Lecture Five:

“An Introduction to the FFC

Dr. Charles Rahal
Department of Sociology and Nuffield College

University of Oxford

2020/2021

LEVERHULME
CENTRE
FOR
DEMOGRAPHIC
SCIENCE

How was the ‘Common Task Method’ used?

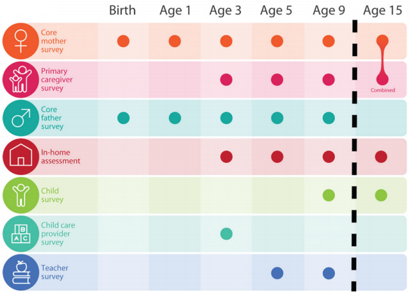
- An organizer designs a prediction task and recruits researchers.
- Of critical importance: The exact same outcomes using the exact same data.
- Predictions are evaluated with the exact same error metric on held-out data.
  - The held-out data is held by the organizer and not available to participants.
  - The participants are free to use any technique to generate predictions.
- The standardization of the prediction task facilitates comparisons between different approaches.
- It also removes concerns about over-fitting, researcher degrees of freedom.

Introducing the data-set: FFCWS

- The Fragile Families and Child Wellbeing Study: multistage, stratified random sample.
- Used in more than 750 published journal articles.
- Rich **longitudinal** data about thousands of families, 3:1 oversample of births to non-married parents.
- Each gave birth to a child in a large (200k+) US city (20, inc. 2 piltos) around the year 2000.
- Motivation: to understand families formed by unmarried parents.
  - And importantly: the lives of children born into these families.
- Six waves: child birth and ages 1, 3, 5, 9, and 15.
- Each wave includes a number of different data collection modules

Introducing the data-set: FFCWS

- Each data collection module is made up of $\sim$ 10 sections.



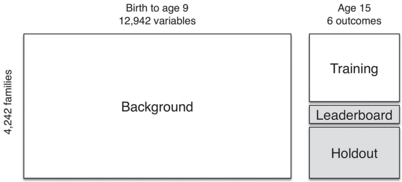
	Birth	Age 1	Age 3	Age 5	Age 9	Age 15
Core mother survey	●	●	●	●	●	●
Primary caregiver survey			●	●	●	● Combined
Core father survey	●	●	●	●	●	
In-home assessment			●	●	●	●
Child survey					●	●
Child care provider survey			●			
Teacher survey				●	●	

- Different family members/associates are interviewed at different times.
- In-person tests conducted in waves 3-5 (i.e. Peabody vocab test), as well as measurements, home
- Over time, the scope of data collection increased observation.

- The 'common task' only possible because contact made after Wave 5.
- Withholding data from public allows it to be truly held-out.

Test, training and leaderboard sets

- The 6th wave was split into three parts; training, test, and leaderboard.
- We're going to talk a lot more about this later today.



- The training data is used for training models.
- The leaderboard set is used to see how you're doing.
- The holdout set is/was really unseen.

- Background data included 4,242 families, 12,942 variables.
- Half the data was reserved for training.

Evaluating the models

- Models were evaluated with the R^2_{Holdout} metric:

$$R^2 = 1 - \frac{\sum_{i \in \text{Holdout}} (y_i - \hat{y}_i)^2}{\sum_{i \in \text{Holdout}} (y_i - \bar{y}_{\text{Training}})^2} \qquad (1)$$

- An $R^2_{\text{Holdout}} = 0$ is no more accurate than predicting the mean.
- No lower bound!
- An $R^2_{\text{Holdout}} = 1$ predicts **perfectly**.
- Note that all participants were compared against a benchmark:
  - Three mother variables (race/ethnicity, marital status, and education)

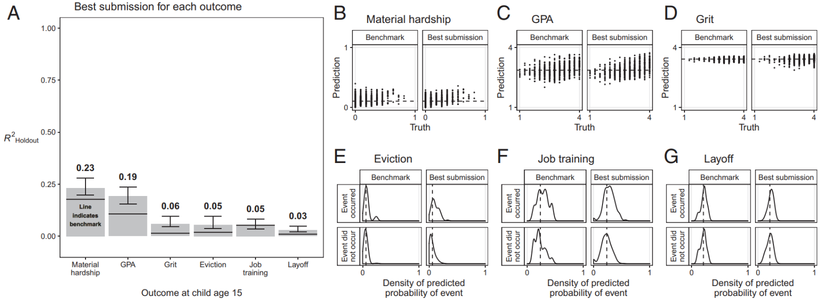
and

  - And a Wave 5 measure of the outcome/proxy ('$t-1$').
- So what did they predict, and how did they do?

What exactly did they predict?

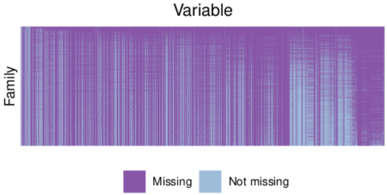
1. Child grade point average (continuous):
   - Reverse coded, A-D, for grades in English, Math, History and Science.
2. Four questions, four responses, Duckworth et al. (2007) style (continuous):
   - Reverse-coded (A-D), four subsections, and averaged.
3. Household eviction (binary):
   - Regardless of whether court ordered informal landlord. **Note: sparse!**
4. Household material hardship (continuous):
   - Average of 11 binary responses.
5. Primary caregiver layoff (binary, fairly exogenous):
   - Have you been laid off at any time since approx. child age 9?
6. Primary caregiver training (binary, very hard to predict):
   - Have you taken any classes to improve skills since approx. child age 9?

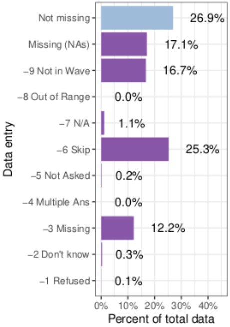
And how did the teams do?



- A few other observations in addition to the results:
    1. Teams used a variety of preprocessing and statistical techniques.
    2. Despite this diversity, predictions were remarkably similar.
    3. Many families were accurately predicted by all, many inaccurately by all.

Additional Thoughts: Missingness

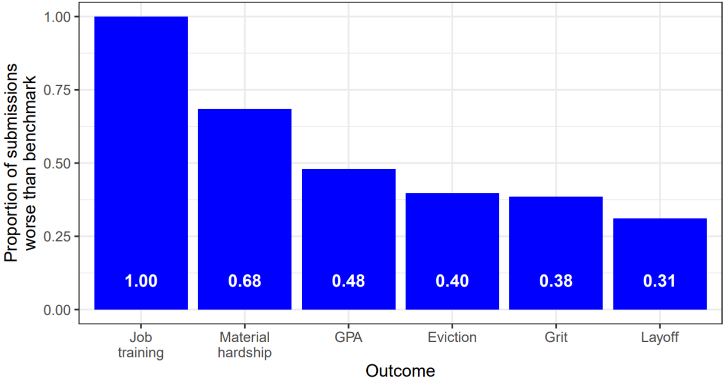




- Of 55m possible entries, (4242 × 12942): 73% of missing.

1. 17% were not in the survey wave.
2. 0.1% refusal to answer was about.
3. 25% due to skip patterns.
4. 6% redacted for privacy/ethics.

Additional Thoughts: Underperformance



- Between 31% (`layoff`)/100% (`training`) are worse than benchmark.
- **WOW!** Special credit to anyone who builds good models for `jobTraining`.

Additional Thoughts: Self-reported narratives

- **A lot** of work went into pre-processing and accounting for constants/missingness.
- A large number of models used random forests.
- A **really** interesting quote from Ridhi's team's submission:

> 'Human-informed subsetting reduces predictive performance'

- Ridhi's team used MTurk to get contact details of FFCWS authors, surveying them.
- A really interesting mix of teams which used expert knowledge, and those using 'aggressive feature selection'.
- Interestingly: some *internal* ensemble work.
- "*It pays off for engineers to 'make friends' with the FFCWS codebooks*"

Additional Thoughts: More self-reported narratives

- Some approaches were very basic: it doesn't mean they did badly!
- Some very autoregressive models: "the past predicts the future."
- 'My approach to the Fragile Families Challenge was to ... create parsimonious but effective model': great!
- 'I have conducted research into eviction law.'
- 'I only trained models for the continuous responses."!

"This was really a joint effort with my professor, Jeremy Freese, and 15 other colleagues who took the course."

Additional Thoughts: Various

- Each account was permitted to upload 10 submissions per day!
  - This resulted in a lot of over-fitting to the leader-board set.
- To construct benchmark models, the `Amelia` package was used.
- The chosen winner depended partly on the luck of the holdout set.
- Combining (weighted) submissions didn't substantially improve performance.
- This is largely due to the high correlation between the teams.
- Families hardest to predict are far from the mean of the training data.
- The main ethical and privacy concern is through re-identification attacks

Is that the case closed, then? Sociology

- Three aspects of timing may decrease predictability:
  1. Six year gap between waves 5 and 6.
  2. A large social disruption (the Great Recession).
  3. The prediction of wave 6 comes at a time when the child was 15 years old.
     - This may be a particularly turbulent time for children and families.
- A largely urban and disadvantaged group living in the contemporary United States may have more unpredictable lives than other groups.
- The results would have been qualitatively different if we selected different outcomes from wave 6 (age 15).

Is that the case closed, then? Data Science/Statistics

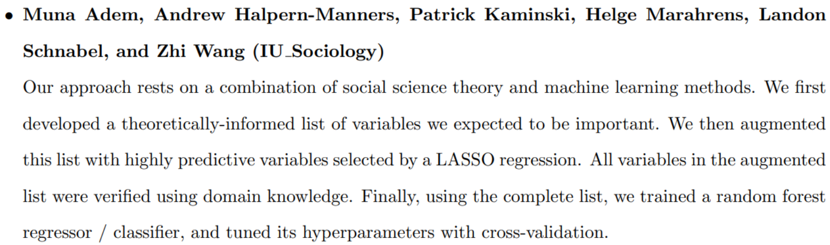
- Three (four?) reasons to be skeptical.
  1. **The curse of dimensionality**:
     - With just 4,242 families, there isnt *that* much space for ML to stretch its legs!
  2. Researchers ‘had access to thousands of predictor variables’: did they?
  3. Of the 12,942 variables, 2,358 were constant.
     - ‘Missing-ness’ is substantial (see following slide).
  4. How homogeneous was the community?
     - Mostly, social data scientists, as opposed to deep learning engineers.
  5. (How appropriate is the R^2_{Holdout} metric?)
- Or does it mean – as Garip (2020, PNAS) puts it:

“. . . *that life outcomes are too idiosyncratic and subject to a predictability ceiling*”.

Your Fragile Families Submission : Some advice!

- Doesnt have to be complex!
- Draw on and Sociological Theory, and Demographic Analysis!
- You can consider 'long seeded childhood characteristics' (e.g. birthweight).
- One submission: 'We read through the literature to find substantively important variables.'
- One submission: 'I used existing sociological theory to identify features'.
- One submission: 'I drew on past research'.
    - My advice: Pick one variable, explore the literature related to it.
    - Two of the Socius SI papers for example focus exclusively on GPA.
    - Build a model as simple or complex as you like!
    - Importantly: **Describe and explain your choices**!
- Total freedom, within the bounds of building *any* model and evaluating it.

Your Fragile Families Submission : Some advice! (Cont.)



- **Muna Adem, Andrew Halpern-Manners, Patrick Kaminski, Helge Marahrens, Landon Schnabel, and Zhi Wang (IU_Sociology)**

  Our approach rests on a combination of social science theory and machine learning methods. We first developed a theoretically-informed list of variables we expected to be important. We then augmented this list with highly predictive variables selected by a LASSO regression. All variables in the augmented list were verified using domain knowledge. Finally, using the complete list, we trained a random forest regressor / classifier, and tuned its hyperparameters with cross-validation.

- Ultimately, something akin to this would be the upper limit of **excellent**!
- Four steps:

1. Data Preparation
2. Variable selection (manual, automated, or hybrid)
3. Training and testing
4. Evaluating

If you haven't already done this: Do it now!

Please Sign Up for the Fragile Families Dataset!

- Please sign up for a copy **now**! Steps to do this:
  1. Please create an account at Princeton University's OPR data archive.
  2. Read the Overview of FFCWS.
  3. Click to 'Sign Up'.
  4. In the box for justification, enter something like:

I am requesting the Fragile Families Challenge data (`ffchallenge_papers_replication_materials.zip` **and** `FFChallenge_v5.zip`) in order to participate in Life Course Research taught as part of the MPhil in Sociology and Demography at the University of Oxford, convened by Jennifer Dowd, Ridhi Kashyap, and Charles Rahal.

- Patiently wait. But sign up now! **They are expecting your application!**

A Request...

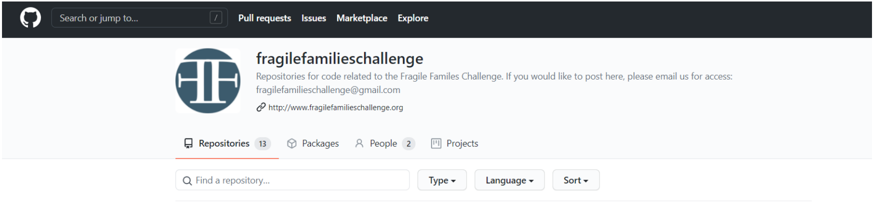
- I have been requested – as I should be – to request three things of you:

1. Please do not post the FFC data online.
2. Please do not share the FFC data with anyone.
3. Please destroy the FFC data when you have submitted the assignment.

The FFC: Baseline code

- Lets now move on to applying some of the ML tools we've learned about.
- What better place than the FFC itself!?
- This should act as a 'kickstarter' for your coursework projects.
- Note: I've provided code for all **six** of the key FFC variables.
    - They're largely equivalent: there is a minor difference with how to visualize binary variables.
    - The pre-processing is in the majority taken care of by an FFC-provided script.
- For you to do: more variables, sociological interpretation, new models, more visualisation.

The FFC: The GitHub



fragilefamilieschallenge

Repositories for code related to the Fragile Familes Challenge. If you would like to post here, please email us for access: fragilefamilieschallenge@gmail.com

http://www.fragilefamilieschallenge.org

- This above is the FFC GitHub!
- You can find **lots** of scripts and things here for inspiration.
- Importantly:

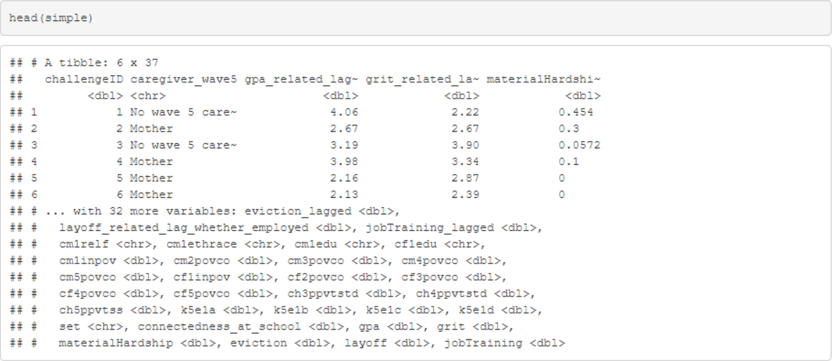
  fragilefamilieschallenge/teaching-support/prepare_simple_data.R

- Unzip ffchallenge_papers_replication_materials.zip and FFChallenge_v5.zip into a single subdirectory.
- Set your working directory, then run `prepare_simple_data()`

The FFC: The simple dataset

- You should now see three new example_prepared_data files in your FFChallenge_v5 folder.
- You can use this for your projects! What has this script done for us?
  1. It's imputed missing values using the `amelia` package.
  2. It's created lagged wave 5 predictors for the benchmarks.
  3. It's reverse coded and created composite variables where appropriate.
- It's got all 4,242 families in it, and 37 variables (including '`challengeID`').
- Note: some variables i.e. `cm1relf` (marital status of mother wave 1) and `cm1edu` (mothers education wave one) are *factor* variables.

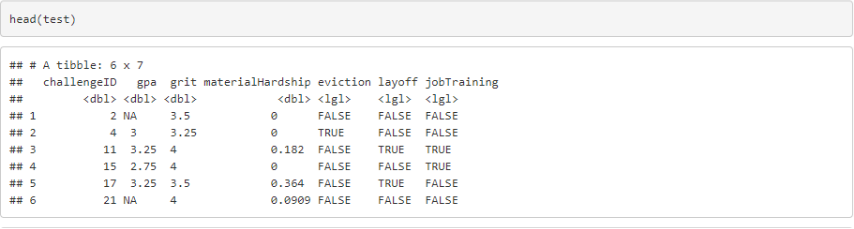
The FFC: The simple dataset – what does it look like?



```
head(simple)
```

```
## # A tibble: 6 x 37
##   challengeID caregiver_wave5 gpa_related_lag~ grit_related_la~ materialHardshi~
##         <dbl> <chr>                      <dbl>            <dbl>            <dbl>
## 1           1 No wave 5 care~             4.06             2.22           0.454
## 2           2 Mother                      2.67             2.67           0.3
## 3           3 No wave 5 care~             3.19             3.90           0.0572
## 4           4 Mother                      3.98             3.34           0.1
## 5           5 Mother                      2.16             2.87           0
## 6           6 Mother                      2.13             2.39           0
## # ... with 32 more variables: eviction_lagged <dbl>,
## #   layoff_related_lag_whether_employed <dbl>, jobTraining_lagged <dbl>,
## #   cm1relf <chr>, cm1ethrace <chr>, cm1edu <chr>, cf1edu <chr>,
## #   cm1inpov <dbl>, cm2povco <dbl>, cm3povco <dbl>, cm4povco <dbl>,
## #   cm5povco <dbl>, cf1inpov <dbl>, cf2povco <dbl>, cf3povco <dbl>,
## #   cf4povco <dbl>, cf5povco <dbl>, ch3ppvtstd <dbl>, ch4ppvtstd <dbl>,
## #   ch5ppvtss <dbl>, k5e1a <dbl>, k5e1b <dbl>, k5e1c <dbl>, k5e1d <dbl>,
## #   set <chr>, connectedness_at_school <dbl>, gpa <dbl>, grit <dbl>,
## #   materialHardship <dbl>, eviction <dbl>, layoff <dbl>, jobTraining <dbl>
```

The FFC: The test dataset – what does it look like?



```r
head(test)
```

```
## # A tibble: 6 x 7
##   challengeID   gpa  grit materialHardship eviction layoff jobTraining
##         <dbl> <dbl> <dbl>            <dbl> <lgl>    <lgl>  <lgl>
## 1           2 NA     3.5            0      FALSE    FALSE  FALSE
## 2           4  3     3.25           0      TRUE     FALSE  FALSE
## 3          11  3.25  4              0.182  FALSE    TRUE   TRUE
## 4          15  2.75  4              0      FALSE    FALSE  TRUE
## 5          17  3.25  3.5            0.364  FALSE    TRUE   FALSE
## 6          21 NA     4              0.0909 FALSE    FALSE  FALSE
```

Lets now focus on one example: `materialHardship`

This is a composite variable, averaged over 11 questions (condensed for brevity):

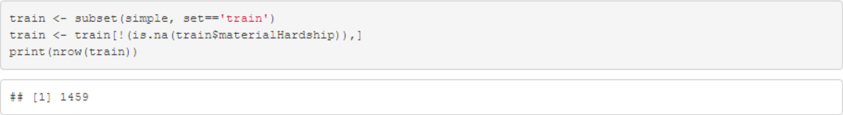
1. Did you receive free food or meals?
2. Were you ever hungry, but didn't eat because you couldn't afford to?
3. Did you ever not pay the full amount of rent or mortgage?
4. Were you evicted from your home or apartment for non-payment?
5. Did you not pay the full amount of gas, oil, or electricity bill?
6. Was your gas or electric services ever turned off?
7. Did you borrow money from friends or family to help pay bills?
8. Did you move in with other people because of financial problems?
9. Did you stay at a shelter, in an abandoned building, or any other place?
10. Was there anyone in your household who needed healthcare but couldn't go?
11. Was your telephone service canceled or disconnected?

If any one of the composites was NA, then `materialHardship` was also NA.

`materialHardship`: Lets check our ‘missingness’

- Lets check that our training data has the right n which links with Table S4 of the PNAS Supplementary Material.

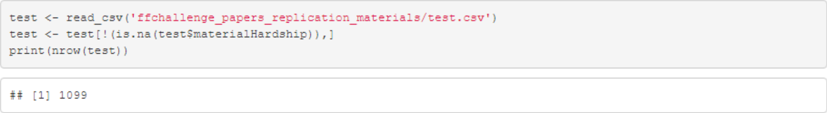
.



```
train <- subset(simple, set=='train')
train <- train[!(is.na(train$materialHardship)),]
print(nrow(train))
```

```
## [1] 1459
```

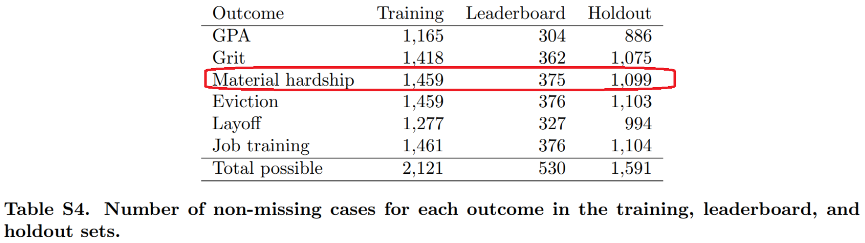
- And then do the same for the test data (which we’ve loaded in from `ffchallenge_papers_replication_materials`):



```
test <- read_csv('ffchallenge_papers_replication_materials/test.csv')
test <- test[!(is.na(test$materialHardship)),]
print(nrow(test))
```

```
## [1] 1099
```

`materialHardship`: Lets check our 'missingness'

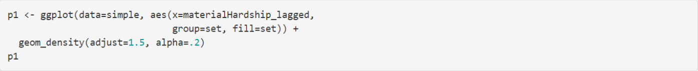


Outcome	Training	Leaderboard	Holdout
GPA	1,165	304	886
Grit	1,418	362	1,075
Material hardship	1,459	375	1,099
Eviction	1,459	376	1,103
Layoff	1,277	327	994
Job training	1,461	376	1,104
Total possible	2,121	530	1,591

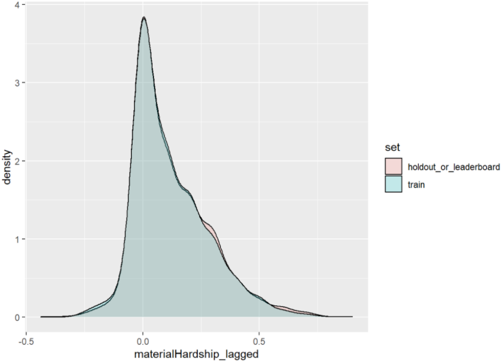
Table S4. Number of non-missing cases for each outcome in the training, leaderboard, and holdout sets.

- I cannot begin to emphasize how important this is!
- The main reason why papers don't replicate exactly is because of different n after preprocessing.
- You'll learn more about this in the Replication Project next term...

materialHardship: Basic EDA



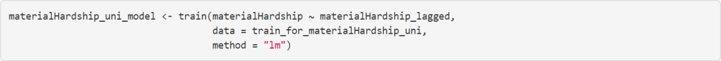
```
p1 <- ggplot(data=simple, aes(x=materialHardship_lagged,
                              group=set, fill=set)) +
  geom_density(adjust=1.5, alpha=.2)
p1
```



- We always want to do some basic ‘Exploratory Data Analysis’!
- Check: distribution of materialHardship roughly the same across splits.

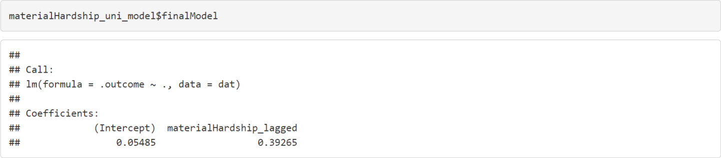
materialHardship: Model Building

- Lets now build a simple *univariate* model.
- Contains only lagged wave 5 predictor, and intercept (by default):
- Note: we're building the model based on the **training** data.



```
materialHardship_uni_model <- train(materialHardship ~ materialHardship_lagged,
                                    data = train_for_materialHardship_uni,
                                    method = "lm")
```

- Like a regular regression (this is one), we can see the fit coefficients:

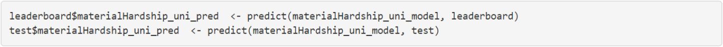


```
materialHardship_uni_model$finalModel
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Coefficients:
##            (Intercept)  materialHardship_lagged
##                0.05485                  0.39265
```

materialHardship: Model Evaluating

- Lets use this model (‘gpa_uni_model’) to create predictions:



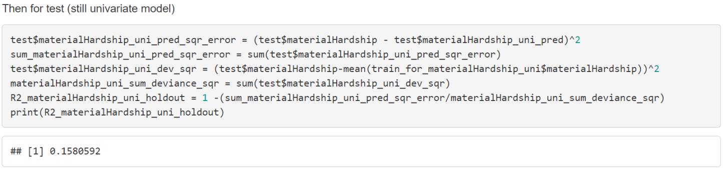
```
leaderboard$materialHardship_uni_pred  <- predict(materialHardship_uni_model, leaderboard)
test$materialHardship_uni_pred  <- predict(materialHardship_uni_model, test)
```

- Recall that the original PNAS article evaluated the models as:

:

$$R^2_{holdout} = 1 - \frac{\sum_{i\in\text{Holdout}}(y_i - \hat{y}_i)^2}{\sum_{i\in\text{Holdout}}(y_i - \bar{y}_{\text{Training}})^2} \qquad (2)$$

- Lets build this calculation in as verbose a fashion as possible!



Then for test (still univariate model)

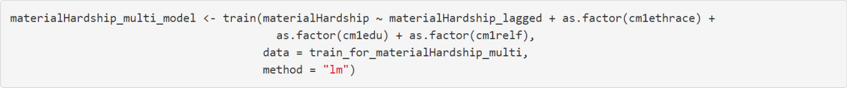
```
test$materialHardship_uni_pred_sqr_error = (test$materialHardship - test$materialHardship_uni_pred)^2
sum_materialHardship_uni_pred_sqr_error = sum(test$materialHardship_uni_pred_sqr_error)
test$materialHardship_uni_dev_sqr = (test$materialHardship-mean(train_for_materialHardship_uni$materialHardship))^2
materialHardship_uni_sum_deviance_sqr = sum(test$materialHardship_uni_dev_sqr)
R2_materialHardship_uni_holdout = 1 -(sum_materialHardship_uni_pred_sqr_error/materialHardship_uni_sum_deviance_sqr)
print(R2_materialHardship_uni_holdout)
```

```
## [1] 0.1580592
```

- Not bad for a *very* simple univariate model (note, full benchmark is 18%)!

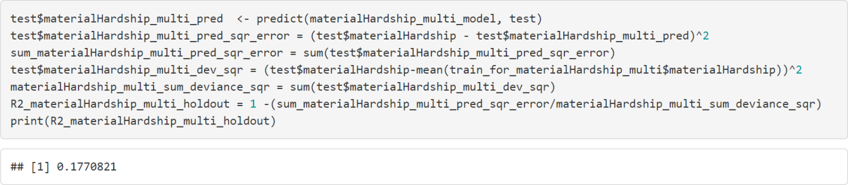
materialHardship: Model Evaluating

- Lets now try and get as close to the benchmark as we can:



```
materialHardship_multi_model <- train(materialHardship ~ materialHardship_lagged + as.factor(cm1ethrace) +
                                        as.factor(cm1edu) + as.factor(cm1relf),
                                      data = train_for_materialHardship_multi,
                                      method = "lm")
```

- And now, evaluate this multivariate model in exactly the same way:

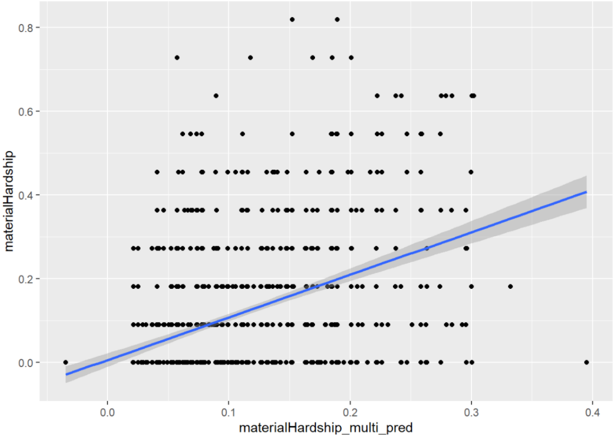


```
test$materialHardship_multi_pred  <- predict(materialHardship_multi_model, test)
test$materialHardship_multi_pred_sqr_error = (test$materialHardship - test$materialHardship_multi_pred)^2
sum_materialHardship_multi_pred_sqr_error = sum(test$materialHardship_multi_pred_sqr_error)
test$materialHardship_multi_dev_sqr = (test$materialHardship-mean(train_for_materialHardship_multi$materialHardship))^2
materialHardship_multi_sum_deviance_sqr = sum(test$materialHardship_multi_dev_sqr)
R2_materialHardship_multi_holdout = 1 -(sum_materialHardship_multi_pred_sqr_error/materialHardship_multi_sum_deviance_sqr)
print(R2_materialHardship_multi_holdout)
```

```
## [1] 0.1770821
```

`materialHardship`: Plot Residuals

- As always, we want to do as much data visualization as possible.



- Looking good! A horizontal line would be the mean of the training data.

Concluding thoughts: Replicating the benchmark

- The boilerplate scripts which I've created for you are written in *very bad* R!
- However, they do everything that they should. You can write better code!
- There are a few reasons why the code doesnt match up *exactly*.
- I've put them into this table on the next slide:
  - *published_value*: the value in the FFC supp matt
  - *rep_value*: the exact value in the FFC replication materials which I've ran
  - *corrected_replication_value*:corrected value which I think is right
  - *our_value*: exact value in the FFC replication materials which I've given you
- You can use any benchmark you want: just be transparent!
- You might think I'm being extreme in my desire for accuracy in replication (these numbers are all fairly similar).
- Scientific integrity is important, and you'll come back to this in the Replication Project next year!

Concluding thoughts: Replication Table

	published	rep	corrected	ours
gpa	0.11	0.104688	n/a	0.136404
grit	0.01	0.014337	n/a	0.006296
materialHardship	0.18	0.183014	n/a	0.177082
eviction	0.02	0.018233	0.01423	0.015071
jobTraining	0.05	0.051886	0.04936	0.041165
layoff	0.01	0.008326	0.009149	0.006686

- We can see that the correlation with the basic code I've prepared for you is already extremely high.
- But a couple of items don't match up: why's that?

Concluding thoughts: Why no exact replications?

- First of all, I've just found a substantial error in the FFC benchmark code.
- This relates to all models they claim being evaluated as logits, actually having been created by OLS (by mistake) on line 266 of their Harvard Dataverse (`aggregate_score_analyses.R`) files!
- There are also two other reasons why our benchmarks dont match.
- The pre-processor (prepare_simple_data.R) makes two important types of simplifications:
  1. Amelia only makes one stochastic pass (even though I set the same seed)
  2. Some simplifications are made, i.e. in feature engineering variables like `cm1relf` into fewer categories (to avoid the 'small cell' problem.
- To repeat and conclude: don't worry about this too much at all, just document about be transparent about what **you** do!
- (p.s. Try and enjoy the challenge where possible, and please do contact me for all the help that you need!)