TOT Cued Recall Analysis

Abhilasha Kumar

September 30, 2018

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("SemanticCuedRecall.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
```

2 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
```

3 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: 21.7

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.42385 -0.60594 -0.07963  0.60866  2.29816

Random effects:
 Groups   Name        Variance Std.Dev.
 Subject  (Intercept) 0.00000  0.00
 Residual             0.06252  0.25
Number of obs: 72, groups:  Subject, 10

Fixed effects:
                                                    Estimate Std. Error
(Intercept)                                          0.71716    0.07907
PrimeConditionUnrelated                             -0.11666    0.11182
CuedRecallAcc1                                      -0.11928    0.11488
TargetAccuracy1                                     -0.40289    0.11488
PrimeConditionUnrelated:CuedRecallAcc1               0.08390    0.16512
PrimeConditionUnrelated:TargetAccuracy1              0.24627    0.16247
CuedRecallAcc1:TargetAccuracy1                       0.31824    0.16459
PrimeConditionUnrelated:CuedRecallAcc1:TargetAccuracy1 -0.16687    0.23647
                                                    t value
(Intercept)                                           9.070
PrimeConditionUnrelated                              -1.043
CuedRecallAcc1                                       -1.038
```

```
TargetAccuracy1                                          -3.507
PrimeConditionUnrelated:CuedRecallAcc1                    0.508
PrimeConditionUnrelated:TargetAccuracy1                   1.516
CuedRecallAcc1:TargetAccuracy1                            1.933
PrimeConditionUnrelated:CuedRecallAcc1:TargetAccuracy1   -0.706

Correlation of Fixed Effects:
            (Intr) PrmCnU CdRcA1 TrgtA1 PrCU:CRA1 PCU:TA CRA1:T
PrmCndtnUnr -0.707
CudRcllAcc1 -0.688  0.487
TrgtAccrcy1 -0.688  0.487  0.474
PrmCnU:CRA1  0.479 -0.677 -0.696 -0.330
PrmCndU:TA1  0.487 -0.688 -0.335 -0.707  0.466
CdRclA1:TA1  0.480 -0.340 -0.698 -0.698  0.486     0.494
PCU:CRA1:TA -0.334  0.473  0.486  0.486 -0.698    -0.687 -0.696
```

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

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

Response: prop
                                          Chisq Df Pr(>Chisq)
PrimeCondition                           0.0031  1   0.955743
CuedRecallAcc                            0.3940  1   0.530189
TargetAccuracy                           8.2911  1   0.003984 **
PrimeCondition:CuedRecallAcc             0.0005  1   0.982879
PrimeCondition:TargetAccuracy            2.0133  1   0.155929
CuedRecallAcc:TargetAccuracy             4.0351  1   0.044563 *
PrimeCondition:CuedRecallAcc:TargetAccuracy 0.4980  1   0.480388
---
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.

4 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
```

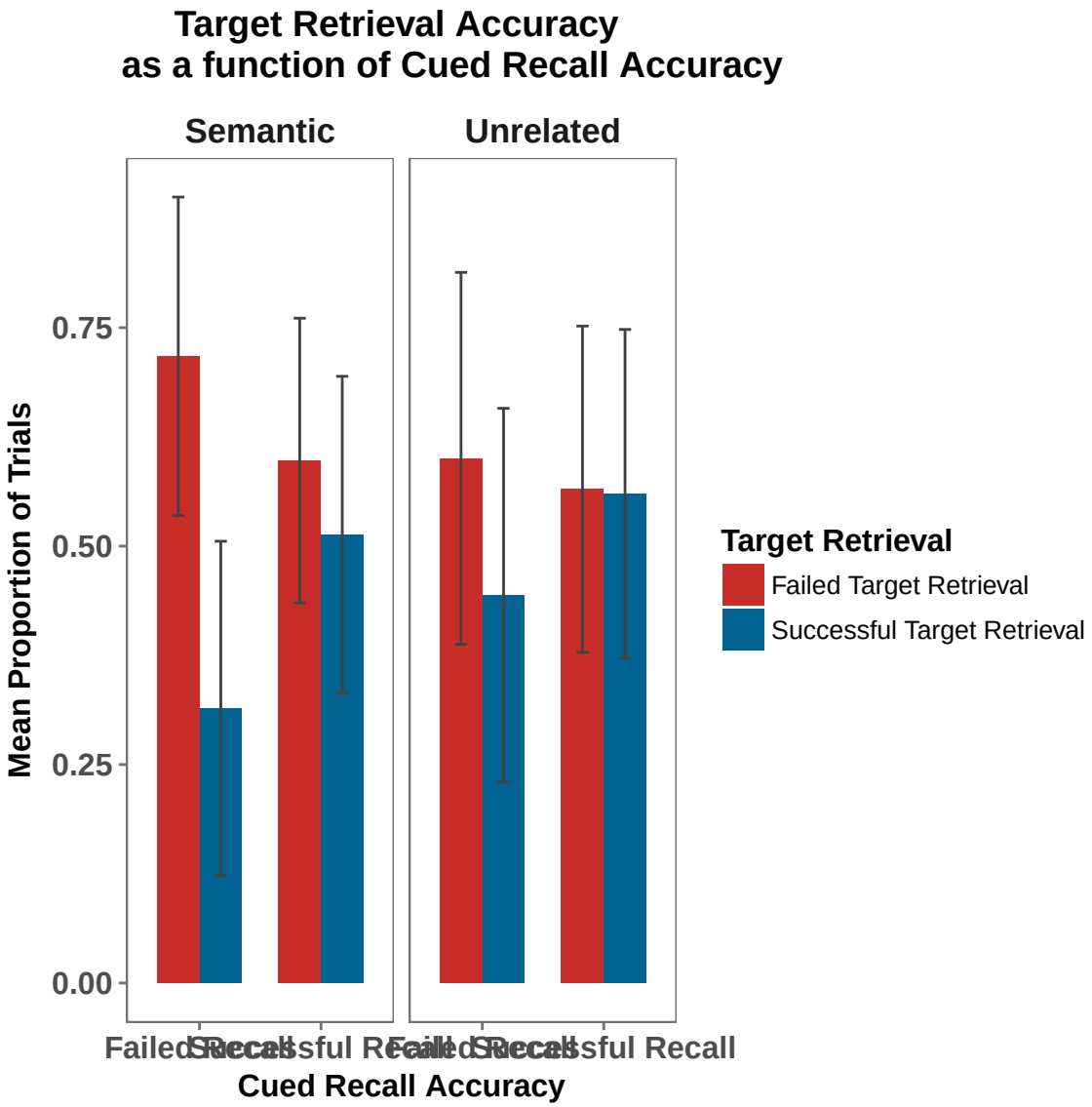
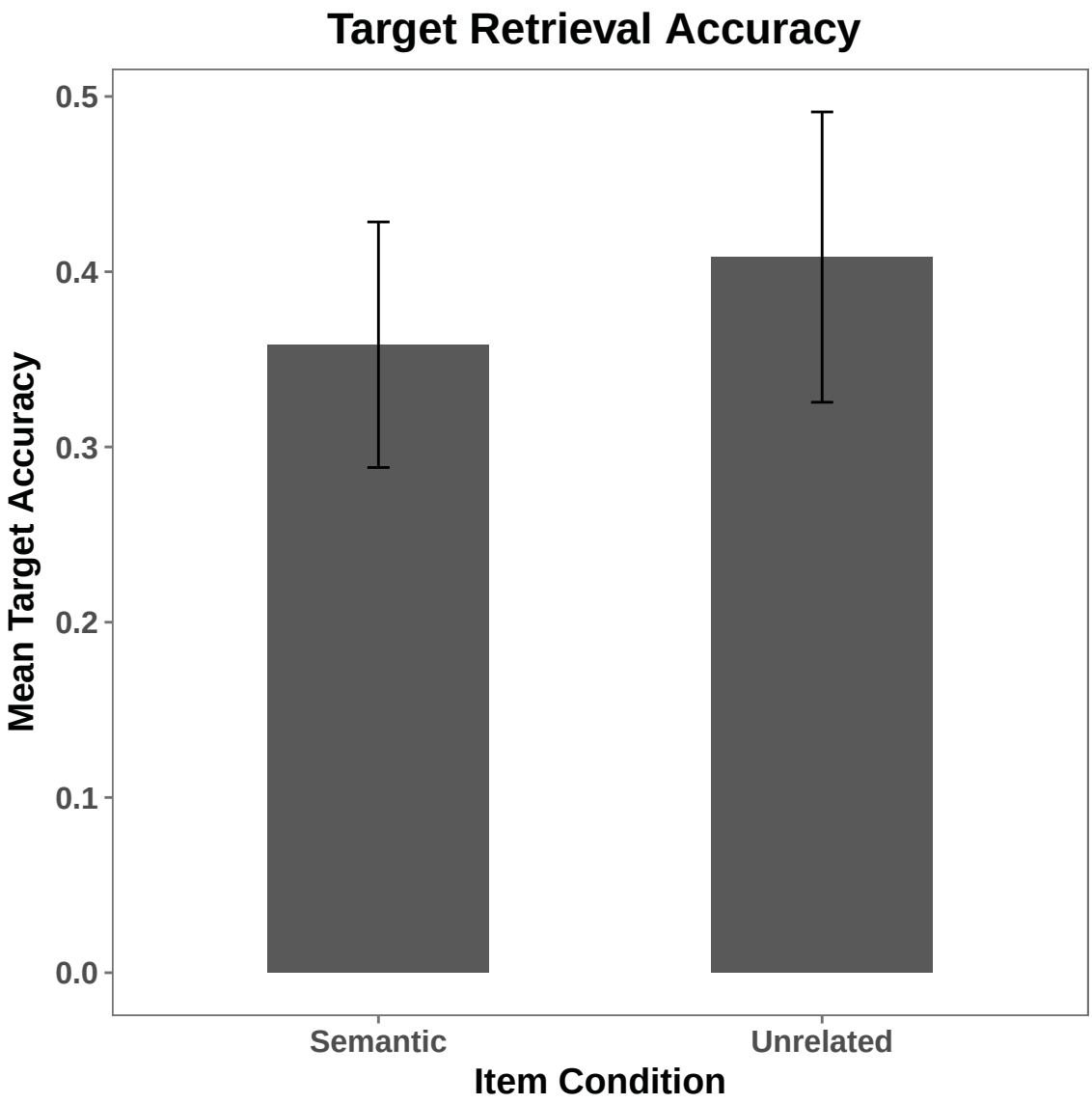


Figure Overall Target Accuracy

```
> 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

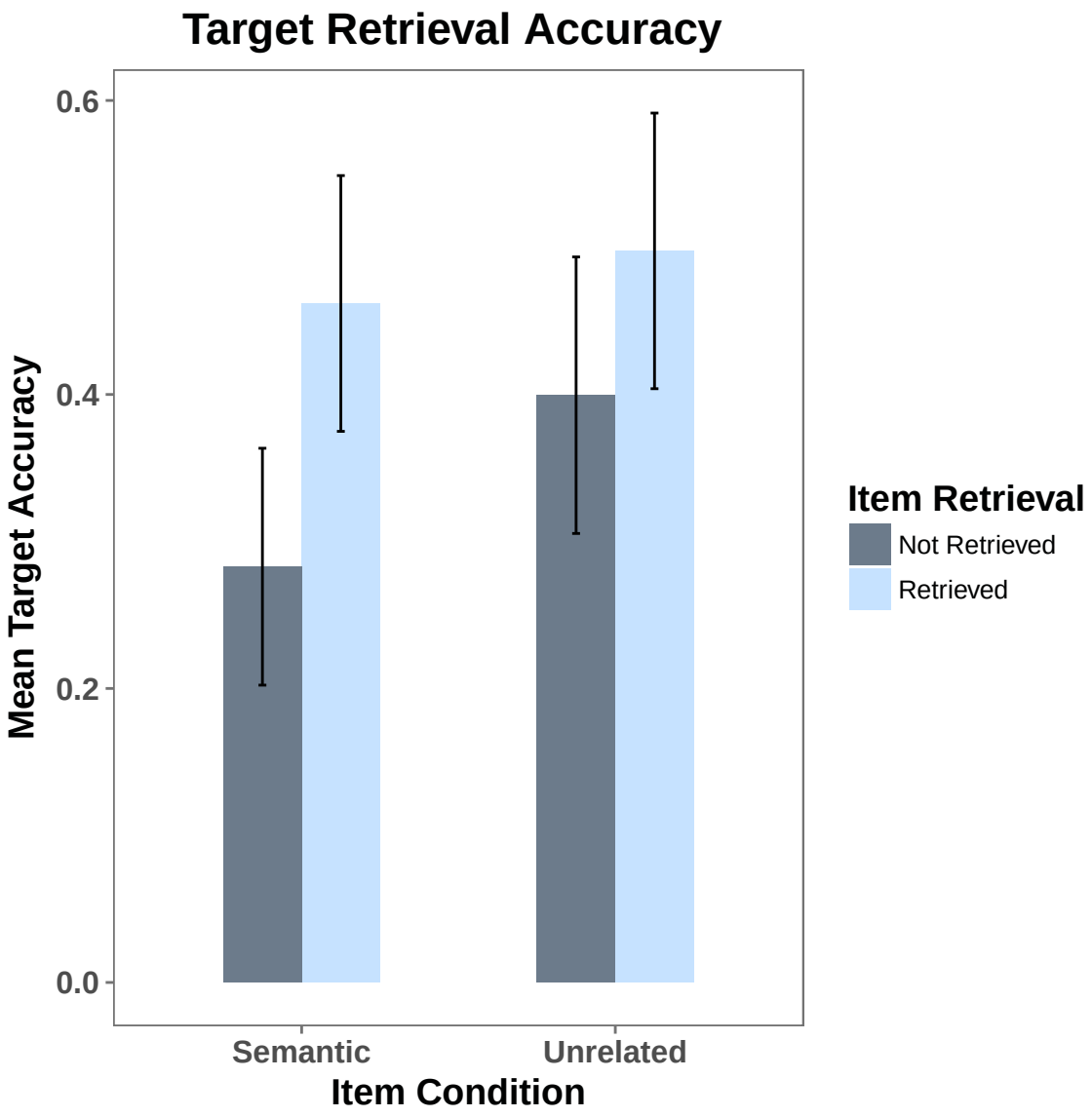
```
> prime_targetacc$Subject = as.factor(prime_targetacc$Subject)
> targetacc_aov = aov(data = prime_targetacc,
+                     TargetAccuracy ~ PrimeCondition +
+                       Error(Subject/PrimeCondition))
> summary(targetacc_aov)
```

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

```
Residuals  9 0.9153  0.1017

Error: Subject:PrimeCondition
               Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition  1 0.0125 0.01250   0.783  0.399
Residuals       9 0.1437 0.01597
```

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))
```



4.1 ANOVA

```
> target_retrievalacc[40,] = 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  9  1.853  0.2059

Error: Subject:PrimeCondition
               Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition  1 0.0263 0.02634   0.371  0.558
Residuals       9 0.6392 0.07102

Error: Subject:CuedRecallAcc
              Df  Sum Sq Mean Sq F value  Pr(>F)
CuedRecallAcc  1 0.12935 0.12935   16.04 0.00308 **
Residuals      9 0.07256 0.00806
---
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 0.0427 0.04268   1.089  0.324
Residuals                     9 0.3527 0.03919
```

5 HLM Model

```
> library(lme4)
> SemanticCuedRecall$TargetAccuracy = as.factor(SemanticCuedRecall$TargetAccuracy)
> SemanticCuedRecall$CuedRecallAcc = as.factor(SemanticCuedRecall$CuedRecallAcc)
> SemanticCuedRecall$FailedRetrieval = ifelse(SemanticCuedRecall$TargetAccuracy == 1,0,1
> SemanticCuedRecall_hlm = glmer(data = SemanticCuedRecall,
+                                TargetAccuracy ~ PrimeCondition*CuedRecallAcc +
+                            (1|Subject), family = "binomial")
> summary(SemanticCuedRecall_hlm)
```

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

     AIC      BIC   logLik deviance df.resid
   570.7    591.6   -280.3    560.7      475

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.8340 -0.7581 -0.3648  0.8381  4.5251

Random effects:
```

```
 Groups  Name        Variance Std.Dev.
 Subject (Intercept) 1.09     1.044
Number of obs: 480, groups:  Subject, 10

Fixed effects:
                                      Estimate Std. Error z value Pr(>|z|)
(Intercept)                            -1.0922     0.3965  -2.754  0.00588 **
PrimeConditionUnrelated                 0.3556     0.2932   1.213  0.22516
CuedRecallAcc1                          0.7326     0.3184   2.301  0.02138 *
PrimeConditionUnrelated:CuedRecallAcc1 -0.1894     0.4218  -0.449  0.65351
---
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) PrmCnU CdRcA1
PrmCndtnUnr -0.385
CudRcllAcc1 -0.380  0.467
PrmCnU:CRA1  0.263 -0.702 -0.647
```

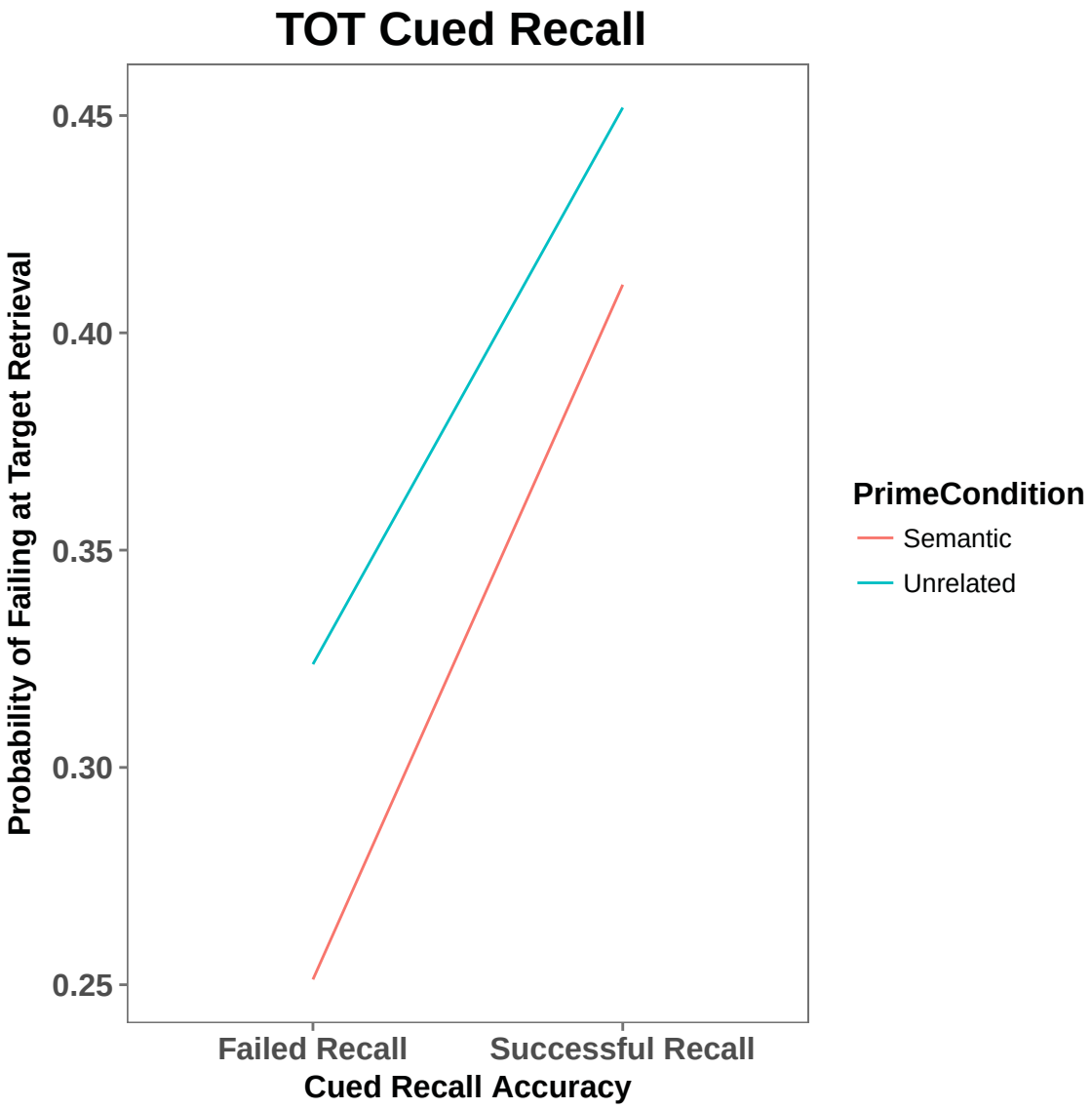
```
> car::Anova(SemanticCuedRecall_hlm)
```

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

Response: TargetAccuracy
                             Chisq Df Pr(>Chisq)
PrimeCondition              1.5900  1   0.207333
CuedRecallAcc               6.9572  1   0.008348 **
PrimeCondition:CuedRecallAcc 0.2015  1   0.653513
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

5.0.1 Plot

```
> fixed.frame ←
+   data.frame(expand.grid( PrimeCondition = c("Semantic", "Unrelated"),
+                           CuedRecallAcc = c("0","1"))) %>%
+   mutate(pred = predict(SemanticCuedRecall_hlm, newdata = ., re.form = NA))
> fixed.frame$odds = exp(fixed.frame$pred)
> fixed.frame$prob = fixed.frame$odds/(1+ fixed.frame$odds)
> fixed.frame$failure = 1 - fixed.frame$prob
> fixed.frame %>%
+   mutate(CuedRecallAccuracy = factor(CuedRecallAcc,
+     levels = unique(CuedRecallAcc),
+                      labels = c("Failed Recall", "Successful Recall")))%>%
+ ggplot(aes(x = CuedRecallAccuracy, y = prob,
+            group = PrimeCondition, color = PrimeCondition))+
+   geom_line()+
```

```
+ # geom_bar(stat = "identity", position = "dodge",
+   #          width = 0.7, color = "black")+
+   theme_few()+
+    xlab("Cued Recall Accuracy") + ylab("Probability of Failing at Target Retrieval") +
+    ggtitle("TOT Cued Recall ")  +
+     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(1.5), hjust = .5),
+           strip.text.x = element_text(face = "bold", size = rel(1.4)))
>
```



6 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",
+                            "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)
>
```

7 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
```

8 Repeating z-scoring

8.1 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)
>
```

8.2 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)
>
```

8.3 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)
>
```

8.4 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(1,4,34)]
> 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"))
```

9 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
   523.6    548.5   -255.8    511.6      461

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.5093 -0.5950 -0.2303  0.6440  2.8904
```

```
Random effects:
 Groups   Name        Variance Std.Dev.
 Target   (Intercept) 0.9539   0.9767
 Subject  (Intercept) 2.4806   1.5750
Number of obs: 467, groups:  Target, 48; Subject, 10

Fixed effects:
                                          Estimate Std. Error z value Pr(>|z|)
(Intercept)                                -0.9849     0.5510  -1.788   0.0738
PrimeConditionUnrelated                     0.3679     0.2363   1.557   0.1195
zPrimeRecogRT_trim                         -0.3580     0.2025  -1.767   0.0772
PrimeConditionUnrelated:zPrimeRecogRT_trim  -0.1440     0.2805  -0.513   0.6077

(Intercept)                                .
PrimeConditionUnrelated
zPrimeRecogRT_trim                         .
PrimeConditionUnrelated:zPrimeRecogRT_trim
---
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) PrmCnU zPRRT_
PrmCndtnUnr -0.221
zPrmRcgRT_t  0.055 -0.122
PrmCU:PRRT_ -0.031  0.078 -0.753
```

```
> car::Anova(RTprime_acc_model)
```

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

Response: TargetAccuracy
                                  Chisq Df Pr(>Chisq)
PrimeCondition                   2.5661  1   0.109178
zPrimeRecogRT_trim              10.7046  1   0.001069 **
PrimeCondition:zPrimeRecogRT_trim  0.2636  1   0.607688
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

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

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

```
REML criterion at convergence: 1306.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.8319 -0.6715 -0.2423  0.4731  3.2825

Random effects:
 Groups    Name        Variance Std.Dev.
 Target    (Intercept) 0.0826   0.2874
 Subject   (Intercept) 0.0000   0.0000
 Residual              0.8797   0.9379
Number of obs: 467, groups:  Target, 48; Subject, 10

Fixed effects:
                                           Estimate Std. Error t value
(Intercept)                                -0.03569    0.07445  -0.479
PrimeConditionUnrelated                     0.06785    0.08740   0.776
zPrimeRecogRT_trim                          0.10424    0.06974   1.495
PrimeConditionUnrelated:zPrimeRecogRT_trim  0.12098    0.09374   1.291

Correlation of Fixed Effects:
            (Intr) PrmCnU zPRRT_
PrmCndtnUnr -0.588
zPrmRcgRT_t  0.096 -0.082
PrmCU:PRRT_ -0.073  0.022 -0.753
```

```
> car::Anova(RTprime_RT_model)
```

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

Response: zTargetRecogRT_trim
                                  Chisq Df Pr(>Chisq)
PrimeCondition                   0.5592  1  0.4546020
zPrimeRecogRT_trim              14.0513  1  0.0001779 ***
PrimeCondition:zPrimeRecogRT_trim  1.6656  1  0.1968430
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> ## 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: 1002

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.28999 -0.73773 -0.03439  0.66594  3.14526

Random effects:
 Groups    Name        Variance Std.Dev.
 Target    (Intercept) 0.1461   0.3823
 Subject   (Intercept) 0.0000   0.0000
 Residual              0.8189   0.9049
Number of obs: 361, groups:  Target, 48; Subject, 10

Fixed effects:
                                           Estimate Std. Error t value
(Intercept)                                 0.04535    0.08866   0.511
PrimeConditionUnrelated                    -0.08614    0.09650  -0.893
zPrimeRecogRT_trim                          0.04364    0.07984   0.547
PrimeConditionUnrelated:zPrimeRecogRT_trim  0.16928    0.10404   1.627

Correlation of Fixed Effects:
            (Intr) PrmCnU zPRRT_
PrmCndtnUnr -0.562
zPrmRcgRT_t  0.080 -0.068
PrmCU:PRRT_ -0.065  0.013 -0.771
```

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

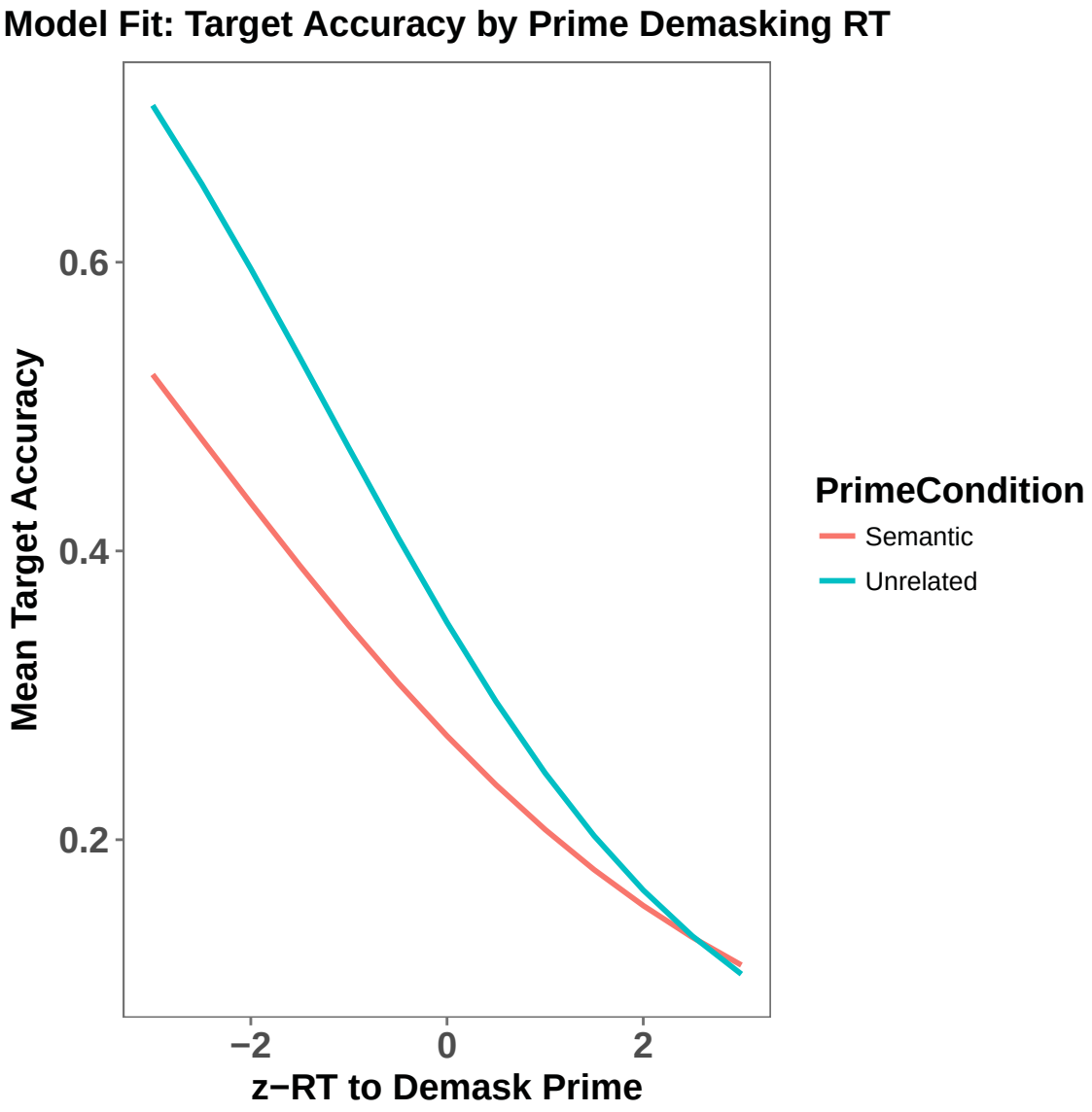
```
Analysis of Deviance Table (Type II Wald chisquare tests)

Response: zTargetRT_trim
                                  Chisq Df Pr(>Chisq)
PrimeCondition                   0.8354  1   0.360707
zPrimeRecogRT_trim               7.9988  1   0.004681 **
PrimeCondition:zPrimeRecogRT_trim 2.6474  1   0.103717
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
>
```

10 Plotting Model Fits

10.1 Model 1

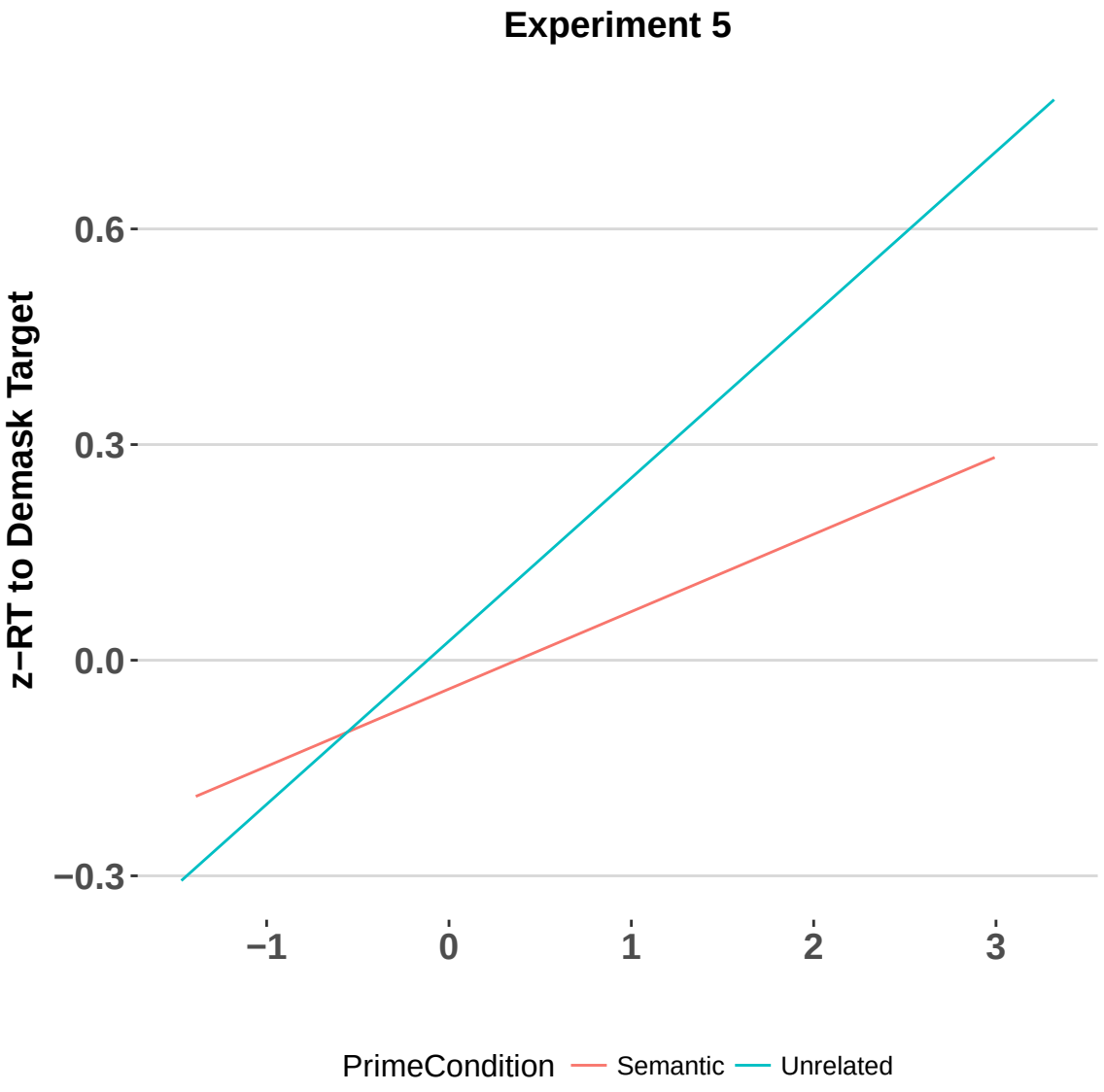
```
> fixed.frame ←
+   data.frame(expand.grid(PrimeCondition = c("Semantic", "Unrelated"),
+     zPrimeRecogRT_trim = seq(-3,3,0.5)))%>%
+   mutate(pred = predict(RTprime_acc_model, newdata = ., re.form = NA))
> fixed.frame$odds = exp(fixed.frame$pred)
> fixed.frame$prob = fixed.frame$odds/(1+fixed.frame$odds)
> fixed.frame %>%
+   ggplot(aes(x = zPrimeRecogRT_trim, y = prob,
+              group = PrimeCondition, color = PrimeCondition )) +
+     geom_line(size = 1) +
+     #ylim(0.10,0.40)+
+     xlab("z-RT to Demask Prime") + ylab ("Mean Target Accuracy")+
+   ggtitle("Model Fit: Target Accuracy by Prime Demasking RT")+
+ theme_few() +
+     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))
```



10.2 RAW RT Model

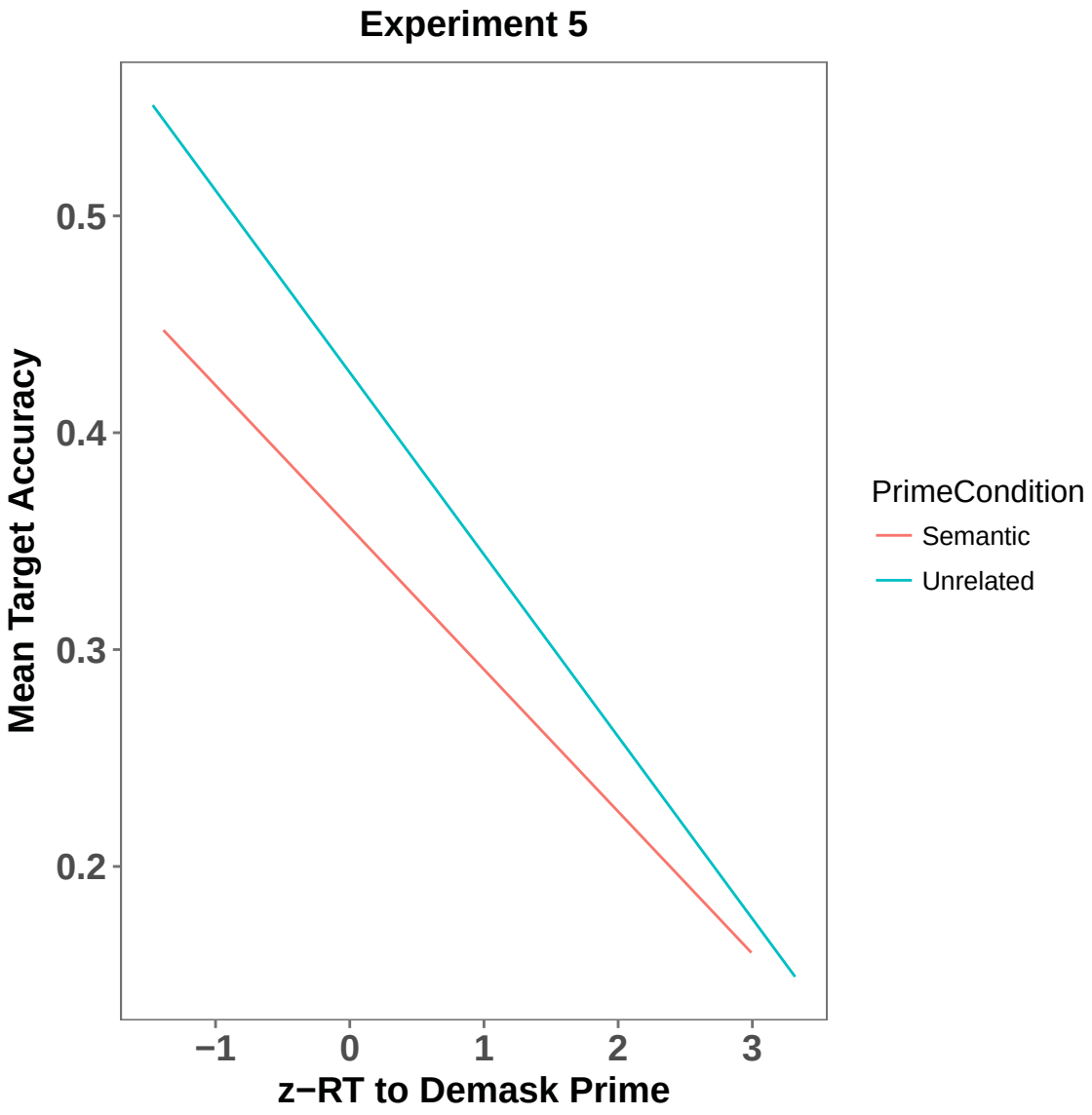
```
> mainplot = SemanticCuedRecall_final_z %>%
+  ggplot(aes(x =zPrimeRecogRT_trim , y = zTargetRecogRT_trim,
+             group = PrimeCondition, color = PrimeCondition)) +
+   geom_smooth(method = "lm", se = FALSE,  size = 0.5)+
+     xlab("") + ylab ("z-RT to Demask Target")+
+   ggtitle("Experiment 5")+
+ theme_hc() +
+     theme(axis.text = element_text(face = "bold", size = rel(1.2)),
```

```
+            axis.title = element_text(face = "bold", size = rel(1.2)),
+            # legend.title = element_blank(),
+            # legend.text = element_blank(),
+            # legend.key = element_blank(),
+      strip.text.x = element_text(face = "bold", size = rel(1.4)),
+      plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
> mainplot
>
```



10.3 RAW ACC Model

```
> SemanticCuedRecall_final_z$TargetAccuracy = as.numeric(as.character(SemanticCuedRecall
> SemanticCuedRecall_final_z1 = SemanticCuedRecall_final_z
> mainplot = SemanticCuedRecall_final_z1 %>%
+  ggplot(aes(x =zPrimeRecogRT_trim , y = TargetAccuracy,
+             group = PrimeCondition, color = PrimeCondition)) +
+   geom_smooth(method = "glm", se = FALSE,  size = 0.5)+
+ #   guides(color = FALSE)+
+     xlab("z-RT to Demask Prime") + ylab ("Mean Target Accuracy")+
+   ggtitle("Experiment 5")+
+ theme_few() +
+     theme(axis.text = element_text(face = "bold", size = rel(1.2)),
+           axis.title = element_text(face = "bold", size = rel(1.2)),
+          # legend.title = element_blank(),
+         #   legend.text = element_blank(),
+          # legend.key = element_blank(),
+     strip.text.x = element_text(face = "bold", size = rel(1.4)),
+     plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
> mainplot
>
```


Experiment 5
Mean Target Accuracy
0.5
0.4
0.3
0.2
−1
0
1
2
3
z−RT to Demask Prime
PrimeCondition
Semantic
Unrelated