# Textbook English: A Multi-Dimensional Approach

**Online Supplements**

Elen Le Foll

2024-03-01

# Table of contents

# Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

# 1 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

# 2 Data import from MFTE output

This document outlines the steps taken to pre-process the Textbook English Corpus (TEC) data.

```
#renv::restore() # Restore the project's dependencies from the lockfile to ensure that same p

library(caret) # For its confusion matrix function
library(here) # For dynamic file paths
library(patchwork) # For Fig. 1
library(PerformanceAnalytics)
library(psych) # For various useful stats function
library(tidyverse)
```

```
# Read in Textbook Corpus data
# This .tsv file corresponds to the "mixed normalised frequency" output of the MFTE Perl v. 3
TxBcounts <- read.delim(here("MFTE_data", "Outputs", "TxB900MDA_3.1_normed_complex_counts.tsv
TxBcounts <- TxBcounts %>% filter(Filename!=".DS_Store") %>% droplevels(.)
str(TxBcounts) # Check sanity of data
```

```
'data.frame':   2014 obs. of  84 variables:
 $ Filename: Factor w/ 2014 levels "Access_1_Informative_0001.txt",..: 333 2008 1644 389 447
 $ Words   : int  931 889 750 979 690 694 547 967 927 840 ...
 $ AWL     : num  4.57 4.48 3.9 3.99 4.7 ...
 $ TTR     : num  0.4 0.435 0.505 0.453 0.59 ...
 $ LD      : num  0.594 0.533 0.516 0.545 0.59 ...
 $ DT      : num  33 41.9 39.4 21.1 28.2 ...
 $ JJAT    : num  7.93 8.73 18.9 7.37 25.74 ...
 $ POS     : num  2.2 0 0 0 0.99 ...
 $ NCOMP   : num  7.93 3.93 4.72 16.84 12.87 ...
 $ QUAN    : num  2.2 5.68 11.02 6.84 2.97 ...
 $ ACT     : num  22.4 36.2 23.9 30.6 43.8 ...
 $ ASPECT  : num  3.731 8.511 2.817 0.901 6.25 ...
 $ CAUSE   : num  0 1.06 1.41 2.7 2.08 ...
 $ COMM    : num  29.85 24.47 9.86 8.11 16.67 ...
 $ CUZ     : num  0 0 1.409 0.901 0 ...
```

```
$ CC      : num  22.4 30.9 45.1 27.9 68.8 ...
$ CONC    : num  0 0 0 0 4.17 ...
$ COND    : num  1.49 0 1.41 1.8 0 ...
$ EX      : num  0.746 0 0 2.703 0 ...
$ EXIST   : num  0.746 2.128 8.451 1.802 12.5 ...
$ ELAB    : num  0 0 0 0 0 ...
$ FREQ    : num  3.731 3.192 2.817 0.901 2.083 ...
$ JJPR    : num  8.21 8.51 12.68 13.51 12.5 ...
$ MENTAL  : num  25.4 21.3 36.6 28.8 20.8 ...
$ OCCUR   : num  0.746 3.192 7.042 0 0 ...
$ DOAUX   : num  8.96 7.45 1.41 17.12 0 ...
$ QUTAG   : num  0 0 0 0 0 0 0 0 0 0 ...
$ QUPR    : num  0.746 0 5.634 0.901 4.167 ...
$ SPLIT   : num  0 0 1.409 0.901 4.167 ...
$ STPR    : num  0.746 3.192 1.409 2.703 0 ...
$ WHQU    : num  18.66 13.83 8.45 11.71 0 ...
$ THSC    : num  1.492 2.128 0 0.901 2.083 ...
$ WHSC    : num  5.22 6.38 5.63 3.6 16.67 ...
$ CONT    : num  3.73 0 19.72 30.63 4.17 ...
$ VBD     : num  1.49 9.57 38.03 3.6 18.75 ...
$ VPRT    : num  35.1 31.9 26.8 75.7 54.2 ...
$ PLACE   : num  0.746 2.128 0 4.505 6.25 ...
$ PROG    : num  5.22 3.19 2.82 7.21 6.25 ...
$ HGOT    : num  0 0 0 1.8 0 ...
$ BEMA    : num  8.96 8.51 19.72 17.12 14.58 ...
$ MDCA    : num  1.49 3.19 1.41 3.6 4.17 ...
$ MDCO    : num  0 0 1.41 0 0 ...
$ TIME    : num  2.99 2.13 5.63 4.5 2.08 ...
$ THATD   : num  5.22 0 0 3.6 0 ...
$ THRC    : num  0 0 0 0 0 0 0 0 0 0 ...
$ VIMP    : num  59.7 53.19 2.82 6.31 6.25 ...
$ MDMM    : num  0 0 0 0 0 0 0 0 0 0 ...
$ ABLE    : num  0 0 0 0 0 0 0 0 0 0 ...
$ MDNE    : num  0 0 4.23 6.31 6.25 ...
$ MDWS    : num  1.49 0 2.82 1.8 10.42 ...
$ MDWO    : num  0.746 2.128 22.535 2.703 0 ...
$ XXO     : num  2.24 4.26 7.04 9.01 4.17 ...
$ PASS    : num  0.746 2.128 2.817 1.802 4.167 ...
$ PGET    : num  0 0 0 0 0 0 0 0 0 0 ...
$ VBG     : num  0.746 10.638 7.042 10.811 14.583 ...
$ VBN     : num  0 3.19 0 0 4.17 ...
$ PEAS    : num  0 0 0 0.901 4.167 ...
$ GTO     : num  0.746 0 2.817 0 0 ...
```

```
 $ FPP1S   : num  0.746 1.064 60.563 45.946 0 ...
 $ FPP1P   : num  0 0 18.3 22.5 20.8 ...
 $ TPP3S   : num  3.731 7.447 1.409 0.901 0 ...
 $ TPP3P   : num  5.97 4.26 0 1.8 12.5 ...
 $ SPP2    : num  23.9 29.8 23.9 29.7 33.3 ...
 $ PIT     : num  1.49 0 7.04 11.71 12.5 ...
 $ PRP     : num  0 0 0 0 0 ...
 $ RP      : num  0 5.319 4.225 0.901 6.25 ...
 $ AMP     : num  0 0.113 0.267 0.102 0.145 ...
 $ CD      : num  0.43 1.012 2.4 0.511 1.304 ...
 $ DEMO    : num  0.537 0.562 0.267 0.306 0.145 ...
 $ DMA     : num  0.107 0 2 0.204 0 ...
 $ DWNT    : num  0 0 0.133 0.204 0 ...
 $ EMO     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ EMPH    : num  0 0.113 0.667 0.715 0.29 ...
 $ FPUH    : num  0 0.113 0.8 0.102 0 ...
 $ HDG     : num  0.107 0 0.667 0 0.145 ...
 $ HST     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ IN      : num  10.31 14.4 10.13 7.97 10.87 ...
 $ LIKE    : num  0.215 0.225 0.4 0.204 0 ...
 $ NN      : num  24.4 25.8 16.9 19.4 29.3 ...
 $ POLITE  : num  0 0 0.4 0.306 0 ...
 $ RB      : num  1.074 0.338 1.467 1.941 0.725 ...
 $ SO      : num  0 0 0.267 0.409 0.29 ...
 $ URL     : num  0 0 0 0 0 ...
 $ YNQU    : num  0.43 1.012 0.667 0.511 0 ...
```

```
nrow(TxBcounts) # Should be 2014 files
```

```
[1] 2014
```

```
# Adding a textbook proficiency level
TxBLevels <- read.delim(here("metadata", "TxB900MDA_ProficiencyLevels.csv"), sep = ",")
TxBcounts <- full_join(TxBcounts, TxBLevels, by = "Filename") %>%
  mutate(Level = as.factor(Level)) %>%
  mutate(Filename = as.factor(Filename))
summary(TxBcounts$Level) # Check distribution and that there are no NAs
```

```
  A   B   C   D   E
292 407 506 478 331
```

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

```
                                    Filename Level
1  Solutions_Intermediate_Plus_Spoken_0018.txt     D
2                   Access_3_Personal_0001.txt     C
3                   JTT_4_Informative_0007.txt     C
4            English_in_Mind_1_Poetry_0001.txt     B
5     English_In_Mind_2_Instructional_0002.txt     C
6                        HT_3_Personal_0001.txt     D
7         New_GreenLine_5_Informative_0012.txt     E
8              Access_4_Instructional_0001.txt     D
9                     Access_5_Spoken_0003.txt     E
10          GreenLine_4_Instructional_0009.txt     D
11      New_GreenLine_4_Instructional_0007.txt     D
12                    HT_3_Narrative_0002.txt     D
13             New_GreenLine_3_Spoken_0020.txt     C
14           Achievers_B2_Informative_0015.txt     E
15 Solutions_Intermediate_Informative_0024.txt     C
16             New_GreenLine_4_Spoken_0015.txt     D
17           English_in_Mind_1_Spoken_0015.txt     B
18             Access_5_Instructional_0003.txt     E
19                          HT_3_Spoken_0007.txt     D
20           English_in_Mind_3_Spoken_0003.txt     C
```

```
# Adding a register variable from the file names
TxBcounts$Register <- as.factor(stringr::str_extract(TxBcounts$Filename, "Spoken|Narrative|O
summary(TxBcounts$Register)
```

```
  Informative Instructional     Narrative      Personal        Poetry
          364           647           285            88            37
       Spoken
          593
```

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

```
 [1] "Filename" "Words"    "AWL"      "TTR"      "LD"       "DT"
 [7] "JJAT"     "POS"      "NCOMP"    "QUAN"     "ACT"      "ASPECT"
[13] "CAUSE"    "COMM"     "CUZ"      "CC"       "CONC"     "COND"
[19] "EX"       "EXIST"    "ELAB"     "FREQ"     "JJPR"     "MENTAL"
```

```
[25] "OCCUR"    "DOAUX"    "QUTAG"    "QUPR"     "SPLIT"    "STPR"
[31] "WHQU"     "THSC"     "WHSC"     "CONT"     "VBD"      "VPRT"
[37] "PLACE"    "PROG"     "HGOT"     "BEMA"     "MDCA"     "MDCO"
[43] "TIME"     "THATD"    "THRC"     "VIMP"     "MDMM"     "ABLE"
[49] "MDNE"     "MDWS"     "MDWO"     "XXO"      "PASS"     "PGET"
[55] "VBG"      "VBN"      "PEAS"     "GTO"      "FPP1S"    "FPP1P"
[61] "TPP3S"    "TPP3P"    "SPP2"     "PIT"      "PRP"      "RP"
[67] "AMP"      "CD"       "DEMO"     "DMA"      "DWNT"     "EMO"
[73] "EMPH"     "FPUH"     "HDG"      "HST"      "IN"       "LIKE"
[79] "NN"       "POLITE"   "RB"       "SO"       "URL"      "YNQU"
[85] "Level"    "Register"
```

```
# Adding a textbook series variable from the file names
TxBcounts$Filename <- stringr::str_replace(TxBcounts$Filename, "English_In_Mind|English_in_Mi
TxBcounts$Filename <- stringr::str_replace(TxBcounts$Filename, "New_GreenLine", "NGL") # Othe
TxBcounts$Filename <- stringr::str_replace(TxBcounts$Filename, "Piece_of_cake", "POC") # Shor
TxBcounts$Series <- as.factor(stringr::str_extract(TxBcounts$Filename, "Access|Achievers|EIM
summary(TxBcounts$Series) # Extract textbook series from (ammended) filenames
```

```
   Access Achievers       EIM GreenLine        HT       JTT        NB       NGL
      315       240       180       209       115       129        44       298
       NM       POC Solutions
       59        98       327
```

```
# Including the French textbooks for the first year of Lycée to their corresponding publishe
TxBcounts$Series <-car::recode(TxBcounts$Series, "c('NB', 'JTT') = 'JTT'; c('NM', 'HT') = 'HT
summary(TxBcounts$Series)
```

```
   Access Achievers       EIM GreenLine        HT       JTT       NGL       POC
      315       240       180       209       174       173       298        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') = 'Germa
summary(TxBcounts$Country)
```

```
 France Germany   Spain
    445     822     747
```

```
# Re-order variables
colnames(TxBcounts)
```

```
 [1] "Filename" "Words"    "AWL"      "TTR"      "LD"       "DT"
 [7] "JJAT"     "POS"      "NCOMP"    "QUAN"     "ACT"      "ASPECT"
[13] "CAUSE"    "COMM"     "CUZ"      "CC"       "CONC"     "COND"
[19] "EX"       "EXIST"    "ELAB"     "FREQ"     "JJPR"     "MENTAL"
[25] "OCCUR"    "DOAUX"    "QUTAG"    "QUPR"     "SPLIT"    "STPR"
[31] "WHQU"     "THSC"     "WHSC"     "CONT"     "VBD"      "VPRT"
[37] "PLACE"    "PROG"     "HGOT"     "BEMA"     "MDCA"     "MDCO"
[43] "TIME"     "THATD"    "THRC"     "VIMP"     "MDMM"     "ABLE"
[49] "MDNE"     "MDWS"     "MDWO"     "XXO"      "PASS"     "PGET"
[55] "VBG"      "VBN"      "PEAS"     "GTO"      "FPP1S"    "FPP1P"
[61] "TPP3S"    "TPP3P"    "SPP2"     "PIT"      "PRP"      "RP"
[67] "AMP"      "CD"       "DEMO"     "DMA"      "DWNT"     "EMO"
[73] "EMPH"     "FPUH"     "HDG"      "HST"      "IN"       "LIKE"
[79] "NN"       "POLITE"   "RB"       "SO"       "URL"      "YNQU"
[85] "Level"    "Register" "Series"   "Country"
```

```
TxBcounts <- TxBcounts %>%
  select(order(names(.))) %>% # Order alphabetically first
  select(Filename, Country, Series, Level, Register, Words, everything())
```

## 2.1 Summary statistics

```
TxBcounts %>%
  group_by(Register) %>%
  summarise(totaltexts = n(), totalwords = sum(Words), mean = as.integer(mean(Words)), sd = a
```

```
# A tibble: 6 x 6
  Register      totaltexts totalwords  mean    sd TTRmean
  <fct>              <int>      <int> <int> <int>   <dbl>
1 Conversation         593     505147   851   301   0.438
2 Fiction              285     241512   847   208   0.472
3 Informative          364     304695   837   177   0.514
4 Instructional        647     585049   904    94   0.421
5 Personal              88      69570   790   177   0.477
6 Poetry                37      26445   714   192   0.436
```

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

# 3 Data preparation for PCA

## 3.1 Removal of Poetry texts

```
nrow(TxBcounts)
```

```
[1] 2014
```

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

nrow(TxBcounts)
```

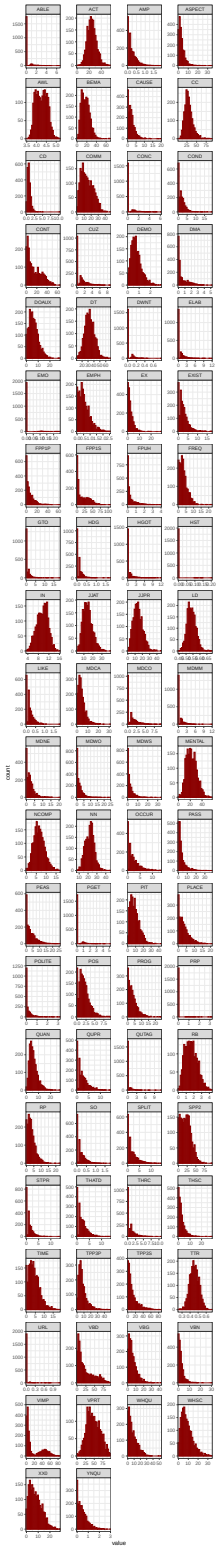
```
[1] 1977
```

```
summary(TxBcounts$Register)
```

```
 Conversation       Fiction   Informative Instructional      Personal
          593           285           364           647            88
```

## 3.2 Feature distributions

```
TxBcounts %>%
  select(-Words) %>%
  keep(is.numeric) %>%
  tidyr::gather() %>% # This function from tidyr converts a selection of variables into two v
  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)
```


ABLE
ACT
AMP
ASPECT
AWL
BEMA
CAUSE
CC
CD
COMM
CONC
COND
CONT
CUZ
DEMO
DWA
DOMJX
DT
DWNT
ELAB
EMO
EMPH
EX
EXIST
FPPSP
FPPSS
FPUH
FREQ
GTO
HDO
HGOT
HST
IN
JJR
JJPR
LO
LIKE
MDCA
MDCO
MDMM
MDNE
MDWO
MDWS
MENTAL
NCOMP
NN
OCCUR
PASS
PEAS
PGET
PIT
PLACE
POLITE
POS
PROG
PRP
QUAN
QUPR
QUTAG
RB
RP
SO
SPLIT
SPP2
STPR
THATD
THEC
THSC
TIME
TPPSP
TPP2S
TTR
URL
VBD
VBG
VBN
VIMP
VPRT
WHQU
WHSC
XXC
YNQU
count
value

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

### 3.2.1 Feature removal I

```
# Removal of meaningless features:
# CD because numbers as digits were mostly removed from the textbooks
# LIKE and SO because they are "bin" features designed to ensure that the counts for these tw
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
  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)
print(zero_features)
```

```
   rowname Percentage
1      GTO      67.07
2     ELAB      69.30
3     MDMM      70.81
4     HGOT      73.75
5     CONC      80.48
6     DWNT      81.44
7    QUTAG      85.99
8     PGET      87.35
```

```
9     ABLE      88.87
10     URL      96.51
11     EMO      97.82
12     PRP      98.33
13     HST      99.44
```

```
# 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_features <- compute_percentage(TxBcounts, TxBcounts == 0, 66.6)
print(zero_features)
```

```
   rowname Percentage
1      GTO      67.07
2     ELAB      69.30
3     MDMM      70.81
4     HGOT      73.75
5     CONC      80.48
6     DWNT      81.44
7    QUTAG      85.99
8      URL      96.51
9      EMO      97.82
10     PRP      98.33
11     HST      99.44
```

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

```
[1] 64
```

```
colnames(TxBcounts)
```

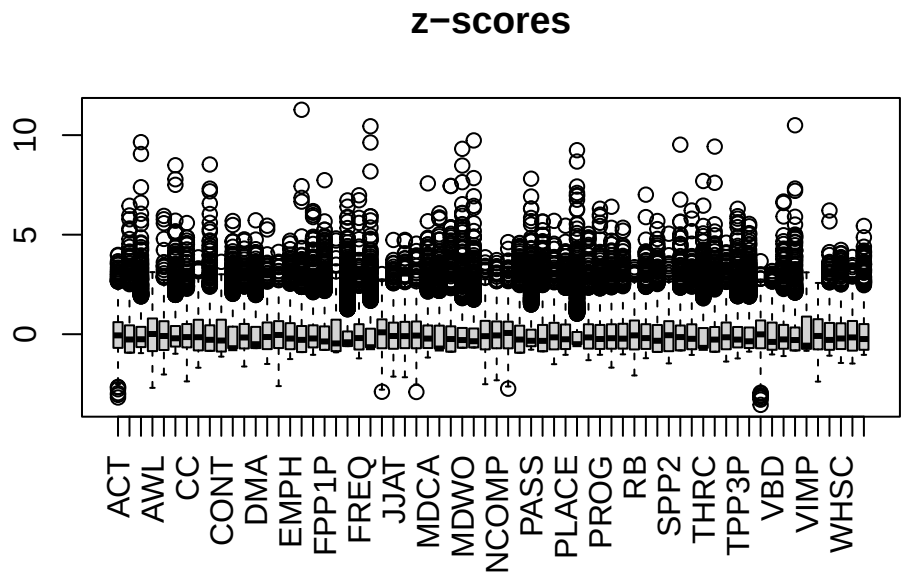
```
 [1] "Filename" "Country"  "Series"   "Level"    "Register" "Words"
 [7] "ACT"      "AMP"      "ASPECT"   "AWL"      "BEMA"     "CAUSE"
[13] "CC"       "COMM"     "COND"     "CONT"     "CUZ"      "DEMO"
[19] "DMA"      "DOAUX"    "DT"       "EMPH"     "EX"       "EXIST"
```

```
[25] "FPP1P"    "FPP1S"    "FPUH"     "FREQ"     "HDG"      "IN"
[31] "JJAT"     "JJPR"     "LD"       "MDCA"     "MDCO"     "MDNE"
[37] "MDWO"     "MDWS"     "MENTAL"   "NCOMP"    "NN"       "OCCUR"
[43] "PASS"     "PEAS"     "PIT"      "PLACE"    "POLITE"   "POS"
[49] "PROG"     "QUAN"     "QUPR"     "RB"       "RP"       "SPLIT"
[55] "SPP2"     "STPR"     "THATD"    "THRC"     "THSC"     "TIME"
[61] "TPP3P"    "TPP3S"    "TTR"      "VBD"      "VBG"      "VBN"
[67] "VIMP"     "VPRT"     "WHQU"     "WHSC"     "XXO"      "YNQU"
```

### 3.2.2 Standardising normalised counts and identifying potential outliers

"As an alternative to removing very sparse feature, we apply a signed logarithmic transformation to deskew the feature distributions." (Neumann & Evert)

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

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))
nrow(TxBdata)
```

```
[1] 1977
```

```
str(TxBdata)
```

```
'data.frame':   1977 obs. of  72 variables:
 $ Filename: chr  "Achievers_A1_Instructional_0012.txt" "Solutions_Pre-Intermediate_Instructi
 $ Country : Factor w/ 3 levels "France","Germany",..: 3 3 1 3 3 1 2 3 2 2 ...
 $ Series  : Factor w/ 9 levels "Access","Achievers",..: 2 9 8 2 2 8 1 2 7 1 ...
 $ Level   : Factor w/ 5 levels "A","B","C","D",..: 1 2 3 2 4 2 4 4 1 1 ...
 $ Register: Factor w/ 5 levels "Conversation",..: 4 4 1 5 3 1 2 4 1 2 ...
 $ Words   : int  931 889 750 979 690 694 547 967 927 840 ...
 $ ACT     : num  -0.2569 1.5417 -0.0539 0.8188 2.531 ...
 $ AMP     : num  -0.929 -0.493 0.104 -0.533 -0.368 ...
 $ ASPECT  : num  0.2073 1.689 -0.0762 -0.6702 0.9881 ...
 $ AWL     : num  1.08 0.776 -1.218 -0.89 1.541 ...
 $ BEMA    : num  -0.984 -1.0355 0.2643 -0.0374 -0.3312 ...
 $ CAUSE   : num  -0.98019 -0.47668 -0.31353 0.29904 0.00587 ...
 $ CC      : num  -0.647 0.133 1.444 -0.136 3.627 ...
 $ COMM    : num  1.824 1.191 -0.525 -0.731 0.275 ...
 $ COND    : num  -0.1074 -0.8371 -0.1484 0.0438 -0.8371 ...
 $ CONT    : num  -0.847 -1.124 0.339 1.149 -0.815 ...
 $ CUZ     : num  -0.7024 -0.7024 0.3261 -0.0446 -0.7024 ...
 $ DEMO    : num  -0.448 -0.393 -1.039 -0.952 -1.305 ...
 $ DMA     : num  -0.564 -0.697 1.795 -0.443 -0.697 ...
 $ DOAUX   : num  0.706 0.335 -1.148 2.711 -1.495 ...
 $ DT      : num  0.0457 1.1124 0.806 -1.3939 -0.5334 ...
 $ EMPH    : num  -1.25 -0.99 0.294 0.406 -0.579 ...
 $ EX      : num  -0.579 -0.897 -0.897 0.254 -0.897 ...
 $ EXIST   : num  -0.726 -0.152 2.473 -0.287 4.154 ...
 $ FPP1P   : num  -0.788 -0.788 1.734 2.314 2.081 ...
 $ FPP1S   : num  -0.869 -0.852 2.336 1.553 -0.909 ...
 $ FPUH    : num  -0.597 -0.391 0.869 -0.41 -0.597 ...
 $ FREQ    : num  0.1757 -0.0252 -0.1645 -0.8775 -0.4375 ...
 $ HDG     : num  0.0361 -0.6537 3.6281 -0.6537 0.2769 ...
 $ IN      : num  0.366 2.282 0.283 -0.733 0.628 ...
 $ JJAT    : num  -1.192 -1.042 0.853 -1.296 2.129 ...
```

```
$ JJPR    : num  -0.922 -0.876 -0.247 -0.12 -0.273 ...
$ LD      : num  1.713 -0.179 -0.714 0.203 1.585 ...
$ MDCA    : num  -0.751251 -0.274434 -0.774825 -0.15878 -0.000748 ...
$ MDCO    : num  -0.719 -0.719 0.302 -0.719 -0.719 ...
$ MDNE    : num  -0.88 -0.88 0.847 1.697 1.674 ...
$ MDWO    : num  -0.38 0.181 8.472 0.415 -0.683 ...
$ MDWS    : num  -0.201 -0.667 0.212 -0.105 2.583 ...
$ MENTAL  : num  0.356 -0.107 1.626 0.746 -0.157 ...
$ NCOMP   : num  0.528 -0.906 -0.621 3.722 2.299 ...
$ NN      : num  0.47 0.741 -0.996 -0.509 1.432 ...
$ OCCUR   : num  -0.627 0.371 1.943 -0.931 -0.931 ...
$ PASS    : num  -0.607 -0.293 -0.136 -0.367 0.171 ...
$ PEAS    : num  -0.858 -0.858 -0.858 -0.627 0.212 ...
$ PIT     : num  -1.253 -1.505 -0.318 0.468 0.601 ...
$ PLACE   : num  -0.802 -0.371 -1.035 0.371 0.916 ...
$ POLITE  : num  -0.5 -0.5 0.742 0.451 -0.5 ...
$ POS     : num  0.191 -1.304 -1.304 -1.304 -0.632 ...
$ PROG    : num  0.5597 -0.0602 -0.1744 1.1647 0.8727 ...
$ QUAN    : num  -1.03058 -0.00848 1.56447 0.33431 -0.80474 ...
$ QUPR    : num  -0.679 -1.011 1.496 -0.61 0.843 ...
$ RB      : num  -0.716 -1.646 -0.22 0.379 -1.157 ...
$ RP      : num  -1.205 0.727 0.33 -0.877 1.065 ...
$ SPLIT   : num  -0.89 -0.89 -0.242 -0.475 1.028 ...
$ SPP2    : num  0.156 0.557 0.161 0.553 0.798 ...
$ STPR    : num  -0.249 1.604 0.253 1.234 -0.815 ...
$ THATD   : num  1.748 -1.041 -1.041 0.883 -1.041 ...
$ THRC    : num  -0.685 -0.685 -0.685 -0.685 -0.685 ...
$ THSC    : num  -0.348 -0.121 -0.882 -0.56 -0.137 ...
$ TIME    : num  -0.4164 -0.6909 0.4315 0.0699 -0.7051 ...
$ TPP3P   : num  -0.294 -0.558 -1.215 -0.937 0.715 ...
$ TPP3S   : num  -0.596 -0.32 -0.769 -0.807 -0.874 ...
$ TTR     : num  -0.92586 -0.31359 0.91095 -0.00745 2.3979 ...
$ VBD     : num  -0.951 -0.594 0.662 -0.858 -0.189 ...
$ VBG     : num  -0.959 0.875 0.208 0.907 1.607 ...
$ VBN     : num  -0.818 0.376 -0.818 -0.818 0.741 ...
$ VIMP    : num  1.963 1.658 -0.703 -0.539 -0.542 ...
$ VPRT    : num  -0.635 -0.796 -1.059 1.438 0.34 ...
$ WHQU    : num  1.4997 0.8327 0.0894 0.54 -1.0783 ...
$ WHSC    : num  -0.569 -0.375 -0.5 -0.84 1.346 ...
$ XXO     : num  -1.049 -0.667 -0.14 0.233 -0.684 ...
$ YNQU    : num  -0.1051 1.1592 0.4093 0.0709 -1.037 ...
```

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

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
```

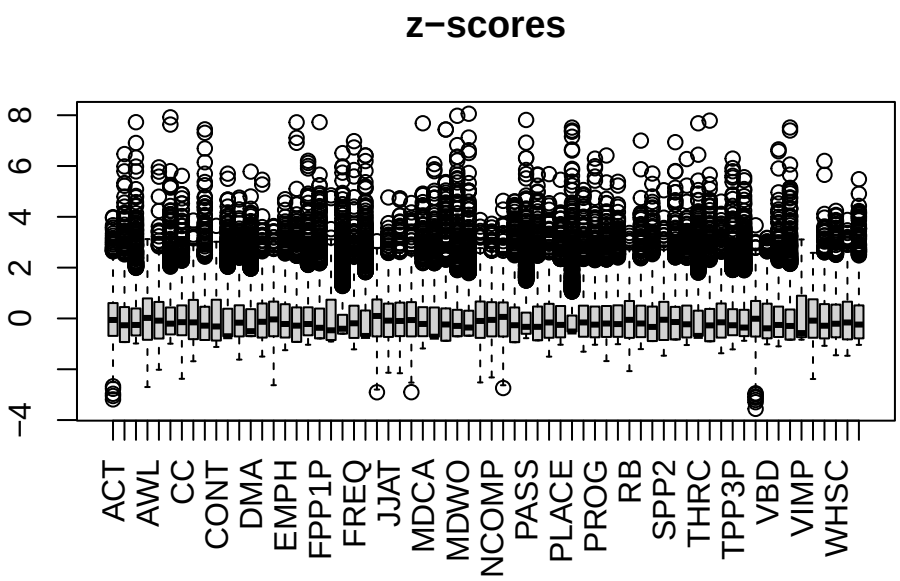
```
TxBcounts <- TxBcounts %>%
  filter(!Filename %in% outliers$Filename)

nrow(TxBcounts)
```

```
[1] 1961
```

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

nrow(TxBzcounts)
```

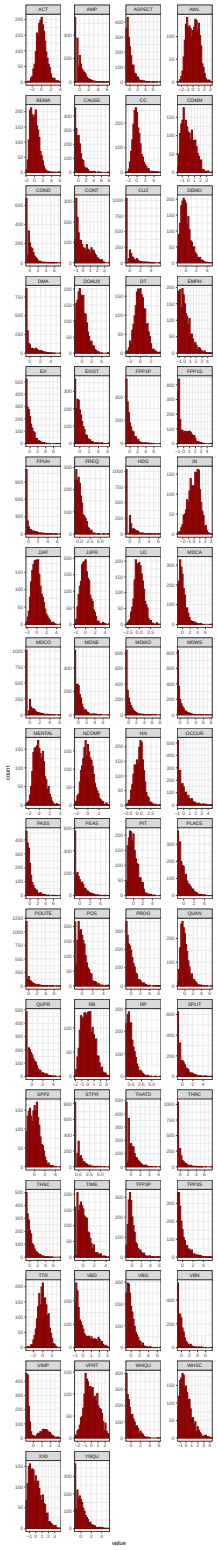
```
[1] 1961
```

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




```
#saveRDS(TxBcounts, here("processed_data", "TxBcounts3.rds")) # Last saved 16 Feb 2024
```

```
TxBzcounts %>%
  as.data.frame() %>%
  gather() %>% # This function from tidyr converts a selection of variables into two variable
  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)
```

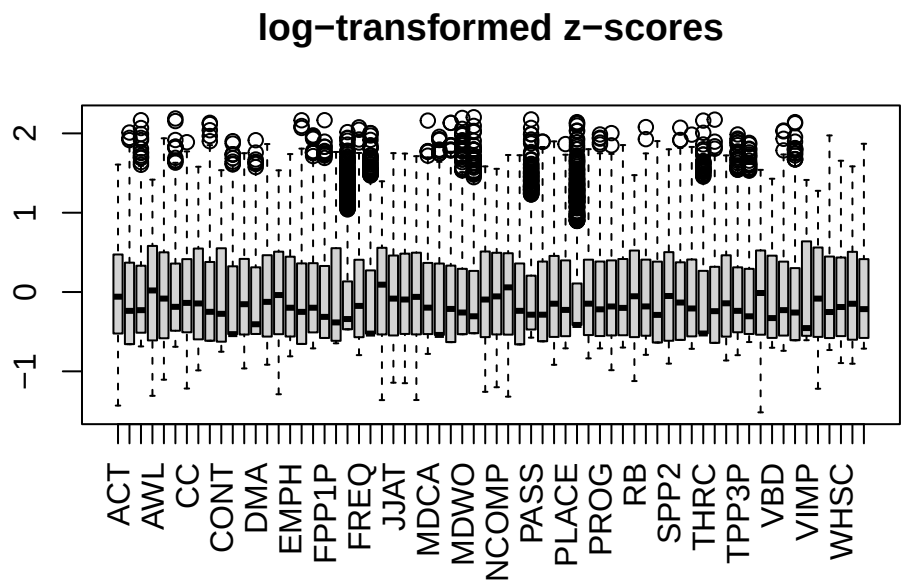

ACT
AMP
ASPECT
AWL
BENIA
CAUSE
CC
COMM
COND
CONT
CUZ
DENO
DMA
DOAUX
DT
EMPH
EX
EXIST
FPFUP
FPP1S
FPUH
FREQ
HDG
IN
INF
JJPR
LD
MDCA
MDCO
MDNE
MDWO
MDWS
MENTAL
NCOMP
NN
OCCUR
PASS
PEAS
PIT
PLACE
POLITE
POS
PREG
QUAN
QUPR
RB
RP
SPLIT
SPIV
STPR
THATD
THRC
THSC
TIME
TPFSP
TPP2S
TTR
VBD
VBG
VBN
VIMP
VPRT
WHQU
WHSC
XX0
YNQU
count
value

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

### 3.2.3 Transforming the features to (partially) deskew these distributions

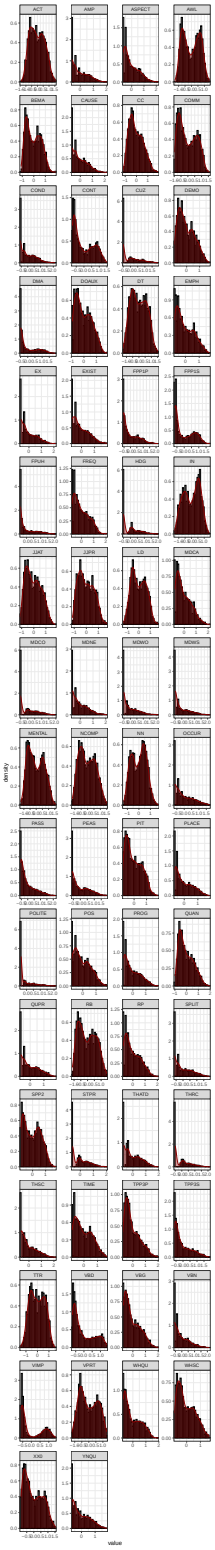
Signed log transformation function inspired by the SignedLog function proposed in https://cran.r-project.org/web/packages/DataVisualizations/DataVisualizations.pdf

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

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

# The function above would only transform the most skewed variables. This is what Lee suggest
# TxBzlogcounts2 <- TxBzcounts %>%
#   as.data.frame() %>%
#     mutate(across(.cols = c(AMP, ASPECT, CAUSE, COND, CUZ, DMA, EMPH, EX, EXIST, FPP1P, FPI
#         .fns = signed.log)) %>%
#     rename_with(.cols = c(AMP, ASPECT, CAUSE, COND, CUZ, DMA, EMPH, EX, EXIST, FPP1P, FPP1S
#                 .fn = ~paste0(., '_signedlog'))

boxplot(TxBzlogcounts, las=3, main="log-transformed z-scores")
```




```
#saveRDS(TxBzlogcounts, here("processed_data", "TxBzlogcounts.rds")) # Last saved 16 Feb 2024
```

```
TxBzlogcounts %>%
  as.data.frame() %>%
  gather() %>% # This function from tidyr converts a selection of variables into two variable
  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)
```

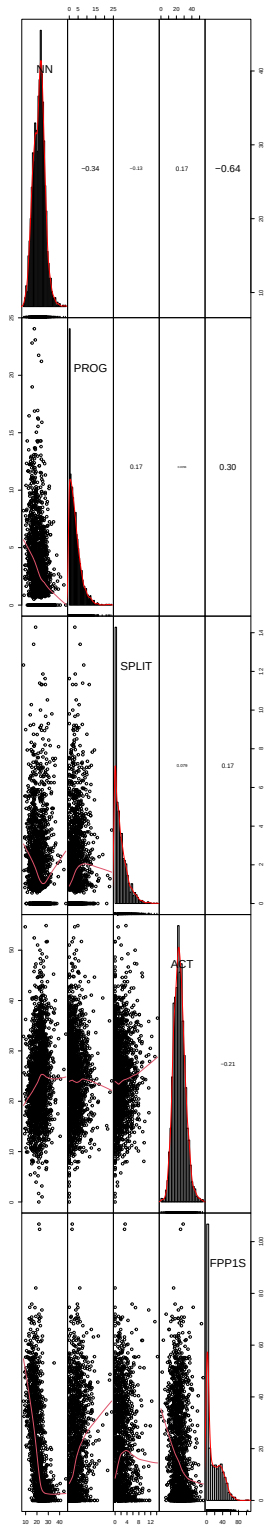

ACT
AMP
ASPECT
AWL
BRAA
CAUSE
CC
COMM
COND
CONT
CU2
DEMO
DMA
DOAUX
DT
EMPH
EX
EXIST
FPFSP
FPFSS
FPUM
FREQ
HDG
IN
JJAT
JJPR
LID
MODA
MOCO
MONE
MONO
MOWS
NEATAL
NCOMP
NU
OCCUR
PAGS
PEAS
PIT
PLACE
POLITE
POS
PROG
QUAN
QUPR
RB
RP
SPLIT
SPP2
STPR
THATD
THEC
THSC
TIME
TPFSP
TPFSS
TTR
VBD
VBG
VBN
VIMP
VPRT
WHQU
WHSC
XXD
YNQU
density
value

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

These plots serve to illustrate the effects of the variable transformations performed in the above chunks.

```
# This is a slightly amended version of the PerformanceAnalytics::chart.Correlation() functio

chart.Correlation.nostars <- function (R, histogram = TRUE, method = c("pearson", "kendall",
  x = checkData(R, method = "matrix")
  if (missing(method))
    method = method[1]
  panel.cor <- function(x, y, digits = 2, prefix = "", use = "pairwise.complete.obs", method
    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
chart.Correlation.nostars(example1, histogram=TRUE, pch=19)
```


NN
PROG
SPLIT
ACT
FPP1S
−0.34
−0.13
0.17
−0.64
0.17
0.30
0.17
−0.21

```
dev.off()
```

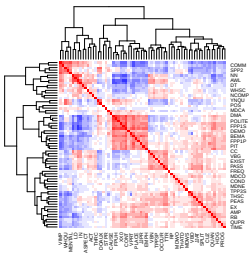
```
null device
          1
```

```
# 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, u
chart.Correlation.nostars(example2, histogram=TRUE, pch=19)
dev.off()
```

```
null device
          1
```

### 3.2.4 Visualisation of feature correlations

```
# 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", r
heatmap(cor(TxBzlogcounts),
        symm=TRUE,
        zlim=c(-1,1),
        col=cor.colours,
        margins=c(7,7))
```



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

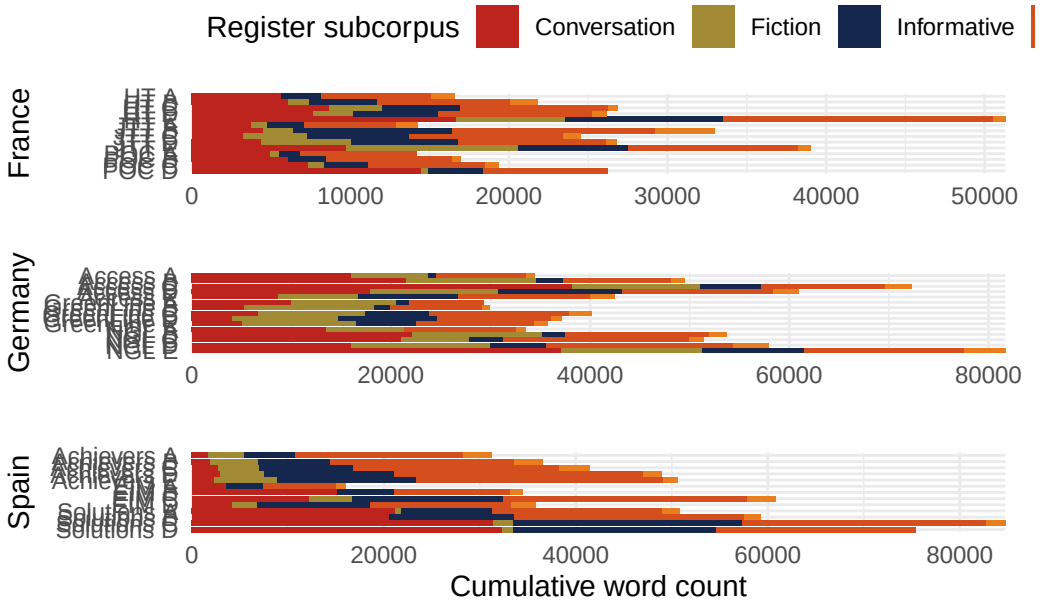
## 3.3 Composition of TEC texts/files entered in the MDAs

```
# Total number of words
TxBcounts %>% summarise(sum(Words))
```

```
  sum(Words)
1    1693650
```

```
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
palette <- c("#BD241E", "#A18A33", "#15274D", "#D54E1E", "#EA7E1E", "#4C4C4C", "#722672", "#

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, le
    theme_minimal() + theme(legend.position = "top", legend.justification = "left")

PlotFr /
PlotGer /
PlotSp
```




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

```
# Meta-data on % of instructional language in each textbook
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)

metadataInstr
```

```
# A tibble: 9 x 2
  Series    InstrWordcount
  <fct>              <int>
1 Achievers         109886
2 Solutions          87829
```

```
3 EIM                59928
4 HT                 51550
5 Access             60938
6 NGL                79312
7 JTT                48375
8 GreenLine          54263
9 POC                30548
```

```
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))
```

```
     Series TECwordcount InstrWordcount InstrucPercent
1    Access       259679          60938          23.47
2       NGL       278316          79312          28.50
3 GreenLine       172267          54263          31.50
4 Solutions       270278          87829          32.50
5       JTT       137557          48375          35.17
6        HT       142676          51550          36.13
7       POC        76714          30548          39.82
8       EIM       147185          59928          40.72
9 Achievers       208978         109886          52.58
```

# 4 Packages used in this script

# 5 Summary

In summary, this book has no content whatsoever.

# References

Knuth, Donald E. 1984. "Literate Programming." *Comput. J.* 27 (2): 97–111. https://doi.org/10.1093/comjnl/27.2.97.