Model Performance Results

Zachary Katz (UNI: zak2132)

2023-02-24

Data Import & Wrangling

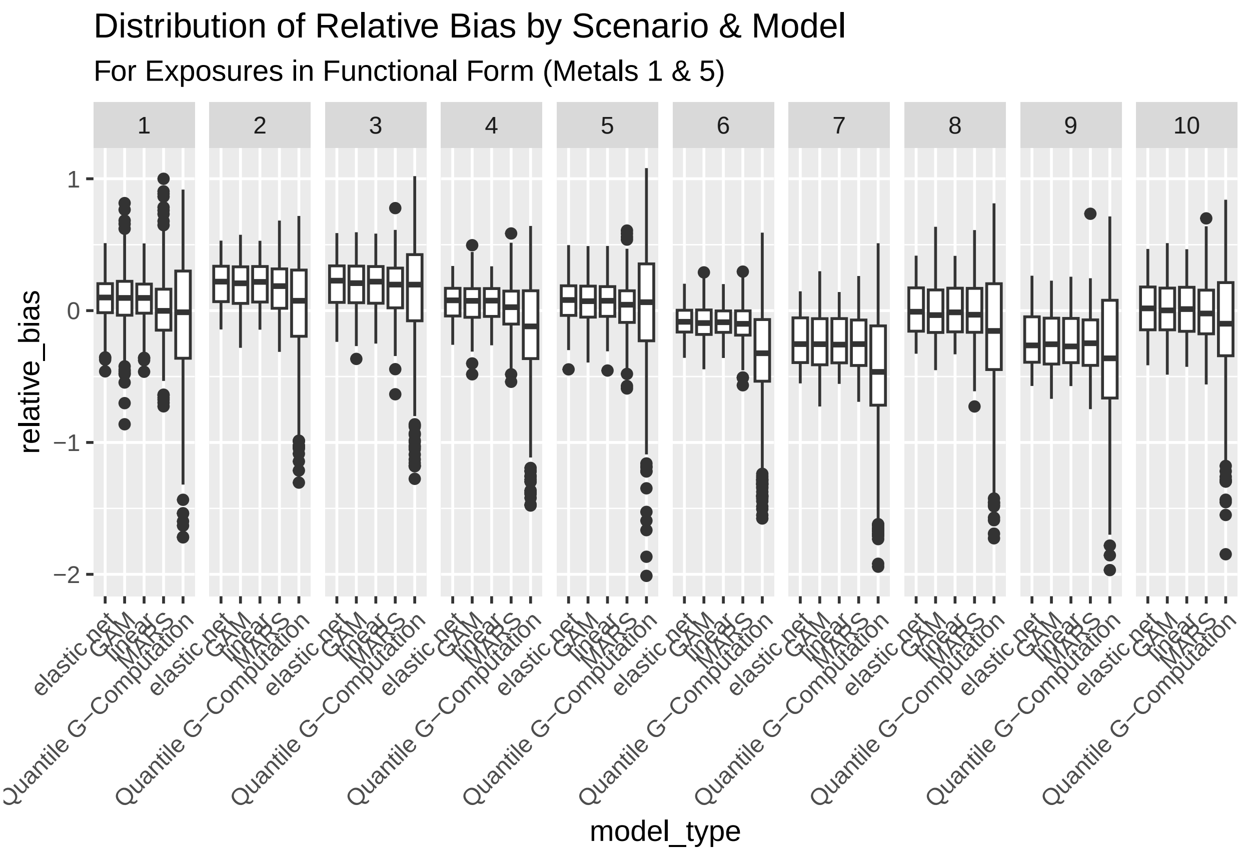
Performance Measurement

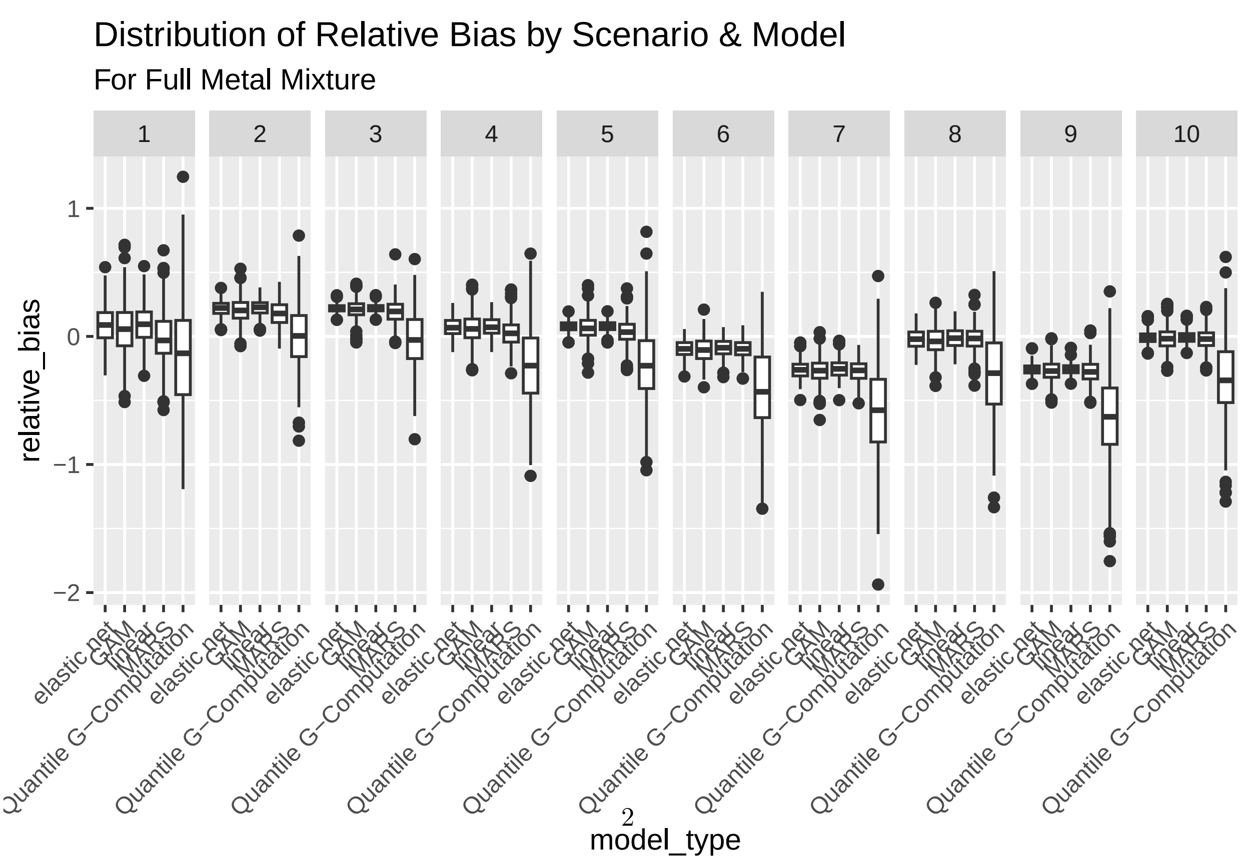
- Scenario 1: baseline (simple exposures, simple confounding)
- Scenario 2: complex exposures (nonlinear / interactions), simple confounding
- Scenario 3: complex exposures (nonlinear / interactions) + multicollinearity, simple confounding
- Scenario 4: complex exposures (trigonometric), simple confounding
- Scenario 5: complex exposures (trigonometric) + multicollinearity, simple confounding
- Scenario 6: simple exposures, complex confounding
- Scenario 7: complex exposures (nonlinear / interactions), complex confounding
- Scenario 8: complex exposures (trigonometric), complex confounding
- Scenario 9: complex exposures (nonlinear / interactions) + multicollinearity, complex confounding
- Scenario 10: complex exposures (trigonometric) + multicollinearity, complex confounding

Note: besides confounders, only exposure metals 1 and 5 are specified in the functional form for each scenario. "Metal" 11 is the complete metal mixture.

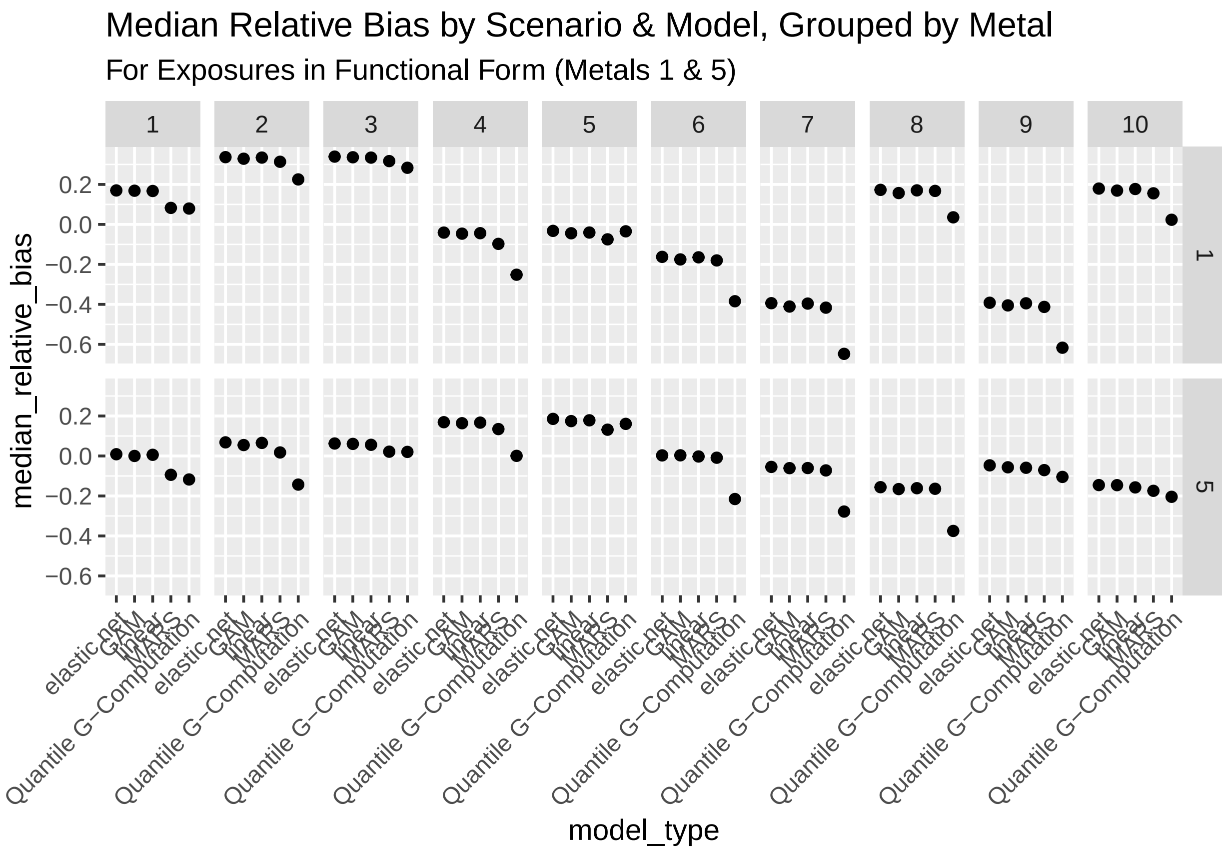
Relative Bias

Data Visualizations

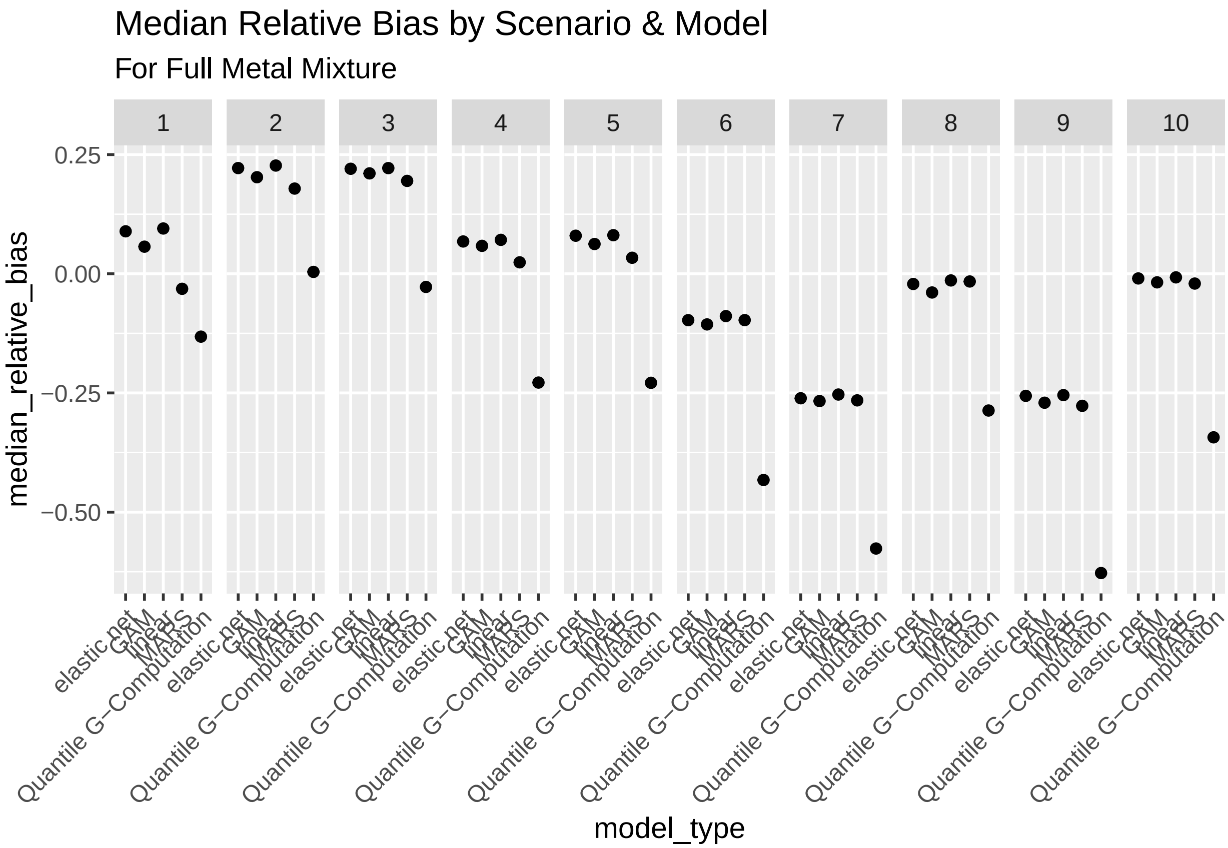




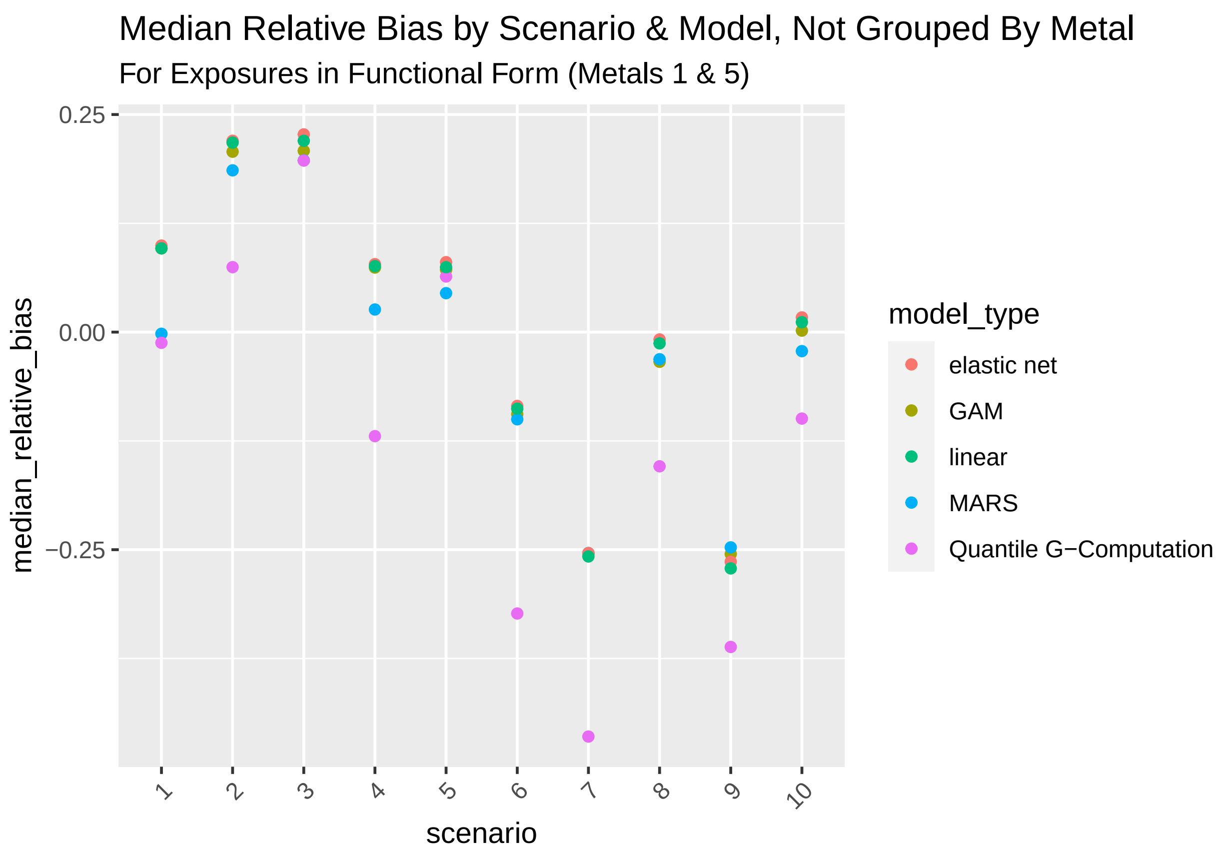
```
## 'summarise()' has grouped output by 'scenario', 'metal'. You can override using
## the '.groups' argument.
```




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




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

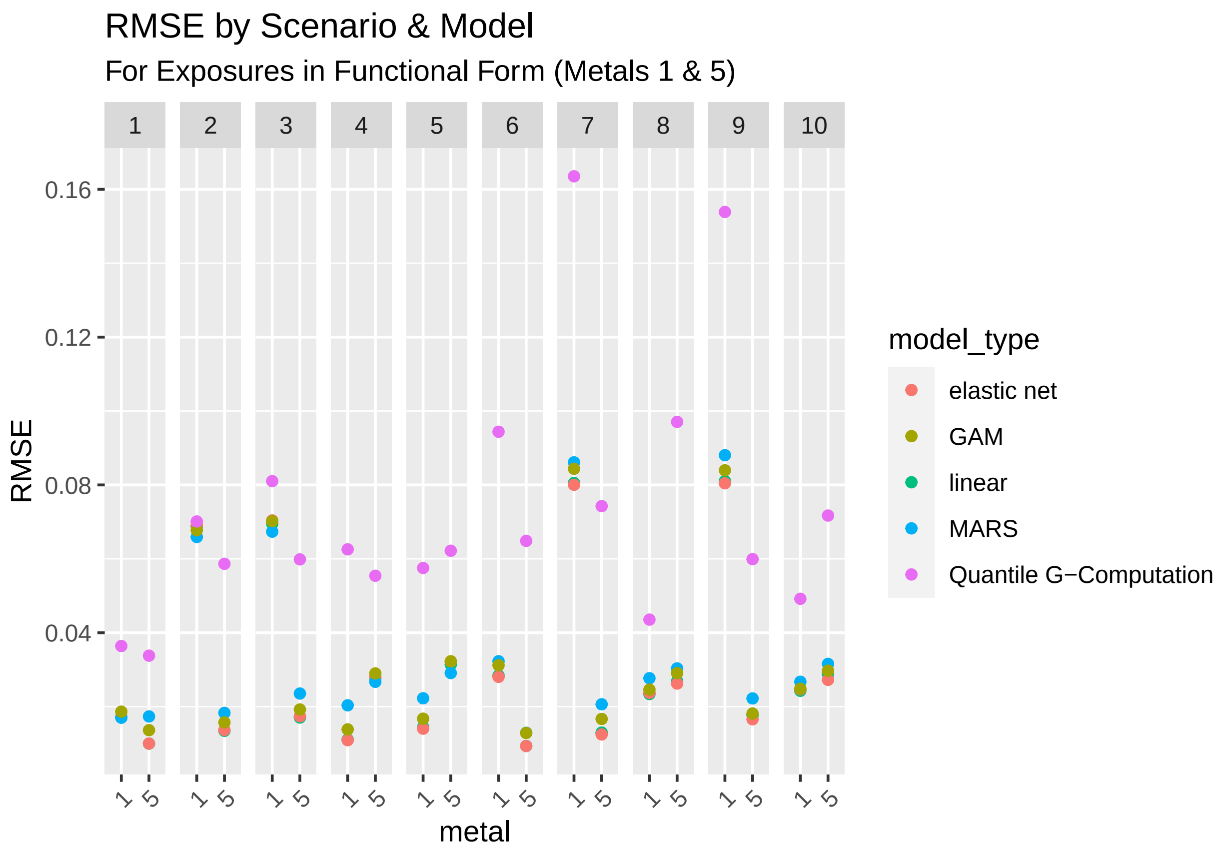


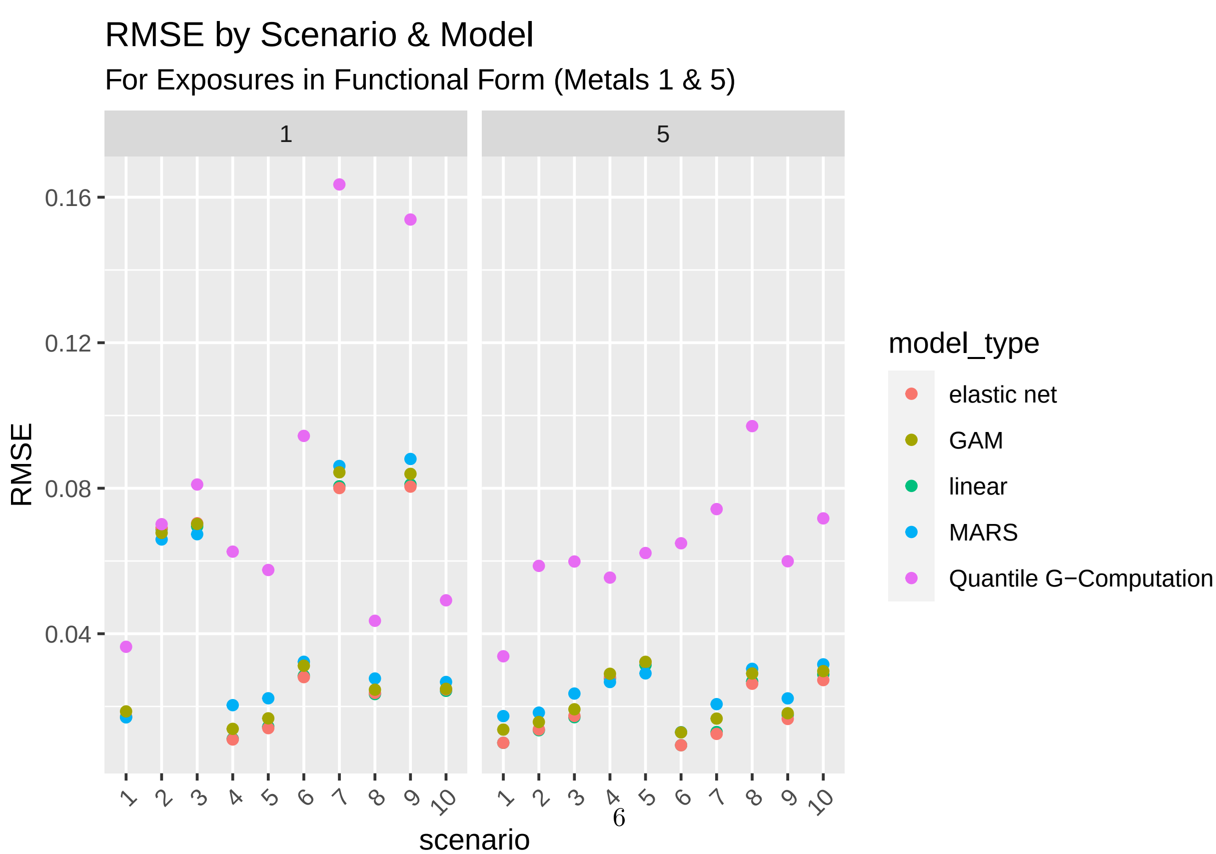
Observations

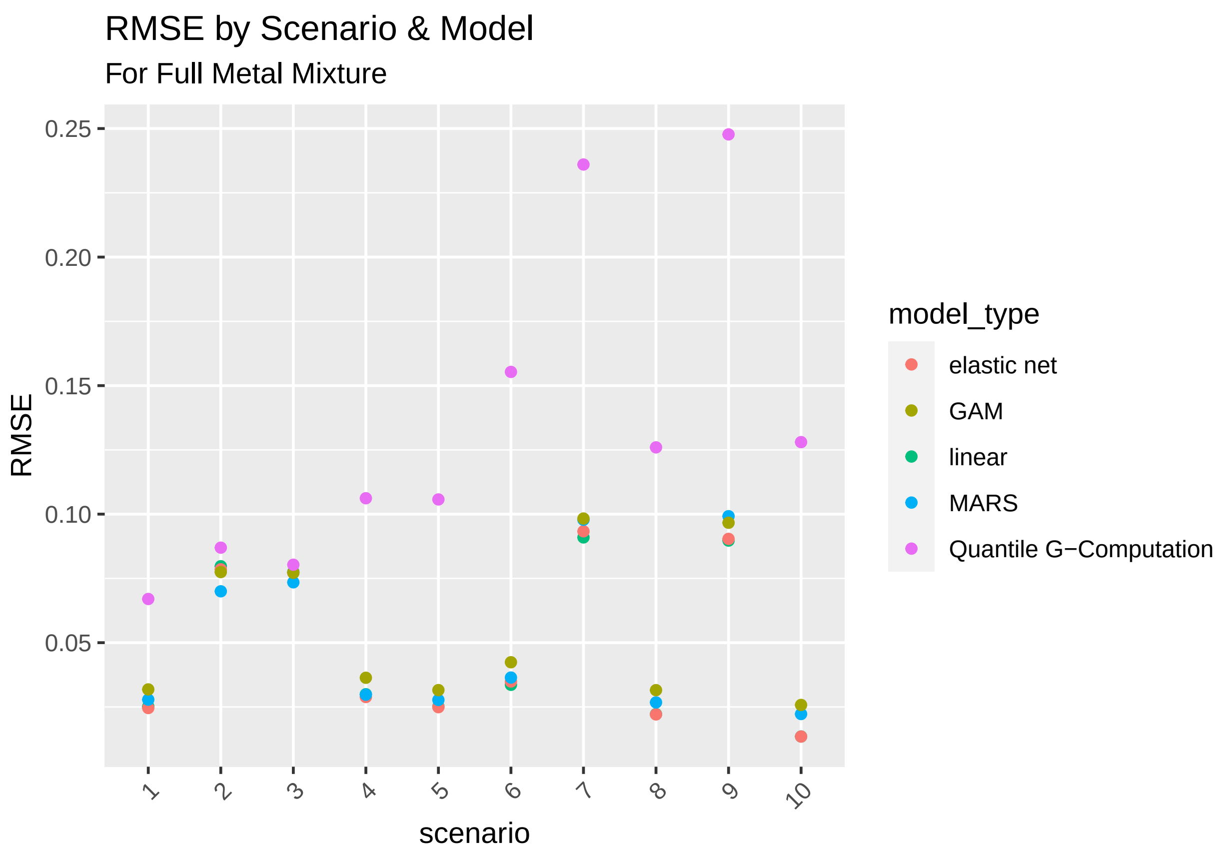
- Positive relative bias (effect is overestimated) in scenarios 1-5, negative relative bias (effect is underestimated) in scenarios 6-10
- Most variability in effect estimates for quantile g-computation
- Scenarios 4, 5, and 6 are less challenging for the model than anticipated (4 and 5 in particular)
- Scenarios 2 and 3; 4 and 5; 7 and 9; 8 and 10 tend to see similar performance, suggesting possible redundancy
- Scenarios 7 and 9 appear to be more challenging for the model than 8 and 10
- Scenarios 2 and 3 appear to be more challenging for the model and 4 and 5
- Multicollinearity tends to substantially increase effect size estimates for quantile g-computation – but not for aggregate mixture profiles
- Scenarios 8 and 10 are easy for aggregate mixtures
- Linear, elastic net, and GAM models tend to perform quite similarly; MARS tends to be a bit better, and quantile g-computation sees substantially more instability

RMSE

Data Visualizations



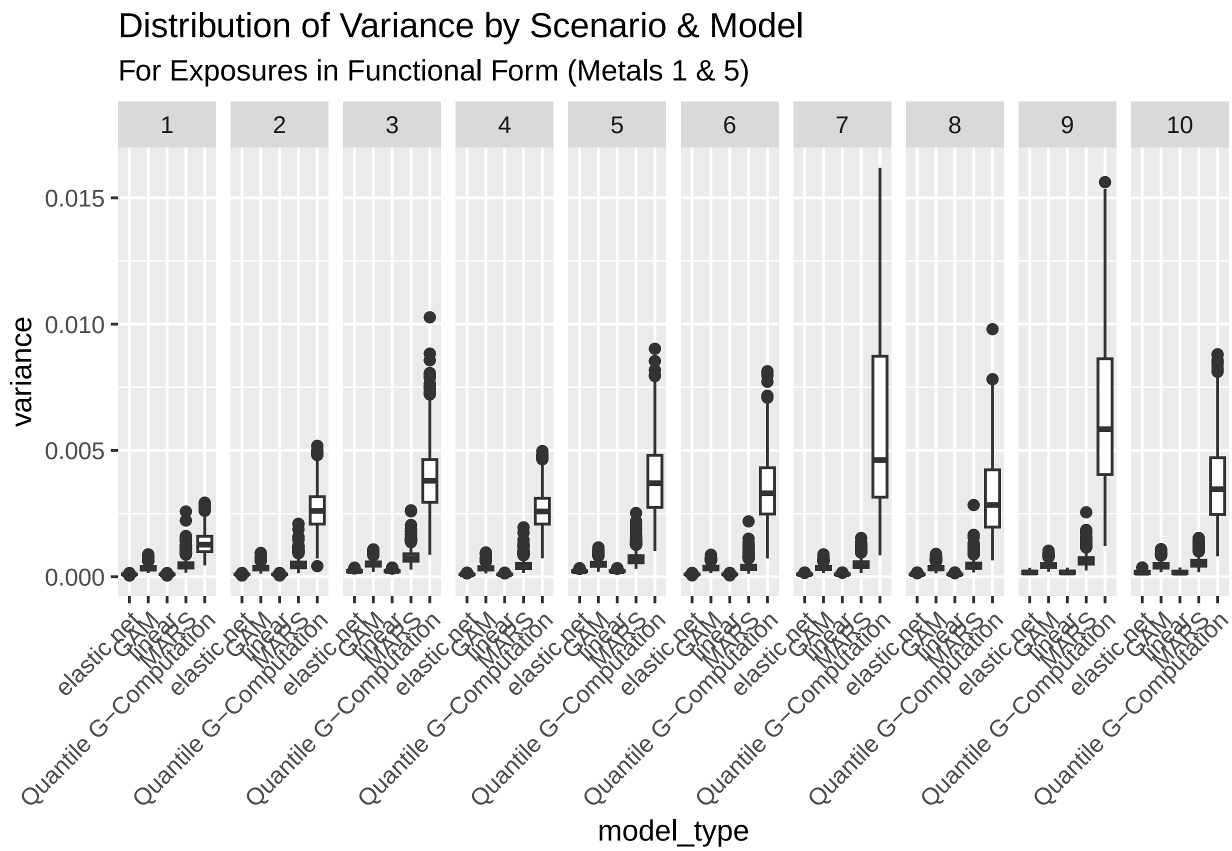


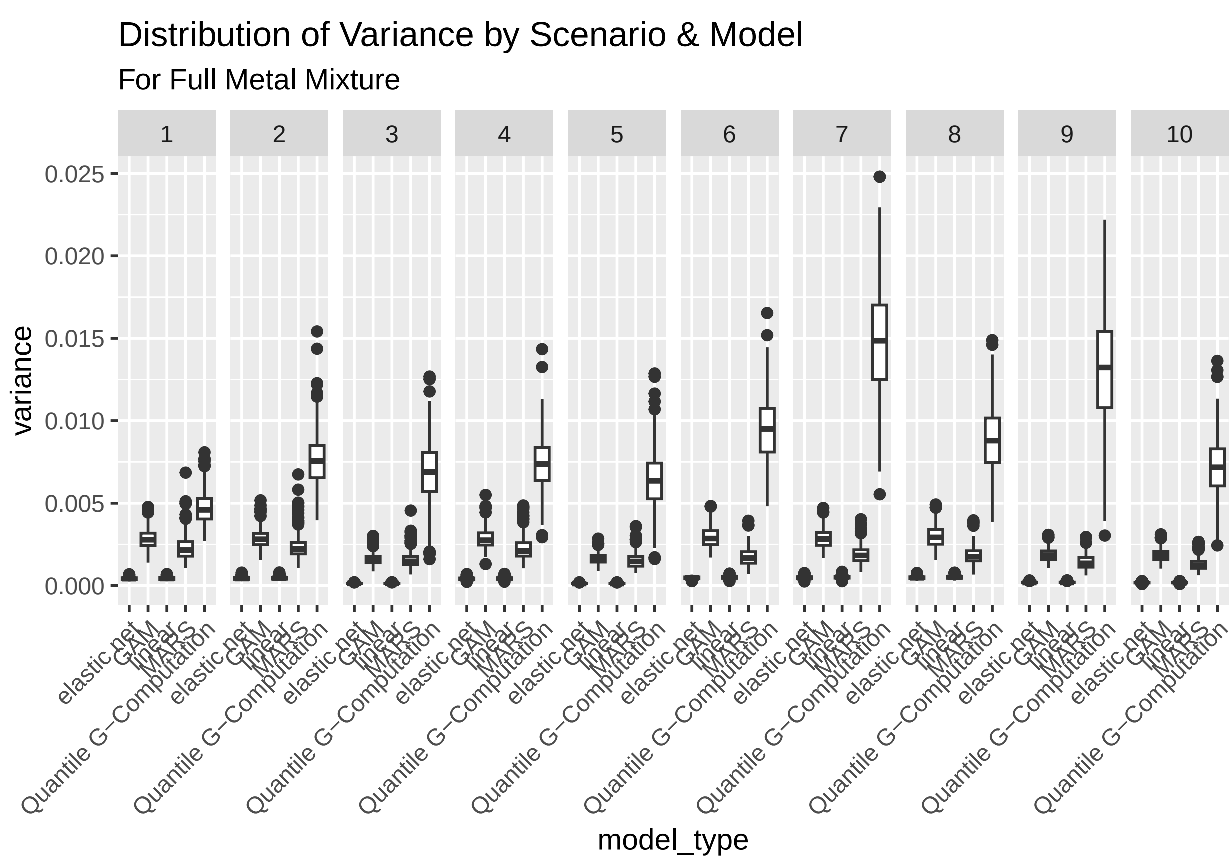


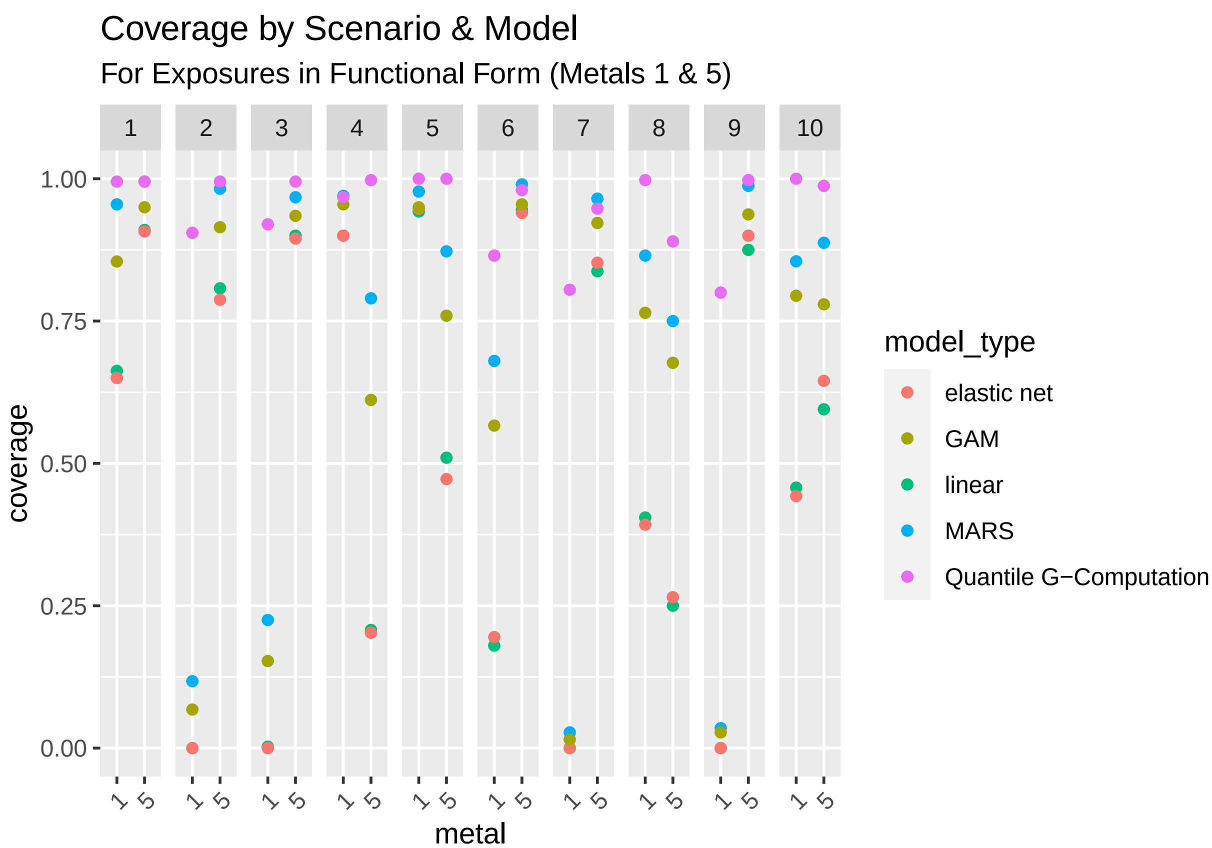
Observations

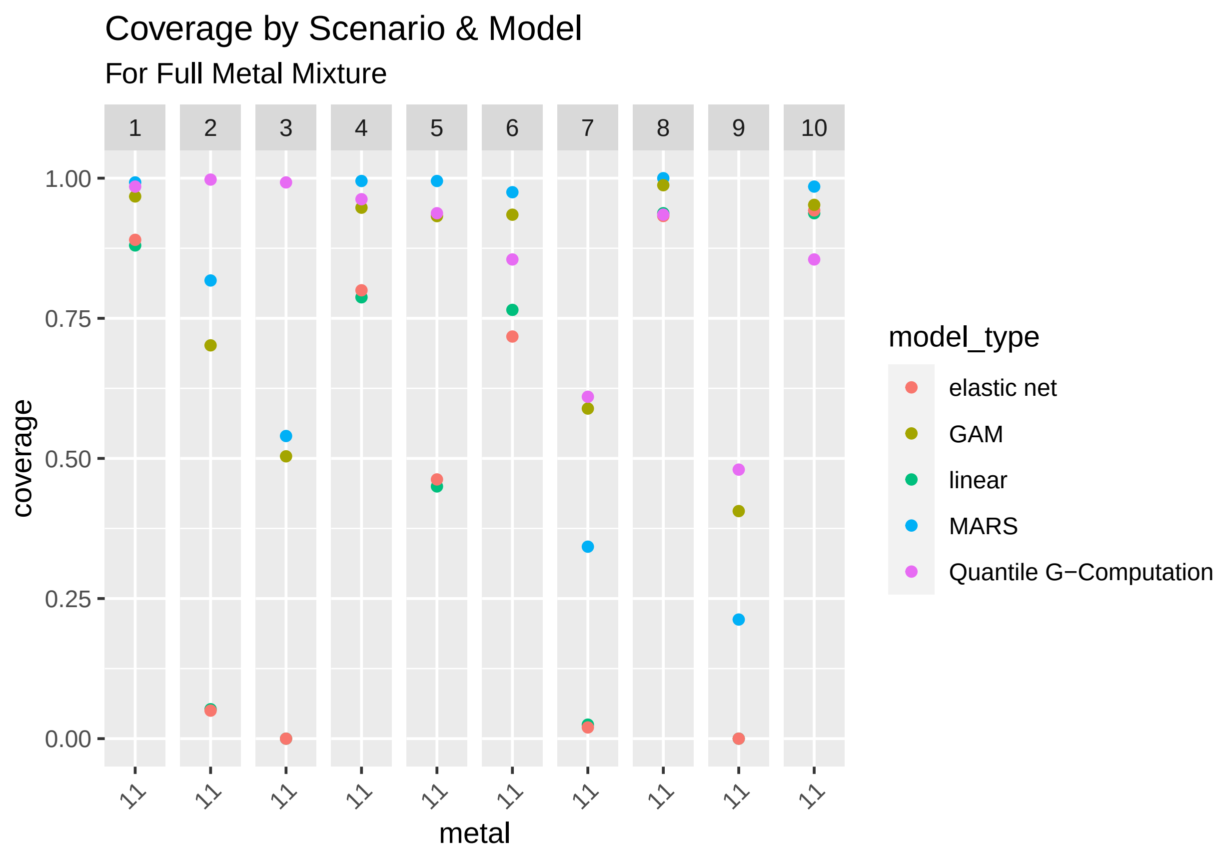
- For metal 1, RMSE lowest for scenarios 1, 4, and 5, whereas for metal 5, RMSE lowest for 1, 2, 6, 7, and 9
- For metal 1, RMSE highest for scenarios 2, 3, 7, and 9, whereas for metal 5, RMSE highest for scenarios 4, 5, 8, and 10
- RMSE estimates quite similar for scenarios 2 and 3, 4 and 5, 7 and 9, and 8 and 10, again suggesting some redundancy
- RMSE estimates much higher for quantile g-computation than for other models, even in vanilla scenario(s)
- MARS models also tend to have slightly higher RMSE than linear, elastic net, and GAM models
- For metal mixture, scenarios 1, 4, 5, 6, 8, and 10 see relatively low RMSE, whereas scenarios 2, 3, 7, and 9 see higher RMSE
- Elastic net and MARS models seems to demonstrate generally best performance across scenarios for the mixture profile, whereas quantile g-computation again fares worse

Other Measures






Coverage by Scenario & Model
For Exposures in Functional Form (Metals 1 & 5)
1
2
3
4
5
6
7
8
9
10
coverage
1.00
0.75
0.50
0.25
0.00
metal
model_type
elastic net
GAM
linear
MARS
Quantile G−Computation



Analysis

Descriptive

- Which scenarios are supposed to be less challenging for the models to estimate? *1, 2, and 4*
- Which scenarios are in actuality less challenging for the models to estimate (i.e. seem to have lowest relative bias and RMSE)? *4, 5, 6, and perhaps 1*
- Which scenarios are supposed to be more challenging for the models to estimate? *3, 5, 9, and 10*
- Which scenarios are in actuality more challenging for the models to estimate (i.e. seem to have highest relative bias and RMSE)? *2, 3, 7, and 9*
- Which scenarios are supposed to be moderately challenging for the models to estimate? *6, 7, and 8*
- Which scenarios are supposed to be relatively similar to each other? *2 and 4; 3 and 5; 7 and 8; 9 and 10*
- Scenarios we thought would be easy that actually are easy: *1 and 4*
- Scenarios we thought would be easy that actually are challenging: *2*
- Scenarios we thought would be challenging that actually are challenging: *3 and 9*
- Scenarios we thought would be challenging that actually are easy: *None (potentially 6)*
- Scenarios that perform similarly to each other we assumed would perform similarly to each other: *None*

- Scenarios that perform similarly to each other we did not assume would perform similarly to each other: *2 and 3; 4 and 5; 7 and 9; 8 and 10*

Performance Insights

- Multicollinearity between the exposures does not seem to make much of a difference when it comes to model performance (perhaps because limited variable selection implemented?)
- Functional forms specified nonlinearly (polynomial) and with interaction terms tend to be more difficult than those specified trigonometrically
- Complex confounding seems to add some – but not *too* much – of challenge for model estimation / performance
- General lack of stratification in performance between linear, elastic net, and GAM models may indicate that our functional forms are a bit too "easy" (specify more exposures?) – but perhaps this is acceptable

Recommendations

- Scenarios we should probably drop: 4, 5, 8, and 10
- Either find a way to make multicollinear scenarios more difficult, or drop scenarios 3 and 9
- For scenarios 6 and 7, find a way to introduce more complex confounding?
- Introduce more complex functional forms, with more exposures?

Appendix

```
## ‘summarise()‘ has grouped output by ’scenario’. You can override using the
## ‘.groups‘ argument.
## ‘summarise()‘ has grouped output by ’scenario’. You can override using the
## ‘.groups‘ argument.
## ‘summarise()‘ has grouped output by ’scenario’. You can override using the
## ‘.groups‘ argument.
## ‘summarise()‘ has grouped output by ’scenario’. You can override using the
## ‘.groups‘ argument.
## ‘summarise()‘ has grouped output by ’scenario’. You can override using the
## ‘.groups‘ argument.
```