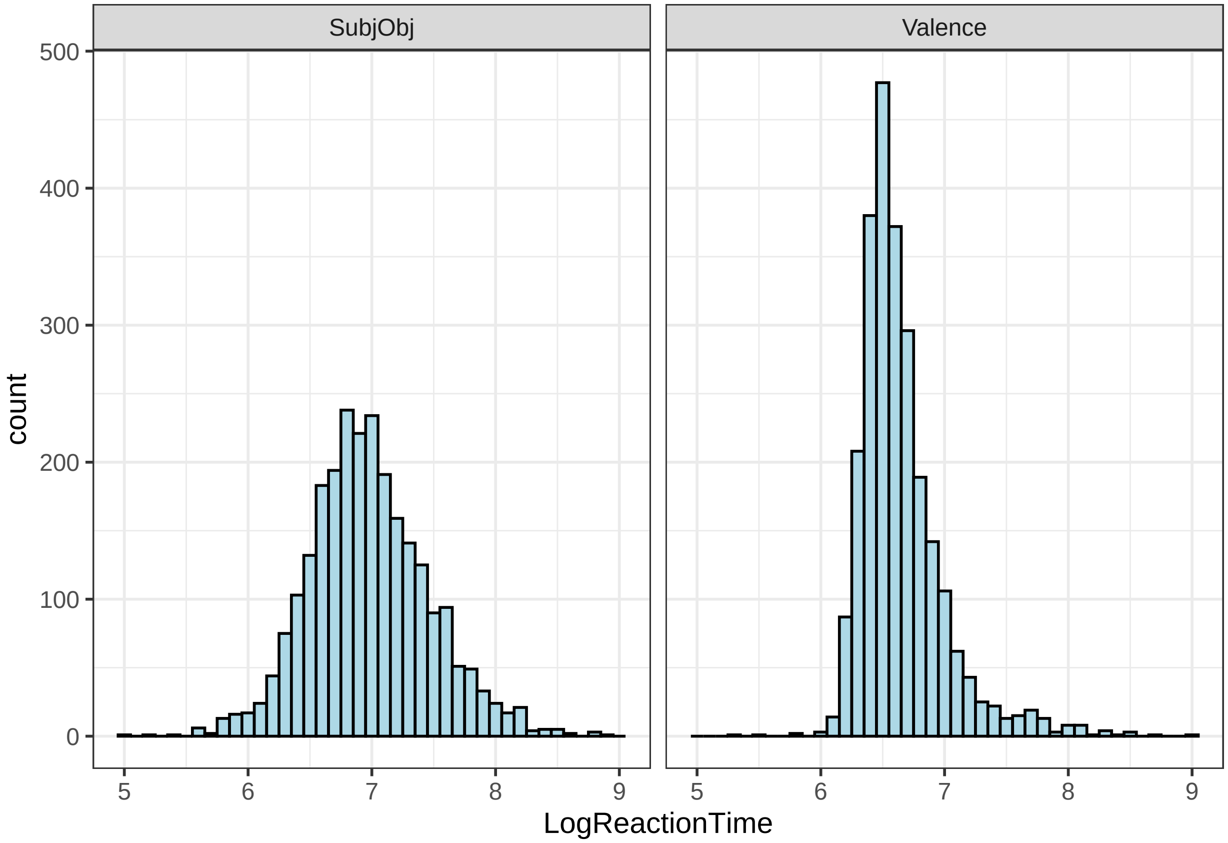
Adjs Subj-Obj: Analysis

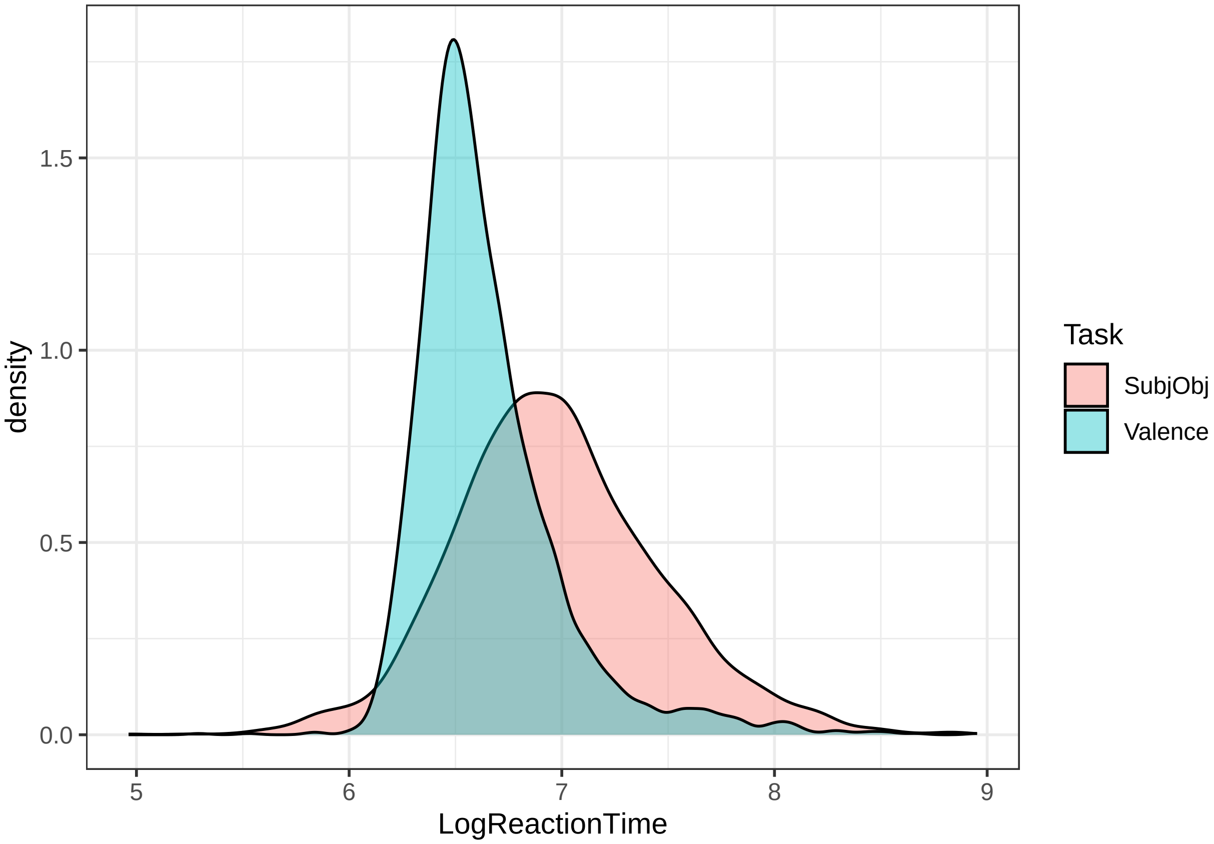
morgan moyer

2025-04-11

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



```r
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)
```

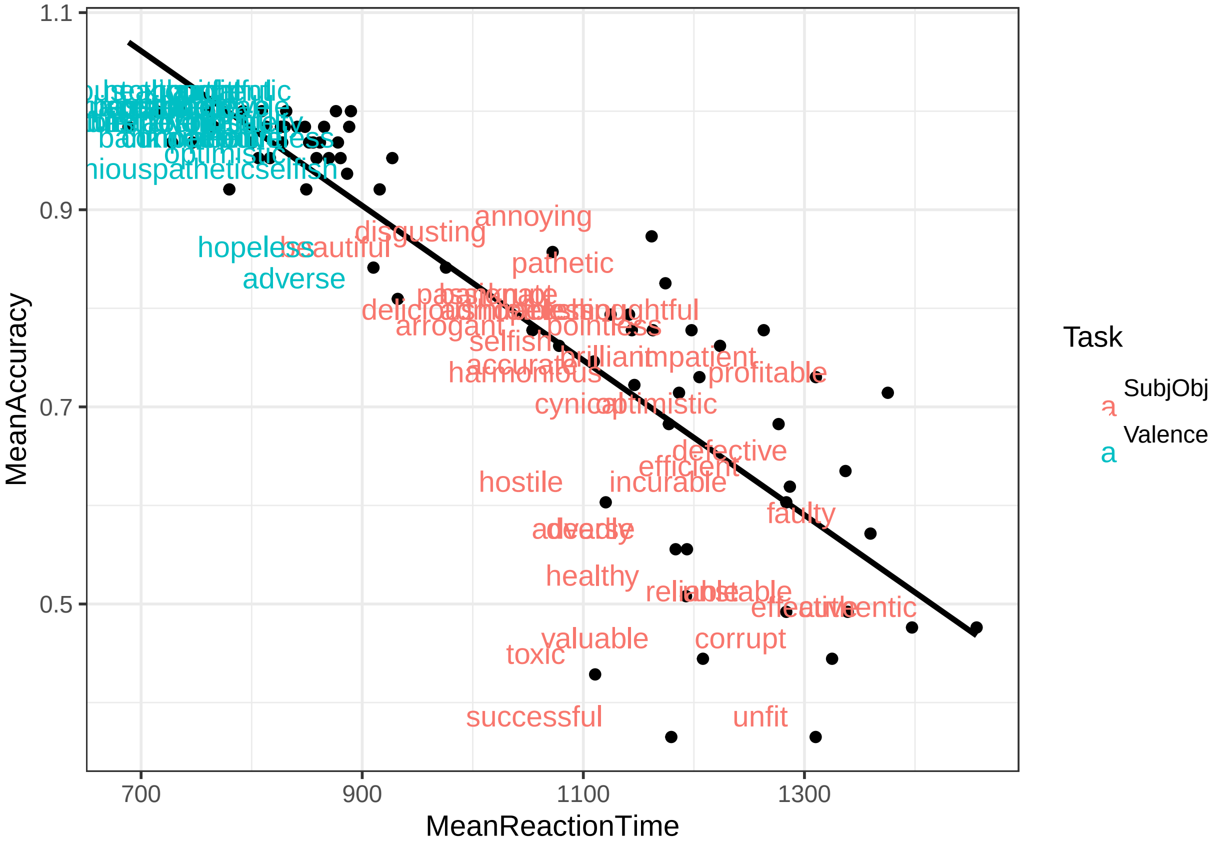
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cAccuracy * cTask + (1 + cTask | Word) + (1 +
##     cTask | ID.true)
##    Data: dcen
##
## REML criterion at convergence: 3512.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.3442 -0.5737 -0.1232  0.4378  6.3695
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  Word     (Intercept) 0.0022361 0.04729
##           cTask       0.0008611 0.02935  -0.16
##  ID.true  (Intercept) 0.0411384 0.20283
##           cTask       0.0971860 0.31175  -0.40
##  Residual             0.1119540 0.33460
## Number of obs: 5040, groups:  Word, 38; ID.true, 21
##
## Fixed effects:
##                   Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)      6.804e+00  4.529e-02  2.141e+01 150.256  < 2e-16 ***
## cAccuracy       -4.802e-03  2.057e-02  4.657e+03  -0.233 0.815426
## cTask           -3.282e-01  6.915e-02  2.053e+01  -4.746 0.000116 ***
## cAccuracy:cTask  1.049e-01  4.124e-02  4.810e+03   2.545 0.010973 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cAccrc cTask
## cAccuracy   -0.052
## cTask       -0.380 -0.093
## cAccrcy:cTs -0.071  0.731 -0.068
```

```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 == "SubjObj") %>%
  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))
```

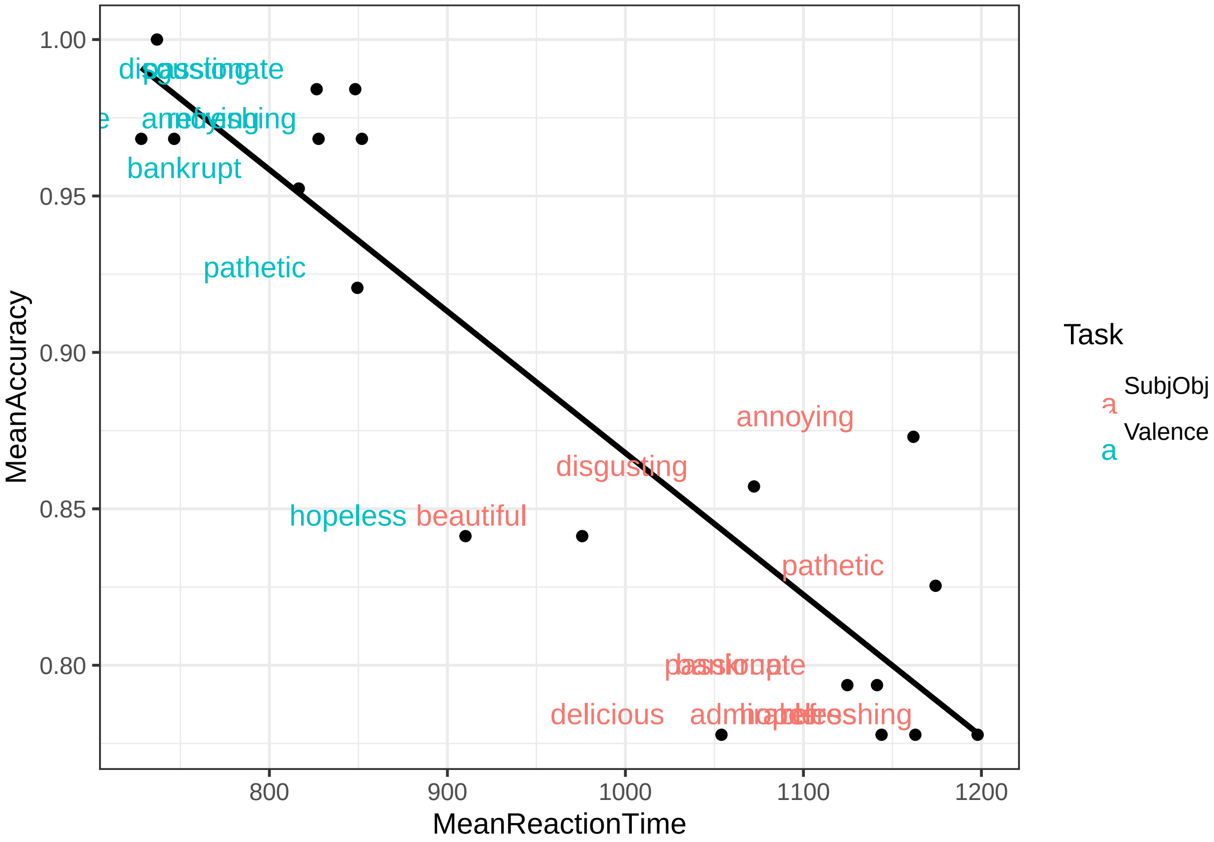
```
## `summarise()` has grouped output by 'Word'. You can override using the
## `.groups` argument.
```

```
print(agr)
```

```
## # A tibble: 20 x 4
## # Groups:   Word [10]
##    Word       Task    MeanAccuracy MeanReactionTime
##    <chr>      <chr>          <dbl>            <dbl>
##  1 admirable  SubjObj        0.778            1144.
##  2 admirable  Valence        0.968             747.
##  3 annoying   SubjObj        0.873            1162.
##  4 annoying   Valence        0.968             828.
##  5 bankrupt   SubjObj        0.794            1125.
##  6 bankrupt   Valence        0.952             816.
##  7 beautiful  SubjObj        0.841             976.
##  8 beautiful  Valence        0.968             728.
##  9 delicious  SubjObj        0.778            1054.
## 10 delicious  Valence        1                 737.
## 11 disgusting SubjObj        0.857            1072.
## 12 disgusting Valence        0.984             827.
## 13 hopeless   SubjObj        0.778            1163.
## 14 hopeless   Valence        0.841             910.
## 15 passionate SubjObj        0.794            1141.
## 16 passionate Valence        0.984             848.
## 17 pathetic   SubjObj        0.825            1174.
## 18 pathetic   Valence        0.921             849.
## 19 refreshing SubjObj        0.778            1198.
## 20 refreshing Valence        0.968             852.
```

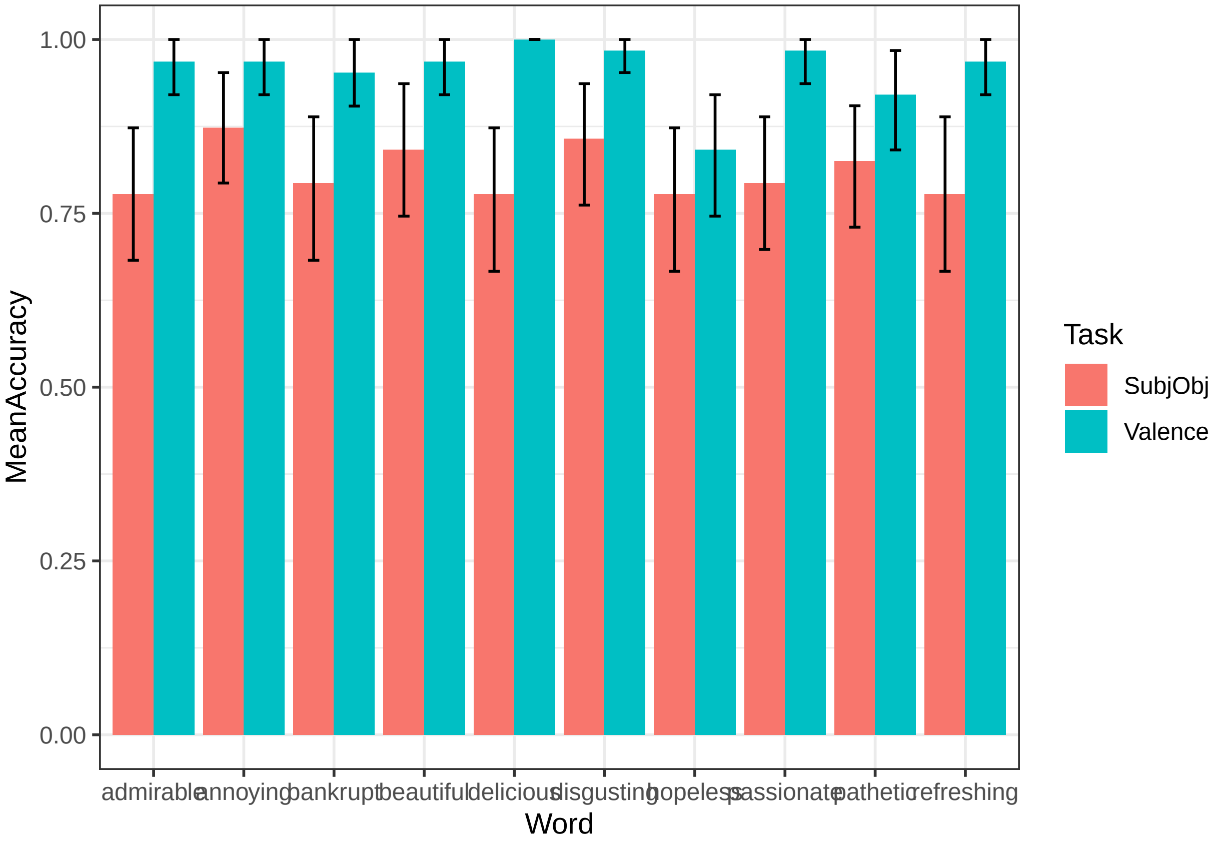
```
ggplot(agr, aes(x = MeanReactionTime, y = MeanAccuracy)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  geom_text(aes(label = Word, color = Task), vjust = -0.5, hjust = 1.5)
```

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

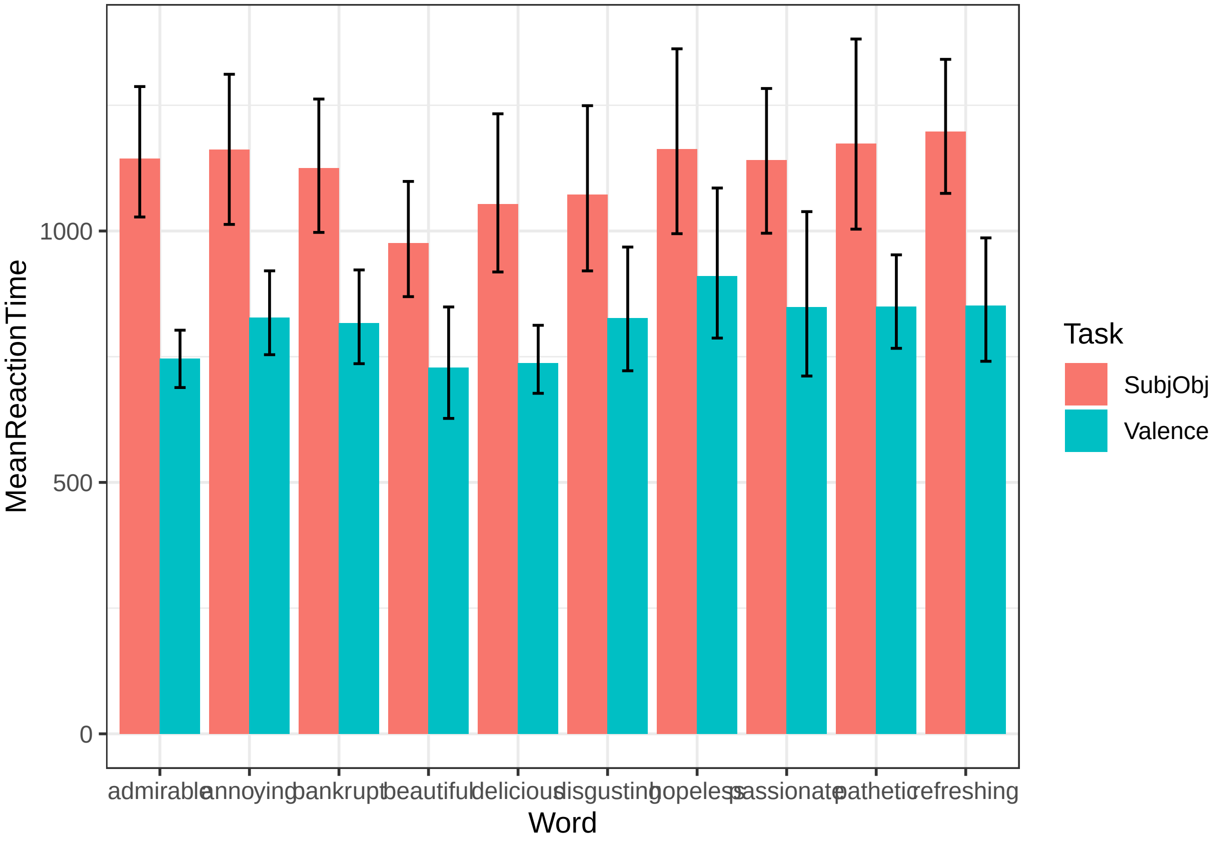


```
  # guides(legend = "none")
  # theme(legend.position = "none")  # Remove the legend
# ggsave("../graphs/exp1b_accXrt.pdf",width = 5, height = 3)
```

```
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] 21
```

```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] 14
```

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

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

```
## [1] 7
```

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] 88.45238
```

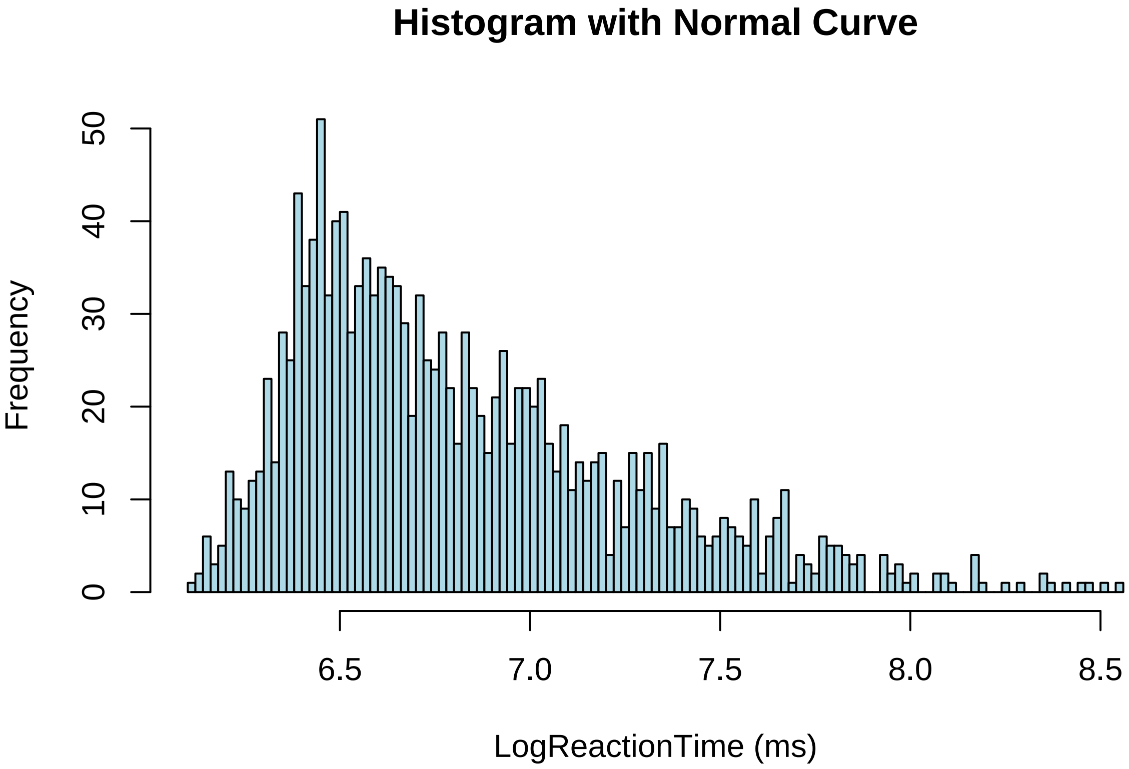
```
# Remove subjects with ReactionTime higher than 3x IQR
summary(d.inaccurate.removed$LogReactionTime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   6.118   6.488   6.714   6.820   7.054   8.560
```

```
  #  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] 6.118097 8.559678
```

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



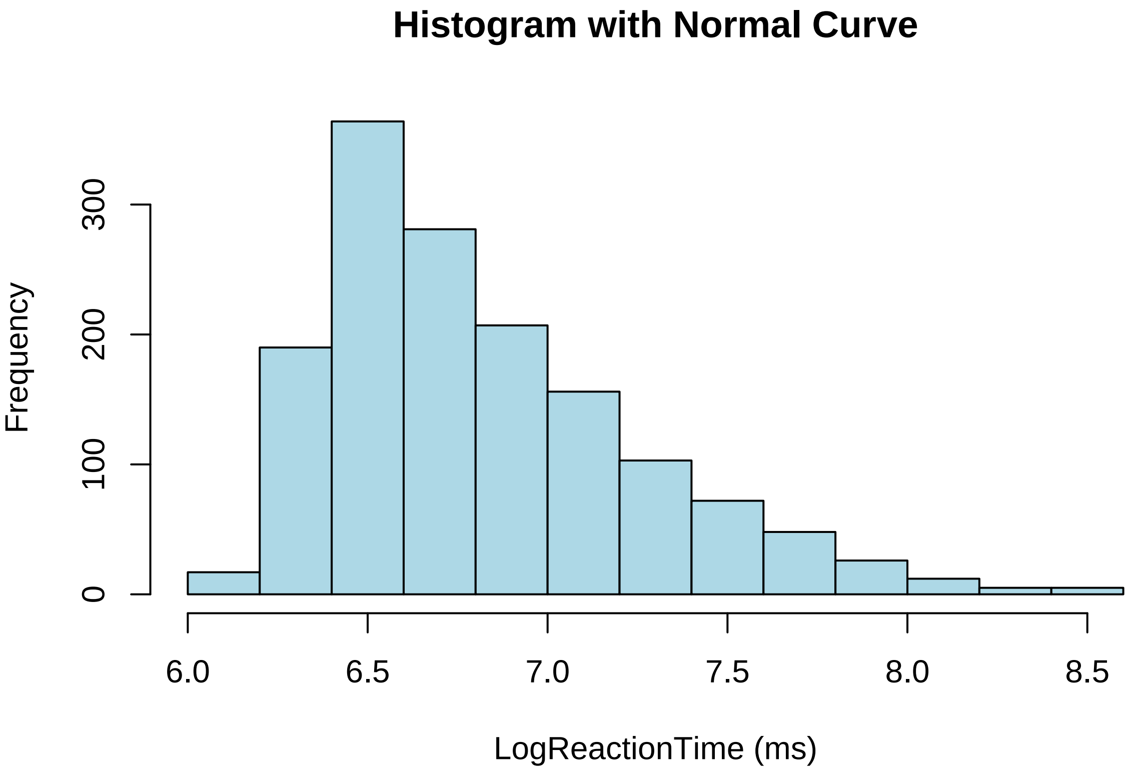
```
quantile(d.inaccurate.removed$LogReactionTime)
```

```
##       0%      25%      50%      75%     100%
## 6.118097 6.488064 6.714171 7.053801 8.559678
```

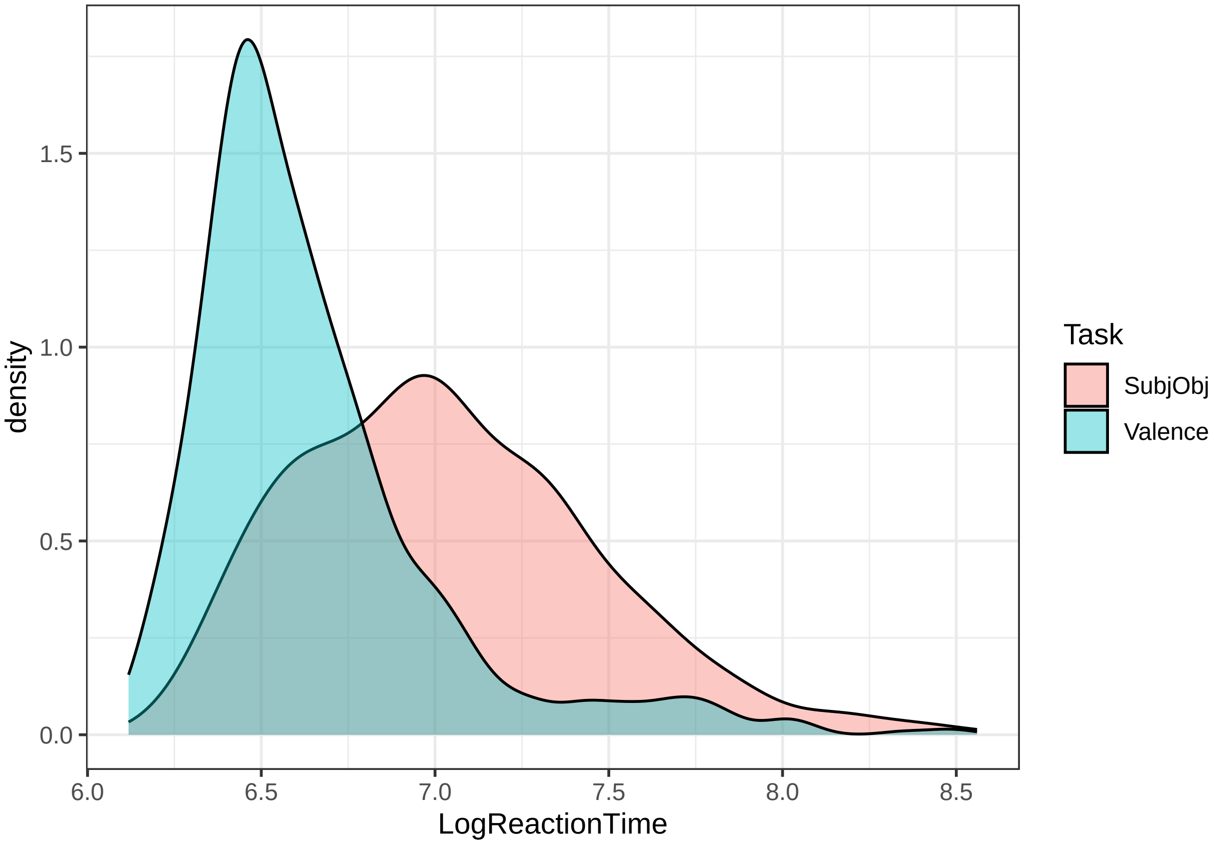
```
IQR(d.inaccurate.removed$LogReactionTime)*3 # 0.7526289
```

```
## [1] 1.697211
```

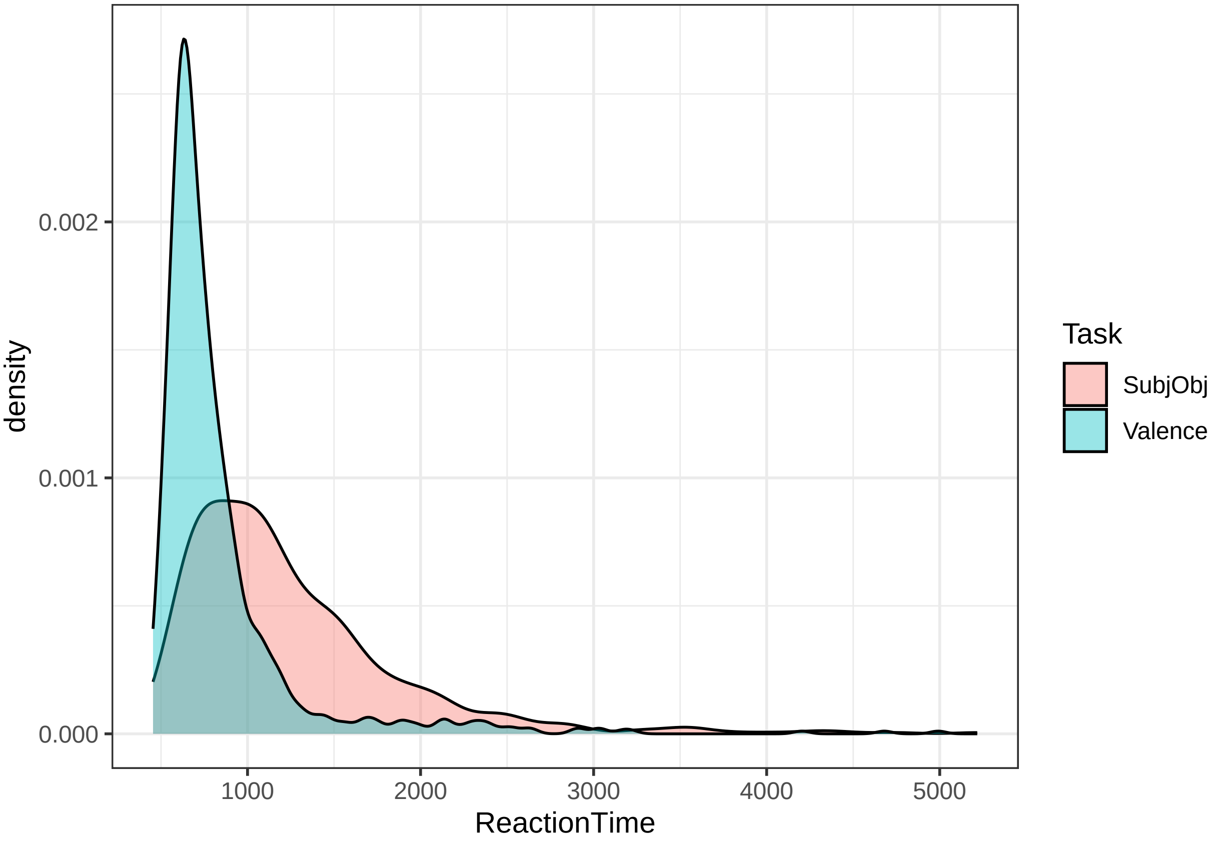
```r
cutoff.high <- quantile(d.inaccurate.removed$LogReactionTime)[4] + IQR(d.inaccurate.removed$LogReaction
cutoff.low <- quantile(d.inaccurate.removed$LogReactionTime)[2] - IQR(d.inaccurate.removed$LogReactionTi


# 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")
```



```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: 895.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1375 -0.5900 -0.1353  0.4659  4.9930
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.003177 0.05637
##           cTask       0.009295 0.09641  -0.82
##  ID.true  (Intercept) 0.044034 0.20984
##           cTask       0.055418 0.23541  -0.03
##  Residual             0.099644 0.31567
## Number of obs: 1486, groups:  Word, 38; ID.true, 7
##
## Fixed effects:
##              Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.82359    0.08027  6.16001  85.003 1.08e-10 ***
## cTask       -0.39395    0.09190  6.36852  -4.287  0.00451 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##       (Intr)
## cTask -0.042
```

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
```

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

```
## Warning: Model failed to converge with 1 negative eigenvalue: -2.7e-01
```

```
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: 868
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1047 -0.5888 -0.1237  0.4456  4.9790
##
## Random effects:
##  Groups   Name                            Variance  Std.Dev. Corr
##  Word     (Intercept)                     0.0026122 0.05111
##           cTask                           0.0031233 0.05589  -1.00
##  ID.true  (Intercept)                     0.0394745 0.19868
##           ConcValComboobjective-positive  0.0009007 0.03001  -0.28
##           ConcValCombosubjective-negative 0.0064645 0.08040   0.20 -1.00
##           ConcValCombosubjective-positive 0.0048498 0.06964   0.49 -0.08  0.04
##           cTask                           0.0579737 0.24078  -0.12 -0.77  0.79
##  Residual                                 0.0966869 0.31095
##
##
##
```

```
##
##
##
##
##   0.36
##
## Number of obs: 1486, groups:  Word, 38; ID.true, 7
##
## Fixed effects:
##                                         Estimate Std. Error        df t value
## (Intercept)                             6.859489   0.078756  6.515303  87.098
## cTask                                  -0.476206   0.099296  7.926117  -4.796
## ConcValComboobjective-positive         -0.009316   0.035985 24.202485  -0.259
## ConcValCombosubjective-negative        -0.036331   0.044726 11.340418  -0.812
## ConcValCombosubjective-positive        -0.084107   0.042149 13.647731  -1.995
## cTask:ConcValComboobjective-positive   -0.009771   0.055144 50.395193  -0.177
## cTask:ConcValCombosubjective-negative   0.227335   0.053859 61.221294   4.221
## cTask:ConcValCombosubjective-positive   0.085446   0.054205 62.497536   1.576
##                                        Pr(>|t|)
## (Intercept)                            3.08e-11 ***
## cTask                                    0.0014 **
## ConcValComboobjective-positive           0.7979
## ConcValCombosubjective-negative          0.4333
## ConcValCombosubjective-positive          0.0664 .
## cTask:ConcValComboobjective-positive     0.8601
## cTask:ConcValCombosubjective-negative  8.19e-05 ***
## cTask:ConcValCombosubjective-positive    0.1200
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                      (Intr) cTask  CncVlCmbb- CncVlCmbsbjctv-n
## cTask                -0.155
## CncVlCmbbj-          -0.284 -0.117
## CncVlCmbsbjctv-n     -0.029  0.578  0.137
## CncVlCmbsbjctv-p      0.121  0.294  0.355      0.317
## cTsk:CncVlCmbb-       0.086 -0.288 -0.381     -0.152
## cTsk:CncVlCmbsbjctv-n  0.088 -0.296 -0.194     -0.264
## cTsk:CncVlCmbsbjctv-p  0.087 -0.293 -0.191     -0.155
##                      CncVlCmbsbjctv-p cTsk:CncVlCmbb- cTsk:CncVlCmbsbjctv-n
## cTask
## CncVlCmbbj-
## CncVlCmbsbjctv-n
## CncVlCmbsbjctv-p
## cTsk:CncVlCmbb-       -0.160
## cTsk:CncVlCmbsbjctv-n -0.165           0.533
## cTsk:CncVlCmbsbjctv-p -0.288           0.528           0.542
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

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: 1460.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0294 -0.6807 -0.2167  0.4965  4.0971
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Word     (Intercept) 0.00146  0.03821
##  ID.true  (Intercept) 0.04345  0.20844
##  Residual             0.15205  0.38993
## Number of obs: 1486, groups:  Word, 38; ID.true, 7
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  6.81939    0.07968 6.07311   85.58 1.36e-10 ***
## ---
## 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')
```

```
## Warning: Model failed to converge with 1 negative eigenvalue: -1.2e+01
```

```
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: 1461.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0760 -0.6932 -0.2181  0.4952  4.1305
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.001480 0.03848
##           cBlockOrder 0.000836 0.02891  -1.00
##  ID.true  (Intercept) 0.050245 0.22415
##  Residual             0.151832 0.38966
## Number of obs: 1486, groups:  Word, 38; ID.true, 7
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  6.81869    0.08557 5.05344  79.681 4.98e-09 ***
## cBlockOrder  0.07832    0.17249 5.00634   0.454    0.669
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cBlockOrder -0.021
## 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)
```

```
## 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: 1462.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.2069 -0.6592 -0.2165  0.4858  4.2569
##
## Random effects:
##  Groups   Name                             Variance Std.Dev. Corr
##  Word     (Intercept)                      0.001361 0.03689
##  ID.true  (Intercept)                      0.036380 0.19073
##           ConcValComboobjective-positive   0.002682 0.05179  -0.18
##           ConcValCombosubjective-negative 0.001394 0.03734   1.00 -0.22
##           ConcValCombosubjective-positive 0.009402 0.09696   0.26  0.90  0.22
##  Residual                                  0.150311 0.38770
## Number of obs: 1486, groups:  Word, 38; ID.true, 7
##
```

```
## Fixed effects:
##                                Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)                    6.827027   0.076064  6.240176  89.753 6.02e-11
## ConcValComboobjective-positive  0.014105   0.039428 11.681069   0.358    0.727
## ConcValCombosubjective-negative  0.004816   0.036215 16.728397   0.133    0.896
## ConcValCombosubjective-positive -0.047684   0.049663  7.629939  -0.960    0.366
##
## (Intercept)                    ***
## ConcValComboobjective-positive
## ConcValCombosubjective-negative
## ConcValCombosubjective-positive
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                  (Intr) CncVlCmbb- CncVlCmbsbjctv-n
## CncVlCmbbj-      -0.282
## CncVlCmbsbjctv-n  0.155  0.370
## CncVlCmbsbjctv-p  0.026  0.631      0.391
## 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

```
str(df_factors)
```

```
## 'data.frame':    1486 obs. of  20 variables:
##  $ X              : Factor w/ 1486 levels "243","245","247",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ ID.true        : Factor w/ 7 levels "58060d08b869700001d70b8c",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ Word           : Factor w/ 38 levels "accurate","admirable",..: 25 3 23 37 1 19 1 38 12 29 ...
##  $ Label          : Factor w/ 2 levels "test_so","test_val": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ConcValCombo   : Factor w/ 4 levels "objective-negative",..: 4 1 3 1 2 4 2 2 1 2 ...
##  $ Task           : Factor w/ 2 levels "SubjObj","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/ 4 levels "negative;positive",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Response       : Factor w/ 4 levels "negative","objective",..: 4 2 4 2 2 4 2 2 2 2 ...
##  $ Accuracy       : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ EventTime      : Factor w/ 1486 levels "1744114925092",..: 834 835 836 837 838 839 840 841 842 84
##  $ Value          : Factor w/ 4 levels "negative","objective",..: 4 2 4 2 2 4 2 2 2 2 ...
##  $ RT             : Factor w/ 262 levels "1325.5","1350.16666666667",..: 144 254 200 199 164 167 164
##  $ ReactionTime   : int  1290 1322 1464 1237 1333 1428 1161 1294 1376 1475 ...
##  $ Key_value_F    : Factor w/ 4 levels "negative","objective",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Key_value_J    : Factor w/ 2 levels "A","B": 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  7.16 7.19 7.29 7.12 7.2 ...
##  $ LogRT          : Factor w/ 262 levels "7.18954502572908",..: 144 254 200 199 164 167 164 119 153
##  $ TrialNumber    : Factor w/ 240 levels "1","2","3","4",..: 3 5 7 8 9 10 14 16 17 18 ...
```

```
sem <- df_factors %>%
  filter(Task == "SubjObj") %>%
  mutate(
         Semantic = ifelse(grepl("objective", ConcValCombo), "objective",
                    ifelse(grepl("subjective", ConcValCombo), "subjective", 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
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: 549.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6549 -0.6466 -0.1275  0.6102  3.0393
##
## Random effects:
##  Groups   Name          Variance  Std.Dev. Corr
##  Word     (Intercept)   0.0056362 0.07507
##           cConcValCombo 0.0001315 0.01147  1.00
##  ID.true  (Intercept)   0.0619195 0.24884
##           cConcValCombo 0.0012960 0.03600  0.57
##  Residual               0.1227098 0.35030
## Number of obs: 663, groups:  Word, 38; ID.true, 7
##
## Fixed effects:
##               Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)    7.03886    0.09590  6.19985  73.397 2.38e-10 ***
## cConcValCombo -0.05454    0.02169  9.98408  -2.514   0.0307 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cConcValCmb 0.370
```

```
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: 541.4
##
```

```
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6859 -0.6426 -0.1104  0.6168  2.9025
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.00460  0.06782
##  ID.true  (Intercept) 0.06449  0.25394
##           cSemantic   0.01195  0.10929  0.05
##  Residual             0.12126  0.34822
## Number of obs: 663, groups:  Word, 38; ID.true, 7
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  7.04263    0.09761  6.15320  72.150 3.03e-10 ***
## cSemantic   -0.14700    0.05462  8.26563  -2.692   0.0266 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## cSemantic 0.039
```

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: -6.5e+00
```

```
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: 316.2
##
```

```
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4277 -0.5600 -0.1492  0.3502  5.4287
##
## Random effects:
##  Groups   Name          Variance  Std.Dev. Corr
##  Word     (Intercept)   0.000e+00 0.000000
##           cConcValCombo 1.270e-03 0.035631  NaN
##  ID.true  (Intercept)   5.408e-02 0.232547
##           cConcValCombo 3.325e-05 0.005766 1.00
##  Residual               8.082e-02 0.284290
## Number of obs: 823, groups:  Word, 38; ID.true, 7
##
## Fixed effects:
##                Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)    6.653358   0.088640  6.050709  75.060 3.23e-10 ***
## cConcValCombo -0.001197   0.011545 16.970367  -0.104    0.919
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cConcValCmb 0.187
## 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: 303.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4055 -0.5552 -0.1591  0.3406  5.5898
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.001183 0.03439
##  ID.true  (Intercept) 0.053983 0.23234
##           cValence    0.003359 0.05795  -0.38
##  Residual             0.079446 0.28186
## Number of obs: 823, groups:  Word, 38; ID.true, 7
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.64760    0.08855  6.04899   75.07 3.25e-10 ***
## cValence    -0.06465    0.03153  7.33607   -2.05   0.0777 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation of Fixed Effects:
##           (Intr)
## cValence -0.258
```