Test challenge logic overview and documentation

My first approach was to fetch the gfts file, but the sheer number of data and my relative unfamiliarity with Python (wish i could have made this in C#) made it seem like a decision that is not valuable, so i simply used bus tables for the times for the formula. I added diferent percentages that when landing on it, add extra lateness or earliness times (earliness is rare and lateness common in accordance with tallinn transport traditions).

The Schedule is parsed into a list of dicts, a simple enough solution that works. (a common theme in my work, first thing is to have a working prototype and move on from that)

Simulation runs were easy to make, I admit python is very comfortable for such a task. Plotting logic I just ripped straight from stackoverflow, same with annotations.

It was an interesting assignment, but frankly most of the difficulty lies in finding gfts data, and having your computer comply with the large amounts of data. Creating odds and theorycrafting how to have the chances work was very fun though.