Repeated Lexical Retrieval: Experiment 1

Abhilasha Kumar

June 6, 2019

1 Reading and Formatting Data

```
> TOT = read.csv("CompiledPrimeFlash.csv", header = TRUE, sep = ",")
```

2 Accuracy per Prime Condition

```
> library(dplyr)
> overall_acc = group_by(TOT) %>%
+   summarise_at(vars(TargetAccuracy), mean)
> overall_acc_subject = group_by(TOT, Subject) %>%
+   summarise_at(vars(TargetAccuracy), mean)
> prime_acc = group_by(TOT, PrimeCondition) %>%
+   summarise_at(vars(TargetAccuracy), mean)
> prime_subject_acc = group_by(TOT, Subject, PrimeCondition) %>%
+   summarise_at(vars(TargetAccuracy), mean)
```

ANOVA

```
> prime_subject_acc$PrimeCondition = as.factor(prime_subject_acc$PrimeCondition)
> prime_subject_acc$Subject = as.factor(prime_subject_acc$Subject)
> #prime_subject_acc = prime_subject_acc %>% filter(!Subject %in% c(9,31,32))
> # this is eliminating subjects <50 % accuracy
> target_aov= aov(data = prime_subject_acc, TargetAccuracy ~ PrimeCondition +
+                                Error(Subject/PrimeCondition))
> summary(target_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 35   2.69 0.07687

Error: Subject:PrimeCondition
                Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition   3 0.1662 0.05541   3.879 0.0113 *
Residuals      105 1.4997 0.01428
```

```
---
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'))
> library(lsmeans)
> library(multcomp)
> imm_lsm = lsmeans::lsmeans(target_aov, c("PrimeCondition"))
> prime_effect = cld(imm_lsm, alpha = 0.05,
+                    adjust = "tukey", details = TRUE)
> library(knitr)
> kable(subset(prime_effect$comparisons,prime_effect$comparisons$p.value < 0.5 ))
```

```
|   |contrast |  estimate|         SE|  df|  t.ratio|   p.value|
|:--|:--------|---------:|---------:|---:|--------:|---------:|
|4  |P - U    | 0.0879630| 0.0281688| 105| 3.122710| 0.0122195|
|5  |P - B    | 0.0740741| 0.0281688| 105| 2.629651| 0.0476518|
|6  |P - R    | 0.0679012| 0.0281688| 105| 2.410513| 0.0812979|
```

```
> ### SPECIFIC T TESTS
> target_p = prime_subject_acc %>% filter(PrimeCondition == "P")
> target_r = prime_subject_acc %>% filter(PrimeCondition == "R")
> target_b = prime_subject_acc %>% filter(PrimeCondition == "B")
> target_u = prime_subject_acc %>% filter(PrimeCondition == "U")
> t.test(target_p$TargetAccuracy, target_r$TargetAccuracy, paired = TRUE)
```

```
        Paired t-test

data:  target_p$TargetAccuracy and target_r$TargetAccuracy
t = 2.2922, df = 35, p-value = 0.02802
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.007764809 0.128037660
sample estimates:
mean of the differences
             0.06790123
```

```
> t.test(target_p$TargetAccuracy, target_b$TargetAccuracy, paired = TRUE)
```

```
        Paired t-test

data:  target_p$TargetAccuracy and target_b$TargetAccuracy
t = 2.6294, df = 35, p-value = 0.01262
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.01688226 0.13126589
sample estimates:
mean of the differences
             0.07407407
```

```
> t.test(target_p$TargetAccuracy, target_u$TargetAccuracy, paired = TRUE)
```

```
        Paired t-test

data:  target_p$TargetAccuracy and target_u$TargetAccuracy
t = 2.8432, df = 35, p-value = 0.007407
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.02515556 0.15077037
sample estimates:
mean of the differences
             0.08796296
```

```
> t.test(target_b$TargetAccuracy, target_r$TargetAccuracy, paired = TRUE)
```

```
        Paired t-test

data:  target_b$TargetAccuracy and target_r$TargetAccuracy
t = -0.19831, df = 35, p-value = 0.844
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.06936528  0.05701960
sample estimates:
mean of the differences
            -0.00617284
```

```
> t.test(target_b$TargetAccuracy, target_u$TargetAccuracy, paired = TRUE)
```

```
        Paired t-test

data:  target_b$TargetAccuracy and target_u$TargetAccuracy
t = 0.56266, df = 35, p-value = 0.5773
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.03622324  0.06400102
sample estimates:
mean of the differences
             0.01388889
```

```
> t.test(target_r$TargetAccuracy, target_u$TargetAccuracy, paired = TRUE)
```

```
        Paired t-test

data:  target_r$TargetAccuracy and target_u$TargetAccuracy
t = 0.85208, df = 35, p-value = 0.4
alternative hypothesis: true difference in means is not equal to 0
```

```
95 percent confidence interval:
 -0.02773619  0.06785964
sample estimates:
mean of the differences
             0.02006173
```

3 Item Analyses

3.1 Prime And Target Accuracy

```
> library(dplyr)
> agg_item_condition ←  group_by(TOT, Stimuli1, PrimeCondition)%>%
+     summarise_at(vars(TargetAccuracy), mean)
> agg_item_condition$Stimuli1 ← as.factor(agg_item_condition$Stimuli1)
> agg_item_condition$PrimeCondition ← as.factor(agg_item_condition$PrimeCondition)
> ## target accuracy anova
>
> item_prime_aov = aov(data = agg_item_condition, TargetAccuracy ∼ PrimeCondition +
+                                        Error(Stimuli1/PrimeCondition))
> summary(item_prime_aov)
```

```
Error: Stimuli1
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 71  12.86  0.1812

Error: Stimuli1:PrimeCondition
                Df Sum Sq Mean Sq F value   Pr(>F)
PrimeCondition   3  0.332 0.11081   6.063 0.000559 ***
Residuals      213  3.893 0.01828
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

4 LME

```
> library(lme4)
> contrasts(TOT$PrimeCondition)= contr.treatment(4, base = 4)
> # prime_lmer = glmer(data = TOT, TargetAccuracy ∼ PrimeCondition +
> #                      (PrimeCondition|Subject) + (PrimeCondition|Stimuli1),
> #                          family = "binomial",
> #                          control=glmerControl(optimizer="bobyqa",
> #              optCtrl=list(maxfun=100000)))
> # summary(prime_lmer)
>
> prime_lmer2 = glmer(data = TOT, TargetAccuracy ∼ PrimeCondition +
+                        (1|Subject) + (1|Stimuli1),
+                              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: TargetAccuracy ~ PrimeCondition + (1 | Subject) + (1 | Stimuli1)
   Data: TOT
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))

     AIC      BIC   logLik deviance df.resid
  2615.9   2651.1  -1301.9   2603.9     2586

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.9787 -0.5382 -0.3010  0.5883  5.0043

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli1 (Intercept) 1.6829   1.2973
 Subject  (Intercept) 0.8583   0.9265
Number of obs: 2592, groups:  Stimuli1, 72; Subject, 36

Fixed effects:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      -1.4432     0.2432  -5.934 2.95e-09 ***
PrimeCondition1   0.1228     0.1444   0.850    0.395
PrimeCondition2   0.5883     0.1424   4.132 3.59e-05 ***
PrimeCondition3   0.1389     0.1446   0.961    0.337
---
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 -0.304
PrimeCndtn2 -0.318  0.515
PrimeCndtn3 -0.302  0.504  0.513
```

```
> #anova(prime_lmer, prime_lmer2)
>
> # > confint(prime_lmer2) with unrelated as reference
> # Computing profile confidence intervals ...
> #                     2.5 %     97.5 %
> # .sig01          1.0694874  1.5928529
> # .sig02          0.7098314  1.2372008
> # (Intercept)    -1.9339944 -0.9678982
> # PrimeCondition1 -0.1644928  0.4104807
> # PrimeCondition2  0.3067886  0.8734067
> # PrimeCondition3 -0.1484873  0.4268885
>
> ## random slopes for prime condition not needed
>
> # prime_lmer3 = glmer(data = TOT, TargetAccuracy ~ PrimeCondition +
> #                    (1|Subject),
> #                          family = "binomial",
> #                          control=glmerControl(optimizer="bobyqa",
> #               optCtrl=list(maxfun=100000)))
> # summary(prime_lmer3)
> # anova(prime_lmer3, prime_lmer2)
> # ## random intercept for item needed
> #
> # prime_lmer4 = glmer(data = TOT, TargetAccuracy ~ PrimeCondition +
> #                    (1|Stimuli1),
> #                          family = "binomial",
> #                          control=glmerControl(optimizer="bobyqa",
> #               optCtrl=list(maxfun=100000)))
> # summary(prime_lmer4)
> # anova(prime_lmer4, prime_lmer2)
> ## random intercept for subject also needed
>
> ## primelmer2 is the best model.
>
```

4.1 No Intercept Model

```
> TOT$Sem = ifelse(TOT$PrimeCondition == "R", 1,0)
> TOT$Both = ifelse(TOT$PrimeCondition == "B", 1,0)
> TOT$Phon = ifelse(TOT$PrimeCondition == "P", 1,0)
> TOT$Unrelated = ifelse(TOT$PrimeCondition == "U", 1,0)
> prime_lmer_nointercept = glmer(data = TOT, TargetAccuracy ~ - 1 + Sem + Both +
+                       Phon + Unrelated+
+                      (1|Subject) + (1|Stimuli1),
+                          family = "binomial",
+                          control=glmerControl(optimizer="bobyqa",
+               optCtrl=list(maxfun=100000)))
> summary(prime_lmer_nointercept)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: TargetAccuracy ~ -1 + Sem + Both + Phon + Unrelated + (1 | Subject) +
    (1 | Stimuli1)
   Data: TOT
```

```
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))

     AIC      BIC   logLik deviance df.resid
  2615.9   2651.1  -1301.9   2603.9     2586

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.9787 -0.5382 -0.3010  0.5883  5.0043

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli1 (Intercept) 1.6829   1.2973
 Subject  (Intercept) 0.8583   0.9265
Number of obs: 2592, groups:  Stimuli1, 72; Subject, 36

Fixed effects:
          Estimate Std. Error z value Pr(>|z|)
Sem        -1.3042     0.2424  -5.380 7.47e-08 ***
Both       -1.3203     0.2421  -5.453 4.96e-08 ***
Phon       -0.8549     0.2395  -3.569 0.000359 ***
Unrelated  -1.4432     0.2432  -5.934 2.95e-09 ***
---
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:
          Sem   Both  Phon
Both      0.823
Phon      0.828 0.828
Unrelated 0.823 0.823 0.826
```

```
> exp(fixef(prime_lmer_nointercept))
```

```
      Sem      Both      Phon Unrelated
0.2713794 0.2670448 0.4253302 0.2361796
```

```
> # exp(confint(prime_lmer_nointercept))
> #
> # > exp(confint(prime_lmer_nointercept))
> # Computing profile confidence intervals ...
> #                 2.5 %    97.5 %
> # .sig01     2.9138853 4.9177590
> # .sig02     2.0336484 3.4459541
> # Sem        0.1663160 0.4361592
> # Both       0.1638056 0.4290484
> # Phon       0.2629295 0.6811378
> # Unrelated 0.1444296 0.3799235
>
> C_prime_main ← matrix(c(1, 0, -1, 0,
+                          0,1, -1,0,
+                          0,0, 1, -1,
+                          1,0,0,-1,
+                          0,1,0,-1), nrow = 5, ncol = 4, byrow = TRUE)
> rownames(C_prime_main) ← c("Sem vs Phon Effect",
+                             "Both vs Phon Effect",
+                             "Phon vs Unrelated Effect",
+                             "Sem vs Unrelated Effect",
+                             "Both vs Unrelated Effect")
> glht_prime_main ← multcomp::glht(prime_lmer_nointercept, linfct = C_prime_main,
+               alternative = "two.sided", rhs = 0)
> summary(glht_prime_main, adjusted(type = "holm"))
```

```
         Simultaneous Tests for General Linear Hypotheses

Fit: glmer(formula = TargetAccuracy ∼ -1 + Sem + Both + Phon + Unrelated +
    (1 | Subject) + (1 | Stimuli1), data = TOT, family = "binomial",
    control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05)))

Linear Hypotheses:
                              Estimate Std. Error z value Pr(>|z|)
Sem vs Phon Effect == 0        -0.4493     0.1415  -3.175  0.00450 **
Both vs Phon Effect == 0       -0.4654     0.1412  -3.297  0.00391 **
Phon vs Unrelated Effect == 0   0.5883     0.1424   4.132  0.00018 ***
Sem vs Unrelated Effect == 0    0.1389     0.1446   0.961  0.67315
Both vs Unrelated Effect == 0   0.1228     0.1444   0.850  0.67315
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
(Adjusted p values reported -- holm method)
```

```
>
```

5 State Analysis

```
> library(dplyr)
> overall_state = dplyr::group_by(TOT, State) %>%
+   summarize(count = n())
> overall_state_subject = dplyr::group_by(TOT,Subject, State) %>%
+   summarize(count = n())
> prime_state = group_by(TOT, PrimeCondition, State) %>%
+   summarize(count = n())
> prime_subject_state = group_by(TOT, Subject, PrimeCondition, State) %>%
+   summarize(count = n())
```

5.1 Raw State ANOVA

```
> overall_state_subject$Subject = as.factor(overall_state_subject$Subject)
> overall_state_subject$State = as.factor(overall_state_subject$State)
> ##anova
> state_aov = aov(data = prime_subject_state, count ~ State +
+                                Error(Subject/(PrimeCondition*State)))
> summary(state_aov)
```

```
Error: Subject
      Df Sum Sq Mean Sq
State  1  3.499   3.499

Error: Subject:PrimeCondition
          Df Sum Sq Mean Sq F value Pr(>F)
State      1  3.858   3.858   16.74 0.0549 .
Residuals  2  0.461   0.231
---
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
      Df Sum Sq Mean Sq
State  1  701.9   701.9

Error: Subject:PrimeCondition:State
          Df Sum Sq Mean Sq F value Pr(>F)
State      1  0.072  0.0718   0.023  0.893
Residuals  2  6.244  3.1222

Error: Within
           Df Sum Sq Mean Sq F value   Pr(>F)
State       1    339   338.8    49.4 6.47e-12 ***
Residuals 528   3621     6.9
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

5.2 State by prime ANOVA

```
> prime_subject_state$PrimeCondition = as.factor(prime_subject_state$PrimeCondition)
> prime_subject_state$Subject = as.factor(prime_subject_state$Subject)
> prime_subject_state$State = as.factor(prime_subject_state$State)
> ##anova
> stateprime_aov = aov(data = prime_subject_state, count ~ PrimeCondition*State +
+                                Error(Subject/(PrimeCondition*State)))
> summary(stateprime_aov)
```

```
Error: Subject
                     Df Sum Sq Mean Sq   F value Pr(>F)    
PrimeCondition        3  21.34    7.11 9.846e+26 <2e-16 ***
State                 3 209.68   69.89 9.675e+27 <2e-16 ***
PrimeCondition:State  4   0.00    0.00 3.890e-01  0.815    
Residuals            25   0.00    0.00                     
---
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
                     Df Sum Sq Mean Sq   F value   Pr(>F)    
PrimeCondition        3   6.38    2.13 7.024e+25  < 2e-16 ***
State                 3 126.75   42.25 1.395e+27  < 2e-16 ***
PrimeCondition:State  7   0.00    0.00 5.080e+00 6.29e-05 ***
Residuals            95   0.00    0.00                       
---
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
                     Df Sum Sq Mean Sq F value   Pr(>F)    
State                 3  994.0   331.3  19.747 5.21e-10 ***
PrimeCondition:State  8  333.7    41.7   2.486   0.0171 *  
Residuals            94 1577.2    16.8                     
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:PrimeCondition:State
                      Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition:State   9   60.5   6.724   1.398  0.188
Residuals            280 1346.5   4.809               
```

5.3 Percentage State Prime Analysis

```
> state = read.csv("TOTPrimeFlash_agg.csv",header = TRUE, sep = ",")
> j_statepercent = state[,c(1,39:54)] # 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[,-2]
> colnames(statepercent) = c( "Subject",
+                             "PrimeCondition", "State", "Percent")
> 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 35 2.672e-18 7.634e-20

Error: Subject:PrimeCondition
                Df    Sum Sq   Mean Sq F value Pr(>F)
PrimeCondition   3 4.700e-20 1.563e-20   0.211  0.888
Residuals      105 7.766e-18 7.396e-20

Error: Subject:State
           Df Sum Sq Mean Sq F value   Pr(>F)
State       3  4.688   1.563      22 3.93e-11 ***
Residuals 105  7.457   0.071
---
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  0.193 0.02148   1.459  0.162
Residuals            315  4.637 0.01472
```

5.3.1 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("E1")  +
+    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
```

5.3.2 know

```
> e1_know = statepercent %>% filter(State == "know")
> e1_know_aov = aov(data = e1_know,
+                            Percent ~ PrimeCondition +
+                          Error(Subject/PrimeCondition))
> summary(e1_know_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 35  2.267 0.06477

Error: Subject:PrimeCondition
                Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition   3 0.1072 0.03572   1.967  0.123
Residuals      105 1.9067 0.01816
```

5.3.3 dont know

```
> e1_dontknow = statepercent %>% filter(State == "dontknow")
> e1_dontknow_aov = aov(data = e1_dontknow,
+                            Percent ~ PrimeCondition +
+                          Error(Subject/PrimeCondition))
> summary(e1_dontknow_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 35  3.318 0.09481

Error: Subject:PrimeCondition
                Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition   3 0.0796 0.02654   2.595 0.0564 .
Residuals      105 1.0739 0.01023
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> e1_dk_2 = e1_dontknow[,-3]
> e1_dk_wide = tidyr::spread(e1_dk_2, PrimeCondition, Percent)
> t.test(e1_dk_wide$r,e1_dk_wide$u, paired = TRUE)
```

```
        Paired t-test

data:  e1_dk_wide$r and e1_dk_wide$u
t = -3.5909, df = 35, p-value = 0.001001
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.10145749 -0.02817214
sample estimates:
mean of the differences
            -0.06481481
```

```
> t.test(e1_dk_wide$r,e1_dk_wide$b, paired = TRUE)
```

```
        Paired t-test

data:  e1_dk_wide$r and e1_dk_wide$b
t = -1.7611, df = 35, p-value = 0.08696
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.09634315  0.00683698
sample estimates:
mean of the differences
            -0.04475309
```

```
> t.test(e1_dk_wide$r,e1_dk_wide$p, paired = TRUE)
```

```
        Paired t-test

data:  e1_dk_wide$r and e1_dk_wide$p
t = -1.6575, df = 35, p-value = 0.1064
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.089266507  0.009019594
sample estimates:
mean of the differences
            -0.04012346
```

```
>
```

5.3.4 other

```
> e1_other = statepercent %>% filter(State == "other")
> e1_other_aov = aov(data = e1_other,
+                           Percent ~ PrimeCondition +
+                         Error(Subject/PrimeCondition))
> summary(e1_other_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 35  1.238 0.03537

Error: Subject:PrimeCondition
                Df Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition   3 0.0023 0.000764   0.091  0.965
Residuals      105 0.8827 0.008407
```

5.3.5 TOT

```
> e1_TOT = statepercent %>% filter(State == "TOT")
> e1_TOT_aov = aov(data = e1_TOT,
+                           Percent ~ PrimeCondition +
+                         Error(Subject/PrimeCondition))
> summary(e1_TOT_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 35 0.6337  0.0181

Error: Subject:PrimeCondition
                Df Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition   3 0.0042 0.001400    0.19  0.903
Residuals      105 0.7736 0.007367
```

6 Cond TOT Analysis

```
> TOT$TOTmeasure = ifelse(TOT$State == "4" ,"TOT",
+                 ifelse(TOT$State == "1" &
+                          TOT$TargetAccuracy == "0","incorrectKnow",
+                ifelse(TOT$State == "2","dontknow",
+              ifelse(TOT$State == "3"&
+                          TOT$TargetAccuracy == "0","incorrectOther","NA"))))
> age_statedata = group_by(TOT, Subject, PrimeCondition, TOTmeasure) %>%
+   summarise(Trials = n())
> library(tidyr)
> age_statedata_wide = spread(age_statedata, TOTmeasure, Trials)
> age_statedata_wide$TOT = ifelse(is.na(age_statedata_wide$TOT),0,
+                                        age_statedata_wide$TOT)
> age_statedata_wide$incorrectKnow = ifelse(is.na(age_statedata_wide$incorrectKnow),0,
age_statedata_wide$incorrectKnow)
> age_statedata_wide$dontknow = ifelse(is.na(age_statedata_wide$dontknow),0,
+                                        age_statedata_wide$dontknow)
> age_statedata_wide$incorrectOther = ifelse(is.na(age_statedata_wide$incorrectOther),0,
age_statedata_wide$incorrectOther)
> age_statedata_wide = mutate(age_statedata_wide,
+                             propTOT = TOT/(TOT + dontknow +
+                                            incorrectKnow +
+                                              incorrectOther))
> age_statedata_wide$Subject = as.factor(age_statedata_wide$Subject)
> propTOT_aov = aov(data = age_statedata_wide, propTOT ~ PrimeCondition +
+                    Error(Subject/PrimeCondition))
> summary(propTOT_aov)
```

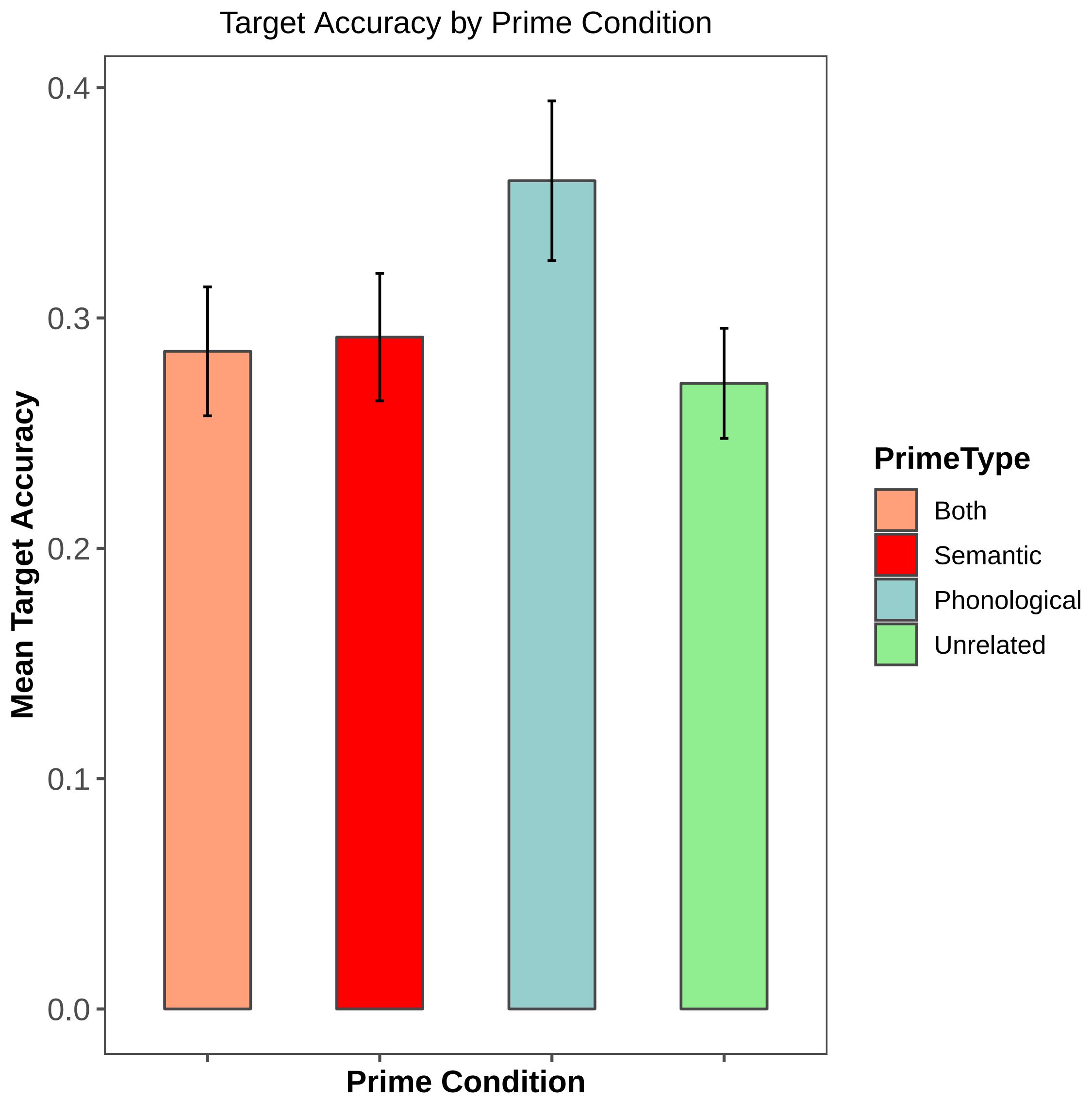
```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 35  1.282 0.03662

Error: Subject:PrimeCondition
                Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition   3 0.0023 0.00076   0.045  0.987
Residuals      105 1.7773 0.01693
```

```
> ## plotting this proportion ## remove subject from dply code
> successTOT_plot_rmisc = Rmisc::summarySE(age_statedata_wide,
+                                           measurevar = "propTOT",
+                                        groupvars = c("PrimeCondition"))
> successTOT_plot = successTOT_plot_rmisc %>%
+ ggplot(aes(x = PrimeCondition, y = propTOT))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.7,
+           color= "black")+
+   geom_errorbar(aes(ymin=propTOT - se, ymax=propTOT + se),
+              width=.2, color = "gray26",
+              position = position_dodge(0.7))+
+  theme_few()+
+ scale_fill_calc()+
+   xlab("Experiment") + ylab("Mean Proportion of TOTs") +
+   ggtitle("TOTs as a proportion of unsuccessful retrievals")  +
+    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)))
> successTOT_plot
```

7 Figures

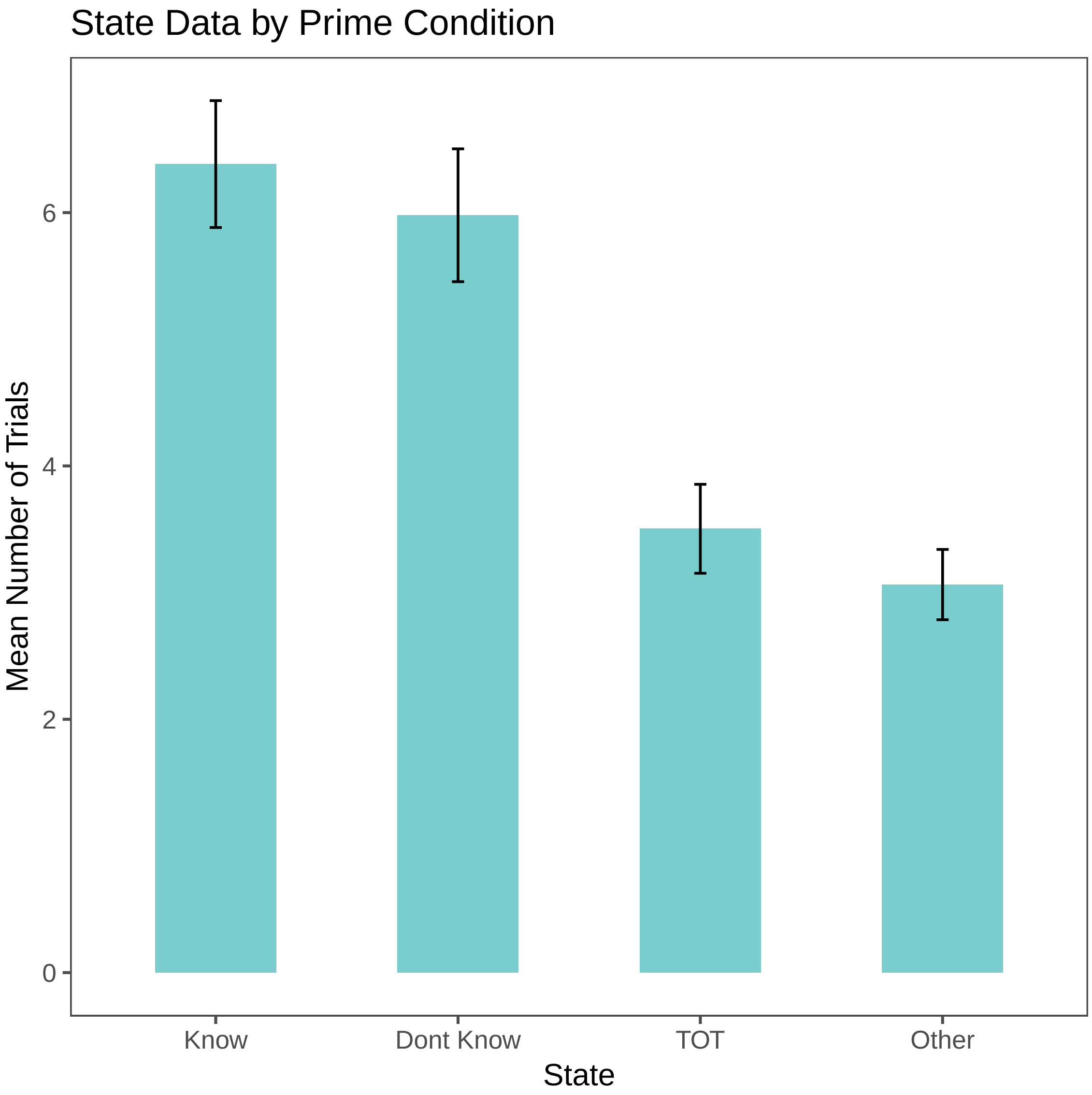
Target Accuracy Figure

```
> prime_subject_acc$primefac = ordered(as.factor(as.character(prime_subject_acc$PrimeCon
> target_rmisc = Rmisc::summarySE(prime_subject_acc,
+                       measurevar = "TargetAccuracy",
+                       groupvars = c("primefac"))
> library(ggplot2)
> library(ggthemes)
> target_rmisc %>% mutate(PrimeType = factor(primefac,
+                                                  levels = unique(primefac),
+                     labels = c("Both", "Semantic",
+                                "Phonological", "Unrelated"))) %>%
+ ggplot(aes(x = PrimeType, y = TargetAccuracy, fill = PrimeType))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+   geom_errorbar(aes(ymin = TargetAccuracy - se, ymax = TargetAccuracy + se),
+                 width=.05, position=position_dodge(.5)) +
+   theme_few()+
+ scale_fill_manual(values = c( "lightsalmon", "red",
+                                "paleturquoise3","lightgreen"))+
+   xlab("Prime Condition") + ylab("Mean Target Accuracy") +
+   ggtitle("Target Accuracy by 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, size = rel(1)),
+         strip.text.x = element_blank(),
+          axis.text.x =element_blank())
```



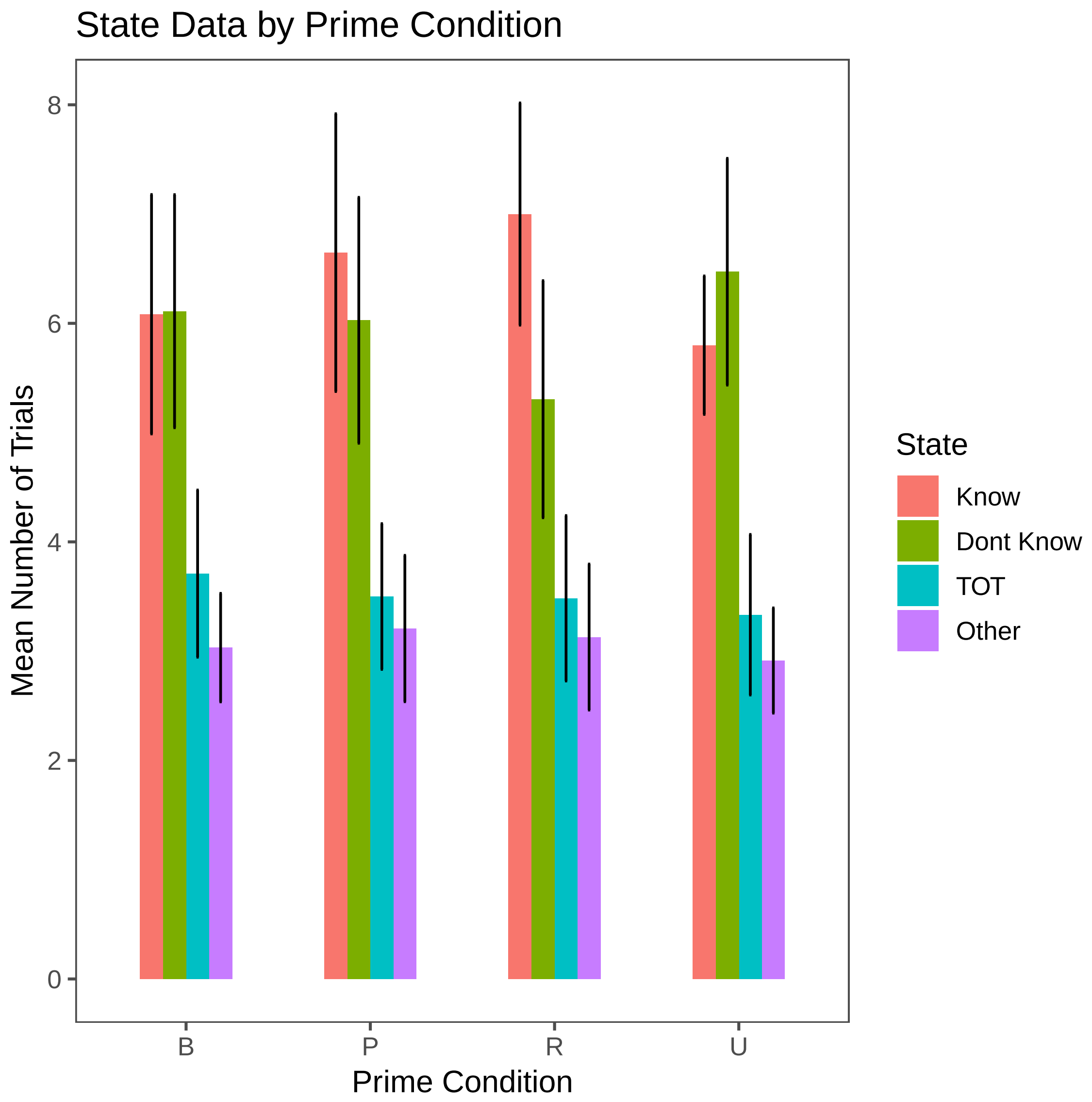
Raw State Data

```
> state_overall = Rmisc::summarySE(prime_subject_state,
+                           measurevar = "count",
+                           groupvars = c("State"))
> library(ggplot2)
> library(ggthemes)
> state_overall %>% mutate(State = factor(State, levels = unique(State),
+                               labels = c("Know", "Dont Know",
+                                          "TOT", "Other")))%>%
+   ggplot(aes(x = State, y = count))+
+  geom_bar(stat = "identity", position = "dodge",
+           width = 0.5, fill = "darkslategray3")+
+   geom_errorbar(aes(ymin = count - ci, ymax = count + ci),
+                 width=.05, position=position_dodge(.5)) +
+   theme_few()+
+   xlab("State") + ylab("Mean Number of Trials") +
+   ggtitle("State Data by Prime Condition")
```



7.1 State by Prime Type

```
> state_rmisc = Rmisc::summarySE(prime_subject_state,
+                       measurevar = "count",
+                       groupvars = c("PrimeCondition", "State"))
> library(ggplot2)
> library(ggthemes)
> prime_state$State = as.factor(as.numeric(prime_state$State))
> state_rmisc %>% mutate(State = factor(State, levels = unique(State),
+                           labels = c("Know", "Dont Know",
+                                      "TOT", "Other")))%>%
+   ggplot(aes(x = PrimeCondition, y = count, fill = State))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.5)+
+   geom_errorbar(aes(ymin = count - ci, ymax = count + ci),
+                 width=.05, position=position_dodge(.5)) +
+   theme_few()+
+   xlab("Prime Condition") + ylab("Mean Number of Trials") +
+   ggtitle("State Data by Prime Condition")
```



8 Target Demasking Analysis

```
> library(dplyr)
> primeflash_firsttrim = subset(TOT,
+                                   TOT$RTrecogniseTarget > 250 &
+                                  TOT$RTrecogniseTarget < 7000)
> ## aggregate per subject all IVs and DVs
> meanRT = group_by(primeflash_firsttrim, Subject) %>%
+   summarise_at(vars(ResponseRT, RTrecogniseTarget), mean)
> colnames(meanRT) = c("Subject", "MeanTargetRT",
+                      "MeanRTrecogTarget")
> sdRT = group_by(primeflash_firsttrim, Subject) %>%
+   summarise_at(vars(ResponseRT, RTrecogniseTarget), sd)
> colnames(sdRT) = c("Subject", "sdTargetRT",
+                      "sdRTrecogTarget")
> RT_agg = merge(meanRT, sdRT, by = "Subject")
> ## merge aggregate info with long data
> primeflash_z = merge(primeflash_firsttrim, RT_agg, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> primeflash_z = primeflash_z %>% mutate(zTargetRT =
+                                         (ResponseRT - MeanTargetRT)/sdTargetRT,
+                                            zTargetRecogRT =
+                                              (RTrecogniseTarget -
+                                                 MeanRTrecogTarget)/sdRTrecogTarget)
> ## checking: subject level means should be zero
>
> sub_pic = group_by(primeflash_z, Subject) %>%
+   summarise_at(vars(zTargetRT,zTargetRecogRT), mean)
>
```

9 Trimming z-RTs

```
> primeflash_z_trimmed_target = subset(primeflash_z,
+                                   primeflash_z$zTargetRecogRT < 3 &
+                                     primeflash_z$zTargetRecogRT > -3)
> primeflash_z_trimmed_targetdef = subset(primeflash_z,
+                                   primeflash_z$zTargetRT < 3 &
+                                     primeflash_z$zTargetRT > -3)
```

10 Repeating z-scoring

10.1 For Target

```
> ## aggregate per subject all IVs and DVs
> meanRT_target = group_by(primeflash_z_trimmed_target, Subject) %>%
+   summarise_at(vars(ResponseRT, RTrecogniseTarget), mean)
> colnames(meanRT_target) = c("Subject", "MeanTargetRT_trim",
+                      "MeanRTrecogTarget_trim")
> sdRT_target = group_by(primeflash_z_trimmed_target, Subject) %>%
+   summarise_at(vars(ResponseRT, RTrecogniseTarget), sd)
> colnames(sdRT_target) = c("Subject", "sdTargetRT_trim",
+                       "sdRTrecogTarget_trim")
> RT_agg_target = merge(meanRT_target, sdRT_target, by = "Subject")
> ## merge aggregate info with long data
> primeflash_final_z_target = merge(primeflash_z_trimmed_target,
+                              RT_agg_target, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> primeflash_final_z_target = primeflash_final_z_target %>%
+                                   mutate(zTargetRT_trim =
+                                              (ResponseRT -
+                                                 MeanTargetRT_trim)/sdTargetRT_trim,
+                                            zTargetRecogRT_trim =
+                                              (RTrecogniseTarget -
+                                       MeanRTrecogTarget_trim)/sdRTrecogTarget_trim)
> ## checking: subject level means should be zero
>
> sub_pic = group_by(primeflash_final_z_target, Subject) %>%
+   summarise_at(vars(zTargetRT_trim, zTargetRecogRT_trim), mean)
> primeflash_final_z = primeflash_final_z_target
```

10.2 For TargetDef

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

10.3 Effect of Prime on Target RT

```
> library(lme4)
> contrasts(primeflash_final_z$PrimeCondition) = contr.treatment(4, base = 4)
> RTprime_RT_model_1 = lmer(data = primeflash_final_z,
+                       zTargetRecogRT_trim ~ PrimeCondition +
+                               (1|Subject) + (1|Stimuli1))
> summary(RTprime_RT_model_1)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: zTargetRecogRT_trim ~ PrimeCondition + (1 | Subject) + (1 | Stimuli1)
   Data: primeflash_final_z

REML criterion at convergence: 6597

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.8372 -0.6961 -0.0897  0.6300  3.4418

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli1 (Intercept) 0.2818   0.5308
 Subject  (Intercept) 0.0000   0.0000
 Residual             0.7217   0.8496
Number of obs: 2544, groups:  Stimuli1, 72; Subject, 36

Fixed effects:
                  Estimate Std. Error t value
(Intercept)      0.0364687  0.0710632   0.513
PrimeCondition1 -0.0009348  0.0477148  -0.020
PrimeCondition2 -0.0151003  0.0475608  -0.317
PrimeCondition3 -0.0663621  0.0476890  -1.392

Correlation of Fixed Effects:
            (Intr) PrmCn1 PrmCn2
PrimeCndtn1 -0.335
PrimeCndtn2 -0.336  0.500
PrimeCndtn3 -0.335  0.499  0.501
```

```
> # confint(RTprime_RT_model_1) refernce is unrelated
> #
> # > confint(RTprime_RT_model_1)
> # Computing profile confidence intervals ...
> #                       2.5 %      97.5 %
> # .sig01            0.44477451 0.63332620
> # .sig02            0.00000000 0.03520198
> # .sigma            0.82591299 0.87327502
> # (Intercept)      -0.10324751 0.17627404
> # PrimeCondition1 -0.09444271 0.09255556
> # PrimeCondition2 -0.10828887 0.07810593
> # PrimeCondition3 -0.15980837 0.02708857
>
```

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

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

Response: zTargetRecogRT_trim
                Chisq Df Pr(>Chisq)
PrimeCondition 2.5781  3     0.4613
```

```
> ## ANOVA
>
> #subject
> targetRT_sub = group_by(primeflash_final_z, Subject, PrimeCondition) %>%
+   summarise_at(vars(zTargetRecogRT_trim), mean)
> targetRT_sub$Subject = as.factor(targetRT_sub$Subject)
> targetRT_aov = aov(data = targetRT_sub, zTargetRecogRT_trim ~ PrimeCondition +
+                      Error(Subject/PrimeCondition))
> summary(targetRT_aov)
```

```
Error: Subject
          Df   Sum Sq   Mean Sq F value Pr(>F)
Residuals 35 0.000803 2.294e-05

Error: Subject:PrimeCondition
                Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition   3  0.094 0.03148    0.66  0.579
Residuals      105  5.010 0.04771
```

```
> #item
> targetRT_item = group_by(primeflash_final_z, Stimuli1, PrimeCondition) %>%
+   summarise_at(vars(zTargetRecogRT_trim), mean)
> targetRT_item$Stimuli1 = as.factor(targetRT_item$Stimuli1)
> targetRTitem_aov = aov(data = targetRT_item, zTargetRecogRT_trim ~ PrimeCondition +
+                      Error(Stimuli1/PrimeCondition))
> summary(targetRTitem_aov)
```

```
Error: Stimuli1
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 71  84.68   1.193

Error: Stimuli1:PrimeCondition
                Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition   3  0.194 0.06466   0.773   0.51
Residuals      213 17.824 0.08368
```

```
> targetRT_sub_wide = tidyr::spread(targetRT_sub, PrimeCondition, zTargetRecogRT_trim)
> t.test(targetRT_sub_wide$R, targetRT_sub_wide$B, paired = TRUE)
```

```
        Paired t-test

data:  targetRT_sub_wide$R and targetRT_sub_wide$B
t = -1.2493, df = 35, p-value = 0.2199
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.14882152  0.03543424
sample estimates:
mean of the differences
            -0.05669364
```

```
>
```

10.4 Incorrect Target

```
> # conditional on targetacc
>
> primeflash_final_z_incorrect = primeflash_final_z %>% filter(TargetAccuracy == "0")
> RTprime_RT_model_2 = lmer(data = primeflash_final_z_incorrect,
+                     zTargetRecogRT_trim ~ PrimeCondition +
+                             (1|Subject) + (1|Stimuli1))
> summary(RTprime_RT_model_2)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: zTargetRecogRT_trim ~ PrimeCondition + (1 | Subject) + (1 | Stimuli1)
   Data: primeflash_final_z_incorrect

REML criterion at convergence: 4543.2

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.4054 -0.6474 -0.0698  0.6512  3.2583

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli1 (Intercept) 0.26825  0.51793
 Subject  (Intercept) 0.00671  0.08191
 Residual             0.69008  0.83071
Number of obs: 1764, groups:  Stimuli1, 72; Subject, 36

Fixed effects:
                Estimate Std. Error t value
(Intercept)      0.20577    0.07399   2.781
PrimeCondition1 -0.01241    0.05539  -0.224
PrimeCondition2  0.03717    0.05690   0.653
PrimeCondition3 -0.05402    0.05549  -0.973
```

```
Correlation of Fixed Effects:
            (Intr) PrmCn1 PrmCn2
PrimeCndtn1 -0.371
PrimeCndtn2 -0.358  0.480
PrimeCndtn3 -0.367  0.491  0.480
```

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

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

Response: zTargetRecogRT_trim
                 Chisq Df Pr(>Chisq)
PrimeCondition 2.5924  3     0.4588
```

```
> primeflash_final_z_correct = primeflash_final_z %>% filter(TargetAccuracy == "1")
> RTprime_RT_model_3 = lmer(data = primeflash_final_z_correct,
+                      zTargetRecogRT_trim ~ PrimeCondition +
+                              (1|Subject) + (1|Stimuli1))
> summary(RTprime_RT_model_3)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: zTargetRecogRT_trim ~ PrimeCondition + (1 | Subject) + (1 | Stimuli1)
   Data: primeflash_final_z_correct

REML criterion at convergence: 1825.4

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.6518 -0.6454 -0.1651  0.4735  4.4125

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli1 (Intercept) 0.152618 0.39066
 Subject  (Intercept) 0.005589 0.07476
 Residual             0.529026 0.72734
Number of obs: 780, groups:  Stimuli1, 71; Subject, 36

Fixed effects:
                Estimate Std. Error t value
(Intercept)     -0.47550    0.07667  -6.202
PrimeCondition1  0.01266    0.07857   0.161
PrimeCondition2  0.03082    0.07474   0.412
PrimeCondition3 -0.05000    0.07793  -0.642

Correlation of Fixed Effects:
            (Intr) PrmCn1 PrmCn2
PrimeCndtn1 -0.532
```

```
PrimeCndtn2 -0.566  0.544
PrimeCndtn3 -0.524  0.512  0.539
```

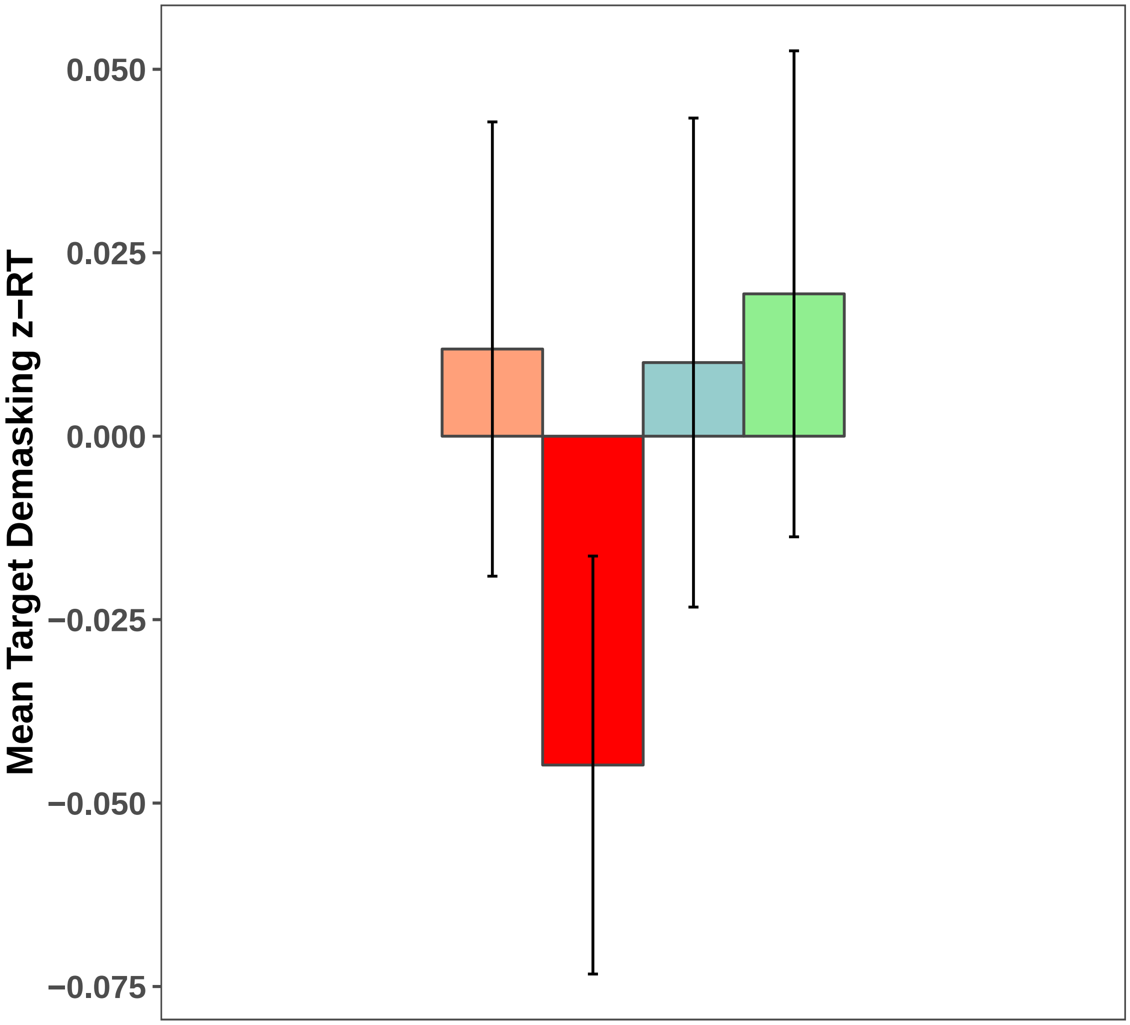
```
> car::Anova(RTprime_RT_model_3)
```

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

Response: zTargetRecogRT_trim
                Chisq Df Pr(>Chisq)
PrimeCondition 1.2873  3     0.7322
```

```
>
>
```

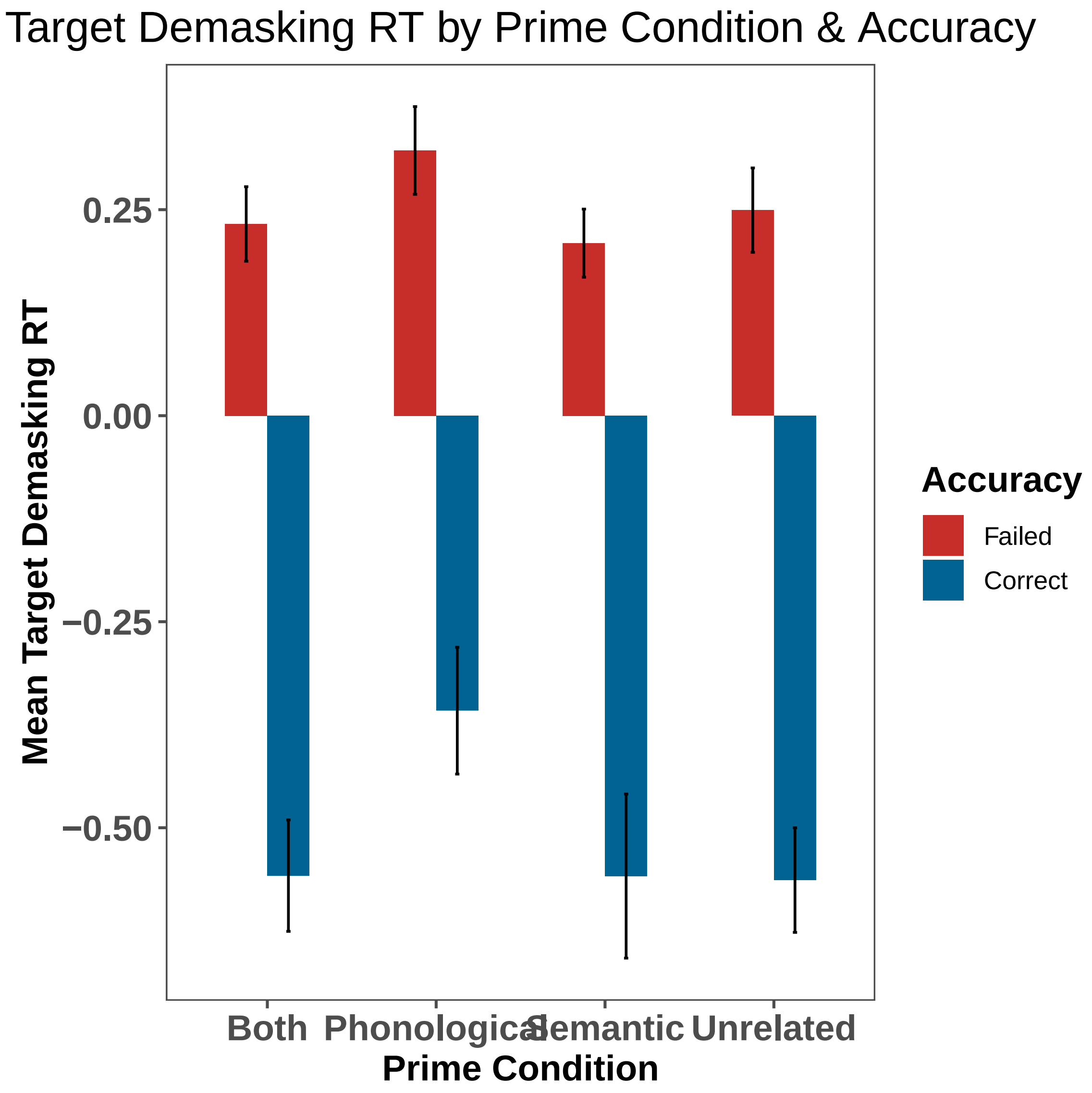
10.5 Target RT Model

```
> targetRT_sub$primefac = ordered(as.factor(as.character(targetRT_sub$PrimeCondition)),
> targetRT_rmisc = Rmisc::summarySE(targetRT_sub,
+                       measurevar = "zTargetRecogRT_trim",
+                       groupvars = c("primefac"))
> targetRT_rmisc$Experiment = 1
> targetRT_rmisc$Experiment = as.factor(targetRT_rmisc$Experiment)
> library(ggplot2)
> library(ggthemes)
> targetRT_rmisc %>% mutate(PrimeType = factor(primefac,
+                                            levels = unique(primefac),
+                     labels = c("Both", "Semantic",
+                                "Phonological", "Unrelated"))) %>%
+ ggplot(aes(x = Experiment,
+            y = zTargetRecogRT_trim, fill = PrimeType))+
+  geom_bar(stat = "identity", position = "dodge",
+          width = 0.5,
+                     color ="gray28")+
+   geom_errorbar(aes(ymin = zTargetRecogRT_trim - se,
+                     ymax = zTargetRecogRT_trim + se),
+                width=.05, position=position_dodge(.5)) +
+   guides(fill = FALSE)+
+   theme_few()+
+  scale_fill_manual(values = c( "lightsalmon", "red",
+                               "paleturquoise3","lightgreen"))+
+   xlab("") + ylab("Mean Target Demasking z-RT") +
+   ggtitle("") +
+     theme(axis.text.y = element_text(face = "bold", size = rel(1.2)),
+          axis.text.x = element_blank(),
+          axis.ticks.x = element_blank(),
+          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))
```



10.6 Target RT-Accuracy Model

```
> targetRT_acc_sub = group_by(primeflash_final_z, Subject,
+                             PrimeCondition, TargetAccuracy) %>%
+   summarise_at(vars(zTargetRecogRT_trim), mean)
> targetRT_rmisc_acc = Rmisc::summarySE(targetRT_acc_sub,
+                       measurevar = "zTargetRecogRT_trim",
+                       groupvars = c("PrimeCondition", "TargetAccuracy"))
> targetRT_rmisc_acc$TargetAccuracy = as.factor(targetRT_rmisc_acc$TargetAccuracy)
> library(ggplot2)
> library(ggthemes)
> targetRT_rmisc_acc %>% mutate(`Prime Condition` = factor(PrimeCondition,
+                                                   levels = unique(PrimeCondition),
+                      labels = c("Both", "Phonological",
+                                 "Semantic", "Unrelated")),
+                      Accuracy = factor(TargetAccuracy,
+                             levels = unique(TargetAccuracy),
+                      labels = c("Failed" , "Correct"))) %>%
+ ggplot(aes(x = `Prime Condition`,
+            y = zTargetRecogRT_trim,
+            fill = Accuracy, group = Accuracy))+
+  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()+
+   scale_fill_wsj()+
+   xlab("Prime Condition") + ylab("Mean Target Demasking RT") +
+   ggtitle("Target Demasking RT by Prime Condition & Accuracy") +
+     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))
```



10.7 Effect of Prime on TargetDef RT

```
> library(lme4)
> contrasts(primeflash_final_z_targetdef$PrimeCondition) = contr.treatment(n = 4)
> RTprime_targetRT_model_1 = lmer(data = primeflash_final_z_targetdef,
+                        zTargetRT_trim ~ PrimeCondition +
+                                (1|Subject) + (1|Stimuli1))
> summary(RTprime_targetRT_model_1)
```

```
Linear mixed model fit by REML ['lmerMod']
```

```
Formula: zTargetRT_trim ~ PrimeCondition + (1 | Subject) + (1 | Stimuli1)
   Data: primeflash_final_z_targetdef

REML criterion at convergence: 7146.1

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.0983 -0.6736  0.0480  0.7663  2.3624

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli1 (Intercept) 0.06365  0.2523
 Subject  (Intercept) 0.00000  0.0000
 Residual             0.92412  0.9613
Number of obs: 2553, groups:  Stimuli1, 72; Subject, 36

Fixed effects:
                 Estimate Std. Error t value
(Intercept)      0.022470   0.048370   0.465
PrimeCondition2 -0.028186   0.053782  -0.524
PrimeCondition3 -0.060130   0.053923  -1.115
PrimeCondition4  0.006746   0.053878   0.125

Correlation of Fixed Effects:
            (Intr) PrmCn2 PrmCn3
PrimeCndtn2 -0.559
PrimeCndtn3 -0.558  0.502
PrimeCndtn4 -0.558  0.502  0.501
```

```
> car::Anova(RTprime_targetRT_model_1)
```

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

Response: zTargetRT_trim
                Chisq Df Pr(>Chisq)
PrimeCondition 1.9228  3     0.5886
```

```
> RTprime_targetRT_model_2 = lmer(data = primeflash_final_z_targetdef,
+                      zTargetRT_trim ~ TargetAccuracy*PrimeCondition +
+                              (1|Subject) + (1|Stimuli1))
> summary(RTprime_targetRT_model_2)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: zTargetRT_trim ~ TargetAccuracy * PrimeCondition + (1 | Subject) +
    (1 | Stimuli1)
   Data: primeflash_final_z_targetdef

REML criterion at convergence: 7118.2
```

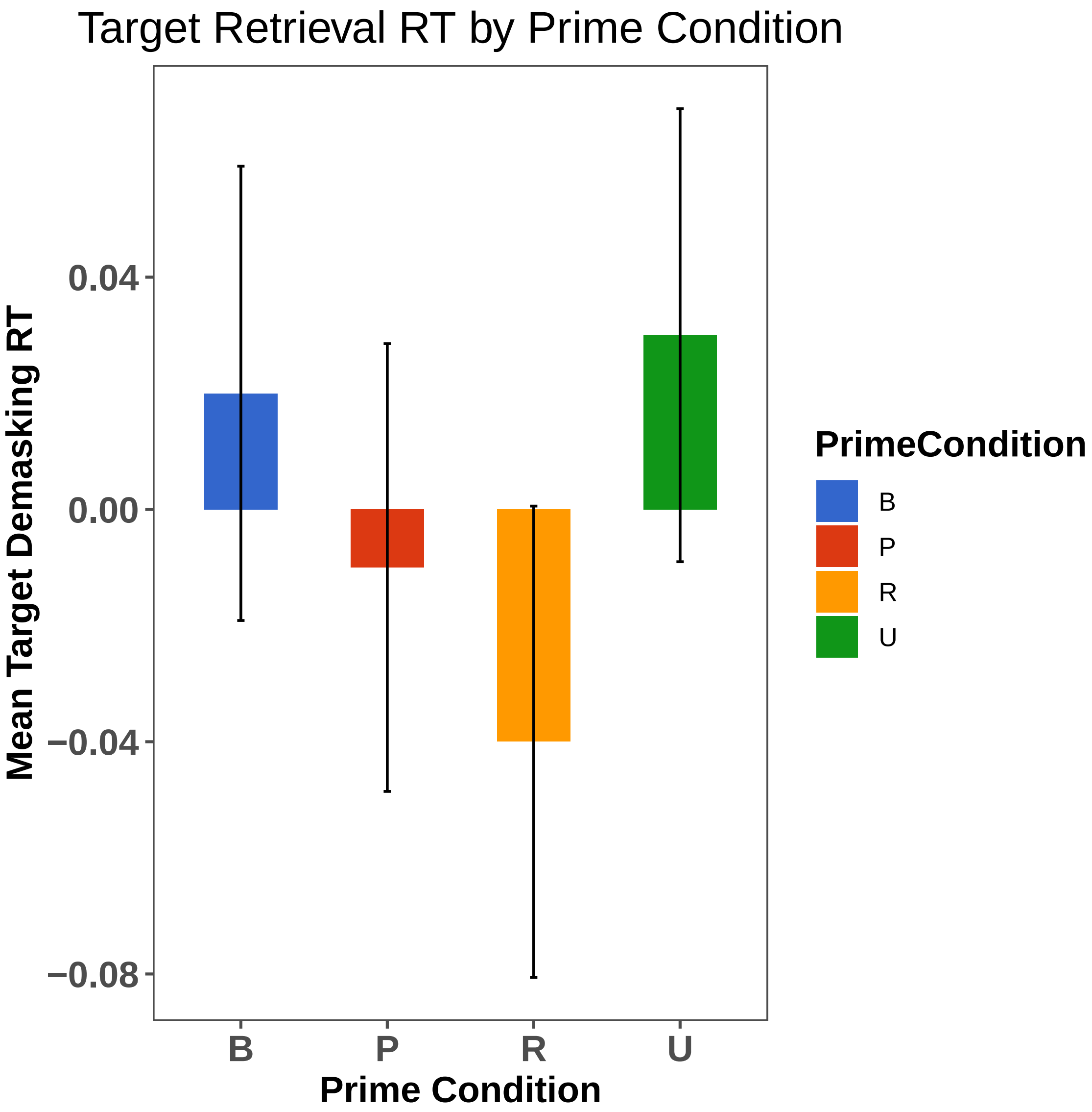
```
Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.92279 -0.69745  0.01799  0.76100  2.44436

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli1 (Intercept) 0.06328  0.2516
 Subject  (Intercept) 0.00000  0.0000
 Residual             0.91087  0.9544
Number of obs: 2553, groups:  Stimuli1, 72; Subject, 36

Fixed effects:
                               Estimate Std. Error t value
(Intercept)                    -0.08006    0.05409  -1.480
TargetAccuracy                  0.35265    0.08505   4.147
PrimeCondition2                -0.01507    0.06531  -0.231
PrimeCondition3                -0.03148    0.06383  -0.493
PrimeCondition4                 0.03414    0.06331   0.539
TargetAccuracy:PrimeCondition2 -0.10493    0.11517  -0.911
TargetAccuracy:PrimeCondition3 -0.10160    0.11846  -0.858
TargetAccuracy:PrimeCondition4 -0.07954    0.11958  -0.665

Correlation of Fixed Effects:
            (Intr) TrgtAc PrmCn2 PrmCn3 PrmCn4 TA:PC2 TA:PC3
TargtAccrcy -0.457
PrimeCndtn2 -0.573  0.365
PrimeCndtn3 -0.586  0.373  0.487
PrimeCndtn4 -0.591  0.378  0.491  0.502
TrgtAcc:PC2  0.327 -0.716 -0.572 -0.278 -0.280
TrgtAcc:PC3  0.316 -0.690 -0.265 -0.544 -0.274  0.515
TrgtAcc:PC4  0.314 -0.686 -0.262 -0.269 -0.535  0.510  0.498
```

```
> car::Anova(RTprime_targetRT_model_2)
```

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

Response: zTargetRT_trim
                                Chisq Df Pr(>Chisq)
TargetAccuracy                39.5885  1  3.135e-10 ***
PrimeCondition                 2.6280  3     0.4526
TargetAccuracy:PrimeCondition  1.0443  3     0.7905
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

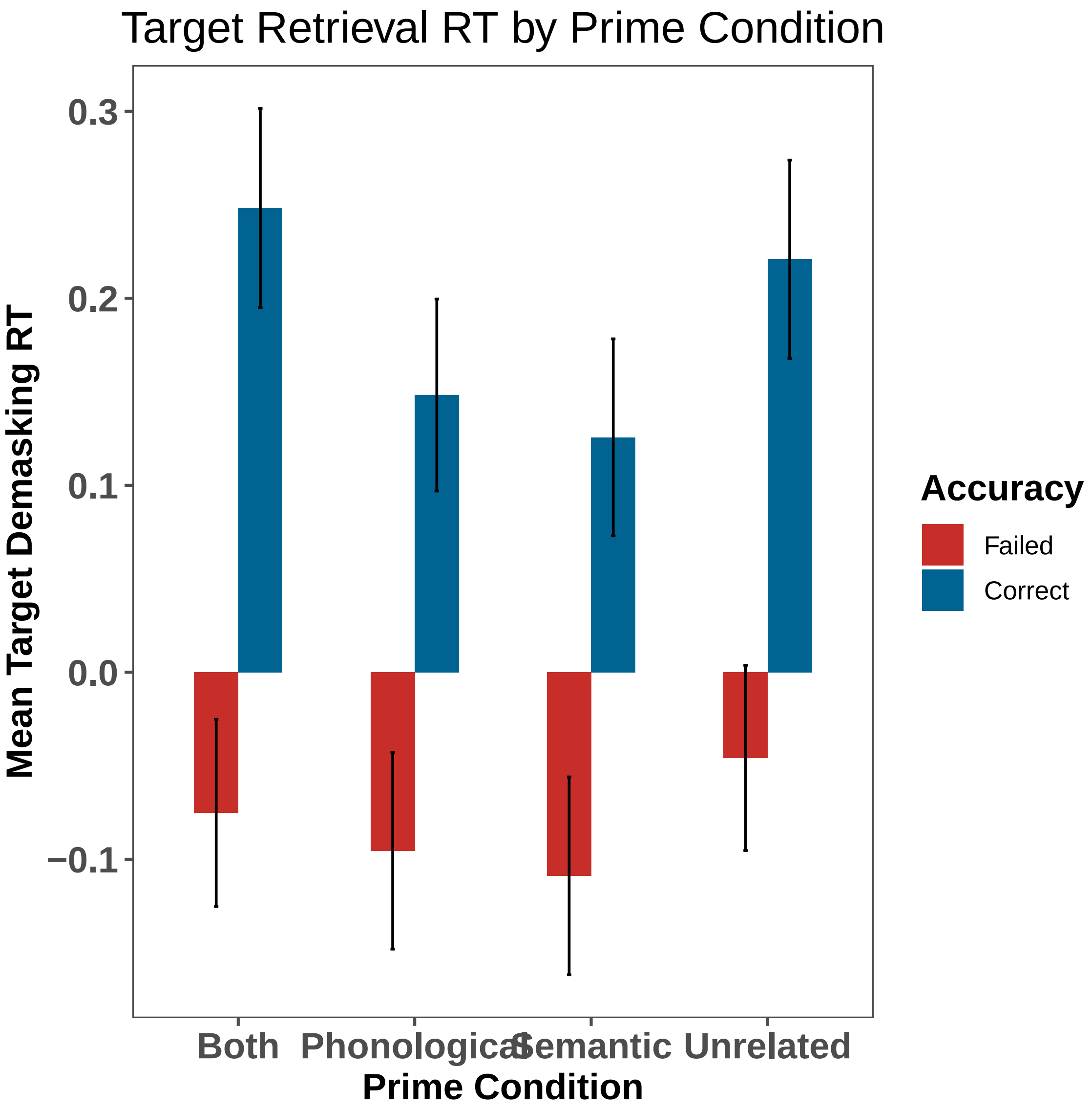
10.8 TargetDef RT Model 1

```
> primeflash_final_z_targetdef$TargetAccuracy = as.factor(primeflash_final_z_targetdef$T
> targetRT_rmisc1 = Rmisc::summarySE(primeflash_final_z_targetdef,
+                       measurevar = "zTargetRT_trim",
+                       groupvars = c("PrimeCondition"))
> targetRT_rmisc1$zTargetRT_trim = round(targetRT_rmisc1$zTargetRT_trim,2)
> library(ggplot2)
> library(ggthemes)
> targetRT_rmisc1 %>%
+ ggplot(aes(x = PrimeCondition,
+            y = zTargetRT_trim, fill = PrimeCondition))+
+  geom_bar(stat = "identity", position = "dodge",
+           width = 0.5)+
+   geom_errorbar(aes(ymin = zTargetRT_trim - se, ymax = zTargetRT_trim + se),
+                 width=.05, position=position_dodge(.5)) +
+   theme_few()+
+   scale_fill_gdocs()+
+   xlab("Prime Condition") + ylab("Mean Target Demasking RT") +
+   ggtitle("Target Retrieval 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))
```



10.9 TargetDef RT Model 2

```
> primeflash_final_z_targetdef$TargetAccuracy = as.factor(primeflash_final_z_targetdef$T
> targetRT_rmisc = Rmisc::summarySE(primeflash_final_z_targetdef,
+                       measurevar = "zTargetRT_trim",
+                       groupvars = c("PrimeCondition", "TargetAccuracy"))
> library(ggplot2)
> library(ggthemes)
> targetRT_rmisc %>% mutate(`Prime Condition` = factor(PrimeCondition,
+                                                      levels = unique(PrimeCondition),
+                     labels = c("Both", "Phonological",
+                                "Semantic", "Unrelated")),
+                     Accuracy = factor(TargetAccuracy,
+                            levels = unique(TargetAccuracy),
+                     labels = c("Failed" , "Correct"))) %>%
+ ggplot(aes(x = `Prime Condition`,
+            y = zTargetRT_trim, fill = Accuracy))+
+  geom_bar(stat = "identity", position = "dodge",
+           width = 0.5)+
+   geom_errorbar(aes(ymin = zTargetRT_trim - se, ymax = zTargetRT_trim + se),
+                 width=.05, position=position_dodge(.5)) +
+   theme_few()+
+   scale_fill_wsj()+
+   xlab("Prime Condition") + ylab("Mean Target Demasking RT") +
+   ggtitle("Target Retrieval 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))
```



11 MTurk Covariate Analyses

```
> itemratings= read.csv("Abhilasha_item_wide.csv",
+                                   header = TRUE, sep = ",")
> main = TOT
> main = main %>% filter(PrimeCondition %in% c("P", "B"))
> main_item = merge(main, itemratings,
+                       by = c("Stimuli1", "PrimeCondition"))
> main_item = dplyr::arrange(main_item, Subject, Stimuli1, PrimeType)
> ## Impacting Ret/NotRet
```

```
>
> m_primeflash = lme4::glmer(data = main_item, TargetAccuracy ~
+                     PrimeCondition +
+                               SoundRating +
+                             (1|Subject) + (1|Stimuli1),
+                            family = "binomial",
+                            control=glmerControl(optimizer="bobyqa",
+             optCtrl=list(maxfun=100000)))
> summary(m_primeflash)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: TargetAccuracy ~ PrimeCondition + SoundRating + (1 | Subject) +
    (1 | Stimuli1)
   Data: main_item
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))

     AIC      BIC   logLik deviance df.resid
  1428.8   1454.6   -709.4   1418.8     1291

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.3099 -0.5887 -0.3257  0.6803  3.9579

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli1 (Intercept) 1.3328   1.154
 Subject  (Intercept) 0.8612   0.928
Number of obs: 1296, groups:  Stimuli1, 72; Subject, 36

Fixed effects:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     -1.32951    0.48029  -2.768  0.00564 **
PrimeCondition1 -0.19013    0.08913  -2.133  0.03291 *
SoundRating      0.06955    0.11049   0.630  0.52900
---
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
PrimeCndtn1 -0.548
SoundRating -0.888  0.628
```

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

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

```
Response: TargetAccuracy
                Chisq Df Pr(>Chisq)
PrimeCondition 4.5506  1    0.03291 *
SoundRating    0.3963  1    0.52900
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> anova(m_primeflash)
```

```
Analysis of Variance Table
               Df  Sum Sq Mean Sq F value
PrimeCondition  1 10.9056 10.9056 10.9056
SoundRating     1  0.4018  0.4018  0.4018
```