Repeated Lexical Retrieval: Experiment 5

Abhilasha Kumar

June 6, 2019

1 Reading the Data File

We first read the file into an object called SemanticCuedRecall. We can also display some part of the data by calling the head() function.

```
> SemanticCuedRecall = read.csv("E5_SemanticCuedRecall_FINAL.csv",
+                               header = TRUE, sep = ",")
> head(SemanticCuedRecall[,c(1,21,22)])
```

```
  Subject CuedRecallAcc TargetAccuracy
1       1             1              0
2       1             1              0
3       1             1              0
4       1             1              0
5       1             1              0
6       1             0              1
```

1.1 LME

```
> library(lme4)
> contrasts(SemanticCuedRecall$PrimeCondition)= contr.treatment(2, base = 2)
> prime_lmer2 = glmer(data = SemanticCuedRecall,
+                    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: SemanticCuedRecall
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))
```

```
     AIC      BIC   logLik deviance df.resid
  1623.8   1644.9   -807.9   1615.8     1436

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.3851 -0.6190 -0.3241  0.6896  4.6869

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli1 (Intercept) 1.4035   1.1847
 Subject  (Intercept) 0.9478   0.9736
Number of obs: 1440, groups:  Stimuli1, 48; Subject, 30

Fixed effects:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      -0.5611     0.2641  -2.125   0.0336 *
PrimeCondition1  -0.1626     0.1286  -1.264   0.2061
---
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)
PrimeCndtn1 -0.247
```

```
> #confint(prime_lmer2)
> # > confint(prime_lmer2)
> # Computing profile confidence intervals ...
> #                        2.5 %      97.5 %
> # .sig01             0.9316669  1.52830645
> # .sig02             0.7217247  1.34753572
> # (Intercept)      -1.0933527 -0.03889619
> # PrimeCondition1 -0.4187832  0.09229101
```

1.2 Percentage State Prime Analysis

```
> state = read.csv("SemanticCuedRecall_AGG.csv",header = TRUE, sep = ",")
> j_statepercent = state[,c(1,21:28)] # 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_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 29 2.338e-18 8.06e-20

Error: Subject:PrimeCondition
               Df    Sum Sq   Mean Sq F value Pr(>F)
PrimeCondition  1 4.200e-21 4.170e-21   0.065  0.801
Residuals      29 1.871e-18 6.451e-20

Error: Subject:State
          Df Sum Sq Mean Sq F value   Pr(>F)
State      3  4.932  1.6441   34.73 7.44e-15 ***
Residuals 87  4.118  0.0473
---
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  3 0.0028 0.000945   0.061   0.98
Residuals            87 1.3531 0.015552
```

1.2.1 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( "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( "red","lightgreen"))+
+   ggtitle("E6")  +
+    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
```

1.2.2 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 29  1.803 0.06219

Error: Subject:PrimeCondition
               Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition  1 0.0003 0.00026   0.012  0.914
Residuals      29 0.6360 0.02193
```

1.2.3 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 29  1.698 0.05855

Error: Subject:PrimeCondition
               Df Sum Sq  Mean Sq F value  Pr(>F)
PrimeCondition  1 0.0010 0.001042   0.059   0.81
Residuals      29 0.5111 0.017625
```

1.2.4 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 29  0.227 0.007826

Error: Subject:PrimeCondition
               Df  Sum Sq   Mean Sq F value Pr(>F)
PrimeCondition  1 0.00012 0.0001157   0.041  0.841
Residuals      29 0.08148 0.0028097
```

1.2.5 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 29 0.3897 0.01344

Error: Subject:PrimeCondition
               Df  Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition  1 0.00142 0.001418    0.33   0.57
Residuals      29 0.12445 0.004291
```

2 Raw Retrieval States

```
> library(dplyr)
> SemanticCuedRecall_Count = group_by(SemanticCuedRecall,
+                                     Subject, PrimeCondition,
+                                     TargetQuestion.RESP.Trial.) %>%
+   summarise(Count = n())
> state_rmisc = Rmisc::summarySE(SemanticCuedRecall_Count,
+                                measurevar = "Count",
+                                groupvars = c("PrimeCondition",
+                                              "TargetQuestion.RESP.Trial."))
> x <- c("1","2", "3", "4")
> state_rmisc = state_rmisc %>%
+   mutate(rstate =  factor(TargetQuestion.RESP.Trial., levels = x)) %>%
+   arrange(rstate)
> library(ggplot2)
> library(ggthemes)
> percentplot = state_rmisc %>%
+   mutate(PrimeType = factor(PrimeCondition, levels = unique(PrimeCondition),
+                    labels = c("Semantic", "Unrelated")),
+    R = factor(rstate, levels = unique(rstate),
+                                 labels = c( "1: Know","2: Dont Know",
+                                             "3:Other", "4: TOT")))%>%
+ ggplot(aes(x = R, y = Count,
+            group = PrimeType, fill = PrimeType))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.7,
+           color= "black")+
+   geom_errorbar(aes(ymin=Count - se, ymax=Count + se),
+              width=.2, color = "gray26",
+              position = position_dodge(0.7))+
+  theme_few()+
+     xlab("") + ylab("Number of trials") +
+  scale_fill_manual(values = c( "red",
+                                "lightgreen"))+
+   ggtitle("E6")  +
+    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 Conditional Target Accuracy

In this section, we calculate the number of trials in which participants correctly or incorrectly recalled the item, and split that by whether they correctly recalled the target from the definition. Then, we calculate the proportion of trials from the raw number of trials.

```
> library(dplyr)
> cued_acc = group_by(SemanticCuedRecall) %>%
+   summarise_at(vars(CuedRecallAcc, TargetAccuracy), mean)
> cued_acc = group_by(SemanticCuedRecall, Subject,
+                     PrimeCondition, CuedRecallAcc) %>%
+   summarise(recalltrials = n())
> conditional_acc = group_by(SemanticCuedRecall, Subject, PrimeCondition,
+                            CuedRecallAcc, TargetAccuracy) %>%
+   summarise(trials = n())
> merge_acc = merge(conditional_acc, cued_acc,
+                   by = c("Subject", "PrimeCondition", "CuedRecallAcc"))
> merge_acc$prop = merge_acc$trials/merge_acc$recalltrials
```

4 ANOVA

In this section, we perform a repeated measures ANOVA on our data, to see if we are indeed seeing a difference in the proportion of unsuccessful trials for failed and successful cued recall.

```
> merge_acc$Subject =
+   as.factor(as.character(merge_acc$Subject))
> merge_acc$CuedRecallAcc =
+   as.factor(as.character(merge_acc$CuedRecallAcc))
> merge_acc$TargetAccuracy =
+   as.factor(as.character(merge_acc$TargetAccuracy))
> merge_acc = merge_acc[order(merge_acc$Subject, merge_acc$CuedRecallAcc),]
> library(lme4)
> cond_aov = lmer(data = merge_acc,
+         prop ~ PrimeCondition*CuedRecallAcc*TargetAccuracy +
+           (1|Subject))
> summary(cond_aov)
```

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

REML criterion at convergence: 1.7

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.4751 -0.6912 -0.0471  0.6519  2.8107

Random effects:
 Groups   Name        Variance Std.Dev.
 Subject  (Intercept) 0.00000  0.0000
 Residual             0.05214  0.2283
```

```
Number of obs: 223, groups:  Subject, 30

Fixed effects:
                                            Estimate Std. Error t value
(Intercept)                                  0.63296    0.04169  15.183
PrimeCondition1                              0.04330    0.05946   0.728
CuedRecallAcc1                              -0.03181    0.06057  -0.525
TargetAccuracy1                             -0.22513    0.06057  -3.717
PrimeCondition1:CuedRecallAcc1              -0.04971    0.08523  -0.583
PrimeCondition1:TargetAccuracy1             -0.09291    0.08523  -1.090
CuedRecallAcc1:TargetAccuracy1               0.13474    0.08767   1.537
PrimeCondition1:CuedRecallAcc1:TargetAccuracy1  0.06086    0.12251   0.497

Correlation of Fixed Effects:
            (Intr) PrmCn1 CdRcA1 TrgtA1 PrC1:CRA1 PC1:TA CRA1:T
PrimeCndtn1 -0.701
CudRcllAcc1 -0.688  0.483
TrgtAccrcy1 -0.688  0.483  0.474
PrmCn1:CRA1  0.489 -0.698 -0.711 -0.337
PrmCnd1:TA1  0.489 -0.698 -0.337 -0.711  0.487
CdRclA1:TA1  0.476 -0.333 -0.691 -0.691  0.491      0.491
PC1:CRA1:TA -0.340  0.485  0.494  0.494 -0.696     -0.696 -0.716
```

```
> car::Anova(cond_aov)
```

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

Response: prop
                                          Chisq Df Pr(>Chisq)
PrimeCondition                           0.1427  1   0.705631
CuedRecallAcc                            0.5799  1   0.446359
TargetAccuracy                          39.4166  1  3.424e-10 ***
PrimeCondition:CuedRecallAcc             0.1094  1   0.740827
PrimeCondition:TargetAccuracy            1.0741  1   0.300028
CuedRecallAcc:TargetAccuracy             7.3388  1   0.006748 **
PrimeCondition:CuedRecallAcc:TargetAccuracy  0.2468  1   0.619356
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
>
```

The ANOVA output tells us that the interaction term is not signiificant. We will next see this in a figure, to better understand our data.

5 Conditional Figure

```
> cond_figure = Rmisc::summarySE(merge_acc,
+                         measurevar = "prop",
+                         groupvars = c("PrimeCondition", "CuedRecallAcc",
+                                       "TargetAccuracy"))
> library(ggplot2)
> library(ggthemes)
> condfigure_plot = cond_figure %>% mutate(Recall = factor(CuedRecallAcc,
+                       levels = unique(CuedRecallAcc),
+                     labels = c("Failed Recall",
+                                "Successful Recall")),
+                     `Target Retrieval` = factor(TargetAccuracy,
+                           levels = unique(TargetAccuracy),
+                        labels = c("Failed Target Retrieval",
+                             "Successful Target Retrieval")))%>%
+ ggplot(aes(x = Recall, y = prop,
+            fill = `Target Retrieval`, group = `Target Retrieval`))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.7)+
+   geom_errorbar(aes(ymin=prop - ci, ymax=prop + ci),
+              width=.2, color = "gray26",
+              position = position_dodge(0.7))+
+   facet_wrap(~PrimeCondition)+
+  theme_few()+
+   scale_fill_wsj()+
+     xlab("Cued Recall Accuracy") + ylab("Mean Proportion of Trials") +
+   ggtitle("Target Retrieval Accuracy
+           as a function of Cued Recall Accuracy")  +
+    theme(axis.text = element_text(face = "bold", 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(face = "bold",
+                   size = rel(1.2), hjust = .5),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
> condfigure_plot
```

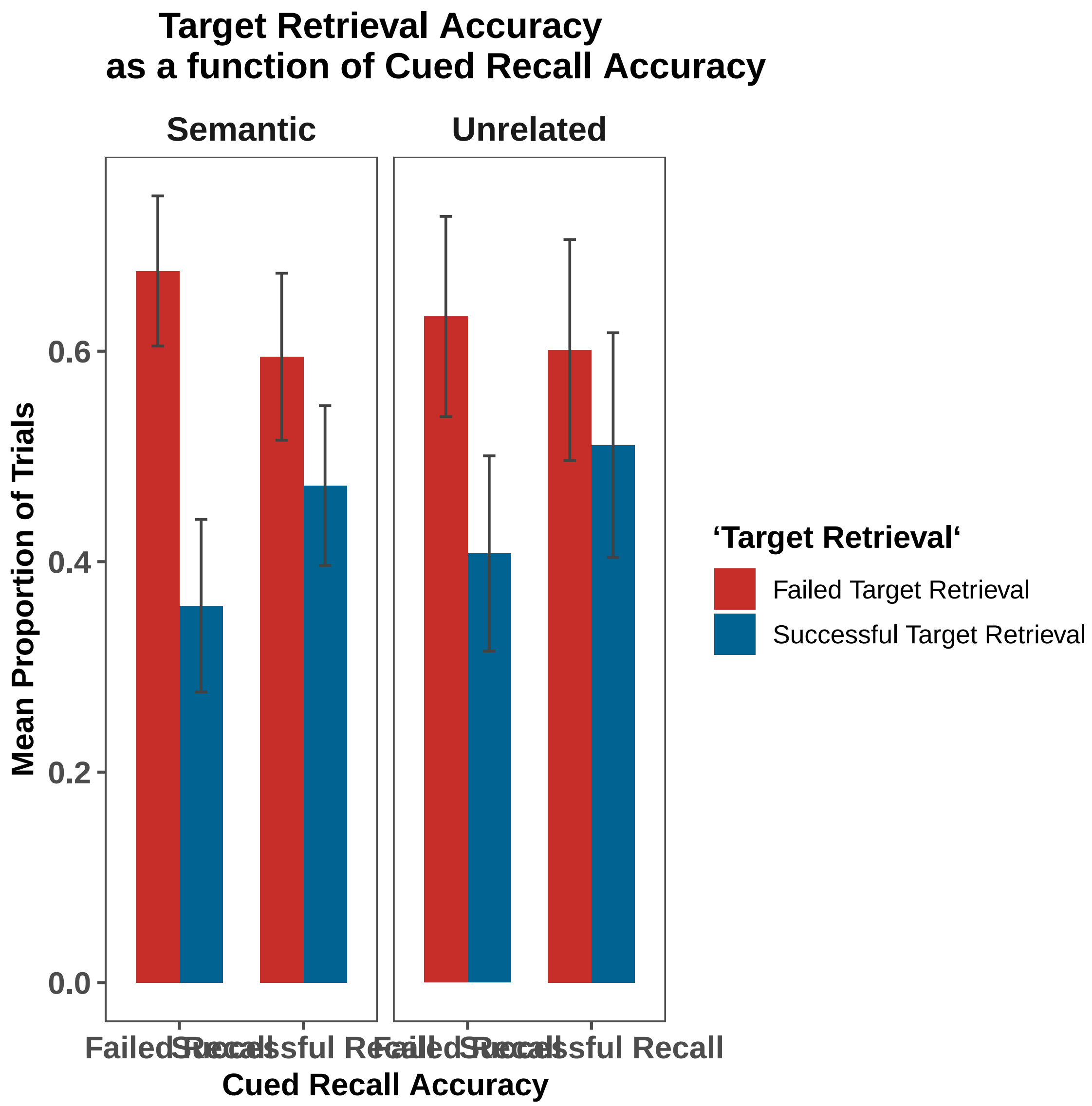
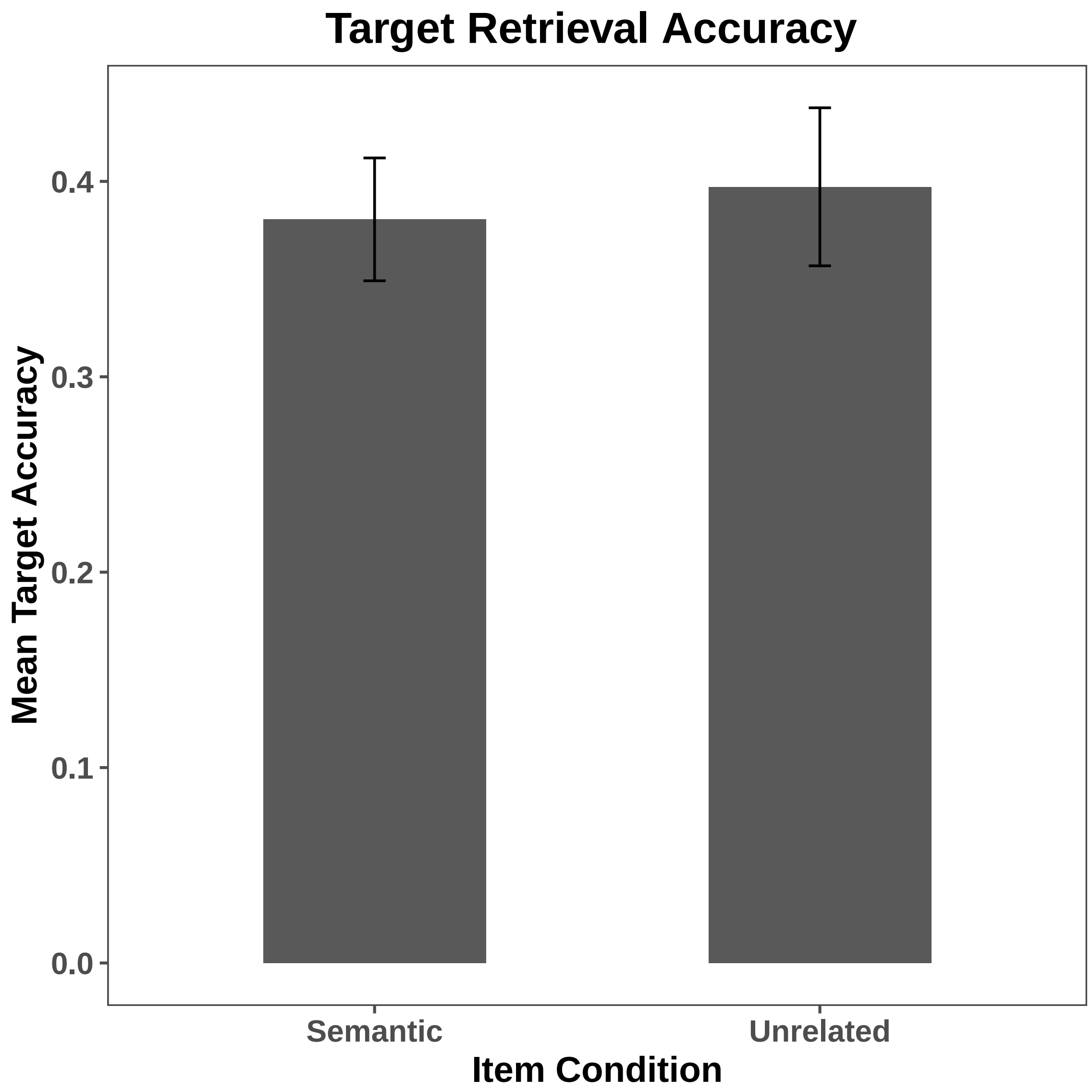


Figure Overall Target Accuracy

```
> prime_targetacc = group_by(SemanticCuedRecall, Subject, PrimeCondition) %>%
+   summarise_at(vars(TargetAccuracy), mean)
> target_rmisc_overall = Rmisc::summarySE(prime_targetacc,
+                           measurevar = "TargetAccuracy",
+                           groupvars = c("PrimeCondition"))
> library(ggplot2)
> library(ggthemes)
> target_rmisc_overall %>%
```

```
+ ggplot(aes(x = PrimeCondition , y = TargetAccuracy))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.5)+
+   geom_errorbar(aes(ymin = TargetAccuracy - se, ymax = TargetAccuracy + se),
+                 width=.05, position=position_dodge(.5)) +
+   theme_few()+
+   scale_fill_manual(values= c("slategray4", "slategray1"))+
+   xlab("Item Condition") + ylab("Mean Target Accuracy") +
+   ggtitle("Target Retrieval Accuracy ") +
+     theme(axis.text = element_text(face = "bold", size = rel(1)),
+           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(1.4), hjust = .5))
```



ANOVA

```
> prime_targetacc$Subject = as.factor(prime_targetacc$Subject)
> targetacc_aov = aov(data = prime_targetacc,
+                     TargetAccuracy ~ PrimeCondition +
+                       Error(Subject/PrimeCondition))
> summary(targetacc_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
```

```
Residuals 29   1.74 0.06001

Error: Subject:PrimeCondition
               Df Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition  1 0.0042 0.004167   0.223  0.641
Residuals      29 0.5427 0.018714
```

```
> ## ITEM
>
> prime_targetacc_item = group_by(SemanticCuedRecall, Stimuli1, PrimeCondition) %>%
+   summarise_at(vars(TargetAccuracy), mean)
> prime_targetacc_item$Stimuli1 = as.factor(prime_targetacc_item$Stimuli1)
> targetacc_aov_item = aov(data = prime_targetacc_item,
+                     TargetAccuracy ~ PrimeCondition +
+                       Error(Stimuli1/PrimeCondition))
> summary(targetacc_aov_item)
```

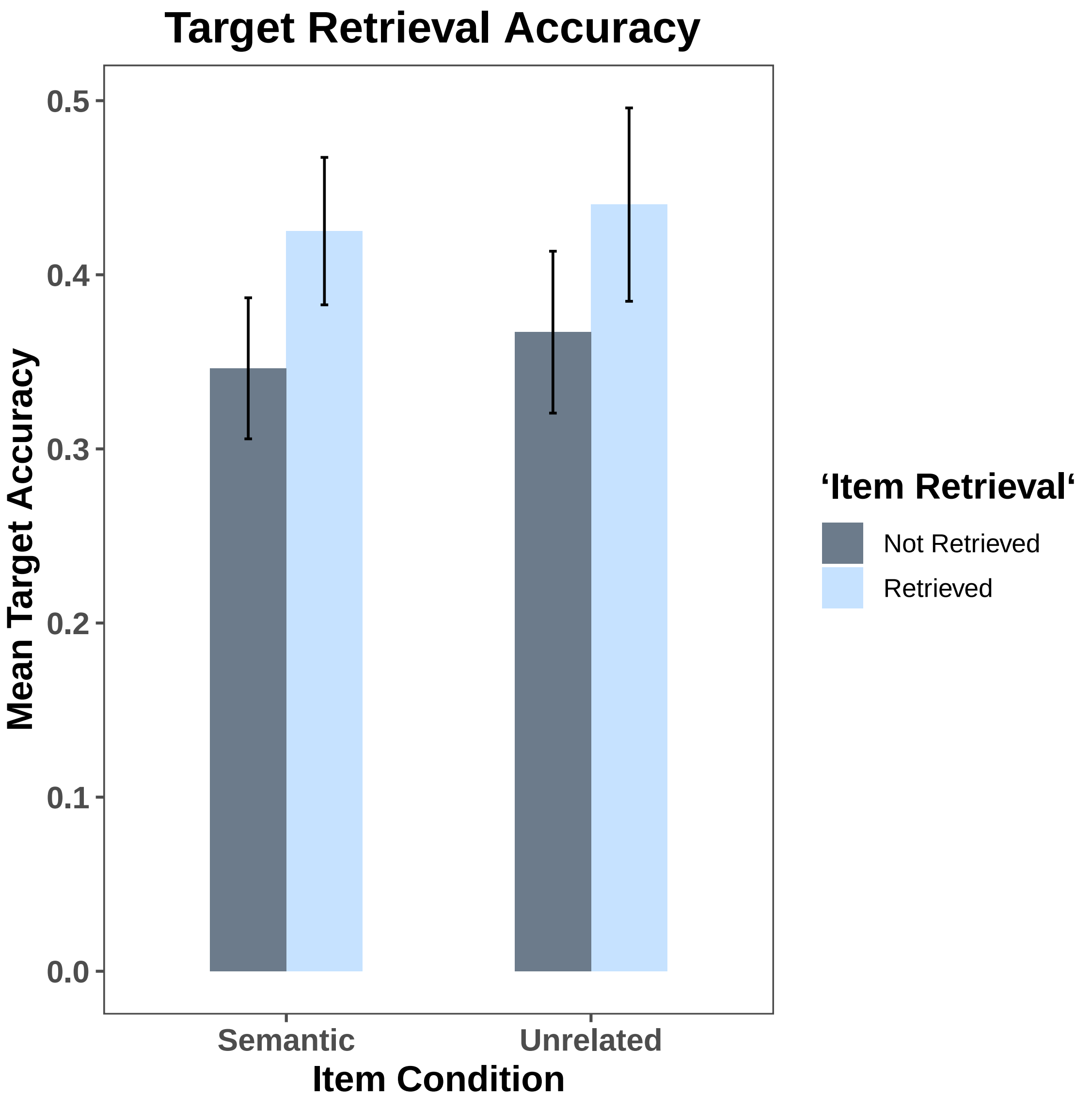
```
Error: Stimuli1
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 47  4.628 0.09847

Error: Stimuli1:PrimeCondition
               Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition  1 0.0328 0.03277   1.662  0.204
Residuals      47 0.9269 0.01972
```

```
>
```

Figure Target Accuracy

```
> target_retrievalacc = group_by(SemanticCuedRecall, Subject, PrimeCondition,
+                                CuedRecallAcc) %>%
+   summarise_at(vars(TargetAccuracy), mean)
> target_rmisc = Rmisc::summarySE(target_retrievalacc,
+                       measurevar = "TargetAccuracy",
+                       groupvars = c("PrimeCondition", "CuedRecallAcc"))
> library(ggplot2)
> library(ggthemes)
> target_rmisc %>% mutate(`Item Retrieval` = factor(CuedRecallAcc,
+                                           levels = unique(CuedRecallAcc),
+                     labels = c("Not Retrieved", "Retrieved")))%>%
+ ggplot(aes(x = PrimeCondition , y = TargetAccuracy,
+       group = `Item Retrieval`, fill = `Item Retrieval`))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.5)+
+   geom_errorbar(aes(ymin = TargetAccuracy - se, ymax = TargetAccuracy + se),
+                 width=.05, position=position_dodge(.5)) +
```

```
+    theme_few()+
+    scale_fill_manual(values= c("slategray4", "slategray1"))+
+    xlab("Item Condition") + ylab("Mean Target Accuracy") +
+    ggtitle("Target Retrieval Accuracy ") +
+      theme(axis.text = element_text(face = "bold", size = rel(1)),
+            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(1.4), hjust = .5))
```



5.1 Masters Retrieval Figure

```
> SemanticCuedRecall_fig = SemanticCuedRecall
> SemanticCuedRecall_fig$primefac = ordered(as.factor(as.character(SemanticCuedRecall_fi
+                       levels = c("Semantic", "Unrelated"))
> SemanticCuedRecall_fig$TargetAccuracy = as.numeric(as.character(SemanticCuedRecall_fig
> SemanticCuedRecall_fig$CuedRecallAcc_Fac = ordered(as.factor(as.character(SemanticCued
> targetacc2  = group_by(SemanticCuedRecall_fig, Subject, primefac,
+                        CuedRecallAcc_Fac) %>%
+   summarise_at(vars(TargetAccuracy), mean)
> ret_figure = Rmisc::summarySE(targetacc2,
+                     measurevar = "TargetAccuracy",
+                 groupvars = c("primefac", "CuedRecallAcc_Fac"))
> library(ggplot2)
> library(ggthemes)
> ret_figure  %>% mutate(PrimeType = factor(primefac,
+                                 levels = unique(primefac),
+                     labels = c("Semantic",
+                                 "Unrelated")),
+                     `Prime Retrieval` = factor(CuedRecallAcc_Fac,
+                                 levels = unique(CuedRecallAcc_Fac),
+                     labels = c("Retrieved", "Not Retrieved")))%>%
+    ggplot(aes(x = `Prime Retrieval`, y = TargetAccuracy,
+                           group =PrimeType ,
+                           fill = PrimeType)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+    geom_errorbar(aes(ymin = TargetAccuracy - se,
+                      ymax = TargetAccuracy + se),
+                 width=.08, position=position_dodge(.5)) +
+   theme_few()+
+ #   scale_fill_canva()+
+  scale_fill_manual(values = c(  "red",
+                                 "lightgreen"))+
+   xlab("Prime Retrieval") + ylab("Mean Target Accuracy") +
+ ggtitle(" Experiment 5") +
+   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)))
>
```

5.2 ANOVA

```
> target_retrievalacc[120,] = c(3, "Unrelated", 1, 0 )
> target_retrievalacc$Subject = as.factor(target_retrievalacc$Subject)
> target_retrievalacc$TargetAccuracy = as.numeric(target_retrievalacc$TargetAccuracy)
```

```
> target_retrievalacc$CuedRecallAcc = as.factor(target_retrievalacc$CuedRecallAcc)
> targetacc_aov = aov(data = target_retrievalacc,
+                     TargetAccuracy ~ PrimeCondition*CuedRecallAcc +
+                       Error(Subject/(PrimeCondition*CuedRecallAcc)))
> summary(targetacc_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 29   4.15  0.1431

Error: Subject:PrimeCondition
               Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition  1 0.0034 0.00341   0.061  0.807
Residuals      29 1.6269 0.05610

Error: Subject:CuedRecallAcc
              Df Sum Sq Mean Sq F value Pr(>F)
CuedRecallAcc  1 0.1416 0.14156   5.597 0.0249 *
Residuals     29 0.7334 0.02529
---
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:CuedRecallAcc
                             Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition:CuedRecallAcc  1 0.0031 0.00306   0.085  0.773
Residuals                    29 1.0482 0.03615
```

6 HLM Model

```
> library(lme4)
> # participant_acc = group_by(SemanticCuedRecall, Subject) %>%
> #   summarise_at(vars(TargetAccuracy, CuedRecallAcc), mean)
> #
> # participant_acc$MeanAcc = (participant_acc$TargetAccuracy +
> #                             participant_acc$CuedRecallAcc)/2
> #
> # colnames(participant_acc) = c("Subject", "TargetAcc", "PrimeAcc", "MeanAcc")
> #
> # SemanticCuedRecall2 = merge(SemanticCuedRecall, participant_acc[,c(1,3,4)],
> #                       by = c("Subject"))
>
> ## accounting for mean prime accuracy
>
> item_acc = group_by(SemanticCuedRecall, Stimuli1, PrimeCondition) %>%
+   summarise_at(vars(CuedRecallAcc), mean)
> colnames(item_acc) = c("Stimuli1","PrimeCondition","PrimeAcc")
```

```
> SemanticCuedRecall2 = merge(SemanticCuedRecall, item_acc,
+                         by = c("Stimuli1", "PrimeCondition"))
> SemanticCuedRecall2$TargetAccuracy = as.factor(SemanticCuedRecall$TargetAccuracy)
> SemanticCuedRecall2$CuedRecallAcc = as.factor(SemanticCuedRecall$CuedRecallAcc)
> SemanticCuedRecall2$FailedRetrieval = ifelse(SemanticCuedRecall2$TargetAccuracy == 1,0
> SemanticCuedRecall$FailedRetrieval = ifelse(SemanticCuedRecall$TargetAccuracy == 1,0,1
> contrasts(SemanticCuedRecall2$PrimeCondition)
```

```
          1
Semantic  1
Unrelated 0
```

```
> SemanticCuedRecall_hlm = glmer(data = SemanticCuedRecall2,
+                               TargetAccuracy ~ PrimeCondition*CuedRecallAcc +
+                                 PrimeAcc+
+                          (1|Subject) + (1|Stimuli1), family = "binomial",
+                            control=glmerControl(optimizer="bobyqa",
+            optCtrl=list(maxfun=100000)))
> summary(SemanticCuedRecall_hlm)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: TargetAccuracy ~ PrimeCondition * CuedRecallAcc + PrimeAcc +
    (1 | Subject) + (1 | Stimuli1)
   Data: SemanticCuedRecall2
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))

     AIC      BIC   logLik deviance df.resid
  1852.6   1889.5   -919.3   1838.6     1433

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.6373 -0.7806 -0.5388  0.9859  2.6904

Random effects:
 Groups   Name        Variance Std.Dev.
 Stimuli1 (Intercept) 0.4197   0.6478
 Subject  (Intercept) 0.0000   0.0000
Number of obs: 1440, groups:  Stimuli1, 48; Subject, 30

Fixed effects:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -0.70968    0.25225  -2.813  0.00490 **
PrimeCondition1                0.05641    0.16027   0.352  0.72487
CuedRecallAcc1                 0.55847    0.17069   3.272  0.00107 **
PrimeAcc                      -0.13461    0.45042  -0.299  0.76506
PrimeCondition1:CuedRecallAcc1 -0.08255    0.23147  -0.357  0.72136
```

```
---
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 CdRcA1 PrmAcc
PrimeCndtn1 -0.294
CudRcllAcc1 -0.307  0.467
PrimeAcc    -0.810 -0.024 -0.014
PrmCn1:CRA1  0.187 -0.700 -0.684  0.040
```

```
> # confint(SemanticCuedRecall_hlm)
> #
> # > confint(SemanticCuedRecall_hlm)
> # Computing profile confidence intervals ...
> #                                     2.5 %        97.5 %
> # .sig01                          0.4780585  0.8696471513
> # .sig02                          0.0000000  0.2099374355
> # (Intercept)                    -0.9371125  0.0004921197
> # PrimeCondition1                -0.2101249  0.2402437248
> # CuedRecallAcc1                 -0.4476967 -0.1124926698
> # PrimeAcc                       -0.9312133  0.8241912472
> # PrimeCondition1:CuedRecallAcc1 -0.1952867  0.2593870052
>
> # car::Anova(SemanticCuedRecall_hlm)
> # options(contrasts = c("contr.sum","contr.poly"))
> # anova(SemanticCuedRecall_hlm)
```

7 z-scoring RTs

RT prime and Target

```
> library(dplyr)
> colnames(SemanticCuedRecall) = c("Subject", "Session",       "Procedure",
+ "Trial",      "ActualPrime",  "PrimeCondition",       "PrimeDef",     "PrimeDefRT",
+ "PrimeDefinition",    "PrimeLength",  "PrimeResponse" ,
+ "PrimeResponseRT", "Stimuli1", "Target",      "TargetDefinition",
+ "TargetDefRT", "State",       "StateRT", "TargetResponse", "TargetResponseRT",
+ "TargetResponse", "RTrecognisePrime", "RTrecogniseTarget",
+ "PrimeRespType", "TargetRespType",
+                            "FailedRetrieval")
> SemanticCuedRecall_firsttrim_target = subset(SemanticCuedRecall,
+                                 SemanticCuedRecall$RTrecogniseTarget > 250 &
+                                SemanticCuedRecall$RTrecogniseTarget < 7000)
> SemanticCuedRecall_firsttrim_prime = subset(SemanticCuedRecall,
+                                 SemanticCuedRecall$RTrecognisePrime > 250 &
+                                SemanticCuedRecall$RTrecognisePrime < 7000)
```

```
> SemanticCuedRecall_firsttrim_targetdef = subset(SemanticCuedRecall,
+                                  SemanticCuedRecall$TargetDefRT > 250 &
+                                 SemanticCuedRecall$TargetDefRT < 9000)
>
```

RTRecogniseprime

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

RTRecogniseTarget

```
> ## FOR TARGET
> ## aggregate per subject all IVs and DVs
> meanRT = group_by(SemanticCuedRecall_firsttrim_target, Subject) %>%
+   summarise_at(vars(RTrecogniseTarget), mean)
> colnames(meanRT) = c("Subject", "MeanRTrecogTarget")
> sdRT = group_by(SemanticCuedRecall_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
> SemanticCuedRecall_z_target= merge(SemanticCuedRecall_firsttrim_target,
+                                RT_agg, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
```

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

TargetDefRT

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

8 Trimming z-RTs

```
> SemanticCuedRecall_z_trimmed_prime = subset(SemanticCuedRecall_z_prime,
+                           SemanticCuedRecall_z_prime$zPrimeRecogRT < 3 &
+                              SemanticCuedRecall_z_prime$zPrimeRecogRT > -3)
> SemanticCuedRecall_z_trimmed_target = subset(SemanticCuedRecall_z_target,
+                          SemanticCuedRecall_z_target$zTargetRecogRT < 3 &
+                              SemanticCuedRecall_z_target$zTargetRecogRT > -3)
> SemanticCuedRecall_z_trimmed_targetdef = subset(SemanticCuedRecall_z_targetdef,
SemanticCuedRecall_z_targetdef$zTargetRT < 3 &                      SemanticCu
```

9 Repeating z-scoring

9.1 For prime

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

9.2 For Target

```
> ## aggregate per subject all IVs and DVs
> meanRT_target = group_by(SemanticCuedRecall_z_trimmed_target, Subject) %>%
+   summarise_at(vars(RTrecogniseTarget), mean)
> colnames(meanRT_target) = c("Subject",
+                           "MeanRTrecogTarget_trim")
> sdRT_target = group_by(SemanticCuedRecall_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
> SemanticCuedRecall_final_z_target = merge(SemanticCuedRecall_z_trimmed_target,
+                                    RT_agg_target, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> SemanticCuedRecall_final_z_target = SemanticCuedRecall_final_z_target %>%
```

```
+                                   mutate( zTargetRecogRT_trim =
+                                              (RTrecogniseTarget -
+                                       MeanRTrecogTarget_trim)/sdRTrecogTarget_trim)
> ## checking: subject level means should be zero
>
> sub_pic = group_by(SemanticCuedRecall_final_z_target, Subject) %>%
+   summarise_at(vars(zTargetRecogRT_trim), mean)
>
```

9.3 For TargetDefRT

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

9.4 Combining z-RT Prime and Target

```
> ## now we have separately z-scored RTprime and RTtarget. Need to combine.
> ## taking only necessary columns
> SemanticCuedRecall_final_z_prime2 =
+   SemanticCuedRecall_final_z_prime[,c(1,4,36)]
> SemanticCuedRecall_final_z = merge(SemanticCuedRecall_final_z_target,
+                              SemanticCuedRecall_final_z_prime2,
+                              by  = c("Subject", "Trial"))
> primefinal_z_targetdef = merge(SemanticCuedRecall_final_z_targetdef,
+                              SemanticCuedRecall_final_z_prime2,
+                              by  = c("Subject", "Trial"))
```

10 Linear Models

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

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

     AIC      BIC   logLik deviance df.resid
  1580.0   1611.4   -784.0   1568.0     1393

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.3064 -0.6326 -0.3146  0.6796  3.6075

Random effects:
 Groups  Name        Variance Std.Dev.
 Target  (Intercept) 1.388    1.178
 Subject (Intercept) 1.035    1.018
Number of obs: 1399, groups:  Target, 48; Subject, 30

Fixed effects:
                                   Estimate Std. Error z value Pr(>|z|)
(Intercept)                         -0.5394     0.2698  -1.999  0.04560 *
PrimeCondition1                     -0.2002     0.1313  -1.524  0.12739
zPrimeRecogRT_trim                  -0.3091     0.1010  -3.059  0.00222 **
PrimeCondition1:zPrimeRecogRT_trim   0.1533     0.1448   1.059  0.28970
---
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 zPRRT_
PrimeCndtn1 -0.246
zPrmRcgRT_t  0.005  0.004
PrmC1:PRRT_ -0.003  0.047 -0.708
```

```
> # > confint(RTprime_acc_model)
> # Computing profile confidence intervals ...
> #                                           2.5 %       97.5 %
> # .sig01                                0.9233411  1.523687468
> # .sig02                                0.7497913  1.420254602
> # (Intercept)                          -1.0853659 -0.005711025
> # PrimeCondition1                      -0.4617874  0.059831448
> # zPrimeRecogRT_trim                   -0.5128525 -0.111067986
> # PrimeCondition1:zPrimeRecogRT_trim  -0.1337061  0.441776343
>
> car::Anova(RTprime_acc_model)
```

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

Response: TargetAccuracy
                                   Chisq Df Pr(>Chisq)
PrimeCondition                     2.485  1   0.114934
zPrimeRecogRT_trim                10.683  1   0.001081 **
PrimeCondition:zPrimeRecogRT_trim  1.121  1   0.289700
---
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_acc_model)
```

```
Analysis of Variance Table
                                  Df  Sum Sq Mean Sq F value
PrimeCondition                     1  2.0344  2.0344  2.0344
zPrimeRecogRT_trim                 1 10.9034 10.9034 10.9034
PrimeCondition:zPrimeRecogRT_trim  1  1.1444  1.1444  1.1444
```


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

```
Analysis of Variance Table
                                  Df  Sum Sq Mean Sq F value
PrimeCondition                     1  2.0344  2.0344  2.0344
zPrimeRecogRT_trim                 1 10.9034 10.9034 10.9034
PrimeCondition:zPrimeRecogRT_trim  1  1.1444  1.1444  1.1444
```


```
> RTprime_RT_model = lmer(data = SemanticCuedRecall_final_z,
+                     zTargetRecogRT_trim ~ zPrimeRecogRT_trim*PrimeCondition +
+                             (1|Subject) + (1|Target))
> summary(RTprime_RT_model)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: zTargetRecogRT_trim ~ zPrimeRecogRT_trim * PrimeCondition + (1 |
    Subject) + (1 | Target)
   Data: SemanticCuedRecall_final_z

REML criterion at convergence: 3772.8

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.8597 -0.6865 -0.1056  0.5176  3.4676

Random effects:
 Groups   Name        Variance Std.Dev.
 Target   (Intercept) 0.1740   0.4171
 Subject  (Intercept) 0.0000   0.0000
 Residual             0.8034   0.8963
Number of obs: 1399, groups:  Target, 48; Subject, 30

Fixed effects:
                                   Estimate Std. Error t value
(Intercept)                         0.01858    0.06921   0.269
zPrimeRecogRT_trim                  0.14199    0.03515   4.039
PrimeCondition1                    -0.02418    0.04836  -0.500
zPrimeRecogRT_trim:PrimeCondition1 -0.10084    0.05095  -1.979

Correlation of Fixed Effects:
            (Intr) zPrRRT_ PrmCn1
zPrmRcgRT_t -0.019
PrimeCndtn1 -0.350  0.024
zPrRRT_:PC1  0.014 -0.688   0.012
```

```
> # > confint(RTprime_RT_model)
> # Computing profile confidence intervals ...
> #                                         2.5 %       97.5 %
> # .sig01                              0.3279392  0.525207923
> # .sig02                              0.0000000  0.049713712
> # .sigma                              0.8625896  0.930180099
> # (Intercept)                        -0.1178693  0.155048382
> # zPrimeRecogRT_trim                  0.0732072  0.210982298
> # PrimeCondition1                    -0.1190239  0.070483170
> # zPrimeRecogRT_trim:PrimeCondition1 -0.2007485 -0.001098669
>
>
> car::Anova(RTprime_RT_model)
```

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

```
Response: zTargetRecogRT_trim
                                    Chisq Df Pr(>Chisq)
zPrimeRecogRT_trim                13.6104  1  0.0002249 ***
PrimeCondition                     0.2268  1  0.6339016
zPrimeRecogRT_trim:PrimeCondition  3.9170  1  0.0478003 *
---
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)
```

```
Analysis of Variance Table
                                  Df  Sum Sq Mean Sq F value
zPrimeRecogRT_trim                 1 11.0842 11.0842 13.7973
PrimeCondition                     1  0.1822  0.1822  0.2268
zPrimeRecogRT_trim:PrimeCondition  1  3.1468  3.1468  3.9170
```

```
> ## TARGET DEF MODEL
>
> RTprime_RTtargetdef_model = lmer(data = primefinal_z_targetdef,
+                     zTargetRT_trim ~ PrimeCondition*zPrimeRecogRT_trim +
+                             (1|Subject) + (1|Target))
> summary(RTprime_RTtargetdef_model)
```

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

REML criterion at convergence: 3166.9

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.67587 -0.76306 -0.08912  0.67217  3.15491

Random effects:
 Groups   Name        Variance Std.Dev.
 Target   (Intercept) 0.1735   0.4165
 Subject  (Intercept) 0.0000   0.0000
 Residual             0.7972   0.8929
Number of obs: 1174, groups:  Target, 48; Subject, 30

Fixed effects:
                                    Estimate Std. Error t value
(Intercept)                        -0.021679   0.070713  -0.307
PrimeCondition1                     0.091415   0.052949   1.726
zPrimeRecogRT_trim                  0.079169   0.037776   2.096
```

```
PrimeCondition1:zPrimeRecogRT_trim -0.009479   0.054834  -0.173

Correlation of Fixed Effects:
            (Intr) PrmCn1 zPRRT_
PrimeCndtn1 -0.374
zPrmRcgRT_t -0.019  0.026
PrmC1:PRRT_  0.014  0.010 -0.682
```

```
> car::Anova(RTprime_RTtargetdef_model)
```

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

Response: zTargetRT_trim
                                    Chisq Df Pr(>Chisq)
PrimeCondition                     2.9867  1   0.083950 .
zPrimeRecogRT_trim                 7.3214  1   0.006814 **
PrimeCondition:zPrimeRecogRT_trim  0.0299  1   0.862752
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
>
```

11 Plotting Model Fits

11.0.1 Model 1

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

```
[1] 0.9842339
```

```
> # this is the model
>
> # RTprime_acc_model = glmer(data = SemanticCuedRecall_final_z,
> #                           TargetAccuracy ~ PrimeCondition*zPrimeRecogRT_trim +
> #                                  (1|Subject) + (1|Target), family = binomial )
> # summary(RTprime_acc_model)
>
> primert_model = lmer(data = SemanticCuedRecall_final_z,
+                      zPrimeRecogRT_trim ~ 1 + (1 | Subject) +
+                        (1|Target))
> summary(primert_model)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: zPrimeRecogRT_trim ~ 1 + (1 | Subject) + (1 | Target)
   Data: SemanticCuedRecall_final_z
```

```
REML criterion at convergence: 3884.1

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.8540 -0.7257 -0.1912  0.5597  3.3669

Random effects:
 Groups   Name        Variance Std.Dev.
 Target   (Intercept) 0.06866  0.2620
 Subject  (Intercept) 0.00000  0.0000
 Residual             0.90120  0.9493
Number of obs: 1399, groups:  Target, 48; Subject, 30

Fixed effects:
            Estimate Std. Error t value
(Intercept) -0.007959   0.045560  -0.175
```

```
> VarCorr(primert_model)
```

```
 Groups   Name        Std.Dev.
 Target   (Intercept) 0.26203
 Subject  (Intercept) 0.00000
 Residual             0.94932
```

```
> 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 = SemanticCuedRecall_final_z,
+                         zPrimeRecogRT_trim ~ 1 + PrimeCondition +
+                       (1|Subject) + (1|Target))
> prime_Inc_1_U <- 0*fixef(primert_model_2)[1]
> prime_Inc_1_R <- 1*fixef(primert_model_2)[2]
> predict_data_U <- with(SemanticCuedRecall_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_R <- with(SemanticCuedRecall_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 = 1))
> predict_data = rbind(predict_data_U,
+                     predict_data_R)
> predict_data$PrimeCondition = ifelse(predict_data$PrimeCondition == 0,
```

```
+                                       "Unrelated","Semantic")
> predict_data = predict_data %>%
+   mutate(predicted_values = predict(RTprime_acc_model,
+           newdata = predict_data, re.form = NA))
> predict_data$prob = exp(predict_data$predicted_values)/(1+exp(predict_data$predicted_v
> predict_data %>%
+   mutate(PrimeType = factor(PrimeCondition, levels = unique(PrimeCondition),
+                     labels = c("Unrelated", "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 5")+
+ theme_few() +
+   scale_color_manual(values = c("lightgreen","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))
```

11.1 Model 2

```
> # RTprime_RT_model = lmer(data = SemanticCuedRecall_final_z,
> #                      zTargetRecogRT_trim ~ zPrimeRecogRT_trim*PrimeCondition +
> #                              (1|Subject) + (1|Target))
> # summary(RTprime_RT_model)
>
> primert_model = lmer(data = SemanticCuedRecall_final_z,
+                      zPrimeRecogRT_trim ~ 1 + (1 | Subject) +
+                        (1|Target))
> summary(primert_model)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: zPrimeRecogRT_trim ~ 1 + (1 | Subject) + (1 | Target)
   Data: SemanticCuedRecall_final_z

REML criterion at convergence: 3884.1

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.8540 -0.7257 -0.1912  0.5597  3.3669

Random effects:
 Groups   Name        Variance Std.Dev.
 Target   (Intercept) 0.06866  0.2620
 Subject  (Intercept) 0.00000  0.0000
 Residual             0.90120  0.9493
```

```
Number of obs: 1399, groups:  Target, 48; Subject, 30

Fixed effects:
             Estimate Std. Error t value
(Intercept) -0.007959   0.045560  -0.175
```

```
> VarCorr(primert_model)
```

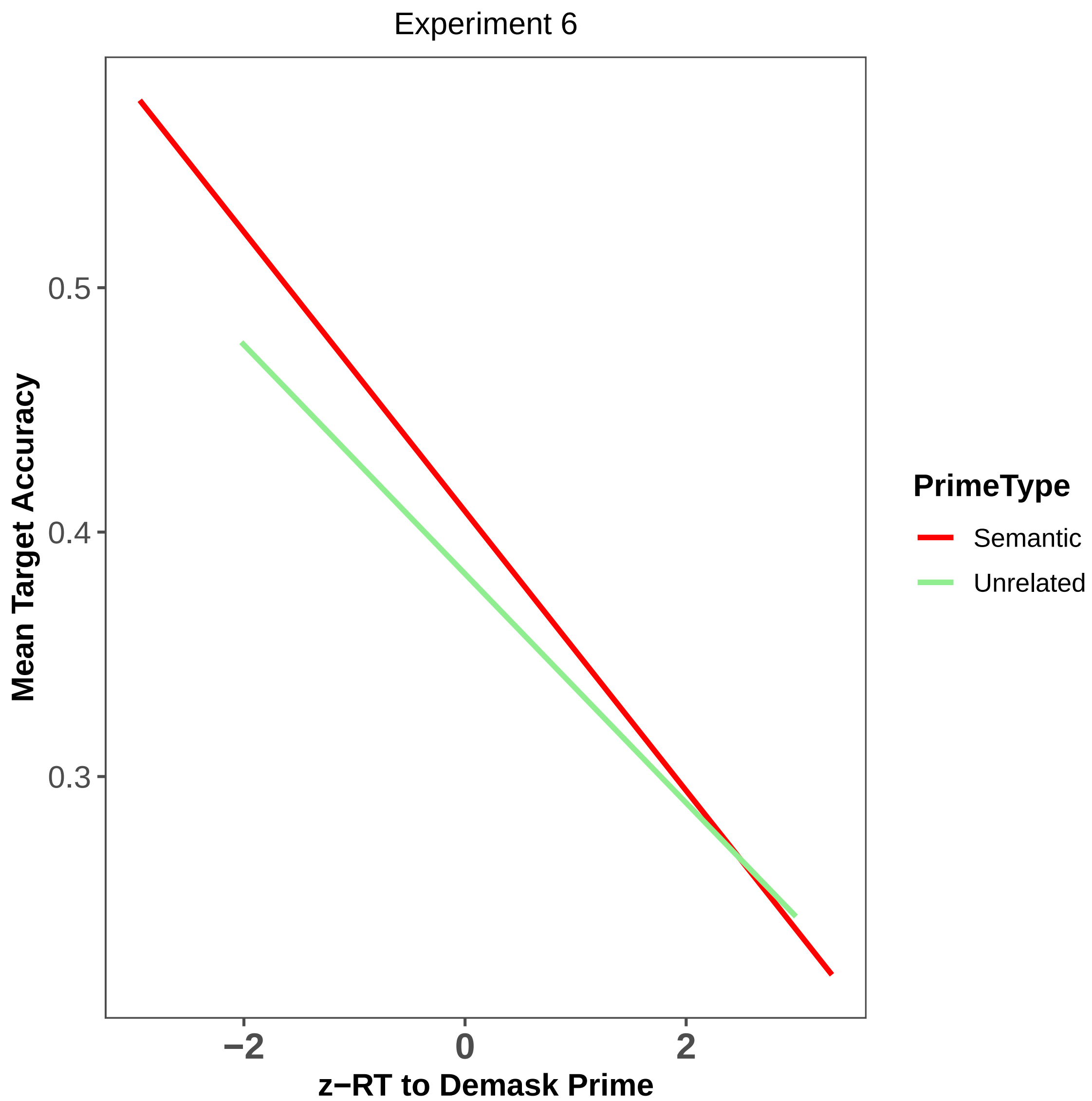
```
 Groups   Name        Std.Dev.
 Target   (Intercept) 0.26203
 Subject  (Intercept) 0.00000
 Residual             0.94932
```

```
> 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 = SemanticCuedRecall_final_z,
+                         zPrimeRecogRT_trim ~ 1 + PrimeCondition +
+                       (1|Subject) + (1|Target))
> prime_Inc_1_U <- 0*fixef(primert_model_2)[1]
> prime_Inc_1_R <- 1*fixef(primert_model_2)[2]
> predict_data_U <- with(SemanticCuedRecall_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_R <- with(SemanticCuedRecall_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 = 1))
> predict_data = rbind(predict_data_U,
+                      predict_data_R)
> predict_data$PrimeCondition = ifelse(predict_data$PrimeCondition == 0,
+                                      "Unrelated","Semantic")
> predict_data = predict_data %>%
+   mutate(predicted_values = predict(RTprime_RT_model,
+           newdata = predict_data, re.form = NA))
> predict_data %>%
+   mutate(PrimeType = factor(PrimeCondition, levels = unique(PrimeCondition),
+                    labels = c("Unrelated", "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 5")+
```

```
+ theme_few() +
+   scale_color_manual(values = c("lightgreen","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))
>
```

11.2 RAW ACC Model

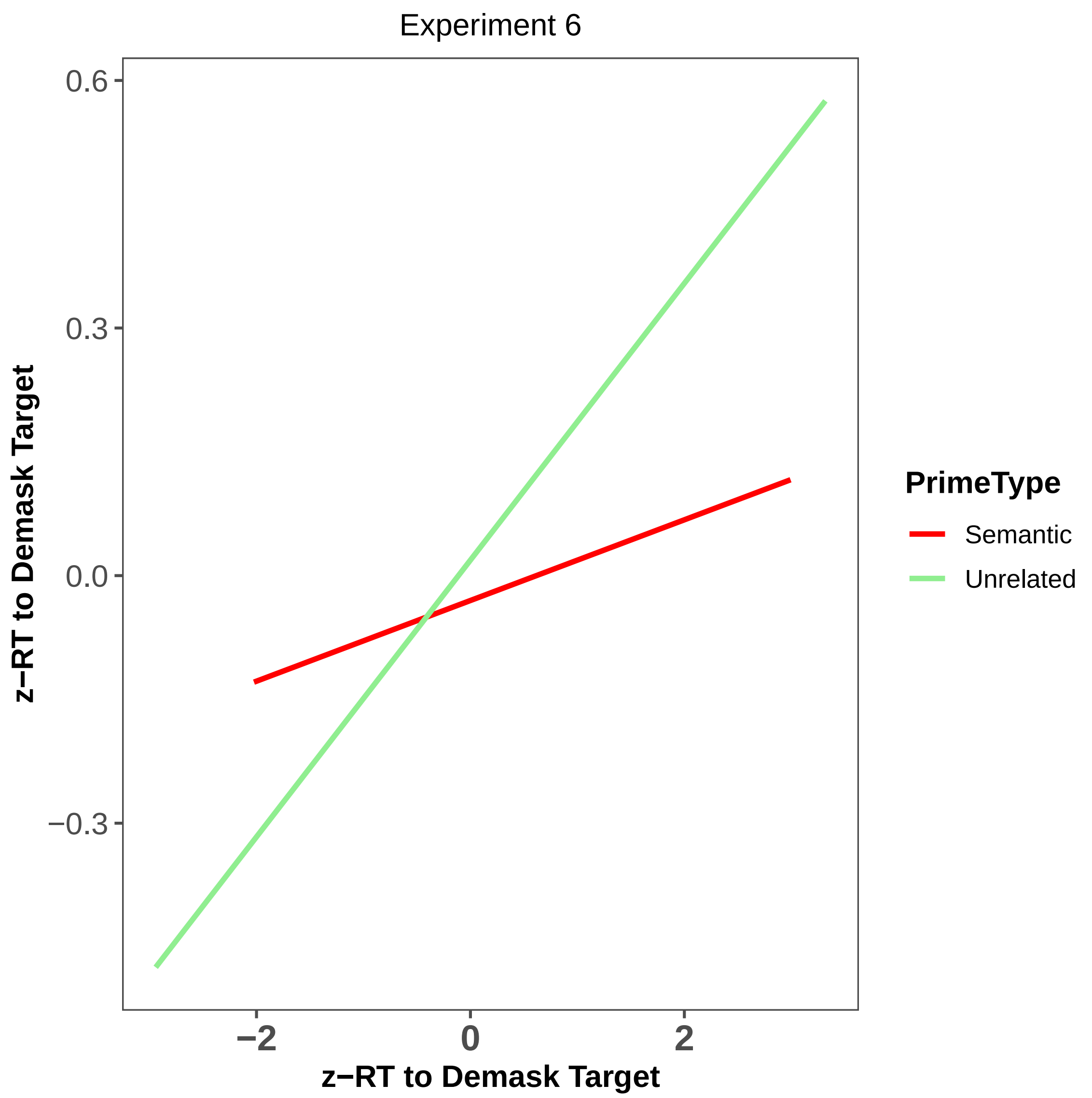
```
> SemanticCuedRecall_final_z$primefac = ordered(as.factor(as.character(SemanticCuedRecal
> SemanticCuedRecall_final_z %>%
+   mutate(PrimeType = factor(primefac, levels = unique(primefac),
+                     labels = c("Semantic",
+                                "Unrelated")))%>%
+   ggplot(aes(x = zPrimeRecogRT_trim, y = TargetAccuracy,
+              group = PrimeType, color = PrimeType)) +
+   geom_smooth(method = "lm", se = FALSE)+
+     xlab("z-RT to Demask Prime") + ylab ("Mean Target Accuracy")+
+   ggtitle("Experiment 6")+
+ theme_few() +
+ scale_color_manual(values = c( "red","lightgreen"))+
+   ggtitle("Experiment 6") +
+   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)))
>
```



11.3 RAW RT Model

```
> SemanticCuedRecall_final_z$PrimeType = ordered(as.factor(as.character(SemanticCuedReca
> SemanticCuedRecall_final_z %>%
+   ggplot(aes(x = zPrimeRecogRT_trim, y = zTargetRecogRT_trim,
+               group = PrimeType, color = PrimeType)) +
+   geom_smooth(method = "lm", se = FALSE, size = 1)+
+     xlab("z-RT to Demask Target") + ylab ("z-RT to Demask Target")+
+ theme_few() +
+ scale_color_manual(values = c( "red","lightgreen"))+
```

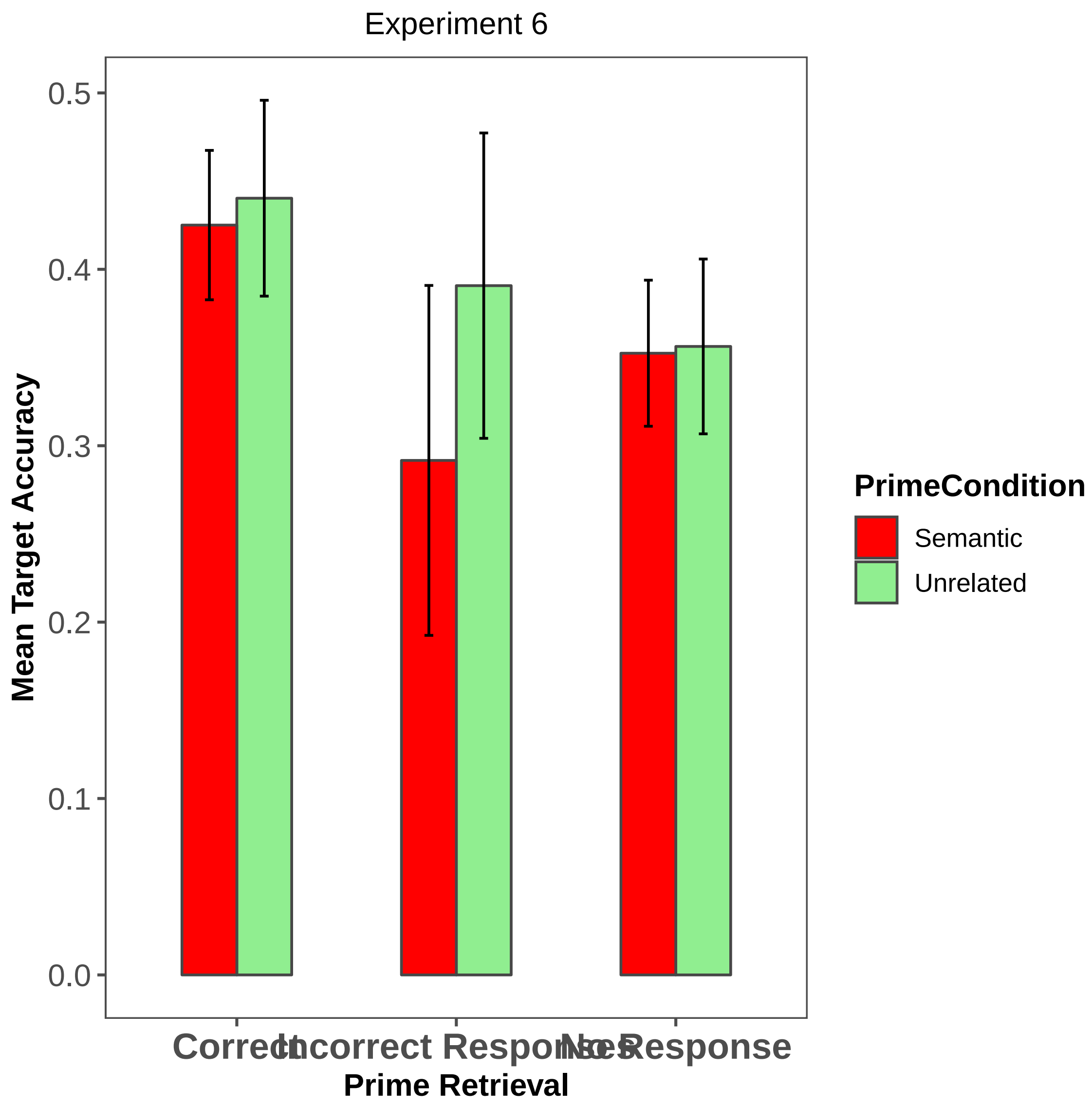
```
+    ggtitle("Experiment 6") +
+    theme(axis.text = element_text(size = rel(1)),
+             axis.title = element_text(face = "bold", size = rel(1)),
+             legend.title = element_text(face = "bold", size = rel(1)),
+            plot.title = element_text(hjust = .5, size = rel(1)),
+            axis.text.x = element_text(face = "bold", size = rel(1.2)))
>
```



12 Response Analysis

12.1 All Responses

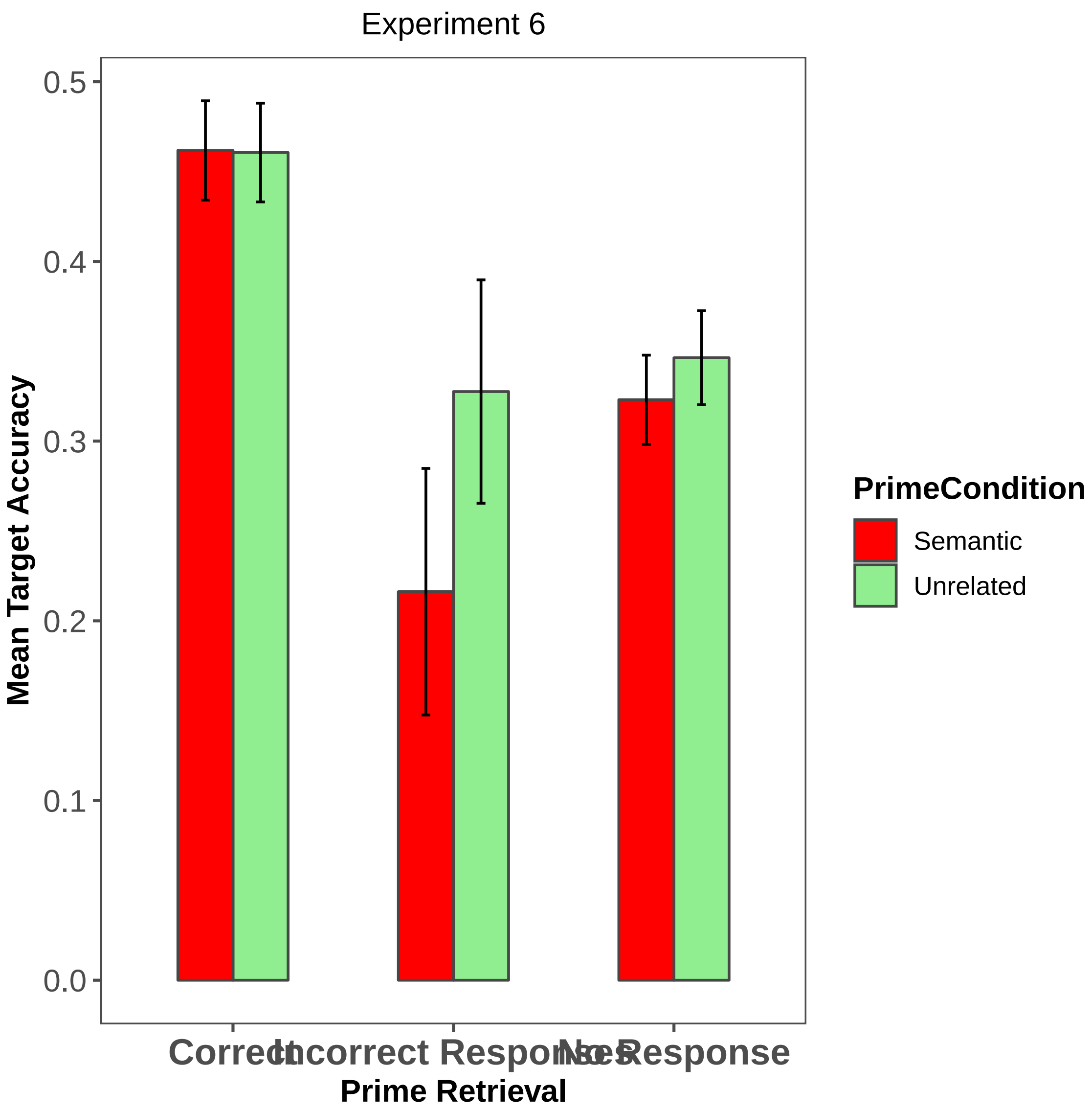
```
> SemanticCuedRecall = read.csv("E5_SemanticCuedRecall_FINAL.csv",
+                          header = TRUE, sep = ",")
> SemanticCuedRecall$Response = ifelse(SemanticCuedRecall$PrimeRespType %in%
+                                   c("Related Word", "Incorrect"), "Incorrect Responses"
+                                 ifelse(SemanticCuedRecall$PrimeRespType == "No Response
+                                        "No Response", "Correct"))
> SemanticCuedRecall$Response = ordered(as.factor(as.character(SemanticCuedRecall$Respon
+                       levels = c("Correct", "Incorrect Responses", "No Response"))
> SemanticCuedRecall_subject = group_by(SemanticCuedRecall,
+                                        Subject, PrimeCondition, Response) %>%
+   summarize_at(vars(TargetAccuracy), mean)
> ret_figure = Rmisc::summarySE(SemanticCuedRecall_subject,
+                     measurevar = "TargetAccuracy",
+                 groupvars = c("PrimeCondition", "Response"))
> library(ggplot2)
> library(ggthemes)
> library(dplyr)
> ret_figure  %>%
+    ggplot(aes(x = Response, y = TargetAccuracy,
+                           group =PrimeCondition ,
+                           fill = PrimeCondition)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+    geom_errorbar(aes(ymin = TargetAccuracy - se,
+                      ymax = TargetAccuracy + se),
+                 width=.08, position=position_dodge(.5)) +
+   theme_few()+
+ #   scale_fill_canva()+
+  scale_fill_manual(values = c(  "red",
+                               "lightgreen"))+
+   xlab("Prime Retrieval") + ylab("Mean Target Accuracy") +
+ ggtitle("Experiment 6") +
+   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)))
```



12.2 Incorrect Responses

```
> ret_figure = Rmisc::summarySE(SemanticCuedRecall,
+                        measurevar = "TargetAccuracy",
+                    groupvars = c("PrimeCondition", "Response"))
> library(ggplot2)
> library(ggthemes)
> library(dplyr)
> ret_figure  %>%
+     ggplot(aes(x = Response, y = TargetAccuracy,
```

```
+                             group =PrimeCondition ,
+                             fill = PrimeCondition)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+    geom_errorbar(aes(ymin = TargetAccuracy - se,
+                      ymax = TargetAccuracy + se),
+                 width=.08, position=position_dodge(.5)) +
+   theme_few()+
+ #   scale_fill_canva()+
+  scale_fill_manual(values = c(  "red",
+                                  "lightgreen"))+
+   xlab("Prime Retrieval") + ylab("Mean Target Accuracy") +
+ ggtitle("Experiment 6") +
+   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)))
```



12.3 LME

```
> SemanticCuedRecall$Response = as.factor(SemanticCuedRecall$Response)
> contrasts(SemanticCuedRecall$Response) = contr.treatment(3, base = 1)
> TOTFeedback_hlm2 = glmer(data = SemanticCuedRecall,
+                               TargetAccuracy ~ PrimeCondition*Response +
+                         (1|Subject) + (1|Target.Trial.), family = "binomial")
> summary(TOTFeedback_hlm2)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
```

```
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: TargetAccuracy ~ PrimeCondition * Response + (1 | Subject) +
    (1 | Target.Trial.)
   Data: SemanticCuedRecall

     AIC      BIC   logLik deviance df.resid
  1622.6   1664.8   -803.3   1606.6     1432

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.4628 -0.6228 -0.3227  0.6841  4.5007

Random effects:
 Groups        Name        Variance Std.Dev.
 Target.Trial. (Intercept) 1.3933   1.1804
 Subject       (Intercept) 0.8767   0.9363
Number of obs: 1440, groups:  Target.Trial., 48; Subject, 30

Fixed effects:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)               -0.42177    0.26339  -1.601   0.1093
PrimeCondition1           -0.09320    0.09381  -0.993   0.3205
Response2                 -0.66002    0.31357  -2.105   0.0353 *
Response3                 -0.39095    0.15630  -2.501   0.0124 *
PrimeCondition1:Response2 -0.28687    0.29778  -0.963   0.3354
PrimeCondition1:Response3  0.05222    0.13667   0.382   0.7024
---
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 Rspns2 Rspns3 PC1:R2
PrimeCndtn1 -0.009
Response2   -0.137  0.010
Response3   -0.298  0.032  0.259
PrmCndt1:R2  0.012 -0.317  0.246 -0.025
PrmCndt1:R3  0.004 -0.704 -0.013 -0.043  0.213
```

```
> car::Anova(TOTFeedback_hlm2)
```

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

Response: TargetAccuracy
                         Chisq Df Pr(>Chisq)
PrimeCondition          1.7300  1    0.18841
Response                7.8537  2    0.01971 *
PrimeCondition:Response 1.2896  2    0.52477
---
```

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