# ModelReportAll

Xiuna Zhu

12/6/2020

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

## Load the packages

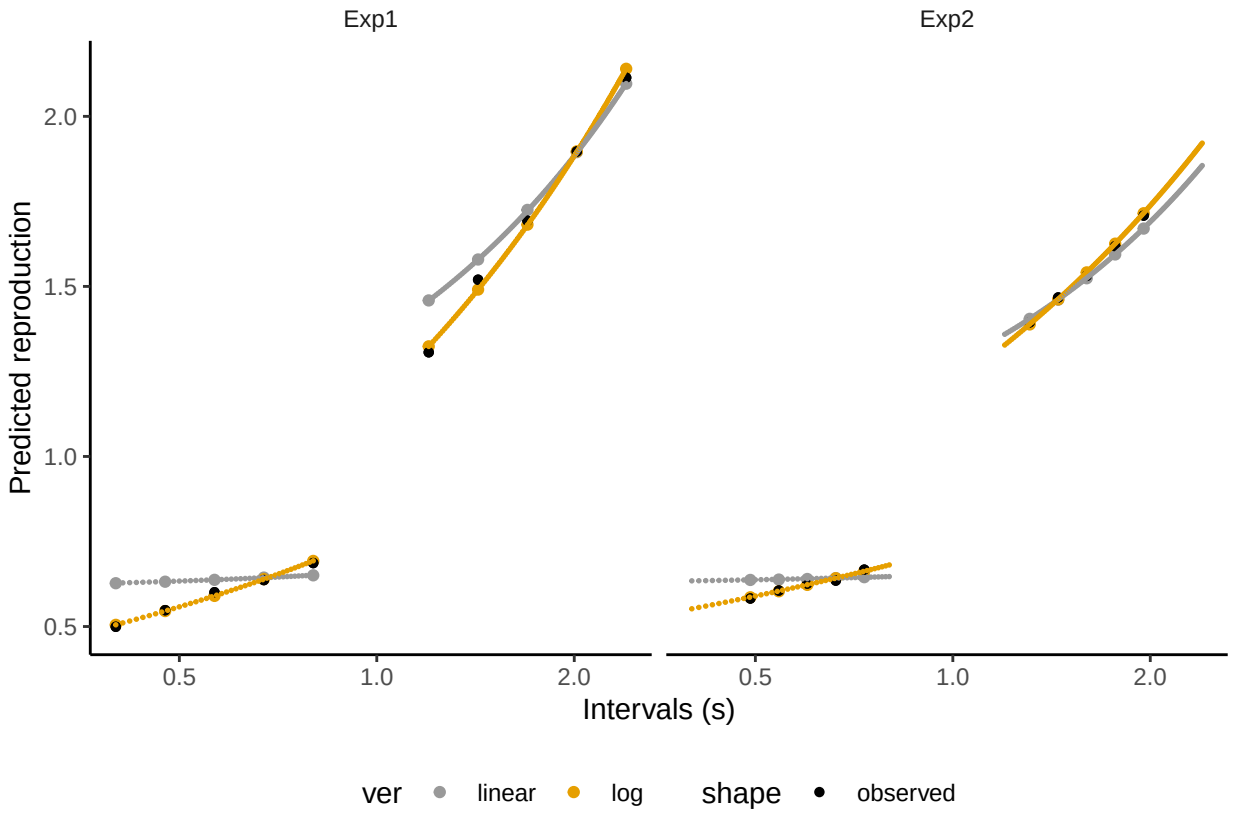
# 1 Baseline model results (for BR condition)

## 1.1 load baseline model results

## 1.2 BR session data prediction

### 1.2.1 plot predicted reproduction based on baseline model

```
fig_predRP_BR = ggplot(m_parameter_BR, aes(targetDur, m_mu_r)) +
  geom_point(aes(color= ver)) +
  geom_point(aes(targetDur, mRP, shape = 'observed'))+ #observed RP
  geom_point(data = m_NewY_BR, aes(targetDur, m_mu_r, color = ver), size = 0.3)+
  facet_wrap(~Exp)+
  theme_new+ colorSet3 +
  theme(strip.background = element_blank()) +
  labs(x = 'Intervals (s)', y = 'Predicted reproduction')+
  #scale_color_manual(labels = c("predicted", "observed"), values = c("blue", "red")) +
  theme(legend.position='bottom')+
  scale_x_continuous(trans='log10') #Transform x axis to log10 scale

fig_predRP_BR
```

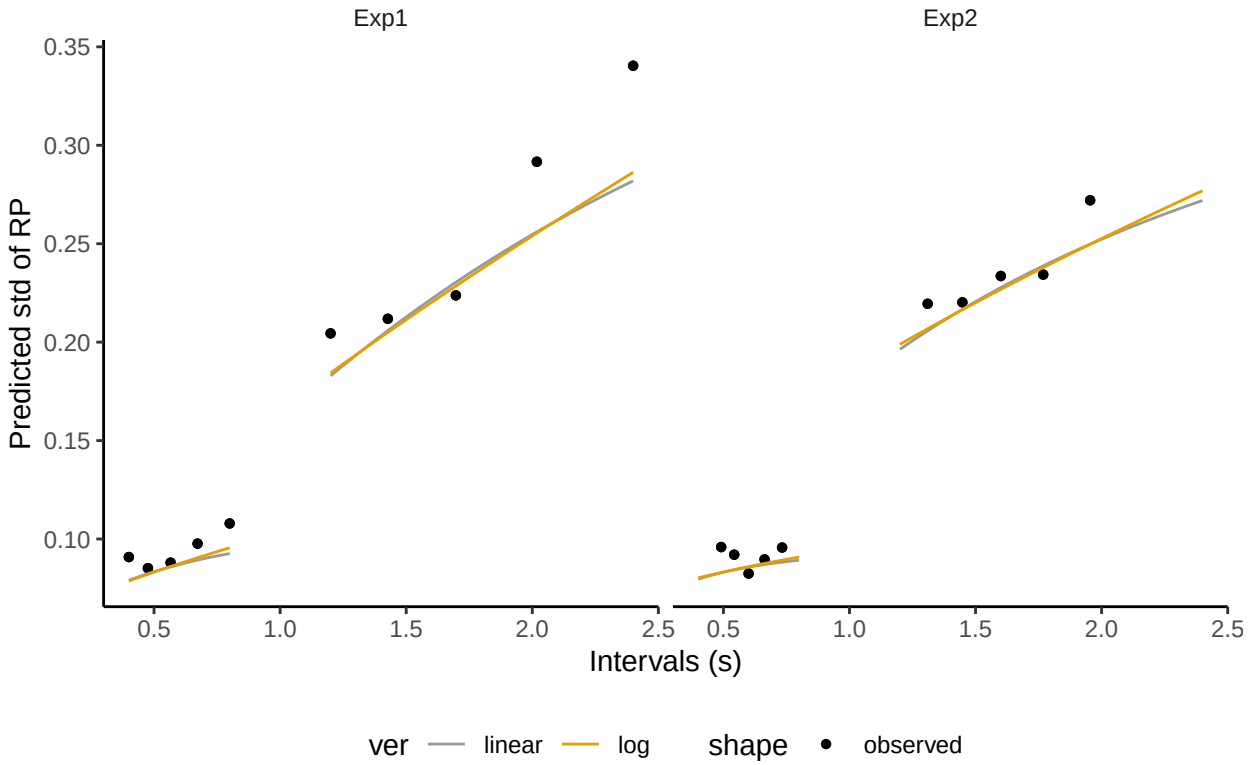



### 1.2.2 predicted SD of RP

```
fig_pred_SD_BR = ggplot(m_parameter_BR) +
  geom_point(aes(targetDur, sd_RP, shape = 'observed'))+ #observed RP
  geom_line(data = m_NewY_BR, aes(targetDur, m_sig_r, group = interaction(group, ver),color = ver))+
  facet_wrap(~Exp)+
  theme_new+ colorSet3 +
  theme(strip.background = element_blank()) +
  labs(x = 'Intervals (s)', y = 'Predicted std of RP')+
  theme(legend.position='bottom')+ ggtitle("")
  scale_x_continuous(trans='log10') #Transform x axis to log10 scale
```

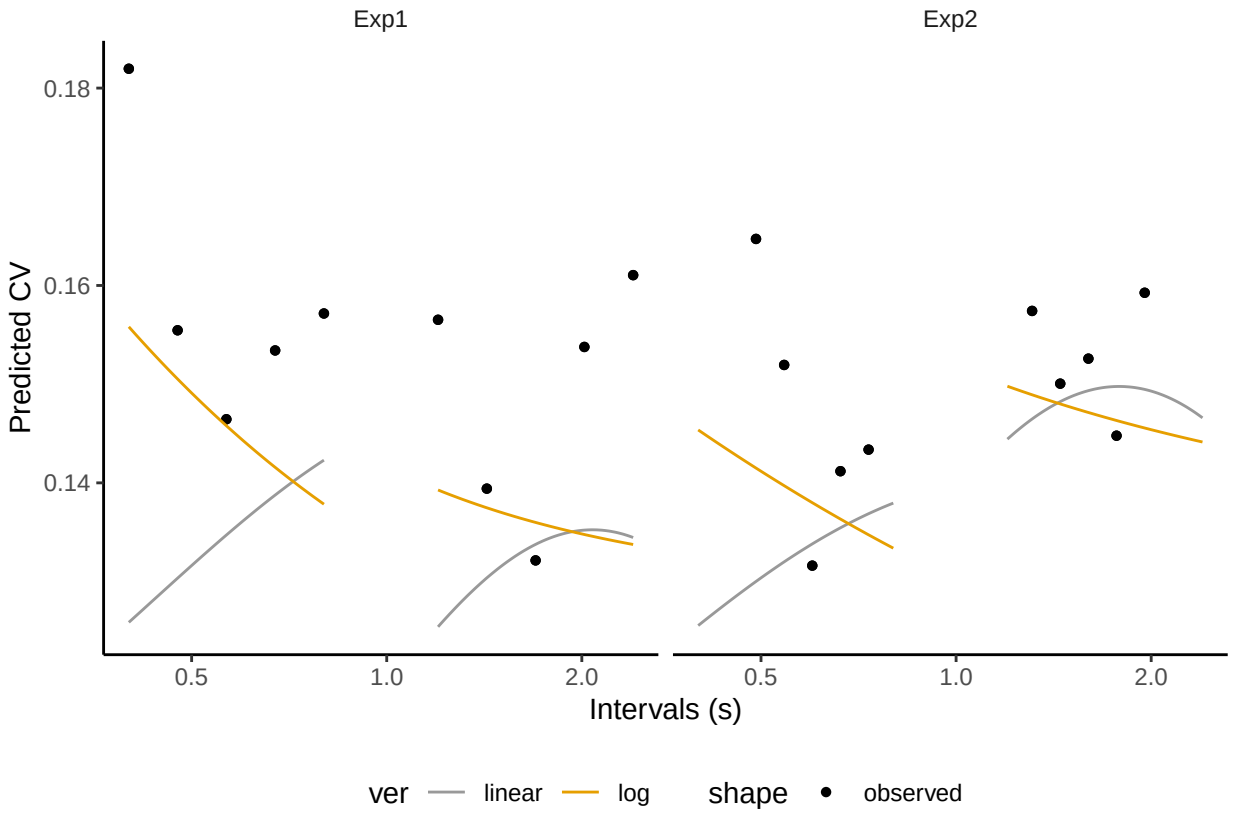
```
## <ScaleContinuousPosition>
##  Range:
##  Limits:    0 --    1
```

```
fig_pred_SD_BR
```



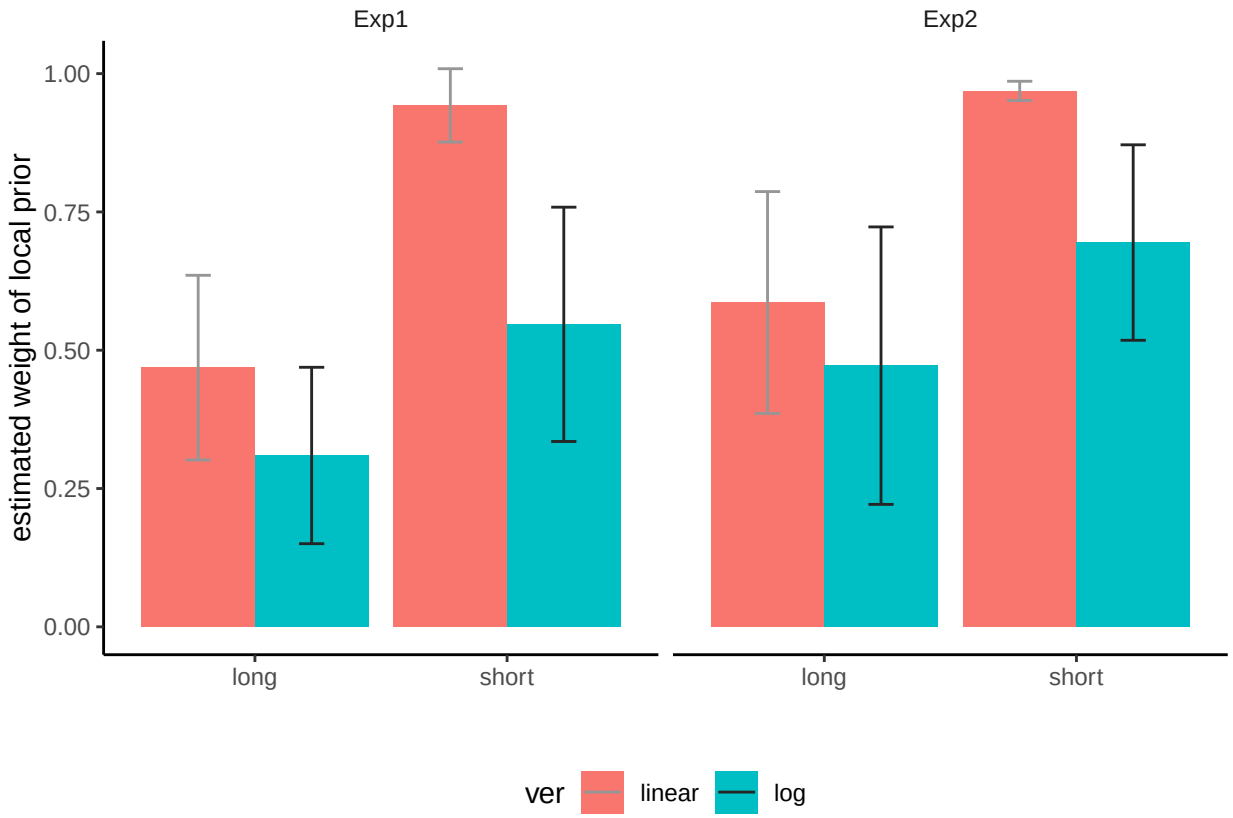

### 1.2.3 predicted CV

```
fig_predCV_BR = ggplot(m_parameter_BR) +
  geom_point(aes(targetDur, sd_RP/mRP, shape = 'observed'))+ #observed RP
  geom_line(data = m_NewY_BR, aes(targetDur, m_sig_r/m_mu_r, group = interaction(group, ver),color = ve
  facet_wrap(~Exp)+
  theme_new+ colorSet3 +
  theme(strip.background = element_blank()) +
  labs(x = 'Intervals (s)', y = 'Predicted CV')+
  theme(legend.position='bottom')+
  scale_x_continuous(trans='log10') #Transform x axis to log10 scale

fig_predCV_BR
```

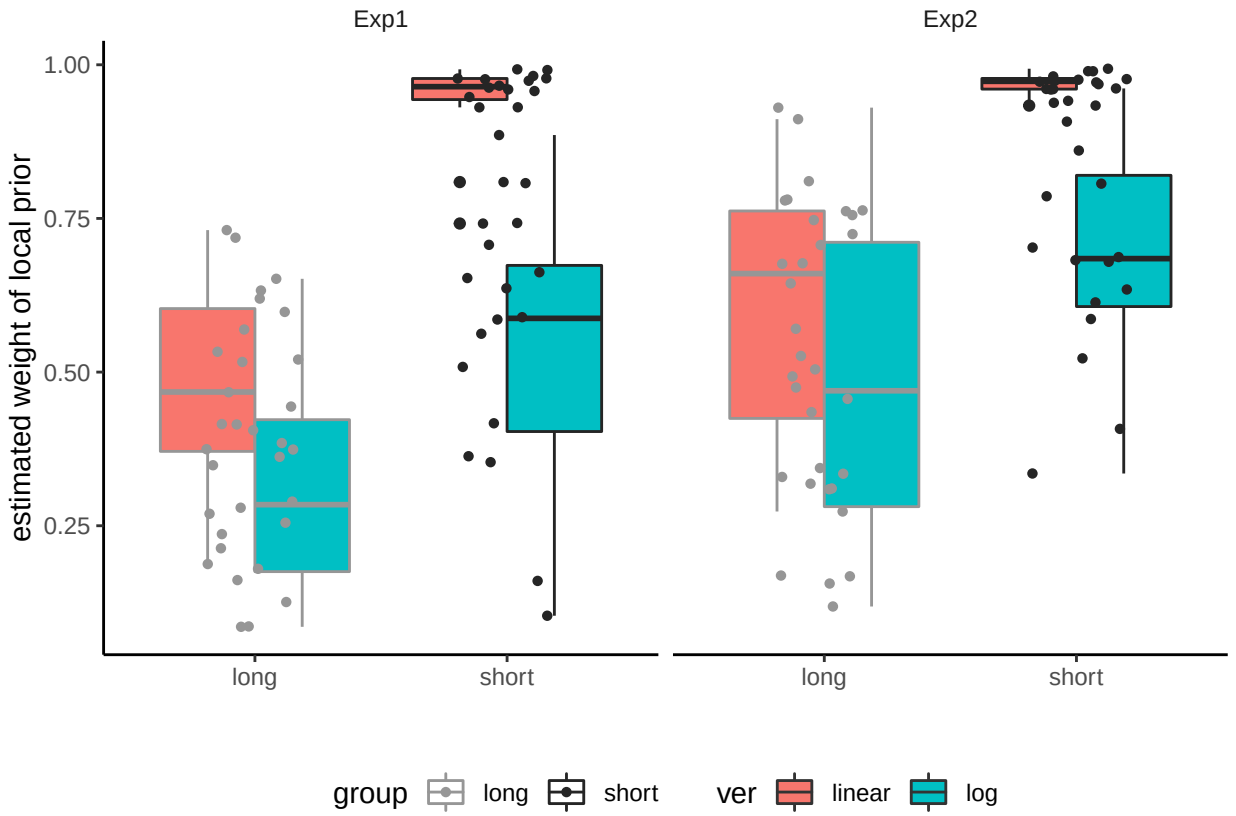



### 1.3 plot weight of local prior based on baseline model

```
ggplot(mm_parameter_BR, aes(x = group, y =m_wp)) +
  geom_bar(aes(fill = ver), position = 'dodge', stat = "identity")+
  geom_errorbar(aes(ymin = m_wp-sd_wp, ymax = m_wp + sd_wp, color = ver, width = 0.2),  position = posi
  facet_wrap(~Exp)+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background = element_blank()) +
  labs(x = ' ', y = 'estimated weight of local prior')+
  theme(legend.position='bottom')
```



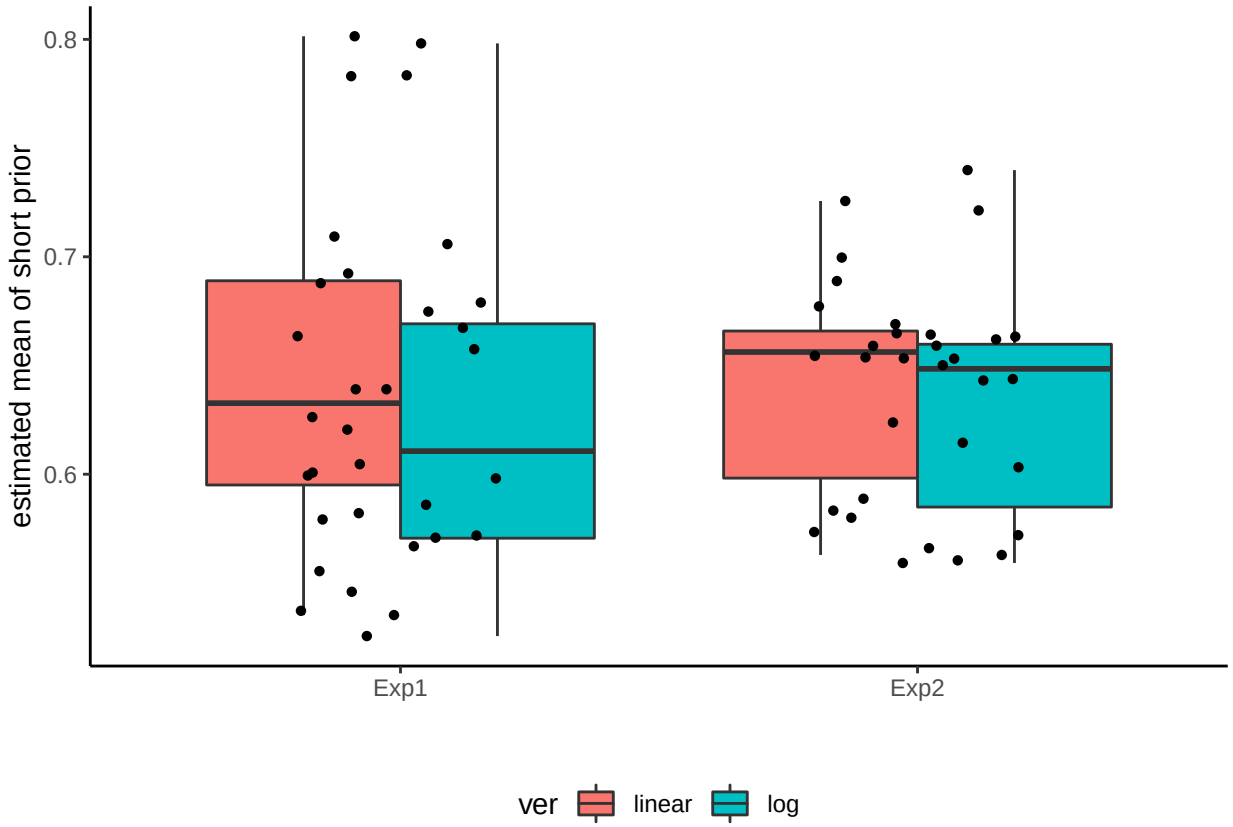

```
fig_wp_BR_subj = ggplot(m_parameter_BR_sub, aes(group, m_wp, color = group)) +
  geom_boxplot(position = position_dodge(), aes(fill = ver)) +
  geom_jitter(shape=16, position=position_jitter(0.2))+
  facet_wrap(~Exp)+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background =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

fig_wp_BR_subj
```



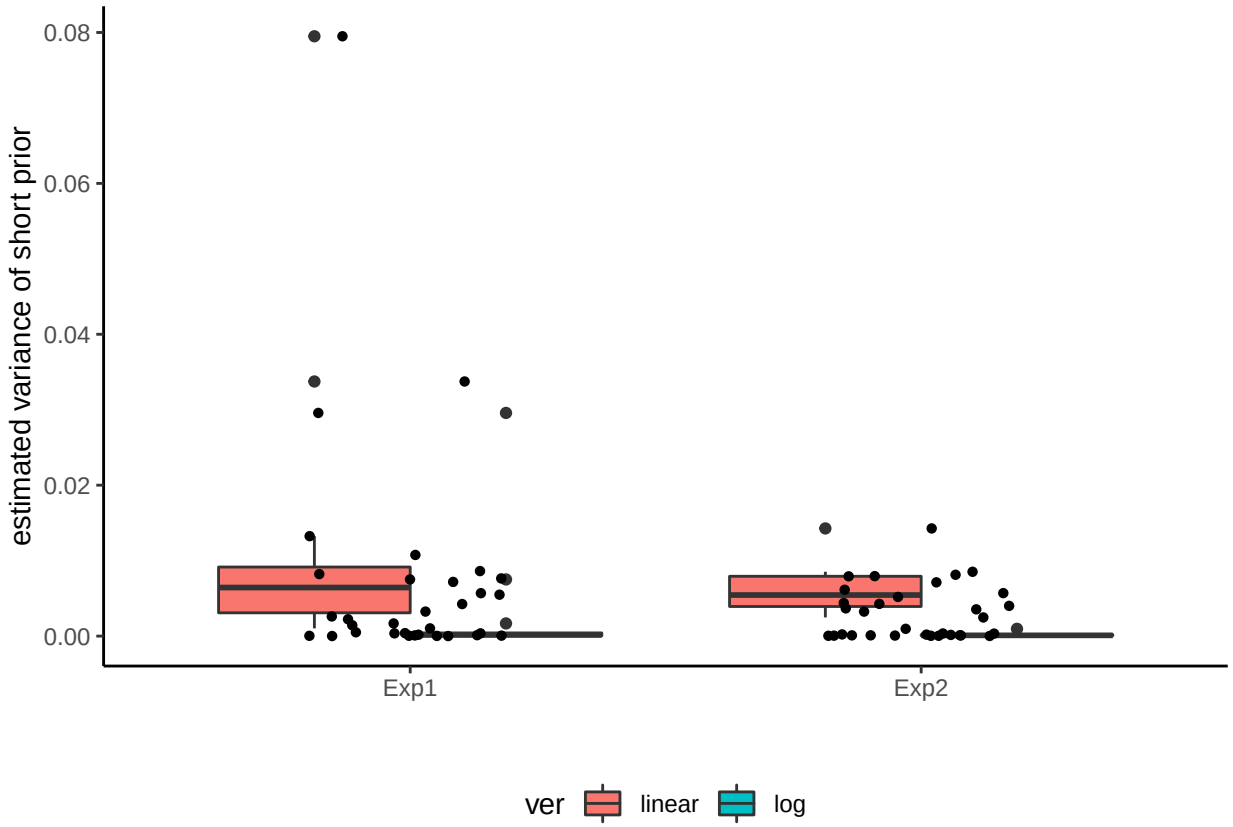

## load the estimated parameters from linear baseline model on BR session

### 1.4 short prior from baseline model

```
#plot estimated short prior from baseline model
ggplot(baseline_BR_Bayparlist, aes(Exp, mu_p_s)) +
  geom_boxplot(position = position_dodge(), aes(fill = ver)) +
  geom_jitter(shape=16, position=position_jitter(0.2))+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background = element_blank()) +
  labs(x = ' ', y = 'estimated mean of short prior')+
  theme(legend.position='bottom')
```

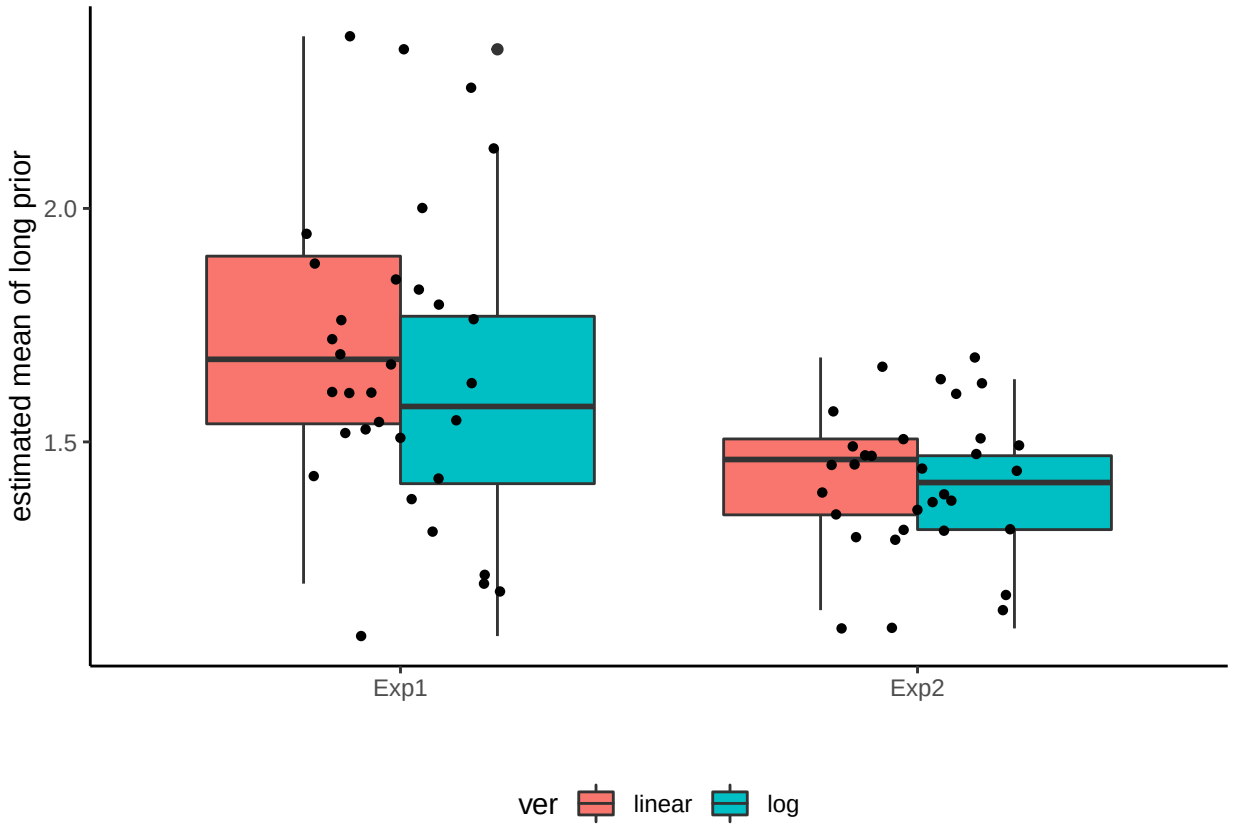



```
ggplot(baseline_BR_Bayparlist, aes(Exp, sig_pr2_s)) +
  geom_boxplot(position = position_dodge(), aes(fill = ver)) +
  geom_jitter(shape=16, position=position_jitter(0.2))+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background = element_blank()) +
  labs(x = ' ', y = 'estimated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

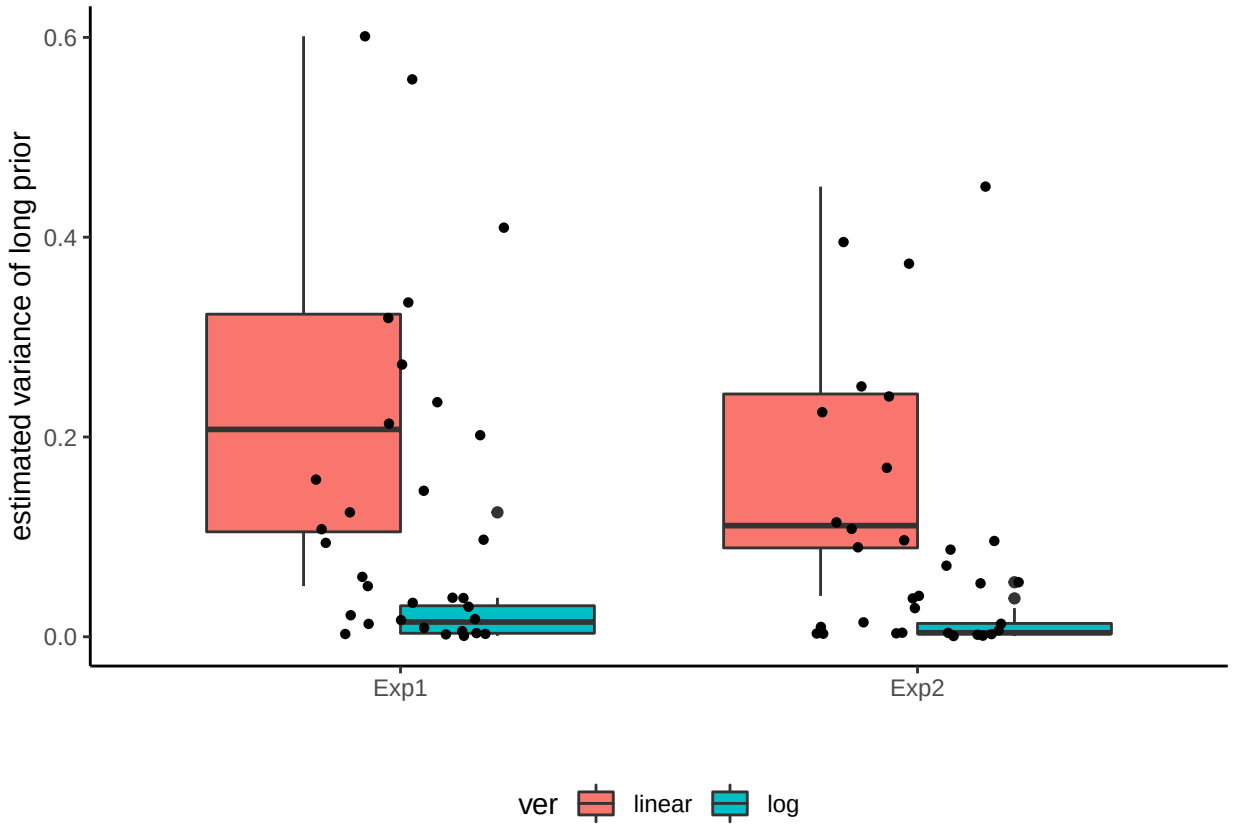



### 1.5 long prior from baseline model

```
## plot estimated long prior from baseline model
ggplot(baseline_BR_Bayparlist, aes(Exp, mu_p_l)) +
  geom_boxplot(position = position_dodge(), aes(fill = ver)) +
  geom_jitter(shape=16, position=position_jitter(0.2))+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background = element_blank()) +
  labs(x = ' ', y = 'estimated mean of long prior')+
  theme(legend.position='bottom')
```



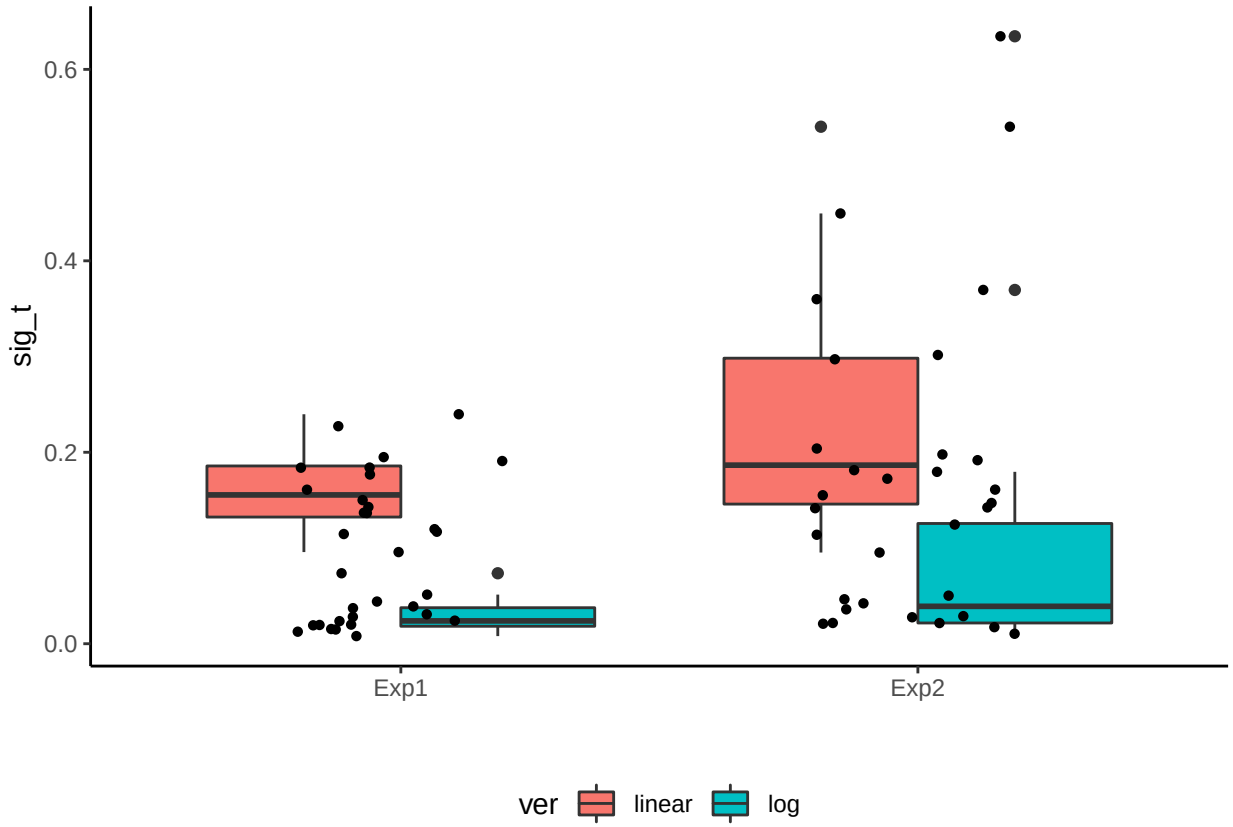

```
ggplot(baseline_BR_Bayparlist, aes(Exp, sig_pr2_l)) +
  geom_boxplot(position = position_dodge(), aes(fill = ver)) +
  geom_jitter(shape=16, position=position_jitter(0.2))+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background = element_blank()) +
  labs(x = ' ', y = 'estimated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




## sig_t from baseline model

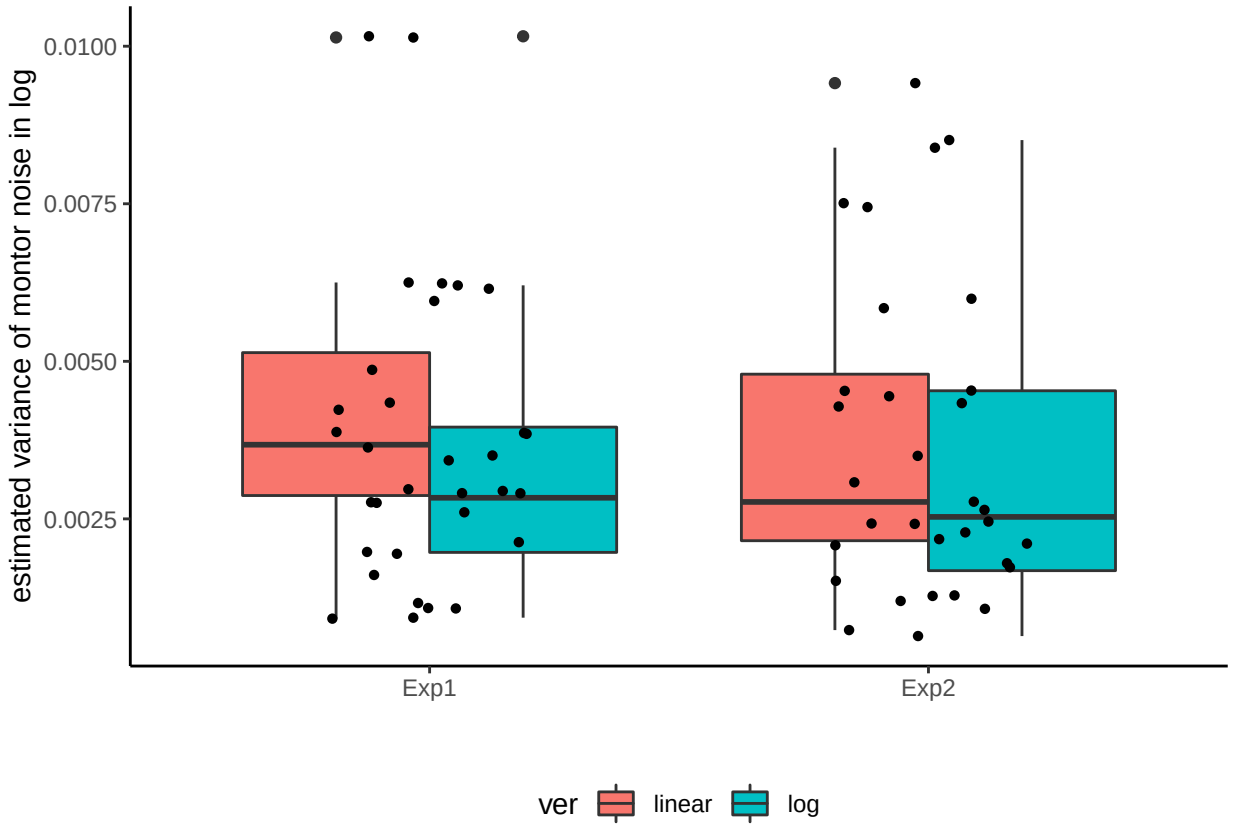
Note that $sigma_t$ in linear model indicates weber fraction.

```
#plot estimated sig_t from baseline model
ggplot(baseline_BR_Bayparlist, aes(Exp, sig_t)) +
  geom_boxplot(position = position_dodge(), aes(fill = ver)) +
  geom_jitter(shape=16, position=position_jitter(0.2))+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background = element_blank()) +
  labs(x = ' ', y = 'sig_t')+
  theme(legend.position='bottom')
```




## variance of montor noise based on baseline model

```
##plot estimated ariance of montor noise from baseline model
ggplot(baseline_BR_Bayparlist, aes(Exp, sig2_mn)) +
  geom_boxplot(position = position_dodge(), aes(fill = ver)) +
  geom_jitter(shape=16, position=position_jitter(0.2))+
  theme_new+ scale_color_manual(values = mycolors) +
  theme(strip.background = element_blank()) +
  labs(x = ' ', y = 'estimated variance of montor noise in log')+
  theme(legend.position='bottom')
```



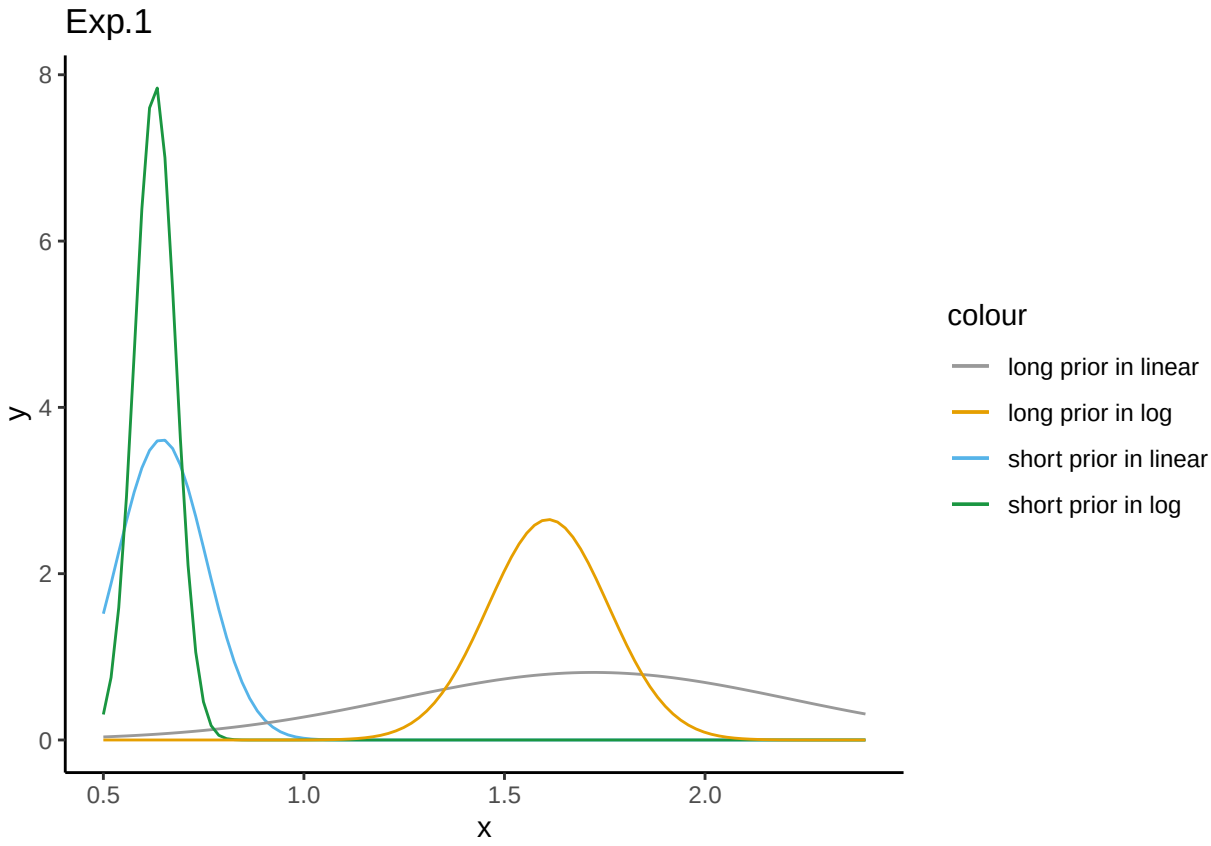

```
mpar_baseline_BR <- dplyr::group_by(baseline_BR_Bayparlist, Exp,ver) %>%
  dplyr::summarize( m_mu_p_s = mean(mu_p_s),
                    m_mu_p_l = mean(mu_p_l),
                    m_sig_pr2_s = mean(sig_pr2_s),
                    m_sig_pr2_l = mean(sig_pr2_l),
                    m_sig2_mn = mean(sig2_mn),
                    m_sig_t = mean(sig_t))
```

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

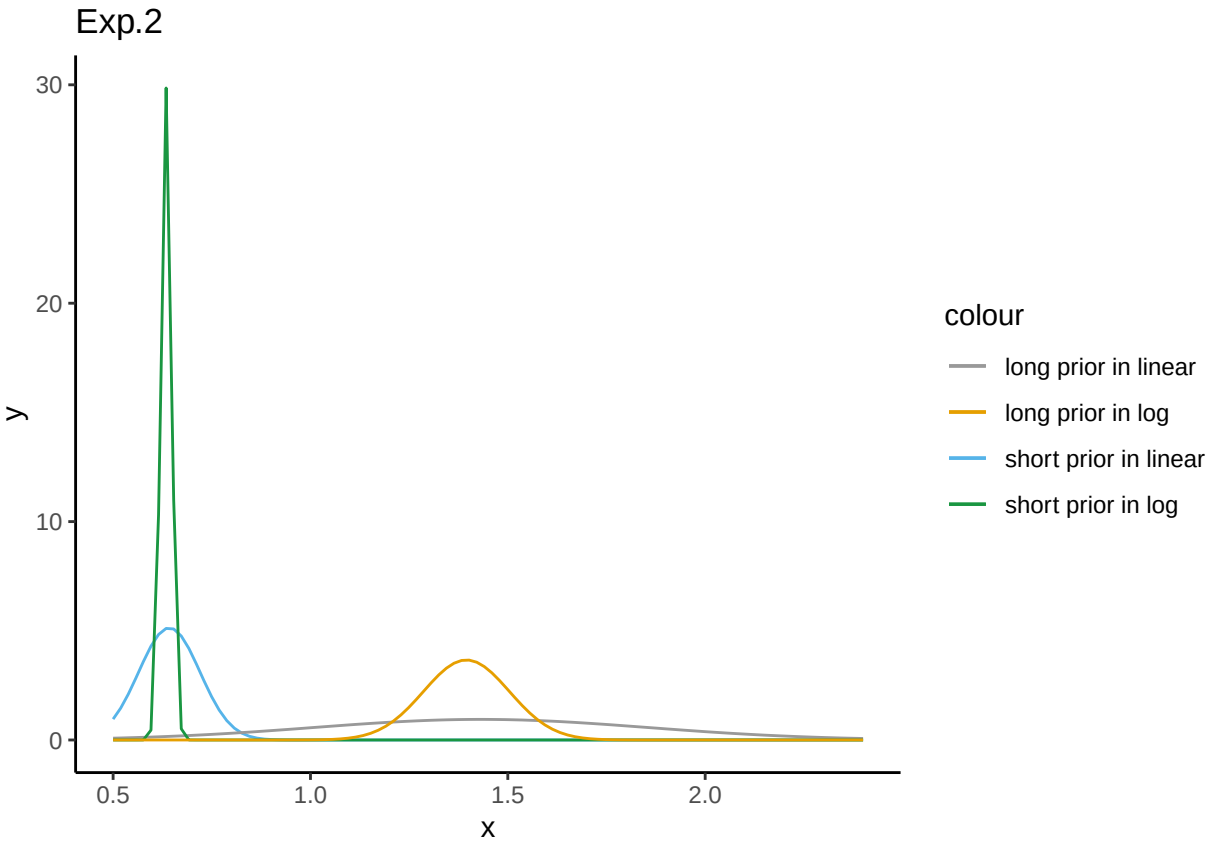
```
# baseline_BR_PredY$targetDur1 = round(baseline_BR_PredY$targetDur, 2)
# baseline_BR_PredY$targetDur1 = as.factor(baseline_BR_PredY$targetDur1)
# ggplot(baseline_BR_PredY%>%filter(ver == 'log'), aes(x = RP, y = targetDur1, fill = stat(x))) +
#   geom_density_ridges_gradient(quantile_lines=TRUE, quantiles = 2, scale = 3, size = 0.3, rel_min_hei
#   scale_fill_viridis_c(name = "RP", option = "C") +
#   labs(title = 'reproduction')+facet_grid(~Exp)
```

```
dens1 = split(mpar_baseline_BR, list(mpar_baseline_BR$Exp, mpar_baseline_BR$ver)) %>%
  map_df(~ tibble(x =seq(0.5, 2.4, 0.01),
                  density= dnorm(x, mean = 0, sd = sqrt(2)), .id="Exp,ver"))
```

```
  ggplot(data = data.frame(x = c(0.5, 2.4)), aes(x)) +
  geom_function(fun = dnorm, n = 100,  aes(col = "short prior in linear"), args = list(mean = mpar_basel
  geom_function(fun = dnorm, n = 100, aes(col ="long prior in linear"), args = list(mean = mpar_baselin
  geom_function(fun = dnorm, n = 100, aes(col = "short prior in log"),  args = list(mean = mpar_baselin
  geom_function(fun = dnorm, n = 100, aes(col = "long prior in log"), args = list(mean = mpar_baseline_l
```




```
ggplot(data = data.frame(x = c(0.5, 2.4)), aes(x)) +
  geom_function(fun = dnorm, n = 100, show.legend= TRUE, aes(col = "short prior in linear"), args = lis
  geom_function(fun = dnorm, n = 100, show.legend= TRUE, aes(col = "long prior in linear"), args = list
  geom_function(fun = dnorm, n = 100, show.legend= TRUE, aes(col = "short prior in log"), args = list(m
  geom_function(fun = dnorm, n = 100, show.legend= TRUE, aes(col = "long prior in log"), args = list(mea
```



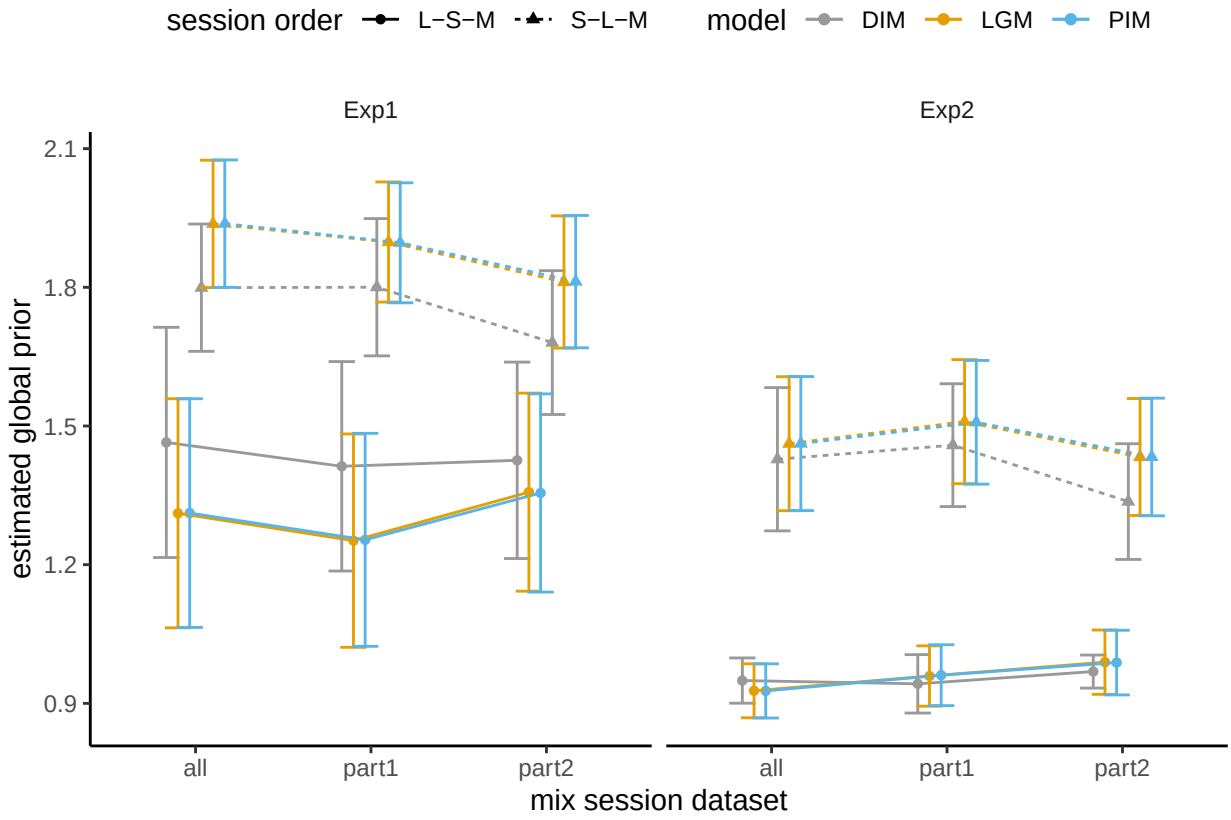

!!! The mean of short prior variance is very small.

# 2 IR session data prediction

## 2.1 load the estimated parameter of all models fitting to IR session data and combine the session order information

### 2.1.1 plot eastimated mean of global prior (LGM, DIM, PIM)

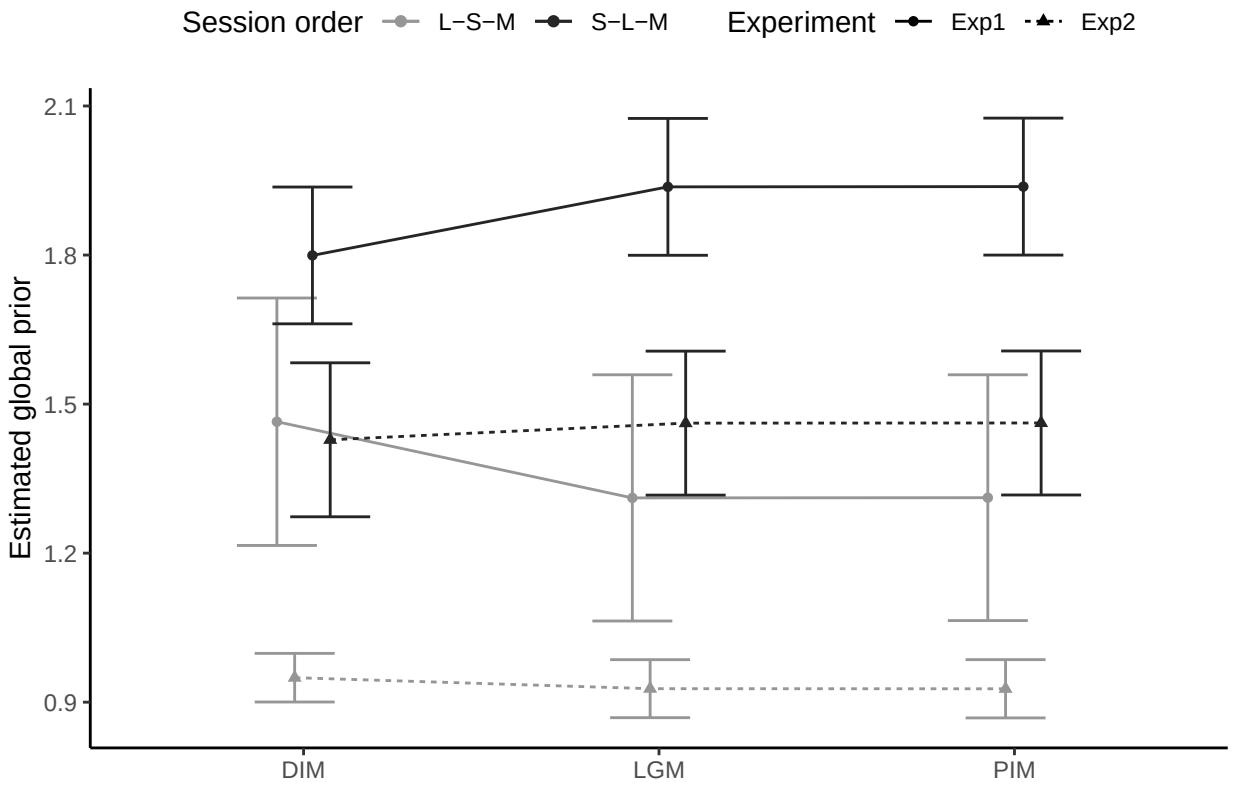
```
mywidth = 0.4
plt_gp_log = ggplot(mm_Baypar%>%filter(model !='IP', ver == 'log'), aes(part, m_mu_p_g,  group = intera
  geom_point(aes(color = model), position = position_dodge(width = mywidth))+ geom_line(aes(color = mod
  geom_errorbar(aes(ymin = m_mu_p_g -se_mu_p_g , ymax = m_mu_p_g + se_mu_p_g, color = model),position =
  theme_minimal()+ theme_new + colorSet3+
  labs(x = "mix session dataset",#one step(all) vs. two steps(part1, part2)
       y = "estimated global prior", linetype = 'session order', shape = "session order")+
  facet_wrap(~Exp)+ theme(legend.position = "top")
ggsave(file.path(figure_path,'plt_gp_log.png'), plt_gp_log, width = 4, height = 3)

plt_gp_log
```




```
mywidth = 0.2
plt_gp_log1 = ggplot(mm_Baypar%>%filter(model !='IP', ver == 'log', part == 'all'), aes(model, m_mu_p_g
  geom_point(aes(color = firstSession), position = position_dodge(width = mywidth))+ geom_line(aes(colo
  geom_errorbar(aes(ymin = m_mu_p_g -se_mu_p_g , ymax = m_mu_p_g + se_mu_p_g, color = firstSession),pos
  theme_minimal()+ theme_new + # colorSet3+
  labs(x = "",
       y = "Estimated global prior", linetype = 'Experiment', shape = 'Experiment', color = 'Session or
 theme(legend.position = "top")+
  scale_color_manual(values = mycolors)
```

```
## Warning: Ignoring unknown aesthetics: shape
```

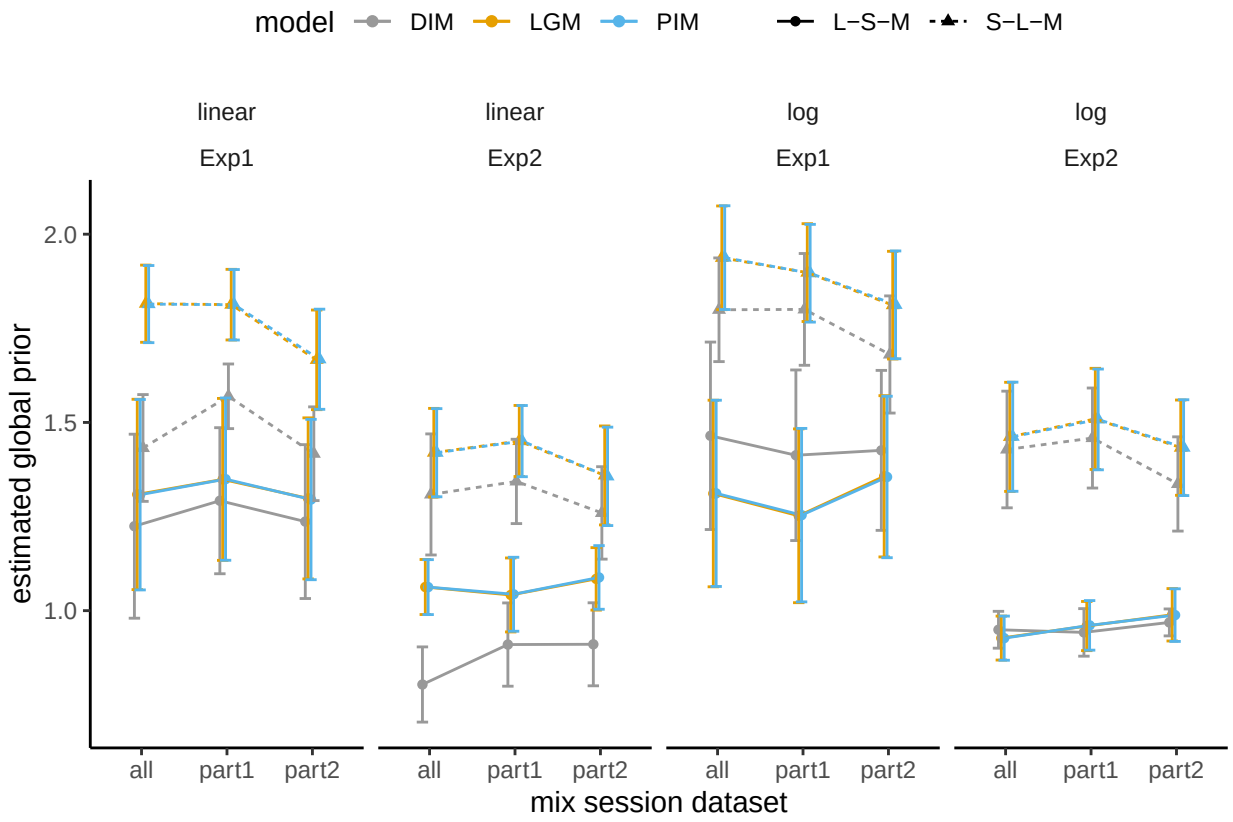
```
ggsave(file.path(figure_path,'plt_gp_log1.png'), plt_gp_log1, width = 5.5, height = 4)

plt_gp_log1
```




Interesting results!

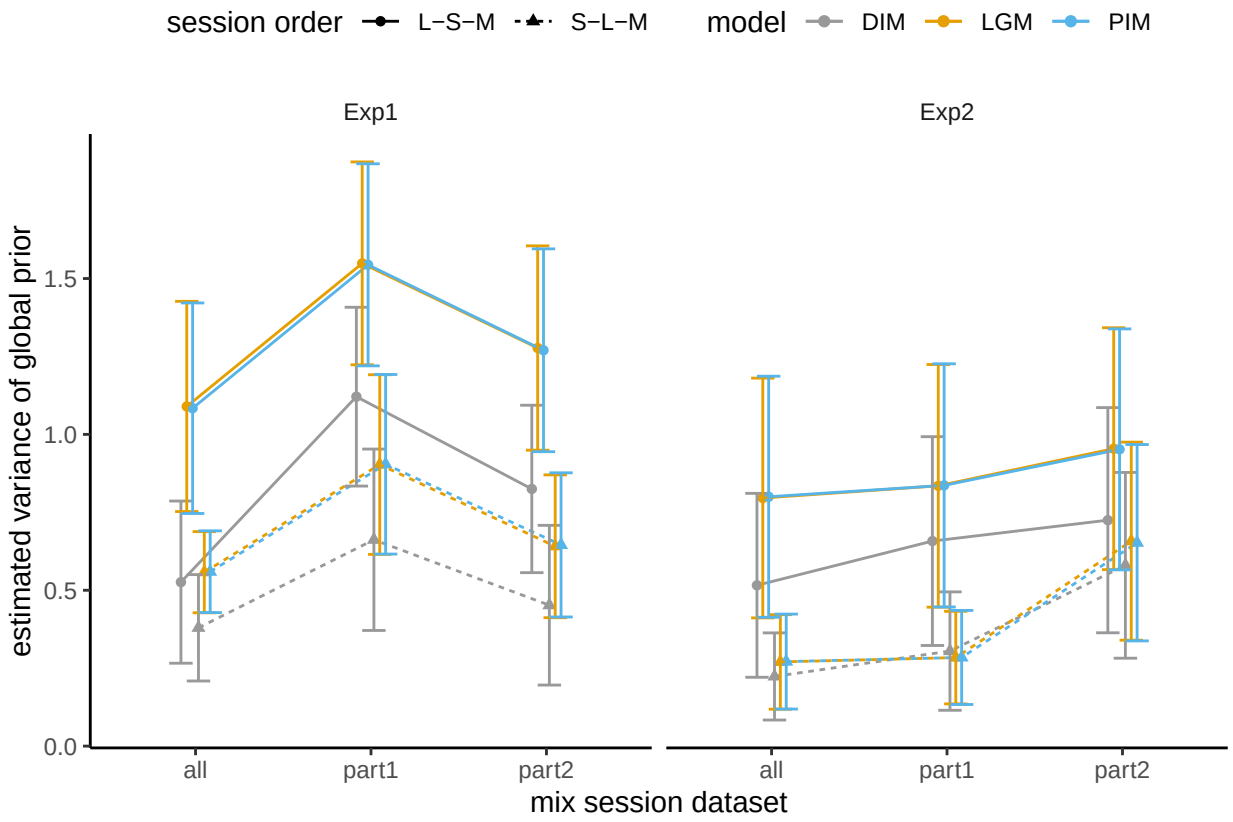
1. For both session order, we observed the similar pattern of global prior .
2. When long session was taken as the first session, the mean of global prior was smaller than the condition of short session as the first session.

```
plt_gp = ggplot(mm_Baypar%>%filter(model !='IP'), aes(part, m_mu_p_g,  group = interaction(model, firstS
  geom_point(aes(color = model), position = position_dodge(width = mywidth))+ geom_line(aes(color = mode
  geom_errorbar(aes(ymin = m_mu_p_g -se_mu_p_g , ymax = m_mu_p_g + se_mu_p_g, color = model, width=.8),p
  theme_minimal()+ theme_new + colorSet3+
  labs(x = "mix session dataset", #one step(all) vs. two steps(part1, part2)
       y = "estimated global prior", linetype = '', shape = "")+
  facet_wrap(ver~Exp, nrow = 1)
plt_gp
```




### 2.1.2 plot eastimated variance of global prior in IP model (IR data)

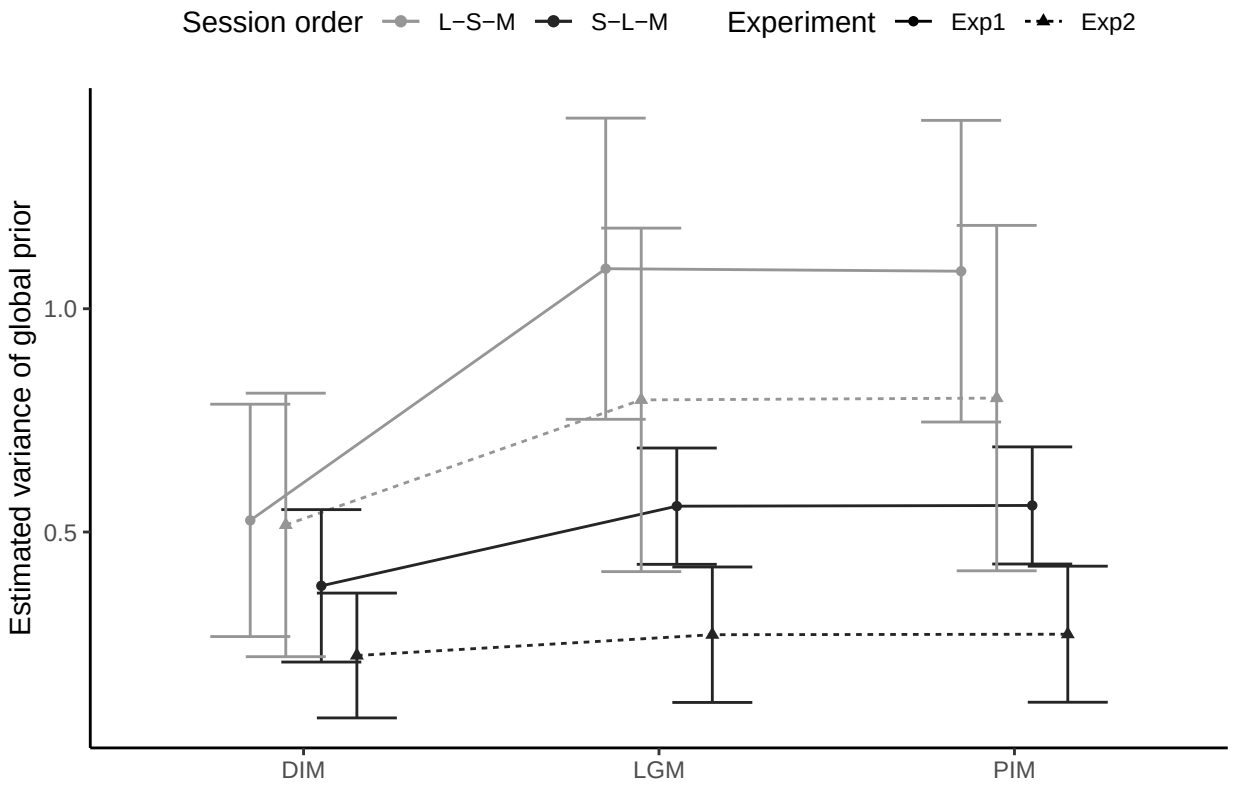
```
plt_gp_var_log = ggplot(mm_Baypar%>%filter(model !='IP', ver == 'log'), aes(part, m_sig2_p_g,  group = 
  geom_point(aes(color = model), position = position_dodge(width = mywidth))+ geom_line(aes(color = mod
  geom_errorbar(aes(ymin = m_sig2_p_g -se_sig2_p_g , ymax = m_sig2_p_g + se_sig2_p_g, color = model, wi
  theme_minimal()+ theme_new + colorSet3+
  labs(x = "mix session dataset", #one step(all) vs. two steps(part1, part2)
       y = "estimated variance of global prior", linetype = 'session order', shape = "session order")+
  facet_wrap(~Exp, nrow = 1)
plt_gp_var_log
```



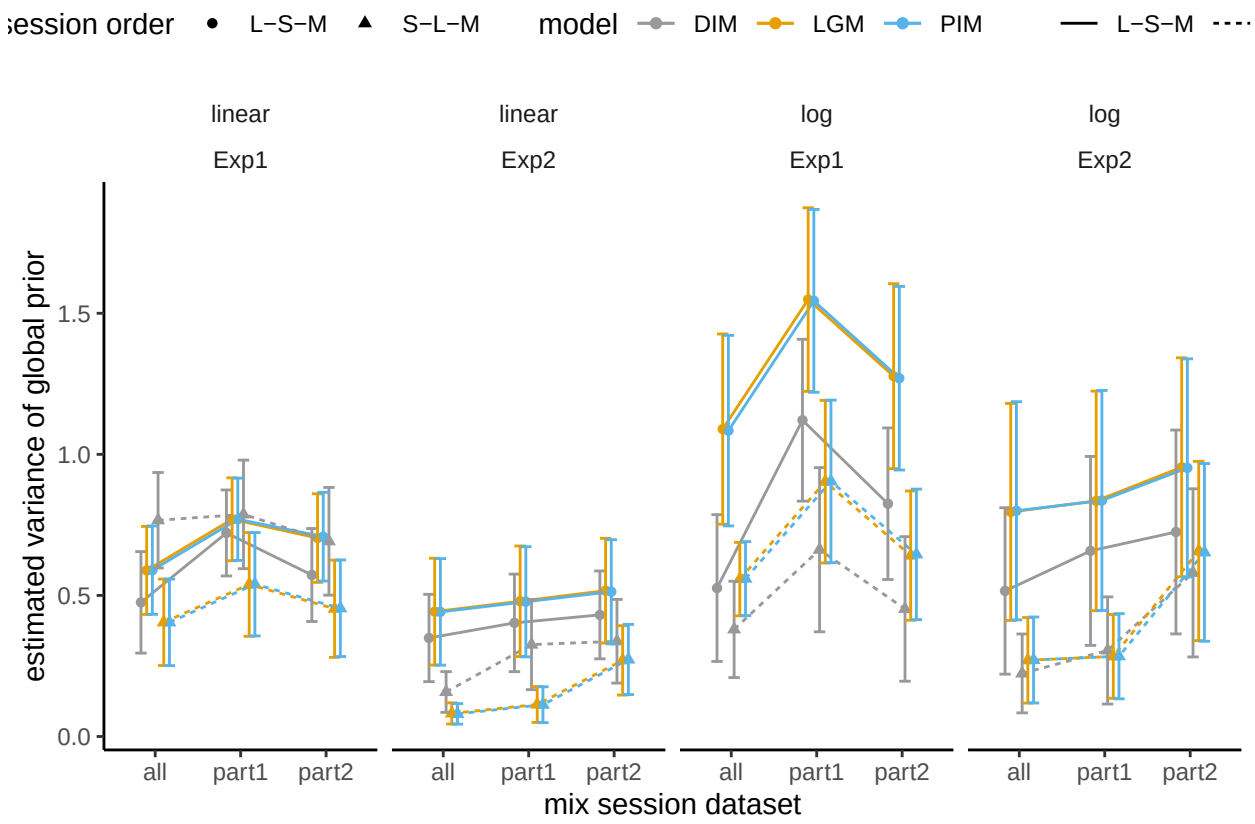

```
mywidth = 0.4
plt_gp_var_log1 = ggplot(mm_Baypar%>%filter(model !='IP', ver == 'log', part == 'all'), aes(model, m_sig
  geom_point(aes(color = firstSession), position = position_dodge(width = mywidth))+ geom_line(aes(color
  geom_errorbar(aes(ymin = m_sig2_p_g -se_sig2_p_g , ymax = m_sig2_p_g + se_sig2_p_g, color = firstSessi
  theme_minimal()+ theme_new +
  labs(x = "",
       y = "Estimated variance of global prior", linetype = 'Experiment', shape = 'Experiment',  color =
 theme(legend.position = "top")+
  scale_color_manual(values = mycolors)
```

```
## Warning: Ignoring unknown aesthetics: shape
```

```
ggsave(file.path(figure_path,'plt_gp_var_log1.png'), plt_gp_var_log1, width = 6, height = 4)

plt_gp_var_log1
```




```
plt_gp_var = ggplot(mm_Baypar%>%filter(model !='IP'), aes(part, m_sig2_p_g,  group = interaction(model,
  geom_point(aes(color = model), position = position_dodge(width = mywidth))+ geom_line(aes(color = mod
  geom_errorbar(aes(ymin = m_sig2_p_g -se_sig2_p_g , ymax = m_sig2_p_g + se_sig2_p_g, color = model, wi
  theme_minimal()+ theme_new + colorSet3+
  labs(x = "mix session dataset", #one step(all) vs. two steps(part1, part2)
       y = "estimated variance of global prior", linetype = '', shape = "session order")+
  facet_wrap(ver~Exp, nrow = 1)
plt_gp_var
```



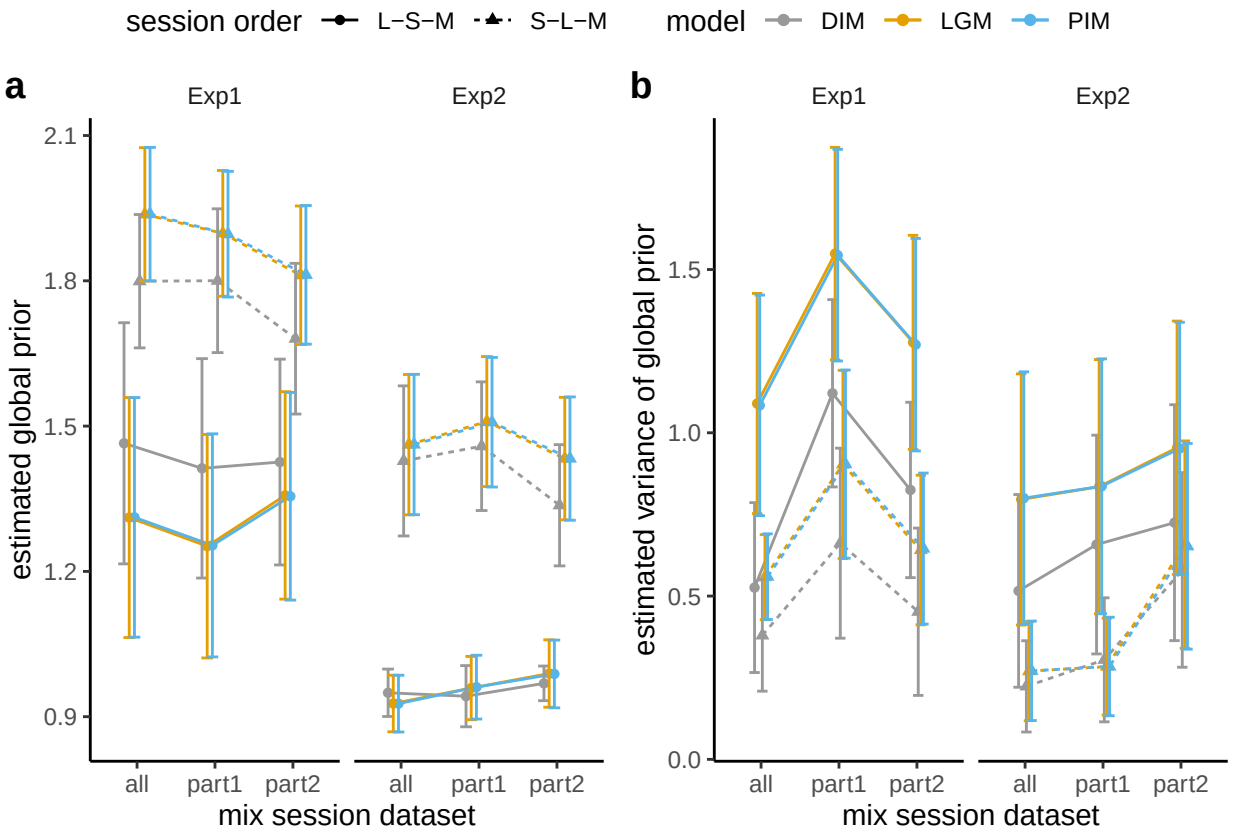

The variance of global prior in Exp1 for part1, part2, and all are high, medium and low, respectively.

But Exp2 have different pattern, compared to Exp1. part2 has higher variance of global prior than part1. How to explain this?

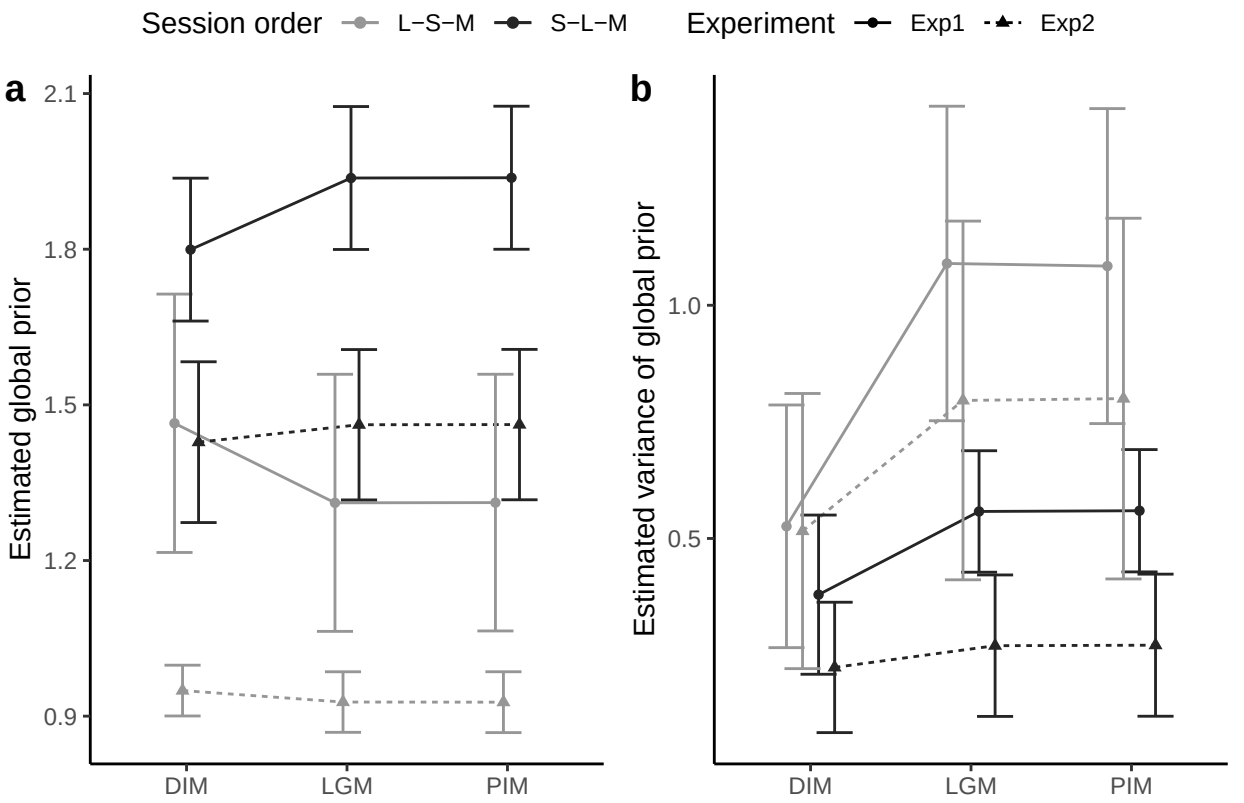
#### 2.1.2.1 difference of global prior variance based on session order (L-S-M - S-L-M)

Interesting results! (How to explain that the DIM Exp1 is different with other conditions? )

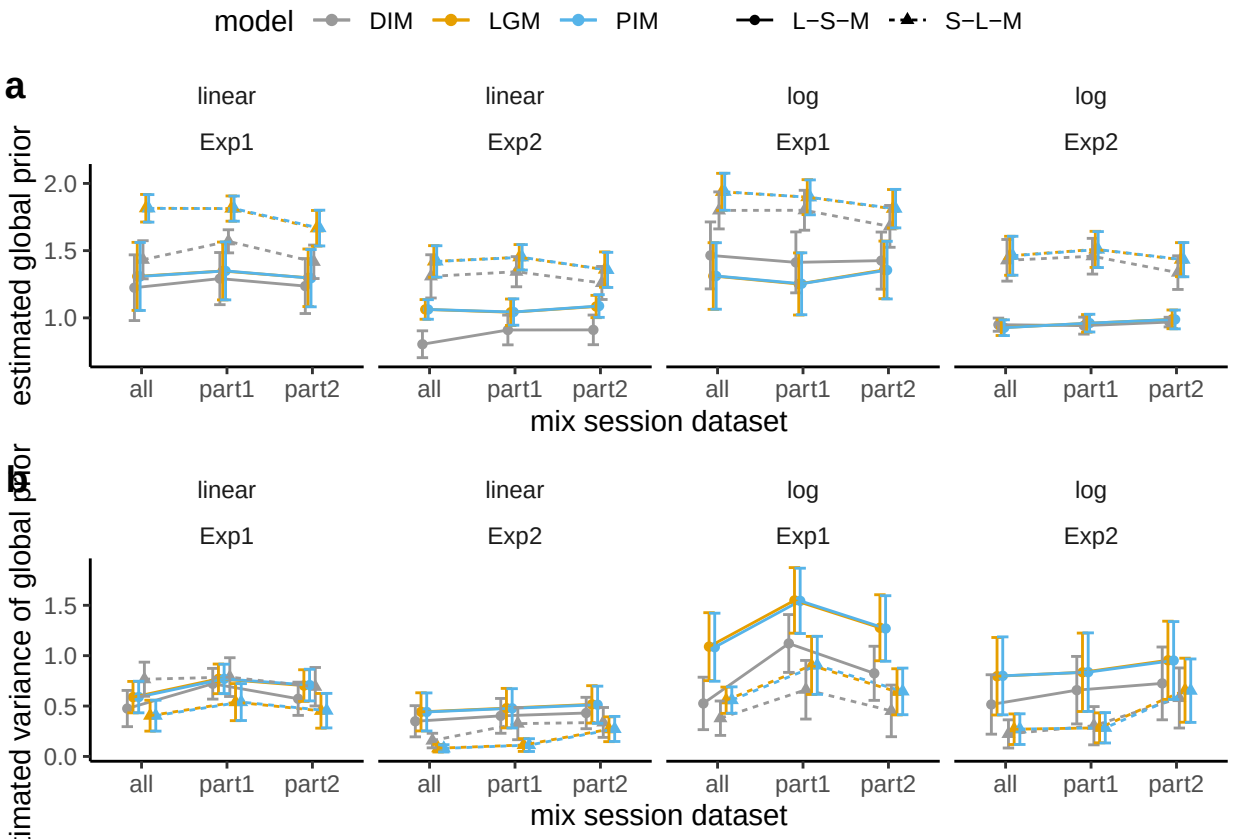
```
fig_globalPrior_log <- ggarrange(plt_gp_log, plt_gp_var_log, common.legend = TRUE, nrow = 1,ncol = 2, la
ggsave(file.path(figure_path,'fig_globalPrior_log.png'), fig_globalPrior_log, width = 8, height = 4)
fig_globalPrior_log
```




```
fig_globalPrior_log1 <- ggarrange(plt_gp_log1, plt_gp_var_log1, common.legend = TRUE, nrow = 1,ncol = 2
ggsave(file.path(figure_path,'fig_globalPrior_log1.png'), fig_globalPrior_log1, width = 8, height = 4)
  fig_globalPrior_log1
```




```
fig_globalPrior <- ggarrange(plt_gp, plt_gp_var, common.legend = TRUE, nrow = 2,ncol = 1, labels = c('a
ggsave(file.path(figure_path,'fig_globalPrior.png'), fig_globalPrior, width = 8, height = 8)
fig_globalPrior
```




### Anonva on estimated global prior

```
#ezANOVA(data = m_Baypar %>% filter(ver == 'log', model == 'LGM',Exp == 'Exp1'), dv = mu_p_g, wid=Nsub,
```

```
ezANOVA(data = m_Baypar %>% filter(ver == 'log', model == 'LGM'), dv = mu_p_g, wid=Nsub, within_full=.(
```

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

## Warning: Converting "Exp" to factor for ANOVA.

## Warning: The column supplied as the wid variable contains non-unique values
## across levels of the supplied between-Ss variables. Automatically fixing this by
## generating unique wid labels.

## Warning: Collapsing data to cell means first using variables supplied to
## "within_full", then collapsing the resulting means to means for the cells
## supplied to "within".

## Coefficient covariances computed by hccm()

## $ANOVA
##   Effect DFn DFd        F          p p<.05       ges
## 1    Exp   1  30 5.258721 0.02901915     * 0.1491467
##
## $`Levene's Test for Homogeneity of Variance`
##   DFn DFd       SSn    SSd        F         p p<.05
## 1   1  30 0.4494795 1.7973 7.502579 0.0102681     *
```

```
ezANOVA(data = m_Baypar %>% filter(ver == 'log', model == 'PIM'), dv = mu_p_g, wid=Nsub, within_full=.(
```

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

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

## Warning: The column supplied as the wid variable contains non-unique values
## across levels of the supplied between-Ss variables. Automatically fixing this by
## generating unique wid labels.

## Warning: Collapsing data to cell means first using variables supplied to
## "within_full", then collapsing the resulting means to means for the cells
## supplied to "within".

## Coefficient covariances computed by hccm()

## $ANOVA
##   Effect DFn DFd        F          p p<.05       ges
## 1    Exp   1  30 5.260748 0.02899055     * 0.1491956
##
## $`Levene's Test for Homogeneity of Variance`
##   DFn DFd       SSn      SSd        F          p p<.05
## 1   1  30 0.4479074 1.796824 7.478317 0.01037891     *
```

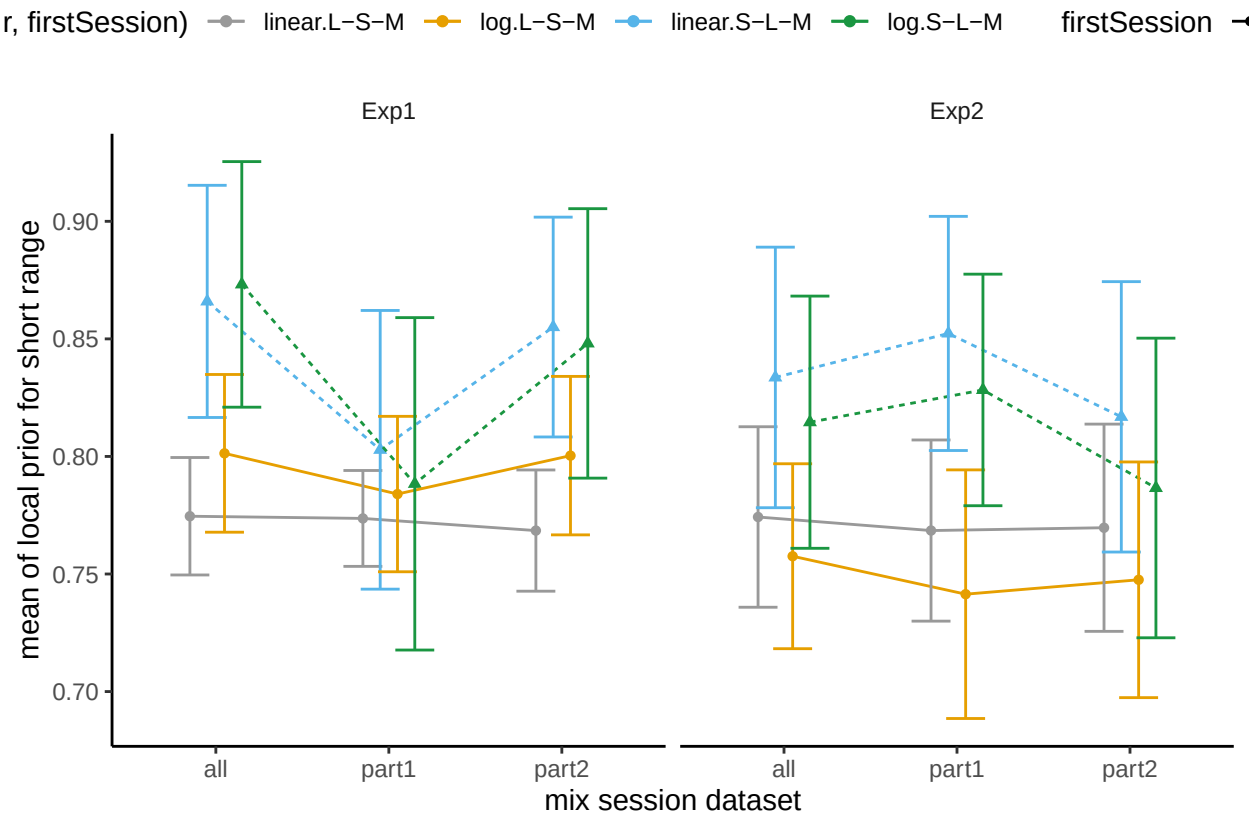
```
ezANOVA(data = m_Baypar %>% filter(ver == 'log', model == 'DIM'), dv = mu_p_g, wid=Nsub, within_full=.(
```

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

## Warning: Converting "Exp" to factor for ANOVA.

## Warning: The column supplied as the wid variable contains non-unique values
## across levels of the supplied between-Ss variables. Automatically fixing this by
## generating unique wid labels.

## Warning: Collapsing data to cell means first using variables supplied to
## "within_full", then collapsing the resulting means to means for the cells
## supplied to "within".

## Coefficient covariances computed by hccm()

## $ANOVA
##   Effect DFn DFd        F          p p<.05     ges
## 1    Exp   1  30 7.362694 0.01092512     * 0.19706
##
## $`Levene's Test for Homogeneity of Variance`
##   DFn DFd       SSn      SSd        F          p p<.05
## 1   1  30 0.2317371 2.336406 2.975558 0.09482077
```

## 2.2 plot parameters for Indipendent Prior(IP) model on IR data

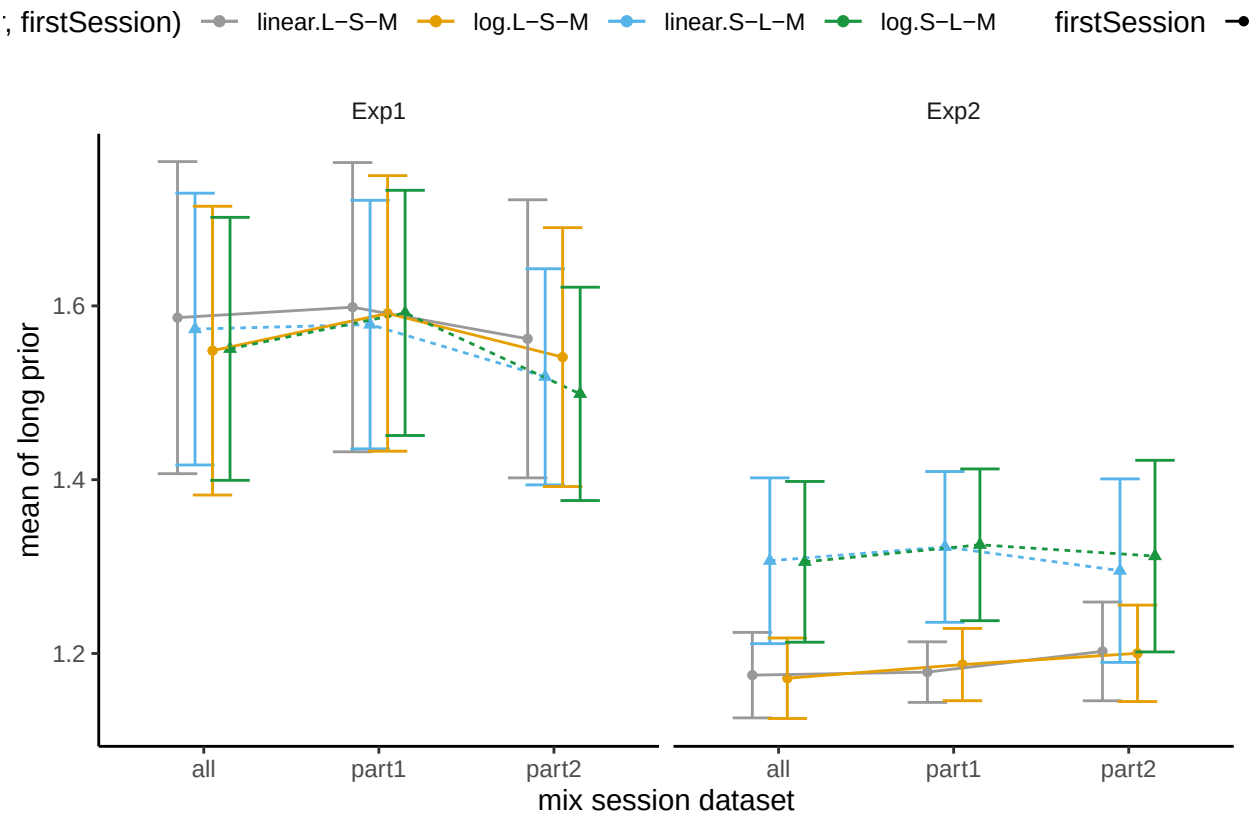
### 2.2.1 plot eastimated mean of prior for short range in IP model (IR data)

```
ggplot(mm_Baypar%>%filter(model =='IP'), aes(part, m_mu_p_s, group = interaction(Exp, firstSession, ver
  geom_point(aes(color = interaction(ver, firstSession), shape = firstSession), position = position_dod
  geom_line(aes(color = interaction(ver, firstSession), linetype = firstSession), position = position_d
  geom_errorbar(aes(ymin = m_mu_p_s -se_mu_p_s , ymax = m_mu_p_s + se_mu_p_s, color = interaction(ver,
  theme_minimal()+ theme_new + colorSet4+
  labs(x = "mix session dataset",
       y = "mean of local prior for short range")+
  facet_wrap(~Exp)
```




### 2.2.2 plot eastimated mean of prior for long range in IP model (IR data)

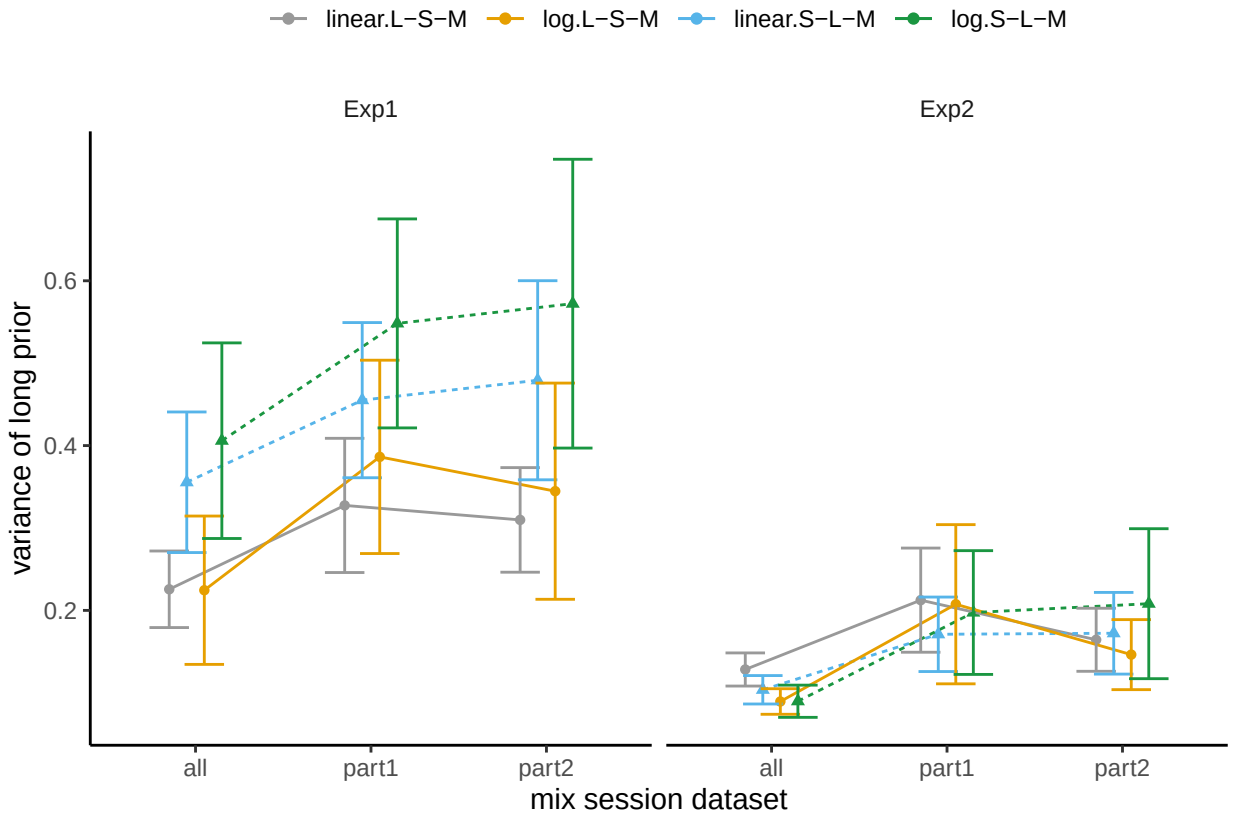
```
ggplot(mm_Baypar%>%filter(model =='IP'), aes(part, m_mu_p_l, group = interaction(Exp, firstSession, ver
  geom_point(aes(color = interaction(ver, firstSession), shape = firstSession), position = position_dod
  geom_line(aes(color = interaction(ver, firstSession), linetype = firstSession), position = position_d
  geom_errorbar(aes(ymin = m_mu_p_l -se_mu_p_l , ymax = m_mu_p_l + se_mu_p_l, color = interaction(ver, 
  theme_minimal()+ theme_new + colorSet4+
  labs(x = "mix session dataset",#one step(all) vs. two steps(part1, part2)
       y = "mean of long prior")+
  facet_wrap(~Exp)
```




### plot variance of local prior for long range in IP model

```
# ggplot(mm_Baypar%>%filter(model =='IP'), aes(part, m_sig2_p_l))+
#   geom_bar(aes(fill = ver), stat = "identity",position = 'dodge')+
#   geom_errorbar(aes(ymin = m_sig2_p_l -se_sig2_p_l, ymax = m_sig2_p_l + se_sig2_p_l, color = ver, wid
#   theme_minimal()+ theme_new +
#   labs(x = "one step(all) vs. two steps(part1, part2)",
#        y = "variance of long prior in IP model")+
#   facet_wrap(firstSession~Exp)
```

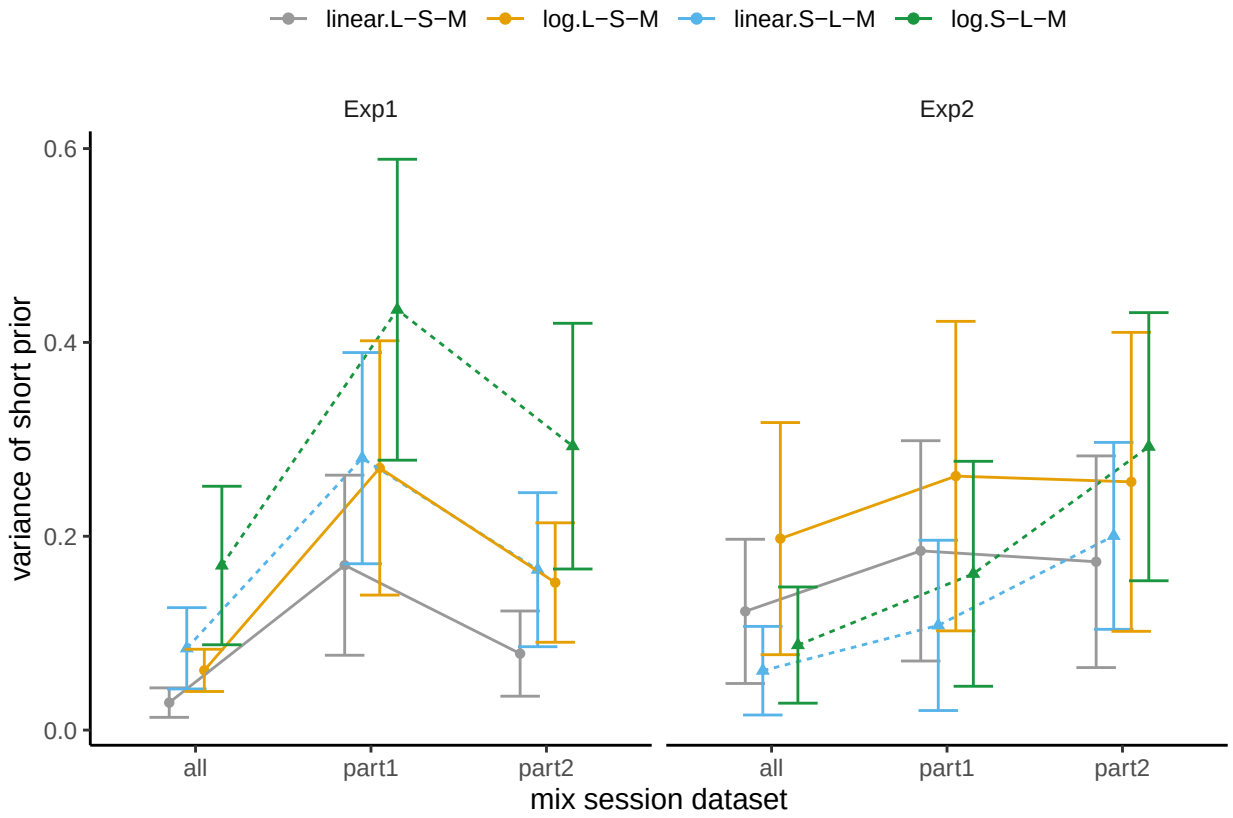
```
ggplot(mm_Baypar%>%filter(model =='IP'), aes(part, m_sig2_p_l, group = interaction(Exp, firstSession, ve
  geom_point(aes(color = interaction(ver, firstSession), shape = firstSession), position = position_dodg
  geom_line(aes(color = interaction(ver, firstSession), linetype = firstSession), position = position_do
  geom_errorbar(aes(ymin = m_sig2_p_l -se_sig2_p_l , ymax = m_sig2_p_l + se_sig2_p_l, color = interactio
  theme_minimal()+ theme_new + colorSet4+
  labs(x = "mix session dataset",
       y = "variance of long prior", color = "")+
  facet_wrap(~Exp)+
  scale_linetype(guide = "none")+scale_shape(guide = "none")
```




### 2.2.3 plot variance of local prior for short range in IP model

```
# ggplot(mm_Baypar%>%filter(model =='IP'), aes(part, m_sig2_p_s))+
#   geom_bar(aes(fill = ver), stat = "identity",position = 'dodge')+
#   geom_errorbar(aes(ymin = m_sig2_p_s -se_sig2_p_s, ymax = m_sig2_p_s + se_sig2_p_s, color = ver, wid
#   theme_minimal()+ theme_new +
#   labs(x = "one step(all) vs. two steps(part1, part2)",
#        y = "variance of prior for short range")+
#   facet_wrap(firstSession~Exp)
```

```
ggplot(mm_Baypar%>%filter(model =='IP'), aes(part, m_sig2_p_s, group = interaction(Exp, firstSession, ve
  geom_point(aes(color = interaction(ver, firstSession), shape = firstSession), position = position_dodg
  geom_line(aes(color = interaction(ver, firstSession), linetype = firstSession), position = position_do
  geom_errorbar(aes(ymin = m_sig2_p_s -se_sig2_p_s , ymax = m_sig2_p_s + se_sig2_p_s, color = interactio
  theme_minimal()+ theme_new + colorSet4+
  labs(x = "mix session dataset", #one step(all) vs. two steps(part1, part2)
       y = "variance of short prior", color = "")+
  facet_wrap(~Exp)+
  scale_linetype(guide = "none")+scale_shape(guide = "none")
```




# 3 Prediction results from models for IR session data

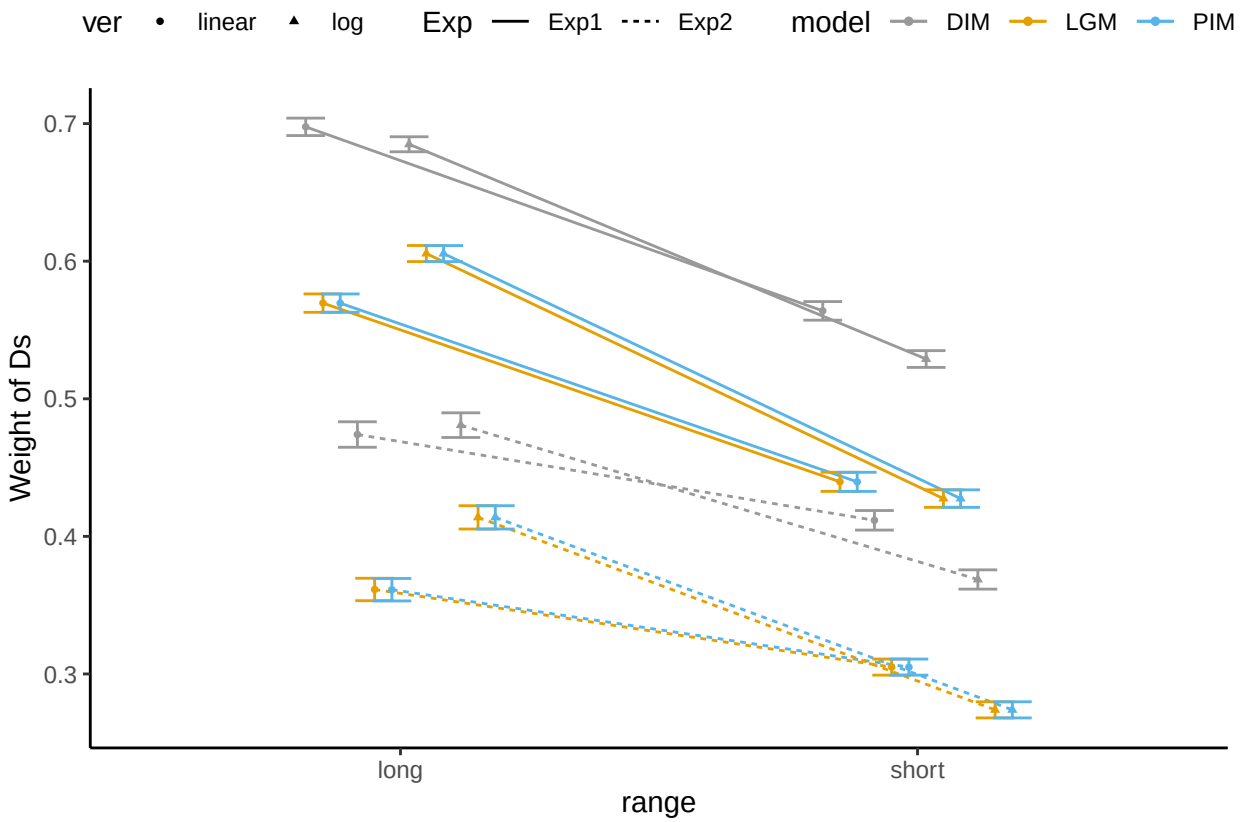
## 3.1 Weights of global prior

### 3.1.1 plot weight of priors

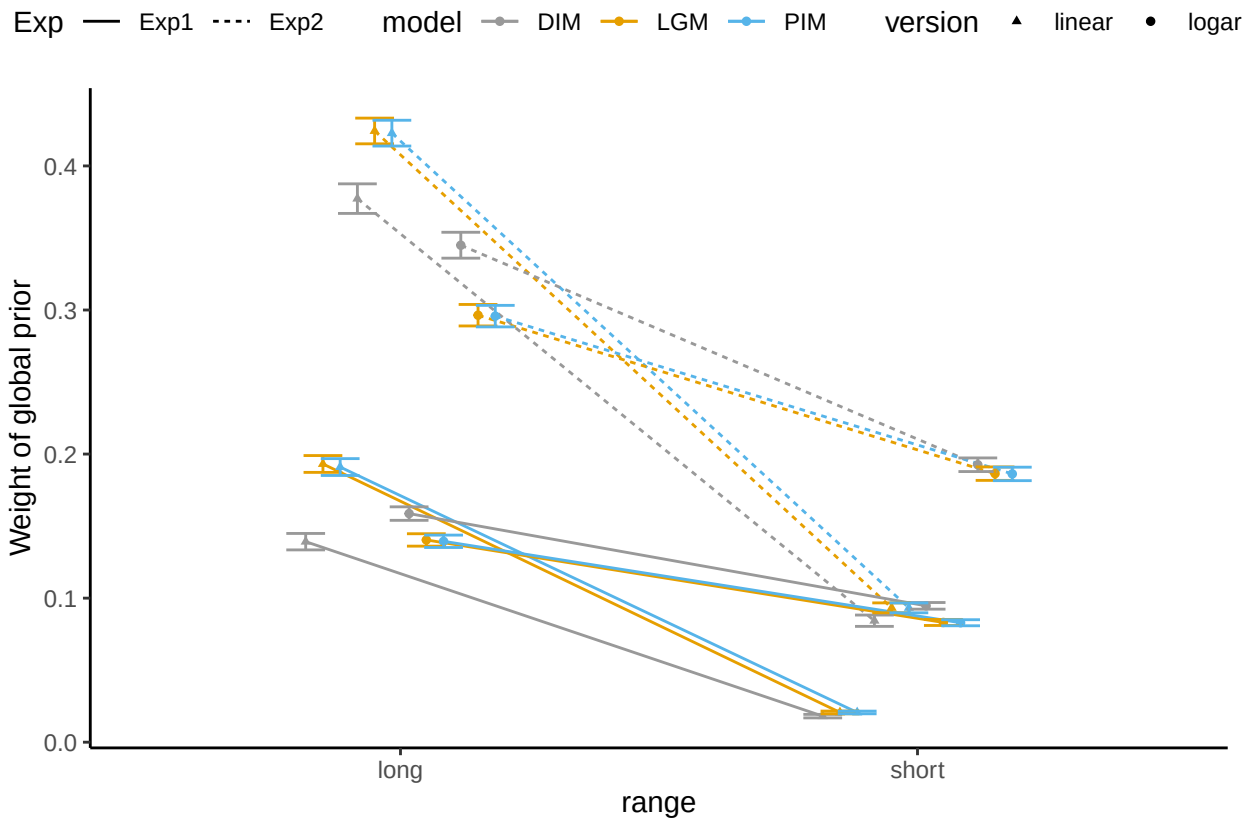
```
mweight = PredY_mixed_dat %>%group_by(model, Exp, ver, part, range) %>%
  dplyr::summarize(m_W_Ds = mean(W_Ds),
                   m_W_P_G = mean(W_P_G),
                   m_W_L = mean(W_L),
                   n = n(),
                   se_W_Ds= sd(W_Ds)/sqrt(n-1),
                   se_W_P_G = sd(W_P_G)/sqrt(n-1),
                   se_W_L = sd(W_L)/sqrt(n-1))
```

```
## `summarise()` has grouped output by 'model', 'Exp', 'ver', 'part'. You can override using the `.grou
```

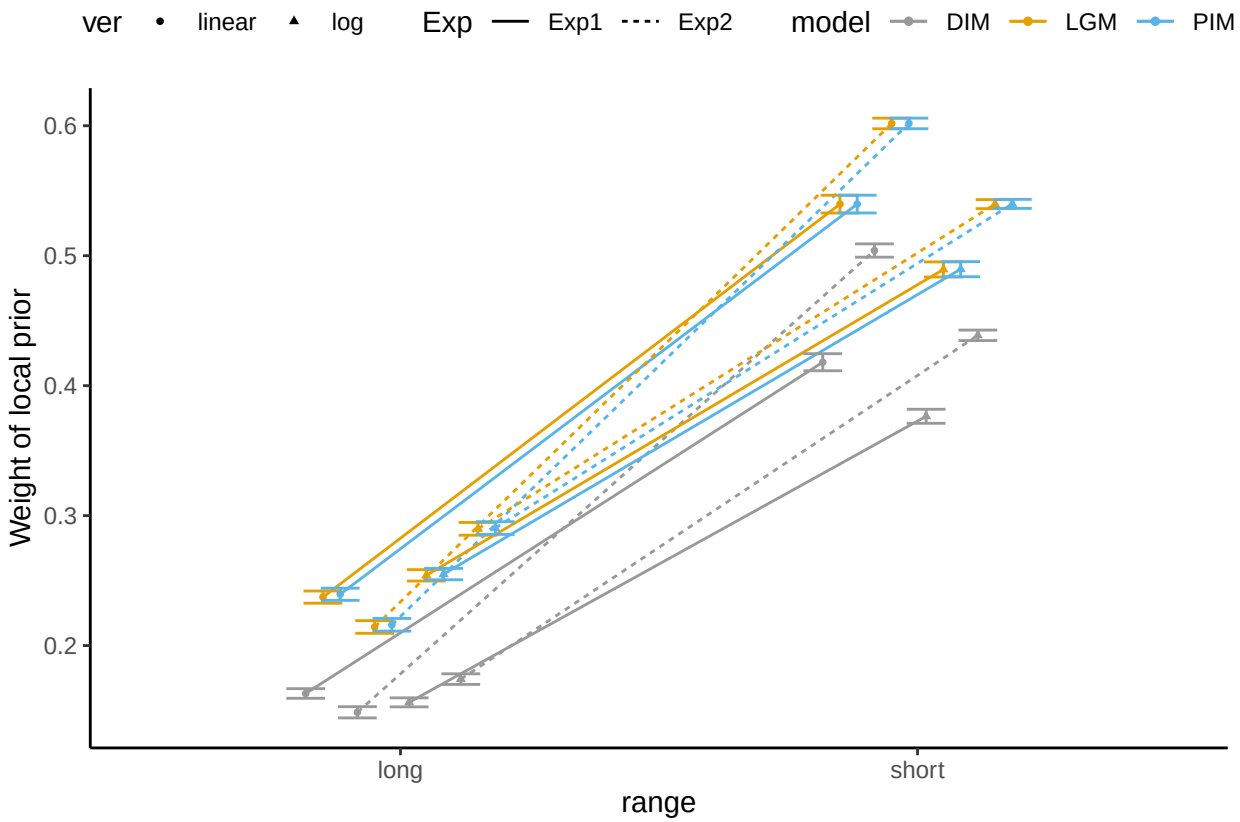
```
plt_W_Ds = mweight %>% ggplot(aes(range, m_W_Ds, group = interaction(model, Exp, ver), color = model, sh
  geom_line(aes(linetype = Exp),position = position_dodge(width = mywidth))+ geom_point(position = posit
  geom_errorbar(aes(ymin = m_W_Ds - se_W_Ds, ymax = m_W_Ds + se_W_Ds), position = position_dodge(width =
  xlab('range') + ylab('Weight of Ds')+ theme_new
plt_W_Ds
```



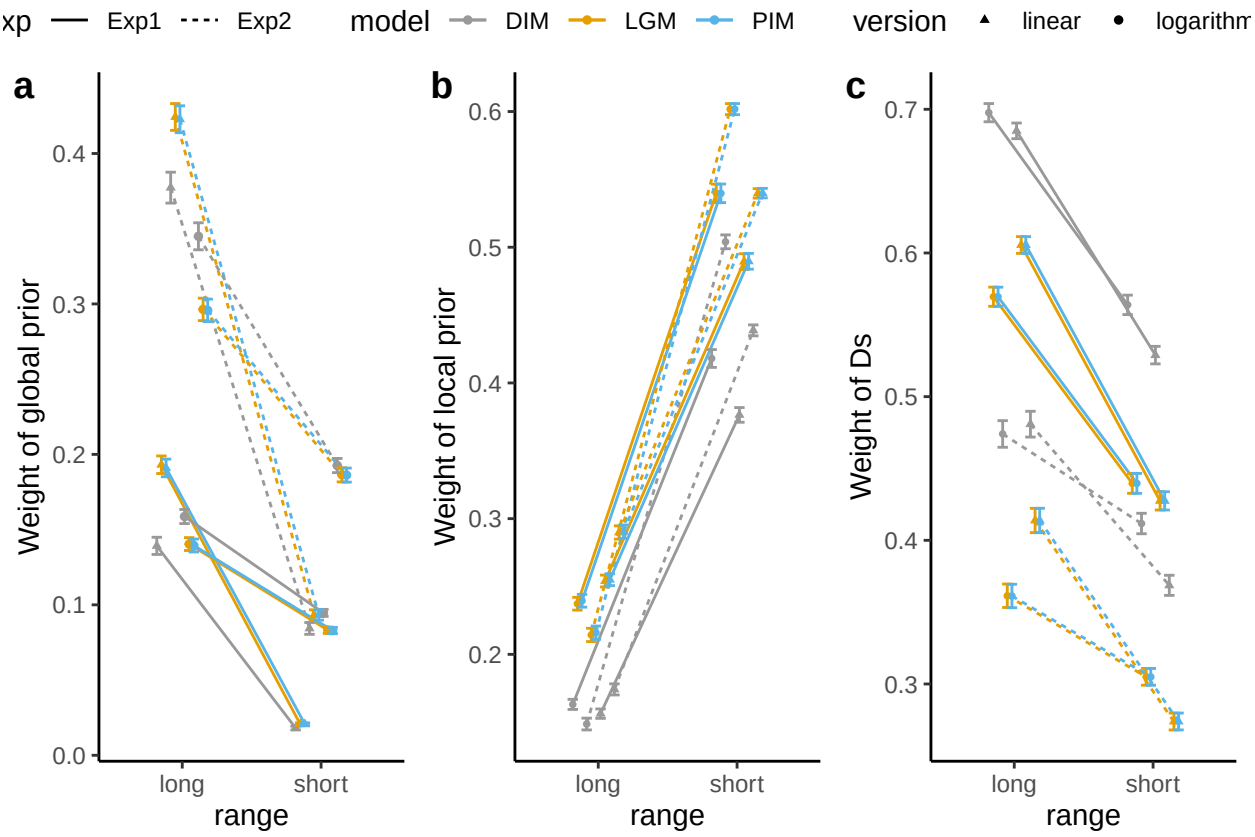

```
plt_W_P_G = mweight %>% ggplot(aes(range, m_W_P_G, group = interaction(model, Exp, ver), color = model,
  geom_errorbar(aes(ymin = m_W_P_G - se_W_P_G, ymax = m_W_P_G + se_W_P_G ), position = position_dodge(wi
  xlab('range') + ylab('Weight of global prior')+ labs(shape = 'version')+ theme_new+
   scale_shape_manual(labels = c("linear", "logarithmic"), values = c("triangle", "circle"))
plt_W_P_G
```




```
plt_W_L = mweight %>% ggplot(aes(range, m_W_L, group = interaction(model, Exp, ver), color =model, shap
  geom_line(aes(linetype = Exp), position = position_dodge(width = mywidth))+ geom_point(position = posi
  geom_errorbar(aes(ymin = m_W_L - se_W_L, ymax = m_W_L + se_W_L),  position = position_dodge(width = my
  xlab('range') + ylab('Weight of local prior')+ theme_new
plt_W_L
```




```
fig_weight <- ggarrange(plt_W_P_G, plt_W_L, plt_W_Ds, common.legend = TRUE, nrow = 1,ncol = 3, labels =
ggsave(file.path(figure_path,'fig_weight.png'), fig_weight, width = 8, height = 4)
fig_weight
```

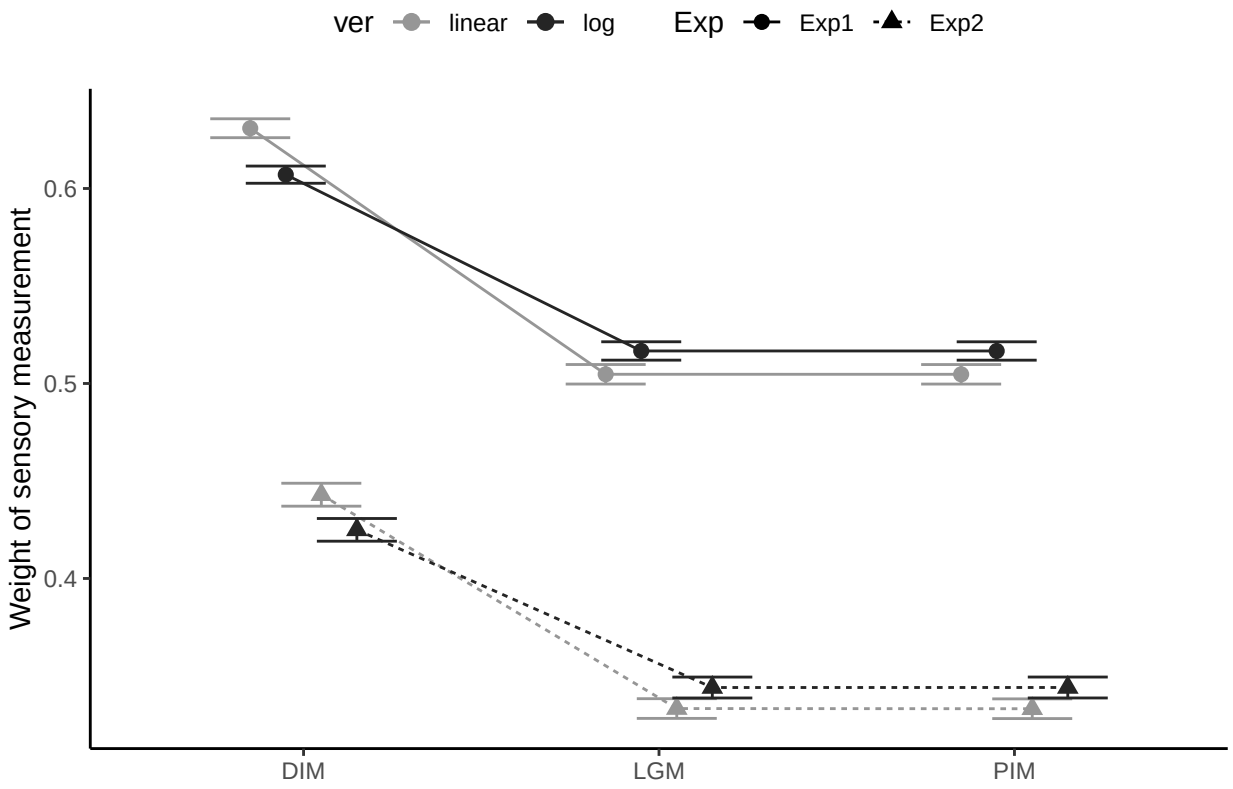



### 3.1.2 plot weight of priors

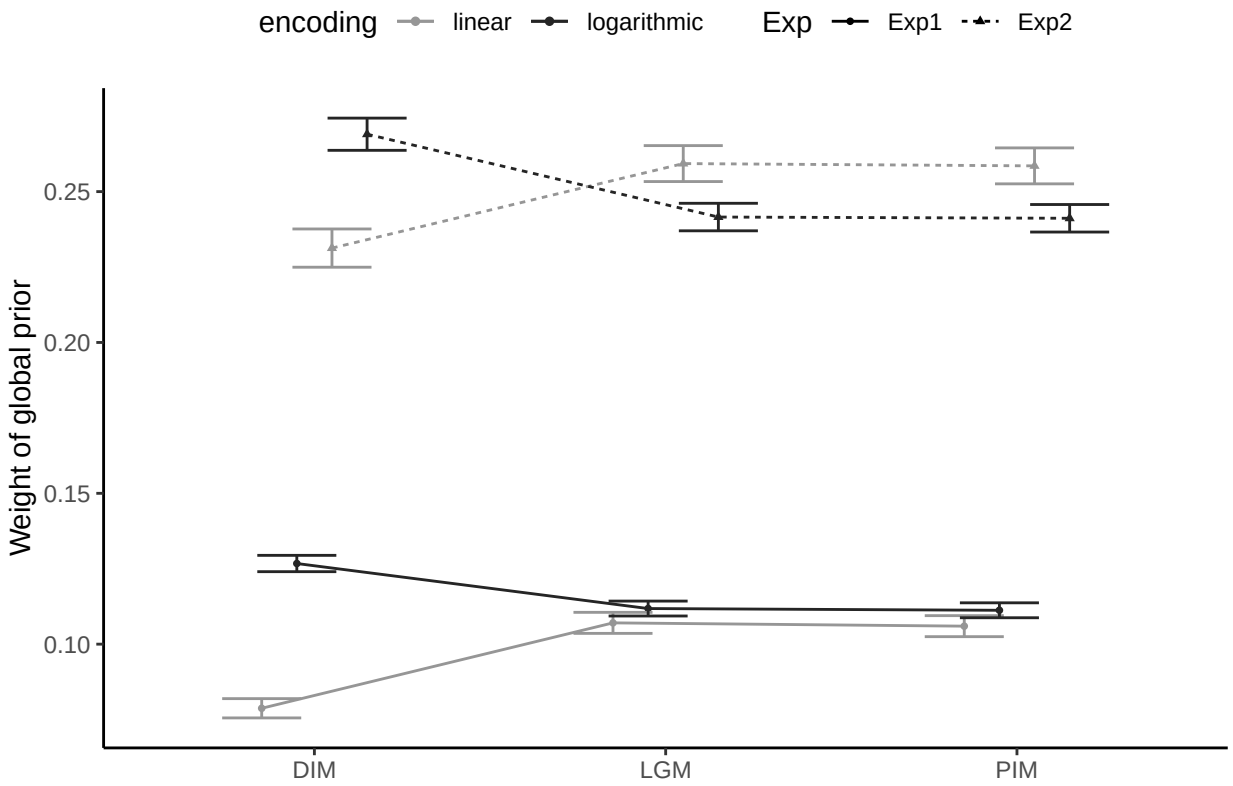
```
mweight2 = PredY_mixed_dat %>%group_by(model, Exp, ver, part) %>%
  dplyr::summarize(m_W_Ds = mean(W_Ds),
                   m_W_P_G = mean(W_P_G),
                   m_W_L = mean(W_L),
                   n = n(),
                   se_W_Ds= sd(W_Ds)/sqrt(n-1),
                   se_W_P_G = sd(W_P_G)/sqrt(n-1),
                   se_W_L = sd(W_L)/sqrt(n-1))
```

```
## `summarise()` has grouped output by 'model', 'Exp', 'ver'. You can override using the `.groups` argu
```

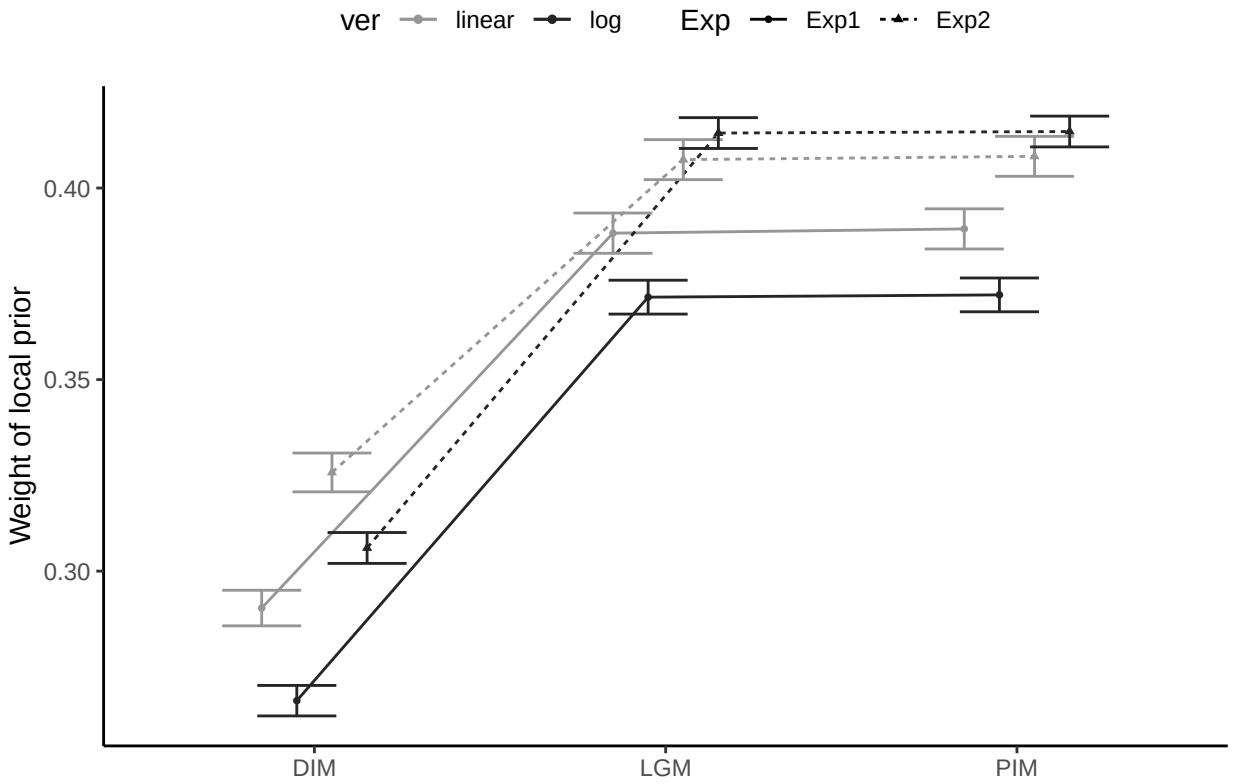
```
plt_W_Ds = mweight2 %>% ggplot(aes(model, m_W_Ds, group = interaction(ver, Exp), color = ver, shape = E
  geom_line(aes(linetype = Exp),position = position_dodge(width = mywidth))+ geom_point(position = posi
  geom_errorbar(aes(ymin = m_W_Ds - se_W_Ds, ymax = m_W_Ds + se_W_Ds), position = position_dodge(width =
  scale_color_manual(values = mycolors)+
  xlab('') + ylab('Weight of sensory measurement')+ theme_new

plt_W_Ds
```



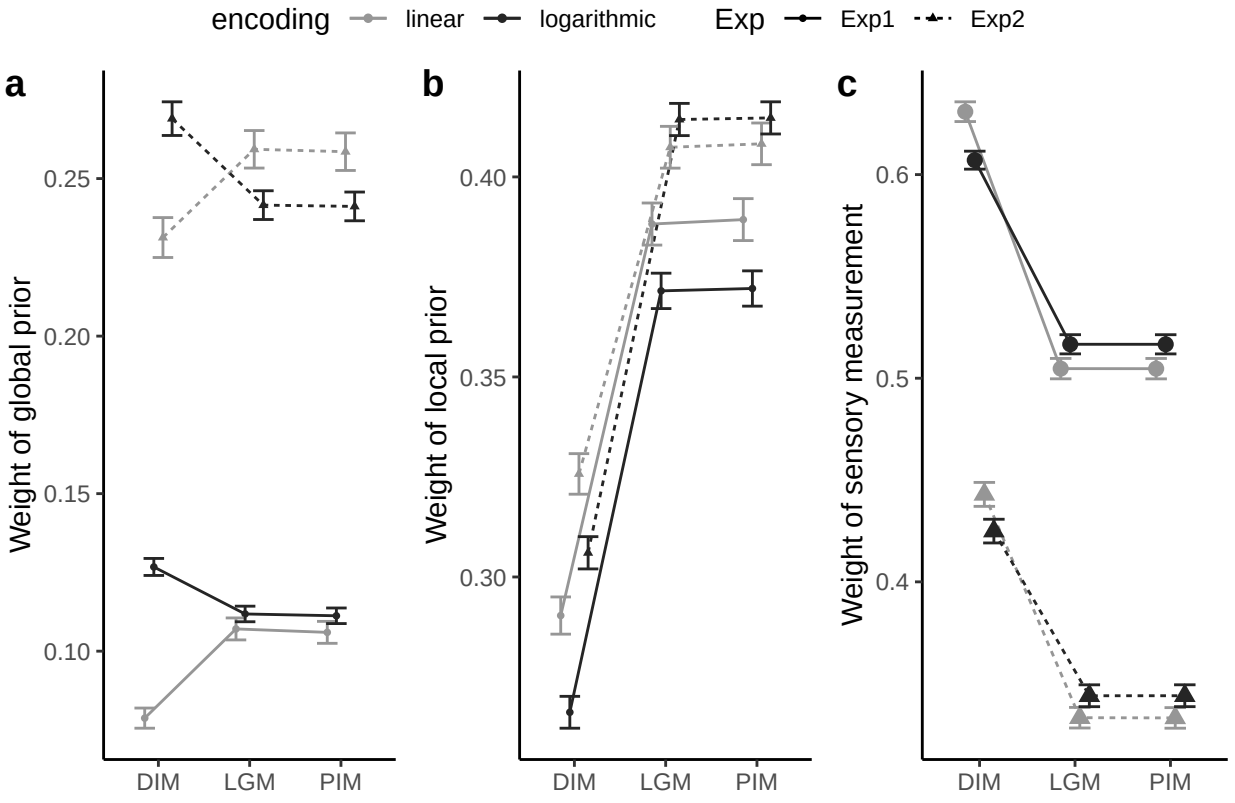

```
plt_W_P_G = mweight2  %>% ggplot(aes(model, m_W_P_G, group = interaction(ver, Exp), color = ver, shape =
  geom_errorbar(aes(ymin = m_W_P_G - se_W_P_G, ymax = m_W_P_G + se_W_P_G ), position = position_dodge(wi
  xlab('') + ylab('Weight of global prior')+ labs(color = 'encoding')+ theme_new+
   scale_color_manual(labels = c("linear", "logarithmic"), values = mycolors)
plt_W_P_G
```



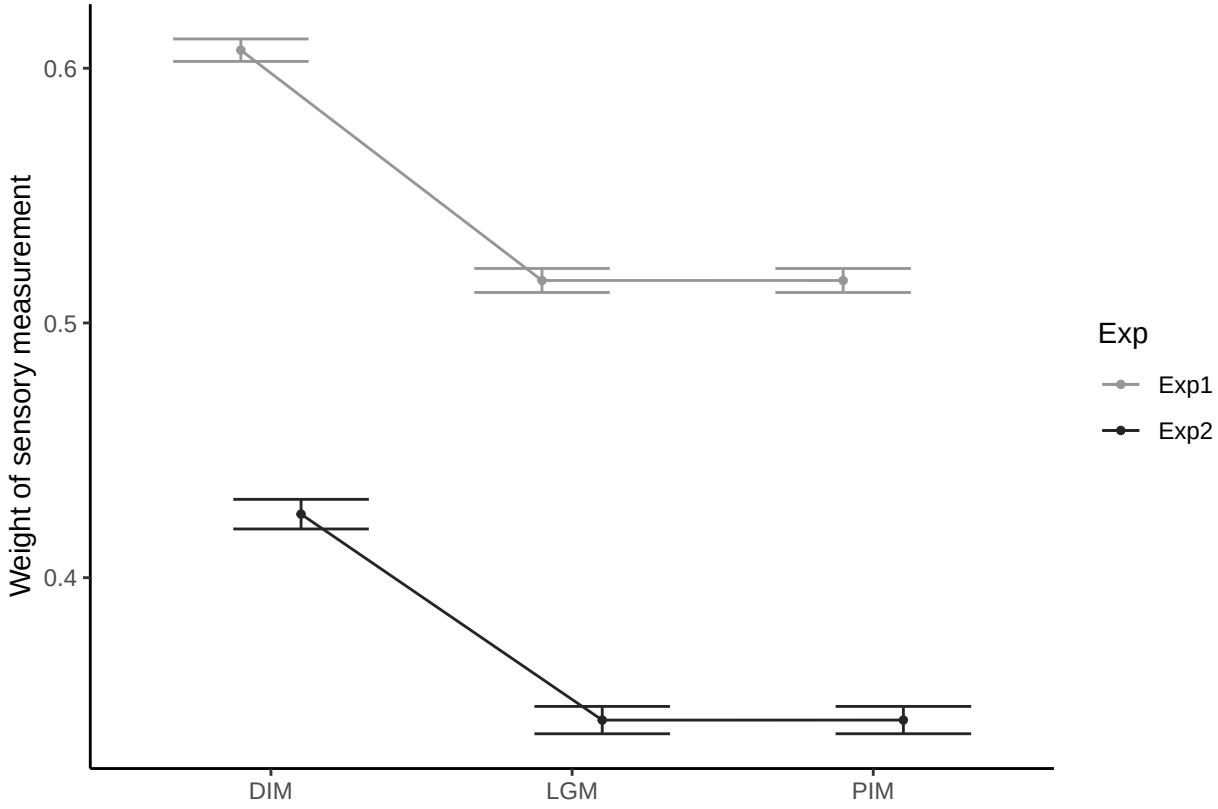

```
plt_W_L = mweight2  %>% ggplot(aes(model, m_W_L, group = interaction(ver, Exp), color = ver, shape = Exp
  geom_line(aes(linetype = Exp), position = position_dodge(width = mywidth))+
  geom_point(position = position_dodge(width = mywidth), size = 1) +
  geom_errorbar(aes(ymin = m_W_L - se_W_L, ymax = m_W_L + se_W_L),  position = position_dodge(width = my
  xlab('') + ylab('Weight of local prior')+ theme_new +
  scale_color_manual(values = mycolors)
plt_W_L
```



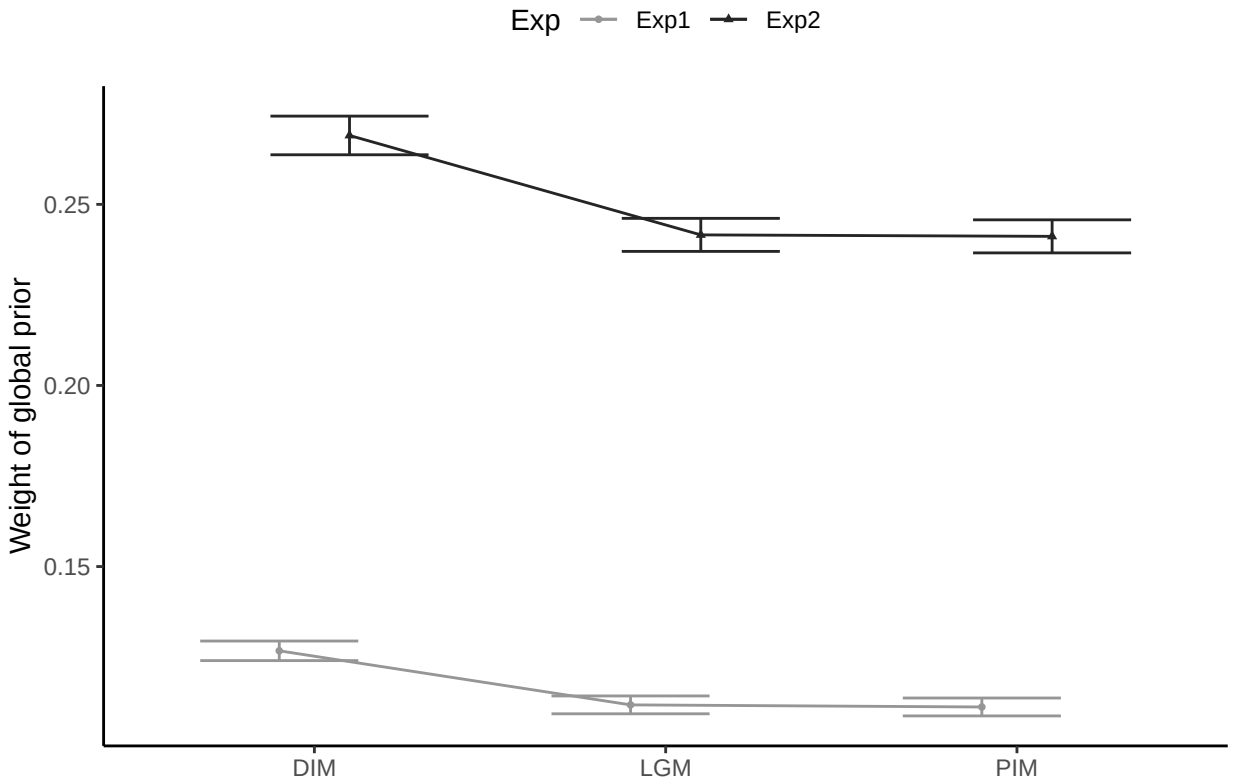

```
fig_weight2 <- ggarrange(plt_W_P_G, plt_W_L, plt_W_Ds, common.legend = TRUE, nrow = 1,ncol = 3, labels =
ggsave(file.path(figure_path,'fig_weight2.png'), fig_weight2, width = 8, height = 3.5)
fig_weight2
```



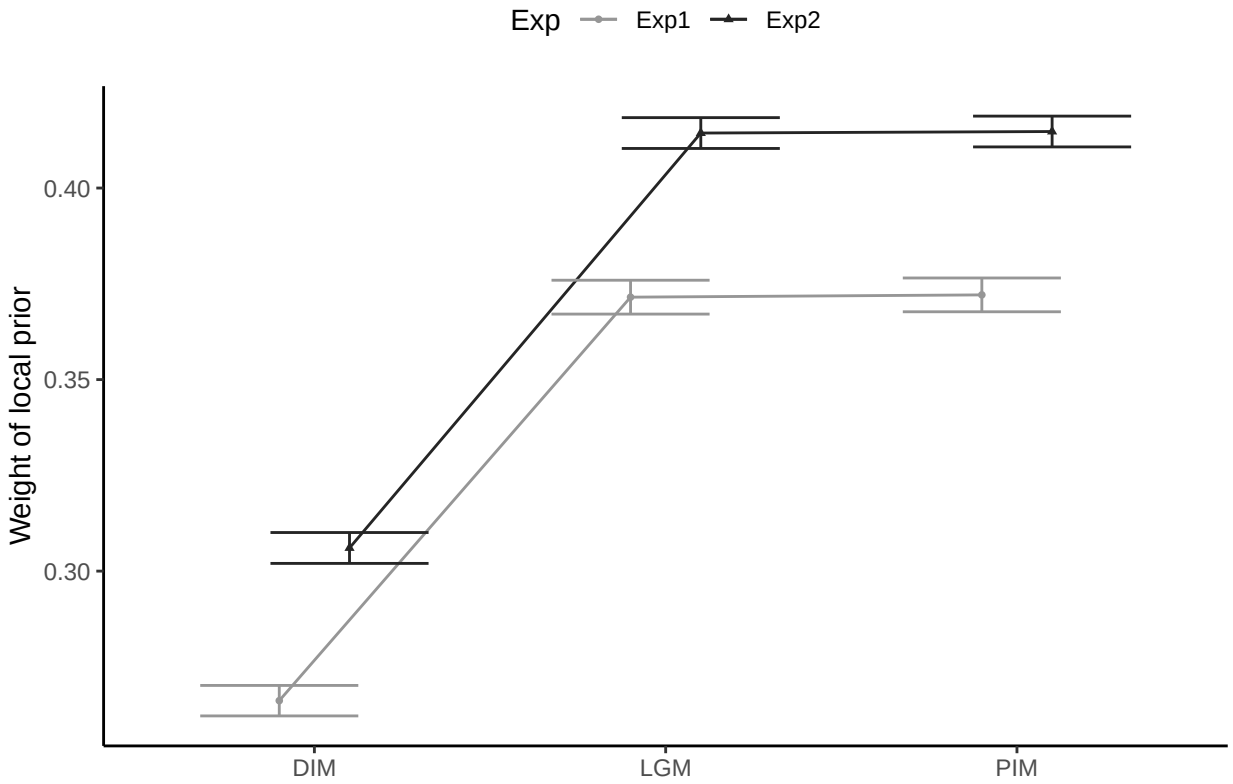
```
plt_W_Ds_log = mweight2%>%filter(ver == 'log') %>% ggplot(aes(model, m_W_Ds, group = Exp, color = Exp))
  geom_line(position = position_dodge(width = mywidth))+ geom_point(position = position_dodge(width = my
  geom_errorbar(aes(ymin = m_W_Ds - se_W_Ds, ymax = m_W_Ds + se_W_Ds), position = position_dodge(width =
  xlab('') + ylab('Weight of sensory measurement')+
  scale_color_manual(values = mycolors)
plt_W_Ds_log
```



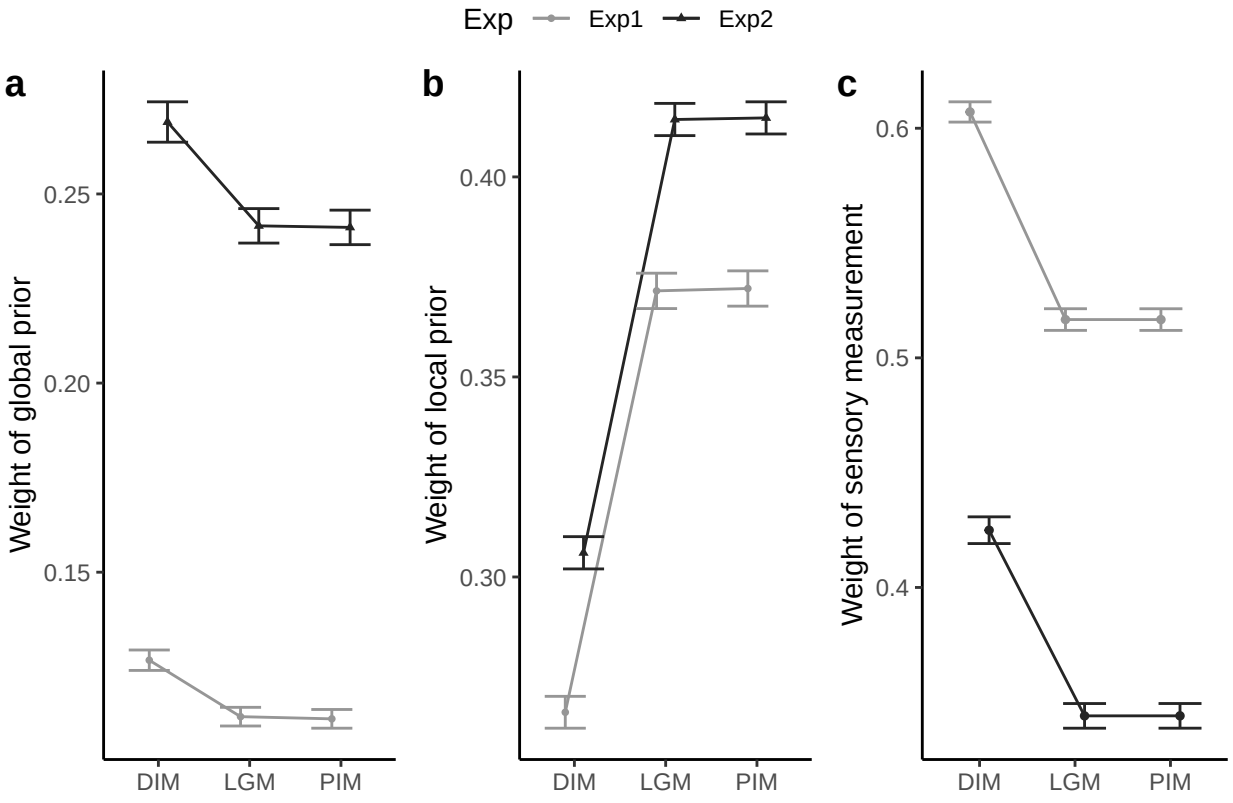

```
plt_W_P_G_log = mweight2 %>%filter(ver == 'log') %>% ggplot(aes(model, m_W_P_G, group =  Exp, color = E
  geom_errorbar(aes(ymin = m_W_P_G - se_W_P_G, ymax = m_W_P_G + se_W_P_G ), position = position_dodge(w
  xlab('') + ylab('Weight of global prior')+ theme_new+
  scale_color_manual(values = mycolors)
plt_W_P_G_log
```




```
plt_W_L_log = mweight2  %>%filter(ver == 'log') %>% ggplot(aes(model, m_W_L, group = Exp, color = Exp, s
  geom_line(position = position_dodge(width = mywidth))+
  geom_point(position = position_dodge(width = mywidth), size = 1) +# colorSet3+
  geom_errorbar(aes(ymin = m_W_L - se_W_L, ymax = m_W_L + se_W_L),  position = position_dodge(width = my
  xlab('') + ylab('Weight of local prior')+ theme_new +
  scale_color_manual(values = mycolors)
plt_W_L_log
```




```
fig_weight2_log <- ggarrange(plt_W_P_G_log, plt_W_L_log, plt_W_Ds_log, common.legend = TRUE, nrow = 1,n
ggsave(file.path(figure_path,'fig_weight2_log.png'), fig_weight2_log, width = 8, height = 3.5)
fig_weight2_log
```



### 3.1.3 anova on weight of prior

```
mweight_sub = PredY_mixed_dat %>% filter(part == 'all', ver == 'log')%>%group_by(NSub, model, Exp) %>%
  dplyr::summarize(m_W_Ds = mean(W_Ds),
                   m_W_P_G = mean(W_P_G),
                   m_W_L = mean(W_L),
                   n = n(),
                   se_W_Ds= sd(W_Ds)/sqrt(n-1),
                   se_W_P_G = sd(W_P_G)/sqrt(n-1),
                   se_W_L = sd(W_L)/sqrt(n-1))
```

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

```
write.csv(mweight_sub, paste0(getwd(), "/generated/mweight_sub.csv"))
```

```
ezANOVA(data = mweight_sub, dv= m_W_P_G, wid=NSub, within=.(model), between= Exp)
```

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

```
## Warning: You have removed one or more levels from variable "model". Refactoring
## for ANOVA.
```

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

```
## Warning: The column supplied as the wid variable contains non-unique values
## across levels of the supplied between-Ss variables. Automatically fixing this by
## generating unique wid labels.
```

```
## $ANOVA
##      Effect DFn DFd         F            p p<.05          ges
## 2       Exp   1  30  5.023909 3.255093e-02     * 0.1426714527
## 3     model   2  60 17.951258 7.755993e-07     * 0.0037366012
## 4 Exp:model   2  60  1.534760 2.238473e-01       0.0003205596
##
## $`Mauchly's Test for Sphericity`
##      Effect            W            p p<.05
## 3     model 0.0008397713 2.513825e-45     *
## 4 Exp:model 0.0008397713 2.513825e-45     *
##
## $`Sphericity Corrections`
##      Effect     GGe        p[GG] p[GG]<.05       HFe        p[HF] p[HF]<.05
## 3     model 0.50021 0.0001979258         * 0.5002318 0.0001978776         *
## 4 Exp:model 0.50021 0.2250122346           0.5002318 0.2250129513
```

```
library(BayesFactor)
```

```
## Loading required package: coda
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

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

```
## ************
## Welcome to BayesFactor 0.9.12-4.2. If you have questions, please contact Richard Morey (richarddmorey
##
```

```
## Type BFManual() to open the manual.
## ************
```

```
##
## Attaching package: 'BayesFactor'
```

```
## The following object is masked from 'package:loo':
##
##     compare
```

```
mweight_sub$NSub = as.factor(mweight_sub$NSub)
mweight_sub$Exp = as.factor(mweight_sub$Exp)
bf <- anovaBF(m_W_P_G ~ model*Exp + NSub, data = mweight_sub, whichRandom = "NSub")
```

```
## Warning: data coerced from tibble to data frame
```

```
summary(bf)
```

```
## Bayes factor analysis
## --------------
## [1] model + NSub                  : 0.1133451 ±0.64%
## [2] Exp + NSub                    : 23864.4   ±14.23%
## [3] model + Exp + NSub            : 2512.819  ±1.31%
## [4] model + Exp + model:Exp + NSub : 397.3928  ±1.97%
##
## Against denominator:
##   m_W_P_G ~ NSub
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
bf_model <- bf[3]/bf[2]
bf_Exp <- bf[3]/bf[1]
bf_int <- bf[4]/bf[3]  #Prior type * ragne interaction
```

```
ezANOVA(data = mweight_sub, dv= m_W_Ds, wid=NSub, within=.(model), between= Exp)
```

```
## Warning: You have removed one or more levels from variable "model". Refactoring
## for ANOVA.
```

```
## Warning: The column supplied as the wid variable contains non-unique values
## across levels of the supplied between-Ss variables. Automatically fixing this by
## generating unique wid labels.
```

```
## $ANOVA
##      Effect DFn DFd           F            p p<.05          ges
## 2       Exp   1  30   5.3984281 2.711920e-02     * 0.1515929006
## 3     model   2  60 153.2971336 2.624022e-24     * 0.0347599547
## 4 Exp:model   2  60   0.5348344 5.885326e-01       0.0001256246
##
## $`Mauchly's Test for Sphericity`
##      Effect            W            p p<.05
## 3     model 1.709799e-05 7.545758e-70     *
## 4 Exp:model 1.709799e-05 7.545758e-70     *
##
## $`Sphericity Corrections`
##      Effect       GGe        p[GG] p[GG]<.05       HFe        p[HF] p[HF]<.05
## 3     model 0.5000043 2.542975e-13         * 0.5000047 2.542918e-13         *
## 4 Exp:model 0.5000043 4.702570e-01           0.5000047 4.702572e-01
```

```
bf <- anovaBF(m_W_Ds ~ model*Exp + NSub, data = mweight_sub, whichRandom = "NSub")
```

```
## Warning: data coerced from tibble to data frame
```

```
bf_model <- bf[3]/bf[2]

bf_Exp <- bf[3]/bf[1]

#Prior type * ragne interaction
bf_int <- bf[4]/bf[3]
```

```
ezANOVA(data = mweight_sub, dv= m_W_L, wid=NSub, within=.(model), between= Exp)
```

```
## Warning: You have removed one or more levels from variable "model". Refactoring
## for ANOVA.
```

```
## Warning: The column supplied as the wid variable contains non-unique values
## across levels of the supplied between-Ss variables. Automatically fixing this by
## generating unique wid labels.
```

```
## $ANOVA
##      Effect DFn DFd            F            p p<.05          ges
## 2       Exp   1  30   0.85844844 3.615675e-01       2.751123e-02
## 3     model   2  60 440.55919081 1.365243e-36     * 1.431131e-01
## 4 Exp:model   2  60   0.07171287 9.308778e-01       2.718547e-05
##
## $`Mauchly's Test for Sphericity`
##      Effect            W            p p<.05
## 3     model 0.0008808258 5.022328e-45     *
## 4 Exp:model 0.0008808258 5.022328e-45     *
##
## $`Sphericity Corrections`
##      Effect       GGe        p[GG] p[GG]<.05       HFe        p[HF] p[HF]<.05
## 3     model 0.5002203 1.710793e-19         * 0.5002431 1.707717e-19         *
## 4 Exp:model 0.5002203 7.908017e-01           0.5002431 7.908131e-01
```

```
bf <- anovaBF(m_W_L ~ model*Exp + NSub, data = mweight_sub, whichRandom = "NSub")
```

```
## Warning: data coerced from tibble to data frame
```

```
bf_model <- bf[3]/bf[2]

bf_Exp <- bf[3]/bf[1]

#Prior type * ragne interaction
bf_int <- bf[4]/bf[3]
```

```
dat1 = mweight_sub%>%filter(Exp == 'Exp1', model =='LGM')
dat2 = mweight_sub%>%filter(Exp == 'Exp2', model =='LGM')
t.test(dat1$m_W_P_G, dat2$m_W_P_G, paired = TRUE, alternative = "two.sided")
```

```
##
##  Paired t-test
##
## data:  dat1$m_W_P_G and dat2$m_W_P_G
## t = -2.3915, df = 15, p-value = 0.03032
## alternative hypothesis: true difference in means is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.25173543 -0.01447349
## sample estimates:
## mean of the differences
##              -0.1331045
```

```
dat1 = mweight_sub%>%filter(Exp == 'Exp1', model =='DIM')
dat2 = mweight_sub%>%filter(Exp == 'Exp2', model =='DIM')
t.test(dat1$m_W_P_G, dat2$m_W_P_G, paired = TRUE, alternative = "two.sided")
```

```
##
##  Paired t-test
##
## data:  dat1$m_W_P_G and dat2$m_W_P_G
## t = -2.3595, df = 15, p-value = 0.03227
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.27786501 -0.01411134
## sample estimates:
## mean of the differences
##              -0.1459882
```

```
dat1 = mweight_sub%>%filter(Exp == 'Exp1', model =='PIM')
dat2 = mweight_sub%>%filter(Exp == 'Exp2', model =='PIM')
t.test(dat1$m_W_P_G, dat2$m_W_P_G, paired = TRUE, alternative = "two.sided")
```

```
##
##  Paired t-test
##
## data:  dat1$m_W_P_G and dat2$m_W_P_G
## t = -2.3913, df = 15, p-value = 0.03034
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.25206636 -0.01448223
## sample estimates:
## mean of the differences
##              -0.1332743
```

```
dat1 = mweight_sub%>%filter(Exp == 'Exp1', model =='LGM')
dat2 = mweight_sub%>%filter(Exp == 'Exp2', model =='LGM')
t.test(dat1$m_W_Ds, dat2$m_W_Ds, paired = TRUE, alternative = "two.sided")
```

```
##
##  Paired t-test
##
## data:  dat1$m_W_Ds and dat2$m_W_Ds
## t = 2.2858, df = 15, p-value = 0.03723
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01185941 0.33950107
## sample estimates:
## mean of the differences
##               0.1756802
```

```
dat1 = mweight_sub%>%filter(Exp == 'Exp1', model =='DIM')
dat2 = mweight_sub%>%filter(Exp == 'Exp2', model =='DIM')
```

```
t.test(dat1$m_W_Ds, dat2$m_W_Ds, paired = TRUE, alternative = "two.sided")
```

```
##
##  Paired t-test
##
## data:  dat1$m_W_Ds and dat2$m_W_Ds
## t = 2.3506, df = 15, p-value = 0.03284
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01731339 0.35416319
## sample estimates:
## mean of the differences
##               0.1857383
```

```
dat1 = mweight_sub%>%filter(Exp == 'Exp1', model =='PIM')
dat2 = mweight_sub%>%filter(Exp == 'Exp2', model =='PIM')
t.test(dat1$m_W_Ds, dat2$m_W_Ds, paired = TRUE, alternative = "two.sided")
```

```
##
##  Paired t-test
##
## data:  dat1$m_W_Ds and dat2$m_W_Ds
## t = 2.2853, df = 15, p-value = 0.03727
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01182421 0.33948699
## sample estimates:
## mean of the differences
##               0.1756556
```

```
dat1 = mweight_sub%>%filter(Exp == 'Exp1', model =='LGM')
dat2 = mweight_sub%>%filter(Exp == 'Exp2', model =='LGM')
t.test(dat1$m_W_L, dat2$m_W_L, paired = TRUE, alternative = "two.sided")
```

```
##
##  Paired t-test
##
## data:  dat1$m_W_L and dat2$m_W_L
## t = -0.95796, df = 15, p-value = 0.3533
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.13730602  0.05215445
## sample estimates:
## mean of the differences
##             -0.04257578
```

```
dat1 = mweight_sub%>%filter(Exp == 'Exp1', model =='DIM')
dat2 = mweight_sub%>%filter(Exp == 'Exp2', model =='DIM')
t.test(dat1$m_W_L, dat2$m_W_L, paired = TRUE, alternative = "two.sided")
```

```
##
##  Paired t-test
##
## data:  dat1$m_W_L and dat2$m_W_L
## t = -0.94333, df = 15, p-value = 0.3605
## alternative hypothesis: true difference in means is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.12956494  0.05006471
## sample estimates:
## mean of the differences
##             -0.03975012
```

```
dat1 = mweight_sub%>%filter(Exp == 'Exp1', model =='PIM')
dat2 = mweight_sub%>%filter(Exp == 'Exp2', model =='PIM')
t.test(dat1$m_W_L, dat2$m_W_L, paired = TRUE, alternative = "two.sided")
```

```
##
##  Paired t-test
##
## data:  dat1$m_W_L and dat2$m_W_L
## t = -0.95402, df = 15, p-value = 0.3552
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.13706894  0.05230634
## sample estimates:
## mean of the differences
##              -0.0423813
```

## 3.2 goodness of model

```
predY_err_new%>%filter(ver == 'log', part == 'all', model %in% c('LGM', 'DIM', 'PIM')) %>%select("Exp",
```

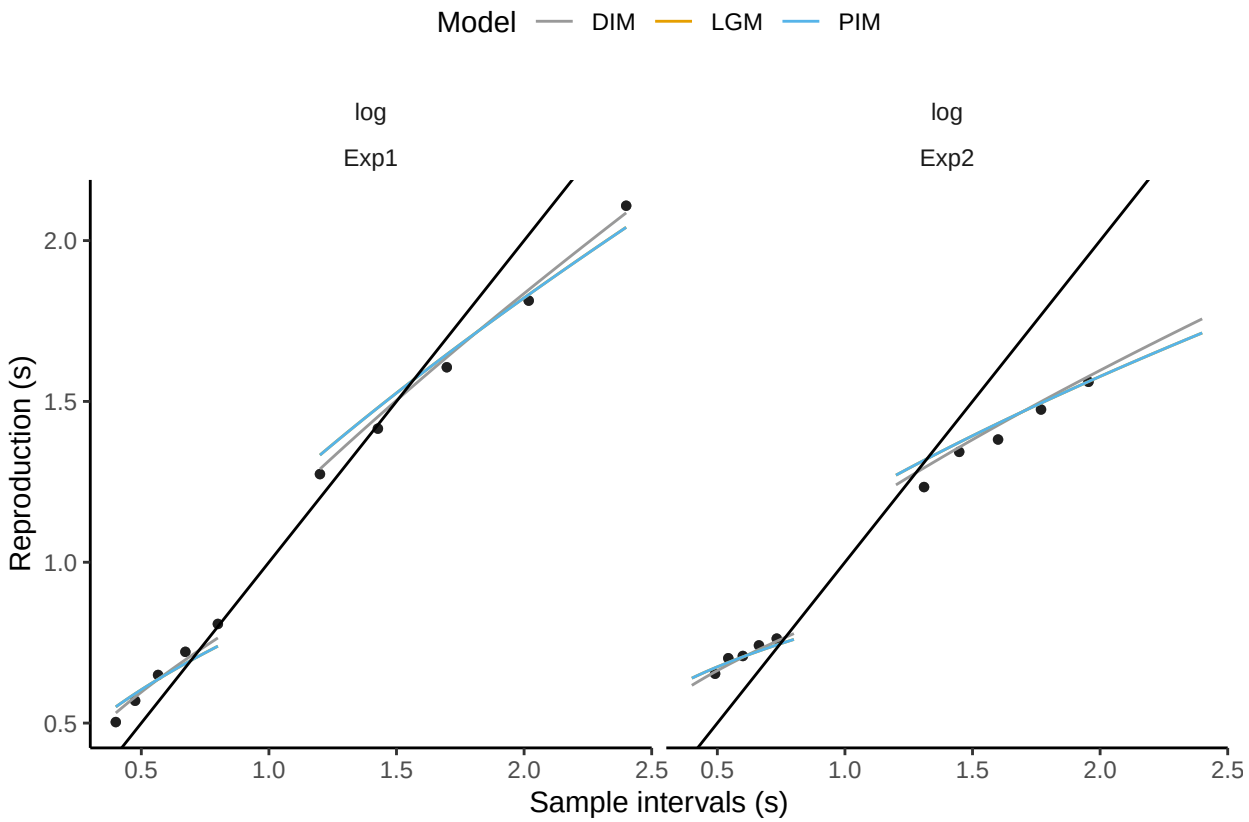
```
## Adding missing grouping variables: `ver`
```

```
## # A tibble: 6 x 5
## # Groups:   Exp, model, ver [6]
##   ver   Exp   model acc_err acc_var
##   <chr> <chr> <fct>   <dbl>   <dbl>
## 1 log   Exp1  DIM      98.1    51.9
## 2 log   Exp1  LGM      97.4    56.3
## 3 log   Exp1  PIM      97.4    56.3
## 4 log   Exp2  DIM      97.8    68.5
## 5 log   Exp2  LGM      97.5    72.7
## 6 log   Exp2  PIM      97.5    72.7
```

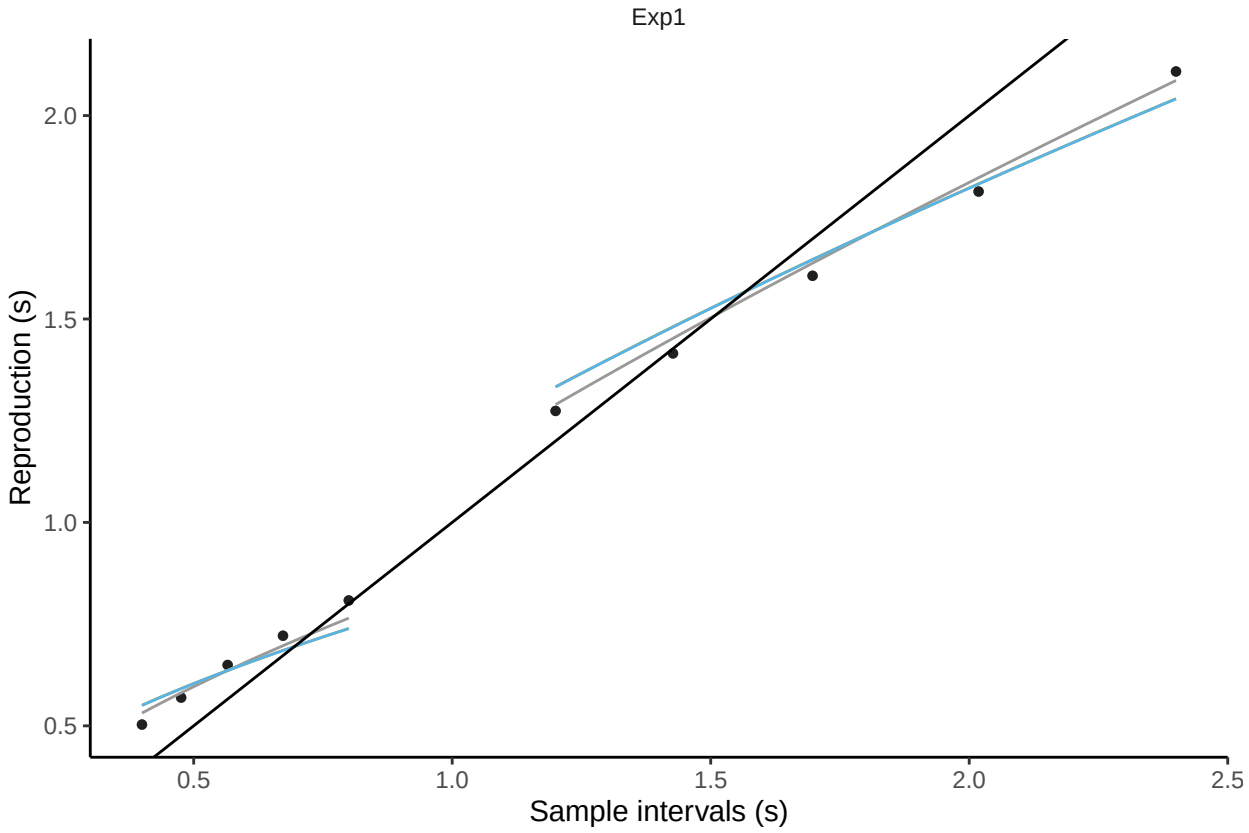
## 3.3 prediction of RP

### 3.3.1 RP

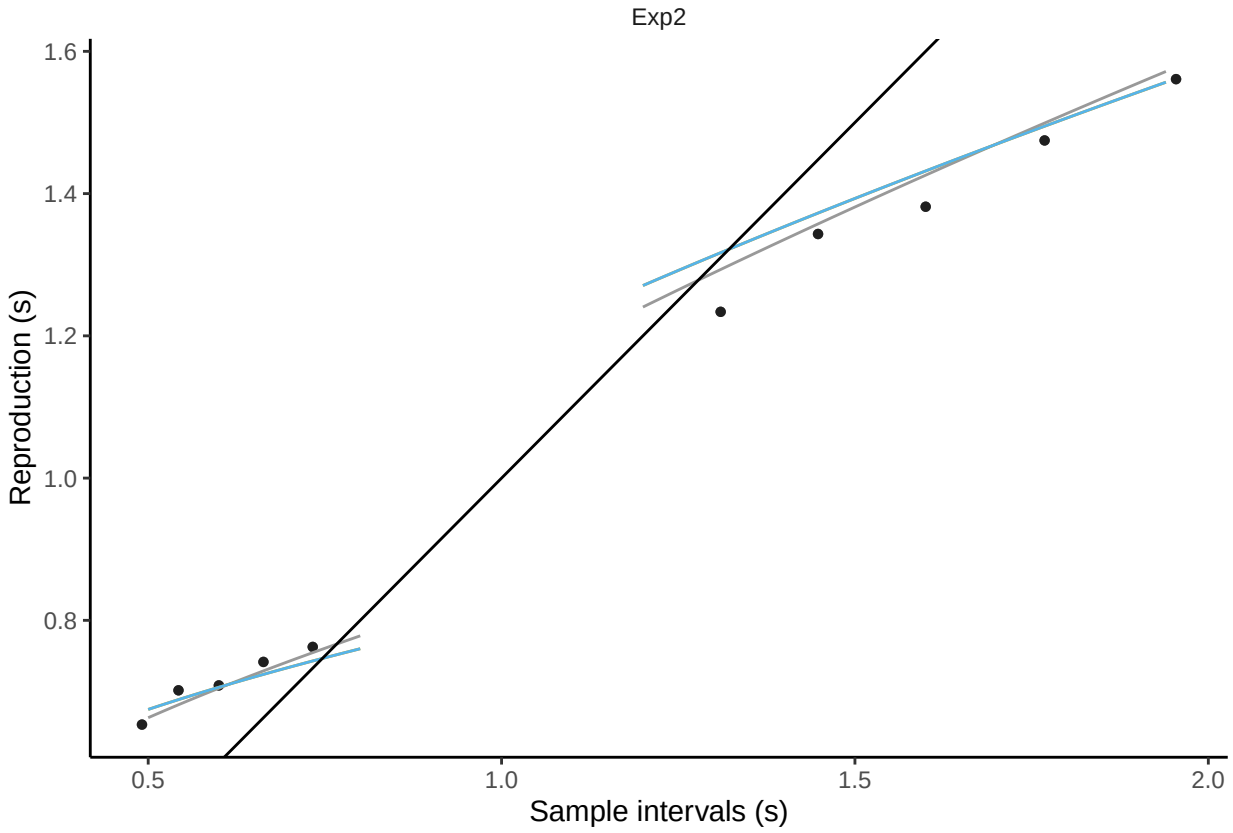
```
RP_log  <- ggplot(data = m_predY%>%filter(part == 'all', ver == 'log', model %in% c('LGM', 'DIM', 'PIM'
  geom_point(size=1.5, alpha = 0.5)+
  geom_line(data= m_newY%>%filter(part == 'all', ver == 'log', model %in% c('LGM', 'DIM', 'PIM')), aes(
  geom_abline(slope=1, intercept=0)+
  facet_wrap(ver~Exp, nrow = 1) +
  labs(x="Sample intervals (s)", y="Reproduction (s)", color = "Model")+
  theme_new+colorSet5+guides(shape="none")

ggsave(file.path(figure_path,'figures/pred_RP_log.png'), RP_log, width = 9, height = 5)

RP_log
```



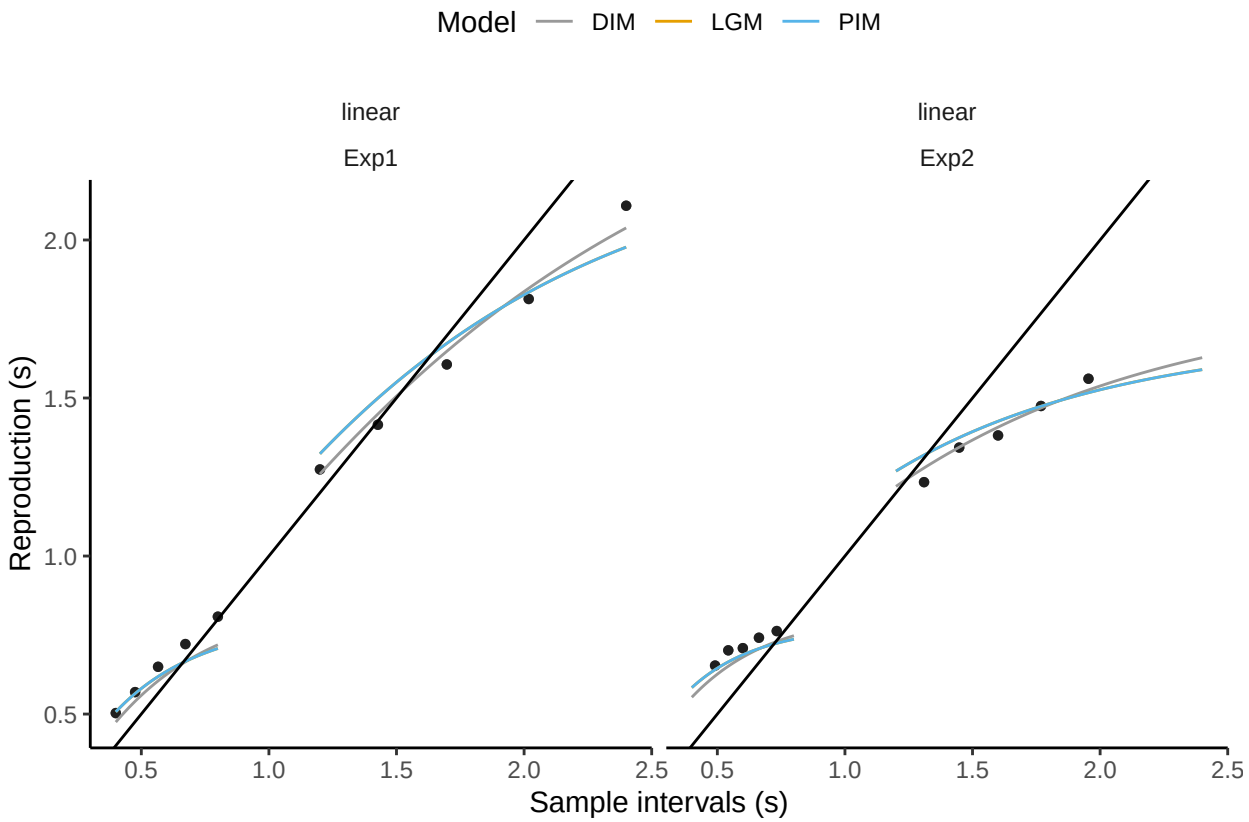

```
RP_log_Exp1  <- ggplot(data = m_predY%>%filter(part == 'all', ver == 'log', Exp == 'Exp1', model %in% c
  geom_point(size=1.5, alpha = 0.5)+
  geom_line(data= m_newY%>%filter(part == 'all', ver == 'log', Exp == 'Exp1', model %in% c('LGM', 'DIM'
  geom_abline(slope=1, intercept=0)+
  facet_wrap(~Exp, nrow = 1) +
  labs(x="Sample intervals (s)", y="Reproduction (s)", color = "Model")+
  theme_new+colorSet5+guides(shape="none")+ theme(legend.position = "none")

RP_log_Exp1
```



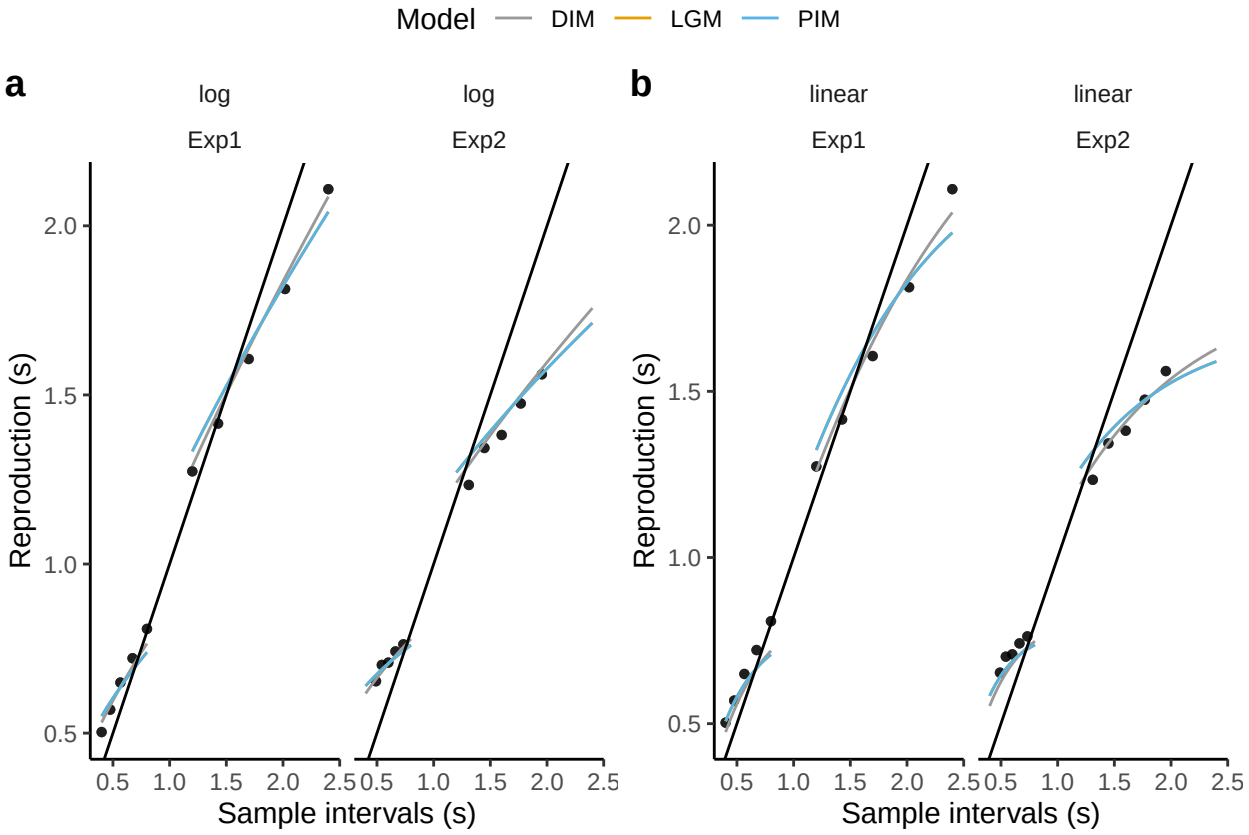

```
RP_log_Exp2  <- ggplot(data = m_predY%>%filter(part == 'all', ver == 'log', Exp == 'Exp2', model %in% c
  geom_point(size=1.5, alpha = 0.5)+
  geom_line(data= m_newY%>%filter(part == 'all', ver == 'log', targetDur < 1.95, targetDur > 0.49, Exp =
  geom_abline(slope=1, intercept=0)+
  facet_wrap(~Exp, nrow = 1) +
  labs(x="Sample intervals (s)", y="Reproduction (s)", color = "Model")+
  theme_new+colorSet5+guides(shape="none")+ theme(legend.position = "none")

RP_log_Exp2
```




```
RP_linear  <- ggplot(data = m_predY%>%filter(part == 'all', ver == 'linear', model %in% c('LGM', 'DIM',
  geom_point(size=1.5, alpha = 0.5)+
  geom_line(data= m_newY%>%filter(part == 'all', ver == 'linear', model %in% c('LGM', 'DIM', 'PIM')), a
  geom_abline(slope=1, intercept=0)+
  facet_wrap(ver~Exp, nrow = 1) +
  labs(x="Sample intervals (s)", y="Reproduction (s)", shape=" ", color = "Model")+
  theme_new+colorSet5+guides(shape="none")

ggsave(file.path(figure_path,'figures/pred_RP_linear.png'), RP_linear, width = 9, height = 5)

RP_linear
```

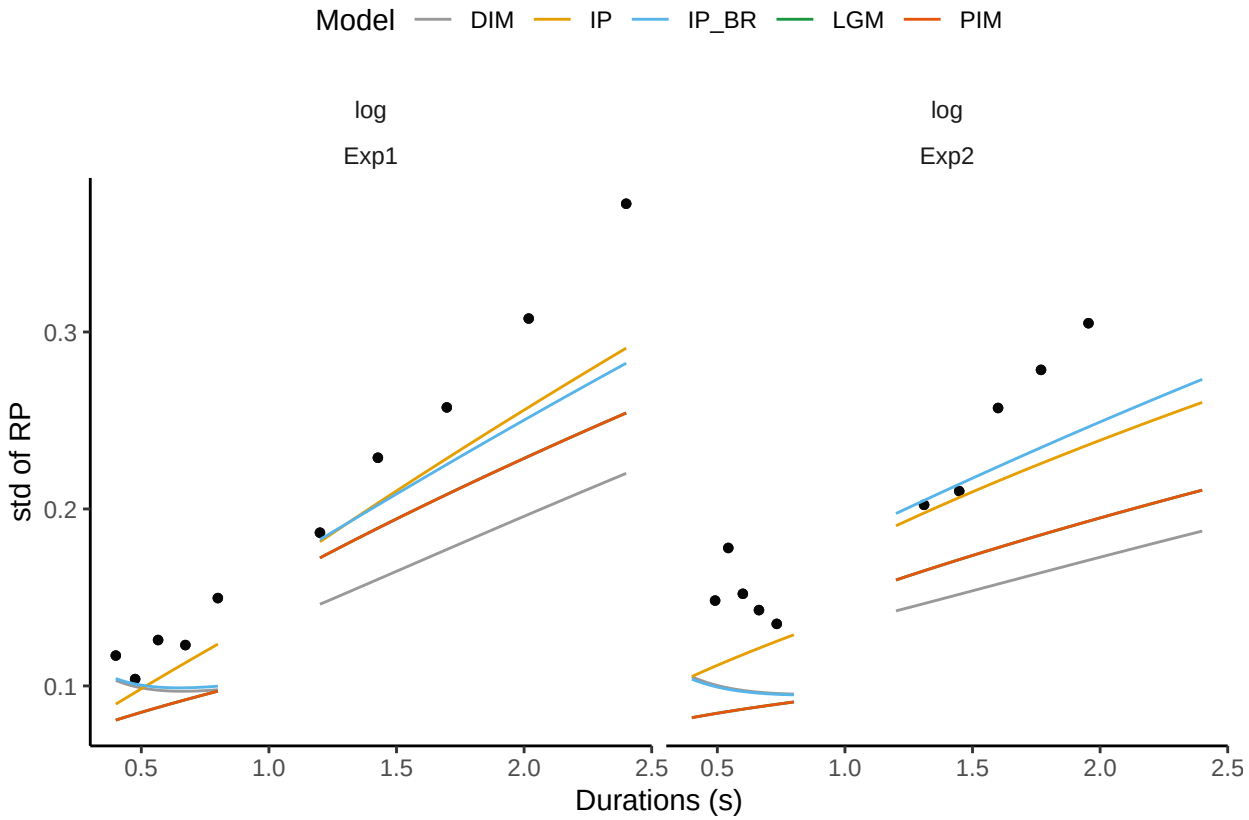



```
fig_RP <- ggarrange(RP_log, RP_linear, common.legend = TRUE, nrow = 1,ncol = 2, labels = c('a', 'b'))
ggsave(file.path(figure_path,'pred_RP.png'), fig_RP, width = 9, height = 5)
fig_RP
```



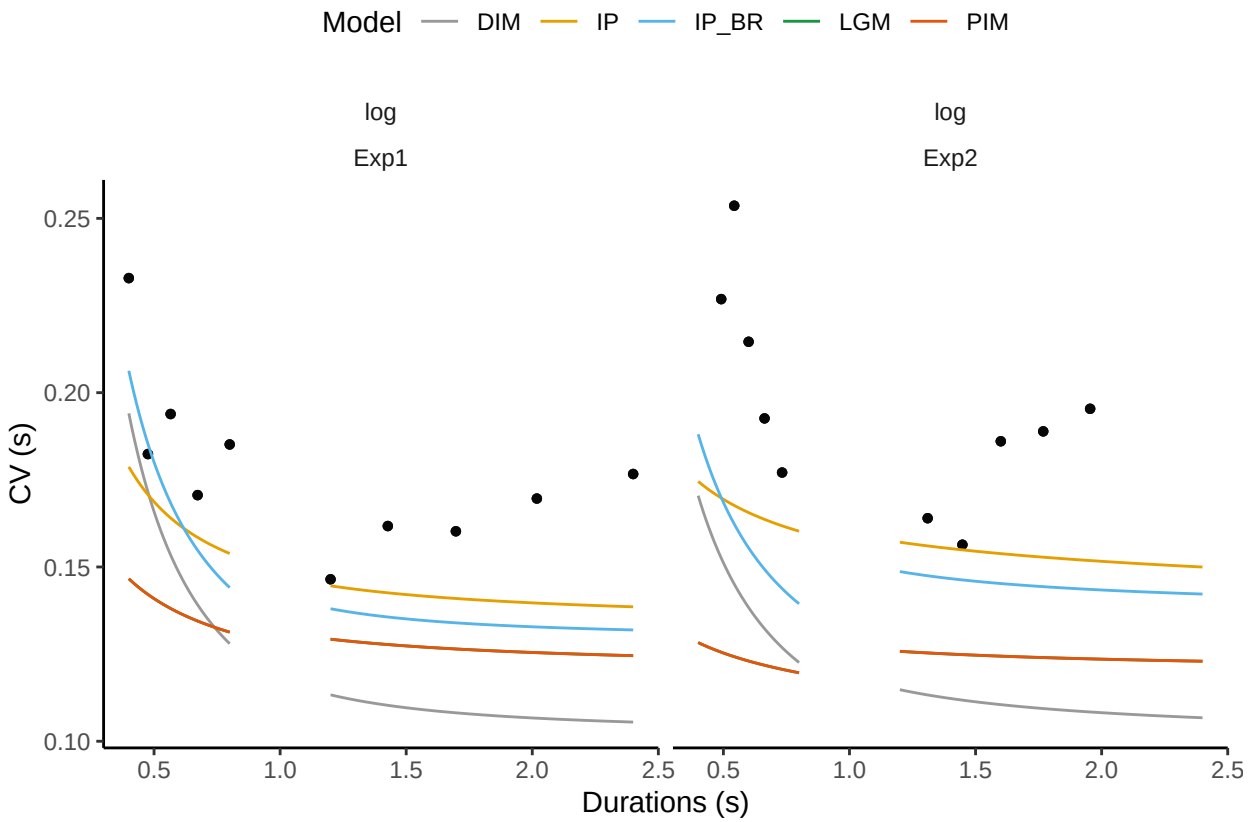

### predicted SD of RP

```
stdRP_log  <- ggplot(data = m_predY%>%filter(part == 'all', ver == 'log'), aes(x = targetDur, y = sd_RP
  geom_point(size=1.5, alpha = 0.5)+
  geom_line(data= m_newY%>%filter(part == 'all', ver == 'log'), aes(x=targetDur, y=m_sig_r,  group =int
  facet_wrap(ver~Exp, nrow = 1) +
  labs(x="Durations (s)", y="std of RP", color = "Model")+
  theme_new+colorSet5+guides(shape="none")

stdRP_log
```

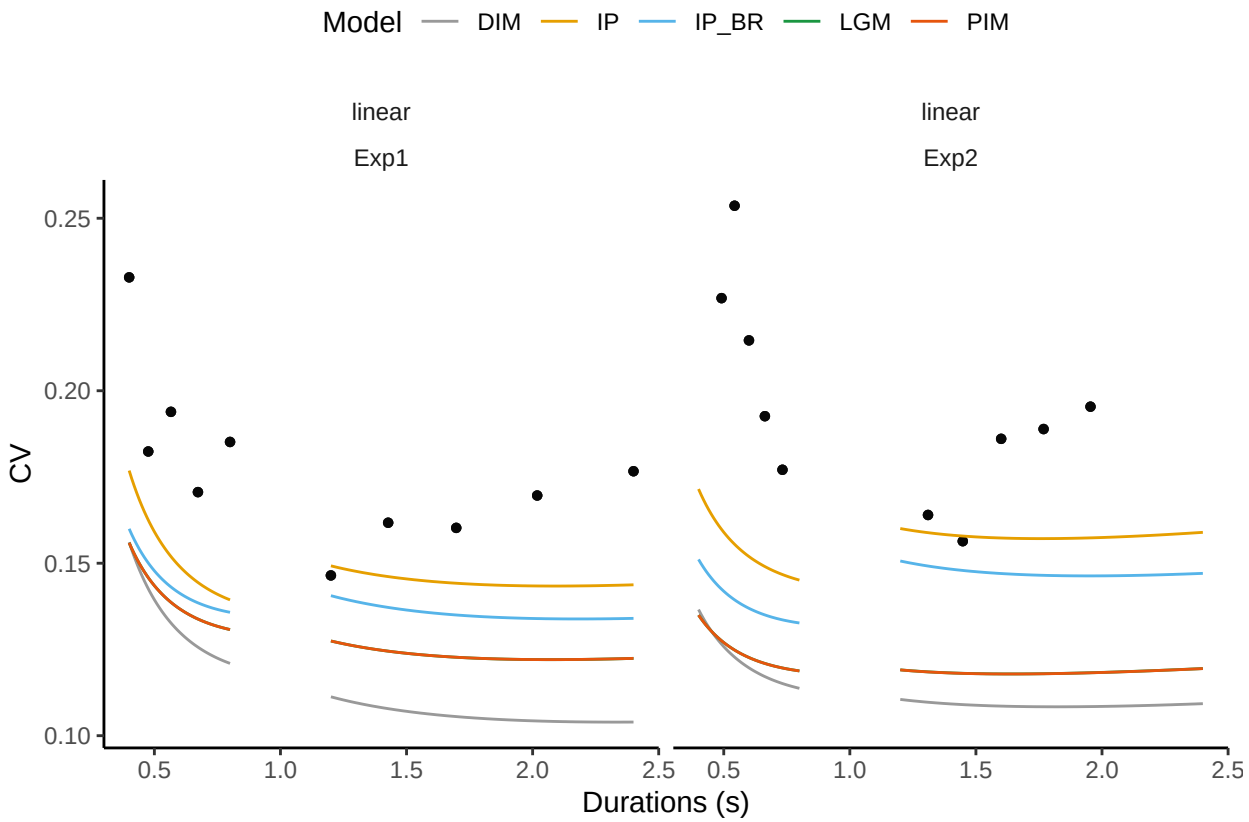



### 3.3.2 predicted CV

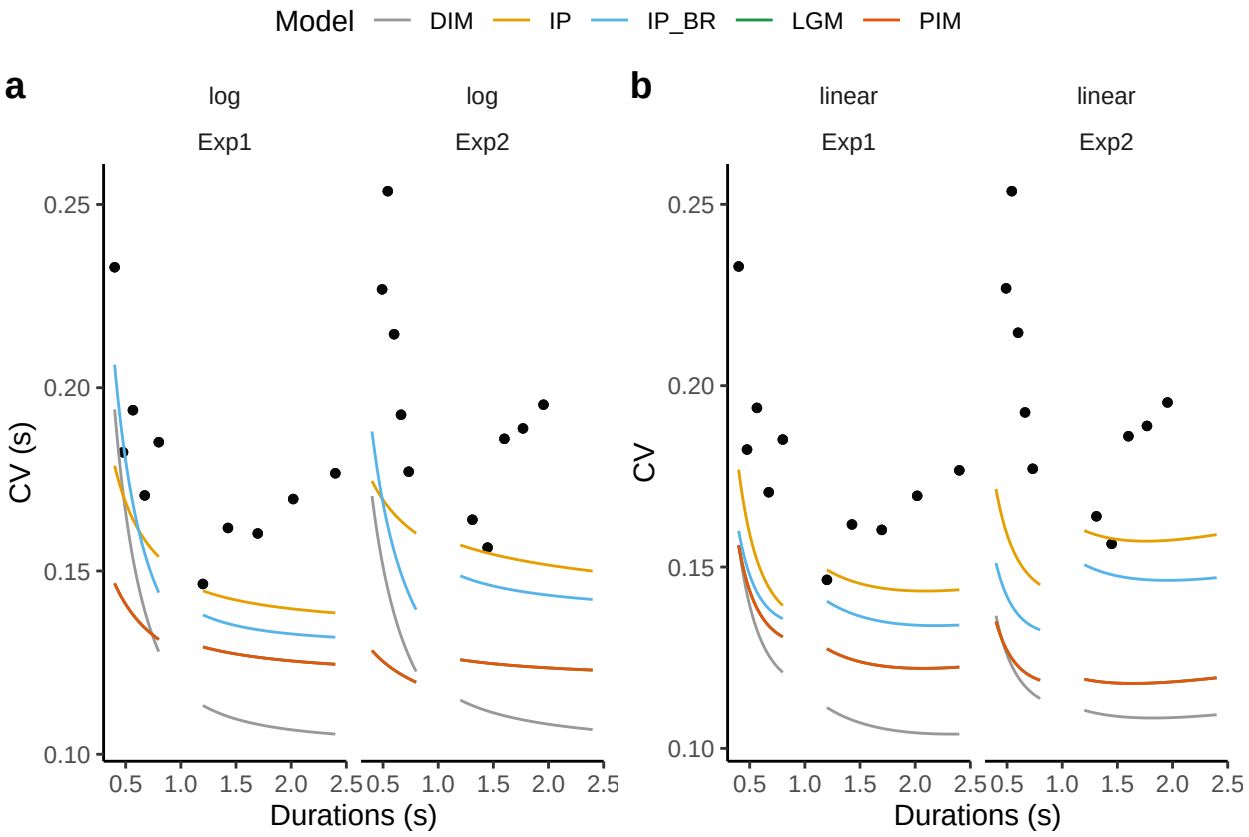
```
CV_log  <- ggplot(data = m_predY%>%filter(part == 'all', ver == 'log'), aes(x = targetDur, y = cv,  grou
  geom_point(size=1.5, alpha = 0.5)+
  geom_line(data= m_newY%>%filter(part == 'all', ver == 'log'), aes(x=targetDur, y=m_sig_r/m_mu_r,  grou
  geom_abline(slope=1, intercept=0)+
  facet_wrap(ver~Exp, nrow = 1) +
  labs(x="Durations (s)", y="CV (s)", color = "Model")+
  theme_new+colorSet5+guides(shape="none")

ggsave(file.path(figure_path,'figures/pred_CV_log.png'), CV_log, width = 9, height = 5)

CV_log
```




```
CV_linear  <- ggplot(data = m_predY%>%filter(part == 'all', ver == 'linear'), aes(x = targetDur, y = cv
  geom_point(size=1.5, alpha = 0.5)+
  geom_line(data= m_newY%>%filter(part == 'all', ver == 'linear'), aes(x=targetDur, y=m_sig_r/m_mu_r,  g
  geom_abline(slope=1, intercept=0)+
  facet_wrap(ver~Exp, nrow = 1) +
  labs(x="Durations (s)", y="CV", shape=" ", color = "Model")+
  theme_new+colorSet5+guides(shape="none")

ggsave(file.path(figure_path,'figures/pred_CV_linear.png'), CV_linear, width = 9, height = 5)

CV_linear
```

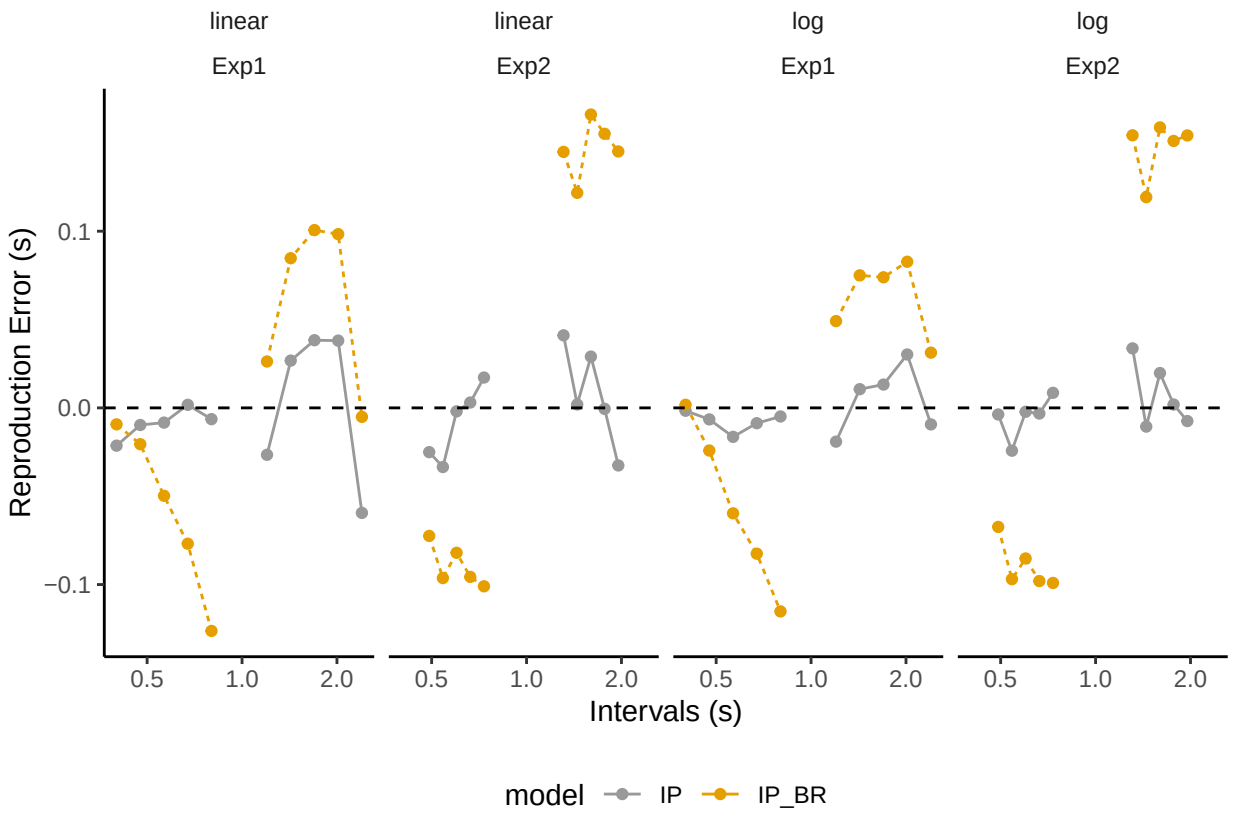



```
fig_CV <- ggarrange(CV_log, CV_linear, common.legend = TRUE, nrow = 1,ncol = 2, labels = c('a', 'b'))
ggsave(file.path(figure_path,'pred_CV.png'), fig_CV, width = 9, height = 5)
fig_CV
```




### 3.3.3 plot observed and predicted Reproduction Error

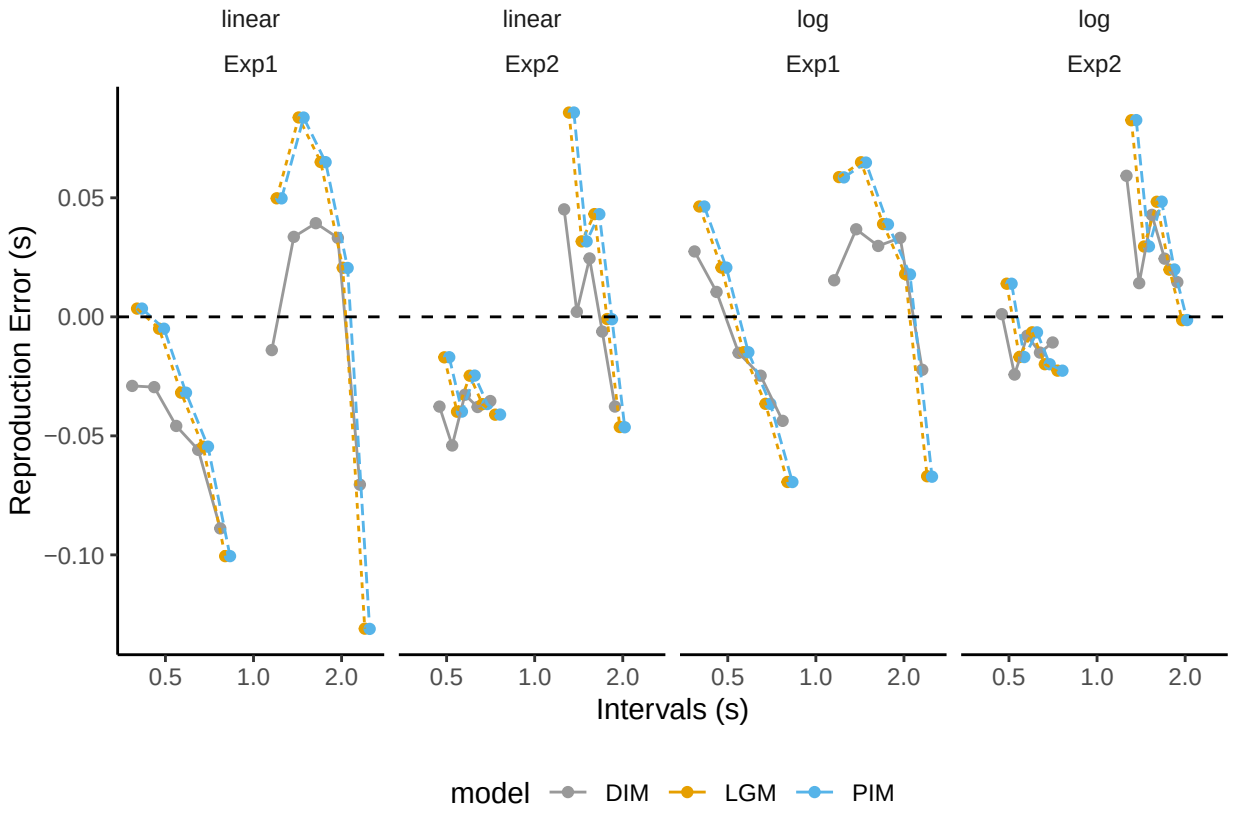
```
fig_mrepError_model_IP <- ggplot(m_predY%>% filter(part =='all', model %in%c("IP", "IP_BR")),
                                 aes(targetDur,
                                     m_mu_r-m_RP,
                                     group = interaction(range, model), color = model)) +
  geom_point()+
  geom_line(size = .5, aes(color = model, linetype = model))+
  geom_hline(yintercept= 0, linetype = 2) +
  theme_minimal()+ theme_new +
  facet_wrap(ver~Exp, nrow = 1 )+
  scale_linetype(guide = "none")+
  theme(strip.background = element_blank()) +
  labs(x = 'Intervals (s)', y = 'Reproduction Error (s)',  color='model')+ theme(legend.position='botto
  scale_x_continuous(trans='log10') +colorSet5

fig_mrepError_model_IP
```




```
ggsave(file.path(figure_path,'fig_mrepError_model_IP.png'), fig_mrepError_model_IP, width = 7, height =
```

```
fig_mrepError_model_all <- ggplot(m_predY%>% filter(part =='all', model %in%c("DIM","LGM", "PIM")),
                                  aes(targetDur,
                                      m_mu_r-m_RP, m_mu_r-m_RP,
                                      group = interaction(range, model), color = model)) +
  geom_point(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 +
  facet_wrap(ver~Exp, nrow = 1 )+
  scale_linetype(guide = "none")+
  theme(strip.background = element_blank()) +
  labs(x = 'Intervals (s)', y = 'Reproduction Error (s)',  color='model')+ theme(legend.position='botto
  scale_x_continuous(trans='log10') +colorSet5

fig_mrepError_model_all
```

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




```
ggsave(file.path(figure_path,'fig_mrepError_model_all.png'), fig_mrepError_model_all, width = 7, height
```

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

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

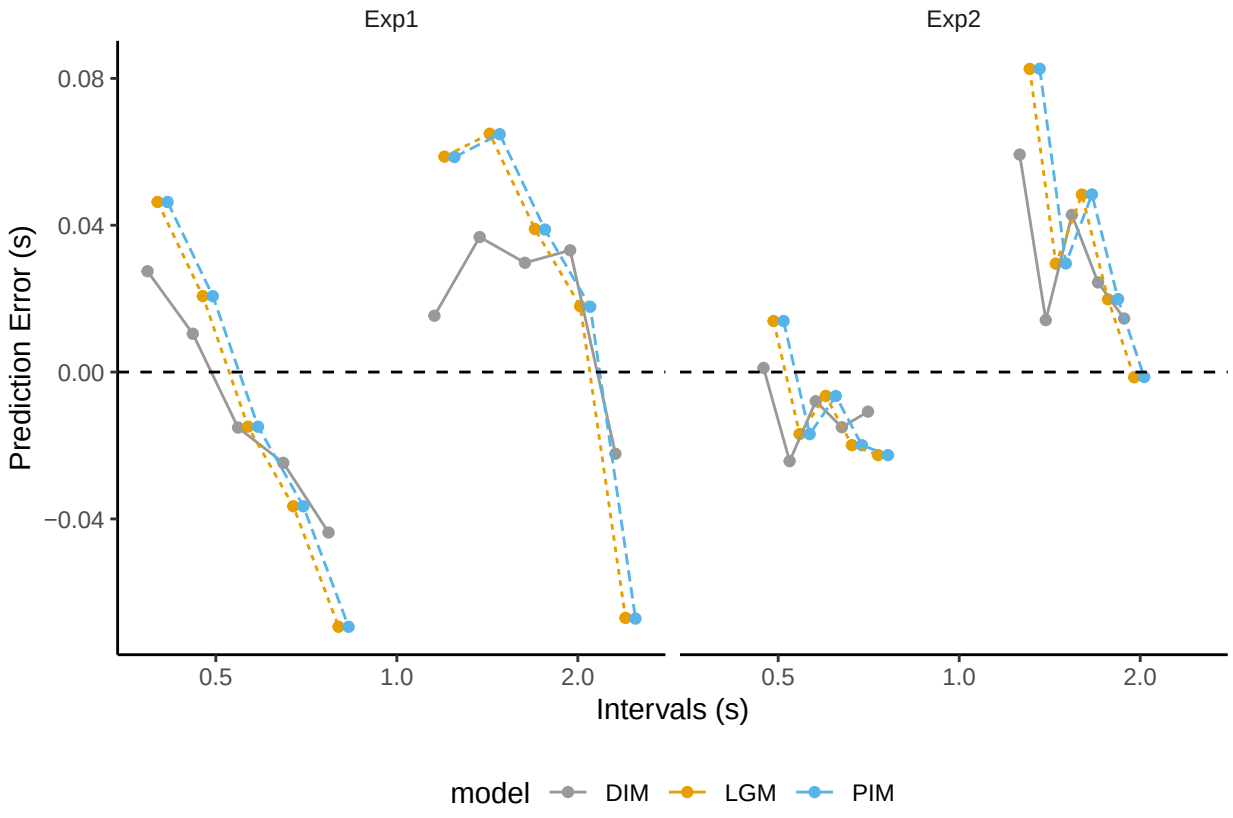
```
## Warning: position_dodge requires non-overlapping x intervals
```

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

```
fig_mrepError_model <- ggplot(m_predY%>% filter(part =='all', ver == 'log', model %in%c("DIM","LGM", "PI
                              aes(targetDur,
                                  m_mu_r-m_RP, m_mu_r-m_RP,
                                  group = interaction(range, model), color = model)) +
  geom_point(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 +
  facet_wrap(~Exp)+
  scale_linetype(guide = "none")+
  theme(strip.background = element_blank()) +
  labs(x = 'Intervals (s)', y = 'Prediction Error (s)',  color='model')+ theme(legend.position='bottom')
  scale_x_continuous(trans='log10') +colorSet5

fig_mrepError_model
```

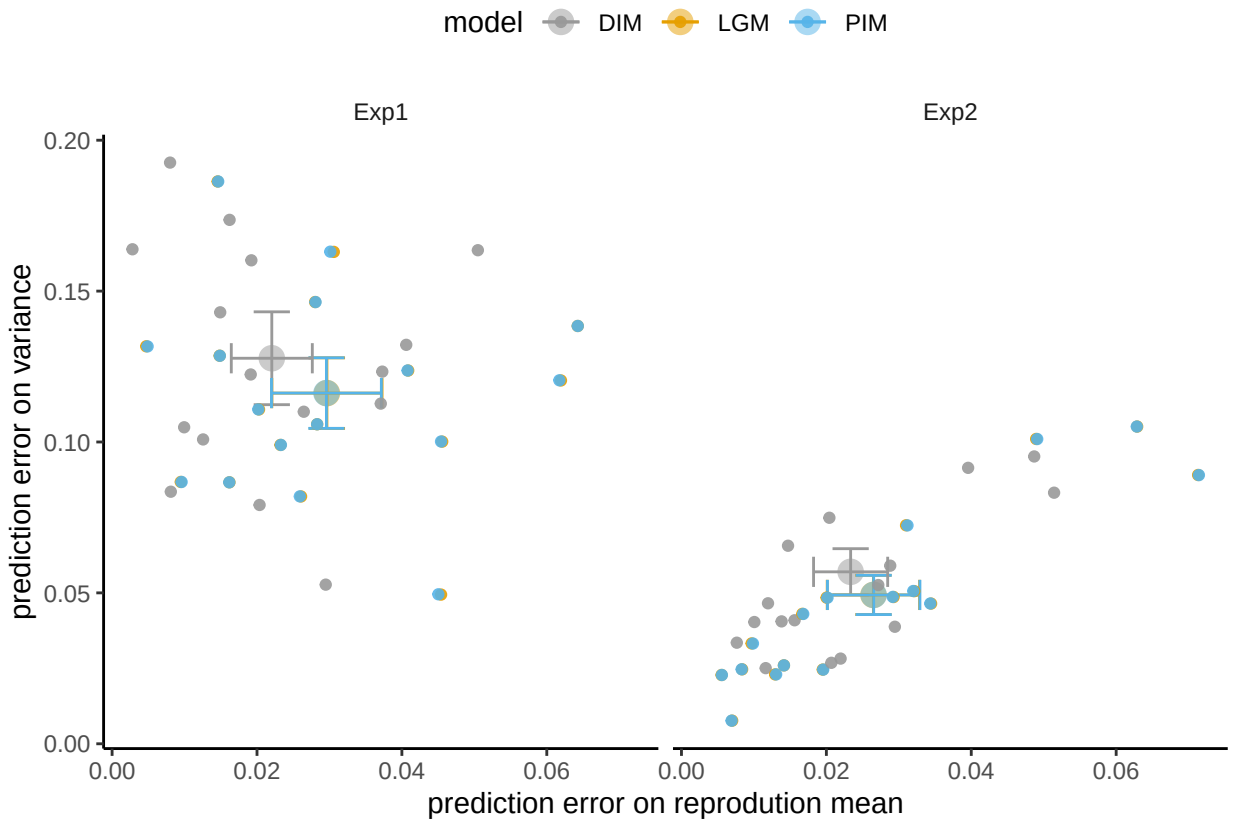
```
## Warning: position_dodge requires non-overlapping x intervals
```

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



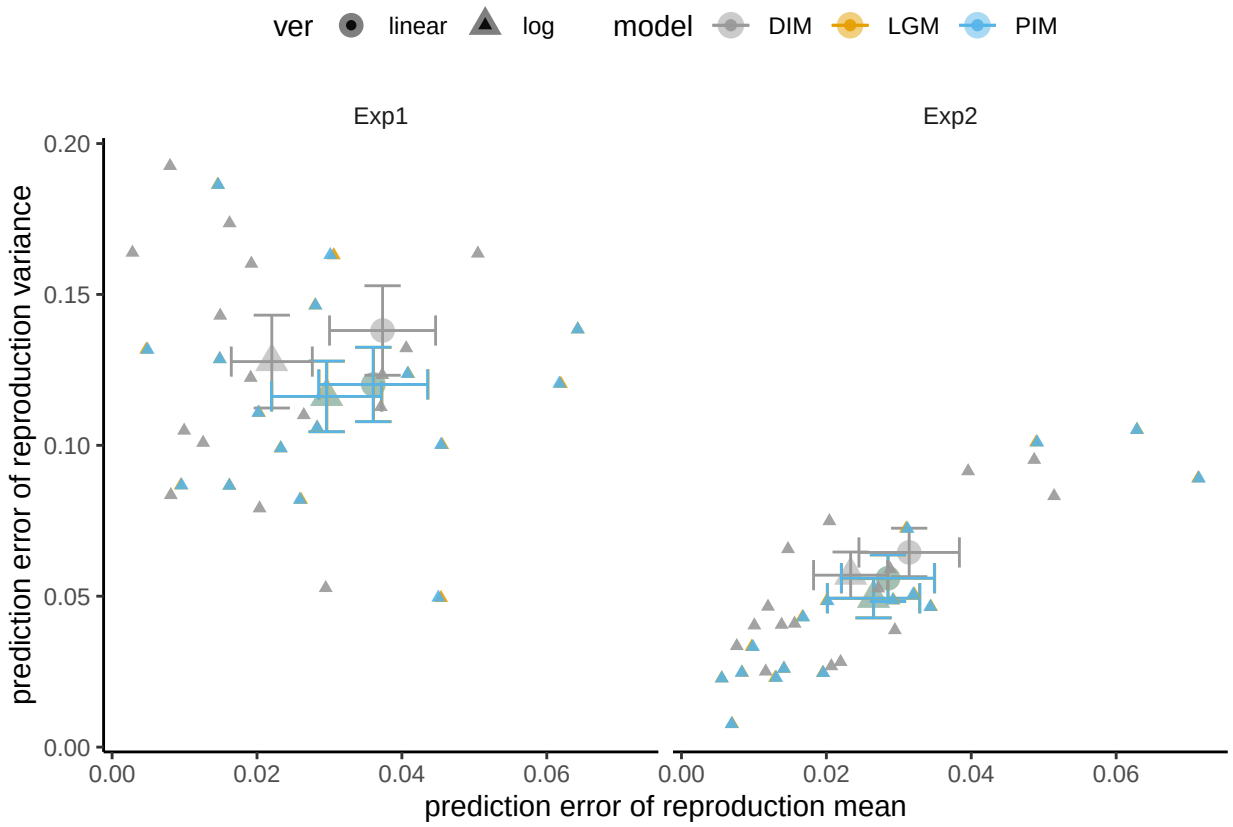

```
#LGM and PIM are overlopped
plt_rErrorScatter = ggplot(data = predY_err_new%>%filter(part =="all", ver=='log', model %in%c("DIM","L
  geom_point(alpha = .5, size = 4, position = position_dodge(width = 0.001))+
  geom_errorbar(aes(ymin = mm_pred_Var_NBIAS-se_pred_Var, ymax = mm_pred_Var_NBIAS+se_pred_Var), width =
  geom_errorbarh(aes(xmin= mm_pred_NBIAS-se_pred_RP_BIAS, xmax = mm_pred_NBIAS+se_pred_RP_BIAS), height
  geom_point(m_predY_sub_new%>%filter(part =="all", ver=='log', model %in%c("DIM","LGM", "PIM")), mappi
  facet_wrap(~Exp) +colorSet3+
  xlab('prediction error on reprodution mean')+
  ylab('prediction error on variance')+
  theme_new+ theme(legend.position = 'top')+guides(size="none")+guides(alpha="none")
plt_rErrorScatter
```

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

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

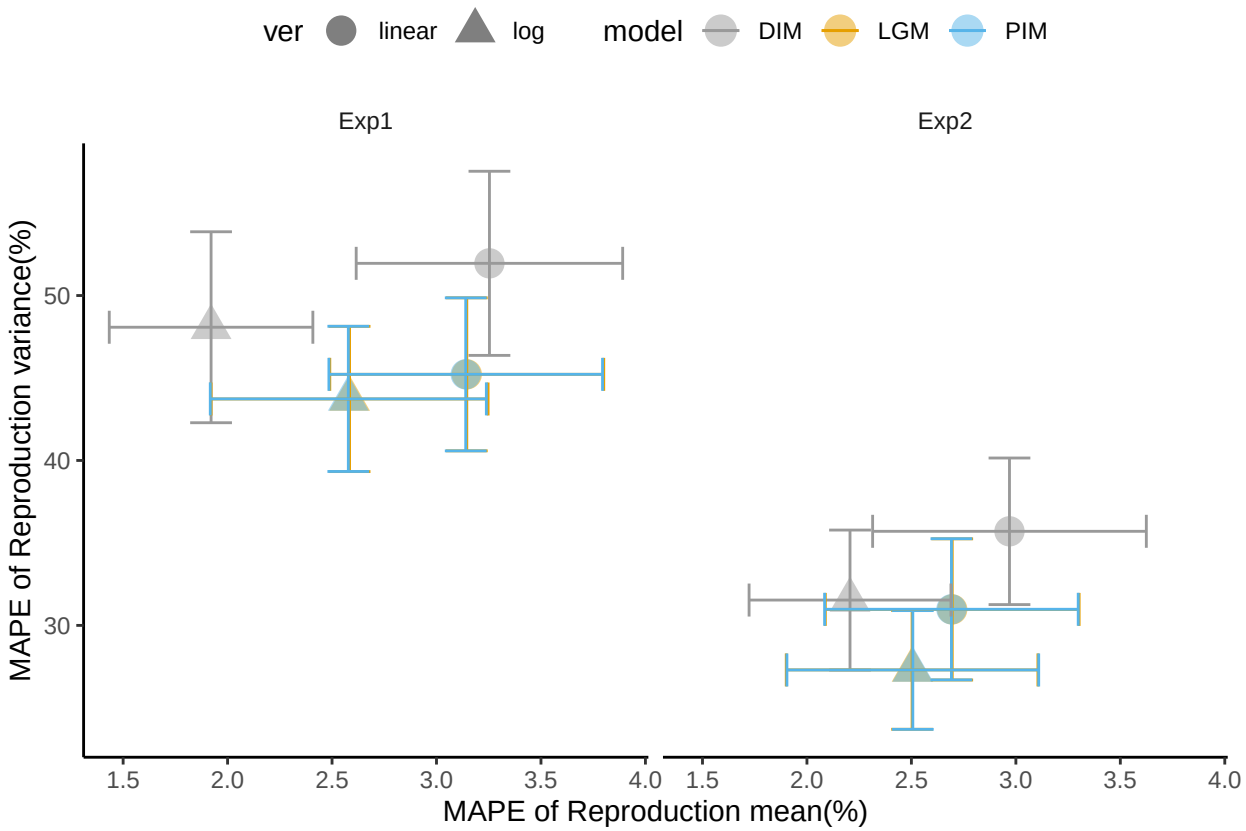



```
plt_rErrorScatter_all = ggplot(data = predY_err_new%>%filter(part =="all", model %in%c("DIM","LGM", "PI
  geom_point(alpha = .5, size = 4, aes(shape = ver), position = position_dodge(width = 0.001))+
  geom_errorbar(aes(ymin = mm_pred_Var_NBIAS-se_pred_Var, ymax = mm_pred_Var_NBIAS+se_pred_Var), width =
  geom_errorbarh(aes(xmin= mm_pred_NBIAS-se_pred_RP_BIAS, xmax = mm_pred_NBIAS+se_pred_RP_BIAS), height
  geom_point(m_predY_sub_new%>%filter(part =="all", ver=='log', model %in%c("DIM","LGM", "PIM")), mappi
  facet_wrap(~Exp) +colorSet3+
  xlab('prediction error of reproduction mean')+
  ylab('prediction error of reproduction variance')+
  theme_new+ theme(legend.position = 'top')+guides(size="none")+guides(alpha="none")
plt_rErrorScatter_all
```

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

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

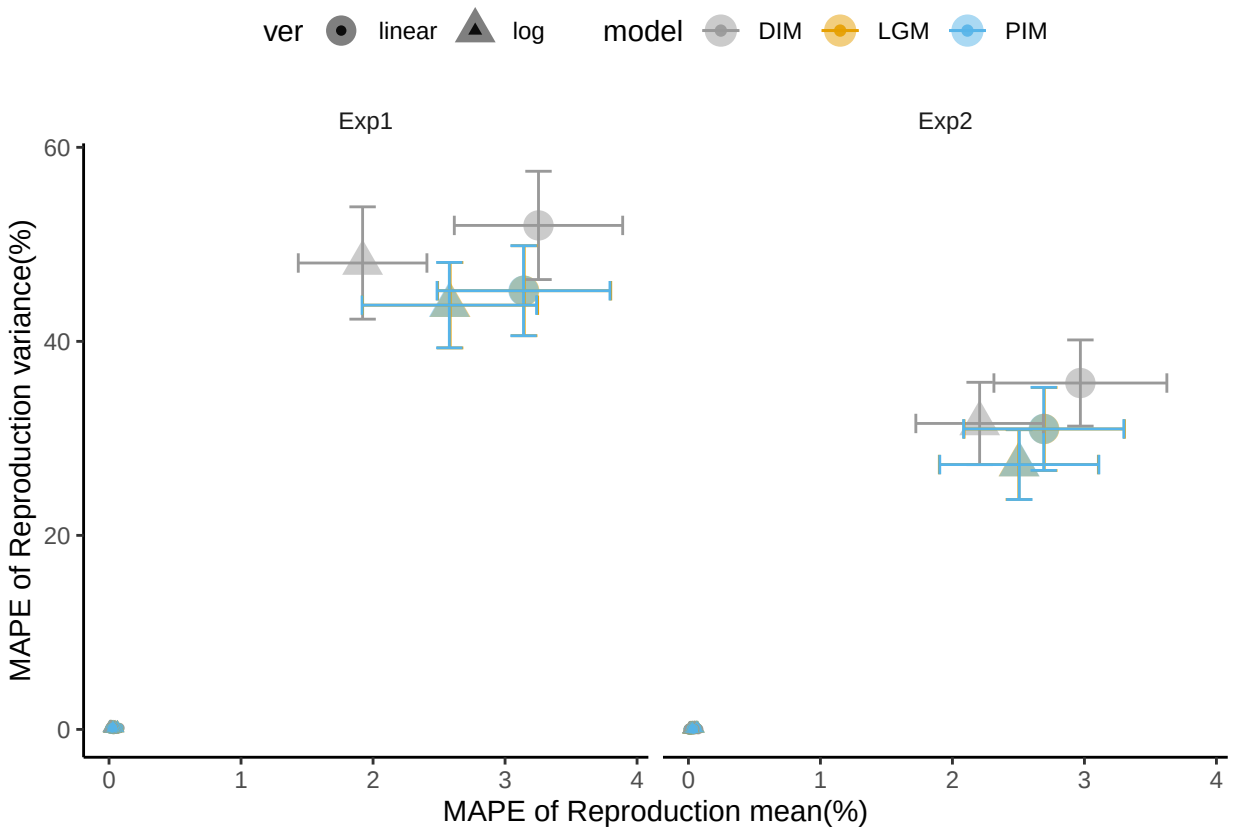



```
plt_rErrorScatter_all1 = ggplot(data = predY_err_new%>%filter(part =="all", model %in%c("DIM","LGM", "P
  geom_point(alpha = .5, size = 5, aes(shape = ver))+
  geom_errorbar(aes(ymin = 100*(mm_pred_Var_NBIAS-se_pred_Var)/mm_sd_RP, ymax = 100*(mm_pred_Var_NBIAS+s
  geom_errorbarh(aes(xmin= 100*(mm_pred_NBIAS-se_pred_RP_BIAS)/mm_m_RP, xmax = 100*(mm_pred_NBIAS+se_pr
  facet_wrap(~Exp) +colorSet5+
  xlab('MAPE of Reproduction mean(%)')+
  ylab('MAPE of Reproduction variance(%)')+
  theme_new+ theme(legend.position = 'top')+guides(size="none")+guides(alpha="none")
plt_rErrorScatter_all1
```




```
plt_rErrorScatter3 = ggplot(data = predY_err_new%>%filter(part =="all", model %in%c("DIM","LGM","PIM"))
  geom_point(alpha = .5, size = 5, aes(shape = ver),  position = position_dodge(width = 0.2))+
  geom_errorbar(aes(ymin = 100*(mm_pred_Var_NBIAS-se_pred_Var)/mm_sd_RP, ymax = 100*(mm_pred_Var_NBIAS+s
  geom_errorbarh(aes(xmin= 100*(mm_pred_NBIAS-se_pred_RP_BIAS)/mm_m_RP, xmax = 100*(mm_pred_NBIAS+se_pr
  geom_point(m_predY_sub_new%>%filter(part =="all", model %in%c("DIM","LGM", "PIM")), mapping = aes(abs
  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
```

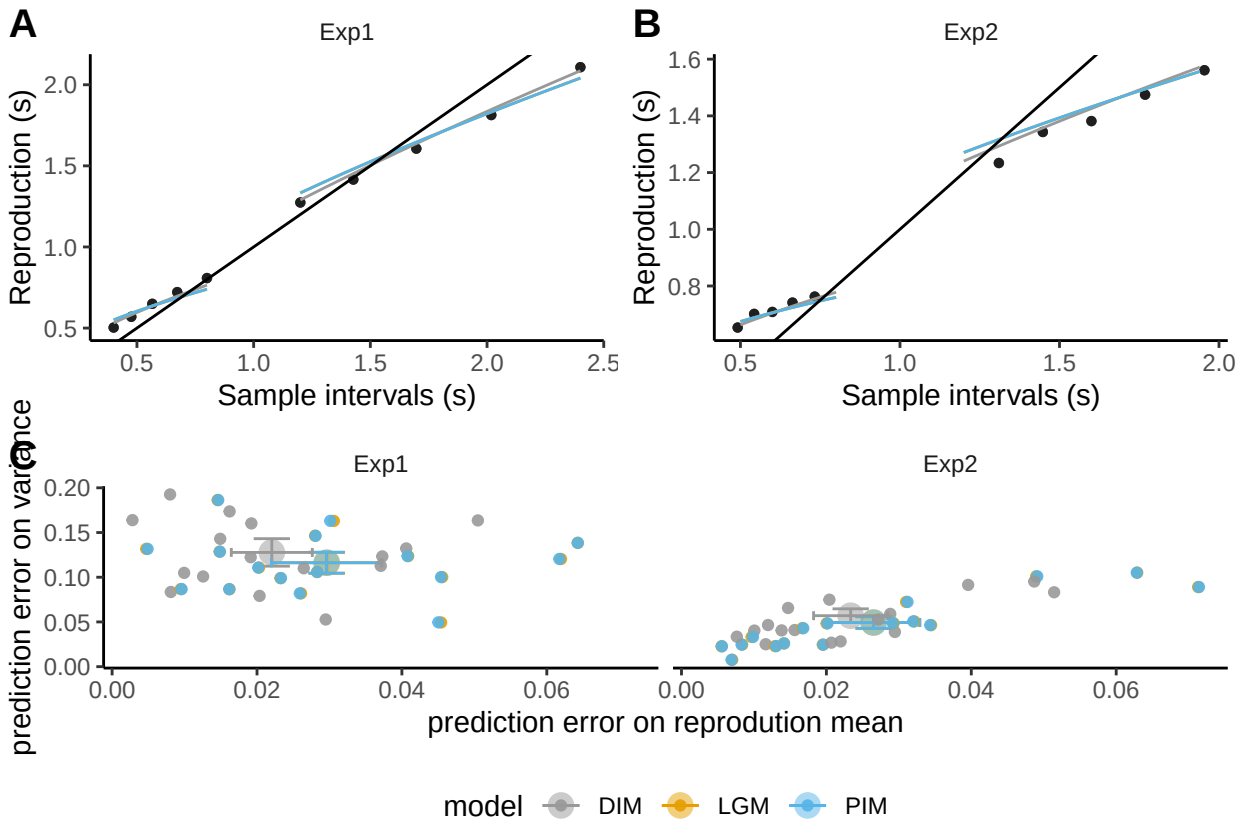
```
## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

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




```
library("cowplot")
fig_ModelRlt2 <- ggdraw() +
  draw_plot(RP_log_Exp1, x = 0, y = .5, width = .5, height = .5) +
  draw_plot(RP_log_Exp2, x = .5, y = .5, width = .5, height = .5) +
  draw_plot(plt_rErrorScatter + theme(legend.position = "bottom") , x = 0, y = 0, width = 1, height = 0
  draw_plot_label(label = c("A", "B", "C"), size = 15,
                  x = c(0, 0.5, 0), y = c(1, 1, 0.5))
```

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

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

```
ggsave(file.path(figure_path,'fig_ModelRlt2.png'), fig_ModelRlt2, width = 7, height = 7)
fig_ModelRlt2
```




```
fig_ModelRlt <- ggarrange(fig_mrepError_model, plt_rErrorScatter, 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: position_dodge requires non-overlapping x intervals
```

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

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

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

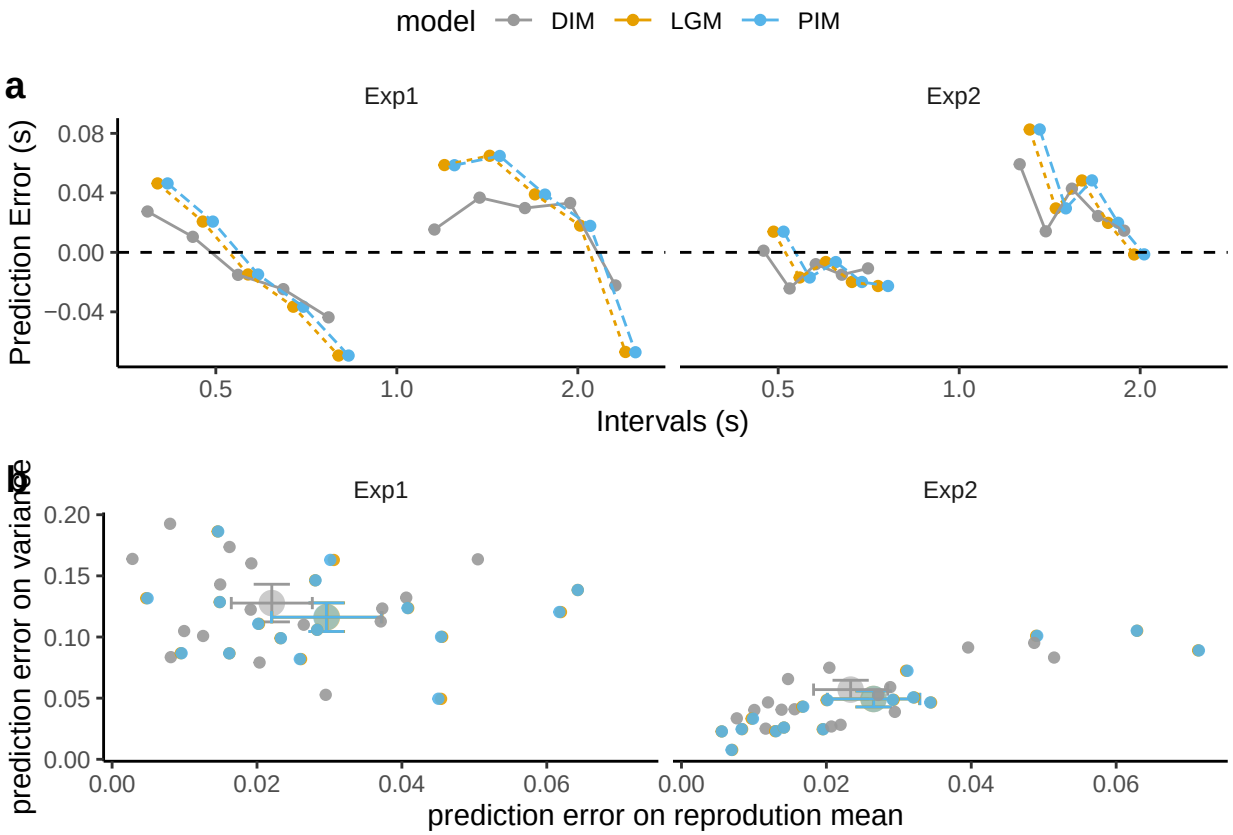
```
## Warning: position_dodge requires non-overlapping x intervals
```

```
## 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(file.path(figure_path,'fig_ModelRlt.png'), fig_ModelRlt, width = 7, height = 6)
fig_ModelRlt
```




```
fig_ModelRlt_all <- ggarrange(fig_mrepError_model_all, plt_rErrorScatter_all, common.legend = TRUE, nrow
```

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

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

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

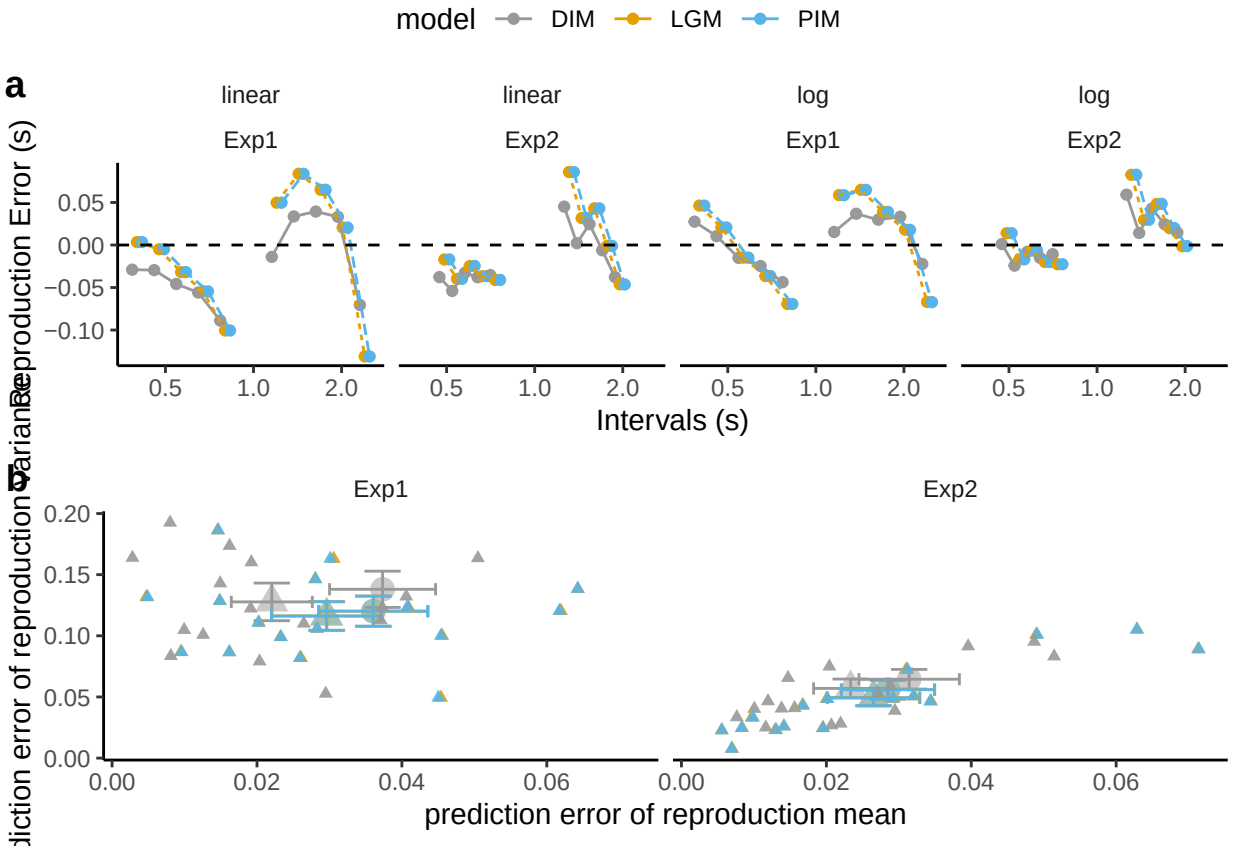
```
## Warning: position_dodge requires non-overlapping x intervals
```

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

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

```
## 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(file.path(figure_path,'fig_ModelRlt_all.png'), fig_ModelRlt_all, width = 7, height = 7)
fig_ModelRlt_all
```



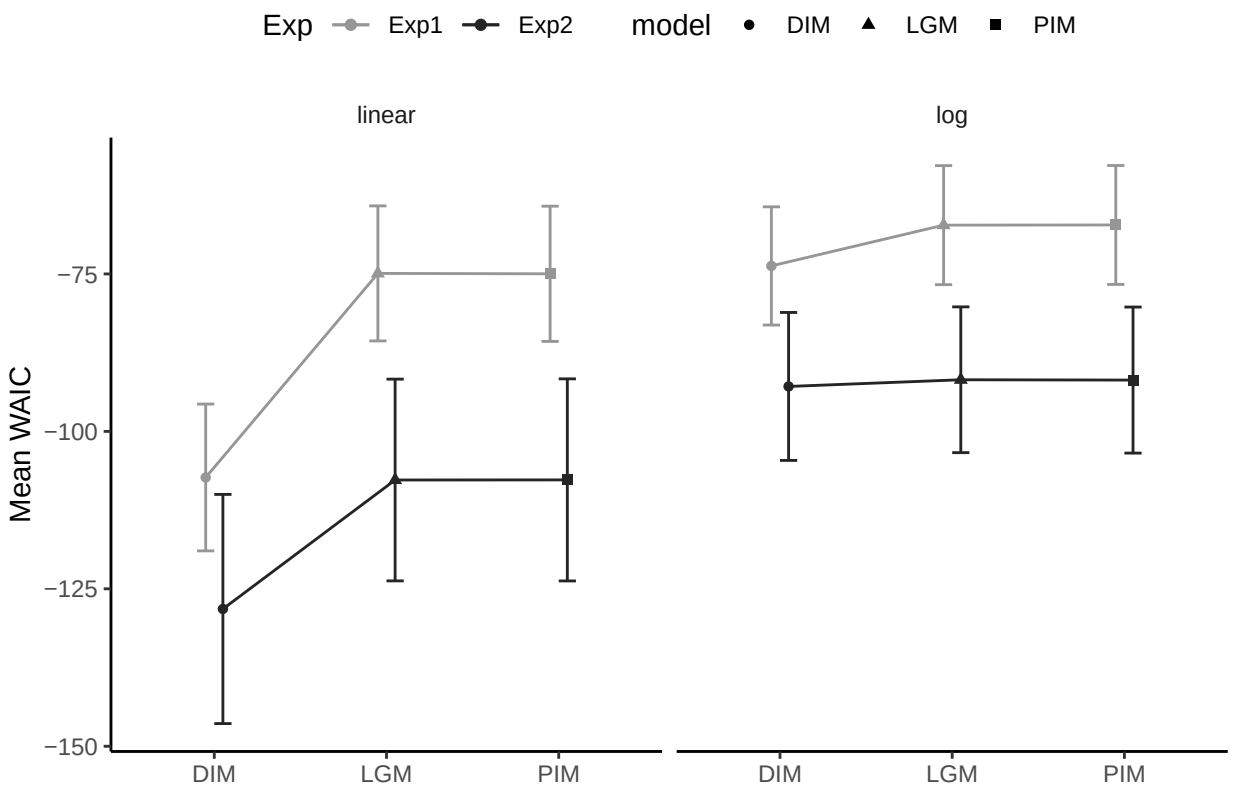

# 4 WAIC and LOO-CV

```
m_WAIC%>%filter(ver == 'log', part == 'all', model %in% c('LGM', 'DIM', 'PIM')) %>%select("Exp", "model
```

```
## Adding missing grouping variables: `ver`
```

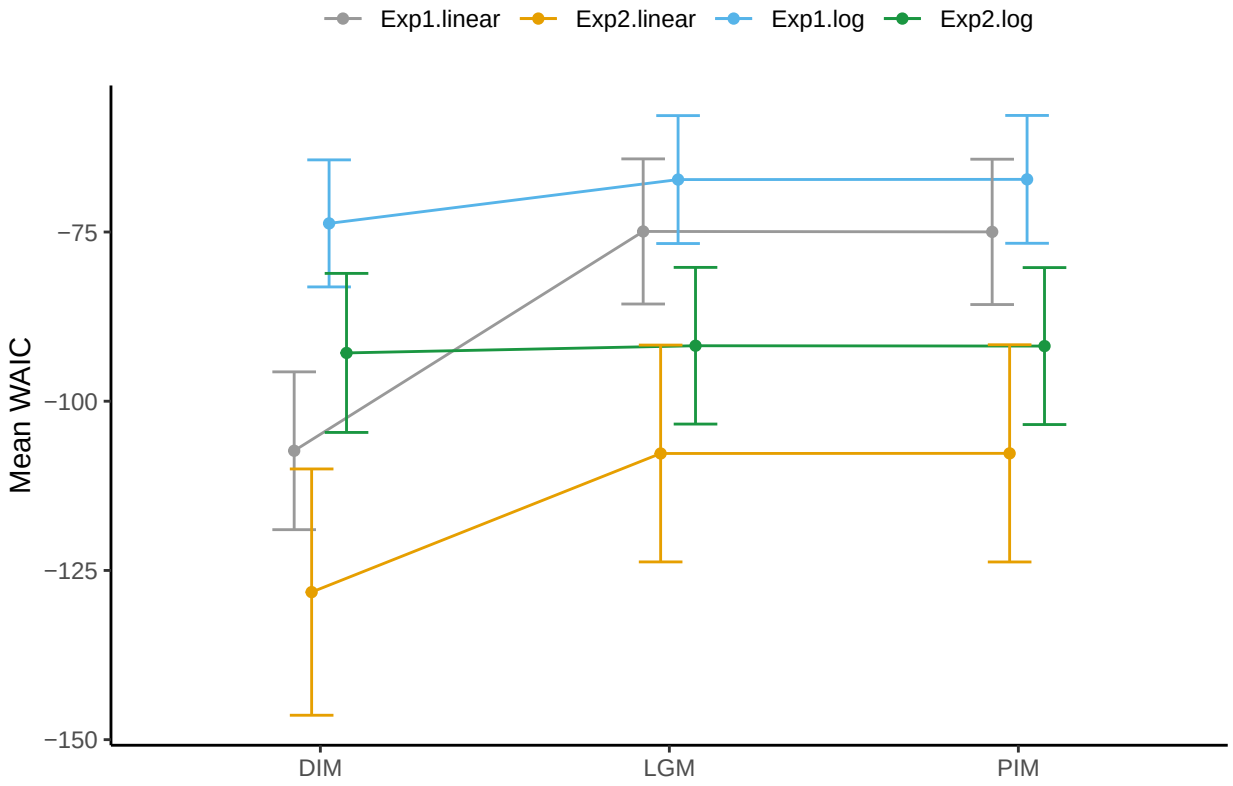
```
## # A tibble: 6 x 5
## # Groups:   Exp, model, ver [6]
##   ver   Exp   model m_waic m_looic
##   <chr> <chr> <chr>  <dbl>   <dbl>
## 1 log   Exp1  DIM    -73.7    45.2
## 2 log   Exp1  LGM    -67.3    51.5
## 3 log   Exp1  PIM    -67.2    52.0
## 4 log   Exp2  DIM    -92.9    26.3
## 5 log   Exp2  LGM    -91.8    27.0
## 6 log   Exp2  PIM    -91.9    26.4
```

```
plt_waic <- ggplot(m_WAIC %>% filter(part == 'all', model %in% c("DIM","LGM", "PIM")), aes(model, m_waic
  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)) +
  labs(x = " ", y = "Mean WAIC") +
  theme_minimal()+ theme_new + facet_wrap(~ver)+
  scale_color_manual(values = mycolors)

plt_waic
```

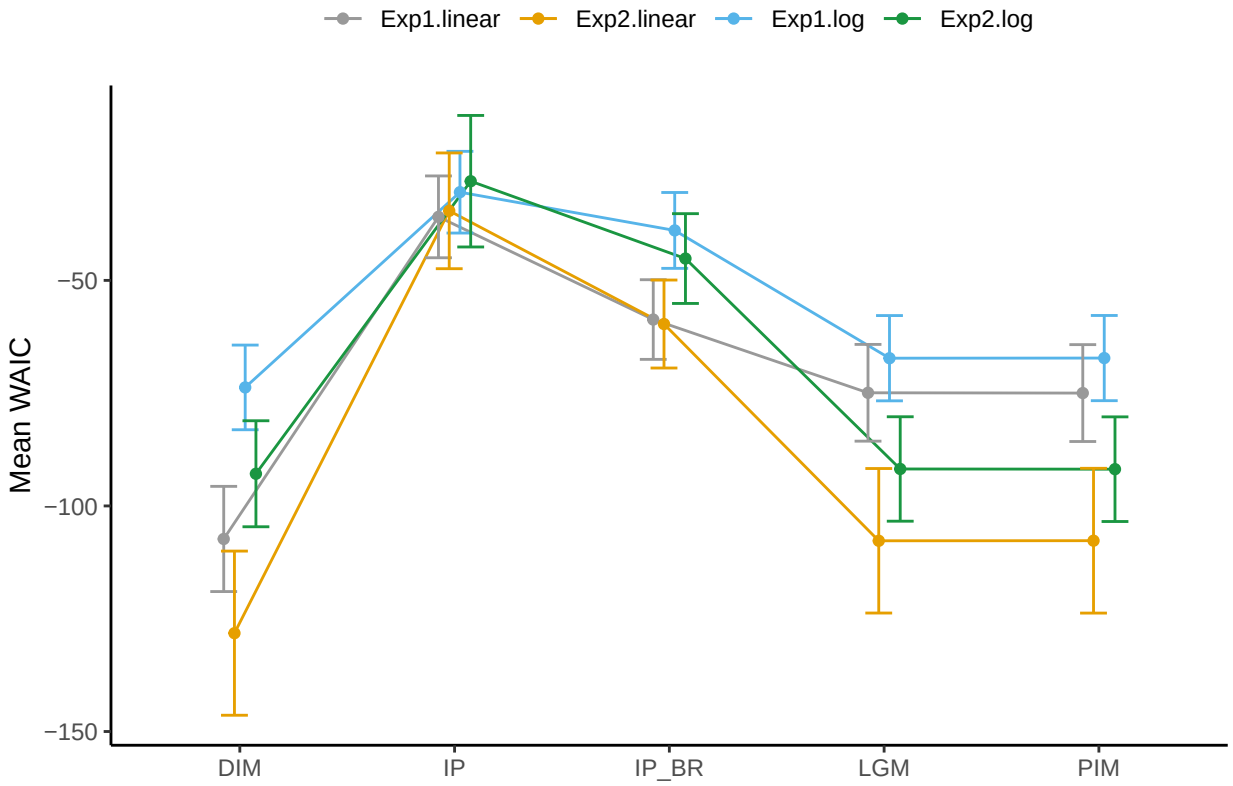



```
ggsave(file.path(figure_path,'plt_waic.png'), plt_waic, width = 6, height = 4)
```

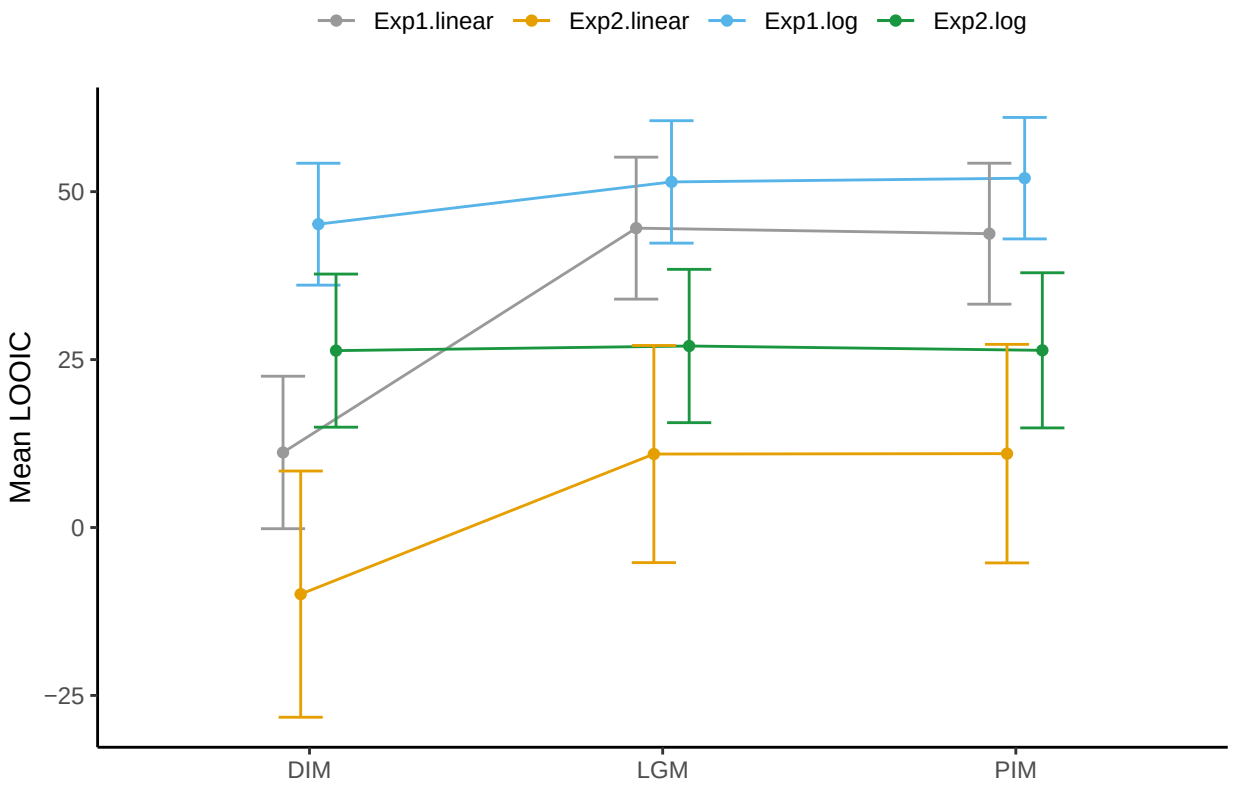
```
plt_waic <- ggplot(m_WAIC %>% filter(part == 'all', model %in%c("DIM","LGM", "PIM")), aes(model, m_waic
  geom_line(stat = "identity",position = position_dodge(width = 0.2))+
  geom_point(stat = "identity",position = position_dodge(width = 0.2))+
  geom_errorbar(width=.5,  position = position_dodge(width = 0.2)) +
  labs(x = " ", y = "Mean WAIC", color = "") +
  theme_minimal()+ theme_new+ colorSet4

plt_waic
```



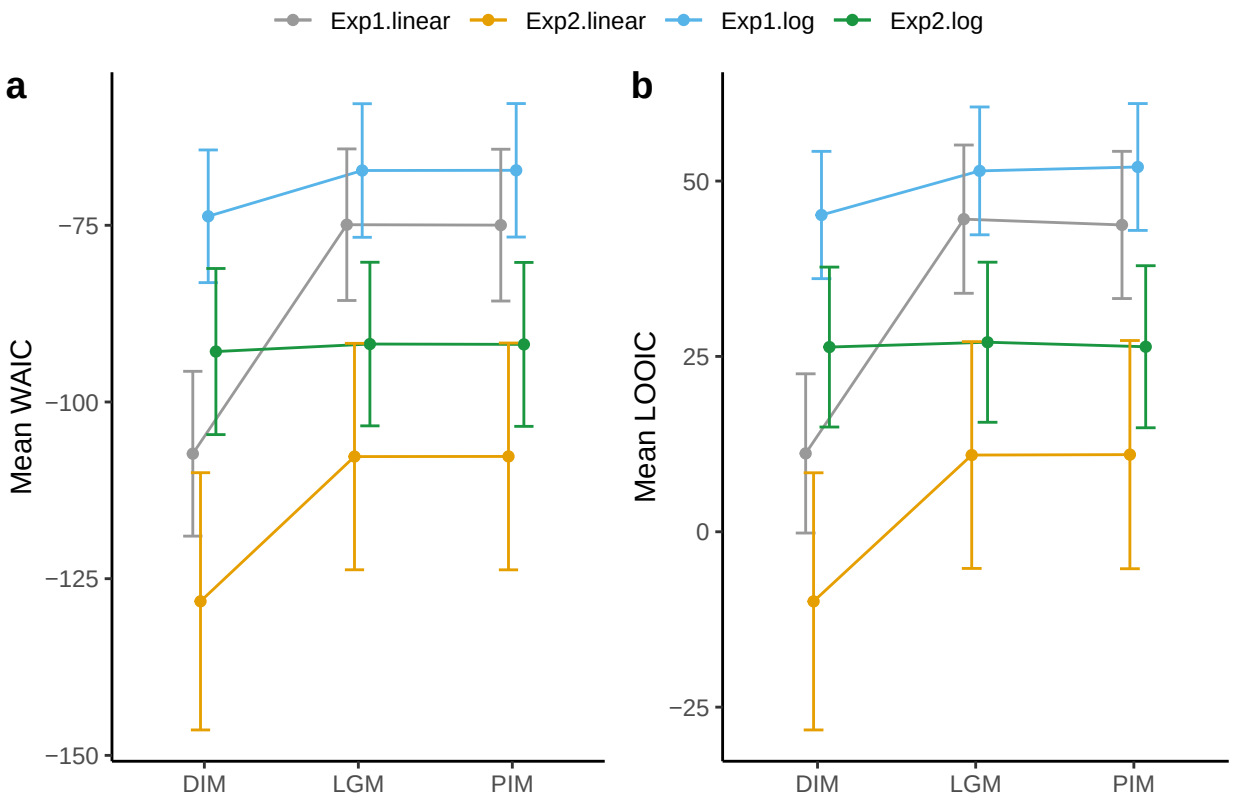

```
plt_waic_all <- ggplot(m_WAIC %>% filter(part == 'all'), aes(model, m_waic, ymin = m_waic - se_waic, yma
  geom_line(stat = "identity",position = position_dodge(width = 0.2))+
  geom_point(stat = "identity",position = position_dodge(width = 0.2))+
  geom_errorbar(width=.5,  position = position_dodge(width = 0.2)) +
  labs(x = " ", y = "Mean WAIC", color = "") +
  theme_minimal()+ theme_new+ colorSet4

plt_waic_all
```




```
plt_looic <- ggplot(m_WAIC %>% filter(part == 'all', model %in%c("DIM","LGM", "PIM")), aes(model, m_looi
  geom_line(stat = "identity",position = position_dodge(width = 0.2))+
  geom_point(stat = "identity",position = position_dodge(width = 0.2))+
  geom_errorbar(width=.5,  position = position_dodge(width = 0.2)) +
  labs(x = " ", y = "Mean LOOIC", color ="") +
  theme_minimal()+ theme_new + colorSet4

plt_looic
```




```
library(ggpubr)
fig_waic_looic<- ggarrange(plt_waic, plt_looic, common.legend = TRUE, nrow = 1, labels = c('a', 'b'))

ggsave(file.path(figure_path,'fig_waic_looic.png'), fig_waic_looic, width = 7, height = 4)
fig_waic_looic
```


Exp1.linear
Exp2.linear
Exp1.log
Exp2.log
a
Mean WAIC
−75
−100
−125
−150
DIM
LGM
PIM
b
Mean LOOIC
50
25
0
−25
DIM
LGM
PIM