04/13 update

- Preliminary results on a small, **balanced** subsample (5000 cases, 5000 controls).
- Training set of 9000 IDs, test set of 1000 IDs.
- Imputed missing variables with median.
- Lab data (blood tests, FOBT) and colonoscopies converted to longitudinal summaries (mean,max,min,...).
- Lab data restricted to time interval from 'indexdate - 3 years' to 'indexdate - 1 year'.
- No ICD code information, no medication information, no smoking status records (smoking status at index is in demographic variables).
- Logistic regression.

Predictors (p = # of predictors)	**Training AUC**	**Test AUC**
Demographic ($p = 10$)	0.814	0.818
Charlson ($p = 18$)	0.705	0.693
Demographic + Charlson ($p = 28$)	0.863	0.862
Labs ($p = 210$)	0.824	0.824
Demographic + Charlson + Labs ($p = 238$)	0.913	0.910

04/20 update

- Preliminary results on a small, **balanced** subsample (5000 cases, 5000 controls).
- Training set of 9000 IDs, test set of 1000 IDs.
- Imputed missing variables with median.
- **Sets of predictors:**
  - Demographic data includes race, age, weight, etc. (**Demographic**).
  - Charlson score inputs plus GERD diagnosis (**Charlson**).
  - Lab data (blood tests, FOBT) and colonoscopy data converted to longitudinal summaries (mean,max,min,...) (**Labs**).
  - Medication (H2R and PPI) converted to longitudinal summaries (mean,max,...) (**Meds**).
- Lab and medication data restricted to time interval from 'indexdate - 3 years' to 'indexdate - 1 year'.
- Logistic regression.

Predictors (p = # of predictors)	**Training AUC**	**Test AUC**
Demographic ($p = 10$)	0.814	0.818
Charlson ($p = 18$)	0.705	0.693
Meds ($p = 10$)	0.604	0.598
Demographic + Charlson + Labs + Meds ($p = 240$)	0.915	0.912

04/27 update

- Baseline logistic regression on entire sample (no blood labs, no medication data).
- Full sample contains $n = 6,649,108$ observations, with $n_{\text{control}} = 6,637,713$ and $n_{\text{case}} = 11,395$.
- Imputed numeric variables with medians, imputed smoking status at random, proportional to non-missing in sample (45% current, 41% former, 14% never).

Predictors (p = # of predictors)	**Training AUC**	**Test AUC**
Demographic ($p = 10$)	0.670	0.675
Charlson ($p = 18$)	0.684	0.705
Demographic + Charlson ($p = 28$)	0.760	0.770

Smoking status missingness.

In the entire sample:

Smoking Status	Current	Former	Never	Missing
Count	1696052	1521806	537727	2893523
Probability	0.26	0.23	0.08	0.44

Among cases:

Smoking Status	Current	Former	Never	Missing
Count	3901	3546	594	3354
Probability	0.34	0.31	0.05	0.29

Among controls:

Smoking Status	Current	Former	Never	Missing
Count	1692151	1518260	537133	2890169
Probability	0.25	0.23	0.08	0.44

05/04 update

- Preliminary results on a balanced **random sample** of 5000 cases and 5000 controls.
- Most missing variables imputed with their median, SmokeStatus imputed at random proportional to non-missing.
- Training on 9000 observations, testing on 1000 observations.
- Baseline logistic regression fit with subsets of predictors, and all predictors.
- Random forest fit with all predictors.

Model/predictors (p = # of predictors)	**Training AUC**	**Test AUC**
Logistic regression	-	-
Demographic ($p = 10$)	0.669	0.676
Charlson ($p = 18$)	0.689	0.659
Meds ($p = 10$)	0.515	0.513
Colonoscopies ($p = 2$)	0.513	0.502
Labs ($p = 200$)	0.671	0.621
All ($p = 240$)	**0.814**	**0.778**
Random forest	-	-
All ($p = 240$)	**0.985**	**0.844**

Random forest variable importance. Most important blood lab measurements (all means of measurements over the 2 year window):

1. Hematocrit value (CBC labs)
2. MCH value (CBC labs)
3. ALT value (LFT labs)
4. Alk. Phos. value (LFT labs)
5. AST value (LFT labs)
6. White blood cells (WBC) value (CBC labs)
7. Glucose value (BMP labs)

Next steps:

- Better approaches to impute missing variables (esp. SmokeStatus).
- Improved non-linear classifiers for subsample.
- Logistic regression baselines for the full sample of 6M observations.

05/11 update

- Baseline logistic regression on full sample (no medication data).
- $n = 6,649,108$ observations, with $n_{\text{control}} = 6,637,713$ and $n_{\text{case}} = 11,395$.
- Most missing variables imputed with their median, SmokeStatus imputed at random proportional to non-missing.
- For lab information, only means of each measurement, no longitudinal information so far.
- 05/27 update: Charlson and Demographic+Charlson updated to exclude Cancer and Metastatic Carcinoma.

Model/predictors (p = # of predictors)	**Training AUC**	**Test AUC**
Demographic ($p = 10$)	0.670	0.675
Charlson ($p = 16$)	0.684	0.705
Demographic + Charlson ($p = 26$)	0.760	0.770
Colonoscopies ($p = 2$)	0.511	0.504
Labs ($p = 34$)	0.604	0.601
Meds ($p = 10$)	0.523	0.512

Next steps:

- Process medication data.
- Approaches to impute missing variables (esp. SmokeStatus).
- Non-linear classifiers for subsample, plots/graphics to interpret predictor effects.

05/18 update

- Medication data processed for full sample, new line (red) added to last week's table of baseline logistic regression AUCs.
- Gradient boosting implemented for the random sample of 5000 cases and 5000 controls.

Model/predictors (p = # of predictors)	**Training AUC**	**Test AUC**
Logistic regression	-	-
All ($p = 240$)	0.814	0.778
Random forest	-	-
All ($p = 240$)	**0.985**	0.844
Gradient boosting	-	-
All ($p = 240$), no interactions	0.900	0.847
All ($p = 240$), 2-way interactions	0.924	0.857
All ($p = 240$), 3-way interactions	0.937	**0.864**

Same gradient boosting results without Charlson score inputs or GERD at index:

Model/predictors (p = # of predictors)	**Training AUC**	**Test AUC**
Gradient boosting	-	-
No Charlson inputs ($p = 222$), no interactions	0.881	0.811
No Charlson inputs ($p = 222$), 2-way interactions	0.898	0.820
No Charlson inputs ($p = 222$), 3-way interactions	0.909	0.817

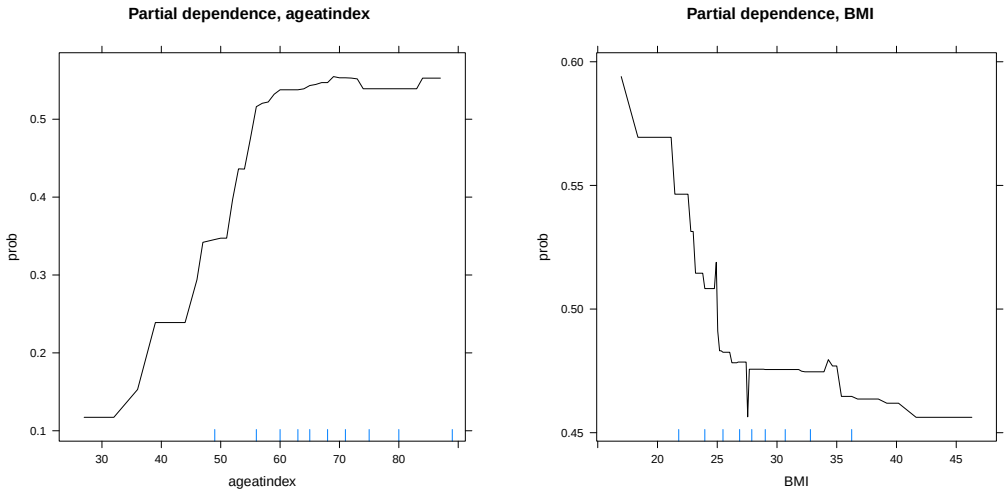
Next steps

- Scaling up gradient boosting, dealing with unbalanced classes in full sample.
- Plots/graphics to interpret black box models.

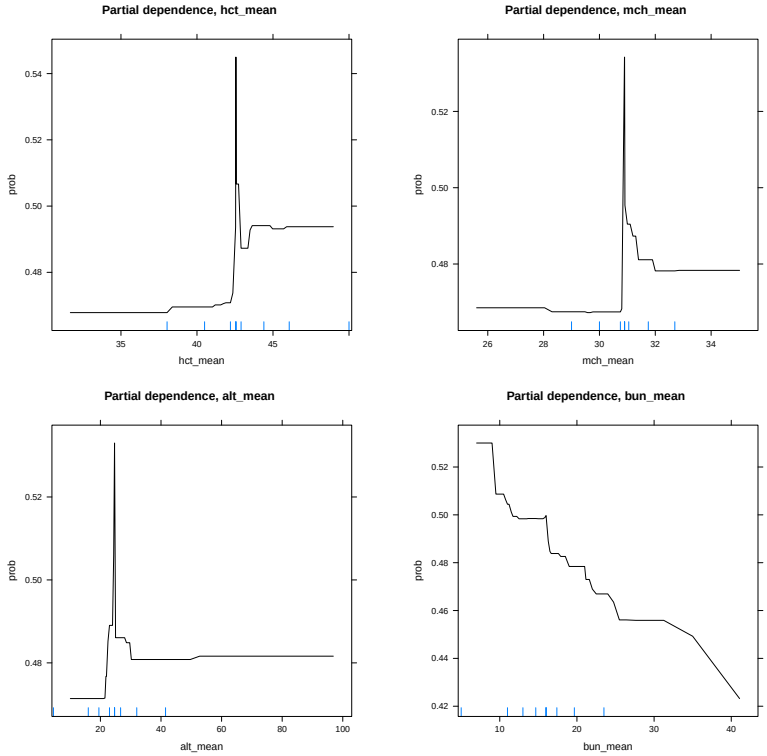
05/25 update

- Results now without Cancer and Metastatic Carcinoma inputs to Charlson score.
- Implemented scalable `xgboost` model, test AUC **0.860**.

Partial dependence plots (from xgboost model). Demographic variables:



Most influential blood lab variables:



Missingness rates for blood lab variables (where 'missing' means zero measurements of that variable in the 2 year window):

Proportion missing	controls	cases
A1c_mean (A1C Labs)	0.51	0.52
bun_mean (BMP Labs)	0.17	0.30
hct_mean (CBC Labs)	0.20	0.50
CRP_mean (CRP Labs)	0.95	0.94
alkphos_mean (LFT Labs)	0.23	0.47
chol_mean (Lipid Labs)	0.19	0.32

08/20: now with the new data, 4 year window from index-1 to index-5

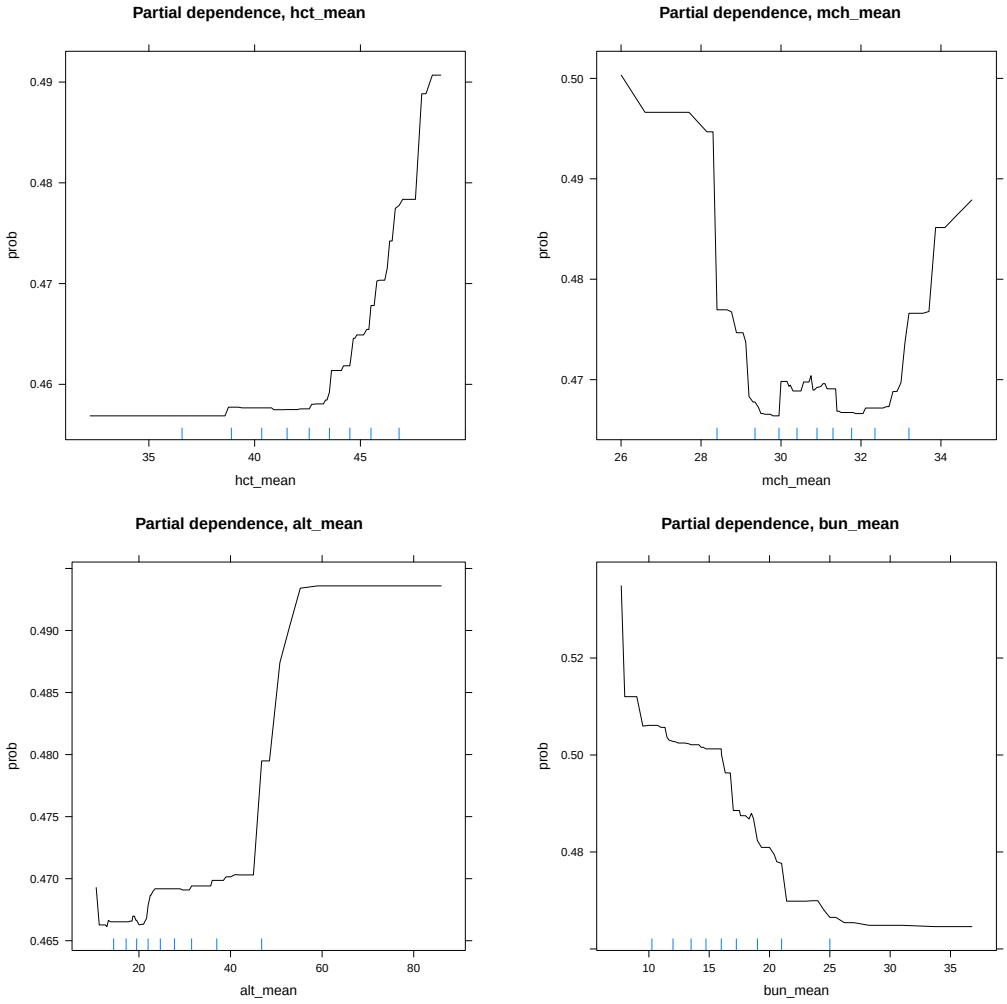
Proportion missing	controls	cases
A1c_mean (A1C Labs)	0.43	0.47
bun_mean (BMP Labs)	0.11	0.26
hct_mean (CBC Labs)	0.12	0.46
CRP_mean (CRP Labs)	0.93	0.91
alkphos_mean (LFT Labs)	0.15	0.42
chol_mean (Lipid Labs)	0.12	0.29

uniformly fewer missing, but rates are not cut in half, moreover heterogeneity of missingness between cases and controls is not fixed.

06/08 update

- Re-coded BMI/weight measurements?
- Re-run xgboost on balanced subsample with `NA`s, test AUC **0.880**, but it is using missing values to identify cases.
- Source of missingness? More detailed breakdown for LFT Labs.

Updated partial dependence plots for blood lab variables. Using new xgboost model on subsample.



Detailed breakdown of missingness for LFT labs. (Prediction window is second and third years before index)

Mean # of labs	controls	cases
First year before index	1.87	2.42
Second year before index	1.47	1.52
Third year before index	1.42	1.34

Detailed breakdown for the prediction window (index minus 3 years to index minus 1 year):

Proportion	controls	cases
At least 1 lab	0.83	0.69
At least 2 labs	0.67	0.59
At least 3 labs	0.45	0.44
At least 4 labs	0.30	0.31
At least 5 labs	0.18	0.21

Next Steps:

- Imputation with xgboost
- Implement xgboost for entire sample

06/22 update

- Set up code on GitLab (waljee-zhu-ml-projects/hosea-project).
- Received updated sample with recoded BMIs. BMIs are more likely to be missing for cases (14%) than controls (8%).
- Continuing to code imputation following Deng and Lumley (2021).

BMI comparison on the original sample data:

Original BMI	< 20	$\in (20, 25]$	$\in (25, 30]$	$\in (30, 35]$	$\in (35, 40]$	> 40
$\mathbb{P}(Control\|BMI)$	0.9960	0.9978	0.9985	0.9985	0.9986	0.9986
$\mathbb{P}(Case\|BMI)$	0.0040	0.0022	0.0015	0.0015	0.0014	0.0014

With the re-coded sample data:

New BMI	< 20	$\in (20, 25]$	$\in (25, 30]$	$\in (30, 35]$	$\in (35, 40]$	> 40
$\mathbb{P}(Control\|BMI)$	0.9983	0.9985	0.9985	0.9984	0.9981	0.9979
$\mathbb{P}(Case\|BMI)$	0.0017	0.0015	0.0015	0.0016	0.0019	0.0021

06/29 update

- Now using updated data (new BMIs, Charlson scores).
- For subsample, comparison of baseline logistic regression and xgboost models for different imputation approaches:
    1. **No imputation**: leave NAs in the data, learn a classification for subjects with missing data (only compatible with xgboost).
    2. **Median imputation**: impute all variables with median/most common class.
    3. **Regression imputation**: impute by (linear) regression of each predictor variable on the others (similar to MICE).
    4. **Random sample imputation**: impute each predictor by sampling at random from the non-missing entries.

AUC results for xgboost/logistic regression for different imputation approaches. Results still on balanced subsample.

Imputation method/model	**Training AUC**	**Test AUC**	**Test AUC (complete records)**
No imputation	-	-	-
xgboost	0.977	0.942	0.643
Median imputation	-	-	-
logistic regression	0.834	0.810	0.585
xgboost	0.970	0.857	0.623
Regression imputation	-	-	-
logistic regression	0.805	0.760	0.557
xgboost	0.948	0.815	0.705
Random sample imputation	-	-	-
logistic regression	0.724	0.689	0.724
xgboost	0.919	0.729	**0.745**

Regression imputation still has 'regression to mean' effect for imputed values, imputed values have smaller variance than the true values (with observation error). Maybe try a version that imputes a rank then resamples from observed values.

07/06 update

- Source of missingness in the data? About 85% of cases missing Charlson score inputs (compared to about 47% of controls).
- Missingness in test set/use case?
- Updated (in red) last weeks results with a test set of 1000 complete records (226 cases, 774 controls).
  - Poor generalization to complete cases for no imputation, median imputation implies that those models are fitting to the missingness, patterns won't continue to hold with fully observed data.
  - Regression imputation still has 'regression to mean' effect for imputed values, imputed values have smaller variance than the observed values (observation error).
  - Good generalization of random sample imputation implies that there is no bias introduced from training on data with missingness.

07/13 update

- Should we evaluate the model on complete records or missing/imputed records?
- Recall: previous results with an additional test set of 1000 complete records (226 cases, 774 controls).
  - New comparison for **multiple** random sample imputation: imputes by sampling from non-missing entries several times (reps) for each missing value, training on the multiple imputed records.
- Memory issues fitting xgboost on the full data, implementing "batched" training.

AUC's for xgboost/logistic regression for different imputation approaches. Results still on balanced subsample.

Imputation method/ prediction model	**Training AUC**	**Test AUC (imputed records)**	**Test AUC (complete records)**
No imputation	-	-	-
xgboost	0.977	0.942	0.643
Median imputation	-	-	-
logistic regression	0.834	0.810	0.585
xgboost	0.970	0.857	0.623
Regression imputation	-	-	-
logistic regression	0.805	0.760	0.557
xgboost	0.948	0.815	0.705
Random sample imputation	-	-	-
logistic regression	0.724	0.689	0.724
xgboost	0.919	0.729	0.745
Multiple rand. samp.	-	-	-
xgboost, 10 reps	0.909	0.742	0.781
xgboost, 20 reps	0.939	0.765	0.820
xgboost, 30 reps	0.948	0.768	0.781

Final two columns suggest two different evaluation metrics, ensuring the model performs well on **both** (A) new patients with imputed data and (B) new patients with fully observed blood labs, etc.

07/20 update

- Continuing to test/tune imputation approaches.
- Notes on decision curve analysis.

Imputation method/ prediction model	**Training AUC**	**Test AUC (imputed records)**	**Test AUC (complete records)**
Separate class	-	-	-
xgboost	0.977	0.942	0.643
Median	-	-	-
logistic regression	0.834	0.810	0.585
xgboost	0.970	0.857	0.623
Regression	-	-	-
logistic regression	0.805	0.760	0.557
xgboost	0.948	0.815	0.705
Regression v2	-	-	-
logistic regression	0.769	0.706	0.564
xgboost	0.928	0.722	0.764
Random sample	-	-	-
logistic regression	0.724	0.689	0.724
xgboost	0.919	0.729	0.756
Multiple rand. samp.	-	-	-
xgboost, 10 imputes $\times$ 50 trees	0.909	0.742	0.781
xgboost, 20 imputes $\times$ 25 trees	0.906	0.750	0.778
xgboost, 30 imputes $\times$ 20 trees	0.909	0.765	0.788
xgboost, 100 imputes $\times$ 5 trees	0.898	0.762	0.806

07/27 update

- Continuing to tweak regression imputation approaches.
- Comparing parameters/number of imputations needed for multiple sample imputation.
- Notes on decision curve analysis.

Next steps:

- Re-process new data when available

Imputation method/ prediction model	**Training AUC**	**Test AUC (imputed records)**	**Test AUC (complete records)**
Separate class	-	-	-
xgboost	0.977	0.942	0.643
Median	-	-	-
logistic regression	0.834	0.810	0.585
xgboost	0.970	0.857	0.623
Regression	-	-	-
logistic regression	0.759	0.704	0.657
xgboost	0.902	0.742	0.757
Single rand. samp.	-	-	-
logistic regression	0.724	0.689	0.724
xgboost	0.919	0.729	0.756
Multiple rand. samp.	-	-	-
xgboost, 10 imputes	0.909	0.742	0.781
xgboost, 20 imputes	0.906	0.750	0.778
xgboost, 30 imputes	0.909	0.765	0.788
xgboost, 100 imputes	0.898	0.762	0.806

08/03 update

- Received new data, loading/processing files
- Still issues with missing Charlson scores (0's vs NAs), relationship between CaseControl indicator and number of visits.

08/17 update

- Processed longitudinal summaries for BMP labs. Missingness rates down but still different for cases and controls. Continuing to process new blood lab data.
- Incidence rates for some example Charlson inputs: Peptic Ulcer Disease, Renal Disease, GERD. Full tables for all inputs are on the next page.

Variable **(ignoring NA's)**	Among cases	Among controls
pud	6.8% (120/1762)	3.8% (142821/3803451)
RD	11.8% (208/1762)	12.1% (460302/3803451)
GERD	-	-

Variable **(impute NA's as 0's)**	Among cases	Among controls
pud	1.1% (120/11395)	2.2% (142821/6637713)
RD	1.8% (208/11395)	6.9% (460302/6637713)
GERD	17.9% (2044/11395)	15.7% (1039206/6637713)

Cross-classified by number of BMP lab results (0, 1 or 2+):

Peptic Ulcer Disease, **controls**:

Number of BMP labs	0	1	NA
0	29.3%	0.9%	69.8%
1	32.2%	0.9%	66.8%
2+	61.5%	2.5%	36.0%

Peptic Ulcer Disease, **cases**:

Number of BMP labs	0	1	NA
0	8.6%	0.7%	90.8%
1	10.7%	0.4%	88.8%
2+	17.0%	1.3%	81.7%

Full tables for all Charlson indicators (red indicates these variables have been dropped from past models):

Variable **(ignoring NA's)**	Among cases	Among controls
CANCER	30.5% (538/1762)	18.3% (695558/3803451)
CHF	12.3% (217/1762)	11.8% (450090/3803451)
CTD	2.9% (51/1762)	3.0% (115492/3803451)
DEM	1.4% (24/1762)	2.3% (85704/3803451)
DIAB_C	15.5% (273/1762)	13.8% (525226/3803451)
HIV	0.3% (6/1762)	0.8% (30507/3803451)
MET_CAR	6.6% (117/1762)	1.3% (47696/3803451)
MLD	9.1% (160/1762)	8.0% (305462/3803451)
MSLD	0.7% (13/1762)	0.8% (29495/3803451)
PARA	1.9% (33/1762)	1.9% (72051/3803451)
RD	11.8% (208/1762)	12.1% (460302/3803451)
cd	16.3% (287/1762)	15.3% (582328/3803451)
copd	43.1% (759/1762)	35.9% (1365577/3803451)
diab_nc	44.6% (786/1762)	44.8% (1702629/3803451)
mi	9.1% (161/1762)	7.0% (264731/3803451)
pud	6.8% (120/1762)	3.8% (142821/3803451)
pvd	20.1% (354/1762)	16.0% (609425/3803451)
GERD	-	-

Variable **(impute NA's as 0's)**	Among cases	Among controls
CANCER	4.7% (538/11395)	10.5% (69558/6637713)
CHF	1.9% (217/11395)	6.8% (450090/6637713)
CTD	0.4% (51/11395)	1.7% (115492/6637713)
DEM	0.2% (24/11395)	1.3% (85704/6637713)
DIAB_C	2.4% (273/11395)	7.9% (525226/6637713)
HIV	0.1% (6/11395)	0.5% (30507/6637713)
MET_CAR	1.0% (117/11395)	0.7% (47696/6637713)
MLD	1.4% (160/11395)	4.6% (305462/6637713)
MSLD	0.1% (13/11395)	0.4% (29459/6637713)
PARA	0.3% (33/11395)	1.1% (72051/6637713)
RD	1.8% (208/11395)	6.9% (460302/6637713)
cd	2.5% (287/11395)	8.8% (582328/6637713)
copd	6.7% (759/11395)	20.6% (1365577/6637713)
diab_nc	6.9% (786/11395)	25.6% (1702629/6637713)
mi	1.4% (161/11365)	4.0% (264731/6637713)
pud	1.1% (120/11395)	2.2% (142821/6637713)
pvd	3.1% (354/11395)	9.2% (609425/6637713)
GERD	17.9% (2044/11395)	15.7% (1039206/6637713)

08/31 update

- Re-fit xgboost models on new subsampled data (compare to 07/27 results). $n_{\text{train}} = 9000$, $n_{\text{test}} = 1000$, $n_{\text{complete.test}} = 1000$.
- Compare (xgboost) model performance with different groups of variables included.
- Ongoing: a proxy for number of visits to better impute Charlson ICD codes?

Comparing different imputation approaches and prediction models:

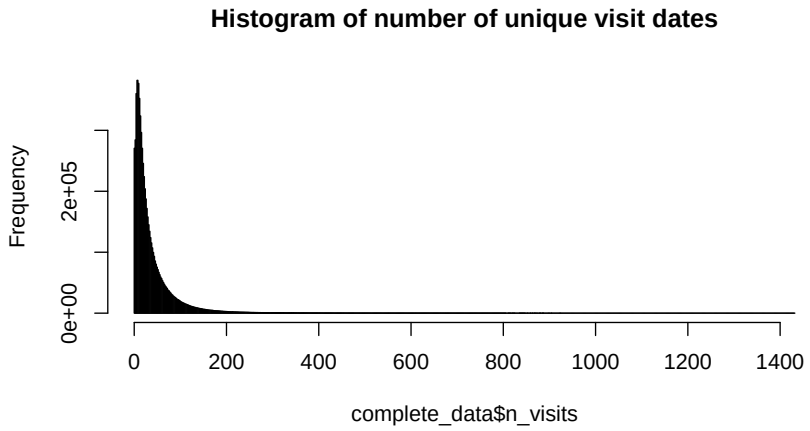
Imputation method/ prediction model	**Training AUC**	**Test AUC (imputed records)**	**Test AUC (complete records)**
Separate class	-	-	-
xgboost	0.985	0.962	0.582
Median	-	-	-
logistic regression	0.853	0.846	0.583
xgboost	0.963	0.911	0.618
Regression	-	-	-
logistic regression	0.772	0.726	0.651
xgboost	0.911	0.750	0.798
Single rand. samp.	-	-	-
logistic regression	0.739	0.695	0.703
xgboost	0.916	0.780	0.833
Multiple rand. samp.	-	-	-
xgboost, 10 imputes	0.942	0.824	0.885
xgboost, 20 imputes	0.935	0.820	0.872
xgboost, 30 imputes	0.943	0.810	0.879
xgboost, 100 imputes	0.931	0.824	0.891

Comparing different included variables (imputing with single random sampling, fit with xgboost):

Variables included	**Training AUC**	**Test AUC (imputed records)**	**Test AUC (complete records)**
Demographic ($p = 11$)	0.743	0.700	0.663
Charlson + GERD ($p = 16$)	0.600	0.542	0.530
Medications ($p = 10$)	0.594	0.501	0.539
Labs ($p = 202$)	0.903	0.687	0.829

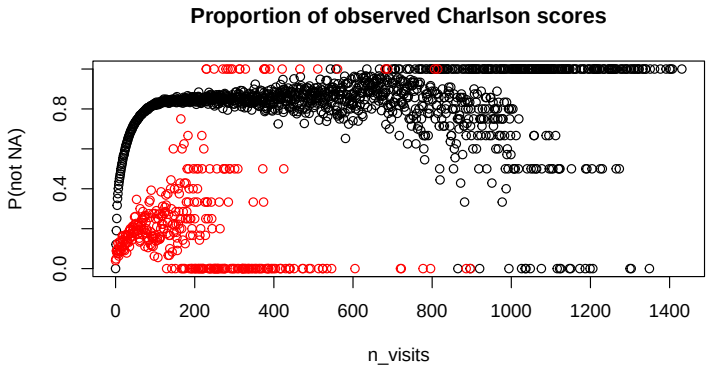
09/21 update

Overall missingness of Charlson score inputs. A histogram of the total number of visits in the 4-year prediction window:

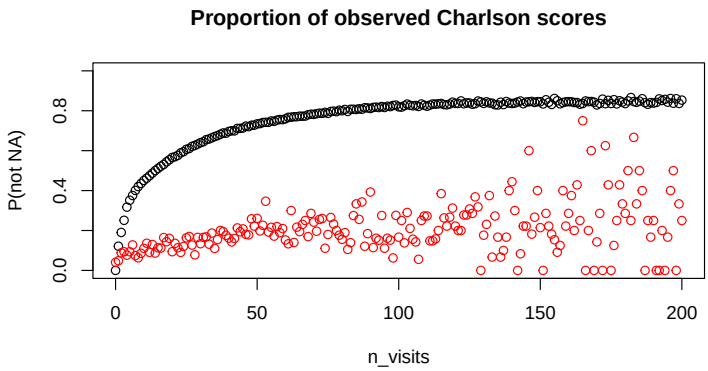


The median control has 22 visits, while the median case has 33 visits.

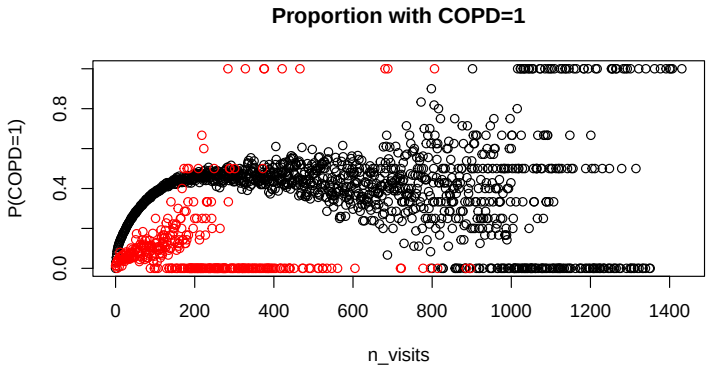
Proportion of observed Charlson scores plotted against number of visits, separating controls (black) and cases (red):



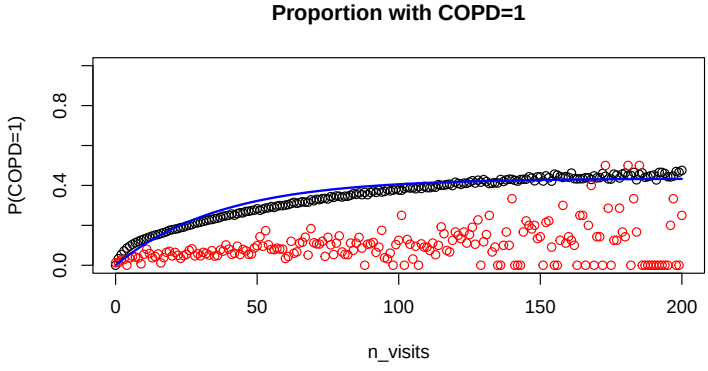
The same plot restricted to ≤ 200 visits.



Example: missingness of COPD. Proportion of patients coded COPD=1, plotted against number of visits, separating controls (black) and cases (red):



The same plot restricted to ≤ 200 visits. The blue line is a model-based estimate of $\mathbb{P}(\text{COPD} = 1|\text{n\_visits})$ using pooled case and control data.

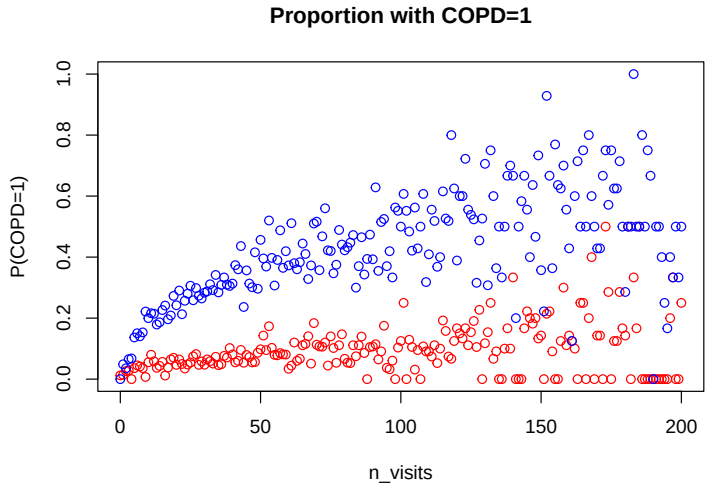


For cases only, created a new COPD indicator based on the 'alldxscx' table. For ICD9, 1 means you have a code '49x', for ICD10, 1 means you have a code starting with 'J44'.

Cross-classification of old and new indicator for total of 11,395 cases (overall proportions in parentheses).

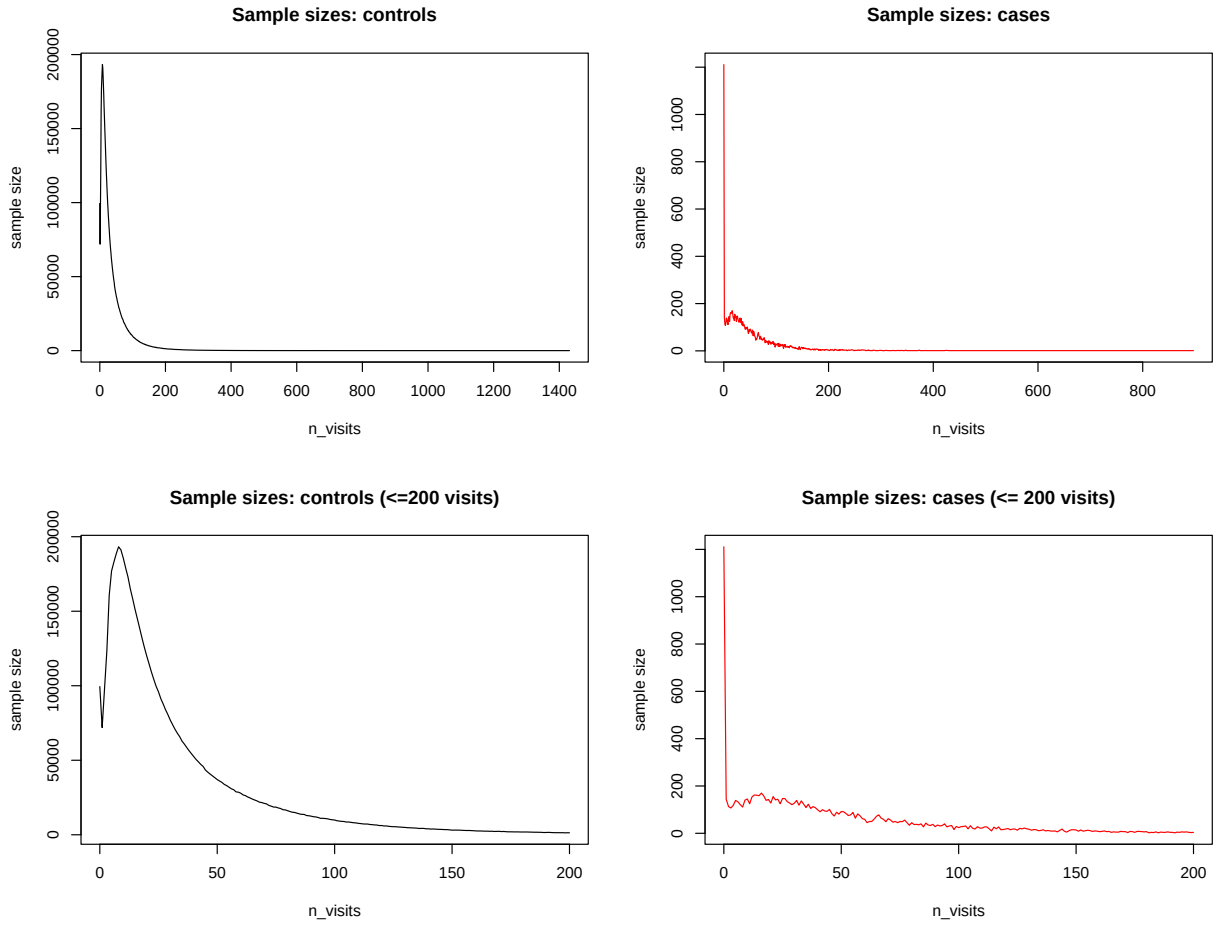
	New COPD=1	New COPD=0/NA
Old COPD=1	7845 (0.688)	2791 (0.245)
New COPD=0/NA	160 (0.014)	599 (0.053)

Proportion of patients with COPD=1 plotted against number of visits, restricted to ≤ 200 visits, with the old indicator in red and the new indicator in blue.



Next steps: similarly code a new indicator for controls, see if it changes results. Generalize to other Charlson inputs (or other entirely new ICD codes), and look at ways to impute by incorporating n_visits.

Sample sizes for controls and cases for calculating the proportions in the above plots.



09/28 update

- Implemented new train/valid/test scheme
  - Imputed records results are not exactly comparable
  - Complete records results are comparable
- Experimentation with regression imputation
- Experimentation with `xgboost` tuning parameters
- Currently working on multiple random sample with new train/valid/test scheme

Imputation method	**Training AUC**	**Test AUC (imputed records)**	**Test AUC (complete records)**
Separate class	0.974 (0.985)	0.953 (0.962)	0.607 (0.582)
Median	0.979 (0.963)	0.939 (0.911)	0.649 (0.618)
Regression	0.952 (0.911)	0.739 (0.750)	0.770 (0.798)
1 rand. samp.	0.987 (0.916)	0.783 (0.780)	0.890 (0.833)
10/04 update			
2 rand. samp.	1.000	0.801	0.918
5 rand. samp.	0.977	0.807	0.928
10 rand. samp.	1.000 (0.942)	0.818 (0.824)	0.915 (0.885)
20 rand. samp.	0.953 (0.935)	0.804 (0.820)	0.909 (0.872)
50 rand. samp.	0.978	0.814	0.912
100 rand. samp.	0.975 (0.931)	0.822 (0.824)	0.925 (0.891)

*all with `xgboost` prediction model; AUCs in parentheses denote previous results.

10/04 update

- Implemented new train/valid/test scheme with multiple imputation
- Updated table on previous page with result with multiple imputation
- Some improvement with a single random simpling imputation
- Some improvement compared to previous results, mostly for "complete records" test set
- Next step: utilize all of the data

10/12 update

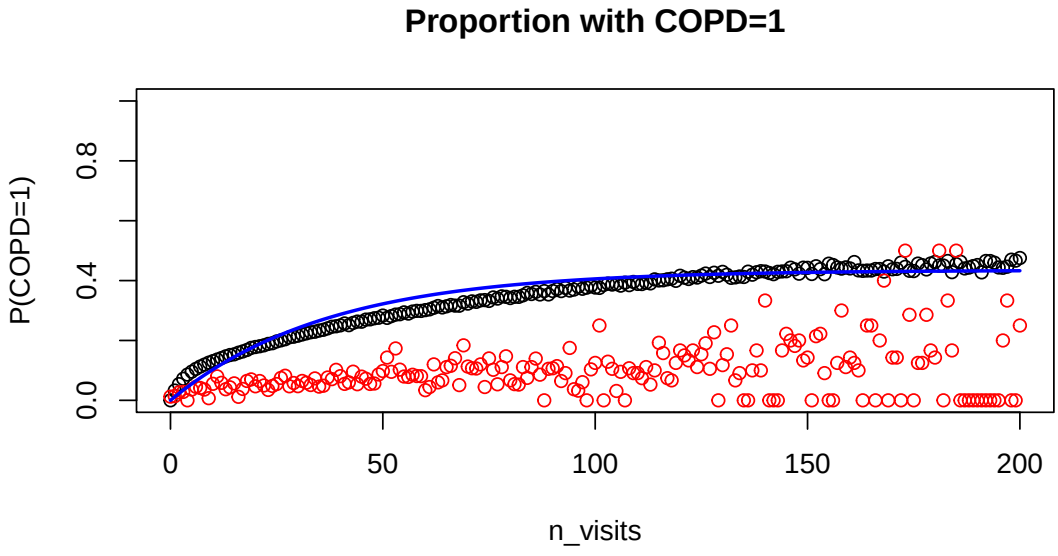
- Redo analysis with various Charlson variable treatments:
  - no Charlson
  - sampling from observed values (what was done previously)
  - imputing a probability using number of visit (see Peter's proposed Geometric model)
  - imputing by sampling using the fitted probability
- Results:
  - see table below
  - dropping Charlson leads to a small decrease in testing performance for both imputed and "complete" records
  - sampling with fitted probability has similar performance compared to the previous methods
  - imputing the fitted probability leads to a huge improvement in performance
- Currently examining where this large improvement comes from, still suspicious
- Try the same with multiple random samples
- Working on utilizing all data ...

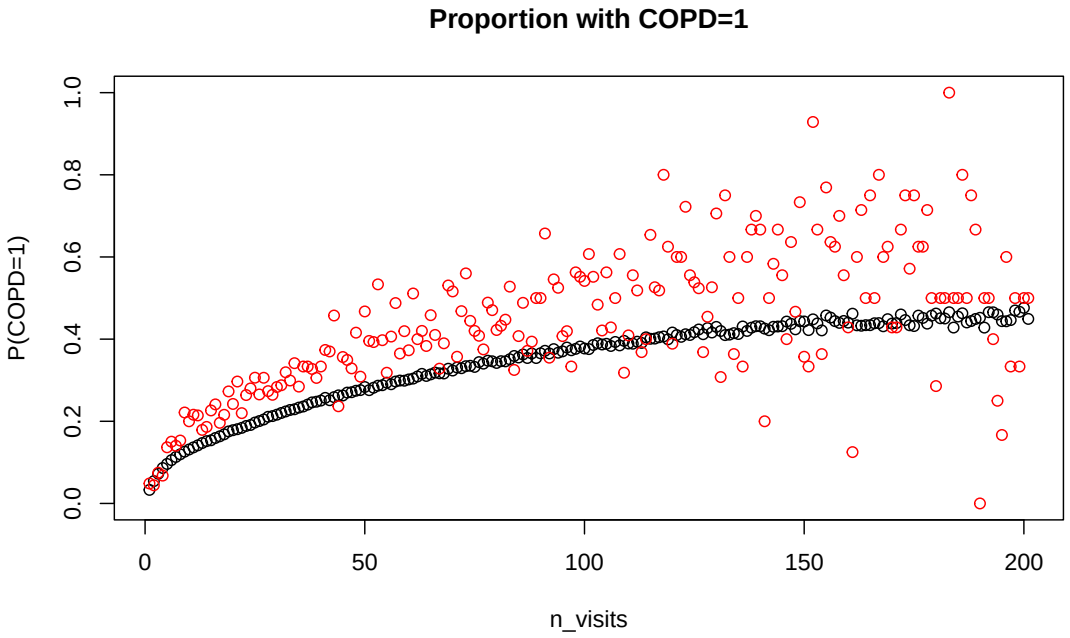
Imputation method	**Training AUC**	**Test AUC (imputed)**	**Test AUC (complete)**
Separate class	0.974	0.953	0.607
Median	0.990	0.941	0.674
Simple random sample			
no Charlson	0.988	0.771	0.873
random Charlson	0.999	0.786	0.905
proba. Charlson	1.000	0.906	0.975
proba.-weighted random Charlson	0.997	0.795	0.896

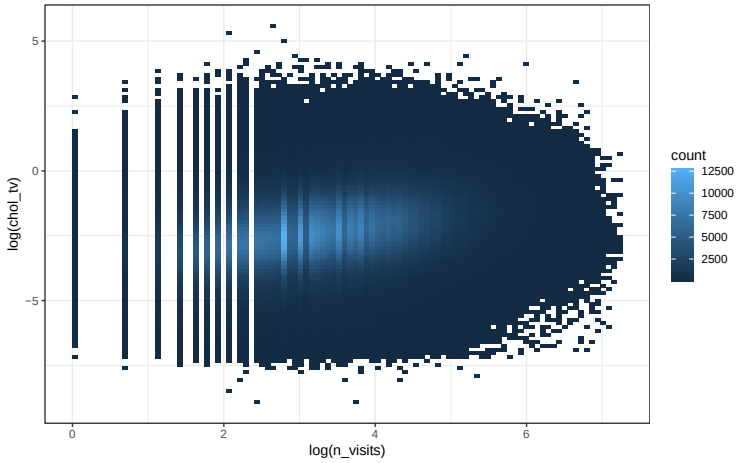
10/26 update

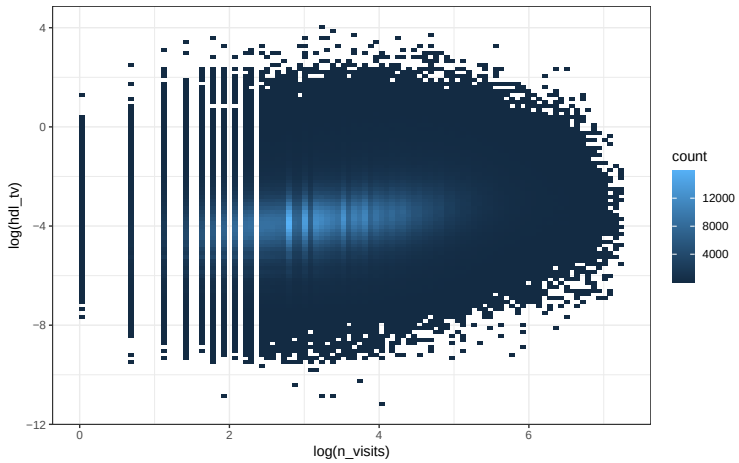
Some follow-ups:

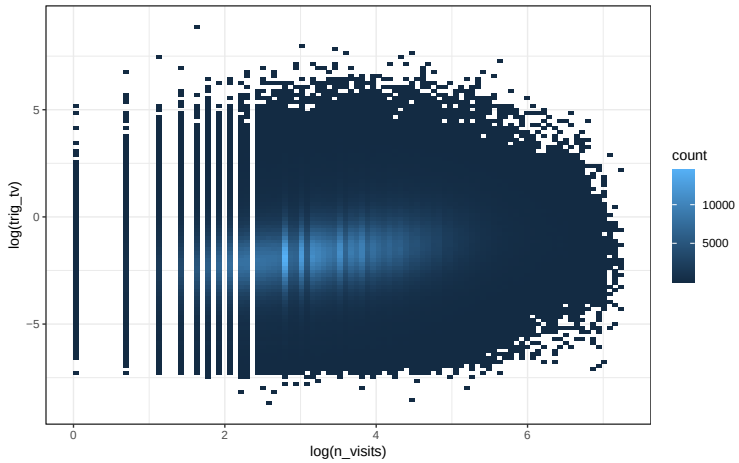
- Using number of visits to impute Charlson/GERD variables:
  - Imputing probabilities is not a good idea as it is easy to spot missing values (between 0 and 1) from observed values (0/1)
  - randomizing using the probability does not improve performance
  - preferable to have a unified treatment
- ICD codes reconding for cases comparing before and after
  - we no longer see the large discrepancy between cases and control
  - see COPD graphs below
- Does the "tv" lab summaries contain some information about the number of measurements?
  - there were some concern that variability would be associated with numer of measurements
  - see density plots for "chol", "hdl" and "trig" below
  - "tv" stands for "total variation" and is the average absolute change between measurement (standardized by time between measurements)
  - there does not seem to be strong associations




Proportion with COPD=1
P(COPD=1)
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
n_visits

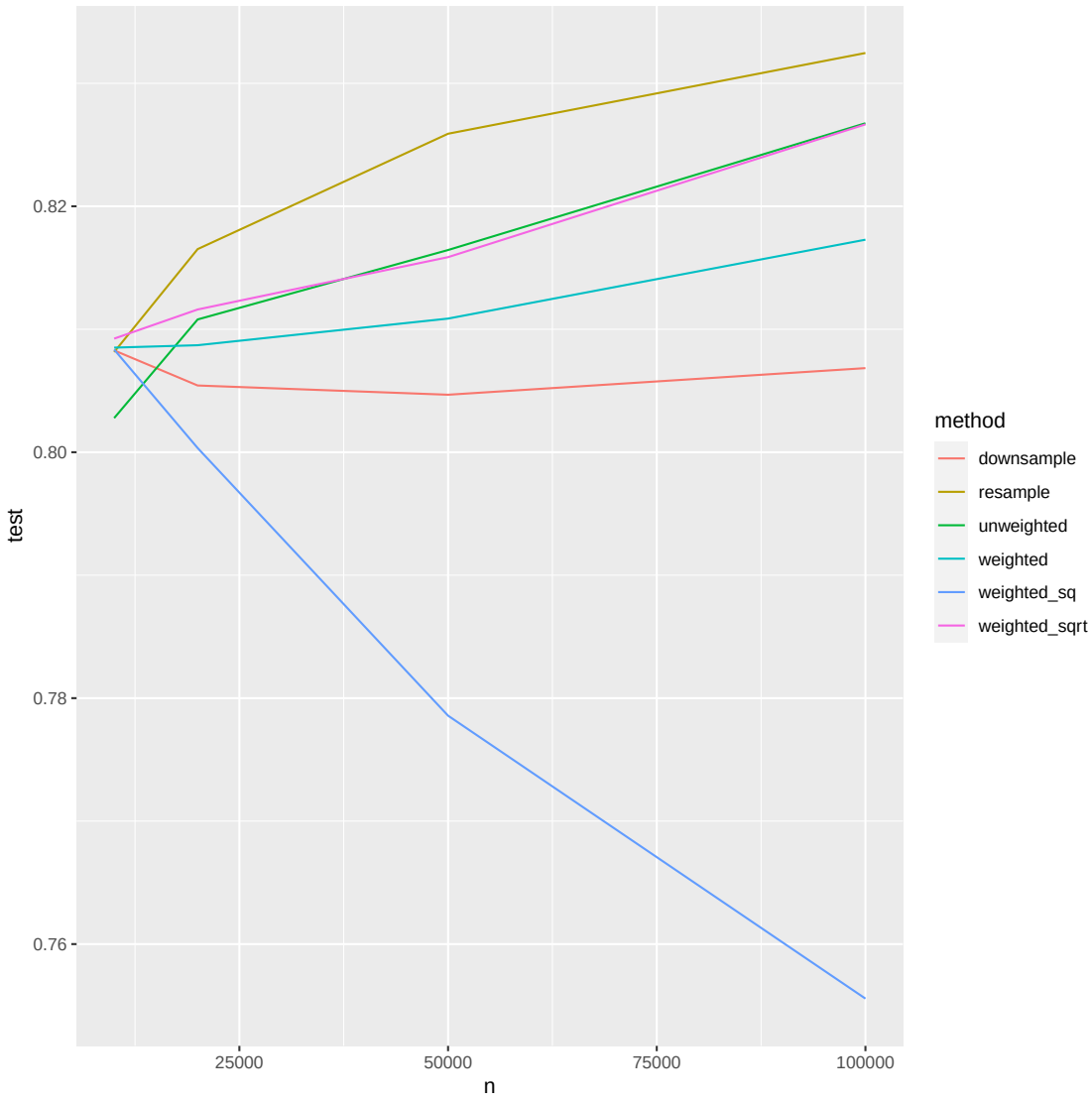

log(chol_tv)
5
0
−5
0
2
4
6
log(n_visits)
count
12500
10000
7500
5000
2500

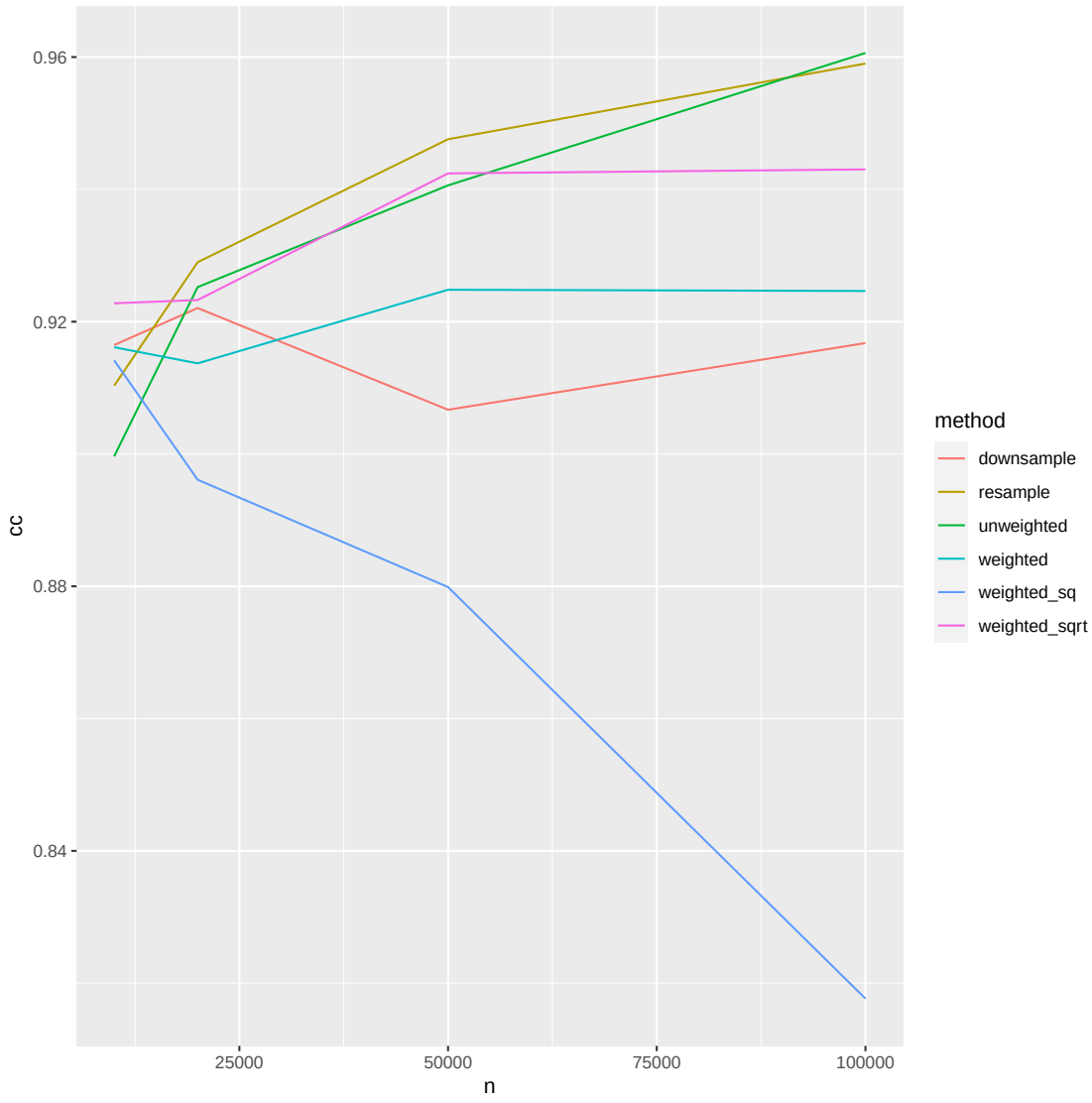

log(hdl_tv)
4
0
−4
−8
−12
0
2
4
6
log(n_visits)
count
12000
8000
4000


log(trig_tv)
5
0
−5
0
2
4
6
log(n_visits)
count
10000
5000

Using all data:

- Scalability study
- Sample 1M controls, keep all 11.4K cases (1M/11.4K)
- Train-test split 75-25 stratified
  - Fixed test set (250K/2.8K)
  - Training using increasing subsets of (750K/8.5K)
- Working training set consisting of downsampling the controls
  - n =10K, 20K, 50K, 100K
  - keeping all 8.5K cases
  - as n increases, the training set becomes more and more unbalanced
  - n =10K is close to balanced
- Validation set: stratified 2-1
  - Model/imputation trained on 67%, validation/model selection on 33%
- Extra testing set: "complete cases" (same as before, 710/290)
  - small, with possible overlap
  - not a super reliable testing set
- Methods (e.g. with 100K/8.5K $\rightarrow$ 66.7K/5.7K)
  - Unweighted: do nothing special
  - Downsample: sample controls to get a balance training set (5.7K/5.7K)
  - Resample: resample cases to get a balance training set (66.7K/66.7K)
  - Weighted: cases' loss reweighted by $w = 66.7/5.7$
  - Weighted_ sqrt: by $\sqrt(w)$
  - Weighted_ sq: by w^2


0.82
0.80
0.78
0.76
test
25000
50000
75000
100000
n
method
downsample
resample
unweighted
weighted
weighted_sq
weighted_sqrt


0.96
0.92
0.88
0.84
cc
25000
50000
75000
100000
n
method
downsample
resample
unweighted
weighted
weighted_sq
weighted_sqrt

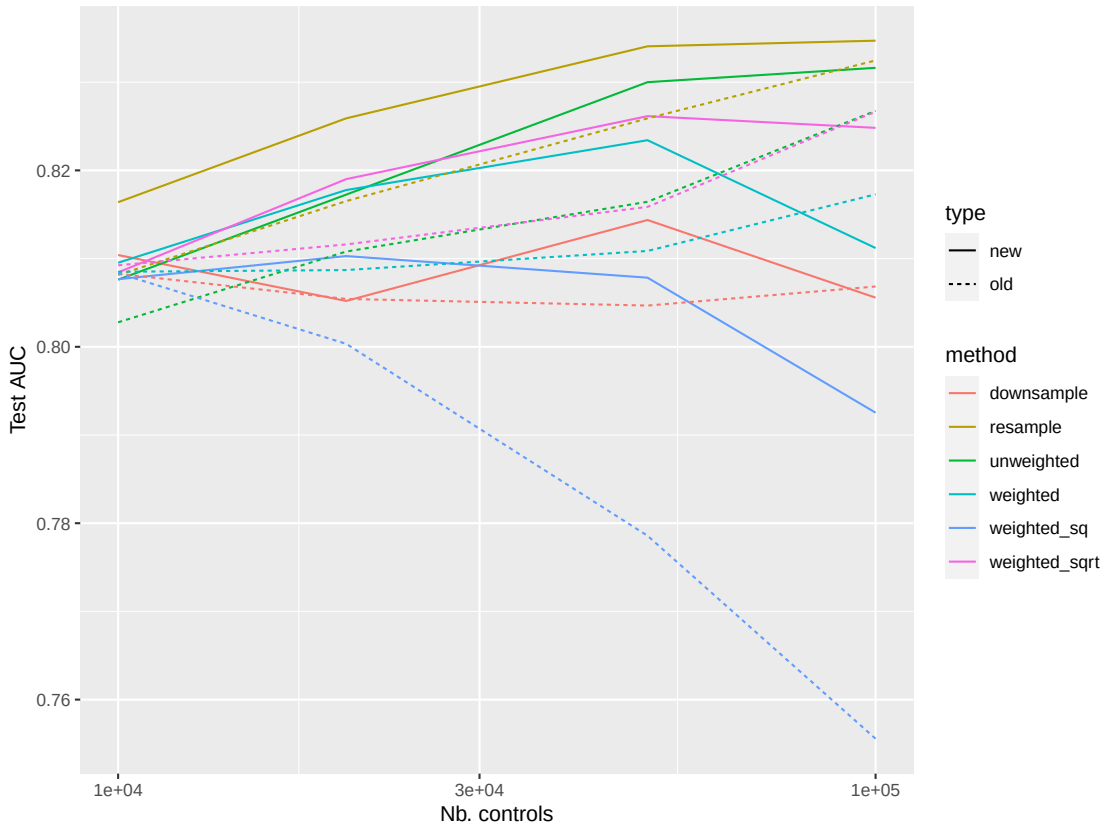
11/02 update

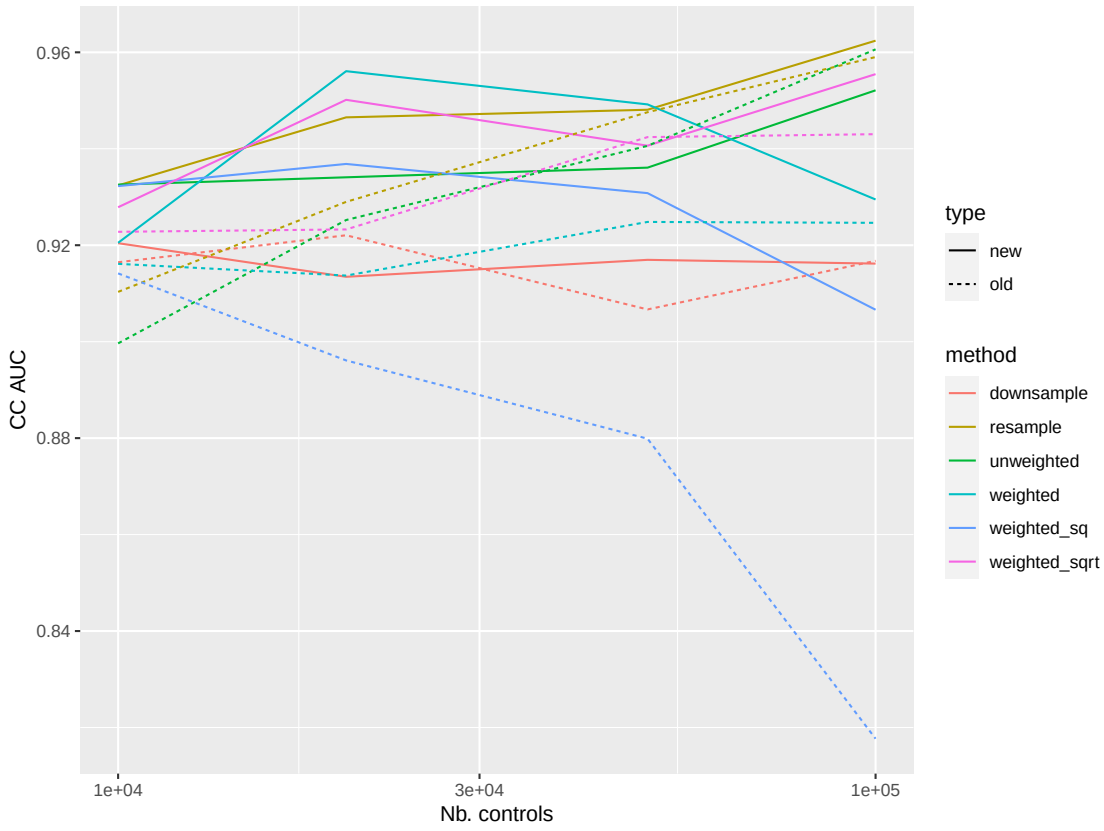
Tuning

- subsampling each iteration, learning rate, tree depth
- Some improvement is still possible
- Dependent on the training set size
  - Tuned on small smaple size, improvments don't necessarly carry to larger sample size
  - prohibitive to tune on large dataset

Training set size

- Training becomes *very* slow
- Early results show improvement beyond 100K
- Exploring ways to speed up training (lightgbm, catboost)


Test AUC
0.82
0.80
0.78
0.76
1e+04
3e+04
1e+05
Nb. controls
type
new
old
method
downsample
resample
unweighted
weighted
weighted_sq
weighted_sqrt


0.96
0.92
0.88
0.84
CC AUC
1e+04
3e+04
1e+05
Nb. controls
type
new
old
method
downsample
resample
unweighted
weighted
weighted_sq
weighted_sqrt

11/09 update

Training set size

- Added results for $> 100K$ controls
- Goal:
  - Harness all data
  - Study how the increasing case/control imbalance affects results
- Similar train/valid/test split as before
  - 2M control + 11K cases (increased from 1M previously to allow the 1M case)
  - All splits are stratified (cases/control proportion preserved)
  - test: 25%, train+valid: 75%
  - N: number of controls in train+valid
  - subsample train+valid controls down to N
  - train: 67%, valid: 33%
  - NB: test set is fixed with N, validation set increases with N
- Imputation:
  - Simple random sampling from the training set into validation and testing sets
- `xgboost` parameters:
  - Default values, except:
  - `subsample = 0.5`
  - `max_depth = 5`
  - `eta = 0.5` (stepsize)
  - NB: these values are represented with vertical dashed lines in the following "Tuning" section
- Methods:

  downsample sample controls down to the number of cases

  resample sample (w/ replacement) the cases up to the number of controls (note that controls appearing more than once get different imputation)

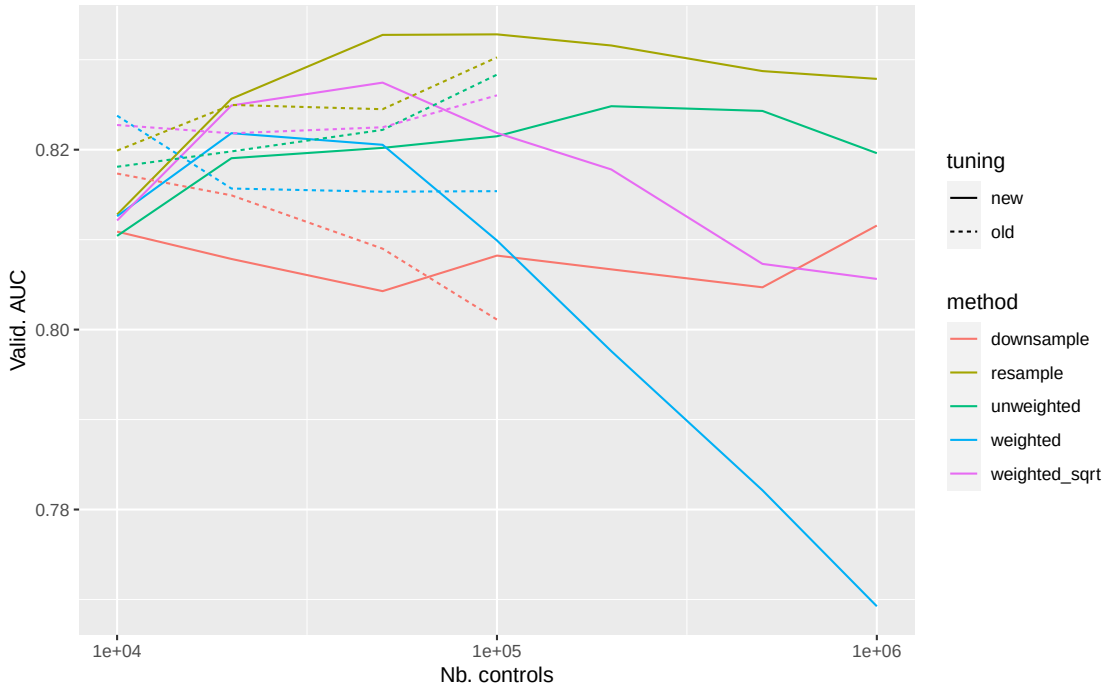
  unweighted no resample, no reweighting

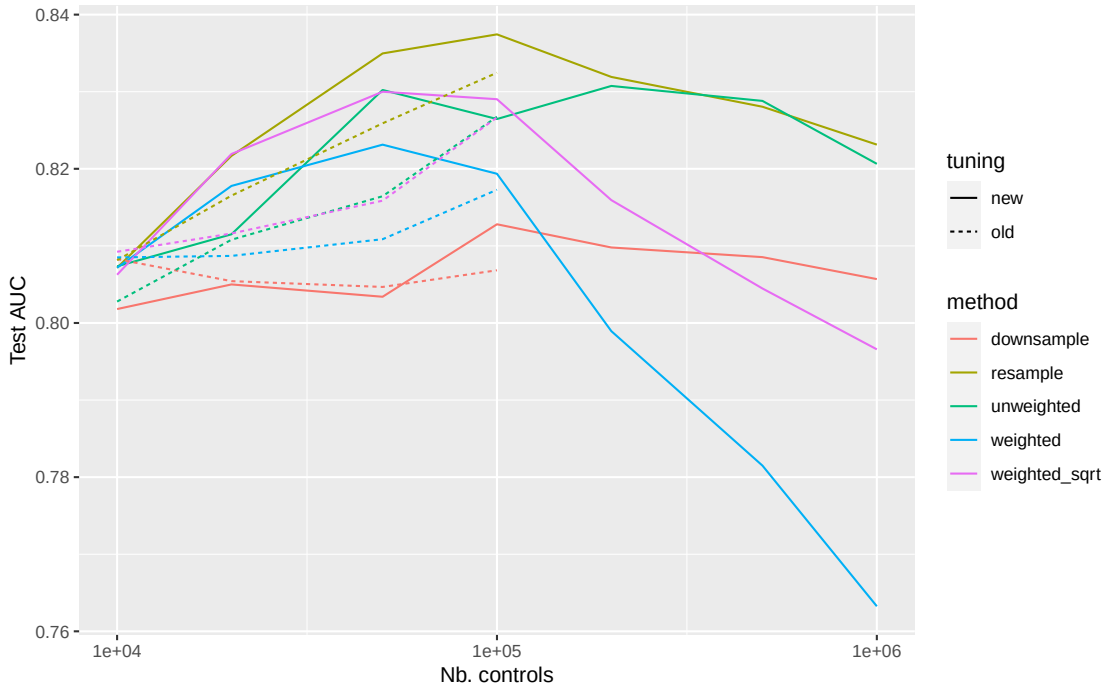
  weighted no resampling, cases upweighted by $n_{\text{controls}}/n_{\text{cases}}$

  weighted_sqrt no resampling, cases upweighted by $\sqrt{n_{\text{controls}}/n_{\text{cases}}}$

weighted_sq no resampling, cases upweighted by $(n_{\text{controls}}/n_{\text{cases}})^2$ (dropped from graph because much worse values and changed the scale too much to see others well)

- Plots:
  - y-axis: Test set AUC and Validation set AUC
  - x-axis: Nb. of controls before train/valid split
  - colors: which method is used to deal with imbalance
  - linetype: new vs. old tuning. The new tuning was performed manually with 100K controls; the old tuning with 10K controls. For more on tuning, see next section.
- Results and analysis:
  - Single repetition, so there is some variablity due to the sampling. To get a sense of the scale of the variability, the "downsample" method should be the same across the range. I might do multiple repititions in the future, but this gets very long for large N, so I am avoiding it for now.
  - The "resampling" method seems to be the best across the board
  - The "unweighted" method seems more stable?
  - Not much to gain beyond 50K-100K (decrease? within variability?). The best tuning parameters might not be the same for larger datasets, as is apparent between old/new tuning curves.




0.84
0.82
0.80
0.78
0.76
Test AUC
1e+04
1e+05
1e+06
Nb. controls
tuning
new
old
method
downsample
resample
unweighted
weighted
weighted_sqrt

Tuning

- Protocol:
  - Same as above, except:
    - Fix Nb. controls to $N = 100K$
    - Use "resample" method only
  - Fix all parameters to their default value stated above (and blotted with a vertical dashed line), vary one parameter along a plausible range
- Parameters:

  max_depth Control the complexity of the tree added at each iteration (deeper trees have more explanatory power, but can overfit)

  subsample Determines what percetage of the training set is used to fit the next tree (more subsampling prevents overfitting but slows down fitting and may lead to under-fitting)

  eta (step-size) controls how much the new tree contributes to the model (small step-size prevents overfitting, but slows down fitting)

- Results and analysis:

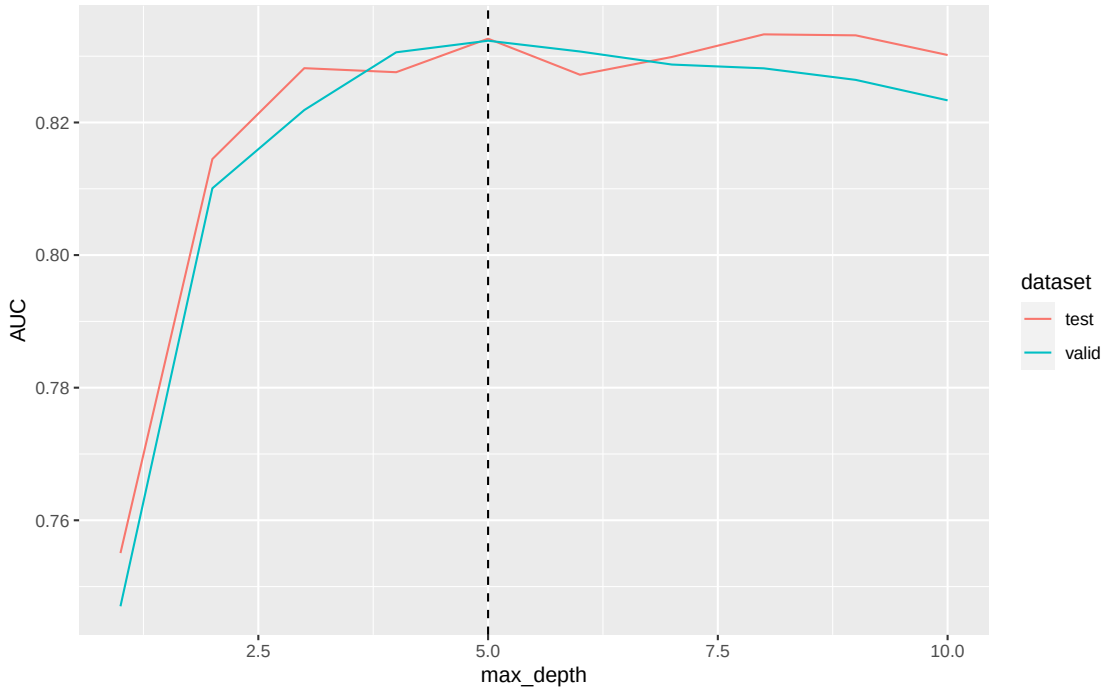
  max_depth Above depth 4-5, not much variation on valid/test AUCs; some improvment on train/cc AUCs

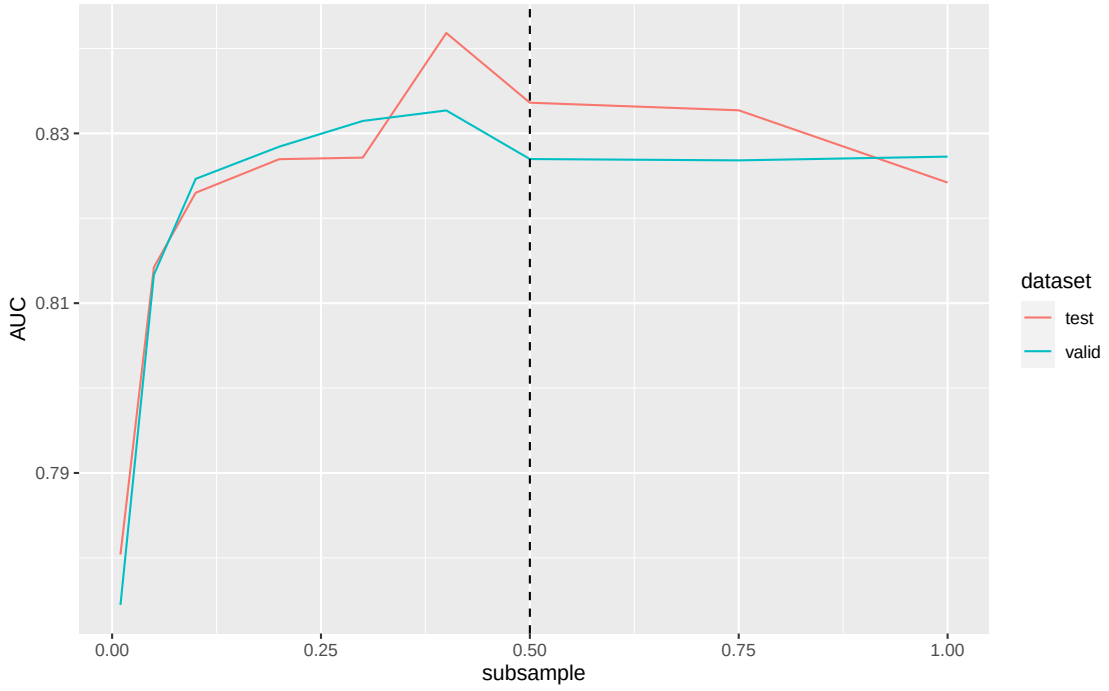
  subsample Except for small proportions, not much variation for all 4 datasets; best seems to be around 0.4-0.75

  eta waiting for results ...

Next steps

- Tuning for larger number of controls
- Alternative metrics


AUC
0.82
0.80
0.78
0.76
2.5
5.0
7.5
10.0
max_depth
dataset
test
valid


AUC
0.83
0.81
0.79
0.00
0.25
0.50
0.75
1.00
subsample
dataset
test
valid

11/16/2021 update

Tuning:

- AUROC, N=100K controls
- step-size: added results, see graph below
- for small eta, I didn't reach the stopping criterion, so take results with a grain of salt
- Seems like the "default" values I was using were mostly good
- Working on N=1M (*very slow...*)

Multiple tests sets:

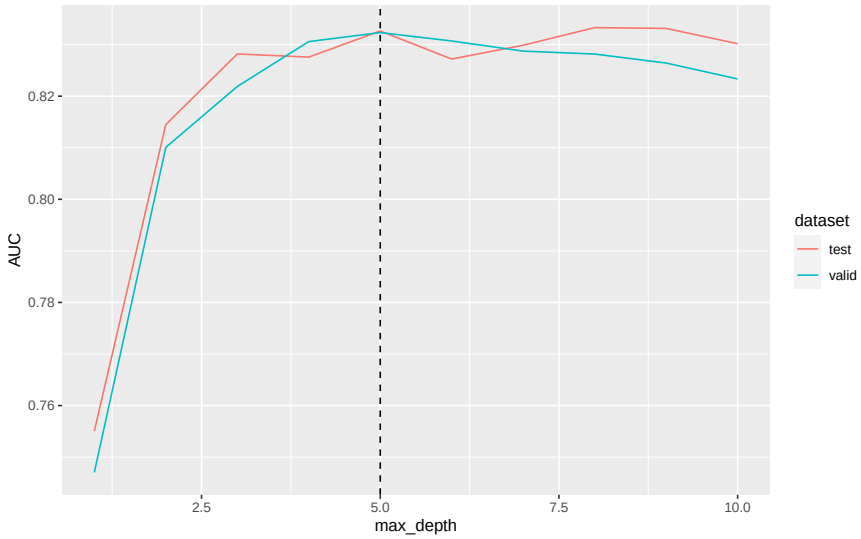
- All: original test set
- Complete cases: subset to only complete observations
- x-y% missing: subset to observations with (x,y]% of missing values
- AUROC graph below (resmapling cases):
  - Seems to stabilize beyond 100K
  - 10-30% missing values have much larger AUC
  - Complete cases is much more variable since smaller

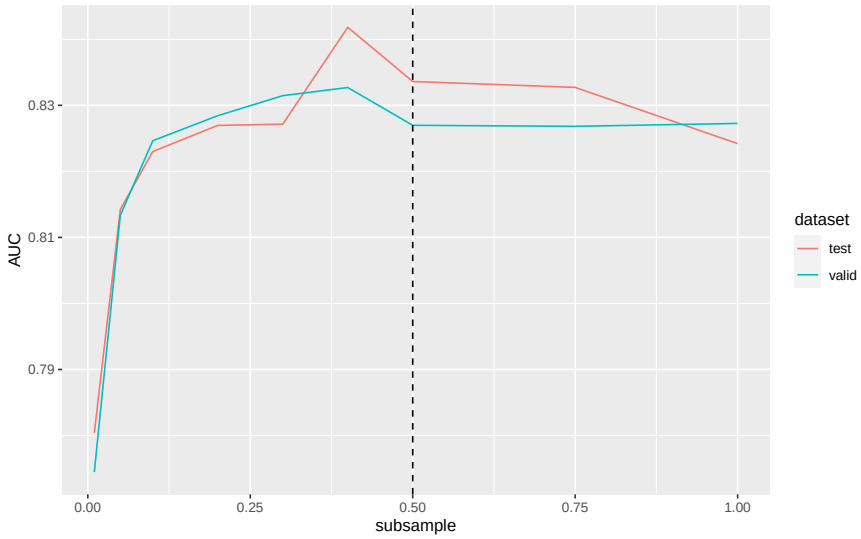
Test set	**Sample size**	**Nb. cases**	**% cases**
All	502849	2849	0.56%
Complete cases	1571	12	0.76%
0-5% missing	83640	286	0.34%
5-10% missing	133194	369	0.28%
10-30% missing	163377	1122	0.69%
30-100% missing	121067	1060	0.86%

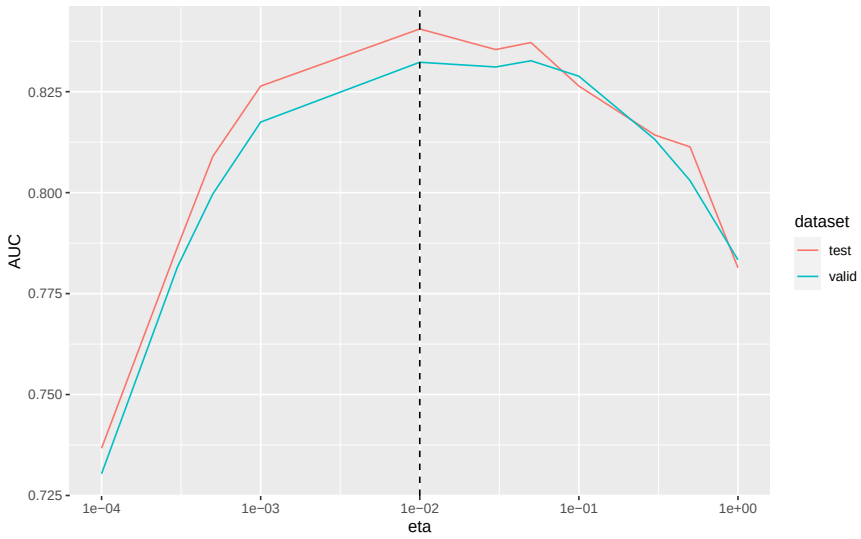
Metrics:

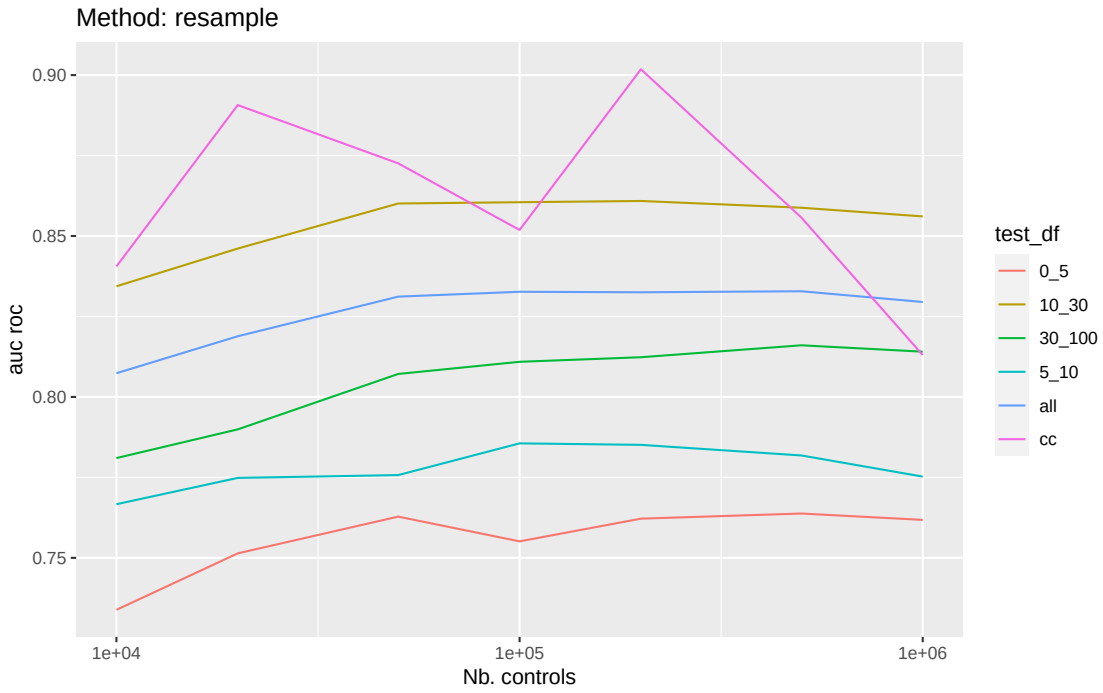
- AUC ROC: area under the ROC curve (sens, 1-spec)
- AUC PRC: area under the precision-recall curve (prec, rec)

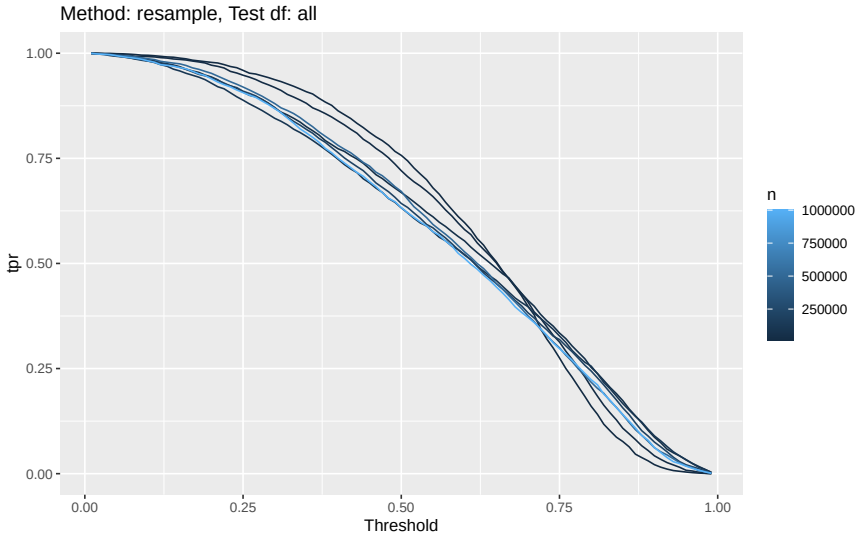
- TPR: true positive rate (*recall, sensitivity*, TP/P, "% of cases that would be tested")
- PPV: positive predictive value (*precision*, TP/DP, "% of cases among those tested"
- Detection prevalance: (DP/N, "% of patients tested")

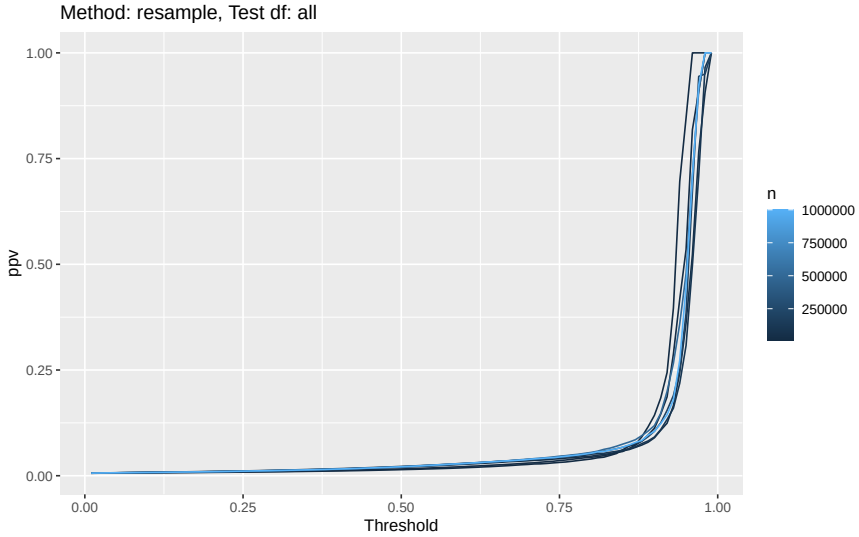

AUC
0.82
0.80
0.78
0.76
2.5
5.0
7.5
10.0
max_depth
dataset
test
valid

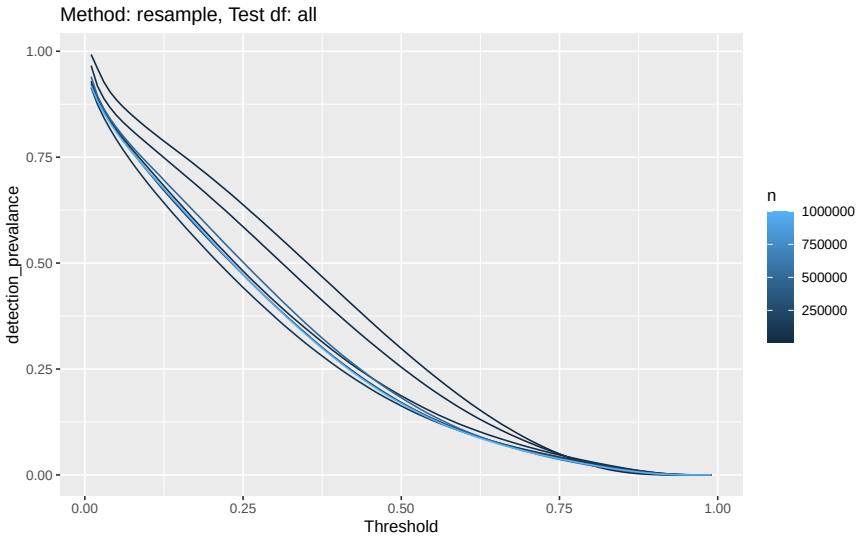

AUC
0.83
0.81
0.79
0.00
0.25
0.50
0.75
1.00
subsample
dataset
test
valid

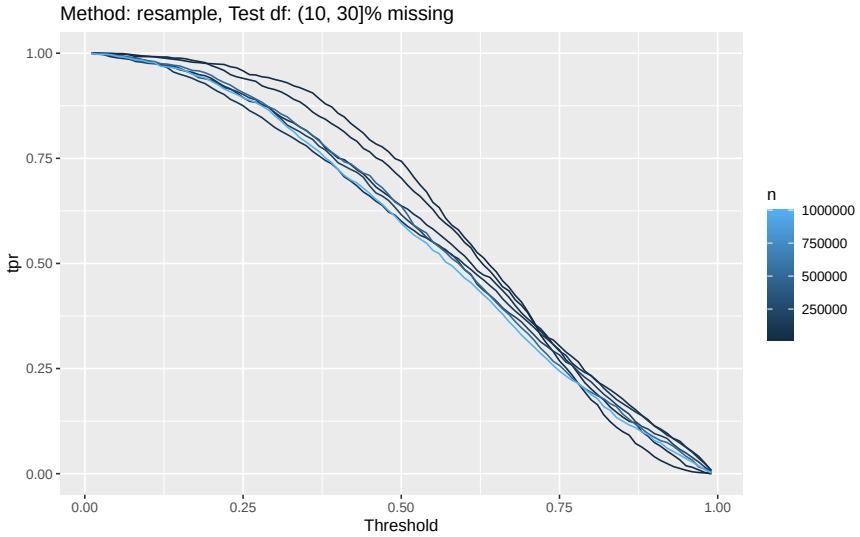

AUC
0.825
0.800
0.775
0.750
0.725
1e−04
1e−03
1e−02
1e−01
1e+00
eta
dataset
test
valid

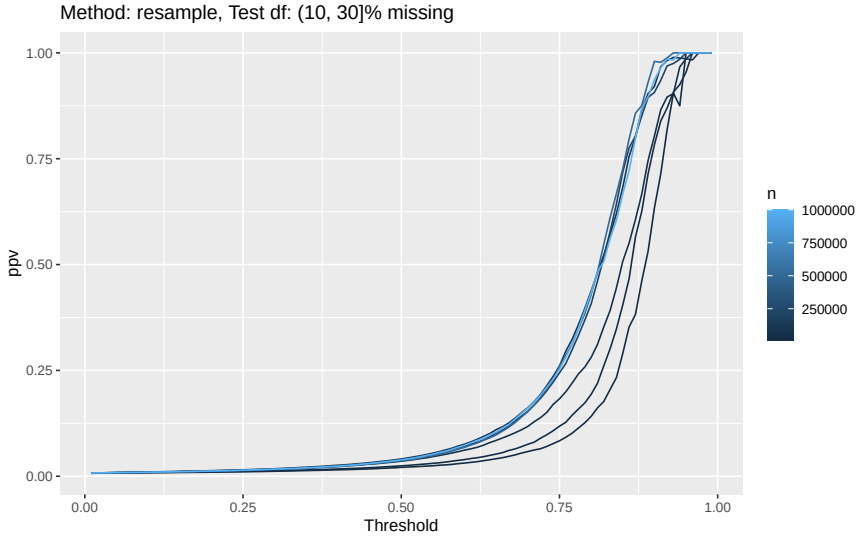

Method: resample
0.90
0.85
0.80
0.75
auc roc
1e+04
1e+05
1e+06
Nb. controls
test_df
0_5
10_30
30_100
5_10
all
cc

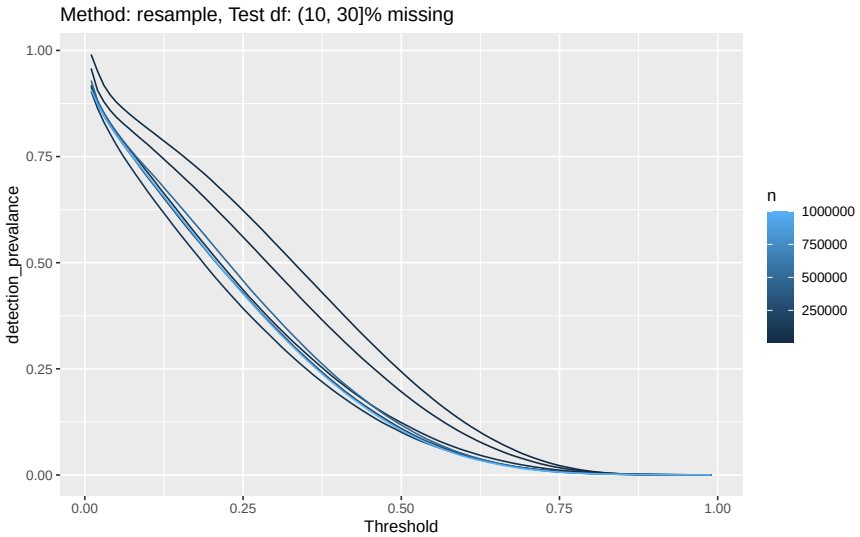

Method: resample, Test df: all
tpr
Threshold
n
1000000
750000
500000
250000


Method: resample, Test df: all
ppv
Threshold
n
1000000
750000
500000
250000


Method: resample, Test df: all
detection_prevalance
Threshold
n
1000000
750000
500000
250000


Method: resample, Test df: (10, 30]% missing
tpr
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
Threshold
n
1000000
750000
500000
250000


Method: resample, Test df: (10, 30]% missing
ppv
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
Threshold
n
1000000
750000
500000
250000


Method: resample, Test df: (10, 30]% missing
detection_prevalance
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
Threshold
n
1000000
750000
500000
250000

11/23/2021 update

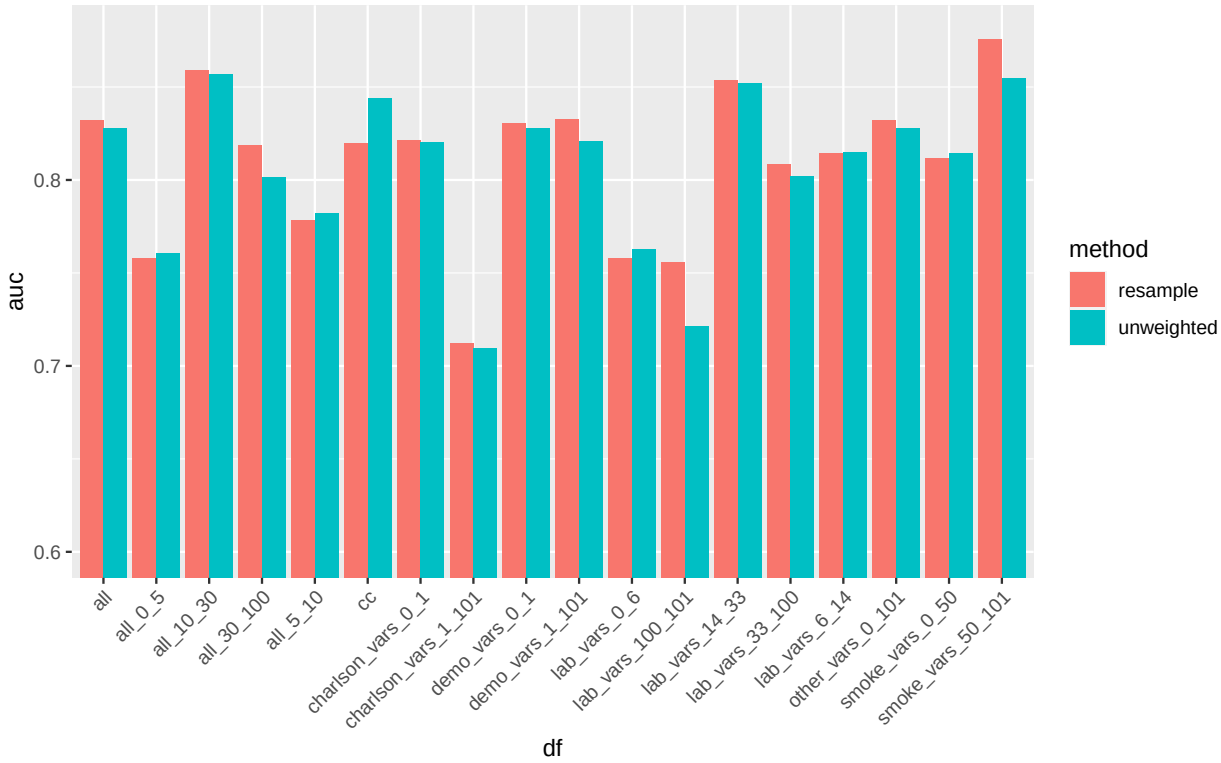
Missing values

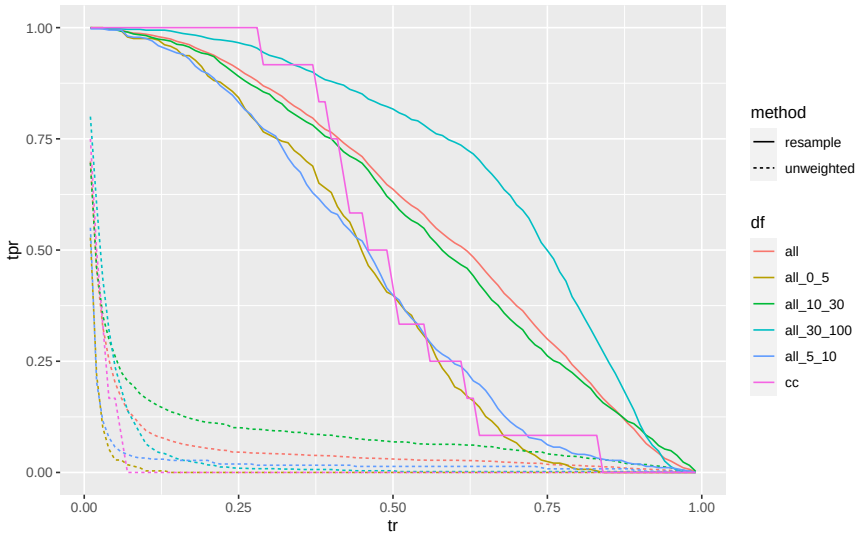
- Last week: split test set by proportion of missing values
- New: split test set by proportion of missing values within groups of variables
- Better for: 10-30% of missing values, 14-33% of missing lab variables, missing smoking variables
- Worse for: 0-10% of missing values, 1-100% missing charlson variables, 0-6 and 100% missing lab variables
- NB: high case % for missing charlson scores (but smaller total observations); higher case % for 100% missing lab variables

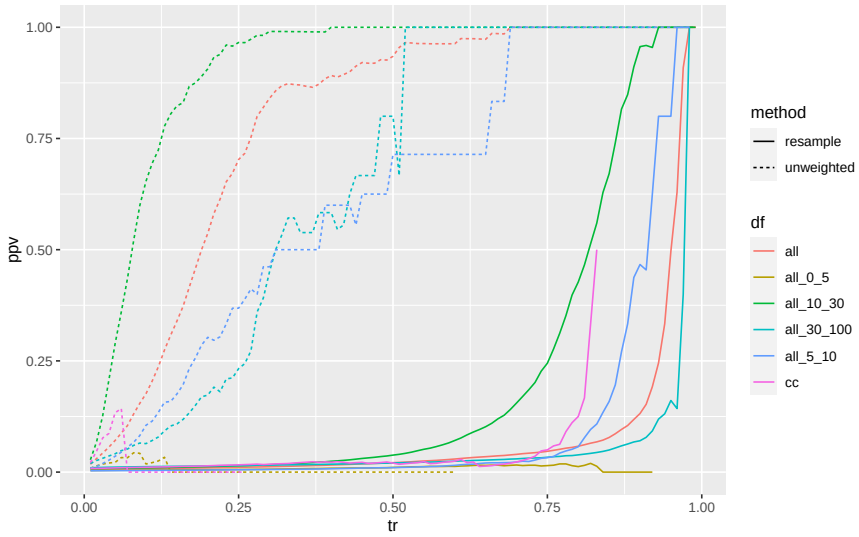
Calibration

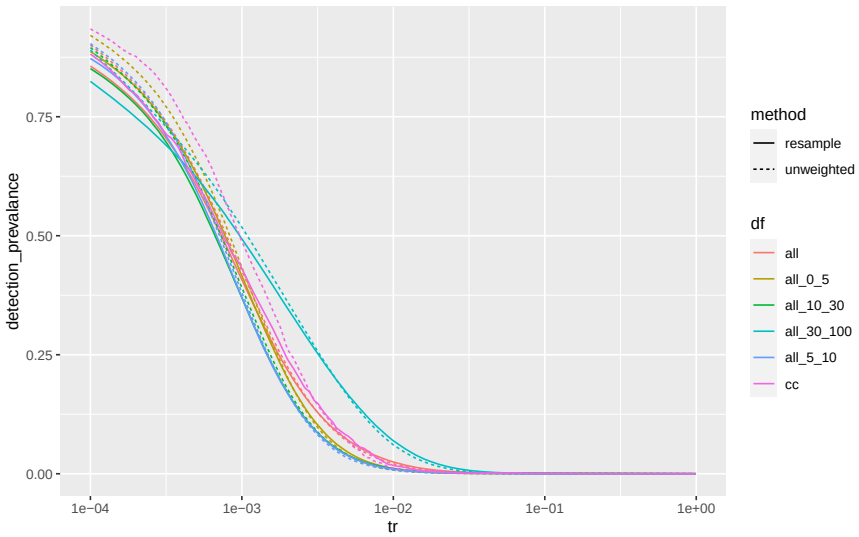
- Using the "resample" technique centers the scores around 50%, but the probabilities become uncalibrated (need to be careful with the interpretation)
- For "unweighted" the model is calibrated for the case proportion in the training set (0.5%)
- weight/resample to case incidence?
- Optimal threshold around 30-70% for "resample"; around 1-5% for "unweighted"
- Note that AUC only depends on the rank of the predicted probabilities, not on the probabilities themselves

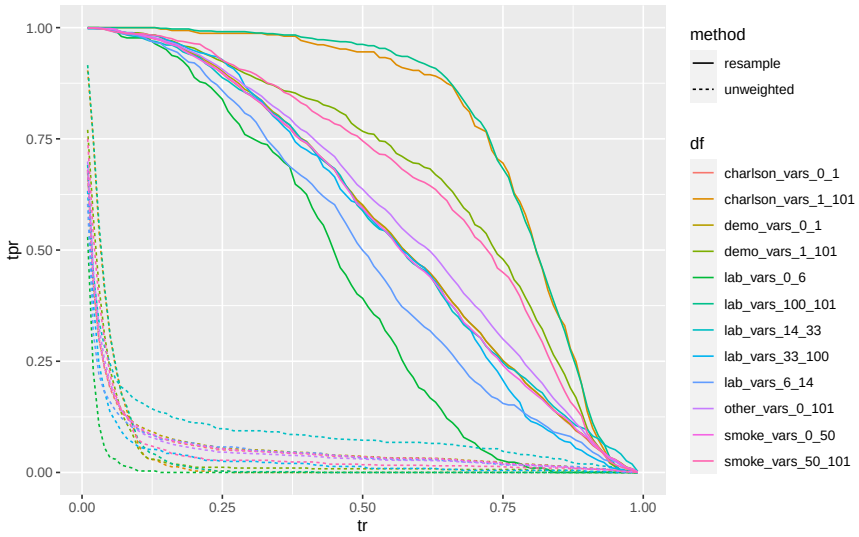
Test set	Sample size	Nb. cases	% cases
all	502849	2849	0.57%
cc	**1571**	12	0.76%
all_0_5	83640	286	0.34%
all_5_10	133193	369	0.28%
all_10_30	163373	1122	0.69%
all_30_100	121072	1060	0.88%
lab_vars_0_6	89372	304	0.34%
lab_vars_6_14	165220	593	0.36%
lab_vars_14_33	120373	843	0.70%
lab_vars_33_100	88946	444	0.50%
lab_vars_100_101	38938	665	**1.71%**
charlson_vars_0_1	495273	2537	0.51%
charlson_vars_1_101	**7576**	312	**4.12%**
demo_vars_0_1	416583	2244	0.54%
demo_vars_1_101	86266	605	0.70%
other_vars_0_101	502849	2849	0.57%
smoke_vars_0_50	284455	2044	0.72%
smoke_vars_50_101	218394	805	0.37%

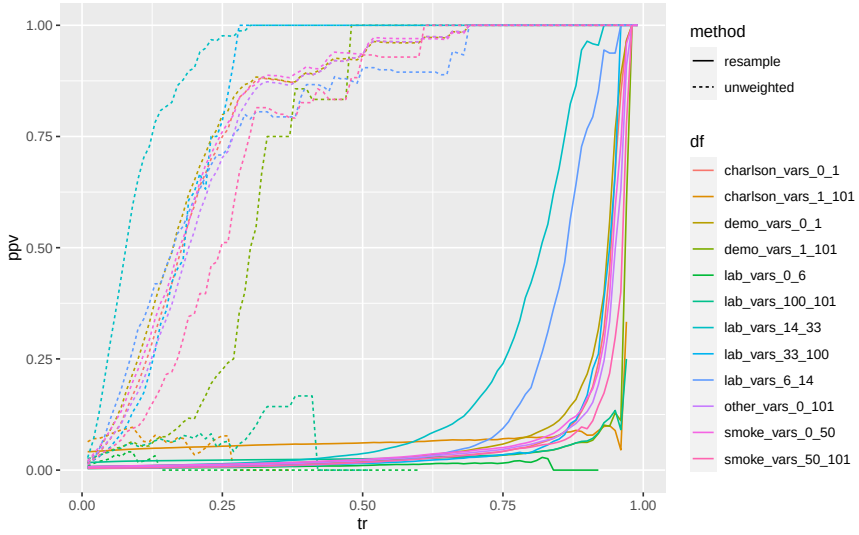


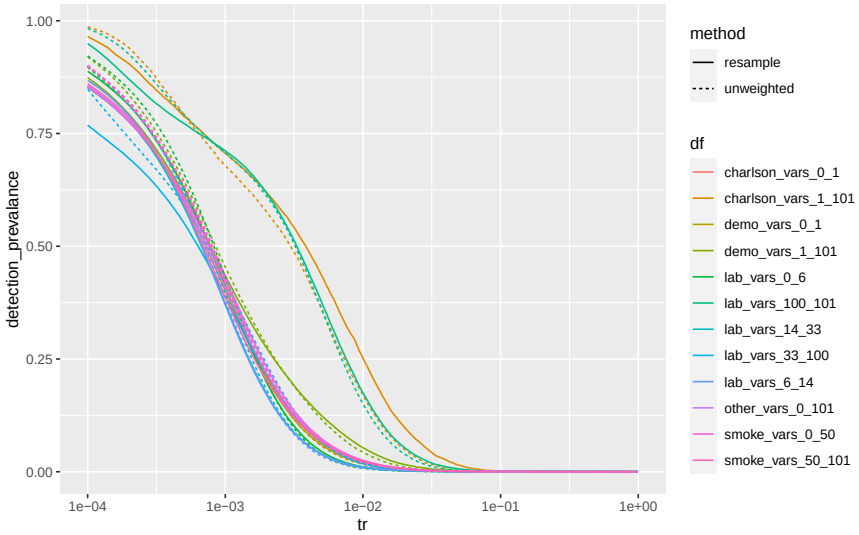

tpr
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
tr
method
resample
unweighted
df
all
all_0_5
all_10_30
all_30_100
all_5_10
cc


ppv
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
tr
method
resample
unweighted
df
all
all_0_5
all_10_30
all_30_100
all_5_10
cc

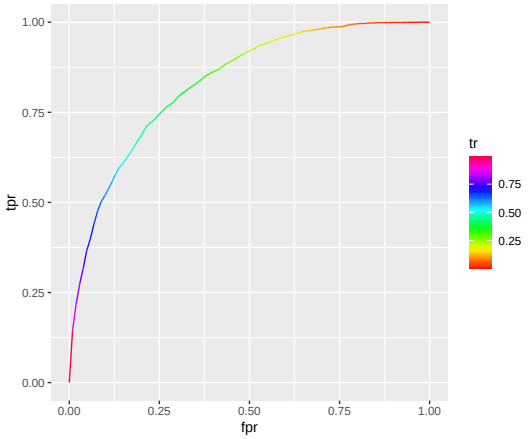

detection_prevalance
0.75
0.50
0.25
0.00
1e−04
1e−03
1e−02
1e−01
1e+00
tr
method
resample
unweighted
df
all
all_0_5
all_10_30
all_30_100
all_5_10
cc


tpr
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
tr
method
resample
unweighted
df
charlson_vars_0_1
charlson_vars_1_101
demo_vars_0_1
demo_vars_1_101
lab_vars_0_6
lab_vars_100_101
lab_vars_14_33
lab_vars_33_100
lab_vars_6_14
other_vars_0_101
smoke_vars_0_50
smoke_vars_50_101

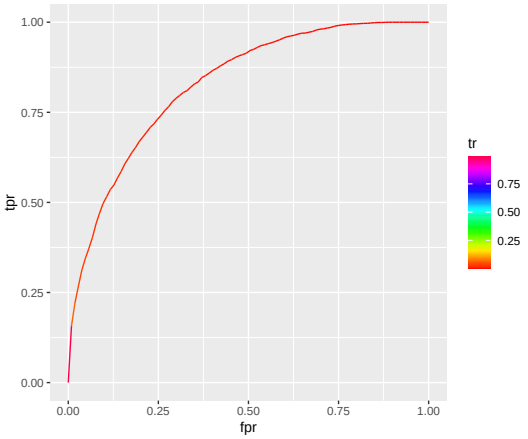

ppv
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
tr
method
resample
unweighted
df
charlson_vars_0_1
charlson_vars_1_101
demo_vars_0_1
demo_vars_1_101
lab_vars_0_6
lab_vars_100_101
lab_vars_14_33
lab_vars_33_100
lab_vars_6_14
other_vars_0_101
smoke_vars_0_50
smoke_vars_50_101


detection_prevalance
1.00
0.75
0.50
0.25
0.00
1e−04
1e−03
1e−02
1e−01
1e+00
tr
method
resample
unweighted
df
charlson_vars_0_1
charlson_vars_1_101
demo_vars_0_1
demo_vars_1_101
lab_vars_0_6
lab_vars_100_101
lab_vars_14_33
lab_vars_33_100
lab_vars_6_14
other_vars_0_101
smoke_vars_0_50
smoke_vars_50_101

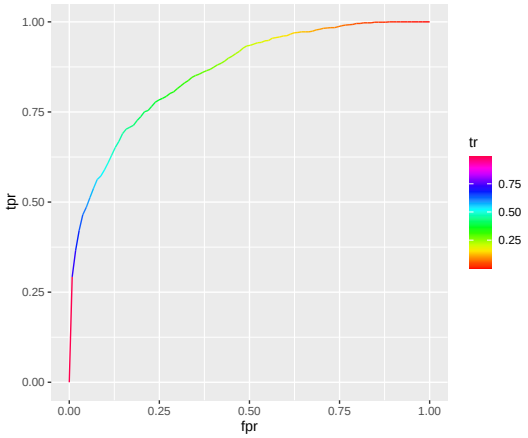
Df: all, method: resample, AUROC=0.8321



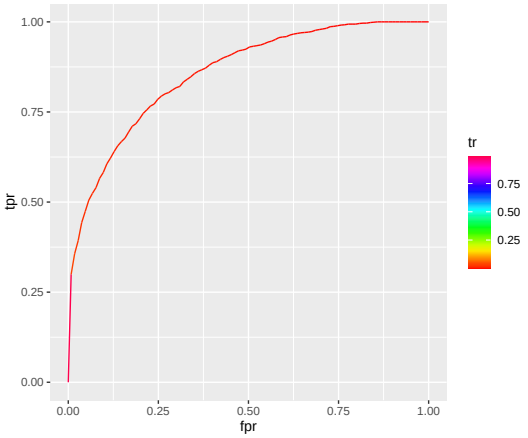
Df: all, method: unweighted, AUROC=0.8277



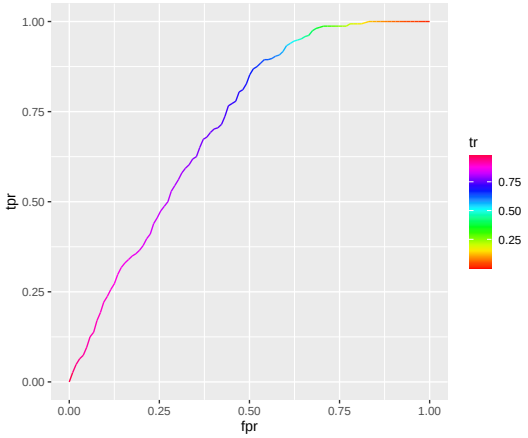
Df: all_10_30, method: resample, AUROC=0.8589



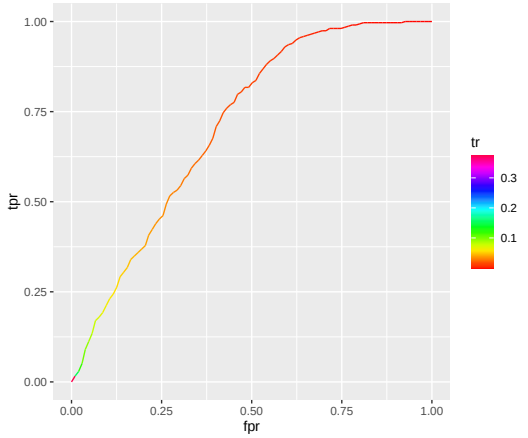
Df: all_10_30, method: unweighted, AUROC=0.857

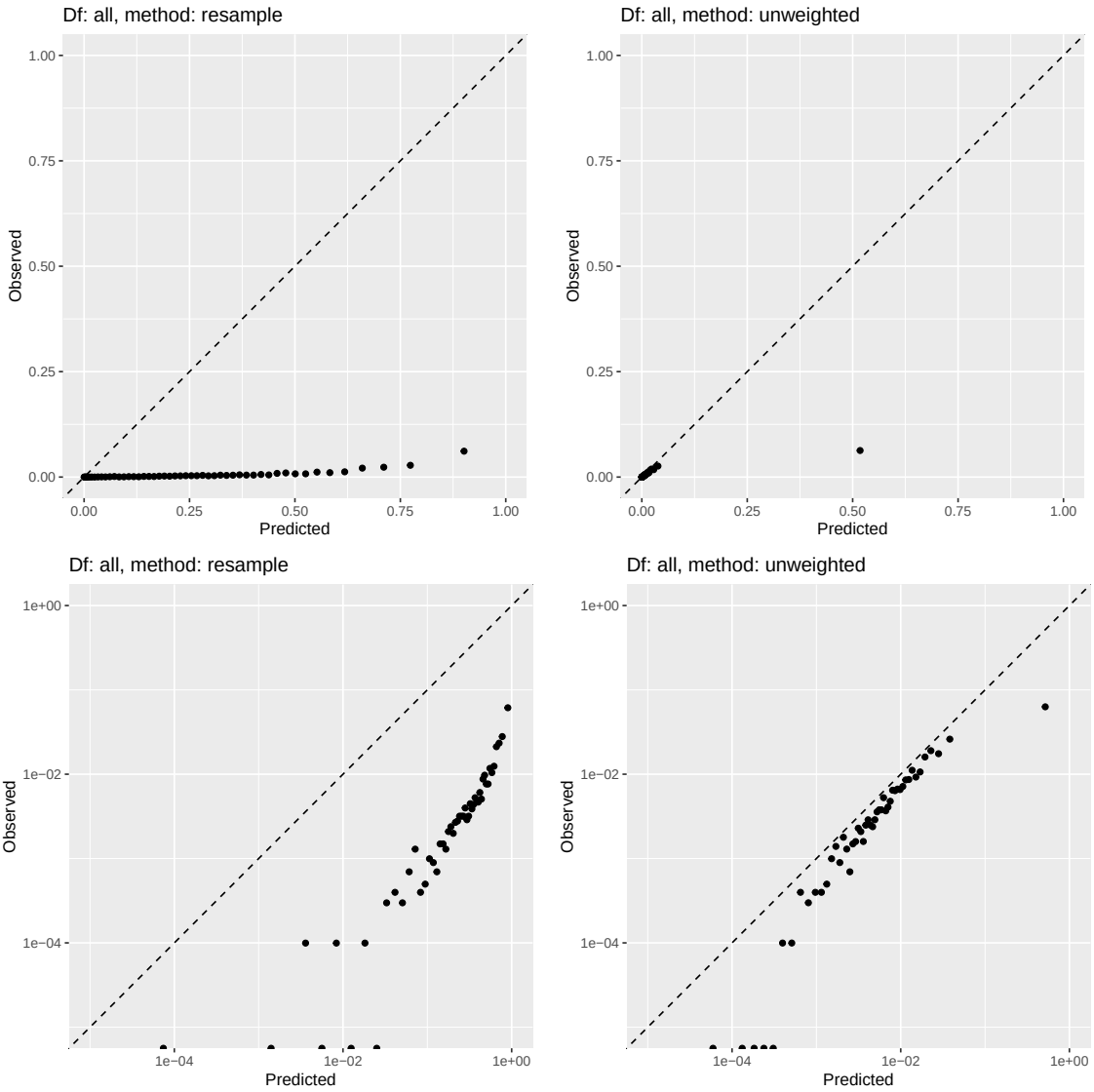


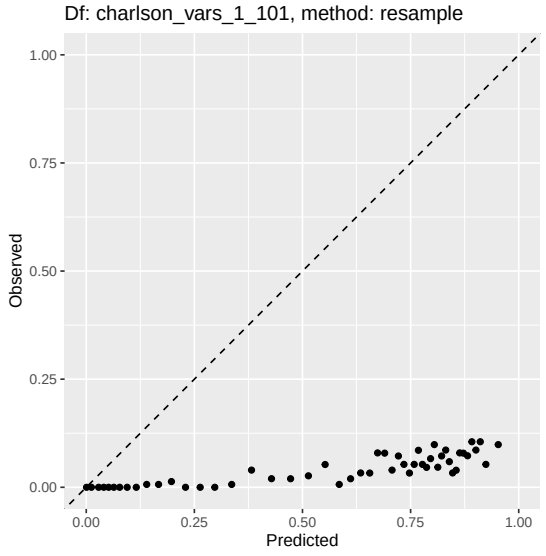
Df: charlson_vars_1_101, method: resample, AUROC=0.7123

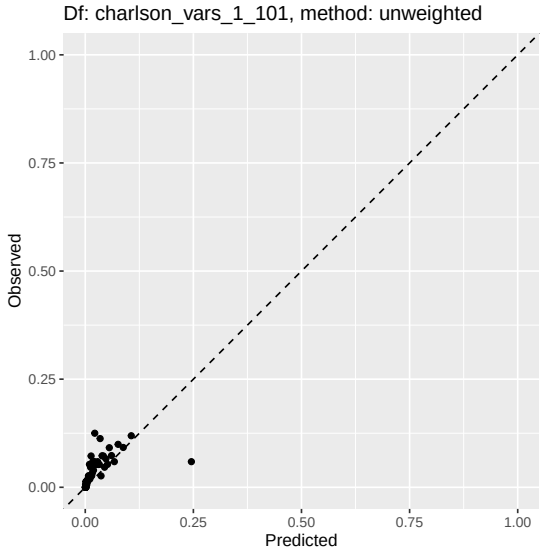


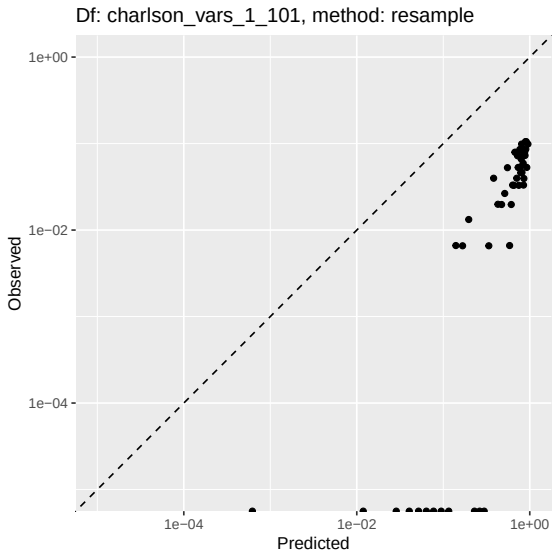
Df: charlson_vars_1_101, method: unweighted, AUROC=0.7099

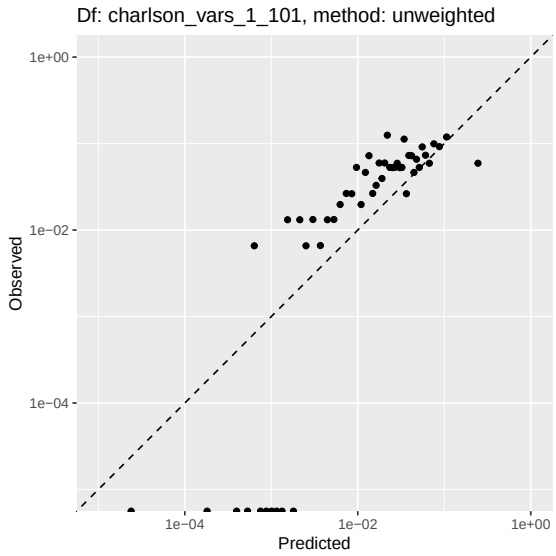



Df: all, method: resample
Df: all, method: unweighted
Df: all, method: resample
Df: all, method: unweighted
Observed
Predicted
1.00
0.75
0.50
0.25
0.00
1e+00
1e−02
1e−04


Df: charlson_vars_1_101, method: resample
Observed
Predicted
0.00
0.25
0.50
0.75
1.00


Df: charlson_vars_1_101, method: unweighted
Observed
Predicted
0.00
0.25
0.50
0.75
1.00

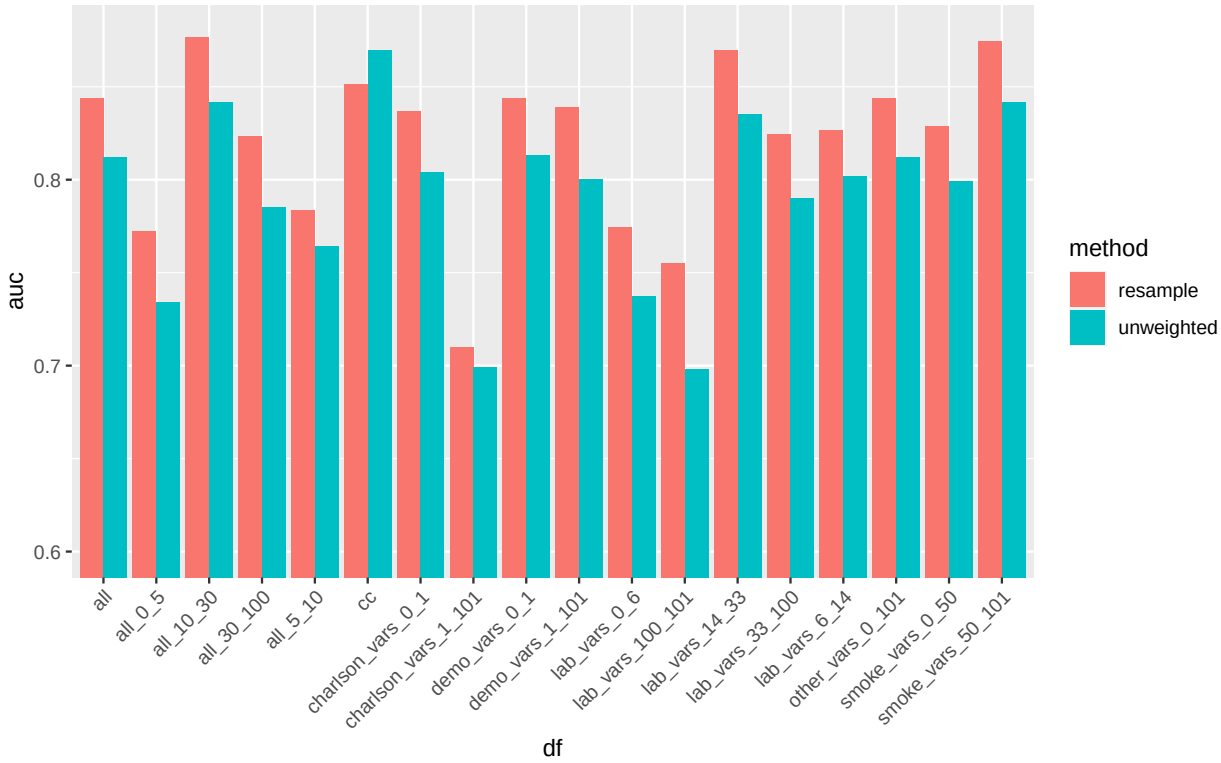

Df: charlson_vars_1_101, method: resample
Observed
Predicted
1e+00
1e−02
1e−04

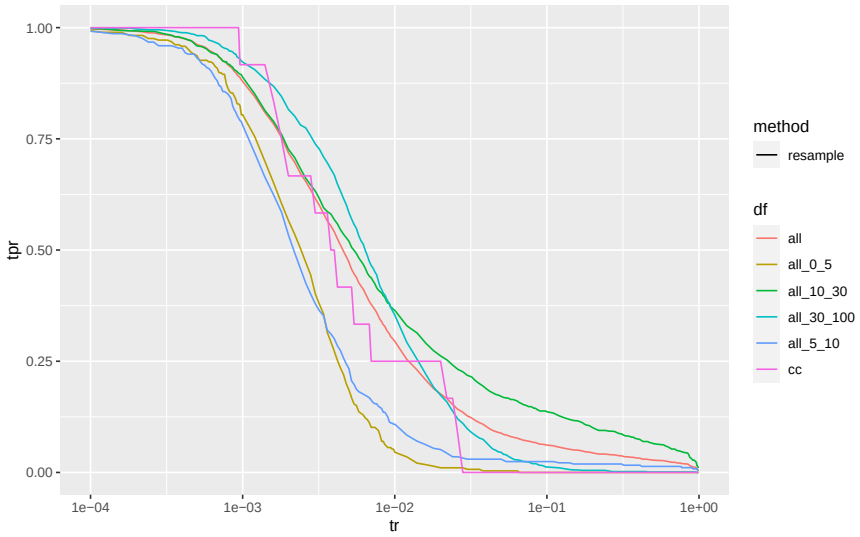

Df: charlson_vars_1_101, method: unweighted
Observed
Predicted
1e+00
1e−02
1e−04

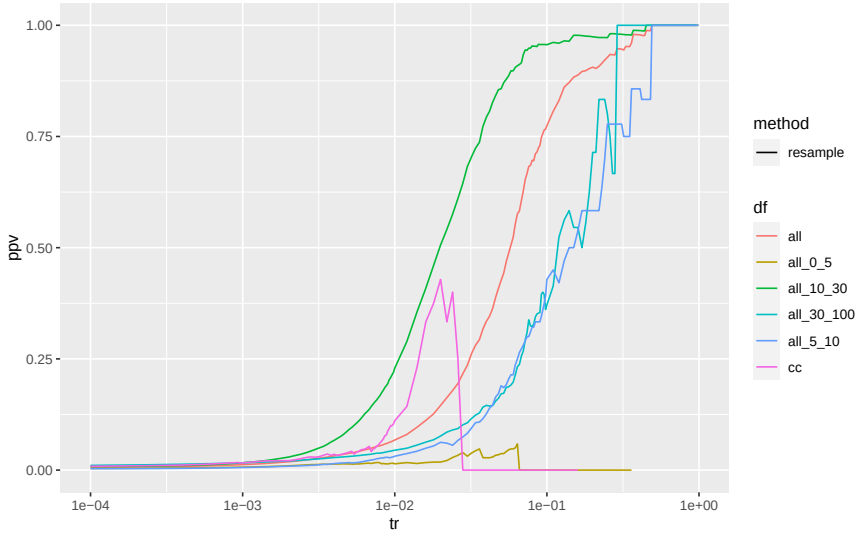
11/30/2021 update

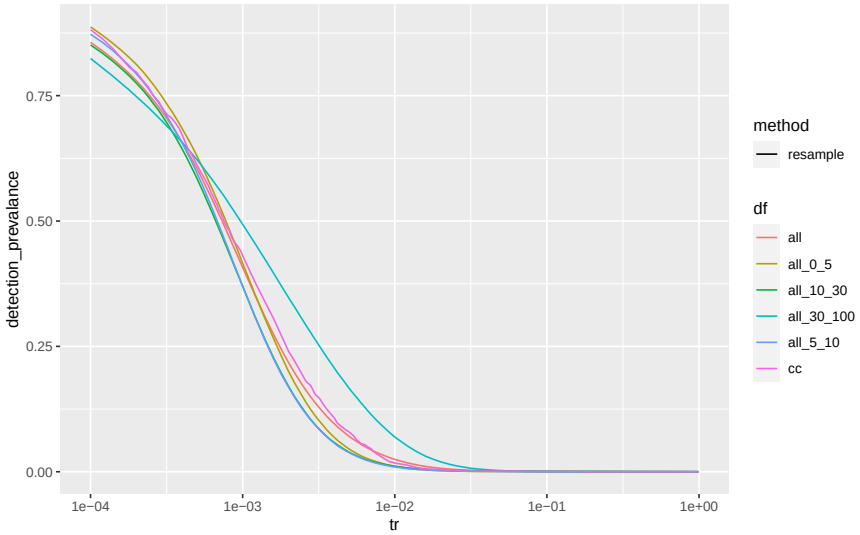
Recalibration

- "resample": upsample cases up to a balanced training set; downweight cases so they have the same total weight as the original dataset
  - Example
  - original training set: 1M controls + 5K cases
  - after resampling: 1M controls + 1M cases
  - reweighing: controls(1M, 1) and cases(1M, 11K/6M)
- "unweighted": reweight cases to have the same weight as in the original dataset
  - reweighing: controls(1M, 1) and cases(5K, 1M*11K/5K/6M)
- Same simulations as last week but with twice as many controls
- Note that the fitted models are calibrated for the 11K/6M proportion but that is not what the testing sets are (2.5K/500K), so we expect some miscalibration on the test sets (there's a factor of ~ 3 between the two)

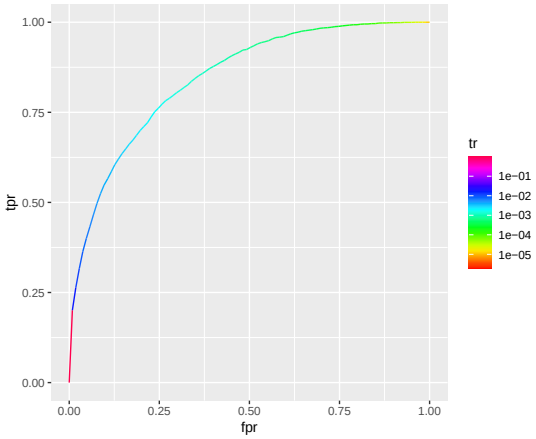



1.00
0.75
0.50
0.25
0.00
tpr
1e−04
1e−03
1e−02
1e−01
1e+00
tr
method
resample
df
all
all_0_5
all_10_30
all_30_100
all_5_10
cc

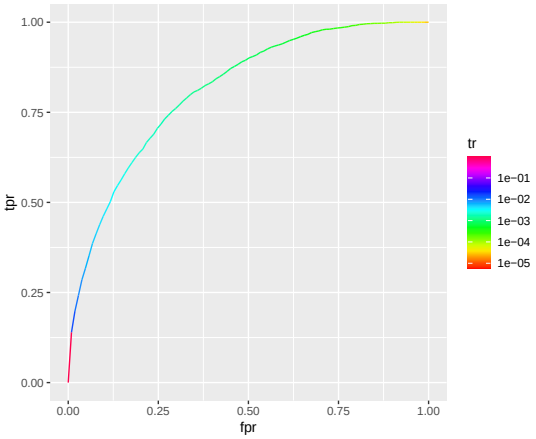

1.00
0.75
0.50
0.25
0.00
ppv
1e−04
1e−03
1e−02
1e−01
1e+00
tr
method
resample
df
all
all_0_5
all_10_30
all_30_100
all_5_10
cc


0.75
0.50
0.25
0.00
detection_prevalance
1e−04
1e−03
1e−02
1e−01
1e+00
tr
method
resample
df
all
all_0_5
all_10_30
all_30_100
all_5_10
cc

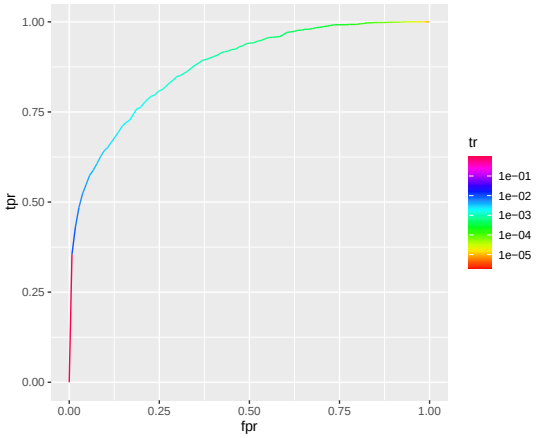
Df: all, method: resample, AUROC=0.8439



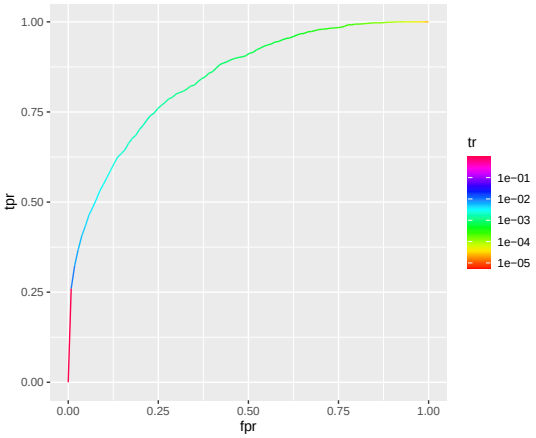
Df: all, method: unweighted, AUROC=0.8118



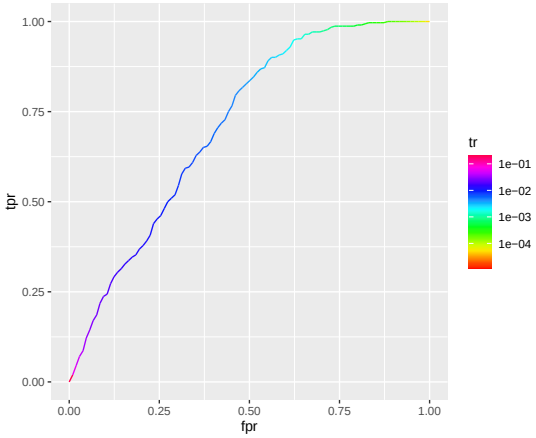
Df: all_10_30, method: resample, AUROC=0.8765



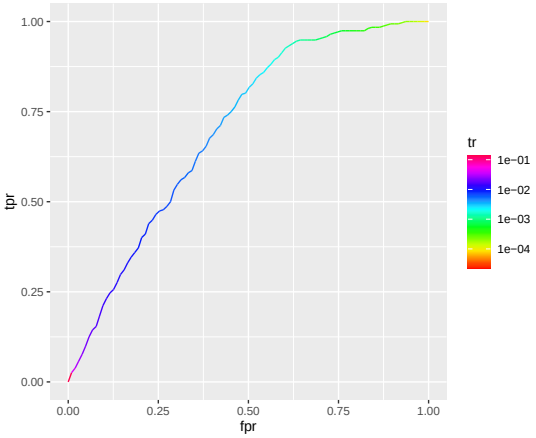
Df: all_10_30, method: unweighted, AUROC=0.8417

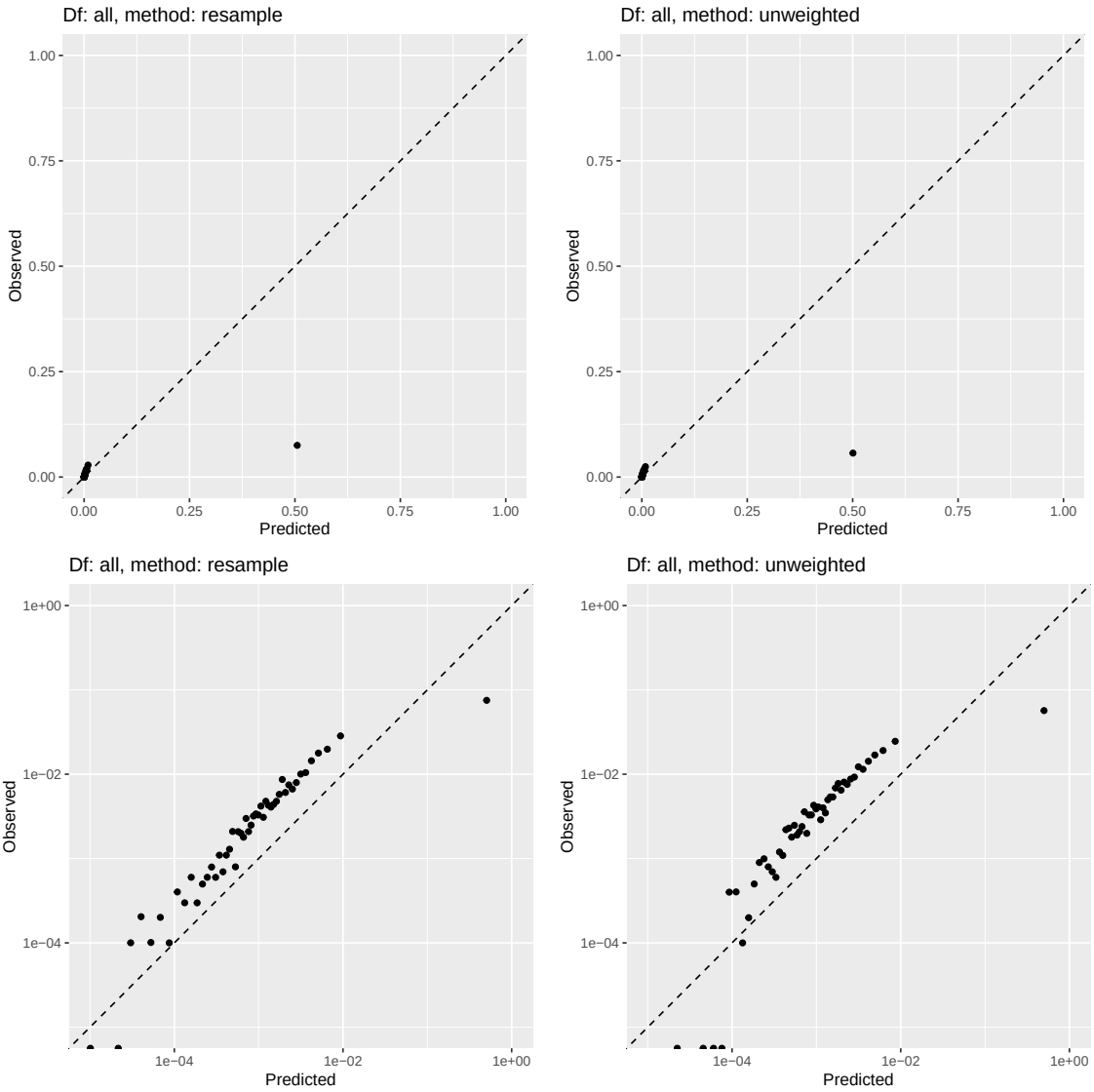


Df: charlson_vars_1_101, method: resample, AUROC=0.7097



Df: charlson_vars_1_101, method: unweighted, AUROC=0.6994

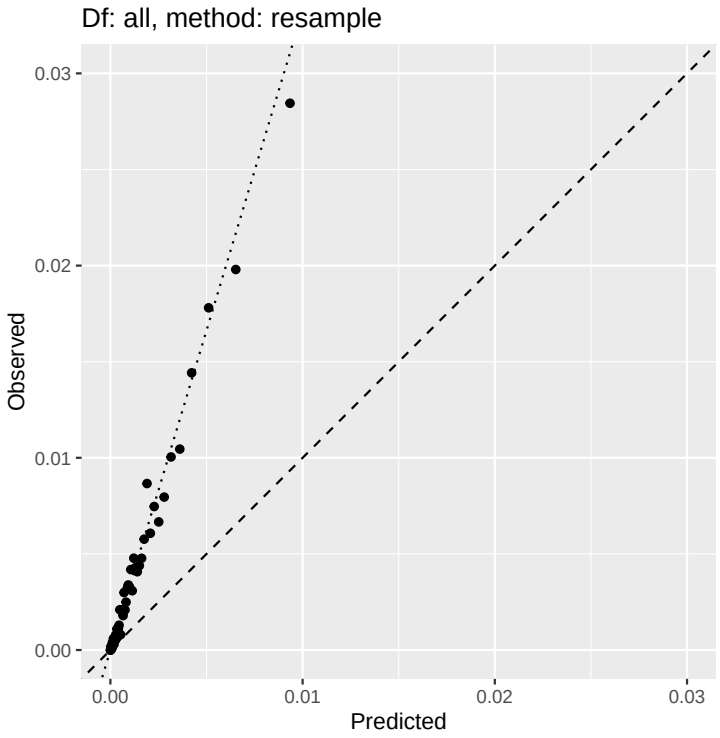



Df: all, method: resample
Df: all, method: unweighted
Df: all, method: resample
Df: all, method: unweighted
Observed
Predicted
1.00
0.75
0.50
0.25
0.00
1e+00
1e−02
1e−04

12/07/2021 update

More on calibration

- Inspection of the miscalibration observed last week showed that it is due to the miscalibration in the testing set.
- See plot below. The dottod line has slope 6.6M/2M which is exactly the ratio between the testing set and the calibration



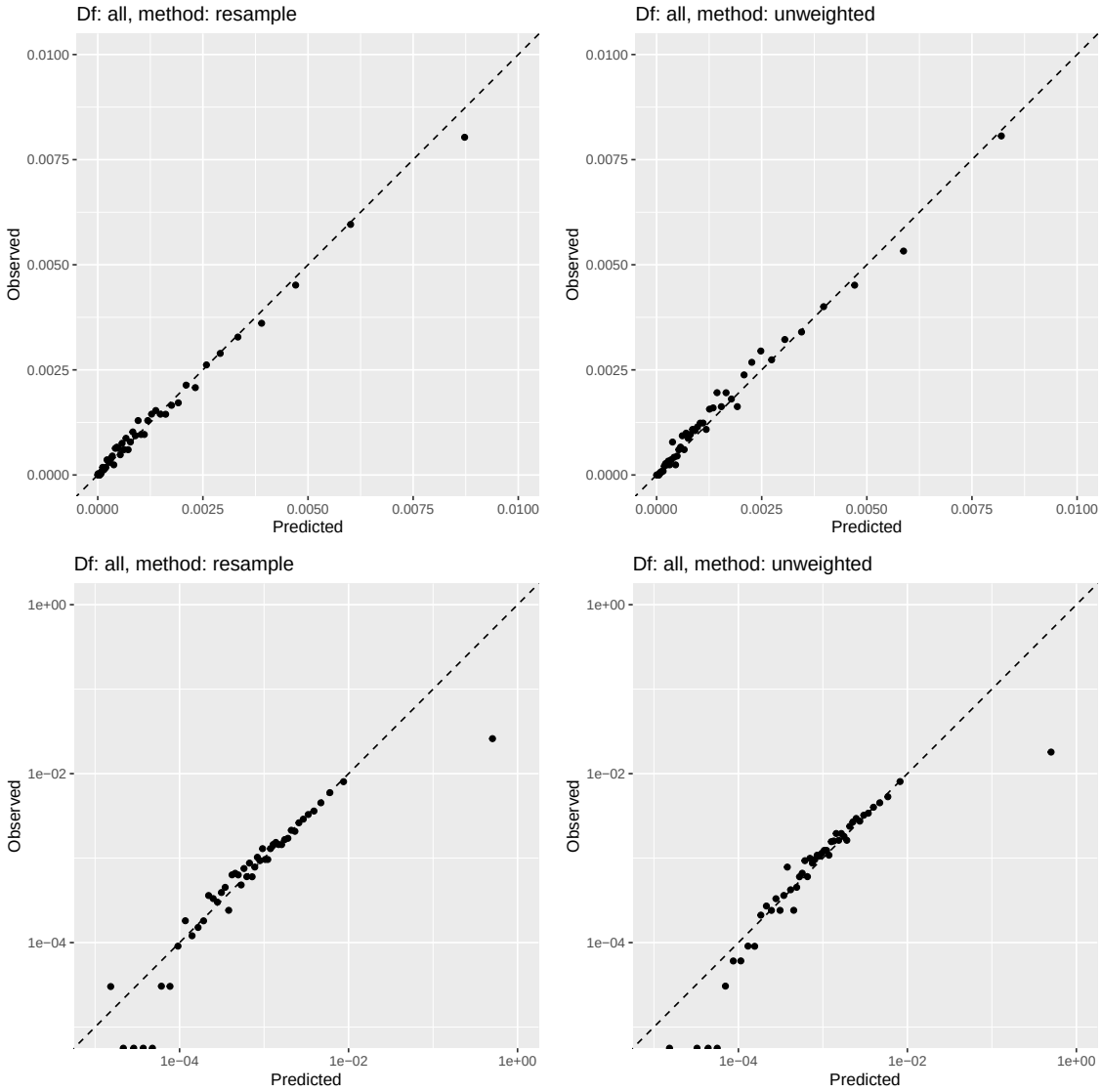
- Training on the whole data set is currently running: we should get correct calibration
- Below is the calibration table depending on the threshold

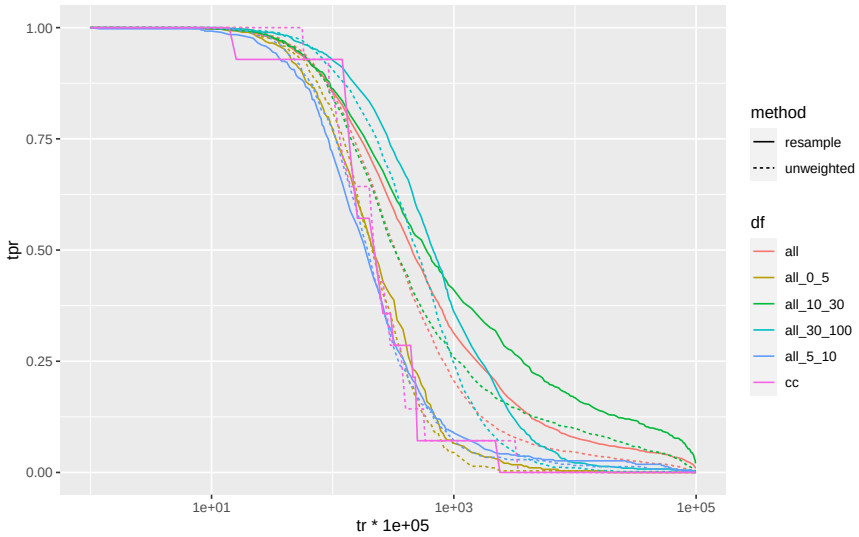
Threshold in /100000, values in %, using miscalibrated model (6.6M/2M), rule of thumb: multiply threshold by 3.2

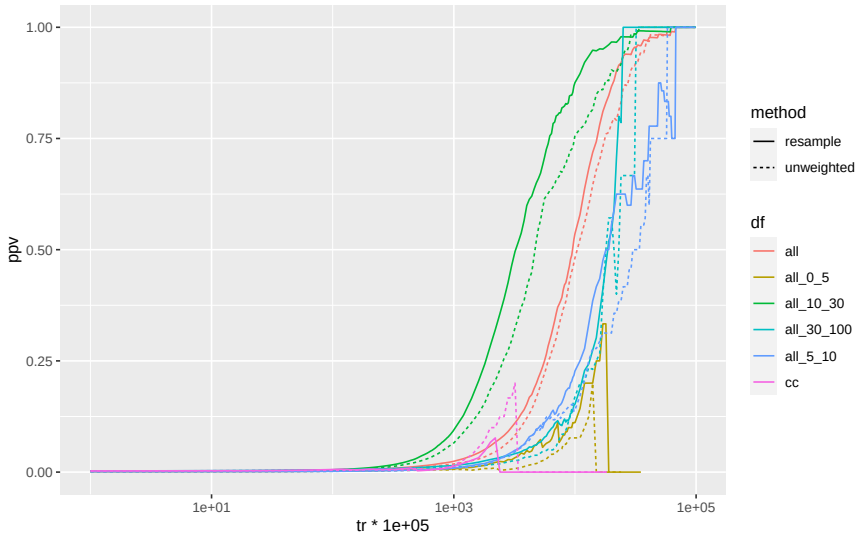
Threshold	TPR	PPV	Det. prev.
10	99.72	0.66	85.64
12	99.61	0.67	83.93
14	99.51	0.68	82.33
16	99.44	0.70	80.80
18	99.30	0.71	79.38
20	99.19	0.72	77.98
22	99.05	0.73	76.64
24	98.84	0.74	75.31
26	98.84	0.76	73.99
28	98.60	0.77	72.76
30	98.46	0.78	71.48
32	98.35	0.79	70.24
34	98.21	0.81	69.07
36	97.93	0.82	67.92
38	97.82	0.83	66.80
40	97.65	0.84	65.69
42	97.44	0.85	64.60
44	97.19	0.87	63.53
46	96.98	0.88	62.50
48	96.67	0.89	61.47
50	96.21	0.90	60.47
52	96.00	0.91	59.46
54	95.86	0.93	58.50
56	95.68	0.94	57.51
58	95.37	0.96	56.55
60	94.91	0.97	55.62
62	94.66	0.98	54.73
64	94.38	0.99	53.85
66	94.10	1.01	52.98
68	93.86	1.02	52.12
70	93.33	1.03	51.31
72	92.98	1.04	50.50
74	92.49	1.05	49.71
76	92.35	1.07	48.92
78	92.07	1.08	48.17
80	91.65	1.10	47.41
82	91.33	1.11	46.66
84	91.08	1.12	45.94
86	90.70	1.14	45.24
88	90.17	1.15	44.56
90	89.93	1.16	43.90
92	89.36	1.17	43.23
94	89.08	1.18	42.61
96	88.77	1.20	41.96
98	88.31	1.21	41.34
100	87.89	1.22	40.75
120	84.49	1.35	35.34
140	80.94	1.48	30.94
160	78.31	1.62	27.34
180	75.36	1.76	24.32
200	71.85	1.87	21.81
220	69.71	2.01	19.68
240	67.22	2.13	17.87
260	65.39	2.27	16.29
280	63.53	2.41	14.93
300	61.60	2.54	13.74
320	59.99	2.67	12.71
340	58.23	2.80	11.79
360	56.55	2.93	10.95
380	55.46	3.07	10.22
400	54.12	3.20	9.57
420	52.51	3.32	8.95
440	51.07	3.43	8.43
460	49.74	3.54	7.96
480	48.51	3.66	7.51
500	47.24	3.78	7.08
600	41.87	4.33	5.48
700	37.77	4.91	4.35
800	34.40	5.48	3.56
900	31.59	6.07	2.95

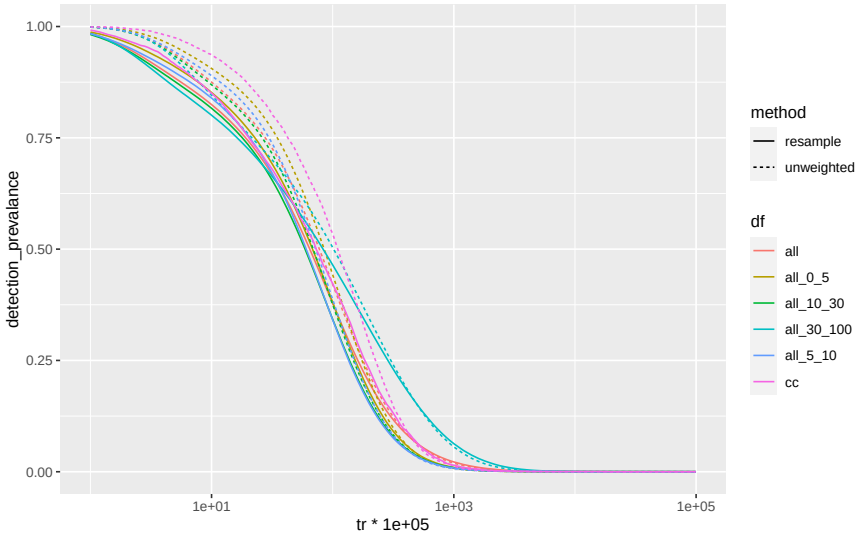
12/14/2021 update

- Follow-up on last week:
  - We were unsure why the miscalibration was in the observed direction
  - Solution: we needed to remember that the testing sets contained 3x too many cases compared to the training set
  - This explains why the observed proportion was 3x higher than expected
  - The fitted probabilities were correct, but the case incidence in the testing set was incorrect
- Results using all 6M controls are in!
- Essentially the same as before with slightly improved AUC (0.8459 vs 0.8439); calibration in the testing set is improved
- We have the same proportion on all sets (training, validation, testing) which makes the model calibrated to the correct incidence and the evaluation uses the correct incidence as well.
- Threshold selection: similar evaluation metrics except ppv/3


Df: all, method: resample
Df: all, method: unweighted
Df: all, method: resample
Df: all, method: unweighted
Observed
Predicted
0.0000
0.0025
0.0050
0.0075
0.0100
1e−04
1e−02
1e+00

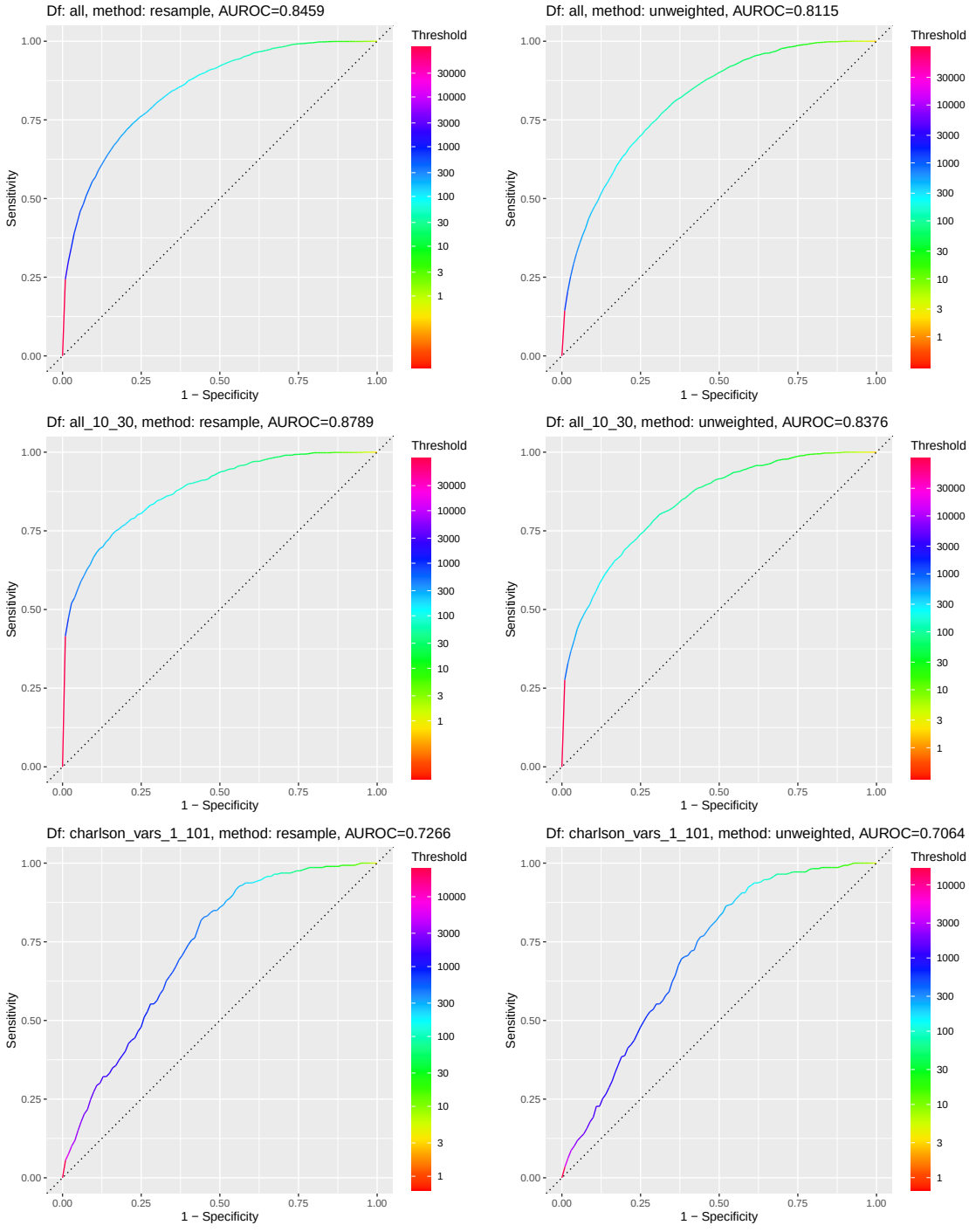

1.00
0.75
0.50
0.25
0.00
tpr
1e+01
1e+03
1e+05
tr * 1e+05
method
resample
unweighted
df
all
all_0_5
all_10_30
all_30_100
all_5_10
cc

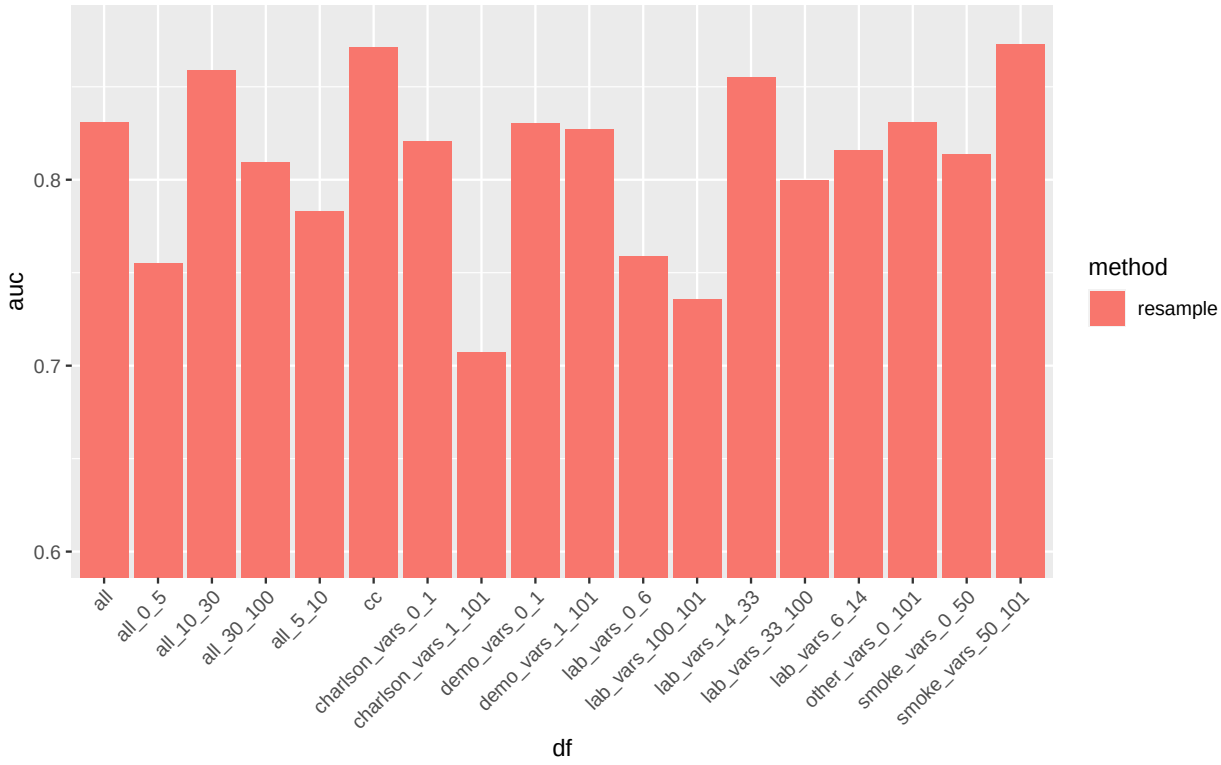

1.00
0.75
0.50
0.25
0.00
ppv
1e+01
1e+03
1e+05
tr * 1e+05
method
resample
unweighted
df
all
all_0_5
all_10_30
all_30_100
all_5_10
cc


1.00
0.75
0.50
0.25
0.00
detection_prevalance
1e+01
1e+03
1e+05
tr * 1e+05
method
resample
unweighted
df
all
all_0_5
all_10_30
all_30_100
all_5_10
cc

Threshold	TPR	PPV	Det. prev.
10	99.79	0.21	82.47
12	99.68	0.21	80.65
14	99.51	0.22	78.94
16	99.33	0.22	77.34
18	99.23	0.22	75.82
20	99.12	0.23	74.37
22	98.91	0.23	72.96
24	98.56	0.24	71.60
26	98.28	0.24	70.27
28	98.07	0.24	68.99
30	97.86	0.25	67.72
32	97.61	0.25	66.51
34	97.30	0.26	65.33
36	96.91	0.26	64.16
38	96.77	0.26	63.03
40	96.60	0.27	61.91
42	96.28	0.27	60.83
44	95.75	0.27	59.77
46	95.37	0.28	58.72
48	95.05	0.28	57.69
50	94.49	0.29	56.70
52	94.28	0.29	55.71
54	94.03	0.29	54.76
56	93.72	0.30	53.81
58	93.37	0.30	52.91
60	92.91	0.31	52.00
62	92.56	0.31	51.14
64	92.31	0.31	50.28
66	91.79	0.32	49.45
68	91.30	0.32	48.64
70	91.19	0.33	47.84
72	90.91	0.33	47.06
74	90.59	0.34	46.30
76	90.14	0.34	45.56
78	89.79	0.34	44.82

Threshold	TPR	PPV	Det. prev.
80	89.61	0.35	44.11
82	89.26	0.35	43.41
84	88.94	0.36	42.73
86	88.49	0.36	42.06
88	88.03	0.36	41.41
90	87.71	0.37	40.77
92	87.43	0.37	40.14
94	87.05	0.38	39.54
96	86.38	0.38	38.95
98	86.00	0.38	38.37
100	85.68	0.39	37.80
120	82.59	0.43	32.72
140	78.98	0.47	28.58
160	76.20	0.52	25.18
180	73.82	0.57	22.37
200	71.29	0.61	20.00
220	68.87	0.66	18.01
240	66.80	0.70	16.30
260	64.55	0.75	14.84
280	62.48	0.79	13.57
300	60.62	0.83	12.46
320	58.97	0.88	11.50
340	57.04	0.92	10.65
360	56.02	0.97	9.89
380	54.62	1.02	9.21
400	53.32	1.06	8.61
500	47.31	1.28	6.33
600	42.86	1.51	4.87
700	39.24	1.74	3.86
800	36.57	2.00	3.14
900	33.17	2.18	2.61
1000	31.27	2.43	2.20
2000	21.83	5.87	0.64
5000	11.51	21.28	0.09
10000	7.83	53.61	0.03


Df: all, method: resample, AUROC=0.8459
Df: all, method: unweighted, AUROC=0.8115
Df: all_10_30, method: resample, AUROC=0.8789
Df: all_10_30, method: unweighted, AUROC=0.8376
Df: charlson_vars_1_101, method: resample, AUROC=0.7266
Df: charlson_vars_1_101, method: unweighted, AUROC=0.7064
Sensitivity
1 − Specificity
Threshold
30000
10000
3000
1000
300
100
30
10
3
1
0.00
0.25
0.50
0.75
1.00


auc
0.8
0.7
0.6
all
all_0_5
all_10_30
all_30_100
all_5_10
cc
charlson_vars_0_1
charlson_vars_1_101
demo_vars_0_1
demo_vars_1_101
lab_vars_0_6
lab_vars_100_101
lab_vars_14_33
lab_vars_33_100
lab_vars_6_14
other_vars_0_101
smoke_vars_0_50
smoke_vars_50_101
df
method
resample

01/04/2022 update

Some follow-ups

- Patient data importance:
  - Smoking status (current: 121/239, former:52/239)
  - HIV indicator (231/239)
- Calibration metrics
  - resampling, using all data (50-25-25 split)
  - Create 51 bins with equal number of observations
    - i.e., every 2nd percentile
    - i.e., the 0.00, 0.02, ..., 0.98, 1.00 quantiles
    - n.b., the last bin is very large [0.0106, 1.0000] and causes problem to compute "expected" counts
  - Using all 51 bins:
    - Pearson correlation (0.920, 95% c.i. [0.864,0.954])
    - Spearman correlation (0.988)
    - Hosmer-Lemeshow (H=30593, df=49, p<2.2e-16)
  - Using all but last bin:
    - Pearson correlation (0.997, 95% c.i. [0.995, 0.998])
    - Spearman correlation (0.988)
    - Hosmer-Lemeshow (H=42.7, df=48, p=0.690)

Comparison to Kunzmann

- "Esophagal condition":
  - I could only match GERD and H2R
- Their cohort has much fewer smokers (US vs UK?, 40% missing?)
- Estimated OR compared to theirs
  - Age & Sex seem to have similar effect
  - BMI and smoking seem to have a weaker effect (smoking status has a lot of missing values, so SRS imputation should hide effect)
  - Esophagal condition, despite the mismatch, seem to have a similar effect

Variable	No EAC	EAC	Kunzmann OR (CI)	OR
Age				
0-50	1,094,536	187	–	0.20
50-55	435,106	400	1.00 (reference)	1.00
55-60	563,542	1,097	1.99 (1.11–3.65)	2.19
60-65	792,624	2,053	2.76 (1.60–4.74)	2.96
65+	2,082,924	4,770	4.03 (2.36–6.89)	3.18
Sex				
Female	346,096	50	1.00 (reference)	1.00
Male	4,632,189	8496	5.16 (3.58–7.44)	6.60
BMI				
0-25	1,024,990	1,531	1.00 (reference)	1.00
25-30	1,753,719	2,640	1.50 (1.02–2.21)	1.00
30-35	1,135,392	1,841	1.91 (1.25–2.94)	1.16
35+	664,785	1,310	2.97 (1.79–4.94)	1.57
Smoking				
Never	405,221	502	1.00 (reference)	1.00
Former	2,031,897	3,705	2.03 (1.47–2.80)	1.05
Current	2,253,878	4,083	3.83 (2.59–5.66)	1.35
Esophagal condition				
No	346,096	50	1.00 (reference)	1.00
Yes	4,632,189	8496	1.88 (1.39–2.54)	1.44

Sensitivity analyses

- 2–5:
  - 4 years of data, predicting 1 years ahead
  - usual analysis
- 4–5:
  - 2 years of data, predicting 3 years ahead
  - i.e., using year -5 and -4 to predict outcome
- same pipeline as before, using 100K controls
- AUCs: 0.7837 (2–5) vs 0.7565 (4–5)
- Models are miscalibrated, needs to be improved ...

Other updates

- Currently working on wrapping everything in a R package for simpler/easier usage and deployment.
- Also, implementation of some functions making sensitivity analyses easier (i.e., automating the data preprocessing)
- Starts with the assumptions that the user would input the same file as we got (patient info, medication, codes, labs)

01/04/2022 update follow-up

Sample sizes

- Training set (75%): 4,986,831 observations, 8,546 cases, 4,978,285 controls
  - Further split in training + validation
  - Training set (50%): 3,324,554 observations, 5,697 cases 3,318,857 controls
  - Validation set (25%): 1,662,277 observations, 2,849 cases 1,659,428 controls
- Testing set (25%): 1,662,277 observations, 2,849 cases, 1,659,428 controls

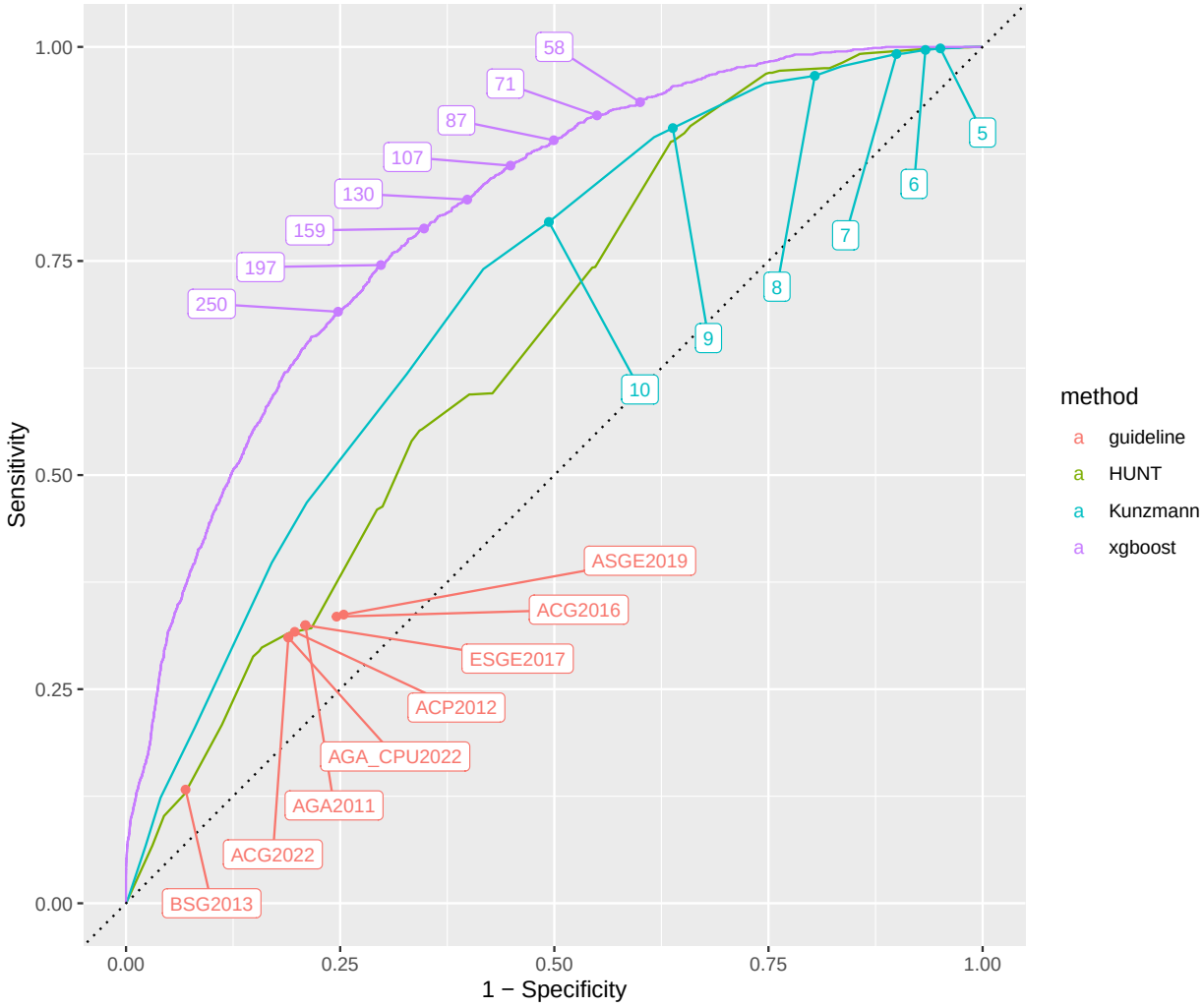
HUNT method

- This is not a logistic regression model so there is not really an "intercept" per se, only a baseline hazard when all variables are turned off. The baseline hazard seems to be 3.6.
- In any case, this does not really matter for computing AUC and plotting the ROC curve as we would only multiply all rates by some factor and this would produce the same results
- I will not be providing classification metrics (sensitivity, specificity, etc.) as this would require choosing a threshold. Let me know if you'd like to have them anyway.

Comparison to HUNT and Kunzmann

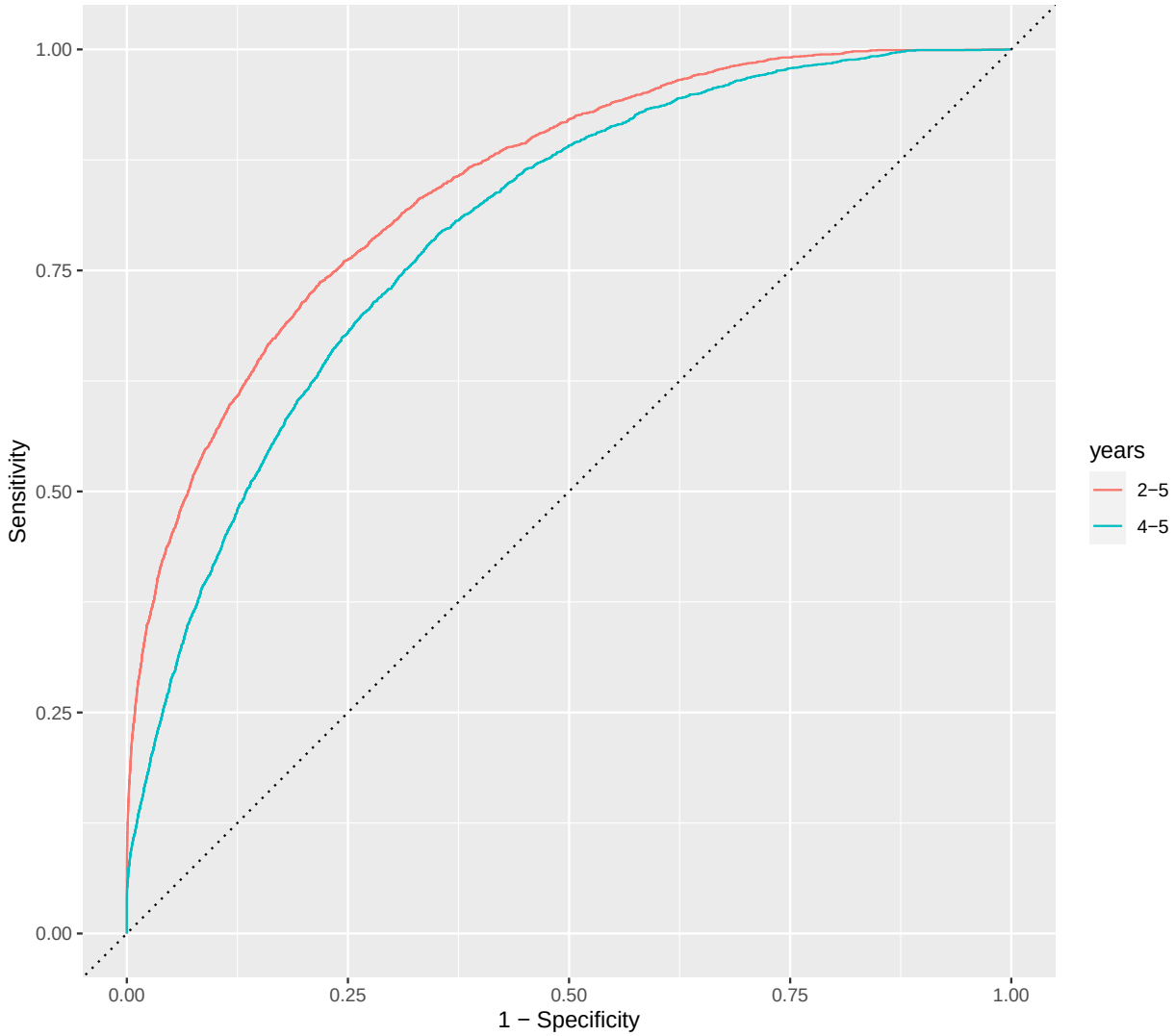
- Take the original testing set (25%: 2,849 cases, 1.66M controls)
- Subset to observations that are complete for both HUNT and Kunzmann (i.e., with observed age, sex, bmi, smoking status, GERD, H2R, PPI)
- Bin variables in the same way as original papers; construct the same indicators (smoking and esophagal condition)
- Resulting test set: 1,816 cases, 872,093 controls
- Used to compute a score using all three methods (xgboost, HUNT, Kunzmann)
- Plot ROC curves, and compute AUC
  - xgboost: 0.7869407
  - HUNT: 0.6500223

– Kunzmann: 0.7046042



Window comparison

- Years 2–5 vs 4–5 to predict outcome
- Same patients in both test sets (same test set as previously: 2.8K cases, 1.66M controls)
- Model: trained on 2–5 data
- AUCs:
  - 2–5: 0.8468027
  - 4–5: 0.7950466



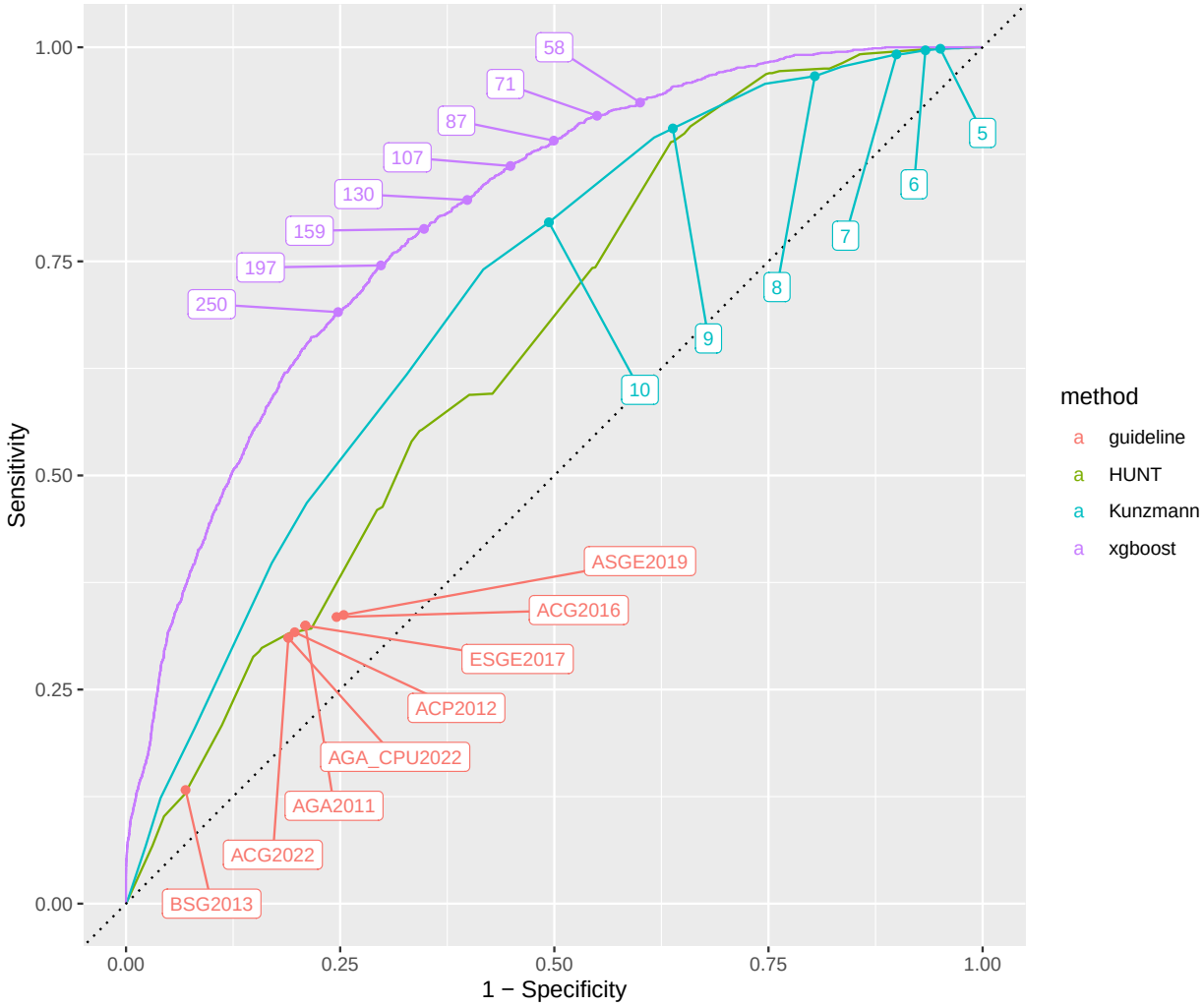
01/11/2022 update

- Server maintenance over the last few days ...
- Sensitivity analyses:
  - Data preparation for any prediction window (fewer years, later prediction, etc.)
    - e.g. 5-1 to predict 0, 5-4 to predict 0, 2-1 to predict 0, etc.
    - The R package will allow to construct these dataset very easily (just change parameters)
  - EAC vs EGJAC:
    - how to get this information? (ICD 10: C15.x vs C15.5?; ICD9: 150.x vs 150.2/5?)
    - We were always working with 'CaseControl' which was precomputed
    - Goal: Compare whether separate models does better than a single model?
    - Compare to HUNT & Kunzmann to check if this explain the big discrepancy?
- Implementation:
  - Working on an R package (hosted on GitLab)
  - Input: SAS files structured as before containing info for a set of patients
  - Read SAS Files, construct standardized data frame, use xgboost model to obtain predictions (averaged over multiple SRS for missing values)
  - Output: (ID, risk score). Format?
  - Standard R package documentation

01/18/2022 update

Added screening guidelines to ROC plot (see below)

- Kunzmann uses more BMI and age bins than HUNT
- Kunzmann uses medication data in addition to GERD
- The guidelines uses even less information and simpler combination
- NB: we cannot have the exact Kunzmann given our data, especially the 'esophagal condition' indicator is incomplete
- NB: similarly for guidelines, we are missing family history and other clinical information



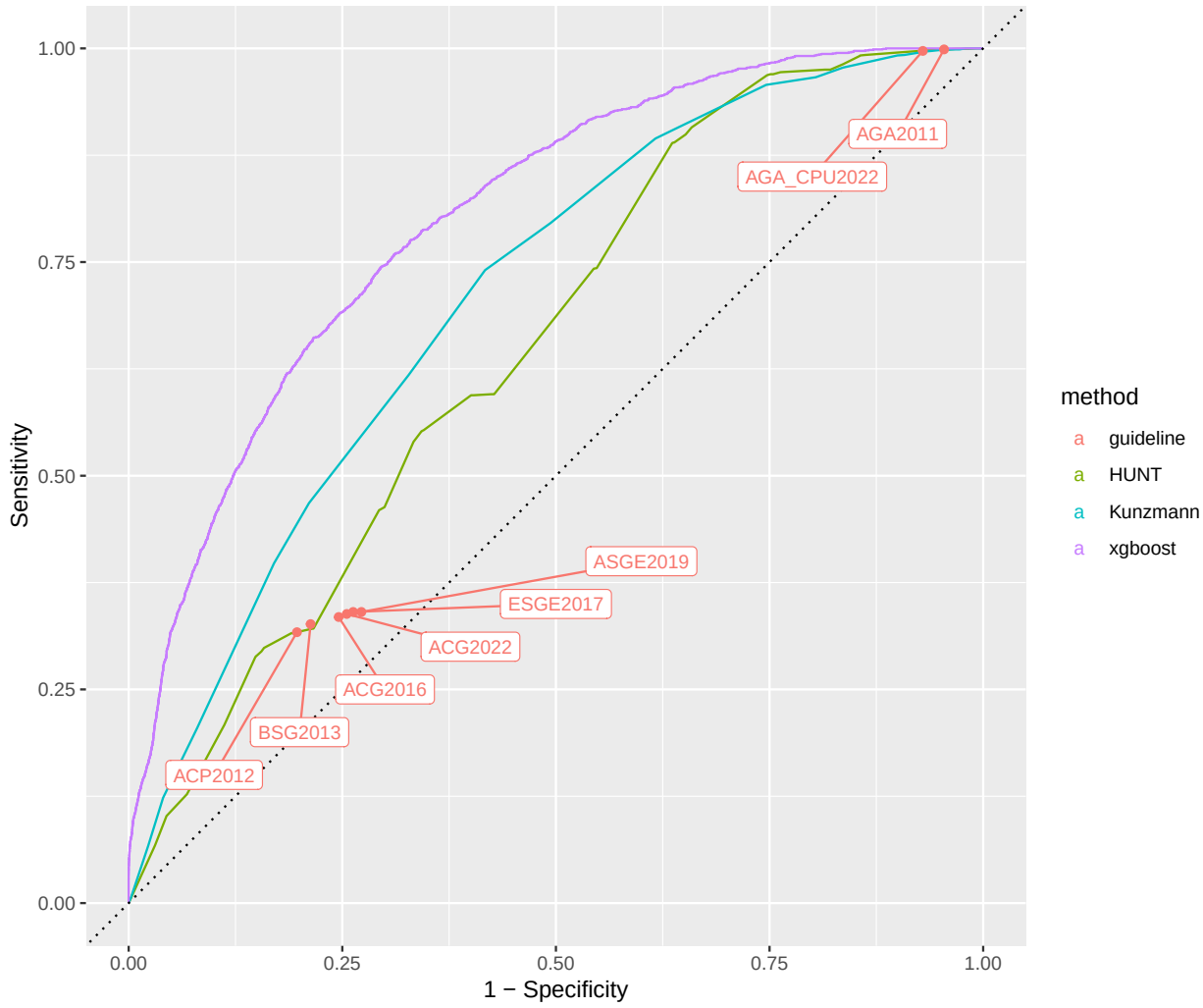
Other updates

- Waiting on EAC/EGJAC indicators
- R package:
  - Data processing functions complete
  - Next: use model to obtain scores and ouput
- Prediction window sensitivity analyses:
  - Takes a long time to preprocess the data
  - Ran into a small bug, could not finish processing
  - Once processing is done, easy to get performance

01/25/2022 update

Last week follow-up

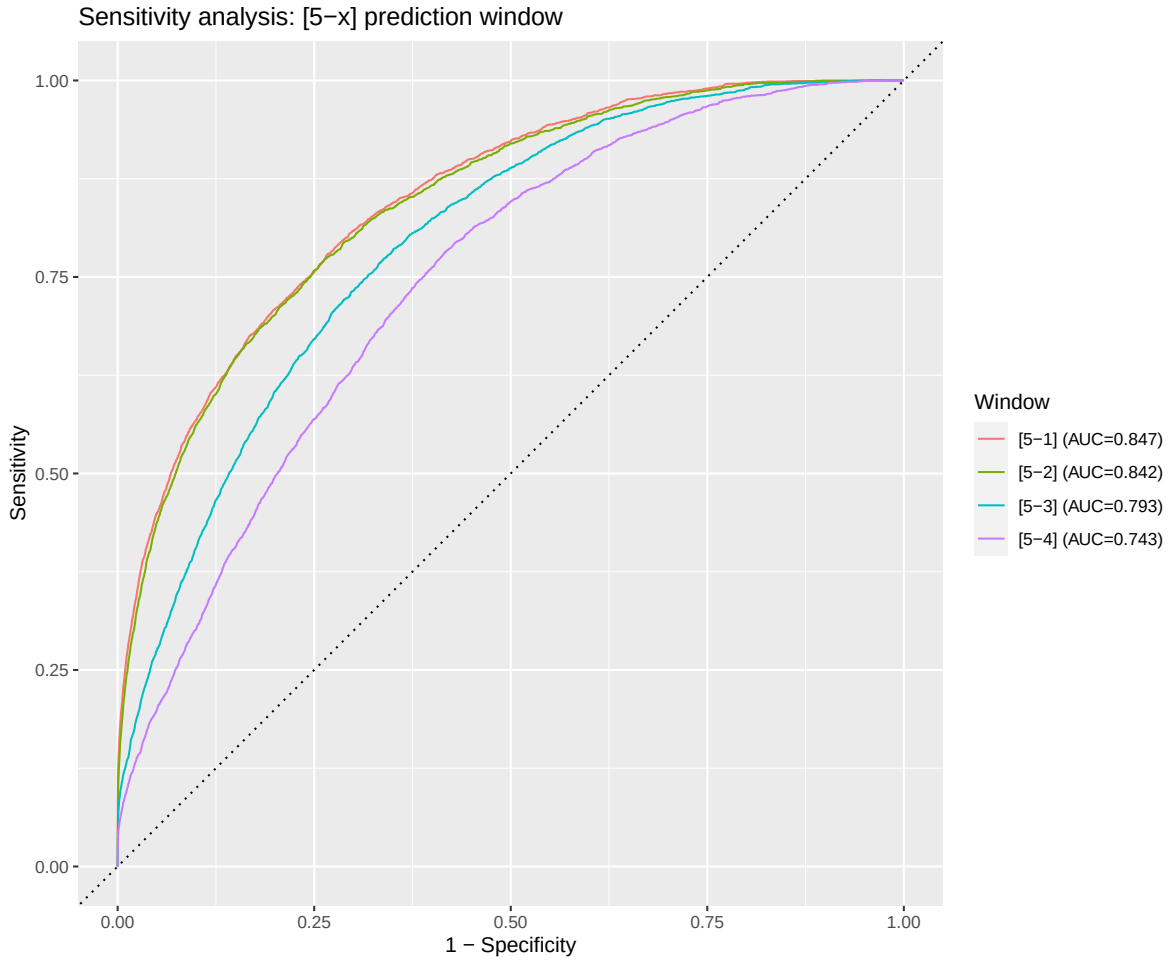
- There was a slight logical error in my calculation of the screening guideline
- Still essentially the same results, except two are pushed to top right
- Any guidline requiring GERD is capped
  - Prevalance is around 34% in cases
  - This means Sensitivity can be at most 34% for those guidelines
  - Prevalance around 27% in controls; other risk factors reduce the FPR a bit

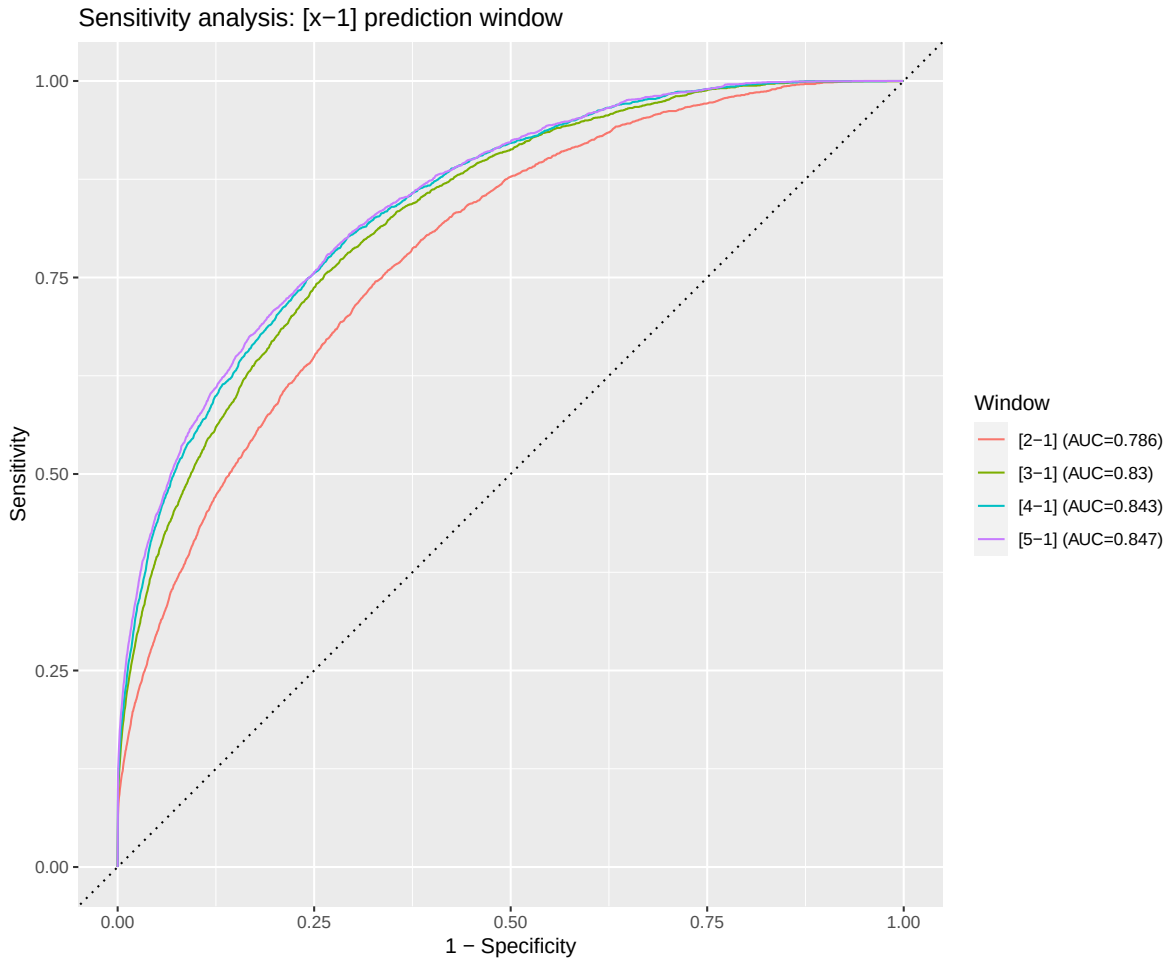


	Controls		Cases	
Nb. of risk factors	N	%	N	%
0	1,853	0.2	0	0.0
1	9,410	1.2	1	0.1
2	44,871	5.6	4	0.2
3	160,169	20.1	110	6.5
4	313,723	39.5	635	37.6
5	217,734	27.4	731	43.3
6	47,169	5.9	207	12.3
	794,911	100.0	1,688	100.0

Sensitivity analysis on prediction window

- I finally have the results!
- Predicting from farther away (+ decreasing aggregation window)
  - Removing the latest year has essentially no effect
  - Removing the last two years or more has a noticable effect
- Decreasing aggregation window (still predicting 1 year later)
  - Dropping the first year has almost no effect on performance
  - Having only 1 year leads to a large gap





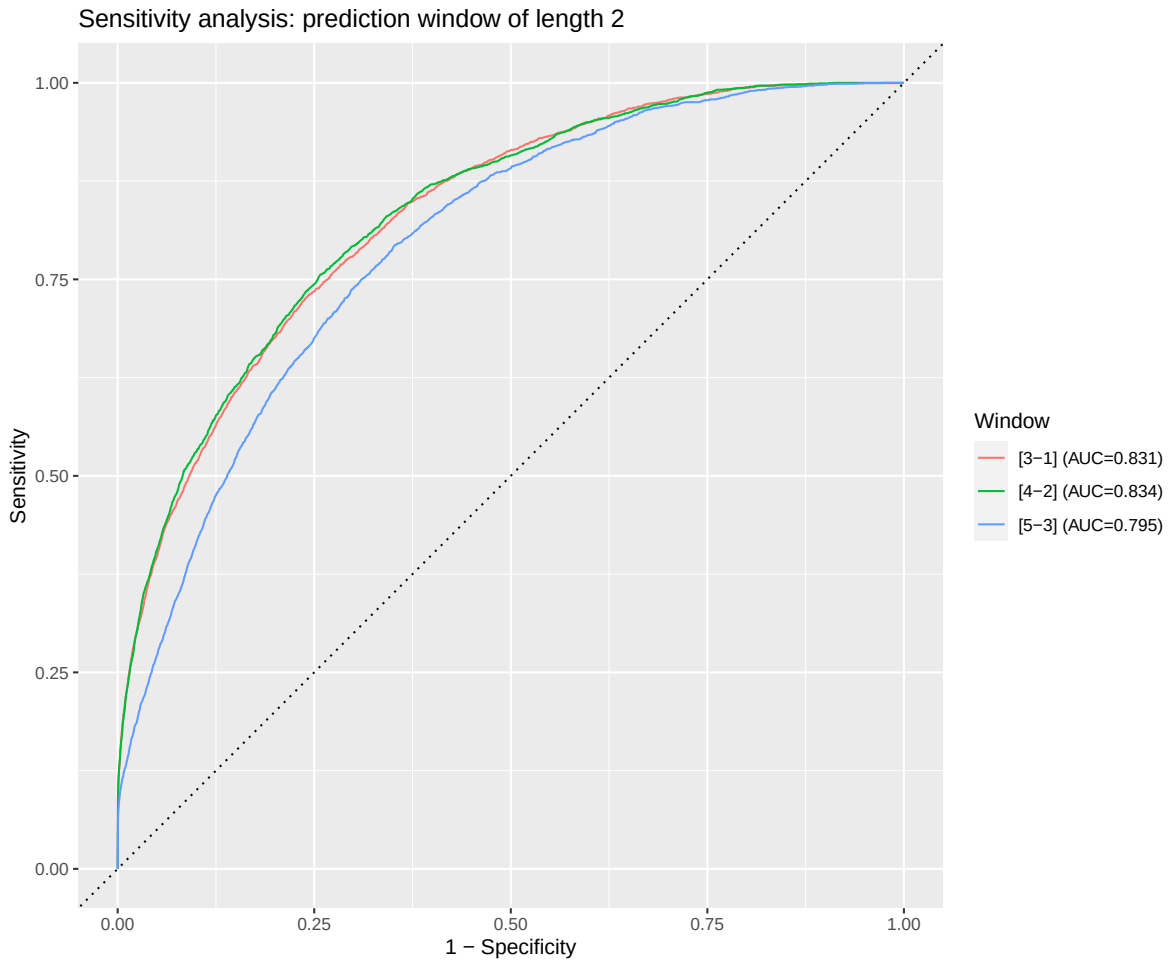
Next steps

- EAC v EGJAC
- Finalizing R package
- Study multiple imputations (effect on performance, variance of predicted risk by % of missing values and which values are missing)

02/01/2022 update

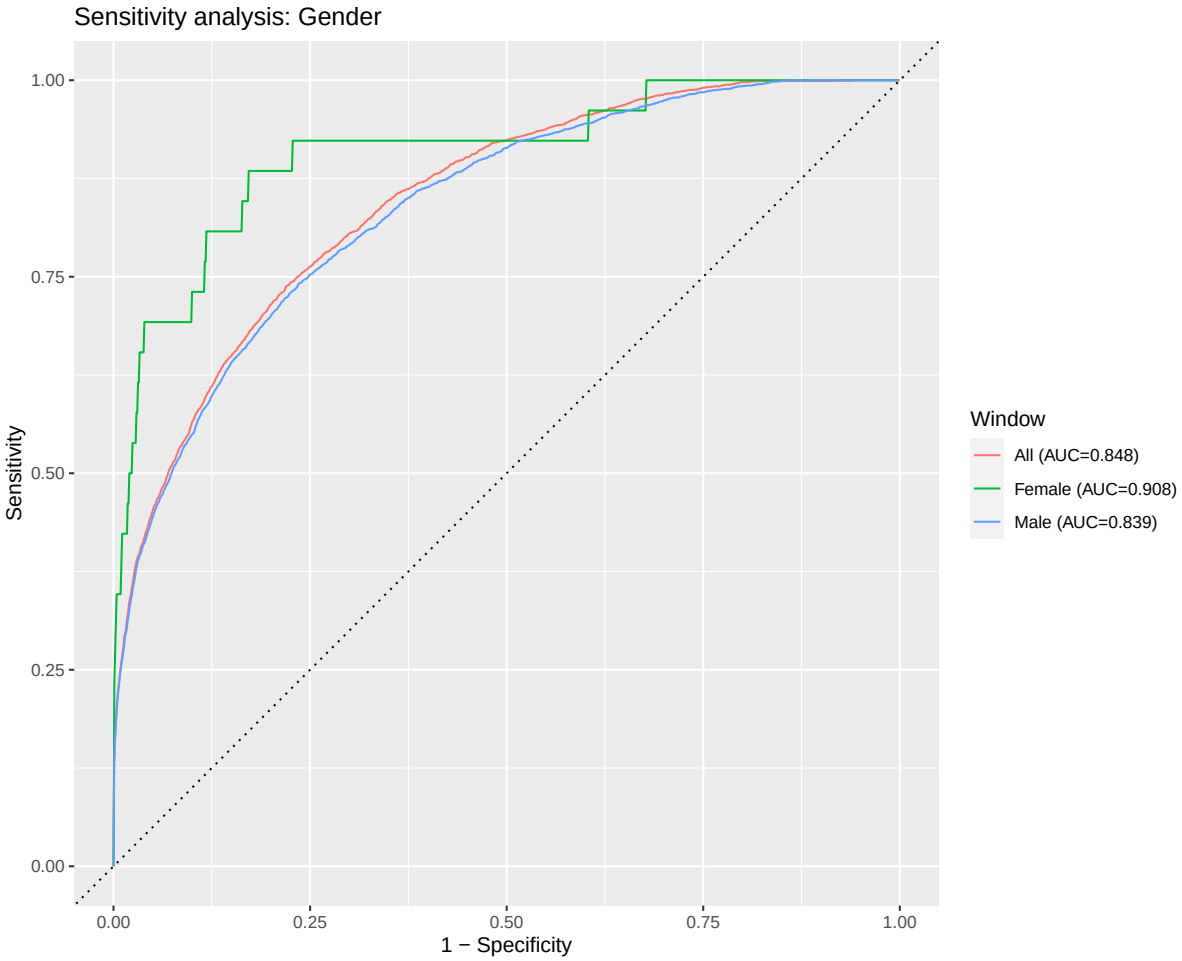
Follow-up on last week

- Fairer comparison when predicting farther in the future
- Prediction window of length 2, slides over the 5y period
- All observations have similar amount of data, but prediction is farther and farther back in time (1, 2, 3 years prior)
- No difference between years 1 and 2, but a significant drop at the 3rd year
- Similar conclusion as last week

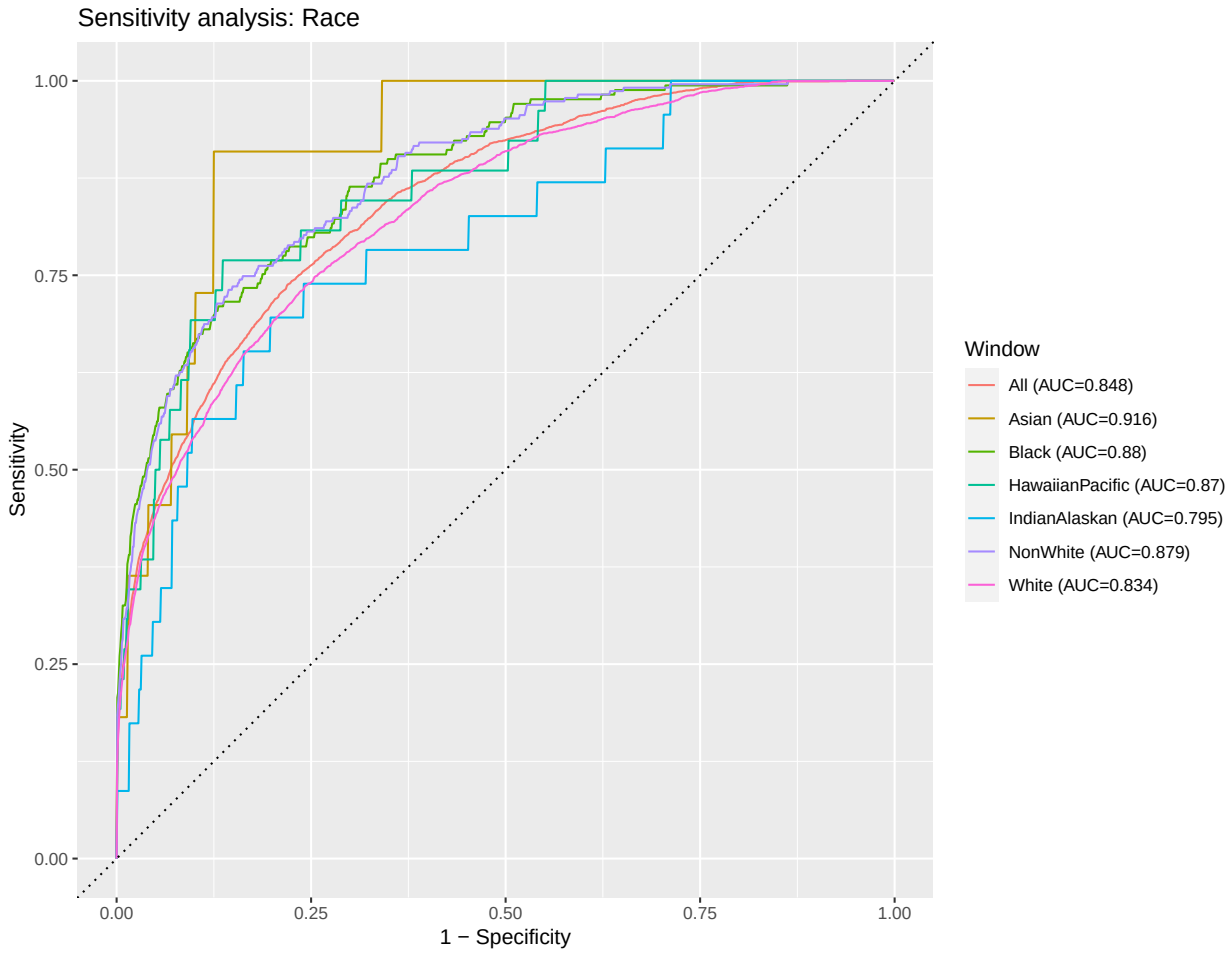


Performance on identity-defined test subsets

- Same test set as before (1.5M Controls, 3K cases)
- Gender:
  - 93% Male, 7% Female
  - Female: 26/115K cases/control

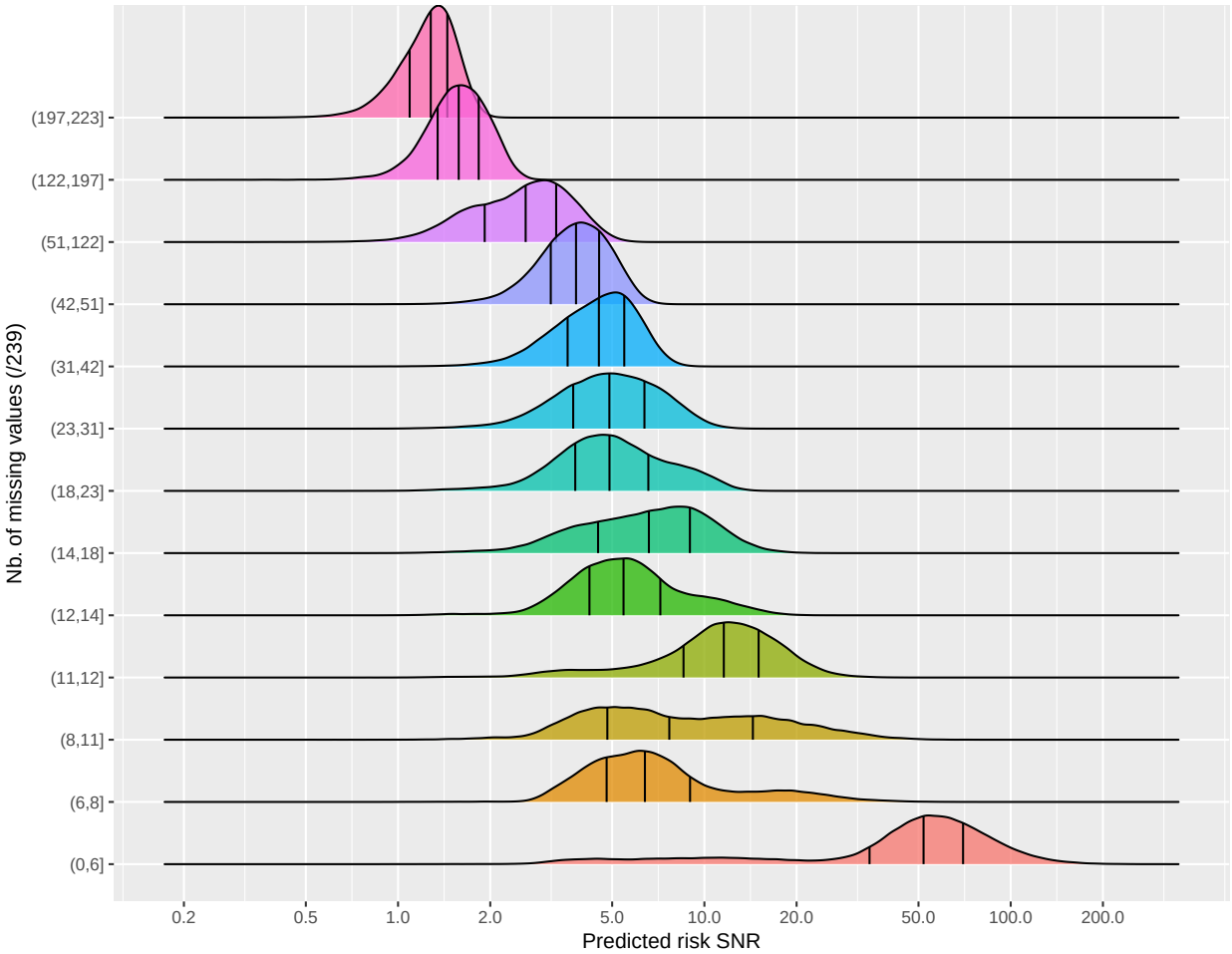


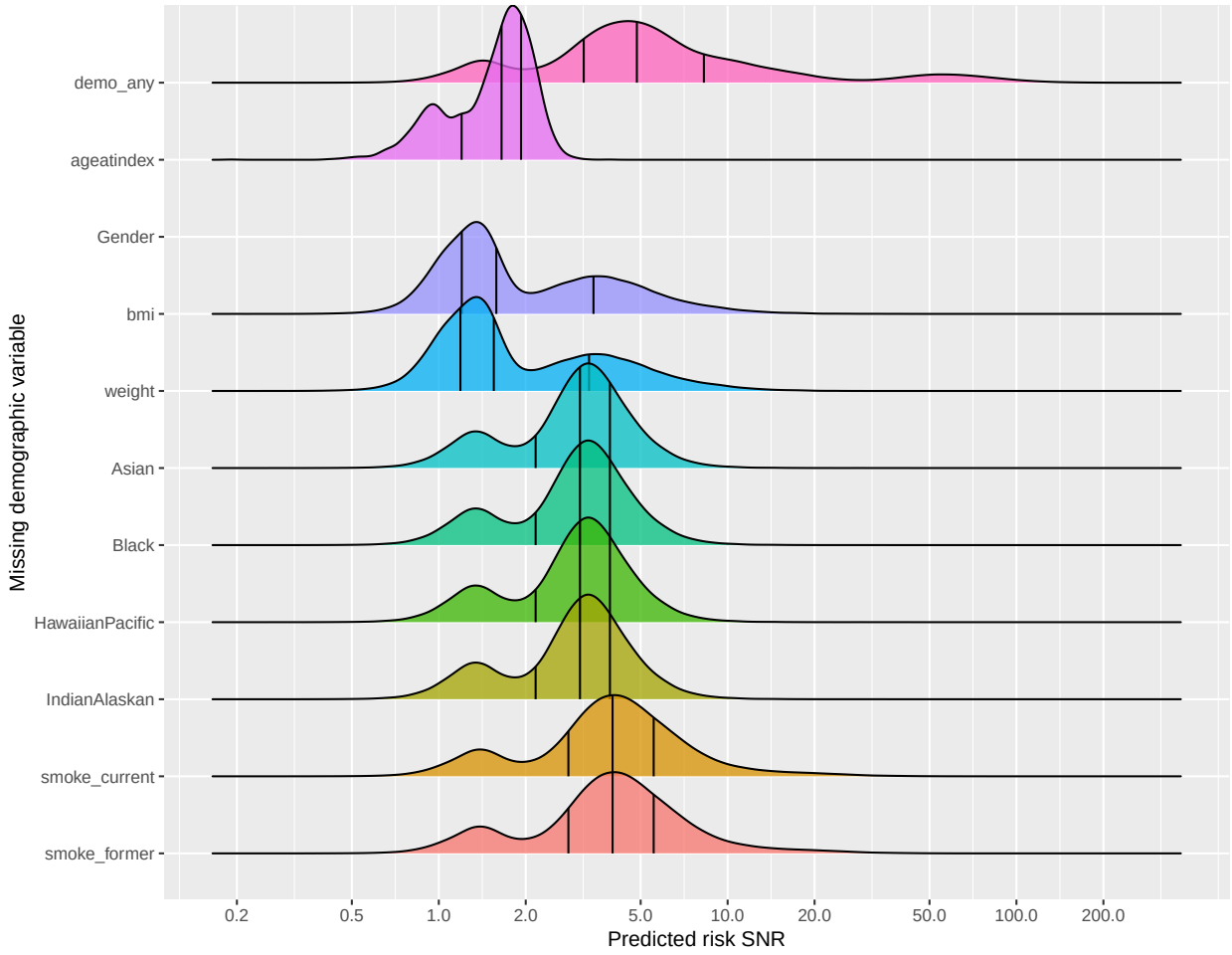
- Race:
  - 80% White, 1% Asian, 17% Black, 1% Hawaiian/Pacific, 1% Indian/Alaskan



Sensitivity analysis: missing values

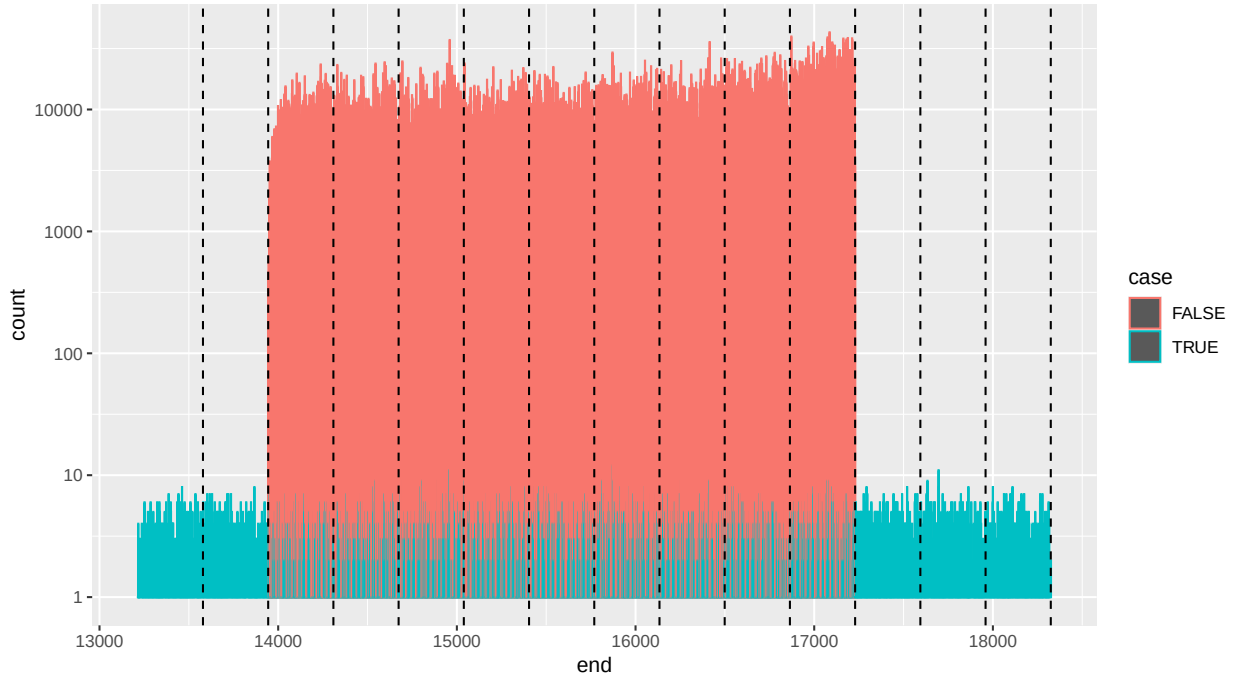
- For each observation:
  - Get 100 imputations
  - Get predicted risk for each imputation
  - Compute signal-to-noise ratio over the 100 risks
- High SNR means that the predicted risk does not change much with imputed values; small SNR means high dependence on the imputed values
- Partition the test set by the number of missing values
- Subsets defined by whether each demographic variable is missing




demo_any
ageatindex
Gender
bmi
weight
Asian
Black
HawaiianPacific
IndianAlaskan
smoke_current
smoke_former
Missing demographic variable
0.2
0.5
1.0
2.0
5.0
10.0
20.0
50.0
100.0
200.0
Predicted risk SNR

Sensitivity analysis: ICD10

- Data processed, results to come
- I settled on using 17229 as the last end date (then use -3 to -1 year)



Sensitivity analysis: Cancer type

- Trained separate models, comparison to come

02/08/2022 update

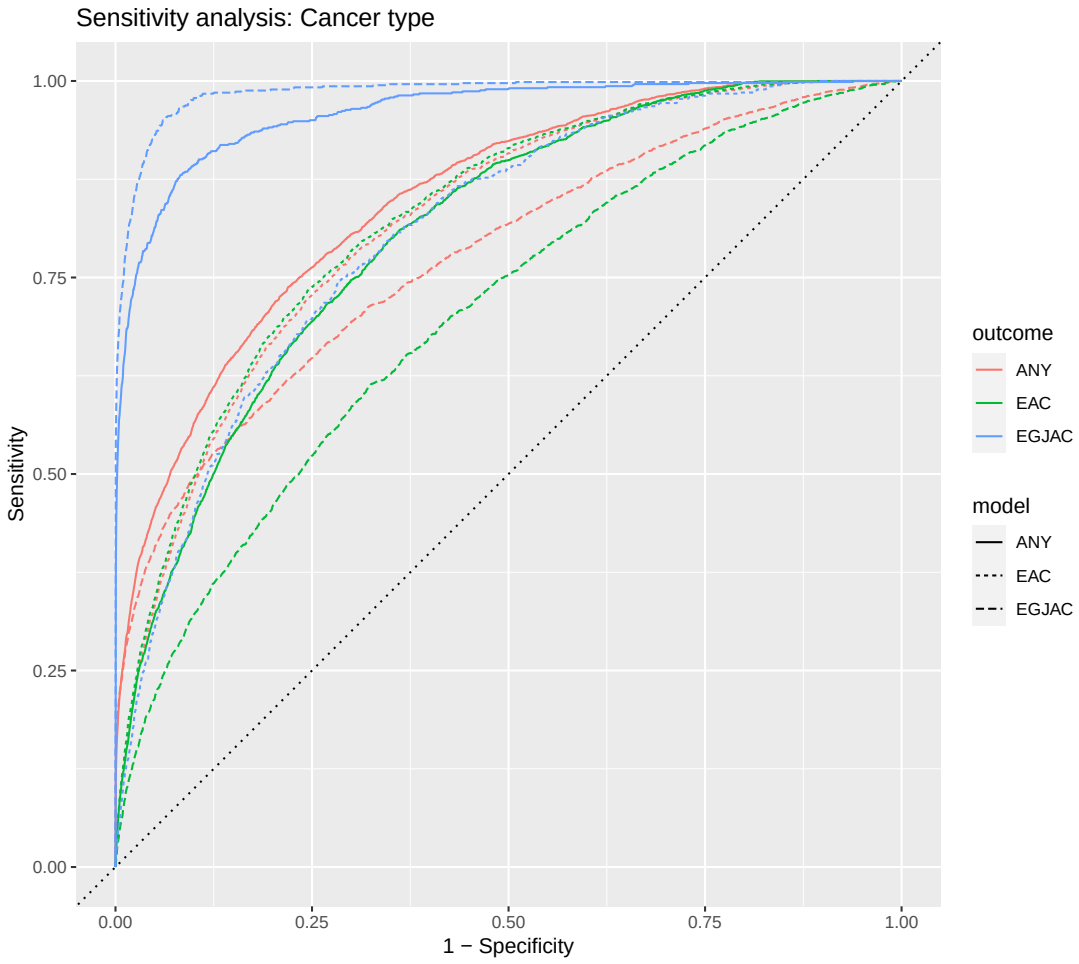
ICD 9/10

- Updated histogram on previous page, now shows cases/controls
- It appears I have 9 years of controls, but 14 years of cases
- Implies no ICD controls

Sensitivity analysis: Cancer type

- 3 outcomes: ANY, EAC, EGJAC (what is y in the test set)
- 3 models: ANY, EAC, EGJAC (what is y in the training set)
- Evaluate test performance of all three models on all three outcomes (same X, different y, same patients)
- As expected, model x performs best on outcome x
- EGJAC seems much easier to predict than EAC, even though there are fewer of them
- The ANY model is almost as good as individual models on each of EAC or EGJAC
- EGJAC is particularly worse at predicting ANY/EAC

Test AUC			
	Outcome		
Model	ANY	EAC	EGJAC
ANY	0.848	0.806	0.960
EAC	0.819	0.824	0.805
EGJAC	0.775	0.698	0.986
Nb. of cases			
Whole sample	11395	8430	2965
Test set	2849	2087	762

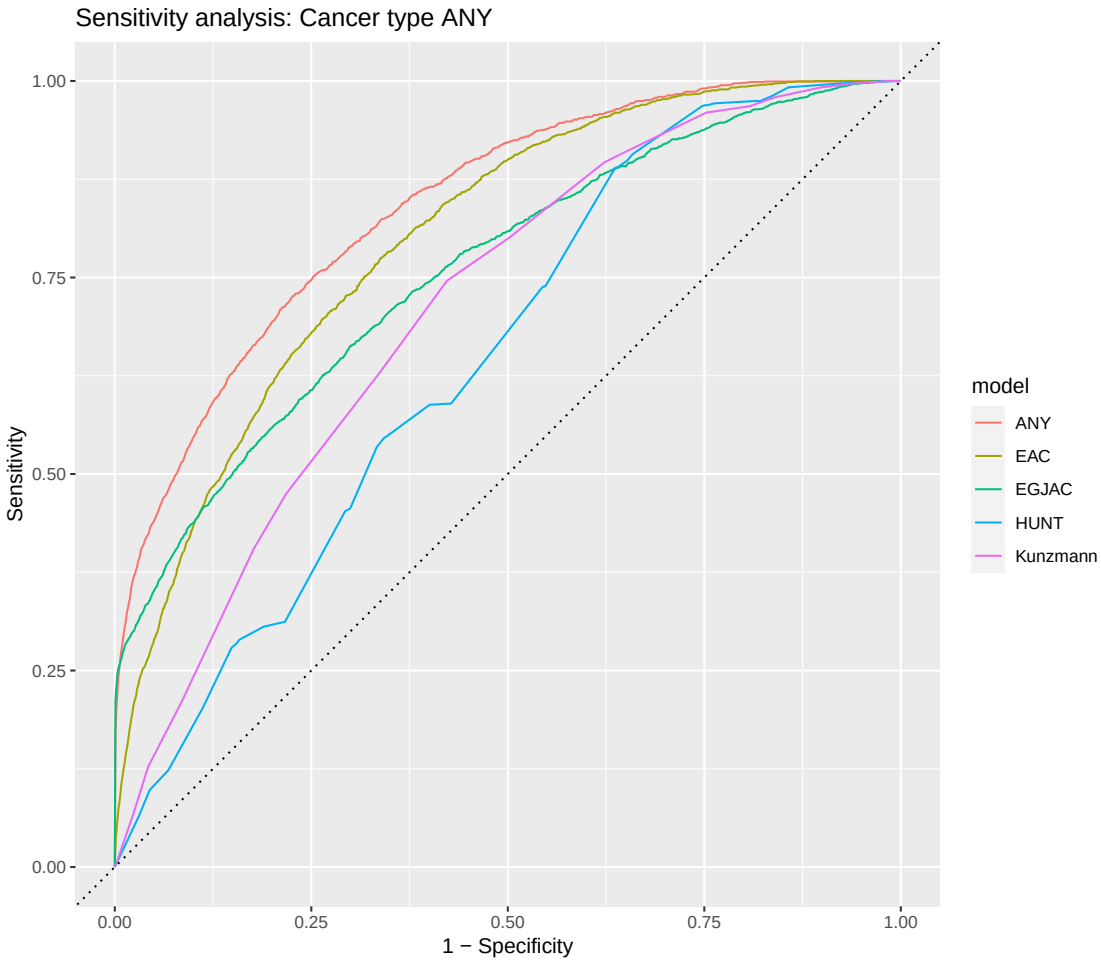

Sensitivity analysis: Cancer type
Sensitivity
1 − Specificity
0.00
0.25
0.50
0.75
1.00
outcome
ANY
EAC
EGJAC
model
ANY
EAC
EGJAC

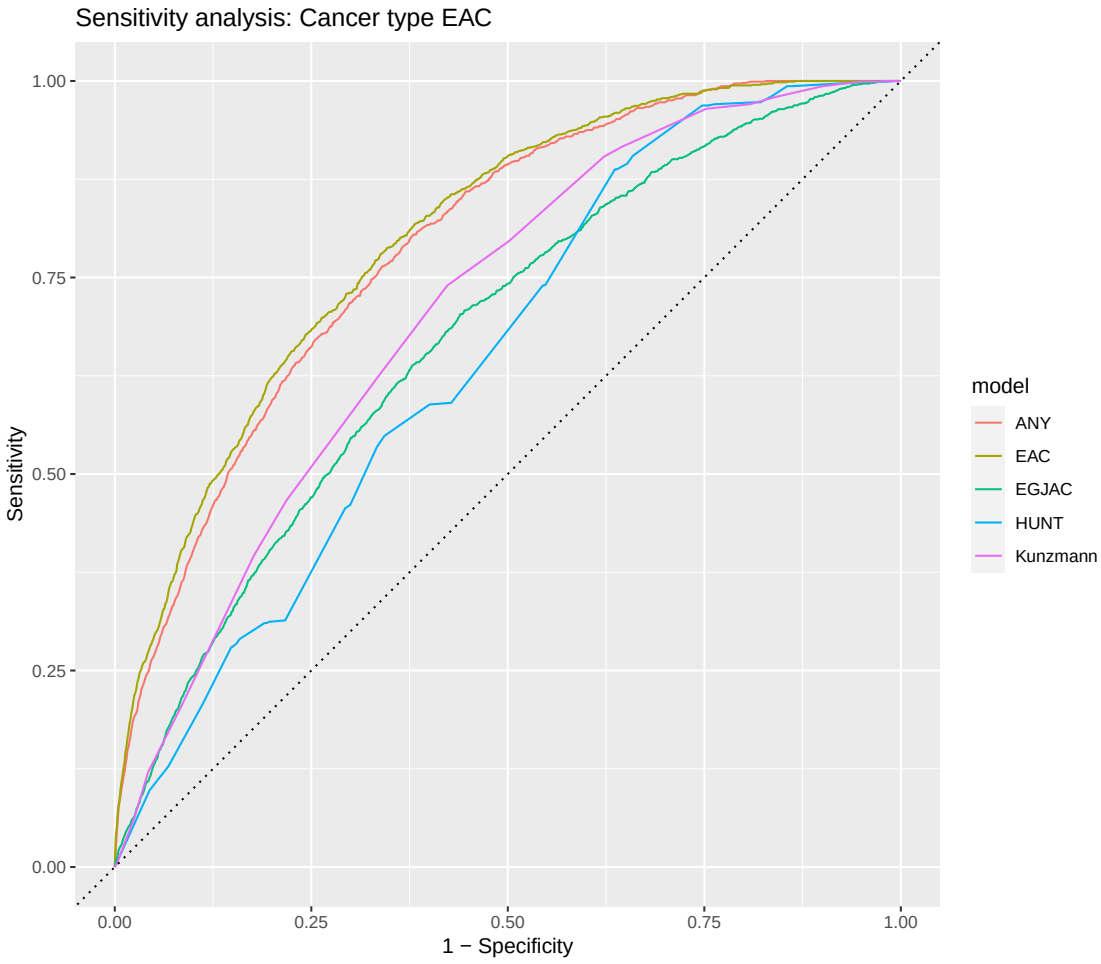
02/15/2022 update

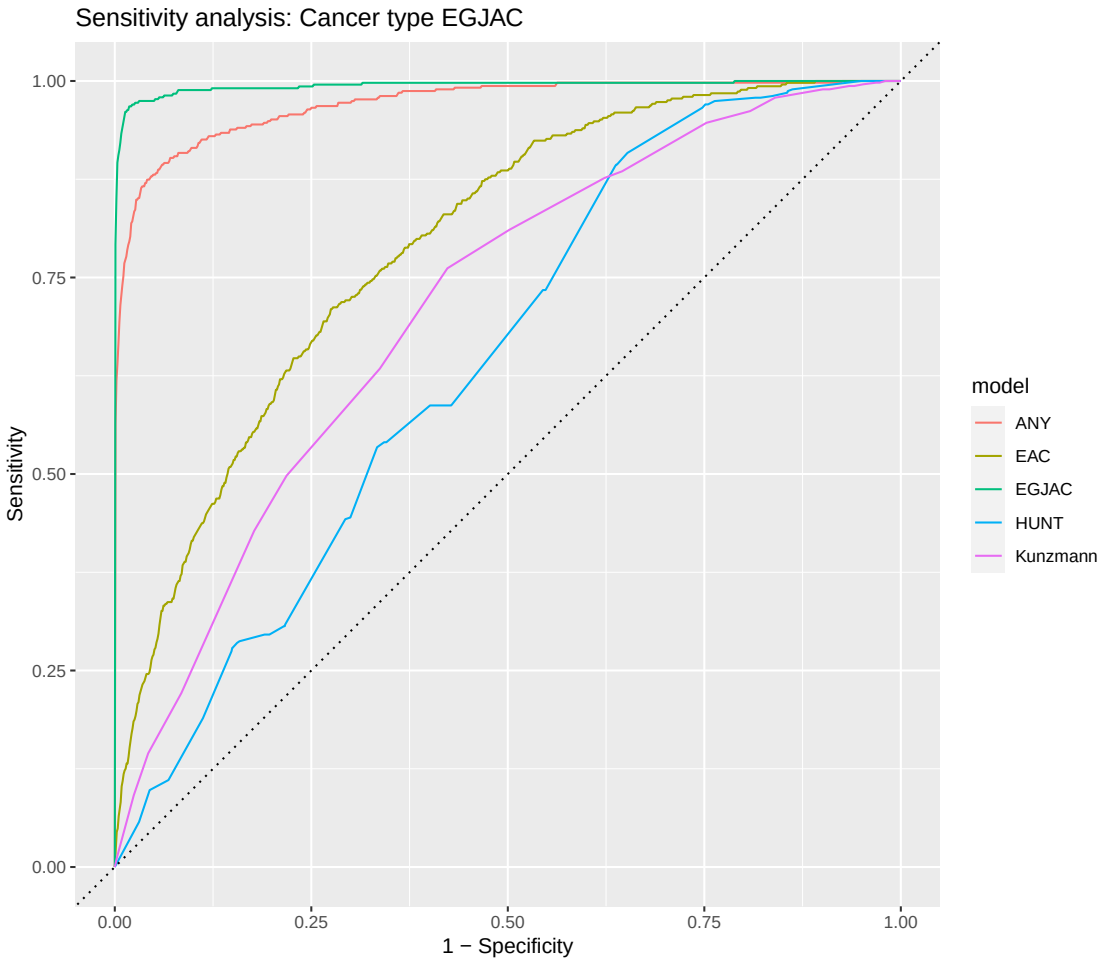
Cancer type analysis: follow-up

- Subset to "complete cases" (wrt HUNT/Kunzmann), about 50% of the original data
- Essentially the same results
- HUNT and Kunzmann: similar performance across all three outcomes

Test AUC			
	Outcome		
Model	ANY	EAC	EGJAC
HUNT	0.649	0.650	0.646
Kunzmann	0.708	0.707	0.712
ANY	0.841	0.791	0.971
EAC	0.799	0.802	0.792
EGJAC	0.758	0.675	0.994
Nb. of cases			
Whole sample	6687	4910	1777
Test set	1688	1218	470


Sensitivity analysis: Cancer type ANY
1.00
0.75
0.50
0.25
0.00
Sensitivity
0.00
0.25
0.50
0.75
1.00
1 − Specificity
model
ANY
EAC
EGJAC
HUNT
Kunzmann


Sensitivity analysis: Cancer type EAC
Sensitivity
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
1 − Specificity
model
ANY
EAC
EGJAC
HUNT
Kunzmann


Sensitivity analysis: Cancer type EGJAC
Sensitivity
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
1 − Specificity
model
ANY
EAC
EGJAC
HUNT
Kunzmann

03/01/2022 update

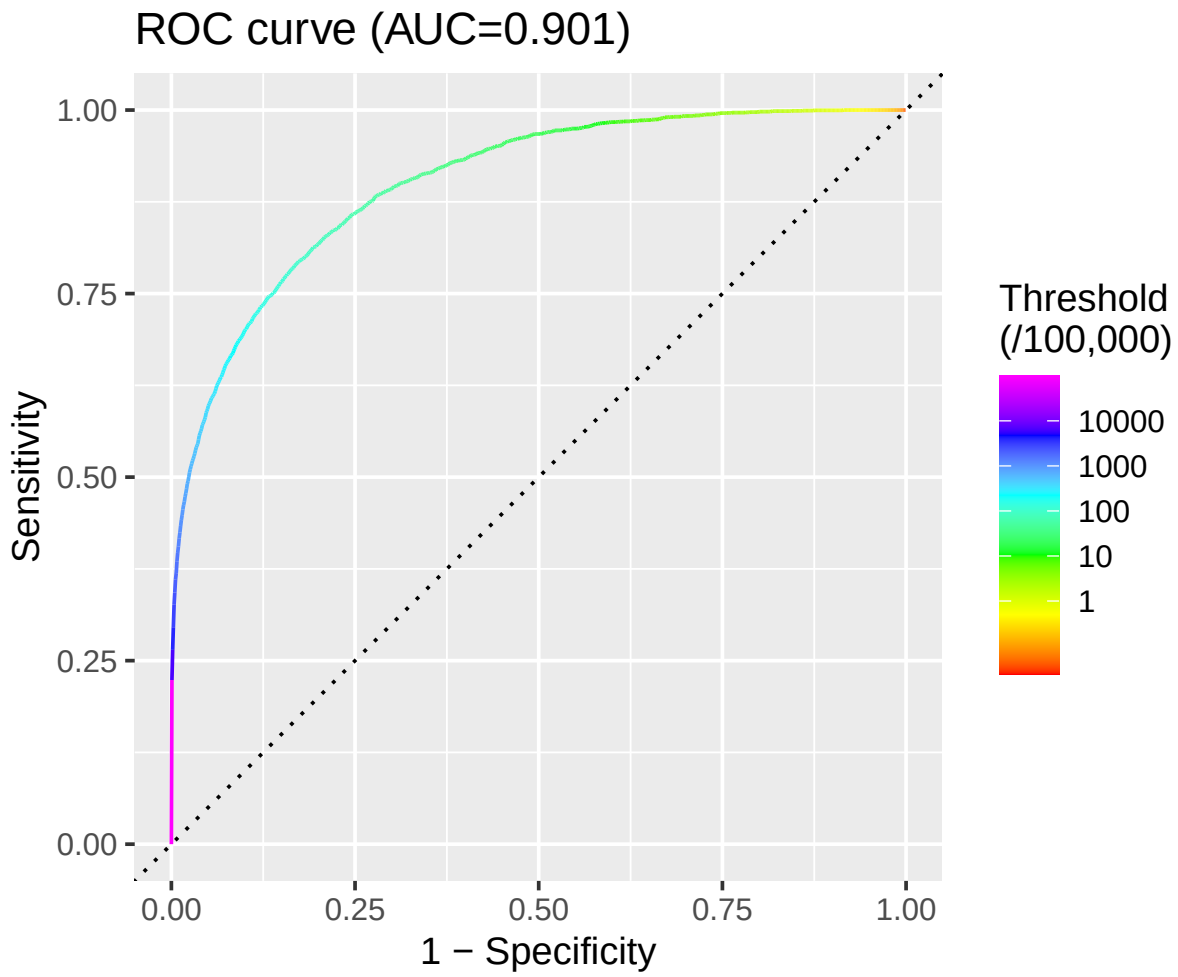
New data

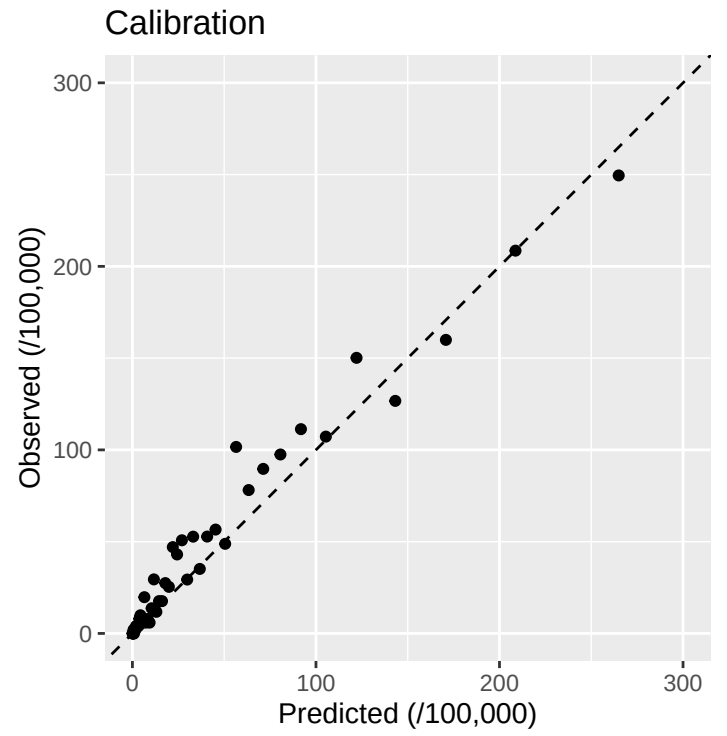
- I was able to process it and fit the full model over the weekend
- As well as some submodels for sensitivity analyses
- I identified a few issues, however:
  - There are more cases than before (about 2x), I was expecting only new controls?
  - There were **repeated IDs** which I missed before running everything (1:10,254,489, 2:27,378, 3:300, 4:16)
  - It seems it occurs for most/all cases (10,287,428 + 22,781, after dropping repeated: 10,254,474 + 22,781, diff: 32954 + 0)
  - Re-running processing at the moment
  - Whe trying to compute Kunzmann & HUNT, I found that Gerd was missing for all cases and only for about 50% of controls
  - This wasn't the case before ...

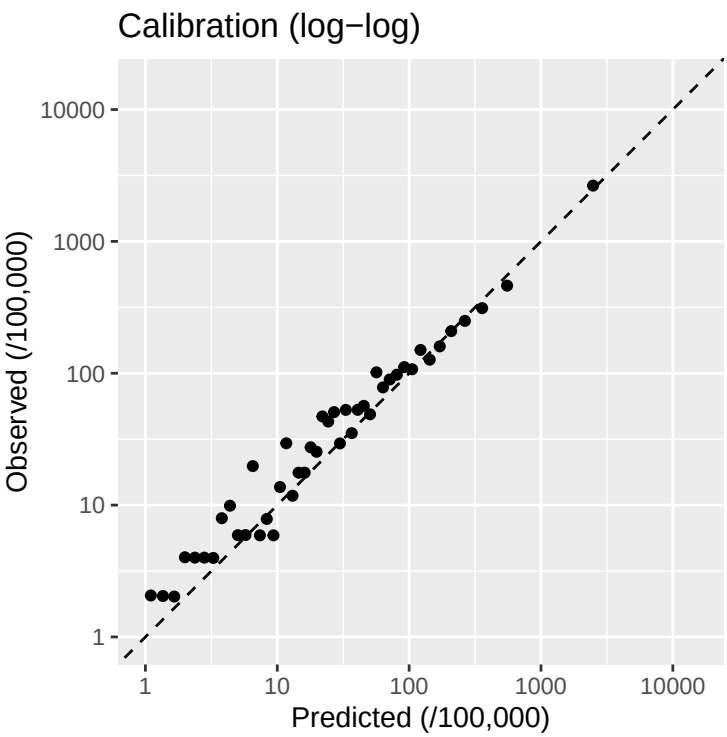
Analyses

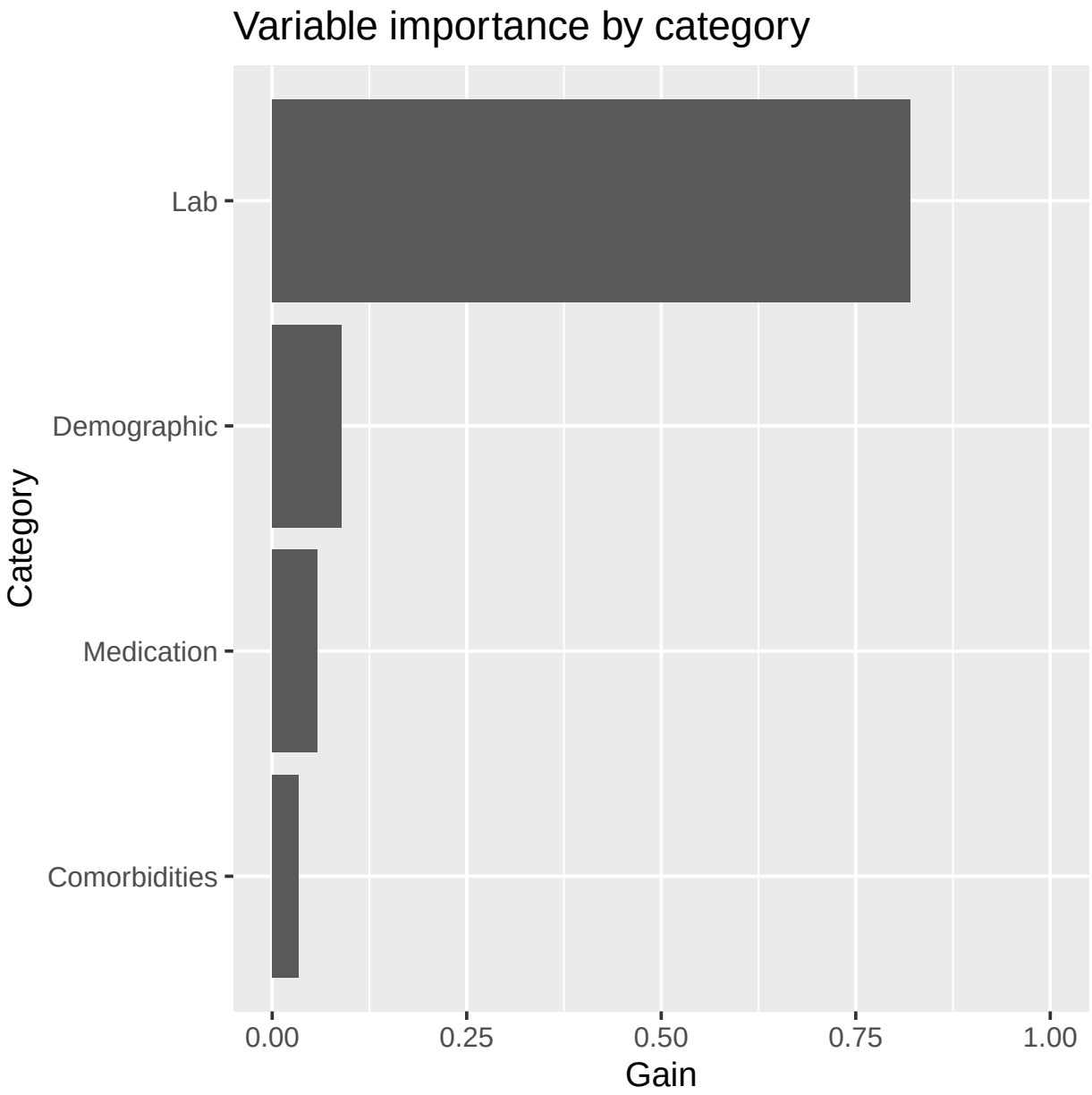
ready	Calibration, ROC, AUC
ready	Threshold table
ready	Variable importance (aggregated by group/category)
ready	Partial dependence of Demographic, Comorbidities and most important features
ready	Aggregation window comparison (x-1, sliding window of size 2)
ready	Cancer type analysis
ready	Identity (Gender, Race)
to do	Comparison to Kunzmann, HUNT and guidelines
to do	ICD 10 considerations
to do	Effect of missing values, multiple imputation
to do	Others?

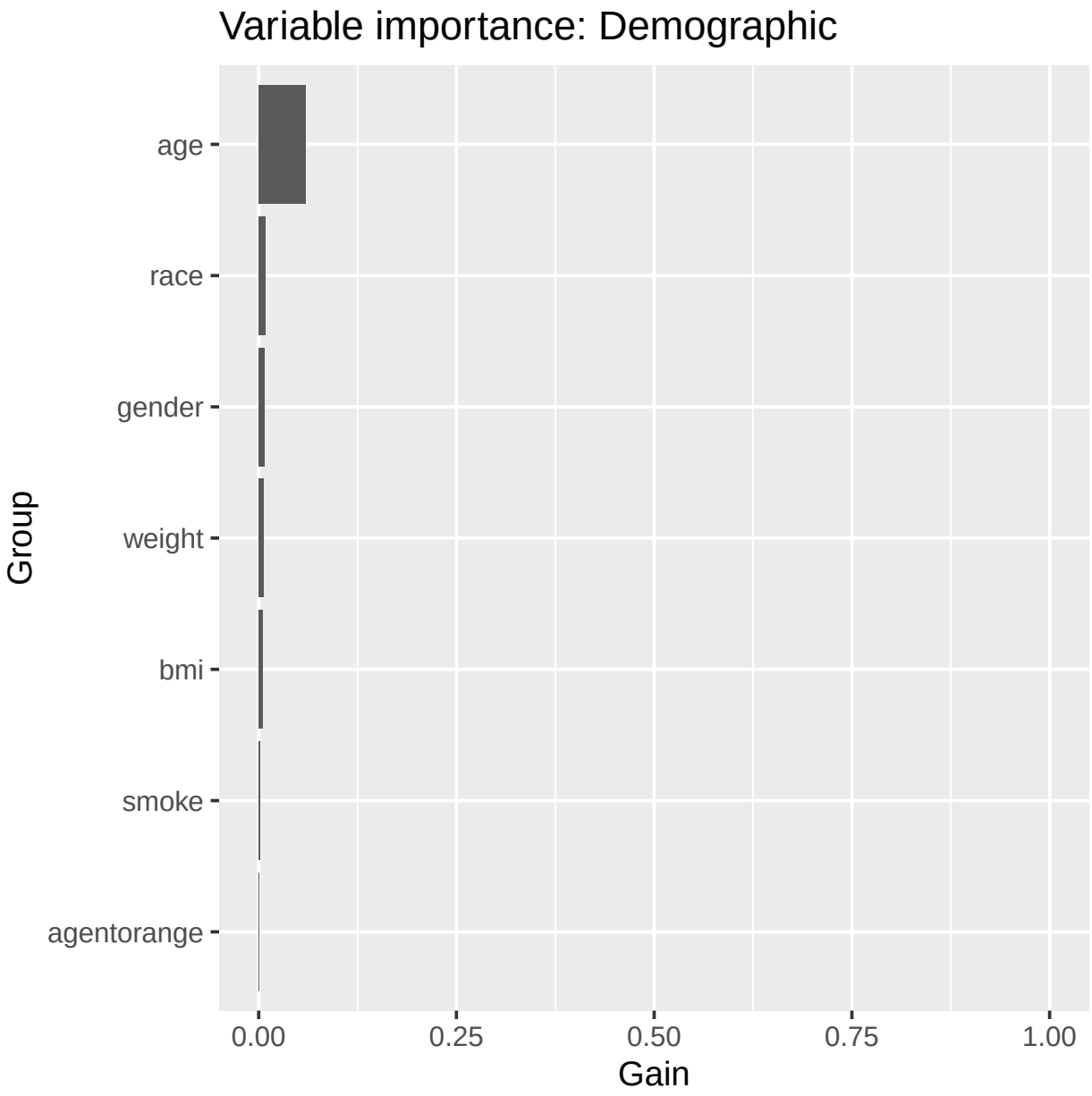
Threshold	TPR	PPV	Det. prev.
10	99.81	0.51	74.92
20	99.36	0.61	63.30
30	98.76	0.69	55.49
40	98.20	0.77	49.49
50	97.69	0.85	44.58
55	97.45	0.89	42.44
60	97.13	0.93	40.50
65	96.75	0.97	38.69
70	96.44	1.01	37.02
75	96.08	1.05	35.45
80	95.76	1.09	33.99
85	95.42	1.13	32.65
90	95.02	1.17	31.38
95	94.62	1.21	30.19
100	94.38	1.25	29.09
105	93.97	1.29	28.03
110	93.60	1.34	27.05
115	93.24	1.38	26.11
120	92.98	1.42	25.24
125	92.65	1.47	24.39
130	92.29	1.51	23.60
135	92.03	1.55	22.85
140	91.70	1.60	22.13
145	91.50	1.65	21.44
150	91.15	1.69	20.79
155	90.83	1.74	20.18
160	90.49	1.78	19.59
165	90.12	1.83	19.03
170	89.85	1.88	18.49
175	89.60	1.92	17.98
180	89.35	1.97	17.48
185	89.03	2.02	17.01
190	88.65	2.07	16.55
195	88.36	2.12	16.12
200	88.10	2.17	15.70
210	87.63	2.27	14.91
220	87.19	2.37	14.19
230	86.68	2.48	13.51
240	86.21	2.58	12.89
250	85.62	2.68	12.31
260	85.20	2.79	11.77
270	84.79	2.91	11.25
280	84.31	3.02	10.77
290	83.83	3.13	10.33
300	83.29	3.24	9.91
325	82.40	3.54	8.98
350	81.30	3.84	8.17
375	80.44	4.15	7.48
400	79.76	4.48	6.87
425	79.02	4.81	6.34
450	78.44	5.16	5.87
475	77.77	5.51	5.45
500	77.12	5.86	5.08

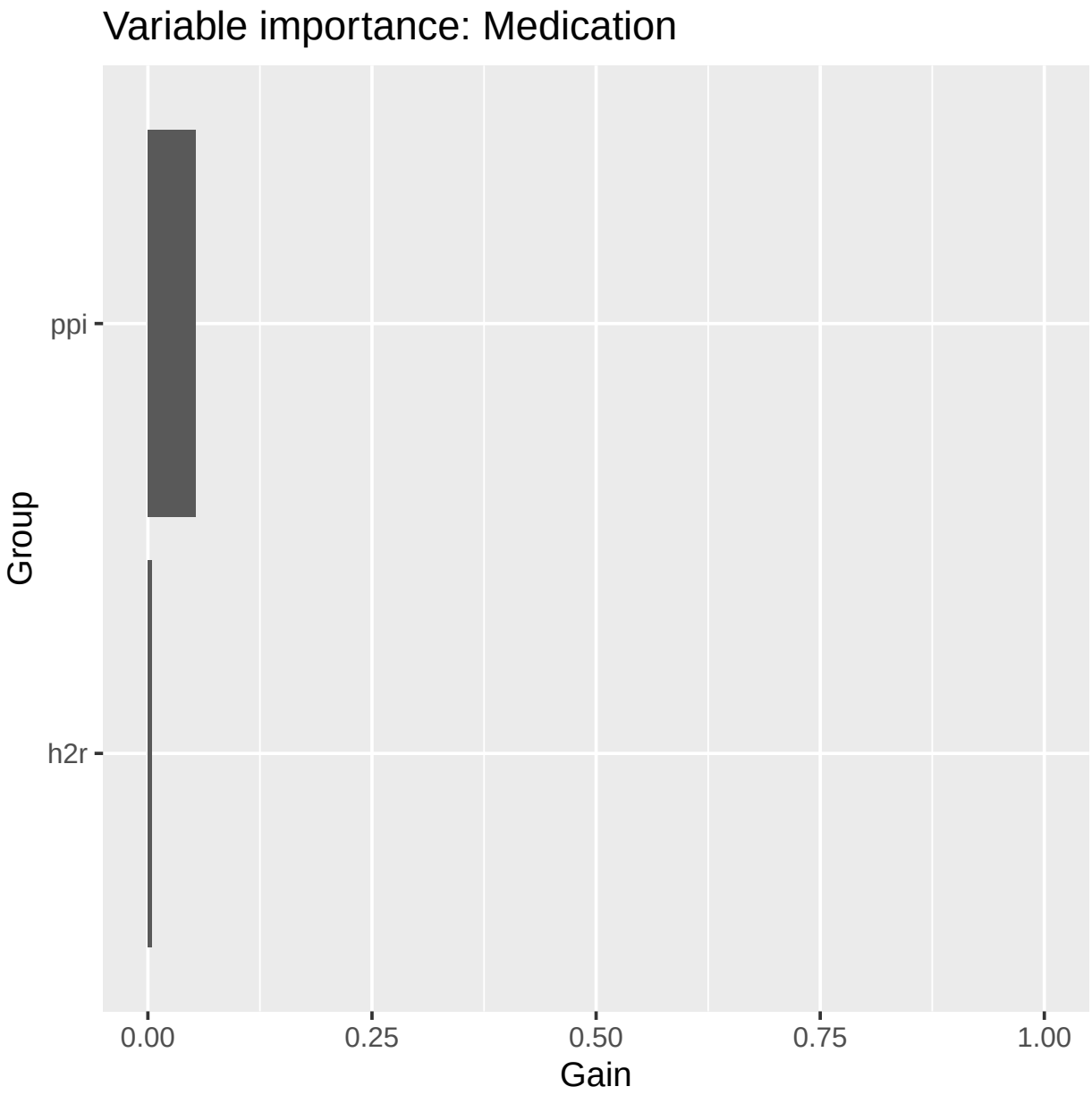

ROC curve (AUC=0.901)
Sensitivity
1 − Specificity
0.00
0.25
0.50
0.75
1.00
Threshold
(/100,000)
10000
1000
100
10
1

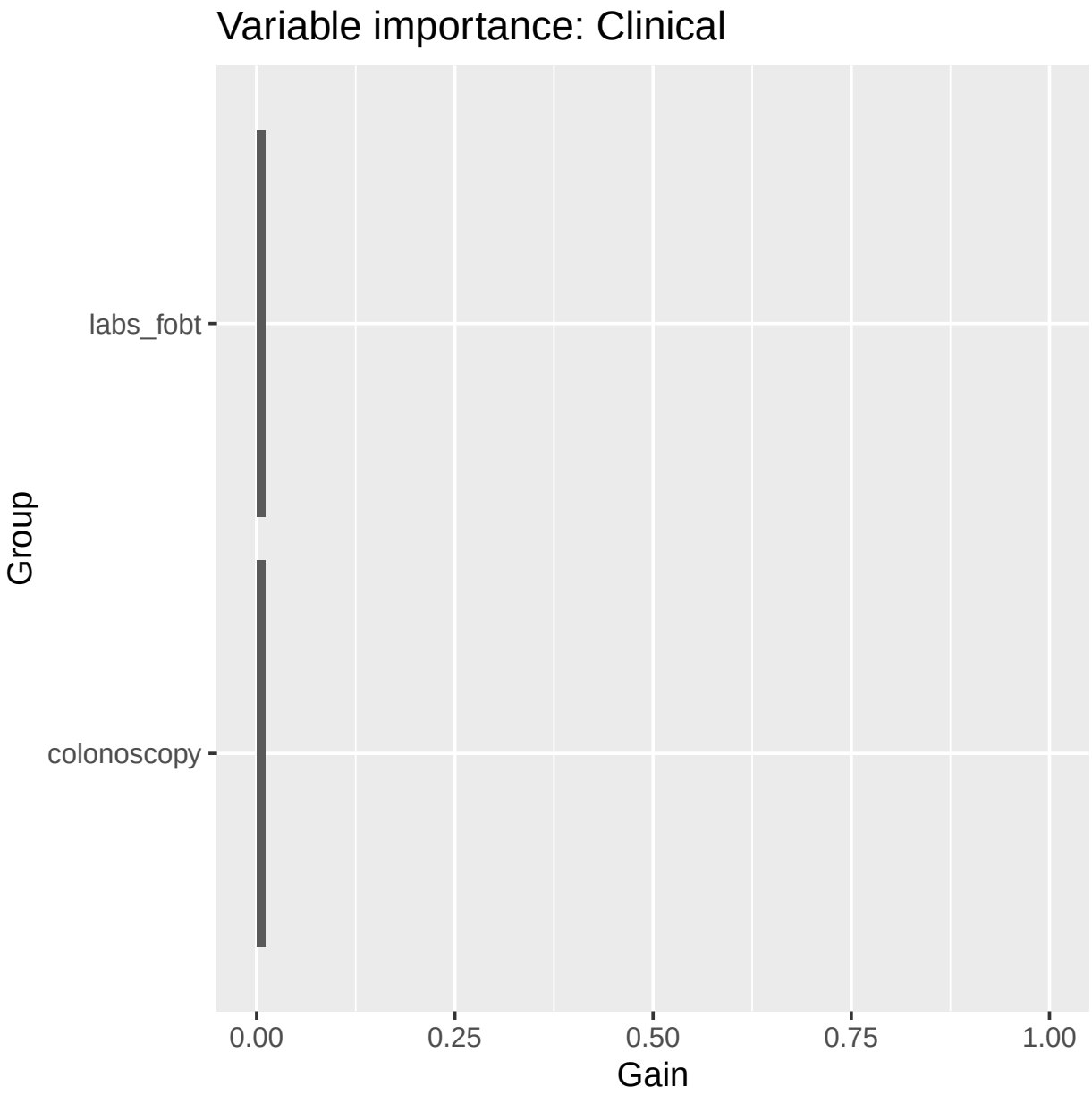

Calibration
Observed (/100,000)
Predicted (/100,000)
0
100
200
300

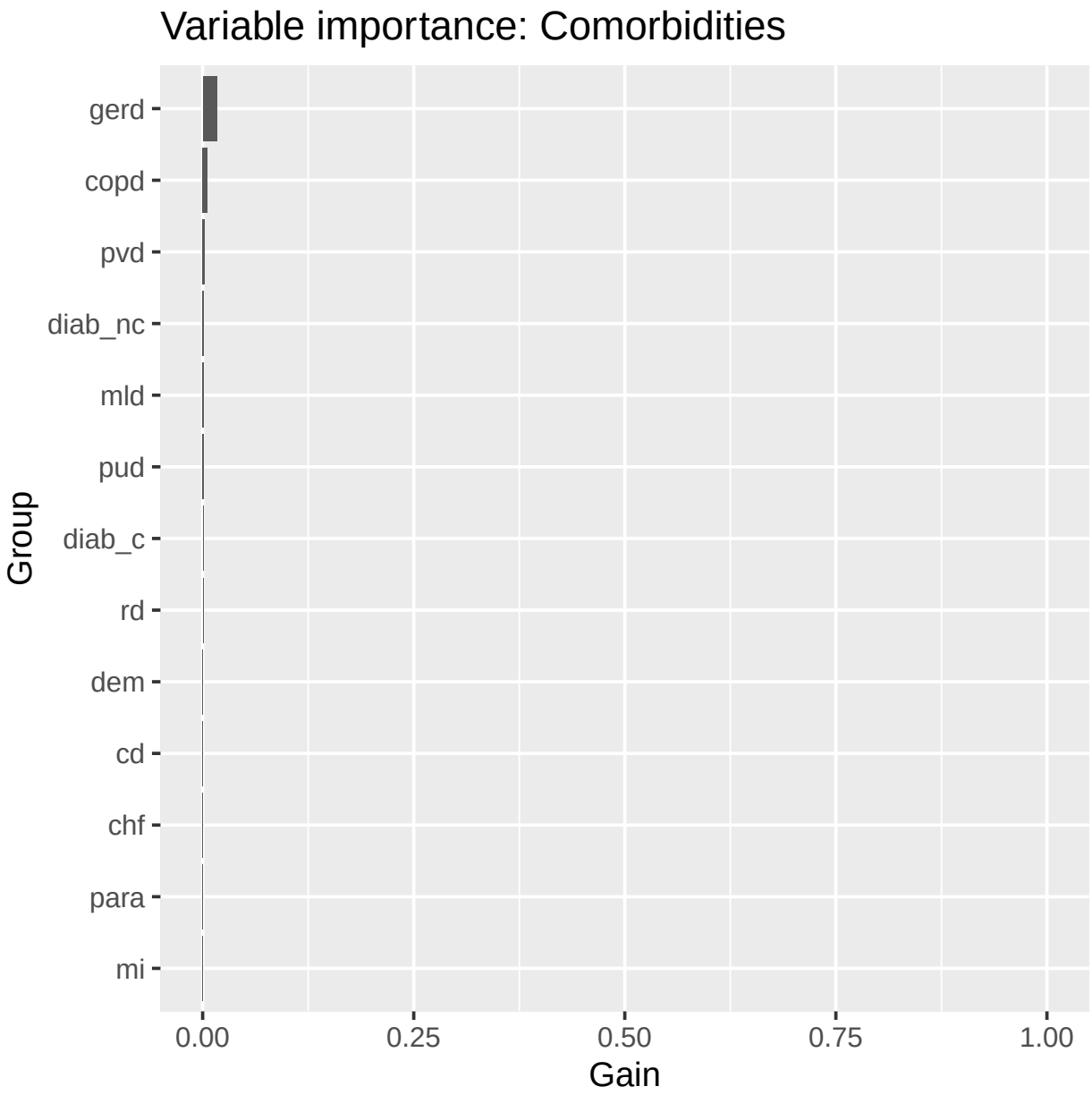

Calibration (log−log)
Observed (/100,000)
Predicted (/100,000)
1
10
100
1000
10000

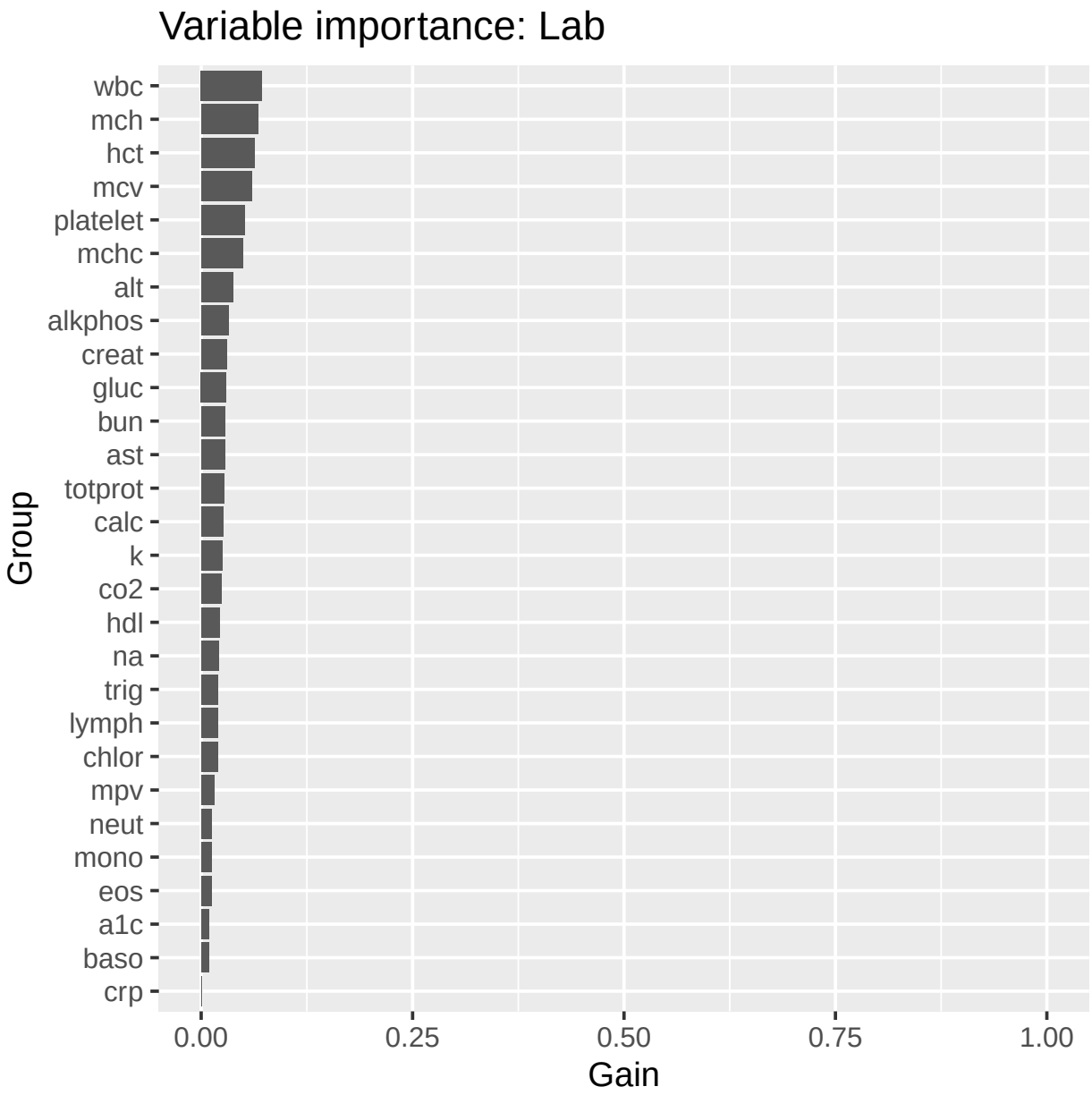

Variable importance by category
Category
Lab
Demographic
Medication
Comorbidities
0.00
0.25
0.50
0.75
1.00
Gain

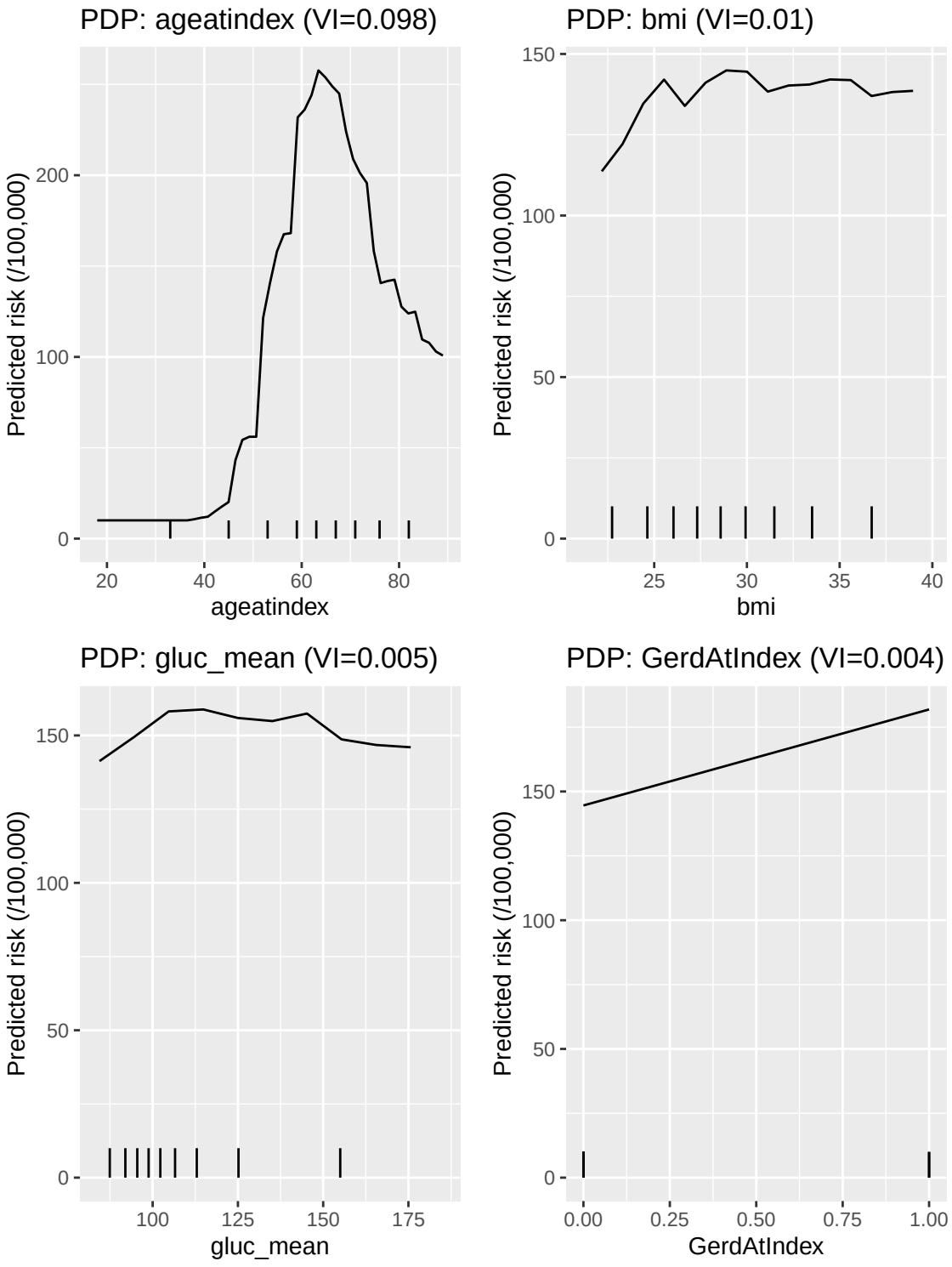

Variable importance: Demographic
Group
age
race
gender
weight
bmi
smoke
agentorange
0.00
0.25
0.50
0.75
1.00
Gain

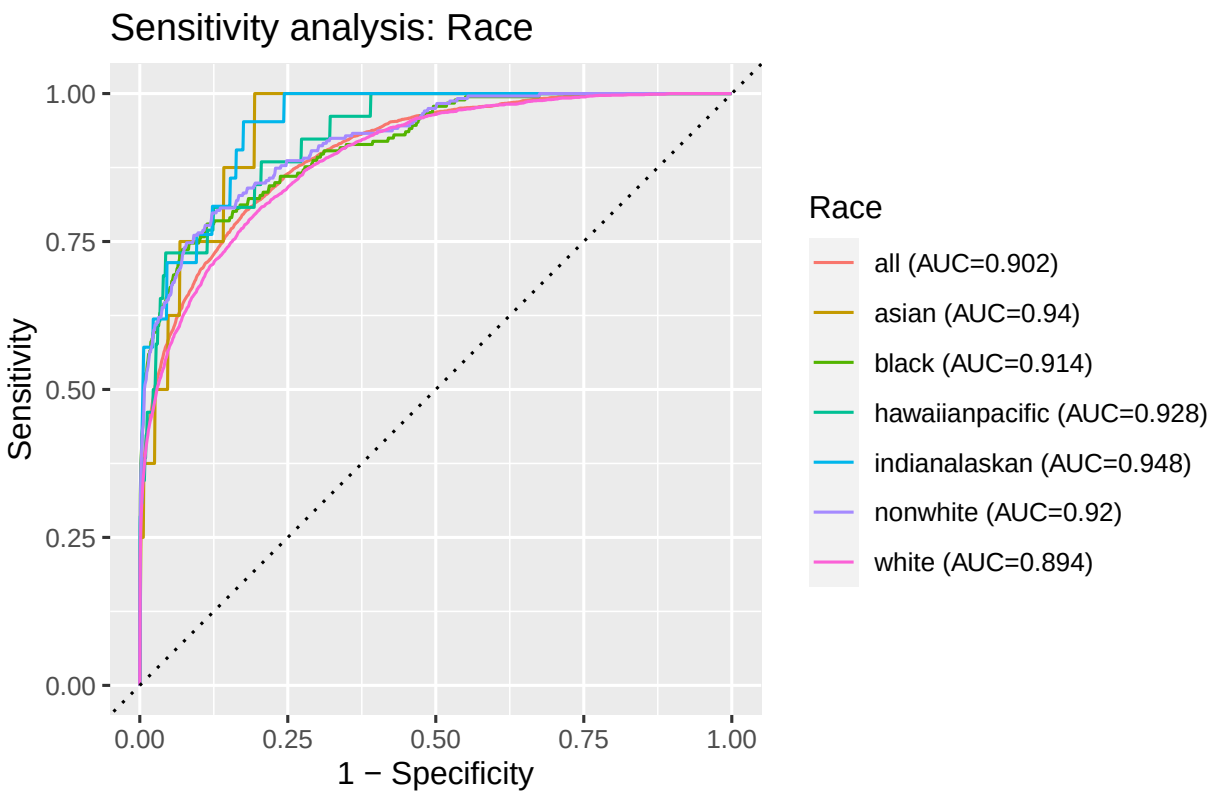

Variable importance: Medication
Group
ppi
h2r
0.00
0.25
0.50
0.75
1.00
Gain

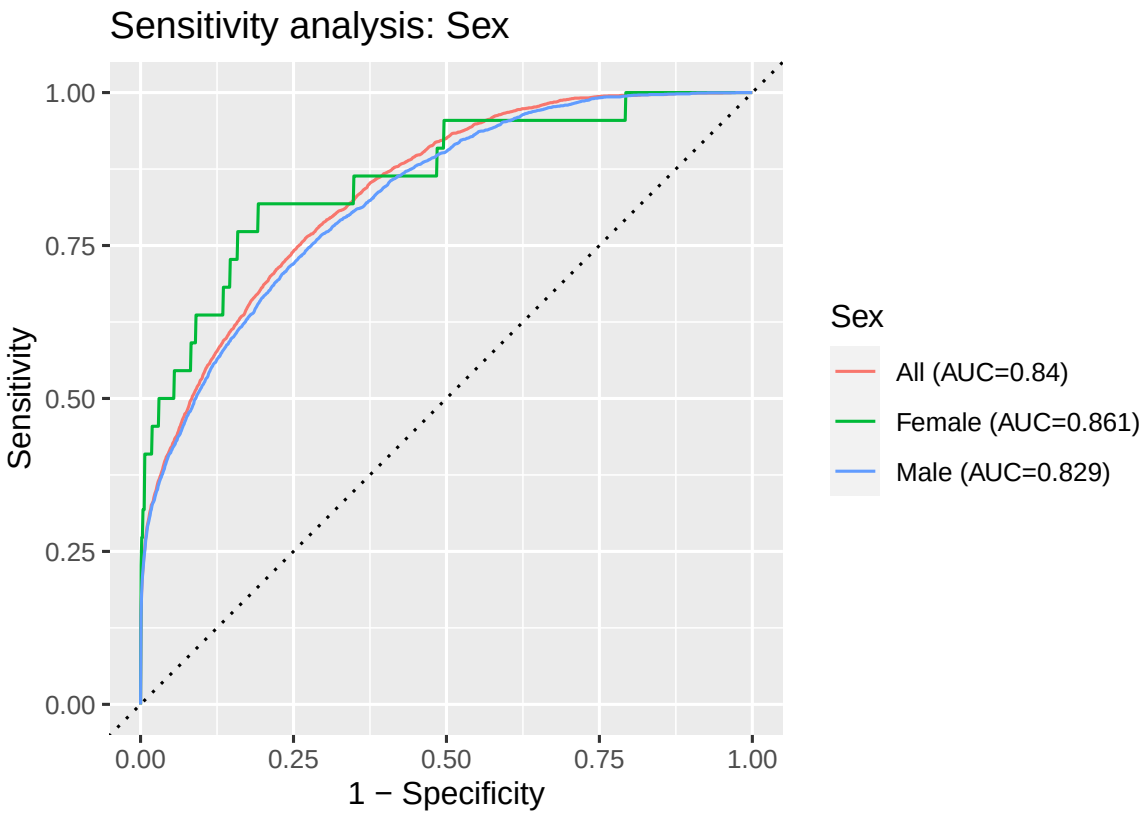

Variable importance: Clinical
Group
labs_fobt
colonoscopy
0.00
0.25
0.50
0.75
1.00
Gain


Variable importance: Comorbidities
gerd
copd
pvd
diab_nc
mld
pud
diab_c
rd
dem
cd
chf
para
mi
Group
0.00
0.25
0.50
0.75
1.00
Gain


Variable importance: Lab
wbc
mch
hct
mcv
platelet
mchc
alt
alkphos
creat
gluc
bun
ast
totprot
calc
k
co2
hdl
na
trig
lymph
chlor
mpv
neut
mono
eos
a1c
baso
crp
Group
0.00
0.25
0.50
0.75
1.00
Gain

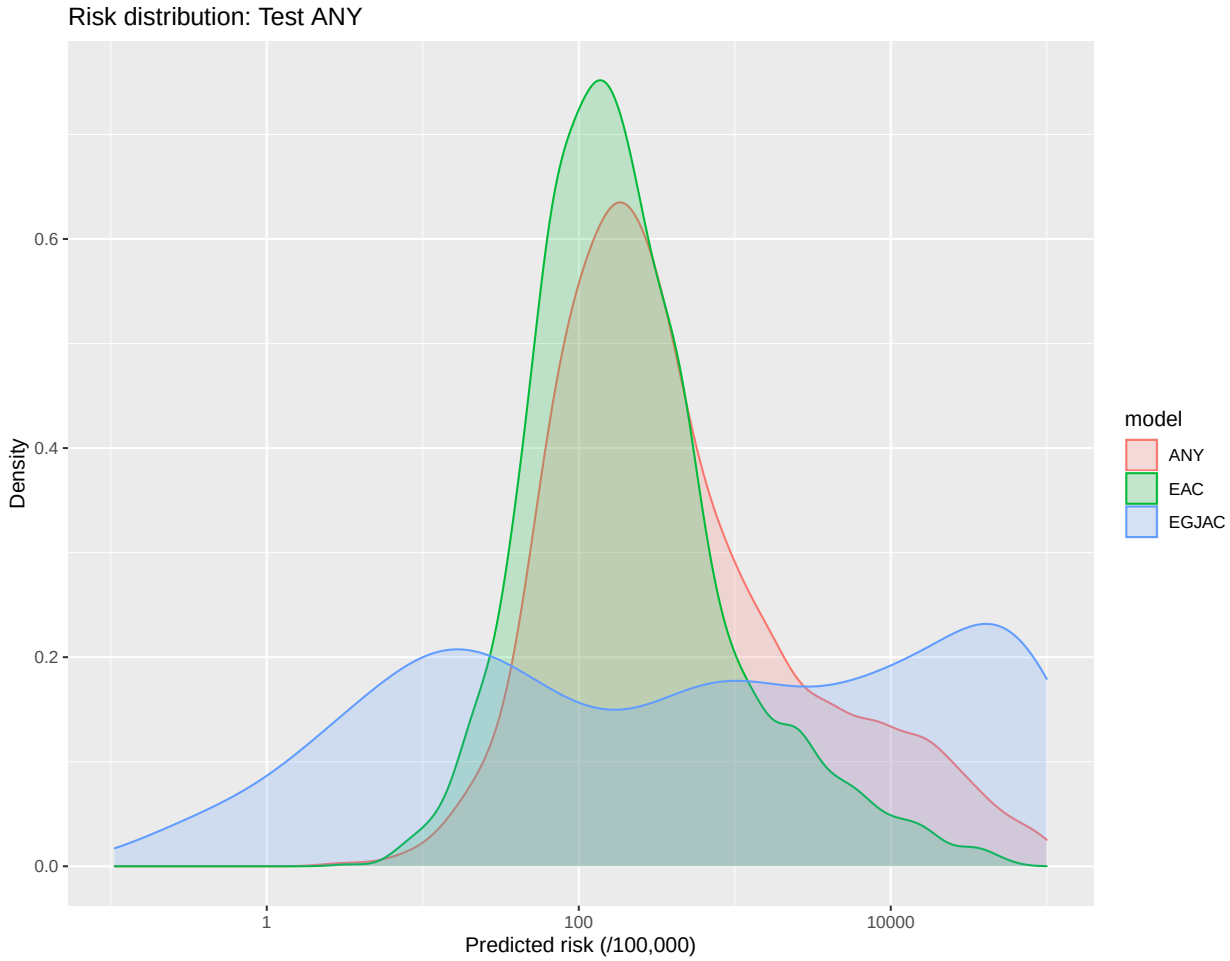

PDP: ageatindex (VI=0.098)
Predicted risk (/100,000)
0
100
200
20
40
60
80
ageatindex
PDP: bmi (VI=0.01)
Predicted risk (/100,000)
0
50
100
150
25
30
35
40
bmi
PDP: gluc_mean (VI=0.005)
Predicted risk (/100,000)
0
50
100
150
100
125
150
175
gluc_mean
PDP: GerdAtIndex (VI=0.004)
Predicted risk (/100,000)
0
50
100
150
0.00
0.25
0.50
0.75
1.00
GerdAtIndex


Sensitivity analysis: Race
Sensitivity
1 − Specificity
0.00
0.25
0.50
0.75
1.00
Race
all (AUC=0.902)
asian (AUC=0.94)
black (AUC=0.914)
hawaiianpacific (AUC=0.928)
indianalaskan (AUC=0.948)
nonwhite (AUC=0.92)
white (AUC=0.894)


Sensitivity analysis: Sex
Sensitivity
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
1 − Specificity
Sex
All (AUC=0.84)
Female (AUC=0.861)
Male (AUC=0.829)

Test AUC

	Outcome		
Model	ANY	EAC	EGJAC
ANY	0.930	0.920	0.945
EAC	0.931	0.931	0.907
EGJAC	0.927	0.808	0.958



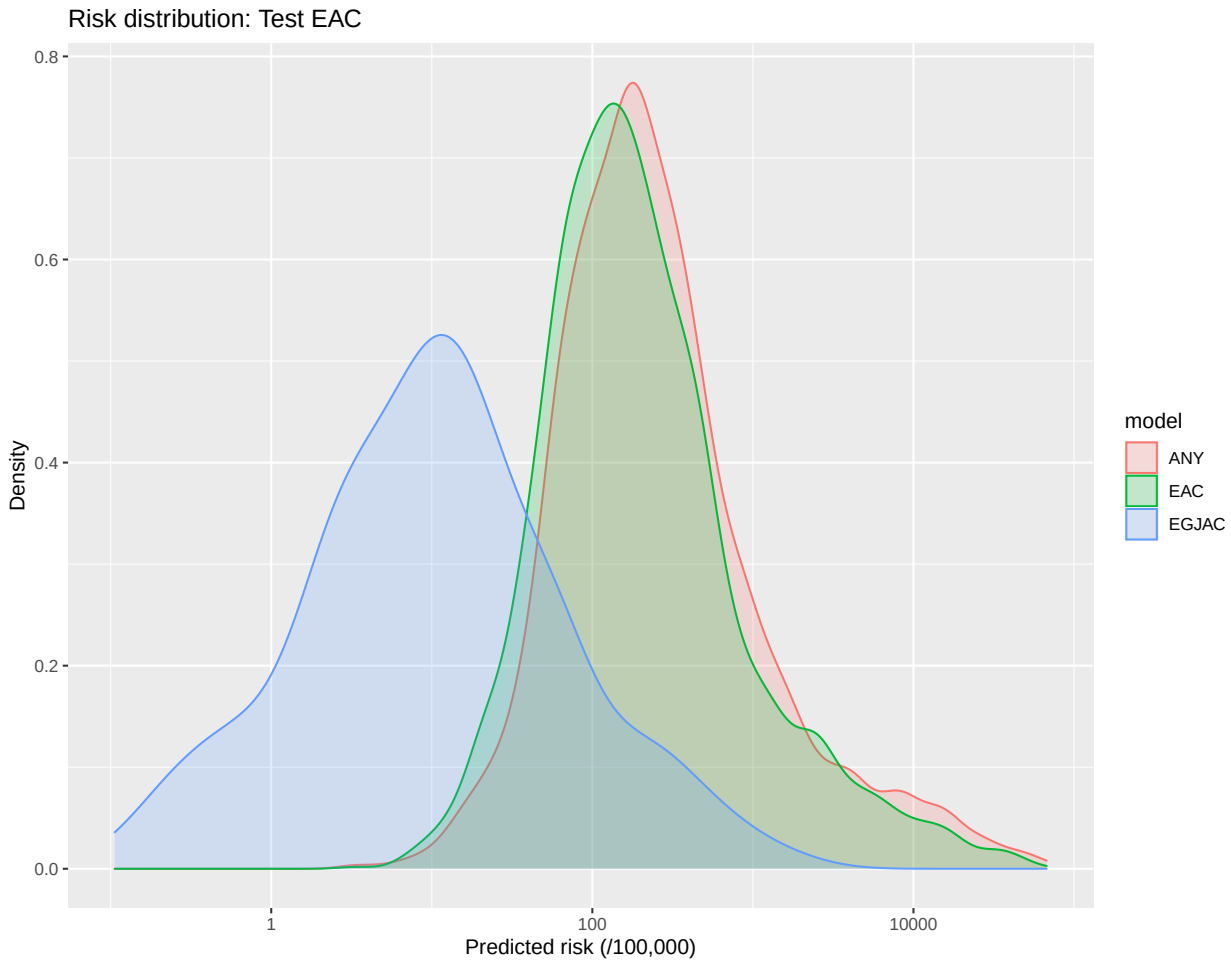


Figure 1: Features that predict EGJAC are not well adjusted to predict EAC

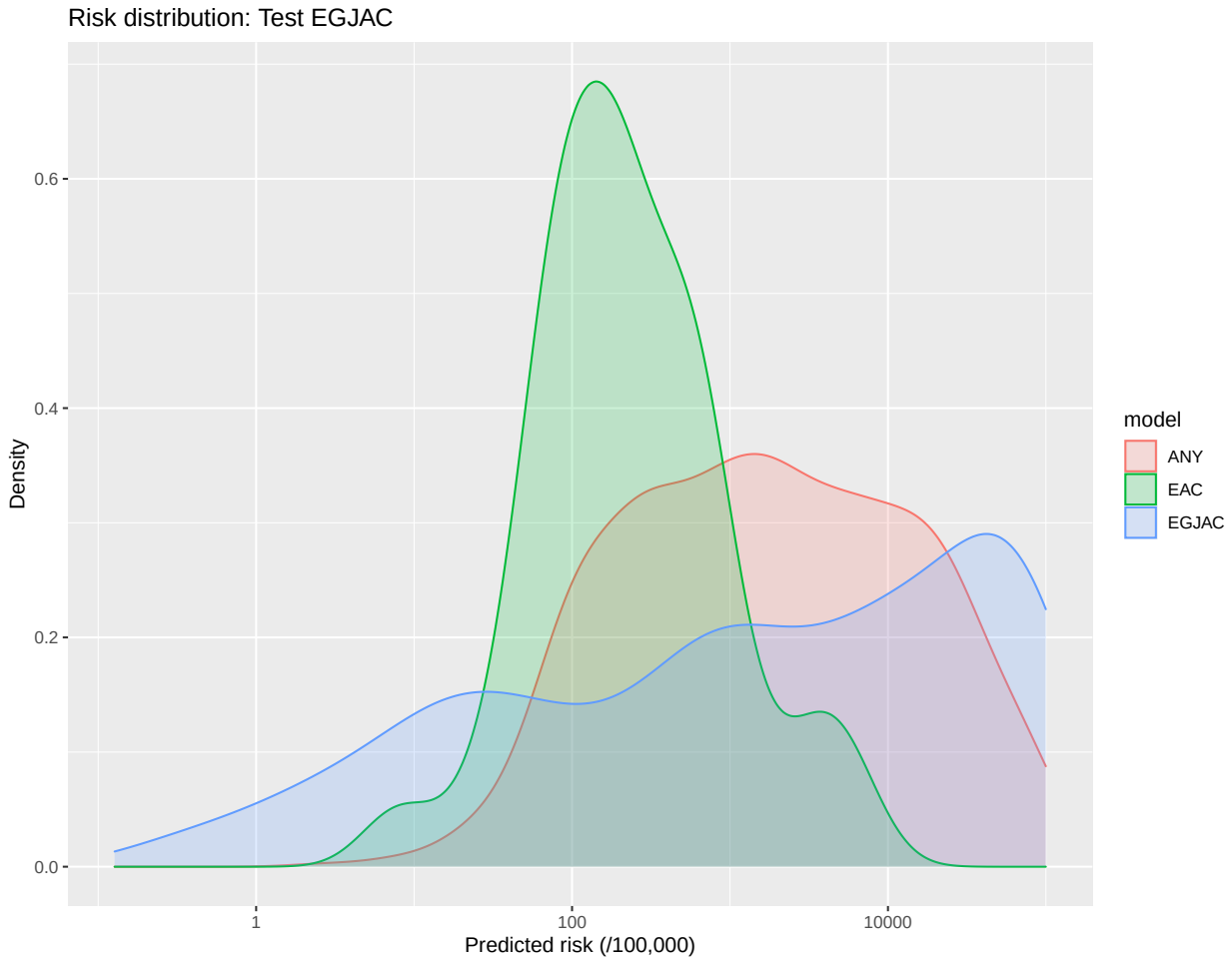
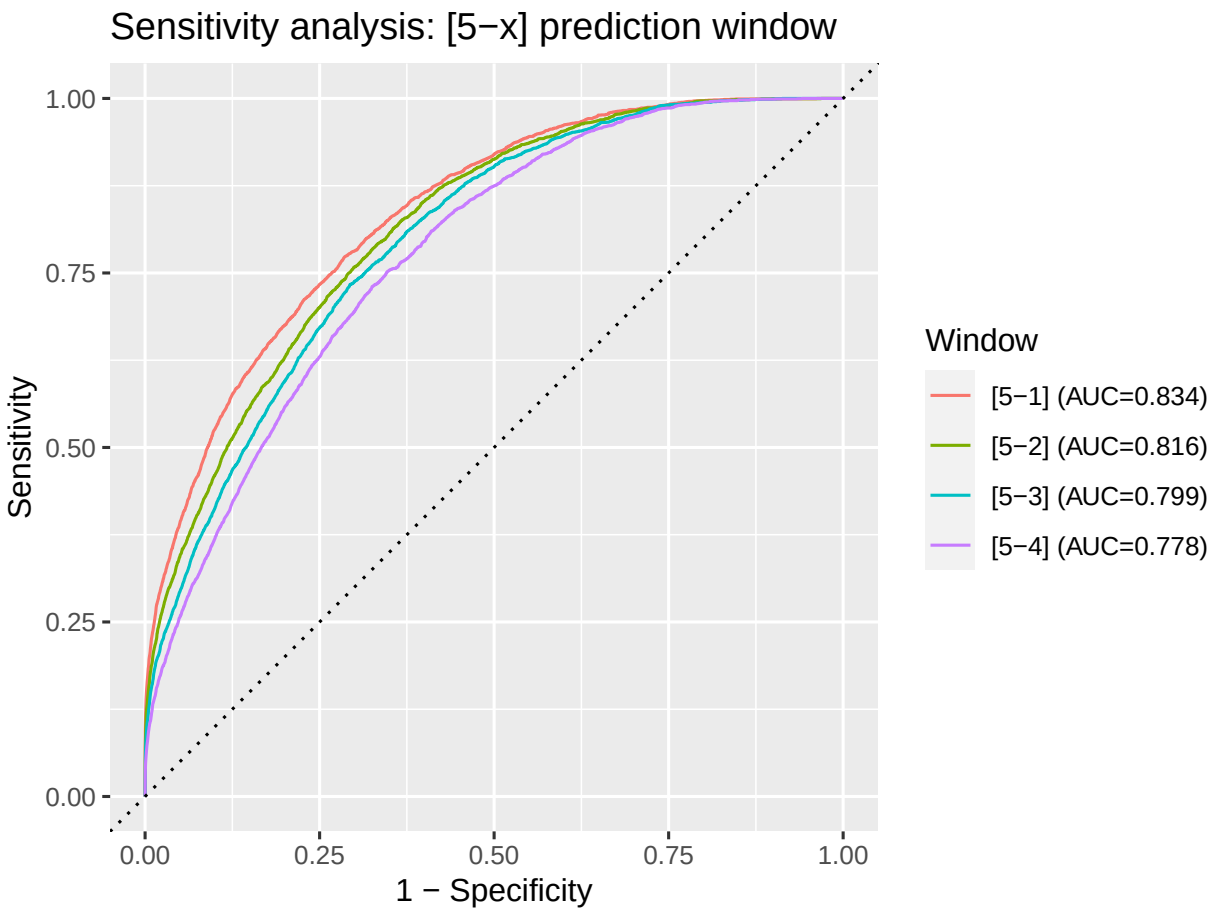
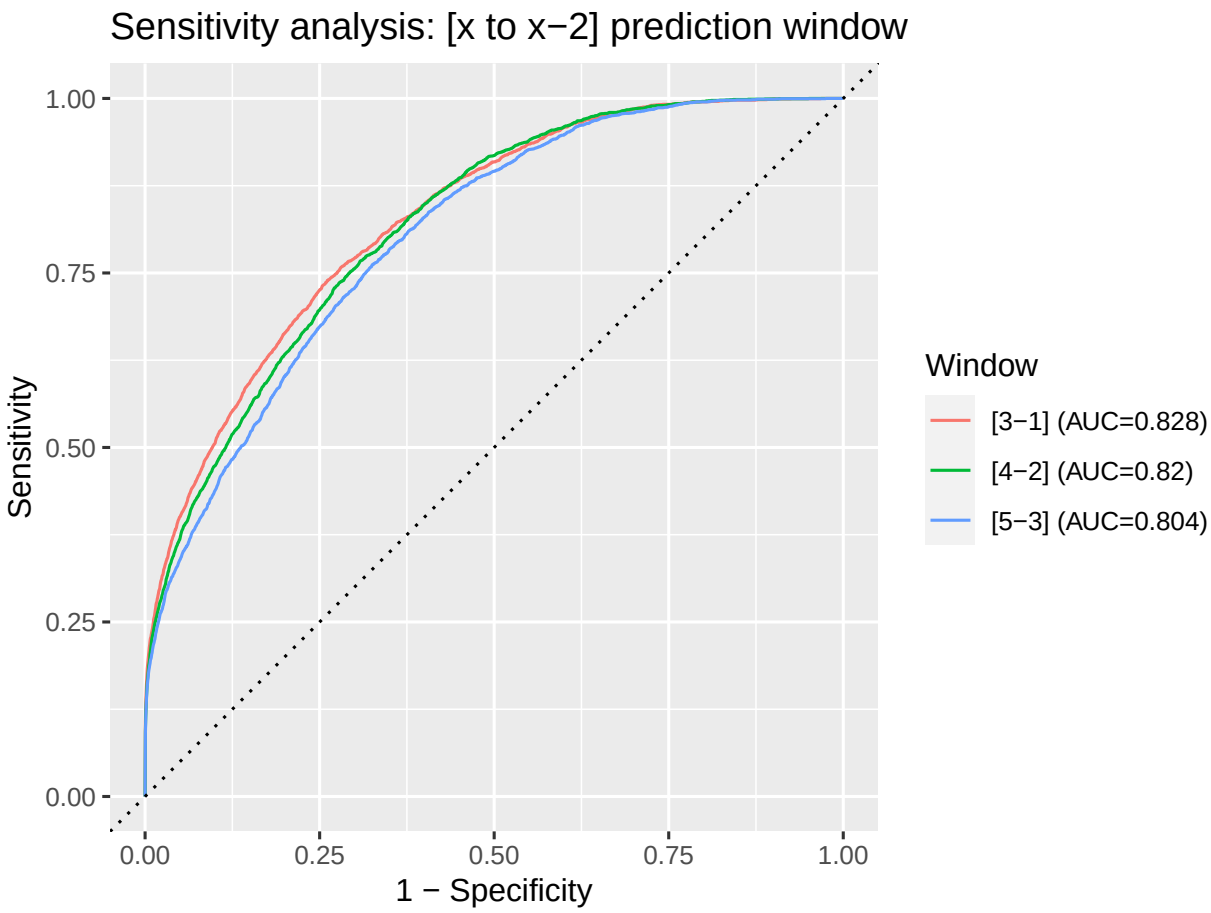


Figure 2: Features that predict EAC are not well adjusted to predict EGJAC (not as severe as the other way, though)


Sensitivity analysis: [5−x] prediction window
Sensitivity
1 − Specificity
0.00
0.25
0.50
0.75
1.00
Window
[5−1] (AUC=0.834)
[5−2] (AUC=0.816)
[5−3] (AUC=0.799)
[5−4] (AUC=0.778)


Sensitivity analysis: [x to x−2] prediction window
Sensitivity
1 − Specificity
0.00
0.25
0.50
0.75
1.00
Window
[3−1] (AUC=0.828)
[4−2] (AUC=0.82)
[5−3] (AUC=0.804)

03/17/2022 update

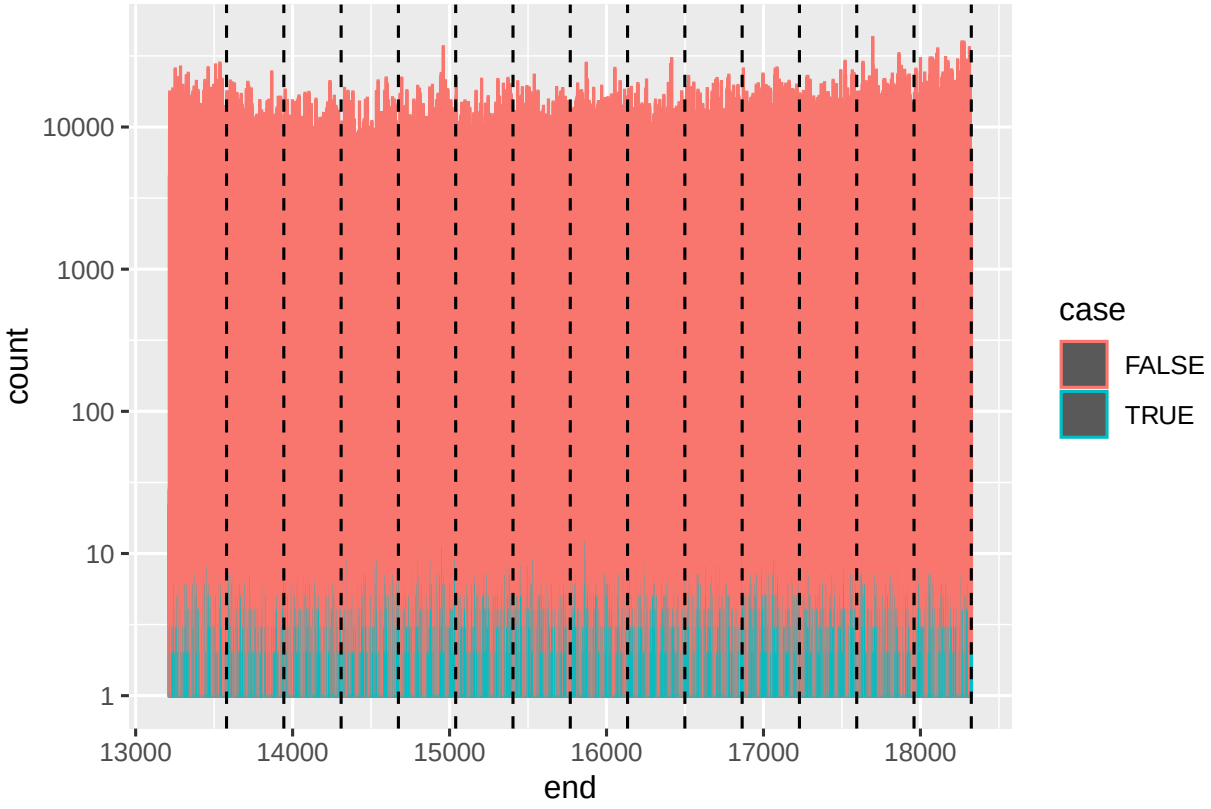


Figure 3: New data processed, everything looks good! No repeated entries; correct number of cases.

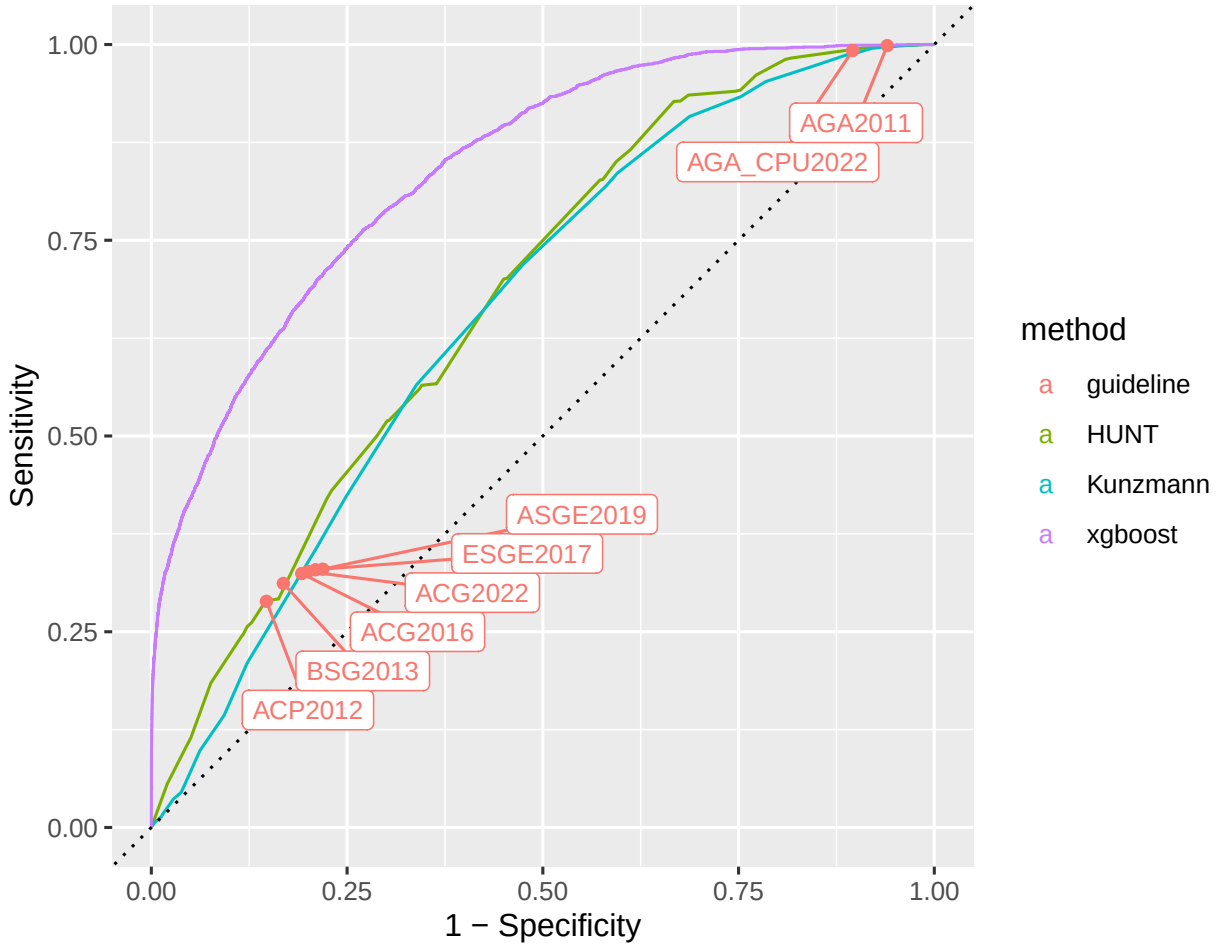


Figure 4: Similar performance as before, still outperforms Kunzmann, HUNT and guidelines by a large margin.

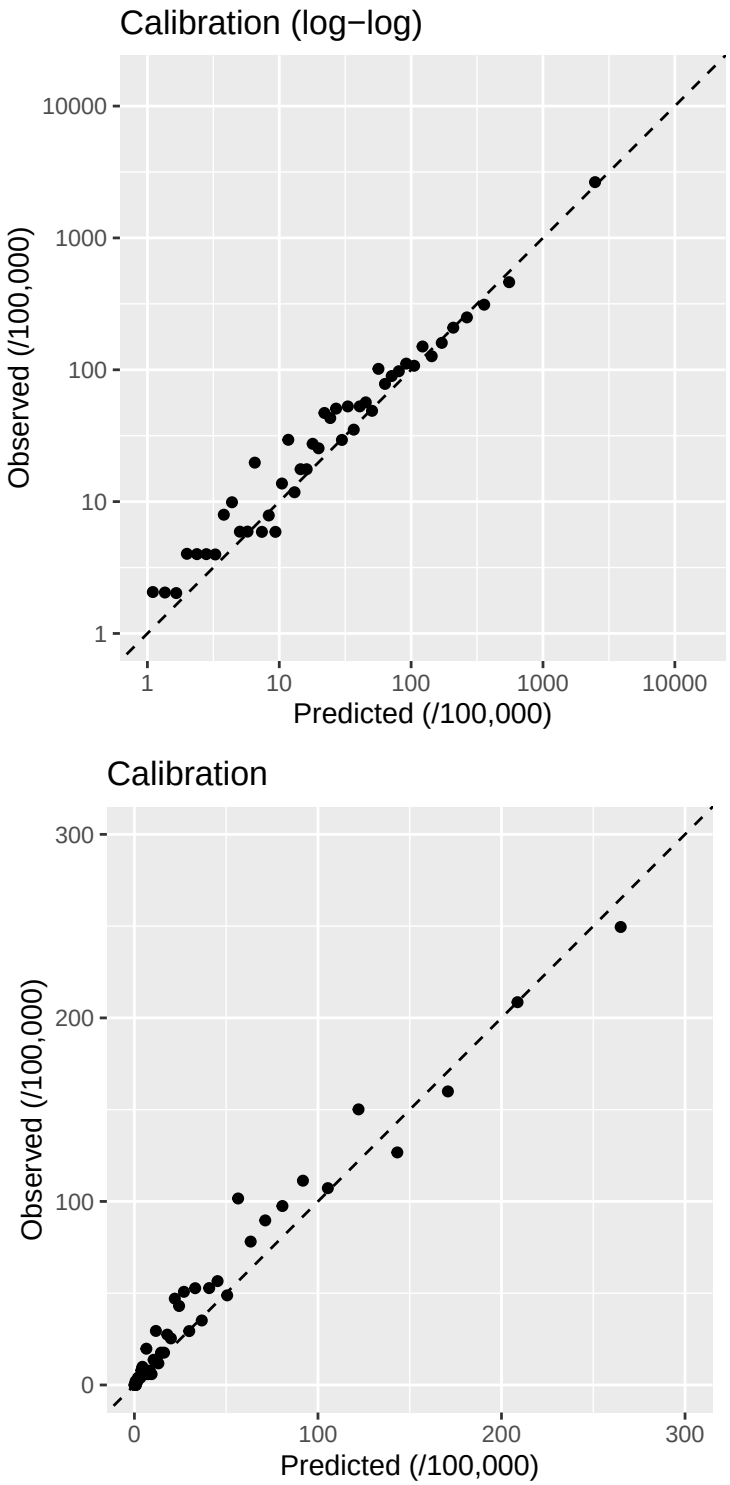
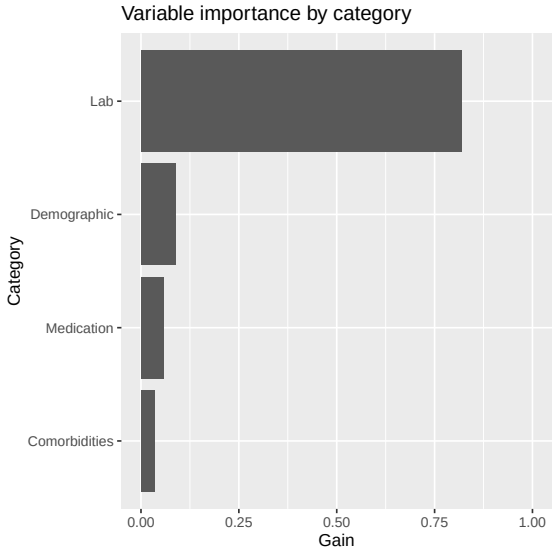
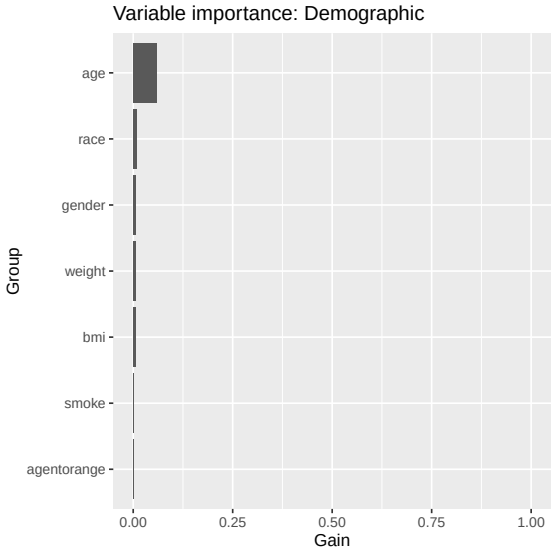
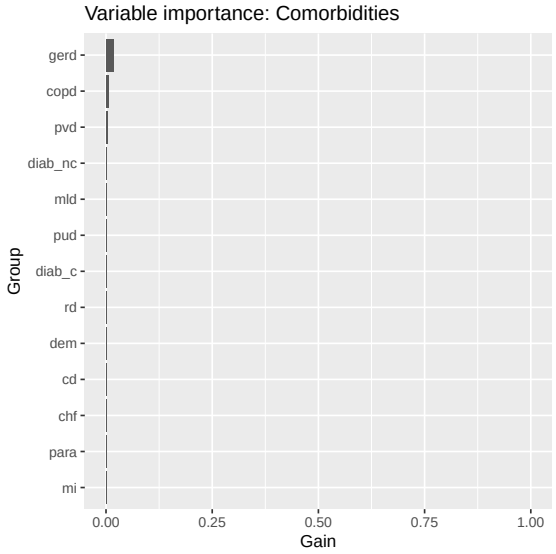
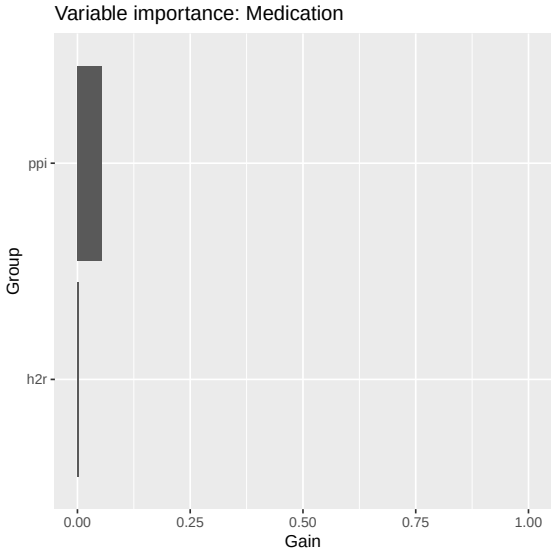


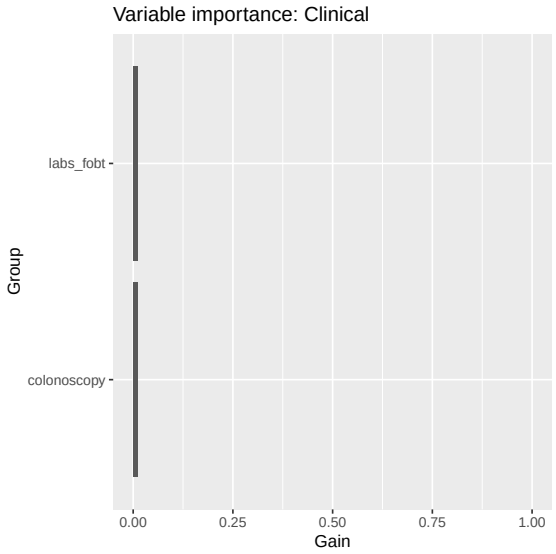
Figure 5: Calibration looks good, although it seems we might be slightly over-estimating risk above 100/100,000. In particular, it doesn't seem perfect for the target range 100-200.

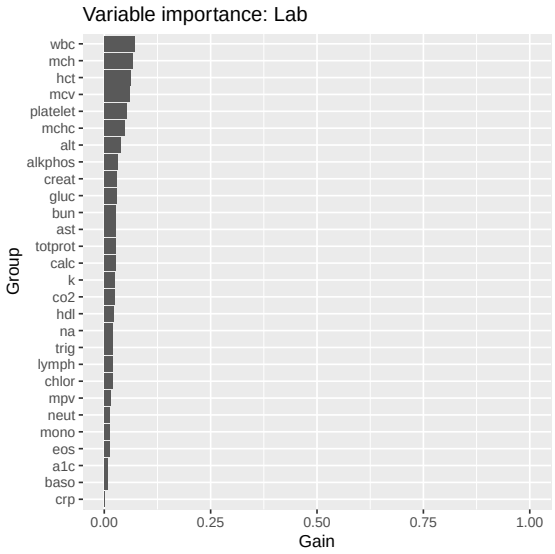

Variable importance by category
Lab
Demographic
Medication
Comorbidities
Category
0.00
0.25
0.50
0.75
1.00
Gain


Variable importance: Demographic
age
race
gender
weight
bmi
smoke
agentorange
Group
0.00
0.25
0.50
0.75
1.00
Gain


Variable importance: Comorbidities
gerd
copd
pvd
diab_nc
mld
pud
diab_c
rd
dem
cd
chf
para
mi
Group
0.00
0.25
0.50
0.75
1.00
Gain


Variable importance: Medication
ppi
h2r
Group
0.00
0.25
0.50
0.75
1.00
Gain


Variable importance: Clinical
labs_fobt
colonoscopy
Group
0.00
0.25
0.50
0.75
1.00
Gain


Variable importance: Lab
wbc
mch
hct
mcv
platelet
mchc
alt
alkphos
creat
gluc
bun
ast
totprot
calc
k
co2
hdl
na
trig
lymph
chlor
mpv
neut
mono
eos
a1c
baso
crp
Group
0.00
0.25
0.50
0.75
1.00
Gain

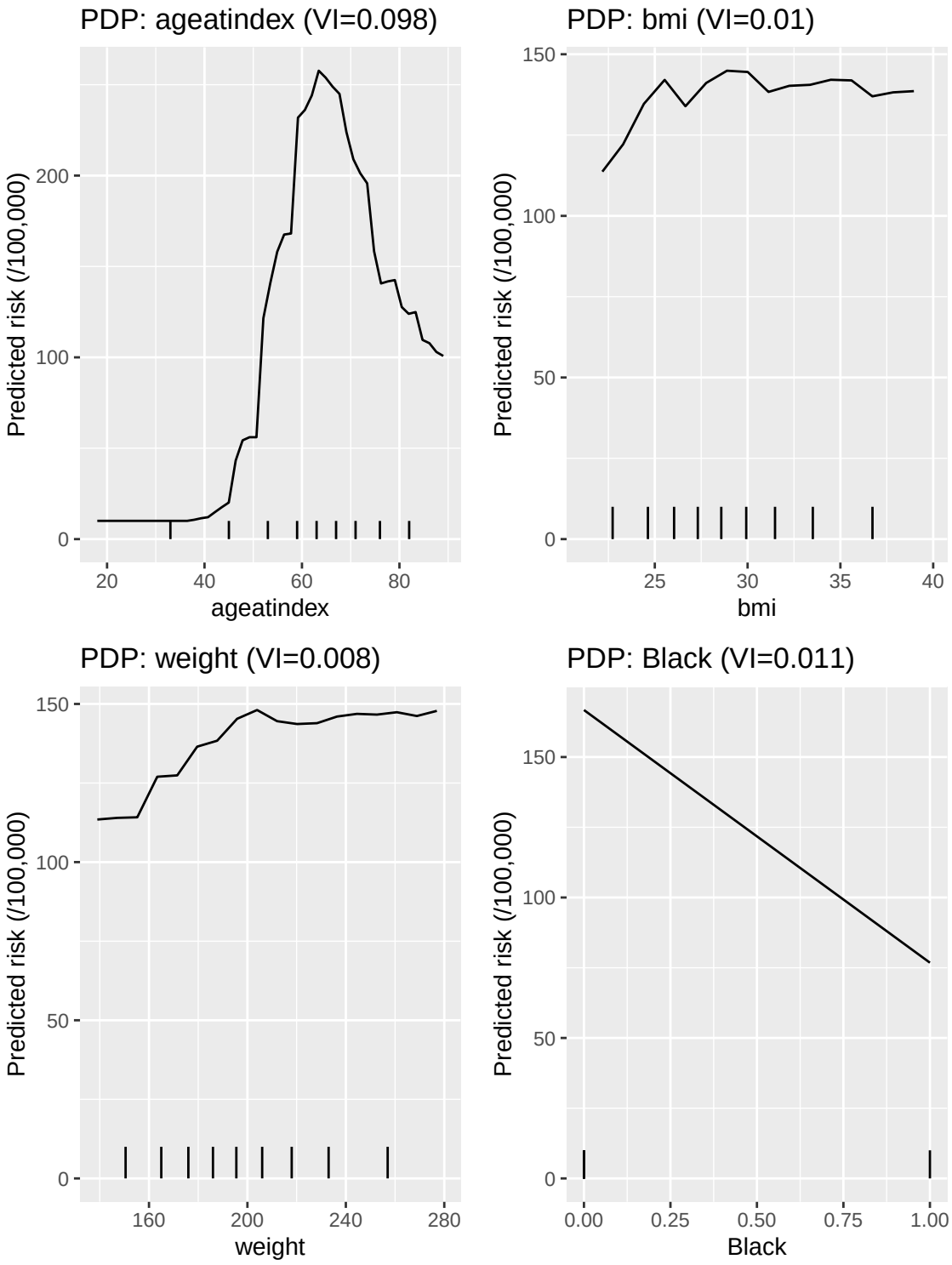
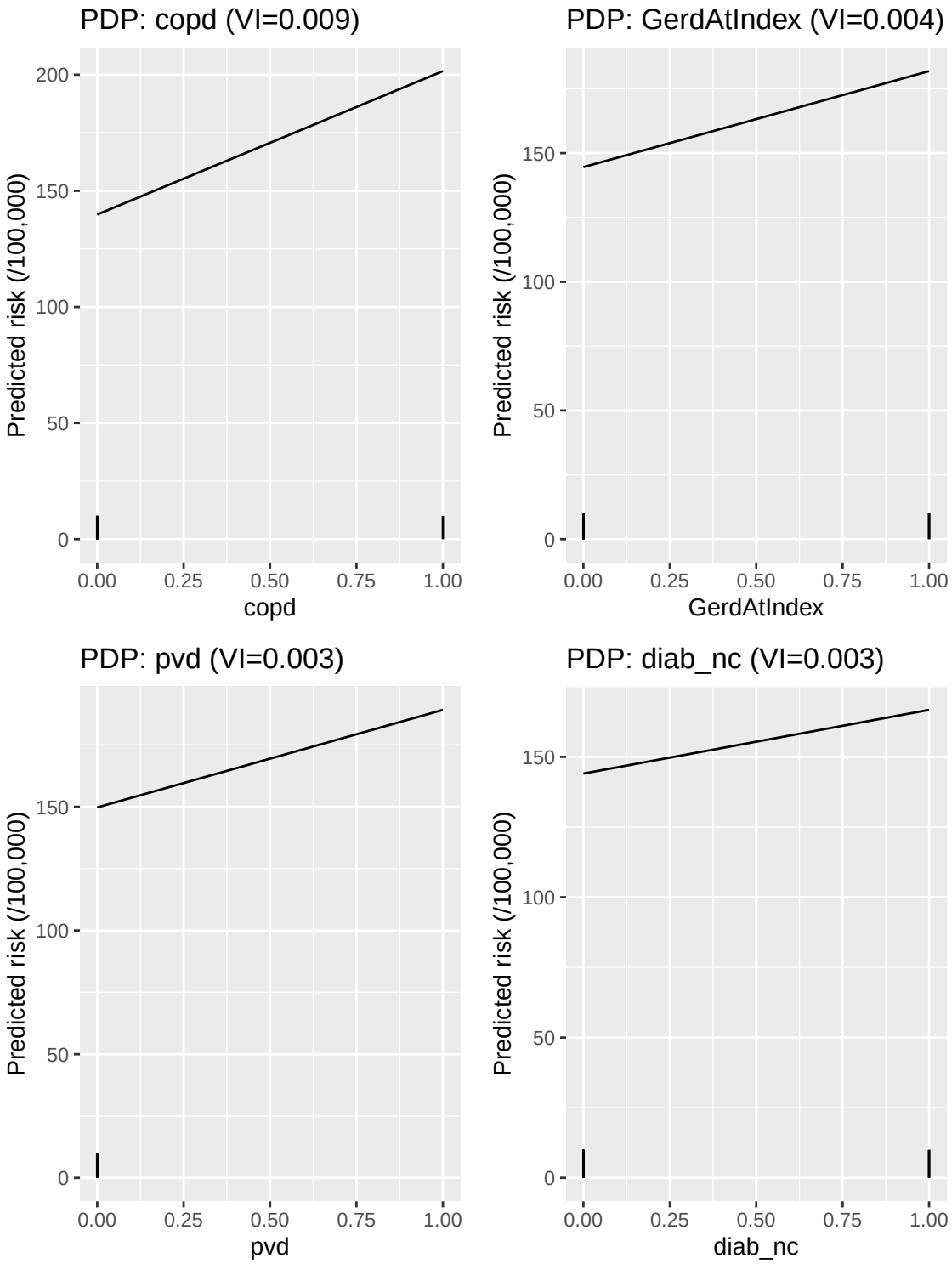
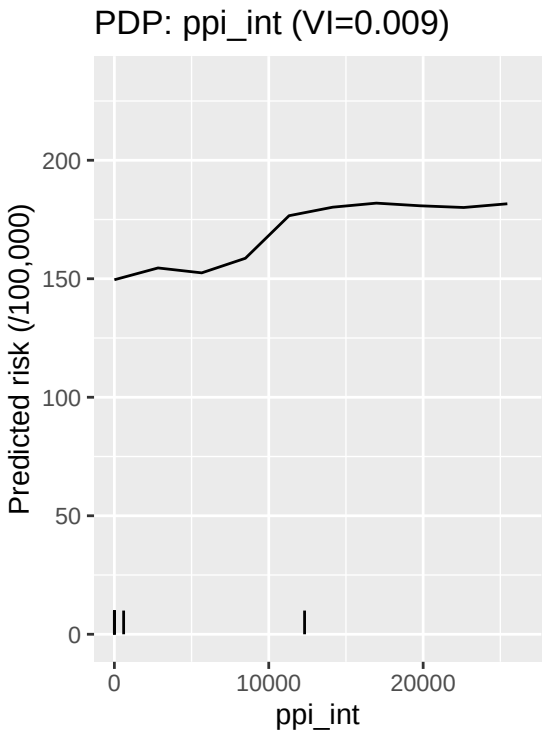


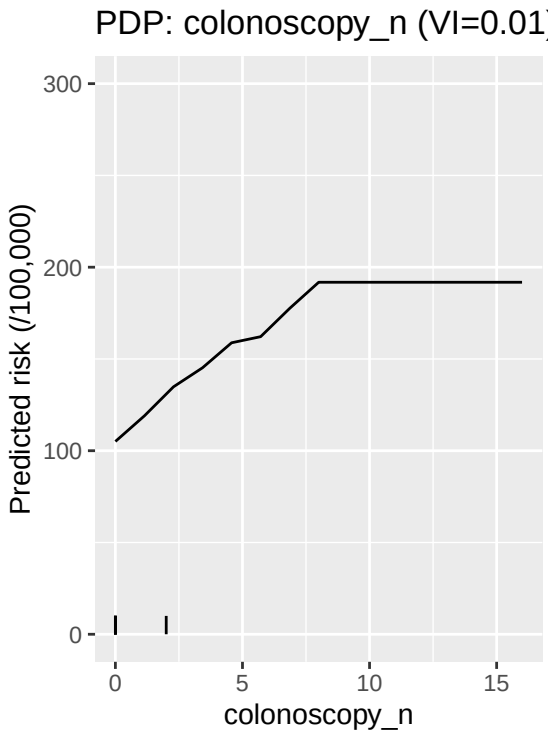
Figure 6: I spoke with Peter, and I think I know why weight/BMI seem off.

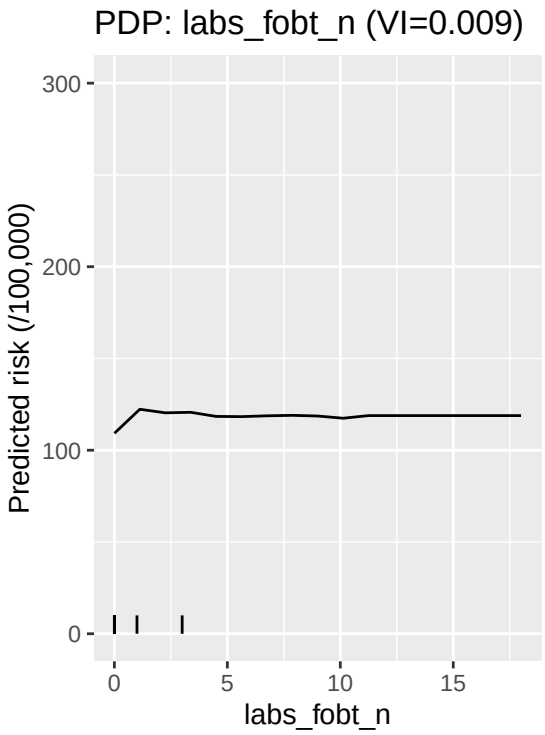
BMI	(0,20]	(20,25]	(25,30]	(30,35]	(35,40]	(40,100]
Control	99.55	99.79	99.86	99.87	99.87	99.87
Case	0.45	0.21	0.14	0.13	0.13	0.13

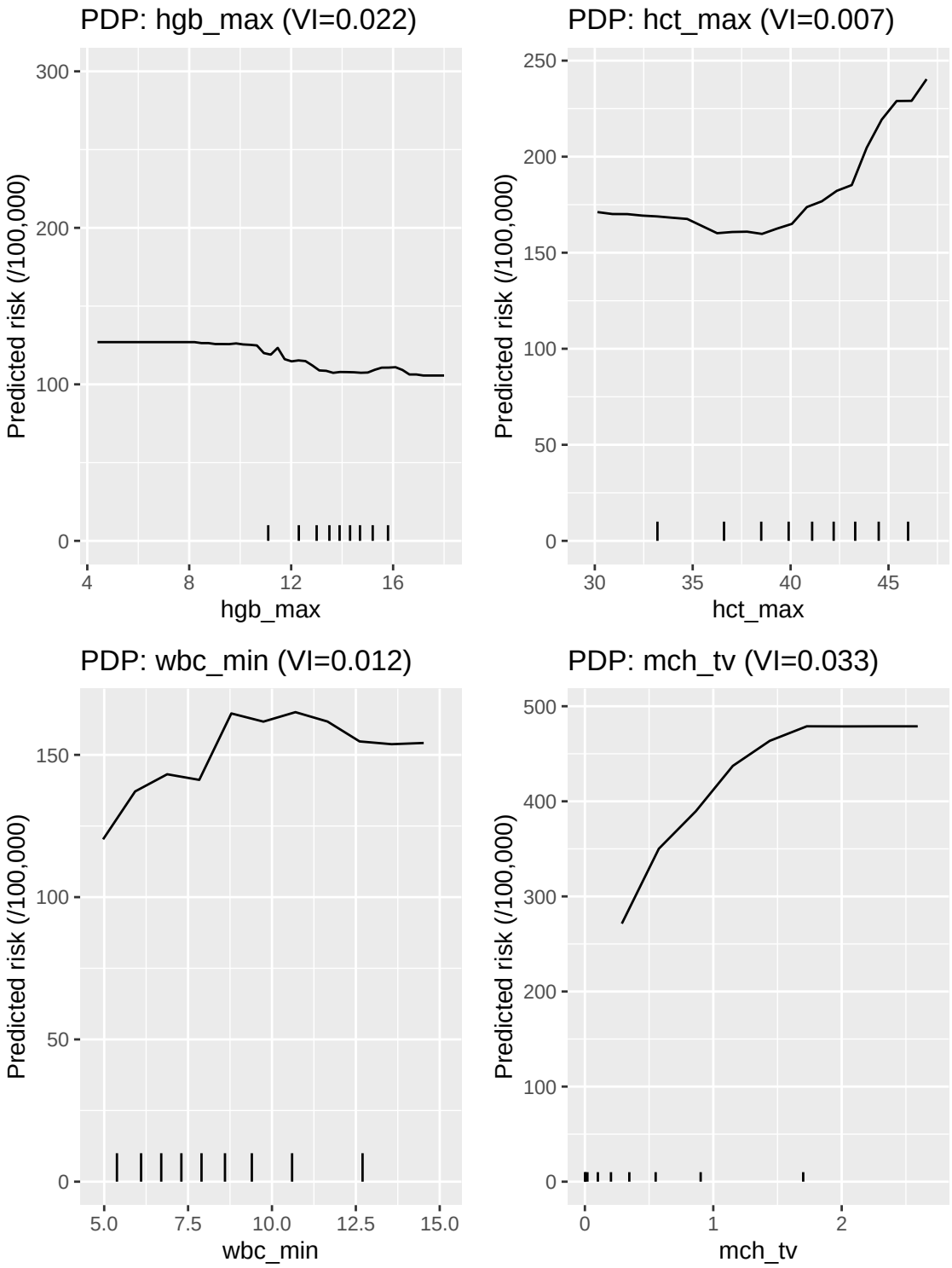
Table 1: Proportion of cases/controls stratified by BMI. Looking at earlier analyses by Peter (e.g., p.11), it seems we have the same issue. I discussed with Peter and he told me that was a result of BMI being collect at the index date, not 1 year prior. Could we check at what time BMI was collected for the new data? & same for weight. Can we also check that smoking status is 1 year prior? I think these are the only ones that I do not compute myself that could change over time.


PDP: copd (VI=0.009)
Predicted risk (/100,000)
200
150
100
50
0
0.00
0.25
0.50
0.75
1.00
copd
PDP: GerdAtIndex (VI=0.004)
Predicted risk (/100,000)
150
100
50
0
0.00
0.25
0.50
0.75
1.00
GerdAtIndex
PDP: pvd (VI=0.003)
Predicted risk (/100,000)
150
100
50
0
0.00
0.25
0.50
0.75
1.00
pvd
PDP: diab_nc (VI=0.003)
Predicted risk (/100,000)
150
100
50
0
0.00
0.25
0.50
0.75
1.00
diab_nc


PDP: ppi_int (VI=0.009)
Predicted risk (/100,000)
200
150
100
50
0
0
10000
20000
ppi_int

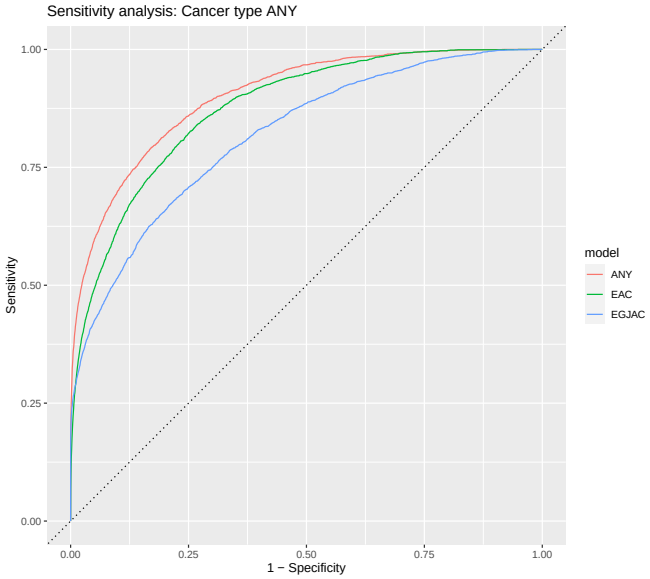

PDP: colonoscopy_n (VI=0.01)
Predicted risk (/100,000)
300
200
100
0
0
5
10
15
colonoscopy_n

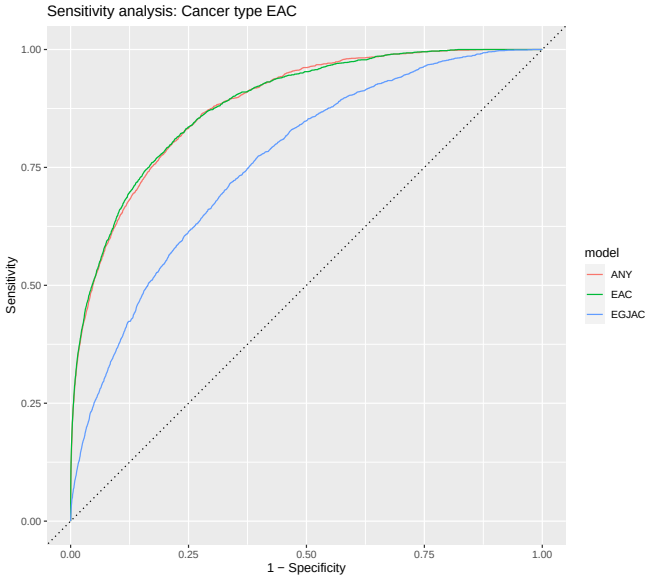

PDP: labs_fobt_n (VI=0.009)
Predicted risk (/100,000)
300
200
100
0
0
5
10
15
labs_fobt_n

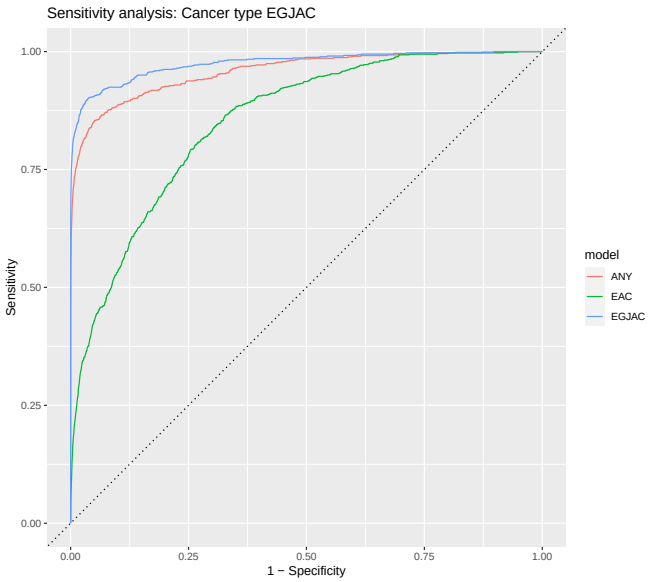

PDP: hgb_max (VI=0.022)
Predicted risk (/100,000)
hgb_max
PDP: hct_max (VI=0.007)
Predicted risk (/100,000)
hct_max
PDP: wbc_min (VI=0.012)
Predicted risk (/100,000)
wbc_min
PDP: mch_tv (VI=0.033)
Predicted risk (/100,000)
mch_tv

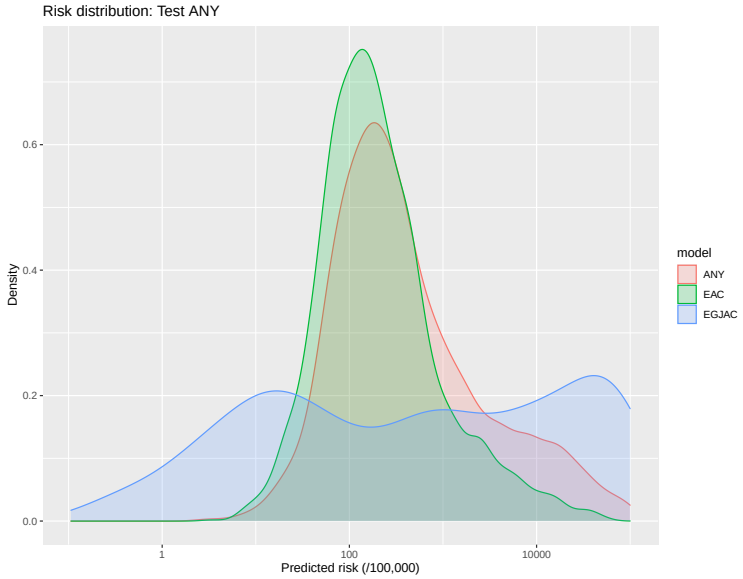
Test AUC			
	Testing		
Training	ANY	EAC	EGJAC
ANY	0.833	0.800	0.921
EAC	0.811	0.811	0.816
EGJAC	0.899	0.680	0.961

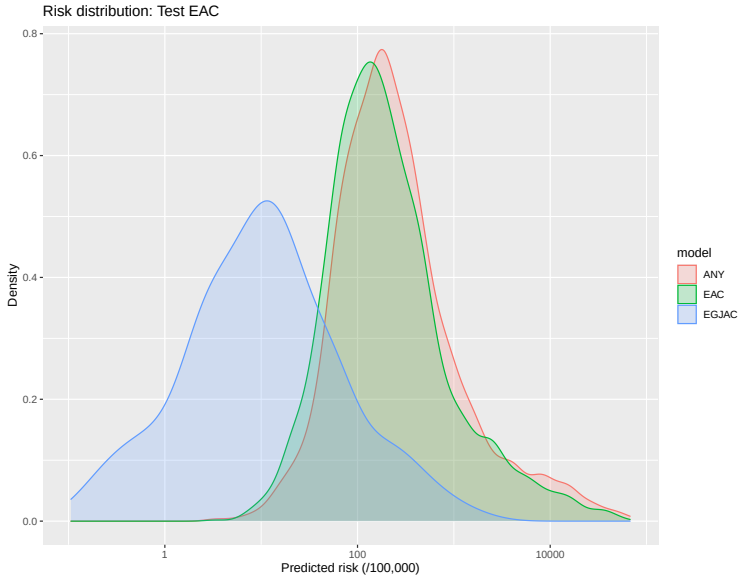
Table 2: Cancer type analysis: similar results as before, but, interestingly, it seems that training on EGJAC improves tesing on ANY? I need to check this again, it is suspicious. There is an issue with how the test sample is constructed, I think I will redo this analysis.

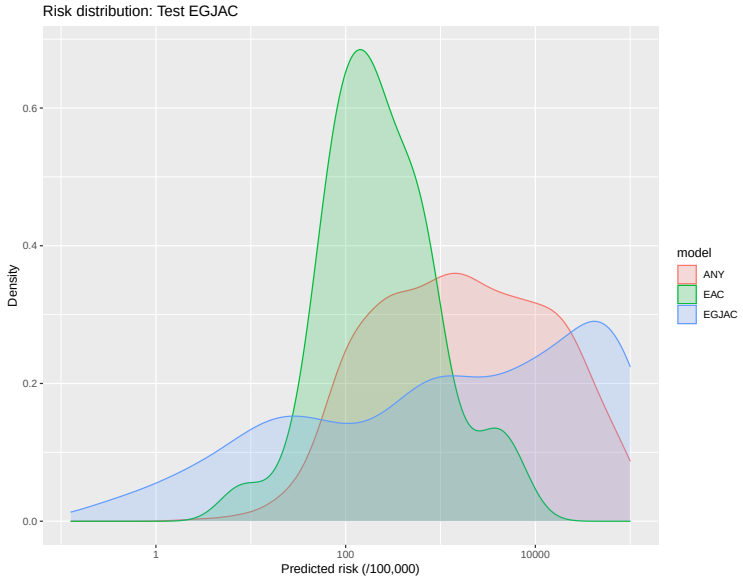

Sensitivity analysis: Cancer type ANY
Sensitivity
1 − Specificity
0.00
0.25
0.50
0.75
1.00
model
ANY
EAC
EGJAC


Sensitivity analysis: Cancer type EAC
Sensitivity
1 − Specificity
0.00
0.25
0.50
0.75
1.00
model
ANY
EAC
EGJAC


Sensitivity analysis: Cancer type EGJAC
Sensitivity
1 − Specificity
0.00
0.25
0.50
0.75
1.00
model
ANY
EAC
EGJAC


Risk distribution: Test ANY
Density
0.0
0.2
0.4
0.6
1
100
10000
Predicted risk (/100,000)
model
ANY
EAC
EGJAC


Risk distribution: Test EAC
Density
0.0
0.2
0.4
0.6
0.8
1
100
10000
Predicted risk (/100,000)
model
ANY
EAC
EGJAC


Risk distribution: Test EGJAC
Density
0.0
0.2
0.4
0.6
1
100
10000
Predicted risk (/100,000)
model
ANY
EAC
EGJAC

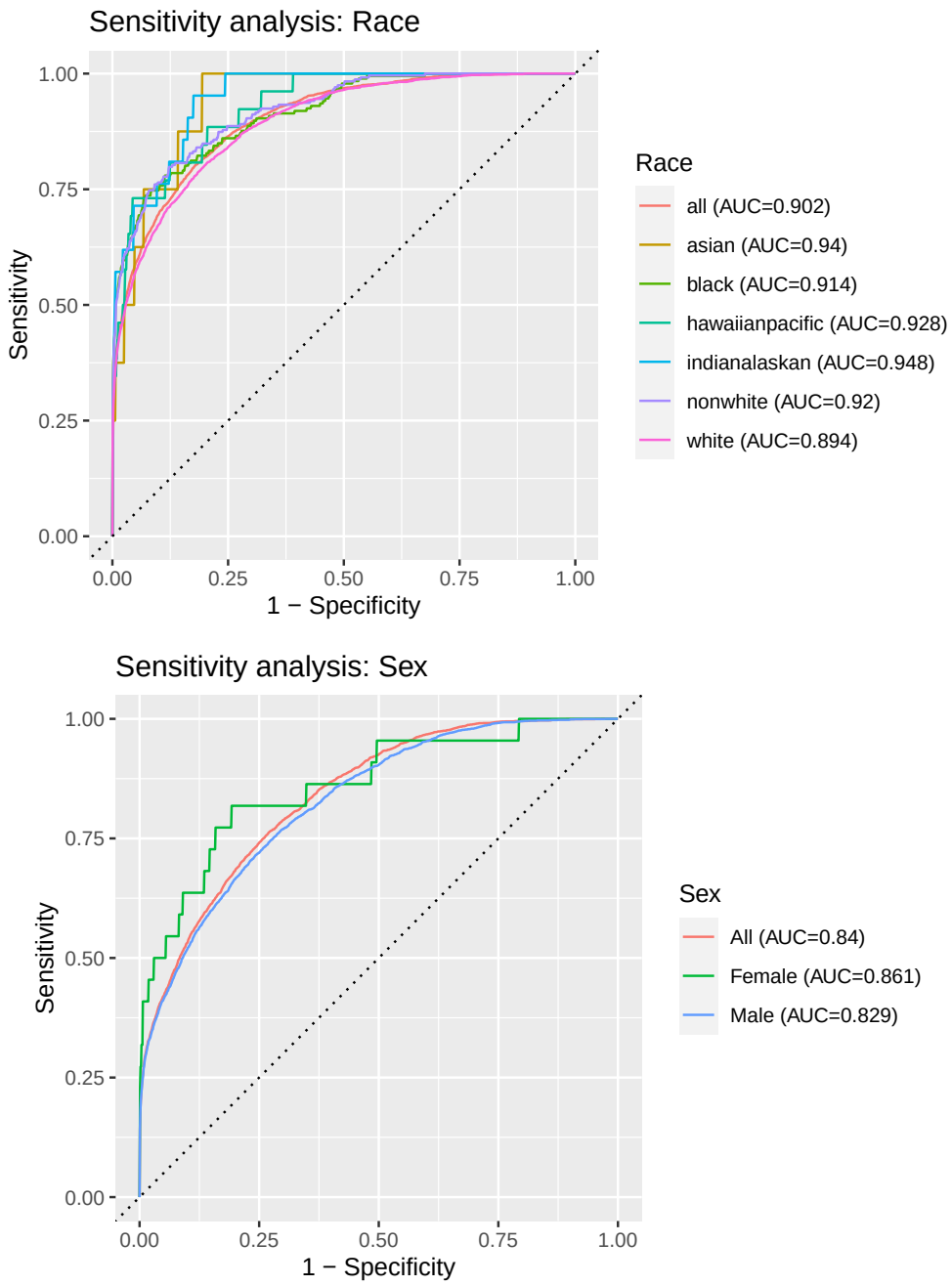
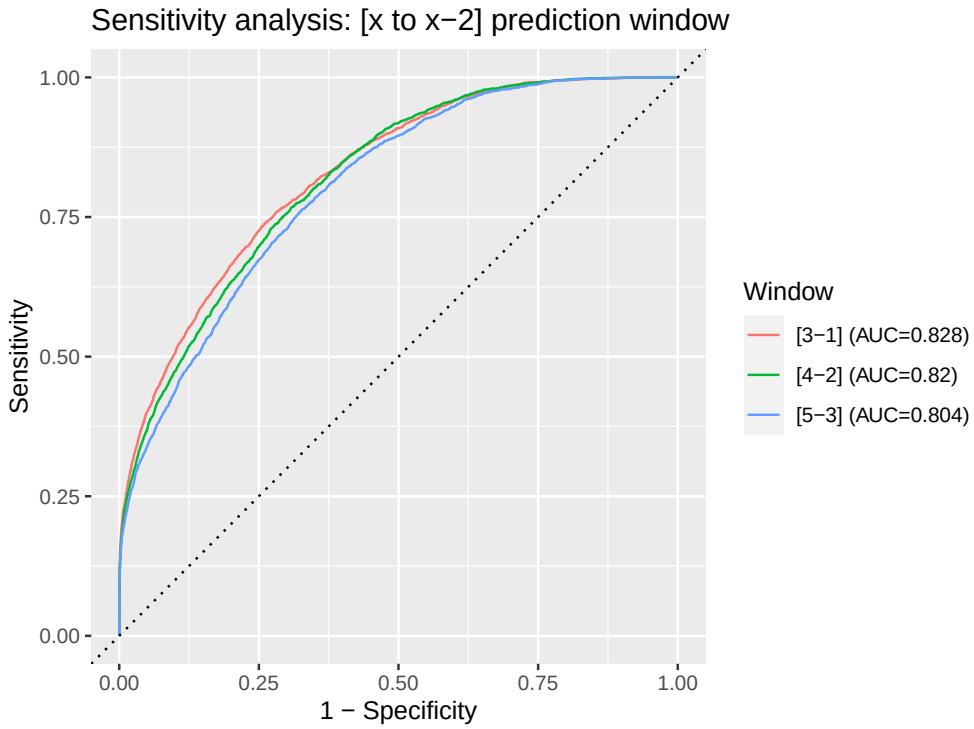
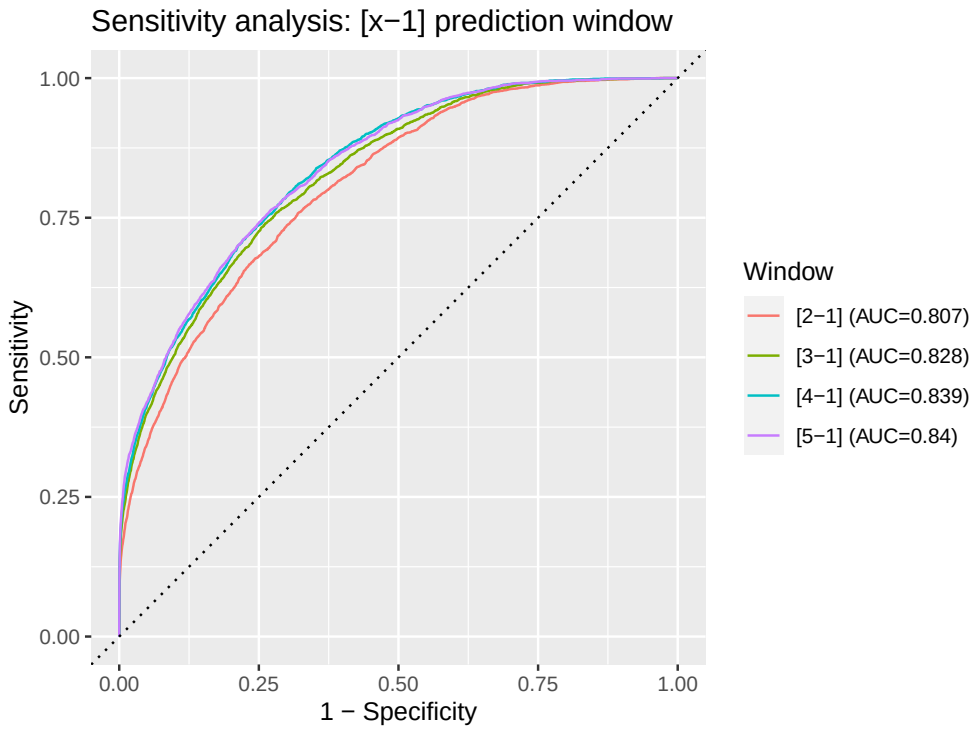


Figure 7: Similar results as before. NB: the first comparison is not perfect since BMI/weight/age is still taken 1 year prior.


Sensitivity analysis: [x to x−2] prediction window
Sensitivity
1 − Specificity
0.00
0.25
0.50
0.75
1.00
Window
[3−1] (AUC=0.828)
[4−2] (AUC=0.82)
[5−3] (AUC=0.804)


Sensitivity analysis: [x−1] prediction window
Sensitivity
1 − Specificity
0.00
0.25
0.50
0.75
1.00
Window
[2−1] (AUC=0.807)
[3−1] (AUC=0.828)
[4−1] (AUC=0.839)
[5−1] (AUC=0.84)

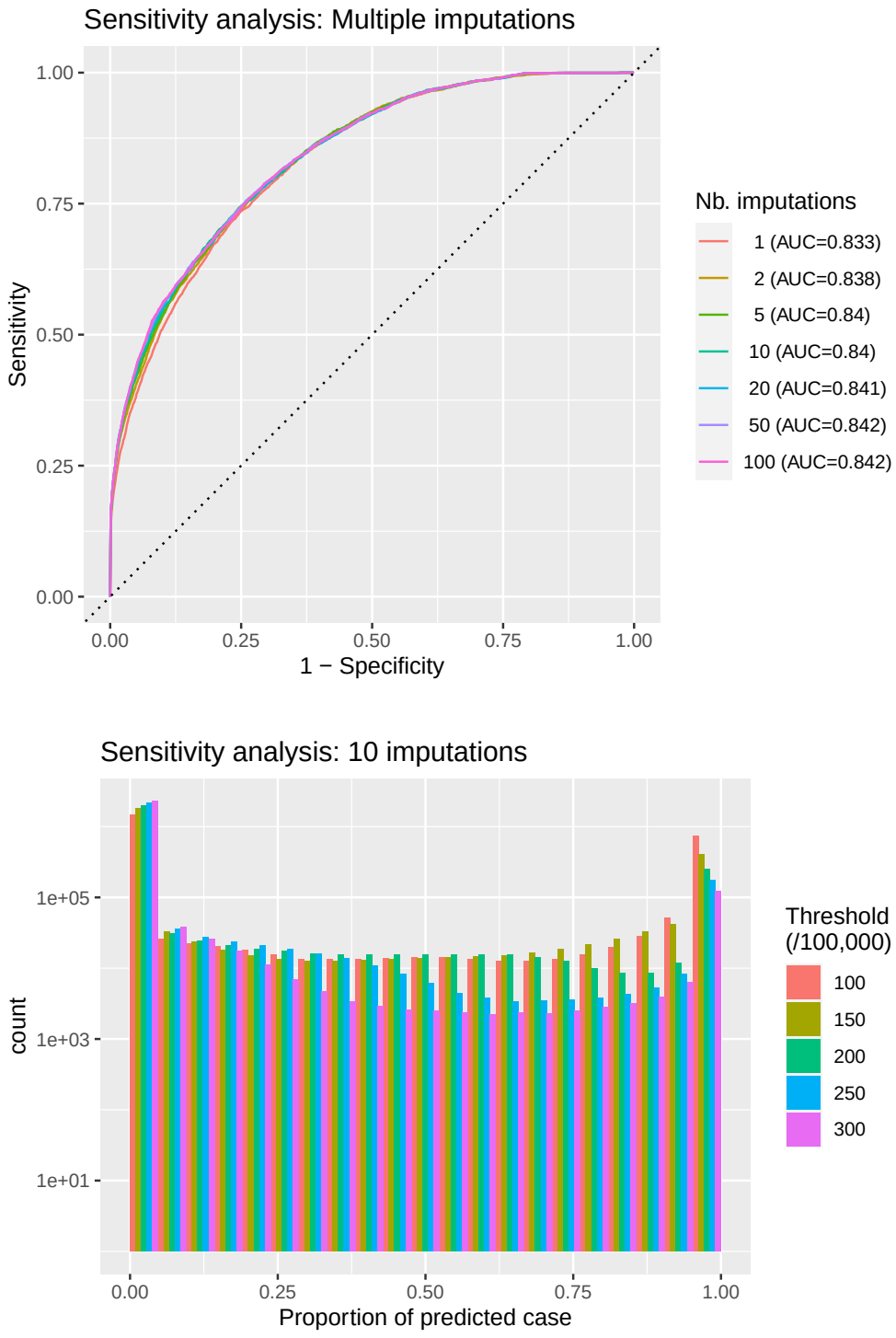


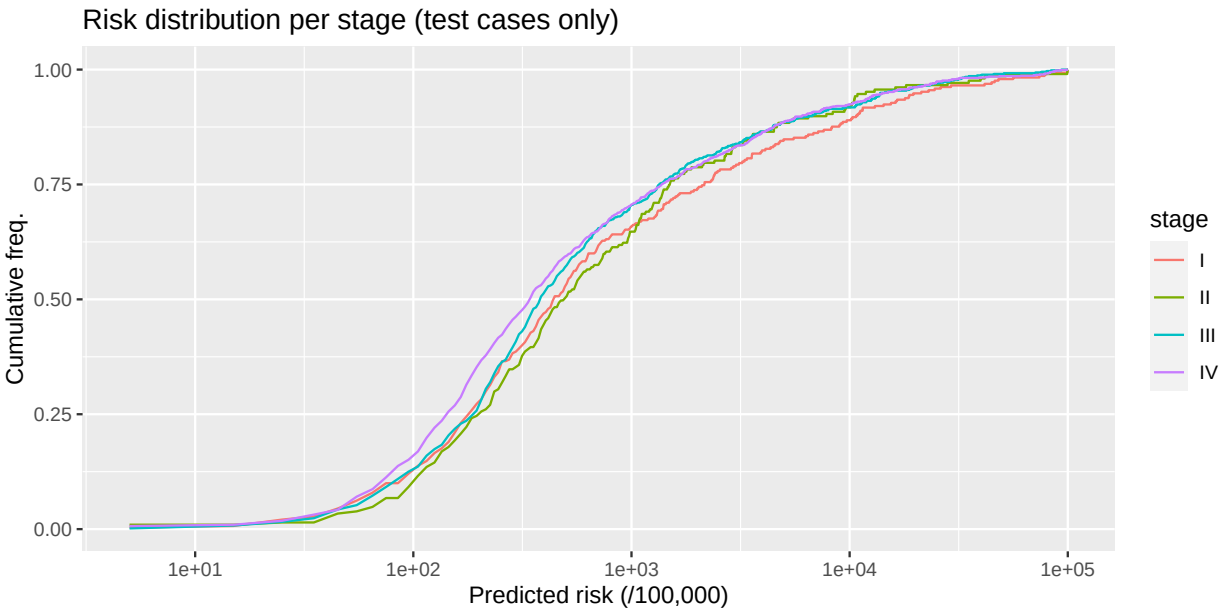
Figure 8: Not much to gain with multiple imputation, it seems most of the action is around the 200-300 threshold, so it doesn't seem to influnce predictions that much! See, e.g., between 100 and 150, we get almost no change in predictions.

Table 3:

Value	Clinical T	Clinical N	Clinical M
None	10257853	10257841	10257914
Total Non-null	10429	10441	10368
88	2	2	2
c0	18	4274	6524
c1	1131	3554	3113
c1A	245	0	109
c1B	237	0	225
c2	1239	945	0
c2A	78	0	0
c2B	34	0	0
c3	3464	479	0
c3A	0	1	0
c4	685	0	0
c4A	136	0	0
c4B	261	0	0
cX	2751	1186	367
pIS	148	0	0
p1	0	0	28

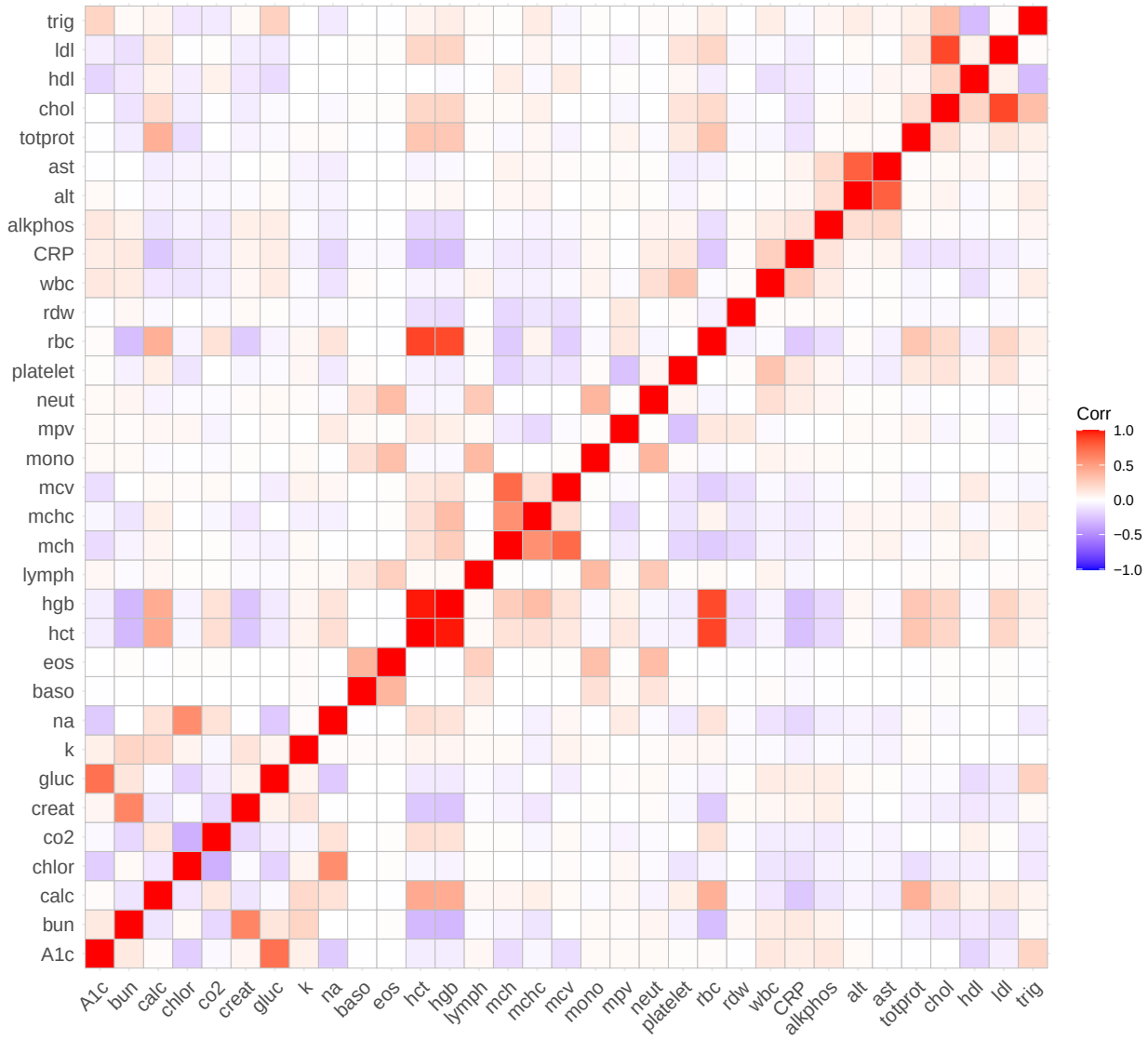
03/30/2022 update

- Started processing new cases
- A few follow-ups from last week
- Stages & ICD 10



stage	auc	n
all	0.834	2814
I	0.843	280
II	0.837	212
III	0.832	633
IV	0.838	1023
I+	0.836	2196
II+	0.835	1916
III+	0.835	1704
IV+	0.838	1023

Table 4: Essentially no difference between stages. Maybe stage III has lower predicted risk and lower AUC, but not a lot.


trig
ldl
hdl
chol
totprot
ast
alt
alkphos
CRP
wbc
rdw
rbc
platelet
neut
mpv
mono
mcv
mchc
mch
lymph
hgb
hct
eos
baso
na
k
gluc
creat
co2
chlor
calc
bun
A1c
A1c
bun
calc
chlor
co2
creat
gluc
k
na
baso
eos
hct
hgb
lymph
mch
mchc
mcv
mono
mpv
neut
platelet
rbc
rdw
wbc
CRP
alkphos
alt
ast
totprot
chol
hdl
ldl
trig
Corr
1.0
0.5
0.0
−0.5
−1.0

	var1	var2	correlation
2	mchc	mch	0.569
3	chlor	na	0.578
6	creat	bun	0.617
8	gluc	A1c	0.707
10	mcv	mch	0.742
11	alt	ast	0.782
13	hgb	rbc	0.857
15	chol	ldl	0.875
18	rbc	hct	0.884
20	hgb	hct	0.976

Table 5: HGB and HCT are indeed very higly correlated

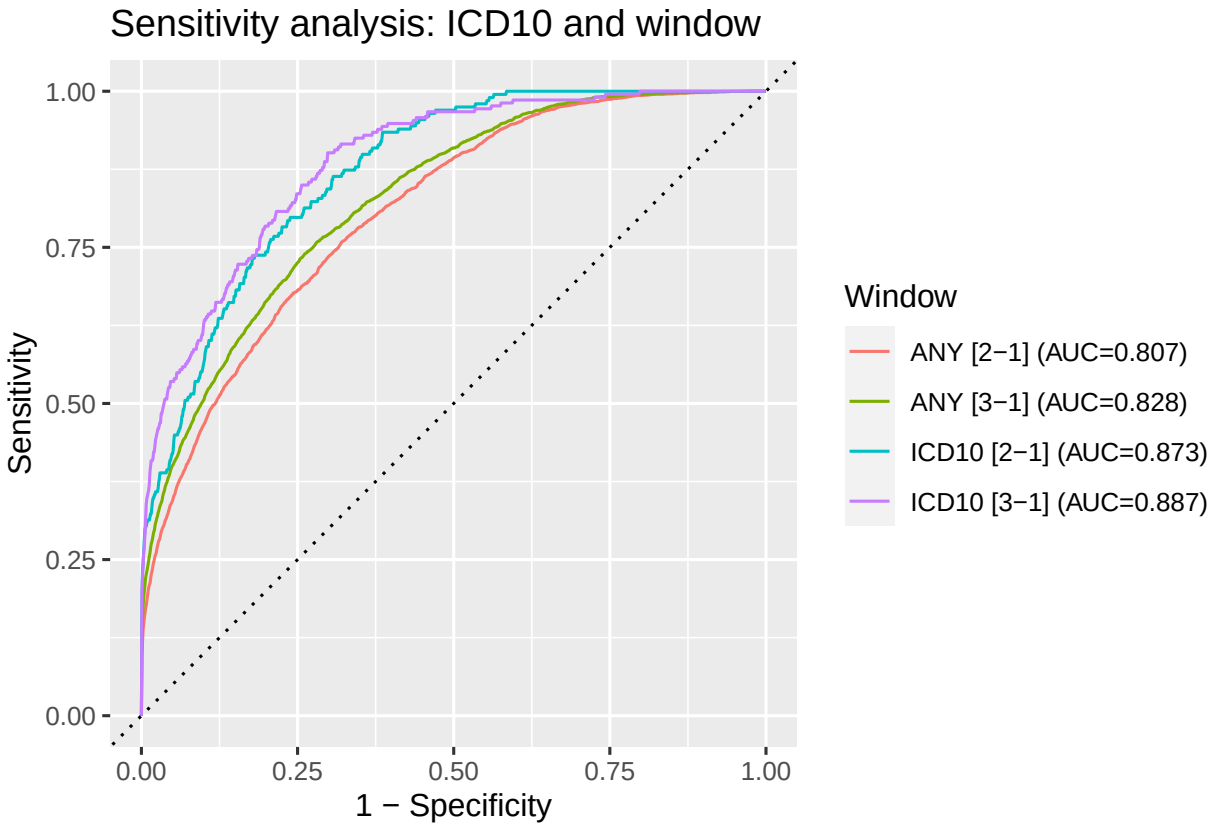
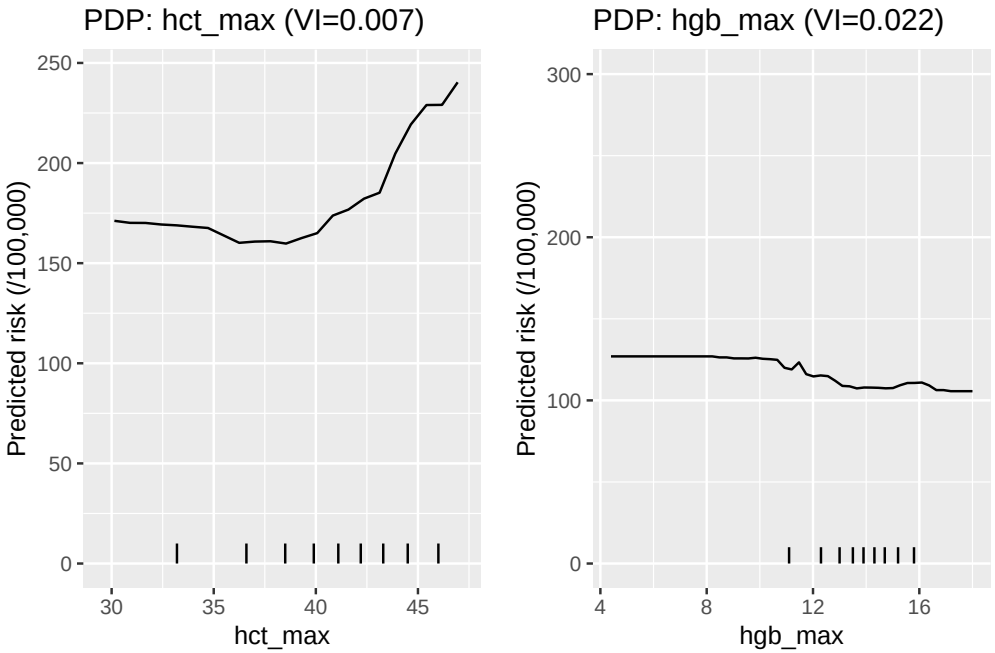
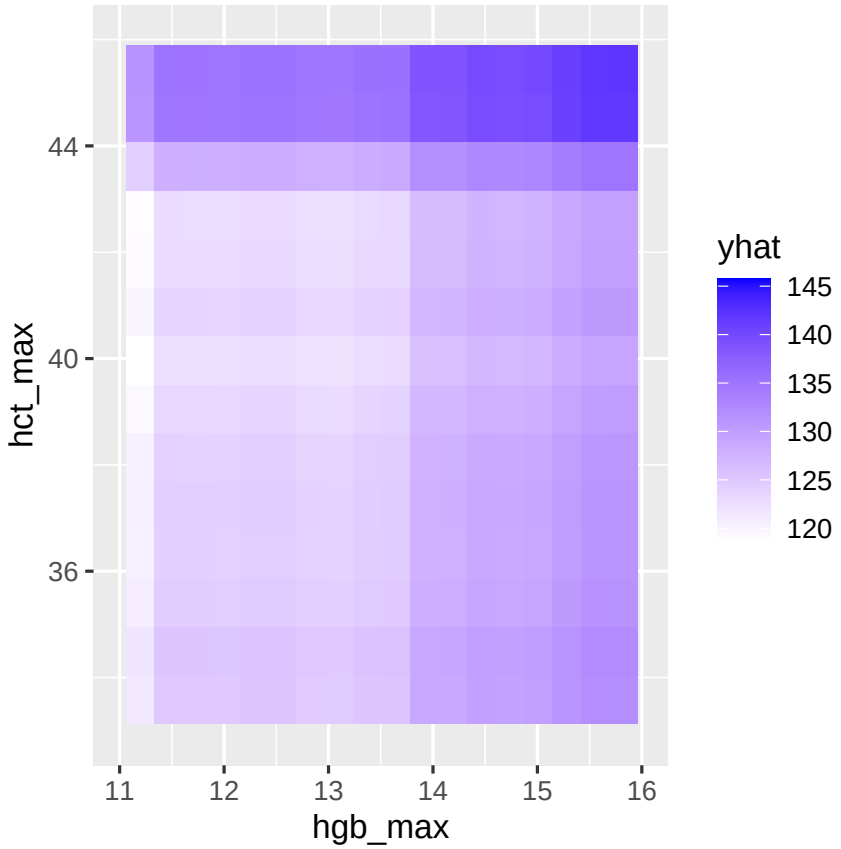


Figure 9: We are doing somewhat better on ICD10-only data!


PDP: hct_max (VI=0.007)
Predicted risk (/100,000)
250
200
150
100
50
0
30
35
40
45
hct_max
PDP: hgb_max (VI=0.022)
Predicted risk (/100,000)
300
200
100
0
4
8
12
16
hgb_max

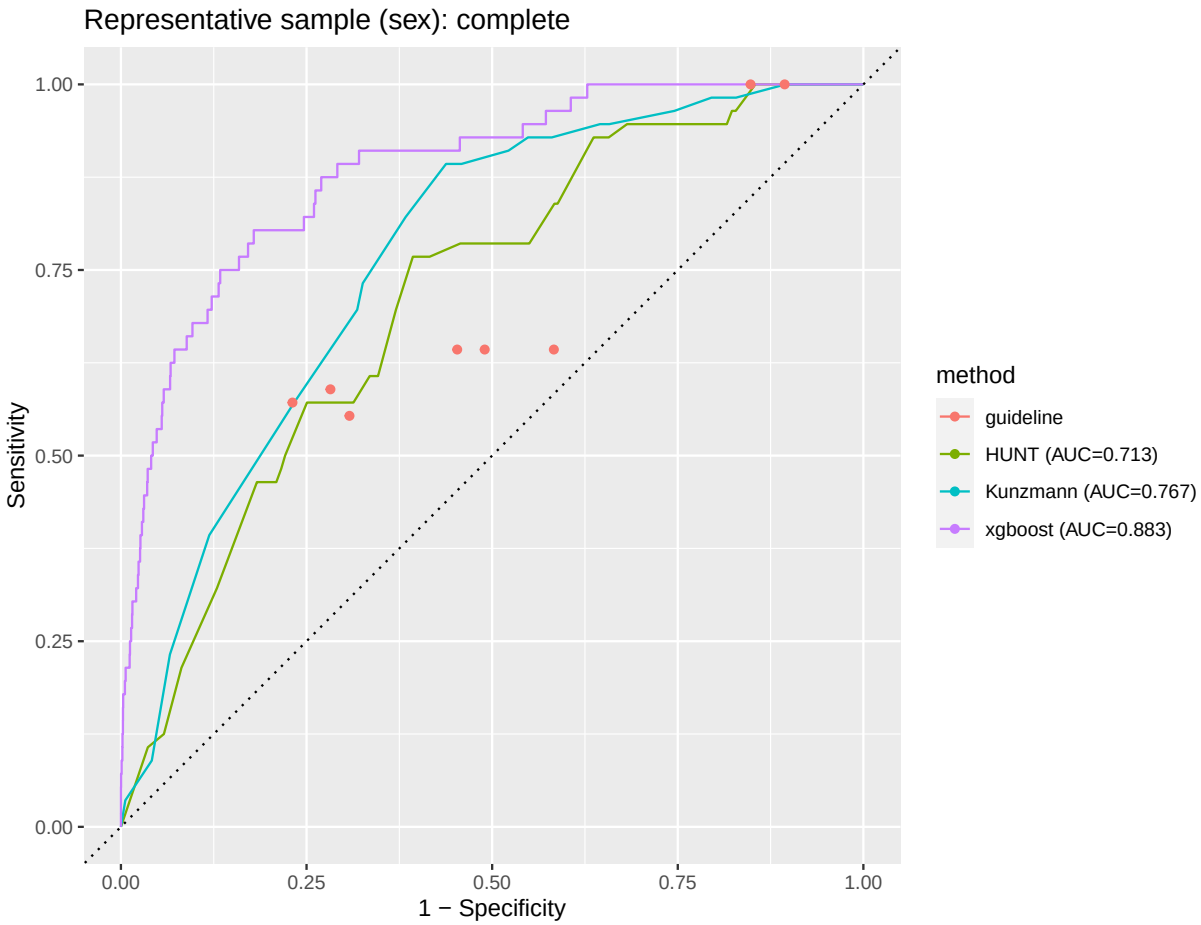

hct_max
44
40
36
11
12
13
14
15
16
hgb_max
yhat
145
140
135
130
125
120

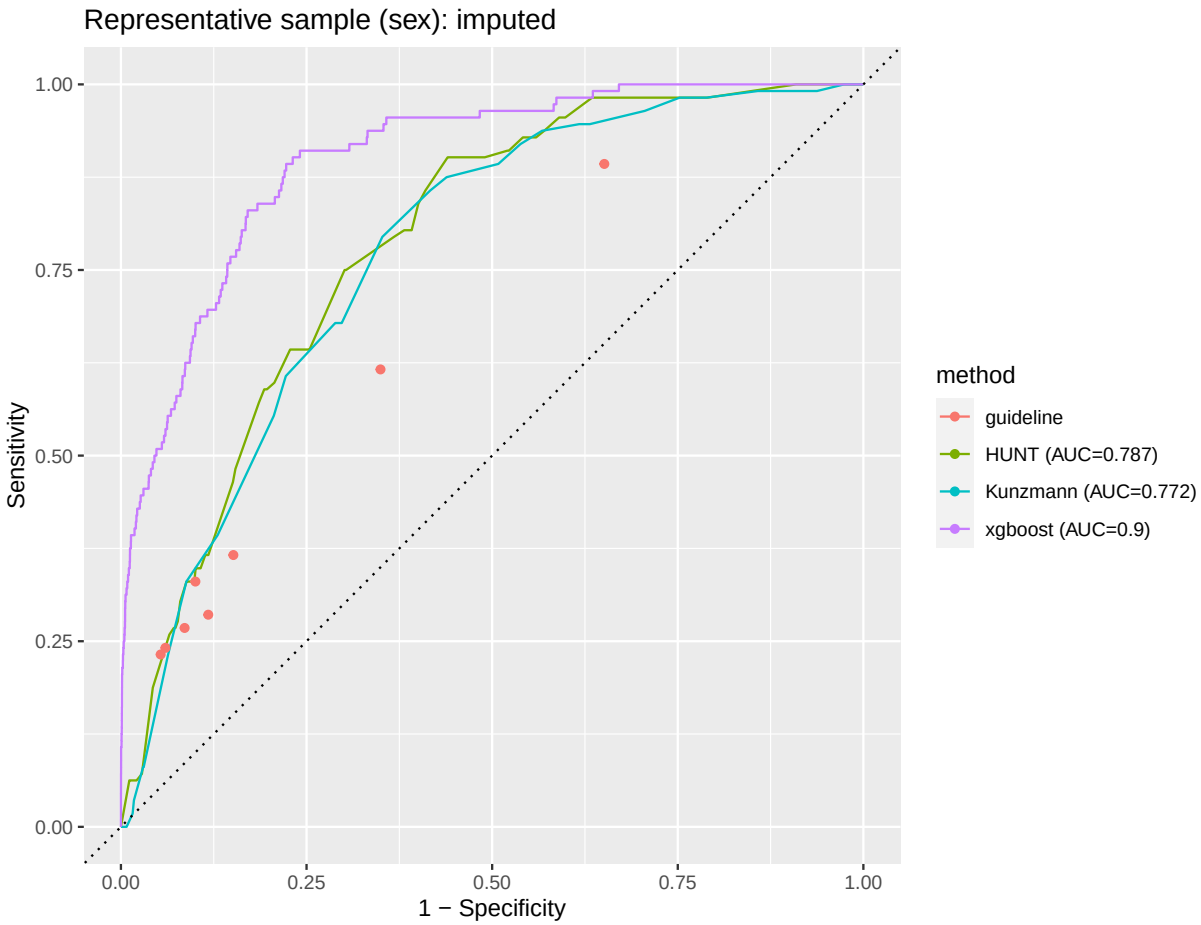
04/26/2022 update

Complete cases only			
Sex	Cases	Controls	Total
F	4	41197	41201
M	*33	**41168	41201
Total	37	82365	82402
Sex available, rest imputed			
F	9	147178	147187
M	*75	**147112	147187
Total	84	294290	294374

**sampled to 8.33:1 ratio*
***sampled to 1:1 ratio*

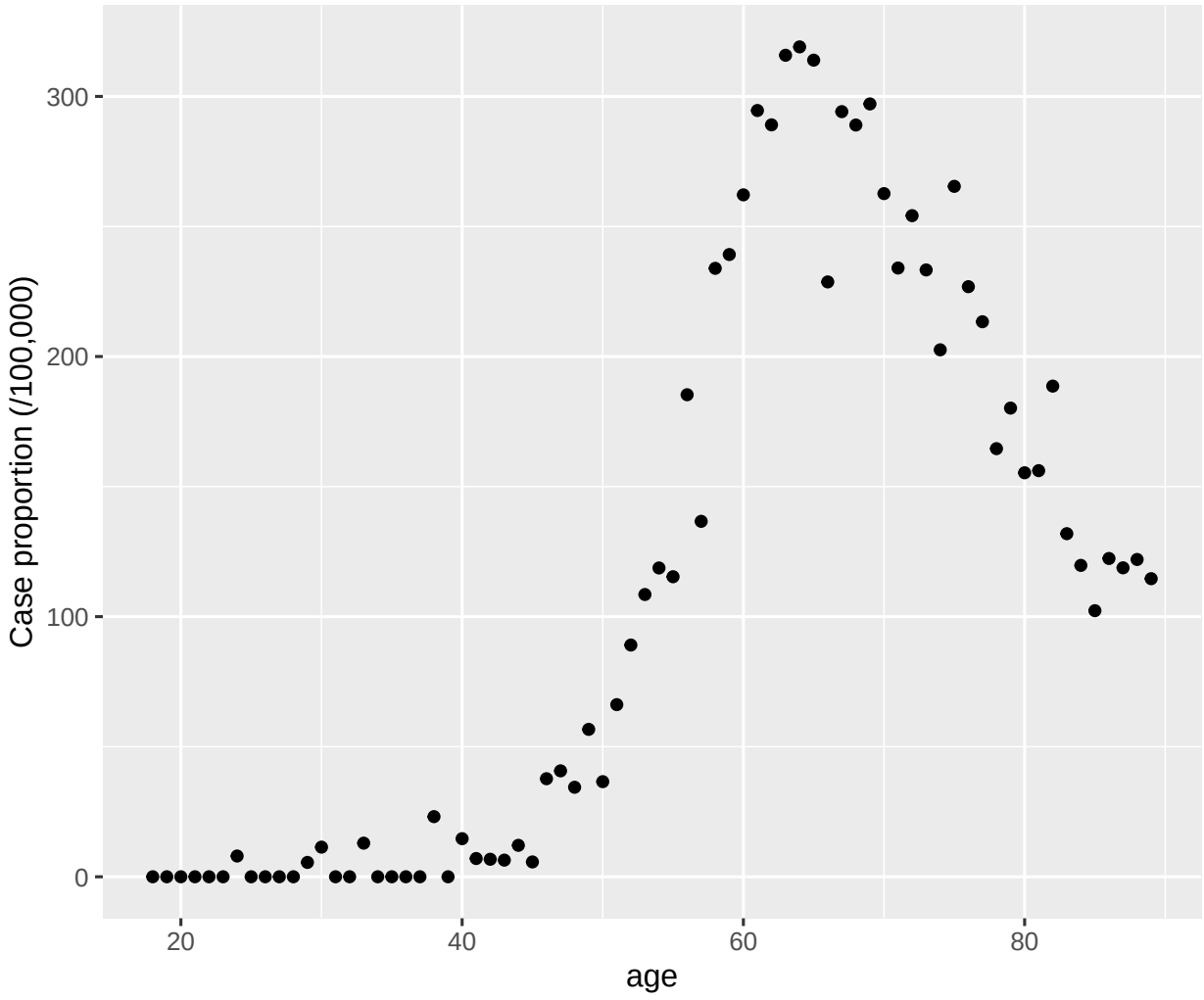
Representative sample (sex)


Representative sample (sex): complete
Sensitivity
1 − Specificity
0.00
0.25
0.50
0.75
1.00
method
guideline
HUNT (AUC=0.713)
Kunzmann (AUC=0.767)
xgboost (AUC=0.883)


Representative sample (sex): imputed
Sensitivity
1 − Specificity
0.00
0.25
0.50
0.75
1.00
method
guideline
HUNT (AUC=0.787)
Kunzmann (AUC=0.772)
xgboost (AUC=0.9)

Age effect

- Proportion of cases by age, SHAP values, PDP and predicted risk per age
- All seem to indicate a drop after 60-65



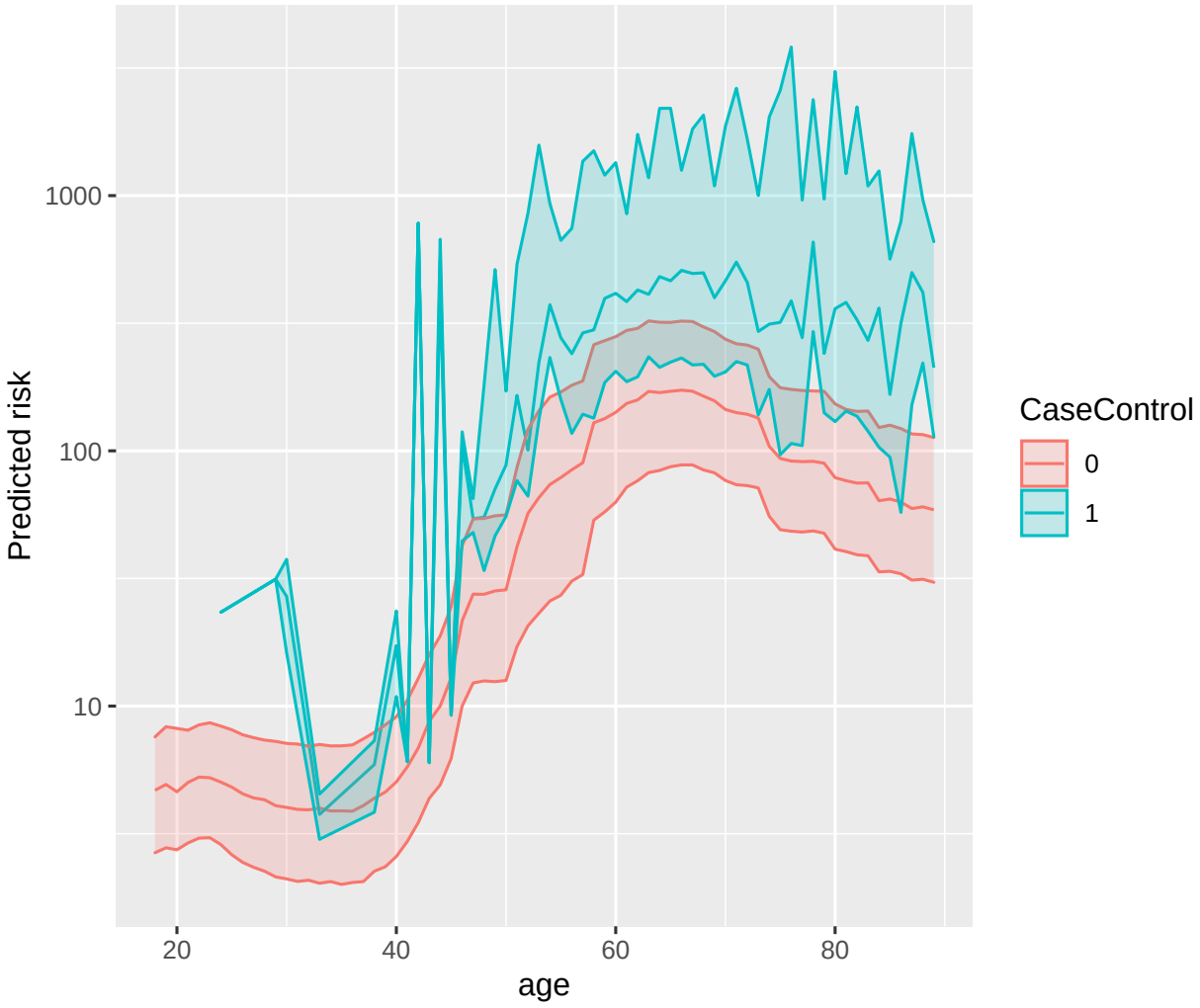
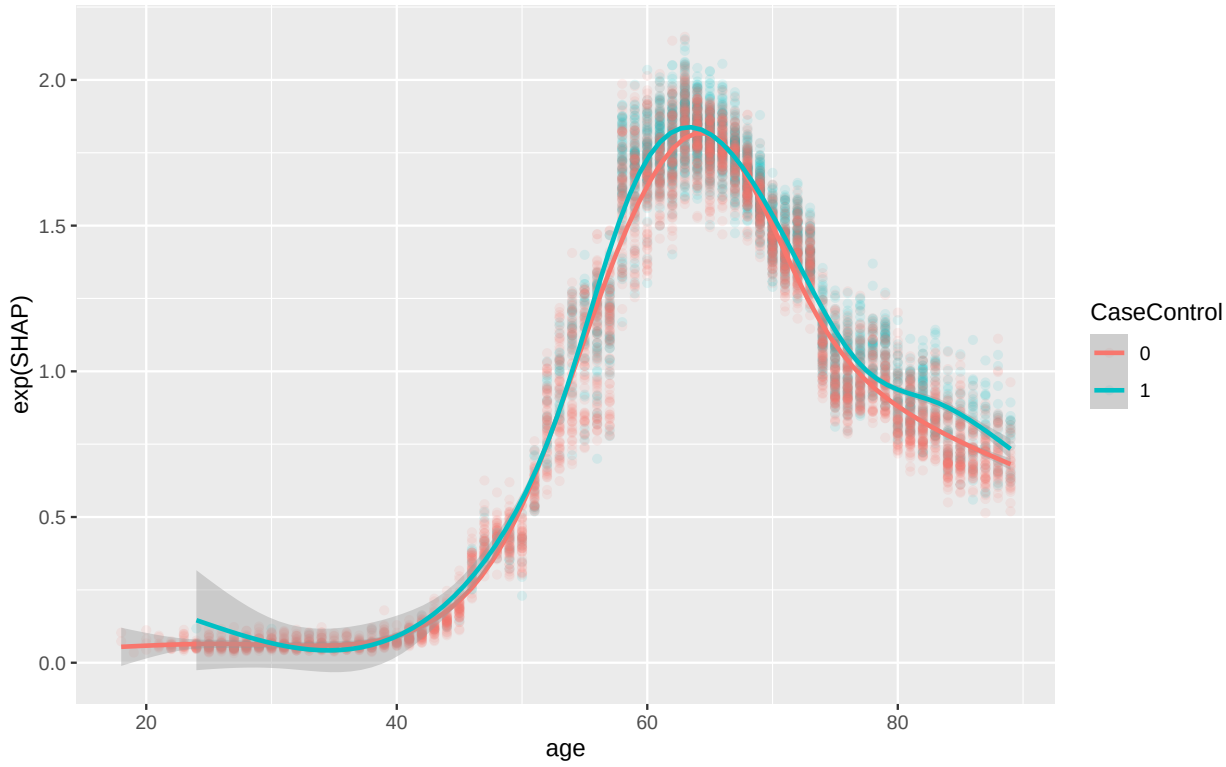
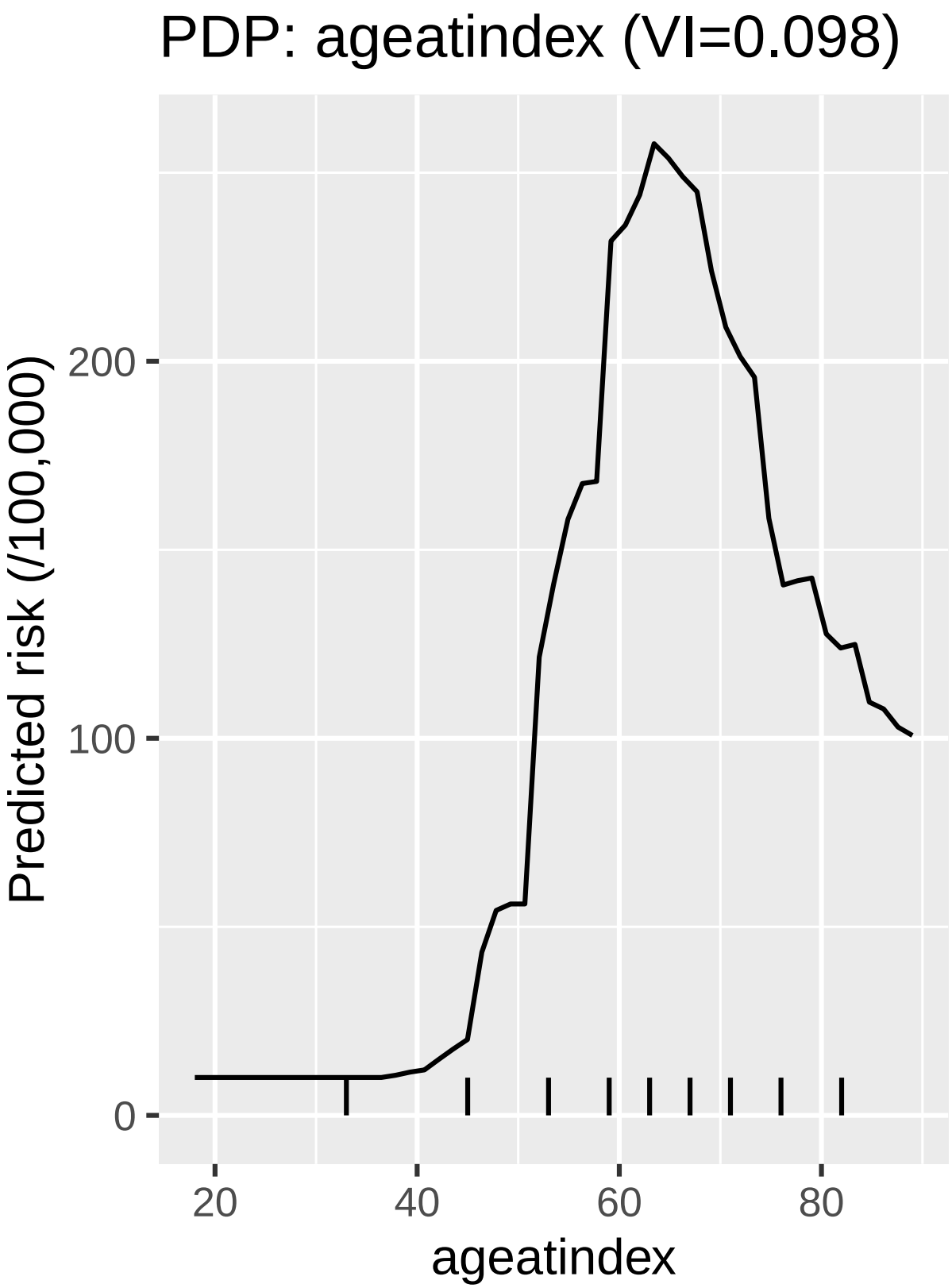


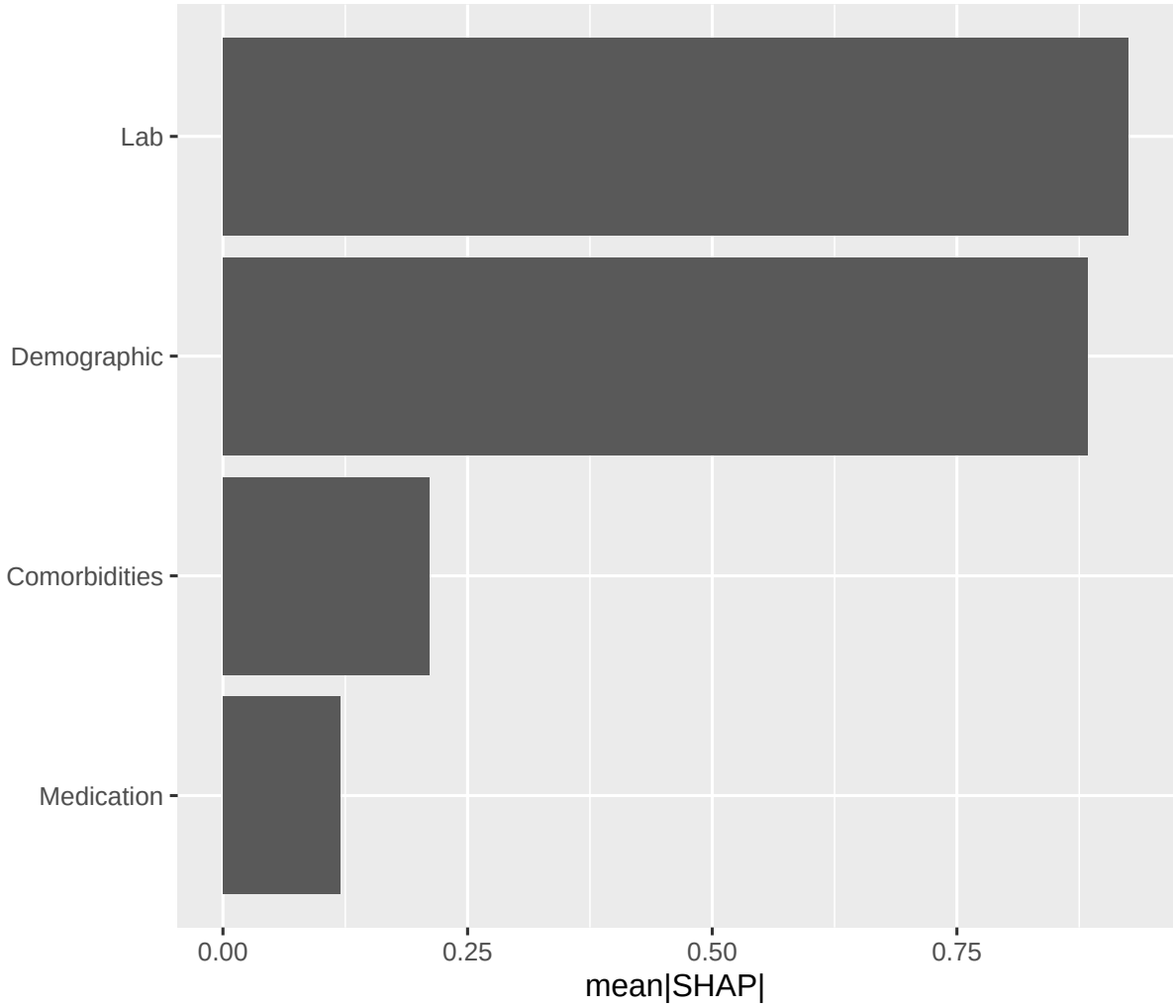
Figure 10: 25-, 50- and 75th percentiles

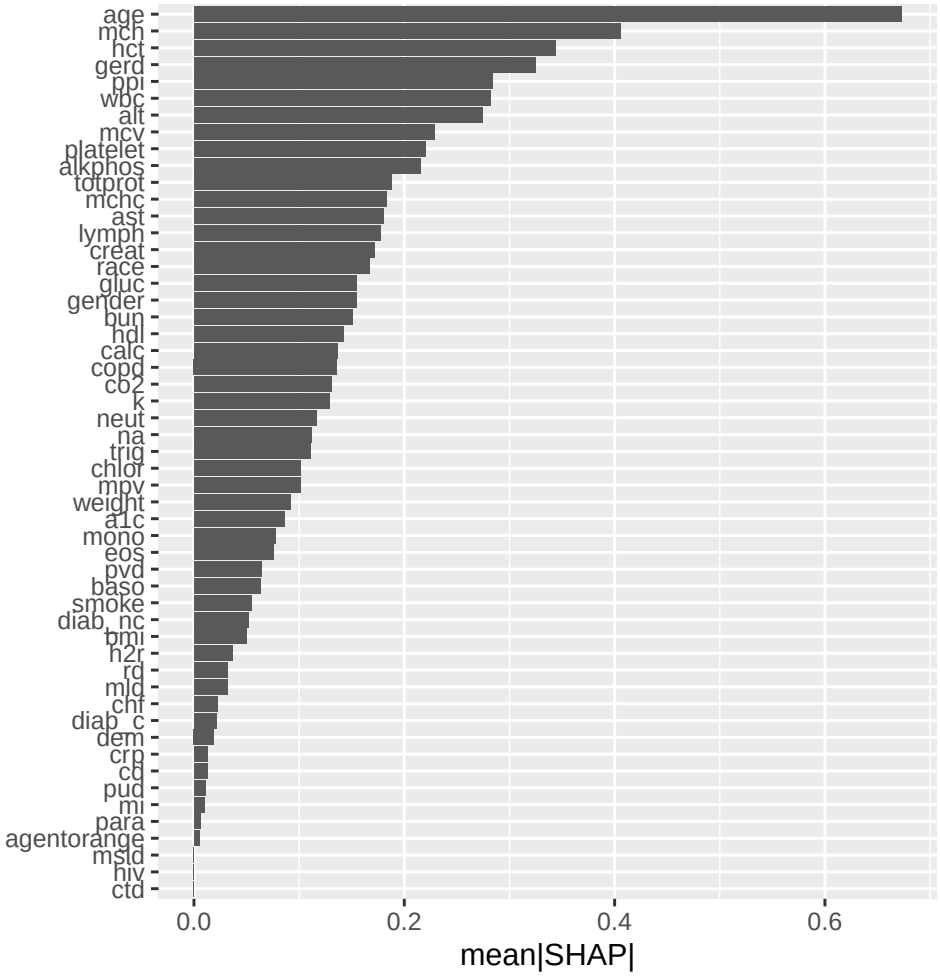

exp(SHAP)
2.0
1.5
1.0
0.5
0.0
20
40
60
80
age
CaseControl
0
1


PDP: ageatindex (VI=0.098)
Predicted risk (/100,000)
200
100
0
20
40
60
80
ageatindex

Variable importance via SHAP values

- SHAP values are an alternative way to inspect variable importance
- very broadly: for each observation, how much that feature changes the model output (here: log-odds)
- Can be aggregated (mean|SHAP|) to global importance
- Additive measure so we can aggregate features into groups
- Gives a somewhat different ranking that standard VI (which gives how much a feature changes the loss function)




age
mch
hct
gerd
ppi
wbc
alt
mcv
platelet
alkphos
totprot
mchc
ast
lymph
creat
race
gluc
gender
bun
hdl
calc
copd
co2
k
neut
na
trig
chlor
mpv
weight
a1c
mono
eos
pvd
baso
smoke
diab_nc
bmi
h2r
rd
mld
chf
diab_c
dem
crp
cd
pud
mi
para
agentorange
msld
hiv
ctd
0.0
0.2
0.4
0.6
mean|SHAP|

To do

- EAC v EGJAC: SHAP values
- PDP/SHAP before after dropping
- Smaller model?
- Finish R package (include model, quality control)

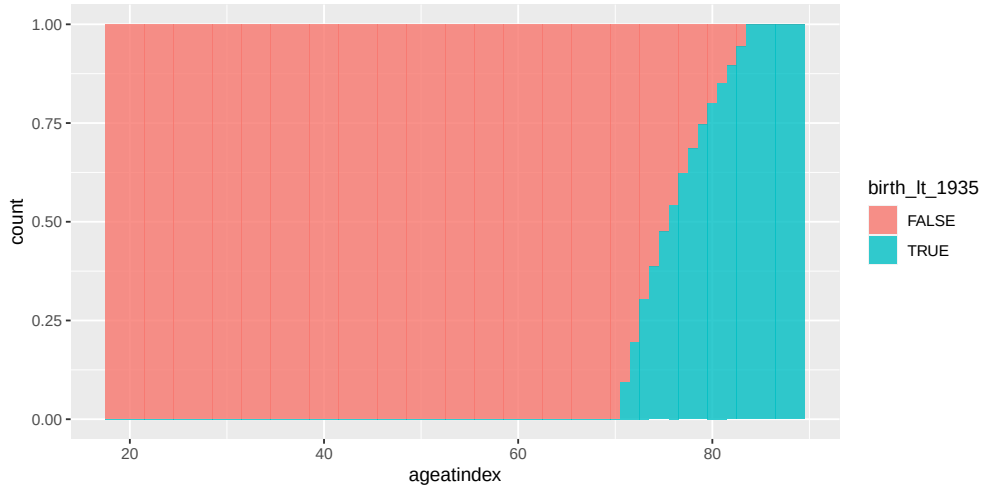
05/03/2022 update

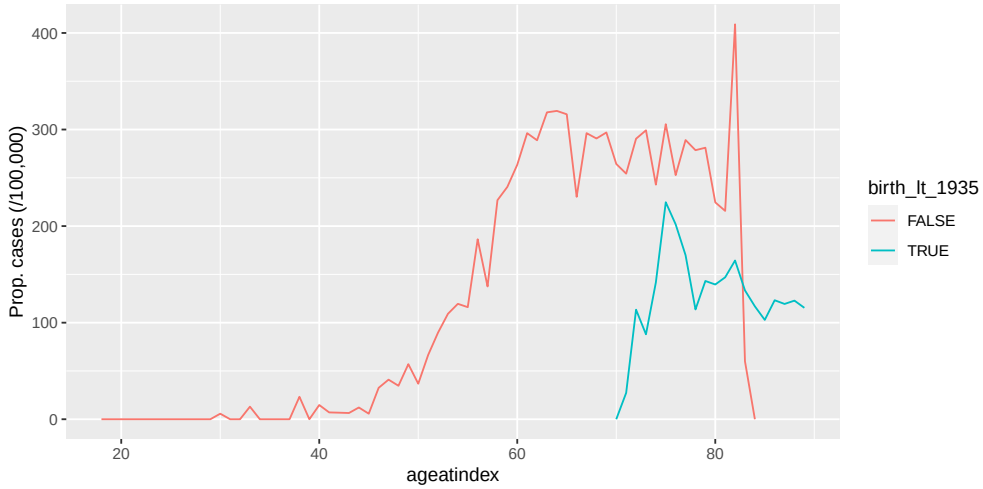
R package update

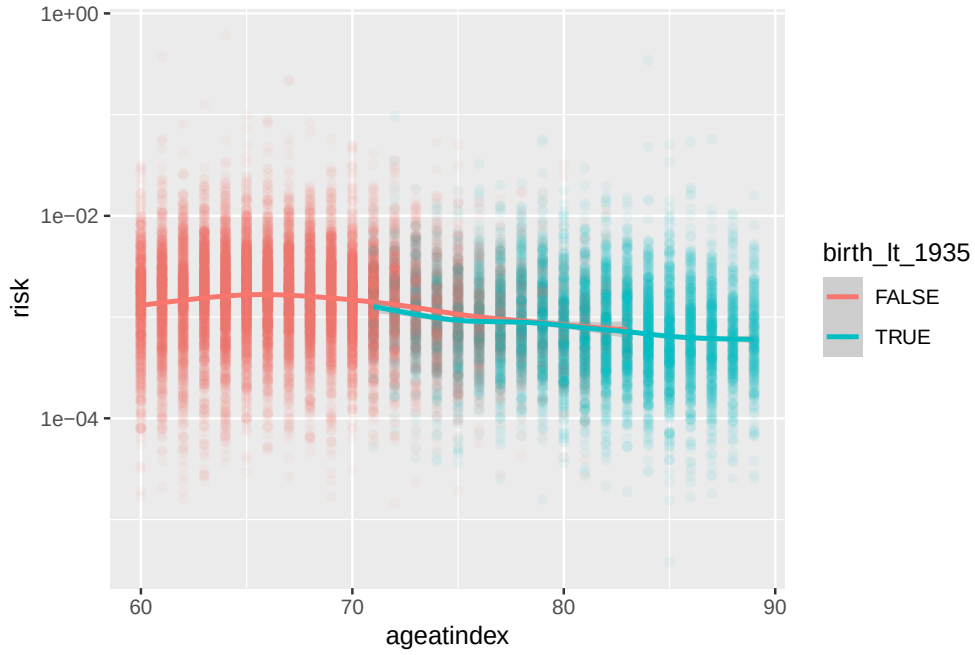
- Added EAC and EGJAC models
- Added some quality control checks

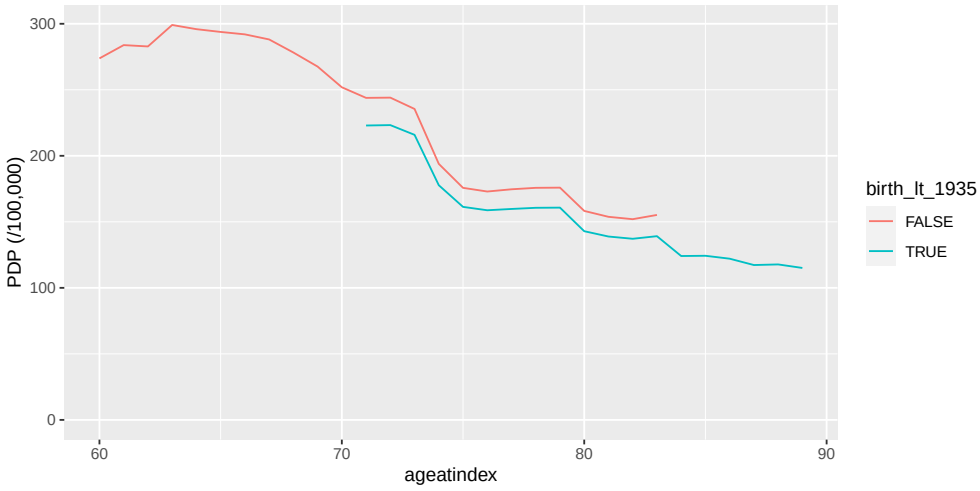
Birth cohort effect

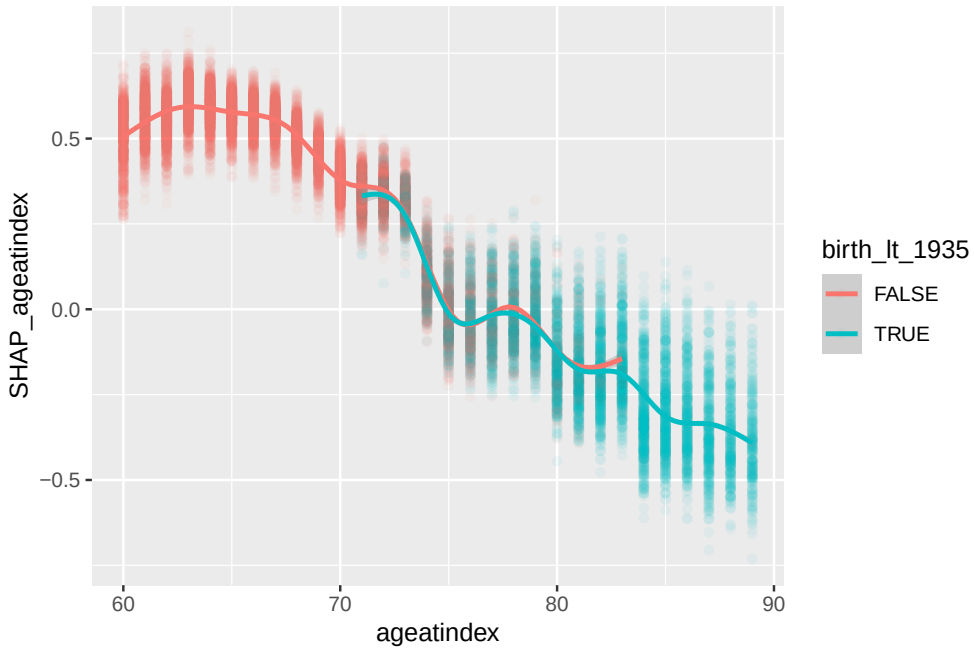
- Using index date, age at index and knowing data ends at 2019, I am able to get birth year (± 1 year)
- 70-83yo is where the switch between ≤ 1935 to > 1935
- There is a significant change in case prevalance between the two cohorts for a fixed age
- prevalance seems flat beyond $\sim$ 60yo
- PDP and predicted risk show some gap during the transition period
- There is a sudden drop at around 73yo in PDP and SHAP, which could indicate that the model is trying its best at modeling the birth cohort effect






1e+00
1e−02
1e−04
risk
60
70
80
90
ageatindex
birth_lt_1935
FALSE
TRUE


300
200
100
0
PDP (/100,000)
60
70
80
90
ageatindex
birth_lt_1935
FALSE
TRUE

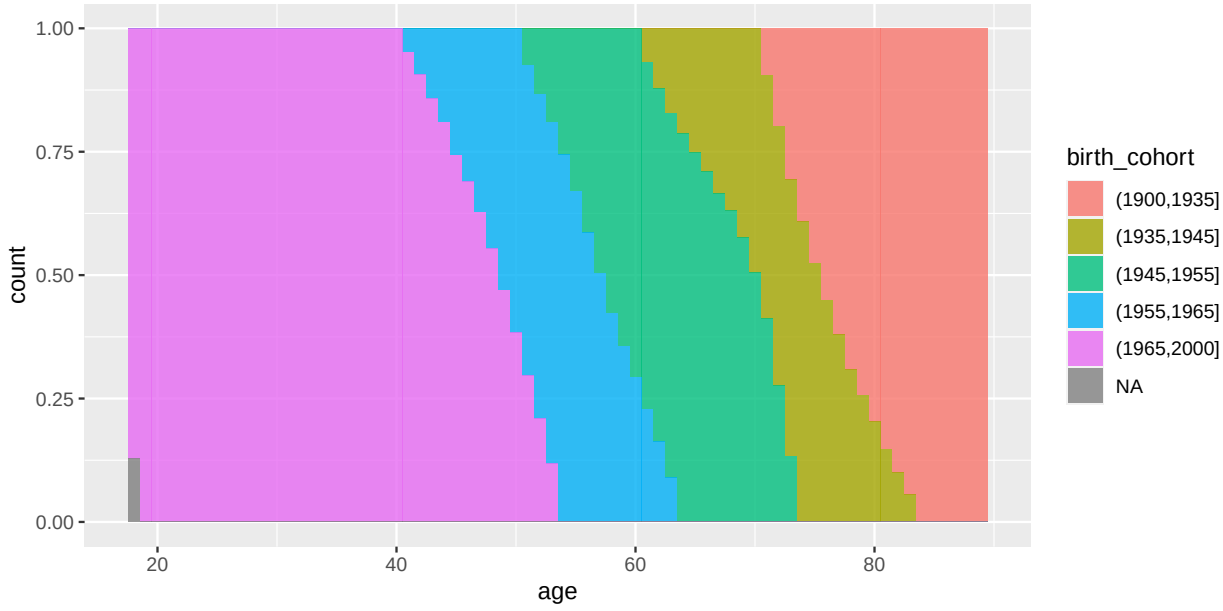

SHAP_ageatindex
0.5
0.0
−0.5
60
70
80
90
ageatindex
birth_lt_1935
FALSE
TRUE

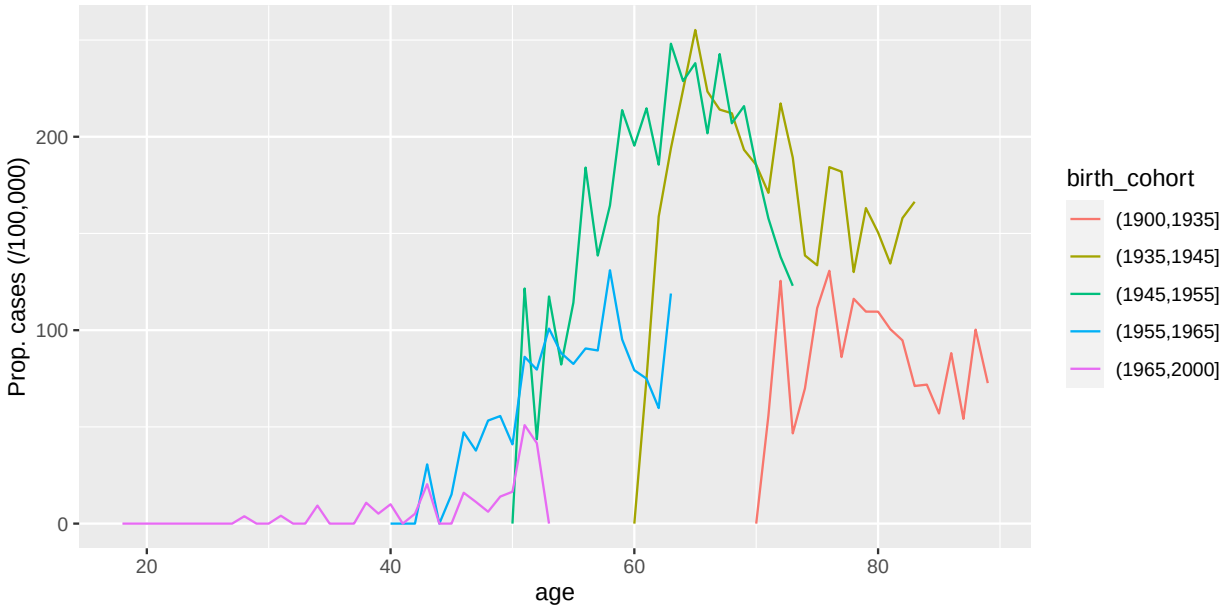
05/10/2022 update

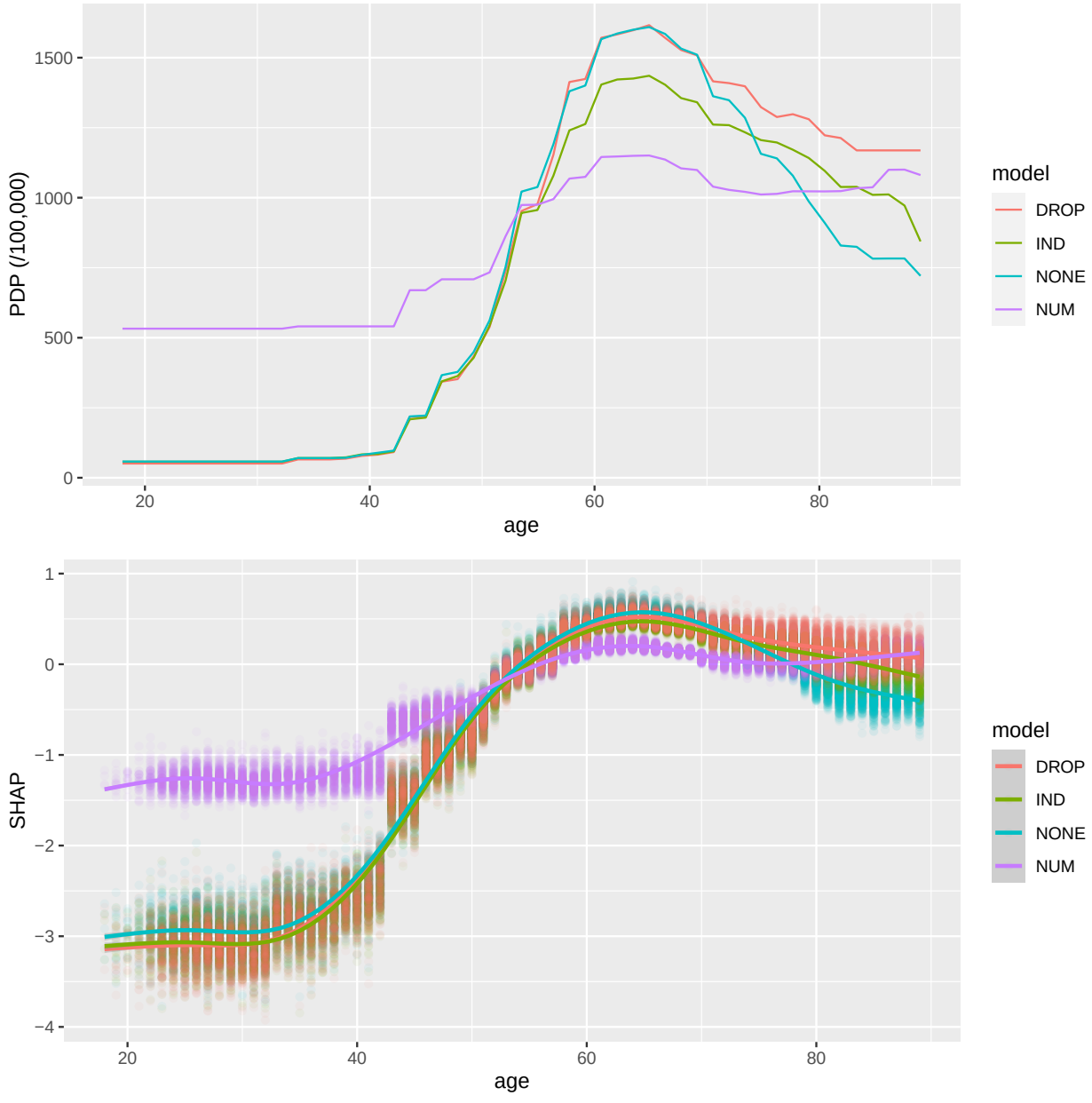
Birth cohort effect – follow-up

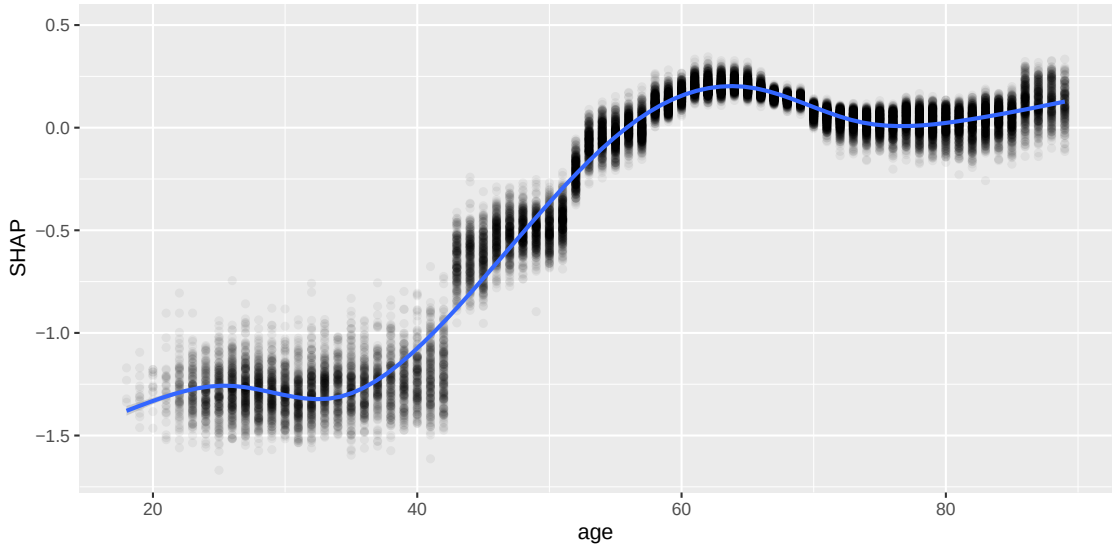
- Fitted four models (on 1M controls):
  - NONE: do nothing special, same model as before
  - DROP: only fit on subjects after 1935
  - IND: add the 1935 indicator
  - NUM: use birth year as a feature directly
- Compare prediction performance (Test AUC)
  - Stratified by a few birth cohort
- PDP and SHAP curves
- NUM: can check PDP and SHAP of birth year and compare
- Notes:
  - Since we use 1M controls, calibration and raw numbers will be off, but the trends are still meaningful
- Results:
  - NUM produces the best predicted risks up to 1965; it is also the best overall (note that it improves on the current model)
  - 1935 does not seem the best cutoff, more like a curve or at least the interval 1935-1955 has higher incidence than the rest
  - Trees will find the "best" cutoff anyway ...
  - Training only on $>$ 1935 subjects (DROP) or using the indicator (IND) seems to alleviate the drop in age effect after 65yo, but not as well as using birth year as feature (NUM)

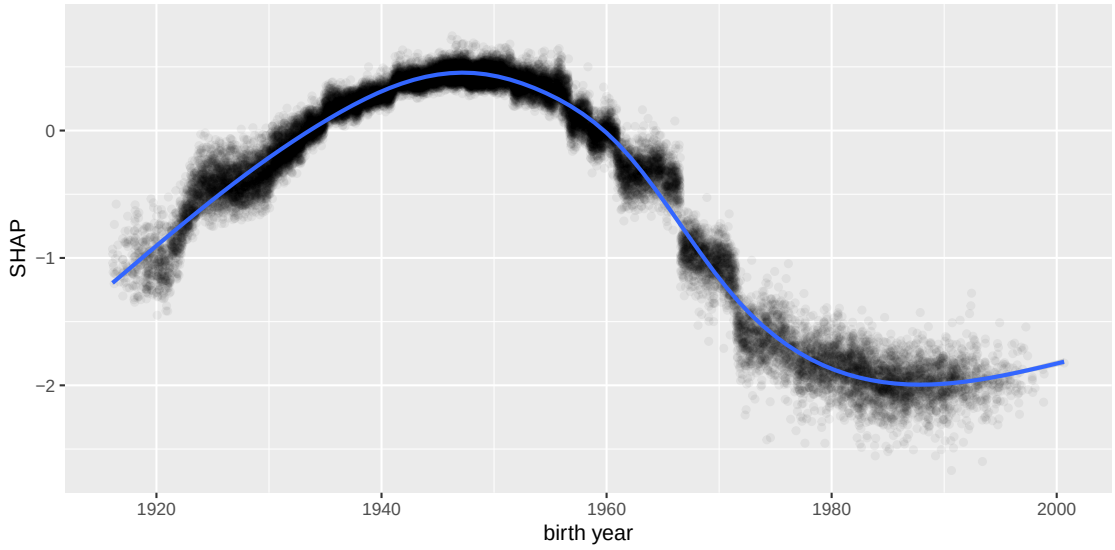
	Test AUC			
Cohort	NONE	IND	NUM	DROP
All	0.845	0.836	**0.850**	0.829
1900-1925	0.818	0.803	**0.826**	0.785
1925-1935	0.800	0.788	**0.805**	0.770
1935-1945	0.797	0.792	**0.806**	0.780
1945-1955	0.820	0.813	**0.829**	0.804
1955-1965	0.816	0.817	**0.827**	0.812
1965-1975	**0.893**	0.878	0.878	0.875
1975-2000	**0.758**	0.742	0.718	0.728


1.00
0.75
0.50
0.25
0.00
count
20
40
60
80
age
birth_cohort
(1900,1935]
(1935,1945]
(1945,1955]
(1955,1965]
(1965,2000]
NA


200
100
0
Prop. cases (/100,000)
20
40
60
80
age
birth_cohort
(1900,1935]
(1935,1945]
(1945,1955]
(1955,1965]
(1965,2000]


PDP (/100,000)
1500
1000
500
0
20
40
60
80
age
model
DROP
IND
NONE
NUM
SHAP
1
0
−1
−2
−3
−4
20
40
60
80
age
model
DROP
IND
NONE
NUM


SHAP
0.5
0.0
−0.5
−1.0
−1.5
20
40
60
80
age


SHAP
0
−1
−2
1920
1940
1960
1980
2000
birth year


PDP (/100,000)
1000
500
1920
1940
1960
1980
2000
birth year

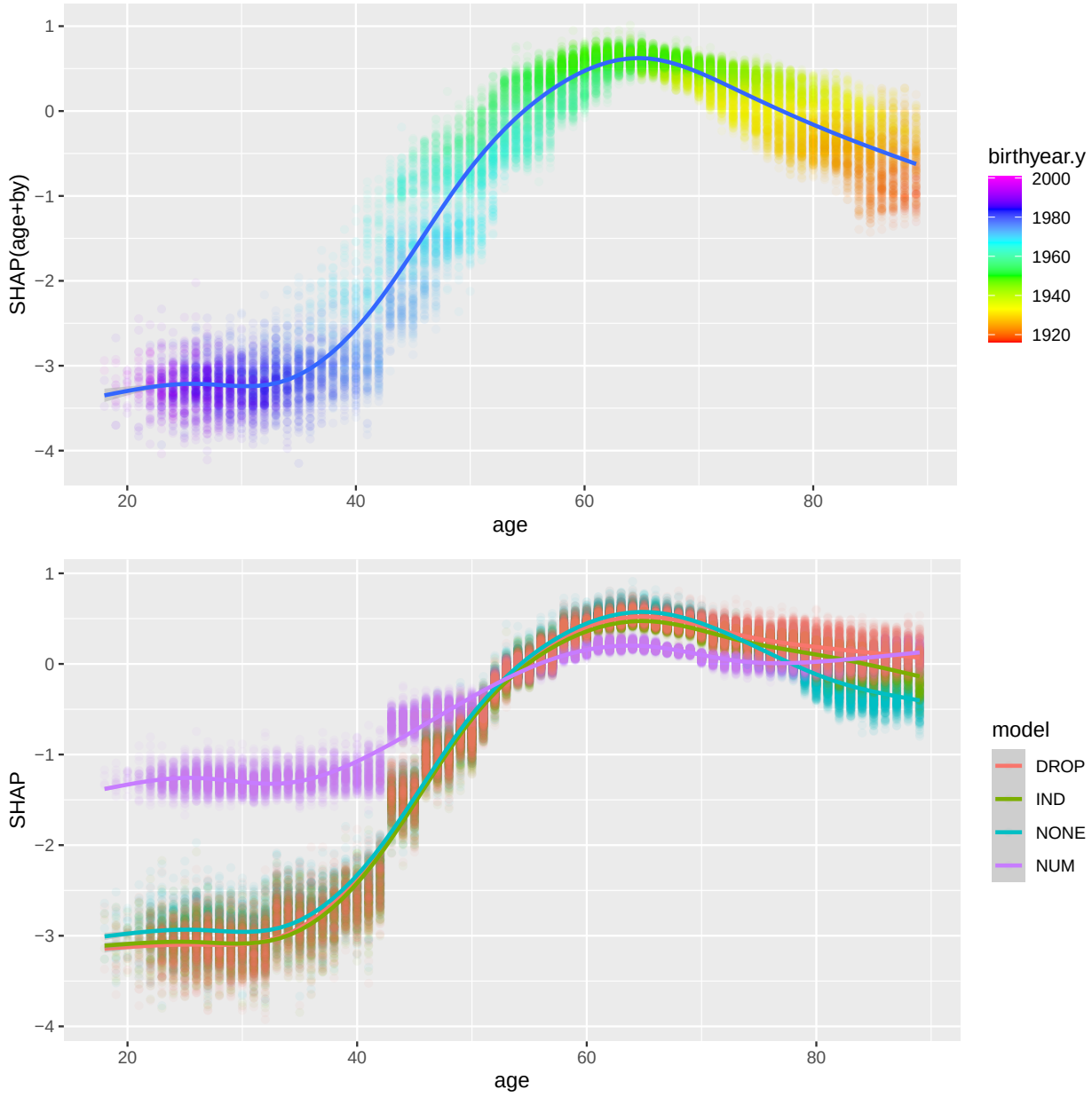
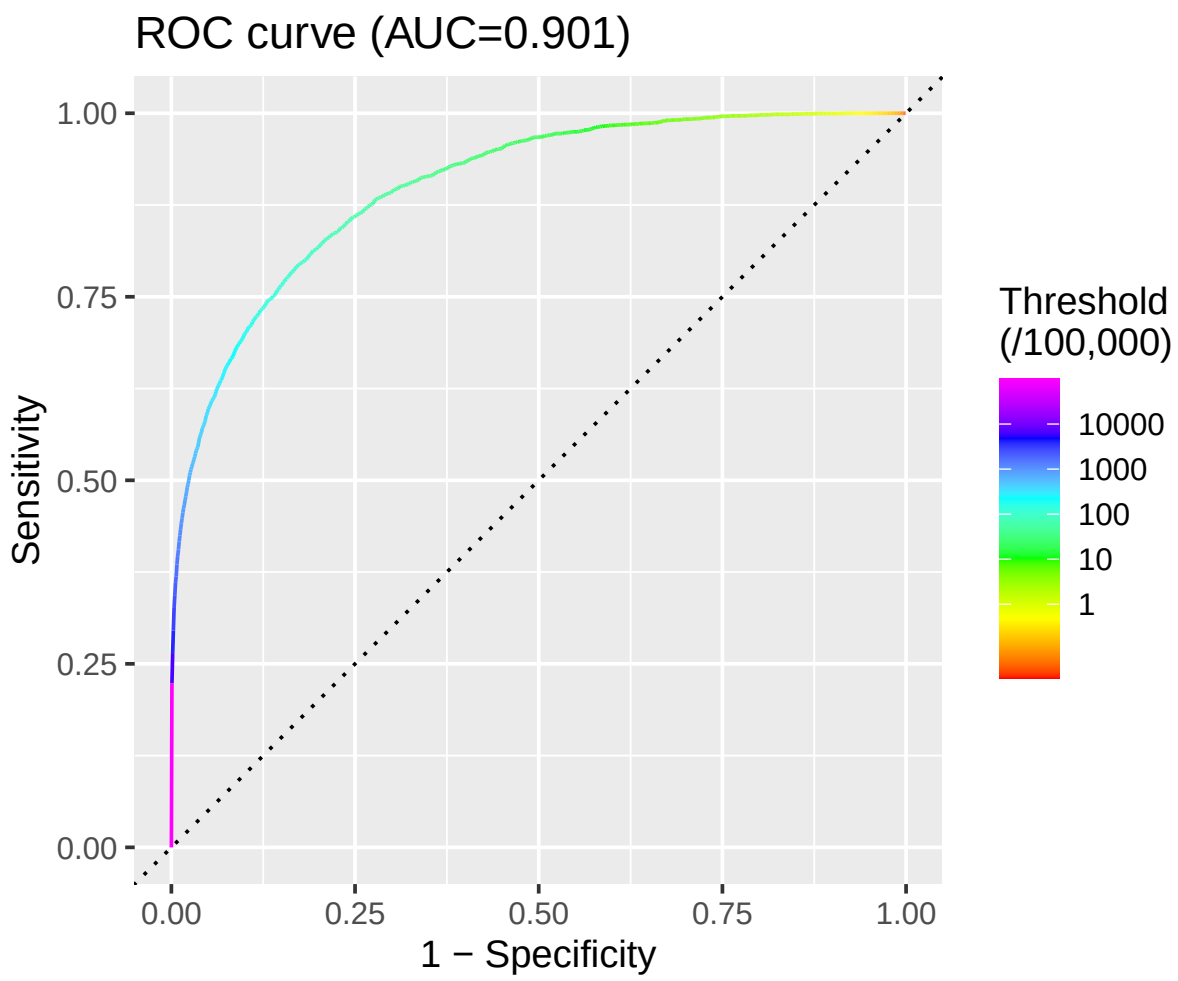
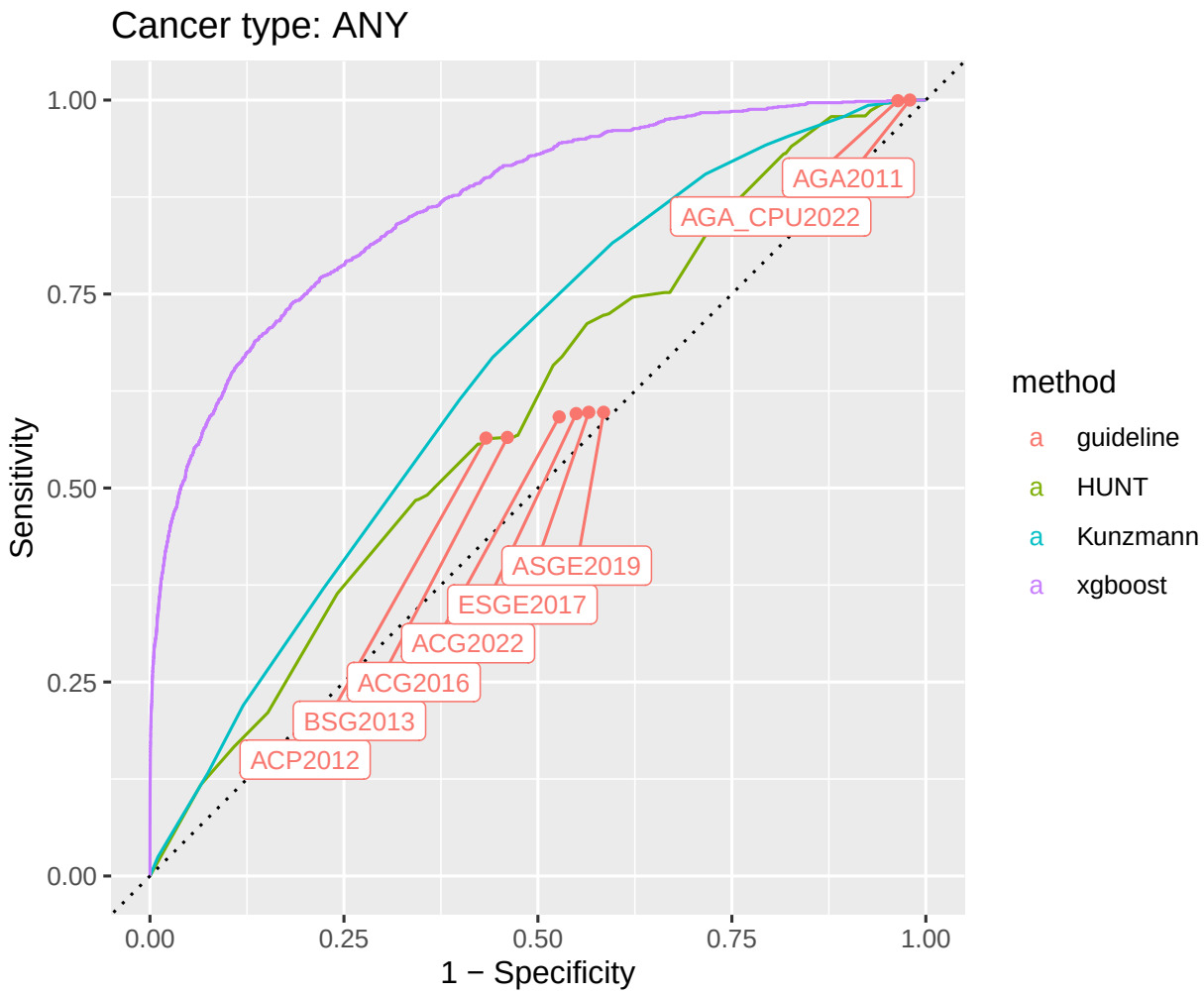


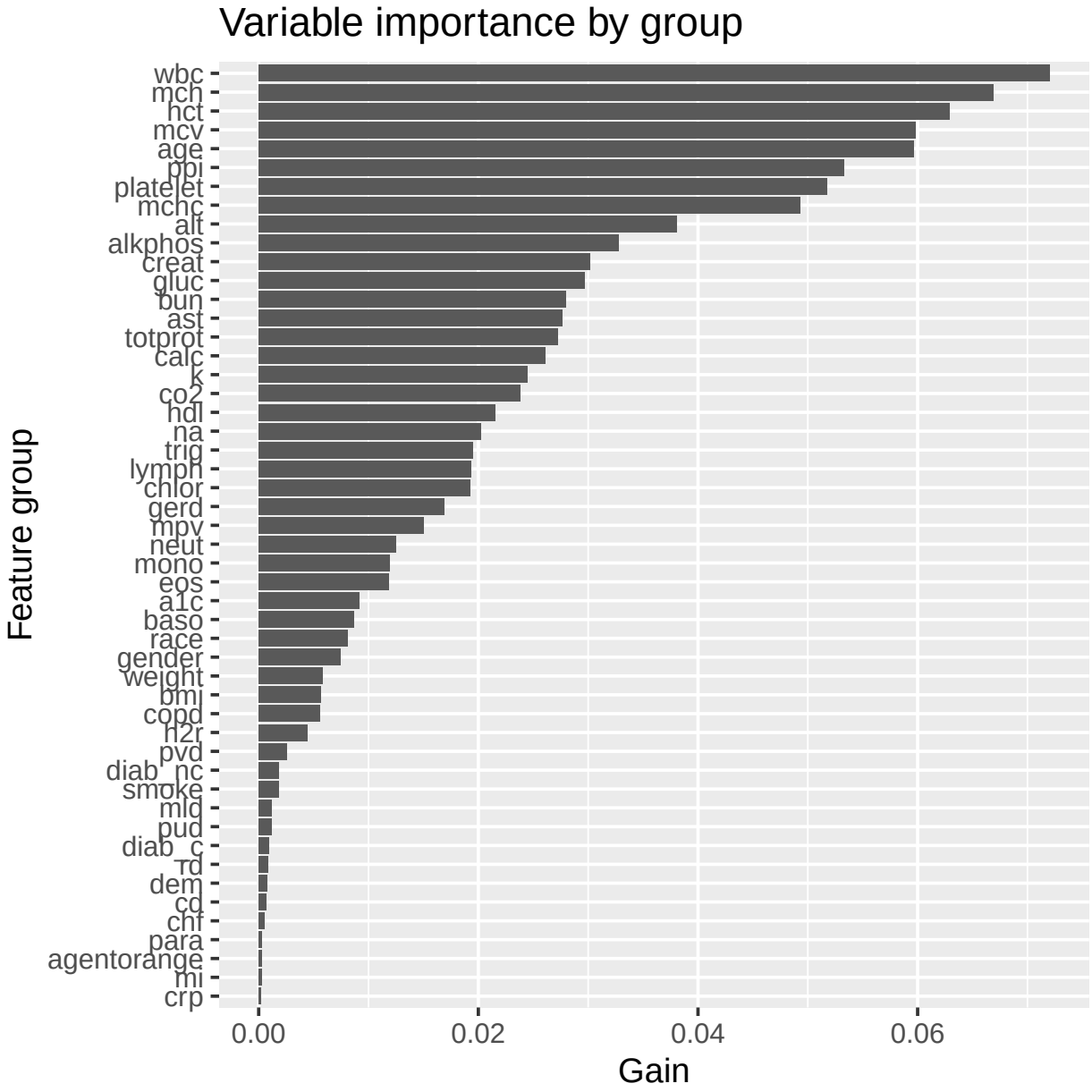
Figure 11: Aggregating the effect of age and birth cohort in the NUM model recovers the SHAP curves for age of the other models. Hence, we are decomposing the effect.

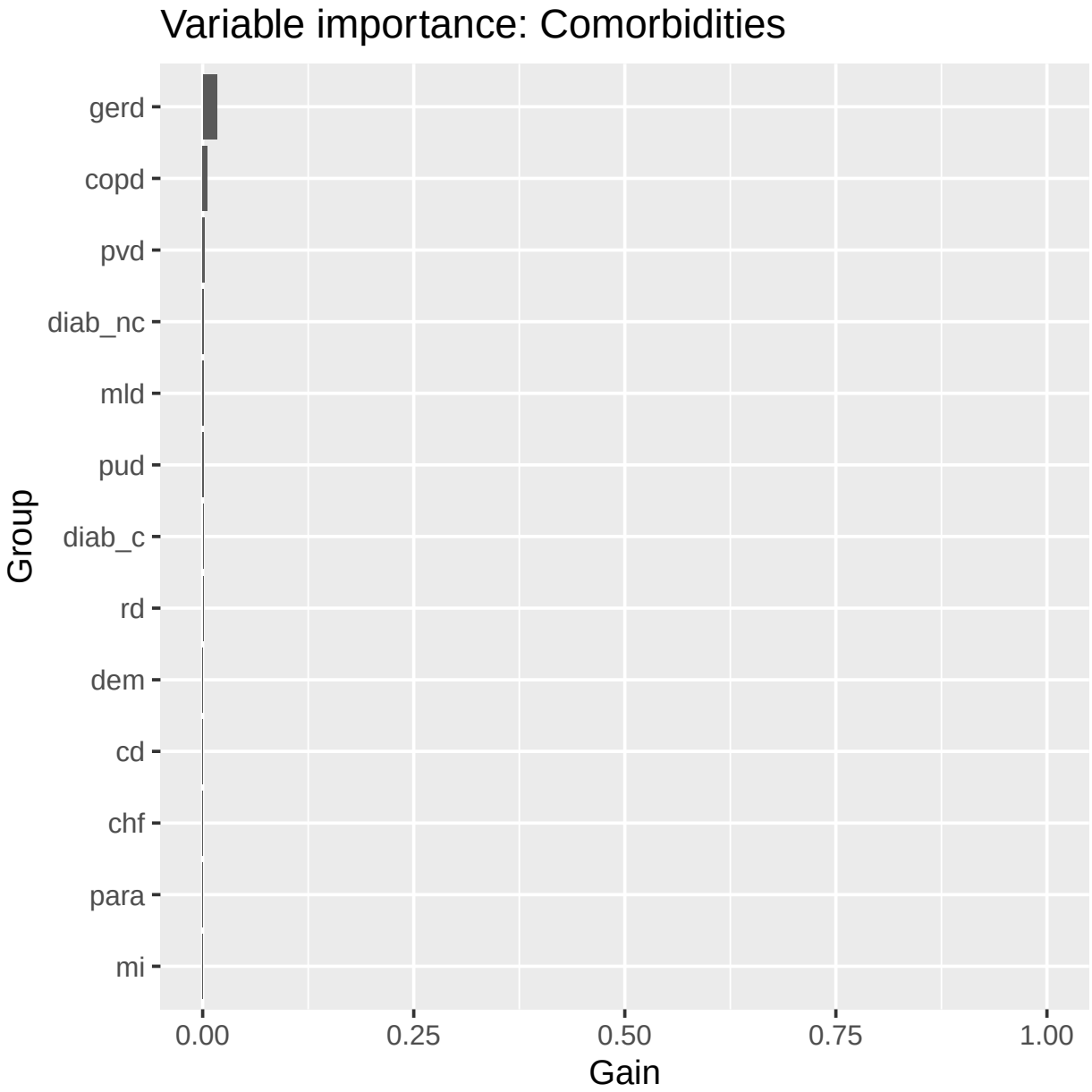
05/17/2022 update

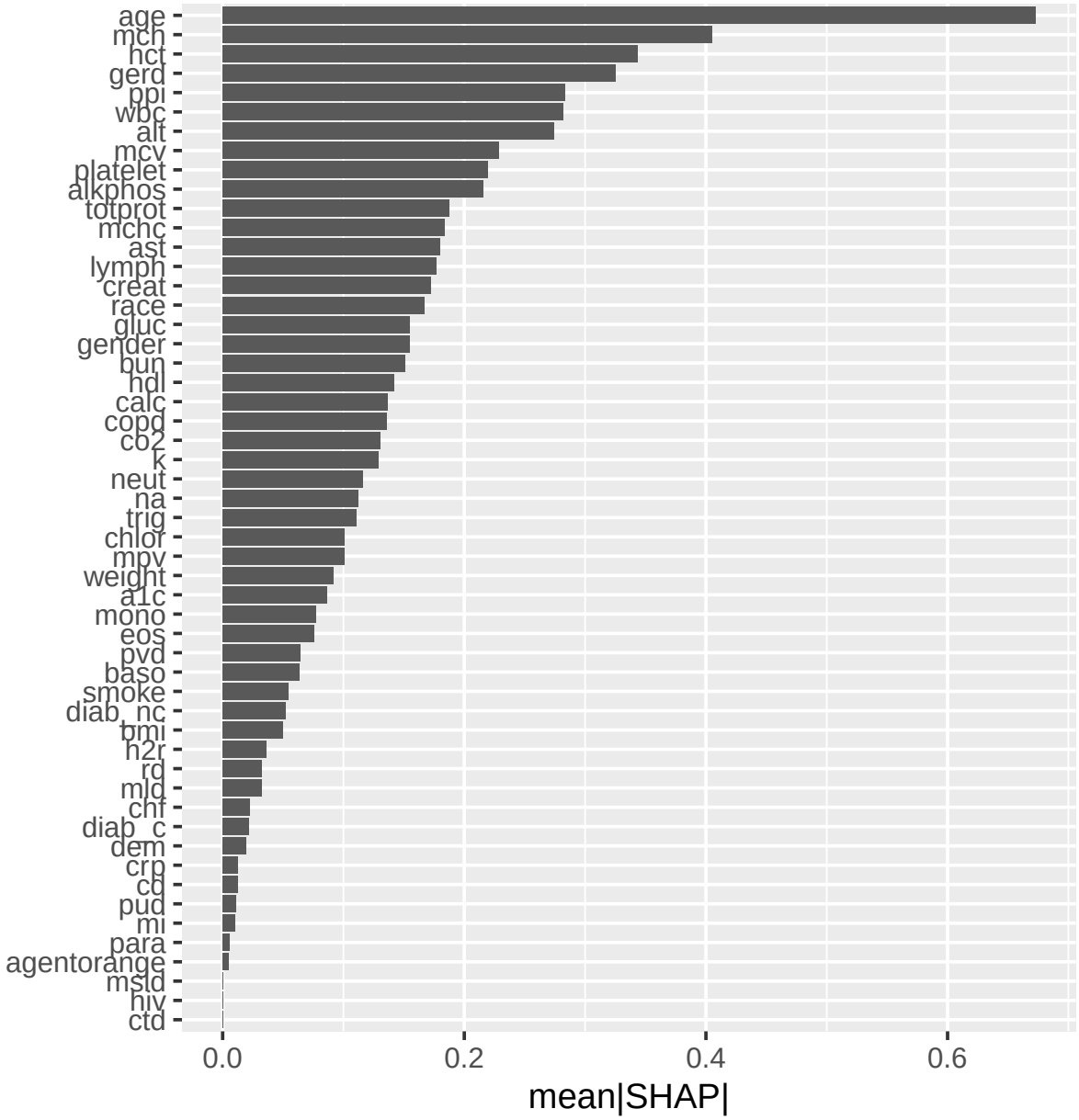
- Updated GERD calculation
- Refitted model with new GERD
  - Improved Test AUC (0.85→0.90)
  - Increased contribution of GERD, stronger effect

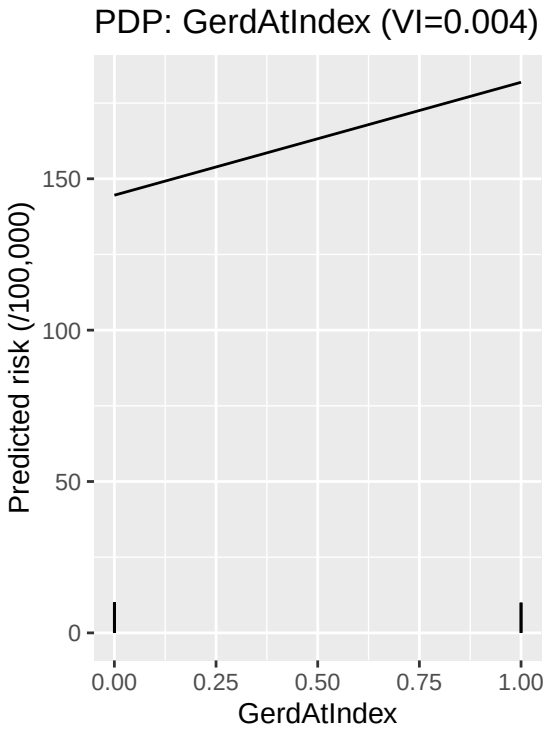


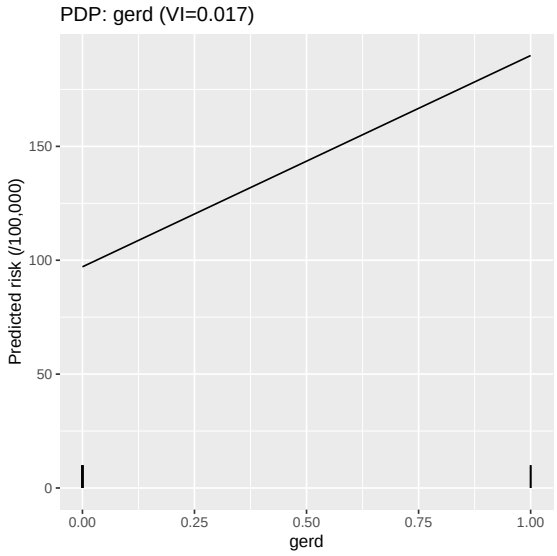

Cancer type: ANY
Sensitivity
1 − Specificity
0.00
0.25
0.50
0.75
1.00
AGA2011
AGA_CPU2022
ASGE2019
ESGE2017
ACG2022
ACG2016
BSG2013
ACP2012
method
guideline
HUNT
Kunzmann
xgboost

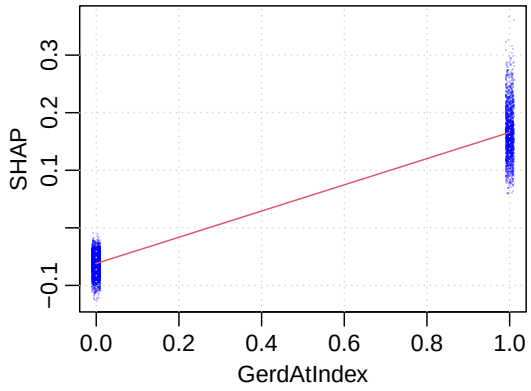

Variable importance by group
Feature group
wbc
mch
hct
mcv
age
ppi
platelet
mchc
alt
alkphos
creat
gluc
bun
ast
totprot
calc
k
co2
hdl
na
trig
lymph
chlor
gerd
mpv
neut
mono
eos
a1c
baso
race
gender
weight
bmi
copd
h2r
pvd
diab_nc
smoke
mld
pud
diab_c
rd
dem
cd
chf
para
agentorange
mi
crp
0.00
0.02
0.04
0.06
Gain

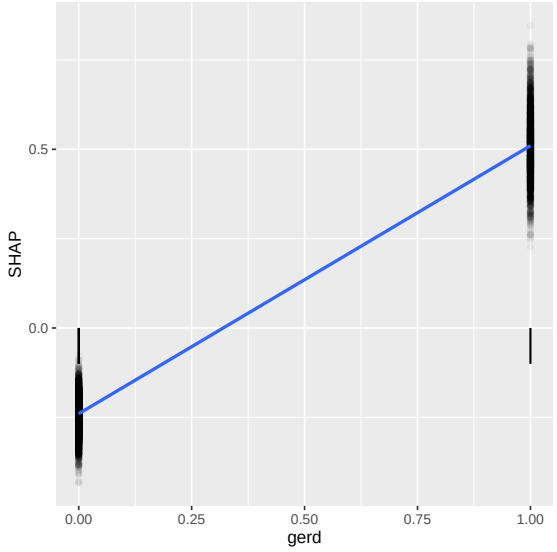

Variable importance: Comorbidities
gerd
copd
pvd
diab_nc
mld
pud
diab_c
rd
dem
cd
chf
para
mi
Group
0.00
0.25
0.50
0.75
1.00
Gain


age
mch
hct
gerd
ppi
wbc
alt
mcv
platelet
alkphos
totprot
mchc
ast
lymph
creat
race
gluc
gender
bun
hdl
calc
copd
co2
k
neut
na
trig
chlor
mpv
weight
a1c
mono
eos
pvd
baso
smoke
diab_nc
bmi
h2r
rd
mld
chf
diab_c
dem
crp
cd
pud
mi
para
agentorange
msld
hiv
ctd
0.0
0.2
0.4
0.6
mean|SHAP|


PDP: GerdAtIndex (VI=0.004)
Predicted risk (/100,000)
150
100
50
0
0.00
0.25
0.50
0.75
1.00
GerdAtIndex


PDP: gerd (VI=0.017)
Predicted risk (/100,000)
150
100
50
0
0.00
0.25
0.50
0.75
1.00
gerd

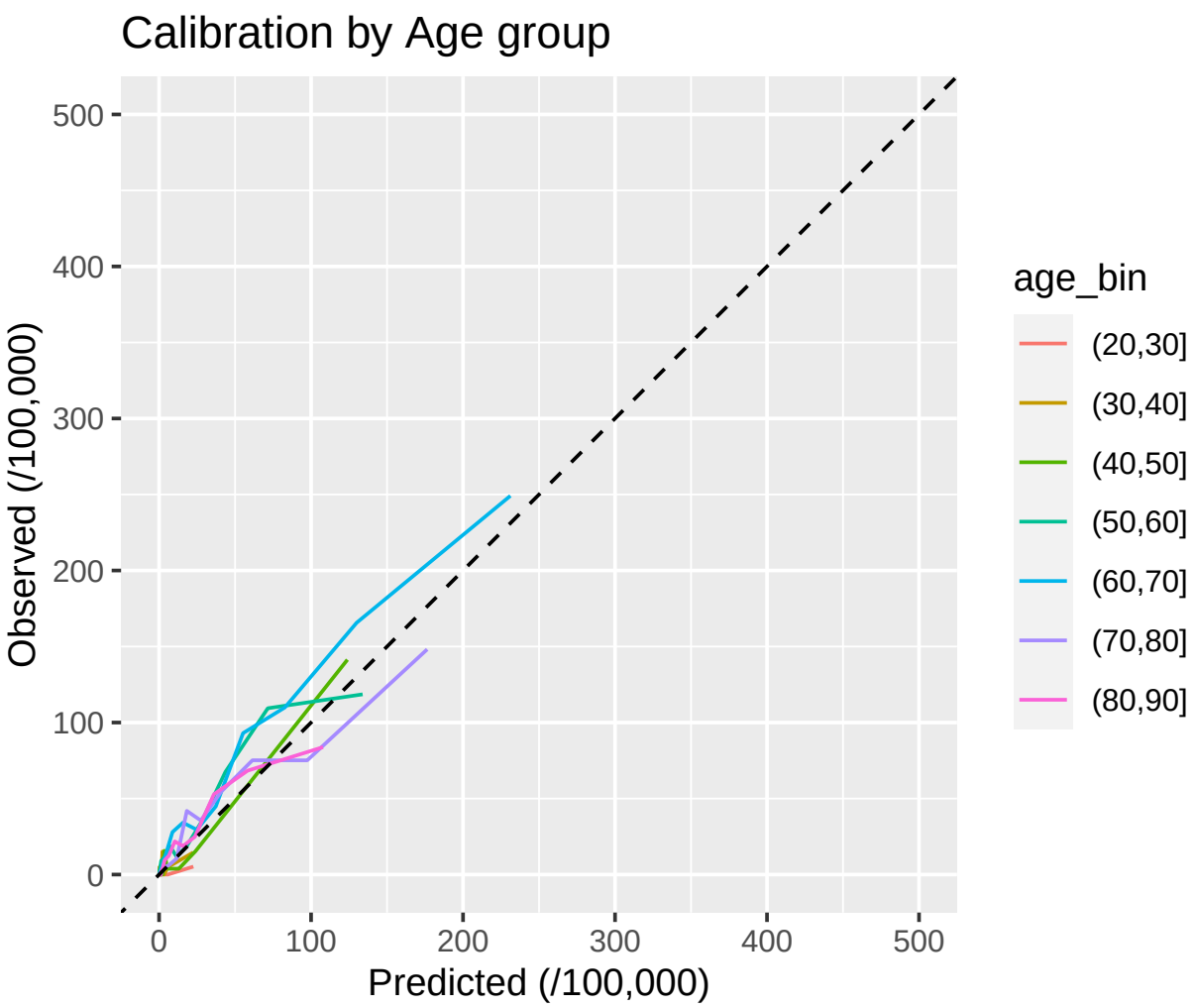

SHAP
0.3
0.2
0.1
−0.1
0.0
0.2
0.4
0.6
0.8
1.0
GerdAtIndex

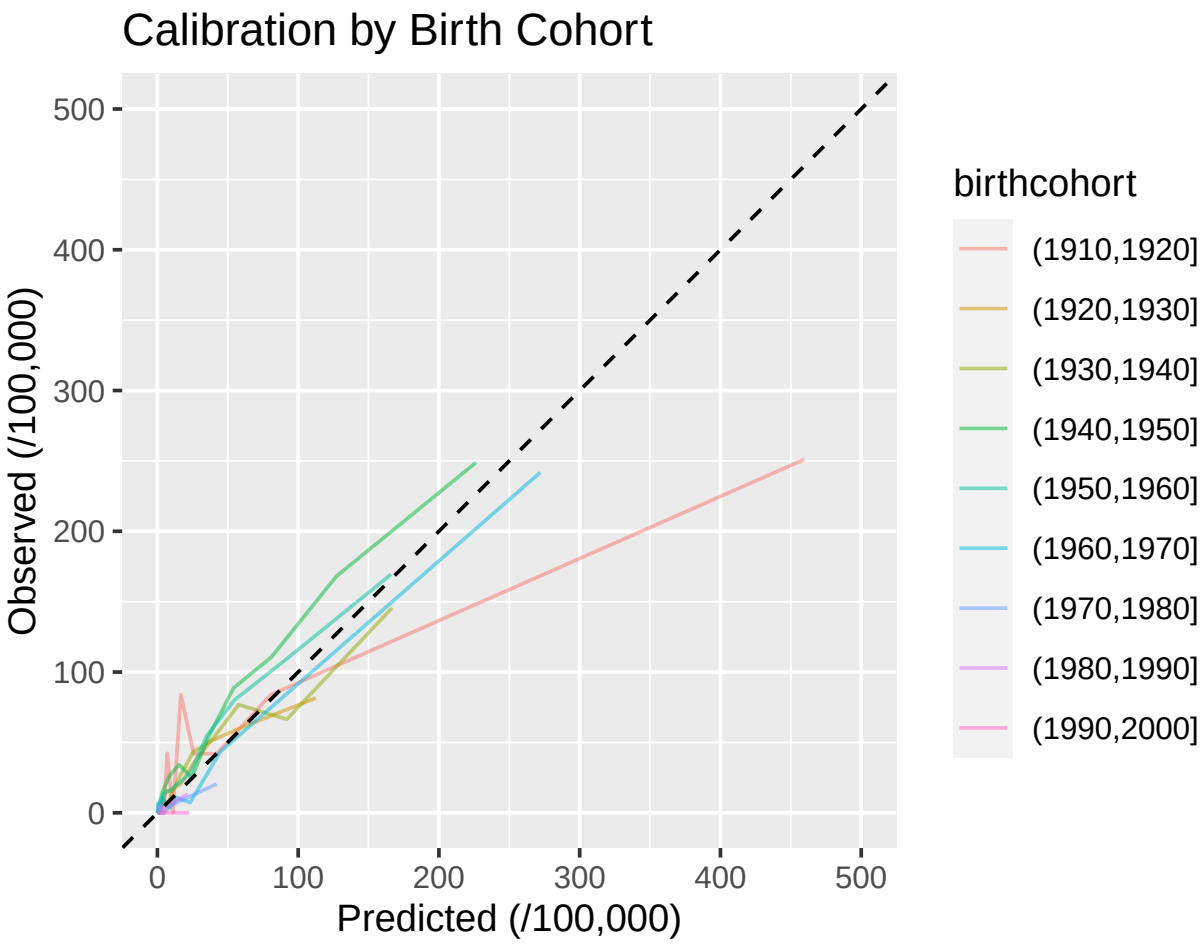

SHAP
0.5
0.0
0.00
0.25
0.50
0.75
1.00
gerd

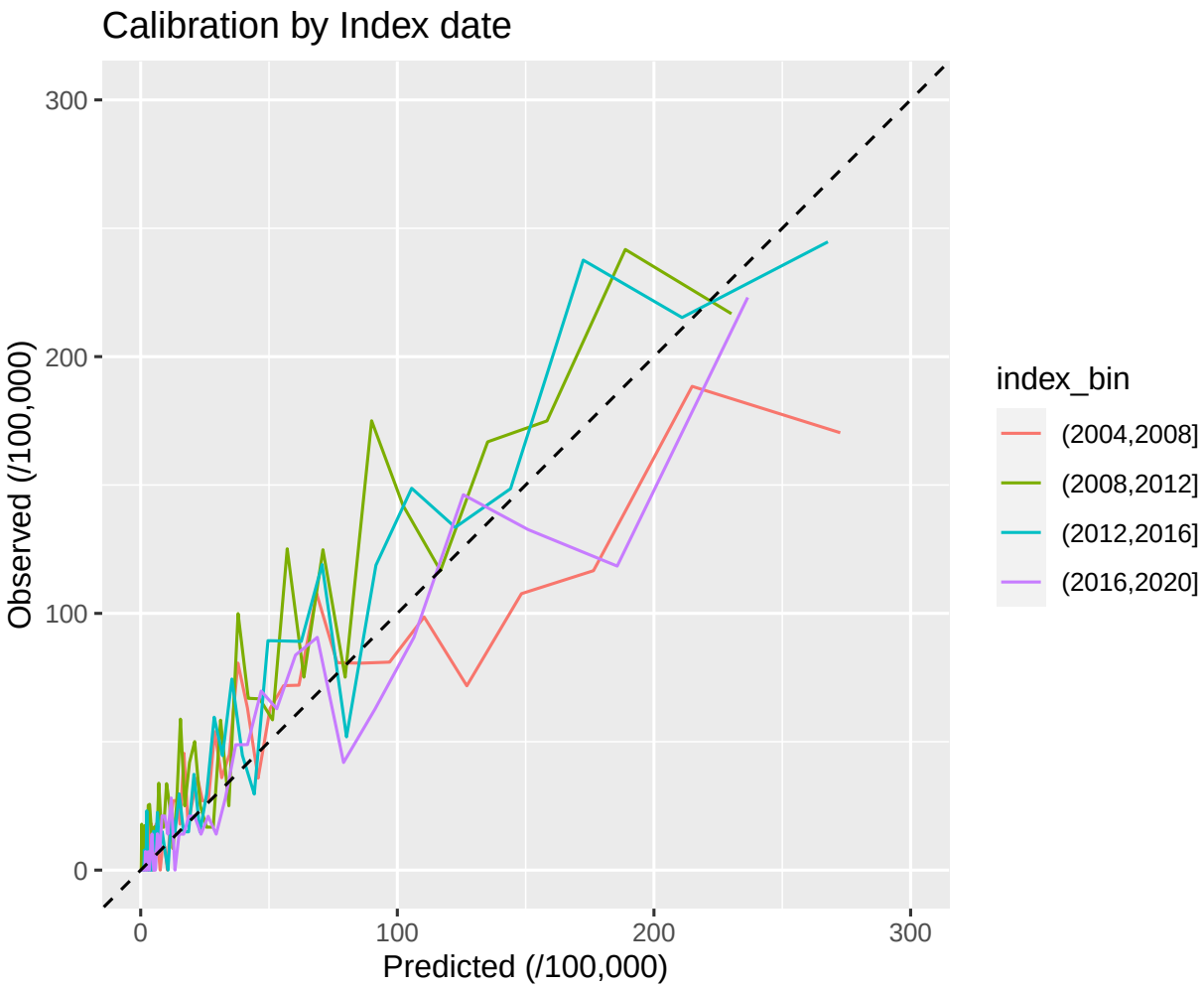
05/24/2022 update

Calibration with respect to birth cohort effect:

- Stratify by age, birth year and index date
- Predicted risk seems to be somewhat high for 70-90 yo
- Same for 1920-1930 birth cohort & 1960-1970
- Over-estimation for 2004-2008 patients, slight under-estimation for 2008-2016 patients


Calibration by Age group
Observed (/100,000)
Predicted (/100,000)
0
100
200
300
400
500
age_bin
(20,30]
(30,40]
(40,50]
(50,60]
(60,70]
(70,80]
(80,90]

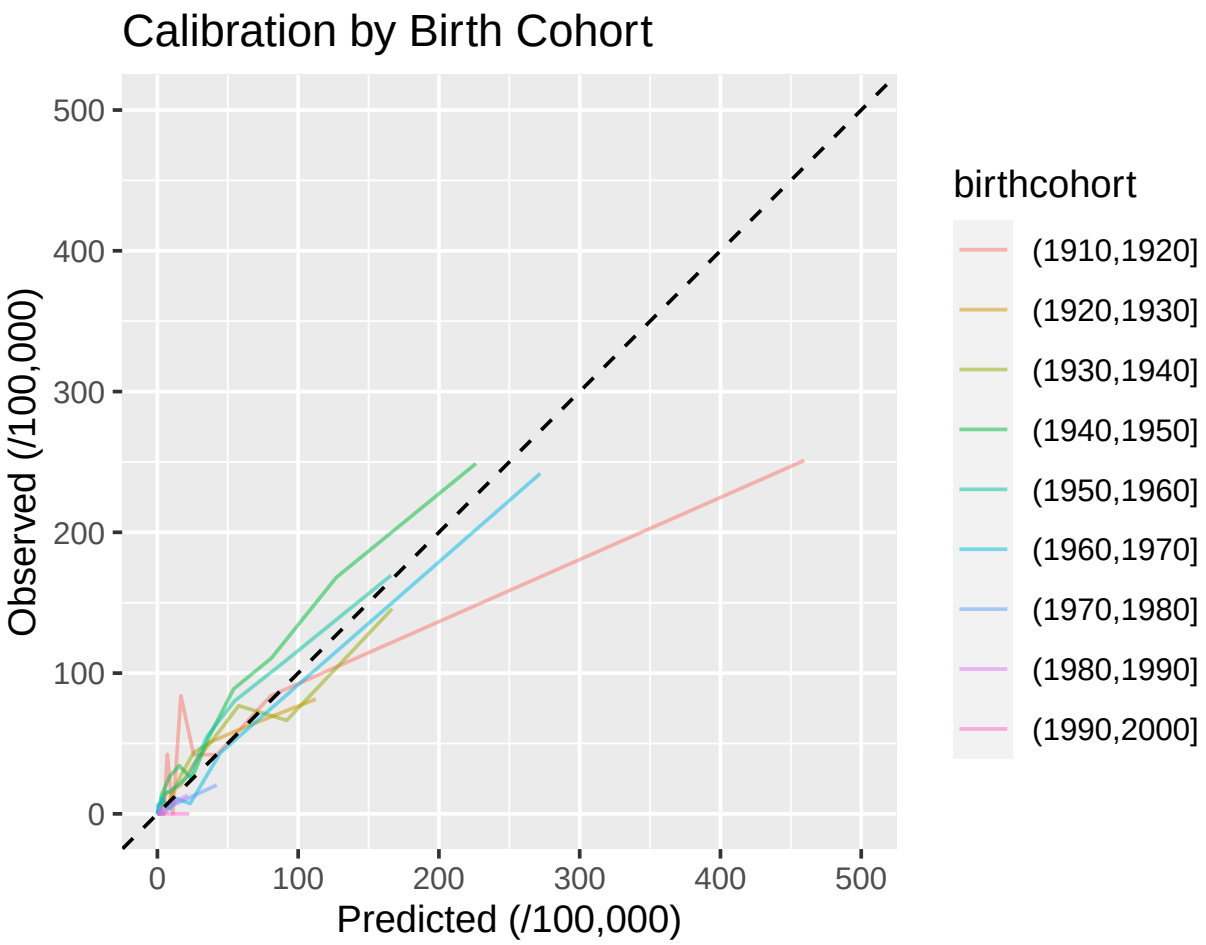

Calibration by Birth Cohort
Observed (/100,000)
Predicted (/100,000)
0
100
200
300
400
500
birthcohort
(1910,1920]
(1920,1930]
(1930,1940]
(1940,1950]
(1950,1960]
(1960,1970]
(1970,1980]
(1980,1990]
(1990,2000]

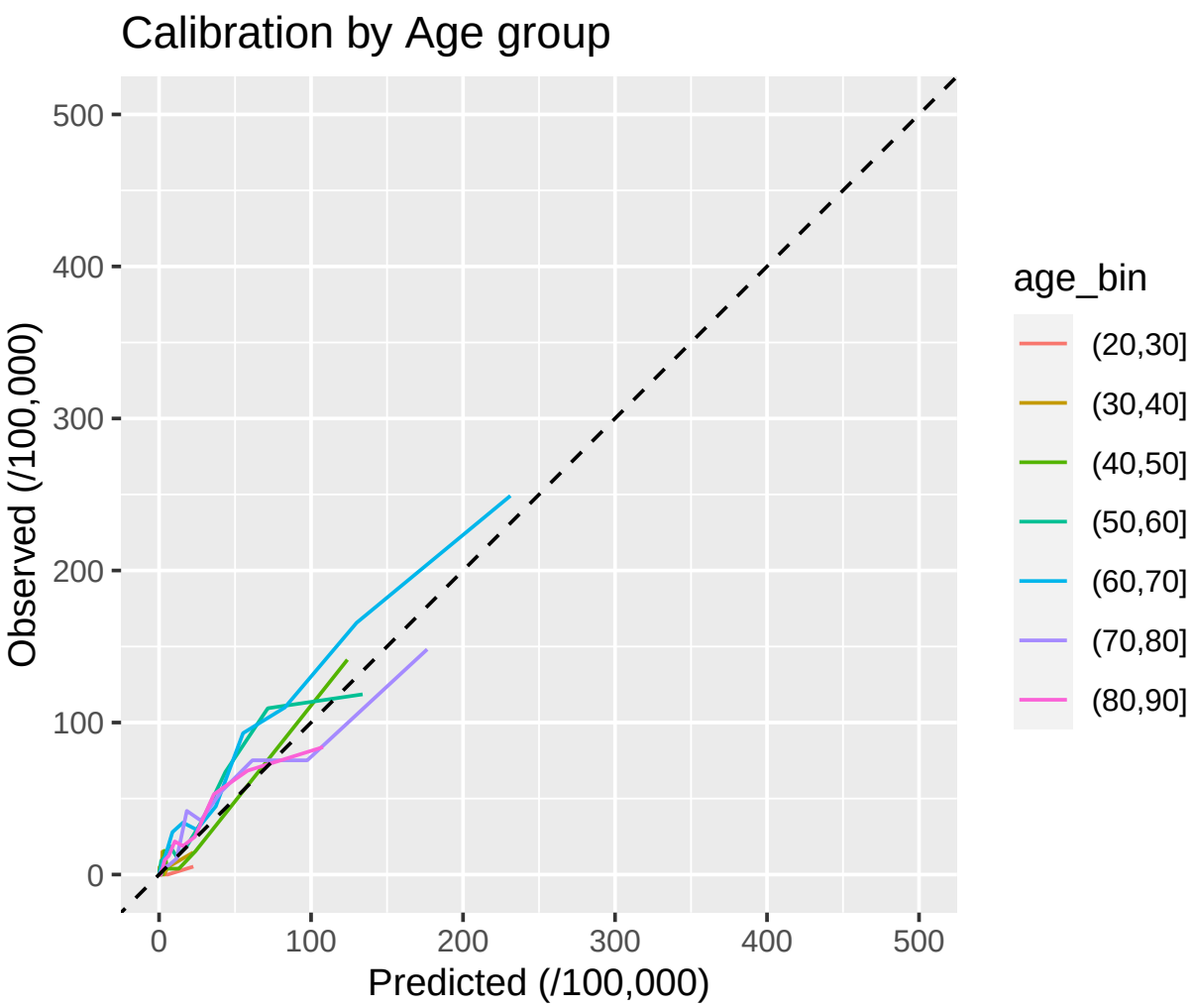

Calibration by Index date
Observed (/100,000)
Predicted (/100,000)
0
100
200
300
index_bin
(2004,2008]
(2008,2012]
(2012,2016]
(2016,2020]

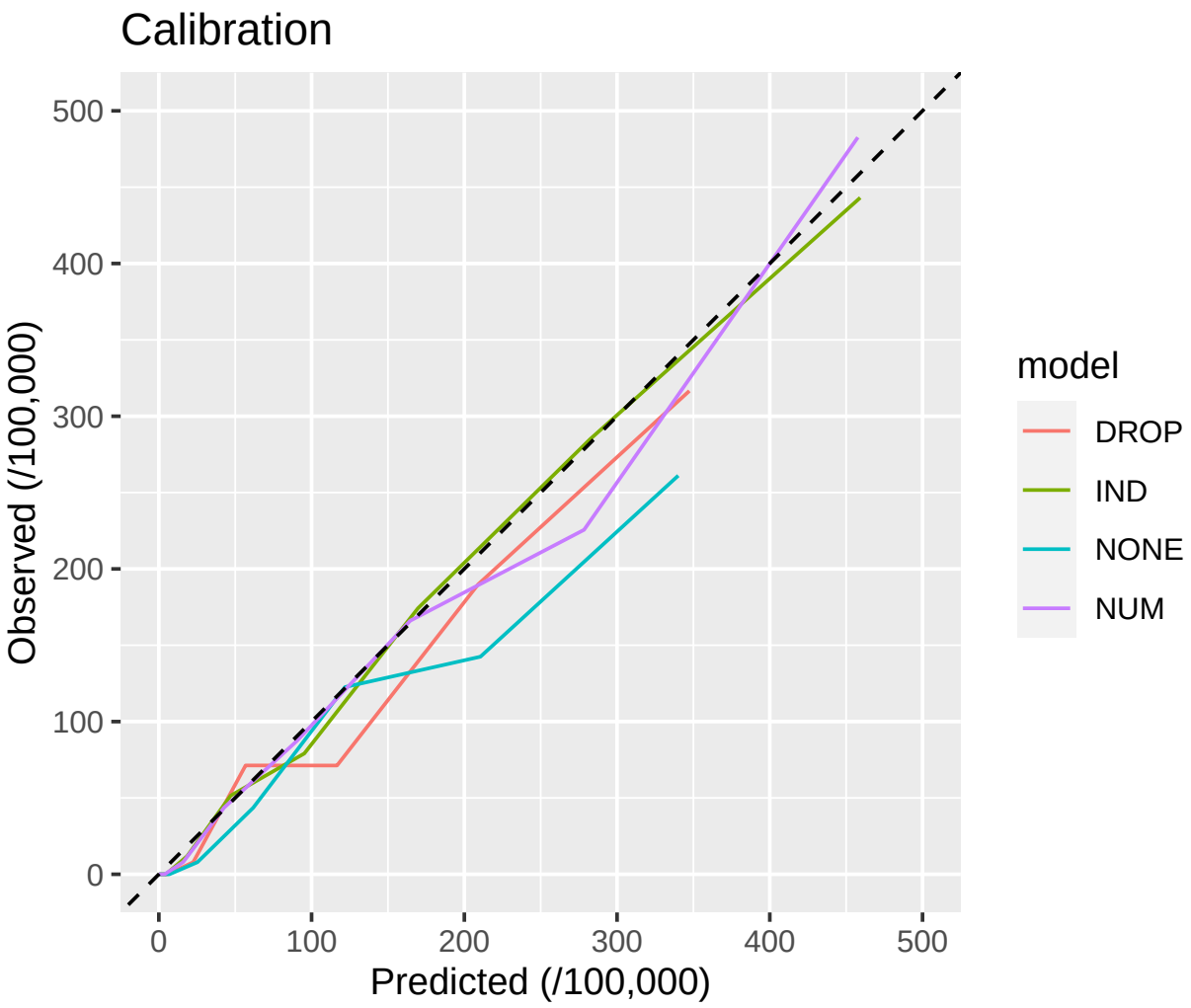
05/31/2022 update

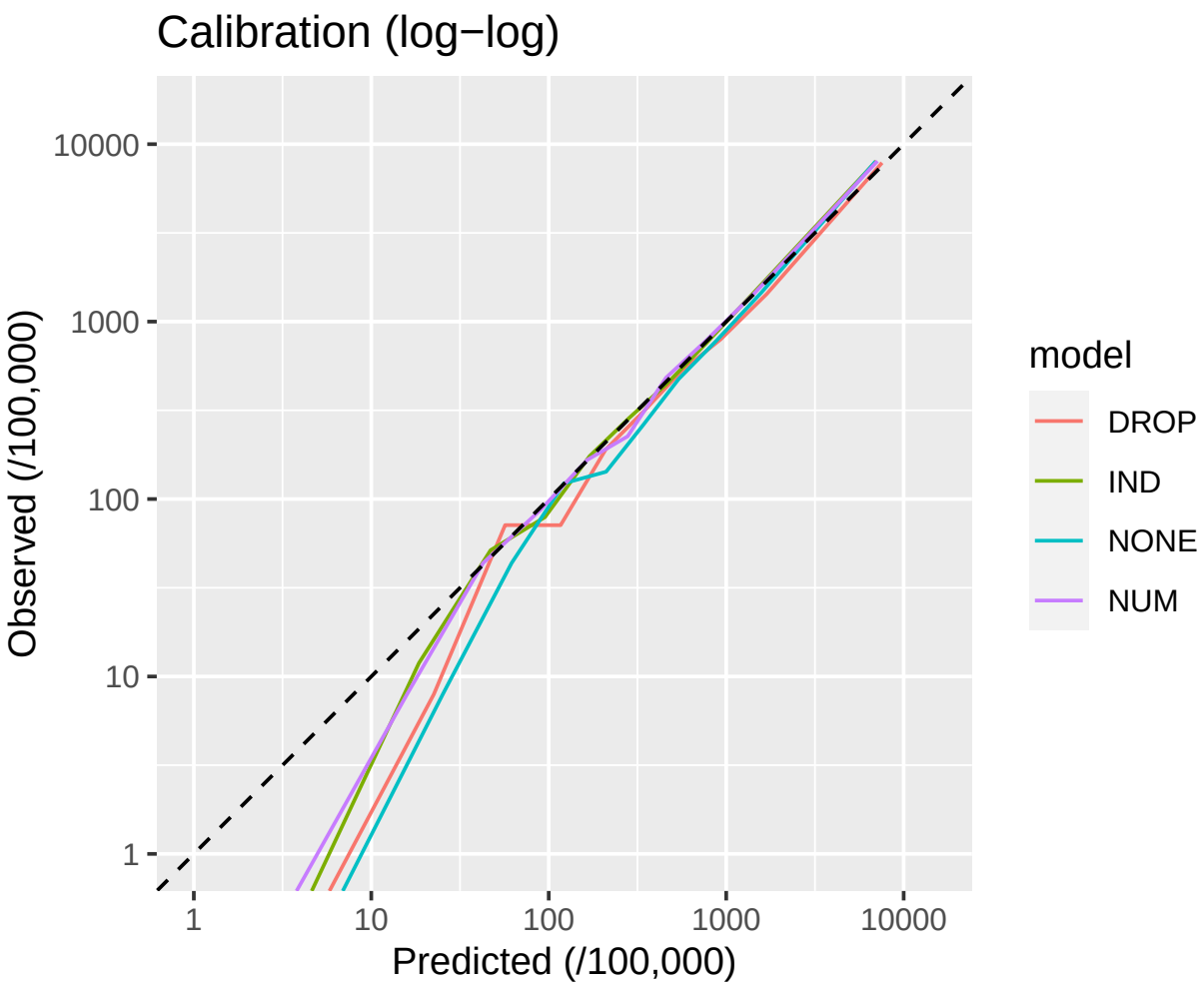
Calibration with respect to birth cohort effect:

- 10 bins to reduce noise
- Calibration for alternative models including birthyear (NONE v. IND, NUM, DROP)




Calibration by Age group
Observed (/100,000)
Predicted (/100,000)
0
100
200
300
400
500
age_bin
(20,30]
(30,40]
(40,50]
(50,60]
(60,70]
(70,80]
(80,90]

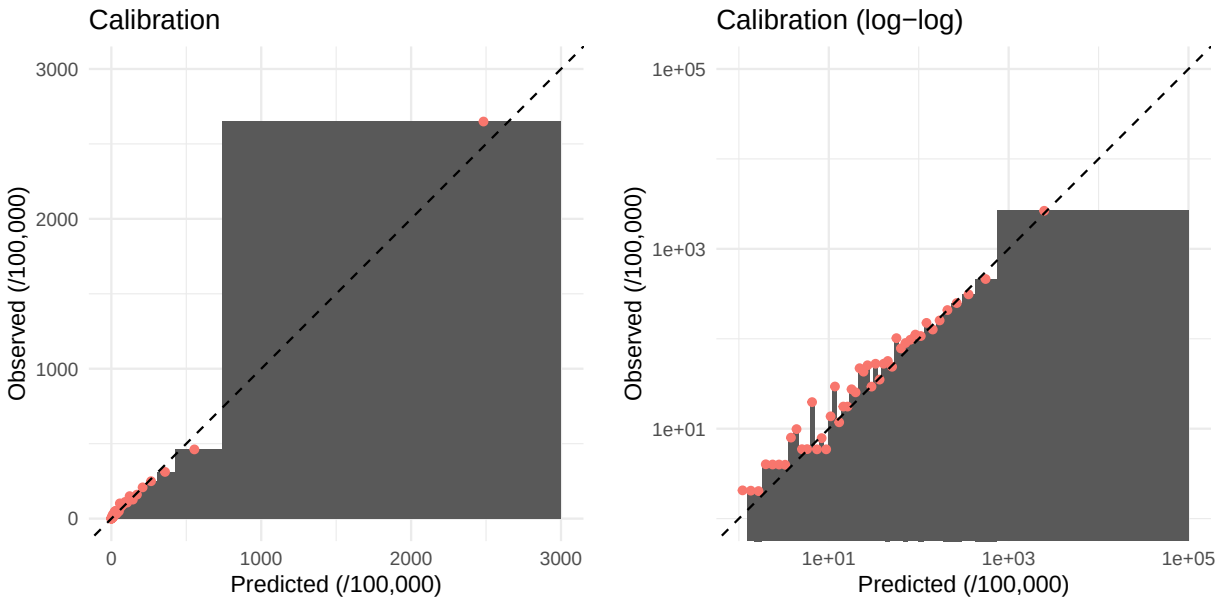

Calibration
Observed (/100,000)
Predicted (/100,000)
0
100
200
300
400
500
model
DROP
IND
NONE
NUM

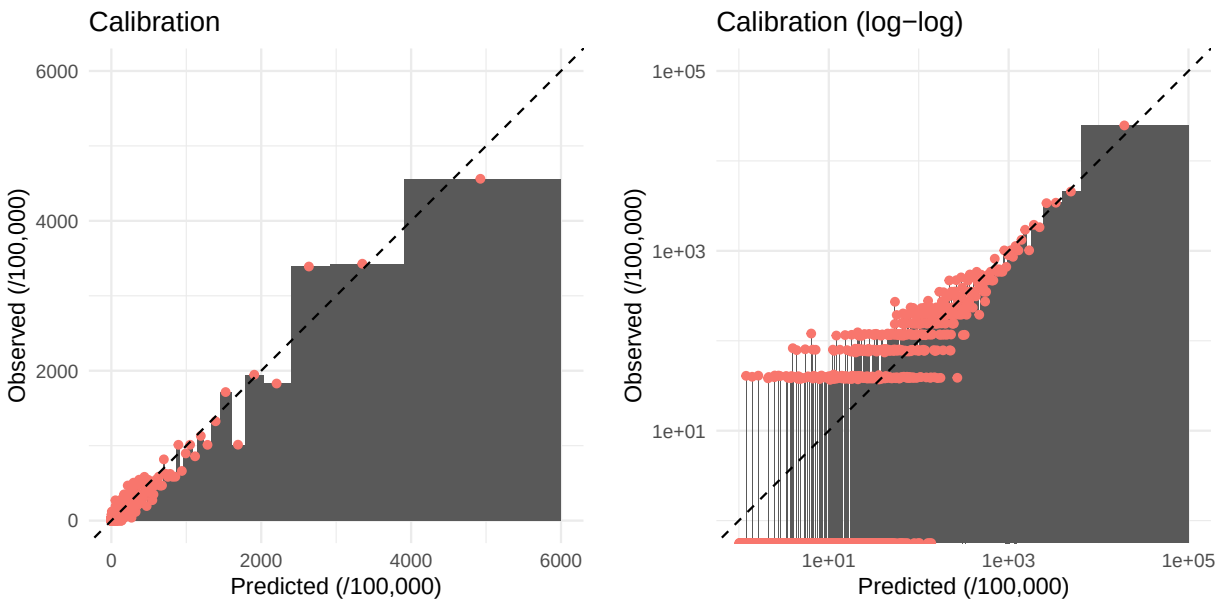

Calibration (log−log)
Observed (/100,000)
Predicted (/100,000)
10000
1000
100
10
1
1
10
100
1000
10000
model
DROP
IND
NONE
NUM

06/07/2022 update

Calibration and reporting cut-off Summary of email exchange

- It is hard to evaluate calibration for high-risk patients
- The "last bin" is hard to display because it is large (e.g., 740-100,000 when using 50 bins)
- Looking at even finer bins, we can see that calibration is decent even up to around 1,500/100,000, but these get very noisy, so it is unclear if calibration is correct or they just line up well just by chance
- I would advise not reporting any precise score above, say, 1,000/100,000 and only report, e.g., ">1,000/100,000".

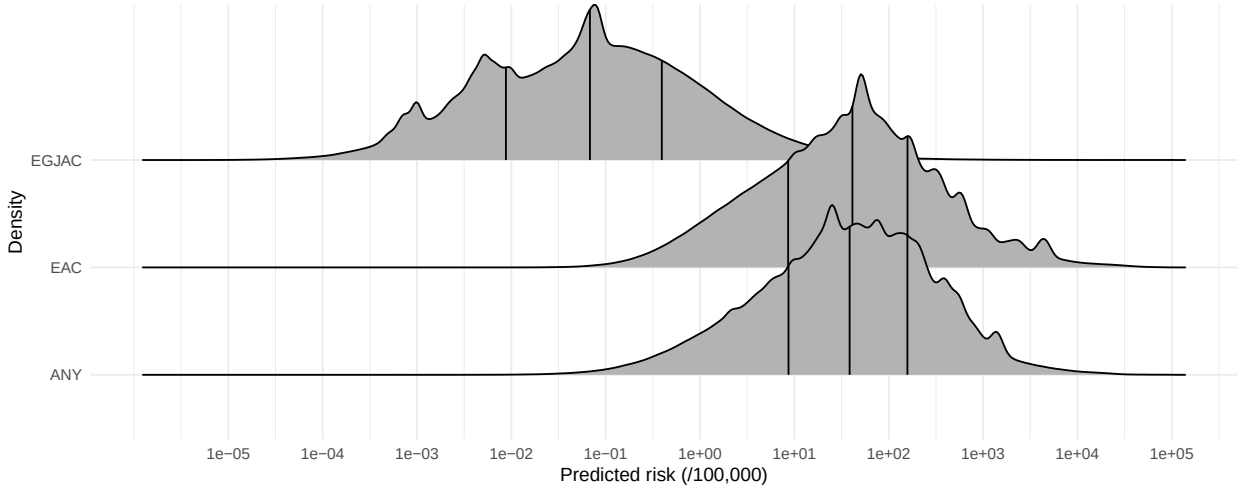


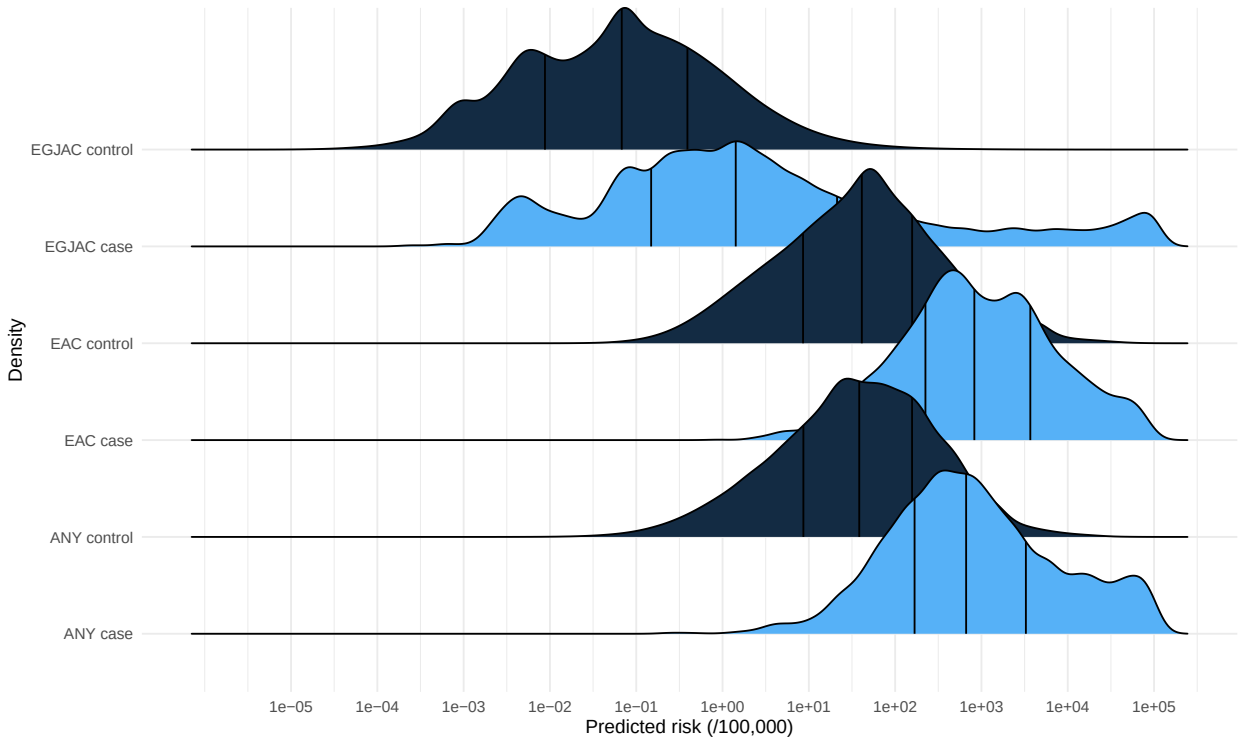

Calibration
Observed (/100,000)
Predicted (/100,000)
0
2000
4000
6000
Calibration (log−log)
Observed (/100,000)
Predicted (/100,000)
1e+01
1e+03
1e+05

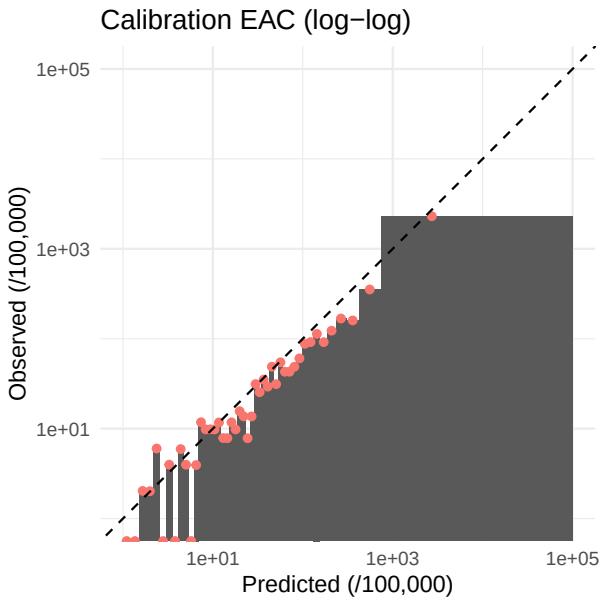
EGJAC risk scores Summary of email exchange

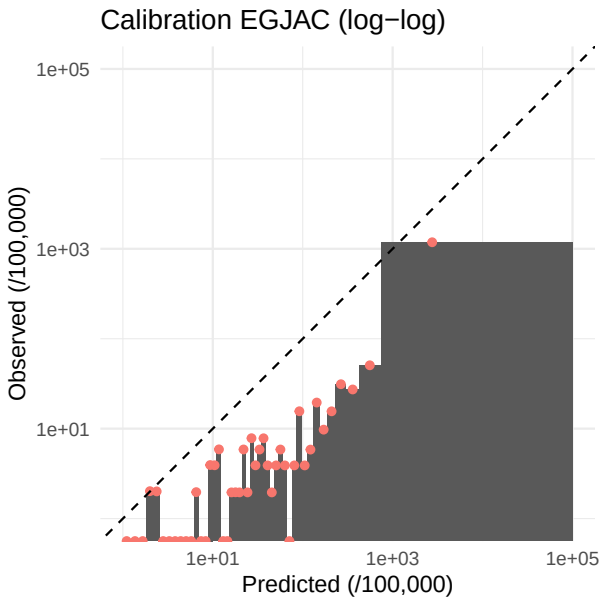
- Rich used to three models to get predicted risk for the upcoming cohort
- Joel identified that EGJAC risk scores were much lower than ANY and EAC
- I looked at how these distribution matches those in the testing set and I found very similar distributions
- This seems to indicate that the observed difference really comes from the model, not from difference in samples
- This lead me to look more into EGJAC's model and I found some serious miscalibration (also some for EAC, but way less)
- Similar behaviour as when there was imbalance between training prevlance and testing prevalance; I still don't have a good explanation for it yet.
- Splitting across case/control, we see the same shift between cases and control (about a 10x increase in risk), but EGJAC has some heavier tail, which seems to explain the better AUC for that model.

Model	n	Mean	SD	Min	Q1	Median	Q3	Max
ANY	356	0.00233	0.00879	0e+00	0.00015	0.00063	0.00177	0.12063
EAC	356	0.00678	0.02677	1e-05	0.00025	0.00094	0.00274	0.34658
EGJAC	356	0.00003	0.00020	0e+00	0.00000	0.00000	0.00001	0.00352




EGJAC control
EGJAC case
EAC control
EAC case
ANY control
ANY case
Density
1e−05
1e−04
1e−03
1e−02
1e−01
1e+00
1e+01
1e+02
1e+03
1e+04
1e+05
Predicted risk (/100,000)


Calibration EAC (log−log)
1e+05
1e+03
1e+01
Observed (/100,000)
1e+01
1e+03
1e+05
Predicted (/100,000)


Calibration EGJAC (log−log)
1e+05
1e+03
1e+01
Observed (/100,000)
1e+01
1e+03
1e+05
Predicted (/100,000)

GERD and ICD 10s Summary of email exchange

- Rich noticed that none of the 356 patients had GERD=1, but GERD should be much more common (we had aorund 20% in the original data)
- Current definition of GERD is
  - ICD9: 787.1 (Heartburn) + 530.81 (Esophageal reflux, no esophagitis)
  - ICD10: R12 (Heartburn)
  - We excluded K21.9 since it requires an EGD?
  - Does 530.81 require one?
- All new patients will only have ICD10 data
- There were no R12 in the data, so it was to be expected to have no GERD=1
- Looking at prevalance of ICD codes, we see that 530.81 constitute most of the GERD=1 in ICD9 (60:1 ratio) so it is not surprising we have no GERD=1 by dropping K21.9
- What should we do about it? It will be important to insure that all patients (training, testing and in practice) have the same definition of features, which is not the case now since most training patients did use reflux to define GERD.