Repeated Lexical Retrieval: Experiment 4

Abhilasha Kumar

June 6, 2019

1 Reading the Data File

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

```
> TOTFeedback = read.csv("TOTwoFeedback_FINAL.csv",
+                        header = TRUE, sep = ",")
> head(TOTFeedback[,c(1,6,7,11)])
```

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

2 Conditional Target Accuracy

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

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

3 ANOVA

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

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

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

REML criterion at convergence: -231.2

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.4508 -0.6614 -0.0026  0.6606  3.4508

Random effects:
 Groups   Name        Variance Std.Dev.
 Subject  (Intercept) 0.00000  0.0000
 Residual             0.02526  0.1589
Number of obs: 318, groups:  Subject, 40

Fixed effects:
                                                    Estimate Std. Error
(Intercept)                                          0.79014    0.02513
PrimeConditionUnrelated                             -0.15840    0.03554
CuedRecallAcc1                                      -0.19168    0.03554
TargetAccuracy1                                     -0.56924    0.03600
PrimeConditionUnrelated:CuedRecallAcc1               0.28248    0.05025
PrimeConditionUnrelated:TargetAccuracy1              0.30576    0.05058
CuedRecallAcc1:TargetAccuracy1                       0.37232    0.05058
PrimeConditionUnrelated:CuedRecallAcc1:TargetAccuracy1 -0.55391    0.07130
                                                    t value
(Intercept)                                          31.446
PrimeConditionUnrelated                              -4.458
CuedRecallAcc1                                       -5.394
```

```
TargetAccuracy1                                               -15.812
PrimeConditionUnrelated:CuedRecallAcc1                          5.621
PrimeConditionUnrelated:TargetAccuracy1                         6.045
CuedRecallAcc1:TargetAccuracy1                                  7.360
PrimeConditionUnrelated:CuedRecallAcc1:TargetAccuracy1         -7.768

Correlation of Fixed Effects:
            (Intr) PrmCnU CdRcA1 TrgtA1 PrCU:CRA1 PCU:TA CRA1:T
PrmCndtnUnr -0.707
CudRcllAcc1 -0.707  0.500
TrgtAccrcy1 -0.698  0.494  0.494
PrmCnU:CRA1  0.500 -0.707 -0.707 -0.349
PrmCndU:TA1  0.497 -0.703 -0.351 -0.712  0.497
CdRclA1:TA1  0.497 -0.351 -0.703 -0.712  0.497      0.506
PCU:CRA1:TA -0.352  0.498  0.498  0.505 -0.705     -0.709 -0.709
```

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

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

Response: prop
                                            Chisq Df Pr(>Chisq)
PrimeCondition                             0.0438  1   0.834255
CuedRecallAcc                              0.0490  1   0.824865
TargetAccuracy                           424.7984  1  < 2.2e-16 ***
PrimeCondition:CuedRecallAcc               0.0423  1   0.837040
PrimeCondition:TargetAccuracy              0.5734  1   0.448924
CuedRecallAcc:TargetAccuracy               6.8856  1   0.008689 **
PrimeCondition:CuedRecallAcc:TargetAccuracy 60.3471  1  7.952e-15 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
>
```

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

4 Conditional Figure

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

```
+                     labels = c("Failed Recall",
+                                "Successful Recall")),
+                     `Target Retrieval` = factor(TargetAccuracy,
+                           levels = unique(TargetAccuracy),
+                        labels = c("Failed Target Retrieval",
+                             "Successful Target Retrieval")))%>%
+ ggplot(aes(x = Recall, y = prop,
+            fill = `Target Retrieval`, group = `Target Retrieval`))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.7)+
+   geom_errorbar(aes(ymin=prop - ci, ymax=prop + ci),
+              width=.2, color = "gray26",
+              position = position_dodge(0.7))+
+   facet_wrap(~PrimeCondition)+
+  theme_few()+
+   scale_fill_wsj()+
+     xlab("Cued Recall Accuracy") + ylab("Mean Proportion of Trials") +
+   ggtitle("Target Retrieval Accuracy
+           as a function of Cued Recall Accuracy")  +
+    theme(axis.text = element_text(face = "bold", size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+           plot.title = element_text(face = "bold",
+                   size = rel(1.2), hjust = .5),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
> condfigure_plot
```

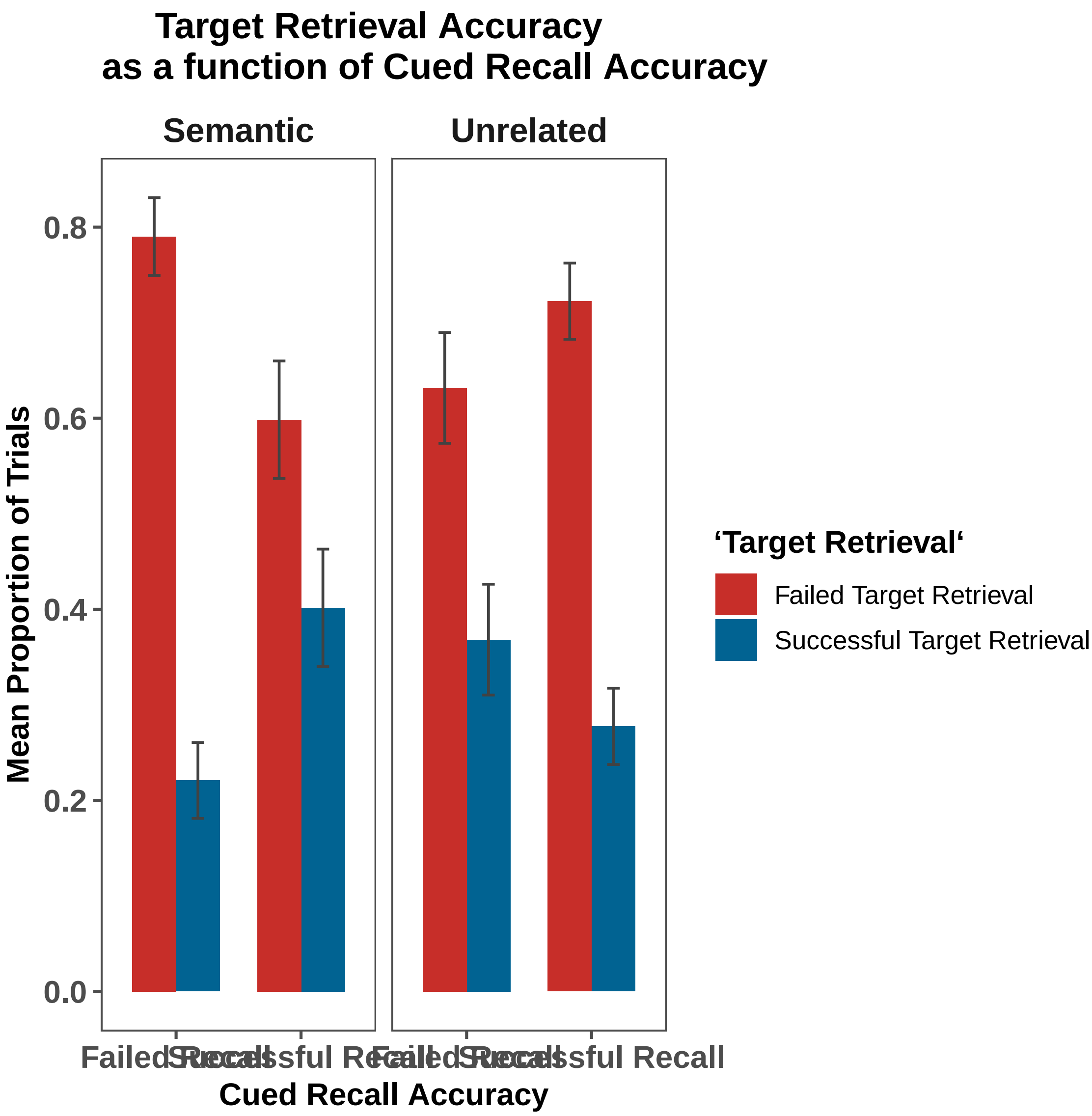
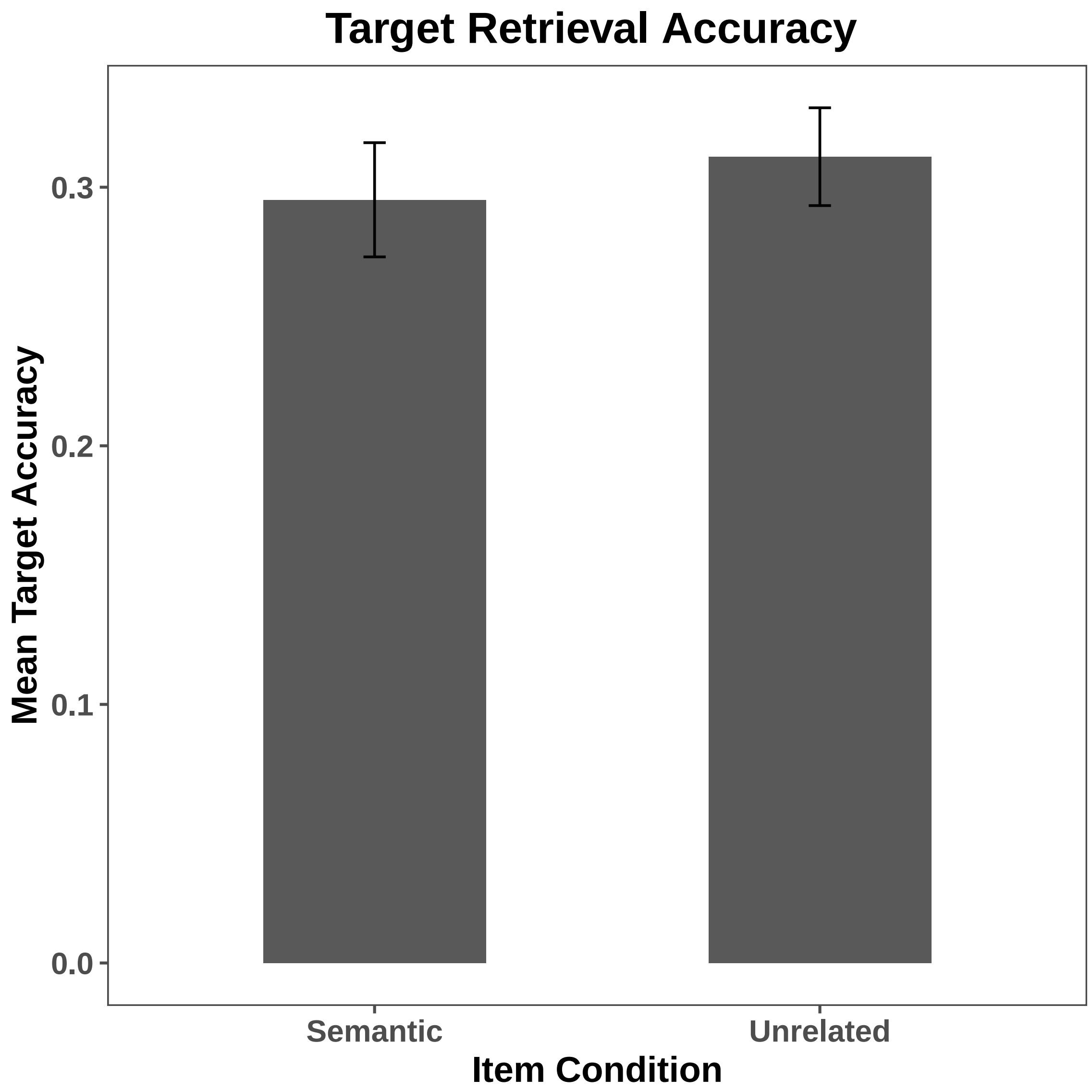


Figure Overall Target Accuracy

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

```
+ ggplot(aes(x = PrimeCondition , y = TargetAccuracy))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.5)+
+   geom_errorbar(aes(ymin = TargetAccuracy - se, ymax = TargetAccuracy + se),
+                 width=.05, position=position_dodge(.5)) +
+   theme_few()+
+   scale_fill_manual(values= c("slategray4", "slategray1"))+
+   xlab("Item Condition") + ylab("Mean Target Accuracy") +
+   ggtitle("Target Retrieval Accuracy ") +
+     theme(axis.text = element_text(face = "bold", size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1.2)),
+           legend.title = element_text(face = "bold", size = rel(1.2)),
+           plot.title = element_text(face = "bold", size = rel(1.4), hjust = .5))
```



ANOVA

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

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

```
Residuals 39  1.054 0.02704

Error: Subject:PrimeCondition
               Df  Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition  1 0.00556 0.005556   0.822   0.37
Residuals      39 0.26373 0.006762
```

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

```
Error: Target
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 71  8.683  0.1223

Error: Target:PrimeCondition
               Df Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition  1   0.01 0.010000   1.224  0.272
Residuals      71   0.58 0.008169
```

```
>
```

4.1 LME

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

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

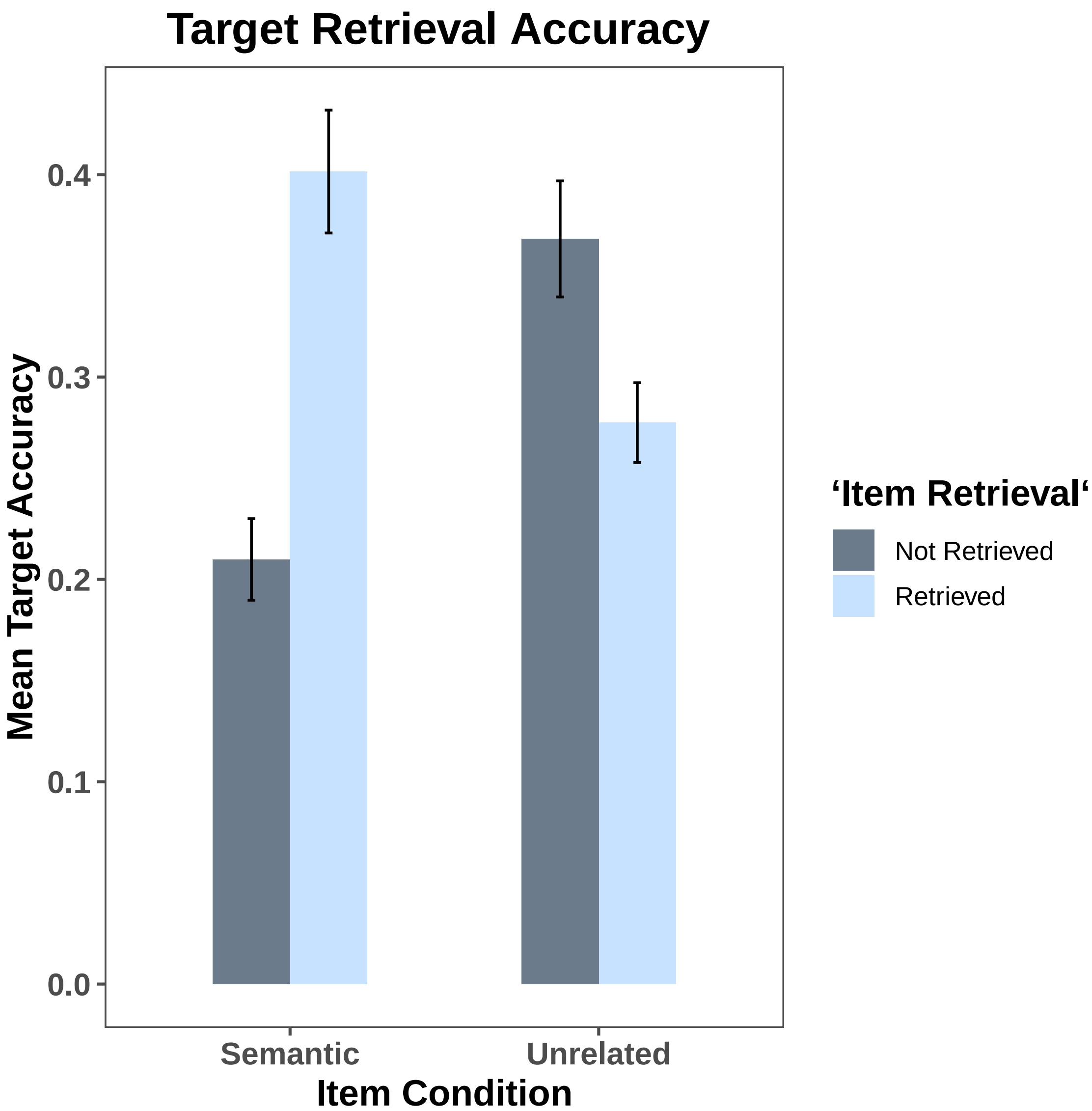
```
     AIC       BIC    logLik deviance df.resid
  2782.6    2806.5   -1387.3    2774.6      2876

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.8354 -0.5050 -0.2841  0.4979  5.6446

Random effects:
 Groups  Name        Variance Std.Dev.
 Target  (Intercept) 2.3546   1.5345
 Subject (Intercept) 0.5502   0.7418
Number of obs: 2880, groups:  Target, 72; Subject, 40

Fixed effects:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     -1.17128    0.22835  -5.129 2.91e-07 ***
PrimeCondition1 -0.13110    0.09788  -1.339     0.18
---
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 -0.206
```

```
>
> # > confint(prime_lmer2)
> # Computing profile confidence intervals ...
> #                      2.5 %      97.5 %
> # .sig01            1.2741502  1.87578523
> # .sig02            0.5706900  0.97947062
> # (Intercept)      -1.6315196 -0.72357416
> # PrimeCondition1 -0.3266536  0.06366817
```

Figure Target Accuracy

```
> target_retrievalacc = group_by(TOTFeedback, Subject, PrimeCondition,
+                               CuedRecallAcc) %>%
+   summarise_at(vars(TargetAccuracy), mean)
> target_rmisc = Rmisc::summarySE(target_retrievalacc,
+                         measurevar = "TargetAccuracy",
+                         groupvars = c("PrimeCondition", "CuedRecallAcc"))
> library(ggplot2)
> library(ggthemes)
> target_rmisc %>% mutate(`Item Retrieval` = factor(CuedRecallAcc,
+                                          levels = unique(CuedRecallAcc),
```

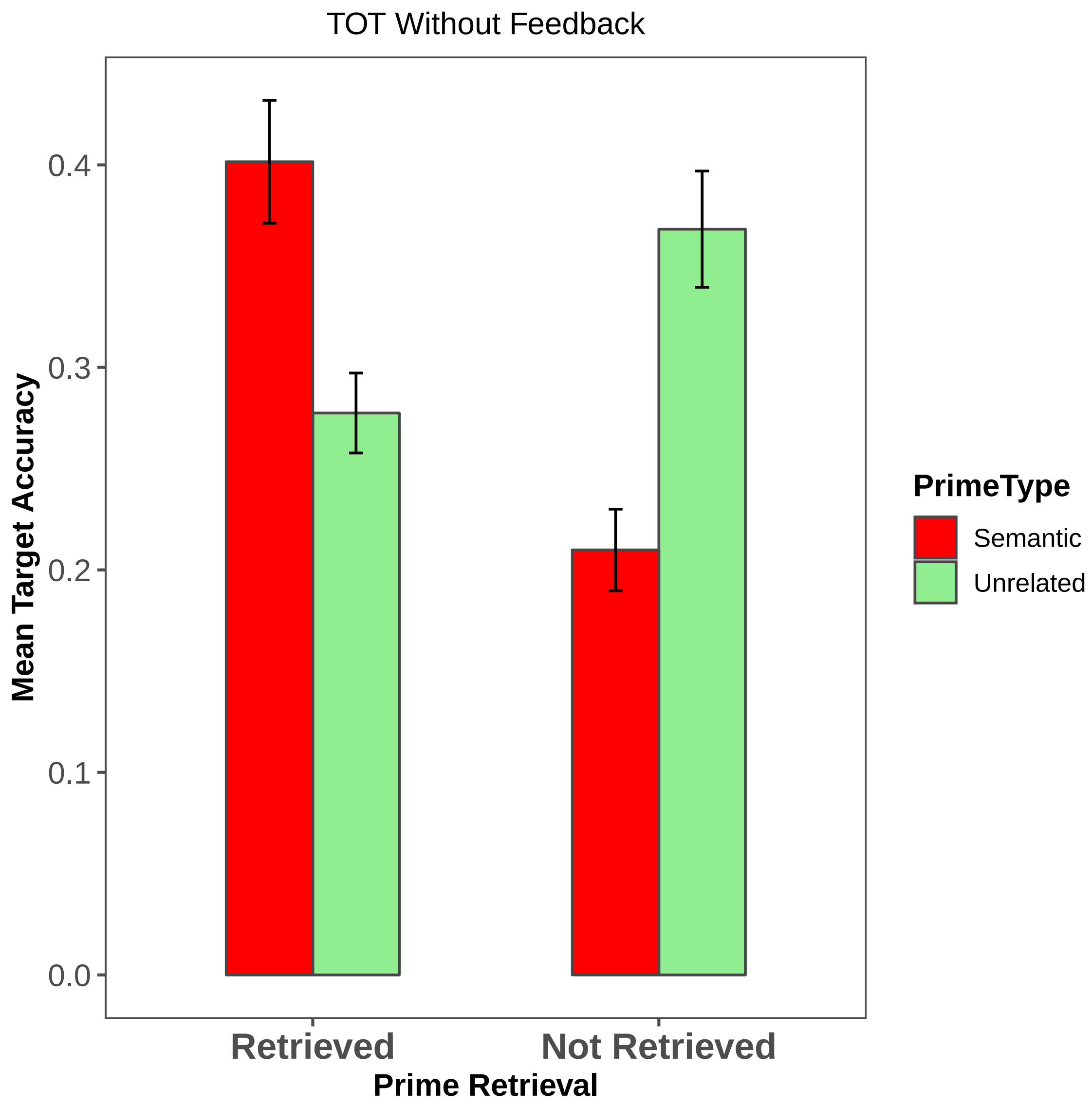
```
+                       labels = c("Not Retrieved", "Retrieved")))%>%
+ ggplot(aes(x = PrimeCondition , y = TargetAccuracy,
+       group = `Item Retrieval`, fill = `Item Retrieval`))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.5)+
+   geom_errorbar(aes(ymin = TargetAccuracy - se, ymax = TargetAccuracy + se),
+                 width=.05, position=position_dodge(.5)) +
+   theme_few()+
+   scale_fill_manual(values= c("slategray4", "slategray1"))+
+   xlab("Item Condition") + ylab("Mean Target Accuracy") +
+   ggtitle("Target Retrieval Accuracy ") +
+     theme(axis.text = element_text(face = "bold", size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1.2)),
+           legend.title = element_text(face = "bold", size = rel(1.2)),
+           plot.title = element_text(face = "bold", size = rel(1.4), hjust = .5))
```



4.2 Masters Retrieval Figure

```
> TOTFeedback_fig = TOTFeedback
> TOTFeedback_fig$primefac = ordered(as.factor(as.character(TOTFeedback_fig$PrimeConditi
+                     levels = c("Semantic", "Unrelated"))
> TOTFeedback_fig$TargetAccuracy = as.numeric(as.character(TOTFeedback_fig$TargetAccurac
> TOTFeedback_fig$CuedRecallAcc_Fac = ordered(as.factor(as.character(TOTFeedback_fig$Cue
> targetacc2  = group_by(TOTFeedback_fig, Subject, primefac,
+                      CuedRecallAcc_Fac) %>%
+   summarise_at(vars(TargetAccuracy), mean)
```

```
> ret_figure = Rmisc::summarySE(targetacc2,
+                     measurevar = "TargetAccuracy",
+                 groupvars = c("primefac", "CuedRecallAcc_Fac"))
> library(ggplot2)
> library(ggthemes)
> ret_figure  %>% mutate(PrimeType = factor(primefac,
+                                         levels = unique(primefac),
+                     labels = c("Semantic",
+                                 "Unrelated")),
+                     `Prime Retrieval` = factor(CuedRecallAcc_Fac,
+                                 levels = unique(CuedRecallAcc_Fac),
+                     labels = c("Retrieved", "Not Retrieved")))%>%
+    ggplot(aes(x = `Prime Retrieval`, y = TargetAccuracy,
+                           group =PrimeType ,
+                           fill = PrimeType)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+    geom_errorbar(aes(ymin = TargetAccuracy - se,
+                      ymax = TargetAccuracy + se),
+                 width=.08, position=position_dodge(.5)) +
+   theme_few()+
+ #   scale_fill_canva()+
+  scale_fill_manual(values = c(  "red",
+                                "lightgreen"))+
+   xlab("Prime Retrieval") + ylab("Mean Target Accuracy") +
+ ggtitle("TOT Without Feedback") +
+   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)))
>
```



4.3 ANOVA

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

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 39  2.157  0.0553

Error: Subject:PrimeCondition
               Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition  1 0.0118 0.01178   0.681  0.414
Residuals      39 0.6752 0.01731

Error: Subject:CuedRecallAcc
              Df Sum Sq Mean Sq F value Pr(>F)
CuedRecallAcc  1 0.1018 0.10177   6.036 0.0186 *
Residuals     39 0.6576 0.01686
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:PrimeCondition:CuedRecallAcc
                             Df Sum Sq Mean Sq F value   Pr(>F)
PrimeCondition:CuedRecallAcc  1 0.7979  0.7979   66.02 6.44e-10 ***
Residuals                    39 0.4713  0.0121
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

5 HLM Model

```
> library(lme4)
> # participant_acc = group_by(TOTFeedback, Subject) %>%
> #   summarise_at(vars(TargetAccuracy, CuedRecallAcc), mean)
> #
> # participant_acc$MeanAcc = (participant_acc$TargetAccuracy +
> #                           participant_acc$CuedRecallAcc)/2
> #
> # colnames(participant_acc) = c("Subject", "TargetAcc", "PrimeAcc", "MeanAcc")
> #
> # TOTFeedback2 = merge(TOTFeedback, participant_acc[,c(1,3,4)],
> #                      by = c("Subject"))
>
>
> item_acc = group_by(TOTFeedback, Target, PrimeCondition) %>%
+   summarise_at(vars(CuedRecallAcc), mean)
> colnames(item_acc) = c("Target","PrimeCondition","PrimeAcc")
> TOTFeedback2 = merge(TOTFeedback, item_acc,
+                      by = c("Target", "PrimeCondition"))
> TOTFeedback_hlm = glmer(data = TOTFeedback2,
+                         TargetAccuracy ~ PrimeCondition*CuedRecallAcc +
+                    PrimeAcc+
```

```
+                          (1|Subject) + (1|Target), family = "binomial")
> summary(TOTFeedback_hlm)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: TargetAccuracy ~ PrimeCondition * CuedRecallAcc + PrimeAcc +
    (1 | Subject) + (1 | Target)
   Data: TOTFeedback2

     AIC      BIC   logLik deviance df.resid
  2766.3   2808.1  -1376.2   2752.3     2873

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.3400 -0.5030 -0.2772  0.4901  6.2871

Random effects:
 Groups  Name        Variance Std.Dev.
 Target  (Intercept) 2.2600   1.5033
 Subject (Intercept) 0.5528   0.7435
Number of obs: 2880, groups:  Target, 72; Subject, 40

Fixed effects:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -1.4103     0.2590  -5.446 5.15e-08 ***
PrimeCondition1               -0.4615     0.1514  -3.048  0.00230 **
CuedRecallAcc                 -0.2916     0.1663  -1.753  0.07952 .
PrimeAcc                       0.7563     0.2454   3.082  0.00206 **
PrimeCondition1:CuedRecallAcc  0.6640     0.2199   3.020  0.00253 **
---
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 CdRclA PrmAcc
PrimeCndtn1 -0.289
CuedRcllAcc -0.140  0.476
PrimeAcc    -0.380  0.021 -0.392
PrmCnd1:CRA  0.216 -0.757 -0.648  0.014
```

```
> car::Anova(TOTFeedback_hlm)
```

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

Response: TargetAccuracy
                            Chisq Df Pr(>Chisq)
PrimeCondition             1.3589  1   0.243724
CuedRecallAcc              0.0707  1   0.790307
```

```
PrimeAcc                          9.4982  1    0.002057 **
PrimeCondition:CuedRecallAcc 9.1186  1    0.002530 **
---
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(TOTFeedback_hlm)
```

```
Analysis of Variance Table
                             Df Sum Sq Mean Sq F value
PrimeCondition                1 1.5872  1.5872  1.5872
CuedRecallAcc                 1 4.1696  4.1696  4.1696
PrimeAcc                      1 9.2330  9.2330  9.2330
PrimeCondition:CuedRecallAcc  1 9.1940  9.1940  9.1940
```

```
>
> # > confint(TOTFeedback_hlm)
> # Computing profile confidence intervals ...
> #                                    2.5 %      97.5 %
> # .sig01                         1.2460017  1.84044905
> # .sig02                         0.5712899  0.98256373
> # (Intercept)                   -1.9325256 -0.90338633
> # PrimeCondition1               -0.7648296 -0.16126250
> # CuedRecallAcc                 -0.6232858  0.03907028
> # PrimeAcc                       0.2698517  1.24940016
> # PrimeCondition1:CuedRecallAcc  0.2273156  1.10316889
```

6 z-scoring RTs

RT prime and Target

```
> hist(TOTFeedback$PrimeDef.RT)
> hist(TOTFeedback$TargetDefinition.RT)
> TOTFeedback_firsttrim_primedef = subset(TOTFeedback,
+                                   TOTFeedback$PrimeDef.RT > 250 &
+                                  TOTFeedback$PrimeDef.RT < 15000)
> TOTFeedback_firsttrim_targetdef = subset(TOTFeedback,
+                                   TOTFeedback$TargetDefinition.RT > 250 &
+                                  TOTFeedback$TargetDefinition.RT < 15000)
>
>
>
```

Prime Def

```
> ## FOR PRIME
> ## aggregate per subject all IVs and DVs
> meanRT = group_by(TOTFeedback_firsttrim_primedef, Subject) %>%
+   summarise_at(vars(PrimeDef.RT), mean)
> colnames(meanRT) = c("Subject",
+                      "MeanRTPrime")
> sdRT = group_by(TOTFeedback_firsttrim_primedef, Subject) %>%
+   summarise_at(vars(PrimeDef.RT), sd)
> colnames(sdRT) = c("Subject",
+                      "sdRTPrime")
> RT_agg = merge(meanRT, sdRT, by = "Subject")
> ## merge aggregate info with long data
> TOTFeedback_z_prime = merge(TOTFeedback_firsttrim_primedef,
+                             RT_agg, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> TOTFeedback_z_prime = TOTFeedback_z_prime %>% mutate(zPrimeRT =
+                                                        (PrimeDef.RT -
+                                                           MeanRTPrime)/sdRTPrime)
> ## checking: subject level means should be zero
>
> sub_pic = group_by(TOTFeedback_z_prime, Subject) %>%
+   summarise_at(vars(zPrimeRT), mean)
```

TargetDefRT

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

```
+   summarise_at(vars(zTargetRT), mean)
>
```

7 Trimming z-RTs

```
> TOTFeedback_z_trimmed_prime = subset(TOTFeedback_z_prime,
+                          TOTFeedback_z_prime$zPrimeRT < 3 &
+                             TOTFeedback_z_prime$zPrimeRT > -3)
> TOTFeedback_z_trimmed_targetdef = subset(TOTFeedback_z_targetdef,
TOTFeedback_z_targetdef$zTargetRT < 3 &                       TOTFeedback_z_tar
```

8 Repeating z-scoring

8.1 For prime

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

8.2 For TargetDefRT

```
> ## aggregate per subject all IVs and DVs
> meanRT_targetdef = group_by(TOTFeedback_z_trimmed_targetdef, Subject) %>%
```

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

8.3 Combining z-RT Prime and Target

```
> ## now we have separately z-scored RTprime and RTtarget. Need to combine.
> ## taking only necessary columns
> TOTFeedback_final_z_prime2 =
+   TOTFeedback_final_z_prime[,c(1,4,25)]
> ## for accuracy
> TOTFeedback_final_z = merge(TOTFeedback_final_z_targetdef,
+                             TOTFeedback_final_z_prime2,
+                             by  = c("Subject", "Trial"))
```

9 Linear Models

```
> # Mean RT to retrieve Target as a function of Prime Condition
>
> # Effect of RT prime on Accuracy
> TOTFeedback_final_z = TOTFeedback_final_z
> contrasts(TOTFeedback_final_z$PrimeCondition) = contr.treatment(2, base = 2)
> library(lme4)
> RTprime_acc_model = glmer(data = TOTFeedback_final_z,
+                      TargetAccuracy ~ PrimeCondition*zPrimeRT_trim +
+                                 (1|Subject) + (1|Target), family = binomial )
> summary(RTprime_acc_model)
```

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

     AIC      BIC   logLik deviance df.resid
  2342.9   2377.5  -1165.4   2330.9     2359

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.2209 -0.5229 -0.2714  0.5411  5.0887

Random effects:
 Groups  Name        Variance Std.Dev.
 Target  (Intercept) 2.609    1.615
 Subject (Intercept) 0.599    0.774
Number of obs: 2365, groups:  Target, 72; Subject, 40

Fixed effects:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -1.1217     0.2419  -4.636 3.55e-06 ***
PrimeCondition1                -0.1552     0.1080  -1.437   0.1507
zPrimeRT_trim                   0.1650     0.0839   1.966   0.0493 *
PrimeCondition1:zPrimeRT_trim  -0.0553     0.1169  -0.473   0.6361
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Correlation of Fixed Effects:
            (Intr) PrmCn1 zPrRT_
PrimeCndtn1 -0.216
zPrimRT_trm -0.024  0.038
PrmCn1:PRT_  0.011 -0.038 -0.732
```

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

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

Response: TargetAccuracy
                              Chisq Df Pr(>Chisq)
PrimeCondition               2.1193  1    0.14546
zPrimeRT_trim                5.6524  1    0.01743 *
PrimeCondition:zPrimeRT_trim 0.2239  1    0.63605
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

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

```
Analysis of Variance Table
                             Df Sum Sq Mean Sq F value
PrimeCondition                1 2.2568  2.2568  2.2568
zPrimeRT_trim                 1 5.7066  5.7066  5.7066
PrimeCondition:zPrimeRT_trim  1 0.2256  0.2256  0.2256
```

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

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

REML criterion at convergence: 6131.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2371 -0.7223 -0.1935  0.6286  3.6099

Random effects:
 Groups   Name        Variance Std.Dev.
 Target   (Intercept) 0.2477   0.4977
 Subject  (Intercept) 0.0000   0.0000
 Residual             0.7204   0.8487
Number of obs: 2365, groups:  Target, 72; Subject, 40

Fixed effects:
                              Estimate Std. Error t value
(Intercept)                    0.09098    0.06374   1.427
PrimeCondition1               -0.15246    0.03508  -4.347
zPrimeRT_trim                  0.04013    0.02758   1.455
PrimeCondition1:zPrimeRT_trim  0.02769    0.03809   0.727

Correlation of Fixed Effects:
            (Intr) PrmCn1 zPrRT_
PrimeCndtn1 -0.279
zPrimRT_trm -0.016  0.030
PrmCn1:PRT_  0.012 -0.021 -0.731
```

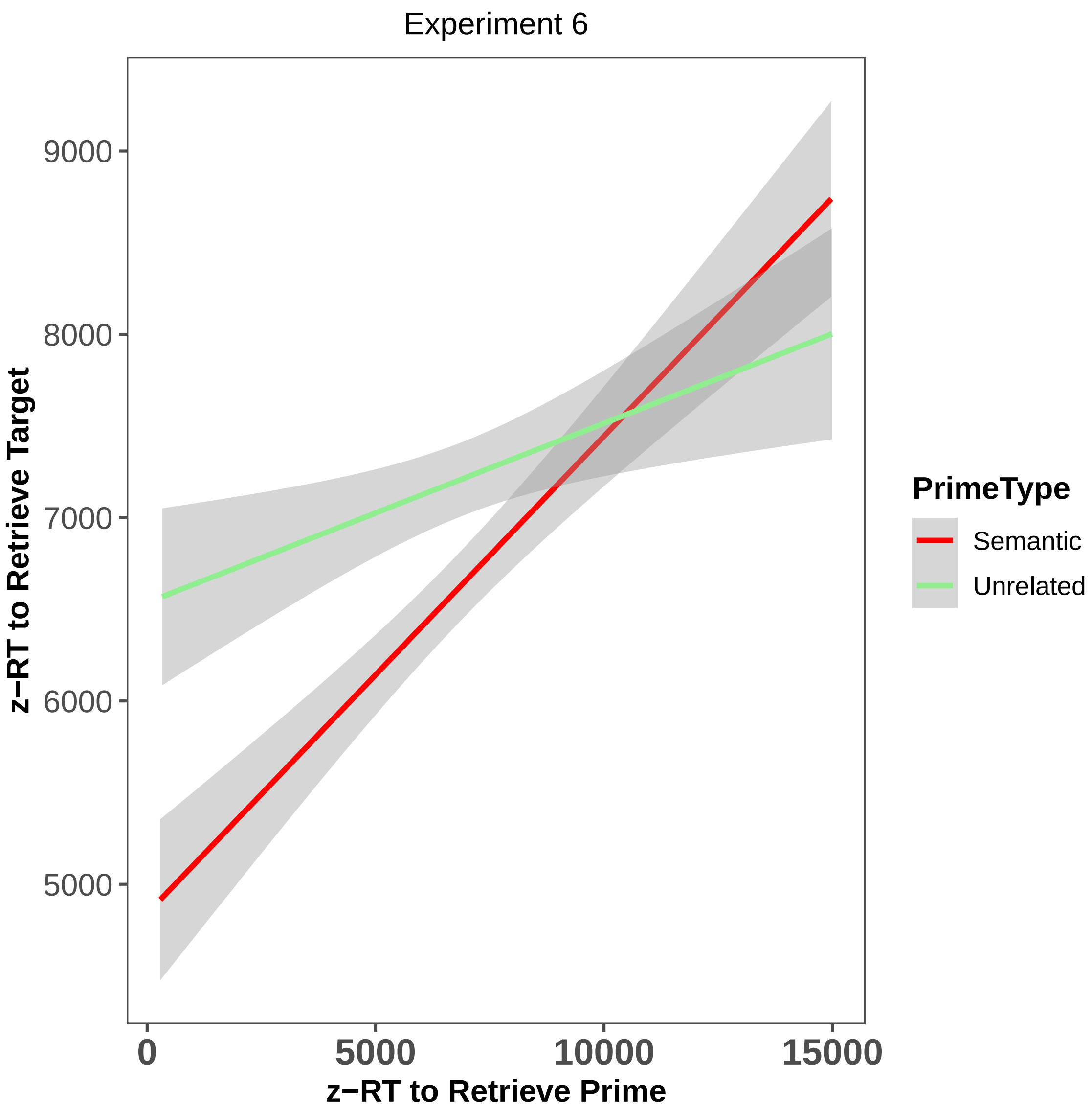
```
> car::Anova(RTprime_RTtargetdef_model)
```

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

Response: zTargetRT_trim
                              Chisq Df Pr(>Chisq)
PrimeCondition              18.7678  1  1.476e-05 ***
zPrimeRT_trim                8.4861  1   0.003579 **
PrimeCondition:zPrimeRT_trim 0.5287  1   0.467133
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
>
```

9.1 RAW RT Model

```
> TOTFeedback_final_z$PrimeType = ordered(as.factor(as.character(TOTFeedback_final_z$Pri
> TOTFeedback_final_z %>%
+   ggplot(aes(x = PrimeDef.RT, y = TargetDefinition.RT,
+              group = PrimeType, color = PrimeType)) +
+   geom_smooth(method = "lm", size = 1)+
+     xlab("z-RT to Retrieve Prime") + ylab ("z-RT to Retrieve Target")+
+ theme_few() +
+ scale_color_manual(values = c( "red","lightgreen"))+
+   ggtitle("Experiment 6") +
+   theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5, size = rel(1)),
+          axis.text.x = element_text(face = "bold", size = rel(1.2)))
>
```



9.2 Acc Model

```
> ## sd for zPrimeRecogRT_trim
> sd(TOTFeedback_final_z$zPrimeRT_trim)
```

```
[1] 0.9934247
```

```
> # this is the model
>
> # RTprime_acc_model = glmer(data = TOTFeedback_final_z,
```

```
> #                    TargetAccuracy ~ PrimeCondition*zPrimeRT_trim +
> #                             (1|Subject) + (1|Target), family = binomial )
> # summary(RTprime_acc_model)
>
> primert_model = lmer(data = TOTFeedback_final_z,
+                      zPrimeRT_trim ~ 1 + (1 | Subject) +
+                        (1|Target))
> summary(primert_model)
```

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

REML criterion at convergence: 6522.9

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.3292 -0.7784 -0.1724  0.6468  3.4998

Random effects:
 Groups   Name        Variance Std.Dev.
 Target   (Intercept) 0.1132   0.3365
 Subject  (Intercept) 0.0000   0.0000
 Residual             0.8767   0.9363
Number of obs: 2365, groups:  Target, 72; Subject, 40

Fixed effects:
            Estimate Std. Error t value
(Intercept)  0.01672    0.04412   0.379
```

```
> VarCorr(primert_model)
```

```
 Groups   Name        Std.Dev.
 Target   (Intercept) 0.33652
 Subject  (Intercept) 0.00000
 Residual             0.93634
```

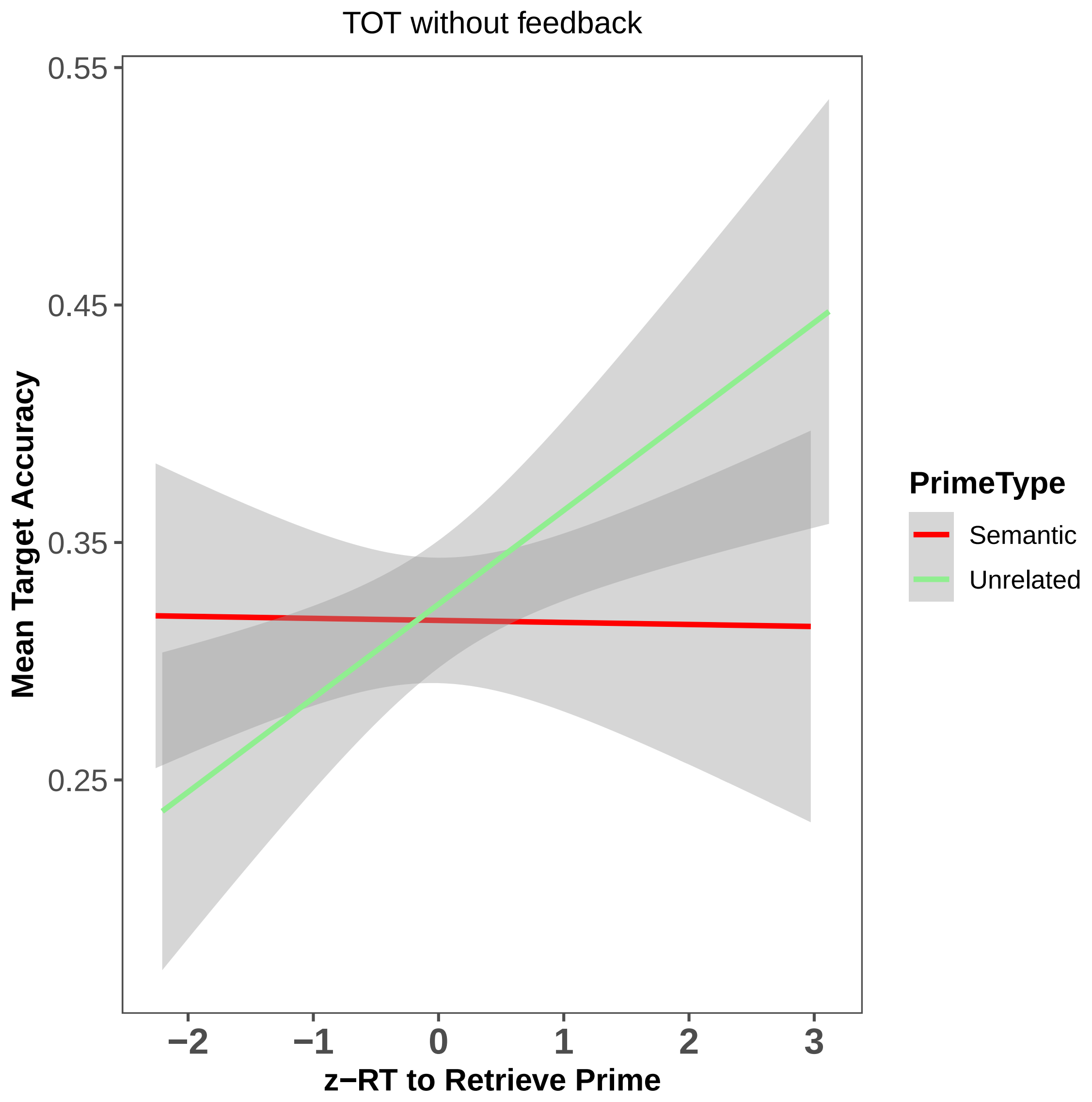
```
> SD_prime ← as.data.frame(VarCorr(primert_model))[3, 5]
> ## now we need to find increments for each prime condition
>
> primert_model_2 ← lmer(data = TOTFeedback_final_z,
+                         zPrimeRT_trim ~ 1 + PrimeCondition +
+                       (1|Subject) + (1|Target))
> prime_Inc_1_U ← 0*fixef(primert_model_2)[1]
> prime_Inc_1_R ← 1*fixef(primert_model_2)[2]
> predict_data_U ← with(TOTFeedback_final_z,
+                    data.frame(school=1,
+  zPrimeRT_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(TOTFeedback_final_z,
+                   data.frame(school=1,
+  zPrimeRT_trim=seq(from=-prime_Inc_1_R-SD_prime,
+           to=-prime_Inc_1_R+SD_prime,
+           by=SD_prime),
+  PrimeCondition = 1))
> predict_data = rbind(predict_data_U,
+                      predict_data_R)
> predict_data$PrimeCondition = ifelse(predict_data$PrimeCondition == 0,
+                                      "Unrelated","Semantic")
> predict_data = predict_data %>%
+   mutate(predicted_values = predict(RTprime_acc_model,
+           newdata = predict_data, re.form = NA))
> predict_data$prob = exp(predict_data$predicted_values)/(1+exp(predict_data$predicted_v
> predict_data$PrimeCondition = ordered(as.factor(as.character(predict_data$PrimeConditi
> predict_data %>%
+   mutate(PrimeType = factor(PrimeCondition, levels = unique(PrimeCondition),
+                      labels = c("Unrelated", "Semantic")))%>%
+   ggplot(aes(x = zPrimeRT_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))
```

9.3 RAW ACC Model

```
> TOTFeedback_final_z$primefac = ordered(as.factor(as.character(TOTFeedback_final_z$Prim
> TOTFeedback_final_z %>%
+   mutate(PrimeType = factor(primefac, levels = unique(primefac),
+                      labels = c("Semantic",
+                                 "Unrelated")))%>%
+   ggplot(aes(x = zPrimeRT_trim, y = TargetAccuracy,
+              group = PrimeType, color = PrimeType)) +
+   geom_smooth(method = "lm")+
+     xlab("z-RT to Retrieve Prime") + ylab ("Mean Target Accuracy")+
+   ggtitle("TOT without feedback")+
+ theme_few() +
+ scale_color_manual(values = c( "red","lightgreen"))+
```

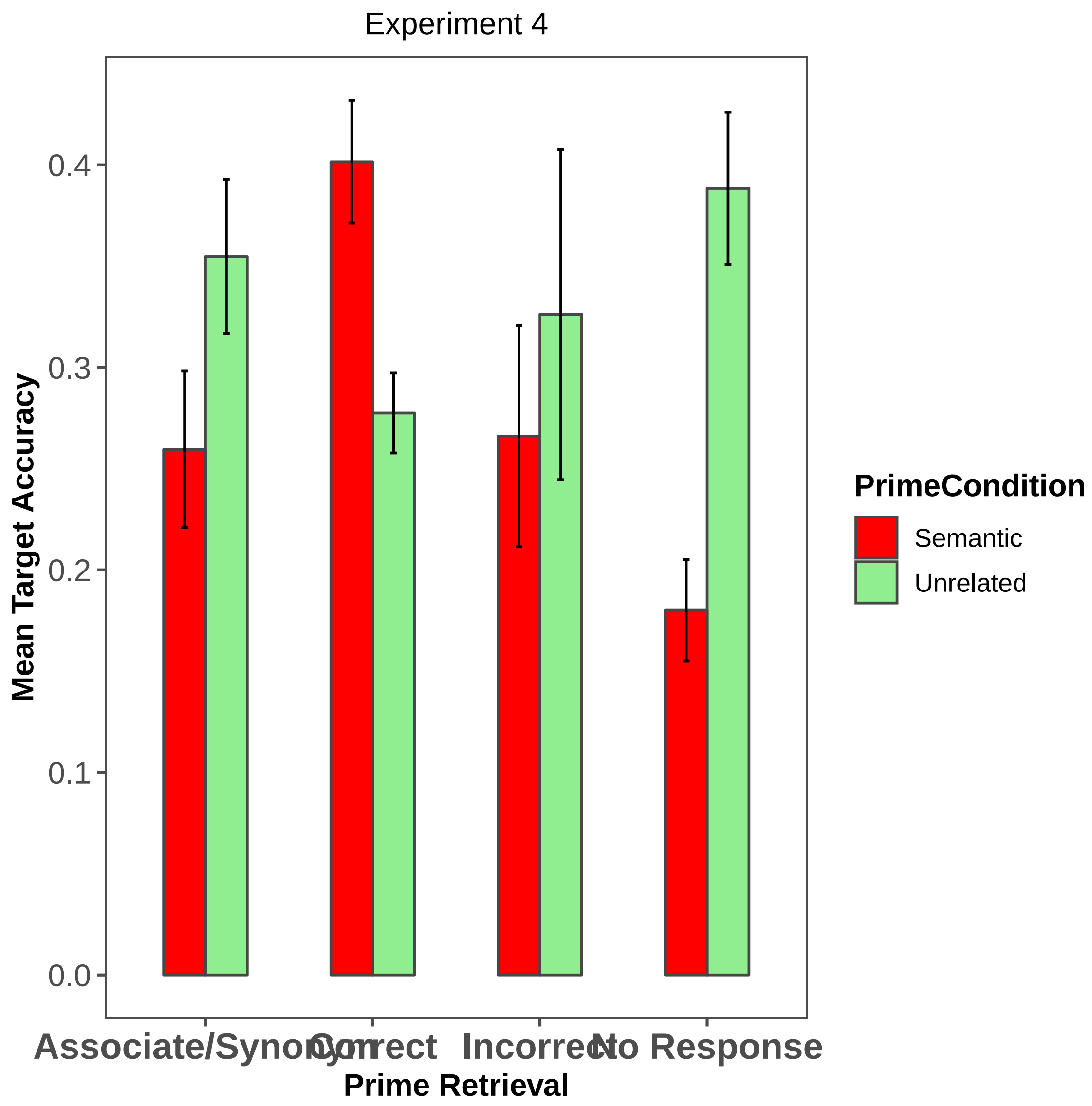
```
+    theme(axis.text = element_text(size = rel(1)),
+            axis.title = element_text(face = "bold", size = rel(1)),
+            legend.title = element_text(face = "bold", size = rel(1)),
+           plot.title = element_text(hjust = .5, size = rel(1)),
+           axis.text.x = element_text(face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



10 Response Analysis

10.1 All Responses

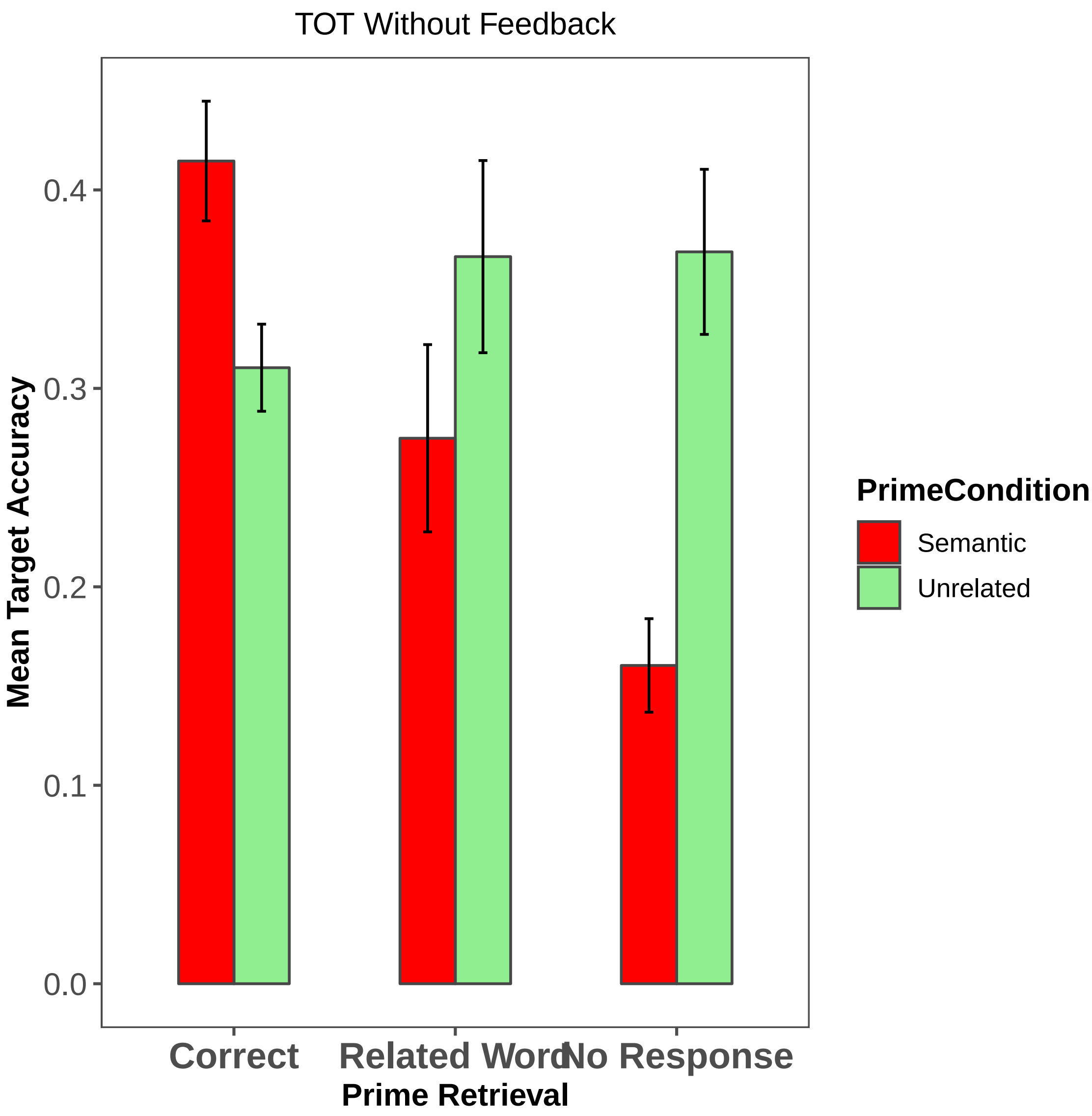
```
>  TOTFeedback = read.csv("TOTwoFeedback_FINAL.csv",
+                          header = TRUE, sep = ",")
> TOTFeedback$AllResponse = ifelse(TOTFeedback$PrimeRespType %in%
+                                 c("Associate", "Synonym"), "Associate/Synonym",
+                                ifelse(TOTFeedback$PrimeRespType == "NoResponse",
+                                       "No Response",
+                             ifelse(TOTFeedback$PrimeRespType == "Correct","Correct",
+                                    "Incorrect")))
> TOTFeedback_subject = group_by(TOTFeedback, Subject, PrimeCondition, AllResponse) %>%
+   summarize_at(vars(TargetAccuracy), mean)
> ret_figure = Rmisc::summarySE(TOTFeedback_subject,
+                     measurevar = "TargetAccuracy",
+                 groupvars = c("PrimeCondition", "AllResponse"))
> library(ggplot2)
> library(ggthemes)
> library(dplyr)
> ret_figure  %>%
+    ggplot(aes(x = AllResponse, y = TargetAccuracy,
+                           group =PrimeCondition ,
+                           fill = PrimeCondition)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+    geom_errorbar(aes(ymin = TargetAccuracy - se,
+                      ymax = TargetAccuracy + se),
+                 width=.08, position=position_dodge(.5)) +
+   theme_few()+
+ #   scale_fill_canva()+
+  scale_fill_manual(values = c(  "red",
+                               "lightgreen"))+
+   xlab("Prime Retrieval") + ylab("Mean Target Accuracy") +
+ ggtitle("Experiment 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)))
```



10.2 3-group Responses

```
> # TOTFeedback = read.csv("TOTwoFeedback_FINAL.csv",
> #                         header = TRUE, sep = ",")
>
> TOTFeedback = TOTFeedback_final_z
> TOTFeedback$Response = ifelse(TOTFeedback$PrimeRespType %in%
+                          c("Associate", "Incorrect"), "Related Word",
+                        ifelse(TOTFeedback$PrimeRespType == "NoResponse",
+                               "No Response", "Correct"))
```

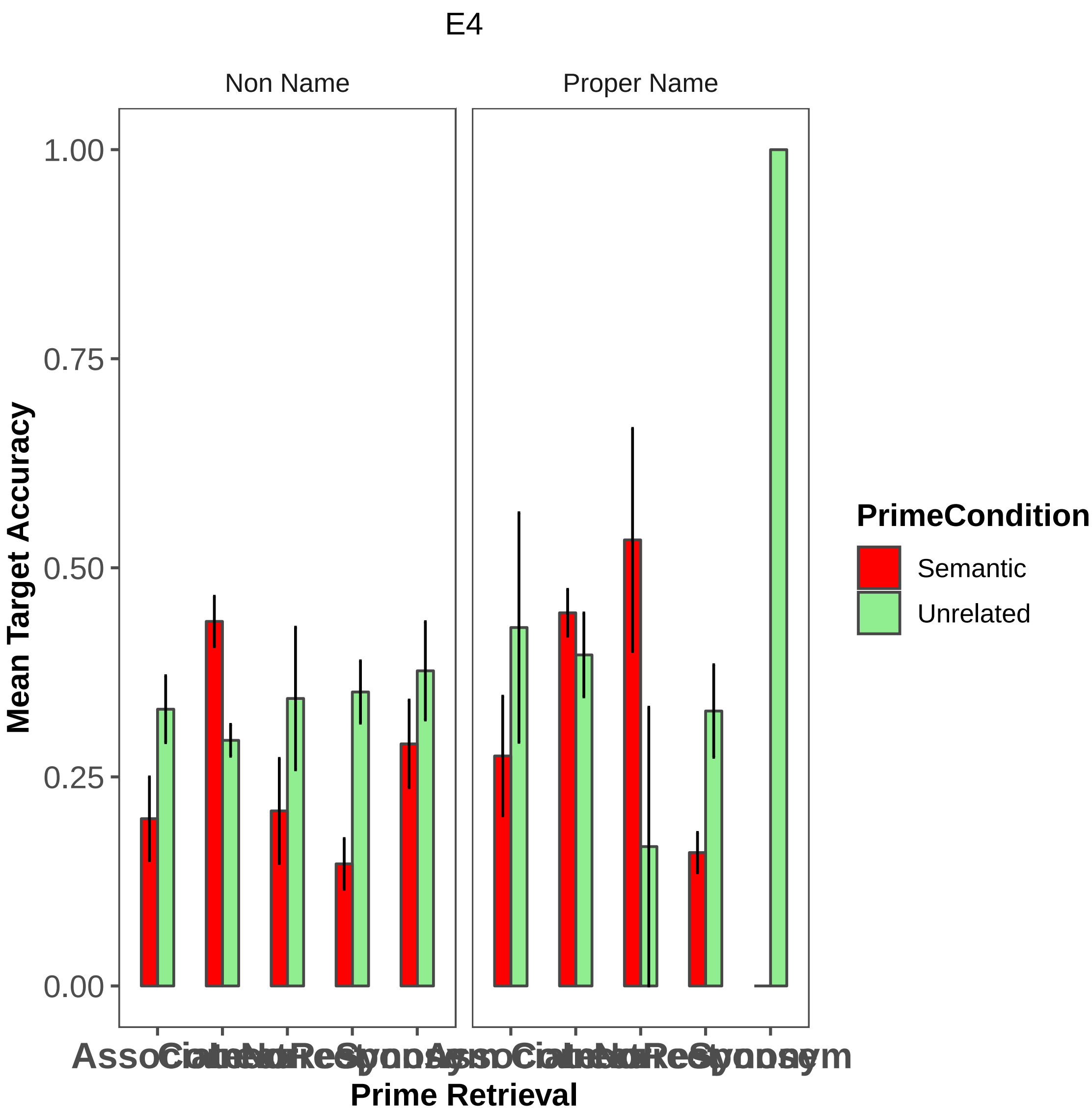
```
> TOTFeedback$Response = ordered(as.factor(as.character(TOTFeedback$Response)),
+                       levels = c("Correct", "Related Word", "No Response"))
> TOTFeedback_subject = group_by(TOTFeedback, Subject, PrimeCondition, Response) %>%
+   summarize_at(vars(TargetAccuracy), mean)
> ret_figure = Rmisc::summarySE(TOTFeedback_subject,
+                     measurevar = "TargetAccuracy",
+                 groupvars = c("PrimeCondition", "Response"))
> library(ggplot2)
> library(ggthemes)
> library(dplyr)
> ret_figure  %>%
+    ggplot(aes(x = Response, y = TargetAccuracy,
+                           group =PrimeCondition ,
+                           fill = PrimeCondition)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+           color ="gray28")+
+    geom_errorbar(aes(ymin = TargetAccuracy - se,
+                      ymax = TargetAccuracy + se),
+                 width=.08, position=position_dodge(.5)) +
+   theme_few()+
+ #   scale_fill_canva()+
+  scale_fill_manual(values = c(  "red",
+                                 "lightgreen"))+
+   xlab("Prime Retrieval") + ylab("Mean Target Accuracy") +
+ ggtitle("TOT Without Feedback") +
+   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)))
```



10.3 POS-split Responses

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

```
+                             group =PrimeCondition ,
+                             fill = PrimeCondition)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+    geom_errorbar(aes(ymin = TargetAccuracy - se,
+                      ymax = TargetAccuracy + se),
+                 width=.08, position=position_dodge(.5)) +
+   theme_few()+
+ facet_wrap(~Prime_POS)+
+   scale_fill_manual(values = c(  "red",
+                                  "lightgreen"))+
+   xlab("Prime Retrieval") + ylab("Mean Target Accuracy") +
+ 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, size = rel(1)),
+          axis.text.x = element_text(face = "bold", size = rel(1.2)))
```



10.4 LME

```
> TOTFeedback$Response = as.factor(TOTFeedback$Response)
> contrasts(TOTFeedback$Response) = contr.treatment(3, base = 1)
> contrasts(TOTFeedback$PrimeCondition) = contr.treatment(2, base = 2)
> TOTFeedback_hlm2 = glmer(data = TOTFeedback,
+                               TargetAccuracy ~ PrimeCondition*Response +
+                          (1|Subject) + (1|Target), family = "binomial")
> summary(TOTFeedback_hlm2)
```

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

     AIC      BIC   logLik deviance df.resid
  2318.8   2365.0  -1151.4   2302.8     2357

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.9572 -0.5183 -0.2643  0.5230  6.6730

Random effects:
 Groups  Name        Variance Std.Dev.
 Target  (Intercept) 2.5051   1.5828
 Subject (Intercept) 0.5307   0.7285
Number of obs: 2365, groups:  Target, 72; Subject, 40

Fixed effects:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)               -1.07697    0.24385  -4.417    1e-05 ***
PrimeCondition1            0.16715    0.14451   1.157 0.247397
Response2                  0.01441    0.22718   0.063 0.949410
Response3                 -0.14960    0.20964  -0.714 0.475472
PrimeCondition1:Response2 -0.40703    0.33628  -1.210 0.226127
PrimeCondition1:Response3 -1.03879    0.29284  -3.547 0.000389 ***
---
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 -0.284
Response2   -0.207  0.359
Response3   -0.220  0.396  0.275
PrmCndt1:R2  0.140 -0.481 -0.681 -0.193
PrmCndt1:R3  0.158 -0.559 -0.197 -0.700  0.266
```

```
> #sjPlot::plot_model(TOTFeedback_hlm2, type = "int")
>
> car::Anova(TOTFeedback_hlm2)
```

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

Response: TargetAccuracy
                          Chisq Df Pr(>Chisq)
```

```
PrimeCondition            1.4983  1   0.220937
Response                 20.0426  2  4.444e-05 ***
PrimeCondition:Response  12.6593  2   0.001783 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

10.5 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 | -1.0769679|
|SC | -0.9098139|
|UO | -1.0625536|
|SO | -1.3024266|
|UN | -1.2265660|
|SN | -2.0982034|
```

```
> 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 = TargetAccuracy ~ PrimeCondition * Response +
    (1 | Subject) + (1 | Target), data = TOTFeedback, family = "binomial")

Linear Hypotheses:
        Estimate Std. Error z value Pr(>|z|)
UC == 0  -1.0770     0.2439  -4.417 5.02e-05 ***
SC == 0  -0.9098     0.2456  -3.705 0.000423 ***
UO == 0  -1.0626     0.2970  -3.578 0.000423 ***
```

```
SO == 0   -1.3024      0.3129   -4.162 9.47e-05 ***
UN == 0   -1.2266      0.2844   -4.312 6.47e-05 ***
SN == 0   -2.0982      0.2843   -7.381 9.42e-13 ***
---
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 = TargetAccuracy ~ PrimeCondition * Response +
    (1 | Subject) + (1 | Target), data = TOTFeedback, family = "binomial")

Linear Hypotheses:
              Estimate Std. Error z value Pr(>|z|)
SC vs UC == 0  -0.1672     0.1445  -1.157  0.57290
SO vs UO == 0   0.2399     0.2953   0.812  0.80087
SN vs UN == 0   0.8716     0.2435   3.580  0.00103 **
---
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)
```

10.6 Specific Comparisons

```
> responses_correct = TOTFeedback %>% 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(TargetAccuracy), 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$TargetAccuracy,
+        responses_correct_sub_unrelated$TargetAccuracy,
+        paired = TRUE)
```

```
        Paired t-test

data:  responses_correct_sub_semantic$TargetAccuracy and responses_correct_sub_unrelated
t = 3.4052, df = 39, p-value = 0.001544
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.04228467 0.16601512
sample estimates:
mean of the differences
              0.1041499
```

```
> responses_other = TOTFeedback %>% 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(TargetAccuracy), mean)
> responses_other_sub = responses_other_sub %>%
+   filter(!Subject %in% c(15, 23,24,29,32,34))
> 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$TargetAccuracy,
+        responses_other_sub_unrelated$TargetAccuracy,
+        paired = TRUE)
```

```
        Paired t-test

data:  responses_other_sub_semantic$TargetAccuracy and responses_other_sub_unrelated$Tar
t = -1.1191, df = 32, p-value = 0.2714
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.20816448  0.06053669
sample estimates:
mean of the differences
             -0.0738139
```

```
> responses_none = TOTFeedback %>% 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(TargetAccuracy), mean)
> responses_none_sub = responses_none_sub %>% filter( Subject!= 35)
> 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$TargetAccuracy,
+        responses_none_sub_unrelated$TargetAccuracy,
+        paired = TRUE)
```

```
        Paired t-test

data:  responses_none_sub_semantic$TargetAccuracy and responses_none_sub_unrelated$Targe
t = -5.3701, df = 37, p-value = 4.475e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.3004115 -0.1358178
sample estimates:
mean of the differences
             -0.2181146
```

10.7 State by Response Type

```
> response_state = group_by(TOTFeedback, Subject, PrimeCondition, TargetRespType, Target
+   summarise(trials = n())
> response_state_rmisc = Rmisc::summarySE(response_state,
+                                         measurevar = "trials",
+                                         groupvars = c("PrimeCondition",
+                                           "TargetRespType", "TargetQuestion"))
> response_state_rmisc$State = ifelse(response_state_rmisc$TargetQuestion == 1,
+                            "Know", ifelse(response_state_rmisc$TargetQuestion == 2,
+                        "Dont Know", ifelse(response_state_rmisc$TargetQuestion == 3,
+                                          "Other Word", "TOT")))
> response_state_rmisc$State = ordered(as.factor(as.character(response_state_rmisc$State
+                         levels = c("Know", "Dont Know", "Other Word", "TOT"))
> response_state_rmisc  %>%
+    ggplot(aes(x = TargetRespType, y = trials,
+                            group =State ,
+                            fill = State)) +
+   geom_bar(stat = "identity", position = "dodge", width = 0.5,
+            color ="gray28")+
+    geom_errorbar(aes(ymin = trials - se,
```

```
+                         ymax = trials + se),
+                    width=.08, position=position_dodge(.5)) +
+   theme_few()+
+   facet_wrap(~PrimeCondition)+
+   xlab("Target Response Types") + ylab("Mean Number of Trials") +
+ ggtitle("E4: Retrieval Without Feedback") +
+   theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5, size = rel(1)),
+          axis.text.x = element_text(face = "bold", size = rel(1.2)))
>
>
```

11 Percentage State Prime Analysis

```
> state = read.csv("E5_TOTwoFeedback_AGG.csv",header = TRUE, sep = ",")
> j_statepercent = state[,c(1,21:28)] # use for prime percents
> j_statepercent$Subject = as.factor(j_statepercent$Subject)
> library(tidyr)
> library(dplyr)
> statepercent <- j_statepercent %>%
+   gather(PrimeState, Percent,
+          prop_r_know, prop_r_dontknow, prop_r_other, prop_r_TOT,
+          prop_u_know, prop_u_dontknow, prop_u_other, prop_u_TOT) %>%
+   separate(PrimeState, c('Prop', 'Prime', 'State'), sep = "_") %>%
+   arrange(Subject)
> #removing prop
> statepercent = statepercent[,-2]
> colnames(statepercent) = c( "Subject",
+                             "PrimeCondition", "State", "Percent")
> statepercent$Subject <- as.factor(statepercent$Subject)
> statepercent$PrimeCondition <- as.factor(statepercent$PrimeCondition)
> statepercent$State <- as.factor(statepercent$State)
> statepercent$Percent <- as.numeric(as.character(statepercent$Percent))
> ## anova
>
> state_aov = aov(data = statepercent, Percent ~ PrimeCondition*State +
+                   Error(Subject/(PrimeCondition*State)))
> summary(state_aov)
```

```
Error: Subject
          Df    Sum Sq   Mean Sq F value Pr(>F)
Residuals 39 3.638e-18 9.327e-20

Error: Subject:PrimeCondition
```

```
                  Df      Sum Sq    Mean Sq F value Pr(>F)
PrimeCondition  1 1.125e-19 1.125e-19   2.053   0.16
Residuals      39 2.137e-18 5.481e-20

Error: Subject:State
             Df Sum Sq Mean Sq F value   Pr(>F)
State         3  3.262  1.0873   27.54 1.51e-13 ***
Residuals 117  4.619  0.0395
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:PrimeCondition:State
                         Df Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition:State    3 0.0159 0.005292   0.823  0.484
Residuals             117 0.7526 0.006433
```

11.0.1 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("E5")  +
+    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
```

11.0.2 know

```
> e1_know = statepercent %>% filter(State == "know")
> e1_know_aov = aov(data = e1_know,
+                            Percent ~ PrimeCondition +
+                          Error(Subject/PrimeCondition))
> summary(e1_know_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 39  1.644 0.04216

Error: Subject:PrimeCondition
               Df  Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition  1 0.00556 0.005556   0.826  0.369
Residuals      39 0.26219 0.006723
```

11.0.3 dont know

```
> e1_dontknow = statepercent %>% filter(State == "dontknow")
> e1_dontknow_aov = aov(data = e1_dontknow,
+                            Percent ~ PrimeCondition +
+                          Error(Subject/PrimeCondition))
> summary(e1_dontknow_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 39  1.661 0.04258

Error: Subject:PrimeCondition
               Df  Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition  1 0.00652 0.006520   0.927  0.342
Residuals      39 0.27434 0.007034
```

11.0.4 other

```
> e1_other = statepercent %>% filter(State == "other")
> e1_other_aov = aov(data = e1_other,
```

```
+                            Percent ~ PrimeCondition +
+                          Error(Subject/PrimeCondition))
> summary(e1_other_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 39  0.494 0.01267

Error: Subject:PrimeCondition
               Df  Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition  1 0.00217 0.002170   0.873  0.356
Residuals      39 0.09698 0.002487
```

11.0.5 TOT

```
> e1_TOT = statepercent %>% filter(State == "TOT")
> e1_TOT_aov = aov(data = e1_TOT,
+                            Percent ~ PrimeCondition +
+                          Error(Subject/PrimeCondition))
> summary(e1_TOT_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 39 0.8206 0.02104

Error: Subject:PrimeCondition
               Df  Sum Sq  Mean Sq F value Pr(>F)
PrimeCondition  1 0.00163 0.001630   0.534  0.469
Residuals      39 0.11913 0.003054
```

11.1 Raw Retrieval States

```
> library(dplyr)
> SemanticCuedRecall_Count = group_by(TOTFeedback,
+                                     Subject, PrimeCondition,
+                                     TargetQuestion) %>%
+   summarise(Count = n())
> state_rmisc = Rmisc::summarySE(SemanticCuedRecall_Count,
+                               measurevar = "Count",
+                               groupvars = c("PrimeCondition",
+                                             "TargetQuestion"))
> x <- c("1","2", "3", "4")
> state_rmisc = state_rmisc %>%
+   mutate(rstate =  factor(TargetQuestion, levels = x)) %>%
+   arrange(rstate)
> library(ggplot2)
```

```
> library(ggthemes)
> percentplot = state_rmisc %>%
+   mutate(PrimeType = factor(PrimeCondition, levels = unique(PrimeCondition),
+                         labels = c("Semantic", "Unrelated")),
+    R = factor(rstate, levels = unique(rstate),
+                                labels = c( "1: Know","2: Dont Know",
+                                            "3:Other", "4: TOT")))%>%
+ ggplot(aes(x = R, y = Count,
+            group = PrimeType, fill = PrimeType))+
+  geom_bar(stat = "identity", position = "dodge", width = 0.7,
+           color= "black")+
+   geom_errorbar(aes(ymin=Count - se, ymax=Count + se),
+              width=.2, color = "gray26",
+              position = position_dodge(0.7))+
+  theme_few()+
+     xlab("") + ylab("Number of trials") +
+  scale_fill_manual(values = c( "red",
+                               "lightgreen"))+
+   ggtitle("E6")  +
+    theme(axis.text = element_text(size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+          plot.title = element_text(hjust = .5),
+                   axis.text.x = element_text(size = rel(1)),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
> percentplot
```