Bayesian Tri-gression

Provider-Level Analysis of Type I, Type II, and Recovery Joint Outcomes

(Tri-gression is pronounced very similarly to *"turducken"*)

1 Caveats

- Analysis was used as an opportunity to explore the capabilities of *PyMC3* (I had not before used this tool for regression or missing value imputation)
- The targeted delivery was a comprehensive uncertainty quantification framework – feature engineering and ML predictive modeling were not points of emphasis

2 Summary of Work

Table 1: dead ends, things left unaddressed, results, and future errata.

Explored	Unexplored	Delivered	Known Issues
Probit regression for error-level analysis: *numeric problems*	Hierarchical modeling for provider-level heterogeneity in measurement precision	Bayesian inference for provider-level, tri-gression	MVN predictive conditional distribution is approximated
Missing data imputation during modeling fitting: *missingness overloading*	Supervised learning framework orientation	Full joint dependency & uncertainty estimation	Only a barebones treatment of feature engineering has been provided
PCA-based regression	Σ-centric specification for correlation inference	Provider evaluation using joint type I/II & recovery joint outcome inference	

3 Analysis Specification

Provider-level joint outcomes of type-I (Y_3), type-I (Y_2) and recovery (Y_1) rates are modeled as trivariate regression

$$\begin{bmatrix} Y_3 \\ Y_2 \\ Y_1 \end{bmatrix} \sim MVN(\mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma})$$

allowing for inference of internal joint dependency and linear model analysis.

The current implementation is defined from the sequential conditional regressions specification

$$\begin{aligned} & N\left(Y_3|\mu_{Y_3} = (Y_2 - \mu_{Y_2})\beta_{Y_2} + (Y_1 - \mu_{Y_1})\beta_{Y_1} + \mathbf{X}\boldsymbol{\beta},\ \sigma^2_{Y_3}\right) \\ \times\ & N\left(Y_2|\mu_{Y_2} = (Y_1 - \mu_{Y_1})\beta_{Y_1} + \mathbf{X}\boldsymbol{\beta},\ \sigma^2_{Y_2}\right) \\ \times\ & N\left(Y_1|\mu_{Y_1} = \mathbf{X}\boldsymbol{\beta},\ \sigma^2_{Y_1}\right) \end{aligned}$$

but a more useful specification would be based directly on $\boldsymbol{\Sigma}$, and could be used to provide direct uncertainty characterization of joint outcome dependency structure, but this is left for a future version of the analysis.

4 Rationalization and Benefits

The specified model provides

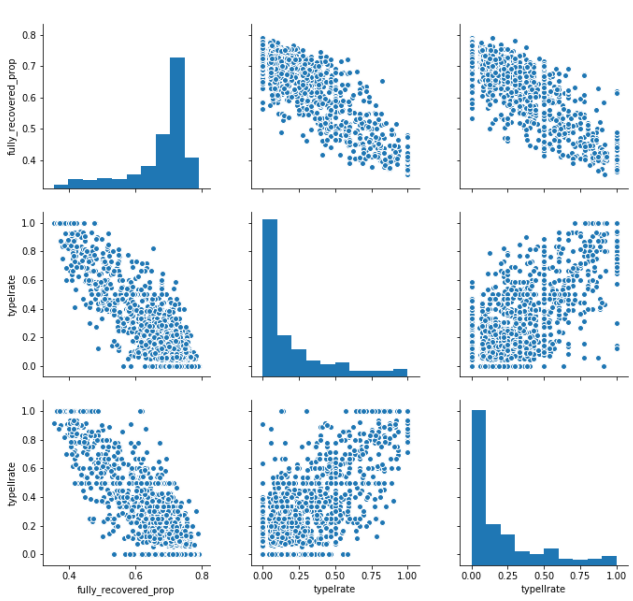
1. **any desired quality metric** based on the joint type-I/II and recovery outcome
2. **assessment of provider quality** that accounts for internal dependency between type-I/II and recovery
3. **ranking of providers** to separate good providers from bad providers
4. **type-I/II and recovery dependency estimation** for inference on quality/patient outcomes associations
5. **proper and complete uncertainty quantifications** on which to based sound conclusions

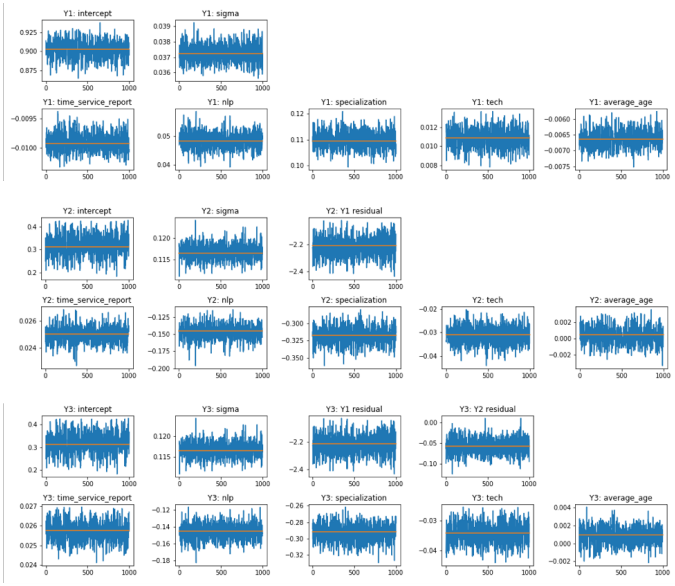
5 Uncertainty Characterization Framework

5.1 Bayesian Joint Outcome and Regression Inference

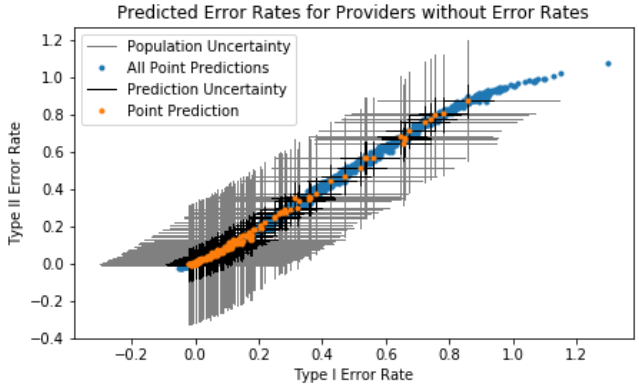
Direct modeling of joint dependency structure between *Type-I, Type-II* and *Recovery* rates

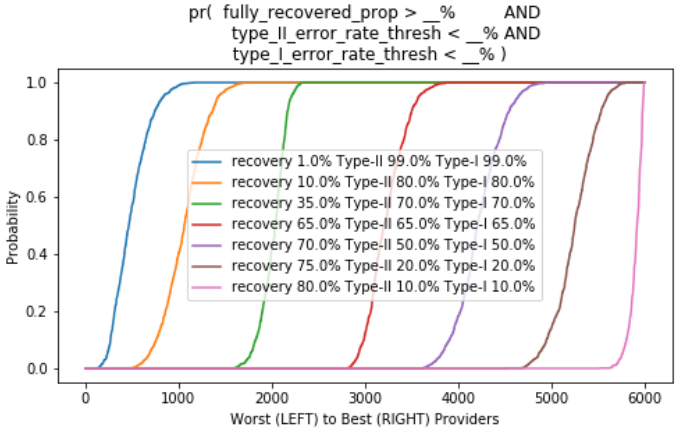
Full Bayesian uncertainty quantification analysis including trivariate linear model regression inference





5.2 Imputation/Prediction/Inference and Ranking





5.3 Full *PyMC3* Specification

A personal objective of this project was to gain familiarity with the *PyMC3*. It is a very mature package and allows sophisticated analyses such as that presented here to be specified with ease. For example, remarkably, the entire model specification used for the analysis presented here can be completed in about 20 lines of code.

```
# Priors for unknown model parameters
alpha_Y1 = pm.Normal('alpha_Y1', mu=0, sd=10)
beta_Y1  = pm.Normal('beta_Y1', mu=0, sd=10, shape=5)
sigma_Y1 = pm.HalfNormal('sigma_Y1', sd=1)
alpha_Y2 = pm.Normal('alpha_Y2', mu=0, sd=10)
beta_Y2  = pm.Normal('beta_Y2', mu=0, sd=10, shape=6)
sigma_Y2 = pm.HalfNormal('sigma_Y2', sd=1)
alpha_Y3 = pm.Normal('alpha_Y3', mu=0, sd=10)
beta_Y3  = pm.Normal('beta_Y3', mu=0, sd=10, shape=7)
sigma_Y3 = pm.HalfNormal('sigma_Y3', sd=1)
```

```
# Likelihood (sampling distribution) of observations
mu_Y1 = alpha_Y1 + beta_Y1[0]*X1 + beta_Y1[1]*X2 + \
        beta_Y1[2]*X3 + beta_Y1[3]*X4 + beta_Y1[4]*X5
y1 = pm.Normal('y1', mu=mu_Y1, sd=sigma_Y1, observed=Y1)
mu_Y2 = alpha_Y2 + beta_Y2[0]*X1 + beta_Y2[1]*X2 + beta_Y2[2]*X3 + \
        beta_Y2[3]*X4 + beta_Y2[4]*X5 + beta_Y2[5]*(y1-mu_Y1)
y2 = pm.Normal('y2', mu=mu_Y2, sd=sigma_Y2, observed=Y2)
mu_Y3 = alpha_Y3 + beta_Y3[0]*X1 + beta_Y3[1]*X2 + beta_Y3[2]*X3 + \
        beta_Y3[3]*X4 + beta_Y3[4]*X5 + beta_Y3[5]*(y1-mu_Y1) + \
        beta_Y3[6]*(y2-mu_Y2)
y3 = pm.Normal('y3', mu=mu_Y3, sd=sigma_Y3, observed=Y3)
```

6 Conclusion

Type-I, Type-II and Recovery rates are highly dependent; further feature engineering efforts are needed to refine and fully leverage inference therein. Provider rankings are statistically quantified with respect to this joint outcome.