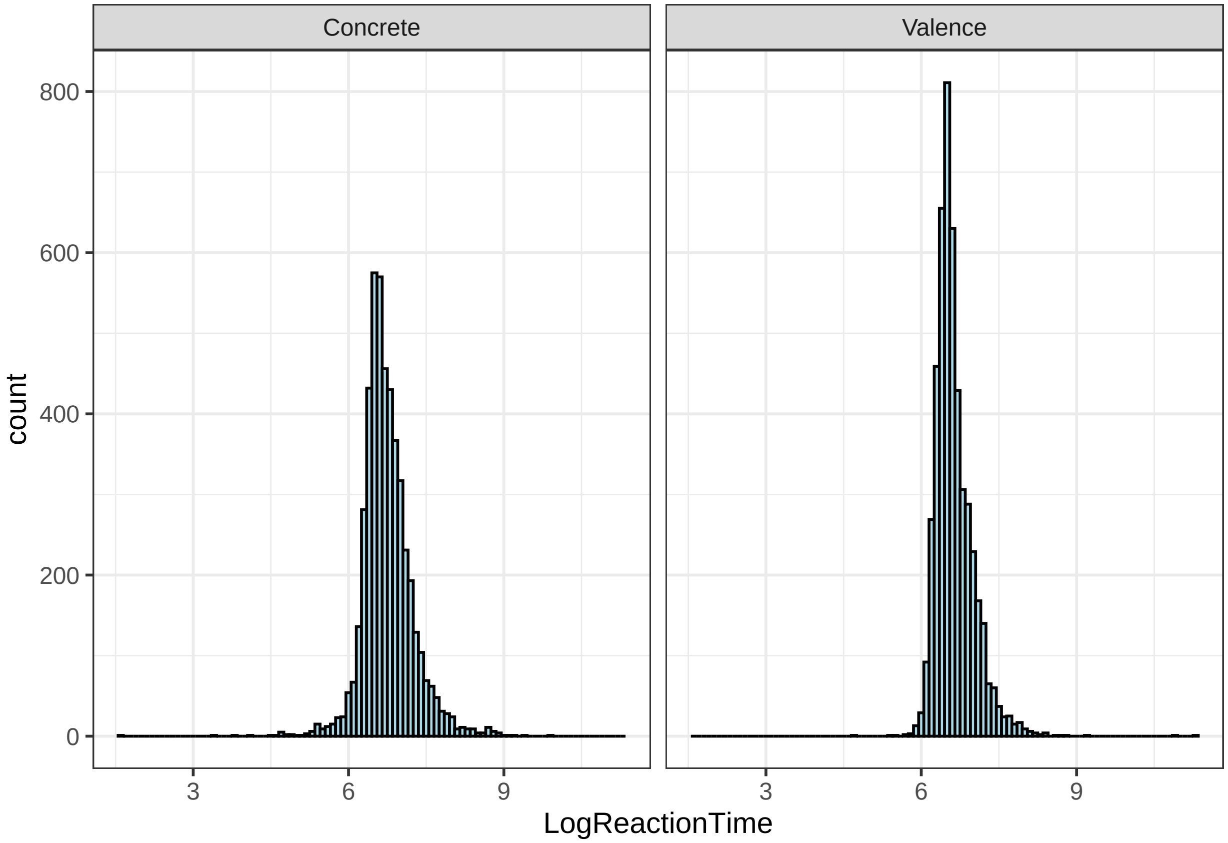
Adjs Conc-Abs: Analysis

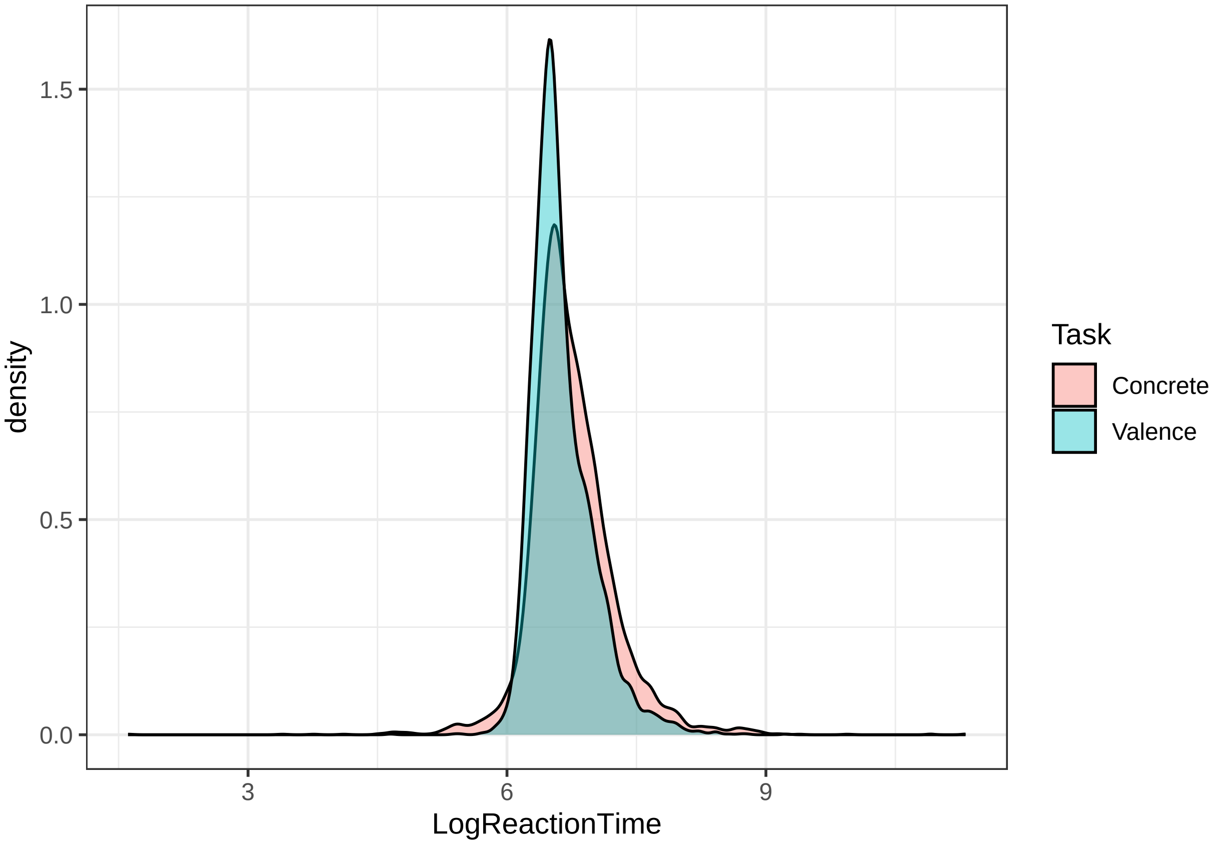
morgan moyer

2025-03-26

```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)
```

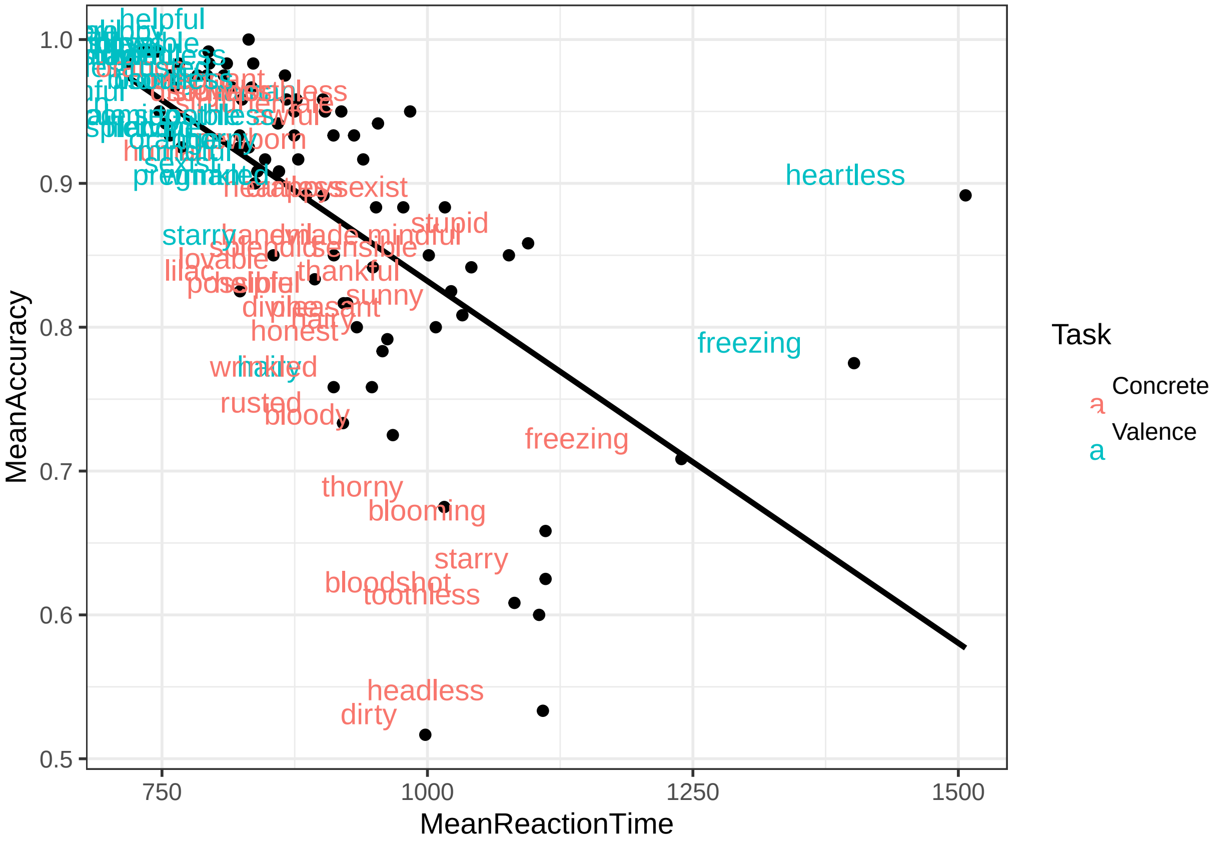
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cAccuracy * cTask + (1 + cTask | Word) + (1 +
##     cTask | ID.true)
##    Data: dcen
##
## REML criterion at convergence: 6365
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -12.3572  -0.5780  -0.1481   0.4092  13.9472
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.002443 0.04942
##           cTask       0.003430 0.05857  -0.12
##  ID.true  (Intercept) 0.066652 0.25817
##           cTask       0.042951 0.20725  -0.44
##  Residual             0.108397 0.32924
## Number of obs: 9600, groups:  Word, 40; ID.true, 40
##
## Fixed effects:
##                   Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)        6.69182    0.04171   41.84694 160.455  < 2e-16 ***
## cAccuracy         -0.01096    0.01261 9401.15604  -0.869  0.38487
## cTask             -0.10819    0.03475   44.70672  -3.114  0.00322 **
## cAccuracy:cTask    0.02952    0.02518 9314.15001   1.173  0.24101
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cAccrc cTask
## cAccuracy   -0.008
## cTask       -0.412 -0.047
## cAccrcy:cTs -0.020  0.408 -0.019
```

```
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 == "Concrete") %>%
  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))
```

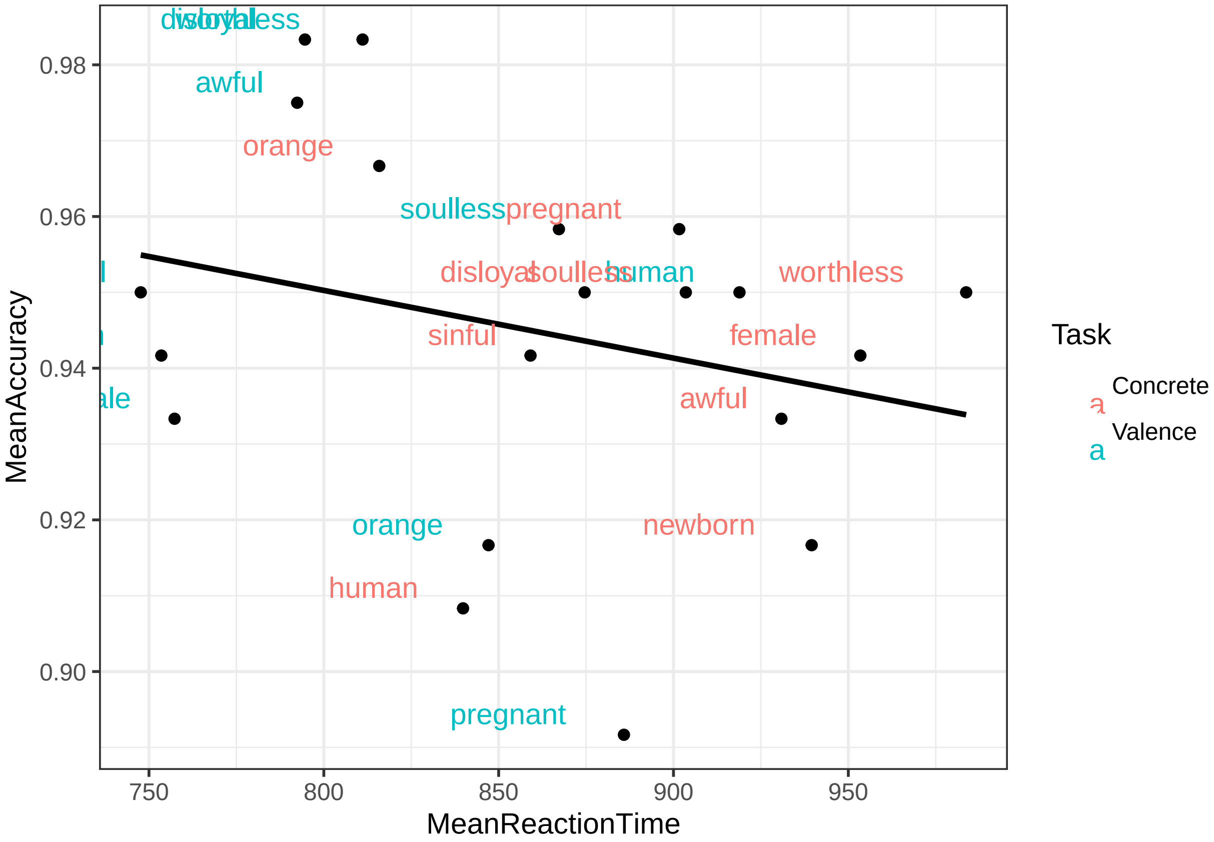
```
## `summarise()` has grouped output by 'Word'. You can override using the
## `.groups` argument.
```

```
print(agr)
```

```
## # A tibble: 20 x 4
## # Groups:   Word [10]
##    Word      Task     MeanAccuracy MeanReactionTime
##    <chr>     <chr>           <dbl>            <dbl>
##  1 awful     Concrete        0.933             931.
##  2 awful     Valence         0.975             792.
##  3 disloyal  Concrete        0.95              875.
##  4 disloyal  Valence         0.983             795.
##  5 female    Concrete        0.942             953.
##  6 female    Valence         0.933             757.
##  7 human     Concrete        0.908             840.
##  8 human     Valence         0.95              919.
##  9 newborn   Concrete        0.917             940.
## 10 newborn   Valence         0.942             754.
## 11 orange    Concrete        0.967             816.
## 12 orange    Valence         0.917             847.
## 13 pregnant  Concrete        0.958             902.
## 14 pregnant  Valence         0.892             886.
## 15 sinful    Concrete        0.942             859.
## 16 sinful    Valence         0.95              748.
## 17 soulless  Concrete        0.95              904.
## 18 soulless  Valence         0.958             867.
## 19 worthless Concrete        0.95              984.
## 20 worthless Valence         0.983             811.
```

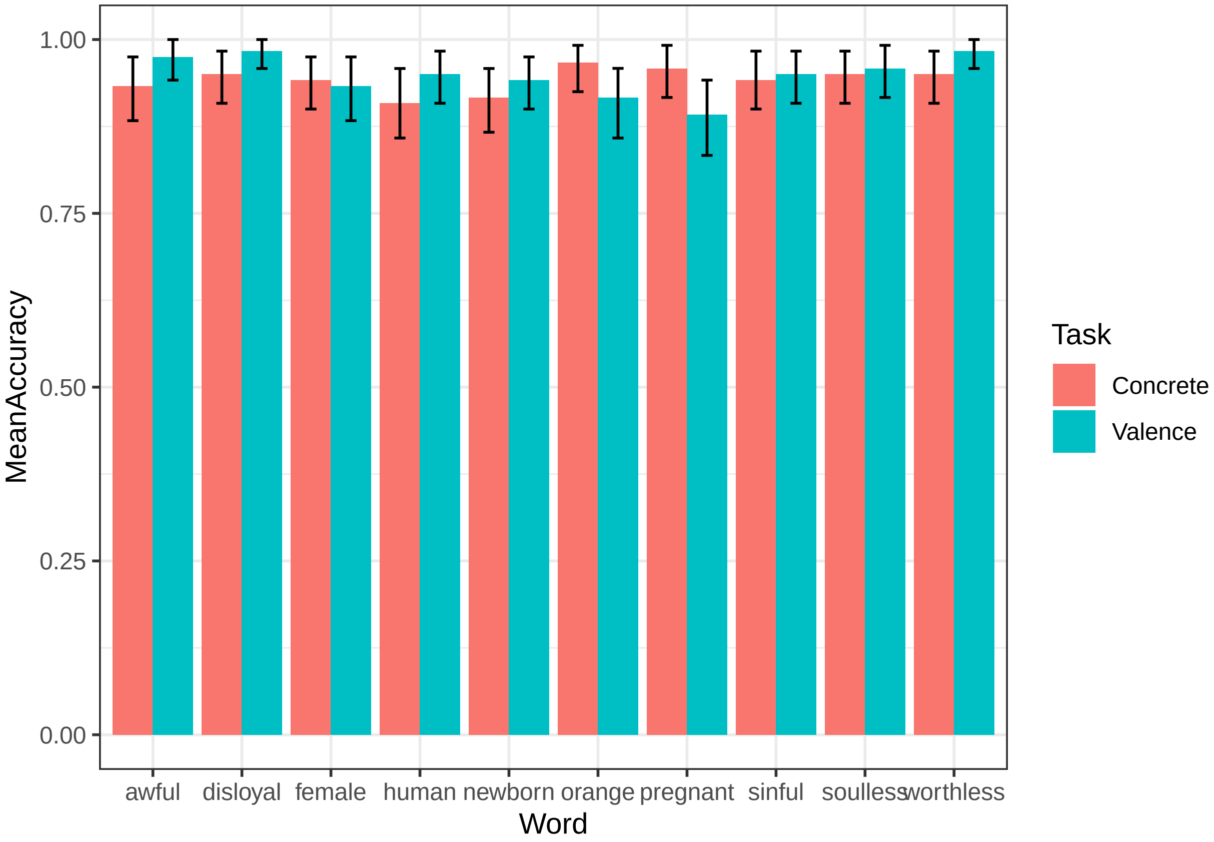
```
ggplot(agr, aes(x = MeanReactionTime, y = MeanAccuracy)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  geom_text(aes(label = Word, color = Task), vjust = -0.5, hjust = 1.5)
```

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

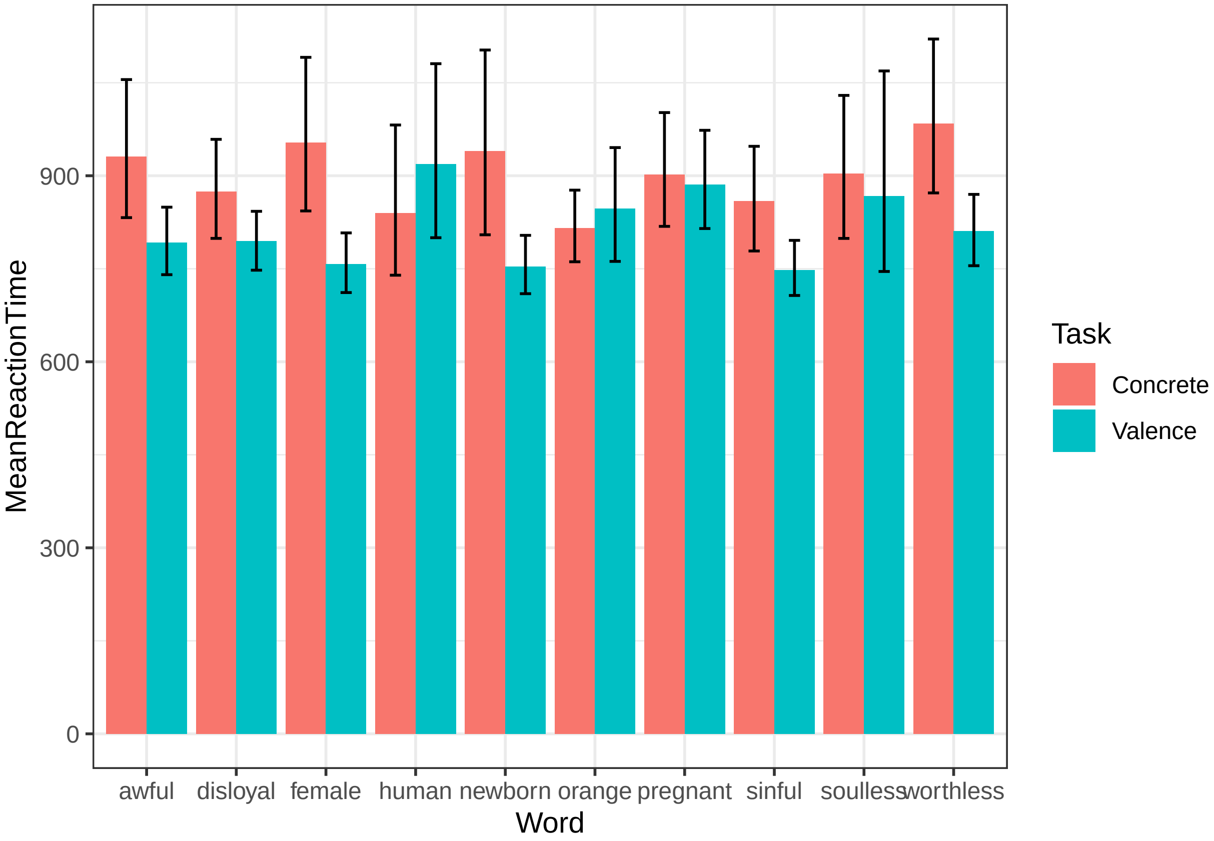


```
  # guides(legend = "none")
  # theme(legend.position = "none")  # Remove the legend
# ggsave("../graphs/exp1b_accXrt.pdf",width = 5, height = 3)
```

```
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] 11
```

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

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

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

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] 92.74425
```

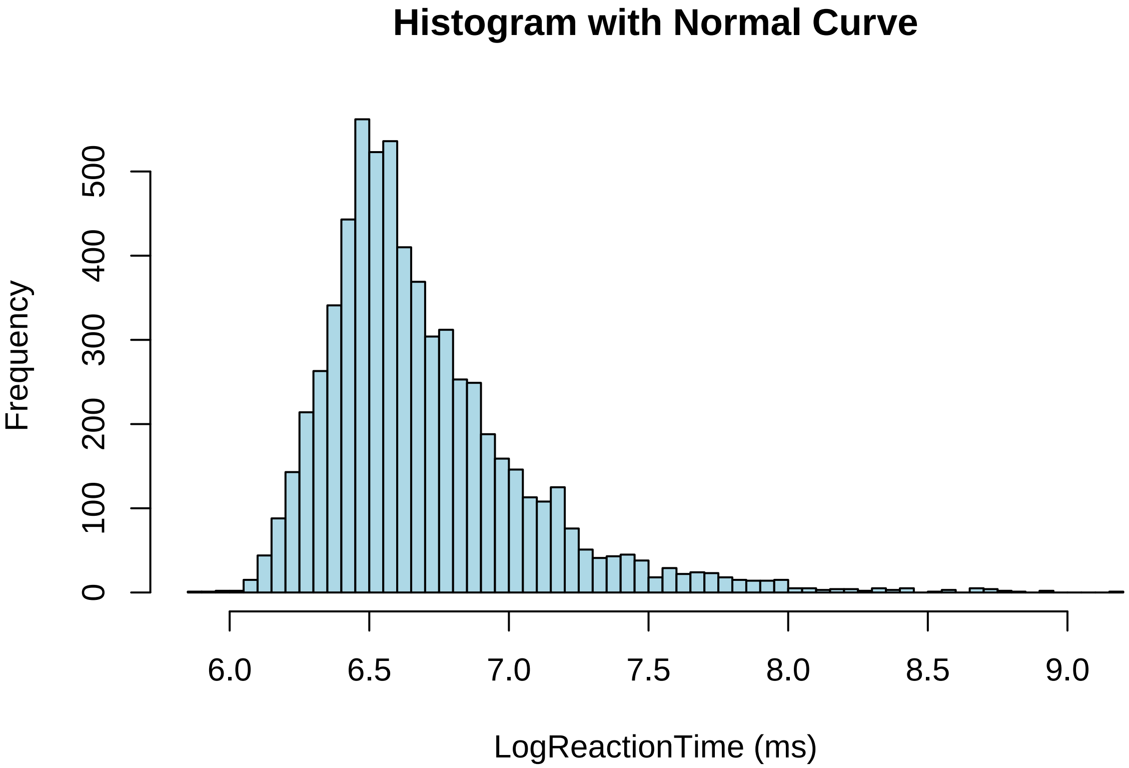
```
# Remove subjects with ReactionTime higher than 3x IQR
summary(d.inaccurate.removed$LogReactionTime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.869   6.455   6.605   6.694   6.853   9.177
```

```
  #  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] 5.869297 9.176887
```

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



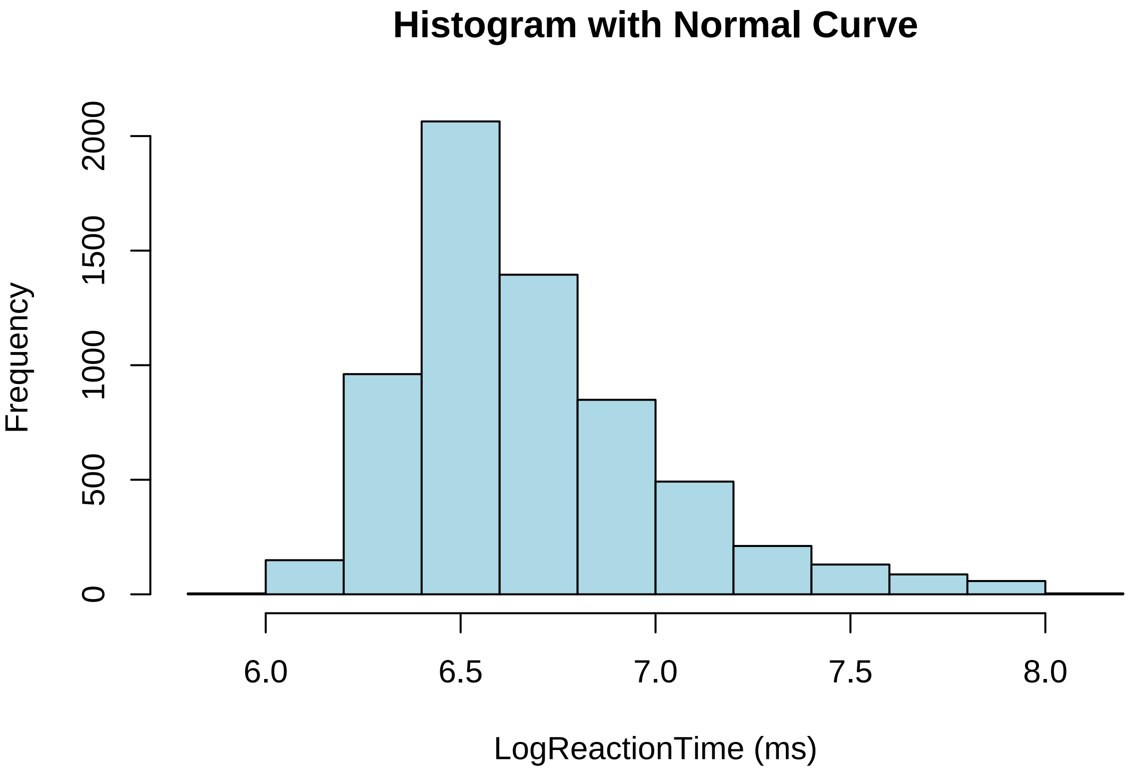
```
quantile(d.inaccurate.removed$LogReactionTime)
```

```
##       0%      25%      50%      75%     100%
## 5.869297 6.455199 6.605298 6.852771 9.176887
```

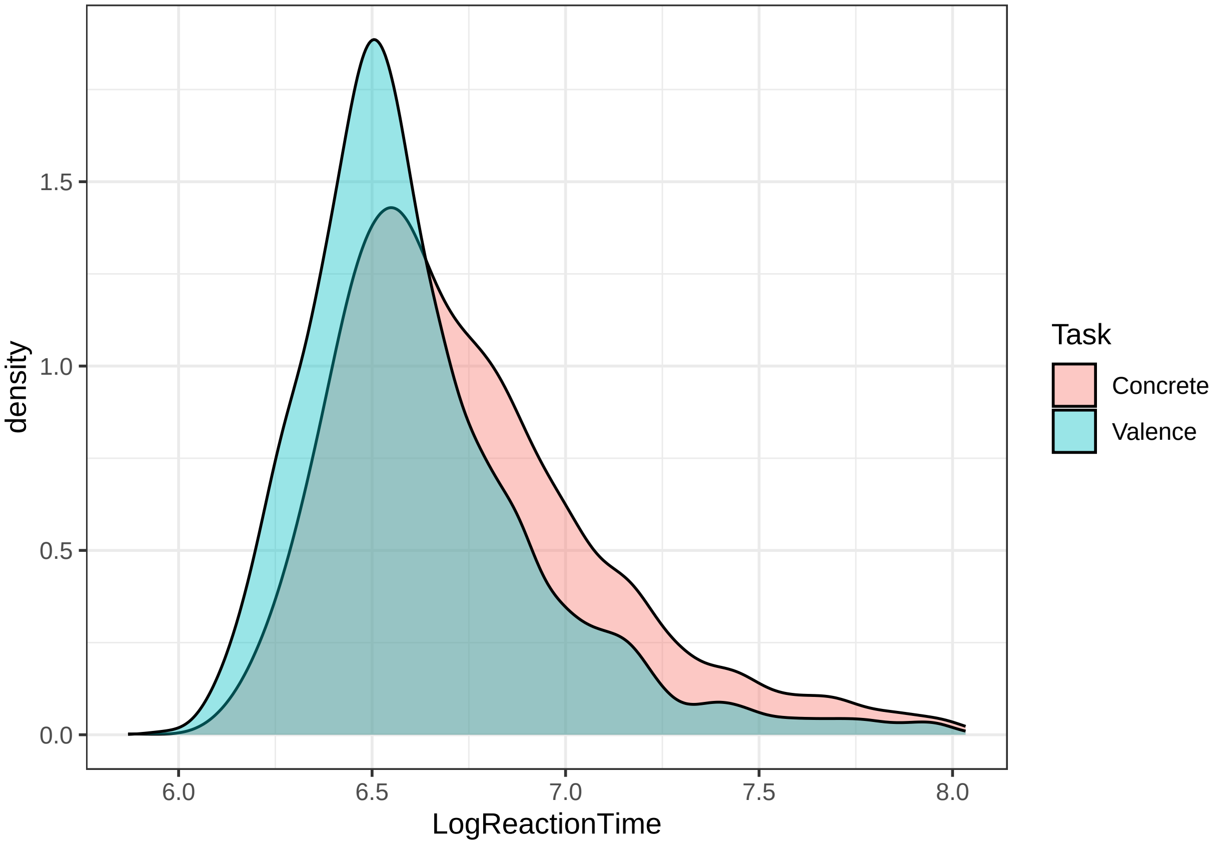
```
IQR(d.inaccurate.removed$LogReactionTime)*3 # 0.7526289
```

```
## [1] 1.192717
```

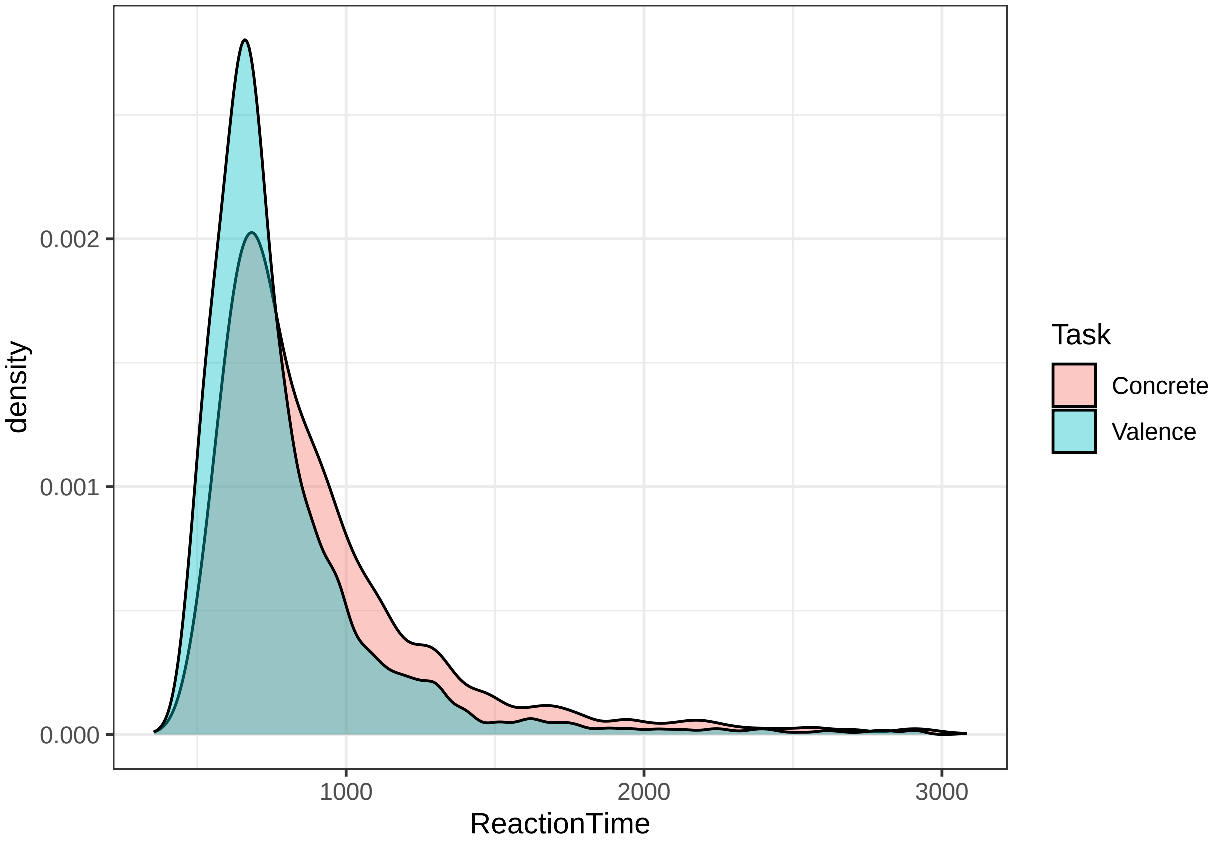
```
cutoff.high <- quantile(d.inaccurate.removed$LogReactionTime)[4] + IQR(d.inaccurate.removed$LogReaction
cutoff.low <- quantile(d.inaccurate.removed$LogReactionTime)[2] - IQR(d.inaccurate.removed$LogReactionTi


# 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")
```



```
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: 1896.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1413 -0.6566 -0.1782  0.4713  4.6790
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.003428 0.05855
##           cTask       0.005003 0.07073  -0.19
##  ID.true  (Intercept) 0.027829 0.16682
##           cTask       0.016165 0.12714  -0.04
##  Residual             0.074623 0.27317
## Number of obs: 6404, groups:  Word, 40; ID.true, 29
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.69052    0.03251 32.89865 205.777  < 2e-16 ***
## cTask       -0.14254    0.02702 38.28428  -5.276 5.48e-06 ***
## ---
## 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.

```
m = lmer(LogReactionTime ~ cTask*ConcValCombo + (1+ConcValCombo+cTask|ID.true) + (1+cTask|Word), data=ce
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: 1833.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1978 -0.6492 -0.1761  0.4667  4.6498
##
## Random effects:
##  Groups   Name                         Variance Std.Dev. Corr
##  Word     (Intercept)                  0.002035 0.04511
##           cTask                        0.005241 0.07240  -0.20
##  ID.true  (Intercept)                  0.027851 0.16689
##           ConcValComboabstract-positive 0.003182 0.05641  -0.01
##           ConcValComboconcrete-negative 0.005079 0.07126   0.18  0.25
##           ConcValComboconcrete-positive 0.003057 0.05529  -0.36  0.47 -0.12
##           cTask                        0.016228 0.12739  -0.11  0.22  0.30
##  Residual                              0.072861 0.26993
##
##
##
##
##
##
##
```

```
##   0.24
##
## Number of obs: 6404, groups:  Word, 40; ID.true, 29
##
## Fixed effects:
##                                     Estimate Std. Error       df t value
## (Intercept)                          6.65895    0.03476 38.51214 191.599
## cTask                               -0.11114    0.03549 57.89120  -3.131
## ConcValComboabstract-positive        0.02120    0.02459 43.00371   0.862
## ConcValComboconcrete-negative        0.09996    0.02601 47.07693   3.842
## ConcValComboconcrete-positive        0.00881    0.02456 43.33443   0.359
## cTask:ConcValComboabstract-positive -0.06389    0.03741 34.45734  -1.708
## cTask:ConcValComboconcrete-negative -0.02482    0.03780 35.89873  -0.657
## cTask:ConcValComboconcrete-positive -0.03563    0.03755 34.97238  -0.949
##                                     Pr(>|t|)
## (Intercept)                          < 2e-16 ***
## cTask                               0.002727 **
## ConcValComboabstract-positive       0.393327
## ConcValComboconcrete-negative       0.000364 ***
## ConcValComboconcrete-positive       0.721500
## cTask:ConcValComboabstract-positive 0.096640 .
## cTask:ConcValComboconcrete-negative 0.515513
## cTask:ConcValComboconcrete-positive 0.349183
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                      (Intr) cTask  CncVlCmbb- CncVlCmbcncrt-n CncVlCmbcncrt-p
## cTask                -0.116
## CncVlCmbbs-          -0.292  0.135
## CncVlCmbcncrt-n      -0.191  0.171  0.440
## CncVlCmbcncrt-p      -0.425  0.140  0.493      0.361
## cTsk:CncVlCmbb-       0.049 -0.527 -0.139     -0.066          -0.070
## cTsk:CncVlCmbcncrt-n  0.049 -0.522 -0.069     -0.139          -0.069
## cTsk:CncVlCmbcncrt-p  0.049 -0.525 -0.069     -0.066          -0.142
##                      cTsk:CncVlCmbb- cTsk:CncVlCmbcncrt-n
## cTask
## CncVlCmbbs-
## CncVlCmbcncrt-n
## CncVlCmbcncrt-p
## cTsk:CncVlCmbb-
## cTsk:CncVlCmbcncrt-n  0.495
## cTsk:CncVlCmbcncrt-p  0.498           0.493
```

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: 2557.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6981 -0.6703 -0.2061  0.4839  4.5247
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Word     (Intercept) 0.003007 0.05484
##  ID.true  (Intercept) 0.027274 0.16515
##  Residual             0.084573 0.29081
## Number of obs: 6404, groups:  Word, 40; ID.true, 29
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.68750    0.03208 32.35924   208.5   <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: 2559.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6930 -0.6679 -0.2051  0.4814  4.5237
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  Word     (Intercept) 3.002e-03 0.054792
##           cBlockOrder 6.539e-05 0.008086 -0.61
##  ID.true  (Intercept) 2.698e-02 0.164250
##  Residual             8.456e-02 0.290787
## Number of obs: 6404, groups:  Word, 40; ID.true, 29
```

```
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.68763    0.03192 31.23747 209.536   <2e-16 ***
## cBlockOrder -0.07054    0.06148 26.98268  -1.147    0.261
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cBlockOrder -0.008
```

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: 2500.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6424 -0.6610 -0.2044  0.4768  4.5352
##
## Random effects:
##  Groups   Name                         Variance Std.Dev. Corr
##  Word     (Intercept)                  0.001769 0.04206
##  ID.true  (Intercept)                  0.026985 0.16427
##           ConcValComboabstract-positive 0.002740 0.05234   0.01
##           ConcValComboconcrete-negative 0.004441 0.06664   0.25  0.17
##           ConcValComboconcrete-positive 0.002526 0.05026  -0.36  0.49 -0.30
##  Residual                              0.082997 0.28809
## Number of obs: 6404, groups:  Word, 40; ID.true, 29
##
## Fixed effects:
##                                Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)                    6.658015   0.034021 37.304596 195.705  < 2e-16
## ConcValComboabstract-positive  0.021754   0.023414 40.903475   0.929 0.358285
## ConcValComboconcrete-negative  0.093421   0.024771 44.530908   3.771 0.000476
## ConcValComboconcrete-positive  0.005274   0.023316 41.106436   0.226 0.822159
##
## (Intercept)                   ***
## ConcValComboabstract-positive
## ConcValComboconcrete-negative ***
## ConcValComboconcrete-positive
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                  (Intr) CncVlCmbb- CncVlCmbcncrt-n
## CncVlCmbbs-     -0.280
## CncVlCmbcncrt-n -0.158  0.427
## CncVlCmbcncrt-p -0.416  0.498      0.333
```

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

- Nope

```
str(df_factors)
```

```
## 'data.frame':    6404 obs. of  20 variables:
##  $ X              : Factor w/ 6404 levels "481","482","483",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ ID.true        : Factor w/ 29 levels "5588b04ffdf99b7a91e75ddb",..: 29 29 29 29 29 29 29 29 29 29
##  $ Word           : Factor w/ 40 levels "awful","bloodshot",..: 17 22 6 3 37 29 23 10 4 2 ...
##  $ Label          : Factor w/ 2 levels "test_conc","test_val": 2 2 2 2 2 2 2 2 2 2 ...
##  $ ConcValCombo   : Factor w/ 4 levels "abstract-negative",..: 2 4 3 3 3 1 4 4 4 3 ...
##  $ Task           : Factor w/ 2 levels "Concrete","Valence": 2 2 2 2 2 2 2 2 2 2 ...
##  $ BlockOrder     : Factor w/ 2 levels "CV","VC": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Group          : Factor w/ 4 levels "abstract;concrete",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Response       : Factor w/ 4 levels "abstract","concrete",..: 4 4 3 3 3 3 4 4 4 3 ...
##  $ Accuracy       : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ EventTime      : Factor w/ 6404 levels "1739194390053",..: 4590 4591 4592 4593 4594 4595 4596 459
##  $ Value          : Factor w/ 4 levels "abstract","concrete",..: 4 4 3 3 3 3 4 4 4 3 ...
##  $ RT             : Factor w/ 1017 levels "1268","1283",..: 210 372 550 534 463 196 260 886 371 543
##  $ ReactionTime   : int  575 1288 638 690 594 993 521 499 781 678 ...
##  $ Key_value_F    : Factor w/ 4 levels "abstract","concrete",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ 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  6.35 7.16 6.46 6.54 6.39 ...
##  $ LogRT          : Factor w/ 1017 levels "7.14519613499717",..: 210 372 550 534 463 196 260 886 371
##  $ TrialNumber    : Factor w/ 240 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
```

```
sem <- df_factors %>%
  filter(Task == "Concrete") %>%
  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)),
        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: 1356.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0744 -0.6783 -0.1973  0.4866  4.3082
##
## Random effects:
##  Groups   Name          Variance Std.Dev. Corr
##  Word     (Intercept)   0.002235 0.04728
##           cConcValCombo 0.002383 0.04882  1.00
##  ID.true  (Intercept)   0.033109 0.18196
##           cConcValCombo 0.001091 0.03304  -0.50
##  Residual               0.085451 0.29232
## Number of obs: 3077, groups:  Word, 40; ID.true, 29
##
## Fixed effects:
##                Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)    6.77670    0.03556 32.30054 190.569   <2e-16 ***
## cConcValCombo  0.03407    0.01206 41.59887   2.826   0.0072 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cConcValCmb -0.066
## 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: 1356.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0932 -0.6791 -0.1892  0.4897  4.3796
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.005227 0.07230
##  ID.true  (Intercept) 0.032873 0.18131
##           cSemantic   0.008981 0.09477  -0.39
##  Residual             0.084752 0.29112
## Number of obs: 3077, groups:  Word, 40; ID.true, 29
```

```
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.76467    0.03596 34.12698 188.121   <2e-16 ***
## cSemantic    0.04073    0.03076 50.91763   1.324    0.191
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## cSemantic -0.223
```

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
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00336745 (tol = 0.002, component 1)
```

```
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: 453.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7858 -0.6471 -0.1693  0.4490  5.0441
##
## Random effects:
##  Groups   Name          Variance  Std.Dev. Corr
##  Word     (Intercept)   3.996e-03 0.06322
##           cConcValCombo 4.584e-05 0.00677  1.00
##  ID.true  (Intercept)   3.091e-02 0.17582
##           cConcValCombo 3.148e-04 0.01774  0.37
##  Residual               6.295e-02 0.25090
## Number of obs: 3327, groups:  Word, 40; ID.true, 29
##
## Fixed effects:
##                 Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)     6.622428   0.034439 33.089674 192.297   <2e-16 ***
## cConcValCombo   0.009882   0.010276 42.662418   0.962    0.342
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cConcValCmb 0.169
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00336745 (tol = 0.002, component 1)
```

```
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: 415.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7697 -0.6416 -0.1698  0.4544  4.9913
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.003507 0.05922
##  ID.true  (Intercept) 0.031003 0.17608
##           cValence    0.005801 0.07616  -0.23
##  Residual             0.061905 0.24881
## Number of obs: 3327, groups:  Word, 40; ID.true, 29
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.62268    0.03428 32.41017 193.166   <2e-16 ***
## cValence    -0.05270    0.02501 51.69915  -2.107     0.04 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## cValence -0.123
```