ModelReportAll

Xiuna Zhu

10/2/2022

```
knitr::opts_chunk$set(echo = TRUE)
```

Load the packages

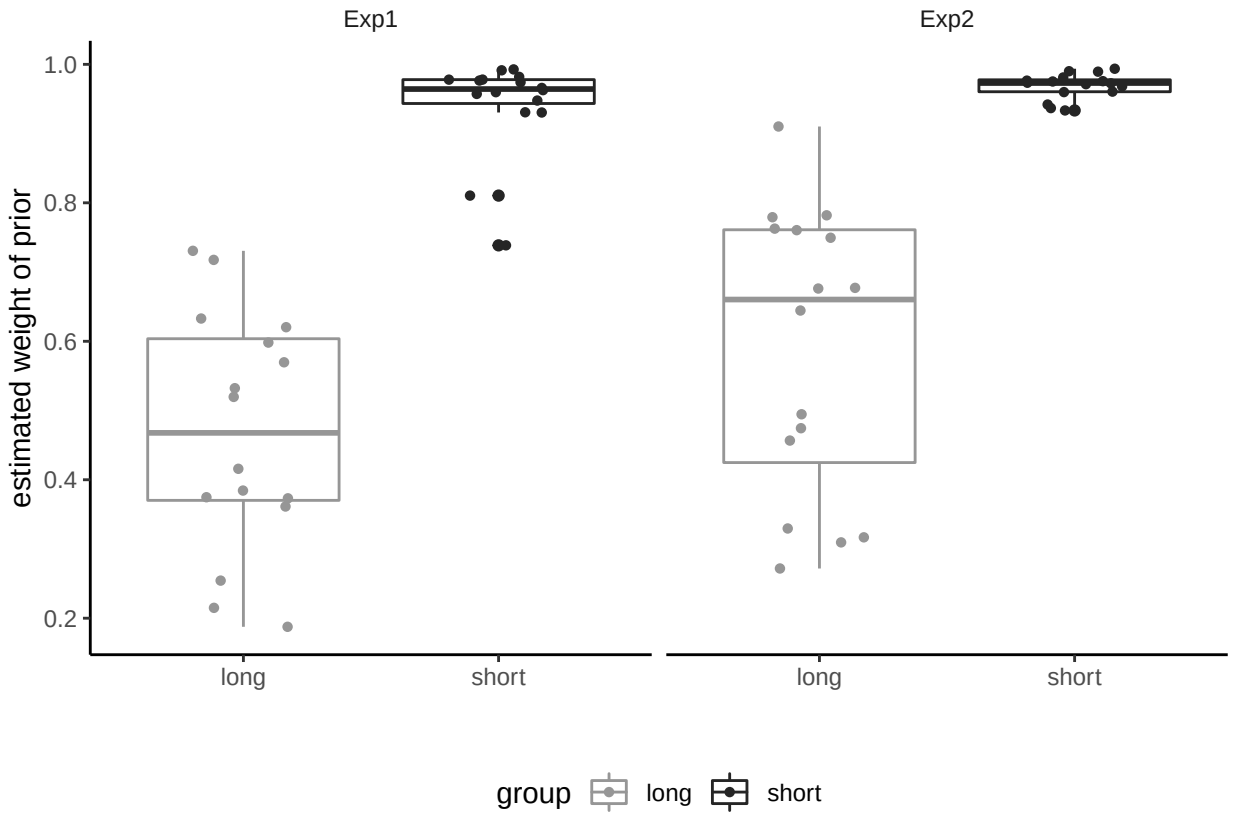
1 Baseline model results (for BR condition)

1.1 load baseline model results

1.2 plot predicted reproduction based on baseline model

1.3 plot weight of local prior based on baseline model

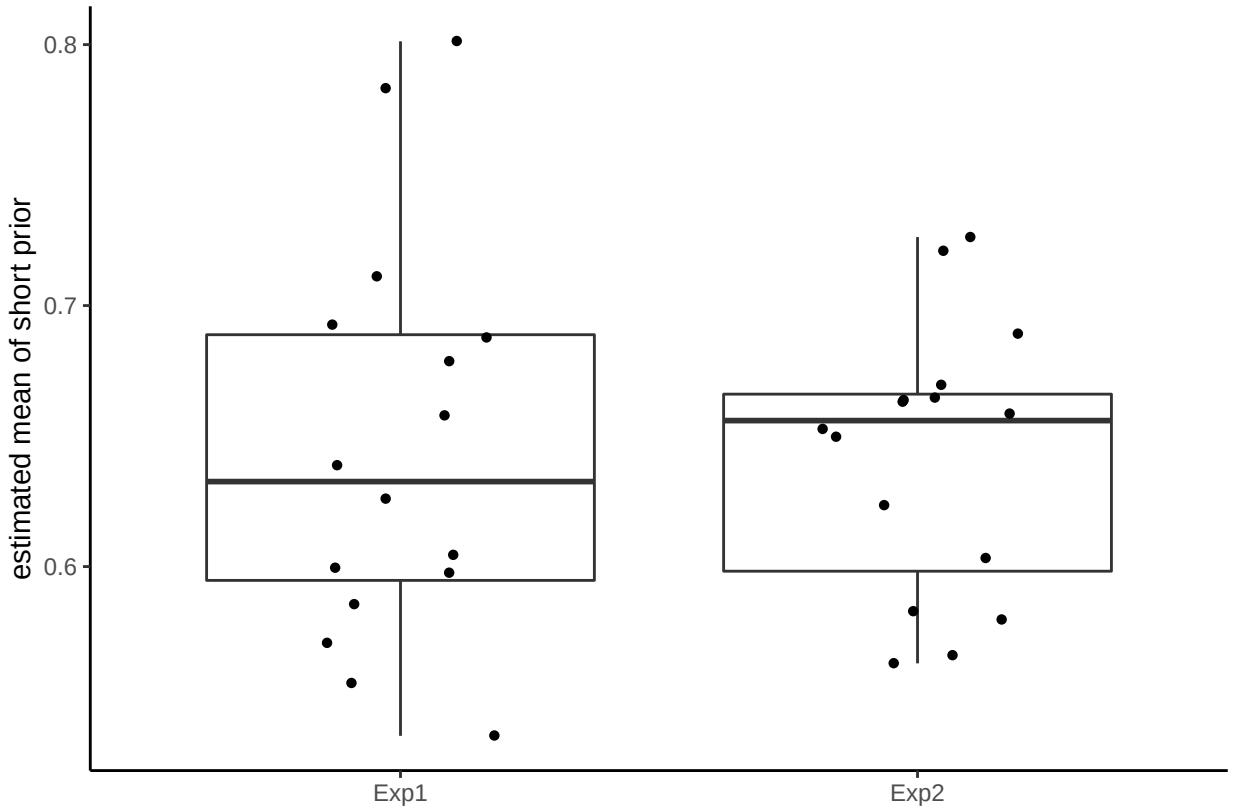
```
fig_wp_BR_subj = ggplot(m_parameter_BR_sub, aes(group, m_wp, color = group)) +
  geom_boxplot(position = position_dodge()) +
  geom_jitter(shape=16, position=position_jitter(0.2))+
  facet_wrap(~Exp)+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background = element_blank()) + # remove subtitle background
  labs(x = ' ', y = 'estimated weight of prior')+
  theme(legend.position='bottom')

fig_wp_BR_subj
```



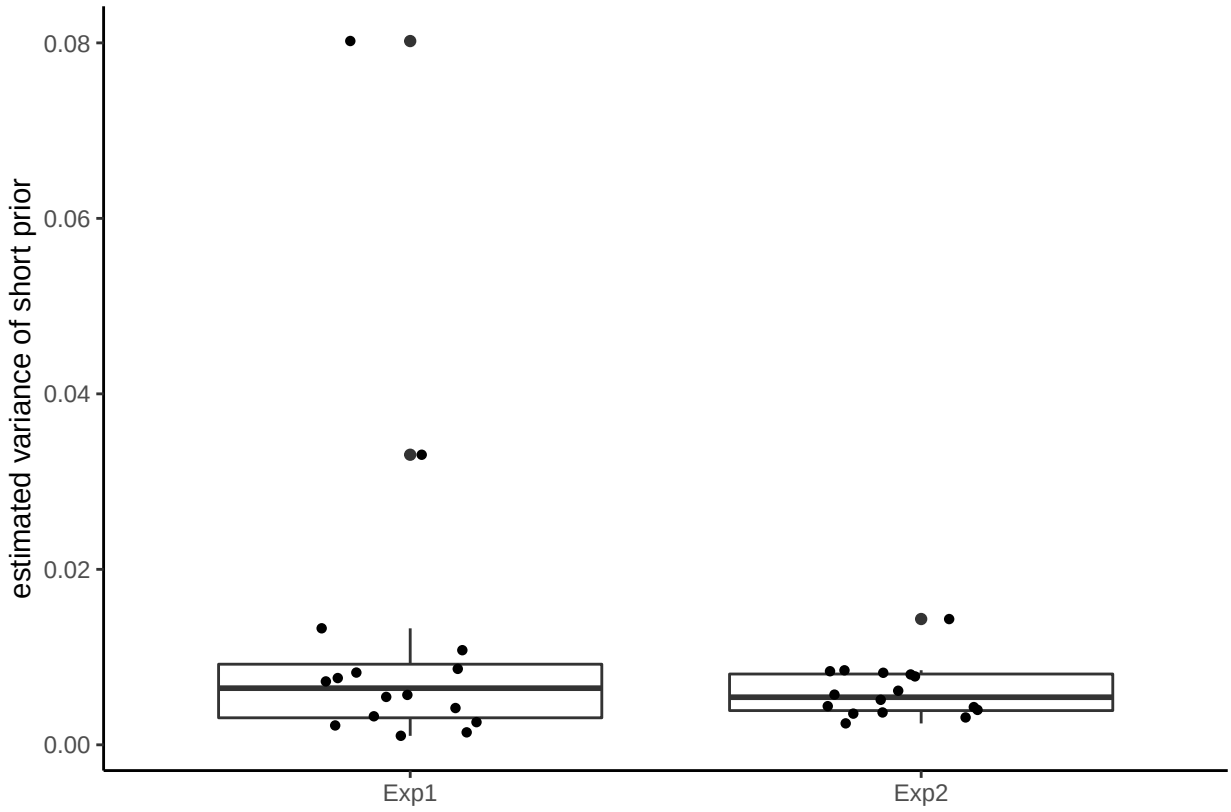
load the estimated parameters from linear baseline model on BR session

1.4 plot estimated short prior from baseline model

```
ggplot(AllDat_Bayparlist_BR, aes(Exp, mu_p_s)) +
  geom_boxplot(position = position_dodge()) +
  geom_jitter(shape=16, position=position_jitter(0.2))+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background = element_blank()) + # remove subtitle background
  labs(x = ' ', y = 'estimated mean of short prior')+
  theme(legend.position='bottom')
```

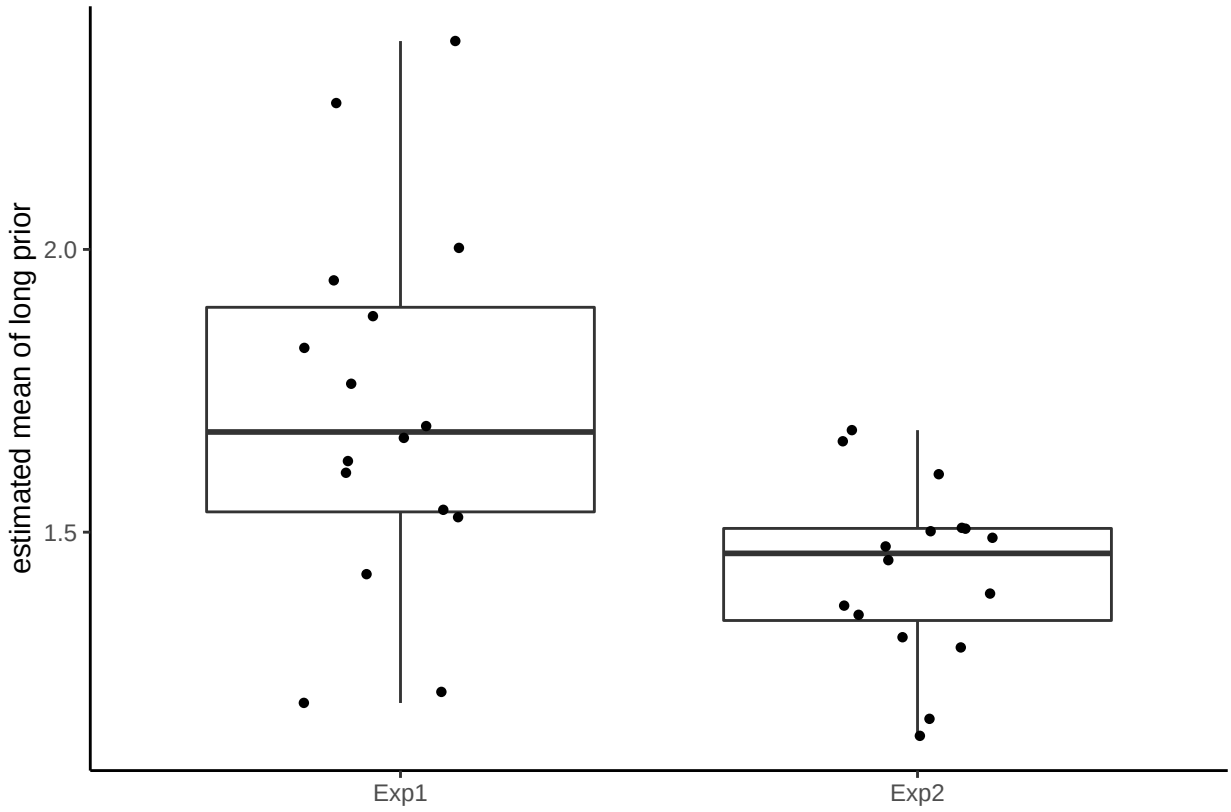



```
ggplot(AllDat_Bayparlist_BR, aes(Exp, sig_pr2_s)) +
  geom_boxplot(position = position_dodge()) +
  geom_jitter(shape=16, position=position_jitter(0.2))+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background = element_blank()) + # remove subtitle background
  labs(x = ' ', y = 'estimated variance of short prior')+
  theme(legend.position='bottom')
```

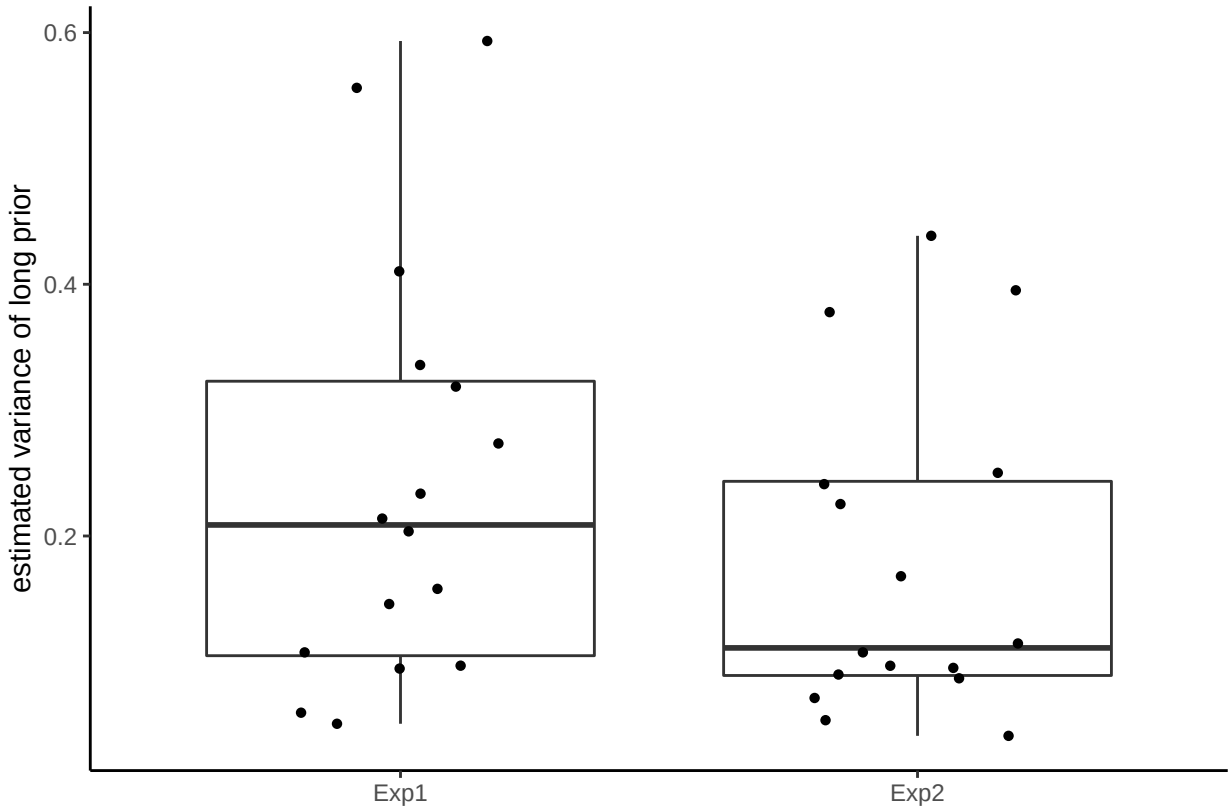


1.5 plot estimated long prior from baseline model

```
ggplot(AllDat_Bayparlist_BR, aes(Exp, mu_p_l)) +
  geom_boxplot(position = position_dodge()) +
  geom_jitter(shape=16, position=position_jitter(0.2))+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background = element_blank()) + # remove subtitle background
  labs(x = ' ', y = 'estimated mean of long prior')+
  theme(legend.position='bottom')
```

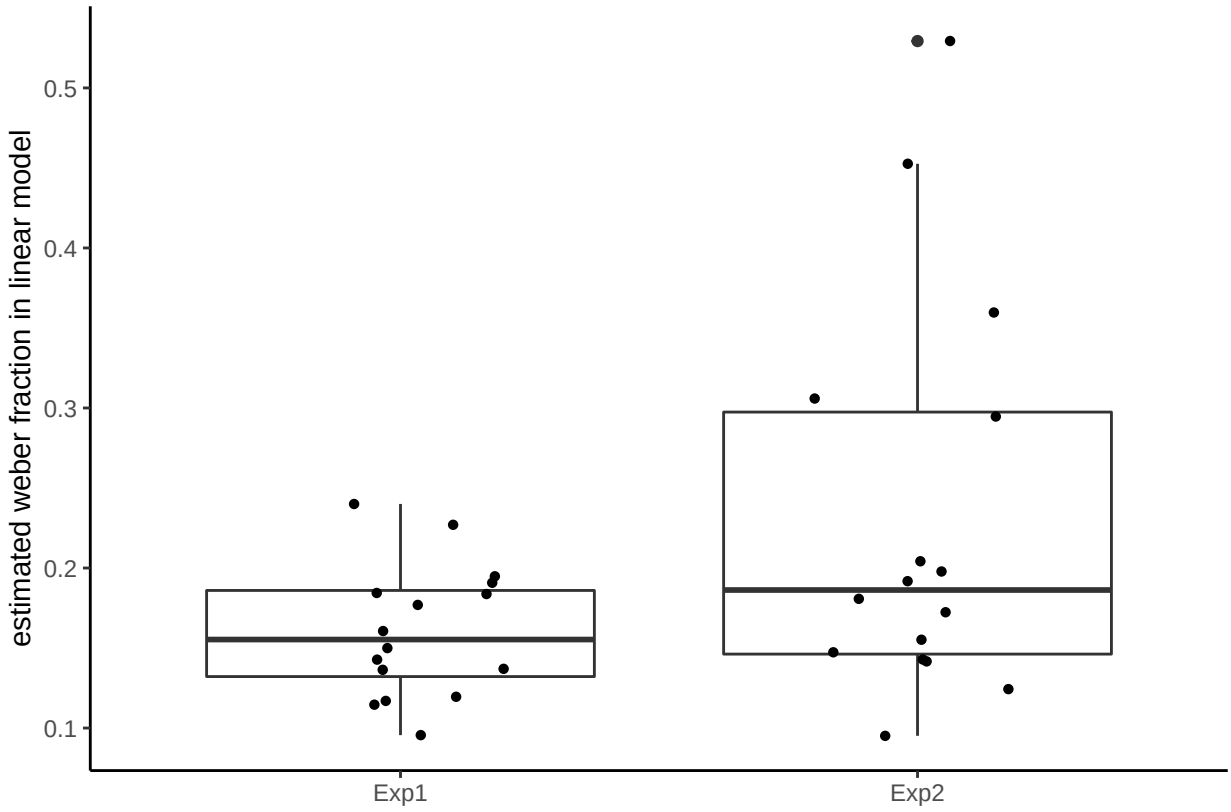



```
ggplot(AllDat_Bayparlist_BR, aes(Exp, sig_pr2_l)) +
  geom_boxplot(position = position_dodge()) +
  geom_jitter(shape=16, position=position_jitter(0.2))+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background = element_blank()) + # remove subtitle background
  labs(x = ' ', y = 'estimated variance of long prior')+
  theme(legend.position='bottom')
```



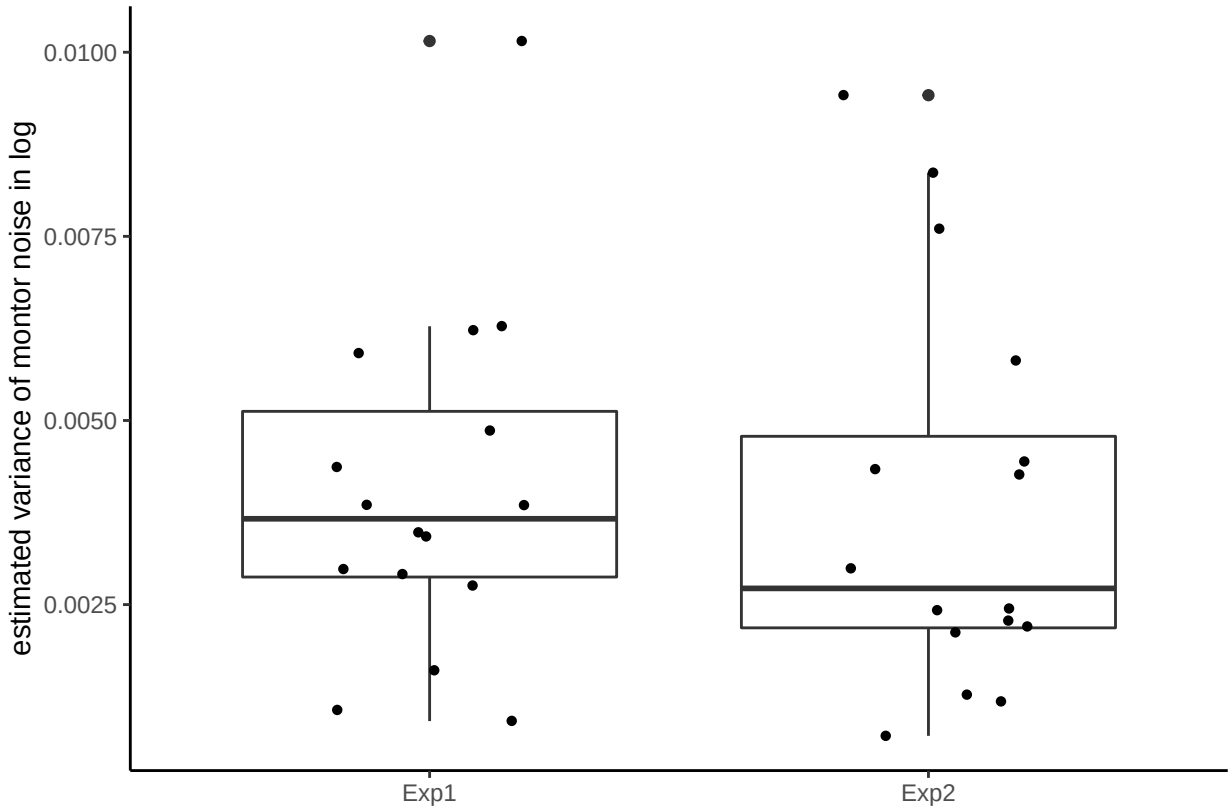
plot estimated sig_t from baseline model

```
ggplot(AllDat_Bayparlist_BR, aes(Exp, sig_t)) +
  geom_boxplot(position = position_dodge()) +
  geom_jitter(shape=16, position=position_jitter(0.2))+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background = element_blank()) + # remove subtitle background
  labs(x = ' ', y = 'estimated weber fraction in linear model')+
  theme(legend.position='bottom')
```



plot estimated variance of montor noise from baseline model

```
ggplot(AllDat_Bayparlist_BR, aes(Exp, sig2_mn)) +
  geom_boxplot(position = position_dodge()) +
  geom_jitter(shape=16, position=position_jitter(0.2))+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background = element_blank()) + # remove subtitle background
  labs(x = ' ', y = 'estimated variance of montor noise in log')+
  theme(legend.position='bottom')
```



1.6 load the estimated parameter of linear models on IR session

```
mm_Baypar <- dplyr::group_by(m_Baypar, Exp, model) %>%
  dplyr::summarize( mm_mu_p_s = mean(m_mu_p_s_IR),
                    mm_mu_p_l = mean(m_mu_p_l_IR),
                    mm_mu_p_g = mean(m_mu_p_g),
                    mm_sig2_p_s = mean(m_mu_p_s_IR),
                    mm_sig2_p_l = mean(m_sig_pr2_l_IR),
                    mm_sig2_p_g = mean(m_sig2_p_g))
```

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

```
mm_Baypar
```

```
## # A tibble: 8 x 8
## # Groups:   Exp [2]
##   Exp   model mm_mu_p_s mm_mu_p_l mm_mu_p_g mm_sig2_p_s mm_sig2_p_l mm_sig2_p_g
##   <chr> <chr>     <dbl>     <dbl>     <dbl>       <dbl>       <dbl>       <dbl>
## 1 Exp1  DIM       0          0         1.33       0           0          0
## 2 Exp1  IP        0.820      1.58      0          0.820       0.291      0.0559
## 3 Exp1  LGM       0          0         1.56       0           0          0
## 4 Exp1  PIM       0          0         1.56       0           0          0
## 5 Exp2  DIM       0          0         1.06       0           0          0
## 6 Exp2  IP        0.804      1.24      0          0.804       0.115      0.0925
## 7 Exp2  LGM       0          0         1.24       0           0          0
## 8 Exp2  PIM       0          0         1.24       0           0          0
```

```
fig_sigma2_long <- ggplot(mm_Baypar, aes(Exp, mm_sig2_p_l, ymin = mm_sig2_p_l -se_sig2_p_l , ymax = mm_s
  geom_line(stat = "identity",position = position_dodge(width = 0.2))+
  geom_point(stat = "identity",position = position_dodge(width = 0.2))+
  geom_errorbar(width=.2,  position = position_dodge(width = 0.2)) +
  theme_minimal()+ theme_new +
  labs(x = " ",
       y = "variance of local prior for long range",
       color = "model")+
  #scale_color_manual(values = mycolors5) +
  scale_shape_manual(values = myshapevalues)
```

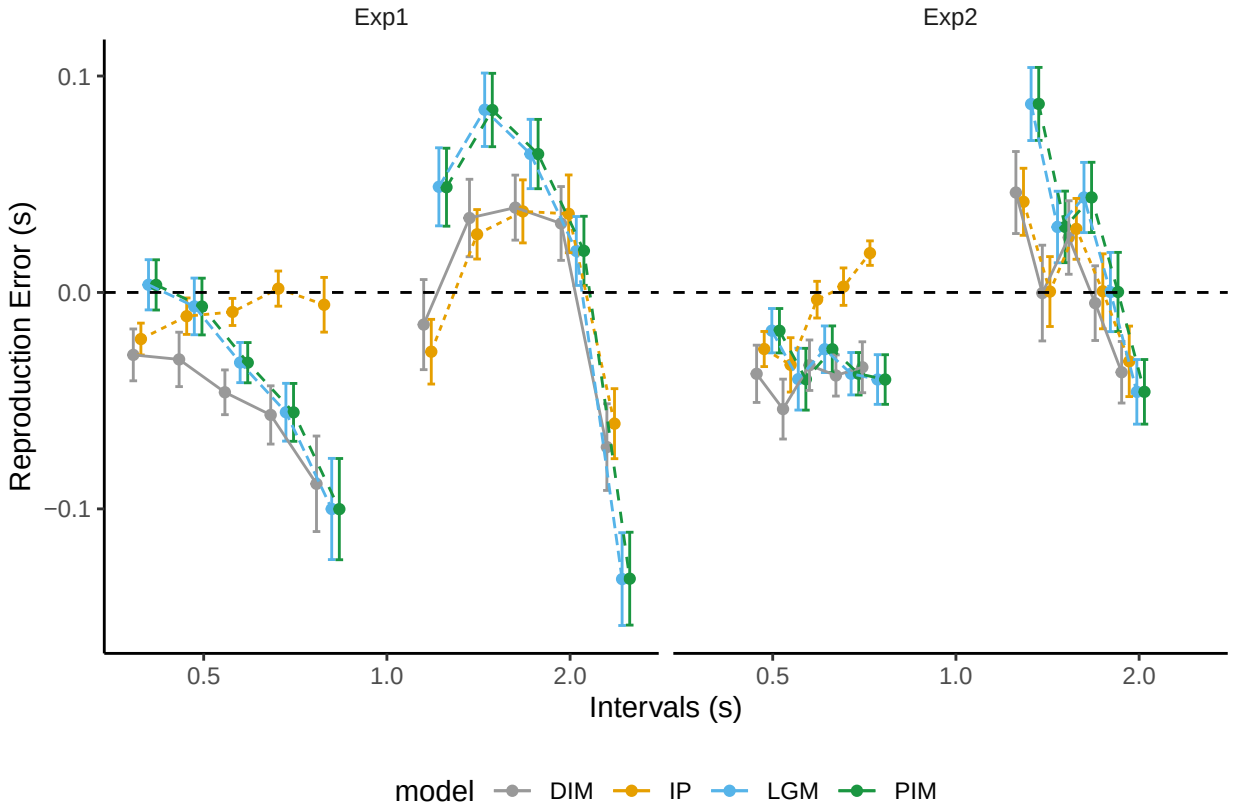
2 Analysis on predicted RP of IR session

```
# m_predY <- dplyr::group_by(All_predY, Exp, targetDur, model) %>%
#   dplyr::summarize( n = n(),
#                     m_RP = mean(RP), #mean of observed RP
#                     sd_RP = sd(RP), #observed sd of RP
#                     m_sig_r = mean(sig_r),
#                     m_mu_r = mean(mu_r),
#                     sd_mu_r = sd(mu_r))
#
# m_predY$merr_mu_r <- (m_predY$m_mu_r - m_predY$m_RP)/m_predY$m_RP
# m_predY$merr_sig_r <- (m_predY$m_sig_r - m_predY$sd_RP)/m_predY$sd_RP
# m_predY
```

2.1 Observed and predicted Reproduction Error

```
exps <- read.csv(paste0(getwd(), '/data/AllExpData_valid.csv'))
gp_var_prior= c("NSub", "targetDur", "Exp", "priortype", "range")
sumexp <- summarizeMeanData(exps, gp_var_prior)
sumexp$model = as.factor('measured')

m_predY$range = 'short'
m_predY[which(m_predY$targetDur > 1),"range"] = 'long'
m_predY$range = as.factor(m_predY$range)


fig_mrepError_model <- ggplot(m_predY,
                              aes(targetDur,
                                  mm_mu_r-m_m_RP,
                                  ymin = m_m_prederr - se_prederr,
                                  ymax = m_m_prederr + se_prederr,
                                  group = interaction(range, model), color = model)) +
  geom_point(position = position_dodge(width = 0.05))+
  geom_errorbar(width = .05, position = position_dodge(width = 0.05))+
  geom_line(size = .5, aes(color = model, linetype = model),  position = position_dodge(width = 0.05))+
  geom_hline(yintercept= 0, linetype = 2) +
  theme_minimal()+ theme_new +
  #scale_color_manual(values = mycolors5) +
  scale_linetype_manual(values =  c('solid','dotted','dashed' ))+
  facet_wrap(~Exp)+
```

```
  scale_linetype(guide = FALSE)+
  theme(strip.background = element_blank()) + # remove subtitle background
  labs(x = 'Intervals (s)', y = 'Reproduction Error (s)',  color='model')+ theme(legend.position='botto
  scale_x_continuous(trans='log10') +colorSet4


ggsave(file.path(modelPath,'figures/fig_mrepError_model.png'), fig_mrepError_model, width = 7, height =
fig_mrepError_model
```




```
predY_err_new <- m_predY_sub %>%dplyr::group_by(Exp, model) %>%
  dplyr::summarize(n= n(),
                   mm_m_RP = mean(m_m_RP),
                   mm_sd_RP = mean(m_sd_RP),
                   mm_pred_Var = mean(mm_sig_r- m_sd_RP),
                   mm_pred_Var_NBIAS = mean(abs(mm_sig_r- m_sd_RP)),
                   se_prederr = sd(m_m_prederr)/sqrt(n-1),
                   se_pred_Var = sd(mm_sig_r- m_sd_RP)/sqrt(n-1),
                   m_m_prederr = mean(m_m_prederr),
                   m_m_pred_NBIAS = mean(abs(m_m_prederr)))
```

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

```
gp_var_sub <- c('NSub', 'Exp', 'model')
m_predY_sub_new <- summarizePredY(predY, gp_var_sub)
```

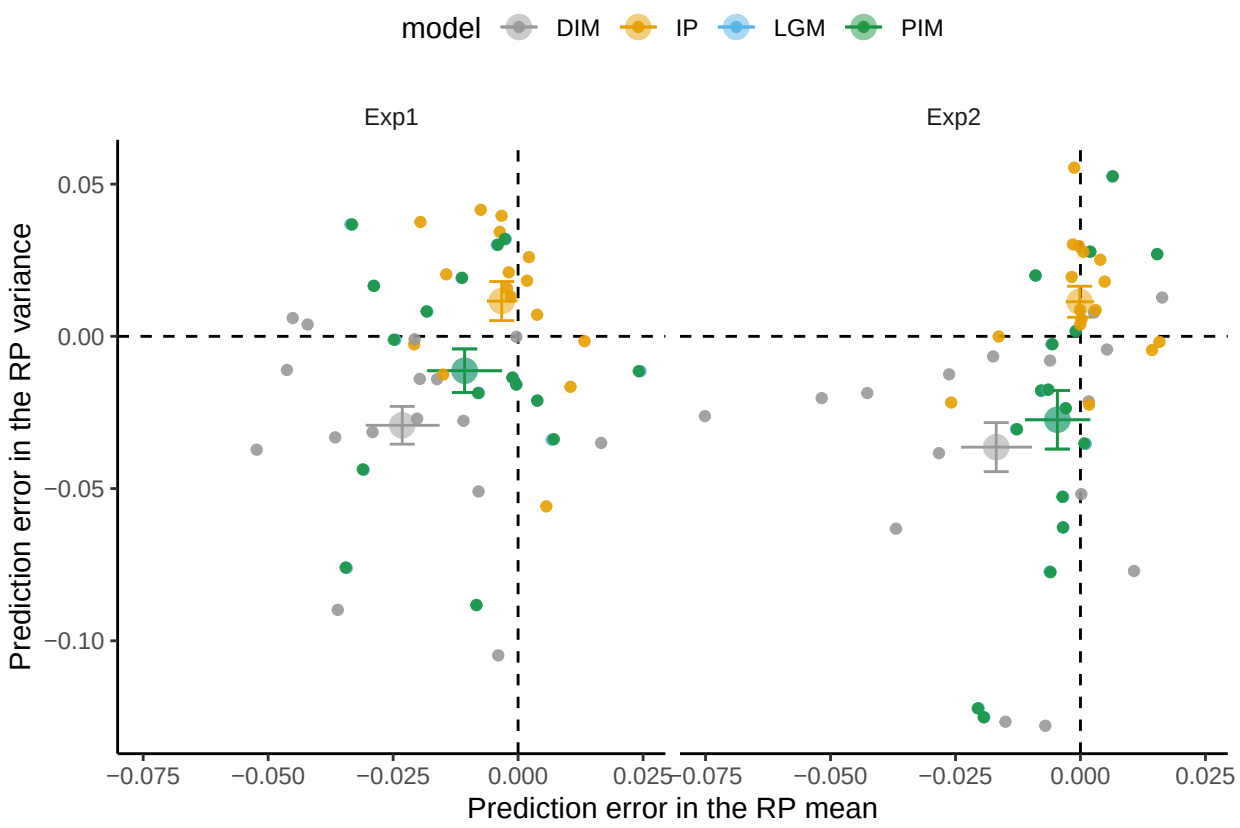
```
## `summarise()` has grouped output by 'NSub', 'Exp'. You can override using the `.groups` argument.
```

```
m_predY_sub_new$mm_pred_Var = m_predY_sub_new$mm_sig_r- m_predY_sub_new$m_sd_RP
plt_rErrorScatter1 = ggplot(data = predY_err_new, aes(m_m_prederr, mm_pred_Var, color = model)) +
```

```
  geom_hline(yintercept = 0, linetype='dashed')+ geom_vline(xintercept = 0, linetype='dashed')+
  geom_point(alpha = .5, size = 4)+
  geom_errorbar(aes(ymin = mm_pred_Var-se_pred_Var, ymax = mm_pred_Var+se_pred_Var), width = .005)+
  geom_errorbarh(aes(xmin=m_m_prederr-se_prederr, xmax = m_m_prederr+se_prederr), width = .1)+
  geom_point(m_predY_sub_new, mapping = aes(m_m_prederr, mm_pred_Var, color = model), alpha = .9) +
  facet_wrap(~Exp) +colorSet4+
  xlab('Prediction error in the RP mean')+
  ylab('Prediction error in the RP variance')+
  theme_new+ theme(legend.position = 'top')+guides(size="none")+guides(alpha="none")
```

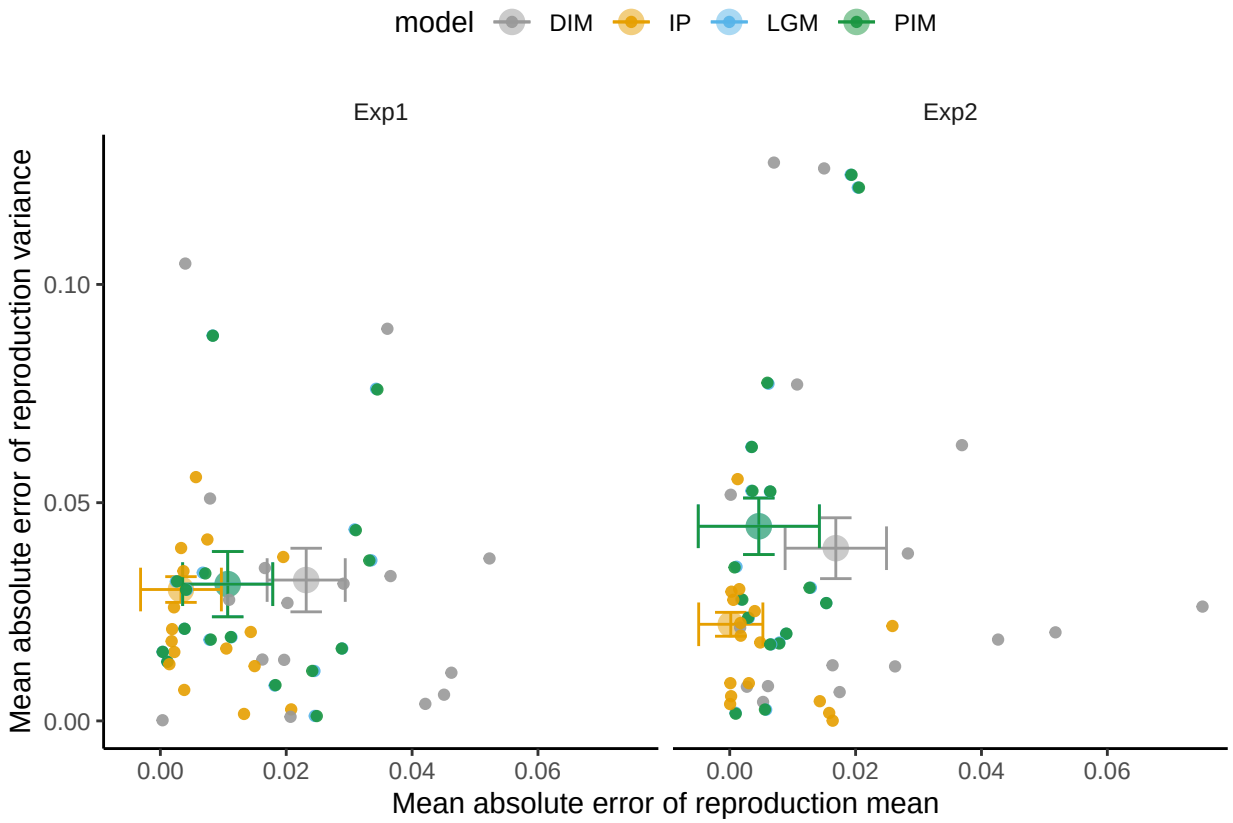
```
## Warning: Ignoring unknown parameters: width
```

```
plt_rErrorScatter1
```

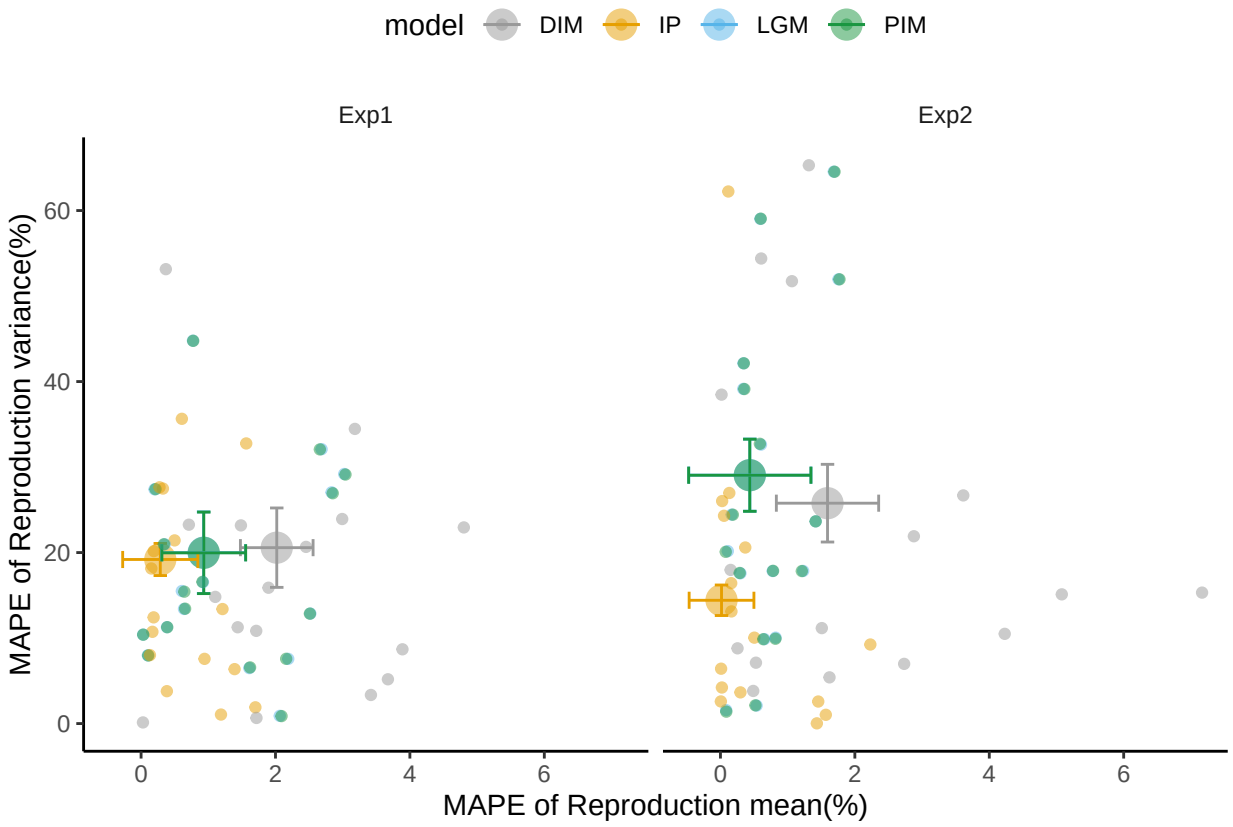



```
predY_err_new$acc_err <- 100*(1-predY_err_new$m_m_pred_NBIAS/predY_err_new$mm_m_RP)
predY_err_new$acc_var <- 100*(1-predY_err_new$mm_pred_Var_NBIAS/predY_err_new$mm_sd_RP)
```

```
plt_rErrorScatter2 = ggplot(data = predY_err_new, aes(m_m_pred_NBIAS, mm_pred_Var_NBIAS, color = model))
  #geom_hline(yintercept = 0, linetype='dashed')+ geom_vline(xintercept = 0, linetype='dashed')+
  geom_point(alpha = .5, size = 4)+
  geom_errorbar(aes(ymin = mm_pred_Var_NBIAS-se_prederr, ymax = mm_pred_Var_NBIAS+se_prederr), width =
  geom_errorbarh(aes(xmin= m_m_pred_NBIAS-se_pred_Var, xmax = m_m_pred_NBIAS+se_pred_Var), height = .01
  geom_point(m_predY_sub_new, mapping = aes(abs(m_m_prederr), abs(mm_pred_Var), color = model), alpha =
  facet_wrap(~Exp) +colorSet4+
  xlab('Mean absolute error of reproduction mean')+
  ylab('Mean absolute error of reproduction variance')+
  theme_new+ theme(legend.position = 'top')+guides(size="none")+guides(alpha="none")
plt_rErrorScatter2
```




```
plt_rErrorScatter3 = ggplot(data = predY_err_new, aes(100*m_m_pred_NBIAS/mm_m_RP, 100*mm_pred_Var_NBIAS/
  #geom_hline(yintercept = 0, linetype='dashed')+ geom_vline(xintercept = 0, linetype='dashed')+
  geom_point(alpha = .5, size = 5)+
  geom_errorbar(aes(ymin = 100*(mm_pred_Var_NBIAS-se_prederr)/mm_sd_RP, ymax = 100*(mm_pred_Var_NBIAS+se
  geom_errorbarh(aes(xmin= 100*(m_m_pred_NBIAS-se_pred_Var)/mm_m_RP, xmax = 100*(m_m_pred_NBIAS+se_pred
  geom_point(m_predY_sub_new, mapping = aes(100*abs(m_m_prederr)/m_m_RP, 100*abs(mm_pred_Var)/m_sd_RP,
  facet_wrap(~Exp) +colorSet4+
  xlab('MAPE of Reproduction mean(%)')+
  ylab('MAPE of Reproduction variance(%)')+
  theme_new+ theme(legend.position = 'top')+guides(size="none")+guides(alpha="none")
plt_rErrorScatter3
```




```
fig_ModelRlt <- ggarrange(fig_mrepError_model, plt_rErrorScatter3, common.legend = TRUE, nrow = 2, ncol
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

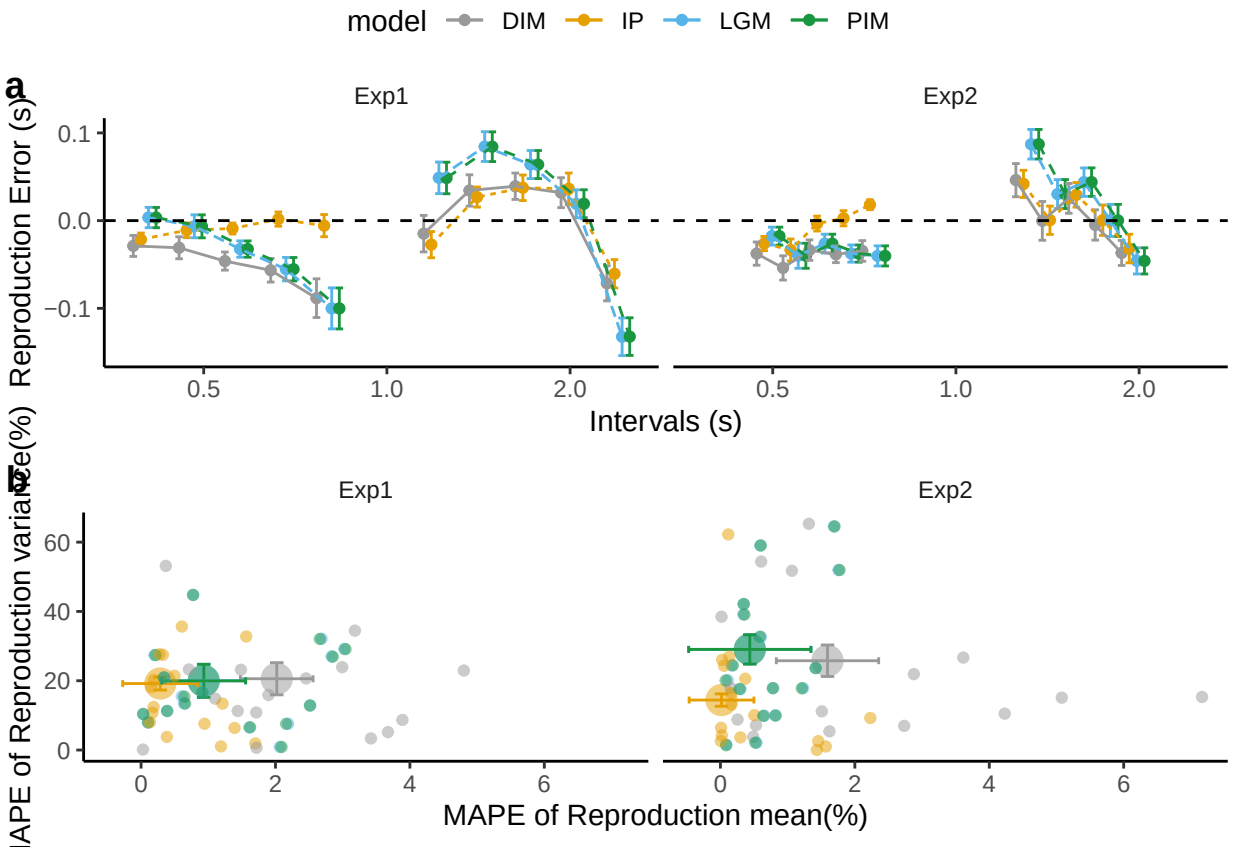
```
## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

```
ggsave(paste0(getwd(), "/", modelPath, "/figures/fig_ModelRlt.png"), fig_ModelRlt, width = 7, height = 
fig_ModelRlt
```



2.2 Compute correlation

```
cor_err_table <- predY %>% dplyr::group_by(Exp, model) %>%
  dplyr::summarise(r2 = cor(m_RP, m_mu_r, method = c("pearson", "kendall", "spearman")))%>% dplyr::grou
  dplyr::summarise(m_r2 = mean(r2))
```

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

```
cor_err_table
```

```
## # A tibble: 8 x 3
## # Groups:   Exp [2]
##   Exp   model  m_r2
##   <chr> <fct> <dbl>
## 1 Exp1  DIM   0.991
## 2 Exp1  IP    0.995
## 3 Exp1  LGM   0.987
## 4 Exp1  PIM   0.987
## 5 Exp2  DIM   0.986
## 6 Exp2  IP    0.990
## 7 Exp2  LGM   0.985
## 8 Exp2  PIM   0.985
```

```
cor_err_table <- predY%>% dplyr::group_by(Exp, model) %>%
  dplyr::summarise(r2 = cor(m_RP, m_mu_r, method = c("pearson", "kendall", "spearman")))%>% dplyr::grou
  dplyr::summarise(m_r2 = mean(r2))
```

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

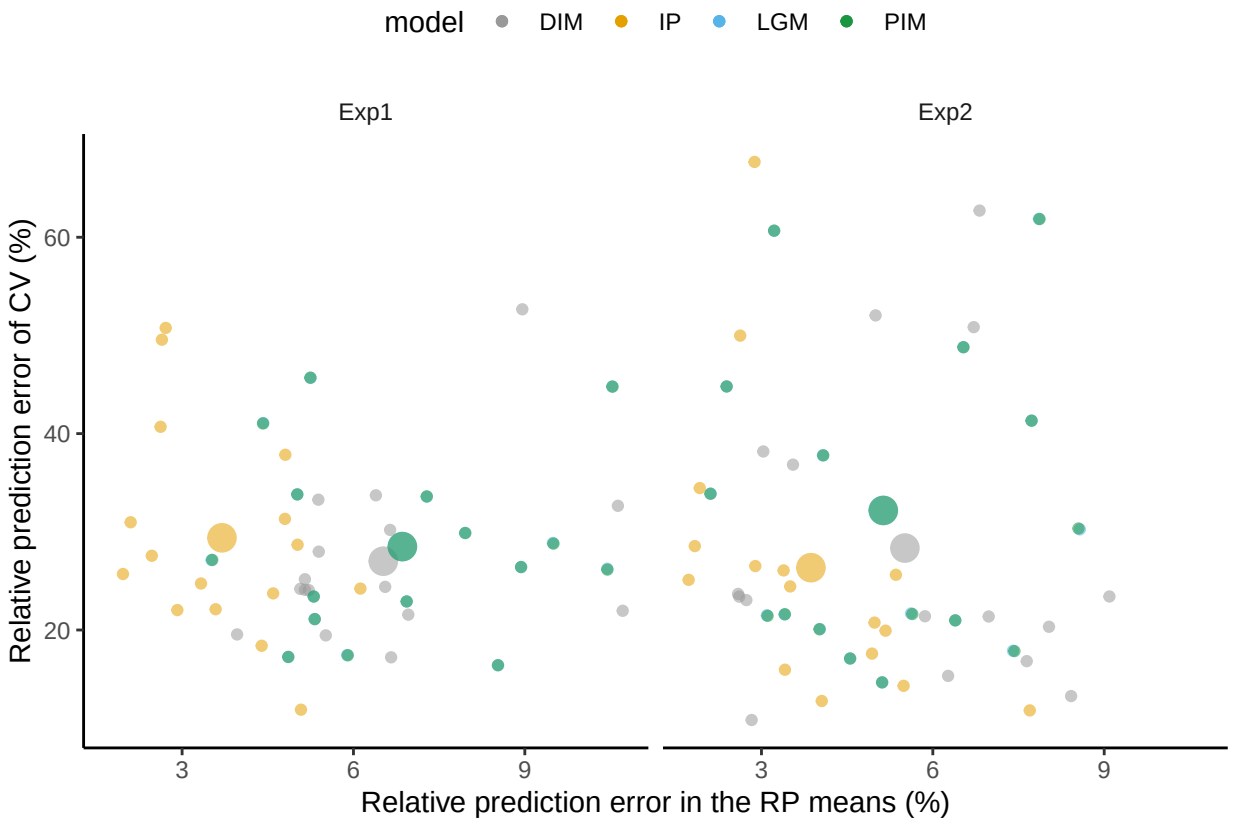
```
cor_err_table
```

```
## # A tibble: 8 x 3
## # Groups:   Exp [2]
##   Exp   model  m_r2
##   <chr> <fct> <dbl>
## 1 Exp1  DIM   0.991
## 2 Exp1  IP    0.995
## 3 Exp1  LGM   0.987
## 4 Exp1  PIM   0.987
## 5 Exp2  DIM   0.986
## 6 Exp2  IP    0.990
## 7 Exp2  LGM   0.985
## 8 Exp2  PIM   0.985
```

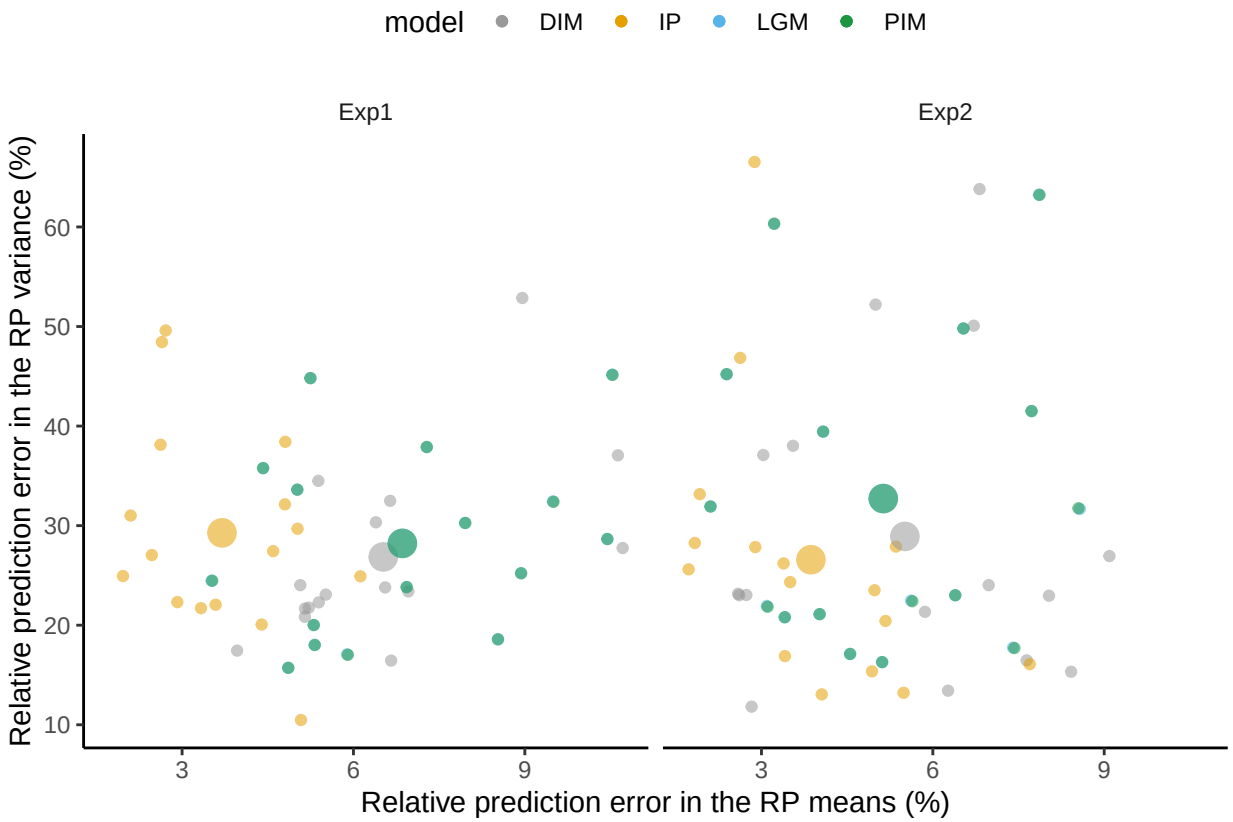
3 Model comparision in prediction

3.0.1 plot prediction error of RP mean vs. CV

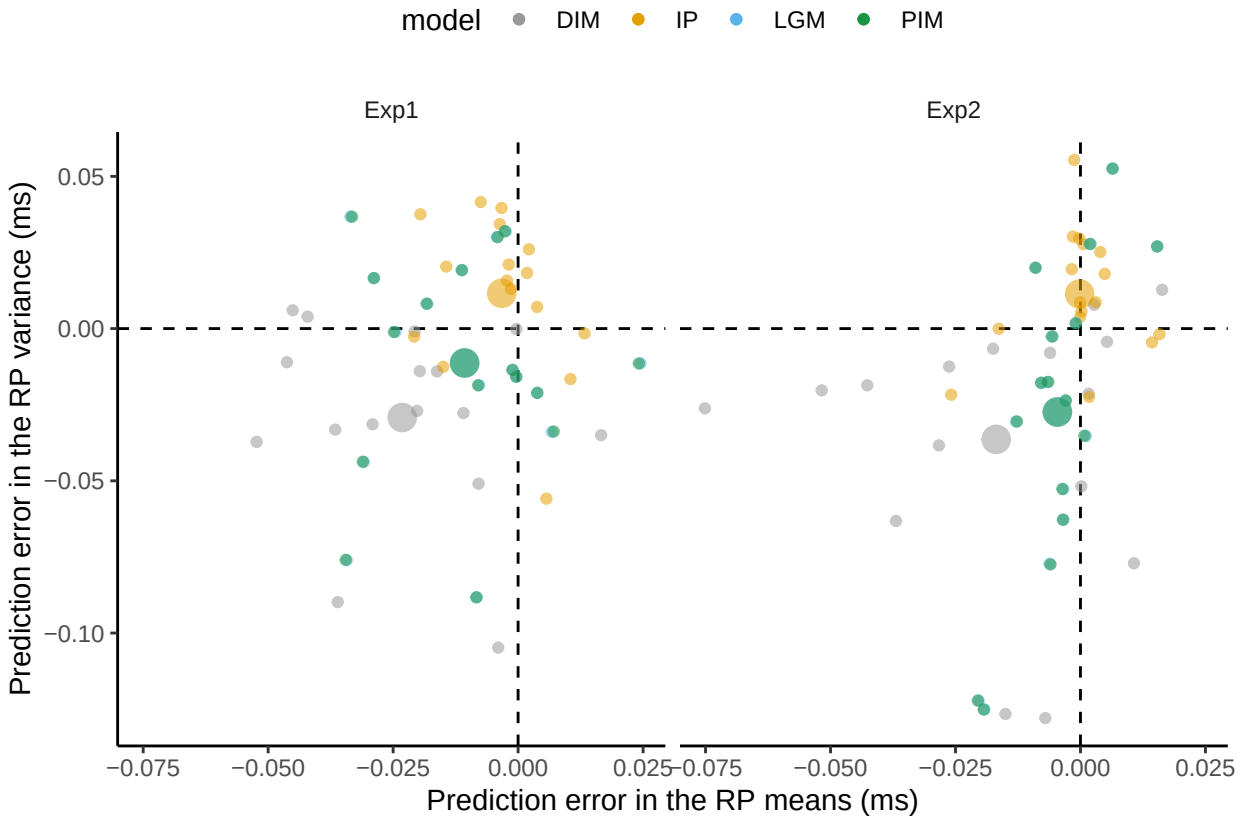
```
ggplot(m_predErr_sub, aes(mpredRP_rerr*100, mpredcv_rerr*100, color = model, alpha = .9)) +
  #geom_hline(yintercept = 0, linetype='dashed')+ geom_vline(xintercept = 0, linetype='dashed')+
  geom_point() +
  geom_point(data = m_predErr, aes(mmpredRP_rerr*100, mmpredcv_rerr*100, color = model, alpha = .9, siz
  xlab('Relative prediction error in the RP means (%)')+ ylab('Relative prediction error of CV (%)')+co
  facet_wrap(~Exp)+
  theme_new+ theme(legend.position = 'top')+guides(size="none")+guides(alpha="none")
```




```
plt_rErrorScatter = ggplot(m_predErr_sub, aes(mpredRP_rerr*100, mpredVar_rerr*100, color = model, alpha
  #geom_hline(yintercept = 0, linetype='dashed')+ geom_vline(xintercept = 0, linetype='dashed')+
  geom_point() +
  geom_point(data = m_predErr, aes(mmpredRP_rerr*100, mmpredVar_rerr*100, color = model, alpha = .9, si
  xlab('Relative prediction error in the RP means (%)')+ ylab('Relative prediction error in the RP varia
  facet_wrap(~Exp)+
  theme_new+ theme(legend.position = 'top')+guides(size="none")+guides(alpha="none")

plt_rErrorScatter
```




```
plt_ErrorScatter = ggplot(m_predErr_sub, aes(mpredRP_err, mpredVar_err, color = model, alpha = .9)) +
  geom_hline(yintercept = 0, linetype='dashed')+ geom_vline(xintercept = 0, linetype='dashed')+
  geom_point() +
  geom_point(data = m_predErr, aes(mmpredRP_err, mmpredVar_err, color = model, alpha = .9, size = 1 ))+
  xlab('Prediction error in the RP means (ms)')+ ylab('Prediction error in the RP variance (ms)')+color$
  facet_wrap(~Exp)+
  theme_new+ theme(legend.position = 'top')+guides(size="none")+guides(alpha="none")

plt_ErrorScatter
```


model
DIM
IP
LGM
PIM
Exp1
Exp2
Prediction error in the RP variance (ms)
0.05
0.00
−0.05
−0.10
−0.075
−0.050
−0.025
0.000
0.025
Prediction error in the RP means (ms)