Preliminary ELSA Machine Learnings Tests

Jonathan Seiden

8/18/2021

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")

#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()) %>%
  select(c(classid, nsweeps, o_c_verbal_talk:o_c_focus_none)) %>%
  summarize(across(everything(), list(mean)))  %>%
  rename_with( ~ str_remove(., '_1')) %>%
  filter(nsweeps >= 10 ) %>% #THIS IS AN ARBITRARY PARAMETER
  group_by(classid) %>%
  mutate(
    nclass = n(),
    #This ugly mess omits the observation of the child when calculating the mean and variance of the ch
    o_c_verbal_talk_classmean = map_dbl(row_number(), ~ mean(o_c_verbal_talk[-.x])),
    o_c_verbal_talk_classvar = map_dbl(row_number(), ~ sd(o_c_verbal_talk[-.x])),
    o_c_verbal_listen_classmean = map_dbl(row_number(), ~ mean(o_c_verbal_listen[-.x])),
    o_c_verbal_listen_classvar = map_dbl(row_number(), ~ sd(o_c_verbal_listen[-.x])),
```

```
    o_c_talk_adult_classmean = map_dbl(row_number(), ~ mean(o_c_talk_adult[-.x])),
    o_c_talk_adult_classvar = map_dbl(row_number(), ~ sd(o_c_talk_adult[-.x])),
    o_c_listen_adult_classmean = map_dbl(row_number(), ~ mean(o_c_listen_adult[-.x])),
    o_c_listen_adult_classvar = map_dbl(row_number(), ~ sd(o_c_listen_adult[-.x])),
    o_c_transition_classmean = map_dbl(row_number(), ~ mean(o_c_transition[-.x])),
    o_c_transition_classvar = map_dbl(row_number(), ~ sd(o_c_transition[-.x])),
    o_c_associative_classmean = map_dbl(row_number(), ~ mean(o_c_associative[-.x])),
    o_c_associative_classvar = map_dbl(row_number(), ~ sd(o_c_associative[-.x])),
    o_c_cooperative_classmean = map_dbl(row_number(), ~ mean(o_c_cooperative[-.x])),
    o_c_cooperative_classvar = map_dbl(row_number(), ~ sd(o_c_cooperative[-.x])),
    o_c_sequential_classmean = map_dbl(row_number(), ~ mean(o_c_sequential[-.x])),
    o_c_sequential_classvar = map_dbl(row_number(), ~ sd(o_c_sequential[-.x])),
    o_c_passive_instr_classmean = map_dbl(row_number(), ~ mean(o_c_passive_instr[-.x])),
    o_c_passive_instr_classvar = map_dbl(row_number(), ~ sd(o_c_passive_instr[-.x])),
    o_c_involve_low_classmean = map_dbl(row_number(), ~ mean(o_c_involve_low[-.x])),
    o_c_involve_low_classvar = map_dbl(row_number(), ~ sd(o_c_involve_low[-.x])),
    o_c_focus_math_classmean = map_dbl(row_number(), ~ mean(o_c_focus_math[-.x])),
    o_c_focus_math_classvar = map_dbl(row_number(), ~ sd(o_c_focus_math[-.x])),
    o_c_focus_read_classmean = map_dbl(row_number(), ~ mean(o_c_focus_read[-.x])),
    o_c_focus_read_classvar = map_dbl(row_number(), ~ sd(o_c_focus_read[-.x])),
  ) %>%
  ungroup
```

```
## Adding missing grouping variables: `cid`
```

```
## `summarise()` has grouped output by 'cid'. 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 %>%
  group_by(o_t_uniqueid , classid) %>%
  mutate(nsweeps = n()) %>% #DO WE WANT CLASS AVERAGES OR AVERAGES OF ADULTS?
  select(c(classid, nsweeps, o_t_focus_ll, o_t_focus_math, o_t_focus_n,
           o_t_tone_pos, o_t_tone_neg, o_t_tone_flat, o_t_attention_s,
           o_t_es_s, o_t_task_i, o_t_sched_t)) %>%
  summarize(across(everything(), list(mean)))  %>%
  rename_with( ~ str_remove(., '_1')) %>%
  group_by(classid) %>%
  summarize(
    nadult = n(),
    o_t_focus_ll = mean(o_t_focus_ll),
    o_t_focus_math = mean(o_t_focus_math),
    o_t_focus_n = mean(o_t_focus_n),
    o_t_tone_pos = mean(o_t_tone_pos),
    o_t_tone_neg = mean(o_t_tone_neg),
    o_t_tone_flat = mean(o_t_tone_flat),
    o_t_attention_s = mean(o_t_attention_s),
    o_t_es_s = mean(o_t_es_s),
    o_t_task_i = mean(o_t_task_i),
    o_t_sched_t = mean(o_t_sched_t)) %>%
  ungroup
```

```
## Adding missing grouping variables: `o_t_uniqueid`
```

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

```
#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
```

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)
```

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
outcomes_and_obs <- left_join(child_outcomes, y1_obs, by = "cid")
```

```r
#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 <- left_join(outcomes_and_obs, caretype, by = "cid") %>%
  mutate(hasobservation = is.na(classid)) %>%
  filter(!is.na(caretype))

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

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

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

#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), 0, var)
}
names(outcomes_and_obs_full)
```

```
##  [1] "cid"                          "gain_c_mefs_str"
##  [3] "gain_c_pt_pcorrect"           "gain_c_ltr_cogsoc_comp"
##  [5] "gain_c_ltr_emo_comp"          "gain_c_pra_total"
##  [7] "gain_c_pbsa_total"            "gain_c_quils_total_raw"
##  [9] "gain_c_wjlw_str"              "gain_c_wjap_str"
## [11] "classid"                      "nsweeps"
## [13] "o_c_verbal_talk"              "o_c_verbal_listen"
## [15] "o_c_talk_adult"               "o_c_listen_adult"
## [17] "o_c_transition"               "o_c_associative"
## [19] "o_c_cooperative"              "o_c_sequential"
## [21] "o_c_passive_instr"            "o_c_involve_low"
## [23] "o_c_focus_math"               "o_c_focus_read"
## [25] "o_c_focus_none"               "nclass"
## [27] "o_c_verbal_talk_classmean"    "o_c_verbal_talk_classvar"
## [29] "o_c_verbal_listen_classmean"  "o_c_verbal_listen_classvar"
## [31] "o_c_talk_adult_classmean"     "o_c_talk_adult_classvar"
## [33] "o_c_listen_adult_classmean"   "o_c_listen_adult_classvar"
## [35] "o_c_transition_classmean"     "o_c_transition_classvar"
## [37] "o_c_associative_classmean"    "o_c_associative_classvar"
## [39] "o_c_cooperative_classmean"    "o_c_cooperative_classvar"
## [41] "o_c_sequential_classmean"     "o_c_sequential_classvar"
## [43] "o_c_passive_instr_classmean"  "o_c_passive_instr_classvar"
## [45] "o_c_involve_low_classmean"    "o_c_involve_low_classvar"
## [47] "o_c_focus_math_classmean"     "o_c_focus_math_classvar"
## [49] "o_c_focus_read_classmean"     "o_c_focus_read_classvar"
```

```
## [51] "nadult"                    "o_t_focus_ll"
## [53] "o_t_focus_math"            "o_t_focus_n"
## [55] "o_t_tone_pos"              "o_t_tone_neg"
## [57] "o_t_tone_flat"             "o_t_attention_s"
## [59] "o_t_es_s"                  "o_t_task_i"
## [61] "o_t_sched_t"               "c_HHS"
## [63] "caretype"                  "actualtype"
```

```
outcomes_and_obs_full <- outcomes_and_obs_full %>%
  mutate_at(vars(o_c_verbal_talk:o_t_sched_t), replace.na)

dim(outcomes_and_obs_full)
```

```
## [1] 1169   64
```

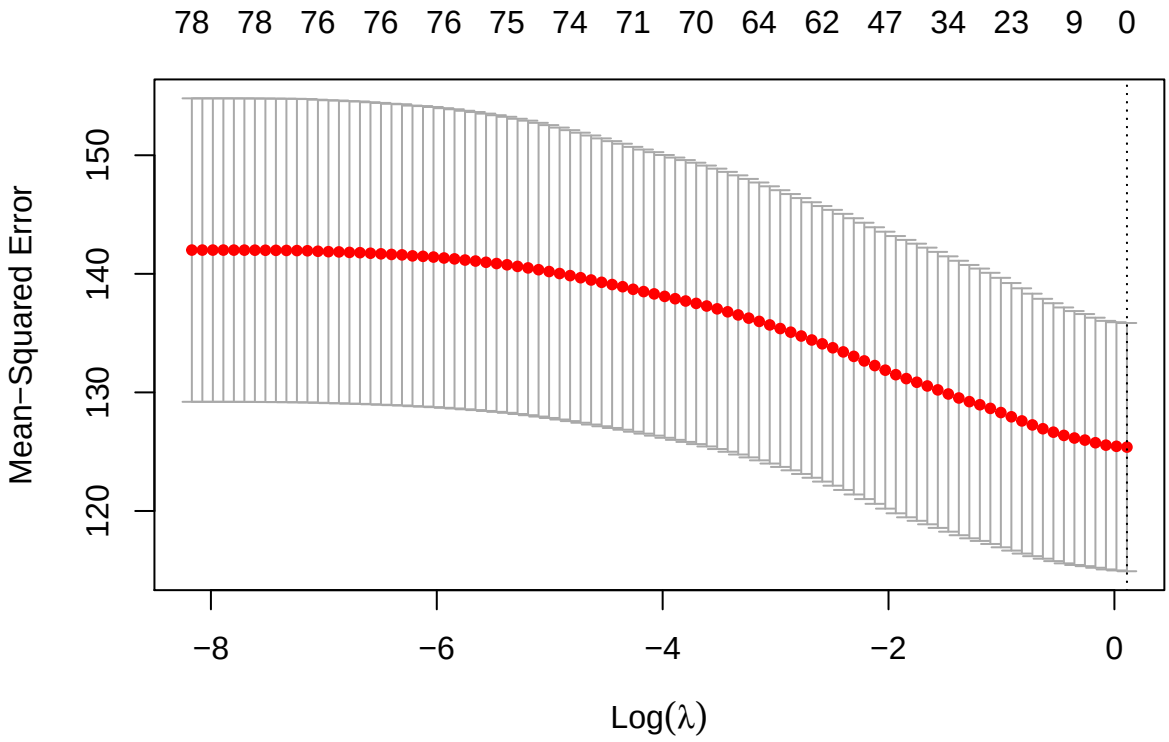
Analysis

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)

#Subset to non-missing gain scores for the MEFS
gain_c_mefs_str_analysis <- outcomes_and_obs_full %>%
  filter(!is.na(gain_c_mefs_str)) %>%
  select(c(gain_c_mefs_str, o_c_verbal_talk:actualtype))

#Make the model matrix
x = model.matrix(gain_c_mefs_str ~ . ,
                 data=gain_c_mefs_str_analysis)
dim(x)
```

```
## [1] 726  83
```

```
y = gain_c_mefs_str_analysis$gain_c_mefs_str
x = x[, -1]

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




```
model_lasso
```

```
##
## Call:  cv.glmnet(x = x, y = y, alpha = 1)
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure    SE Nonzero
## min  1.118     1   125.4 10.47       0
## 1se  1.118     1   125.4 10.47       0
```

```
# identify the best choice lambda (using the Minimum Lambda rule)
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]
cc
```

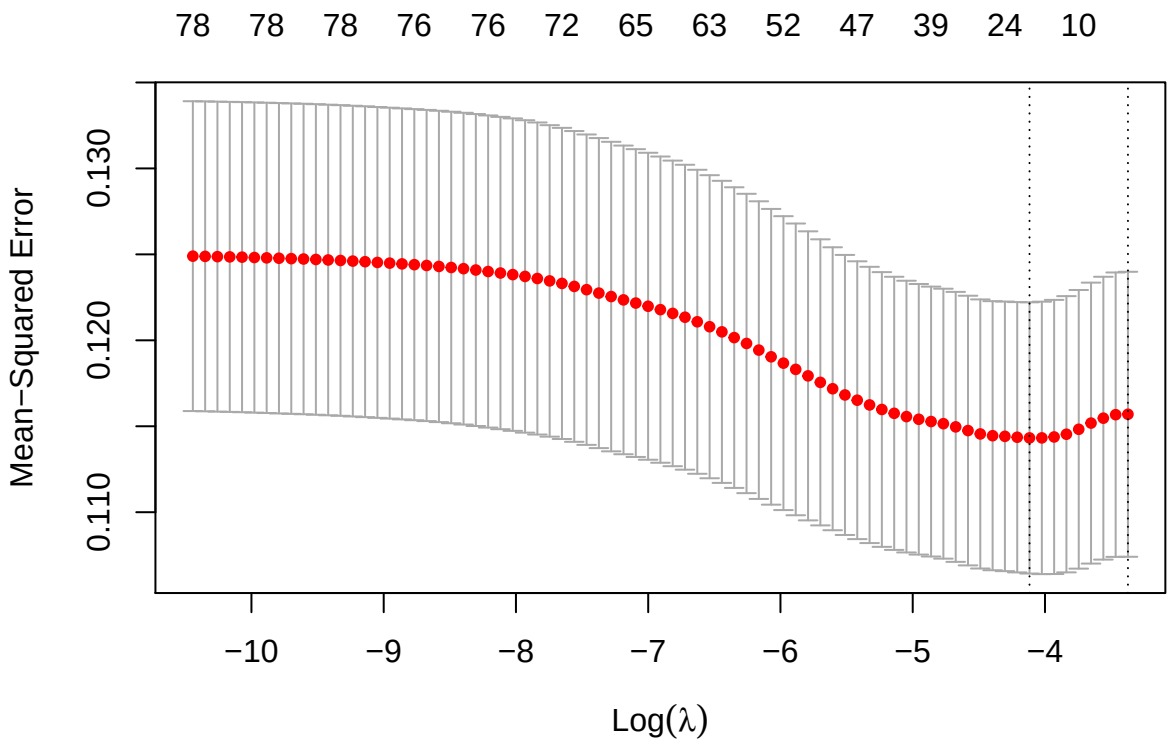
```
## [1] 2.363636
```

This is not very satisfying! This is suggesting that *nothing* in our dataset is doing a particularly good job of predicting the gain in the MEFS score. Let's try with another variable.

```r
#Subset to non-missing gain scores for the MEFS
gain_c_pt_pcorrect_analysis <- outcomes_and_obs_full %>%
  filter(!is.na(gain_c_pt_pcorrect)) %>%
  select(c(gain_c_pt_pcorrect, o_c_verbal_talk:actualtype))

#Make the model matrix
x = model.matrix(gain_c_pt_pcorrect ~ . ,
                 data=gain_c_pt_pcorrect_analysis)
dim(x)
```

```
## [1] 845  83
```

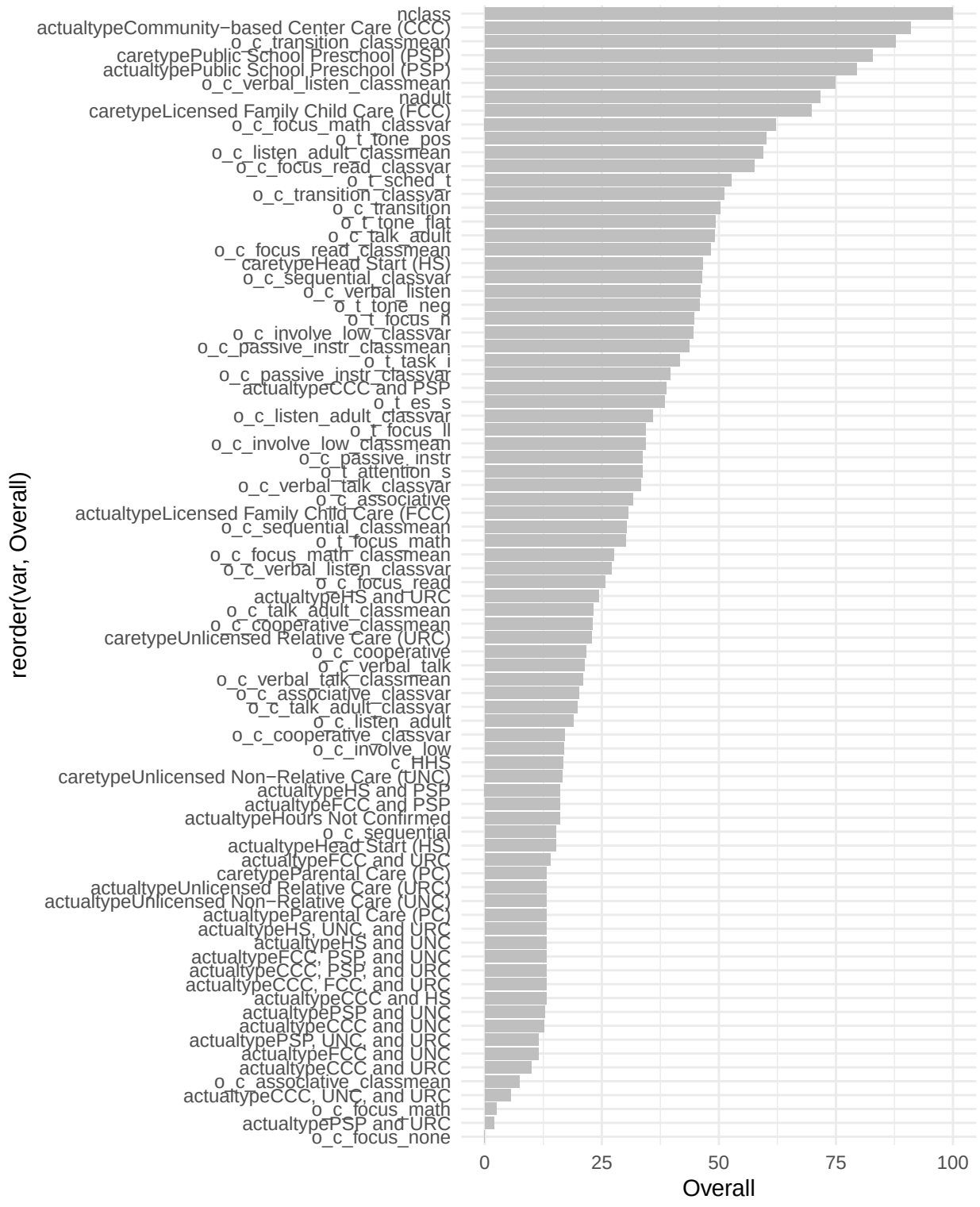
```r
y = gain_c_pt_pcorrect_analysis$gain_c_pt_pcorrect
x = x[, -1]

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




```r
# identify the best choice lambda (use the 1 SE rule)
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]
sort(cc)
```

```
##                o_c_focus_math_classvar                o_c_transition_classvar
##                          -0.4614455120                          -0.4562477370
##                o_c_focus_read_classvar                     o_c_verbal_listen
##                          -0.2080819563                          -0.0770074065
##                   actualtypeCCC and HS                       o_c_cooperative
##                          -0.0751244698                          -0.0324267550
##  caretypePublic School Preschool (PSP)              o_c_involve_low_classmean
##                          -0.0282745947                          -0.0188485467
##                            o_t_focus_n                                 nclass
##                          -0.0107189376                          -0.0021546157
##                            o_t_task_i                caretypeHead Start (HS)
##                           0.0000778147                           0.0009882915
##              o_c_verbal_talk_classmean caretypeUnlicensed Relative Care (URC)
##                           0.0051038991                           0.0162615855
##                         o_t_attention_s                    actualtypeHS and URC
##                           0.0212640150                           0.1211850336
##                   actualtypeCCC and PSP            actualtypeCCC, PSP, and URC
##                           0.1970001976                           0.2072334356
##              actualtypeHS, UNC, and URC                            (Intercept)
##                           0.3120208103                           0.3524809886
```

```r
# Fit the random forest model
rf_fit <- train(gain_c_pt_pcorrect ~ ., #Use all variables in the prediction
                data = gain_c_pt_pcorrect_analysis, #Use the training data
                method = "ranger",
                importance = "permutation",
                num.trees = 500)

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()
```


nclass
actualtypeCommunity−based Center Care (CCC)
o_c_transition_classmean
caretypePublic School Preschool (PSP)
actualtypePublic School Preschool (PSP)
o_c_verbal_listen_classmean
nadult
caretypeLicensed Family Child Care (FCC)
o_c_focus_math_classvar
o_t_tone_pos
o_c_listen_adult_classmean
o_c_focus_read_classvar
o_t_sched_t
o_c_transition_classvar
o_c_transition
o_t_tone_flat
o_c_talk_adult
o_c_focus_read_classmean
caretypeHead Start (HS)
o_c_sequential_classvar
o_c_verbal_listen
o_t_tone_neg
o_t_focus_n
o_c_involve_low_classvar
o_c_passive_instr_classmean
o_t_task_i
o_c_passive_instr_classvar
actualtypeCCC and PSP
o_t_es_s
o_c_listen_adult_classvar
o_t_focus_ll
o_c_involve_low_classmean
o_c_passive_instr
o_t_attention_s
o_c_verbal_talk_classvar
o_c_associative
actualtypeLicensed Family Child Care (FCC)
o_c_sequential_classmean
o_t_focus_math
o_c_focus_math_classmean
o_c_verbal_listen_classvar
o_c_focus_read
actualtypeHS and URC
o_c_talk_adult_classmean
o_c_cooperative_classmean
caretypeUnlicensed Relative Care (URC)
o_c_cooperative
o_c_verbal_talk
o_c_verbal_talk_classmean
o_c_associative_classvar
o_c_talk_adult_classvar
o_c_listen_adult
o_c_cooperative_classvar
o_c_involve_low
c_HHS
caretypeUnlicensed Non−Relative Care (UNC)
actualtypeHS and PSP
actualtypeFCC and PSP
actualtypeHours Not Confirmed
o_c_sequential
actualtypeHead Start (HS)
actualtypeFCC and URC
caretypeParental Care (PC)
actualtypeUnlicensed Relative Care (URC)
actualtypeUnlicensed Non−Relative Care (UNC)
actualtypeParental Care (PC)
actualtypeHS, UNC, and URC
actualtypeHS and UNC
actualtypeFCC, PSP, and UNC
actualtypeCCC, PSP, and URC
actualtypeCCC, FCC, and URC
actualtypeCCC and HS
actualtypePSP and UNC
actualtypeCCC and UNC
actualtypePSP, UNC, and URC
actualtypeFCC and UNC
actualtypeCCC and URC
o_c_associative_classmean
actualtypeCCC, UNC, and URC
o_c_focus_math
actualtypePSP and URC
o_c_focus_none
reorder(var, Overall)
0
25
50
75
100
Overall