Animacy Nouns: Reaction Time Graphs

morgan moyer

2025-03-27

Looking at overall Log ReactionTime for the data

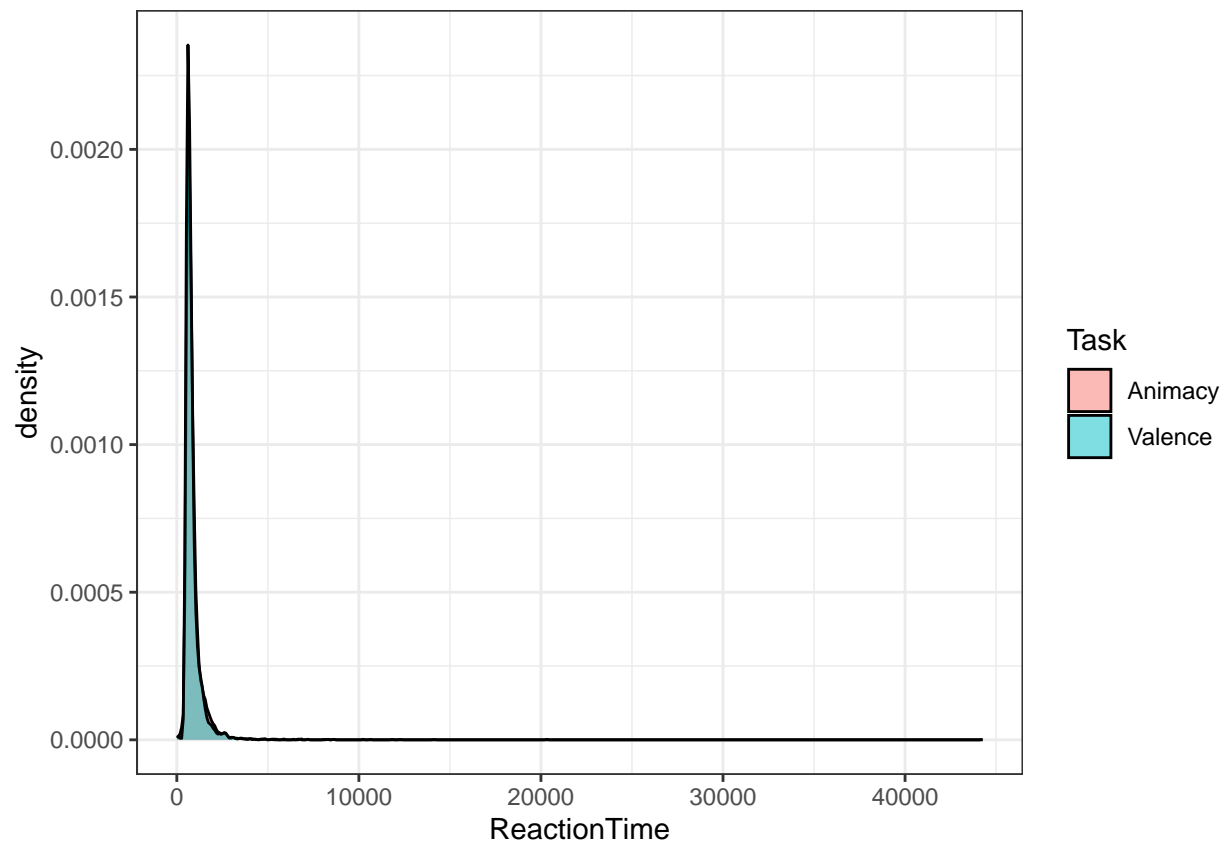Before removing outliers

Summary Stats

```
agr <- d %>%
  group_by(Task) %>%
  summarize(MeanRT = mean(ReactionTime),
            SD = sd(ReactionTime),
            MeanLogRT = mean(LogReactionTime))
print(agr)
```
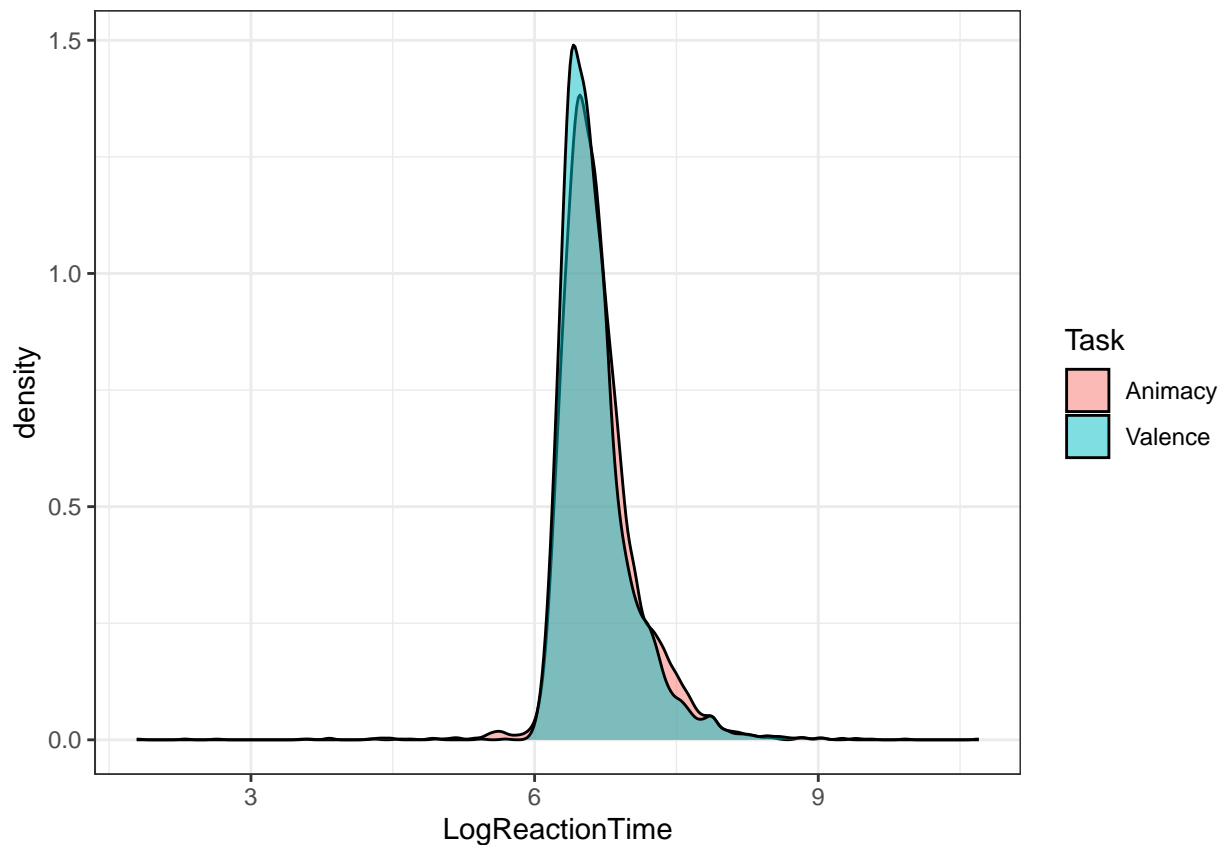
```
## # A tibble: 2 x 4
##   Task      MeanRT    SD MeanLogRT
##   <chr>      <dbl> <dbl>     <dbl>
## 1 Animacy    877.  575.      6.67
## 2 Valence    857.  883.      6.64
```

```
ggplot(d, aes(ReactionTime, fill=Task)) +
  geom_density(alpha = .5)
```

Long tail justifies outlier removal?

```
ggplot(d, aes(LogReactionTime, fill=Task)) +
  geom_density(alpha = .5)
```

```r
summary(d$LogReactionTime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.792   6.404   6.582   6.658   6.819  10.698
```

First Remove participants who aren't super , aggregating over Task

```r
length(unique(d$ID.true))
```

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

```r
inacc.parts <- d %>%
  group_by(ID.true,Task) %>%
  summarise(MeanAccuracy = mean(Accuracy)) %>%
  filter(MeanAccuracy < .75)
```

```
## `summarise()` has grouped output by 'ID.true'. You can override using the
## `.groups` argument.
# How many participants have Accuracy < .75?
length(unique(inacc.parts$ID.true))
```

```
## [1] 4
```

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

# Sanity check
```

3

```
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
```
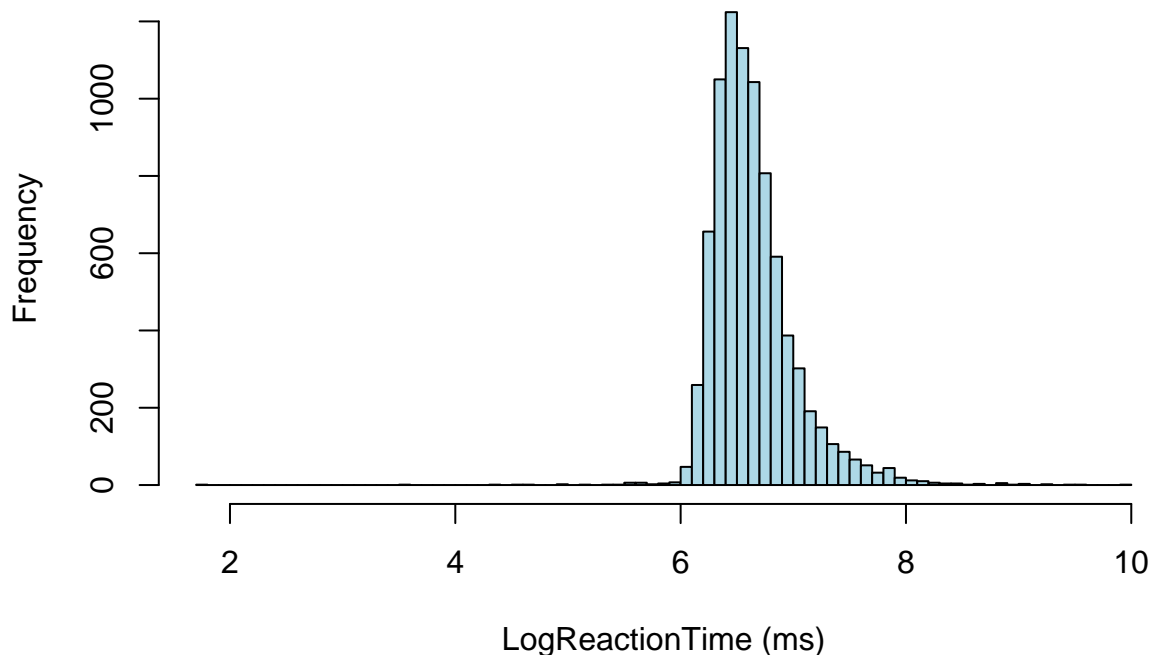
```
## [1] 96.41204
```
```
# Remove subjects with ReactionTime higher than 3x IQR
summary(d.inaccurate.removed$LogReactionTime)
```

```
##    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")
```

Histogram with Normal Curve


```
quantile(d.inaccurate.removed$LogReactionTime, na.rm = TRUE)
```

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

```
IQR(d.inaccurate.removed$LogReactionTime, na.rm = TRUE)*3 # 0.7526289
```

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

```
cutoff.high <- quantile(d.inaccurate.removed$LogReactionTime, na.rm = TRUE)[4] + IQR(d.inaccurate.remove
cutoff.low <- quantile(d.inaccurate.removed$LogReactionTime, na.rm = TRUE)[2] - IQR(d.inaccurate.remove

# 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")
```

Histogram with Normal Curve
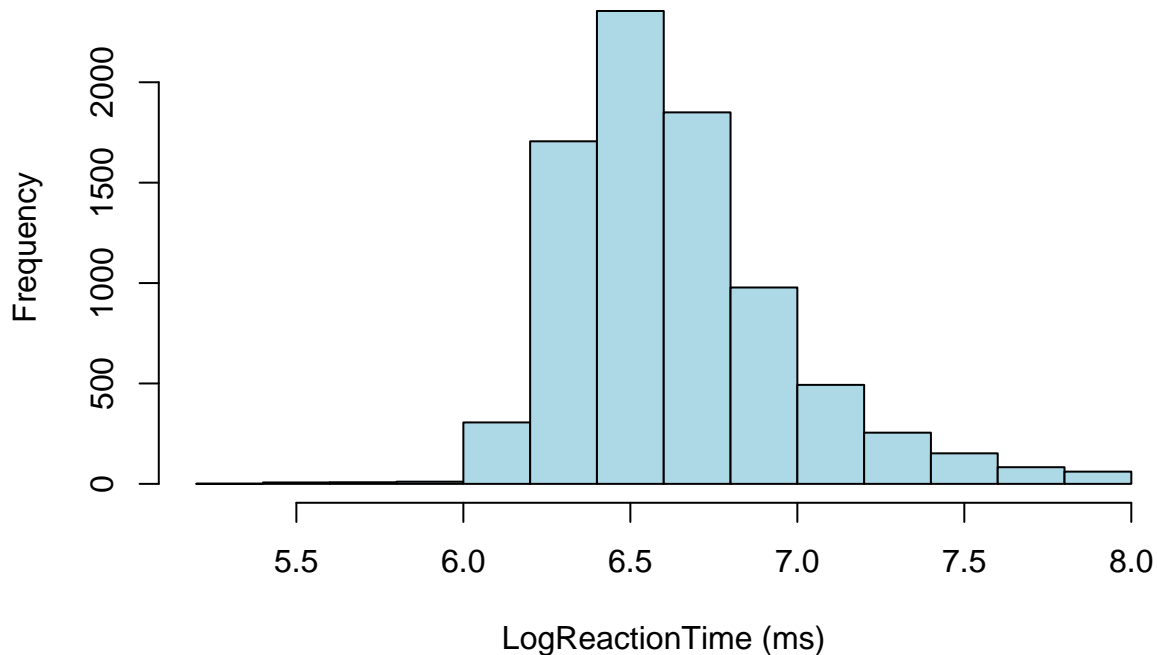
Summary Stats

```
agr <- df.outliers.removed %>%
  group_by(Task) %>%
  summarize(MeanRT = mean(ReactionTime),
            SD = sd(ReactionTime),
            MeanLogRT = mean(LogReactionTime))
print(agr)
```
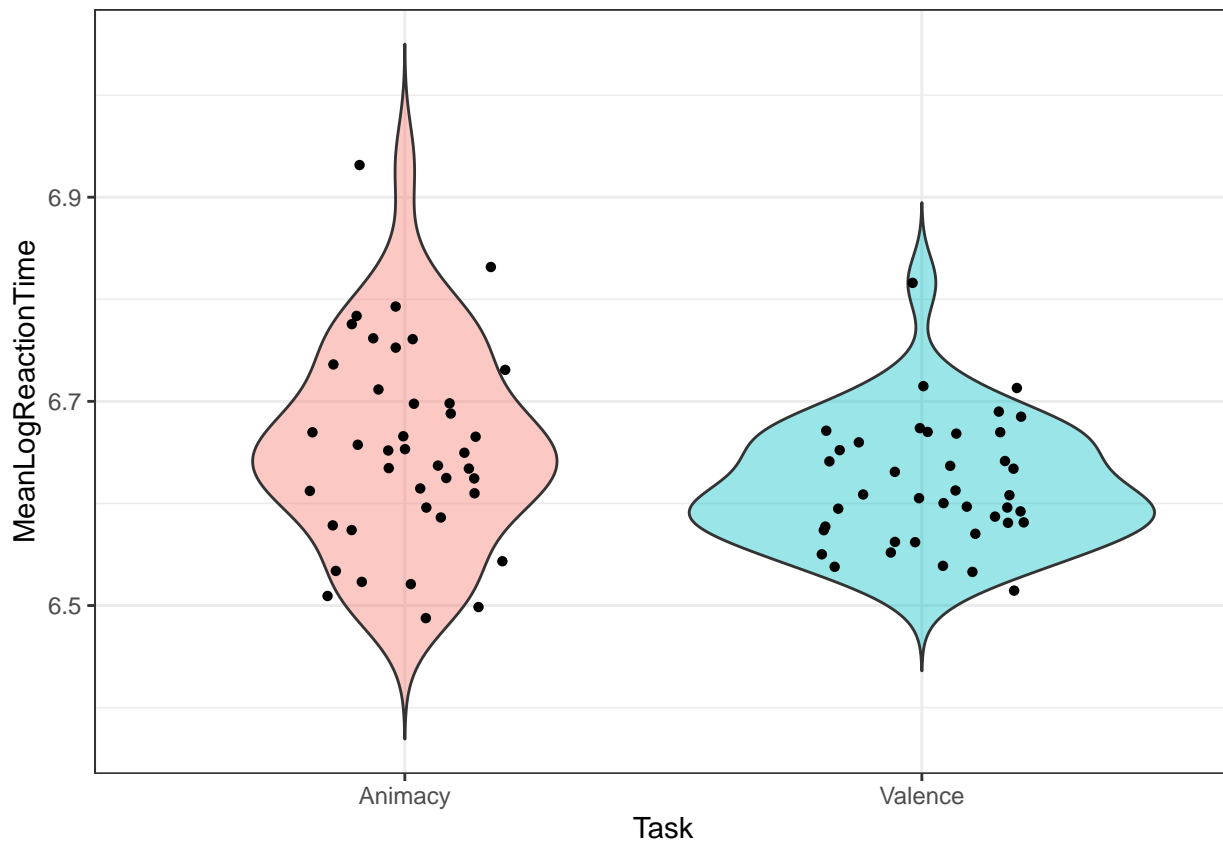
```
## # A tibble: 2 x 4
##   Task      MeanRT    SD MeanLogRT
##   <chr>      <dbl> <dbl>     <dbl>
## 1 Animacy     827.  353.      6.65
## 2 Valence     793.  323.      6.62
```

LogReactionTime by Task

```
agr <- df.outliers.removed %>%
    group_by(Task,Word) %>%
    summarize(MeanLogReactionTime = mean(LogReactionTime),
              CILow = ci.low(LogReactionTime),
              CIHigh = ci.high(LogReactionTime)) %>%
    mutate(YMin = MeanLogReactionTime - CILow,
           YMax = MeanLogReactionTime + CIHigh)
```

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

```
ggplot(agr, aes(x=Task, y=MeanLogReactionTime,fill=Task)) +
    geom_violin(trim=FALSE,alpha=.4) +
    geom_jitter(shape=16, position=position_jitter(0.2)) +
  guides(fill = "none")
```
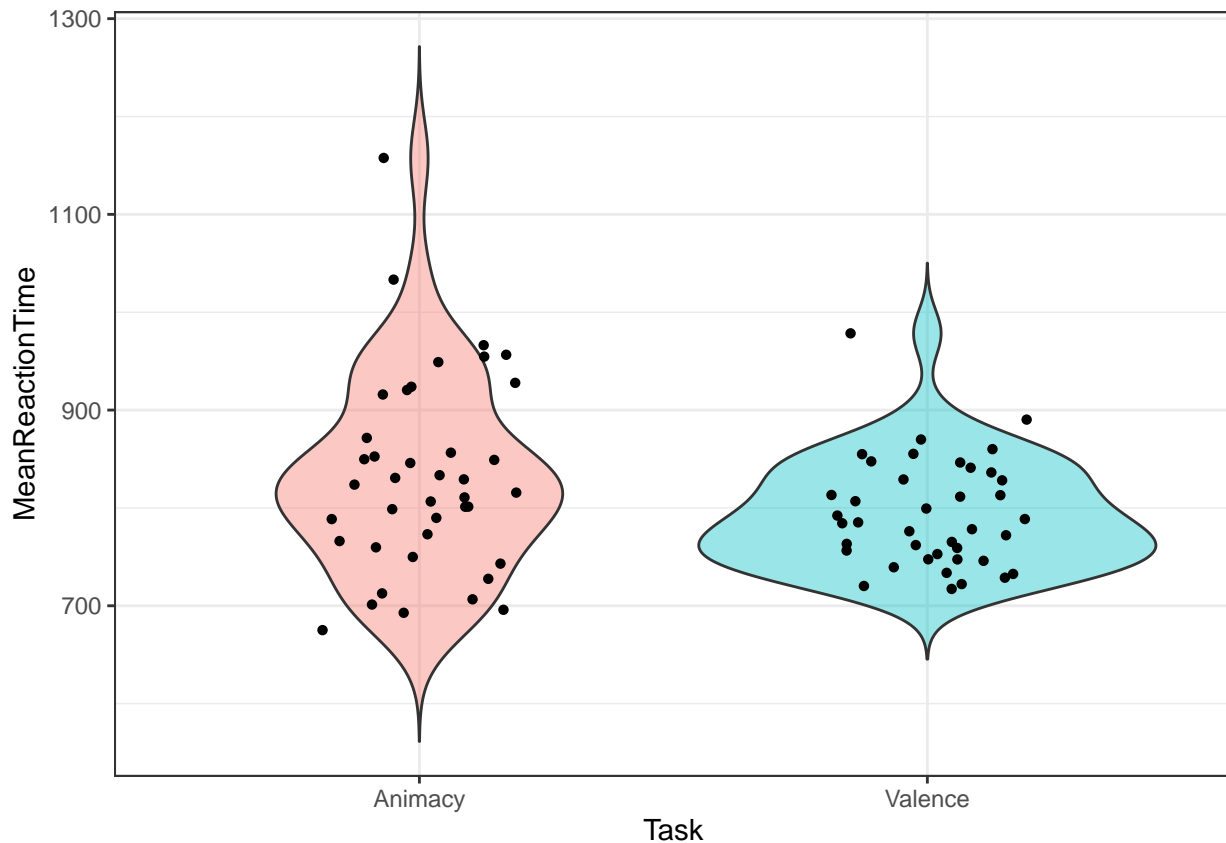

ReactionTime by Task

```
agr <- df.outliers.removed %>%
    group_by(Task,Word) %>%
    summarize(MeanReactionTime = mean(ReactionTime),
              CILow = ci.low(ReactionTime),
              CIHigh = ci.high(ReactionTime)) %>%
    mutate(YMin = MeanReactionTime - CILow,
```

```
            YMax = MeanReactionTime + CIHigh)
```

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

```
ggplot(agr, aes(x=Task, y=MeanReactionTime,fill=Task)) +
    geom_violin(trim=FALSE,alpha=.4) +
    geom_jitter(shape=16, position=position_jitter(0.2)) +
  guides(fill = "none")
```


```
ggsave("../graphs/exp3.pdf",width = 3, height = 2)
```

## ReactionTime by BlockOrder and Task

```
agr <- df.outliers.removed %>%
    group_by(BlockOrder,Task,Word) %>%
    summarize(MeanReactionTime = mean(ReactionTime),
              CILow = ci.low(ReactionTime),
              CIHigh = ci.high(ReactionTime)) %>%
    mutate(YMin = MeanReactionTime - CILow,
           YMax = MeanReactionTime + CIHigh)
```

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

```
ggplot(agr, aes(x=Task, y=MeanReactionTime,fill=BlockOrder)) +
    geom_violin(trim=FALSE,alpha=.4) +
```

```
    geom_jitter(shape=16, position=position_jitter(0.2))
```
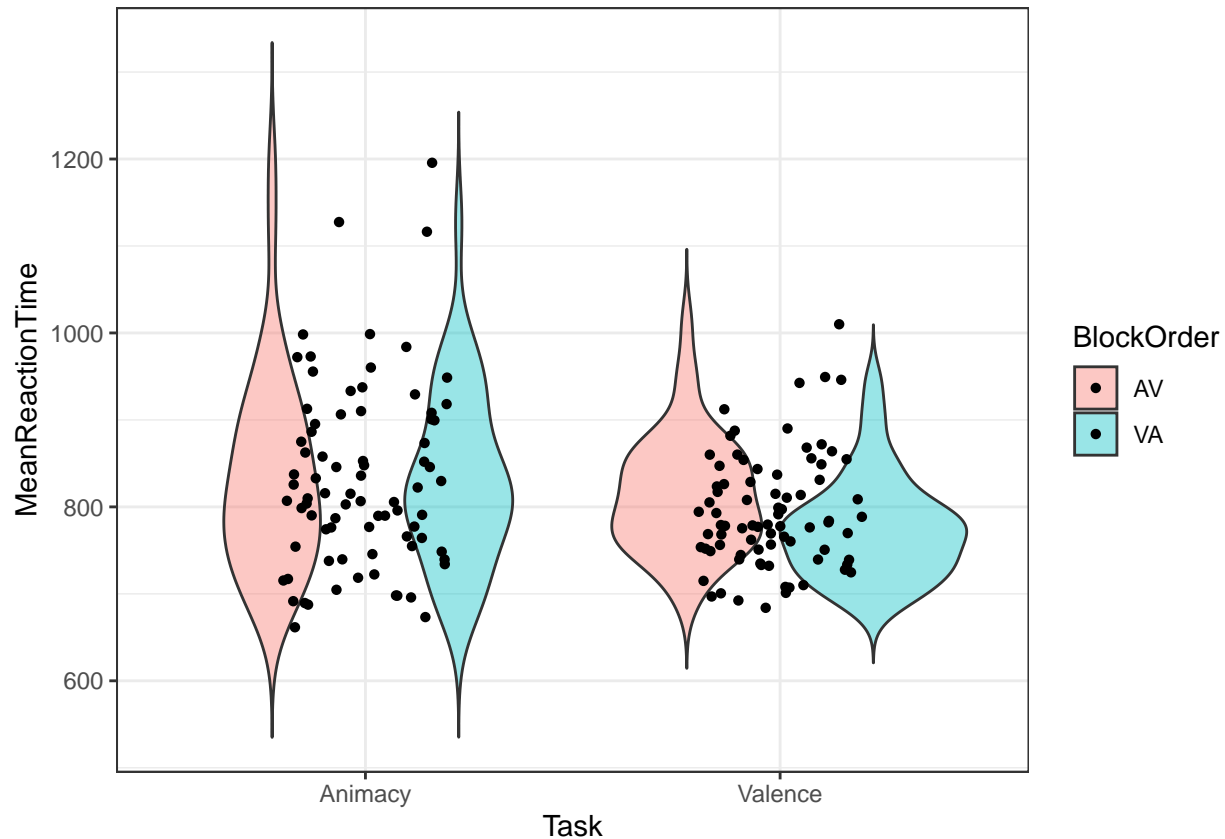

By Item

```
agr <- df.outliers.removed %>%
    group_by(Task,Word) %>%
    summarize(MeanReactionTime = mean(ReactionTime), CILow = ci.low(ReactionTime), CIHigh = ci.high(Rea
    mutate(YMin = MeanReactionTime - CILow, YMax = MeanReactionTime + CIHigh)
```

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

```
dodge = position_dodge(.9)
ggplot(data=agr, aes(x=Task,y=MeanReactionTime,fill=Task)) +
  geom_bar(position=dodge,stat="identity") +
  facet_wrap(~Word) +
  geom_errorbar(aes(ymin=YMin,ymax=YMax),width=.25,position=position_dodge(0.9)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
# guides(fill = "none")
```

By Word Feature category and Task

Mean Raw ReactionTime and Effects of Word Valence/Animacy

```
agr <- df.outliers.removed %>%
    group_by(Task,Valence) %>%
    reframe(MeanReactionTime = mean(ReactionTime), CILow = ci.low(ReactionTime), CIHigh = ci.high(React
    mutate(YMin = MeanReactionTime - CILow, YMax = MeanReactionTime + CIHigh)

dodge = position_dodge(.9)
ggplot(data=agr, aes(x=Task,y=MeanReactionTime,fill=Valence)) +
  geom_bar(position=dodge,stat="identity") +
  # facet_wrap(~Task) +
  geom_errorbar(aes(ymin=YMin,ymax=YMax),width=.25,position=position_dodge(0.9))
```
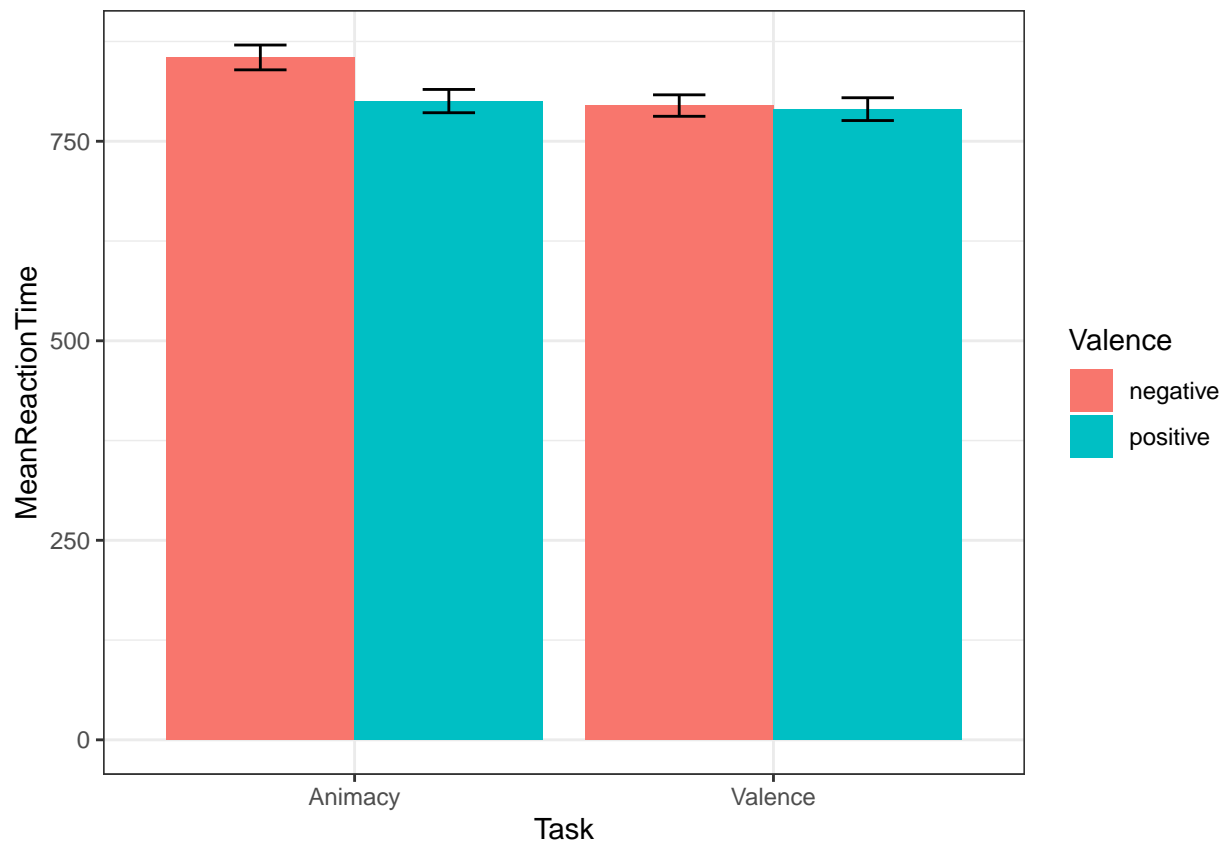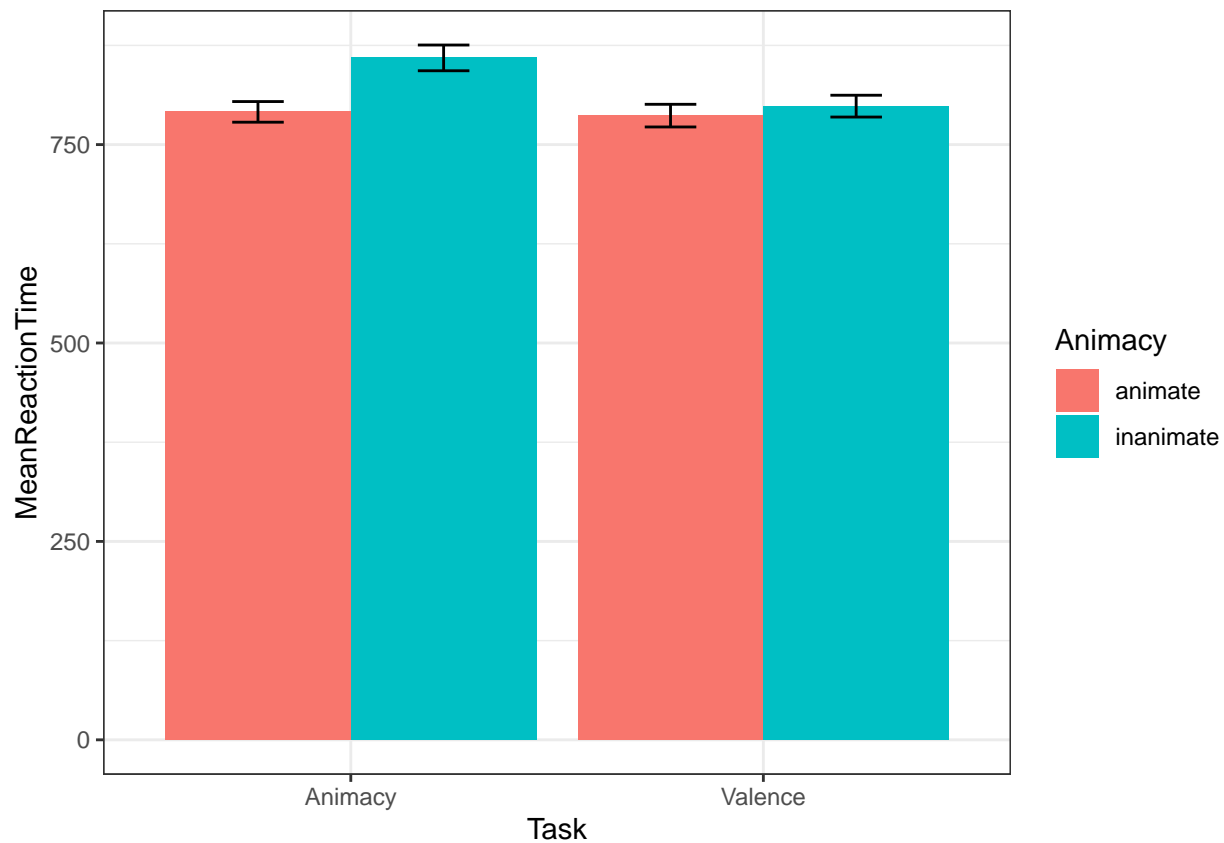
```
# theme(axis.text.x = element_text(angle = 45, hjust = 1))
# guides(fill = "none")
```

```
agr <- df.outliers.removed %>%
    group_by(Task,Animacy) %>%
    reframe(MeanReactionTime = mean(ReactionTime), CILow = ci.low(ReactionTime), CIHigh = ci.high(React
    mutate(YMin = MeanReactionTime - CILow, YMax = MeanReactionTime + CIHigh)

dodge = position_dodge(.9)
ggplot(data=agr, aes(x=Task,y=MeanReactionTime,fill=Animacy)) +
  geom_bar(position=dodge,stat="identity") +
  # facet_wrap(~Task) +
  geom_errorbar(aes(ymin=YMin,ymax=YMax),width=.25,position=position_dodge(0.9))
```

```
# theme(axis.text.x = element_text(angle = 45, hjust = 1))
# guides(fill = "none")
```

It seems that positive animate nouns are exceptionally fast compared to all the others.

```
agr <- df.outliers.removed %>%
    group_by(Task,Valence,Animacy) %>%
    reframe(MeanReactionTime = mean(ReactionTime), CILow = ci.low(ReactionTime), CIHigh = ci.high(React
    mutate(YMin = MeanReactionTime - CILow, YMax = MeanReactionTime + CIHigh)

dodge = position_dodge(.9)
ggplot(data=agr, aes(x=Animacy,y=MeanReactionTime,fill=Valence)) +
  geom_bar(position=dodge,stat="identity") +
  facet_wrap(~Task) +
  geom_errorbar(aes(ymin=YMin,ymax=YMax),width=.25,position=position_dodge(0.9))
```
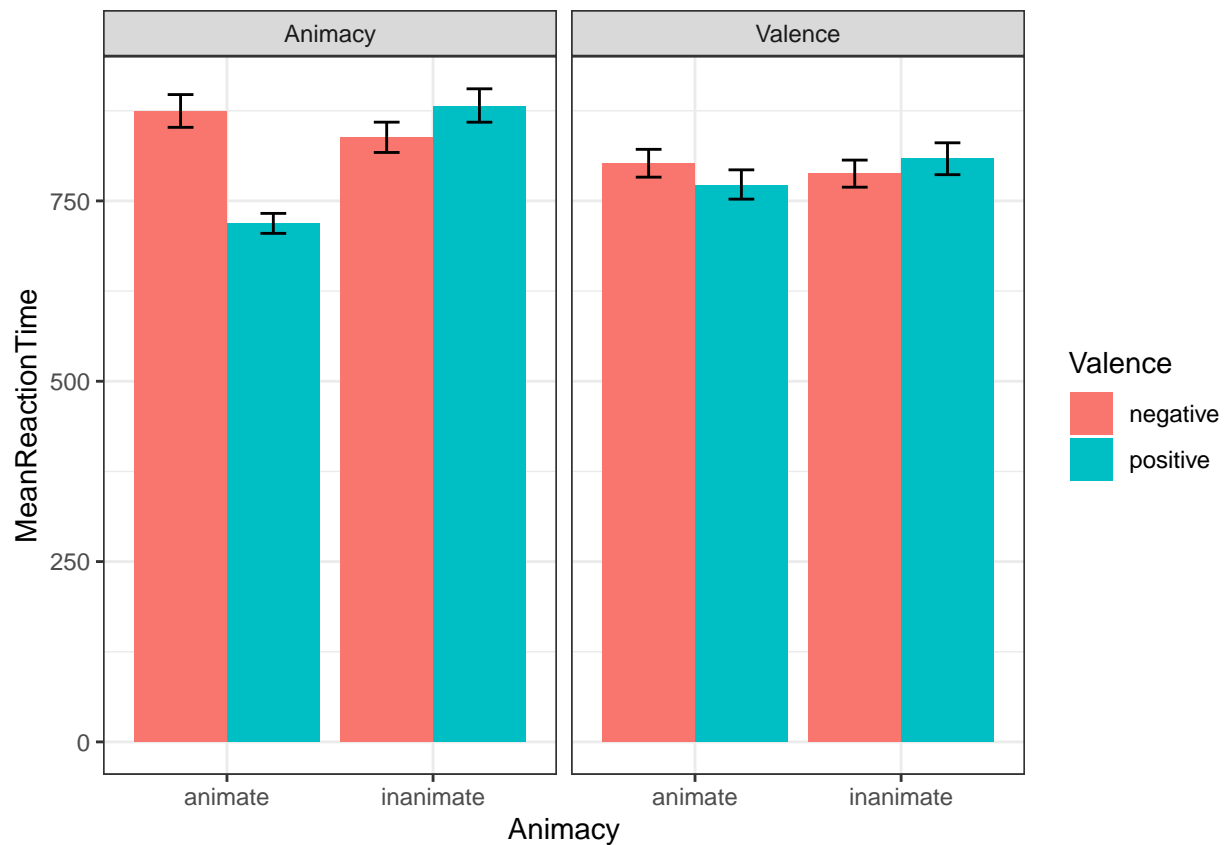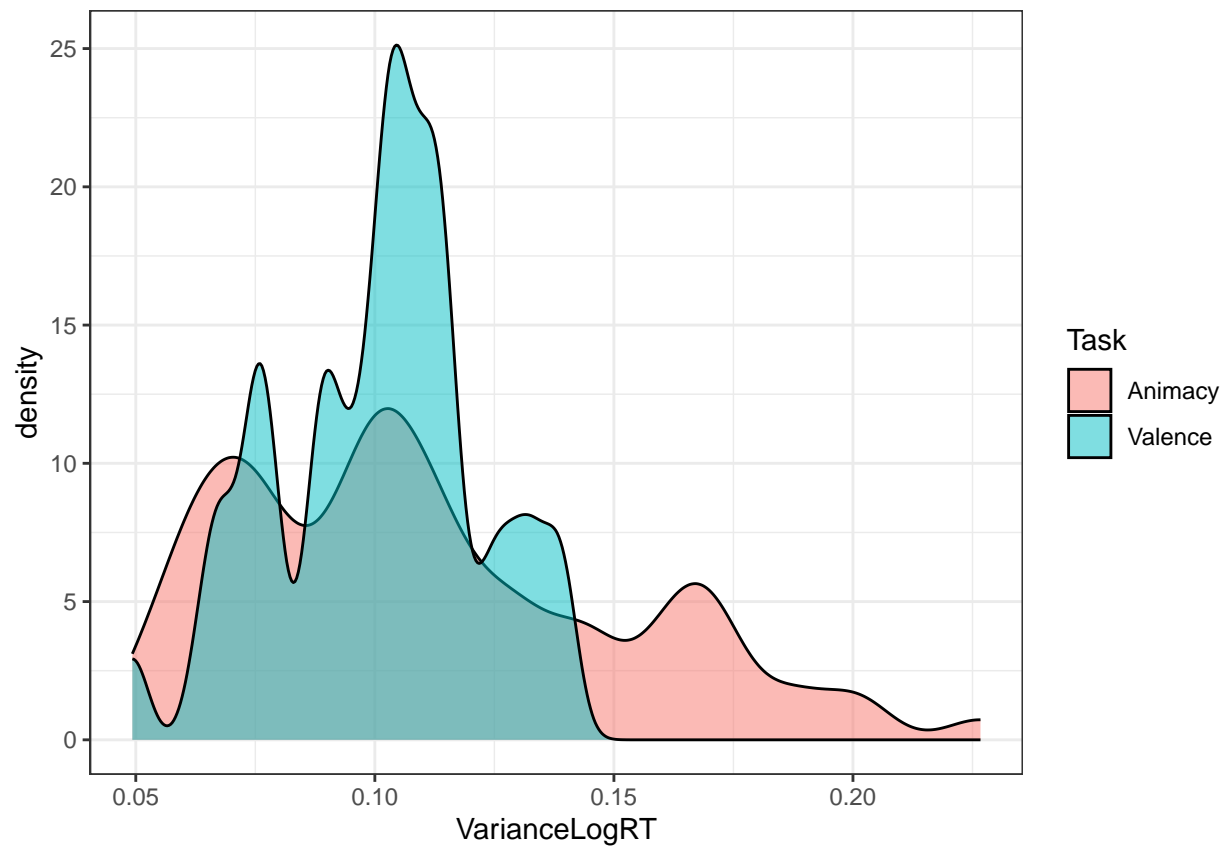
```
# theme(axis.text.x = element_text(angle = 45, hjust = 1))
# guides(fill = "none")
```
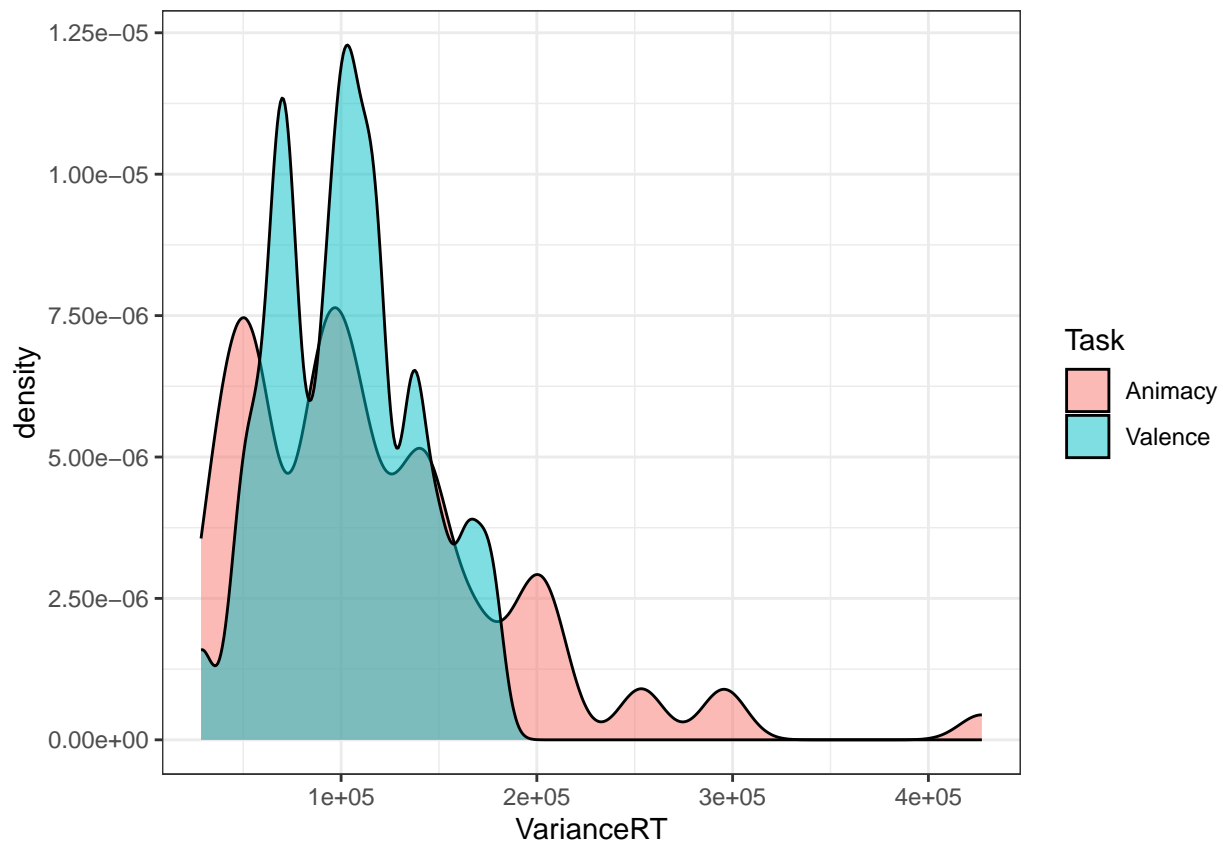
Variance

```
agr <- df.outliers.removed %>%
  group_by(Task,Word) %>%
  mutate(VarianceLogRT = var(LogReactionTime))

ggplot(agr, aes(VarianceLogRT, fill=Task)) +
  geom_density(alpha = .5)
```

```
agr <- df.outliers.removed %>%
  group_by(Task,Word) %>%
  mutate(VarianceRT = var(ReactionTime))

ggplot(agr, aes(VarianceRT, fill=Task)) +
  geom_density(alpha = .5)
```
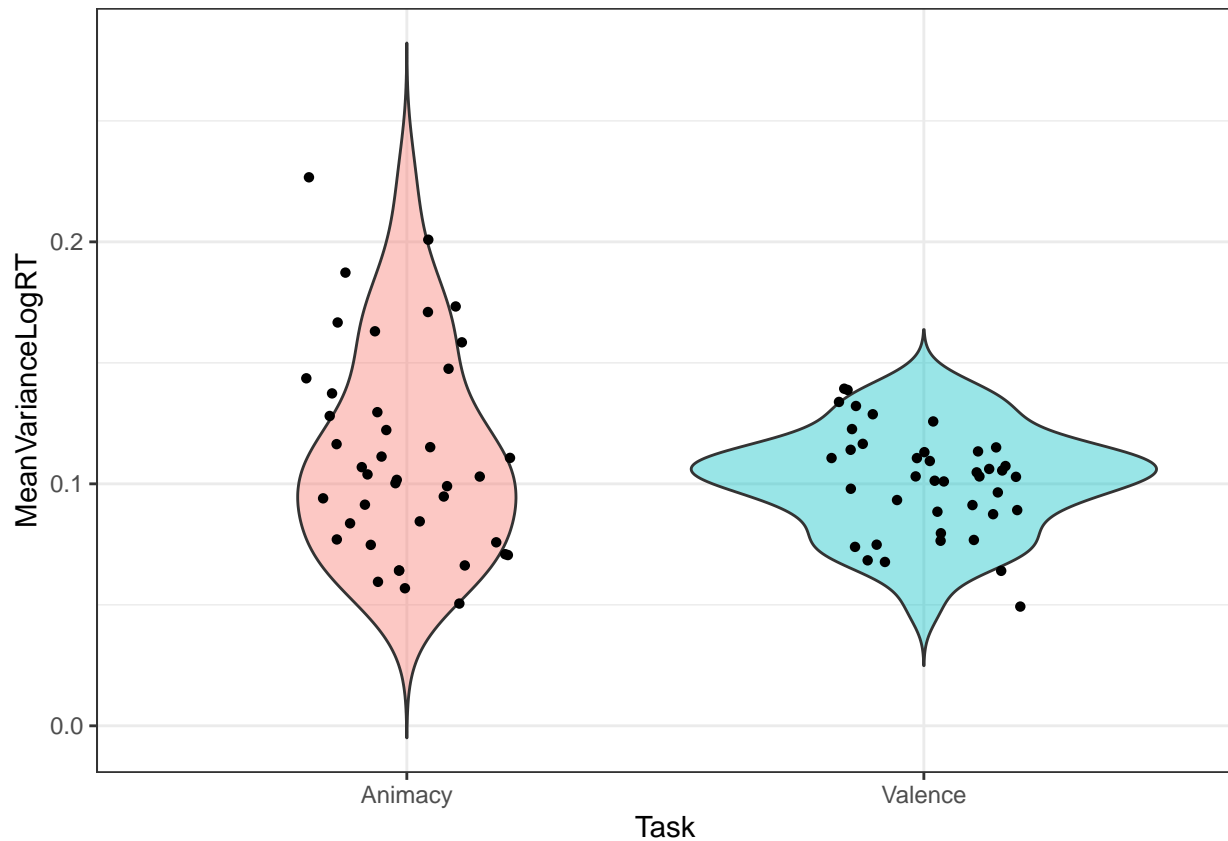
Variance by Task

```
agr <- df.outliers.removed %>%
    group_by(Task,Word) %>%
    mutate(VarianceLogRT = var(LogReactionTime)) %>%
    summarize(MeanVarianceLogRT = mean(VarianceLogRT),
              CILow = ci.low(VarianceLogRT),
              CIHigh = ci.high(VarianceLogRT)) %>%
    mutate(YMin = MeanVarianceLogRT - CILow,
           YMax = MeanVarianceLogRT + CIHigh)
```

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

```
ggplot(agr, aes(x=Task, y=MeanVarianceLogRT,fill=Task)) +
    geom_violin(trim=FALSE,alpha=.4) +
    geom_jitter(shape=16, position=position_jitter(0.2)) +
  guides(fill = "none")
```
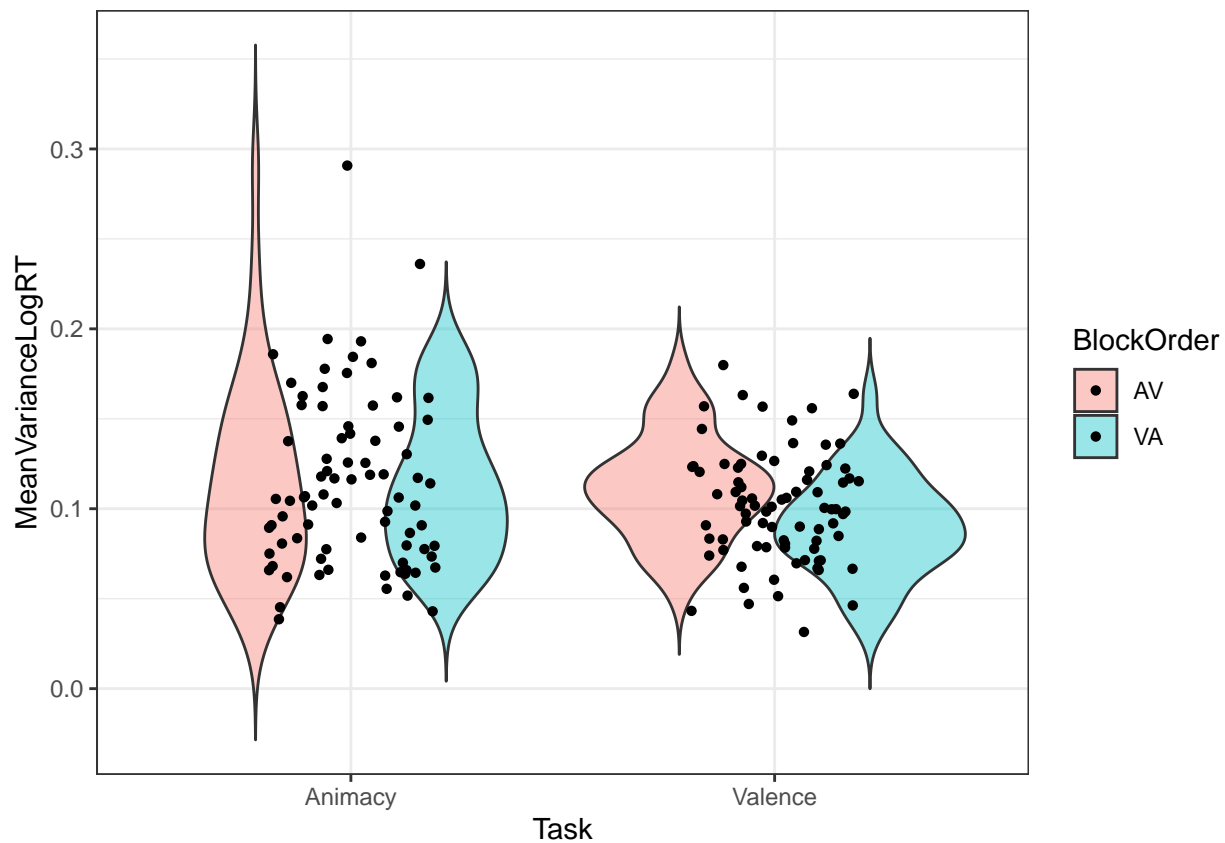
Variance by BlockOrder and Task

```r
agr <- df.outliers.removed %>%
    group_by(BlockOrder,Task,Word) %>%
    mutate(VarianceLogRT = var(LogReactionTime)) %>%
    summarize(MeanVarianceLogRT = mean(VarianceLogRT),
              CILow = ci.low(VarianceLogRT),
              CIHigh = ci.high(VarianceLogRT)) %>%
    mutate(YMin = MeanVarianceLogRT - CILow,
           YMax = MeanVarianceLogRT + CIHigh)
```

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

```r
ggplot(agr, aes(x=Task, y=MeanVarianceLogRT,fill=BlockOrder)) +
    geom_violin(trim=FALSE,alpha=.4) +
    geom_jitter(shape=16, position=position_jitter(0.2))
```

By Item

```r
agr <- df.outliers.removed %>%
    group_by(Task,Word) %>%
    mutate(VarianceLogRT = var(LogReactionTime)) %>%
    summarize(MeanVarianceLogRT = mean(VarianceLogRT),
              CILow = ci.low(VarianceLogRT),
              CIHigh = ci.high(VarianceLogRT)) %>%
    mutate(YMin = MeanVarianceLogRT - CILow,
           YMax = MeanVarianceLogRT + CIHigh)
```

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

```r
dodge = position_dodge(.9)
ggplot(data=agr, aes(x=Task,y=MeanVarianceLogRT,fill=Task)) +
  geom_bar(position=dodge,stat="identity") +
  facet_wrap(~Word) +
  geom_errorbar(aes(ymin=YMin,ymax=YMax),width=.25,position=position_dodge(0.9))
```

```
# theme(axis.text.x = element_text(angle = 45, hjust = 1))
# guides(fill = "none")
```