Script execution and description

Gurobi packages

* Academic License: <https://www.gurobi.com/features/academic-named-user-license/>
* How to install on python (I follow the conda instructions): <https://support.gurobi.com/hc/en-us/articles/360044290292-How-do-I-install-Gurobi-for-Python->
* Include these imports to use gurobi
  + import gurobipy as gp
  + from gurobipy import GRB

The latest workflow of the code:

1. Start with execute\_v2.py, sets the parameters of the scenario, creates the full set of vehicle requests, incrementally steps through the scenario and determines which vehicles should be considered in the current window and the resulting parking schedule
   1. Calls PAP.py, original PAP optimization framework, objective is to minimize the total minutes of unscheduled service duration
   2. Calls PAP\_min\_deviation.py, takes the solution from PAP.py and attempts to move vehicles from their scheduled time to as close as possible to their originally requested time, counter acting any scheduling at the far end of the requested arrival time +/- phi
   3. Can call gen\_Pitt\_arrivals.py, which pulls real-world observations from the Pittsburgh dataset and populates them into the full set of vehicle requests for the scenario
   4. Call seq\_arrival\_new.py, allows for a FCFS assessment of the full day vehicle request schedule, for comparison with the sliding time window model

Other function included on GitHub

* Execute\_v2\_loop.py, attempt to loop through a set of zeta and tau values and then create graphics based on the scenarios, was working, but likely is not up to date
* SimpleAnalysis.py and CleanDataChicago.py, Connor generated functions to study Uber trip durations from Chicago
* AP.py, function from the first paper, has not been adapted yet for use in the sliding time window paper, may not be necessary either
  + GenBids.py is associated with AP.py

Old / not sure hahaha

* Visualize\_sliding.py
* Execute.py