M21 RT (Continuous Predictors)

Joanna Morris

2025-06-22

Contents

Setup **1**

1 Load Files and Format Files **1**
1.1 Load Files . . . 1
1.2 Format Files . . . 1
1.3 Plot RT distributions . . . 2
1.4 Test for Skewness . . . 2
1.5 ANOVA Words . . . 5
1.5.1 Anova with Continuous `Log10BF` and Continuous `Log10FS` . . . 6
1.5.2 Effects . . . 6
1.5.3 Plots . . . 7
1.6 ANOVA Non-Words . . . 8
1.6.1 Anova with Continuous `Log10BF` and Categorical `Complexity` . . . 9
1.6.2 Effects . . . 9
1.6.3 Plots . . . 10

Setup

Load libraries

1. Set `ggplot2` parameters

1 Load Files and Format Files

1.1 Load Files

```
rt_1 <- read_csv("rt_data_chrt1.csv")
rt_2 <- read_csv("rt_data_chrt2.csv")
frq_w <- read_csv("frq_cw.csv")
frq_nw <- read_csv("frq_nw.csv")
dmg <- read_csv("demo_lang_vsl_pca.csv")
```

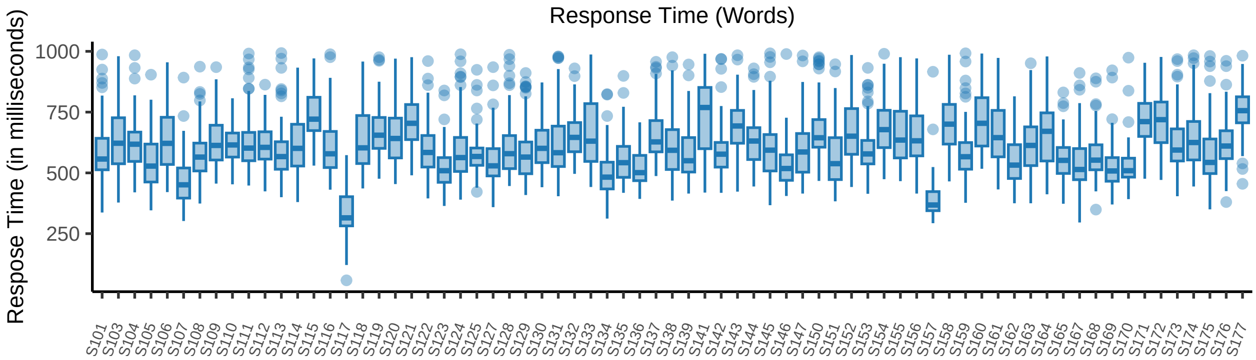
1.2 Format Files

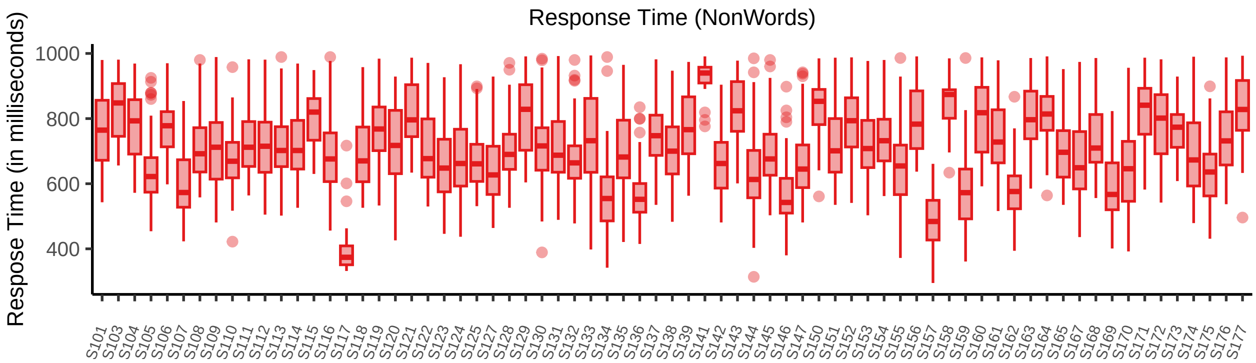
```
# Concatenate datasets
rt <- bind_rows(Hampshire = rt_1,
                Providence =rt_2,
                .id = "location")
rt_dmg<- right_join(dmg, rt, join_by(SubjID == subject_nr))  # Join Participant Demographic and Lang Data
rt_dmg <- rt_dmg |> mutate(target = tolower(target))
rt_dmg_cor <- rt_dmg |> filter(correct == 1)

# Divide into Experimental and Filler Items
rt_fill <- rt_dmg_cor |> filter(str_detect(targ_type, "^FILL"))
rt_exp <- rt_dmg_cor |> filter(!str_detect(targ_type, "^FILL"))

# Define Factors and Conditions
rt_exp_cln <- rt_exp |>
```

```
  separate(targ_type, into = c("trial_type", "family_size", "complexity"), sep = "_",
           remove = TRUE,
           extra = "drop",
           fill = "right")

# Divide into Words and Nonwords
rt_words <- rt_exp_cln |> filter(trial_type == "CW") |> select(- complexity)
rt_nwords <- rt_exp_cln |> filter(trial_type == "NW")

# Join Stimulus Frequency Data
rt_words_frq <- left_join(rt_words, frq_w, join_by(target))|>
  select(-cond_trig.y, -word_trig.y) |>
  rename(cond_trig = cond_trig.x, word_trig = word_trig.x)  # remove duplicate columns

rt_nwords_frq <-  left_join(rt_nwords, frq_nw, join_by(target==word))



# Create a median-split factor for base frequency
rt_words_frq$BF_MedianSplit <- ifelse(
  rt_words_frq$Log10BF <= median(rt_words_frq$Log10BF, na.rm = TRUE),
  "Low", "High")

rt_words_frq$BF_Split <- factor(rt_words_frq$BF_Split)
rt_words_frq$FS_Split <- factor(rt_words_frq$FS_Split)

rt_nwords_frq$BF_Split <- factor(rt_nwords_frq$BF_Split)
rt_nwords_frq$FS_Split <- factor(rt_nwords_frq$FS_Split)

# Divide into cohorts
rt_words_1 <- rt_words_frq |> filter(location == "Hampshire") |> select(- location)
rt_words_2 <- rt_words_frq |> filter(location == "Providence") |> select(- location)

rt_nwords_1 <- rt_nwords_frq |> filter(location == "Hampshire") |> select(- location)
rt_nwords_2 <- rt_nwords_frq |> filter(location == "Providence") |> select(- location)
# str(rt_words_1)
```

1.3 Plot RT distributions

...





1.4 Test for Skewness

```
Response Time
```

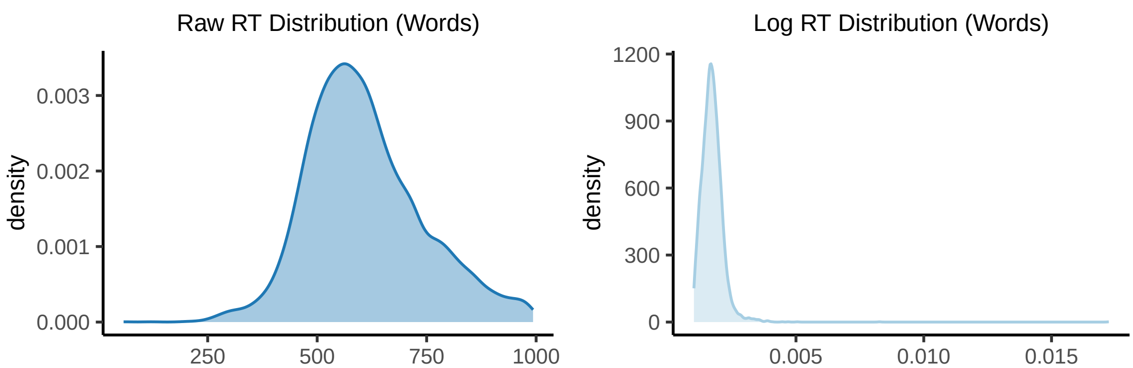
```r
# Words Skewness values
skewness(rt_words_1$response_time, na.rm = TRUE)
```

```
|| [1] 0.4868724
```

```r
skewness(rt_words_1$LogRT, na.rm = TRUE)
```

```
|| [1] -0.4362045
```

```r
# Words Raw RT Distribution
p1 <- rt_words_1 |> ggplot(aes(x = response_time)) +
  geom_density(colour = "#1F78B4", fill = "#1F78B4", alpha = .4) +
  labs(title = "Raw RT Distribution (Words)") +
  theme( plot.title = element_text(size = 9, hjust = .5),
         legend.title = element_blank(),
         axis.title.x = element_blank(),
         axis.text.x = element_text(size = 8))

# Words Log RT Distribution
p2 <- rt_words_1 |> ggplot(aes(x = InvRT)) +
  geom_density(colour = "#A6CEE3", fill = "#A6CEE3", alpha = .4)  +
  labs(title = "Log RT Distribution (Words)") +
  theme( plot.title = element_text(size = 9, hjust = .5),
         legend.title = element_blank(),
         axis.title.x = element_blank(),
         axis.text.x = element_text(size = 8))
plot_grid(p1, p2, ncol = 2)
```



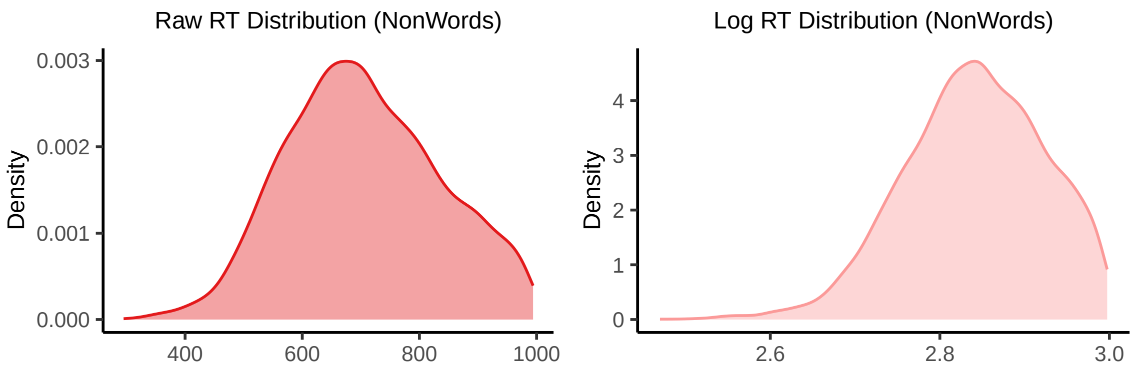

```r
# NONWORDS
# Skewness values
skewness(rt_nwords_1$response_time, na.rm = TRUE)
```

```
|| [1] 0.1000102
```

```r
skewness(rt_nwords_1$LogRT, na.rm = TRUE)
```

```
|| [1] -0.3817821
```

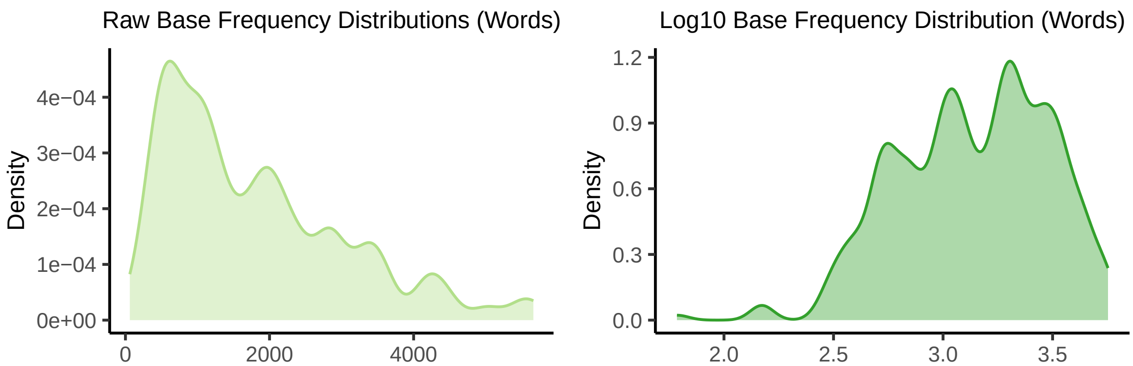
```r
# Raw RT
p1 <- rt_nwords_1 |> ggplot(aes(x = response_time)) +
  geom_density(colour = "#E31A1C", fill = "#E31A1C", alpha = .4) +
  labs(y = "Density",title = "Raw RT Distribution (NonWords)") +
  theme( plot.title = element_text(size = 9, hjust = .5),
         legend.title = element_blank(),
         axis.title.x = element_blank(),
         axis.text.x = element_text(size = 8))
# LogRT
p2 <- rt_nwords_1 |> ggplot(aes(x = LogRT)) +
  geom_density(colour = "#FB9A99", fill = "#FB9A99", alpha = .4)  +
  labs(y = "Density", title = "Log RT Distribution (NonWords)") +
  theme( plot.title = element_text(size = 9, hjust = .5),
         legend.title = element_blank(),
         axis.title.x = element_blank(),
         axis.text.x = element_text(size = 8))
plot_grid(p1, p2, ncol = 2)
```



Base Frequency

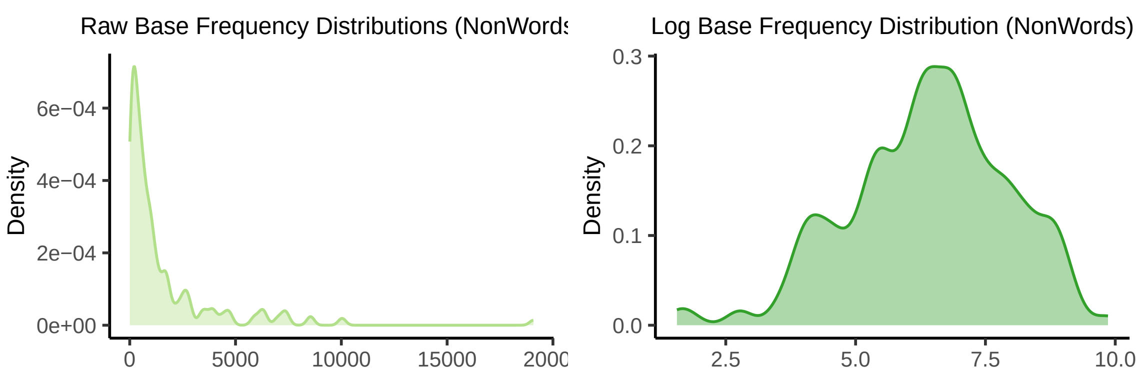
```
|| [1] 0.9870676
```

```
|| [1] -0.4166518
```




```
|| [1] 3.404106
```

```
|| [1] -0.3931931
```



Family Size

```r
# Words
# Skewness values
rt_words_1 <- rt_words_1 |> mutate(Log10FS = log10(FS))
skewness(rt_words_1$FS, na.rm = TRUE)
```

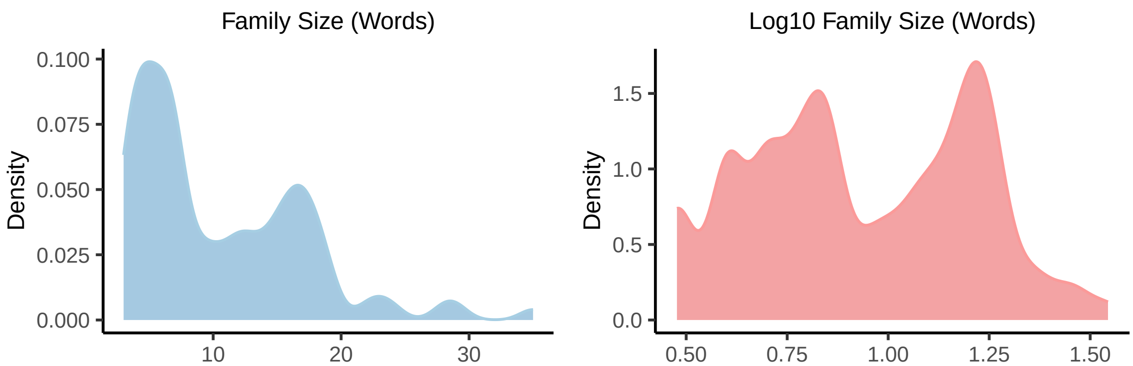
```
|| [1] 1.101473
```

```r
skewness(rt_words_1$Log10FS, na.rm = TRUE)
```

```
|| [1] 0.05781409
```

```r
# Raw FS
p1 <- ggplot(rt_words_1, aes(x = FS)) +
  geom_density(colour = "#A6CEE3", fill = "#1F78B4", alpha = .4) +
  labs(title = "Family Size (Words)", y = "Density") +
  theme( plot.title = element_text(size = 9, hjust = .5),
         legend.title = element_blank(),
         axis.title.x = element_blank(),
         axis.text.x = element_text(size = 8))
```

```
# Log10 FS
p2 <- ggplot(rt_words_1, aes(x = Log10FS)) +
  geom_density(colour = "#FB9A99", fill = "#E31A1C", alpha = .4) +
  labs(title = "Log10 Family Size (Words)", y = "Density") +
  theme( plot.title = element_text(size = 9, hjust = .5),
         legend.title = element_blank(),
         axis.title.x = element_blank(),
         axis.text.x = element_text(size = 8))
plot_grid(p1, p2, ncol = 2)
```



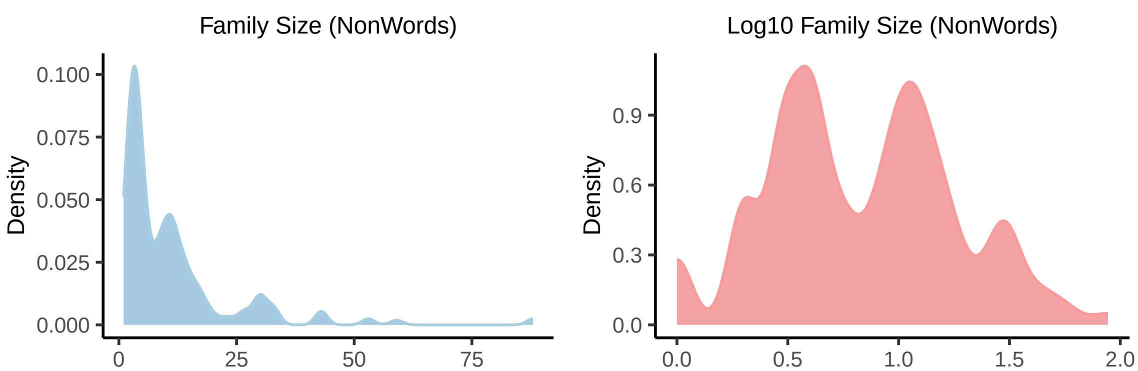

```
# NonWords
# Skewness values
rt_nwords_1 <- rt_nwords_1 |> mutate(Log10FS = log10(FS))
skewness(rt_nwords_1$FS, na.rm = TRUE)
```

```
|| [1] 2.909422
```

```
skewness(rt_nwords_1$Log10FS, na.rm = TRUE)
```

```
|| [1] 0.1536901
```

```
# Raw FS
p1 <- ggplot(rt_nwords_1, aes(x = FS)) +
  geom_density(colour = "#A6CEE3", fill = "#1F78B4", alpha = .4) +
  labs(title = "Family Size (NonWords)", y = "Density") +
  theme( plot.title = element_text(size = 9, hjust = .5),
         legend.title = element_blank(),
         axis.title.x = element_blank(),
         axis.text.x = element_text(size = 8))
# Log10 FS
p2 <- ggplot(rt_nwords_1, aes(x = Log10FS)) +
  geom_density(colour = "#FB9A99", fill = "#E31A1C", alpha = .4) +
  labs(title = "Log10 Family Size (NonWords)", y = "Density") +
  theme( plot.title = element_text(size = 9, hjust = .5),
         legend.title = element_blank(),
         axis.title.x = element_blank(),
         axis.text.x = element_text(size = 8))
plot_grid(p1, p2, ncol = 2)
```



1.5 ANOVA Words

Use `complete.cases()` to find which rows had missing data in the model-relevant variables:

Standardize the predictors

```r
rt_words_1_cmpl$Log10BF_std <- as.numeric(scale(rt_words_1_cmpl$Log10BF, center = TRUE, scale = TRUE))
rt_words_1_cmpl$FS_std <- as.numeric(scale(rt_words_1_cmpl$FS, center = TRUE, scale = TRUE))
rt_words_1_cmpl$Log10WF_std <- as.numeric(scale(rt_words_1_cmpl$Log10WF, center = TRUE, scale = TRUE))
rt_words_1_cmpl$Log10FS_std <- as.numeric(scale(rt_words_1_cmpl$Log10FS, center = TRUE, scale = TRUE))
rt_words_1_cmpl$Dim.2_std <- as.numeric(scale(rt_words_1_cmpl$Dim.2, center = TRUE, scale = TRUE))
```

1.5.1 Anova with Continuous Log10BF and Continuous Log10FS

```r
# Anova with Continuous Log10BF AND Continous FS
anova_model_1 <- mixed(
  response_time ~ Log10BF_std * Log10FS_std * lang_type_ortho + (1  | SubjID),
  data = rt_words_1_cmpl,
  method = "KR"
)
anova_model_1
```

```
|| Mixed Model Anova Table (Type 3 tests, KR-method)
||
|| Model: response_time ~ Log10BF_std * Log10FS_std * lang_type_ortho +
|| Model:     (1 | SubjID)
|| Data: rt_words_1_cmpl
||                                    Effect         df         F p.value
|| 1                             Log10BF_std 1, 5792.77 46.75 ***   <.001
|| 2                             Log10FS_std 1, 5792.45 29.26 ***   <.001
|| 3                         lang_type_ortho   1, 64.04    3.12 +    .082
|| 4                 Log10BF_std:Log10FS_std 1, 5792.56      0.03    .859
|| 5             Log10BF_std:lang_type_ortho 1, 5792.77      2.24    .135
|| 6             Log10FS_std:lang_type_ortho 1, 5792.45      0.39    .531
|| 7 Log10BF_std:Log10FS_std:lang_type_ortho 1, 5792.56      0.30    .587
|| ---
|| Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```r
summary(anova_model_1)
```

```
|| Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
|| Formula: response_time ~ Log10BF_std * Log10FS_std * lang_type_ortho +      (1 | SubjID)
||    Data: data
||
|| REML criterion at convergence: 71992.1
||
|| Scaled residuals:
||     Min      1Q  Median      3Q     Max
|| -3.0168 -0.6888 -0.1441  0.5144  4.7125
||
|| Random effects:
||  Groups   Name        Variance Std.Dev.
||  SubjID   (Intercept)  5304     72.83
||  Residual             12135    110.16
|| Number of obs: 5864, groups:  SubjID, 66
||
|| Fixed effects:
||                                           Estimate Std. Error        df t value Pr(>|t|)
|| (Intercept)                               607.1261     9.1885   63.8979  66.075  < 2e-16 ***
|| Log10BF_std                               -10.5521     1.5434 5792.6233  -6.837 8.91e-12 ***
|| Log10FS_std                                -7.9926     1.4776 5792.3073  -5.409 6.59e-08 ***
|| lang_type_ortho1                          -16.2208     9.1885   63.8979  -1.765   0.0823 .
|| Log10BF_std:Log10FS_std                     0.2821     1.5836 5792.4183   0.178   0.8586
|| Log10BF_std:lang_type_ortho1               -2.3074     1.5434 5792.6233  -1.495   0.1350
|| Log10FS_std:lang_type_ortho1                0.9269     1.4776 5792.3073   0.627   0.5305
|| Log10BF_std:Log10FS_std:lang_type_ortho1   -0.8608     1.5836 5792.4183  -0.544   0.5868
|| ---
|| Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
||
|| Correlation of Fixed Effects:
||                (Intr) Lg10BF_ Lg10FS_ lng__1 Lg10BF_:L10FS_ L10BF_:_ L10FS_:
|| Log10BF_std    -0.005
|| Log10FS_std     0.003 -0.134
|| lng_typ_rt1     0.152 -0.001   0.002
|| Lg10BF_:L10FS_ -0.018  0.301  -0.122  -0.003
|| Lg10BF_:__1    -0.001  0.181  -0.022  -0.005  0.061
|| Lg10FS_:__1     0.002 -0.022   0.161   0.003 -0.024         -0.134
|| L10BF_:L10FS_: -0.003  0.061  -0.024  -0.018  0.182          0.301   -0.122
```

1.5.2 Effects

Effect	df	F	p.value
Log10BF_std	1, 5792.77	46.75 ***	<.001
Log10FS_std	1, 5792.45	29.26 ***	<.001

Effect	df	F	p.value
lang_type_ortho	1, 64.04	3.12 +	.082
Log10BF_std:lang_type_ortho	1, 5792.77	2.24	.135

Main Effect of Family Size and Base Frequency

```
emm_options(pbkrtest.limit = 5864)
emtrends(anova_model_1, ~1, var = "Log10FS_std")
```

```
||  1       Log10FS_std.trend   SE   df lower.CL upper.CL
||  overall             -7.99 1.48 5792    -10.9     -5.1
||
|| Results are averaged over the levels of: lang_type_ortho
|| Degrees-of-freedom method: kenward-roger
|| Confidence level used: 0.95
```

```
emtrends(anova_model_1, ~1, var = "Log10BF_std")
```

```
||  1       Log10BF_std.trend   SE   df lower.CL upper.CL
||  overall             -10.6 1.54 5793    -13.6    -7.53
||
|| Results are averaged over the levels of: lang_type_ortho
|| Degrees-of-freedom method: kenward-roger
|| Confidence level used: 0.95
```

Base Frequency by Orthographic Sensitivity Interaction

```
# estimate simple slopes of base frequency by group:
emtrends(anova_model_1, ~ lang_type_ortho, var = "Log10BF_std")
```

```
||  lang_type_ortho   Log10BF_std.trend   SE   df lower.CL upper.CL
||  High Orthographic            -12.86 2.37 5793    -17.5    -8.21
||  Low Orthographic              -8.24 1.98 5793    -12.1    -4.37
||
|| Degrees-of-freedom method: kenward-roger
|| Confidence level used: 0.95
```

```
emtrends(anova_model_1, pairwise ~ lang_type_ortho, var = "Log10BF_std")
```

```
|| $emtrends
||  lang_type_ortho   Log10BF_std.trend   SE   df lower.CL upper.CL
||  High Orthographic            -12.86 2.37 5793    -17.5    -8.21
||  Low Orthographic              -8.24 1.98 5793    -12.1    -4.37
||
|| Degrees-of-freedom method: kenward-roger
|| Confidence level used: 0.95
||
|| $contrasts
||  contrast                              estimate   SE   df t.ratio p.value
||  High Orthographic - Low Orthographic     -4.61 3.09 5793  -1.495  0.1350
||
|| Degrees-of-freedom method: kenward-roger
```

```
# Estimate marginal means of RT at the mean of both predictors
emm <- emmeans(anova_model_1, ~ lang_type_ortho, at = list(Log10BF_std = 0, Log10FS_std = 0))
emm_df <- as.data.frame(emm)

print(emm_df)
```

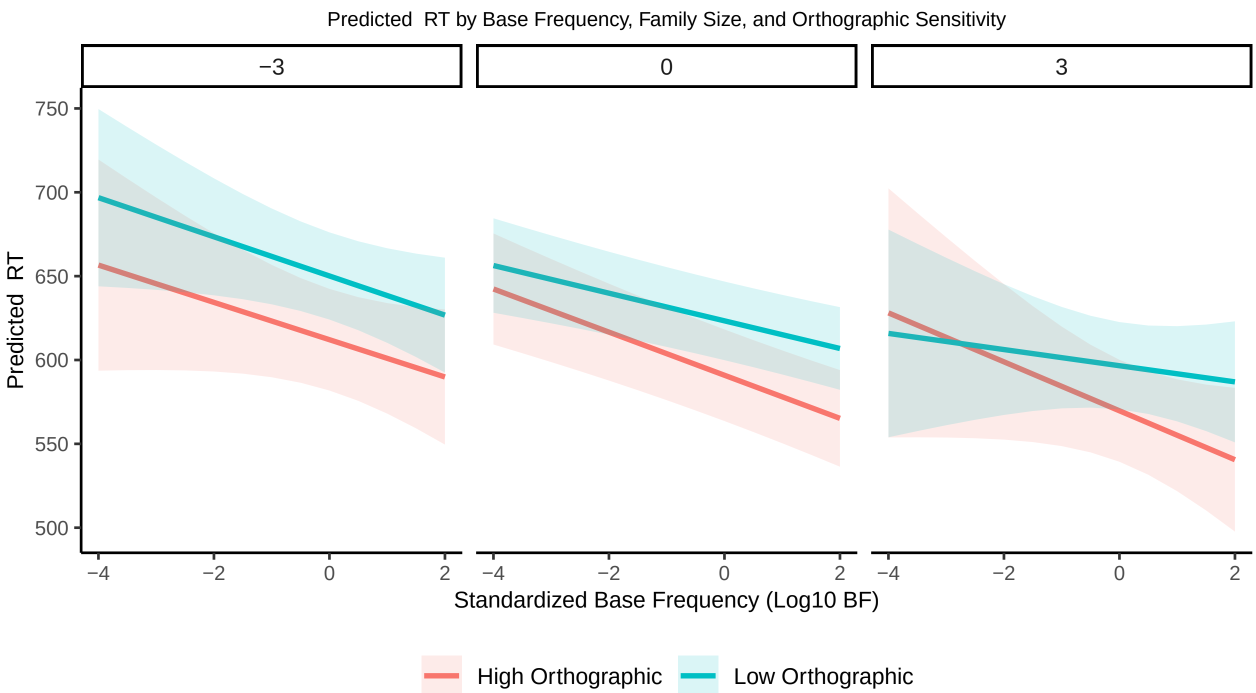
```
||  lang_type_ortho     emmean       SE    df lower.CL upper.CL
||  High Orthographic 590.9053 13.94459 64.05 563.0482 618.7624
||  Low Orthographic  623.3469 11.96926 64.03 599.4358 647.2580
||
|| Degrees-of-freedom method: kenward-roger
|| Confidence level used: 0.95
```

1.5.3 Plots

Family Size x Base Frequency x Orthographic Sensitivity

```
# re-run anova with lmer to use `ggeffects`
anova_model_lmer <- lmer(response_time ~ Log10BF_std * Log10FS_std * lang_type_ortho + (1 | SubjID), data = rt_words_1_cmpl)
# Generate predicted values
preds <- ggpredict(anova_model_lmer, terms = c("Log10BF_std", "lang_type_ortho", "Log10FS_std [-3,0,3]"))

# Plot
ggplot(preds, aes(x = x, y = predicted, color = group, fill = group)) +
  geom_line(linewidth = 1) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = 0.15, color = NA) +
  facet_wrap(~facet, labeller = label_value) +
  labs(x = "Standardized Base Frequency (Log10 BF)",
```

```
       y = "Predicted  RT",
       color = "Family Size (Log10 FS)",
       fill = "Family Size (Log10 FS)",
       title = "Predicted  RT by Base Frequency, Family Size, and Orthographic Sensitivity")+
  theme( plot.title = element_text(size = 8, hjust = .5),
         legend.title = element_blank(),
         axis.text.x = element_text(size = 8))
```



1.6 ANOVA Non-Words

Use `complete.cases()` to find which rows had missing data in the model-relevant variables:

Standardize the predictors

```
rt_nwords_1_cmpl$LogBF_std <- as.numeric(scale(rt_nwords_1_cmpl$LogBF, center = TRUE, scale = TRUE))
rt_nwords_1_cmpl$FS_std <- as.numeric(scale(rt_nwords_1_cmpl$FS, center = TRUE, scale = TRUE))
rt_nwords_1_cmpl$Log10FS_std <- as.numeric(scale(rt_nwords_1_cmpl$Log10FS, center = TRUE, scale = TRUE))
rt_nwords_1_cmpl$BF_std <- as.numeric(scale(rt_nwords_1_cmpl$BF, center = TRUE, scale = TRUE))
rt_nwords_1_cmpl$Dim.2_std <- as.numeric(scale(rt_nwords_1_cmpl$Dim.2, center = TRUE, scale = TRUE))
rt_nwords_1_cmpl <- rt_nwords_1_cmpl |>  select(-complexity.x)
rt_nwords_1_cmpl <- rename(rt_nwords_1_cmpl, complexity = complexity.y)
```

Test Correlation between `Base Frequency` and `Complexity`

```
t.test(LogBF ~ complexity, data = rt_nwords_1_cmpl)
```

```
||
||  Welch Two Sample t-test
||
|| data:  LogBF by complexity
|| t = -1.0101, df = 4526.2, p-value = 0.3125
|| alternative hypothesis: true difference in means between group Complex and group Simple is not equal to 0
|| 95 percent confidence interval:
||  -0.13262805  0.04242954
|| sample estimates:
|| mean in group Complex  mean in group Simple
||              6.377598              6.422697
```

```
# Create a contingency table
table_data <- table(rt_nwords_1_cmpl$complexity, rt_nwords_1_cmpl$BF_Split)

# Run the chi-square test
chisq.test(table_data)
```

```
||
||  Pearson's Chi-squared test with Yates' continuity correction
||
|| data:  table_data
```

```
|| X-squared = 3.9314, df = 1, p-value = 0.04739
```

1.6.1 Anova with Continuous Log10BF and Categorical Complexity

```
anova_model_2 <- mixed(
  response_time ~ complexity * Log10FS_std * lang_type_ortho + (1  | SubjID),
  data = rt_nwords_1_cmpl,
  method = "KR"
)
anova_model_2
```

```
|| Mixed Model Anova Table (Type 3 tests, KR-method)
||
|| Model: response_time ~ complexity * Log10FS_std * lang_type_ortho +
|| Model:     (1 | SubjID)
|| Data: rt_nwords_1_cmpl
||                                Effect         df          F p.value
|| 1                          complexity 1, 4600.75 102.69 ***   <.001
|| 2                         Log10FS_std 1, 4599.87       2.26    .133
|| 3                     lang_type_ortho   1, 63.99     2.81 +    .099
|| 4              complexity:Log10FS_std 1, 4600.27     4.11 *    .043
|| 5          complexity:lang_type_ortho 1, 4600.75       0.59    .443
|| 6         Log10FS_std:lang_type_ortho 1, 4599.87       0.56    .454
|| 7 complexity:Log10FS_std:lang_type_ortho 1, 4600.27       0.00    .958
|| ---
|| Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```
summary(anova_model_2)
```

```
|| Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
|| Formula: response_time ~ complexity * Log10FS_std * lang_type_ortho +      (1 | SubjID)
||    Data: data
||
|| REML criterion at convergence: 56997.7
||
|| Scaled residuals:
||     Min      1Q  Median      3Q     Max
|| -3.1146 -0.7108 -0.0823  0.6394  4.1145
||
|| Random effects:
||  Groups   Name        Variance Std.Dev.
||  SubjID   (Intercept)  7364     85.81
||  Residual             11145    105.57
|| Number of obs: 4671, groups:  SubjID, 66
||
|| Fixed effects:
||                                          Estimate Std. Error         df t value Pr(>|t|)
|| (Intercept)                             711.37606   10.81603   62.49198  65.771   <2e-16 ***
|| complexity1                              15.98864    1.57778 4599.25340  10.134   <2e-16 ***
|| Log10FS_std                               2.38062    1.58399 4598.35522   1.503   0.1329
|| lang_type_ortho1                        -18.12880   10.81603   62.49198  -1.676   0.0987 .
|| complexity1:Log10FS_std                   3.21492    1.58505 4598.76384   2.028   0.0426 *
|| complexity1:lang_type_ortho1             -1.21116    1.57778 4599.25340  -0.768   0.4427
|| Log10FS_std:lang_type_ortho1             -1.18496    1.58399 4598.35522  -0.748   0.4544
|| complexity1:Log10FS_std:lang_type_ortho1  -0.08256    1.58505 4598.76384  -0.052   0.9585
|| ---
|| Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
||
|| Correlation of Fixed Effects:
||             (Intr) cmplx1 Lg10FS_ lng__1 cm1:L10FS_ c1:__1 L10FS_:
|| complexity1  0.023
|| Log10FS_std  0.002  0.029
|| lng_typ_rt1  0.150  0.001 -0.002
|| cmp1:L10FS_  0.005  0.008  0.180   0.001
|| cmplxt1:__1  0.001  0.115  0.004   0.023 -0.006
|| Lg10FS_:__1 -0.002  0.004  0.117   0.002  0.034      0.029
|| c1:L10FS_:_  0.001 -0.006  0.034   0.005  0.117      0.008  0.180
```

1.6.2 Effects

Effect	df	F	p.value
complexity	1, 4600.75	102.69 ***	<.001
complexity:Log10FS_std	1, 4600.27	4.11 *	.043

```
pairs <- emmeans(anova_model_2, pairwise ~ complexity, adjust = "bonferroni", pbkrtest.limit = 6480)
pairs_df <- as.data.frame(pairs$contrasts)
```

```r
cohensd <- as.data.frame(cohens_d(response_time ~ complexity, data = rt_nwords_1_cmpl))
(complexity_contrasts_df <- bind_cols(pairs_df,cohensd))
```

```
||  contrast         estimate       SE      df t.ratio p.value Cohens_d   CI    CI_low   CI_high
||  Complex - Simple 31.97728 3.155579 4600.75  10.134  <.0001  0.21345 0.95 0.1552988 0.2715784
||
|| Results are averaged over the levels of: lang_type_ortho
|| Degrees-of-freedom method: kenward-roger
```

```r
(complexity_means <- as.data.frame(pairs$emmeans))
```

```
||  complexity   emmean       SE    df lower.CL upper.CL
||  Complex    727.3647 10.96642 67.62 705.4793 749.2501
||  Simple     695.3874 10.89456 65.87 673.6349 717.1399
||
|| Results are averaged over the levels of: lang_type_ortho
|| Degrees-of-freedom method: kenward-roger
|| Confidence level used: 0.95
```

```r
# Get estimated marginal means for each level of complexity
em_means_complexity <- emmeans(anova_model_2, ~ complexity)
contrast(em_means_complexity, method = "pairwise")
```

```
||  contrast         estimate   SE   df t.ratio p.value
||  Complex - Simple       32 3.16 4601  10.134  <.0001
||
|| Results are averaged over the levels of: lang_type_ortho
|| Degrees-of-freedom method: kenward-roger
```

```r
summary(em_means_complexity)
```

```
||  complexity emmean   SE   df lower.CL upper.CL
||  Complex       727 11.0 67.6      705      749
||  Simple        695 10.9 65.9      674      717
||
|| Results are averaged over the levels of: lang_type_ortho
|| Degrees-of-freedom method: kenward-roger
|| Confidence level used: 0.95
```

```r
# Estimate the effect of complexity at low, mean, and high FS
em_complexity <- emmeans(anova_model_2, ~ complexity | Log10FS_std, at = list(Log10FS_std = c(-1, 0, 1)))
summary(em_complexity)
```

```
|| Log10FS_std = -1:
||  complexity emmean   SE   df lower.CL upper.CL
||  Complex       722 11.2 74.0      699      744
||  Simple        696 11.1 70.8      674      718
||
|| Log10FS_std =  0:
||  complexity emmean   SE   df lower.CL upper.CL
||  Complex       727 11.0 67.6      705      749
||  Simple        695 10.9 65.9      674      717
||
|| Log10FS_std =  1:
||  complexity emmean   SE   df lower.CL upper.CL
||  Complex       733 11.3 74.9      711      755
||  Simple        695 11.1 70.3      672      717
||
|| Results are averaged over the levels of: lang_type_ortho
|| Degrees-of-freedom method: kenward-roger
|| Confidence level used: 0.95
```

```r
# Estimate the slope of Log10FS within each complexity level
em_trends <- emtrends(anova_model_2, ~ complexity, var = "Log10FS_std")
summary(em_trends)
```

```
||  complexity Log10FS_std.trend   SE   df lower.CL upper.CL
||  Complex                5.596 2.43 4600    0.823    10.37
||  Simple                -0.834 2.03 4600   -4.812     3.14
||
|| Results are averaged over the levels of: lang_type_ortho
|| Degrees-of-freedom method: kenward-roger
|| Confidence level used: 0.95
```

1.6.3 Plots

...

```r
# Plot

# Refit the model using lmer()
anova_model_lmer <- lmer(
  response_time ~ complexity * Log10FS_std * lang_type_ortho + (1 | SubjID),
```

```r
  data = rt_nwords_1_cmpl,
  REML = FALSE
)

# Get predicted values
preds <- ggpredict(anova_model_lmer, terms = c("Log10FS_std", "complexity"))

# Plot
ggplot(preds, aes(x = x, y = predicted, color = group, fill = group)) +
  geom_line(linewidth = 1) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = 0.2, color = NA) +
  labs(x = "Standardized Log Family Size",
       y = "Predicted RT (ms)",
       color = "Complexity",
       fill = "Complexity",
       title = "Interaction of Morphological Complexity and Family Size on RT") +
  theme( plot.title = element_text(size = 8, hjust = .5),
         legend.title = element_blank(),
         axis.text.x = element_text(size = 8))
```