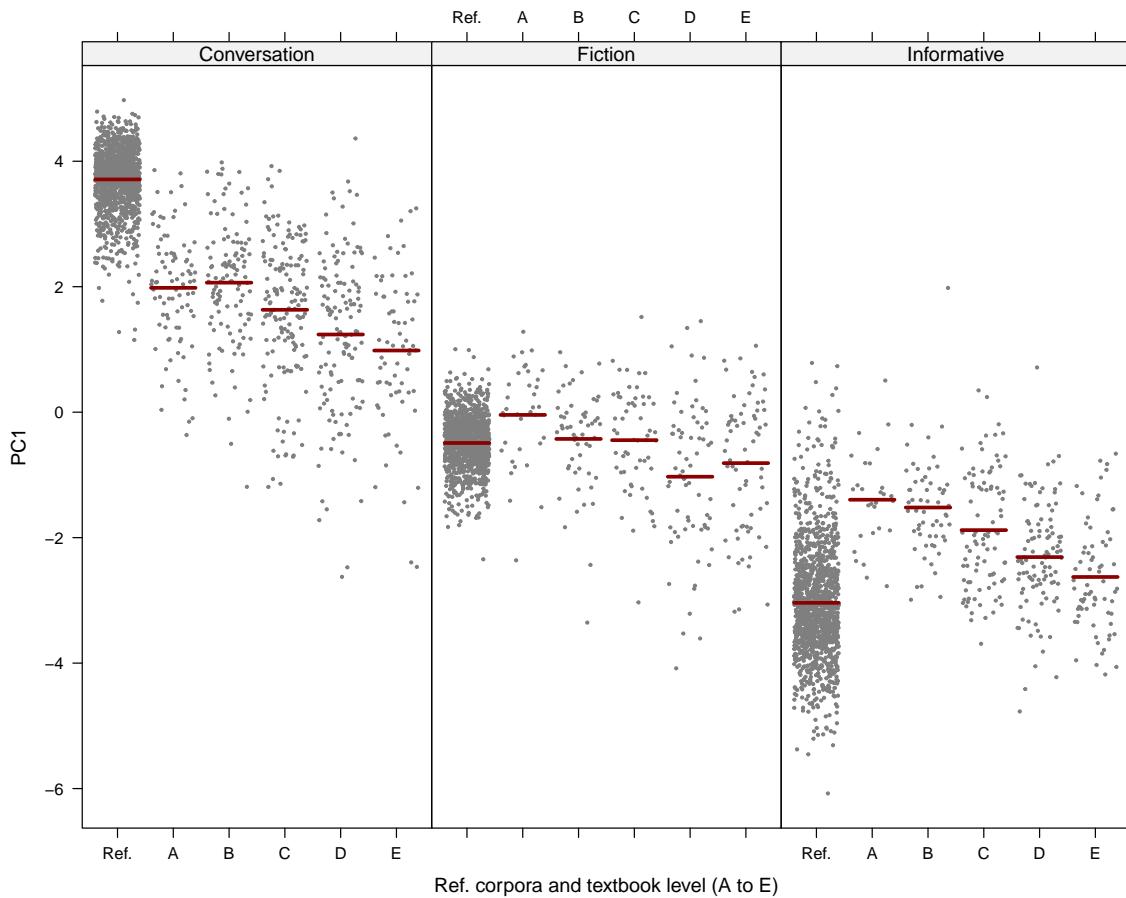
```



```
#ggsave(here("plots", "TxBRef3Reg_PC1_lmer_fixed.svg"), height = 6, width =
#   ↳ 9)
```

The `visreg` function is used to visualise the distributions of the modelled Dim1 scores:

```
# svg(here("plots", "TxBRReg3Reg_predicted_PC1_scores_interactions.svg"),
  ↵ height = 8, width = 9)
visreg(md_interaction, xvar = "Level", by="Register",
       #type = "contrast",
       type = "conditional",
       line=list(col="darkred"),
       points=list(cex=0.3),
       xlab = "Ref. corpora and textbook level (A to E)", ylab = "PC1",
       layout=c(3,1)
)
```



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

For PC2 to PC4, the models with random intercepts and slopes failed to converge, which is why only slopes are included in the following models.

```
# Function to avoid repeating model fitting and comparison process for each
# PC.
run_anova <- function(response_var, data) {
  # Fit the initial model
  md_source <- lmer(formula = paste(response_var, "~ 1 + (1|Source)", data =
  data, REML = FALSE)

  # Update models
  md_corpus <- update(md_source, . ~ . + Level)
  md_register <- update(md_source, . ~ . + Register)
  md_both <- update(md_corpus, . ~ . + Register)
  md_interaction <- update(md_both, . ~ . + Level:Register)

  # Perform ANOVA
  anova_results <- anova(md_source, md_corpus, md_register, md_both,
  md_interaction)

  # Print ANOVA results
  print(anova_results)

  # Save model object with appropriate name
  pc_number <- gsub("PC", "", response_var)
  assign(paste("md_interaction_PC", pc_number, sep = ""), md_interaction,
  envir = .GlobalEnv)

  # Return tabulated model
  return(md_interaction)
}
```

G.10.3 Dimension 2: ‘Narrative vs. Non-narrative’

We first compare various models and then present a tabular summary of the best-fitting one.

```
PC2_models <- run_anova("PC2", res.ind)
```

```

Data: data
Models:
md_source: PC2 ~ 1 + (1 | Source)
md_register: PC2 ~ (1 | Source) + Register
md_corpus: PC2 ~ (1 | Source) + Level
md_both: PC2 ~ (1 | Source) + Level + Register
md_interaction: PC2 ~ (1 | Source) + Level + Register + Level:Register
npar   AIC    BIC  logLik deviance   Chisq Df Pr(>Chisq)
md_source      3 12281 12301 -6137.6     12275
md_register     5 10761 10794 -5375.7     10751 1523.882  2 < 2.2e-16 ***
md_corpus       8 12138 12190 -6061.1     12122  0.000  3        1
md_both         10 10616 10681 -5298.1     10596 1525.963  2 < 2.2e-16 ***
md_interaction  20 10550 10680 -5254.9     10510  86.436 10  2.718e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

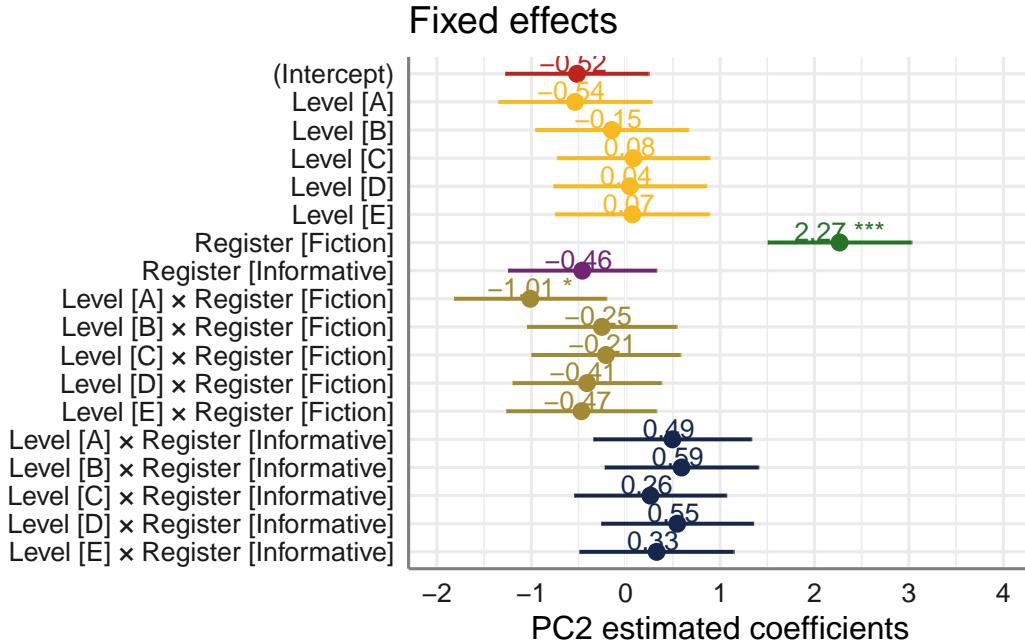
```
tab_model(md_interaction_PC2) # R2 = 0.671 / 0.753
```

Visualisation of the coefficient estimates of the fixed effects:

```

plot_model(md_interaction_PC2,
            #type = "re", # Option to visualise random effects
            show.intercept = TRUE,
            show.values=TRUE,
            show.p=TRUE,
            value.offset = .4,
            value.size = 3.5,
            colors = palette[c(1:3,7:9)],
            group.terms = c(1,5,5,5,5,6,4,2,2,2,2,2,3,3,3,3,3),
            title="Fixed effects",
            wrap.labels = 40,
            axis.title = "PC2 estimated coefficients") +
            theme_sjplot2()

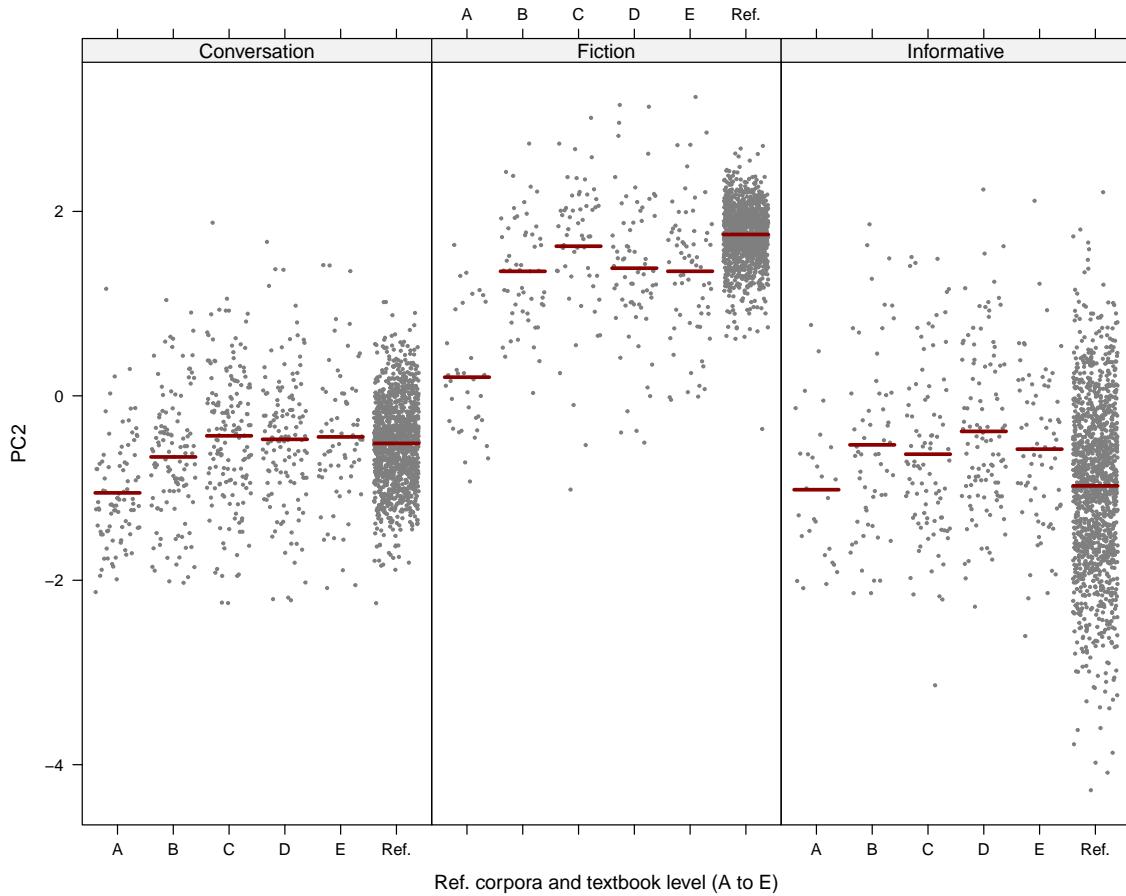
```



```
#ggsave(here("plots", "TxBRef3Reg_PC2_lmer_fixed.svg"), height = 6, width =
  ↵ 9)
```

Visualisation of the predicted Dim2 scores:

```
# svg(here("plots", "TxBReg3Reg_predicted_PC2_scores_interactions.svg"),
  ↵ height = 8, width = 9)
visreg(md_interaction_PC2, xvar = "Level", by="Register",
  #type = "contrast",
  type = "conditional",
  line=list(col="darkred"),
  points=list(cex=0.3),
  xlab = "Ref. corpora and textbook level (A to E)", ylab = "PC2",
  layout=c(3,1)
)}
```

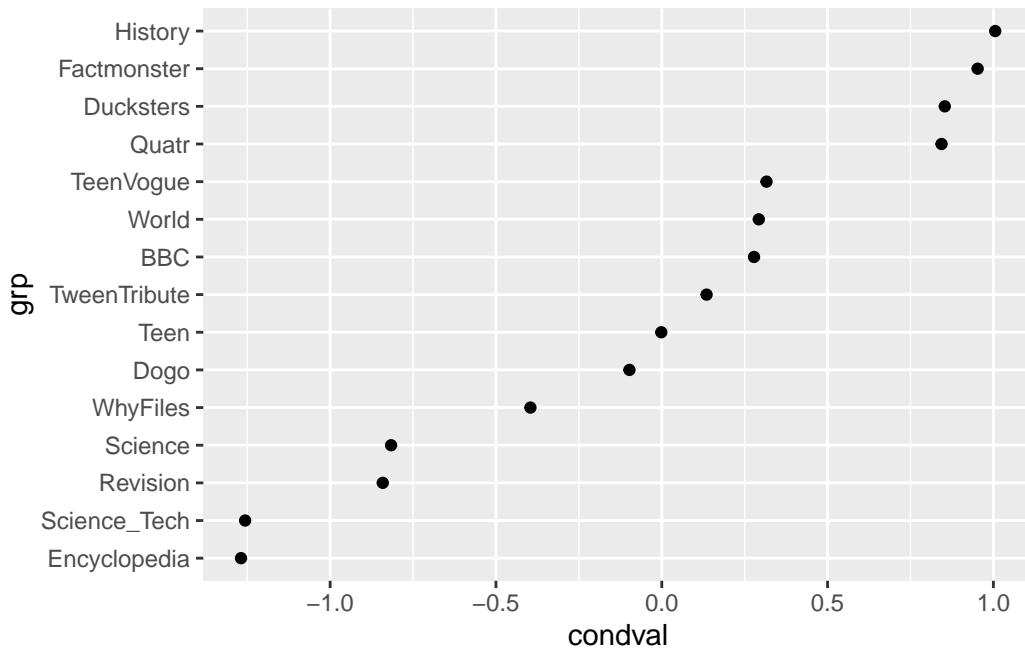


```
# dev.off()
```

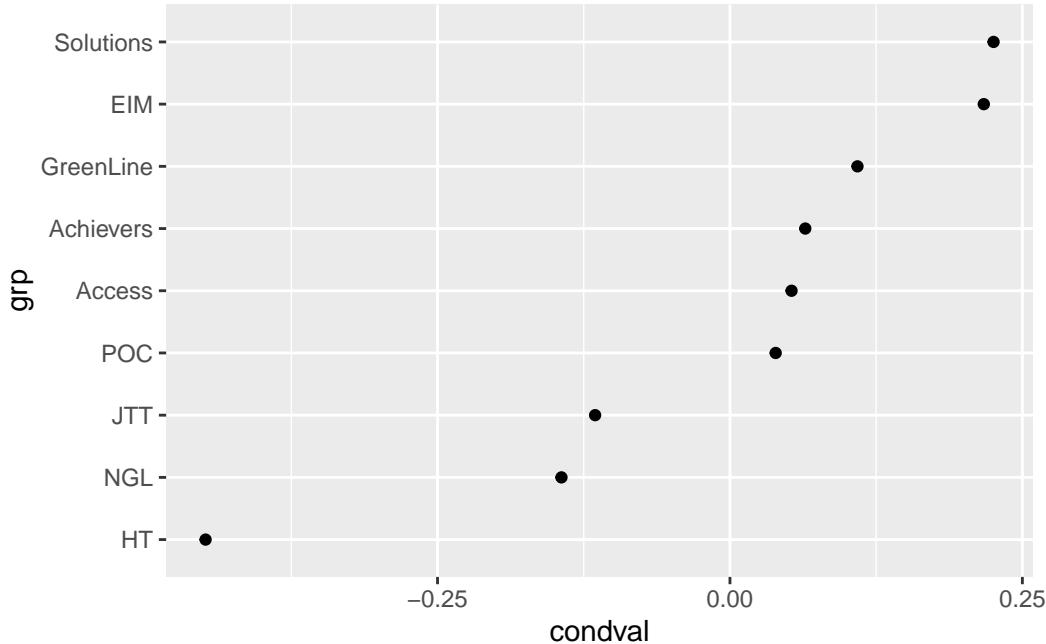
We can also explore the random effect structure.

```
# Random effects
ranef <- as.data.frame(ranef(md_interaction_PC2))

# Exploring the random effects of the sources of the Info Teens corpus
ranef |>
  filter(grp %in% c("TeenVogue", "BBC", "Dogo", "Ducksters", "Encyclopedia",
    "Factmonster", "History", "Quatr", "Revision", "Science",
    "Science_Tech", "Teen", "TweenTribute", "WhyFiles", "World")) |>
  ggplot(aes(x = grp, y = condval)) +
  geom_point() +
  coord_flip()
```



```
# Exploring the random effects associated with textbook series
ranef |>
  filter(grp %in% levels(data$Series)) |>
  ggplot(aes(x = grp, y = condval)) +
  geom_point() +
  coord_flip()
```



G.10.4 Dimension 3: ‘Pedagogically adapted vs. Natural’

We first compare various models and then present a tabular summary of the best-fitting one.

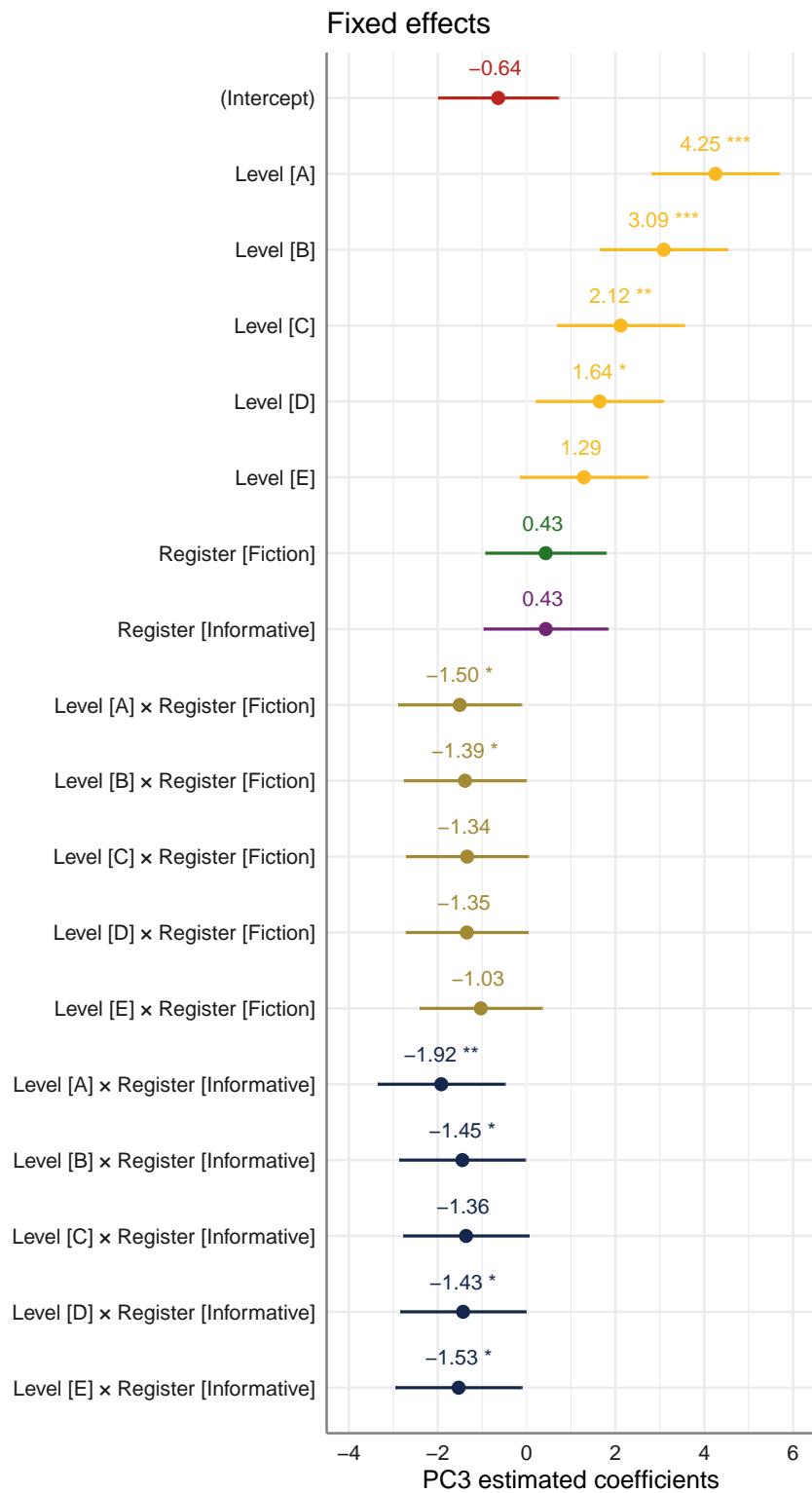
```
PC3_models <- run_anova("PC3", res.ind)
```

```
Data: data
Models:
md_source: PC3 ~ 1 + (1 | Source)
md_register: PC3 ~ (1 | Source) + Register
md_corpus: PC3 ~ (1 | Source) + Level
md_both: PC3 ~ (1 | Source) + Level + Register
md_interaction: PC3 ~ (1 | Source) + Level + Register + Level:Register
      npar   AIC   BIC logLik deviance    Chisq Df Pr(>Chisq)
md_source       3 13523 13542 -6758.3     13517
md_register     5 12988 13020 -6489.0     12978  538.750  2 < 2.2e-16 ***
md_corpus       8 11928 11981 -5956.2     11912 1065.455  3 < 2.2e-16 ***
md_both        10 11466 11531 -5722.8     11446  466.870  2 < 2.2e-16 ***
md_interaction  20 11461 11592 -5710.7     11421  24.264 10  0.006929 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
tab_model(md_interaction_PC3) # R2 = 0.425 / 0.700
```

Visualisation of the coefficient estimates of the fixed effects:

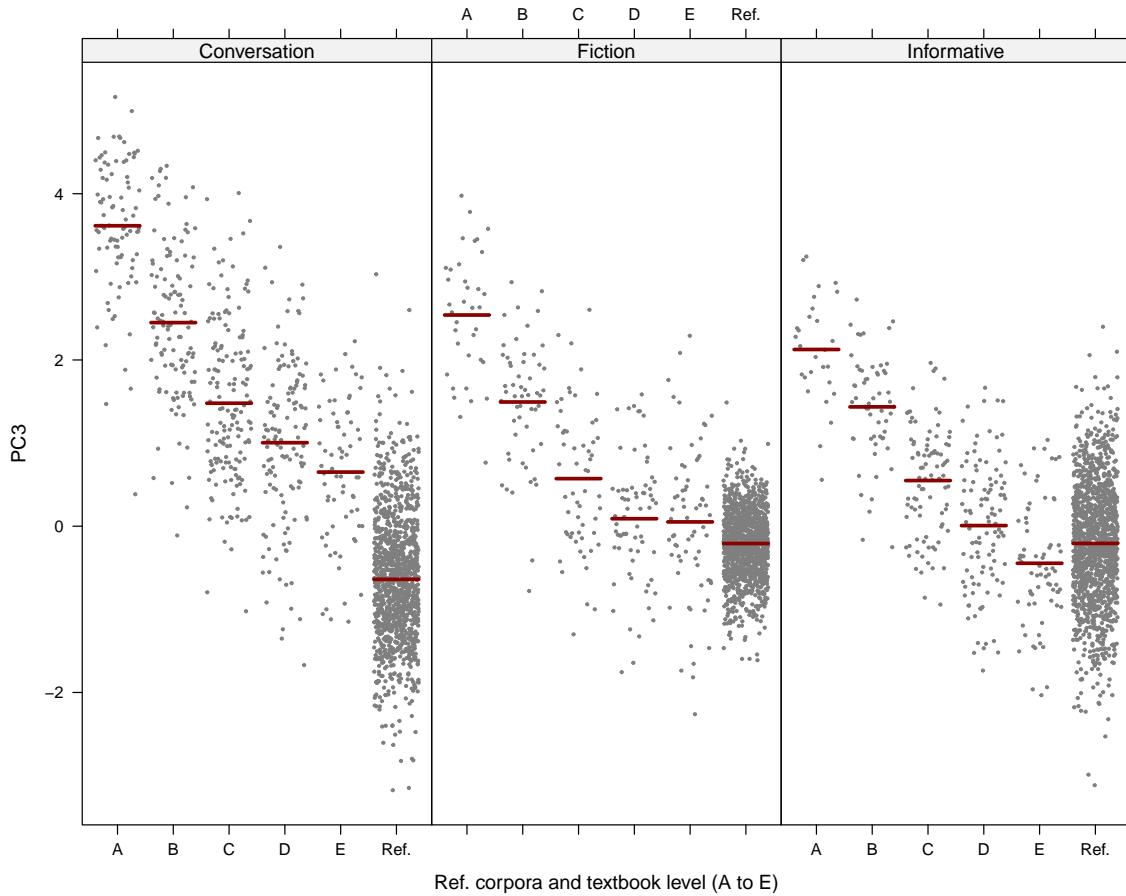
```
plot_model(md_interaction_PC3,
            #type = "re", # Option to visualise random effects
            show.intercept = TRUE,
            show.values=TRUE,
            show.p=TRUE,
            value.offset = .4,
            value.size = 3.5,
            colors = palette[c(1:3,7:9)],
            group.terms = c(1,5,5,5,5,5,6,4,2,2,2,2,2,3,3,3,3,3),
            title="Fixed effects",
            wrap.labels = 40,
            axis.title = "PC3 estimated coefficients") +
theme_sjplot2()
```



```
#ggsave(here("plots", "TxBRef3Reg_PC3_lmer_fixed.svg"), height = 6, width =
  ↵ 9)
```

Visualisation of the predicted Dim3 scores:

```
# svg(here("plots", "TxBReg3Reg_predicted_PC3_scores_interactions.svg"),
  ↵ height = 8, width = 9)
visreg(md_interaction_PC3, xvar = "Level", by="Register",
       #type = "contrast",
       type = "conditional",
       line=list(col="darkred"),
       points=list(cex=0.3),
       xlab = "Ref. corpora and textbook level (A to E)", ylab = "PC3",
       layout=c(3,1)
)
```

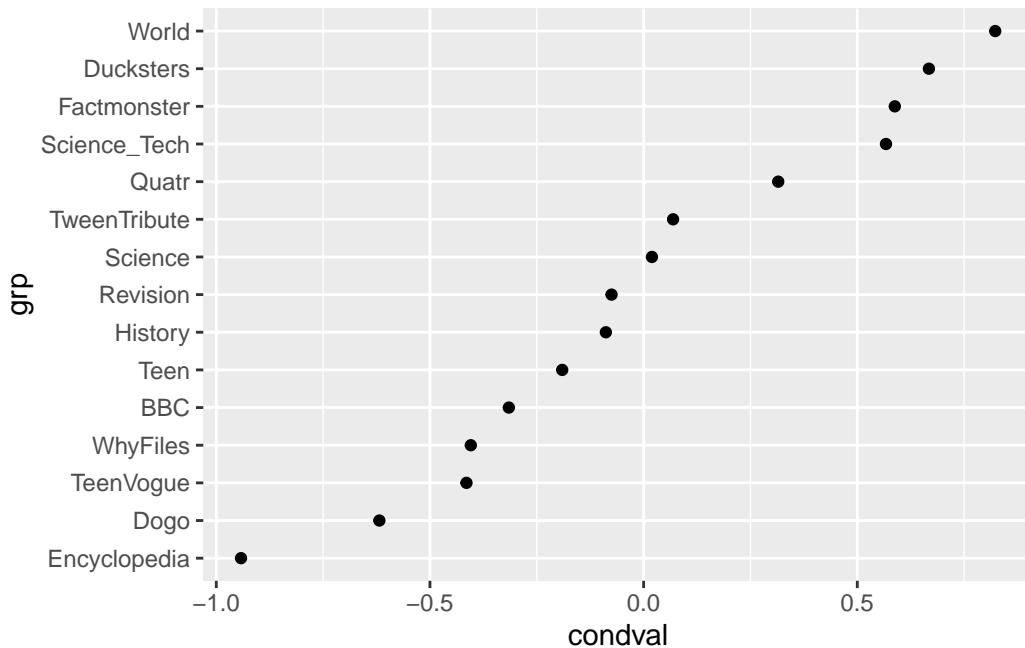


```
# dev.off()
```

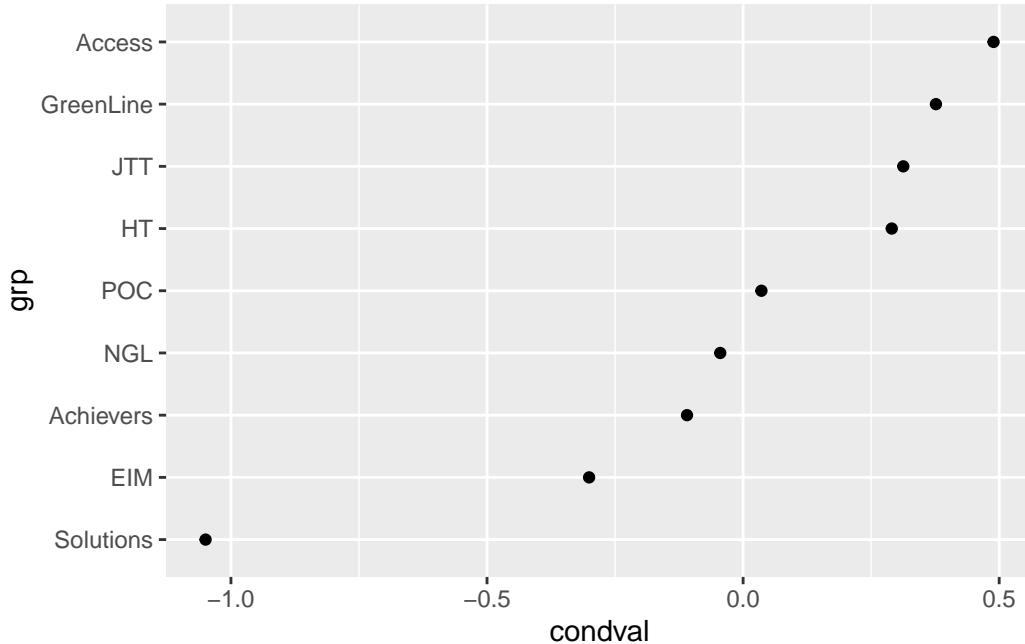
We can also explore the random effect structure.

```
# Random effects
ranef <- as.data.frame(ranef(md_interaction_PC3))

# Exploring the random effects of the sources of the Info Teens corpus
ranef |>
  filter(grp %in% c("TeenVogue", "BBC", "Dogo", "Ducksters", "Encyclopedia",
    "Factmonster", "History", "Quatr", "Revision", "Science",
    "Science_Tech", "Teen", "TweenTribute", "WhyFiles", "World")) |>
  ggplot(aes(x = grp, y = condval)) +
  geom_point() +
  coord_flip()
```



```
# Exploring the random effects associated with textbook series
ranef |>
  filter(grp %in% levels(data$Series)) |>
  ggplot(aes(x = grp, y = condval)) +
  geom_point() +
  coord_flip()
```



G.10.5 Dimension 4: ‘Factual vs. Speculative’ / ‘Simple vs. complex verb forms’?

We first compare various models and then present a tabular summary of the best-fitting one.

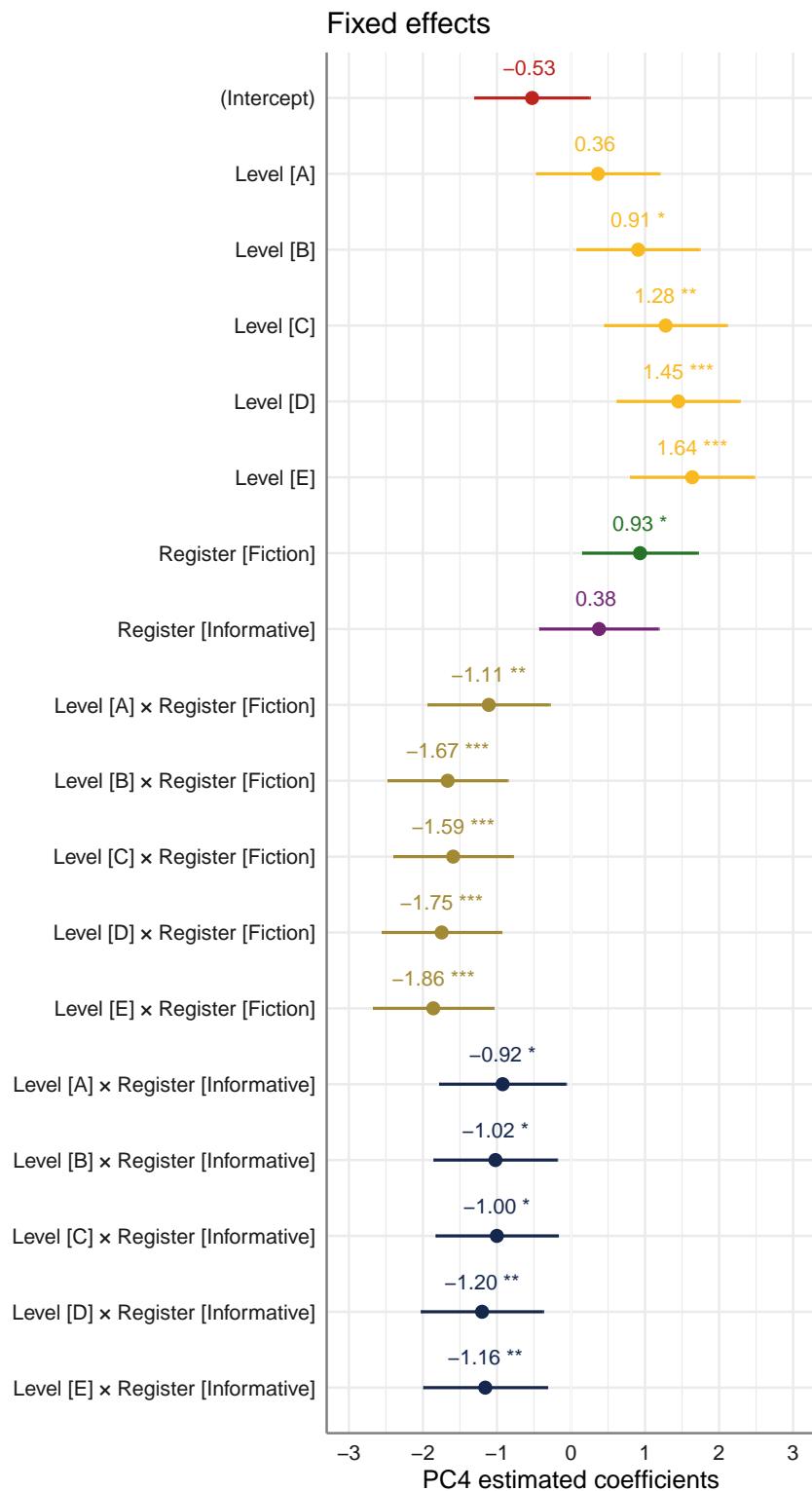
```
PC4_models <- run_anova("PC4", res.ind)
```

```
Data: data
Models:
md_source: PC4 ~ 1 + (1 | Source)
md_register: PC4 ~ (1 | Source) + Register
md_corpus: PC4 ~ (1 | Source) + Level
md_both: PC4 ~ (1 | Source) + Level + Register
md_interaction: PC4 ~ (1 | Source) + Level + Register + Level:Register
      npar   AIC   BIC logLik deviance    Chisq Df Pr(>Chisq)
md_source       3 11019 11039 -5506.5     11013
md_register     5 10825 10857 -5407.4     10815 198.2593  2    < 2e-16 ***
md_corpus       8 10827 10879 -5405.3     10811  4.0956  3    0.2513
md_both         10 10563 10628 -5271.6     10543 267.5805  2    < 2e-16 ***
md_interaction  20 10527 10657 -5243.3     10487 56.4370 10    1.7e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
tab_model(md_interaction_PC4) # R2 = 0.234 / 0.434
```

Visualisation of the coefficient estimates of the fixed effects:

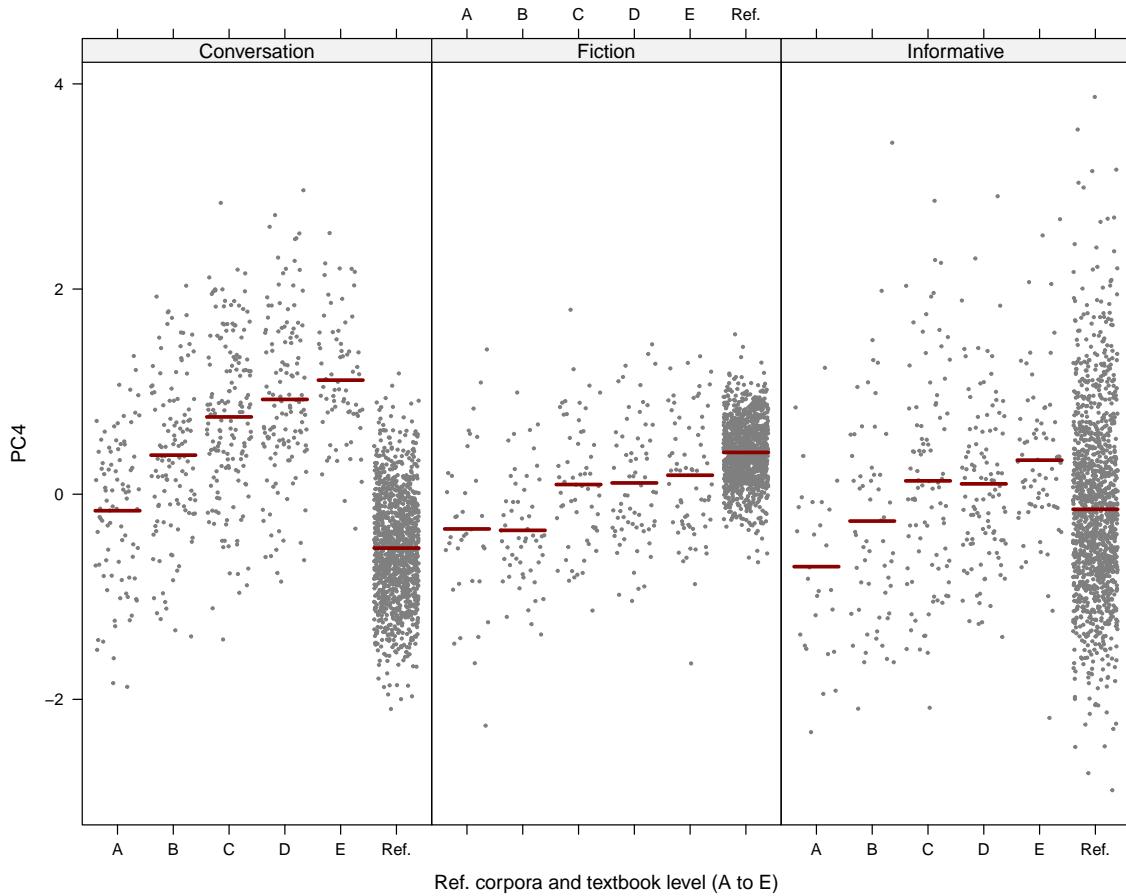
```
plot_model(md_interaction_PC4,
            #type = "re", # Option to visualise random effects
            show.intercept = TRUE,
            show.values=TRUE,
            show.p=TRUE,
            value.offset = .4,
            value.size = 3.5,
            colors = palette[c(1:3,7:9)],
            group.terms = c(1,5,5,5,5,5,6,4,2,2,2,2,2,3,3,3,3,3),
            title="Fixed effects",
            wrap.labels = 40,
            axis.title = "PC4 estimated coefficients") +
theme_sjplot2()
```



```
#ggsave(here("plots", "TxBRef3Reg_PC4_lmer_fixed.svg"), height = 6, width =
  ↵ 9)
```

Visualisation of the predicted Dim4 scores:

```
# svg(here("plots", "TxBReg3Reg_predicted_PC4_scores_interactions.svg"),
  ↵ height = 8, width = 9)
visreg(md_interaction_PC4, xvar = "Level", by="Register",
       #type = "contrast",
       type = "conditional",
       line=list(col="darkred"),
       points=list(cex=0.3),
       xlab = "Ref. corpora and textbook level (A to E)", ylab = "PC4",
       layout=c(3,1)
)
```

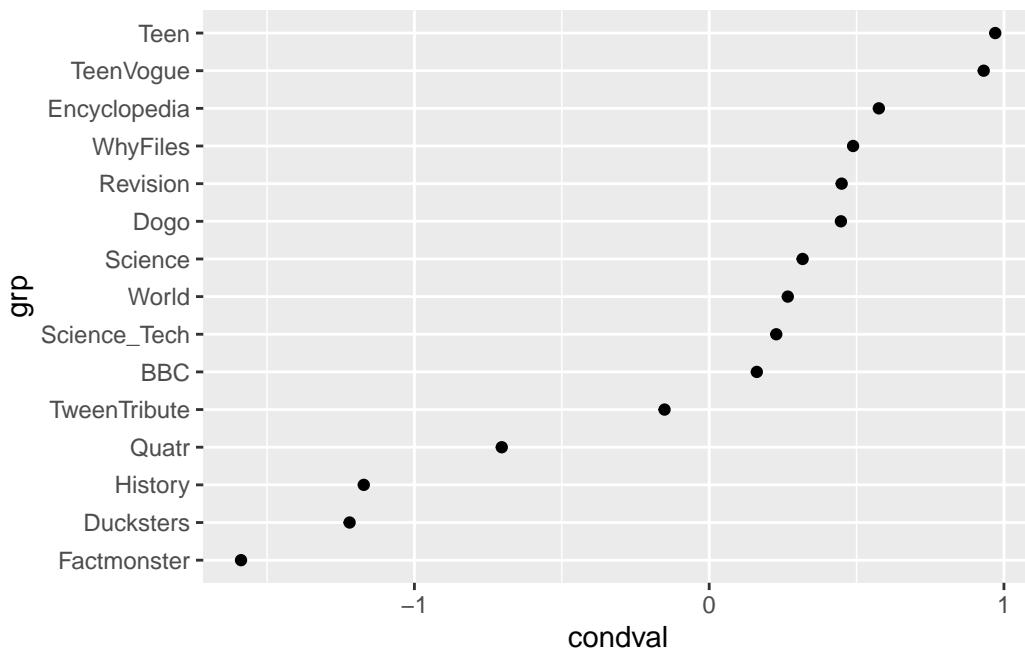


```
# dev.off()
```

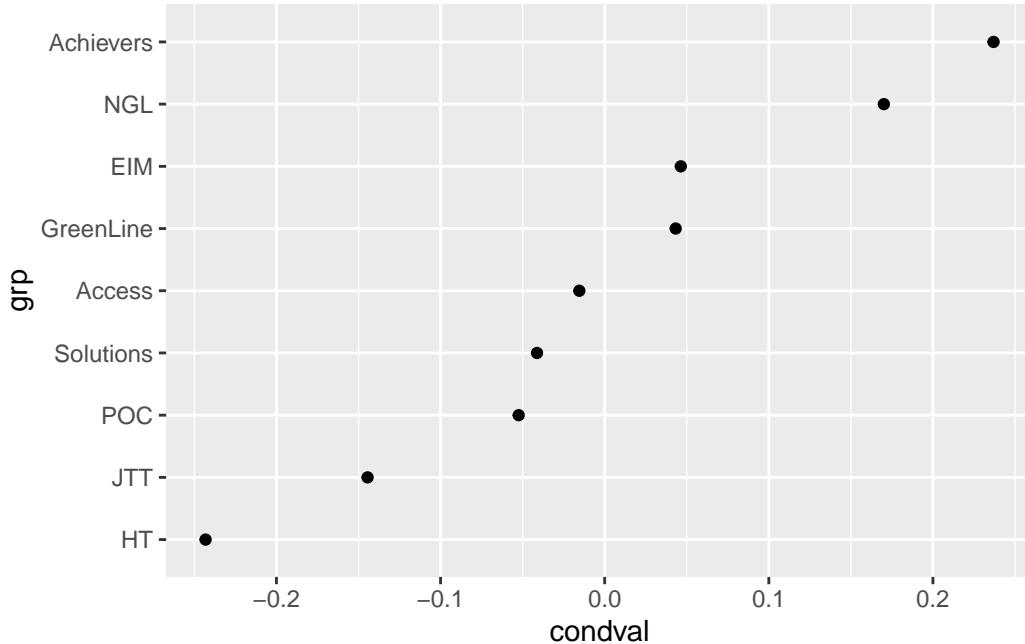
We can also explore the random effect structure.

```
# Random effects
ranef <- as.data.frame(ranef(md_interaction_PC4))

# Exploring the random effects of the sources of the Info Teens corpus
ranef |>
  filter(grp %in% c("TeenVogue", "BBC", "Dogo", "Ducksters", "Encyclopedia",
    "Factmonster", "History", "Quatr", "Revision", "Science",
    "Science_Tech", "Teen", "TweenTribute", "WhyFiles", "World")) |>
  ggplot(aes(x = grp, y = condval)) +
  geom_point() +
  coord_flip()
```



```
# Exploring the random effects associated with textbook series
ranef |>
  filter(grp %in% levels(data$Series)) |>
  ggplot(aes(x = grp, y = condval)) +
  geom_point() +
  coord_flip()
```

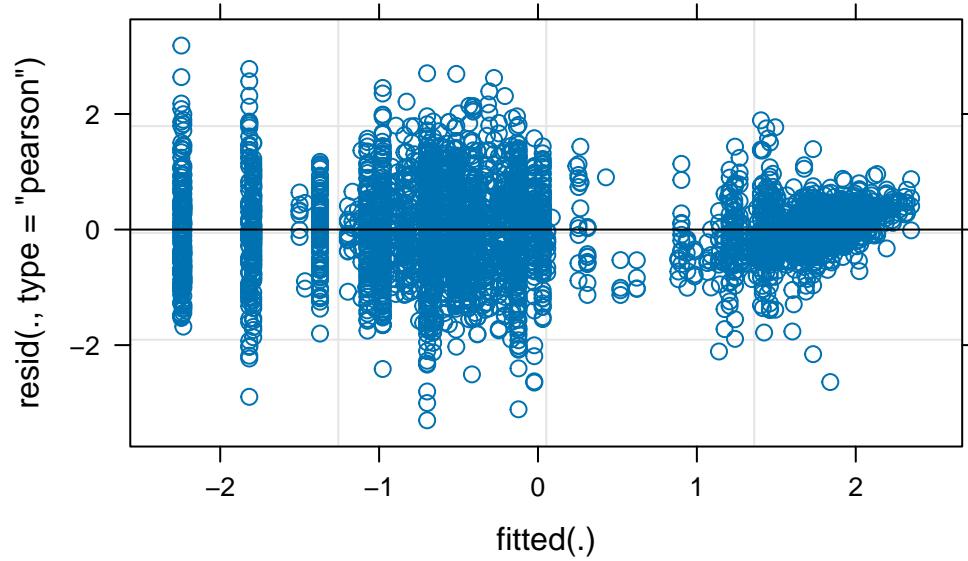


G.10.6 Testing model assumptions

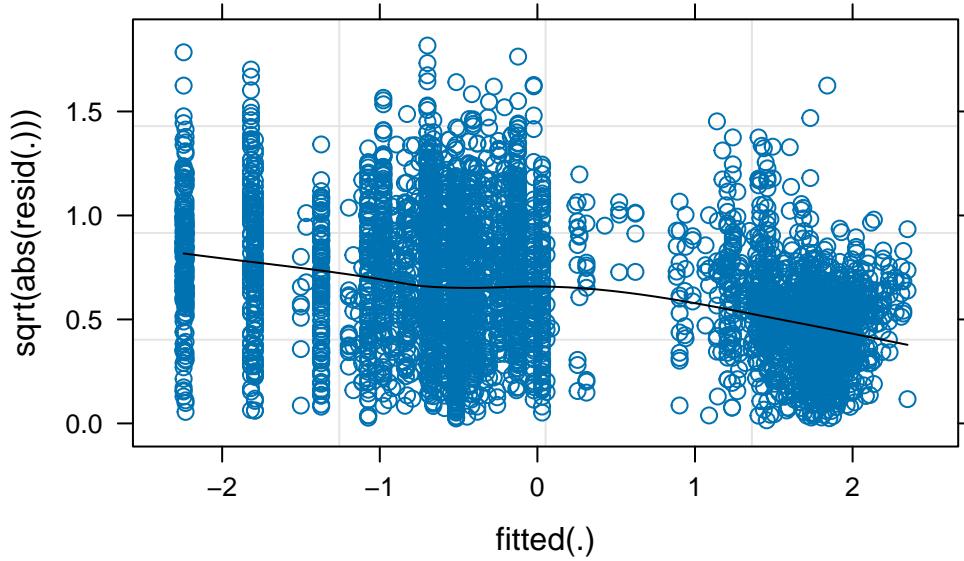
This chunk can be used to check the assumptions of all of the models computed above. In the following example, we examine the final model selected to predict Dim2 scores.

```
model2test <- md_interaction_PC2

# 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)
```

