Repeated Lexical Retrieval: Experiment 2

Abhilasha Kumar

June 6, 2019

1 Reading File

```
> PrimeRetrieval <- read.csv("E3_YA_Responses.csv",
+                            header = TRUE, sep = ",")
> library(dplyr)
> PrimeRetrieval = PrimeRetrieval %>% filter(AgeGroup == "Young")
```

2 LME

```
> library(lme4)
> contrasts(PrimeRetrieval$PrimeCondition)= contr.treatment(4, base = 4)
> prime_lmer2 = glmer(data = PrimeRetrieval,
+                     Accuracy ~ PrimeCondition +
+                       (1|Subject) + (1|Stimuli2),
+                            family = "binomial",
+                            control=glmerControl(optimizer="bobyqa",
+              optCtrl=list(maxfun=100000)))
> summary(prime_lmer2)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: Accuracy ~ PrimeCondition + (1 | Subject) + (1 | Stimuli2)
   Data: PrimeRetrieval
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))

     AIC      BIC   logLik deviance df.resid
  3634.6   3671.5  -1811.3   3622.6     3450

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.1337 -0.5810 -0.3166  0.6242  6.6789

Random effects:
 Groups   Name        Variance Std.Dev.
```

```
 Stimuli2 (Intercept) 1.8714   1.3680
 Subject  (Intercept) 0.5131   0.7163
Number of obs: 3456, groups:  Stimuli2, 72; Subject, 48

Fixed effects:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     -0.77392    0.21007  -3.684  0.00023 ***
PrimeCondition1 -0.25640    0.11965  -2.143  0.03212 *
PrimeCondition2 -0.07342    0.11869  -0.619  0.53616
PrimeCondition3 -0.15953    0.11951  -1.335  0.18192
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Correlation of Fixed Effects:
            (Intr) PrmCn1 PrmCn2
PrimeCndtn1 -0.278
PrimeCndtn2 -0.281  0.495
PrimeCndtn3 -0.278  0.491  0.497
```

```
>
> # confint(prime_lmer2)
> #
> # > confint(prime_lmer2)
> # Computing profile confidence intervals ...
> #                      2.5 %      97.5 %
> # .sig01           1.1418314  1.66248456
> # .sig02           0.5641225  0.92149953
> # (Intercept)     -1.1916777 -0.36114974
> # PrimeCondition1 -0.4950192 -0.01945911
> # PrimeCondition2 -0.3090286  0.16187213
> # PrimeCondition3 -0.3970849  0.07727542
```

3 Prime And Target Accuracy

```
> library(dplyr)
> agg_condition ← group_by(PrimeRetrieval, PrimeCondition)%>%
+     summarise_at(vars(Accuracy), mean)
> agg_sub_condition ←  group_by(PrimeRetrieval, Subject, PrimeCondition)%>%
+     summarise_at(vars(Accuracy), mean)
> agg_sub_condition$Subject ← as.factor(agg_sub_condition$Subject)
> agg_sub_condition$PrimeCondition ← as.factor(agg_sub_condition$PrimeCondition)
> agg_sub_prime = group_by(PrimeRetrieval, Subject, PrimeCondition) %>%
+    summarise_at(vars(PrimeFirstResp_ACC), mean)
> ## target accuracy anova
>
> prime_aov = aov(data = agg_sub_condition, Accuracy ~ PrimeCondition +
```

```
+                                         Error(Subject/PrimeCondition))
> summary(prime_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 47  2.865 0.06097

Error: Subject:PrimeCondition
                Df Sum Sq Mean Sq F value  Pr(>F)
PrimeCondition   3 0.0485 0.01616   1.058   0.369
Residuals      141 2.1537 0.01527
```

```
> ## prime accuracy anova
> agg_sub_prime$Subject = as.factor(agg_sub_prime$Subject)
> primeaccuracy_aov = aov(data = agg_sub_prime,
+                         PrimeFirstResp_ACC ~ PrimeCondition +
+                                         Error(Subject/PrimeCondition))
> summary(primeaccuracy_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 47  2.068 0.04401

Error: Subject:PrimeCondition
                Df Sum Sq Mean Sq F value   Pr(>F)
PrimeCondition   3 0.6128 0.20426   15.53 8.81e-09 ***
Residuals      141 1.8548 0.01315
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> ## specific t-tests
> prime_p = agg_sub_prime %>% filter(PrimeCondition == "P")
> prime_r = agg_sub_prime %>% filter(PrimeCondition == "R")
> prime_b = agg_sub_prime %>% filter(PrimeCondition == "B")
> prime_u = agg_sub_prime %>% filter(PrimeCondition == "U")
> t.test(prime_p$PrimeFirstResp_ACC, prime_r$PrimeFirstResp_ACC, paired = TRUE)
```

```
        Paired t-test

data:  prime_p$PrimeFirstResp_ACC and prime_r$PrimeFirstResp_ACC
t = 4.9091, df = 47, p-value = 1.143e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.0689931 0.1648032
sample estimates:
mean of the differences
              0.1168981
```

```
> t.test(prime_p$PrimeFirstResp_ACC, prime_b$PrimeFirstResp_ACC, paired = TRUE)
```

```
        Paired t-test

data:  prime_p$PrimeFirstResp_ACC and prime_b$PrimeFirstResp_ACC
t = 6.5065, df = 47, p-value = 4.589e-08
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1055408 0.2000148
sample estimates:
mean of the differences
              0.1527778
```

```
> t.test(prime_p$PrimeFirstResp_ACC, prime_u$PrimeFirstResp_ACC, paired = TRUE)
```

```
        Paired t-test

data:  prime_p$PrimeFirstResp_ACC and prime_u$PrimeFirstResp_ACC
t = 4.0477, df = 47, p-value = 0.0001917
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.04599164 0.13687873
sample estimates:
mean of the differences
             0.09143519
```

```
> t.test(prime_b$PrimeFirstResp_ACC, prime_r$PrimeFirstResp_ACC, paired = TRUE)
```

```
        Paired t-test

data:  prime_b$PrimeFirstResp_ACC and prime_r$PrimeFirstResp_ACC
t = -1.83, df = 47, p-value = 0.07359
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.075322360  0.003563101
sample estimates:
mean of the differences
            -0.03587963
```

```
> t.test(prime_b$PrimeFirstResp_ACC, prime_u$PrimeFirstResp_ACC, paired = TRUE)
```

```
        Paired t-test

data:  prime_b$PrimeFirstResp_ACC and prime_u$PrimeFirstResp_ACC
t = -2.3122, df = 47, p-value = 0.0252
alternative hypothesis: true difference in means is not equal to 0
```

```
95 percent confidence interval:
 -0.114714633 -0.007970552
sample estimates:
mean of the differences
            -0.06134259
```

```
> t.test(prime_r$PrimeFirstResp_ACC, prime_u$PrimeFirstResp_ACC, paired = TRUE)
```

```
        Paired t-test

data:  prime_r$PrimeFirstResp_ACC and prime_u$PrimeFirstResp_ACC
t = -1.0649, df = 47, p-value = 0.2924
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.07356625  0.02264032
sample estimates:
mean of the differences
            -0.02546296
```

```
> ## accounting for mean participant accuracy on the prime?
>
> participant_acc = group_by(PrimeRetrieval, Subject) %>%
+   summarise_at(vars(Accuracy, PrimeFirstResp_ACC), mean)
> participant_acc$MeanAcc = (participant_acc$Accuracy +
+                            participant_acc$PrimeFirstResp_ACC)/2
> median_acc = median(participant_acc$MeanAcc)
> colnames(participant_acc) = c("Subject", "TargetAcc", "PrimeAcc", "MeanAcc")
> ## accounting for mean prime accuracy
>
> item_acc = group_by(PrimeRetrieval, Stimuli2, PrimeCondition) %>%
+   summarise_at(vars(PrimeFirstResp_ACC), mean)
> colnames(item_acc) = c("Stimuli2","PrimeCondition","PrimeAcc")
> PrimeRetrieval = merge(PrimeRetrieval, item_acc,
+                        by = c("Stimuli2", "PrimeCondition"))
> PrimeRetrieval2 = merge(PrimeRetrieval, participant_acc[,c(1,3,4)],
+                        by = c("Subject"))
```

3.1 Using lmer

```
> #  since finaldata has several missing trials -- need 704 have 662, ANOVA is probably
>
> contrasts(PrimeRetrieval$PrimeCondition) = contr.treatment(4, base = 4)
> PrimeRetrieval$PrimeFirstResp_ACC = as.factor(PrimeRetrieval$PrimeFirstResp_ACC)
> m_young_prime = glmer(data = PrimeRetrieval, Accuracy ~
+                         PrimeFirstResp_ACC*PrimeCondition + PrimeAcc +
+                         (1|Subject) + (1|Stimuli2),
+                       family = "binomial",
```

```
+                             control=glmerControl(optimizer="bobyqa",
+               optCtrl=list(maxfun=100000)))
> summary(m_young_prime)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: Accuracy ~ PrimeFirstResp_ACC * PrimeCondition + PrimeAcc + (1 |
    Subject) + (1 | Stimuli2)
   Data: PrimeRetrieval
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))

     AIC      BIC   logLik deviance df.resid
  3624.6   3692.2  -1801.3   3602.6     3445

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.0435 -0.5758 -0.3106  0.6261  5.3680

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli2 (Intercept) 1.803    1.3428
 Subject  (Intercept) 0.473    0.6877
Number of obs: 3456, groups:  Stimuli2, 72; Subject, 48

Fixed effects:
                                    Estimate Std. Error z value Pr(>|z|)
(Intercept)                         -0.74421    0.23661  -3.145  0.00166 **
PrimeFirstResp_ACC1                  0.05991    0.17359   0.345  0.72999
PrimeCondition1                     -0.54380    0.17284  -3.146  0.00165 **
PrimeCondition2                     -0.08893    0.17923  -0.496  0.61978
PrimeCondition3                     -0.41195    0.17348  -2.375  0.01757 *
PrimeAcc                            -0.10312    0.20631  -0.500  0.61719
PrimeFirstResp_ACC1:PrimeCondition1  0.60408    0.25594   2.360  0.01826 *
PrimeFirstResp_ACC1:PrimeCondition2  0.03717    0.25465   0.146  0.88394
PrimeFirstResp_ACC1:PrimeCondition3  0.50539    0.25337   1.995  0.04608 *
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Correlation of Fixed Effects:
             (Intr) PrFR_ACC1 PrmCn1 PrmCn2 PrmCn3 PrmAcc PFR_ACC1:PC1
PrmFrR_ACC1  -0.288
PrimeCndtn1  -0.387  0.451
PrimeCndtn2  -0.346  0.447     0.457
PrimeCndtn3  -0.372  0.449     0.473  0.448
PrimeAcc     -0.354 -0.154     0.141  0.060  0.111
PFR_ACC1:PC1  0.300 -0.640    -0.719 -0.329 -0.341 -0.175
PFR_ACC1:PC2  0.305 -0.640    -0.356 -0.744 -0.342 -0.193  0.501
```

```
PFR_ACC1:PC3   0.296  -0.640     -0.349  -0.322  -0.722  -0.168   0.492
                PFR_ACC1:PC2
PrmFrR_ACC1
PrimeCndtn1
PrimeCndtn2
PrimeCndtn3
PrimeAcc
PFR_ACC1:PC1
PFR_ACC1:PC2
PFR_ACC1:PC3   0.493
```

```
> options(contrasts = c("contr.sum","contr.poly"))
> car::Anova(m_young_prime)
```

```
Analysis of Deviance Table (Type II Wald chisquare tests)

Response: Accuracy
                                      Chisq Df  Pr(>Chisq)
PrimeFirstResp_ACC                   8.4535  1    0.003643 **
PrimeCondition                       4.6254  3    0.201370
PrimeAcc                             0.2498  1    0.617186
PrimeFirstResp_ACC:PrimeCondition    9.0422  3    0.028735 *
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> anova(m_young_prime)
```

```
Analysis of Variance Table
                                   Df  Sum Sq Mean Sq F value
PrimeFirstResp_ACC                  1 11.5732 11.5732 11.5732
PrimeCondition                      3  4.6078  1.5359  1.5359
PrimeAcc                            1  0.0264  0.0264  0.0264
PrimeFirstResp_ACC:PrimeCondition   3  9.1290  3.0430  3.0430
```

```
> # > confint(m_young_prime) unrelated and primeret0 as baseline
> # Computing profile confidence intervals ...
> #                                          2.5 %        97.5 %
> # .sig01                               1.1189433   1.633996424
> # .sig02                               0.5387244   0.888068921
> # (Intercept)                         -1.1652470  -0.267459539
> # PrimeFirstResp_ACC1                 -0.2023619   0.142159609
> # PrimeCondition1                     -0.4804850  -0.004245685
> # PrimeCondition2                     -0.3087080   0.167440999
> # PrimeCondition3                     -0.3977035   0.078274447
> # PrimeAcc                            -0.5127288   0.311643685
> # PrimeFirstResp_ACC1:PrimeCondition1 -0.5565369  -0.048968479
> # PrimeFirstResp_ACC1:PrimeCondition2 -0.2714417   0.233790347
```

```
> # PrimeFirstResp_ACC1:PrimeCondition3 -0.5042746 -0.002000942
>
> m_young_prime2 = lme4::glmer(data = PrimeRetrieval, Accuracy ~
+                              PrimeFirstResp_ACC*PrimeCondition +
+                              (1|Subject) + (1|Stimuli2),
+                             family = "binomial",
+                             control=glmerControl(optimizer="bobyqa",
+             optCtrl=list(maxfun=100000)))
> summary(m_young_prime2)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: Accuracy ~ PrimeFirstResp_ACC * PrimeCondition + (1 | Subject) +
    (1 | Stimuli2)
   Data: PrimeRetrieval
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))

     AIC      BIC   logLik deviance df.resid
  3622.9   3684.3  -1801.4   3602.9     3446

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.0687 -0.5779 -0.3095  0.6294  5.4175

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli2 (Intercept) 1.7910   1.338
 Subject  (Intercept) 0.4761   0.690
Number of obs: 3456, groups:  Stimuli2, 72; Subject, 48

Fixed effects:
                                    Estimate Std. Error z value Pr(>|z|)
(Intercept)                         -0.76268    0.20531  -3.715 0.000203 ***
PrimeFirstResp_ACC1                 -0.02328    0.08568  -0.272 0.785888
PrimeCondition1                     -0.24100    0.12007  -2.007 0.044725 *
PrimeCondition2                     -0.07732    0.11904  -0.650 0.515991
PrimeCondition3                     -0.16052    0.12000  -1.338 0.180994
PrimeFirstResp_ACC1:PrimeCondition1 -0.29089    0.12590  -2.310 0.020863 *
PrimeFirstResp_ACC1:PrimeCondition2 -0.00619    0.12485  -0.050 0.960454
PrimeFirstResp_ACC1:PrimeCondition3 -0.24211    0.12485  -1.939 0.052481 .
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Correlation of Fixed Effects:
            (Intr) PrFR_ACC1 PrmCn1 PrmCn2 PrmCn3 PFR_ACC1:PC1 PFR_ACC1:PC2
PrmFrR_ACC1 -0.018
PrimeCndtn1 -0.279  0.031
```

```
PrimeCndtn2   -0.284  0.030      0.486
PrimeCndtn3   -0.280  0.031      0.482  0.487
PFR_ACC1:PC1   0.008 -0.686     -0.034 -0.020 -0.024
PFR_ACC1:PC2   0.011 -0.691     -0.025  0.073 -0.023  0.483
PFR_ACC1:PC3   0.009 -0.684     -0.026 -0.023 -0.009  0.476         0.475
```

```
> car::Anova(m_young_prime2)
```

```
Analysis of Deviance Table (Type II Wald chisquare tests)

Response: Accuracy
                                   Chisq Df Pr(>Chisq)
PrimeFirstResp_ACC               11.2801  1  0.0007834 ***
PrimeCondition                    4.6235  3  0.2015394
PrimeFirstResp_ACC:PrimeCondition  8.8303  3  0.0316346 *
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> anova(m_young_prime, m_young_prime2) ## prime acc not needed
```

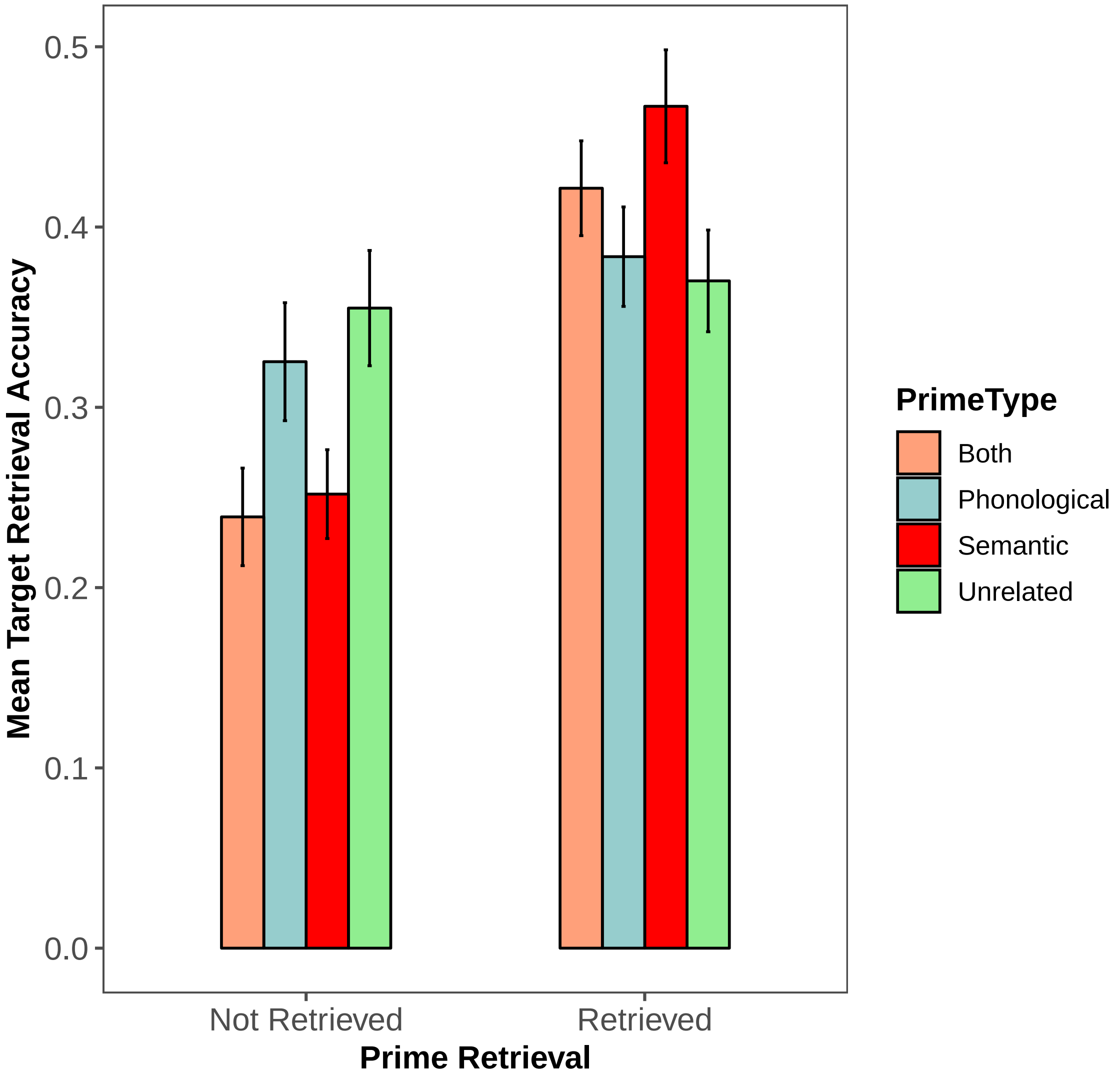
```
Data: PrimeRetrieval
Models:
m_young_prime2: Accuracy ~ PrimeFirstResp_ACC * PrimeCondition + (1 | Subject) +
m_young_prime2:     (1 | Stimuli2)
m_young_prime: Accuracy ~ PrimeFirstResp_ACC * PrimeCondition + PrimeAcc + (1 |
m_young_prime:     Subject) + (1 | Stimuli2)
               Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
m_young_prime2 10 3622.9 3684.3 -1801.4   3602.9
m_young_prime  11 3624.6 3692.2 -1801.3   3602.6 0.2443      1     0.6211
```

Figures: Mean Accuracy

Retrieval by Primes

```
> agg_sub_condition_retrieval = group_by(PrimeRetrieval, Subject,
+                                        PrimeCondition, PrimeFirstResp_ACC) %>%
+                                       summarise_at(vars(Accuracy), mean)
> agg_acc_retrieval = Rmisc::summarySE(agg_sub_condition_retrieval,
+                          measurevar = "Accuracy",
+                          groupvars = c("PrimeCondition", "PrimeFirstResp_ACC"))
> library(ggplot2)
> library(ggthemes)
>  agg_acc_retrieval %>% mutate(PrimeType = factor(PrimeCondition,
+                                                  levels = unique(PrimeCondition),
+                        labels = c("Both", "Phonological",
+                                   "Semantic", "Unrelated")),
+                         `Prime Retrieval` = factor(PrimeFirstResp_ACC,
```

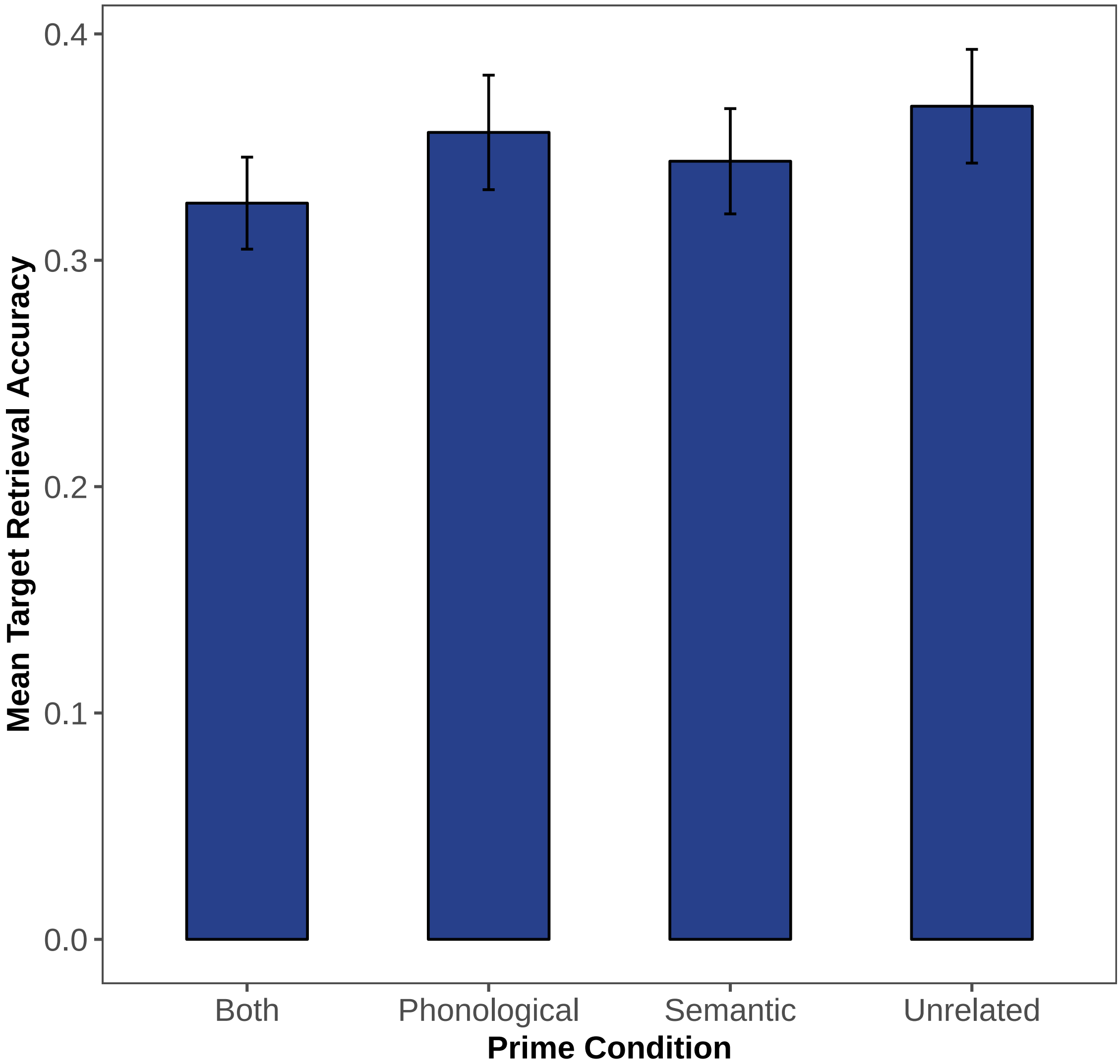
```
+                                  levels = unique(PrimeFirstResp_ACC),
+                      labels = c("Not Retrieved", "Retrieved")))%>%
+   ggplot(aes(x = `Prime Retrieval`, y = Accuracy,
+              fill = PrimeType)) +
+  geom_bar(stat = "identity", position = "dodge", width = 0.5, color = "black")+
+    geom_errorbar(aes(ymin = Accuracy - se, ymax = Accuracy + se),
+                 width=.05, position=position_dodge(.5)) +
+     theme_few()+
+    xlab("Prime Retrieval") + ylab("Mean Target Retrieval Accuracy") +
+   ggtitle("")  +
+     scale_fill_manual(values = c( "lightsalmon",
+                                 "paleturquoise3","red","lightgreen"))+
+    theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
>
```



Target

```
> agg_acc = Rmisc::summarySE(agg_sub_condition,
+                       measurevar = "Accuracy",
+                       groupvars = c("PrimeCondition"))
> library(ggplot2)
> library(ggthemes)
>  agg_acc %>% mutate(PrimeType = factor(PrimeCondition,
+                                                    levels = unique(PrimeCondition),
+                     labels = c("Both", "Phonological",
+                                "Semantic", "Unrelated")))%>%
```

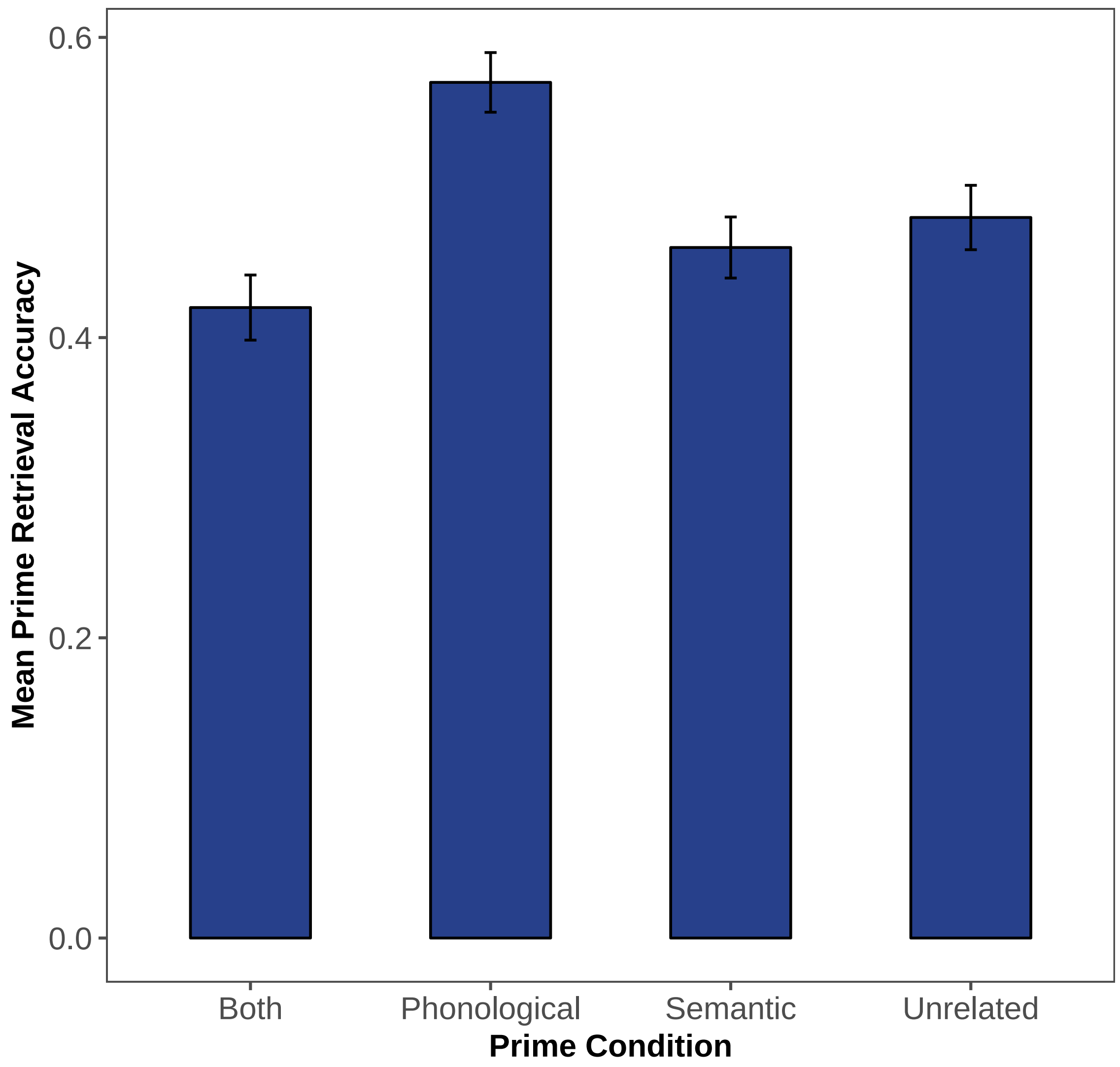
```
+    ggplot(aes(x = PrimeType, y = Accuracy)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            fill = "royalblue4", color = "black")+
+     geom_errorbar(aes(ymin = Accuracy - se, ymax = Accuracy + se),
+                   width=.05, position=position_dodge(.5)) +
+      theme_few()+
+     xlab("Prime Condition") + ylab("Mean Target Retrieval Accuracy") +
+    ggtitle("")  +
+     theme(axis.text = element_text(size = rel(1)),
+            axis.title = element_text(face = "bold", size = rel(1)),
+            legend.title = element_text(face = "bold", size = rel(1)),
+           plot.title = element_text(hjust = .5),
+           strip.text.x = element_text(face = "bold", size = rel(1.4)))
>
```



Prime

```
> agg_prime_acc = Rmisc::summarySE(agg_sub_prime,
+                       measurevar = "PrimeFirstResp_ACC",
+                       groupvars = c("PrimeCondition"))
> agg_prime_acc$PrimeFirstResp_ACC = round(agg_prime_acc$PrimeFirstResp_ACC,
+                                       digits = 2)
> library(ggplot2)
> library(ggthemes)
>  agg_prime_acc %>% mutate(PrimeType = factor(PrimeCondition,
+                                      levels = unique(PrimeCondition),
```

```
+                     labels = c("Both", "Phonological",
+                                "Semantic", "Unrelated")))%>%
+   ggplot(aes(x = PrimeType, y = PrimeFirstResp_ACC)) +
+  geom_bar(stat = "identity", position = "dodge", width = 0.5,
+           fill = "royalblue4", color = "black")+
+    geom_errorbar(aes(ymin = PrimeFirstResp_ACC - se,
+                      ymax = PrimeFirstResp_ACC + se),
+                 width=.05, position=position_dodge(.5)) +
+     theme_few()+
+    xlab("Prime Condition") + ylab("Mean Prime Retrieval Accuracy") +
+   ggtitle("")  +
+    theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
>
```



State Data

```
> state_prime_counts = group_by(PrimeRetrieval,Subject,
+                               PrimeCondition, TargetQuestion) %>%
+   summarise(Trials = n())
> state_prime = Rmisc::summarySE(state_prime_counts,
+                        measurevar = "Trials",
+                        groupvars = c("PrimeCondition", "TargetQuestion"))
> library(ggplot2)
> library(ggthemes)
```

```
> state_prime %>% mutate(PrimeType = factor(PrimeCondition,
+                     levels = unique(PrimeCondition),
+                     labels = c("Both", "Phonological",
+                                "Semantic", "Unrelated")),
+   State1 = factor(TargetQuestion, levels = unique(TargetQuestion),
+                          labels = c("Know", "Dont Know",
+                                     "Other", "TOT")))%>%
+   ggplot(aes(x = PrimeType, y = Trials, fill = State1))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.5)+
+   geom_errorbar(aes(ymin = Trials - ci, ymax = Trials + ci),
+                 width=.05, position=position_dodge(.5)) +
+   scale_fill_colorblind()+
+   theme_few()+
+   xlab("Prime Condition") + ylab("Mean Number of Trials") +
+   ggtitle("YA: Target Retrieval States by Prime Condition")+
+   theme(axis.text = element_text(face = "bold", size = rel(1.2)),
+           axis.title = element_text(face = "bold", size = rel(1.2)),
+           legend.title = element_text(face = "bold", size = rel(1.2)),
+                    strip.text.x = element_text(face = "bold", size = rel(1.4)),
+       plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
>
```

3.2 Percentage State Prime Analysis

```
> state = read.csv("YAOA_agg_FINAL.csv",header = TRUE, sep = ",")
> state = state %>% filter(Age == "Young")
> j_statepercent = state[,c(2,3,160:175)] # use for prime percents
> j_statepercent$Subject = as.factor(j_statepercent$Subject)
> library(tidyr)
> library(dplyr)
> statepercent <- j_statepercent %>%
+   gather(PrimeState, Percent,
+          prop_r_know, prop_r_dontknow, prop_r_other, prop_r_TOT,
+          prop_p_know, prop_p_dontknow, prop_p_other, prop_p_TOT,
+          prop_b_know, prop_b_dontknow, prop_b_other, prop_b_TOT,
+          prop_u_know, prop_u_dontknow, prop_u_other, prop_u_TOT) %>%
+   separate(PrimeState, c('Prop', 'Prime', 'State'), sep = "_") %>%
+   arrange(Subject)
> #removing prop
> statepercent = statepercent[,-3]
> colnames(statepercent) = c( "Subject","AgeGroup",
+                            "PrimeCondition", "State", "Percent")
> statepercent$AgeGroup <- as.factor(statepercent$AgeGroup)
> statepercent$Subject <- as.factor(statepercent$Subject)
> statepercent$PrimeCondition <- as.factor(statepercent$PrimeCondition)
> statepercent$State <- as.factor(statepercent$State)
> statepercent$Percent <- as.numeric(as.character(statepercent$Percent))
```

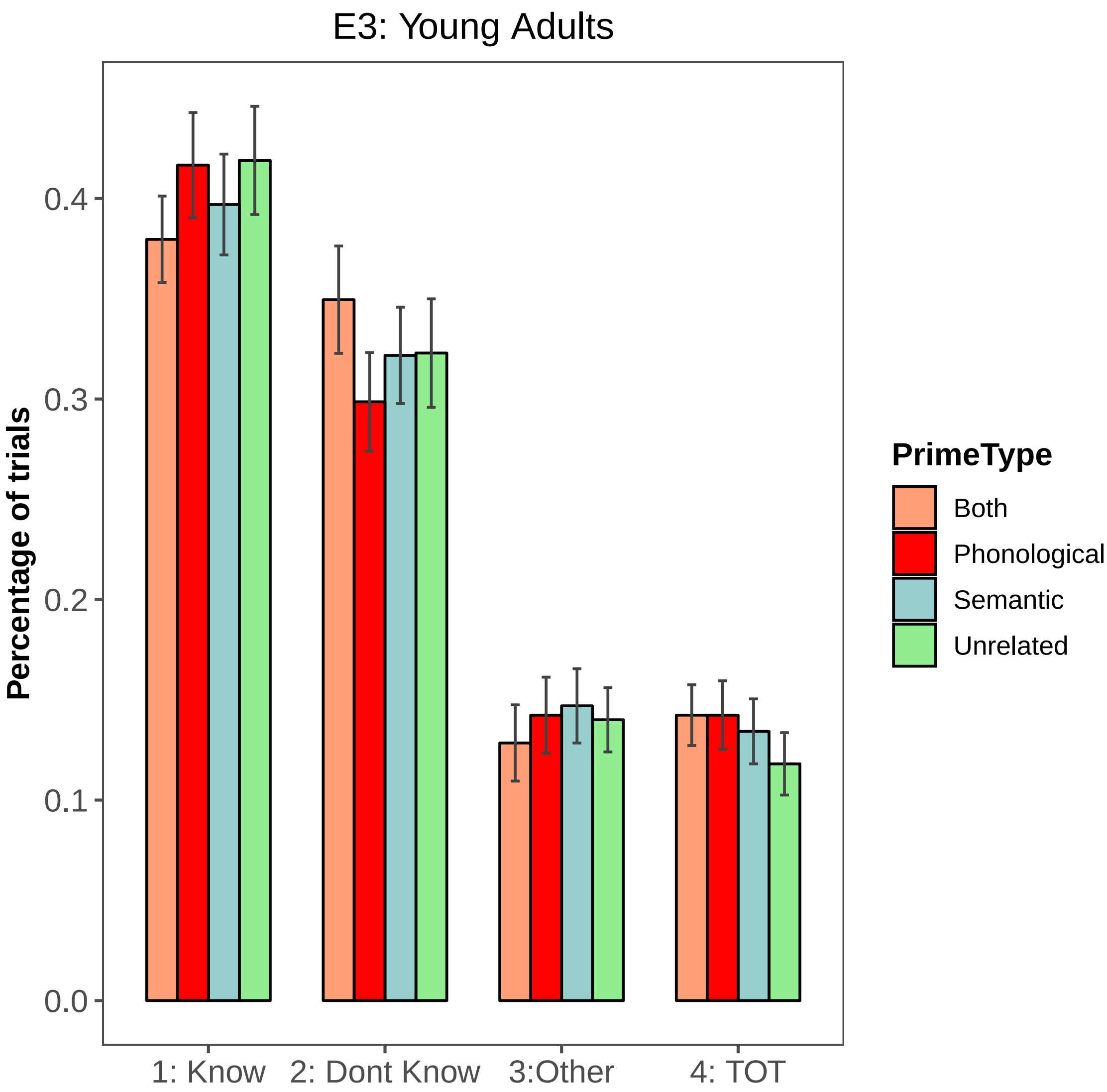
```
> ## anova
>
> state_aov = aov(data = statepercent, Percent ~ PrimeCondition*State +
+                   Error(Subject/(PrimeCondition*State)))
> summary(state_aov)
```

```
Error: Subject
          Df    Sum Sq   Mean Sq F value Pr(>F)
Residuals 47 3.395e-18 7.222e-20

Error: Subject:PrimeCondition
                Df    Sum Sq   Mean Sq F value Pr(>F)
PrimeCondition   3 1.500e-19 4.991e-20   0.684  0.563
Residuals      141 1.029e-17 7.296e-20

Error: Subject:State
           Df Sum Sq Mean Sq F value Pr(>F)
State       3  10.45   3.482   44.48 <2e-16 ***
Residuals 141  11.04   0.078
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:PrimeCondition:State
                      Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition:State   9  0.139 0.01549   1.118  0.348
Residuals            423  5.861 0.01385
```

3.2.1 plot

```
> ## figure
> state_rmisc = Rmisc::summarySE(statepercent,
+                               measurevar = "Percent",
+                               groupvars = c("PrimeCondition","State"))
> x ← c("know","dontknow", "other", "TOT")
> state_rmisc = state_rmisc %>%
+   mutate(rstate =  factor(State, levels = x)) %>%
+   arrange(rstate)
> library(ggplot2)
> library(ggthemes)
> percentplot = state_rmisc %>%
+   mutate(PrimeType = factor(PrimeCondition, levels = unique(PrimeCondition),
+                      labels = c("Both", "Phonological",
+                               "Semantic", "Unrelated")),
+    R = factor(rstate, levels = unique(rstate),
+                                labels = c( "1: Know","2: Dont Know",
+                                            "3:Other", "4: TOT")))%>%
+
```

```
+ ggplot(aes(x = R, y = Percent,
+            group = PrimeType, fill = PrimeType))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.7,
+           color= "black")+
+   geom_errorbar(aes(ymin=Percent - se, ymax=Percent + se),
+              width=.2, color = "gray26",
+              position = position_dodge(0.7))+
+  theme_few()+
+     xlab("") + ylab("Percentage of trials") +
+  scale_fill_manual(values = c( "lightsalmon", "red",
+                                "paleturquoise3","lightgreen"))+
+   ggtitle("E3: Young Adults")  +
+    theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5),
+                  axis.text.x = element_text(size = rel(1)),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
> percentplot
```



3.2.2 know

```
> e3_know = statepercent %>% filter(State == "know")
> e3_know_aov = aov(data = e3_know,
+                                Percent ~ PrimeCondition +
+                              Error(Subject/PrimeCondition))
> summary(e3_know_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 47  3.334 0.07094
```

```

Error: Subject:PrimeCondition
                 Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition    3 0.0492 0.01639    0.99  0.399
Residuals       141 2.3343 0.01656
```

3.2.3 dont know

```
> e3_dontknow = statepercent %>% filter(State == "dontknow")
> e3_dontknow_aov = aov(data = e3_dontknow,
+                           Percent ~ PrimeCondition +
+                          Error(Subject/PrimeCondition))
> summary(e3_dontknow_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 47  4.257 0.09058

Error: Subject:PrimeCondition
                 Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition    3 0.0624 0.02081    1.75  0.159
Residuals       141 1.6760 0.01189
```

3.2.4 other

```
> e3_other = statepercent %>% filter(State == "other")
> e3_other_aov = aov(data = e3_other,
+                           Percent ~ PrimeCondition +
+                          Error(Subject/PrimeCondition))
> summary(e3_other_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 47  1.846 0.03927

Error: Subject:PrimeCondition
                 Df Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition    3 0.0089 0.002979   0.371  0.774
Residuals       141 1.1330 0.008036
```

3.2.5 TOT

```
> e3_TOT = statepercent %>% filter(State == "TOT")
> e3_TOT_aov = aov(data = e3_TOT,
+                           Percent ~ PrimeCondition +
```

```
+                          Error(Subject/PrimeCondition))
> summary(e3_TOT_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 47  1.599 0.03403

Error: Subject:PrimeCondition
                Df Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition   3 0.0189 0.006301   1.239  0.298
Residuals      141 0.7172 0.005087
```

3.3 Split by Prime and Target Accuracy

3.3.1 anova

```
> state_acc = state[,c(2,3,96:159)]
> state_acc$Subject = as.factor(state_acc$Subject)
> library(tidyr)
> library(dplyr)
> stateaccnums <- state_acc %>%
+   gather(PrimeStatePrimeRetTarget, Trials,
+          r_know_p1_t1, r_know_p1_t0, r_know_p0_t1, r_know_p0_t0,
+          p_know_p1_t1, p_know_p1_t0, p_know_p0_t1, p_know_p0_t0,
+          b_know_p1_t1, b_know_p1_t0, b_know_p0_t1, b_know_p0_t0,
+          u_know_p1_t1, u_know_p1_t0, u_know_p0_t1, u_know_p0_t0,
+    r_dontknow_p1_t1,r_dontknow_p1_t0, r_dontknow_p0_t1,r_dontknow_p0_t0,
+    p_dontknow_p1_t1,p_dontknow_p1_t0, p_dontknow_p0_t1,p_dontknow_p0_t0,
+    b_dontknow_p1_t1,b_dontknow_p1_t0, b_dontknow_p0_t1,b_dontknow_p0_t0,
+    u_dontknow_p1_t1,u_dontknow_p1_t0, u_dontknow_p0_t1,u_dontknow_p0_t0,
+     r_other_p1_t1, r_other_p1_t0,r_other_p0_t1, r_other_p0_t0,
+     p_other_p1_t1, p_other_p1_t0,p_other_p0_t1, p_other_p0_t0,
+     b_other_p1_t1, b_other_p1_t0,b_other_p0_t1, b_other_p0_t0,
+     u_other_p1_t1, u_other_p1_t0,u_other_p0_t1, u_other_p0_t0,
+     r_TOT_p1_t1, r_TOT_p1_t0, r_TOT_p0_t1, r_TOT_p0_t0,
+     p_TOT_p1_t1, p_TOT_p1_t0, p_TOT_p0_t1, p_TOT_p0_t0,
+     b_TOT_p1_t1, b_TOT_p1_t0, b_TOT_p0_t1, b_TOT_p0_t0,
+     u_TOT_p1_t1, u_TOT_p1_t0, u_TOT_p0_t1, u_TOT_p0_t0) %>%
+   separate(PrimeStatePrimeRetTarget, c( 'Prime', 'State',
+                              'PrimeRet', 'TargetAcc'), sep = "_") %>%
+   arrange(Subject)
> stateaccnums$Subject <- as.factor(stateaccnums$Subject)
> stateaccnums$Prime <- as.factor(stateaccnums$Prime)
> stateaccnums$State <- as.factor(stateaccnums$State)
> stateaccnums$PrimeRet <- as.factor(stateaccnums$PrimeRet)
> stateaccnums$TargetAcc <- as.factor(stateaccnums$TargetAcc)
> stateaccnums$Trials <- as.numeric(as.character(stateaccnums$Trials))
```

```
> mainstate_aov = aov(data = stateaccnums,
+                     Trials ~ Prime*State*PrimeRet*TargetAcc +
+                       Error(Subject/(Prime*State*PrimeRet*TargetAcc)))
> summary(mainstate_aov)
```

```
Error: Subject
          Df    Sum Sq   Mean Sq F value Pr(>F)
Residuals 47 3.355e-25 7.138e-27

Error: Subject:Prime
           Df    Sum Sq   Mean Sq F value Pr(>F)
Prime       3 2.960e-27 9.868e-28   0.997  0.396
Residuals 141 1.395e-25 9.895e-28

Error: Subject:State
           Df Sum Sq Mean Sq F value Pr(>F)
State       3  846.1  282.02   44.48 <2e-16 ***
Residuals 141  893.9    6.34
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:PrimeRet
          Df Sum Sq Mean Sq F value Pr(>F)
PrimeRet   1   4.08   4.083   1.145   0.29
Residuals 47 167.54   3.565

Error: Subject:TargetAcc
          Df Sum Sq Mean Sq F value   Pr(>F)
TargetAcc  1  357.5   357.5    72.4 4.48e-11 ***
Residuals 47  232.1     4.9
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:Prime:State
             Df Sum Sq Mean Sq F value Pr(>F)
Prime:State   9   11.3   1.255   1.118  0.348
Residuals   423  474.7   1.122

Error: Subject:Prime:PrimeRet
                Df Sum Sq Mean Sq F value   Pr(>F)
Prime:PrimeRet   3  49.64  16.545   15.53 8.81e-09 ***
Residuals      141 150.24   1.066
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:State:PrimeRet
                Df Sum Sq Mean Sq F value   Pr(>F)
State:PrimeRet   3  133.9   44.62   24.46 8.42e-13 ***
```

```
Residuals       141  257.3     1.82
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:Prime:TargetAcc
                 Df Sum Sq Mean Sq F value Pr(>F)
Prime:TargetAcc   3   3.93   1.309   1.058  0.369
Residuals       141 174.45   1.237

Error: Subject:State:TargetAcc
                 Df Sum Sq Mean Sq F value Pr(>F)
State:TargetAcc   3 2558.0   852.7   193.4 <2e-16 ***
Residuals       141  621.6     4.4
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:PrimeRet:TargetAcc
                   Df Sum Sq Mean Sq F value   Pr(>F)
PrimeRet:TargetAcc  1  88.02   88.02   51.17 4.79e-09 ***
Residuals          47  80.85    1.72
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:Prime:State:PrimeRet
                     Df Sum Sq Mean Sq F value   Pr(>F)
Prime:State:PrimeRet   9   34.7   3.857   3.246 0.000793 ***
Residuals            423  502.7   1.188
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:Prime:State:TargetAcc
                      Df Sum Sq Mean Sq F value Pr(>F)
Prime:State:TargetAcc   9    7.8  0.8701   1.012  0.429
Residuals             423  363.5  0.8594

Error: Subject:Prime:PrimeRet:TargetAcc
                         Df Sum Sq Mean Sq F value   Pr(>F)
Prime:PrimeRet:TargetAcc   3  45.26  15.087   17.46 1.08e-09 ***
Residuals                141 121.86   0.864
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:State:PrimeRet:TargetAcc
                         Df Sum Sq Mean Sq F value   Pr(>F)
State:PrimeRet:TargetAcc   3   50.5  16.834    7.38 0.000125 ***
Residuals                141  321.6   2.281
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

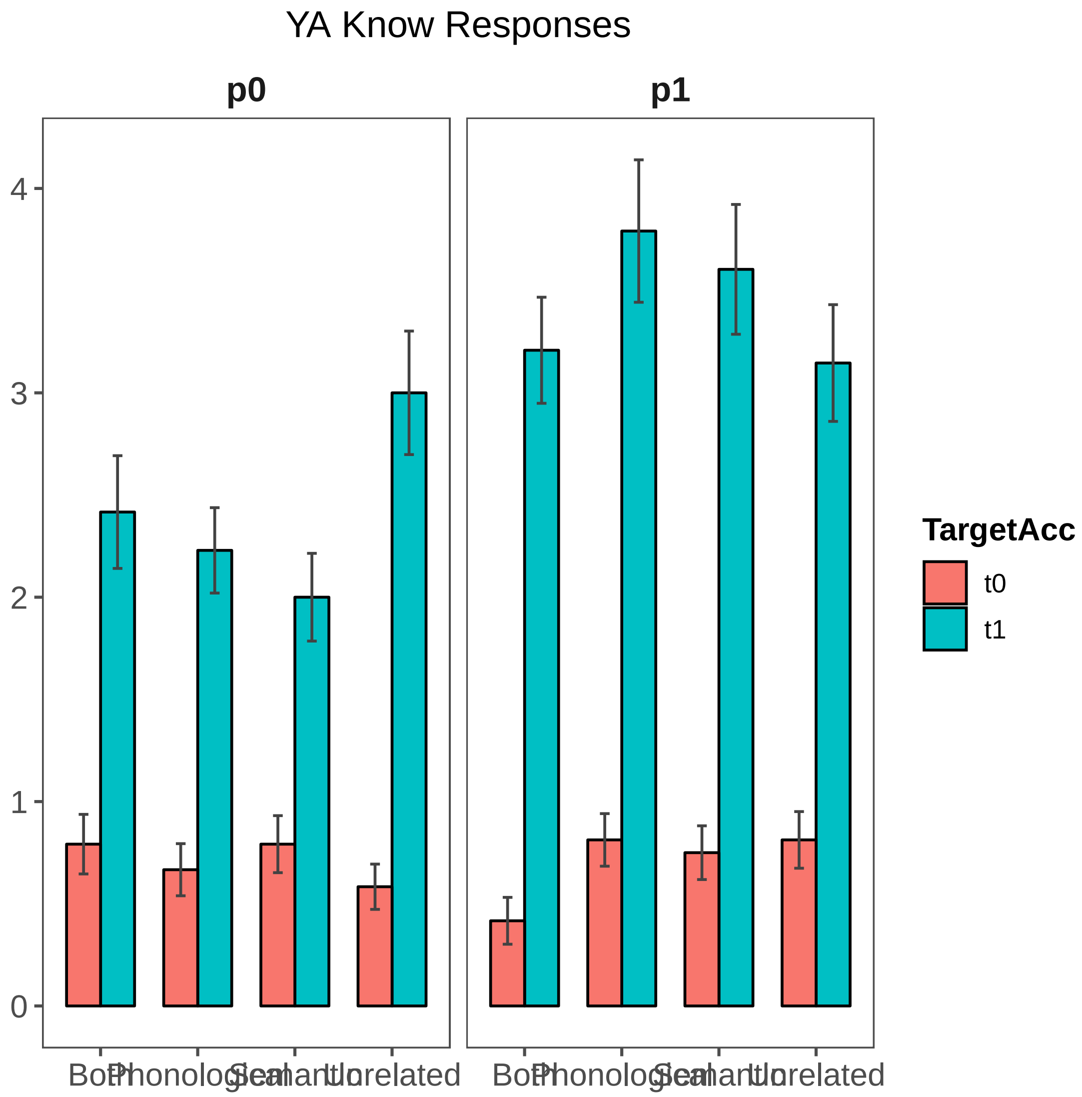
```
Error: Subject:Prime:State:PrimeRet:TargetAcc
                                Df Sum Sq Mean Sq F value    Pr(>F)
Prime:State:PrimeRet:TargetAcc    9   37.6   4.181   4.009 6.19e-05 ***
Residuals                       423  441.2   1.043
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
>
```

3.3.2 plot

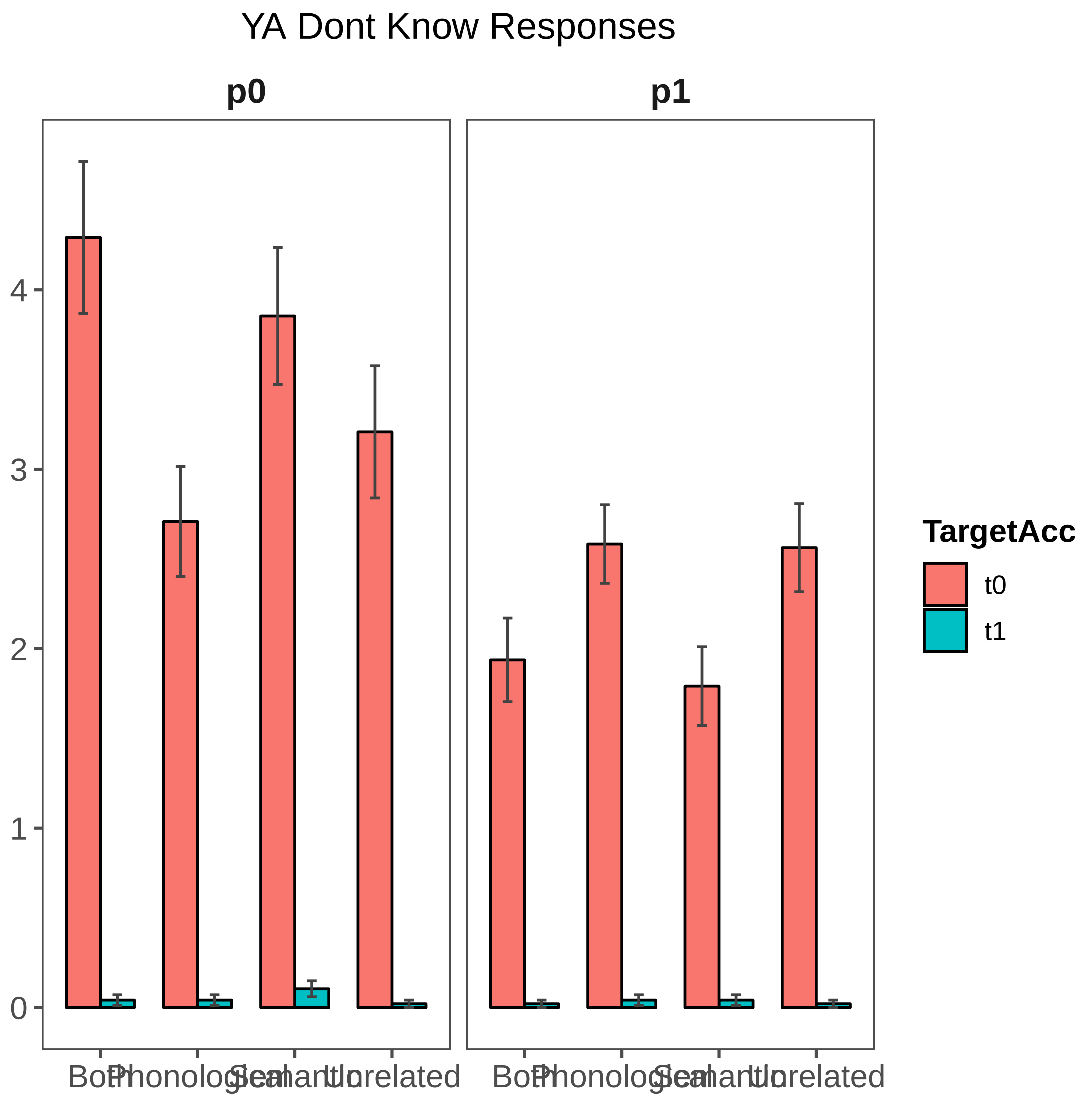
```
> ## figure
> stateacc_rmisc = Rmisc::summarySE(stateaccnums,
+                               measurevar = "Trials",
+                               groupvars = c("Prime","State",
+                                             "PrimeRet", "TargetAcc"))
> x <- c("know","dontknow", "other", "TOT")
> stateacc_rmisc = stateacc_rmisc %>%
+   mutate(rstate =  factor(State, levels = x)) %>%
+   arrange(rstate)
> know_rmisc = stateacc_rmisc %>% filter(State == "know")
> dontknow_rmisc = stateacc_rmisc %>% filter(State == "dontknow")
> other_rmisc = stateacc_rmisc %>% filter(State == "other")
> TOT_rmisc = stateacc_rmisc %>% filter(State == "TOT")
> library(ggplot2)
> library(ggthemes)
> know_percentplot = know_rmisc %>%
+   mutate(PrimeType = factor(Prime, levels = unique(Prime),
+                   labels = c("Both", "Phonological",
+                              "Semantic", "Unrelated")))%>%
+ ggplot(aes(x = PrimeType, y = Trials,
+              fill = TargetAcc, group=TargetAcc)) +
+  geom_bar(stat = "identity", position = "dodge", width = 0.7,
+           color= "black")+
+   geom_errorbar(aes(ymin=Trials - se, ymax=Trials + se),
+              width=.2, color = "gray26",
+              position = position_dodge(0.7))+
+  theme_few()+
+   facet_wrap(~PrimeRet)+
+     xlab("") + ylab("") +
+   ggtitle("YA Know Responses")  +
+    theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5),
+                   axis.text.x = element_text(size = rel(1)),
```

```
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
> know_percentplot
```



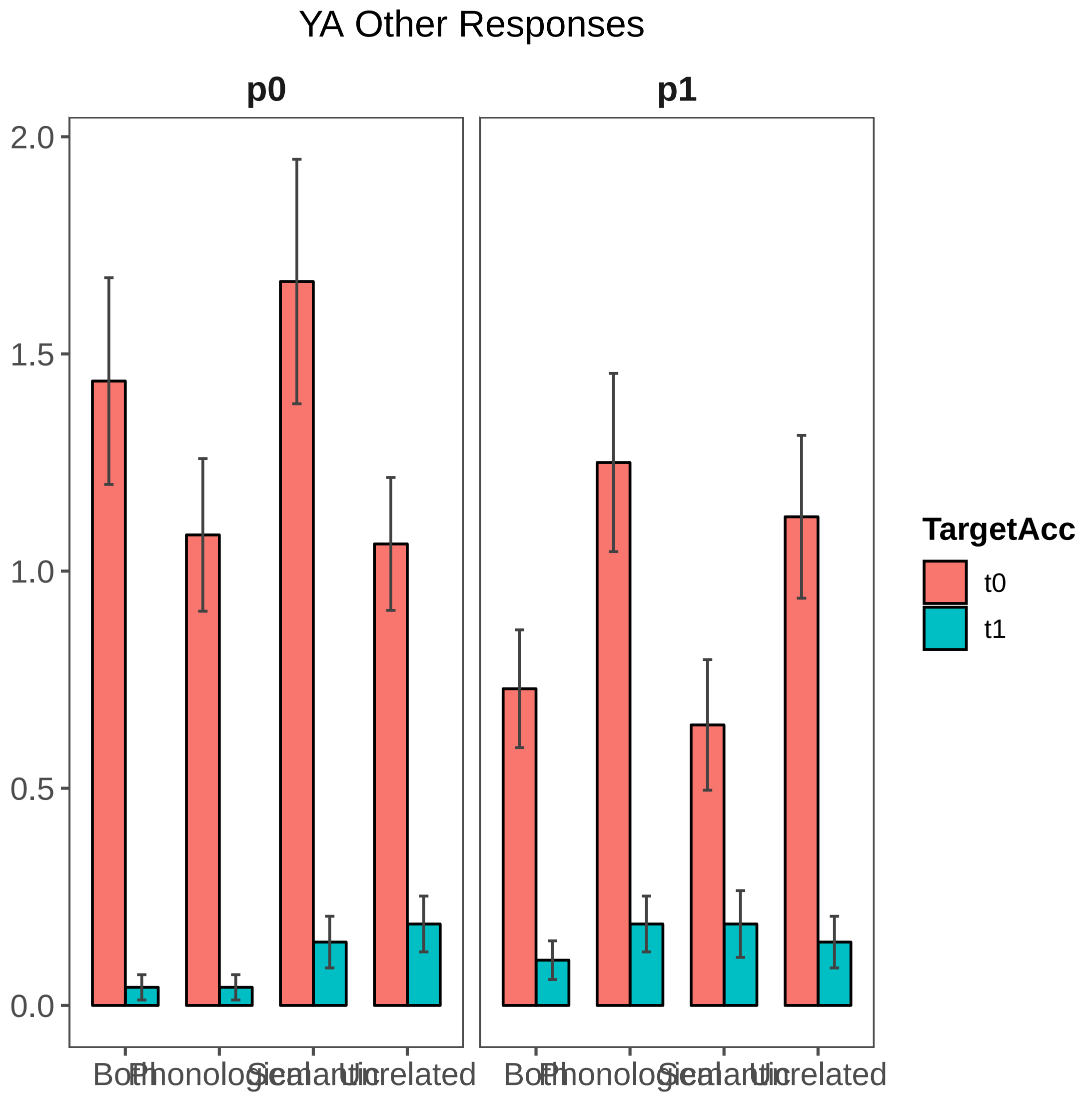

```
> dontknow_percentplot = dontknow_rmisc %>%
+   mutate(PrimeType = factor(Prime, levels = unique(Prime),
+                      labels = c("Both", "Phonological",
+                                 "Semantic", "Unrelated")))%>%
+ ggplot(aes(x = PrimeType, y = Trials,
+              fill = TargetAcc, group=TargetAcc)) +
+  geom_bar(stat = "identity", position = "dodge", width = 0.7,
+          color= "black")+
```

```
+   geom_errorbar(aes(ymin=Trials - se, ymax=Trials + se),
+               width=.2, color = "gray26",
+               position = position_dodge(0.7))+
+  theme_few()+
+   facet_wrap(~PrimeRet)+
+     xlab("") + ylab("") +
+   ggtitle("YA Dont Know Responses")  +
+    theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5),
+                  axis.text.x = element_text(size = rel(1)),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
> dontknow_percentplot
```

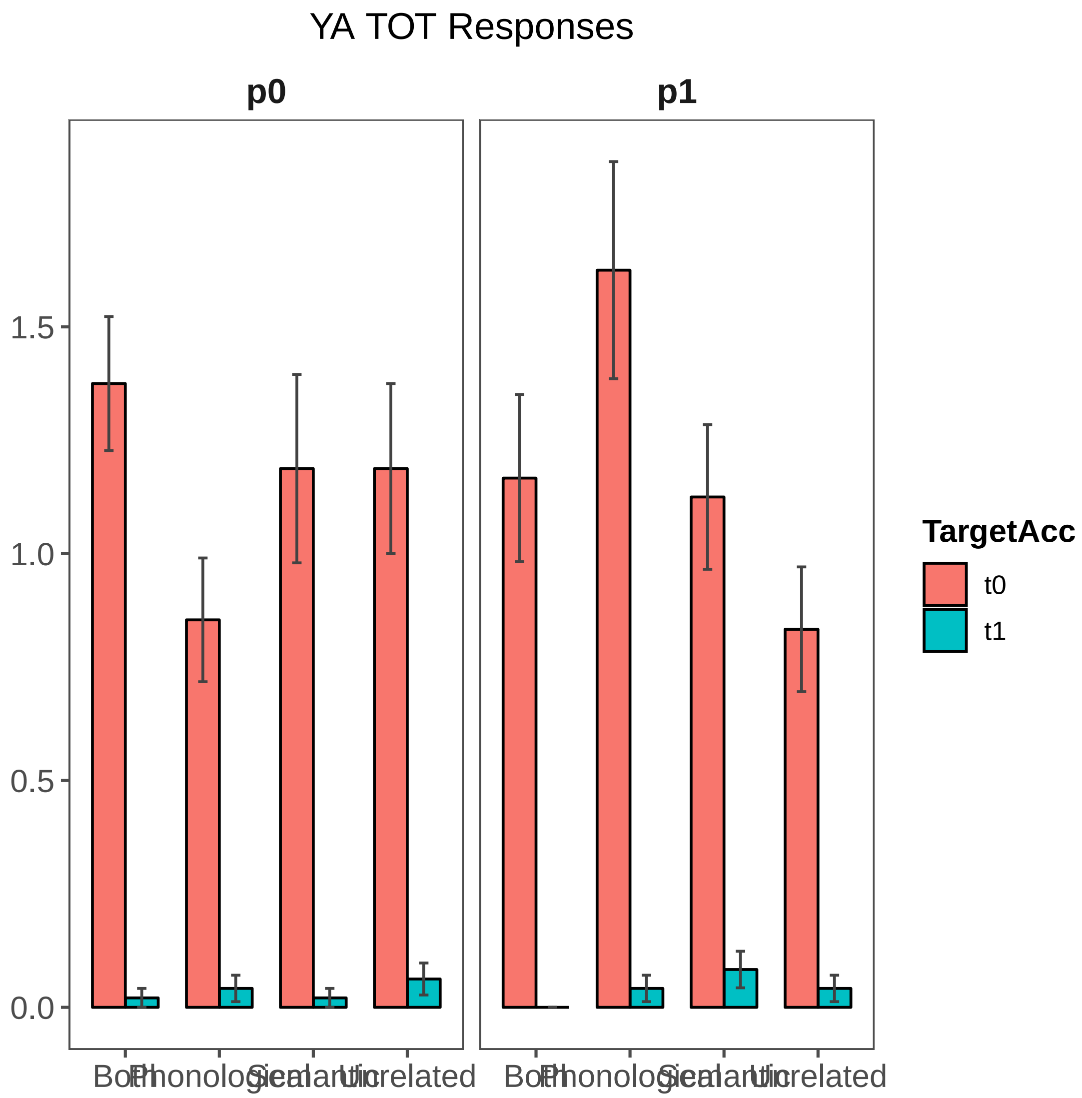



```
> other_percentplot = other_rmisc %>%
+   mutate(PrimeType = factor(Prime, levels = unique(Prime),
+                       labels = c("Both", "Phonological",
+                                  "Semantic", "Unrelated")))%>%
+ ggplot(aes(x = PrimeType, y = Trials,
+              fill = TargetAcc, group=TargetAcc)) +
+  geom_bar(stat = "identity", position = "dodge", width = 0.7,
+           color= "black")+
+   geom_errorbar(aes(ymin=Trials - se, ymax=Trials + se),
+              width=.2, color = "gray26",
+              position = position_dodge(0.7))+
```

```
+   theme_few()+
+    facet_wrap(~PrimeRet)+
+      xlab("") + ylab("") +
+    ggtitle("YA Other Responses")  +
+     theme(axis.text = element_text(size = rel(1)),
+            axis.title = element_text(face = "bold", size = rel(1)),
+            legend.title = element_text(face = "bold", size = rel(1)),
+           plot.title = element_text(hjust = .5),
+                   axis.text.x = element_text(size = rel(1)),
+           strip.text.x = element_text(face = "bold", size = rel(1.4)))
> other_percentplot
```



```
> TOT_percentplot = TOT_rmisc %>%
+   mutate(PrimeType = factor(Prime, levels = unique(Prime),
+                         labels = c("Both", "Phonological",
+                                    "Semantic", "Unrelated")))%>%
+ ggplot(aes(x = PrimeType, y = Trials,
+              fill = TargetAcc, group=TargetAcc)) +
+  geom_bar(stat = "identity", position = "dodge", width = 0.7,
+           color= "black")+
+   geom_errorbar(aes(ymin=Trials - se, ymax=Trials + se),
+              width=.2, color = "gray26",
+              position = position_dodge(0.7))+
+  theme_few()+
+   facet_wrap(~PrimeRet)+
+     xlab("") + ylab("") +
+   ggtitle("YA TOT Responses")  +
+    theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5),
+                   axis.text.x = element_text(size = rel(1)),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
> TOT_percentplot
```



3.4 Split by Target Accuracy in each state

3.4.1 anova

```
> state_targetacc = state[,c(2,3,192:223)]
> state_targetacc$Subject = as.factor(state_targetacc$Subject)
> library(tidyr)
> library(dplyr)
> stateaccnums_target <- state_targetacc %>%
+   gather(PrimeStatePrimeRetTarget, Trials,
+          r_know_t1, r_know_t0,
```

```
+           p_know_t1 , p_know_t0 ,
+           b_know_t1 , b_know_t0 ,
+           u_know_t1 , u_know_t0 ,
+     r_dontknow_t1 ,r_dontknow_t0 ,
+     p_dontknow_t1 ,p_dontknow_t0 ,
+     b_dontknow_t1 ,b_dontknow_t0 ,
+     u_dontknow_t1 ,u_dontknow_t0 ,
+      r_other_t1 , r_other_t0 ,
+      p_other_t1 , p_other_t0 ,
+      b_other_t1 , b_other_t0 ,
+      u_other_t1 , u_other_t0 ,
+      r_TOT_t1 , r_TOT_t0 ,
+      p_TOT_t1 , p_TOT_t0 ,
+      b_TOT_t1 , b_TOT_t0 ,
+      u_TOT_t1 , u_TOT_t0) %>%
+    separate(PrimeStatePrimeRetTarget , c( 'Prime', 'State',
+                                'TargetAcc'), sep = "_") %>%
+    arrange(Subject)
> stateaccnums_target$Subject <- as.factor(stateaccnums_target$Subject)
> stateaccnums_target$Prime <- as.factor(stateaccnums_target$Prime)
> stateaccnums_target$State <- as.factor(stateaccnums_target$State)
> stateaccnums_target$TargetAcc <- as.factor(stateaccnums_target$TargetAcc)
> stateaccnums_target$Trials <- as.numeric(as.character(stateaccnums_target$Trials))
> statetargetacc_aov = aov(data = stateaccnums_target ,
+                     Trials ~ Prime*State*TargetAcc +
+                       Error(Subject/(Prime*State*TargetAcc)))
> summary(statetargetacc_aov)
```

```
Error: Subject
          Df    Sum Sq   Mean Sq F value Pr(>F)
Residuals 47 1.374e-25 2.923e-27

Error: Subject:Prime
           Df    Sum Sq   Mean Sq F value Pr(>F)
Prime       3 2.980e-27 9.949e-28   0.749  0.524
Residuals 141 1.872e-25 1.328e-27

Error: Subject:State
           Df Sum Sq Mean Sq F value Pr(>F)
State       3   1692   564.0   44.48 <2e-16 ***
Residuals 141   1788    12.7
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:TargetAcc
          Df Sum Sq Mean Sq F value   Pr(>F)
TargetAcc  1  715.0   715.0    72.4 4.48e-11 ***
Residuals 47  464.2     9.9
```

```
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:Prime:State
             Df Sum Sq Mean Sq F value Pr(>F)
Prime:State   9   22.6   2.510   1.118  0.348
Residuals   423  949.4   2.244

Error: Subject:Prime:TargetAcc
                 Df Sum Sq Mean Sq F value Pr(>F)
Prime:TargetAcc   3    7.9   2.618   1.058  0.369
Residuals       141  348.9   2.474

Error: Subject:State:TargetAcc
                 Df Sum Sq Mean Sq F value Pr(>F)
State:TargetAcc   3   5116  1705.3   193.4 <2e-16 ***
Residuals       141   1243     8.8
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:Prime:State:TargetAcc
                       Df Sum Sq Mean Sq F value Pr(>F)
Prime:State:TargetAcc   9   15.7   1.740   1.012  0.429
Residuals             423  727.1   1.719
```

```
>
```

3.4.2 plot

```
> ## figure
> statetargetacc_rmisc = Rmisc::summarySE(stateaccnums_target,
+                                measurevar = "Trials",
+                                groupvars = c("Prime","State", "TargetAcc"))
> x <- c("know","dontknow", "other", "TOT")
> statetargetacc_rmisc = statetargetacc_rmisc %>%
+   mutate(rstate =  factor(State, levels = x)) %>%
+   arrange(rstate)
> library(ggplot2)
> library(ggthemes)
> statetargetacc_plot = statetargetacc_rmisc %>%
+   mutate(PrimeType = factor(Prime, levels = unique(Prime),
+                   labels = c("Both", "Phonological",
+                              "Semantic", "Unrelated")),
+          TargetAccuracy = factor(TargetAcc, levels = unique(TargetAcc),
+                   labels = c("Failed", "Correct")),
+    R = factor(rstate, levels = unique(rstate),
+                               labels = c( "1: Know","2: Dont Know",
```

```
+                                             "3:Other", "4: TOT")))%>%
+
+ ggplot(aes(x = PrimeType, y = Trials,
+            group = TargetAccuracy, fill = TargetAccuracy))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.7,
+          color= "black")+
+   geom_errorbar(aes(ymin=Trials - se, ymax=Trials + se),
+              width=.2, color = "gray26",
+              position = position_dodge(0.7))+
+  theme_few()+
+   facet_wrap(~R)+
+     xlab("") + ylab("") +
+   ggtitle("YA States")  +
+   scale_fill_wsj()+
+    theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5),
+                   axis.text.x = element_text(size = rel(1)),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
> statetargetacc_plot
```

4 Prime Demasking Analysis

```
> library(dplyr)
> PrimeRetrieval = PrimeRetrieval %>% arrange(Subject, Stimuli2)
> colnames(PrimeRetrieval) = c("Stimuli2", "PrimeCondition",
+                             "AgeGroup", "ID", "Subject",
+                              "Procedure",
+                             "Prime",  "Trial",
+                              "PrimeDefResp", "PrimeFirstResp_ACC",
+                             "PrimeDefRT",  "PrimeRespRESP", "PrimeRespRT",
+                              "Target", "TargetDefResp", "Accuracy",
+                             "TargetDefRT",
+                             "TargetRespRESP", "TargetRespRT",
+                             "State", "StateRT",
+                            "RTrecognisePrime", "RTrecogniseTarget",
+                            "Count", "PrimeRespType", "TargetRespType",
+                            "Prime_POS", "Target_POS",
+                            "PrimeAcc")
> #PrimeRetrieval = PrimeRetrieval %>% filter(PrimeAcc == 1)
> primewith_firsttrim_target = subset(PrimeRetrieval,
+                                PrimeRetrieval$RTrecogniseTarget > 250 &
+                               PrimeRetrieval$RTrecogniseTarget < 7000)
> primewith_firsttrim_prime = subset(PrimeRetrieval,
+                                PrimeRetrieval$RTrecognisePrime > 250 &
```

```
+                                PrimeRetrieval$RTrecognisePrime < 7000)
> primewith_firsttrim_targetdef = subset(PrimeRetrieval,
+                                 PrimeRetrieval$TargetDefRT > 250 &
+                                PrimeRetrieval$TargetDefRT < 9000)
```

RTRecogniseprime

```
> ## FOR PRIME
> ## aggregate per subject all IVs and DVs
> meanRT = group_by(primewith_firsttrim_prime, Subject) %>%
+   summarise_at(vars(RTrecognisePrime), mean)
> colnames(meanRT) = c("Subject",
+                      "MeanRTrecogPrime")
> sdRT = group_by(primewith_firsttrim_prime, Subject) %>%
+   summarise_at(vars(RTrecognisePrime), sd)
> colnames(sdRT) = c("Subject",
+                      "sdRTrecogPrime")
> RT_agg = merge(meanRT, sdRT, by = "Subject")
> ## merge aggregate info with long data
> primewith_z_prime = merge(primewith_firsttrim_prime,
+                             RT_agg, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> primewith_z_prime = primewith_z_prime %>% mutate(zPrimeRecogRT =
+                                                 (RTrecognisePrime -
+                                                    MeanRTrecogPrime)/sdRTrecogPrime)
> ## checking: subject level means should be zero
>
> sub_pic = group_by(primewith_z_prime, Subject) %>%
+   summarise_at(vars(zPrimeRecogRT), mean)
```

RTRecogniseTarget

```
> ## FOR TARGET
> ## aggregate per subject all IVs and DVs
> meanRT = group_by(primewith_firsttrim_target, Subject) %>%
+   summarise_at(vars(RTrecogniseTarget), mean)
> colnames(meanRT) = c("Subject", "MeanRTrecogTarget")
> sdRT = group_by(primewith_firsttrim_target, Subject) %>%
+   summarise_at(vars(RTrecogniseTarget), sd)
> colnames(sdRT) = c("Subject", "sdRTrecogTarget")
> RT_agg = merge(meanRT, sdRT, by = "Subject")
> ## merge aggregate info with long data
> primewith_z_target= merge(primewith_firsttrim_target,
+                             RT_agg, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
```

```
> library(dplyr)
> primewith_z_target = primewith_z_target %>% mutate( zTargetRecogRT =
+                                              (RTrecogniseTarget -
+                                                 MeanRTrecogTarget)/sdRTrecogTarget)
> ## checking: subject level means should be zero
>
> sub_pic = group_by(primewith_z_target, Subject) %>%
+   summarise_at(vars(zTargetRecogRT), mean)
>
```

TargetDefRT

```
> ## FOR TARGET
> ## aggregate per subject all IVs and DVs
> meanRT = group_by(primewith_firsttrim_targetdef, Subject) %>%
+   summarise_at(vars(TargetDefRT), mean)
> colnames(meanRT) = c("Subject", "MeanTargetRT")
> sdRT = group_by(primewith_firsttrim_targetdef, Subject) %>%
+   summarise_at(vars(TargetDefRT), sd)
> colnames(sdRT) = c("Subject", "sdTargetRT")
> RT_agg = merge(meanRT, sdRT, by = "Subject")
> ## merge aggregate info with long data
> primewith_z_targetdef = merge(primewith_firsttrim_targetdef,
+                               RT_agg, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> primewith_z_targetdef = primewith_z_targetdef %>% mutate( zTargetRT =
+                                              (TargetDefRT -
+                                                 MeanTargetRT)/sdTargetRT)
> ## checking: subject level means should be zero
>
> sub_pic = group_by(primewith_z_targetdef, Subject) %>%
+   summarise_at(vars(zTargetRT), mean)
>
```

5 Trimming z-RTs

```
> #Note: We are trimming based on PrimeRecog RT because that's the RT we care about most
> primewith_z_trimmed_prime = subset(primewith_z_prime,
+                                 primewith_z_prime$zPrimeRecogRT < 3 &
+                                   primewith_z_prime$zPrimeRecogRT > -3)
> primewith_z_trimmed_target = subset(primewith_z_target,
+                                 primewith_z_target$zTargetRecogRT < 3 &
+                                   primewith_z_target$zTargetRecogRT > -3)
> primewith_z_trimmed_targetdef = subset(primewith_z_targetdef,
```

```
+                                    primewith_z_targetdef$zTargetRT < 3 &
+                                      primewith_z_targetdef$zTargetRT > -3)
```

6 Repeating z-scoring

6.1 For prime

```
> ## aggregate per subject all IVs and DVs
> meanRT_prime = group_by(primewith_z_trimmed_prime, Subject) %>%
+   summarise_at(vars(RTrecognisePrime), mean)
> colnames(meanRT_prime) = c("Subject",
+                          "MeanRTrecogPrime_trim")
> sdRT_prime = group_by(primewith_z_trimmed_prime, Subject) %>%
+   summarise_at(vars(RTrecognisePrime), sd)
> colnames(sdRT_prime) = c("Subject",
+                          "sdRTrecogPrime_trim")
> RT_agg_prime = merge(meanRT_prime, sdRT_prime, by = "Subject")
> ## merge aggregate info with long data
> primewith_final_z_prime = merge(primewith_z_trimmed_prime,
+                                 RT_agg_prime, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> primewith_final_z_prime = primewith_final_z_prime %>%
+                                    mutate( zPrimeRecogRT_trim =
+                                                (RTrecognisePrime -
+                                        MeanRTrecogPrime_trim)/sdRTrecogPrime_trim)
> ## checking: subject level means should be zero
>
> sub_pic = group_by(primewith_final_z_prime, Subject) %>%
+   summarise_at(vars(zPrimeRecogRT_trim), mean)
>
```

6.2 For Target

```
> ## aggregate per subject all IVs and DVs
> meanRT_target = group_by(primewith_z_trimmed_target, Subject) %>%
+   summarise_at(vars(RTrecogniseTarget), mean)
> colnames(meanRT_target) = c("Subject",
+                          "MeanRTrecogTarget_trim")
> sdRT_target = group_by(primewith_z_trimmed_target, Subject) %>%
+   summarise_at(vars(RTrecogniseTarget), sd)
> colnames(sdRT_target) = c("Subject",
+                           "sdRTrecogTarget_trim")
> RT_agg_target = merge(meanRT_target, sdRT_target, by = "Subject")
> ## merge aggregate info with long data
> primewith_final_z_target = merge(primewith_z_trimmed_target,
```

```
+                               RT_agg_target, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> primewith_final_z_target = primewith_final_z_target %>%
+                                    mutate( zTargetRecogRT_trim =
+                                               (RTrecogniseTarget -
+                                       MeanRTrecogTarget_trim)/sdRTrecogTarget_trim)
> ## checking: subject level means should be zero
>
> sub_pic = group_by(primewith_final_z_target, Subject) %>%
+   summarise_at(vars(zTargetRecogRT_trim), mean)
>
```

6.3 For TargetDefRT

```
> ## aggregate per subject all IVs and DVs
> meanRT_targetdef = group_by(primewith_z_trimmed_targetdef, Subject) %>%
+   summarise_at(vars(TargetDefRT), mean)
> colnames(meanRT_targetdef) = c("Subject", "MeanTargetRT_trim")
> sdRT_targetdef = group_by(primewith_z_trimmed_targetdef, Subject) %>%
+   summarise_at(vars(TargetDefRT), sd)
> colnames(sdRT_targetdef) = c("Subject", "sdTargetRT_trim")
> RT_agg_targetdef = merge(meanRT_targetdef, sdRT_targetdef, by = "Subject")
> ## merge aggregate info with long data
> primewith_final_z_targetdef = merge(primewith_z_trimmed_targetdef,
+                               RT_agg_targetdef, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> primewith_final_z_targetdef = primewith_final_z_targetdef %>%
+                                    mutate(zTargetRT_trim =
+                                               (TargetDefRT -
+                                                  MeanTargetRT_trim)/sdTargetRT_trim)
> ## checking: subject level means should be zero
>
> sub_pic = group_by(primewith_final_z_targetdef, Subject) %>%
+   summarise_at(vars(zTargetRT_trim), mean)
>
```

6.4 Combining z-RT Prime and Target

```
> ## now we have separately z-scored RTprime and RTtarget. Need to combine.
> ## taking only necessary columns
> primewith_final_z_prime2 = primewith_final_z_prime[,c(1,8,35)]
> primewith_final_z = merge(primewith_final_z_target,
+                               primewith_final_z_prime2,
+                               by  = c("Subject", "Trial"))
```

```
> primefinal_z_targetdef = merge(primewith_final_z_targetdef,
+                               primewith_final_z_prime2,
+                               by  = c("Subject", "Trial"))
```

7 Linear Models

```
> # Mean RT to retrieve Target as a function of Prime Condition
>
> # Effect of RT prime on Accuracy
> library(lme4)
> RTprime_acc_model = glmer(data = primewith_final_z,
+                           Accuracy ~ zPrimeRecogRT_trim +
+                             (1|Subject) + (1|Stimuli2), family = binomial )
> summary(RTprime_acc_model)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: Accuracy ~ zPrimeRecogRT_trim + (1 | Subject) + (1 | Stimuli2)
   Data: primewith_final_z

     AIC      BIC   logLik deviance df.resid
  3565.2   3589.7  -1778.6   3557.2     3348

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.9656 -0.5903 -0.3173  0.6488  5.8856

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli2 (Intercept) 1.7950   1.3398
 Subject  (Intercept) 0.5028   0.7091
Number of obs: 3352, groups:  Stimuli2, 72; Subject, 48

Fixed effects:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)         -0.8793     0.1943  -4.526 6.01e-06 ***
zPrimeRecogRT_trim  -0.1268     0.0455  -2.786  0.00533 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Correlation of Fixed Effects:
            (Intr)
zPrmRcgRT_t 0.009
```

```
> contrasts(primewith_final_z_prime$PrimeCondition) = contr.treatment(n = 4, base = 4)
```

```
> ### NOTE: for Acc analysis, use the full primewith_final_z_prime data, why exclude
> ## the RTrecogniseTarget when not using: greater power with this!
> RTprime_acc_model_2 = glmer(data = primewith_final_z_prime,
+                             Accuracy ~ zPrimeRecogRT_trim*PrimeCondition +
+                               (1|Subject) + (1|Stimuli2),
+                             family = "binomial",
+                             control=glmerControl(optimizer="bobyqa",
+             optCtrl=list(maxfun=100000)))
> summary(RTprime_acc_model_2)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: Accuracy ~ zPrimeRecogRT_trim * PrimeCondition + (1 | Subject) +
    (1 | Stimuli2)
   Data: primewith_final_z_prime
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))

     AIC      BIC   logLik deviance df.resid
  3583.6   3645.0  -1781.8   3563.6     3402

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2817 -0.5794 -0.3084  0.6266  6.2235

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli2 (Intercept) 1.854    1.3614
 Subject  (Intercept) 0.515    0.7176
Number of obs: 3412, groups:  Stimuli2, 72; Subject, 48

Fixed effects:
                                   Estimate Std. Error z value Pr(>|z|)
(Intercept)                      -0.7666952  0.2097205  -3.656 0.000256 ***
zPrimeRecogRT_trim               -0.0005049  0.0877473  -0.006 0.995409
PrimeCondition1                  -0.2521303  0.1209911  -2.084 0.037172 *
PrimeCondition2                  -0.0830203  0.1198147  -0.693 0.488368
PrimeCondition3                  -0.1793436  0.1207394  -1.485 0.137444
zPrimeRecogRT_trim:PrimeCondition1 -0.3310171  0.1306205  -2.534 0.011271 *
zPrimeRecogRT_trim:PrimeCondition2  0.0121897  0.1282882   0.095 0.924300
zPrimeRecogRT_trim:PrimeCondition3 -0.2129789  0.1279263  -1.665 0.095942 .
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Correlation of Fixed Effects:
            (Intr) zPrRRT_ PrmCn1 PrmCn2 PrmCn3 zPRRT_:PC1 zPRRT_:PC2
zPrmRcgRT_t -0.005
PrimeCndtn1 -0.276  0.011
```

```
PrimeCndtn2 -0.281  0.012   0.489
PrimeCndtn3 -0.278  0.013   0.485  0.491
zPrRRT_:PC1  0.002 -0.681  -0.002 -0.008 -0.010
zPrRRT_:PC2  0.002 -0.694  -0.010  0.058 -0.010  0.473
zPrRRT_:PC3  0.003 -0.687  -0.009 -0.009  0.049  0.470     0.479
```

```
> options(contrasts = c("contr.sum","contr.poly"))
> car::Anova(RTprime_acc_model_2)
```

```
Analysis of Deviance Table (Type II Wald chisquare tests)

Response: Accuracy
                                  Chisq Df Pr(>Chisq)
zPrimeRecogRT_trim               7.4749  1   0.006257 **
PrimeCondition                   4.7007  3   0.195076
zPrimeRecogRT_trim:PrimeCondition 9.7558  3   0.020760 *
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> anova(RTprime_acc_model_2)
```

```
Analysis of Variance Table
                                  Df Sum Sq Mean Sq F value
zPrimeRecogRT_trim                 1 7.3436  7.3436  7.3436
PrimeCondition                     3 4.7037  1.5679  1.5679
zPrimeRecogRT_trim:PrimeCondition  3 9.8225  3.2742  3.2742
```

```
> # > confint(RTprime_acc_model_2)
> # Computing profile confidence intervals ...
> #                                           2.5 %       97.5 %
> # .sig01                                  1.1351567   1.65598118
> # .sig02                                  0.5650623   0.92276591
> # (Intercept)                            -1.1834774  -0.35464605
> # zPrimeRecogRT_trim                     -0.1747754   0.17314296
> # PrimeCondition1                        -0.4926117  -0.01292099
> # PrimeCondition2                        -0.3210261   0.15457528
> # PrimeCondition3                        -0.4193510   0.05985194
> # zPrimeRecogRT_trim:PrimeCondition1     -0.5915886  -0.07351300
> # zPrimeRecogRT_trim:PrimeCondition2     -0.2425930   0.26641632
> # zPrimeRecogRT_trim:PrimeCondition3     -0.4676063   0.03977605
>
> RTprime_acc_model_2_2 = glmer(data = primewith_final_z_prime,
+                         Accuracy ~ zPrimeRecogRT_trim*PrimeCondition +
+                           (1|Subject),
+                         family = binomial ,
+                         control=glmerControl(optimizer="bobyqa",
+             optCtrl=list(maxfun=100000)))
> summary(RTprime_acc_model_2_2)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: Accuracy ~ zPrimeRecogRT_trim * PrimeCondition + (1 | Subject)
   Data: primewith_final_z_prime
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))

     AIC      BIC   logLik deviance df.resid
  4274.9   4330.1  -2128.4   4256.9     3403

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.5798 -0.7374 -0.5527  1.1029  3.2497

Random effects:
 Groups  Name        Variance Std.Dev.
 Subject (Intercept) 0.2701   0.5198
Number of obs: 3412, groups:  Subject, 48

Fixed effects:
                                   Estimate Std. Error z value Pr(>|z|)
(Intercept)                        -0.57248    0.10480  -5.463 4.69e-08 ***
zPrimeRecogRT_trim                 -0.01755    0.07675  -0.229  0.81916
PrimeCondition1                    -0.15557    0.10535  -1.477  0.13976
PrimeCondition2                    -0.07988    0.10419  -0.767  0.44326
PrimeCondition3                    -0.13943    0.10486  -1.330  0.18362
zPrimeRecogRT_trim:PrimeCondition1 -0.35005    0.11021  -3.176  0.00149 **
zPrimeRecogRT_trim:PrimeCondition2 -0.09575    0.10962  -0.873  0.38240
zPrimeRecogRT_trim:PrimeCondition3 -0.33199    0.11115  -2.987  0.00282 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Correlation of Fixed Effects:
            (Intr) zPrRRT_ PrmCn1 PrmCn2 PrmCn3 zPRRT_:PC1 zPRRT_:PC2
zPrmRcgRT_t  0.008
PrimeCndtn1 -0.481 -0.008
PrimeCndtn2 -0.487 -0.008   0.485
PrimeCndtn3 -0.484 -0.008   0.481  0.487
zPrRRT_:PC1 -0.005 -0.699   0.002  0.006  0.006
zPrRRT_:PC2 -0.004 -0.702   0.006  0.074  0.006  0.490
zPrRRT_:PC3 -0.004 -0.692   0.006  0.006  0.082  0.484      0.485
```

```
> car::Anova(RTprime_acc_model_2_2)
```

```
Analysis of Deviance Table (Type II Wald chisquare tests)

Response: Accuracy
```

```
                                  Chisq Df Pr(>Chisq)
zPrimeRecogRT_trim              28.1246  1  1.137e-07 ***
PrimeCondition                   2.3782  3   0.497712
zPrimeRecogRT_trim:PrimeCondition 14.7883  3   0.002007 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> anova(RTprime_acc_model_2_2, RTprime_acc_model_2)
```

```
Data: primewith_final_z_prime
Models:
RTprime_acc_model_2_2: Accuracy ~ zPrimeRecogRT_trim * PrimeCondition + (1 | Subject)
RTprime_acc_model_2: Accuracy ~ zPrimeRecogRT_trim * PrimeCondition + (1 | Subject) +
RTprime_acc_model_2:     (1 | Stimuli2)
                      Df    AIC    BIC  logLik deviance  Chisq Chi Df
RTprime_acc_model_2_2  9 4274.9 4330.1 -2128.4   4256.9
RTprime_acc_model_2   10 3583.6 3645.0 -1781.8   3563.6 693.25      1
                      Pr(>Chisq)
RTprime_acc_model_2_2
RTprime_acc_model_2    < 2.2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> y = sjPlot::plot_model(RTprime_acc_model_2, type = "int")
> y + theme_few()+
+       xlab("RT to Demask Prime") + ylab("Predicted Target Accuracy") +
+ ggtitle("YA: Target Accuracy ~ \nDemasking RT x Prime Condition") +
+   theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
> RTprime_acc_model_3 = glmer(data = primewith_final_z_prime,
+              Accuracy ~ zPrimeRecogRT_trim*PrimeFirstResp_ACC*PrimeCondition +
+                              (1|Subject) + (1|Stimuli2), family = "binomial",
+     control=glmerControl(optimizer="bobyqa",
+             optCtrl=list(maxfun=100000)))
> summary(RTprime_acc_model_3)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: Accuracy ~ zPrimeRecogRT_trim * PrimeFirstResp_ACC * PrimeCondition +
    (1 | Subject) + (1 | Stimuli2)
   Data: primewith_final_z_prime
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))

     AIC      BIC   logLik deviance df.resid
```

```
  3587.5   3698.0  -1775.8   3551.5     3394

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2062 -0.5788 -0.3041  0.6261  5.2902

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli2 (Intercept) 1.8025   1.3426
 Subject  (Intercept) 0.4922   0.7016
Number of obs: 3412, groups:  Stimuli2, 72; Subject, 48

Fixed effects:
                                                        Estimate Std. Error
(Intercept)                                            -0.799660   0.211642
zPrimeRecogRT_trim                                     -0.014903   0.101821
PrimeFirstResp_ACC1                                    -0.008229   0.097069
PrimeCondition1                                        -0.291344   0.140179
PrimeCondition2                                        -0.067197   0.134560
PrimeCondition3                                        -0.132196   0.137570
zPrimeRecogRT_trim:PrimeFirstResp_ACC1                  0.087752   0.102876
zPrimeRecogRT_trim:PrimeCondition1                     -0.278557   0.152503
zPrimeRecogRT_trim:PrimeCondition2                      0.034654   0.143395
zPrimeRecogRT_trim:PrimeCondition3                     -0.104447   0.145961
PrimeFirstResp_ACC1:PrimeCondition1                    -0.175200   0.143707
PrimeFirstResp_ACC1:PrimeCondition2                    -0.022500   0.138736
PrimeFirstResp_ACC1:PrimeCondition3                    -0.213980   0.140863
zPrimeRecogRT_trim:PrimeFirstResp_ACC1:PrimeCondition1  0.093286   0.151875
zPrimeRecogRT_trim:PrimeFirstResp_ACC1:PrimeCondition2 -0.041207   0.142383
zPrimeRecogRT_trim:PrimeFirstResp_ACC1:PrimeCondition3 -0.092252   0.146773
                                                       z value Pr(>|z|)
(Intercept)                                             -3.778 0.000158 ***
zPrimeRecogRT_trim                                      -0.146 0.883636
PrimeFirstResp_ACC1                                     -0.085 0.932443
PrimeCondition1                                         -2.078 0.037675 *
PrimeCondition2                                         -0.499 0.617510
PrimeCondition3                                         -0.961 0.336587
zPrimeRecogRT_trim:PrimeFirstResp_ACC1                   0.853 0.393666
zPrimeRecogRT_trim:PrimeCondition1                      -1.827 0.067765 .
zPrimeRecogRT_trim:PrimeCondition2                       0.242 0.809035
zPrimeRecogRT_trim:PrimeCondition3                      -0.716 0.474248
PrimeFirstResp_ACC1:PrimeCondition1                     -1.219 0.222790
PrimeFirstResp_ACC1:PrimeCondition2                     -0.162 0.871164
PrimeFirstResp_ACC1:PrimeCondition3                     -1.519 0.128747
zPrimeRecogRT_trim:PrimeFirstResp_ACC1:PrimeCondition1   0.614 0.539063
zPrimeRecogRT_trim:PrimeFirstResp_ACC1:PrimeCondition2  -0.289 0.772269
zPrimeRecogRT_trim:PrimeFirstResp_ACC1:PrimeCondition3  -0.629 0.529650
---
```

```
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> car::Anova(RTprime_acc_model_3)
```

```
Analysis of Deviance Table (Type II Wald chisquare tests)

Response: Accuracy
                                                   Chisq Df Pr(>Chisq)
zPrimeRecogRT_trim                                2.0263  1    0.15459
PrimeFirstResp_ACC                                5.2388  1    0.02209 *
PrimeCondition                                    4.6224  3    0.20163
zPrimeRecogRT_trim:PrimeFirstResp_ACC             1.9268  1    0.16511
zPrimeRecogRT_trim:PrimeCondition                 4.4479  3    0.21699
PrimeFirstResp_ACC:PrimeCondition                 3.4170  3    0.33168
zPrimeRecogRT_trim:PrimeFirstResp_ACC:PrimeCondition 1.5861  3    0.66255
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> z = sjPlot::plot_model(RTprime_acc_model_3, type = "int",
+                    terms = c("zPrimeRecogRT_trim", "PrimeFirstResp_ACC"))
> z + theme_few()+
+       xlab("RT to Demask Prime") + ylab("Predicted Target Accuracy") +
+ ggtitle("YA: Target Accuracy ~ \nDemasking RT x Prime Condition x Prime Retrieval Accu
+   theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
> anova(RTprime_acc_model_2, RTprime_acc_model_3)
```

```
Data: primewith_final_z_prime
Models:
RTprime_acc_model_2: Accuracy ~ zPrimeRecogRT_trim * PrimeCondition + (1 | Subject) +
RTprime_acc_model_2:     (1 | Stimuli2)
RTprime_acc_model_3: Accuracy ~ zPrimeRecogRT_trim * PrimeFirstResp_ACC * PrimeCondition
RTprime_acc_model_3:     (1 | Subject) + (1 | Stimuli2)
                    Df    AIC  BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
RTprime_acc_model_2 10 3583.6 3645 -1781.8   3563.6
RTprime_acc_model_3 18 3587.5 3698 -1775.8   3551.5 12.129      8     0.1455
```

```
>
```

7.1 Effect of Prime RT on Target RT

```
> library(lme4)
> library(lmerTest)
```

```
> contrasts(primewith_final_z$PrimeCondition) = contr.treatment(n = 4, base = 4)
> RTprime_RT_model_2 = lmer(data = primewith_final_z,
+                     zTargetRecogRT_trim ~ zPrimeRecogRT_trim*PrimeCondition +
+                             (1|Subject) + (1|Stimuli2))
> summary(RTprime_RT_model_2)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: zTargetRecogRT_trim ~ zPrimeRecogRT_trim * PrimeCondition + (1 |
    Subject) + (1 | Stimuli2)
   Data: primewith_final_z

REML criterion at convergence: 8464.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2682 -0.6667 -0.1178  0.5877  4.4333

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli2 (Intercept) 0.2953   0.5434
 Subject  (Intercept) 0.0000   0.0000
 Residual             0.6788   0.8239
Number of obs: 3352, groups:  Stimuli2, 72; Subject, 48

Fixed effects:
                                    Estimate Std. Error         df t value
(Intercept)                          0.02964    0.07010   91.94932   0.423
zPrimeRecogRT_trim                   0.06724    0.03018 3285.20340   2.228
PrimeCondition1                     -0.04970    0.04052 3273.61861  -1.227
PrimeCondition2                     -0.03517    0.04048 3272.91098  -0.869
PrimeCondition3                      0.01614    0.04024 3272.76873   0.401
zPrimeRecogRT_trim:PrimeCondition1   0.10499    0.04213 3298.55774   2.492
zPrimeRecogRT_trim:PrimeCondition2   0.04839    0.04366 3295.77421   1.108
zPrimeRecogRT_trim:PrimeCondition3   0.07813    0.04245 3292.06652   1.840
                                   Pr(>|t|)
(Intercept)                          0.6734
zPrimeRecogRT_trim                   0.0259 *
PrimeCondition1                      0.2200
PrimeCondition2                      0.3851
PrimeCondition3                      0.6885
zPrimeRecogRT_trim:PrimeCondition1   0.0128 *
zPrimeRecogRT_trim:PrimeCondition2   0.2678
zPrimeRecogRT_trim:PrimeCondition3   0.0658 .
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Correlation of Fixed Effects:
```

```
             (Intr) zPrRRT_ PrmCn1 PrmCn2 PrmCn3 zPRRT_:PC1 zPRRT_:PC2
zPrmRcgRT_t  0.001
PrimeCndtn1 -0.286  0.000
PrimeCndtn2 -0.287 -0.001   0.496
PrimeCndtn3 -0.288 -0.001   0.499  0.499
zPrRRT_:PC1  0.000 -0.725  -0.071  0.001  0.000
zPrRRT_:PC2  0.000 -0.701  -0.002  0.061 -0.001  0.512
zPrRRT_:PC3 -0.001 -0.707   0.000  0.002  0.023  0.514      0.500
```

```
> car::Anova(RTprime_RT_model_2)
```

```
Analysis of Deviance Table (Type II Wald chisquare tests)

Response: zTargetRecogRT_trim
                                    Chisq Df Pr(>Chisq)
zPrimeRecogRT_trim                70.1551  1    < 2e-16 ***
PrimeCondition                     2.8380  3    0.41728
zPrimeRecogRT_trim:PrimeCondition  6.8156  3    0.07801 .
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> options(contrasts = c("contr.sum","contr.poly"))
> anova(RTprime_RT_model_2)
```

```
Type III Analysis of Variance Table with Satterthwaite's method
                                  Sum Sq Mean Sq NumDF  DenDF F value  Pr(>F)
zPrimeRecogRT_trim                46.913  46.913     1 3297.6 69.1120 < 2e-16
PrimeCondition                     2.314   0.771     3 3273.3  1.1363 0.33292
zPrimeRecogRT_trim:PrimeCondition  4.626   1.542     3 3297.4  2.2719 0.07822

zPrimeRecogRT_trim                ***
PrimeCondition
zPrimeRecogRT_trim:PrimeCondition .
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> # > confint(RTprime_RT_model_2)
> # Computing profile confidence intervals ...
> #                                            2.5 %       97.5 %
> # .sig01                                0.457422647 0.64602628
> # .sig02                                0.000000000 0.02942538
> # .sigma                                0.803488757 0.84333785
> # (Intercept)                          -0.108282360 0.16762085
> # zPrimeRecogRT_trim                    0.008125974 0.12633493
> # PrimeCondition1                      -0.129050764 0.02964582
> # PrimeCondition2                      -0.114453497 0.04412090
> # PrimeCondition3                      -0.062682860 0.09495232
```

```
> # zPrimeRecogRT_trim:PrimeCondition1  0.022506875 0.18754099
> # zPrimeRecogRT_trim:PrimeCondition2 -0.037073414 0.13395257
> # zPrimeRecogRT_trim:PrimeCondition3 -0.004960774 0.16134136
>
> RTprime_RT_model_1 = lmer(data = primewith_final_z,
+                    zTargetRecogRT_trim ~ PrimeCondition +
+                            (1|Subject) + (1|Stimuli2))
> summary(RTprime_RT_model_1)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: zTargetRecogRT_trim ~ PrimeCondition + (1 | Subject) + (1 | Stimuli2)
   Data: primewith_final_z

REML criterion at convergence: 8520.1

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.0891 -0.6751 -0.1183  0.5899  4.8914

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli2 (Intercept) 0.3072   0.5543
 Subject  (Intercept) 0.0000   0.0000
 Residual             0.6933   0.8327
Number of obs: 3352, groups:  Stimuli2, 72; Subject, 48

Fixed effects:
                 Estimate Std. Error         df t value Pr(>|t|)
(Intercept)       0.02931    0.07140   91.63782   0.411    0.682
PrimeCondition1  -0.02500    0.04073 3276.93776  -0.614    0.539
PrimeCondition2  -0.04768    0.04077 3276.99052  -1.169    0.242
PrimeCondition3   0.01046    0.04065 3276.78506   0.257    0.797

Correlation of Fixed Effects:
            (Intr) PrmCn1 PrmCn2
PrimeCndtn1 -0.286
PrimeCndtn2 -0.285  0.500
PrimeCndtn3 -0.286  0.502  0.501
```

```
> anova(RTprime_RT_model_1, RTprime_RT_model_2)
```

```
Data: primewith_final_z
Models:
RTprime_RT_model_1: zTargetRecogRT_trim ~ PrimeCondition + (1 | Subject) + (1 | Stimuli2
RTprime_RT_model_2: zTargetRecogRT_trim ~ zPrimeRecogRT_trim * PrimeCondition + (1 |
RTprime_RT_model_2:     Subject) + (1 | Stimuli2)
                  Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
```

```
RTprime_RT_model_1  7 8516.1 8558.9 -4251.0   8502.1
RTprime_RT_model_2 11 8447.8 8515.1 -4212.9   8425.8 76.285      4  1.066e-15

RTprime_RT_model_1
RTprime_RT_model_2 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> car::Anova(RTprime_RT_model_1)
```

```
Analysis of Deviance Table (Type II Wald chisquare tests)

Response: zTargetRecogRT_trim
                Chisq Df Pr(>Chisq)
PrimeCondition 2.4645  3     0.4817
```

```
>
```

7.2 Constrast Codes

```
> RT_fixedeff = matrix(fixef(RTprime_acc_model_2))
> both = RT_fixedeff[1]
> phon = RT_fixedeff[1] + RT_fixedeff[3]
> sem =  RT_fixedeff[1] + RT_fixedeff[4]
> unrel = RT_fixedeff[1] + RT_fixedeff[5]
> final_means = as.data.frame(rbind(both, phon, sem, unrel))
> final_means$odds = exp(final_means$V1)
> final_means$prob = final_means$odds/(1+final_means$odds)
>
```

7.3 Collapsing P and U conditions

```
> primewith_final_z$NewPrimes = ifelse(primewith_final_z$PrimeCondition == "P" |
+                     primewith_final_z$PrimeCondition == "U", "Unrelated",
+                     ifelse(primewith_final_z$PrimeCondition == "B",
+                            "Both", "Semantic" ))
> RTprime_acc_model_2_new = glmer(data = primewith_final_z,
+                           Accuracy ~ zPrimeRecogRT_trim*NewPrimes +
+                             (1|Subject) + (1|Stimuli2), family = binomial )
> summary(RTprime_acc_model_2_new)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: Accuracy ~ zPrimeRecogRT_trim * NewPrimes + (1 | Subject) + (1 |
```

```
    Stimuli2)
   Data: primewith_final_z

     AIC      BIC   logLik deviance df.resid
  3559.1   3608.1  -1771.6   3543.1     3344

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2033 -0.5883 -0.3116  0.6398  5.8780

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli2 (Intercept) 1.8020   1.3424
 Subject  (Intercept) 0.5054   0.7109
Number of obs: 3352, groups:  Stimuli2, 72; Subject, 48

Fixed effects:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -0.90559    0.19535  -4.636 3.56e-06 ***
zPrimeRecogRT_trim            -0.17658    0.04939  -3.576 0.000349 ***
NewPrimes1                    -0.10179    0.06785  -1.500 0.133530
NewPrimes2                    -0.01741    0.06767  -0.257 0.796916
zPrimeRecogRT_trim:NewPrimes1 -0.15645    0.07434  -2.105 0.035334 *
zPrimeRecogRT_trim:NewPrimes2 -0.02908    0.07287  -0.399 0.689883
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Correlation of Fixed Effects:
            (Intr) zPrRRT_ NwPrm1 NwPrm2 zPRRT_:NP1
zPrmRcgRT_t  0.011
NewPrimes1   0.036 -0.032
NewPrimes2   0.033  0.052  -0.644
zPrRRT_:NP1 -0.008  0.175   0.040 -0.045
zPrRRT_:NP2  0.012  0.133  -0.044  0.094 -0.651
```

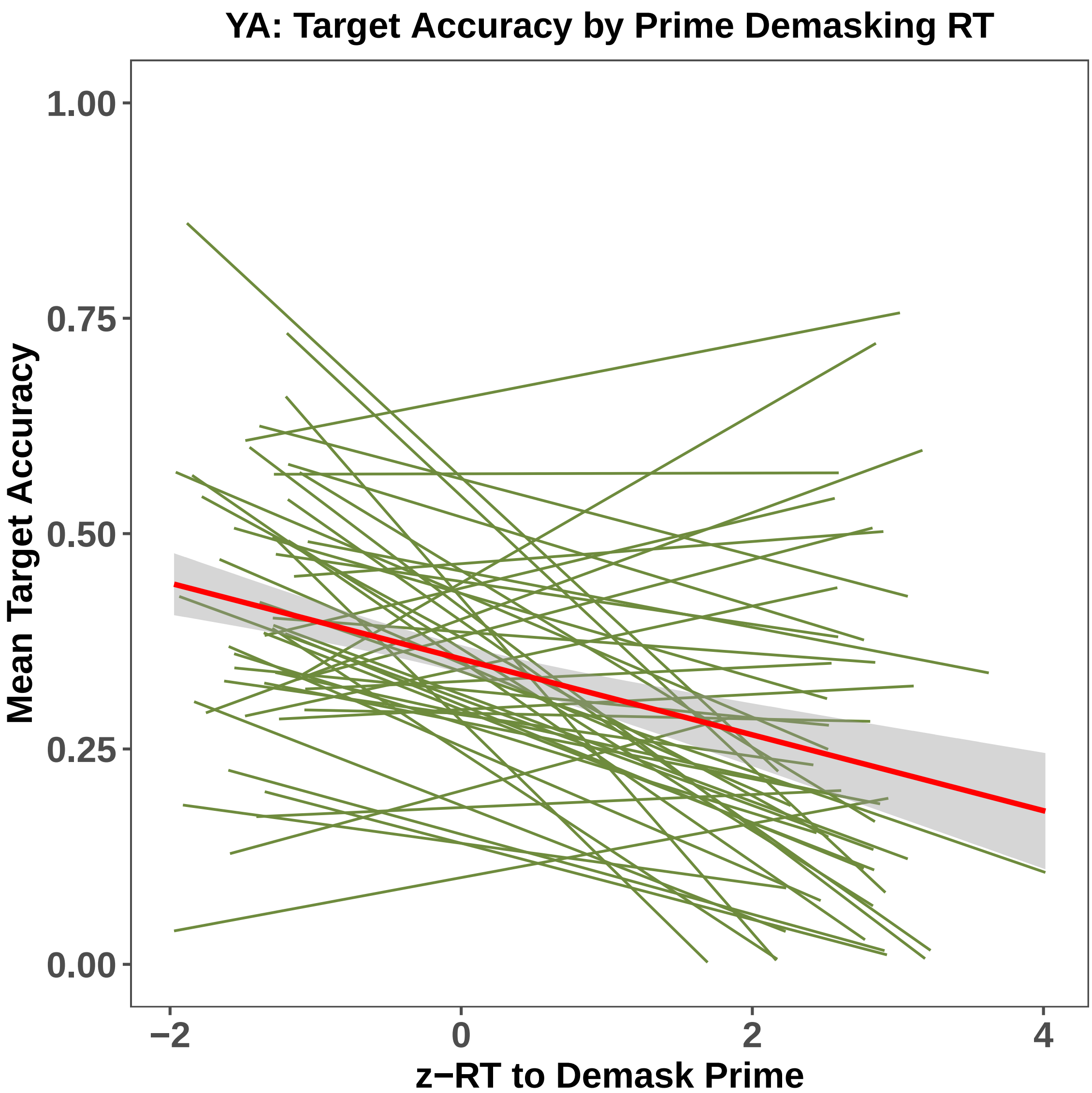
```
> car::Anova(RTprime_acc_model_2_new)
```

```
Analysis of Deviance Table (Type II Wald chisquare tests)

Response: Accuracy
                              Chisq Df Pr(>Chisq)
zPrimeRecogRT_trim           6.9553  1   0.008357 **
NewPrimes                    4.2522  2   0.119299
zPrimeRecogRT_trim:NewPrimes 9.8484  2   0.007269 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

8 Plotting Raw Data

8.1 Model 1

```
> library(ggplot2)
> library(ggthemes)
> mainplot = primewith_final_z %>%
+   ggplot(aes(x =zPrimeRecogRT_trim , y = Accuracy,
+              group = factor(Subject))) +
+   geom_smooth(method = "lm", se = FALSE, color = "darkolivegreen4", size = 0.5)+
+   guides(color = FALSE)+
+     xlab("z-RT to Demask Prime") + ylab ("Mean Target Accuracy")+
+   ggtitle("YA: Target Accuracy by Prime Demasking RT")+
+ theme_few() +
+   ylim(0,1) +
+     theme(axis.text = element_text(face = "bold", size = rel(1.2)),
+           axis.title = element_text(face = "bold", size = rel(1.2)),
+           legend.title = element_text(face = "bold", size = rel(1.2)),
+           plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
> mainplot + stat_smooth(aes(group = 1), method = "lm", color = "red")
>
```



8.2 Model 2

```
> ## sd for zPrimeRecogRT_trim
> sd(primewith_final_z_prime$zPrimeRecogRT_trim)
```

```
[1] 0.9930866
```

```
> # this is the model
>
> # RTprime_acc_model_2 = glmer(data = primewith_final_z_prime,
```

```
> #                             Accuracy ~ zPrimeRecogRT_trim*PrimeCondition +
> #                               (1|Subject) + (1|Stimuli2),
> #                             family = "binomial",
> #                             control=glmerControl(optimizer="bobyqa",
> #              optCtrl=list(maxfun=100000)))
> # summary(RTprime_acc_model_2)
>
> primert_model = lmer(data = primewith_final_z_prime,
+                     zPrimeRecogRT_trim ~ 1 + (1 | Subject) +
+                       (1|Stimuli2))
> summary(primert_model)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: zPrimeRecogRT_trim ~ 1 + (1 | Subject) + (1 | Stimuli2)
   Data: primewith_final_z_prime

REML criterion at convergence: 9521.4

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.1255 -0.7335 -0.2088  0.5849  4.1482

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli2 (Intercept) 0.06299  0.2510
 Subject  (Intercept) 0.00000  0.0000
 Residual             0.92414  0.9613
Number of obs: 3412, groups:  Stimuli2, 72; Subject, 48

Fixed effects:
             Estimate Std. Error        df t value Pr(>|t|)
(Intercept) 7.774e-04  3.385e-02 7.097e+01   0.023    0.982
```

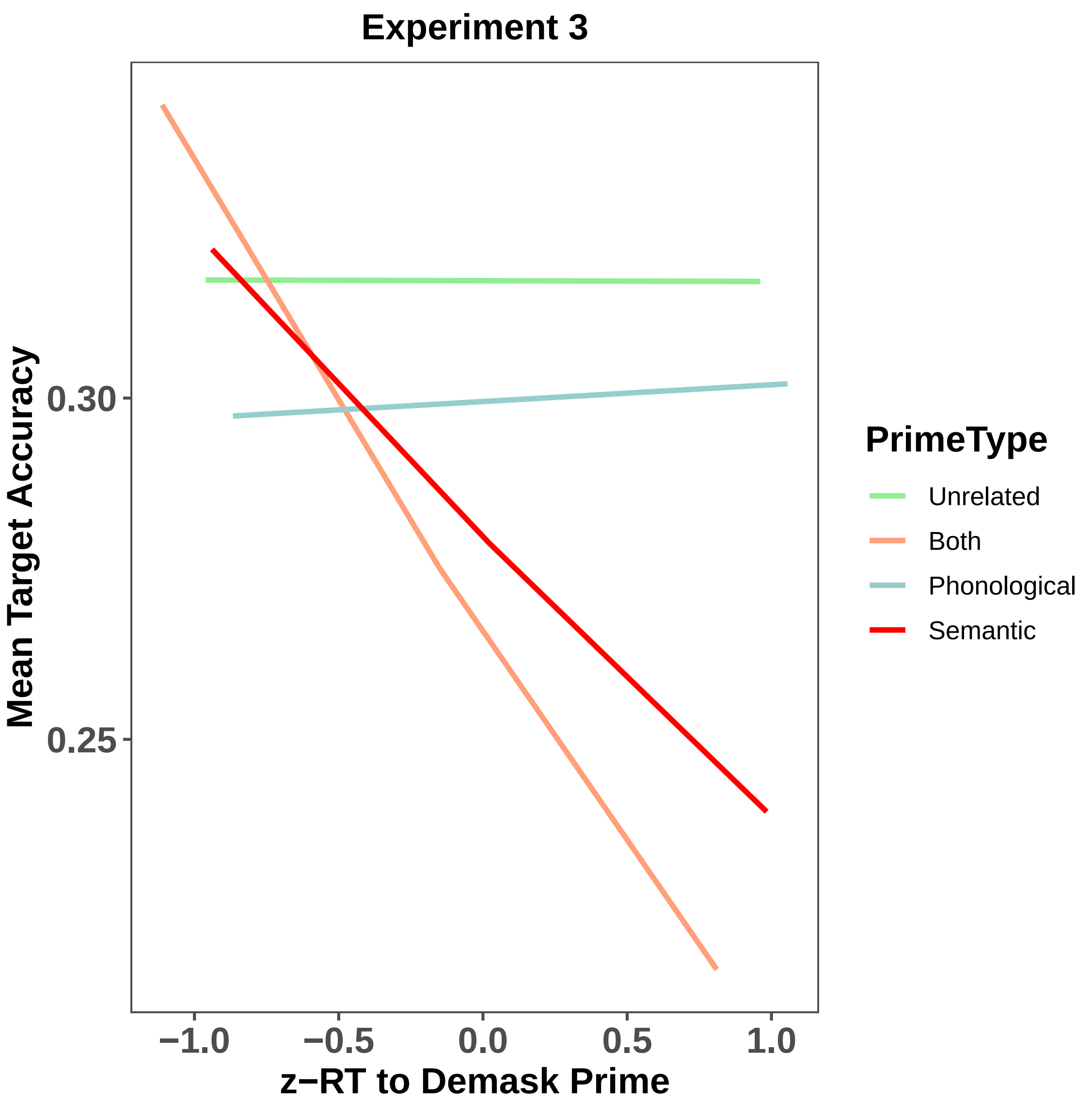
```
> VarCorr(primert_model)
```

```
 Groups   Name        Std.Dev.
 Stimuli2 (Intercept) 0.25097
 Subject  (Intercept) 0.00000
 Residual             0.96132
```

```
> SD_prime <- as.data.frame(VarCorr(primert_model))[3, 5]
> ## now we need to find increments for each prime condition
>
> primert_model_2 <- lmer(data = primewith_final_z_prime,
+                        zPrimeRecogRT_trim ~ 1 + PrimeCondition +
+                      (1|Subject) + (1|Stimuli2))
> prime_Inc_1_U <- 0*fixef(primert_model_2)[1]
```

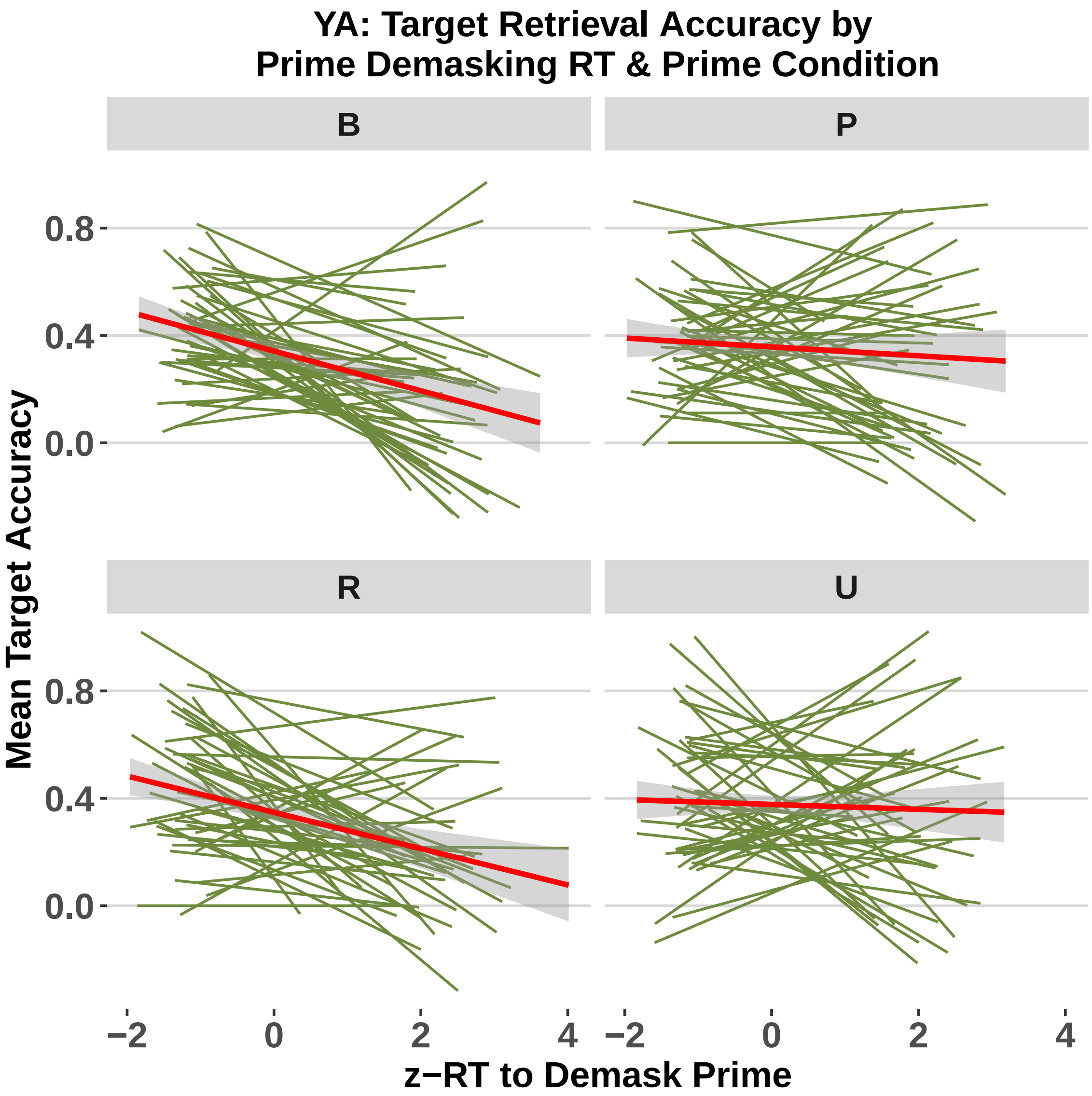
```
> prime_Inc_1_B ← 1*fixef(primert_model_2)[2]
> prime_Inc_1_P ← 1*fixef(primert_model_2)[3]
> prime_Inc_1_R ← 1*fixef(primert_model_2)[4]
> predict_data_U ← with(primewith_final_z_prime,
+                   data.frame(school=1,
+  zPrimeRecogRT_trim=seq(from=-prime_Inc_1_U-SD_prime,
+          to=-prime_Inc_1_U+SD_prime,
+          by=SD_prime),
+  PrimeCondition = 0))
> predict_data_B ← with(primewith_final_z_prime,
+                   data.frame(school=1,
+  zPrimeRecogRT_trim=seq(from=-prime_Inc_1_B-SD_prime,
+          to=-prime_Inc_1_B+SD_prime,
+          by=SD_prime),
+  PrimeCondition = 1))
> predict_data_P ← with(primewith_final_z_prime,
+                   data.frame(school=1,
+  zPrimeRecogRT_trim=seq(from=-prime_Inc_1_P-SD_prime,
+          to=-prime_Inc_1_P+SD_prime,
+          by=SD_prime),
+  PrimeCondition = 2))
> predict_data_R ← with(primewith_final_z_prime,
+                   data.frame(school=1,
+  zPrimeRecogRT_trim=seq(from=-prime_Inc_1_R-SD_prime,
+          to=-prime_Inc_1_R+SD_prime,
+          by=SD_prime),
+  PrimeCondition = 3))
> predict_data = rbind(predict_data_U,
+                      predict_data_B,
+                      predict_data_P,
+                      predict_data_R)
> predict_data$PrimeCondition = ifelse(predict_data$PrimeCondition == 0, "U",
+                     ifelse(predict_data$PrimeCondition == 1, "B",
+                     ifelse(predict_data$PrimeCondition == 2, "P", "R")))
> predict_data = predict_data %>%
+   mutate(predicted_values = predict(RTprime_acc_model_2,
+          newdata = predict_data, re.form = NA))
> predict_data$prob = exp(predict_data$predicted_values)/(1+exp(predict_data$predicted_v
> predict_data$PrimeCondition = ordered(as.factor(as.character(predict_data$PrimeConditi
> predict_data %>%
+   mutate(PrimeType = factor(PrimeCondition, levels = unique(PrimeCondition),
+                     labels = c("Unrelated",
+                                "Both","Phonological","Semantic")))%>%
+   ggplot(aes(x = zPrimeRecogRT_trim, y = prob,
+             color = PrimeType)) +
+     geom_line(size = 1) +
+     xlab("z-RT to Demask Prime") + ylab ("Mean Target Accuracy")+
+   ggtitle("Experiment 3")+
```

```
+ theme_few() +
+   scale_color_manual(values = c( "lightgreen", "lightsalmon",
+                                  "paleturquoise3","red"))+
+     theme(axis.text = element_text(face = "bold", size = rel(1.2)),
+           axis.title = element_text(face = "bold", size = rel(1.2)),
+           legend.title = element_text(face = "bold", size = rel(1.2)),
+     plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
```



8.3 Model 2: Raw data

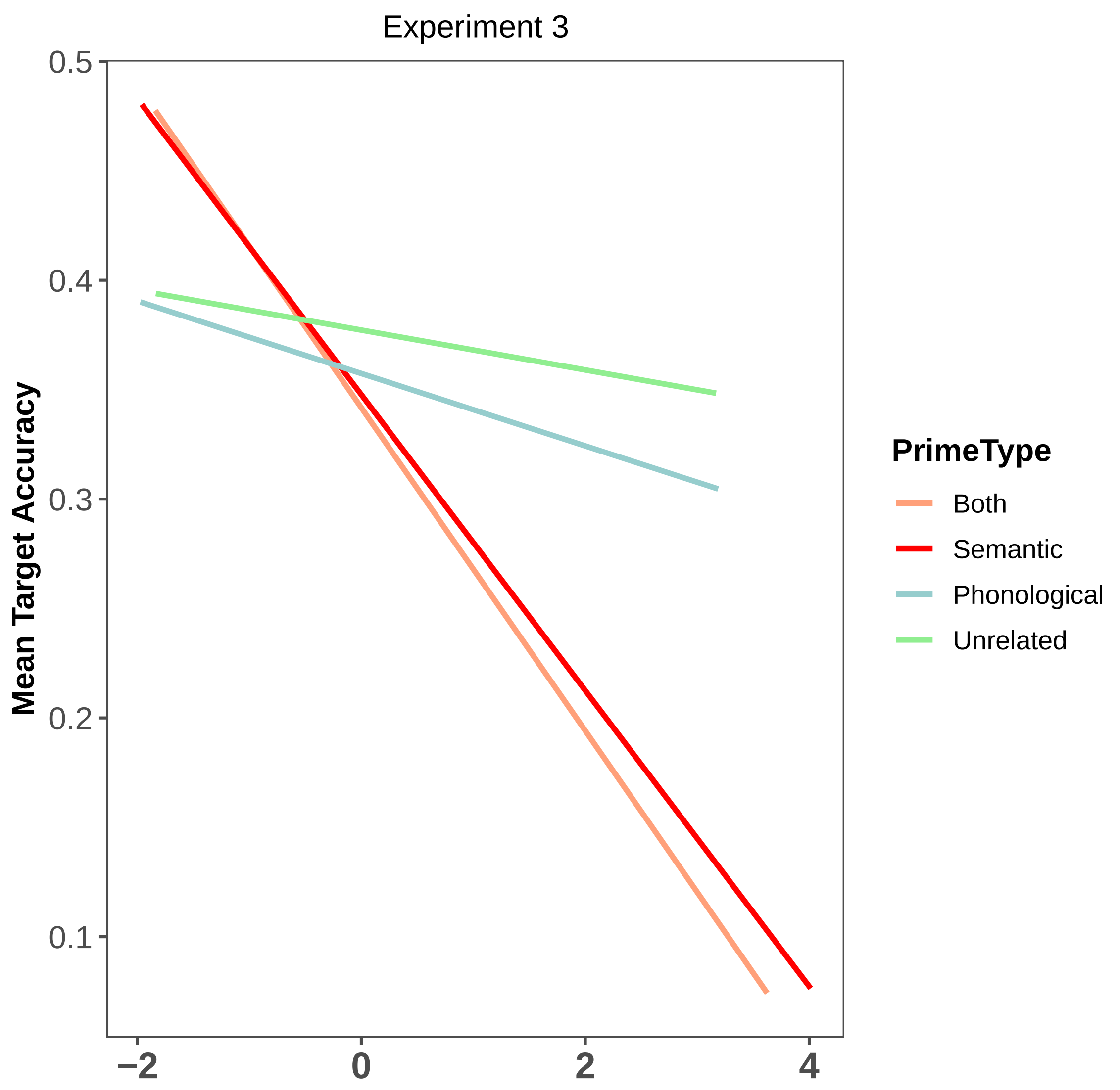
```
> primeplot = primewith_final_z %>%
+   mutate(PrimeType = factor(PrimeCondition,
+                       levels = unique(PrimeCondition),
+                       labels = c("Both Prime", "Phonological Prime",
+                                  "Semantic Prime", "Unrelated Prime")))%>%
+   ggplot(aes(x = zPrimeRecogRT_trim, y = Accuracy,
+              group = factor(Subject))) +
+   geom_smooth(method = "lm", se = FALSE, color = "darkolivegreen4", size = 0.5)+
+   facet_wrap(~PrimeCondition)+
+     xlab("z-RT to Demask Prime") + ylab ("Mean Target Accuracy")+
+   ggtitle("YA: Target Retrieval Accuracy by \nPrime Demasking RT & Prime Condition")+
+ theme_hc() +
+     theme(axis.text = element_text(face = "bold", size = rel(1.2)),
+           axis.title = element_text(face = "bold", size = rel(1.2)),
+           legend.title = element_text(face = "bold", size = rel(1.2)),
+                   strip.text.x = element_text(face = "bold", size = rel(1.4)),
+           plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
> primeplot + stat_smooth(aes(group = PrimeCondition), method = "lm", color = "red")
```



8.4 Model 2: Raw data: No subject lines

```
> primewith_final_z$primefac = ordered(as.factor(as.character(primewith_final_z$PrimeCon
> primewith_final_z %>%
+   mutate(PrimeType = factor(primefac, levels = unique(primefac),
+                     labels = c("Both","Semantic", "Phonological",
+                                "Unrelated")))%>%
+   ggplot(aes(x = zPrimeRecogRT_trim, y = Accuracy,
+              group = PrimeType, color = PrimeType)) +
+   geom_smooth(method = "lm", se = FALSE)+
```

```
+      xlab("") + ylab ("Mean Target Accuracy")+
+    ggtitle("Experiment 3")+
+  theme_few() +
+  scale_color_manual(values = c( "lightsalmon", "red",
+                                "paleturquoise3","lightgreen"))+
+    ggtitle("Experiment 3") +
+    theme(axis.text = element_text(size = rel(1)),
+            axis.title = element_text(face = "bold", size = rel(1)),
+            legend.title = element_text(face = "bold", size = rel(1)),
+           plot.title = element_text(hjust = .5, size = rel(1)),
+           axis.text.x = element_text(face = "bold", size = rel(1.2)))
```



8.5 RTprimeRTmodel2

```
> # RTprime_RT_model_2 = lmer(data = primewith_final_z,
> #                       zTargetRecogRT_trim ~ zPrimeRecogRT_trim*PrimeCondition +
> #                             (1|Subject) + (1|Stimuli2))
> # summary(RTprime_RT_model_2)
>
> primert_model = lmer(data = primewith_final_z,
+                      zPrimeRecogRT_trim ~ 1 + (1 | Subject) +
+                        (1|Stimuli2))
> summary(primert_model)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: zPrimeRecogRT_trim ~ 1 + (1 | Subject) + (1 | Stimuli2)
   Data: primewith_final_z

REML criterion at convergence: 9341.3

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.1121 -0.7343 -0.2065  0.5818  4.1568

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli2 (Intercept) 0.06042  0.2458
 Subject  (Intercept) 0.00000  0.0000
 Residual             0.92097  0.9597
Number of obs: 3352, groups:  Stimuli2, 72; Subject, 48

Fixed effects:
            Estimate Std. Error       df t value Pr(>|t|)
(Intercept) -0.00394    0.03338 70.57723  -0.118    0.906
```

```
> VarCorr(primert_model)
```

```
 Groups   Name        Std.Dev.
 Stimuli2 (Intercept) 0.24580
 Subject  (Intercept) 0.00000
 Residual             0.95967
```

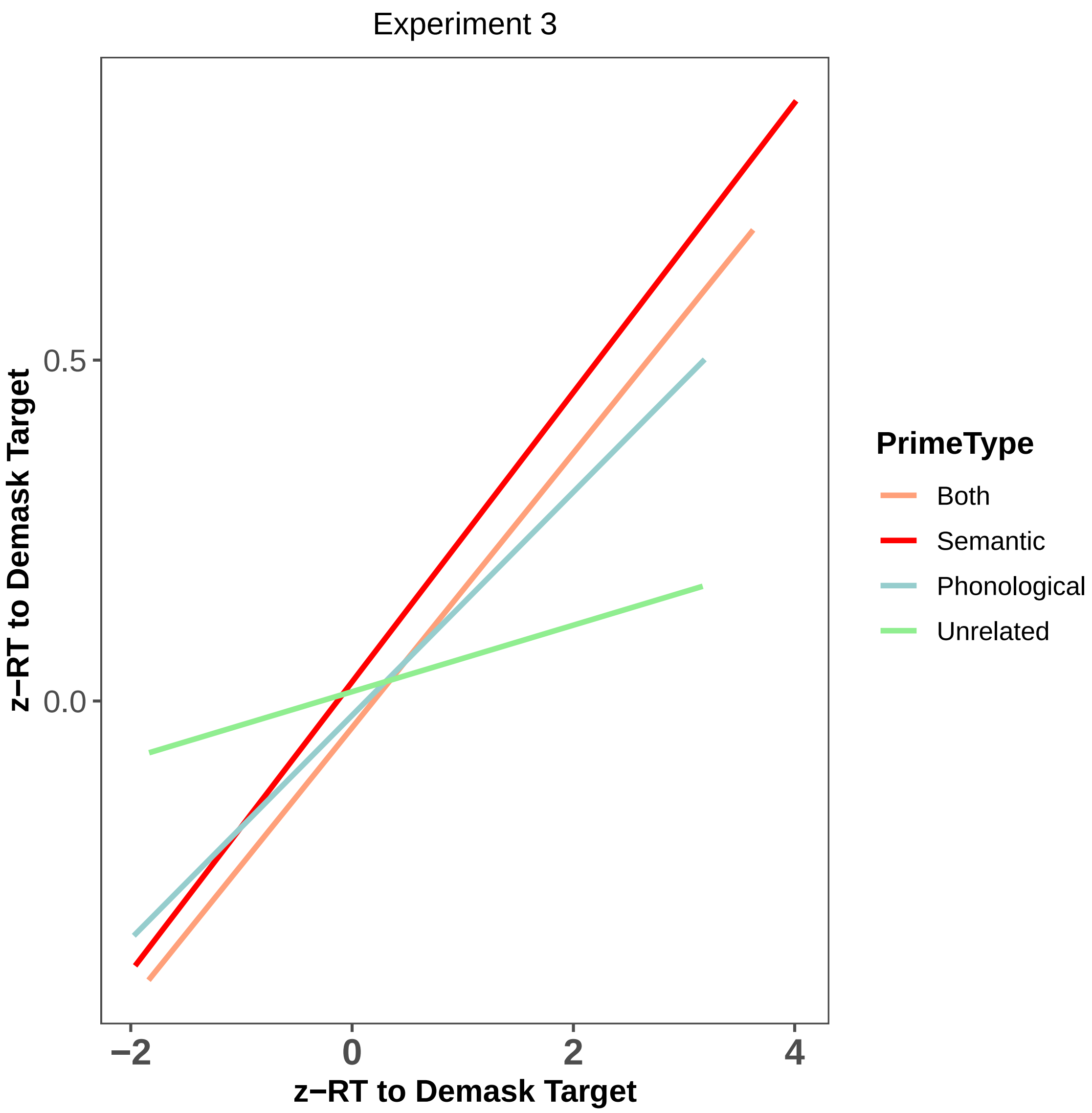
```
> SD_prime <- as.data.frame(VarCorr(primert_model))[3, 5]
> ## now we need to find increments for each prime condition
>
> primert_model_2 <- lmer(data = primewith_final_z,
+                         zPrimeRecogRT_trim ~ 1 + PrimeCondition +
+                       (1|Subject) + (1|Stimuli2))
> prime_Inc_1_U <- 0*fixef(primert_model_2)[1]
```

```
> prime_Inc_1_B ← 1*fixef(primert_model_2)[2]
> prime_Inc_1_P ← 1*fixef(primert_model_2)[3]
> prime_Inc_1_R ← 1*fixef(primert_model_2)[4]
> predict_data_U ← with(primewith_final_z,
+                   data.frame(school=1,
+  zPrimeRecogRT_trim=seq(from=-prime_Inc_1_U-SD_prime,
+          to=-prime_Inc_1_U+SD_prime,
+          by=SD_prime),
+  PrimeCondition = 0))
> predict_data_B ← with(primewith_final_z,
+                   data.frame(school=1,
+  zPrimeRecogRT_trim=seq(from=-prime_Inc_1_B-SD_prime,
+          to=-prime_Inc_1_B+SD_prime,
+          by=SD_prime),
+  PrimeCondition = 1))
> predict_data_P ← with(primewith_final_z,
+                   data.frame(school=1,
+  zPrimeRecogRT_trim=seq(from=-prime_Inc_1_P-SD_prime,
+          to=-prime_Inc_1_P+SD_prime,
+          by=SD_prime),
+  PrimeCondition = 2))
> predict_data_R ← with(primewith_final_z,
+                   data.frame(school=1,
+  zPrimeRecogRT_trim=seq(from=-prime_Inc_1_R-SD_prime,
+          to=-prime_Inc_1_R+SD_prime,
+          by=SD_prime),
+  PrimeCondition = 3))
> predict_data = rbind(predict_data_U,
+                      predict_data_B,
+                      predict_data_P,
+                      predict_data_R)
> predict_data$PrimeCondition = ifelse(predict_data$PrimeCondition == 0, "U",
+                     ifelse(predict_data$PrimeCondition == 1, "B",
+                     ifelse(predict_data$PrimeCondition == 2, "P", "R")))
> predict_data = predict_data %>%
+   mutate(predicted_values = predict(RTprime_RT_model_2,
+          newdata = predict_data, re.form = NA))
> predict_data$PrimeCondition = ordered(as.factor(as.character(predict_data$PrimeConditi
> predict_data %>%
+   mutate(PrimeType = factor(PrimeCondition, levels = unique(PrimeCondition),
+                      labels = c("Unrelated",
+                               "Both","Phonological","Semantic")))%>%
+   ggplot(aes(x = zPrimeRecogRT_trim, y = predicted_values,
+             color = PrimeType)) +
+     geom_line(size = 1) +
+     xlab("z-RT to Demask Prime") + ylab ("z-RT to Demask Target")+
+   ggtitle("Experiment 3")+
+ theme_few() +
```

```
+   scale_color_manual(values = c( "lightgreen", "lightsalmon",
+                                "paleturquoise3","red"))+
+     theme(axis.text = element_text(face = "bold", size = rel(1.2)),
+           axis.title = element_text(face = "bold", size = rel(1.2)),
+           legend.title = element_text(face = "bold", size = rel(1.2)),
+     plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
>
```

8.6 Target RT Model 2: Raw data: No subject lines

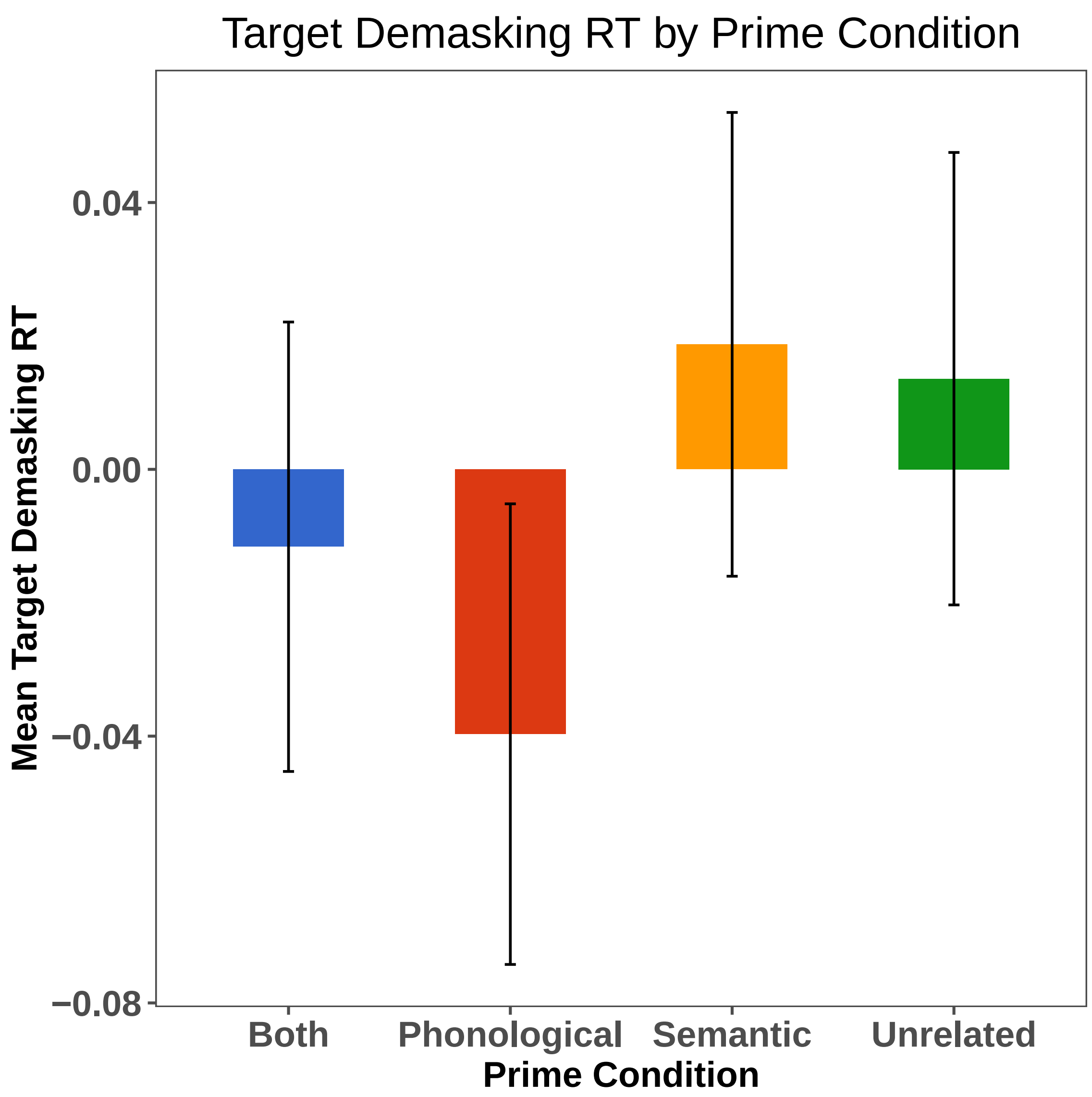
```
> primewith_final_z %>%
+   mutate(PrimeType = factor(primefac, levels = unique(primefac),
+                      labels = c("Both","Semantic", "Phonological",
+                                  "Unrelated")))%>%
+   ggplot(aes(x = zPrimeRecogRT_trim, y = zTargetRecogRT_trim,
+              group = PrimeType, color = PrimeType)) +
+   geom_smooth(method = "lm", se = FALSE, size = 1)+
+  # ylim(-0.5,0.5)+
+   #facet_wrap(~PrimeCondition, nrow = 1)+
+     xlab("z-RT to Demask Target") + ylab ("z-RT to Demask Target")+
+ theme_few() +
+ scale_color_manual(values = c( "lightsalmon", "red",
+                                "paleturquoise3","lightgreen"))+
+   ggtitle("Experiment 3") +
+   theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5, size = rel(1)),
+          axis.text.x = element_text(face = "bold", size = rel(1.2)))
```



8.7 Target RT Model 1

```
> targetRT_rmisc = Rmisc::summarySE(primewith_final_z,
+                          measurevar = "zTargetRecogRT_trim",
+                          groupvars = c("PrimeCondition"))
> library(ggplot2)
> library(ggthemes)
> targetRT_rmisc %>% mutate(`Prime Condition` = factor(PrimeCondition,
+                                                  levels = unique(PrimeCondition),
+                         labels = c("Both", "Phonological",
```

```
+                                "Semantic", "Unrelated"))) %>%
+ ggplot(aes(x = `Prime Condition`,
+           y = zTargetRecogRT_trim, fill = `Prime Condition`))+
+  geom_bar(stat = "identity", position = "dodge",
+           width = 0.5)+
+   geom_errorbar(aes(ymin = zTargetRecogRT_trim - se, ymax = zTargetRecogRT_trim + se),
+                 width=.05, position=position_dodge(.5)) +
+   theme_few()+
+   guides (fill = FALSE)+
+   scale_fill_gdocs()+
+   xlab("Prime Condition") + ylab("Mean Target Demasking RT") +
+   ggtitle("Target Demasking RT by Prime Condition") +
+     theme(axis.text = element_text(face = "bold", size = rel(1.2)),
+           axis.title = element_text(face = "bold", size = rel(1.2)),
+           legend.title = element_text(face = "bold", size = rel(1.2)),
+           plot.title = element_text( size = rel(1.4), hjust = .5))
```



9 MTurk Covariate Analyses

```
> itemratings= read.csv("Abhilasha_item_wide.csv",
+                              header = TRUE, sep = ",")
> main = PrimeRetrieval
> main = main %>% filter(PrimeCondition %in% c("P", "R"))
> main_item = merge(main, itemratings,
+                  by = c("Stimuli2", "PrimeCondition"))
> main_item = dplyr::arrange(main_item, ID, Stimuli2, PrimeType)
> ## Impacting Ret/NotRet
```

```
>
> main_item$PrimeFirstResp_ACC = as.factor(main_item$PrimeFirstResp_ACC)
> m_young_prime2 = lme4::glmer(data = main_item, Accuracy ~
+                    PrimeFirstResp_ACC*PrimeCondition  + PrimeAcc +
+                             MeaningRating +
+                           (1|Subject) + (1|Stimuli2),
+                          family = "binomial",
+                          control=glmerControl(optimizer="bobyqa",
+              optCtrl=list(maxfun=100000)))
> summary(m_young_prime2)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula:
Accuracy ~ PrimeFirstResp_ACC * PrimeCondition + PrimeAcc + MeaningRating +
    (1 | Subject) + (1 | Stimuli2)
   Data: main_item
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))

     AIC      BIC   logLik deviance df.resid
  1915.4   1959.0   -949.7   1899.4     1720

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2423 -0.5983 -0.3353  0.6753  3.2464

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli2 (Intercept) 1.4808   1.2169
 Subject  (Intercept) 0.4796   0.6926
Number of obs: 1728, groups:  Stimuli2, 72; Subject, 48

Fixed effects:
                                   Estimate Std. Error z value Pr(>|z|)
(Intercept)                       -0.869161   0.403129  -2.156   0.0311 *
PrimeFirstResp_ACC1               -0.232409   0.080057  -2.903   0.0037 **
PrimeCondition1                   -0.002185   0.168317  -0.013   0.9896
PrimeAcc                           0.119265   0.301918   0.395   0.6928
MeaningRating                     -0.014424   0.092324  -0.156   0.8759
PrimeFirstResp_ACC1:PrimeCondition1  0.142212   0.066209   2.148   0.0317 *
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Correlation of Fixed Effects:
            (Intr) PrFR_ACC1 PrmCn1 PrmAcc MnngRt
PrmFrR_ACC1 -0.200
PrimeCndtn1 -0.724  0.000
```

```
PrimeAcc     -0.411  0.537     -0.028
MeaningRtng -0.801  0.008      0.933  0.039
PFR_ACC1:PC  0.090  0.002     -0.058  0.009 -0.101
```

```
> options(contrasts = c("contr.sum","contr.poly"))
> car::Anova(m_young_prime2)
```

```
Analysis of Deviance Table (Type II Wald chisquare tests)

Response: Accuracy
                                   Chisq Df Pr(>Chisq)
PrimeFirstResp_ACC                8.4552  1    0.00364 **
PrimeCondition                    0.0125  1    0.91103
PrimeAcc                          0.1560  1    0.69282
MeaningRating                     0.0244  1    0.87585
PrimeFirstResp_ACC:PrimeCondition 4.6135  1    0.03172 *
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> anova(m_young_prime2)
```

```
Analysis of Variance Table
                                  Df  Sum Sq Mean Sq F value
PrimeFirstResp_ACC                 1 14.2595 14.2595 14.2595
PrimeCondition                     1  0.0429  0.0429  0.0429
PrimeAcc                           1  0.1428  0.1428  0.1428
MeaningRating                      1  0.0047  0.0047  0.0047
PrimeFirstResp_ACC:PrimeCondition  1  4.6981  4.6981  4.6981
```

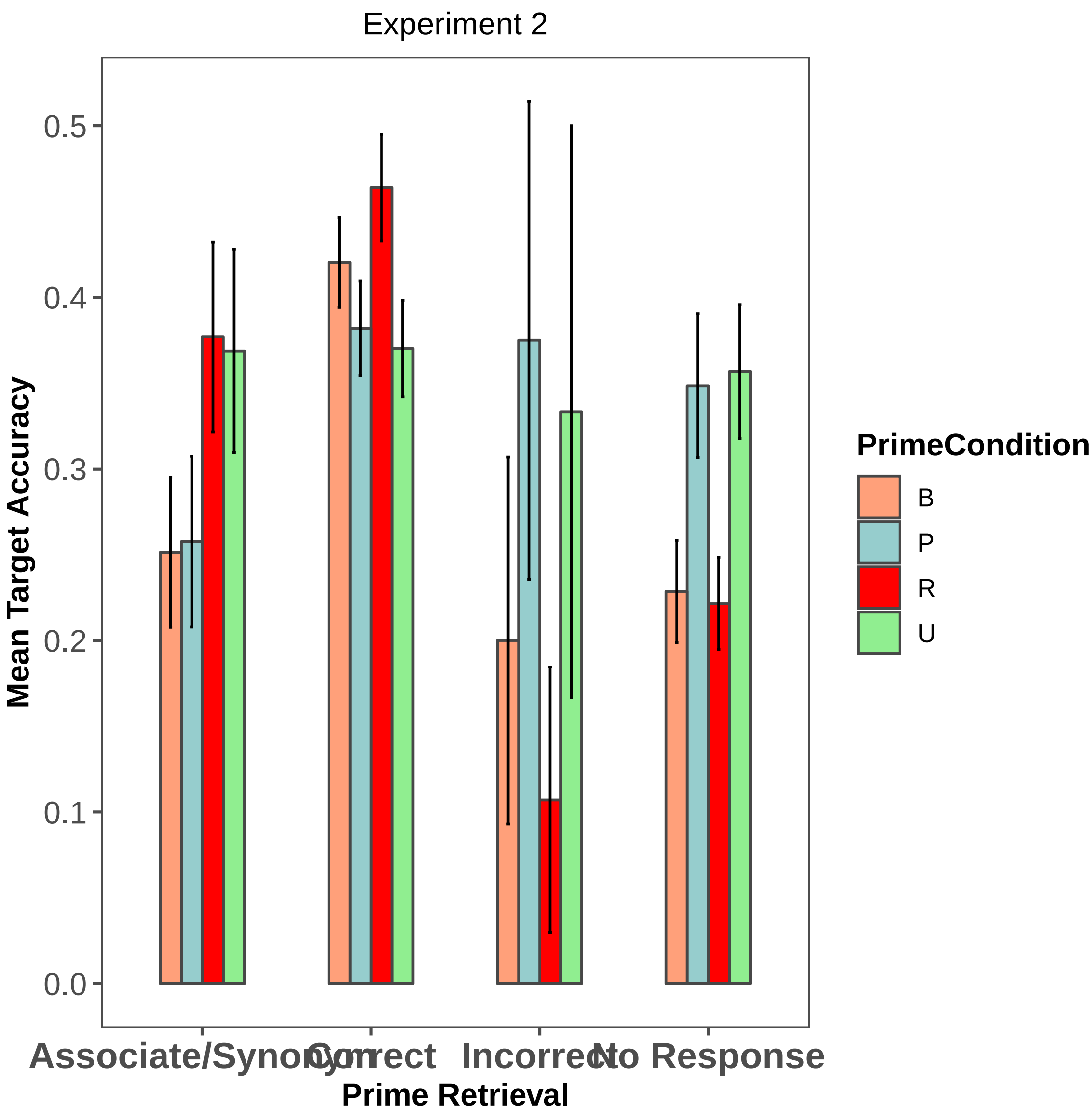
```
> #sjPlot::plot_model(m_young_prime2, type = "int")
```

10 Response Analysis

10.1 All Responses

```
> E3_YA = read.csv("E3_YA_Responses.csv",
+                        header = TRUE, sep = ",")
> E3_YA$AllResponse = ifelse(E3_YA$PrimeRespType %in%
+                              c("Associate", "Synonym"), "Associate/Synonym",
+                            ifelse(E3_YA$PrimeRespType == "NoResponse",
+                                   "No Response",
+                          ifelse(E3_YA$PrimeRespType == "Correct","Correct",
+                                 "Incorrect")))
> E3_YA_subject = group_by(E3_YA, Subject, PrimeCondition, AllResponse) %>%
+   summarize_at(vars(Accuracy), mean)
> ret_figure = Rmisc::summarySE(E3_YA_subject,
```

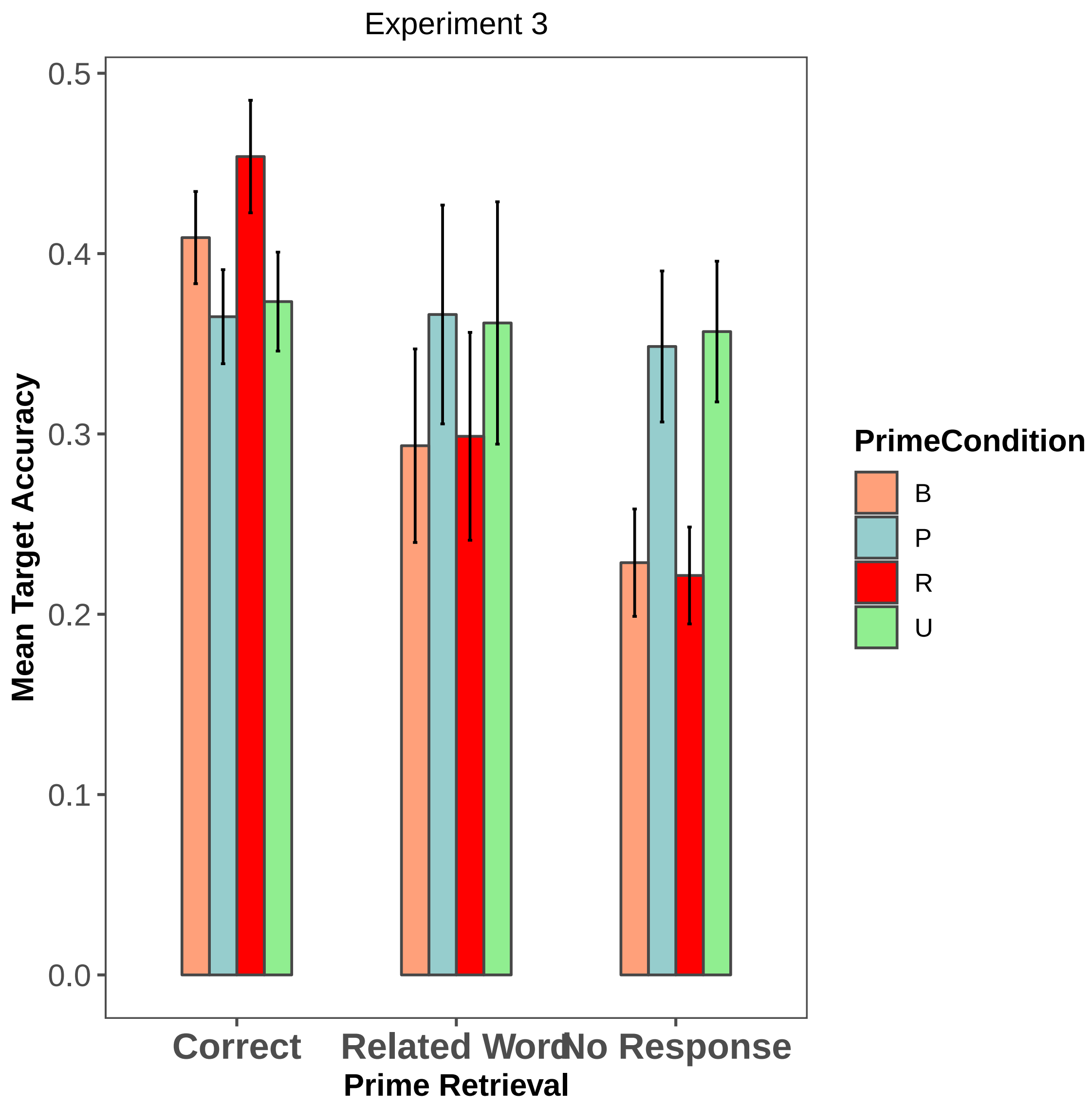
```
+                       measurevar = "Accuracy",
+                   groupvars = c("PrimeCondition", "AllResponse"))
> library(ggplot2)
> library(ggthemes)
> library(dplyr)
> ret_figure  %>%
+    ggplot(aes(x = AllResponse, y = Accuracy,
+                              group =PrimeCondition ,
+                              fill = PrimeCondition)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+    geom_errorbar(aes(ymin = Accuracy - se,
+                         ymax = Accuracy + se),
+                   width=.08, position=position_dodge(.5)) +
+   theme_few()+
+ #   scale_fill_canva()+
+  scale_fill_manual(values = c( "lightsalmon","paleturquoise3",
+                                   "red", "lightgreen"))+
+   xlab("Prime Retrieval") + ylab("Mean Target Accuracy") +
+ ggtitle("Experiment 2") +
+   theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5, size = rel(1)),
+          axis.text.x = element_text(face = "bold", size = rel(1.2)))
```



10.2 3-group Responses

```
> E3_YA = read.csv("E3_YA_Responses.csv",
+                            header = TRUE, sep = ",")
> E3_YA$Response = ifelse(E3_YA$PrimeRespType %in%
+                                    c("Associate", "Incorrect"), "Related Word",
+                                  ifelse(E3_YA$PrimeRespType == "NoResponse",
+                                         "No Response", "Correct"))
> E3_YA$Response = ordered(as.factor(as.character(E3_YA$Response)),
+                         levels = c("Correct", "Related Word", "No Response"))
```

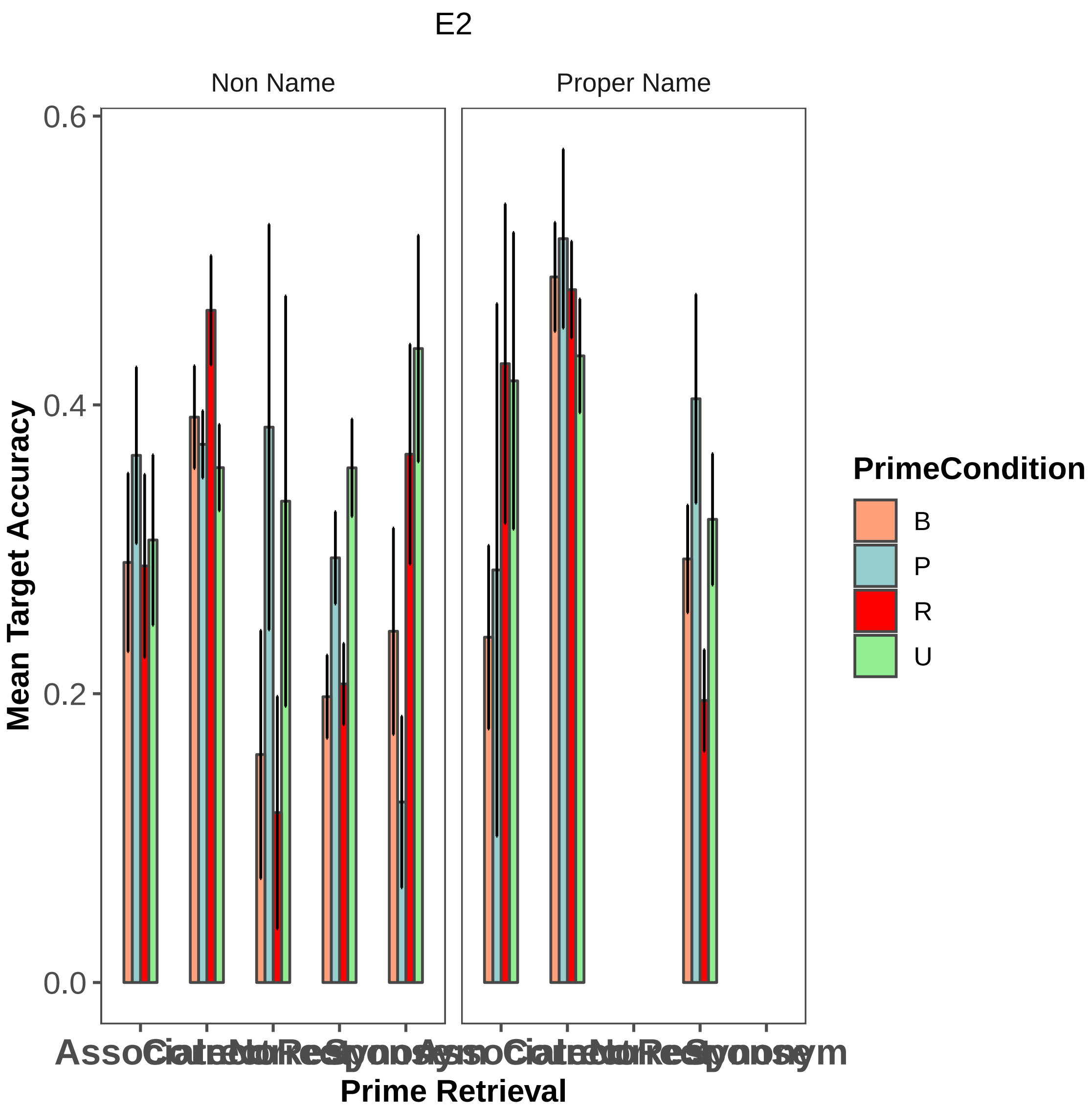
```
> E3_YA_subject = group_by(E3_YA, Subject, PrimeCondition, Response) %>%
+   summarize_at(vars(Accuracy), mean)
> ret_figure = Rmisc::summarySE(E3_YA_subject,
+                         measurevar = "Accuracy",
+                     groupvars = c("PrimeCondition", "Response"))
> library(ggplot2)
> library(ggthemes)
> library(dplyr)
> ret_figure  %>%
+    ggplot(aes(x = Response, y = Accuracy,
+                            group =PrimeCondition ,
+                            fill = PrimeCondition)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+    geom_errorbar(aes(ymin = Accuracy - se,
+                       ymax = Accuracy + se),
+                   width=.08, position=position_dodge(.5)) +
+   theme_few()+
+ #   scale_fill_canva()+
+  scale_fill_manual(values = c( "lightsalmon","paleturquoise3",
+                                 "red", "lightgreen"))+
+   xlab("Prime Retrieval") + ylab("Mean Target Accuracy") +
+ ggtitle("Experiment 3") +
+   theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5, size = rel(1)),
+          axis.text.x = element_text(face = "bold", size = rel(1.2)))
```



10.3 POS-split Responses

```
> ret_figure = Rmisc::summarySE(E3_YA,
+                         measurevar = "Accuracy",
+                     groupvars = c("Prime_POS", "PrimeCondition", "PrimeRespType"))
> library(ggplot2)
> library(ggthemes)
> library(dplyr)
> ret_figure  %>%
+    ggplot(aes(x = PrimeRespType, y = Accuracy,
```

```
+                             group =PrimeCondition ,
+                             fill = PrimeCondition)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+    geom_errorbar(aes(ymin = Accuracy - se,
+                      ymax = Accuracy + se),
+                 width=.08, position=position_dodge(.5)) +
+   theme_few()+
+ facet_wrap(~Prime_POS)+
+   scale_fill_manual(values = c( "lightsalmon","paleturquoise3",
+                                  "red", "lightgreen"))+
+   xlab("Prime Retrieval") + ylab("Mean Target Accuracy") +
+ ggtitle("E2") +
+   theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5, size = rel(1)),
+          axis.text.x = element_text(face = "bold", size = rel(1.2)))
```



10.4 LME

```
> E3_YA$Response = as.factor(E3_YA$Response)
> contrasts(E3_YA$Response) = contr.treatment(3, base = 1)
> contrasts(E3_YA$PrimeCondition) = contr.treatment(4, base = 4)
> # E3_YA$Relationship = ifelse(E3_YA$PrimeCondition %in% c("B", "P"), "Unrelated",
> #                          "Related")
> #
> # E3_YA$Relationship = as.factor(E3_YA$Relationship)
> # contrasts(E3_YA$Relationship) = contr.treatment(2, base = 2)
```

```
>
> TOTFeedback_hlm2 = glmer(data = E3_YA,
+                               Accuracy ~ PrimeCondition*Response +
+                         (1|Subject) + (1|Stimuli2), family = "binomial",
+                           control=glmerControl(optimizer="bobyqa",
+             optCtrl=list(maxfun=100000)))
> summary(TOTFeedback_hlm2)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: Accuracy ~ PrimeCondition * Response + (1 | Subject) + (1 | Stimuli2)
   Data: E3_YA
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))

     AIC      BIC   logLik deviance df.resid
  3631.4   3717.4  -1801.7   3603.4     3442

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.1100 -0.5755 -0.3081  0.6292  5.5292

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli2 (Intercept) 1.8043   1.3432
 Subject  (Intercept) 0.4739   0.6884
Number of obs: 3456, groups:  Stimuli2, 72; Subject, 48

Fixed effects:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)               -0.71061    0.22106  -3.215  0.00131 **
PrimeCondition1           -0.06505    0.16804  -0.387  0.69865
PrimeCondition2           -0.16015    0.15944  -1.004  0.31516
PrimeCondition3            0.03629    0.16480   0.220  0.82571
Response2                 -0.07667    0.27805  -0.276  0.78275
Response3                 -0.12719    0.18650  -0.682  0.49526
PrimeCondition1:Response2 -0.33413    0.39133  -0.854  0.39320
PrimeCondition2:Response2  0.25382    0.40774   0.623  0.53361
PrimeCondition3:Response2 -0.18625    0.41057  -0.454  0.65009
PrimeCondition1:Response3 -0.41870    0.27227  -1.538  0.12409
PrimeCondition2:Response3  0.18800    0.27301   0.689  0.49107
PrimeCondition3:Response3 -0.52886    0.27096  -1.952  0.05096 .
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Correlation of Fixed Effects:
            (Intr) PrmCn1 PrmCn2 PrmCn3 Rspns2 Rspns3 PC1:R2 PC2:R2 PC3:R2
PrimeCndtn1 -0.360
```

```
PrimeCndtn2 -0.382  0.514
PrimeCndtn3 -0.365  0.492  0.518
Response2   -0.222  0.295  0.305  0.296
Response3   -0.340  0.446  0.472  0.450  0.263
PrmCndt1:R2  0.159 -0.447 -0.228 -0.217 -0.713 -0.198
PrmCndt2:R2  0.150 -0.212 -0.400 -0.206 -0.677 -0.182  0.494
PrmCndt3:R2  0.147 -0.204 -0.211 -0.418 -0.675 -0.177  0.484  0.462
PrmCndt1:R3  0.227 -0.645 -0.330 -0.312 -0.183 -0.677  0.288  0.137  0.127
PrmCndt2:R3  0.231 -0.316 -0.618 -0.314 -0.176 -0.684  0.140  0.244  0.124
PrmCndt3:R3  0.227 -0.308 -0.326 -0.632 -0.180 -0.676  0.136  0.128  0.264
            PC1:R3 PC2:R3
PrimeCndtn1
PrimeCndtn2
PrimeCndtn3
Response2
Response3
PrmCndt1:R2
PrmCndt2:R2
PrmCndt3:R2
PrmCndt1:R3
PrmCndt2:R3  0.476
PrmCndt3:R3  0.465  0.471
```

```
> sjPlot::plot_model(TOTFeedback_hlm2, type = "int")
> car::Anova(TOTFeedback_hlm2)
```

```
Analysis of Deviance Table (Type II Wald chisquare tests)

Response: Accuracy
                         Chisq Df Pr(>Chisq)
PrimeCondition          4.7464  3   0.191334
Response                9.4662  2   0.008799 **
PrimeCondition:Response 9.9930  6   0.124946
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

10.5 Specific Comparisons

```
> ## correct responses
> responses_correct = E3_YA %>% filter(Response == "Correct")
> responses_correct_sub = group_by(responses_correct, Subject, PrimeCondition) %>%
+   summarise_at(vars(Accuracy), mean)
> responses_correct_wide = spread(responses_correct_sub, PrimeCondition, Accuracy)
> t.test(responses_correct_wide$R, responses_correct_wide$U,
+        paired = TRUE)
```

```
        Paired t-test

data:  responses_correct_wide$R and responses_correct_wide$U
t = 2.116, df = 47, p-value = 0.03967
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.003963509 0.156902708
sample estimates:
mean of the differences
             0.08043311
```

```
> ## other responses
> responses_other = E3_YA %>% filter(Response == "Related Word")
> responses_other_sub = group_by(responses_other, Subject, PrimeCondition) %>%
+   summarise_at(vars(Accuracy), mean)
> responses_other_wide = spread(responses_other_sub, PrimeCondition, Accuracy)
> t.test(responses_other_wide$R, responses_other_wide$U,
+        paired = TRUE)
```

```
        Paired t-test

data:  responses_other_wide$R and responses_other_wide$U
t = -0.36576, df = 31, p-value = 0.717
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.2055052  0.1430052
sample estimates:
mean of the differences
               -0.03125
```

```
> t.test(responses_other_wide$B, responses_other_wide$U,
+        paired = TRUE)
```

```
        Paired t-test

data:  responses_other_wide$B and responses_other_wide$U
t = -1.1901, df = 34, p-value = 0.2423
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.26928897  0.07037741
sample estimates:
mean of the differences
            -0.09945578
```

```
> t.test(responses_other_wide$P, responses_other_wide$U,
+        paired = TRUE)
```

```
	Paired t-test

data:  responses_other_wide$P and responses_other_wide$U
t = 0.32673, df = 31, p-value = 0.7461
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.1392471  0.1923721
sample estimates:
mean of the differences
              0.0265625
```

```
> responses_none = E3_YA %>% filter(Response == "No Response")
> ## no response
> responses_none_sub = group_by(responses_none, Subject, PrimeCondition) %>%
+   summarise_at(vars(Accuracy), mean)
> responses_none_wide = spread(responses_none_sub, PrimeCondition, Accuracy)
> t.test(responses_none_wide$R, responses_none_wide$U,
+        paired = TRUE)
```

```
	Paired t-test

data:  responses_none_wide$R and responses_none_wide$U
t = -3.51, df = 45, p-value = 0.001031
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.19467697 -0.05271802
sample estimates:
mean of the differences
             -0.1236975
```