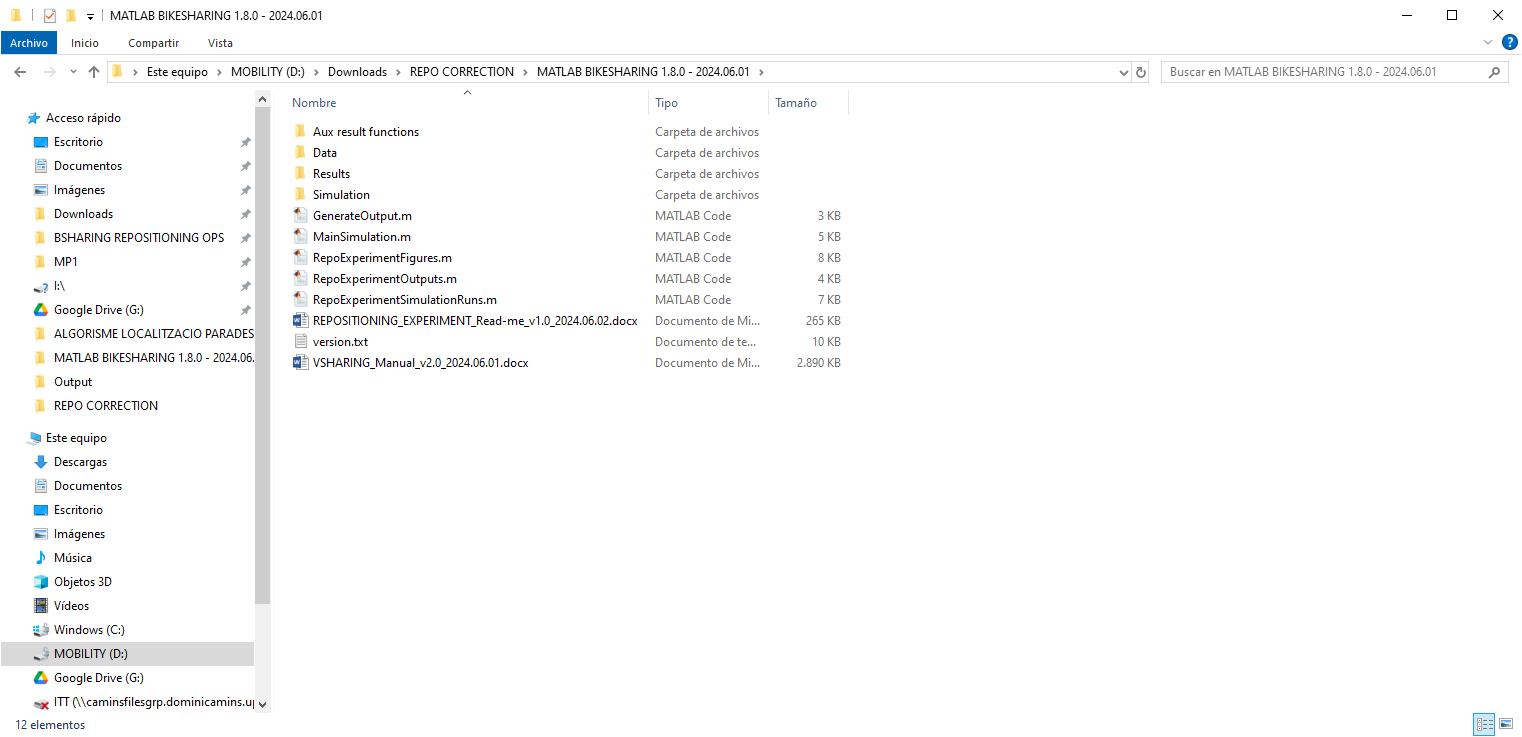
**Bike-sharing repositioning experiment – Read-me file (v.1.0)**

The purpose of this document is to provide a guide for replicating experiments of repositioning strategies comparison, as done in the paper [(Jiménez-Meroño & Soriguera, 2024)]. This document references the general-purpose user manual of the Vsharing Simulator.

In order to run these bike-sharing repositioning comparison experiments, in addition to the core files of the simulator (see section 1.2. in the Vsharing Simulator User Manual 2.0.0.), the following support files must be included in the Vsharing Simulator folder:

* **“Aux result functions” folder:** This folder contains some custom functions coded to ease the result postprocess.
* **“RepoExperimentSimulationRuns.m” file:** This file is a MATLAB script that runs a set of experiments testing all the repositioning strategies under the same simulation conditions (service area, layout, potential demand trips…).
* **“RepoExperimentOutputs.m” file:** This file is a MATLAB script that, given a raw simulation data result file, returns also two Excel files with raw results. These results are the same as presented in [(Jiménez-Meroño & Soriguera, 2024)].
* **“RepoExperimentFigures.m” file:** This file is a MATLAB script that, given a processed and aggregated station-by-station result Excel file, returns graphic representations. Figures obtained are the same as presented in [(Jiménez-Meroño & Soriguera, 2024)].
* **“REPOSITIONING\_EXPERIMENT\_Read-me” file:** This is the present manual with the guidelines to run the experiment and process its results.



***Figure 1.*** *Folder and file structure of the simulator.*

**Part 1 – Simulation**

1. **Preparation of the inputs**

Simulation time is set to 6 days, divided into a cycle of 3 days for warm-up and another cycle of 3 days for recording results. This is a recommendation from our experience. Bigger cycles could result in excessively large register files that cannot be stored to process the results later, and also implies a longer computing time due to warm-up (the Vsharing Simulator sets the minimum warm-up to at least one cycle, which would increase if it is defined as a longer period). Smaller cycles would not be efficient enough to aggregate results later, and would need of more experiment repetitions. Anyway, any other period could be considered if properly defined. (Check section 3.2. of the Vsharing User Manual 2.0.0 for more info about the configuration of simulation time.)

In order to set-up this 3+3-days simulation time the following tasks must be fulfilled:

* If the O/D demand feed come from known matrices, it must be ensured that the matrices cover the simulation cycle time (in this case, 3 days).
* The input Excel file “input.xlsx” must be checked to introduce properly the simulation parameters:
  + Set: “TotalTime” = 4320 [min]. In order to set the 3-day duration cycles.
  + Set: “WarmUpCycles = 1. In order to set warm-up to a single cycle (defined as 3 days).
  + Set the rest of parameters as considered for the experiment. Check section 2.1 of the Vsharing User Manual 2.0.0 for more info about the simulation inputs file.

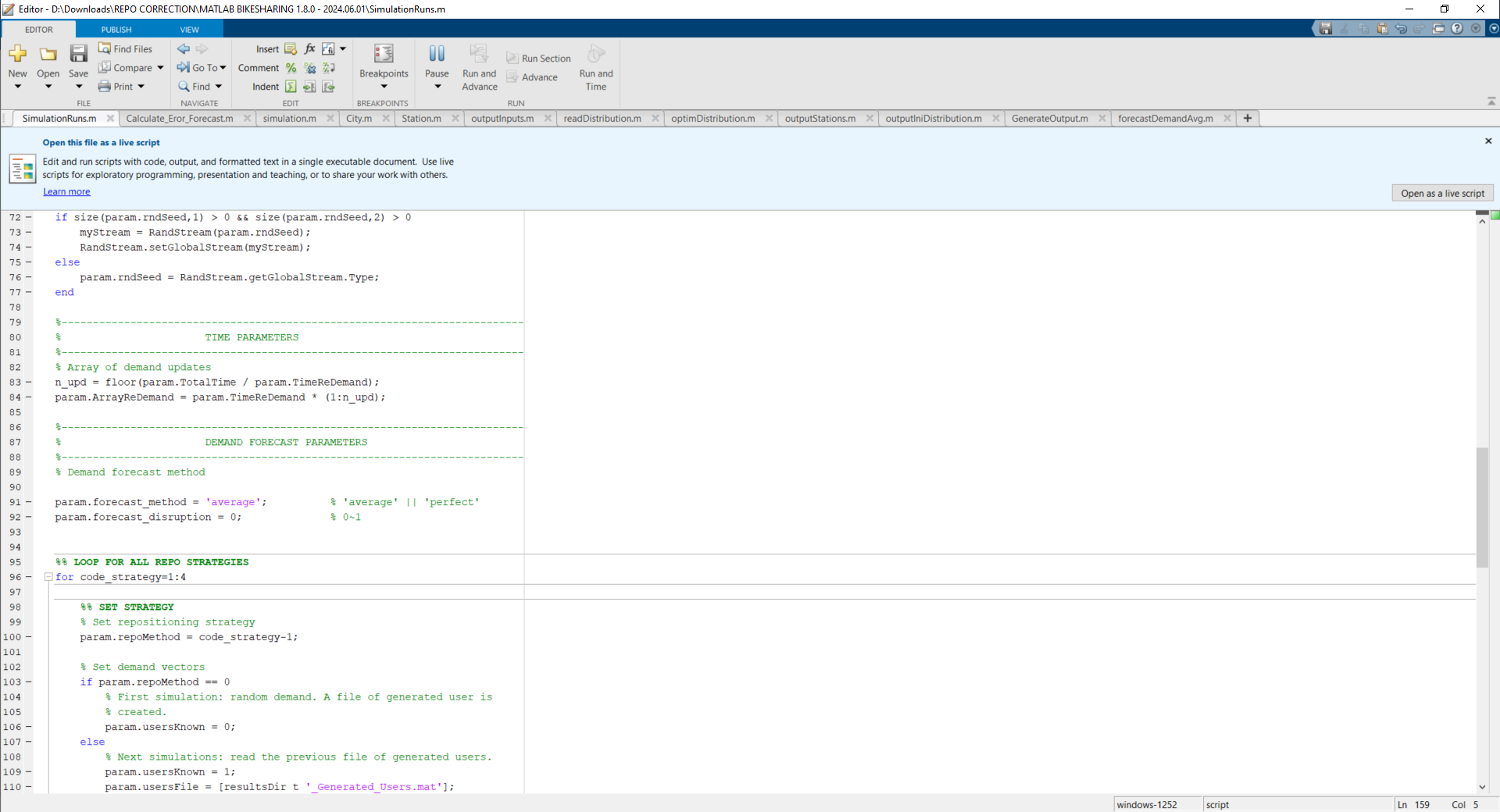
In order to replicate the same experiments, all inputs in “inputs.xlsx” file must remain constant for all the experiments except for the random seed (“rndSeed”), which must be changed on each run of the simulation script to generate different trip arrays.

1. **Run the simulation script *“RepoExperimentSimulationRuns.m”***

Script “RepoExperimentSimulationRuns.m” has been coded to ease the experiment repetition. The script will run four simulation experiments in one:

* First, it will run a base simulation without repositioning. Potential trips are generated randomly, and a register of them is stored in an array (see section 6.2.2.4. of the Vsharing User Manual v.2.0.0). All data from the simulation is stored as normally (see 6.2.1. and 6.2.2. sections in the Vsharing User Manual v.2.0.0).
* After that, the script will run three other simulations. On each one the repositioning strategy is changed. In all three cases the potential trips are not random now, but read from the trip register from the base simulation. That way, the script results in a set of results of four simulations with the same conditions and potential trips, but changing the repositioning strategy.

Note that one important parameter in the experiments is the demand forecast error. In order to have different errors on the forecast, the script allows to choose between two forecast methods and the introduction of disruptions.



***Figure 2.*** *Input of demand forecast options in “RepoExperimentSimulationRuns.m”.*

After each run, the simulation returns the average forecast error on demand requests and returns. Note that, since the error depends on the actual performance of the simulator (users change their behavior and the stations to go on origin and destination depending on the availability), it cannot be estimated beforehand. So, a process of tuning is necessary on the demand forecast parameters in order to obtain forecast errors similar for all strategies and in the desired range.

1. **Repetition**

To get larger set of results, the process must be repeated doing the following:

* Change the random seed (“rndSeed”) in the input file to generate a different set of potential trips.
* Run the simulation script again several times to tune the forecast errors.

The process is repeated until achieving an adequate number of total days for each combination of repositioning strategy and demand forecast error.

**Part 2 – Postprocessing of results**

In this second part, data from the simulation runs is processed in order to obtain results. The lack of a high-capacity DB system which allows to organize, classify, and aggregate the simulation experiments is limiting, and it forces this results generation process to be way more manual than the Part 1. Also for that reason, this explanation and the process automation is limited to generate the results shown in the paper. However, any user could obtain other types of results trough the general postprocessing script “GenerateOutput.m” as explained in section 6 of the Vsharing User Manual 2.0.0.

1. **Classification of results**

Simulation results files and folders must be classified carefully and manually. To ease these tasks is it advisable to check the input summary file, and rename the simulation result file and folder according to the repositioning strategy, the demand forecast error, and the random seed. It is also advisable to group several results files and folder into a main folder that indicates the repositioning strategy and the demand forecast error.

1. **Check the output generation variables “output.xlsx” excel file**

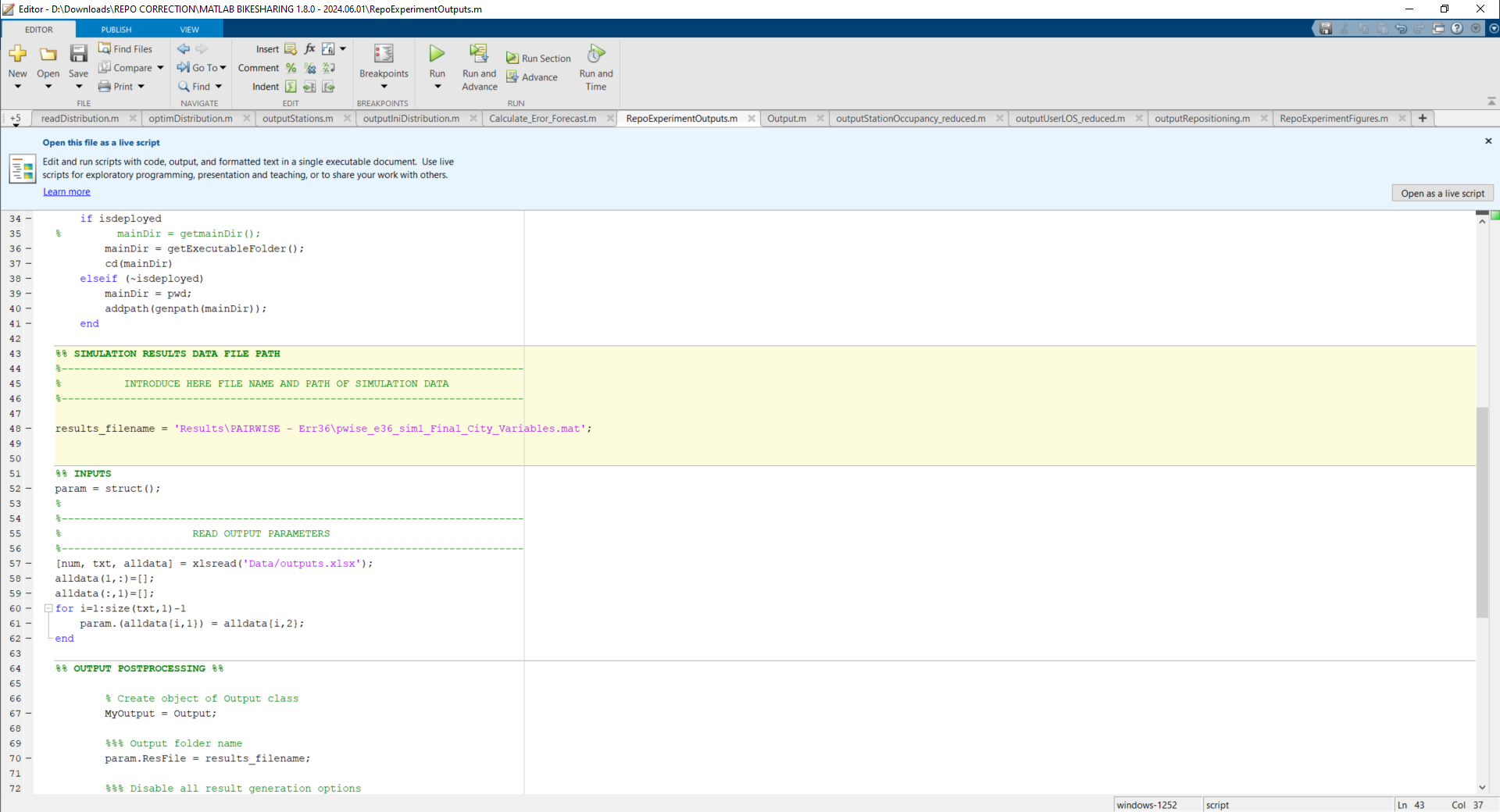
As a previous step to running the scripts, the input parameters for the postprocessing must be checked. Note that, since the scripts generate specific results, many options will be limited.

* SIMULATION variables -> Is it possible to turn on/off the additional information shown in the MATLAB command window during the postprocess. The rest of fields are not used in the scripts. The simulation data file name (“ResFile”) won’t be introduced here but directly in the script.
* DETAL LEVEL and CATEGORY variables -> These fields are not used and changes wouldn’t have any effect. The script “RepoExperimentOutput.m” will select automatically which categories and detail level returns.
* USER COST and AGENCY COST variables -> These fields must be filled coherently to the simulation inputs in “inputs.xlsx”. They will remain

For more information about the “output.xlsx” file, check section 6.1. in the Vsharing User Manual 2.0.0.

1. **Run the postprocessing script** **“RepoExperimentOutput.m”**

Script “RepoExperimentOutput.m” has been coded to ease the task of generating customized results. The script must be run once for each simulation result data file. In order to ease the process, the simulation data filename must be introduced directly in the corresponding line of the script.



***Figure 2.*** *Input of simulation data result file in “*RepoExperimentOutput*.m”.*

The script will automatically create an output folder with the processed performance results. The process is analogous to the result postprocess and the output folder and files explained in section 6.2.3. of the Vsharing User Manual 2.0.0. Therefore, the structure will be similar, and folder’s name will be also automatically generated from the simulation data filename with the suffix “\_output”.

Two results are generated and stored inside the output folder:

* System-level results -> These are overall KPIs of the whole system. They are stored in the summary excel file “Table\_summary.xlsx”.
* Station-by-station results -> These are KPIs that consider a single station. In this particular case, the calculated KPIs are the % of time full and % of time empty for each station. They are stored in an excel file called “TableFullEmpty.xlsx”.

1. **Aggregation**

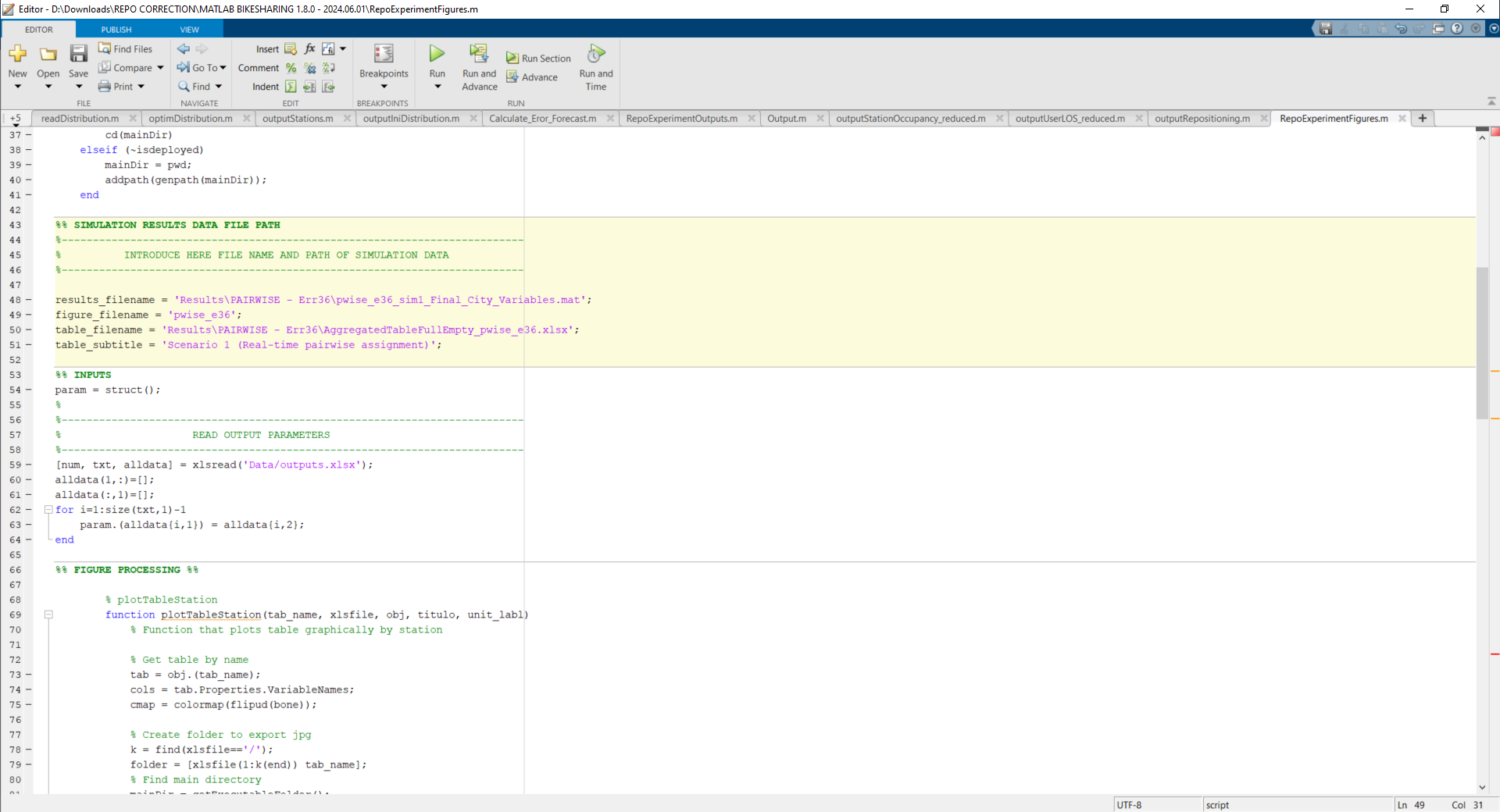
Results need of manual aggregation to consider the whole set of days with same repositioning strategy and demand forecasting errors. Aggregation procedure must be different depending on the type of results:

* For system-level KPIs, the values from the corresponding result summary files “Table\_summary.xlsx” are aggregated through a weighted average.
* In the case of station-by-station KPIs, a new excel results file must be created with the same structure to the station-by-station results “TableFullEmpty.xlsx”. A way to do it is to copy one of the station-by-station excel files and substitute the KPIs values with their corresponding aggregated values (i.e. weighted average).

1. **Run figure creation script “RepoExperimentFigures.m”.**

This script creates the two figures from the aggregated station-by-station results. User must input:

* Simulation data filename (“results\_filename”) -> These data are used to obtain the stations’ IDs and locations. Note that any simulation data file from the experiment would suffice, since in all cases the stations remain the same.
* Figure name (“figure\_filename”) -> Any character string. It is advisable to introduce a name that identifies the repositioning strategy and demand forecast error. The figure outputs will include the suffix “\_full” or “\_empty” to their names.
* Station-by-station result file (“table\_filename”) -> Path and name of the Excel file created in the previous step to aggregate station-by-station KPIs.
* Table subtitle (“table\_subtitle”) -> Any character string to print as subtitle of the figure. It is advisable to introduce here something that identifies the repositioning strategy or whatever that defines the experiment.



***Figure 3.*** *Inputs of the figure creation script “*RepoExperimentFigures*.m”.*

The script will automatically create the map figures and save them in the “Results” folder.

**REFERENCES**

Jiménez-Meroño, E. & Soriguera, F. (2024) “Optimization of bike-sharing repositioning operations: A reactive real-time approach.”

Jiménez-Meroño, E. & Soriguera, F. (2020) Vsharing Simulator User Manual (ver. 2.0.0).