TOT Cued Recall Analysis

Abhilasha Kumar

February 13, 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.

```
> 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)
```

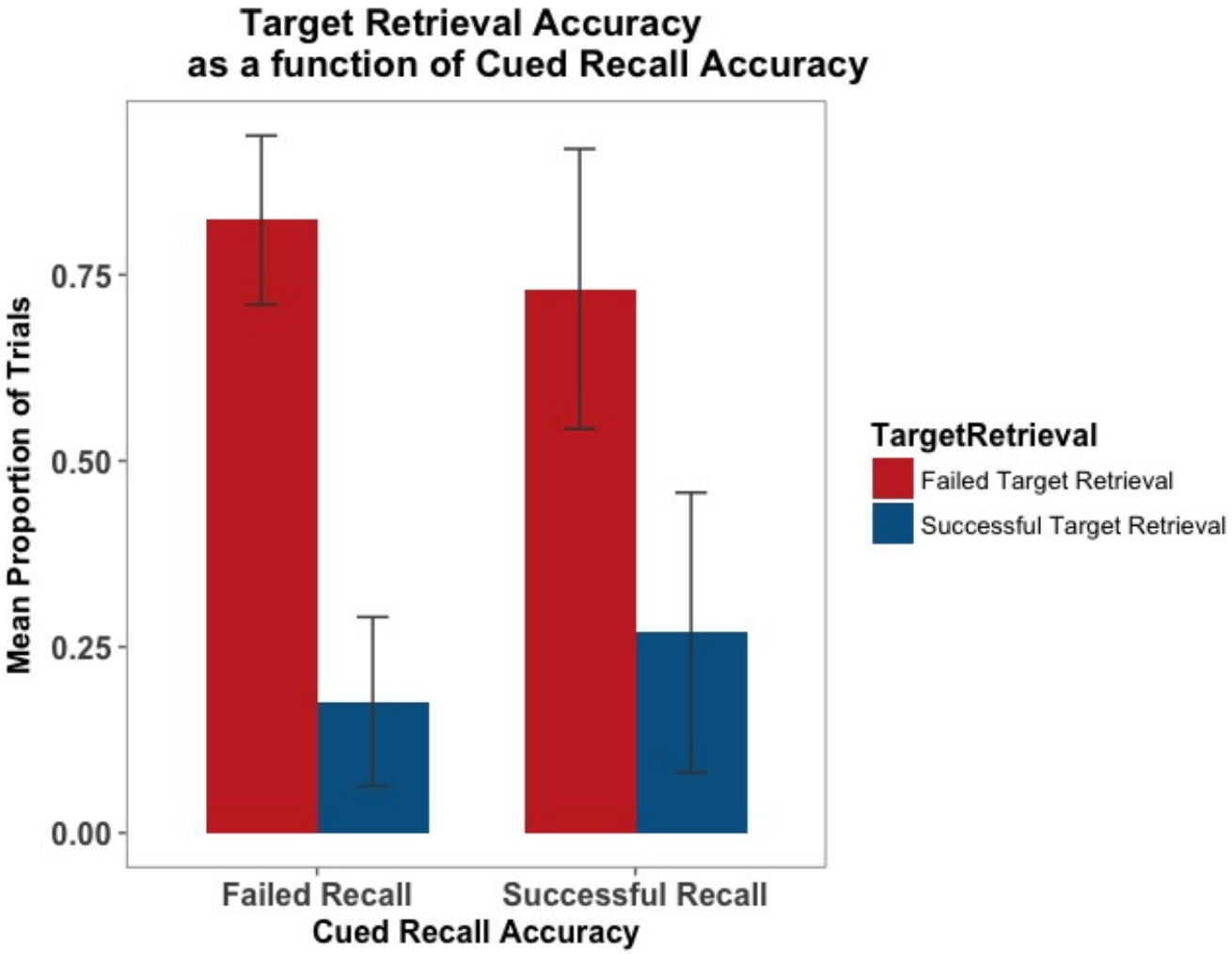
```
Error: Subject
          Df    Sum Sq   Mean Sq F value Pr(>F)
Residuals  3 1.541e-32 5.136e-33

Error: Subject:CuedRecallAcc
              Df    Sum Sq   Mean Sq F value Pr(>F)
CuedRecallAcc  1 1.930e-34 1.930e-34    0.03  0.873
Residuals      3 1.907e-32 6.356e-33

Error: Subject:TargetAccuracy
               Df Sum Sq Mean Sq F value Pr(>F)
TargetAccuracy  1 1.2295  1.2295   63.77 0.0041 **
Residuals       3 0.0578  0.0193
---
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 0.03423 0.03423   1.813  0.271
Residuals                     3 0.05665 0.01888
```

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

```
> library(Rmisc)
> cond_figure = 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")),
+                     TargetRetrieval = factor(TargetAccuracy,
+                           levels = unique(TargetAccuracy),
+                        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 - 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)))
```



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)
```

```
        Paired t-test

data:  failed_wide$diff and successful_wide$diff
t = 1.3464, df = 3, p-value = 0.2709
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.2523130  0.6223534
sample estimates:
mean of the differences
              0.1850202
```