Memory Demasking Analysis

Abhilasha Kumar

June 25, 2018

1 Reading the Data File

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

```
> mem = read.csv("Compiled_MemoryDemasking.csv", header = TRUE,
+                sep = ",", stringsAsFactors = TRUE)
> head(mem)
```

```
             ExperimentName Subject Session Procedure.Block. Trial
1 MemoryDemasking_Immediate       1       1               S1    77
2 MemoryDemasking_Immediate       1       1               S1    87
3 MemoryDemasking_Immediate       1       1               S1   119
4 MemoryDemasking_Immediate       1       1               S1    65
5 MemoryDemasking_Immediate       1       1               S1   113
6 MemoryDemasking_Immediate       1       1               S1    93
  CorrectAnswer    Cue  Prime PrimeCondition TargetResponse.RESP
1         above  PLANE  above      Identical        above{SPACE}
2         alive   BEAR  alive      Identical        alive{SPACE}
3         angel    SKY  angel      Identical        angel{SPACE}
4         anger PRISON  anger      Identical        anger{SPACE}
5        animal    PET animal      Identical       animal{SPACE}
6          baby  GRASP   baby      Identical         baby{SPACE}
  TargetResponse.RT Target TargetAccuracyImmediate RTImmediate
1              1924  above                       1        1455
2              1578  alive                       1        1443
3              1211  angel                       1        1412
4              1323  anger                       1        1345
5              1370 animal                       1         812
6              1929   baby                       1         561
  TargetAccuracyDelayed RTDelayed
1                     1      1255
2                     1      1550
3                     1      1652
4                     0      1191
5                     1       982
6                     1      1324
```

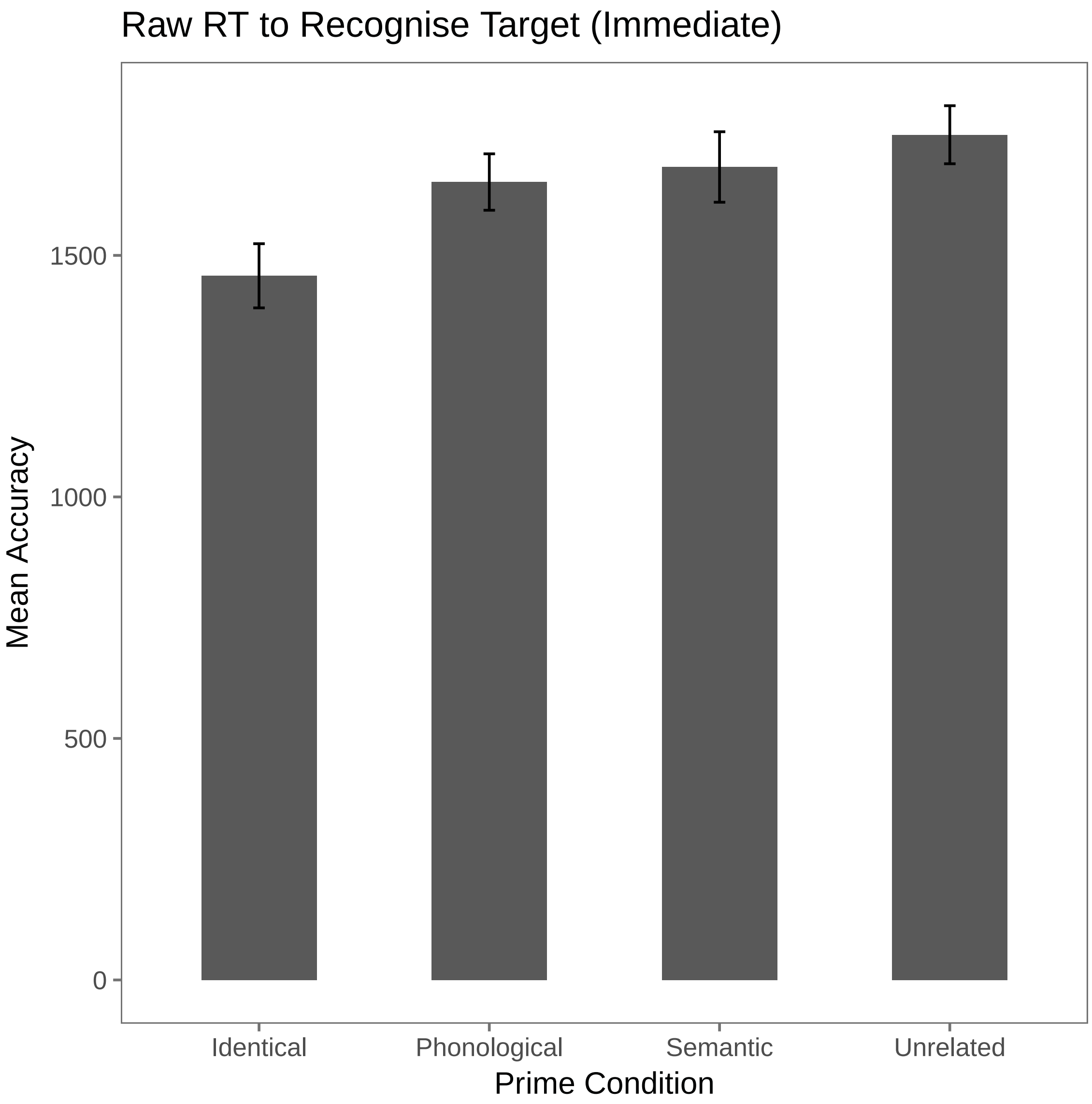
2 Raw RT as a function of Prime

2.1 Raw RT Figure

First, we see whether the RT to recognize the target word varies as a function of the prime condition. Note that we only want to consider trials in which the target word was correctly identified by the participant. We first do this separately for Immediate and Delayed.

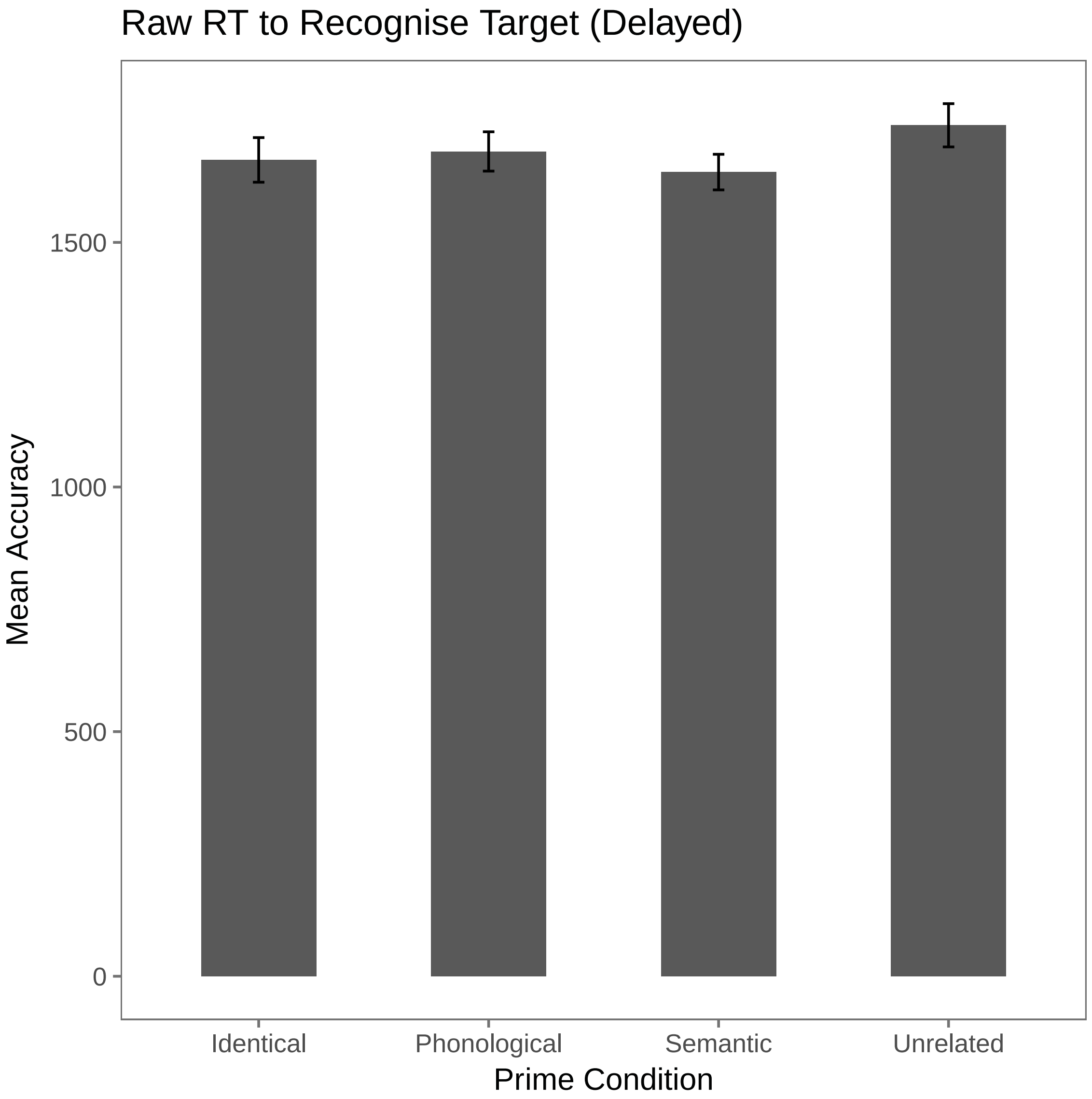
2.1.1 Immediate

```
> library(dplyr)
> mem_correct_imm = mem %>% filter(TargetAccuracyImmediate == "1")
> RT_rmisc = Rmisc::summarySE(mem_correct_imm,
+                             measurevar = "RTImmediate",
+                             groupvars = "PrimeCondition")
> library(ggplot2)
> library(ggthemes)
> library(dplyr)
> RT_rmisc %>%
+   ggplot(aes(x = PrimeCondition, y = RTImmediate))+
+   geom_bar(stat = "identity", position = "dodge", width = 0.5)+
+   geom_errorbar(aes(ymin = RTImmediate - se, ymax = RTImmediate + se),
+                 width=.05, position=position_dodge(.5)) +
+   theme_few()+
+   scale_fill_solarized()+
+   xlab("Prime Condition") + ylab("Mean Accuracy") +
+   ggtitle("Raw RT to Recognise Target (Immediate)")
```



2.1.2 Delayed

```
> mem_correct_del = mem %>% filter(TargetAccuracyDelayed == "1")
> RT_rmisc = Rmisc::summarySE(mem_correct_del,
+                             measurevar = "RTDelayed",
+                             groupvars = "PrimeCondition")
> library(ggplot2)
> library(ggthemes)
> library(dplyr)
> RT_rmisc %>%
+   ggplot(aes(x = PrimeCondition, y = RTDelayed))+
+   geom_bar(stat = "identity", position = "dodge", width = 0.5)+
+   geom_errorbar(aes(ymin = RTDelayed - se, ymax = RTDelayed + se),
+                 width=.05, position=position_dodge(.5)) +
+   theme_few()+
+   scale_fill_solarized()+
+   xlab("Prime Condition") + ylab("Mean Accuracy") +
+   ggtitle("Raw RT to Recognise Target (Delayed)")
```



2.2 Raw RT ANOVA

2.2.1 Immediate

```
> rt_prime_subject_imm = group_by(mem_correct_imm, Subject, PrimeCondition) %>%
+   summarise_at(vars(RTImmediate), mean)
> rt_prime_subject_imm$Subject = as.factor(rt_prime_subject_imm$Subject)
> rt_aov_imm = aov(data = rt_prime_subject_imm, RTImmediate ∼ PrimeCondition +
+                      Error(Subject/PrimeCondition))
> summary(rt_aov_imm)
```

```
Error: Subject
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals 10 7593855  759385

Error: Subject:PrimeCondition
               Df Sum Sq Mean Sq F value  Pr(>F)
PrimeCondition  3 511392  170464   7.037 0.00101 **
Residuals      30 726703   24223
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

2.2.2 Delayed

```
> rt_prime_subject_del = group_by(mem_correct_del, Subject, PrimeCondition) %>%
+   summarise_at(vars(RTDelayed), mean)
> rt_prime_subject_del$Subject = as.factor(rt_prime_subject_del$Subject)
> rt_aov_del = aov(data = rt_prime_subject_del, RTDelayed ∼ PrimeCondition +
+                      Error(Subject/PrimeCondition))
> summary(rt_aov_del)
```

```
Error: Subject
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals  9 1774739  197193

Error: Subject:PrimeCondition
               Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition  3  43581   14527    0.74  0.537
Residuals      27 530039   19631
```

3 z-scoring RTs

3.1 Trimming Error Trials and Below 250, Above 7s

```
> memory_imm = mem_correct_imm %>% filter(RTImmediate ≥ 250 & RTImmediate ≤7000)
> memory_del = mem_correct_del %>% filter(RTDelayed ≥250 & RTDelayed ≤ 7000)
```

3.2 Making z-scores

3.2.1 For immediate RTs

```
>  library(dplyr)
>  ## aggregate per subject all IVs and DVs
>  memory_mean = group_by(memory_imm, Subject) %>%
+    summarise_at(vars(RTImmediate), mean)
>  colnames(memory_mean) = c("Subject", "meanRTimm")
>  memory_sd = group_by(memory_imm, Subject) %>%
+    summarise_at(vars(RTImmediate), sd)
>  colnames(memory_sd) = c("Subject", "sdRTimm")
>  memory_agg = merge(memory_mean, memory_sd, by = "Subject")
>  ## merge aggregate info with long data
>  memory_imm = merge(memory_imm, memory_agg, by = "Subject", all.x = T)
>  ## person and grand-mean centered scores using original and aggregate
>  library(dplyr)
>  memory_imm = memory_imm %>% mutate(zRT_imm =
+                                        (RTImmediate - meanRTimm)/sdRTimm)
>  ## checking: subject level means should be zero
>
>  sub_mem = group_by(memory_imm, Subject) %>%
+    summarise_at(vars(zRT_imm), mean)
```

3.2.2 For delayed RTs

```
>  library(dplyr)
>  ## aggregate per subject all IVs and DVs
>  memory_mean = group_by(memory_del, Subject) %>%
+    summarise_at(vars(RTDelayed), mean)
>  colnames(memory_mean) = c("Subject", "meanRTdel")
>  memory_sd = group_by(memory_del, Subject) %>%
+    summarise_at(vars(RTDelayed), sd)
>  colnames(memory_sd) = c("Subject", "sdRTdel")
>  memory_agg = merge(memory_mean, memory_sd, by = "Subject")
>  ## merge aggregate info with long data
>  memory_del = merge(memory_del, memory_agg, by = "Subject", all.x = T)
>  ## person and grand-mean centered scores using original and aggregate
>  library(dplyr)
>  memory_del = memory_del %>% mutate(zRT_del =
+                                        (RTDelayed - meanRTdel)/sdRTdel)
>  ## checking: subject level means should be zero
>
>  sub_mem = group_by(memory_del, Subject) %>%
+    summarise_at(vars(zRT_del), mean)
```

3.3 Trimming and repeat z-scoring

```
> memory_imm_trim = memory_imm %>% filter(zRT_imm ≥ -3 & zRT_imm ≤ 3)
> memory_del_trim = memory_del %>% filter(zRT_del ≥ -3 & zRT_del ≤ 3)
```

3.3.1 For immediate RTs

```
>  library(dplyr)
>  ## aggregate per subject all IVs and DVs
>  memory_mean = group_by(memory_imm_trim, Subject) %>%
+    summarise_at(vars(RTImmediate), mean)
>  colnames(memory_mean) = c("Subject", "meanRTimm_final")
>  memory_sd = group_by(memory_imm_trim, Subject) %>%
+    summarise_at(vars(RTImmediate), sd)
>  colnames(memory_sd) = c("Subject", "sdRTimm_final")
>  memory_agg = merge(memory_mean, memory_sd, by = "Subject")
>  ## merge aggregate info with long data
>  memory_imm_trim = merge(memory_imm_trim,
+                             memory_agg, by = "Subject", all.x = T)
>  ## person and grand-mean centered scores using original and aggregate
>  library(dplyr)
>  memory_imm_trim = memory_imm_trim %>% mutate(zRT_imm_final =
+                            (RTImmediate - meanRTimm_final)/sdRTimm_final)
>  ## checking: subject level means should be zero
>
>  sub_mem = group_by(memory_imm_trim, Subject) %>%
+    summarise_at(vars(zRT_imm_final), mean)
```

3.3.2 For delayed RTs

```
>  library(dplyr)
>  ## aggregate per subject all IVs and DVs
>  memory_mean = group_by(memory_del_trim, Subject) %>%
+    summarise_at(vars(RTDelayed), mean)
>  colnames(memory_mean) = c("Subject", "meanRTdel_final")
>  memory_sd = group_by(memory_del_trim, Subject) %>%
+    summarise_at(vars(RTDelayed), sd)
>  colnames(memory_sd) = c("Subject", "sdRTdel_final")
>  memory_agg = merge(memory_mean, memory_sd, by = "Subject")
>  ## merge aggregate info with long data
>  memory_del_trim = merge(memory_del_trim,
+                             memory_agg, by = "Subject", all.x = T)
>  ## person and grand-mean centered scores using original and aggregate
>  library(dplyr)
>  memory_del_trim = memory_del_trim %>% mutate(zRT_del_final =
+                               (RTDelayed - meanRTdel_final)/sdRTdel_final)
```

```
>  ## checking: subject level means should be zero
>
>  sub_mem = group_by(memory_del_trim, Subject) %>%
+    summarise_at(vars(zRT_del_final), mean)
```

4 zRT ANOVAs: Immediate and Delayed

```
>  z_RT_imm = group_by(memory_imm_trim, Subject, PrimeCondition) %>%
+  summarise_at(vars(zRT_imm_final, RTImmediate), mean)
>  z_RT_del = group_by(memory_del_trim, Subject, PrimeCondition) %>%
+  summarise_at(vars(zRT_del_final, RTDelayed), mean)
>  z_RT_imm$PrimeCondition = as.factor(as.character(z_RT_imm$PrimeCondition))
>  z_RT_imm$Subject = as.factor(as.character(z_RT_imm$Subject))
>  z_RT_del$PrimeCondition = as.factor(as.character(z_RT_del$PrimeCondition))
>  z_RT_del$Subject = as.factor(as.character(z_RT_del$Subject))
>  z_RT_imm$zRT_imm_final = as.numeric(as.character(z_RT_imm$zRT_imm_final))
>  z_RT_imm$RTImmediate = as.numeric(as.character(z_RT_imm$RTImmediate))
>  z_RT_del$zRT_del_final = as.numeric(as.character(z_RT_del$zRT_del_final))
>  z_RT_del$RTDelayed = as.numeric(as.character(z_RT_del$RTDelayed))
>   ## IMMEDIATE
>
>  z_immediate_aov = aov(data = z_RT_imm,
+                                      RTImmediate ~ PrimeCondition +
+                                  Error(Subject/PrimeCondition))
>  summary(z_immediate_aov)
```

```
Error: Subject
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals 10 7164641  716464

Error: Subject:PrimeCondition
               Df Sum Sq Mean Sq F value   Pr(>F)
PrimeCondition  3 492898  164299   10.44 7.24e-05 ***
Residuals      30 471972   15732
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
>  options(contrasts = c('contr.sum', 'contr.poly'))
>  library(lsmeans)
>  library(multcomp)
>  imm_lsm = lsmeans::lsmeans(z_immediate_aov, c("PrimeCondition"))
>  prime_effect = cld(imm_lsm, alpha = 0.05,
+                 adjust = "tukey", details = TRUE)
>  library(knitr)
>  kable(subset(prime_effect$comparisons,prime_effect$comparisons$p.value < 0.05 ))
```

```
|   |contrast                  | estimate|       SE| df|  t.ratio|   p.value|
|:--|:-------------------------|--------:|--------:|--:|--------:|---------:|
|1  |Semantic - Identical      | 193.2342| 53.48303| 30| 3.613000| 0.0057050|
|2  |Phonological - Identical  | 218.1665| 53.48303| 30| 4.079172| 0.0016600|
|4  |Unrelated - Identical     | 284.5092| 53.48303| 30| 5.319616| 0.0000537|
```

```
>  ## DELAYED
>
>  z_delayed_aov = aov(data = z_RT_del,
+                                      RTDelayed ~ PrimeCondition +
+                                 Error(Subject/PrimeCondition))
>  summary(z_delayed_aov)
```

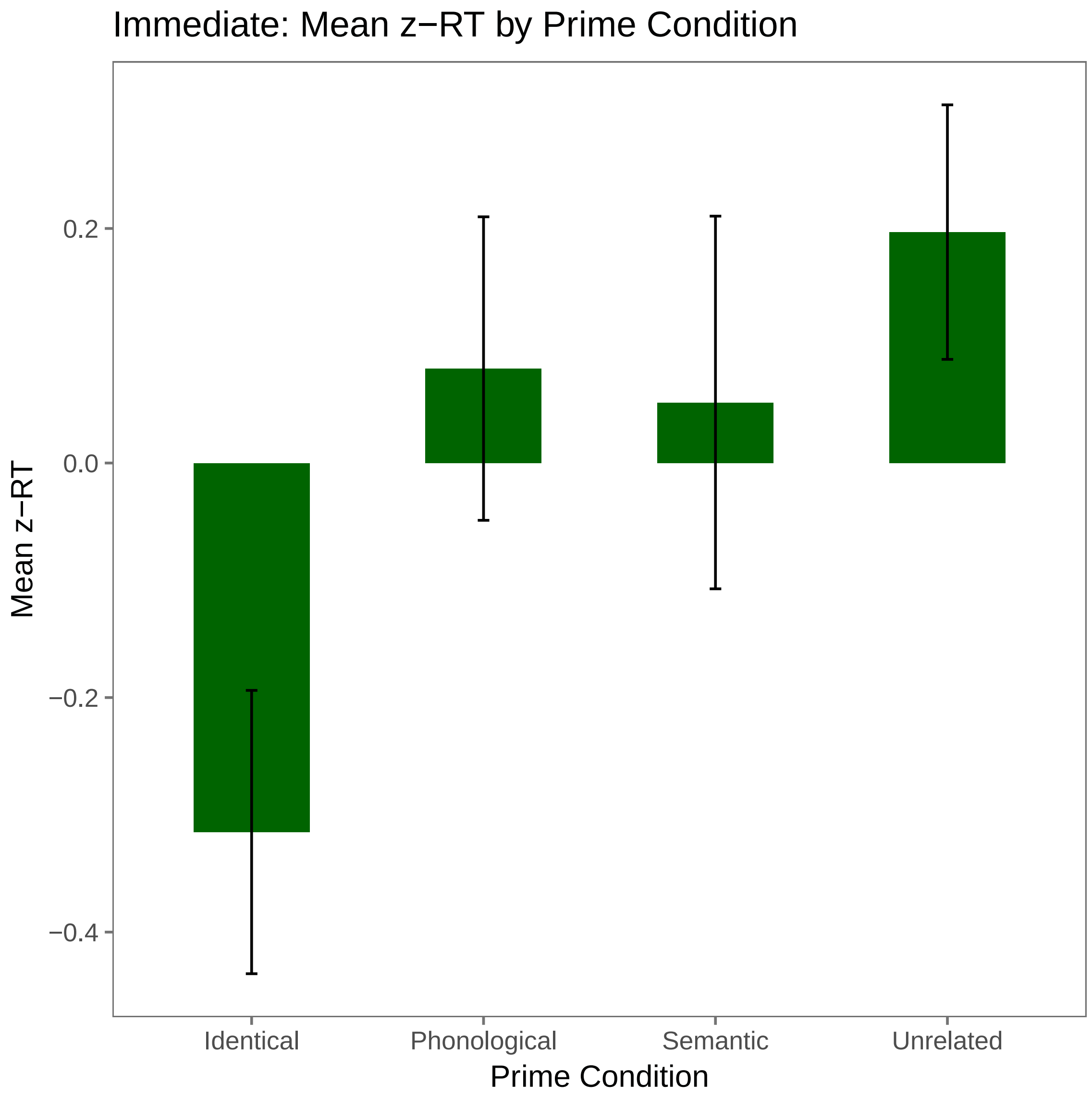
```
Error: Subject
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals  9 1499949  166661

Error: Subject:PrimeCondition
               Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition  3  20949    6983   0.551  0.652
Residuals      27 342392   12681
```

5 Figures for zRT

```
>  zRT_imm = Rmisc::summarySE(z_RT_imm,
+                         measurevar = "zRT_imm_final",
+                         groupvars = c("PrimeCondition"))
>  zRT_del = Rmisc::summarySE(z_RT_del,
+                         measurevar = "zRT_del_final",
+                         groupvars = c("PrimeCondition"))
```

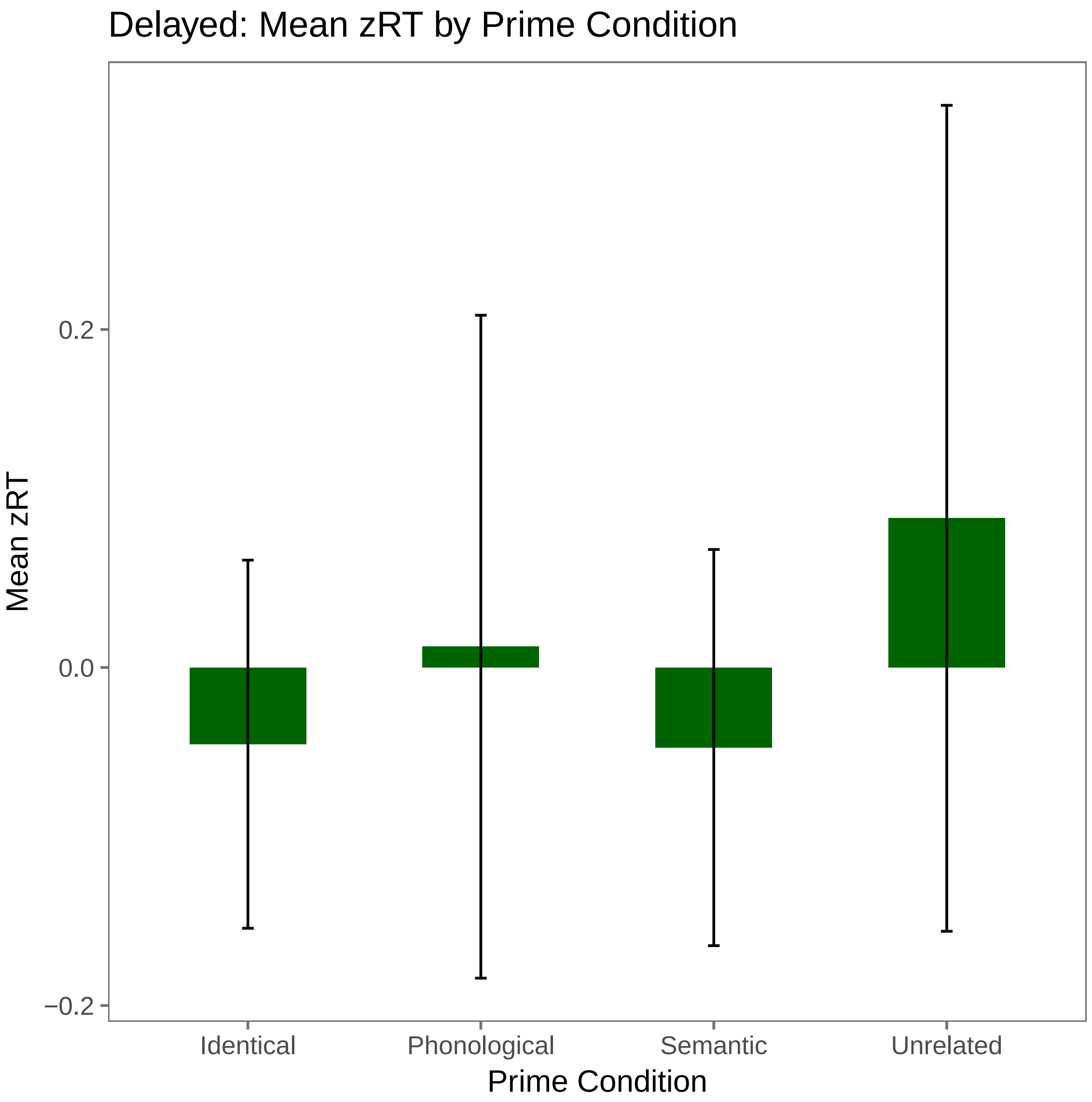
5.1 Immediate

```
>  library(ggplot2)
>  library(ggthemes)
>  zRT_imm %>%
+    ggplot(aes(x = PrimeCondition, y = zRT_imm_final))+
+    geom_bar(stat = "identity", position = "dodge", width = 0.5,
+             fill = "darkgreen")+
+    geom_errorbar(aes(ymin = zRT_imm_final - ci, ymax = zRT_imm_final + ci),
+                  width=.05, position=position_dodge(.5)) +
+    theme_few()+
+    xlab("Prime Condition") + ylab("Mean z-RT") +
+    ggtitle("Immediate: Mean z-RT by Prime Condition")
```



5.2 Delayed

```
> zRT_del %>%
+ ggplot(aes(x = PrimeCondition, y = zRT_del_final))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.5, fill = "darkgreen")+
+   geom_errorbar(aes(ymin = zRT_del_final - ci, ymax = zRT_del_final + ci),
+                 width=.05, position=position_dodge(.5)) +
+   theme_few()+
+   xlab("Prime Condition") + ylab("Mean zRT") +
+   ggtitle("Delayed: Mean zRT by Prime Condition")
```



6 Conditional Analyses

6.1 Merging Immediate and Delayed

```
> main_imm = memory_imm_trim[,c(1,5,6,9,13:16,22)]
> main_del = memory_del_trim[,c(1,5,6,9,13:16, 22)]
> main_combined = merge(main_imm, main_del,
+                       by =c("Subject", "Trial", "CorrectAnswer", "PrimeCondition",
+                             "TargetAccuracyImmediate",
+                             "RTImmediate", "TargetAccuracyDelayed","RTDelayed"))
```

```
> main_combined= main_combined %>% arrange(Subject, Trial)
```

6.2 Effect of Immediate on Delayed

```
> library(lme4)
> m1 = lmer(data = main_combined,
+            zRT_del_final ~ zRT_imm_final*PrimeCondition +
+              (1|Subject) + (1|CorrectAnswer))
> summary(m1)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: zRT_del_final ~ zRT_imm_final * PrimeCondition + (1 | Subject) +
    (1 | CorrectAnswer)
   Data: main_combined

REML criterion at convergence: 1582.4

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.8105 -0.5596 -0.0819  0.5017  3.5118

Random effects:
 Groups        Name        Variance Std.Dev.
 CorrectAnswer (Intercept) 0.04895  0.2212
 Subject       (Intercept) 0.00000  0.0000
 Residual                  0.90434  0.9510
Number of obs: 561, groups:  CorrectAnswer, 60; Subject, 10

Fixed effects:
                              Estimate Std. Error t value
(Intercept)                   -0.01906    0.05000  -0.381
zRT_imm_final                  0.10712    0.04360   2.457
PrimeCondition1               -0.01110    0.07203  -0.154
PrimeCondition2                0.03841    0.07041   0.546
PrimeCondition3               -0.06133    0.07084  -0.866
zRT_imm_final:PrimeCondition1 -0.01810    0.06796  -0.266
zRT_imm_final:PrimeCondition2  0.05076    0.07370   0.689
zRT_imm_final:PrimeCondition3 -0.03921    0.08123  -0.483

Correlation of Fixed Effects:
            (Intr) zRT_m_ PrmCn1 PrmCn2 PrmCn3 zRT__:PC1 zRT__:PC2
zRT_imm_fnl -0.035
PrimeCndtn1  0.010  0.169
PrimeCndtn2 -0.021 -0.021 -0.329
PrimeCndtn3 -0.012 -0.051 -0.333 -0.318
zRT_mm_:PC1  0.157 -0.166  0.168 -0.078 -0.064
zRT_mm_:PC2 -0.022 -0.012 -0.070 -0.067  0.055 -0.266
```

```
zRT_mm_:PC3 -0.035  0.143 -0.052  0.050 -0.089 -0.340     -0.381
```

```
> car::Anova(m1)
```

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

Response: zRT_del_final
                              Chisq Df Pr(>Chisq)  
zRT_imm_final                6.3470  1    0.01176 *
PrimeCondition               1.1072  3    0.77534  
zRT_imm_final:PrimeCondition 0.5864  3    0.89953  
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```