Preliminary ELSA Machine Learnings Tests

Jonathan Seiden, Thu Pham

8/08/2022

Data consolidation process

In this section, we process the data to get it into a format where each row is a child.

Child and Teacher Observation

First we input the child and teacher observations and process them.

For each child in the COP data, we calculate:

1) The average for the child for each of the indicators across sweeps
2) The class average for each indicator *omitting* the child her/himself
3) The class standard deviation for each indicator *omitting* the child her/himself (only including children with 10 or more sweeps)

We then calculate the class average of the TOP indicators for the adults in the class by averaging across sweeps, and merge this data (one to many) with the child-level data. This merged data set contains XXX children in XXX classes

```
#Input the Year One long child and teacher observation data
y1_child_obs_raw <- read.dta13("y1o_c_long.dta")
y1_teacher_obs_raw <- read.dta13("y1o_t_long.dta")
y1_coverpage_obs <- read.dta13("y1o_coverpage.dta")

#Re-format the child data so that it is one row per child

y1_child_obs <- y1_child_obs_raw %>%
  mutate(cid = ifelse(childid == "N/A", o_c_uniqueid, childid)) %>%
  group_by(cid, classid) %>%
  mutate(nsweeps = n()) %>%
  mutate_at(vars(o_c_verbal:o_c_focus), as.character) %>%
  select(c(classid, nsweeps, o_c_verbal:o_c_focus)) %>%
  dummy_cols(select_columns = c("o_c_verbal", "o_c_towhom", "o_c_schedule", "o_c_interaction", "o_c_typ
             remove_selected_columns = TRUE) %>%
  group_by(classid, cid) %>%
  # replaced everything() with nsweeps:last_col()
  summarize(across(nsweeps:last_col(), ~ mean(.x, na.rm = TRUE))) %>%
  filter(nsweeps >= 10 ) %>% #THIS IS AN ARBITRARY PARAMETER
  group_by(classid) %>%
  mutate(nclass = n()) %>%
  # ifelse takes care of the case where there is only one student per class
  mutate(across(starts_with("o_c"), ~
                  (ifelse(get('nclass') == 1, .x, ((sum(.x, na.rm = TRUE) -.x) /
                                                     get('nclass')))), .names =
```

```
                  "{col}_classmean")) %>%
  mutate(across(starts_with("o_c") & !ends_with("classmean"), ~
                  (ifelse(get('nclass') == 1, 0, sqrt((sum((.x - get(str_c(cur_column(),
                                                                           '_classmean')))^2) -
                                                        (.x - get(str_c(cur_column(),
                                                                        '_classmean')))^2) /
                                                      get('nclass')))), .names =
                  "{col}_classsd")) %>%
  ungroup
```

```
## Adding missing grouping variables: `cid`
## `summarise()` has grouped output by 'classid'. You can override using the
## `.groups` argument.
```

```
#Re-format the teacher data so that it is one row per class
y1_teacher_obs <- y1_teacher_obs_raw %>%
  dummy_cols(select_columns = c("o_t_verbal_o", "o_t_whom_o", "o_t_schedule_o",
                                "o_t_task_o", "o_t_instruct", "o_t_focus_o",
                                "o_t_tone_o", "o_t_attention_o", "o_t_es_o"),
             remove_selected_columns = TRUE) %>%
    group_by(classid) %>%
  summarize(
    nsweeps = n(),
    nadult = length(unique(o_t_uniqueid)),
    across(starts_with(c("o_t_verbal_o", "o_t_whom_o", "o_t_schedule_o",
                                "o_t_task_o", "o_t_instruct", "o_t_focus_o",
                                "o_t_tone_o", "o_t_attention_o", "o_t_es_o")),
           ~ mean(.x, na.rm = TRUE))) %>%
  select(-ends_with("_")) %>%
  ungroup

#Merge teacher and child observations
y1_obs <- left_join(y1_child_obs, y1_teacher_obs, by = "classid")
```

```
table(y1_teacher_obs$nadult)
```

```
##
##   1   2   3   4   5
## 131 376 135  25   5
```

```
#Input the Year One long child and teacher observation data
y2_child_obs_raw <- read.dta13("y2o_c_long.dta")
y2_teacher_obs_raw <- read.dta13("y2o_t_long.dta")
y2_coverpage_obs <- read.dta13("y2o_coverpage.dta")

mean(y2_coverpage_obs$classid %in% y1_coverpage_obs$classid)
```

```
## [1] 0.2031008
```

```
#Re-format the child data so that it is one row per child
y2_child_obs <- y2_child_obs_raw %>%
 mutate(cid = ifelse(cid == "N/A", o_c_uniqueid, cid)) %>%
  group_by(cid, classid) %>%
  mutate(nsweeps = n()) %>%
  mutate_at(vars(o_c_verbal:o_c_focus), as.character) %>%
  select(c(classid, nsweeps, o_c_verbal:o_c_focus)) %>%
```

```
  dummy_cols(select_columns = c("o_c_verbal", "o_c_towhom", "o_c_schedule", "o_c_interaction", "o_c_type
             remove_selected_columns = TRUE) %>%
  group_by(classid, cid) %>%
  # replaced everything() with nsweeps:last_col()
  summarize(across(nsweeps:last_col(), ~ mean(.x, na.rm = TRUE))) %>%
  filter(nsweeps >= 10 ) %>% #THIS IS AN ARBITRARY PARAMETER
  group_by(classid) %>%
  mutate(nclass = n()) %>%
    # ifelse takes care of the case where there is only one student per class
  mutate(across(starts_with("o_c"), ~
                 (ifelse(get('nclass') == 1, .x, ((sum(.x, na.rm = TRUE) -.x) /
                                                    get('nclass')))), .names =
                 "{col}_classmean")) %>%
  mutate(across(starts_with("o_c") & !ends_with("classmean"), ~
                 (ifelse(get('nclass') == 1, 0, sqrt((sum((.x - get(str_c(cur_column(),
                                                                          '_classmean')))^2) -
                                                       (.x - get(str_c(cur_column(),
                                                                       '_classmean')))^2) /
                                                     get('nclass')))), .names =
                 "{col}_classsd")) %>%
  ungroup
```

```
## Adding missing grouping variables: `cid`
## `summarise()` has grouped output by 'classid'. You can override using the
## `.groups` argument.
```

```
#Re-format the teacher data so that it is one row per class
y2_teacher_obs <- y2_teacher_obs_raw %>%
  dummy_cols(select_columns = c("o_t_verbal_o", "o_t_whom_o", "o_t_schedule_o",
                                "o_t_task_o", "o_t_instruct", "o_t_focus_o",
                                "o_t_tone_o", "o_t_attention_o", "o_t_es_o"),
             remove_selected_columns = TRUE) %>%
    group_by(classid) %>%
  summarize(
    nsweeps = n(),
    nadult = length(unique(o_t_uniqueid)),
    across(starts_with(c("o_t_verbal_o", "o_t_whom_o", "o_t_schedule_o",
                                "o_t_task_o", "o_t_instruct", "o_t_focus_o",
                                "o_t_tone_o", "o_t_attention_o", "o_t_es_o")),
           ~ mean(.x, na.rm = TRUE))) %>%
  select(-ends_with("_")) %>%
  ungroup

#Extract the caretype from the observation sheet
## ASK -- what is meant by this?

#Merge teacher and child observations
y2_obs <- left_join(y2_child_obs, y2_teacher_obs, by = "classid")
```

Below we now input the child-level outcome data. We focus on the outcomes that Emily suggested, and extract the year 1 and year 2 values for each child and then merge to create a single dataset.

```
#Get Year 1 and Year 2 child data
y1_child_outcomes_raw <- read.dta13("y1c.dta")
y2_child_outcomes_raw <- read.dta13("y2c.dta")
```

```
#Rename all y1 variables and y2 variables so we don't lose them when merging
y1_child_outcomes <-  y1_child_outcomes_raw %>%
  select(cid, c_mefs_str, c_pt_pcorrect, c_ltr_cogsoc_comp, c_ltr_emo_comp,
         c_pra_total, c_pbsa_total, c_quils_total_raw, c_wjlw_str, c_wjap_str) %>%
  rename_all( ~ paste0("y1_", .x)) %>%
  mutate(cid = as.character(y1_cid))

y2_child_outcomes <-  y2_child_outcomes_raw %>%
  select(cid, c_mefs_str, c_pt_pcorrect, c_ltr_cogsoc_comp, c_ltr_emo_comp,
         c_pbsa_allgrades_total, c_pra_allgrades_total, c_quils_total_raw,
         c_wjlw_str, c_wjap_str, c_age_cat_test, c_age_test) %>%
  rename(c_pra_total = c_pra_allgrades_total,
         c_pbsa_total = c_pbsa_allgrades_total) %>%
  rename_all( ~ paste0("y2_", .x)) %>%
  mutate(cid = as.character(y2_cid))

#Merge Y2 and Y1 data together and calculate the gain score for each of the outcomes
child_outcomes <- merge(y1_child_outcomes, y2_child_outcomes, by = "cid") %>%
  mutate(gain_c_mefs_str = y2_c_mefs_str - y1_c_mefs_str,
         gain_c_pt_pcorrect = y2_c_pt_pcorrect - y1_c_pt_pcorrect,
         gain_c_ltr_cogsoc_comp = y2_c_ltr_cogsoc_comp - y1_c_ltr_cogsoc_comp,
         gain_c_ltr_emo_comp = y2_c_ltr_emo_comp - y1_c_ltr_emo_comp,
         gain_c_pra_total = y2_c_pra_total - y1_c_pra_total,
         gain_c_pbsa_total = y2_c_pbsa_total - y1_c_pbsa_total,
         gain_c_quils_total_raw = y2_c_quils_total_raw - y1_c_quils_total_raw,
         gain_c_wjlw_str = y2_c_wjlw_str - y1_c_wjlw_str,
         gain_c_wjap_str = y2_c_wjap_str - y1_c_wjap_str,
         cid = as.numeric(cid))
```

Finally, we merge together the Year 1 observation data with the Year 1 & 2 child outcome data and add in care type. We omit observations that have no classroom observation resulting in a total analytic dataframe of 1169 observations of 64 variables.

```
#Merge in the outcomes data

y1_obs <- y1_obs %>%
  filter(!is.na(cid)) %>%
  mutate(cid = as.numeric(cid))
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
outcomes_and_obs_y1 <- left_join(child_outcomes, y1_obs, by = "cid") %>%
  mutate(cid = as.character(cid))

#Add in the care type
caretype <- read.dta13("y1caretype.dta") %>%
  mutate(cid = as.character(cid))

#Remove observations that have no care type or no classroom observation
outcomes_and_obs_full_y1 <- left_join(outcomes_and_obs_y1, caretype, by = "cid") %>%
  mutate(hasobservation = is.na(classid)) %>%
  filter(!is.na(caretype)) %>%
  filter(!is.na(classid))

# outcomes_and_obs_full_y1 <- outcomes_and_obs %>%
```

```
#   filter(!is.na(classid))

#Remove Y1 and Y2 data for cleanliness
outcomes_and_obs_full_y1 <- outcomes_and_obs_full_y1 %>%
  select(-starts_with(c("y1", "y2")))

#Remove some irrelevant variables
outcomes_and_obs_full_y1 <- outcomes_and_obs_full_y1 %>%
  select(-c(famid, dob:dob_uncertain, actual_fcc:hasobservation)) %>%
  mutate(caretype = as.factor(caretype),
         actualtype = as.factor(actualtype))
```

```
# replaces missing data with mean (if numeric) and mode (if factor)

getmode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

replace.na <- function(var){
  ifelse(is.na(var) | is.nan(var),
         ifelse(is.factor(var), getmode(var), mean(var)), var)
}
```

```
# names(outcomes_and_obs_full_y1)
outcomes_and_obs_full_y1 <- outcomes_and_obs_full_y1 %>%
  mutate_at(vars("o_c_verbal_Fuss/Cry (FC)":actualtype), replace.na)

# what is o_t_sched_t supposed to be for? replacing with actualtype for now just to run analysis

# replacing all NaNs in hopes that it will fix the model matrix issue
# originally was o_c_verbal_talk:o_t_sched_t

dim(outcomes_and_obs_full_y1)
```

```
## [1] 1169  284
```

Analysis

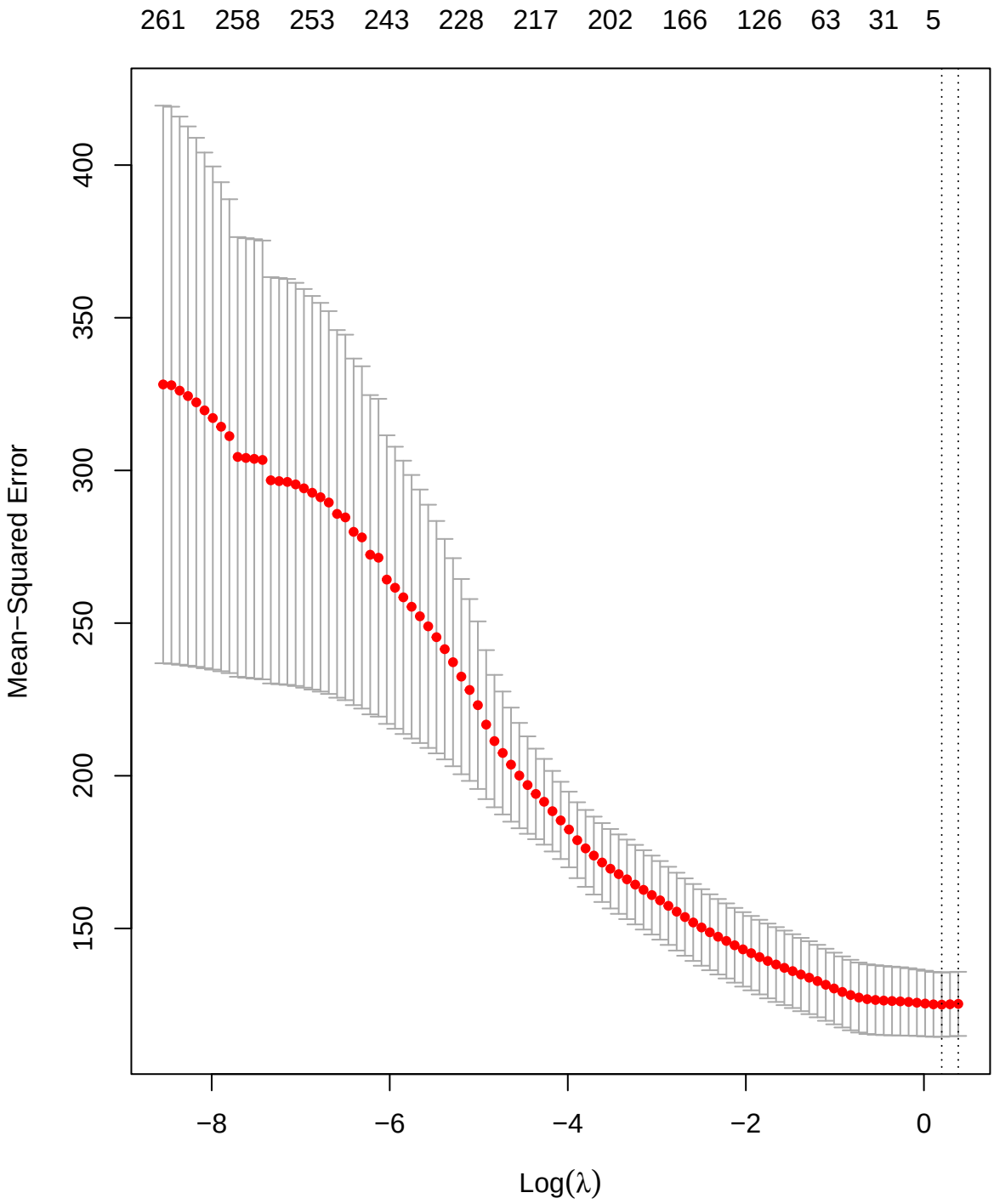
We will first try a cross-validated LASSO, which will aggressively remove variables that do little to improve the predictive accuracy of the model.

```
# looping through gain variables

# initiate two lists to store results of each model (for the graph and the coefficients)

# https://stackoverflow.com/questions/9332417/whats-a-good-way-to-store-multiple-models-in-an-r-data-st
set.seed(4224)

gain_ind <- which(startsWith(colnames(outcomes_and_obs_full_y1), "gain"))

models_y1 <- list()
coefs_y1 <- list()
# double check that the number of rows matches up with number of non-NAs
for (i in gain_ind) {
```

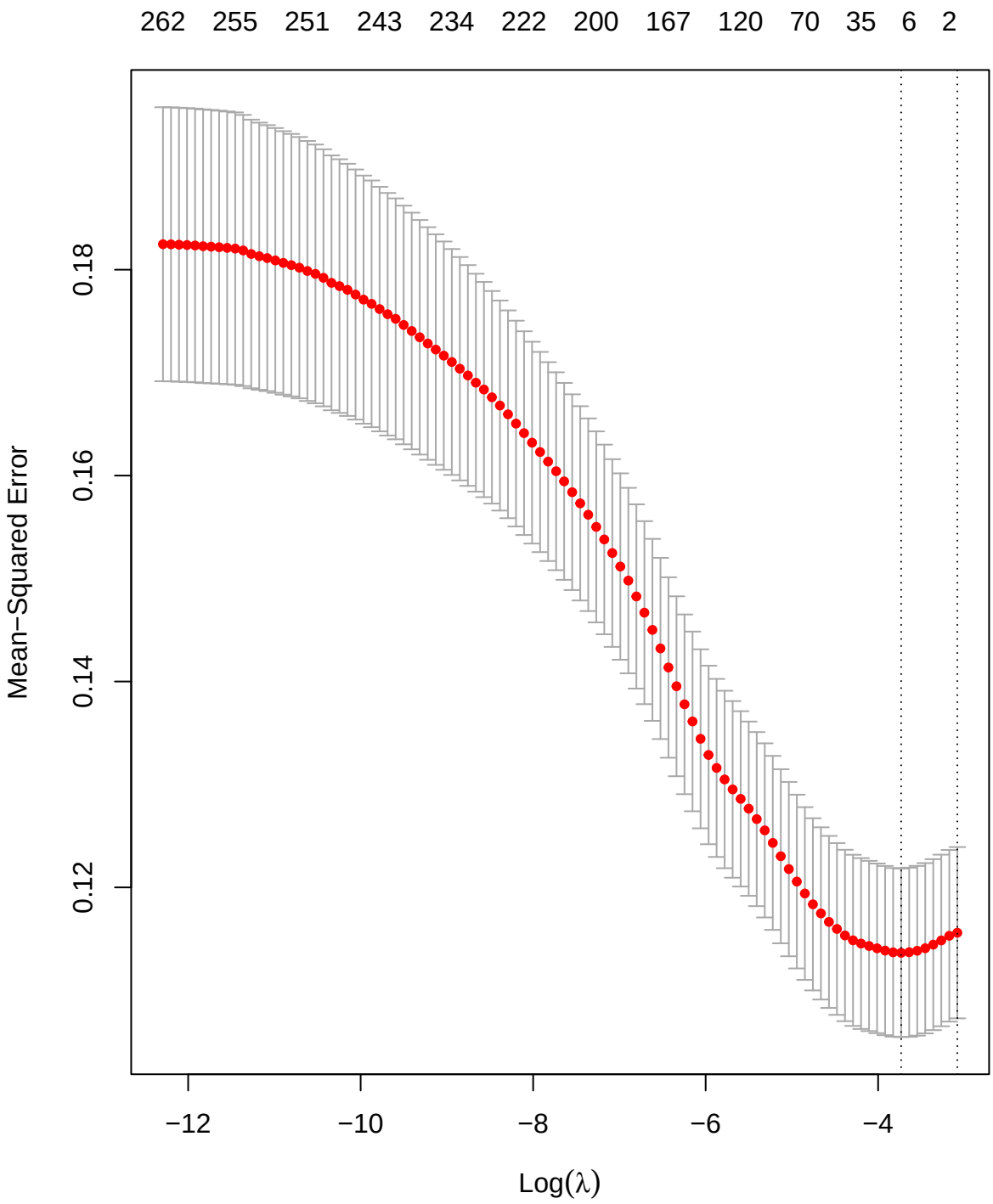
```
  name <- names(outcomes_and_obs_full_y1)[i]
  df_analysis <- outcomes_and_obs_full_y1 %>%
    filter(!is.na(outcomes_and_obs_full_y1[[name]])) %>%
    select(c(name, "o_c_verbal_Fuss/Cry (FC)":actualtype))
  print(name)
  print(sum(!is.na(outcomes_and_obs_full_y1[[name]])))
  # double check that this is okay
  # options(na.action="na.pass")
  x = model.matrix(as.formula(paste(name, "~ .")), data = df_analysis)

  y = df_analysis[[name]]
  print(dim(x))
  x = x[, -1]


  # call cv.glmnet()
  model_lasso <- cv.glmnet(x = x, y = y, alpha = 1)
  plot(model_lasso)

  models_y1[[name]] <- model_lasso

  cc = coef(model_lasso, s = model_lasso$lambda.min)

  # print out the model coefficients and store in a list.
  cc = cc[cc[,1]!=0,1]
  coefs_y1[[name]] <- cc
  print(cc)

}
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(name)` instead of `name` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
## [1] "gain_c_mefs_str"
## [1] 726
## [1] 726 273
```

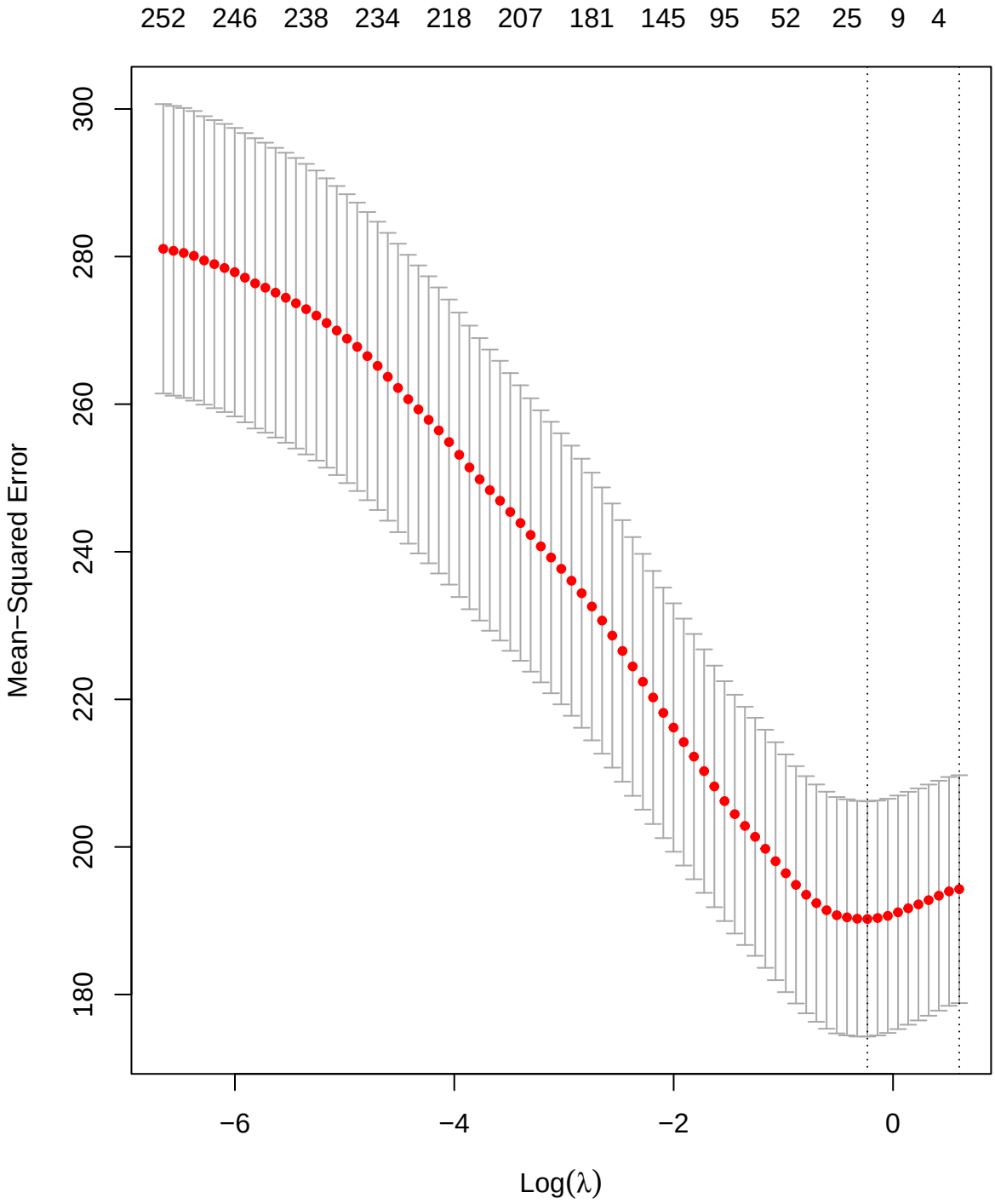



```
##                           (Intercept)                  `o_c_verbal_No (N)`
##                           2.173678878                          0.221169310
##      `o_c_towhom_No talk/listen (NT)`     `o_c_interaction_Unoccupied (UN)`
##                           0.001092019                          3.364308156
##               `o_c_typetask_None (N)` `o_c_schedule_Playground (P)_classsd`
##                           0.744365466                        -13.154955132
## [1] "gain_c_pt_pcorrect"
## [1] 845
## [1] 845 273
```



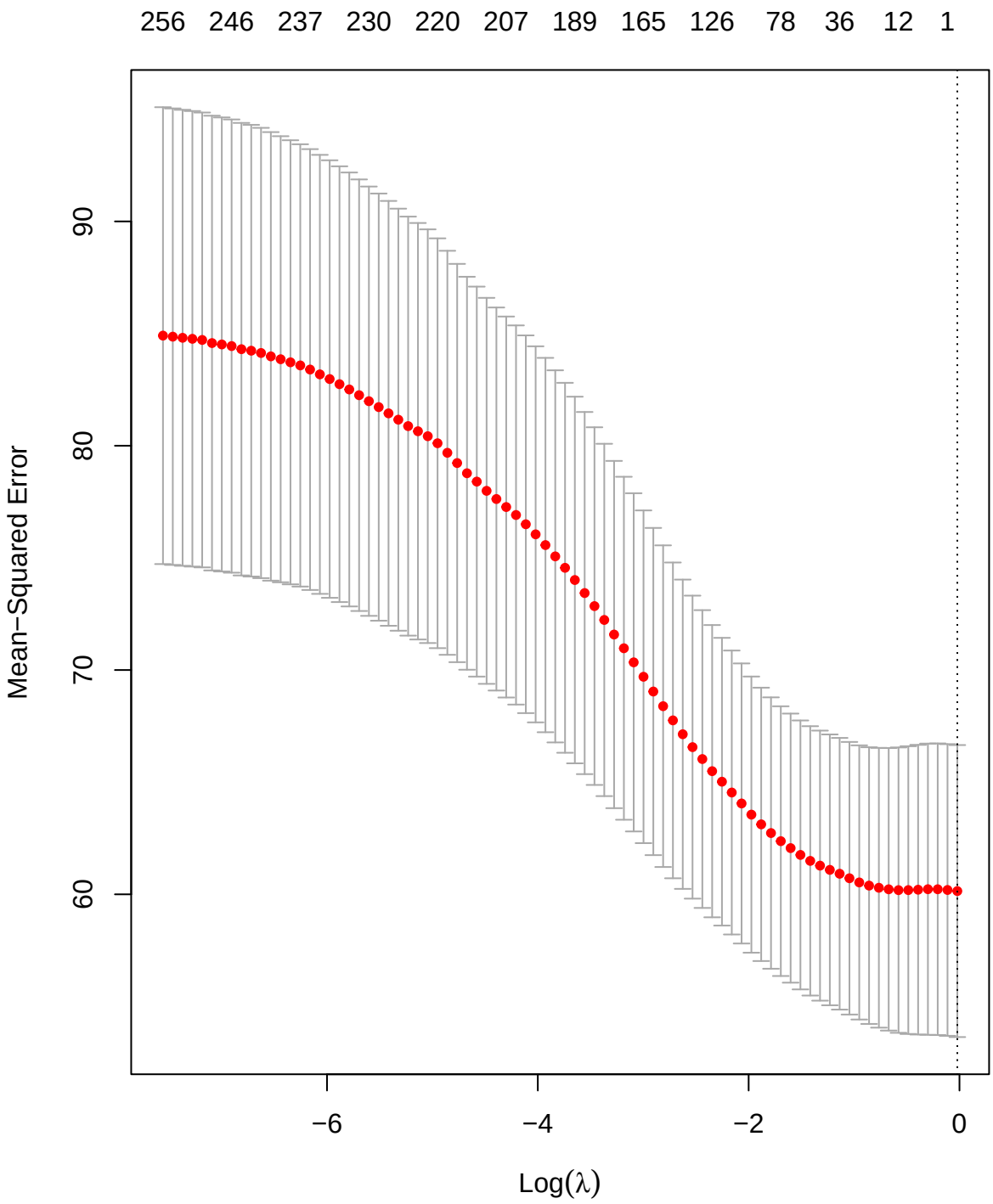

```
##                                  (Intercept)
##                                0.3052493192
##                  `o_c_verbal_Listening (L)`
##                                -0.0459336520
##                                       nclass
##                                -0.0006026907
##          `o_c_verbal_Listening (L)_classmean`
##                                -0.0055634594
##       `o_c_focus_Other Language (OL)_classmean`
```

```
##                                      4.2481342277
## `o_c_towhom_Whole Group, No Teacher (WG)_classsd`
##                                      0.1853180202
##            `o_c_focus_Language Arts (LA)_classsd`
##                                     -0.0523185360
##       `o_c_focus_Literacy - Writing (LW)_classsd`
##                                     -0.9904268094
##                       `o_c_focus_Math (M)_classsd`
##                                     -0.0407739367
##                                     o_t_whom_o_WG
##                                      0.0059724025
##                                     o_t_task_o_MA
##                                     -0.1351439694
##                                    o_t_focus_o_MM
##                                      0.2820775333
## [1] "gain_c_ltr_cogsoc_comp"
## [1] 992
## [1] 992 273
```

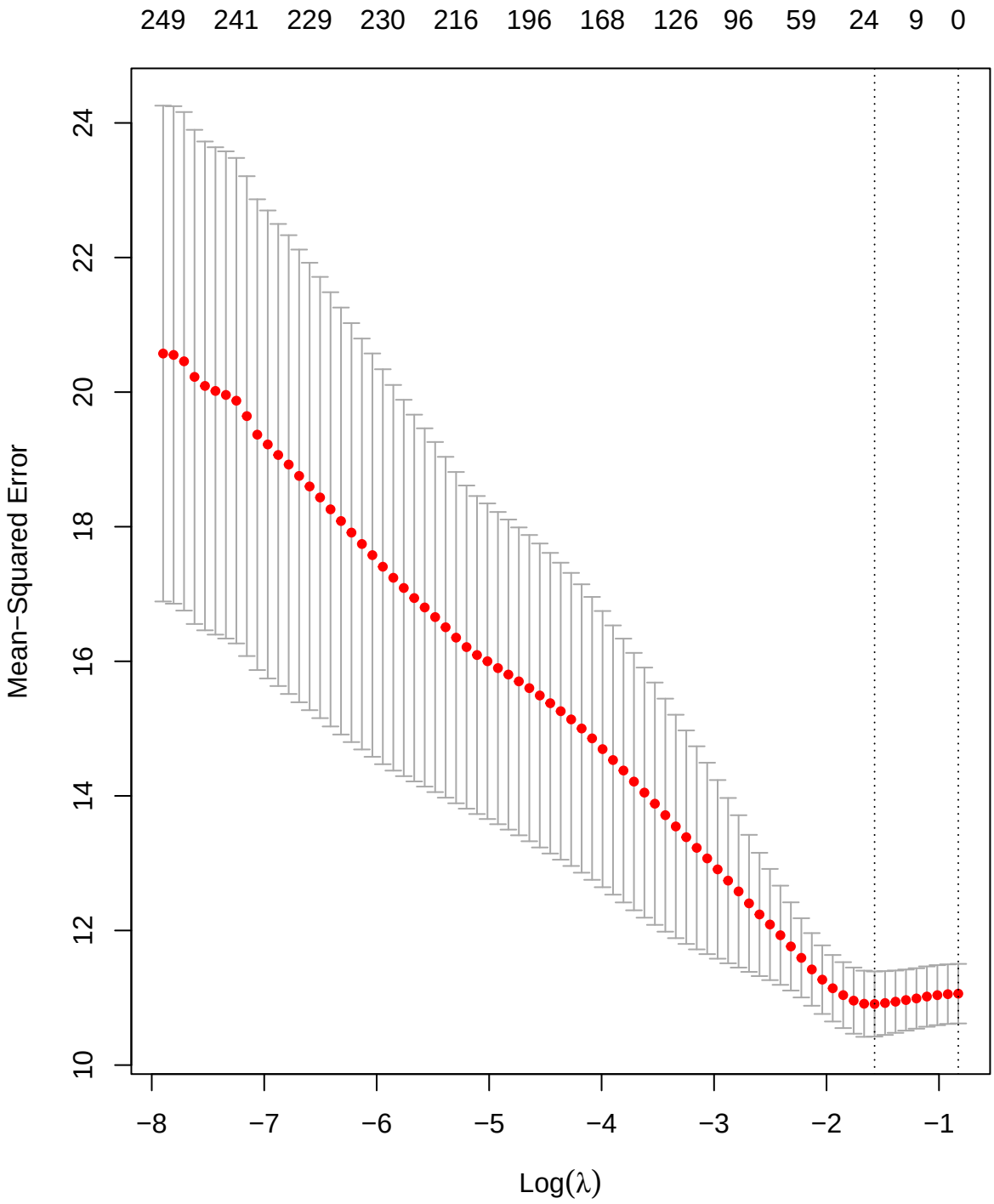



```
##                                                 (Intercept)
##                                                   1.4161757
##                                `o_c_verbal_Fuss/Cry (FC)`
##                                                 -17.7609386
##                  `o_c_towhom_Whole Group, No Teacher (WG)`
##                                                   2.1151848
##                               `o_c_schedule_Meal Time (MT)`
##                                                  -0.4642319
##                             `o_c_interaction_Time Out (TO)`
```

```
##                                                    -25.1202456
##                                 `o_c_typetask_Sequential (SQ)`
##                                                     -0.8077267
##               `o_c_towhom_Whole Group, No Teacher (WG)_classmean`
##                                                     31.0552407
##                     `o_c_involvement_Medium High (MH)_classmean`
##                                                      5.4951417
##                                      `o_c_verbal_Talk (T)_classsd`
##                                                     -7.7373801
## `o_c_schedule_Combination of Small Group and Centers (SGC)_classsd`
##                                                     12.1845513
##                                   `o_c_focus_Literacy (L)_classsd`
##                                                     -6.8335190
##                                                   o_t_whom_o_WG
##                                                      5.2697447
##                                                  o_t_instruct_4
##                                                    183.9629565
##                                                   o_t_focus_o_A
##                                                     13.2264235
##                                                  o_t_focus_o_LA
##                                                      3.4019861
##                                                     o_t_es_o_N
##                                                     -1.8451693
## [1] "gain_c_ltr_emo_comp"
## [1] 992
## [1] 992 273
```

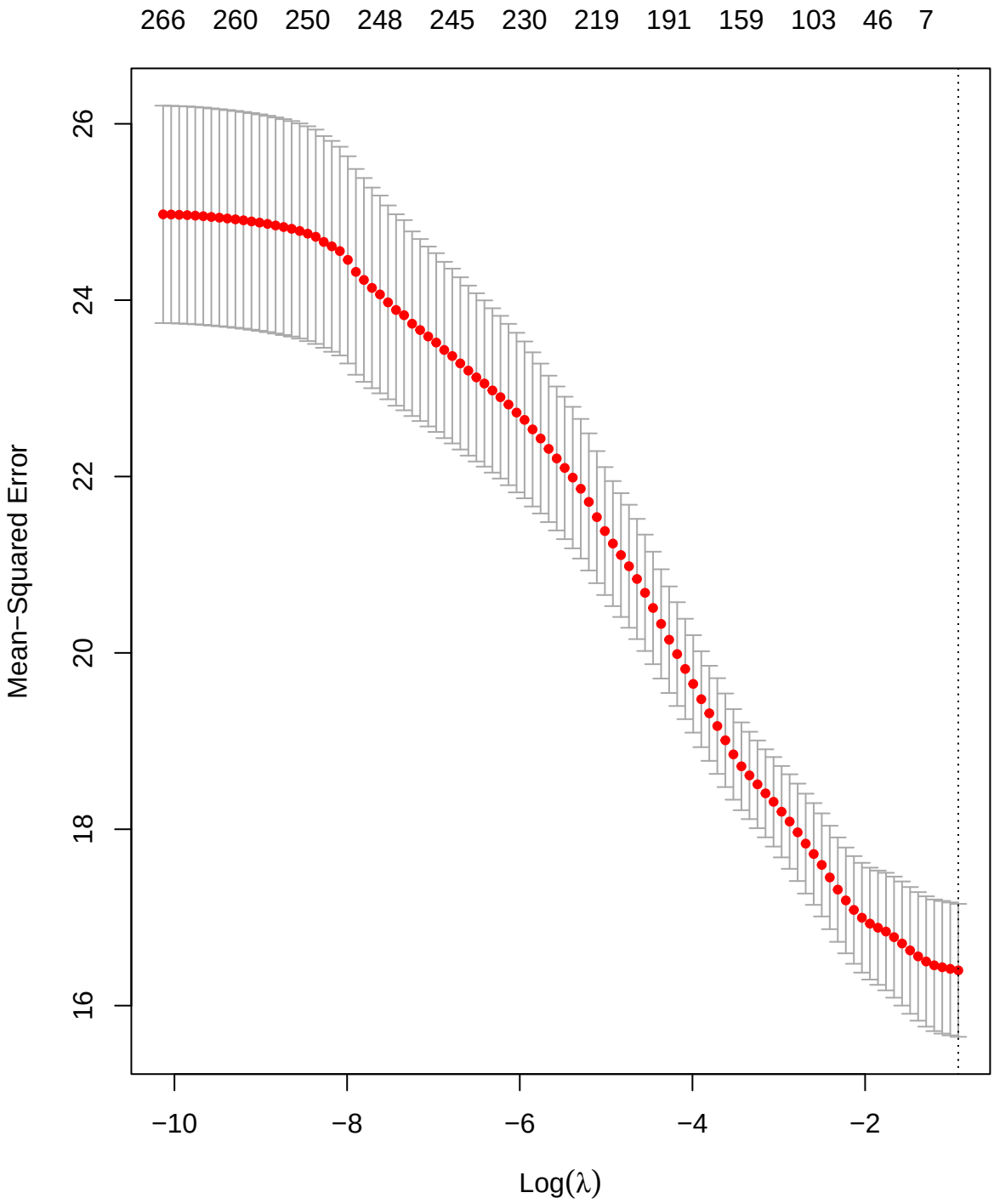



```
## [1] 0.6048387
## [1] "gain_c_pra_total"
## [1] 873
## [1] 873 273
```

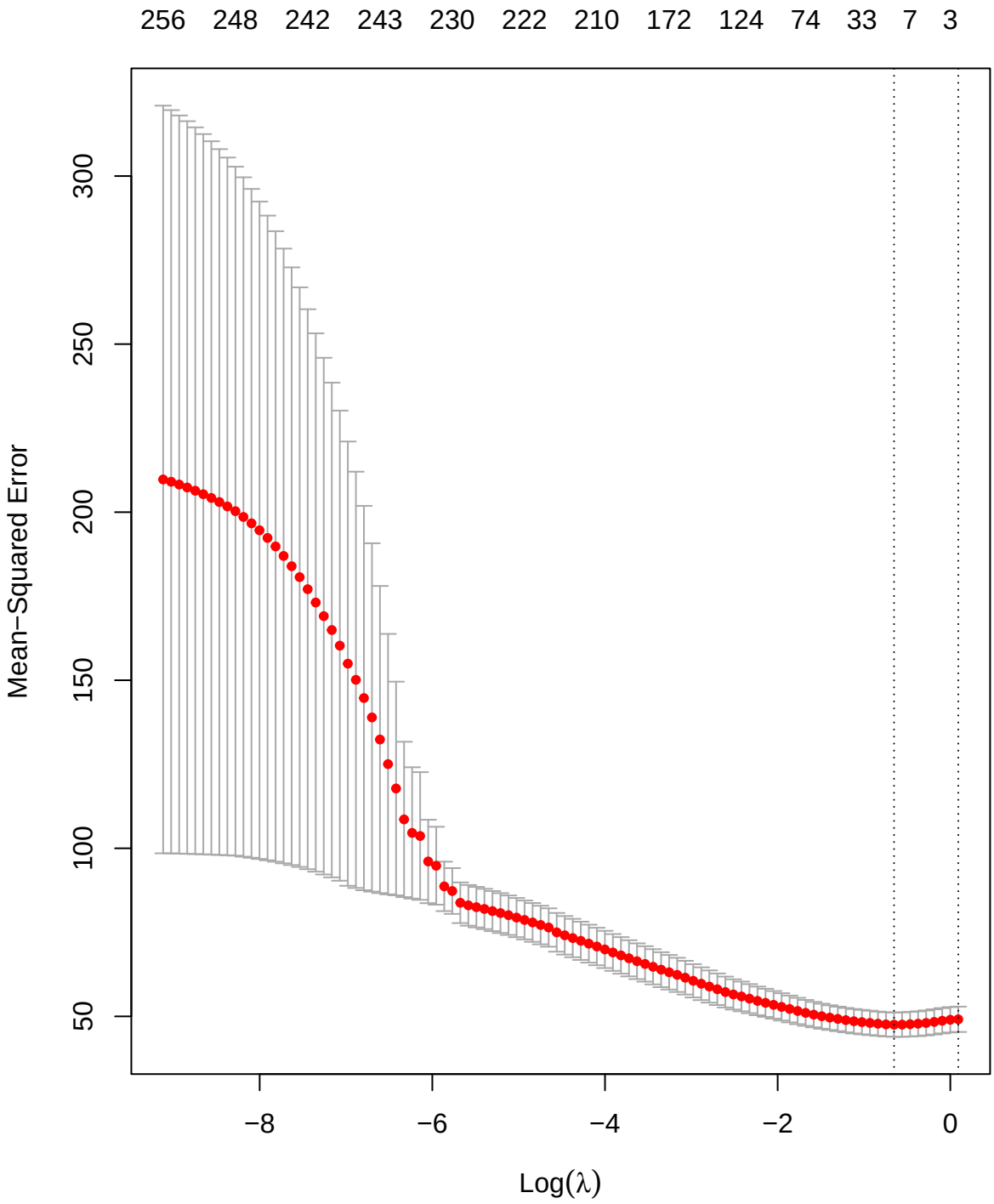



```
##                                   (Intercept)
##                                    3.53315067
##         `o_c_towhom_Whole Group, No Teacher (WG)`
##                                    3.31374180
##                       `o_c_interaction_Alone (AL)`
##                                   -0.12821727
##                           `o_c_typetask_None (N)`
##                                    1.68921434
##                       `o_c_focus_Gross Motor (GM)`
```

```
##                                       -0.49475305
##                                `o_c_focus_Math (M)`
##                                       -0.16109778
##              `o_c_verbal_Listening (L)_classmean`
##                                       -0.47228926
## `o_c_towhom_Child (C): One other child_classmean`
##                                       -1.47793536
##      `o_c_interaction_Associative (AS)_classmean`
##                                       -1.13489454
##       `o_c_interaction_Cooperative (C)_classmean`
##                                       -2.50188111
##      `o_c_involvement_Medium High (MH)_classmean`
##                                       -0.97150006
##     `o_c_focus_Literacy - Writing (LW)_classmean`
##                                       -1.61965602
##                    `o_c_focus_Math (M)_classmean`
##                                       -0.02760322
##                              o_c_focus_NA_classmean
##                                      246.89104580
##         `o_c_interaction_Cooperative (C)_classsd`
##                                       -0.48343072
##       `o_c_interaction_Non-Academic (NA)_classsd`
##                                       -1.69647491
##            `o_c_interaction_Parallel (P)_classsd`
##                                        0.85640793
##                         o_c_interaction_NA_classsd
##                                       -6.91981028
##        `o_c_involvement_Medium High (MH)_classsd`
##                                       -1.07076624
## `o_c_focus_Language Arts - Writing (LAW)_classsd`
##                                        0.56805682
##       `o_c_focus_Literacy - Writing (LW)_classsd`
##                                       -2.09035333
##                       `o_c_focus_Math (M)_classsd`
##                                       -0.38630437
##                                     o_t_focus_o_M
##                                       -0.16405547
## [1] "gain_c_pbsa_total"
## [1] 882
## [1] 882 273
```

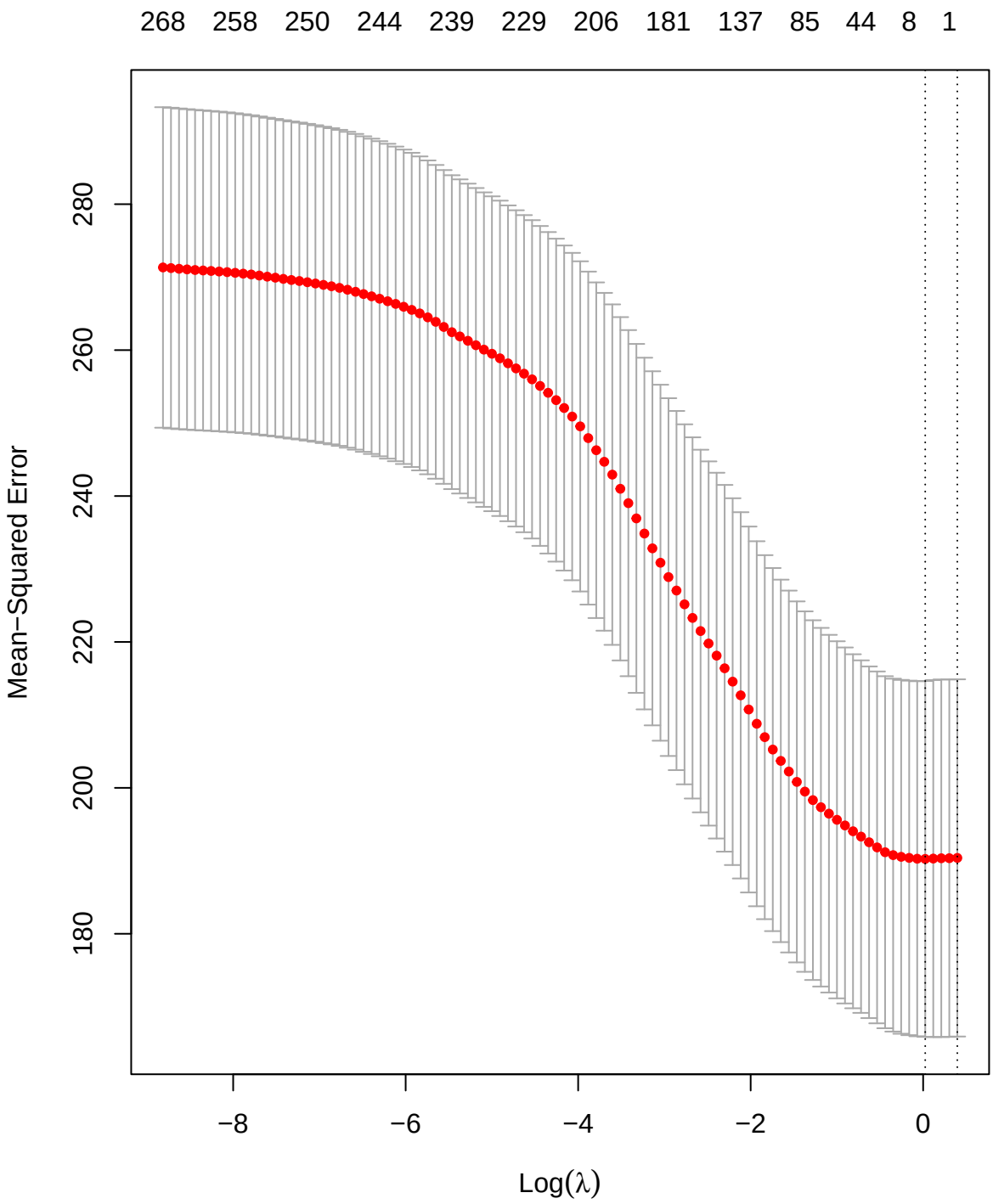



```
## [1] 4.002268
## [1] "gain_c_quils_total_raw"
## [1] 598
## [1] 598 273
```

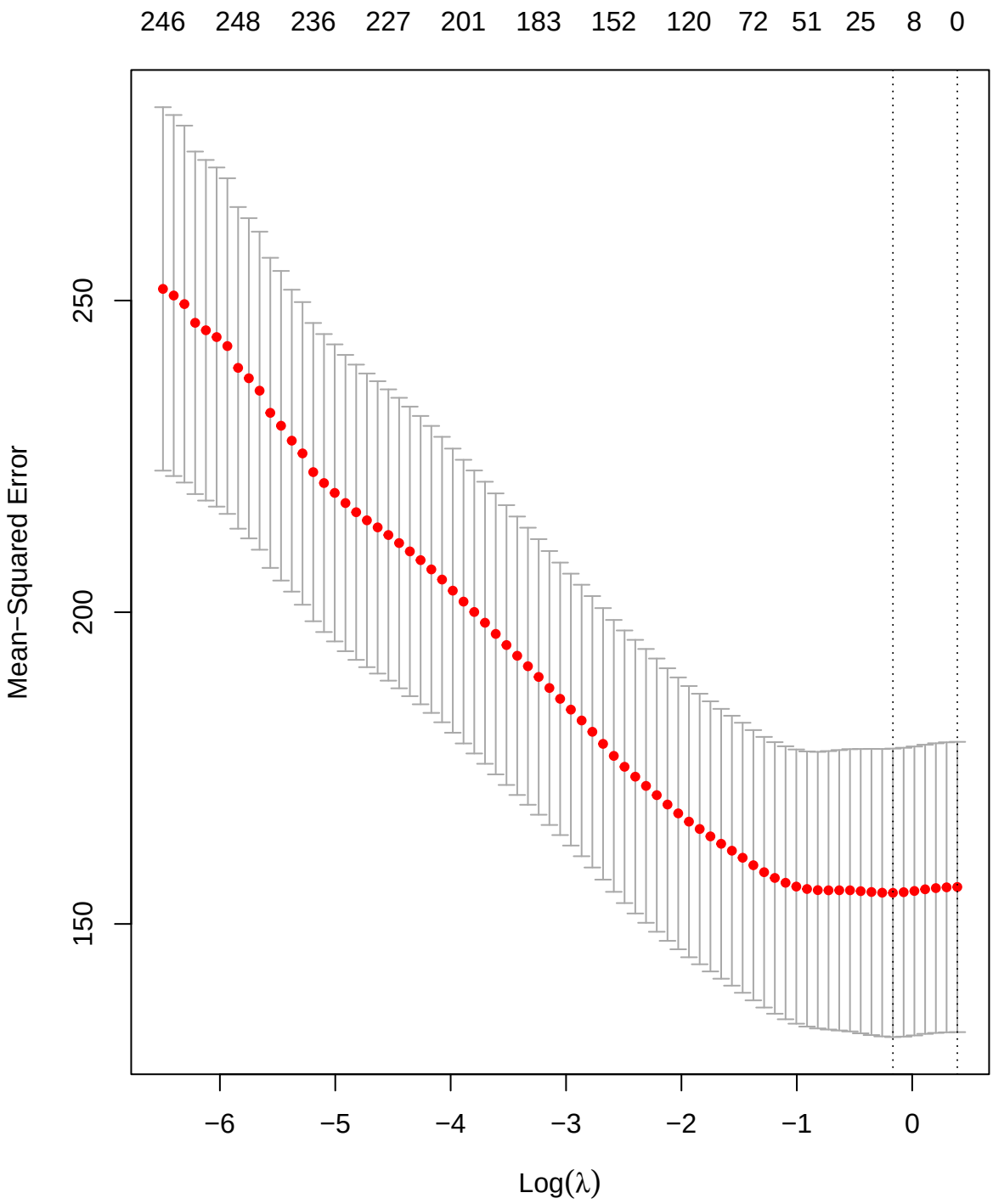



```
##                                (Intercept)
##                                  9.88256166
##                       `o_c_verbal_No (N)`
##                                  0.99034318
##                       `o_c_focus_Art (A)`
##                                  0.27686281
##                  `o_c_focus_Literacy (L)`
##                                 -8.87783847
##                      `o_c_focus_Math (M)`
```

```
##                                       -0.27749697
##              `o_c_verbal_Listening (L)_classmean`
##                                       -0.25690251
## `o_c_towhom_Child (C): One other child_classmean`
##                                       -4.26085286
##       `o_c_focus_Literacy - Writing (LW)_classsd`
##                                      -29.81522243
##                                    o_t_whom_o_WGT
##                                       -7.30061756
##                                     o_t_task_o_PC
##                                        0.80159404
##                                        o_t_es_o_N
##                                       -0.85522749
##                                             c_HHS
##                                       -0.03029131
##                                          caretype
##                                       -0.05881235
## [1] "gain_c_wjlw_str"
## [1] 963
## [1] 963 273
```



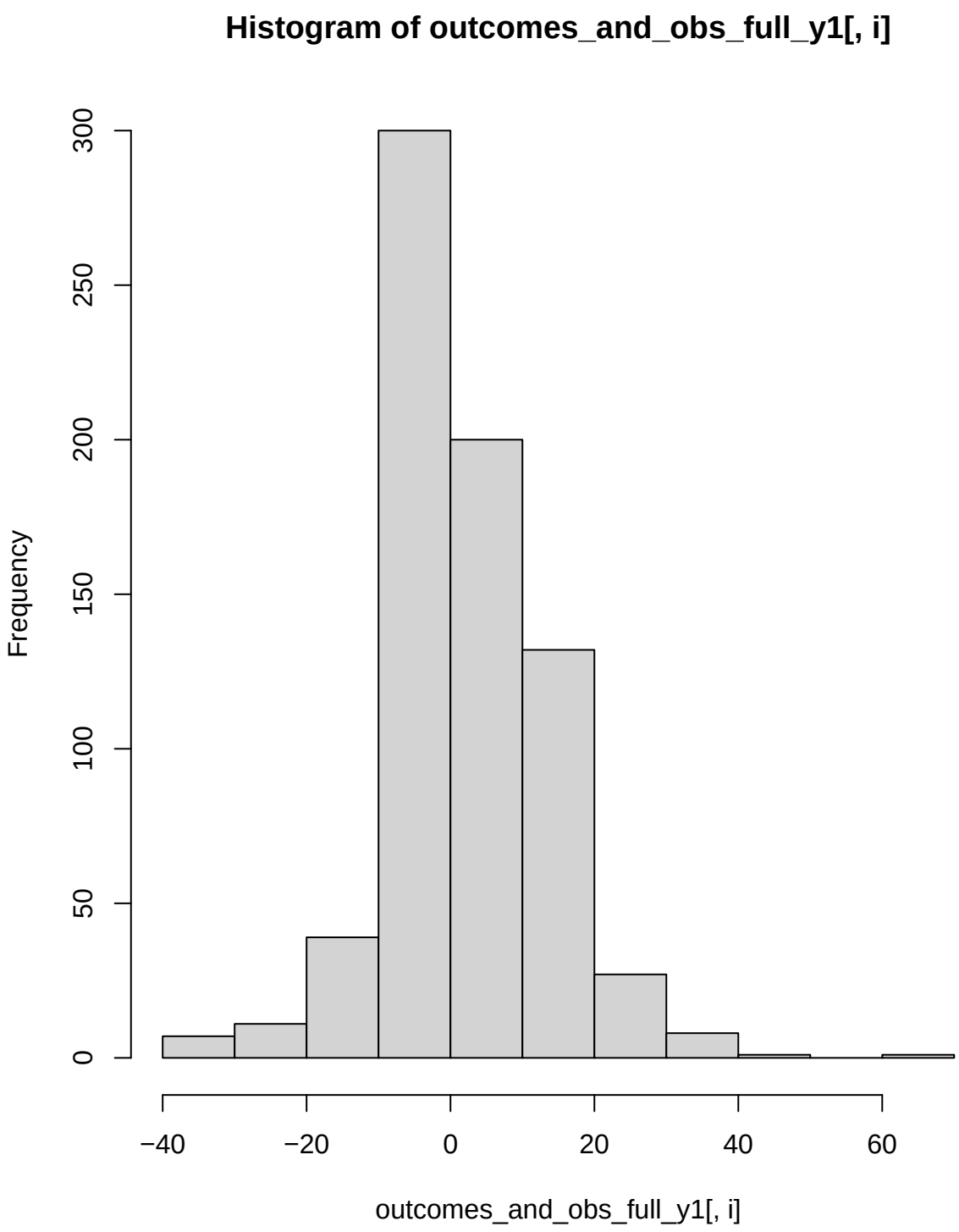

```
##                                (Intercept)          `o_c_interaction_Onlooker (ON)`
##                                3.223785223                               -0.454364262
##                     `o_c_typetask_None (N)`                                    nclass
##                                -1.248977404                                0.003590305
## `o_c_schedule_Playground (P)_classmean`           `o_c_schedule_Gym (G)_classsd`
##                                -1.803445789                              130.948094825
## [1] "gain_c_wjap_str"
## [1] 930
## [1] 930 273
```

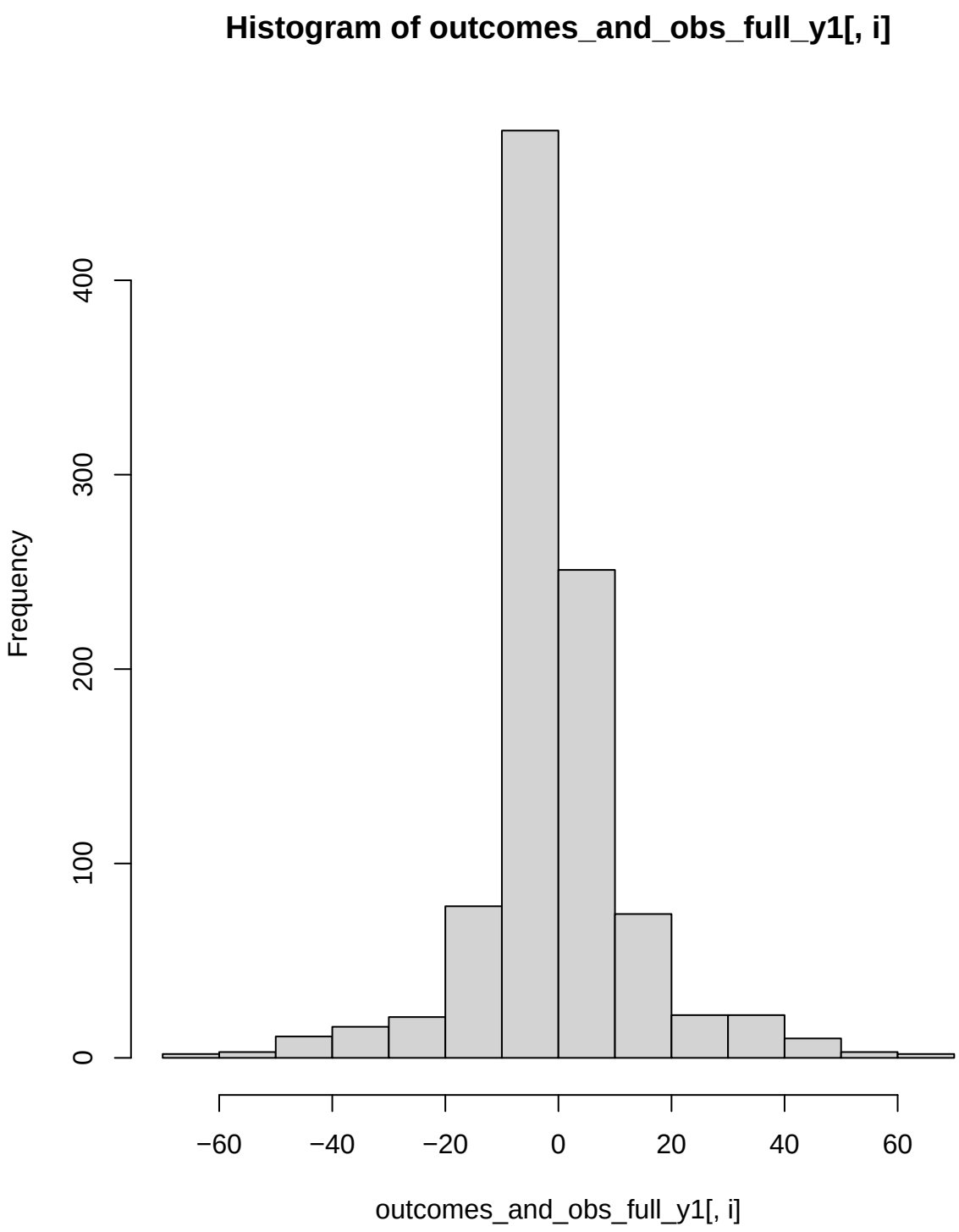


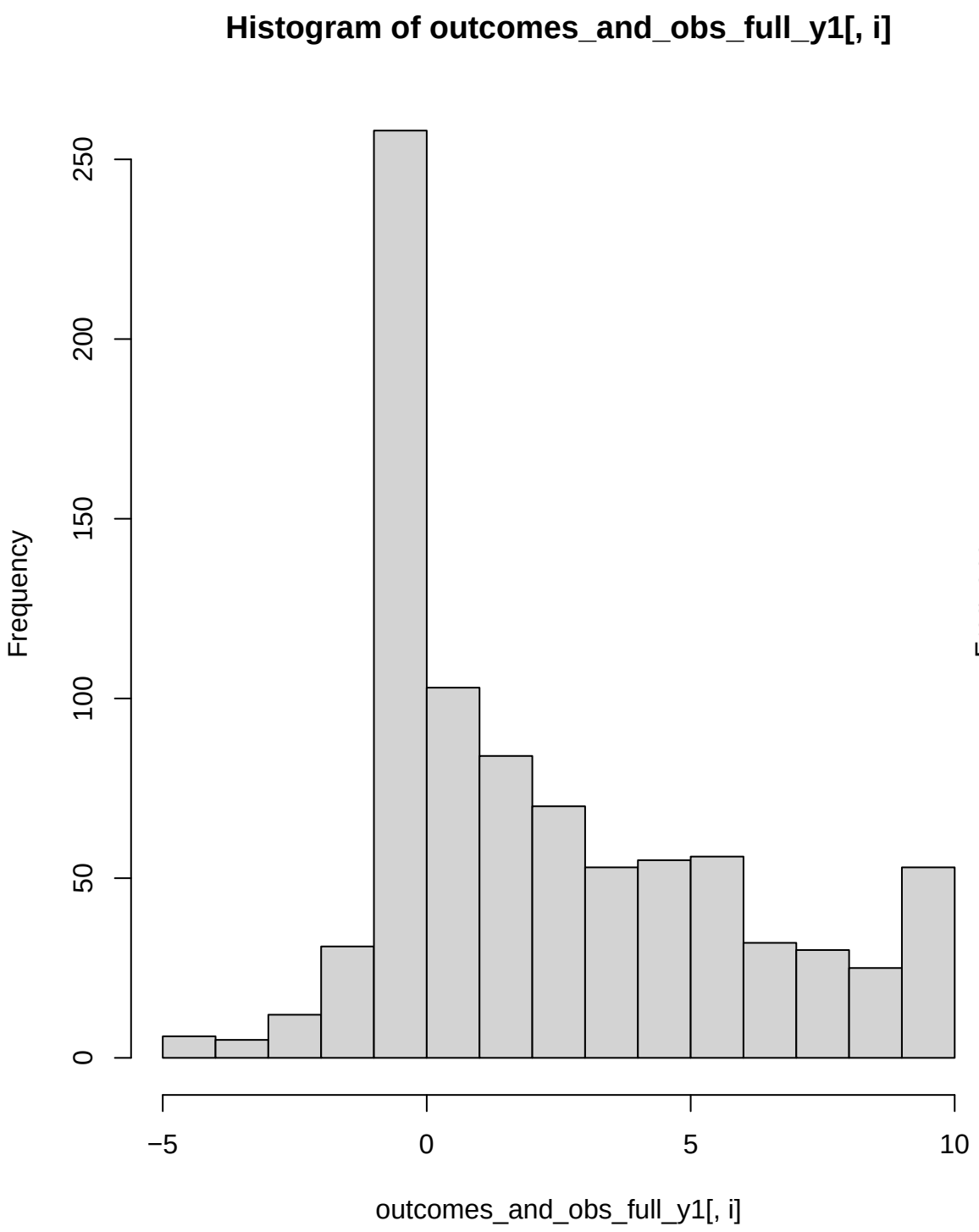

```
##                                        (Intercept)
##                                         -1.7118845
##                             `o_c_verbal_Fuss/Cry (FC)`
##                                          3.9152671
## `o_c_towhom_Teacher (T): Teacher or assistant (adult)`
##                                         -0.9543685
##                           `o_c_typetask_Disruptive (D)`
##                                          1.9997722
##              `o_c_schedule_Small Groups (SG)_classmean`
```

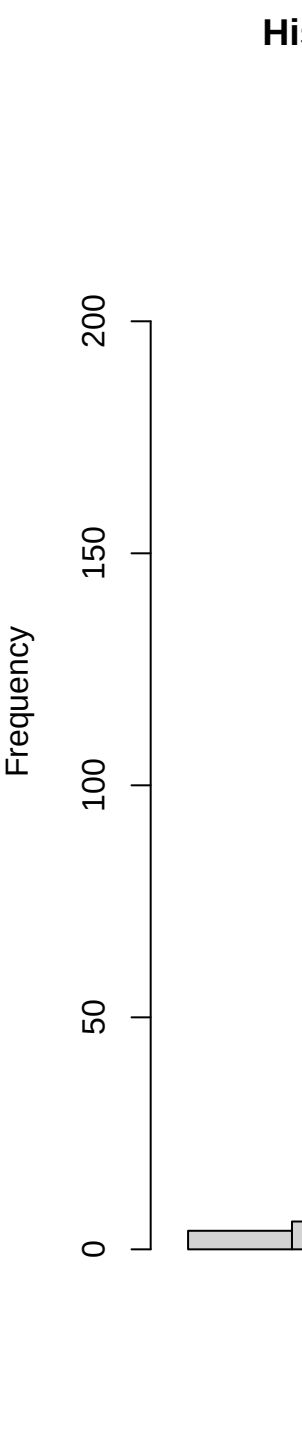
```
##                                           -0.2748754
##                       `o_c_focus_Literacy (L)_classmean`
##                                           -7.3312070
##                    `o_c_schedule_Meal Time (MT)_classsd`
##                                           22.4575001
##              `o_c_focus_Music and Movement (MM)_classsd`
##                                           18.5014581
##                                        o_t_task_o_MO
##                                            2.5417554
##                                       o_t_instruct_4
##                                         -136.8327412
##                                           o_t_es_o_A
##                                           11.5824504
##                                           o_t_es_o_I
##                                           61.4676665
```

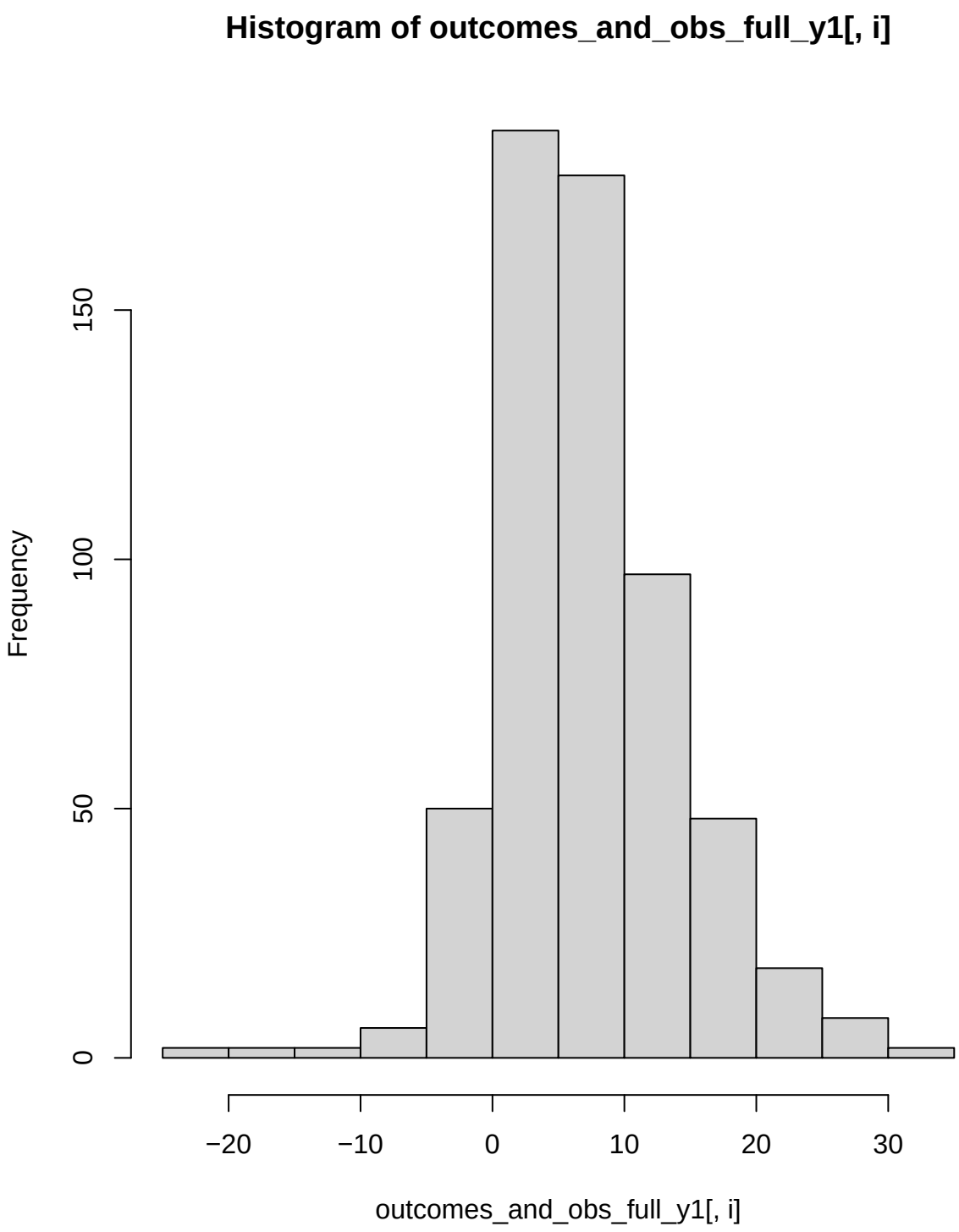
```
for (i in gain_ind) {
  name
  hist(outcomes_and_obs_full_y1[, i])
}
```

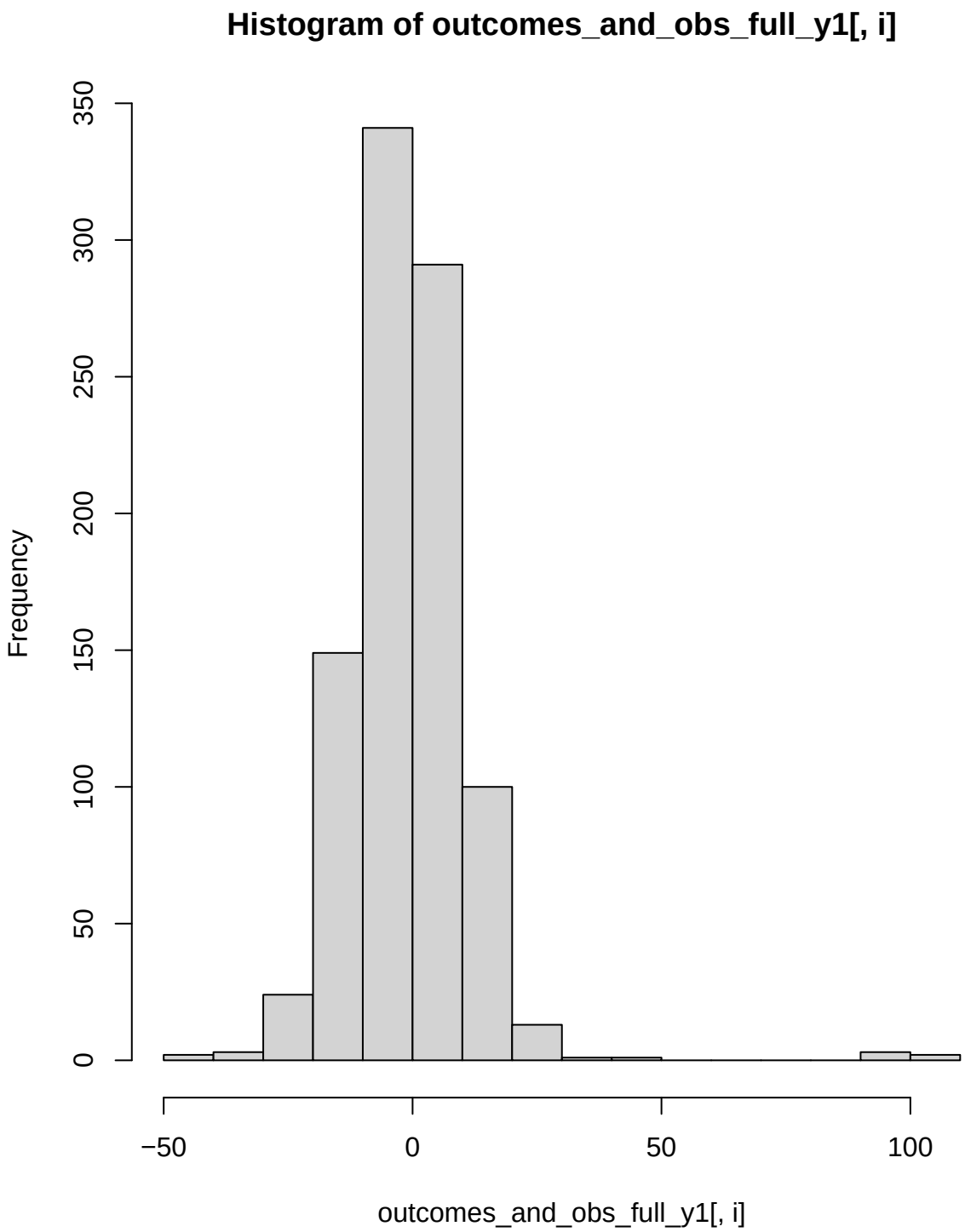

Histogram of outcomes_and_obs_full_y1[, i]
Frequency
0
50
100
150
200
250
300
−40
−20
0
20
40
60
outcomes_and_obs_full_y1[, i]
His
Frequency
0
50
100
150
200
250
−1.0


Histogram of outcomes_and_obs_full_y1[, i]
Frequency
0
100
200
300
400
−60
−40
−20
0
20
40
60
outcomes_and_obs_full_y1[, i]
His
Frequency
0
100
200
300
400
500
600
−40


Histogram of outcomes_and_obs_full_y1[, i]
Frequency
0
50
100
150
200
250
−5
0
5
10
outcomes_and_obs_full_y1[, i]


Hi
Frequency
0
50
100
150
200


Histogram of outcomes_and_obs_full_y1[, i]
Frequency
0
50
100
150
−20
−10
0
10
20
30
outcomes_and_obs_full_y1[, i]
His
Frequency
0
100
200
300
400




```
# loop through the names associated with each outcome -- add 1 to corresponding entry

count_coefs_y1 <- list()
for (outcome in coefs_y1) {
  for (name in names(outcome)) {
    if (name == '(Intercept)') {
      next
    }
```

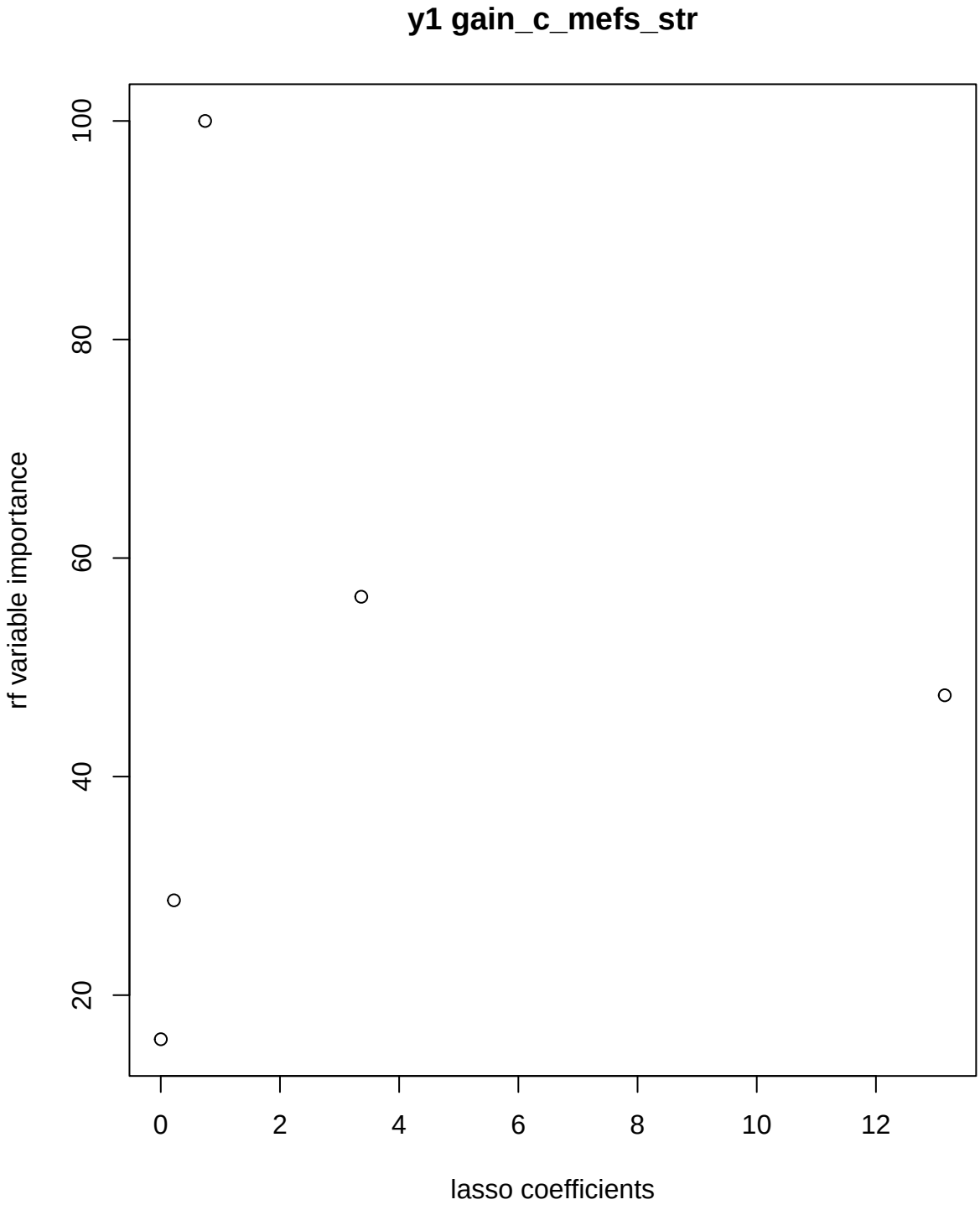
```
    count_coefs_y1[[name]] <- ifelse(is.null(count_coefs_y1[[name]]), 1,
                                     count_coefs_y1[[name]] + 1)
  }
}
```

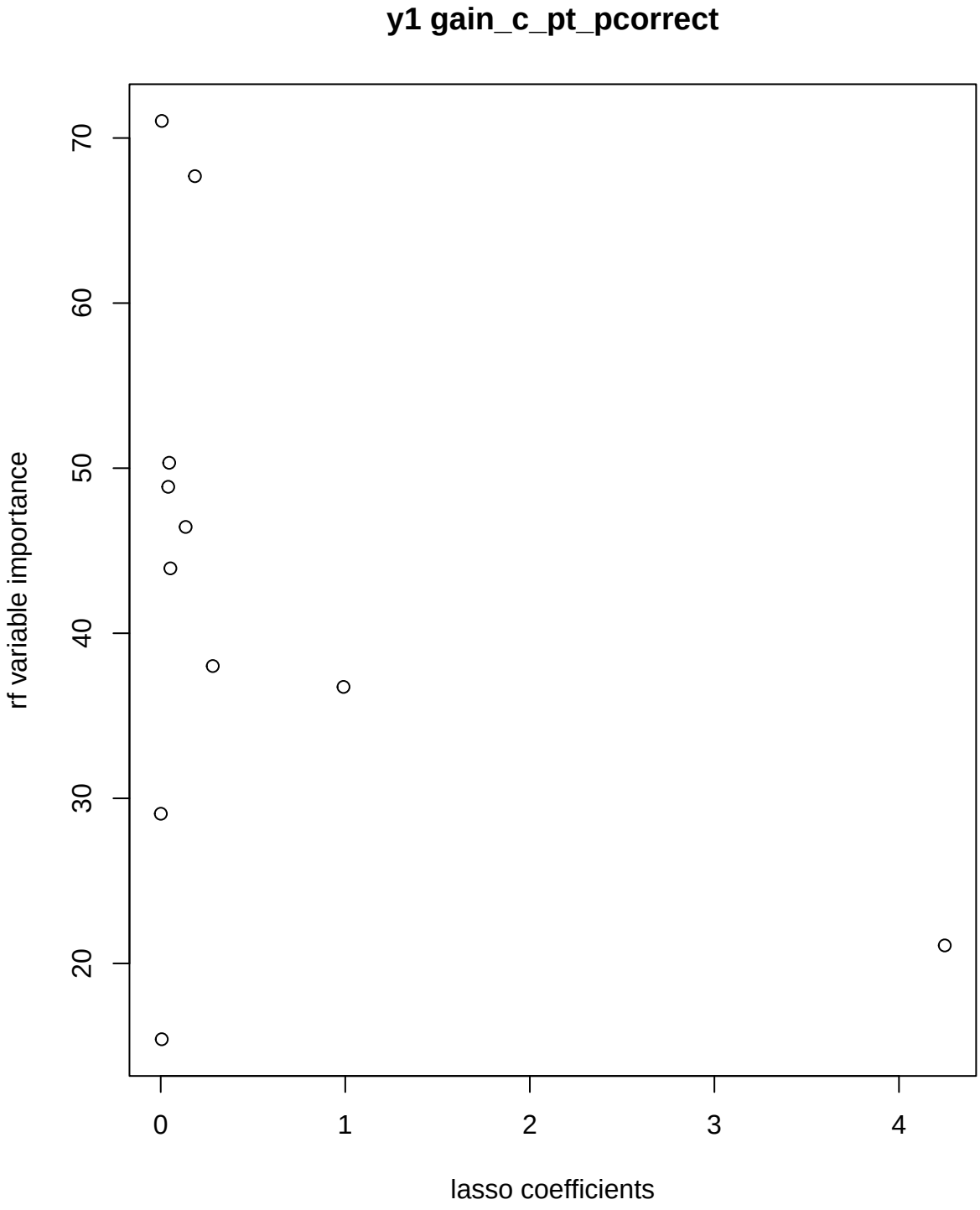
```
varimp_y1 <- list()
# NOTE: change to full data set here
test_data <- head(outcomes_and_obs_full_y1, 100)
# make sure this works with small subset of data
for (i in gain_ind) {
  name <- names(outcomes_and_obs_full_y1)[i]
  df_analysis <- test_data %>%
    filter(!is.na(test_data[[name]])) %>%
    select(c(name, "o_c_verbal_Fuss/Cry (FC)":actualtype))
    # ask about verbal_fuss vs o_c_verbal_Talk(T) (original)
    # mutate_at(vars("o_c_verbal_Fuss/Cry (FC)":actualtype), replace.na)
  print(name)
  # options(na.action="na.pass")
  x = model.matrix(as.formula(paste(name, "~ .")), data = df_analysis)

  # Fit the random forest model
  rf_fit <- train(as.formula(paste(name, "~ .")), #Use all variables in the prediction
                  data = df_analysis, #Use the training data
                  method = "ranger",
                  importance = "permutation",
                  # ntree = 500,
                  na.action=na.pass)

  varImp(rf_fit) %>%
    pluck(1) %>%
    rownames_to_column("var") %>%
    ggplot(aes(x = reorder(var, Overall), y = Overall)) +
    geom_col(fill = "grey75") +
    coord_flip() +
    theme_minimal()

  # store variable importances as a data frame
  df <- as.data.frame(varImp(rf_fit)$importance)
  # arrange in descending order (most to least important), and put into list
  varimp_y1[[name]] <- df %>% arrange(desc(Overall))
}
```

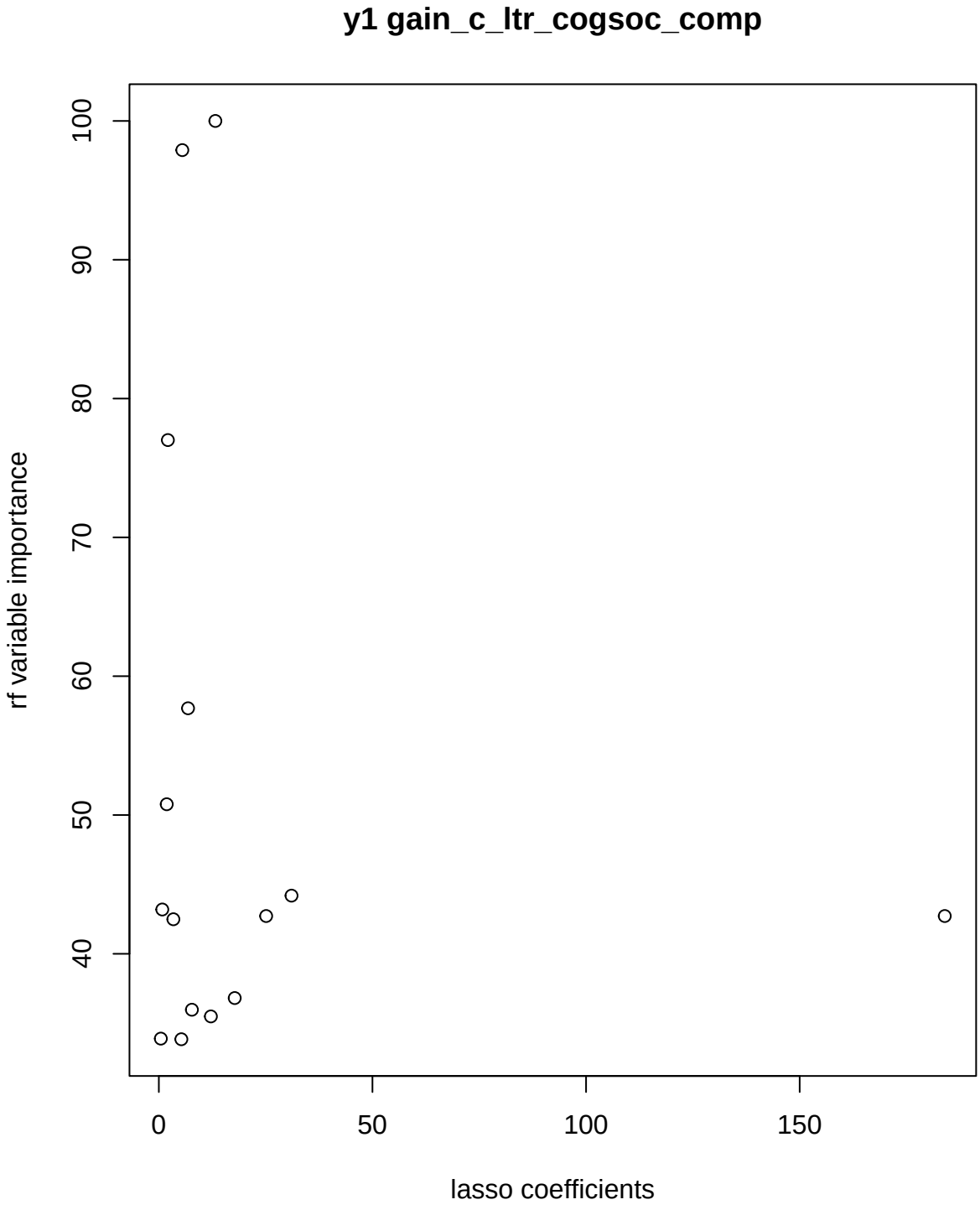
```
## [1] "gain_c_mefs_str"
## [1] "gain_c_pt_pcorrect"
## [1] "gain_c_ltr_cogsoc_comp"
## [1] "gain_c_ltr_emo_comp"
## [1] "gain_c_pra_total"
## [1] "gain_c_pbsa_total"
## [1] "gain_c_quils_total_raw"
## [1] "gain_c_wjlw_str"
## [1] "gain_c_wjap_str"
```

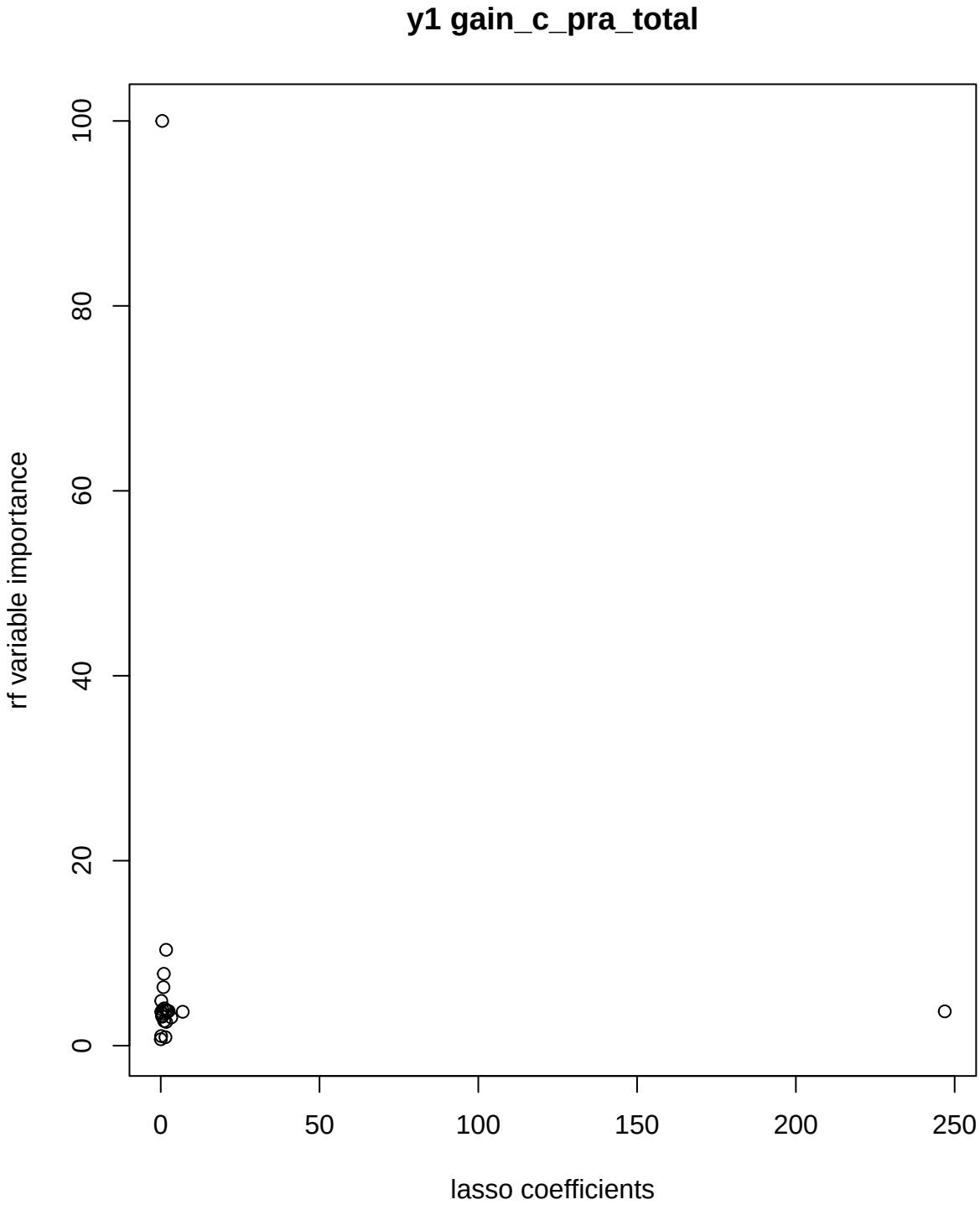
```
# comparing lasso and random forest results, for year 1
outcomes <- names(outcomes_and_obs_full_y1)[gain_ind]
```

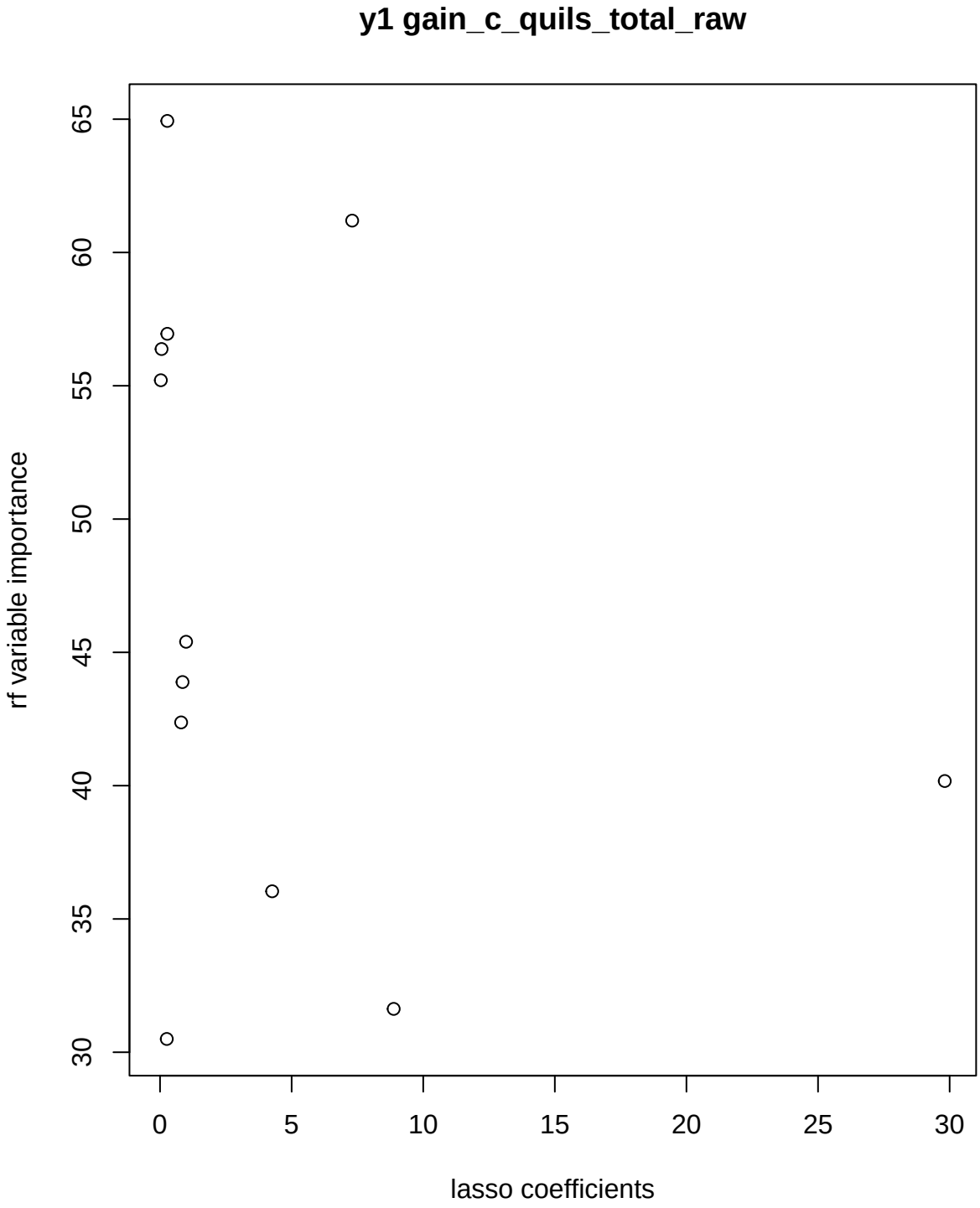
```r
for (outcome in outcomes) {
  # print(outcome)
  lasso_coefs <- coefs_y1[[outcome]]
  # get rid of intercept
  if (length(lasso_coefs) == 1) {
    next
  }
  lasso_coefs <- lasso_coefs[2:length(lasso_coefs)]
  rf_coefs <- varimp_y1[[outcome]]
  # this will be in the order of the coefficients in the lasso model
  overlap <- rf_coefs[names(lasso_coefs),]
  x <- rep(NA, length(lasso_coefs))
  for (i in 1:length(lasso_coefs)) {
    x[i] <- abs(lasso_coefs[[i]])
  }
  plot(x, overlap, xlab='lasso coefficients', ylab='rf variable importance', main=paste('y1', outcome))
}
```

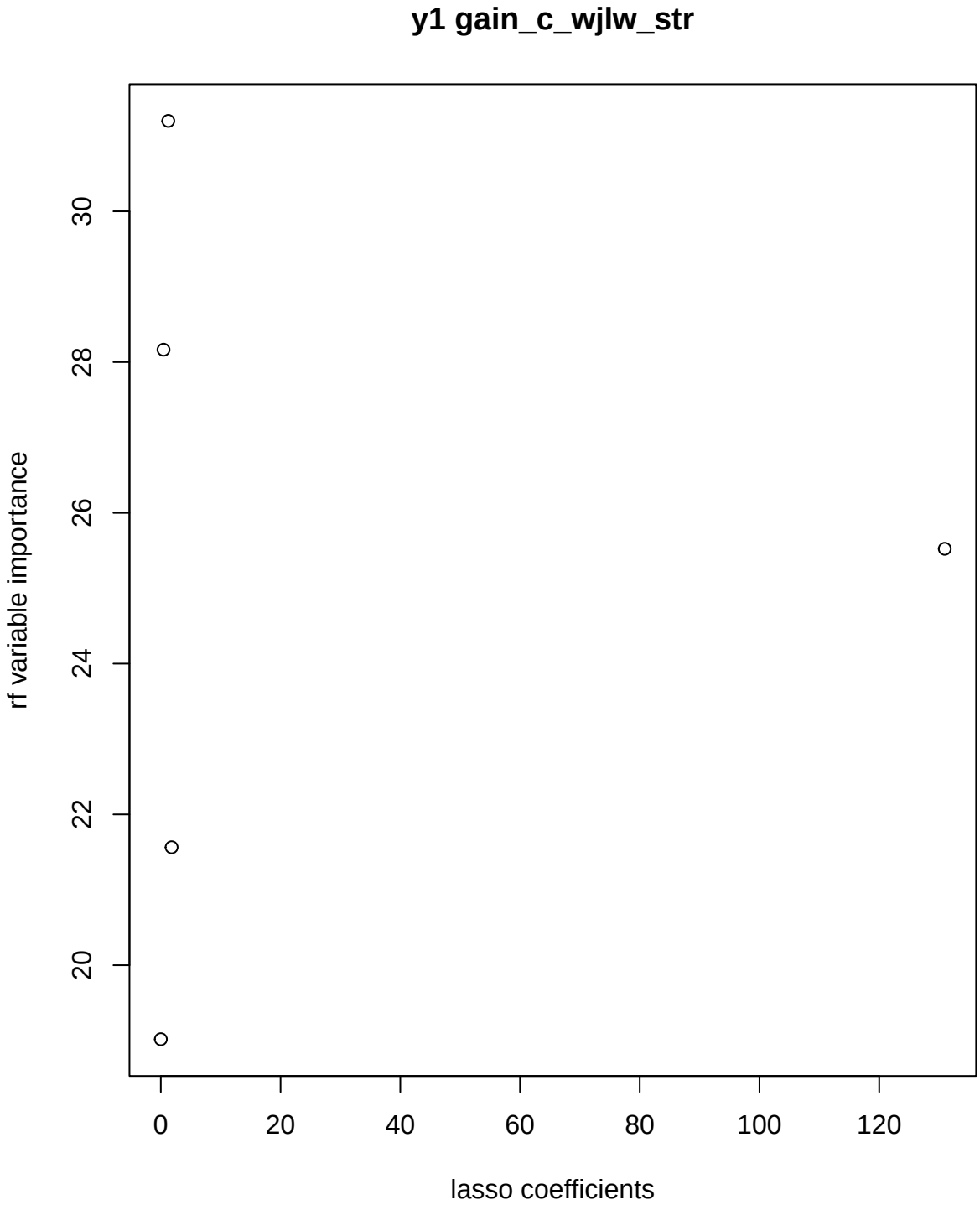

y1 gain_c_mefs_str
rf variable importance
lasso coefficients
20
40
60
80
100
0
2
4
6
8
10
12

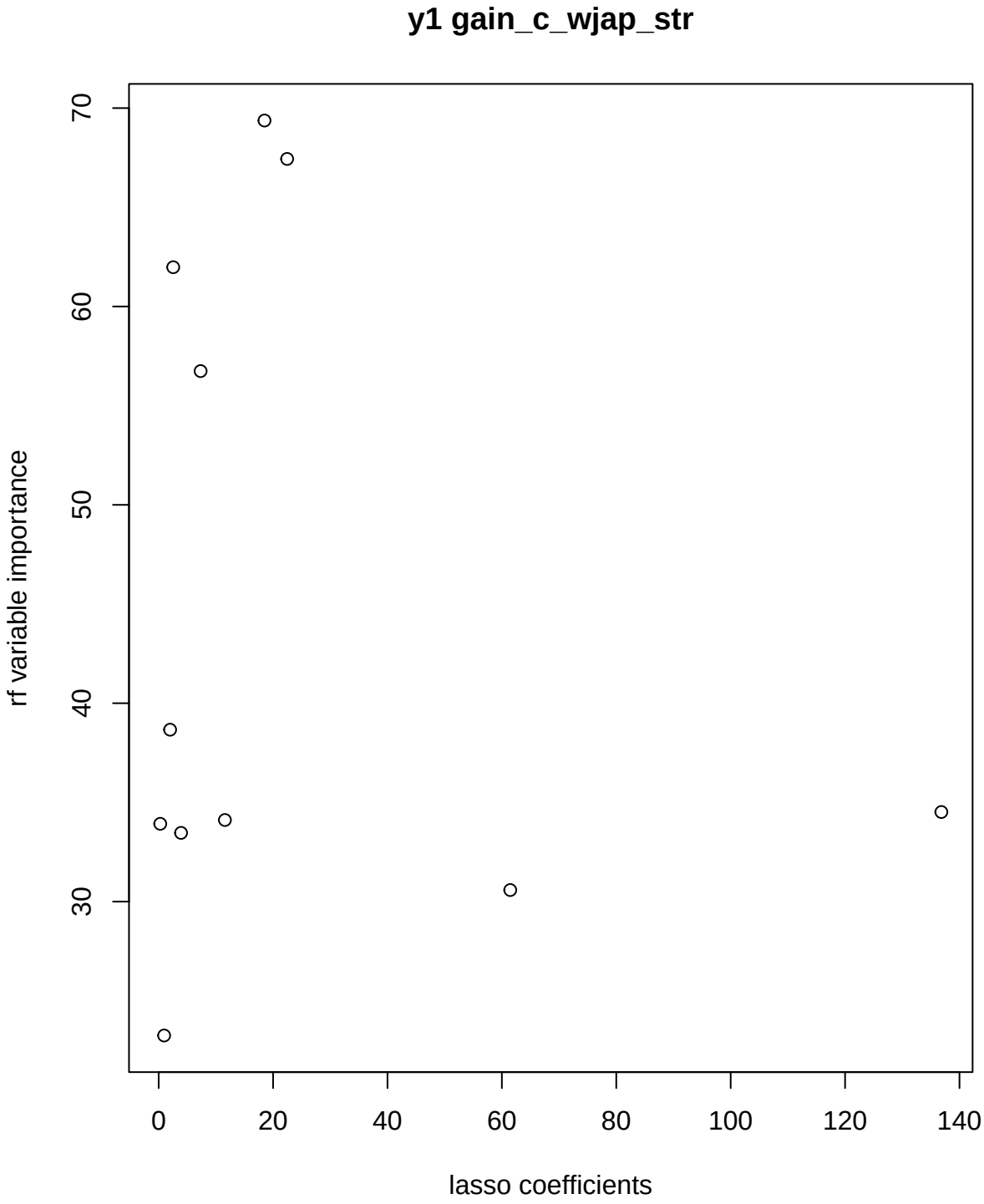

y1 gain_c_pt_pcorrect
rf variable importance
70
60
50
40
30
20
0
1
2
3
4
lasso coefficients


y1 gain_c_ltr_cogsoc_comp
rf variable importance
100
90
80
70
60
50
40
0
50
100
150
lasso coefficients


y1 gain_c_pra_total
rf variable importance
lasso coefficients
0
20
40
60
80
100
0
50
100
150
200
250


y1 gain_c_quils_total_raw
rf variable importance
lasso coefficients
30
35
40
45
50
55
60
65
0
5
10
15
20
25
30


y1 gain_c_wjlw_str
rf variable importance
20
22
24
26
28
30
0
20
40
60
80
100
120
lasso coefficients




```
#NOTES FROM MEETINGI 8/8
#1. Try Y2 observation data instead of Y1, and compare results
#2. Try a model with a bunch of predictors from Y1 Y2 -- looking at more demographic variables
# wait for Jonathan to see which demographic variables are important
#3. Use the particular caretype from the  setting that is being observed in. (prov_type)
#Collapse CC-Community Based & CC- License Exempt --
#4. Include leave-out standard deviation for each predictor as well
#5. Incorporte child-level covariates -- wait for this one as well
```

```
#6. Age and elapsed time for assessment
# look for date in the cleaning process -- flag it and let Jonathan take closer look

## 4 is a priority, plus other things we discussed (on google doc)

# comparing random forest and lasso results -- scatter the absolute value of coefficients (double check
# should hopefully see some
```

```
# As far as I can tell, variable importance is measuring either: a) the percentage that the prediction

# I'm not sure if I will try to "unstandardize" the random forest results. I will use the magnitude of

# todo:
# 1. extract coefficient values for each selected variable from lasso - maybe store in data frame?
# 2. find a way to extract numeric value of variable importance - also store in data frame
# 3. merge data frames -- thinking one data frame per variable? so list of data frames
# 4. scatter values against each other
```

```
#Merge in the outcomes data
#  Try Y2 observation data instead of y1, and compare results

# add this line in so that the merge is correctly executed

y2_obs <- y2_obs %>%
  filter(!is.na(cid)) %>%
  mutate(cid = as.numeric(cid))

outcomes_and_obs_y2 <- left_join(child_outcomes, y2_obs, by = "cid") %>%
  mutate(cid = as.character(cid))

#Add in the care type -- confirm that it is supposed to be year-specific
caretype <- read.dta13("y2caretype.dta", nonint.factors = TRUE) %>%
  mutate(cid = as.character(cid))

#Remove observations that have no care type or no classroom observation
outcomes_and_obs_full_y2 <- left_join(outcomes_and_obs_y2, caretype, by = "cid") %>%
  mutate(hasobservation = is.na(classid)) %>%
  filter(!is.na(caretype)) %>%
  filter(!is.na(classid))

# outcomes_and_obs_full_y1 <- outcomes_and_obs %>%
#   filter(!is.na(classid))

#Remove Y1 and Y2 data for cleanliness
outcomes_and_obs_full_y2 <- outcomes_and_obs_full_y2 %>%
  select(-starts_with(c("y1", "y2")))

#Remove some irrelevant variables
outcomes_and_obs_full_y2 <- outcomes_and_obs_full_y2 %>%
  # seems like provid is the same thing as famid in this case?
  select(-c(provid, dob, actualtype_fcc:hasobservation)) %>%
  mutate(caretype = as.factor(caretype),
         actualtype = as.factor(actualtype))
```

```
## INSERT SOME REPLACE.NA FUNCTION HERE
```

```
#There are some issues with NA and NaN in the observation data that will mess up our analysis. We will

replace.na <- function(var){
  ifelse(is.na(var) | is.nan(var), mean(var), var)
}
# names(outcomes_and_obs_full_y1)
# double check whether mean imputation works for factors
outcomes_and_obs_full_y2 <- outcomes_and_obs_full_y2 %>%
  mutate_at(vars("o_c_verbal_Fuss/Cry (FC)":actualtype), replace.na)

# what is o_t_sched_t supposed to be for? replacing with actualtype for now just to run analysis

# replacing all NaNs in hopes that it will fix the model matrix issue
# originally was o_c_verbal_talk:o_t_sched_t

dim(outcomes_and_obs_full_y2)
```

```
## [1] 765 303
```

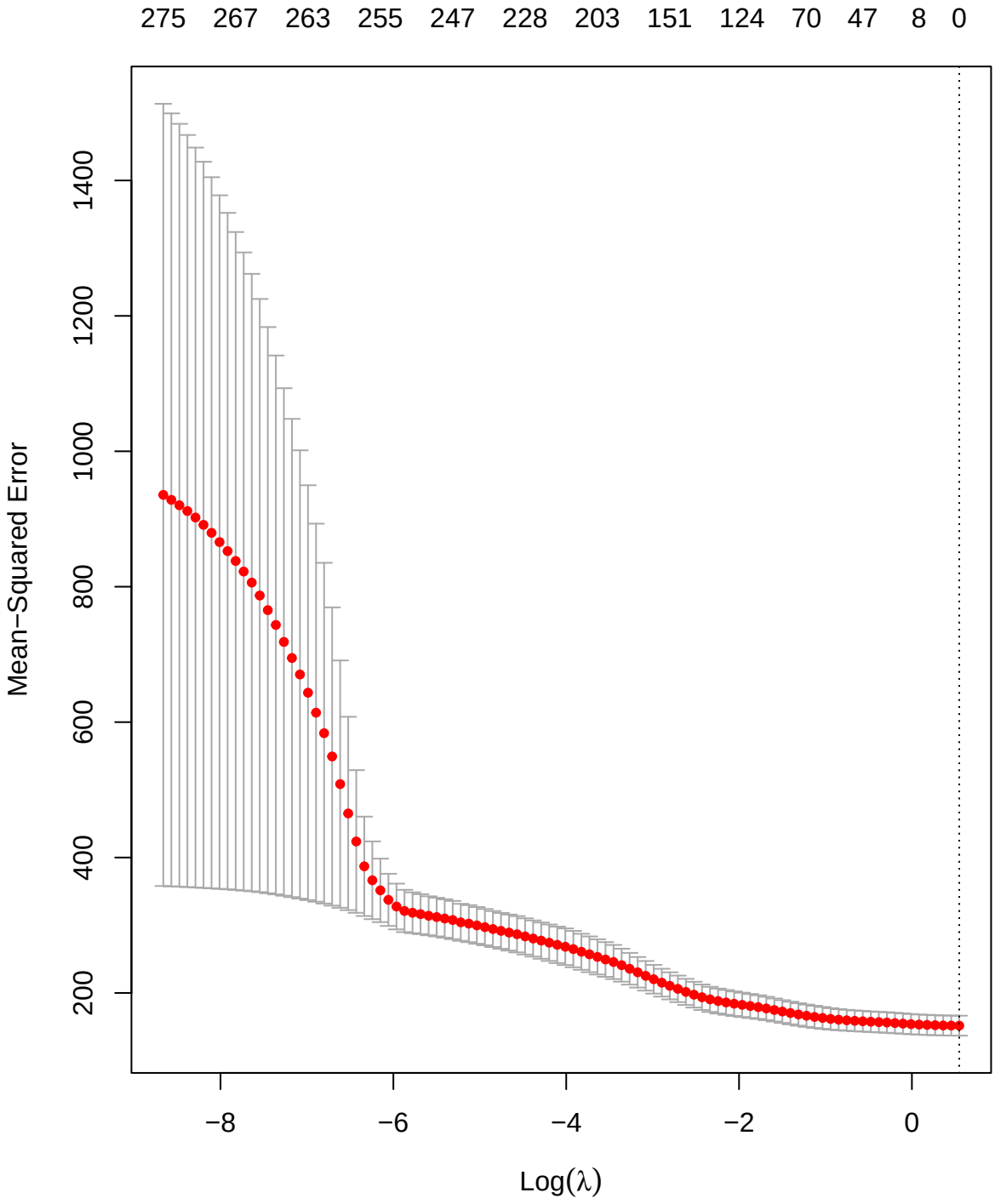
Analysis

We will first try a cross-validated LASSO, which will aggressively remove variables that do little to improve the predictive accuracy of the model.

```
set.seed(4224)

gain_ind <- which(startsWith(colnames(outcomes_and_obs_full_y2), "gain"))

models_y2 <- list()
coefs_y2 <- list()

for (i in gain_ind) {
  name <- names(outcomes_and_obs_full_y2)[i]
  df_analysis <- outcomes_and_obs_full_y2 %>%
    filter(!is.na(outcomes_and_obs_full_y2[[name]])) %>%
    select(c(name, "o_c_verbal_Fuss/Cry (FC)":actualtype))
  print(name)
  options(na.action="na.pass")
  x = model.matrix(as.formula(paste(name, "~ .")), data = df_analysis)

  y = df_analysis[[name]]
  print(dim(x))
  x = x[, -1]


  # call cv.glmnet()
  model_lasso <- cv.glmnet(x = x, y = y, alpha = 1)
  plot(model_lasso)

  models_y2[[name]] <- model_lasso

  cc = coef(model_lasso, s = model_lasso$lambda.min)
```

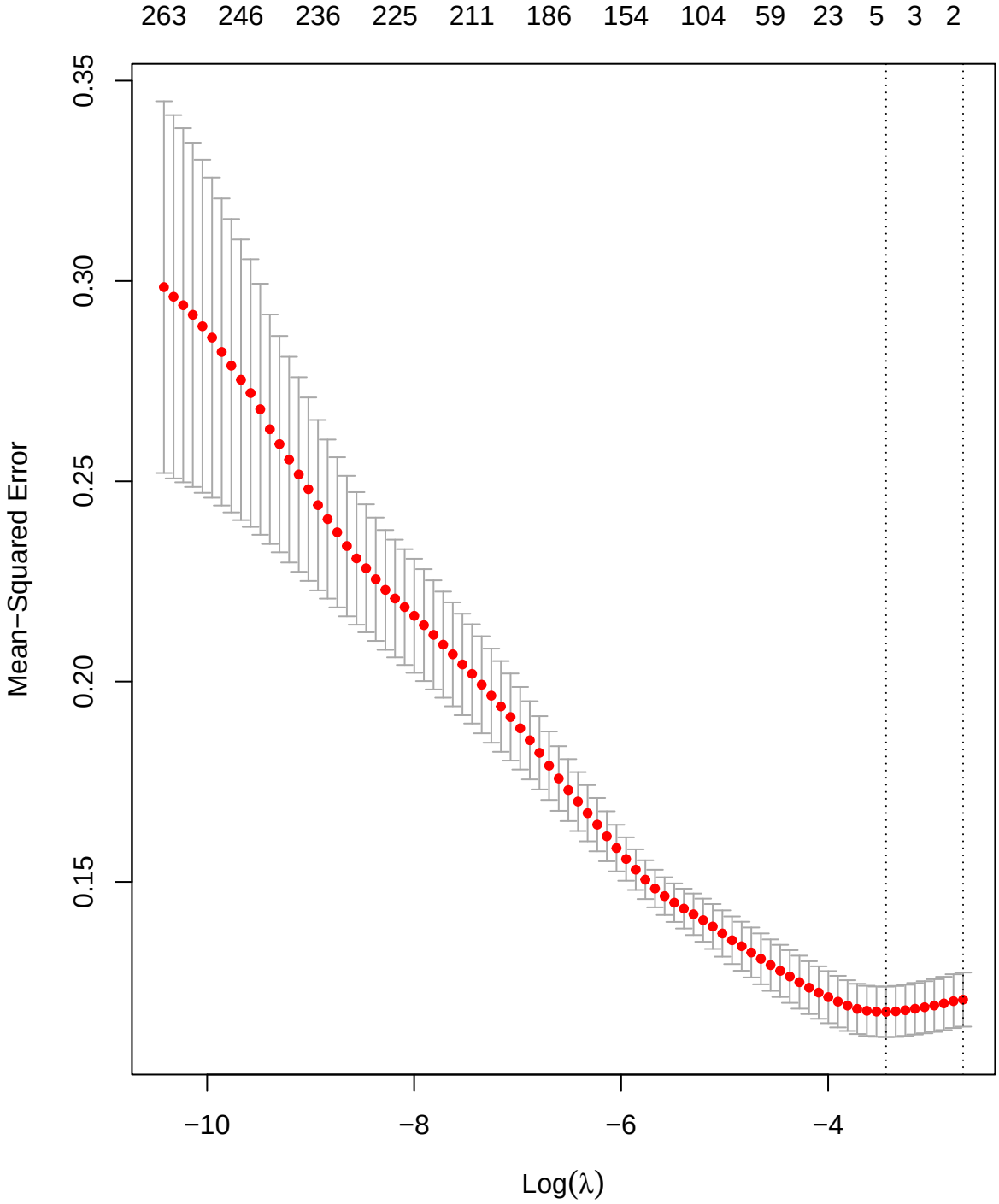
```
  # print out the model coefficients and store in a list.
  cc = cc[cc[,1]!=0,1]
  coefs_y2[[name]] <- cc
  print(cc)
}
```

```
## [1] "gain_c_mefs_str"
## [1] 522 286
```

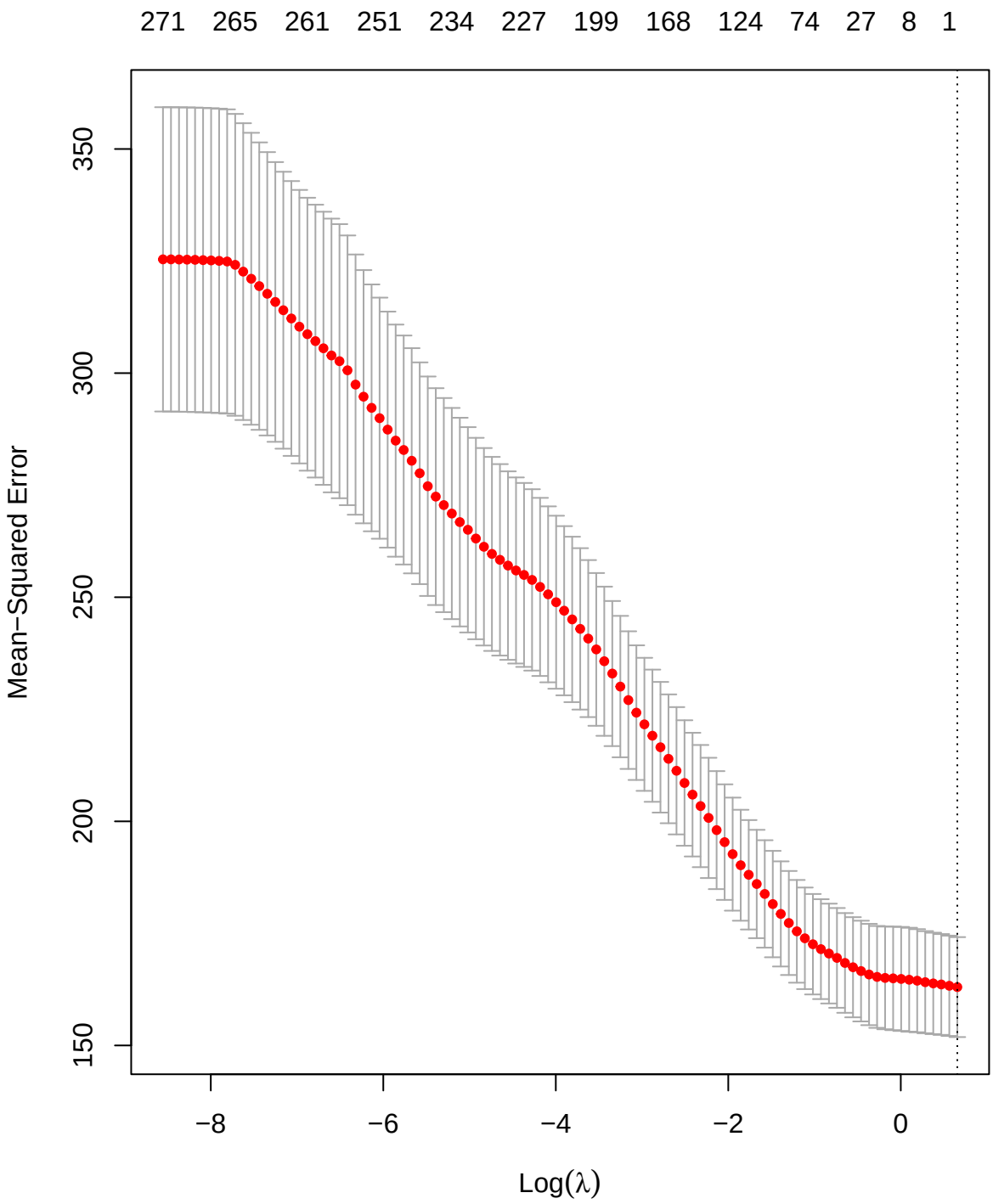



```
## [1] 3.467433
```

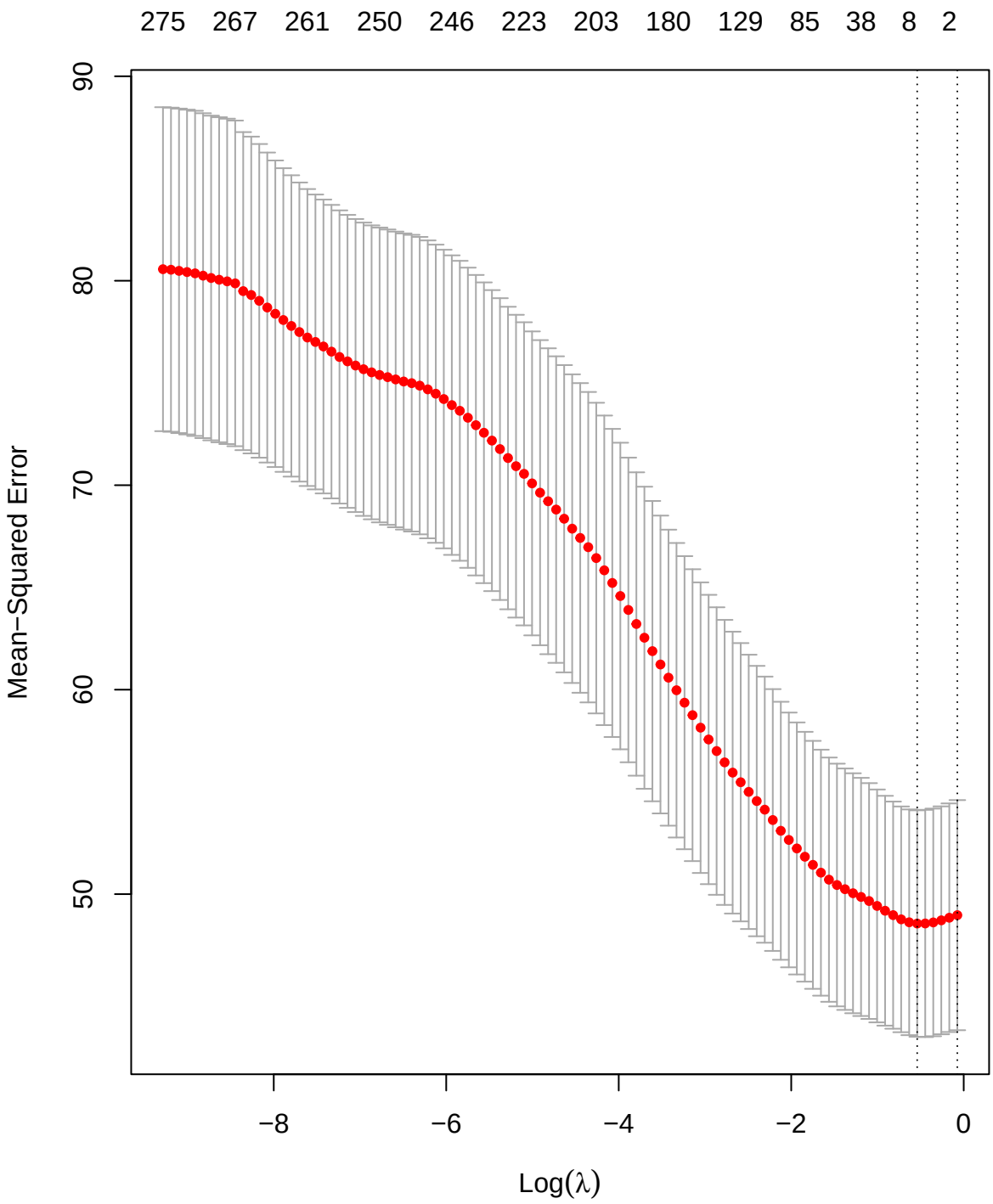
```
## [1] "gain_c_pt_pcorrect"
## [1] 609 286
```




```
##             (Intercept) `o_c_focus_Reading (R)`           o_t_focus_o_A
##            2.357163e-01           -1.361160e-01           -4.934031e-02
##           grade_y2Pre-K         grade_kinder_y2
##            6.949116e-02           -7.533423e-15
## [1] "gain_c_ltr_cogsoc_comp"
## [1] 687 286
```

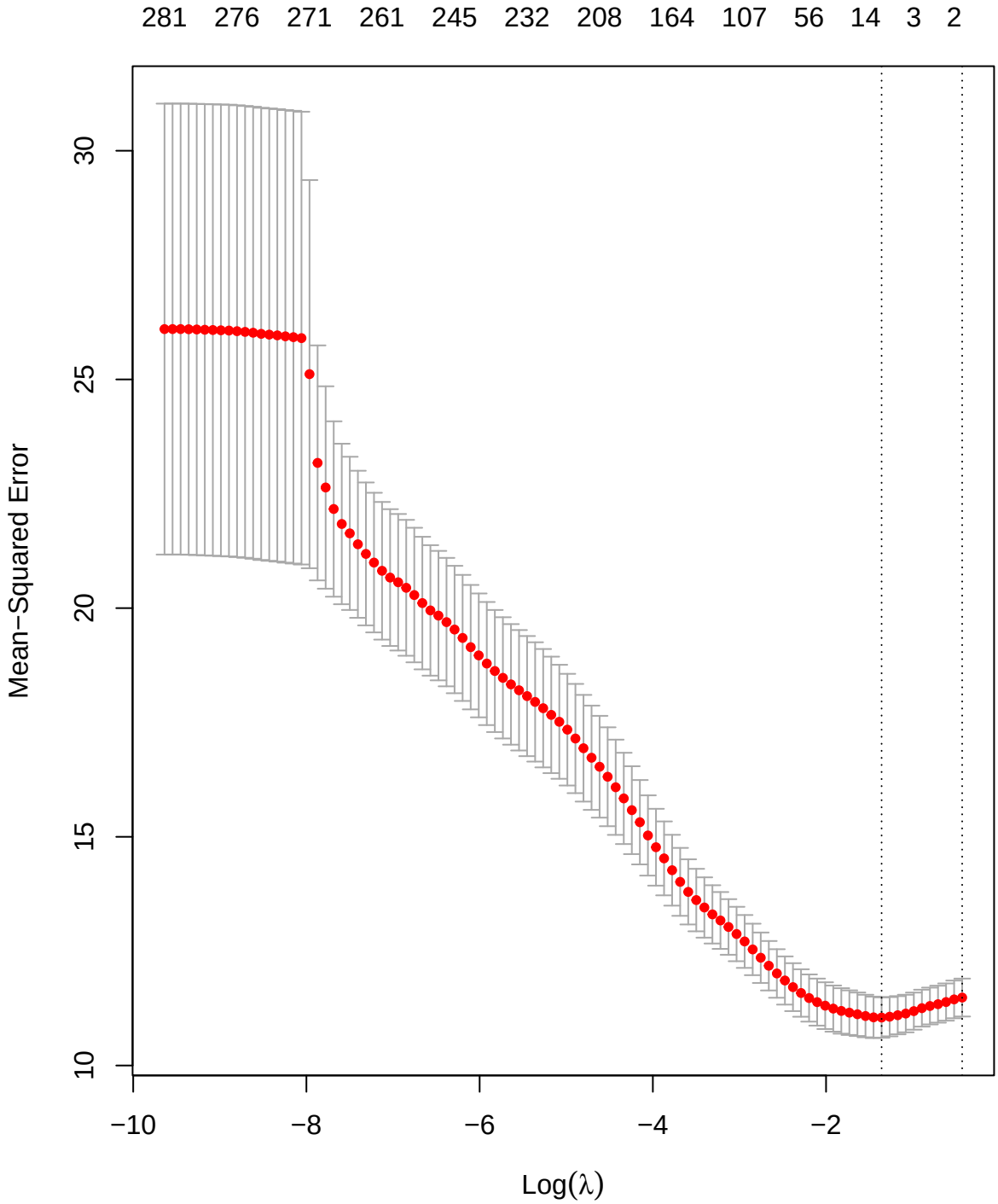



```
## [1] 1.524017
## [1] "gain_c_ltr_emo_comp"
## [1] 687 286
```



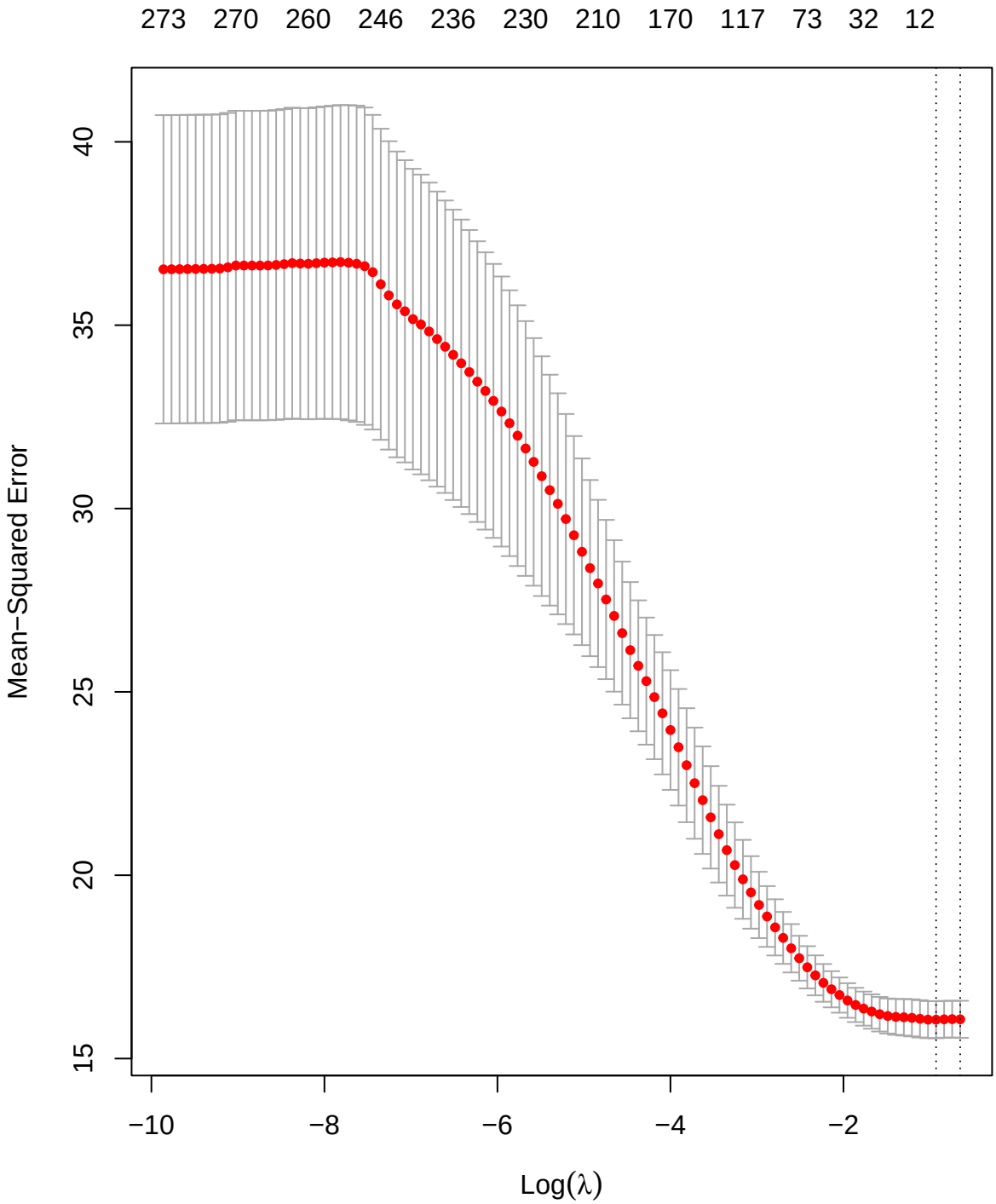

```
##                                      (Intercept)
##                                        1.7119943
##                          `o_c_involvement_High (H)`
##                                        0.7776921
##                       `o_c_schedule_Nap (N)_classmean`
##                                     -164.4309306
## `o_c_towhom_Small Group, No Teacher (SG)_classsd`
##                                      -18.4915296
##                                    o_t_verbal_o_T
```

```
##                                       -1.4038409
## [1] "gain_c_pra_total"
## [1] 612 286
```



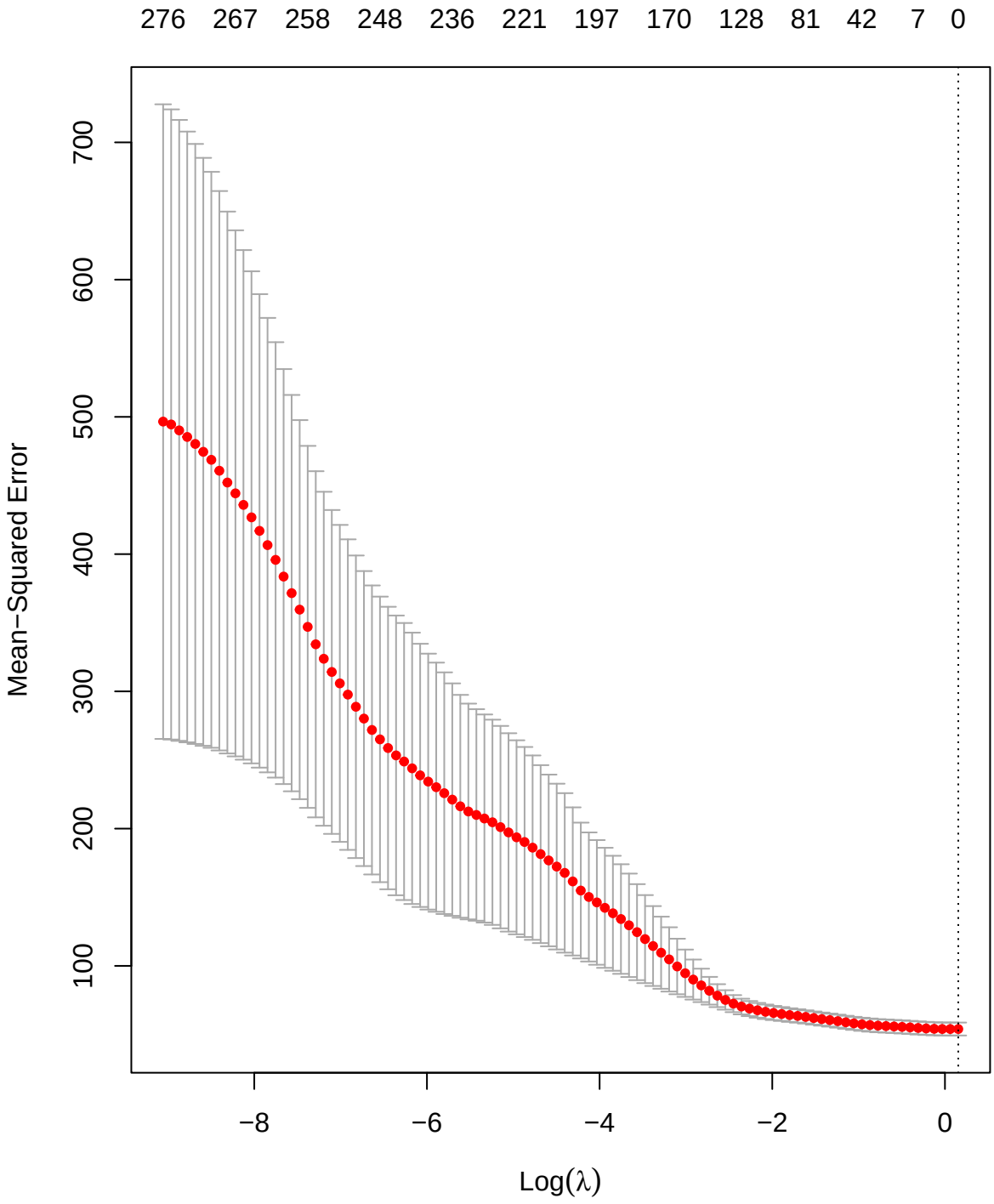

```
##                     (Intercept) `o_c_typetask_Fantasy/Drama (FD)`
##                       2.5505442                        -0.7000711
##   `o_c_focus_Drama (D)_classmean`                      o_t_task_o_N
##                      -1.8333482                        -4.3587290
##                    o_t_instruct_3                         o_t_es_o_A
##                      -2.9883149                         0.3499218
```

```
##                     grade_y2Pre-K                       caretype_fcc
##                         0.8209538                          0.1591124
## [1] "gain_c_pbsa_total"
## [1] 622 286
```

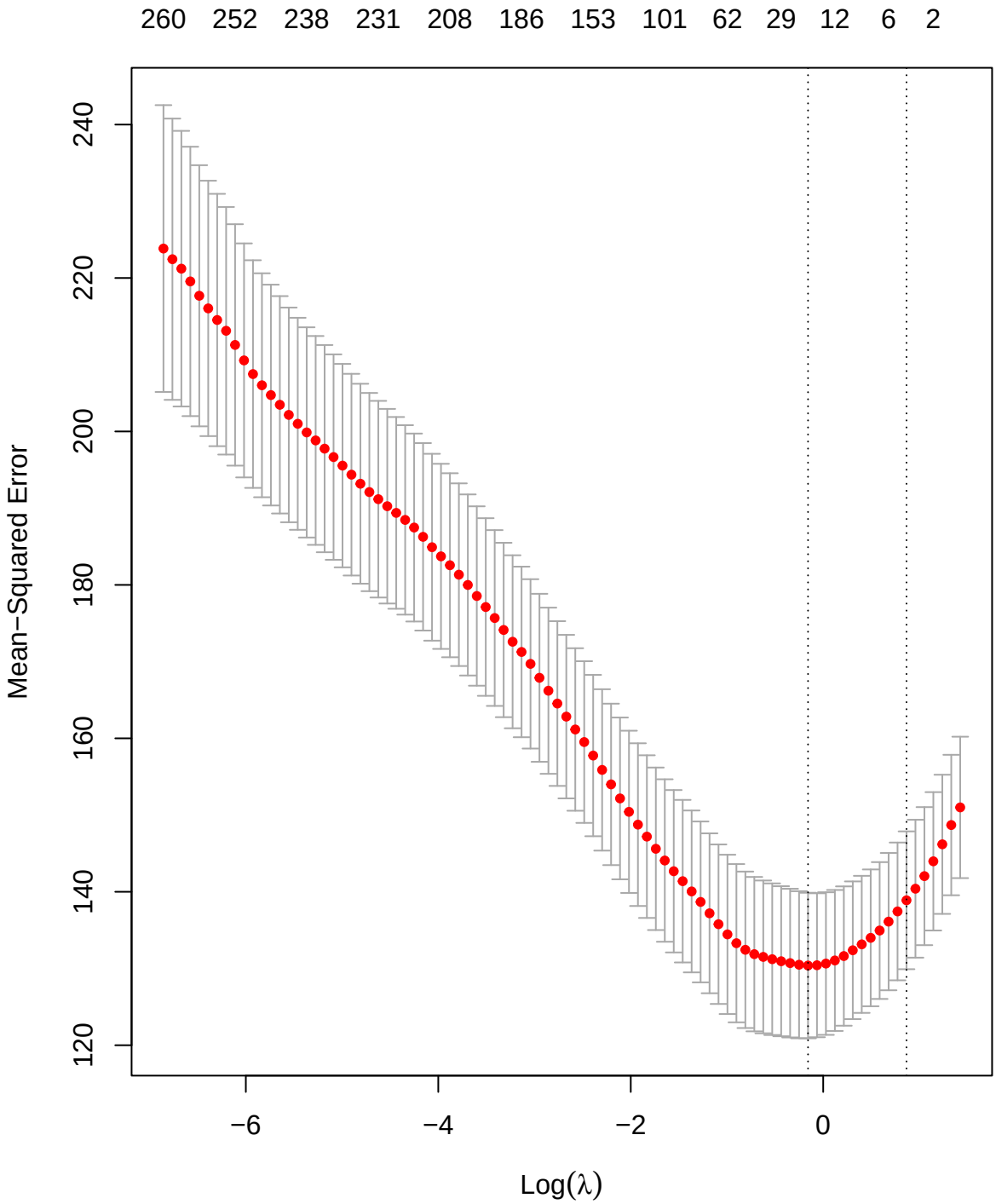



```
##                                       (Intercept)
##                                        3.04724701
##                         `o_c_typetask_Fantasy/Drama (FD)`
##                                       -1.58963833
## `o_c_towhom_Whole Group, With Teacher (WGT)_classmean`
```

```
##                                           2.34429006
##                   `o_c_schedule_Playground (P)_classsd`
##                                           0.07973181
##                                      o_t_attention_o_N
##                                           0.55209034
##                                         grade_y2Pre-K
##                                           0.25237137
## [1] "gain_c_quils_total_raw"
## [1] 434 286
```

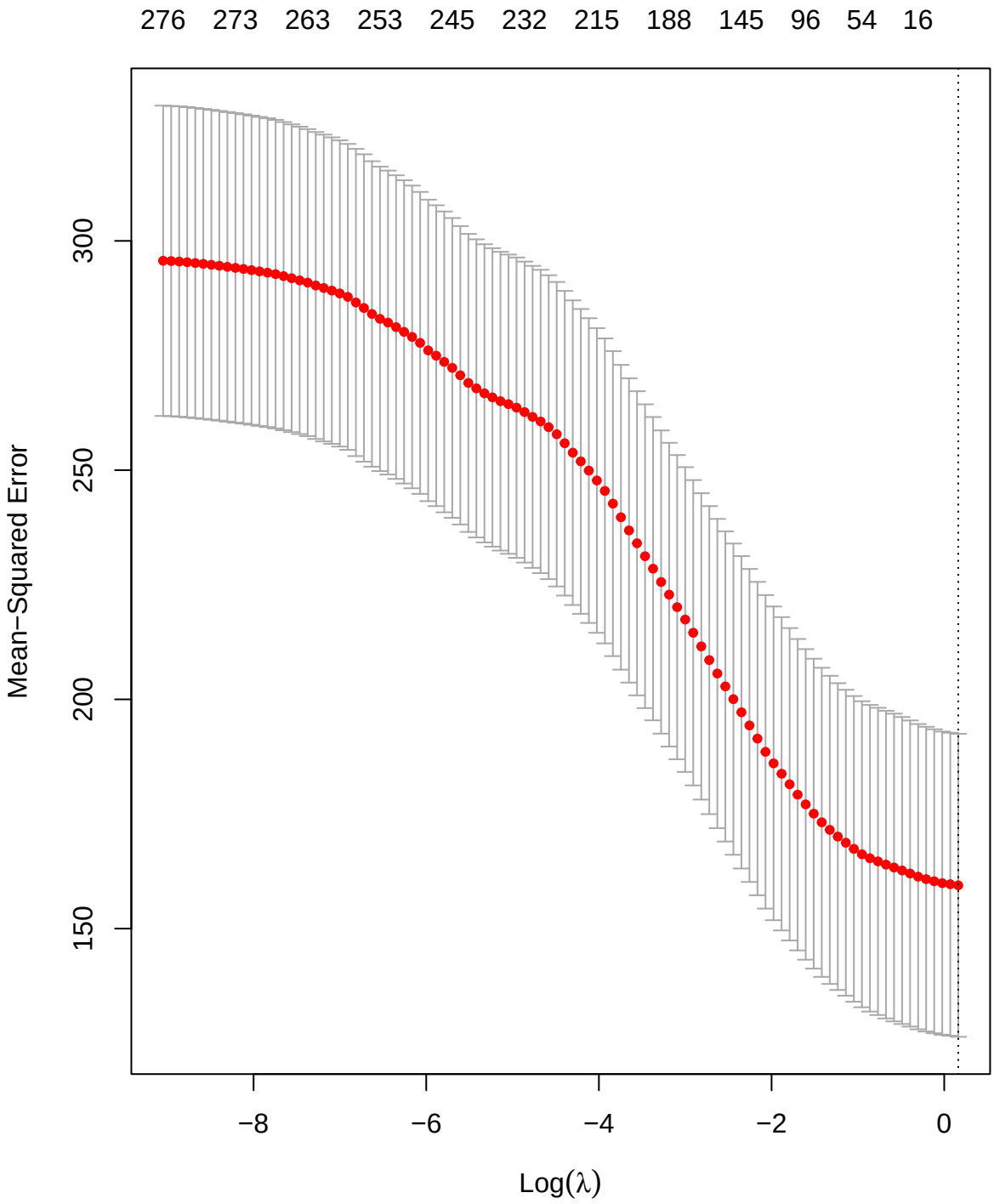


```
## [1] 7.654378
## [1] "gain_c_wjlw_str"
## [1] 683 286
```




```
##                                                 (Intercept)
##                                                   6.0418066
##                                       `o_c_schedule_Centers (C)`
##                                                  -0.2683750
##                  `o_c_schedule_Individual activity w teacher (IAT)`
##                                                  16.1210138
```

```
##                                `o_c_typetask_Non-Sequential (NS)`
##                                                         -2.3982309
##                               `o_c_focus_Literacy - Writing (LW)`
##                                                          7.7788245
##                                            `o_c_focus_Reading (R)`
##                                                          0.7127827
##                                         `o_c_focus_Science (SC)`
##                                                         -0.9956814
## `o_c_towhom_Teacher (T): Teacher or assistant (adult)_classmean`
##                                                         -0.7381536
##               `o_c_towhom_Whole Group, No Teacher (WG)_classmean`
##                                                         28.0438131
##                                     `o_c_focus_Drama (D)_classmean`
##                                                        -10.2046087
##                         `o_c_interaction_Associative (AS)_classsd`
##                                                         -5.1845594
##                         `o_c_involvement_Medium High (MH)_classsd`
##                                                         -0.3416571
##                                       `o_c_focus_Drama (D)_classsd`
##                                                         -4.6885222
##                        `o_c_focus_Literacy - Writing (LW)_classsd`
##                                                         25.4056639
##                                  `o_c_focus_Literacy (L)_classsd`
##                                                          2.7929700
##                                                      o_t_verbal_o_T
##                                                         -1.7902595
##                                                       o_t_task_o_BD
##                                                         -1.3189334
##                                                     o_t_instruct_NA
##                                                          7.3439521
##                                                      o_t_focus_o_LW
##                                                          7.1771911
##                                                         o_t_es_o_N
##                                                         -1.9107235
##                                                    grade_y2Pre-K
##                                                         -2.0409834
##                                                       caretype_ps
##                                                          3.0740472
##                                                       caretype_hs
##                                                         -0.1552296
## [1] "gain_c_wjap_str"
## [1] 661 286
```




```
## [1] -1.331316
```

```
# loop through the names associated with each outcome -- add 1 to corresponding entry
count_coefs_y2 <- list()
for (outcome in coefs_y2) {
  for (name in names(outcome)) {
    if (name == '(Intercept)') {
      next
    }
    count_coefs_y2[[name]] <- ifelse(is.null(count_coefs_y2[[name]]), 1,
```

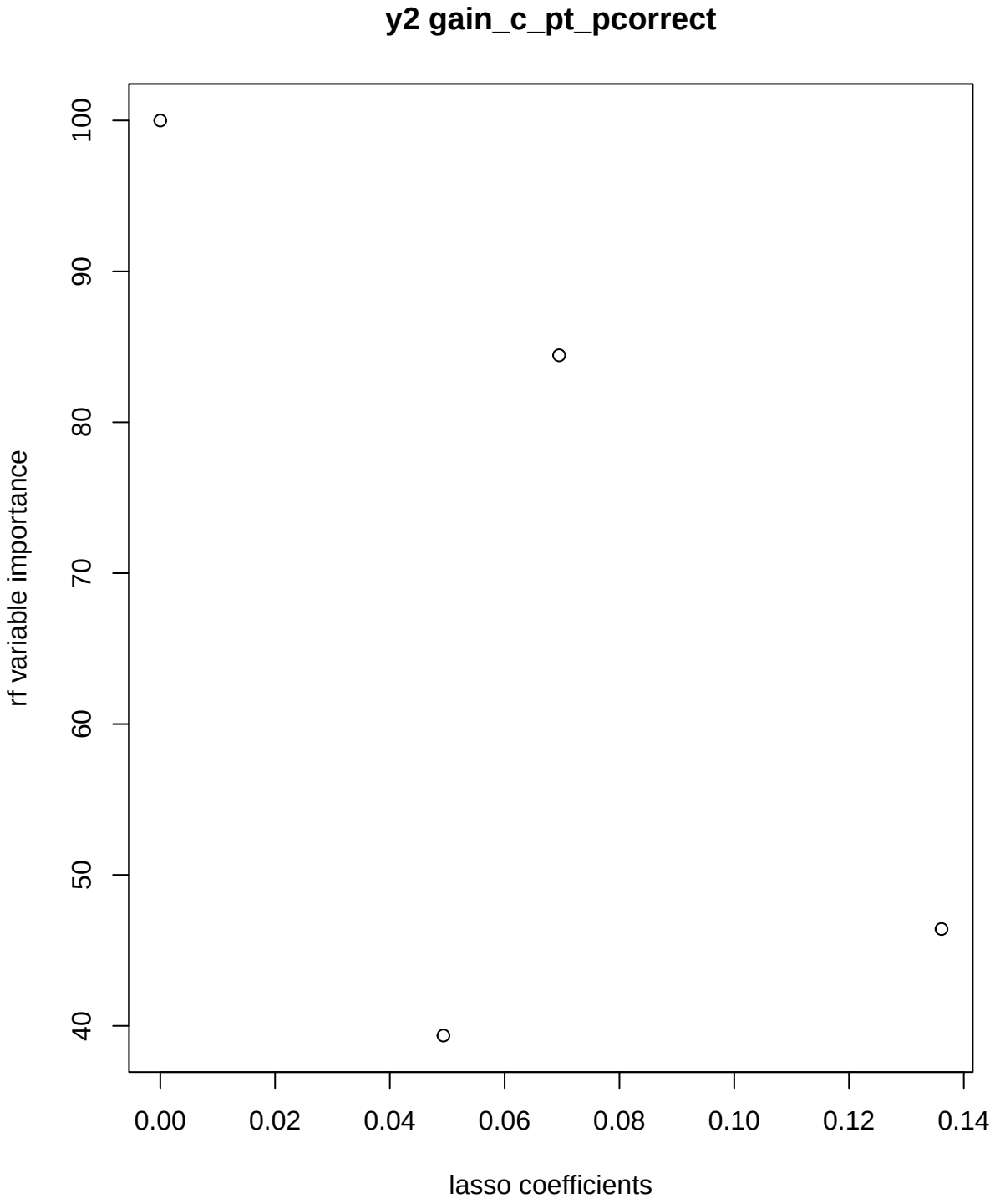
```
                                    count_coefs_y2[[name]] + 1)
  }
}
```

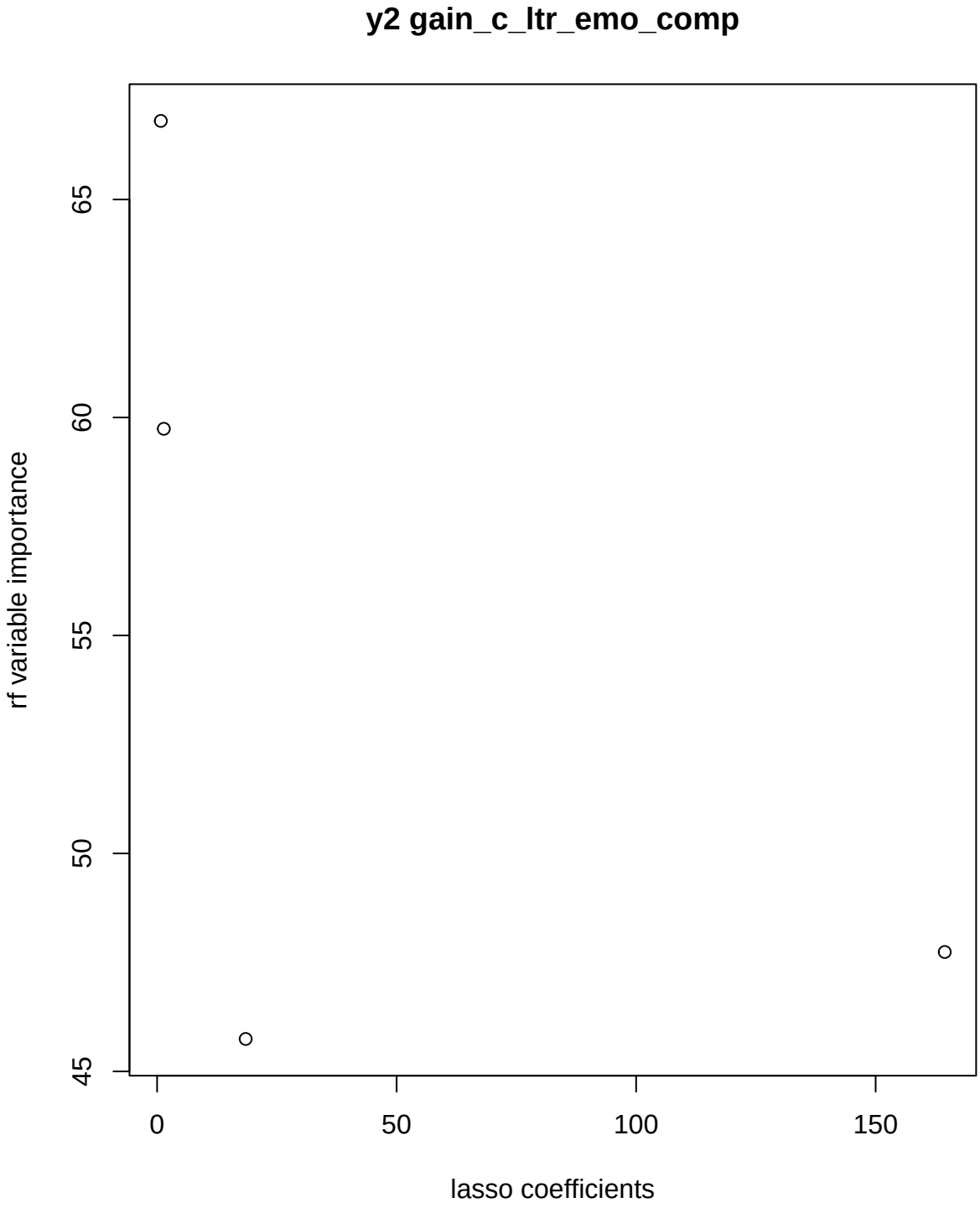
```
varimp_y2 <- list()
# NOTE: change to full data set here
test_data <- head(outcomes_and_obs_full_y2, 100)
# make sure this works with small subset of data
for (i in gain_ind) {
  name <- names(outcomes_and_obs_full_y2)[i]
  df_analysis <- test_data %>%
    filter(!is.na(test_data[[name]])) %>%
    select(c(name, "o_c_verbal_Fuss/Cry (FC)":actualtype))
    # ask about verbal_fuss vs o_c_verbal_Talk(T) (original)
    # mutate_at(vars("o_c_verbal_Fuss/Cry (FC)":actualtype), replace.na)
  print(name)
  # options(na.action="na.pass")
  x = model.matrix(as.formula(paste(name, "~ .")), data = df_analysis)

  # Fit the random forest model
  rf_fit <- train(as.formula(paste(name, "~ .")), #Use all variables in the prediction
                  data = df_analysis, #Use the training data
                  method = "ranger",
                  importance = "permutation",
                  # ntree = 500,
                  na.action=na.pass)

  varImp(rf_fit) %>%
    pluck(1) %>%
    rownames_to_column("var") %>%
    ggplot(aes(x = reorder(var, Overall), y = Overall)) +
    geom_col(fill = "grey75") +
    coord_flip() +
    theme_minimal()

  # store variable importances as a data frame
  df <- as.data.frame(varImp(rf_fit)$importance)
  # arrange in descending order (most to least important), and put into list
  varimp_y2[[name]] <- df %>% arrange(desc(Overall))
}
```

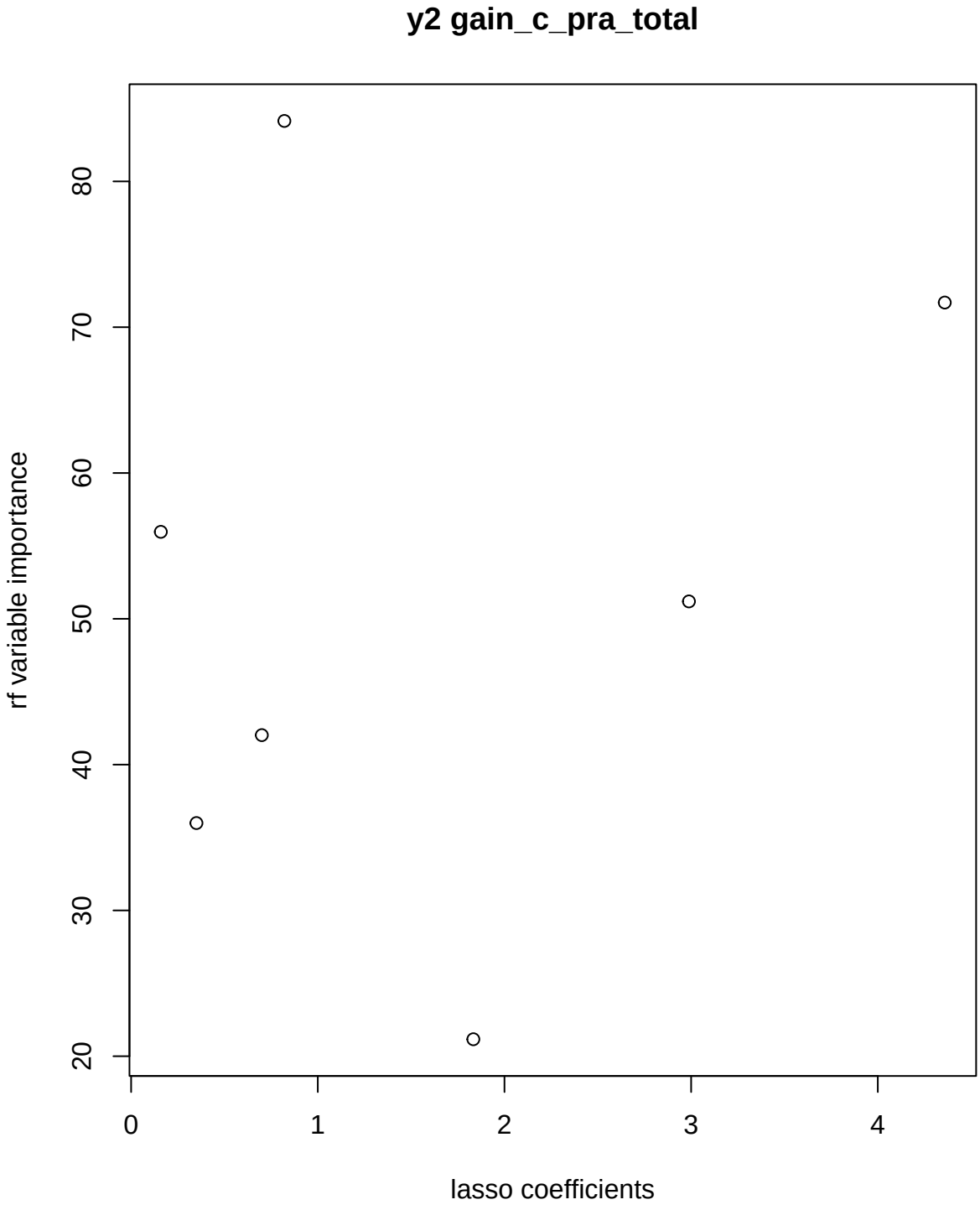
```
## [1] "gain_c_mefs_str"
## [1] "gain_c_pt_pcorrect"
## [1] "gain_c_ltr_cogsoc_comp"
## [1] "gain_c_ltr_emo_comp"
## [1] "gain_c_pra_total"
## [1] "gain_c_pbsa_total"
## [1] "gain_c_quils_total_raw"
## [1] "gain_c_wjlw_str"
## [1] "gain_c_wjap_str"
```

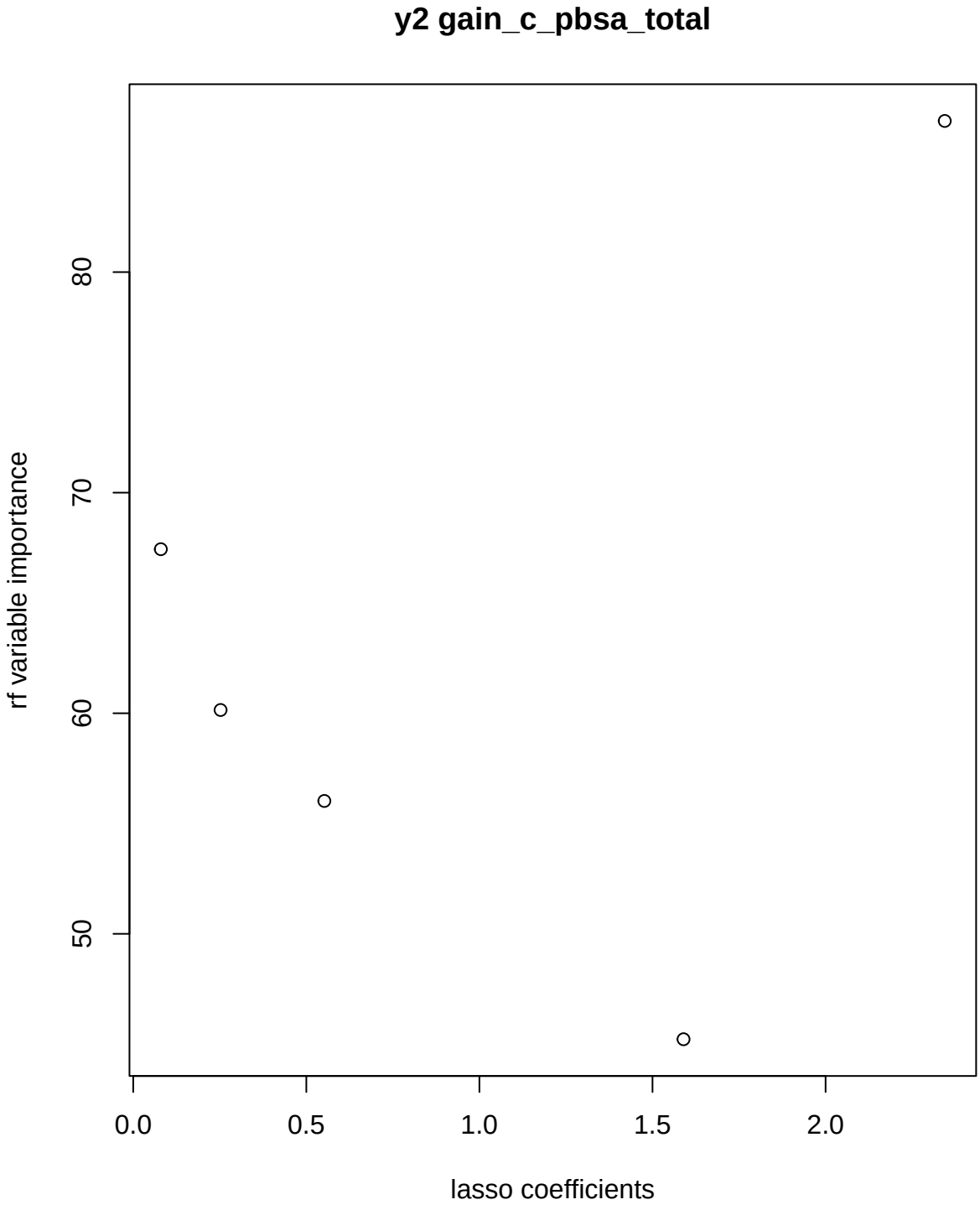
```
outcomes <- names(outcomes_and_obs_full_y2)[gain_ind]

for (outcome in outcomes) {
```

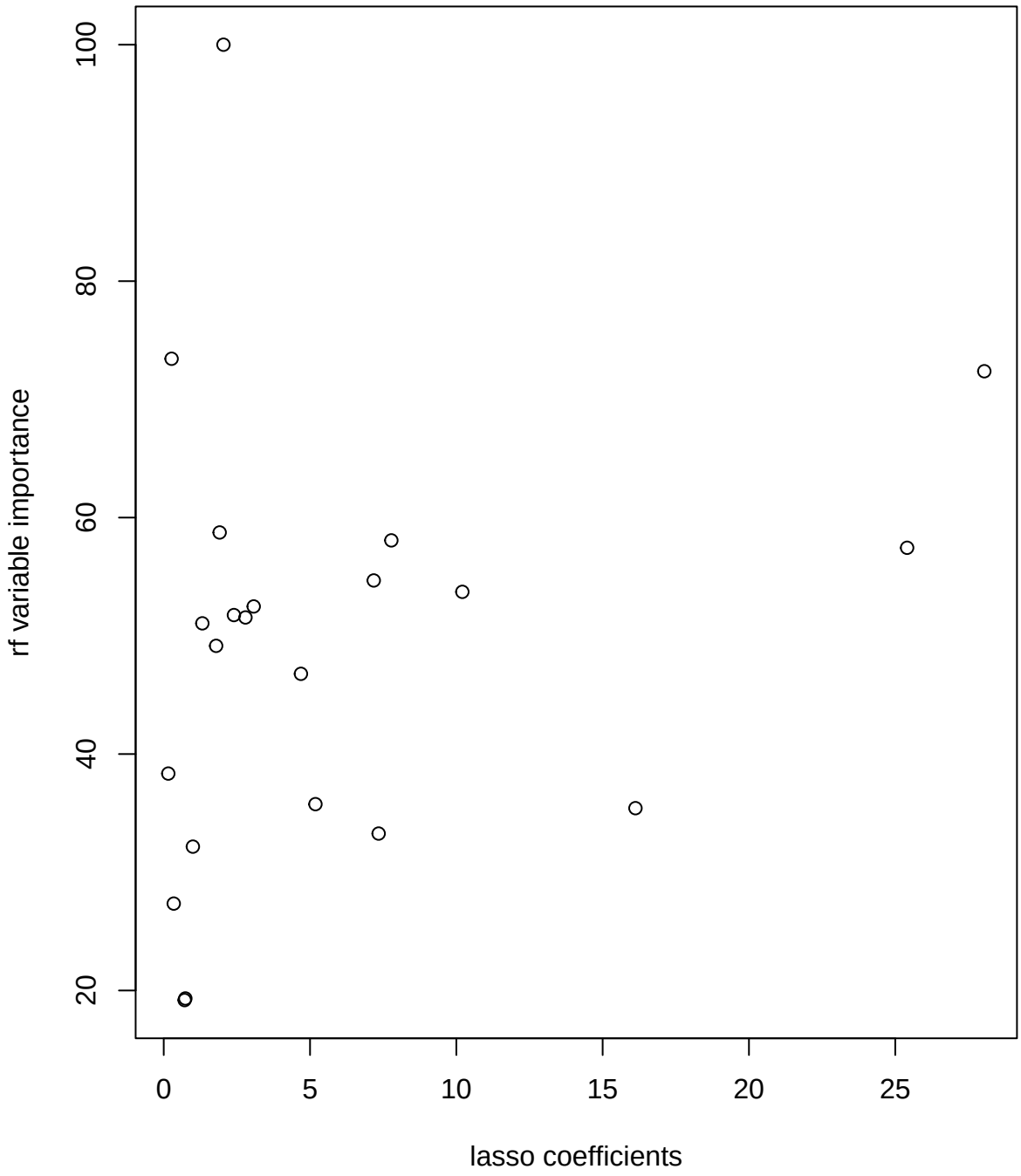
```
  # print(outcome)
  lasso_coefs <- coefs_y2[[outcome]]
  # get rid of intercept
  if (length(lasso_coefs) == 1) {
    next
  }
  lasso_coefs <- lasso_coefs[2:length(lasso_coefs)]
  rf_coefs <- varimp_y2[[outcome]]
  # this will be in the order of the coefficients in the lasso model
  overlap <- rf_coefs[names(lasso_coefs),]
  x <- rep(NA, length(lasso_coefs))
  for (i in 1:length(lasso_coefs)) {
    x[i] <- abs(lasso_coefs[[i]])
  }
  plot(x, overlap, xlab='lasso coefficients', ylab='rf variable importance', main=paste('y2', outcome))
}
```


y2 gain_c_pt_pcorrect
rf variable importance
100
90
80
70
60
50
40
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
lasso coefficients


y2 gain_c_ltr_emo_comp
rf variable importance
45
50
55
60
65
0
50
100
150
lasso coefficients


y2 gain_c_pra_total
rf variable importance
20
30
40
50
60
70
80
0
1
2
3
4
lasso coefficients


y2 gain_c_pbsa_total
rf variable importance
50
60
70
80
0.0
0.5
1.0
1.5
2.0
lasso coefficients




```
# Try a model with a bunch of predictors from Y1 Y2 -- question: what would be predicted, then?
# since it looks like the gains are different from year to year
```

```
#3. Use the particular caretype from the  setting that is being observed in. (prov_type)
#Collapse CC-Community Based & CC- License Exempt
```

```
#4. Drop all variances and instead focus on means
```

```
#5. Incorporte child-level covariates

#6. Age and elapsed time for assessment
```