JunjieZeng_STATS485_Unit3_Paper_V1

Junjie Zeng

2025-04-08

Contents

1 Overview

This file produces outcomes in paper Prognostic Modeling with Texas Education Data. This paper is based on the URPS project in Winter 2025 semester supervised by Prof. Ben Hansen and PhD student Julian Bernado. We used TEA_2019.csv data to model Texas students' math and reading test scores in grade 3-8 in 2019. Caroline Moy collaborated with me on this project. She was in charge of the modeling for grade 6-8 reading scores and grade 3-5 math scores. Since our paper depends on long-running computations, we will show the modeling process only for grade 4 reading scores as the representatives using 30 percent schools in this file. Other models can be built using basically the same way. This file is divided into several parts:

1. Data preprocessing
2. Identify Potential important variable

3. Build simple models
4. Select the best simple model
5. Normalization
6. Splines
7. Best Model for Grade 4 Reading
8. Extremely Low Math Scores

2 Part 0: Data Loading

Reproduciability

```
set.seed(489)
```

We load the data and necessary packages.

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lme4)
```

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

```
library(purrr)
library(ggplot2)
library(stringr)
library(splines)
data <- read_csv("/home/rstudio/TEA_2019.csv")
```

```
## Rows: 2506956 Columns: 91
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): replacement_id
## dbl (64): acadyear, districtid_nces_enroll_m1, districtid_nces_enroll_p0, sc...
## lgl (26): frl_3yr, frl_5yr, frl_high, lep_3yr, lep_5yr, lep_high, rfep_now, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

3 Grade 4 Reading Score Modeling

4 Part 1: Data preprocessing

We clean our data follow the following steps: 1. We randomly sample 30 percent of schools to run faster. 2. Remove all variables with only NA values. Those variables contains no information. 3. Since we don't want to use present math score to model present reading score, we take out all math score variables. We also take out alternative test scores, they are not being modeled for now. We also take out all variables ends with `_midd` because we are modeling primary school students. `replacement_id` and `acadyear` are also irrelevant. 4. We filter grade 4 students data and remove NA's. 5. We add `age_int` variable, which is the rounded version of `age`. We will treat age as categorical variable rather than continuous variable. Similarly for `attend_p0` and `attend_m1`, percentage of attendance present year/last year.

```
unique_schools <- unique(data$schoolid_nces_enroll_p0)
schools_sampled <- sample(unique_schools,
                          size = round(0.3 * length(unique_schools)))
df <- data %>%
  filter(schoolid_nces_enroll_p0 %in% schools_sampled) %>%
  # Remove variables with only NA's
  select(where(~ !all(is.na(.)))) %>%
  select(-c('glmath_ver_p0',
            'glmath_lan_p0',
            'glmath_scr_p0',
            'glmath_alt_scr_m1', 'glmath_alt_scr_p0',
            'readng_alt_scr_m1', 'readng_alt_scr_p0',
            'replacement_id', 'acadyear'),-ends_with("_midd")) %>%
  # Remove those didn't have exam scores
  filter(!is.na(readng_scr_p0), !is.na(readng_scr_m1),
         !is.na(glmath_scr_m1), gradelevel == 4) %>%
  mutate(age_int = round(age),
         attend_p0_d1 = round(attend_p0, 1),
         attend_m1_d1 = round(attend_m1, 1))
```

5 Part 2: Identify Potential important variable

We select categorical variables with number of categories less than 10 and create summary values for them to see whether there's a relatively big difference between categories.

```
vars <- names(df)[sapply(df, function(x) length(unique(x)) < 10)]
get_summary <- function(var){
  df %>%
  group_by(!!sym(var)) %>%
  summarize(mean(readng_scr_p0),
            median(readng_scr_p0),
            count = n(),
            proportion = n()/nrow(df))
}
summary_list <- map(vars, get_summary)
summary_list
```

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

```
## # A tibble: 6 x 5
##   enrfay_school `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##           <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1         0.167                 1476.                    1471    928    0.00821
## 2         0.333                 1464.                    1456   1149    0.0102
## 3         0.5                   1460.                    1456   1430    0.0126
## 4         0.667                 1461.                    1456   1086    0.00961
## 5         0.833                 1451.                    1441    881    0.00779
## 6         1                     1522.                    1519 107589    0.952
##
## [[2]]
## # A tibble: 6 x 5
##   enrfay_district `mean(readng_scr_p0)` median(readng_scr_p0~1  count proportion
##             <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1           0.167                 1478.                    1487    847    0.00749
## 2           0.333                 1465.                    1471    788    0.00697
## 3           0.5                   1465.                    1471    983    0.00869
## 4           0.667                 1461.                    1456    774    0.00685
## 5           0.833                 1457.                    1456    722    0.00639
## 6           1                     1521.                    1519 108949    0.964
## # i abbreviated name: 1: `median(readng_scr_p0)`
##
## [[3]]
## # A tibble: 6 x 5
##   enrfay_state `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##          <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1        0.167                 1466.                    1402      5  0.0000442
## 2        0.333                 1440.                    1470     11  0.0000973
## 3        0.5                   1497.                    1519     23  0.000203
## 4        0.667                 1468.                    1487     49  0.000433
## 5        0.833                 1451.                    1456    103  0.000911
## 6        1                     1519.                    1519 112872  0.998
##
## [[4]]
## # A tibble: 1 x 5
##   gradelevel `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##        <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1          4                 1519.                    1519 113063          1
##
## [[5]]
## # A tibble: 2 x 5
##   gender `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##    <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1      1                 1530.                    1536 55781      0.493
## 2      2                 1508.                    1502 57282      0.507
##
## [[6]]
## # A tibble: 7 x 5
##   raceth `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##    <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1      1                 1499.                    1502   455    0.00402
## 2      2                 1623.                    1619  5162    0.0457
## 3      3                 1469.                    1456 14056    0.124
## 4      4                 1500.                    1502 59308    0.525
```

```
## 5      5                 1523.                     1519   137    0.00121
## 6      6                 1560.                     1574 31159    0.276
## 7     NA                 1542.                     1550  2786    0.0246
##
## [[7]]
## # A tibble: 2 x 5
##   frl_now `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##     <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1       0                 1559.                    1550 54764      0.484
## 2       1                 1482.                    1487 58299      0.516
##
## [[8]]
## # A tibble: 2 x 5
##   frl_2yr `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##     <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1       0                 1569.                    1574 46070      0.407
## 2       1                 1485.                    1487 66993      0.593
##
## [[9]]
## # A tibble: 2 x 5
##   frl_ever `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##      <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1        0                 1569.                    1574 46070      0.407
## 2        1                 1485.                    1487 66993      0.593
##
## [[10]]
## # A tibble: 2 x 5
##   frl_elem `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##      <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1        0                 1569.                    1574 46070      0.407
## 2        1                 1485.                    1487 66993      0.593
##
## [[11]]
## # A tibble: 2 x 5
##   lep_now `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##     <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1       0                 1533.                    1536 86344      0.764
## 2       1                 1474.                    1471 26719      0.236
##
## [[12]]
## # A tibble: 2 x 5
##   lep_2yr `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##     <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1       0                 1531.                    1536 84225      0.745
## 2       1                 1483.                    1487 28838      0.255
##
## [[13]]
## # A tibble: 2 x 5
##   lep_ever `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##      <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1        0                 1531.                    1536 84225      0.745
## 2        1                 1483.                    1487 28838      0.255
##
## [[14]]
```

```
## # A tibble: 2 x 5
##   lep_elem `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##      <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1        0                 1531.                    1536 84225      0.745
## 2        1                 1483.                    1487 28838      0.255
##
## [[15]]
## # A tibble: 2 x 5
##   migrant_now `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##         <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1           0                 1519.                    1519 112692    0.997
## 2           1                 1480.                    1487    371    0.00328
##
## [[16]]
## # A tibble: 2 x 5
##   migrant_2yr `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##         <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1           0                 1519.                    1519 112603    0.996
## 2           1                 1477.                    1487    460    0.00407
##
## [[17]]
## # A tibble: 2 x 5
##   migrant_ever `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##          <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1            0                 1519.                    1519 112603    0.996
## 2            1                 1477.                    1487    460    0.00407
##
## [[18]]
## # A tibble: 2 x 5
##   migrant_elem `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##          <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1            0                 1519.                    1519 112603    0.996
## 2            1                 1477.                    1487    460    0.00407
##
## [[19]]
## # A tibble: 2 x 5
##   homeless_now `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##          <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1            0                 1520.                    1519 111512     0.986
## 2            1                 1454.                    1441   1551     0.0137
##
## [[20]]
## # A tibble: 2 x 5
##   homeless_2yr `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##          <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1            0                 1522.                    1519 107454     0.950
## 2            1                 1471.                    1471   5609     0.0496
##
## [[21]]
## # A tibble: 2 x 5
##   homeless_ever `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##           <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1             0                 1522.                    1519 107454     0.950
## 2             1                 1471.                    1471   5609     0.0496
```

```
##
## [[22]]
## # A tibble: 2 x 5
##   homeless_elem `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##           <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1             0                 1522.                    1519 107454     0.950
## 2             1                 1471.                    1471   5609     0.0496
##
## [[23]]
## # A tibble: 2 x 5
##   specialed_now `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##           <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1             0                 1534.                    1536 100655      0.890
## 2             1                 1398.                    1369  12408      0.110
##
## [[24]]
## # A tibble: 2 x 5
##   specialed_2yr `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##           <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1             0                 1534.                    1536 99868      0.883
## 2             1                 1404.                    1384 13195      0.117
##
## [[25]]
## # A tibble: 2 x 5
##   specialed_ever `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##            <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1              0                 1534.                    1536 99868      0.883
## 2              1                 1404.                    1384 13195      0.117
##
## [[26]]
## # A tibble: 2 x 5
##   specialed_elem `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##            <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1              0                 1534.                    1536 99868      0.883
## 2              1                 1404.                    1384 13195      0.117
##
## [[27]]
## # A tibble: 1 x 5
##   withdrawal_date_p0 `mean(readng_scr_p0)` `median(readng_scr_p0)`  count
##                <dbl>                 <dbl>                   <dbl>  <int>
## 1                 NA                 1519.                    1519 113063
## # i 1 more variable: proportion <dbl>
##
## [[28]]
## # A tibble: 1 x 5
##   dropout_inferred_m1 `mean(readng_scr_p0)` `median(readng_scr_p0)`  count
##                 <dbl>                 <dbl>                   <dbl>  <int>
## 1                   0                 1519.                    1519 113063
## # i 1 more variable: proportion <dbl>
##
## [[29]]
## # A tibble: 2 x 5
##   dropout_inferred_p0 `mean(readng_scr_p0)` `median(readng_scr_p0)`  count
##                 <dbl>                 <dbl>                   <dbl>  <int>
```

```
## 1                   0                1519.                     1519 111942
## 2                   1                1547.                     1550   1121
## # i 1 more variable: proportion <dbl>
##
## [[30]]
## # A tibble: 1 x 5
##   persist_inferred_m1 `mean(readng_scr_p0)` `median(readng_scr_p0)`  count
##                 <dbl>                <dbl>                   <dbl>  <int>
## 1                   1                1519.                    1519 113063
## # i 1 more variable: proportion <dbl>
##
## [[31]]
## # A tibble: 2 x 5
##   persist_inferred_p0 `mean(readng_scr_p0)` `median(readng_scr_p0)`  count
##                 <dbl>                <dbl>                   <dbl>  <int>
## 1                   0                1547.                    1550   1121
## 2                   1                1519.                    1519 111942
## # i 1 more variable: proportion <dbl>
##
## [[32]]
## # A tibble: 3 x 5
##   transferred_out_m1 `mean(readng_scr_p0)` `median(readng_scr_p0)` count
##                <dbl>                <dbl>                   <dbl> <int>
## 1                  0                1523.                    1519 99242
## 2                  1                1492.                    1487  6905
## 3                  2                1490.                    1487  6916
## # i 1 more variable: proportion <dbl>
##
## [[33]]
## # A tibble: 3 x 5
##   transferred_out_p0 `mean(readng_scr_p0)` `median(readng_scr_p0)`  count
##                <dbl>                <dbl>                   <dbl>  <int>
## 1                  0                1523.                    1519 101484
## 2                  1                1485.                    1487   4906
## 3                  2                1491.                    1487   6673
## # i 1 more variable: proportion <dbl>
##
## [[34]]
## # A tibble: 2 x 5
##   chronic_absentee_m1 `mean(readng_scr_p0)` `median(readng_scr_p0)`  count
##                 <dbl>                <dbl>                   <dbl>  <int>
## 1                   0                1522.                    1519 107826
## 2                   1                1451.                    1456   5237
## # i 1 more variable: proportion <dbl>
##
## [[35]]
## # A tibble: 2 x 5
##   chronic_absentee_p0 `mean(readng_scr_p0)` `median(readng_scr_p0)`  count
##                 <dbl>                <dbl>                   <dbl>  <int>
## 1                   0                1523.                    1519 107921
## 2                   1                1441.                    1441   5142
## # i 1 more variable: proportion <dbl>
##
## [[36]]
```

```
## # A tibble: 1 x 5
##   glmath_ver_m1 `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##           <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1             1                 1519.                    1519 113063          1
##
## [[37]]
## # A tibble: 2 x 5
##   glmath_lan_m1 `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##           <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1             5                 1521.                    1519 108979     0.964
## 2             6                 1456.                    1452   4084     0.0361
##
## [[38]]
## # A tibble: 1 x 5
##   readng_ver_m1 `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##           <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1             1                 1519.                    1519 113063          1
##
## [[39]]
## # A tibble: 1 x 5
##   readng_ver_p0 `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##           <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1             1                 1519.                    1519 113063          1
##
## [[40]]
## # A tibble: 2 x 5
##   readng_lan_m1 `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##           <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1             5                 1524.                    1519 104385     0.923
## 2             6                 1463.                    1456   8678     0.0768
##
## [[41]]
## # A tibble: 2 x 5
##   readng_lan_p0 `mean(readng_scr_p0)` `median(readng_scr_p0)`  count proportion
##           <dbl>                 <dbl>                   <dbl>  <int>      <dbl>
## 1             5                 1522.                    1519 107222     0.948
## 2             6                 1460.                    1452   5841     0.0517
##
## [[42]]
## # A tibble: 6 x 5
##   age_int `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##     <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1       9                 1605.                    1619    43 0.000380
## 2      10                 1519.                    1519 47758 0.422
## 3      11                 1527.                    1526 60753 0.537
## 4      12                 1420.                    1412  4423 0.0391
## 5      13                 1428.                    1412    85 0.000752
## 6      15                 1412                     1412     1 0.00000884
##
## [[43]]
## # A tibble: 9 x 5
##   attend_p0_d1 `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##          <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1          0.2                 839                      839      1 0.00000884
```

```
## 2          0.3                 1060                      839      3 0.0000265
## 3          0.4                 1090.                    1056.     4 0.0000354
## 4          0.5                 1251.                    1315     16 0.000142
## 5          0.6                 1344.                    1354     37 0.000327
## 6          0.7                 1362.                    1384    168 0.00149
## 7          0.8                 1419.                    1412   1180 0.0104
## 8          0.9                 1488.                    1487  25038 0.221
## 9          1                   1530.                    1536  86616 0.766
##
## [[44]]
## # A tibble: 8 x 5
##   attend_m1_d1 `mean(readng_scr_p0)` `median(readng_scr_p0)` count proportion
##          <dbl>                 <dbl>                   <dbl> <int>      <dbl>
## 1          0.4                 1148.                    1148.     2  0.0000177
## 2          0.5                 1388.                    1376.    12  0.000106
## 3          0.6                 1376                     1384.    22  0.000195
## 4          0.7                 1396.                    1412    139  0.00123
## 5          0.8                 1431.                    1434   1164  0.0103
## 6          0.9                 1492.                    1487  25340  0.224
## 7          1                   1528.                    1536  86365  0.764
## 8          9                   1434.                    1412     19  0.000168
```

From the summaries, it seem every category variable in the list has some impact on reading scores. What about continuous variable reading score from last year?

```
ggplot(df, aes(x = readng_scr_m1, y = readng_scr_p0)) +
  geom_hex(bins = 50) +
  geom_smooth(method = "loess", se = FALSE, color = "red") +                          # or your catego
  labs(title = "Current vs Past score (smoothed)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

We see past year reading score and present year reading score are nonlinearly related. This also indicates the usage of polynomial terms/spines.

6 Part 3: Simple Models

We will use pseudo forward selection to create different models. For each model, we will look at each variables' t-value. If t-value is too small, for example magnitude <1, we remove this variable from this formula. We start with a baseline model with only random effects and intercept as `mod0`. For `mod1` we start with variables that are naturally important for people (`race` and `gender`) and commonly viewed as important for predicting scores (last year score).

```
G4_R <- list()

G4_R[["mod0"]] <- lmer(readng_scr_p0 ~ 1 + (1 | schoolid_nces_enroll_p0),
                       data = df, REML = FALSE)

G4_R[["mod1"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth
                       + (1 | schoolid_nces_enroll_p0),
                       data = df, REML = FALSE)

G4_R[["mod2"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth + age_int
                       + (1 | schoolid_nces_enroll_p0),
                       data = df, REML = FALSE)

G4_R[["mod3"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth
```

```
                        + age_int + frl_ever
                        + (1 | schoolid_nces_enroll_p0),
                        data = df, REML = FALSE)

G4_R[["mod4"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth
                        + age_int + frl_now
                        + (1 | schoolid_nces_enroll_p0),
                        data = df, REML = FALSE)

G4_R[["mod5"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now
                        + (1 | schoolid_nces_enroll_p0),
                        data = df, REML = FALSE)

G4_R[["mod6"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now
                        + lep_now + (1 | schoolid_nces_enroll_p0),
                        data = df, REML = FALSE)

G4_R[["mod7"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now
                        + lep_ever + (1 | schoolid_nces_enroll_p0),
                        data = df, REML = FALSE)

G4_R[["mod8"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now
                        + specialed_now + (1 | schoolid_nces_enroll_p0),
                        data = df, REML = FALSE)

G4_R[["mod9"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now
                        + specialed_now + enrfay_school
                        + (1 | schoolid_nces_enroll_p0),
                        data = df, REML = FALSE)

G4_R[["mod10"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth
                         + frl_now + specialed_now + enrfay_state
                         + (1 | schoolid_nces_enroll_p0),
                         data = df, REML = FALSE)

G4_R[["mod11"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth
                         + frl_now + specialed_now + enrfay_school
                         + transferred_out_p0 + (1 | schoolid_nces_enroll_p0),
                         data = df, REML = FALSE)

G4_R[["mod12"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender
                         + raceth + frl_now + specialed_now
                         + enrfay_school + transferred_out_p0
                         +chronic_absentee_m1 + (1 | schoolid_nces_enroll_p0),
                         data = df, REML = FALSE)

G4_R[["mod13"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth
                         + frl_now + specialed_now + enrfay_school
                         + transferred_out_p0 +chronic_absentee_m1
                         + readng_lan_p0 + (1 | schoolid_nces_enroll_p0),
                         data = df, REML = FALSE)

G4_R[["mod14"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth
```

```
                        + frl_now + specialed_now + enrfay_school
                        + transferred_out_p0 + chronic_absentee_m1
                        + readng_lan_p0 + homeless_now
                        + (1 | schoolid_nces_enroll_p0), data = df, REML = FALSE)

G4_R[["mod15"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth
                        + frl_now + specialed_now + enrfay_school
                        + transferred_out_p0 + chronic_absentee_m1
                        + readng_lan_p0 + homeless_now + migrant_now
                        + (1 | schoolid_nces_enroll_p0),
                        data = df, REML = FALSE)

G4_R[["mod16"]] <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth
                        + frl_now + specialed_now + enrfay_school
                        + transferred_out_p0 + chronic_absentee_m1
                        + readng_lan_p0 + homeless_now + migrant_now
                        + persist_inferred_p0 + (1 | schoolid_nces_enroll_p0),
                        data = df, REML = FALSE)
```

```
for(i in seq_along(G4_R)){
  print(summary(G4_R[[i]]))
}
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: readng_scr_p0 ~ 1 + (1 | schoolid_nces_enroll_p0)
##    Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
## 1440816.9 1440845.8 -720405.4 1440810.9    113060
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.9118 -0.6628 -0.0125  0.6200  4.3816
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)  2936     54.18
##  Residual                             19457    139.49
## Number of obs: 113063, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 1515.344      1.558   972.7
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## readng_scr_p0 ~ readng_scr_m1 + gender + raceth + (1 | schoolid_nces_enroll_p0)
##    Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
## 1333046.6 1333104.2 -666517.3 1333034.6    110271
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -10.0346  -0.5892  -0.0301   0.5549   8.0391
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)   670.8   25.9
##  Residual                             10185.9  100.9
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)   495.105654   3.640790 135.989
## readng_scr_m1   0.712062   0.002254 315.953
## gender         -7.105770   0.612833 -11.595
## raceth          1.897814   0.298104   6.366
##
## Correlation of Fixed Effects:
##             (Intr) rdn__1 gender
## rdng_scr_m1 -0.874
## gender      -0.318  0.078
## raceth      -0.266 -0.098 -0.014
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: readng_scr_p0 ~ readng_scr_m1 + gender + raceth + age_int + (1 |
##     schoolid_nces_enroll_p0)
##    Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
## 1332923.5 1332990.8 -666454.8 1332909.5    110270
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.0128  -0.5897  -0.0300   0.5546   8.0655
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)   668     25.85
##  Residual                             10175    100.87
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)   559.608515   6.819741  82.057
## readng_scr_m1   0.711602   0.002253 315.876
## gender         -6.755548   0.613291 -11.015
## raceth          1.953085   0.297959   6.555
## age_int        -6.081280   0.543646 -11.186
##
## Correlation of Fixed Effects:
##             (Intr) rdn__1 gender raceth
## rdng_scr_m1 -0.482
## gender      -0.126  0.077
## raceth      -0.128 -0.098 -0.013
## age_int     -0.846  0.019 -0.051 -0.017
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
```

```
## readng_scr_p0 ~ readng_scr_m1 + gender + raceth + age_int + frl_ever +
##     (1 | schoolid_nces_enroll_p0)
##    Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
## 1332180.5 1332257.4 -666082.3 1332164.5    110269
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.0857  -0.5895  -0.0299   0.5540   7.9330
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)   608.9   24.68
##  Residual                             10114.7  100.57
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)   587.816800   6.874231  85.510
## readng_scr_m1   0.700520   0.002283 306.842
## gender         -6.993239   0.611494 -11.436
## raceth          0.609426   0.300558   2.028
## age_int        -5.476076   0.542408 -10.096
## frl_ever      -20.682639   0.754832 -27.400
##
## Correlation of Fixed Effects:
##             (Intr) rdn__1 gender raceth age_nt
## rdng_scr_m1 -0.497
## gender      -0.127  0.078
## raceth      -0.149 -0.066 -0.010
## age_int     -0.830  0.011 -0.052 -0.023
## frl_ever    -0.152  0.183  0.015  0.164 -0.041
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: readng_scr_p0 ~ readng_scr_m1 + gender + raceth + age_int + frl_now +
##     (1 | schoolid_nces_enroll_p0)
##    Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
## 1332207.2 1332284.1 -666095.6 1332191.2    110269
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.0720  -0.5887  -0.0300   0.5542   7.9580
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)   625.5   25.01
##  Residual                             10114.5  100.57
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)   584.232157   6.858820  85.180
```

```
## readng_scr_m1   0.701265   0.002279 307.672
## gender         -6.954032   0.611486 -11.372
## raceth          0.730634   0.300167   2.434
## age_int        -5.493776   0.542420 -10.128
## frl_now       -19.751410   0.734966 -26.874
##
## Correlation of Fixed Effects:
##             (Intr) rdn__1 gender raceth age_nt
## rdng_scr_m1 -0.494
## gender      -0.127  0.078
## raceth      -0.146 -0.069 -0.011
## age_int     -0.832  0.012 -0.051 -0.022
## frl_now     -0.135  0.173  0.012  0.152 -0.041
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now + (1 |
##     schoolid_nces_enroll_p0)
##    Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
## 1332307.7 1332375.0 -666146.9 1332293.7    110270
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.0925  -0.5897  -0.0284   0.5539   7.9350
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)   626.7   25.03
##  Residual                             10123.8  100.62
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                Estimate Std. Error t value
## (Intercept)   526.42360    3.80460 138.365
## readng_scr_m1   0.70153    0.00228 307.664
## gender         -7.27280    0.61096 -11.904
## raceth          0.66225    0.30024   2.206
## frl_now       -20.05368    0.73472 -27.294
##
## Correlation of Fixed Effects:
##             (Intr) rdn__1 gender raceth
## rdng_scr_m1 -0.874
## gender      -0.306  0.079
## raceth      -0.297 -0.069 -0.012
## frl_now     -0.305  0.174  0.010  0.151
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now + lep_now +
##     (1 | schoolid_nces_enroll_p0)
##    Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
## 1332222.1 1332299.0 -666103.1 1332206.1    110269
##
## Scaled residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -10.0938  -0.5903  -0.0283   0.5541   7.9401
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)   629.8   25.1
##  Residual                             10115.2  100.6
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)   532.802685   3.863598 137.903
## readng_scr_m1   0.698865   0.002297 304.288
## gender         -7.237899   0.610713 -11.852
## raceth          0.402726   0.301423   1.336
## frl_now       -19.415424   0.737654 -26.320
## lep_now        -7.617398   0.813716  -9.361
##
## Correlation of Fixed Effects:
##             (Intr) rdn__1 gender raceth frl_nw
## rdng_scr_m1 -0.875
## gender      -0.301  0.077
## raceth      -0.307 -0.057 -0.013
## frl_now     -0.283  0.160  0.011  0.142
## lep_now     -0.176  0.123 -0.006  0.092 -0.092
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now + lep_ever +
##     (1 | schoolid_nces_enroll_p0)
##    Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
## 1332291.0 1332367.9 -666137.5 1332275.0    110269
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.0933  -0.5905  -0.0286   0.5539   7.9219
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)   628     25.06
##  Residual                             10122    100.61
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)   529.117410   3.854745 137.264
## readng_scr_m1   0.700568   0.002291 305.842
## gender         -7.255548   0.610914 -11.877
## raceth          0.518401   0.302058   1.716
## frl_now       -19.749263   0.738039 -26.759
## lep_ever       -3.432443   0.793877  -4.324
##
## Correlation of Fixed Effects:
```

```
##             (Intr) rdn__1 gender raceth frl_nw
## rdng_scr_m1 -0.874
## gender      -0.301  0.078
## raceth      -0.309 -0.058 -0.013
## frl_now     -0.285  0.163  0.011  0.139
## lep_ever    -0.161  0.096 -0.007  0.110 -0.095
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now + specialed_now +
##     (1 | schoolid_nces_enroll_p0)
##    Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
## 1330353.4 1330430.2 -665168.7 1330337.4    110269
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.0908  -0.5873  -0.0326   0.5503   7.7584
##
## Random effects:
##  Groups                  Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)  656.7   25.63
##  Residual                             9939.5   99.70
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)   570.762941   3.904057 146.197
## readng_scr_m1   0.670501   0.002364 283.598
## gender         -5.031721   0.607511  -8.283
## raceth          1.107769   0.298093   3.716
## frl_now       -19.962391   0.729214 -27.375
## specialed_now -45.425764   1.022002 -44.448
##
## Correlation of Fixed Effects:
##             (Intr) rdn__1 gender raceth frl_nw
## rdng_scr_m1 -0.882
## gender      -0.274  0.051
## raceth      -0.278 -0.076 -0.009
## frl_now     -0.294  0.165  0.010  0.151
## speciald_nw -0.254  0.293 -0.083 -0.034 -0.002
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now + specialed_now +
##     enrfay_school + (1 | schoolid_nces_enroll_p0)
##    Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
## 1330250.8 1330337.3 -665116.4 1330232.8    110268
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.0926  -0.5867  -0.0333   0.5507   7.7526
##
```

```
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)  650.8   25.51
##  Residual                             9930.9   99.65
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)   546.502157   4.563515 119.755
## readng_scr_m1   0.669577   0.002365 283.126
## gender         -5.016176   0.607245  -8.261
## raceth          1.088092   0.297917   3.652
## frl_now       -19.776196   0.728983 -27.128
## specialed_now -45.495984   1.021565 -44.536
## enrfay_school  26.251423   2.566365  10.229
##
## Correlation of Fixed Effects:
##             (Intr) rdn__1 gender raceth frl_nw spcld_
## rdng_scr_m1 -0.733
## gender      -0.235  0.050
## raceth      -0.235 -0.075 -0.009
## frl_now     -0.265  0.164  0.010  0.151
## speciald_nw -0.213  0.293 -0.083 -0.034 -0.002
## enrfay_schl -0.519 -0.040  0.002 -0.007  0.025 -0.007
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now + specialed_now +
##     enrfay_state + (1 | schoolid_nces_enroll_p0)
##    Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
## 1330351.8 1330438.3 -665166.9 1330333.8    110268
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.0908  -0.5870  -0.0327   0.5503   7.7580
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)  656.7   25.63
##  Residual                             9939.2   99.70
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)   530.167821  21.957028  24.146
## readng_scr_m1   0.670445   0.002364 283.556
## gender         -5.035701   0.607505  -8.289
## raceth          1.108858   0.298089   3.720
## frl_now       -19.962020   0.729204 -27.375
## specialed_now -45.429526   1.021988 -44.452
## enrfay_state   40.698856  21.662144   1.879
```

```
##
## Correlation of Fixed Effects:
##             (Intr) rdn__1 gender raceth frl_nw spcld_
## rdng_scr_m1 -0.144
## gender      -0.045  0.051
## raceth      -0.051 -0.076 -0.009
## frl_now     -0.053  0.165  0.010  0.151
## speciald_nw -0.043  0.293 -0.083 -0.034 -0.002
## enrfay_stat -0.984 -0.013 -0.003  0.002  0.000 -0.002
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now + specialed_now +
##     enrfay_school + transferred_out_p0 + (1 | schoolid_nces_enroll_p0)
##    Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
## 1330221.2 1330317.3 -665100.6 1330201.2    110267
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.0943  -0.5869  -0.0331   0.5501   7.7465
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)  645.6   25.41
##  Residual                             9928.8   99.64
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                       Estimate Std. Error t value
## (Intercept)        548.587325   4.578107 119.828
## readng_scr_m1        0.669295   0.002365 282.975
## gender              -5.026657   0.607180  -8.279
## raceth               1.083019   0.297836   3.636
## frl_now            -19.663447   0.729052 -26.971
## specialed_now      -45.524448   1.021455 -44.568
## enrfay_school       25.114700   2.574101   9.757
## transferred_out_p0  -3.452982   0.614245  -5.622
##
## Correlation of Fixed Effects:
##             (Intr) rdn__1 gender raceth frl_nw spcld_ enrfy_
## rdng_scr_m1 -0.732
## gender      -0.235  0.050
## raceth      -0.234 -0.075 -0.009
## frl_now     -0.261  0.163  0.010  0.151
## speciald_nw -0.213  0.293 -0.083 -0.034 -0.003
## enrfay_schl -0.522 -0.038  0.003 -0.006  0.023 -0.007
## trnsfrrd__0 -0.083  0.024  0.003  0.003 -0.028  0.006  0.079
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
```

```
## readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now + specialed_now +
##     enrfay_school + transferred_out_p0 + chronic_absentee_m1 +
##     (1 | schoolid_nces_enroll_p0)
##    Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
## 1330163.1 1330268.8 -665070.5 1330141.1    110266
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.0936  -0.5867  -0.0329   0.5506   7.7412
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)  640.7   25.31
##  Residual                             9924.1   99.62
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                       Estimate Std. Error t value
## (Intercept)         551.438401   4.591558 120.098
## readng_scr_m1         0.668438   0.002367 282.367
## gender               -5.027853   0.607031  -8.283
## raceth                1.131603   0.297783   3.800
## frl_now             -19.323085   0.730086 -26.467
## specialed_now       -45.270866   1.021703 -44.309
## enrfay_school        23.556311   2.581327   9.126
## transferred_out_p0   -3.275876   0.614516  -5.331
## chronic_absentee_m1 -11.331786   1.460530  -7.759
##
## Correlation of Fixed Effects:
##             (Intr) rdn__1 gender raceth frl_nw spcld_ enrfy_ trn__0
## rdng_scr_m1 -0.733
## gender      -0.234  0.050
## raceth      -0.231 -0.076 -0.009
## frl_now     -0.255  0.160  0.010  0.152
## speciald_nw -0.210  0.291 -0.083 -0.033 -0.001
## enrfay_schl -0.525 -0.034  0.003 -0.008  0.018 -0.009
## trnsfrrd__0 -0.080  0.022  0.003  0.004 -0.026  0.007  0.076
## chrnc_bsn_1 -0.081  0.048  0.000 -0.021 -0.061 -0.032  0.078 -0.038
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now + specialed_now +
##     enrfay_school + transferred_out_p0 + chronic_absentee_m1 +
##     readng_lan_p0 + (1 | schoolid_nces_enroll_p0)
##    Data: df
##
##      AIC      BIC   logLik deviance df.resid
##  1329862  1329977  -664919  1329838   110265
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -10.1030  -0.5866  -0.0350   0.5490   7.7351
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)  625.7   25.01
##  Residual                             9898.9   99.49
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                       Estimate Std. Error t value
## (Intercept)         690.095556   9.189088  75.099
## readng_scr_m1         0.666778   0.002366 281.817
## gender               -5.085870   0.606257  -8.389
## raceth                1.089336   0.297276   3.664
## frl_now             -18.846977   0.729294 -25.843
## specialed_now       -45.851540   1.020917 -44.912
## enrfay_school        24.799874   2.578877   9.617
## transferred_out_p0   -3.565698   0.613890  -5.808
## chronic_absentee_m1 -12.085554   1.459236  -8.282
## readng_lan_p0       -27.172066   1.559262 -17.426
##
## Correlation of Fixed Effects:
##             (Intr) rdn__1 gender raceth frl_nw spcld_ enrfy_ trn__0 chr__1
## rdng_scr_m1 -0.402
## gender      -0.122  0.051
## raceth      -0.123 -0.076 -0.009
## frl_now     -0.094  0.158  0.010  0.151
## speciald_nw -0.133  0.292 -0.083 -0.033 -0.002
## enrfay_schl -0.238 -0.035  0.003 -0.008  0.019 -0.010
## trnsfrrd__0 -0.063  0.023  0.003  0.004 -0.027  0.008  0.075
## chrnc_bsn_1 -0.066  0.049  0.000 -0.021 -0.062 -0.031  0.077 -0.037
## redng_ln_p0 -0.867  0.042  0.006  0.008 -0.038  0.033 -0.027  0.026  0.029
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now + specialed_now +
##     enrfay_school + transferred_out_p0 + chronic_absentee_m1 +
##     readng_lan_p0 + homeless_now + (1 | schoolid_nces_enroll_p0)
##    Data: df
##
##      AIC      BIC   logLik deviance df.resid
##  1329854  1329979  -664914  1329828   110264
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.1029  -0.5871  -0.0349   0.5492   7.7347
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)  624.5   24.99
##  Residual                             9898.1   99.49
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
```

```
## Fixed effects:
##                       Estimate Std. Error t value
## (Intercept)         691.025038   9.193123  75.168
## readng_scr_m1         0.666685   0.002366 281.768
## gender               -5.095597   0.606242  -8.405
## raceth                1.086015   0.297255   3.653
## frl_now             -18.733572   0.730117 -25.658
## specialed_now       -45.827295   1.020902 -44.889
## enrfay_school        24.354111   2.582610   9.430
## transferred_out_p0   -3.495872   0.614263  -5.691
## chronic_absentee_m1 -11.866811   1.460820  -8.123
## readng_lan_p0       -27.229461   1.559230 -17.463
## homeless_now         -8.364714   2.634521  -3.175
##
## Correlation of Fixed Effects:
##             (Intr) rdn__1 gender raceth frl_nw spcld_ enrfy_ trn__0 chr__1
## rdng_scr_m1 -0.402
## gender      -0.122  0.051
## raceth      -0.123 -0.076 -0.009
## frl_now     -0.093  0.157  0.010  0.151
## speciald_nw -0.133  0.292 -0.083 -0.033 -0.001
## enrfay_schl -0.239 -0.034  0.003 -0.008  0.017 -0.010
## trnsfrrd__0 -0.061  0.022  0.003  0.004 -0.025  0.008  0.073
## chrnc_bsn_1 -0.064  0.049  0.000 -0.021 -0.059 -0.030  0.074 -0.035
## redng_ln_p0 -0.867  0.042  0.006  0.008 -0.039  0.033 -0.027  0.026  0.029
## homeless_nw -0.032  0.013  0.005  0.004 -0.049 -0.007  0.055 -0.036 -0.047
##             rdn__0
## rdng_scr_m1
## gender
## raceth
## frl_now
## speciald_nw
## enrfay_schl
## trnsfrrd__0
## chrnc_bsn_1
## redng_ln_p0
## homeless_nw  0.012
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now + specialed_now +
##     enrfay_school + transferred_out_p0 + chronic_absentee_m1 +
##     readng_lan_p0 + homeless_now + migrant_now + (1 | schoolid_nces_enroll_p0)
##    Data: df
##
##       AIC       BIC    logLik  deviance  df.resid
## 1329854.9 1329989.5 -664913.5 1329826.9    110263
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.1029  -0.5871  -0.0348   0.5492   7.7346
##
## Random effects:
```

```
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)  624.6   24.99
##  Residual                             9898.0   99.49
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                       Estimate Std. Error t value
## (Intercept)         691.079896   9.193264  75.172
## readng_scr_m1         0.666663   0.002366 281.748
## gender               -5.096335   0.606239  -8.406
## raceth                1.084640   0.297258   3.649
## frl_now             -18.723483   0.730184 -25.642
## specialed_now       -45.831140   1.020903 -44.893
## enrfay_school        24.335699   2.582661   9.423
## transferred_out_p0   -3.496625   0.614261  -5.692
## chronic_absentee_m1 -11.849085   1.460917  -8.111
## readng_lan_p0       -27.226277   1.559232 -17.461
## homeless_now         -8.328414   2.634753  -3.161
## migrant_now          -5.329247   5.285903  -1.008
##
## Correlation of Fixed Effects:
##             (Intr) rdn__1 gender raceth frl_nw spcld_ enrfy_ trn__0 chr__1
## rdng_scr_m1 -0.402
## gender      -0.122  0.051
## raceth      -0.123 -0.076 -0.009
## frl_now     -0.092  0.157  0.010  0.151
## speciald_nw -0.133  0.292 -0.083 -0.033 -0.001
## enrfay_schl -0.240 -0.034  0.003 -0.008  0.016 -0.010
## trnsfrrd__0 -0.061  0.022  0.003  0.004 -0.025  0.008  0.073
## chrnc_bsn_1 -0.064  0.049  0.000 -0.021 -0.059 -0.030  0.074 -0.035
## redng_ln_p0 -0.867  0.042  0.006  0.008 -0.039  0.033 -0.027  0.026  0.029
## homeless_nw -0.032  0.013  0.005  0.003 -0.049 -0.007  0.054 -0.036 -0.047
## migrant_now -0.006  0.009  0.001  0.005 -0.014  0.004  0.007  0.001 -0.012
##             rdn__0 hmlss_
## rdng_scr_m1
## gender
## raceth
## frl_now
## speciald_nw
## enrfay_schl
## trnsfrrd__0
## chrnc_bsn_1
## redng_ln_p0
## homeless_nw  0.012
## migrant_now -0.002 -0.014
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now + specialed_now +
##     enrfay_school + transferred_out_p0 + chronic_absentee_m1 +
##     readng_lan_p0 + homeless_now + migrant_now + persist_inferred_p0 +
##     (1 | schoolid_nces_enroll_p0)
##    Data: df
```

```
##
##       AIC       BIC    logLik  deviance  df.resid
## 1329856.6 1330000.8 -664913.3 1329826.6    110262
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.1031  -0.5872  -0.0349   0.5491   7.7344
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  schoolid_nces_enroll_p0 (Intercept)  624.7   24.99
##  Residual                             9898.0   99.49
## Number of obs: 110277, groups:  schoolid_nces_enroll_p0, 1352
##
## Fixed effects:
##                       Estimate Std. Error t value
## (Intercept)         689.534995   9.676061  71.262
## readng_scr_m1         0.666674   0.002366 281.741
## gender               -5.097142   0.606240  -8.408
## raceth                1.086099   0.297271   3.654
## frl_now             -18.729151   0.730268 -25.647
## specialed_now       -45.830275   1.020903 -44.892
## enrfay_school        24.324139   2.582756   9.418
## transferred_out_p0   -3.506416   0.614559  -5.706
## chronic_absentee_m1 -11.839088   1.461045  -8.103
## readng_lan_p0       -27.231073   1.559260 -17.464
## homeless_now         -8.323358   2.634768  -3.159
## migrant_now          -5.320454   5.285924  -1.007
## persist_inferred_p0   1.578062   3.082669   0.512

##
## Correlation matrix not shown by default, as p = 13 > 12.
## Use print(summary(G4_R[[i]]), correlation=TRUE)  or
##     vcov(summary(G4_R[[i]]))        if you need it

## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

7 Part 4: Select the best simple Model

We'll select the best simple model using AIC, BIC and MSE. The lower the better. We first print out the AIC/BIC/MSE values for all models.

```r
G4_R_Sum <- list()

for(i in seq_along(G4_R)) {
  G4_R_Sum[[i]] <- list(
    model = G4_R[[i]],
    AIC = AIC(G4_R[[i]]),
    BIC = BIC(G4_R[[i]]),
    MSE = mean(residuals(G4_R[[i]])^2)
  )
```

```
}
names(G4_R_Sum) <- names(G4_R)

aic_values <- sapply(G4_R_Sum, function(x) x$AIC)
aic_values
```

```
##    mod0    mod1    mod2    mod3    mod4    mod5    mod6    mod7    mod8    mod9
## 1440817 1333047 1332924 1332181 1332207 1332308 1332222 1332291 1330353 1330251
##   mod10   mod11   mod12   mod13   mod14   mod15   mod16
## 1330352 1330221 1330163 1329862 1329854 1329855 1329857
```

```
bic_values <- sapply(G4_R_Sum, function(x) x$BIC)
bic_values
```

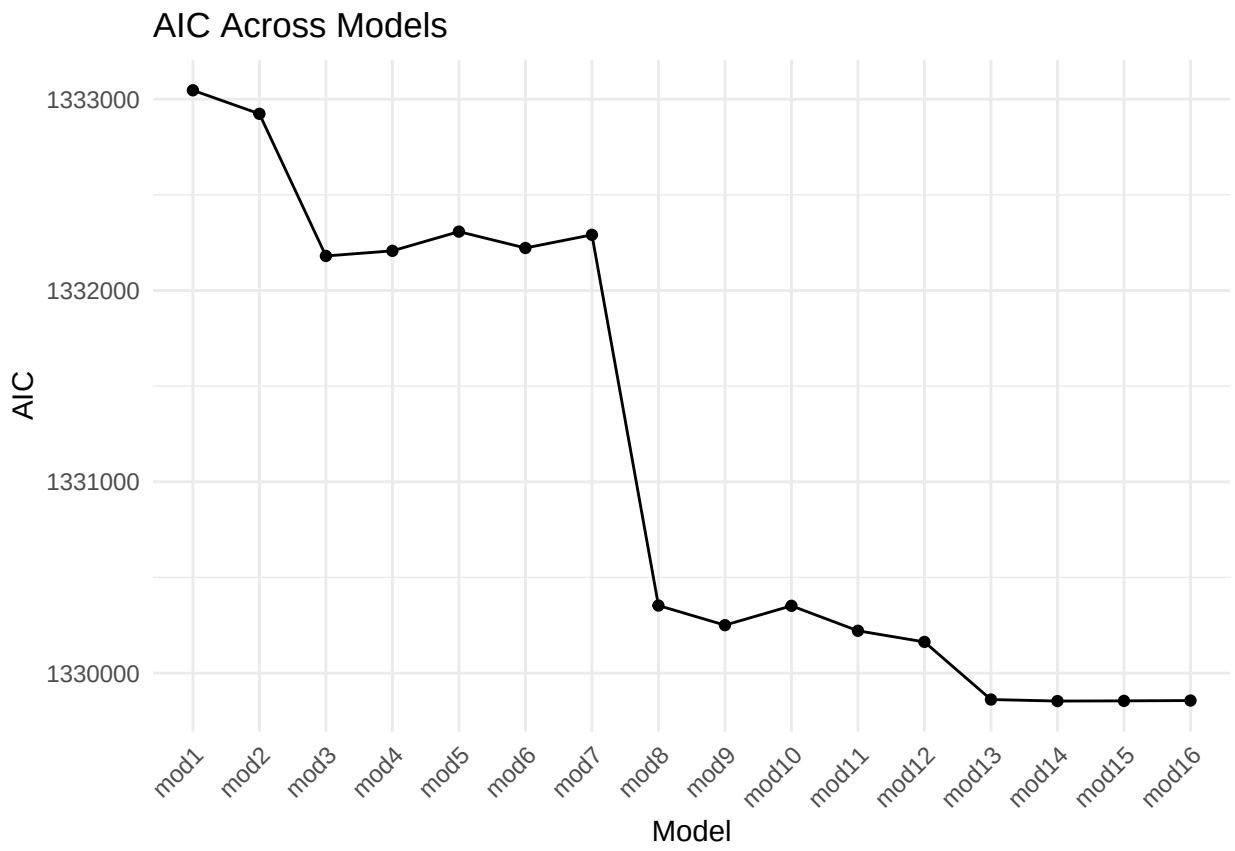
```
##    mod0    mod1    mod2    mod3    mod4    mod5    mod6    mod7    mod8    mod9
## 1440846 1333104 1332991 1332257 1332284 1332375 1332299 1332368 1330430 1330337
##   mod10   mod11   mod12   mod13   mod14   mod15   mod16
## 1330438 1330317 1330269 1329977 1329979 1329989 1330001
```

```
mse_values <- sapply(G4_R_Sum, function(x) x$MSE)
mse_values
```

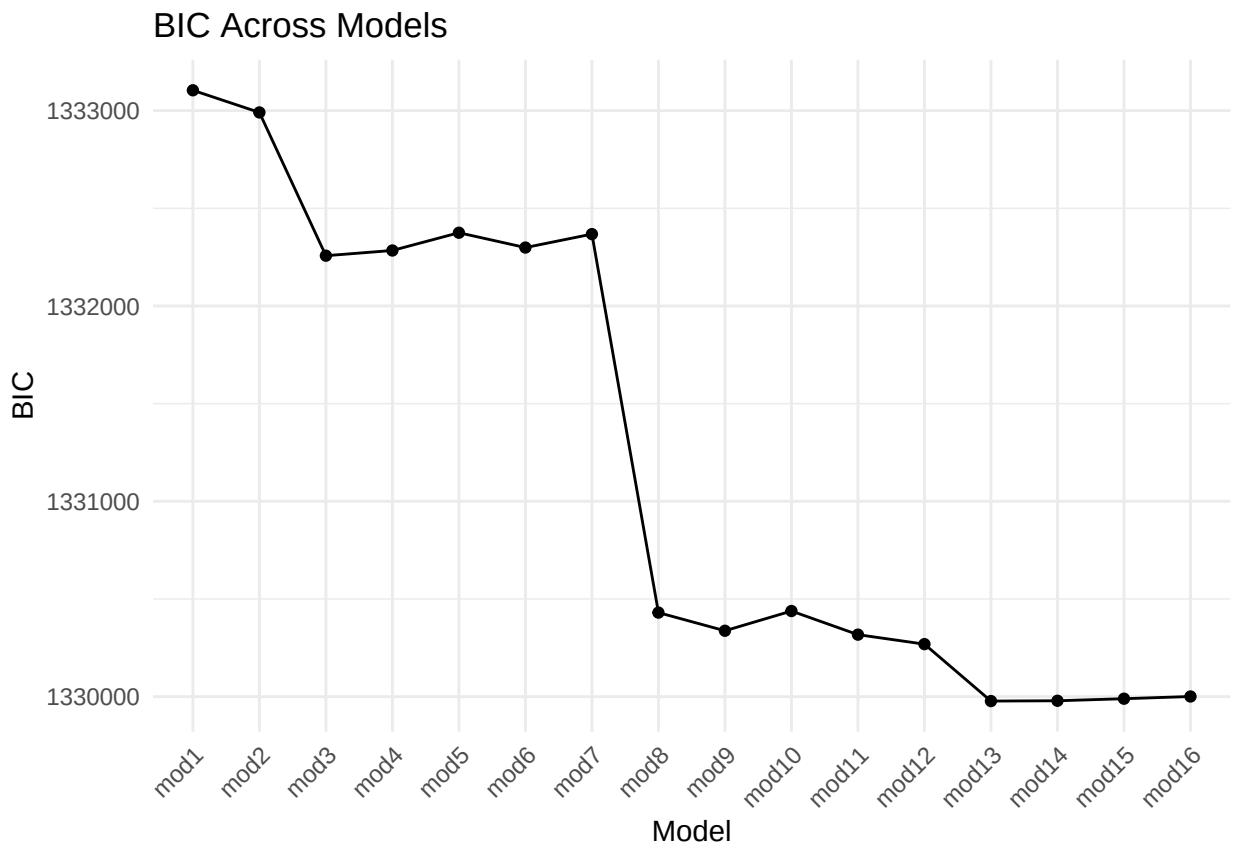
```
##      mod0      mod1      mod2      mod3      mod4      mod5      mod6      mod7
## 19249.178 10086.110 10075.037 10017.204 10016.524 10025.727 10017.108 10023.778
##      mod8      mod9     mod10     mod11     mod12     mod13     mod14     mod15
##  9842.044  9833.642  9841.720  9831.718  9827.167  9802.577  9801.889  9801.780
##     mod16
##  9801.751
```

We create line plots to visualize how AIC/BIC/MSE change with the increase of variables.

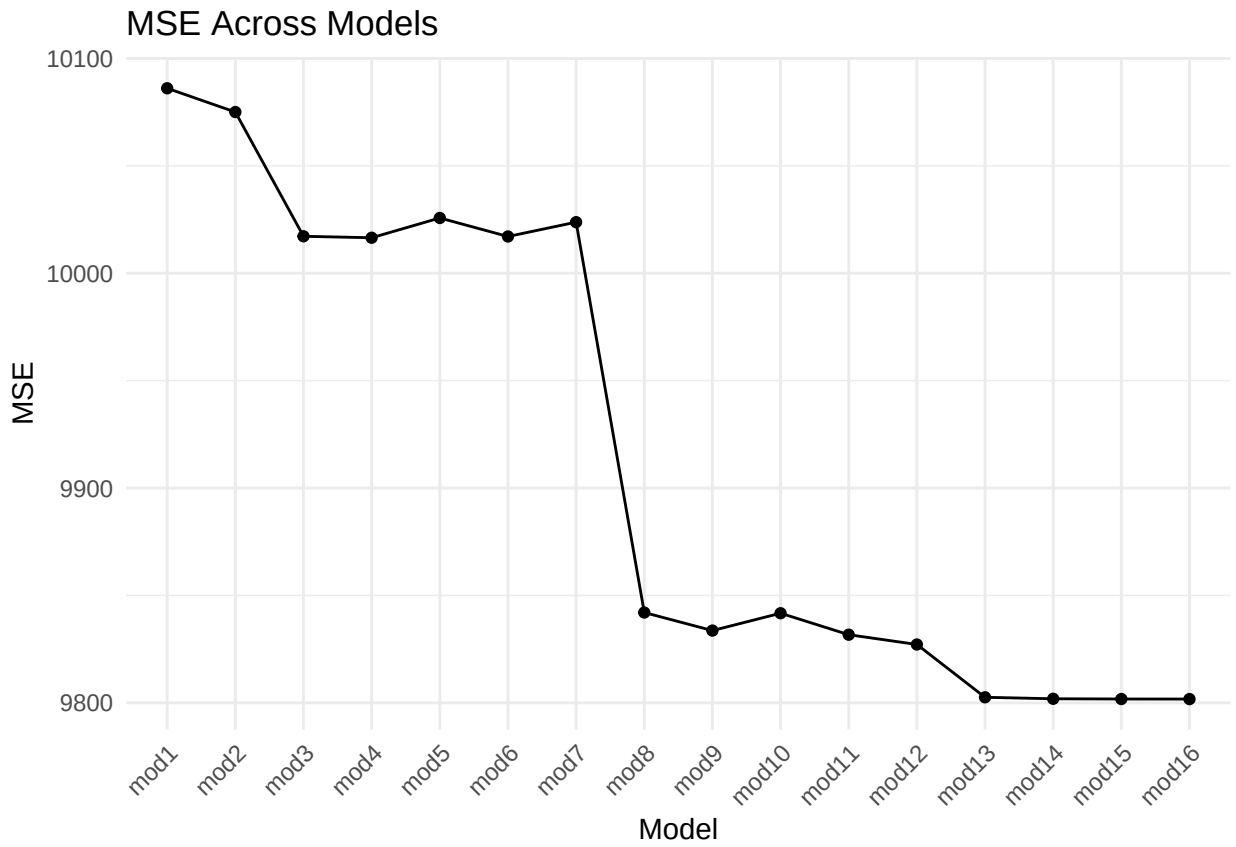
```
aic_df <- data.frame(
  Model = names(aic_values),
  AIC = aic_values
)
aic_df <- subset(aic_df, Model != "mod0")
aic_df$Order <- as.numeric(str_extract(aic_df$Model, "\\d+"))
aic_df <- aic_df[order(aic_df$Order), ]
aic_df$Model <- factor(aic_df$Model, levels = aic_df$Model)

ggplot(aic_df, aes(x = Model, y = AIC, group = 1)) +
  geom_line() +
  geom_point() +
  labs(title = "AIC Across Models", x = "Model", y = "AIC") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
bic_df <- data.frame(
  Model = names(bic_values),
  BIC = bic_values
)
bic_df <- subset(bic_df, Model != "mod0")
bic_df$Order <- as.numeric(str_extract(bic_df$Model, "\\d+"))
bic_df <- bic_df[order(bic_df$Order), ]
bic_df$Model <- factor(bic_df$Model, levels = bic_df$Model)
ggplot(bic_df, aes(x = Model, y = BIC, group = 1)) +
  geom_line() +
  geom_point() +
  labs(title = "BIC Across Models", x = "Model", y = "BIC") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
mse_df <- data.frame(
  Model = names(mse_values),
  MSE = mse_values
)
mse_df <- subset(mse_df, Model != "mod0")
mse_df$Order <- as.numeric(str_extract(mse_df$Model, "\\d+"))
mse_df <- mse_df[order(mse_df$Order), ]
mse_df$Model <- factor(mse_df$Model, levels = mse_df$Model)

ggplot(mse_df, aes(x = Model, y = MSE, group = 1)) +
  geom_line() +
  geom_point() +
  labs(title = "MSE Across Models", x = "Model", y = "MSE") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Both AIC/BIC/MSE plots indicates that model 13,14,15,16 are better choices. We then choose the simpler model, model 13 as the winner.

readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now + specialed_now + enrfay_school + transferred_out_p0 +chronic_absentee_m1 + readng_lan_p0 + (1 | schoolid_nces_enroll_p0)

8 Part 5: Normalization

In this part, we investigate whether normalization will help with our model. We will use the winner model, model 13, as the control.

```
unscaled <- lmer(readng_scr_p0 ~ readng_scr_m1 + gender + raceth + frl_now
                 + specialed_now + enrfay_school + transferred_out_p0
                 +chronic_absentee_m1 + readng_lan_p0
                 + (1 | schoolid_nces_enroll_p0),
                 data = df, REML = FALSE)

scaled <- lmer(scale(readng_scr_p0) ~ scale(readng_scr_m1)
               + gender + raceth + frl_now + specialed_now
               + enrfay_school + transferred_out_p0
               +chronic_absentee_m1 + readng_lan_p0
               + (1 | schoolid_nces_enroll_p0),
               data = df, REML = FALSE)

cat("\nThe AIC for unnormalized model is", AIC(unscaled))
```

```
##
## The AIC for unnormalized model is 1329862
```

```
cat("\nThe AIC for normalized model is", AIC(scaled))
```

```
##
## The AIC for normalized model is 224715.1
```

```
cat("\nThe BIC for unnormalized model is", BIC(unscaled))
```

```
##
## The BIC for unnormalized model is 1329977
```

```
cat("\nThe BIC for normalized model is", BIC(scaled))
```

```
##
## The BIC for normalized model is 224830.5
```

We see normalization results in a better model. So we update our best model to the normalized one.

9 Part 6: Splines

Now we explore the use of spline for our model. `mod0` is without splines, `modi` is has spline of degree i. We create 11 models, first one without any splines, the rest natural splines 1-10.

```
splines <- list()
splines[['mod0']] <- lmer(scale(readng_scr_p0) ~ ns(scale(readng_scr_m1),i)
                          + gender + raceth + frl_now + specialed_now
                          + enrfay_school + transferred_out_p0
                          +chronic_absentee_m1 + readng_lan_p0
                          + (1 | schoolid_nces_enroll_p0),
                          data = df, REML = FALSE)
for(i in 1:10){
  splines[[paste('mod', i)]] <- lmer(scale(readng_scr_p0) ~
                                       ns(scale(readng_scr_m1),i)
                                     + gender + raceth + frl_now
                                     + specialed_now + enrfay_school
                                     + transferred_out_p0 +chronic_absentee_m1
                                     + readng_lan_p0
                                     + (1 | schoolid_nces_enroll_p0),
                                     data = df, REML = FALSE)
}
```

Now like we did before, we get AIC/BIC/MSEs for each mdoel.

```
splines_Sum <- list()

for(i in seq_along(splines)) {
  splines_Sum[[i]] <- list(
```

```
    model = splines[[i]],
    AIC = AIC(splines[[i]]),
    BIC = BIC(splines[[i]]),
    MSE = mean(residuals(splines[[i]])^2)
  )
}
names(splines_Sum) <- names(splines)

aic_values <- sapply(splines_Sum, function(x) x$AIC)
aic_values
```

```
##     mod0    mod 1    mod 2    mod 3    mod 4    mod 5    mod 6    mod 7
## 218354.5 224715.1 224545.1 218545.3 218467.8 218429.6 218422.3 218404.7
##    mod 8    mod 9   mod 10
## 218401.1 218400.5 218388.7
```

```
bic_values <- sapply(splines_Sum, function(x) x$BIC)
bic_values
```

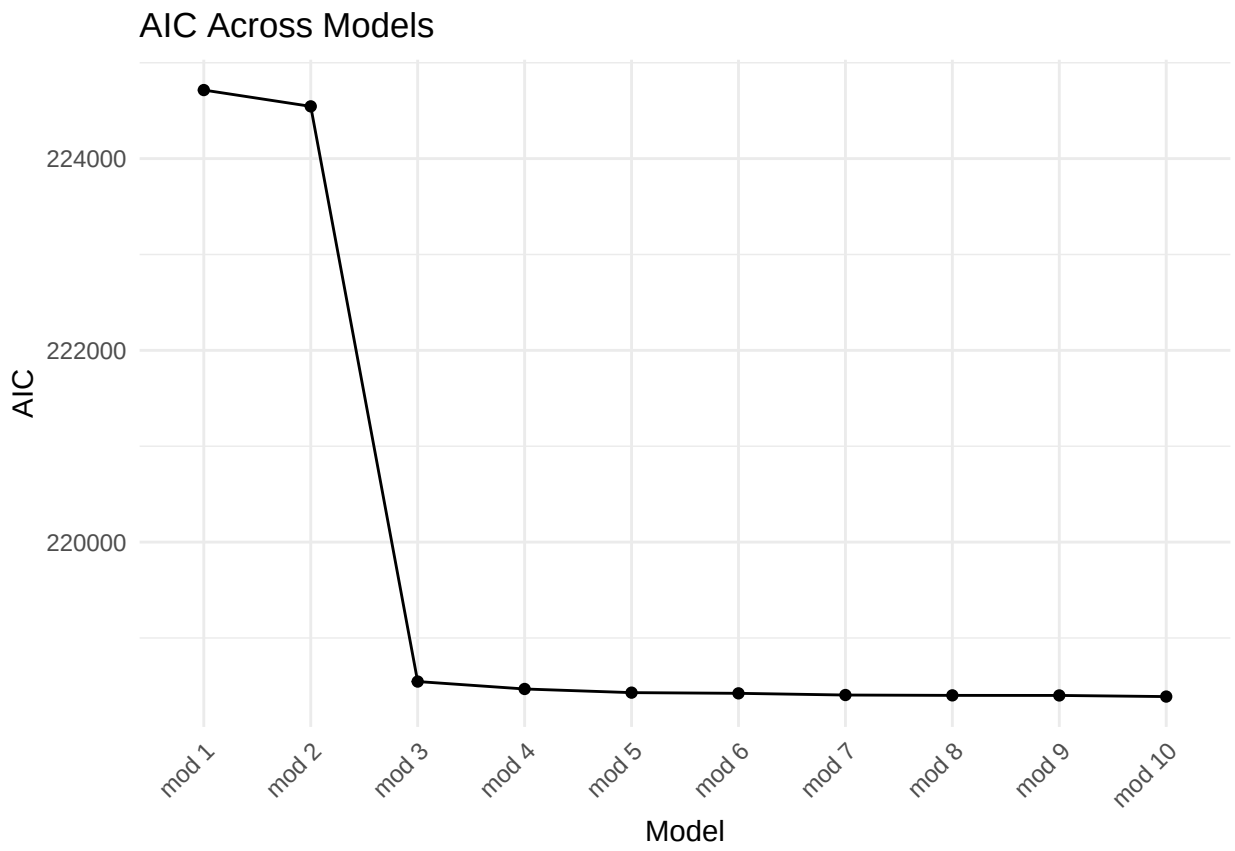
```
##     mod0    mod 1    mod 2    mod 3    mod 4    mod 5    mod 6    mod 7
## 218623.6 224830.5 224670.1 218679.9 218612.0 218583.4 218585.7 218577.7
##    mod 8    mod 9   mod 10
## 218583.7 218592.7 218590.5
```

```
mse_values <- sapply(splines_Sum, function(x) x$MSE)
mse_values
```

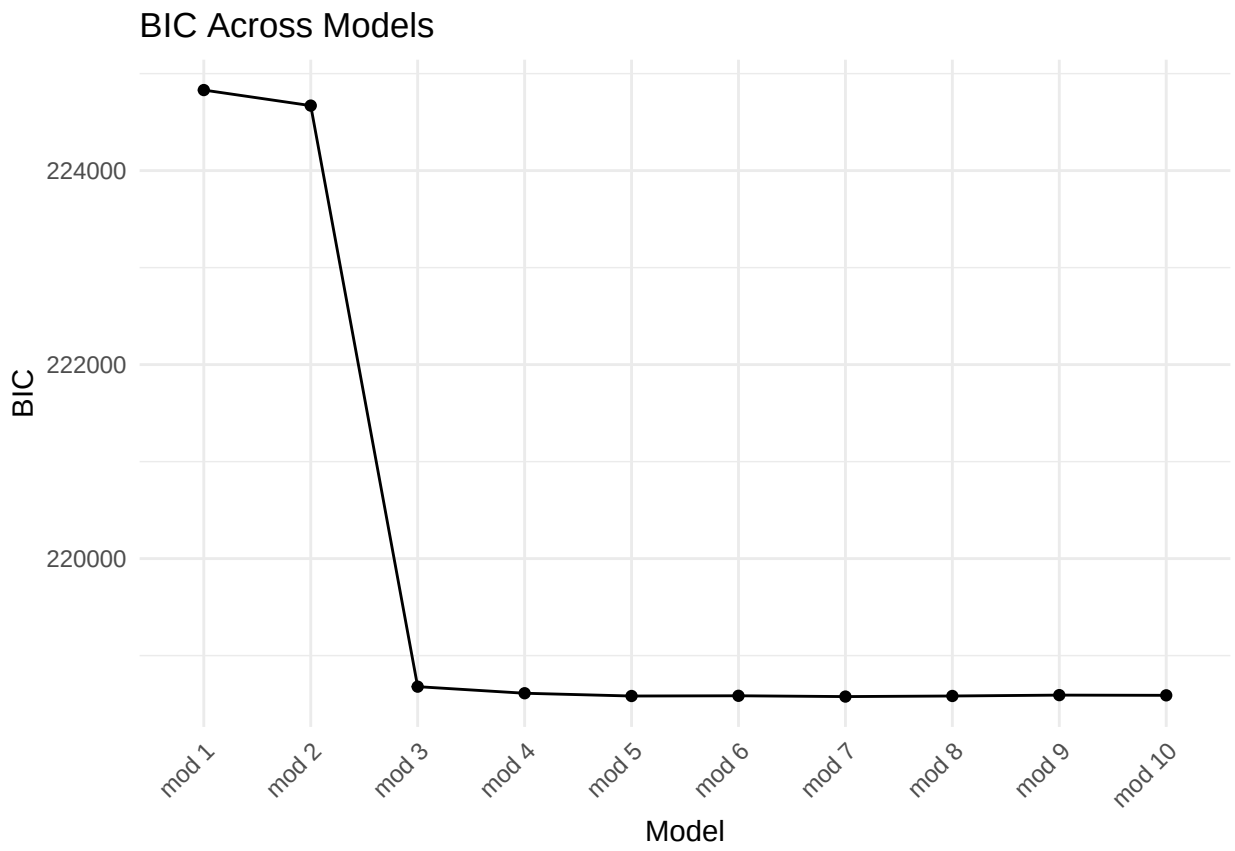
```
##      mod0     mod 1     mod 2     mod 3     mod 4     mod 5     mod 6     mod 7
## 0.4111787 0.4355468 0.4348633 0.4119668 0.4116949 0.4115531 0.4115187 0.4114453
##     mod 8     mod 9    mod 10
## 0.4114215 0.4114131 0.4113599
```

We create line plots to visualize how AIC/BIC/MSE change with the increase of spline degrees.

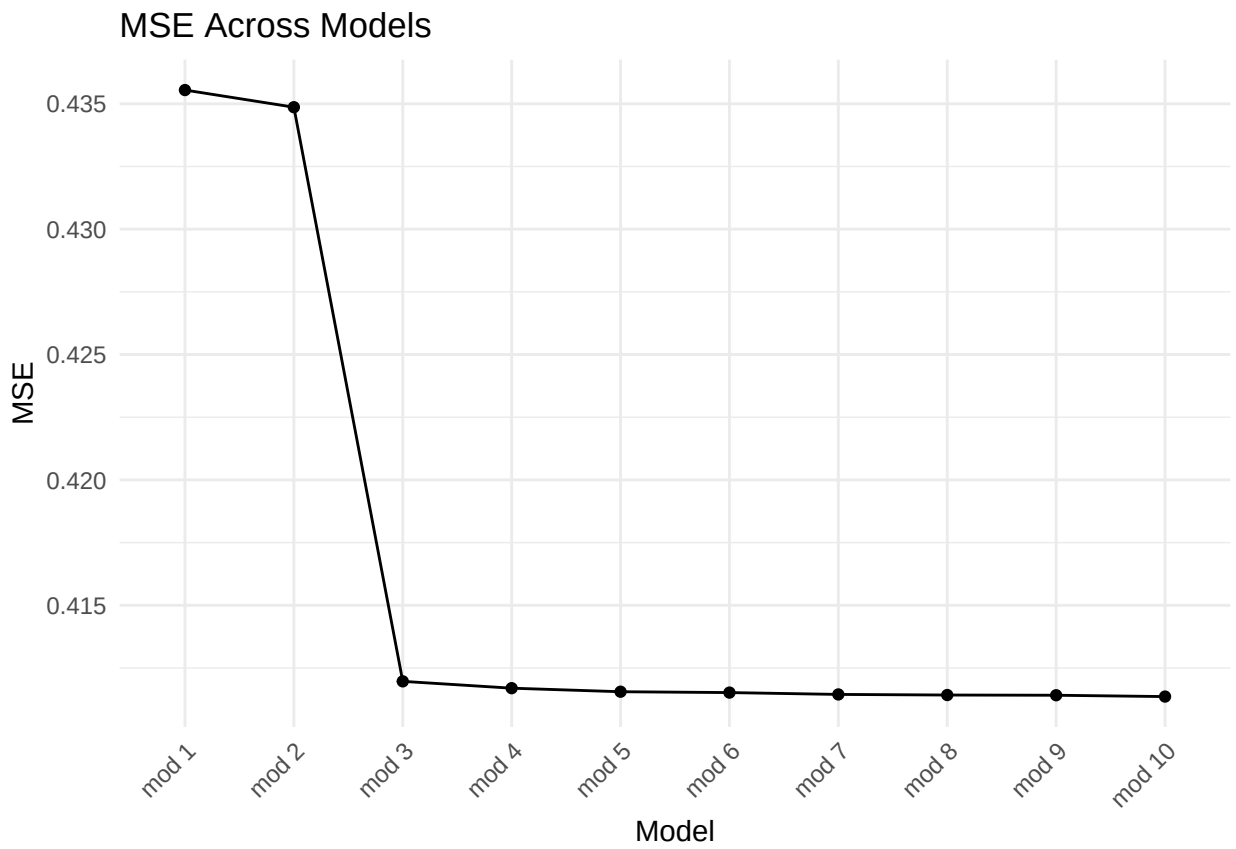
```
aic_df <- data.frame(
  Model = names(aic_values),
  AIC = aic_values
)
aic_df <- subset(aic_df, Model != "mod0")
aic_df$Order <- as.numeric(str_extract(aic_df$Model, "\\d+"))
aic_df <- aic_df[order(aic_df$Order), ]
aic_df$Model <- factor(aic_df$Model, levels = aic_df$Model)

ggplot(aic_df, aes(x = Model, y = AIC, group = 1)) +
  geom_line() +
  geom_point() +
  labs(title = "AIC Across Models", x = "Model", y = "AIC") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
bic_df <- data.frame(
  Model = names(bic_values),
  BIC = bic_values
)
bic_df <- subset(bic_df, Model != "mod0")
bic_df$Order <- as.numeric(str_extract(bic_df$Model, "\\d+"))
bic_df <- bic_df[order(bic_df$Order), ]
bic_df$Model <- factor(bic_df$Model, levels = bic_df$Model)
ggplot(bic_df, aes(x = Model, y = BIC, group = 1)) +
  geom_line() +
  geom_point() +
  labs(title = "BIC Across Models", x = "Model", y = "BIC") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
mse_df <- data.frame(
  Model = names(mse_values),
  MSE = mse_values
)
mse_df <- subset(mse_df, Model != "mod0")
mse_df$Order <- as.numeric(str_extract(mse_df$Model, "\\d+"))
mse_df <- mse_df[order(mse_df$Order), ]
mse_df$Model <- factor(mse_df$Model, levels = mse_df$Model)

ggplot(mse_df, aes(x = Model, y = MSE, group = 1)) +
  geom_line() +
  geom_point() +
  labs(title = "MSE Across Models", x = "Model", y = "MSE") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Both AIC,BIC,MSE suggests natural splines with 3 degrees of freedom is the best choice.

10 Part 7: Best Model for Grade 4 Reading

So based on our analysis, the best model we have for grade 4 reading is

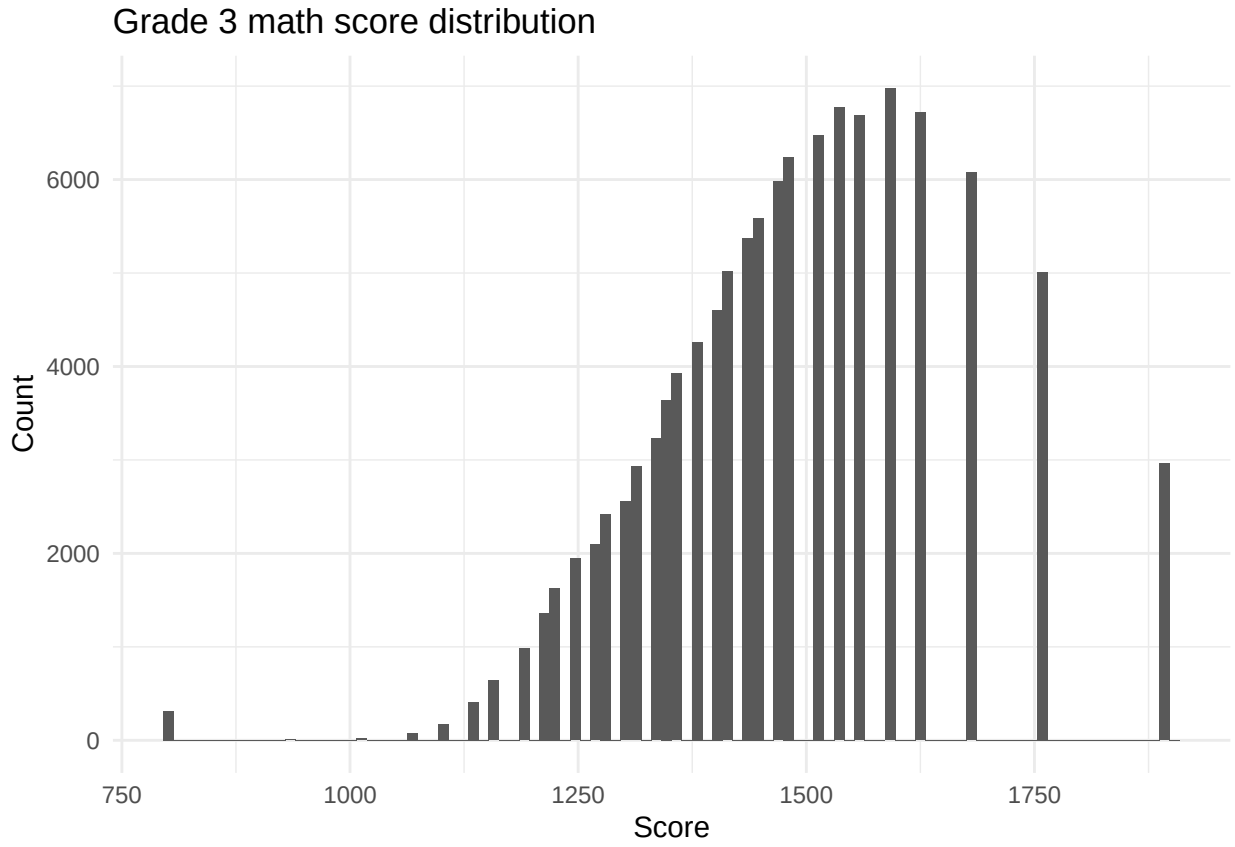
scale(readng_scr_p0) ~ ns(scale(readng_scr_m1),3) + gender + raceth + frl_now + specialed_now + enrfay_school + transferred_out_p0 +chronic_absentee_m1 + readng_lan_p0 + (1 | schoolid_nces_enroll_p0)

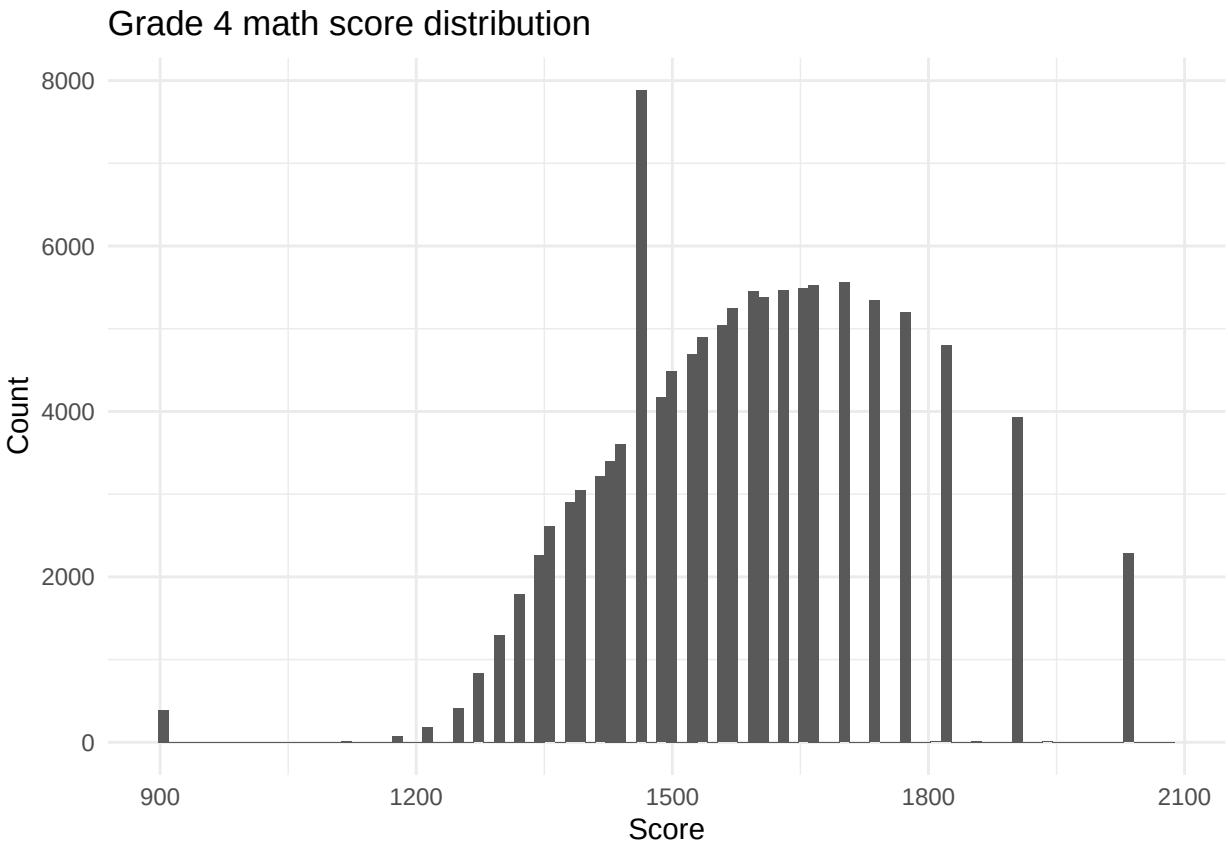
11 Part 8: Extremely Low Math Scores

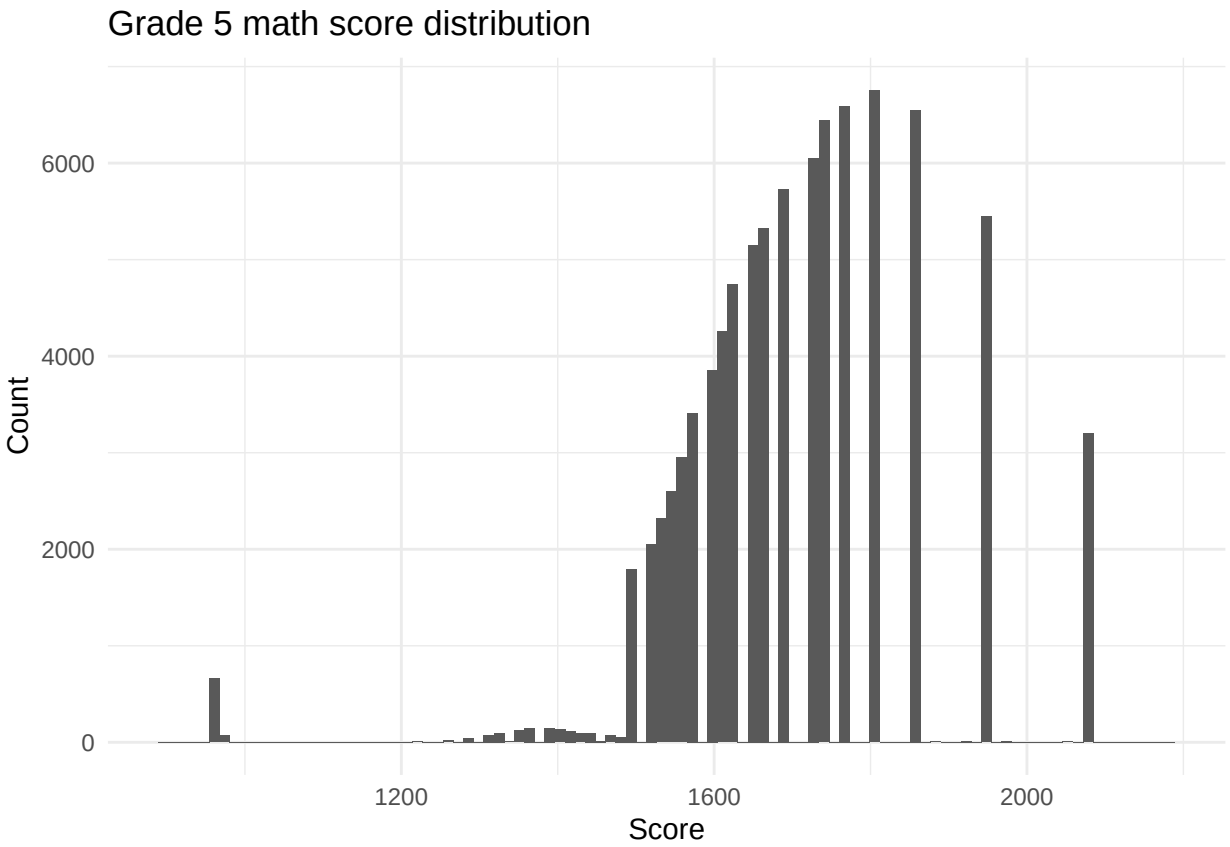
One worth-noting pattern in our data is extremely low math scores for students in grade 6-8. We have lots of students scored 1043, which we can reasonably guess corresponding to raw score 0 in STAAR math test based on past grading schemes (Grading scheme for 2019 was not found online). Here's a histogram show the pattern. Grade 6 math score distribution:

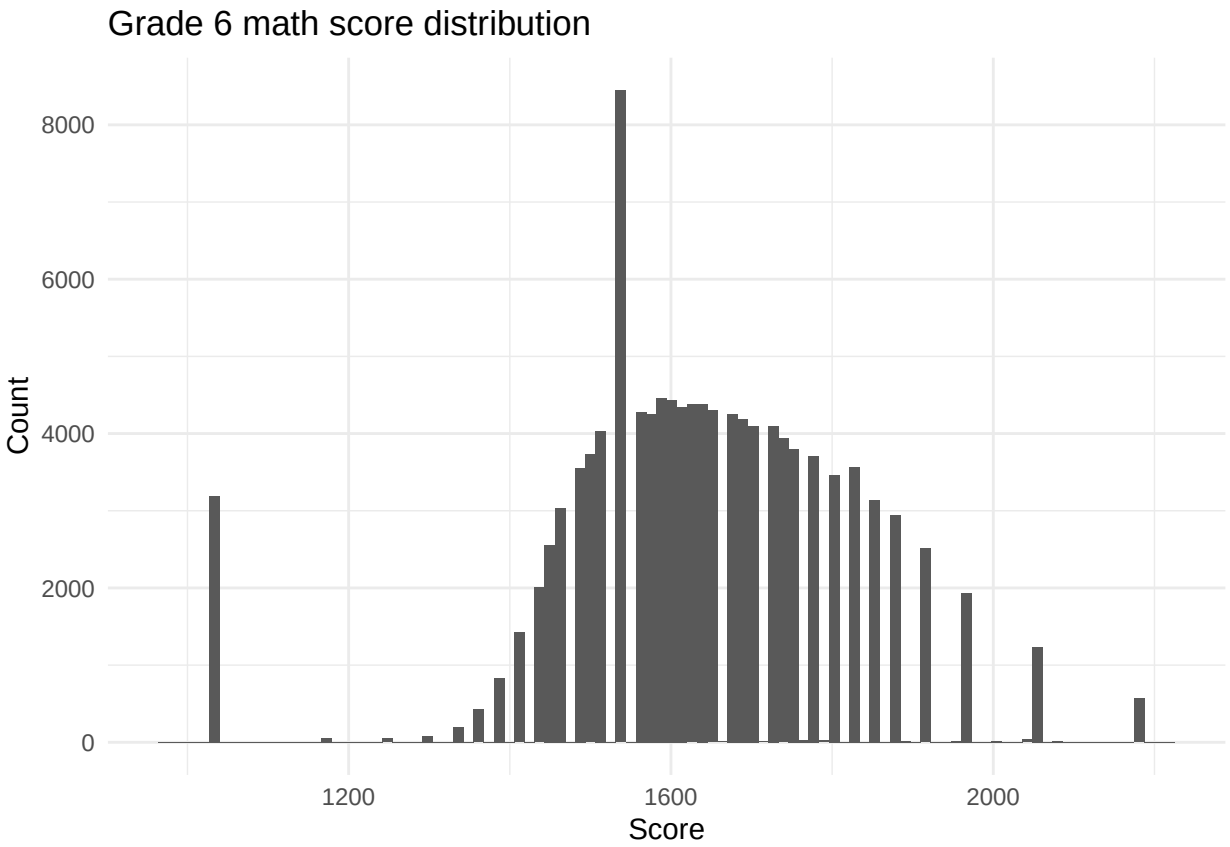
```
data_sampled <- data %>%
  filter(schoolid_nces_enroll_p0 %in% schools_sampled)  %>%
  select(gradelevel, glmath_scr_p0) %>%
  na.omit()
for(i in 3:8){
  p <- data_sampled %>%
    filter(gradelevel == i) %>%
    ggplot(aes(x = glmath_scr_p0)) +
```

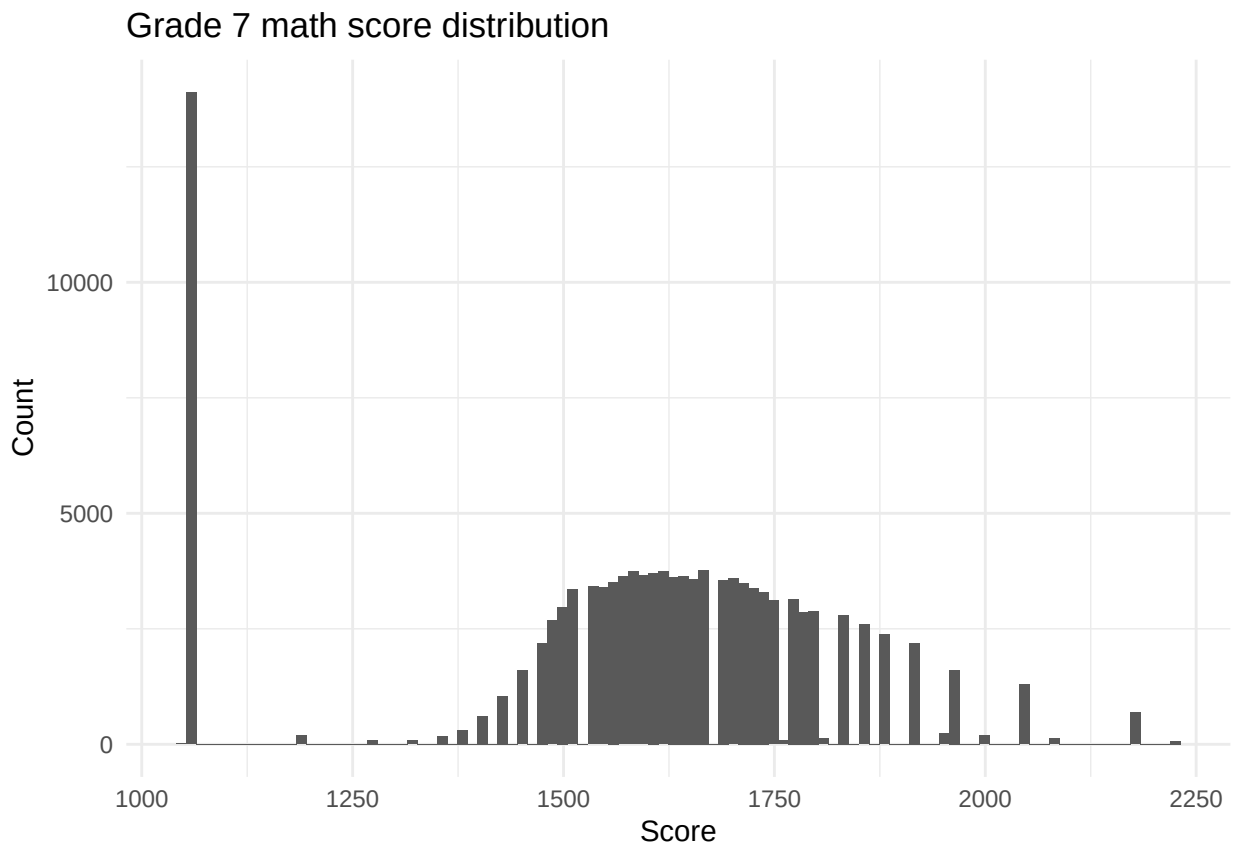
```
    geom_histogram(bins = 100) +
    labs(
      title = paste("Grade", i, "math score distribution"),
      x = "Score",
      y = "Count"
    ) +
    theme_minimal()

  print(p)
}
```

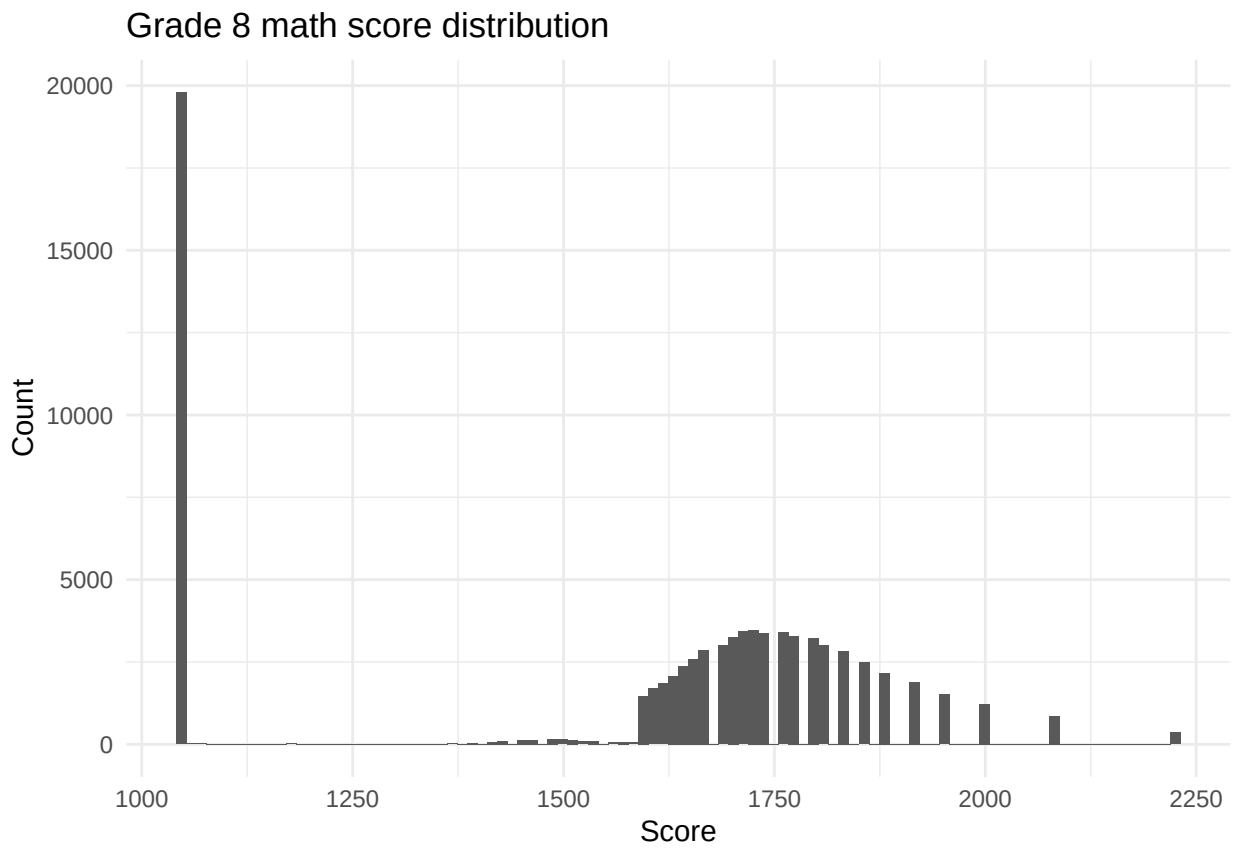



Grade 4 math score distribution
8000
6000
4000
2000
0
Count
900
1200
1500
1800
2100
Score


Grade 5 math score distribution
6000
4000
2000
0
Count
1200
1600
2000
Score


Grade 6 math score distribution
8000
6000
4000
2000
0
Count
1200
1600
2000
Score


Grade 7 math score distribution
Count
10000
5000
0
1000
1250
1500
1750
2000
2250
Score



The unusual high frequency of scores near 1500 for grade 4 and near 1600 for grade 6 students is also worth investigating. Anyway, a more robust way of modeling such math scores is needed, potentially `robustlmm` package in R. But this leaves to future work.