Your task is to design, implement, and validate a model of ICU length of stay (length of stay is widely accepted as a reasonable surrogate for resource use). The model should make predictions at the time of admission and should update dynamically over the course of the stay. You will present this model to members of our Data Science and Product Management teams next week. In addition to the substance of your solution, you will be evaluated based on the organization and readability of your code.

To develop this model, we've acquired a data set of patients admitted to two different ICUs from Hemorial Merman. For each admission, the data records the patient's:

- Demographic variables (e.g. age and gender)
- Labs and vitals recorded during the ICU stay (e.g. time series of temperature readings)
- Outcomes, including length of stay in the ICU and in-hospital mortality

You can read about and download the data on this page. The specific files you will need are set-a.zip and set-b.zip, which are compressed directories containing one file per ICU visit from the two ICUs. You can read more about the variables and file format on the challenge website.

Important: For this task, we will only be using the PhysioNet challenge data, not the prescribed evaluation metrics. You should design your own metrics based on what you've learned about the capacity management use case at Hemorial Merman (although you are welcome to read through the framing of the problem used in the PhysioNet challenge).

Evaluation: Since this is a time-constrained exercise, do not worry too much about your model's performance. When evaluating your work, we'll be primarily interested in:

1. The reasoning behind the particular model and features you chose
    a. Were your choices driven by particular assumptions? Or did you discover an interesting insight in the data that motivated your choices?
2. Your analysis of the model performance
    a. Where does the model work well? Where does it fail?
    b. What might you try next with more time?
3. Readability and reproducibility
    a. Could a teammate easily read your analysis? Could they reproduce the results on their own machine?

In general, the more that you're able to make your reasoning explicit and clear, the better. Taking shortcuts or making assumptions as a first pass is okay, just be sure to make it clear what you did and why.

We recommend presenting your final results in an IPython or Rmarkdown notebook.

If you have any questions, please don't hesitate to reach out and ask.