Stat 427 Consulting Project Group 7

Contents

Loading the Packages and Data

```
Sys.setenv(LANGUAGE = "en")

# Load Exploratory data analysis packages
library(dlookr)
```

```
## Imported Arial Narrow fonts.

##
## Attaching package: 'dlookr'

## The following object is masked from 'package:base':
##
##     transform
```

```r
library(GGally)
```

```
## Loading required package: ggplot2

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
library(readxl)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::extract() masks dlookr::extract()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```r
# Load data modeling packages
library(lme4)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```r
library(emmeans)
```

```
##
## Attaching package: 'emmeans'

## The following object is masked from 'package:GGally':
##
##     pigs
```

```r
library(optimx)
```

```
## Warning: package 'optimx' was built under R version 4.1.3
```

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

## The following object is masked from 'package:dlookr':
##
##     describe
```

```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:psych':
##
##     logit

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some
```

```
library(MuMIn)
```

```
## Warning: package 'MuMIn' was built under R version 4.1.3
```

```
library(pwr)
```

```
## Warning: package 'pwr' was built under R version 4.1.3
```

```
# Load data visualization packages
library(ggplot2)
library(ggpubr)

# Load data
priming_dataset <- read_excel("priming_dataset_cleaned_v2.xlsx")

# Remove gender, age, bilingual_type and lang_variety columns
priming_dataset <- priming_dataset %>%
  select(-c(gender,age,bilingual_type,lang_variety))
```

Feature Types:

Numerical Features:

- BLP
- language_use_span
- language_use_eng
- MLU_spa
- Words_Min_spa
- VOCD_spa
- MLU_eng
- Words_Min_eng
- VOCD_eng

Categorical Features:

- subject
- group
- phase
- construction
- mode
- target
- n_item

```
# Convert Categorical features into factors
Cat_features <- c("subject","group","phase","construction","mode","target","n_item")
priming_dataset[,Cat_features] <- lapply(priming_dataset[,Cat_features] , factor)

# View the data
priming_dataset %>%
  head()
```

```
## # A tibble: 6 x 16
##   subject group    phase    construction mode   target n_item   BLP language_use_sp~
##   <fct>   <fct>    <fct>    <fct>        <fct>  <fct>  <fct>  <dbl>            <dbl>
## 1 101     heritage pre-test acc          within no     1      -7.00               34
## 2 101     heritage pre-test acc          within no     2      -7.00               34
## 3 101     heritage pre-test acc          within no     3      -7.00               34
## 4 101     heritage pre-test acc          within no     4      -7.00               34
## 5 101     heritage pre-test acc          within no     5      -7.00               34
## 6 101     heritage pre-test acc          within no     6      -7.00               34
## # ... with 7 more variables: language_use_eng <dbl>, MLU_spa <dbl>,
## #   Words_Min_spa <dbl>, VOCD_spa <dbl>, MLU_eng <dbl>, Words_Min_eng <dbl>,
## #   VOCD_eng <dbl>
```

Missing Value

```r
# Check columns containing missing value
priming_dataset %>%
  select_if(function(x) any(is.na(x))) %>%
  summarise_each(funs(sum(is.na(.))))
```

```
## Warning: `summarise_each_()` was deprecated in dplyr 0.7.0.
## Please use `across()` instead.

## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))

## # A tibble: 1 x 0
```

No missing value

RQ1

Is the priming effect stronger with the ACC or with the SPE construction?

Exploratory Data Analysis (Categorical Features)

```r
# Filter the data set for RQ1
priming_dataset_rq1 <- priming_dataset %>%
  filter(mode=="within")

# Check the structure of RQ1 data
str(priming_dataset_rq1[,Cat_features])
```

```
## tibble [14,880 x 7] (S3: tbl_df/tbl/data.frame)
##  $ subject     : Factor w/ 124 levels "101","102","103",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ group       : Factor w/ 3 levels "first-gen","heritage",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ phase       : Factor w/ 3 levels "post-test","pre-test",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ construction: Factor w/ 2 levels "acc","spe": 1 1 1 1 1 1 1 1 1 1 ...
##  $ mode        : Factor w/ 2 levels "cross","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ target      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ n_item      : Factor w/ 180 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
```

Explore the relationship between "group" and "target"

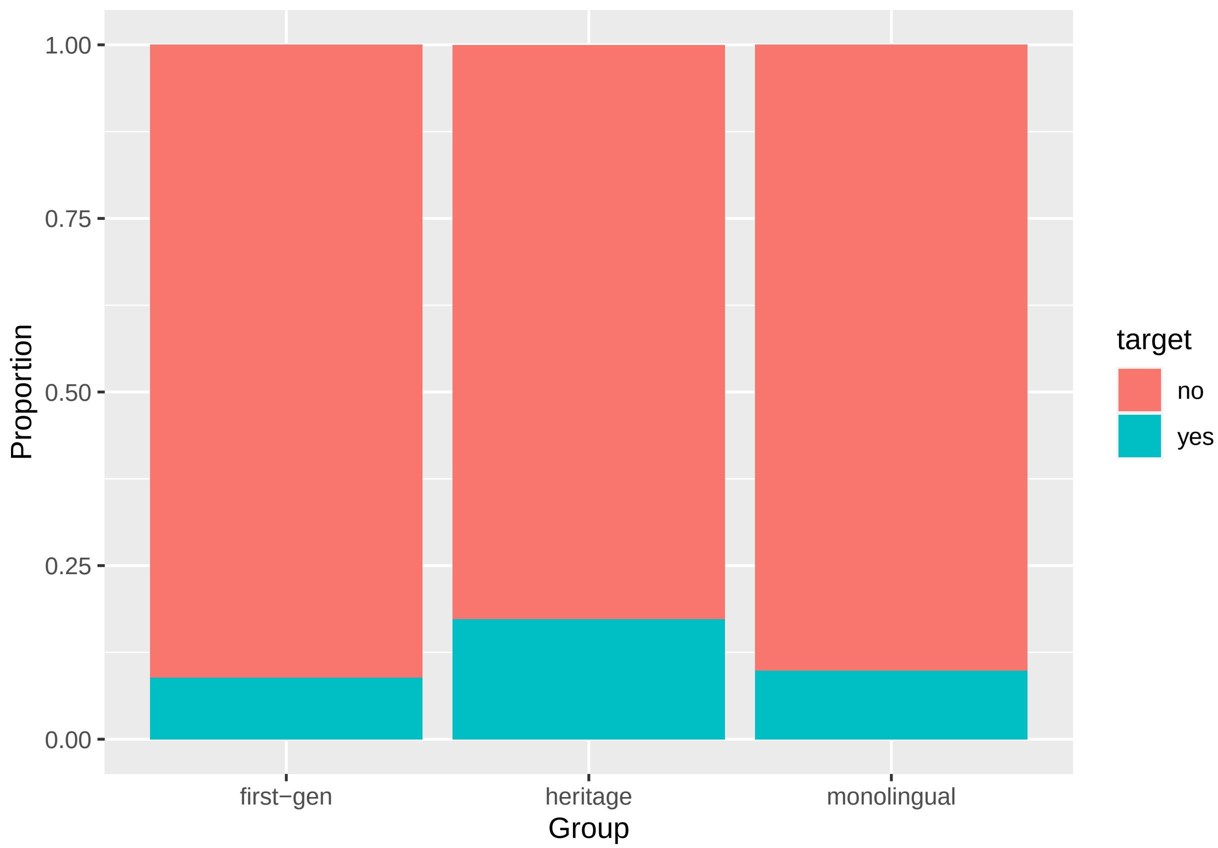
```
# Create a contingency table of the "target" and "group"
addmargins(table(priming_dataset_rq1$group, priming_dataset_rq1$target))
```

```
##
##                  no   yes   Sum
##   first-gen    2625   255  2880
##   heritage     3971   829  4800
##   monolingual  6488   712  7200
##   Sum         13084  1796 14880
```

```
# Create a proportional contingency table of the "target" and "group"
prop.table(table(priming_dataset_rq1$group, priming_dataset_rq1$target), margin=1)*100
```

```
##
##                      no       yes
##   first-gen   91.145833  8.854167
##   heritage    82.729167 17.270833
##   monolingual 90.111111  9.888889
```

```
# Visualize the proportional contingency table
ggplot(priming_dataset_rq1) +
  aes(x = group, fill = target) +
  geom_bar(position = "fill") +
  xlab("Group") +
  ylab("Proportion")
```




```r
# Chi-Square concept: https://data-flair.training/blogs/chi-square-test-in-r/

# Perform Chi-Square test
set.seed(1)
priming_dataset_rq1_sample <- priming_dataset_rq1[sample(length(priming_dataset_rq1$subject), 1000), ]
chisq.test(priming_dataset_rq1_sample$group, priming_dataset_rq1_sample$target)
```

```
##
##  Pearson's Chi-squared test
##
## data:  priming_dataset_rq1_sample$group and priming_dataset_rq1_sample$target
## X-squared = 16.993, df = 2, p-value = 0.0002042
```

Explore the relationship between “phase” and “target”

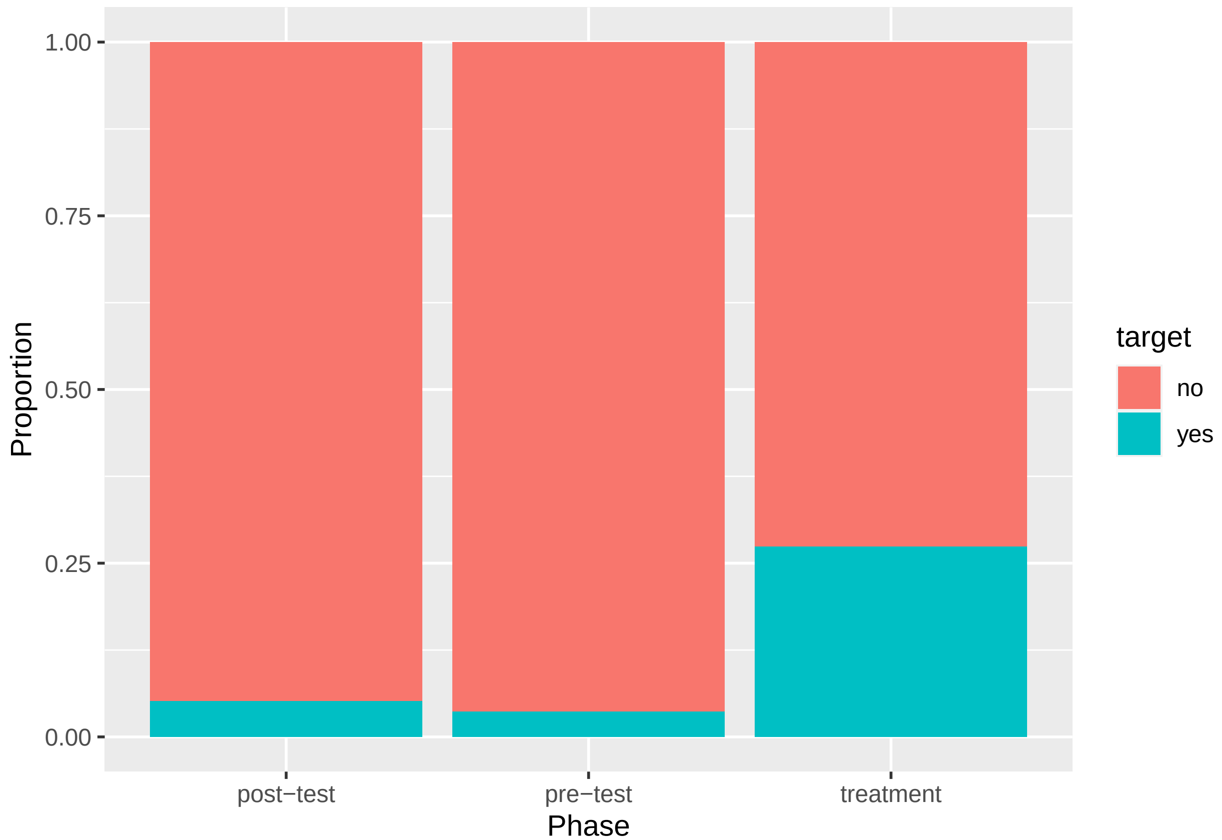
```r
# Create a contingency table of the "target" and "phase"
addmargins(table(priming_dataset_rq1$phase, priming_dataset_rq1$target))
```

```
##
##                no   yes   Sum
##   post-test  4704   256  4960
##   pre-test   4779   181  4960
##   treatment  3601  1359  4960
##   Sum       13084  1796 14880
```

```r
# Create a proportional contingency table of the "target" and "phase"
prop.table(table(priming_dataset_rq1$phase, priming_dataset_rq1$target), margin=1)*100
```

```
##
##                    no       yes
##   post-test 94.838710  5.161290
##   pre-test  96.350806  3.649194
##   treatment 72.600806 27.399194
```

```r
# Visualize the proportional contingency table
ggplot(priming_dataset_rq1) +
  aes(x = phase, fill = target) +
  geom_bar(position = "fill") +
  xlab("Phase") +
  ylab("Proportion")
```




```r
# Perform Chi-Square test
set.seed(1)
priming_dataset_rq1_sample <- priming_dataset_rq1[sample(length(priming_dataset_rq1$subject), 1000), ]
chisq.test(priming_dataset_rq1_sample$phase, priming_dataset_rq1_sample$target)
```

```
##
##  Pearson's Chi-squared test
##
```

```
## data:  priming_dataset_rq1_sample$phase and priming_dataset_rq1_sample$target
## X-squared = 116.65, df = 2, p-value < 2.2e-16
```

Explore the relationship between “construction” and “target”

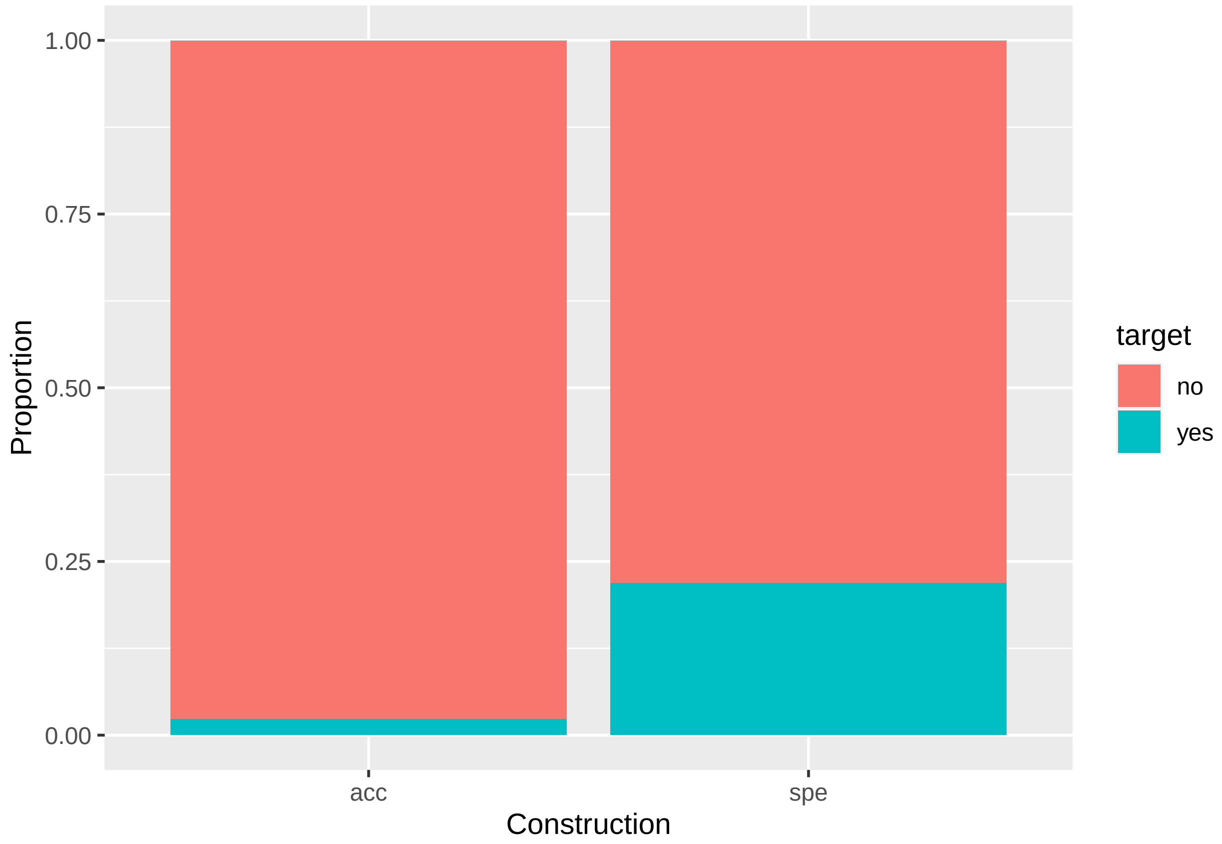
```r
# Create a contingency table of the "target" and "construction"
addmargins(table(priming_dataset_rq1$construction, priming_dataset_rq1$target))
```

```
##
##          no   yes   Sum
##   acc  7270   170  7440
##   spe  5814  1626  7440
##   Sum 13084  1796 14880
```

```r
# Create a proportional contingency table of the "target" and "construction"
prop.table(table(priming_dataset_rq1$construction, priming_dataset_rq1$target), margin=1)*100
```

```
##
##               no       yes
##   acc 97.715054  2.284946
##   spe 78.145161 21.854839
```

```r
# Visualize the proportional contingency table
ggplot(priming_dataset_rq1) +
  aes(x = construction, fill = target) +
  geom_bar(position = "fill") +
  xlab("Construction") +
  ylab("Proportion")
```




```
# Perform Chi-Square test
set.seed(1)
priming_dataset_rq1_sample <- priming_dataset_rq1[sample(length(priming_dataset_rq1$subject), 1000), ]
chisq.test(priming_dataset_rq1$construction, priming_dataset_rq1$target)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  priming_dataset_rq1$construction and priming_dataset_rq1$target
## X-squared = 1340.5, df = 1, p-value < 2.2e-16
```

Data Modeling

Generalized linear mixed model fit by maximum likelihood - item level

```
# Reorder the level of the features
priming_dataset_rq1$phase = relevel(priming_dataset_rq1$phase, ref = "pre-test")
priming_dataset_rq1$construction = relevel(priming_dataset_rq1$construction, ref = "spe")
priming_dataset_rq1$target = relevel(priming_dataset_rq1$target, ref = "no")

# How to avoid the convergence problem (Change a optimizer):  https://stats.stackexchange.com/questions

# Run the glm model
```

```r
glm_rq1 = lme4::glmer(target ~ phase * construction + group + (1|subject)+ (1|n_item),
                      data = priming_dataset_rq1, family = "binomial",
                      control = glmerControl(optimizer ='optimx', optCtrl=list(method='nlminb')))
summary(glm_rq1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: target ~ phase * construction + group + (1 | subject) + (1 |
##     n_item)
##    Data: priming_dataset_rq1
## Control: glmerControl(optimizer = "optimx", optCtrl = list(method = "nlminb"))
##
##      AIC      BIC   logLik deviance df.resid
##   6543.9   6620.0  -3262.0   6523.9    14870
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.9497 -0.2390 -0.1156 -0.0371 20.5335
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  subject (Intercept) 1.1286   1.0624
##  n_item  (Intercept) 0.3499   0.5915
## Number of obs: 14880, groups:  subject, 124; n_item, 120
##
## Fixed effects:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -3.7857     0.2847 -13.296  < 2e-16 ***
## phasepost-test                    0.3888     0.2203   1.765   0.0776 .
## phasetreatment                    3.2105     0.2145  14.966  < 2e-16 ***
## constructionacc                  -3.7316     0.4593  -8.125 4.48e-16 ***
## groupheritage                     1.2372     0.2944   4.203 2.64e-05 ***
## groupmonolingual                  0.2704     0.2769   0.977   0.3288
## phasepost-test:constructionacc    1.1400     0.5398   2.112   0.0347 *
## phasetreatment:constructionacc    0.2730     0.5064   0.539   0.5898
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) phsps- phstrt cnstrc grphrt grpmnl phsp-:
## phaspst-tst -0.404
## phasetrtmnt -0.440  0.535
## constrctncc -0.183  0.244  0.244
## groupheritg -0.669  0.002  0.020 -0.004
## groupmnlngl -0.702  0.000  0.004  0.000  0.677
## phspst-tst:  0.162 -0.406 -0.214 -0.844  0.000 -0.001
## phstrtmnt:c  0.178 -0.223 -0.414 -0.903 -0.005 -0.002  0.765
```

```r
# Estimate marginal means by construction
pairs(emmeans(glm_rq1, "phase", by = "construction"))
```

```
## construction = spe:
```

```
##  contrast                  estimate    SE  df z.ratio p.value
##  (pre-test) - (post-test)   -0.389 0.220 Inf  -1.765  0.1815
##  (pre-test) - treatment     -3.211 0.215 Inf -14.966  <.0001
##  (post-test) - treatment    -2.822 0.210 Inf -13.457  <.0001
##
## construction = acc:
##  contrast                  estimate    SE  df z.ratio p.value
##  (pre-test) - (post-test)   -1.529 0.493 Inf  -3.098  0.0055
##  (pre-test) - treatment     -3.484 0.461 Inf  -7.558  <.0001
##  (post-test) - treatment    -1.955 0.294 Inf  -6.648  <.0001
##
## Results are averaged over the levels of: group
## Results are given on the log odds ratio (not the response) scale.
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```r
# Estimate marginal means by phase
pairs(emmeans(glm_rq1, "construction", by = "phase"))
```

```
## phase = pre-test:
##  contrast  estimate    SE  df z.ratio p.value
##  spe - acc     3.73 0.459 Inf   8.125  <.0001
##
## phase = post-test:
##  contrast  estimate    SE  df z.ratio p.value
##  spe - acc     2.59 0.289 Inf   8.954  <.0001
##
## phase = treatment:
##  contrast  estimate    SE  df z.ratio p.value
##  spe - acc     3.46 0.218 Inf  15.876  <.0001
##
## Results are averaged over the levels of: group
## Results are given on the log odds ratio (not the response) scale.
```

Power Analysis

```r
# Pseudo-R-square for the GLM Function: https://search.r-project.org/CRAN/refmans/MuMIn/html/r.squaredG

# Calculate Pseudo-R-squared for Generalized Mixed-Effect models
MuMIn::r.squaredGLMM(glm_rq1)
```

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

```
## Warning: the null model is correct only if all variables used by the original
## model remain unchanged.
```

```
##                   R2m       R2c
## theoretical 0.5091542 0.6613498
## delta       0.3294668 0.4279505
```

```r
# Power Analysis in linguistic area (Literature Review): https://www.jstor.org/stable/3587103

# Use R to calculate the power https://cran.r-project.org/web/packages/pwr/vignettes/pwr-vignette.html

# Calculate the power for RQ1
pwr.f2.test(u = 7, v =  124-7-1, f2 = 0.3294668/(1-0.3294668), sig.level = 0.05)
```

```
##
##      Multiple regression power calculation
##
##               u = 7
##               v = 116
##              f2 = 0.4913505
##       sig.level = 0.05
##           power = 0.999994
```

RQ2

Is the priming effect stronger in within-language mode or in cross-linguistic mode?

Exploratory Data Analysis (Categorical Features)

```r
# Filter the data set for RQ1
priming_dataset_rq2 <- priming_dataset %>%
  filter(construction=="spe")

# Check the structure of RQ1 data
str(priming_dataset_rq2[,Cat_features])
```

```
## tibble [14,880 x 7] (S3: tbl_df/tbl/data.frame)
##  $ subject     : Factor w/ 124 levels "101","102","103",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ group       : Factor w/ 3 levels "first-gen","heritage",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ phase       : Factor w/ 3 levels "post-test","pre-test",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ construction: Factor w/ 2 levels "acc","spe": 2 2 2 2 2 2 2 2 2 2 ...
##  $ mode        : Factor w/ 2 levels "cross","within": 1 1 1 1 1 1 1 1 1 1 ...
##  $ target      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ n_item      : Factor w/ 180 levels "1","2","3","4",..: 61 62 63 64 65 66 67 68 69 70 ...
```

Explore the relationship between "group" and "target"

```r
# Create a contingency table of the "target" and "group"
addmargins(table(priming_dataset_rq2$group, priming_dataset_rq2$target))
```

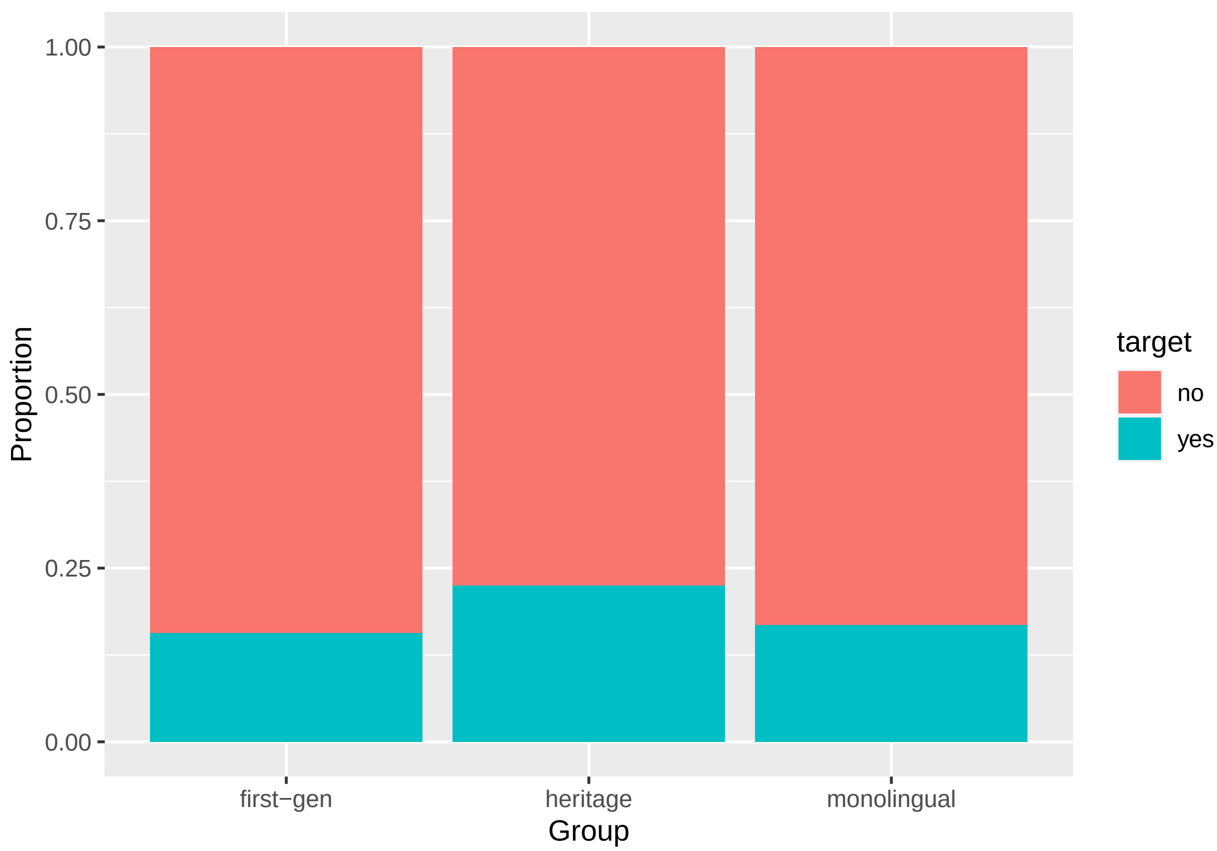
```
##
##                  no   yes   Sum
##   first-gen    2429   451  2880
##   heritage     3722  1078  4800
```

```
##   monolingual  5989  1211  7200
##   Sum         12140  2740 14880
```

```
# Create a proportional contingency table of the "target" and "group"
prop.table(table(priming_dataset_rq2$group, priming_dataset_rq2$target), margin=1)*100
```

```
##
##                     no      yes
##   first-gen   84.34028 15.65972
##   heritage    77.54167 22.45833
##   monolingual 83.18056 16.81944
```

```
# Visualize the proportional contingency table
ggplot(priming_dataset_rq2) +
  aes(x = group, fill = target) +
  geom_bar(position = "fill") +
  xlab("Group") +
  ylab("Proportion")
```




```
# Perform Chi-Square test
set.seed(1)
priming_dataset_rq2_sample <- priming_dataset_rq2[sample(length(priming_dataset_rq2$subject), 1000), ]
chisq.test(priming_dataset_rq2_sample$group, priming_dataset_rq2_sample$target)
```

```
##
```

```
##  Pearson's Chi-squared test
##
## data:  priming_dataset_rq2_sample$group and priming_dataset_rq2_sample$target
## X-squared = 12.131, df = 2, p-value = 0.002321
```

Explore the relationship between "phase" and "target"

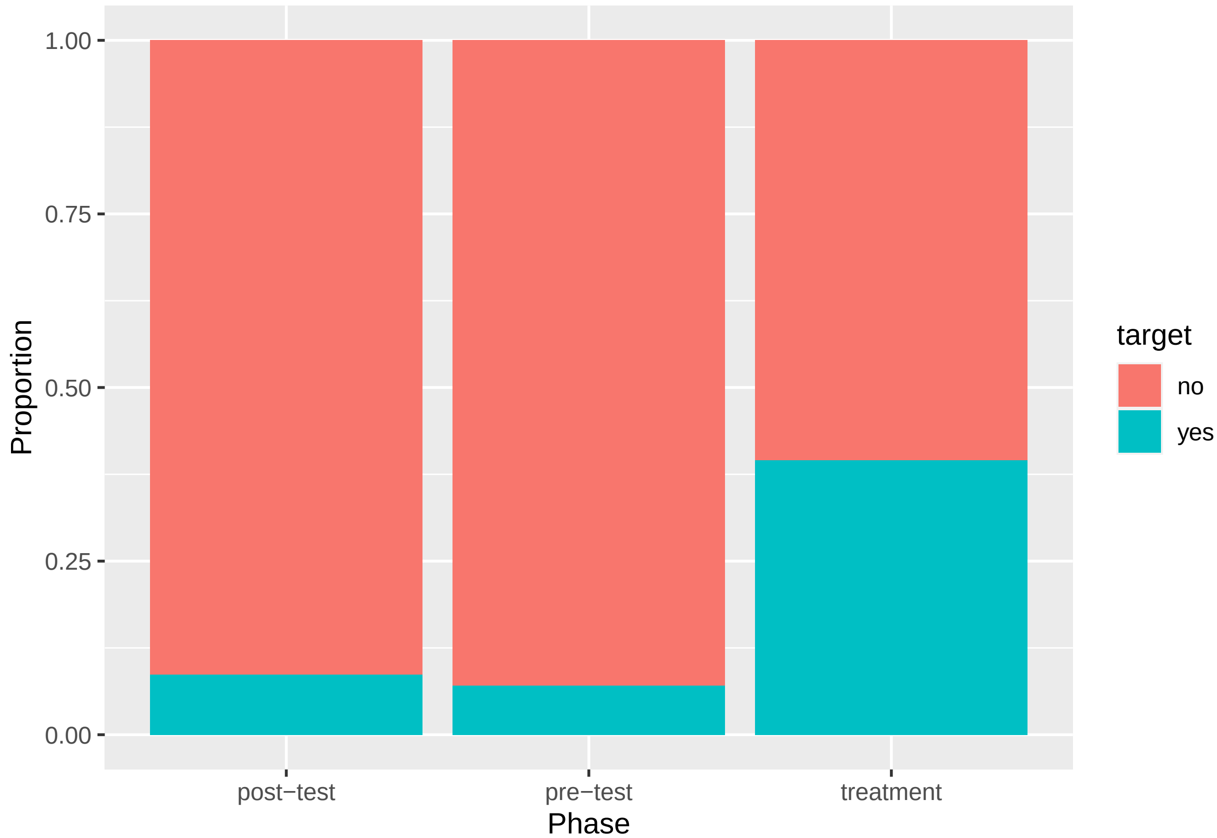
```r
# Create a contingency table of the "target" and "phase"
addmargins(table(priming_dataset_rq2$phase, priming_dataset_rq2$target))
```

```
##
##                no   yes   Sum
##   post-test  4531   429  4960
##   pre-test   4609   351  4960
##   treatment  3000  1960  4960
##   Sum       12140  2740 14880
```

```r
# Create a proportional contingency table of the "target" and "phase"
prop.table(table(priming_dataset_rq2$phase, priming_dataset_rq2$target), margin=1)*100
```

```
##
##                    no       yes
##   post-test 91.350806  8.649194
##   pre-test  92.923387  7.076613
##   treatment 60.483871 39.516129
```

```r
# Visualize the proportional contingency table
ggplot(priming_dataset_rq2) +
  aes(x = phase, fill = target) +
  geom_bar(position = "fill") +
  xlab("Phase") +
  ylab("Proportion")
```




```
# Perform Chi-Square test
set.seed(1)
priming_dataset_rq2_sample <- priming_dataset_rq2[sample(length(priming_dataset_rq2$subject), 1000), ]
chisq.test(priming_dataset_rq2_sample$phase, priming_dataset_rq2_sample$target)
```

```
##
##  Pearson's Chi-squared test
##
## data:  priming_dataset_rq2_sample$phase and priming_dataset_rq2_sample$target
## X-squared = 151.37, df = 2, p-value < 2.2e-16
```

Explore the relationship between "construction" and "target"

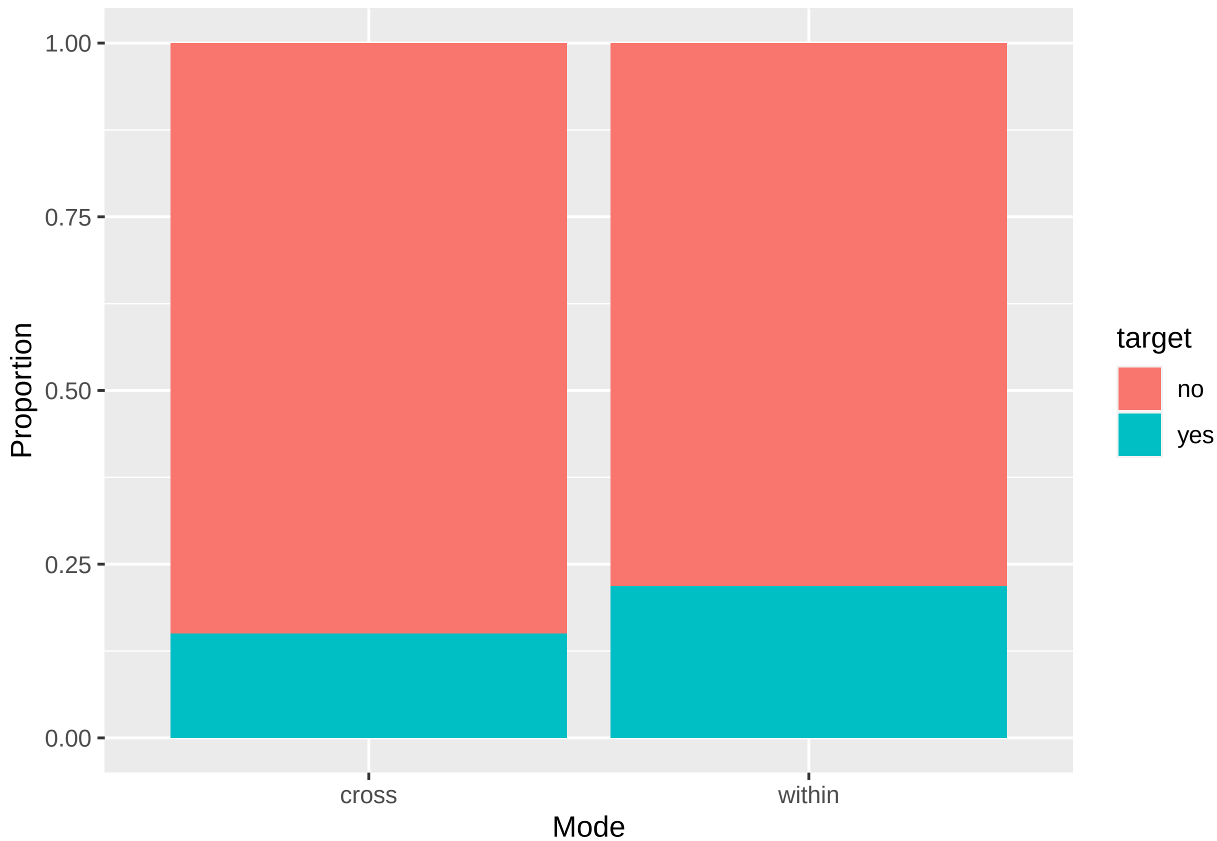
```
# Create a contingency table of the "target" and "mode"
addmargins(table(priming_dataset_rq2$mode, priming_dataset_rq2$target))
```

```
##
##              no   yes   Sum
##   cross    6326  1114  7440
##   within   5814  1626  7440
##   Sum     12140  2740 14880
```

Create a proportional contingency table of the "target" and "mode"
```
# Create a proportional contingency table of the "target" and "mode"
prop.table(table(priming_dataset_rq2$mode, priming_dataset_rq2$target), margin=1)*100
```

```
##
##                no      yes
##   cross  85.02688 14.97312
##   within 78.14516 21.85484
```

```
# Visualize the proportional contingency table
ggplot(priming_dataset_rq2) +
  aes(x = mode, fill = target) +
  geom_bar(position = "fill") +
  xlab("Mode") +
  ylab("Proportion")
```




```
# Perform Chi-Square test
set.seed(1)
priming_dataset_rq2_sample <- priming_dataset_rq2[sample(length(priming_dataset_rq2$subject), 1000), ]
chisq.test(priming_dataset_rq2_sample$mode, priming_dataset_rq2_sample$target)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  priming_dataset_rq2_sample$mode and priming_dataset_rq2_sample$target
## X-squared = 13.736, df = 1, p-value = 0.0002103
```

Data Modeling

Generalized linear mixed model fit by maximum likelihood - item level

```
# Reorder the level of the features
priming_dataset_rq2$phase = relevel(priming_dataset_rq2$phase, ref = "pre-test")
priming_dataset_rq2$mode = relevel(priming_dataset_rq2$mode, ref = "within")
priming_dataset_rq2$target = relevel(priming_dataset_rq2$target, ref = "no")

# Run the glm model
glm_rq2 = lme4::glmer(target ~ phase * mode + group + (1|subject)+ (1|n_item),
                     data = priming_dataset_rq2, family = "binomial",
                     control = glmerControl(optimizer ='optimx', optCtrl=list(method='nlminb')))

summary(glm_rq2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: target ~ phase * mode + group + (1 | subject) + (1 | n_item)
##    Data: priming_dataset_rq2
## Control: glmerControl(optimizer = "optimx", optCtrl = list(method = "nlminb"))
##
##      AIC      BIC   logLik deviance df.resid
##  10056.4  10132.5  -5018.2  10036.4    14870
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.8400 -0.3489 -0.2054 -0.1108 16.5800
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  subject (Intercept) 1.212    1.1010
##  n_item  (Intercept) 0.419    0.6473
## Number of obs: 14880, groups:  subject, 124; n_item, 120
##
## Fixed effects:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -3.4822     0.2905 -11.988  < 2e-16 ***
## phasepost-test             0.4071     0.2373   1.716 0.086206 .
## phasetreatment             3.2150     0.2301  13.973  < 2e-16 ***
## modecross                  0.0161     0.2410   0.067 0.946754
## groupheritage              0.5716     0.2959   1.932 0.053389 .
## groupmonolingual           0.1071     0.2773   0.386 0.699230
## phasepost-test:modecross  -0.2149     0.3365  -0.639 0.523012
## phasetreatment:modecross  -1.0860     0.3237  -3.355 0.000794 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) phsps- phstrt mdcrss grphrt grpmnl phsp-:
## phaspst-tst -0.425
## phasetrtmnt -0.449  0.536
```

```
## modecross   -0.416  0.509  0.525
## groupheritg -0.644  0.001  0.006  0.000
## groupmnlngl -0.684  0.000  0.002  0.000  0.671
## phspst-tst:  0.299 -0.705 -0.377 -0.717  0.000  0.000
## phstrtmnt:m  0.313 -0.380 -0.704 -0.744 -0.001  0.000  0.534
```

```
# Estimate marginal means by mode
pairs(emmeans(glm_rq2, "phase", by = "mode"))
```

```
## mode = within:
##  contrast                 estimate    SE  df z.ratio p.value
##  (pre-test) - (post-test)   -0.407 0.237 Inf  -1.716  0.1992
##  (pre-test) - treatment     -3.215 0.230 Inf -13.973  <.0001
##  (post-test) - treatment    -2.808 0.225 Inf -12.466  <.0001
##
## mode = cross:
##  contrast                 estimate    SE  df z.ratio p.value
##  (pre-test) - (post-test)   -0.192 0.239 Inf  -0.805  0.6996
##  (pre-test) - treatment     -2.129 0.230 Inf  -9.256  <.0001
##  (post-test) - treatment    -1.937 0.228 Inf  -8.512  <.0001
##
## Results are averaged over the levels of: group
## Results are given on the log odds ratio (not the response) scale.
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```
# Estimate marginal means by phase
pairs(emmeans(glm_rq2, "mode", by = "phase"))
```

```
## phase = pre-test:
##  contrast       estimate    SE  df z.ratio p.value
##  within - cross  -0.0161 0.241 Inf  -0.067  0.9468
##
## phase = post-test:
##  contrast       estimate    SE  df z.ratio p.value
##  within - cross   0.1988 0.235 Inf   0.847  0.3969
##
## phase = treatment:
##  contrast       estimate    SE  df z.ratio p.value
##  within - cross   1.0699 0.216 Inf   4.950  <.0001
##
## Results are averaged over the levels of: group
## Results are given on the log odds ratio (not the response) scale.
```

Why mode not significant?

```
# Split the data set based on phase
priming_dataset_rq2_pre <- priming_dataset_rq2 %>%
  filter(phase=="pre-test")
priming_dataset_rq2_treat <- priming_dataset_rq2 %>%
  filter(phase=="treatment")
priming_dataset_rq2_post <- priming_dataset_rq2 %>%
```

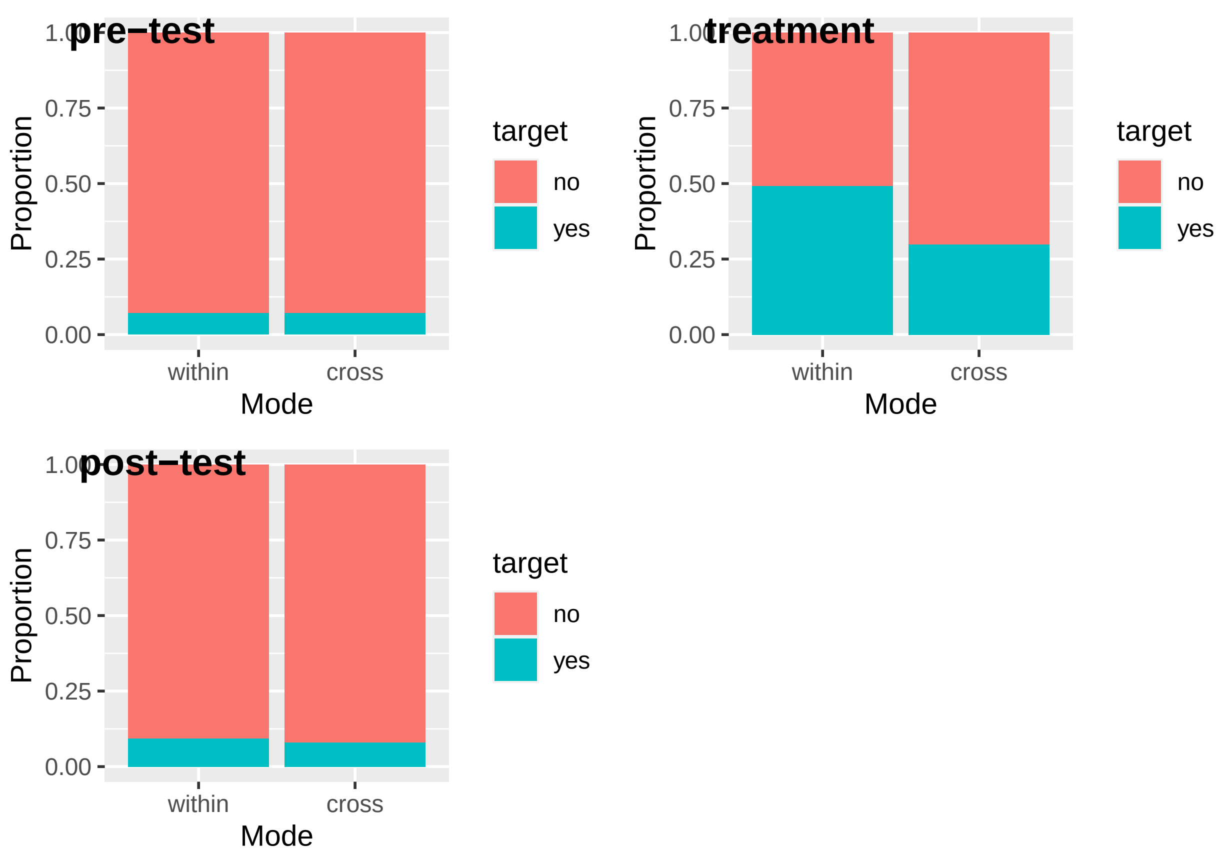
```r
  filter(phase=="post-test")

# Visualize the proportional contingency tables by different phases
pre_plot <- ggplot(priming_dataset_rq2_pre) +
  aes(x = mode, fill = target) +
  geom_bar(position = "fill") +
  xlab("Mode") +
  ylab("Proportion")

treat_plot <- ggplot(priming_dataset_rq2_treat) +
  aes(x = mode, fill = target) +
  geom_bar(position = "fill") +
  xlab("Mode") +
  ylab("Proportion")

post_plot <- ggplot(priming_dataset_rq2_post) +
  aes(x = mode, fill = target) +
  geom_bar(position = "fill") +
  xlab("Mode") +
  ylab("Proportion")

ggarrange(pre_plot, treat_plot, post_plot,
          labels = c("pre-test", "treatment", "post-test"),
          ncol = 2, nrow = 2)
```



Power Analysis

```
# Calculate Pseudo-R-squared for Generalized Mixed-Effect models
MuMIn::r.squaredGLMM(glm_rq2)
```

```
## Warning: the null model is correct only if all variables used by the original
## model remain unchanged.
```

```
##                   R2m       R2c
## theoretical 0.2433331 0.4941472
## delta       0.1616664 0.3283031
```

```
# Calculate the power for RQ2
pwr.f2.test(u = 7, v =  124-7-1, f2 = 0.1616664/(1-0.1616664), sig.level = 0.05)
```

```
##
##      Multiple regression power calculation
##
##               u = 7
##               v = 116
##              f2 = 0.1928426
##       sig.level = 0.05
##           power = 0.9564702
```

RQ3

Which individual variables are associated with a strong priming effect?

Exploratory Data Analysis (Numerical Features)

```
# Select numerical features
priming_dataset_rq3 <- priming_dataset %>%
  select(c("subject","n_item","target","BLP","language_use_span","language_use_eng","MLU_spa","Words_Mi

priming_dataset_rq3 %>%
  head()
```

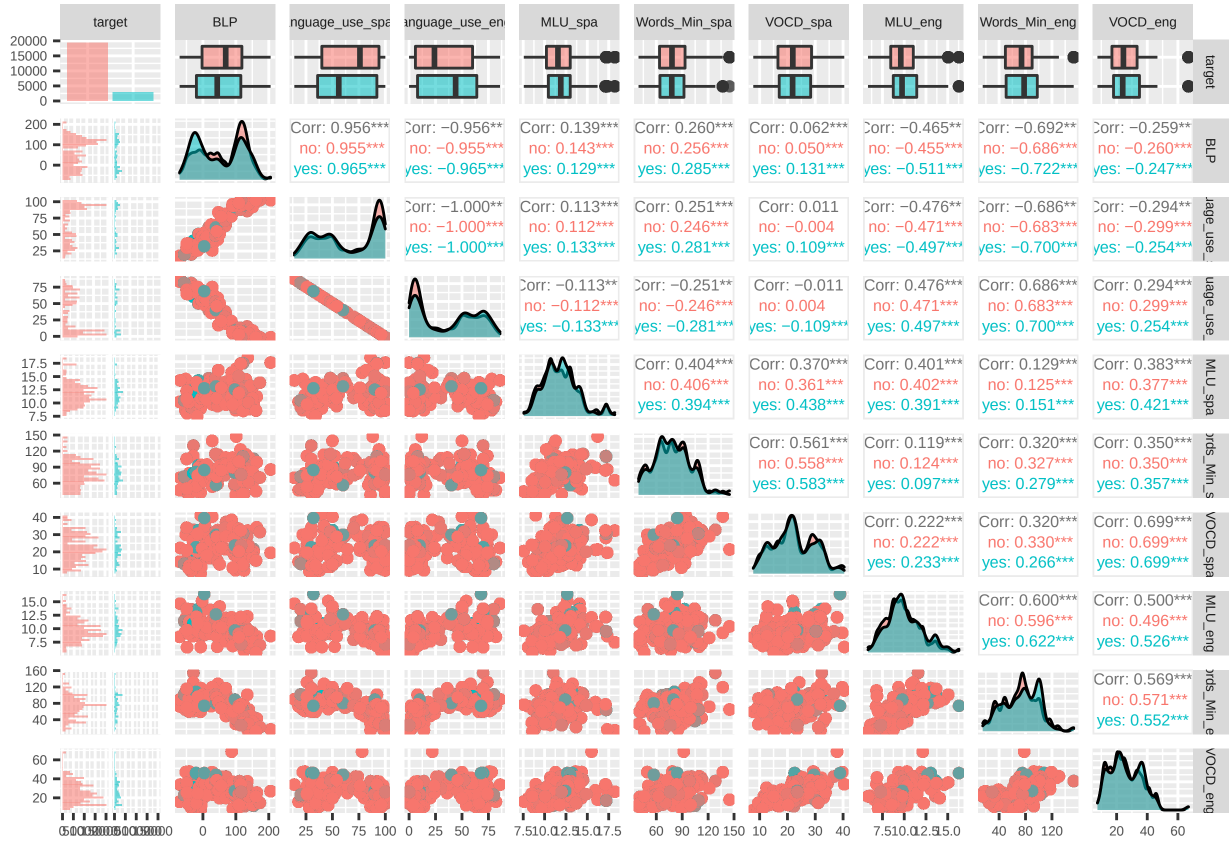
```
## # A tibble: 6 x 12
##   subject n_item target   BLP language_use_span language_use_eng MLU_spa
##   <fct>   <fct>  <fct>  <dbl>             <dbl>            <dbl>   <dbl>
## 1 101     1      no     -7.00                34               66       9
## 2 101     2      no     -7.00                34               66       9
## 3 101     3      no     -7.00                34               66       9
## 4 101     4      no     -7.00                34               66       9
## 5 101     5      no     -7.00                34               66       9
## 6 101     6      no     -7.00                34               66       9
## # ... with 5 more variables: Words_Min_spa <dbl>, VOCD_spa <dbl>,
## #   MLU_eng <dbl>, Words_Min_eng <dbl>, VOCD_eng <dbl>
```

Pairplot Analysis

```r
# Pairplot theme https://ggplot2.tidyverse.org/reference/theme.html
# Pairplot text size: https://stackoverflow.com/questions/8599685/how-to-change-correlation-text-size-i

# Check the pair plot
ggpairs(priming_dataset_rq3, columns = 3:12,upper=list(continuous = wrap("cor",size=2)), aes(colour=tar
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Correlation analysis

```r
# Check strongly correlated variables
cor_matrix <- as.data.frame(cor(priming_dataset_rq3[,4:12],method="pearson"))
```

```r
cor_matrix[abs(cor_matrix) < 0.5] <- ""
cor_matrix
```

```
##                                  BLP  language_use_span   language_use_eng
## BLP                                1  0.956369260693689 -0.956369260693689
## language_use_span  0.956369260693689                  1                 -1
## language_use_eng  -0.956369260693689                 -1                  1
## MLU_spa
## Words_Min_spa
## VOCD_spa
## MLU_eng
## Words_Min_eng     -0.691594036188055 -0.685953052061741  0.685953052061741
## VOCD_eng
##                   MLU_spa     Words_Min_spa          VOCD_spa           MLU_eng
## BLP
## language_use_span
## language_use_eng
## MLU_spa                 1
## Words_Min_spa                                 1 0.561369796562773
## VOCD_spa                  0.561369796562773                  1
## MLU_eng                                                                       1
## Words_Min_eng                                                 0.600384696831456
## VOCD_eng                                      0.698833441089408 0.500405942491648
##                        Words_Min_eng          VOCD_eng
## BLP               -0.691594036188055
## language_use_span -0.685953052061741
## language_use_eng   0.685953052061741
## MLU_spa
## Words_Min_spa
## VOCD_spa                             0.698833441089408
## MLU_eng            0.600384696831456 0.500405942491648
## Words_Min_eng                      1  0.56868544813707
## VOCD_eng            0.56868544813707                 1
```

- "BLP" is strongly correlated with "language_use_span","language_use_eng" and "Words_Min_eng"
- "language_use_span" is exactly correlated with "language_use_eng" and strongly correlated with "Words_Min_eng"

```r
# Remove "language_use_eng"
priming_dataset_rq3 <- priming_dataset_rq3 %>%
  select(-language_use_eng)

priming_dataset_rq3 %>%
  head()
```

```
## # A tibble: 6 x 11
##   subject n_item target   BLP language_use_span MLU_spa Words_Min_spa VOCD_spa
##   <fct>   <fct>  <fct>  <dbl>             <dbl>   <dbl>         <dbl>    <dbl>
## 1 101     1      no     -7.00                34       9          43.6     7.69
## 2 101     2      no     -7.00                34       9          43.6     7.69
## 3 101     3      no     -7.00                34       9          43.6     7.69
## 4 101     4      no     -7.00                34       9          43.6     7.69
```

```
## 5 101     5      no     -7.00                34       9           43.6     7.69
## 6 101     6      no     -7.00                34       9           43.6     7.69
## # ... with 3 more variables: MLU_eng <dbl>, Words_Min_eng <dbl>, VOCD_eng <dbl>
```

```
# The the mean difference of individual variables between target yes and target no
priming_dataset_rq3 %>%
  group_by(target) %>%
  summarize(BLP_mean=mean(BLP), language_use_span_mean=mean(language_use_span), MLU_spa_mean=mean(MLU_s
```

```
## # A tibble: 2 x 9
##   target BLP_mean language_use_span~ MLU_spa_mean Words_Min_spa_m~ VOCD_spa_mean
##   <fct>     <dbl>              <dbl>        <dbl>            <dbl>         <dbl>
## 1 no         60.2               67.0         11.8             79.0          22.6
## 2 yes        48.7               62.6         11.9             77.9          22.3
## # ... with 3 more variables: MLU_eng_mean <dbl>, Words_Min_eng_mean <dbl>,
## #   VOCD_eng_mean <dbl>
```

Data Modeling

Linear Regression (subject level)

We recommend our client to use glm method not linear regression in RQ3. Therefore, this part can be used just as FYI

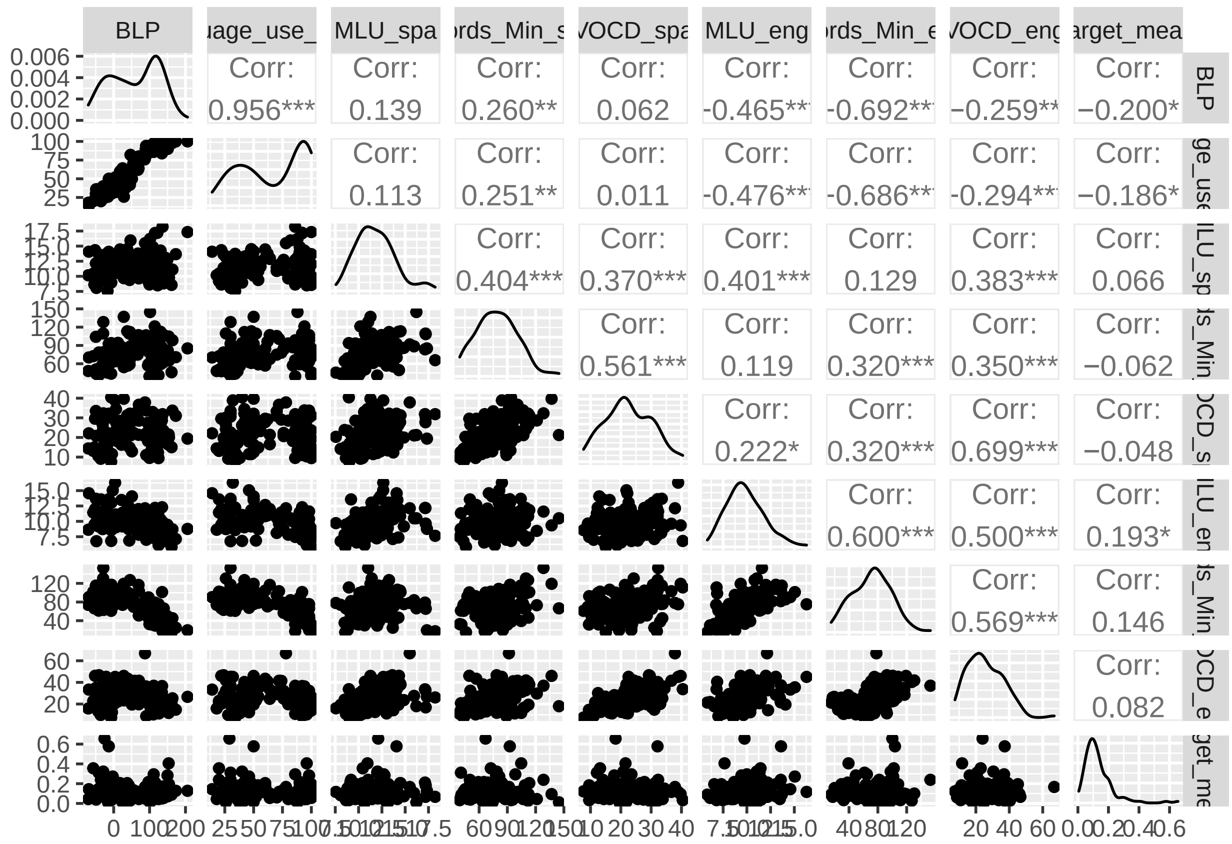
```
# Simplify the data set into subject level
priming_dataset_rq3_sbj <- priming_dataset_rq3 %>%
  select(-n_item) %>%
  mutate(target_num=ifelse(target=="no",0,1)) %>%
  group_by(subject,BLP,language_use_span,MLU_spa,Words_Min_spa,VOCD_spa,MLU_eng,Words_Min_eng,VOCD_eng)
  summarize(target_mean=mean(target_num)) %>%
  as.data.frame()
```

```
## `summarise()` has grouped output by 'subject', 'BLP', 'language_use_span', 'MLU_spa', 'Words_Min_spa
```

```
priming_dataset_rq3_sbj%>%
  head()
```

```
##   subject     BLP language_use_span MLU_spa Words_Min_spa VOCD_spa MLU_eng
## 1     101  -6.998                34   9.000        43.636     7.69   8.571
## 2     102  17.710                60  14.455        84.444    27.80  10.840
## 3     103 -28.792                30  11.022       128.372    32.30  11.591
## 4     104  23.608                50  12.952        64.000    21.84  11.045
## 5     108 -29.338                26  11.410       109.733    24.24  11.059
## 6     109 -53.674                22   8.636        47.802    14.93   9.474
##   Words_Min_eng VOCD_eng target_mean
## 1        61.165     7.49  0.12222222
## 2       104.460    39.08  0.20555556
## 3       152.948    36.95  0.23888889
## 4        75.542    38.18  0.01666667
## 5       130.541    41.79  0.10000000
## 6        66.475    20.29  0.12222222
```

```r
# Check the pair plot
ggpairs(priming_dataset_rq3_sbj, columns = 2:10)
```



```r
# Perform the linear regression model
lm.fit <- lm(target_mean~BLP+language_use_span+MLU_spa+Words_Min_spa+VOCD_spa+MLU_eng+Words_Min_eng+VOCI
summary(lm.fit)
```

```
##
## Call:
## lm(formula = target_mean ~ BLP + language_use_span + MLU_spa +
##     Words_Min_spa + VOCD_spa + MLU_eng + Words_Min_eng + VOCD_eng,
##     data = priming_dataset_rq3_sbj)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.13954 -0.05191 -0.01441  0.02711  0.50434
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6.919e-02  8.359e-02   0.828    0.410
## BLP               -3.240e-04  4.669e-04  -0.694    0.489
## language_use_span  2.996e-04  1.092e-03   0.274    0.784
## MLU_spa            3.357e-03  5.308e-03   0.632    0.528
## Words_Min_spa     -8.598e-05  6.767e-04  -0.127    0.899
## VOCD_spa          -1.445e-03  1.902e-03  -0.760    0.449
```

```
## MLU_eng            5.101e-03  6.372e-03   0.800    0.425
## Words_Min_eng     -7.185e-05  7.323e-04  -0.098    0.922
## VOCD_eng           5.856e-04  1.428e-03   0.410    0.682
##
## Residual standard error: 0.09572 on 115 degrees of freedom
## Multiple R-squared:  0.0642, Adjusted R-squared:  -0.0009007
## F-statistic: 0.9862 on 8 and 115 DF,  p-value: 0.4505
```

```
# vif and multicollinearity https://www.analyticsvidhya.com/blog/2020/03/what-is-multicollinearity/
vif(lm.fit)
```

```
##               BLP language_use_span           MLU_spa     Words_Min_spa
##         13.573668         12.539584          1.661628          2.859228
##          VOCD_spa           MLU_eng     Words_Min_eng          VOCD_eng
##          2.740503          2.137383          5.185029          3.054041
```

BLP and language_use_span have vif greater than 10. Therefore, remove one of them (language_use_span) to address the multicollinearity problem.

```
# Perform the linear regression model
lm.fit <- lm(target_mean~BLP+MLU_spa+Words_Min_spa+VOCD_spa+MLU_eng+Words_Min_eng+VOCD_eng, data = primi
summary(lm.fit)
```

```
##
## Call:
## lm(formula = target_mean ~ BLP + MLU_spa + Words_Min_spa + VOCD_spa +
##     MLU_eng + Words_Min_eng + VOCD_eng, data = priming_dataset_rq3_sbj)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.14001 -0.04955 -0.01415  0.02734  0.50301
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.0842635  0.0627660   1.343    0.182
## BLP           -0.0002156  0.0002480  -0.869    0.386
## MLU_spa        0.0033397  0.0052865   0.632    0.529
## Words_Min_spa -0.0000533  0.0006635  -0.080    0.936
## VOCD_spa      -0.0015131  0.0018783  -0.806    0.422
## MLU_eng        0.0050281  0.0063414   0.793    0.429
## Words_Min_eng -0.0000947  0.0007246  -0.131    0.896
## VOCD_eng       0.0005846  0.0014221   0.411    0.682
##
## Residual standard error: 0.09534 on 116 degrees of freedom
## Multiple R-squared:  0.06359,    Adjusted R-squared:  0.007078
## F-statistic: 1.125 on 7 and 116 DF,  p-value: 0.3521
```

```
# Perform stepwise regression for variable selection
full <- lm(target_mean~BLP+MLU_spa+Words_Min_spa+VOCD_spa+MLU_eng+Words_Min_eng+VOCD_eng, data = priming

null <- lm(target_mean~1, data = priming_dataset_rq3_sbj)

step(null,scope=list(upper=full,lower=null), data =priming_dataset_rq3_sbj, direction="both")
```

```
## Start:  AIC=-581
## target_mean ~ 1
##
##                 Df Sum of Sq    RSS     AIC
## + BLP            1  0.045232 1.0807 -584.09
## + MLU_eng        1  0.042041 1.0839 -583.72
## + Words_Min_eng  1  0.024109 1.1018 -581.69
## <none>                       1.1260 -581.00
## + VOCD_eng       1  0.007570 1.1184 -579.84
## + MLU_spa        1  0.004890 1.1211 -579.54
## + Words_Min_spa  1  0.004397 1.1216 -579.49
## + VOCD_spa       1  0.002623 1.1233 -579.29
##
## Step:  AIC=-584.09
## target_mean ~ BLP
##
##                 Df Sum of Sq    RSS     AIC
## <none>                       1.0807 -584.09
## + MLU_eng        1  0.014392 1.0663 -583.75
## + MLU_spa        1  0.010111 1.0706 -583.25
## + VOCD_spa       1  0.001456 1.0793 -582.26
## + VOCD_eng       1  0.001096 1.0796 -582.21
## + Words_Min_spa  1  0.000132 1.0806 -582.10
## + Words_Min_eng  1  0.000128 1.0806 -582.10
## - BLP            1  0.045232 1.1260 -581.00
```

```
##
## Call:
## lm(formula = target_mean ~ BLP, data = priming_dataset_rq3_sbj)
##
## Coefficients:
## (Intercept)          BLP
##   0.1468994   -0.0002816
```

```
# Perform the linear regression model after stepwise variable selection
lm.fit_sig <- lm(target_mean ~ BLP, data = priming_dataset_rq3_sbj)
summary(lm.fit_sig)
```

```
##
## Call:
## lm(formula = target_mean ~ BLP, data = priming_dataset_rq3_sbj)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.14021 -0.04886 -0.01792  0.03072  0.50147
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.1468994  0.0111761   13.14   <2e-16 ***
## BLP         -0.0002816  0.0001246   -2.26   0.0256 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.09412 on 122 degrees of freedom
## Multiple R-squared:  0.04017,    Adjusted R-squared:  0.0323
## F-statistic: 5.106 on 1 and 122 DF,  p-value: 0.02562
```

Generalized linear mixed model fit by maximum likelihood

```r
# Parameters or bounds appear to have different scalings can cause poor performance in optimization. Th

# Scale the features
priming_dataset_rq3_std <- priming_dataset_rq3 %>%
  mutate_at(colnames(priming_dataset_rq3)[4:11], ~(scale(.) %>% as.vector))

# Perform the generalized linear mixed model fit by maximum likelihood
glm_rq3 <- lme4::glmer(target ~ BLP +  language_use_span + MLU_spa + Words_Min_spa + VOCD_spa + MLU_eng
                     data = priming_dataset_rq3_std, family = "binomial",
                     control = glmerControl(optimizer ='optimx', optCtrl=list(method='nlminb')))
summary(glm_rq3)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: target ~ BLP + language_use_span + MLU_spa + Words_Min_spa +
##     VOCD_spa + MLU_eng + Words_Min_eng + VOCD_eng + (1 | subject) +
##     (1 | n_item)
##    Data: priming_dataset_rq3_std
## Control: glmerControl(optimizer = "optimx", optCtrl = list(method = "nlminb"))
##
##      AIC      BIC   logLik deviance df.resid
##  11759.6  11847.7  -5868.8  11737.6    22309
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.6792 -0.2825 -0.1549 -0.0697 16.5958
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  n_item  (Intercept) 3.443    1.856
##  subject (Intercept) 1.053    1.026
## Number of obs: 22320, groups:  n_item, 180; subject, 124
##
## Fixed effects:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -3.13628    0.17281 -18.149   <2e-16 ***
## BLP               -0.24429    0.35631  -0.686    0.493
## language_use_span  0.14608    0.34157   0.428    0.669
## MLU_spa            0.05452    0.12357   0.441    0.659
## Words_Min_spa     -0.06522    0.16321  -0.400    0.689
## VOCD_spa          -0.15593    0.15915  -0.980    0.327
## MLU_eng            0.13362    0.13994   0.955    0.340
## Words_Min_eng     -0.05211    0.21849  -0.238    0.811
## VOCD_eng           0.14496    0.16739   0.866    0.386
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) BLP    lngg__ MLU_sp Wrds_Mn_s VOCD_s MLU_ng Wrds_Mn_n
## BLP          0.003
## langg_s_spn -0.003 -0.848
## MLU_spa     -0.001 -0.046  0.006
## Words_Mn_sp  0.002 -0.164 -0.166 -0.250
## VOCD_spa     0.003 -0.119  0.128 -0.012 -0.369
## MLU_eng     -0.003  0.005  0.045 -0.433  0.134     0.097
## Words_Mn_ng  0.001  0.307  0.101  0.183 -0.637     0.102 -0.282
## VOCD_eng    -0.003 -0.038  0.007 -0.166  0.200    -0.638 -0.169 -0.295
```

```r
# Check the vif score
vif(glm_rq3)
```

```
##               BLP language_use_span           MLU_spa     Words_Min_spa
##         13.875358         12.691955          1.655096          2.860635
##          VOCD_spa           MLU_eng     Words_Min_eng          VOCD_eng
##          2.753717          2.139191          5.202090          3.065345
```

```r
# Perform the generalized linear mixed model fit by maximum likelihood again after remove language_use_
glm_rq3 <- lme4::glmer(target ~ BLP + MLU_spa + Words_Min_spa + VOCD_spa +
                        MLU_eng + Words_Min_eng + VOCD_eng + (1|subject) + (1|n_item),
                     data = priming_dataset_rq3_std, family = "binomial",
                     control = glmerControl(optimizer ='optimx', optCtrl=list(method='nlminb')))
summary(glm_rq3)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: target ~ BLP + MLU_spa + Words_Min_spa + VOCD_spa + MLU_eng +
##     Words_Min_eng + VOCD_eng + (1 | subject) + (1 | n_item)
##    Data: priming_dataset_rq3_std
## Control: glmerControl(optimizer = "optimx", optCtrl = list(method = "nlminb"))
##
##      AIC      BIC   logLik deviance df.resid
##  11757.8  11837.9  -5868.9  11737.8    22310
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.6817 -0.2828 -0.1550 -0.0697 16.5875
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  n_item  (Intercept) 3.443    1.856
##  subject (Intercept) 1.054    1.027
## Number of obs: 22320, groups:  n_item, 180; subject, 124
##
## Fixed effects:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.13616    0.17281 -18.148   <2e-16 ***
## BLP           -0.11514    0.18888  -0.610    0.542
```

```
## MLU_spa        0.05420    0.12360   0.438    0.661
## Words_Min_spa -0.05363    0.16096  -0.333    0.739
## VOCD_spa      -0.16465    0.15794  -1.042    0.297
## MLU_eng        0.13092    0.13981   0.936    0.349
## Words_Min_eng -0.06160    0.21738  -0.283    0.777
## VOCD_eng       0.14452    0.16746   0.863    0.388
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) BLP    MLU_sp Wrds_Mn_s VOCD_s MLU_ng Wrds_Mn_n
## BLP          0.001
## MLU_spa      0.000 -0.077
## Words_Mn_sp  0.002 -0.583 -0.252
## VOCD_spa     0.003 -0.020 -0.013 -0.356
## MLU_eng     -0.003  0.081 -0.434  0.143     0.092
## Words_Mn_ng  0.001  0.745  0.183 -0.632     0.090 -0.288
## VOCD_eng    -0.004 -0.061 -0.166  0.204    -0.644 -0.169 -0.297
```

```
# Check the vif score again
vif(glm_rq3)
```

```
##           BLP       MLU_spa Words_Min_spa      VOCD_spa       MLU_eng
##      3.897342      1.654804      2.780532      2.709804      2.134202
## Words_Min_eng      VOCD_eng
##      5.146430      3.066021
```

```
# Check the model after stepwise variable selection as well
glm_rq3_sig <- lme4::glmer(target ~ BLP + (1|subject) + (1|n_item),
                     data = priming_dataset_rq3_std, family = "binomial",
                     control = glmerControl(optimizer ='optimx', optCtrl=list(method='nlminb')))
summary(glm_rq3_sig)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: target ~ BLP + (1 | subject) + (1 | n_item)
##    Data: priming_dataset_rq3_std
## Control: glmerControl(optimizer = "optimx", optCtrl = list(method = "nlminb"))
##
##      AIC      BIC   logLik deviance df.resid
##  11749.9  11782.0  -5871.0  11741.9    22316
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.6986 -0.2827 -0.1550 -0.0697 16.6706
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  n_item  (Intercept) 3.445    1.856
##  subject (Intercept) 1.093    1.045
## Number of obs: 22320, groups:  n_item, 180; subject, 124
##
```

```
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.13627    0.17372 -18.054   <2e-16 ***
## BLP         -0.18770    0.09729  -1.929   0.0537 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##     (Intr)
## BLP 0.005
```

Power Analysis

```r
# Calculate Pseudo-R-squared for Generalized Mixed-Effect models
MuMIn::r.squaredGLMM(glm_rq3)
```

```
## Warning: the null model is correct only if all variables used by the original
## model remain unchanged.
```

```
##                     R2m       R2c
## theoretical 0.009438995 0.5815094
## delta       0.005638479 0.3473705
```

```r
# Calculate the power for RQ3
pwr.f2.test(u = 7, v =  124-7-1, f2 = 0.005760835/(1-0.005760835), sig.level = 0.05)
```

```
##
##      Multiple regression power calculation
##
##               u = 7
##               v = 116
##              f2 = 0.005794215
##       sig.level = 0.05
##           power = 0.07588516
```