The Dutch Diminutive: Between Inflection and Derivation

Analyis Script

Daniil Bondarenko

2023-01-27

0. Important notice

This script is the only one you should need to run in order to replicate the results of the study. While "preprocessing.Rmd" relies on the externally-accessed DLP2 files in order to build the experimental dataset in the first place, here the dataset is already assembled and ready for final data transformations and subsequent analysis. If you're looking to get a full picture of the pipeline and account for every data-related step within this project, feel free to peek into the preprocessing script. Otherwise, you should be good to go!

1. Import the necessary R packages

```r
library(dplyr) # For data cleanup and transformations
library(readr) # For more straightforward ways to import/export data
library(lme4) # For mixed-effects linear regression modelling
library(car) # For alternative options when it comes to plots and regressions
library(ggplot2) # For a more comprehensive way of plotting data
library(ggpubr) # For including statistical values in plots
library(effsize) # For Cohen's d, etc.
library(afex) # For p-values in lmer summaries
library(MuMIn) # For extracting R-squared values from lmers
options(scipen = 999) # For easier interpretability of the slopes
```

2. Import the experimental dataset, perform some final cleanup

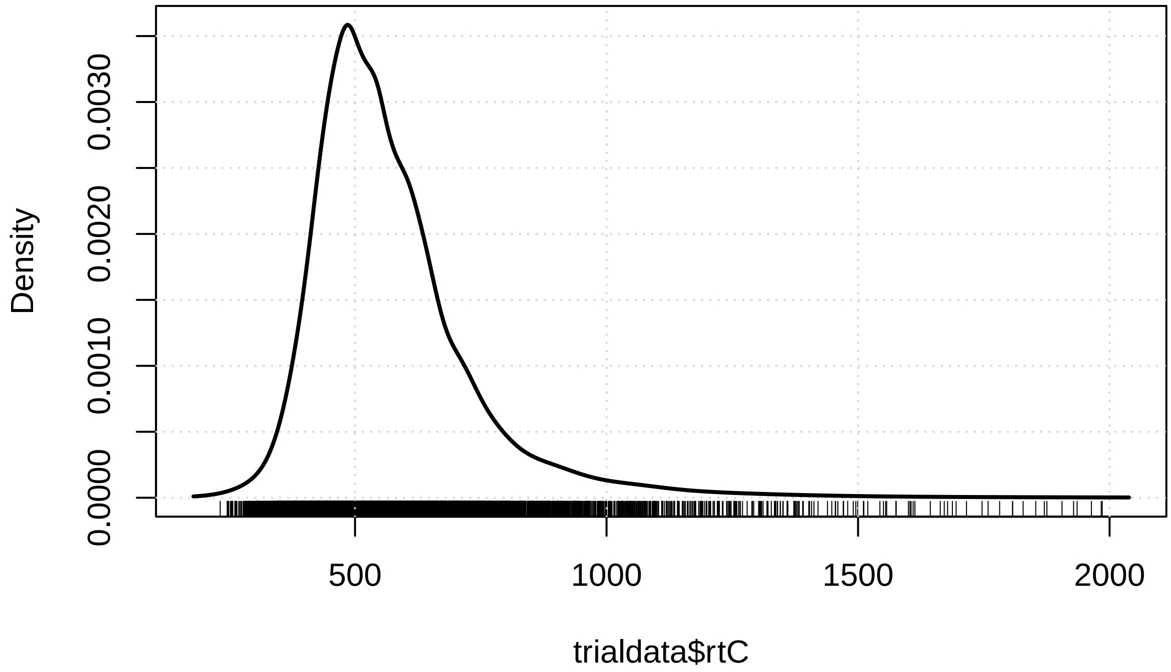
```r
trialdata <- read_csv("../corpus/trialdata.csv", show_col_types = FALSE)
# Move all relevant columns to the left for better readability
trialdata <- select(trialdata,
                    "spelling":"dec_criterion",
                    "rtC":"rateC",
                    "Length":"Colt_Nphon",
                    "item",
                    "participant",
                    "lower":"rateR",
                    "rtI":"zrateI",
                    "acc.mean":"zrateI.sd"
)
# Filter out the irrelevant columns based on theory-supported decisions
# NOTE: Done here so variables could be reintroduced at will if need be
trialdata <- select(trialdata,
                    -"N_phonemes",
                    # OLD20 chosen as the neighbourhood var: filter out the rest
                    -"Colt_N",
                    -"PLD30",
                    -"Colt_Nphon",
                    # RTs chosen as the dependent var; filter out the rest,
                    # do the centering and z-Transforming within the script;
                    # filter out the pre-existing variables
                    -"lower",
                    -"upper",
                    -"rtR",
                    -"rateR",
                    -"rtI",
                    -"rateI",
                    -"zrtC",
                    -"zrateC",
                    -"zrtI",
                    -"zrateI",
                    -"rtC.mean",
                    -"rtC.sd",
                    -"rateC.mean",
                    -"rateC.sd",
                    -"zrtC.mean",
                    -"zrtC.sd",
                    -"zrateC.mean",
                    -"zrateC.sd",
                    -"rtI.mean",
                    -"rtI.sd",
                    -"rateI.mean",
                    -"rateI.sd",
                    -"zrtI.mean",
                    -"zrtI.sd",
                    -"zrateI.mean",
                    -"zrateI.sd"
)
```

3. Run sanity checks

```
# Make sure the response variable looks fine
summary(trialdata$rtC) # Seems some responses are unreasonably long
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   232.0   466.5   540.0   574.3   637.2  1985.3
```

```
densityPlot(trialdata$rtC) # Few past 1000, even fewer past 1500
```



```
cat(
  "Only", NROW(filter(trialdata, rtC > 1500)), "observations past 1500 \n",
  "Only", NROW(filter(trialdata, rtC > 1250)), "observations past 1250 \n",
  "Only", NROW(filter(trialdata, rtC > 1000)), "observations past 1000 \n"
)
```

```
## Only 37 observations past 1500
##  Only 103 observations past 1250
##  Only 320 observations past 1000
```

```
# Need to subset by some number; 1250 as the sweet spot.
# NOTE: not subsetting the data here leads to an pretty awful skew in model
# residuals; seems not even log-transforming the rt value is enough to
# eliminate the effect of these more extreme values.
trialdata <- filter(trialdata, rtC < 1250)

# Compare number of rows per diminutive condition
trialdata <- mutate(trialdata, dim_type=factor(dim_type))
summary(trialdata$dim_type)
```

```
##  both deriv  infl
##  1461  4773  5012
```

```r
# Check distributions of all relevant predictors
trialdata %>% select("SUBTLEX2",
                     "Length",
                     "Nsyl",
                     "nmorph",
                     "OLD20",
                     "Concreteness",
                     "AoA",
                     "Word_prevalence1") %>% summary()
```

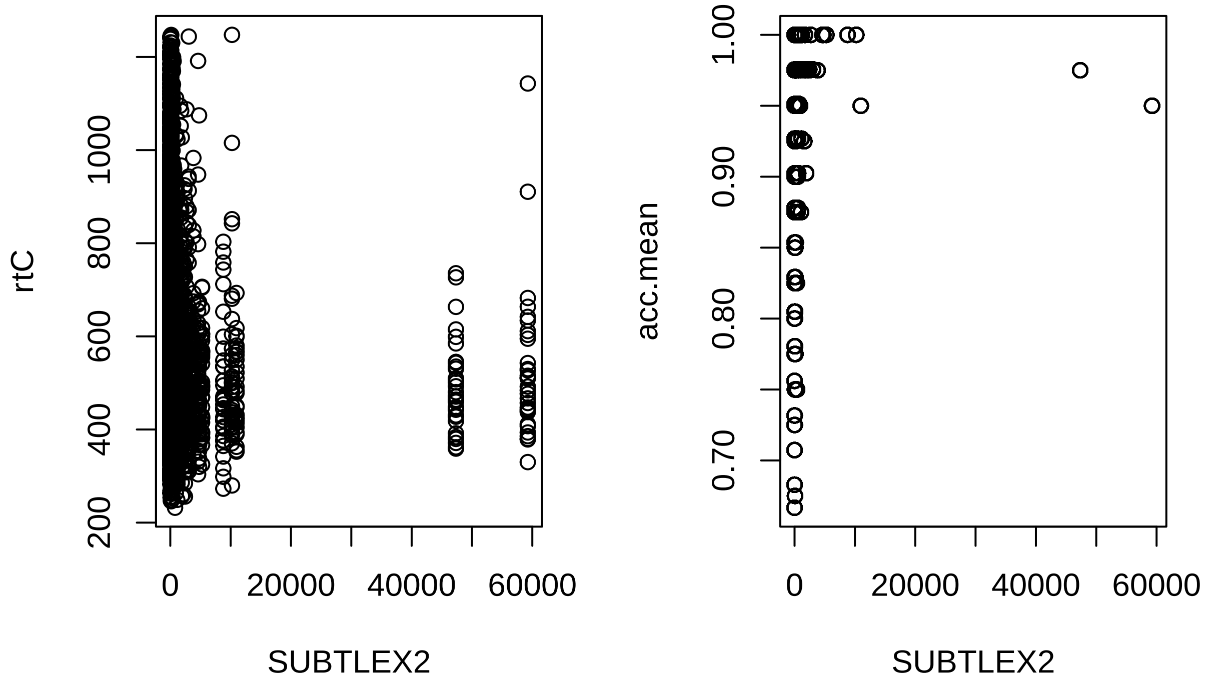
```
##     SUBTLEX2           Length            Nsyl          nmorph    
##  Min.   :    0.0   Min.   : 5.000   Min.   :2.00   Min.   :2.00  
##  1st Qu.:   32.0   1st Qu.: 6.000   1st Qu.:2.00   1st Qu.:2.00  
##  Median :  104.0   Median : 8.000   Median :3.00   Median :2.00  
##  Mean   :  740.2   Mean   : 8.251   Mean   :2.65   Mean   :2.38  
##  3rd Qu.:  268.0   3rd Qu.:10.000   3rd Qu.:3.00   3rd Qu.:3.00  
##  Max.   :59247.0   Max.   :16.000   Max.   :5.00   Max.   :4.00  
##      OLD20        Concreteness        AoA         Word_prevalence1
##  Min.   :1.000   Min.   :1.733   Min.   : 3.750   Min.   :0.561   
##  1st Qu.:1.700   1st Qu.:3.143   1st Qu.: 6.500   1st Qu.:2.212   
##  Median :1.900   Median :4.000   Median : 7.722   Median :2.504   
##  Mean   :2.402   Mean   :3.794   Mean   : 7.944   Mean   :2.434   
##  3rd Qu.:3.350   3rd Qu.:4.571   3rd Qu.: 9.333   3rd Qu.:2.765   
##  Max.   :6.050   Max.   :5.000   Max.   :13.950   Max.   :3.213
```

```r
# Check for correlations using Pearson's R
cat(
  "SUBTLEX2~Length =", cor(trialdata$SUBTLEX2, trialdata$Length), "\n",
  "Length~Nsyl =", cor(trialdata$Length, trialdata$Nsyl), "\n",
  "Length~nmorph =", cor(trialdata$Length, trialdata$nmorph), "\n",
  "Nsyl~nmorph =", cor(trialdata$Nsyl, trialdata$nmorph), "\n",
  "Length~OLD20 =", cor(trialdata$Length, trialdata$OLD20), "\n",
  "nmorph~OLD20 =", cor(trialdata$nmorph, trialdata$OLD20)
)
```

```
## SUBTLEX2~Length = -0.1234078 
##  Length~Nsyl = 0.868184 
##  Length~nmorph = 0.7210269 
##  Nsyl~nmorph = 0.7229786 
##  Length~OLD20 = 0.8893472 
##  nmorph~OLD20 = 0.759867
```

```r
# Both length and morpheme count pretty highly correlated with syllable count;
# Keeping Nsyl has much less of a theoretical reason and might impact the
# effects of the other two predictors; exclude it from the analysis.
trialdata <- select(trialdata, -"Nsyl")

# There's only two items past the 11000 mark
par(mfrow = c(1, 2))
with(trialdata, plot(SUBTLEX2, rtC))
with(trialdata, plot(SUBTLEX2, acc.mean))
```



```
# Find out what they are
trialdata %>%
  select("spelling", "SUBTLEX2") %>%
  distinct() %>%
  arrange(desc(SUBTLEX2)) %>%
  head()
```

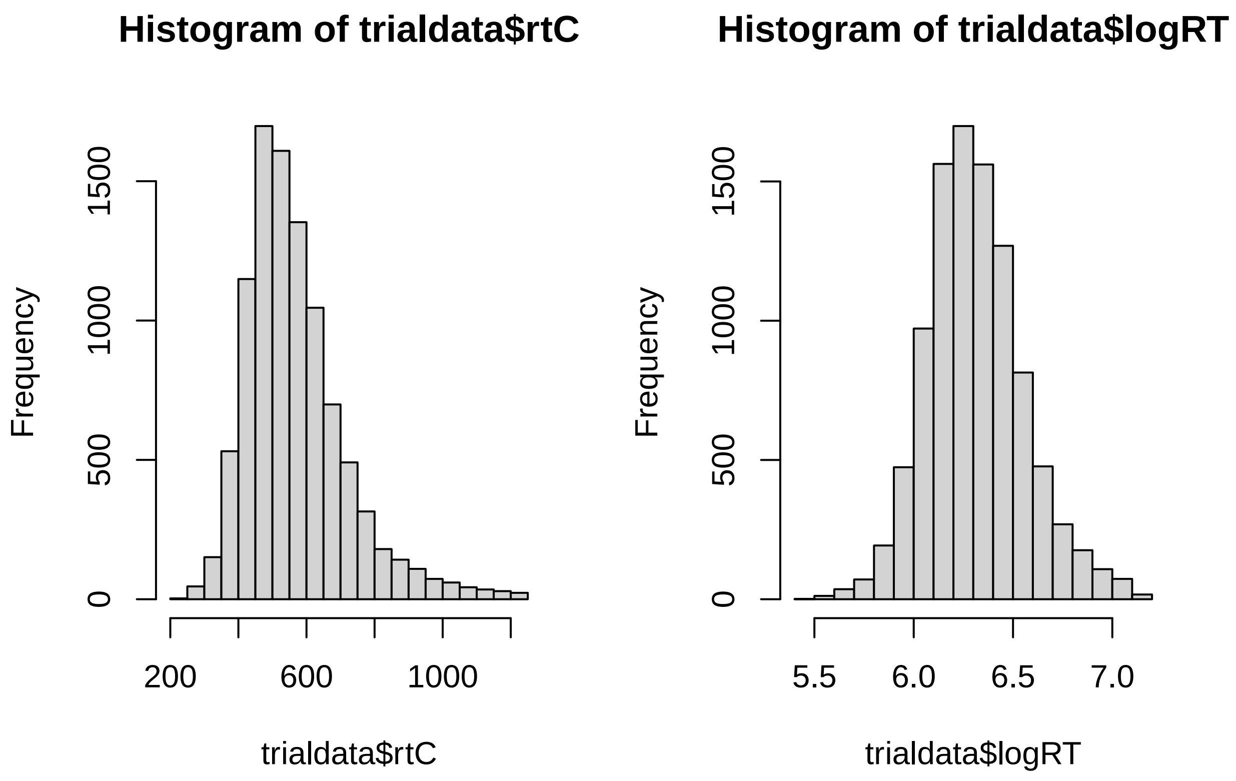
```
## # A tibble: 6 x 2
##   spelling SUBTLEX2
##   <chr>       <dbl>
## 1 beetje      59247
## 2 meisje      47345
## 3 liefje      10969
## 4 grapje      10226
## 5 feestje      8787
## 6 drankje      5265
```

```
# "beetje" and "meisje" both have egregiously extreme freq values;
# all the more reason to use Zipf-vals (See SUBTLEX-UK by Van Heuven et al., 2014)
```

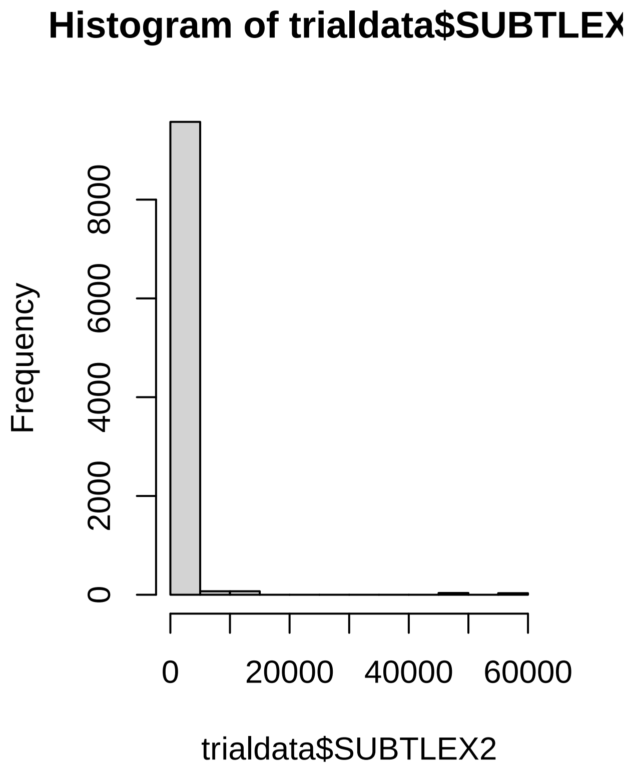
4. Finalise fine preprocessing

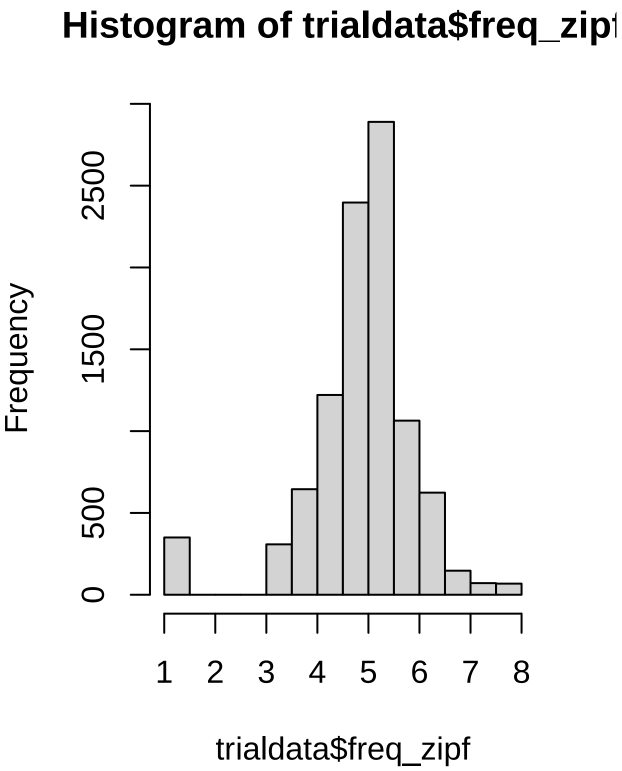
```
# Optional step: exclude the non-core diminutive items first,
# i.e. the ones with multiple ways to parse.
# NOTE: make sure to change contrasts below!
trialdata <- filter(trialdata, dim_type !="both")

# Log-transform reaction times using the base logarithm (see Winter 2019)
trialdata$logRT <- log(trialdata$rtC)
par(mfrow = c(1, 2))
hist(trialdata$rtC)
hist(trialdata$logRT)
```



```
# Zipf-values for frequency (see Van Heuven et al., 2014)
trialdata$freq_zipf <- log10(trialdata$SUBTLEX2+0.01)+3
par(mfrow = c(1, 2))
hist(trialdata$SUBTLEX2)
hist(trialdata$freq_zipf)
```





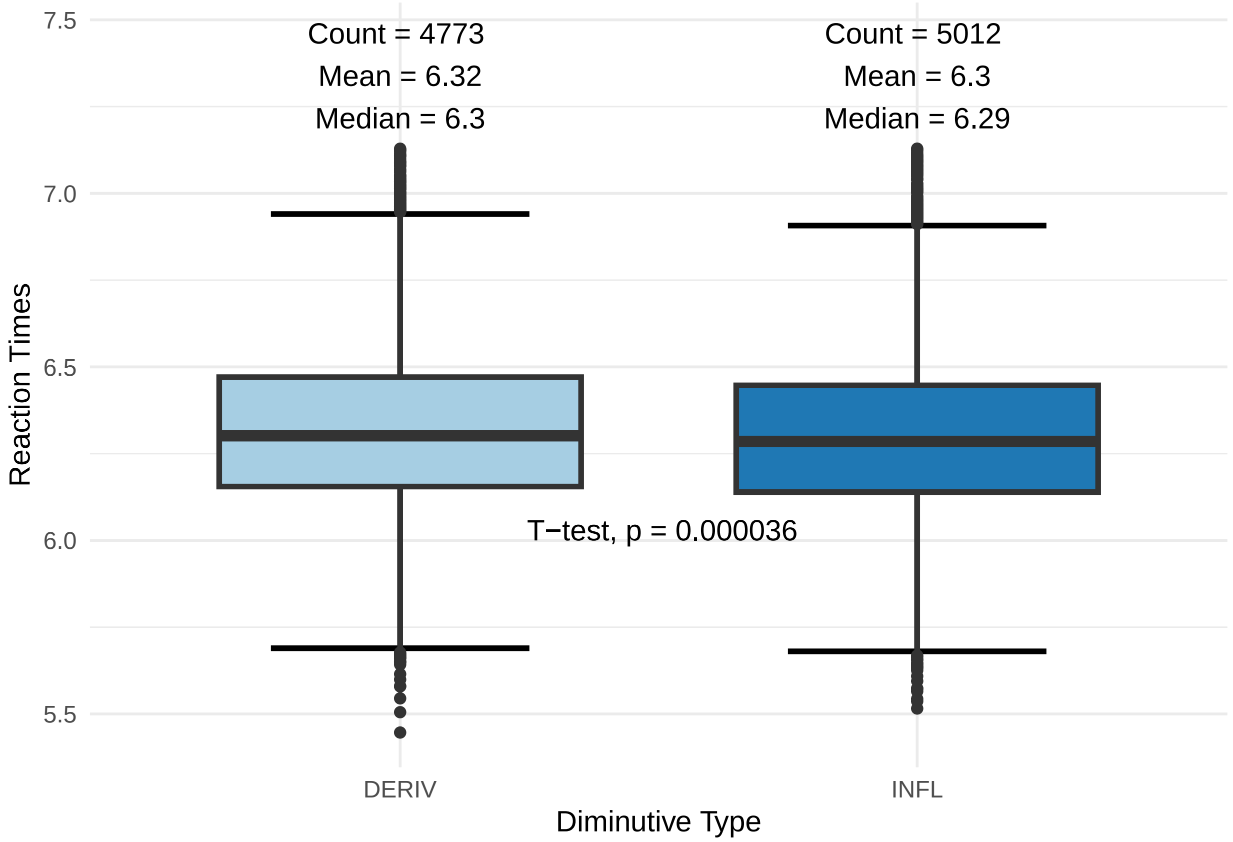
```
# Standardise all the relevant predictors for an easier comparison of effect size
trialdata <- mutate(trialdata,
                    zipf_z = scale(freq_zipf),
                    len_z = scale(Length),
                    nmorph_z = scale(nmorph),
                    old_z = scale(OLD20),
                    conc_z = scale(Concreteness),
                    aoa_z = scale(AoA),
                    # Word prevalence already standardised, just center
                    wp_z = scale(Word_prevalence1, scale = FALSE))
```

5. Descriptive Statistics

```
# Get means for the diminutives
print(trialdata %>% group_by(dim_type) %>%
  summarize(M = mean(rtC), SD = sd(rtC)))
```

```
## # A tibble: 2 x 3
##   dim_type     M    SD
##   <fct>    <dbl> <dbl>
## 1 deriv     574.  150.
## 2 infl      562.  147.
```

```r
# Make a plot with all the relevant statistics, throw in a t-test as well
trialdata %>% ggplot(aes(x = dim_type, y = logRT, fill = dim_type)) +
  stat_boxplot(geom ='errorbar', linewidth=1, width=.5) +
  geom_boxplot(lwd=1, width=.7, show.legend = FALSE) + theme_minimal() +
  labs(
    x = "Diminutive Type",
    y = "Reaction Times",
  ) +
  scale_x_discrete(labels=c('DERIV', 'INFL')) +
  stat_summary(fun.data = get_box_stats, geom = "text", hjust = 0.5, vjust = 0.9) +
  scale_fill_brewer(palette = "Paired") +
  stat_compare_means(method="t.test", label.y = 6.0, label.x = 1.35)
```



```r
# Save the plot for later
ggsave("../figures/dim_box.png", width = 8, height = 6)
```

6. Inferential Statistics

Here, we will finally handle some tests that will let us draw conclusions from the sample about the population at large. We start with assigning contrasts, then run a simple t-test and then move on to modelling.

6.1. Contrasts and t-tests

The next two code chunks handle the contrast coding; they are mutually exclusive depending on the decision to include observations with “both” as a value for dim_type. In order to switch between them, go to the chunk options and specify *eval=FALSE* to stop the script from running the code and *include=FALSE* to prevent the code from being included in the final knitted document.

The motivation for sum-coding between “deriv” and “infl” is taken from Winter 2020, where it is argued that sum-coding is better suited for mixed-effects models with interactions and random effects; additionally, sum-coding is argued to make the coefficients easier to interpret.

```
# Sum-coding for a purely deriv/infl dataset (see above)
contrasts(trialdata$dim_type) <- contr.sum(2)
contrasts(trialdata$dim_type)
```

```
##       [,1]
## deriv    1
## infl    -1
```

```
t.test(rtC ~ dim_type, data = trialdata)
```

```
##
##  Welch Two Sample t-test
##
## data:  rtC by dim_type
## t = 3.9364, df = 9731.1, p-value = 0.00008329
## alternative hypothesis: true difference in means between group deriv and group infl is not equal to
## 95 percent confidence interval:
##   5.935466 17.710467
## sample estimates:
## mean in group deriv  mean in group infl
##            574.2418            562.4188
```

```
t.test(logRT ~ dim_type, data = trialdata)
```

```
##
##  Welch Two Sample t-test
##
## data:  logRT by dim_type
## t = 4.1361, df = 9751.9, p-value = 0.00003562
## alternative hypothesis: true difference in means between group deriv and group infl is not equal to
## 95 percent confidence interval:
##  0.01073313 0.03007119
## sample estimates:
## mean in group deriv  mean in group infl
##            6.322096            6.301694
```

```
cohen.d(rtC ~ dim_type, data = trialdata)
```

```
##
## Cohen’s d
```

```
##
## d estimate: 0.07965868 (negligible)
## 95 percent confidence interval:
##      lower      upper
## 0.03999865 0.11931871
```

```
cohen.d(logRT ~ dim_type, data = trialdata)
```

```
##
## Cohen's d
##
## d estimate: 0.08366724 (negligible)
## 95 percent confidence interval:
##      lower      upper
## 0.04400559 0.12332889
```

The motivation for Helmert-coding between "deriv", "infl", and "both" is simple: for wordforms with either "deriv" or "infl", a single decomposition pipeline is proposed. Assuming structural differences, recognizing either is the same type of operation being performed in a different location in the structure. Compare with "both", where the decomposition apparatus presumably pursues both leads at once; therefore, this should be reflected in the coding as such:

- First, compare the differences between deriv and infl (either x or y)
- Then, compare their average RTs to both (x and y at the same time)

6.2. Modelling

The formula for the experimental model closely follows the formula reported in the DLP2 paper, with the addition of two predictors:

a. dim_type, a factor with values "deriv" and "infl" (possibly "both" for the expanded dataset)
b. nmorph, a numeric with values reflecting the (observable and theoretically motivated) morpheme count in the structure of each wordform: assuming full decomposition, every word is broken down to the most primitive units (= morphemes), so it logically follows that the more morphemes a wordform consists of, the more there is to break down and recompose, and the more time it should take to recognize a word.

Note the absence of two predictors reported for the DLP2 model, namely Nsyl and Length. The justification for both is the high degree of correlation between Length, Nsyl, nmoprh and OLD20 as explanatory variables that all in some ways have to deal with length. Nsyl, the number of syllables, was excluded first in favour of the more theoretically relevant nmoprh. Dropping length was a tough decision that was ultimately made after some stepwise modelling to establish which of the two factors, length or OLD20, would contribute more to explaining the variance in the dataset.

The final formula of the model is therefore this (see "mod_full" below):

RT ~ Diminutive Type*Frequency + Morpheme count + OLD20 ratings + Concreteness + Age of acquisition + Word prevalence + Varying intercepts by participant and by item

```
# Run the base model with only frequency and random intercepts
# Mention including the participant slopes and item intercepts as the
# motivation to satisfy the independence assumption (Winter 2020, Ch.14)
```

```
mod_base <- lmer(logRT ~ zipf_z +
                   (1|participant) +
                   (1|item), data = trialdata, REML = TRUE)
summary(mod_base) # Make sure the frequency findings aren't wildly off
```

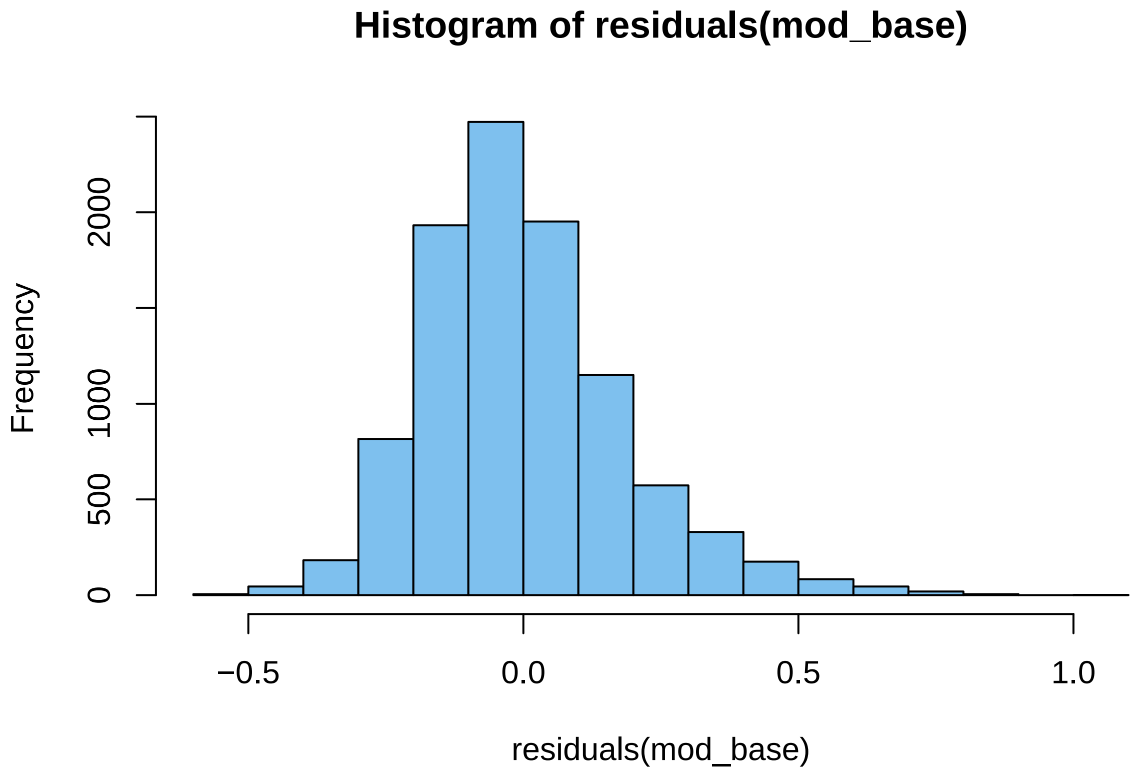
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: logRT ~ zipf_z + (1 | participant) + (1 | item)
##    Data: trialdata
##
## REML criterion at convergence: -4382.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0186 -0.6652 -0.1257  0.5158  5.3901
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  item        (Intercept) 0.002922 0.05405
##  participant (Intercept) 0.020267 0.14236
##  Residual                0.034708 0.18630
## Number of obs: 9785, groups:  item, 270; participant, 81
##
## Fixed effects:
##               Estimate Std. Error         df t value            Pr(>|t|)
## (Intercept)   6.314158   0.016268  86.797105   388.1 <0.0000000000000002 ***
## zipf_z       -0.051223   0.003711 271.628543   -13.8 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##        (Intr)
## zipf_z 0.006
```

```
r.squaredGLMM(mod_base)
```

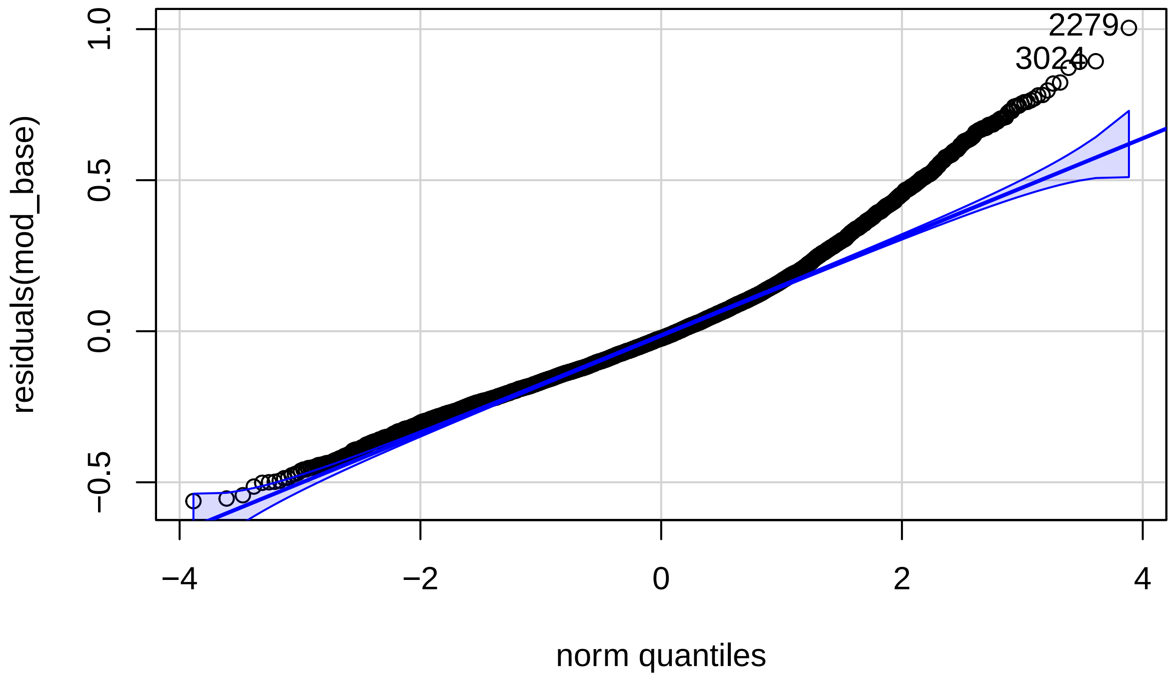
```
## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help page.
```

```
##             R2m      R2c
## [1,] 0.04335228 0.426511
```

```
hist(residuals(mod_base), col = 'skyblue2') # Plot 1, histogram
```

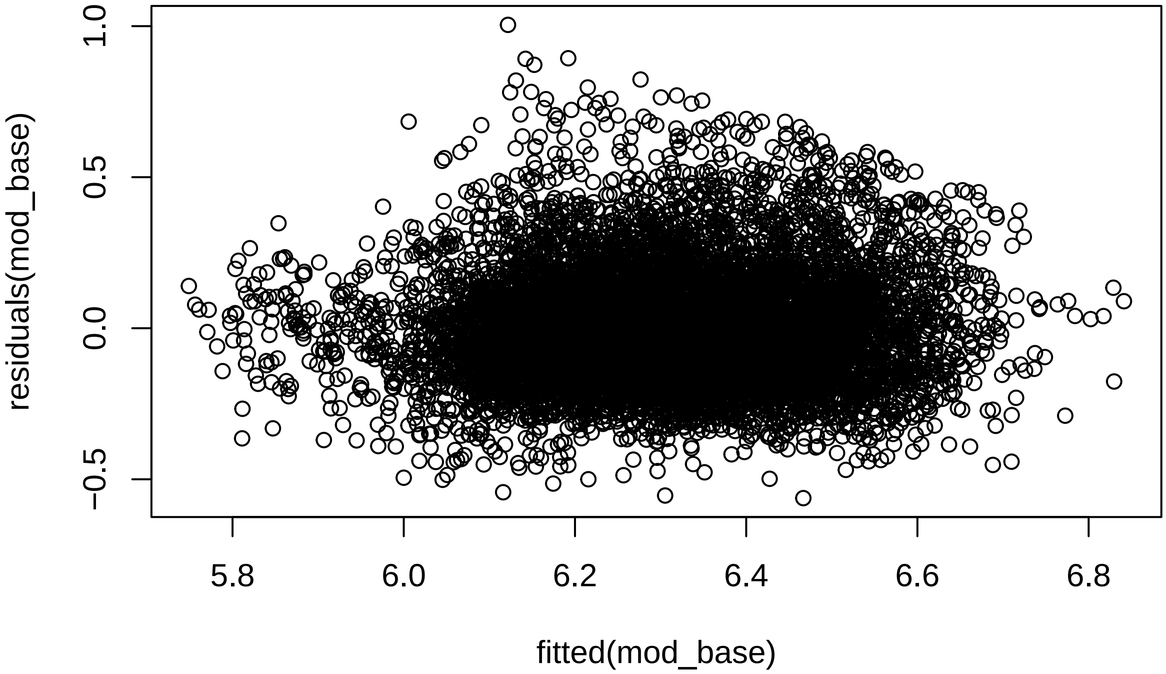


```
qqPlot(residuals(mod_base)) # Plot 2, Q-Q plot
```



```
## [1] 2279 3024
```

```
plot(fitted(mod_base), residuals(mod_base)) # Plot 3, residual plot
```



```
# Run the full model
mod_full <- lmer(logRT ~
                dim_type*zipf_z +
                nmorph_z +
                old_z +
                conc_z +
                aoa_z +
                wp_z +
                (1|participant) +
                (1|item), data = trialdata)
summary(mod_full)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: logRT ~ dim_type * zipf_z + nmorph_z + old_z + conc_z + aoa_z +
##     wp_z + (1 | participant) + (1 | item)
##    Data: trialdata
##
## REML criterion at convergence: -4415.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2041 -0.6637 -0.1245  0.5150  5.3553
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  item        (Intercept) 0.001819 0.04265
##  participant (Intercept) 0.020229 0.14223
##  Residual                0.034707 0.18630
## Number of obs: 9785, groups:  item, 270; participant, 81
##
## Fixed effects:
##                    Estimate Std. Error         df t value             Pr(>|t|)
## (Intercept)        6.313728   0.016128  84.231420 391.478 < 0.0000000000000002
## dim_type1         -0.003221   0.003563 254.798842  -0.904             0.366893
## zipf_z            -0.024480   0.004470 262.209924  -5.476         0.0000001017
## nmorph_z           0.002623   0.005119 253.955012   0.512             0.608858
## old_z              0.020231   0.005164 252.299591   3.918             0.000115
## conc_z             0.002022   0.003812 254.785560   0.531             0.596199
## aoa_z              0.016781   0.004342 252.592626   3.865             0.000141
## wp_z              -0.047086   0.008497 262.656860  -5.542         0.0000000728
## dim_type1:zipf_z   0.007063   0.003234 270.076237   2.184             0.029825
##
## (Intercept)      ***
## dim_type1
## zipf_z           ***
## nmorph_z
## old_z            ***
## conc_z
## aoa_z            ***
## wp_z             ***
## dim_type1:zipf_z *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##             (Intr) dm_ty1 zipf_z nmrph_ old_z  conc_z aoa_z  wp_z
## dim_type1    0.006
## zipf_z       0.000 -0.096
## nmorph_z     0.000 -0.125  0.221
## old_z        0.001  0.021  0.145 -0.665
## conc_z       0.001  0.208  0.305  0.040 -0.017
## aoa_z       -0.002 -0.153  0.232  0.006 -0.194  0.319
## wp_z         0.005  0.068 -0.455 -0.118 -0.163 -0.181  0.289
## dm_typ1:zp_  0.008  0.081 -0.112 -0.062  0.026  0.108  0.035  0.072
```

```
Anova(mod_full)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: logRT
##                   Chisq Df    Pr(>Chisq)
## dim_type         1.1761  1     0.2781613
## zipf_z          27.7154  1 0.00000014054 ***
## nmorph_z         0.2625  1     0.6084128
## old_z           15.3514  1 0.00008925600 ***
## conc_z           0.2815  1     0.5957365
## aoa_z           14.9372  1     0.0001111 ***
## wp_z            30.7096  1 0.00000002997 ***
## dim_type:zipf_z  4.7697  1     0.0289649 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

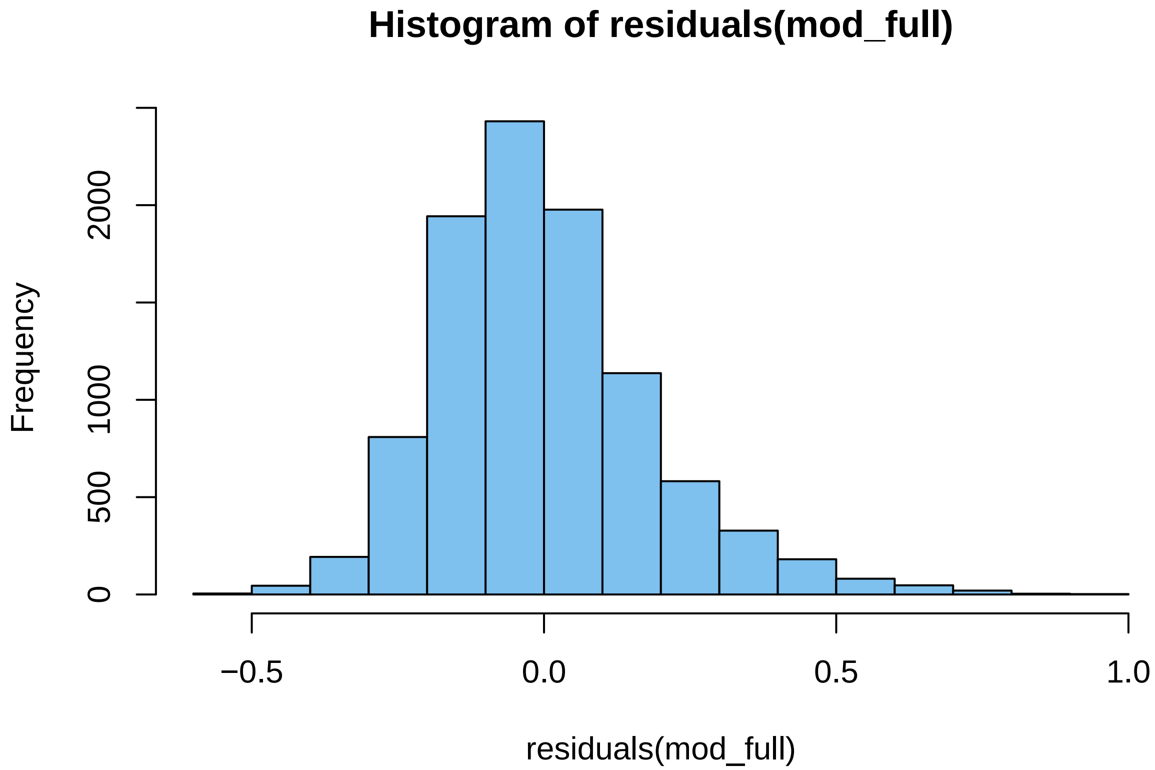
```
r.squaredGLMM(mod_full) # Fixed effects: 6.1%; Fixed+Random: 44%
```

```
##             R2m       R2c
## [1,] 0.06160806 0.4261597
```

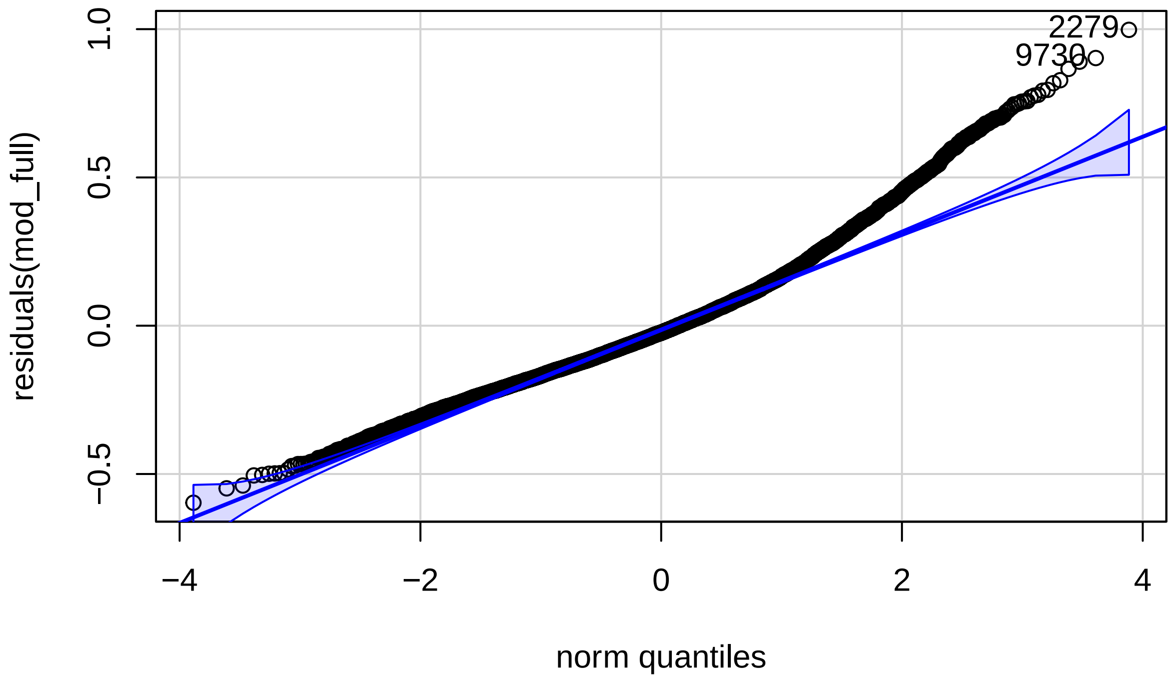
```
vif(mod_full) # Get variance inflation factors to check for collinearity
```

```
##        dim_type          zipf_z        nmorph_z           old_z          conc_z
##        1.201597        2.011141        2.542262        2.581457        1.408611
##           aoa_z            wp_z dim_type:zipf_z
##        1.850146        1.710761        1.043269
```

```
# Plots not ideal, but not as grossly off as before
hist(residuals(mod_full), col = 'skyblue2') # Plot 1, histogram
```

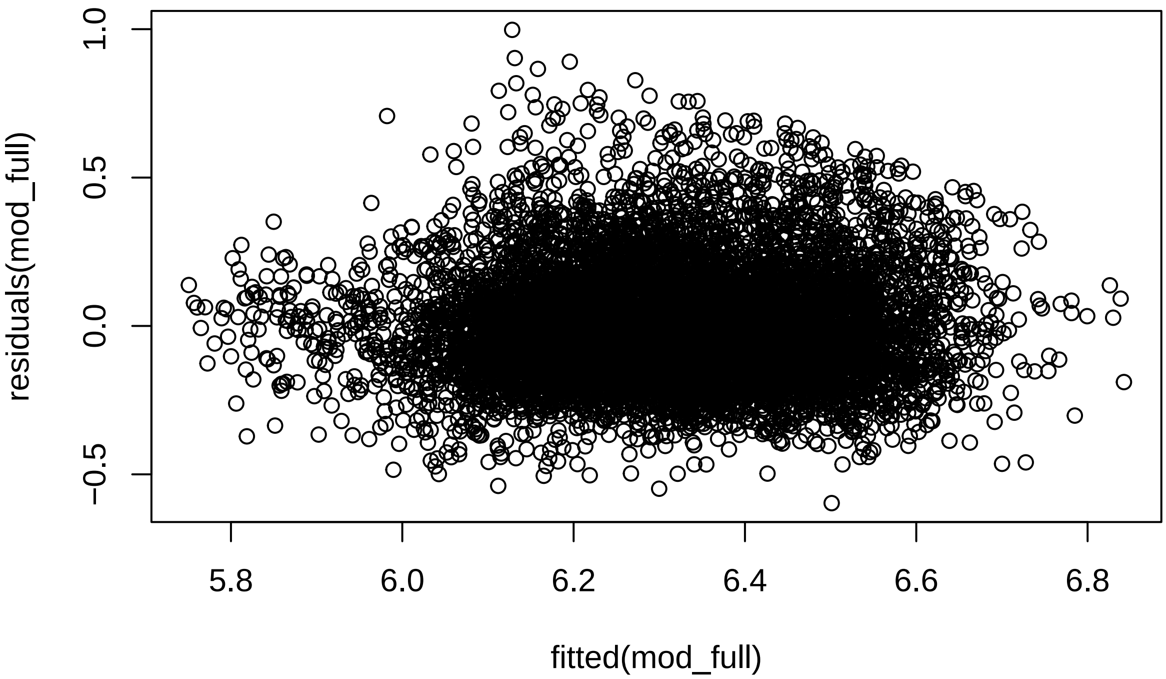


```
qqPlot(residuals(mod_full)) # Plot 2, Q-Q plot
```



```
## [1] 2279 9730
```

```
plot(fitted(mod_full), residuals(mod_full)) # Plot 3, residual plot
```



```
fixef(mod_full)
```

```
##      (Intercept)       dim_type1          zipf_z        nmorph_z
##      6.313727931    -0.003220900    -0.024480431     0.002622865
##            old_z          conc_z           aoa_z            wp_z
##      0.020231471     0.002022307     0.016780979    -0.047085511
## dim_type1:zipf_z
##      0.007062738
```

```
model_table <- as.data.frame(round(summary(mod_full)$coefficients, 4))
print(model_table)
```

```
##                  Estimate Std. Error       df  t value Pr(>|t|)
## (Intercept)        6.3137     0.0161  84.2314 391.4777   0.0000
## dim_type1         -0.0032     0.0036 254.7988  -0.9039   0.3669
## zipf_z            -0.0245     0.0045 262.2099  -5.4762   0.0000
## nmorph_z           0.0026     0.0051 253.9550   0.5123   0.6089
## old_z              0.0202     0.0052 252.2996   3.9181   0.0001
## conc_z             0.0020     0.0038 254.7856   0.5305   0.5962
## aoa_z              0.0168     0.0043 252.5926   3.8649   0.0001
## wp_z              -0.0471     0.0085 262.6569  -5.5416   0.0000
## dim_type1:zipf_z   0.0071     0.0032 270.0762   2.1840   0.0298
```