Repeated Lexical Retrieval: Experiment 3

Abhilasha Kumar

June 6, 2019

1 Comparing TOT Unrelated and TOT Semantic

```
> US = read.csv("US_TOT_Responses.csv", header = TRUE, sep = ",")
```

1.1 LME

```
> contrasts(US$PrimeCondition)= contr.treatment(2, base = 2)
> library(lme4)
> prime_lmer2 = glmer(data = US,
+                   TargetFirstResp_ACC ~ PrimeCondition +
+                     (1|Subject) + (1|Target.Trial.),
+                          family = "binomial",
+                          control=glmerControl(optimizer="bobyqa",
+             optCtrl=list(maxfun=100000)))
> summary(prime_lmer2)
```

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

     AIC      BIC   logLik deviance df.resid
  4032.1   4057.4  -2012.0   4024.1     4172

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.6099 -0.5307 -0.2859  0.5031  5.6758

Random effects:
 Groups        Name        Variance Std.Dev.
 Target.Trial. (Intercept) 2.1794   1.4763
 Subject       (Intercept) 0.6628   0.8141
```

```
Number of obs: 4176, groups:  Target.Trial., 72; Subject, 58

Fixed effects:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      -1.1647     0.2371  -4.912 9.03e-07 ***
PrimeCondition1  -0.1425     0.2292  -0.622    0.534
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Correlation of Fixed Effects:
            (Intr)
PrimeCndtn1 -0.464
```

```
> # confint(prime_lmer2)
> #
> # > confint(prime_lmer2)
> # Computing profile confidence intervals ...
> #                     2.5 %     97.5 %
> # .sig01          1.2331309  1.7946488
> # .sig02          0.6586465  1.0185114
> # (Intercept)    -1.6374292 -0.6988686
> # PrimeCondition1 -0.6009819  0.3140128
```

1.2 Prime and Target Acc

```
> ## PRIME ACCURACY
> ## AOV by subject
> library(dplyr)
> primeacc  = group_by(US, Subject, PrimeCondition ) %>%
+   summarise_at(vars(PrimeFirstResp_ACC), mean)
>  primeacc_aov = aov(data = primeacc, PrimeFirstResp_ACC ~ PrimeCondition)
>  summary(primeacc_aov)
```

```
               Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition  1 0.0514 0.05143    4.67  0.035 *
Residuals      56 0.6167 0.01101
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
>  prime_r = primeacc %>% filter(PrimeCondition == "Semantic")
>   prime_u = primeacc %>% filter(PrimeCondition == "Unrelated")
> t.test(prime_r$PrimeFirstResp_ACC, prime_u$PrimeFirstResp_ACC, paired = FALSE)
```

```
        Welch Two Sample t-test

data:  prime_r$PrimeFirstResp_ACC and prime_u$PrimeFirstResp_ACC
```

```
t = -2.1485, df = 53.071, p-value = 0.03625
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.115218008 -0.003961886
sample estimates:
mean of x mean of y
0.4325397 0.4921296
```

```
>  ## AOV by item
>
> primeacc2  = group_by(US, Target.Trial., PrimeCondition ) %>%
+   summarise_at(vars(PrimeFirstResp_ACC), mean)
>  primeacc_aov2 = aov(data = primeacc2, PrimeFirstResp_ACC ~ PrimeCondition +
+                     Error(Target.Trial./PrimeCondition))
>  summary(primeacc_aov2)
```

```
Error: Target.Trial.
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 71  6.535 0.09204

Error: Target.Trial.:PrimeCondition
               Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition  1  0.128 0.12783   1.513  0.223
Residuals      71  5.997 0.08446
```

```
>  ## TARGET ACCURACY
> ## AOV by subject
>
> targetacc  = group_by(US, Subject, PrimeCondition ) %>%
+   summarise_at(vars(TargetFirstResp_ACC), mean)
>  targetacc_aov = aov(data = targetacc, TargetFirstResp_ACC ~ PrimeCondition)
>  summary(targetacc_aov)
```

```
               Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition  1 0.0047 0.00467   0.285  0.595
Residuals      56 0.9163 0.01636
```

```
>  ## AOV by item
>
> targetacc2  = group_by(US, Target.Trial., PrimeCondition ) %>%
+   summarise_at(vars(TargetFirstResp_ACC), mean)
>  targetacc_aov2 = aov(data = targetacc2, TargetFirstResp_ACC ~ PrimeCondition +
+                     Error(Target.Trial./PrimeCondition))
>  summary(targetacc_aov2)
```

```
Error: Target.Trial.
          Df Sum Sq Mean Sq F value Pr(>F)
```

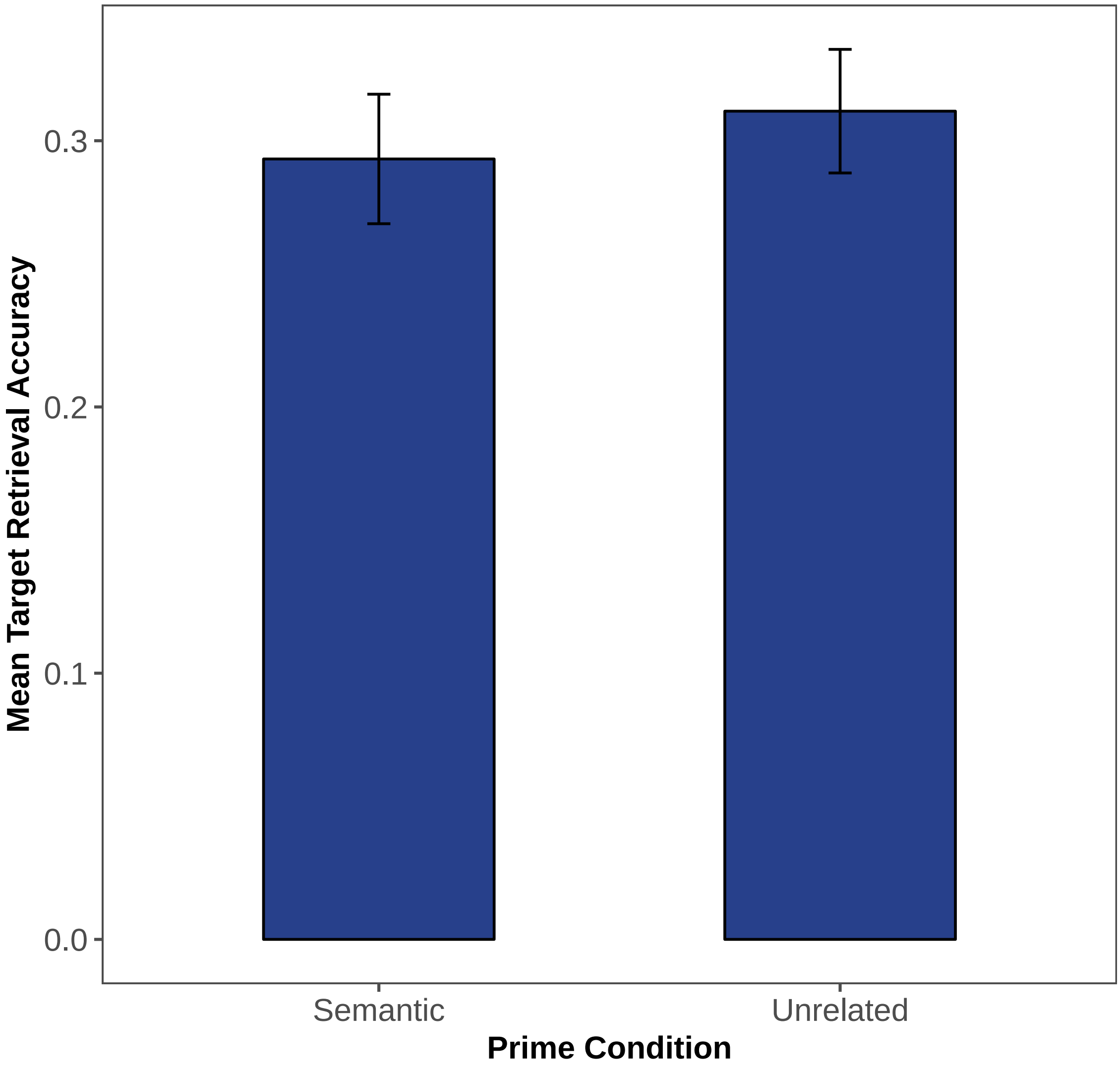
```
Residuals 71  7.582  0.1068

Error: Target.Trial.:PrimeCondition
               Df Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition  1 0.0116 0.011607   2.182  0.144
Residuals      71 0.3777 0.005319
```

Figures: Mean Accuracy

Target

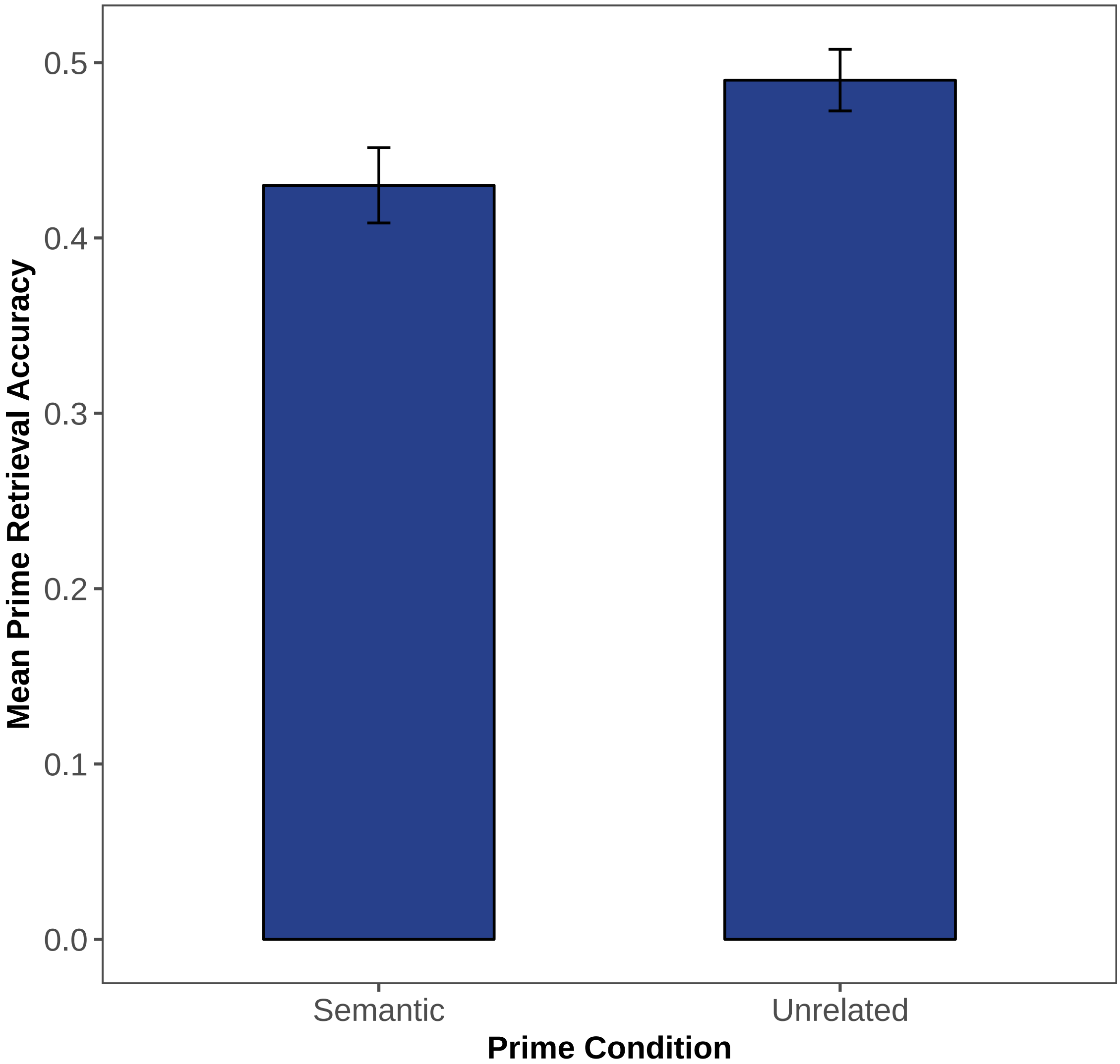
```
> agg_acc = Rmisc::summarySE(targetacc,
+                       measurevar = "TargetFirstResp_ACC",
+                       groupvars = c("PrimeCondition"))
> library(ggplot2)
> library(ggthemes)
>  agg_acc %>% mutate(PrimeType = factor(PrimeCondition,
+                                                  levels = unique(PrimeCondition),
+                      labels = c("Semantic", "Unrelated")))%>%
+   ggplot(aes(x = PrimeType, y = TargetFirstResp_ACC)) +
+  geom_bar(stat = "identity", position = "dodge", width = 0.5,
+           fill = "royalblue4", color = "black")+
+    geom_errorbar(aes(ymin = TargetFirstResp_ACC - se, ymax = TargetFirstResp_ACC + se)
+                 width=.05, position=position_dodge(.5)) +
+     theme_few()+
+    xlab("Prime Condition") + ylab("Mean Target Retrieval Accuracy") +
+   ggtitle("")  +
+    theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
>
```



Prime

```
> agg_prime_acc = Rmisc::summarySE(primeacc,
+                       measurevar = "PrimeFirstResp_ACC",
+                       groupvars = c("PrimeCondition"))
> agg_prime_acc$PrimeFirstResp_ACC = round(agg_prime_acc$PrimeFirstResp_ACC,
+                                          digits = 2)
> library(ggplot2)
> library(ggthemes)
>  agg_prime_acc %>% mutate(PrimeType = factor(PrimeCondition,
+                                    levels = unique(PrimeCondition),
```

```
+                       labels = c("Semantic", "Unrelated")))%>%
+    ggplot(aes(x = PrimeType, y = PrimeFirstResp_ACC)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            fill = "royalblue4", color = "black")+
+     geom_errorbar(aes(ymin = PrimeFirstResp_ACC - se,
+                       ymax = PrimeFirstResp_ACC + se),
+                  width=.05, position=position_dodge(.5)) +
+      theme_few()+
+     xlab("Prime Condition") + ylab("Mean Prime Retrieval Accuracy") +
+    ggtitle("")  +
+     theme(axis.text = element_text(size = rel(1)),
+            axis.title = element_text(face = "bold", size = rel(1)),
+            legend.title = element_text(face = "bold", size = rel(1)),
+           plot.title = element_text(hjust = .5),
+           strip.text.x = element_text(face = "bold", size = rel(1.4)))
>
```



1.3 Proportion Ret/Not Ret

```
> library(dplyr)
> cued_acc = group_by(US, ExperimentName) %>%
+   summarise_at(vars(PrimeFirstResp_ACC, TargetFirstResp_ACC), mean)
> cued_acc = group_by(US, ExperimentName, Subject, PrimeFirstResp_ACC) %>%
+   summarise(recalltrials = n())
> conditional_acc = group_by(US, ExperimentName, Subject,
+                            PrimeFirstResp_ACC, TargetFirstResp_ACC) %>%
+   summarise(trials = n())
```

```
> merge_acc = merge(conditional_acc, cued_acc,
+                   by = c("Subject", "PrimeFirstResp_ACC", "ExperimentName"))
> merge_acc$prop = merge_acc$trials/merge_acc$recalltrials
> merge_acc$Subject =
+   as.factor(as.character(merge_acc$Subject))
> merge_acc$PrimeFirstResp_ACC =
+   as.factor(as.character(merge_acc$PrimeFirstResp_ACC))
> merge_acc$TargetFirstResp_ACC =
+   as.factor(as.character(merge_acc$TargetFirstResp_ACC))
> cond_aov = aov(data = merge_acc,
+         prop ~ ExperimentName*PrimeFirstResp_ACC*TargetFirstResp_ACC +
+         Error(Subject/(PrimeFirstResp_ACC*TargetFirstResp_ACC)))
> summary(cond_aov)
```

```
Error: Subject
               Df    Sum Sq   Mean Sq F value   Pr(>F)
ExperimentName  1 4.978e-30 4.978e-30   25.95 4.27e-06 ***
Residuals      56 1.074e-29 1.920e-31
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:PrimeFirstResp_ACC
                                  Df    Sum Sq   Mean Sq F value Pr(>F)
PrimeFirstResp_ACC                 1 1.800e-30 1.799e-30   0.459  0.501
ExperimentName:PrimeFirstResp_ACC  1 1.650e-30 1.649e-30   0.420  0.519
Residuals                         56 2.197e-28 3.923e-30

Error: Subject:TargetFirstResp_ACC
                                   Df Sum Sq Mean Sq F value Pr(>F)
TargetFirstResp_ACC                 1  8.799   8.799 143.179 <2e-16 ***
ExperimentName:TargetFirstResp_ACC  1  0.003   0.003   0.052  0.821
Residuals                          56  3.441   0.061
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:PrimeFirstResp_ACC:TargetFirstResp_ACC
                                                      Df Sum Sq Mean Sq F value
PrimeFirstResp_ACC:TargetFirstResp_ACC                 1 0.3334  0.3334   38.60
ExperimentName:PrimeFirstResp_ACC:TargetFirstResp_ACC  1 0.5862  0.5862   67.86
Residuals                                             56 0.4838  0.0086
                                                        Pr(>F)
PrimeFirstResp_ACC:TargetFirstResp_ACC                6.84e-08 ***
ExperimentName:PrimeFirstResp_ACC:TargetFirstResp_ACC 3.15e-11 ***
Residuals
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> ## prime condition effect
```

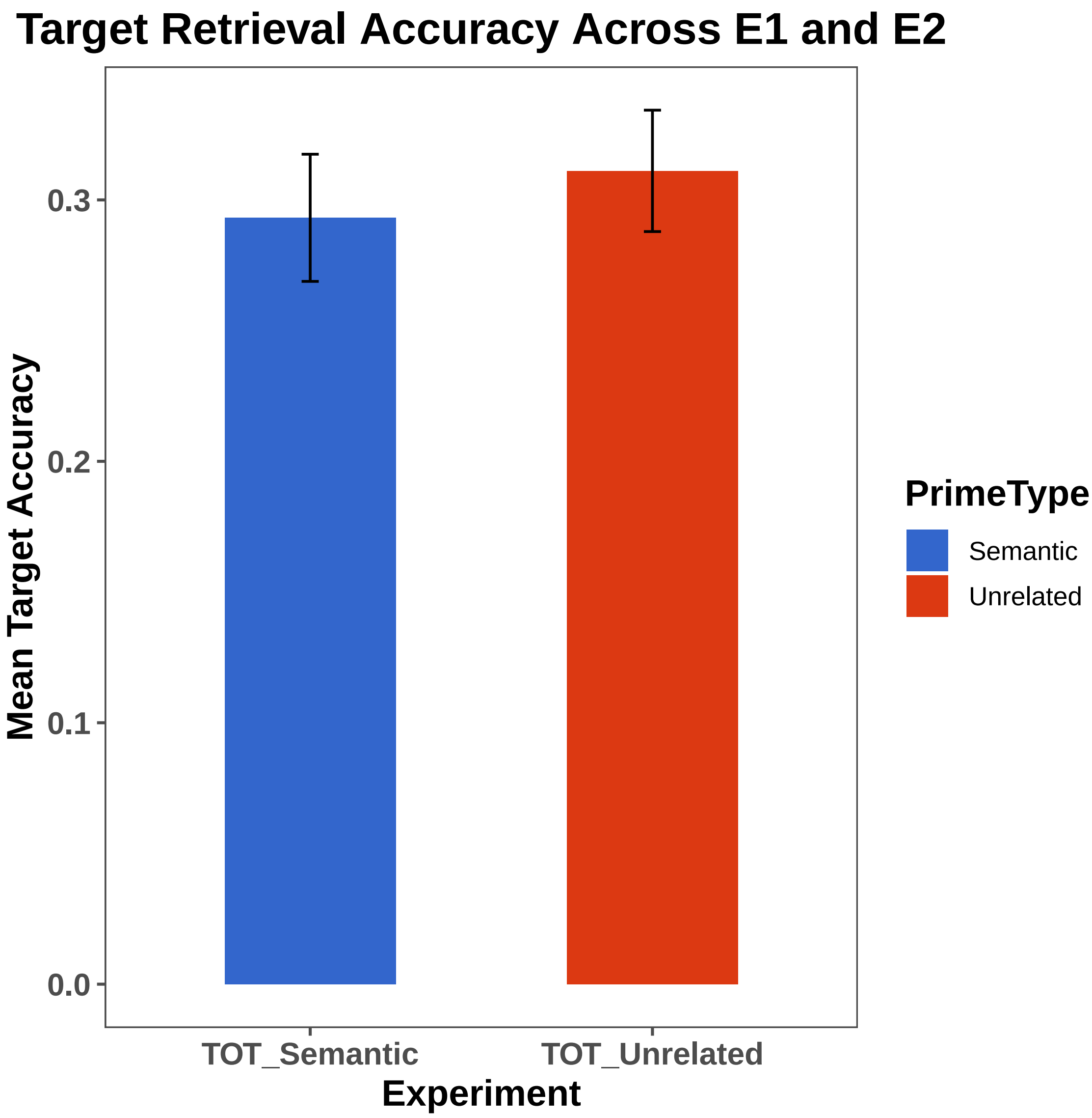
```
>
> prime_sub = group_by(US, ExperimentName, Subject, PrimeCondition) %>%
+   summarise_at(vars(TargetFirstResp_ACC), mean)
> prime_aov = aov(data = prime_sub, TargetFirstResp_ACC ~ PrimeCondition)
> summary(prime_aov)
```

```
               Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition  1 0.0047 0.00467   0.285  0.595
Residuals      56 0.9163 0.01636
```

```
>
```

Target Accuracy Figure

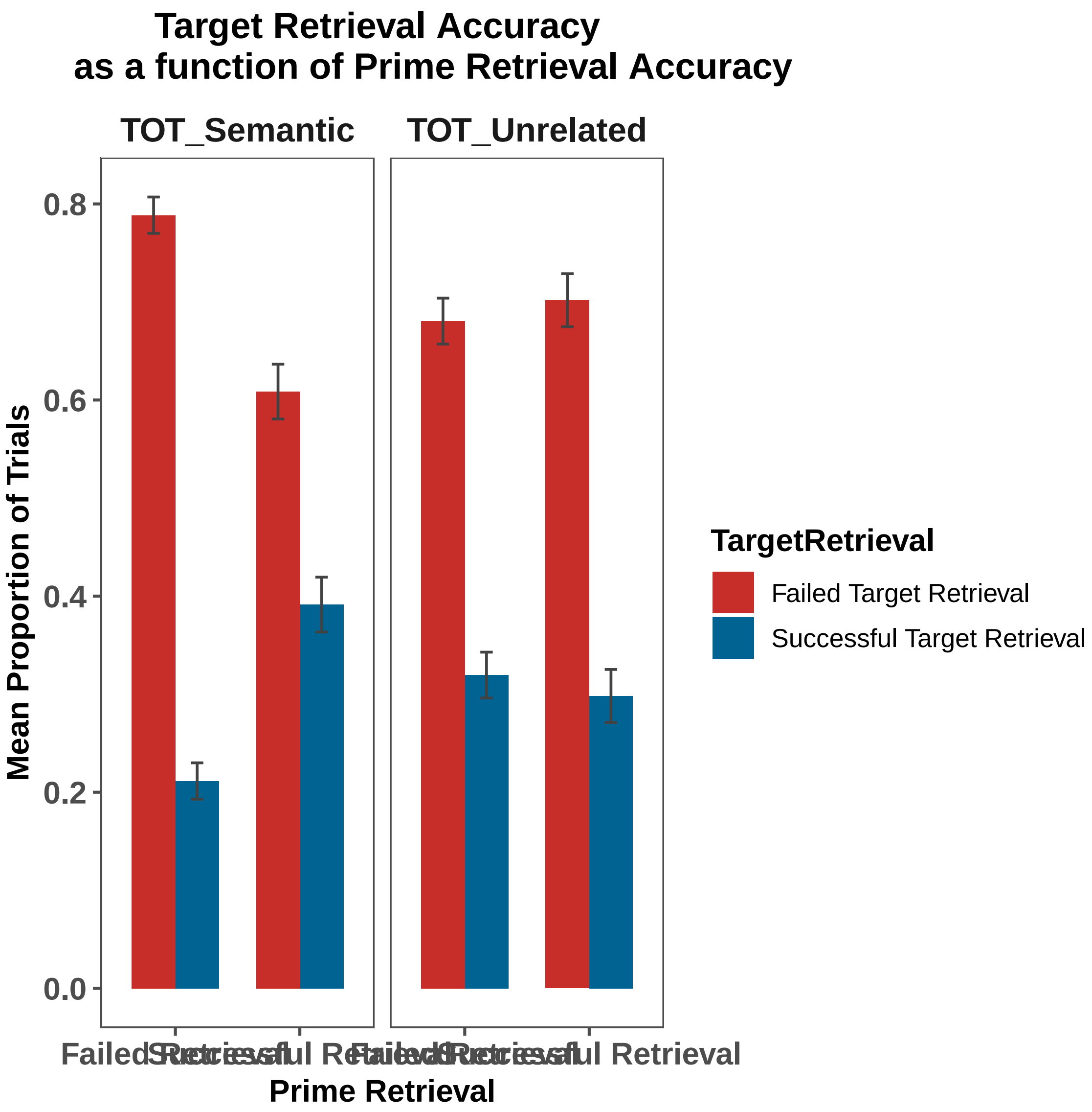
```
> target_rmisc = Rmisc::summarySE(prime_sub,
+                       measurevar = "TargetFirstResp_ACC",
+                       groupvars = c("ExperimentName","PrimeCondition"))
> library(ggplot2)
> library(ggthemes)
> target_rmisc %>% mutate(PrimeType = factor(PrimeCondition,
+                                              levels = unique(PrimeCondition),
+                     labels = c("Semantic", "Unrelated"))) %>%
+ ggplot(aes(x = ExperimentName, y = TargetFirstResp_ACC,
+            group = PrimeType, fill = PrimeType))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.5)+
+   geom_errorbar(aes(ymin = TargetFirstResp_ACC - se, ymax = TargetFirstResp_ACC + se),
+                 width=.05, position=position_dodge(.5)) +
+   theme_few()+
+   scale_fill_gdocs()+
+   xlab("Experiment") + ylab("Mean Target Accuracy") +
+   ggtitle("Target Retrieval Accuracy Across E1 and E2") +
+     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))
```



1.4 Conditional Figure

```
> cond_figure = Rmisc::summarySE(merge_acc,
+                         measurevar = "prop",
+                         groupvars = c("ExperimentName", "PrimeFirstResp_ACC",
+                                       "TargetFirstResp_ACC"))
> library(ggplot2)
> library(ggthemes)
> condfigure_plot = cond_figure %>% mutate(Recall = factor(PrimeFirstResp_ACC,
+                       levels = unique(PrimeFirstResp_ACC),
```

```
+                     labels = c("Failed Retrieval",
+                                "Successful Retrieval")),
+                     TargetRetrieval = factor(TargetFirstResp_ACC,
+                           levels = unique(TargetFirstResp_ACC),
+                        labels = c("Failed Target Retrieval",
+                             "Successful Target Retrieval")))%>%
+ ggplot(aes(x = Recall, y = prop,
+            fill = TargetRetrieval, group = TargetRetrieval))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.7)+
+   geom_errorbar(aes(ymin=prop - se, ymax=prop + se),
+              width=.2, color = "gray26",
+              position = position_dodge(0.7))+
+  theme_few()+
+   facet_wrap(~ExperimentName)+
+   scale_fill_wsj()+
+     xlab("Prime Retrieval") + ylab("Mean Proportion of Trials") +
+   ggtitle("Target Retrieval Accuracy
+           as a function of Prime Retrieval 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
```



1.5 Follow Up Tests

For each subject, we will calculate a difference score for drop off in accuracy when they failed to recall the item vs. when they successfully retrieved the item.

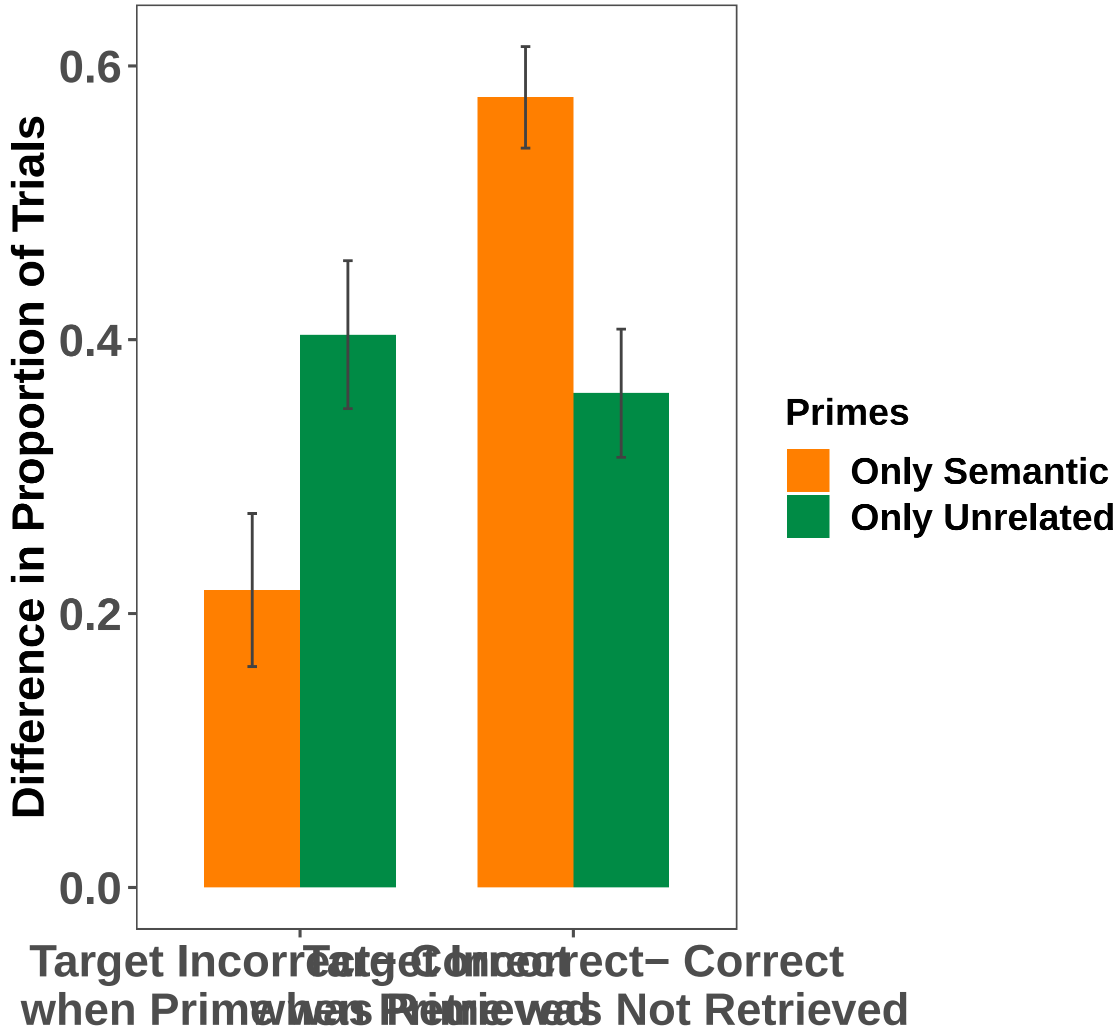
```
> failedrecall = merge_acc %>% filter(PrimeFirstResp_ACC == "0")
> failedrecall = failedrecall[,-c(2,5,6)]
> successfulrecall = merge_acc %>% filter(PrimeFirstResp_ACC == "1")
> successfulrecall = successfulrecall[,-c(2,5,6)]
> ## need to convert from long to wide: using spread
> library(tidyr)
```

```
> failed_wide = failedrecall %>%
+   spread(TargetFirstResp_ACC, prop)
> failed_wide$cost = failed_wide$`0` - failed_wide$`1`
> colnames(failed_wide) = c("Subject", "ExperimentName", "Failed:Incorrect", "Failed:Cor
> successful_wide = successfulrecall %>%
+   spread(TargetFirstResp_ACC, prop)
> successful_wide$benefit = successful_wide$`0` - successful_wide$`1`
> colnames(successful_wide) = c("Subject", "ExperimentName", "Successful:Incorrect", "Su
> merged_cost_benefit = merge(failed_wide, successful_wide, by = c("Subject", "Experimen
> merged_cost_benefit = merged_cost_benefit[,-c(3,4,6,7)]
> ## convert to long for plotting
>
> costbenefit_long = merged_cost_benefit %>%
+   gather(Difference, Proportion, Cost:Benefit)
```

1.6 Difference Figure

```
> costbenefit_plot = Rmisc::summarySE(costbenefit_long,
+                         measurevar = "Proportion",
+                         groupvars = c("ExperimentName", "Difference"))
> library(ggplot2)
> library(ggthemes)
> costbenefit_plot_fig = costbenefit_plot %>% mutate(`Difference Type` = factor(Differen
+                       levels = unique(Difference),
+                     labels = c("Target Incorrect- Correct\n when Prime was Retrieved",
+                  "Target Incorrect- Correct\n when Prime was Not Retrieved")),
+                     Primes = factor(ExperimentName,
+                           levels = unique(ExperimentName),
+                        labels = c("Only Semantic",
+                             "Only Unrelated")))%>%
+ ggplot(aes(x = `Difference Type`, y = Proportion,
+            fill = Primes, group = Primes))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.7)+
+   geom_errorbar(aes(ymin=Proportion - se, ymax=Proportion + se),
+              width=.07, color = "gray26",
+              position = position_dodge(0.7))+
+  theme_few()+
+   scale_fill_manual(values = c("darkorange1", "springgreen4"))+
+     xlab("") + ylab("Difference in Proportion of Trials") +
+   ggtitle("")  +
+    theme(axis.text = element_text(face = "bold", size = rel(1.4)),
+          axis.title.y = element_text(face = "bold", size = rel(1.4)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1.2)),
+           plot.title = element_text(face = "bold",
+                   size = rel(1.4), hjust = .5),
+          legend.text = element_text(face = "bold", size = rel(1.2)),
```

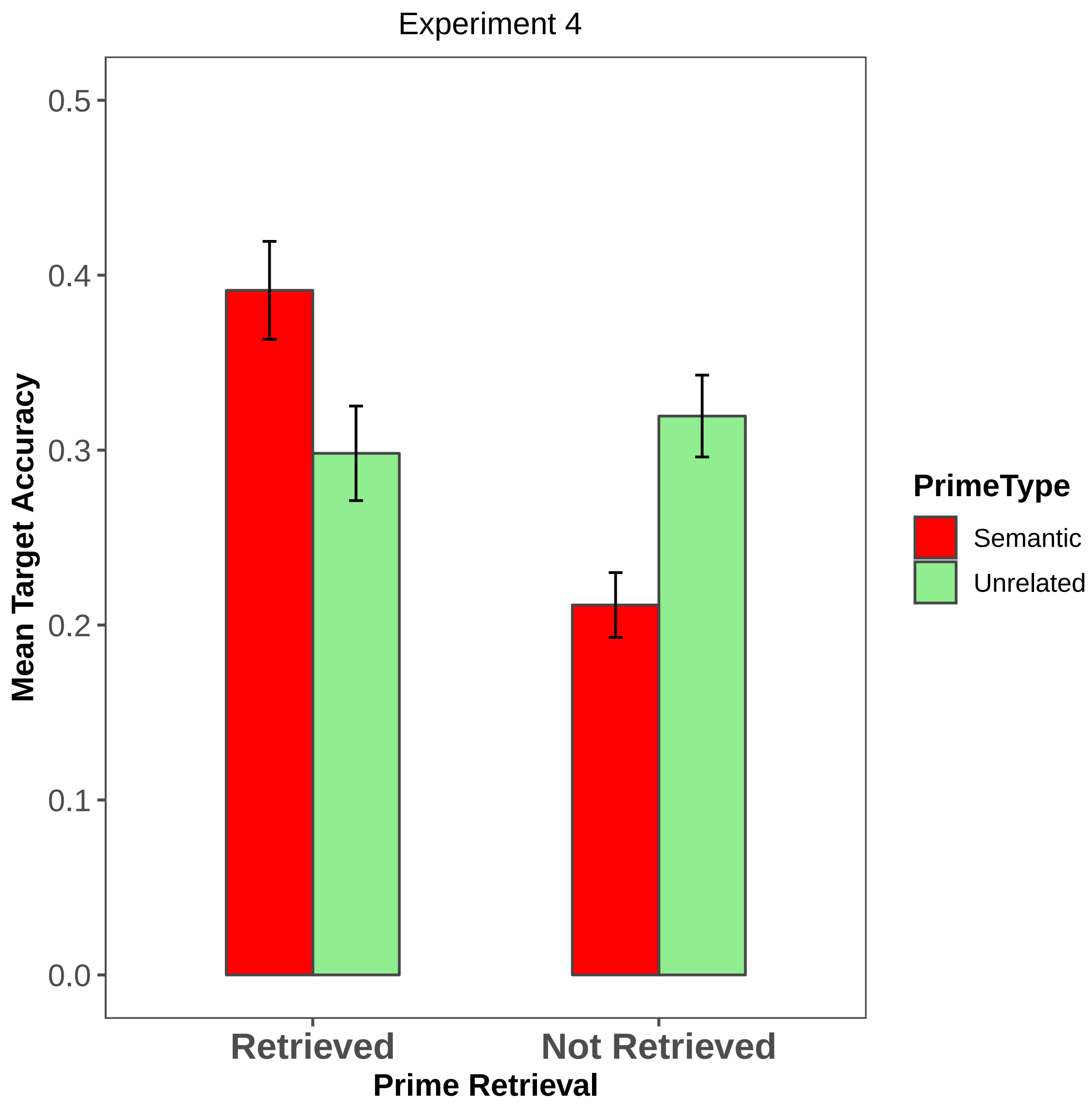
```
+             strip.text.x = element_text(face = "bold", size = rel(1.4)))
> costbenefit_plot_fig
```



1.7 Retrieval Figure

```
> US_fig = US
> US_fig$primefac = ordered(as.factor(as.character(US_fig$PrimeCondition)),
+                       levels = c("Semantic", "Unrelated"))
> US_fig$PrimeFirstResp_ACC_fac = ordered(as.factor(as.character(US_fig$PrimeFirstResp_A
> targetacc2  = group_by(US_fig, Subject, primefac, PrimeFirstResp_ACC_fac ) %>%
+   summarise_at(vars(TargetFirstResp_ACC), mean)
```

```
> ret_figure = Rmisc::summarySE(targetacc2,
+                     measurevar = "TargetFirstResp_ACC",
+                 groupvars = c("primefac", "PrimeFirstResp_ACC_fac"))
> ret_figure = ret_figure %>% arrange(PrimeFirstResp_ACC_fac)
> library(ggplot2)
> library(ggthemes)
> ret_figure  %>% mutate(PrimeType = factor(primefac,
+                                         levels = unique(primefac),
+                     labels = c("Semantic",
+                                "Unrelated")),
+                     `Prime Retrieval` = factor(PrimeFirstResp_ACC_fac,
+                                levels = unique(PrimeFirstResp_ACC_fac),
+                     labels = c("Retrieved", "Not Retrieved")))%>%
+    ggplot(aes(x = `Prime Retrieval`, y = TargetFirstResp_ACC,
+                           group =PrimeType ,
+                           fill = PrimeType)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+    geom_errorbar(aes(ymin = TargetFirstResp_ACC - se,
+                      ymax = TargetFirstResp_ACC + 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") +
+   ylim (0,0.5)+
+ ggtitle(" Experiment 4") +
+   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)))
>
```



2 Percent State Analysis

```
> state = read.csv("TOTUnrelatedSemantic_agg.csv",header = TRUE, sep = ",")
> j_statepercent = state[,c(1,2,12:15)] # use for prime percents
> j_statepercent$Subject = as.factor(j_statepercent$Subject)
> library(tidyr)
> library(dplyr)
> statepercent <- j_statepercent %>%
+   gather(State, Percent,
+          prop_know, prop_dontknow, prop_other, prop_TOT)%>%
```

```
+      separate(State, c('Prop', 'State'), sep = "_") %>%
+    arrange(Subject)
> #removing prop
> statepercent = statepercent[,-3]
> colnames(statepercent) = c( "PrimeCondition", "Subject", "State", "Percent")
> statepercent$Subject ← as.factor(statepercent$Subject)
> statepercent$PrimeCondition ← as.factor(statepercent$PrimeCondition)
> statepercent$State ← as.factor(statepercent$State)
> statepercent$Percent ← as.numeric(as.character(statepercent$Percent))
> ## anova
>
> state_aov = aov(data = statepercent, Percent ~ PrimeCondition*State +
+                     Error(Subject/(State)))
> summary(state_aov)
```

```
Error: Subject
               Df    Sum Sq   Mean Sq F value Pr(>F)
PrimeCondition  1 3.800e-20 3.795e-20   0.785  0.379
Residuals      56 2.708e-18 4.835e-20

Error: Subject:State
                      Df Sum Sq Mean Sq F value Pr(>F)
State                  3  2.698  0.8995  44.267 <2e-16 ***
PrimeCondition:State   3  0.014  0.0046   0.225  0.879
Residuals            168  3.414  0.0203
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

2.0.1 plot

```
> ## figure
> state_rmisc = Rmisc::summarySE(statepercent,
+                                measurevar = "Percent",
+                                groupvars = c("PrimeCondition","State"))
> x ← c("know","dontknow", "other", "TOT")
> state_rmisc = state_rmisc %>%
+    mutate(rstate =  factor(State, levels = x)) %>%
+    arrange(rstate)
> library(ggplot2)
> library(ggthemes)
> percentplot = state_rmisc %>%
+    mutate(PrimeType = factor(PrimeCondition, levels = unique(PrimeCondition),
+                    labels = c("Semantic", "Unrelated")),
+     R = factor(rstate, levels = unique(rstate),
+                             labels = c( "1: Know","2: Dont Know",
+                                         "3:Other", "4: TOT")))%>%
+ ggplot(aes(x = R, y = Percent,
```

```
+              group = PrimeType, fill = PrimeType))+
+   geom_bar(stat = "identity", position = "dodge", width = 0.7,
+              color= "black")+
+    geom_errorbar(aes(ymin=Percent - se, ymax=Percent + se),
+                  width=.2, color = "gray26",
+                  position = position_dodge(0.7))+
+   theme_few()+
+      xlab("") + ylab("Percentage of trials") +
+   scale_fill_manual(values = c( "red",
+                                 "lightgreen"))+
+    ggtitle("E4")  +
+     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
```

2.0.2 know

```
> e4_know = statepercent %>% filter(State == "know")
> e4_know_aov = aov(data = e4_know,
+                         Percent ~ PrimeCondition)
> summary(e4_know_aov)
```

```
               Df Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition  1 0.0011 0.001138   0.058  0.811
Residuals      56 1.1078 0.019782
```

2.0.3 dont know

```
> e4_dontknow = statepercent %>% filter(State == "dontknow")
> e4_dontknow_aov = aov(data = e4_dontknow,
+                         Percent ~ PrimeCondition )
> summary(e4_dontknow_aov)
```

```
               Df Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition  1  0.004 0.004007   0.168  0.684
Residuals      56  1.337 0.023866
```

2.0.4 other

```
> e4_other = statepercent %>% filter(State == "other")
> e4_other_aov = aov(data = e4_other,
```

```
+                              Percent ~ PrimeCondition )
> summary(e4_other_aov)
```

```
               Df Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition  1 0.0000 0.000021   0.002  0.964
Residuals      56 0.5479 0.009785
```

2.0.5 TOT

```
> e4_TOT = statepercent %>% filter(State == "TOT")
> e4_TOT_aov = aov(data = e4_TOT,
+                              Percent ~ PrimeCondition)
> summary(e4_TOT_aov)
```

```
               Df Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition  1 0.0086 0.008556   1.137  0.291
Residuals      56 0.4214 0.007526
```

3 LMER model

```
> library(lme4)
> ## adding prime acc as a covariate
>
> # participant_acc = group_by(US, Subject) %>%
> #   summarise_at(vars(TargetFirstResp_ACC, PrimeFirstResp_ACC), mean)
> #
> # participant_acc$MeanAcc = (participant_acc$TargetFirstResp_ACC +
> #                            participant_acc$PrimeFirstResp_ACC)/2
> #
> # colnames(participant_acc) = c("Subject", "TargetAcc", "PrimeAcc", "MeanAcc")
> #
> # US2 = merge(US, participant_acc[,c(1,3,4)],
> #                        by = c("Subject"))
>
> ## accounting for mean prime accuracy
>
> item_acc = group_by(US, Target.Trial., PrimeCondition) %>%
+   summarise_at(vars(PrimeFirstResp_ACC), mean)
> colnames(item_acc) = c("Target.Trial.","PrimeCondition","PrimeAcc")
> US2 = merge(US, item_acc,
+                        by = c("Target.Trial.", "PrimeCondition"))
> contrasts(US2$PrimeCondition) = contr.treatment(2, base = 2)
> US2$PrimeFirstResp_ACC = as.factor(US2$PrimeFirstResp_ACC)
> lmer_model_acc = lme4::glmer(data = US2, TargetFirstResp_ACC ~
+                             PrimeFirstResp_ACC*PrimeCondition + PrimeAcc +
```

```
+                           (1|Subject) + (1|Target.Trial.),
+                          family = "binomial",
+                          control=glmerControl(optimizer="bobyqa",
+           optCtrl=list(maxfun=100000)))
> summary(lmer_model_acc)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula:
TargetFirstResp_ACC ~ PrimeFirstResp_ACC * PrimeCondition + PrimeAcc +
    (1 | Subject) + (1 | Target.Trial.)
   Data: US2
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))

     AIC      BIC   logLik deviance df.resid
  3993.1   4037.4  -1989.5   3979.1     4169

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.5504 -0.5192 -0.2794  0.4877  7.2087

Random effects:
 Groups        Name        Variance Std.Dev.
 Target.Trial. (Intercept) 2.2508   1.5003
 Subject       (Intercept) 0.5847   0.7647
Number of obs: 4176, groups:  Target.Trial., 72; Subject, 58

Fixed effects:
                                    Estimate Std. Error z value Pr(>|z|)
(Intercept)                          -0.7611     0.2565  -2.967  0.00301 **
PrimeFirstResp_ACC1                   0.2761     0.1357   2.034  0.04193 *
PrimeCondition1                      -0.4401     0.2354  -1.869  0.06158 .
PrimeAcc                             -1.1112     0.2306  -4.818 1.45e-06 ***
PrimeFirstResp_ACC1:PrimeCondition1   0.5679     0.1805   3.147  0.00165 **
---
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) PrFR_ACC1 PrmCn1 PrmAcc
PrmFrR_ACC1 -0.113
PrimeCndtn1 -0.453  0.241
PrimeAcc    -0.330 -0.351     0.028
PFR_ACC1:PC  0.173 -0.647    -0.383  0.004
```

```
> car::Anova(lmer_model_acc)
```

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

```
Response: TargetFirstResp_ACC
                                      Chisq Df Pr(>Chisq)
PrimeFirstResp_ACC                  28.4789  1  9.473e-08 ***
PrimeCondition                       0.5172  1   0.472056
PrimeAcc                            23.2175  1  1.447e-06 ***
PrimeFirstResp_ACC:PrimeCondition    9.9016  1   0.001651 **
---
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(lmer_model_acc)
```

```
Analysis of Variance Table
                                  Df  Sum Sq Mean Sq F value
PrimeFirstResp_ACC                 1 12.6485 12.6485 12.6485
PrimeCondition                     1  0.3203  0.3203  0.3203
PrimeAcc                           1 23.6301 23.6301 23.6301
PrimeFirstResp_ACC:PrimeCondition  1  9.9603  9.9603  9.9603
```

```
>
> ### NOTE: Ask about best way to covary out prime accuracy
```

4 z-scoring RTs

RT prime and Target

```
> library(dplyr)
> colnames(US) = c( "ExperimentName", "Subject","ID", "Session", "Procedure", "Trial", "
+                              "PrimeDefRT", "PrimeResp",
+                             "PrimeRespRT", "Stimuli1",
+                             "Target", "TargetDefResp", "TargetRT",
+                              "State", "StateRT", "TargetResp", "TargetRespRT",
+                              "PrimeAcc", "Accuracy",
+                              "RTrecognisePrime", "RTrecogniseTarget",
+                   "PrimeCondition", "PrimeRespType", "TargetRespType", "Prime_POS",
+                   "Target_POS")
> #US_firsttrim = US %>% filter(PrimeAcc == 1)
> US_firsttrim_target = subset(US,
+                                   US$RTrecogniseTarget > 250 &
+                                  US$RTrecogniseTarget < 7000)
> US_firsttrim_prime = subset(US,
+                                   US$RTrecognisePrime > 250 &
+                                  US$RTrecognisePrime < 7000)
> US_firsttrim_targetdef = subset(US,
+                                   US$TargetDefRT > 250 &
+                                  US$TargetDefRT < 9000)
```

RTRecogniseprime

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

RTRecogniseTarget

```
> ## FOR TARGET
> ## aggregate per subject all IVs and DVs
> meanRT = group_by(US_firsttrim_target, Subject) %>%
+   summarise_at(vars(RTrecogniseTarget), mean)
> colnames(meanRT) = c("Subject", "MeanRTrecogTarget")
> sdRT = group_by(US_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
> US_z_target= merge(US_firsttrim_target,
+                              RT_agg, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> US_z_target = US_z_target %>% mutate( zTargetRecogRT =
+                                              (RTrecogniseTarget -
+                                                 MeanRTrecogTarget)/sdRTrecogTarget)
> ## checking: subject level means should be zero
>
```

```
> sub_pic = group_by(US_z_target, Subject) %>%
+   summarise_at(vars(zTargetRecogRT), mean)
>
```

TargetDefRT

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

4.1 Trimming z-RTs

```
> #Note: trimming separately!!
> US_z_trimmed_prime = subset(US_z_prime, US_z_prime$zPrimeRecogRT < 3 &
+                                  US_z_prime$zPrimeRecogRT > -3)
> US_z_trimmed_target = subset(US_z_target,
+                                US_z_target$zTargetRecogRT < 3 &
+                                  US_z_target$zTargetRecogRT > -3)
> US_z_trimmed_targetdef = subset(US_z_targetdef,
+                                US_z_targetdef$zTargetRT < 3 &
+                                  US_z_targetdef$zTargetRT > -3)
```

4.2 Repeating z-scoring

4.3 For RTrecogniseprime

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

4.4 For RTrecognisetarget

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

```
>
> sub_pic = group_by(US_final_z_target, Subject) %>%
+   summarise_at(vars(zTargetRecogRT_trim), mean)
>
```

4.5 For TargetDefRT

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

4.6 Combining z-RT Prime and Target

```
> ## now we have separately z-scored RTprime and RTtarget. Need to combine.
> ## taking only necessary columns
> US_final_z_prime2 = US_final_z_prime[,c(2,6,34)]
> US_final_z = merge(US_final_z_target,
+                              US_final_z_prime2,
+                              by  = c("Subject", "Trial"))
> US_final_z_targetdef = merge(US_final_z_targetdef,
+                              US_final_z_prime2,
+                              by  = c("Subject", "Trial"))
```

4.7 Linear Models

```
> # Mean RT to retrieve Target as a function of Prime Condition
>
> # Effect of RT prime on Accuracy
> library(lme4)
> contrasts(US_final_z_prime$ExperimentName) = contr.treatment(2, base = 2)
> RTprime_acc_model = glmer(data = US_final_z_prime,
+                          Accuracy ~ ExperimentName*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: Accuracy ~ ExperimentName * zPrimeRecogRT_trim + (1 | Subject) +
    (1 | Target)
   Data: US_final_z_prime

     AIC      BIC   logLik deviance df.resid
  3898.7   3936.6  -1943.3   3886.7     4056

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.7846 -0.5194 -0.2821  0.4978  5.8965

Random effects:
 Groups  Name        Variance Std.Dev.
 Target  (Intercept) 2.1623   1.470
 Subject (Intercept) 0.6839   0.827
Number of obs: 4062, groups:  Target, 72; Subject, 57

Fixed effects:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                       -1.16959    0.24010  -4.871 1.11e-06 ***
ExperimentName1                   -0.15191    0.23472  -0.647   0.5175
zPrimeRecogRT_trim                -0.01920    0.06337  -0.303   0.7619
ExperimentName1:zPrimeRecogRT_trim -0.22940    0.09080  -2.526   0.0115 *
---
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) ExprN1 zPRRT_
ExpermntNm1 -0.477
zPrmRcgRT_t -0.001  0.001
ExpN1:PRRT_  0.002  0.017 -0.684
```

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

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

Response: Accuracy
                                  Chisq Df Pr(>Chisq)
ExperimentName                   0.3651  1   0.545678
zPrimeRecogRT_trim               7.7327  1   0.005423 **
ExperimentName:zPrimeRecogRT_trim 6.3821  1   0.011528 *
---
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
ExperimentName                     1 0.3060  0.3060  0.3060
zPrimeRecogRT_trim                 1 7.7398  7.7398  7.7398
ExperimentName:zPrimeRecogRT_trim  1 6.4002  6.4002  6.4002
```

```
> # > confint(RTprime_acc_model)
> # Computing profile confidence intervals ...
> #                                         2.5 %      97.5 %
> # .sig01                             1.2274544  1.78845270
> # .sig02                             0.6684390  1.03710333
> # (Intercept)                       -1.6476725 -0.69786185
> # ExperimentName1                   -0.6214896  0.31564775
> # zPrimeRecogRT_trim                -0.1456978  0.10624292
> # ExperimentName1:zPrimeRecogRT_trim -0.4103775 -0.04937913
>
> contrasts(US_final_z$ExperimentName) = contr.treatment(2, base = 2)
> library(lmerTest)
> RTprime_RT_model = lmer(data = US_final_z,
+                   zTargetRecogRT_trim ~ ExperimentName*zPrimeRecogRT_trim +
+                             (1|Subject) + (1|Target))
> summary(RTprime_RT_model)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: zTargetRecogRT_trim ~ ExperimentName * zPrimeRecogRT_trim + (1 |
    Subject) + (1 | Target)
   Data: US_final_z

REML criterion at convergence: 9950.9

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.4569 -0.6513 -0.0947  0.5920  4.0751
```

```
Random effects:
 Groups   Name        Variance Std.Dev.
 Target   (Intercept) 0.3326   0.5767
 Subject  (Intercept) 0.0000   0.0000
 Residual             0.6674   0.8170
Number of obs: 3983, groups:  Target, 72; Subject, 57

Fixed effects:
                                   Estimate Std. Error        df t value
(Intercept)                       2.387e-02  7.036e-02  7.548e+01   0.339
ExperimentName1                  -6.526e-03  2.590e-02  3.908e+03  -0.252
zPrimeRecogRT_trim                4.983e-02  1.999e-02  3.952e+03   2.492
ExperimentName1:zPrimeRecogRT_trim  7.073e-02  2.818e-02  3.949e+03   2.510
                                 Pr(>|t|)
(Intercept)                        0.7354
ExperimentName1                    0.8011
zPrimeRecogRT_trim                 0.0127 *
ExperimentName1:zPrimeRecogRT_trim   0.0121 *
---
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) ExprN1 zPRRT_
ExpermntNm1 -0.181
zPrmRcgRT_t  0.000  0.000
ExpN1:PRRT_  0.000 -0.002 -0.696
```

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

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

Response: zTargetRecogRT_trim
                                   Chisq Df Pr(>Chisq)
ExperimentName                    0.0609  1    0.80512
zPrimeRecogRT_trim               34.9151  1  3.444e-09 ***
ExperimentName:zPrimeRecogRT_trim  6.2990  1    0.01208 *
---
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)
```

```
Type III Analysis of Variance Table with Satterthwaite's method
                                  Sum Sq Mean Sq NumDF  DenDF F value
ExperimentName                    0.0424  0.0424     1 3907.8  0.0635
zPrimeRecogRT_trim               23.5328 23.5328     1 3955.7 35.2595
ExperimentName:zPrimeRecogRT_trim  4.2041  4.2041     1 3948.8  6.2990
```

```
                                         Pr(>F)
ExperimentName                          0.80107
zPrimeRecogRT_trim                    3.134e-09 ***
ExperimentName:zPrimeRecogRT_trim       0.01212 *
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
>
> # > confint(RTprime_RT_model)
> # Computing profile confidence intervals ...
> #                                         2.5 %      97.5 %
> # .sig01                              0.48676697 0.68438335
> # .sig02                              0.00000000 0.02668791
> # .sigma                              0.79887469 0.83508277
> # (Intercept)                        -0.11479021 0.16260058
> # ExperimentName1                    -0.05728127 0.04423402
> # zPrimeRecogRT_trim                  0.01066445 0.08901858
> # ExperimentName1:zPrimeRecogRT_trim  0.01550415 0.12595953
>
>
```

4.7.1 Model 1

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

```
[1] 0.9930812
```

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

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: zPrimeRecogRT_trim ~ 1 + (1 | Subject) + (1 | Target)
   Data: US_final_z_prime

REML criterion at convergence: 10920
```

```
Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.7548 -0.6992 -0.1521  0.5687  3.6681

Random effects:
 Groups   Name        Variance Std.Dev.
 Target   (Intercept) 0.1673   0.4090
 Subject  (Intercept) 0.0000   0.0000
 Residual             0.8228   0.9071
Number of obs: 4062, groups:  Target, 72; Subject, 57

Fixed effects:
            Estimate Std. Error       df t value Pr(>|t|)
(Intercept)  0.00334    0.05026 70.87856   0.066    0.947
```

```
> VarCorr(primert_model)
```

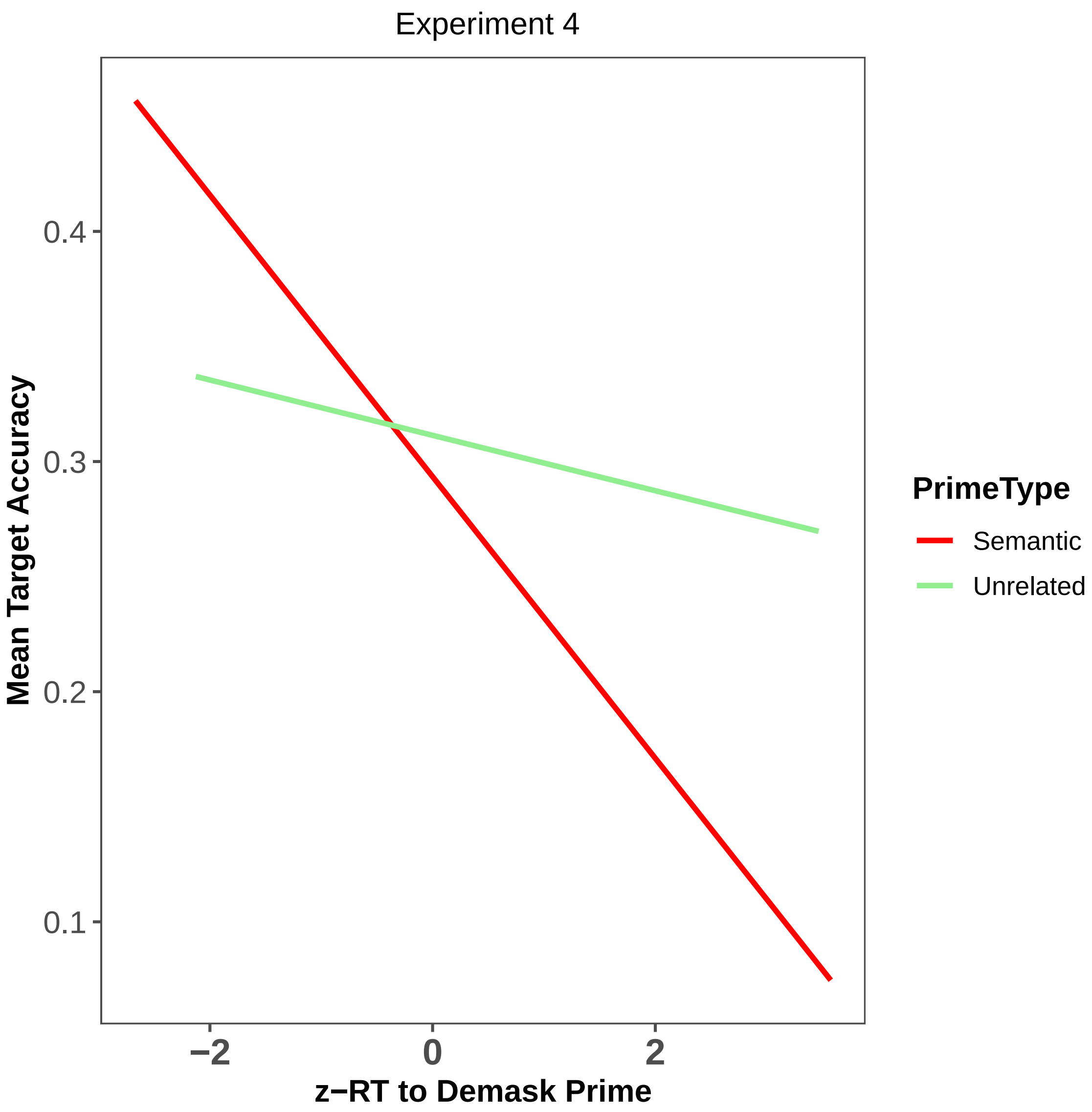
```
 Groups   Name        Std.Dev.
 Target   (Intercept) 0.40897
 Subject  (Intercept) 0.00000
 Residual             0.90708
```

```
> SD_prime <- as.data.frame(VarCorr(primert_model))[3, 5]
> ## now we need to find increments for each prime condition
>
> primert_model_2 <- lmer(data = US_final_z_prime,
+                         zPrimeRecogRT_trim ~ 1 + ExperimentName +
+                       (1|Subject) + (1|Target))
> prime_Inc_1_U <- 0*fixef(primert_model_2)[1]
> prime_Inc_1_R <- 1*fixef(primert_model_2)[2]
> predict_data_U <- with(US_final_z_prime,
+                   data.frame(school=1,
+  zPrimeRecogRT_trim=seq(from=-prime_Inc_1_U-SD_prime,
+         to=-prime_Inc_1_U+SD_prime,
+         by=SD_prime),
+  PrimeCondition = 0))
> predict_data_R <- with(US_final_z_prime,
+                   data.frame(school=1,
+  zPrimeRecogRT_trim=seq(from=-prime_Inc_1_R-SD_prime,
+         to=-prime_Inc_1_R+SD_prime,
+         by=SD_prime),
+  PrimeCondition = 1))
> predict_data = rbind(predict_data_U,
+                      predict_data_R)
> predict_data$ExperimentName = ifelse(predict_data$PrimeCondition == 0,
+                                      "TOT_Unrelated","TOT_Semantic")
> predict_data = predict_data %>%
+   mutate(predicted_values = predict(RTprime_acc_model,
```

```
+           newdata = predict_data, re.form = NA))
> predict_data$prob = exp(predict_data$predicted_values)/(1+exp(predict_data$predicted_v
> predict_data$ExperimentName = ordered(as.factor(as.character(predict_data$ExperimentNa
> predict_data %>%
+   mutate(PrimeType = factor(ExperimentName, levels = unique(ExperimentName),
+                    labels = c("Unrelated", "Semantic")))%>%
+   ggplot(aes(x = zPrimeRecogRT_trim, y = prob,
+              color = PrimeType)) +
+     geom_line(size = 1) +
+     xlab("z-RT to Demask Prime") + ylab ("Mean Target Accuracy")+
+   ggtitle("Experiment 4")+
+ theme_few() +
+   scale_color_manual(values = c("lightgreen","red"))+
+     theme(axis.text = element_text(face = "bold", size = rel(1.2)),
+           axis.title = element_text(face = "bold", size = rel(1.2)),
+           legend.title = element_text(face = "bold", size = rel(1.2)),
+     plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
```

4.7.2 Raw Data 1

```
> library(ggplot2)
> library(ggthemes)
> US_final_z$Accuracy = as.numeric(as.character(US_final_z$Accuracy))
> mainplot = US_final_z_prime %>%
+   mutate(PrimeType = factor(ExperimentName,
+                          levels = unique(ExperimentName),
+                       labels = c("Semantic",
+                            "Unrelated")))%>%
+   ggplot(aes(x =zPrimeRecogRT_trim , y = Accuracy,
+              group = PrimeType, color = PrimeType)) +
+   geom_smooth(method = "glm", se = FALSE, size = 1)+
+     xlab("z-RT to Demask Prime") + ylab ("Mean Target Accuracy")+
+ theme_few() +
+   scale_color_manual(values = c("red", "lightgreen"))+
+     ggtitle(" Experiment 4") +
+   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
>
```



4.8 Model 2

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

```
> summary(primert_model)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: zPrimeRecogRT_trim ~ 1 + (1 | Subject) + (1 | Target)
   Data: US_final_z

REML criterion at convergence: 10709.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.7547 -0.6981 -0.1551  0.5650  3.6481

Random effects:
 Groups   Name        Variance Std.Dev.
 Target   (Intercept) 0.1672   0.4089
 Subject  (Intercept) 0.0000   0.0000
 Residual             0.8228   0.9071
Number of obs: 3983, groups:  Target, 72; Subject, 57

Fixed effects:
             Estimate Std. Error        df t value Pr(>|t|)
(Intercept)  0.001401   0.050292 71.026690   0.028    0.978
```

```
> VarCorr(primert_model)
```

```
 Groups   Name        Std.Dev.
 Target   (Intercept) 0.40888
 Subject  (Intercept) 0.00000
 Residual             0.90706
```

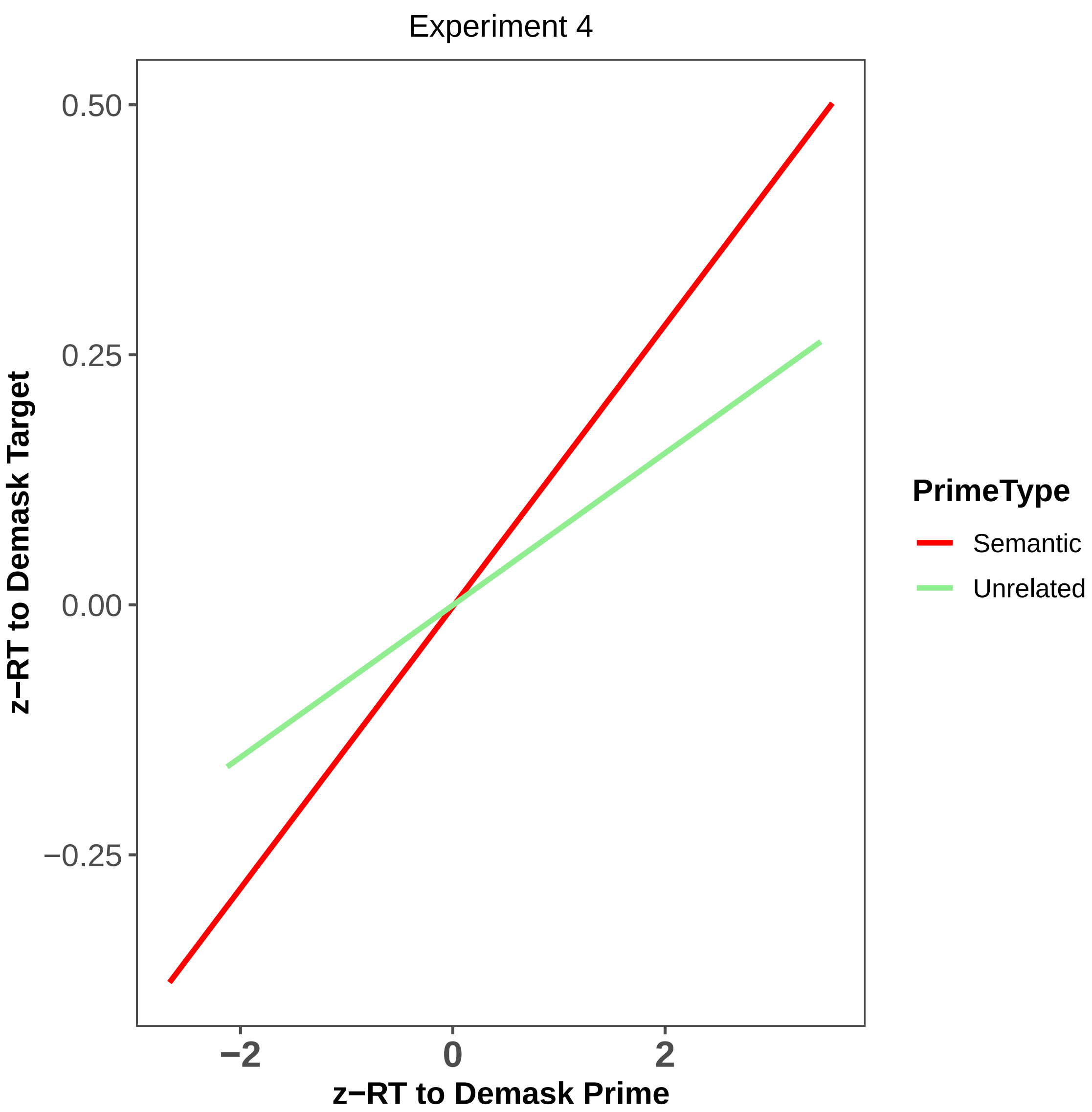
```
> SD_prime ← as.data.frame(VarCorr(primert_model))[3, 5]
> ## now we need to find increments for each prime condition
>
> primert_model_2 ← lmer(data = US_final_z,
+                         zPrimeRecogRT_trim ~ 1 + PrimeCondition +
+                       (1|Subject) + (1|Target))
> prime_Inc_1_U ← 0*fixef(primert_model_2)[1]
> prime_Inc_1_R ← 1*fixef(primert_model_2)[2]
> predict_data_U ← with(US_final_z,
+                    data.frame(school=1,
+  zPrimeRecogRT_trim=seq(from=-prime_Inc_1_U-SD_prime,
+         to=-prime_Inc_1_U+SD_prime,
+         by=SD_prime),
+  PrimeCondition = 0))
> predict_data_R ← with(US_final_z,
+                    data.frame(school=1,
+  zPrimeRecogRT_trim=seq(from=-prime_Inc_1_R-SD_prime,
```

```
+           to=-prime_Inc_1_R+SD_prime,
+           by=SD_prime),
+  PrimeCondition = 1))
> predict_data = rbind(predict_data_U,
+                      predict_data_R)
> predict_data$ExperimentName = ifelse(predict_data$PrimeCondition == 0,
+                                      "TOT_Unrelated","TOT_Semantic")
> predict_data = predict_data %>%
+   mutate(predicted_values = predict(RTprime_RT_model,
+          newdata = predict_data, re.form = NA))
> predict_data$ExperimentName = ordered(as.factor(as.character(predict_data$ExperimentNa
> predict_data %>%
+   mutate(PrimeType = factor(ExperimentName, levels = unique(ExperimentName),
+                       labels = c("Unrelated", "Semantic")))%>%
+   ggplot(aes(x = zPrimeRecogRT_trim, y = predicted_values,
+              color = PrimeType)) +
+     geom_line(size = 1) +
+     xlab("z-RT to Demask Prime") + ylab ("z-RT to Demask Target")+
+   ggtitle("Experiment 4")+
+ theme_few() +
+   scale_color_manual(values = c("lightgreen","red"))+
+     theme(axis.text = element_text(face = "bold", size = rel(1.2)),
+           axis.title = element_text(face = "bold", size = rel(1.2)),
+           legend.title = element_text(face = "bold", size = rel(1.2)),
+     plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
>
```

4.8.1 Raw Data 2

```
> library(ggplot2)
> library(ggthemes)
> mainplot2 = US_final_z %>%
+   mutate(PrimeType = factor(ExperimentName,
+                             levels = unique(ExperimentName),
+                          labels = c("Semantic",
+                              "Unrelated")))%>%
+   ggplot(aes(x =zPrimeRecogRT_trim , y = zTargetRecogRT_trim,
+              group = PrimeType, color = PrimeType)) +
+   geom_smooth(method = "glm", se = FALSE, size = 1)+
+     xlab("z-RT to Demask Prime") + ylab ("z-RT to Demask Target")+
+  # ylim(-0.5,0.5)+
+ theme_few() +
+    scale_color_manual(values = c("red", "lightgreen"))+
+     ggtitle("Experiment 4") +
+   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)),
```

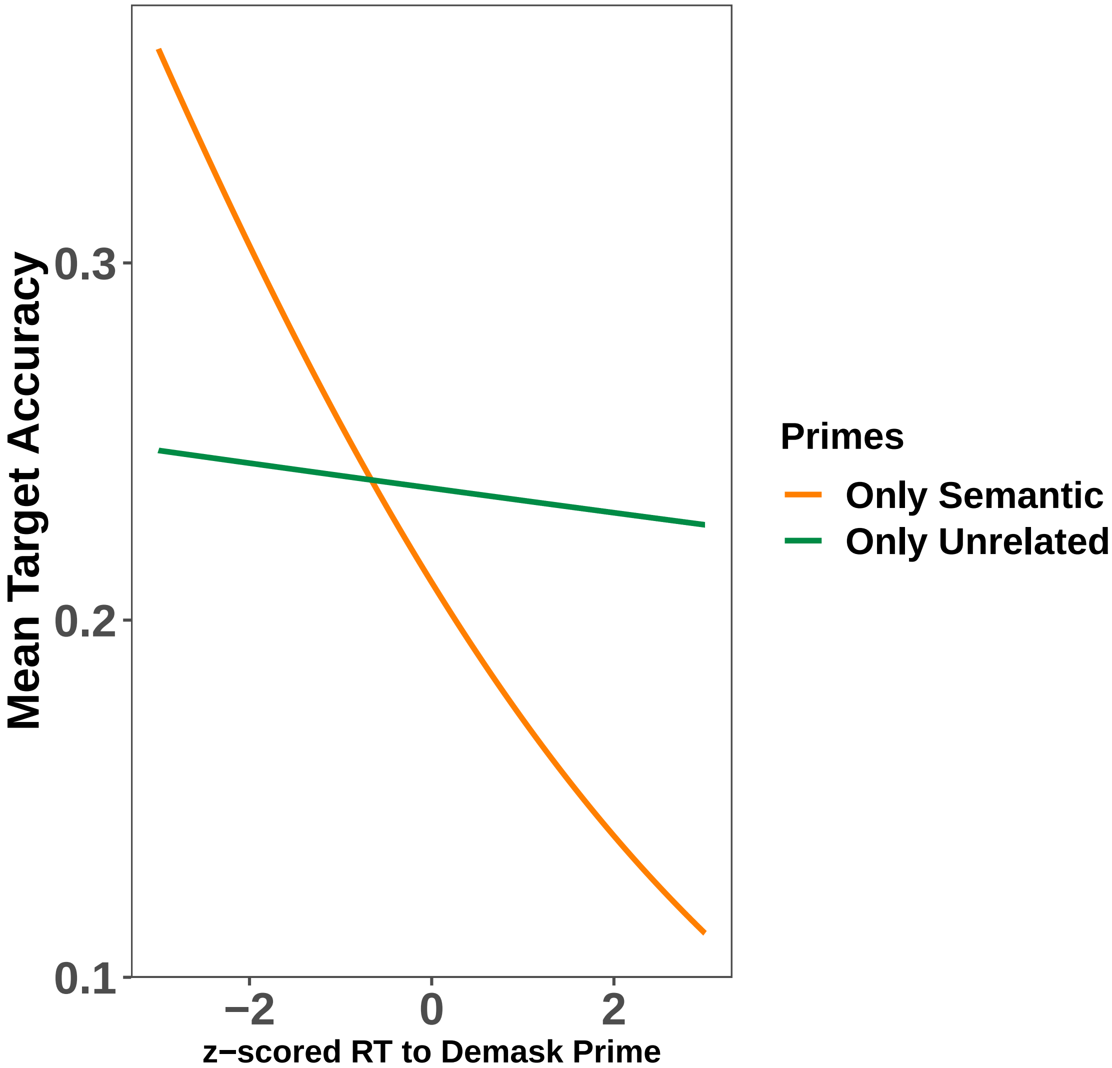
```
+           plot.title = element_text(hjust = .5, size = rel(1)),
+           axis.text.x = element_text(face = "bold", size = rel(1.2)))
> mainplot2
>
```



4.8.2 Model Plot 1

```
> library(ggplot2)
> library(ggthemes)
> library(dplyr)
> fixed.frame ←
```

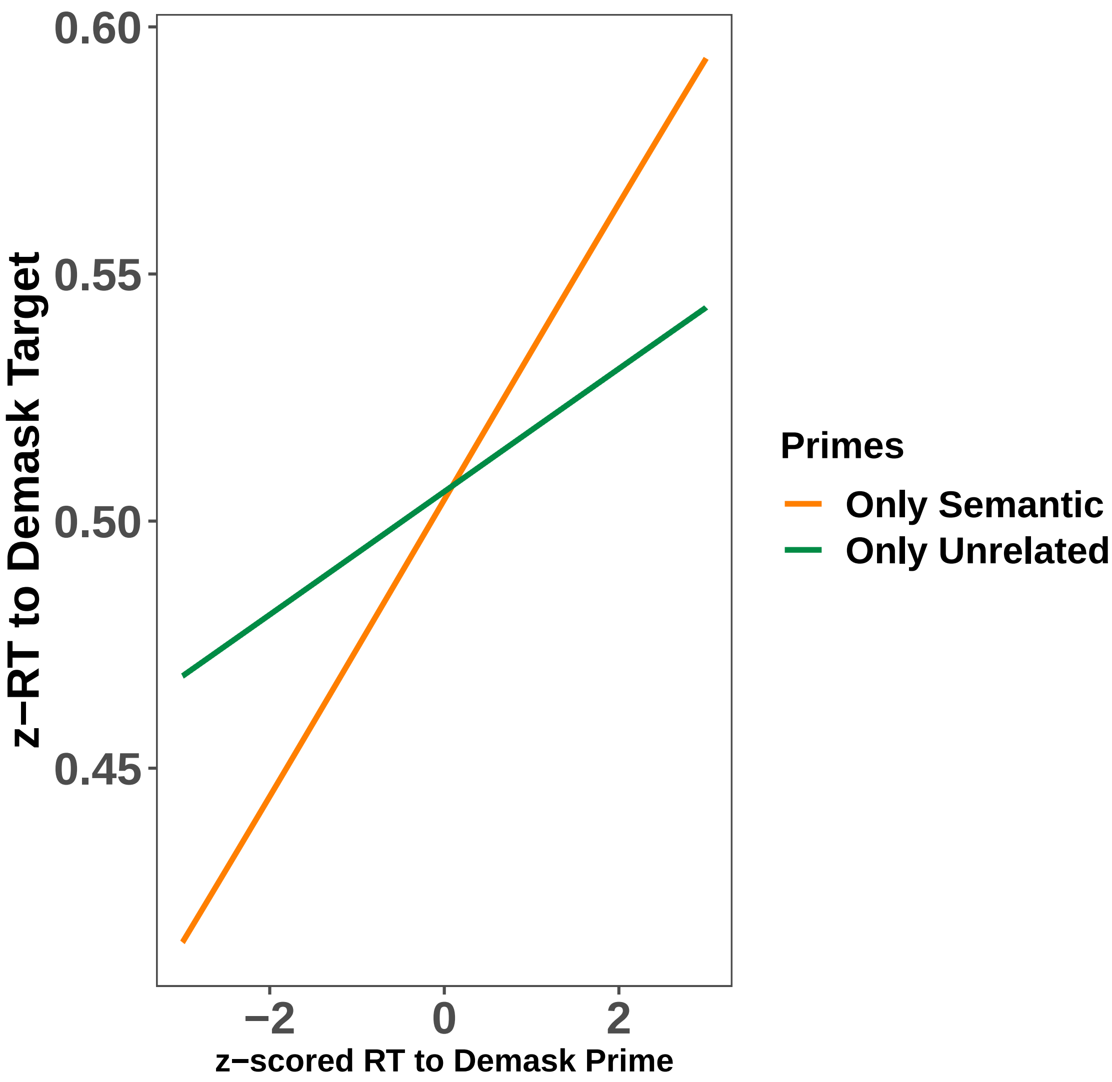
```
+   data.frame(
+     expand.grid(
+       ExperimentName = c("TOT_Semantic", "TOT_Unrelated"),
+       zPrimeRecogRT_trim = seq(-3, 3, 0.001)))
> fixed.frame$pred = predict(RTprime_acc_model, newdata = fixed.frame, re.form = NA, typ
> fixed.frame$prob = exp(fixed.frame$pred)/(1+exp(fixed.frame$pred))
> fixed.frame %>%
+   mutate(Primes = factor(ExperimentName,
+                          levels = unique(ExperimentName),
+                       labels = c("Only Semantic",
+                              "Only Unrelated")))%>%
+   ggplot(aes(x =zPrimeRecogRT_trim , y = prob,
+              group = Primes, color = Primes)) +
+ geom_line(size = 1)+
+       xlab("z-scored RT to Demask Prime") + ylab ("Mean Target Accuracy")+
+   ggtitle("")+
+ theme_few() +
+   scale_color_manual(values = c("darkorange1", "springgreen4"))+
+     theme(axis.text = element_text(face = "bold", size = rel(1.4)),
+          axis.title.y = element_text(face = "bold", size = rel(1.4)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1.2)),
+           plot.title = element_text(face = "bold",
+                   size = rel(1.4), hjust = .5),
+          legend.text = element_text(face = "bold", size = rel(1.2)),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
```



4.8.3 Model Plot 2

```
> library(ggplot2)
> library(ggthemes)
> library(dplyr)
> fixed.frame ←
+   data.frame(
+     expand.grid(
+       ExperimentName = c("TOT_Semantic", "TOT_Unrelated"),
+       zPrimeRecogRT_trim = seq(-3, 3, 0.001)))
> fixed.frame$pred = predict(RTprime_RT_model, newdata = fixed.frame, re.form = NA, type
```

```
> fixed.frame$prob = exp(fixed.frame$pred)/(1+exp(fixed.frame$pred))
> fixed.frame %>%
+   mutate(Primes = factor(ExperimentName,
+                          levels = unique(ExperimentName),
+                       labels = c("Only Semantic",
+                            "Only Unrelated")))%>%
+   ggplot(aes(x =zPrimeRecogRT_trim , y = prob,
+              group = Primes, color = Primes)) +
+ geom_line(size = 1)+
+       xlab("z-scored RT to Demask Prime") + ylab ("z-RT to Demask Target")+
+   ggtitle("")+
+ theme_few() +
+   scale_color_manual(values = c("darkorange1", "springgreen4"))+
+     theme(axis.text = element_text(face = "bold", size = rel(1.4)),
+          axis.title.y = element_text(face = "bold", size = rel(1.4)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1.2)),
+           plot.title = element_text(face = "bold",
+                   size = rel(1.4), hjust = .5),
+          legend.text = element_text(face = "bold", size = rel(1.2)),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
```



4.9 Effect on Target Def RT

```
>
> ## not reliable: very noisy data
> # library(lme4)
> # contrasts(US_final_z_targetdef$PrimeCondition) = contr.treatment(2, base = 1)
> # RTprime_targetdefRT_model_1 = lmer(data = US_final_z_targetdef,
> #                        zTargetRT_trim ~ PrimeCondition +
> #                                 (1|Subject) + (1|Stimuli1))
> # summary(RTprime_targetdefRT_model_1)
```

```
> # car::Anova(RTprime_targetdefRT_model_1)
> #
> #
> # RTprime_targetdefRT_model_2 = lmer(data = US_final_z_targetdef,
> #                     zTargetRT_trim ~ PrimeAcc*PrimeCondition +
> #                             (1|Subject) + (1|Stimuli1))
> # summary(RTprime_targetdefRT_model_2)
> # car::Anova(RTprime_targetdefRT_model_2)
> #
> # RTprime_targetdefRT_model_3 = lmer(data = US_final_z_targetdef,
> #             zTargetRT_trim ~ PrimeAcc*zPrimeRecogRT_trim*PrimeCondition +
> #                             (1|Subject) + (1|Stimuli1))
> # summary(RTprime_targetdefRT_model_3)
> # car::Anova(RTprime_targetdefRT_model_3)
> #
> # anova(RTprime_targetdefRT_model_1, RTprime_targetdefRT_model_2)
> #
> # RTprime_targetdefRT_model_4 = lmer(data = US_final_z_targetdef,
> #             zTargetRT_trim ~ PrimeAcc +
> #                             (1|Subject) + (1|Stimuli1))
> # summary(RTprime_targetdefRT_model_4)
> # car::Anova(RTprime_targetdefRT_model_4)
> # anova(RTprime_targetdefRT_model_4, RTprime_targetdefRT_model_2)
> #
> # RTprime_targetdefRT_model_5 = lmer(data = US_final_z_targetdef,
> #             zTargetRT_trim ~ zPrimeRecogRT_trim*PrimeCondition +
> #                             (1|Subject) + (1|Stimuli1))
> # summary(RTprime_targetdefRT_model_5)
> # car::Anova(RTprime_targetdefRT_model_5)
> # anova(RTprime_targetdefRT_model_5, RTprime_targetdefRT_model_2)
>
```

4.9.1 Model 1

```
> # targetdefRT_rmisc = Rmisc::summarySE(US_final_z_targetdef,
> #                                   measurevar = "zTargetRT_trim",
> #                                   groupvars = c("PrimeCondition"))
> #
> # ggplot(targetdefRT_rmisc, aes(x = PrimeCondition, y = zTargetRT_trim,
> #                         fill = PrimeCondition))+
> #  geom_bar(stat = "identity", position = "dodge", width = 0.7,
> #           color= "black")+
> #   geom_errorbar(aes(ymin=zTargetRT_trim - se, ymax=zTargetRT_trim + se),
> #                 width=.2, color = "gray26",
> #                 position = position_dodge(0.7))+
> #  theme_few()+
> #     xlab("Prime Condition") + ylab("z-RT") +
```

```
> #    ggtitle("YA: Effect of Prime on RT to Retrieving Target")  +
> #    scale_fill_gdocs()+
> #     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)))
```

4.9.2 Model 2

```
> # targetdefRT_rmisc2 = Rmisc::summarySE(US_final_z_targetdef,
> #                                     measurevar = "zTargetRT_trim",
> #                                     groupvars = c("PrimeAcc",
> #                                                  "PrimeCondition"))
> # targetdefRT_rmisc2$PrimeAcc = as.factor(targetdefRT_rmisc2$PrimeAcc)
> # ggplot(targetdefRT_rmisc2, aes(x = PrimeCondition, y = zTargetRT_trim,
> #                    group = PrimeAcc, fill = PrimeAcc))+
> #  geom_bar(stat = "identity", position = "dodge", width = 0.7,
> #            color= "black")+
> #    geom_errorbar(aes(ymin=zTargetRT_trim - se, ymax=zTargetRT_trim + se),
> #               width=.2, color = "gray26",
> #               position = position_dodge(0.7))+
> #  theme_few()+
> #   xlab("Prime Condition") + ylab("z-RT to Retrieve Target") +
> #   ggtitle("YA: Effect of Prime on Retrieving Target")  +
> #   scale_fill_wsj()+
> #     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)))
```

4.9.3 Model 5

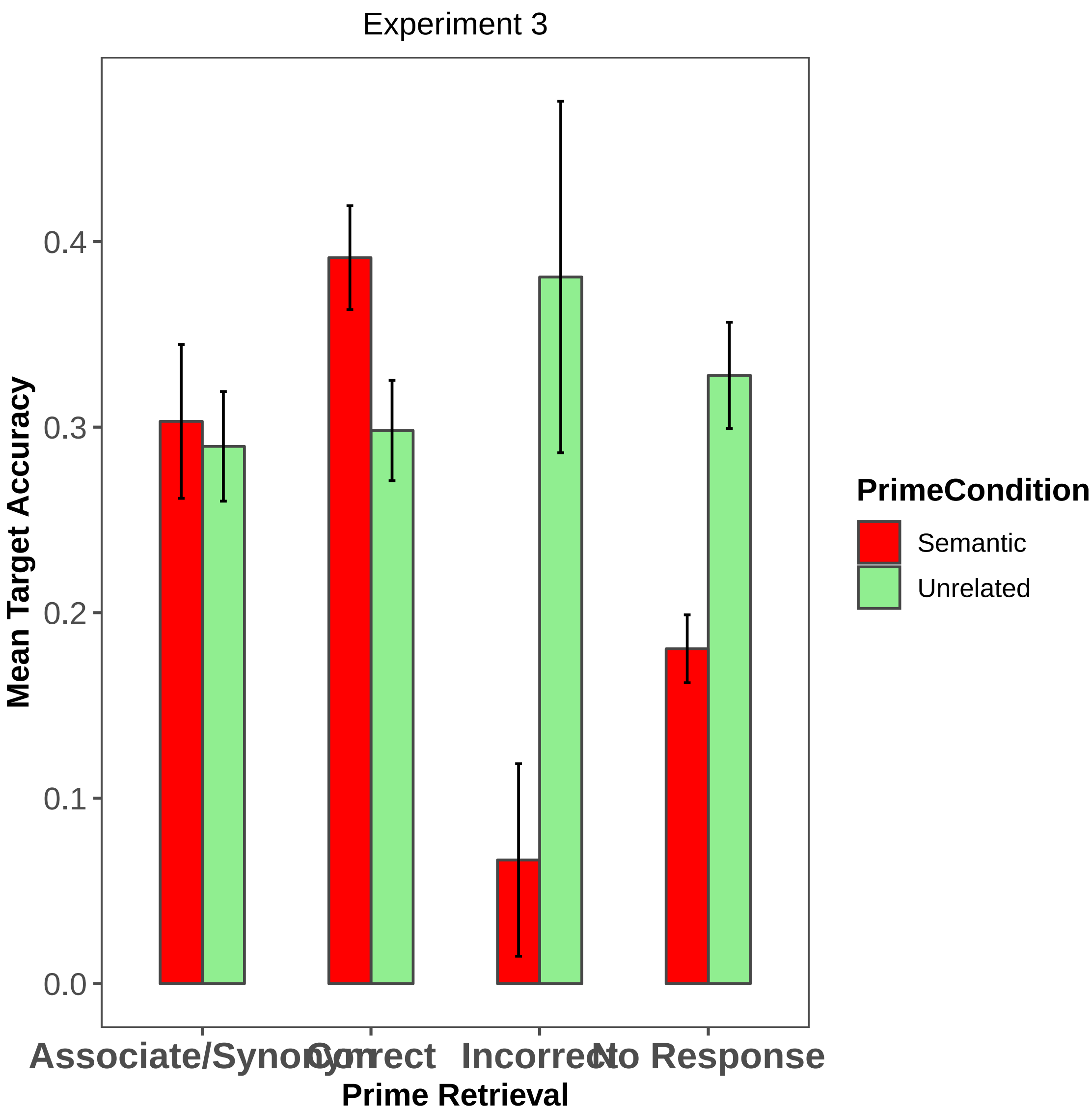
```
> # US_final_z_targetdef %>%
> #   ggplot(aes(x = zPrimeRecogRT_trim, y = zTargetRT_trim,
> #               group = PrimeCondition, color = PrimeCondition)) +
> #   geom_smooth(method = "lm", se = FALSE, size = 1)+
> #   facet_wrap(~PrimeCondition, nrow = 2)+
> #     xlab("z-RT to Demask Prime") + ylab ("z-RT to Retrieve Target")+
> #   ggtitle("YA: Effect of Prime on Retrieving Target")  +
> # theme_hc() +
> # scale_color_manual(values = c( "darkorange1", "red",
> #                                 "dodgerblue3", "springgreen3"))+
```

```
> #    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)))
```

5 Response Analysis

5.1 All Responses

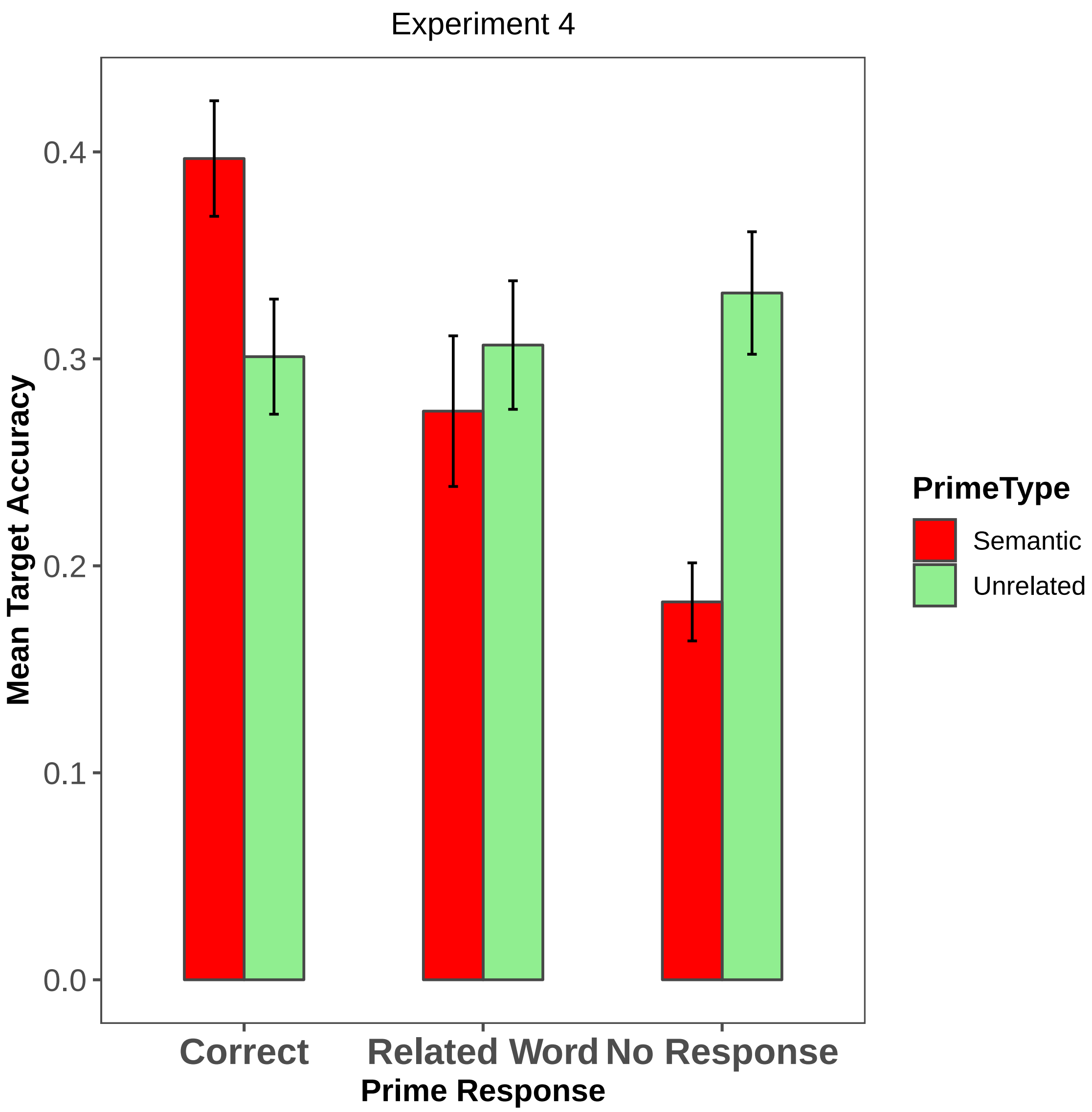
```
> US_responses = read.csv("US_TOT_Responses.csv", header = TRUE, sep = ",")
> US_responses$AllResponse = ifelse(US_responses$PrimeRespType %in%
+                               c("Associate", "Synonym"), "Associate/Synonym",
+                             ifelse(US_responses$PrimeRespType == "NoResponse",
+                                    "No Response",
+                          ifelse(US_responses$PrimeRespType == "Correct","Correct",
+                                 "Incorrect")))
> US_responses_subject = group_by(US_responses, Subject, PrimeCondition, AllResponse) %>%
+   summarize_at(vars(TargetFirstResp_ACC), mean)
> ret_figure = Rmisc::summarySE(US_responses_subject,
+                         measurevar = "TargetFirstResp_ACC",
+                      groupvars = c("PrimeCondition", "AllResponse"))
> library(ggplot2)
> library(ggthemes)
> library(dplyr)
> ret_figure  %>%
+    ggplot(aes(x = AllResponse, y = TargetFirstResp_ACC,
+                          group =PrimeCondition ,
+                          fill = PrimeCondition)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+    geom_errorbar(aes(ymin = TargetFirstResp_ACC - se,
+                      ymax = TargetFirstResp_ACC + 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 3") +
+   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 3-group Responses

```
> #US_responses = read.csv("E4_TOT_Responses.csv", header = TRUE, sep = ",")
>
> US_responses = US_final_z
> US_responses$Response = ifelse(US_responses$PrimeRespType %in%
+                                   c("Associate", "Incorrect", "Synonym"), "Related Word
+                                 ifelse(US_responses$PrimeRespType == "NoResponse",
+                                        "No Response","Correct"))
> US_responses$Response = ordered(as.factor(as.character(US_responses$Response)),
```

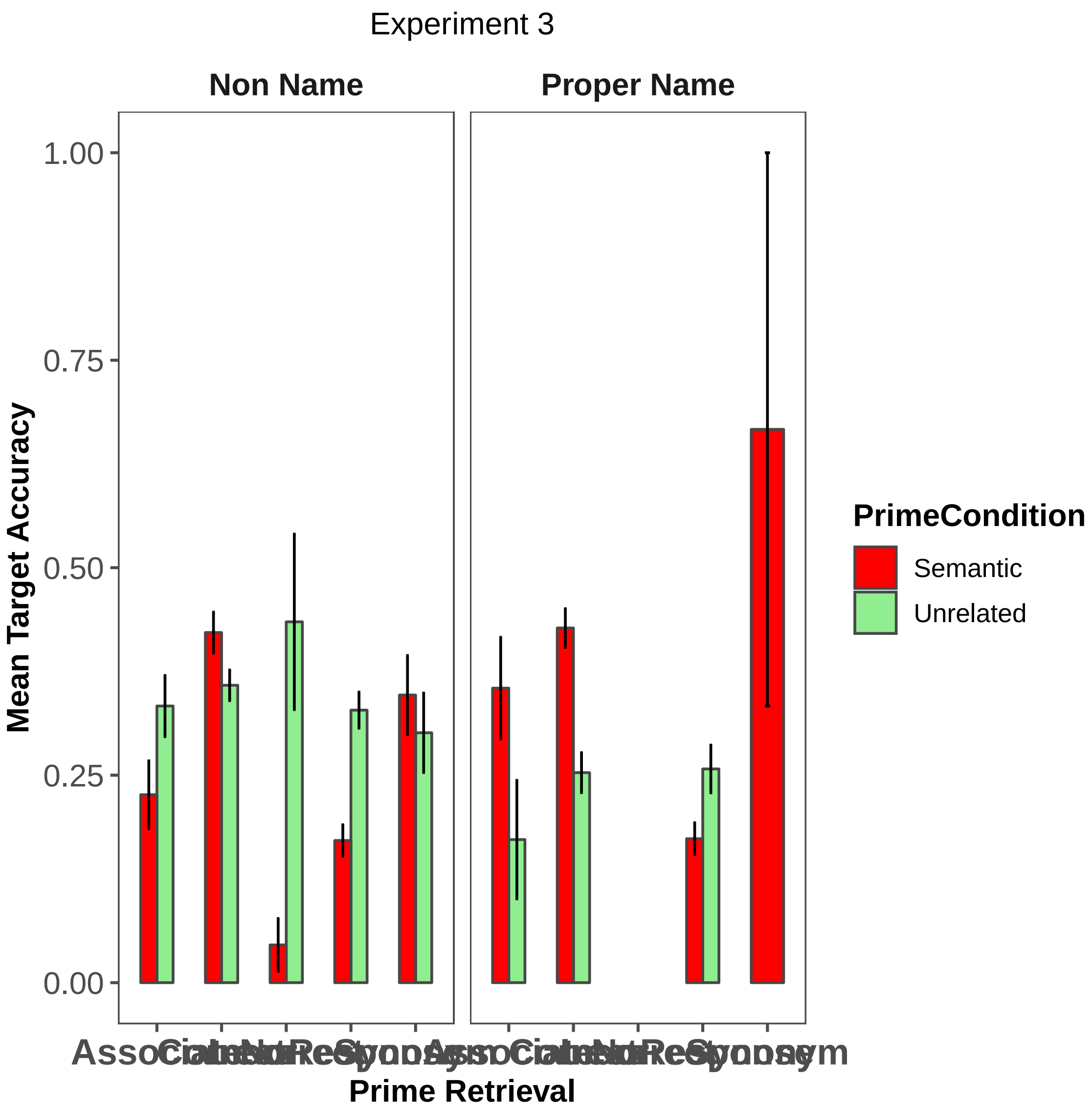
```
+                            levels = c("Correct",  "Related Word", "No Response"))
> US_responses_subject = group_by(US_responses, Subject, PrimeCondition, Response) %>%
+   summarize_at(vars(Accuracy), mean)
> ret_figure = Rmisc::summarySE(US_responses_subject,
+                          measurevar = "Accuracy",
+                      groupvars = c( "PrimeCondition", "Response"))
> library(ggplot2)
> library(ggthemes)
> library(dplyr)
> ret_figure  %>%
+   mutate(PrimeType = factor(PrimeCondition, levels = unique(PrimeCondition),
+                          labels = c( "Semantic", "Unrelated")))%>%
+    ggplot(aes(x = Response, y = Accuracy,
+                                group =PrimeType ,
+                                fill = PrimeType)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+    geom_errorbar(aes(ymin = Accuracy - se,
+                           ymax = Accuracy + se),
+                     width=.08, position=position_dodge(.5)) +
+   theme_few()+
+   scale_fill_manual(values = c(  "red",
+                                      "lightgreen"))+
+   xlab("Prime Response") + ylab("Mean Target Accuracy") +
+ ggtitle("Experiment 4") +
+   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)))
```



5.3 POS split Responses

```
> ret_figure = Rmisc::summarySE(US_responses,
+                     measurevar = "Accuracy",
+                   groupvars = c("Prime_POS", "PrimeCondition", "PrimeRespType"))
> library(ggplot2)
> library(ggthemes)
> library(dplyr)
> ret_figure  %>%
+    ggplot(aes(x = PrimeRespType, y = Accuracy,
```

```
+                            group =PrimeCondition ,
+                            fill = PrimeCondition)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+    geom_errorbar(aes(ymin = Accuracy - se,
+                       ymax = Accuracy + se),
+                  width=.08, position=position_dodge(.5)) +
+   theme_few()+
+ facet_wrap(~Prime_POS)+
+  scale_fill_manual(values = c(  "red",
+                                 "lightgreen"))+
+   xlab("Prime Retrieval") + ylab("Mean Target Accuracy") +
+ ggtitle("Experiment 3") +
+   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)),
+         strip.text.x = element_text(face = "bold", size = rel(1.2)))
```



5.4 LME

```
> US_responses$Response = as.factor(US_responses$Response)
> contrasts(US_responses$Response) = contr.treatment(3, base = 1)
> contrasts(US_responses$PrimeCondition) = contr.treatment(2, base = 2)
> TOTFeedback_hlm2 = glmer(data = US_responses,
+                               Accuracy ~ PrimeCondition*Response +
+                        (1|Subject) + (1|Target), family = "binomial",
+                          control=glmerControl(optimizer="bobyqa",
+             optCtrl=list(maxfun=100000)))
```

```
> summary(TOTFeedback_hlm2)
```

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

     AIC      BIC   logLik deviance df.resid
  3855.4   3905.7  -1919.7   3839.4     3975

Scaled residuals:
    Min      1Q  Median      3Q     Max
-5.3859 -0.5251 -0.2855  0.5081  7.2519

Random effects:
 Groups  Name        Variance Std.Dev.
 Target  (Intercept) 2.0625   1.4361
 Subject (Intercept) 0.6108   0.7815
Number of obs: 3983, groups:  Target, 72; Subject, 57

Fixed effects:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)               -1.11575    0.24060  -4.637 3.53e-06 ***
PrimeCondition1            0.18171    0.24171   0.752    0.452
Response2                 -0.14247    0.18262  -0.780    0.435
Response3                 -0.01656    0.14197  -0.117    0.907
PrimeCondition1:Response2 -0.06998    0.25765  -0.272    0.786
PrimeCondition1:Response3 -0.83335    0.20227  -4.120 3.79e-05 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Correlation of Fixed Effects:
            (Intr) PrmCn1 Rspns2 Rspns3 PC1:R2
PrimeCndtn1 -0.487
Response2   -0.182  0.174
Response3   -0.246  0.238  0.334
PrmCndt1:R2  0.127 -0.267 -0.709 -0.243
PrmCndt1:R3  0.163 -0.344 -0.213 -0.681  0.336
```

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

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

Response: Accuracy
                         Chisq Df Pr(>Chisq)
PrimeCondition          0.2648  1  0.6068164
```

```
Response                    15.7945   2   0.0003718 ***
PrimeCondition:Response 18.3686   2   0.0001026 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

5.5 Contrasts

```
> ## first reproduce means
>
> means_contrasts = matrix(c(1, 0, 0, 0, 0, 0, # UC
+                       1, 1, 0, 0, 0, 0, # SC
+                       1, 0, 1, 0, 0, 0, # UO
+                       1, 1, 1, 0, 1, 0, # SO
+                       1, 0, 0, 1, 0, 0, # UN
+                       1, 1, 0, 1, 0, 1) , nrow = 6, # SN
+ ncol = 6, byrow = TRUE)
> # Give the weight matrix some meaningful row names.
> rownames(means_contrasts) ← c("UC", "SC", "UO", "SO", "UN", "SN")
> model_means = means_contrasts %*% fixef(TOTFeedback_hlm2)
> colnames(model_means) = "Logits"
> knitr::kable(model_means)
```

```
|   |     Logits|
|:--|----------:|
|UC | -1.1157547|
|SC | -0.9340412|
|UO | -1.2582280|
|SO | -1.1464959|
|UN | -1.1323190|
|SN | -1.7839508|
```

```
> library(multcomp)
> glht_means ← glht(TOTFeedback_hlm2, linfct = means_contrasts,
+                    alternative = "two.sided", rhs = 0)
> summary(glht_means, adjusted(type = "holm"))
```

```
         Simultaneous Tests for General Linear Hypotheses

Fit: glmer(formula = Accuracy ~ PrimeCondition * Response + (1 | Subject) +
    (1 | Target), data = US_responses, family = "binomial", control = glmerControl(optim
    optCtrl = list(maxfun = 1e+05)))

Linear Hypotheses:
        Estimate Std. Error z value Pr(>|z|)
UC == 0   -1.1158     0.2406  -4.637 1.76e-05 ***
SC == 0   -0.9340     0.2443  -3.823 0.000132 ***
UO == 0   -1.2582     0.2743  -4.587 1.80e-05 ***
```

```
SO == 0   -1.1465      0.2732   -4.197 5.42e-05 ***
UN == 0   -1.1323      0.2474   -4.576 1.80e-05 ***
SN == 0   -1.7840      0.2518   -7.085 8.37e-12 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
(Adjusted p values reported -- holm method)
```

```
> ## create contrast matrix that needs to be multiplied
> contrast_matrix ← matrix(c(1, -1, 0, 0, 0, 0,
+                            0, 0, 1, -1, 0, 0,
+                            0, 0, 0, 0, 1, -1),
+                   nrow = 3, ncol = 6, byrow = TRUE)
> rownames(contrast_matrix) ← c("SC vs UC",
+                               "SO vs UO",
+                               "SN vs UN")
> matrix_for_glht ←contrast_matrix %*% means_contrasts
> matrix_for_glht
```

```
         [,1] [,2] [,3] [,4] [,5] [,6]
SC vs UC    0   -1    0    0    0    0
SO vs UO    0   -1    0    0   -1    0
SN vs UN    0   -1    0    0    0   -1
```

```
> glht_model1 ← multcomp::glht(TOTFeedback_hlm2,
+                             linfct = matrix_for_glht,
+                  alternative = "two.sided", rhs = 0)
> summary(glht_model1)
```

```
         Simultaneous Tests for General Linear Hypotheses

Fit: glmer(formula = Accuracy ~ PrimeCondition * Response + (1 | Subject) +
    (1 | Target), data = US_responses, family = "binomial", control = glmerControl(optim
    optCtrl = list(maxfun = 1e+05)))

Linear Hypotheses:
              Estimate Std. Error z value Pr(>|z|)
SC vs UC == 0  -0.1817     0.2417  -0.752   0.7692
SO vs UO == 0  -0.1117     0.3026  -0.369   0.9614
SN vs UN == 0   0.6516     0.2563   2.542   0.0282 *
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
(Adjusted p values reported -- single-step method)
```

5.6 Specific Comparisons

```
> responses_correct = US_responses %>% filter(Response == "Correct")
```

```
> ## get an estimate of semantic and unrelated per subject: this is between subjects her
>
> responses_correct_sub = group_by(responses_correct, Subject, PrimeCondition) %>%
+   summarise_at(vars(Accuracy), mean)
> responses_correct_sub_semantic = responses_correct_sub %>%
+   filter(PrimeCondition == "Semantic")
> responses_correct_sub_unrelated = responses_correct_sub %>%
+   filter(PrimeCondition == "Unrelated")
> t.test(responses_correct_sub_semantic$Accuracy, responses_correct_sub_unrelated$Accura
+        paired = FALSE)
```

```
        Welch Two Sample t-test

data:  responses_correct_sub_semantic$Accuracy and responses_correct_sub_unrelated$Accur
t = 2.4303, df = 54.974, p-value = 0.01838
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.01679081 0.17467036
sample estimates:
mean of x mean of y
0.3967913 0.3010607
```

```
> responses_other = US_responses %>% filter(Response == "Related Word")
> ## get an estimate of semantic and unrelated per subject: this is between subjects her
>
> responses_other_sub = group_by(responses_other, Subject, PrimeCondition) %>%
+   summarise_at(vars(Accuracy), mean)
> responses_other_sub_semantic = responses_other_sub %>%
+   filter(PrimeCondition == "Semantic")
> responses_other_sub_unrelated = responses_other_sub %>%
+   filter(PrimeCondition == "Unrelated")
> t.test(responses_other_sub_semantic$Accuracy, responses_other_sub_unrelated$Accuracy,
+        paired = FALSE)
```

```
        Welch Two Sample t-test

data:  responses_other_sub_semantic$Accuracy and responses_other_sub_unrelated$Accuracy
t = -0.66745, df = 53.352, p-value = 0.5074
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.12786469  0.06400637
sample estimates:
mean of x mean of y
0.2747373 0.3066665
```

```
> responses_none = US_responses %>% filter(Response == "No Response")
> ## get an estimate of semantic and unrelated per subject: this is between subjects her
>
```

```
> responses_none_sub = group_by(responses_none, Subject, PrimeCondition) %>%
+   summarise_at(vars(Accuracy), mean)
> responses_none_sub_semantic = responses_none_sub %>%
+   filter(PrimeCondition == "Semantic")
> responses_none_sub_unrelated = responses_none_sub %>%
+   filter(PrimeCondition == "Unrelated")
> t.test(responses_none_sub_semantic$Accuracy, responses_none_sub_unrelated$Accuracy,
+        paired = FALSE)
```

```
        Welch Two Sample t-test

data:  responses_none_sub_semantic$Accuracy and responses_none_sub_unrelated$Accuracy
t = -4.2526, df = 47.283, p-value = 9.892e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.2198389 -0.0786556
sample estimates:
mean of x mean of y
0.1825733 0.3318206
```