SPL analysis

2023-10-31

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
##
## Attaching package: 'scales'
##
##
## The following object is masked from 'package:purrr':
##
##     discard
##
##
## The following object is masked from 'package:readr':
##
##     col_factor
##
##
##
## Attaching package: 'plotrix'
##
##
## The following object is masked from 'package:scales':
##
##     rescale
##
##
##
## Attaching package: 'janitor'
##
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
##
##
##
## Attaching package: 'data.table'
##
```

```
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
##
## The following object is masked from 'package:purrr':
##
##     transpose
##
##
##
## Attaching package: 'gridExtra'
##
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
##
## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 4.3.1

##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

cleaning up

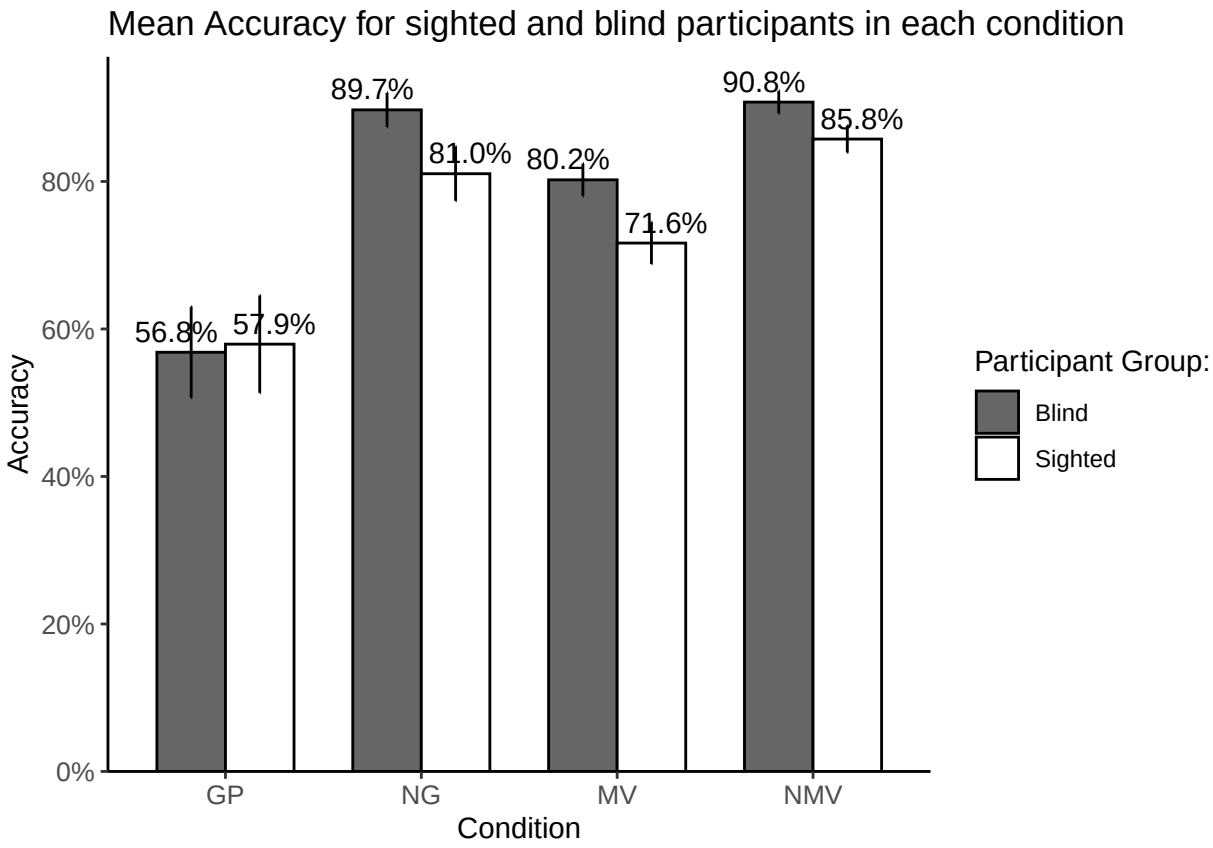
```
final_SPL_CB_Q$"RT+Duration" = final_SPL_CB_Q$"RT+dur(secs)"
final_SPL_CB_Q = final_SPL_CB_Q[, !names(final_SPL_CB_Q) %in% c("RT+dur(secs)")]

final_SPL_CB_S$"RT+Duration" = final_SPL_CB_S$"RT+dur(secs)"
final_SPL_CB_S = final_SPL_CB_S[, !names(final_SPL_CB_S) %in% c("RT+dur(secs)")]
```

#h3 Accuracy

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

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

```
acc_comparison <- subset(acc_comparison, CONDITION != 'FL')
ggplot(data = acc_comparison, aes(x = CONDITION, y = Accuracy, fill=GROUP,
                                                    ymin=Accuracy-se, ymax=Accuracy+se))+
  geom_bar(width = 0.7, position="dodge", stat = "identity", color = "black") +
  scale_fill_grey(start = 0.4, end = 1)+
  theme_bw()+
  scale_y_continuous(labels = scales::percent, breaks = seq(0, 1, by = 0.2), expand = expansion(mult = c
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  geom_errorbar(width = 0, position = position_dodge(0.7))+
  xlab("Condition") +
  labs(fill = "Participant Group:") +
  geom_text(aes(x=CONDITION, Accuracy,label=(perc)), position=position_dodge(width=1), vjust=-0.5, size=
  ggtitle("Mean Accuracy for sighted and blind participants in each condition")
```



#h4 Accuracy analysis: GP

```r
gp_cb_df <- subset(final_SPL_CB_Q[c("CONDITION", "Item", "ID", "Correct")], CONDITION %in% c("GP", "NG"))
gp_sc_df <- subset(final_SPL_SC_Q[c("CONDITION", "Item", "ID","Correct")], CONDITION %in% c("GP", "NG"))
gp_cb_df$Group = 0
gp_sc_df$Group = 1

acc_CB_q_gp <- data.frame(rbind(gp_cb_df, gp_sc_df))

filt_GPLG_glm_cb <- glmer(Correct ~ CONDITION * Group + (1|Item) + (1|ID),
                          data=acc_CB_q_gp, family=binomial(link="logit"),
                          control = glmerControl(optimizer = "bobyqa"))

summary(filt_GPLG_glm_cb)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Correct ~ CONDITION * Group + (1 | Item) + (1 | ID)
##    Data: acc_CB_q_gp
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   1648.7   1681.9   -818.4   1636.7     1851
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -13.6910  -0.3711   0.2141   0.4431   2.4373
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Item   (Intercept) 1.035    1.017
##  ID     (Intercept) 2.566    1.602
## Number of obs: 1857, groups:  Item, 168; ID, 45
##
## Fixed effects:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        0.52010    0.39240   1.325   0.1850
## CONDITIONNG        2.87014    0.33868   8.475   <2e-16 ***
## Group             -0.05491    0.55755  -0.098   0.9216
## CONDITIONNG:Group -1.11952    0.44414  -2.521   0.0117 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) CONDITIONNG Group
## CONDITIONNG  -0.279
## Group        -0.703  0.201
## CONDITIONNG:  0.218 -0.735      -0.312
```

#h4 Accuracy analysis: CB group only

```r
filt_GP_glm_CBonly <- glmer(Correct ~ CONDITION + (1|Item) + (1|ID),
                          data=gp_cb_df, family=binomial(link="logit"),
                          control = glmerControl(optimizer = "bobyqa"))
```

```r
summary(filt_GP_glm_CBonly)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Correct ~ CONDITION + (1 | Item) + (1 | ID)
##    Data: gp_cb_df
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##    754.6    774.0   -373.3    746.6      947
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -16.3704  -0.2420   0.1535   0.3448   2.8314
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Item   (Intercept) 1.944    1.394
##  ID     (Intercept) 3.519    1.876
## Number of obs: 951, groups:  Item, 84; ID, 23
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.5732     0.4693   1.222    0.222
## CONDITIONNG   3.1793     0.4311   7.375 1.65e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## CONDITIONNG -0.285
```

#h4 Accuracy analysis: SC group only

```r
filt_GP_glm_SConly <- glmer(Correct ~ CONDITION + (1|Item) + (1|ID),
                            data=gp_sc_df, family=binomial(link="logit"),
                            control = glmerControl(optimizer = "bobyqa"))

summary(filt_GP_glm_SConly)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Correct ~ CONDITION + (1 | Item) + (1 | ID)
##    Data: gp_sc_df
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##    888.0    907.3   -440.0    880.0      902
##
## Scaled residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -5.9391 -0.5306  0.2832  0.5091  2.0293
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Item   (Intercept) 0.4319   0.6572
##  ID     (Intercept) 1.9036   1.3797
## Number of obs: 906, groups:  Item, 84; ID, 22
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.4370     0.3329   1.313    0.189
## CONDITIONNG   1.6090     0.2391   6.730  1.7e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## CONDITIONNG -0.285
```

#h4 Accuracy analysis: MV

```
mv_cb_df <- subset(final_SPL_CB_Q[c("CONDITION", "Item", "ID", "Correct")], CONDITION %in% c("MV", "NMV
mv_sc_df <- subset(final_SPL_SC_Q[c("CONDITION", "Item", "ID","Correct")], CONDITION %in% c("MV", "NMV"]
mv_cb_df$Group = 0
mv_sc_df$Group = 1

acc_CB_q_mv <- data.frame(rbind(mv_cb_df, mv_sc_df))

filt_MV_glm_cb <- glmer(Correct ~ CONDITION *Group + (1|Item) + (1|ID),
                        data=acc_CB_q_mv, family=binomial(link="logit"),
                        control = glmerControl(optimizer = "bobyqa"))

summary(filt_MV_glm_cb)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Correct ~ CONDITION * Group + (1 | Item) + (1 | ID)
##    Data: acc_CB_q_mv
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   1580.4   1613.5   -784.2   1568.4     1846
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9058  0.1905  0.3041  0.4385  1.4917
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Item   (Intercept) 0.9882   0.9941
##  ID     (Intercept) 0.3448   0.5872
```

```
## Number of obs: 1852, groups:  Item, 168; ID, 45
##
## Fixed effects:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.72864    0.24251   7.128 1.02e-12 ***
## CONDITIONNMV         1.10074    0.31801   3.461 0.000537 ***
## Group               -0.59303    0.33308  -1.780 0.075004 .
## CONDITIONNMV:Group   0.04286    0.43230   0.099 0.921022
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##               (Intr) CONDITIONNMV Group
## CONDITIONNMV  -0.519
## Group         -0.715  0.386
## CONDITIONNMV:  0.397 -0.725       -0.548
```

#h4 Accuracy analysis: CB group only

```
filt_MV_glm_CBonly <- glmer(Correct ~ CONDITION + (1|Item) + (1|ID),
                            data=mv_cb_df, family=binomial(link="logit"),
                            control = glmerControl(optimizer = "bobyqa"))

summary(filt_MV_glm_CBonly)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Correct ~ CONDITION + (1 | Item) + (1 | ID)
##    Data: mv_cb_df
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##    716.3    735.6   -354.1    708.3      937
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.5354  0.1799  0.2737  0.4034  1.4967
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Item   (Intercept) 1.0546   1.0269
##  ID     (Intercept) 0.3197   0.5654
## Number of obs: 941, groups:  Item, 84; ID, 23
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.7346     0.2482   6.989 2.78e-12 ***
## CONDITIONNMV   1.1089     0.3268   3.394 0.000689 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
```

```
##             (Intr)
## CONDITIONNM -0.492
```

#h4 Accuracy analysis: SC group only

```
filt_MV_glm_SConly <- glmer(Correct ~ CONDITION + (1|Item) + (1|ID),
                           data=mv_sc_df, family=binomial(link="logit"),
                           control = glmerControl(optimizer = "bobyqa"))

summary(filt_MV_glm_SConly)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Correct ~ CONDITION + (1 | Item) + (1 | ID)
##    Data: mv_sc_df
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##    868.0    887.2   -430.0    860.0      907
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9144  0.1933  0.3367  0.4902  1.4742
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Item   (Intercept) 0.9359   0.9674
##  ID     (Intercept) 0.3633   0.6028
## Number of obs: 911, groups:  Item, 84; ID, 22
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.1333     0.2329   4.866 1.14e-06 ***
## CONDITIONNMV   1.1355     0.2951   3.849 0.000119 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## CONDITIONNM -0.501
```

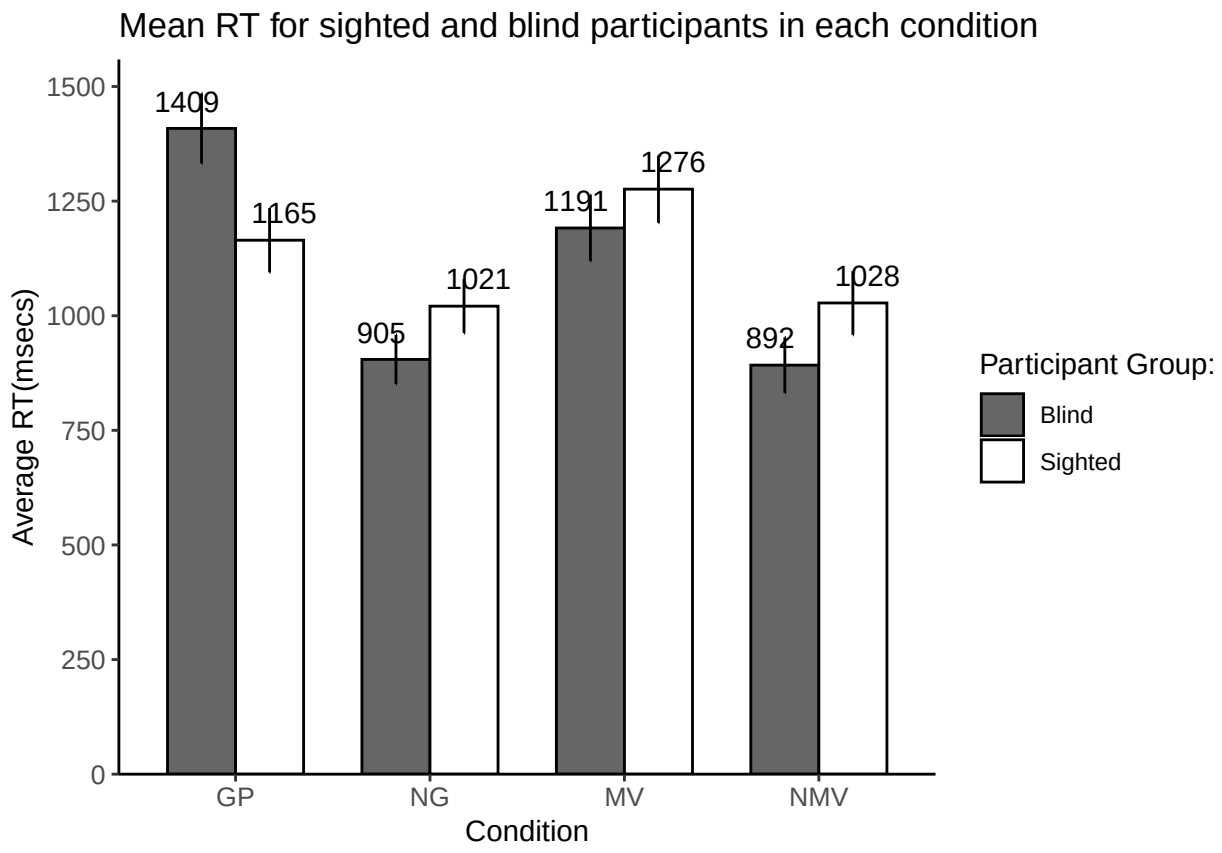
#h3 Question RT

```
filt_q_rt_df <- data.frame(
  CONDITION = c('GP','NG','MV','NMV','GP','NG','MV','NMV'),
  GROUP = c(rep("Blind",4),rep("Sighted",4)),
  avg = c(
    mean(subset(final_SPL_CB_Q, CONDITION == 'GP')$RT),
    mean(subset(final_SPL_CB_Q, CONDITION == 'NG')$RT),
    mean(subset(final_SPL_CB_Q, CONDITION == 'MV')$RT),
    mean(subset(final_SPL_CB_Q, CONDITION == 'NMV')$RT),
    mean(subset(final_SPL_SC_Q, CONDITION == 'GP')$RT),
```

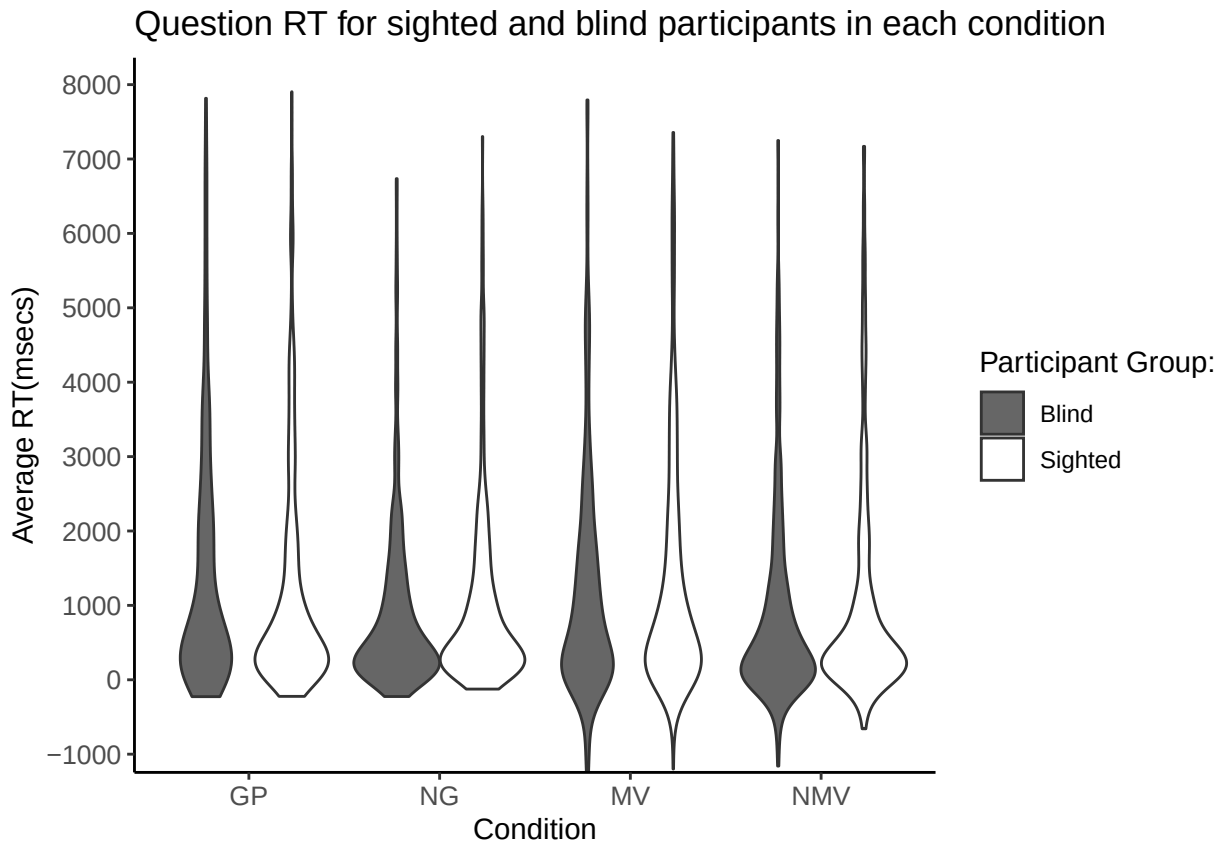
```
    mean(subset(final_SPL_SC_Q, CONDITION == 'NG')$RT),
    mean(subset(final_SPL_SC_Q, CONDITION == 'MV')$RT),
    mean(subset(final_SPL_SC_Q, CONDITION == 'NMV')$RT)
  ),
  se = c(
    std.error(subset(final_SPL_CB_Q, CONDITION == 'GP')$RT),
    std.error(subset(final_SPL_CB_Q, CONDITION == 'NG')$RT),
    std.error(subset(final_SPL_CB_Q, CONDITION == 'MV')$RT),
    std.error(subset(final_SPL_CB_Q, CONDITION == 'NMV')$RT),
    std.error(subset(final_SPL_SC_Q, CONDITION == 'GP')$RT),
    std.error(subset(final_SPL_SC_Q, CONDITION == 'NG')$RT),
    std.error(subset(final_SPL_SC_Q, CONDITION == 'MV')$RT),
    std.error(subset(final_SPL_SC_Q, CONDITION == 'NMV')$RT)
  ),
  avg_w_dur = c(
    mean(subset(final_SPL_CB_Q, CONDITION == 'GP')$"RT+Duration"),
    mean(subset(final_SPL_CB_Q, CONDITION == 'NG')$"RT+Duration"),
    mean(subset(final_SPL_CB_Q, CONDITION == 'MV')$"RT+Duration"),
    mean(subset(final_SPL_CB_Q, CONDITION == 'NMV')$"RT+Duration"),
    mean(subset(final_SPL_SC_Q, CONDITION == 'GP')$"RT+Duration"),
    mean(subset(final_SPL_SC_Q, CONDITION == 'NG')$"RT+Duration"),
    mean(subset(final_SPL_SC_Q, CONDITION == 'MV')$"RT+Duration"),
    mean(subset(final_SPL_SC_Q, CONDITION == 'NMV')$"RT+Duration")
  ),
  se_w_dur = c(
    std.error(subset(final_SPL_CB_Q, CONDITION == 'GP')$"RT+Duration"),
    std.error(subset(final_SPL_CB_Q, CONDITION == 'NG')$"RT+Duration"),
    std.error(subset(final_SPL_CB_Q, CONDITION == 'MV')$"RT+Duration"),
    std.error(subset(final_SPL_CB_Q, CONDITION == 'NMV')$"RT+Duration"),
    std.error(subset(final_SPL_SC_Q, CONDITION == 'GP')$"RT+Duration"),
    std.error(subset(final_SPL_SC_Q, CONDITION == 'NG')$"RT+Duration"),
    std.error(subset(final_SPL_SC_Q, CONDITION == 'MV')$"RT+Duration"),
    std.error(subset(final_SPL_SC_Q, CONDITION == 'NMV')$"RT+Duration")
  )
)
```

```
ggplot(data = filt_q_rt_df, aes(x = factor(CONDITION, levels = c('GP','NG','MV','NMV')), y = avg, fill=
                                                    ymin=avg-se, ymax=avg+se))+
  geom_bar(width = 0.7, position="dodge", stat = "identity", color = "black") +
  scale_fill_grey(start = 0.4, end = 1)+
  theme_bw()+
  scale_y_continuous(breaks = seq(0, 1500, by = 250), expand = expansion(mult = c(0, 0.05))) +
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  geom_errorbar(width = 0, position = position_dodge(0.7))+
  xlab("Condition") +
  ylab("Average RT(msecs)") +
  labs(fill = "Participant Group:") +
  geom_text(aes(x=CONDITION, avg,label=(round(avg,0))), position=position_dodge(width=1), vjust=-0.8, si
```

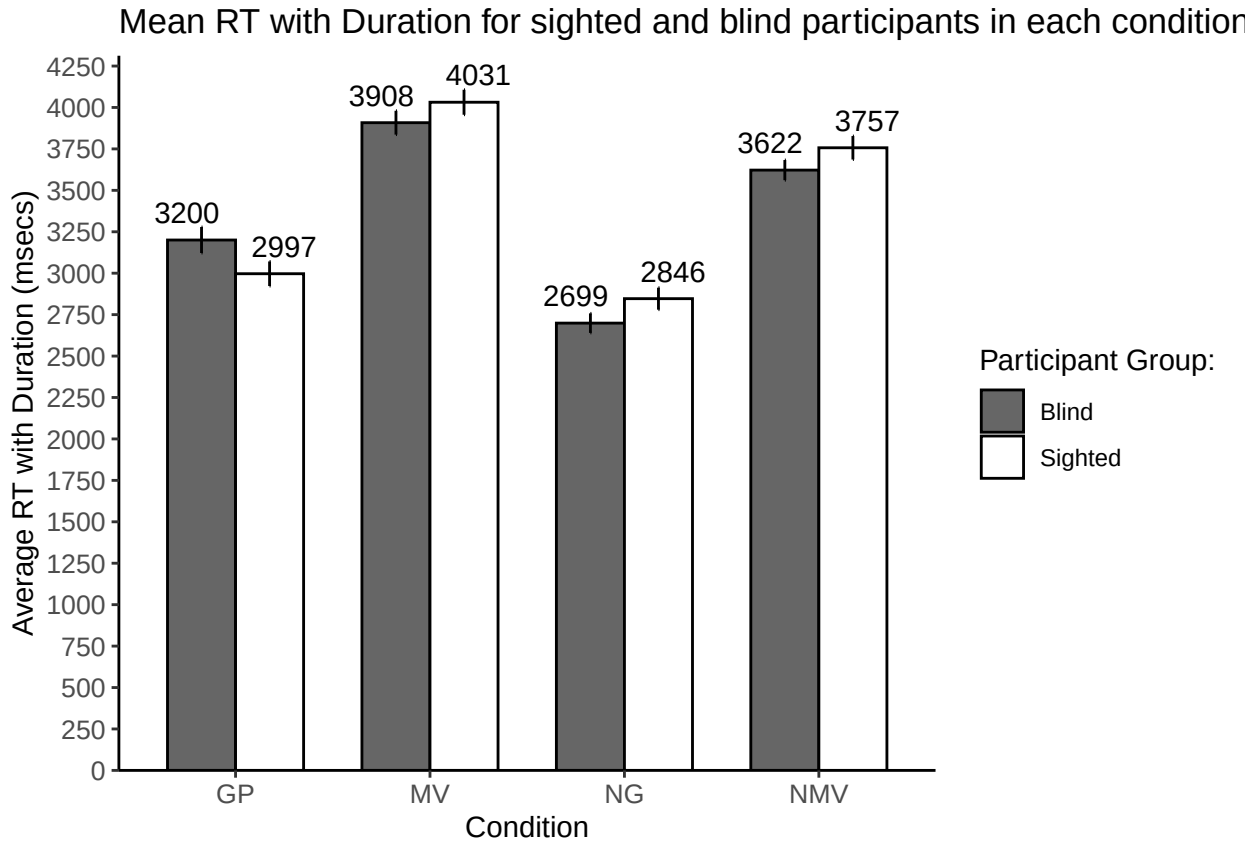
```
  ggtitle("Mean RT for sighted and blind participants in each condition")
```




```
cb_temp = subset(final_SPL_CB_Q, CONDITION != 'FL')[,c('ID','CONDITION','Item','RT','RT+Duration')]
cb_temp$Group = 'Blind'
sc_temp = subset(final_SPL_SC_Q, CONDITION != 'FL')[,c('ID','CONDITION','Item','RT','RT+Duration')]
sc_temp$Group = 'Sighted'
SPL_Q_violin <- data.frame(rbind(cb_temp,sc_temp))

ggplot(data = SPL_Q_violin, aes(x = CONDITION, y = RT,fill=Group))+
  geom_violin()+
  scale_fill_grey(start = 0.4, end = 1)+
  theme_bw()+
  scale_y_continuous(breaks = seq(-10000, 15000, by = 1000), expand = expansion(mult = c(0, 0.05))) +
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  xlab("Condition") +
  ylab("Average RT(msecs)") +
  labs(fill = "Participant Group:") +
  ggtitle("Question RT for sighted and blind participants in each condition")
```




```
ggplot(data = filt_q_rt_df, aes(x = CONDITION, y = avg_w_dur, fill=GROUP,
                                                ymin=avg_w_dur-se_w_dur, ymax=avg_w_dur+se_w_dur))+
  geom_bar(width = 0.7, position="dodge", stat = "identity", color = "black") +
  scale_fill_grey(start = 0.4, end = 1)+
  theme_bw()+
  scale_y_continuous(breaks = seq(0, 4750, by = 250), expand = expansion(mult = c(0, 0.05))) +
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  geom_errorbar(width = 0, position = position_dodge(0.7))+
  xlab("Condition") +
  ylab("Average RT with Duration (msecs)") +
  labs(fill = "Participant Group:") +
  geom_text(aes(CONDITION, avg_w_dur,label=(round(avg_w_dur,0))), position=position_dodge(width=1), vjus
  ggtitle("Mean RT with Duration for sighted and blind participants in each condition")
```



#h3 RT analysis

Applying 2000ms constant and natural log transformation

```
q_gp_cb_df <- subset(final_SPL_CB_Q[c("CONDITION", "Item", "ID","Correct", "RT", "RT+Duration")], CONDI
q_gp_sc_df <- subset(final_SPL_SC_Q[c("CONDITION", "Item", "ID","Correct","RT", "RT+Duration")], CONDIT
q_gp_cb_df$Group = 0
q_gp_sc_df$Group = 1

rt_q_gp <- data.frame(rbind(q_gp_cb_df, q_gp_sc_df))

rt_q_gp$ln_RT <- log(rt_q_gp$RT+2000)
rt_q_gp$ln_RT_Duration <- log(rt_q_gp$RT.Duration)
```

```
#log transformed RT
log_q_gp_lm <- lmer(ln_RT ~ CONDITION * Group + (1|Item) + (1|ID), data = rt_q_gp, REML = FALSE)
summary(log_q_gp_lm)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: ln_RT ~ CONDITION * Group + (1 | Item) + (1 | ID)
##    Data: rt_q_gp
##
##      AIC      BIC   logLik deviance df.resid
##   1058.6   1097.3   -522.3   1044.6     1850
##
## Scaled residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -2.6402 -0.6656 -0.1730  0.4642  3.5824
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Item     (Intercept) 0.007718 0.08785
##  ID       (Intercept) 0.031909 0.17863
##  Residual             0.090691 0.30115
## Number of obs: 1857, groups:  Item, 168; ID, 45
##
## Fixed effects:
##                   Estimate Std. Error t value
## (Intercept)        8.04486    0.04198 191.617
## CONDITIONNG       -0.12610    0.02738  -4.605
## Group             -0.05587    0.06004  -0.931
## CONDITIONNG:Group  0.08659    0.03904   2.218
##
## Correlation of Fixed Effects:
##              (Intr) CONDITIONNG Group
## CONDITIONNG  -0.326
## Group        -0.699  0.228
## CONDITIONNG:  0.229 -0.701      -0.326
```

```
#RT+duration
q_dur_gp_lm<- lmer(RT.Duration ~ CONDITION * Group + (1|Item) + (1|ID), data = rt_q_gp,REML = FALSE,con
summary(q_dur_gp_lm)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: RT.Duration ~ CONDITION * Group + (1 | Item) + (1 | ID)
##    Data: rt_q_gp
##
##      AIC      BIC   logLik deviance df.resid
##  32101.4  32140.1 -16043.7  32087.4     1850
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4751 -0.6041 -0.2024  0.2854  4.4909
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Item     (Intercept)  255068   505.0
##  ID       (Intercept)  412852   642.5
##  Residual             1611431  1269.4
## Number of obs: 1857, groups:  Item, 168; ID, 45
##
## Fixed effects:
##                   Estimate Std. Error t value
## (Intercept)         3217.3      165.6  19.426
## CONDITIONNG         -508.8      137.6  -3.697
## Group               -192.4      236.6  -0.813
## CONDITIONNG:Group    331.6      195.8   1.693
##
## Correlation of Fixed Effects:
##              (Intr) CONDITIONNG Group
```

```
## CONDITIONNG  -0.416
## Group        -0.700  0.291
## CONDITIONNG:  0.292 -0.703      -0.415
```

MV transformation

```
q_mv_cb_df <- subset(final_SPL_CB_Q[c("CONDITION", "Item", "ID","Correct", "RT", "RT+Duration")], CONDI
q_mv_sc_df <- subset(final_SPL_SC_Q[c("CONDITION", "Item", "ID","Correct","RT", "RT+Duration")], CONDIT
q_mv_cb_df$Group = 0
q_mv_sc_df$Group = 1

rt_q_mv <- data.frame(rbind(q_mv_cb_df, q_mv_sc_df))

rt_q_mv$ln_RT <- log(rt_q_mv$RT+2000)
rt_q_mv$ln_RT_Duration <- log(rt_q_mv$RT.Duration)
```

```
#log transformed RT
log_q_mv_lm <- lmer(ln_RT ~ CONDITION * Group + (1|Item) + (1|ID), data = rt_q_mv, REML = FALSE,control
summary(log_q_mv_lm)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: ln_RT ~ CONDITION * Group + (1 | Item) + (1 | ID)
##    Data: rt_q_mv
##
##      AIC      BIC   logLik deviance df.resid
##   1413.4   1452.1   -699.7   1399.4     1845
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2349 -0.6061 -0.1623  0.4360  3.4306
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Item     (Intercept) 0.02663  0.1632
##  ID       (Intercept) 0.02681  0.1637
##  Residual             0.10432  0.3230
## Number of obs: 1852, groups:  Item, 168; ID, 45
##
## Fixed effects:
##                       Estimate Std. Error t value
## (Intercept)         7.9798114  0.0450074 177.300
## CONDITIONNMV       -0.0872763  0.0413900  -2.109
## Group               0.0318615  0.0641804   0.496
## CONDITIONNMV:Group  0.0009835  0.0587439   0.017
##
## Correlation of Fixed Effects:
##               (Intr) CONDITIONNMV Group
## CONDITIONNMV  -0.461
## Group         -0.701  0.324
## CONDITIONNMV:  0.325 -0.705       -0.459
```

```
#RT+duration
q_dur_mv_lm<- lmer(RT.Duration ~ CONDITION * Group + (1|Item) + (1|ID), data = rt_q_mv, REML = FALSE,co
summary(log_q_mv_lm)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: ln_RT ~ CONDITION * Group + (1 | Item) + (1 | ID)
##    Data: rt_q_mv
##
##      AIC      BIC   logLik deviance df.resid
##   1413.4   1452.1   -699.7   1399.4     1845
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2349 -0.6061 -0.1623  0.4360  3.4306
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Item     (Intercept) 0.02663  0.1632
##  ID       (Intercept) 0.02681  0.1637
##  Residual             0.10432  0.3230
## Number of obs: 1852, groups:  Item, 168; ID, 45
##
## Fixed effects:
##                      Estimate Std. Error t value
## (Intercept)         7.9798114  0.0450074 177.300
## CONDITIONNMV       -0.0872763  0.0413900  -2.109
## Group               0.0318615  0.0641804   0.496
## CONDITIONNMV:Group  0.0009835  0.0587439   0.017
##
## Correlation of Fixed Effects:
##               (Intr) CONDITIONNMV Group
## CONDITIONNMV  -0.461
## Group         -0.701  0.324
## CONDITIONNMV:  0.325 -0.705       -0.459
```

#h3 Segment by Segment RT

```
library(plotrix)
gp_CB_S <- subset(final_SPL_CB_S, CONDITION == 'GP')
ng_CB_S <- subset(final_SPL_CB_S, CONDITION == 'NG')


df_gp_CB_SegRT <- data.frame(
  s_label <- c(1:8),
  s_words <- c("While", "the man","hunted","the pheasant","the brown","and graceful deer", "ran", "into
  gp_avg <- c(
    mean(subset(gp_CB_S, Seg == 'seg01')$RT),
    mean(subset(gp_CB_S, Seg == 'seg02')$RT),
    mean(subset(gp_CB_S, Seg == 'seg03')$RT),
    NA,
    mean(subset(gp_CB_S, Seg == 'seg04')$RT),
    mean(subset(gp_CB_S, Seg == 'seg05')$RT),
    mean(subset(gp_CB_S, Seg == 'seg06')$RT),
```

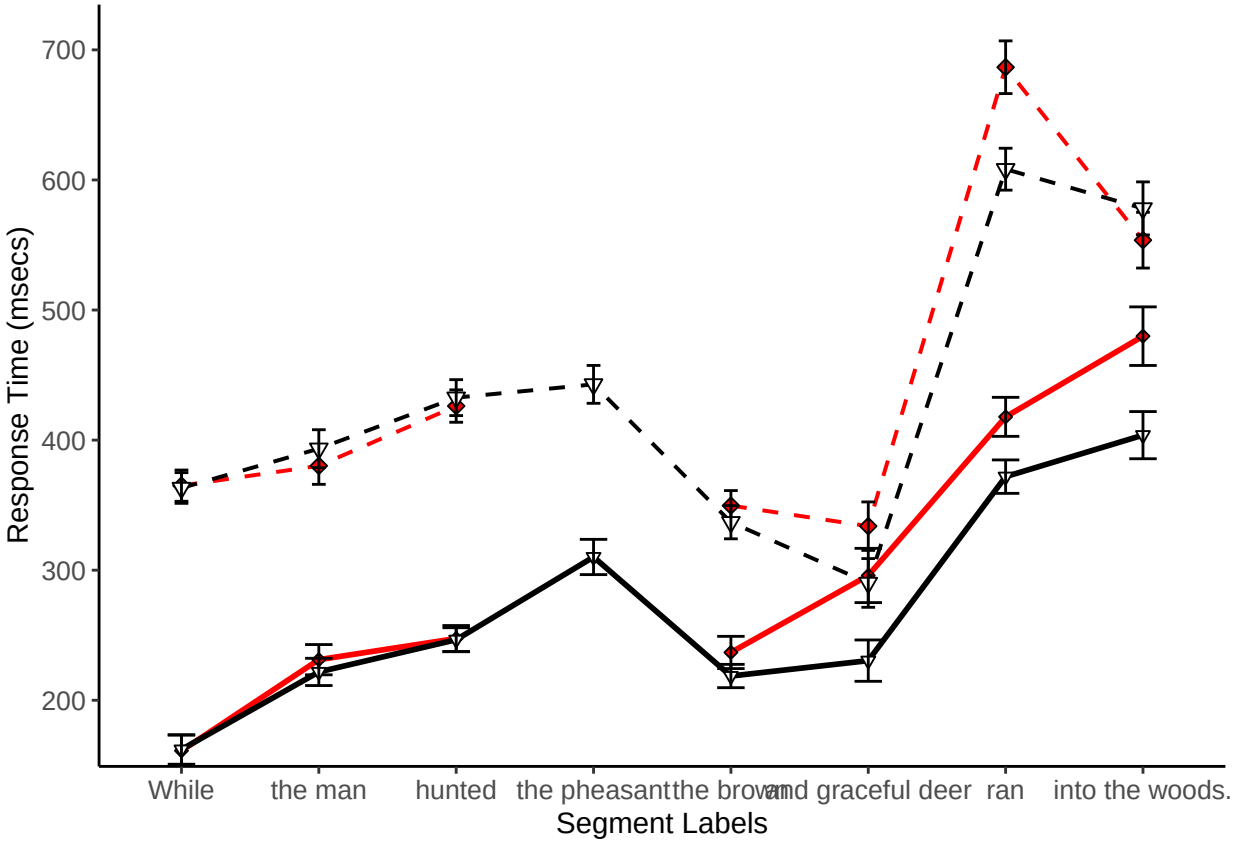
```
    mean(subset(gp_CB_S, Seg == 'seg07')$RT)
  ),
  gp_se <- c(
    std.error(subset(gp_CB_S, Seg == 'seg01')$RT),
    std.error(subset(gp_CB_S, Seg == 'seg02')$RT),
    std.error(subset(gp_CB_S, Seg == 'seg03')$RT),
    NA,
    std.error(subset(gp_CB_S, Seg == 'seg04')$RT),
    std.error(subset(gp_CB_S, Seg == 'seg05')$RT),
    std.error(subset(gp_CB_S, Seg == 'seg06')$RT),
    std.error(subset(gp_CB_S, Seg == 'seg07')$RT)
  ),
  ng_avg <- c(
    mean(subset(ng_CB_S, Seg == 'seg01')$RT),
    mean(subset(ng_CB_S, Seg == 'seg02')$RT),
    mean(subset(ng_CB_S, Seg == 'seg03')$RT),
    mean(subset(ng_CB_S, Seg == 'seg04')$RT),
    mean(subset(ng_CB_S, Seg == 'seg05')$RT),
    mean(subset(ng_CB_S, Seg == 'seg06')$RT),
    mean(subset(ng_CB_S, Seg == 'seg07')$RT),
    mean(subset(ng_CB_S, Seg == 'seg08')$RT)
  ),
  ng_se <- c(
    std.error(subset(ng_CB_S, Seg == 'seg01')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg02')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg03')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg04')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg05')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg06')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg07')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg08')$RT)
  )
)
colnames(df_gp_CB_SegRT)<-(c("s_label", "s_words","gp_avg","gp_se","ng_avg","ng_se"))


gp_SC_S <- subset(final_SPL_SC_S, CONDITION == 'GP')
ng_SC_S <- subset(final_SPL_SC_S, CONDITION == 'NG')


df_gp_SC_SegRT <- data.frame(
  s_label <- c(1:8),
  s_words <- c("While", "the man","hunted","the pheasant","the brown","and graceful deer", "ran", "into
  gp_avg <- c(
    mean(subset(gp_SC_S, Seg == 'seg01')$RT),
    mean(subset(gp_SC_S, Seg == 'seg02')$RT),
    mean(subset(gp_SC_S, Seg == 'seg03')$RT),
    NA,
    mean(subset(gp_SC_S, Seg == 'seg04')$RT),
    mean(subset(gp_SC_S, Seg == 'seg05')$RT),
    mean(subset(gp_SC_S, Seg == 'seg06')$RT),
    mean(subset(gp_SC_S, Seg == 'seg07')$RT)
  ),
```

```
  gp_se <- c(
    std.error(subset(gp_SC_S, Seg == 'seg01')$RT),
    std.error(subset(gp_SC_S, Seg == 'seg02')$RT),
    std.error(subset(gp_SC_S, Seg == 'seg03')$RT),
    NA,
    std.error(subset(gp_SC_S, Seg == 'seg04')$RT),
    std.error(subset(gp_SC_S, Seg == 'seg05')$RT),
    std.error(subset(gp_SC_S, Seg == 'seg06')$RT),
    std.error(subset(gp_SC_S, Seg == 'seg07')$RT)
  ),
  ng_avg <- c(
    mean(subset(ng_SC_S, Seg == 'seg01')$RT),
    mean(subset(ng_SC_S, Seg == 'seg02')$RT),
    mean(subset(ng_SC_S, Seg == 'seg03')$RT),
    mean(subset(ng_SC_S, Seg == 'seg04')$RT),
    mean(subset(ng_SC_S, Seg == 'seg05')$RT),
    mean(subset(ng_SC_S, Seg == 'seg06')$RT),
    mean(subset(ng_SC_S, Seg == 'seg07')$RT),
    mean(subset(ng_SC_S, Seg == 'seg08')$RT)
  ),
  ng_se <- c(
    std.error(subset(ng_SC_S, Seg == 'seg01')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg02')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg03')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg04')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg05')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg06')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg07')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg08')$RT)
  )
)
colnames(df_gp_SC_SegRT)<-(c("s_label", "s_words","gp_avg","gp_se","ng_avg","ng_se"))

df_gp_SegRT <- data.frame(rbind(df_gp_CB_SegRT,df_gp_SC_SegRT))
df_gp_SegRT$Group <- c(rep("Blind",8),rep("Sighted",8))
```

```
df_gp_CB_SegRT$s_words <- factor(df_gp_CB_SegRT$s_words, levels = c("While", "the man","hunted","the pho
ggplot(data = df_gp_CB_SegRT, aes(x = s_words))+
  geom_line(aes(y = gp_avg, group = 1), color = "red",size = 1) +
  geom_point(aes(y = gp_avg, group = 1),shape = 23, linewidth = 3, fill = "red", color = "black")+
  geom_errorbar(aes(y = gp_avg, group = 1, ymin=gp_avg-gp_se, ymax=gp_avg+gp_se),width = .2, position =
  geom_line(aes(y = ng_avg, group = 1), size = 1) +
  geom_point(aes(y = ng_avg, group = 1),shape = 25, linewidth = 3,fill = "white", color = "black")+
  geom_errorbar(aes(y = ng_avg, group = 1, ymin=ng_avg-ng_se, ymax=ng_avg+ng_se),width = .2, position =

  geom_line(data = df_gp_SC_SegRT, aes(y = gp_avg, group = 1), color = "red",linewidth = 0.7, linetype =
  geom_point(data = df_gp_SC_SegRT,aes(y = gp_avg, group = 1),shape = 23, size = 2, fill = "red", color
  geom_errorbar(data = df_gp_SC_SegRT,aes(y = gp_avg, group = 1, ymin=gp_avg-gp_se, ymax=gp_avg+gp_se),w
  geom_line(data = df_gp_SC_SegRT,aes(y = ng_avg, group = 1), linewidth = 0.7, linetype = "dashed") +
  geom_point(data = df_gp_SC_SegRT,aes(y = ng_avg, group = 1),shape = 25, size = 2,fill = "white", colo
  geom_errorbar(data = df_gp_SC_SegRT,aes(y = ng_avg, group = 1, ymin=ng_avg-ng_se, ymax=ng_avg+ng_se),w

  scale_fill_grey(start = 0.4, end = 1)+
```

```
  theme_bw()+
  scale_y_continuous(breaks = seq(0, 700, by = 100), expand = expansion(mult = c(0, 0.05))) +
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  xlab("Segment Labels") +
  ylab("Response Time (msecs)")+
  labs(color = "Legend", shape = "Group")
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

## Warning in geom_point(aes(y = gp_avg, group = 1), shape = 23, linewidth = 3, :
## Ignoring unknown parameters: 'linewidth'

## Warning in geom_point(aes(y = ng_avg, group = 1), shape = 25, linewidth = 3, :
## Ignoring unknown parameters: 'linewidth'

## Warning: Removed 1 rows containing missing values ('geom_point()').
## Removed 1 rows containing missing values ('geom_point()').
```



Creating segment average dataframe with different format for RT plotting

```
library(plotrix)
gp_CB_S <- subset(final_SPL_CB_S, CONDITION == 'GP')
ng_CB_S <- subset(final_SPL_CB_S, CONDITION == 'NG')


df_gp_CB_SegRT <- data.frame(
  s_label <- c(1:8),
  s_words <- c("While", "the man","hunted","the pheasant","the brown","and graceful deer", "ran", "into
  CONDITION <- c(rep("GP",8),rep("NG",8)),
  avg <- c(
    mean(subset(gp_CB_S, Seg == 'seg01')$RT),
    mean(subset(gp_CB_S, Seg == 'seg02')$RT),
    mean(subset(gp_CB_S, Seg == 'seg03')$RT),
    NA,
    mean(subset(gp_CB_S, Seg == 'seg04')$RT),
    mean(subset(gp_CB_S, Seg == 'seg05')$RT),
    mean(subset(gp_CB_S, Seg == 'seg06')$RT),
    mean(subset(gp_CB_S, Seg == 'seg07')$RT),

    mean(subset(ng_CB_S, Seg == 'seg01')$RT),
    mean(subset(ng_CB_S, Seg == 'seg02')$RT),
    mean(subset(ng_CB_S, Seg == 'seg03')$RT),
    mean(subset(ng_CB_S, Seg == 'seg04')$RT),
    mean(subset(ng_CB_S, Seg == 'seg05')$RT),
    mean(subset(ng_CB_S, Seg == 'seg06')$RT),
    mean(subset(ng_CB_S, Seg == 'seg07')$RT),
    mean(subset(ng_CB_S, Seg == 'seg08')$RT)
  ),
  se <- c(
    std.error(subset(gp_CB_S, Seg == 'seg01')$RT),
    std.error(subset(gp_CB_S, Seg == 'seg02')$RT),
    std.error(subset(gp_CB_S, Seg == 'seg03')$RT),
    NA,
    std.error(subset(gp_CB_S, Seg == 'seg04')$RT),
    std.error(subset(gp_CB_S, Seg == 'seg05')$RT),
    std.error(subset(gp_CB_S, Seg == 'seg06')$RT),
    std.error(subset(gp_CB_S, Seg == 'seg07')$RT),

    std.error(subset(ng_CB_S, Seg == 'seg01')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg02')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg03')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg04')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg05')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg06')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg07')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg08')$RT)
  )
)
colnames(df_gp_CB_SegRT)<-(c("s_label", "s_words","CONDITION","avg","se"))


gp_SC_S <- subset(final_SPL_SC_S, CONDITION == 'GP')
```

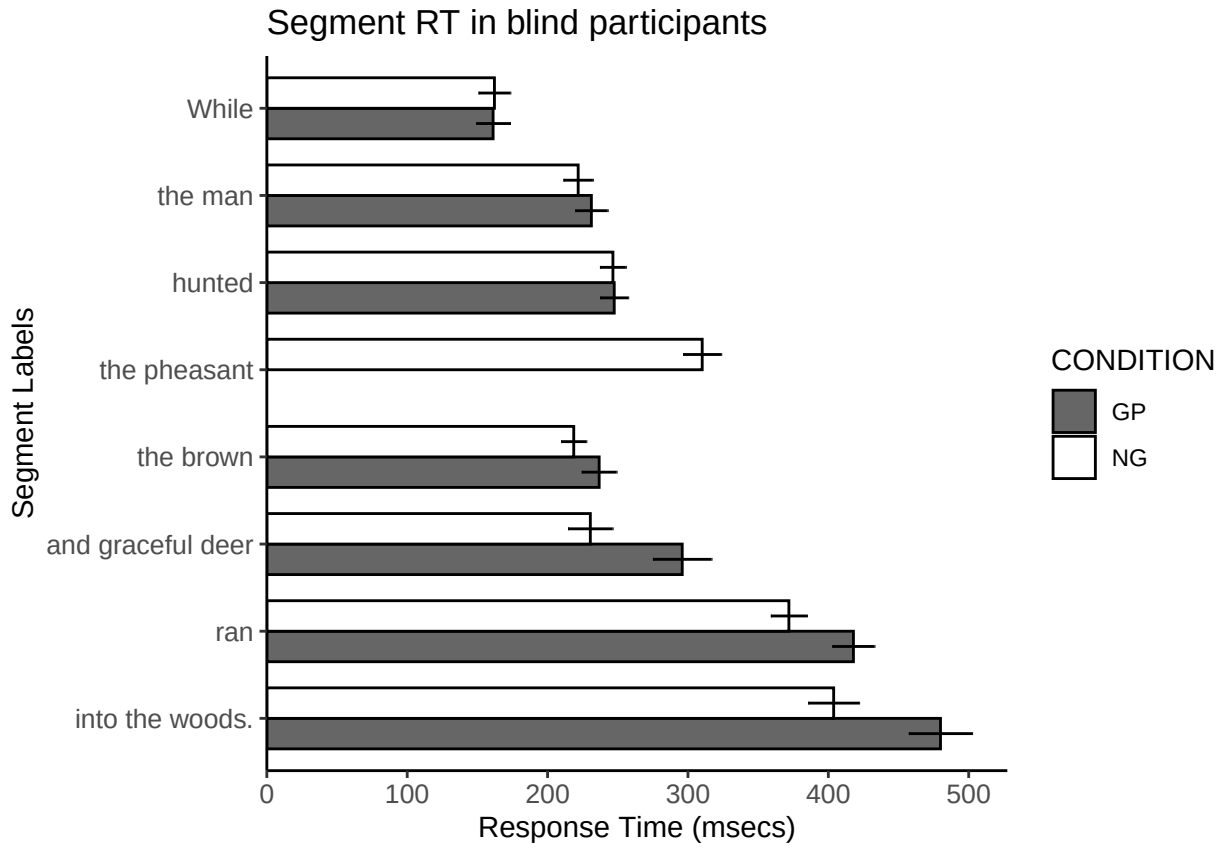
```
ng_SC_S <- subset(final_SPL_SC_S, CONDITION == 'NG')

df_gp_SC_SegRT <- data.frame(
  s_label <- c(1:8),
  s_words <- c("While", "the man","hunted","the pheasant","the brown","and graceful deer", "ran", "into
  CONDITION <- c(rep("GP",8),rep("NG",8)),
  avg <- c(
    mean(subset(gp_SC_S, Seg == 'seg01')$RT),
    mean(subset(gp_SC_S, Seg == 'seg02')$RT),
    mean(subset(gp_SC_S, Seg == 'seg03')$RT),
    NA,
    mean(subset(gp_SC_S, Seg == 'seg04')$RT),
    mean(subset(gp_SC_S, Seg == 'seg05')$RT),
    mean(subset(gp_SC_S, Seg == 'seg06')$RT),
    mean(subset(gp_SC_S, Seg == 'seg07')$RT),

    mean(subset(ng_SC_S, Seg == 'seg01')$RT),
    mean(subset(ng_SC_S, Seg == 'seg02')$RT),
    mean(subset(ng_SC_S, Seg == 'seg03')$RT),
    mean(subset(ng_SC_S, Seg == 'seg04')$RT),
    mean(subset(ng_SC_S, Seg == 'seg05')$RT),
    mean(subset(ng_SC_S, Seg == 'seg06')$RT),
    mean(subset(ng_SC_S, Seg == 'seg07')$RT),
    mean(subset(ng_SC_S, Seg == 'seg08')$RT)
  ),
  se <- c(
    std.error(subset(gp_SC_S, Seg == 'seg01')$RT),
    std.error(subset(gp_SC_S, Seg == 'seg02')$RT),
    std.error(subset(gp_SC_S, Seg == 'seg03')$RT),
    NA,
    std.error(subset(gp_SC_S, Seg == 'seg04')$RT),
    std.error(subset(gp_SC_S, Seg == 'seg05')$RT),
    std.error(subset(gp_SC_S, Seg == 'seg06')$RT),
    std.error(subset(gp_SC_S, Seg == 'seg07')$RT),

    std.error(subset(ng_SC_S, Seg == 'seg01')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg02')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg03')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg04')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg05')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg06')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg07')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg08')$RT)
  )
)
colnames(df_gp_SC_SegRT)<-(c("s_label", "s_words","CONDITION","avg","se"))

df_gp_SegRT <- data.frame(rbind(df_gp_CB_SegRT,df_gp_SC_SegRT))
df_gp_SegRT$Group <- c(rep("Blind",16),rep("Sighted",16))


df_gp_CB_SegRT$s_words <- factor(df_gp_CB_SegRT$s_words, levels = c("While", "the man","hunted","the ph
df_gp_SC_SegRT$s_words <- factor(df_gp_SC_SegRT$s_words, levels = c("While", "the man","hunted","the ph
```

```
df_gp_SegRT$s_words <- factor(df_gp_SegRT$s_words, levels = c("While", "the man","hunted","the pheasant
```

Plotting GP vs NG Segment RT

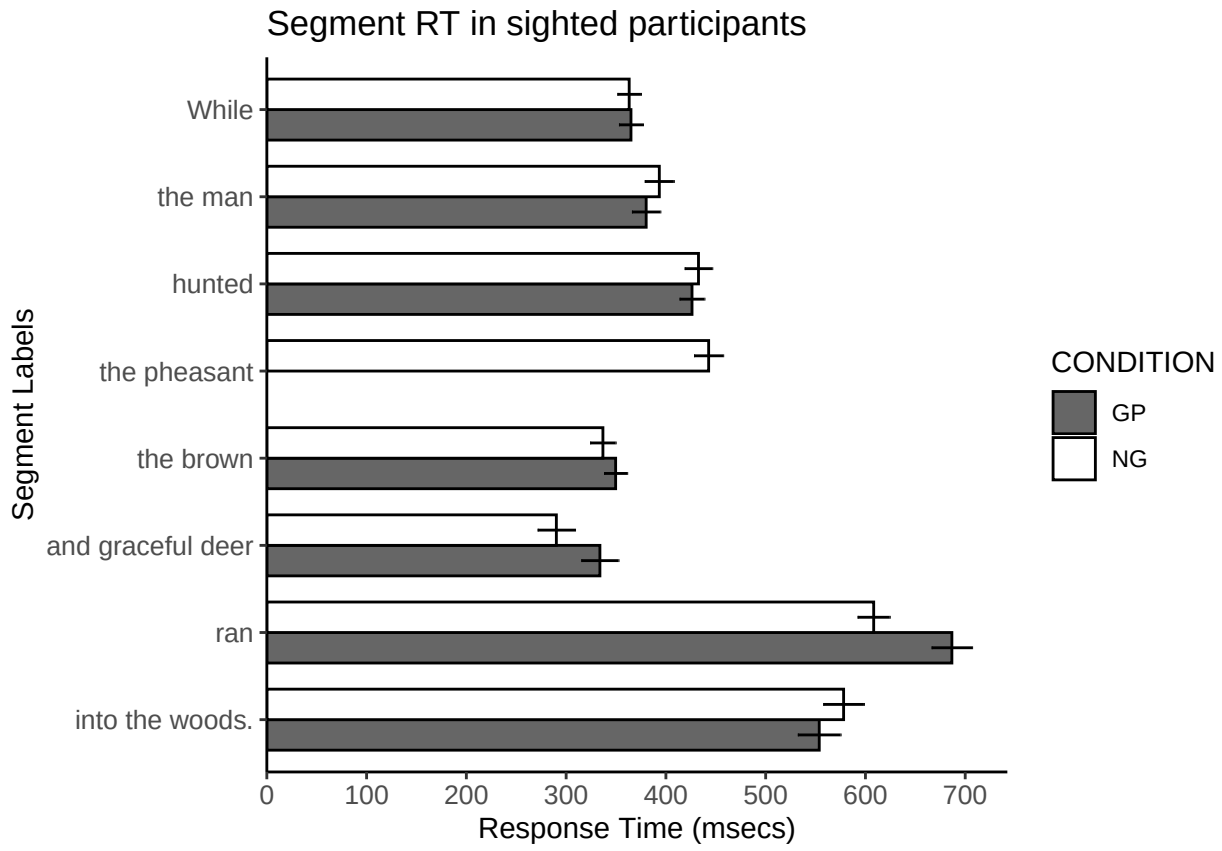
```
ggplot(data = df_gp_CB_SegRT, aes(x = avg, y = fct_rev(s_words), fill = CONDITION, xmin=avg-se, xmax=avg
  geom_bar(width = 0.7, position="dodge",stat = "identity", color = "black") +
  geom_errorbar(width = 0, position = position_dodge(0.7)) +
  scale_fill_grey(start = 0.4, end = 1) +
  scale_x_continuous(breaks = seq(0, 800, by = 100), expand = expansion(mult = c(0, 0.05))) +
  theme_bw()+
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  ylab("Segment Labels") +
  xlab("Response Time (msecs)") +
  ggtitle("Segment RT in blind participants")
```

```
## Warning: Removed 1 rows containing missing values (`geom_bar()`).
```



```
ggplot(data = df_gp_SC_SegRT, aes(x = avg, y = fct_rev(s_words), fill = CONDITION, xmin=avg-se, xmax=avg
  geom_bar(width = 0.7, position="dodge",stat = "identity", color = "black") +
  geom_errorbar(width = 0, position = position_dodge(0.7)) +
  scale_fill_grey(start = 0.4, end = 1) +
  scale_x_continuous(breaks = seq(0, 800, by = 100), expand = expansion(mult = c(0, 0.05))) +
  theme_bw()+
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  ylab("Segment Labels") +
  xlab("Response Time (msecs)") +
  ggtitle("Segment RT in sighted participants")
```

```
## Warning: Removed 1 rows containing missing values (‘geom_bar()‘).
```



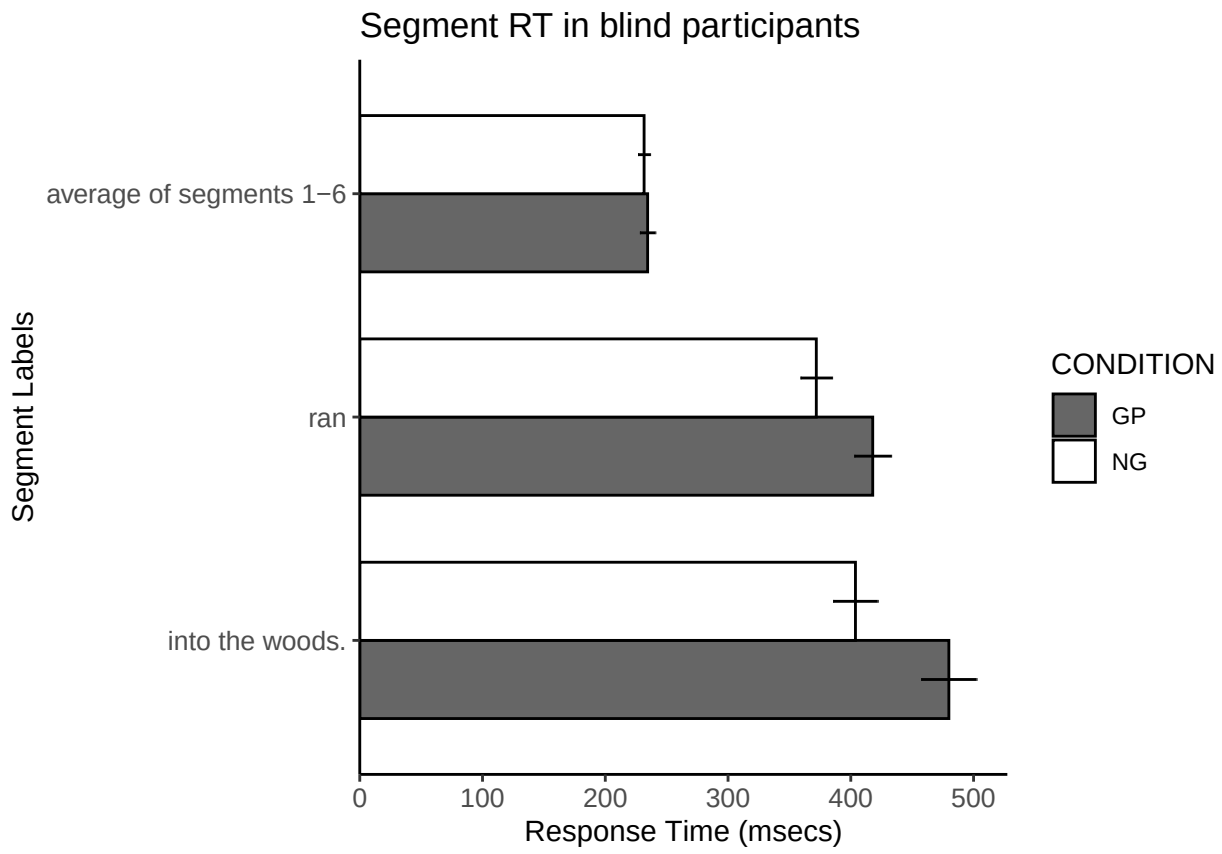
Average non-critical segments

```
df_gp_SPL_CB_avg <- data.frame(
  s_label <- c(1:3),
  s_words <- c("average of segments 1-6", "ran", "into the woods."),
  CONDITION <- c(rep("GP",3),rep("NG",3)),
  avg <- c(
```

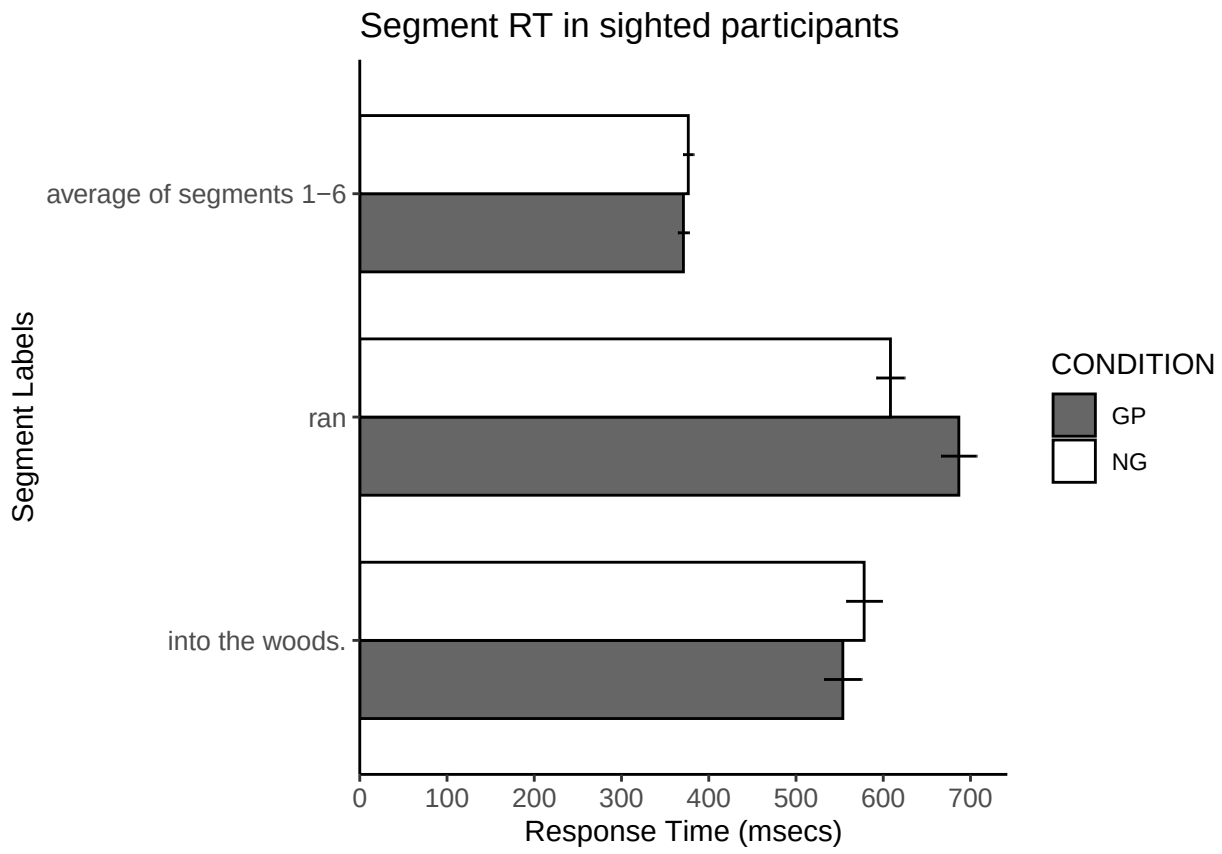
```
    mean(df_gp_CB_SegRT[c(1,2,3,5,6),]$avg),
    mean(subset(gp_CB_S, Seg == 'seg06')$RT),
    mean(subset(gp_CB_S, Seg == 'seg07')$RT),

    mean(df_gp_CB_SegRT[c(9,10,11,12,13,14),]$avg),
    mean(subset(ng_CB_S, Seg == 'seg07')$RT),
    mean(subset(ng_CB_S, Seg == 'seg08')$RT)
  ),
  se <- c(
    std.error(subset(gp_CB_S, !(Seg %in% c("seg06",'seg07')))$RT),
    std.error(subset(gp_CB_S, Seg == 'seg06')$RT),
    std.error(subset(gp_CB_S, Seg == 'seg07')$RT),

    std.error(subset(ng_CB_S, !(Seg %in% c("seg07",'seg08')))$RT),
    std.error(subset(ng_CB_S, Seg == 'seg07')$RT),
    std.error(subset(ng_CB_S, Seg == 'seg08')$RT)
  )
)
colnames(df_gp_SPL_CB_avg)<-(c("s_label", "s_words","CONDITION","avg","se"))


df_gp_SPL_SC_avg <- data.frame(
  s_label <- c(1:3),
  s_words <- c("average of segments 1-6", "ran", "into the woods."),
  CONDITION <- c(rep("GP",3),rep("NG",3)),
  avg <- c(
    mean(df_gp_SC_SegRT[c(1,2,3,5,6),]$avg),
    mean(subset(gp_SC_S, Seg == 'seg06')$RT),
    mean(subset(gp_SC_S, Seg == 'seg07')$RT),

    mean(df_gp_SC_SegRT[c(9,10,11,12,13,14),]$avg),
    mean(subset(ng_SC_S, Seg == 'seg07')$RT),
    mean(subset(ng_SC_S, Seg == 'seg08')$RT)
  ),
  se <- c(
    std.error(subset(gp_SC_S, !(Seg %in% c("seg06",'seg07')))$RT),
    std.error(subset(gp_SC_S, Seg == 'seg06')$RT),
    std.error(subset(gp_SC_S, Seg == 'seg07')$RT),

    std.error(subset(ng_SC_S, !(Seg %in% c("seg07",'seg08')))$RT),
    std.error(subset(ng_SC_S, Seg == 'seg07')$RT),
    std.error(subset(ng_SC_S, Seg == 'seg08')$RT)
  )
)
colnames(df_gp_SPL_SC_avg)<-(c("s_label", "s_words","CONDITION","avg","se"))

df_gp_SegRT_avg <- data.frame(rbind(df_gp_SPL_CB_avg,df_gp_SPL_SC_avg))
df_gp_SegRT_avg$Group <- c(rep("Blind",6),rep("Sighted",6))


df_gp_SPL_CB_avg$s_words <- factor(df_gp_SPL_CB_avg$s_words, levels = c("average of segments 1-6", "ran
df_gp_SPL_SC_avg$s_words <- factor(df_gp_SPL_SC_avg$s_words, levels = c("average of segments 1-6", "ran
df_gp_SPL_SC_avg$s_words <- factor(df_gp_SPL_SC_avg$s_words, levels = c("average of segments 1-6", "ran
```

```
#creating dfs for violin plot
cb_gp_s_temp = subset(final_SPL_CB_S, (CONDITION == 'GP' & !Seg %in% c('seg06','seg07')) | (CONDITION ==
cb_gp_s_temp$Group = 'Blind'
cb_gp_s_temp$Seg = 'Avg'

sc_gp_s_temp = subset(final_SPL_CB_S, (CONDITION == 'GP' & !Seg %in% c('seg06','seg07')) | (CONDITION ==
sc_gp_s_temp$Group = 'Sighted'
sc_gp_s_temp$Seg = 'Avg'

cb_gp_s_temp1 = subset(final_SPL_CB_S, (CONDITION == 'GP' & Seg %in% c('seg06','seg07')) | (CONDITION ==
cb_gp_s_temp1$Group = 'Blind'
cb_gp_s_temp1$Seg = 'Avg_crit'
sc_gp_s_temp1 = subset(final_SPL_SC_S, (CONDITION == 'GP' & Seg %in% c('seg06','seg07')) | (CONDITION ==
sc_gp_s_temp1$Group = 'Sighted'
sc_gp_s_temp1$Seg = 'Avg_crit'

SPL_GP_noncrit_violin <- data.frame(rbind(cb_gp_s_temp,sc_gp_s_temp))

SPL_GP_crit_violin <- data.frame(rbind(cb_gp_s_temp1,sc_gp_s_temp1))
```

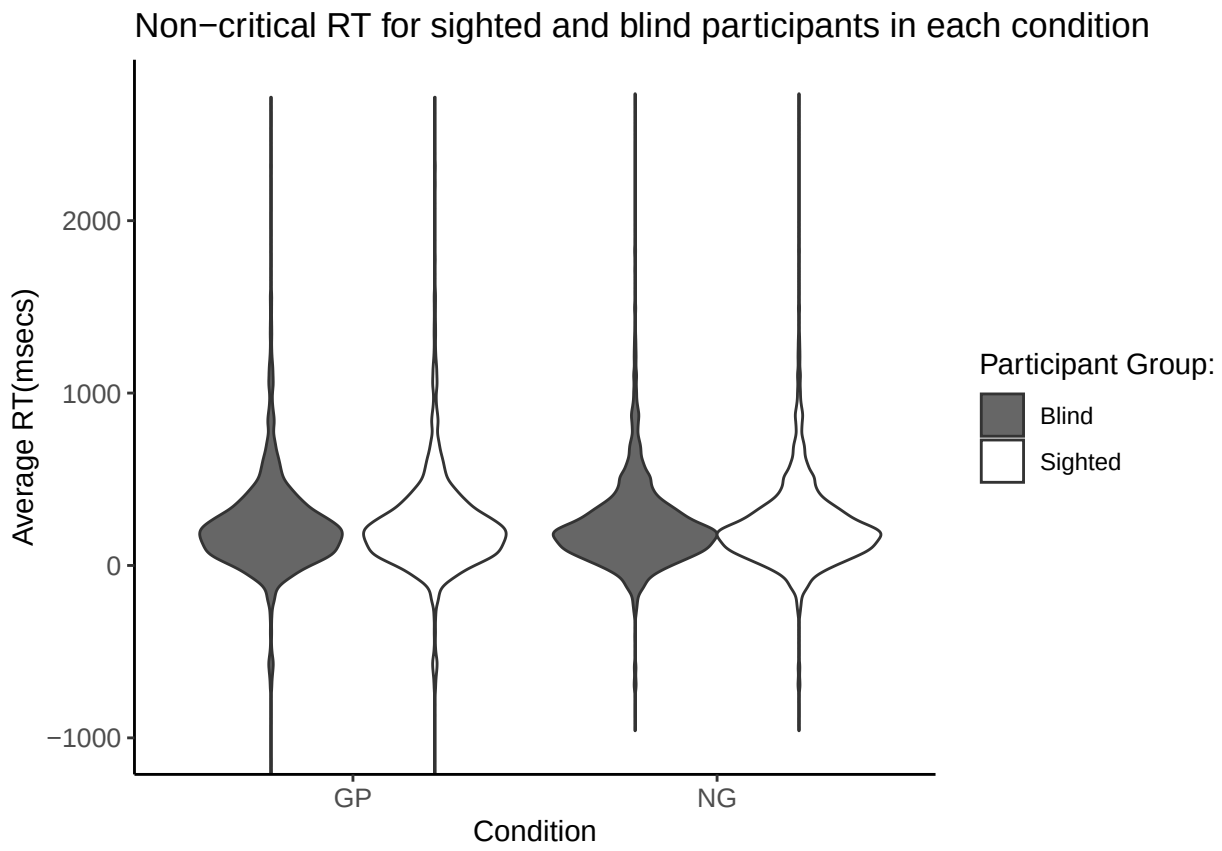
```
ggplot(data = df_gp_SPL_CB_avg, aes(x = avg, y = fct_rev(s_words), fill = CONDITION, xmin=avg-se, xmax=a
  geom_bar(width = 0.7, position="dodge",stat = "identity", color = "black") +
  geom_errorbar(width = 0, position = position_dodge(0.7)) +
  scale_fill_grey(start = 0.4, end = 1) +
  scale_x_continuous(breaks = seq(0, 800, by = 100), expand = expansion(mult = c(0, 0.05))) +
  theme_bw()+
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  ylab("Segment Labels") +
  xlab("Response Time (msecs)") +
  ggtitle("Segment RT in blind participants")
```



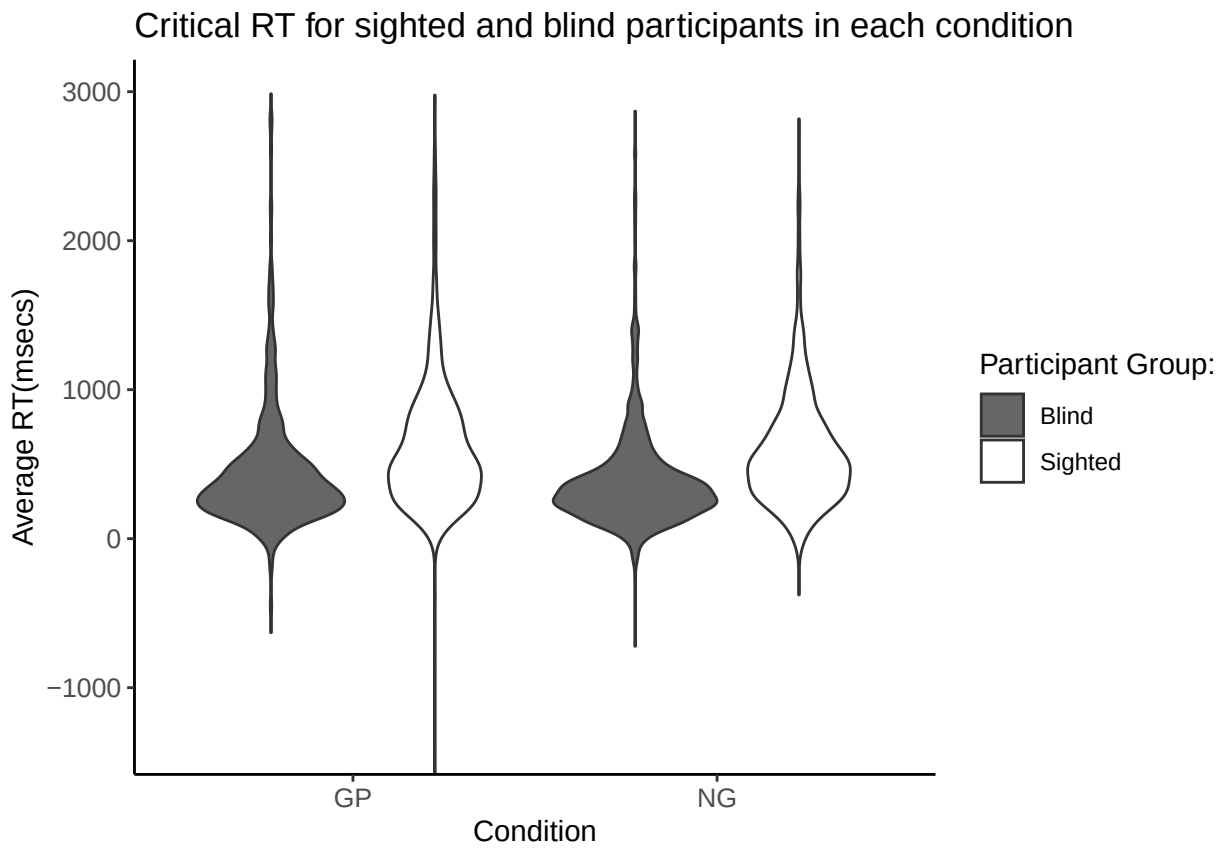

```
ggplot(data = df_gp_SPL_SC_avg, aes(x = avg, y = fct_rev(s_words), fill = CONDITION, xmin=avg-se, xmax=a
  geom_bar(width = 0.7, position="dodge",stat = "identity", color = "black") +
  geom_errorbar(width = 0, position = position_dodge(0.7)) +
  scale_fill_grey(start = 0.4, end = 1) +
  scale_x_continuous(breaks = seq(0, 800, by = 100), expand = expansion(mult = c(0, 0.05))) +
  theme_bw()+
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  ylab("Segment Labels") +
  xlab("Response Time (msecs)") +
  ggtitle("Segment RT in sighted participants")
```




```
ggplot(data = SPL_GP_noncrit_violin, aes(x = CONDITION, y = RT, fill=Group))+
  geom_violin()+
  scale_fill_grey(start = 0.4, end = 1)+
  theme_bw()+
  scale_y_continuous(breaks = seq(-10000, 15000, by = 1000), expand = expansion(mult = c(0, 0.05))) +
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  xlab("Condition") +
  ylab("Average RT(msecs)") +
  labs(fill = "Participant Group:") +
  ggtitle("Non-critical RT for sighted and blind participants in each condition")
```




```
ggplot(data = SPL_GP_crit_violin, aes(x = CONDITION, y = RT, fill=Group))+
  geom_violin()+
  scale_fill_grey(start = 0.4, end = 1)+
  theme_bw()+
  scale_y_continuous(breaks = seq(-10000, 15000, by = 1000), expand = expansion(mult = c(0, 0.05))) +
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  xlab("Condition") +
  ylab("Average RT(msecs)") +
  labs(fill = "Participant Group:") +
  ggtitle("Critical RT for sighted and blind participants in each condition")
```



Plotting MV vs NMV Segment RT

```
library(plotrix)
mv_CB_S <- subset(final_SPL_CB_S, CONDITION == 'MV')
nmv_CB_S <- subset(final_SPL_CB_S, CONDITION == 'NMV')


df_mv_SPL_CB_SegRT <- data.frame(
  s_label <- c(1:8),
  s_words <- factor(c("segment 1", "segment 2","segment 3","segment 4","segment 5","always", "has to be
  CONDITION <- c(rep("MV",8),rep("NMV",8)),
  avg <- c(
    mean(subset(mv_CB_S, Seg == 'seg01')$RT),
    mean(subset(mv_CB_S, Seg == 'seg02')$RT),
    mean(subset(mv_CB_S, Seg == 'seg03')$RT),
    mean(subset(mv_CB_S, Seg == 'seg04')$RT),
    mean(subset(mv_CB_S, Seg == 'seg05')$RT),
    mean(subset(mv_CB_S, Seg == 'seg06')$RT),
    mean(subset(mv_CB_S, Seg == 'seg07')$RT),
    mean(subset(mv_CB_S, Seg == 'seg08')$RT),

    mean(subset(nmv_CB_S, Seg == 'seg01')$RT),
    mean(subset(nmv_CB_S, Seg == 'seg02')$RT),
    mean(subset(nmv_CB_S, Seg == 'seg03')$RT),
    mean(subset(nmv_CB_S, Seg == 'seg04')$RT),
    mean(subset(nmv_CB_S, Seg == 'seg05')$RT),
```

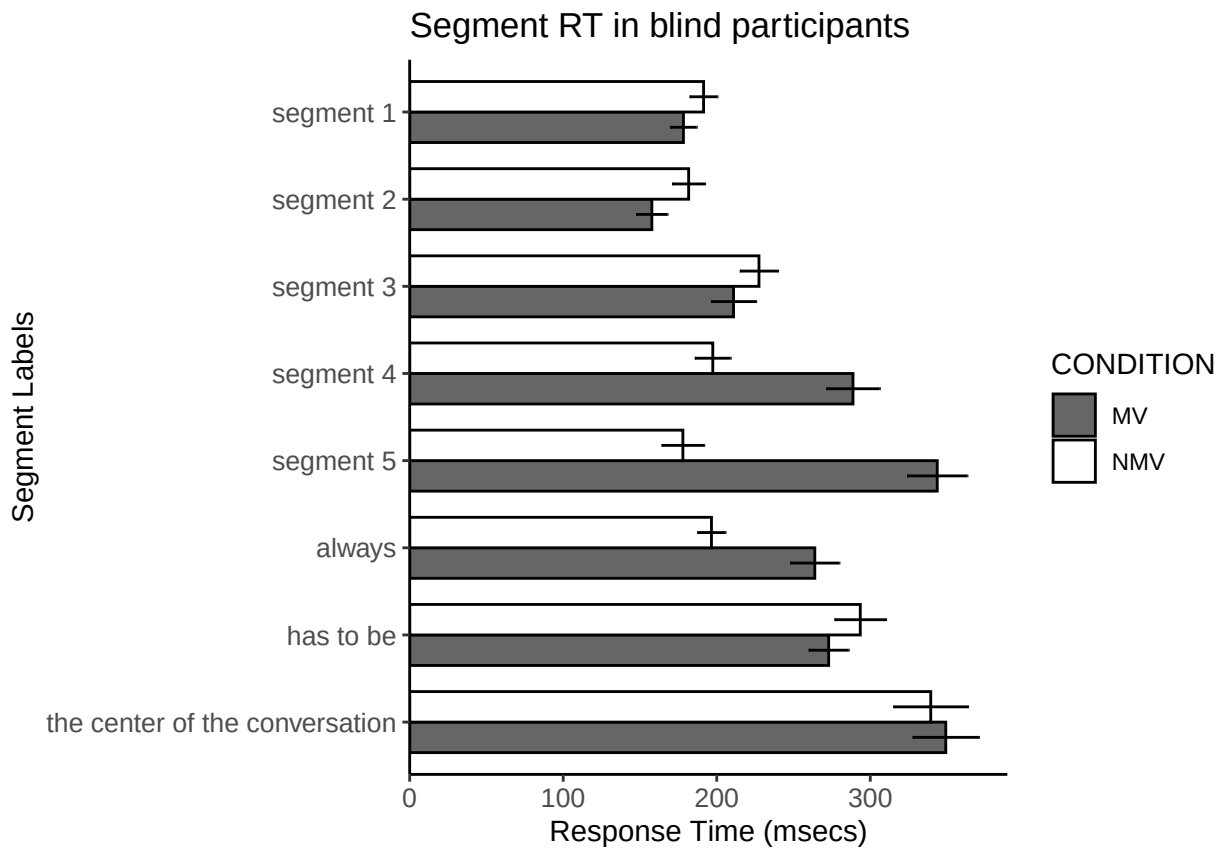
```
    mean(subset(nmv_CB_S, Seg == 'seg06')$RT),
    mean(subset(nmv_CB_S, Seg == 'seg07')$RT),
    mean(subset(nmv_CB_S, Seg == 'seg08')$RT)
  ),
  se <- c(
    std.error(subset(mv_CB_S, Seg == 'seg01')$RT),
    std.error(subset(mv_CB_S, Seg == 'seg02')$RT),
    std.error(subset(mv_CB_S, Seg == 'seg03')$RT),
    std.error(subset(mv_CB_S, Seg == 'seg04')$RT),
    std.error(subset(mv_CB_S, Seg == 'seg05')$RT),
    std.error(subset(mv_CB_S, Seg == 'seg06')$RT),
    std.error(subset(mv_CB_S, Seg == 'seg07')$RT),
    std.error(subset(mv_CB_S, Seg == 'seg08')$RT),

    std.error(subset(nmv_CB_S, Seg == 'seg01')$RT),
    std.error(subset(nmv_CB_S, Seg == 'seg02')$RT),
    std.error(subset(nmv_CB_S, Seg == 'seg03')$RT),
    std.error(subset(nmv_CB_S, Seg == 'seg04')$RT),
    std.error(subset(nmv_CB_S, Seg == 'seg05')$RT),
    std.error(subset(nmv_CB_S, Seg == 'seg06')$RT),
    std.error(subset(nmv_CB_S, Seg == 'seg07')$RT),
    std.error(subset(nmv_CB_S, Seg == 'seg08')$RT)
  )
)
colnames(df_mv_SPL_CB_SegRT)<-(c("s_label", "s_words","CONDITION","avg","se"))


mv_SC_S <- subset(final_SPL_SC_S, CONDITION == 'MV')
nmv_SC_S <- subset(final_SPL_SC_S, CONDITION == 'NMV')

df_mv_SPL_SC_SegRT <- data.frame(
  s_label <- c(1:8),
  s_words <- factor(c("segment 1", "segment 2","segment 3","segment 4","segment 5","always", "has to be
  CONDITION <- c(rep("MV",8),rep("NMV",8)),
  avg <- c(
    mean(subset(mv_SC_S, Seg == 'seg01')$RT),
    mean(subset(mv_SC_S, Seg == 'seg02')$RT),
    mean(subset(mv_SC_S, Seg == 'seg03')$RT),
    mean(subset(mv_SC_S, Seg == 'seg04')$RT),
    mean(subset(mv_SC_S, Seg == 'seg05')$RT),
    mean(subset(mv_SC_S, Seg == 'seg06')$RT),
    mean(subset(mv_SC_S, Seg == 'seg07')$RT),
    mean(subset(mv_SC_S, Seg == 'seg08')$RT),

    mean(subset(nmv_SC_S, Seg == 'seg01')$RT),
    mean(subset(nmv_SC_S, Seg == 'seg02')$RT),
    mean(subset(nmv_SC_S, Seg == 'seg03')$RT),
    mean(subset(nmv_SC_S, Seg == 'seg04')$RT),
    mean(subset(nmv_SC_S, Seg == 'seg05')$RT),
    mean(subset(nmv_SC_S, Seg == 'seg06')$RT),
    mean(subset(nmv_SC_S, Seg == 'seg07')$RT),
    mean(subset(nmv_SC_S, Seg == 'seg08')$RT)
  ),
```

```
  se <- c(
    std.error(subset(mv_SC_S, Seg == 'seg01')$RT),
    std.error(subset(mv_SC_S, Seg == 'seg02')$RT),
    std.error(subset(mv_SC_S, Seg == 'seg03')$RT),
    std.error(subset(mv_SC_S, Seg == 'seg04')$RT),
    std.error(subset(mv_SC_S, Seg == 'seg05')$RT),
    std.error(subset(mv_SC_S, Seg == 'seg06')$RT),
    std.error(subset(mv_SC_S, Seg == 'seg07')$RT),
    std.error(subset(mv_SC_S, Seg == 'seg08')$RT),

    std.error(subset(nmv_SC_S, Seg == 'seg01')$RT),
    std.error(subset(nmv_SC_S, Seg == 'seg02')$RT),
    std.error(subset(nmv_SC_S, Seg == 'seg03')$RT),
    std.error(subset(nmv_SC_S, Seg == 'seg04')$RT),
    std.error(subset(nmv_SC_S, Seg == 'seg05')$RT),
    std.error(subset(nmv_SC_S, Seg == 'seg06')$RT),
    std.error(subset(nmv_SC_S, Seg == 'seg07')$RT),
    std.error(subset(nmv_SC_S, Seg == 'seg08')$RT)
  )
)
colnames(df_mv_SPL_SC_SegRT)<-(c("s_label", "s_words","CONDITION","avg","se"))

df_mv_SPL_SegRT <- data.frame(rbind(df_mv_SPL_CB_SegRT,df_mv_SPL_SC_SegRT))
df_mv_SPL_SegRT$Group <- c(rep("Blind",16),rep("Sighted",16))

df_mv_SPL_CB_SegRT$s_words <- factor(df_mv_SPL_CB_SegRT$s_words, levels = c("segment 1", "segment 2","se
df_mv_SPL_SC_SegRT$s_words <- factor(df_mv_SPL_SC_SegRT$s_words, levels = c("segment 1", "segment 2","se
df_mv_SPL_SegRT$s_words <- factor(df_mv_SPL_SegRT$s_words, levels = c("segment 1", "segment 2","segment



## Averaging non critical segments

df_mv_SPL_CB_avg <- data.frame(
  s_label <- c(1:4),
  s_words <- c("average of segments 1-5", "always", "has to be","the center of the conversation."),
  CONDITION <- c(rep("MV",4),rep("NMV",4)),
  avg <- c(
    mean(df_mv_SPL_CB_SegRT[c(1,2,3,4,5),]$avg),
    mean(subset(mv_CB_S, Seg == 'seg06')$RT),
    mean(subset(mv_CB_S, Seg == 'seg07')$RT),
    mean(subset(mv_CB_S, Seg == 'seg07')$RT),

    mean(df_mv_SPL_CB_SegRT[c(9,10,11,12,13),]$avg),
    mean(subset(nmv_CB_S, Seg == 'seg06')$RT),
    mean(subset(nmv_CB_S, Seg == 'seg07')$RT),
    mean(subset(nmv_CB_S, Seg == 'seg08')$RT)
  ),
  se <- c(
    std.error(subset(mv_CB_S, !(Seg %in% c("seg06",'seg07','seg08')))$RT),
    std.error(subset(mv_CB_S, Seg == 'seg06')$RT),
    std.error(subset(mv_CB_S, Seg == 'seg07')$RT),
    std.error(subset(mv_CB_S, Seg == 'seg08')$RT),
```

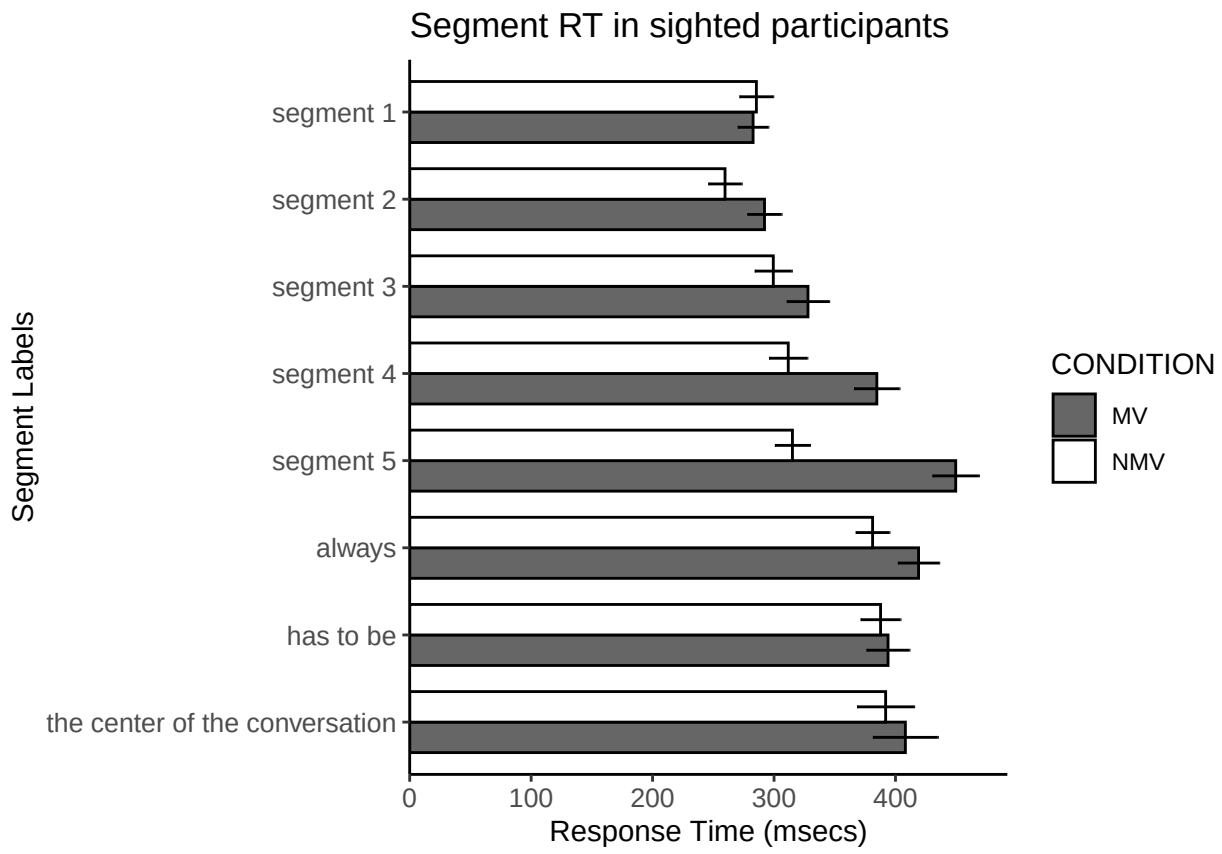
```
    std.error(subset(nmv_CB_S, !(Seg %in% c('seg06',"seg07",'seg08')))$RT),
    std.error(subset(nmv_CB_S, Seg == 'seg06')$RT),
    std.error(subset(nmv_CB_S, Seg == 'seg07')$RT),
    std.error(subset(nmv_CB_S, Seg == 'seg08')$RT)
  )
)
colnames(df_mv_SPL_CB_avg)<-(c("s_label", "s_words","CONDITION","avg","se"))


df_mv_SPL_SC_avg <- data.frame(
  s_label <- c(1:4),
  s_words <- c("average of segments 1-5", "always", "has to be","the center of the conversation."),
  CONDITION <- c(rep("GP",4),rep("NG",4)),
  avg <- c(
    mean(df_mv_SPL_SC_SegRT[c(1,2,3,4,5),]$avg),
    mean(subset(mv_SC_S, Seg == 'seg06')$RT),
    mean(subset(mv_SC_S, Seg == 'seg07')$RT),
    mean(subset(mv_SC_S, Seg == 'seg08')$RT),

    mean(df_mv_SPL_SC_SegRT[c(9,10,11,12,13),]$avg),
    mean(subset(nmv_SC_S, Seg == 'seg06')$RT),
    mean(subset(nmv_SC_S, Seg == 'seg07')$RT),
    mean(subset(nmv_SC_S, Seg == 'seg08')$RT)
  ),
  se <- c(
    std.error(subset(mv_SC_S, !(Seg %in% c("seg06",'seg07','seg08')))$RT),
    std.error(subset(mv_SC_S, Seg == 'seg06')$RT),
    std.error(subset(mv_SC_S, Seg == 'seg07')$RT),
    std.error(subset(mv_SC_S, Seg == 'seg08')$RT),

    std.error(subset(nmv_SC_S, !(Seg %in% c('seg06',"seg07",'seg08')))$RT),
    std.error(subset(nmv_SC_S, Seg == 'seg06')$RT),
    std.error(subset(nmv_SC_S, Seg == 'seg07')$RT),
    std.error(subset(nmv_SC_S, Seg == 'seg08')$RT)
  )
)
colnames(df_mv_SPL_SC_avg)<-(c("s_label", "s_words","CONDITION","avg","se"))

df_mv_SegRT_avg <- data.frame(rbind(df_mv_SPL_CB_avg,df_mv_SPL_SC_avg))
df_mv_SegRT_avg$Group <- c(rep("Blind",8),rep("Sighted",8))


df_mv_SPL_CB_avg$s_words <- factor(df_mv_SPL_CB_avg$s_words, levels = c("average of segments 1-5", "alwa
df_mv_SPL_SC_avg$s_words <- factor(df_mv_SPL_SC_avg$s_words, levels = c("average of segments 1-5", "alwa
df_mv_SPL_SC_avg$s_words <- factor(df_mv_SPL_SC_avg$s_words, levels = c("average of segments 1-5", "alwa


#creating dfs for violin plot

cb_mv_s_temp = subset(final_SPL_CB_S, CONDITION %in% c('MV','NMV') & !Seg %in% c('seg06','seg07','seg08
cb_mv_s_temp$Group = 'Blind'
cb_mv_s_temp$Seg = 'Avg'
sc_mv_s_temp = subset(final_SPL_SC_S, CONDITION %in% c('MV','NMV') & !Seg %in% c('seg06','seg07','seg08
```

```
sc_mv_s_temp$Group = 'Sighted'
sc_mv_s_temp$Seg = 'Avg'

cb_mv_s_temp1 = subset(final_SPL_CB_S, CONDITION %in% c('MV','NMV') & Seg %in% c('seg06','seg07','seg08
cb_mv_s_temp1$Group = 'Blind'
cb_mv_s_temp1$Seg = 'Avg_crit'
sc_mv_s_temp1 = subset(final_SPL_SC_S, CONDITION %in% c('MV','NMV') & Seg %in% c('seg06','seg07','seg08
sc_mv_s_temp1$Group = 'Sighted'
sc_mv_s_temp1$Seg = 'Avg_crit'

SPL_MV_noncrit_violin <- data.frame(rbind(cb_mv_s_temp,sc_mv_s_temp))

SPL_MV_crit_violin <- data.frame(rbind(cb_mv_s_temp1,sc_mv_s_temp1))
```

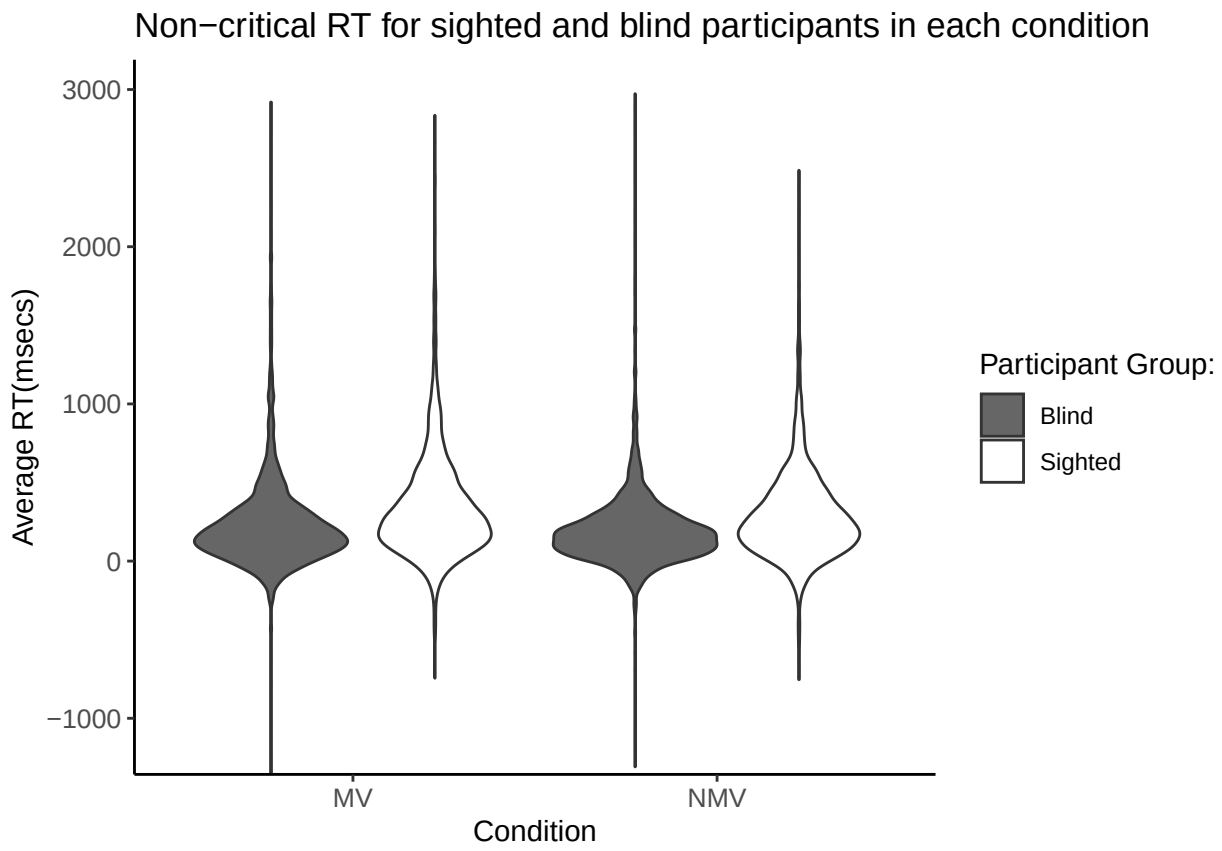
```
ggplot(data = df_mv_SPL_CB_SegRT, aes(x = avg, y = fct_rev(s_words), fill = CONDITION, xmin=avg-se, xma
  geom_bar(width = 0.7, position="dodge",stat = "identity", color = "black") +
  geom_errorbar(width = 0, position = position_dodge(0.7)) +
  scale_fill_grey(start = 0.4, end = 1) +
  scale_x_continuous(breaks = seq(0, 500, by = 100), expand = expansion(mult = c(0, 0.05))) +
  theme_bw()+
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  ylab("Segment Labels") +
  xlab("Response Time (msecs)")+
  ggtitle("Segment RT in blind participants")
```



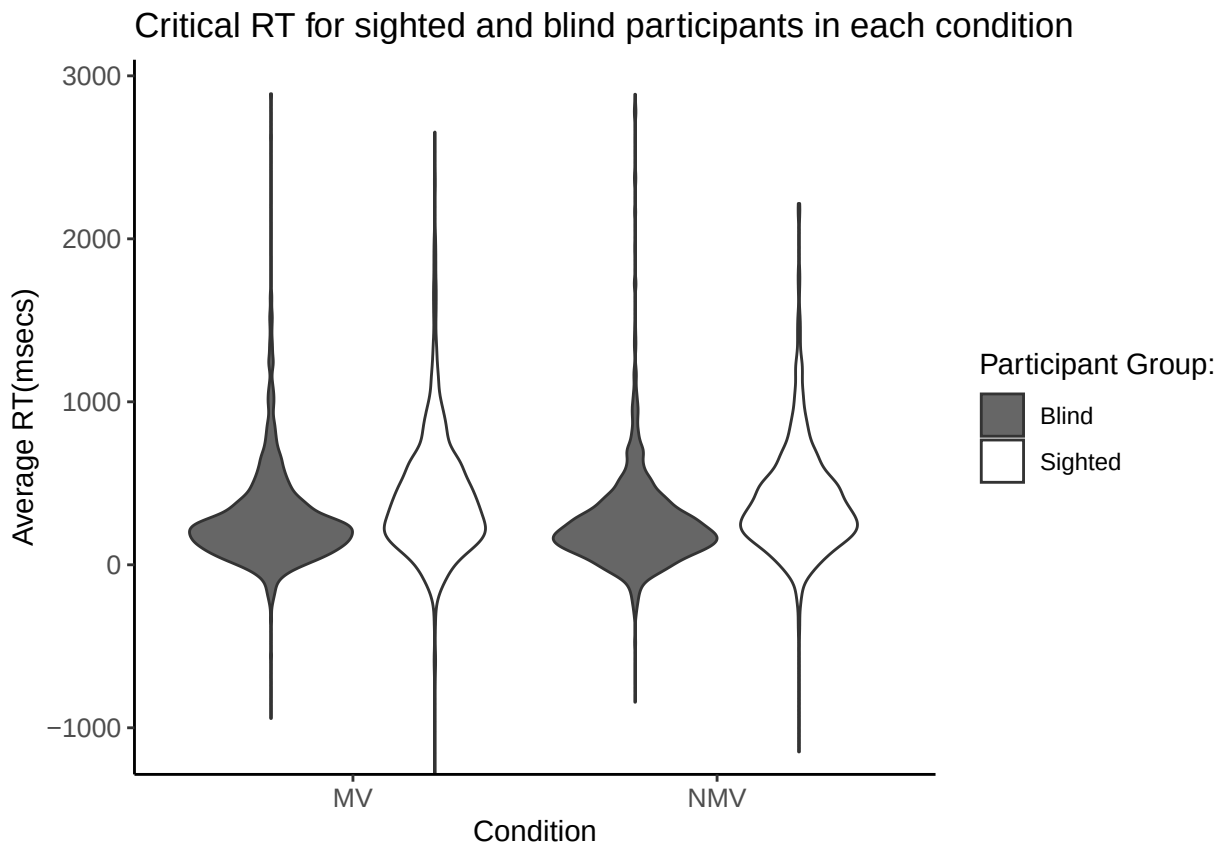

```
ggplot(data = df_mv_SPL_SC_SegRT, aes(x = avg, y = fct_rev(s_words), fill = CONDITION, xmin=avg-se, xma
  geom_bar(width = 0.7, position="dodge",stat = "identity", color = "black") +
  geom_errorbar(width = 0, position = position_dodge(0.7)) +
  scale_fill_grey(start = 0.4, end = 1) +
  scale_x_continuous(breaks = seq(0, 700, by = 100), expand = expansion(mult = c(0, 0.05))) +
  theme_bw()+
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  ylab("Segment Labels") +
  xlab("Response Time (msecs)")+
  ggtitle("Segment RT in sighted participants")
```




```
ggplot(data = SPL_MV_noncrit_violin, aes(x = CONDITION, y = RT, fill=Group))+
  geom_violin()+
  scale_fill_grey(start = 0.4, end = 1)+
  theme_bw()+
  scale_y_continuous(breaks = seq(-10000, 15000, by = 1000), expand = expansion(mult = c(0, 0.05))) +
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  xlab("Condition") +
  ylab("Average RT(msecs)") +
  labs(fill = "Participant Group:") +
  ggtitle("Non-critical RT for sighted and blind participants in each condition")
```




```
ggplot(data = SPL_MV_crit_violin, aes(x = CONDITION, y = RT, fill=Group))+
  geom_violin()+
  scale_fill_grey(start = 0.4, end = 1)+
  theme_bw()+
  scale_y_continuous(breaks = seq(-10000, 15000, by = 1000), expand = expansion(mult = c(0, 0.05))) +
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  xlab("Condition") +
  ylab("Average RT(msecs)") +
  labs(fill = "Participant Group:") +
  ggtitle("Critical RT for sighted and blind participants in each condition")
```




```
#check segment average per condition across all segments
gp_CB_avg_all_seg <- mean(subset(final_SPL_CB_S, CONDITION == 'GP')$RT)
gp_SC_avg_all_seg <- mean(subset(final_SPL_SC_S, CONDITION == 'GP')$RT)
print("GP Blind group average: ")
```

```
## [1] "GP Blind group average: "
```

```
print(gp_CB_avg_all_seg)
```

```
## [1] 295.1621
```

```
print("GP Sighted group average: ")
```

```
## [1] "GP Sighted group average: "
```

```
print(gp_SC_avg_all_seg)
```

```
## [1] 441.7441
```

```
ng_CB_avg_all_seg <- mean(subset(final_SPL_CB_S, CONDITION == 'NG')$RT)
ng_SC_avg_all_seg <- mean(subset(final_SPL_SC_S, CONDITION == 'NG')$RT)
print("NG Blind group average: ")
```

```
## [1] "NG Blind group average: "
```

```
print(ng_CB_avg_all_seg)
```

```
## [1] 270.5826
```

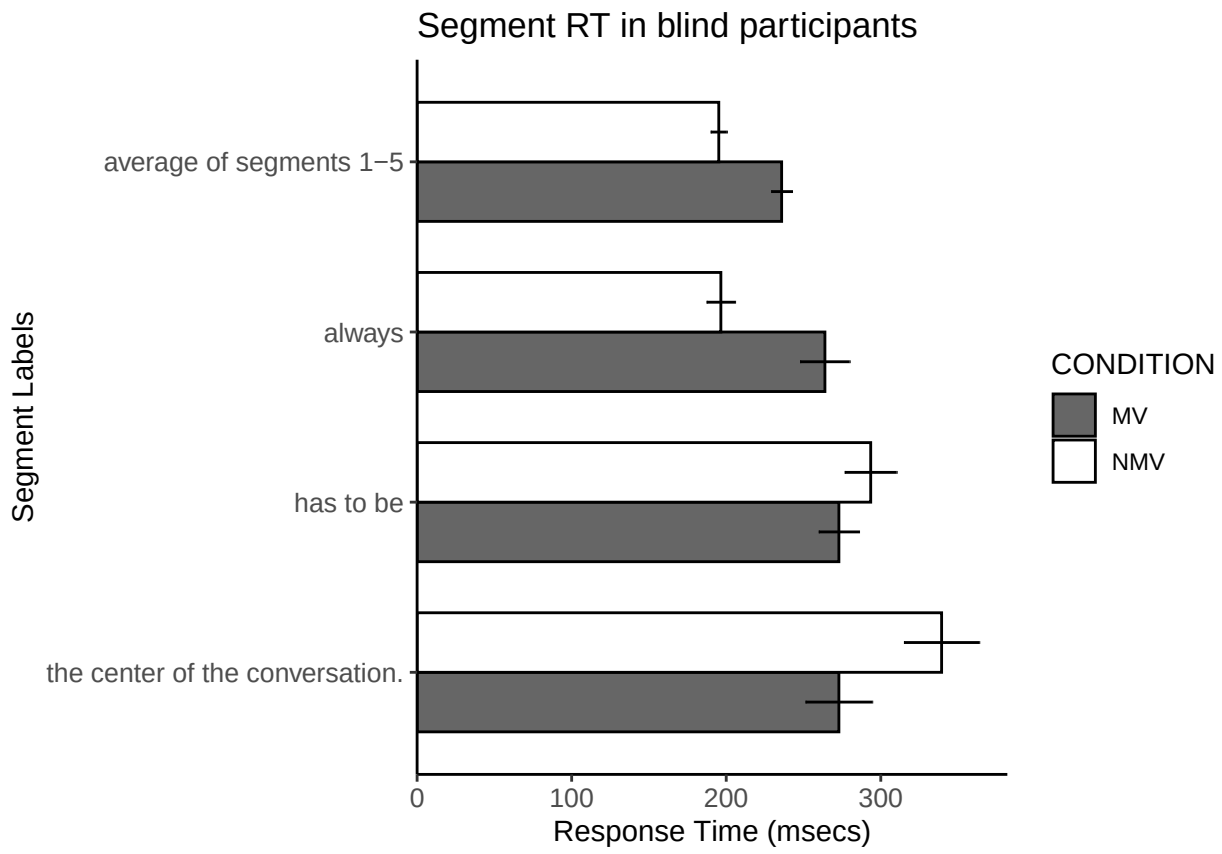
```
print("NG Sighted group average: ")
```

```
## [1] "NG Sighted group average: "
```

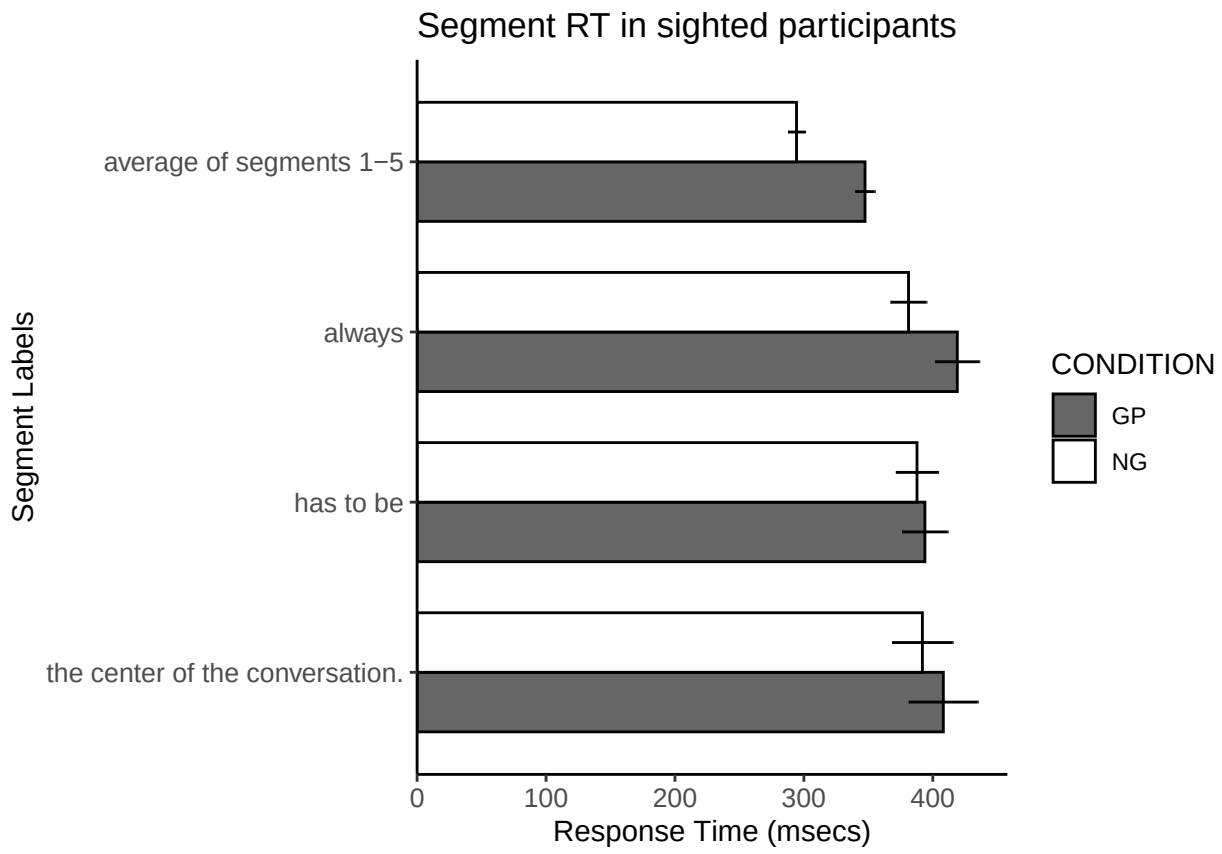
```
print(ng_SC_avg_all_seg)
```

```
## [1] 430.4226
```

```
ggplot(data = df_mv_SPL_CB_avg, aes(x = avg, y = fct_rev(s_words), fill = CONDITION, xmin=avg-se, xmax=a
  geom_bar(width = 0.7, position="dodge",stat = "identity", color = "black") +
  geom_errorbar(width = 0, position = position_dodge(0.7)) +
  scale_fill_grey(start = 0.4, end = 1) +
  scale_x_continuous(breaks = seq(0, 800, by = 100), expand = expansion(mult = c(0, 0.05))) +
  theme_bw()+
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  ylab("Segment Labels") +
  xlab("Response Time (msecs)") +
  ggtitle("Segment RT in blind participants")
```




```
ggplot(data = df_mv_SPL_SC_avg, aes(x = avg, y = fct_rev(s_words), fill = CONDITION, xmin=avg-se, xmax=a
  geom_bar(width = 0.7, position="dodge",stat = "identity", color = "black") +
  geom_errorbar(width = 0, position = position_dodge(0.7)) +
  scale_fill_grey(start = 0.4, end = 1) +
  scale_x_continuous(breaks = seq(0, 800, by = 100), expand = expansion(mult = c(0, 0.05))) +
  theme_bw()+
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.text=element_text(size = 10)) +
  ylab("Segment Labels") +
  xlab("Response Time (msecs)") +
  ggtitle("Segment RT in sighted participants")
```




```
#critical segment within CONDITION across GROUP
gp_crit_CB_S <- subset(gp_CB_S[c("CONDITION", "Item", "ID","Correct", "Seg","RT", "RT+Duration")], Seg =
gp_crit_SC_S <- subset(gp_SC_S[c("CONDITION", "Item", "ID","Correct", "Seg","RT", "RT+Duration")], Seg =
t.test(gp_crit_CB_S$RT, gp_crit_SC_S$RT)
```

```
##
##  Welch Two Sample t-test
##
## data:  gp_crit_CB_S$RT and gp_crit_SC_S$RT
## t = -10.66, df = 851.03, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -318.2414 -219.2758
## sample estimates:
## mean of x mean of y
##  417.8724  686.6310
```

```
#Across Condition
ng_crit_CB_S <- subset(ng_CB_S[c("CONDITION", "Item", "ID","Correct", "Seg","RT", "RT+Duration")], Seg =
t.test(gp_crit_CB_S$RT, ng_crit_CB_S$RT)
```

```
##
##  Welch Two Sample t-test
##
## data:  gp_crit_CB_S$RT and ng_crit_CB_S$RT
```

```
## t = 2.3271, df = 936.06, p-value = 0.02017
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   7.20130 84.71789
## sample estimates:
## mean of x mean of y
##  417.8724  371.9128
```

```
ng_crit_SC_S <- subset(ng_SC_S[c("CONDITION", "Item", "ID","Correct", "Seg","RT", "RT+Duration")], Seg =
```

```
#posterior
gp_post_CB_S <- subset(gp_CB_S[c("CONDITION", "Item", "ID","Correct", "Seg","RT", "RT+Duration")], Seg =
gp_post_SC_S <- subset(gp_SC_S[c("CONDITION", "Item", "ID","Correct","Seg", "RT", "RT+Duration")], Seg =
t.test(gp_post_CB_S$RT, gp_post_SC_S$RT)
```

```
##
##  Welch Two Sample t-test
##
## data:  gp_post_CB_S$RT and gp_post_SC_S$RT
## t = -2.3737, df = 921.16, p-value = 0.01781
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -134.77834  -12.78091
## sample estimates:
## mean of x mean of y
##  479.8977  553.6773
```

```
#Across Condition
ng_post_CB_S <- subset(ng_CB_S[c("CONDITION", "Item", "ID","Correct","Seg", "RT", "RT+Duration")], Seg =
t.test(gp_post_CB_S$RT, ng_post_CB_S$RT)
```

```
##
##  Welch Two Sample t-test
##
## data:  gp_post_CB_S$RT and ng_post_CB_S$RT
## t = 2.6322, df = 905.18, p-value = 0.008628
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   19.35785 132.83075
## sample estimates:
## mean of x mean of y
##  479.8977  403.8034
```

```
ng_post_SC_S <- subset(ng_SC_S[c("CONDITION", "Item", "ID","Correct","Seg", "RT", "RT+Duration")], Seg =
```

```
gp_crit_CB_S$Group = 0
gp_crit_SC_S$Group = 1

ng_crit_CB_S$Group = 0
ng_crit_SC_S$Group = 1
```

```r
gp_crit_df <- data.frame(rbind(gp_crit_CB_S, gp_crit_SC_S, ng_crit_CB_S, ng_crit_SC_S))

gp_crit_df$ln_RT <- log(gp_crit_df$RT+2000)
#gp_crit_CB_S$ln_RT_Duration <- log(gp_crit_df$RT.Duration)


#posterior
gp_post_CB_S$Group = 0
gp_post_SC_S$Group = 1

ng_post_CB_S$Group = 0
ng_post_SC_S$Group = 1

gp_post_df <- data.frame(rbind(gp_post_CB_S, gp_post_SC_S, ng_post_CB_S, ng_post_SC_S))

gp_post_df$ln_RT <- log(gp_post_df$RT+2000)
```

TODO: violin plot for question reaction time TODO: run linear model (with and without log) for critical and posterior TODO: compare avg across RT in blind and sighted groups

```r
log_gp_crit_lm <- lmer(ln_RT ~ CONDITION * Group + (1|Item) + (1|ID), data = gp_crit_df, REML = FALSE,c
summary(log_gp_crit_lm)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: ln_RT ~ CONDITION * Group + (1 | Item) + (1 | ID)
##    Data: gp_crit_df
##
##      AIC      BIC   logLik deviance df.resid
##  -3347.2  -3308.4   1680.6  -3361.2     1870
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.2428 -0.5086 -0.1023  0.4132  5.8834
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  Item     (Intercept) 0.0007889 0.02809
##  ID       (Intercept) 0.0061922 0.07869
##  Residual             0.0084495 0.09192
## Number of obs: 1877, groups:  Item, 168; ID, 45
##
## Fixed effects:
##                     Estimate Std. Error t value
## (Intercept)         7.783843   0.017483 445.230
## CONDITIONNG        -0.017394   0.008534  -2.038
## Group               0.101293   0.024996   4.052
## CONDITIONNG:Group -0.007283   0.012160  -0.599
##
## Correlation of Fixed Effects:
##              (Intr) CONDITIONNG Group
## CONDITIONNG  -0.244
## Group        -0.699  0.171
## CONDITIONNG:  0.171 -0.702      -0.243
```

```r
log_gp_post_lm <- lmer(ln_RT ~ CONDITION * Group + (1|Item) + (1|ID), data = gp_post_df, REML = FALSE,c
summary(log_gp_post_lm)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: ln_RT ~ CONDITION * Group + (1 | Item) + (1 | ID)
##    Data: gp_post_df
##
##      AIC      BIC   logLik deviance df.resid
##  -2058.5  -2019.7   1036.2  -2072.5     1858
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -13.5872  -0.5312  -0.0975   0.3547   4.3444
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Item     (Intercept) 0.001289 0.03590
##  ID       (Intercept) 0.008650 0.09301
##  Residual             0.016949 0.13019
## Number of obs: 1865, groups:  Item, 168; ID, 45
##
## Fixed effects:
##                   Estimate Std. Error t value
## (Intercept)        7.80227    0.02105 370.737
## CONDITIONNG       -0.02920    0.01152  -2.535
## Group              0.02982    0.03008   0.991
## CONDITIONNG:Group  0.03936    0.01642   2.397
##
## Correlation of Fixed Effects:
##              (Intr) CONDITIONNG Group
## CONDITIONNG  -0.276
## Group        -0.700  0.193
## CONDITIONNG:  0.193 -0.702      -0.274
```