accuracy_models_per_grade

Mole Game

```r
#Load the data from the benchmark paper
# I need to change the data and include grades 1 and 2 but at the moment let's use this data to explore

train_logs_40_mdr <- readRDS(file = "/home/user-047/code_seyma/WMBenchmark/data/train_logs_40_mdr.rds")
items_40_temp<-readRDS(file = "/home/user-047/code_seyma/WMBenchmark/data/items_40_temp.rds")
```

```r
# now half of the students because there might be pre-registration
set.seed(871)
# Select unique students with grade info
students <- train_logs_40_mdr %>%
  select(user_id, grade) %>%
  distinct()
# Sample 50% of students within each grade
sampled_students <- students %>%
  group_by(grade) %>%
  sample_frac(0.5) %>%
  ungroup()

train_logs_40_mdr_half <- train_logs_40_mdr %>%
  filter(user_id %in% sampled_students$user_id)
```

```r
#Grade levels

grades <- unique(train_logs_40_mdr_half$grade)

for (g in grades) {
  assign(paste0("train_logs_40_mdr_half_grade", g),
         train_logs_40_mdr_half %>% filter(grade == g))
}
```

Grade 3

```r
# priors <- c(
#  set_prior("normal(0, 1)", class = "Intercept"),
#  set_prior("normal(0, 1)", class = "b", coef = "difficulty"),
#  set_prior("normal(0, 1.5)", class = "b", coef = "position_rate"),
#  set_prior("normal(0, 1)", class = "b", coef = "set_size"),
#  set_prior("normal(0, 1)", class = "b", coef = "duplicateTRUE"),
#  set_prior("normal(0, 1)", class = "b", coef = "structuredTRUE"),
#  set_prior("normal(0, 1)", class = "b", coef = "last_itemTRUE"),
#  set_prior("normal(0, 1)", class = "b", coef = "last_second_itemTRUE"),
#  set_prior("normal(0, 2)", class = "sd")
# )

# mod_bys_accuracy_mole_grade3 <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate+ set_size + duplicate + struc
#   data = train_logs_40_mdr_half_grade3_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4
# )
# saveRDS(mod_bys_accuracy_mole_grade3, file = "~/code_seyma/WMDevelopmentProject/mod_bys_accuracy_mole
```

```
mod_bys_accuracy_mole_grade3 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod_bys_accura
```

```
mod_bys_accuracy_mole_grade3
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + duplicat
##    Data: train_logs_40_mdr_half_grade3_long (Number of observations: 681585)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 600)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.34      0.01     0.31     0.36 1.00     1191     2606
##
## ~user_id (Number of levels: 2360)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.56      0.01     0.54     0.58 1.00     1322     2228
##
## Regression Coefficients:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                2.64      0.05     2.54     2.75 1.01      984
## difficulty              -0.04      0.01    -0.05    -0.02 1.00     5204
## position_rate           -1.38      0.05    -1.48    -1.28 1.00     3660
## set_size                -0.10      0.01    -0.13    -0.07 1.01      990
## duplicateTRUE            0.08      0.02     0.04     0.12 1.00     8434
## structuredTRUE           0.13      0.01     0.11     0.15 1.00     6730
## last_itemTRUE            1.07      0.03     1.01     1.14 1.00     3700
## last_second_itemTRUE     0.28      0.02     0.25     0.32 1.00     4030
##                      Tail_ESS
## Intercept                2048
## difficulty               5042
## position_rate            3761
## set_size                 2156
## duplicateTRUE            4700
## structuredTRUE           5316
## last_itemTRUE            3994
## last_second_itemTRUE     3991
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Grade 4

```
# mod_bys_accuracy_mole_grade4 <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate+ set_size + duplicate + struc
#   data = train_logs_40_mdr_half_grade4_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4
# )
# saveRDS(mod_bys_accuracy_mole_grade4, file = "~/code_seyma/WMDevelopmentProject/mod_bys_accuracy_mole
```

```
mod_bys_accuracy_mole_grade4 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod_bys_accurac
```

```
mod_bys_accuracy_mole_grade4
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + duplicate
##    Data: train_logs_40_mdr_half_grade4_long (Number of observations: 624728)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 600)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.36      0.01     0.34     0.39 1.01      931     1941
##
## ~user_id (Number of levels: 2830)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.61      0.01     0.59     0.63 1.00     1254     2383
##
## Regression Coefficients:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                3.17      0.06     3.05     3.30 1.00     1187
## difficulty              -0.03      0.01    -0.04    -0.01 1.00     5230
## position_rate           -1.31      0.05    -1.40    -1.22 1.00     4209
## set_size                -0.20      0.02    -0.23    -0.17 1.00     1102
## duplicateTRUE            0.09      0.02     0.05     0.13 1.00     9257
## structuredTRUE           0.10      0.01     0.08     0.12 1.00     6728
## last_itemTRUE            1.01      0.03     0.95     1.07 1.00     4334
## last_second_itemTRUE     0.26      0.02     0.22     0.29 1.00     4874
##                      Tail_ESS
## Intercept                2048
## difficulty               4788
## position_rate            4329
## set_size                 2009
## duplicateTRUE            5048
## structuredTRUE           4804
## last_itemTRUE            4679
## last_second_itemTRUE     4435
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Grade 5

```
# mod_bys_accuracy_mole_grade5 <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate+ set_size + duplicate + struc
#   data = train_logs_40_mdr_half_grade5_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4
# )
# saveRDS(mod_bys_accuracy_mole_grade5, file = "~/code_seyma/WMDevelopmentProject/models/mod_bys_accura
```

```
mod_bys_accuracy_mole_grade5 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod_bys_accura
```

```
mod_bys_accuracy_mole_grade5
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + duplicate
##    Data: train_logs_40_mdr_half_grade5_long (Number of observations: 338651)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 600)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.38      0.02     0.35     0.42 1.00     1321     2474
##
## ~user_id (Number of levels: 1771)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.63      0.02     0.60     0.66 1.00     1454     2604
##
## Regression Coefficients:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                3.34      0.08     3.18     3.49 1.00     1875
## difficulty              -0.06      0.01    -0.08    -0.04 1.00     5543
## position_rate           -1.20      0.06    -1.31    -1.08 1.00     6586
## set_size                 -0.20      0.02    -0.24    -0.16 1.00     1728
## duplicateTRUE             0.13      0.03     0.08     0.18 1.00    10188
## structuredTRUE            0.09      0.01     0.06     0.12 1.00     9365
## last_itemTRUE             0.91      0.04     0.84     0.99 1.00     6744
## last_second_itemTRUE      0.21      0.02     0.17     0.26 1.00     6968
##                      Tail_ESS
## Intercept                3400
## difficulty               5168
## position_rate            4402
## set_size                 2551
## duplicateTRUE            5143
## structuredTRUE           5296
## last_itemTRUE            4731
## last_second_itemTRUE     4759
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# mod_accuracy_mole_grade5_interaction <- glmer(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + duplicate + stru
#   data = train_logs_40_mdr_half_grade5_long,
#   family = binomial,
#   control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 100000))
# )
# mod_accuracy_mole_grade5_interaction <- read("~/code_seyma/WMDevelopmentProject/mod_accuracy_mole_gra
```

Grade 6

```
# mod_bys_accuracy_mole_grade6 <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate+ set_size + duplicate + struc
#   data = train_logs_40_mdr_half_grade6_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4
# )
# saveRDS(mod_bys_accuracy_mole_grade6, file = "~/code_seyma/WMDevelopmentProject/models/mod_bys_accura
```

```
mod_bys_accuracy_mole_grade6 <- readRDS( file = "~/code_seyma/WMDevelopmentProject/models/mod_bys_accura
```

```
mod_bys_accuracy_mole_grade6
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + duplicate
##    Data: train_logs_40_mdr_half_grade6_long (Number of observations: 200034)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
```

```
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 597)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.38      0.02     0.34     0.42 1.00     1417     2846
##
## ~user_id (Number of levels: 1019)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.67      0.02     0.63     0.71 1.00     1274     2383
##
## Regression Coefficients:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                3.33      0.11     3.11     3.54 1.00     2379
## difficulty              -0.05      0.02    -0.08    -0.02 1.00     5159
## position_rate           -1.21      0.07    -1.35    -1.07 1.00     6102
## set_size                -0.18      0.02    -0.23    -0.13 1.00     2272
## duplicateTRUE            0.08      0.03     0.01     0.14 1.00    11478
## structuredTRUE           0.06      0.02     0.02     0.09 1.00     9678
## last_itemTRUE            0.94      0.05     0.85     1.03 1.00     6298
## last_second_itemTRUE     0.22      0.03     0.16     0.28 1.00     6215
##                      Tail_ESS
## Intercept                3767
## difficulty               4700
## position_rate            4676
## set_size                 3832
## duplicateTRUE            4918
## structuredTRUE           5244
## last_itemTRUE            4725
## last_second_itemTRUE     5037
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Grade 7

```
# priors <- c(
#  set_prior("normal(0, 1)", class = "Intercept"),
#  set_prior("normal(0, 1)", class = "b", coef = "difficulty"),
#  set_prior("normal(0, 1.5)", class = "b", coef = "position_rate"),
#  set_prior("normal(0, 1)", class = "b", coef = "set_size"),
#  set_prior("normal(0, 1)", class = "b", coef = "duplicateTRUE"),
#  set_prior("normal(0, 1)", class = "b", coef = "structuredTRUE"),
#  set_prior("normal(0, 1)", class = "b", coef = "last_itemTRUE"),
#  set_prior("normal(0, 1)", class = "b", coef = "last_second_itemTRUE"),
#  set_prior("normal(0, 2)", class = "sd")
# )

# mod_bys_accuracy_mole_grade7 <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate+ set_size + duplicate + struc
#   data = train_logs_40_mdr_half_grade7_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4
# )
# saveRDS(mod_bys_accuracy_mole_grade7, file = "~/code_seyma/WMDevelopmentProject/models/mod_bys_accura
```

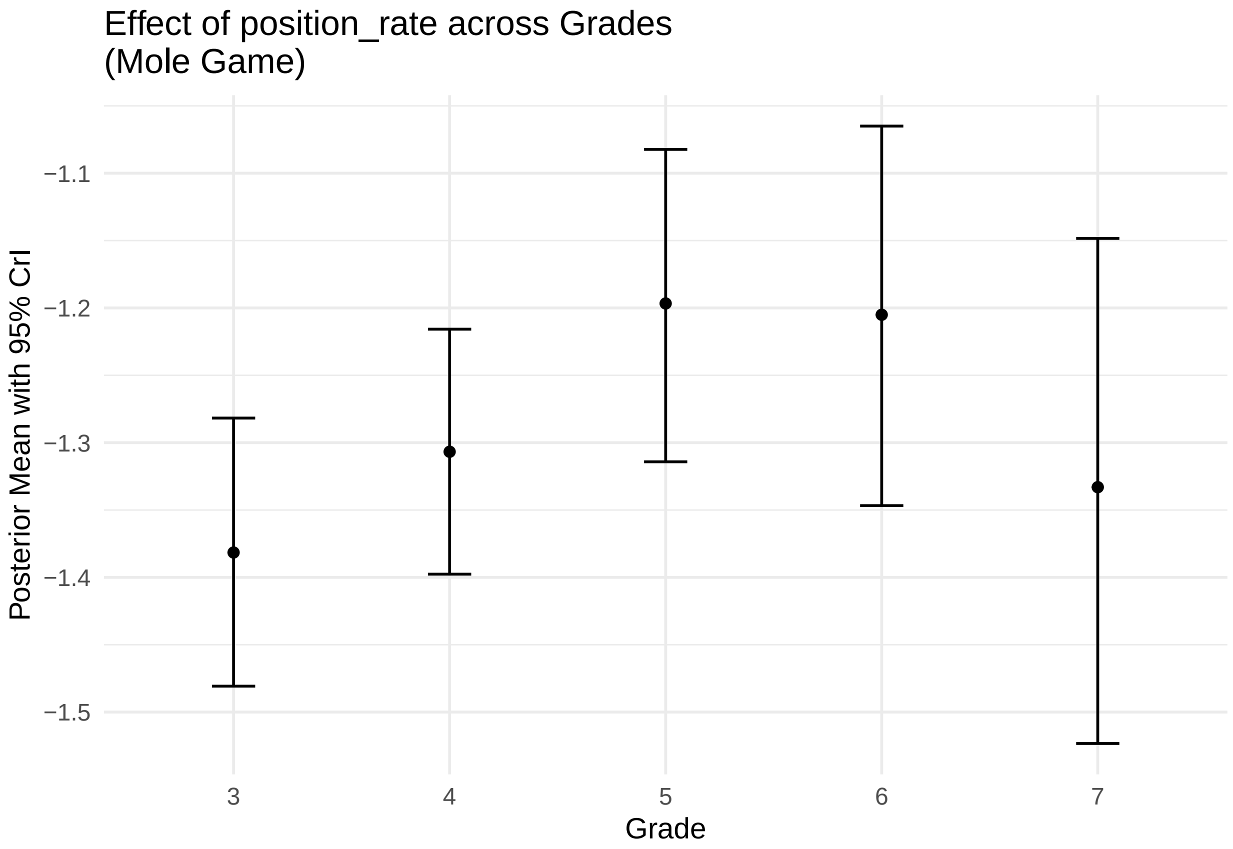
```
mod_bys_accuracy_mole_grade7 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod_bys_accura
```

```
mod_bys_accuracy_mole_grade7
```

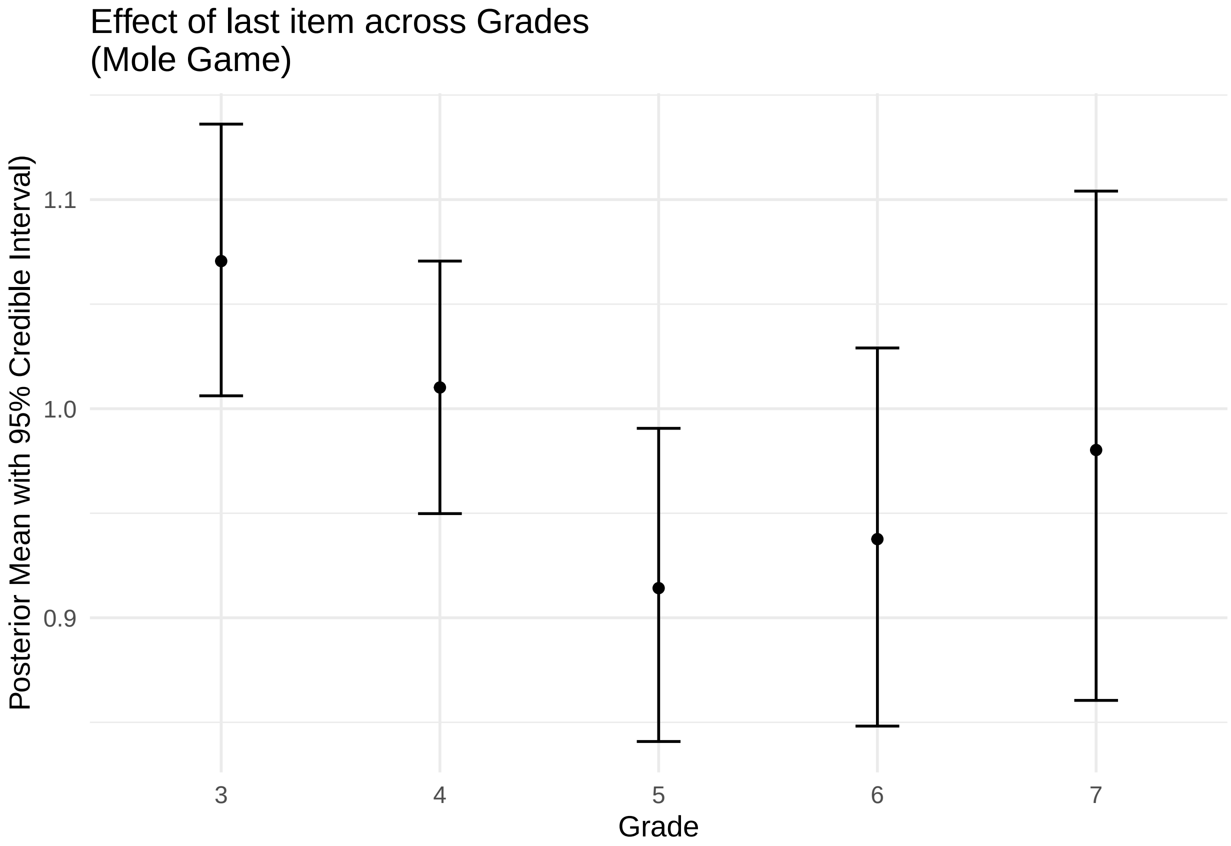
```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + duplicate
##    Data: train_logs_40_mdr_half_grade7_long (Number of observations: 99179)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 472)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.41      0.02     0.36     0.46 1.00     1465     2679
##
## ~user_id (Number of levels: 465)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.65      0.03     0.59     0.71 1.00     1382     2240
##
## Regression Coefficients:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                3.58      0.14     3.31     3.87 1.00     2944
## difficulty              -0.11      0.02    -0.16    -0.07 1.00     4317
## position_rate           -1.33      0.09    -1.52    -1.15 1.00     4145
## set_size                -0.19      0.03    -0.25    -0.13 1.00     2705
## duplicateTRUE            0.05      0.04    -0.03     0.14 1.00     8787
## structuredTRUE           0.07      0.02     0.02     0.12 1.00     7527
## last_itemTRUE            0.98      0.06     0.86     1.10 1.00     4715
## last_second_itemTRUE     0.25      0.04     0.17     0.33 1.00     4902
##                      Tail_ESS
## Intercept                3630
## difficulty               4464
## position_rate            4094
## set_size                 3401
## duplicateTRUE            4677
## structuredTRUE           4967
## last_itemTRUE            4211
## last_second_itemTRUE     4383
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Plots-Benchmarks by grade

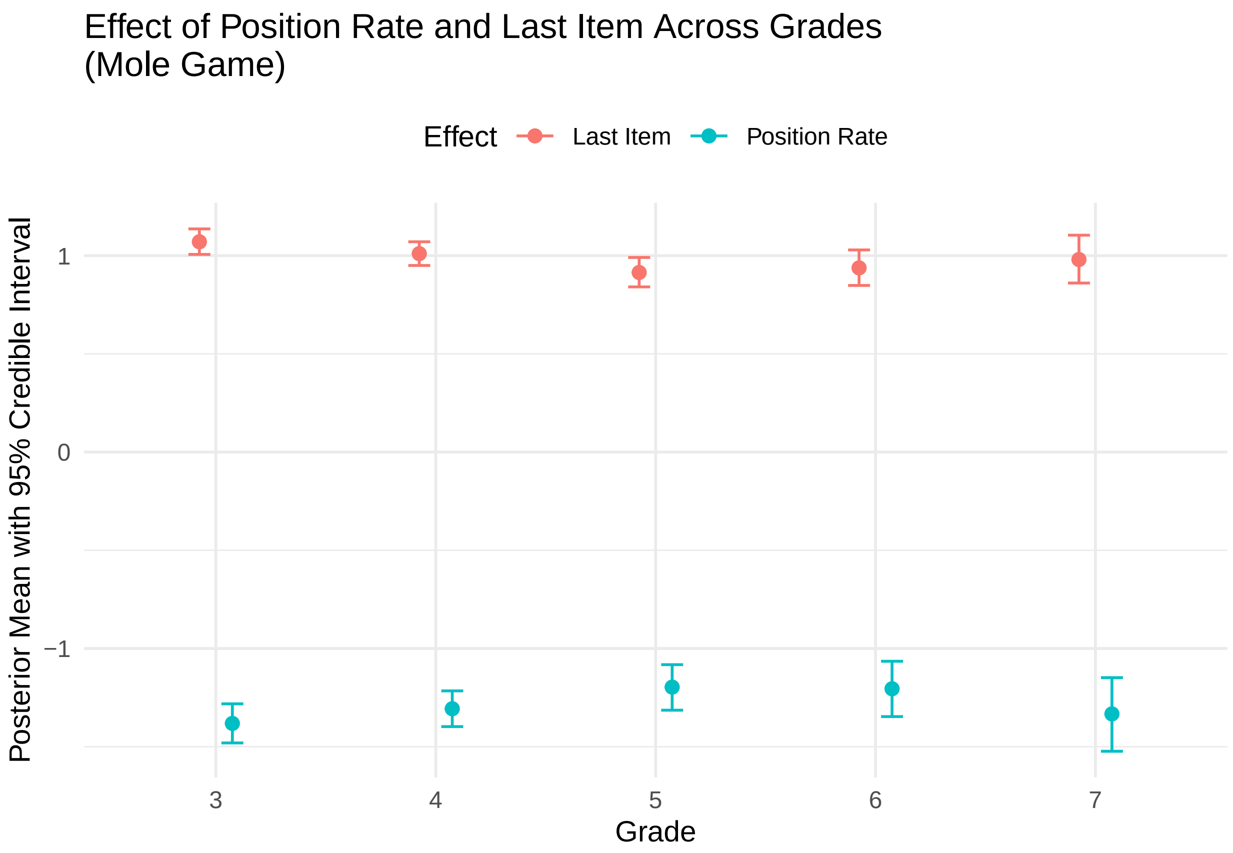
Primacy



Recency



```
grade_coefficients_mole$type <- "Position Rate"
grade_coefficients_mole_recency$type <- "Last Item"

grade_coefficients_combined <- bind_rows(
  grade_coefficients_mole,
  grade_coefficients_mole_recency
)

ggplot(grade_coefficients_combined,
       aes(x = factor(grade), y = estimate, color = type)) +
  geom_point(position = position_dodge(width = 0.3), size = 2) +
  geom_errorbar(aes(ymin = Q2.5, ymax = Q97.5),
                position = position_dodge(width = 0.3), width = 0.2) +
  labs(
    title = "Effect of Position Rate and Last Item Across Grades\n(Mole Game)",
    x = "Grade",
    y = "Posterior Mean with 95% Credible Interval",
    color = "Effect"
  ) +
  theme_minimal() +
  theme(
    legend.position = "top"
  )
```

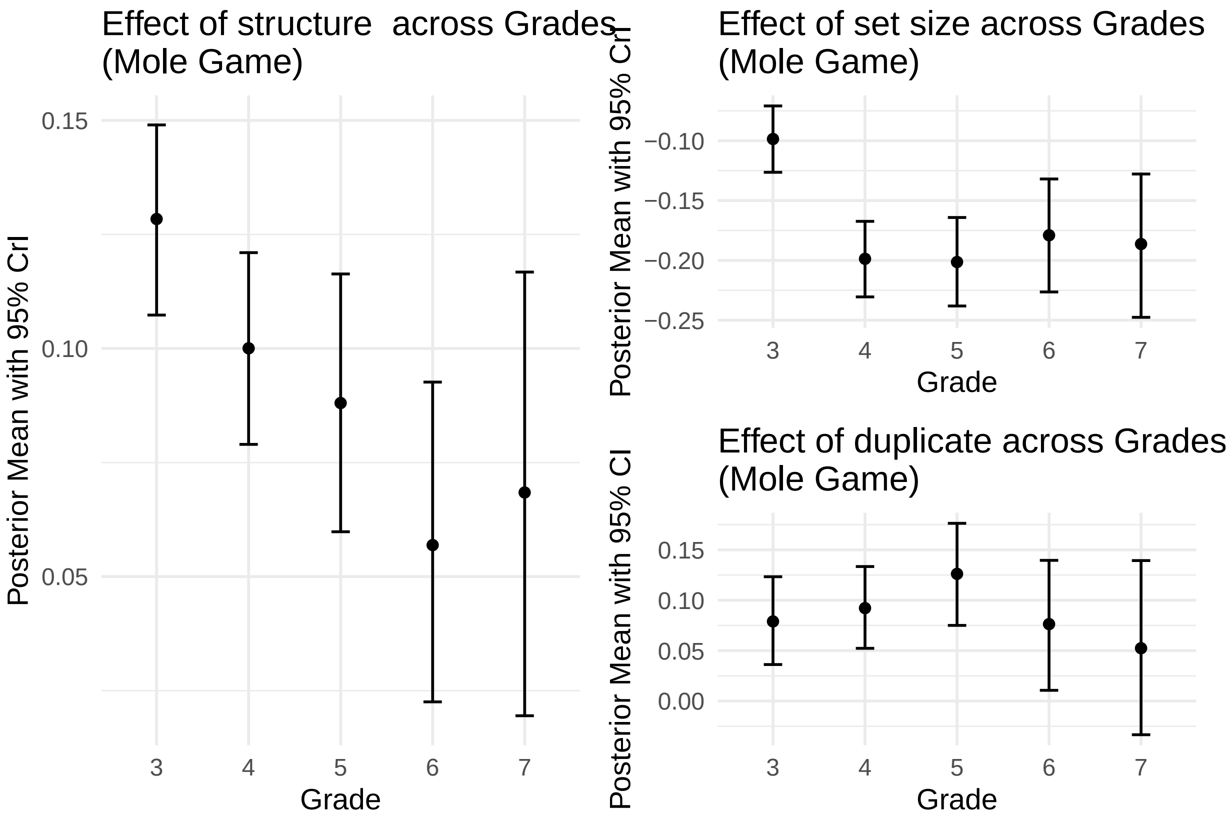


Set size

Structured

Duplicate

```
grade_coefficients_mole_structured_plot  + grade_coefficients_mole_setsize_plot /  grade_coefficients_mo
```

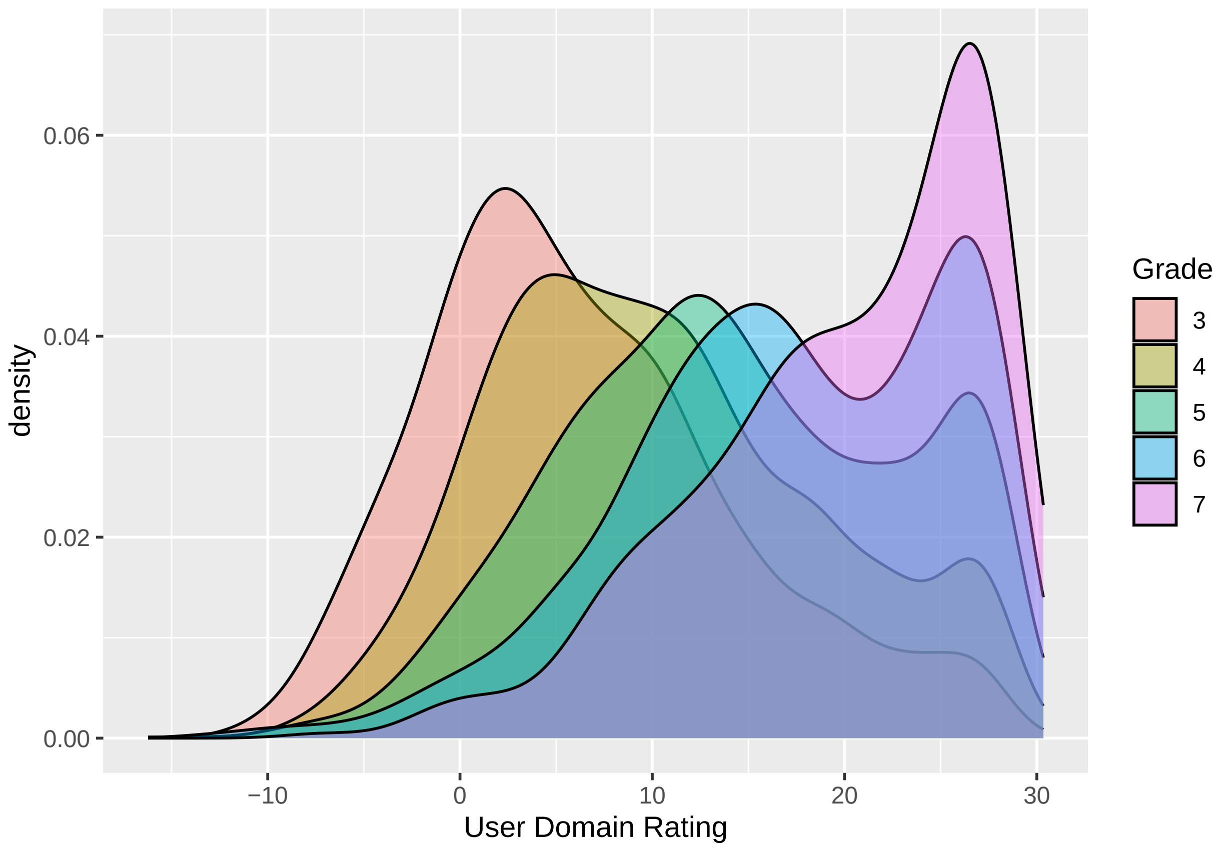


Students with Similar Ability

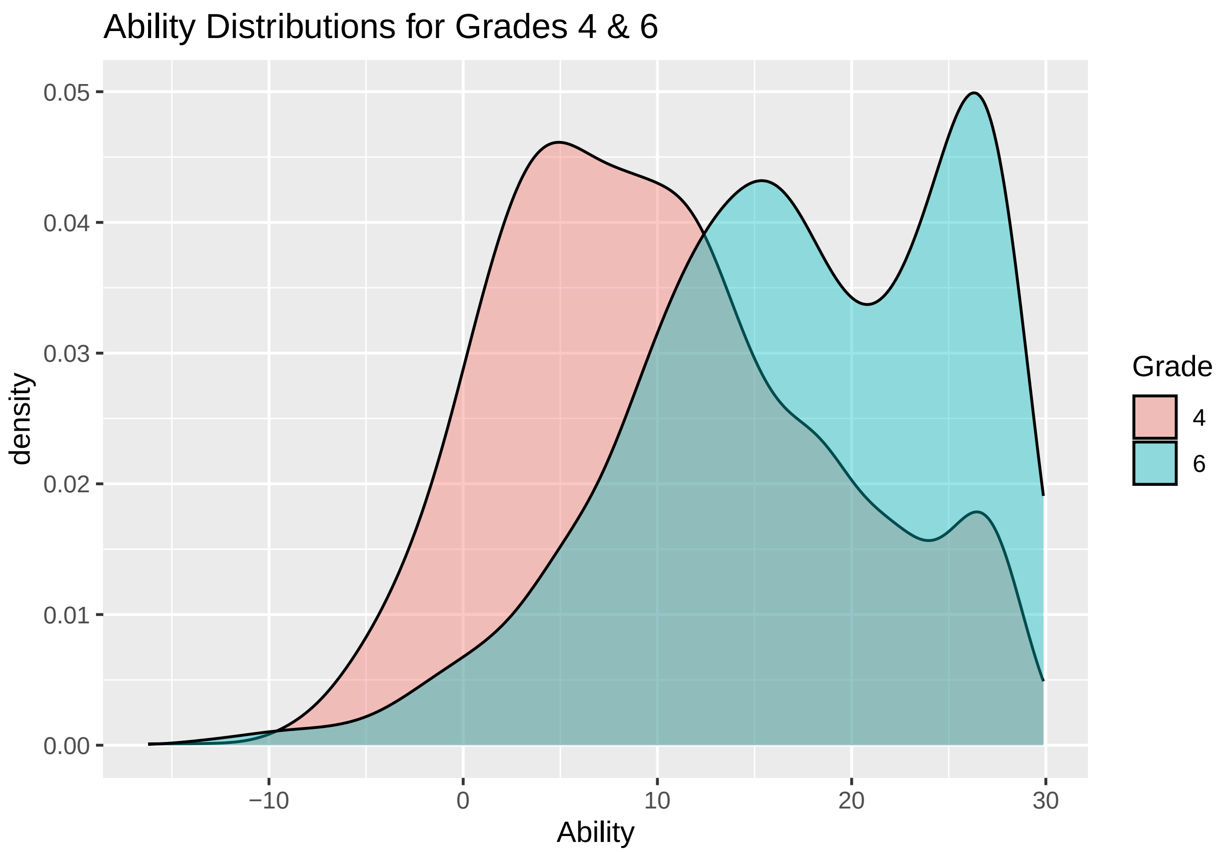
We have checked the estimates for each grade above. Now, we select students with similar ability from grades 4 and 6 to compare the estimates.

```r
train_logs_40_mdr_half_last_ability <- train_logs_40_mdr_half %>%
  group_by(user_id, grade) %>%
  filter(created == max(created)) %>%
  ungroup()
```

```r
train_logs_40_mdr_half_last_ability %>%
  ggplot(aes(x = new_user_domain_rating, fill = factor(grade))) +
  geom_density(alpha = 0.4) +
  labs(x = "User Domain Rating", fill = "Grade")
```



```
train_logs_40_mdr_half_last_ability %>%
  filter(grade %in% c(4, 6)) %>%
  ggplot(aes(x = new_user_domain_rating, fill = factor(grade))) +
  geom_density(alpha = 0.4) +
  labs(fill = "Grade", x = "Ability", title = "Ability Distributions for Grades 4 & 6")
```



```
train_logs_40_mdr_half_last_ability %>%
  filter(grade %in% c(4, 6)) %>%
  group_by(grade) %>%
  summarise(min = min(new_user_domain_rating),
            q1 = quantile(new_user_domain_rating, 0.25),
            median = median(new_user_domain_rating),
            q3 = quantile(new_user_domain_rating, 0.75),
            max = max(new_user_domain_rating),
            .groups = "drop")
```

```
## # A tibble: 2 x 6
##   grade   min    q1 median    q3   max
##   <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     4 -16.2  3.79   9.30  16.1  29.7
## 2     6 -13.2 11.7   17.4   24.6  29.9
```

```
similar_students <- train_logs_40_mdr_half_last_ability %>%
  filter(grade %in% c(4, 6),
         new_user_domain_rating > 14, #q1 rating for grade 6 : 11
         new_user_domain_rating < 16)  #q3 rating for grade 4 : 16
similar_students %>% count(grade)
```

```
## # A tibble: 2 x 2
##   grade     n
##   <dbl> <int>
## 1     4   164
## 2     6    92
```

```
# I want to pull userid-grade info and select in the logs

similar_students <- similar_students %>%
  mutate(user_grade = paste0(user_id, "_", grade)) %>%
  pull(user_grade)

train_logs_40_mdr_half_grade46_similarAbility <- train_logs_40_mdr_half %>%
  mutate(user_grade_logs = paste0(user_id, "_", grade)) %>%
  filter(user_grade_logs %in% similar_students)

train_logs_40_mdr_half_grade4_similarAbility <- train_logs_40_mdr_half_grade46_similarAbility %>%
  filter(grade == 4)

train_logs_40_mdr_half_grade6_similarAbility <- train_logs_40_mdr_half_grade46_similarAbility %>%
  filter(grade == 6)
```

Grade 4

```
# priors <- c(
#  set_prior("normal(0, 1)", class = "Intercept"),
#  set_prior("normal(0, 1)", class = "b", coef = "difficulty"),
#  set_prior("normal(0, 1.5)", class = "b", coef = "position_rate"),
#  set_prior("normal(0, 1)", class = "b", coef = "set_size"),
#  set_prior("normal(0, 1)", class = "b", coef = "duplicateTRUE"),
#  set_prior("normal(0, 1)", class = "b", coef = "structuredTRUE"),
#  set_prior("normal(0, 1)", class = "b", coef = "last_itemTRUE"),
#  set_prior("normal(0, 1)", class = "b", coef = "last_second_itemTRUE"),
#  set_prior("normal(0, 2)", class = "sd")
# )

# mod_bys_40_grade4_similarAbility_strict <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate+ set_size + duplicate + struc
#   data = train_logs_40_mdr_half_grade4_similarAbility_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4
# )
# saveRDS(mod_bys_40_grade4_similarAbility_strict, file = "~/code_seyma/WMDevelopmentProject/mod_bys_40
```

```
mod_bys_40_grade4_similarAbility_strict <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod
summary(mod_bys_40_grade4_similarAbility_strict)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + duplicat
##    Data: train_logs_40_mdr_half_grade4_similarAbility_long (Number of observations: 40183)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 370)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.47      0.04     0.39     0.56 1.00     1324     2489
##
## ~user_id (Number of levels: 164)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.48      0.04     0.41     0.55 1.00     1675     2821
##
## Regression Coefficients:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                3.52      0.20     3.13     3.92 1.00     2501
## difficulty               0.02      0.03    -0.04     0.07 1.00     4897
## position_rate           -1.12      0.19    -1.48    -0.75 1.00     2779
## set_size                -0.21      0.05    -0.30    -0.11 1.00     1986
## duplicateTRUE            0.22      0.08     0.06     0.38 1.00     6854
## structuredTRUE           0.02      0.04    -0.06     0.10 1.00     5111
## last_itemTRUE            0.80      0.12     0.56     1.03 1.00     2805
## last_second_itemTRUE     0.13      0.07    -0.01     0.27 1.00     3076
##                      Tail_ESS
## Intercept                3400
## difficulty               4537
## position_rate            3735
## set_size                 2871
## duplicateTRUE            4770
## structuredTRUE           4897
## last_itemTRUE            3808
## last_second_itemTRUE     4241
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Grade 6

```
# priors <- c(
#  set_prior("normal(0, 1)", class = "Intercept"),
#  set_prior("normal(0, 1)", class = "b", coef = "difficulty"),
#  set_prior("normal(0, 1.5)", class = "b", coef = "position_rate"),
#  set_prior("normal(0, 1)", class = "b", coef = "set_size"),
#  set_prior("normal(0, 1)", class = "b", coef = "duplicateTRUE"),
#  set_prior("normal(0, 1)", class = "b", coef = "structuredTRUE"),
#  set_prior("normal(0, 1)", class = "b", coef = "last_itemTRUE"),
#  set_prior("normal(0, 1)", class = "b", coef = "last_second_itemTRUE"),
#  set_prior("normal(0, 2)", class = "sd")
# )

# mod_bys_40_grade6_similarAbility_strict <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate+ set_size + duplicate + struc
#   data = train_logs_40_mdr_half_grade6_similarAbility_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4
# )
# saveRDS(mod_bys_40_grade6_similarAbility_strict, file = "~/code_seyma/WMDevelopmentProject/mod_bys_40_
```

```
mod_bys_40_grade6_similarAbility_strict <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod_
summary(mod_bys_40_grade6_similarAbility_strict)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + duplicate
##    Data: train_logs_40_mdr_half_grade4_similarAbility_long (Number of observations: 40183)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 370)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.47      0.04     0.39     0.56 1.00     1324     2489
##
## ~user_id (Number of levels: 164)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.48      0.04     0.41     0.55 1.00     1675     2821
##
## Regression Coefficients:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                3.52      0.20     3.13     3.92 1.00     2501
## difficulty               0.02      0.03    -0.04     0.07 1.00     4897
## position_rate           -1.12      0.19    -1.48    -0.75 1.00     2779
## set_size                -0.21      0.05    -0.30    -0.11 1.00     1986
## duplicateTRUE            0.22      0.08     0.06     0.38 1.00     6854
## structuredTRUE           0.02      0.04    -0.06     0.10 1.00     5111
## last_itemTRUE            0.80      0.12     0.56     1.03 1.00     2805
## last_second_itemTRUE     0.13      0.07    -0.01     0.27 1.00     3076
##                      Tail_ESS
## Intercept                3400
## difficulty               4537
## position_rate            3735
## set_size                 2871
## duplicateTRUE            4770
## structuredTRUE           4897
## last_itemTRUE            3808
## last_second_itemTRUE     4241
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

###Plots

```r
capacity4mole <- train_logs_40_mdr_half_grade46_similarAbility %>%
  filter(grade == 4) %>%
  group_by(user_id) %>%
  summarise(capacity = max(set_size))
capacity6mole <- train_logs_40_mdr_half_grade46_similarAbility %>%
  filter(grade == 6) %>%
  group_by(user_id) %>%
  summarise(capacity = max(set_size))

prop.table(table(capacity4mole$capacity))
```

```
##
##          5          6
## 0.98780488 0.01219512
```

```r
prop.table(table(capacity6mole$capacity))
```

```
##
##          4          5          6
## 0.01086957 0.95652174 0.03260870
```

```r
#Set size 5 is the most frequent ones for both grades but there is also setsize 4 in grade 6.

tbl4mole <- prop.table(table(capacity4mole$capacity)) * 100
percent_grade4mole <- paste(names(tbl4mole), "=", round(tbl4mole, 2), "%", collapse = ", ")


tbl6mole <- prop.table(table(capacity6mole$capacity)) * 100
percent_grade6mole <- paste(names(tbl6mole), "=", round(tbl6mole, 2), "%", collapse = ", ")
```

```r
capacity4moleAll <- train_logs_40_mdr_half_grade4 %>%
  group_by(user_id) %>%
  summarise(capacity = max(set_size))
capacity6moleAll <- train_logs_40_mdr_half_grade6 %>%
  group_by(user_id) %>%
  summarise(capacity = max(set_size))

prop.table(table(capacity4moleAll$capacity))
```

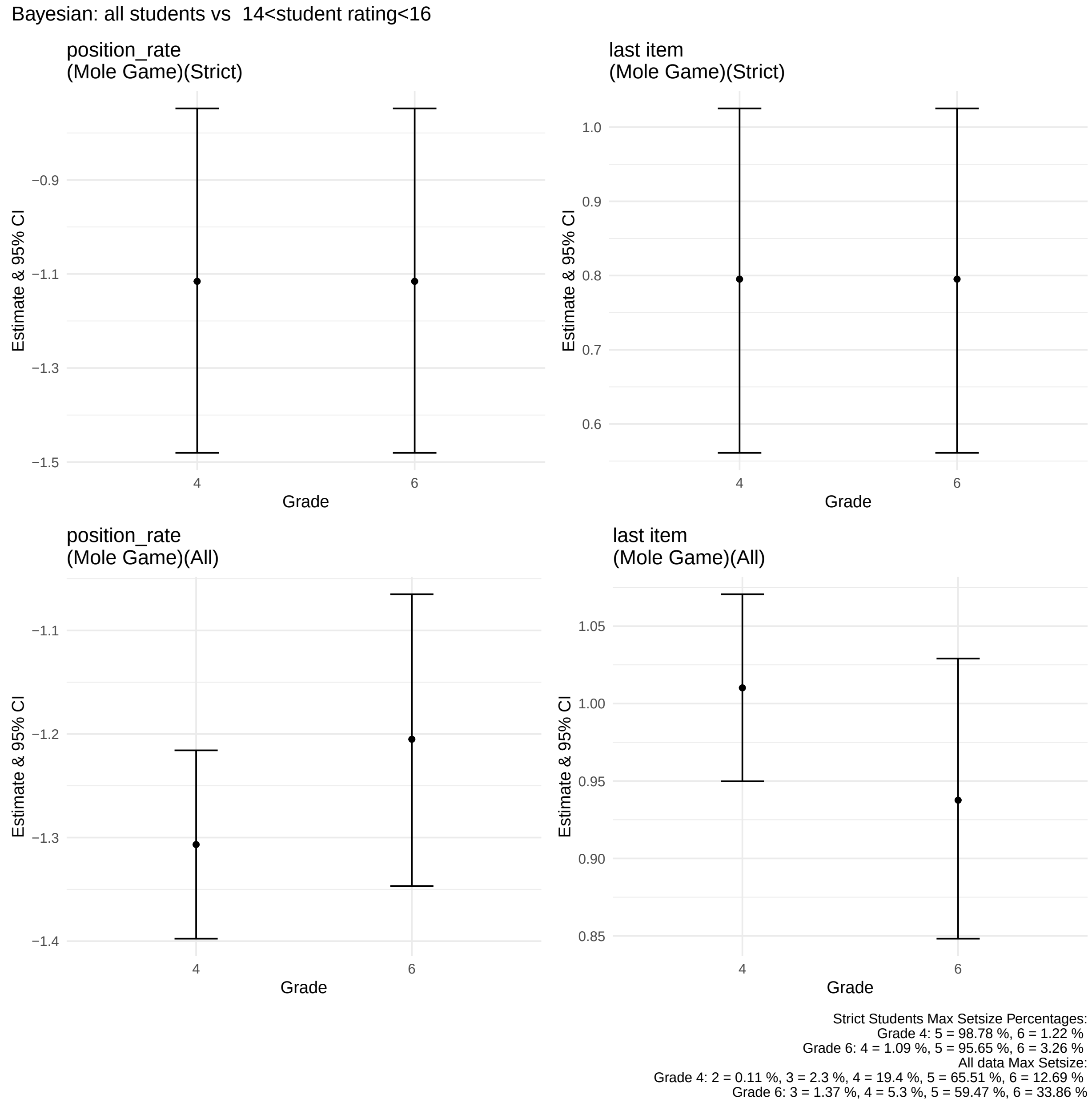
```
##
##           2           3           4           5           6
## 0.001060071 0.022968198 0.193992933 0.655123675 0.126855124
```

```r
prop.table(table(capacity6moleAll$capacity))
```

```
##
##          3          4          5          6
## 0.01373896 0.05299313 0.59470069 0.33856722
```

```
tbl4moleALL <- prop.table(table(capacity4moleAll$capacity)) * 100
percent_grade4ALLmole <- paste(names(tbl4moleALL), "=", round(tbl4moleALL, 2), "%", collapse = ", ")


tbl6moleALL <- prop.table(table(capacity6moleAll$capacity)) * 100
percent_grade6ALLmole <- paste(names(tbl6moleALL), "=", round(tbl6moleALL, 2), "%", collapse = ", ")
```

```
(grade46_strict_primacy_mole_plot + grade46_strict_recency_mole_plot) /
(grade46_primacy_mole_all_plot + grade46_recncy_mole_all_plot) +
  plot_annotation(title =  'Bayesian: all students vs  14<student rating<16',
    caption = paste("Strict Students Max Setsize Percentages:\nGrade 4:", percent_grade4mole, "\nGrade
  )
```



Interaction Models

In the mole game we check both set size interaction and structure interaction seperatly. mod0 is the model without interaction - what we have done above, mod1 is the interaction of set size with position rate and last item. I haven't included the second last item as the estimate is small and the model is already complex. mod2 is for the interaction between structure and position rate & last item.

- The model don't converge for grade 3.
- I could not compare models using BF, AIC, loo. All just failed. I used 10 percent data at some point and the I could compare some models. I add the results of comparisons at the end but in short, the complex model are not or barely better than simpler models statistically.

Grade 4

```
# mod1_accuracy_mole_bys_grade4 <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + duplicate + stru
#   data = train_logs_40_mdr_half_grade4_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4,
#   save_pars = save_pars(all = TRUE)
# )
# saveRDS(mod1_accuracy_mole_bys_grade4, file = "~/code_seyma/WMDevelopmentProject/mod1_accuracy_mole_b
```

```
mod1_accuracy_mole_bys_grade4 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod1_accuracy_
summary(mod1_accuracy_mole_bys_grade4)
```

Set Size Interaction

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + duplicate
##    Data: train_logs_40_mdr_half_grade4_long (Number of observations: 624728)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 600)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.36      0.01     0.34     0.39 1.00      879     1997
##
## ~user_id (Number of levels: 2830)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.61      0.01     0.59     0.63 1.01     1221     2145
##
## Regression Coefficients:
```

```
##                          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                    2.18      0.09     2.00     2.36 1.00     1371
## difficulty                  -0.03      0.01    -0.04    -0.01 1.00     3550
## position_rate                0.80      0.16     0.50     1.10 1.00     1844
## set_size                     0.02      0.02    -0.03     0.06 1.00     1183
## duplicateTRUE                0.13      0.02     0.09     0.17 1.00     7179
## structuredTRUE               0.06      0.01     0.04     0.08 1.00     4390
## last_itemTRUE                0.02      0.10    -0.18     0.21 1.00     1842
## last_second_itemTRUE         0.15      0.02     0.11     0.19 1.00     3330
## position_rate:set_size      -0.43      0.03    -0.49    -0.37 1.00     2038
## set_size:last_itemTRUE       0.19      0.02     0.15     0.23 1.00     2018
##                          Tail_ESS
## Intercept                    2628
## difficulty                   4385
## position_rate                3401
## set_size                     2277
## duplicateTRUE                4580
## structuredTRUE               4532
## last_itemTRUE                3425
## last_second_itemTRUE         4761
## position_rate:set_size       3288
## set_size:last_itemTRUE       3700
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# mod2_accuracy_mole_bys_grade4 <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * structured + duplicate + se
#   data = train_logs_40_mdr_half_grade4_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4,
#   save_pars = save_pars(all = TRUE)
# )
# saveRDS(mod2_accuracy_mole_bys_grade4, file = "~/code_seyma/WMDevelopmentProject/models/mod2_accuracy
```

```
mod2_accuracy_mole_bys_grade4 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod2_accuracy
summary(mod2_accuracy_mole_bys_grade4)
```

Structure Interaction

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * structured + duplica
##    Data: train_logs_40_mdr_half_grade4_long (Number of observations: 624728)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 600)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.37      0.01     0.34     0.40 1.00      735     1289
##
## ~user_id (Number of levels: 2830)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.61      0.01     0.59     0.63 1.01      936     1598
##
## Regression Coefficients:
##                              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                        3.12      0.07     2.99     3.25 1.00      593
## difficulty                      -0.03      0.01    -0.04    -0.01 1.00     3579
## position_rate                   -1.20      0.05    -1.30    -1.11 1.00     2445
## structuredTRUE                   0.26      0.04     0.18     0.34 1.00     2432
## duplicateTRUE                    0.09      0.02     0.05     0.14 1.00     5737
## set_size                        -0.19      0.02    -0.22    -0.16 1.00      580
## last_itemTRUE                    0.89      0.04     0.82     0.96 1.00     2322
## last_second_itemTRUE             0.28      0.02     0.24     0.32 1.00     2391
## position_rate:structuredTRUE    -0.36      0.07    -0.49    -0.22 1.00     2185
## structuredTRUE:last_itemTRUE     0.28      0.04     0.21     0.35 1.00     2407
##                              Tail_ESS
## Intercept                        1375
## difficulty                       4333
## position_rate                    3311
## structuredTRUE                   3571
## duplicateTRUE                    4486
## set_size                         1231
## last_itemTRUE                    3797
## last_second_itemTRUE             3412
## position_rate:structuredTRUE     3254
## structuredTRUE:last_itemTRUE     3576
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Grade 5

```
# mod1_accuracy_mole_bys_grade5 <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + duplicate + stru
#   data = train_logs_40_mdr_half_grade5_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4,
#   save_pars = save_pars(all = TRUE)
# )
# saveRDS(mod1_accuracy_mole_bys_grade5, file = "~/code_seyma/WMDevelopmentProject/model/mod1_accuracy_
```

```r
mod1_accuracy_mole_bys_grade5 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod1_accuracy
summary(mod1_accuracy_mole_bys_grade5)
```

Set Size Interaction

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + duplicat
##    Data: train_logs_40_mdr_half_grade5_long (Number of observations: 338651)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 600)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.38      0.02     0.35     0.42 1.00      866     1610
##
## ~user_id (Number of levels: 1771)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.63      0.02     0.60     0.66 1.00      786     1405
##
## Regression Coefficients:
##                         Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                   2.34      0.12     2.09     2.58 1.00     1205
## difficulty                 -0.06      0.01    -0.08    -0.04 1.00     2972
## position_rate               0.89      0.21     0.48     1.30 1.00     1571
## set_size                    0.01      0.03    -0.05     0.06 1.00     1063
## duplicateTRUE               0.16      0.03     0.10     0.21 1.00     7018
## structuredTRUE              0.06      0.01     0.03     0.08 1.00     4895
## last_itemTRUE              -0.07      0.13    -0.33     0.18 1.00     1695
## last_second_itemTRUE        0.12      0.03     0.07     0.17 1.00     3284
## position_rate:set_size     -0.42      0.04    -0.50    -0.34 1.00     1699
## set_size:last_itemTRUE      0.19      0.03     0.14     0.24 1.00     1882
##                         Tail_ESS
## Intercept                   2566
## difficulty                  3655
## position_rate               3026
## set_size                    2085
## duplicateTRUE               4967
## structuredTRUE              4279
## last_itemTRUE               3280
## last_second_itemTRUE        4022
## position_rate:set_size      2992
## set_size:last_itemTRUE      3319
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# mod2_accuracy_mole_bys_grade5 <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * structured + duplicate + se
#   data = train_logs_40_mdr_half_grade5_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4,
#   save_pars = save_pars(all = TRUE)
# )
#saveRDS(mod2_accuracy_mole_bys_grade5, file = "~/code_seyma/WMDevelopmentProject/models/mod2_accuracy_
```

```
mod2_accuracy_mole_bys_grade5 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod2_accuracy_
summary(mod2_accuracy_mole_bys_grade5)
```

Structure Interaction

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * structured + duplica
##    Data: train_logs_40_mdr_half_grade5_long (Number of observations: 338651)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 600)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.38      0.02     0.35     0.42 1.00      955     1674
##
## ~user_id (Number of levels: 1771)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.63      0.02     0.60     0.66 1.01      870     1824
##
## Regression Coefficients:
##                              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                        3.28      0.08     3.12     3.45 1.00     1124
## difficulty                      -0.06      0.01    -0.08    -0.04 1.00     3274
## position_rate                   -1.10      0.06    -1.22    -0.98 1.00     3210
## structuredTRUE                   0.22      0.05     0.12     0.32 1.00     2901
## duplicateTRUE                    0.12      0.03     0.07     0.18 1.00     5648
## set_size                        -0.20      0.02    -0.23    -0.16 1.00      882
## last_itemTRUE                    0.79      0.05     0.70     0.87 1.00     3120
## last_second_itemTRUE             0.24      0.03     0.19     0.28 1.00     3173
## position_rate:structuredTRUE    -0.31      0.08    -0.47    -0.15 1.00     2862
## structuredTRUE:last_itemTRUE     0.28      0.05     0.19     0.38 1.00     3406
##                              Tail_ESS
## Intercept                        2127
## difficulty                       4218
## position_rate                    3575
## structuredTRUE                   4061
## duplicateTRUE                       4689
## set_size                            1882
## last_itemTRUE                       3374
## last_second_itemTRUE                4314
## position_rate:structuredTRUE        3735
## structuredTRUE:last_itemTRUE        4331
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Grade 6

```
# mod1_accuracy_mole_bys_grade6 <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + duplicate + stru
#   data = train_logs_40_mdr_half_grade6_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4,
#   save_pars = save_pars(all = TRUE)
# )
# saveRDS(mod1_accuracy_mole_bys_grade6, file = "~/code_seyma/WMDevelopmentProject/models/mod1_accuracy_
```

```
mod1_accuracy_mole_bys_grade6 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod1_accuracy_
summary(mod1_accuracy_mole_bys_grade6)
```

Set Size Interaction

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + duplicate
##    Data: train_logs_40_mdr_half_grade6_long (Number of observations: 200034)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 597)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.38      0.02     0.34     0.42 1.00     1084     2351
##
## ~user_id (Number of levels: 1019)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.67      0.02     0.63     0.71 1.00      980     1755
##
## Regression Coefficients:
##                          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                    2.40      0.17     2.07     2.73 1.00     1510
## difficulty                -0.05      0.02    -0.08    -0.02 1.00     2253
## position_rate              0.72      0.27     0.20     1.26 1.00     1650
## set_size                   0.01      0.04    -0.06     0.08 1.00     1553
## duplicateTRUE              0.10      0.03     0.04     0.16 1.00     7088
## structuredTRUE             0.03      0.02    -0.00     0.07 1.00     5378
## last_itemTRUE             -0.03      0.17    -0.37     0.30 1.00     1801
## last_second_itemTRUE       0.15      0.03     0.08     0.21 1.00     3611
## position_rate:set_size    -0.38      0.05    -0.48    -0.28 1.00     1795
## set_size:last_itemTRUE     0.19      0.03     0.12     0.25 1.00     2007
##                        Tail_ESS
## Intercept                  2677
## difficulty                 4145
## position_rate              2777
## set_size                   2848
## duplicateTRUE              5053
## structuredTRUE             4552
## last_itemTRUE              3323
## last_second_itemTRUE       4176
## position_rate:set_size     2702
## set_size:last_itemTRUE     3845
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
# mod2_accuracy_mole_bys_grade6 <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * structured + duplicate + se
#   data = train_logs_40_mdr_half_grade6_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4,
#   save_pars = save_pars(all = TRUE)
# )
# saveRDS(mod2_accuracy_mole_bys_grade6, file = "~/code_seyma/WMDevelopmentProject/models/mod2_accuracy
```

```r
mod2_accuracy_mole_bys_grade6 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod2_accuracy
summary(mod2_accuracy_mole_bys_grade6)
```

Structure Interaction

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * structured + duplica
##    Data: train_logs_40_mdr_half_grade6_long (Number of observations: 200034)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
```

```
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 597) 
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.38      0.02     0.35     0.42 1.00     1168     2331
## 
## ~user_id (Number of levels: 1019) 
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.67      0.02     0.63     0.71 1.00     1052     2059
## 
## Regression Coefficients:
##                              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                        3.28      0.11     3.08     3.49 1.00     1344
## difficulty                      -0.05      0.02    -0.08    -0.02 1.00     3158
## position_rate                   -1.11      0.07    -1.25    -0.97 1.00     3429
## structuredTRUE                   0.13      0.06     0.00     0.25 1.00     3217
## duplicateTRUE                    0.07      0.03     0.01     0.14 1.00     7962
## set_size                        -0.18      0.02    -0.22    -0.13 1.00     1205
## last_itemTRUE                    0.79      0.05     0.68     0.89 1.00     3197
## last_second_itemTRUE             0.23      0.03     0.17     0.29 1.00     3559
## position_rate:structuredTRUE    -0.21      0.11    -0.41     0.00 1.00     3047
## structuredTRUE:last_itemTRUE     0.28      0.06     0.15     0.39 1.00     3383
##                              Tail_ESS
## Intercept                        2834
## difficulty                       4264
## position_rate                    4234
## structuredTRUE                   4075
## duplicateTRUE                    4483
## set_size                         2460
## last_itemTRUE                    3978
## last_second_itemTRUE             4577
## position_rate:structuredTRUE     3735
## structuredTRUE:last_itemTRUE     4067
## 
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Grade 7

```
#model0:

# mod_bys_accuracy_mole_grade7 <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate+ set_size + duplicate + struc
#   data = train_logs_40_mdr_half_grade7_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4
#   save_pars = save_pars(all = TRUE)
# )
# saveRDS(mod_bys_accuracy_mole_grade7, file = "~/code_seyma/WMDevelopmentProject/mod_bys_accuracy_mole
```

```r
mod0_accuracy_mole_bys_grade7 <-  readRDS("~/code_seyma/WMDevelopmentProject/models/mod_bys_accuracy_mol
```

####Set Size Interaction

```r
# mod1_accuracy_mole_bys_grade7 <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + duplicate + stru
#   data = train_logs_40_mdr_half_grade7_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4,
#   save_pars = save_pars(all = TRUE)
# )
# saveRDS(mod1_accuracy_mole_bys_grade7, file = "~/code_seyma/WMDevelopmentProject/mod1_accuracy_mole_b
```

```r
mod1_accuracy_mole_bys_grade7 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod1_accuracy_
summary(mod1_accuracy_mole_bys_grade7)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + duplicate
##    Data: train_logs_40_mdr_half_grade7_long (Number of observations: 99179)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 472)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.41      0.02     0.36     0.46 1.00     1348     2430
##
## ~user_id (Number of levels: 465)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.65      0.03     0.59     0.70 1.01      941     2013
##
## Regression Coefficients:
##                         Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                   2.74      0.23     2.30     3.17 1.00     1789
## difficulty                 -0.11      0.02    -0.16    -0.07 1.00     2353
## position_rate               0.26      0.37    -0.45     0.98 1.00     1940
## set_size                   -0.01      0.05    -0.10     0.08 1.00     1824
## duplicateTRUE               0.07      0.05    -0.01     0.16 1.00     7857
## structuredTRUE              0.05      0.03     0.00     0.10 1.00     7196
## last_itemTRUE               0.40      0.24    -0.07     0.87 1.00     2066
## last_second_itemTRUE        0.21      0.04     0.13     0.29 1.00     4863
## position_rate:set_size     -0.32      0.07    -0.46    -0.18 1.00     2097
## set_size:last_itemTRUE      0.11      0.05     0.02     0.20 1.00     2305
##                         Tail_ESS
## Intercept                   3294
## difficulty                  2959
## position_rate               3198
## set_size                    3222
## duplicateTRUE              4238
## structuredTRUE             5041
## last_itemTRUE              3364
## last_second_itemTRUE       4403
## position_rate:set_size     3427
## set_size:last_itemTRUE     3587
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

- The larger the set size the more primacy and receny effect. But the main effects are gone now.

####Structure Interaction

```
# mod2_accuracy_mole_bys_grade7 <- brm(
#   accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * structured + duplicate + se
#   data = train_logs_40_mdr_half_grade7_long,
#   family = bernoulli(),
#   prior = priors,
#   warmup = 500,
#   iter = 2000,
#   cores = 4,
#   save_pars = save_pars(all = TRUE)
# )
# saveRDS(mod2_accuracy_mole_bys_grade7, file = "~/code_seyma/WMDevelopmentProject/mod2_accuracy_mole_b
```

```
mod2_accuracy_mole_bys_grade7 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod2_accuracy_
summary(mod2_accuracy_mole_bys_grade7)
```

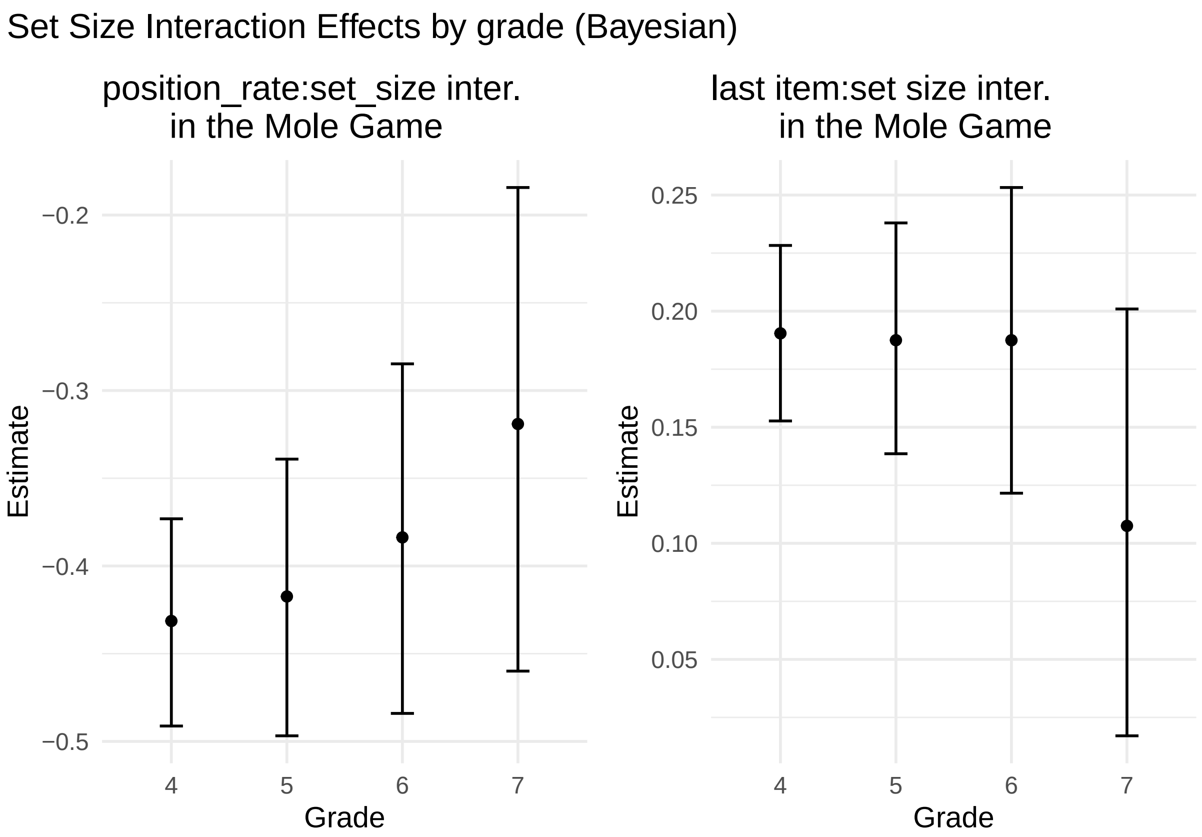
```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * structured + duplica
##    Data: train_logs_40_mdr_half_grade7_long (Number of observations: 99179)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 472)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.41      0.02     0.36     0.45 1.00     1785     3059
##
## ~user_id (Number of levels: 465)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.65      0.03     0.59     0.71 1.00     1258     2218
##
## Regression Coefficients:
##                             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                       3.56      0.15     3.27     3.84 1.00     2105
## difficulty                     -0.11      0.02    -0.16    -0.07 1.00     3299
## position_rate                  -1.25      0.10    -1.44    -1.05 1.00     4024
## structuredTRUE                  0.04      0.08    -0.13     0.20 1.00     3611
## duplicateTRUE                   0.04      0.04    -0.04     0.13 1.00     9543
## set_size                         -0.19      0.03    -0.25    -0.13 1.00     1839
## last_itemTRUE                     0.82      0.08     0.67     0.97 1.00     4046
## last_second_itemTRUE              0.24      0.05     0.15     0.32 1.00     3910
## position_rate:structuredTRUE     -0.03      0.14    -0.31     0.24 1.00     3439
## structuredTRUE:last_itemTRUE      0.23      0.08     0.07     0.39 1.00     3985
##                              Tail_ESS
## Intercept                         3437
## difficulty                        4040
## position_rate                     4131
## structuredTRUE                    4217
## duplicateTRUE                     4364
## set_size                          3044
## last_itemTRUE                     4458
## last_second_itemTRUE              4593
## position_rate:structuredTRUE      3960
## structuredTRUE:last_itemTRUE      4320
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

- position_rate seems to have a similar negative effect for both structured and unstructured cases. -being the last item boosts accuracy even more in structured sequences compared to unstructured ones.

I cannot compare the models. Loo, waic terminates R. bayes_factor function cannot compute log marg. likelyhood. I will try with less data. It's already half data with grade 7.

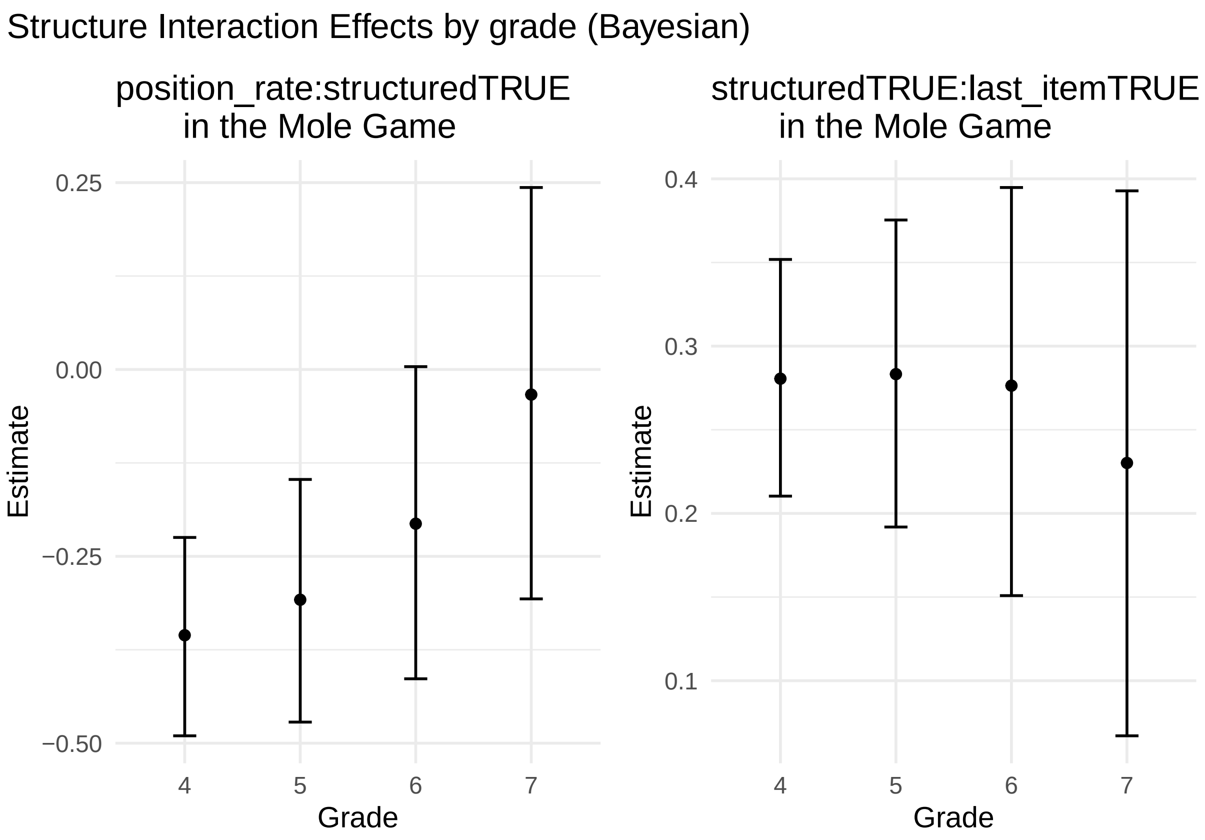
Mod1 estimate change by grade

```r
mod1_grade_recency_number_plot <- grade_recencySetsize_number %>%
  ggplot(aes(x = factor(grade), y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = Q2.5, ymax = Q97.5), width = 0.2) +
  labs(title = "last item:set size inter.
       in the Mole Game",
       x = "Grade", y = "Estimate") +
  theme_minimal()
mod1_grade_primacy_number_plot <- grade_primacySetsize_number %>%
  ggplot(aes(x = factor(grade), y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = Q2.5, ymax = Q97.5), width = 0.2) +
  labs(title = "position_rate:set_size inter.
       in the Mole Game",
       x = "Grade", y = "Estimate") +
  theme_minimal()

library(patchwork)
mod1_primacyRecency_plot <- mod1_grade_primacy_number_plot + mod1_grade_recency_number_plot +
  plot_annotation(title = "Set Size Interaction Effects by grade (Bayesian)")

mod1_primacyRecency_plot
```



### Mod2 estimate change by grade

```
mod2_grade_recency_number_plot <- grade_recencyStruct_number %>%
  ggplot(aes(x = factor(grade), y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = Q2.5, ymax = Q97.5), width = 0.2) +
  labs(title = "structuredTRUE:last_itemTRUE
       in the Mole Game",
       x = "Grade", y = "Estimate") +
  theme_minimal()
mod2_grade_primacy_number_plot <- grade_primacyStruct_number %>%
  ggplot(aes(x = factor(grade), y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = Q2.5, ymax = Q97.5), width = 0.2) +
  labs(title = "position_rate:structuredTRUE
       in the Mole Game",
       x = "Grade", y = "Estimate") +
  theme_minimal()

library(patchwork)
mod2_primacyRecency_plot <- mod2_grade_primacy_number_plot + mod2_grade_recency_number_plot +
  plot_annotation(title = "Structure Interaction Effects by grade (Bayesian)")

mod2_primacyRecency_plot
```



Model Comparisons for interaction models

We used 10 percent of the students.

```
# mod0_accuracy_mole_bys_grade7_10 <-  readRDS("~/code_seyma/WMDevelopmentProject/mod0_accuracy_mole_bys
# mod1_accuracy_mole_bys_grade7_10 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/mod1_accuracy_m
# mod2_accuracy_mole_bys_grade7_10 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/mod2_accuracy_m

# bridge0 <- bridge_sampler(mod0_accuracy_mole_bys_grade7_10, method = "warp3")
# bridge1 <- bridge_sampler(mod1_accuracy_mole_bys_grade7_10, method = "warp3")
# bridge2 <- bridge_sampler(mod2_accuracy_mole_bys_grade7_10, method = "warp3") #did not work :/

#setsize_int_bf <- bf(bridge0,bridge1 )
```

The BF for the set size interaction model is less than model without interaction (Estimated Bayes factor in favor of bridge0 over bridge1: 1469.59736). However, posterior for recency and setsize is different for 10 percent student. So i guess we need to count on the frequentist model comparison which says the model does not get better with interaction.

Summary Mole Game

1. **Models by grade**
   1. *Primacy*

      1. There is no clear developmental chances - *no linear relationship*
      2. There is a reverse u shape for the estimates
         1. *More primacy effect for grade 3 and getting smaller till 5* and then increasing again with increased CI
      3. Grade 3 takes too much time. Maybe different priors?
   2. *Recency*
      1. Same as the Primacy effect - *no linear relationship*
   3. *Set Size*
      1. *No change* except grade 3 - Rhat is 1.01 for set size in the model in grade 3 -
   4. *Duplicate*
      1. The reverse of primacy and recency- *no linear relationship*
   5. *Structure*
      1. Similar to primacy and recency (more effect for smaller grade levels). - *more linear except grade 7*
2. **Students with similar Ability (Comparing grade 4 and 6)**
   1. When comparing all students, we see that primacy and recency effects are bigger for grade 4
   2. *There is no difference between different grades who has similar ability levels*
3. **Interaction models**
   1. *The interaction models do not fit better*. They are too complicated.
   2. Set Size Interaction
      1. For grades 4, 5, 6, the interaction lose the main effect of set size and last item. For grade 7, position_rate main effect is also lost.
      2. The interaction of set size with the last item is similar between grade levels, except for grade 7.
      3. The interaction of set size with the position rate increases very slightly by grade.
   3. Structure Interaction
      1. Position_rate: Structure -> No in grade 7. Small increase from grade 3 to 6
      2. Last: Structure -> No chance by grade

To sum up, there is no linear relationship between grades. More primacy, receny, syructure effect for earlier grades till grade 5 then in grade 6 and 7 effects increase. Grade 7 is usually exception. Earlier grade models harder to fit.

Number Game

```
#Load the data from the benchmark paper
# I need to change the data and include grades 1 and 2 but at the moment let's use this data to explore

train_logs_66_mdr <- readRDS(file = "/home/user-047/code_seyma/WMBenchmark/data/train_logs_66_review.rds
items_66_temp<-readRDS( file = "/home/user-047/code_seyma/WMBenchmark/data/items_66_temp.rds")
```

```
# now half of the students because there would be always pre-registration
set.seed(871)
# Select unique students with grade info
students66 <- train_logs_66_mdr %>%
  select(user_id, grade) %>%
  distinct()
# Sample 50% of students within each grade
sampled_students66 <- students66 %>%
  group_by(grade) %>%
  sample_frac(0.5) %>%
  ungroup()

train_logs_66_mdr_half <- train_logs_66_mdr %>%
  filter(user_id %in% sampled_students66$user_id)
```

```
#Grade levels

grades <- unique(train_logs_66_mdr_half$grade)

for (g in grades) {
  assign(paste0("train_logs_66_mdr_half_grade", g),
         train_logs_66_mdr_half %>% filter(grade == g))
}
```

Grade 3

```
# priors <- c(
#   set_prior("normal(0, 1)", class = "Intercept"),
#   set_prior("normal(0, 1)", class = "b", coef = "difficulty"),
#   set_prior("normal(0, 1.5)", class = "b", coef = "position_rate"),
#   set_prior("normal(0, 1)", class = "b", coef = "set_size"),
#   set_prior("normal(0, 1)", class = "b", coef = "repetitionTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "duplicateTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "last_itemTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "last_second_itemTRUE"),
#   set_prior("normal(0, 2)", class = "sd")
# )

# mod_bys_accuracy_number_grade3 <- brm(
#     accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + repetition + d
#     data = train_logs_66_mdr_half_grade3_long,
#     family = bernoulli(),
#     prior = priors,
#     warmup = 500,
#     iter = 2000,
#     cores = 4)
# saveRDS(mod_bys_accuracy_number_grade3, file = "~/code_seyma/WMDevelopmentProject/mod_bys_accuracy_nu
```

```
mod_bys_accuracy_number_grade3 <-  readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod_bys_accu
summary(mod_bys_accuracy_number_grade3)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + repetiti
##    Data: train_logs_66_mdr_half_grade3_long (Number of observations: 609515)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 1006)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.43      0.02     0.41     0.46 1.01     1090     2032
##
## ~user_id (Number of levels: 2347)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.59      0.01     0.57     0.61 1.00     1254     2306
##
## Regression Coefficients:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                3.26      0.06     3.15     3.38 1.00     1105
## difficulty              -0.12      0.01    -0.14    -0.10 1.00     5529
## position_rate           -1.56      0.04    -1.65    -1.47 1.00     5108
## set_size                -0.15      0.01    -0.18    -0.12 1.00     1272
## repetitionTRUE           0.36      0.01     0.33     0.39 1.00    10196
## duplicateTRUE           -0.09      0.02    -0.12    -0.06 1.00     8877
## last_itemTRUE            1.05      0.03     0.99     1.10 1.00     4754
## last_second_itemTRUE     0.22      0.02     0.18     0.25 1.00     4824
##                      Tail_ESS
## Intercept                1896
## difficulty               4965
## position_rate            4557
## set_size                 2129
## repetitionTRUE           4739
## duplicateTRUE            4760
## last_itemTRUE            4176
## last_second_itemTRUE     4283
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Grade 4

```
# priors <- c(
#   set_prior("normal(0, 1)", class = "Intercept"),
#   set_prior("normal(0, 1)", class = "b", coef = "difficulty"),
#   set_prior("normal(0, 1.5)", class = "b", coef = "position_rate"),
#   set_prior("normal(0, 1)", class = "b", coef = "set_size"),
#   set_prior("normal(0, 1)", class = "b", coef = "repetitionTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "duplicateTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "last_itemTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "last_second_itemTRUE"),
#   set_prior("normal(0, 2)", class = "sd")
# )

# mod_bys_accuracy_number_grade4 <- brm(
#     accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + repetition + d
#     data = train_logs_66_mdr_half_grade4_long,
#     family = bernoulli(),
#     prior = priors,
#     warmup = 500,
#     iter = 2000,
#     cores = 4)
# saveRDS(mod_bys_accuracy_number_grade4, file = "~/code_seyma/WMDevelopmentProject/mod_bys_accuracy_nu
```

```
mod_bys_accuracy_number_grade4 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod_bys_accu
summary(mod_bys_accuracy_number_grade4)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + repetiti
##    Data: train_logs_66_mdr_half_grade4_long (Number of observations: 553223)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 1192)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.52      0.02     0.49     0.56 1.01      789     1785
##
## ~user_id (Number of levels: 2326)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.66      0.01     0.64     0.69 1.00     1185     2116
##
## Regression Coefficients:
##                       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                 3.82      0.06     3.70     3.94 1.01     1025
## difficulty               -0.15      0.01    -0.17    -0.13 1.00     4935
## position_rate            -1.66      0.04    -1.74    -1.59 1.00     4132
## set_size                 -0.23      0.01    -0.26    -0.21 1.01     1179
## repetitionTRUE            0.33      0.02     0.30     0.36 1.00     8217
## duplicateTRUE            -0.10      0.02    -0.13    -0.06 1.00     9004
## last_itemTRUE             1.11      0.03     1.06     1.16 1.00     4192
## last_second_itemTRUE      0.22      0.02     0.19     0.25 1.00     4154
##                       Tail_ESS
## Intercept                 2279
## difficulty                4499
## position_rate             4848
## set_size                  2024
## repetitionTRUE            4792
## duplicateTRUE             5288
## last_itemTRUE             4539
## last_second_itemTRUE      4184
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Grade 5

```
# priors <- c(
#   set_prior("normal(0, 1)", class = "Intercept"),
#   set_prior("normal(0, 1)", class = "b", coef = "difficulty"),
#   set_prior("normal(0, 1.5)", class = "b", coef = "position_rate"),
#   set_prior("normal(0, 1)", class = "b", coef = "set_size"),
#   set_prior("normal(0, 1)", class = "b", coef = "repetitionTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "duplicateTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "last_itemTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "last_second_itemTRUE"),
#   set_prior("normal(0, 2)", class = "sd")
# )
#
# mod_bys_accuracy_number_grade5 <- brm(
#     accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + repetition + d
#     data = train_logs_66_mdr_half_grade5_long,
#     family = bernoulli(),
#     prior = priors,
#     warmup = 500,
#     iter = 2000,
#     cores = 4)
# saveRDS(mod_bys_accuracy_number_grade5, file = "~/code_seyma/WMDevelopmentProject/mod_bys_accuracy_nu
```

```
mod_bys_accuracy_number_grade5 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod_bys_accu
summary(mod_bys_accuracy_number_grade5)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + repetiti
##    Data: train_logs_66_mdr_half_grade5_long (Number of observations: 267385)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 1234)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.59      0.02     0.56     0.63 1.00     1653     2987
##
## ~user_id (Number of levels: 1168)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.70      0.02     0.66     0.74 1.01     1309     1977
##
## Regression Coefficients:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                4.07      0.08     3.92     4.24 1.00     2313
## difficulty              -0.12      0.01    -0.15    -0.09 1.00     5340
## position_rate           -1.82      0.05    -1.92    -1.73 1.00     7159
## set_size                -0.23      0.01    -0.26    -0.20 1.00     2595
## repetitionTRUE           0.33      0.02     0.29     0.38 1.00    10586
## duplicateTRUE           -0.09      0.02    -0.13    -0.04 1.00    10210
## last_itemTRUE            1.18      0.03     1.12     1.25 1.00     7033
## last_second_itemTRUE     0.22      0.02     0.17     0.26 1.00     6832
##                      Tail_ESS
## Intercept                3162
## difficulty               4476
## position_rate            4793
## set_size                 3441
## repetitionTRUE           4588
## duplicateTRUE            5004
## last_itemTRUE            5007
## last_second_itemTRUE     5072
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Grade 6

```
# priors <- c(
#   set_prior("normal(0, 1)", class = "Intercept"),
#   set_prior("normal(0, 1)", class = "b", coef = "difficulty"),
#   set_prior("normal(0, 1.5)", class = "b", coef = "position_rate"),
#   set_prior("normal(0, 1)", class = "b", coef = "set_size"),
#   set_prior("normal(0, 1)", class = "b", coef = "repetitionTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "duplicateTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "last_itemTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "last_second_itemTRUE"),
#   set_prior("normal(0, 2)", class = "sd")
# )

# mod_bys_accuracy_number_grade6 <- brm(
#     accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + repetition + d
#     data = train_logs_66_mdr_half_grade6_long,
#     family = bernoulli(),
#     prior = priors,
#     warmup = 500,
#     iter = 2000,
#     cores = 4)
# saveRDS(mod_bys_accuracy_number_grade6, file = "~/code_seyma/WMDevelopmentProject/mod_bys_accuracy_nu
```

```
mod_bys_accuracy_number_grade6 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod_bys_accu
summary(mod_bys_accuracy_number_grade6)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + repetiti
##    Data: train_logs_66_mdr_half_grade6_long (Number of observations: 152225)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 1235)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.60      0.02     0.56     0.65 1.00     1602     3119
##
## ~user_id (Number of levels: 611)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.77      0.03     0.71     0.83 1.00     1264     2608
##
## Regression Coefficients:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                4.06      0.10     3.86     4.27 1.00     2458
## difficulty              -0.17      0.02    -0.21    -0.13 1.00     4690
## position_rate           -1.98      0.06    -2.11    -1.86 1.00     6874
## set_size                -0.16      0.02    -0.19    -0.12 1.00     3330
## repetitionTRUE           0.28      0.03     0.23     0.34 1.00     9718
## duplicateTRUE           -0.05      0.03    -0.10     0.01 1.00     7812
## last_itemTRUE            1.28      0.04     1.19     1.37 1.00     6806
## last_second_itemTRUE     0.24      0.03     0.18     0.29 1.00     7138
##                      Tail_ESS
## Intercept                4073
## difficulty               4769
## position_rate            4437
## set_size                 4331
## repetitionTRUE           4847
## duplicateTRUE            4007
## last_itemTRUE            4452
## last_second_itemTRUE     5122
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Grade 7

```
# priors <- c(
#   set_prior("normal(0, 1)", class = "Intercept"),
#   set_prior("normal(0, 1)", class = "b", coef = "difficulty"),
#   set_prior("normal(0, 1.5)", class = "b", coef = "position_rate"),
#   set_prior("normal(0, 1)", class = "b", coef = "set_size"),
#   set_prior("normal(0, 1)", class = "b", coef = "repetitionTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "duplicateTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "last_itemTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "last_second_itemTRUE"),
#   set_prior("normal(0, 2)", class = "sd")
# )
#
# mod_bys_accuracy_number_grade7 <- brm(
#     accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + repetition + d
#     data = train_logs_66_mdr_half_grade7_long,
#     family = bernoulli(),
#     prior = priors,
#     warmup = 500,
#     iter = 2000,
#     cores = 4)
# saveRDS(mod_bys_accuracy_number_grade7, file = "~/code_seyma/WMDevelopmentProject/mod_bys_accuracy_nu
```

```
mod_bys_accuracy_number_grade7<- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod_bys_accura
summary(mod_bys_accuracy_number_grade7)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + repetiti
##    Data: train_logs_66_mdr_half_grade7_long (Number of observations: 50964)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 1174)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.91      0.04     0.84     0.98 1.00     1887     3232
##
## ~user_id (Number of levels: 241)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.98      0.06     0.87     1.11 1.00     1270     2128
##
## Regression Coefficients:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                4.58      0.18     4.24     4.93 1.00     2263
## difficulty              -0.11      0.04    -0.18    -0.04 1.00     5001
## position_rate           -2.14      0.10    -2.32    -1.94 1.00     5982
## set_size                -0.24      0.03    -0.29    -0.18 1.00     2984
## repetitionTRUE           0.29      0.05     0.19     0.38 1.00    10600
## duplicateTRUE            0.14      0.05     0.04     0.24 1.00     8538
## last_itemTRUE            1.40      0.07     1.26     1.54 1.00     6219
## last_second_itemTRUE     0.26      0.05     0.16     0.35 1.00     6180
##                      Tail_ESS
## Intercept                4254
## difficulty               4544
## position_rate            5044
## set_size                 4451
## repetitionTRUE           4803
## duplicateTRUE            4600
## last_itemTRUE            5371
## last_second_itemTRUE     5037
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

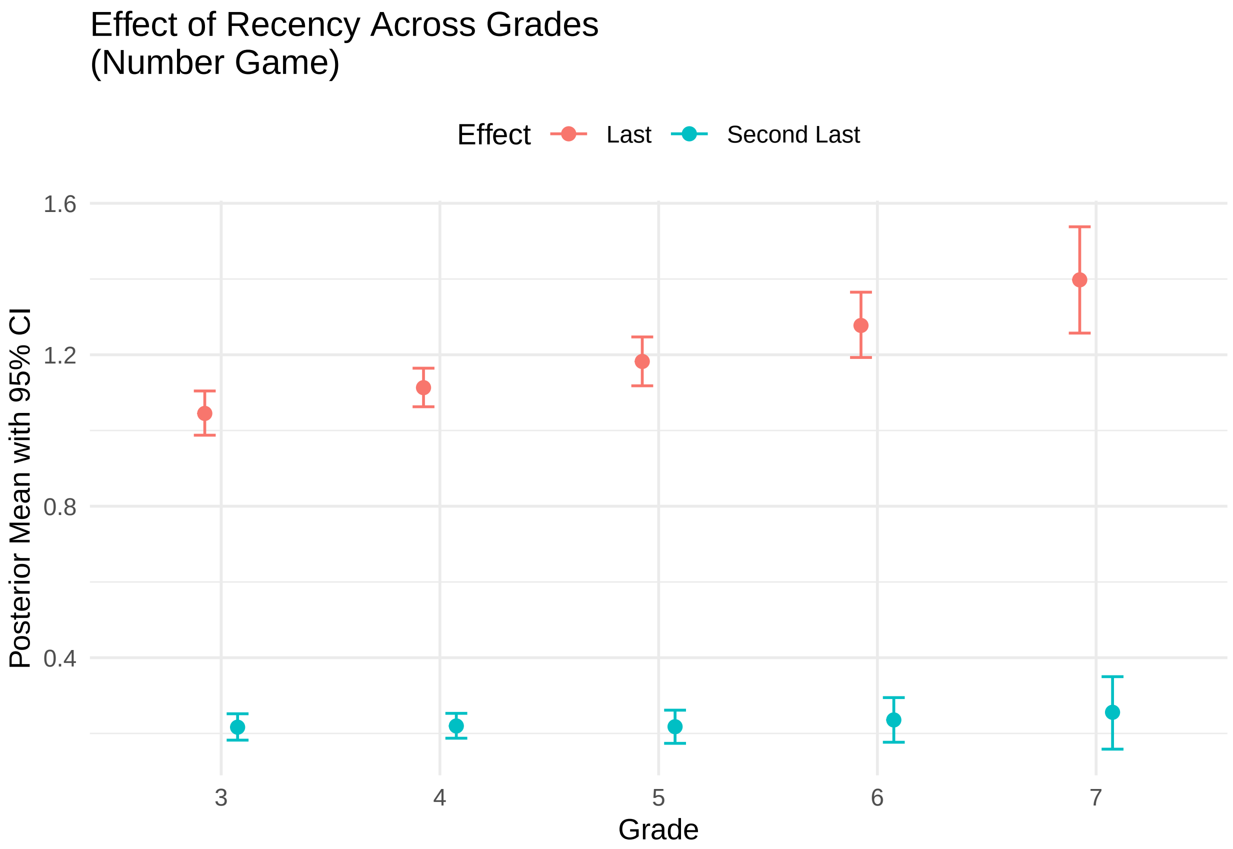
Plot

Primacy

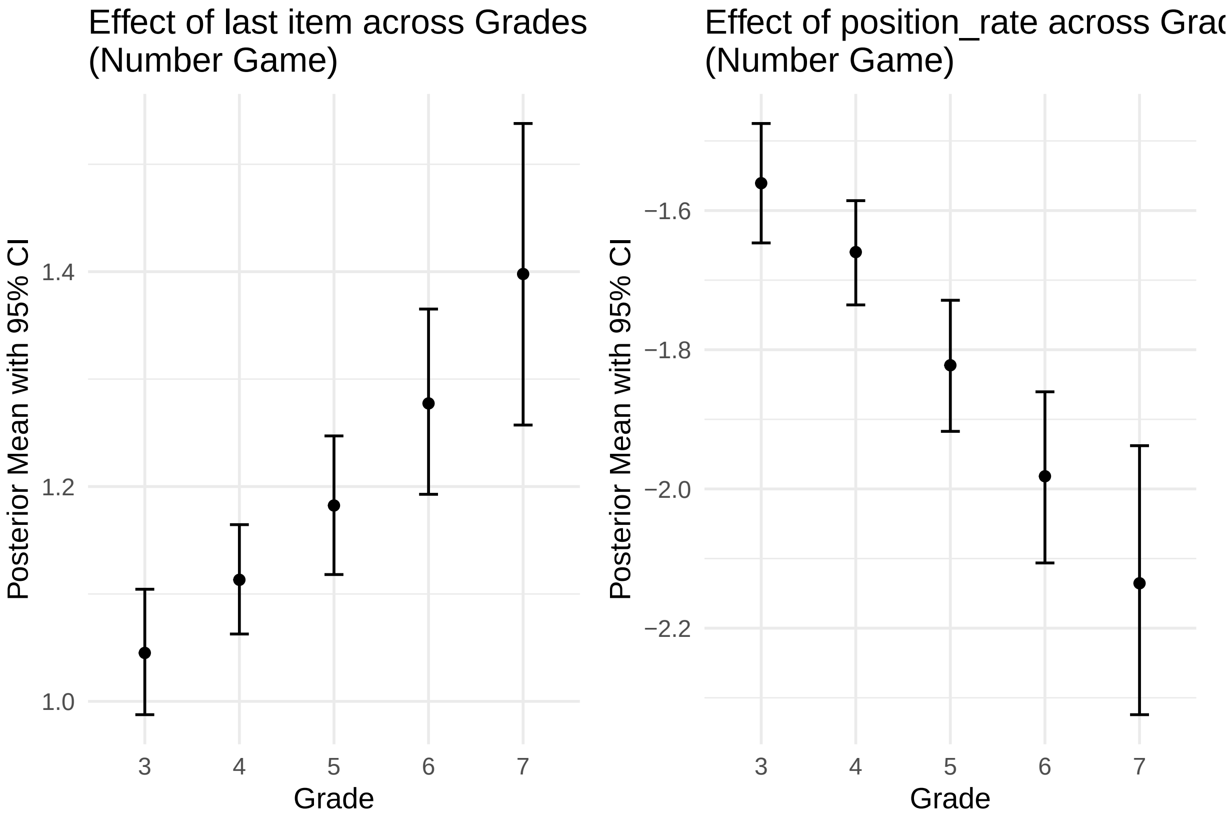
```r
grade_coefficients_number <- data.frame(
  grade = c(3, 4, 5, 6, 7),
  estimate = c(
    fixef(mod_bys_accuracy_number_grade3)["position_rate", "Estimate"],
    fixef(mod_bys_accuracy_number_grade4)["position_rate", "Estimate"],
    fixef(mod_bys_accuracy_number_grade5)["position_rate", "Estimate"],
    fixef(mod_bys_accuracy_number_grade6)["position_rate", "Estimate"],
    fixef(mod_bys_accuracy_number_grade7)["position_rate", "Estimate"]
  ),
  Q2.5 = c(
    fixef(mod_bys_accuracy_number_grade3)["position_rate", "Q2.5"],
    fixef(mod_bys_accuracy_number_grade4)["position_rate", "Q2.5"],
    fixef(mod_bys_accuracy_number_grade5)["position_rate", "Q2.5"],
    fixef(mod_bys_accuracy_number_grade6)["position_rate", "Q2.5"],
    fixef(mod_bys_accuracy_number_grade7)["position_rate", "Q2.5"]
  ),
    Q97.5 = c(
    fixef(mod_bys_accuracy_number_grade3)["position_rate", "Q97.5"],
    fixef(mod_bys_accuracy_number_grade4)["position_rate", "Q97.5"],
    fixef(mod_bys_accuracy_number_grade5)["position_rate", "Q97.5"],
    fixef(mod_bys_accuracy_number_grade6)["position_rate", "Q97.5"],
    fixef(mod_bys_accuracy_number_grade7)["position_rate", "Q97.5"]
  )
)

grade_coefficients_number_plot <- grade_coefficients_number %>%
  ggplot(aes(x = factor(grade), y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = Q2.5, ymax = Q97.5), width = 0.2) +
  labs(
    title = "Effect of position_rate across Grades\n(Number Game)",
    x = "Grade",
    y = "Posterior Mean with 95% CI"
  ) +
  theme_minimal()
```

Recency

```r
grade_coefficients_recency_number$type <- "Last"
grade_coefficients_recency2_number$type <- "Second Last"

grade_coefficients_combined_recency <- bind_rows(
  grade_coefficients_recency_number,
  grade_coefficients_recency2_number
)

# Plot
ggplot(grade_coefficients_combined_recency,
       aes(x = factor(grade), y = estimate, color = type)) +
  geom_point(position = position_dodge(width = 0.3), size = 2) +
  geom_errorbar(aes(ymin = Q2.5, ymax = Q97.5),
                position = position_dodge(width = 0.3), width = 0.2) +
  labs(
    title = "Effect of Recency Across Grades\n(Number Game)",
    x = "Grade",
    y = "Posterior Mean with 95% CI",
    color = "Effect"
  ) +
  theme_minimal() +
  theme(
    legend.position = "top"
  )
```



```
grade_coefficients_recency_number_plot + grade_coefficients_number_plot
```


Effect of last item across Grades (Number Game)

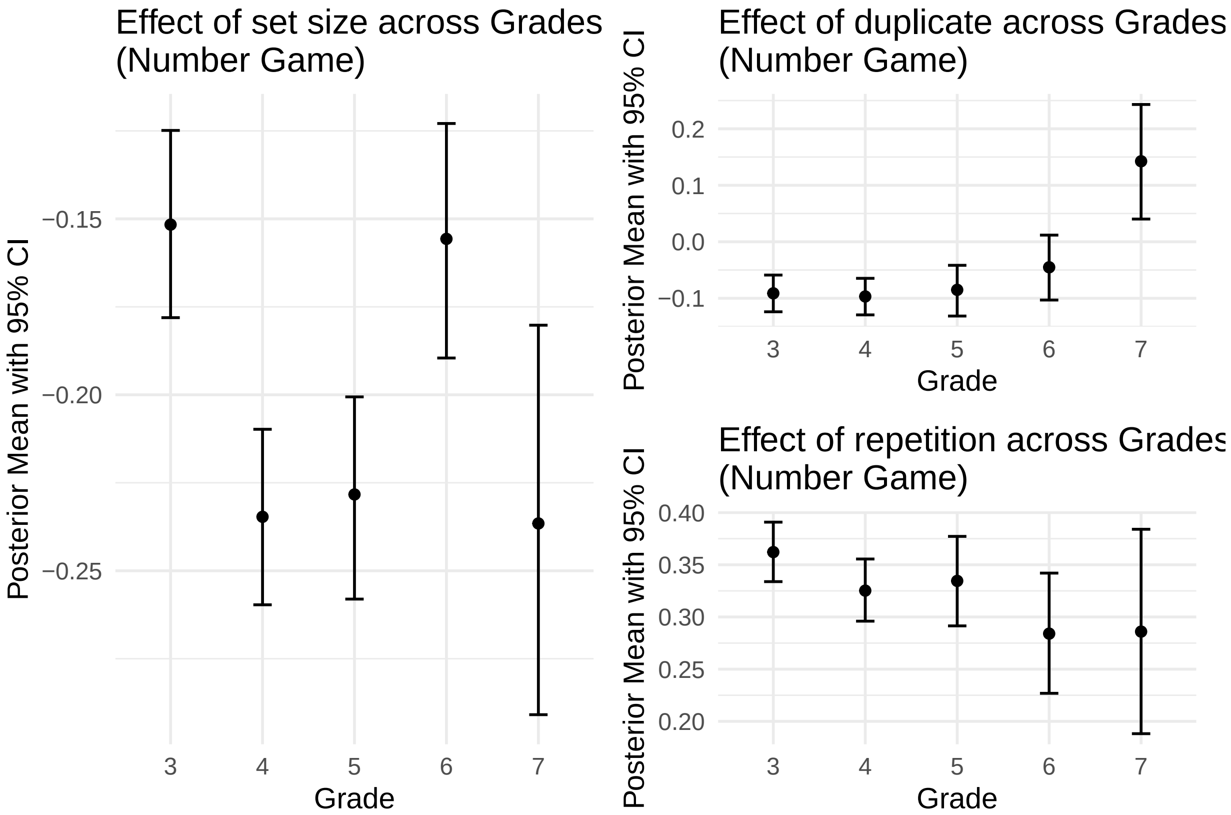
Effect of position_rate across Grac (Number Game)

Set Size

Same process for other predictors.

Repetition and duplicate

```
grade_coefficients_setsize_number_plot + grade_coefficients_duplicate_number_plot / grade_coefficients_
```



Similar Ability Student with different grades

In the frequentist ability we applied lenient and strict data selections with different results. I will start with strict data selection and compare it with the general results.

```
train_logs_66_mdr_half_last_ability <- train_logs_66_mdr_half %>%
  group_by(user_id, grade) %>%
  filter(created == max(created)) %>%
  ungroup()

similar_students_strict <- train_logs_66_mdr_half_last_ability %>%
  filter(grade %in% c(4, 6),
         new_user_domain_rating > 1,
         new_user_domain_rating < 2)
similar_students_strict %>% count(grade)
```

```
## # A tibble: 2 x 2
##   grade     n
##   <dbl> <int>
## 1     4   213
## 2     6    89
```

```
similar_students_strict <- similar_students_strict %>%
  mutate(user_grade = paste0(user_id, "_", grade)) %>%
  pull(user_grade)

train_logs_66_mdr_half_grade46_similarAbility_strict <- train_logs_66_mdr_half %>%
  mutate(user_grade_logs = paste0(user_id, "_", grade)) %>%
  filter(user_grade_logs %in% similar_students_strict)

train_logs_66_mdr_half_grade4_similarAbility_strict <- train_logs_66_mdr_half_grade46_similarAbility_sti
  filter(grade == 4)

train_logs_66_mdr_half_grade6_similarAbility_strict <- train_logs_66_mdr_half_grade46_similarAbility_sti
  filter(grade == 6)
```

```
capacity4 <- train_logs_66_mdr_half_grade46_similarAbility_strict %>%
  filter(grade == 4) %>%
  group_by(user_id) %>%
  summarise(capacity = max(set_size))
capacity6 <- train_logs_66_mdr_half_grade46_similarAbility_strict %>%
  filter(grade == 6) %>%
  group_by(user_id) %>%
  summarise(capacity = max(set_size))

prop.table(table(capacity4$capacity))
```

```
##
##          5          6          7          8
## 0.03286385 0.16431925 0.36619718 0.43661972
```

```
prop.table(table(capacity6$capacity))
```

```
##
##          5          6          7          8
## 0.01123596 0.21348315 0.33707865 0.43820225
```

In the strict selection, max set size of users are similar.

```


# priors <- c(
#   set_prior("normal(0, 1)", class = "Intercept"),
#   set_prior("normal(0, 1)", class = "b", coef = "difficulty"),
#   set_prior("normal(0, 1.5)", class = "b", coef = "position_rate"),
#   set_prior("normal(0, 1)", class = "b", coef = "set_size"),
#   set_prior("normal(0, 1)", class = "b", coef = "repetitionTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "duplicateTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "last_itemTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "last_second_itemTRUE"),
#   set_prior("normal(0, 2)", class = "sd")
# )

# mod_accuracy_number_grade4_similarAbility_strict_bys <- brm(
#     accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + repetition + d
#     data = train_logs_66_mdr_half_grade4_similarAbility_strict_long,
#     family = bernoulli(),
#     prior = priors,
#     warmup = 500,
#     iter = 2000,
#     cores = 4)
# saveRDS(mod_accuracy_number_grade4_similarAbility_strict_bys, file = "~/code_seyma/WMDevelopmentProje
```

```
mod_accuracy_number_grade4_similarAbility_strict_bys <- readRDS(file = "~/code_seyma/WMDevelopmentProje
summary(mod_accuracy_number_grade4_similarAbility_strict_bys)
```

Grade 4

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + repetiti
##    Data: train_logs_66_mdr_half_grade4_similarAbility_stric (Number of observations: 69537)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 866)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.72      0.03     0.66     0.79 1.00     1490     2477
##
## ~user_id (Number of levels: 213)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.46      0.03     0.40     0.53 1.00     1517     2593
##
## Regression Coefficients:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                4.48      0.15     4.19     4.78 1.00     3431
## difficulty              -0.13      0.03    -0.19    -0.08 1.00     4786
## position_rate           -2.16      0.10    -2.35    -1.96 1.00     5647
## set_size                -0.23      0.03    -0.28    -0.17 1.00     3336
## repetitionTRUE           0.35      0.04     0.27     0.44 1.00     8143
## duplicateTRUE           -0.03      0.04    -0.12     0.06 1.00     7788
## last_itemTRUE            1.29      0.07     1.16     1.42 1.00     5150
## last_second_itemTRUE     0.26      0.05     0.17     0.35 1.00     5202
##                      Tail_ESS
## Intercept                3685
## difficulty               4340
## position_rate            4271
## set_size                 3836
## repetitionTRUE           4718
## duplicateTRUE            4561
## last_itemTRUE            4370
## last_second_itemTRUE     4481
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
# priors <- c(
#   set_prior("normal(0, 1)", class = "Intercept"),
#   set_prior("normal(0, 1)", class = "b", coef = "difficulty"),
#   set_prior("normal(0, 1.5)", class = "b", coef = "position_rate"),
#   set_prior("normal(0, 1)", class = "b", coef = "set_size"),
#   set_prior("normal(0, 1)", class = "b", coef = "repetitionTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "duplicateTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "last_itemTRUE"),
#   set_prior("normal(0, 1)", class = "b", coef = "last_second_itemTRUE"),
#   set_prior("normal(0, 2)", class = "sd")
# )

# mod_accuracy_number_grade6_similarAbility_strict_bys <- brm(
#     accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + repetition + d
#     data = train_logs_66_mdr_half_grade6_similarAbility_strict_long,
#     family = bernoulli(),
#     prior = priors,
#     warmup = 500,
#     iter = 2000,
#     cores = 4)
# saveRDS(mod_accuracy_number_grade6_similarAbility_strict_bys, file = "~/code_seyma/WMDevelopmentProje
```

```r
mod_accuracy_number_grade6_similarAbility_strict_bys <- readRDS(file = "~/code_seyma/WMDevelopmentProje
mod_accuracy_number_grade6_similarAbility_strict_bys
```

Grade 6

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate + set_size + repetiti
##    Data: train_logs_66_mdr_half_grade6_similarAbility_stric (Number of observations: 24780)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 677)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.96      0.05     0.86     1.07 1.00     1442     2724
##
## ~user_id (Number of levels: 89)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.47      0.05     0.39     0.58 1.00     1834     3379
##
## Regression Coefficients:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                3.87      0.25     3.39     4.39 1.00     2229
## difficulty              -0.22      0.05    -0.31    -0.13 1.00     4278
## position_rate           -1.73      0.16    -2.04    -1.42 1.00     3346
## set_size                -0.12      0.05    -0.21    -0.03 1.00     2443
## repetitionTRUE           0.41      0.07     0.26     0.55 1.00     6575
## duplicateTRUE           -0.05      0.07    -0.19     0.09 1.00     6718
## last_itemTRUE            1.17      0.11     0.96     1.38 1.00     3433
## last_second_itemTRUE     0.16      0.08     0.01     0.31 1.00     3618
##                      Tail_ESS
## Intercept                3147
## difficulty               4402
## position_rate            3623
## set_size                 3372
## repetitionTRUE           4592
## duplicateTRUE            4894
## last_itemTRUE            3527
## last_second_itemTRUE     3953
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Plots

```
capacity4number <- train_logs_66_mdr_half_grade46_similarAbility_strict %>%
  filter(grade == 4) %>%
  group_by(user_id) %>%
  summarise(capacity = max(set_size))
capacity6number <- train_logs_66_mdr_half_grade46_similarAbility_strict %>%
  filter(grade == 6) %>%
  group_by(user_id) %>%
  summarise(capacity = max(set_size))

prop.table(table(capacity4number$capacity))
```

```
##
##          5          6          7          8
## 0.03286385 0.16431925 0.36619718 0.43661972
```

```
prop.table(table(capacity6number$capacity))
```

```
##
##          5          6          7          8
## 0.01123596 0.21348315 0.33707865 0.43820225
```

```
#Set size 7 and 8 are the most frequent ones for both grades.

tbl4number <- prop.table(table(capacity4number$capacity)) * 100
percent_grade4number <- paste(names(tbl4number), "=", round(tbl4number, 2), "%", collapse = ", ")


tbl6number <- prop.table(table(capacity6number$capacity)) * 100
percent_grade6number <- paste(names(tbl6number), "=", round(tbl6number, 2), "%", collapse = ", ")
```

```
capacity4numberAll <- train_logs_66_mdr_half_grade4 %>%
  group_by(user_id) %>%
  summarise(capacity = max(set_size))
capacity6numberAll <- train_logs_66_mdr_half_grade6 %>%
  group_by(user_id) %>%
  summarise(capacity = max(set_size))

prop.table(table(capacity4numberAll$capacity))
```

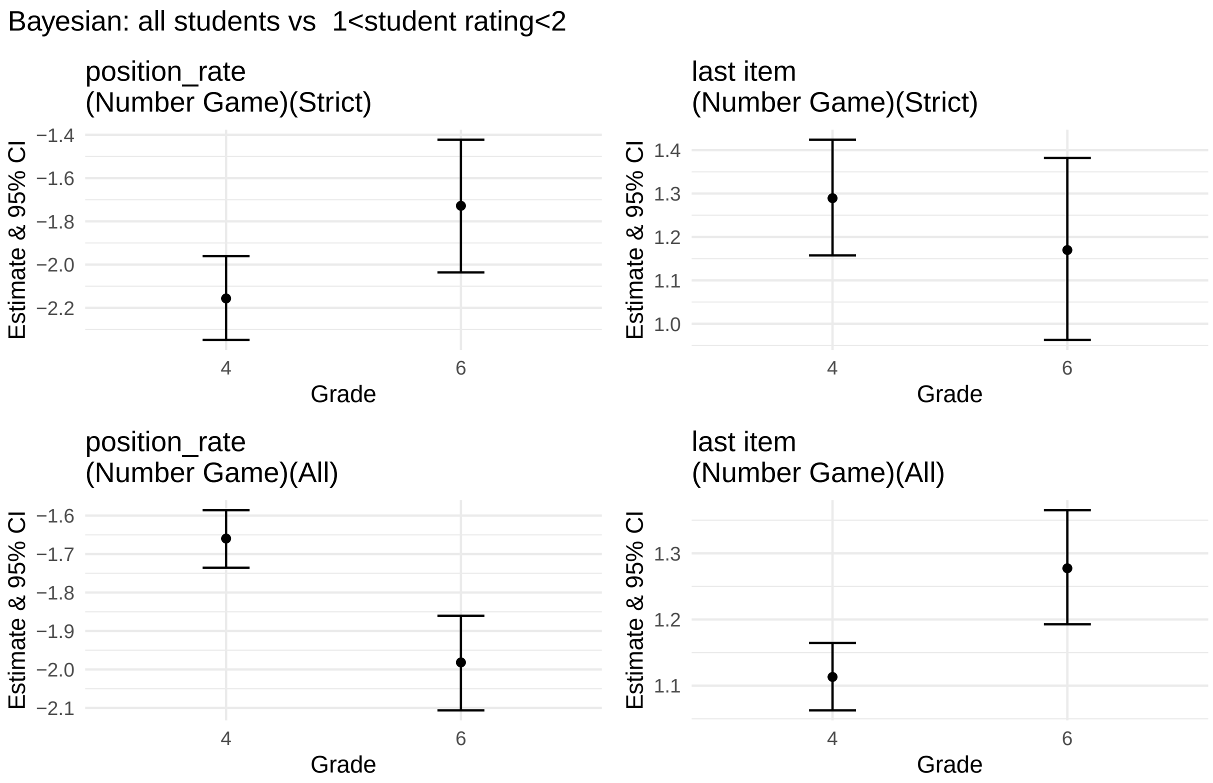
```
##
##          3          4          5          6          7          8
## 0.01590714 0.09114359 0.23602752 0.23258813 0.27085125 0.15348237
```

```
prop.table(table(capacity6numberAll$capacity))
```

```
##
##           4           5           6           7           8
## 0.006546645 0.068739771 0.178396072 0.175122750 0.571194763
```

```
tbl4numberALL <- prop.table(table(capacity4numberAll$capacity)) * 100
percent_grade4ALLnumber <- paste(names(tbl4numberALL), "=", round(tbl4numberALL, 2), "%", collapse = ",


tbl6numberALL <- prop.table(table(capacity6numberAll$capacity)) * 100
percent_grade6ALLnumber <- paste(names(tbl6numberALL), "=", round(tbl6numberALL, 2), "%", collapse = ",
```

```
(grade46_strict_primacy_number_plot + grade46_strict_recency_number_plot)/ (grade46_primacy_number_all_
  plot_annotation(title =  'Bayesian: all students vs  1<student rating<2',
    caption = paste("Strict Students Max Setsize Percentages:\nGrade 4:", percent_grade4number, "\nGrad
  )
```



Strict Students Max Setsize Percentages:
Grade 4: 5 = 3.29 %, 6 = 16.43 %, 7 = 36.62 %, 8 = 43.66 %
Grade 6: 4 = 0.65 %, 5 = 6.87 %, 6 = 17.84 %, 7 = 17.51 %, 8 = 57.12 %
All data Max Setsize:
Grade 4: 3 = 1.59 %, 4 = 9.11 %, 5 = 23.6 %, 6 = 23.26 %, 7 = 27.09 %, 8 = 15.35 %
Grade 6: 4 = 0.65 %, 5 = 6.87 %, 6 = 17.84 %, 7 = 17.51 %, 8 = 57.12 %

SetSize Interaction

Go to beginning of the number game section for the data for grade levels.

- The grade 3 model does not converge.

```
# mod1_bys_accuracy_number_grade4 <- brm(
#     accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + repetition + d
#     data = train_logs_66_mdr_half_grade4_long,
#     family = bernoulli(),
#     prior = priors,
#     warmup = 500,
#     iter = 2000,
#     cores = 4)
# saveRDS(mod1_bys_accuracy_number_grade4, file = "~/code_seyma/WMDevelopmentProject/mod1_bys_accuracy_
```

```
mod1_bys_accuracy_number_grade4 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod1_bys_acc
summary(mod1_bys_accuracy_number_grade4)
```

Grade 4

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + repetiti
##    Data: train_logs_66_mdr_half_grade4_long (Number of observations: 553223)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 1192)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.52      0.02     0.49     0.56 1.00     1072     2109
##
## ~user_id (Number of levels: 2326)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.66      0.01     0.64     0.69 1.00     1179     2232
##
## Regression Coefficients:
##                         Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                   2.96      0.09     2.80     3.14 1.00     1810
## difficulty                 -0.15      0.01    -0.17    -0.13 1.00     4786
## position_rate               0.15      0.13    -0.11     0.39 1.00     3749
## set_size                   -0.07      0.02    -0.10    -0.03 1.00     2195
## repetitionTRUE              0.31      0.02     0.28     0.34 1.00    10648
## duplicateTRUE              -0.06      0.02    -0.09    -0.03 1.00     9329
## last_itemTRUE               0.28      0.09     0.11     0.45 1.00     3924
## last_second_itemTRUE        0.13      0.02     0.09     0.16 1.00     6741
## position_rate:set_size     -0.34      0.02    -0.39    -0.30 1.00     3994
## set_size:last_itemTRUE      0.15      0.02     0.11     0.18 1.00     4136
##                         Tail_ESS
## Intercept                   3178
## difficulty                  4608
## position_rate               4080
## set_size                    3328
## repetitionTRUE              5148
## duplicateTRUE               4979
## last_itemTRUE               4662
## last_second_itemTRUE        4570
## position_rate:set_size      4371
## set_size:last_itemTRUE      4501
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# mod1_bys_accuracy_number_grade5 <- brm(
#     accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + repetition + d
#     data = train_logs_66_mdr_half_grade5_long,
#     family = bernoulli(),
#     prior = priors,
#     warmup = 500,
#     iter = 2000,
#     cores = 4)
# saveRDS(mod1_bys_accuracy_number_grade5, file = "~/code_seyma/WMDevelopmentProject/mod1_bys_accuracy_
```

```
mod1_bys_accuracy_number_grade5 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod1_bys_ac
```

```
summary(mod1_bys_accuracy_number_grade5)
```

Grade 5

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + repetiti
##    Data: train_logs_66_mdr_half_grade5_long (Number of observations: 267385)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 1234)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.60      0.02     0.56     0.63 1.00     1478     3000
##
## ~user_id (Number of levels: 1168)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.70      0.02     0.66     0.75 1.00     1213     2108
##
## Regression Coefficients:
##                        Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                  3.03      0.11     2.81     3.25 1.00     3097
## difficulty                -0.12      0.01    -0.14    -0.09 1.00     5332
## position_rate              0.28      0.18    -0.07     0.63 1.00     4172
## set_size                  -0.04      0.02    -0.08    -0.00 1.00     3518
## repetitionTRUE             0.32      0.02     0.28     0.36 1.00    11057
## duplicateTRUE             -0.04      0.02    -0.09     0.00 1.00     9521
## last_itemTRUE              0.48      0.13     0.23     0.74 1.00     4327
## last_second_itemTRUE       0.12      0.02     0.07     0.17 1.00     7987
## position_rate:set_size    -0.37      0.03    -0.43    -0.31 1.00     4387
## set_size:last_itemTRUE     0.10      0.02     0.05     0.15 1.00     4762
##                        Tail_ESS
## Intercept                  4218
## difficulty                 4902
## position_rate              4345
## set_size                   4811
## repetitionTRUE             4952
## duplicateTRUE              4886
## last_itemTRUE              4666
## last_second_itemTRUE       5160
## position_rate:set_size     4632
## set_size:last_itemTRUE     5100
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# mod1_bys_accuracy_number_grade6 <- brm(
#     accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + repetition + d
#     data = train_logs_66_mdr_half_grade6_long,
#     family = bernoulli(),
#     prior = priors,
#     warmup = 500,
#     iter = 2000,
#     cores = 4)
# saveRDS(mod1_bys_accuracy_number_grade6, file = "~/code_seyma/WMDevelopmentProject/mod1_bys_accuracy_
```

```
mod1_bys_accuracy_number_grade6 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod1_bys_ac
summary(mod1_bys_accuracy_number_grade6)
```

Grade 6

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + repetiti
##    Data: train_logs_66_mdr_half_grade6_long (Number of observations: 152225)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 1235)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.61      0.02     0.57     0.65 1.00     1779     2932
##
## ~user_id (Number of levels: 611)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.77      0.03     0.71     0.83 1.01     1038     2298
##
## Regression Coefficients:
##                        Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                  3.24      0.15     2.95     3.53 1.00     3322
## difficulty                -0.17      0.02    -0.21    -0.13 1.00     4620
## position_rate             -0.30      0.23    -0.75     0.16 1.00     4172
## set_size                  -0.02      0.02    -0.07     0.03 1.00     3846
## repetitionTRUE             0.28      0.03     0.22     0.34 1.00    12558
## duplicateTRUE             -0.02      0.03    -0.08     0.04 1.00    10200
## last_itemTRUE              0.55      0.18     0.19     0.89 1.00     4362
## last_second_itemTRUE       0.15      0.03     0.09     0.22 1.00     8856
## position_rate:set_size    -0.28      0.04    -0.35    -0.21 1.00     4323
## set_size:last_itemTRUE     0.11      0.03     0.05     0.18 1.00     4934
##                         Tail_ESS
## Intercept                   4536
## difficulty                  4895
## position_rate               4055
## set_size                    4541
## repetitionTRUE              4093
## duplicateTRUE               5392
## last_itemTRUE               4382
## last_second_itemTRUE        4549
## position_rate:set_size      4098
## set_size:last_itemTRUE      4543
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

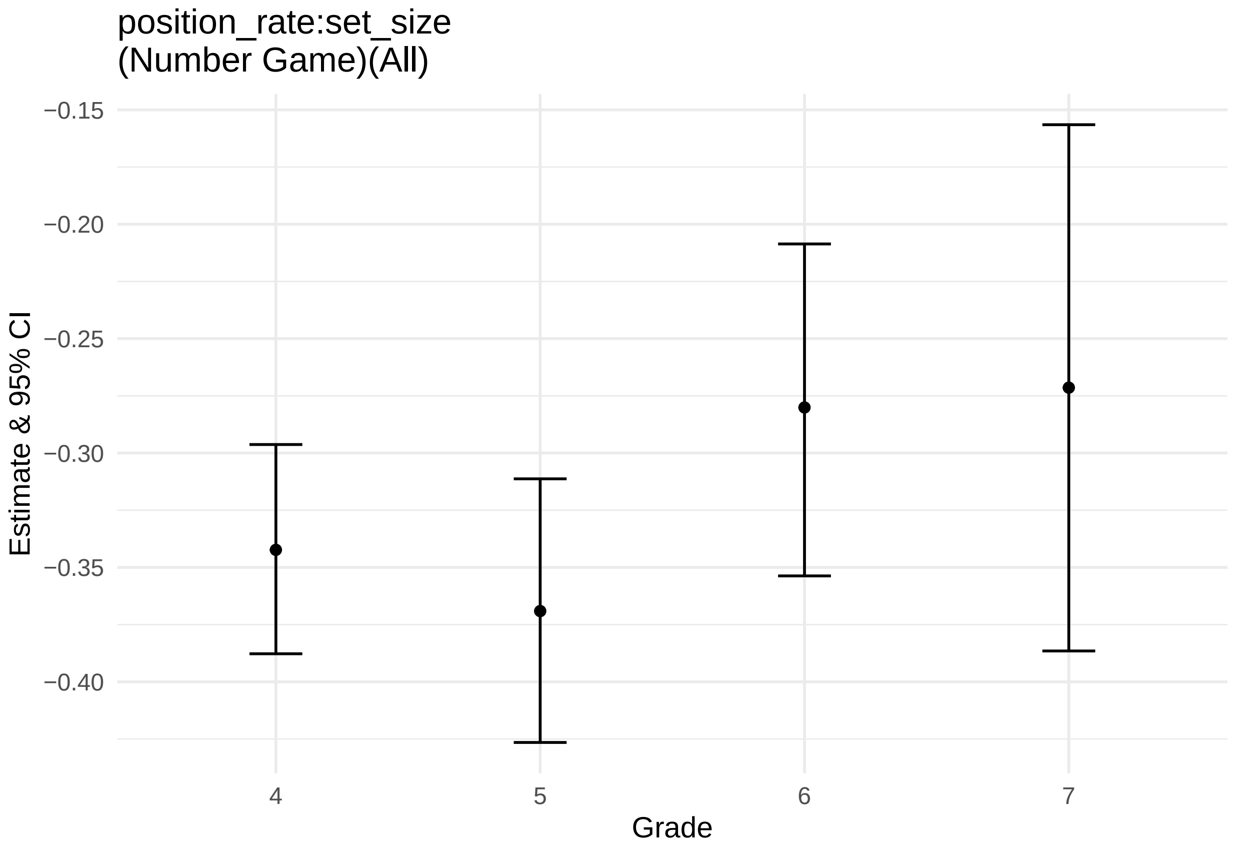
```
# mod1_bys_accuracy_number_grade7 <- brm(
#     accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + repetition + d
#     data = train_logs_66_mdr_half_grade7_long,
#     family = bernoulli(),
#     prior = priors,
#     warmup = 500,
#     iter = 2000,
#     cores = 4)
# saveRDS(mod1_bys_accuracy_number_grade7, file = "~/code_seyma/WMDevelopmentProject/mod1_bys_accuracy_
```

```
mod1_bys_accuracy_number_grade7 <- readRDS(file = "~/code_seyma/WMDevelopmentProject/models/mod1_bys_acc
mod1_bys_accuracy_number_grade7
```

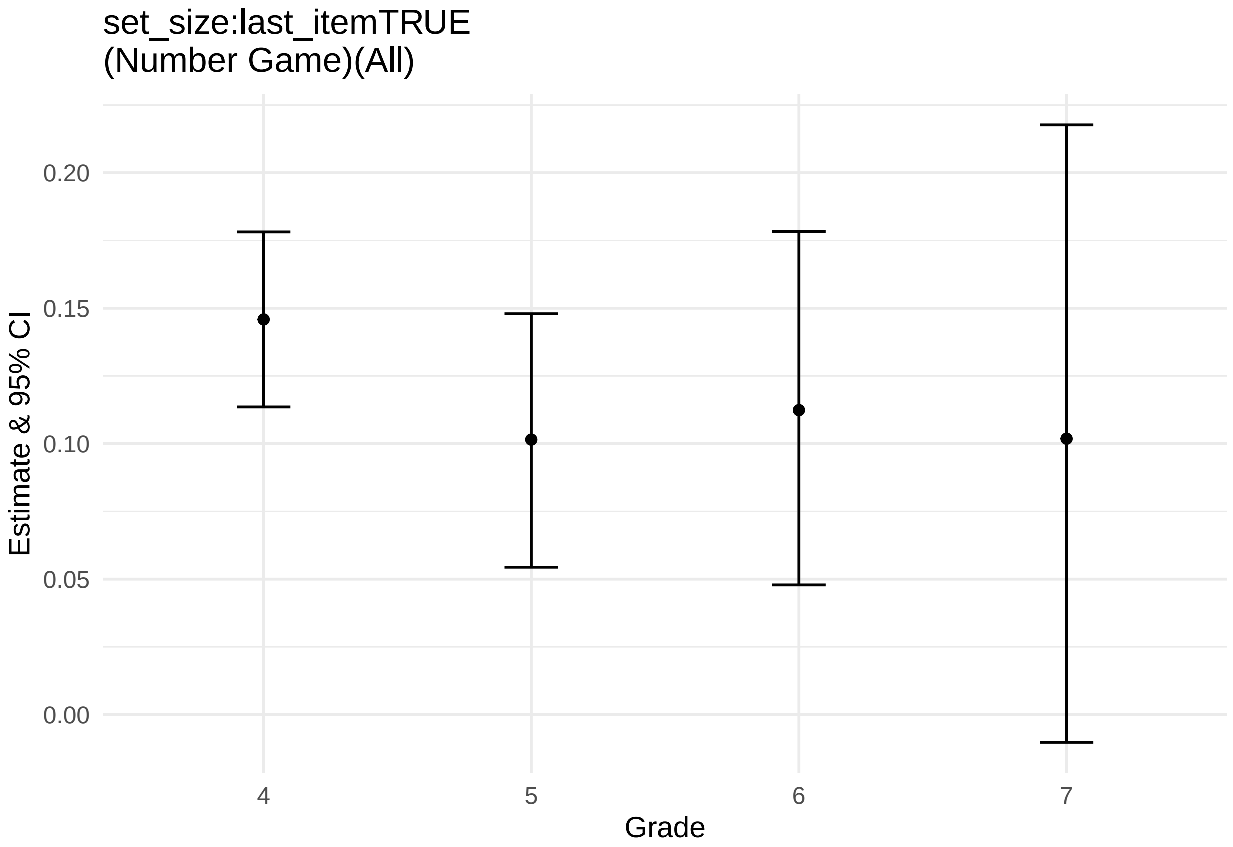
Grade 7

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: accuracy ~ (1 | user_id) + (1 | item_id) + difficulty + position_rate * set_size + repetiti
##    Data: train_logs_66_mdr_half_grade7_long (Number of observations: 50964)
##   Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##          total post-warmup draws = 6000
##
## Multilevel Hyperparameters:
## ~item_id (Number of levels: 1174)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.91      0.04     0.84     0.98 1.00     1774     3002
##
## ~user_id (Number of levels: 241)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.98      0.06     0.86     1.10 1.01     1057     1868
##
## Regression Coefficients:
##                         Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                     3.76      0.25     3.29     4.26 1.00     2215
## difficulty                   -0.11      0.04    -0.18    -0.04 1.00     3290
## position_rate                -0.44      0.38    -1.19     0.31 1.00     3350
## set_size                     -0.10      0.04    -0.18    -0.03 1.00     2881
## repetitionTRUE                0.28      0.05     0.19     0.38 1.00     8332
## duplicateTRUE                 0.17      0.05     0.07     0.27 1.00     6921
## last_itemTRUE                 0.70      0.32     0.06     1.32 1.00     3339
## last_second_itemTRUE          0.17      0.06     0.06     0.28 1.00     6380
## position_rate:set_size       -0.27      0.06    -0.39    -0.16 1.00     3446
## set_size:last_itemTRUE        0.10      0.06    -0.01     0.22 1.00     3598
##                         Tail_ESS
## Intercept                   3719
## difficulty                  4230
## position_rate               3722
## set_size                    3991
## repetitionTRUE              5156
## duplicateTRUE               4841
## last_itemTRUE               3661
## last_second_itemTRUE        4492
## position_rate:set_size      3816
## set_size:last_itemTRUE      3872
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
setsize_int_primacy_plot <- setsize_int_primacy %>%
  ggplot(aes(x = factor(grade), y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = Q2.5, ymax = Q97.5), width = 0.2) +
  labs(
    title = "position_rate:set_size \n(Number Game)(All)",
    x = "Grade",
    y = "Estimate & 95% CI"
  ) +
  theme_minimal()
setsize_int_primacy_plot
```



```r
#Plots

setsize_int_recency_plot <- setsize_int_recency %>%
  ggplot(aes(x = factor(grade), y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = Q2.5, ymax = Q97.5), width = 0.2) +
  labs(
    title = "set_size:last_itemTRUE \n(Number Game)(All)",
    x = "Grade",
    y = "Estimate & 95% CI"
  ) +
  theme_minimal()
setsize_int_recency_plot
```



Summary Number Game

1. **Models by grade**
   1. *Primacy*
      1. Position rate decrease (primact effect *increase*) from grade 3 to 7
   2. *Recency*
      1. Recent effect *increase* with grade level
   3. *Set Size*
      1. The effect between grades are *non-linear* and *varies*
   4. *Duplicate*
      1. The effect do not vary and is lost for grades 6 and 7.

   5. *Repetition*
      1. Wider uncertainty for later grades so no changes.
2. **Students with similar Ability (Comparing grade 4 and 6)**
   1. When comparing all students, we see that primacy and recency effects are bigger for grade 6 (opposite of the number game)
   2. Same ability
      1. Primacy: Bigger for grade 4
      2. Recency: CI is big for grade 6 so not much difference (meaning positive towards grade 4)
3. **Interaction models**

1. *The interaction models do not fit better*. They are too complicated.
2. Set Size Interaction
    1. For grades 4, 5, the interaction loses the main effect of position_effect (other way around mole game). For grade 6, set_size main effect is also lost.
    2. The interaction of set size with the last item and position rate does not change much as CI gets bigger with grade level

To sum up, primacy and recency increase. CI grows with grade levels for other predictors. With the same ability students, the effect reverts. The interactions do not explain much.