Fragile Families Challenge

COS424 Pilot Analysis

Alex Kindel

Princeton University

18 April 2017

Fragile Families Challenge

The Fragile Families Challenge

A scientific mass collaboration combining predictive modeling, causal inference, and qualitative interviews to improve the lives of disadvantaged children in the US.

Your task

Predict six age 15 outcomes (GPA, Grit, Material hardship, Eviction, Job loss, Job training) using a large subset of the Fragile Families data.

Our task

Make sense of the predictions and models you've generously provided us.

Let us know what sounds right and what needs more tuning!

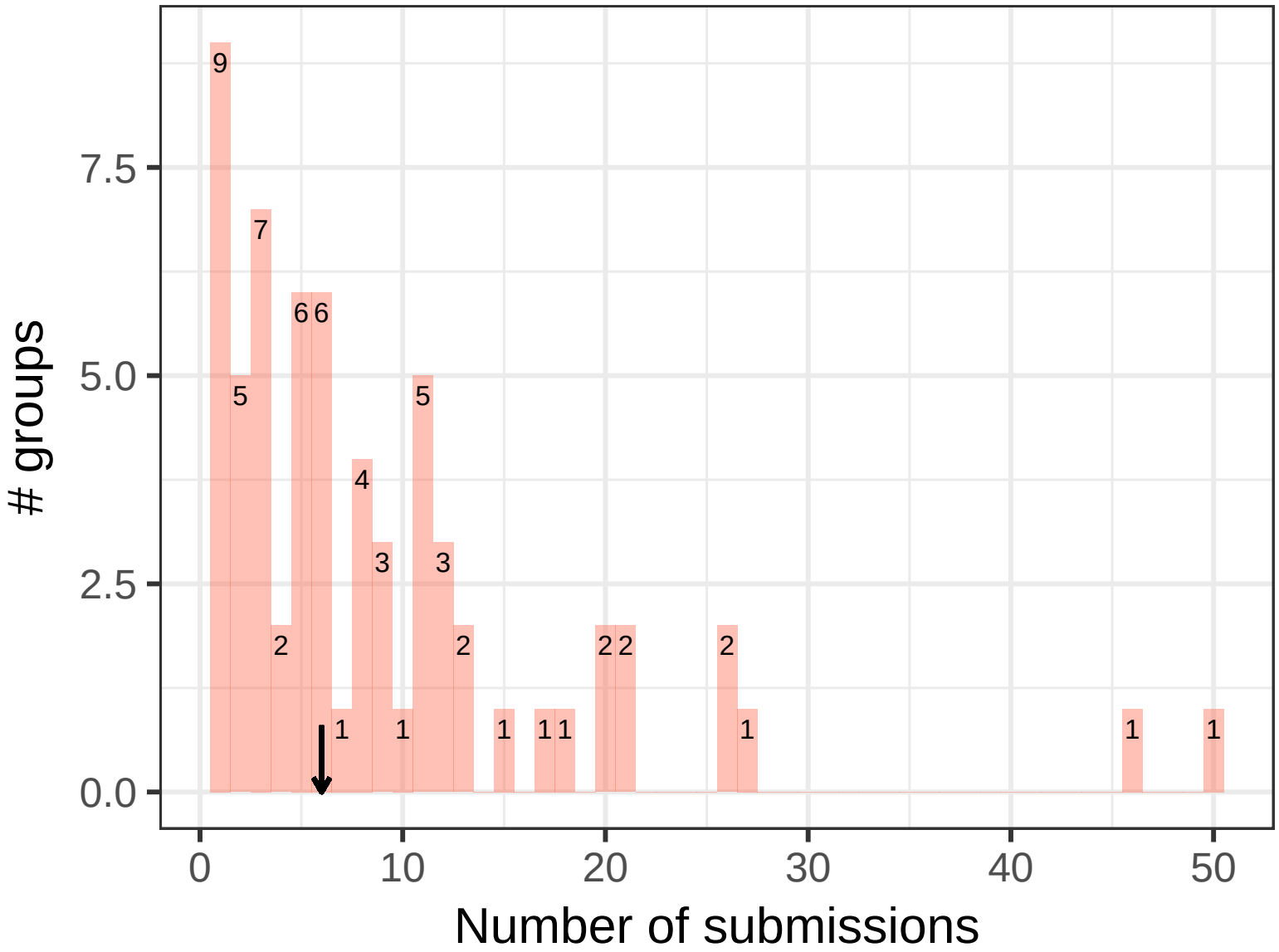
Scores and submissions

COS424 submissions

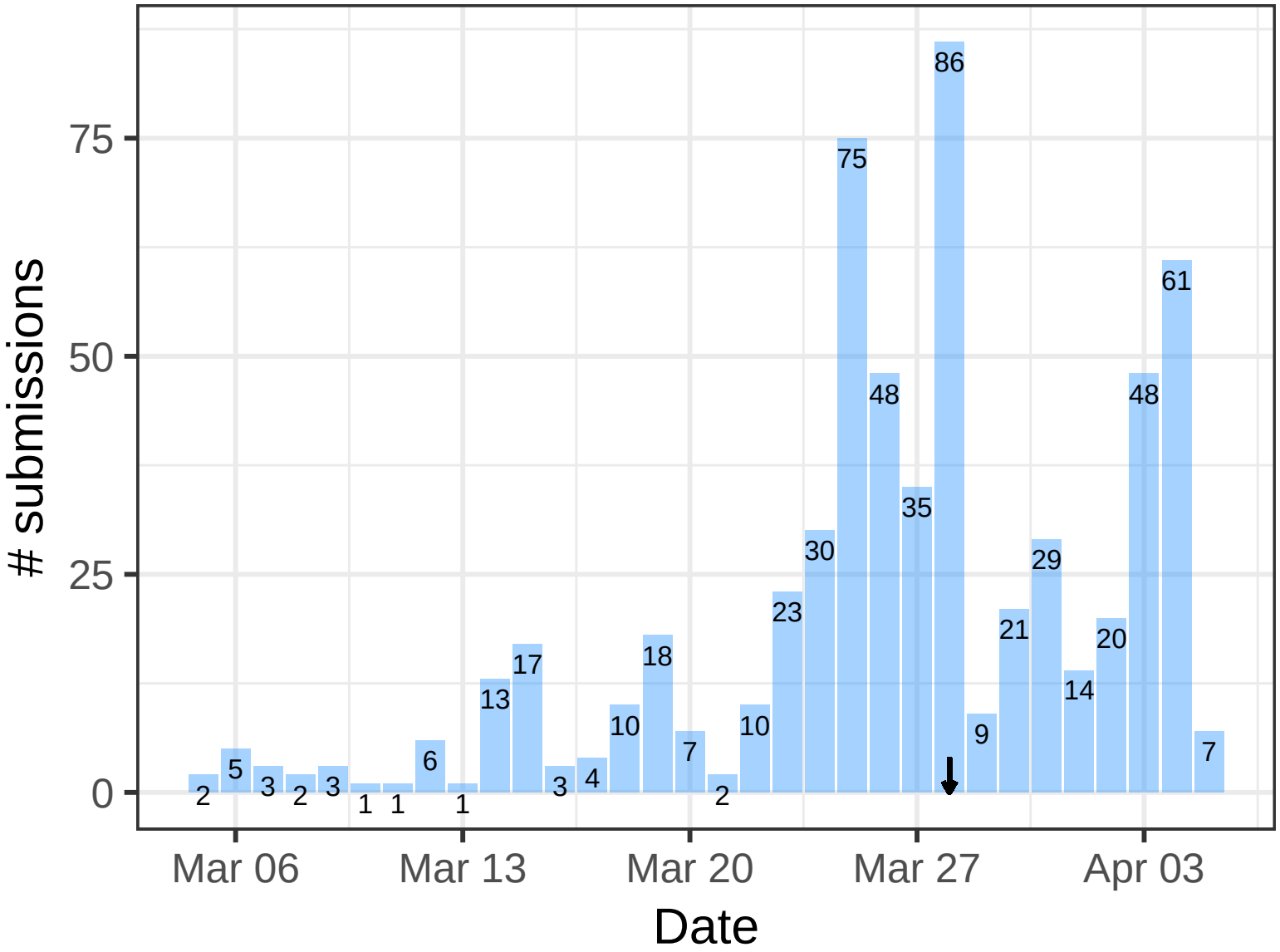
First, how much have COS424 student teams contributed to the challenge?

- Total number of teams: **72**
- Number of teams that submitted successfully: **64**
- Number of leaderboard submissions: **614**

Distribution of submissions per team



Number of submissions by date

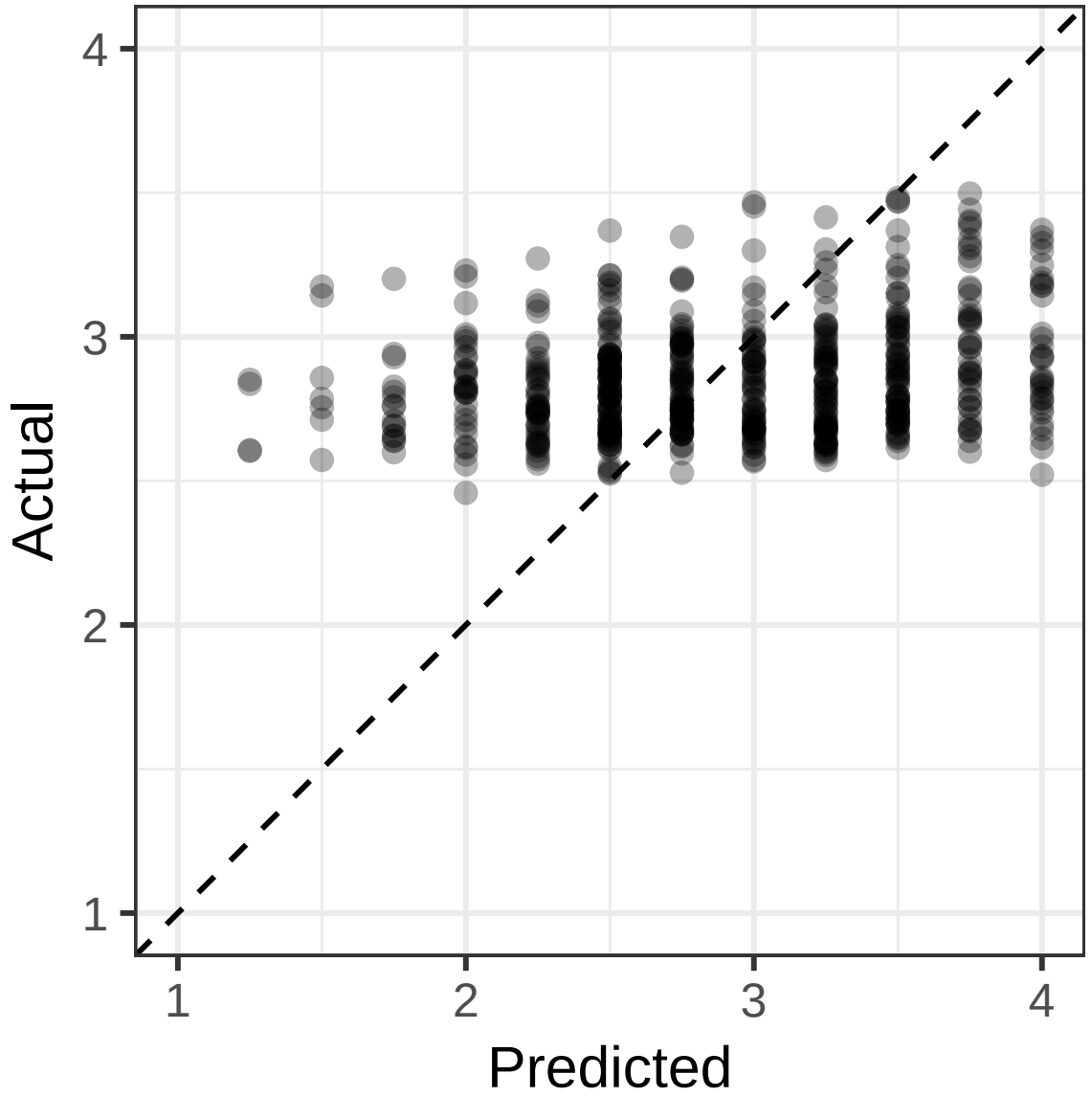


Examining top performers

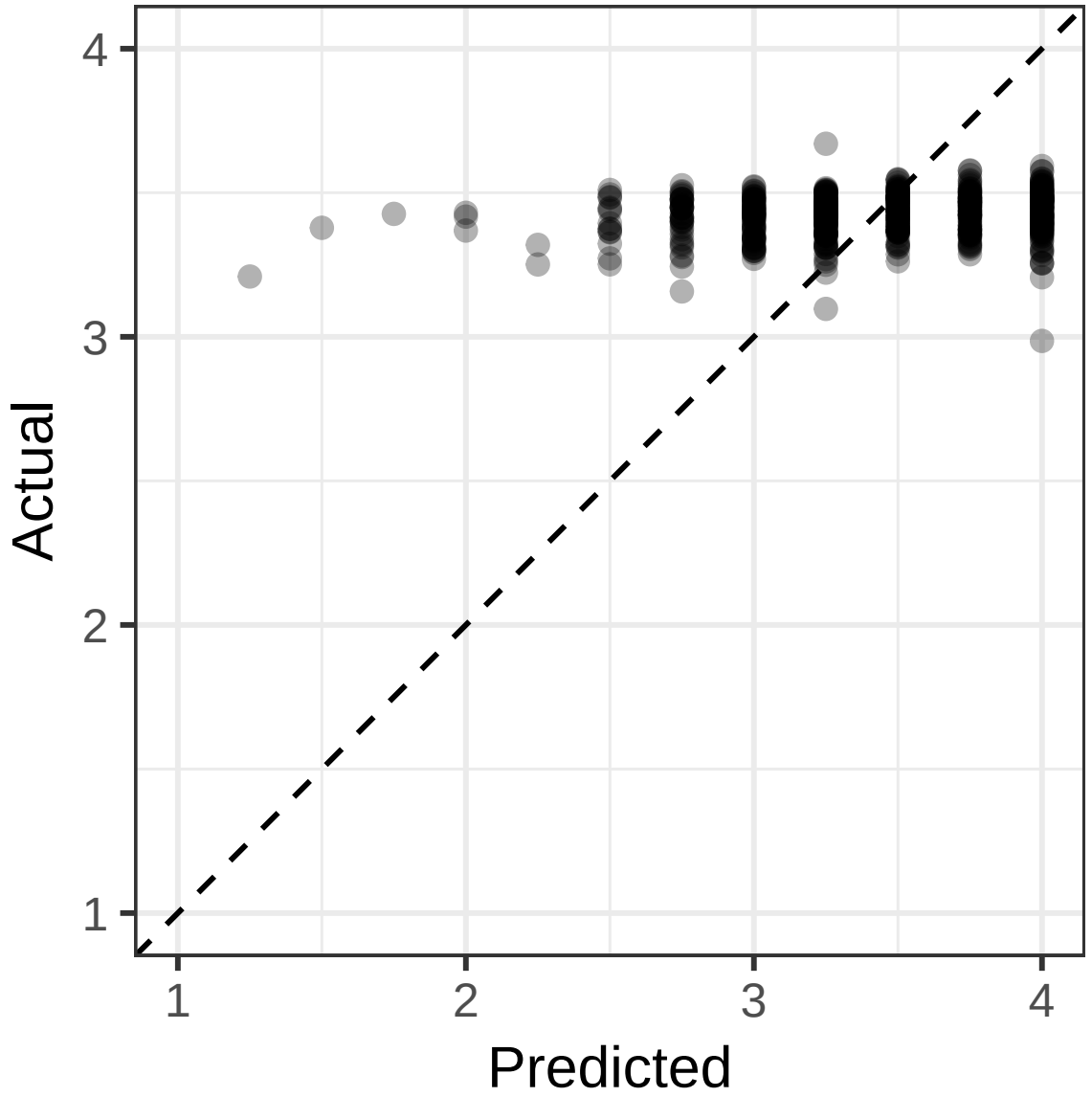
Scores are computed as the mean squared error of predictions on the leaderboard test set. But, this loses some information about how well the predictions are performing.

How do the best (continuous) predictions compare to the actual leaderboard values?

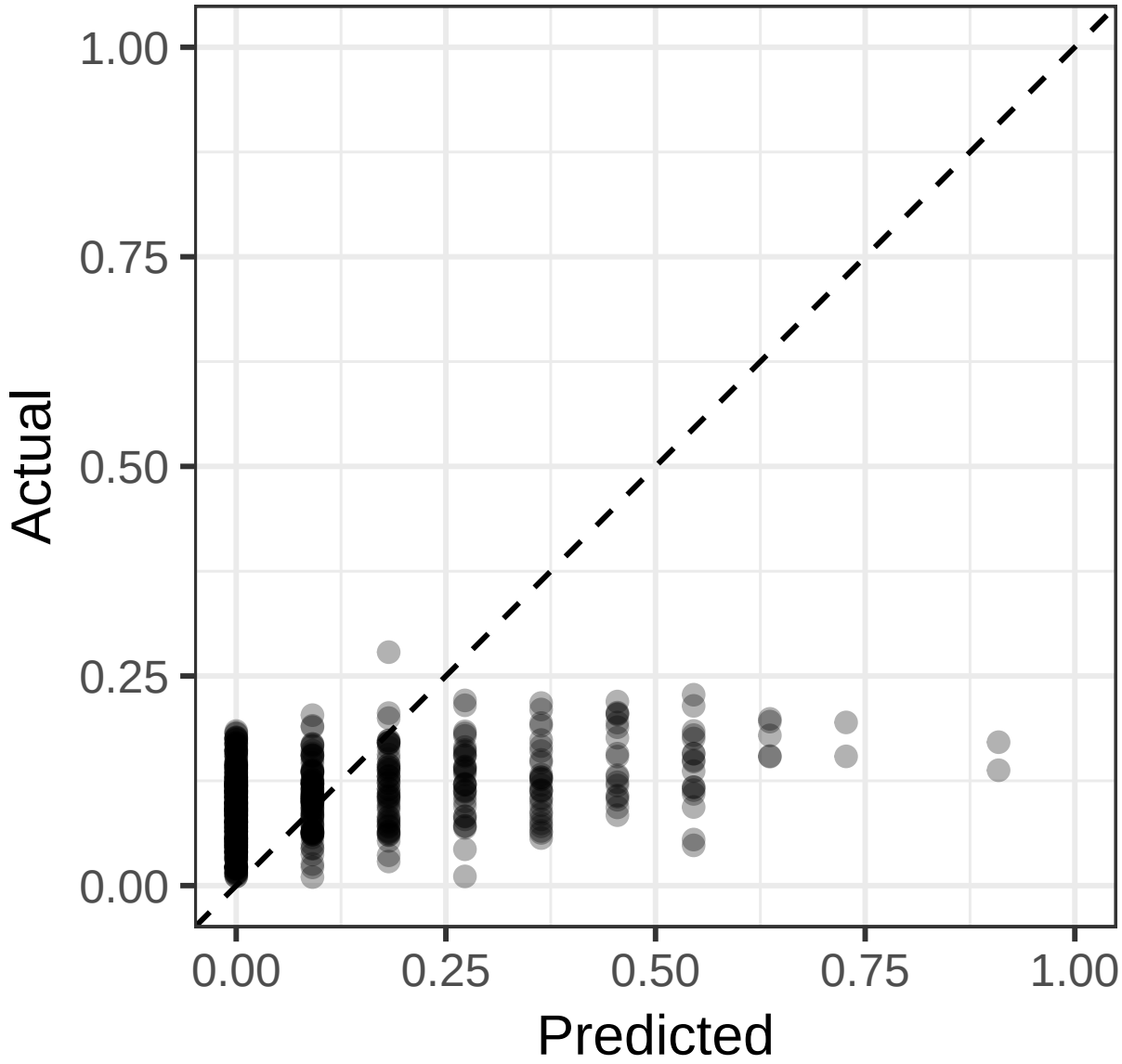
Predicted vs. actual: GPA (best score)



Predicted vs. actual: Grit (best score)



Predicted vs. actual: Material hardship (best score)



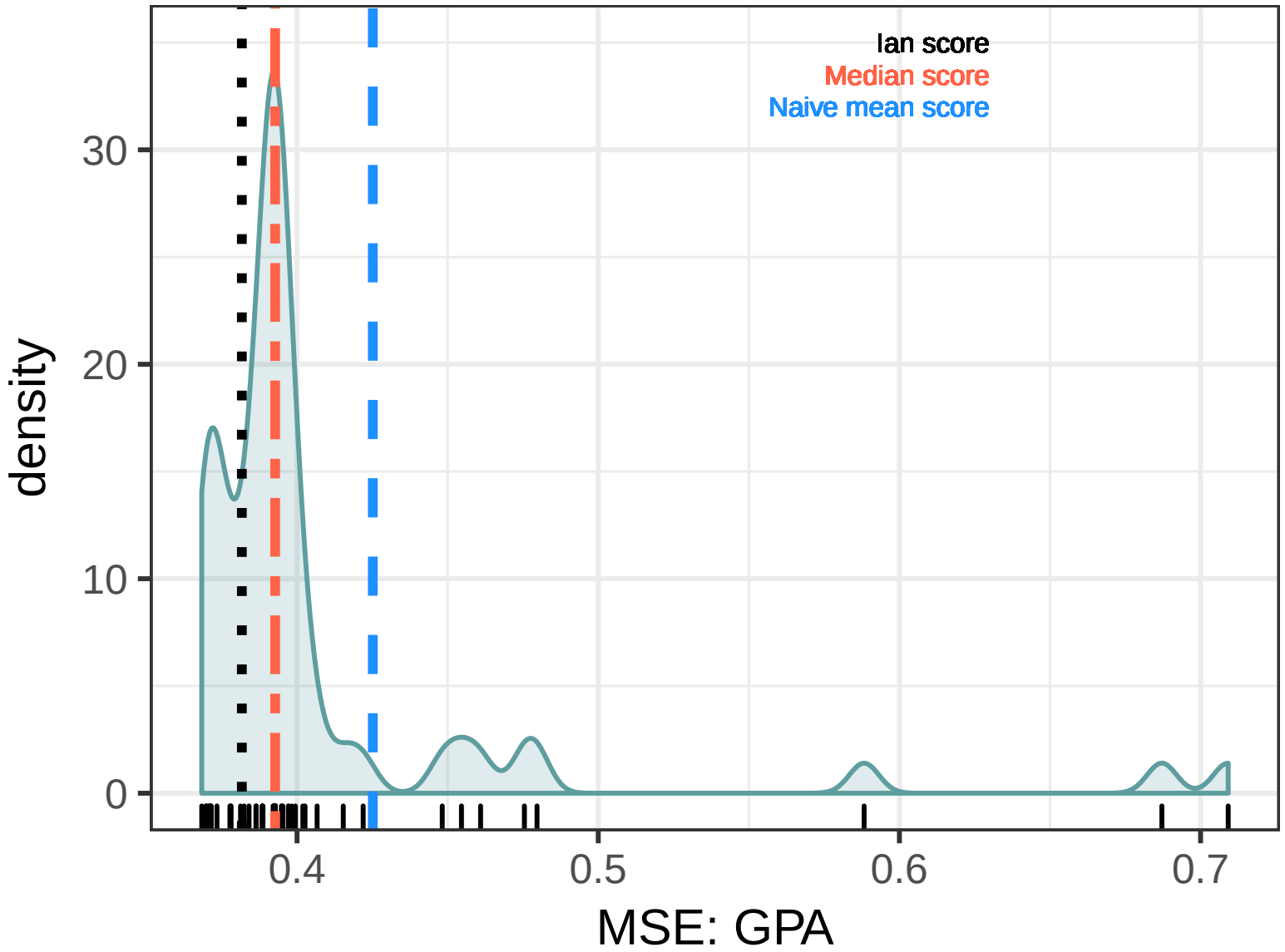
Score distributions by variable

Next, we'll examine the distribution of scores for each outcome variable among COS424 participant teams (`n = 64`).

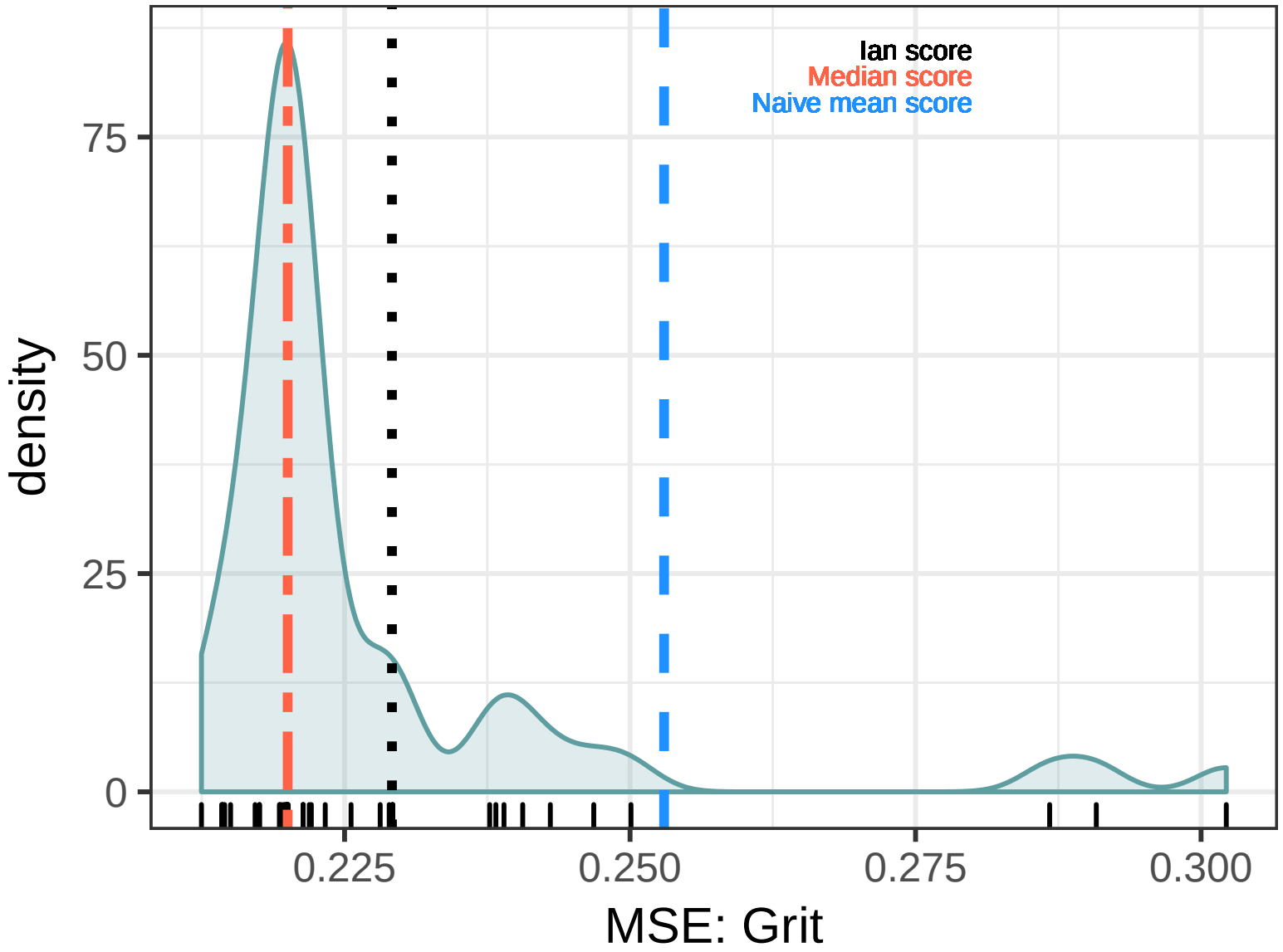
The **red line** indicates the median score among participants. The **blue line** indicates the score you'd get if you predicted the mean outcome for every row. The **dotted line** indicates Ian's benchmark submission score.

We're hoping to see the **red line** to the left of the **blue line** — or ideally, both lines!

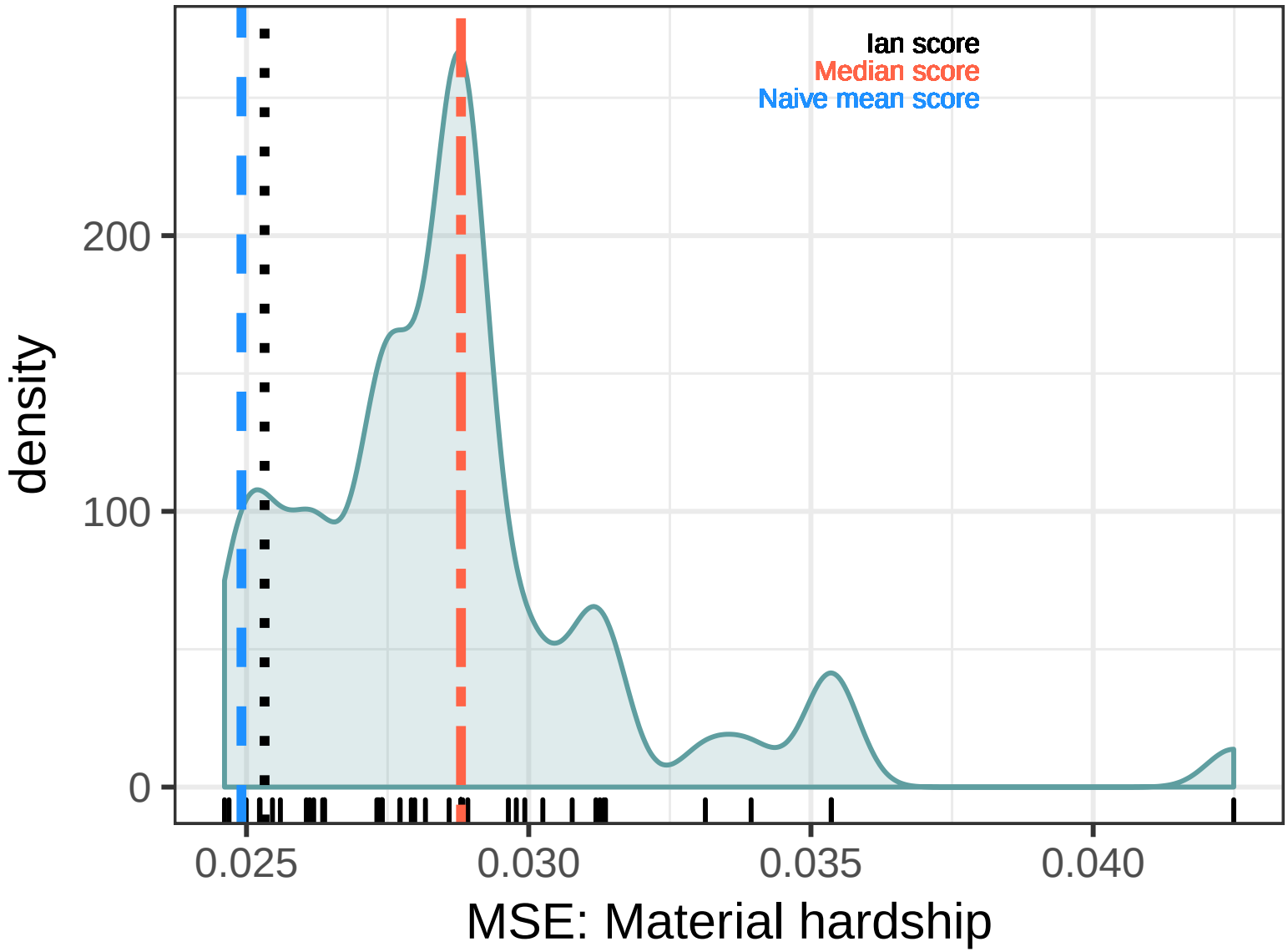
Score distribution: GPA



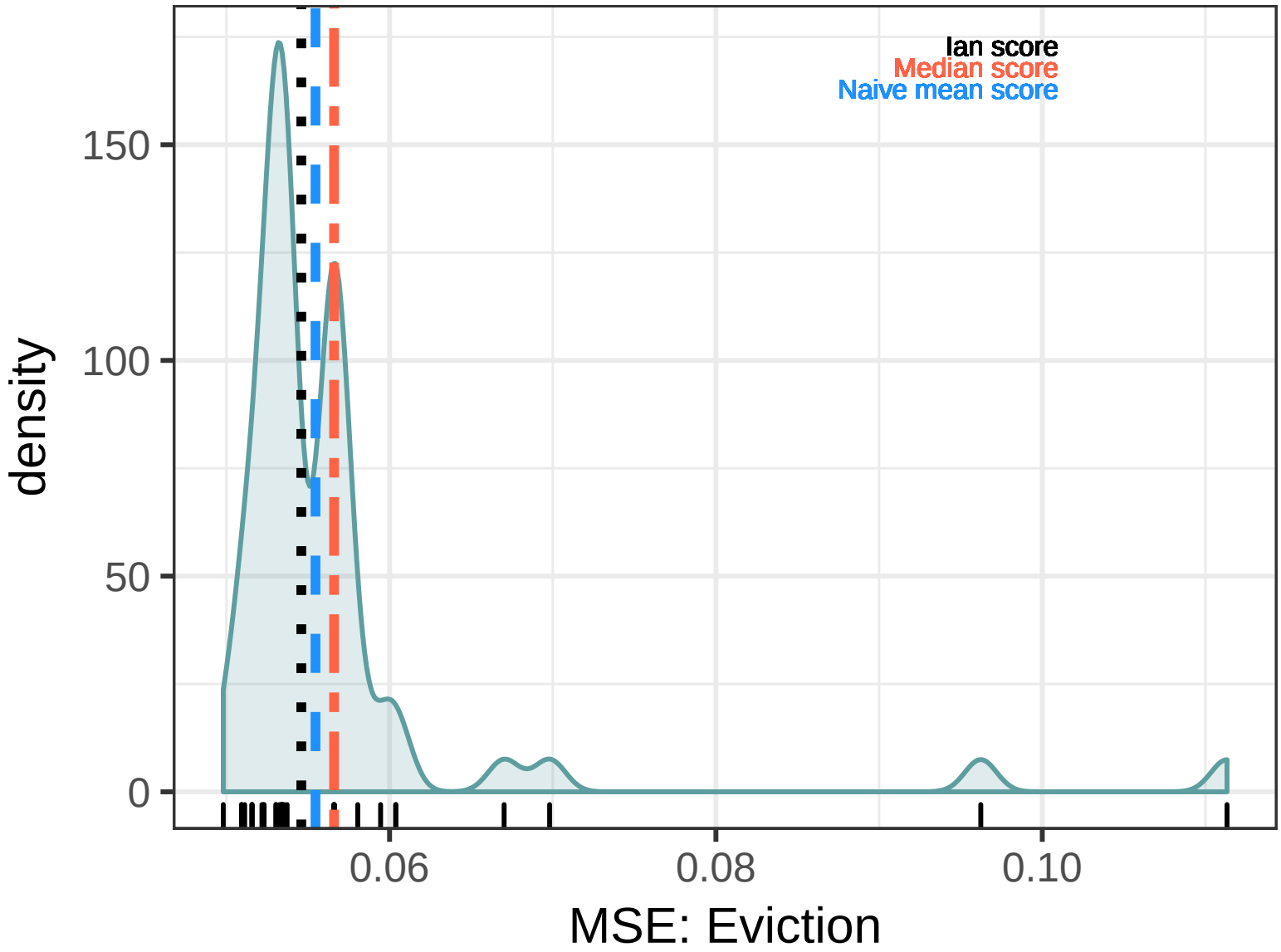
Score distribution: Grit



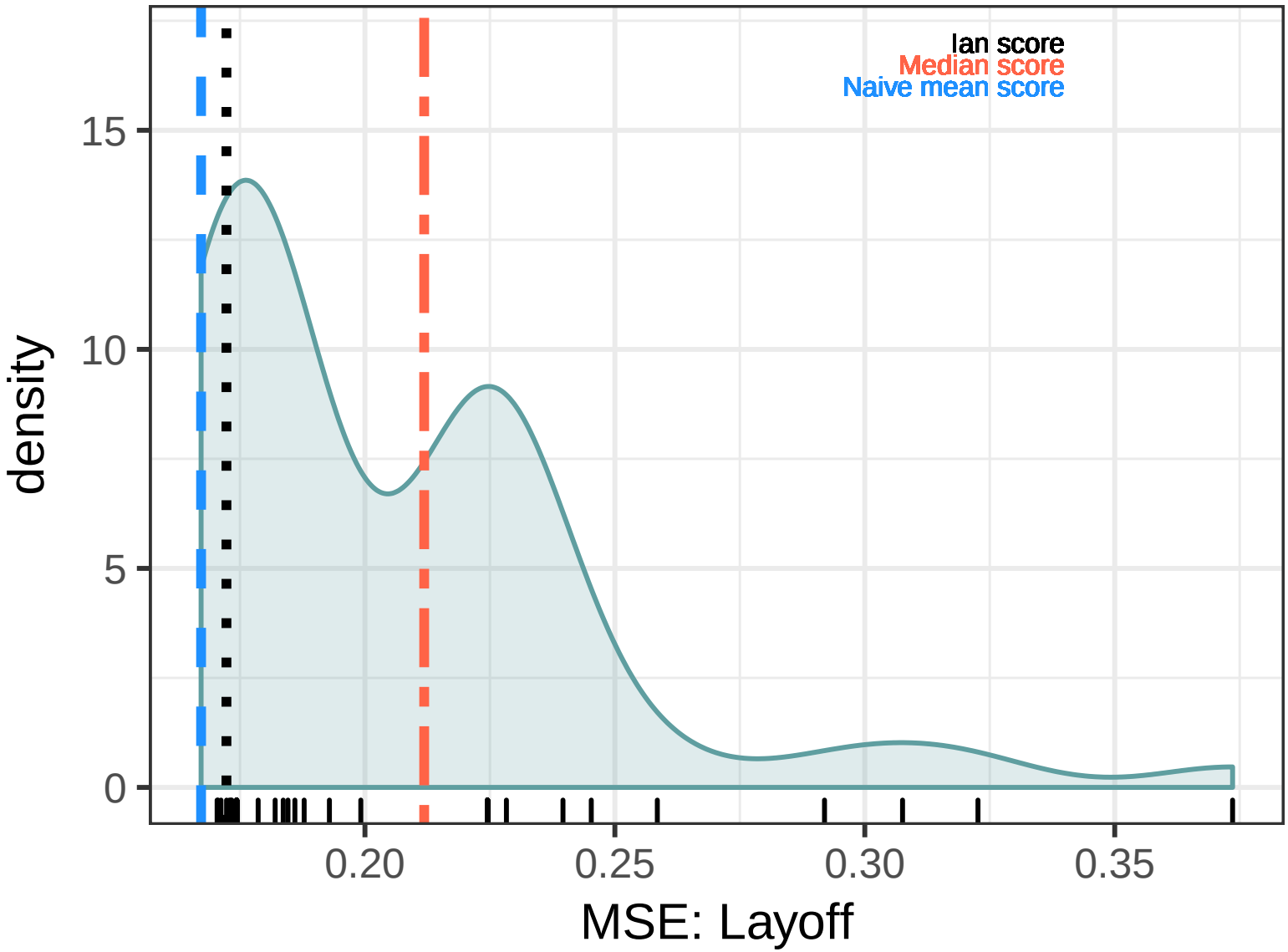
Score distribution: Material hardship



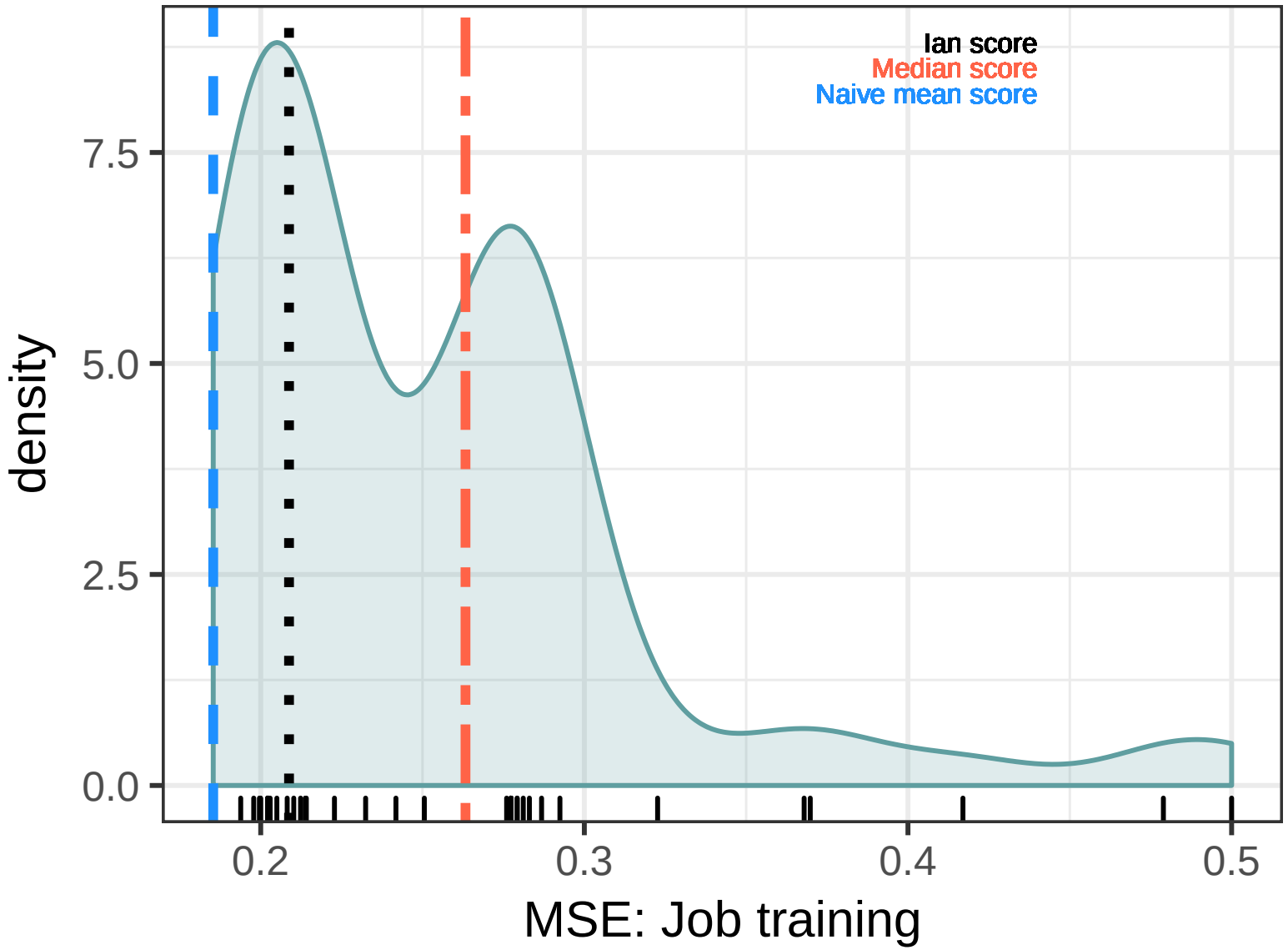
Score distribution: Eviction



Score distribution: Layoff



Score distribution: Job training



Which outcomes are easiest to predict?

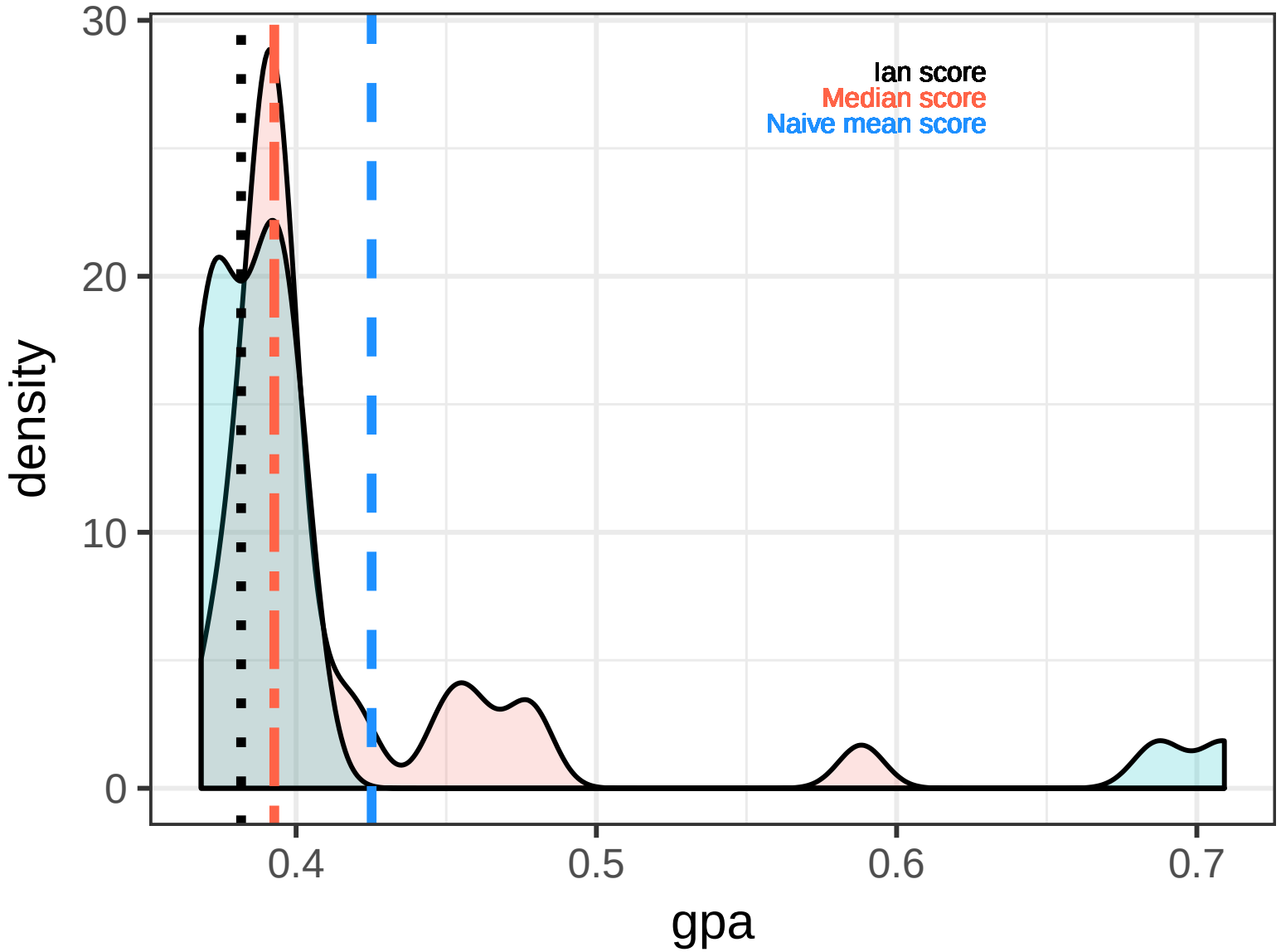
Median MSE by variable:

- GPA: 0.39273
- Grit: 0.22002
- Material hardship: 0.0288
- Eviction: 0.0566
- Layoff: 0.21185
- Job training: 0.263275

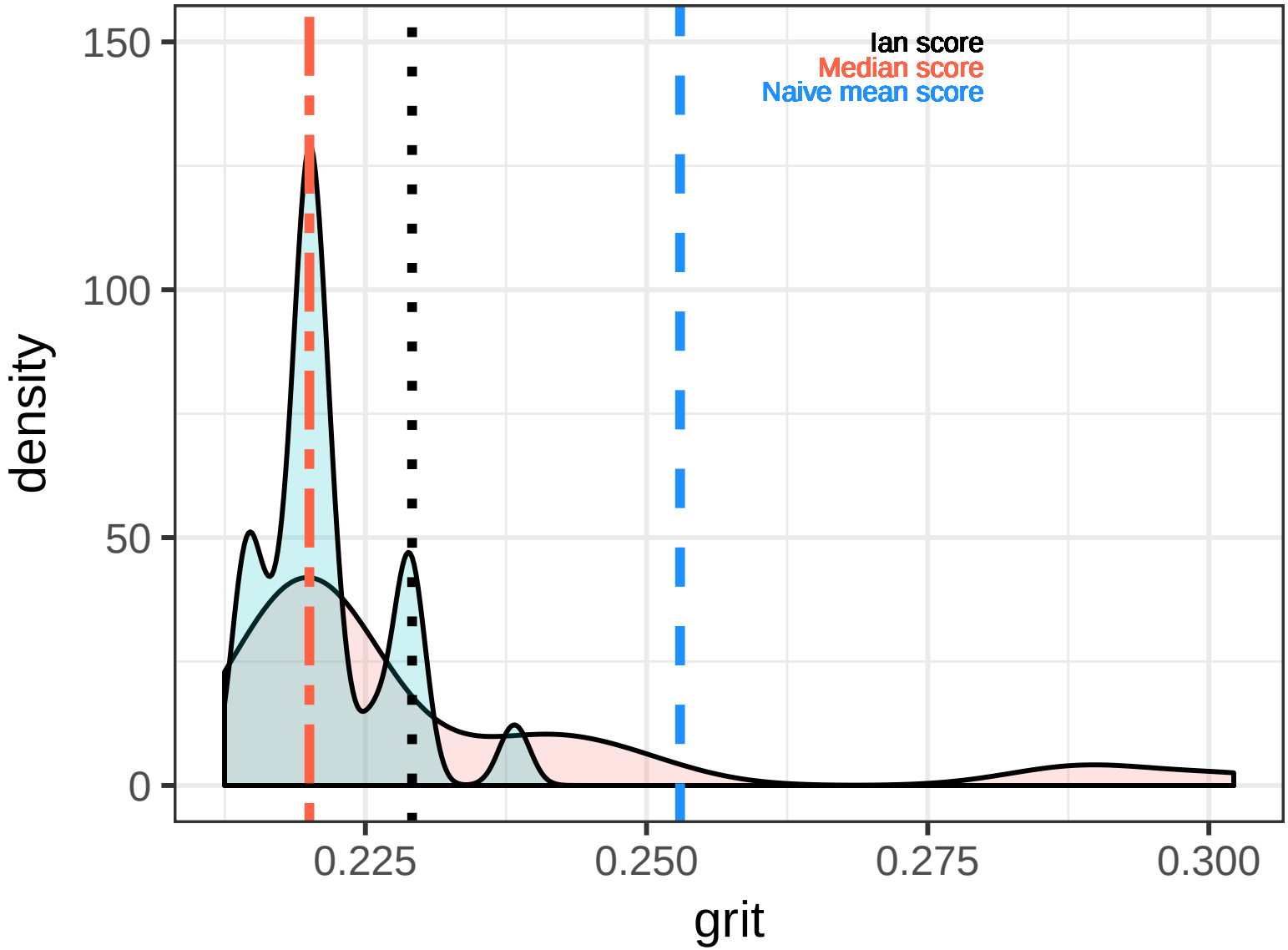
Does it help to submit more?

How did teams who submitted more than the median team (≥ 6 submissions) compare to teams who submitted less often? (Could be overfitting, better model over time, more effort, etc.)

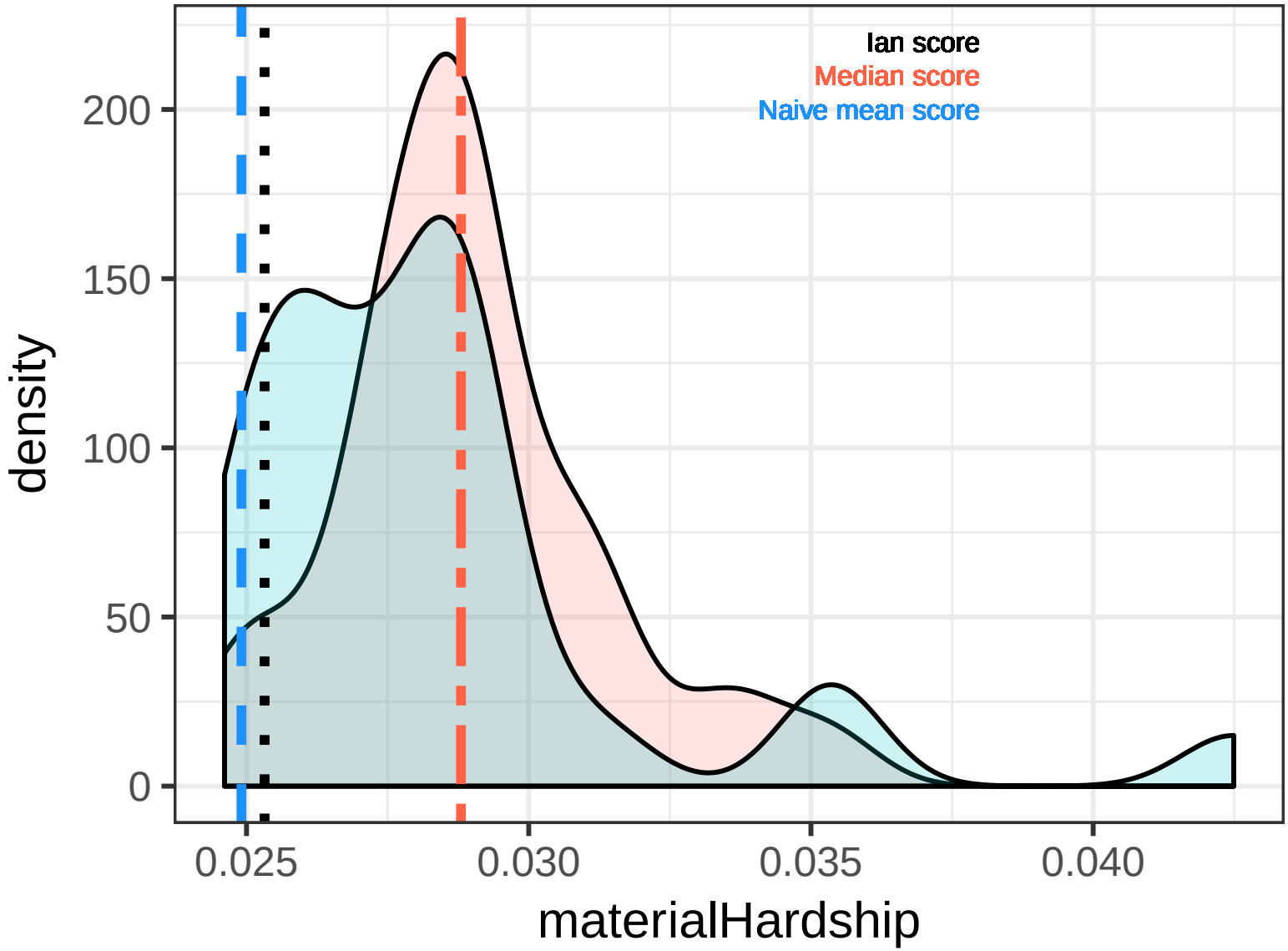
Teams that submit more have lower scores: GPA



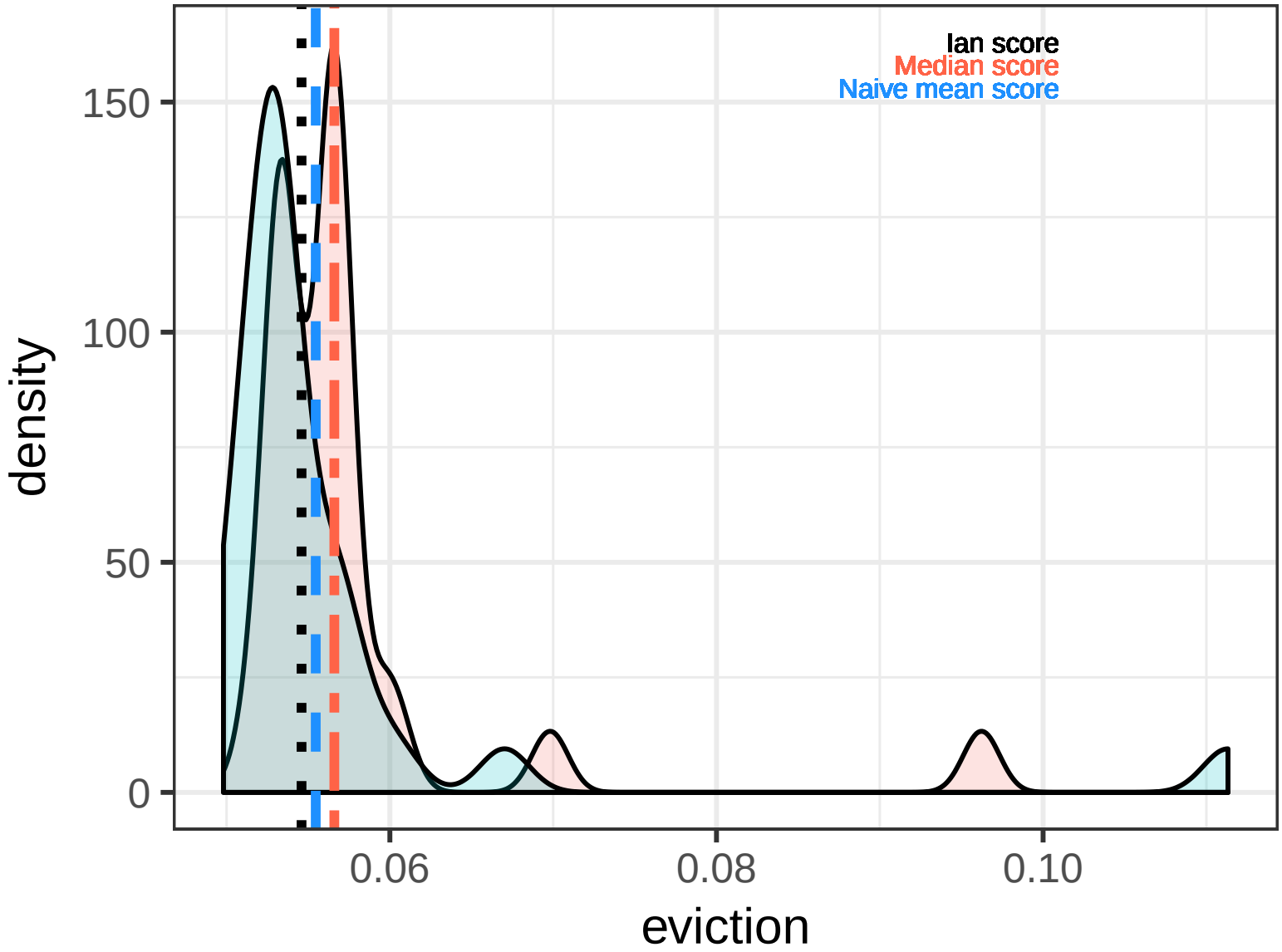
Teams that submit more have lower scores: Grit



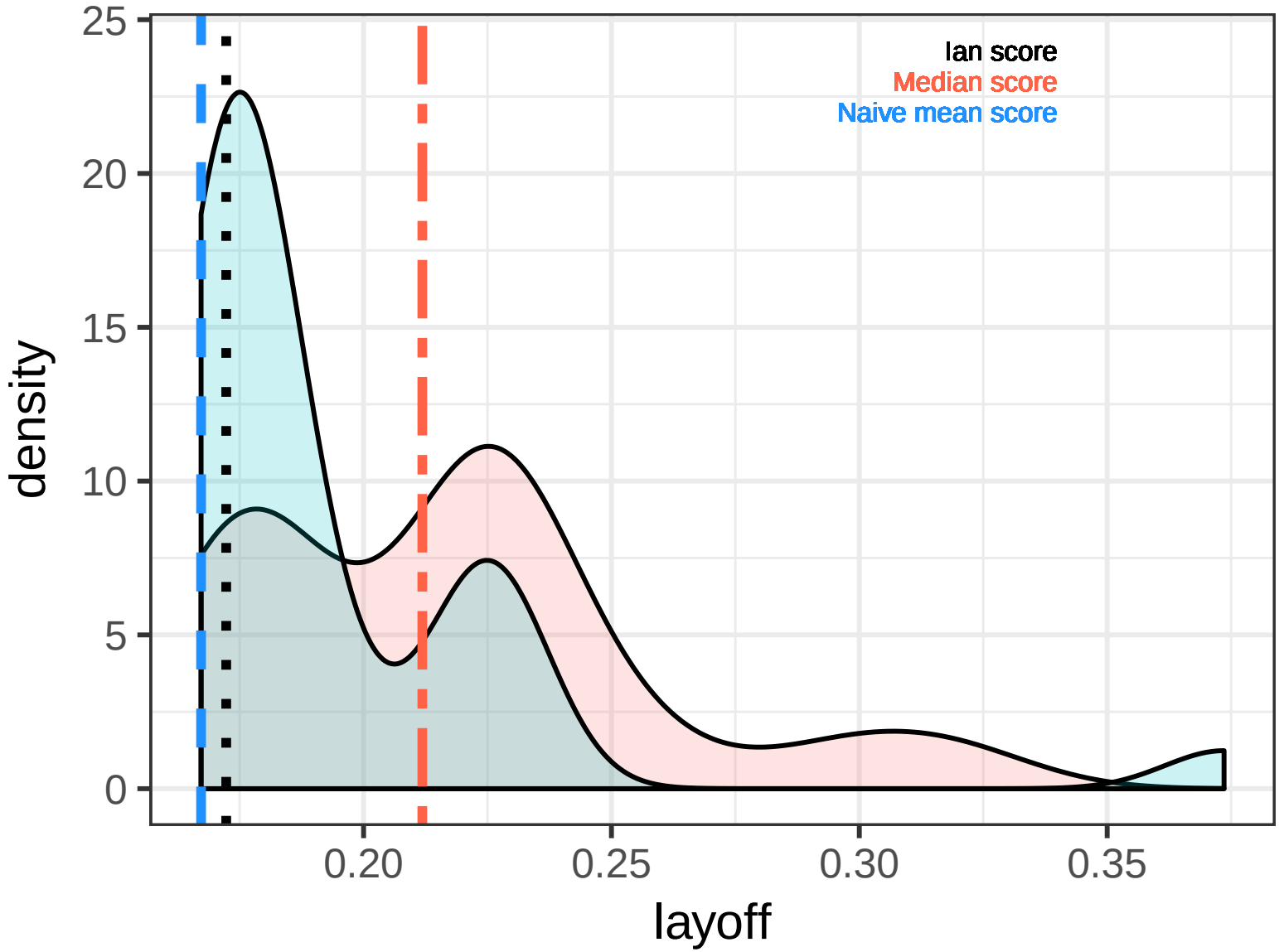
Teams that submit more have lower scores: Material hardship



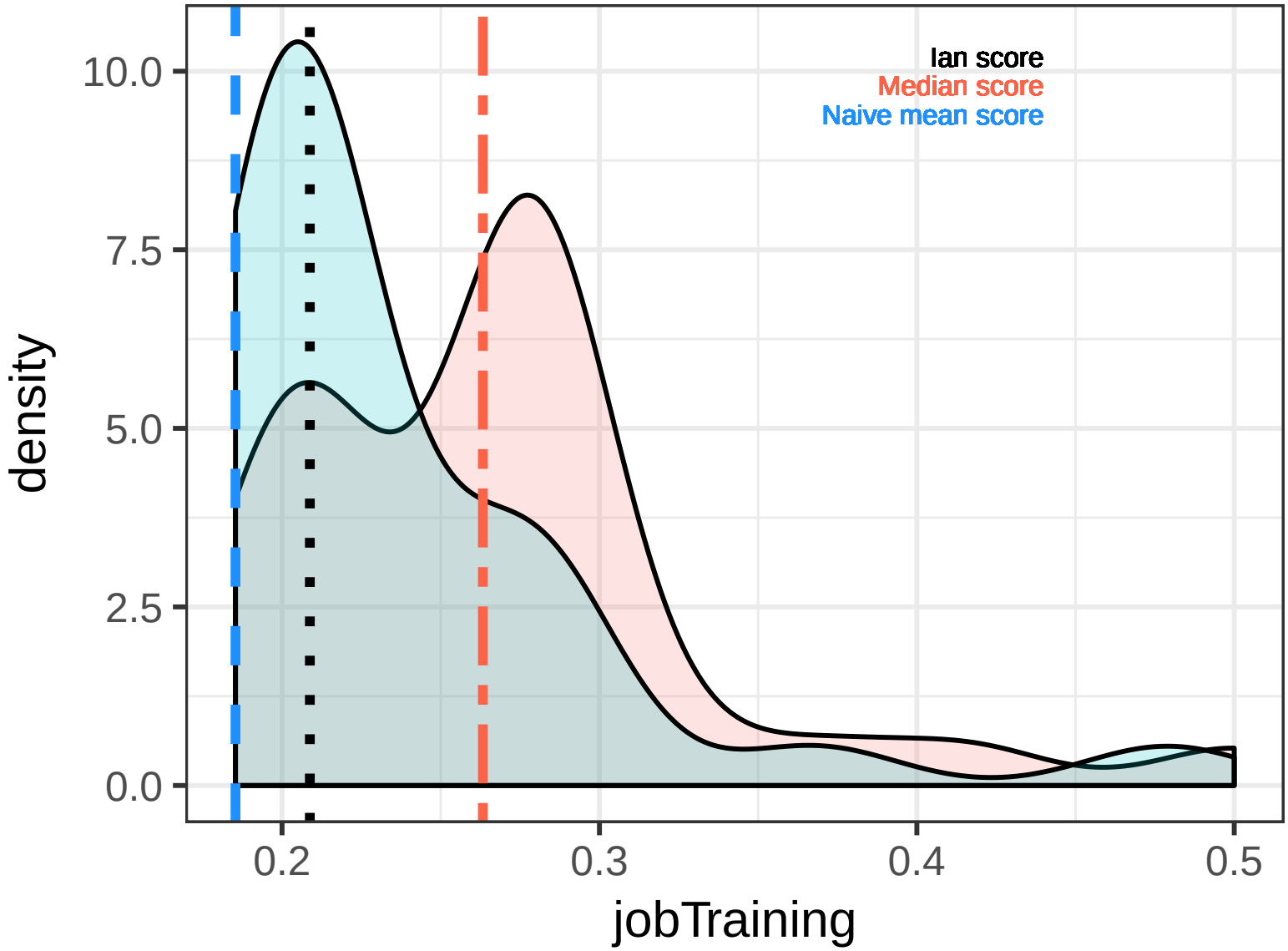
Teams that submit more have lower scores: Eviction



Teams that submit more have lower scores: Layoff



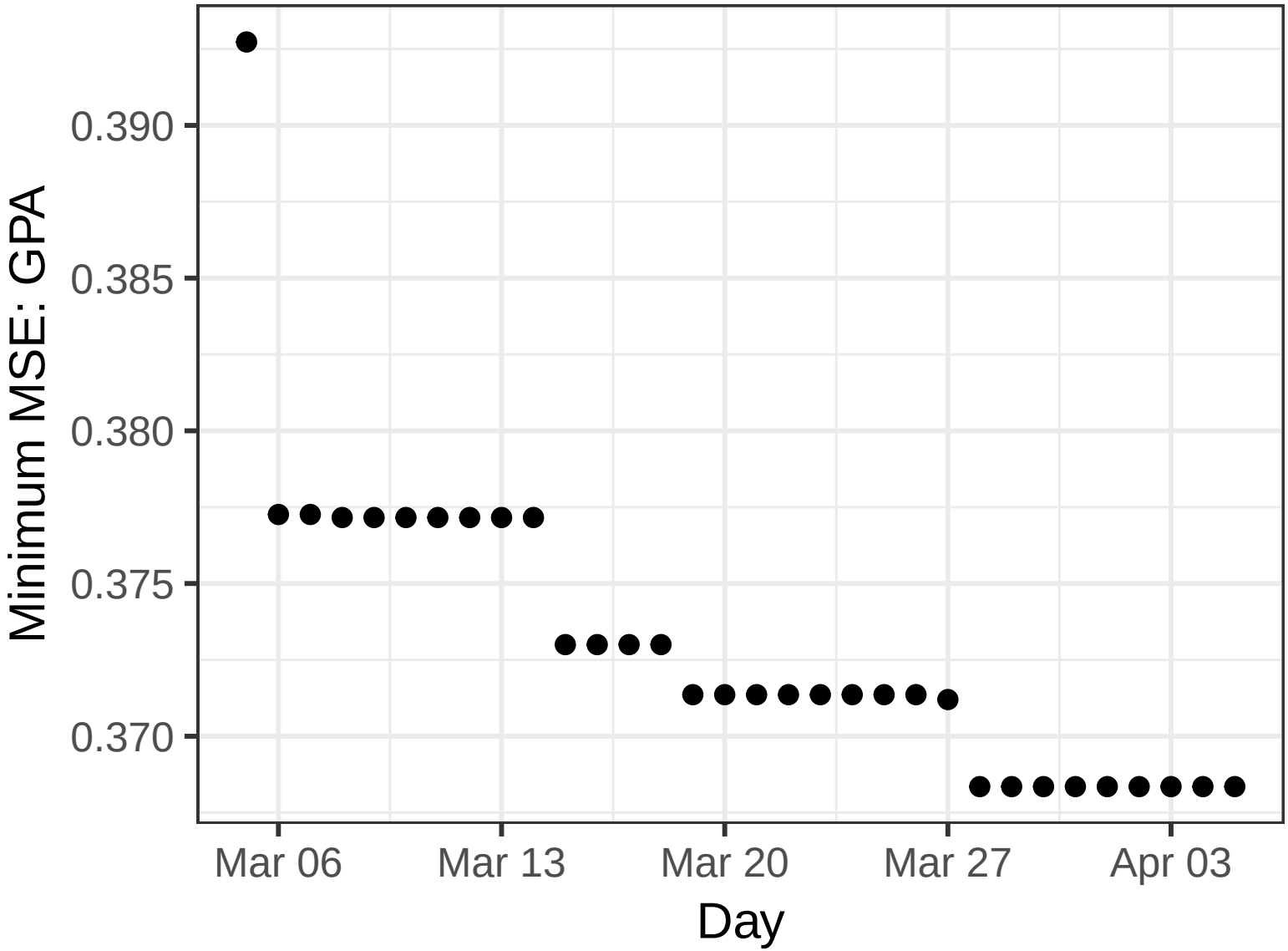
Teams that submit more have lower scores: Job training



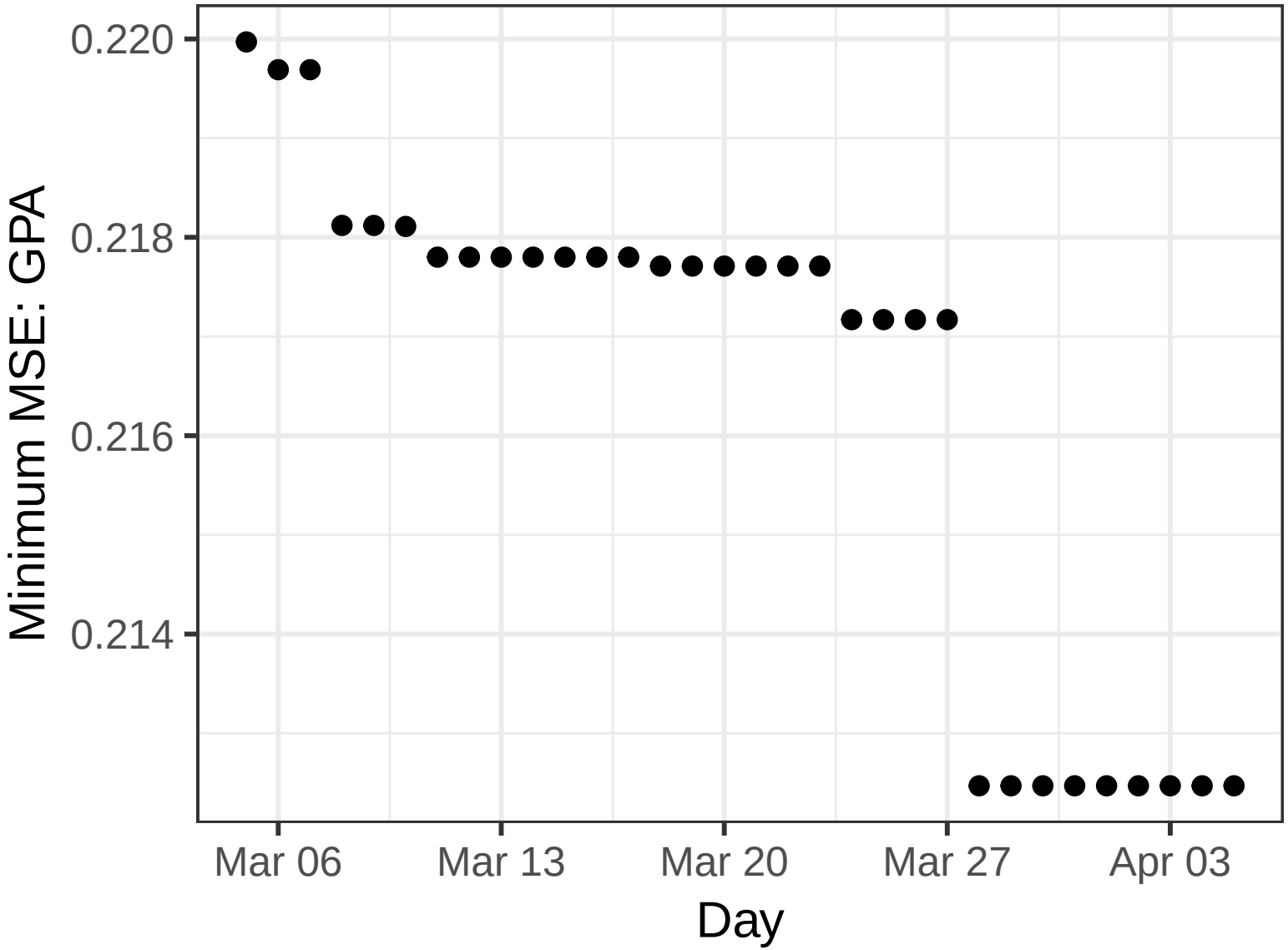
Community score trends by variable

Next, let's take a look at the score trend in the community over time. We'll plot the minimum MSE for each variable against the day of the challenge.

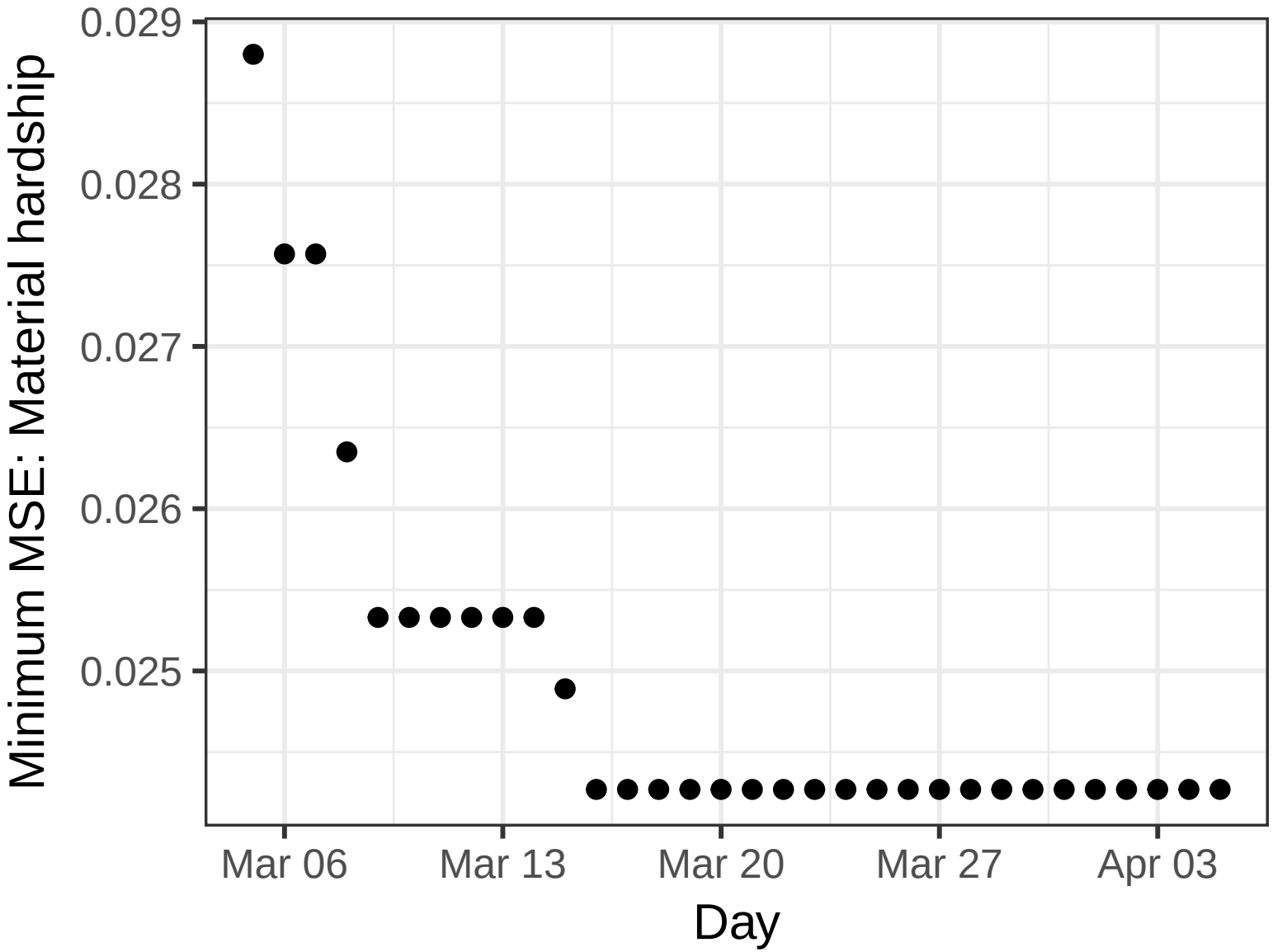
Community score trend: GPA



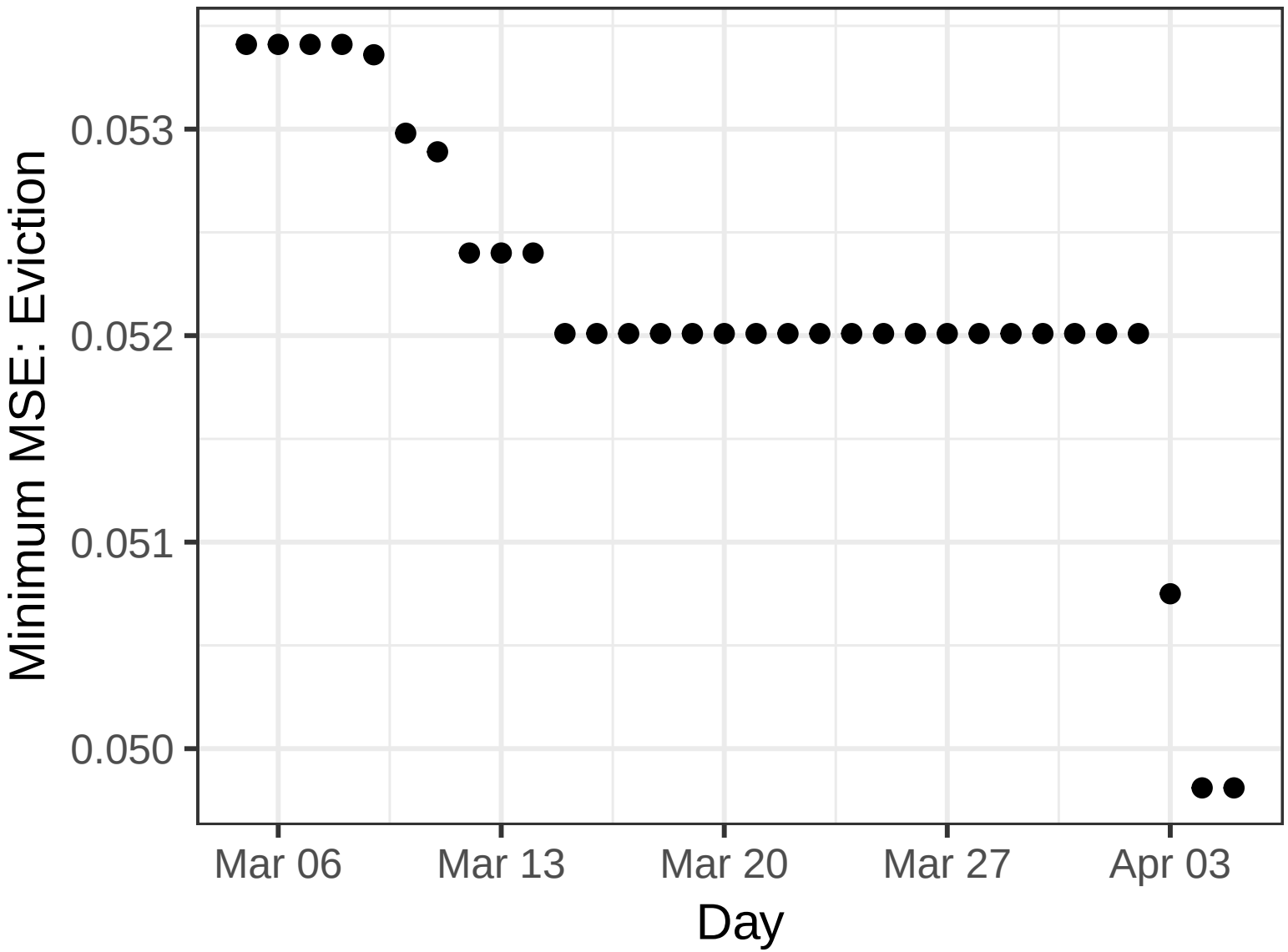
Community score trend: Grit



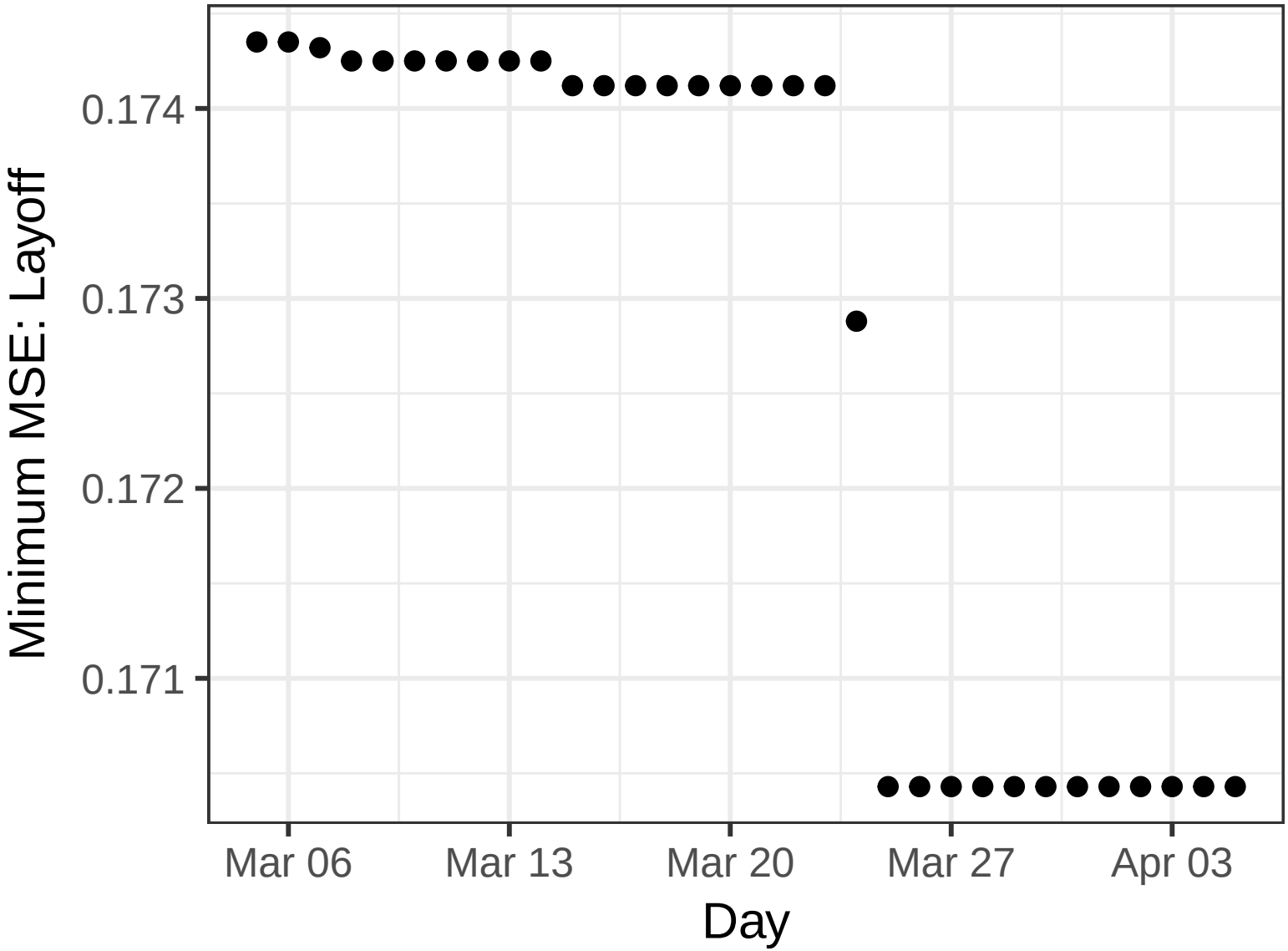
Community score trend: Material hardship



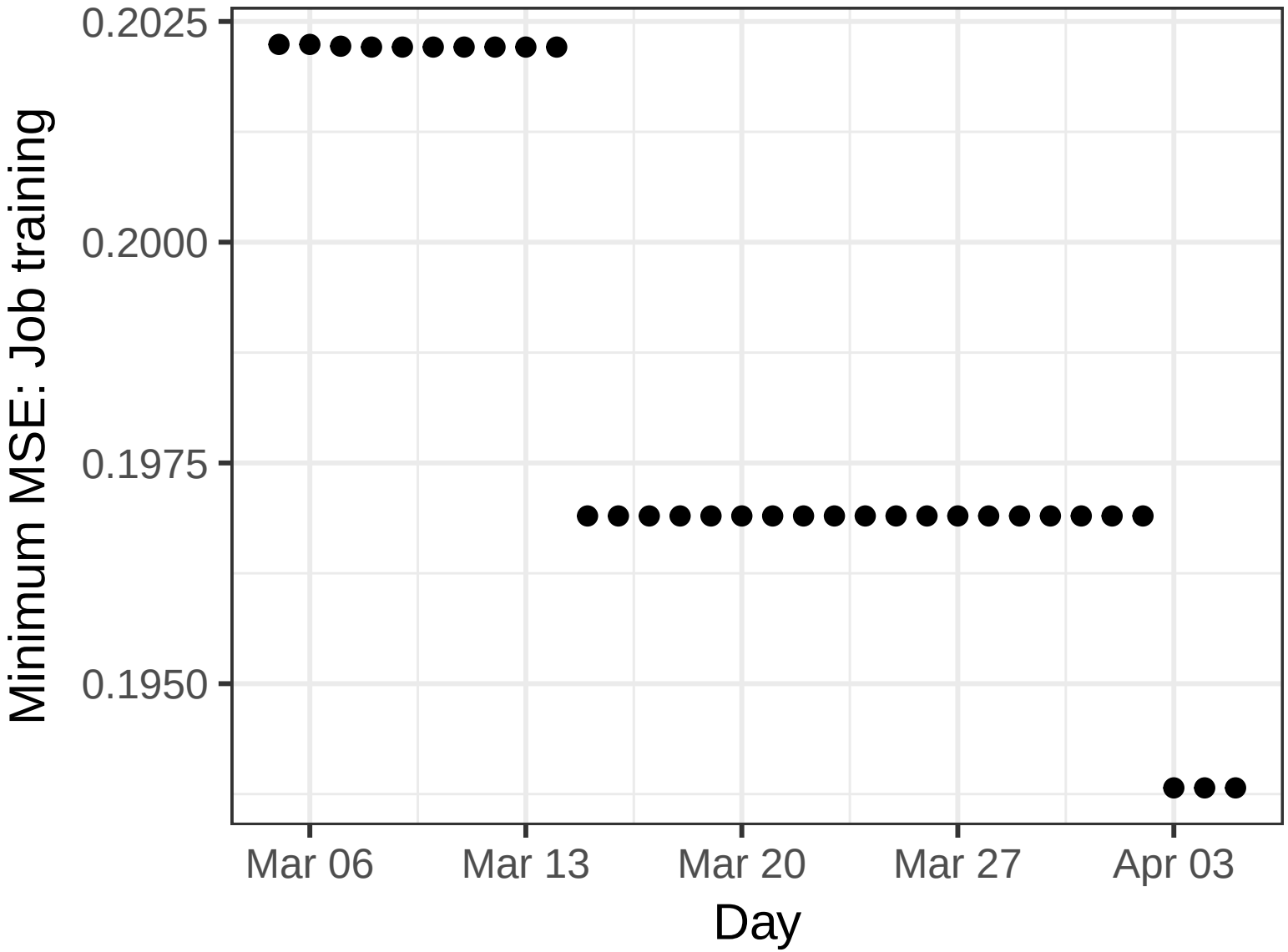
Community score trend: Eviction



Community score trend: Layoff



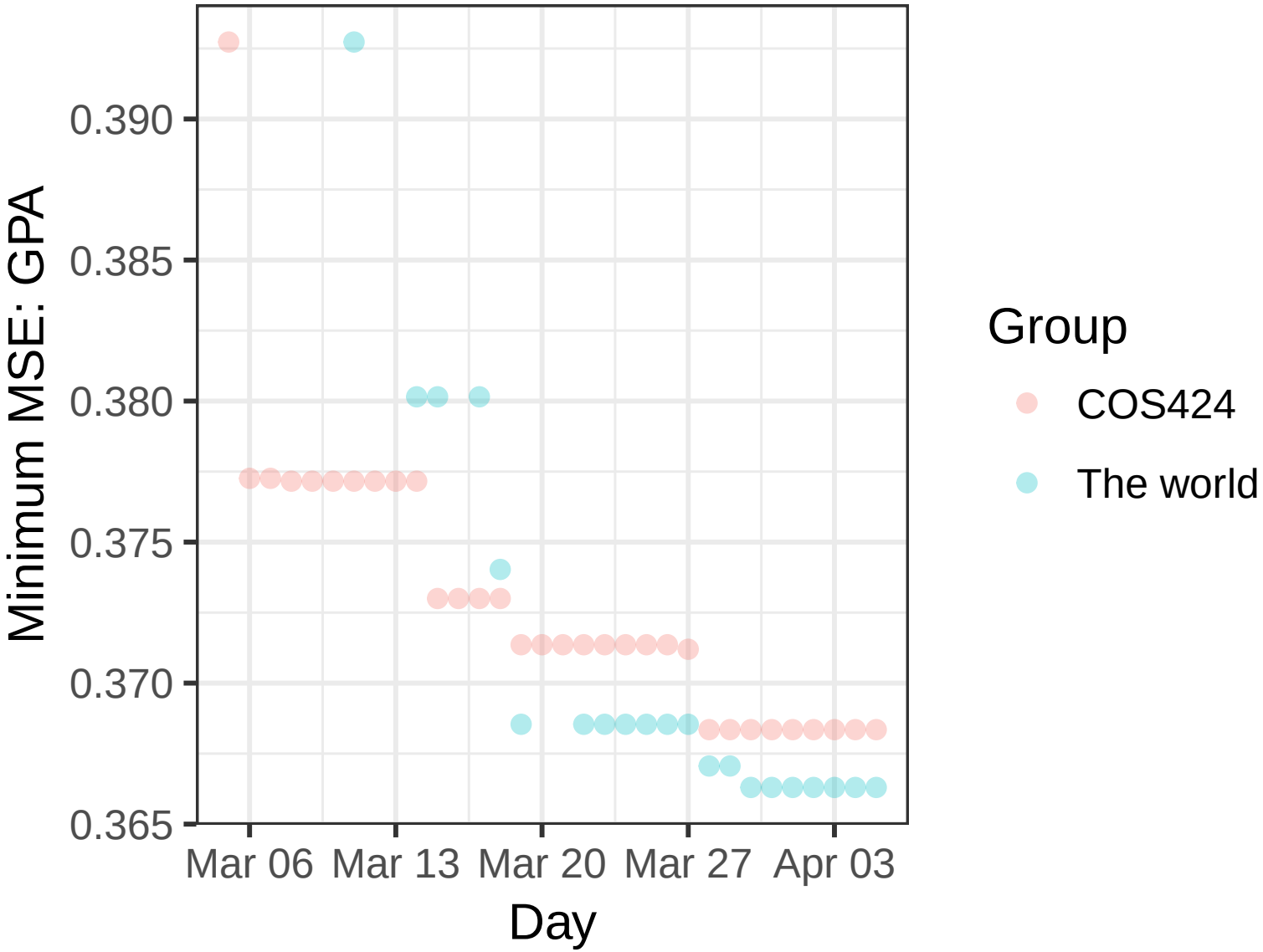
Community score trend: Job training



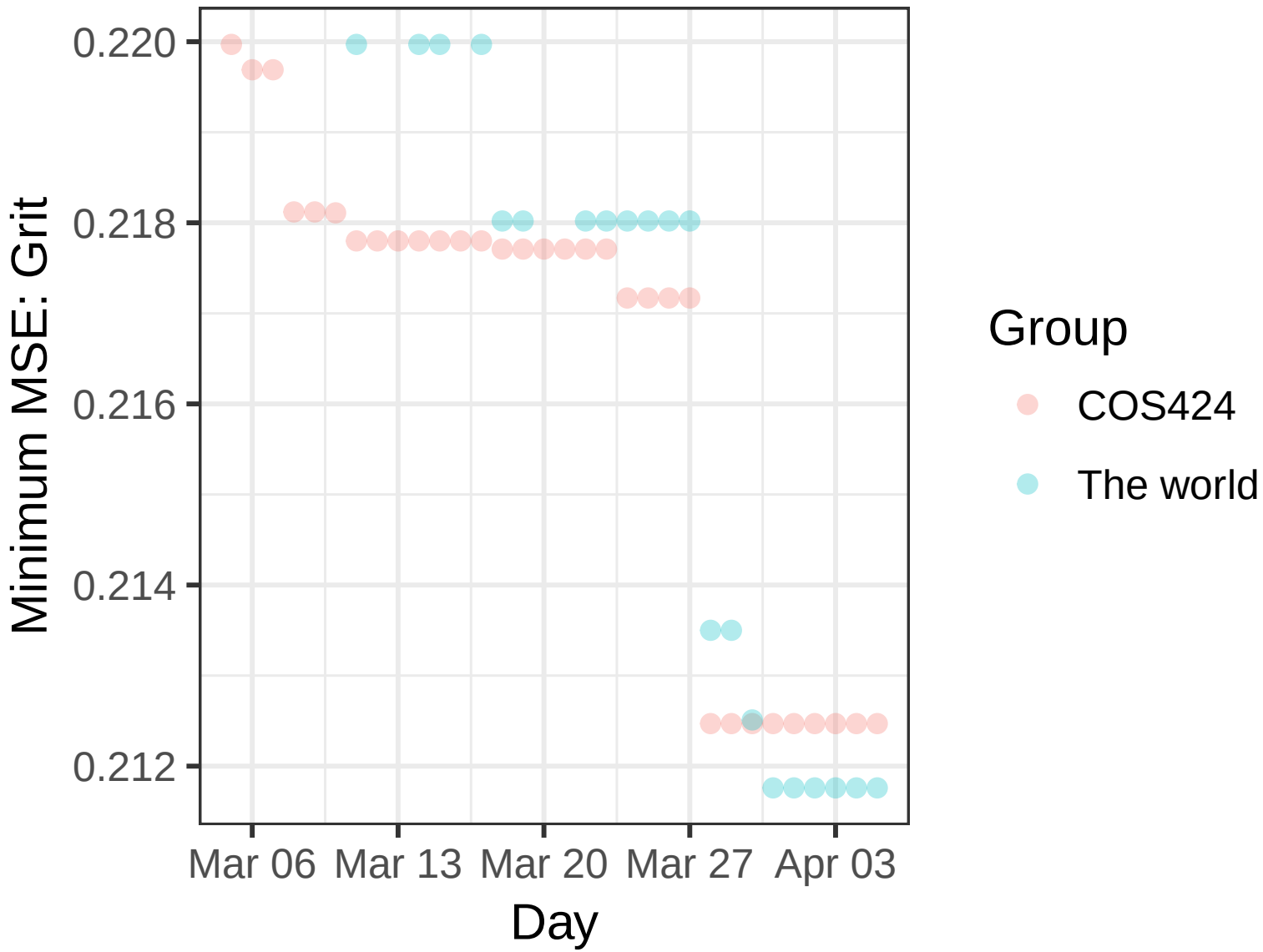
COS 424 vs. the world

How do COS424 participants compare to other participants in the challenge?

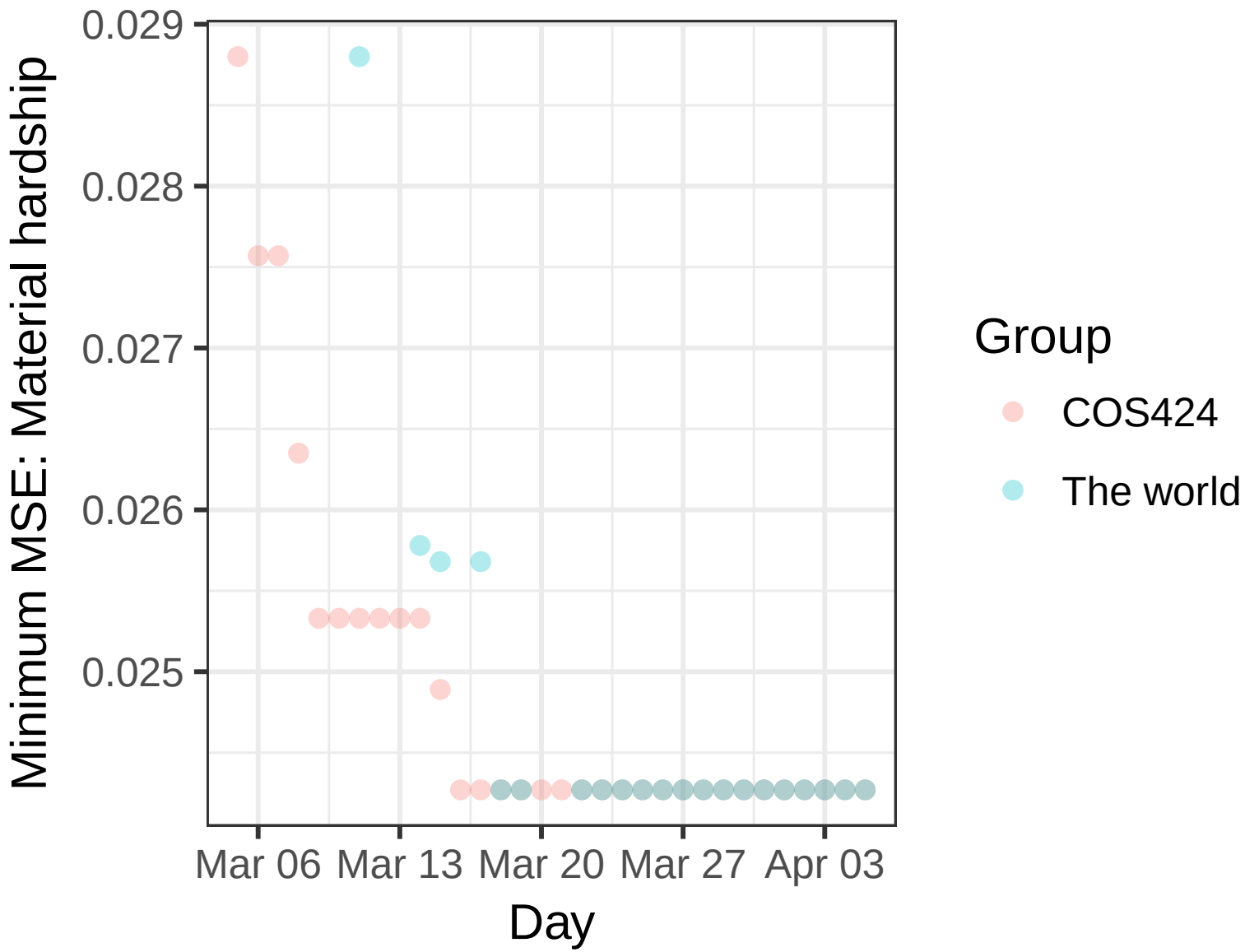
COS 424 vs. the world: GPA



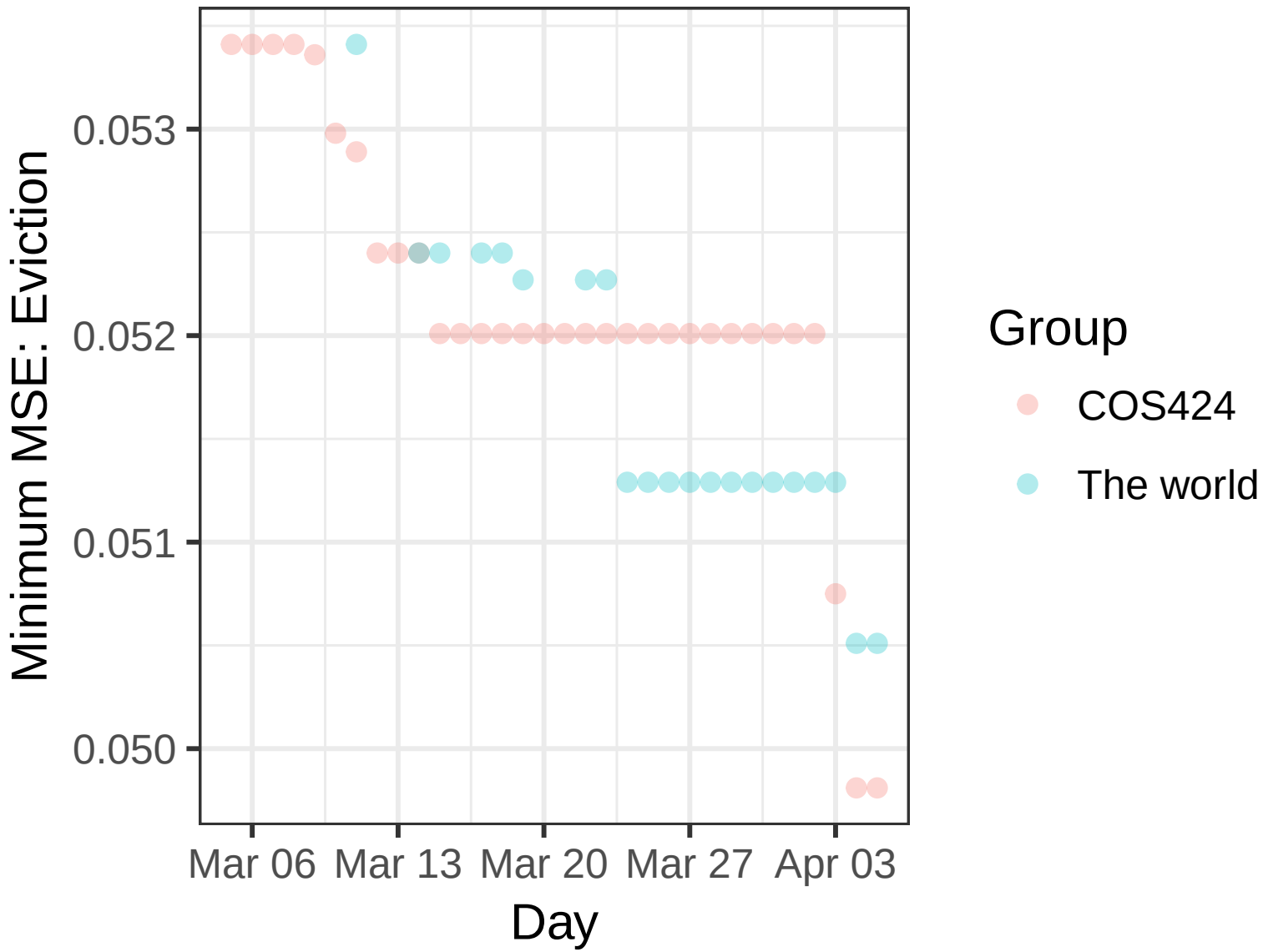
COS 424 vs. the world: Grit



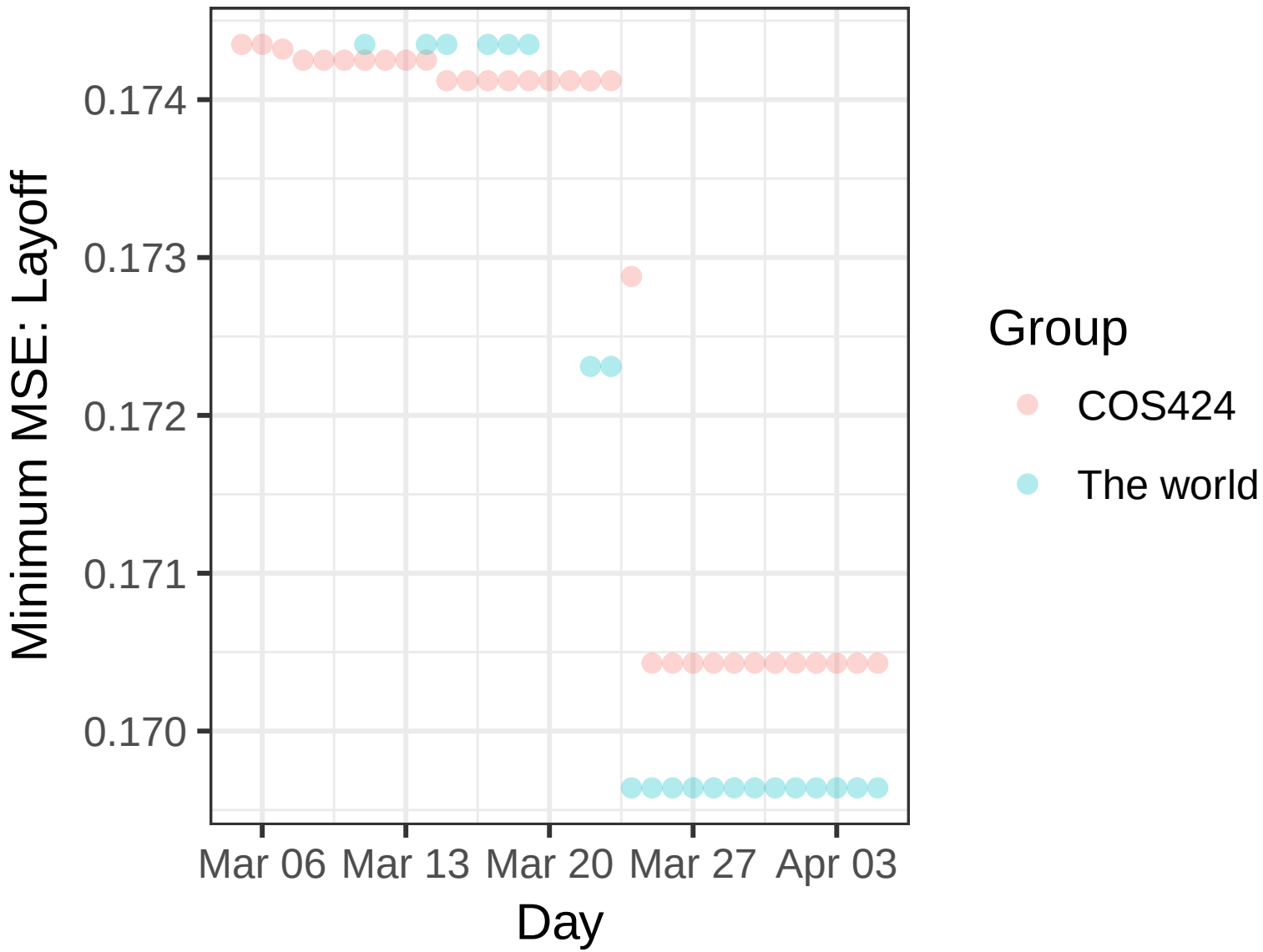
COS 424 vs. the world: Material hardship



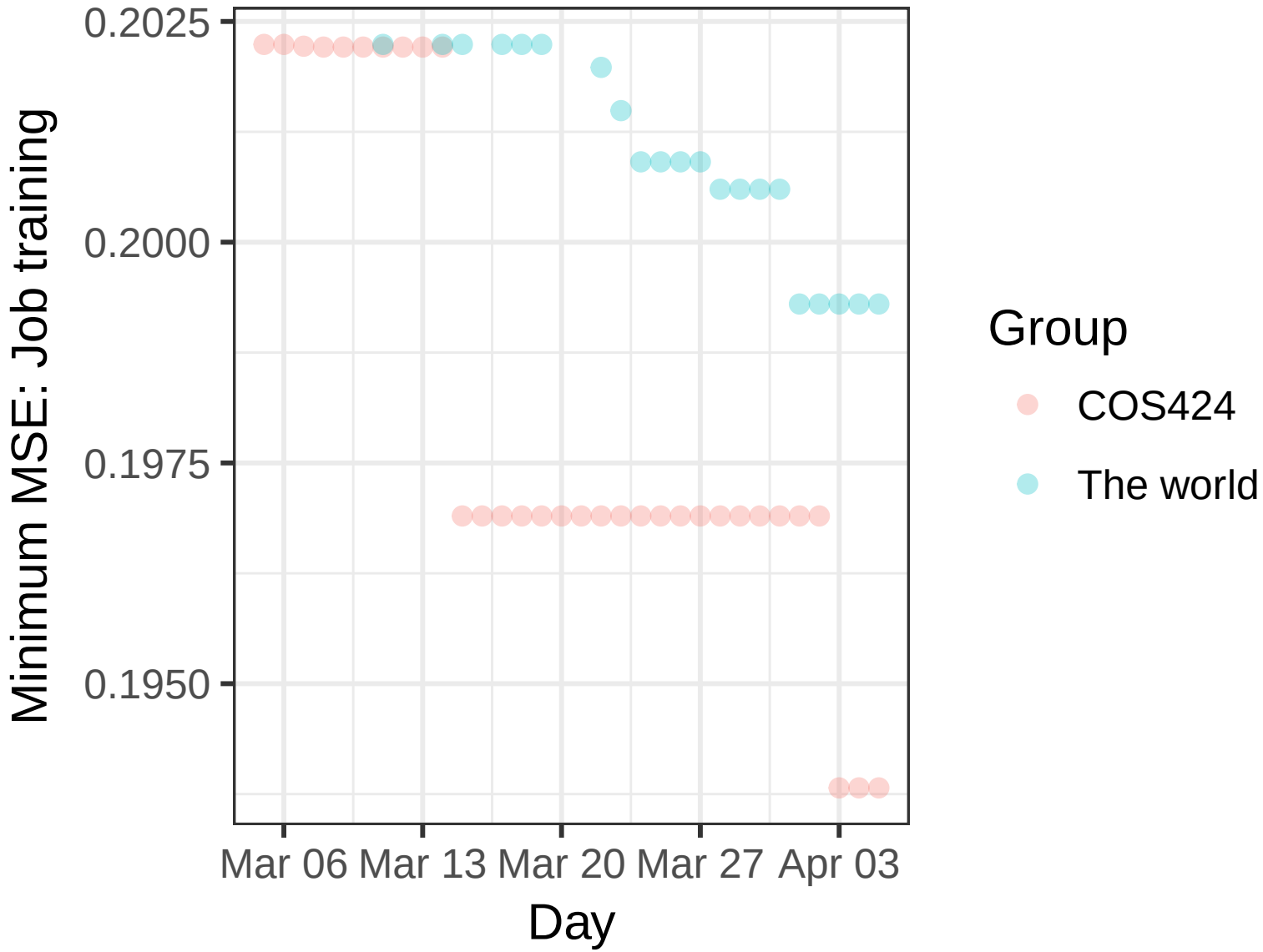
COS 424 vs. the world: Eviction



COS 424 vs. the world: Layoff



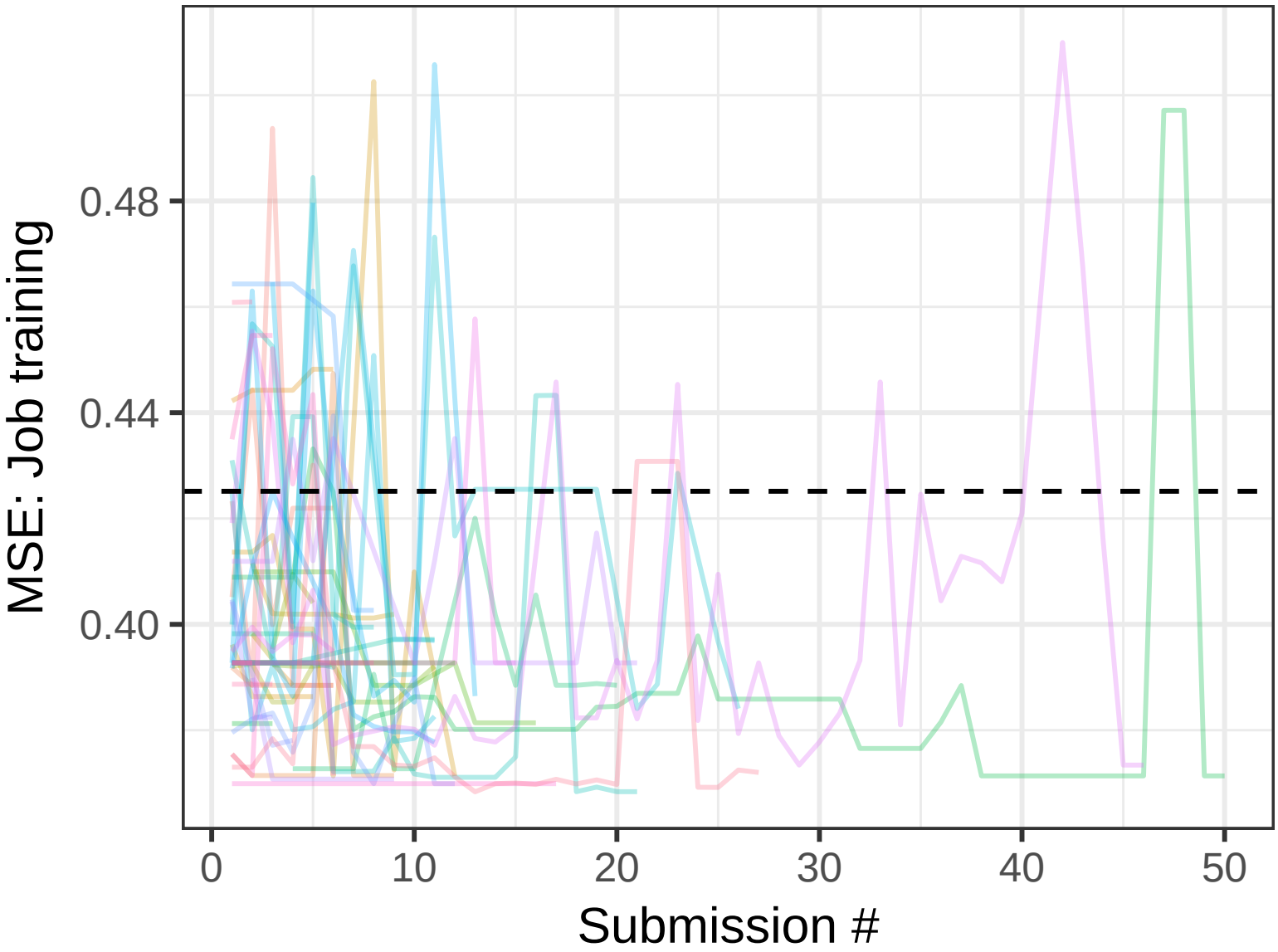
COS 424 vs. the world: Job training



Individual score trends are noisy

User scores move around quite a bit with successive attempts.

Individual score trends (e.g. GPA)



Languages and libraries

Primary language used

- Python users: **62**
  - Jupyter users: **10**
- R users: **6**
  - RMarkdown users: **2**
- Other: **1**
- Missing code or submission: **4**

Let's dig a little more into the Python models.

Top 5 libraries

- `numpy`: **61**
- `scikit-learn`: **55**
- `pandas`: **54**
- `matplotlib`: **33**
- `time`: **25**

Top 5 `scikit` modules

- `linear_model`: **55**
- `metrics`: **38**
- `model_selection`: **36**
- `preprocessing`: **30**
- `ensemble`: **28**

Missing data

- **46** groups used some form of imputation to address the missing data problem.
  - Mode imputation was most popular (**34** groups), followed by mean imputation (**14** groups) and median imputation (**8**).
  - Less frequently used techniques included PCA, KNN, and manual imputation.
  - Only **5** groups used multiple imputation.
- **6** groups dropped missing data entirely.
- **12** groups did not explicitly address missing data in their write-ups or code.

Variable selection

- Almost every group performed variable selection automatically.
- Few groups explicitly inspected individual variables.

Model selection and evaluation

- Most teams evaluated multiple models.
  - (This makes an automated code analysis much more difficult!)
- **32** teams imported a cross-validation tool as they built their models.
- Popular model evaluation measure imports (not mutually exclusive)
  - MSE: **21**
  - R2: **12**
  - Precision/recall: **6**
  - Predictive accuracy: **5**
  - AUC: **3**
- Model selection imports (not mutually exclusive)
  - GridSearchCV: **13**
  - K-fold CV: **7**
  - Stratified k-fold CV: **5**
  - LOOCV: **1**
- **12** groups appear to have constructed their own holdout test sets

Regularization

- Approximately 1/2 of groups used some form of regularization import to make their predictions.
- Popular regularized model imports (not mutually exclusive)
  - Lasso: **26**
  - Ridge regression: **11**
  - ElasticNet: **11**

Modeling strategies

Which tools and techniques were frequently used together?

Answer: Let's look at the covariance of some commonly-used tools.

From tools to scores

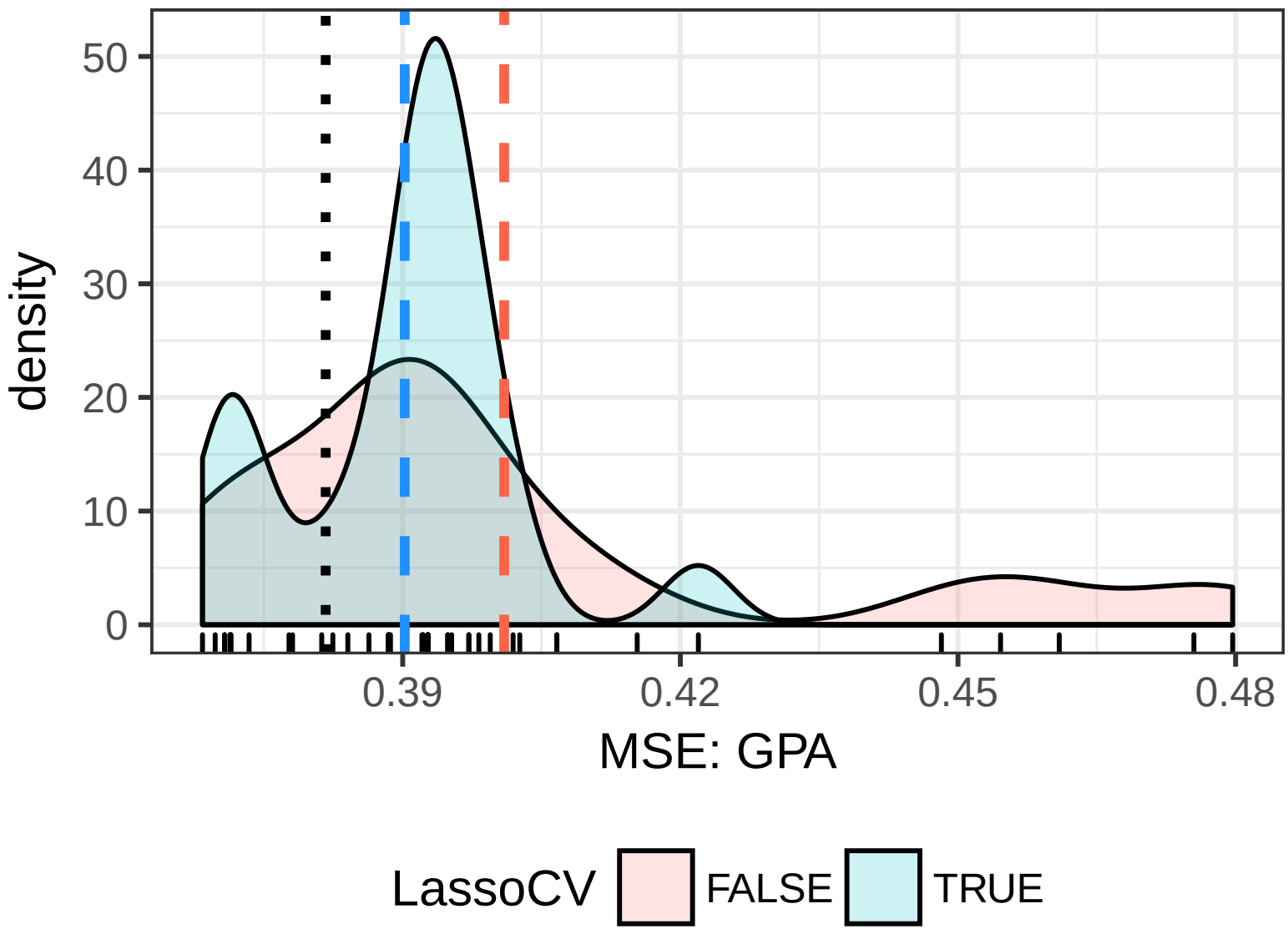
Tools and scores

Are some tools associated with better predictive scores?

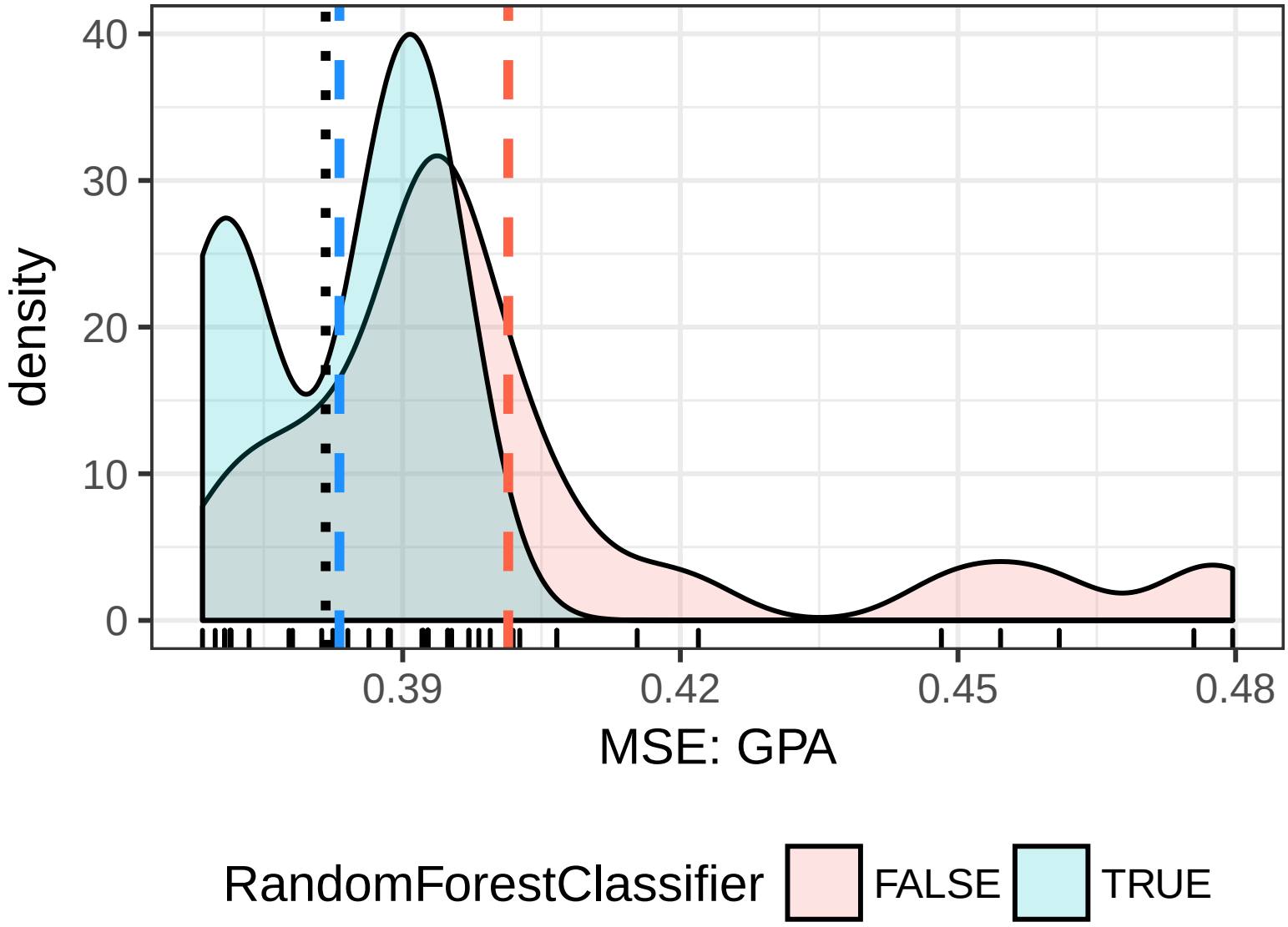
Answer: Let's decompose the score distributions according to tool usage.

Note: We're using the **blue line** to indicate the mean among users for a given tool, and the **red line** to indicate the mean among non-users. (The **dotted line** is still Ian's benchmark.)

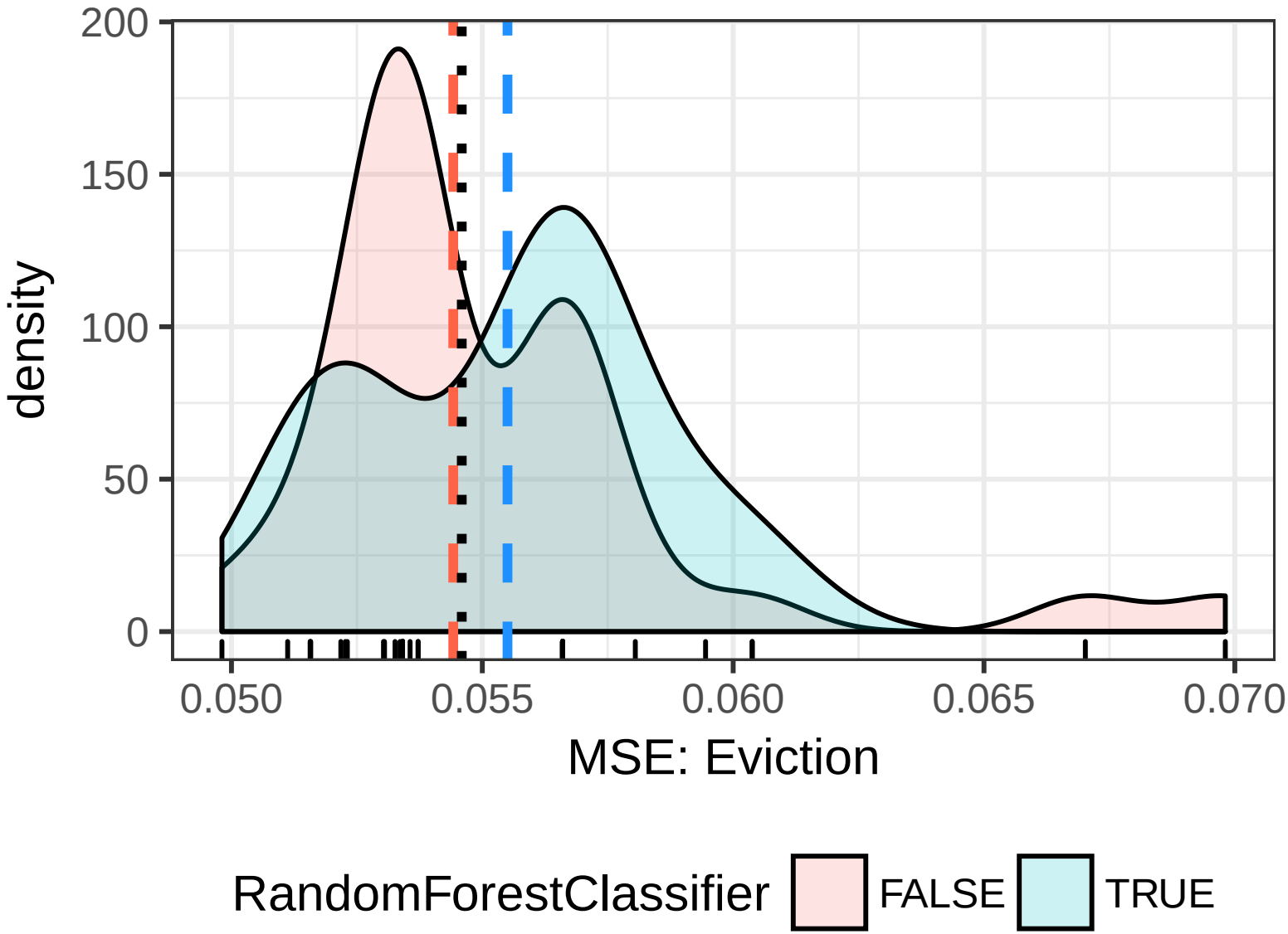
LassoCV users perform somewhat better on GPA



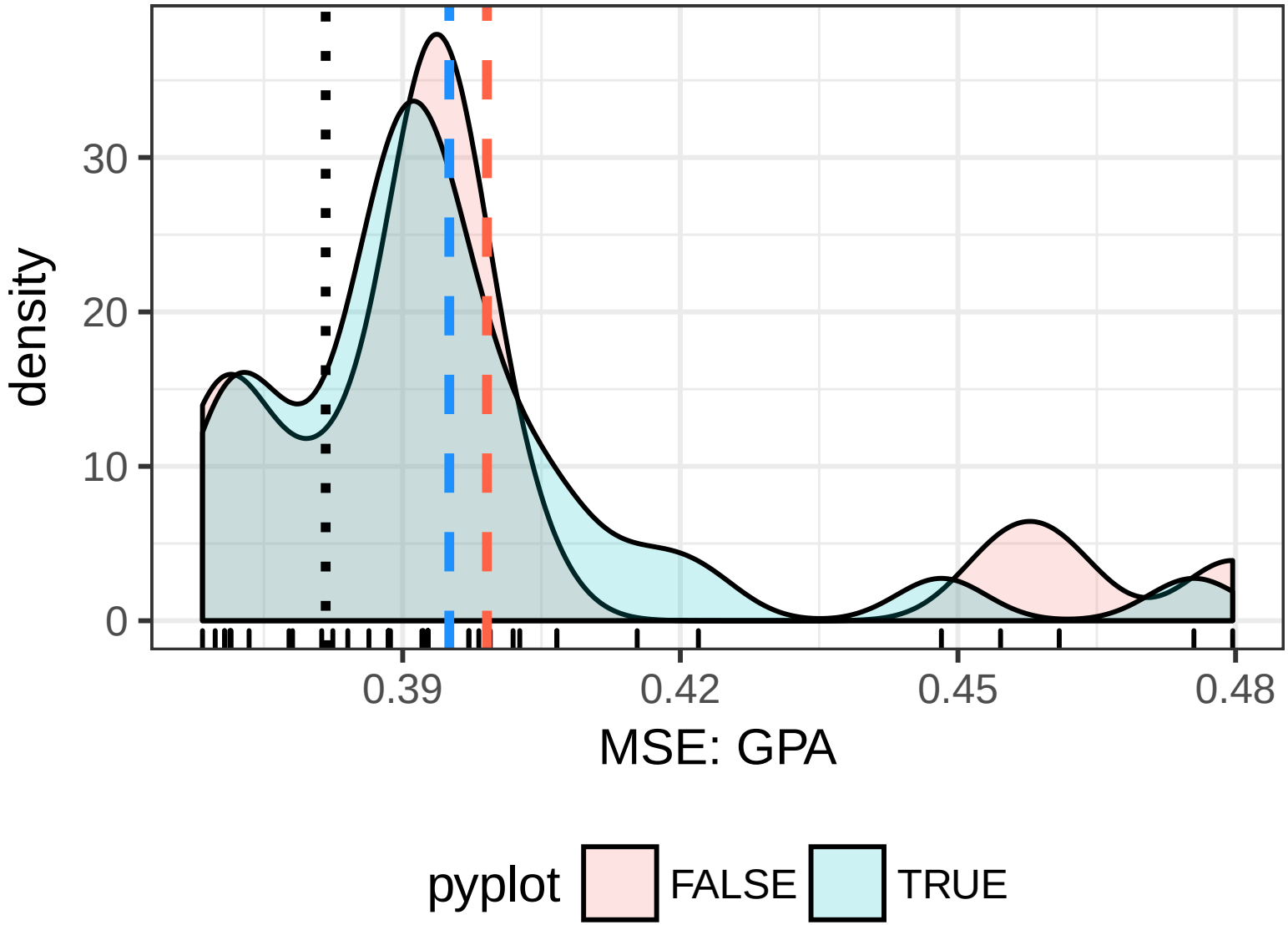
RandomForestClassifiers do *significantly* better on GPA (and grit)...



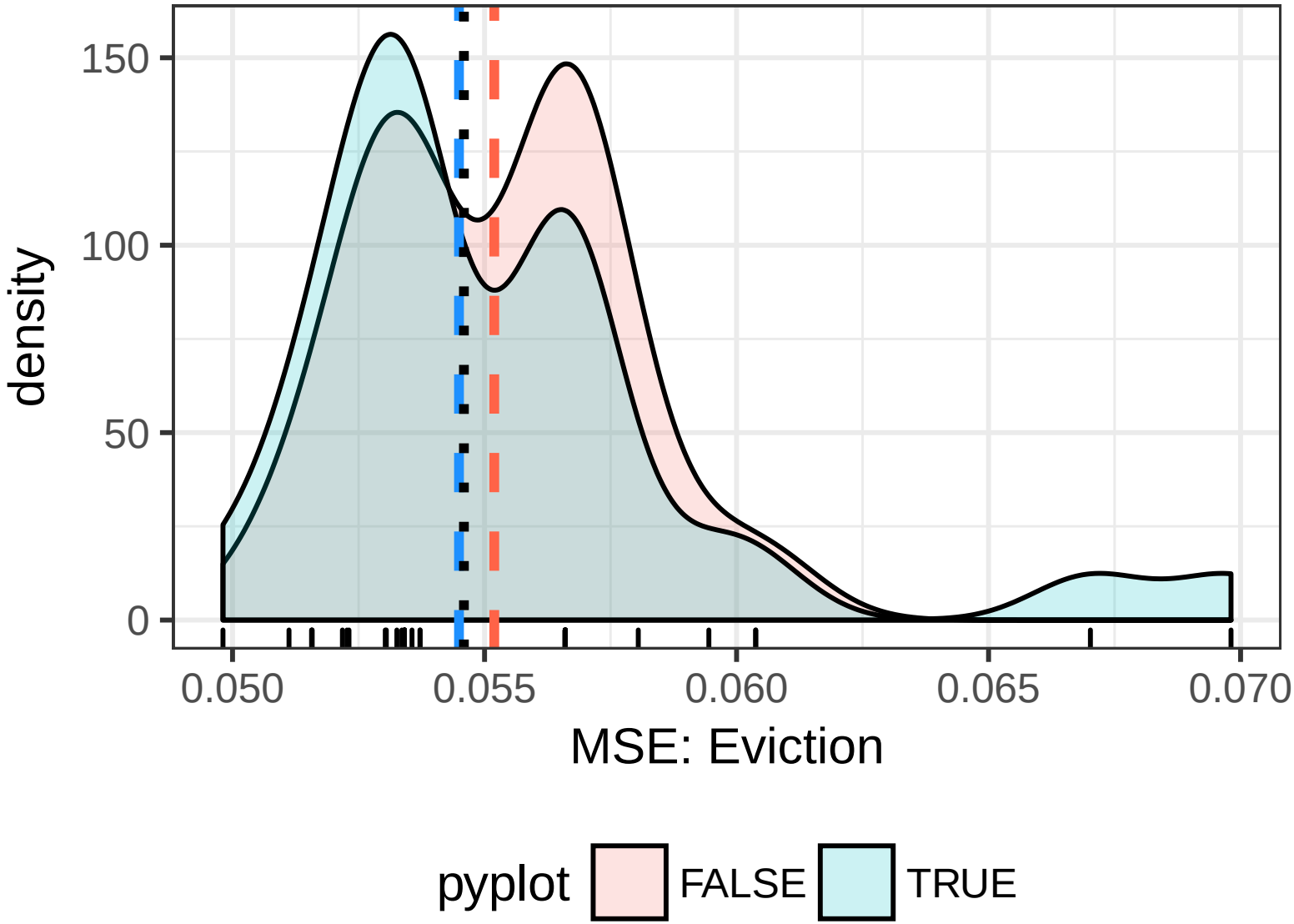
. . . but fare no better on discrete variables.



Data visualizers perform about as well as non-visualizers (GPA)



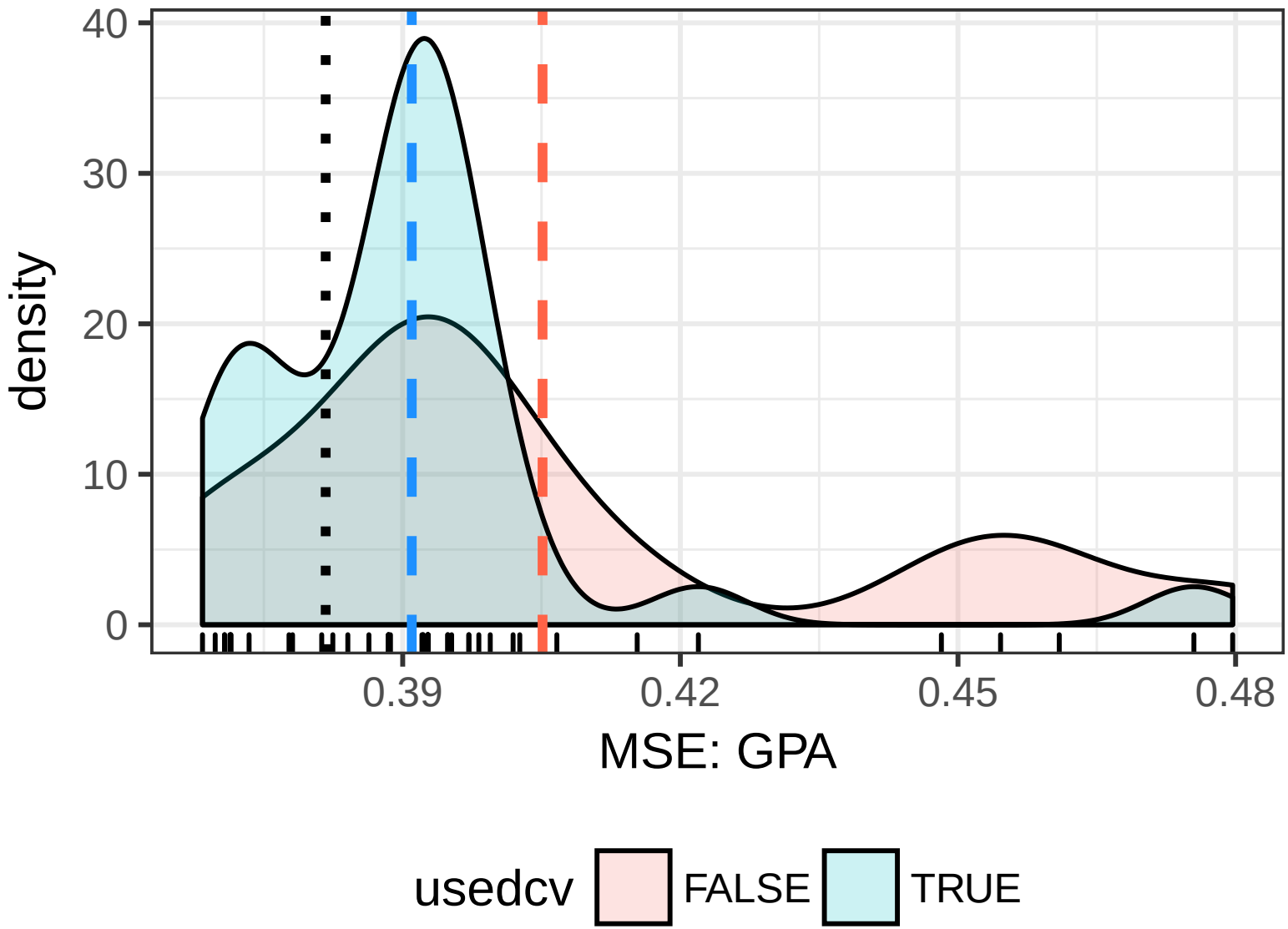
Data visualizers perform about as well as non-visualizers (eviction)



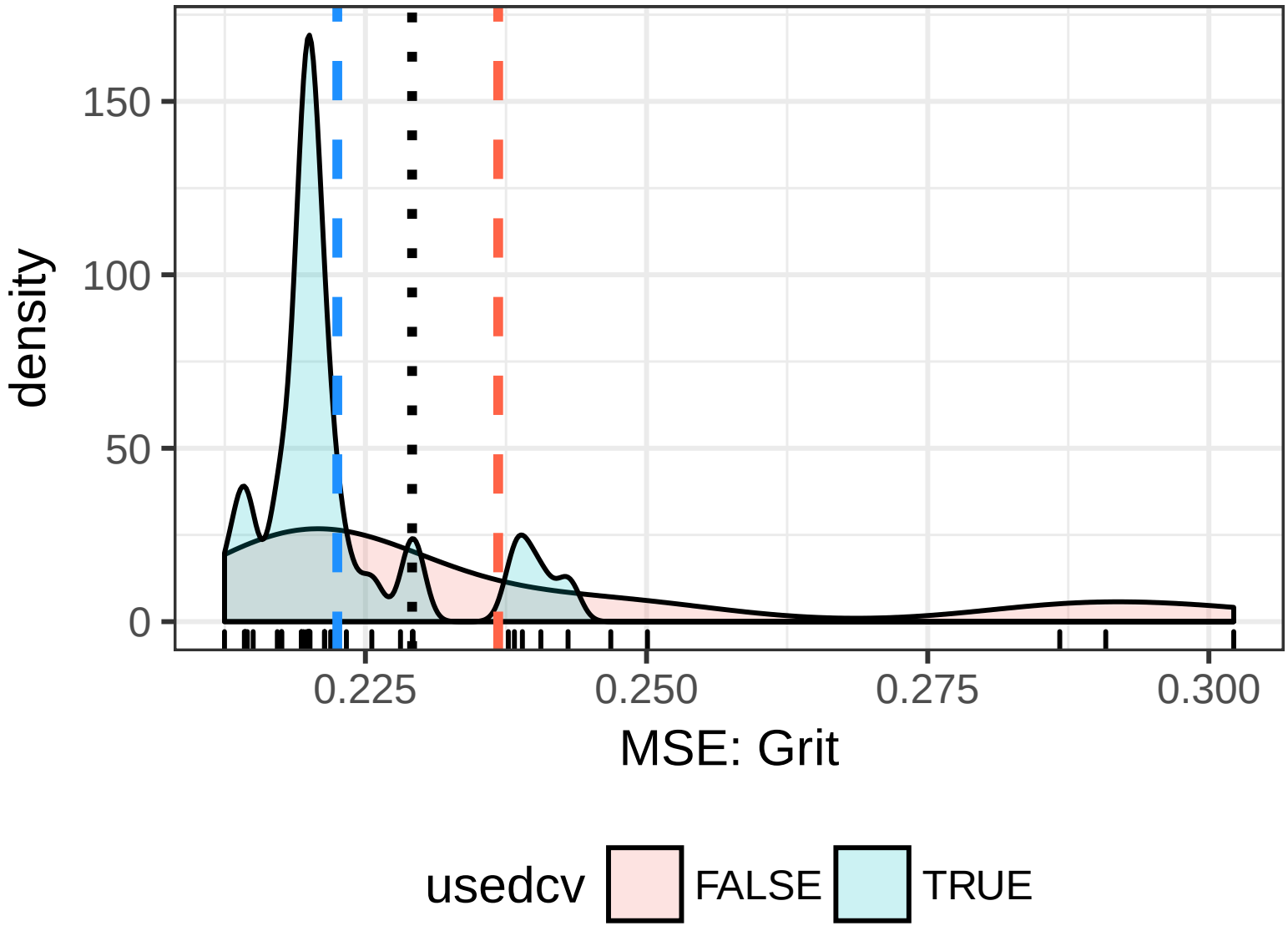
Does cross validation pay off?

We can also split the submissions into groups that used cross validation, and those that didn't.

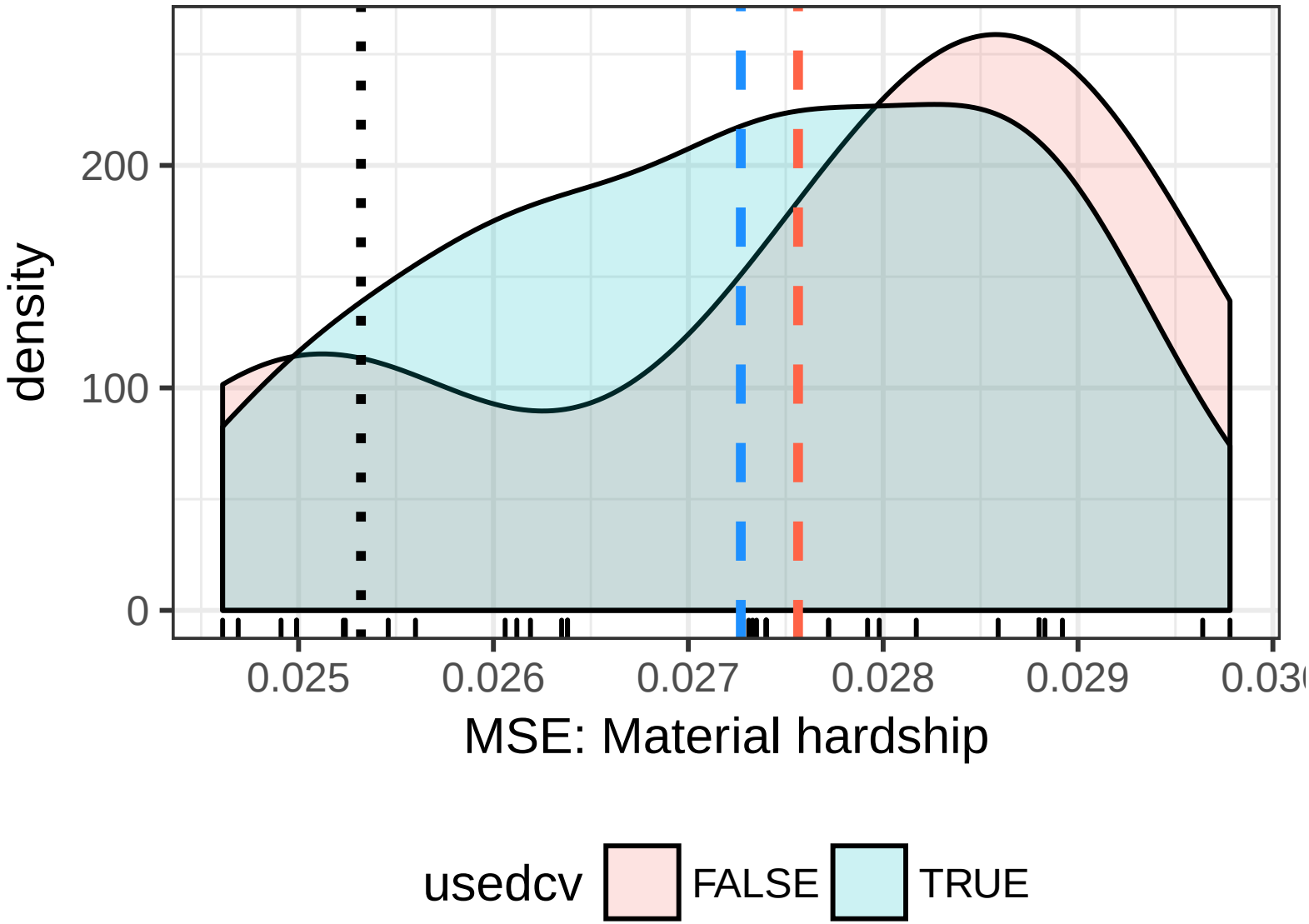
Cross validated model scores: GPA



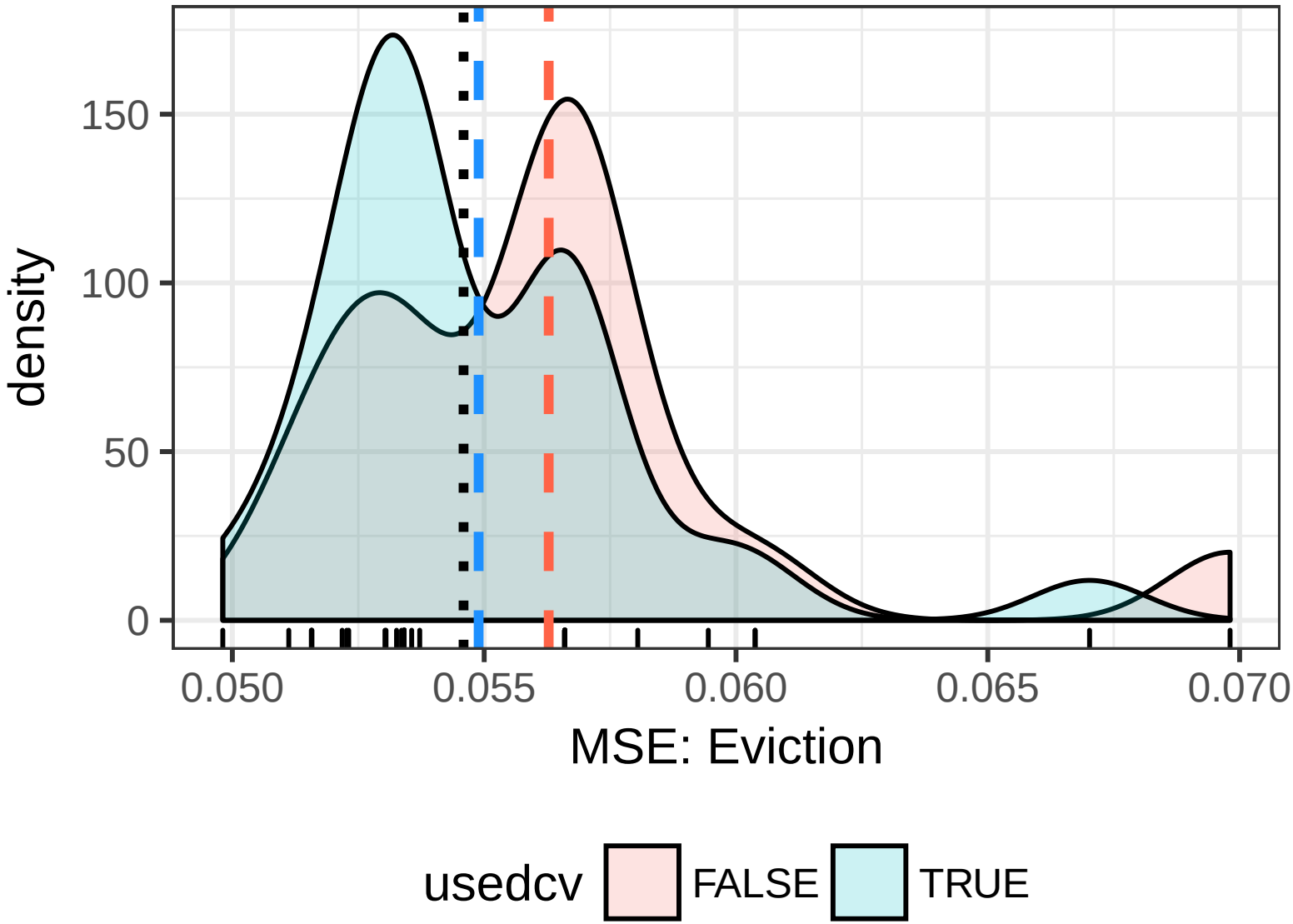
Cross validated model scores: Grit



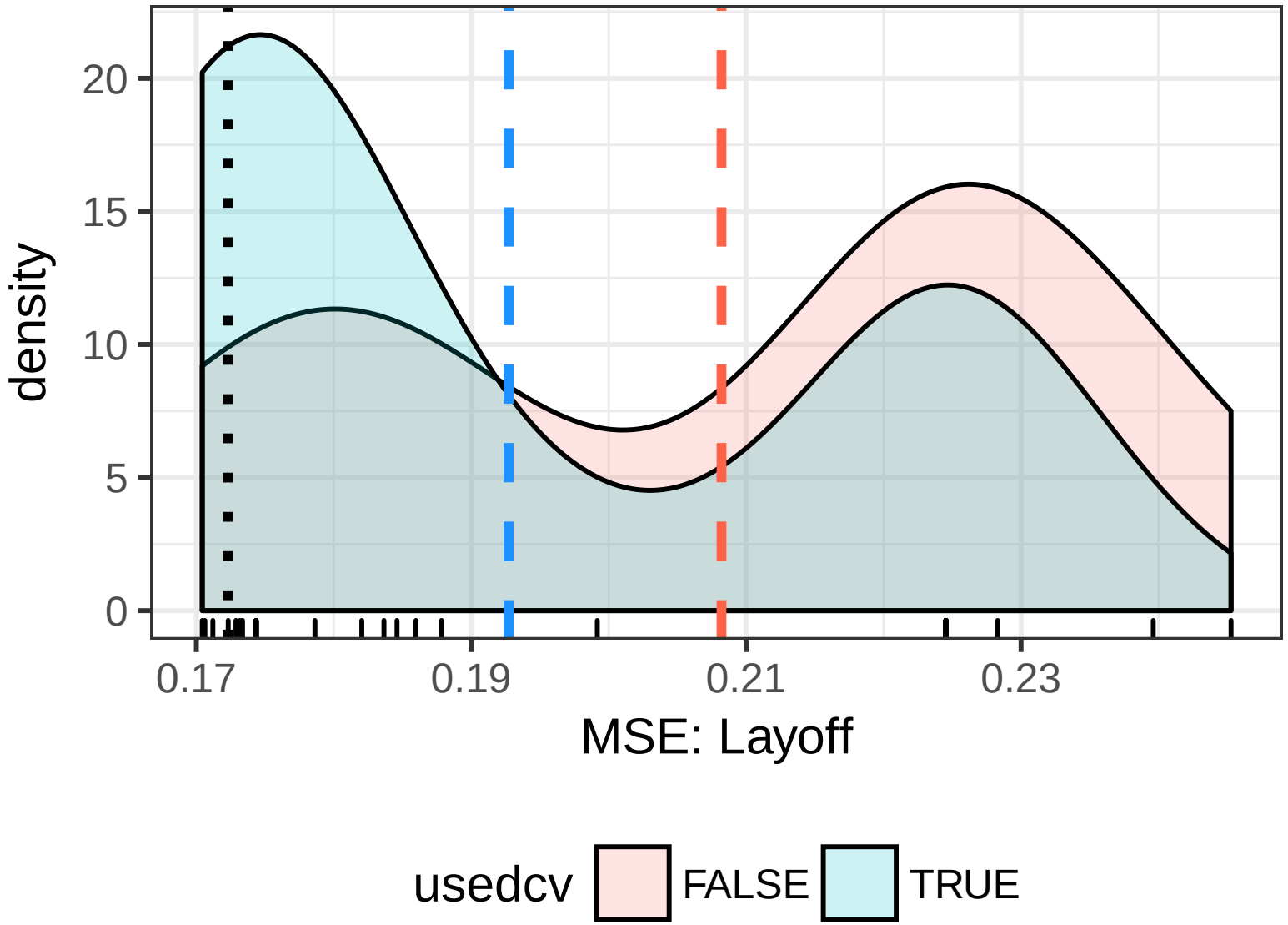
Cross validated model scores: Material hardship



Cross validated model scores: Eviction



Cross validated model scores: Layoff



Cross validated model scores: Job training

Next steps

Code analysis

- Include additional languages
  - Particularly R and Stata
- Revise import analyses
  - Function call data
  - Pattern matching (for non-library code)
    - ★ Results are sensitive where homebrew calculations are more likely (e.g. MSE)
- Include information on order of function calls
- Examine evolution of strategies over time
- Less supervised approach?

Future questions for the Fragile Families study

Contributing to FFC

- Work and independent study opportunities for summer and fall
  - Building interactive tools for monitoring predictive modeling competitions
  - Improving the CodaLab app
  - Creating an ur-model from submissions
  - Modernizing the FF codebook
- Keep competing!

Adapting solutions to a final project