**2 Minute Summary:**

* Discuss assessment of performance of different models for emulation. Mention ensembling approaches, talk about the best performing models and why they performed the best **(weighting with correlation to downplay models that perform badly in each and up-play the ones that work better automatically)**
  + **Use narrative OLS/PR -> GPs -> Other methods and ensemble (diversification and variance reduction)**
* Mention optimisation work, talk about the insights gleaned from working upon this, limitations in the model structure **(built a convex function)**
* Talk about sensitivity analysis -> beds, linear models that just accounted for all variance via beds considering staff only impacts processing times (not factoring in staff exhaustion), other models were better at this like random forest, sobol indices numerical estimation error. Different models performed well here than above

**Who is Responsible for which Section:**

Introduction -> All

Simulation setup (selecting the ward, automating the simulator) -> Sotiris

Exploring the data (number of steps, inputs and outputs, visualization of inputs and outputs) -> Sotiris

Emulation methods (Gaussian processes, OLS, linear fit vs more flexible approaches) ->

Neural networks -> Alejandro

GPs -> All

Ensembling emulators (weighting according to correlations) -> Ben

Sensitivity Analysis (first order vs total order, accounting all input variation in beds, simulator has near-zero first-order indices for the queue policy) -> Ryan

Optimisation (loss function, which properties it satisfies) -> Ben

Conclusion (simple models like linear regression oversimplifying relationships between inputs/outputs) -> Ryan

Future work (fit Sobol indices more accurately, explore stress-testing by largely increasing max arrivals per steps, could also look at the relationships between different wards as additional work) -> Sotiris

**Any questions to prep for:**

* How did we handle the discrete input problem?
  + Ignore integers and treat as continuous
  + Poisson based methods (i.e. Poisson regression/ poisson RF splitting)
  + Use of random forests and gradient boosted trees as robust to integers/ categorical variables
  + Tried exponentiating the input space for GP regression, but predictably found it made no difference vs GPs in normal space
* Anything that we didn’t include in the paper that we would like to discuss?
  + Mixed integer programming techniques used as current industry standards:
    - M Mistry, D Letsios, G Krennrich, RM Lee, R Misener. Mixed-integer convex nonlinear optimization with gradient-boosted trees embedded. INFORMS Journal on Computing, https://doi.org/10.1287/ijoc.2020.0993, 2020
    - ‘Branch and Bound’ methods
* Why do we think queue policy was not relevant?
  + Distributions of occupancy and queue length for all three queue policies strongly mirror each other. First-Come First-Served was chosen as it is a more widely-used approach in actual hospital settings
  + Oversimplification of simulation context? (i.e. ignoring other wards?)
  + Not as strong of a driving factor as other variables, only comes into play once there is a queue (often happened later in the simulation)
  + Maybe queue policy is more important if we take into account more wards and there is movement from one ward to another
  + Could be important in stress test
* Why didn't we use EmuKit?
  + Tried to mirror what was done in the lecture demo, but our loop ran for over 12 hours without output.
* Why did we use correlations and not MSE or accuracy?
  + (think about prediction vs prediction index ‘time-series’ like plot)
  + MSE can incorrectly be optimised by flat-lining (we saw this happen repeatedly in models that were too complex, when errors were near the main trend line)
  + Accuracy isn’t suitable for regression problems
  + Correlations better represented the similarity between results from different models, not as suitable if relationships are
* How did we go about selecting the models to assess?
  + Chose models of a variety of different types, from simplest (OLS) to much more flexible (random forest, neural networks). Tried to choose models that work well on both continuous and discrete inputs
* If we had more time, what would we have done?
  + Future work (fit Sobol indices more accurately, explore stress-testing by largely increasing max arrivals per steps, could also look at the relationships between different wards as additional work)
  + See how we can model the interaction between different wards (graph problem)
  + See if queue policy actually makes an impact when we use more wards
  + Instead of 80/20 we could use a 90/10 train-test split and apply data augmentation to that 10%. We didn;’t try it as we have a medical dataset and it could dilute the data.
* Why is emulation useful for this task?
  + Understanding in greater depth how the inputs affect the outputs and assessing how these different variables affect the outputs (better interpretability)
  + In resources with low levels of computational power, these emulation results could be useful
* How does the simulator actually work?
  + Graph theory, queue theory, discrete event simulation, it’s quite complex so we wanted to emulate to better understand relationships between inputs and outputs
* Discuss the literature review
  + Wanted to see if others in the field had done much with this dataset that could inform our approach. Wanted to assess whether it would be useful to create a tool to run this automatically and save the results
* How we went about choosing our outputs of interest:
  + First outputs: total cost, mean hospital occupancy and queue length (abs)
  + Second phase: total cost, mean hospital occupancy and mean queue length
  + Third phase: total cost, mean emergency occupancy and queue length (abs) + 48h
  + Mention that we were using queue policy as an input feature, but it was irrelevant and we end up ignoring ⅔ of the points.
  + These are two key indicators of hospital efficiency
* Why did we focus on the emergency room
  + Deals with the greatest flows in patients coming in and out, most interesting to emulate
* Why do we think it is the reason why beds are much more important than staff/resources?
  + Maybe because increasing the number of beds certainly results in a lower occupancy (%)?
  + Model might be inherently too simple, underplayed as a factor in the simulator, idea that staff don’t get tired (efficiency remains the same)
    - The staff/resources is a proxy for the number of patients processed per hour (stochastic poisson variable) and the rate is the same even when few staff are ‘overworked’ (would expect to decrease)
* Why random forest?
  + Doesn’t assume a function shape, quite hard to overfit, robust to hyperparameters. Easy to tune, hard to overfit. It is an ensemble method in a way, so all the benefits of ensemble methods apply to random forest
* Why GP?
  + Quite a flexible model, can model both complex and simple functions, lots of additions that you can add in
  + Used RBF as we didn’t know what the shape looked like, tuned the amplitude, length scale. Tried to tune the covariance function but it didn’t make a difference.
* Model decision narrative:
  + Started with OLS and GPs
  + Moved from OLS to Poisson
  + Moved from GPs to Random Forests
  + We wanted more diverse models
  + Always using highly correlated models won’t give diversity
* Why poisson splitting?
  + Assumes discrete values so more apt for our input space, interpretable model
* Why XGBoosting?
  + Similar to random forests, but makes the trees sequentially and uses feedback from the error of the previous tree to build the next one. This is an improvement but slower.
  + Also quite easy to control overfitting via parameter tuning (e.g. depth and width).
* Impact of not being able to put down beds to 1 for all of the other wards? How to quantify this error?
  + Our inputs were all at least 10x bigger so this should be a negligible difference, if we had wanted to, we could also increase the size of emergency room even more to account for this
* Why ensemble?
  + Decrease variance in the outputs by averaging predictions, likely would have lower generalizability error.
  + Slight decrease in the performance, but more generalisable.
  + Can mention ‘Condorcet’s Theorem’ to justify ensemble methods: <https://en.wikipedia.org/wiki/Condorcet%27s_jury_theorem#:~:text=Condorcet's%20jury%20theorem%20is%20a,the%20Probability%20of%20Majority%20Decisions.>
* What is a weakness of the paper?
  + Simplification of inputs to discrete
  + Not accounting for interplay between different wards
  + Fixing maximum number of patients, so there’s a lot of randomness
* What’s a strength of the paper?
  + Ensemble method works
  + Rigorous treatment
  + We delved into creation of emulation methods, optimization, and sobol indices for exploring the data
* How could we have reduced the randomness in the data?
  + We got the number of arrivals of patients
  + Could generate the same set of patients
  + Could increase the scale of things, number of datapoints, number of patients, number of beds, etc. Sparsity issues