Domain-general cognitive motivation: evidence from economic decision-making

Jennifer L. Crawford

8/24/2021

Summary of findings from online data collection for multi-domain discounting project

104 Younger adult participants

```
rm(list=ls())

#Packages
library(brms); library(bayestestR); library(BayesFactor); library(correlation); library(knitr);
library(RColorBrewer); library(tidyverse);
source("summarySEwithin2.R")

#Package versions used to run the analyses reported in the manuscript
##R version 4.1.0
##BayesFactor_0.9.12-4.2, bayestestR_0.10.5, brms_2.16.1, correlation_0.6.1, knitr_1.33,
##RColorBrewer_1.1-2, tidyverse_1.3.1

#Create data directories
#Cog-ED
coged.wm.path<-"https://raw.githubusercontent.com/jlcrawford/MDD/master/Data/Online/Discounting/MDD_WMC
coged.wm.full.path <-"https://raw.githubusercontent.com/jlcrawford/MDD/master/Data/Online/Discounting/M
coged.speech.path<-"https://raw.githubusercontent.com/jlcrawford/MDD/master/Data/Online/Discounting/MDD
#Individual Difference Questionnaires
NCS.path <- "https://raw.githubusercontent.com/jlcrawford/MDD/master/Data/Online/Individual-Difference-
SPSRQ.path <- "https://raw.githubusercontent.com/jlcrawford/MDD/master/Data/Online/Individual-Differenc
BISBAS.path <- "https://raw.githubusercontent.com/jlcrawford/MDD/master/Data/Online/Individual-Differen
GRAPES.path <- "https://raw.githubusercontent.com/jlcrawford/MDD/master/Data/Online/Individual-Differen
#Working Memory Capacity
LSpan.path <- "https://raw.githubusercontent.com/jlcrawford/MDD/master/Data/Online/Working-Memory-Capac
OSpan.path <- "https://raw.githubusercontent.com/jlcrawford/MDD/master/Data/Online/Working-Memory-Capac
SymmSpan.path <- "https://raw.githubusercontent.com/jlcrawford/MDD/master/Data/Online/Working-Memory-Ca
#complete subject list
subjects.path <- "https://raw.githubusercontent.com/jlcrawford/MDD/master/Data/Online/Discounting/subje
#Make data frames for Cog-ED(s)
coged.wm<- read.csv(coged.wm.path, header = T)
coged.wm.full <- read.csv(coged.wm.full.path, header = T)
coged.speech<- read.csv(coged.speech.path, header = T)
#Make data frames for individual difference questionnaires
```

```
NCS <- read.csv(NCS.path, header = T)
SPSRQ <- read.csv(SPSRQ.path, header = T)
BISBAS <- read.csv(BISBAS.path, header = T)
GRAPES <- read.csv(GRAPES.path, header = T)
#Make data frames for working memory capacity tasks
LSpan <- read.csv(LSpan.path, header = T)
OSpan <- read.csv(OSpan.path, header = T)
SymmSpan <- read.csv(SymmSpan.path, header = T)
#Make data frame for usable subject info (i.e., subjects who have completed all tasks and questionnaire
Subjects <- read.csv(subjects.path, header = F)
colnames(Subjects) <- "subjectid"
```

Cognitive Effort Discounting

Testing for the effects of task load and domain (e.g., working memory, speech) on subjective value

```
#Cog-ED Data
#clean data frame(s) with Cog-ED subjective value (SV) estimates and transform data so that SV estimate
#across both domains (i.e., speech, WM)

##working memory
d.coged.wm <- coged.wm %>% select(subjectid,completed,fixedAmount_N2_1,fixedAmount_N2_2,fixedAmount_N2_
                            fixedAmount_N3_1,fixedAmount_N3_2,fixedAmount_N3_3,fixedAmount_N4_1,
                            fixedAmount_N4_2,fixedAmount_N4_3,
                            IP12_1,IP12_2,IP12_3,IP13_1,IP13_2,IP13_3,IP14_1,IP14_2,IP14_3) %>%
  filter(completed == 1) %>%
  group_by(subjectid) %>%
  mutate(Domain = "WM",
         domainCode = 0,
         SV2_1 = ifelse(fixedAmount_N2_1 == "X", IP12_1/2, ((2-IP12_1)/2)+1),
         SV2_2 = ifelse(fixedAmount_N2_2 == "X", IP12_2/3, ((3-IP12_2)/3)+1),
         SV2_3 = ifelse(fixedAmount_N2_3 == "X", IP12_1/4, ((4-IP12_3)/4)+1),
         SV3_1 = ifelse(fixedAmount_N3_1 == "X", IP13_1/2, ((2-IP13_1)/2)+1),
         SV3_2 = ifelse(fixedAmount_N3_2 == "X", IP13_2/3, ((3-IP13_2)/3)+1),
         SV3_3 = ifelse(fixedAmount_N3_3 == "X", IP13_1/4, ((4-IP13_3)/4)+1),
         SV4_1 = ifelse(fixedAmount_N4_1 == "X", IP14_1/2, ((2-IP14_1)/2)+1),
         SV4_2 = ifelse(fixedAmount_N4_2 == "X", IP14_2/3, ((3-IP14_2)/3)+1),
         SV4_3 = ifelse(fixedAmount_N4_3 == "X", IP14_3/4, ((4-IP14_3)/4)+1),
         SV_red = (SV2_1 + SV2_2 + SV2_3)/3,
         SV_blue = (SV3_1 + SV3_2 + SV3_3)/3,
         SV_purple = (SV4_1 + SV4_2 + SV4_3)/3)

##speech comprehension
d.coged.speech <- coged.speech %>%
  select(subjectid,completed,fixedAmount_N2_1,fixedAmount_N2_2,fixedAmount_N2_3,fixedAmount_N3_1,
                            fixedAmount_N3_2,fixedAmount_N3_3,fixedAmount_N4_1,
                            fixedAmount_N4_2,fixedAmount_N4_3,
                            IP12_1,IP12_2,IP12_3,IP13_1,IP13_2,IP13_3,IP14_1,IP14_2,IP14_3) %>%
  filter(completed == 1) %>%
  group_by(subjectid) %>%
```

```r
  mutate(Domain = "Speech",
         domainCode = 1,
         SV2_1 = ifelse(fixedAmount_N2_1 == "X", IP12_1/2, ((2-IP12_1)/2)+1),
         SV2_2 = ifelse(fixedAmount_N2_2 == "X", IP12_2/3, ((3-IP12_2)/3)+1),
         SV2_3 = ifelse(fixedAmount_N2_3 == "X", IP12_1/4, ((4-IP12_3)/4)+1),
         SV3_1 = ifelse(fixedAmount_N3_1 == "X", IP13_1/2, ((2-IP13_1)/2)+1),
         SV3_2 = ifelse(fixedAmount_N3_2 == "X", IP13_2/3, ((3-IP13_2)/3)+1),
         SV3_3 = ifelse(fixedAmount_N3_3 == "X", IP13_1/4, ((4-IP13_3)/4)+1),
         SV4_1 = ifelse(fixedAmount_N4_1 == "X", IP14_1/2, ((2-IP14_1)/2)+1),
         SV4_2 = ifelse(fixedAmount_N4_2 == "X", IP14_2/3, ((3-IP14_2)/3)+1),
         SV4_3 = ifelse(fixedAmount_N4_3 == "X", IP14_3/4, ((4-IP14_3)/4)+1),
         SV_red = (SV2_1 + SV2_2 + SV2_3)/3,
         SV_blue = (SV3_1 + SV3_2 + SV3_3)/3,
         SV_purple = (SV4_1 + SV4_2 + SV4_3)/3)

#Merge WM and Speech Cog-ED data frames
coged.merged <- rbind(d.coged.wm, d.coged.speech)

#Filter out subjects who have not completed all tasks in the protocol
coged.merged <-inner_join(Subjects, coged.merged)

#Add dummy variables (task, domain) for multilevel models
d.coged.SV <- coged.merged %>% select(subjectid,Domain,domainCode, SV_red, SV_blue, SV_purple) %>%
  pivot_longer(names_to = "tmp", values_to = "SV", -c(subjectid,Domain,domainCode)) %>%
  separate(col = tmp, into=c(NA,"Task"), sep = "_") %>%
  mutate(taskCode = factor(Task, levels=c("red","blue","purple"), labels=c(-1,0,1)))
d.coged.SV$taskCode <- as.numeric(d.coged.SV$taskCode)
d.coged.SV$domainCode <- as.numeric(d.coged.SV$domainCode)
#Multilevel model of subjective value with task and domain as predictors
m.SV.coged <- brm(data = d.coged.SV, SV ~ taskCode*domainCode + (1 | subjectid),
file = "models/m.SV.coged.rds")
summary(m.SV.coged)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: SV ~ taskCode * domainCode + (1 | subjectid)
##    Data: d.coged.SV (Number of observations: 624)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Group-Level Effects:
## ~subjectid (Number of levels: 104)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.21      0.02     0.18     0.25 1.00     1461     2568
##
## Population-Level Effects:
##                     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept               0.66      0.05     0.56     0.74 1.00     1460     2741
## taskCode               -0.09      0.02    -0.13    -0.05 1.00     2974     3419
## domainCode              0.12      0.06     0.01     0.24 1.00     2729     2967
## taskCode:domainCode    -0.10      0.03    -0.15    -0.04 1.00     2566     2485
##
## Family Specific Parameters:
```

```
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.27      0.01     0.26     0.29 1.00     4023     2842
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
bayes_R2(m.SV.coged)
```

```
##     Estimate  Est.Error      Q2.5     Q97.5
## R2 0.4459888 0.02596868 0.3920713 0.4930921
```

Plotting SV estimates across working memory and speech comprehension domains

Figure 1

```
#Plotting SV estimates
#Create summary stats for group plots
CogED_sum <- summarySEwithin2(d.coged.SV, measurevar = "SV", withinvars = c("Task","Domain"),
                              idvar = "subjectid")
CogED_sum$Task <- factor(CogED_sum$Task, levels = c("red","blue","purple"),
                         labels = c("red","blue","purple"))
CogED_sum$Domain <- factor(CogED_sum$Domain, levels = c("WM","Speech"),
                           labels = c("WM","Speech"))

#Plotting SV across both gain and loss domains
##Create domain labels
Domain.labs <- c("Speech Comprehension", "Working Memory")
names(Domain.labs) <- c("Speech", "WM")

#Figure 1
##Subjective value estimates across working memory and speech comprehension domains
###Low effort = 2-back; -4 SNR, Medium effort = 3-back;-8 SNR, High effort = 4-back;-12 SNR
fig.1 <- ggplot(CogED_sum, aes(x=Task, y=SV, fill=Task, color=Task)) +
  theme(plot.title = element_text(hjust = 0.5), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), panel.background = element_blank(),
        axis.line = element_line(colour = "black"), axis.title.y = element_text(face="bold",
        size=16),legend.title = element_text(face="bold", size=16)) +
  geom_bar(stat="identity", position=position_dodge(), alpha=.45, size=1.5) +
  geom_errorbar(position=position_dodge(width=0.9), aes(ymin=SV-ci, ymax=SV+ci), width=.2, size=1.25) +
  geom_point(data = d.coged.SV, aes(x=Task, y=SV, color=Task),
             stat="identity", alpha=0.7, position = "jitter") +
  scale_x_discrete(breaks=NULL) +
  xlab("") + ylab("Subjective Value") +
  facet_wrap(.~ Domain, labeller = labeller(Domain = Domain.labs))
fig.1 + scale_fill_brewer(palette = "Set1",  name="Task Effort Level", labels=c("Low","Medium","High"))
  scale_color_brewer(palette = "Set1", name="Task Effort Level",labels=c("Low","Medium","High"))
```

Stage I of Analysis: Zero-order correlation between average subjective value across working memory and speech comprehension domains

```
#Correlating Average SV (within-subject) across working memory and speech comprehesion domains
##Creating data frame with SV estimates across both domains
average.SV <- coged.merged %>% select(subjectid, Domain, SV_red, SV_blue, SV_purple) %>%
  group_by(subjectid, Domain) %>%
  dplyr::summarise(SV_avg = (SV_red + SV_blue + SV_purple)/3) %>%
  pivot_wider(values_from = "SV_avg", names_from = "Domain")

#T-tests between average SV estimates across WM and speech domains
##Average across all effort levels
SV.t <- t.test(x= average.SV$WM, y= average.SV$Speech, paired = TRUE)
SV.t
```

```
##
##  Paired t-test
##
## data:  average.SV$WM and average.SV$Speech
## t = 2.1175, df = 103, p-value = 0.03663
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.004621649 0.141211685
```

```
## sample estimates:
## mean of the differences
##              0.07291667
```

```
SV.BF <- ttestBF(x= average.SV$WM, y= average.SV$Speech, paired = TRUE)
SV.BF
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 0.9222015 ±0%
##
## Against denominator:
##   Null, mu = 0
## ---
## Bayes factor type: BFoneSample, JZS
```

```
#Low effort (2-back; -4SNR)
SV.red.t <- t.test(x= d.coged.SV$SV[d.coged.SV$Domain=="WM" & d.coged.SV$Task=="red"],
                   y= d.coged.SV$SV[d.coged.SV$Domain=="Speech" & d.coged.SV$Task=="red"],
                   paired = TRUE)
SV.red.t
```

```
##
##  Paired t-test
##
## data:  d.coged.SV$SV[d.coged.SV$Domain == "WM" & d.coged.SV$Task == "red"] and d.coged.SV$SV[d.coged
## t = -0.74026, df = 103, p-value = 0.4608
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.11580290  0.05285218
## sample estimates:
## mean of the differences
##             -0.03147536
```

```
SV.red.BF <- ttestBF(x= d.coged.SV$SV[d.coged.SV$Domain=="WM" & d.coged.SV$Task=="red"],
                     y= d.coged.SV$SV[d.coged.SV$Domain=="Speech" & d.coged.SV$Task=="red"],
                     paired = TRUE)
summary(SV.red.BF)
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 0.1417658 ±0.08%
##
## Against denominator:
##   Null, mu = 0
## ---
## Bayes factor type: BFoneSample, JZS
```

```
#Medium effort (3-back; -8 SNR)
SV.blue.t <- t.test(x= d.coged.SV$SV[d.coged.SV$Domain=="WM" & d.coged.SV$Task=="blue"],
                    y= d.coged.SV$SV[d.coged.SV$Domain=="Speech" & d.coged.SV$Task=="blue"],
                    paired = TRUE)
SV.blue.t
```

```
##
##  Paired t-test
##
## data:  d.coged.SV$SV[d.coged.SV$Domain == "WM" & d.coged.SV$Task == "blue"] and d.coged.SV$SV[d.coged
## t = 2.1855, df = 103, p-value = 0.03111
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.008215902 0.169318153
## sample estimates:
## mean of the differences
##              0.08876703
```

```
SV.blue.BF <- ttestBF(x= d.coged.SV$SV[d.coged.SV$Domain=="WM" & d.coged.SV$Task=="blue"],
                      y= d.coged.SV$SV[d.coged.SV$Domain=="Speech" & d.coged.SV$Task=="blue"],
                      paired = TRUE)
summary(SV.blue.BF)
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 1.05727 ±0%
##
## Against denominator:
##   Null, mu = 0
## ---
## Bayes factor type: BFoneSample, JZS
```

```
#High effort (4-back; -12SNR)
SV.purple.t <- t.test(x= d.coged.SV$SV[d.coged.SV$Domain=="WM" & d.coged.SV$Task=="purple"],
                      y= d.coged.SV$SV[d.coged.SV$Domain=="Speech" & d.coged.SV$Task=="purple"],
                      paired = TRUE)
SV.purple.t
```

```
##
##  Paired t-test
##
## data:  d.coged.SV$SV[d.coged.SV$Domain == "WM" & d.coged.SV$Task == "purple"] and d.coged.SV$SV[d.co
## t = 4.1322, df = 103, p-value = 7.331e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.08396587 0.23895080
## sample estimates:
## mean of the differences
##               0.1614583
```

```
SV.purple.BF <- ttestBF(x= d.coged.SV$SV[d.coged.SV$Domain=="WM" & d.coged.SV$Task=="purple"],
                        y= d.coged.SV$SV[d.coged.SV$Domain=="Speech" & d.coged.SV$Task=="purple"],
                        paired = TRUE)
summary(SV.purple.BF)
```

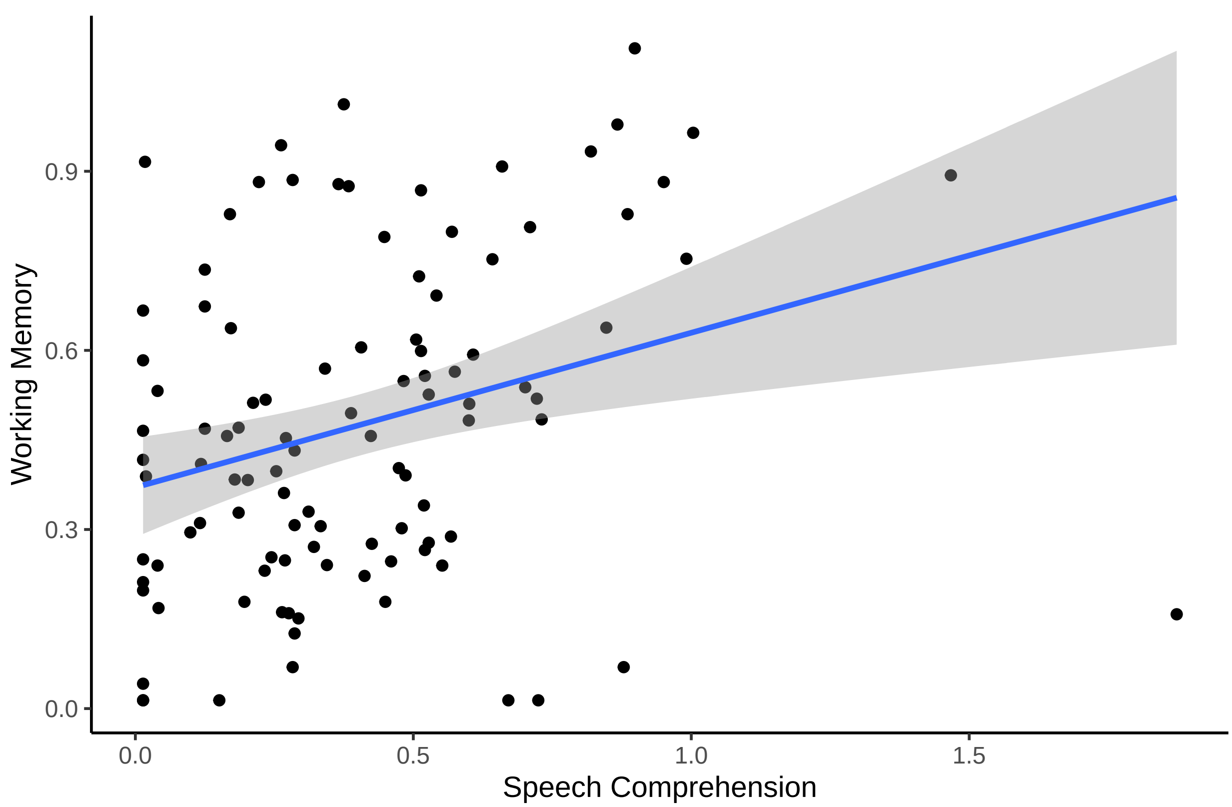
```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 241.976 ±0%
```

```
##
## Against denominator:
##   Null, mu = 0
## ---
## Bayes factor type: BFoneSample, JZS
```

```
#Testing for correlation between cognitive effort discounting across working memory & speech domains
CogED.cor <- cor_test(data = average.SV, x = "Speech", y = "WM", bayesian = TRUE, bayesian_prior = 0.707
#Summarize Bayes Factor from correlation
CogED.cor
```

```
## Parameter1 | Parameter2 |  rho |       95% CI |       pd | % in ROPE |               Prior |      BF
## ---------------------------------------------------------------------------------------------------
## Speech     |         WM | 0.29 | [0.14, 0.42] | 99.83%** |     2.17% | Beta (1.41 +- 1.41) | 14.69**
##
## Observations: 104
```

```
#Plot of correlation between working memory & speech comprehension domains
##Not included in manuscript
fig.corr <- ggplot(average.SV, aes(Speech, WM)) +
   theme(plot.title = element_text(hjust = 0.5), panel.grid.major = element_blank(),
         panel.grid.minor = element_blank(), panel.background = element_blank(),
         axis.line = element_line(colour = "black")) +
  geom_point() + geom_smooth(method=lm) +ggtitle("") +
    xlab("Speech Comprehension") + ylab("Working Memory")
fig.corr
```



Stage II of Analysis: Partialing out task level and performance on subjective value estimates

Familiarization Phase Performance

Speech

```
#Summarizing intelligibility ratings across all speech-in-noise task levels
performance.speech <- coged.speech %>% select(subjectid, completed, percentCorrect_N1, percentCorrect_N
   group_by(subjectid) %>% filter(completed == 1) %>% select(-completed) %>%
  pivot_longer(names_to = "level", values_to = "performance", -c(subjectid)) %>%
  separate(col = level, into=c(NA,"Task"), sep = "_") %>% inner_join(Subjects)
performance.speech$task <- factor(performance.speech$Task, levels = c("N1","N2","N3","N4"), labels = c(

performance_sum <- summarySEwithin2(performance.speech, measurevar = "performance", withinvars = c("Tas
performance_sum$Task <- factor(performance_sum$Task, levels = c("N1","N2","N3","N4"), labels = c("0 SNR
```

Working Memory

```
#Summarizing working memory performance on the N-back across all load levels
##Hit rate, correct rejections
performance.wm <- coged.wm %>% select(subjectid, completed, hitrate_N1, CRrate_N1, hitrate_N2, CRrate_N
   group_by(subjectid) %>% filter(completed == 1) %>% select(-completed) %>%
  pivot_longer(names_to = "level", values_to = "performance", -c(subjectid)) %>%
  separate(col = level, into=c("Metric","Task"), sep = "_") %>%
  pivot_wider(names_from = Metric, values_from = performance)  %>% inner_join(Subjects)
performance.wm$task <- factor(performance.wm$Task, levels = c("N1","N2","N3","N4"), labels = c("black",

performance.sum.wm <- summarySEwithin2(performance.wm, measurevar = "hitrate", withinvars = c("Task"), 
performance.sum.wm$Task <- factor(performance.sum.wm$Task, levels = c("N1","N2","N3","N4"), labels = c(

##RT
performance.wm.RT <- coged.wm.full %>% select(subjectid, blockcode, phase, response, latency) %>% filte
  rename(task = "blockcode") %>% filter(task != "ratingSummary") %>% filter(response != 0)
performance.wm.RT$task <- factor(performance.wm.RT$task, levels = c("1back","2back","3back","4back"), la

performance.wm.RT.sum <-inner_join(performance.wm.RT, Subjects) %>% group_by(subjectid, task) %>% summa
```

Correlation between averaged residuals from models controlling for the effects of task level & performance

```
#Working Memory Cog-ED
#Summarize average performance on N-Back
d.coged.wm.clean <- d.coged.wm %>% select(subjectid, SV_red, SV_blue, SV_purple) %>%
  pivot_longer(names_to = "tmp", values_to = "SV", -subjectid) %>%
   separate(col = tmp, into=c(NA,"task"), sep = "_")

d.coged.wm.partial <- d.coged.wm.clean %>% group_by(subjectid,task) %>% summarise(meanSV = mean(SV)) %>
  inner_join(performance.wm, by = c("subjectid","task")) %>% inner_join(performance.wm.RT.sum) %>%
```

```r
  mutate(taskCode = factor(task, levels =c( "black","red","blue","purple"), labels = c(-2,-1,0,1)),
         FArate = 1-CRrate,
         HR_z = scale(hitrate),
         FAR_z = scale(FArate),
         dPrime = HR_z - FAR_z)

d.coged.wm.partial$taskCode <- as.numeric(d.coged.wm.partial$taskCode)

#Multilevel model predicting SV from N-Back performance measures (e.g., hit rate, correct rejection rat
m.SV.wm <- brm(data = d.coged.wm.partial, meanSV ~ taskCode + (1 | subjectid),
                       file = "models/m.SV.wm.rds")
summary(m.SV.wm)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: meanSV ~ taskCode + (1 | subjectid)
##    Data: d.coged.wm.partial (Number of observations: 312)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Group-Level Effects:
## ~subjectid (Number of levels: 104)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.25      0.02     0.21     0.30 1.00     1380     2481
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.74      0.05     0.65     0.84 1.00     3086     2868
## taskCode     -0.09      0.01    -0.12    -0.06 1.00     9074     2656
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.20      0.01     0.18     0.22 1.00     3702     3294
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
bayes_R2(m.SV.wm)
```

```
##     Estimate  Est.Error      Q2.5     Q97.5
## R2 0.6400561 0.02839911 0.5788057 0.6887164
```

```r
m.SV.wm.partial <- brm(data = d.coged.wm.partial, meanSV ~ taskCode + hitrate + CRrate + meanRT + (1 | s
                       file = "models/m.SV.wm.partial.rds")
summary(m.SV.wm.partial)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: meanSV ~ taskCode + hitrate + CRrate + meanRT + (1 | subjectid)
##    Data: d.coged.wm.partial (Number of observations: 312)
```

```
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Group-Level Effects:
## ~subjectid (Number of levels: 104)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.25      0.02     0.21     0.29 1.00     1203     2102
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.72      0.13     0.48     0.98 1.00     1881     2854
## taskCode     -0.07      0.01    -0.10    -0.04 1.00     4999     3043
## hitrate       0.20      0.06     0.09     0.31 1.00     2967     3160
## CRrate       -0.05      0.09    -0.22     0.13 1.00     1978     2787
## meanRT       -0.00      0.00    -0.00     0.00 1.00     2586     3161
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.19      0.01     0.18     0.21 1.00     3026     2919
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
bayes_R2(m.SV.wm.partial)
```

```
##     Estimate  Est.Error      Q2.5     Q97.5
## R2 0.6541862 0.02766197 0.5952982 0.7021746
```

```
#create data frame with residuals from multilevel model
subj.resid.wm <- m.SV.wm.partial[["data"]][["subjectid"]] %>% as_tibble()
colnames(subj.resid.wm) <- "subjectid"
res.WM <- residuals(m.SV.wm.partial) %>% as_tibble() %>% select(Estimate)
colnames(res.WM) <- "resid.wm"
res.subj.WM <- cbind(subj.resid.wm, res.WM)

#Speech Cog-ED
#Summarize average performance on speech-in-noise task
d.coged.speech.clean <- d.coged.speech %>% select(subjectid, SV_red, SV_blue, SV_purple) %>%
  pivot_longer(names_to = "tmp", values_to = "SV", -subjectid) %>%
   separate(col = tmp, into=c(NA,"task"), sep = "_")

d.coged.speech.partial <- d.coged.speech.clean %>% group_by(subjectid,task) %>% summarise(meanSV = mean
  inner_join(performance.speech, by = c("subjectid","task")) %>%
  mutate(taskCode = factor(task, levels =c("red","blue","purple"), labels = c(-1,0,1)))
d.coged.speech.partial$taskCode <- as.numeric(d.coged.speech.partial$taskCode)

#Multilevel model predicting SV from speech-in-noise task performance measures (e.g., intelligibility)
m.SV.speech <- brm(data = d.coged.speech.partial, meanSV ~ taskCode + (1 | subjectid),
                           file = "models/m.SV.speech.rds")
summary(m.SV.speech)
```

```
##  Family: gaussian
```

```
##   Links: mu = identity; sigma = identity
## Formula: meanSV ~ taskCode + (1 | subjectid)
##    Data: d.coged.speech.partial (Number of observations: 312)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Group-Level Effects:
## ~subjectid (Number of levels: 104)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.29      0.03     0.24     0.34 1.00     1313     2186
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.77      0.04     0.69     0.86 1.00     2744     3152
## taskCode     -0.19      0.02    -0.22    -0.16 1.00     9570     2560
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.24      0.01     0.21     0.26 1.00     3197     3211
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
bayes_R2(m.SV.speech)
```

```
##    Estimate Est.Error      Q2.5     Q97.5
## R2 0.652492 0.0279619 0.5912372 0.7009902
```

```
m.SV.speech.partial <- brm(data = d.coged.speech.partial, meanSV ~ taskCode + performance + (1 | subjec
                           file = "models/m.SV.speech.partial.rds")
summary(m.SV.speech.partial)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: meanSV ~ taskCode + performance + (1 | subjectid)
##    Data: d.coged.speech.partial (Number of observations: 312)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Group-Level Effects:
## ~subjectid (Number of levels: 104)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.29      0.03     0.24     0.34 1.00     1063     2359
##
## Population-Level Effects:
##             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept       0.53      0.13     0.27     0.79 1.00     2137     2726
## taskCode       -0.12      0.04    -0.19    -0.04 1.00     2455     2866
## performance     0.00      0.00    -0.00     0.00 1.00     2219     3009
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

```
## sigma     0.23      0.01     0.21     0.26 1.00     2818     2923
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

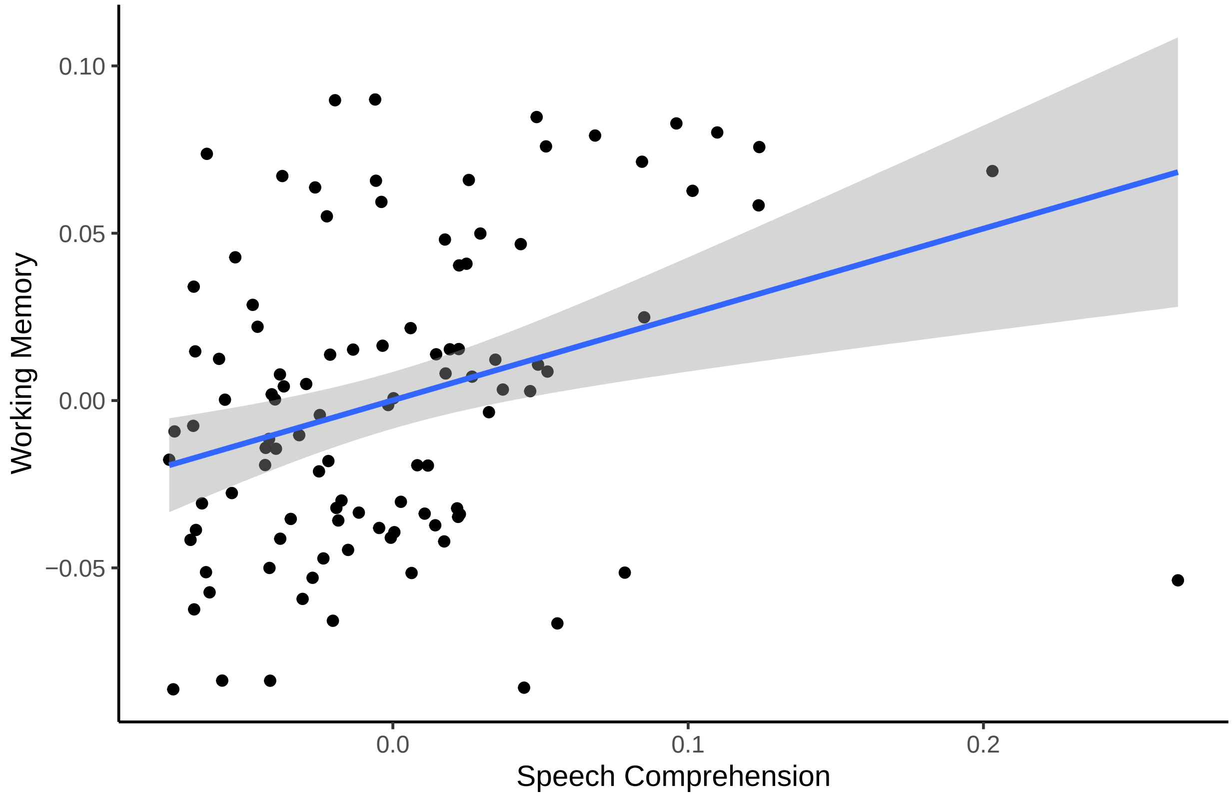
```
bayes_R2(m.SV.speech.partial)
```

```
##     Estimate  Est.Error      Q2.5     Q97.5
## R2 0.6584696 0.02746014 0.5987702 0.7063963
```

```
#create data frame with residuals from multilevel model
subj.resid.speech <- m.SV.speech.partial[["data"]][["subjectid"]] %>% as_tibble()
colnames(subj.resid.speech) <- "subjectid"
res.speech <-residuals(m.SV.speech.partial) %>% as_tibble() %>% select(Estimate)
colnames(res.speech) <- "resid.speech"
res.subj.Speech <- cbind(subj.resid.speech, res.speech)

SV.resids <- cbind(res.subj.WM, res.speech) %>% group_by(subjectid) %>%
  summarise(mean.resid.wm = mean(resid.wm), mean.resid.speech = mean(resid.speech))
#Testing for correlation between cognitive effort discounting across working memory & speech domains
#Controlling for task performance
CogED.cor.partial <- cor_test(data = SV.resids, x = "mean.resid.wm", y = "mean.resid.speech",
                              bayesian = TRUE, bayesian_prior = 0.707107)
#Summarize Bayes Factor from correlation controlling for task performance
CogED.cor.partial
```

```
## Parameter1    |        Parameter2 |  rho |       95% CI |        pd | % in ROPE |                Prio
## ------------------------------------------------------------------------------------------------------
## mean.resid.wm | mean.resid.speech | 0.32 | [0.17, 0.45] | 99.98%*** |     1.10% | Beta (1.41 +- 1.41
##
## Observations: 104
```

```
#Plot of correlation between working memory & speech comprehension domains controlling for task level &
#Not included in manuscript
fig.resid <- ggplot(SV.resids, aes(mean.resid.speech, mean.resid.wm)) +
   theme(plot.title = element_text(hjust = 0.5), panel.grid.major = element_blank(),
         panel.grid.minor = element_blank(), panel.background = element_blank(),
         axis.line = element_line(colour = "black")) +
  geom_point() + geom_smooth(method=lm) +ggtitle("") +
    xlab("Speech Comprehension") + ylab("Working Memory")
fig.resid
```



Stage III of Analysis: Partial Correlation

Reward Motivation (BIS/BAS, SPSRQ, GRAPES)

```
#BIS/BAS
#Importing and cleaning BIS/BAS
BISBAS.clean <- BISBAS %>% select(c(subjectid,completed, BAS_Drive, BAS_Fun, BAS_Reward, BIS)) %>%
  distinct(subjectid, .keep_all=T) %>% mutate(BAS_total = BAS_Drive + BAS_Fun + BAS_Reward)
BISBAS.SV <- inner_join(average.SV, BISBAS.clean)

#SPSRQ
#Importing and cleaning SPSRQ
SPSRQ.clean <- SPSRQ %>% select(c(subjectid, completed, SensitivityToReward, SensitivityToPunishment)) %
  distinct(subjectid, .keep_all=T)
SPSRQ.SV <- inner_join(average.SV, SPSRQ.clean)

#GRAPES
#Importing and cleaning GRAPES
GRAPES.clean <- GRAPES %>% select(c(subjectid, ends_with("response"))) %>% distinct(subjectid, .keep_al
  mutate(GRAPES.rew = q1_response + q4_response + q6_response + q7_response + q9_response + q10_respons
         q16_response + q16_response + q19_response + q20_response + q21_response + q25_response + q26_
         GRAPES.pun = q2_response + q3_response + q5_response + q8_response + q11_response + q12_respons
         q14_response + q18_response + q22_response + q23_response + q24_response + q28_response + q29_
GRAPES.SV <- inner_join(average.SV, GRAPES.clean)
```

Working Memory Capacity (Listing Span, Operation Span, Symmetry Span)

```
#Importing and cleaning WMC measures
#Listening Span
LSpan.clean <- LSpan %>% select(c(subjectid, completed, ListeningSpanScore)) %>%
  distinct(subjectid, .keep_all=T)
LSpan.SV <- inner_join(average.SV, LSpan.clean)

#Operation Span
OSpan.clean <- OSpan %>% select(c(subjectid, completed, ospan)) %>%
  distinct(subjectid, .keep_all=T)
OSpan.SV <- inner_join(average.SV, OSpan.clean)

#Symmetry Span
SymSpan.clean <- SymmSpan %>% select(c(subjectid, completed, sspan)) %>%
  distinct(subjectid, .keep_all=T)
SymSpan.SV <- inner_join(average.SV, SymSpan.clean)
```

Partial correlation controlling for WMC and reward sensitivity (from residualized SV estimates)

```
#Create data frame with WMC measures, get z-scores, and create a composite measure
LSpan.comp <- LSpan.SV %>% select(-c(Speech, WM))
LSpan.comp$LSpan.z <- scale(LSpan.comp$ListeningSpanScore)
OSpan.comp <- OSpan.SV %>% select(-c(Speech, WM))
OSpan.comp$OSpan.z <- scale(OSpan.comp$ospan)
SymSpan.comp <- SymSpan.SV %>% select(-c(Speech, WM))
SymSpan.comp$SSpan.z <- scale(SymSpan.comp$sspan)

WMC <- inner_join(LSpan.comp, OSpan.comp, by = "subjectid") %>%
  inner_join(SymSpan.comp, by = "subjectid") %>%
  filter(completed.x == 1) %>% filter(completed.y == 1) %>% filter(completed == 1) %>%
  select(-c("completed.x","completed.y","completed")) %>%
  mutate(WMC.composite = (LSpan.z + OSpan.z + SSpan.z)) %>%
  select(subjectid, WMC.composite)

#Correlating the working memory composite with subjective value in WM domain
WMC.composite.SV <- inner_join(average.SV, WMC, by = "subjectid")
WMC.WMSV.cor <- cor_test(data = WMC.composite.SV, x = "WM", y = "WMC.composite", bayesian = TRUE,
                         partial_bayesian = TRUE, bayesian_prior  = 0.707107)
WMC.WMSV.cor
```

```
## Parameter1 |    Parameter2 |  rho |        95% CI |     pd | % in ROPE |               Prior |    BF
## ----------------------------------------------------------------------------------------------------
## WM         | WMC.composite | 0.13 | [-0.03, 0.27] | 89.90% |    38.55% | Beta (1.41 +- 1.41) | 0.347
##
## Observations: 104
```

```
#Correlating the reward composite with subjective value in Speech domain
WMC.SpeechSV.cor <- cor_test(data = WMC.composite.SV, x = "Speech", y = "WMC.composite", bayesian = TRUE
```

```
                          partial_bayesian = TRUE, bayesian_prior  = 0.707107)
WMC.SpeechSV.cor
```

```
## Parameter1 |    Parameter2 |   rho |        95% CI |     pd | % in ROPE |                Prior |    BI
## ------------------------------------------------------------------------------------------------------
## Speech     | WMC.composite | -0.14 | [-0.29, 0.01] | 93.12% |    32.57% | Beta (1.41 +- 1.41) | 0.45
##
## Observations: 104
```

```
#Create data frame with reward sensitivity measures, get z-scores, and create a composite measure
BISBAS.comp <- BISBAS.SV %>% select(-c(Speech, WM, BAS_Drive, BAS_Fun, BAS_Reward, BIS))
BISBAS.comp$BAS.z <- scale(BISBAS.comp$BAS_total)
SPSRQ.comp <- SPSRQ.SV %>% select(-c(Speech, WM, SensitivityToPunishment))
SPSRQ.comp$SPSRQ.rew.z <- scale(SPSRQ.comp$SensitivityToReward)
GRAPES.comp <- GRAPES.SV %>% select(c(subjectid, GRAPES.rew))
GRAPES.comp$GRAPES.rew.z <- scale(GRAPES.comp$GRAPES.rew)

Reward.composite <- inner_join(BISBAS.comp, SPSRQ.comp, by = "subjectid") %>%
  inner_join(GRAPES.comp, by = "subjectid") %>%
  filter(completed.x == 1) %>% filter(completed.y == 1) %>%
  select(-c("completed.x","completed.y")) %>%
  mutate(Rew.composite = (BAS.z + SPSRQ.rew.z + GRAPES.rew.z)) %>%
  select(subjectid, Rew.composite)

#Correlating the reward composite with subjective value in WM domain
Reward.composite.SV <- inner_join(average.SV, Reward.composite, by = "subjectid")
Reward.WMSV.cor <- cor_test(data = Reward.composite.SV, x = "WM", y = "Rew.composite", bayesian = TRUE,
                          partial_bayesian = TRUE, bayesian_prior  = 0.707107)
Reward.WMSV.cor
```

```
## Parameter1 |    Parameter2 |  rho |        95% CI |     pd | % in ROPE |                Prior |    BF
## ------------------------------------------------------------------------------------------------------
## WM         | Rew.composite | 0.13 | [-0.02, 0.29] | 90.67% |    36.30% | Beta (1.41 +- 1.41) | 0.365
##
## Observations: 104
```

```
#Correlating the reward composite with subjective value in Speech domain
Reward.SpeechSV.cor <- cor_test(data = Reward.composite.SV, x = "Speech", y = "Rew.composite", bayesian
                          partial_bayesian = TRUE, bayesian_prior  = 0.707107)
Reward.SpeechSV.cor
```

```
## Parameter1 |    Parameter2 |   rho |        95% CI |     pd | % in ROPE |                Prior |    BI
## ------------------------------------------------------------------------------------------------------
## Speech     | Rew.composite | -0.09 | [-0.24, 0.06] | 82.67% |    50.70% | Beta (1.41 +- 1.41) | 0.23
##
## Observations: 104
```

```
#Testing for partial correlation between residuals (from stage two) controlling for WMC and reward sens
SV.composite.resid <- inner_join(WMC, Reward.composite, by = "subjectid") %>%
  inner_join(SV.resids, by = "subjectid")
SV.composite.resid.clean <- cbind(SV.composite.resid$WMC.composite, SV.composite.resid$Rew.composite,
```

```
                                     SV.composite.resid$mean.resid.speech, SV.composite.resid$mean.resid.wm
colnames(SV.composite.resid.clean) <- c("WMC", "Reward", "Speech", "WM")

WMC.resid.cor <- cor_test(data = SV.composite.resid.clean, x = "WM", y = "Speech", bayesian = TRUE,
                           partial_bayesian = TRUE, bayesian_prior  = 0.707107)
#Summarize Bayes Factor from correlation
WMC.resid.cor
```

```
## Parameter1 | Parameter2 |  rho |       95% CI |       pd | % in ROPE |               Prior |       BF
## -----------------------------------------------------------------------------------------------------
## WM         |     Speech | 0.34 | [0.19, 0.47] | 99.90%** |     0.95% | Beta (1.41 +- 1.41) | 99.76***
##
## Observations: 104
```

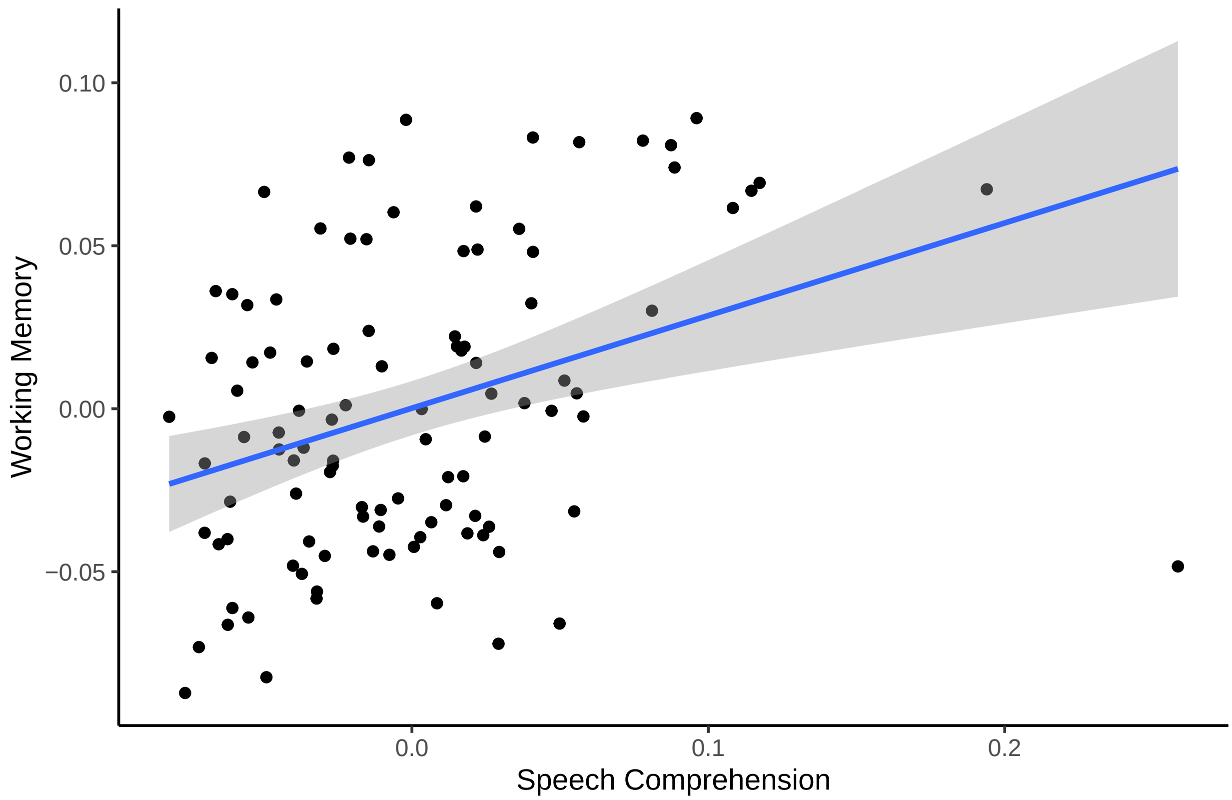
```
#Correlating WMC and Reward Sensitivity composites
d.composites <- inner_join(WMC, Reward.composite, by = "subjectid")
WMC.Reward.cor <- cor_test(data = d.composites, x = "WMC.composite", y = "Rew.composite", bayesian = TRU
                           partial_bayesian = TRUE, bayesian_prior  = 0.707107)
WMC.Reward.cor
```

```
## Parameter1    |    Parameter2 |   rho |        95% CI |     pd | % in ROPE |               Prior |
## -----------------------------------------------------------------------------------------------------
## WMC.composite | Rew.composite | -0.14 | [-0.29, 0.02] | 91.47% |    34.90% | Beta (1.41 +- 1.41) | 0
##
## Observations: 104
```

Figure 2; correlation between working memory & speech comprehension domains controlling for task level, performance, WMC, and reward sensitivity

```
#WM Cog-ED
d.coged.wm.partial.test <- inner_join(d.coged.wm.partial, WMC, by = "subjectid") %>% inner_join(Reward.c
m.SV.wm.partial.test <- brm(data = d.coged.wm.partial.test, meanSV ~ taskCode + hitrate + CRrate + meanR

subj.resid.wm.test <- m.SV.wm.partial.test[["data"]][["subjectid"]] %>% as_tibble()
colnames(subj.resid.wm.test) <- "subjectid"
res.WM.test <- resid(m.SV.wm.partial.test) %>% as_tibble() %>% select(Estimate)
colnames(res.WM.test) <- "resid.wm"
res.subj.WM.test <- cbind(subj.resid.wm.test, res.WM.test)

#Speech Cog-ED
d.coged.speech.partial.test <- inner_join(d.coged.speech.partial, WMC, by = "subjectid") %>% inner_join
m.SV.speech.partial.test <- brm(data = d.coged.speech.partial.test, meanSV ~ taskCode + performance + WM
                                file = "models/m.SV.speech.partial.test.rds")

subj.resid.speech.test <- m.SV.speech.partial.test[["data"]][["subjectid"]] %>% as_tibble()
colnames(subj.resid.speech.test) <- "subjectid"
res.speech.test <-residuals(m.SV.speech.partial.test) %>% as_tibble() %>% select(Estimate)
colnames(res.speech.test) <- "resid.speech"
res.subj.Speech.test <- cbind(subj.resid.speech.test, res.speech.test)

SV.resids.test <- cbind(res.subj.WM.test, res.speech.test) %>% group_by(subjectid) %>% summarise(mean.re
```

```r
#Testing for correlation between cognitive effort discounting across working memory & speech domains (c
CogED.cor.partial.test <- cor_test(data = SV.resids.test, x = "mean.resid.wm", y = "mean.resid.speech",

#Plot of correlation between working memory & speech comprehension domains controlling for task level, j
##Figure 2
fig.resid.2 <- ggplot(SV.resids.test, aes(mean.resid.speech, mean.resid.wm)) +
   theme(plot.title = element_text(hjust = 0.5), panel.grid.major = element_blank(), panel.grid.minor =
  geom_point() + geom_smooth(method=lm) +ggtitle("") +
    xlab("Speech Comprehension") + ylab("Working Memory")
fig.resid.2
```



Exploratory Analyses

Need for Cognition

Relationship between average subjective value estimates and NCS scores

```r
#Importing and cleaning NCS
NCS.clean <- NCS %>% select(c(subjectid, completed, totalscore)) %>%
              distinct(subjectid, .keep_all=T)

NCS.SV <- inner_join(average.SV, NCS.clean)
NCS.SV$WM.z <- scale(NCS.SV$WM)
```

```r
NCS.SV$Speech.z <- scale(NCS.SV$Speech)
NCS.SV$NCS.comp <- NCS.SV$WM.z + NCS.SV$Speech.z

#Testing for the correlation between NCS and SV estimates from both WM and Speech comprehension domains
NCS.wm.cor <- cor_test(data = NCS.SV, x = "WM", y = "totalscore",
                       bayesian = TRUE, bayesian_prior = 0.707107)
NCS.wm.cor
```

```
## Parameter1 | Parameter2 |       rho |        95% CI |     pd | % in ROPE |               Prior |   B
## -----------------------------------------------------------------------------------------------------
## WM         | totalscore | -1.29e-04 | [-0.16, 0.15] | 50.08% |    69.55% | Beta (1.41 +- 1.41) | 0.15
##
## Observations: 104
```

```r
NCS.speech.cor <- cor_test(data = NCS.SV, x = "Speech", y = "totalscore",
                           bayesian = TRUE, bayesian_prior = 0.707107)
NCS.speech.cor
```

```
## Parameter1 | Parameter2 |  rho |        95% CI |     pd | % in ROPE |               Prior |    BF
## -------------------------------------------------------------------------------------------------
## Speech     | totalscore | 0.13 | [-0.03, 0.27] | 90.08% |    37.10% | Beta (1.41 +- 1.41) | 0.364
##
## Observations: 104
```

```r
NCS.comp.cor <- cor_test(data = NCS.SV, x = "NCS.comp", y = "totalscore",
                         bayesian = TRUE, bayesian_prior = 0.707107)
NCS.comp.cor
```

```
## Parameter1 | Parameter2 |  rho |        95% CI |     pd | % in ROPE |               Prior |    BF
## -------------------------------------------------------------------------------------------------
## NCS.comp   | totalscore | 0.08 | [-0.07, 0.23] | 79.00% |    55.62% | Beta (1.41 +- 1.41) | 0.211
##
## Observations: 104
```

```r
#Testing for the correlation between NCS and SV estimates from both WM and Speech comprehension domains
NCS.SV.resid <- inner_join(SV.resids.test, NCS.clean)
NCS.wm.resid.cor <- cor_test(data = NCS.SV.resid, x = "mean.resid.wm", y = "totalscore",
                             bayesian = TRUE, bayesian_prior = 0.707107)
NCS.wm.resid.cor
```

```
## Parameter1    | Parameter2 |   rho |        95% CI |     pd | % in ROPE |               Prior |    B
## -----------------------------------------------------------------------------------------------------
## mean.resid.wm | totalscore | -0.03 | [-0.19, 0.12] | 63.08% |    68.05% | Beta (1.41 +- 1.41) | 0.15
##
## Observations: 104
```

```r
NCS.speech.resid.cor <- cor_test(data = NCS.SV.resid, x = "mean.resid.speech", y = "totalscore",
                                 bayesian = TRUE, bayesian_prior = 0.707107)
NCS.speech.resid.cor
```

```
## Parameter1        | Parameter2 |  rho |        95% CI |     pd | % in ROPE |               Prior |
## ---------------------------------------------------------------------------------------------------
## mean.resid.speech | totalscore | 0.12 | [-0.04, 0.27] | 87.78% |    41.73% | Beta (1.41 +- 1.41) | 0
##
## Observations: 104
```