Adjs Conc-Abs: Graphs for Accuracy

morgan moyer

2025-03-26

```r
table(d$Task,d$Label)
```

```
## 
##             test_conc test_val
##   Concrete       4800        0
##   Valence           0     4800
```

```r
print(unique(d$Word))
```

```
##  [1] "rusted"    "mindful"   "freezing"  "disloyal"  "soulless"  "evil"     
##  [7] "starry"    "female"    "headless"  "hairy"     "crappy"    "lovable"  
## [13] "pleasant"  "bloody"    "pregnant"  "newborn"   "human"     "honest"   
## [19] "worthless" "possible"  "divine"    "dirty"     "toothless" "bloodshot"
## [25] "heartless" "thorny"    "handmade"  "lilac"     "thankful"  "blooming" 
## [31] "sunny"     "helpful"   "orange"    "splendid"  "awful"     "wrinkled" 
## [37] "stupid"    "sexist"    "sensible"  "sinful"
```

Summary Stats

```r
agr <- d %>%
  group_by(Task) %>%
  summarize(MeanAccuracy = mean(Accuracy),
            SD = sd(Accuracy))
print(agr)
```

```
## # A tibble: 2 x 3
##   Task     MeanAccuracy    SD
##   <chr>           <dbl> <dbl>
## 1 Concrete        0.81  0.392
## 2 Valence         0.940 0.237
```

```r
print(unique(d$Word))
```

```
##  [1] "rusted"    "mindful"   "freezing"  "disloyal"  "soulless"  "evil"     
##  [7] "starry"    "female"    "headless"  "hairy"     "crappy"    "lovable"  
## [13] "pleasant"  "bloody"    "pregnant"  "newborn"   "human"     "honest"   
## [19] "worthless" "possible"  "divine"    "dirty"     "toothless" "bloodshot"
## [25] "heartless" "thorny"    "handmade"  "lilac"     "thankful"  "blooming" 
## [31] "sunny"     "helpful"   "orange"    "splendid"  "awful"     "wrinkled" 
## [37] "stupid"    "sexist"    "sensible"  "sinful"
```

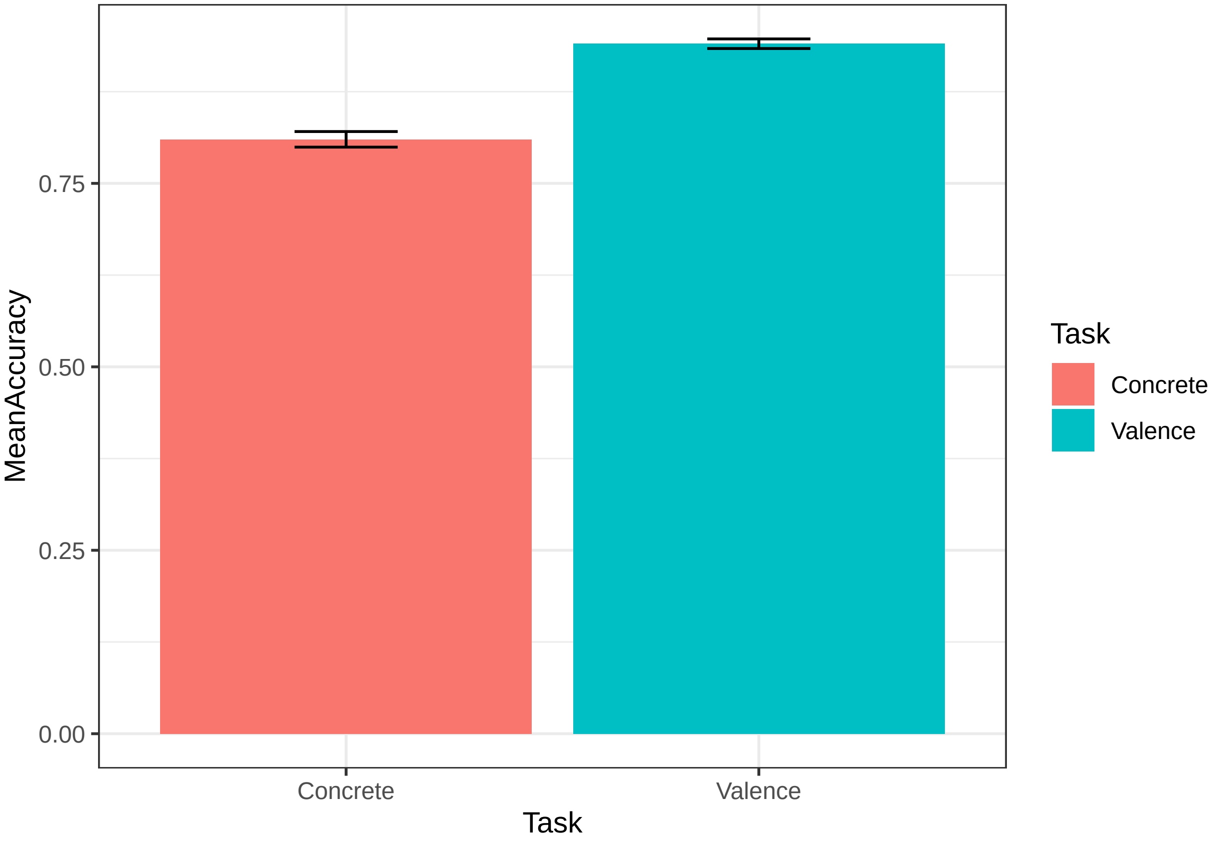
Graph Accuracy by Word

Values for valence/concreteness were gathered/normed first from Warriner et al and Brysbaert et al. From those studies, we can establish what an Accurate response is.

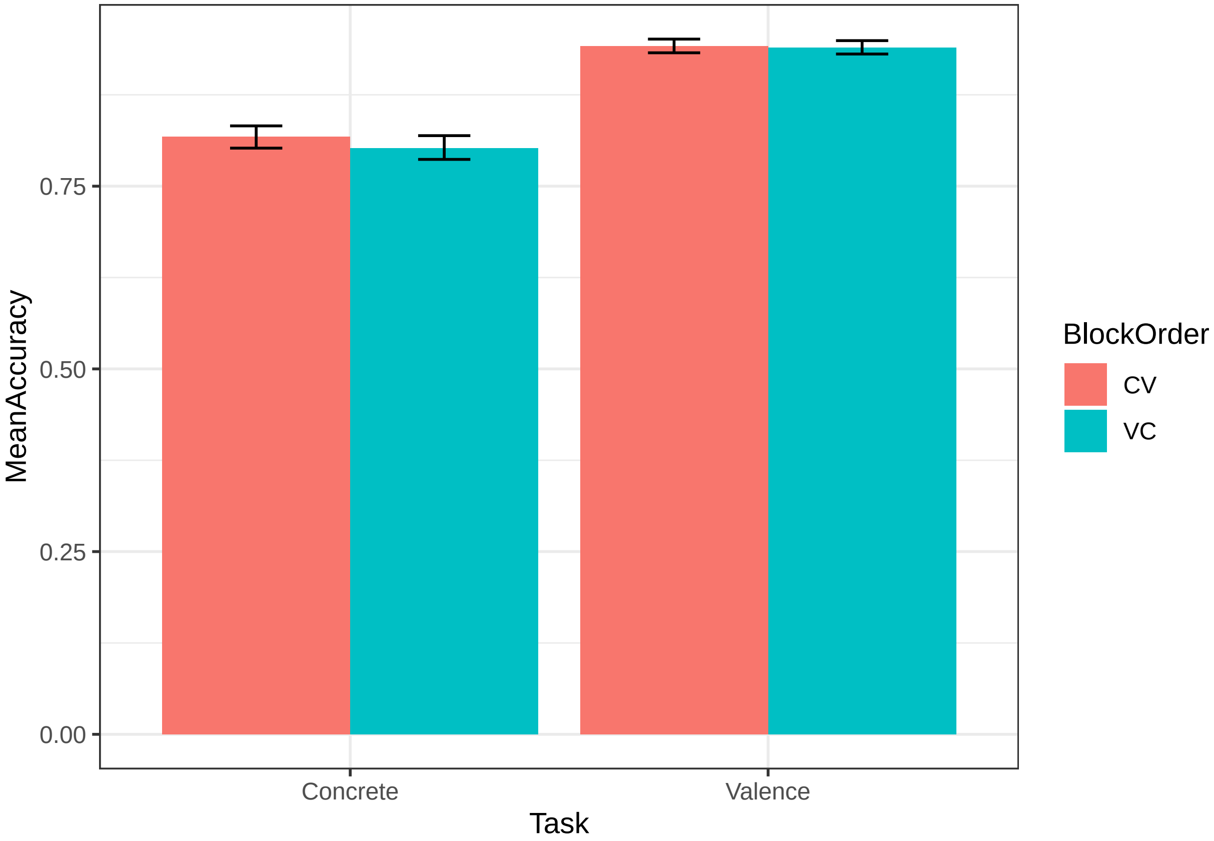
A response is accurate (coded as 1) if the participant response was consistent with the norming study; innacurate (or 0) otherwise.

Overall Accuracy

```
agr <- d %>%
  group_by(Task) %>%
  reframe(MeanAccuracy = mean(Accuracy),
          CILow = ci.low(Accuracy),
          CIHigh = ci.high(Accuracy)) %>%
  mutate(YMin = MeanAccuracy - CILow,
         YMax = MeanAccuracy + CIHigh)
# View(agr)

dodge = position_dodge(.9)
ggplot(data=agr, aes(x=Task,y=MeanAccuracy, fill=Task)) +
  geom_bar(position=dodge,stat="identity") +
  geom_errorbar(aes(ymin=YMin,ymax=YMax),width=.25,position=position_dodge(0.9))
```



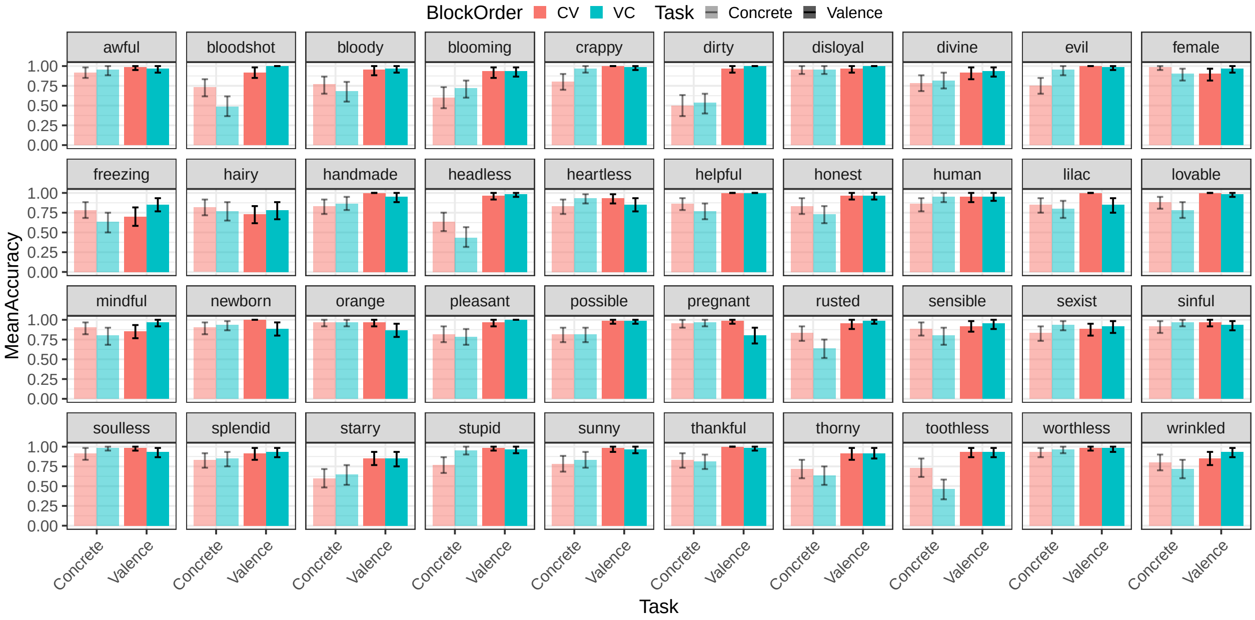
```
agr <- d %>%
  group_by(Task,BlockOrder) %>%
  reframe(MeanAccuracy = mean(Accuracy),
          CILow = ci.low(Accuracy),
          CIHigh = ci.high(Accuracy)) %>%
  mutate(YMin = MeanAccuracy - CILow,
         YMax = MeanAccuracy + CIHigh)
# View(agr)

dodge = position_dodge(.9)
ggplot(data=agr, aes(x=Task,y=MeanAccuracy, fill=BlockOrder)) +
  geom_bar(position=dodge,stat="identity") +
  geom_errorbar(aes(ymin=YMin,ymax=YMax),width=.25,position=position_dodge(0.9))
```



Mean Accuracy by Word / Task

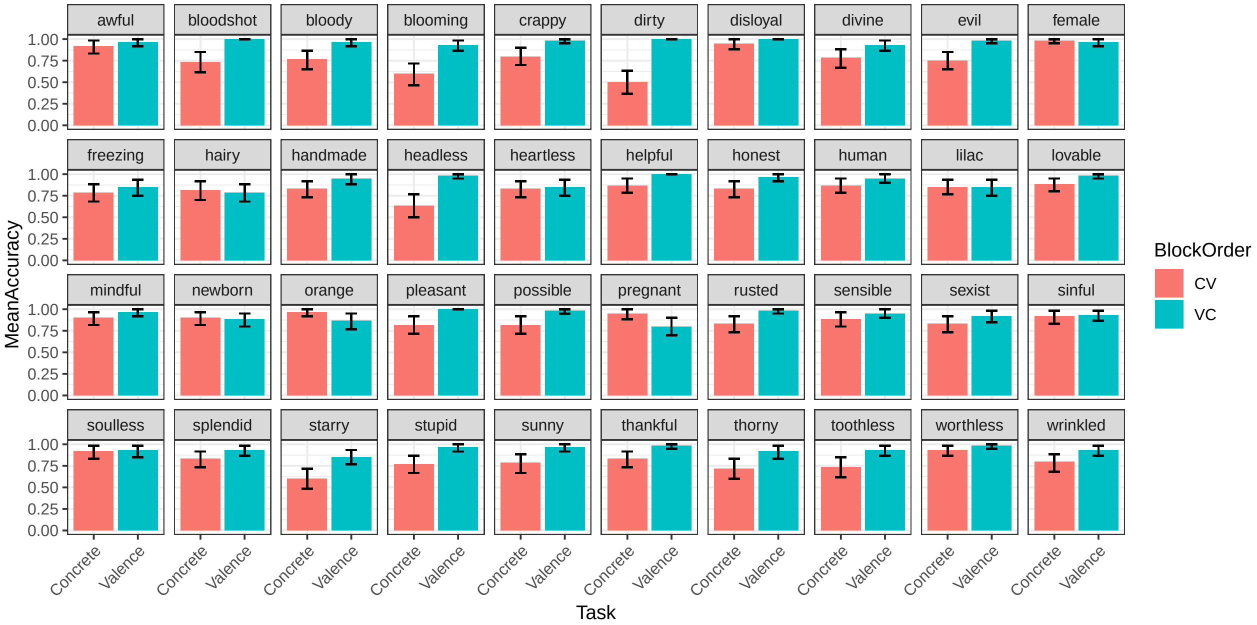
```
agr <- d %>%
  group_by(Word,Task,BlockOrder) %>%
  reframe(MeanAccuracy = mean(Accuracy),
          CILow = ci.low(Accuracy),
          CIHigh = ci.high(Accuracy)) %>%
  mutate(YMin = MeanAccuracy - CILow,
         YMax = MeanAccuracy + CIHigh)
# View(agr)

# dodge = position_dodge(.9)
# ggplot(data=agr, aes(x=Task,y=MeanAccuracy,fill=BlockOrder,alpha=Task)) +
#   geom_bar(position=dodge,stat="identity",alpha = 0.8) +
#   facet_wrap(~Word,ncol=10) +
#   geom_errorbar(aes(ymin=YMin,ymax=YMax),width=.25,position=position_dodge(0.9)) +
#   theme(axis.text.x = element_text(angle = 45, hjust = 1))

ggplot(agr,aes(x=Task, y=MeanAccuracy, alpha=Task, fill=BlockOrder)) +
  geom_bar(position="dodge",stat="identity") +
  geom_errorbar(aes(ymin=YMin,ymax=YMax),width=.25,position=position_dodge(0.9)) +
  facet_wrap(~Word, ncol=10) +
  xlab("Task") +
  ylab("MeanAccuracy") +
  # guides(fill=FALSE) +
  guides(alpha=guide_legend(title="Task")) +
  theme(legend.key.size = unit(0.3, "cm"),
        legend.position = "top", # c(.5,1)
        legend.direction = "horizontal",
        legend.margin=margin(0,0,0,0),
        legend.box.margin=margin(0,0,-5,-5),legend.spacing.y = unit(0.001, 'cm')) +
    # scale_fill_manual(values=cbPalette) +
    # scale_color_manual(values=cbPalette) +
    scale_alpha_discrete(range = c(.5,1)) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Using alpha for a discrete variable is not advised.
```



Looking at only the first block

```
agr <- d %>%
  group_by(Task,Word,BlockOrder) %>%
  filter((Task == "Valence") & (BlockOrder == "VC") |
           (Task == "Concrete") & (BlockOrder == "CV")) %>%
  mutate(MeanAccuracy = mean(Accuracy),
          CILow = ci.low(Accuracy),
          CIHigh = ci.high(Accuracy)) %>%
  mutate(YMin = MeanAccuracy - CILow,
         YMax = MeanAccuracy + CIHigh)

agrr <- agr %>%
  group_by(Word,Task) %>%
  select(Word,Task,MeanAccuracy) %>%
  unique()

dodge = position_dodge(.9)
ggplot(data=agr, aes(x=Task,y=MeanAccuracy,fill=BlockOrder)) +
  geom_bar(position=dodge,stat="identity") +
  facet_wrap(~Word,ncol=10) +
  geom_errorbar(aes(ymin=YMin,ymax=YMax),width=.25,position=position_dodge(0.9)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
m <- lmer(MeanAccuracy ~ BlockOrder + (1|Word), data =agr)
summary(m)
```

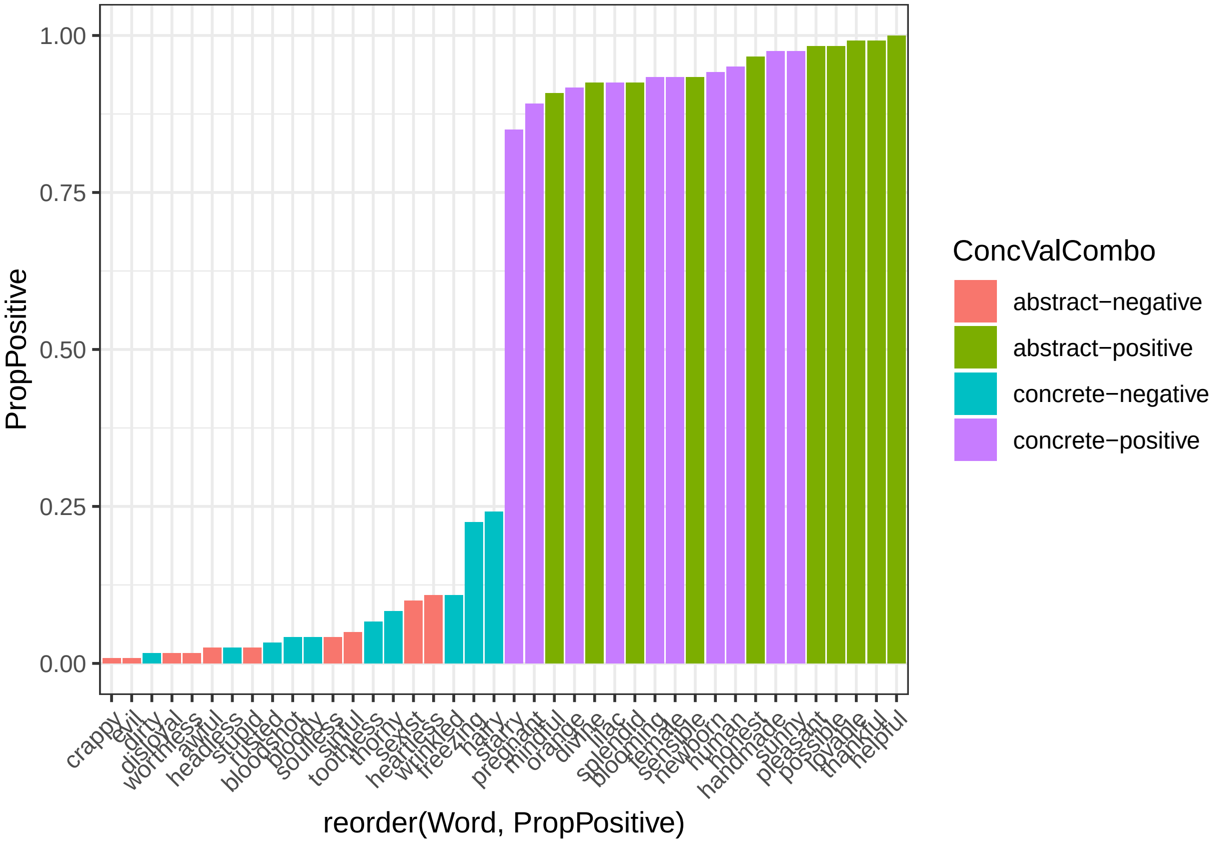
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: MeanAccuracy ~ BlockOrder + (1 | Word)
##    Data: agr
##
## REML criterion at convergence: -13009.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.11584 -0.57461  0.00202  0.59951  3.07556
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Word     (Intercept) 0.003224 0.05678
##  Residual             0.003734 0.06111
## Number of obs: 4800, groups:  Word, 40
##
## Fixed effects:
##              Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)  8.179e-01  9.064e-03 3.975e+01   90.24   <2e-16 ***
## BlockOrderVC 1.217e-01  1.764e-03 4.759e+03   68.97   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## BlockOrdrVC -0.097
```

PropPositive and PropConcrete

```
val <- d %>%
  filter(Task == "Valence") %>%
  # filter(Word %in% conc$Word) %>%
  group_by(Word,ConcValCombo) %>%
  mutate(Response.n = as.numeric(factor(Response, levels = c("negative", "positive"))) - 1) %>%  # Conv
  summarize(PropPositive = mean(Response.n))
```

```
## `summarise()` has grouped output by 'Word'. You can override using the
## `.groups` argument.
```

```
  # filter(PropPositive > .1 | PropPositive < .9)

dodge = position_dodge(.9)
ggplot(data=val, aes(x=reorder(Word,PropPositive),y=PropPositive,fill=ConcValCombo)) +
  geom_bar(position=dodge,stat="identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

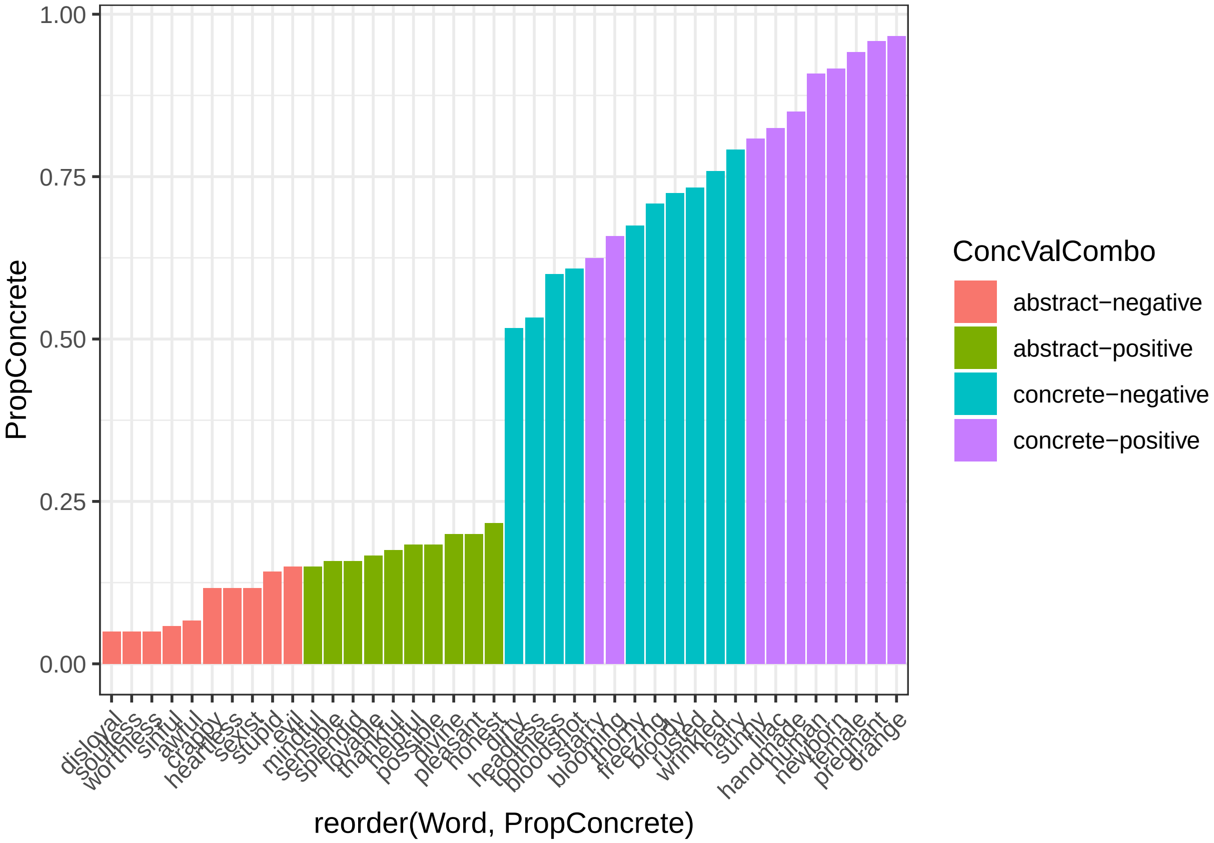


```
  # guides(fill = "none")
```

```
conc <- d %>%
  filter(Task == "Concrete") %>%
  # filter(Word %in% conc$Word) %>%
  group_by(Word,ConcValCombo) %>%
  mutate(Response.n = as.numeric(factor(Response, levels = c("abstract", "concrete"))) - 1) %>%  # Conv
  summarize(PropConcrete = mean(Response.n))
```

```
## `summarise()` has grouped output by 'Word'. You can override using the
## `.groups` argument.
```

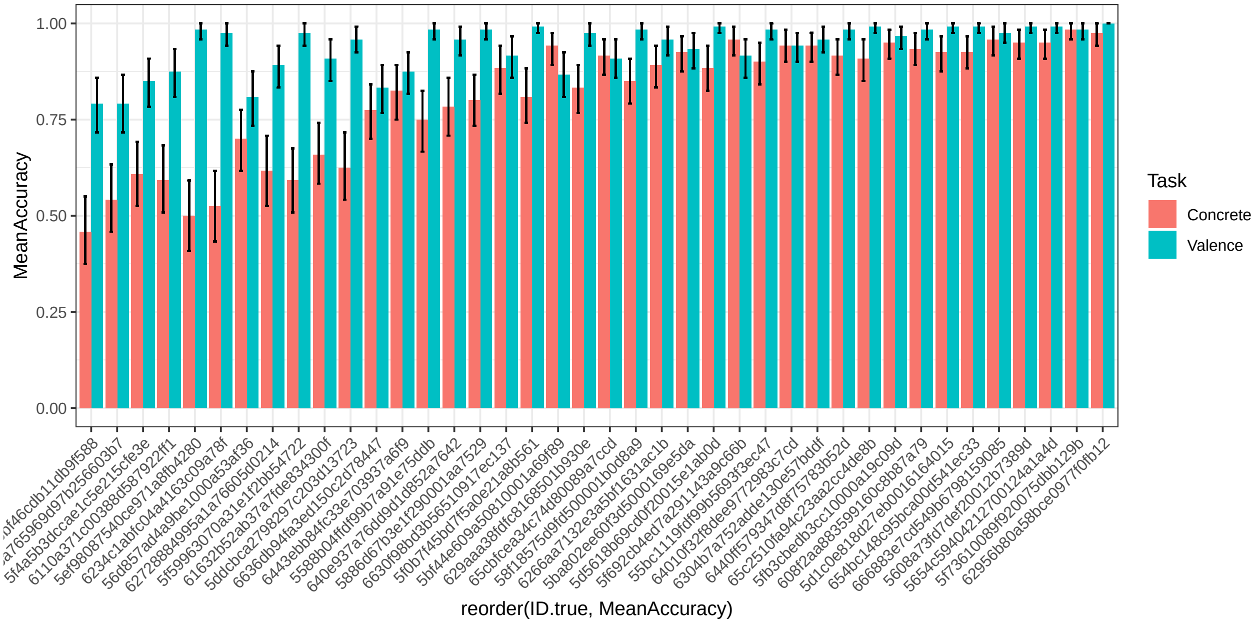
```
  # filter(PropPositive > .1 | PropPositive < .9)

dodge = position_dodge(.9)
ggplot(data=conc, aes(x=reorder(Word,PropConcrete),y=PropConcrete,fill=ConcValCombo)) +
  geom_bar(position=dodge,stat="identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
  # guides(fill = "none")
```

Accuracy by Participant

```
agr <- d %>%
  # filter(PennElementType == "Selector") %>%
  select(ID.true,Task,Accuracy) %>%
  group_by(ID.true,Task) %>%
  mutate(MeanAccuracy = mean(Accuracy),
         CILow = ci.low(Accuracy),
          CIHigh = ci.high(Accuracy)) %>%
  mutate(YMin = MeanAccuracy - CILow,
        YMax = MeanAccuracy + CIHigh)

dodge = position_dodge(.9)
ggplot(data=agr, aes(x=reorder(ID.true,MeanAccuracy),y=MeanAccuracy,fill=Task)) +
  geom_bar(position=dodge,stat="identity") +
  geom_errorbar(aes(ymin=YMin,ymax=YMax),width=.25,position=position_dodge(0.9)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



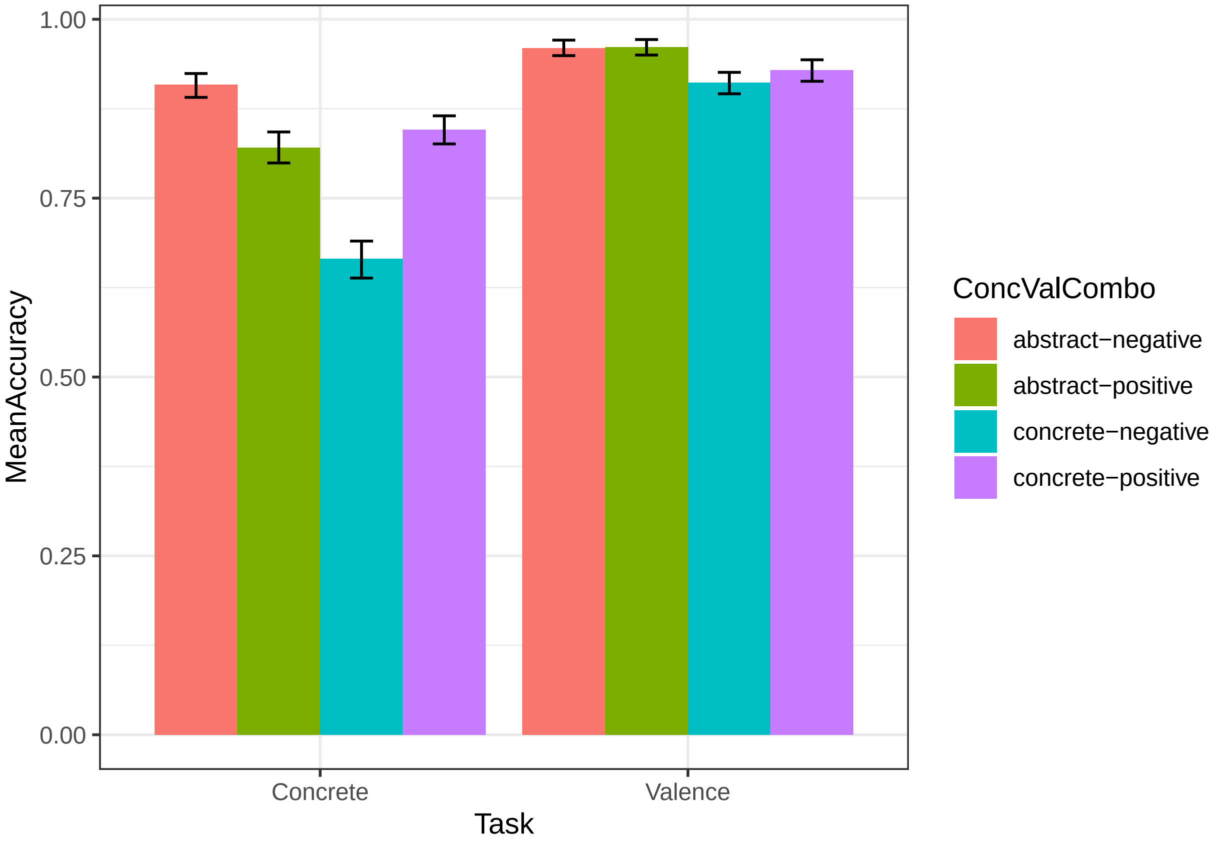
```
  # guides(fill = "none")
```

Mean Accuracy by ConcValCombo

```
agr <- d %>%
  group_by(Task,ConcValCombo) %>%
  summarize(MeanAccuracy = mean(Accuracy), CILow = ci.low(Accuracy), CIHigh = ci.high(Accuracy)) %>%
  mutate(YMin = MeanAccuracy - CILow, YMax = MeanAccuracy + CIHigh)
```

```
## `summarise()` has grouped output by 'Task'. You can override using the
## `.groups` argument.
```

```
dodge = position_dodge(.9)
ggplot(data=agr, aes(x=Task,y=MeanAccuracy,fill=ConcValCombo)) +
  geom_bar(position=dodge,stat="identity") +
  # facet_wrap(~Task) +
  geom_errorbar(aes(ymin=YMin,ymax=YMax),width=.25,position=position_dodge(0.9))
```



```
  # theme(axis.text.x = element_text(angle = 45, hjust = 1))
  # guides(fill = "none")
```

Remove participants who aren't accurate

```
length(unique(d$ID.true))
```

```
## [1] 40
```

```
inacc.parts <- d %>%
  group_by(ID.true,Task) %>%
  summarise(MeanAccuracy = mean(Accuracy)) %>%
  filter(MeanAccuracy < .75)
```

```
## `summarise()` has grouped output by 'ID.true'. You can override using the
## `.groups` argument.
```

```
# How many participants have Accuracy < .75?
length(unique(inacc.parts$ID.true))
```

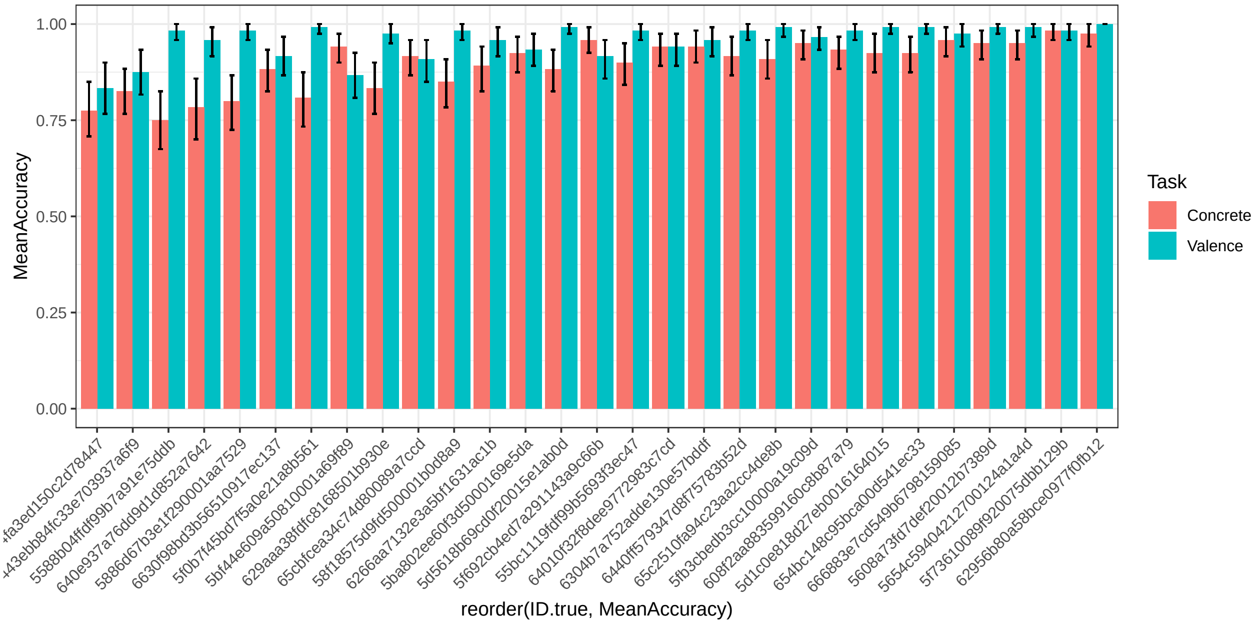
```
## [1] 11
```

```
d.inaccurate.removed <- d %>%
  anti_join(inacc.parts, by = "ID.true")

# Sanity check
length(unique(d.inaccurate.removed$ID.true))
```

```
## [1] 29
```

Accuracy by Participant

```
agr <- d.inaccurate.removed %>%
  # filter(PennElementType == "Selector") %>%
  select(ID.true,Task,Accuracy) %>%
  group_by(ID.true,Task) %>%
  mutate(MeanAccuracy = mean(Accuracy),
         CILow = ci.low(Accuracy),
         CIHigh = ci.high(Accuracy)) %>%
  mutate(YMin = MeanAccuracy - CILow,
        YMax = MeanAccuracy + CIHigh)

dodge = position_dodge(.9)
ggplot(data=agr, aes(x=reorder(ID.true,MeanAccuracy),y=MeanAccuracy,fill=Task)) +
  geom_bar(position=dodge,stat="identity") +
  geom_errorbar(aes(ymin=YMin,ymax=YMax),width=.25,position=position_dodge(0.9)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

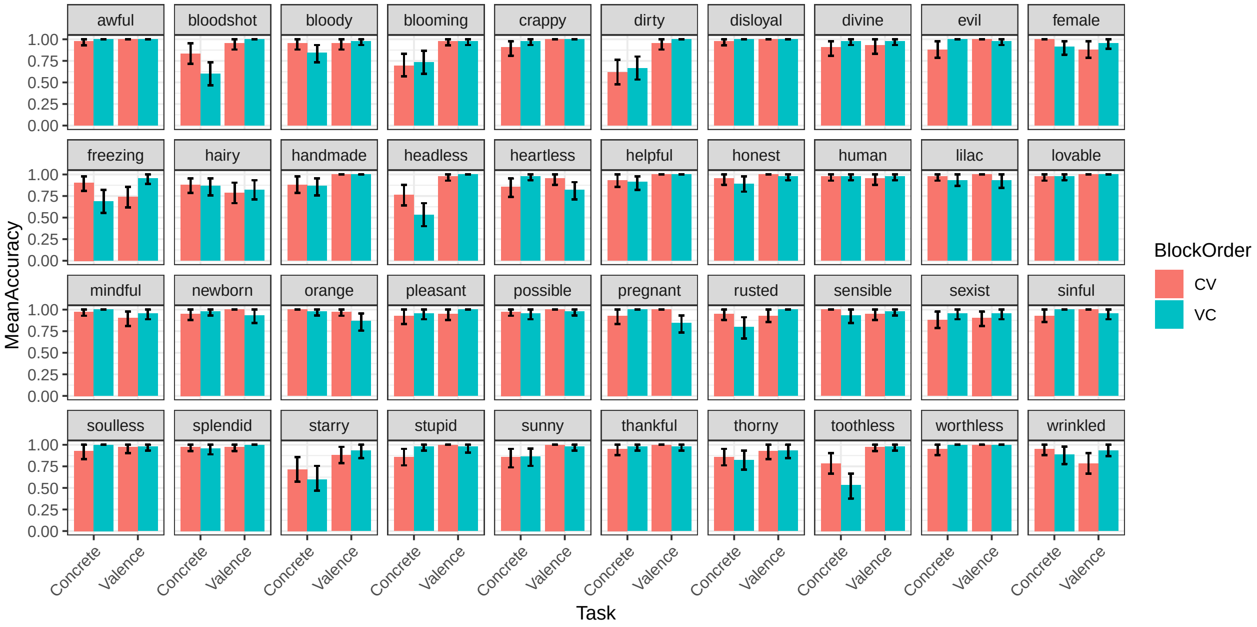


```
  # guides(fill = "none")
```

Mean Accuracy by Word / Task

Looking at only the first block

```
agr <- d.inaccurate.removed %>%
  group_by(Task,Word,BlockOrder) %>%
  # filter((Task == "Valence") & (BlockOrder == "VC") |
  #          (Task == "Concrete") & (BlockOrder == "CV")) %>%
  mutate(MeanAccuracy = mean(Accuracy),
         CILow = ci.low(Accuracy),
         CIHigh = ci.high(Accuracy)) %>%
  mutate(YMin = MeanAccuracy - CILow,
         YMax = MeanAccuracy + CIHigh)

agrr <- agr %>%
  group_by(Word,Task) %>%
  select(Word,Task,MeanAccuracy) %>%
  unique()

dodge = position_dodge(.9)
ggplot(data=agr, aes(x=Task,y=MeanAccuracy,fill=BlockOrder)) +
  geom_bar(position=dodge,stat="identity") +
  facet_wrap(~Word,ncol=10) +
  geom_errorbar(aes(ymin=YMin,ymax=YMax),width=.25,position=position_dodge(0.9)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


awful
bloodshot
bloody
blooming
crappy
dirty
disloyal
divine
evil
female
freezing
hairy
handmade
headless
heartless
helpful
honest
human
lilac
lovable
mindful
newborn
orange
pleasant
possible
pregnant
rusted
sensible
sexist
sinful
soulless
splendid
starry
stupid
sunny
thankful
thorny
toothless
worthless
wrinkled
MeanAccuracy
Concrete
Valence
Task
BlockOrder
CV
VC