TOT Cued Recall 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.

```
> TOTcuedrecall = read.csv("Compiled_TOTCuedRecall.csv",
+                          header = TRUE, sep = ",")
> head(TOTcuedrecall[,c(1,21,22)])
```

```
  Subject CuedRecallAcc TargetAccuracy
1       1             0              1
2       1             0              0
3       1             0              0
4       1             0              0
5       1             0              0
6       1             0              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(TOTcuedrecall) %>%
+   summarise_at(vars(CuedRecallAcc, TargetAccuracy), mean)
> cued_acc = group_by(TOTcuedrecall, Subject, CuedRecallAcc) %>%
+   summarise(recalltrials = n())
> conditional_acc = group_by(TOTcuedrecall, Subject,
+                            CuedRecallAcc, TargetAccuracy) %>%
+   summarise(trials = n())
> merge_acc = merge(conditional_acc, cued_acc,
+                   by = c("Subject", "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.

```
> ## mergeacc has 3 rows missing
> ez::ezDesign(merge_acc, Subject, CuedRecallAcc)
> merge_acc[101,] = c(12, 0, 0, 0, 0, 0)
> merge_acc[102,] = c(17, 1, 0, 0 ,0, 0)
> merge_acc[103,] = c(18, 0, 0, 0, 0, 0)
> merge_acc[104,] = c(21, 0, 0, 0, 0, 0)
> 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))
> cond_aov = aov(data = merge_acc,
+         prop ~ CuedRecallAcc*TargetAccuracy +
+         Error(Subject/(CuedRecallAcc*TargetAccuracy)))
> summary(cond_aov)
```

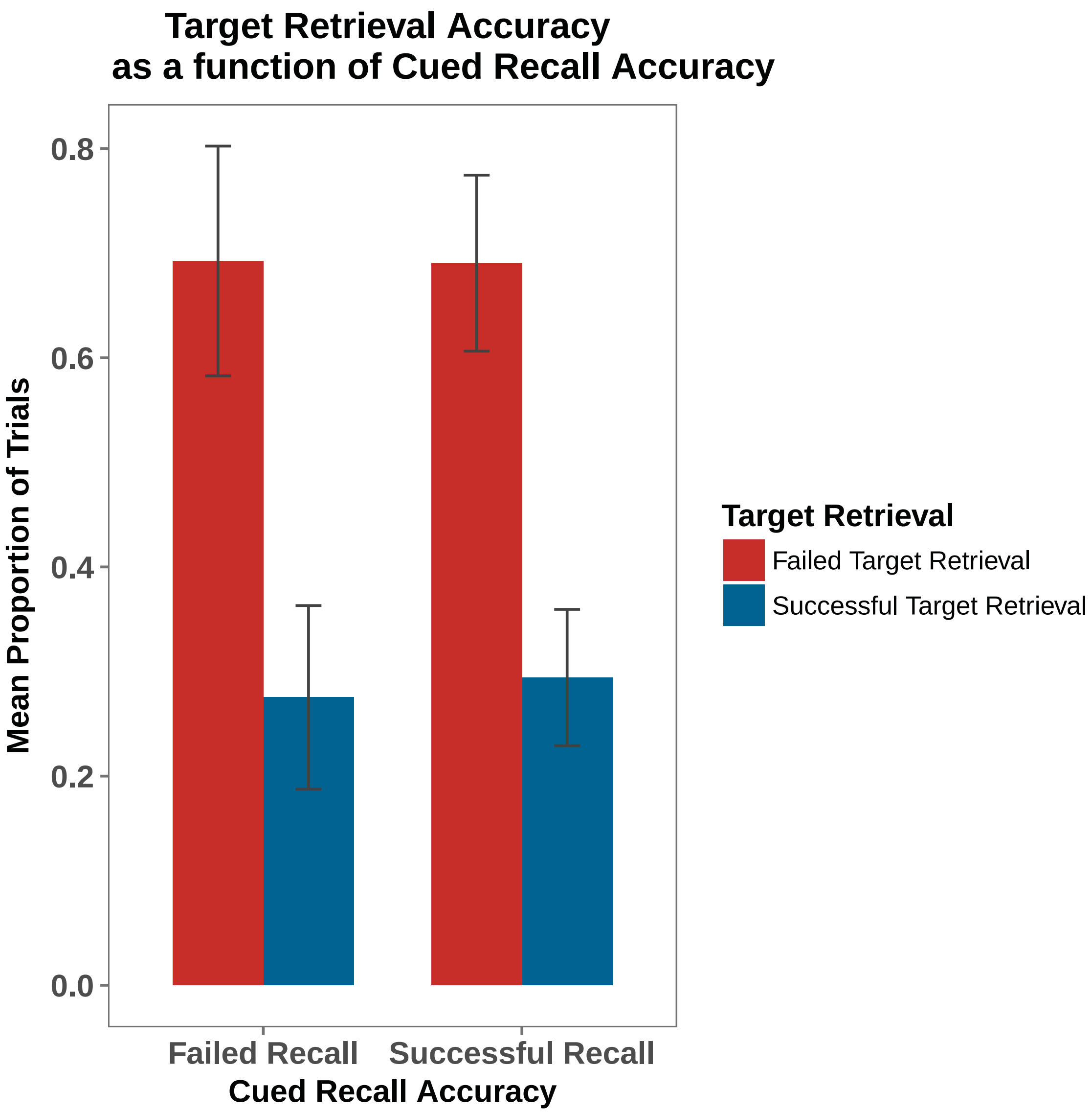
```
Error: Subject
                             Df    Sum Sq   Mean Sq F value Pr(>F)
TargetAccuracy                1 4.620e-32 4.617e-32   0.546  0.467
CuedRecallAcc:TargetAccuracy  1 9.780e-32 9.783e-32   1.158  0.293
Residuals                    23 1.944e-30 8.452e-32

Error: Subject:CuedRecallAcc
                             Df  Sum Sq   Mean Sq F value Pr(>F)
CuedRecallAcc                 1 2.0e-32 2.000e-32   0.015  0.904
TargetAccuracy                1 3.0e-32 3.010e-32   0.022  0.883
CuedRecallAcc:TargetAccuracy  1 3.5e-32 3.530e-32   0.026  0.873
Residuals                    23 3.1e-29 1.348e-30

Error: Subject:TargetAccuracy
                             Df Sum Sq Mean Sq F value   Pr(>F)
TargetAccuracy                1  4.531   4.531   47.57 3.93e-07 ***
CuedRecallAcc:TargetAccuracy  1  0.000   0.000    0.00    0.984
Residuals                    24  2.286   0.095
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: Subject:CuedRecallAcc:TargetAccuracy
                             Df Sum Sq Mean Sq F value Pr(>F)
CuedRecallAcc:TargetAccuracy  1 0.0091 0.00905   0.222  0.642
Residuals                    22 0.8978 0.04081

Error: Within
```

```
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  3    1.5     0.5
```

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("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))+
+  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
```



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(CuedRecallAcc == "0")
> # failedrecall = failedrecall[,-c(2,4,5)]
> # successfulrecall = merge_acc %>% filter(CuedRecallAcc == "1")
> # successfulrecall = successfulrecall[,-c(2,4,5)]
> #
> # ## need to convert from long to wide: using spread
```

```
> # library(tidyr)
> # failed_wide = failedrecall %>%
> #   spread(TargetAccuracy, prop)
> # failed_wide$diff = failed_wide$`0` - failed_wide$`1`
> #
> # successful_wide = successfulrecall %>%
> #   spread(TargetAccuracy, prop)
> # successful_wide$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)
```

6 HLM Model

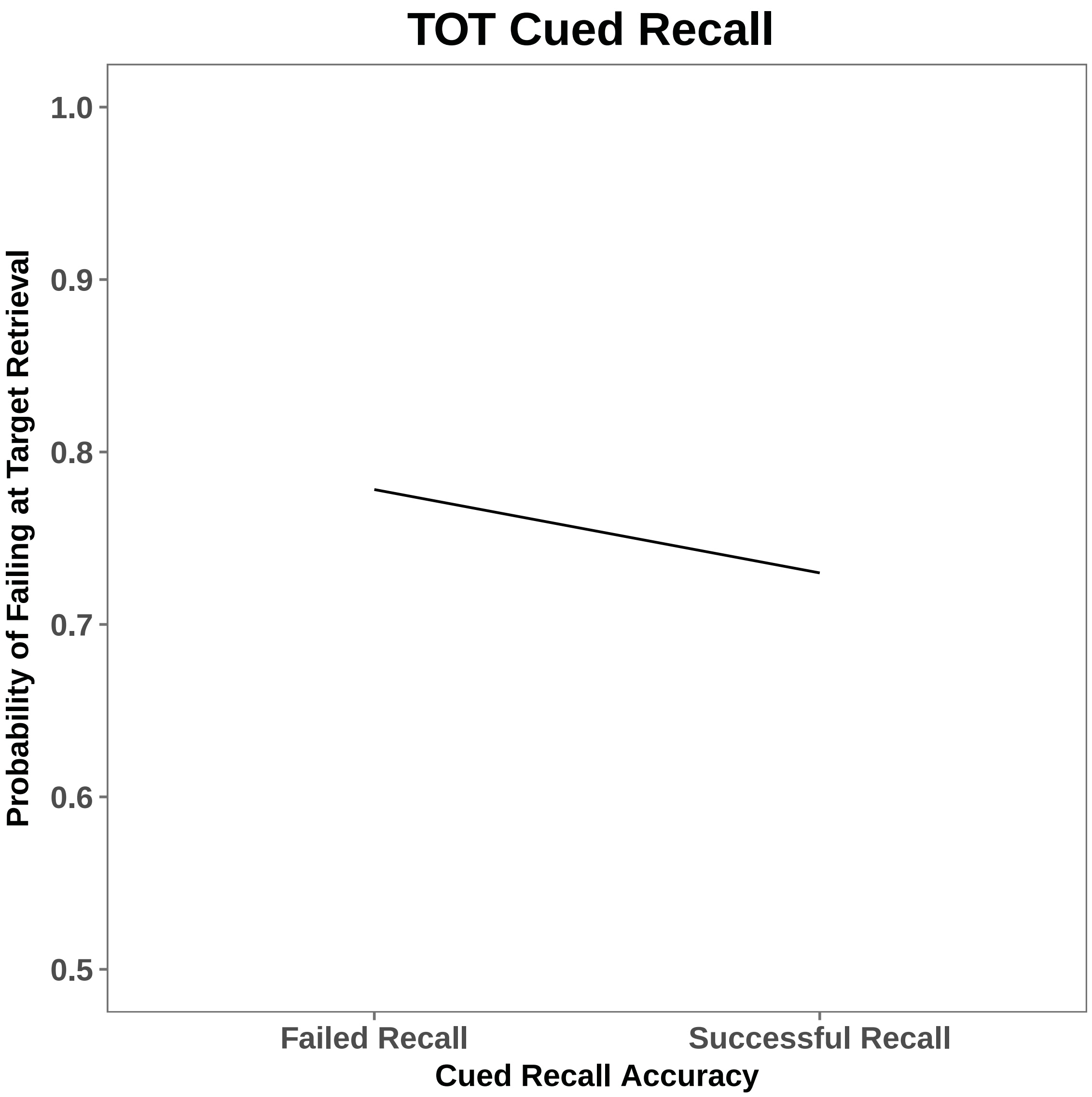
```
> library(lme4)
> TOTcuedrecall$TargetAccuracy = as.factor(TOTcuedrecall$TargetAccuracy)
> TOTcuedrecall$CuedRecallAcc = as.factor(TOTcuedrecall$CuedRecallAcc)
> TOTcuedrecall$FailedRetrieval = ifelse(TOTcuedrecall$TargetAccuracy == 1,0,1)
> totcuedrecall_hlm = glmer(data = TOTcuedrecall, FailedRetrieval ~ CuedRecallAcc +
+                          (1|Subject), family = "binomial")
> summary(totcuedrecall_hlm)
```

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

     AIC      BIC   logLik deviance df.resid
  1380.3   1395.7   -687.2   1374.3     1245

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.0862 -0.8156  0.4808  0.5790  1.2260

Random effects:
 Groups  Name        Variance Std.Dev.
 Subject (Intercept) 0.365    0.6042
Number of obs: 1248, groups:  Subject, 26

Fixed effects:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)      1.2553     0.1562   8.036 9.25e-16 ***
CuedRecallAcc1  -0.2613     0.1543  -1.694   0.0903 .
```

```
---
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)
CudRcllAcc1 -0.476
```

6.0.1 Plot

```
> fixed.frame ←
+   data.frame(expand.grid( CuedRecallAcc = c("0","1"))) %>%
+   mutate(pred = predict(totcuedrecall_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(CuedRecallAccuracy = factor(CuedRecallAcc,
+     levels = unique(CuedRecallAcc),
+                     labels = c("Failed Recall", "Successful Recall")))%>%
+ ggplot(aes(x = CuedRecallAccuracy, y = prob))+
+   geom_line(group = 1)+
+   ylim(.5,1)+
+ # geom_bar(stat = "identity", position = "dodge",
+  #          width = 0.7, color = "black")+
+  theme_few()+
+   xlab("Cued Recall Accuracy") + ylab("Probability of Failing at Target Retrieval") +
+   ggtitle("TOT Cued Recall ")  +
+    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(TOTcuedrecall) = c("Subject", "Session", "Procedure", "Trial",
+                              "Prime", "PrimeDefResp",
+                              "PrimeDefRT", "Cue","PrimeLength",
+                              "PrimeResp", "PrimeRespRT", "Stimuli1",
+                              "StimuliNo",
```

```
+                              "Target", "TargetDefResp", "TargetRT",
+                               "State", "StateRT", "TargetResp", "TargetRespRT",
+                               "CuedRecallAcc", "Accuracy",
+                               "RTrecognisePrime", "RTrecogniseTarget",
+                              "FailedRetrieval")
> ## aggregate per subject all IVs and DVs
> meanRT = group_by(TOTcuedrecall, Subject) %>%
+   summarise_at(vars(PrimeDefRT, TargetRT, RTrecognisePrime, RTrecogniseTarget), mean)
> colnames(meanRT) = c("Subject","MeanPrimeRT", "MeanTargetRT",
+                      "MeanRTrecogPrime", "MeanRTrecogTarget")
> sdRT = group_by(TOTcuedrecall, 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
> TOTcuedrecall_z = merge(TOTcuedrecall, RT_agg, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> TOTcuedrecall_z = TOTcuedrecall_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(TOTcuedrecall_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(TOTcuedrecall_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(TOTcuedrecall_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(TOTcuedrecall_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(TOTcuedrecall_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(TOTcuedrecall_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(TOTcuedrecall_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
> TOTcuedrecall_z_trimmed = subset(TOTcuedrecall_z,
+                             TOTcuedrecall_z$zPrimeRecogRT < 3 &
+                               TOTcuedrecall_z$zPrimeRecogRT > -3)
```

10 Repeating z-scoring

```
> ## aggregate per subject all IVs and DVs
> meanRT = group_by(TOTcuedrecall_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(TOTcuedrecall_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
> TOTcuedrecall_final_z = merge(TOTcuedrecall_z_trimmed,
+                              RT_agg, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> TOTcuedrecall_final_z = TOTcuedrecall_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(TOTcuedrecall_final_z, Subject) %>%
+   summarise_at(vars(zTargetRT_trim,zPrimeRecogRT_trim, zTargetRecogRT_trim), mean)
>
```

11 Final RT distributions

```
> # ggplot(TOTcuedrecall_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(TOTcuedrecall_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(TOTcuedrecall_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")
>
```

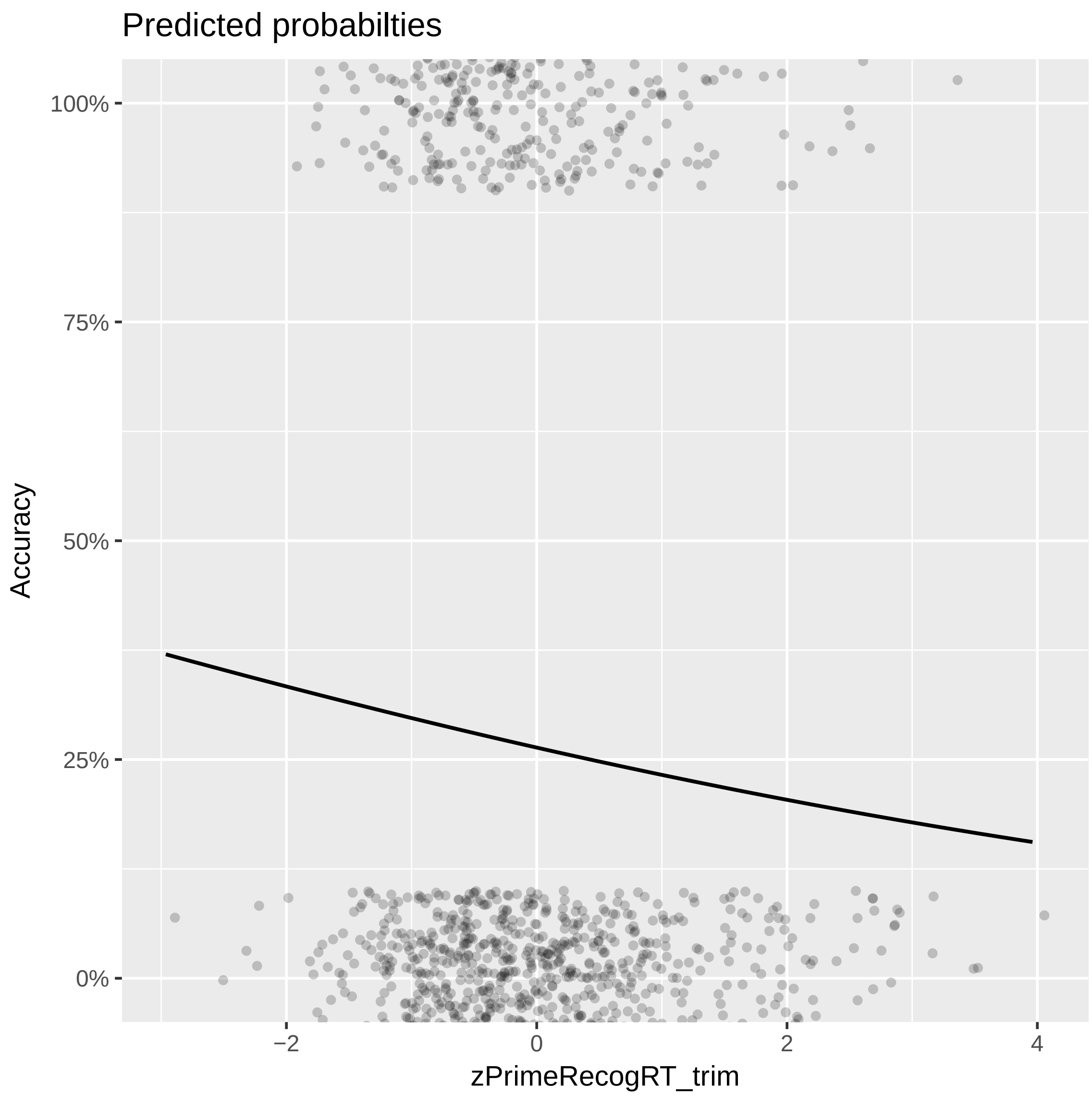
12 Linear Models

```
> # Mean RT to retrieve Target as a function of Prime Condition
>
> # Effect of RT prime on Accuracy
> TOTcuedrecall_final_z = TOTcuedrecall_final_z %>%
+                          filter(!Subject %in% c(1))
> library(lme4)
> RTprime_acc_model = glmer(data = TOTcuedrecall_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: TOTcuedrecall_final_z

     AIC      BIC   logLik deviance df.resid
  1090.8   1110.9   -541.4   1082.8     1131

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.4032 -0.4827 -0.2887  0.4082  4.4228

Random effects:
 Groups  Name        Variance Std.Dev.
 Target  (Intercept) 1.8795   1.3709
 Subject (Intercept) 0.7802   0.8833
Number of obs: 1135, groups:  Target, 48; Subject, 24

Fixed effects:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1.47272    0.28418  -5.182 2.19e-07 ***
zPrimeRecogRT_trim -0.12043    0.08595  -1.401    0.161
```

```
---
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 0.015
```

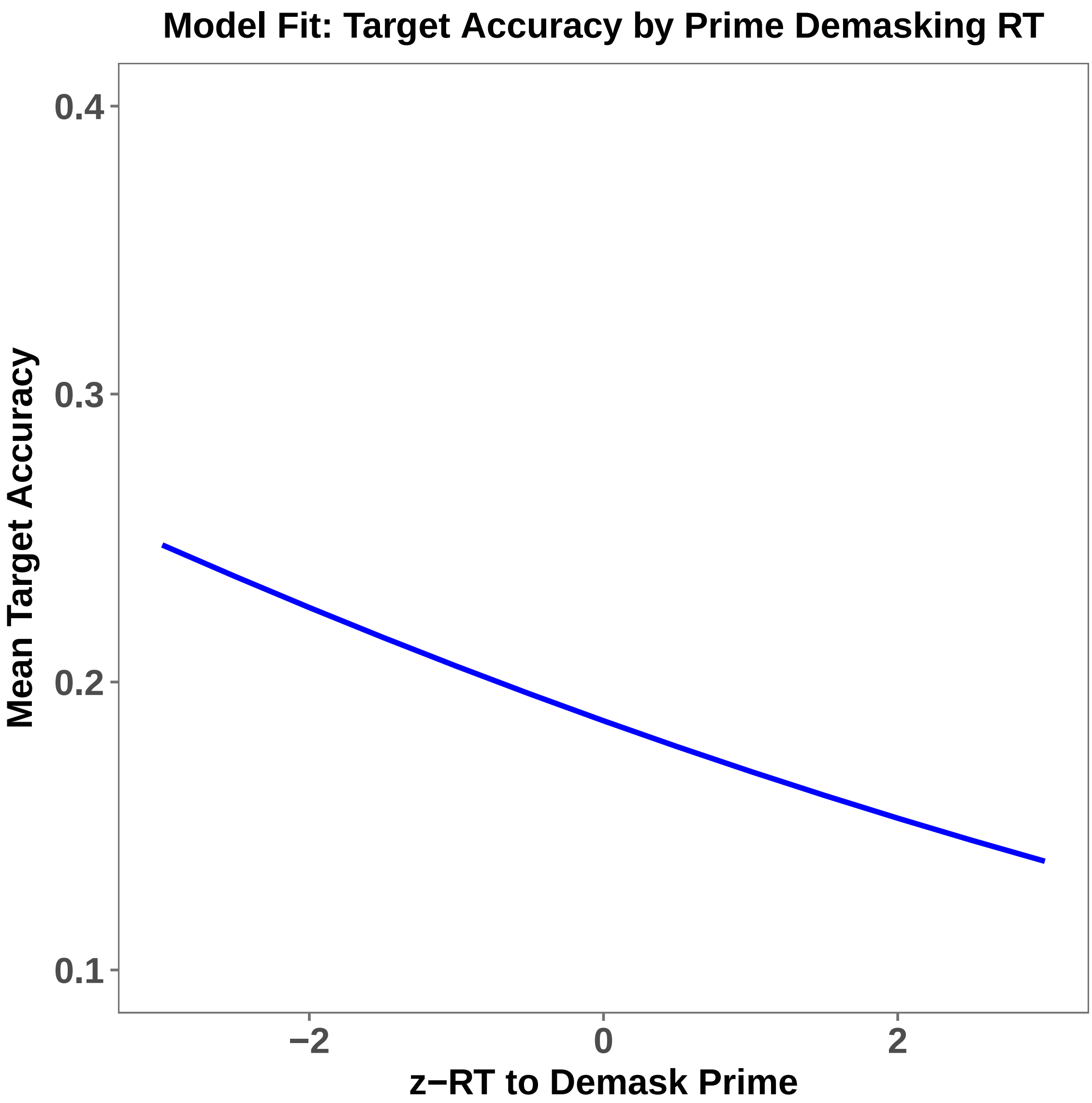
```
> library(sjPlot)
> sjp.glmer(RTprime_acc_model, type = "pred", vars = "zPrimeRecogRT_trim")
>
```



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

```
> TOTcuedrecall_final_z$Accuracy = as.numeric(as.character(TOTcuedrecall_final_z$Accurac
> TOTcuedrecall_final_z1 = TOTcuedrecall_final_z %>% filter(Subject != "6")
> mainplot = TOTcuedrecall_final_z1 %>%
+  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(-0.2,1)+
+     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")
>
```