EPA133a. Advanced Simulation
Assignment 2. Building Components for Data-Driven Simulation

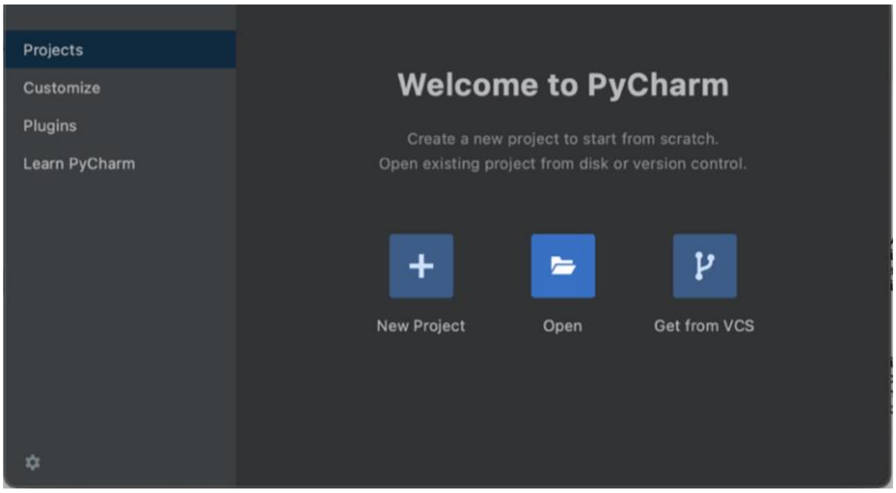
Introduction

This assignment will look at the model generation process and how components can help in the generation. We will use Mesa 2.1.4 in this and the following exercises. Mesa is a Python3 agent-based simulation environment. Object-orientation makes the creation of user-defined objects and user-defined libraries relatively easy. We will learn to create and place user-defined models from data automatically. We will develop a small demo model of goods transport over a number of the cleaned roads in Bangladesh. Bridges and potentially ferries play an important role in Bangladesh transport, as they can cause delay or rerouting, especially after flooding, mudslides, earthquakes, or cyclones.

Lab Assignment 2 in PyCharm

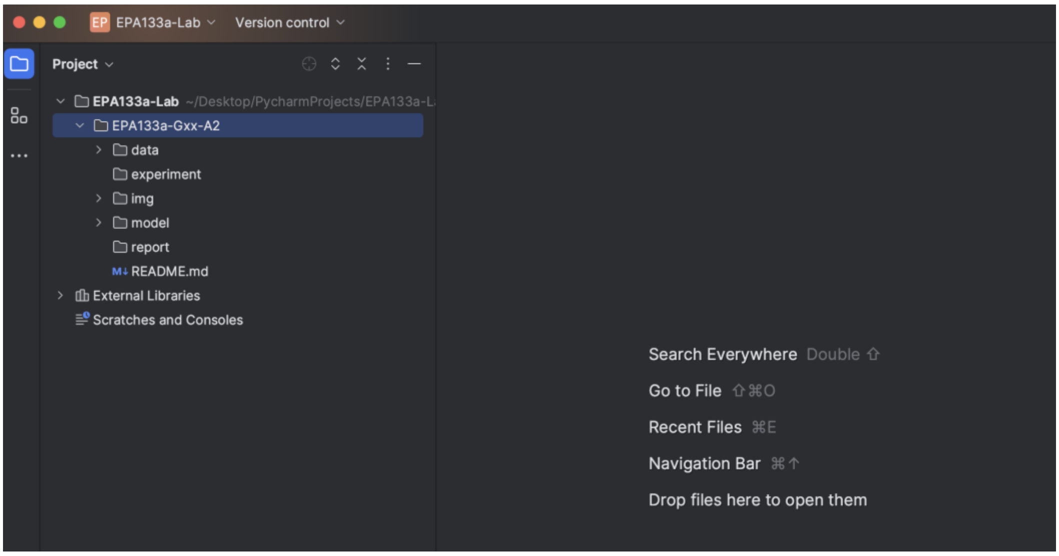
This instruction is written based on PyCharm 2023.3 running on macOS. The interface may differ from the description if you use another version of PyCharm or another operating system.

1. Download the “EPA133a-Lab.zip” file from BrightSpace. Unzip the file into a chosen directory on your PC.
2. Launch PyCharm. Open “EPA133a-Lab” as an existing project.

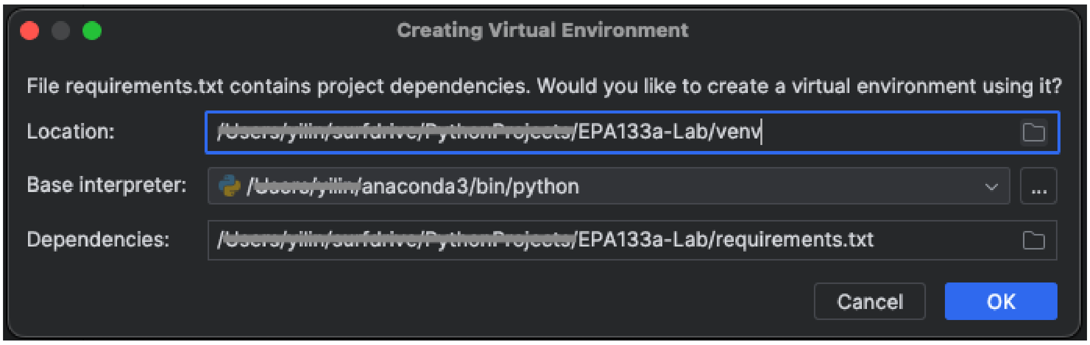


Welcome to PyCharm

Create a new project to start from scratch.
Open existing project from disk or version control.



3. Ensure you updated your Python 3 interpreter; the assignment files are tested with 3.11.
4. If you use a newer PyCharm version, you may get a prompt window (because of the requirements.txt file in the EPA133a-Lab folder). Choose the right setting and press OK.



Creating Virtual Environment

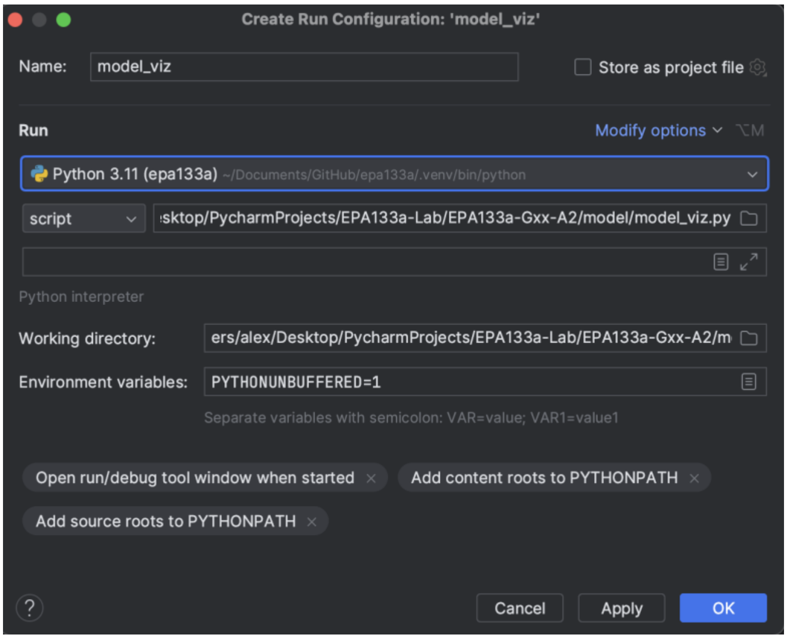
File requirements.txt contains project dependencies. Would you like to create a virtual environment using it?

Location: [redacted]/EPA133a-Lab/venv

Base interpreter: /[redacted]/anaconda3/bin/python

Dependencies: [redacted]/EPA133a-Lab/requirements.txt

5. If you did not get the prompt box shown in Step 4, create a virtual environment with the following steps:
   - Go to Settings -> Project: epa133a -> Python Interpreter
   - Click "Add Interpreter"
   - Select "Add Local Interpreter"
   - Select Virtualenv Environment
   - Select New environment
   - Select Base interpreter as, e.g., Python 3.11
   - Click OK and also close the settings with OK
   - Afterwards, you should see "Python 3.11" (epa133a) in the bottom-right corner of the PyCharm window. To install the requirements, open a terminal/command line window in PyCharm and type:
     - pip install -r requirements.txt
     - Note that the program works with **Mesa 2.1.4**, not the latest Mesa version.
6. Right click “EPA133a-Gxx-A2”, choose Refactor/Rename… the folder with your group number, e.g., “EPA133a-G01-A2”. Click “Refactor”
7. In the PyCharm Project window, click open the “EPA133a-Lab > EPA133a-Gxx-A2 > model” folder, right-click the “model_viz.py” file, and choose “Modify Run Configuration…”.
8. Rename “model_viz” to “A2 model_viz”; click “Apply,” then “OK.”



Create Run Configuration: 'model_viz'

Name: model_viz

□ Store as project file

Run

Python 3.11 (epa133a) ~/Documents/GitHub/epa133a/.venv/bin/python

script sktop/PycharmProjects/EPA133a-Lab/EPA133a-Gxx-A2/model/model_viz.py

Python interpreter

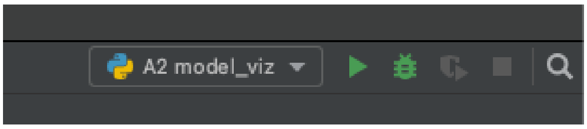
Working directory: ers/alex/Desktop/PycharmProjects/EPA133a-Lab/EPA133a-Gxx-A2/m

Environment variables: PYTHONUNBUFFERED=1

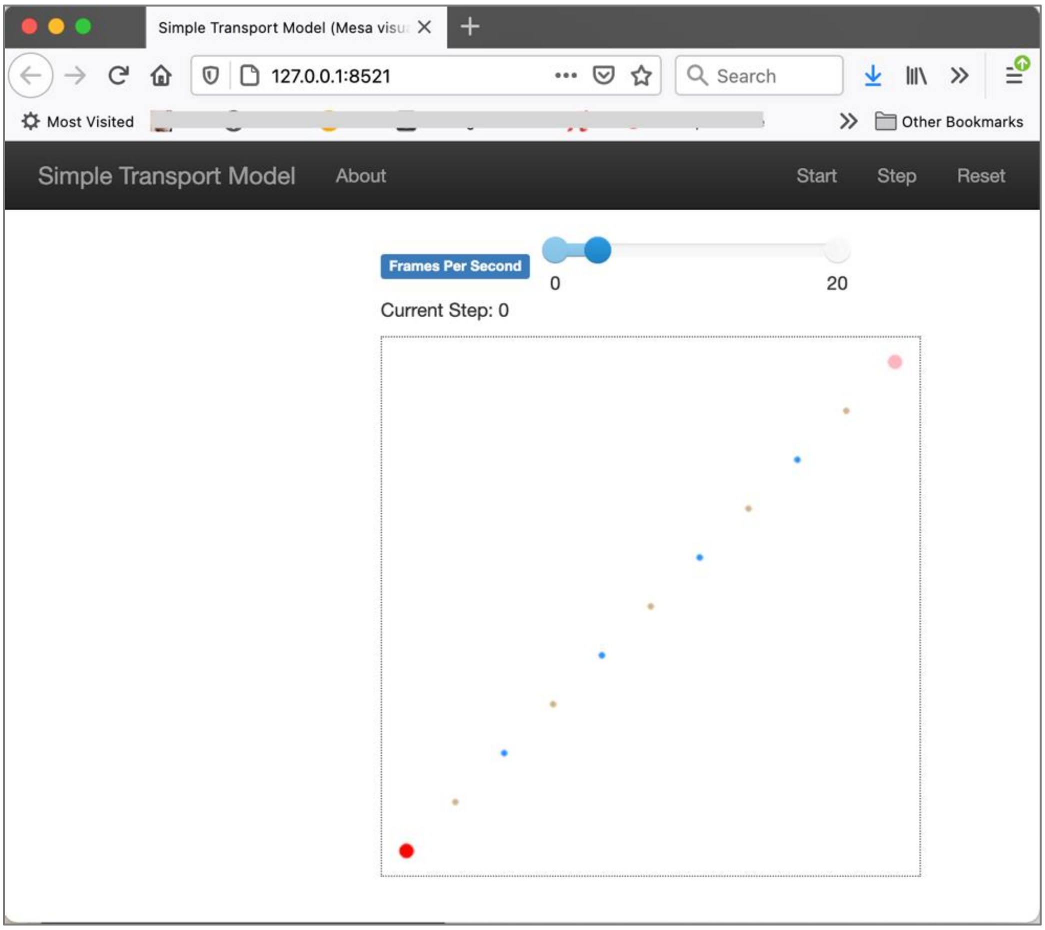
Separate variables with semicolon: VAR=value; VAR1=value1

Open run/debug tool window when started × Add content roots to PYTHONPATH × Add source roots to PYTHONPATH ×

9. Click the green Run arrow icon at the top right of the window.



10. A tab will open in your default browser. It looks like this:



Simple Transport Model (Mesa visu

127.0.0.1:8521

Current Step: 0

11. The demo simulation model visualisation is now launched. You can click the Start, Step, and Reset buttons (top right of the browser window) to see what the visualisation does. You can also use the slide bar to adjust the speed of the simulation. (You can run “model_run.py” instead of “model_viz.py”. It launches simulation without visualisation.)

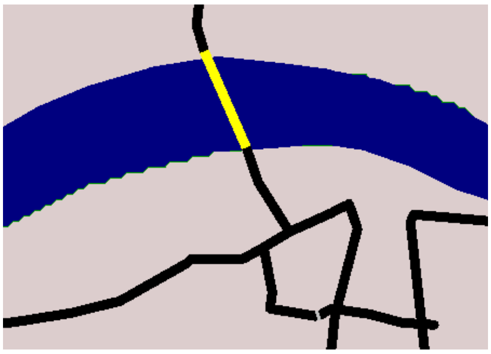
In this Mesa simulation demo (demo-1), vehicles (trucks) are generated every 5 ticks at the button left corner (red dot) by a component named Source. They drive through the links and bridges (links visualised in rose dots, bridges in blue) and end their trips at the top right corner (pink dot) once they reach a Sink component. A link is an edge that connects two components, e.g., a source to a bridge, a bridge to a bridge. The trucks move at a certain speed and wait at bridges with a specific delay time. You can find all the component definitions in the “components.py” file.

Try out and Study the Model

Your first task is to study the model and the components. You shall first gain a good understanding of how the demo model works to proceed! Begin with the “README.md” file (Always do so before you look into any code!) in the “model” folder. (README files can exist in many different folders.) Read it carefully in relation to the model files. To see how model generation works, start at “class BangladeshModel” in “model.py”. Check how the model components are created. Do you understand how the “generate_model” routine reads the data file and creates the component instances? Change the input file “demo-1.csv” to “demo-2.csv” or “demo-3.csv” to see how the generated model changes according to the input data. Note that in demo-1 and demo-2, one source and one sink are generated. The vehicles are generated at the source, drive from source to sink, and are removed at sink. In demo-3, both ends of the route act as source AND sink. The vehicles can be generated and removed at both ends, and they can “drive” in both directions. Look carefully at the vehicle and infrastructure components and see how and why they work as they do. Use the debugger to inspect the model. Ask the TAs when you have questions.

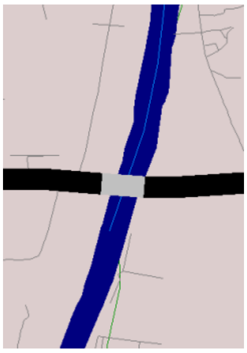
Road and bridge Dataset on BrightSpace

Note that the "WEBSIM_Lab1_cleanedDataset" contains a 95% cleaned dataset, and log files explaining what was cleaned and left out in relation to the raw data. The file '_roads3.csv' used as the input contains an extra column to indicate some information gathered about the start and end of bridges, ferries, and the absence of a road as gathered from the RMMS files. As this had to be obtained from human-entered text, this data is still incomplete and not always right. Bridges, ferries, and gaps are also shown when running the latest Java program wbsim.jar:



Ferries are in yellow

Discontinuation of the road in white



Bridges in gray

You work with this dataset for this assignment.

Assignment and Deadline

Automatically generate a Mesa model to study the effects of bridge maintenance or unavailability on traffic throughput for a major road in Bangladesh. Let's take the economically very important **N1 from Chittagong to Dhaka** as an example for the assignment. Think carefully about how to create the data needed for model generation. **To be able to do this, you need to understand the demo model fully.** Use the demo csv files as examples. Also, think about a Mesa component for bridges that can cause delays according to the quality of the bridges, e.g., when repairs on that bridge would be necessary. Quality D bridges would break down much more often than Quality A bridges. In some way, this has to be incorporated into your component(s) for the bridge. Note that when there is an "L" and an "R" bridge at the same location, you must decide (and document!) how to treat this in the model you generated.

Send trucks (entities) every 5 minutes from Chittagong to Dhaka to measure delays and travel time. (Think about how to collect the data.) Driving the trucks in one direction is sufficient for this assignment. Under normal circumstances (and without traffic jams), the driving time for a truck from Chittagong to Dhaka is around 6 hours for the 287 km long highway, resulting in an average speed of 48 km/h. Use this as the speed on the entire road.

First, make a "business as usual" model (Scenario 0), where no bridges break down. Automatically create a Source for entities (trucks) at the start of the N1 near Chittagong and automatically generate a Sink near Dhaka. Automatically generate the roads and the bridges as well. Create trucks every 5 minutes and run the model for 5 x 24 hours runtime. When you run the experiments, visualization is not recommended. Note that every tick in the simulation represents one minute in the demo model. Gather the driving time as the most important response. Export the experimental output to a “scenario0.csv” file.

Make sure your model has controls for the probability of Cat A, Cat B, Cat C, and Cat D bridges to break down, e.g., you can expand the model to take those probabilities as input. The delay time for a truck at a bridge that is in repair or partly broken down varies based on the length of a bridge; see the table below. Running models for congestion is not needed; use a pure delay time at a broken bridge that could be different for each truck. Gather the average driving time as the most important response.

Bridge length	**Delay time for a truck**
Over 200 m	Triangular(1, 2, 4) hours
Between 50 and 200 m	Uniform(45, 90) minutes
Between 10 and 50 m	Uniform(15, 60) minutes
Under 10 m	Uniform(10, 20) minutes

Then run experiments for the following scenarios (10 replications using different seeds; each replication has different physical bridges breaking down): (10 replications are usually not sufficient for real-world analysis. But for this exercise, we use 10 replications to reduce runtime.)

Scenario	**Cat A %**	**Cat B %**	**Cat C %**	**Cat D %**
1	0	0	0	5
2	0	0	0	10
3	0	0	5	10
4	0	0	10	20
5	0	5	10	20

6	0	10	20	40
7	5	10	20	40
8	10	20	40	80

Name the experimental output files as “scenario1.csv”, “scenario2.csv”, etc. Place these files in the “experiment” folder of the submission folder “EPA133a-Gxx-A2”. The xx shall be replaced with your group number. Study the effects of each scenario on the driving time for the trucks and discuss the results in a short report. Are 10 replications for each scenario producing reliable enough results for analysis? Why or why not? Discuss this and the implications in the report. Place the report in the “report” folder.

Bonus Exercises. It is not necessary to do the bonus exercises. If you choose to, discuss the outcome in the report.

1. Try to find out which 5 bridges on the N1 the Bangladesh Government should invest in to decrease lost travel time best, when we weigh scenarios 1 though 7 equally.
2. What model and model component design (and/or improvement) did you do with respect to, e.g., modularity and cohesion? Specify and discuss those with the rationale.

Hand in the “EPA133a-Gxx-A2” Zip file with the model, data, experiments, and a short report of how you prepared the data, designed your model, and what the results of your experiments were. The recommended report length is 3-4 pages (this means 1500-2000 words), excluding visuals such as images and tables. Prepare the Zip file according to the Submission Guidelines. Upload the Zip file to the file area on Brightspace.

Some Mesa resources online

- https://mesa.readthedocs.io/en/stable/mesa.html
- https://readthedocs.org/projects/mesa/downloads/pdf/stable/
- https://github.com/projectmesa/mesa
- The Mesa user mailing list: https://groups.google.com/g/projectmesa

Time to spend and Support

There are 8 lab hours dedicated to completing the lab assignment. In addition, you are each expected to spend another **8 hours maximally** per person on carrying out the exercise. Don't overspend your hours, and see how far you can get with component building and model generation in 16 hours total. You could already get a passing mark if you prepare the data for model generation, modify the bridge components, generate the model, and can run Scenario 0 and 1. Divide the work well within your group, and make sure you use the available hours of all team members combined well, through good collaboration and communication.

The deadline for handing in the Zip file of Assignment 2 is Friday in week 4 at 18:00.

Only upload using the Assignment function, don’t use the File Locker or email to hand in – we will base the grading on what you hand in as Assignment 2 on Brightspace.