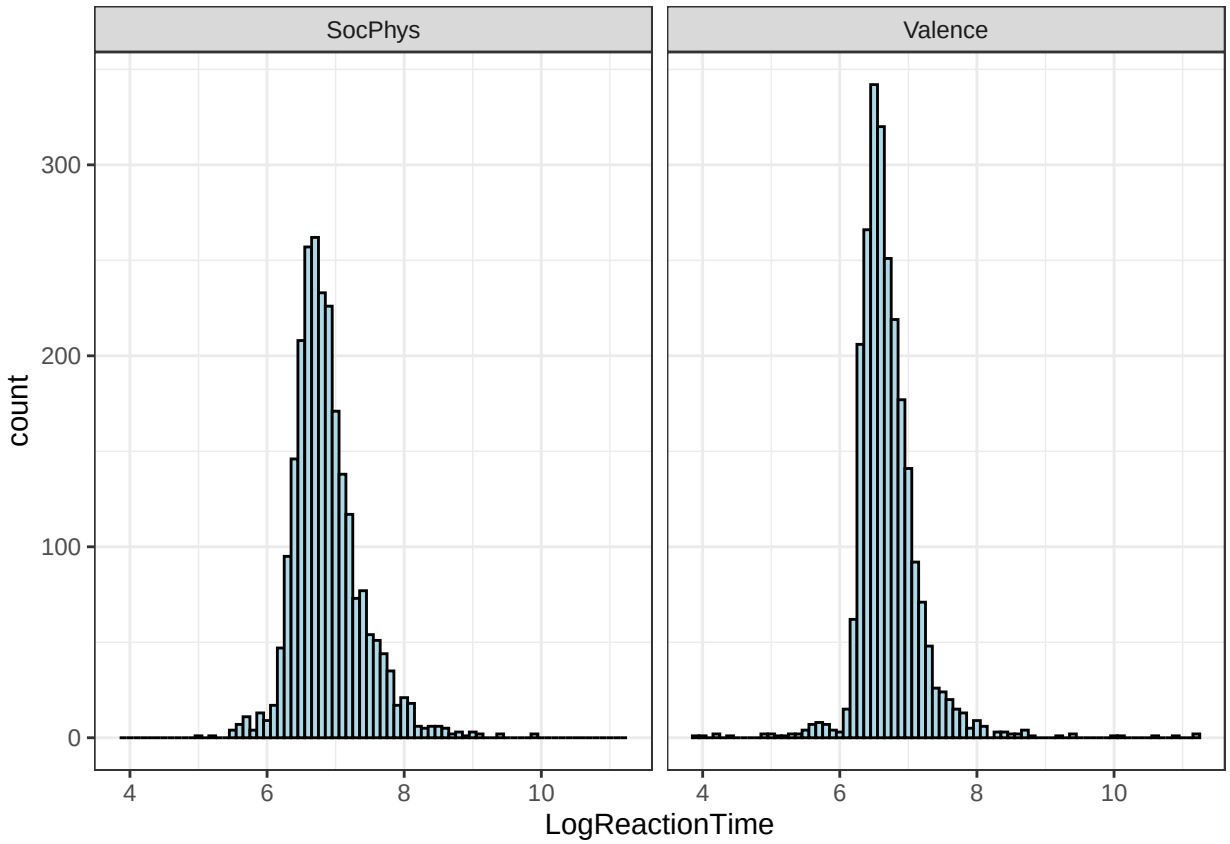
# Adjs Soc-Phys: Analysis

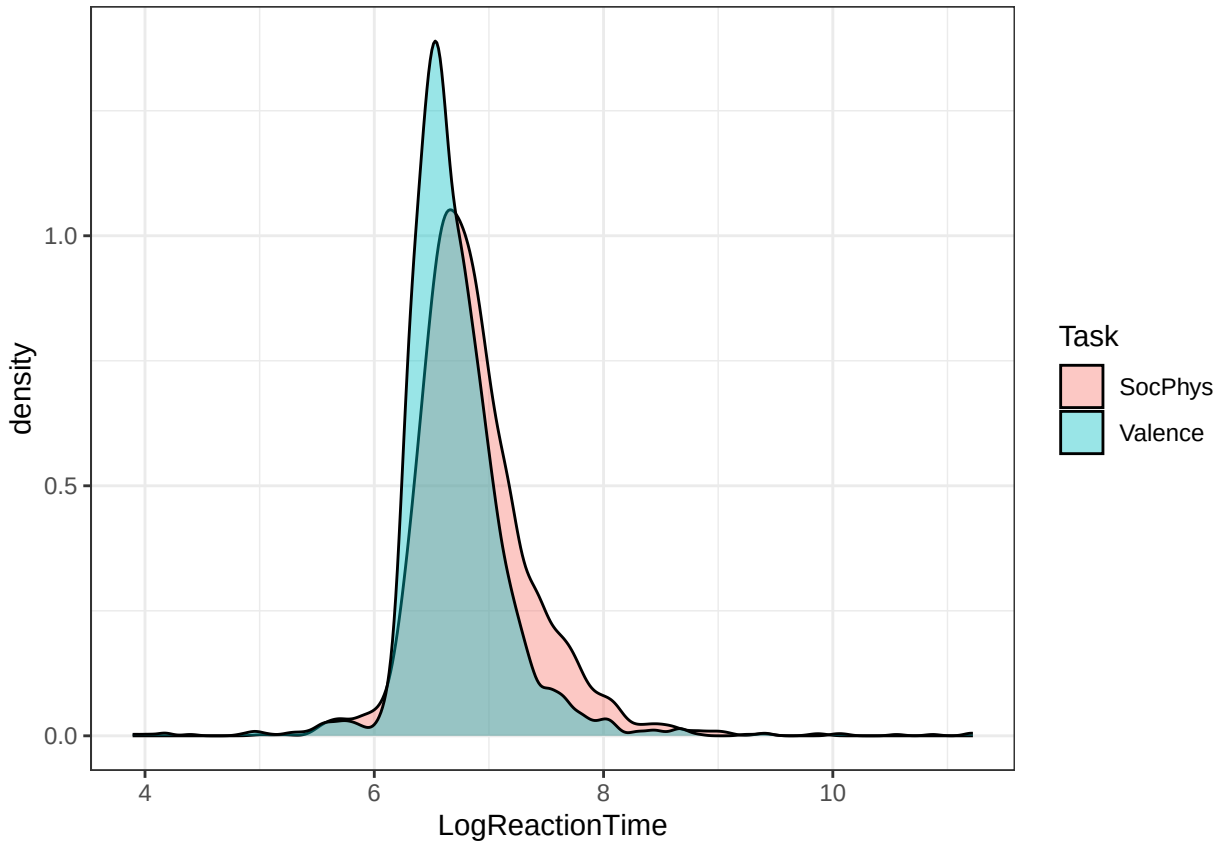
morgan moyer

2025-05-19

```
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)
```




```
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"
```

```
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: 5815.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
```

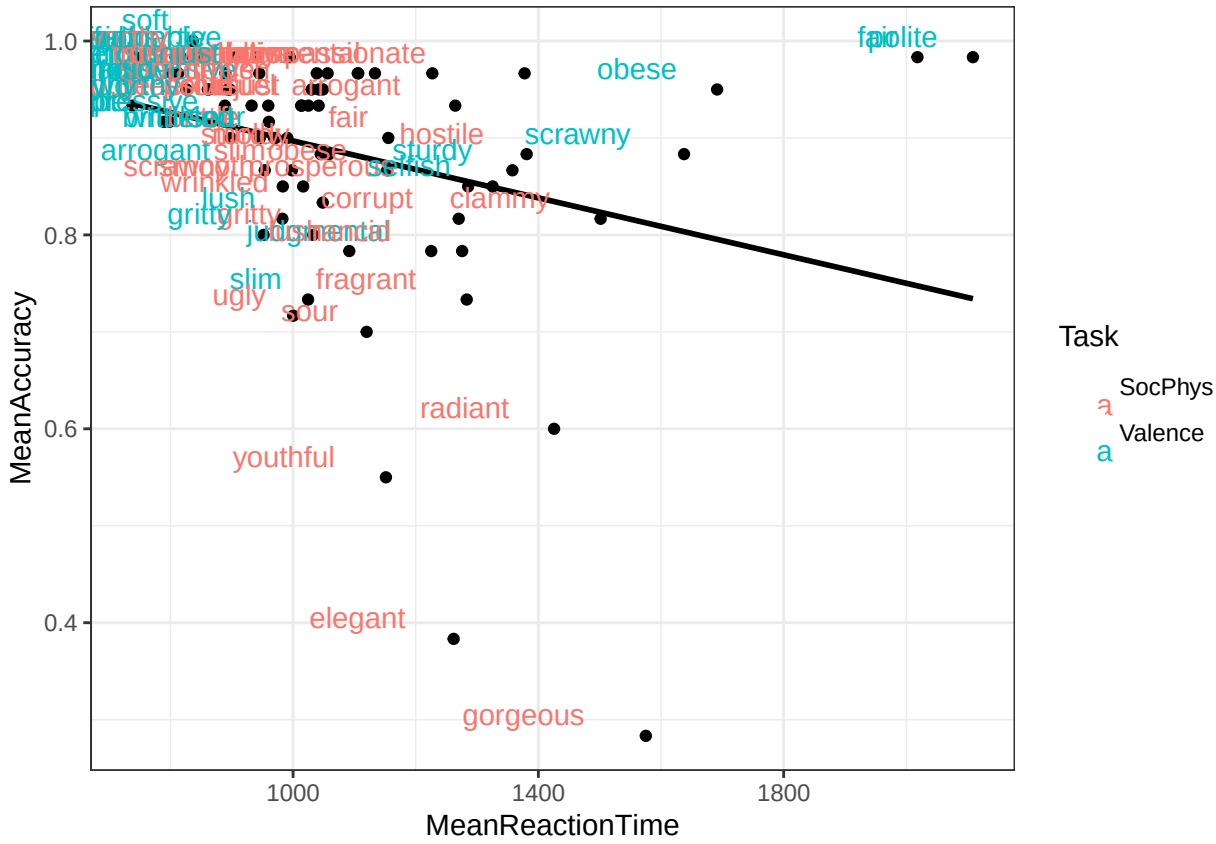
```
## -6.8487 -0.5274 -0.1597  0.3530 10.3751
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  Word     (Intercept) 0.0007582 0.02754
##           cTask       0.0113768 0.10666  -0.23
##  ID.true  (Intercept) 0.0318649 0.17851
##           cTask       0.0449873 0.21210  -0.57
##  Residual             0.1889451 0.43468
## Number of obs: 4800, groups:  Word, 40; ID.true, 20
##
## Fixed effects:
##                   Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)        6.79263    0.04065   19.45981 167.090  < 2e-16 ***
## cAccuracy         -0.03667    0.02365 4162.62134  -1.550  0.12116
## cTask             -0.17243    0.05192   23.40350  -3.321  0.00293 **
## cAccuracy:cTask   -0.07025    0.04697 3760.30553  -1.496  0.13484
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cAccrc cTask
## cAccuracy   -0.008
## cTask       -0.522 -0.040
## cAccrcy:cTs -0.025  0.316 -0.013
```

```
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.
```

```
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))
```

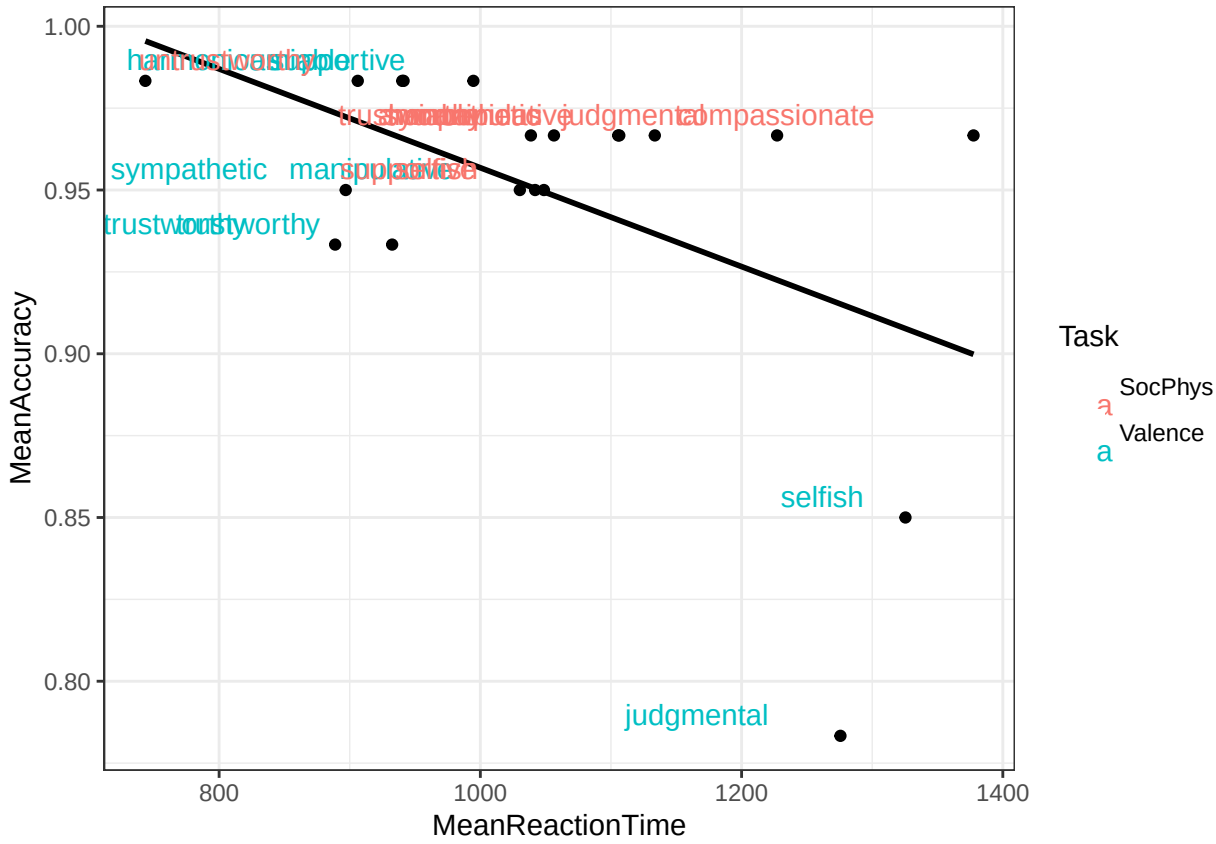
```
## `summarise()` has grouped output by 'Word'. You can override using the
## `.groups` argument.
```

```r
print(agr)
```

```
## # A tibble: 20 x 4
## # Groups:   Word [10]
##    Word          Task    MeanAccuracy MeanReactionTime
##    <chr>         <chr>          <dbl>            <dbl>
##  1 amiable       SocPhys        0.967            1039.
##  2 amiable       Valence        0.983             940.
##  3 compassionate SocPhys        0.967            1378.
##  4 compassionate Valence        0.983             744.
##  5 harmonious    SocPhys        0.967            1106.
##  6 harmonious    Valence        0.983             906.
##  7 judgmental    SocPhys        0.967            1227.
##  8 judgmental    Valence        0.783            1276.
##  9 manipulative  SocPhys        0.967            1134.
## 10 manipulative  Valence        0.95             1042.
## 11 selfish       SocPhys        0.95             1030.
## 12 selfish       Valence        0.85             1325.
## 13 supportive    SocPhys        0.95             1049.
## 14 supportive    Valence        0.983             995.
## 15 sympathetic   SocPhys        0.967            1106.
## 16 sympathetic   Valence        0.95              897.
## 17 trustworthy   SocPhys        0.967            1056.
## 18 trustworthy   Valence        0.933             932.
## 19 untrustworthy SocPhys        0.983             941.
## 20 untrustworthy Valence        0.933             889.
```

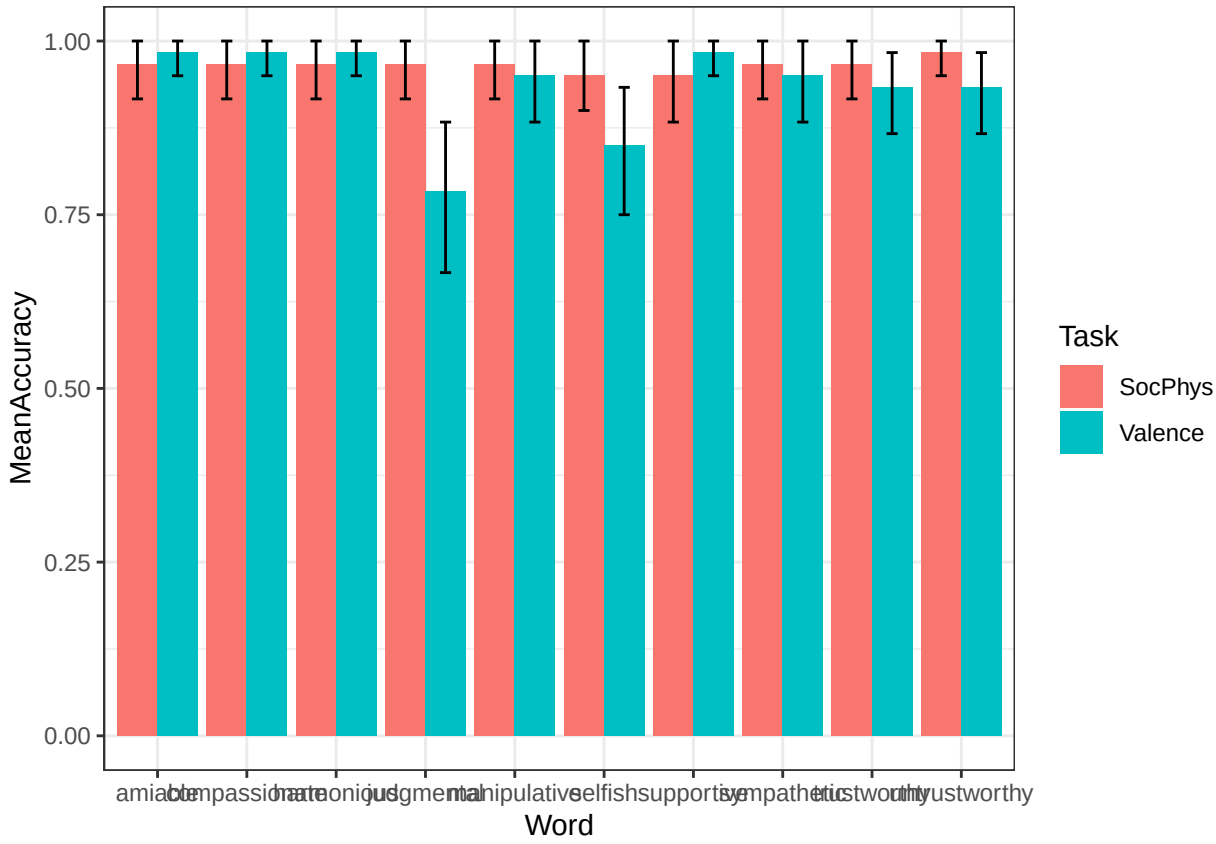
```r
ggplot(agr, aes(x = MeanReactionTime, y = MeanAccuracy)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  geom_text(aes(label = Word, color = Task), vjust = -0.5, hjust = 1.5)
```

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

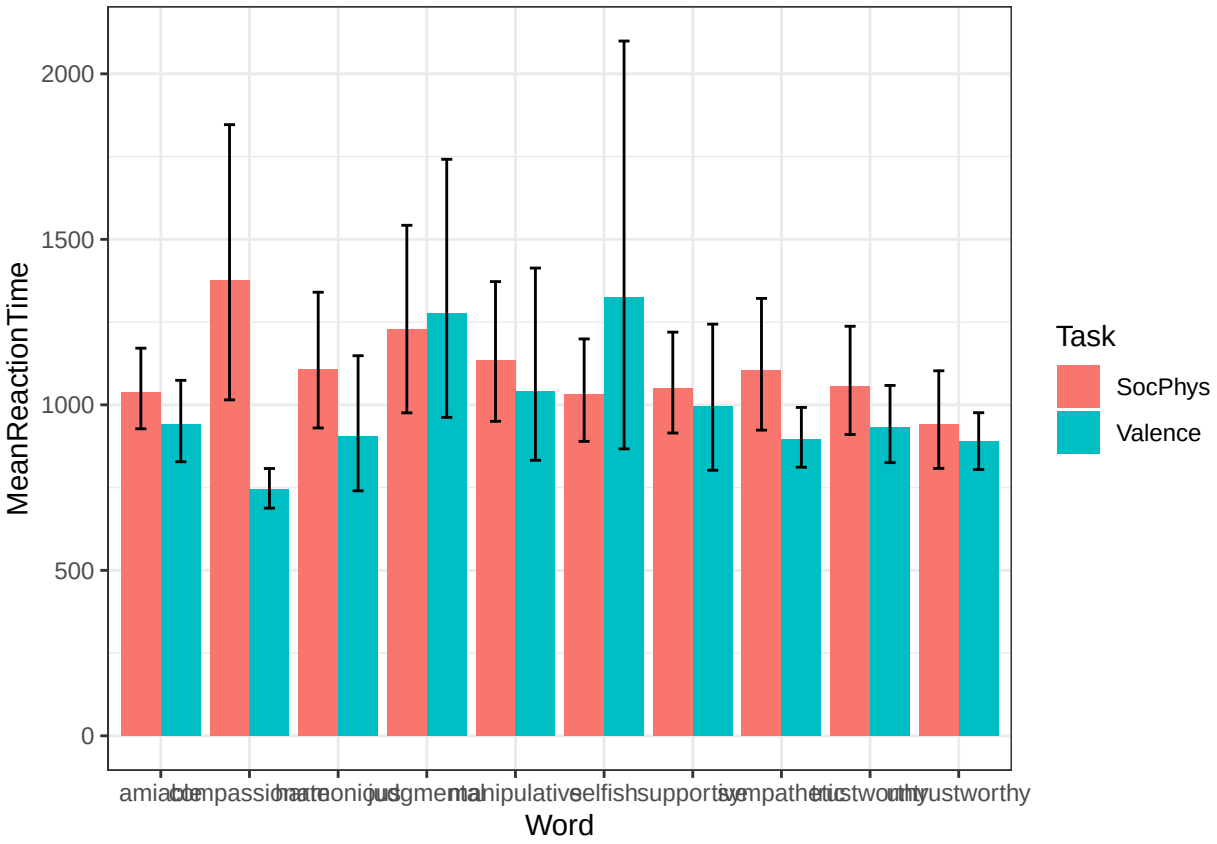



```
  # guides(legend = "none")
  # theme(legend.position = "none")  # Remove the legend
# ggsave("../graphs/exp1b_accXrt.pdf",width = 5, height = 3)
```

```
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))
```




```r
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

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

```
## [1] 20
```

```
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] 2
```

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

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

```
## [1] 18
```

## 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] 91.96759
```

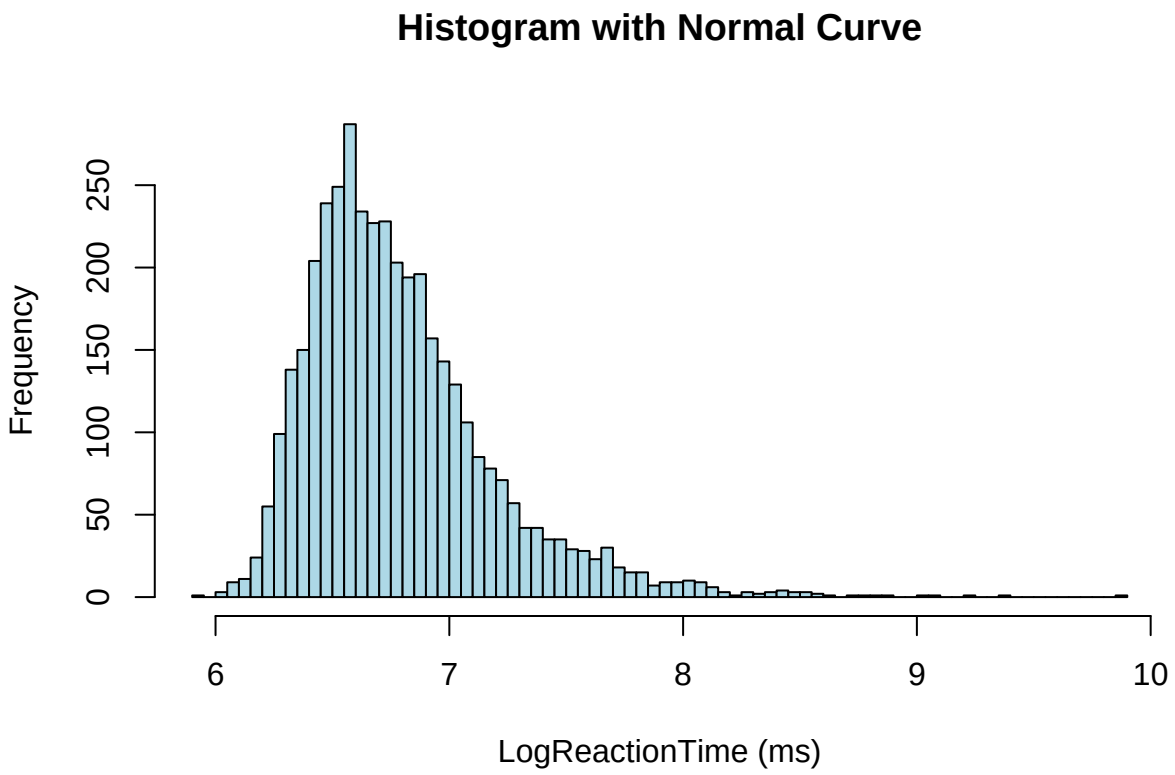
```
# Remove subjects with ReactionTime higher than 3x IQR
summary(d.inaccurate.removed$LogReactionTime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.935   6.513   6.713   6.790   6.974   9.879
```

```
  #  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] 5.934894 9.878836
```

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



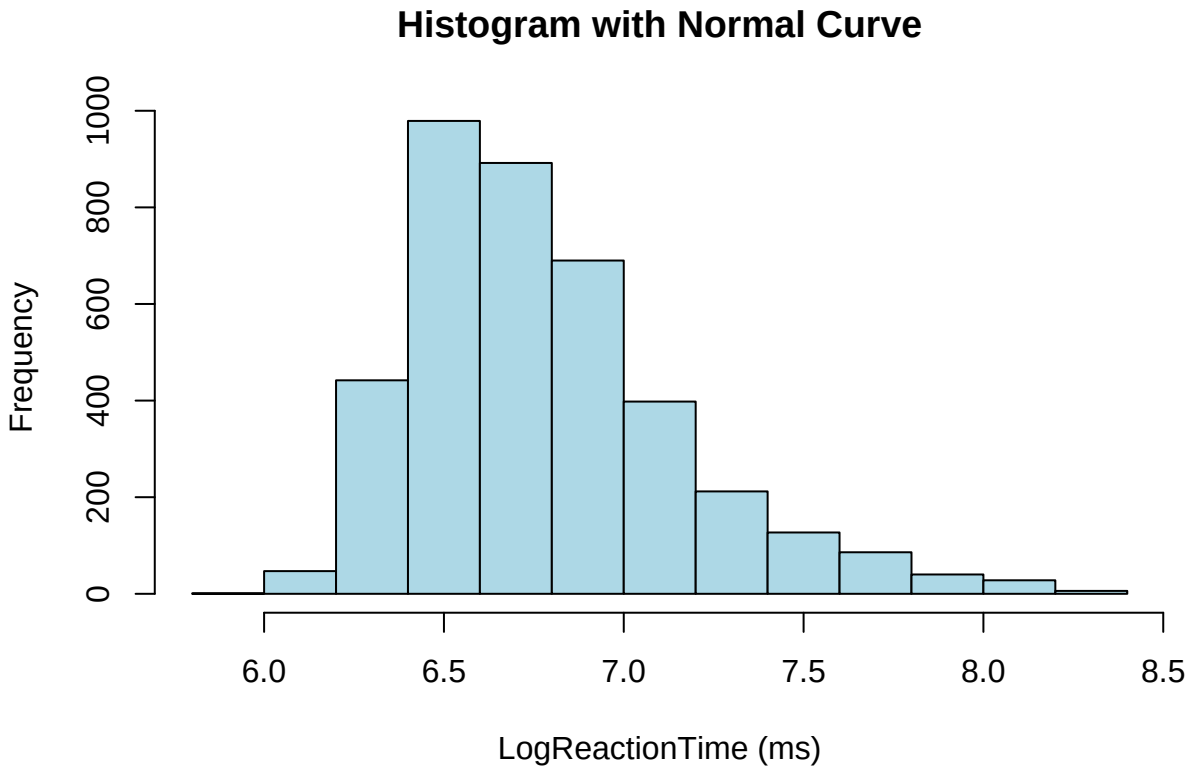

```
quantile(d.inaccurate.removed$LogReactionTime)
```

```
##       0%      25%      50%      75%     100%
## 5.934894 6.513230 6.712956 6.973543 9.878836
```

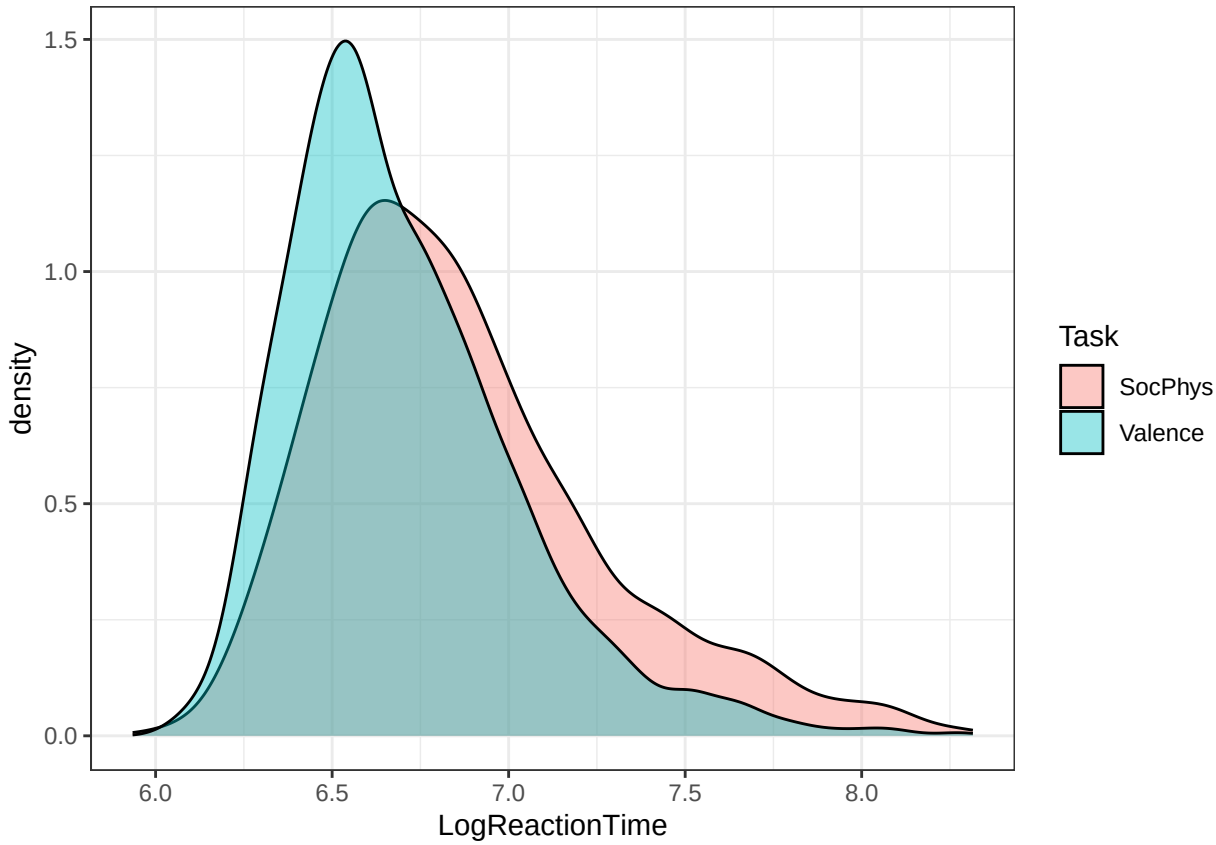
```
IQR(d.inaccurate.removed$LogReactionTime)*3 # 0.7526289
```

```
## [1] 1.380939
```

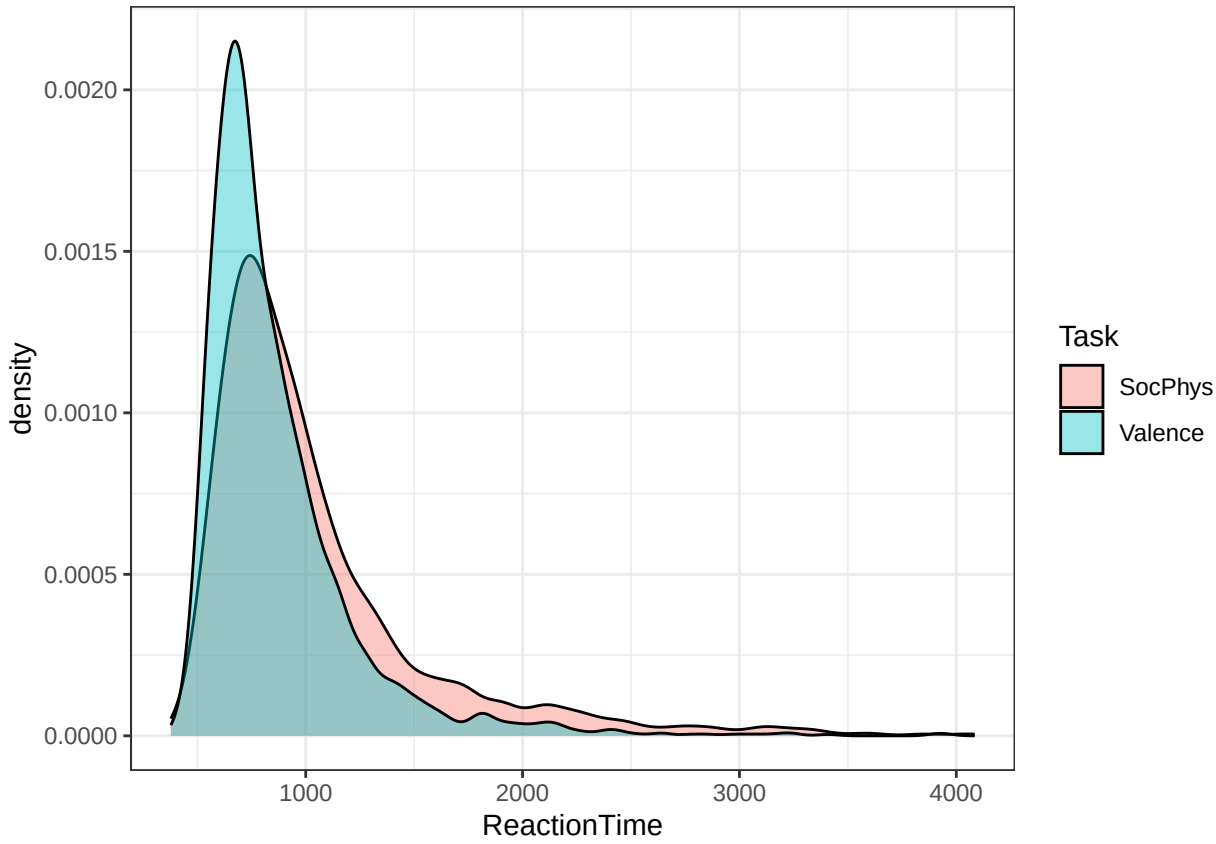
```r
cutoff.high <- quantile(d.inaccurate.removed$LogReactionTime)[4] + IQR(d.inaccurate.removed$LogReaction
cutoff.low <- quantile(d.inaccurate.removed$LogReactionTime)[2] - IQR(d.inaccurate.removed$LogReactionT


# 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")
```




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




```
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: 2142.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5676 -0.6650 -0.1965  0.4518  5.5572
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  Word     (Intercept) 0.0008998 0.0300
##           cTask       0.0112645 0.1061   0.02
##  ID.true  (Intercept) 0.0297162 0.1724
##           cTask       0.0286424 0.1692   -0.52
```

```
##  Residual             0.0955607 0.3091
## Number of obs: 3948, groups:  Word, 40; ID.true, 18
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.78252    0.04121 17.44882 164.596  < 2e-16 ***
## cTask       -0.17054    0.04440 22.65637  -3.841 0.000852 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##       (Intr)
## cTask -0.464
```

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

Y.

```
m = lmer(LogReactionTime ~ cTask*ConcValCombo + (1+ConcValCombo+cTask|ID.true) + (1+cTask|Word), data=c
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: 2138.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7266 -0.6625 -0.1913  0.4490  5.3862
##
## Random effects:
##  Groups   Name                           Variance  Std.Dev. Corr
##  Word     (Intercept)                    0.0008797 0.02966
##           cTask                          0.0101026 0.10051  0.06
##  ID.true  (Intercept)                    0.0283987 0.16852
##           ConcValCombophysical-positive 0.0020979 0.04580   0.23
##           ConcValCombosocial-negative   0.0050846 0.07131  -0.32  0.58
##           ConcValCombosocial-positive   0.0027056 0.05202   0.23  0.92  0.43
##           cTask                          0.0288195 0.16976  -0.42 -0.58 -0.15
##  Residual                                0.0943896 0.30723
##
##
##
##
##
##
##
```

```
##  -0.82
##
## Number of obs: 3948, groups:  Word, 40; ID.true, 18
##
## Fixed effects:
##                                    Estimate Std. Error        df t value
## (Intercept)                         6.767037   0.041988 18.649341 161.165
## cTask                              -0.135267   0.054792 39.965583  -2.469
## ConcValCombophysical-positive       0.028351   0.022375 25.763470   1.267
## ConcValCombosocial-negative         0.030759   0.025455 24.920522   1.208
## ConcValCombosocial-positive         0.005350   0.022676 26.475270   0.236
## cTask:ConcValCombophysical-positive -0.121075   0.053467 36.110144  -2.264
## cTask:ConcValCombosocial-negative    0.008211   0.052714 34.251775   0.156
## cTask:ConcValCombosocial-positive   -0.034412   0.052662 34.119674  -0.653
##                                    Pr(>|t|)
## (Intercept)                          <2e-16 ***
## cTask                                0.0179 *
## ConcValCombophysical-positive        0.2165
## ConcValCombosocial-negative          0.2382
## ConcValCombosocial-positive          0.8153
## cTask:ConcValCombophysical-positive  0.0296 *
## cTask:ConcValCombosocial-negative    0.8771
## cTask:ConcValCombosocial-positive    0.5178
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                    (Intr) cTask  CncVlCmbp- CncVlCmbscl-n CncVlCmbscl-p
## cTask              -0.281
## CncVlCmbph-        -0.090 -0.216
## CncVlCmbscl-n      -0.372 -0.084  0.510
## CncVlCmbscl-p      -0.076 -0.338  0.605      0.474
## cTsk:CncVlCmbp-    -0.007 -0.478  0.003      0.011         0.012
## cTsk:CncVlCmbscl-n -0.007 -0.485  0.013      0.032         0.012
## cTsk:CncVlCmbscl-p -0.007 -0.486  0.013      0.011         0.033
##                    cTsk:CncVlCmbp- cTsk:CncVlCmbscl-n
## cTask
## CncVlCmbph-
## CncVlCmbscl-n
## CncVlCmbscl-p
## cTsk:CncVlCmbp-
## cTsk:CncVlCmbscl-n  0.497
## cTsk:CncVlCmbscl-p  0.497           0.505
```

## 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: 2653.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4128 -0.6682 -0.2134  0.4816  5.1685
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  Word     (Intercept) 0.0007276 0.02697
##  ID.true  (Intercept) 0.0293163 0.17122
##  Residual             0.1119159 0.33454
## Number of obs: 3948, groups:  Word, 40; ID.true, 18
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.77858    0.04093 17.36051   165.6   <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)
```

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

```
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: 2656.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4144 -0.6688 -0.2135  0.4808  5.1629
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  Word     (Intercept) 7.278e-04 0.026977
##           cBlockOrder 1.991e-06 0.001411 1.00
##  ID.true  (Intercept) 3.029e-02 0.174041
##  Residual             1.119e-01 0.334537
```

```
## Number of obs: 3948, groups:  Word, 40; ID.true, 18
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.77872    0.04159 16.32917 163.000   <2e-16 ***
## cBlockOrder  0.05630    0.08273 15.99183   0.681    0.506
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cBlockOrder 0.005
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

## effect of ConcValCombo on ReactionTime?

nope.

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

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: 2656.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.3384 -0.6585 -0.2114  0.4753  5.0200
##
## Random effects:
##  Groups   Name                           Variance  Std.Dev. Corr
##  Word     (Intercept)                    0.0006815 0.02611
##  ID.true  (Intercept)                    0.0262184 0.16192
##           ConcValCombophysical-positive 0.0017386 0.04170   0.39
##           ConcValCombosocial-negative   0.0036798 0.06066  -0.20  0.68
##           ConcValCombosocial-positive   0.0021765 0.04665   0.46  0.79  0.32
##  Residual                               0.1110397 0.33323
## Number of obs: 3948, groups:  Word, 40; ID.true, 18
##
## Fixed effects:
##                                Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)                    6.764743   0.040482 18.306904 167.105   <2e-16
## ConcValCombophysical-positive  0.008502   0.021714 21.055141   0.392    0.699
## ConcValCombosocial-negative    0.035173   0.023723 22.160801   1.483    0.152
## ConcValCombosocial-positive    0.010681   0.021854 21.568688   0.489    0.630
##
## (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-   -0.040
## CncVlCmbscl-n -0.303  0.538
## CncVlCmbscl-p  0.013  0.564      0.449
```

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

- Nope

```
str(df_factors)
```

```
## 'data.frame':    3948 obs. of  20 variables:
##  $ X              : Factor w/ 3948 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ ID.true        : Factor w/ 18 levels "5b0b77a9d282ff00015b7531",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ Word           : Factor w/ 40 levels "amiable","arrogant",..: 25 9 1 23 30 2 21 7 13 31 ...
##  $ Label          : Factor w/ 2 levels "test_sp","test_val": 2 2 2 2 2 2 2 2 2 2 ...
##  $ ConcValCombo   : Factor w/ 4 levels "physical-negative",..: 1 4 4 4 2 3 3 3 1 1 ...
##  $ Task           : Factor w/ 2 levels "SocPhys","Valence": 2 2 2 2 2 2 2 2 2 2 ...
##  $ BlockOrder     : Factor w/ 2 levels "SV","VS": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Group          : Factor w/ 4 levels "negative;positive",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Response       : Factor w/ 4 levels "negative","physical",..: 1 3 3 3 3 1 1 1 1 1 ...
##  $ Accuracy       : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ EventTime      : Factor w/ 3946 levels "1744715370410",..: 2173 2174 2175 2176 2177 2178 2179 218
##  $ Value          : Factor w/ 4 levels "negative","physical",..: 1 3 3 3 3 1 1 1 1 1 ...
##  $ RT             : Factor w/ 667 levels "1306.33333333333",..: 387 199 234 202 127 494 518 228 248 1
##  $ ReactionTime   : int  1010 887 934 704 824 1135 979 697 640 753 ...
##  $ Key_value_F    : Factor w/ 4 levels "negative","physical",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Key_value_J    : Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Comments       : Factor w/ 0 levels: NA NA NA NA NA NA NA NA NA NA ...
##  $ LogReactionTime: num  6.92 6.79 6.84 6.56 6.71 ...
##  $ LogRT          : Factor w/ 667 levels "7.17497950953152",..: 387 199 234 202 127 494 518 228 248 1
##  $ TrialNumber    : Factor w/ 240 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
```

```
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')
```

```
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: 1283.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5883 -0.6754 -0.1967  0.4788  4.4846
##
## Random effects:
##  Groups   Name          Variance Std.Dev. Corr
##  Word     (Intercept)   0.003000 0.05477
##           cConcValCombo 0.001187 0.03445  -1.00
##  ID.true  (Intercept)   0.053482 0.23126
##           cConcValCombo 0.001258 0.03547  0.14
##  Residual               0.107778 0.32830
## Number of obs: 1889, groups:  Word, 40; ID.true, 18
##
## Fixed effects:
##                Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)    6.874472   0.055956 18.128276 122.855   <2e-16 ***
## cConcValCombo -0.008629   0.013868 31.524268  -0.622    0.538
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cConcValCmb -0.021
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
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: 1290.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.3636 -0.6748 -0.1936  0.4985  4.5131
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.003701 0.06083
```

```
##  ID.true  (Intercept) 0.053459 0.23121
##           cSemantic   0.005461 0.07390  -0.05
##  Residual             0.108194 0.32893
## Number of obs: 1889, groups:  Word, 40; ID.true, 18
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.87125    0.05588 18.03749 122.970   <2e-16 ***
## cSemantic   -0.01956    0.03022 28.70732  -0.647    0.523
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## cSemantic -0.020
```

## 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')
```

```
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: 829.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1663 -0.6585 -0.2015  0.4219  5.9262
##
## Random effects:
##  Groups   Name          Variance  Std.Dev. Corr
##  Word     (Intercept)   3.853e-03 0.062069
##           cConcValCombo 6.167e-05 0.007853 -1.00
```

```
##  ID.true  (Intercept)   2.177e-02 0.147555
##           cConcValCombo 1.162e-04 0.010778 1.00
##  Residual               8.265e-02 0.287488
## Number of obs: 2059, groups:  Word, 40; ID.true, 18
##
## Fixed effects:
##               Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)    6.701032   0.036714 19.672712 182.518   <2e-16 ***
## cConcValCombo  0.001924   0.010737 38.822190   0.179    0.859
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cConcValCmb 0.164
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
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: 823.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1120 -0.6473 -0.2032  0.4314  5.8552
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.003530 0.05942
##  ID.true  (Intercept) 0.021789 0.14761
##           cValence    0.002123 0.04607  0.62
##  Residual             0.082288 0.28686
## Number of obs: 2059, groups:  Word, 40; ID.true, 18
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.70077    0.03659 19.39301 183.130   <2e-16 ***
## cValence    -0.03805    0.02513 34.48392  -1.514    0.139
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## cValence 0.254
```