Confirmatory Hypothesis Testing

Erin M. Buchanan

Last Update 2022-03-09

Libraries

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(psych)
library(tidyr)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

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

```
library(rio)
library(faux)
```

```
##
## ************
## Welcome to faux. For support and examples visit:
## https://debruine.github.io/faux/
## - Get and set global package options with: faux_options()
## ************
```

```
set.seed(58902)
library(nlme)
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
library(MuMIn)
```

Import the Files

We will import these files from our data folder for the final analyses. This example analysis examines the Semantic Priming Project data to simulate how to answer questions. s.

```
# targets_only <- read.csv("../05_Data/output_data/prime_data.csv")
# targets_only_no2.5 <- read.csv("../05_Data/output_data/prime_data_no2.5.csv")
# targets_only_no3.0 <- read.csv("../05_Data/output_data/prime_data_no3.0.csv")

SPP <- import("spp_ldt_prime.xlsx")
```

Languages

This demonstration with the SPP only has English data. However, we will have 10 languages for our completed data. Therefore, we will simulate data that is nearly similar for 10 other languages (no heterogeneity) and different for the other 10 languages (heterogeneity).

```
# use our names so we can update this code easier
sim_data <- SPP %>%
  select(long_1st_rel, long_1st_un, firstassoc_fas, target) %>%
  dplyr::rename(avgZ_RT_related = long_1st_rel) %>%
  dplyr::rename(avgZ_RT_unrelated = long_1st_un) %>%
  mutate(Language = "English") %>%
  sample_n(1000) %>%
  mutate(samplesize_related = round(rnorm(1000, mean = 75, sd = 10)),
         samplesize_unrelated = round(rnorm(1000, mean = 75, sd = 10)),
         seZ_RT_related = round(rnorm(1000, mean = .09, sd = .02), digits = 3),
         seZ_RT_unrelated = round(rnorm(1000, mean = .09, sd = .02), digits = 3))

# build fake data
lang2 <- sim_df(data = sim_data, n = nrow(sim_data), id = "target")
lang3 <- sim_df(data = sim_data, n = nrow(sim_data), id = "target")
lang4 <- sim_df(data = sim_data, n = nrow(sim_data), id = "target")
lang5 <- sim_df(data = sim_data, n = nrow(sim_data), id = "target")
lang6 <- sim_df(data = sim_data, n = nrow(sim_data), id = "target")
lang7 <- sim_df(data = sim_data, n = nrow(sim_data), id = "target")
lang8 <- sim_df(data = sim_data, n = nrow(sim_data), id = "target")
lang9 <- sim_df(data = sim_data, n = nrow(sim_data), id = "target")
lang10 <- sim_df(data = sim_data, n = nrow(sim_data), id = "target")

lang2$target <- lang3$target <- lang4$target <- lang5$target <- lang6$target <- lang7$target <- lang8$target <- lang9$target <- lang10$target <- s

lang2$Language <- "Language 2"
lang3$Language <- "Language 3"
lang4$Language <- "Language 4"
lang5$Language <- "Language 5"
lang6$Language <- "Language 6"
lang7$Language <- "Language 7"
lang8$Language <- "Language 8"
lang9$Language <- "Language 9"
lang10$Language <- "Language 10"

sim_data_same <- rbind(sim_data, lang2, lang3, lang4, lang5, lang6, lang7, lang8, lang9, lang10) %>%
  mutate(samplesize_related = round(samplesize_related),
         samplesize_unrelated = round(samplesize_unrelated),
         avgZ_prime = avgZ_RT_unrelated - avgZ_RT_related)

tapply(sim_data_same$avgZ_prime, sim_data_same$Language, mean)
```

```
##     English Language 10  Language 2  Language 3  Language 4  Language 5
##   0.1760220   0.1805976   0.1719426   0.1808329   0.1835229   0.1636394
##  Language 6  Language 7  Language 8  Language 9
##   0.1672451   0.1707778   0.1673586   0.1771468
```

```
sim_data_diff <- sim_data_same %>%
  group_by(Language) %>%
  mutate(avgZ_prime = avgZ_prime + rnorm(1, mean = 0, sd = .20))

tapply(sim_data_diff$avgZ_prime, sim_data_diff$Language, mean)
```

```
##     English Language 10  Language 2  Language 3  Language 4  Language 5
##  0.13245182  0.30049332  0.15599814 -0.15040381  0.48028796  0.25218827
##  Language 6  Language 7  Language 8  Language 9
## -0.18076795  0.00728217 -0.05345230  0.33511100
```

Hypothesis 1 - Languages Same

Is semantic priming a non-zero effect?

```
hyp1 <- lm(avgZ_prime ~ 1, data = sim_data_same)
summary(hyp1)
```

```
##
## Call:
## lm(formula = avgZ_prime ~ 1, data = sim_data_same)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33860 -0.18791  0.00041  0.18681  1.02206
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.173909   0.002805   61.99   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2805 on 9999 degrees of freedom
```

```
confint(hyp1)
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.1684095 0.1794076
```

Hypothesis 2 - Languages Same

Is semantic priming consistent by language?

```
hyp2 <- lme(avgZ_prime ~ 1,
            data = sim_data_same,
            method = "REML",
            random = ~1|Language)
summary(hyp2)
```

```
## Linear mixed-effects model fit by REML
##   Data: sim_data_same
##        AIC      BIC    logLik
##   2972.507 2994.138 -1483.253
##
## Random effects:
##  Formula: ~1 | Language
##          (Intercept)  Residual
## StdDev: 1.211236e-05 0.2805344
##
## Fixed effects:  avgZ_prime ~ 1
##                 Value   Std.Error   DF  t-value p-value
## (Intercept) 0.1739086 0.002805347 9990 61.99183       0
##
## Standardized Within-Group Residuals:
##          Min           Q1          Med           Q3          Max
## -4.771592857 -0.669843820  0.001458942  0.665903668  3.643275295
##
## Number of Observations: 10000
## Number of Groups: 10
```

```
intervals(hyp2)
```

```
## Approximate 95% confidence intervals
##
##  Fixed effects:
##                 lower      est.     upper
## (Intercept) 0.1684095 0.1739086 0.1794076
## attr(,"label")
## [1] "Fixed effects:"
##
##  Random Effects:
##   Level: Language
##                          lower         est.       upper
## sd((Intercept)) 6.108095e-21 1.211236e-05 24018812699
##
##  Within-group standard error:
##     lower      est.     upper
## 0.2766731 0.2805344 0.2844496
```

```
AIC(hyp1)
```

```
## [1] 2960.592
```

```
AIC(hyp2)
```

```
## [1] 2972.507
```

```
r.squaredGLMM(hyp2)
```

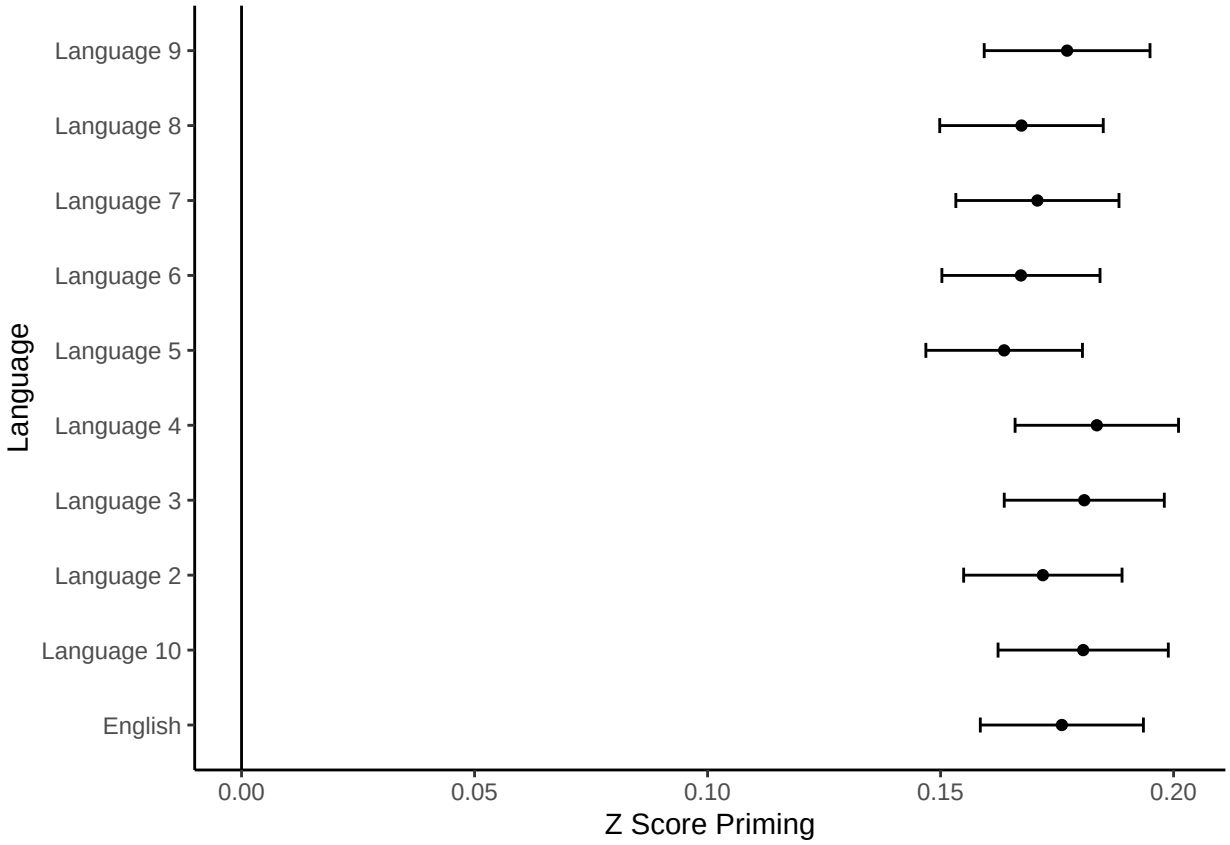
```
## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help page.
```

```
##      R2m          R2c
## [1,]   0 1.864168e-09
```

```
# example plot
ggplot(sim_data_same, aes(Language, avgZ_prime)) +
  stat_summary(fun.data = mean_cl_normal,
               geom = "errorbar",
```

```
              width = .2,
              position = position_dodge(width = .2)) +
  stat_summary(fun = mean,
              geom = "point",
              position = position_dodge()) +
  xlab("Language") +
  ylab("Z Score Priming") +
  theme_classic() +
  coord_flip() +
  geom_hline(yintercept = 0)
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```



Hypothesis 1 - Languages Different

Is semantic priming a non-zero effect?

```
hyp1 <- lm(avgZ_prime ~ 1, data = sim_data_diff)
summary(hyp1)
```

```
##
## Call:
## lm(formula = avgZ_prime ~ 1, data = sim_data_diff)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3198 -0.2379 -0.0015  0.2401  1.2120
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.127919   0.003493   36.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3493 on 9999 degrees of freedom
```

```
confint(hyp1)
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.1210723 0.1347654
```

Hypothesis 2 - Languages Different

Is semantic priming consistent by language?

```
hyp2 <- lme(avgZ_prime ~ 1,
            data = sim_data_diff,
            method = "REML",
            random = ~1|Language)
summary(hyp2)
```

```
## Linear mixed-effects model fit by REML
##   Data: sim_data_diff
##        AIC      BIC    logLik
##   3034.088 3055.719 -1514.044
##
## Random effects:
##  Formula: ~1 | Language
##         (Intercept)  Residual
## StdDev:   0.2192456 0.2805882
##
## Fixed effects:  avgZ_prime ~ 1
##                 Value  Std.Error   DF  t-value p-value
## (Intercept) 0.1279189 0.06938829 9990 1.843522  0.0653
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -4.78101074 -0.67347272  0.00153643  0.66719029  3.61974274
##
## Number of Observations: 10000
## Number of Groups: 10
```

```
intervals(hyp2)
```

```
## Approximate 95% confidence intervals
##
##  Fixed effects:
##                     lower      est.     upper
## (Intercept) -0.008096171 0.1279189 0.2639339
## attr(,"label")
## [1] "Fixed effects:"
##
##  Random Effects:
##   Level: Language
##                     lower      est.     upper
## sd((Intercept)) 0.1382992 0.2192456 0.3475697
##
##  Within-group standard error:
##     lower      est.     upper
## 0.2767245 0.2805882 0.2845060
```

```
AIC(hyp1)
```

```
## [1] 7344.088
```

```
AIC(hyp2)
```

```
## [1] 3034.088
```

```
r.squaredGLMM(hyp2)
```

```
##      R2m       R2c
## [1,]   0 0.3790949
```

```
# example plot
ggplot(sim_data_diff, aes(Language, avgZ_prime)) +
  stat_summary(fun.data = mean_cl_normal,
               geom = "errorbar",
               width = .2,
               position = position_dodge(width = .2)) +
  stat_summary(fun = mean,
               geom = "point",
               position = position_dodge()) +
  xlab("Language") +
  ylab("Z Score Priming") +
  theme_classic() +
```

```
  coord_flip() +
  geom_hline(yintercept = 0)
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```