

# Adjs Soc-Phys: Reaction Time Graphs

morgan moyer

2025-05-02

## Looking at overall Log ReactionTime for the data

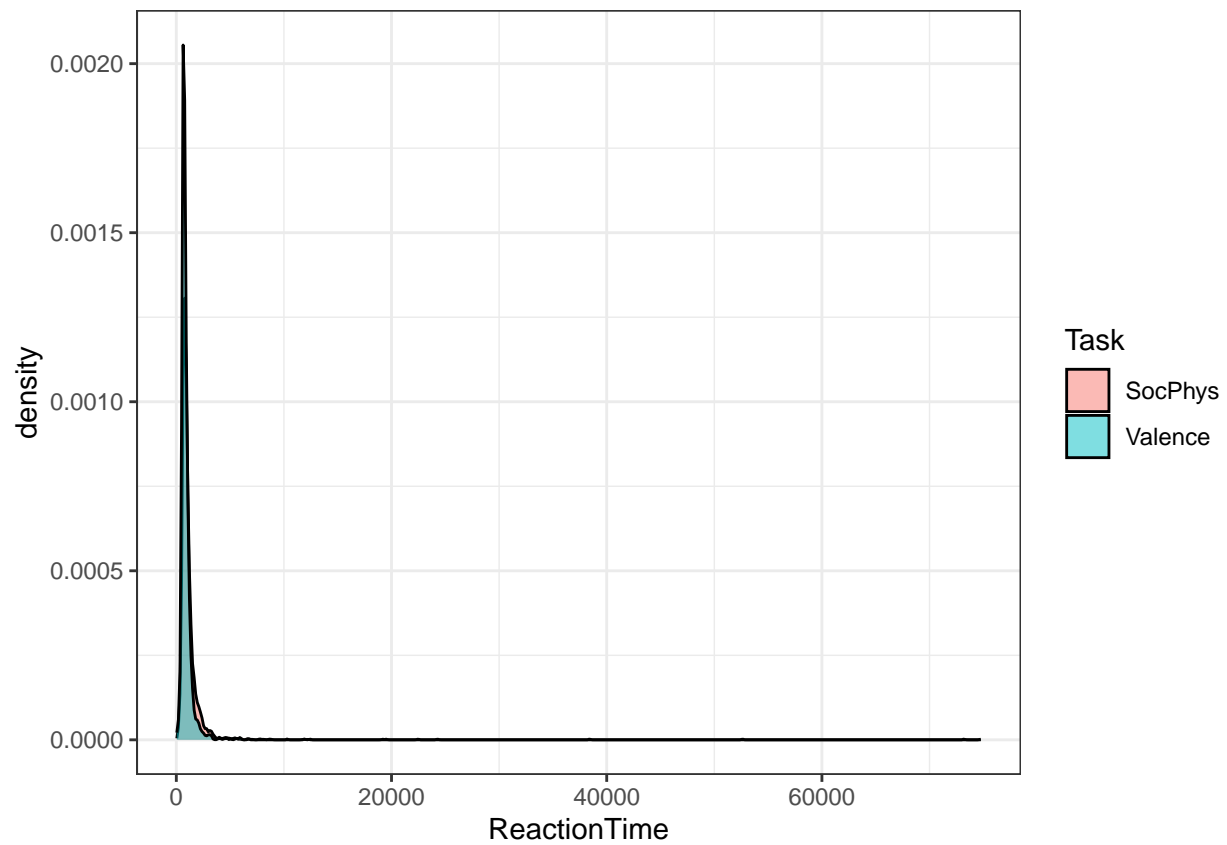
Before removing outliers

### Summary Stats

```
agr <- d %>%  
  group_by(Task) %>%  
  summarize(MeanRT = mean(ReactionTime),  
            SD = sd(ReactionTime),  
            MeanLogRT = mean(LogReactionTime))  
print(agr)
```

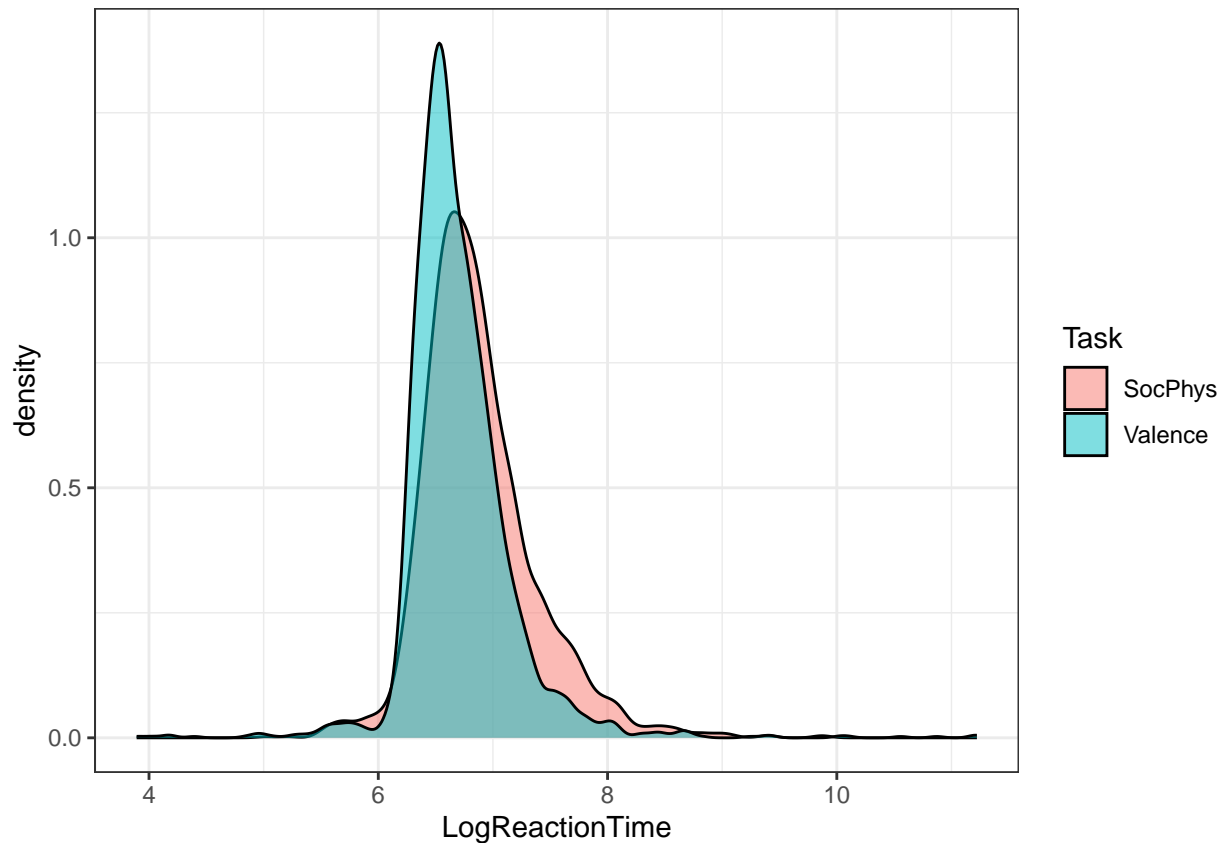
```
## # A tibble: 2 x 4  
##   Task    MeanRT    SD MeanLogRT  
##   <chr>    <dbl> <dbl>    <dbl>  
## 1 SocPhys  1134.  982.    6.88  
## 2 Valence  1017. 2641.    6.70
```

```
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)
```



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

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  3.892   6.497   6.713   6.791   6.996   11.222
```

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

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

```
## [1] 20
```

```
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] 18
```

## 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] 91.96759
```

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

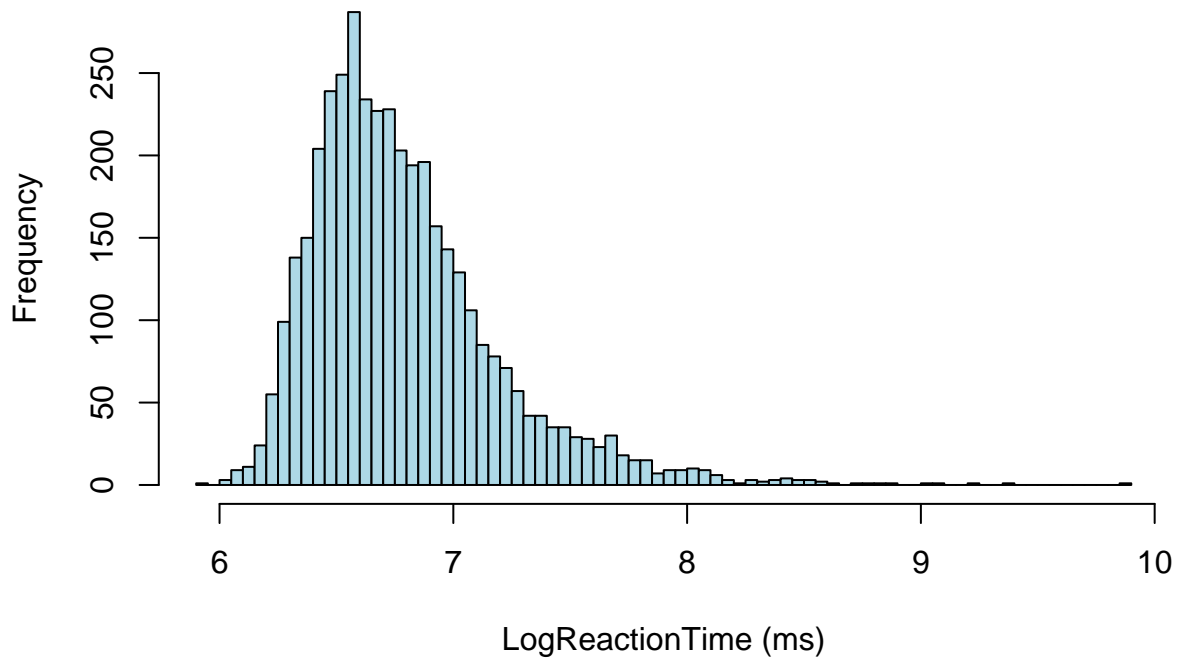
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.935   6.513   6.713   6.790   6.974   9.879
```

```
# 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] 5.934894 9.878836
```

```
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%
```

```
## 5.934894 6.513230 6.712956 6.973543 9.878836
```

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

```
## [1] 1.380939
```

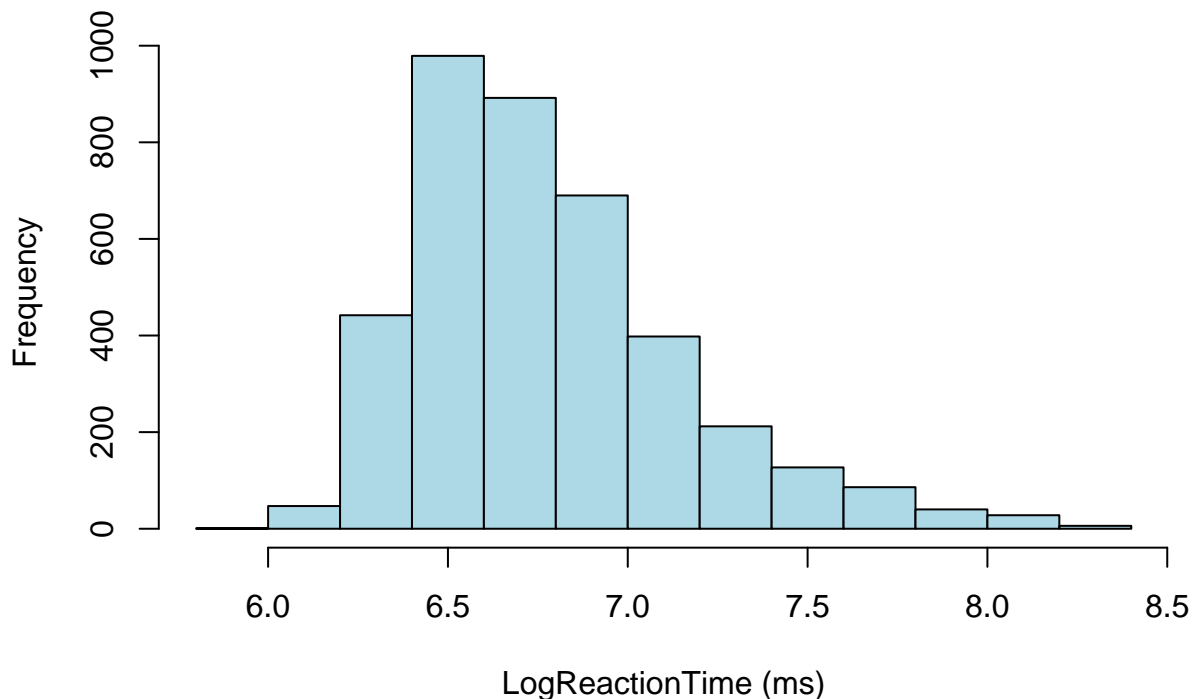
```
cutoff.high <- quantile(d.inaccurate.removed$LogReactionTime, na.rm = TRUE)[4] + IQR(d.inaccurate.removed$LogReactionTime, na.rm = TRUE)
cutoff.low <- quantile(d.inaccurate.removed$LogReactionTime, na.rm = TRUE)[2] - IQR(d.inaccurate.removed$LogReactionTime, na.rm = TRUE)
```

```
# 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

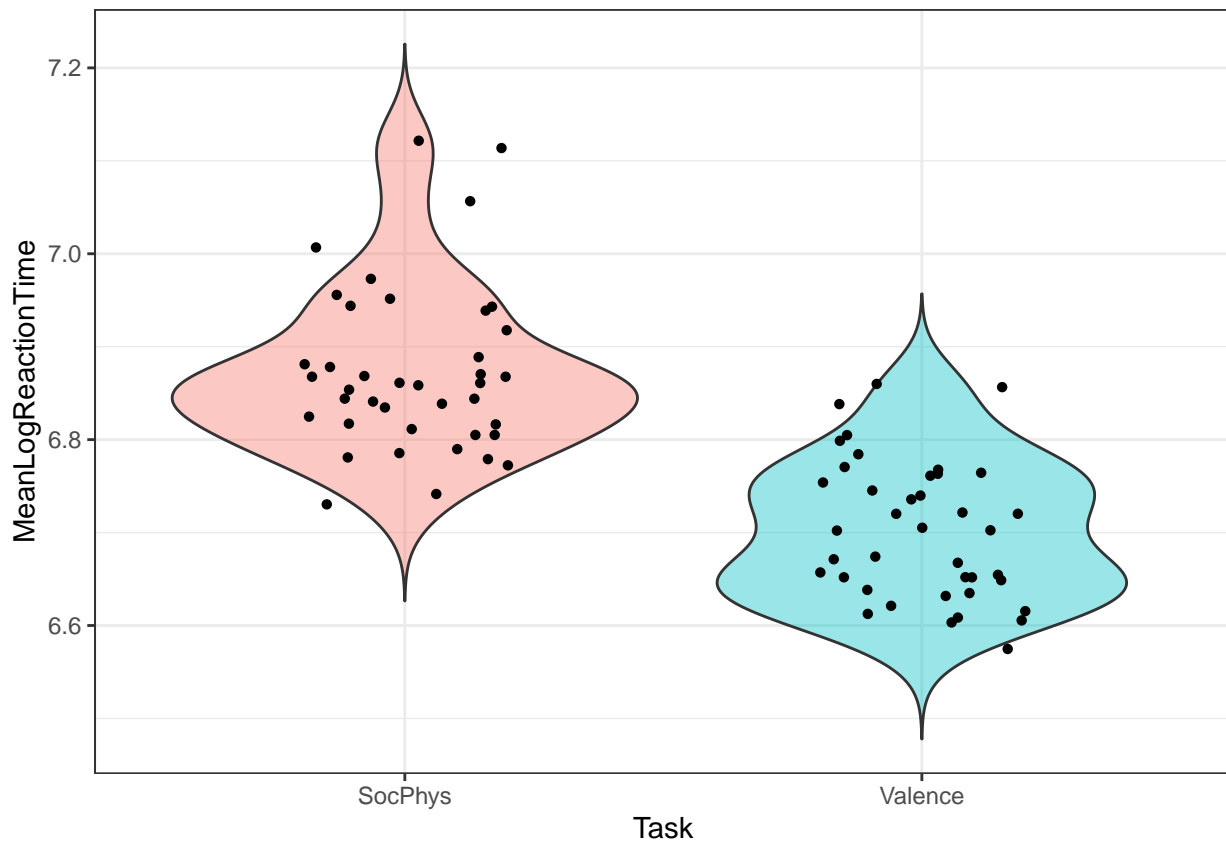
```
agr <- df.outliers.removed %>%
  group_by(Task) %>%
  summarize(MeanReactionTime = mean(ReactionTime),
            SD = sd(ReactionTime),
            MeanLogReactionTime = mean(LogReactionTime))
print(agr)
```

```
## # A tibble: 2 x 4
```

```
##   Task      MeanReactionTime    SD MeanLogReactionTime
##   <chr>          <dbl> <dbl>          <dbl>
## 1 SocPhys      1047.   522.           6.86
## 2 Valence       863.   358.           6.70
```

## 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")
```



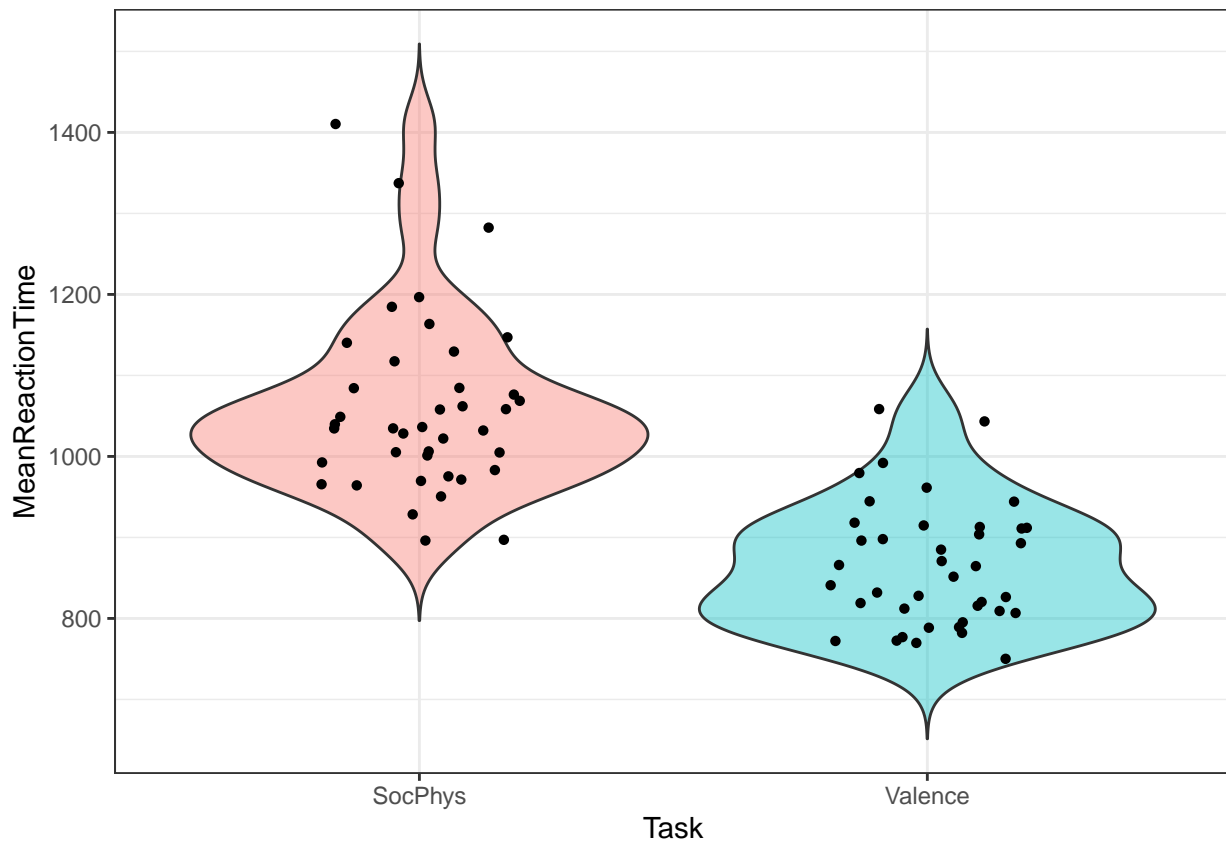
## ReactionTime by Task

```
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")
```

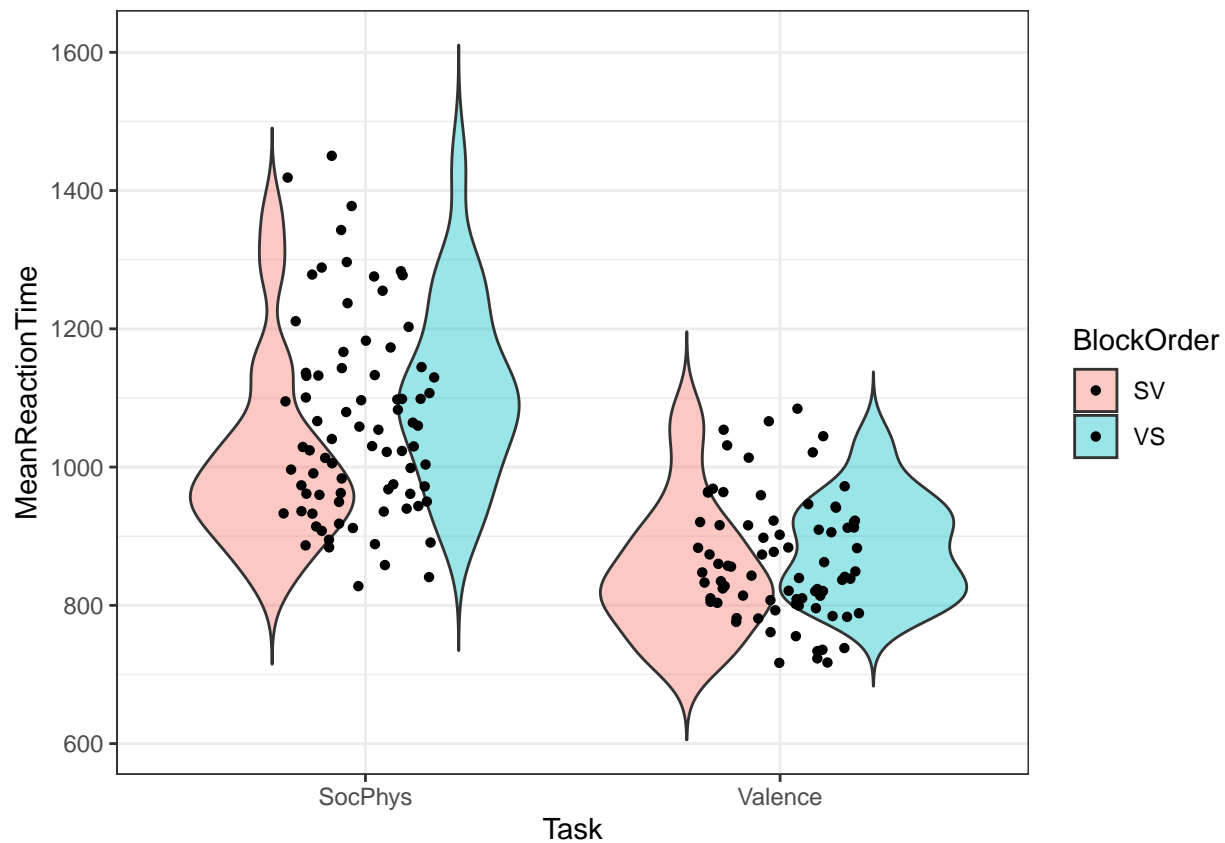


## ReactionTime by BlockOrder and Task

```
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(ReactionTime))
  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")

agr <- df.outliers.removed %>%
  group_by(Task, Word, BlockOrder) %>%
  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', 'Word'. You can override using the
## `.groups` argument.

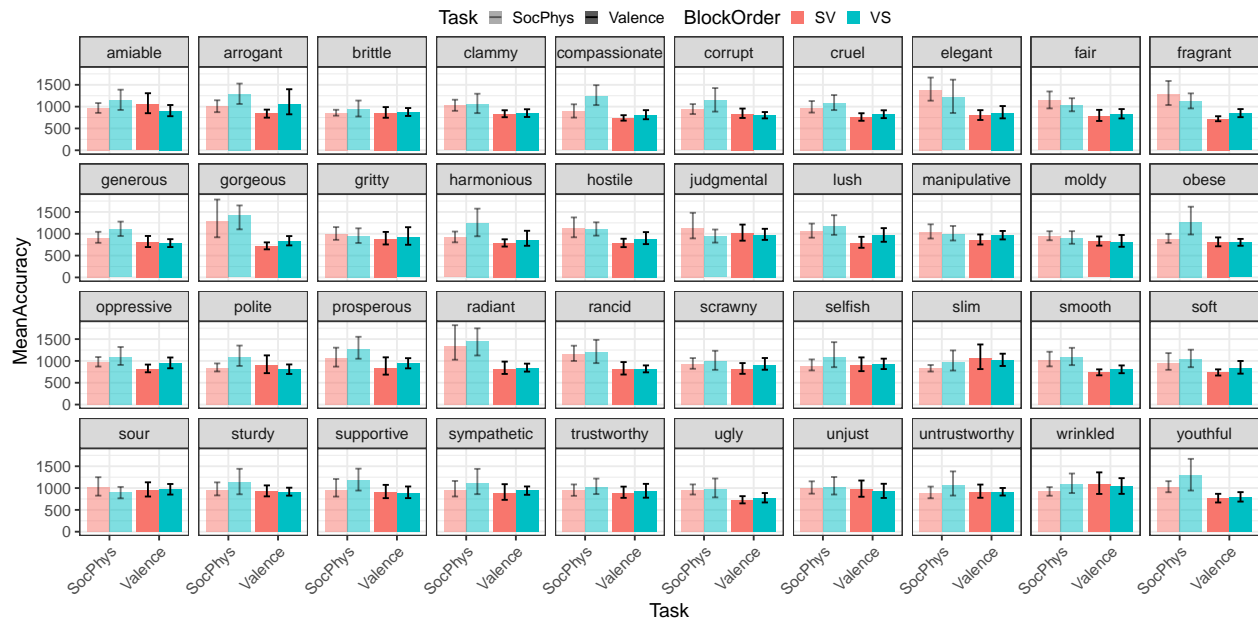
ggplot(agr, aes(x=Task, y=MeanReactionTime, alpha=Task, fill=BlockOrder)) +
  geom_bar(position="dodge", stat="identity") +
  geom_errorbar(aes(ymin=YMin, ymax=YMax), width=.25, position=position_dodge(0.9)) +
  facet_wrap(~Word, ncol=10) +
  xlab("Task") +
  ylab("MeanAccuracy") +
  # guides(fill=FALSE) +
  # guides(alpha=guide_legend(title="Task")) +
  theme(legend.key.size = unit(0.3, "cm"),
        legend.position = "top", # c(.5,1)
```

```

legend.direction = "horizontal",
legend.margin=margin(0,0,0,0),
legend.box.margin=margin(0,0,-5,-5),legend.spacing.y = unit(0.001, 'cm')) +
# scale_fill_manual(values=cbPalette) +
# scale_color_manual(values=cbPalette) +
scale_alpha_discrete(range = c(.5,1)) +
theme(axis.text.x = element_text(angle = 45, hjust = 1))

```

## Warning: Using alpha for a discrete variable is not advised.



## By ConcValCombo category and Task

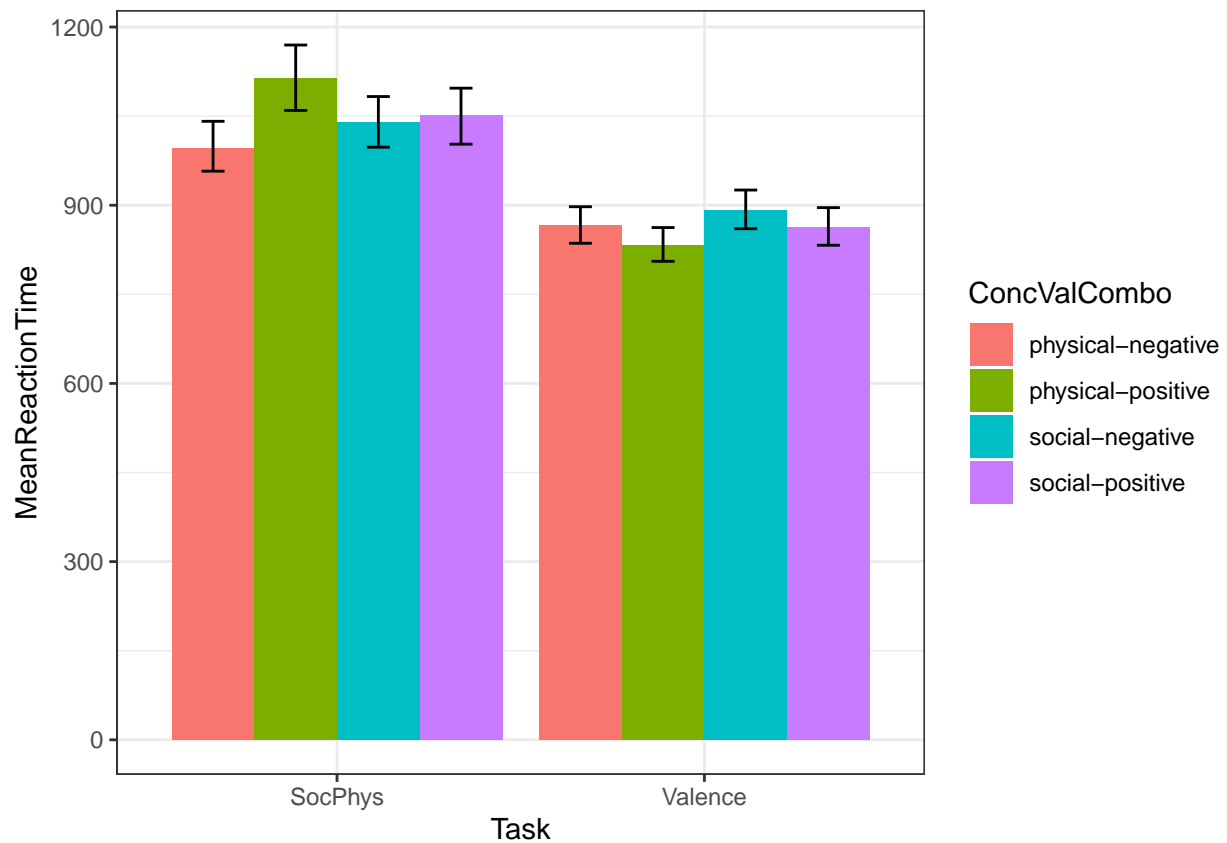
Mean Raw ReactionTime and Effects of Word Valence/Concreteness

```

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

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

```

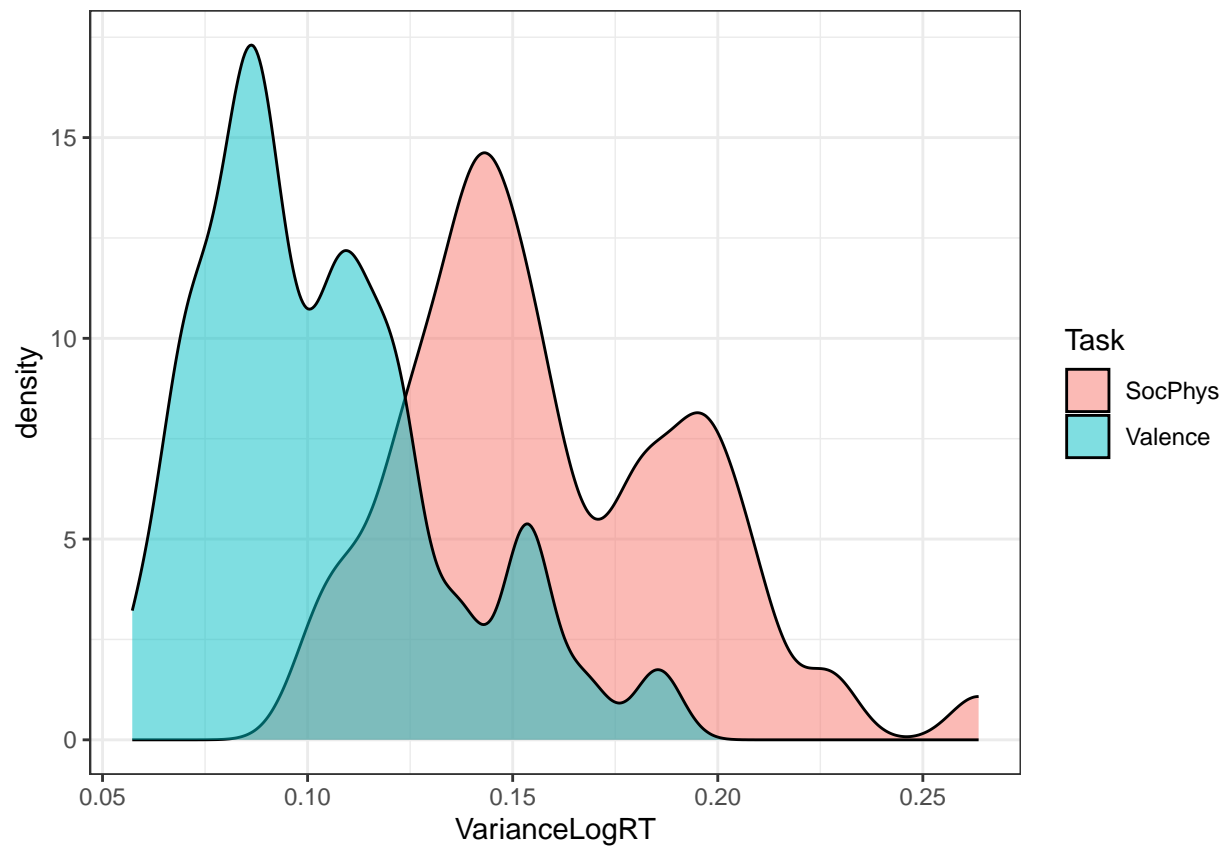


```
# theme(axis.text.x = element_text(angle = 45, hjust = 1))
# guides(fill = "none")
```

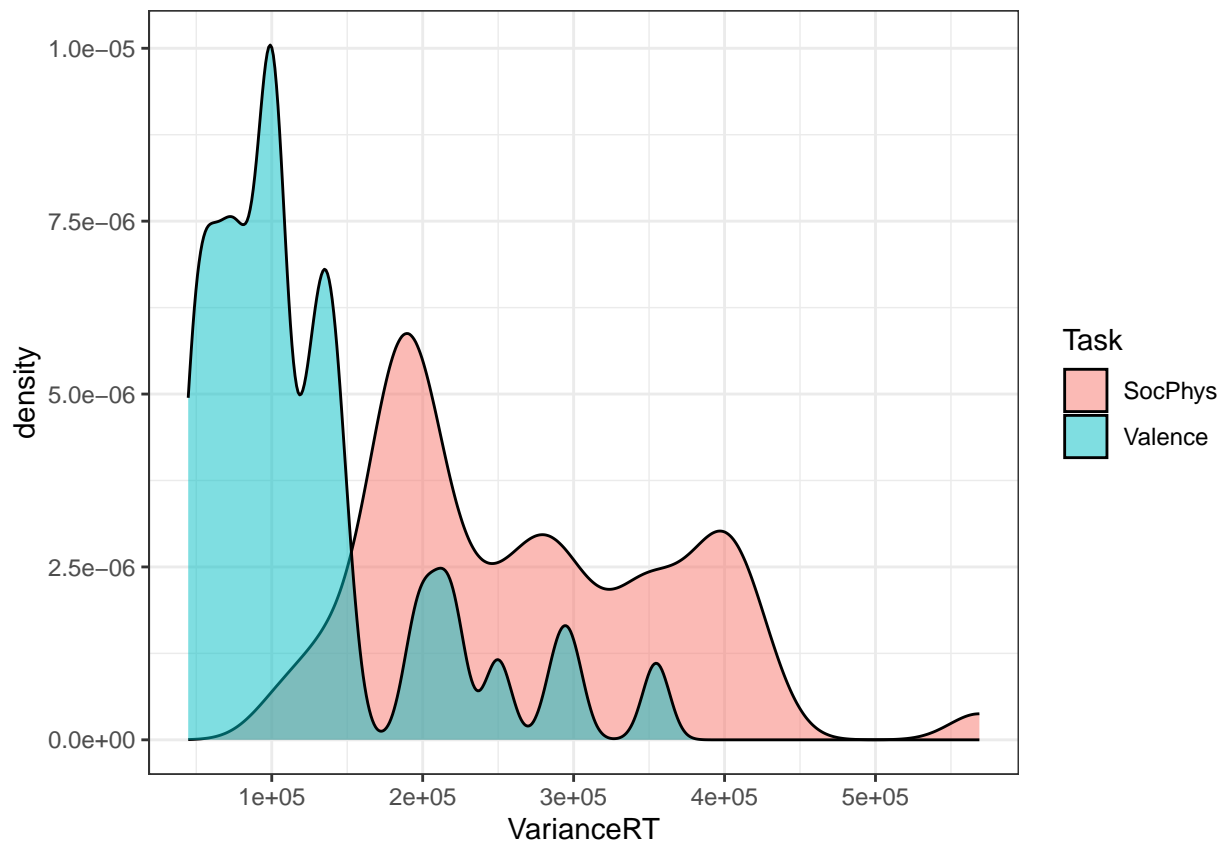
## Variance

```
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)
```

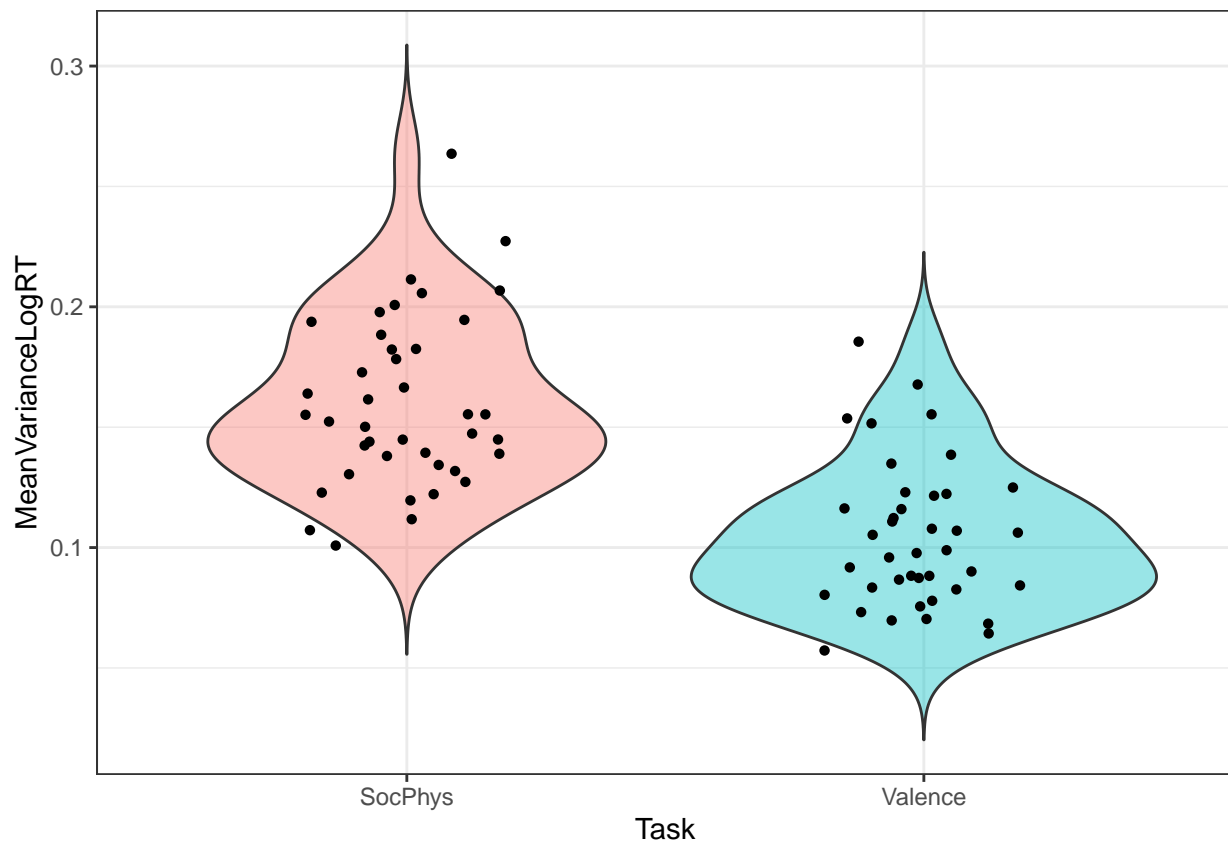


## ReactionTime by Task

```
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")
```

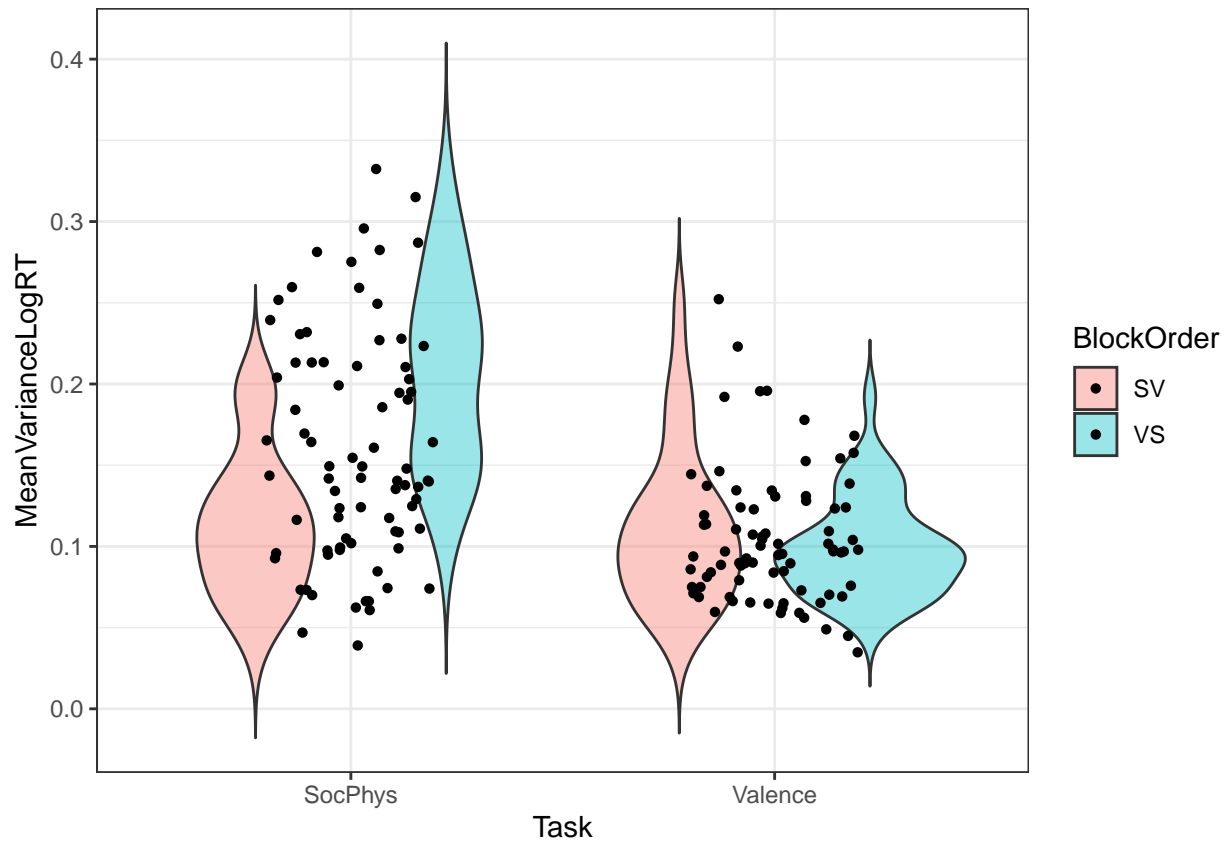


## LogReactionTime by BlockOrder and Task

```
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.

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

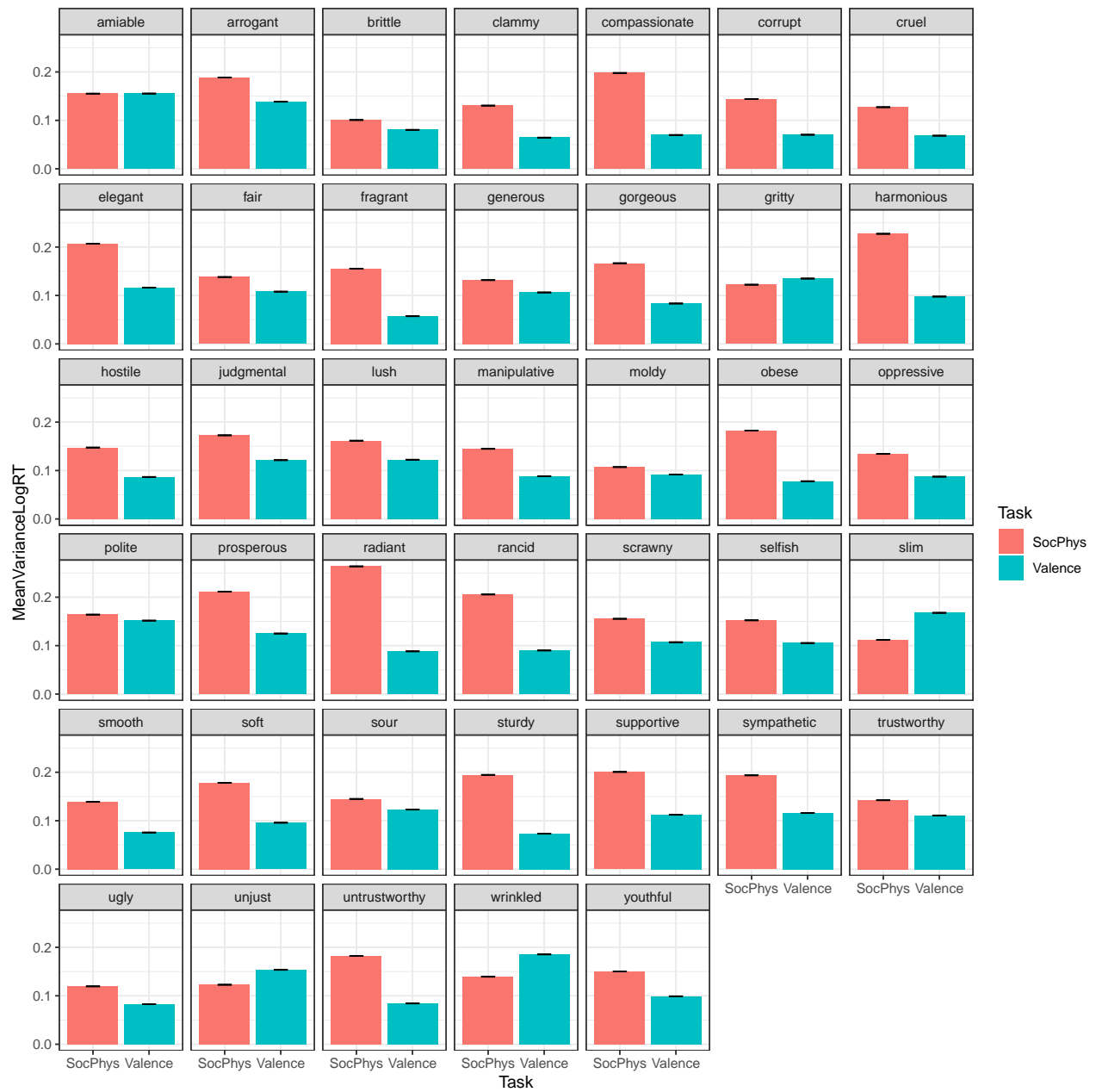


## By Item

```
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.

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")
```