Temporal bisection is influenced by ensemble statistics of the stimulus set (Data Analysis Report)

Xiuna Zhu

12 Februar 2019

```
mDist <-  dat_all %>% dplyr::group_by(NSub, cond) %>%
  summarise(m_cDur = mean(curDur), n = dplyr::n(),
            sd_Dur = sd(curDur),
            gm_Dur = gm_mean(curDur),
            cDur_se = sd(curDur)/sqrt(n-1))
```

```
## `summarise()` regrouping output by 'NSub' (override with `.groups` argument)
```

```
mDist
```

```
## # A tibble: 90 x 7
## # Groups:   NSub [15]
##     NSub cond       m_cDur     n sd_Dur gm_Dur cDur_se
##    <int> <fct>       <dbl> <int>  <dbl>  <dbl>   <dbl>
##  1     1 PS           888.   336   401.   800.    21.9
##  2     1 NS          1112    336   402.  1018.    21.9
##  3     1 DF           800    448   347.   728.    16.4
##  4     1 AF          1200    448   347.  1138.    16.4
##  5     1 U-shaped     925    288   491.   781.    29.0
##  6     1 I T-shaped   925    288   175.   906.    10.3
##  7     2 PS           888.   336   401.   800.    21.9
##  8     2 NS          1112    336   402.  1018.    21.9
##  9     2 DF           800    448   347.   728.    16.4
## 10     2 AF          1200    448   347.  1138.    16.4
## # ... with 80 more rows
```

Experiment 1

positive skew condition (400, 504, 636, 800, 1008, 1270, and 1600 ms)

negative skew condition (400, 730, 992, 1200, 1366, 1496, and 1600 ms)

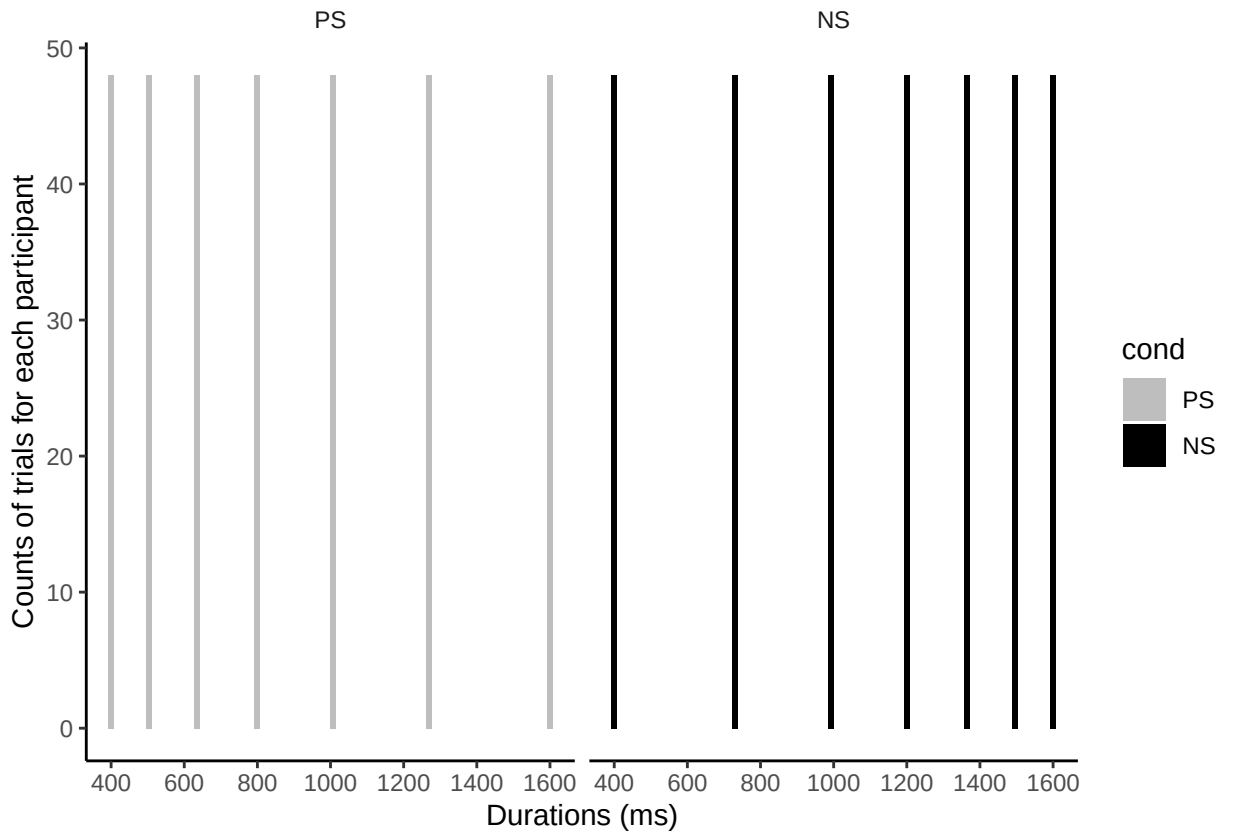
Range Frequency Theory

To understand the pattern of results concerning schifts in the temporal bisection point, We have to check the distribution of stimuli within the sets firstly.

```
sumfreq_exp1_sub1 <-  dat_exp1 %>% filter(NSub == 1) %>% group_by(cond, curDur) %>% summarise(count = n
```

```
## `summarise()` regrouping output by 'cond' (override with `.groups` argument)
```

```
count_exp1 <- sum(sumfreq_exp1_sub1$count)/2
sumfreq_exp1_sub1$exp <- 'Exp1'
sumfreq_exp1_sub1$freq <- sumfreq_exp1_sub1$count/count_exp1*100

# rename columns
fig_exp1_sub1_dist <- ggplot(data=sumfreq_exp1_sub1, aes(x=curDur, y=count, fill=cond)) +
  geom_bar(stat="identity", position=position_dodge()) +
  #geom_smooth(method="loess",se = FALSE)+
  xlab('Durations (ms)') + ylab('Counts of trials for each participant') +
  scale_x_continuous(breaks=seq(0, 1600, 200)) +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +mytheme+ facet_wrap(~cond)

fig_exp1_sub1_dist
```



plot the distribution of stimulus set

```
mDist_exp1 <-  dat_exp1 %>% filter(NSub == 1) %>% dplyr::group_by(cond) %>%
  summarise(m_cDur = mean(curDur), n = dplyr::n(),
            cDur_se = sd(curDur)/sqrt(n-1))
```

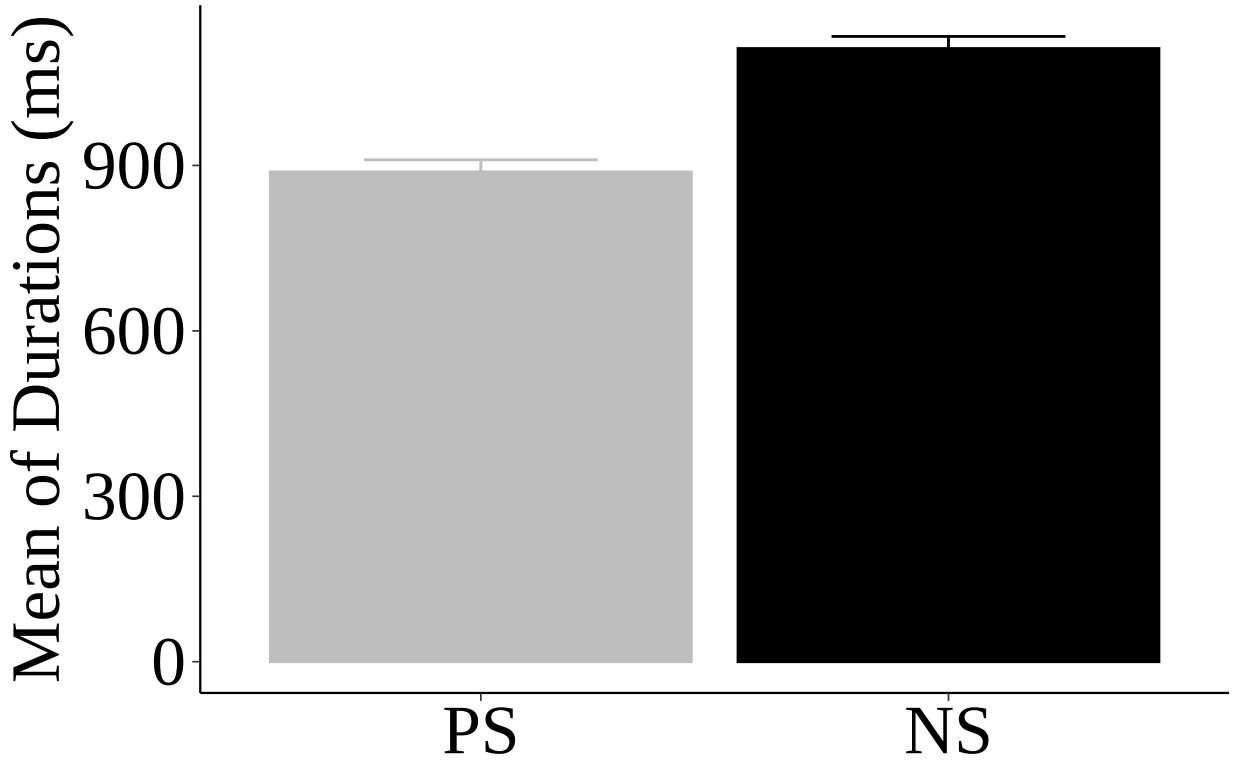
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
mDist_exp1
```

```
## # A tibble: 2 x 4
##   cond  m_cDur     n cDur_se
##   <fct>  <dbl> <int>   <dbl>
```

```
## 1 PS      888.   336    21.9
## 2 NS     1112    336    21.9
```

```
plot_mDist_exp1 <- ggplot(mDist_exp1,aes(cond, m_cDur, color = cond, fill = cond))+
  geom_bar(stat='identity') +
  geom_errorbar(aes(ymin = m_cDur - cDur_se, ymax = m_cDur + cDur_se), width =0.5) +
  labs(x = " ", y = "Mean of Durations (ms)") +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none", legend.title = element_blank())+
  theme(panel.grid.major=element_line(colour=NA))+
  theme(axis.text.y = element_text(size=26,colour = "black"),
        axis.text.x   = element_text(size=26, colour = "black"),
        axis.title.y  = element_text(size=26, colour = "black"),
        axis.title.x  = element_text(size=26, colour = "black"),
        panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"))

plot_mDist_exp1
```



fit psychometric functions

```
fits_exp1 = quickpsy(dat_exp1, x = curDur, k =RP, prob = .5,
                     grouping = .(cond, NSub), thresholds = FALSE)
```

```
## Warning: `group_by_()` is deprecated as of dplyr 0.7.0.
## Please use `group_by()` instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
## Warning: `summarise_()` is deprecated as of dplyr 0.7.0.
## Please use `summarise()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
head(fits_exp1$thresholds)
```

```
##   cond NSub      thre prob   threinf   thresup
## 1   NS    1 1176.8920  0.5 1131.1668 1213.6404
## 2   NS   10  901.0311  0.5  854.2135  943.4593
## 3   NS   11  620.3980  0.5  565.7694  680.8397
## 4   NS   12  881.9966  0.5  846.0813  914.5038
## 5   NS   13 1026.7736  0.5  975.1354 1069.3622
## 6   NS   14  840.6738  0.5  791.7247  884.4138
```

```
fits_exp1_all = quickpsy(dat_exp1, x = curDur, k =RP, prob = .5, grouping = .(cond))
fits_exp1_all
```

```
## # A tibble: 4 x 5
## # Groups:   cond [2]
##   cond  parn    par parinf parsup
##   <fct> <fct> <dbl>  <dbl>  <dbl>
## 1 PS    p1     850.   836.   864.
## 2 PS    p2     246.   236.   256.
## 3 NS    p1     936.   924.   950.
## 4 NS    p2     267.   255.   277.
## NULL
##   cond     thre prob  threinf  thresup
## 1   NS 935.7437  0.5 923.9106 950.2928
## 2   PS 850.1932  0.5 836.4492 863.8107
```

```
# plot fitted function
plot_fit_exp1 <- plot(fits_exp1_all) +
  labs(x = "Durations (ms)", y = "Proportion of 'Long' responses") +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = c(0.7, 0.2), legend.title = element_blank())
```

```
## Warning: Ignoring unknown aesthetics: x
```

```
plot_fit_exp1
```

```
aic(fits_exp1_all)
```

```
## # A tibble: 2 x 2
## # Groups:   cond [2]
##   cond    aic
##   <fct> <dbl>
## 1 PS    121.
## 2 NS     56.3
```

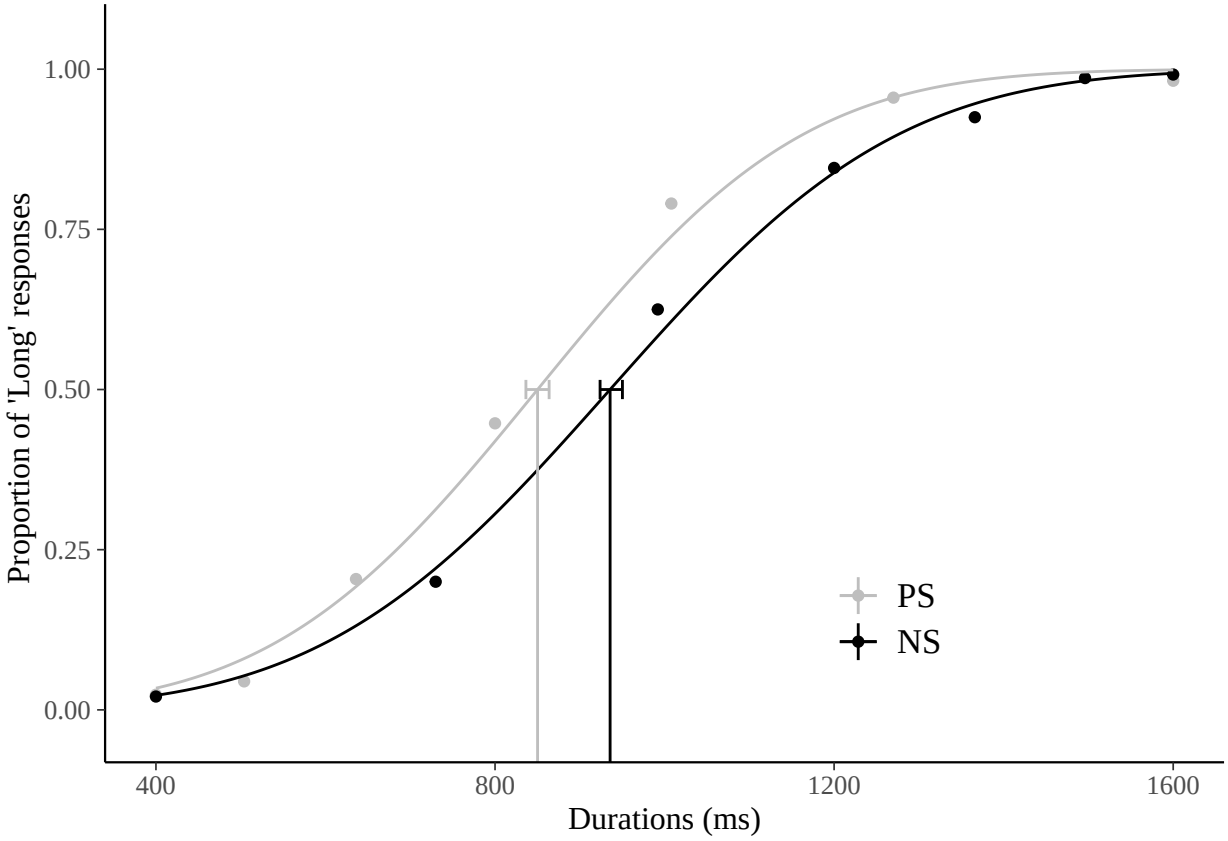


Figure 1: PSE: Bisection task performance in Experiment Spacing.

```
ypred(fits_exp1_all)
```

```
## # A tibble: 14 x 3
## # Groups:   cond [2]
##    cond      x  ypred
##    <fct> <dbl>  <dbl>
##  1 PS      400 0.0338
##  2 PS      504 0.0800
##  3 PS      636 0.192
##  4 PS      800 0.419
##  5 PS     1008 0.739
##  6 PS     1270 0.956
##  7 PS     1600 0.999
##  8 NS      400 0.0226
##  9 NS      730 0.221
## 10 NS      992 0.583
## 11 NS     1200 0.838
## 12 NS     1366 0.946
## 13 NS     1496 0.982
## 14 NS     1600 0.993
```

======= ##slope From the equation of psychometric fuction (PF), we can easily obtained the point of subjective equality (PSE, α) and just noticeable difference (JND). Notice, the JND is defined by the difference between thresholds P=0.5 and P=0.75, so we have

$$JND = log3/\beta$$

Therefore,

$$\beta = 3/e^{JND}$$

```
exp1_all_thres<- fits_exp1_all$thresholds
exp1_all_thres$slope = 3/exp(exp1_all_thres$thre/1000) #slope
exp1_all_thres
```

```
##   cond     thre prob  threinf  thresup    slope
## 1   NS 935.7437  0.5 923.9106 950.2928 1.176882
## 2   PS 850.1932  0.5 836.4492 863.8107 1.281997
```

```
exp1_thres<- fits_exp1$thresholds
exp1_thres$slope = 3/exp(exp1_thres$thre/1000) #slope
exp1_thres
```

```
##    cond NSub      thre prob   threinf   thresup     slope
## 1    NS    1 1176.8920  0.5 1131.1668 1213.6404 0.9247058
## 2    NS   10  901.0311  0.5  854.2135  943.4593 1.2184520
## 3    NS   11  620.3980  0.5  565.7694  680.8397 1.6131912
## 4    NS   12  881.9966  0.5  846.0813  914.5038 1.2418668
## 5    NS   13 1026.7736  0.5  975.1354 1069.3622 1.0744820
## 6    NS   14  840.6738  0.5  791.7247  884.4138 1.2942593
## 7    NS   15 1126.8929  0.5 1087.2640 1171.1666 0.9721155
## 8    NS    2 1136.4382  0.5 1086.3376 1179.8907 0.9628805
## 9    NS    3  892.3247  0.5  821.4648  960.2049 1.2291066
## 10   NS    4  834.3695  0.5  780.6554  888.6759 1.3024443
## 11   NS    5  770.9447  0.5  702.5812  830.7877 1.3877276
```

```
## 12   NS    6  952.6421  0.5  903.7147 1010.7078 1.1571617
## 13   NS    7  970.2696  0.5  928.4461 1014.6108 1.1369425
## 14   NS    8 1030.4110  0.5  987.5801 1068.2808 1.0705808
## 15   NS    9  863.6584  0.5  823.7427  909.3899 1.2648505
## 16   PS    1 1091.8930  0.5 1036.0332 1150.0165 1.0067420
## 17   PS   10  768.6700  0.5  728.7985  801.8850 1.3908878
## 18   PS   11  620.1452  0.5  594.5960  651.4560 1.6135990
## 19   PS   12  700.7757  0.5  671.8167  728.4581 1.4886008
## 20   PS   13  919.8234  0.5  891.2356  953.2192 1.1957683
## 21   PS   14  787.7762  0.5  752.2932  824.6859 1.3645656
## 22   PS   15  927.0499  0.5  898.5615  953.9815 1.1871582
## 23   PS    2 1032.5088  0.5  997.5131 1082.6790 1.0683373
## 24   PS    3  839.7529  0.5  789.2323  890.7459 1.2954516
## 25   PS    4  678.4811  0.5  638.2115  712.9463 1.5221612
## 26   PS    5  661.1666  0.5  621.6173  713.0752 1.5487462
## 27   PS    6  996.0497  0.5  946.4430 1060.6671 1.1080066
## 28   PS    7  917.5433  0.5  865.5339  954.7186 1.1984979
## 29   PS    8  874.2151  0.5  829.3827  917.8361 1.2515681
## 30   PS    9  920.7368  0.5  883.8067  963.3274 1.1946766
```

list all of the parameter for each subjects

```
par_exp1 = fits_exp1$par %>%
  dplyr::select(-parinf,-parsup) %>%
  spread(parn, par) %>%
  rename(pse =p1, jnd = p2)  # rename columns
par_exp1$slope = 3/exp(par_exp1$jnd/1000) #slope

par_exp1
```

```
## # A tibble: 30 x 5
## # Groups:   cond, NSub [30]
##    cond   NSub   pse   jnd slope
##    <fct> <int> <dbl> <dbl> <dbl>
##  1 PS        1 1092.  224.  2.40
##  2 PS        2 1033.  185.  2.49
##  3 PS        3  840.  317.  2.18
##  4 PS        4  678.  194.  2.47
##  5 PS        5  661.  247.  2.34
##  6 PS        6  996.  299.  2.22
##  7 PS        7  918.  221.  2.41
##  8 PS        8  874.  228.  2.39
##  9 PS        9  921.  190.  2.48
## 10 PS       10  769.  164.  2.55
## 11 PS       11  620.  151.  2.58
## 12 PS       12  701.  108.  2.69
## 13 PS       13  920.  120.  2.66
## 14 PS       14  788.  216.  2.42
## 15 PS       15  927.  106.  2.70
## 16 NS        1 1177.  207.  2.44
## 17 NS        2 1136.  295.  2.23
## 18 NS        3  892.  321.  2.18
## 19 NS        4  834.  241.  2.36
## 20 NS        5  771.  312.  2.20
```

```
## 21 NS        6  953.  254.  2.33
## 22 NS        7  970.  245.  2.35
## 23 NS        8 1030.  201.  2.45
## 24 NS        9  864.  176.  2.52
## 25 NS       10  901.  182.  2.50
## 26 NS       11  620.  247.  2.34
## 27 NS       12  882.  118.  2.67
## 28 NS       13 1027.  167.  2.54
## 29 NS       14  841.  212.  2.43
## 30 NS       15 1127.  187.  2.49
```

```
mpars_exp1 = par_exp1%>%
  group_by(cond) %>%
  summarise(m_pse = mean(pse), m_jnd = mean(jnd),  m_slope = mean(slope), n = n(),
            pse_se = sd(pse)/sqrt(n-1),
            jnd_se = sd(jnd)/sqrt(n-1),
            slope_se = sd(slope)/sqrt(n-1))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
mpars_exp1
```

```
## # A tibble: 2 x 8
##   cond  m_pse m_jnd m_slope     n pse_se jnd_se slope_se
##   <fct> <dbl> <dbl>   <dbl> <int>  <dbl>  <dbl>    <dbl>
## 1 PS     849.  198.    2.47    15   38.1   16.9   0.0413
## 2 NS     935.  224.    2.40    15   39.8   15.2   0.0363
```

======= ##Export for JASP

List all of the parameters for exp1 and export data for JASP

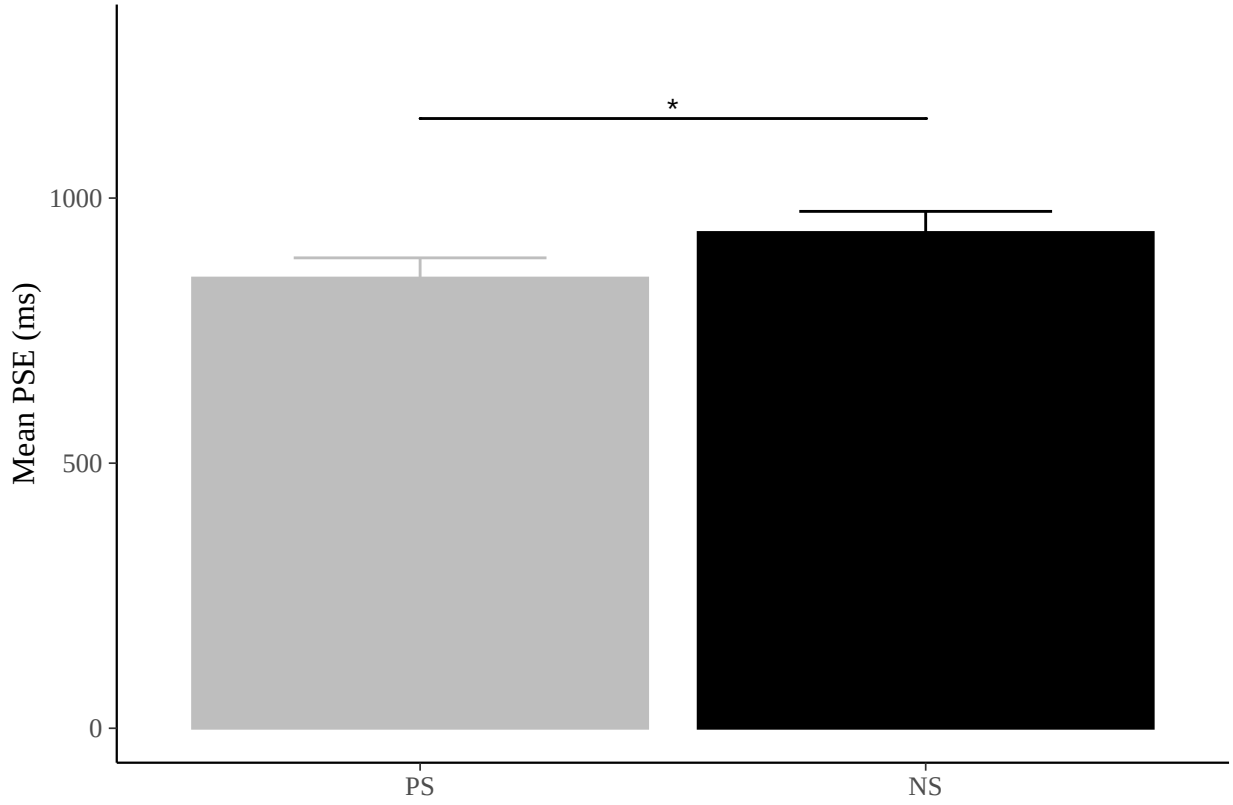
```
exp1_jasp <- par_exp1%>%
  unite(pse.jnd.slope, pse, jnd, slope)%>%
  spread(cond, pse.jnd.slope)%>%
  separate(`PS`, c("PSC_pse", "PSC_jnd","PSC_slope"), sep = "_")%>%
  separate(`NS`, c("NSC_pse", "NSC_jnd","NSC_slope"), sep = "_")

exp1_jasp
```

```
## # A tibble: 15 x 7
## # Groups:   NSub [15]
##     NSub PSC_pse    PSC_jnd    PSC_slope    NSC_pse   NSC_jnd   NSC_slope
##    <int> <chr>      <chr>      <chr>        <chr>     <chr>     <chr>
##  1     1 1091.8929~ 223.61483~ 2.398869184~ 1176.891~ 207.3750~ 2.43814423~
##  2     2 1032.5088~ 184.96938~ 2.493389183~ 1136.438~ 295.1010~ 2.23336907~
##  3     3 839.75294~ 317.31210~ 2.184310432~ 892.3246~ 321.0167~ 2.17623325~
##  4     4 678.48114~ 193.92171~ 2.471167167~ 834.3695~ 241.0707~ 2.35735801~
##  5     5 661.16659~ 246.92671~ 2.343593833~ 770.9446~ 311.6254~ 2.19676718~
##  6     6 996.04972~ 299.48244~ 2.223605200~ 952.6421~ 253.9645~ 2.32715780~
##  7     7 917.54325~ 220.57503~ 2.406172365~ 970.2696~ 245.1127~ 2.34784899~
##  8     8 874.21505~ 227.72608~ 2.389027070~ 1030.411~ 201.1599~ 2.45334482~
##  9     9 920.73675~ 189.56690~ 2.481952082~ 863.6583~ 175.7679~ 2.51643771~
## 10    10 768.67001~ 163.51119~ 2.547470980~ 901.0311~ 181.8657~ 2.50113987~
## 11    11 620.14517~ 150.66613~ 2.580404458~ 620.3979~ 246.9644~ 2.34350537~
## 12    12 700.77567~ 107.74983~ 2.693556544~ 881.9965~ 118.3684~ 2.66510593~
## 13    13 919.82337~ 119.71543~ 2.661518584~ 1026.773~ 166.5968~ 2.53962256~
```

```
## 14    14 787.77616~ 215.51749~ 2.418372492~ 840.6737~ 211.7434~ 2.42751681~
## 15    15 927.04991~ 105.81063~ 2.698784951~ 1126.892~ 186.8933~ 2.48859652~
```

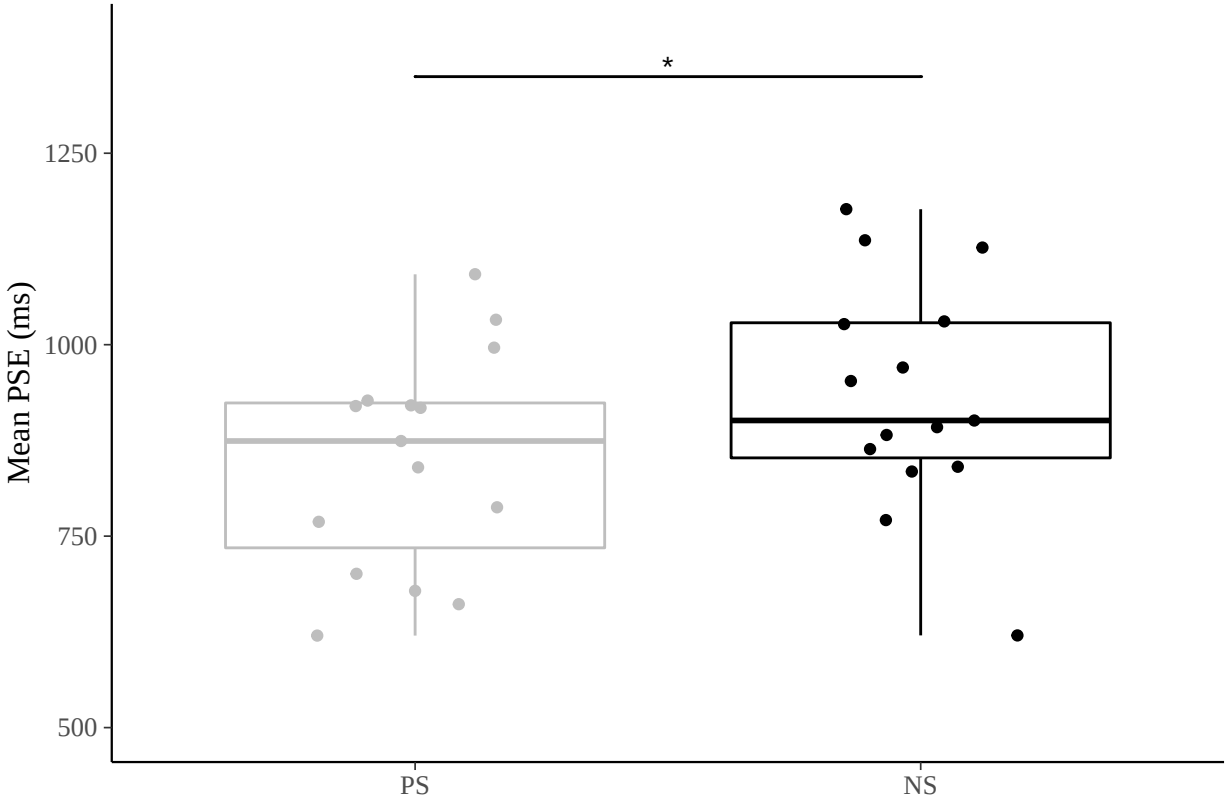
plot PSE

```
plot_pse_exp1 <- ggplot(mpars_exp1,aes(cond, m_pse, color = cond, fill = cond))+
  geom_bar(stat='identity') +
  geom_errorbar(aes(ymin = m_pse - pse_se, ymax = m_pse + pse_se), width =0.5) +
  geom_signif(comparisons=list(c("PS", "NS")), annotations="*", y_position = 1150, tip_length = 0, vjus
  labs(x = " ", y = "Mean PSE (ms)") +
  coord_cartesian(ylim = c(0,1300)) +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none")

plot_pse_exp1
```



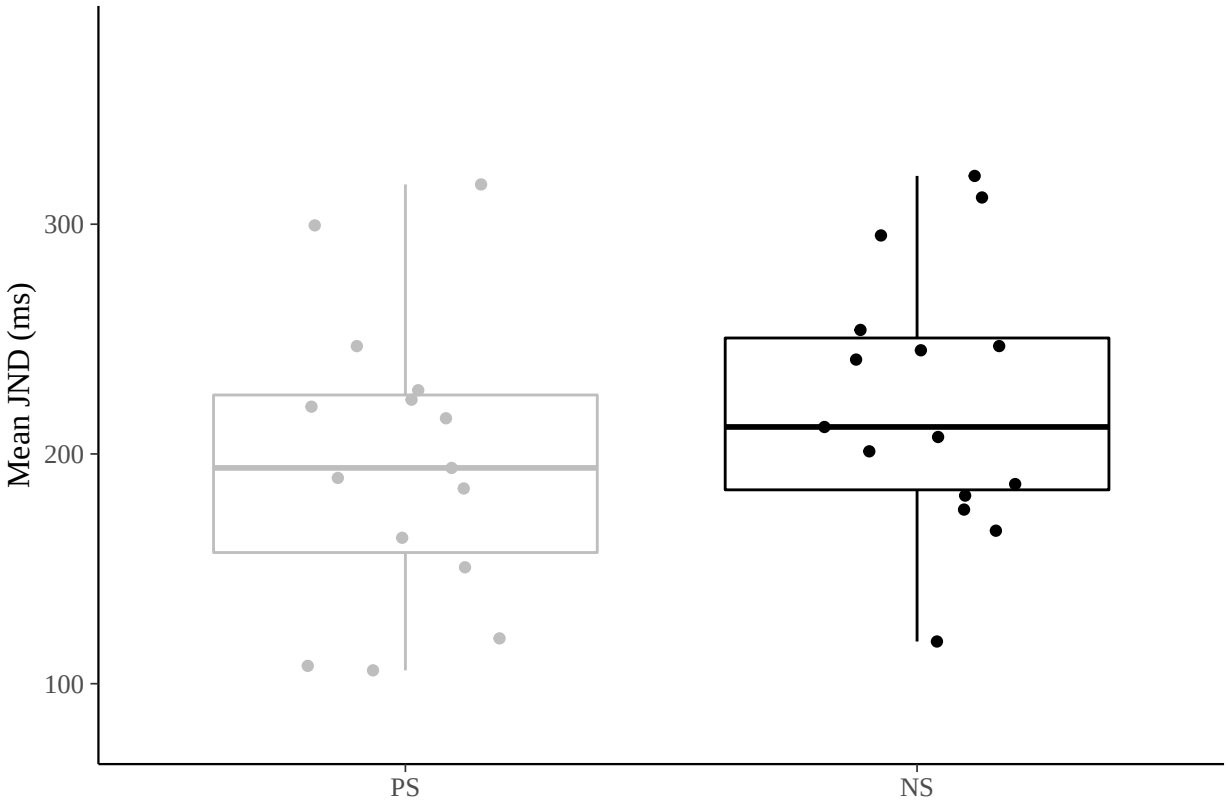

```
plot_pse_exp1_box <- ggplot(par_exp1,aes(x=cond, y=pse, color = cond))+
  geom_boxplot() +
  geom_jitter(position=position_jitter(0.2))+
  geom_signif(comparisons=list(c("PS", "NS")), annotations="*", y_position = 1350, tip_length = 0, vjus
  labs(x = " ", y = "Mean PSE (ms)") +
  coord_cartesian(ylim = c(500,1400)) +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none")
```

```
plot_pse_exp1_box
```

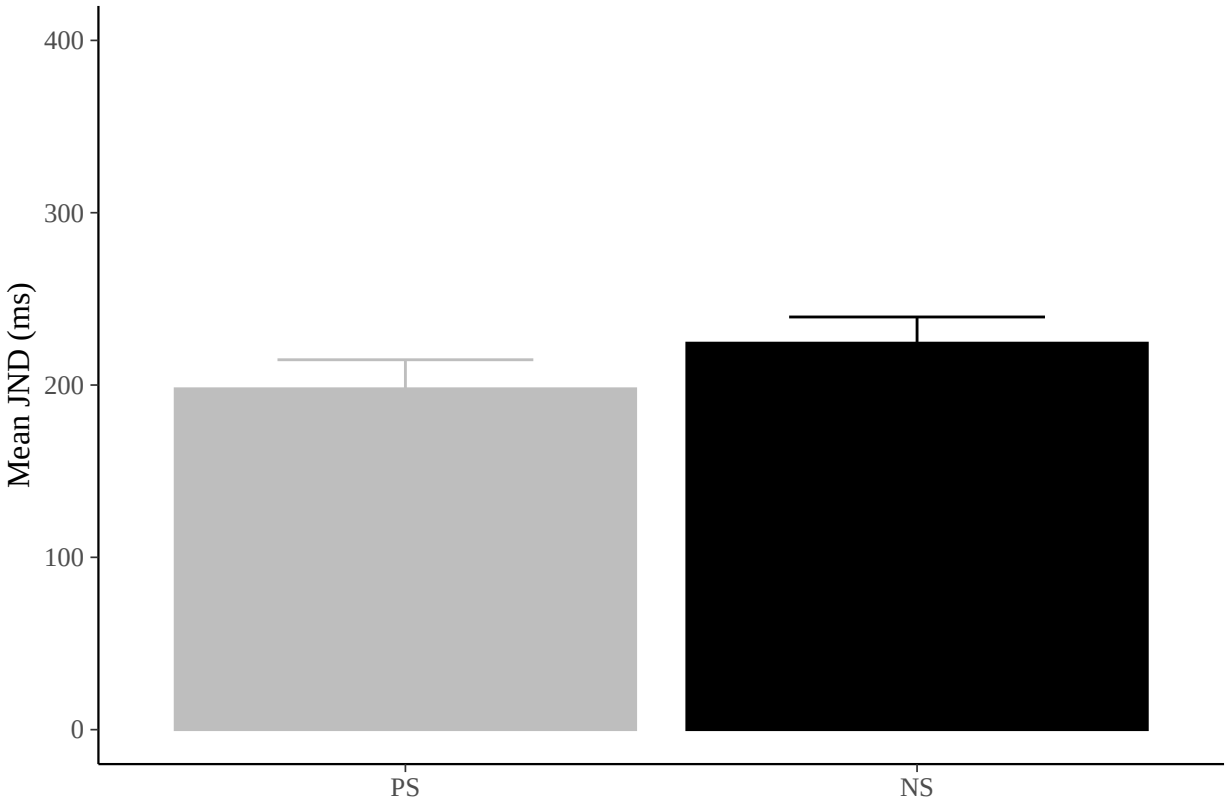


plot JND

```
plot_jnd_exp1_box <- ggplot(par_exp1,aes(x=cond, y=jnd, color = cond))+
  geom_boxplot() +
  geom_jitter(position=position_jitter(0.2))+
  labs(x = " ", y = "Mean JND (ms)") +
  coord_cartesian(ylim = c(80,380))+
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) + theme(legend.position = "none")

plot_jnd_exp1_box
```




```
plot_jnd_exp1 <- ggplot(mpars_exp1,aes(cond, m_jnd, color = cond, fill = cond))+
  geom_bar(stat='identity') +
  geom_errorbar(aes(ymin = m_jnd - jnd_se, ymax = m_jnd + jnd_se), width =0.5) +
  labs(x = " ", y = "Mean JND (ms)") +
  coord_cartesian(ylim = c(0,400))+
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) + theme(legend.position = "none")

plot_jnd_exp1
```



ANOVA and T-test

```
Anova_exp1_pse<- ezANOVA(data = par_exp1, dv= pse, wid=NSub, within=.(cond))
```

```
## Warning: Converting "NSub" to factor for ANOVA.
```

```
Anova_exp1_pse
```

```
## $ANOVA
##   Effect DFn DFd        F           p p<.05        ges
## 2   cond   1  14 18.64126 0.000709063     * 0.08508602
```

```
Anova_exp1_jnd<- ezANOVA(data = par_exp1, dv= jnd, wid=NSub, within=.(cond))
```

```
## Warning: Converting "NSub" to factor for ANOVA.
```

```
Anova_exp1_jnd
```

```
## $ANOVA
##   Effect DFn DFd        F          p p<.05        ges
## 2   cond   1  14 4.859249 0.04473224     * 0.04960966
```

```
Anova_exp1_slope<- ezANOVA(data = par_exp1, dv= slope, wid=NSub, within=.(cond))
```

```
## Warning: Converting "NSub" to factor for ANOVA.
```

```
Anova_exp1_slope
```

```
## $ANOVA
```

```
##   Effect DFn DFd        F          p p<.05        ges
## 2   cond   1  14 5.054868 0.04118543     * 0.05122824
```

Experiment 2

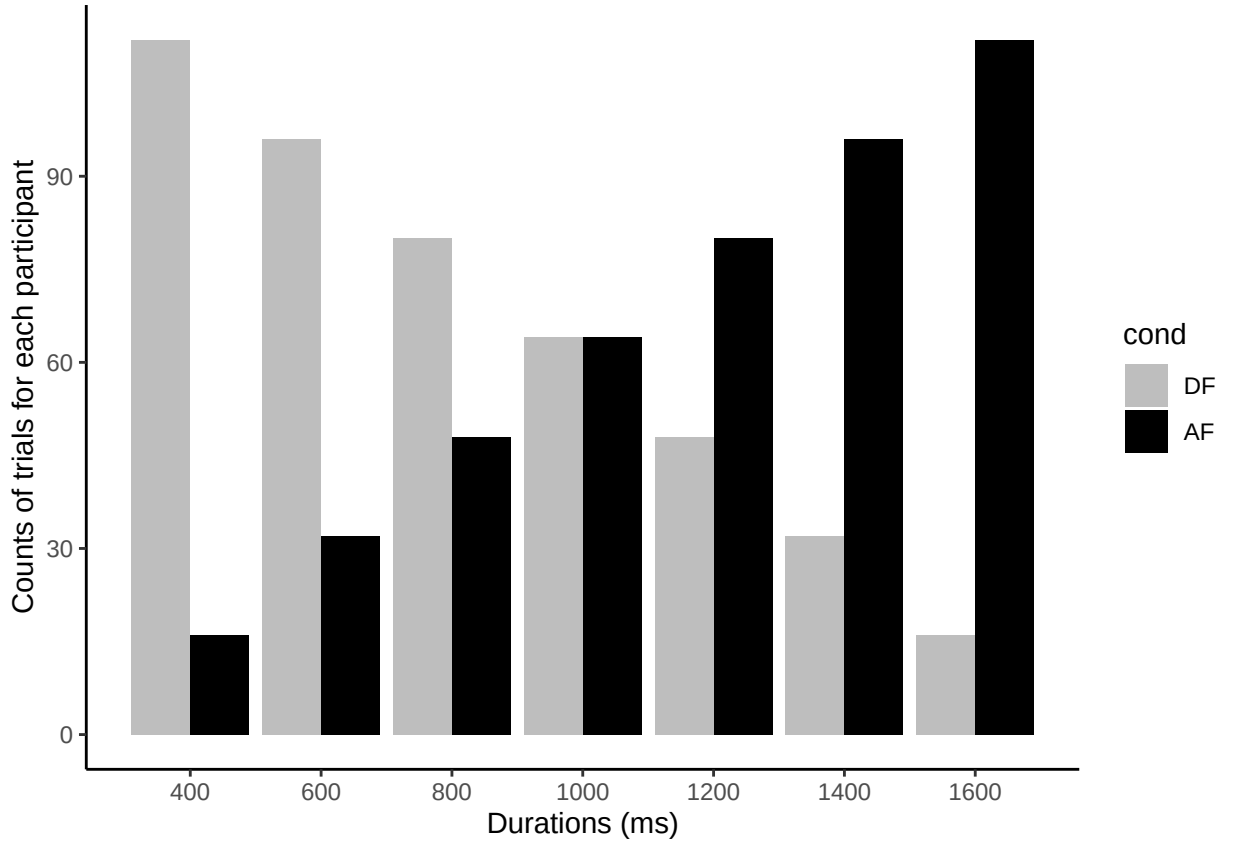
Range Frequency Theory

A particular focus is the distribution of stimuli within the sets to be judged. To understand the pattern of results concerning schifts in the temporal bisection point, We have to check the frequency of the sets firstly.

```
sumfreq_exp2_sub1 <-  dat_exp2 %>% filter(NSub == 1) %>% group_by(cond, curDur) %>% summarise(count = n
```

```
## `summarise()` regrouping output by 'cond' (override with `.groups` argument)
```

```
count_exp2 <- sum(sumfreq_exp2_sub1$count)/2
sumfreq_exp2_sub1$exp <- 'Exp2'
sumfreq_exp2_sub1$freq <- sumfreq_exp2_sub1$count/count_exp2*100
fig_exp2_sub1_dist <- ggplot(data=sumfreq_exp2_sub1, aes(x=curDur, y=count, fill=cond)) +
  geom_bar(stat="identity", position=position_dodge()) +
  xlab('Durations (ms)') + ylab('Counts of trials for each participant') +
  scale_x_continuous(breaks=seq(0, 1600, 200)) +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +mytheme

fig_exp2_sub1_dist
```



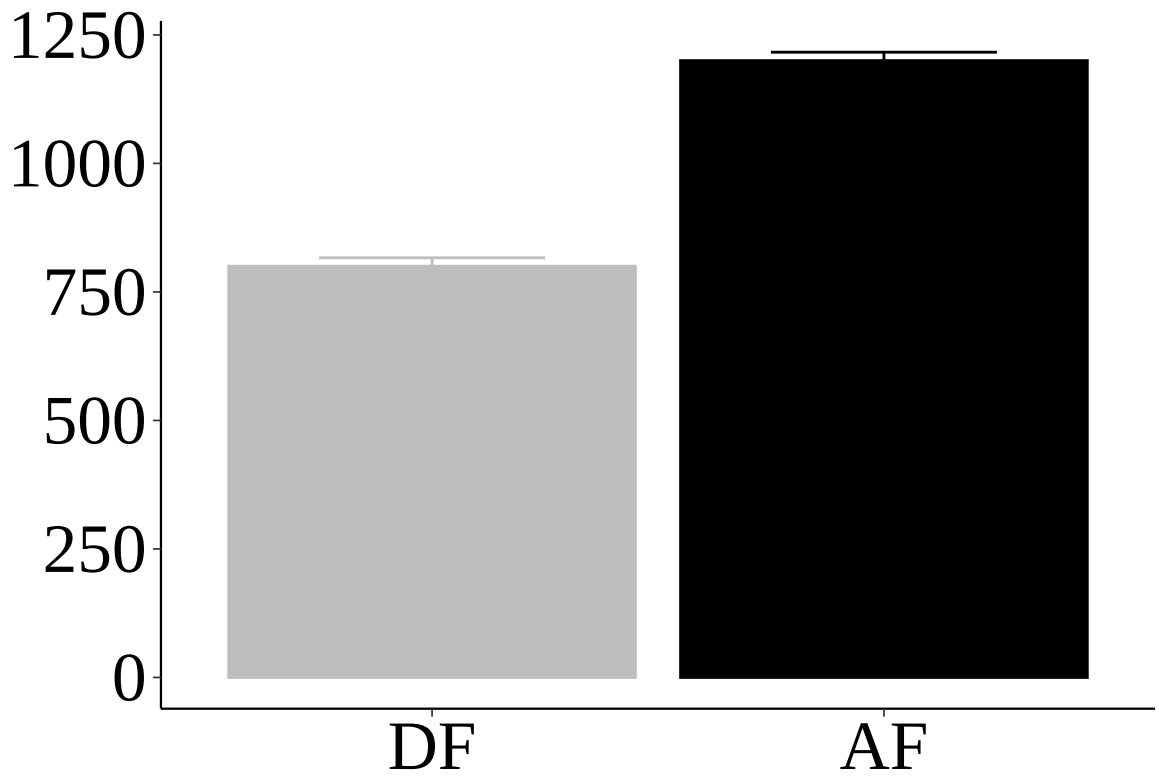
```r
mDist_exp2 <-  dat_exp2 %>% filter(NSub == 1) %>% dplyr::group_by(cond) %>%
  summarise(m_cDur = mean(curDur), n = dplyr::n(),
            cDur_se = sd(curDur)/sqrt(n-1))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
mDist_exp2
```

```
## # A tibble: 2 x 4
##   cond  m_cDur     n cDur_se
##   <fct>  <dbl> <int>   <dbl>
## 1 DF       800   448    16.4
## 2 AF      1200   448    16.4
```

```r
plot_mDist_exp2 <- ggplot(mDist_exp2,aes(cond, m_cDur, color = cond, fill = cond))+
  geom_bar(stat='identity') +
  geom_errorbar(aes(ymin = m_cDur - cDur_se, ymax = m_cDur + cDur_se), width =0.5) +
  #geom_text(aes(label=paste("(",m_cDur,",",round(cDur_se),")") ),size=8, hjust=0.5, vjust=-2, color =
  labs(x = " ", y = " ") +
  #coord_cartesian(ylim = c(0,1600))+
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none", legend.title = element_blank())+
  theme(panel.grid.major=element_line(colour=NA))+
  theme(axis.text.y = element_text(size=26,colour = "black"),
        axis.text.x   = element_text(size=26, colour = "black"),
        axis.title.y  = element_text(size=26, colour = "black"),
        axis.title.x  = element_text(size=26, colour = "black"),
        panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black")
  )

plot_mDist_exp2
```



Fit psychometric functions

```
fits_exp2 = quickpsy(dat_exp2, x = curDur, k =RP, prob = .5, grouping = .(cond, NSub))
```

```
fits_exp2_all = quickpsy(dat_exp2, x = curDur, k =RP,
                         grouping = .(cond))
fits_exp2_all
```

```
## # A tibble: 4 x 5
## # Groups:   cond [2]
##   cond  parn    par parinf parsup
##   <fct> <fct> <dbl>  <dbl>  <dbl>
## 1 DF    p1     828.   818.   835.
## 2 DF    p2     217.   208.   229.
## 3 AF    p1     995.   982.  1012.
## 4 AF    p2     284.   268.   295.
## NULL
##   cond     thre prob  threinf   thresup
## 1   AF 994.9393  0.5 982.0360 1011.6056
## 2   DF 827.8988  0.5 818.0314  835.3432
```

```
# plot fitted function
plot_fit_exp2 <- plot(fits_exp2_all) +
  labs(x = "Durations (ms)", y = "Proportion of 'Long' responses") +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
```

```
  theme(legend.position = c(0.7, 0.2), legend.title = element_blank())
```

```
## Warning: Ignoring unknown aesthetics: x
```

```
plot_fit_exp2
```

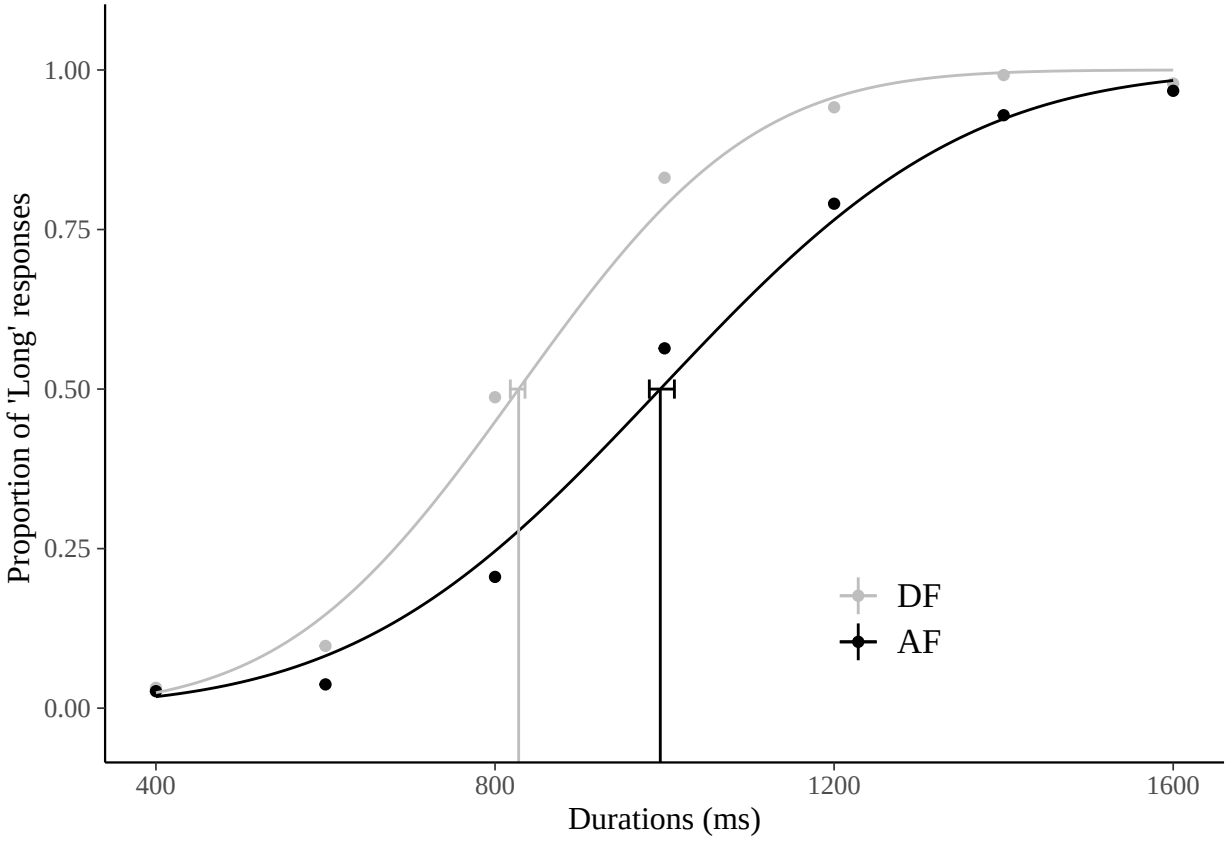


Figure 2: PSE: Bisection task performance in Experiment 2

Average fitted paramters and plot

```
par_exp2 = fits_exp2$par %>%
  dplyr::select(-parinf,-parsup) %>%
  spread(parn, par) %>%
  rename(pse =p1, jnd = p2)  # rename columns

par_exp2$slope = 3/exp(par_exp2$pse/1000) #add slope

par_exp2
```

```
## # A tibble: 30 x 5
## # Groups:   cond, NSub [30]
##    cond   NSub   pse   jnd slope
##    <fct> <int> <dbl> <dbl> <dbl>
##  1 DF        1  777. 141.  1.38
##  2 DF        2  694. 171.  1.50
##  3 DF        3  815. 109.  1.33
```

```
##  4 DF        4  820.  92.8 1.32
##  5 DF        5  997. 155.  1.11
##  6 DF        6 1160. 204.  0.940
##  7 DF        7  814. 132.  1.33
##  8 DF        8  816. 121.  1.33
##  9 DF        9  788. 139.  1.36
## 10 DF       10  562. 640.  1.71
## 11 DF       11  836. 200.  1.30
## 12 DF       12  868. 100.  1.26
## 13 DF       13  755. 192.  1.41
## 14 DF       14  705. 127.  1.48
## 15 DF       15  930. 147.  1.18
## 16 AF        1  910. 140.  1.21
## 17 AF        2  865. 240.  1.26
## 18 AF        3  827. 217.  1.31
## 19 AF        4 1065. 168.  1.03
## 20 AF        5 1031. 180.  1.07
## 21 AF        6 1409. 239.  0.733
## 22 AF        7 1023. 223.  1.08
## 23 AF        8  787. 125.  1.37
## 24 AF        9  853. 167.  1.28
## 25 AF       10  867. 757.  1.26
## 26 AF       11 1032. 252.  1.07
## 27 AF       12 1039. 116.  1.06
## 28 AF       13 1252. 177.  0.858
## 29 AF       14  924. 236.  1.19
## 30 AF       15 1075. 159.  1.02
```

```
mpse_exp2 = par_exp2%>%
  group_by(cond) %>%
  summarise(m_pse = mean(pse), m_jnd = mean(jnd), m_slope = mean(slope), n = n(),
            pse_se = sd(pse)/sqrt(n-1),
            jnd_se = sd(jnd)/sqrt(n-1),
            slope_se = sd(slope)/sqrt(n-1))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
mpse_exp2
```

```
## # A tibble: 2 x 8
##   cond  m_pse m_jnd m_slope     n pse_se jnd_se slope_se
##   <fct> <dbl> <dbl>   <dbl> <int>  <dbl>  <dbl>    <dbl>
## 1 DF     823.  178.    1.33    15   36.7   35.4   0.0471
## 2 AF     997.  226.    1.12    15   44.7   40.9   0.0463
```

Export for JASP

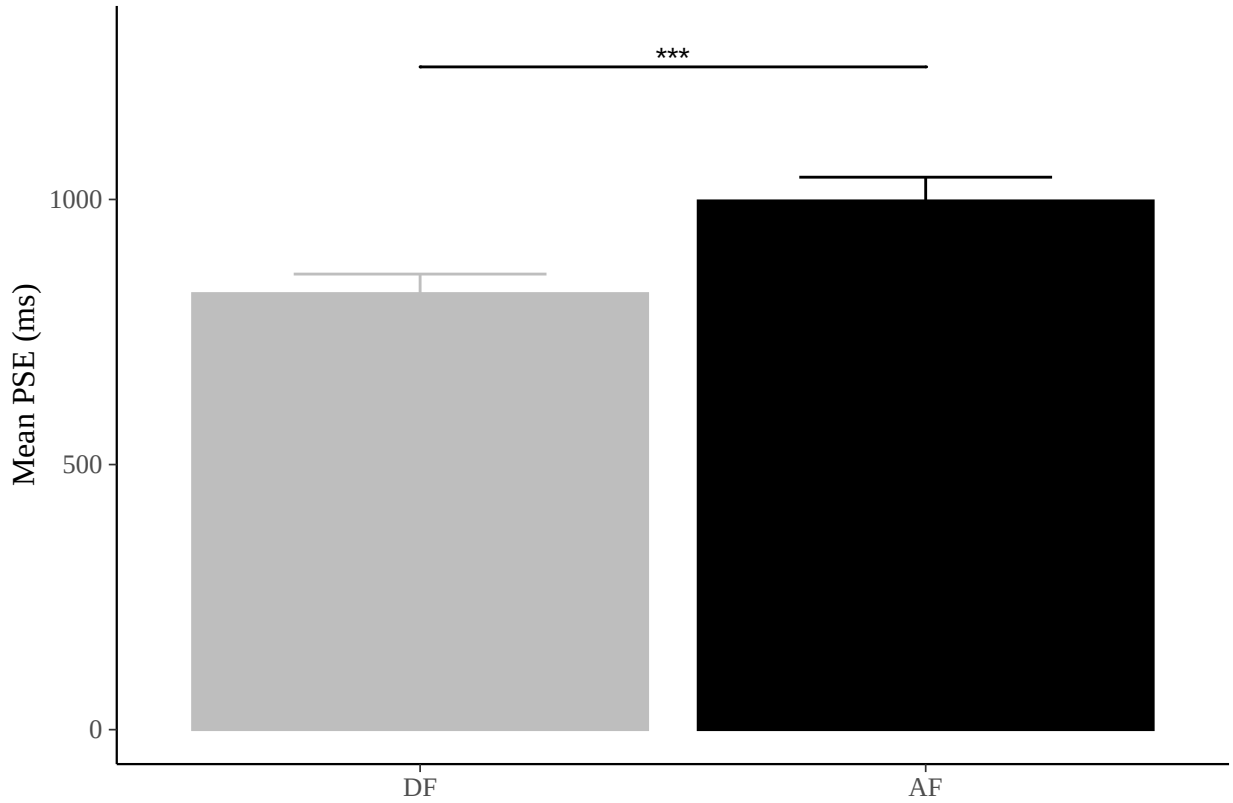
List all of the parameters for Exp2 and export data for JASP

```
exp2_jasp <- par_exp2%>%
  unite(pse.jnd.slope, pse, jnd, slope)%>%
  spread(cond, pse.jnd.slope)%>%
  separate(DF, c("DF_pse", "DF_jnd","DF_slope"), sep = "_")%>%
  separate(AF, c("AF_pse", "AF_jnd", "AF_slope"), sep = "_")

exp2_jasp
```

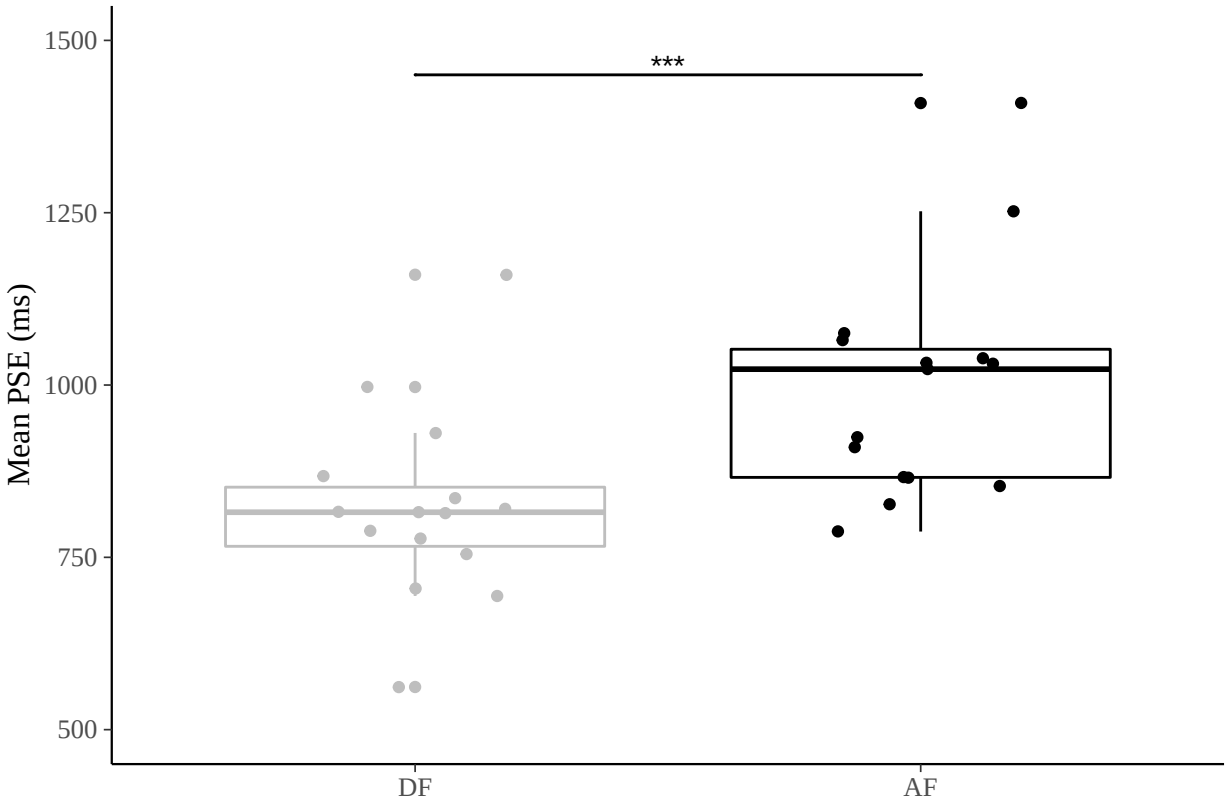
```
## # A tibble: 15 x 7
## # Groups:   NSub [15]
##     NSub DF_pse     DF_jnd     DF_slope     AF_pse    AF_jnd    AF_slope
##    <int> <chr>      <chr>      <chr>        <chr>     <chr>     <chr>
##  1     1 777.28414~ 140.82967~ 1.378957998~ 909.9241~ 140.1035~ 1.20766426~
##  2     2 693.94757~ 171.31923~ 1.498799895~ 865.4780~ 240.0498~ 1.26255091~
##  3     3 815.43339~ 109.49353~ 1.327342588~ 826.8661~ 217.4768~ 1.31225388~
##  4     4 820.19746~ 92.807867~ 1.321034079~ 1064.985~ 168.1606~ 1.03419902~
##  5     5 997.09079~ 154.85708~ 1.106853704~ 1030.886~ 180.0318~ 1.07007146~
##  6     6 1160.0308~ 203.53824~ 0.940429544~ 1409.068~ 238.8523~ 0.73311250~
##  7     7 813.94575~ 132.42829~ 1.329318665~ 1023.017~ 222.6214~ 1.07852512~
##  8     8 816.01219~ 121.06384~ 1.326574551~ 787.4202~ 125.4335~ 1.36505133~
##  9     9 788.38611~ 138.66421~ 1.363733526~ 853.4634~ 167.2376~ 1.27781144~
## 10    10 561.88660~ 640.04304~ 1.710397304~ 866.5211~ 756.9234~ 1.26123463~
## 11    11 835.74039~ 199.94361~ 1.300660086~ 1032.161~ 251.9229~ 1.06870872~
## 12    12 867.83135~ 100.48151~ 1.259583272~ 1038.798~ 115.7143~ 1.06163933~
## 13    13 754.66593~ 192.14947~ 1.410502961~ 1252.092~ 177.1462~ 0.85771767~
## 14    14 704.79297~ 126.57233~ 1.482632629~ 924.3530~ 235.9702~ 1.19036406~
## 15    15 930.44828~ 146.93851~ 1.183130634~ 1075.011~ 158.8315~ 1.02388144~
```

plot PSE

```
plot_pse_exp2 <- ggplot(mpse_exp2,aes(cond, m_pse, color = cond, fill = cond))+
  geom_bar(stat='identity') +
  geom_errorbar(aes(ymin = m_pse - pse_se, ymax = m_pse + pse_se), width =0.5) +
  geom_signif(comparisons=list(c("DF", "AF")), annotations="***", y_position = 1250, tip_length = 0, vj
  labs(x = " ", y = "Mean PSE (ms)") +
  coord_cartesian(ylim = c(0,1300))+
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none")

plot_pse_exp2
```

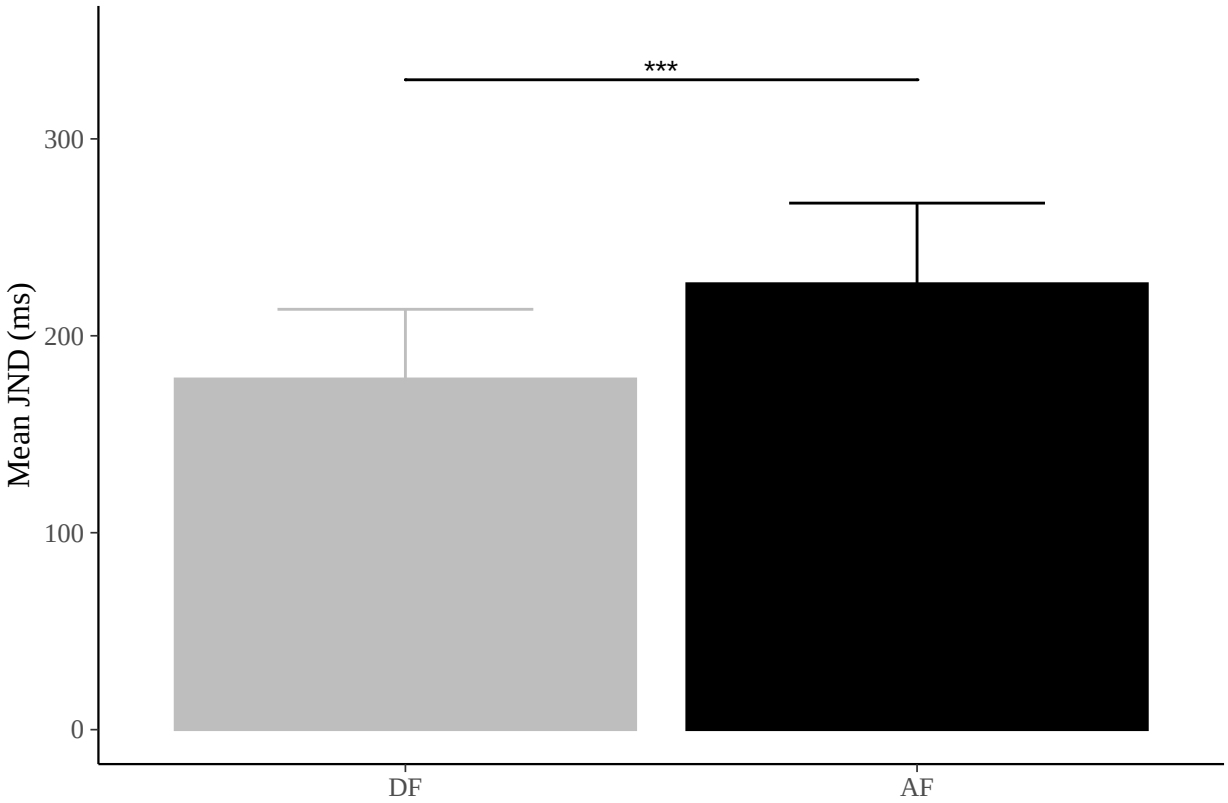



```
plot_pse_exp2_box <- ggplot(par_exp2,aes(x=cond, y=pse, color = cond))+
  geom_boxplot() +
  geom_jitter(position=position_jitter(0.2))+
  geom_signif(comparisons=list(c("DF", "AF")), annotations="***", y_position = 1450, tip_length = 0, vj
  labs(x = " ", y = "Mean PSE (ms)") +
  coord_cartesian(ylim = c(500,1500))+
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none")

plot_pse_exp2_box
```

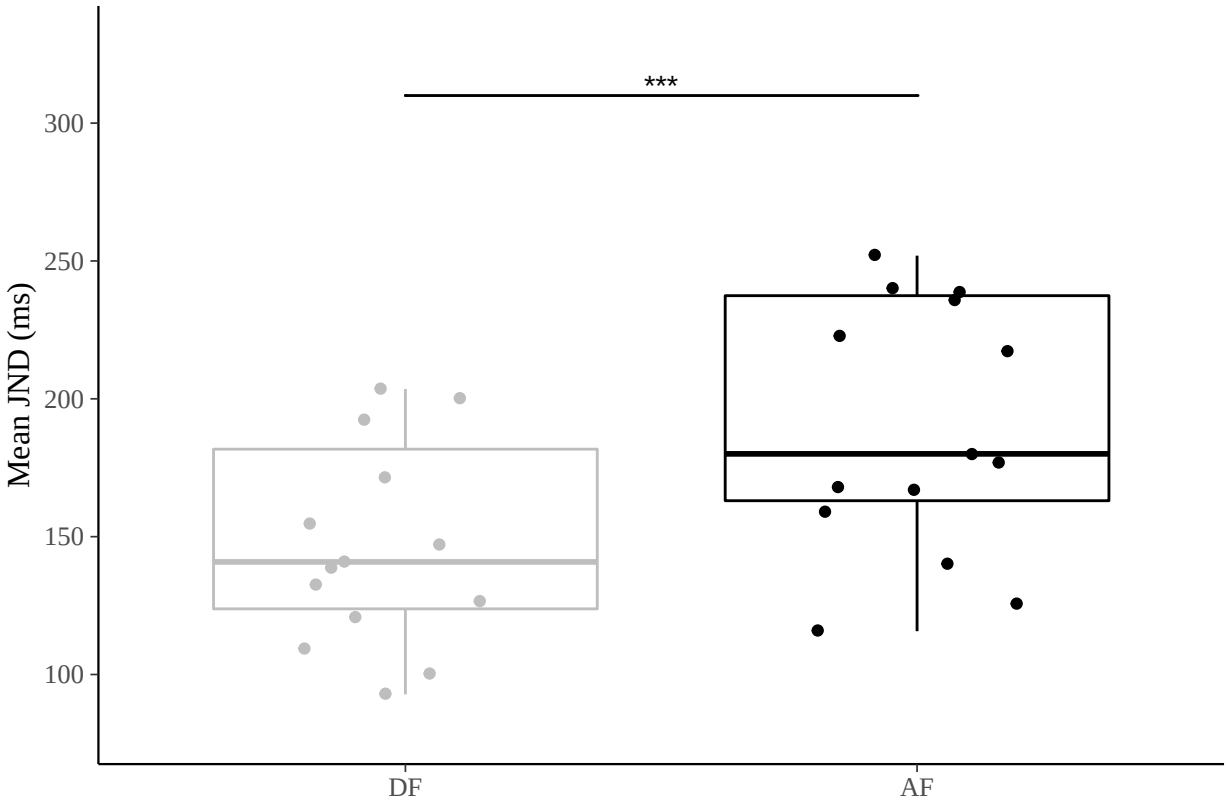


plot JND

```
plot_jnd_exp2 <- ggplot(mpse_exp2,aes(cond, m_jnd, color = cond, fill = cond))+
  geom_bar(stat='identity') +
  geom_errorbar(aes(ymin = m_jnd - jnd_se, ymax = m_jnd + jnd_se), width =0.5) +
  geom_signif(comparisons=list(c("DF", "AF")), annotations="***", y_position = 330, tip_length = 0, vju
  labs(x = " ", y = "Mean JND (ms)") +
  coord_cartesian(ylim = c(0,350))+
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none")

plot_jnd_exp2
```




```
plot_jnd_exp2_box <- ggplot(par_exp2,aes(x=cond, y=jnd, color = cond))+
  geom_boxplot() +
  geom_jitter(position=position_jitter(0.2)) +
  geom_signif(comparisons=list(c("DF", "AF")), annotations="***", y_position = 310, tip_length = 0, vju
  labs(x = " ", y = "Mean JND (ms)") +
  coord_cartesian(ylim = c(80,330))+
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none")

plot_jnd_exp2_box
```



ANOVA

```
Anova_exp2_pse <- ezANOVA(data = par_exp2, dv= pse, wid=NSub, within=.(cond))
```

```
## Warning: Converting "NSub" to factor for ANOVA.
```

```
Anova_exp2_pse
```

```
## $ANOVA
##   Effect DFn DFd        F            p p<.05       ges
## 2   cond   1  14 27.00217 0.0001354841     * 0.2590581
```

```
Anova_exp2_jnd <- ezANOVA(data = par_exp2, dv= jnd, wid=NSub, within=.(cond))
```

```
## Warning: Converting "NSub" to factor for ANOVA.
```

```
Anova_exp2_jnd
```

```
## $ANOVA
##   Effect DFn DFd        F            p p<.05        ges
## 2   cond   1  14 18.34452 0.0007580069     * 0.02965873
```

```
Anova_exp2_slope <- ezANOVA(data = par_exp2, dv= slope, wid=NSub, within=.(cond))
```

```
## Warning: Converting "NSub" to factor for ANOVA.
```

```
Anova_exp2_slope
```

```
## $ANOVA
```

```
##   Effect DFn DFd        F            p p<.05       ges
## 2   cond   1  14 27.01807 0.0001351058     * 0.2768802
```

Experiment 3

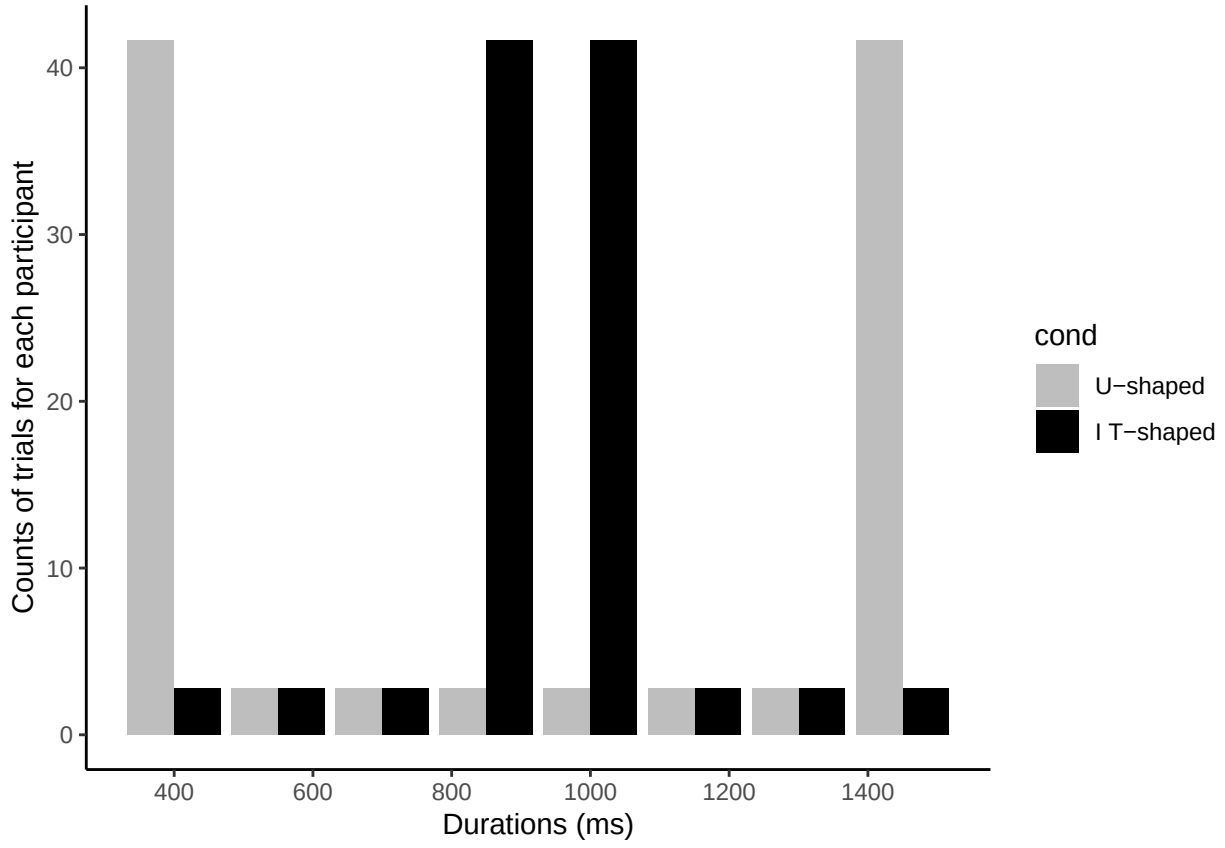
Range Frequency Theory

To understand the pattern of results concerning schifts in the temporal bisection point, We have to check the distribution of stimuli within the sets firstly.

```
sumfreq_exp3_sub1 <-  dat_exp3 %>% filter(NSub ==1) %>% group_by(cond, curDur) %>% summarise(count = n()
```

```
## `summarise()` regrouping output by 'cond' (override with `.groups` argument)
```

```
# rename columns
sumfreq_exp3_sub1$exp <- 'Exp3'
count_exp3 <- sum(sumfreq_exp3_sub1$count)/2
sumfreq_exp3_sub1$freq <- sumfreq_exp3_sub1$count/count_exp3*100

fig_exp3_sub1_dist <- ggplot(data=sumfreq_exp3_sub1, aes(x=curDur, y=freq, fill=cond)) +
  geom_bar(stat="identity", position=position_dodge()) +
  xlab('Durations (ms)') + ylab('Counts of trials for each participant') +
  scale_x_continuous(breaks=seq(0, 1600, 200)) +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +mytheme

fig_exp3_sub1_dist
```



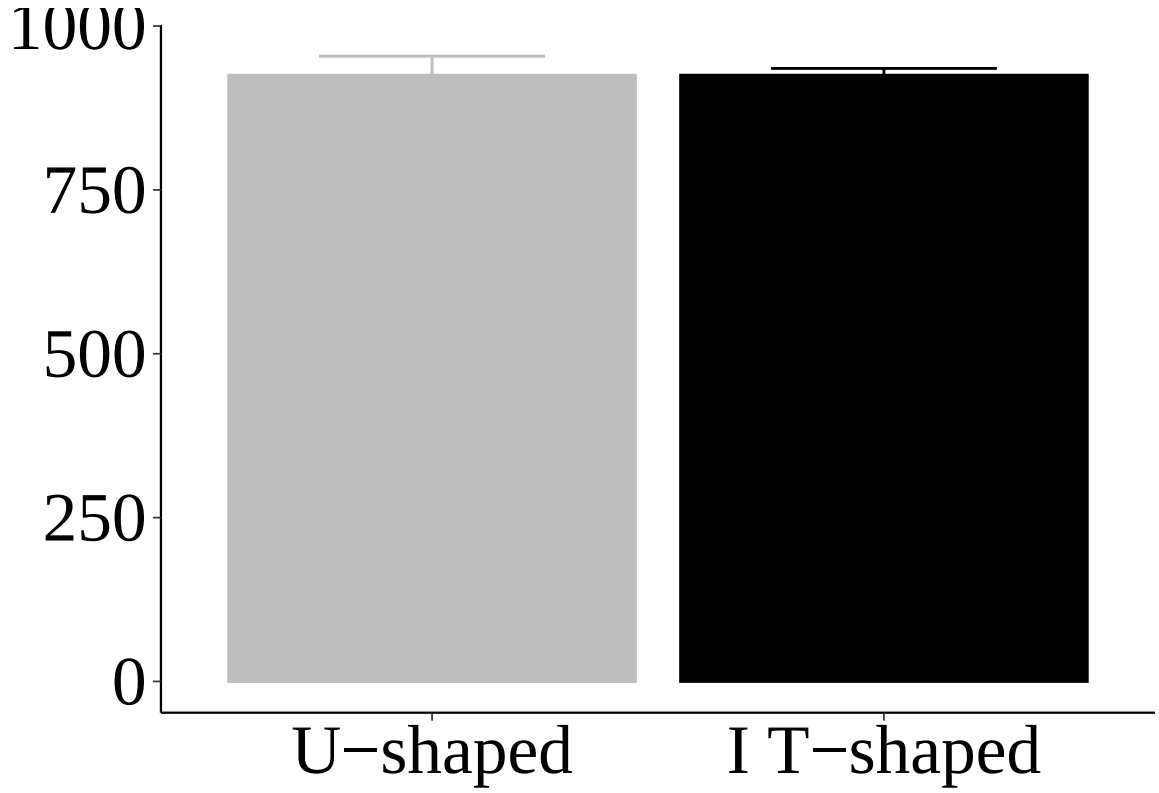
```r
mDist_exp3 <-  dat_exp3 %>% filter(NSub == 1) %>% dplyr::group_by(cond) %>%
  summarise(m_cDur = mean(curDur), n = dplyr::n(),
            cDur_se = sd(curDur)/sqrt(n-1))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
mDist_exp3
```

```
## # A tibble: 2 x 4
##   cond       m_cDur     n cDur_se
##   <fct>       <dbl> <int>   <dbl>
## 1 U-shaped      925   288    29.0
## 2 I T-shaped    925   288    10.3
```

```r
plot_mDist_exp3 <- ggplot(mDist_exp3,aes(cond, m_cDur, color = cond, fill = cond))+
  geom_bar(stat='identity') +
  geom_errorbar(aes(ymin = m_cDur - cDur_se, ymax = m_cDur + cDur_se), width =0.5) +
  labs(x = " ", y = " ") +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none", legend.title = element_blank())+
  theme(panel.grid.major=element_line(colour=NA))+
  theme(axis.text.y = element_text(size=26,colour = "black"),
        axis.text.x   = element_text(size=26, colour = "black"),
        axis.title.y  = element_text(size=26, colour = "black"),
        axis.title.x  = element_text(size=26, colour = "black"),
        panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"))
plot_mDist_exp3
```



Fit psychometric functions

```
fits_exp3 = quickpsy(dat_exp3, x = curDur, k =RP, prob = .5,
                     grouping = .(cond, NSub), thresholds = FALSE)
```

```
fits_exp3_all = quickpsy(dat_exp3, x = curDur, k =RP,
                         grouping = .(cond), thresholds = FALSE)
fits_exp3_all
```

```
## # A tibble: 4 x 5
## # Groups:   cond [2]
##   cond       parn    par parinf parsup
##   <fct>      <fct> <dbl>  <dbl>  <dbl>
## 1 U-shaped   p1     874.   852.   890.
## 2 U-shaped   p2     235.   224.   246.
## 3 I T-shaped p1     864.   859.   871.
## 4 I T-shaped p2     161.   148.   172.
## NULL
```

```
# plot fitted function
plot_fit_exp3 <- plot(fits_exp3_all) +
  labs(x = "Durations (ms)", y = "Proportion of 'Long' responses") +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = c(0.7, 0.2), legend.title = element_blank())
```

```
plot_fit_exp3
```

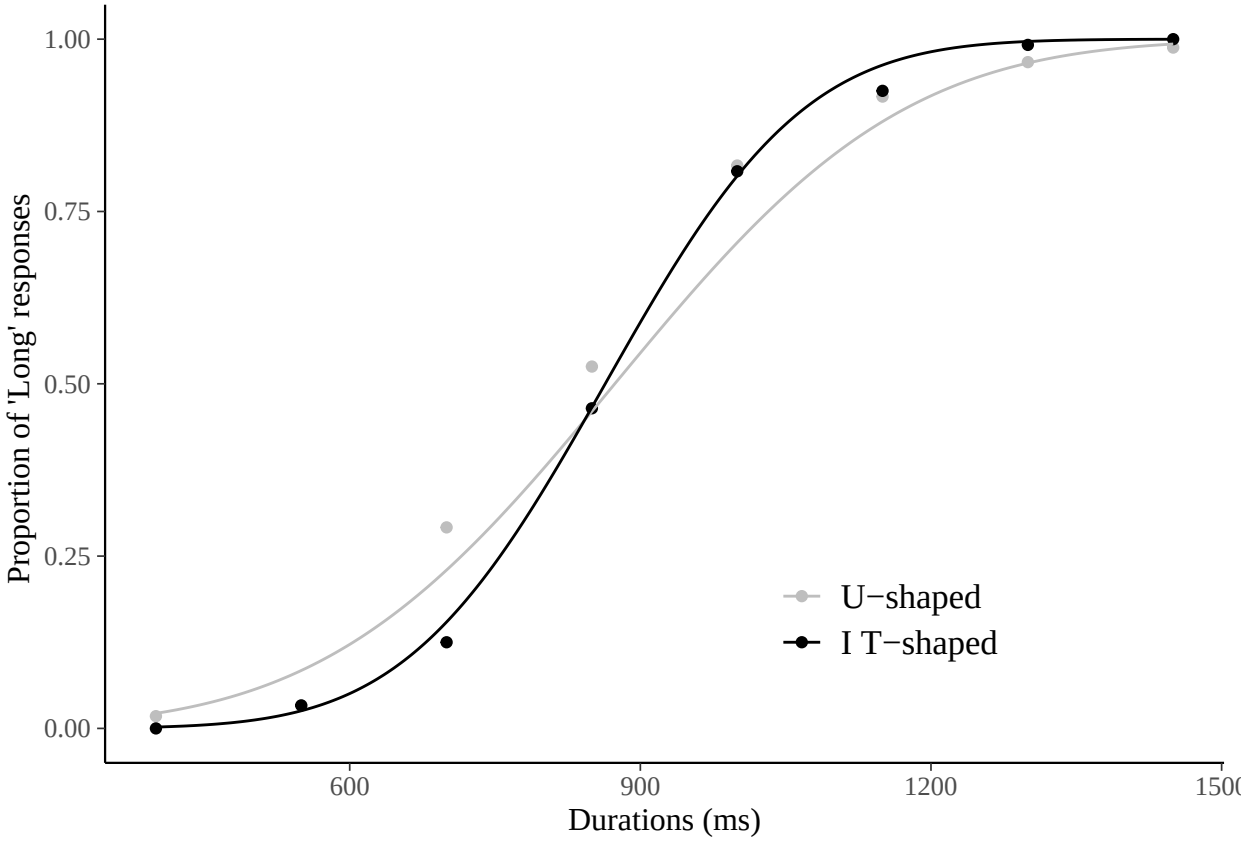


Figure 3: PSE: Bisection task performance in Experiment 3.

List all of the parameter for each subjects

```
par_exp3 = fits_exp3$par %>%
  dplyr::select(-parinf,-parsup) %>%
  spread(parn, par) %>%
  rename(pse =p1, jnd = p2)  # rename columns

par_exp3$slope = 3/exp(par_exp3$pse/1000) #add slope

par_exp3
```

```
## # A tibble: 30 x 5
## # Groups:   cond, NSub [30]
##    cond        NSub   pse   jnd slope
##    <fct>      <int> <dbl> <dbl> <dbl>
##  1 U-shaped       1  950. 113.  1.16
##  2 U-shaped       2  805. 231.  1.34
##  3 U-shaped       3 1184. 121.  0.918
##  4 U-shaped       4  922. 216.  1.19
##  5 U-shaped       5  896. 242.  1.22
##  6 U-shaped       6  739.  86.2 1.43
```

```
##  7 U-shaped       7  938. 227.  1.17
##  8 U-shaped       8  775. 138.  1.38
##  9 U-shaped       9  712. 149.  1.47
## 10 U-shaped      10  991. 215.  1.11
## 11 U-shaped      11  801. 266.  1.35
## 12 U-shaped      12  795. 232.  1.35
## 13 U-shaped      13  880. 193.  1.24
## 14 U-shaped      14  824. 348.  1.32
## 15 U-shaped      15  870. 135.  1.26
## 16 I T-shaped     1  928.  85.5 1.19
## 17 I T-shaped     2  872. 133.  1.25
## 18 I T-shaped     3  847.  99.3 1.29
## 19 I T-shaped     4  931. 124.  1.18
## 20 I T-shaped     5  983. 243.  1.12
## 21 I T-shaped     6  868.  97.3 1.26
## 22 I T-shaped     7 1057. 152.  1.04
## 23 I T-shaped     8  921. 103.  1.19
## 24 I T-shaped     9  830. 121.  1.31
## 25 I T-shaped    10  896. 126.  1.22
## 26 I T-shaped    11  813. 205.  1.33
## 27 I T-shaped    12  678. 182.  1.52
## 28 I T-shaped    13  699.  85.2 1.49
## 29 I T-shaped    14  753. 158.  1.41
## 30 I T-shaped    15  879. 110.  1.25
```

```
mpars_exp3 = par_exp3%>%
  group_by(cond) %>%
  summarise(m_pse = mean(pse), m_jnd = mean(jnd), m_slope = mean(slope), n = n(),pse_se = sd(pse)/sqrt(n
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
mpars_exp3
```

```
## # A tibble: 2 x 8
##   cond       m_pse m_jnd m_slope     n pse_se jnd_se slope_se
##   <fct>      <dbl> <dbl>   <dbl> <int>  <dbl>  <dbl>    <dbl>
## 1 U-shaped    872.  194.    1.26    15   31.6   18.6   0.0376
## 2 I T-shaped  864.  135.    1.27    15   27.0   12.2   0.0347
```

##Export for JASP

List all of the parameters for Exp3 and export data for JASP

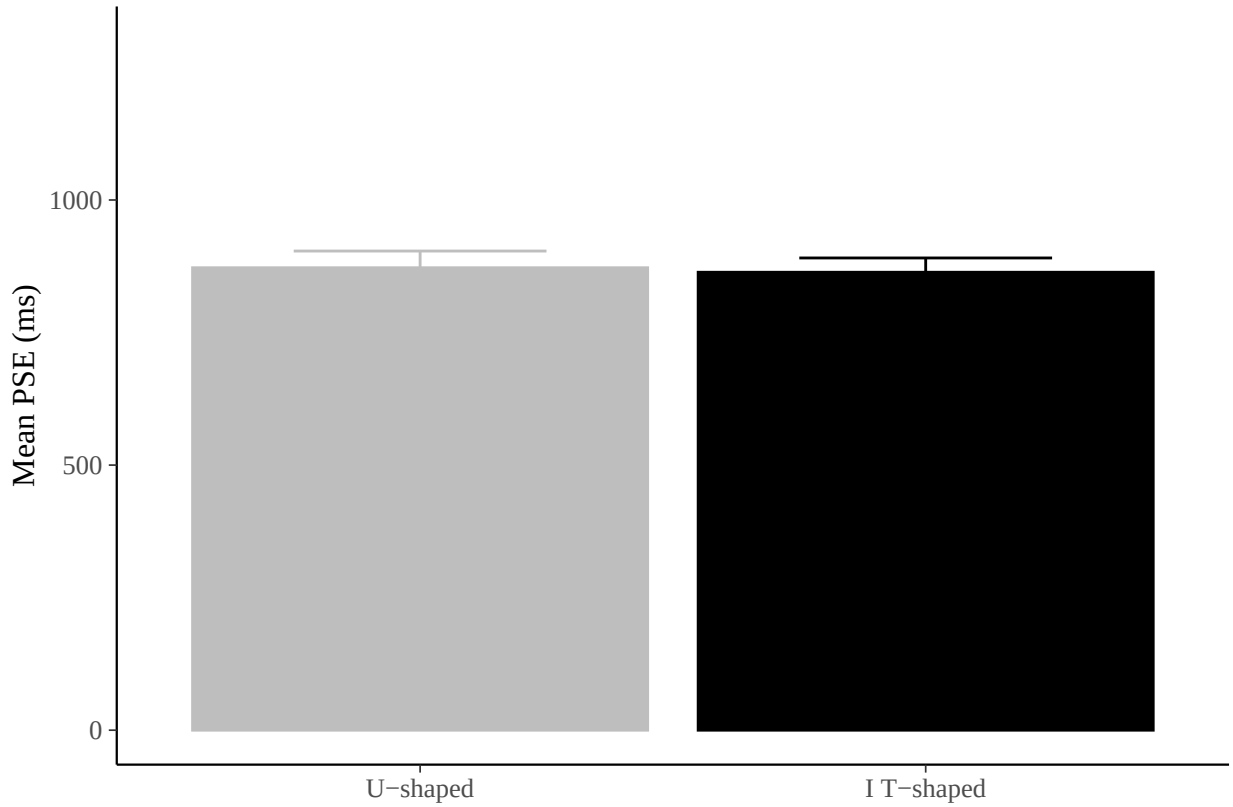
```
exp3_jasp <- par_exp3%>%
  unite(pse.jnd.slope, pse, jnd, slope)%>%
  spread(cond, pse.jnd.slope)%>%
  separate(`U-shaped`, c("U_pse", "U_jnd", "U_slope"), sep = "_")%>%
  separate(`I T-shaped`, c("B_pse", "B_jnd", "B_slope"), sep = "_")

exp3_jasp
```

```
## # A tibble: 15 x 7
## # Groups:   NSub [15]
##     NSub U_pse      U_jnd      U_slope      B_pse      B_jnd     B_slope
##    <int> <chr>      <chr>      <chr>        <chr>      <chr>     <chr>
##  1     1 950.37144~ 113.45943~ 1.159792196~ 927.60576~ 85.47959~ 1.1864984~
##  2     2 805.37337~ 231.32833~ 1.340763079~ 872.24309~ 133.3283~ 1.2540385~
```

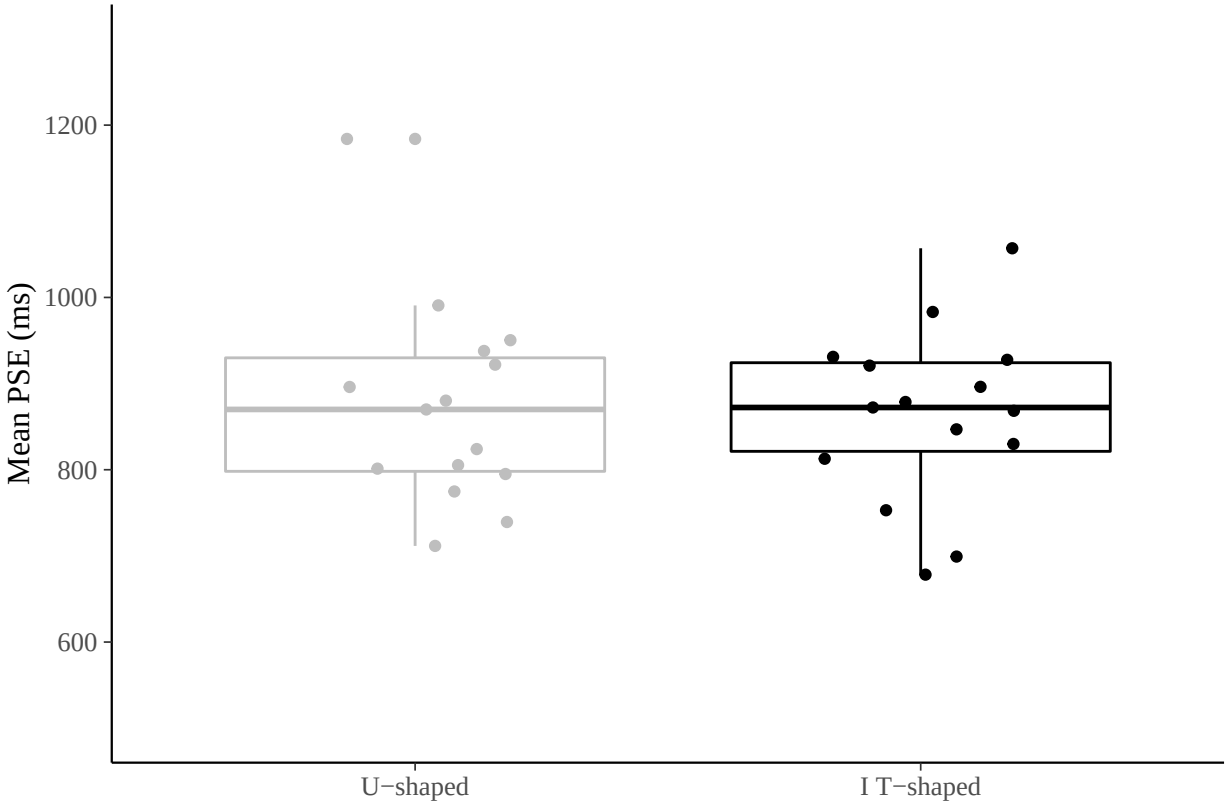
```
##  3     3 1183.9743~ 120.64725~ 0.918179762~ 846.93929~ 99.30002~ 1.2861753~
##  4     4 921.95193~ 215.88884~ 1.193225747~ 931.08489~ 124.0465~ 1.1823776~
##  5     5 896.07916~ 242.26542~ 1.224500645~ 983.13667~ 243.3646~ 1.1224071~
##  6     6 739.37513~ 86.162249~ 1.432236417~ 868.37765~ 97.26130~ 1.2588953~
##  7     7 937.80392~ 227.27937~ 1.174459879~ 1057.0570~ 151.8832~ 1.0424307~
##  8     8 774.81078~ 138.39203~ 1.382372883~ 920.85843~ 102.8389~ 1.1945312~
##  9     9 711.59143~ 149.24600~ 1.472587204~ 829.95280~ 121.1778~ 1.3082096~
## 10    10 990.69492~ 215.27685~ 1.113955693~ 896.20004~ 126.0216~ 1.2243526~
## 11    11 801.29480~ 266.02018~ 1.346242640~ 812.79508~ 205.4282~ 1.3308491~
## 12    12 795.03191~ 232.16155~ 1.354700475~ 678.19405~ 181.9216~ 1.5225982~
## 13    13 880.20132~ 192.69018~ 1.244098238~ 699.19913~ 85.15340~ 1.4909494~
## 14    14 823.95528~ 347.57990~ 1.316079185~ 753.01857~ 157.5293~ 1.4128284~
## 15    15 869.99661~ 134.60638~ 1.256858898~ 878.67685~ 110.3727~ 1.2459962~
```

plot PSE

```
plot_pse_exp3 <- ggplot(mpars_exp3,aes(cond, m_pse, color = cond, fill = cond))+
  geom_bar(stat='identity') +
  geom_errorbar(aes(ymin = m_pse - pse_se, ymax = m_pse + pse_se), width =0.5) +
  coord_cartesian(ylim = c(0,1300)) +
  labs(x = " ", y = "Mean PSE (ms)") +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none", legend.title = element_blank())
plot_pse_exp3
```

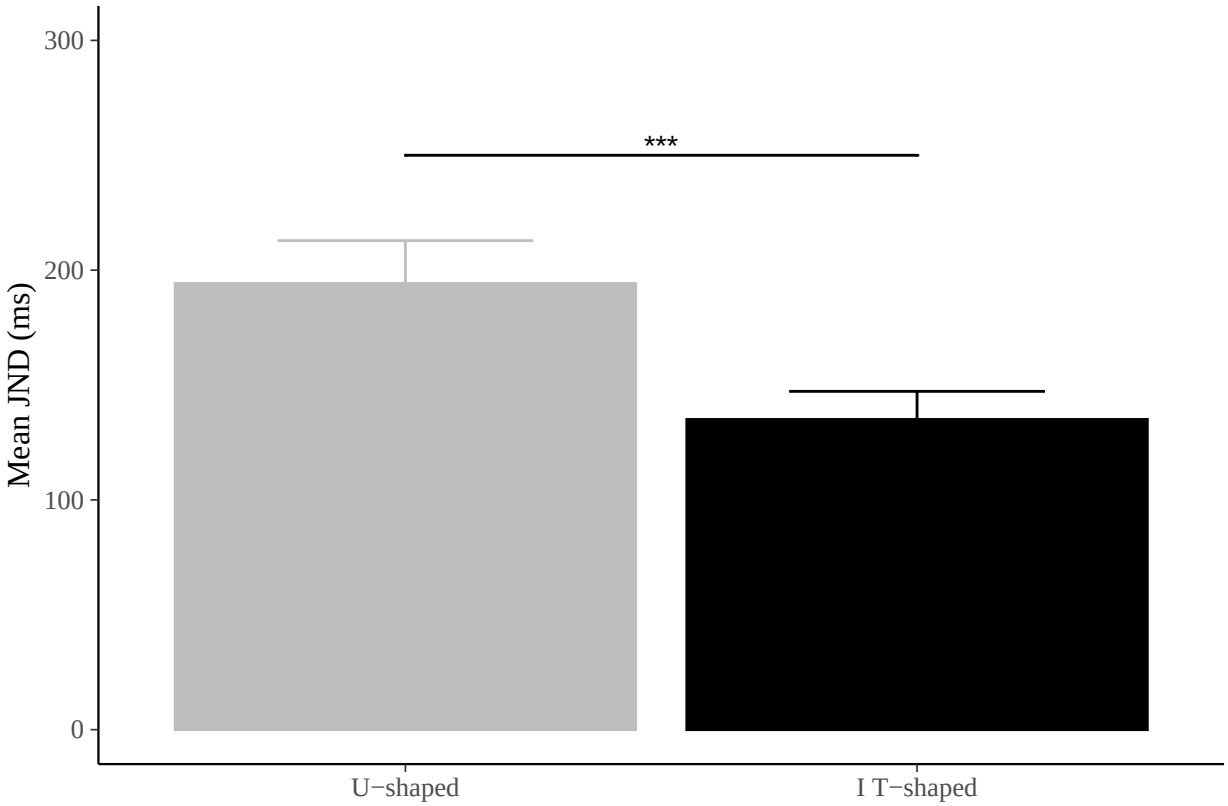


```
plot_pse_exp3_box <- ggplot(par_exp3,aes(x=cond, y=pse, color = cond))+
  geom_boxplot() +
  geom_jitter(position=position_jitter(0.2))+
  coord_cartesian(ylim = c(500,1300)) +
  labs(x = " ", y = "Mean PSE (ms)") +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none", legend.title = element_blank())
plot_pse_exp3_box
```

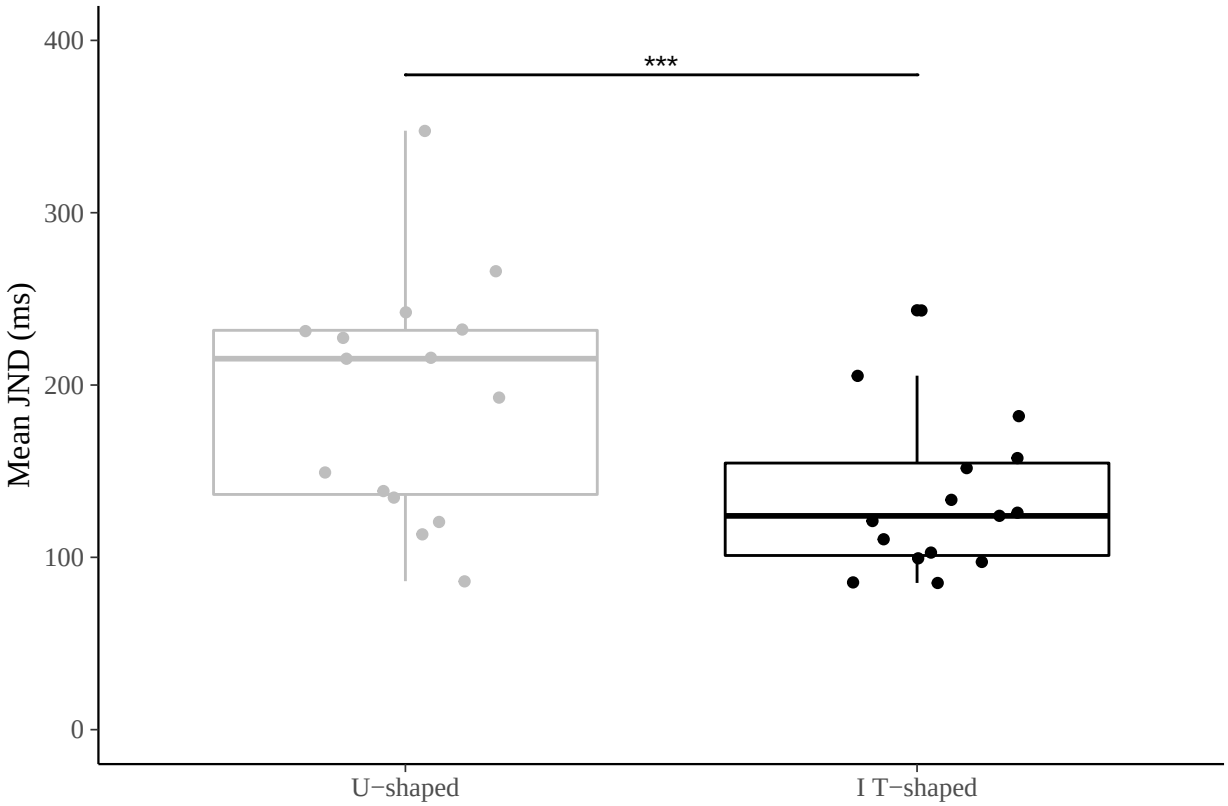


plot JND

```
plot_jnd_exp3 <- ggplot(mpars_exp3,aes(cond, m_jnd, color = cond, fill = cond))+
  geom_bar(stat='identity') +
  geom_errorbar(aes(ymin = m_jnd - jnd_se, ymax = m_jnd + jnd_se), width =0.5) +
  coord_cartesian(ylim = c(0,300))+
  geom_signif(comparisons=list(c("U-shaped", "I T-shaped")), annotations="***", y_position = 250, tip_l
  labs(x = " ", y = "Mean JND (ms)") +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none")
plot_jnd_exp3
```




```
plot_jnd_exp3_box <- ggplot(par_exp3,aes(x=cond, y=jnd, color = cond))+
  geom_boxplot() +
  geom_jitter(position=position_jitter(0.2))+
  coord_cartesian(ylim = c(0,400))+
  geom_signif(comparisons=list(c("U-shaped", "I T-shaped")), annotations="***", y_position = 380, tip_l
  labs(x = " ", y = "Mean JND (ms)") +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none")
plot_jnd_exp3_box
```



ANOVA

```
Anova_exp3_pse<- ezANOVA(data = par_exp3, dv= pse, wid=NSub, within=.(cond))
```

```
## Warning: Converting "NSub" to factor for ANOVA.
```

```
Anova_exp3_pse
```

```
## $ANOVA
##   Effect DFn DFd          F         p p<.05        ges
## 2   cond   1  14 0.05990591 0.8101957       0.00159169
```

```
Anova_exp3_jnd<- ezANOVA(data = par_exp3, dv= jnd, wid=NSub, within=.(cond))
```

```
## Warning: Converting "NSub" to factor for ANOVA.
```

```
Anova_exp3_jnd
```

```
## $ANOVA
##   Effect DFn DFd        F            p p<.05       ges
## 2   cond   1  14 19.87924 0.0005403918     * 0.2123367
```

```
Anova_exp3_slope<- ezANOVA(data = par_exp3, dv= slope, wid=NSub, within=.(cond))
```

```
## Warning: Converting "NSub" to factor for ANOVA.
```

```
Anova_exp3_slope
```

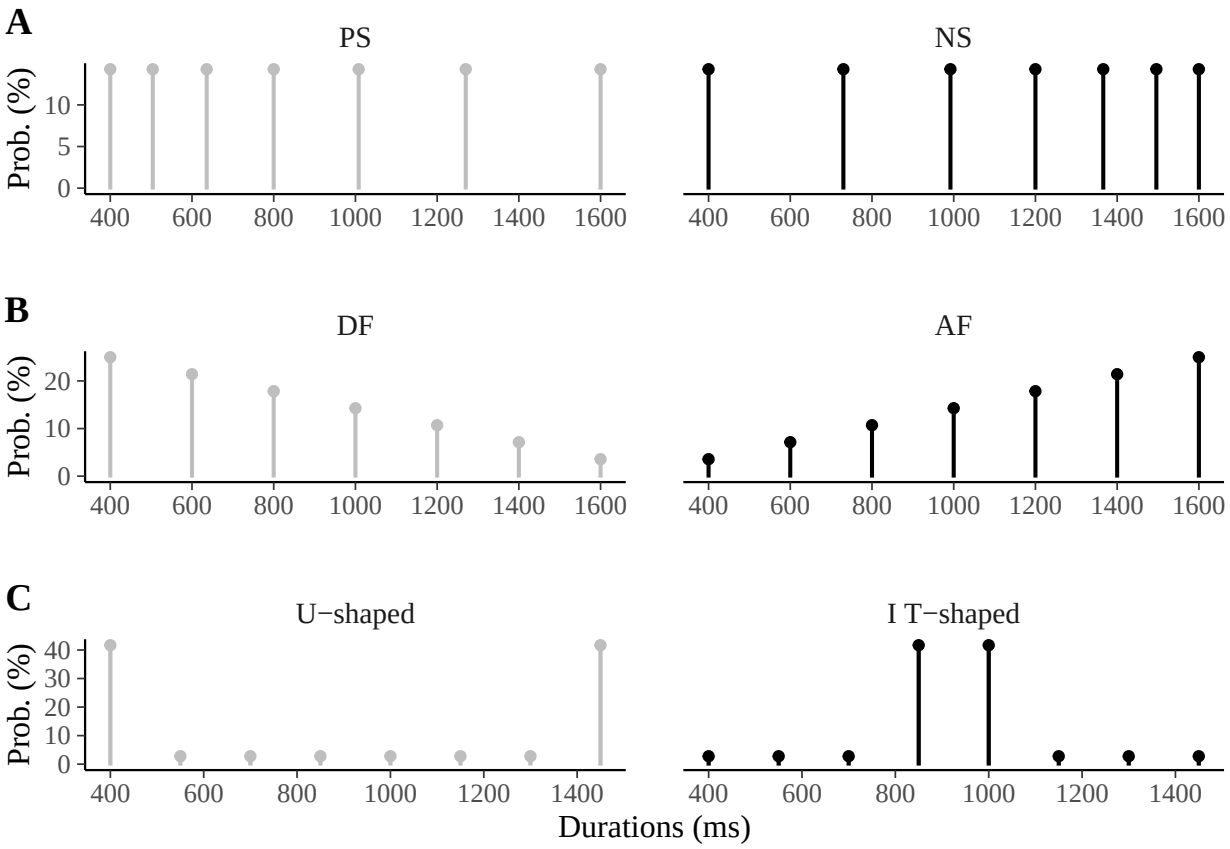
```
## $ANOVA
```

```
##   Effect DFn DFd          F         p p<.05         ges
## 2   cond   1  14 0.04422096 0.8364728       0.001150098
```

figures in manuscript

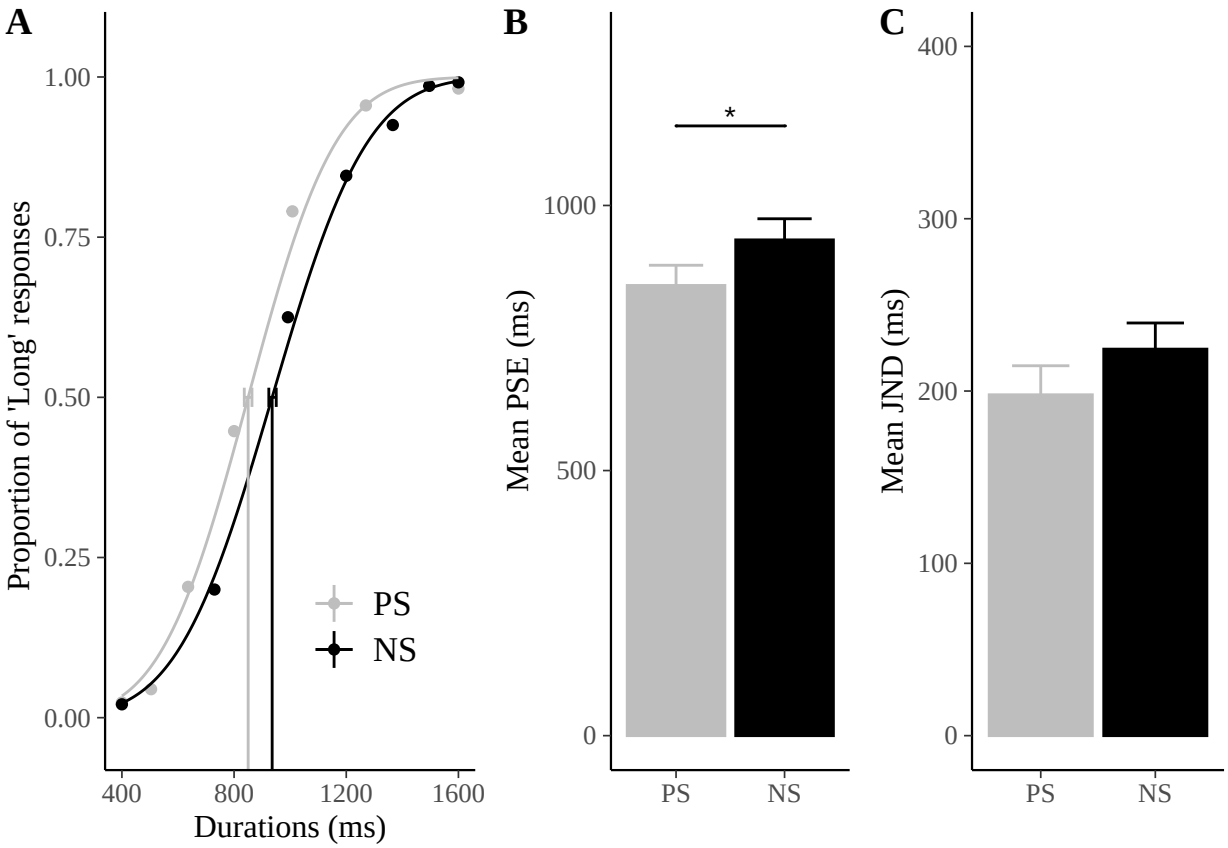
plot figure 1 in manuscript

```
fig_exp1_sub1_dist2 <- ggplot(data=sumfreq_exp1_sub1, aes(x=curDur, y=freq, fill=cond, color = cond)) +
  geom_bar(stat="identity", position=position_dodge(),width = 3) +
  geom_point()+
  xlab('') + ylab('Prob. (%)') +
  scale_x_continuous(breaks=seq(0, 1600, 200)) +
  scale_y_continuous(breaks=seq(0, 50, 5)) +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none", legend.title = element_blank()) + facet_wrap(~cond)

fig_exp2_sub1_dist_2 <- ggplot(data=sumfreq_exp2_sub1, aes(x=curDur, y=freq, fill=cond, color = cond)) +
  geom_bar(stat="identity", position=position_dodge(), width = 3) +
  geom_point()+
  scale_x_continuous(breaks=seq(0, 1600, 200)) +
  scale_y_continuous(breaks=seq(0, 50, 10)) +
  xlab('') + ylab('Prob. (%)') +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = 'none', legend.title = element_blank()) + facet_wrap(~cond)

fig_exp3_sub1_dist2 <- ggplot(data=sumfreq_exp3_sub1, aes(x=curDur, y=freq, fill=cond, color = cond)) +
  geom_bar(stat="identity", position=position_dodge(), width = 3) +
  geom_point()+
  xlab('Durations (ms)') + ylab('Prob. (%)') +
  scale_x_continuous(breaks=seq(0, 1600, 200)) +
  scale_y_continuous(breaks=seq(0, 50, 10)) +
  scale_color_manual(values = c('gray','black')) +
  scale_fill_manual(values = c('gray','black')) +
  theme(legend.position = "none", legend.title = element_blank())+ facet_wrap(~cond)

plot_grid(fig_exp1_sub1_dist2, fig_exp2_sub1_dist_2, fig_exp3_sub1_dist2,  ncol = 1, labels = c("A", "B
```



plot Figure 2 in manuscript

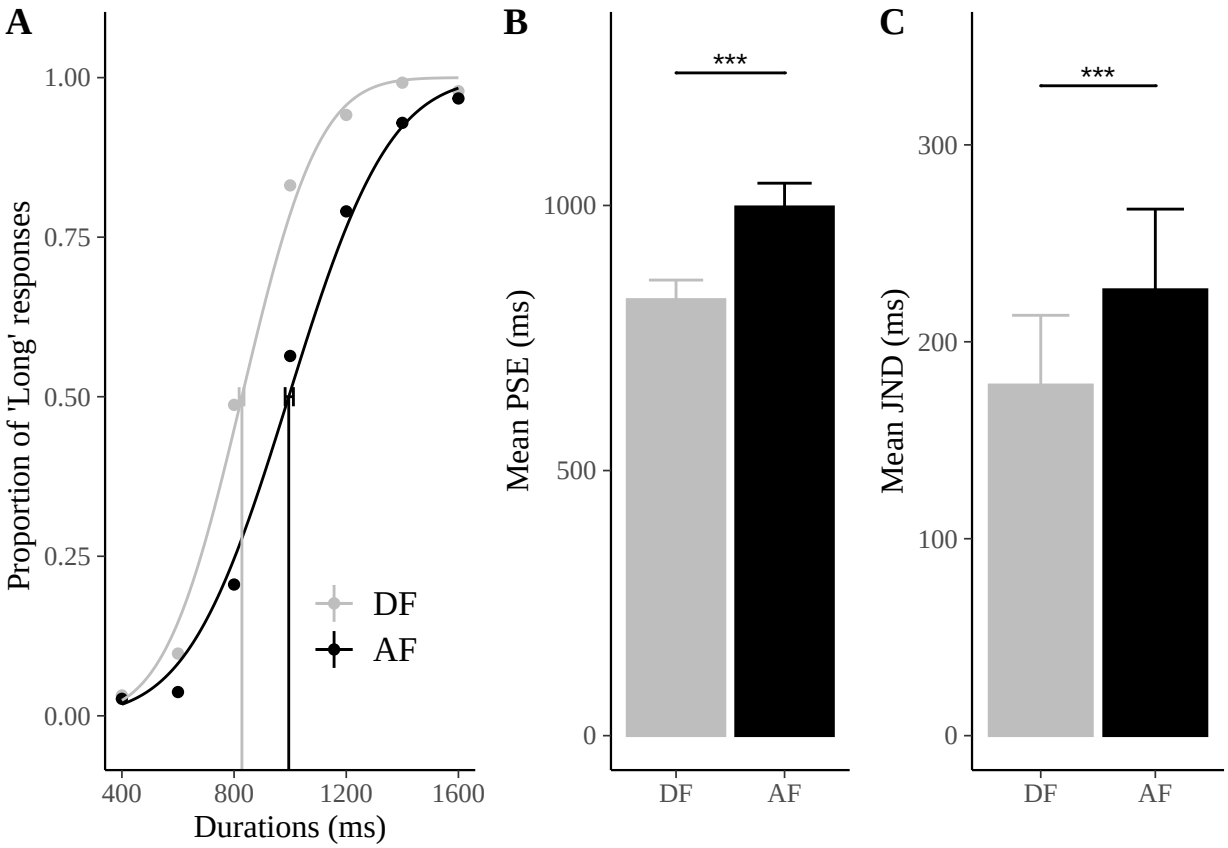
```
plot_grid(plot_fit_exp1, plot_pse_exp1, plot_jnd_exp1,  nrow = 1, labels = c("A", "B", "C"), rel_widths
```




```
fig2 <- plot_grid(plot_fit_exp1, plot_pse_exp1_box, plot_jnd_exp1_box,  nrow = 1, labels = c("A", "B", "
```

plot Figure 3 in manuscript

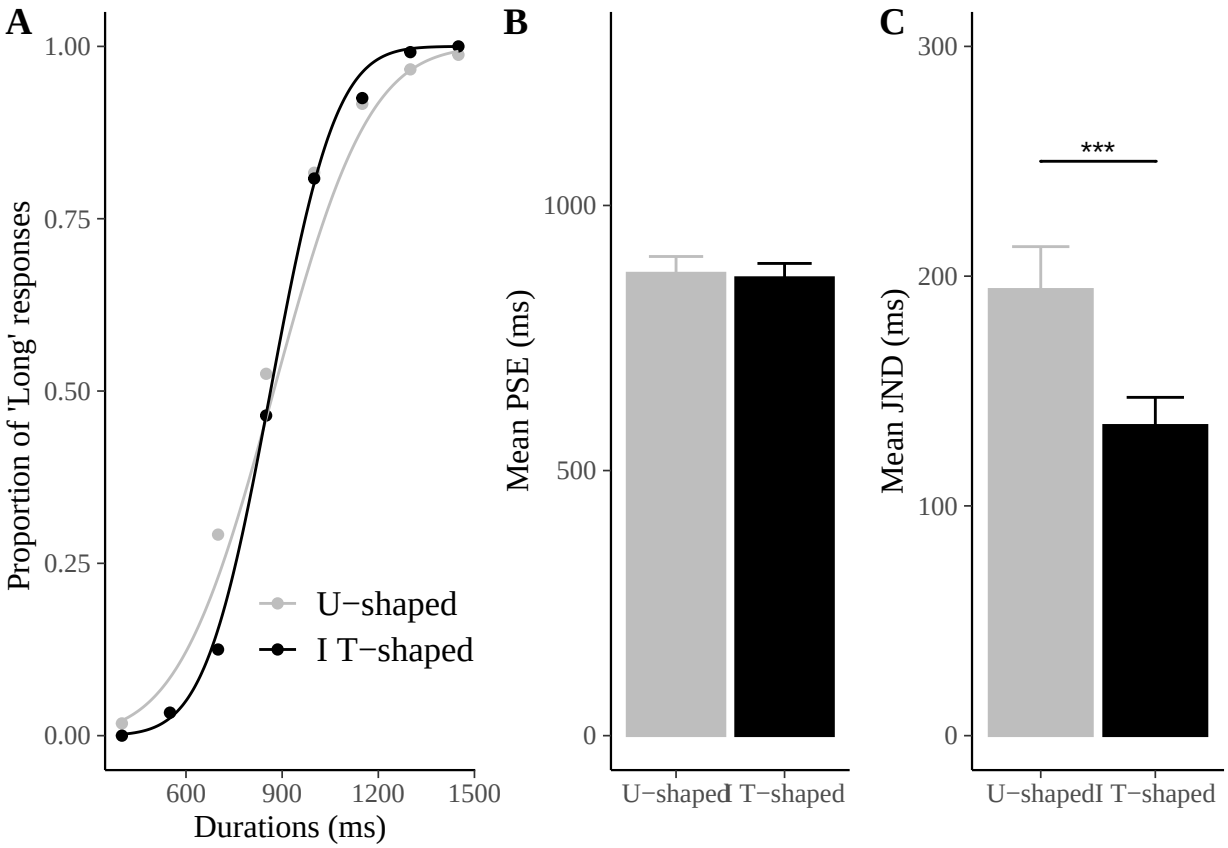
```
plot_grid(plot_fit_exp2, plot_pse_exp2, plot_jnd_exp2,  nrow = 1, labels = c("A", "B", "C"), rel_widths
```




```
fig3 <- plot_grid(plot_fit_exp2, plot_pse_exp2_box, plot_jnd_exp2_box,  nrow = 1, labels = c("A", "B", "
```

plot Figure 4 in manuscript

```
plot_grid(plot_fit_exp3, plot_pse_exp3, plot_jnd_exp3,  nrow = 1, labels = c("A", "B", "C"), rel_widths
```




```
fig4 <- plot_grid(plot_fit_exp3, plot_pse_exp3_box, plot_jnd_exp3_box,  nrow = 1, labels = c("A", "B", "
```

Model Results

load the Model result

```
#customize theme
theme_new <- theme_bw() +
  theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line=element_line(colour="black"),
        strip.background = element_rect(color = "white", fill = "white"),
        panel.grid = element_blank())

models <- c('model1', 'model2','model3', 'model4','model5')
expDat <- list(dat_exp1, dat_exp2, dat_exp3)
#load the data and remove the outliers

par_observed <- rbind(rbind(par_exp1, par_exp2), par_exp3)
par_observed <- par_observed %>% filter(!((NSub == 10|NSub == 6 ) & (cond == 'AF'| cond =='DF')))%>% fi
modelresults<- read.csv("../data/modelresults.csv")  %>% filter(!((NSub == 10|NSub == 6) & (cond == 'AF
```

```
displayModellist <- c("Bisection Model",  "Spacing Model", "Ensemble Mean",  "Two-stage Ensemble Mean",
#combine oberserved and predicated PSEs and JNDs
```

```
par_comb = left_join(modelresults, par_observed, by = c('NSub', 'cond'))
par_comb$errpse <- par_comb$PSEmap-par_comb$pse
par_comb$errjnd <- par_comb$JNDmap-par_comb$jnd
par_comb$model = factor(par_comb$model, labels = displayModellist)
head(par_comb)
```

```
##   X xmean xmeanhat   alphaMAP  betaMAP    PSEmap    JNDmap NSub
## 1 2  1000     1000 -0.2613433 5.835077 1000.0448 0.1882773    2
## 2 3  1000     1000  1.2657652 3.938268  999.6786 0.2789582    3
## 3 4  1000     1000  2.2730541 4.520620  999.4972 0.2430225    4
## 4 5  1000     1000  2.0103210 3.275906  999.3863 0.3353614    5
## 5 6  1000     1000  0.6195048 4.613632  999.8657 0.2381231    6
## 6 7  1000     1000  0.8615030 5.585518  999.8458 0.1966894    7
##             model cond  Exp       pse      jnd    slope     errpse
## 1 Bisection Model   PS exp1 1032.5088 184.9694 2.493389 -32.464028
## 2 Bisection Model   PS exp1  839.7529 317.3121 2.184310 159.925653
## 3 Bisection Model   PS exp1  678.4811 193.9217 2.471167 321.016032
## 4 Bisection Model   PS exp1  661.1666 246.9267 2.343594 338.219735
## 5 Bisection Model   PS exp1  996.0497 299.4824 2.223605   3.815999
## 6 Bisection Model   PS exp1  917.5433 220.5750 2.406172  82.302507
##      errjnd
## 1 -184.7811
## 2 -317.0331
## 3 -193.6787
## 4 -246.5913
## 5 -299.2443
## 6 -220.3783
```

```
#calculate AIC and BIC
numParam <- 3
mpar_comb <- par_comb %>% dplyr::group_by(model, cond) %>%
  dplyr::summarise(sumerrpse=sum((errpse)^2), n=n(),
                   sumerrjnd=sum((errjnd)^2)) %>%
  dplyr::group_by(model, cond) %>%
  summarise(aicPse= 2 * numParam +  n*log(sumerrpse/n),
            bicPse = n*log(sumerrpse/n) + numParam*log(n),
            aicJnd= 2 * numParam +  n*log(sumerrjnd/n),
            bicJnd = n*log(sumerrjnd/n) + numParam*log(n), )
```

```
## `summarise()` regrouping output by 'model' (override with `.groups` argument)
## `summarise()` regrouping output by 'model' (override with `.groups` argument)
```

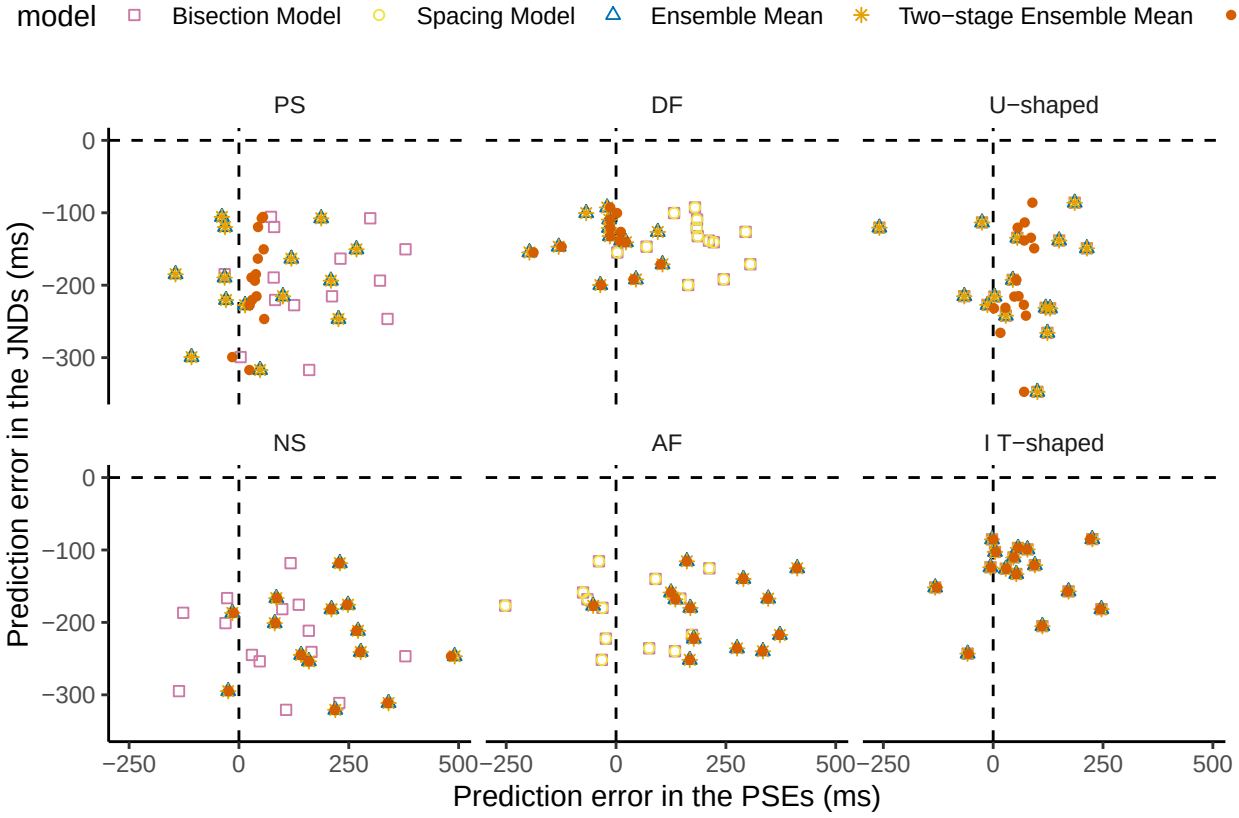
```
### show the mean in each model
m_par <- modelresults %>% group_by(model, cond, NSub) %>%
  summarise(xmean= mean(xmean), xmeanhat= mean(xmeanhat))  %>%
  group_by(model, cond) %>%
  summarise(xmean= mean(xmean), xmeanhat= mean(xmeanhat))
```

```
## `summarise()` regrouping output by 'model', 'cond' (override with `.groups` argument)
## `summarise()` regrouping output by 'model' (override with `.groups` argument)
```

plot figure 7 in manuscript

```
par_comb <- arrange(transform(par_comb,
                              cond=factor(cond,levels=c("PS","DF","U-shaped","NS","AF","I T-shaped"))),
```

```
cbp1 <- c( "#CC79A7", "#F0E442", "#0072B2", "#E69F00", "#D55E00")
plt_ErrorScatter = ggplot(par_comb, aes(errpse, errjnd, color = model, shape = model)) +
  geom_hline(yintercept = 0, linetype='dashed')+ geom_vline(xintercept = 0, linetype='dashed')+
  geom_point() +
  xlab('Prediction error in the PSEs (ms)')+ ylab('Prediction error in the JNDs (ms)')+
  scale_color_manual(values = cbp1) +
  scale_shape_manual(values=c(0, 1, 2, 8, 16))+
  facet_wrap(~cond)+
  theme_new+ theme(legend.position = 'top')

plt_ErrorScatter
```



plot figure 6 in manuscript

```
m_parcomb <- par_comb %>% dplyr::group_by(model, cond) %>%
  dplyr::summarise(mPSEmap=mean(PSEmap), pse = mean(pse), mJNDmap=mean(JNDmap),jnd = mean(jnd),
                   merrpse =mean(errpse), merrjnd = mean(errjnd),n=n(),
                   PSEmap_se = sd(PSEmap)/sqrt(n-1), JNDmap_se = sd(JNDmap)/sqrt(n-1),
                   errpse_se =sd(errpse)/sqrt(n-1), errjnd_se =sd(errjnd)/sqrt(n-1))
```

```
## `summarise()` regrouping output by 'model' (override with `.groups` argument)
```

```
m_par_observed <- par_observed %>% dplyr::group_by(cond) %>%
  dplyr::summarise(mpse=mean(pse), mjnd= mean(jnd), n=n(),
                   pse_se =sd(pse)/sqrt(n-1), jnd_se =sd(jnd)/sqrt(n-1))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
m_par_observed$mPSEmap<-m_par_observed$mpse
m_par_observed$mJNDmap<-m_par_observed$mjnd

m_parcomb2 <- arrange(transform(m_parcomb,
             cond=factor(cond,levels=c("PS","DF","U-shaped","NS","AF","I T-shaped"))),cond)


fig_estimation = ggplot(m_parcomb2, aes(x=mPSEmap, y=mJNDmap, color = model, shape = model)) +
  geom_point(data =m_par_observed, aes(x = mpse, y =mjnd, size=2), width =2, shape =16, alpha = 0.5,  co
  geom_point(data = par_comb, aes(x = pse, y =jnd), alpha = 0.25, color = 'lightgray', shape=16) +
  geom_point() +
  geom_errorbar(aes(ymax=mJNDmap+JNDmap_se, ymin=mJNDmap-JNDmap_se), size=0.6)+
  geom_errorbarh(aes(xmax=mPSEmap+PSEmap_se, xmin=mPSEmap-PSEmap_se),size=0.6)+
  ylab('Observed JNDs (ms)')+xlab('Observed PSEs (ms)') +
  scale_color_manual(values = cbp1) +
  scale_shape_manual(values=c(0, 1, 2, 8, 16))+
  theme_new+ theme(legend.position = 'top') +
  facet_wrap(.~cond,  ncol=3)+
  scale_size(guide = 'none')
```

```
## Warning: Ignoring unknown parameters: width
```

```
fig_estimation
```