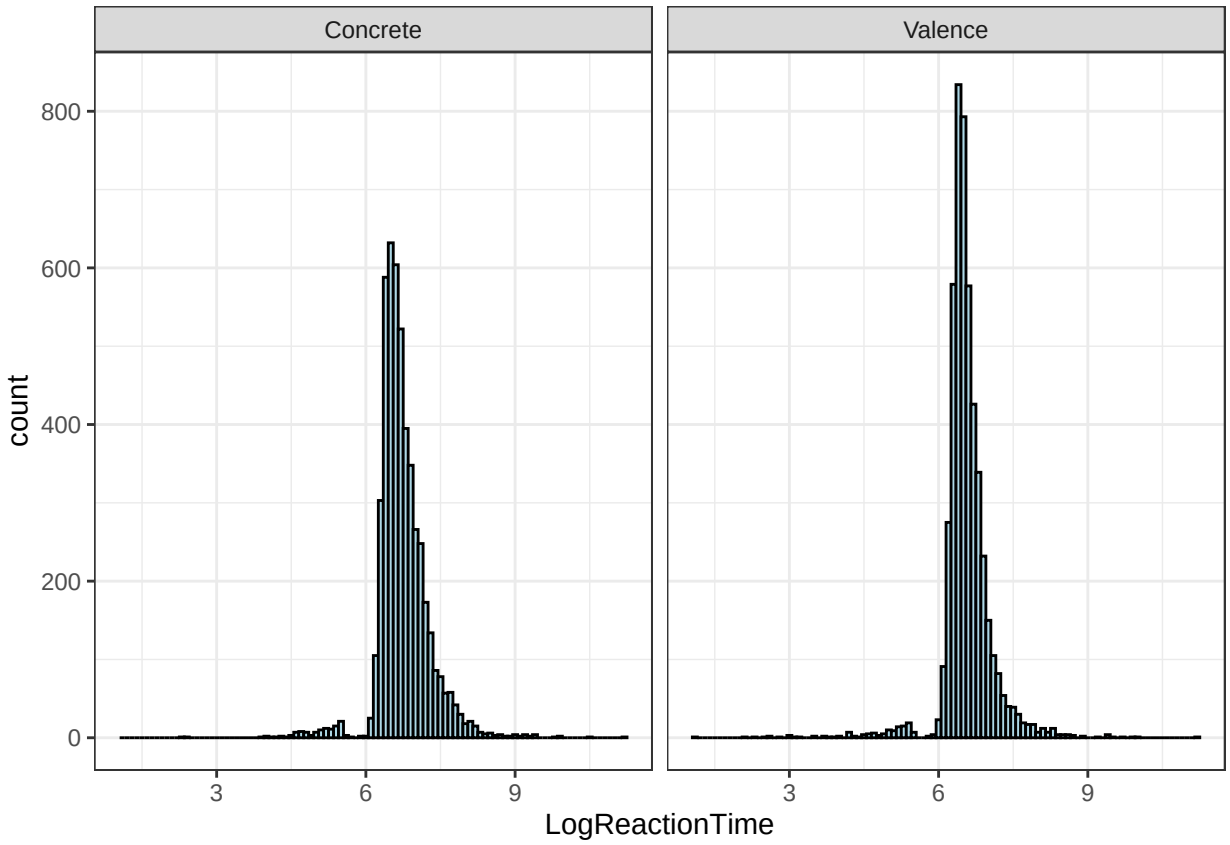
# Nouns Conc-Abs: Analysis without outliers: ReactionTime

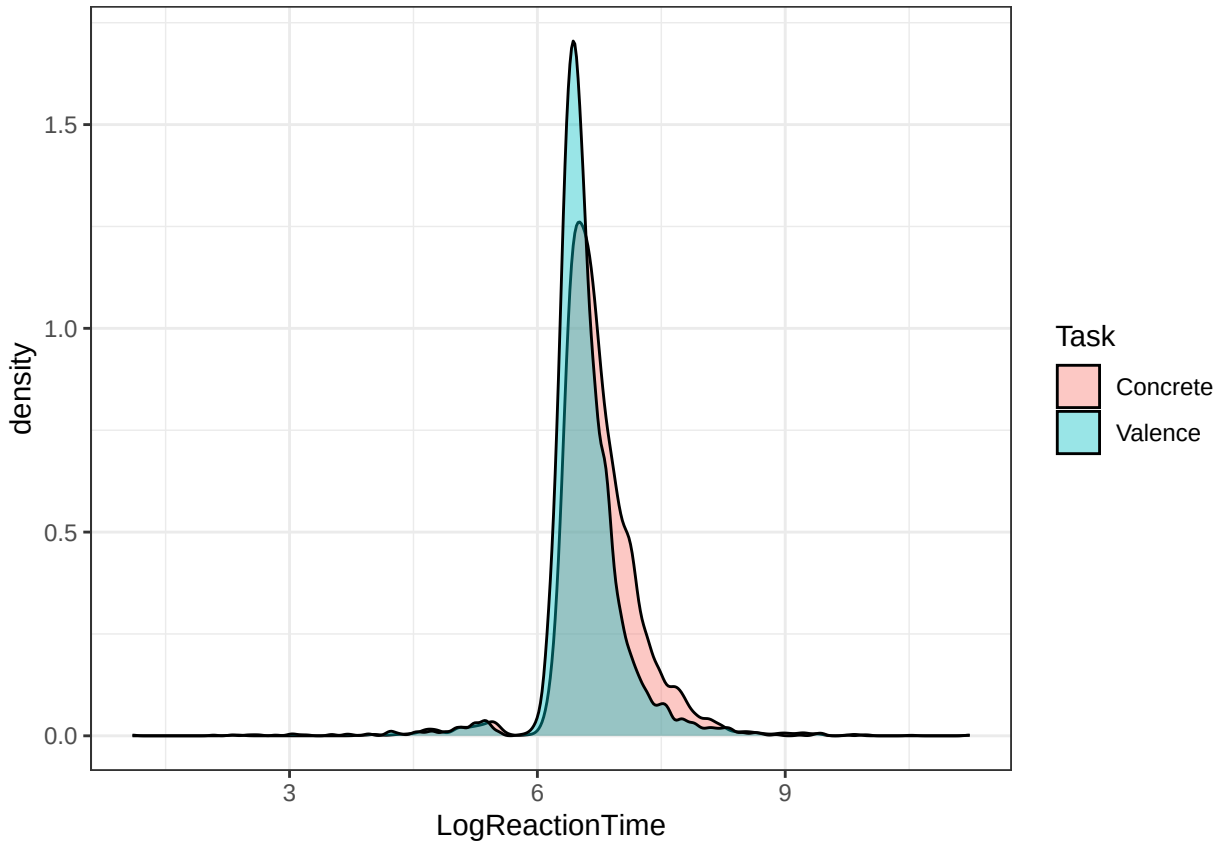
morgan moyer

2025-03-25

```
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: 9610.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
```

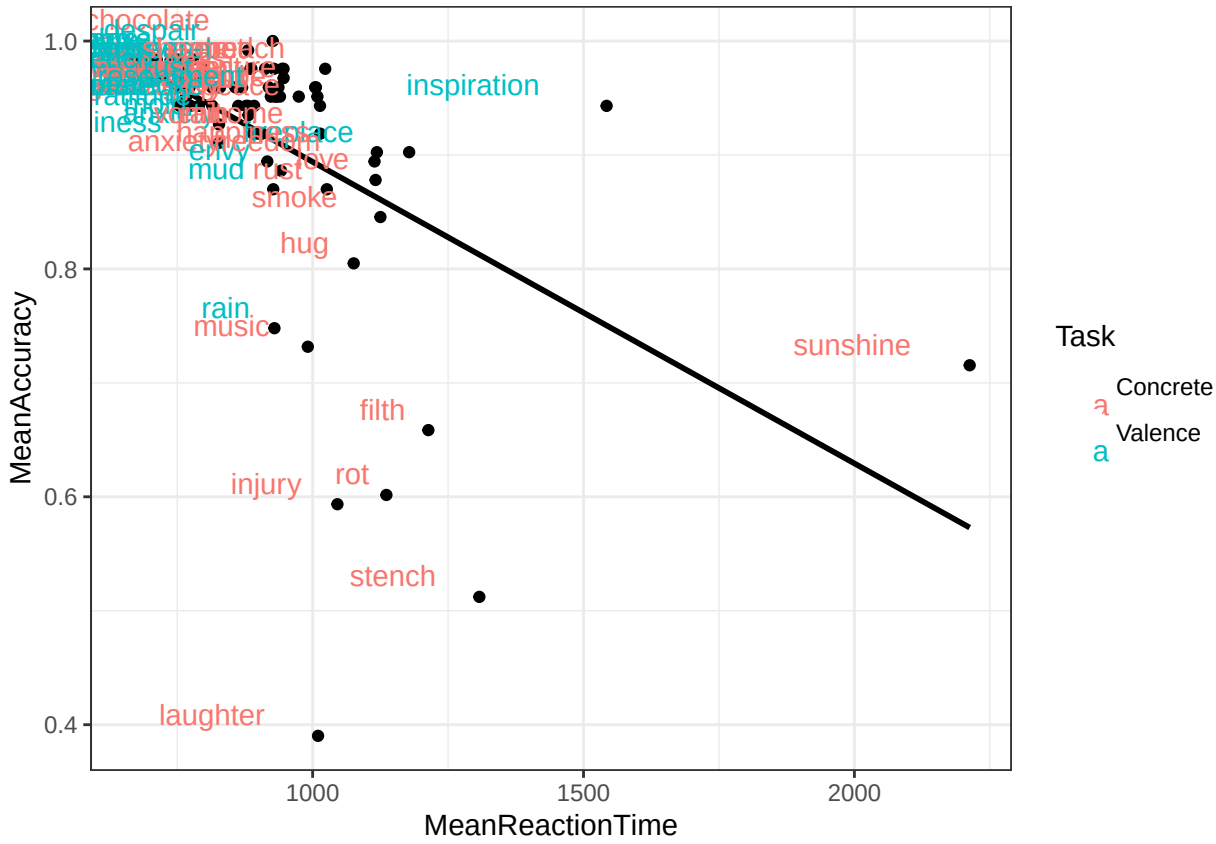
```
## -11.9217  -0.5415  -0.1484   0.3698  11.4676
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.001831 0.04279
##           cTask       0.016931 0.13012  -0.18
##  ID.true  (Intercept) 0.110705 0.33272
##           cTask       0.020169 0.14202  0.00
##  Residual             0.148540 0.38541
## Number of obs: 9840, groups:  Word, 40; ID.true, 40
##
## Fixed effects:
##                   Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)      6.650e+00  5.319e-02  4.025e+01 125.033  < 2e-16 ***
## cAccuracy       -2.909e-03  1.693e-02  9.603e+03  -0.172    0.864
## cTask           -1.777e-01  3.147e-02  6.854e+01  -5.646 3.42e-07 ***
## cAccuracy:cTask  3.184e-02  3.364e-02  9.214e+03   0.947    0.344
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cAccrc cTask
## cAccuracy   -0.004
## cTask       -0.015 -0.039
## cAccrcy:cTs -0.012  0.347 -0.014
```

```
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)
```

```
# 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)
```

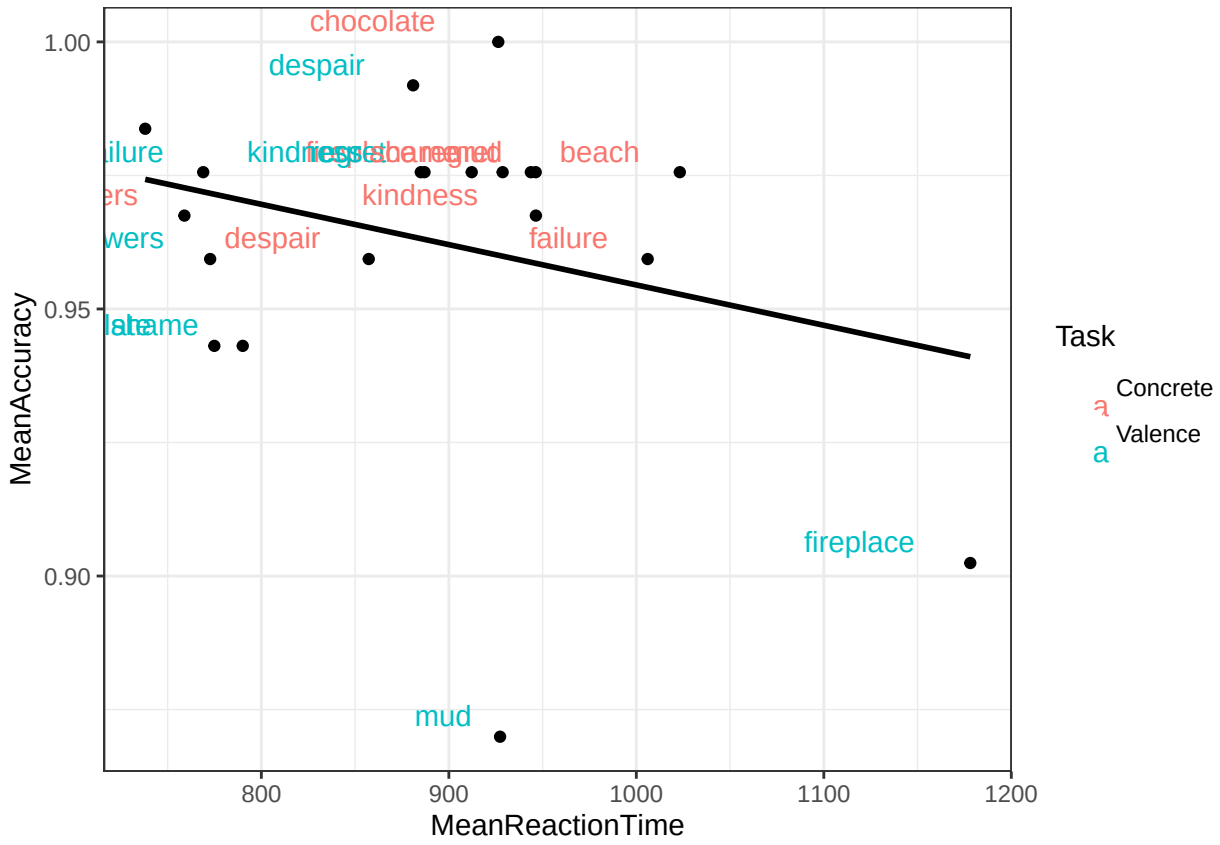
```
## `summarise()` has grouped output by 'Word'. You can override using the
## `.groups` argument.
```

```
# 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.
```

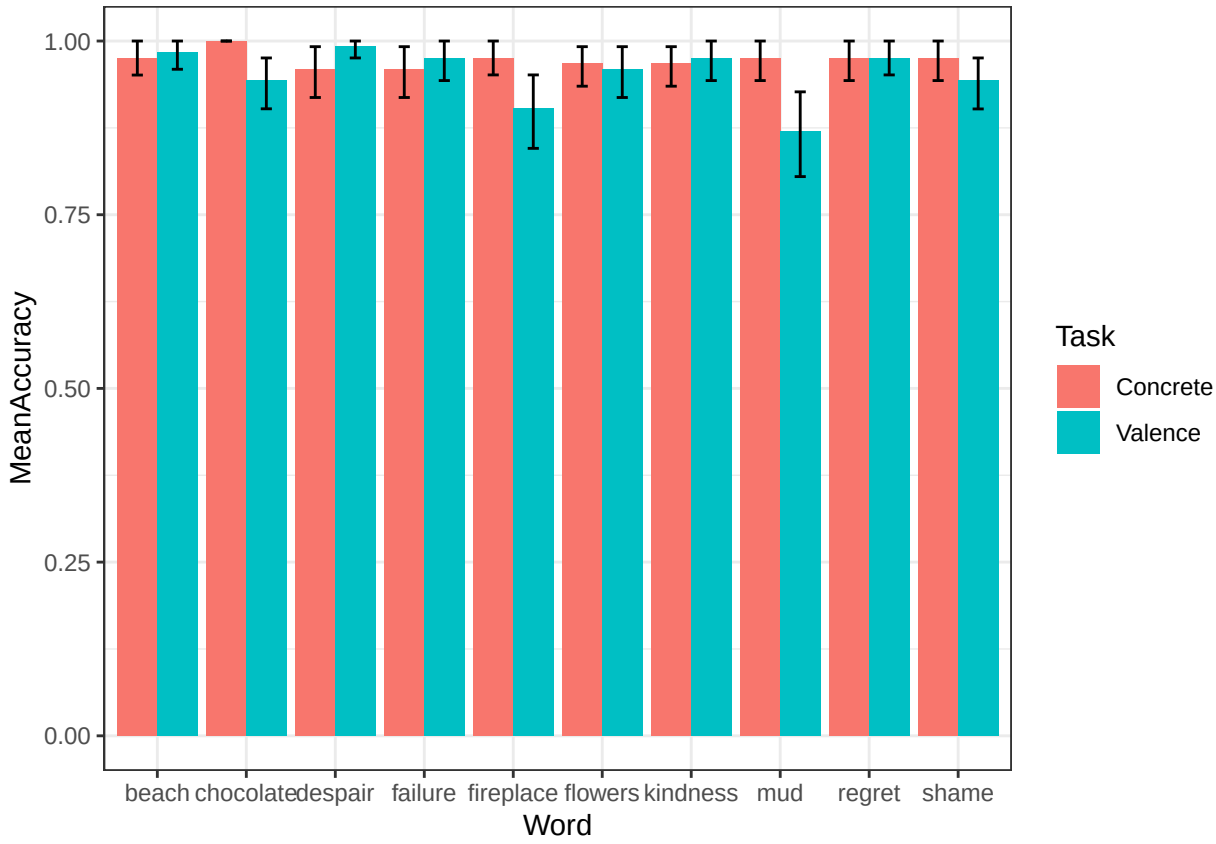
```
ggplot(agr, aes(x = MeanReactionTime, y = MeanAccuracy)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  geom_text(aes(label = Word, color = Task), vjust = -0.5, hjust = 1.5)
```

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

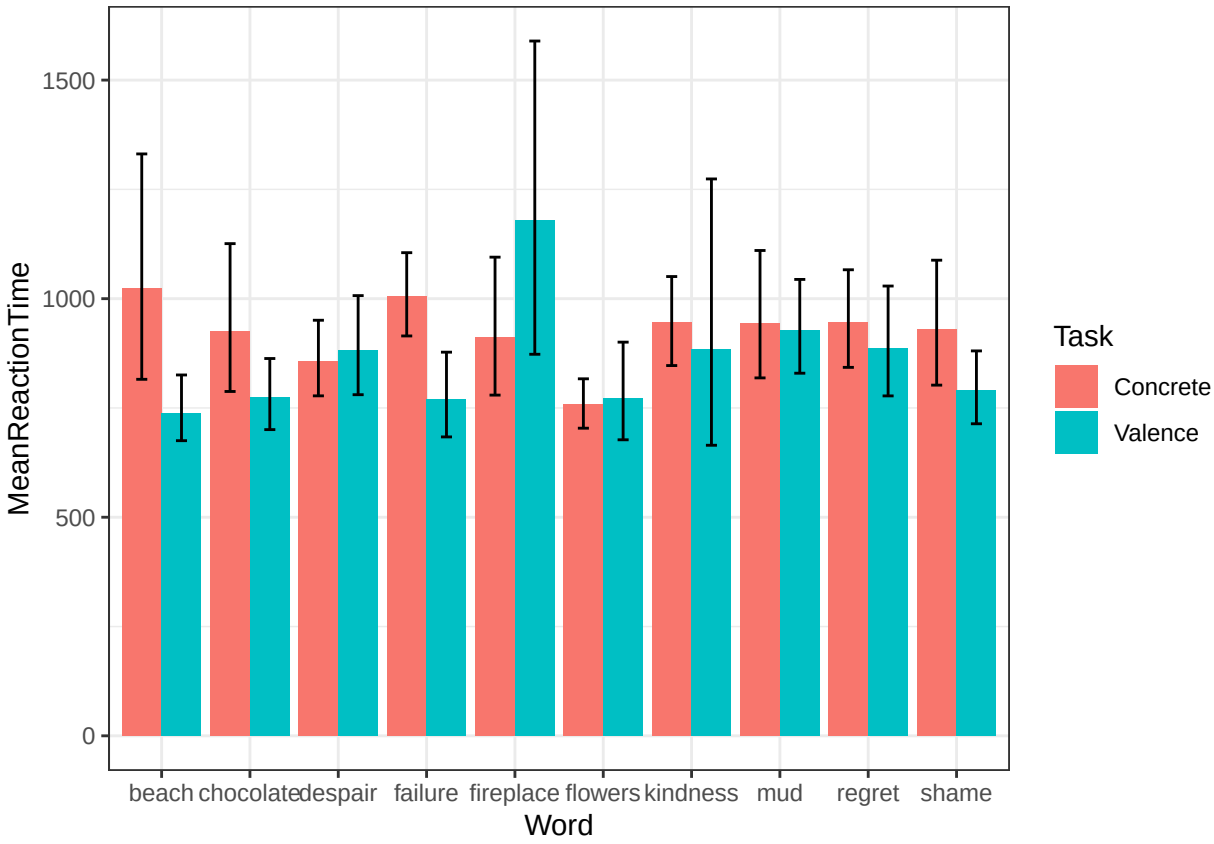



```
  # guides(legend = "none")
  # theme(legend.position = "none")  # Remove the legend
# ggsave("../graphs/exp1b_accXrt.pdf",width = 5, height = 3)
```

```
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] 2
```

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

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

```
## [1] 38
```

## 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] 92.87393
```

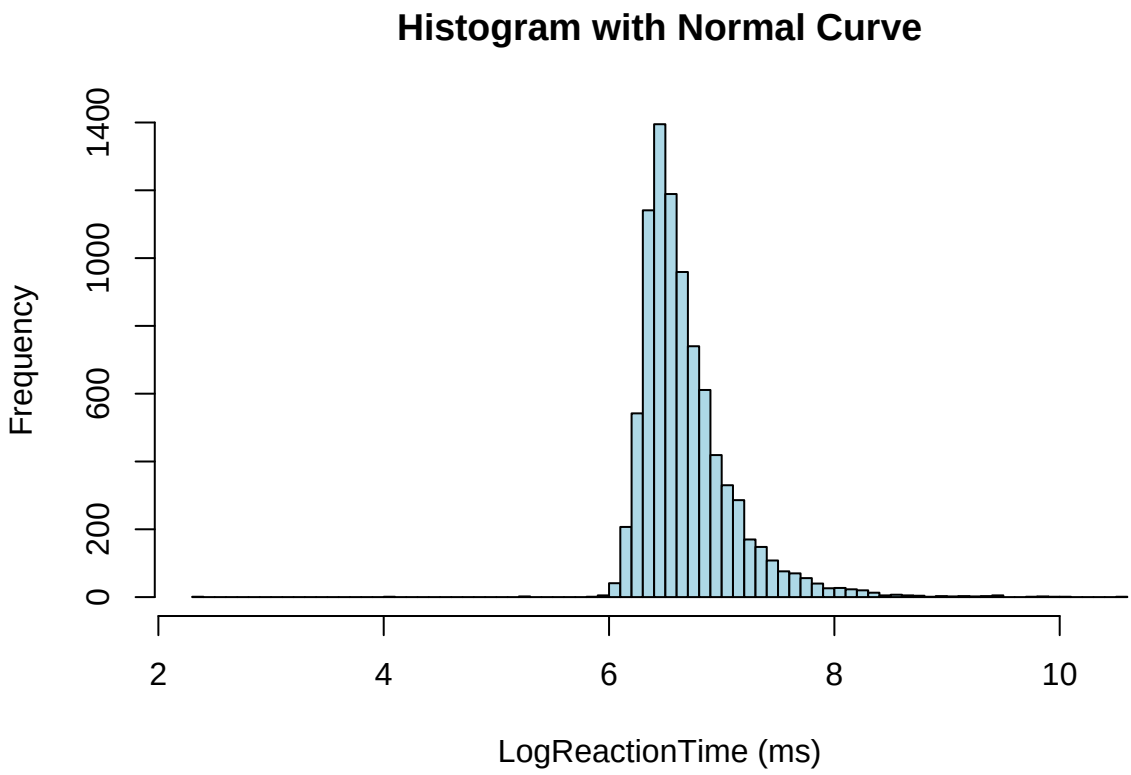
```
# Remove subjects with ReactionTime higher than 3x IQR
summary(d.inaccurate.removed$LogReactionTime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.303   6.417   6.583   6.685   6.844  10.528
```

```
  #  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]  2.302585 10.527553
```

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



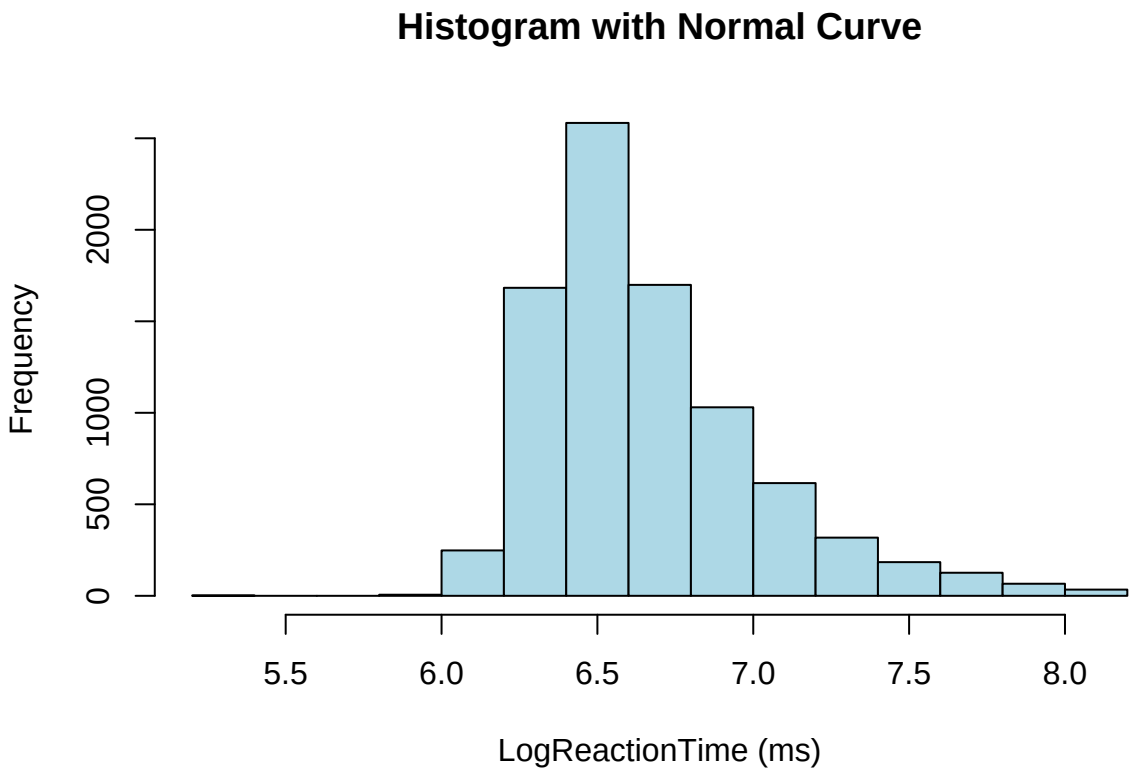

```
quantile(d.inaccurate.removed$LogReactionTime)
```

```
##        0%       25%       50%       75%      100%
##  2.302585  6.416732  6.583409  6.843750 10.527553
```

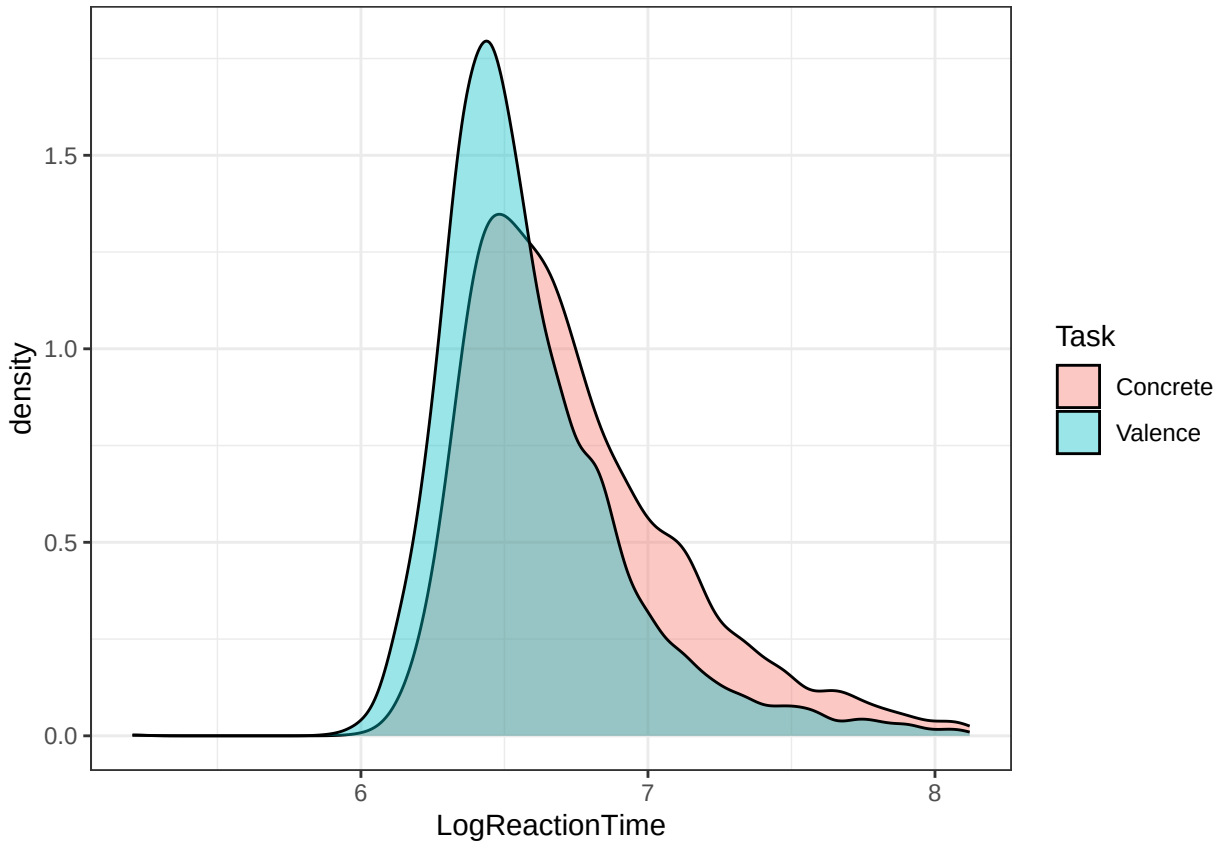
```
IQR(d.inaccurate.removed$LogReactionTime)*3 # 0.7526289
```

```
## [1] 1.281053
```

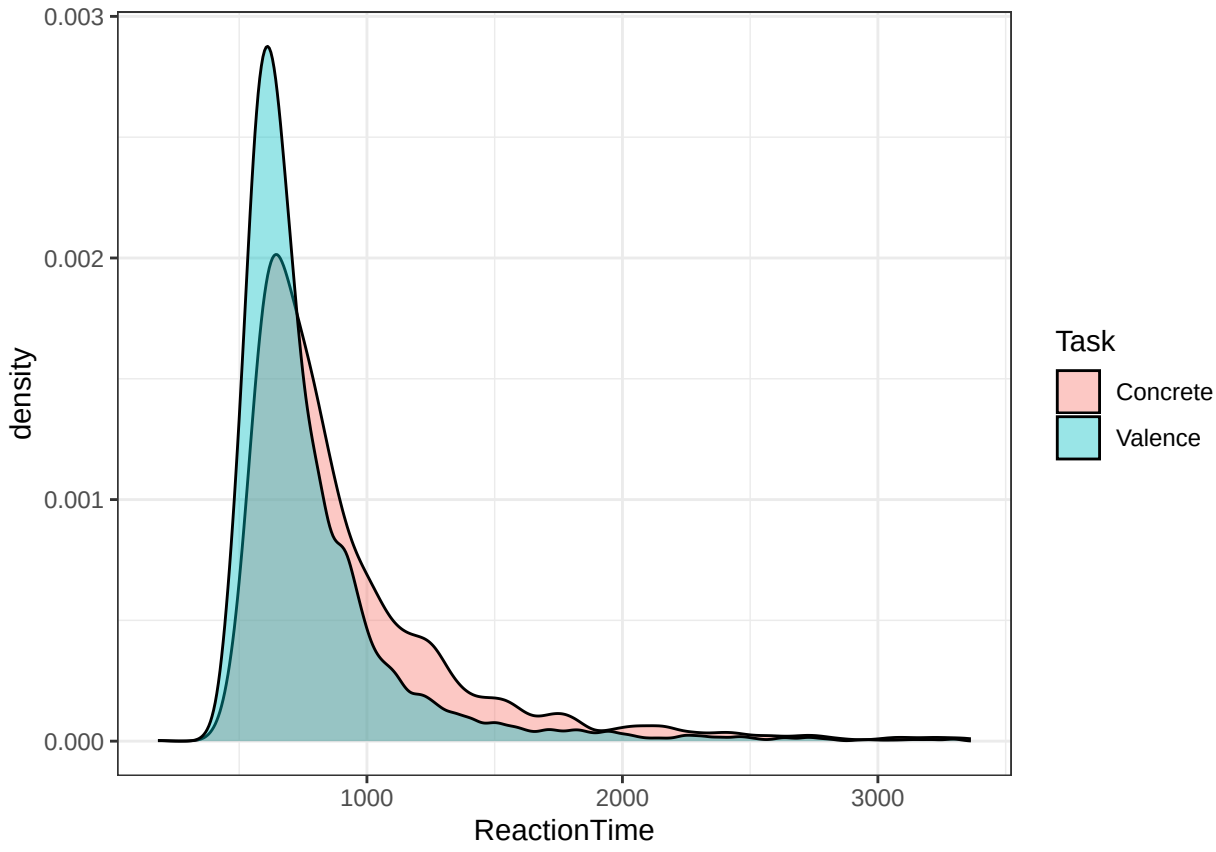
```r
cutoff.high <- quantile(d.inaccurate.removed$LogReactionTime)[4] + IQR(d.inaccurate.removed$LogReaction
cutoff.low <- quantile(d.inaccurate.removed$LogReactionTime)[2] - IQR(d.inaccurate.removed$LogReactionT


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

hist(df.outliers.removed$LogReactionTime, col="lightblue", xlab="LogReactionTime (ms)",
        main="Histogram with Normal Curve")
```




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




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




## convert everything to factors

## Is there a difference between Semantic and Valence Tasks?

Yes

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

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cTask + (1 + cTask | ID.true) + (1 + cTask |
##     Word)
##    Data: center
##
## REML criterion at convergence: 3815
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.6966 -0.6296 -0.1712  0.4579  4.9763
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.002426 0.04925
##           cTask       0.015390 0.12406  -0.35
##  ID.true  (Intercept) 0.025989 0.16121
```

```
##           cTask       0.018896 0.13746  -0.34
##  Residual             0.086842 0.29469
## Number of obs: 8596, groups:  Word, 40; ID.true, 38
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.67343    0.02748 43.28457 242.868  < 2e-16 ***
## cTask       -0.16943    0.03040 67.39518  -5.574 4.74e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##       (Intr)
## cTask -0.304
```

## 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
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00319275 (tol = 0.002, component 1)
```

```
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: 3782.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.4796 -0.6263 -0.1720  0.4528  4.9739
##
## Random effects:
##  Groups   Name                           Variance Std.Dev. Corr
##  Word     (Intercept)                    0.001906 0.04366
##           cTask                          0.013804 0.11749  -0.37
##  ID.true  (Intercept)                    0.028097 0.16762
##           ConcValComboabstract-positive 0.002085 0.04567  -0.44
##           ConcValComboconcrete-negative 0.002275 0.04769   0.11  0.14
##           ConcValComboconcrete-positive 0.004562 0.06754  -0.32  0.51  0.77
##           cTask                          0.018893 0.13745  -0.32 -0.23  0.03
##  Residual                                0.085746 0.29282
##
##
```

```
##
##
##
##
##
##   0.04
##
## Number of obs: 8596, groups:  Word, 40; ID.true, 38
##
## Fixed effects:
##                                    Estimate Std. Error       df t value
## (Intercept)                         6.67242    0.03113 52.25312 214.363
## cTask                              -0.08248    0.04508 54.08582  -1.830
## ConcValComboabstract-positive      -0.01989    0.02266 39.64024  -0.878
## ConcValComboconcrete-negative       0.04753    0.02288 40.76562   2.077
## ConcValComboconcrete-positive      -0.02013    0.02415 46.35046  -0.834
## cTask:ConcValComboabstract-positive -0.12481    0.05540 34.36253  -2.253
## cTask:ConcValComboconcrete-negative -0.14585    0.05559 34.83238  -2.624
## cTask:ConcValComboconcrete-positive -0.08041    0.05556 34.74630  -1.447
##                                    Pr(>|t|)
## (Intercept)                          <2e-16 ***
## cTask                                0.0728 .
## ConcValComboabstract-positive        0.3854
## ConcValComboconcrete-negative        0.0441 *
## ConcValComboconcrete-positive        0.4088
## cTask:ConcValComboabstract-positive  0.0308 *
## cTask:ConcValComboconcrete-negative  0.0128 *
## cTask:ConcValComboconcrete-positive  0.1568
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                      (Intr) cTask  CncVlCmbb- CncVlCmbcncrt-n CncVlCmbcncrt-p
## cTask                -0.270
## CncVlCmbbs-          -0.450  0.146
## CncVlCmbcncrt-n      -0.291  0.187  0.457
## CncVlCmbcncrt-p      -0.432  0.181  0.496      0.534
## cTsk:CncVlCmbb-       0.108 -0.614 -0.300     -0.148          -0.140
## cTsk:CncVlCmbcncrt-n  0.108 -0.612 -0.149     -0.303          -0.140
## cTsk:CncVlCmbcncrt-p  0.108 -0.613 -0.149     -0.147          -0.285
##                      cTsk:CncVlCmbb- cTsk:CncVlCmbcncrt-n
## cTask
## CncVlCmbbs-
## CncVlCmbcncrt-n
## CncVlCmbcncrt-p
## cTsk:CncVlCmbb-
## cTsk:CncVlCmbcncrt-n  0.498
## cTsk:CncVlCmbcncrt-p  0.499           0.497
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00319275 (tol = 0.002, component 1)
```

## 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: 4917.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.0713 -0.6478 -0.1866  0.4659  4.6172
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Word     (Intercept) 0.001606 0.04007
##  ID.true  (Intercept) 0.025535 0.15980
##  Residual             0.101170 0.31807
## Number of obs: 8596, groups:  Word, 40; ID.true, 38
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.66728    0.02691 41.20476   247.8   <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: 4917
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -5.0798 -0.6481 -0.1844  0.4652  4.6384
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  Word     (Intercept) 1.603e-03 0.040039
##           cBlockOrder 9.592e-05 0.009794 -0.67
##  ID.true  (Intercept) 2.346e-02 0.153157
##  Residual             1.011e-01 0.318037
## Number of obs: 8596, groups:  Word, 40; ID.true, 38
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.66561    0.02588 40.43042 257.515   <2e-16 ***
## cBlockOrder -0.10349    0.05026 36.02080  -2.059   0.0468 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cBlockOrder 0.026
```

## effect of ConcValCombo on ReactionTime?

nope.

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

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0198304 (tol = 0.002, component 1)
```

```
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: 4901.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.9521 -0.6444 -0.1862  0.4637  4.6411
##
## Random effects:
##  Groups   Name                              Variance Std.Dev. Corr
##  Word     (Intercept)                       0.001129 0.03360
##  ID.true  (Intercept)                       0.027867 0.16693
##           ConcValComboabstract-positive     0.001476 0.03841  -0.51
##           ConcValComboconcrete-negative     0.001567 0.03958   0.13 -0.10
##           ConcValComboconcrete-positive     0.003105 0.05573  -0.37  0.63  0.69
##  Residual                                   0.100394 0.31685
## Number of obs: 8596, groups:  Word, 40; ID.true, 38
##
## Fixed effects:
```

```
##                                Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)                     6.67346    0.02986 46.32498 223.500   <2e-16 ***
## ConcValComboabstract-positive -0.01991    0.01884 38.36429  -1.057   0.2972
## ConcValComboconcrete-negative  0.03154    0.01902 39.89565   1.658   0.1052
## ConcValComboconcrete-positive -0.03492    0.02005 42.92816  -1.742   0.0887 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                 (Intr) CncVlCmbb- CncVlCmbcncrt-n
## CncVlCmbbs-     -0.435
## CncVlCmbcncrt-n -0.238  0.429
## CncVlCmbcncrt-p -0.417  0.514      0.521
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.0198304 (tol = 0.002, component 1)
```

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

- Nope

```
str(df_factors)
```

```
## 'data.frame':    8596 obs. of  20 variables:
##  $ X              : Factor w/ 8596 levels "2","3","4","5",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ ID.true        : Factor w/ 38 levels "5731e4005b8b06000f7e36b3",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Word           : Factor w/ 40 levels "anxiety","beach",..: 3 39 14 12 7 13 19 25 9 6 ...
##  $ 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",..: 4 4 1 3 1 2 4 1 4 1 ...
##  $ 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/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Response       : Factor w/ 4 levels "abstract","concrete",..: 4 4 3 3 3 4 4 3 4 3 ...
##  $ Accuracy       : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ EventTime      : Factor w/ 8595 levels "1737993435128",..: 460 461 462 463 464 465 466 467 468 469
##  $ Value          : Factor w/ 4 levels "abstract","concrete",..: 4 4 3 3 3 4 4 3 4 3 ...
##  $ RT             : Factor w/ 1324 levels "1267.66666666667",..: 165 386 71 107 22 306 50 442 160 119
##  $ ReactionTime   : int  839 551 512 499 550 630 604 749 774 622 ...
##  $ 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  6.73 6.31 6.24 6.21 6.31 ...
##  $ LogRT          : Factor w/ 1324 levels "7.14493321926015",..: 165 386 71 107 22 306 50 442 160 119
##  $ TrialNumber    : Factor w/ 448 levels "1","2","3","4",..: 2 3 4 5 6 7 8 9 10 11 ...
```

```
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: 2248.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.0697 -0.6449 -0.1719  0.4843  4.5761
##
## Random effects:
##  Groups   Name          Variance Std.Dev. Corr
##  Word     (Intercept)   0.004098 0.06402
##           cConcValCombo 0.004844 0.06960  0.95
##  ID.true  (Intercept)   0.038846 0.19709
##           cConcValCombo 0.001217 0.03488  0.05
##  Residual               0.094494 0.30740
## Number of obs: 4133, groups:  Word, 40; ID.true, 38
##
## Fixed effects:
##               Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)    6.77374    0.03479 47.35157 194.708   <2e-16 ***
## cConcValCombo  0.03704    0.01479 37.41334   2.505   0.0167 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cConcValCmb 0.256
```

```
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: 2241.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.0929 -0.6439 -0.1747  0.4794  4.5607
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
```

```
##  Word     (Intercept) 0.009159 0.09570
##  ID.true  (Intercept) 0.038833 0.19706
##           cSemantic   0.008903 0.09436  0.02
##  Residual             0.093824 0.30631
## Number of obs: 4133, groups:  Word, 40; ID.true, 38
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.76213    0.03572 51.88604 189.290   <2e-16 ***
## cSemantic    0.05201    0.03531 48.43900   1.473    0.147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## cSemantic -0.018
```

## In the Valence task , is there a difference between positive and negative on ReactionTime?

- Nope.

```
val <- df_factors %>%
  filter(Task == "Valence") %>%
  mutate(
         Semantic = ifelse(grepl("concrete", ConcValCombo), "concrete",
                    ifelse(grepl("abstract", ConcValCombo), "abstract", NA)),
         Valence = ifelse(grepl("positive", ConcValCombo), "positive",
                    ifelse(grepl("negative", ConcValCombo), "negative", NA)),
         cConcValCombo = as.numeric(ConcValCombo) - mean(as.numeric(ConcValCombo)),
         cValence = as.numeric(factor(Valence)) - mean(as.numeric(factor(Valence)))
         )

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

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

```
summary(m)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cConcValCombo + (1 + cConcValCombo | ID.true) +
##     (1 + cConcValCombo | Word)
##    Data: val
##
## REML criterion at convergence: 1519.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -6.0649 -0.6137 -0.1668  0.4269  5.2469
##
## Random effects:
##  Groups   Name          Variance  Std.Dev. Corr
##  Word     (Intercept)   3.891e-03 0.062376
```

```
##           cConcValCombo 3.584e-05 0.005986 1.00
##  ID.true  (Intercept)   2.317e-02 0.152229
##           cConcValCombo 1.126e-04 0.010611 -0.34
##  Residual               7.823e-02 0.279699
## Number of obs: 4463, groups:  Word, 40; ID.true, 38
##
## Fixed effects:
##                Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)    6.591965   0.026944 47.761937 244.655   <2e-16 ***
## cConcValCombo -0.013787   0.009725 38.831847  -1.418    0.164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## cConcValCmb 0.014
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
m = lmer(LogReactionTime ~ cValence + (1+cValence|ID.true) + (1|Word), data=val)
summary(m)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogReactionTime ~ cValence + (1 + cValence | ID.true) + (1 |
##     Word)
##    Data: val
##
## REML criterion at convergence: 1473.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.8193 -0.6083 -0.1559  0.4083  5.3118
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Word     (Intercept) 0.003291 0.05737
##  ID.true  (Intercept) 0.023454 0.15315
##           cValence    0.005980 0.07733  -0.46
##  Residual             0.076980 0.27745
## Number of obs: 4463, groups:  Word, 40; ID.true, 38
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  6.59261    0.02678 46.00933 246.202   <2e-16 ***
## cValence    -0.05837    0.02359 53.63558  -2.475   0.0165 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## cValence -0.228
```