TOT Cued Recall Analysis

Abhilasha Kumar

December 13, 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_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
```

2 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: -0.9

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.5181 -0.6902 -0.0704  0.6484  2.8557

Random effects:
 Groups   Name        Variance Std.Dev.
 Subject  (Intercept) 0.00000  0.000
 Residual             0.05152  0.227
Number of obs: 223, groups:  Subject, 30

Fixed effects:
                                                    Estimate Std. Error
(Intercept)                                          0.68264    0.04215
PrimeConditionUnrelated                             -0.05016    0.05911
CuedRecallAcc1                                      -0.08857    0.05961
TargetAccuracy1                                     -0.33081    0.05961
PrimeConditionUnrelated:CuedRecallAcc1               0.05989    0.08472
PrimeConditionUnrelated:TargetAccuracy1              0.10668    0.08472
CuedRecallAcc1:TargetAccuracy1                       0.20976    0.08507
PrimeConditionUnrelated:CuedRecallAcc1:TargetAccuracy1 -0.08154    0.12178
                                                    t value
(Intercept)                                          16.197
PrimeConditionUnrelated                              -0.849
CuedRecallAcc1                                       -1.486
```

```
TargetAccuracy1                                          -5.550
PrimeConditionUnrelated:CuedRecallAcc1                    0.707
PrimeConditionUnrelated:TargetAccuracy1                   1.259
CuedRecallAcc1:TargetAccuracy1                            2.466
PrimeConditionUnrelated:CuedRecallAcc1:TargetAccuracy1   -0.670

Correlation of Fixed Effects:
            (Intr) PrmCnU CdRcA1 TrgtA1 PrCU:CRA1 PCU:TA CRA1:T
PrmCndtnUnr -0.713
CudRcllAcc1 -0.707  0.504
TrgtAccrcy1 -0.707  0.504  0.500
PrmCnU:CRA1  0.497 -0.698 -0.704 -0.352
PrmCndU:TA1  0.497 -0.698 -0.352 -0.704  0.487
CdRclA1:TA1  0.495 -0.353 -0.701 -0.701  0.493     0.493
PCU:CRA1:TA -0.346  0.485  0.489  0.489 -0.696    -0.696 -0.699
```

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

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

Response: prop
                                          Chisq Df Pr(>Chisq)
PrimeCondition                           0.1424  1   0.705948
CuedRecallAcc                            0.5846  1   0.444527
TargetAccuracy                          41.5723  1  1.136e-10 ***
PrimeCondition:CuedRecallAcc             0.1126  1   0.737171
PrimeCondition:TargetAccuracy            1.2197  1   0.269426
CuedRecallAcc:TargetAccuracy             7.7958  1   0.005237 **
PrimeCondition:CuedRecallAcc:TargetAccuracy  0.4483  1   0.503128
---
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
```

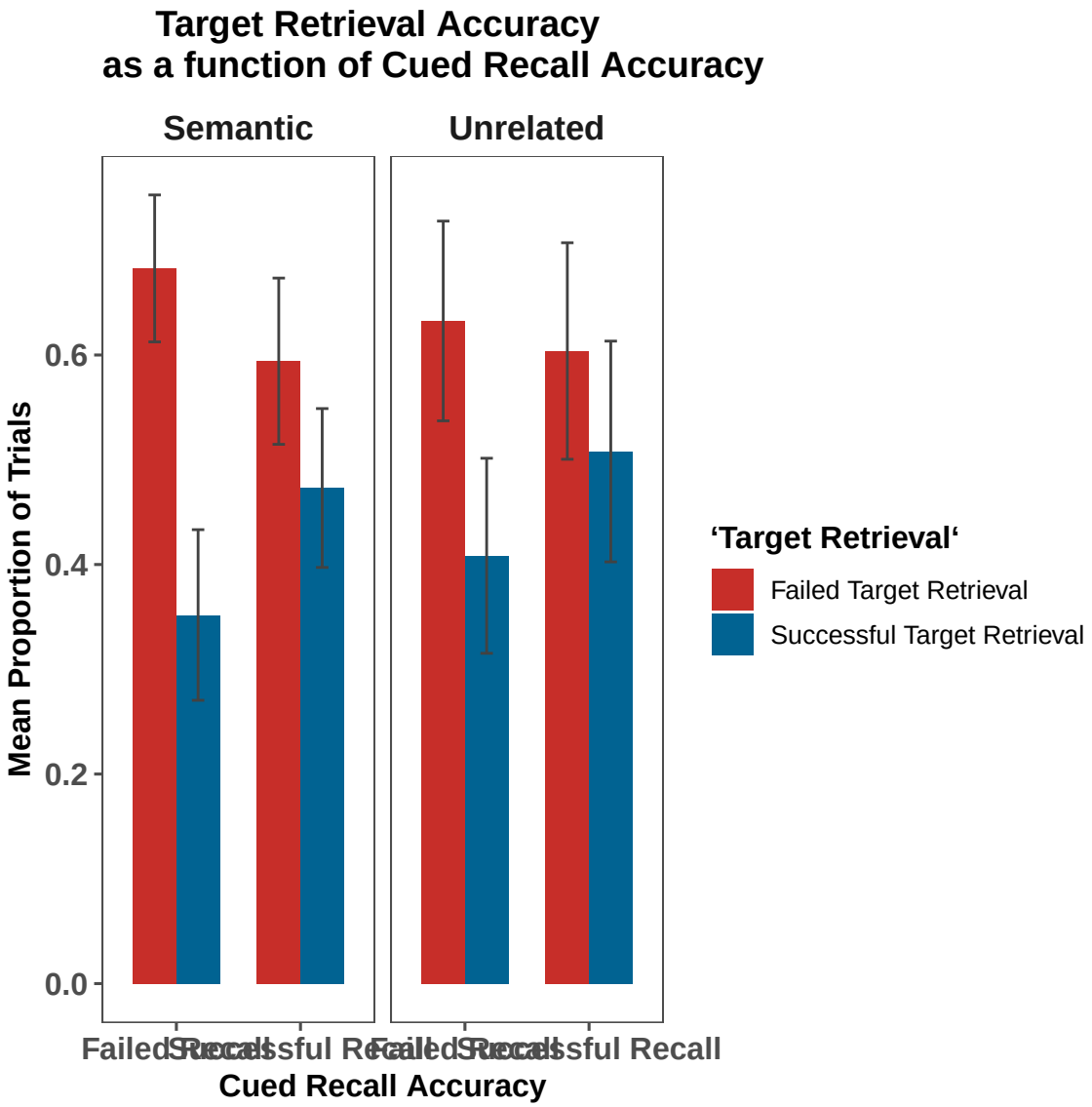
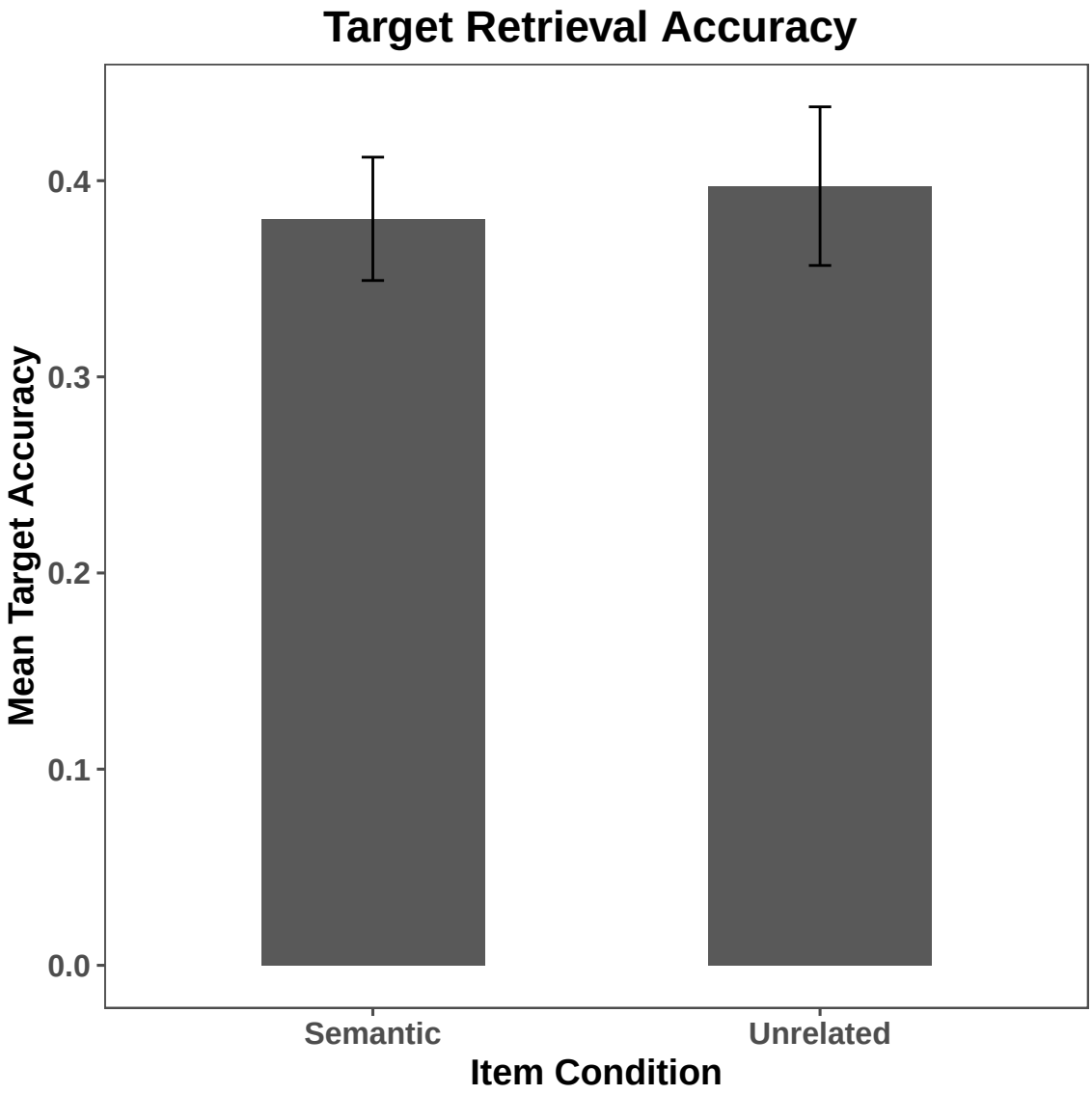


Figure Overall Target Accuracy

```
> 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)
```

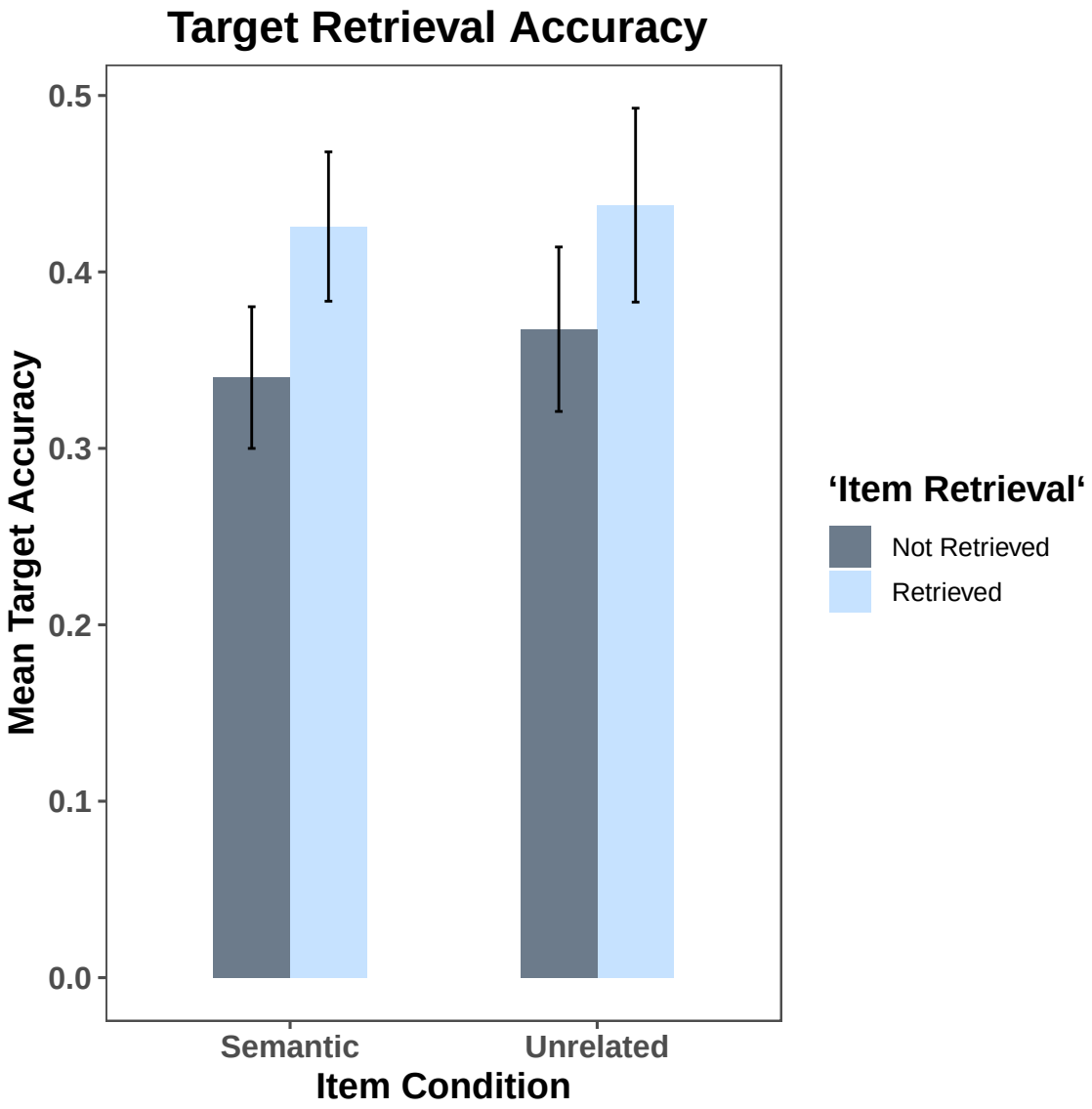
```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
```

```
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
```

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 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)))
>
```

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  4.093  0.1411
```

```
Error: Subject:PrimeCondition
               Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition  1 0.0047 0.00466   0.083  0.776
Residuals      29 1.6312 0.05625

Error: Subject:CuedRecallAcc
              Df Sum Sq Mean Sq F value Pr(>F)
CuedRecallAcc  1 0.1498 0.14982   6.098 0.0197 *
Residuals     29 0.7126 0.02457
---
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 0.0067 0.00670   0.184  0.671
Residuals                    29 1.0533 0.03632
```

6 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
  1821.0   1847.4   -905.5   1811.0     1435

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.6028 -0.7539 -0.5426  0.9448  3.3615

Random effects:
 Groups  Name        Variance Std.Dev.
 Subject (Intercept) 0.5012   0.708
Number of obs: 1440, groups:  Subject, 30
```

```
Fixed effects:
                                          Estimate Std. Error z value Pr(>|z|)
(Intercept)                               -0.7926     0.1744  -4.544 5.53e-06 ***
PrimeConditionUnrelated                    0.1405     0.1599   0.879  0.37952
CuedRecallAcc1                             0.5212     0.1729   3.014  0.00258 **
PrimeConditionUnrelated:CuedRecallAcc1    -0.1294     0.2302  -0.562  0.57394
---
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 -0.464
CudRcllAcc1 -0.476  0.467
PrmCnU:CRA1  0.322 -0.698 -0.664
```

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

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

Response: TargetAccuracy
                            Chisq Df Pr(>Chisq)
PrimeCondition             0.4613  1  0.4970261
CuedRecallAcc             12.4828  1  0.0004107 ***
PrimeCondition:CuedRecallAcc  0.3161  1  0.5739432
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

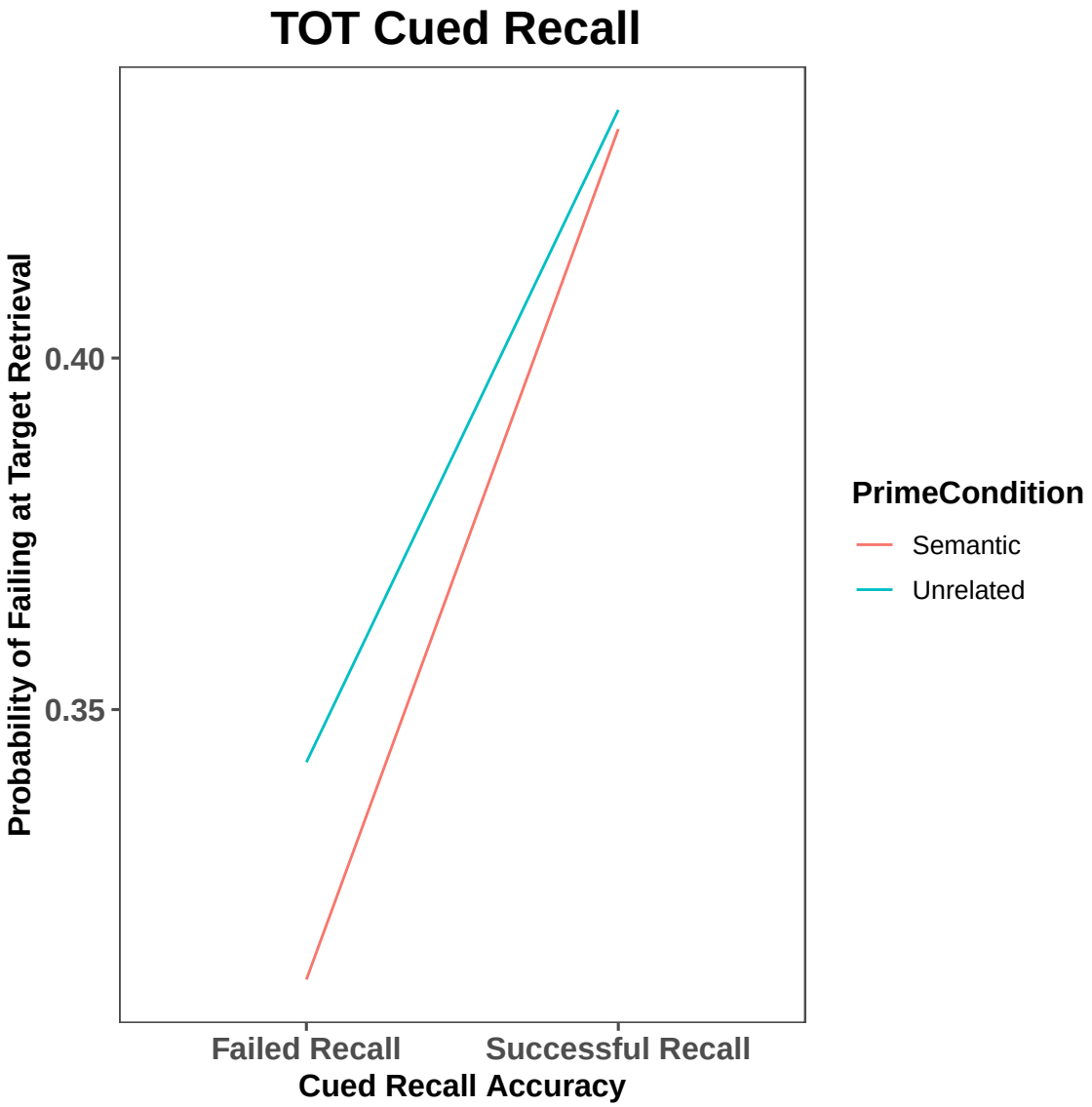
```
> options(contrasts = c("contr.sum","contr.poly"))
> anova(SemanticCuedRecall_hlm)
```

```
Analysis of Variance Table
                           Df  Sum Sq Mean Sq F value
PrimeCondition              1  0.4319  0.4319  0.4319
CuedRecallAcc               1 12.6875 12.6875 12.6875
PrimeCondition:CuedRecallAcc  1  0.3201  0.3201  0.3201
```

6.0.1 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)))
>
```



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

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)                          -0.63948    0.26153  -2.445  0.01448 *
PrimeCondition1                      -0.10008    0.06565  -1.524  0.12739
zPrimeRecogRT_trim                   -0.23242    0.07141  -3.255  0.00113 **
PrimeCondition1:zPrimeRecogRT_trim    0.07666    0.07240   1.059  0.28967
---
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 -0.003
zPrmRcgRT_t  0.018  0.053
PrmC1:PRRT_  0.009  0.047  0.013
```

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

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

Response: TargetAccuracy
                                  Chisq Df Pr(>Chisq)
PrimeCondition                   2.4850  1   0.114934
zPrimeRecogRT_trim              10.6834  1   0.001081 **
PrimeCondition:zPrimeRecogRT_trim  1.1212  1   0.289670
---
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 10.9035 10.9035 10.9035
PrimeCondition:zPrimeRecogRT_trim  1  1.1446  1.1446  1.1446
```

```
> 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 10.9035 10.9035 10.9035
PrimeCondition:zPrimeRecogRT_trim  1  1.1446  1.1446  1.1446
```

```
> 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: 3775.6

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)                      0.006495   0.064818   0.100
PrimeCondition1                 -0.012089   0.024178  -0.500
zPrimeRecogRT_trim               0.091564   0.025551   3.584
PrimeCondition1:zPrimeRecogRT_trim -0.050422   0.025477  -1.979

Correlation of Fixed Effects:
            (Intr) PrmCn1 zPRRT_
PrimeCndtn1 -0.001
zPrmRcgRT_t  0.004  0.046
PrmC1:PRRT_  0.019  0.012  0.051
```

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

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

Response: zTargetRecogRT_trim
                                 Chisq Df Pr(>Chisq)
PrimeCondition                  0.2268  1  0.6339015
zPrimeRecogRT_trim             13.6104  1  0.0002249 ***
PrimeCondition:zPrimeRecogRT_trim  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
PrimeCondition                       1  0.3324  0.3324  0.4137
zPrimeRecogRT_trim                   1 10.9341 10.9341 13.6104
PrimeCondition:zPrimeRecogRT_trim    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: 3169.7

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)                        0.02403    0.06557   0.366
PrimeCondition1                    0.04571    0.02647   1.726
zPrimeRecogRT_trim                 0.07443    0.02766   2.691
PrimeCondition1:zPrimeRecogRT_trim -0.00474    0.02742  -0.173

Correlation of Fixed Effects:
            (Intr) PrmCn1 zPRRT_
PrimeCndtn1 0.000
zPrmRcgRT_t 0.004  0.045
PrmC1:PRRT_ 0.019  0.010  0.059
```

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

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

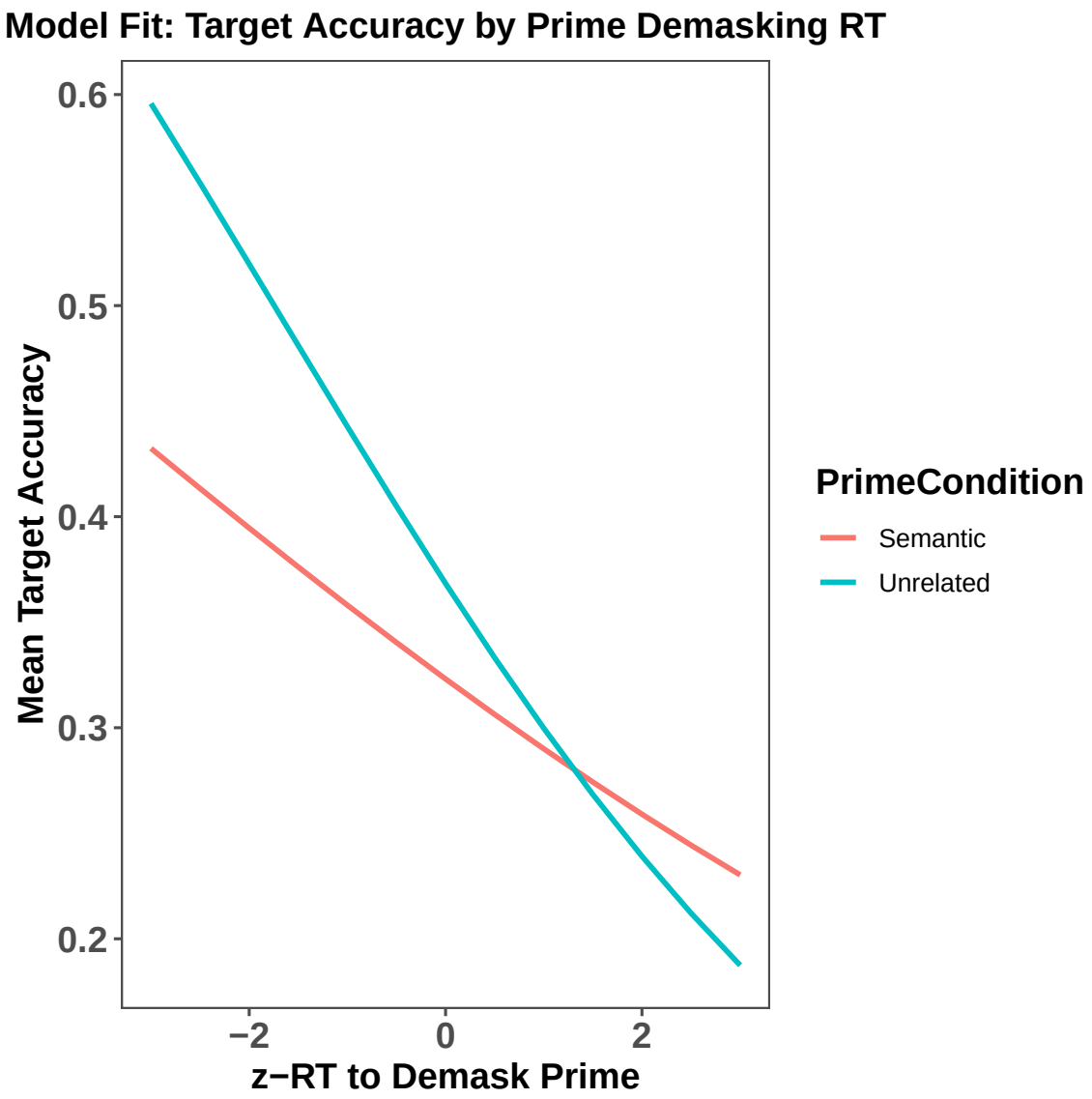
```
Response: zTargetRT_trim
                                  Chisq Df Pr(>Chisq)
PrimeCondition                   2.9867  1   0.083950 .
zPrimeRecogRT_trim               7.3214  1   0.006814 **
PrimeCondition:zPrimeRecogRT_trim 0.0299  1   0.862752
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01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

```
>
```

11 Plotting Model Fits

11.1 Model 1

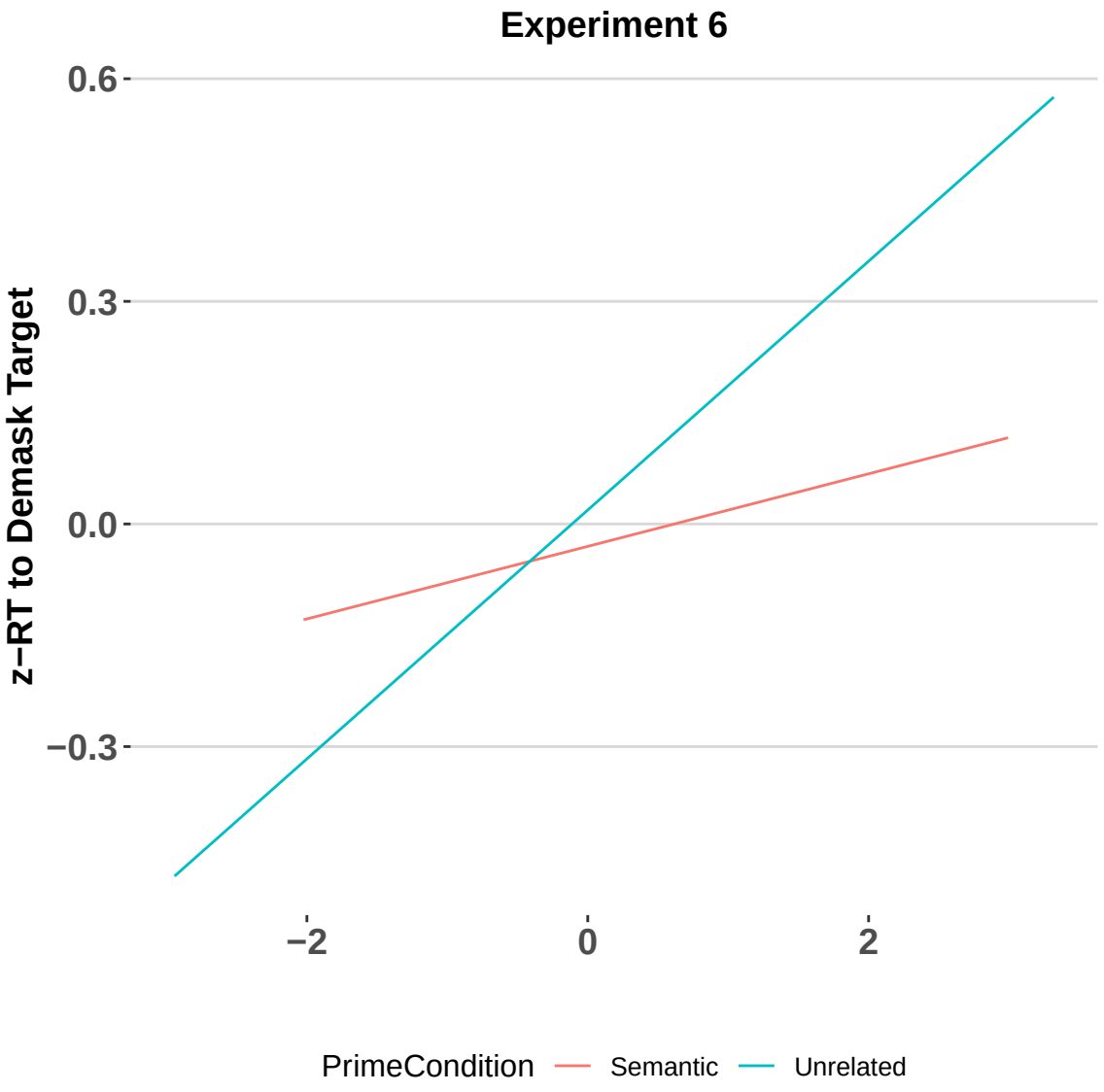
```
> fixed.frame ←
+   data.frame(expand.grid(PrimeCondition = c("Semantic", "Unrelated"),
+     zPrimeRecogRT_trim = seq(-3,3,0.5)))%>%
+   mutate(pred = predict(RTprime_acc_model, newdata = ., re.form = NA))
> fixed.frame$odds = exp(fixed.frame$pred)
> fixed.frame$prob = fixed.frame$odds/(1+fixed.frame$odds)
> 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))
```



11.2 RAW RT Model

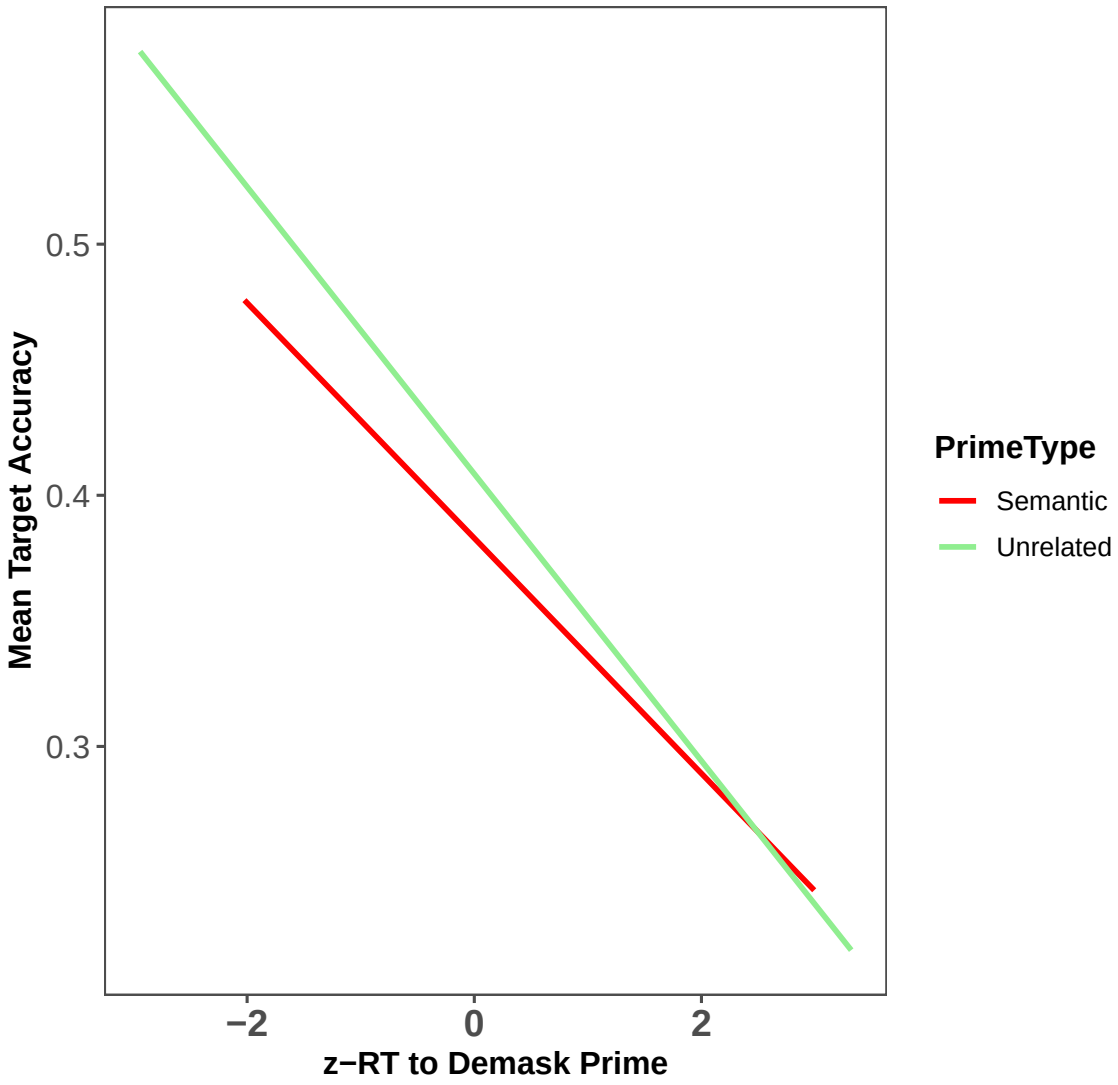
```
> mainplot = SemanticCuedRecall_final_z %>%
+  ggplot(aes(x =zPrimeRecogRT_trim , y = zTargetRecogRT_trim,
+             group = PrimeCondition, color = PrimeCondition)) +
+   geom_smooth(method = "lm", se = FALSE,  size = 0.5)+
+     xlab("") + ylab ("z-RT to Demask Target")+
+   ggtitle("Experiment 6")+
+ 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
>
```



11.3 RAW ACC Model

```
> SemanticCuedRecall_final_z$TargetAccuracy = as.numeric(as.character(SemanticCuedRecall
> SemanticCuedRecall_final_z1 = SemanticCuedRecall_final_z
> SemanticCuedRecall_final_z1$PrimeType = SemanticCuedRecall_final_z1$PrimeCondition
> mainplot = SemanticCuedRecall_final_z1 %>%
+  ggplot(aes(x =zPrimeRecogRT_trim , y = TargetAccuracy,
+             group = PrimeType, color = PrimeType)) +
+   geom_smooth(method = "glm",  se = FALSE)+
+ #   guides(color = FALSE)+
+     xlab("z-RT to Demask Prime") + ylab ("Mean Target Accuracy")+
+   ggtitle("")+
+ theme_few() +
+   scale_color_manual(values = c( "red",
+                           "lightgreen"))+
+    theme(axis.text = element_text(size = rel(1)),
+          axis.title = element_text(face = "bold", size = rel(1)),
+          legend.title = element_text(face = "bold", size = rel(1)),
+         plot.title = element_text(hjust = .5, size = rel(1)),
+         axis.text.x = element_text(face = "bold", size = rel(1.2)))
> mainplot
>
```


Mean Target Accuracy
0.5
0.4
0.3
−2
0
2
z−RT to Demask Prime
PrimeType
Semantic
Unrelated