

TOT Semantic Analysis

Abhilasha Kumar

May 1, 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.

```
> TOTSemantic = read.csv("TOTSemantic_AllSubjects.csv",
+                         header = TRUE, sep = ",")
> head(TOTSemantic)
```

	Subject	Session	Trial	Prime.Trial.	PrimeDef.RESP.Trial.	PrimeDef.RT.Trial.
1	1	1	46	r e s i g n	resign{SPACE}	9526
2	1	1	52	n a t i v e	native{SPACE}	5628
3	1	1	28	r e f u s e	begrudge{SPACE}	8855
4	1	1	68	c o n d o n e	allow{SPACE}	5587
5	1	1	9	B r o n t e	{SPACE}	3286
6	1	1	27	m i s p l a c e	{SPACE}	6449
	PrimeResponse.RESP.Trial.		PrimeResponse.RT.Trial.		Target.Trial.	
1	resign{SPACE}		2363		a b d i c a t e	
2	native{SPACE}		1531		a b o r i g i n e	
3	refuse{SPACE}		3260		a b s t a i n	
4	condone{SPACE}		1907		a d v o c a t e	
5	bronte{SPACE}		3381		A l c o t t	
6	misplace{SPACE}		1744		a n a c h r o n i s m	
	TargetDefinition.RESP.Trial.		TargetDefinition.RT.Trial.			
1	stepdown{SPACE}		5399			
2	{SPACE}		4314			
3	{SPACE}		4875			
4	{SPACE}		4554			
5	{SPACE}		1871			
6	{SPACE}		6598			
	TargetQuestion.RESP.Trial.		TargetQuestion.RT.Trial.			
1	3		1292			
2	2		924			
3	2		1014			
4	2		947			
5	2		1112			
6	2		1169			

	TargetResponse.RESP.Trial.	TargetResponse.RT.Trial.	PrimeFirstResp_ACC
1	{SPACE}	1032	1
2	aborigine{SPACE}	2253	1
3	abstain{SPACE}	4050	0
4	advocate{SPACE}	2307	0
5	alcott{SPACE}	1689	0
6	anachronism{SPACE}	3560	0

	TargetFirstResp_ACC	RTrecognisePrime	RTrecogniseTarget
1	0	3937	3770
2	0	2986	1237
3	0	2666	3943
4	0	3721	2233
5	0	5180	3923
6	0	3406	6769

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(TOTSemantic) %>%
+   summarise_at(vars(PrimeFirstResp_ACC, TargetFirstResp_ACC), mean)
> cued_acc = group_by(TOTSemantic, Subject, PrimeFirstResp_ACC) %>%
+   summarise(recalltrials = n())
> conditional_acc = group_by(TOTSemantic, 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 27 3.848e-30 1.425e-31

Error: Subject:PrimeFirstResp_ACC
      Df      Sum Sq    Mean Sq F value Pr(>F)
PrimeFirstResp_ACC 1 1.40e-31 1.399e-31 0.332 0.57
Residuals          27 1.14e-29 4.221e-31

Error: Subject:TargetFirstResp_ACC
      Df Sum Sq Mean Sq F value Pr(>F)
TargetFirstResp_ACC 1 4.442 4.442 83.25 9.76e-10 ***
Residuals          27 1.441 0.053
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: Subject:PrimeFirstResp_ACC:TargetFirstResp_ACC
      Df Sum Sq Mean Sq F value Pr(>F)
PrimeFirstResp_ACC:TargetFirstResp_ACC 1 0.8501 0.8501 77.2 2.11e-09 ***
Residuals          27 0.2973 0.0110
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

4 Conditional Figure

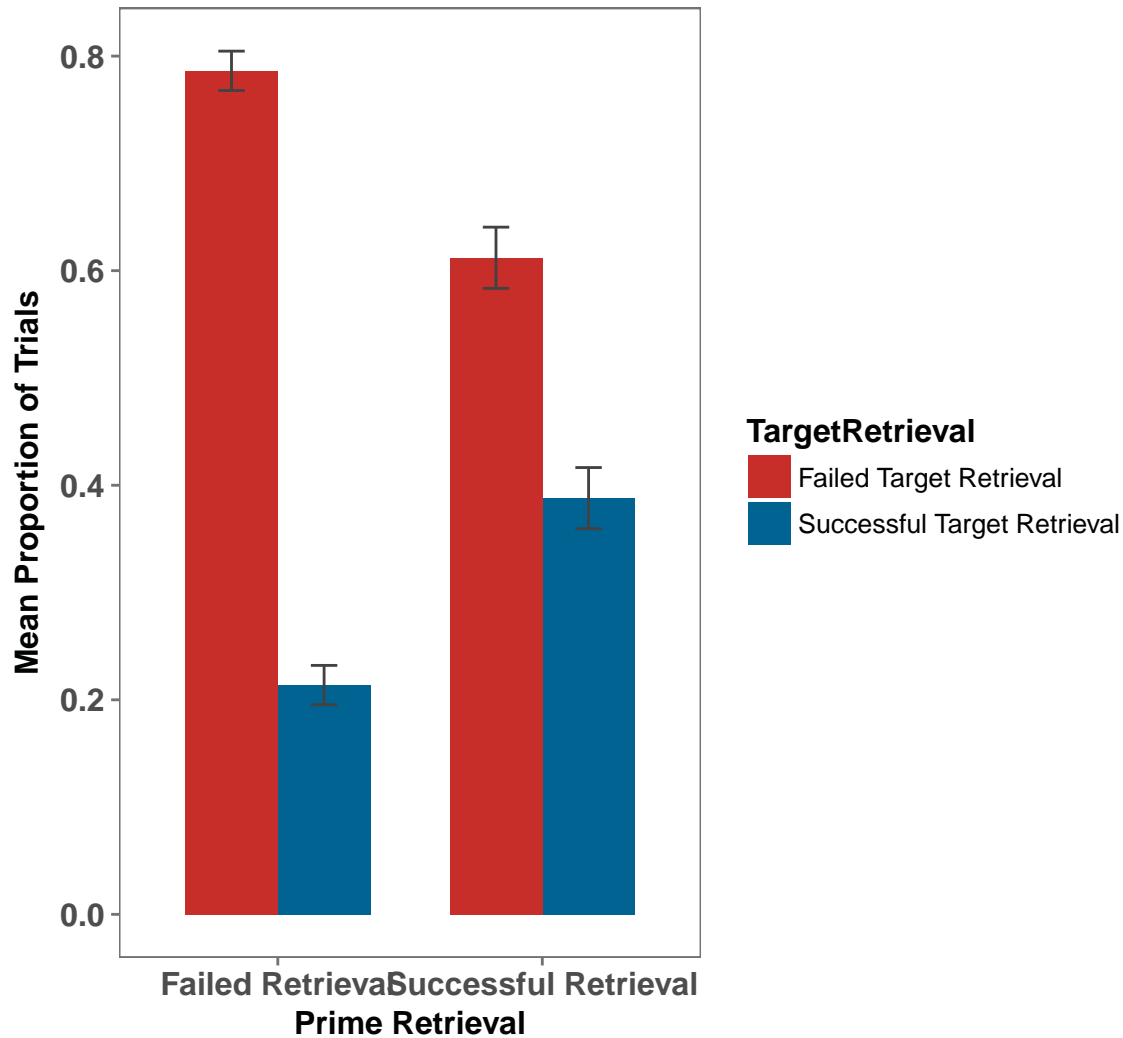
```
> cond_figure = Rmisc::summarySE(merge_acc,
+                               measurevar = "prop",
+                               groupvars = c("PrimeFirstResp_ACC",
+                                              "TargetFirstResp_ACC"))
> library(ggplot2)
> library(ggthemes)
> condfigure_plot = cond_figure %>% 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_ws() +
+   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

```

Target Retrieval Accuracy as a function of Prime Retrieval Accuracy



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 = 8.7862, df = 27, p-value = 2.112e-09
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.2671025 0.4298641
sample estimates:
mean of the differences
      0.3484833
```

6 HLM Model

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

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

      AIC      BIC   logLik deviance df.resid
 2279.1   2296.0  -1136.6   2273.1     2013

Scaled residuals:
    Min       1Q   Median       3Q      Max
-3.2017 -0.9521  0.4473  0.6431  1.1773
```

```

Random effects:
  Groups   Name      Variance Std.Dev.
  Subject (Intercept) 0.2824   0.5314
Number of obs: 2016, groups: Subject, 28

Fixed effects:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)      1.3978     0.1260  11.096   <2e-16 ***
PrimeFirstResp_ACC1 -0.9420     0.1047  -8.996   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
      (Intr)
PrmFrR_ACC1 -0.429

```

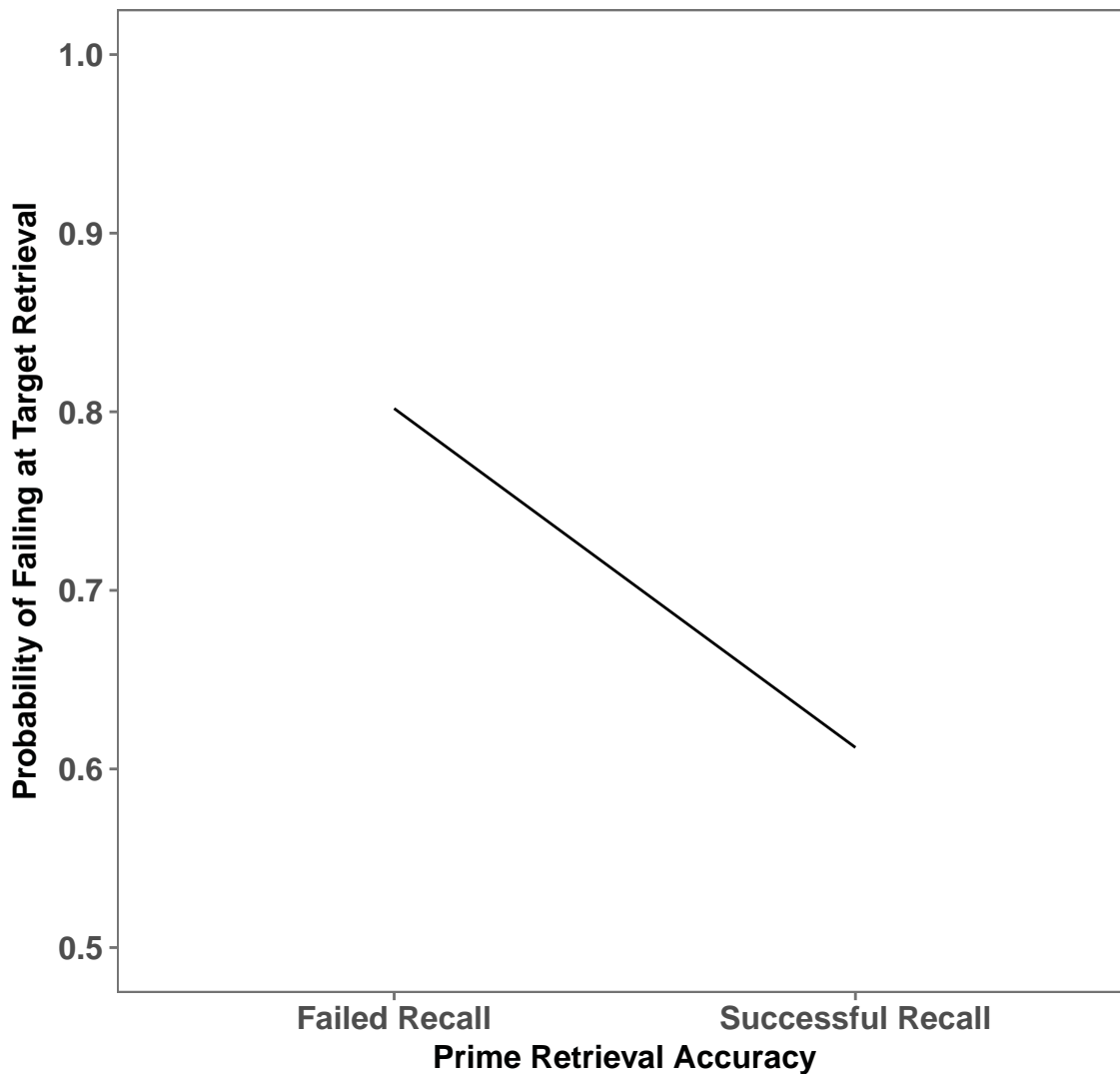
6.0.1 Plot

```

> fixed.frame <-
+   data.frame(expand.grid( PrimeFirstResp_ACC = c("0","1"))) %>%
+   mutate(pred = predict(TOTSemantic_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 Semantic ") +
+     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)))
>

```

TOT Semantic



7 z-scoring RTs

RT prime and Target

```
> library(dplyr)
> colnames(TOTSemantic) = c( "Subject", "Session", "Trial", "Prime", "PrimeDefResp",
+                             "PrimeDefRT", "PrimeResp",
+                             "PrimeRespRT",
+                             "Target", "TargetDefResp", "TargetRT",
+                             "State", "StateRT", "TargetResp", "TargetRespRT",
```



```

+           "PrimeAcc", "Accuracy",
+           "RTrecognisePrime", "RTrecogniseTarget",
+           "FailedRetrieval")
> TOTSemantic$PrimeDefRT = as.numeric(as.character(TOTSemantic$PrimeDefRT))
> ## aggregate per subject all IVs and DVs
> meanRT = group_by(TOTSemantic, Subject) %>%
+   summarise_at(vars(PrimeDefRT, TargetRT, RTrecognisePrime, RTrecogniseTarget), mean)
> colnames(meanRT) = c("Subject", "MeanPrimeRT", "MeanTargetRT",
+   "MeanRTrecogPrime", "MeanRTrecogTarget")
> sdRT = group_by(TOTSemantic, 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
> TOTSemantic_z = merge(TOTSemantic, RT_agg, by = "Subject", all.x = T)
> ## person and grand-mean centered scores using original and aggregate
> library(dplyr)
> TOTSemantic_z = TOTSemantic_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(TOTSemantic_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(TOTSemantic_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(TOTSemantic_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(TOTSemantic_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(TOTSemantic_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(TOTSemantic_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(TOTSemantic_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
> TOTSemantic_z_trimmed = subset(TOTSemantic_z,
+                               TOTSemantic_z$zPrimeRecogRT < 3 &
+                               TOTSemantic_z$zPrimeRecogRT > -3)

```

10 Repeating z-scoring

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

11 Final RT distributions

```
> # ggplot(TOTSemantic_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(TOTSemantic_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(TOTSemantic_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
> TOTSemantic_final_z = TOTSemantic_final_z
> library(lme4)
> RTprime_acc_model = glmer(data = TOTSemantic_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: TOTSemantic_final_z

```

AIC	BIC	logLik	deviance	df.resid
2004.1	2026.5	-998.1	1996.1	1993

Scaled residuals:

Min	1Q	Median	3Q	Max
-3.7006	-0.5298	-0.3088	0.5151	4.7159

Random effects:

Groups	Name	Variance	Std.Dev.
Target	(Intercept)	1.7163	1.3101
Subject	(Intercept)	0.6696	0.8183

Number of obs: 1997, groups: Target, 72; Subject, 28

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.26395	0.22883	-5.523	3.32e-08 ***
zPrimeRecogRT_trim	-0.24264	0.06994	-3.469	0.000522 ***

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

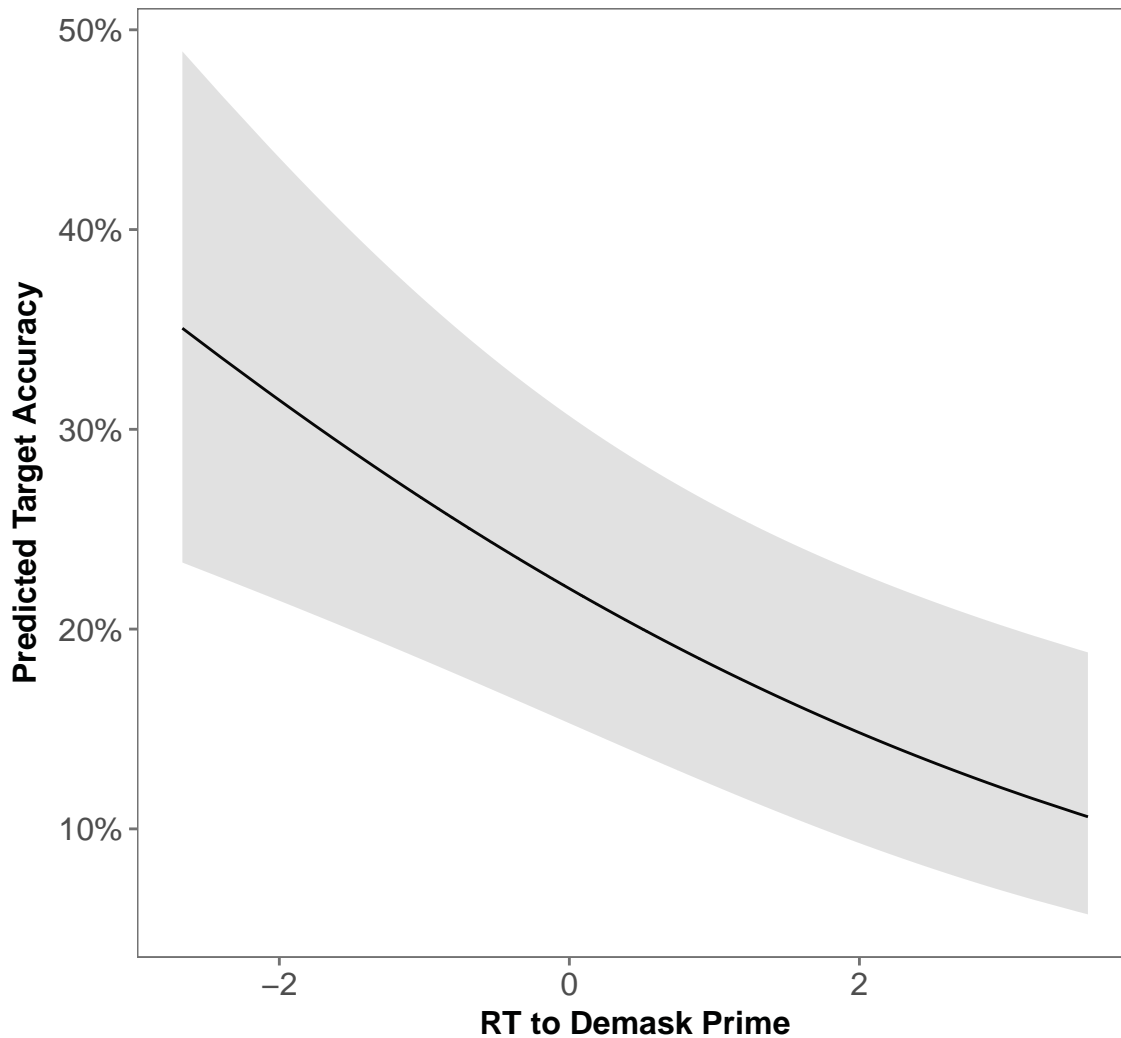
```
Correlation of Fixed Effects:
```

```
(Intr)
```

```
zPrmRcgRT_t 0.019
```

```
> 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)))
>
```

Target Accuracy ~ Demasking RT

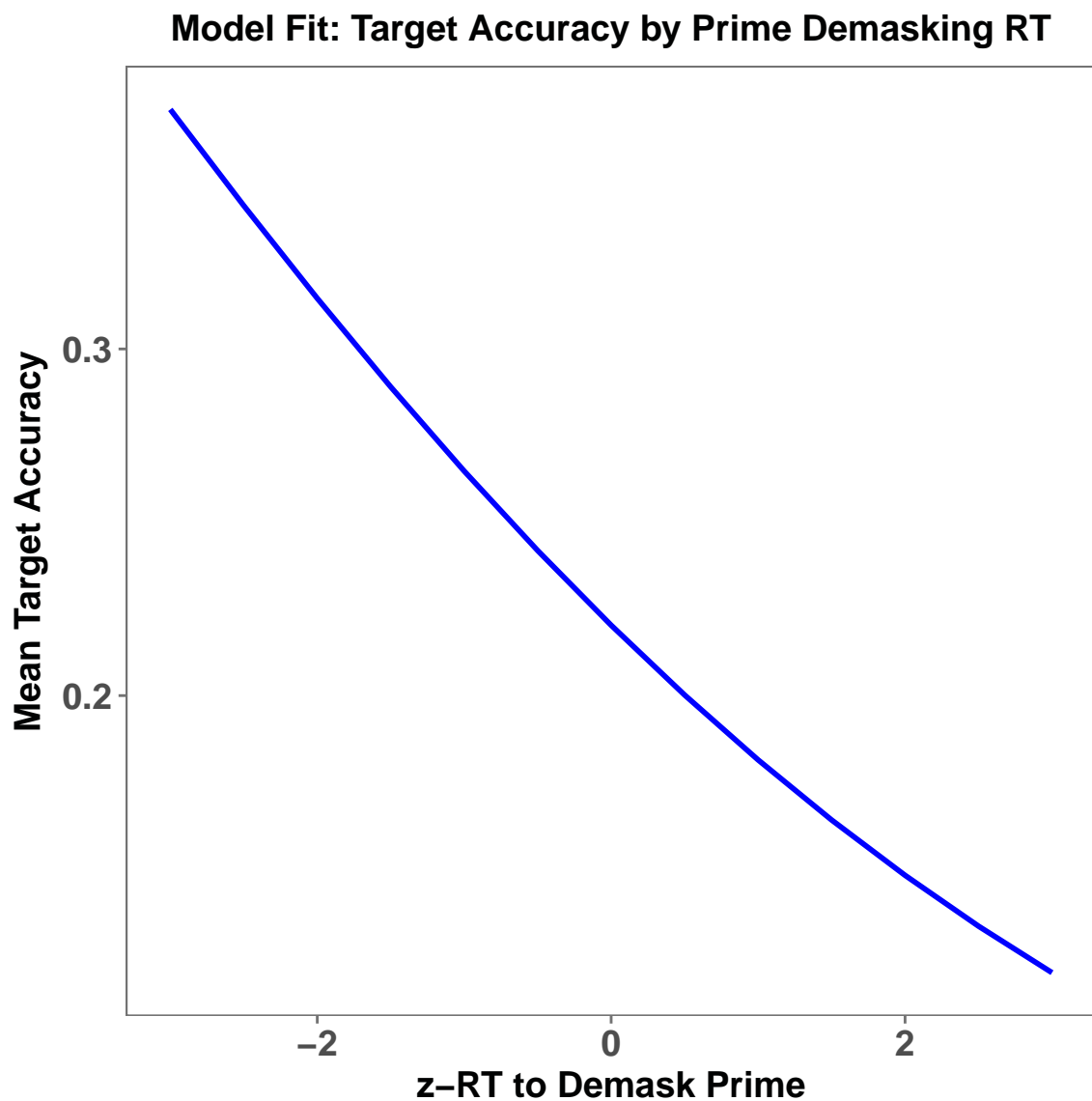


13 Plotting Model Fits

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") +
+   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)
> TOTSemantic_final_z$Accuracy = as.numeric(as.character(TOTSemantic_final_z$Accuracy))
> mainplot = TOTSemantic_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

