# TOT Unrelated Analysis

Abhilasha Kumar

April 17, 2018

## 1 Reading the Data File

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

```
> TOTUnrelated = read.csv("TOTUnrelated_AllSubjects.csv",
+                         header = TRUE, sep = ",")
> head(TOTUnrelated)
```

```
  ExperimentName Subject Session Procedure.Block. Trial        Prime.Trial.
1  TOT_Unrelated       1       1               S1    24         o b v i o u s
2  TOT_Unrelated       1       1               S1    67 c o m p a t i b l e
3  TOT_Unrelated       1       1               S1    71               M e s s i
4  TOT_Unrelated       1       1               S1    63           M a d o n n a
5  TOT_Unrelated       1       1               S1    29             r e f u s e
6  TOT_Unrelated       1       1               S1    56 c o n v e n i e n t
  PrimeDef.RESP.Trial. PrimeDef.RT.Trial. PrimeResponse.RESP.Trial.
1              {SPACE}               3631            obvious{SPACE}
2              {SPACE}               9405         compatible{SPACE}
3              {SPACE}               3218              messi{SPACE}
4       madonna{SPACE}               7206            madonna{SPACE}
5              {SPACE}               6657             refuse{SPACE}
6              {SPACE}               6057         convenient{SPACE}
  PrimeResponse.RT.Trial. Stimuli1         Target.Trial.
1                    2169        1       a b d i c a t e
2                    3986        2     a b o r i g i n e
3                    1733        3         a b s t a i n
4                    2212        4       a d v o c a t e
5                    1589        5           A l c o t t
6                   12704        6 a n a c h r o n i s m
  TargetDefinition.RESP.Trial. TargetDefinition.RT.Trial.
1                      {SPACE}                       6267
2           aboriginal{SPACE}                        9862
3                      {SPACE}                       4281
4                      {SPACE}                       4036
5                      {SPACE}                       1983
6                      {SPACE}                       6718
```

```
  TargetQuestion.RESP.Trial. TargetQuestion.RT.Trial.
1                          2                     1309
2                          1                     1502
3                          2                     1441
4                          3                     1990
5                          2                     3049
6                          3                     2061
  TargetResponse.RESP.Trial. TargetResponse.RT.Trial. PrimeFirstResp_ACC
1           abdicate{SPACE}                     5203                  0
2          aborigine{SPACE}                     3693                  0
3            abstain{SPACE}                     3112                  0
4           advocate{SPACE}                     4358                  1
5             alcott{SPACE}                     2463                  0
6        anachronism{SPACE}                     3717                  0
  TargetFirstResp_ACC RTrecognisePrime RTrecogniseTarget
1                   0             1692              5230
2                   1             2384              7322
3                   0             3718              1609
4                   0             2886              1706
5                   0             1821              3464
6                   0             1550              3652
```

## 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(TOTUnrelated) %>%
+   summarise_at(vars(PrimeFirstResp_ACC, TargetFirstResp_ACC), mean)
> cued_acc = group_by(TOTUnrelated, Subject, PrimeFirstResp_ACC) %>%
+   summarise(recalltrials = n())
> conditional_acc = group_by(TOTUnrelated, Subject,
+                            PrimeFirstResp_ACC, TargetFirstResp_ACC) %>%
+   summarise(trials = n())
> merge_acc = merge(conditional_acc, cued_acc,
+                   by = c("Subject", "PrimeFirstResp_ACC"))
> 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.

```
> ## mergeacc has no row missing
>
```

```
>
> 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 ~ PrimeFirstResp_ACC*TargetFirstResp_ACC +
+         Error(Subject/(PrimeFirstResp_ACC*TargetFirstResp_ACC)))
> summary(cond_aov)
```

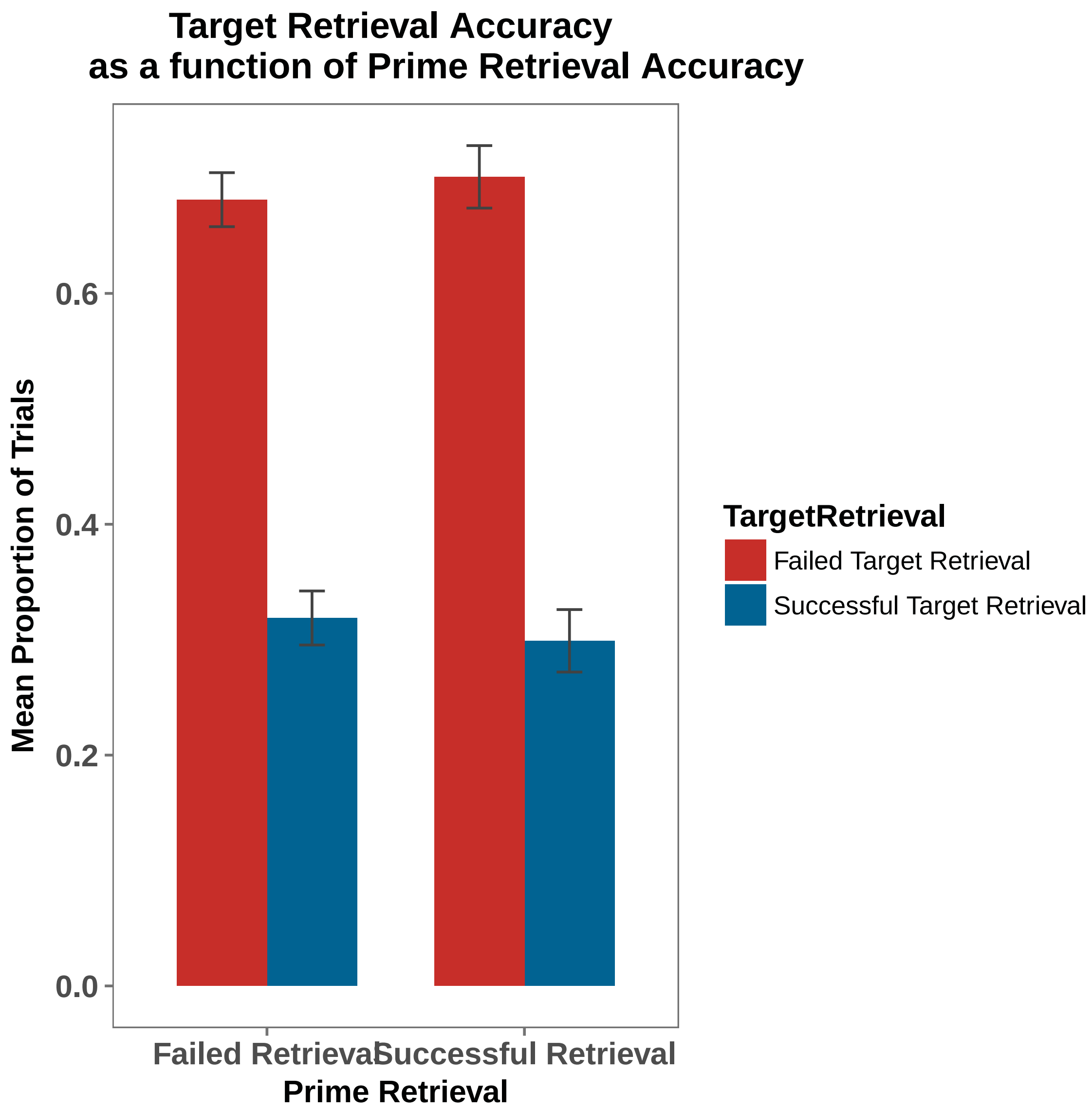
```
Error: Subject
          Df    Sum Sq   Mean Sq F value Pr(>F)
Residuals 29 6.619e-30 2.282e-31

Error: Subject:PrimeFirstResp_ACC
                   Df    Sum Sq   Mean Sq F value Pr(>F)
PrimeFirstResp_ACC  1 4.200e-31 4.211e-31   0.376  0.545
Residuals          29 3.252e-29 1.121e-30

Error: Subject:TargetFirstResp_ACC
                    Df Sum Sq Mean Sq F value Pr(>F)
TargetFirstResp_ACC  1  4.382   4.382   63.27  9e-09 ***
Residuals           29  2.009   0.069
---
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 Pr(>F)
PrimeFirstResp_ACC:TargetFirstResp_ACC  1 0.01174 0.011738   1.554  0.223
Residuals                              29 0.21909 0.007555
```

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("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()+
+   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
```




## 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(2,4,5)]
> successfulrecall = merge_acc %>% filter(PrimeFirstResp_ACC == "1")
> successfulrecall = successfulrecall[,-c(2,4,5)]
> ## need to convert from long to wide: using spread
> library(tidyr)
```

```
> failed_wide = failedrecall %>%
+   spread(TargetFirstResp_ACC, prop)
> failed_wide$diff = failed_wide$`0` - failed_wide$`1`
> successful_wide = successfulrecall %>%
+   spread(TargetFirstResp_ACC, prop)
> successful_wide$diff = successful_wide$`0` - successful_wide$`1`
```

Now we have two datasets, each contains a difference score for each subject, for failed and successful cued recall. Now, we can perform a paired t-test (why paired? because the data for failed and successful recall comes from the same subjects i.e., it is a within-subjects design).

```
> t.test(failed_wide$diff, successful_wide$diff, paired = TRUE)
```

```
        Paired t-test

data:  failed_wide$diff and successful_wide$diff
t = -1.2465, df = 29, p-value = 0.2226
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.10447172  0.02535091
sample estimates:
mean of the differences
            -0.03956041
```

# 6 HLM Model

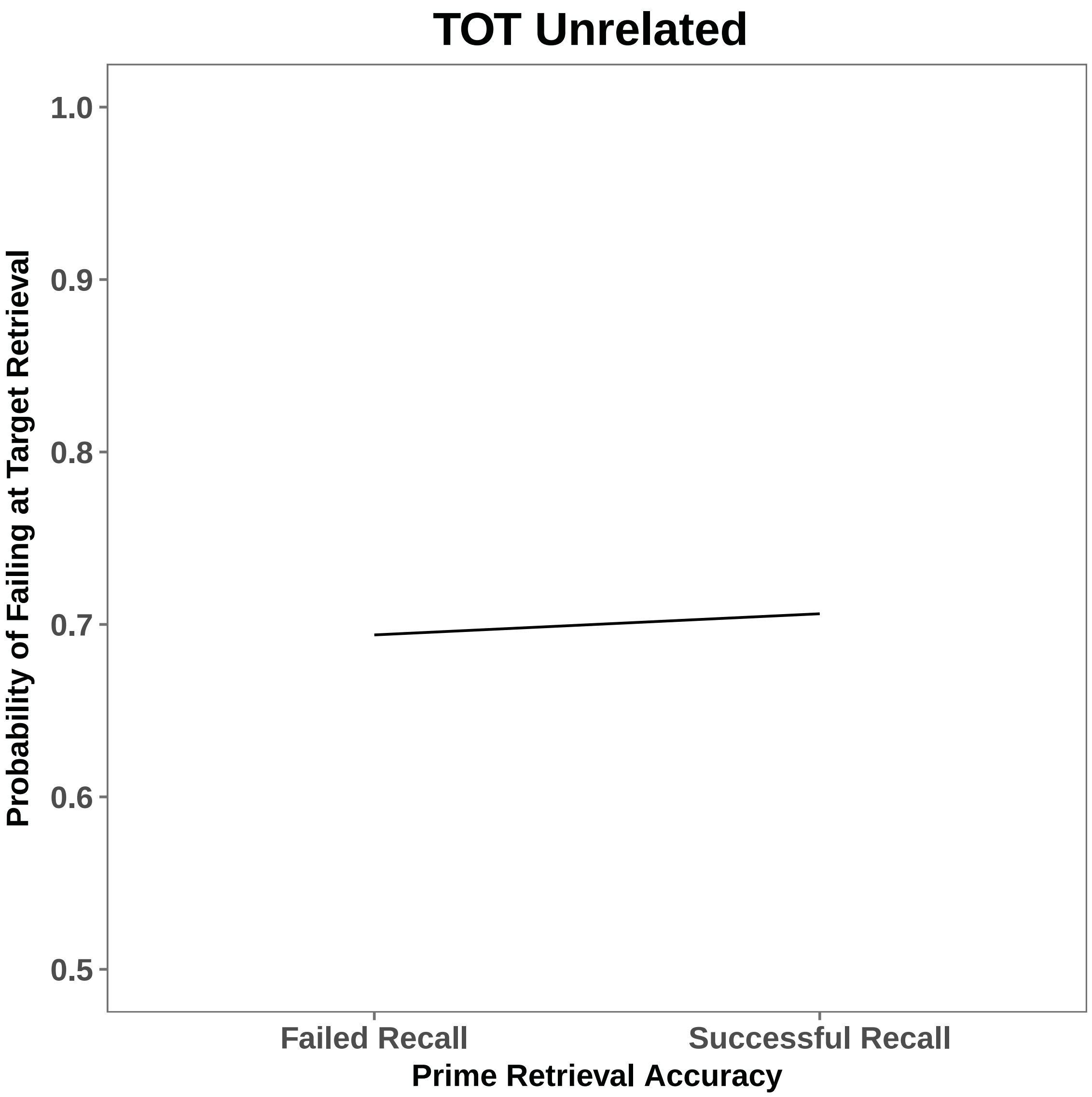
```
> library(lme4)
> TOTUnrelated$TargetFirstResp_ACC = as.factor(TOTUnrelated$TargetFirstResp_ACC)
> TOTUnrelated$PrimeFirstResp_ACC = as.factor(TOTUnrelated$PrimeFirstResp_ACC)
> TOTUnrelated$FailedRetrieval = ifelse(TOTUnrelated$TargetFirstResp_ACC == 1,0,1)
> totunrelated_hlm = glmer(data = TOTUnrelated, FailedRetrieval ~ PrimeFirstResp_ACC +
+                         (1|Subject), family = "binomial")
> summary(totunrelated_hlm)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: FailedRetrieval ~ PrimeFirstResp_ACC + (1 | Subject)
   Data: TOTUnrelated

     AIC      BIC   logLik deviance df.resid
  2605.3   2622.3  -1299.6   2599.3     2157

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.4041 -1.1549  0.5472  0.7061  1.1228
```

```
Random effects:
 Groups  Name        Variance Std.Dev.
 Subject (Intercept) 0.3012   0.5488
Number of obs: 2160, groups:  Subject, 30

Fixed effects:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          0.81853    0.12102   6.764 1.35e-11 ***
PrimeFirstResp_ACC1  0.05841    0.09768   0.598     0.55
---
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)
PrmFrR_ACC1 -0.389
```

#### 6.0.1 Plot

```
> fixed.frame <-
+   data.frame(expand.grid( PrimeFirstResp_ACC = c("0","1"))) %>%
+   mutate(pred = predict(totunrelated_hlm, newdata = ., re.form = NA))
> fixed.frame$odds = exp(fixed.frame$pred)
> fixed.frame$prob = fixed.frame$odds/(1+ fixed.frame$odds)
> fixed.frame$failure = 1 - fixed.frame$prob
> fixed.frame %>%
+   mutate(PrimeRetrieval = factor(PrimeFirstResp_ACC,
+     levels = unique(PrimeFirstResp_ACC),
+                     labels = c("Failed Recall", "Successful Recall")))%>%
+ ggplot(aes(x = PrimeRetrieval, y = prob))+
+   geom_line(group = 1)+
+   ylim(.5,1)+
+ # geom_bar(stat = "identity", position = "dodge",
+  #          width = 0.7, color = "black")+
+  theme_few()+
+   xlab("Prime Retrieval Accuracy") + ylab("Probability of Failing at Target Retrieval"
+   ggtitle("TOT Unrelated ")  +
+    theme(axis.text = element_text(face = "bold", size = rel(1)),
+           axis.title = element_text(face = "bold", size = rel(1)),
+           legend.title = element_text(face = "bold", size = rel(1)),
+      plot.title = element_text(face = "bold", size = rel(1.5), hjust = .5),
+          strip.text.x = element_text(face = "bold", size = rel(1.4)))
>
```




## 7 z-scoring RTs

### RT prime and Target

```
> library(dplyr)
> colnames(TOTUnrelated) = c("Experiment", "Subject", "Session",
+                            "Procedure", "Trial", "Prime", "PrimeDefResp",
+                             "PrimeDefRT", "PrimeResp",
+                            "PrimeRespRT", "Stimuli1",
+                            "Target", "TargetDefResp", "TargetRT",
```

```
+                           "State", "StateRT", "TargetResp", "TargetRespRT",
+                           "PrimeAcc", "Accuracy",
+                           "RTrecognisePrime", "RTrecogniseTarget",
+                          "FailedRetrieval")
> ## aggregate per subject all IVs and DVs
> meanRT = group_by(TOTUnrelated, Subject) %>%
+   summarise_at(vars(PrimeDefRT, TargetRT, RTrecognisePrime, RTrecogniseTarget), mean)
> colnames(meanRT) = c("Subject","MeanPrimeRT", "MeanTargetRT",
+                      "MeanRTrecogPrime", "MeanRTrecogTarget")
> sdRT = group_by(TOTUnrelated, Subject) %>%
+   summarise_at(vars(PrimeDefRT, TargetRT, RTrecognisePrime, RTrecogniseTarget), sd)
> colnames(sdRT) = c("Subject","sdPrimeRT", "sdTargetRT",
+                      "sdRTrecogPrime", "sdRTrecogTarget")
> RT_agg = merge(meanRT, sdRT, by = "Subject")
> ## merge aggregate info with long data
> TOTUnrelated_z = merge(TOTUnrelated, RT_agg, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> TOTUnrelated_z = TOTUnrelated_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(TOTUnrelated_z, Subject) %>%
+   summarise_at(vars(zTargetRT,zPrimeRecogRT, zTargetRecogRT), mean)
>
```

# 8 RT distributions

### RT to Demask Prime

```
> library(ggplot2)
> library(ggthemes)
> ## RT to demask prime
> # ggplot(TOTUnrelated_z, aes(x = RTrecognisePrime))+
> #   geom_histogram(binwidth = 500, color = "gray26", fill = "goldenrod")+
> #   theme_few()+
> #   xlab("RT to recognise Prime") + ylab("Count") +
> # ggtitle("Raw RT to Recognize Prime")
> #
```

```
> # ggplot(TOTUnrelated_z, aes(x = zPrimeRecogRT))+
> #  geom_histogram(binwidth = 0.2, color = "gray26", fill = "goldenrod")+
> #  theme_few()+
> #  xlab("z-RT to recognise Prime") + ylab("Count") +
> # ggtitle("z-RT to Recognize Prime")
```

### RT to Retrieve Target

```
> ## RT to retrieve target
> # ggplot(TOTUnrelated_z, aes(x = TargetRT))+
> #  geom_histogram(binwidth = 100, color = "gray26", fill = "goldenrod")+
> #  theme_few()+
> #  xlab("RT to Retrieve Target (ms)") + ylab("Count") +
> # ggtitle("Raw RT to Retrieve Target")
> #
> # ggplot(TOTUnrelated_z, aes(x = zTargetRT))+
> #  geom_histogram(binwidth = 0.1, color = "gray26", fill = "goldenrod")+
> #  theme_few()+
> #  xlab("z-RT to Retrieve Target") + ylab("Count") +
> # ggtitle("z-RT to Retrieve Target")
>
```

### RT to Demask Target

```
> ## RT to demask target
> # ggplot(TOTUnrelated_z, aes(x = RTrecogniseTarget))+
> #  geom_histogram(binwidth = 200, color = "gray26", fill = "goldenrod")+
> #  theme_few()+
> #  xlab("RT to Retrieve Target (ms)") + ylab("Count") +
> # ggtitle("Raw RT to Recognize Target")
> #
> # ggplot(TOTUnrelated_z, aes(x = zTargetRecogRT))+
> #  geom_histogram(binwidth = 0.1, color = "gray26", fill = "goldenrod")+
> #  theme_few()+
> #  xlab("z-RT to Retrieve Target") + ylab("Count") +
> # ggtitle("z-RT to Recognize Target")
>
```

## 9 Trimming z-RTs

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

## 10 Repeating z-scoring

```
> ## aggregate per subject all IVs and DVs
> meanRT = group_by(TOTUnrelated_z_trimmed, Subject) %>%
+   summarise_at(vars(PrimeDefRT, TargetRT, RTrecognisePrime, RTrecogniseTarget), mean)
> colnames(meanRT) = c("Subject","MeanPrimeRT_trim", "MeanTargetRT_trim",
+                      "MeanRTrecogPrime_trim", "MeanRTrecogTarget_trim")
> sdRT = group_by(TOTUnrelated_z_trimmed, Subject) %>%
+   summarise_at(vars(PrimeDefRT, TargetRT, RTrecognisePrime, RTrecogniseTarget), sd)
> colnames(sdRT) = c("Subject", "sdPrimeRT_trim","sdTargetRT_trim",
+                      "sdRTrecogPrime_trim", "sdRTrecogTarget_trim")
> RT_agg = merge(meanRT, sdRT, by = "Subject")
> ## merge aggregate info with long data
> TOTUnrelated_final_z = merge(TOTUnrelated_z_trimmed,
+                              RT_agg, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> TOTUnrelated_final_z = TOTUnrelated_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(TOTUnrelated_final_z, Subject) %>%
+   summarise_at(vars(zTargetRT_trim,zPrimeRecogRT_trim, zTargetRecogRT_trim), mean)
>
```

## 11 Final RT distributions

```
> # ggplot(TOTUnrelated_final_z, aes(x = zPrimeRecogRT_trim))+
> #  geom_histogram(binwidth = 0.2, color = "gray26", fill = "goldenrod")+
> #  theme_few()+
> #  xlab("z-RT to recognise Prime") + ylab("Count") +
> # ggtitle("z-RT to Recognize Prime")
> #
> # ggplot(TOTUnrelated_final_z, aes(x = zTargetRT_trim))+
> #  geom_histogram(binwidth = 0.2, color = "gray26", fill = "goldenrod")+
> #  theme_few()+
```

```
> #  xlab("z-RT to recognise Target") + ylab("Count") +
> # ggtitle("z-RT to Recognize Target")
> #
> # ggplot(TOTUnrelated_final_z, aes(x = zTargetRecogRT_trim))+
> #  geom_histogram(binwidth = 0.2, color = "gray26", fill = "goldenrod")+
> #  theme_few()+
> #  xlab("z-RT to Retrieve Target") + ylab("Count") +
> # ggtitle("z-RT to Retrieve Target")
>
```

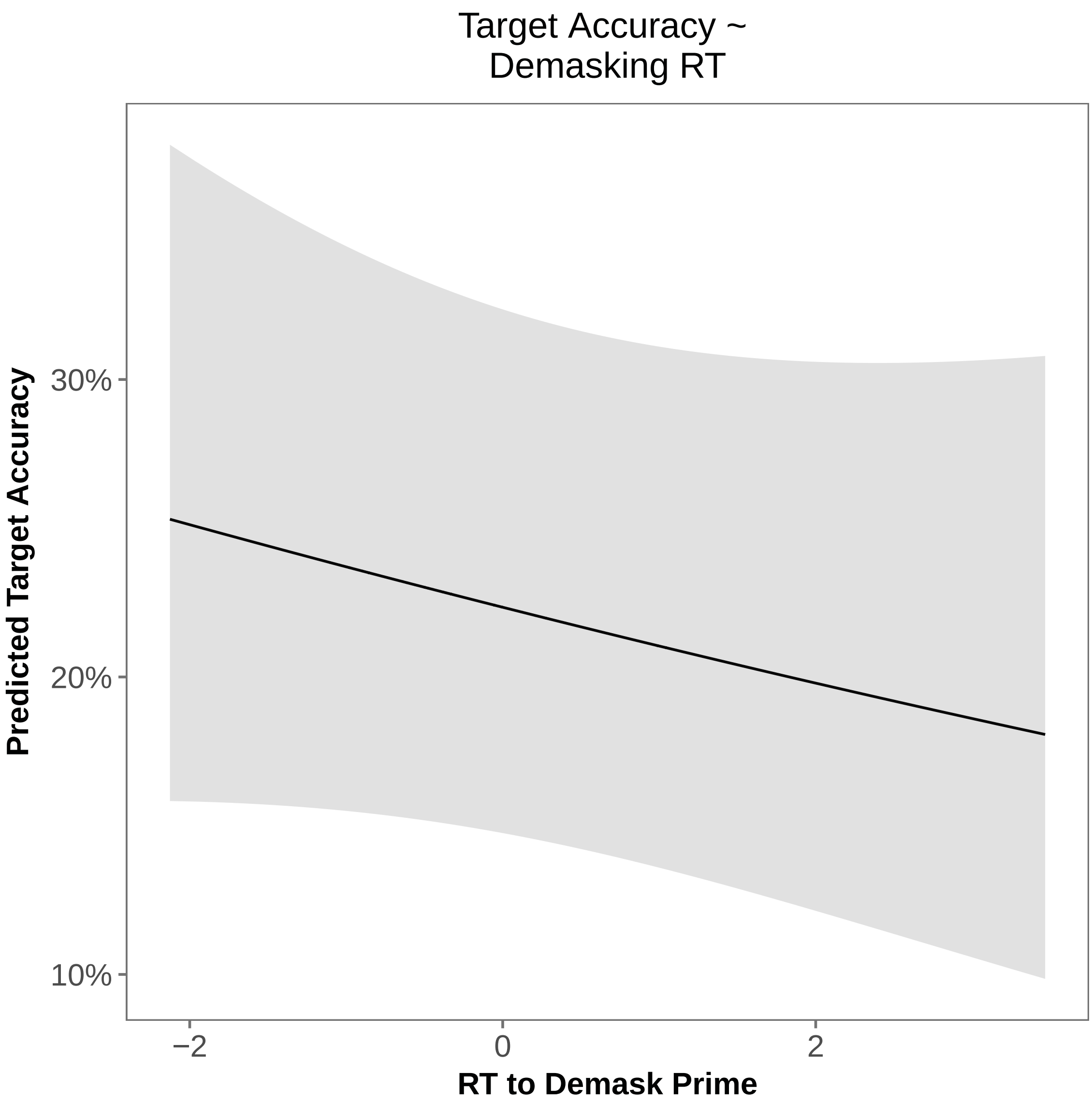
## 12 Linear Models

```
> # Mean RT to retrieve Target as a function of Prime Condition
>
> # Effect of RT prime on Accuracy
> TOTUnrelated_final_z = TOTUnrelated_final_z %>%
+                          filter(!Subject %in% c(26))
> library(lme4)
> RTprime_acc_model = glmer(data = TOTUnrelated_final_z,
+                          Accuracy ~ 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 ~ zPrimeRecogRT_trim + (1 | Subject) + (1 | Target)
   Data: TOTUnrelated_final_z

     AIC      BIC   logLik deviance df.resid
  2008.4   2031.0  -1000.2   2000.4     2062

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.8422 -0.5189 -0.2665  0.4814  4.2090

Random effects:
 Groups  Name        Variance Std.Dev.
 Target  (Intercept) 2.7396   1.6552
 Subject (Intercept) 0.6967   0.8347
Number of obs: 2066, groups:  Target, 72; Subject, 29

Fixed effects:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1.24578    0.25939  -4.803 1.56e-06 ***
zPrimeRecogRT_trim -0.07675    0.07119  -1.078    0.281
```

```
---
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)
zPrmRcgRT_t 0.001
```

```
> t = sjPlot::plot_model(RTprime_acc_model, type = "eff",
+                        terms = "zPrimeRecogRT_trim")
> t + theme_few()+
+       xlab("RT to Demask Prime") + ylab("Predicted Target Accuracy") +
+ ggtitle("Target Accuracy ~ \nDemasking RT") +
+   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)))
>
```

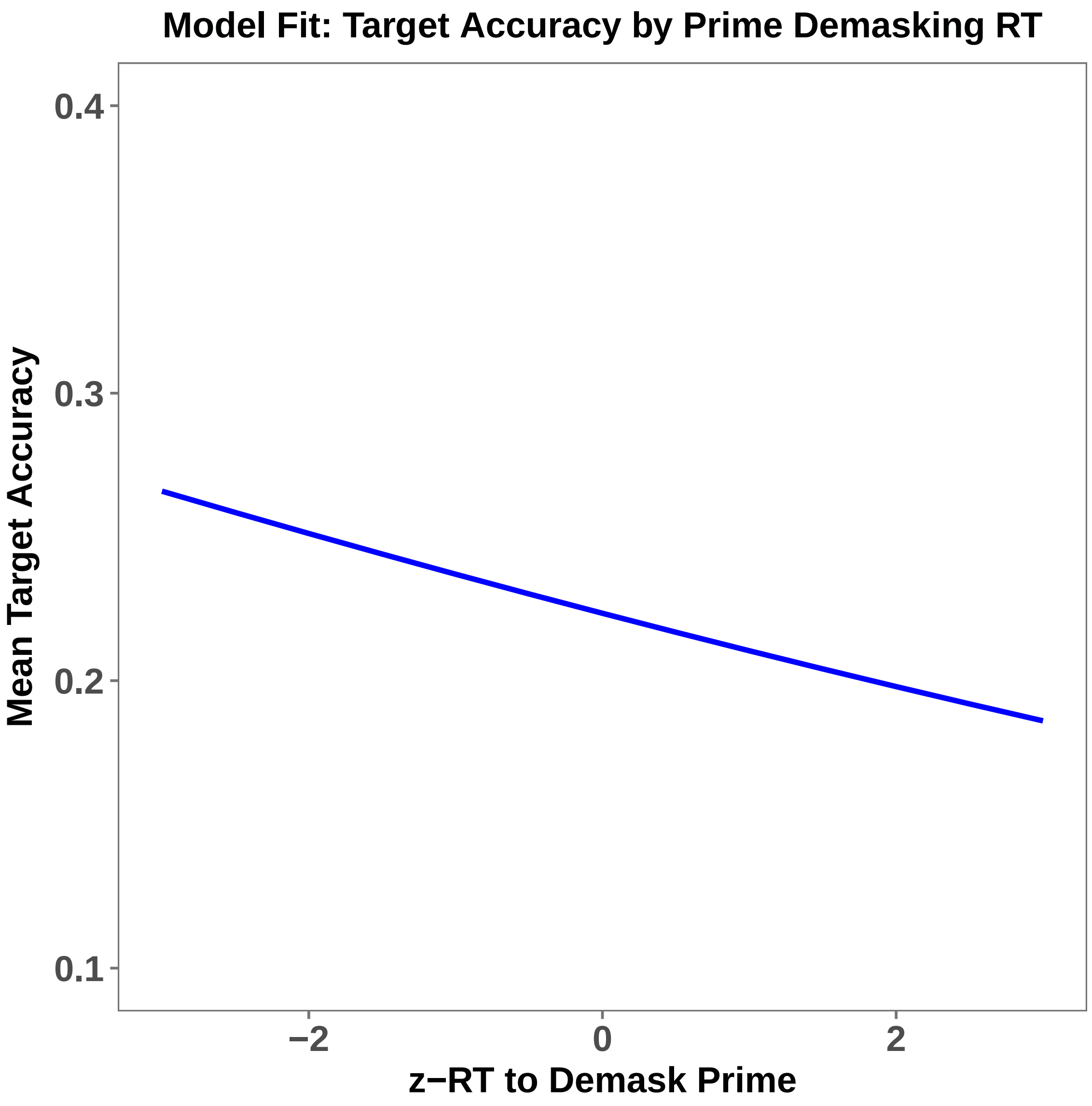



# 13 Plotting Model Fits

## 13.1 Model 1

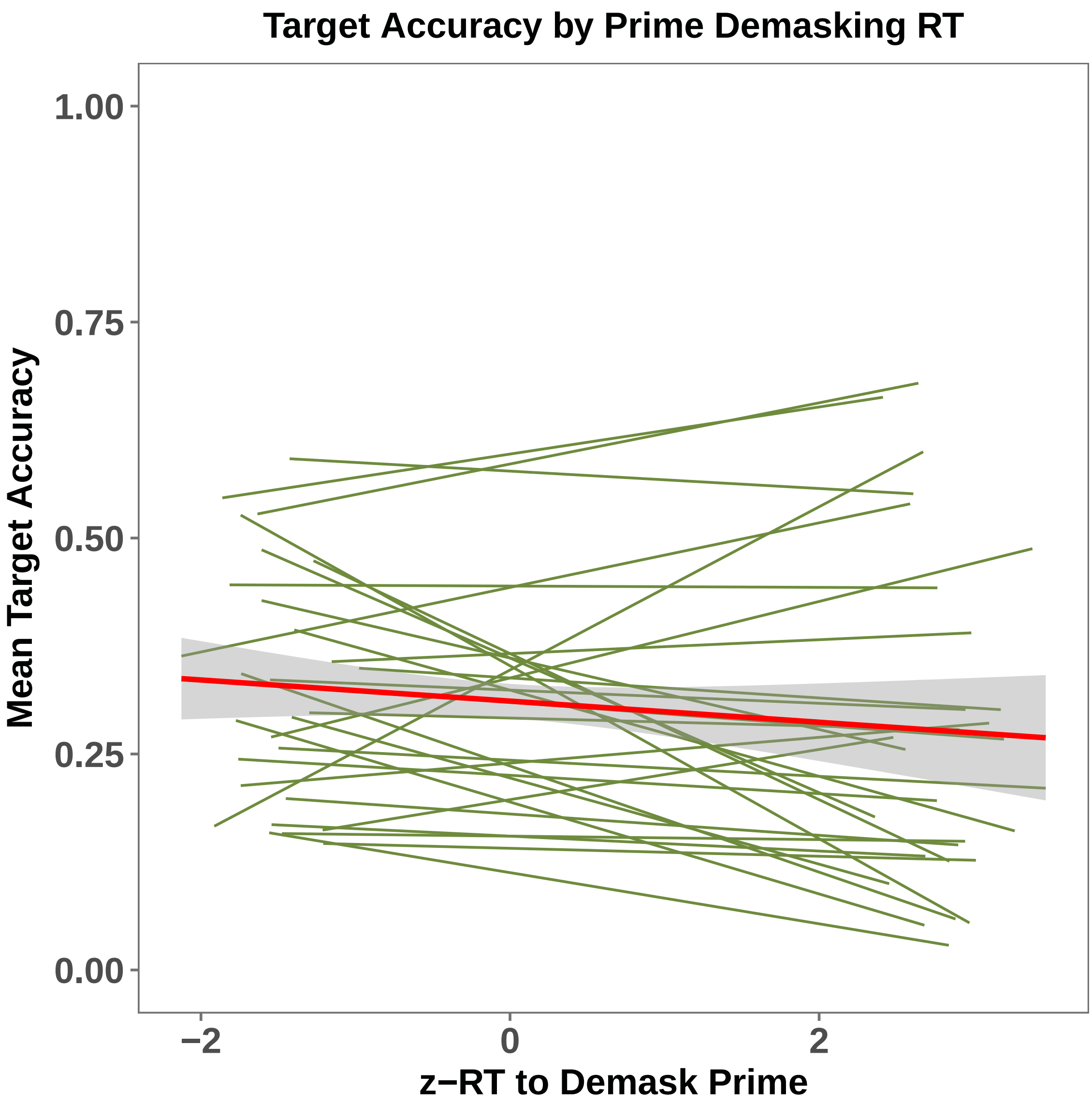
```
> fixed.frame ←
+   data.frame(expand.grid(zPrimeRecogRT_trim = seq(-3,3,0.5)))%>%
+   mutate(pred = predict(RTprime_acc_model, newdata = ., re.form = NA))
> fixed.frame$odds = exp(fixed.frame$pred)
> fixed.frame$prob = fixed.frame$odds/(1+fixed.frame$odds)
> fixed.frame %>%
```

```
+   ggplot(aes(x = zPrimeRecogRT_trim, y = prob)) +
+     geom_line(size = 1, color = "blue") +
+   ylim(0.10,0.40)+
+     xlab("z-RT to Demask Prime") + ylab ("Mean Target Accuracy")+
+   ggtitle("Model Fit: Target Accuracy by Prime Demasking RT")+
+ theme_few() +
+     theme(axis.text = element_text(face = "bold", size = rel(1.2)),
+           axis.title = element_text(face = "bold", size = rel(1.2)),
+           legend.title = element_text(face = "bold", size = rel(1.2)),
+           plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
```



### 13.2 Raw Data

```
> library(ggplot2)
> library(ggthemes)
> TOTUnrelated_final_z$Accuracy = as.numeric(as.character(TOTUnrelated_final_z$Accuracy))
> mainplot = TOTUnrelated_final_z %>%
+   ggplot(aes(x =zPrimeRecogRT_trim , y = Accuracy,
+              group = factor(Subject))) +
+   geom_smooth(method = "lm", se = FALSE, color = "darkolivegreen4", size = 0.5)+
+   guides(color = FALSE)+
+     xlab("z-RT to Demask Prime") + ylab ("Mean Target Accuracy")+
+   ggtitle("Target Accuracy by Prime Demasking RT")+
+ theme_few() +
+   ylim(0,1)+
+     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))
> mainplot + stat_smooth(aes(group = 1), method = "lm", color = "red")
>
```



Target Accuracy by Prime Demasking RT
Mean Target Accuracy
1.00
0.75
0.50
0.25
0.00
−2
0
2
z−RT to Demask Prime