All TOT studies Analysis

Abhilasha Kumar

April 30, 2018

1 Reading the Data File

```
> TOT = read.csv("CombinedPrimeFlash_PrimeDemask_CSV.csv", header = TRUE, sep = ",")
```

2 Accuracy per Prime Condition

```
> library(dplyr)
> overall_acc = group_by(TOT, Experiment) %>%
+   summarise_at(vars(TargetAccuracy), mean)
> overall_itemacc = group_by(TOT, Stimuli1) %>%
+   summarise_at(vars(TargetAccuracy), mean)
> #low_acc = overall_itemacc %>% filter(TargetAccuracy < .25)
> #low_acc = low_acc[order(low_acc$TargetAccuracy),]
>
> overall_acc_subject = group_by(TOT, Experiment, Subject) %>%
+   summarise_at(vars(TargetAccuracy), mean)
> prime_acc = group_by(TOT, Experiment, PrimeCondition) %>%
+   summarise_at(vars(TargetAccuracy), mean)
> prime_subject_acc = group_by(TOT, Experiment, Subject, PrimeCondition) %>%
+   summarise_at(vars(TargetAccuracy), mean)
> prime_subject_acc_E1_E2 = group_by(TOT, ExperimentName, Subject, PrimeCondition) %>%
+   summarise_at(vars(TargetAccuracy), mean)
>
```

ANOVA

```
> prime_subject_acc$Experiment = as.factor(prime_subject_acc$Experiment)
> prime_subject_acc$PrimeCondition = as.factor(prime_subject_acc$PrimeCondition)
> prime_subject_acc$Subject = as.factor(prime_subject_acc$Subject)
> exp_flash_demask = prime_subject_acc %>% filter(Experiment == "PrimeFlash" |
+                                                Experiment == "PrimeDemask")
> target_aov= aov(data = exp_flash_demask, TargetAccuracy ~ Experiment*PrimeCondition +
+                                Error(Subject/PrimeCondition))
> summary(target_aov)
```

```
Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Experiment  1  0.042 0.04241   0.505   0.48
Residuals  55  4.618 0.08397

Error: Subject:PrimeCondition
                            Df Sum Sq Mean Sq F value Pr(>F)
PrimeCondition               3 0.0888 0.02961   2.506 0.0609 .
Experiment:PrimeCondition    3 0.0869 0.02898   2.452 0.0653 .
Residuals                  165 1.9500 0.01182
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> options(contrasts = c('contr.sum', 'contr.poly'))
> library(lsmeans)
> library(multcomp)
> imm_lsm = lsmeans::lsmeans(target_aov, c("Experiment", "PrimeCondition"))
> prime_effect = cld(imm_lsm, alpha = 0.05,
+                 adjust = "tukey", details = TRUE, by = "PrimeCondition")
> library(knitr)
> kable(subset(prime_effect$comparisons,prime_effect$comparisons$p.value < 0.1 ))
```

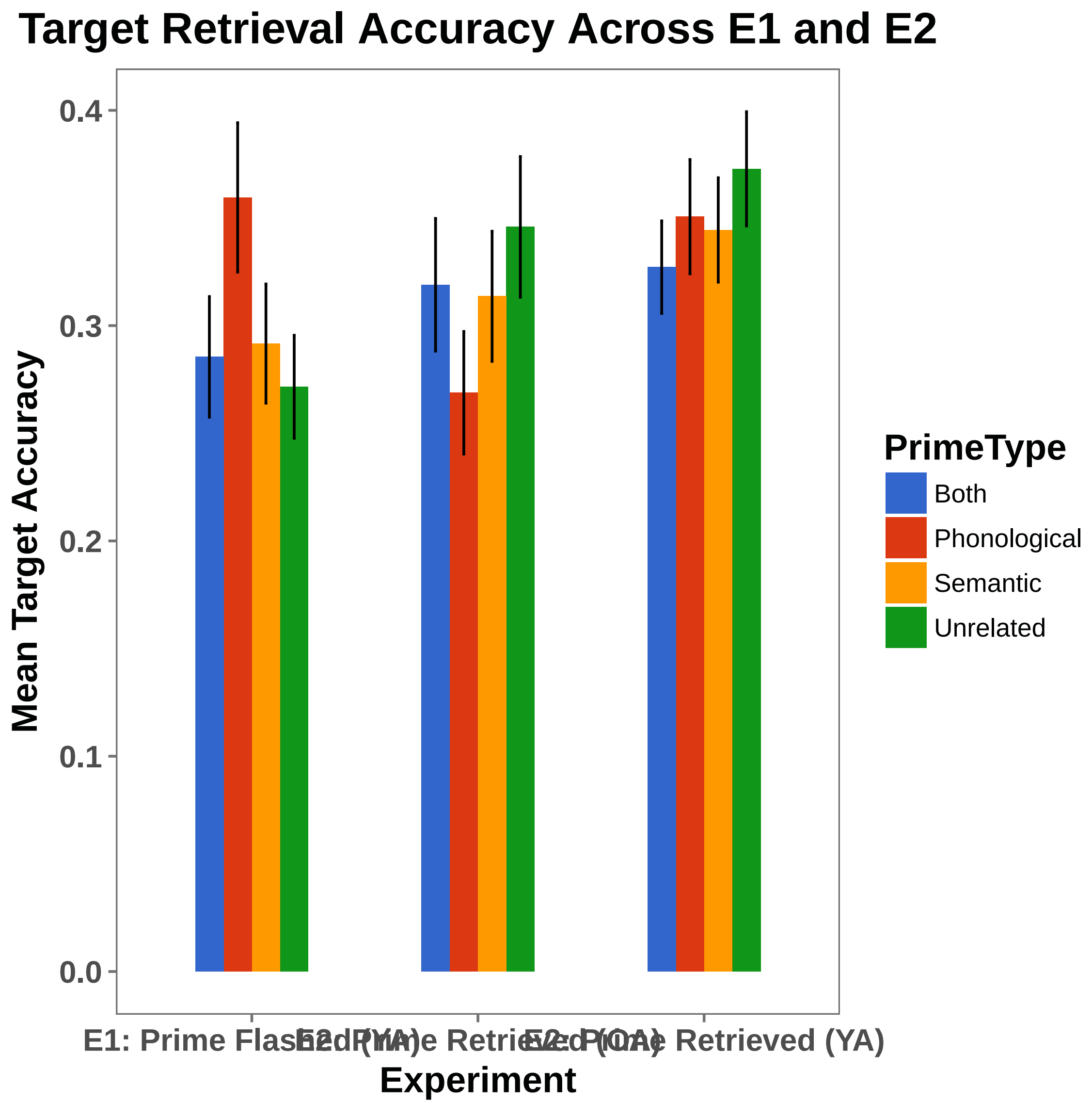
```
|   |contrast                  |PrimeCondition |  estimate|        SE|       df|
t.ratio|   p.value|
|:--|:-------------------------|:--------------|---------:|---------:|--------:|--------:
|2  |PrimeFlash - PrimeDemask  |P              | 0.0997024| 0.0580068| 174.7432| 1.718806
```

3 Figures

Target Accuracy Figure

```
> target_rmisc = Rmisc::summarySE(prime_subject_acc,
+                       measurevar = "TargetAccuracy",
+                       groupvars = c("Experiment","PrimeCondition"))
> target_rmisc_e1e2 = target_rmisc %>% filter(Experiment != "PrimeDemask")
> target_rmisc_e1e2$`Experiment Name` = ifelse(target_rmisc_e1e2$Experiment == "PrimeRet
+                                              "E2: Prime Retrieved (YA)",
+                                 ifelse(target_rmisc_e1e2$Experiment == "PrimeRetriev
+                                        "E2: Prime Retrieved (OA)", "E1: Prime Flashed (
> library(ggplot2)
> library(ggthemes)
> target_rmisc_e1e2 %>% mutate(PrimeType = factor(PrimeCondition,
+                                                 levels = unique(PrimeCondition),
+                     labels = c("Both", "Phonological",
+                                "Semantic", "Unrelated"))) %>%
+ ggplot(aes(x = `Experiment Name`, y = TargetAccuracy,
```

```
+             group = PrimeType, fill = PrimeType))+
+  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_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))
```



4 Comparing TOT Unrelated and TOT Semantic

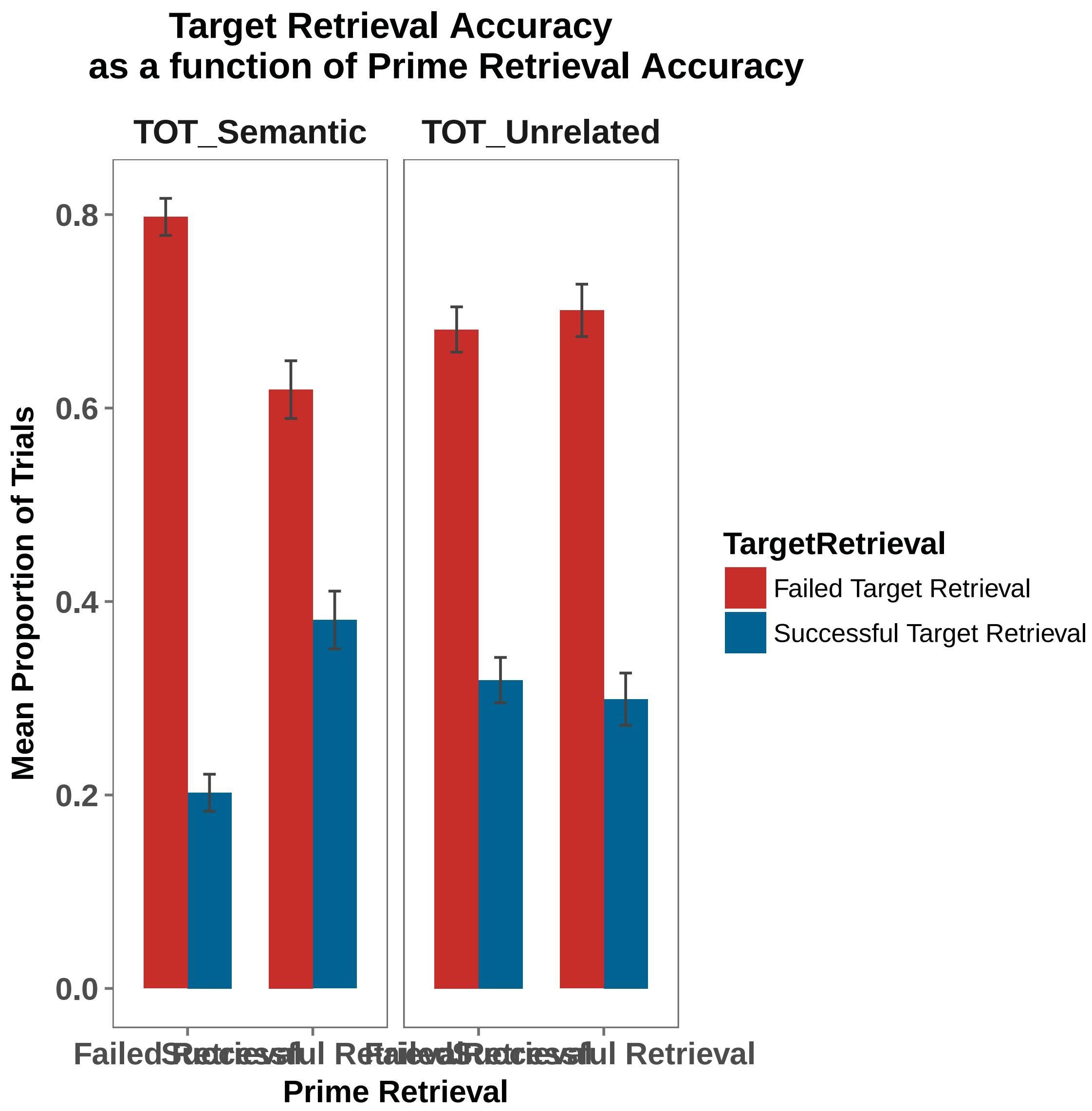
```
> US = read.csv("TOTUnrelatedAndSemantic.csv", header = TRUE, sep = ",")
> library(dplyr)
> cued_acc = group_by(US, ExperimentName) %>%
+   summarise_at(vars(PrimeFirstResp_ACC, TargetFirstResp_ACC), mean)
> cued_acc = group_by(US, ExperimentName, Subject, PrimeFirstResp_ACC) %>%
+   summarise(recalltrials = 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 1.75e-31 1.752e-31   0.522  0.473
Residuals      53 1.78e-29 3.358e-31

Error: Subject:PrimeFirstResp_ACC
                                  Df  Sum Sq   Mean Sq F value Pr(>F)
PrimeFirstResp_ACC                 1 1.0e-32 1.280e-32   0.010  0.921
ExperimentName:PrimeFirstResp_ACC  1 2.0e-32 2.320e-32   0.018  0.894
Residuals                         53 6.9e-29 1.302e-30

Error: Subject:TargetFirstResp_ACC
                                   Df Sum Sq Mean Sq F value Pr(>F)
TargetFirstResp_ACC                 1  8.707   8.707 141.427 <2e-16 ***
ExperimentName:TargetFirstResp_ACC  1  0.016   0.016   0.264   0.61
Residuals                          53  3.263   0.062
---
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 0.2725  0.2725   30.66
ExperimentName:PrimeFirstResp_ACC:TargetFirstResp_ACC  1 0.5366  0.5366   60.37
Residuals                                             53 0.4711  0.0089
                                                        Pr(>F)
PrimeFirstResp_ACC:TargetFirstResp_ACC                9.77e-07 ***
ExperimentName:PrimeFirstResp_ACC:TargetFirstResp_ACC 2.61e-10 ***
Residuals
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

4.1 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
```



4.2 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)
```

4.3 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
```

4.4 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")
> US$PrimeDefRT = as.numeric(as.character(US$PrimeDefRT))
> ## aggregate per subject all IVs and DVs
> meanRT = group_by(US, ExperimentName, Subject) %>%
+   summarise_at(vars(PrimeDefRT, TargetRT, RTrecognisePrime, RTrecogniseTarget), mean)
> colnames(meanRT) = c("ExperimentName", "Subject","MeanPrimeRT", "MeanTargetRT",
+                      "MeanRTrecogPrime", "MeanRTrecogTarget")
> sdRT = group_by(US, ExperimentName,Subject) %>%
+   summarise_at(vars(PrimeDefRT, TargetRT, RTrecognisePrime, RTrecogniseTarget), sd)
> colnames(sdRT) = c("ExperimentName", "Subject","sdPrimeRT", "sdTargetRT",
+                      "sdRTrecogPrime", "sdRTrecogTarget")
> RT_agg = merge(meanRT, sdRT, by = c("ExperimentName", "Subject"))
> ## merge aggregate info with long data
> US_z = merge(US, RT_agg, by = c("ExperimentName", "Subject"), all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> US_z = US_z %>% mutate(zPrimeRT = (PrimeDefRT - MeanPrimeRT)/sdPrimeRT,
+                                            zTargetRT =
+                                              (TargetRT - MeanTargetRT)/sdTargetRT,
+                                            zPrimeRecogRT =
+                                              (RTrecognisePrime -
+                                                 MeanRTrecogPrime)/sdRTrecogPrime,
+                                            zTargetRecogRT =
+                                              (RTrecogniseTarget -
+                                                 MeanRTrecogTarget)/sdRTrecogTarget)
> ## checking: subject level means should be zero
>
> sub_pic = group_by(US_z, Subject) %>%
+   summarise_at(vars(zTargetRT,zPrimeRecogRT, zTargetRecogRT), mean)
>
```

4.5 Trimming z-RTs

```
> #Note: We are trimming based on PrimeRecog RT because that's the RT we care about most
> US_z_trimmed = subset(US_z, US_z$zPrimeRecogRT < 3 &
+                               US_z$zPrimeRecogRT > -3)
```

4.6 Repeating z-scoring

```
> ## aggregate per subject all IVs and DVs
> meanRT = group_by(US_z_trimmed, ExperimentName, Subject) %>%
+   summarise_at(vars(PrimeDefRT, TargetRT, RTrecognisePrime, RTrecogniseTarget), mean)
> colnames(meanRT) = c("ExperimentName", "Subject","MeanPrimeRT_trim", "MeanTargetRT_tri
+                      "MeanRTrecogPrime_trim", "MeanRTrecogTarget_trim")
> sdRT = group_by(US_z_trimmed, ExperimentName, Subject) %>%
+   summarise_at(vars(PrimeDefRT, TargetRT, RTrecognisePrime, RTrecogniseTarget), sd)
> colnames(sdRT) = c("ExperimentName", "Subject", "sdPrimeRT_trim","sdTargetRT_trim",
+                    "sdRTrecogPrime_trim", "sdRTrecogTarget_trim")
> RT_agg = merge(meanRT, sdRT, by = c("ExperimentName", "Subject"))
> ## merge aggregate info with long data
> US_final_z = merge(US_z_trimmed,
+                            RT_agg, by = c("ExperimentName", "Subject"), all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> US_final_z = US_final_z %>% mutate(zPrimeRT_trim =
+                                              (PrimeDefRT -
+                                                 MeanPrimeRT_trim)/sdPrimeRT_trim,
+                                              zTargetRT_trim =
+                                              (TargetRT -
+                                                 MeanTargetRT_trim)/sdTargetRT_trim,
+                                            zPrimeRecogRT_trim =
+                                              (RTrecognisePrime -
+                                       MeanRTrecogPrime_trim)/sdRTrecogPrime_trim,
+                                            zTargetRecogRT_trim =
+                                              (RTrecogniseTarget -
+                                       MeanRTrecogTarget_trim)/sdRTrecogTarget_trim)
> ## checking: subject level means should be zero
>
> sub_pic = group_by(US_final_z, Subject) %>%
+   summarise_at(vars(zTargetRT_trim,zPrimeRecogRT_trim, zTargetRecogRT_trim), mean)
>
```

4.7 Linear Models

```
> # Mean RT to retrieve Target as a function of Prime Condition
>
> # Effect of RT prime on Accuracy
> library(lme4)
```

```
> RTprime_acc_model = glmer(data = US_final_z,
+                           Accuracy ~ ExperimentName*zPrimeRecogRT_trim +
+                             (1|Subject) + (1|Target), family = binomial )
> summary(RTprime_acc_model)
```

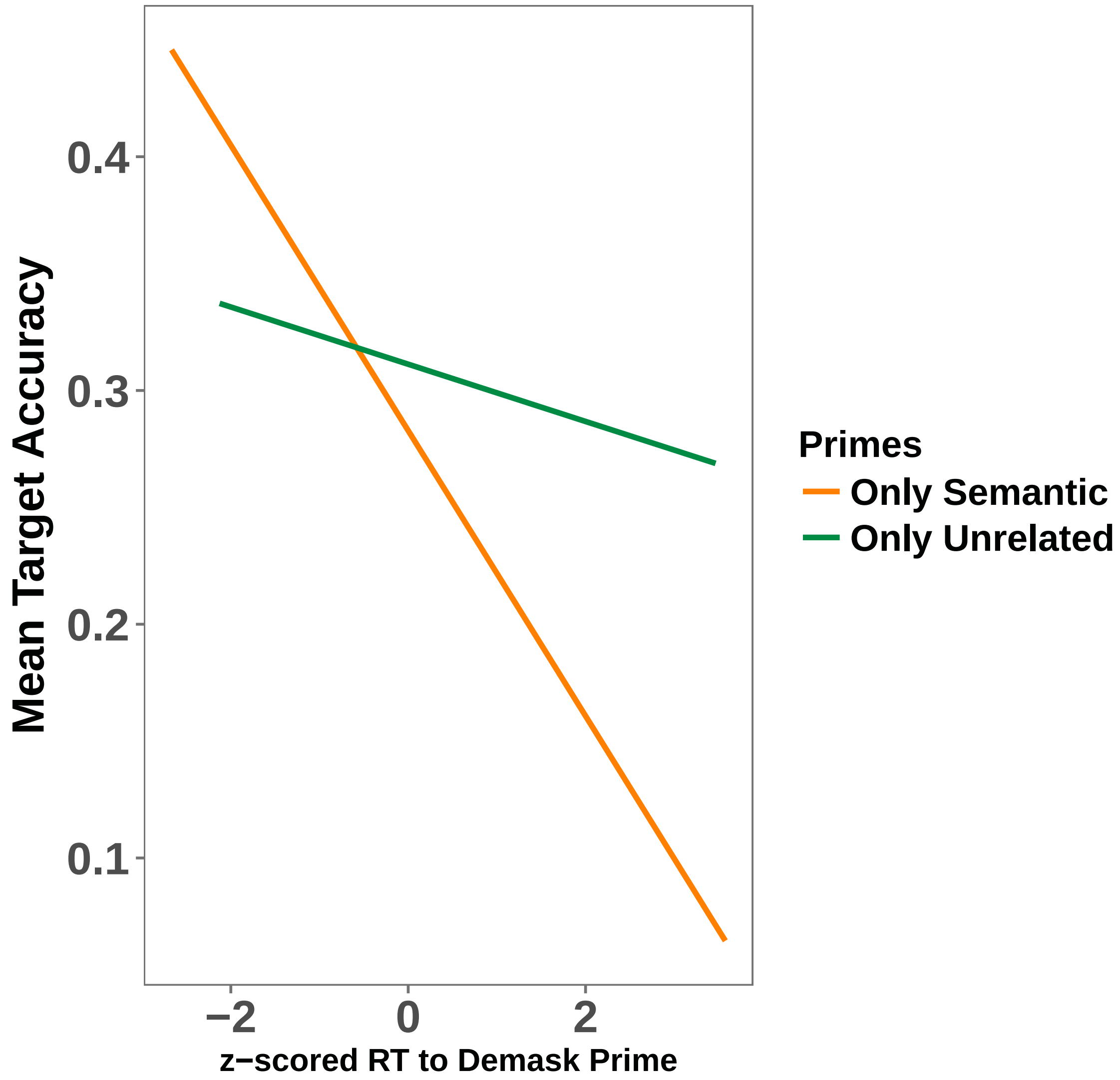
```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: Accuracy ~ ExperimentName * zPrimeRecogRT_trim + (1 | Subject) +
    (1 | Target)
   Data: US_final_z

     AIC      BIC   logLik deviance df.resid
  3670.0   3707.5  -1829.0   3658.0     3842

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.7168 -0.5182 -0.2785  0.4831  5.9119

Random effects:
 Groups  Name        Variance Std.Dev.
 Target  (Intercept) 2.2067   1.4855
 Subject (Intercept) 0.6835   0.8268
Number of obs: 3848, groups:  Target, 72; Subject, 54

Fixed effects:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                       -1.29066    0.21475  -6.010 1.86e-09 ***
ExperimentName1                   -0.11339    0.12099  -0.937  0.34865
zPrimeRecogRT_trim                -0.13001    0.04796  -2.711  0.00671 **
ExperimentName1:zPrimeRecogRT_trim -0.11226    0.04697  -2.390  0.01684 *
---
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 0.047
zPrmRcgRT_t 0.011  0.019
ExpN1:PRRT_ 0.013  0.019  0.095
```

```
>
>
```

4.7.1 Raw Data

```
> library(ggplot2)
> library(ggthemes)
```

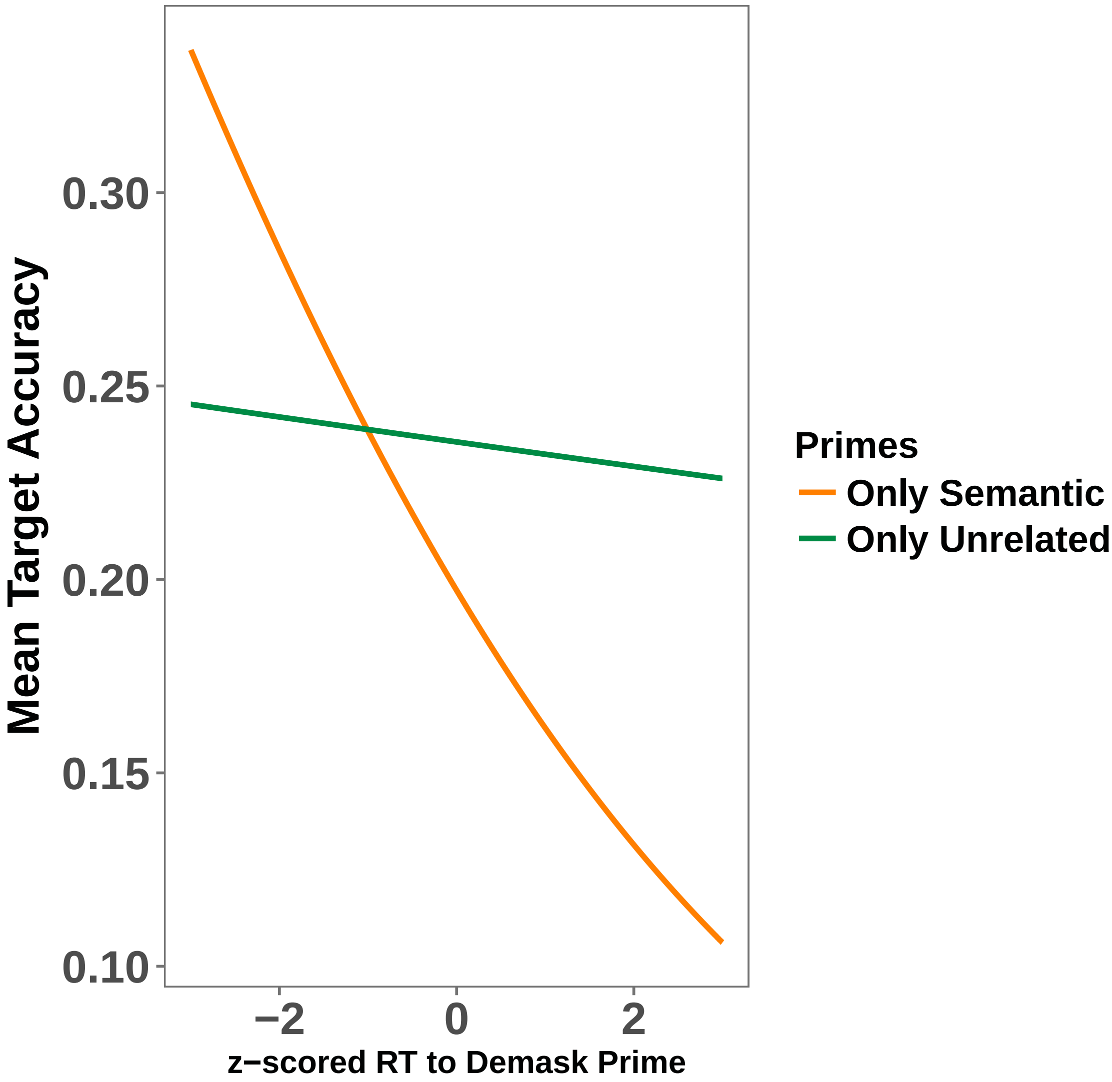
```
> US_final_z$Accuracy = as.numeric(as.character(US_final_z$Accuracy))
> mainplot = US_final_z %>%
+   mutate(Primes = factor(ExperimentName,
+                              levels = unique(ExperimentName),
+                           labels = c("Only Semantic",
+                                   "Only Unrelated")))%>%
+   ggplot(aes(x =zPrimeRecogRT_trim , y = Accuracy,
+               group = Primes, color = Primes)) +
+   geom_smooth(method = "glm", se = FALSE, 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)))
> mainplot
>
```



4.7.2 Model Plot

```
> 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)))
```



Primes
Only Semantic
Only Unrelated
Mean Target Accuracy
0.30
0.25
0.20
0.15
0.10
−2
0
2
z−scored RT to Demask Prime