Primed Memory Retrieval: Conditional Accuracy

Abhilasha Kumar

November 16, 2017

Reading and Formatting Data

```
memory500 = read.csv("Compiled_Final500ms_CSV.csv", header = TRUE, sep = ",")
memory32 = read.csv("Compiled_Final32ms_CSV.csv", header = TRUE, sep = ",")
memory125 = read.csv("Compiled_Final125ms_CSV.csv", header = TRUE, sep = ",")
memory48 = read.csv("Compiled_Final48ms_CSV.csv", header = TRUE, sep = ",")
```

Means Per Prime Condition

32 ms

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
prime_acc_32 = group_by(memory32, PrimeCondition) %>%
  summarise_at(vars(ImmediateAccuracy, DelayedAccuracy,RTImmediate, RTDelayed), mean)

prime_subject_acc_32 = group_by(memory32, Subject, PrimeCondition) %>%
  summarise_at(vars(ImmediateAccuracy, DelayedAccuracy,RTImmediate, RTDelayed), mean)
```

48 ms

```
library(dplyr)
prime_acc_48 = group_by(memory48, PrimeCondition) %>%
  summarise_at(vars(ImmediateAccuracy, DelayedAccuracy,RTImmediate, RTDelayed), mean)

prime_subject_acc_48 = group_by(memory48, Subject, PrimeCondition) %>%
  summarise_at(vars(ImmediateAccuracy, DelayedAccuracy,RTImmediate, RTDelayed), mean)
```

500 ms

```
library(dplyr)
prime_acc_500 = group_by(memory500, PrimeCondition) %>%
  summarise_at(vars(ImmediateAccuracy, DelayedAccuracy,RTImmediate, RTDelayed), mean)

prime_subject_acc_500 = group_by(memory500, Subject, PrimeCondition) %>%
  summarise_at(vars(ImmediateAccuracy, DelayedAccuracy,RTImmediate, RTDelayed), mean)
```

ANOVAS

32 ms

ANOVA for Immediate Recall

```
prime_subject_acc_32$PrimeCondition = as.factor(prime_subject_acc_32$PrimeCondition)
prime_subject_acc_32$Subject = as.factor(prime_subject_acc_32$Subject)

immediate_aov_32 = aov(data = prime_subject_acc_32,
                       ImmediateAccuracy ~ PrimeCondition +
                               Error(Subject/PrimeCondition))
summary(immediate_aov_32)
```

```
##
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  9  2.019  0.2243
##
## Error: Subject:PrimeCondition
##                Df Sum Sq Mean Sq F value   Pr(>F)
## PrimeCondition  4 0.2536 0.06340   6.055 0.000783 ***
## Residuals      36 0.3769 0.01047
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
options(contrasts = c('contr.sum', 'contr.poly'))
library(lsmeans)
```

```
## Loading required package: estimability
```

```
library(multcomp)
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: survival
```

```
## Loading required package: TH.data
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
```

```
##     select
```

```
##
## Attaching package: 'TH.data'
```

```
## The following object is masked from 'package:MASS':
##
##     geyser
```

```
imm_lsm = lsmeans::lsmeans(immediate_aov_32, c("PrimeCondition"))
```

```
## Warning in lsm.basis.aovlist(object, trms, xlev, grid, ...): Some predictors are correlated with the
## May help to re-fit with different contrasts, e.g. 'contr.sum'
```

```
prime_effect = cld(imm_lsm, alpha = 0.05,
                adjust = "tukey", details = TRUE)
library(knitr)
kable(subset(prime_effect$comparisons,prime_effect$comparisons$p.value < 0.05 ))
```

	contrast	estimate	SE	df	t.ratio	p.value
4	Neutral - Semantic	0.1333333	0.0457617	36	2.913643	0.0452347
7	Identical - Semantic	0.2083333	0.0457617	36	4.552567	0.0005311
8	Identical - Phonological	0.1583333	0.0457617	36	3.459951	0.0115319

ANOVA for Delayed Recall

```
delayed_aov_32 = aov(data = prime_subject_acc_32,
                     DelayedAccuracy ~ PrimeCondition +
                                Error(Subject/PrimeCondition))
summary(delayed_aov_32)
```

```
##
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  9  1.653  0.1836
##
## Error: Subject:PrimeCondition
##                Df Sum Sq Mean Sq F value  Pr(>F)
## PrimeCondition  4 0.1694 0.04236   4.427 0.00517 **
## Residuals      36 0.3444 0.00957
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

48 ms

ANOVA for Immediate Recall

```
prime_subject_acc_48$PrimeCondition = as.factor(prime_subject_acc_48$PrimeCondition)
prime_subject_acc_48$Subject = as.factor(prime_subject_acc_48$Subject)

immediate_aov_48 = aov(data = prime_subject_acc_48,
                       ImmediateAccuracy ~ PrimeCondition +
```

```
                                  Error(Subject/PrimeCondition))
summary(immediate_aov_48)
```

```
##
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 17  4.164  0.2449
##
## Error: Subject:PrimeCondition
##                 Df Sum Sq Mean Sq F value   Pr(>F)
## PrimeCondition   4 0.8179 0.20448   9.243 5.03e-06 ***
## Residuals       68 1.5043 0.02212
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
options(contrasts = c('contr.sum', 'contr.poly'))
library(lsmeans)
library(multcomp)
imm_lsm = lsmeans::lsmeans(immediate_aov_48, c("PrimeCondition"))
prime_effect = cld(imm_lsm, alpha = 0.05,
                adjust = "tukey", details = TRUE)
library(knitr)
kable(subset(prime_effect$comparisons,prime_effect$comparisons$p.value < 0.05 ))
```

	contrast	estimate	SE	df	t.ratio	p.value
7	Identical - UnSemantic	0.2685185	0.0495786	68	5.416013	0.0000084
8	Identical - Neutral	0.2407407	0.0495786	68	4.855736	0.0000706
9	Identical - Phonological	0.2129630	0.0495786	68	4.295459	0.0005303
10	Identical - Semantic	0.2037037	0.0495786	68	4.108700	0.0010071

ANOVA for Delayed Recall

```
delayed_aov_48 = aov(data = prime_subject_acc_48,
                     DelayedAccuracy ~ PrimeCondition +
                                  Error(Subject/PrimeCondition))
summary(delayed_aov_48)
```

```
##
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 17  6.004  0.3532
##
## Error: Subject:PrimeCondition
##                 Df Sum Sq  Mean Sq F value Pr(>F)
## PrimeCondition   4 0.0350 0.008758   0.691  0.601
## Residuals       68 0.8622 0.012679
```

500 ms

ANOVA for Immediate Recall

```
prime_subject_acc_500$PrimeCondition = as.factor(prime_subject_acc_500$PrimeCondition)
prime_subject_acc_500$Subject = as.factor(prime_subject_acc_500$Subject)
immediate_aov_500 = aov(data = prime_subject_acc_500,
                        ImmediateAccuracy ~ PrimeCondition +
                                Error(Subject/PrimeCondition))
summary(immediate_aov_500)
```

```
##
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 19  6.901  0.3632
##
## Error: Subject:PrimeCondition
##                Df Sum Sq Mean Sq F value   Pr(>F)
## PrimeCondition  4 1.6199  0.4050   31.31 2.09e-15 ***
## Residuals      76 0.9829  0.0129
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
options(contrasts = c('contr.sum', 'contr.poly'))
library(lsmeans)
library(multcomp)
imm_lsm = lsmeans::lsmeans(immediate_aov_500, c("PrimeCondition"))
prime_effect = cld(imm_lsm, alpha = 0.05,
                adjust = "tukey", details = TRUE)
library(knitr)
kable(subset(prime_effect$comparisons,prime_effect$comparisons$p.value < 0.05 ))
```

	contrast	estimate	SE	df	t.ratio	p.value
4	Semantic - UnSemantic	0.1791667	0.0359626	76	4.982022	0.0000372
5	Semantic - Neutral	0.1750000	0.0359626	76	4.866161	0.0000581
6	Semantic - Phonological	0.1250000	0.0359626	76	3.475830	0.0073354
7	Identical - UnSemantic	0.3333333	0.0359626	76	9.268879	0.0000000
8	Identical - Neutral	0.3291667	0.0359626	76	9.153018	0.0000000
9	Identical - Phonological	0.2791667	0.0359626	76	7.762686	0.0000000
10	Identical - Semantic	0.1541667	0.0359626	76	4.286856	0.0004933

ANOVA for Delayed Recall

```
delayed_aov_500 = aov(data = prime_subject_acc_500,
                      DelayedAccuracy ~ PrimeCondition +
                                Error(Subject/PrimeCondition))
summary(delayed_aov_500)
```

```
##
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 19   9.19  0.4837
```

```
##
## Error: Subject:PrimeCondition
##                Df Sum Sq Mean Sq F value  Pr(>F)
## PrimeCondition  4 0.2753 0.06882   8.803 6.8e-06 ***
## Residuals      76 0.5942 0.00782
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Conditional Accuracy

32 ms

Organizing Data

```
library(dplyr)
## overall delayed acc means split by imm acc (0/1)
cond_32 = group_by(memory32, ImmediateAccuracy, PrimeCondition) %>%
  summarise_at(vars(DelayedAccuracy), mean)
##per subject
cond_sub_32 = group_by(memory32, Subject, ImmediateAccuracy, PrimeCondition) %>%
  summarise_at(vars(DelayedAccuracy), mean)
##taking only correct immediate responses from this:
cond_sub_correct_32 = subset(cond_sub_32, cond_sub_32$ImmediateAccuracy == "1")

## Merging immediate and delayed into one file:

## now we have conditional delayed accuracies in cond_sub_correct_32
## and we have immediate accuracies in prime_subject_acc_32
## now we must combine the two files.

colnames(cond_sub_correct_32) = c("Subject", "IA", "PrimeCondition", "Delayed")
colnames(prime_subject_acc_32) = c("Subject", "PrimeCondition", "Immediate", "DA",
                                   "RTImm", "RTdel")

final_combined_32 = merge(cond_sub_correct_32, prime_subject_acc_32,
                      by = c("Subject", "PrimeCondition"))

# now we have immediate and delayed accuracy in wide format, i.e. side to side.
# We need to convert to long format

#first we remove uneccesary columns, i.e. IA and DA

final_combined_32 = final_combined_32[, -c(3,6,7,8)]

# converting to long

library(tidyr)
final_long_32 <- final_combined_32 %>%
  gather(Delay, Accuracy, Immediate, Delayed)
```

```r
final_long_32 <- final_long_32[order(final_long_32$Subject),]
final_long_32$Delay = as.factor(as.character(final_long_32$Delay))
final_long_32$Subject = as.factor(as.character(final_long_32$Subject))
final_long_32$Accuracy = as.numeric(as.character(final_long_32$Accuracy))
```

Conditional ANOVA

```r
final_aov_32 = aov(data = final_long_32, Accuracy ~ Delay*PrimeCondition +
                   Error(Subject/(Delay*PrimeCondition)))
summary(final_aov_32)
```

```
##
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  9  1.382  0.1535
##
## Error: Subject:Delay
##           Df Sum Sq Mean Sq F value   Pr(>F)
## Delay      1 3.1345  3.1345   34.41 0.000239 ***
## Residuals  9 0.8199  0.0911
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Subject:PrimeCondition
##                Df Sum Sq Mean Sq F value  Pr(>F)
## PrimeCondition  4 0.2328 0.05819    4.22 0.00666 **
## Residuals      36 0.4964 0.01379
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Subject:Delay:PrimeCondition
##                       Df Sum Sq Mean Sq F value Pr(>F)
## Delay:PrimeCondition   4 0.1231 0.03078   1.291  0.292
## Residuals             36 0.8581 0.02384
```

```r
library(multcomp)
library(knitr)
options(contrasts = c('contr.sum', 'contr.poly'))

conditional_lsm = lsmeans::lsmeans(final_aov_32, c("Delay", "PrimeCondition"))
prime_effect = cld(conditional_lsm, alpha = 0.05,
                adjust = "tukey", details = TRUE, by = "Delay")
kable(subset(prime_effect$comparisons,prime_effect$comparisons$p.value < 0.05 ))
```

	contrast	Delay	estimate	SE	df	t.ratio	p.value
17	Unrelated - Semantic	Immediate	0.2083333	0.0613411	67.20807	3.396308	0.0098118

48 ms

Organizing Data

```
library(dplyr)
## overall delayed acc means split by imm acc (0/1)
cond_48 = group_by(memory48, ImmediateAccuracy, PrimeCondition) %>%
  summarise_at(vars(DelayedAccuracy), mean)
##per subject
cond_sub_48 = group_by(memory48, Subject, ImmediateAccuracy, PrimeCondition) %>%
  summarise_at(vars(DelayedAccuracy), mean)
##taking only correct immediate responses from this:
cond_sub_correct_48 = subset(cond_sub_48, cond_sub_48$ImmediateAccuracy == "1")

#for 48 ms, S12 has 2 rows missing

cond_sub_correct_48[89,] = c("12",1,"Semantic", 0 )
cond_sub_correct_48[90,] = c("12",1,"UnSemantic", 0 )

##  Merging immediate and delayed into one file:

## now we have conditional delayed accuracies in cond_sub_correct_48
## and we have immediate accuracies in prime_subject_acc_48
## now we must combine the two files.

colnames(cond_sub_correct_48) = c("Subject", "IA", "PrimeCondition", "Delayed")
colnames(prime_subject_acc_48) = c("Subject", "PrimeCondition", "Immediate", "DA",
                                   "RTImm", "RTDel")

final_combined_48 = merge(cond_sub_correct_48, prime_subject_acc_48,
                          by = c("Subject", "PrimeCondition"))

# now we have immediate and delayed accuracy in wide format, i.e. side to side.
# We need to convert to long format

#first we remove uneccesary columns, i.e. IA and DA

final_combined_48 = final_combined_48[, -c(3,6,7,8)]

# converting to long

library(tidyr)
final_long_48 <- final_combined_48 %>%
  gather(Delay, Accuracy, Immediate, Delayed)


final_long_48 <- final_long_48[order(final_long_48$Subject),]
final_long_48$Delay = as.factor(as.character(final_long_48$Delay))
final_long_48$Subject = as.factor(as.character(final_long_48$Subject))
final_long_48$Accuracy = as.numeric(as.character(final_long_48$Accuracy))
```

Conditional ANOVA

```
final_aov_48 = aov(data = final_long_48, Accuracy ~ Delay*PrimeCondition +
                 Error(Subject/(Delay*PrimeCondition)))
summary(final_aov_48)
```

```
##
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 17  6.505  0.3827
##
## Error: Subject:Delay
##           Df Sum Sq Mean Sq F value   Pr(>F)
## Delay      1 2.2764  2.2764    42.4 5.33e-06 ***
## Residuals 17 0.9127  0.0537
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Subject:PrimeCondition
##                 Df Sum Sq Mean Sq F value Pr(>F)
## PrimeCondition   4 0.0569 0.01422   0.704  0.592
## Residuals       68 1.3733 0.02020
##
## Error: Subject:Delay:PrimeCondition
##                       Df Sum Sq Mean Sq F value   Pr(>F)
## Delay:PrimeCondition   4  1.251 0.31270   10.95 6.55e-07 ***
## Residuals             68  1.943 0.02857
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(multcomp)
library(knitr)
options(contrasts = c('contr.sum', 'contr.poly'))

conditional_lsm = lsmeans::lsmeans(final_aov_48, c("Delay", "PrimeCondition"))
prime_effect = cld(conditional_lsm, alpha = 0.05,
                adjust = "tukey", details = TRUE, by = "PrimeCondition")
kable(subset(prime_effect$comparisons,prime_effect$comparisons$p.value < 0.05 ))
```

	contrast	PrimeCondition	estimate	SE	df	t.ratio	p.value
2	Immediate - Delayed	Neutral	0.3456790	0.0610939	78.01866	5.658156	0.0000002
3	Immediate - Delayed	Phonological	0.3040765	0.0610939	78.01866	4.977196	0.0000038
4	Immediate - Delayed	Semantic	0.2299563	0.0610939	78.01866	3.763979	0.0003223
5	Immediate - Delayed	UnSemantic	0.3427690	0.0610939	78.01866	5.610523	0.0000003

500 ms

Organizing Data

```
library(dplyr)
## overall delayed acc means split by imm acc (0/1)
```

```
cond_500 = group_by(memory500, ImmediateAccuracy, PrimeCondition) %>%
  summarise_at(vars(DelayedAccuracy), mean)
##per subject
cond_sub_500 = group_by(memory500, Subject, ImmediateAccuracy, PrimeCondition) %>%
  summarise_at(vars(DelayedAccuracy), mean)
##taking only correct immediate responses from this:
cond_sub_correct_500 = subset(cond_sub_500, cond_sub_500$ImmediateAccuracy == "1")

#for 500 ms, S36 and S42 have one row each missing

cond_sub_correct_500[99,] = c("36",1,"Neutral", 0 )
cond_sub_correct_500[100,] = c("42",1,"Phonological", 0 )

##  Merging immediate and delayed into one file:

## now we have conditional delayed accuracies in cond_sub_correct_500
## and we have immediate accuracies in prime_subject_acc_500
## now we must combine the two files.

colnames(cond_sub_correct_500) = c("Subject", "IA", "PrimeCondition", "Delayed")
colnames(prime_subject_acc_500) = c("Subject", "PrimeCondition", "Immediate",
                                    "DA", "RTImm", "RTDel")

final_combined_500 = merge(cond_sub_correct_500, prime_subject_acc_500,
                       by = c("Subject", "PrimeCondition"))

# now we have immediate and delayed accuracy in wide format, i.e. side to side.
# We need to convert to long format

#first we remove uneccesary columns, i.e. IA and DA

final_combined_500 = final_combined_500[, -c(3,6,7,8)]

# converting to long

library(tidyr)
final_long_500 <- final_combined_500 %>%
  gather(Delay, Accuracy, Immediate, Delayed)


final_long_500 <- final_long_500[order(final_long_500$Subject),]
final_long_500$Delay = as.factor(as.character(final_long_500$Delay))
final_long_500$Subject = as.factor(as.character(final_long_500$Subject))
final_long_500$Accuracy = as.numeric(as.character(final_long_500$Accuracy))
```

Conditional ANOVA

```
final_aov_500 = aov(data = final_long_500, Accuracy ~ Delay*PrimeCondition +
                  Error(Subject/(Delay*PrimeCondition)))
summary(final_aov_500)
```

```
##
```

```
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 19  9.147  0.4814
##
## Error: Subject:Delay
##           Df Sum Sq Mean Sq F value   Pr(>F)
## Delay      1  2.349  2.3495   42.78 2.91e-06 ***
## Residuals 19  1.044  0.0549
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Subject:PrimeCondition
##                Df Sum Sq Mean Sq F value Pr(>F)
## PrimeCondition  4 0.2012 0.05030   1.892   0.12
## Residuals      76 2.0198 0.02658
##
## Error: Subject:Delay:PrimeCondition
##                       Df Sum Sq Mean Sq F value   Pr(>F)
## Delay:PrimeCondition   4  2.178  0.5446   16.93 5.79e-10 ***
## Residuals             76  2.445  0.0322
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(multcomp)
library(knitr)
options(contrasts = c('contr.sum', 'contr.poly'))

conditional_lsm = lsmeans::lsmeans(final_aov_500, c("Delay", "PrimeCondition"))
prime_effect = cld(conditional_lsm, alpha = 0.05,
                adjust = "tukey", details = TRUE, by = "PrimeCondition")
kable(subset(prime_effect$comparisons,prime_effect$comparisons$p.value < 0.05 ))
```

	contrast	PrimeCondition	estimate	SE	df	t.ratio	p.value
2	Immediate - Delayed	Neutral	0.3904545	0.0605987	89.50216	6.443278	0.00e+00
3	Immediate - Delayed	Phonological	0.2636688	0.0605987	89.50216	4.351061	3.58e-05
5	Immediate - Delayed	UnSemantic	0.4607323	0.0605987	89.50216	7.603001	0.00e+00

Figures

32 ms

Immediate Recall

```
library(Rmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: plyr
```

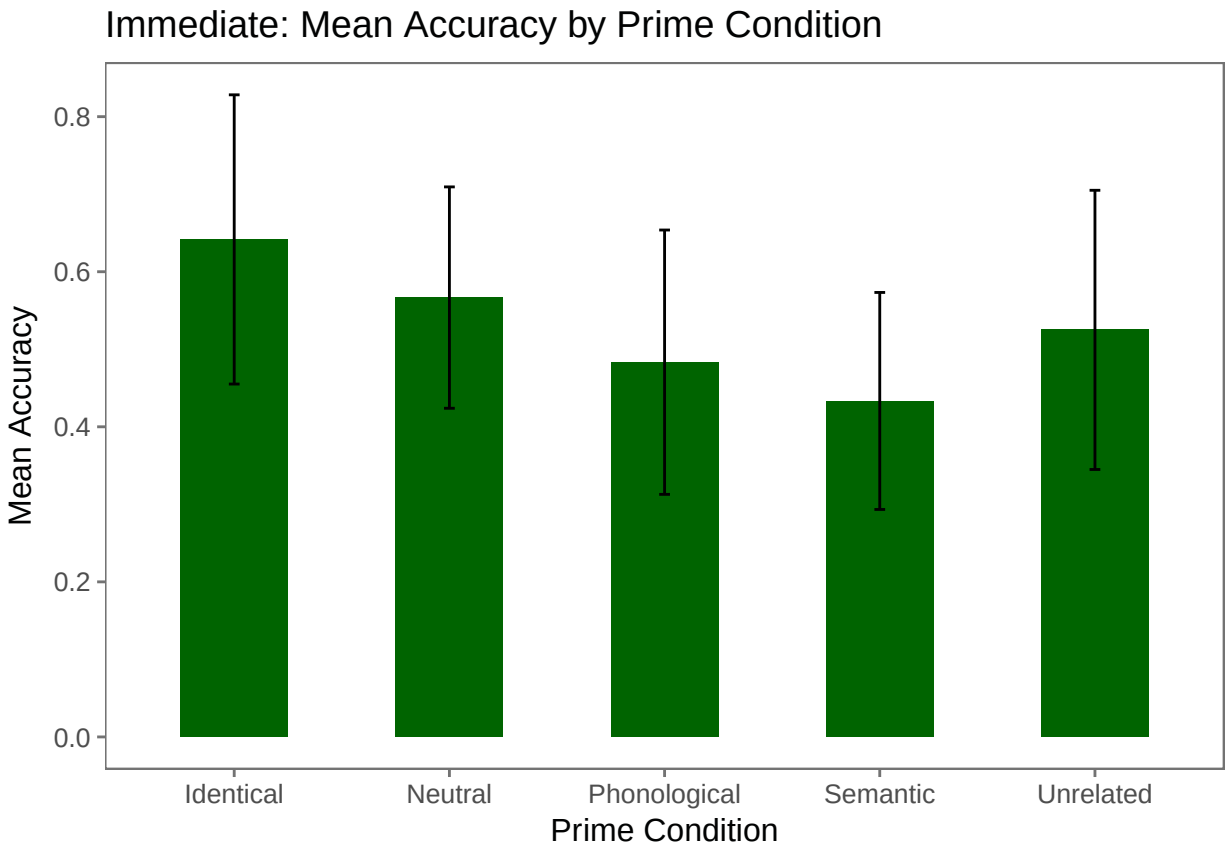
```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
```

```
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------
##
## Attaching package: 'plyr'
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

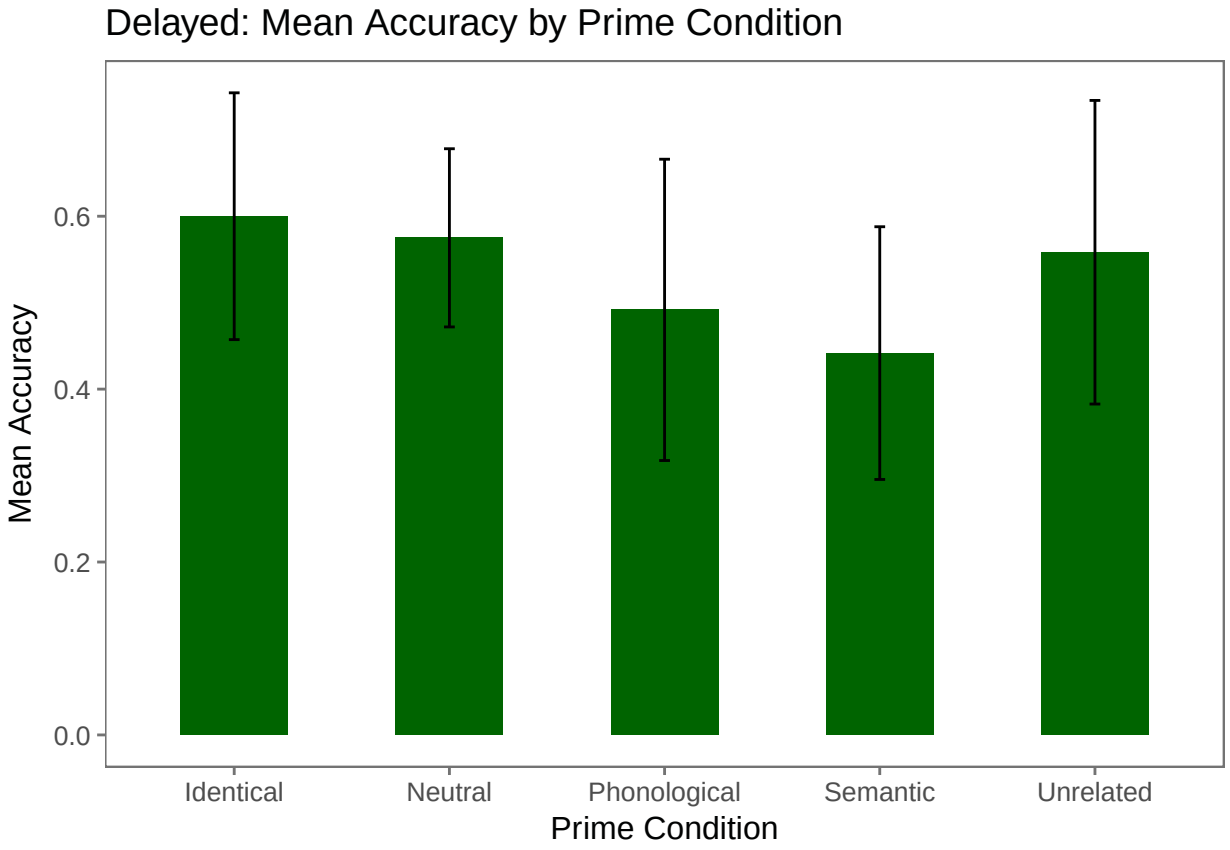
```
immediateagg_rmisc_32 = summarySE(prime_subject_acc_32,
                      measurevar = "Immediate",
                      groupvars = c("PrimeCondition"))

## IMMEDIATE
library(ggplot2)
library(ggthemes)
immediateagg_rmisc_32 %>% mutate(PrimeCondition = factor(PrimeCondition,
                                                levels = unique(PrimeCondition),
                    labels = c("Identical", "Neutral",
                               "Phonological", "Semantic", "Unrelated")))%>%
ggplot(aes(x = PrimeCondition, y = Immediate))+
 geom_bar(stat = "identity", position = "dodge", width = 0.5, fill = "darkgreen")+
  geom_errorbar(aes(ymin = Immediate - ci, ymax = Immediate + ci),
                width=.05, position=position_dodge(.5)) +
  theme_few()+
  xlab("Prime Condition") + ylab("Mean Accuracy") +
  ggtitle("Immediate: Mean Accuracy by Prime Condition")
```



Delayed Recall

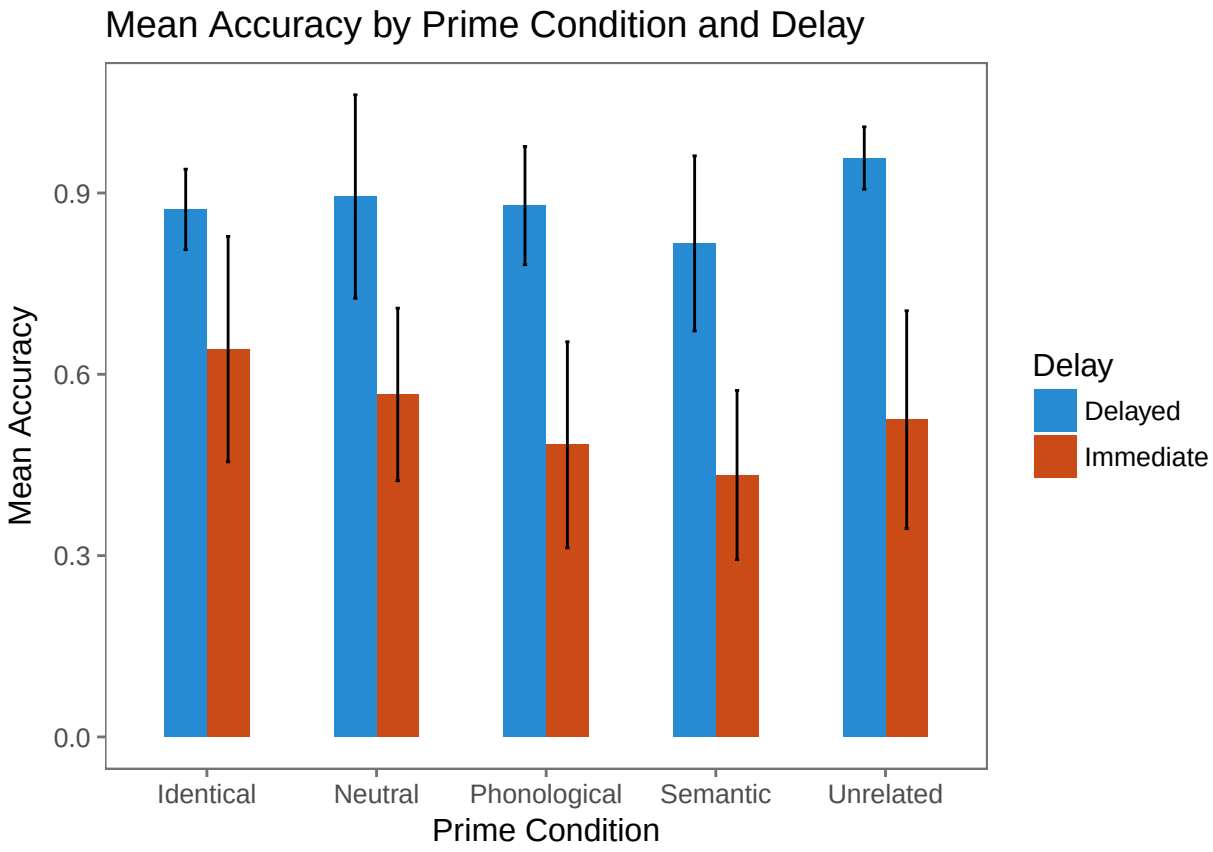
```
library(Rmisc)
delayedagg_rmisc_32 = summarySE(prime_subject_acc_32,
                      measurevar = "DA",
                      groupvars = c("PrimeCondition"))
#DELAYED
delayedagg_rmisc_32 %>% mutate(PrimeCondition = factor(PrimeCondition,
                                                       levels = unique(PrimeCondition),
                    labels = c("Identical", "Neutral",
                               "Phonological", "Semantic", "Unrelated")))%>%
ggplot(aes(x = PrimeCondition, y = DA))+
 geom_bar(stat = "identity", position = "dodge", width = 0.5, fill = "darkgreen")+
  geom_errorbar(aes(ymin = DA - ci, ymax = DA + ci),
                width=.05, position=position_dodge(.5)) +
  theme_few()+
  xlab("Prime Condition") + ylab("Mean Accuracy") +
  ggtitle("Delayed: Mean Accuracy by Prime Condition")
```



Conditional Recall

```
## PLOTTING INTERACTION
agg_final_combined_32 = group_by(final_long_32, Delay, PrimeCondition)%>%
  summarise_at(vars(Accuracy), mean)
library(Rmisc)
agg_rmisc_32 = summarySE(final_long_32,
                      measurevar = "Accuracy",
                      groupvars = c("Delay", "PrimeCondition"))

agg_rmisc_32 %>% mutate(PrimeCondition = factor(PrimeCondition,
                                                levels = unique(PrimeCondition),
                    labels = c("Identical", "Neutral",
                               "Phonological", "Semantic", "Unrelated")))%>%
  ggplot(aes(x = PrimeCondition, y = Accuracy, fill = Delay))+
  geom_bar(stat = "identity", position = "dodge", width = 0.5)+
  geom_errorbar(aes(ymin = Accuracy - ci, ymax = Accuracy + ci),
                width=.05, position=position_dodge(.5)) +
  theme_few()+
  scale_fill_solarized()+
  xlab("Prime Condition") + ylab("Mean Accuracy") +
  ggtitle("Mean Accuracy by Prime Condition and Delay")
```

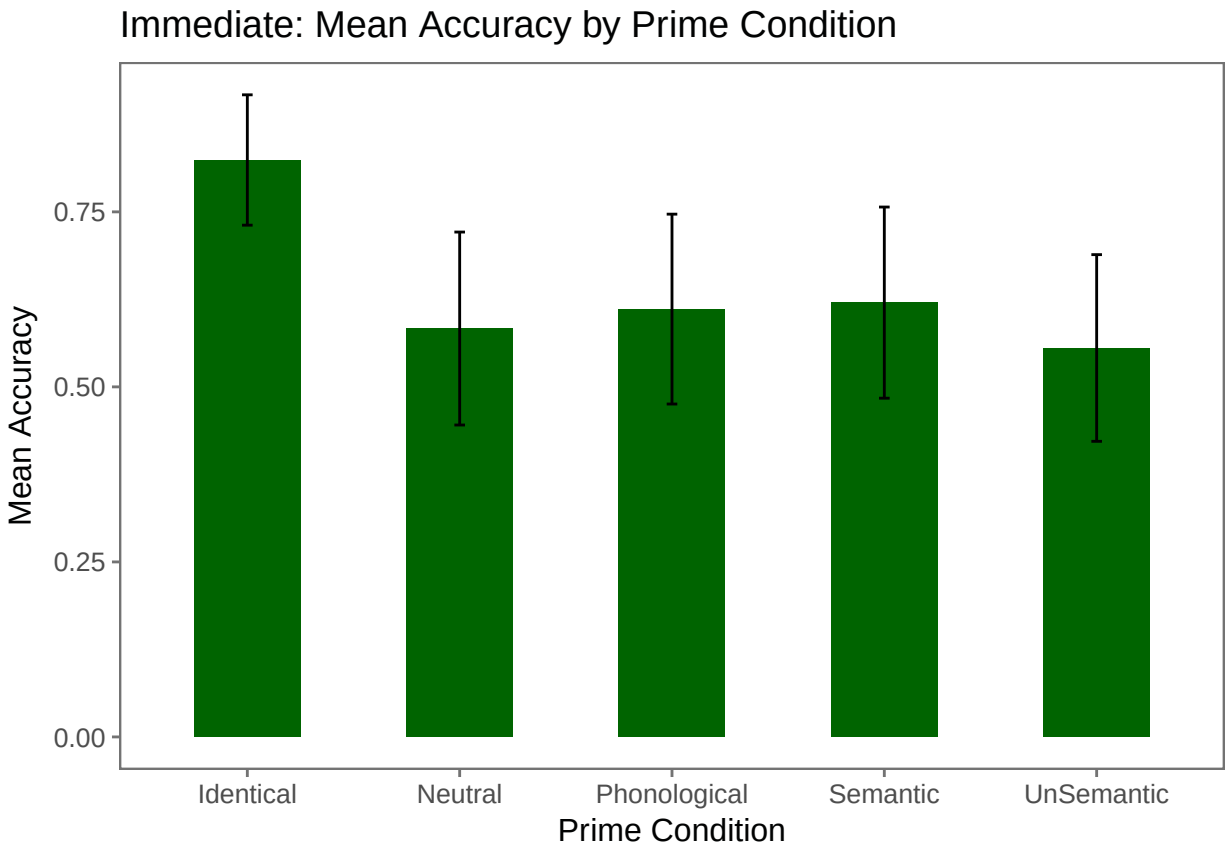


48ms

Immediate Recall

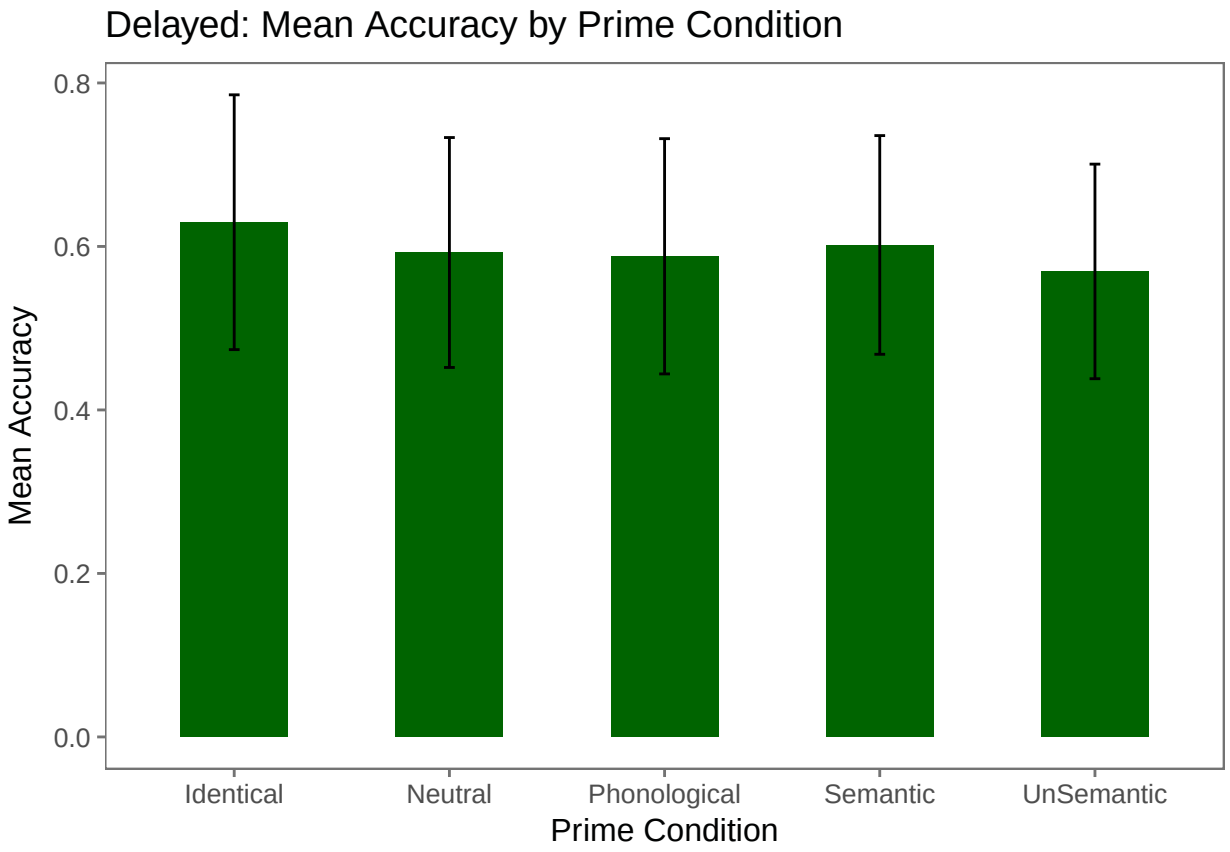
```
library(Rmisc)
immediateagg_rmisc_48 = summarySE(prime_subject_acc_48,
                      measurevar = "Immediate",
                      groupvars = c("PrimeCondition"))

## IMMEDIATE
library(ggplot2)
library(ggthemes)
ggplot(immediateagg_rmisc_48, aes(x = PrimeCondition, y = Immediate))+
 geom_bar(stat = "identity", position = "dodge", width = 0.5, fill = "darkgreen")+
  geom_errorbar(aes(ymin = Immediate - ci, ymax = Immediate + ci),
                width=.05, position=position_dodge(.5)) +
  theme_few()+
  xlab("Prime Condition") + ylab("Mean Accuracy") +
  ggtitle("Immediate: Mean Accuracy by Prime Condition")
```



Delayed Recall

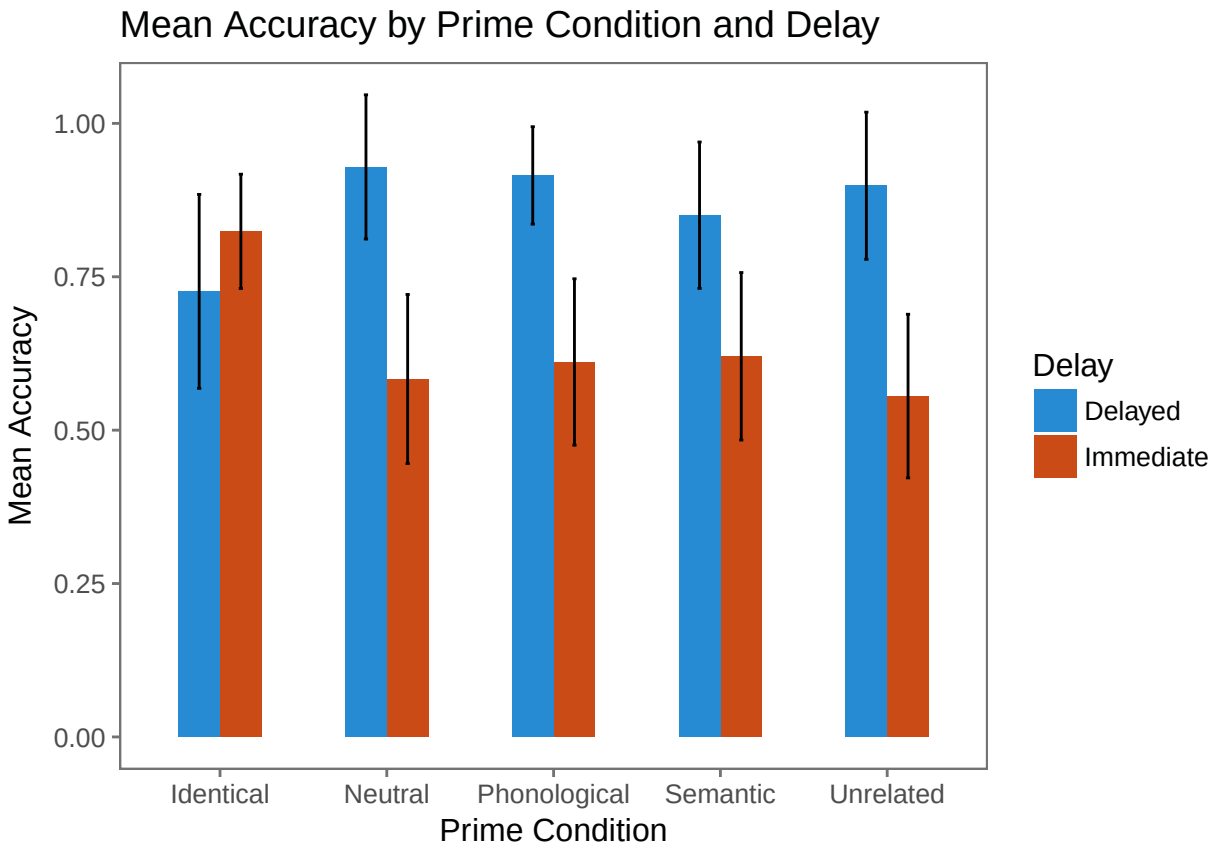
```
library(Rmisc)
delayedagg_rmisc_48 = summarySE(prime_subject_acc_48,
                      measurevar = "DA",
                      groupvars = c("PrimeCondition"))
#DELAYED
ggplot(delayedagg_rmisc_48, aes(x = PrimeCondition, y = DA))+
 geom_bar(stat = "identity", position = "dodge", width = 0.5, fill = "darkgreen")+
  geom_errorbar(aes(ymin = DA - ci, ymax = DA + ci),
                width=.05, position=position_dodge(.5)) +
  theme_few()+
  xlab("Prime Condition") + ylab("Mean Accuracy") +
  ggtitle("Delayed: Mean Accuracy by Prime Condition")
```



Conditional Recall

```
## PLOTTING INTERACTION
agg_final_combined_48 = group_by(final_long_48, Delay, PrimeCondition)%>%
  summarise_at(vars(Accuracy), mean)
library(Rmisc)
agg_rmisc_48 = summarySE(final_long_48,
                      measurevar = "Accuracy",
                      groupvars = c("Delay", "PrimeCondition"))

agg_rmisc_48 %>% mutate(PrimeCondition = factor(PrimeCondition,
                                                levels = unique(PrimeCondition),
                    labels = c("Identical", "Neutral",
                               "Phonological", "Semantic", "Unrelated")))%>%
  ggplot(aes(x = PrimeCondition, y = Accuracy, fill = Delay))+
  geom_bar(stat = "identity", position = "dodge", width = 0.5)+
  geom_errorbar(aes(ymin = Accuracy - ci, ymax = Accuracy + ci),
                width=.05, position=position_dodge(.5)) +
  theme_few()+
  scale_fill_solarized()+
  xlab("Prime Condition") + ylab("Mean Accuracy") +
  ggtitle("Mean Accuracy by Prime Condition and Delay")
```

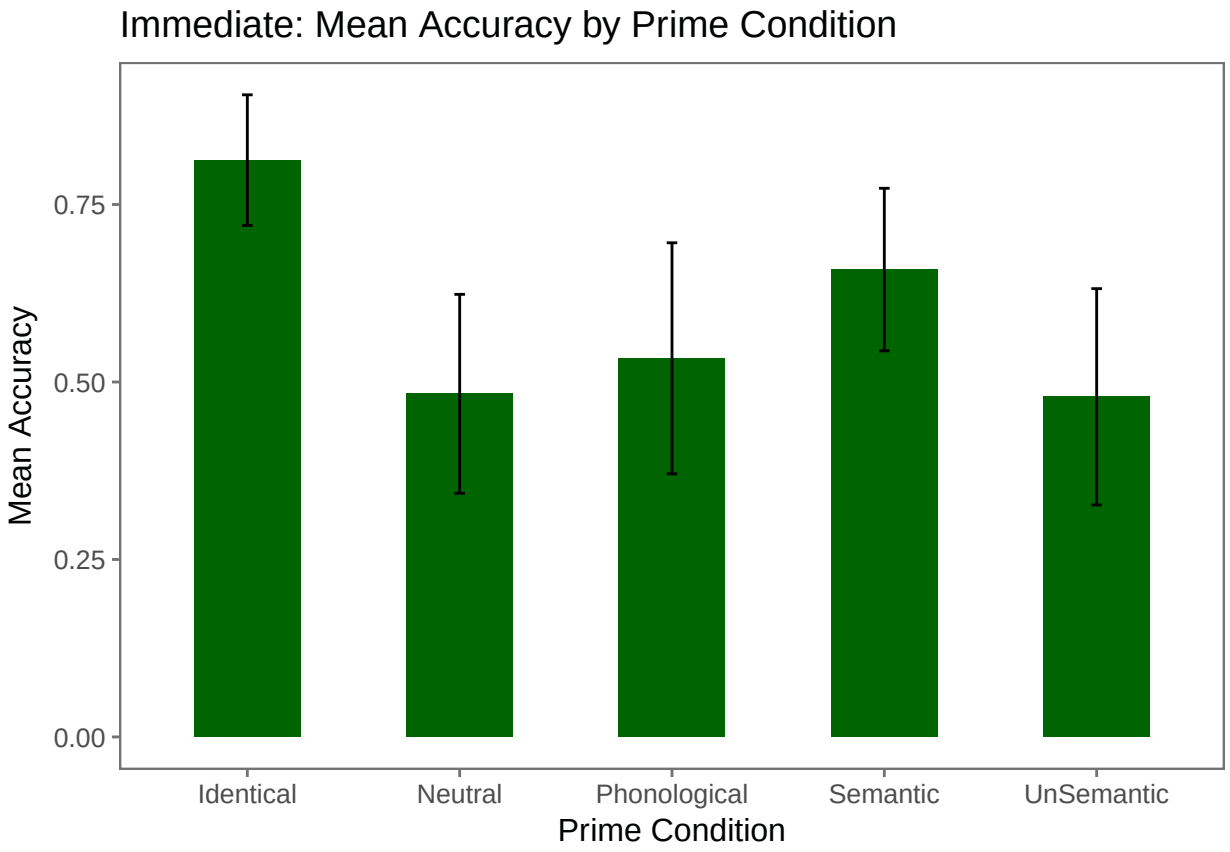


500 ms

Immediate Recall

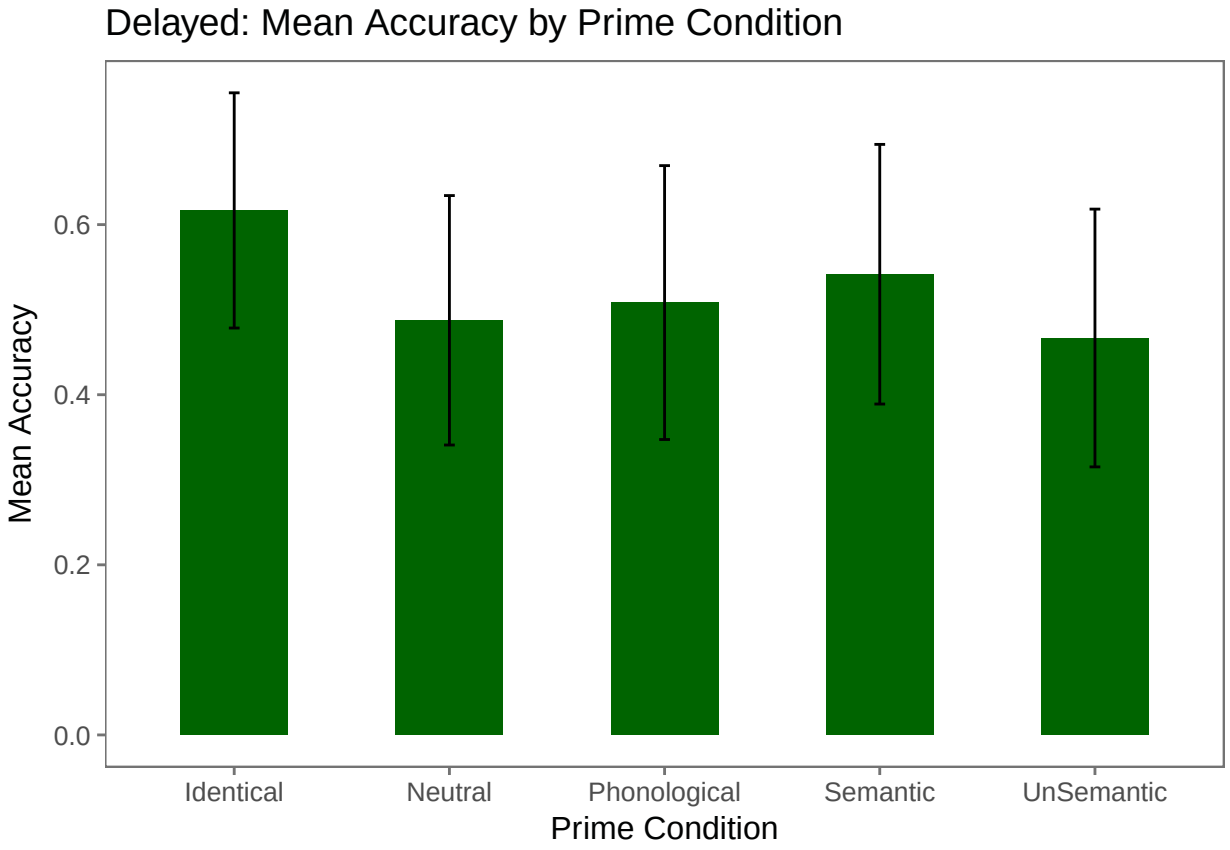
```
library(Rmisc)
immediateagg_rmisc_500 = summarySE(prime_subject_acc_500,
                      measurevar = "Immediate",
                      groupvars = c("PrimeCondition"))

## IMMEDIATE
library(ggplot2)
library(ggthemes)
ggplot(immediateagg_rmisc_500, aes(x = PrimeCondition, y = Immediate))+
 geom_bar(stat = "identity", position = "dodge", width = 0.5, fill = "darkgreen")+
  geom_errorbar(aes(ymin = Immediate - ci, ymax = Immediate + ci),
                width=.05, position=position_dodge(.5)) +
  theme_few()+
  xlab("Prime Condition") + ylab("Mean Accuracy") +
  ggtitle("Immediate: Mean Accuracy by Prime Condition")
```



Delayed Recall

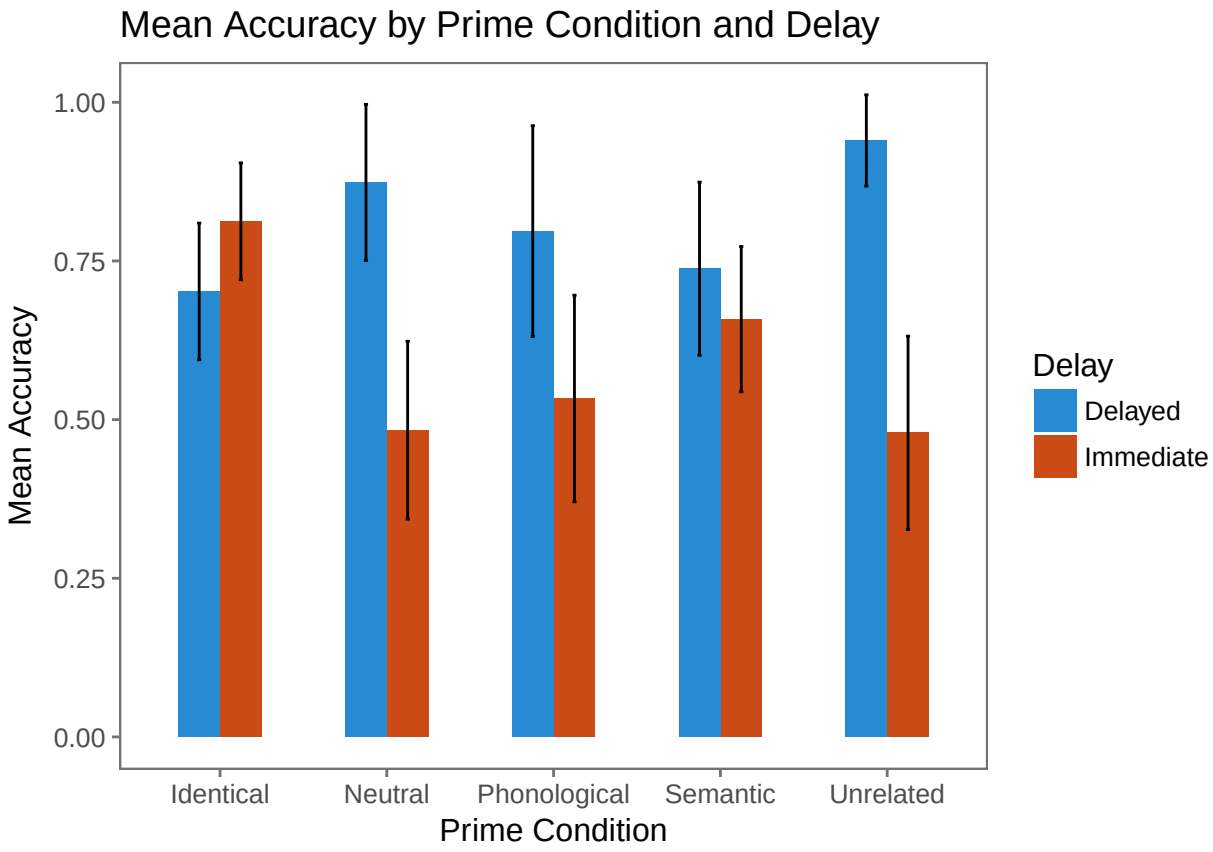
```
library(Rmisc)
delayedagg_rmisc_500 = summarySE(prime_subject_acc_500,
                      measurevar = "DA",
                      groupvars = c("PrimeCondition"))
#DELAYED
ggplot(delayedagg_rmisc_500, aes(x = PrimeCondition, y = DA))+
 geom_bar(stat = "identity", position = "dodge", width = 0.5, fill = "darkgreen")+
  geom_errorbar(aes(ymin = DA - ci, ymax = DA + ci),
                width=.05, position=position_dodge(.5)) +
  theme_few()+
  xlab("Prime Condition") + ylab("Mean Accuracy") +
  ggtitle("Delayed: Mean Accuracy by Prime Condition")
```



Conditional Recall

```
## PLOTTING INTERACTION
agg_final_combined_500 = group_by(final_long_500, Delay, PrimeCondition)%>%
  summarise_at(vars(Accuracy), mean)
library(Rmisc)
agg_rmisc_500 = summarySE(final_long_500,
                      measurevar = "Accuracy",
                      groupvars = c("Delay", "PrimeCondition"))

agg_rmisc_500 %>% mutate(PrimeCondition = factor(PrimeCondition,
                                                 levels = unique(PrimeCondition),
                    labels = c("Identical", "Neutral",
                               "Phonological", "Semantic", "Unrelated")))%>%
  ggplot(aes(x = PrimeCondition, y = Accuracy, fill = Delay))+
 geom_bar(stat = "identity", position = "dodge", width = 0.5)+
  geom_errorbar(aes(ymin = Accuracy - ci, ymax = Accuracy + ci),
                width=.05, position=position_dodge(.5)) +
  theme_few()+
  scale_fill_solarized()+
  xlab("Prime Condition") + ylab("Mean Accuracy") +
  ggtitle("Mean Accuracy by Prime Condition and Delay")
```


Mean Accuracy by Prime Condition and Delay
Mean Accuracy
1.00
0.75
0.50
0.25
0.00
Identical
Neutral
Phonological
Semantic
Unrelated
Prime Condition
Delay
Delayed
Immediate