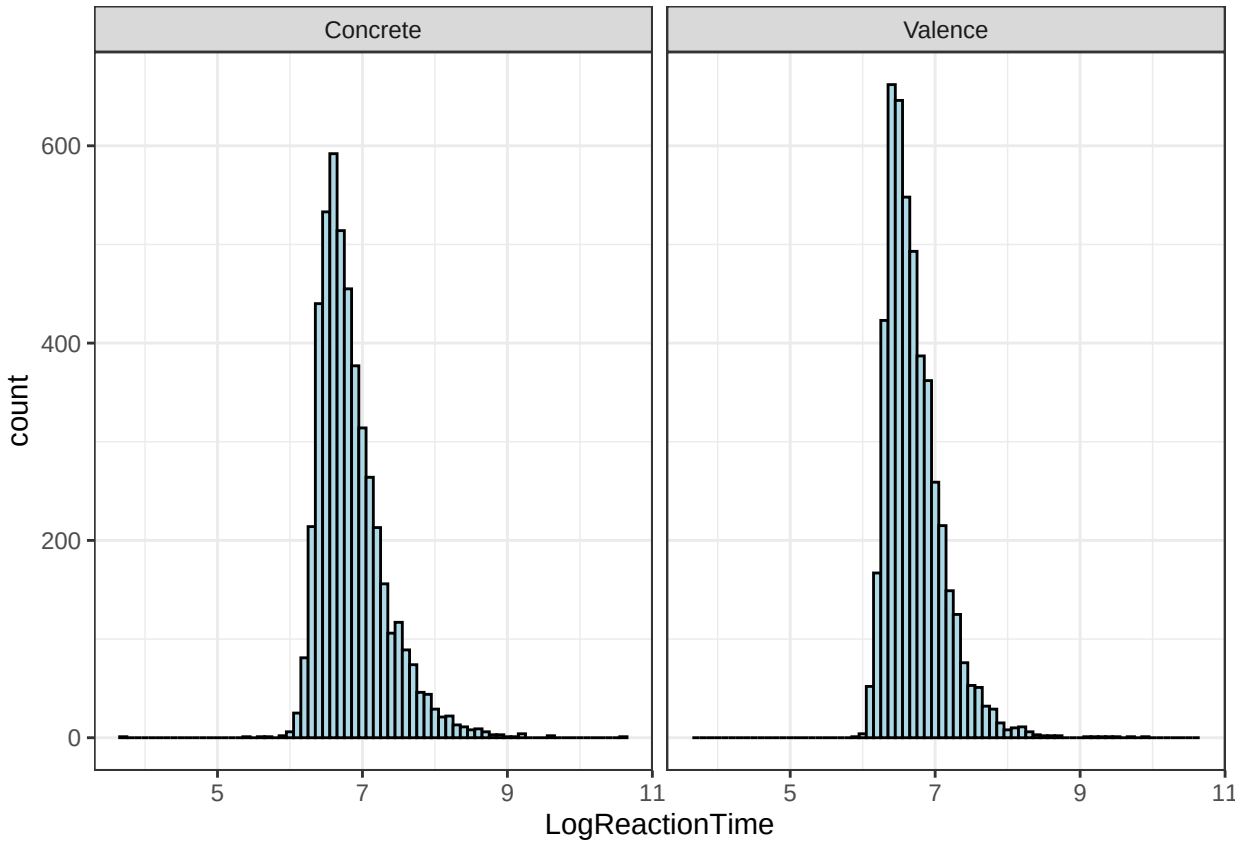
Verbs Conc-Abs: Analysis

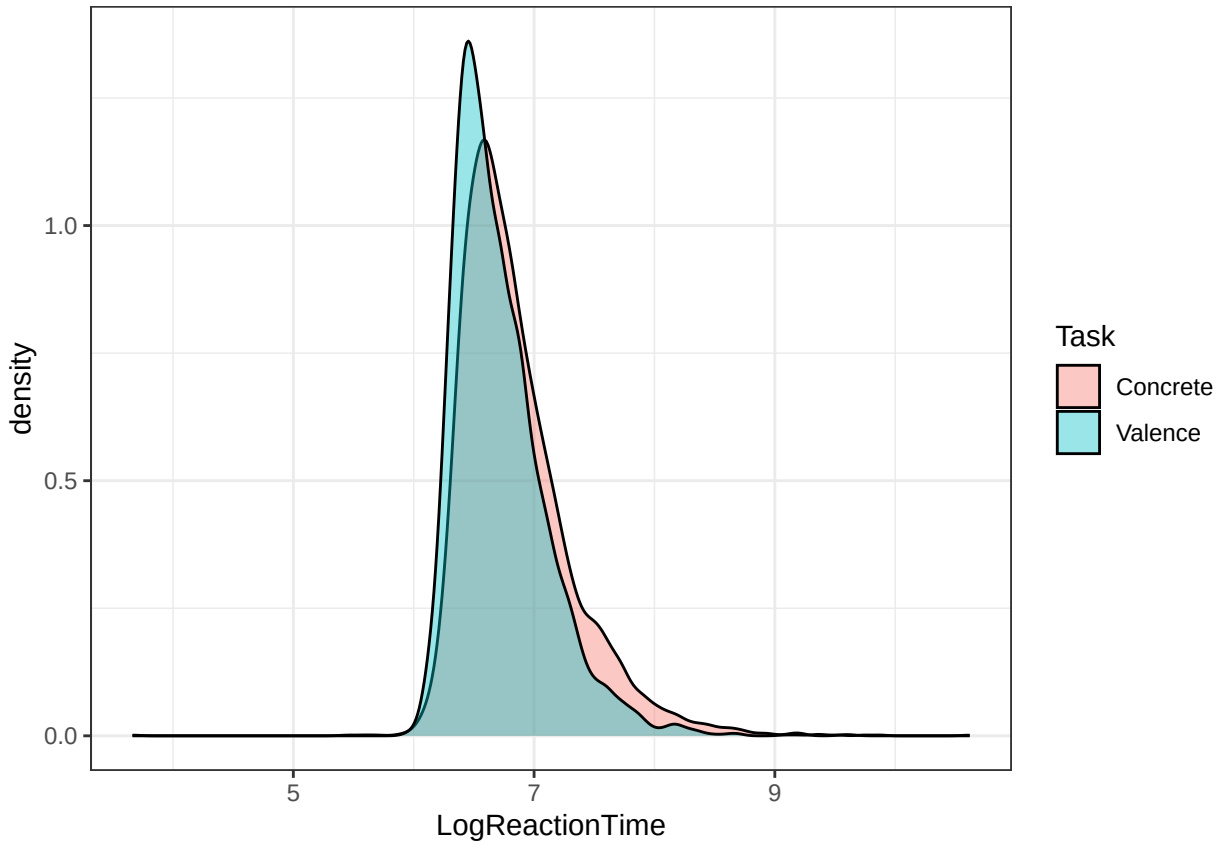
morgan moyer

2025-03-24

```
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: 7409.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
```

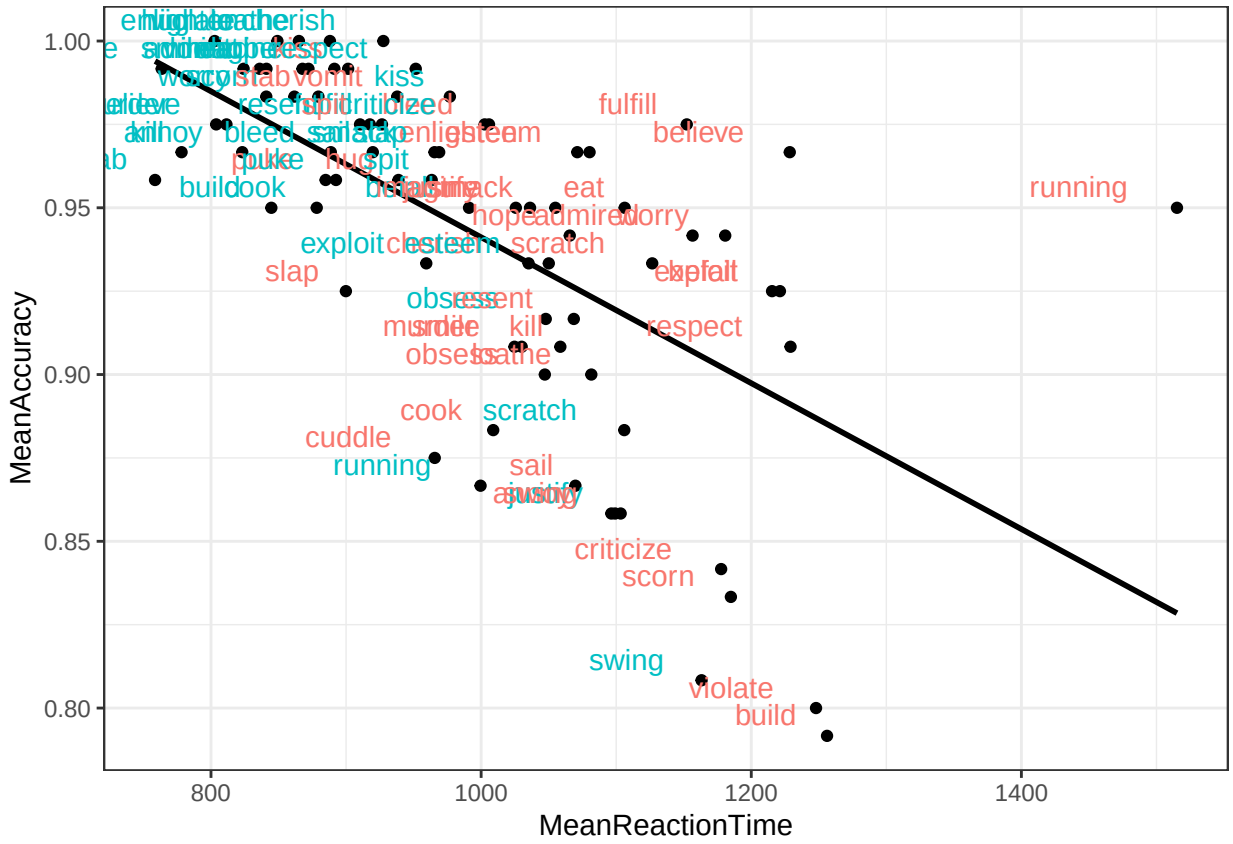
```
## -9.8359 -0.5999 -0.1593  0.4131  9.5533
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.003690 0.06075
##           cTask       0.006214 0.07883  0.00
##  ID.true  (Intercept) 0.047166 0.21718
##           cTask       0.028544 0.16895  -0.08
##  Residual             0.120935 0.34776
## Number of obs: 9600, groups:  Word, 40; ID.true, 40
##
## Fixed effects:
##                   Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)        6.78049    0.03583   44.93308 189.216  < 2e-16 ***
## cAccuracy         -0.14525    0.01678 9520.87440  -8.654  < 2e-16 ***
## cTask             -0.13001    0.03033   52.61728  -4.287 7.76e-05 ***
## cAccuracy:cTask    0.12953    0.03355 9520.62607   3.860 0.000114 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cAccrc cTask
## cAccuracy   -0.003
## cTask       -0.063 -0.022
## cAccrcy:cTs -0.009  0.310 -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) +
  # guides(legend = "none")
  theme(legend.position = "none")  # Remove the legend
```

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




```
ggsave("../graphs/exp1b_accXrt.pdf",width = 5, height = 3)
```

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

```
# 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)
```

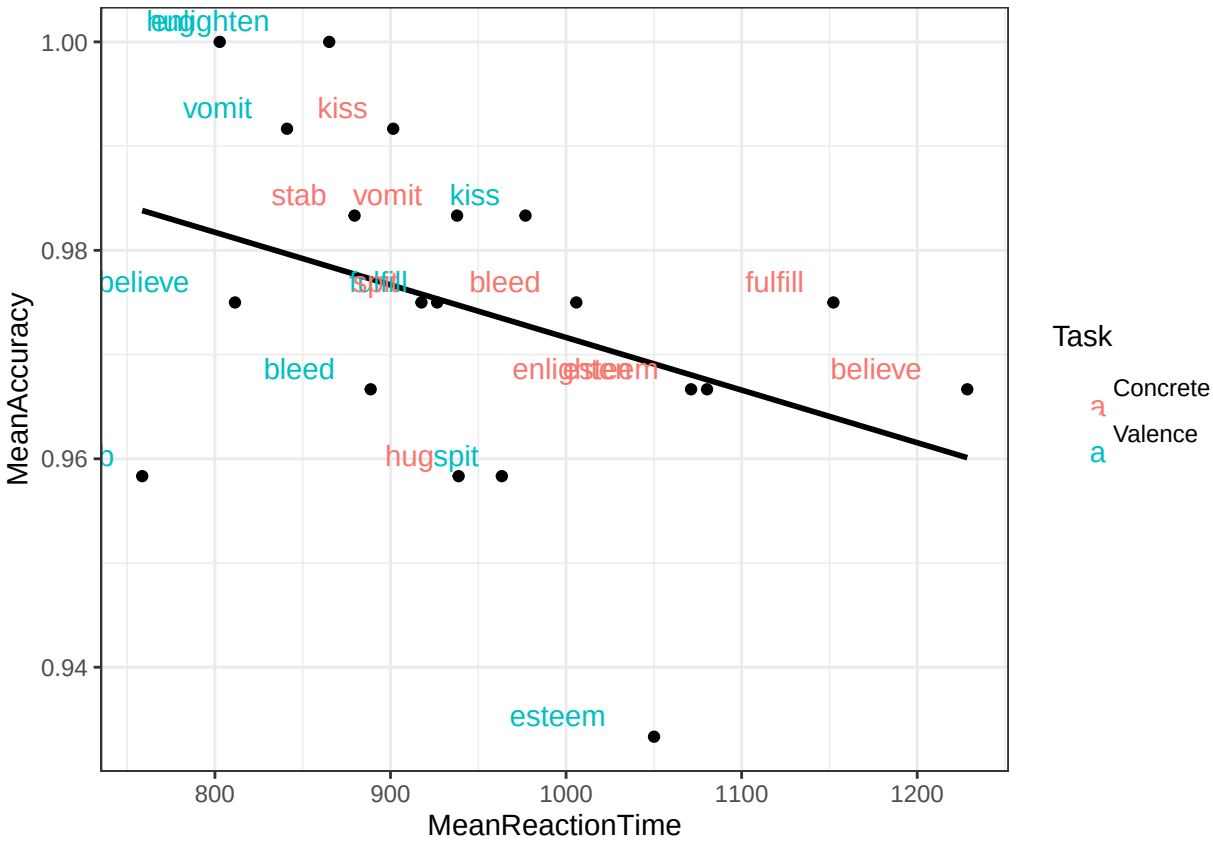
```
## `summarise()` has grouped output by 'Word'. You can override using the
## `.groups` argument.
```

```
View(concrete_accuracy)


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)
```

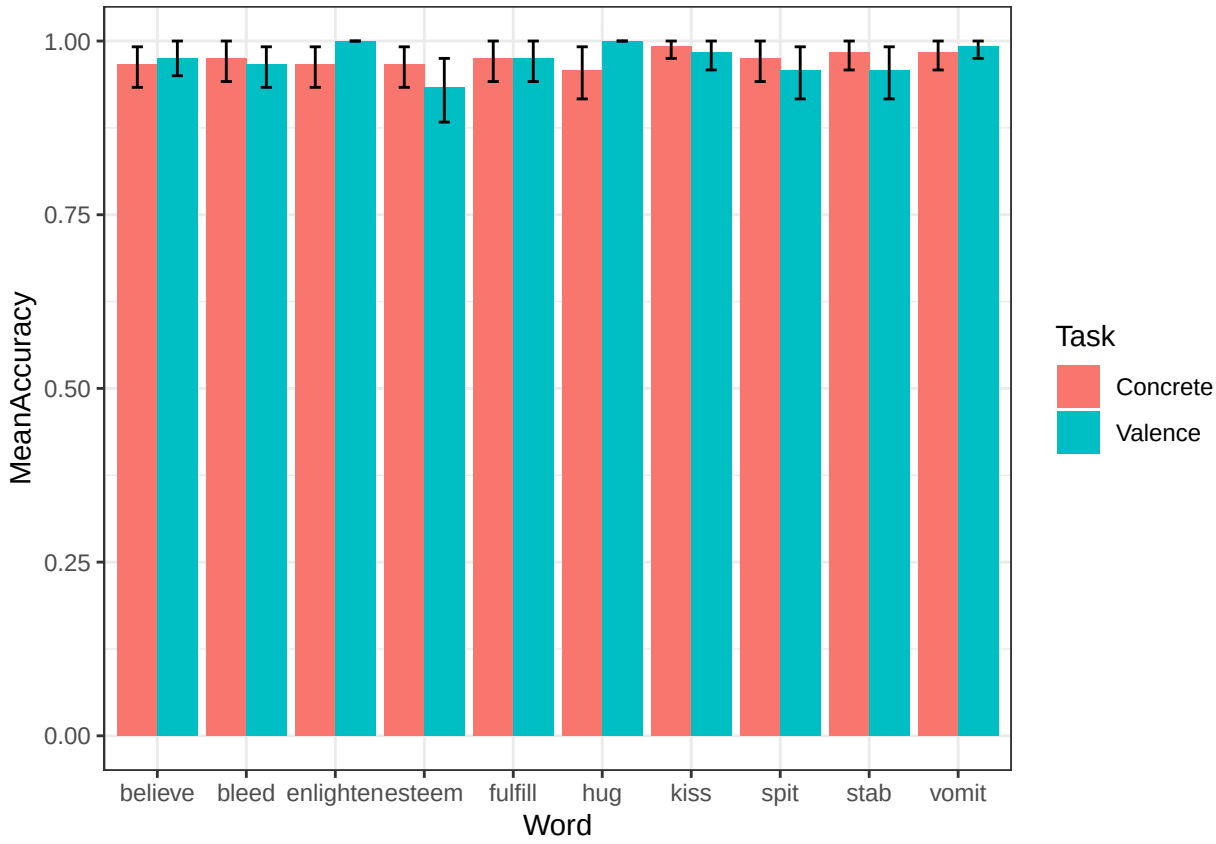
```
## `geom_smooth()` using formula = 'y ~ x'
```



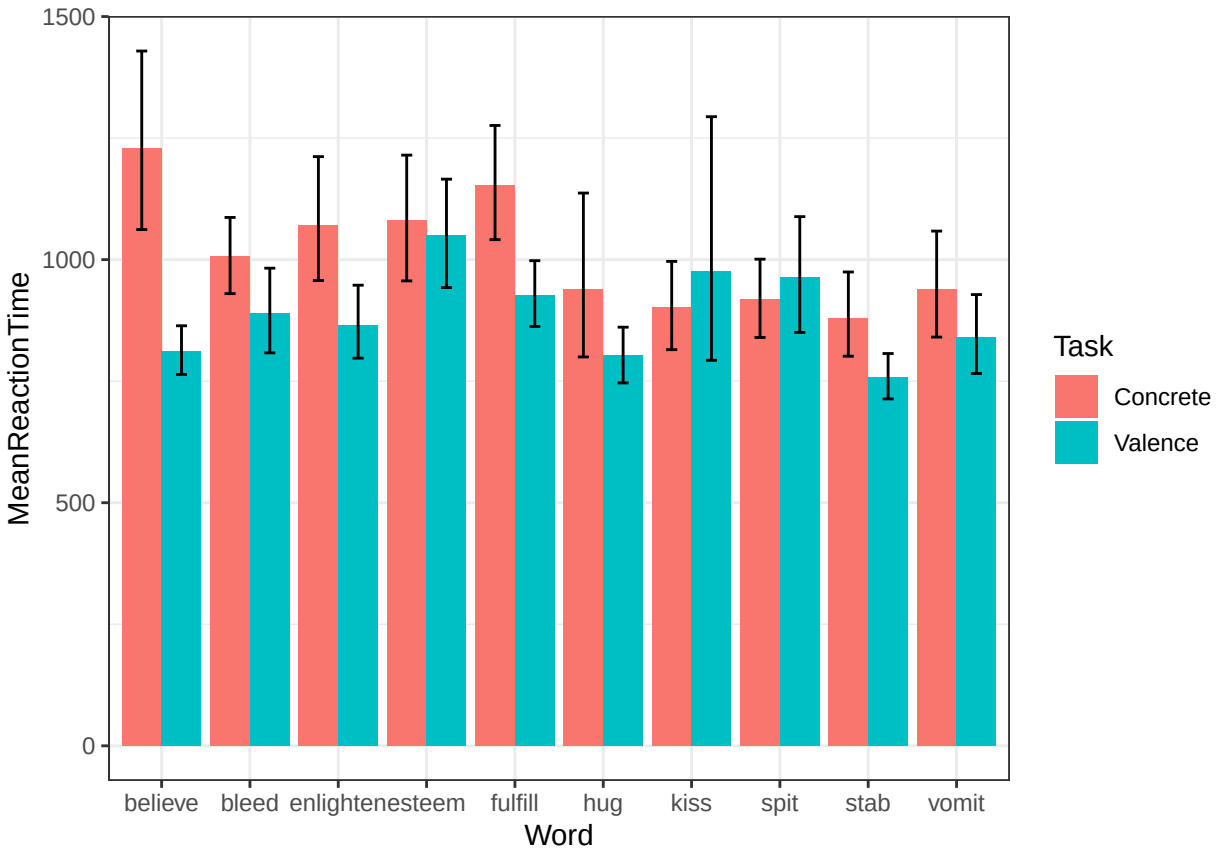

```
  # guides(legend = "none")
  # theme(legend.position = "none")  # Remove the legend
ggsave("../graphs/exp1b_accXrt.pdf",width = 5, height = 3)
```

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

```
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) %>%
  summarise(MeanAccuracy = mean(Accuracy)) %>%
  filter(MeanAccuracy < .75)

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

```
## [1] 0
```

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

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

```
## [1] 40
```

```
# remove all inaccurate trials
orig <- nrow(d.inaccurate.removed)
d.inaccurate.removed <- d.inaccurate.removed %>%
  filter(Accuracy == 1)
```

```r
nrow(d.inaccurate.removed)/orig*100
```

```
## [1] 94.19792
```

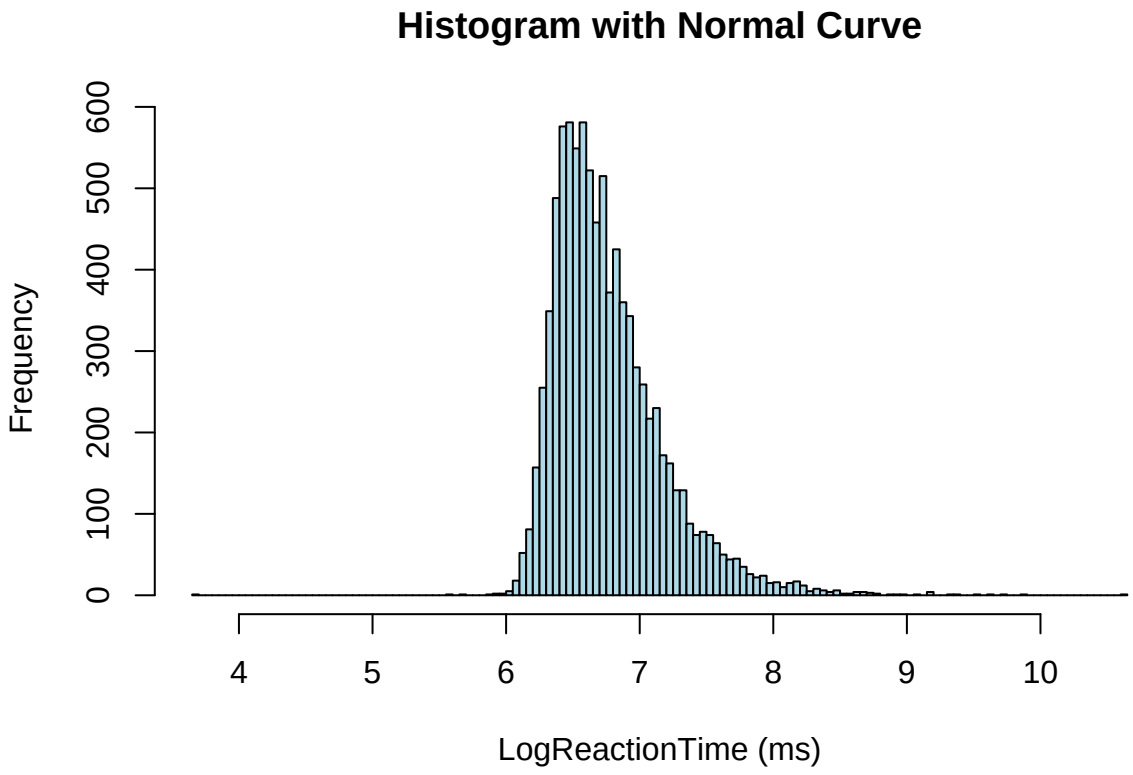
```r
# Remove subjects with ReactionTime higher than 3x IQR
summary(d.inaccurate.removed$LogReactionTime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.664   6.472   6.683   6.767   6.964  10.619
```

```r
  #  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]  3.663562 10.618714
```

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




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

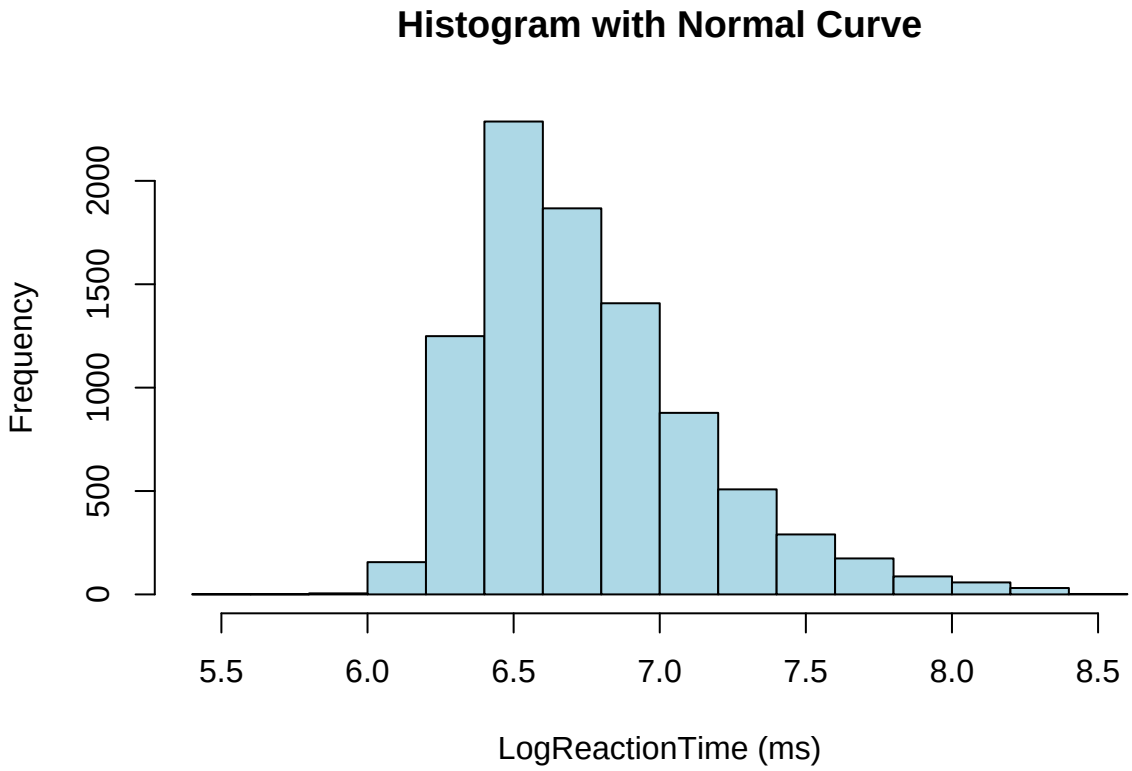
```
##        0%       25%       50%       75%      100%
##  3.663562  6.472346  6.683361  6.964136 10.618714
```

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

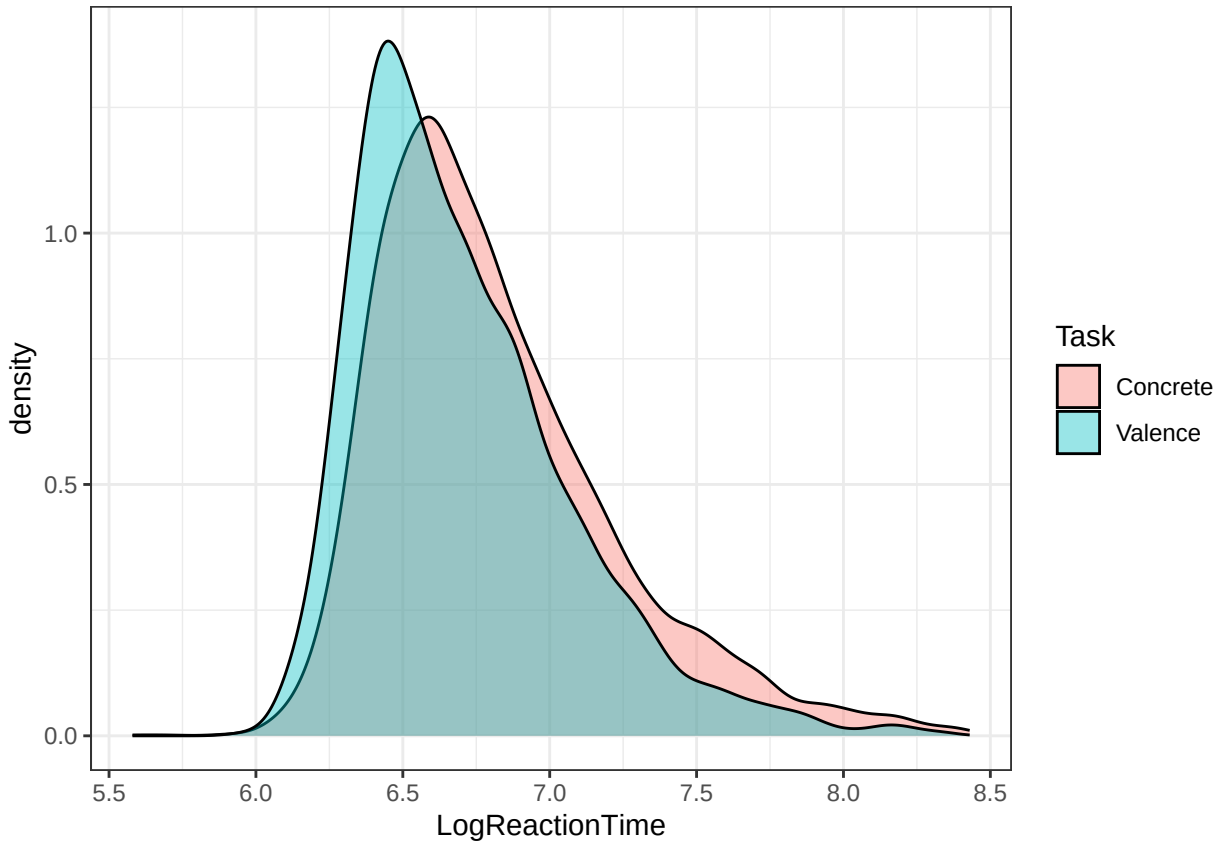
```
## [1] 1.475368
```

```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
```

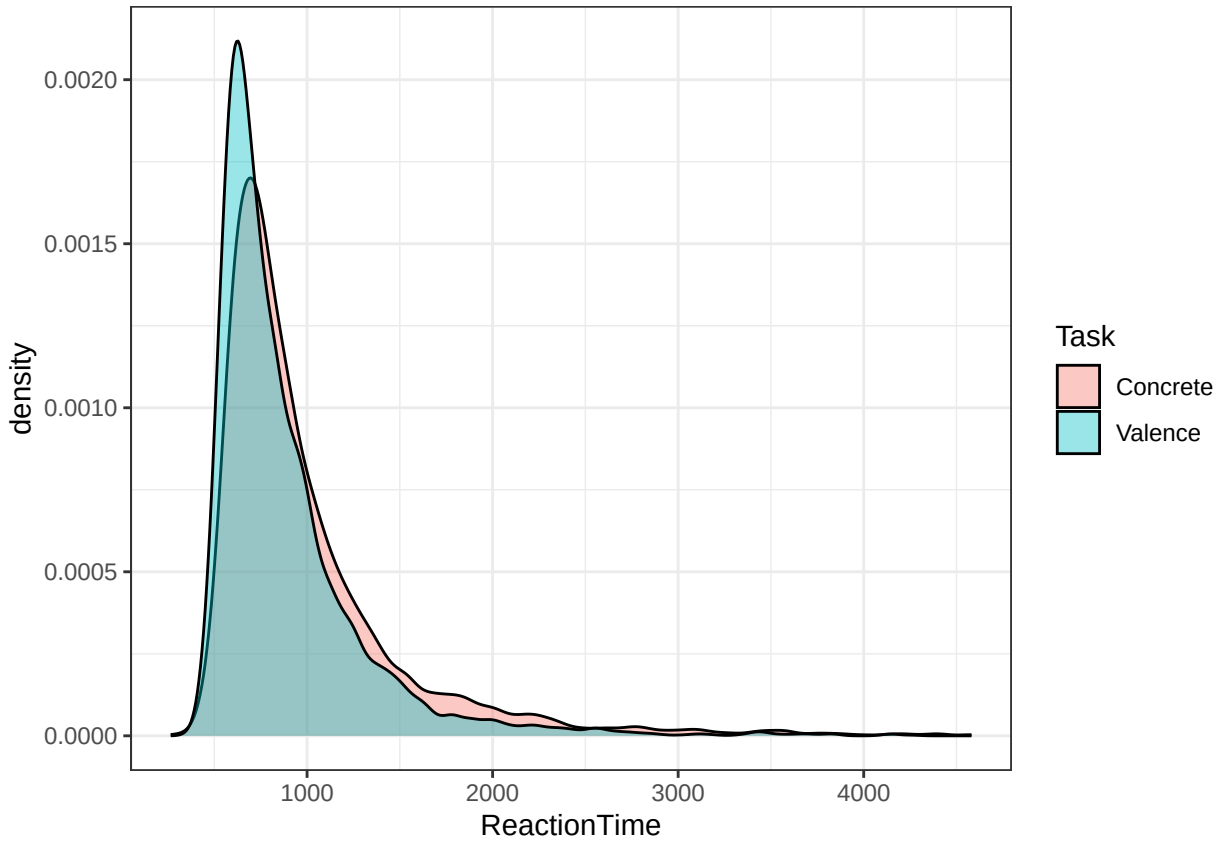
```r
df.outliers.removed <- subset(d.inaccurate.removed, (d.inaccurate.removed$LogReactionTime > cutoff.low)

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: 4949.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.4410 -0.6368 -0.1738  0.4459  5.1267
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  ID.true  (Intercept) 0.042593 0.20638
##           cTask       0.021685 0.14726  0.00
##  Word     (Intercept) 0.003296 0.05741
##           cTask       0.006609 0.08129  0.04
```

```
##  Residual             0.096594 0.31080
## Number of obs: 9002, groups:  ID.true, 40; Word, 40
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.76237    0.03403 44.87210   198.7  < 2e-16 ***
## cTask       -0.11779    0.02740 56.50604    -4.3 6.84e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##       (Intr)
## cTask 0.008
```

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_corrected.rds")

# m <- readRDS("../models/model-Task-ConcValCombo_outlier_excl_ReactionTime_corrected.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: 4855.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.4220 -0.6385 -0.1727  0.4445  5.3998
##
## Random effects:
##  Groups   Name                          Variance Std.Dev. Corr
##  ID.true  (Intercept)                   0.046069 0.21464
##           ConcValComboabstract-positive 0.002303 0.04799  -0.04
##           ConcValComboconcrete-negative 0.007090 0.08420  -0.34  0.43
##           ConcValComboconcrete-positive 0.007089 0.08420  -0.25  0.27  0.94
##           cTask                         0.022502 0.15001  -0.12  0.58  0.53
##  Word     (Intercept)                   0.001980 0.04450
##           cTask                         0.006300 0.07937  0.45
##  Residual                               0.094740 0.30780
##
##
##
##
##
##   0.24
##
```

```
##
##
## Number of obs: 9002, groups:  ID.true, 40; Word, 40
##
## Fixed effects:
##                                     Estimate Std. Error        df t value
## (Intercept)                         6.805942   0.037326 50.763795 182.337
## cTask                              -0.156566   0.036971 67.999578  -4.235
## ConcValComboabstract-positive      -0.008891   0.023203 41.290061  -0.383
## ConcValComboconcrete-negative      -0.091325   0.025662 52.538448  -3.559
## ConcValComboconcrete-positive      -0.069527   0.025688 52.532509  -2.707
## cTask:ConcValComboabstract-positive  0.006013   0.039989 35.589586   0.150
## cTask:ConcValComboconcrete-negative  0.068828   0.040013 35.672774   1.720
## cTask:ConcValComboconcrete-positive  0.072813   0.040085 35.926127   1.816
##                                    Pr(>|t|)
## (Intercept)                         < 2e-16 ***
## cTask                              7.03e-05 ***
## ConcValComboabstract-positive       0.70356
## ConcValComboconcrete-negative       0.00080 ***
## ConcValComboconcrete-positive       0.00915 **
## cTask:ConcValComboabstract-positive  0.88133
## cTask:ConcValComboconcrete-negative  0.09407 .
## cTask:ConcValComboconcrete-positive  0.07765 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                      (Intr) cTask  CncVlCmbb- CncVlCmbcncrt-n CncVlCmbcncrt-p
## cTask                 0.044
## CncVlCmbbs-          -0.290 -0.058
## CncVlCmbcncrt-n      -0.411  0.014  0.478
## CncVlCmbcncrt-p      -0.368 -0.085  0.451      0.619
## cTsk:CncVlCmbb-      -0.104 -0.544  0.338      0.151           0.151
## cTsk:CncVlCmbcncrt-n -0.103 -0.544  0.167      0.306           0.150
## cTsk:CncVlCmbcncrt-p -0.103 -0.543  0.166      0.150           0.303
##                      cTsk:CncVlCmbb- cTsk:CncVlCmbcncrt-n
## cTask
## CncVlCmbbs-
## CncVlCmbcncrt-n
## CncVlCmbcncrt-p
## cTsk:CncVlCmbb-
## cTsk:CncVlCmbcncrt-n  0.502
## cTsk:CncVlCmbcncrt-p  0.501           0.501
```

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: 5664.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9475 -0.6627 -0.1898  0.4584  5.2991
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  ID.true  (Intercept) 0.0411503 0.20286
##  Word     (Intercept) 0.0032216 0.05676
##           cBlockOrder 0.0002784 0.01668  -0.20
##  Residual             0.1065894 0.32648
## Number of obs: 9002, groups:  ID.true, 40; Word, 40
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.76006    0.03348 43.77160 201.889   <2e-16 ***
## cBlockOrder -0.08936    0.06457 38.11958  -1.384    0.174
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cBlockOrder -0.002
```

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: 5604.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0391 -0.6618 -0.1824  0.4553  5.5499
##
## Random effects:
##  Groups   Name                        Variance  Std.Dev. Corr
##  ID.true  (Intercept)                 0.0441056 0.21001
```

```
##           ConcValComboabstract-positive 0.0007609 0.02758   0.18
##           ConcValComboconcrete-negative 0.0051704 0.07191  -0.28  0.18
##           ConcValComboconcrete-positive 0.0055897 0.07476  -0.21  0.00  0.95
##  Word     (Intercept)                   0.0019771 0.04447
##  Residual                               0.1052272 0.32439
## Number of obs: 9002, groups:  ID.true, 40; Word, 40
##
## Fixed effects:
##                                Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)                    6.799018   0.036717 51.075736 185.172  < 2e-16
## ConcValComboabstract-positive -0.002779   0.022541 36.746648  -0.123  0.90255
## ConcValComboconcrete-negative -0.084016   0.024875 47.940709  -3.378  0.00146
## ConcValComboconcrete-positive -0.066836   0.025115 48.904312  -2.661  0.01050
##
## (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-    -0.266
## CncVlCmbcncrt-n -0.385  0.454
## CncVlCmbcncrt-p -0.355  0.433      0.597
```

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

- Nope

```
str(df_factors)
```

```
## 'data.frame':    9002 obs. of  20 variables:
##  $ X              : Factor w/ 9002 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ ID.true        : Factor w/ 40 levels "558a035bfdf99b2d75651378",..: 14 14 14 14 14 14 14 14 14 14
##  $ Word           : Factor w/ 40 levels "admired","annoy",..: 25 4 32 14 39 33 7 28 29 5 ...
##  $ Label          : Factor w/ 2 levels "test_conc","test_val": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ConcValCombo   : Factor w/ 4 levels "abstract-negative",..: 3 2 3 1 3 3 2 4 4 3 ...
##  $ Task           : Factor w/ 2 levels "Concrete","Valence": 1 1 1 1 1 1 1 1 1 1 ...
##  $ BlockOrder     : Factor w/ 2 levels "CV","VC": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Group          : Factor w/ 2 levels "A","B": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Response       : Factor w/ 4 levels "abstract","concrete",..: 2 1 2 1 2 2 1 2 2 2 ...
##  $ Accuracy       : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ EventTime      : Factor w/ 9002 levels "1732203168805",..: 6512 6513 6514 6515 6516 6517 6518 6519
##  $ Value          : Factor w/ 4 levels "abstract","concrete",..: 2 1 2 1 2 2 1 2 2 2 ...
##  $ RT             : Factor w/ 1385 levels "1270","1287.83333333333",..: 311 990 749 385 531 149 50 54
##  $ ReactionTime   : int  965 1809 1020 799 715 754 668 1252 719 1205 ...
##  $ Key_value_F    : Factor w/ 4 levels "abstract","concrete",..: 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  6.87 7.5 6.93 6.68 6.57 ...
##  $ LogRT          : Factor w/ 1385 levels "7.14677217945264",..: 311 990 749 385 531 149 50 542 662 7
```

```
##  $ TrialNumber    : Factor w/ 240 levels "1","2","3","4",..: 1 2 3 4 8 9 10 11 12 14 ...
```

```
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)
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: 3016.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2380 -0.6385 -0.1703  0.4680  5.1398
##
## Random effects:
##  Groups   Name          Variance  Std.Dev. Corr
##  ID.true  (Intercept)   0.0487592 0.22081
##           cConcValCombo 0.0031780 0.05637  -0.16
##  Word     (Intercept)   0.0022190 0.04711
##           cConcValCombo 0.0003197 0.01788  0.58
##  Residual               0.1092111 0.33047
## Number of obs: 4396, groups:  ID.true, 40; Word, 40
##
## Fixed effects:
##               Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)    6.82551    0.03618 42.86888 188.651  < 2e-16 ***
## cConcValCombo -0.04647    0.01241 45.00765  -3.745  0.00051 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cConcValCmb -0.062
```

```
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: 2983
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2902 -0.6387 -0.1801  0.4684  5.1300
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  ID.true  (Intercept) 0.048623 0.22051
##           cSemantic   0.018249 0.13509  -0.21
##  Word     (Intercept) 0.001838 0.04287
##  Residual             0.108476 0.32936
## Number of obs: 4396, groups:  ID.true, 40; Word, 40
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.82580    0.03587 41.76458 190.290  < 2e-16 ***
## cSemantic   -0.11175    0.02721 53.30476  -4.106 0.000139 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## cSemantic -0.163
```

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| = 0.00331201 (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: 1756.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9001 -0.6409 -0.1724  0.4253  5.5412
##
## Random effects:
##  Groups   Name          Variance  Std.Dev. Corr
##  ID.true  (Intercept)   0.0479600 0.21900
##           cConcValCombo 0.0003191 0.01786  0.08
##  Word     (Intercept)   0.0040271 0.06346
##           cConcValCombo 0.0008182 0.02860  0.27
##  Residual               0.0807067 0.28409
## Number of obs: 4606, groups:  ID.true, 40; Word, 40
##
## Fixed effects:
##               Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)    6.70438    0.03660 46.62800 183.170   <2e-16 ***
## cConcValCombo -0.01744    0.01153 26.90471  -1.513    0.142
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cConcValCmb 0.069
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00331201 (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: 1759.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9972 -0.6376 -0.1684  0.4222  5.6141
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  ID.true  (Intercept) 0.048008 0.21911
##           cValence    0.001278 0.03575  0.15
##  Word     (Intercept) 0.005371 0.07328
##  Residual             0.080784 0.28423
## Number of obs: 4606, groups:  ID.true, 40; Word, 40
##
## Fixed effects:
##              Estimate Std. Error        df t value Pr(>|t|)
```

```
## (Intercept)  6.705000   0.036770 47.318316 182.348   <2e-16 ***
## cValence     0.008149   0.025286 40.503327   0.322    0.749
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## cValence 0.032
```