Adjs Soc-Phys Weighted/Normed: Analysis

morgan moyer

2025-05-28

What's going on with those parts only in Valence task?

```
table(d$Group, d$Response)
```

```
## < table of extent 0 x 4 >
```

```
# Step 1: Summarize per ID.Ibex
ids_valence_only <- d %>%
  group_by(ID.true) %>%
  summarise(
    has_valence = any(Task == "Valence"),
    has_socphys = any(Task == "SocPhys"),
    .groups = "drop"
  ) %>%
  filter(has_valence & !has_socphys) %>%
  pull(ID.true)


d_bad <- d %>%
  filter(ID.true %in% ids_valence_only)

print(unique(d_bad$Response))
```

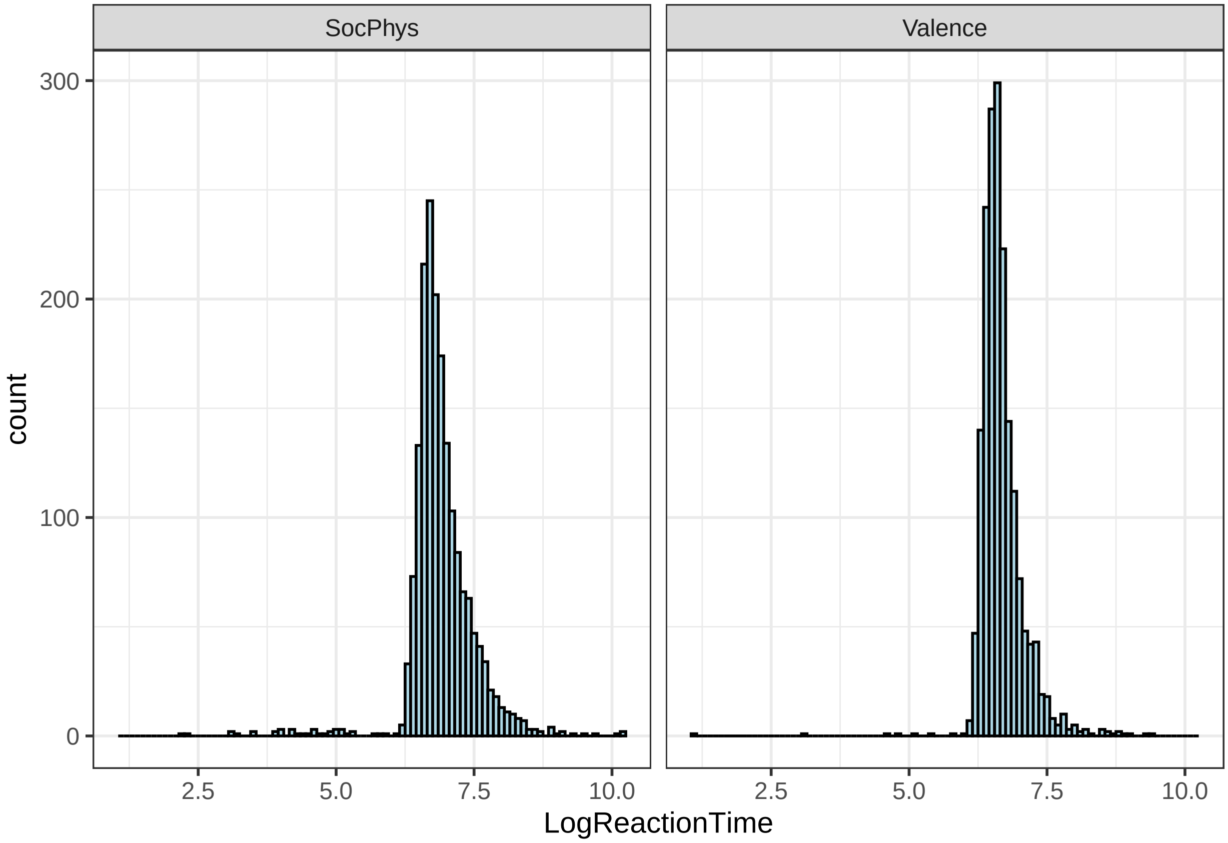
```
## [1] "negative" "positive"
```

```
print(table(d$Response))
```

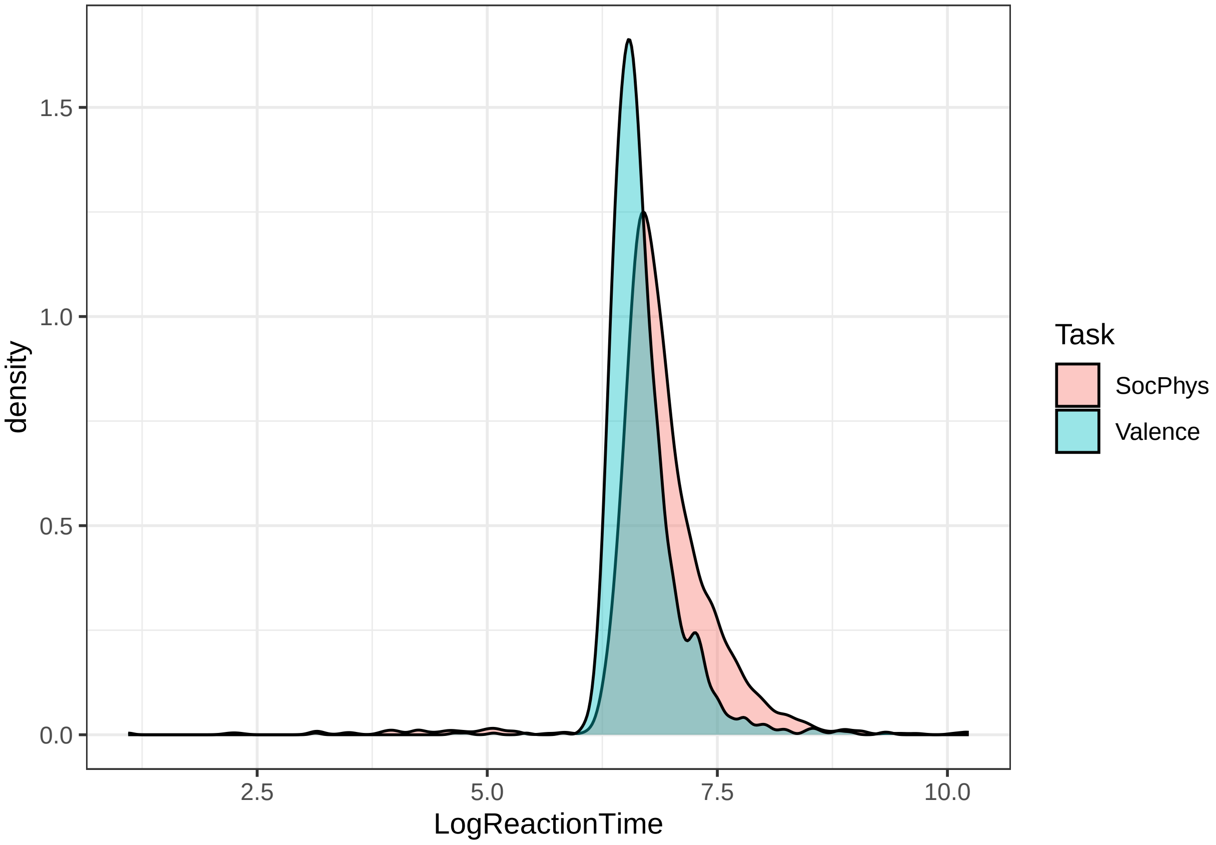
```
##
## negative physical positive   social
##     1240      740     1160     1060
```

```
d <- d %>%
  filter(!ID.true %in% ids_valence_only)
```

```
ggplot(d, aes(x=LogReactionTime)) +
  geom_histogram(binwidth = .1,fill = "lightblue", color = "black") +
  facet_wrap(~Task)
```



```
ggplot(d, aes(x=LogReactionTime, fill=Task)) +
  geom_density(alpha = .4)
```



```r
names(d)
```

```
##  [1] "X"               "ID.true"         "Word"            "Label"          
##  [5] "ConcValCombo"    "Task"            "BlockOrder"      "Group"          
##  [9] "Response"        "Accuracy"        "EventTime"       "Value"          
## [13] "RT"              "ReactionTime"    "Key_value_F"     "Key_value_J"    
## [17] "Comments"        "LogReactionTime" "LogRT"           "TrialNumber"
```

```r
dcen <- d %>%
  mutate(Word = as.factor(Word),
         ID.true = as.factor(ID.true),
         Task = as.factor(Task),
         cAccuracy = as.numeric(Accuracy)-mean(as.numeric(Accuracy)),
         cTask = as.numeric(Task)-mean(as.numeric(Task)))

m <- lmer(LogReactionTime ~ cAccuracy*cTask + (1+cTask|Word) + (1+cTask|ID.true), data = dcen)
summary(m)
```

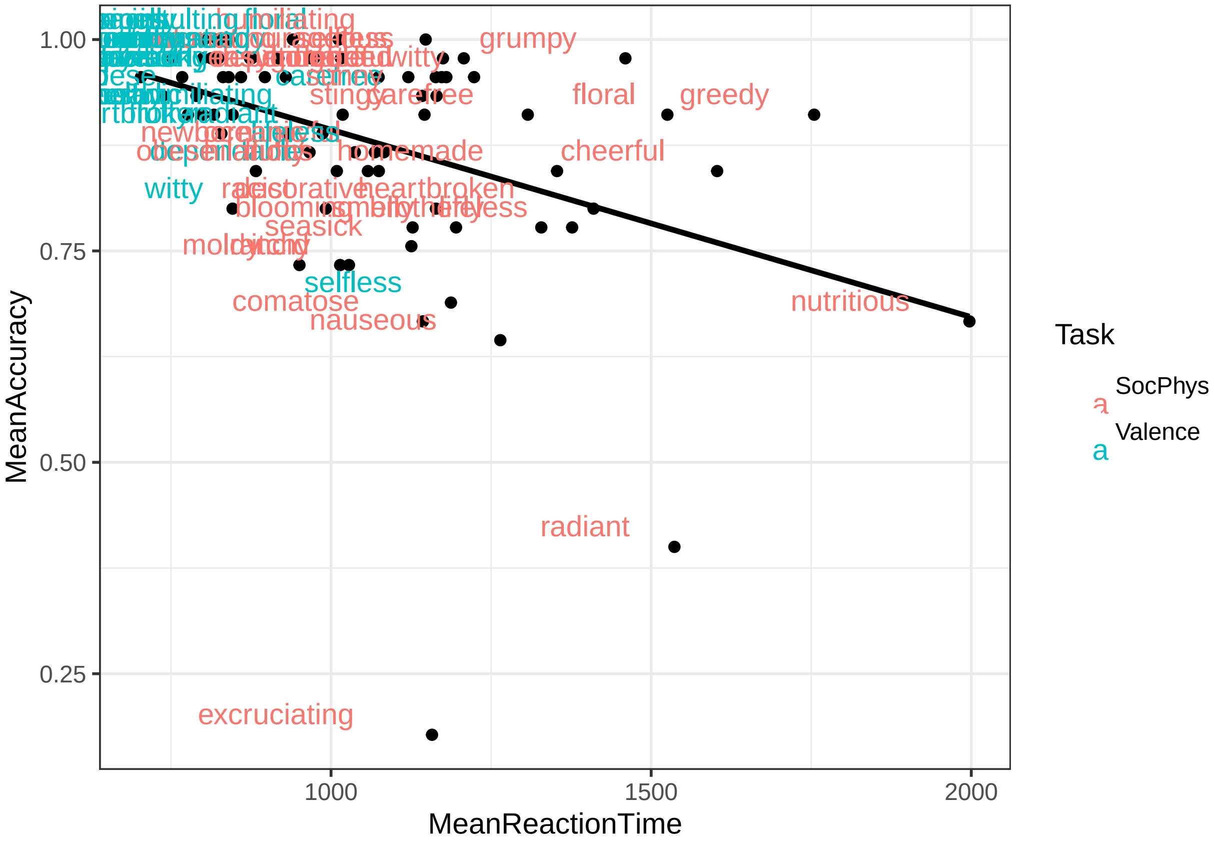
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cAccuracy * cTask + (1 + cTask | Word) + (1 +  
##     cTask | ID.true)
##    Data: dcen
## 
## REML criterion at convergence: 4938.8
## 
## Scaled residuals: 
##      Min       1Q   Median       3Q      Max
## -11.2188  -0.4306  -0.1081   0.3331   7.7838
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.002287 0.04782
##           cTask       0.001896 0.04355  0.12
##  ID.true  (Intercept) 0.032692 0.18081
##           cTask       0.044242 0.21034  -0.31
##  Residual             0.222245 0.47143
## Number of obs: 3600, groups:  Word, 40; ID.true, 15
##
## Fixed effects:
##                    Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)        6.78756    0.04797   14.73674 141.497  < 2e-16 ***
## cAccuracy          0.01762    0.02915 2979.86499   0.604  0.54556
## cTask             -0.22823    0.05705   14.42785  -4.001  0.00124 **
## cAccuracy:cTask    0.10409    0.05840 3140.78842   1.783  0.07475 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cAccrc cTask
## cAccuracy   -0.014
## cTask       -0.286 -0.058
## cAccrcy:cTs -0.034  0.397 -0.023
```

```r
agr <- d %>%
  group_by(Word,Task) %>%
  summarize(MeanAccuracy = mean(Accuracy),
            MeanReactionTime = mean(ReactionTime))
```

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

```r
ggplot(agr, aes(x = MeanReactionTime, y = MeanAccuracy)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  geom_text(aes(label = Word, color = Task), vjust = -0.5, hjust = 1.5)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
  # guides(legend = "none")
  # theme(legend.position = "none")  # Remove the legend
# ggsave("../graphs/exp1b_accXrt.pdf",width = 5, height = 3)
```

```
# Compute highest accuracy for Concrete
concrete_accuracy <- d %>%
  group_by(Word,Task) %>%
  summarize(MeanAccuracy = mean(Accuracy),
            MeanReactionTime = mean(ReactionTime)) %>%
  filter(Task == "SocPhys") %>%
  select(Word, MeanAccuracy) %>%
  rename(ConcreteAccuracy = MeanAccuracy) %>%
  arrange(desc(ConcreteAccuracy)) %>%
  head(10)
```

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

```
agr <- d %>%
  filter(Word %in% concrete_accuracy$Word) %>%
  group_by(Word,Task) %>%
  summarize(MeanAccuracy = mean(Accuracy),
            MeanReactionTime = mean(ReactionTime))
```

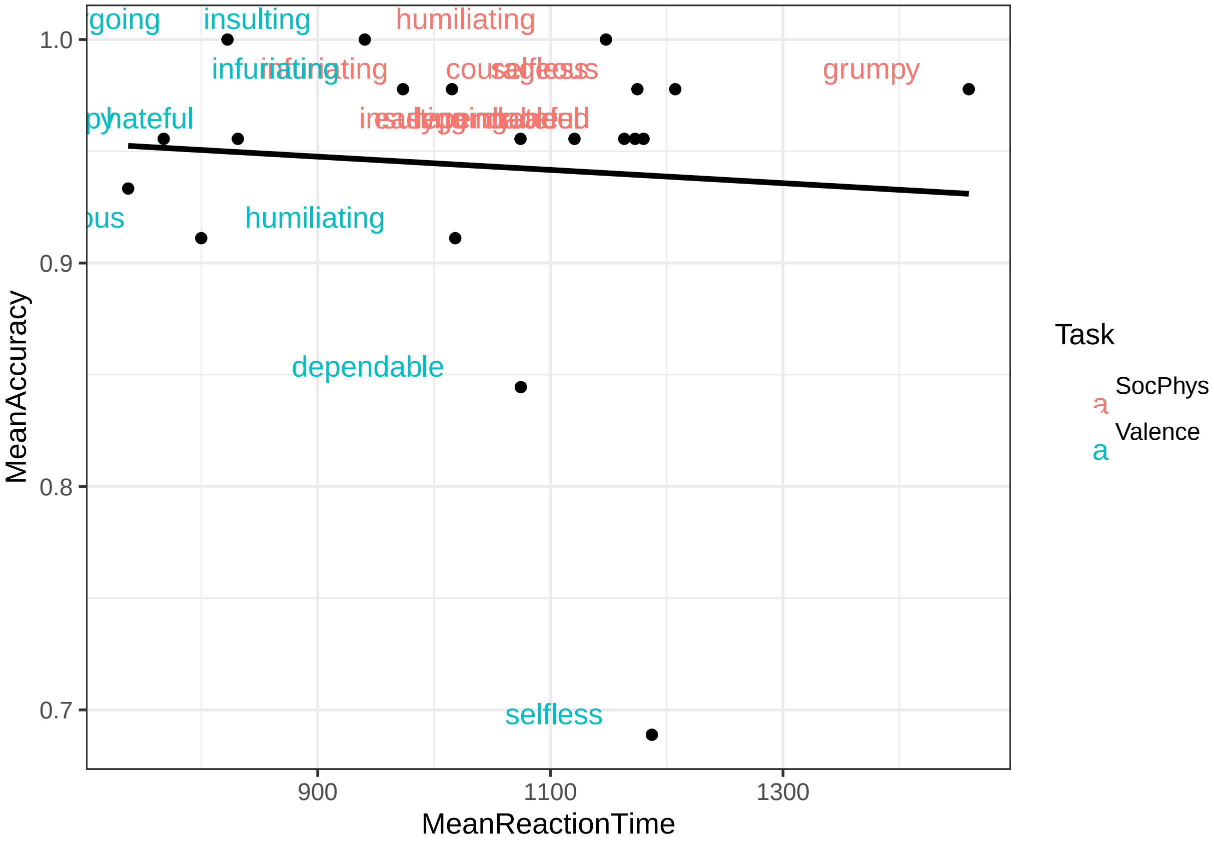
```
## `summarise()` has grouped output by 'Word'. You can override using the
## `.groups` argument.
```

```r
print(agr)
```

```
## # A tibble: 20 x 4
## # Groups:   Word [10]
##    Word        Task    MeanAccuracy MeanReactionTime
##    <chr>       <chr>          <dbl>            <dbl>
##  1 courageous  SocPhys        0.978            1207.
##  2 courageous  Valence        0.911             800.
##  3 dependable  SocPhys        0.956            1173.
##  4 dependable  Valence        0.844            1075.
##  5 dreaded     SocPhys        0.956            1180.
##  6 dreaded     Valence        0.933             737.
##  7 easygoing   SocPhys        0.956            1121.
##  8 easygoing   Valence        1                 822.
##  9 grumpy      SocPhys        0.978            1460.
## 10 grumpy      Valence        0.956             768.
## 11 hateful     SocPhys        0.956            1164.
## 12 hateful     Valence        0.956             831.
## 13 humiliating SocPhys        1                1148.
## 14 humiliating Valence        0.911            1018.
## 15 infuriating SocPhys        0.978            1015.
## 16 infuriating Valence        0.978             973.
## 17 insulting   SocPhys        0.956            1074.
## 18 insulting   Valence        1                 940.
## 19 selfless    SocPhys        0.978            1175.
## 20 selfless    Valence        0.689            1187.
```

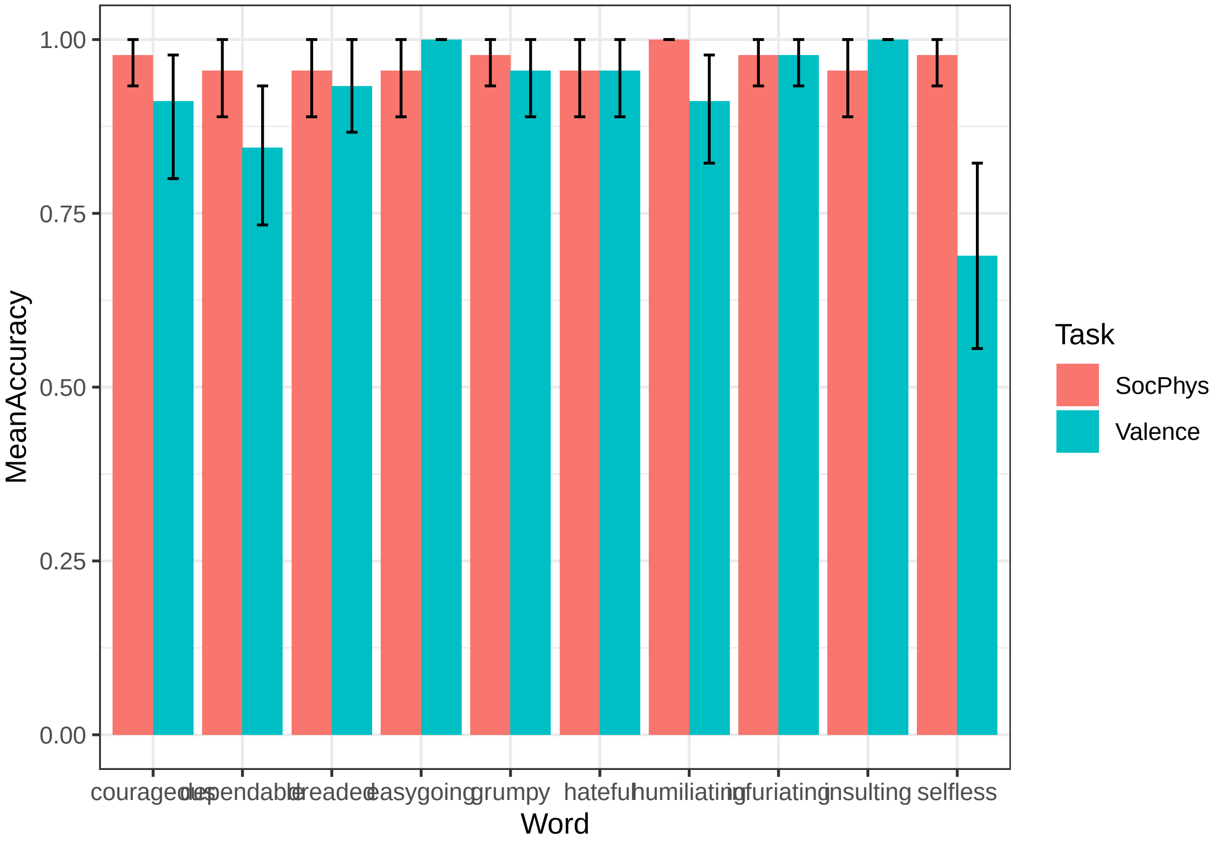
```r
ggplot(agr, aes(x = MeanReactionTime, y = MeanAccuracy)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  geom_text(aes(label = Word, color = Task), vjust = -0.5, hjust = 1.5)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

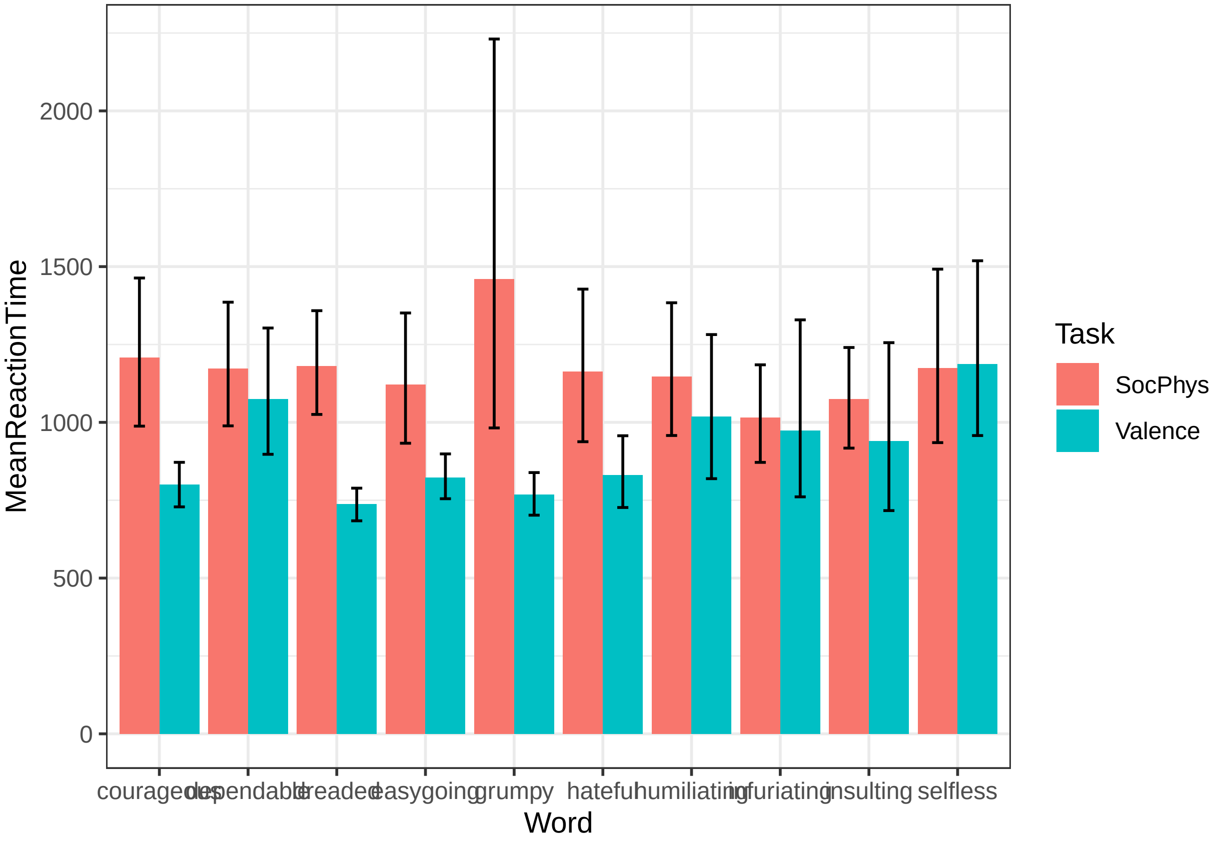


```
  # guides(legend = "none")
  # theme(legend.position = "none")  # Remove the legend
# ggsave("../graphs/exp1b_accXrt.pdf",width = 5, height = 3)
```

```
agr <- d %>%
  filter(Word %in% concrete_accuracy$Word) %>%
  group_by(Word,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=Word,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 %>%
  filter(Word %in% concrete_accuracy$Word) %>%
  group_by(Word,Task) %>%
  reframe(MeanReactionTime = mean(ReactionTime),
          CILow = ci.low(ReactionTime),
          CIHigh = ci.high(ReactionTime)) %>%
  mutate(YMin = MeanReactionTime - CILow,
         YMax = MeanReactionTime + CIHigh)
# View(agr)

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



First Remove participants who aren't super , aggregating over Task

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

```
## [1] 15
```

```r
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.
```

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

```
## [1] 4
```

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

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

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

remove all inaccurate trials

```
orig <- nrow(d.inaccurate.removed)
d.inaccurate.removed <- d.inaccurate.removed %>%
  filter(Accuracy == 1)
nrow(d.inaccurate.removed)/orig*100
```

```
## [1] 92.87879
```

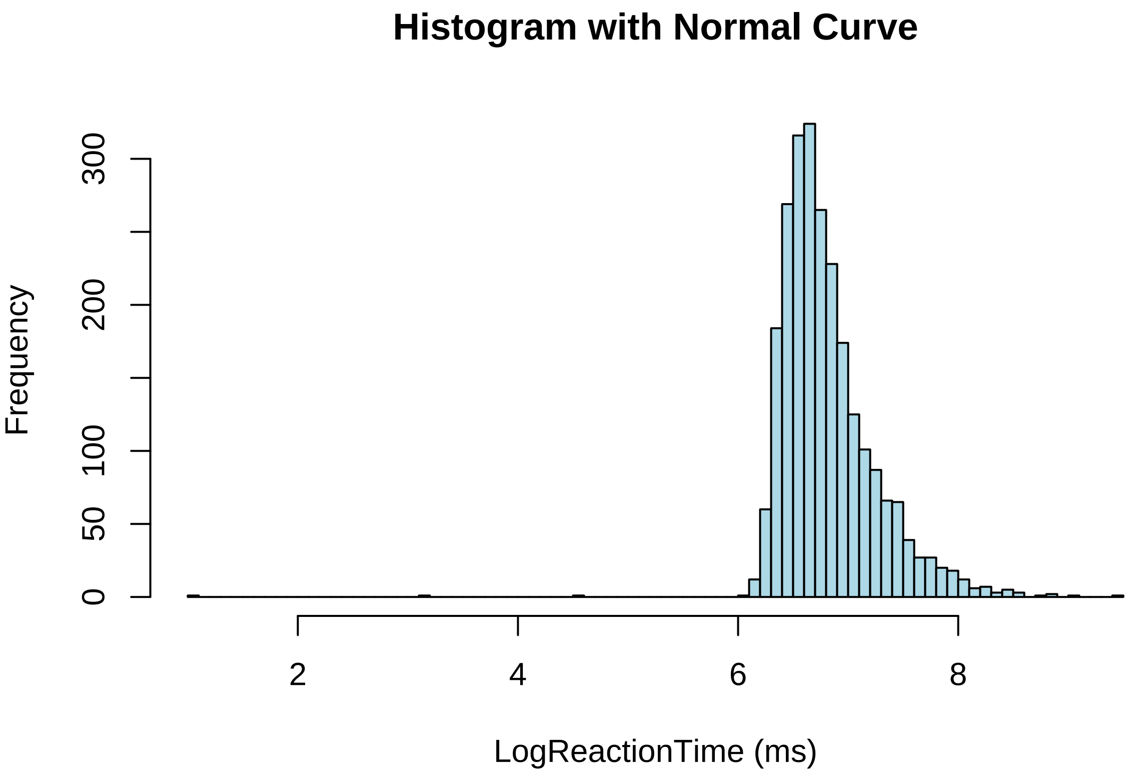
```
# Remove subjects with ReactionTime higher than 3x IQR
summary(d.inaccurate.removed$LogReactionTime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.099   6.529   6.721   6.813   7.002   9.489
```

```
  #  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  # 6.924   7.328   7.436   7.479   7.579  10.008
range(d.inaccurate.removed$LogReactionTime)
```

```
## [1] 1.098612 9.488805
```

```
hist(d.inaccurate.removed$LogReactionTime, breaks=100, col="lightblue", xlab="LogReactionTime (ms)",
        main="Histogram with Normal Curve")
```



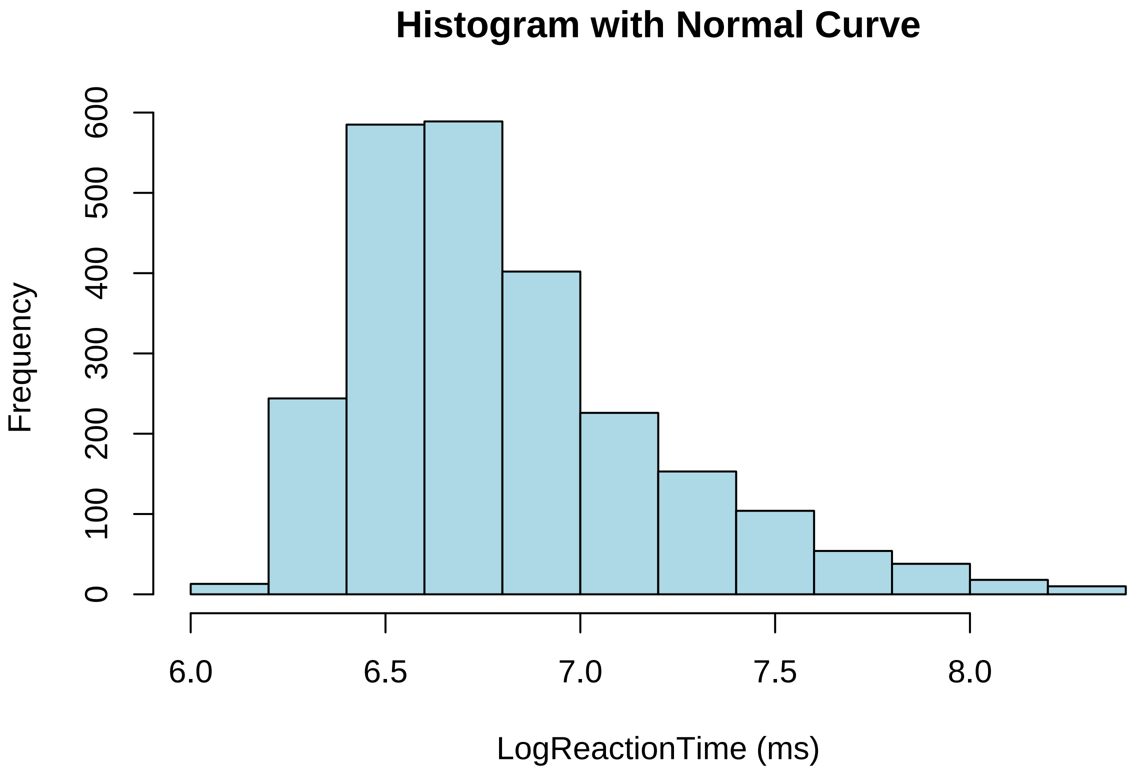
```
quantile(d.inaccurate.removed$LogReactionTime)
```

```
##       0%      25%      50%      75%     100%
## 1.098612 6.529419 6.721426 7.002383 9.488805
```

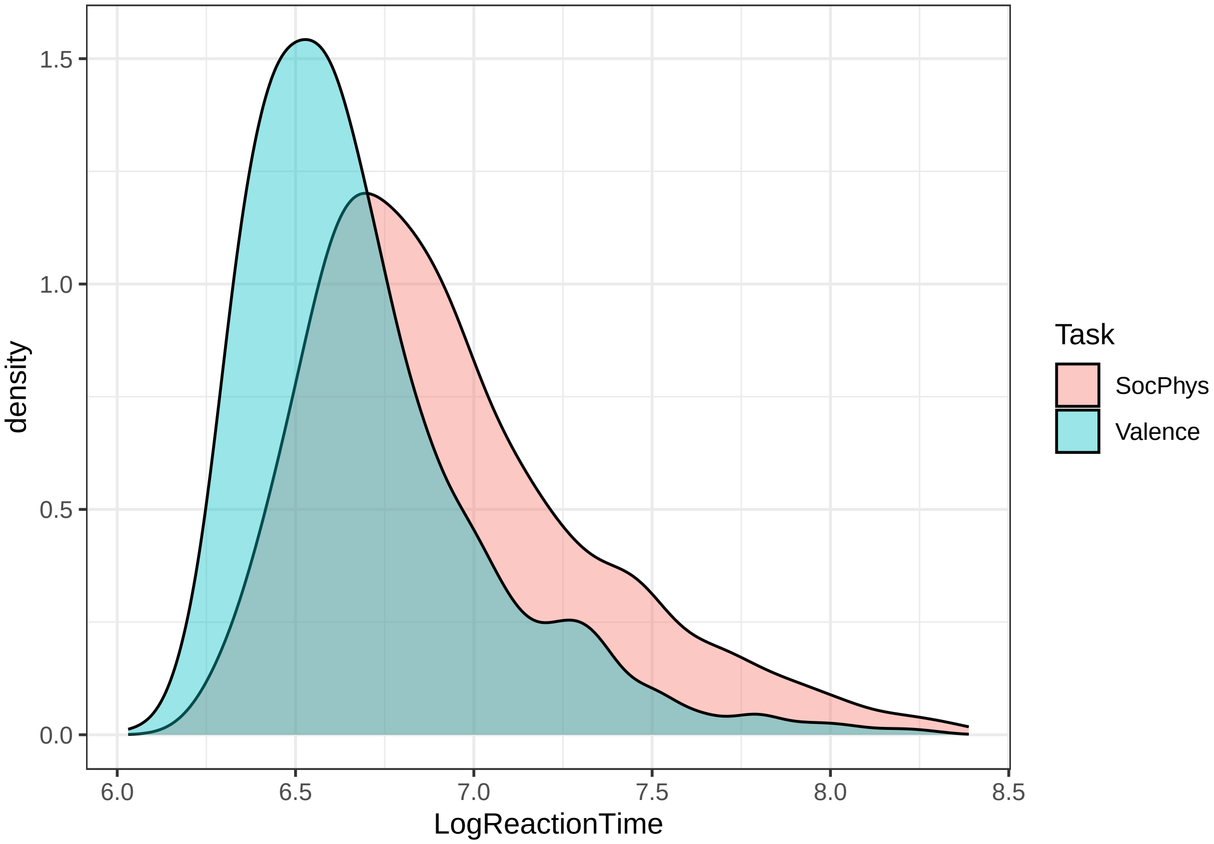
```
IQR(d.inaccurate.removed$LogReactionTime)*3 # 0.7526289
```

```
## [1] 1.418893
```

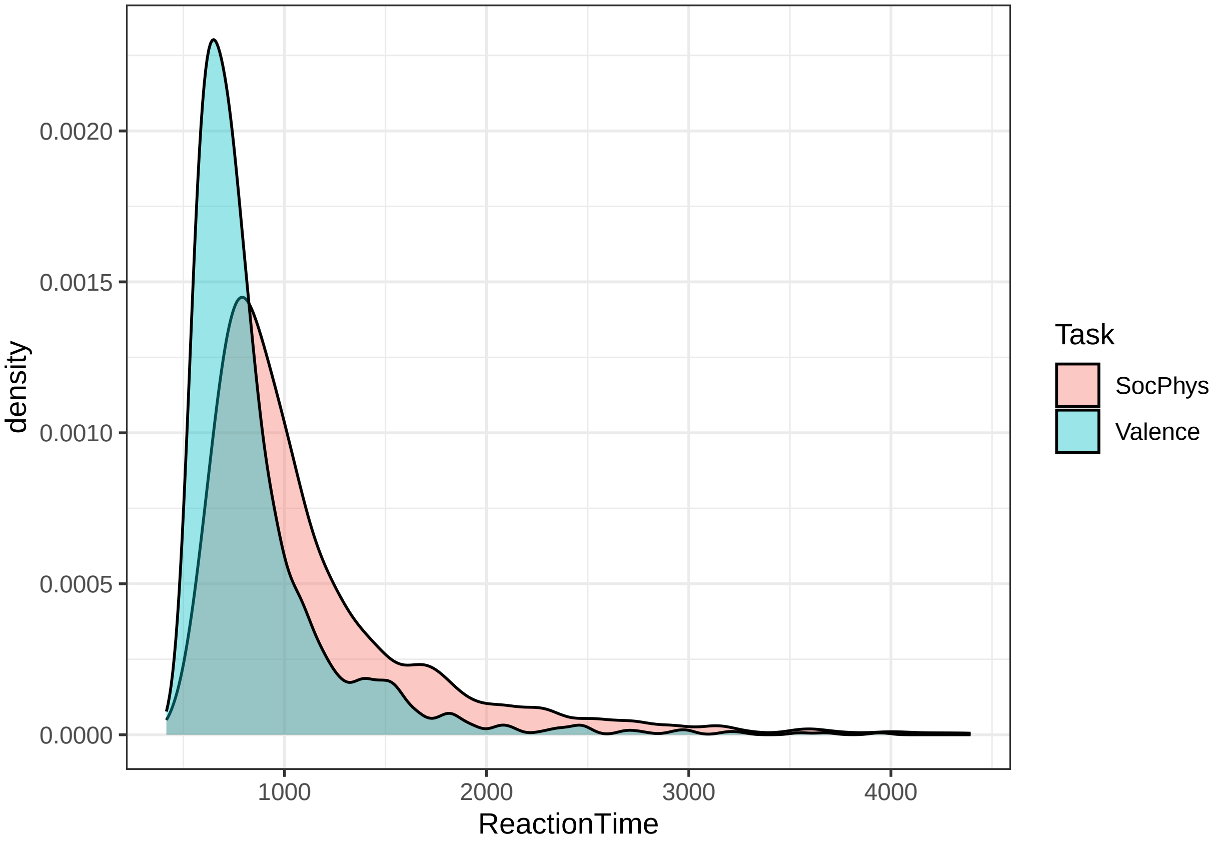
```
cutoff.high <- quantile(d.inaccurate.removed$LogReactionTime)[4] + IQR(d.inaccurate.removed$LogReactionT
cutoff.low <- quantile(d.inaccurate.removed$LogReactionTime)[2] - IQR(d.inaccurate.removed$LogReactionTi


# remove subjects with ReactionTime higher than 3 x IQR
df.outliers.removed <- subset(d.inaccurate.removed, (d.inaccurate.removed$LogReactionTime > cutoff.low)

hist(df.outliers.removed$LogReactionTime, col="lightblue", xlab="LogReactionTime (ms)",
        main="Histogram with Normal Curve")
```



```
ggplot(df.outliers.removed, aes(x=LogReactionTime, fill=Task)) +
  # facet_wrap(~BlockOrder) +
  geom_density(alpha = .4)
```



```r
ggplot(df.outliers.removed, aes(x=ReactionTime, fill=Task)) +
  # facet_wrap(~BlockOrder) +
  geom_density(alpha = .4)
```



convert everything to factors

Is there a difference between Semantic and Valence Tasks?

Yes

```
m = lmer(LogReactionTime ~ cTask + (1+cTask|ID.true) + (1+cTask|Word), data=center)
summary(m)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cTask + (1 + cTask | ID.true) + (1 + cTask |
##     Word)
##    Data: center
##
## REML criterion at convergence: 1352.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5222 -0.6270 -0.1628  0.4901  4.2174
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.003001 0.05478
##           cTask       0.003104 0.05572  0.24
##  ID.true  (Intercept) 0.035088 0.18732
##           cTask       0.019489 0.13960  -0.07
##  Residual             0.096390 0.31047
## Number of obs: 2436, groups:  Word, 40; ID.true, 11
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.81039    0.05749 10.46863 118.467  < 2e-16 ***
## cTask       -0.25831    0.04483 10.76928  -5.762 0.000137 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##       (Intr)
## cTask -0.060
```

Is there an Interaction between Task and WordType (ConcValCombo)?

Y.

```r
m = lmer(LogReactionTime ~ cTask*ConcValCombo + (1+ConcValCombo+cTask|ID.true) + (1+cTask|Word), data=c
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00217965 (tol = 0.002, component 1)
```

```r
saveRDS(m, "../models/model-Task-ConcValCombo_outlier_excl_ReactionTime.rds")

# m <- readRDS("../models/model-Task-ConcValCombo_outlier_excl_ReactionTime.rds")
summary(m)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cTask * ConcValCombo + (1 + ConcValCombo +
##     cTask | ID.true) + (1 + cTask | Word)
##    Data: center
##
## REML criterion at convergence: 1348.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6385 -0.6322 -0.1452  0.4943  4.3021
##
## Random effects:
##  Groups   Name                        Variance Std.Dev. Corr
##  Word     (Intercept)                 0.003077 0.05547
##           cTask                       0.003877 0.06226  0.23
##  ID.true  (Intercept)                 0.034485 0.18570
##           ConcValCombophysical-positive 0.002684 0.05181  -0.08
##           ConcValCombosocial-negative   0.002178 0.04667  -0.14  0.40
##           ConcValCombosocial-positive   0.011666 0.10801  -0.03  0.22  0.97
##           cTask                       0.019708 0.14039   0.04 -0.72 -0.32
##  Residual                             0.094286 0.30706
##
##
##
```

```
##
##
##
##
##  -0.27
##
## Number of obs: 2436, groups:  Word, 40; ID.true, 11
##
## Fixed effects:
##                                      Estimate Std. Error        df t value
## (Intercept)                          6.829501   0.060087 11.899826 113.660
## cTask                               -0.277283   0.053436 19.913225  -5.189
## ConcValCombophysical-positive       -0.045144   0.034413 24.553216  -1.312
## ConcValCombosocial-negative         -0.021303   0.033636 29.791063  -0.633
## ConcValCombosocial-positive         -0.006644   0.044690 18.886569  -0.149
## cTask:ConcValCombophysical-positive  0.014516   0.045613 36.815738   0.318
## cTask:ConcValCombosocial-negative    0.018886   0.045300 36.022809   0.417
## cTask:ConcValCombosocial-positive    0.040950   0.045421 36.391968   0.902
##                                     Pr(>|t|)
## (Intercept)                          < 2e-16 ***
## cTask                               4.52e-05 ***
## ConcValCombophysical-positive          0.202
## ConcValCombosocial-negative            0.531
## ConcValCombosocial-positive            0.883
## cTask:ConcValCombophysical-positive    0.752
## cTask:ConcValCombosocial-negative      0.679
## cTask:ConcValCombosocial-positive      0.373
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                    (Intr) cTask  CncVlCmbp- CncVlCmbscl-n CncVlCmbscl-p
## cTask               0.045
## CncVlCmbph-        -0.264 -0.285
## CncVlCmbscl-n      -0.289 -0.133  0.487
## CncVlCmbscl-p      -0.200 -0.178  0.382      0.613
## cTsk:CncVlCmbp-    -0.018 -0.436  0.079      0.033         0.025
## cTsk:CncVlCmbscl-n -0.018 -0.440  0.032      0.090         0.025
## cTsk:CncVlCmbscl-p -0.018 -0.438  0.032      0.033         0.072
##                    cTsk:CncVlCmbp- cTsk:CncVlCmbscl-n
## cTask
## CncVlCmbph-
## CncVlCmbscl-n
## CncVlCmbscl-p
## cTsk:CncVlCmbp-
## cTsk:CncVlCmbscl-n  0.515
## cTsk:CncVlCmbscl-p  0.513           0.517
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00217965 (tol = 0.002, component 1)
```

Does Accuracy predict reaction time?

In other words, is reaction time affected by certainty about the categorization? - No.

```
m = lmer(LogReactionTime ~ cAccuracy + (1|ID.true) + (1|Word), data=center)
```

```
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

```
summary(m)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cAccuracy + (1 | ID.true) + (1 | Word)
##    Data: center
##
## REML criterion at convergence: 1800.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1670 -0.6915 -0.1771  0.4927  4.3979
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Word     (Intercept) 0.002306 0.04802
##  ID.true  (Intercept) 0.034702 0.18628
##  Residual             0.118682 0.34450
## Number of obs: 2436, groups:  Word, 40; ID.true, 11
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.80917    0.05711 10.36228   119.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

Main Effect of Block Order

On ReactionTime

- No.

```
m = lmer(LogReactionTime ~ cBlockOrder + (1|ID.true) + (1+cBlockOrder|Word), data=center)

summary(m)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cBlockOrder + (1 | ID.true) + (1 + cBlockOrder |
##     Word)
##    Data: center
##
## REML criterion at convergence: 1800.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.2053 -0.6921 -0.1722  0.4951  4.4006
##
## Random effects:
```

```
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.002305 0.04801
##           cBlockOrder 0.002853 0.05341  0.23
##  ID.true  (Intercept) 0.037936 0.19477
##  Residual             0.118116 0.34368
## Number of obs: 2436, groups:  Word, 40; ID.true, 11
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.80945    0.05963  9.29604 114.197 5.99e-16 ***
## cBlockOrder -0.05399    0.13305  9.06715  -0.406    0.694
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cBlockOrder -0.009
```

effect of ConcValCombo on ReactionTime?

nope.

```
m = lmer(LogReactionTime ~ ConcValCombo + (1+ConcValCombo|ID.true) + (1|Word), data=center)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(m)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ ConcValCombo + (1 + ConcValCombo | ID.true) +
##     (1 | Word)
##    Data: center
##
## REML criterion at convergence: 1794.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4115 -0.6894 -0.1713  0.4964  4.5038
##
## Random effects:
##  Groups   Name                          Variance  Std.Dev.  Corr
##  Word     (Intercept)                   2.285e-03 0.0478056
##  ID.true  (Intercept)                   3.498e-02 0.1870275
##           ConcValCombophysical-positive 9.831e-07 0.0009915 -1.00
##           ConcValCombosocial-negative   1.956e-03 0.0442252 -0.23  0.23
##           ConcValCombosocial-positive   1.065e-02 0.1032220 -0.07  0.07  0.99
##  Residual                               1.170e-01 0.3421076
## Number of obs: 2436, groups:  Word, 40; ID.true, 11
##
## Fixed effects:
##                                Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)                    6.820127   0.060107 11.373188 113.467   <2e-16
## ConcValCombophysical-positive -0.039049   0.029248 36.176948  -1.335    0.190
## ConcValCombosocial-negative   -0.010491   0.032019 23.249571  -0.328    0.746
## ConcValCombosocial-positive    0.005921   0.042654 13.776476   0.139    0.892
##
## (Intercept)                   ***
## ConcValCombophysical-positive
## ConcValCombosocial-negative
## ConcValCombosocial-positive
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##               (Intr) CncVlCmbp- CncVlCmbscl-n
## CncVlCmbph-   -0.256
## CncVlCmbscl-n -0.313  0.463
## CncVlCmbscl-p -0.214  0.347      0.617
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

In the Concreteness task, is there a difference between concreteness and abstractness on ReactionTime?

- Nope

```r
str(df_factors)
```

```
## 'data.frame':    2436 obs. of  20 variables:
##  $ X              : Factor w/ 2436 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ ID.true        : Factor w/ 11 levels "580cab86b869700001d71628",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Word           : Factor w/ 40 levels "blooming","brotherly",..: 39 32 2 3 12 21 1 17 26 13 ...
##  $ Label          : Factor w/ 2 levels "test_sp","test_val": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ConcValCombo   : Factor w/ 4 levels "physical-negative",..: 2 3 4 4 2 3 2 3 1 2 ...
##  $ Task           : Factor w/ 2 levels "SocPhys","Valence": 1 1 1 1 1 1 1 1 1 1 ...
##  $ BlockOrder     : Factor w/ 2 levels "SV","VS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Group          : Factor w/ 0 levels: NA NA NA NA NA NA NA NA NA NA ...
##  $ Response       : Factor w/ 4 levels "negative","physical",..: 2 4 4 4 2 4 2 4 2 2 ...
##  $ Accuracy       : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ EventTime      : Factor w/ 2434 levels "1748356915313",..: 2210 2211 2212 2213 2214 2215 2216 221
##  $ Value          : Factor w/ 4 levels "negative","physical",..: 2 4 4 4 2 4 2 4 2 2 ...
##  $ RT             : Factor w/ 421 levels "1271.5","1351.16666666667",..: 355 176 339 85 249 337 282
##  $ ReactionTime   : int  2503 1752 1530 936 1979 2292 2742 1874 1473 2244 ...
##  $ Key_value_F    : Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Key_value_J    : Factor w/ 4 levels "negative;positive",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Comments       : Factor w/ 0 levels: NA NA NA NA NA NA NA NA NA NA ...
##  $ LogReactionTime: num  7.83 7.47 7.33 6.84 7.59 ...
##  $ LogRT          : Factor w/ 421 levels "7.14795258486217",..: 355 176 339 85 249 337 282 418 273 4
##  $ TrialNumber    : Factor w/ 240 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
```

```r
sem <- df_factors %>%
  filter(Task == "SocPhys") %>%
  mutate(
         Semantic = ifelse(grepl("physical", ConcValCombo), "physical",
                    ifelse(grepl("social", ConcValCombo), "social", NA)),
         Valence =  ifelse(grepl("positive", ConcValCombo), "positive",
                    ifelse(grepl("negative", ConcValCombo), "negative", NA)),
        cConcValCombo = as.numeric(ConcValCombo) - mean(as.numeric(ConcValCombo)),
        cSemantic = as.numeric(factor(Semantic)) - mean(as.numeric(factor(Semantic)))
  )

m = lmer(LogReactionTime ~ cConcValCombo + (1+cConcValCombo|ID.true) + (1+cConcValCombo|Word), data=sem
```

```
## boundary (singular) fit: see help('isSingular')
```

```r
summary(m)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cConcValCombo + (1 + cConcValCombo | ID.true) +
##     (1 + cConcValCombo | Word)
##    Data: sem
##
## REML criterion at convergence: 909.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.3113 -0.6842 -0.1617  0.5679  3.7856
##
## Random effects:
##  Groups   Name          Variance Std.Dev. Corr
##  Word     (Intercept)   0.001286 0.03586
##           cConcValCombo 0.001426 0.03777  -1.00
##  ID.true  (Intercept)   0.042788 0.20685
##           cConcValCombo 0.002714 0.05210  0.33
##  Residual               0.118698 0.34453
## Number of obs: 1174, groups:  Word, 40; ID.true, 11
##
## Fixed effects:
##                Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)    6.944213   0.063700 10.326177 109.015   <2e-16 ***
## cConcValCombo -0.001511   0.019970 13.528396  -0.076    0.941
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cConcValCmb 0.208
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```r
m = lmer(LogReactionTime ~ cSemantic + (1+cSemantic|ID.true) + (1|Word), data=sem)
summary(m)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cSemantic + (1 + cSemantic | ID.true) + (1 |
##     Word)
##    Data: sem
##
## REML criterion at convergence: 915
##
## Scaled residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -2.2789 -0.6703 -0.1605  0.5795  3.7415
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.002155 0.04642
##  ID.true  (Intercept) 0.042745 0.20675
##           cSemantic   0.012230 0.11059  0.13
##  Residual             0.119552 0.34576
## Number of obs: 1174, groups:  Word, 40; ID.true, 11
##
## Fixed effects:
##              Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)  6.943668   0.063590 10.272201 109.195   <2e-16 ***
## cSemantic    0.001342   0.041769 12.559647   0.032    0.975
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## cSemantic 0.103
```

In the Valence task , is there a difference between positive and negative on ReactionTime?

- Nope.

```
val <- df_factors %>%
  filter(Task == "Valence") %>%
  mutate(
         Semantic = ifelse(grepl("concrete", ConcValCombo), "concrete",
                    ifelse(grepl("abstract", ConcValCombo), "abstract", NA)),
         Valence = ifelse(grepl("positive", ConcValCombo), "positive",
                    ifelse(grepl("negative", ConcValCombo), "negative", NA)),
         cConcValCombo = as.numeric(ConcValCombo) - mean(as.numeric(ConcValCombo)),
         cValence = as.numeric(factor(Valence)) - mean(as.numeric(factor(Valence)))
         )

m = lmer(LogReactionTime ~ cConcValCombo + (1+cConcValCombo|ID.true) + (1+cConcValCombo|Word), data=val
```

```
## boundary (singular) fit: see help('isSingular')
```

```
## Warning: Model failed to converge with 1 negative eigenvalue: -3.1e+02
```

```
summary(m)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cConcValCombo + (1 + cConcValCombo | ID.true) +
##     (1 + cConcValCombo | Word)
##    Data: val
##
## REML criterion at convergence: 350
##
## Scaled residuals:
```

```
##     Min      1Q  Median      3Q     Max 
## -2.9238 -0.6065 -0.1254  0.4629  5.1264 
## 
## Random effects:
##  Groups   Name          Variance Std.Dev. Corr
##  Word     (Intercept)   0.000000 0.00000      
##           cConcValCombo 0.005004 0.07074   NaN
##  ID.true  (Intercept)   0.038120 0.19524      
##           cConcValCombo 0.001469 0.03832  0.23
##  Residual               0.070755 0.26600      
## Number of obs: 1262, groups:  Word, 40; ID.true, 11
## 
## Fixed effects:
##               Estimate Std. Error       df t value Pr(>|t|)    
## (Intercept)    6.67701    0.06003 10.47006 111.227   <2e-16 ***
## cConcValCombo  0.01104    0.01853 24.41520   0.595    0.557    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation of Fixed Effects:
##             (Intr)
## cConcValCmb 0.143 
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```r
m = lmer(LogReactionTime ~ cValence + (1+cValence|ID.true) + (1|Word), data=val)
summary(m)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cValence + (1 + cValence | ID.true) + (1 |  
##     Word)
##    Data: val
## 
## REML criterion at convergence: 361.1
## 
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -2.6767 -0.6233 -0.1519  0.4376  4.8531 
## 
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr 
##  Word     (Intercept) 0.005607 0.07488       
##  ID.true  (Intercept) 0.038043 0.19505       
##           cValence    0.003559 0.05966  -0.19
##  Residual             0.071453 0.26731       
## Number of obs: 1262, groups:  Word, 40; ID.true, 11
## 
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)    
## (Intercept)  6.68651    0.06046 10.80678 110.593   <2e-16 ***
## cValence    -0.00684    0.03334 26.01547  -0.205    0.839    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
```

```
## Correlation of Fixed Effects:
##          (Intr)
## cValence -0.102
```