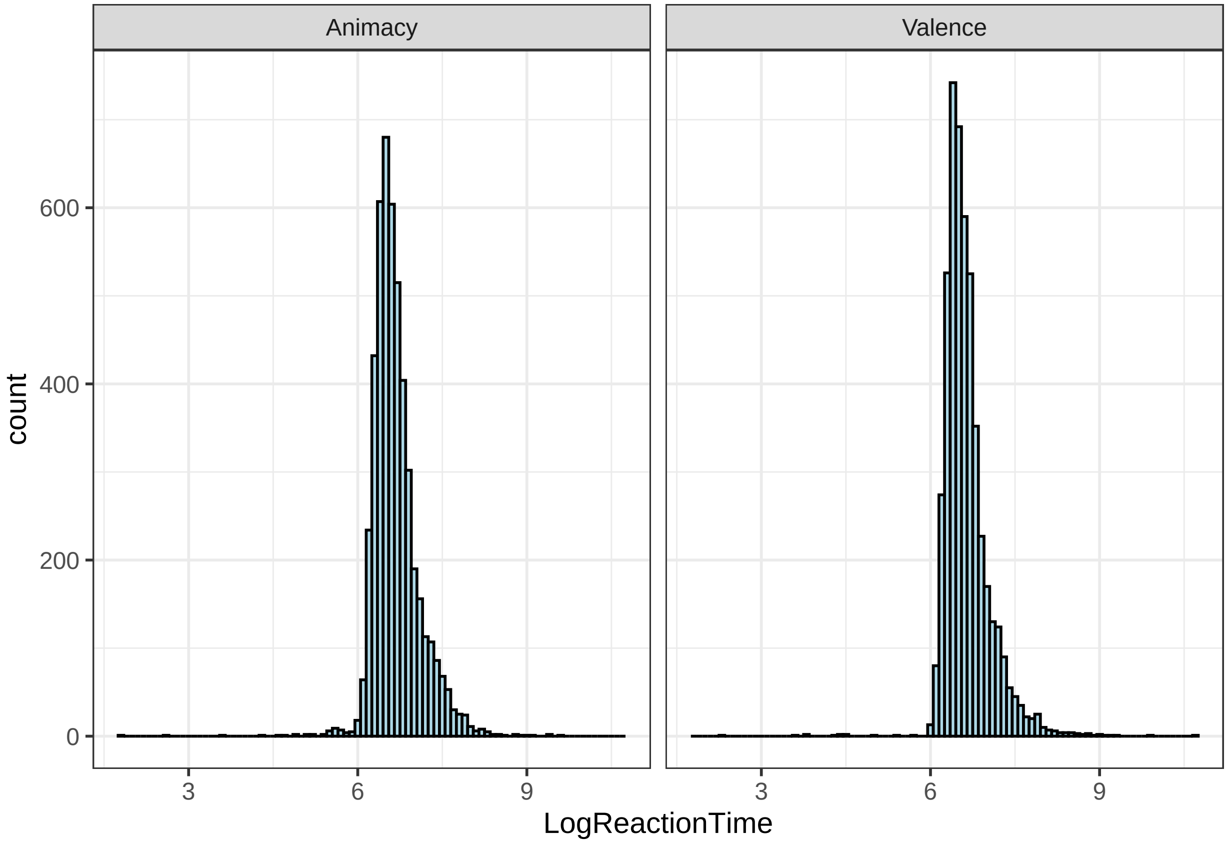
Animacy Nouns: Analysis

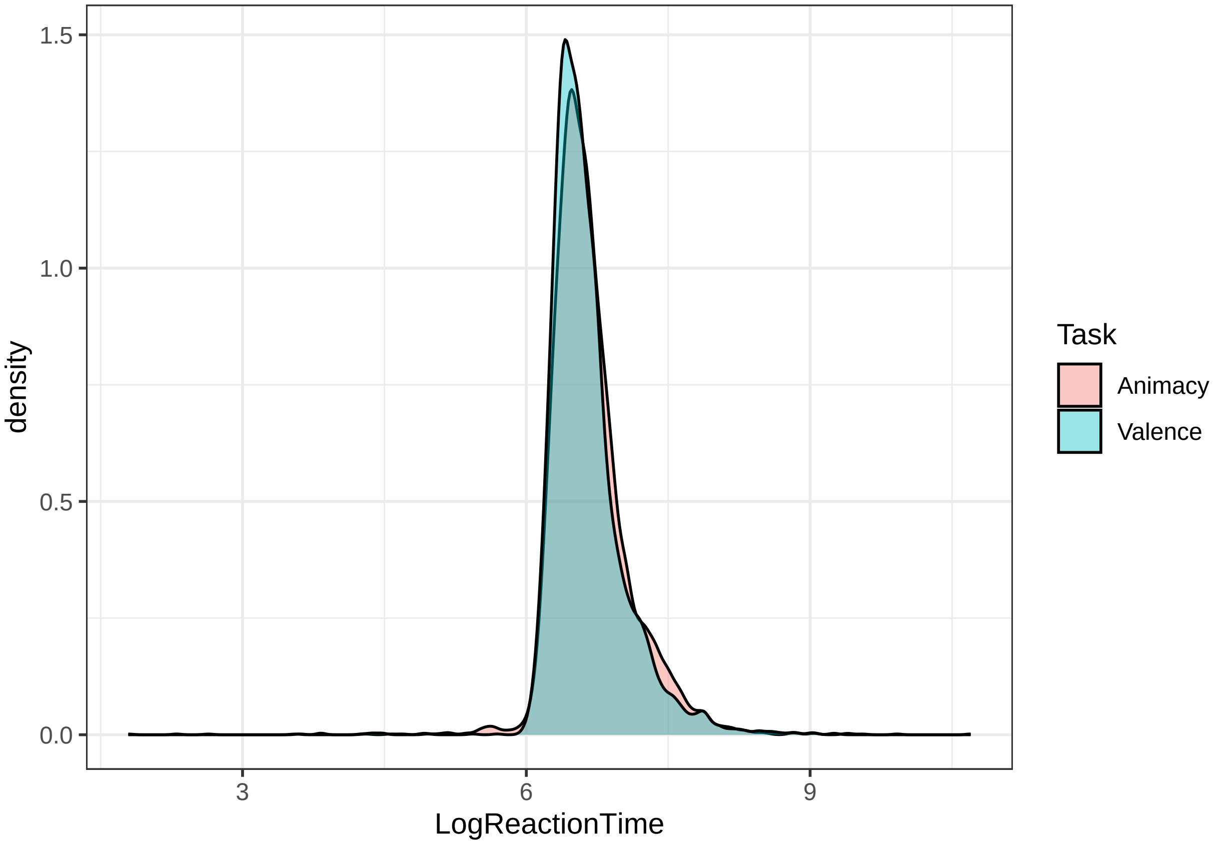
morgan moyer

2025-03-27

```
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)
```



Interaction between accuracy and task?

```
names(d)
```

```
##  [1] "X"                 "ID.true"           "Word"              "Label"
##  [5] "Animacy"           "Valence"           "Task"              "BlockOrder"
##  [9] "Group"             "Response"          "Accuracy"          "EventTime"
## [13] "Value"             "RT"                "ReactionTime"      "Key_value_F"
## [17] "Key_value_J"       "Comments"          "LogReactionTime" "LogRT"
## [21] "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
##
```

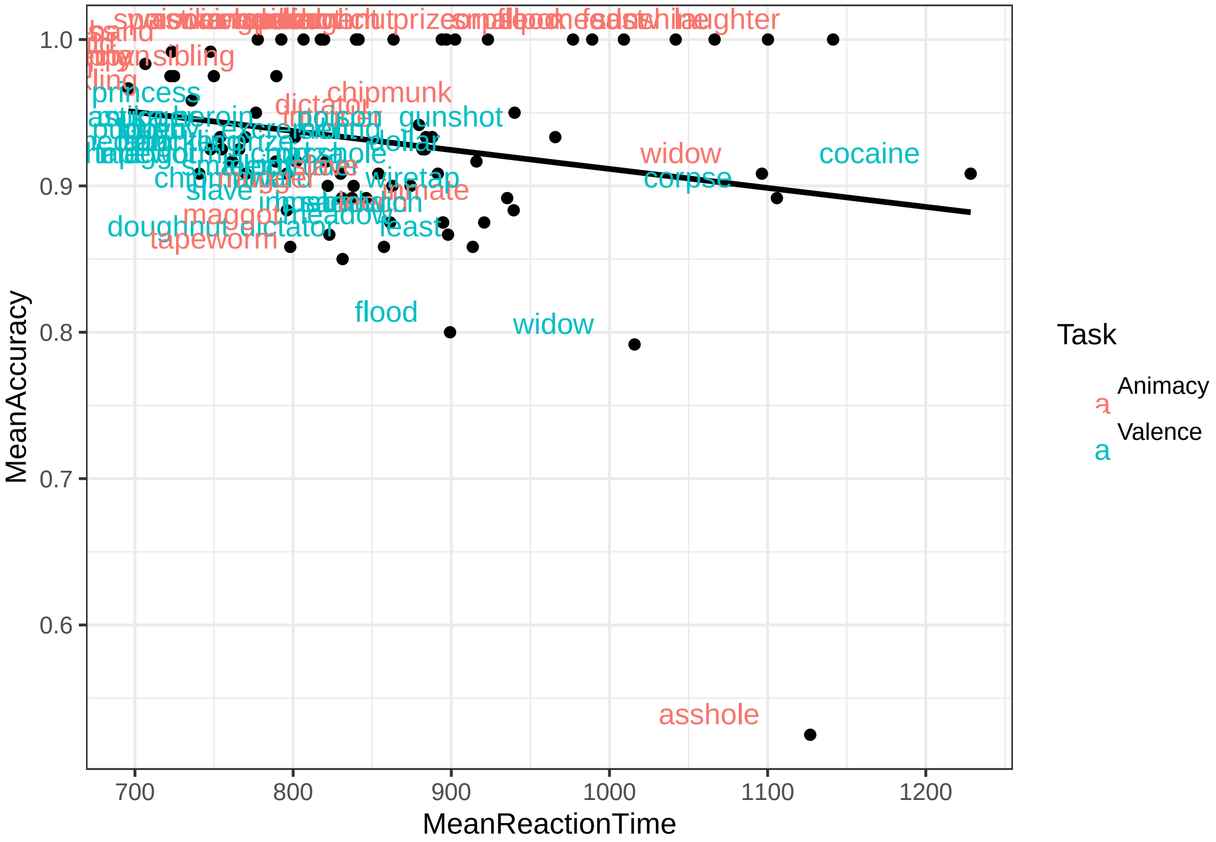
```
## REML criterion at convergence: 7023.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -12.9542  -0.5349  -0.1556   0.3430  12.4309
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.004451 0.06672
##           cTask       0.007721 0.08787  -0.60
##  ID.true  (Intercept) 0.043466 0.20849
##           cTask       0.016712 0.12928  0.29
##  Residual             0.116399 0.34117
## Number of obs: 9600, groups:  Word, 40; ID.true, 40
##
## Fixed effects:
##                   Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)        6.65713    0.03479   46.72277 191.353   <2e-16 ***
## cAccuracy          0.02135    0.01795 8711.25738   1.190    0.234
## cTask             -0.03251    0.02570   61.42544  -1.265    0.211
## cAccuracy:cTask   -0.05011    0.03580 8337.54377  -1.400    0.162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) cAccrc cTask
## cAccuracy   -0.003
## cTask        0.122  0.041
## cAccrcy:cTs  0.015 -0.169 -0.007
```

```
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/nouns2_accXrt.pdf",width = 5, height = 3)
```

Look at words most accurate for animacy

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

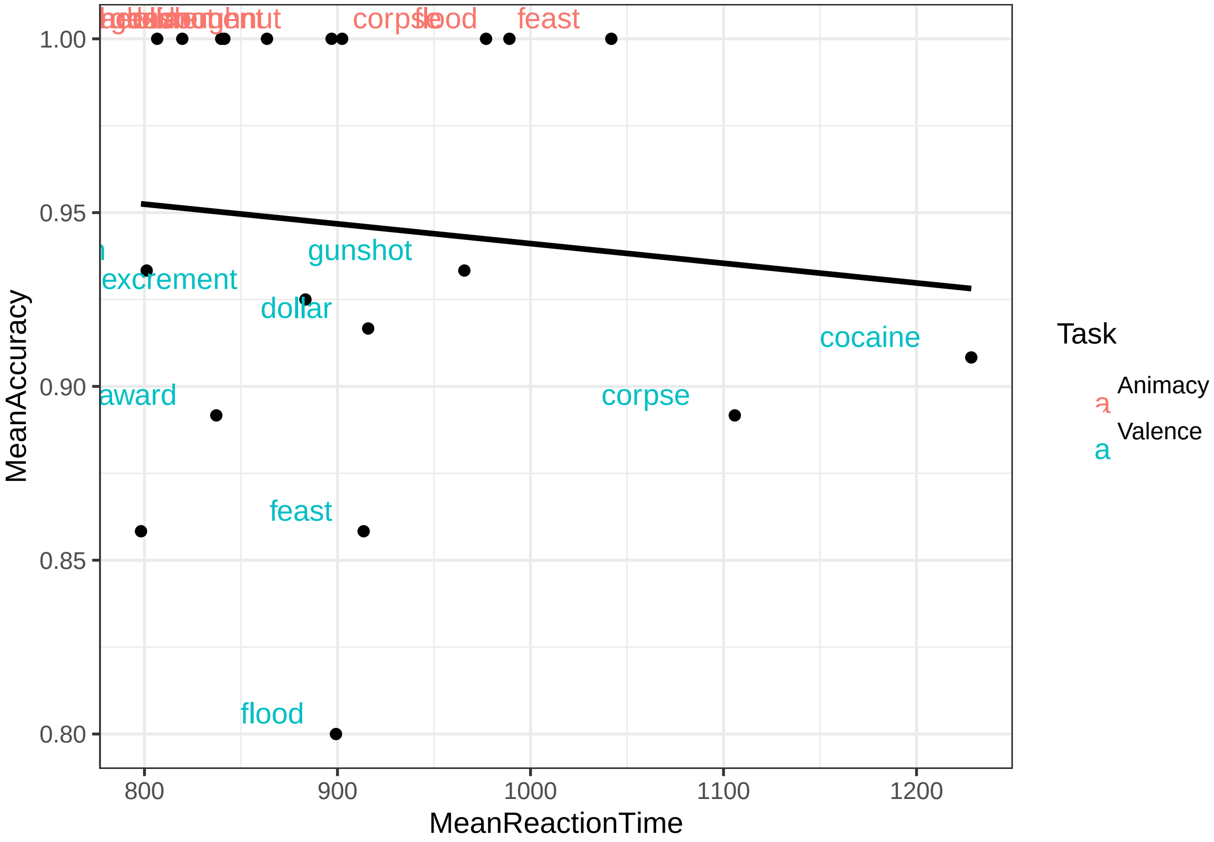
```
## `summarise()` has grouped output by 'Word'. You can override using the
## `.groups` argument.
```

```
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.
```

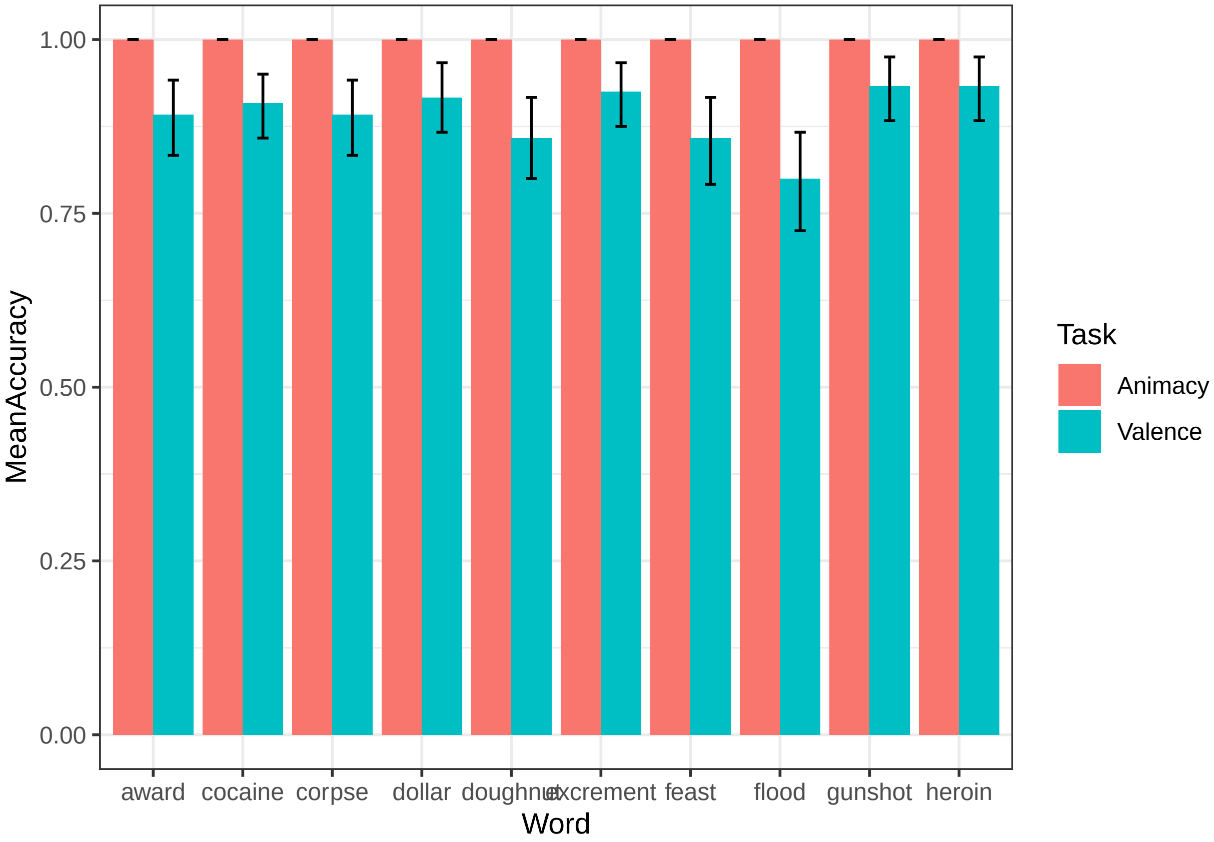
```
ggplot(agr, aes(x = MeanReactionTime, y = MeanAccuracy)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  geom_text(aes(label = Word, color = Task), vjust = -0.5, hjust = 1.5)
```

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

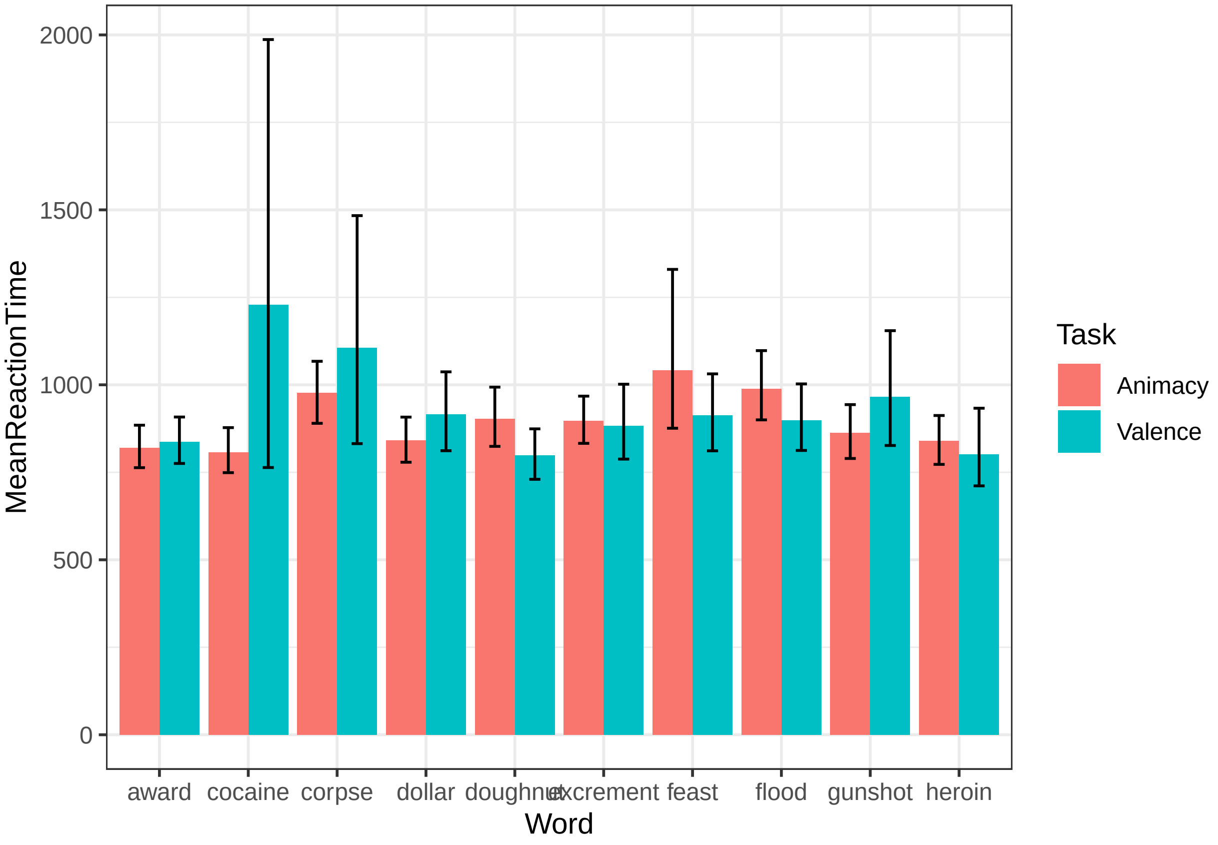


```
  # guides(legend = "none")
  # theme(legend.position = "none")  # Remove the legend
# ggsave("../graphs/exp1b_accXrt.pdf",width = 5, height = 3)
```

```
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

```
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] 4
```

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

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

```
## [1] 36
```

```
# 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] 96.41204
```

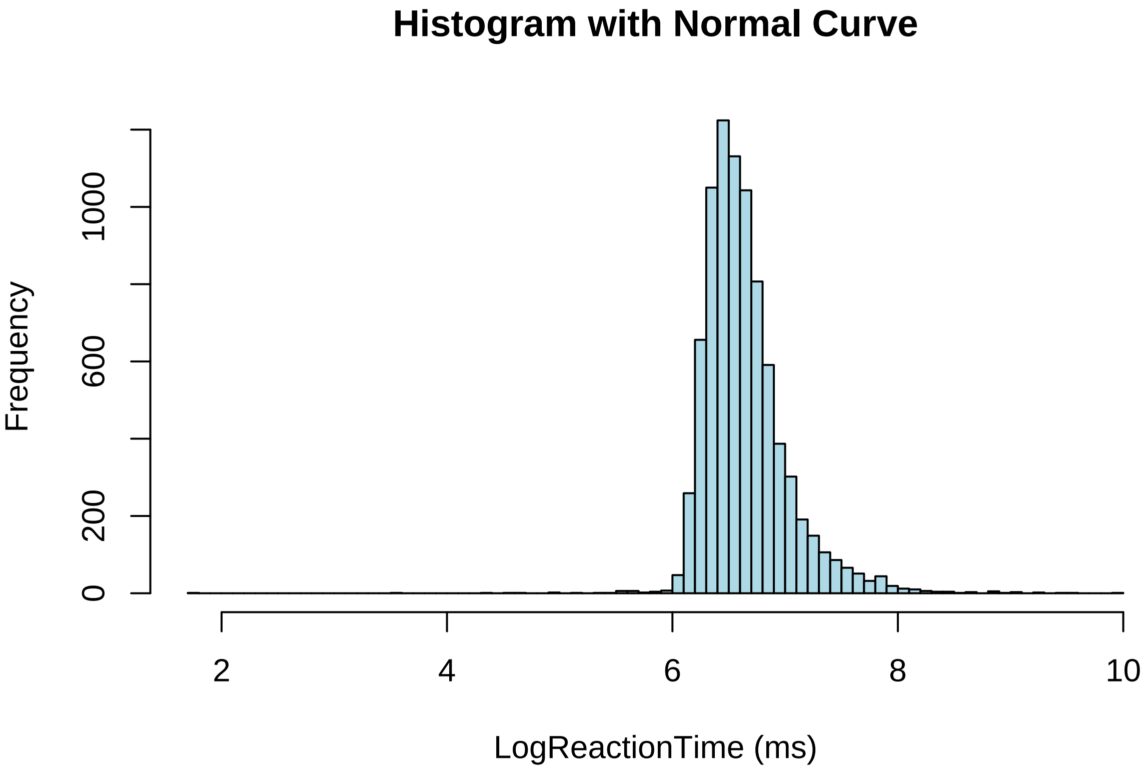
```
# Remove subjects with ReactionTime higher than 3x IQR
summary(d.inaccurate.removed$LogReactionTime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.792   6.402   6.578   6.643   6.799   9.919
```

```
  #  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] 1.791759 9.919459
```

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



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

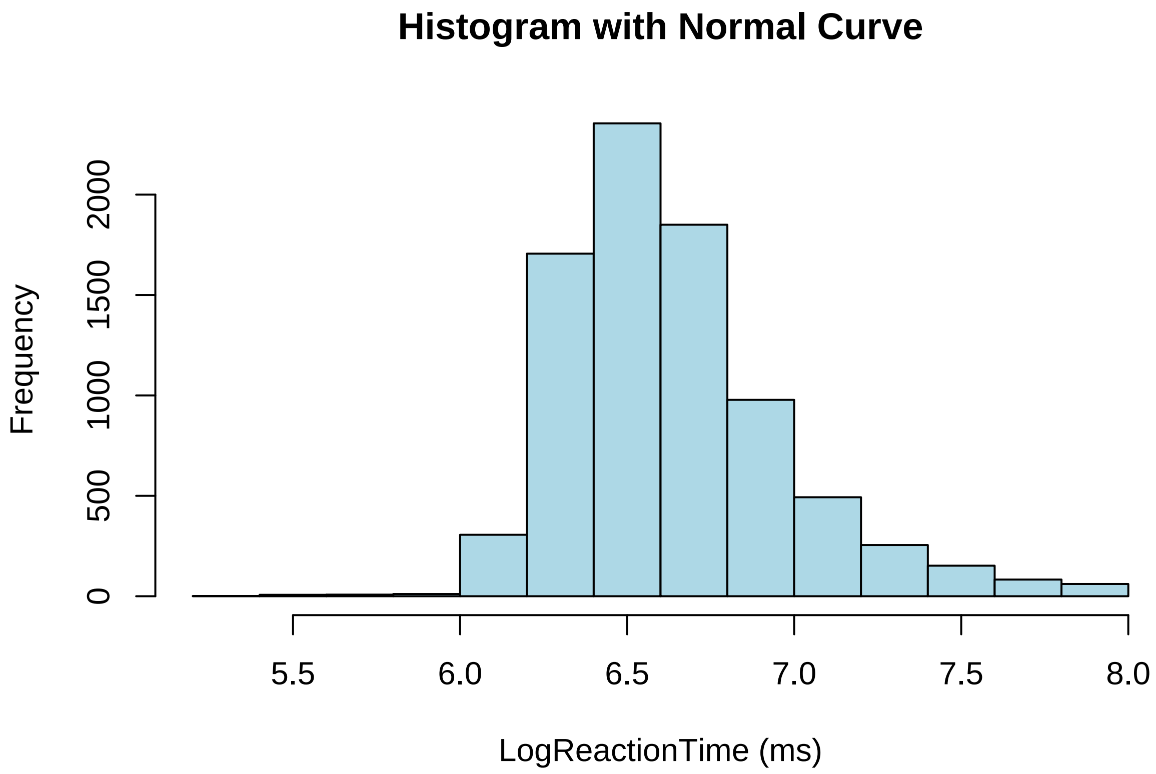
```
##       0%      25%      50%      75%     100%
## 1.791759 6.401917 6.577861 6.799056 9.919459
```

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

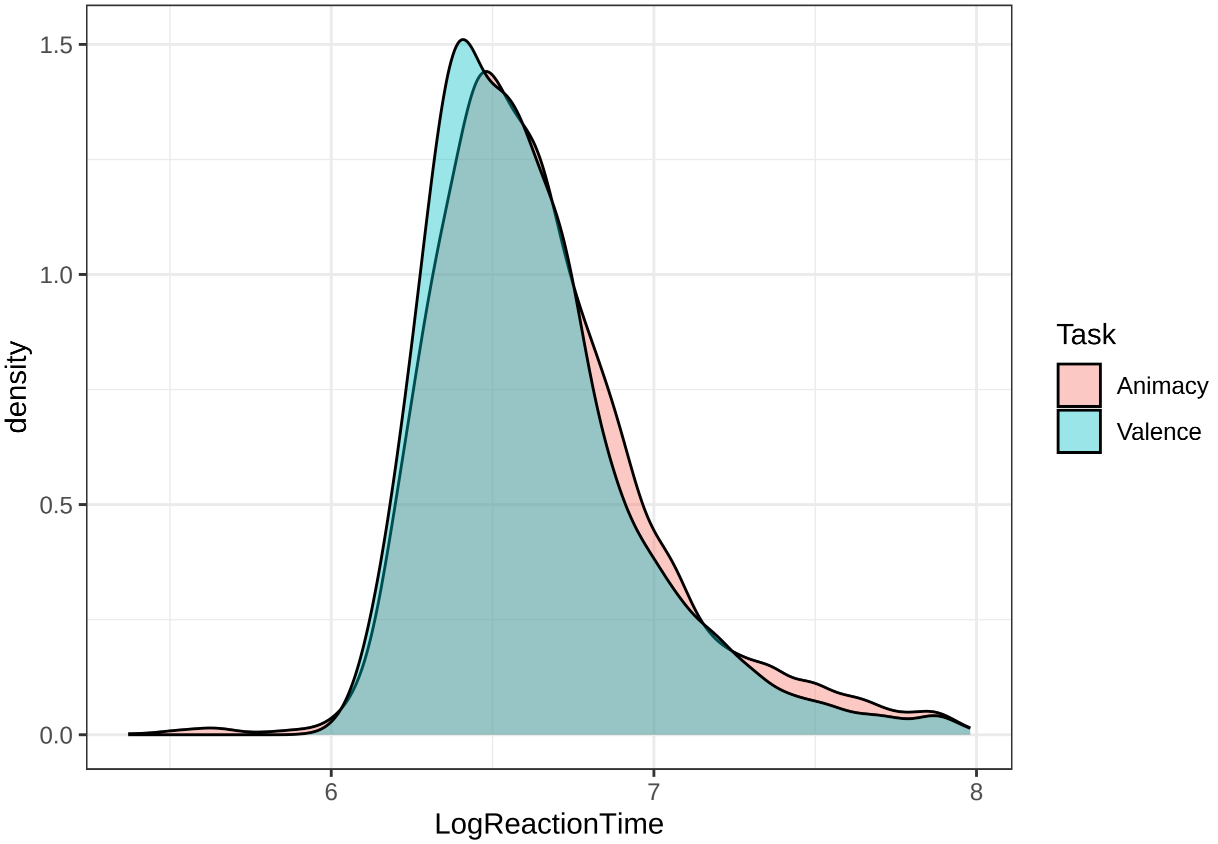
```
## [1] 1.191416
```

```
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
```

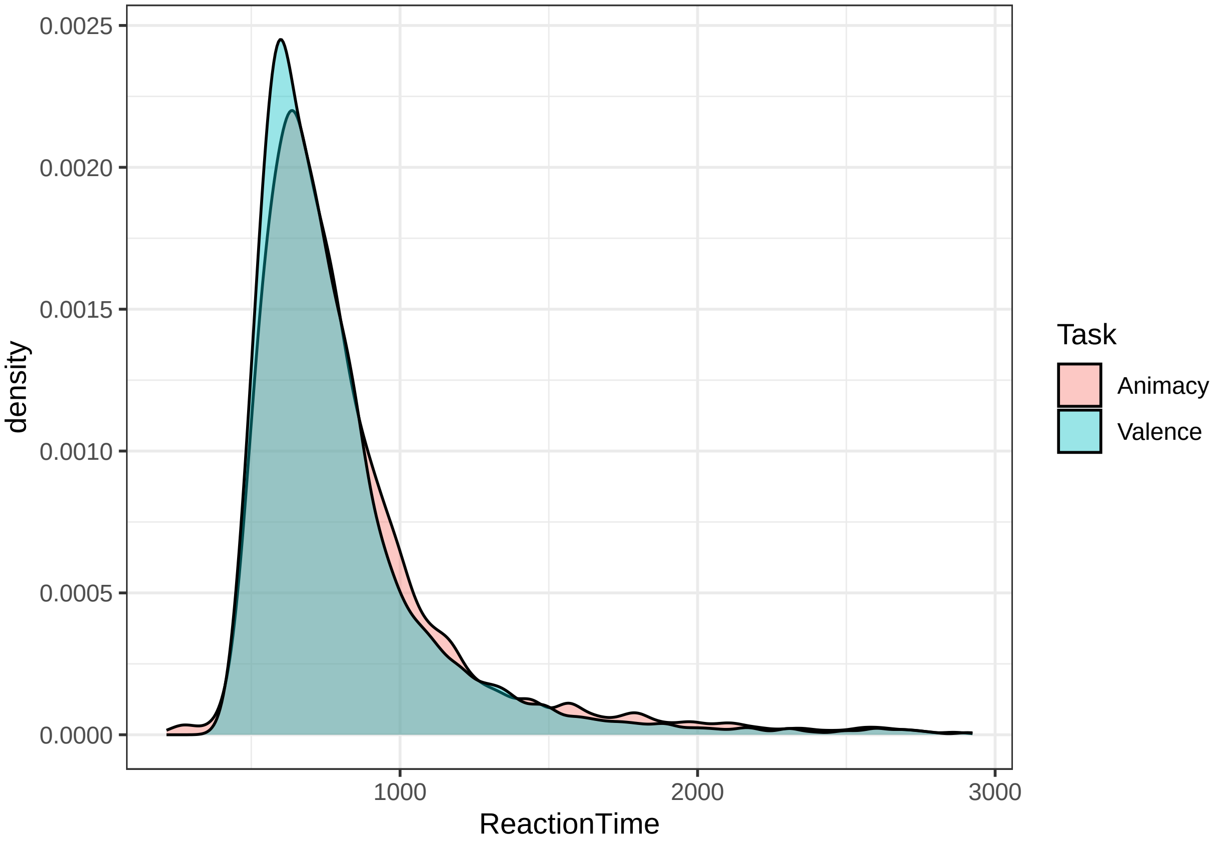
```r
# remove subjects with ReactionTime higher than 3 x IQR
df.outliers.removed <- subset(d.inaccurate.removed, (d.inaccurate.removed$LogReactionTime > cutoff.low)

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 an effect of Task?

No

```
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: 2717.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.1941 -0.6206 -0.1790  0.4012  5.2021
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.003648 0.06040
##           cTask       0.006919 0.08318  -0.50
##  ID.true  (Intercept) 0.026643 0.16323
##           cTask       0.011252 0.10608  0.22
##  Residual             0.077515 0.27842
## Number of obs: 8266, groups:  Word, 40; ID.true, 36
##
## Fixed effects:
##              Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.63441    0.02900 43.33162 228.811   <2e-16 ***
## cTask       -0.03238    0.02288 58.58037  -1.416    0.162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##       (Intr)
## cTask 0.062
```

Does Accuracy predict reaction time?

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

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

```
## fixed-effect model matrix is rank deficient so dropping 2 columns / coefficients
```

```
summary(m)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cTask * cAccuracy + (1 + cTask | ID.true) +
##     (1 + cTask | Word)
##    Data: center
##
## REML criterion at convergence: 2717.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.1941 -0.6206 -0.1790  0.4012  5.2021
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.003648 0.06040
##           cTask       0.006919 0.08318  -0.50
##  ID.true  (Intercept) 0.026643 0.16323
##           cTask       0.011252 0.10608  0.22
##  Residual             0.077515 0.27842
## Number of obs: 8266, groups:  Word, 40; ID.true, 36
##
## Fixed effects:
##              Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.63441    0.02900 43.33162 228.811   <2e-16 ***
## cTask       -0.03238    0.02288 58.58037  -1.416    0.162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##       (Intr)
## cTask 0.062
## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 2 columns / coefficients
```

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: 3049.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.0001 -0.6348 -0.2002  0.4073  5.1660
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  Word     (Intercept) 0.0035065 0.05922
##           cBlockOrder 0.0001922 0.01386  -0.31
##  ID.true  (Intercept) 0.0274048 0.16554
##  Residual             0.0820769 0.28649
## Number of obs: 8266, groups:  Word, 40; ID.true, 36
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.63285    0.02931 41.57717 226.327   <2e-16 ***
## cBlockOrder -0.02118    0.05558 34.08946  -0.381    0.706
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cBlockOrder -0.004
```

effect of word features on ReactionTime?

yes

```
m = lmer(LogReactionTime ~ cValence*cAnimacy + (1+cValence+cAnimacy|ID.true) + (1+cValence+cAnimacy|Wor
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 2 negative eigenvalues
## Warning: Model failed to converge with 2 negative eigenvalues: -2.5e-02
## -3.1e-02
```

```
summary(m)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cValence * cAnimacy + (1 + cValence + cAnimacy |
##     ID.true) + (1 + cValence + cAnimacy | Word)
##    Data: center
##
## REML criterion at convergence: 3013.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9257 -0.6379 -0.1948  0.4015  5.1702
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  Word     (Intercept) 0.0004289 0.02071
##           cValence    0.0047449 0.06888  -0.39
##           cAnimacy    0.0005715 0.02391  -0.57  0.04
##  ID.true  (Intercept) 0.0266948 0.16339
##           cValence    0.0011823 0.03438   0.09
##           cAnimacy    0.0012485 0.03533   0.60 -0.10
##  Residual             0.0815287 0.28553
## Number of obs: 8266, groups:  Word, 40; ID.true, 36
##
## Fixed effects:
##                    Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)         6.63307    0.02820 38.94821 235.253  < 2e-16 ***
## cValence           -0.03850    0.01578 39.55555  -2.441  0.01923 *
## cAnimacy            0.04199    0.01579 40.28897   2.659  0.01120 *
## cValence:cAnimacy   0.12945    0.02946 32.75582   4.395  0.00011 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cValnc cAnmcy
## cValence    -0.032
## cAnimacy     0.188 -0.004
## cVlnc:cAnmc  0.003 -0.112 -0.242
## optimizer (nloptwrap) convergence code: 0 (OK)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 2 negative eigenvalues
```

```
m = lmer(LogReactionTime ~ cTask*cValence*cAnimacy + (1+cTask+cValence+cAnimacy|ID.true) + (1+cTask+cVal
```

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

```
## Warning: Model failed to converge with 1 negative eigenvalue: -3.4e-01
```

```
summary(m)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cTask * cValence * cAnimacy + (1 + cTask +
##     cValence + cAnimacy | ID.true) + (1 + cTask + cValence +
##     cAnimacy | Word)
##    Data: center
##
## REML criterion at convergence: 2681
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.1202 -0.6165 -0.1794  0.4059  5.2337
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  Word     (Intercept) 0.0003645 0.01909
##           cTask       0.0047479 0.06890   0.17
##           cValence    0.0050310 0.07093  -0.59  0.13
##           cAnimacy    0.0015907 0.03988  -0.52 -0.92  0.00
##  ID.true  (Intercept) 0.0266273 0.16318
##           cTask       0.0112412 0.10602   0.22
##           cValence    0.0012370 0.03517   0.12  0.27
##           cAnimacy    0.0014687 0.03832   0.59 -0.19 -0.05
##  Residual             0.0768498 0.27722
## Number of obs: 8266, groups:  Word, 40; ID.true, 36
##
## Fixed effects:
##                         Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)              6.63456    0.02827 39.43220 234.721  < 2e-16 ***
## cTask                   -0.03229    0.02164 52.28180  -1.492 0.141706
## cValence                -0.04032    0.01657 28.12994  -2.433 0.021606 *
## cAnimacy                 0.03982    0.01671 29.21295   2.383 0.023897 *
## cTask:cValence           0.05249    0.02499 35.02060   2.100 0.042986 *
## cTask:cAnimacy          -0.04746    0.02499 35.02451  -1.899 0.065875 .
## cValence:cAnimacy        0.13369    0.03109 22.54020   4.300 0.000277 ***
## cTask:cValence:cAnimacy -0.14462    0.04998 35.03489  -2.893 0.006517 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cTask  cValnc cAnmcy cTsk:V cTsk:A cVln:A
## cTask        0.182
## cValence    -0.047  0.122
## cAnimacy     0.178 -0.231 -0.011
## cTask:cVlnc  0.023  0.008  0.046 -0.008
## cTsk:cAnmcy -0.088  0.007 -0.008  0.058  0.005
## cVlnc:cAnmc -0.001 -0.005 -0.146 -0.314 -0.319  0.083
## cTsk:cVln:A -0.002  0.003 -0.299  0.077  0.011  0.013  0.059
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```