# Supplementary material

Language structure is influenced by the proportion of non-native speakers: A reply to Koplenig (2019)

Henri Kauhanen, Sarah Einhaus & George Walkden

15 December 2022

## Contents

**1 Introduction** **1**

**2 Prerequisites** **2**

**3 Data preprocessing** **2**

**4 Descriptive statistics** **2**
4.1 General characteristics of the dataset . . . 2
4.2 How many non-vehiculars have an imputed L2 proportion? . . . 3
4.3 In how many cases is the data imputation wrong? . . . 3

**5 Remove zero-imputation from uncertain non-vehiculars** **3**

**6 Overall missingness in the data** **4**

**7 Family and area coverage** **5**

**8 Complete cases analysis** **5**
8.1 Morphological complexity . . . 6
8.2 Morphological complexity, $\geq 6$ features . . . 6
8.3 Information-theoretic complexity . . . 7

**9 Multiple imputation analysis** **8**
9.1 Preparatory steps . . . 8
9.2 Separate imputation models . . . 9
9.2.1 Morphological complexity . . . 9
9.2.2 Information-theoretic complexity . . . 10
9.3 Single imputation model . . . 12

**10 Plots** **14**
10.1 Histogram of L2 speaker proportion . . . 14
10.2 Effects plots (complete cases analysis) . . . 15
10.3 Histogram of L2 proportion coefficients in imputation analysis . . . 15

# 1 Introduction

This supplementary materials document provides all the statistical analyses reported in the paper, plus additional analyses that include interaction effects and a second imputation analysis in which a single imputation model is used to regress both types of complexity.

Tested with R version 4.0.4.

# 2 Prerequisites

The following R packages are required:

```
library(lme4)
library(lmerTest)
library(mice)
library(broom.mixed)
library(lattice)
library(effects)
library(gridExtra)
```

# 3 Data preprocessing

The raw data resides in two files (Koplenig's original dataset and our additions) in the `../data` directory. We first merge these two datasets and carry out a few transformations that will facilitate data analysis.

```
kop <- read.csv("../data/rsos181274supp2.csv", stringsAsFactors=FALSE)
new <- read.csv("../data/koplenig-reply.csv", stringsAsFactors=FALSE)
new <- new[, c("ISO", "ethnologue_L2_users", "used_as_L2_by", "notes")]
```

Format the data slightly differently:

```
for (i in 1:nrow(new)) {
  tmp <- paste(stringr::str_extract_all(new[i, ]$used_as_L2_by,
                                        pattern="\\[[a-z]+\\]")[[1]], collapse=":")
  new[i, ]$used_as_L2_by <- stringr::str_replace_all(tmp, pattern="[\\[\\]]", "")
}
new$used_as_L2_by <- ifelse(new$used_as_L2_by=="", NA, new$used_as_L2_by)
```

Merge the two dataframes:

```
data <- merge(kop, new, by="ISO")
```

For some languages, the area is missing, but these are read in as empty strings rather than as missing values. Need to fix that:

```
data$Area <- ifelse(data$Area == "", NA, data$Area)
data$Family <- ifelse(data$Family == "", NA, data$Family)
```

Make sure language family and area are factors (important for imputation model and regression analysis):

```
data$Family <- factor(data$Family)
data$Area <- factor(data$Area)
```

Encode the logarithm of population size and the logarithm of the range size as variables in the dataframe (useful for some of the regressions and plots):

```
data$logPop <- log(data$Population)
data$logRangesize <- log(data$Rangesize)
```

# 4 Descriptive statistics

## 4.1 General characteristics of the dataset

There are a total of

```
nrow(data)
```

```
## [1] 2143
```

languages in the dataset. However, not every language has data for each column of the data frame. The number of vehicular languages is

```r
nrow(data[data$vehicularity==1, ])
```

```
## [1] 241
```

Of these,

```r
nrow(data[data$vehicularity==1 & is.na(data$L2prop), ])
```

```
## [1] 152
```

do not have an L2 proportion estimate (either real or imputed).

The number of non-vehicular languages is

```r
nrow(data[data$vehicularity==0, ])
```

```
## [1] 1902
```

These all have an L2 proportion estimate, either real or imputed:

```r
nrow(data[data$vehicularity==0 & is.na(data$L2prop), ])
```

```
## [1] 0
```

### 4.2 How many non-vehiculars have an imputed L2 proportion?

The number of non-vehicular languages with a zero L2 proportion is

```r
nv0 <- nrow(data[data$vehicularity==0 & data$L2prop==0, ])
nv0
```

```
## [1] 1824
```

Of these, Ethnologue actually provides a numerical zero L2 proportion estimate for

```r
nv0E <- nrow(data[data$vehicularity==0 & data$L2prop==0 &
             data$ethnologue_L2_users==TRUE, ])
nv0E
```

```
## [1] 4
```

languages. The rest have been imputed.

### 4.3 In how many cases is the data imputation wrong?

Ethnologue notes that the language is used as an L2 by speakers of some other set of languages (without giving numerical estimates) in

```r
asL2 <- nrow(data[data$vehicularity==0 & data$L2prop==0 &
             !is.na(data$used_as_L2_by), ])
asL2
```

```
## [1] 404
```

of these cases. In other words, the data imputation is definitely wrong for

```r
asL2/(nv0 - nv0E)
```

```
## [1] 0.221978
```

of the dataset.

## 5 Remove zero-imputation from uncertain non-vehiculars

We now remove the zero-imputed L2 proportions from uncertain non-vehicular languages:

```r
data2 <- data
data2$L2prop <- ifelse(data2$vehicularity==0 & data2$L2prop==0 &
                        data2$ethnologue_L2_users==FALSE, NA, data2$L2prop)
```

There are now

```r
nrow(data2[!is.na(data2$L2prop), ])
```

```
## [1] 171
```

languages with a non-`NA` L2 proportion. Of these,

```r
nrow(data2[!is.na(data2$L2prop) & data2$vehicularity==1, ])
```

```
## [1] 89
```

are vehicular and

```r
nrow(data2[!is.na(data2$L2prop) & data2$vehicularity==0, ])
```

```
## [1] 82
```

non-vehicular.

We point out that there are missing values also in the response variables, morphological complexity and information-theoretic complexity. In other words, the two complexity measures are available for different subsets of languages:

```r
nrow(data2[!is.na(data2$MC), ])
```

```
## [1] 1581
```

```r
nrow(data2[!is.na(data2$H), ])
```

```
## [1] 1088
```

In particular, in the subset of languages with a non-missing L2 proportion, these numbers are:

```r
nrow(data2[!is.na(data2$L2prop) & !is.na(data2$MC), ])
```

```
## [1] 148
```

```r
nrow(data2[!is.na(data2$L2prop) & !is.na(data2$H), ])
```

```
## [1] 94
```

# 6 Overall missingness in the data

The variables in the dataset now have this many missing values:

```r
nrow(data2[is.na(data2$Family), ])
```

```
## [1] 0
```

```r
nrow(data2[is.na(data2$Area), ])
```

```
## [1] 414
```

```r
nrow(data2[is.na(data2$MC), ])
```

```
## [1] 562
```

```r
nrow(data2[is.na(data2$H), ])
```

```
## [1] 1055
```

```r
nrow(data2[is.na(data2$L2prop), ])
```

```
## [1] 1972
```

```
nrow(data2[is.na(data2$Population), ])
```

```
## [1] 0
```

```
nrow(data2[is.na(data2$Rangesize), ])
```

```
## [1] 22
```

# 7 Family and area coverage

In the entire dataset, there are

```
length(unique(data2$Family))
```

```
## [1] 126
```

unique language families and

```
length(unique(data2$Area))
```

```
## [1] 25
```

unique linguistic areas. The three most frequent families have a fraction of

```
sum(sort(as.numeric(table(data2$Family)),
     decreasing=TRUE)[1:3])/sum(!is.na(data2$Family))
```

```
## [1] 0.3952403
```

of the languages. The three most frequent areas have a fraction of

```
sum(sort(as.numeric(table(data2$Area)),
     decreasing=TRUE)[1:3])/sum(!is.na(data2$Area))
```

```
## [1] 0.3233083
```

of the languages (not counting languages for which area is missing).

The above statistics for our reduced sample are:

```
data2b <- data2
data2b <- data2b[!is.na(data2b$L2prop), ]
length(unique(data2b$Family))
```

```
## [1] 29
```

```
length(unique(data2b$Area))
```

```
## [1] 21
```

```
sum(sort(as.numeric(table(data2b$Family)),
     decreasing=TRUE)[1:3])/sum(!is.na(data2b$Family))
```

```
## [1] 0.4853801
```

```
sum(sort(as.numeric(table(data2b$Area)),
     decreasing=TRUE)[1:3])/sum(!is.na(data2b$Area))
```

```
## [1] 0.4580645
```

# 8 Complete cases analysis

In the complete cases analysis, we do not impute any missing values.

## 8.1 Morphological complexity

```
mod <- lmer(MC~L2prop+logPop+(1|Family)+(1|Area), data2)
summary(mod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: MC ~ L2prop + logPop + (1 | Family) + (1 | Area)
##    Data: data2
##
## REML criterion at convergence: 21.5
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.80485 -0.54828  0.04692  0.49637  2.73758
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  Family   (Intercept) 0.0002365 0.01538
##  Area     (Intercept) 0.0136480 0.11682
##  Residual             0.0551430 0.23483
## Number of obs: 144, groups:  Family, 28; Area, 19
##
## Fixed effects:
##               Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)   0.816737   0.084203  52.135172   9.700 2.87e-13 ***
## L2prop       -0.243060   0.082497 114.936041  -2.946  0.00389 **
## logPop       -0.015513   0.005734  36.541790  -2.705  0.01030 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##        (Intr) L2prop
## L2prop -0.482
## logPop -0.848  0.251
```

Adding an interaction between L2 proportion and population size leads to a worse model:

```
modb <- lmer(MC~L2prop*logPop+(1|Family)+(1|Area), data2)
AIC(mod)
```

```
## [1] 33.49946
```

```
AIC(modb)
```

```
## [1] 39.12415
```

## 8.2 Morphological complexity, $\geq$ 6 features

When only looking at languages in which at least 6 features are available for the determination of morphological complexity, we cannot include the same random effects structure because it leads to a singular fit:

```
mod6 <- lmer(MC~L2prop+logPop+(1|Family)+(1|Area), data2[data2$NumChap>=6, ])
```

```
## boundary (singular) fit: see ?isSingular
```

Apparently, this is because the area is missing for many languages. Hence we run the following, simpler model instead:

```
mod6 <- lmer(MC~L2prop+logPop+(1|Family), data2[data2$NumChap>=6, ])
summary(mod6)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: MC ~ L2prop + logPop + (1 | Family)
##    Data: data2[data2$NumChap >= 6, ]
##
## REML criterion at convergence: -28.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.60076 -0.49290  0.05634  0.62396  1.99292
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Family   (Intercept) 0.01760  0.1327
##  Residual             0.03071  0.1752
## Number of obs: 101, groups:  Family, 24
##
## Fixed effects:
##              Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)  0.780778   0.078449 66.528252   9.953 8.38e-15 ***
## L2prop      -0.218190   0.078221 97.854214  -2.789  0.00635 **
## logPop      -0.017764   0.005502 62.692744  -3.229  0.00198 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##        (Intr) L2prop
## L2prop -0.474
## logPop -0.846  0.260
```

Adding an interaction between L2 proportion and population size again leads to a worse model:

```
mod6b <- lmer(MC~L2prop*logPop+(1|Family), data2[data2$NumChap>=6, ])
AIC(mod6)
```

```
## [1] -18.35238
```

```
AIC(mod6b)
```

```
## [1] -9.698234
```

## 8.3 Information-theoretic complexity

For information-theoretic complexity, the random effects structure with a random intercept for area again leads to a singular fit:

```
modIC <- lmer(H~L2prop+logPop+(1|Family)+(1|Area), data2)
```

```
## boundary (singular) fit: see ?isSingular
```

Hence we only include a random intercept for family:

```
modIC <- lmer(H~L2prop+logPop+(1|Family), data2)
summary(modIC)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: H ~ L2prop + logPop + (1 | Family)
##    Data: data2
##
## REML criterion at convergence: 14.4
##
```

```
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1438 -0.4771 -0.1613  0.3000  4.0062
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Family   (Intercept) 0.01599  0.1265
##  Residual             0.05339  0.2311
## Number of obs: 94, groups:  Family, 13
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  1.42854    0.18511 71.44401   7.717  5.4e-11 ***
## L2prop      -0.16136    0.10037 90.95967  -1.608    0.111
## logPop       0.01810    0.01182 77.91547   1.532    0.130
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##        (Intr) L2prop
## L2prop -0.329
## logPop -0.955  0.201
```

Adding an interaction between L2 proportion and population size leads to a worse model:

```
modICb <- lmer(H~L2prop*logPop+(1|Family), data2)
AIC(modIC)
```

```
## [1] 24.4098
```

```
AIC(modICb)
```

```
## [1] 31.4476
```

# 9 Multiple imputation analysis

## 9.1 Preparatory steps

We provide two different kinds of analysis: first, one which has separate imputation models for the two kinds of complexities, and second, an analysis which has a single imputation model for both complexities.

The rationale for constructing separate imputation models (the analysis reported in the main paper) is that the number of languages for which both morphological complexity and information-theoretic complexity are attested is rather small:

```
nrow(data2[!is.na(data2$MC), ])
```

```
## [1] 1581
```

```
nrow(data2[!is.na(data2$H), ])
```

```
## [1] 1088
```

```
nrow(data2[!is.na(data2$MC) & !is.na(data2$H), ])
```

```
## [1] 526
```

All of our imputation models take language family as a clustering variable. The implementation in *mice* requires this as a numeric:

```
datai <- data2
datai$cluster <- as.numeric(datai$Family)
```

Also, we want to make sure that impossible L2 speaker proportions (outside the interval [0,1]) are never imputed. To do this, we take a logit transform of L2 speaker proportion:

```
epsilon <- 10^-5
datai$L2prop_t <- epsilon + (1 - 2*epsilon)*datai$L2prop
datai$L2prop_t <- log(datai$L2prop_t/(1 - datai$L2prop_t))
```

## 9.2 Separate imputation models

### 9.2.1 Morphological complexity

We need the following variables in this imputation model:

```
datai_MC <- datai[, c("ISO", "Language", "Family", "Area", "L2prop_t", "MC",
                      "logPop", "logRangesize", "cluster")]
```

We first set up the predictor matrix. L2 speaker proportion is imputed using morphological complexity, logarithmic population size and logarithmic range size, with language family as a clustering variable. We do not impute other missing values; we have tried to do so, but the model becomes too complicated to run.

```
pred_MC <- make.predictorMatrix(datai_MC)
pred_MC[1:nrow(pred_MC), ] <- 0
pred_MC["L2prop_t", ] <- c(0, 0, 0, 0, 0, 1, 1, 1, -2)
pred_MC
```

```
##              ISO Language Family Area L2prop_t MC logPop logRangesize cluster
## ISO            0        0      0    0        0  0      0            0       0
## Language       0        0      0    0        0  0      0            0       0
## Family         0        0      0    0        0  0      0            0       0
## Area           0        0      0    0        0  0      0            0       0
## L2prop_t       0        0      0    0        0  1      1            1      -2
## MC             0        0      0    0        0  0      0            0       0
## logPop         0        0      0    0        0  0      0            0       0
## logRangesize   0        0      0    0        0  0      0            0       0
## cluster        0        0      0    0        0  0      0            0       0
```

We also need to set the imputation method:

```
impmethod_MC <- character(ncol(datai_MC))
names(impmethod_MC) <- colnames(datai_MC)
impmethod_MC["L2prop_t"] <- "2l.lmer"
impmethod_MC
```

```
##          ISO     Language       Family         Area     L2prop_t           MC
##           ""           ""           ""           ""    "2l.lmer"           ""
##       logPop logRangesize      cluster
##           ""           ""           ""
```

We now construct the imputation model. Note that we do not need more than one iteration, as missing values are only imputed in one variable.

```
imp_MC <- mice(datai_MC, method=impmethod_MC, predictorMatrix=pred_MC, maxit=1,
               m=100, print=FALSE, seed=202212)
```

Finally, we run the regression analysis on the $m = 100$ completed copies of the dataset:

```
modImp <- with(imp_MC, lmer(MC~L2prop_t+logPop+(1|Family)+(1|Area)))
tidy(pool(modImp))
```

```
##          term    estimate   std.error statistic      p.value            b
## 1 (Intercept)  0.72986298 0.036108997 20.212773 0.000000e+00 2.868957e-04
## 2    L2prop_t -0.01874615 0.006962809 -2.692326 8.837320e-03 4.267541e-05
## 3      logPop -0.01365290 0.003093889 -4.412861 1.200876e-05 2.471982e-06
```

```
##          df dfcom       fmi    lambda   m       riv         ubar
## 1 734.64192  1493 0.2243449 0.2222361 100 0.2857372 1.014095e-03
## 2  71.27845  1493 0.8920452 0.8890580 100 8.0137216 5.378545e-06
## 3 627.13709  1493 0.2631759 0.2608298 100 0.3528684 7.075448e-06
```

For purposes of illustration, here is the regression on one of the 100 completed datasets (that is, before pooling):

```
summary(modImp$analyses[[1]])
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: MC ~ L2prop_t + logPop + (1 | Family) + (1 | Area)
##
## REML criterion at convergence: 343.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.93930 -0.64458 -0.04188  0.79286  2.52285
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Family   (Intercept) 0.005923 0.07696
##  Area     (Intercept) 0.007056 0.08400
##  Residual             0.068388 0.26151
## Number of obs: 1499, groups:  Family, 122; Area, 24
##
## Fixed effects:
##               Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)  7.374e-01  3.232e-02  1.233e+02  22.813  < 2e-16 ***
## L2prop_t    -7.829e-03  2.412e-03  1.412e+03  -3.246   0.0012 **
## logPop      -1.231e-02  2.718e-03  1.021e+03  -4.530  6.6e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) L2prp_
## L2prop_t  0.025
## logPop   -0.699  0.166
```

### 9.2.2 Information-theoretic complexity

We need the following variables in this imputation model:

```
datai_H <- datai[, c("ISO", "Language", "Family", "Area", "L2prop_t", "H",
                     "logPop", "logRangesize", "cluster")]
```

We first set up the predictor matrix. L2 speaker proportion is imputed using information-theoretic complexity, logarithmic population size and logarithmic range size, with language family as a clustering variable. We do not impute other missing values; we have tried to do so, but the model becomes too complicated to run.

```
pred_H <- make.predictorMatrix(datai_H)
pred_H[1:nrow(pred_H), ] <- 0
pred_H["L2prop_t", ] <- c(0, 0, 0, 0, 0, 1, 1, 1, -2)
pred_H
```

```
##              ISO Language Family Area L2prop_t H logPop logRangesize cluster
## ISO            0        0      0    0        0 0      0            0       0
## Language       0        0      0    0        0 0      0            0       0
## Family         0        0      0    0        0 0      0            0       0
```

```
## Area            0         0      0    0        0 0      0            0       0
## L2prop_t        0         0      0    0        0 1      1            1      -2
## H               0         0      0    0        0 0      0            0       0
## logPop          0         0      0    0        0 0      0            0       0
## logRangesize    0         0      0    0        0 0      0            0       0
## cluster         0         0      0    0        0 0      0            0       0
```

We also need to set the imputation method:

```
impmethod_H <- character(ncol(datai_H))
names(impmethod_H) <- colnames(datai_H)
impmethod_H["L2prop_t"] <- "2l.lmer"
impmethod_H
```

```
##          ISO     Language       Family          Area     L2prop_t              H
##           ""           ""           ""            ""    "2l.lmer"             ""
##       logPop logRangesize      cluster
##           ""           ""           ""
```

We now construct the imputation model. Note that we do not need more than one iteration, as missing values are only imputed in one variable.

```
imp_H <- mice(datai_H, method=impmethod_H, predictorMatrix=pred_H, maxit=1,
              m=100, print=FALSE, seed=2202212)
```

Regression for information-theoretic complexity:

```
modImpIC <- with(imp_H, lmer(H~L2prop_t+logPop+(1|Family)+(1|Area)))
tidy(pool(modImpIC))
```

```
##          term     estimate   std.error  statistic      p.value            b
## 1 (Intercept)  1.298043169 0.048003183 27.0407728 0.000000e+00 4.231238e-05
## 2    L2prop_t -0.002607529 0.004203372 -0.6203422 5.360764e-01 1.068608e-05
## 3      logPop  0.021600461 0.002743905  7.8721601 1.465494e-14 4.375615e-07
##          df dfcom        fmi     lambda   m        riv         ubar
## 1 694.1809   711 0.02136143 0.01854593 100 0.01889639 2.261570e-03
## 2 135.2499   711 0.61649288 0.61086341 100 1.56979177 6.875396e-06
## 3 652.2416   711 0.06157101 0.05869786 100 0.06235815 7.087079e-06
```

For purposes of illustration, here is the regression on one of the 100 completed datasets (that is, before pooling):

```
summary(modImpIC$analyses[[1]])
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: H ~ L2prop_t + logPop + (1 | Family) + (1 | Area)
##
## REML criterion at convergence: -741
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.4771 -0.5538 -0.0305  0.5657  7.5260
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  Family   (Intercept) 0.0007237 0.0269
##  Area     (Intercept) 0.0288797 0.1699
##  Residual             0.0178086 0.1334
## Number of obs: 717, groups:  Family, 79; Area, 23
##
## Fixed effects:
```

```
##                Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)    1.291823   0.047295  42.449622  27.314  < 2e-16 ***
## L2prop_t      -0.006657   0.002607 646.960573  -2.554   0.0109 *
## logPop         0.021585   0.002626 654.644187   8.219  1.1e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) L2prp_
## L2prop_t  0.046
## logPop   -0.613  0.051
```

## 9.3 Single imputation model

For the analysis with a single imputation model, we include both morphological and information-theoretic complexity:

```
datai_s <- datai[, c("ISO", "Language", "Family", "Area", "L2prop_t", "MC",
                      "H", "logPop", "logRangesize", "cluster")]
```

We first set up the predictor matrix. L2 speaker proportion is imputed using morphological complexity, information-theoretic complexity, logarithmic population size and logarithmic range size, with language family as a clustering variable. We do not impute other missing values; we have tried to do so, but the model becomes too complicated to run.

```
pred_s <- make.predictorMatrix(datai_s)
pred_s[1:nrow(pred_s), ] <- 0
pred_s["L2prop_t", ] <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, -2)
pred_s
```

```
##              ISO Language Family Area L2prop_t MC H logPop logRangesize cluster
## ISO            0        0      0    0        0  0 0      0            0       0
## Language       0        0      0    0        0  0 0      0            0       0
## Family         0        0      0    0        0  0 0      0            0       0
## Area           0        0      0    0        0  0 0      0            0       0
## L2prop_t       0        0      0    0        0  1 1      1            1      -2
## MC             0        0      0    0        0  0 0      0            0       0
## H              0        0      0    0        0  0 0      0            0       0
## logPop         0        0      0    0        0  0 0      0            0       0
## logRangesize   0        0      0    0        0  0 0      0            0       0
## cluster        0        0      0    0        0  0 0      0            0       0
```

We also need to set the imputation method:

```
impmethod_s <- character(ncol(datai_s))
names(impmethod_s) <- colnames(datai_s)
impmethod_s["L2prop_t"] <- "2l.lmer"
impmethod_s
```

```
##          ISO     Language       Family         Area     L2prop_t           MC
##           ""           ""           ""           ""    "2l.lmer"           ""
##            H       logPop logRangesize      cluster
##           ""           ""           ""           ""
```

We now construct the imputation model. Note that we do not need more than one iteration, as missing values are only imputed in one variable.

```
imp_s <- mice(datai_s, method=impmethod_s, predictorMatrix=pred_s, maxit=1,
              m=100, print=FALSE, seed=3202212)
```

Finally, we run the regression analyses on the $m = 100$ completed copies of the dataset:

```r
modImp_s <- with(imp_s, lmer(MC~L2prop_t+logPop+(1|Family)+(1|Area)))
tidy(pool(modImp_s))
```

```
##          term    estimate   std.error statistic     p.value            b
## 1 (Intercept)  0.71200919 0.064058014 11.115068 0.000000000 1.251487e-03
## 2    L2prop_t -0.02830966 0.010399004 -2.722343 0.008467932 8.789569e-05
## 3      logPop -0.01275560 0.005096583 -2.502774 0.012819845 6.873378e-06
##          df dfcom       fmi    lambda   m       riv         ubar
## 1 285.85950   571 0.3128265 0.3080355 100 0.4451608 2.839427e-03
## 2  60.16309   571 0.8265988 0.8209287 100 4.5843681 1.936464e-05
## 3 320.51934   571 0.2717895 0.2672597 100 0.3647399 1.903305e-05
```

```r
modImpIC_s <- with(imp_s, lmer(H~L2prop_t+logPop+(1|Family)+(1|Area)))
tidy(pool(modImpIC_s))
```

```
##          term     estimate   std.error   statistic      p.value            b
## 1 (Intercept) 1.2833032476 0.054378382 23.59951148 0.000000e+00 7.215062e-05
## 2    L2prop_t 0.0002734669 0.005019497  0.05448093 9.566425e-01 1.469735e-05
## 3      logPop 0.0229421835 0.003336011  6.87713126 2.016787e-11 8.028186e-07
##         df dfcom        fmi     lambda   m        riv         ubar
## 1 491.0824   507 0.02859202 0.02464387 100 0.02526653 2.884136e-03
## 2 120.1038   507 0.59584354 0.58916899 100 1.43409085 1.035103e-05
## 3 456.7498   507 0.07689236 0.07285912 100 0.07858474 1.031812e-05
```

Example regressions:

```r
summary(modImp_s$analyses[[1]])
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: MC ~ L2prop_t + logPop + (1 | Family) + (1 | Area)
##
## REML criterion at convergence: 79.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.92403 -0.63112 -0.05689  0.70294  2.48337
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Family   (Intercept) 0.003768 0.06138
##  Area     (Intercept) 0.011394 0.10674
##  Residual             0.059184 0.24328
## Number of obs: 577, groups:  Family, 80; Area, 24
##
## Fixed effects:
##                Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)    0.770688   0.054821 140.764179  14.058  < 2e-16 ***
## L2prop_t      -0.023100   0.003984 515.692456  -5.799 1.17e-08 ***
## logPop        -0.015045   0.004497 311.559315  -3.346 0.000922 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) L2prp_
## L2prop_t -0.198
## logPop   -0.853  0.284
```

```r
summary(modImpIC_s$analyses[[1]])
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: H ~ L2prop_t + logPop + (1 | Family) + (1 | Area)
##
## REML criterion at convergence: -457.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.2441 -0.5593 -0.0349  0.5194  7.1771
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  Family   (Intercept) 0.0005678 0.02383
##  Area     (Intercept) 0.0311854 0.17659
##  Residual             0.0197975 0.14070
## Number of obs: 513, groups:  Family, 74; Area, 22
##
## Fixed effects:
##              Estimate Std. Error        df t value Pr(>|t|)
## (Intercept) 1.280e+00  5.450e-02 5.202e+01  23.493  < 2e-16 ***
## L2prop_t    9.496e-04  2.779e-03 2.695e+02   0.342    0.733
## logPop      2.321e-02  3.319e-03 4.133e+02   6.992  1.1e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) L2prp_
## L2prop_t -0.195
## logPop   -0.689  0.312
```

# 10 Plots

## 10.1 Histogram of L2 speaker proportion

```
mypar1 <- list(par.main.text=list(just="left", x=grid::unit(21.5, "mm")))
mypar2 <- list(par.main.text=list(just="left", x=grid::unit(27.5, "mm")))
mycol <- "azure2"

g1 <- histogram(~L2prop, data2[!is.na(data2$L2prop) & data2$vehicularity==0, ],
                type="percent",
                col=mycol, xlab="L2 speaker proportion", par.settings=mypar1,
                main=list("Non-vehicular languages", cex=1.0), nint=13)

g2 <- histogram(~L2prop, data2[!is.na(data2$L2prop) & data2$vehicularity==1, ],
                type="percent",
                col=mycol, xlab="L2 speaker proportion", par.settings=mypar2,
                main=list("Vehicular languages", cex=1.0), nint=13)

# save as pdf
pdf("../plots/histogram.pdf", height=3, width=7)
grid.arrange(g1, g2, nrow=1, ncol=2)
dev.off()
```

```
## pdf
##   2
```

## 10.2 Effects plots (complete cases analysis)

```
g1 <- plot(predictorEffect("L2prop", mod), xlab="L2 speaker proportion",
          ylim=c(0,1), ylab="Morphological complexity",
          main="A                                  ")
g2 <- plot(predictorEffect("logPop", mod), xlab="log(population size)",
          ylim=c(0,1), ylab="Morphological complexity",
          main="B                                  ")
g3 <- plot(predictorEffect("L2prop", modIC), xlab="L2 speaker proportion",
          ylim=c(1.3, 2.0), ylab="Information-theoretic complexity",
          main="C                                  ")
g4 <- plot(predictorEffect("logPop", modIC), xlab="log(population size)",
          ylim=c(1.3, 2.0), ylab="Information-theoretic complexity",
          main="D                                  ")

# pdf out
pdf("../plots/result.pdf", height=7.5, width=7)
grid.arrange(g1, g2, g3, g4, nrow=2, ncol=2)
dev.off()
```

```
## pdf
##   2
```

## 10.3 Histogram of L2 proportion coefficients in imputation analysis

```
mypar1 <- list(par.main.text=list(just="left", x=grid::unit(21.5, "mm")))
mypar2 <- list(par.main.text=list(just="left", x=grid::unit(14.5, "mm")))
mycol <- "azure2"

getcoef <- function(X) {
  estimate <- coef(X)$Family$L2prop_t[1]
  estimate
}

d1 <- do.call(rbind, lapply(X=modImp$analyses, FUN=getcoef))
d2 <- do.call(rbind, lapply(X=modImpIC$analyses, FUN=getcoef))

g1 <- histogram(d1, type="percent",
                xlab=expression("Coefficient estimate for"~rho*"'"),
                col=mycol, par.settings=mypar1,
                main=list("Morphological complexity", cex=1.0), nint=13)

g2 <- histogram(d2, type="percent",
                xlab=expression("Coefficient estimate for"~rho*"'"),
                col=mycol, par.settings=mypar2,
                main=list("Information-theoretic complexity", cex=1.0), nint=13)

# save as pdf
pdf("../plots/imputation.pdf", height=3, width=7)
grid.arrange(g1, g2, nrow=1, ncol=2)
dev.off()
```

```
## pdf
##   2
```