Laboratory 6

Introduction to Transport Modeling: Gravity Model & Demand Regression

Module: Transport Data Collection and Management

Duration: 180 Minutes

Contents

Objective

In this lab you will:

- use a simple **gravity model** to perform trip distribution for small OD systems,
- calibrate an **exponential friction function** by changing the parameter β,
- experiment with interactive Python tools for gravity modeling and friction factors,
- manually specify a custom friction matrix F_{ij} and observe its effect on trip patterns,
- use a regression model to **predict transport demand** from distance, fare and route data,
- apply sensitivity analysis and prepare a small homework study with your own OD system and β values.

The focus is on following the practical steps clearly and interpreting outputs from Excel and the Colab notebooks.

1 Files & Tools

1.1 Files Used in This Lab

- **Excel:**
  - `Trip Distribution Lab Example.xlsx`
- **Python / Colab notebooks:**
  - `Gravity_Model_Student_Input.ipynb`
  - `Interactive_Gravity_Model_Lab.ipynb`
  - `Gravity_Model_Student_Input(Exponential_Deterrence_Function).ipynb`
  - `Gravity_Model_Manual_Friction.ipynb`
  - `Transport_Demand_Regression_Lab.ipynb`

1.2 Software Platforms

- Microsoft Excel (or Google Sheets).
- Google Colab (preferred) or local Jupyter Notebook with Python 3.10+.
- Python libraries: `numpy`, `pandas`, `matplotlib`, `seaborn`, `scikit-learn` (typically pre-installed on Colab).

Reflection & Insight

The theoretical background (gravity model, friction functions, zoning, input tables) is summarised in the Lab 6 slides. Use those for reference if you get lost in the formulas.

2 Activity 1 – Gravity Model in Excel

In this activity, you will look at a simple 3-zone example in Excel and see how the gravity model uses Productions P_i, Attractions A_j, a Cost matrix C_{ij} and a friction function F_{ij} to produce a trip matrix T_{ij}.

Task 1: Inspect Productions Attractions and Cost Matrix

Goal: Understand the basic inputs for the gravity model.
Steps:
1. Open `Trip Distribution Lab Example.xlsx`.
2. Identify the sheet that contains:
   - the table of **Productions** P_i and **Attractions** A_j (zone totals),
   - the **skim/cost matrix** C_{ij} (travel time or generalised cost between zones).
3. Confirm:
   - how many zones the example uses,
   - the total production $\sum_i P_i$ and total attraction $\sum_j A_j$.

Checkpoint: You should be able to say, for example, "We have 3 zones, total production = ..., total attraction = ...".

Task 2: Friction Factor and Trip Matrix

Goal: See how the exponential friction function and T_{ij} matrix are implemented.

$$F_{ij} = e^{-\beta C_{ij}}$$

Steps:
1. Find the sheet (or range) where the friction factor F_{ij} is calculated.
2. Inspect the formula used in one cell and verify it matches the exponential form (e.g. something like `=EXP(-beta * Cij)`).
3. Look at the resulting **trip matrix** T_{ij}:
   - Check row sums (should be close to P_i),
   - Check column sums (should be close to A_j).
4. Find (or calculate) the **average trip length** or a similar summary statistic if it is provided.

Questions:
- As cost C_{ij} increases, what happens to F_{ij}?
- Are most trips short-distance (within or between nearby zones) or long-distance?

Reflection & Insight

Excel here is acting as a "transparent" gravity model: you can see every formula and each step of the balancing iteration. Later we will hide these details inside Python code.

3 Activity 2 – Python Gravity Model (Student Input, Fixed β)

Now you move to Python and re-create the gravity model with your *own input data*. The notebook `Gravity_Model_Student_Input.ipynb` uses a fixed exponential friction function with $\beta = -0.035$.

Task 3: Load Notebook and Enter OD Data

Goal: Define a small OD system and run the gravity model with fixed β.
Steps:
1. Upload `Gravity_Model_Student_Input.ipynb` to Google Colab.
2. Run the initial cells that:
   - import required libraries (e.g. `numpy, pandas, matplotlib`),
   - define helper functions.
3. Find the input cell where you specify:
   - number of zones (e.g. 3 or 4),
   - Productions P_i (as a list or table),
   - Attractions A_j,
   - Cost matrix C_{ij}.
4. For a first test, you can copy the same P, A, and C values used in the Excel example.

Checkpoint: Make sure the sums of P and A are equal, or that the notebook explains how it handles differences.

Task 4: Run Gravity Model with Fixed β

Goal: Compute trips using the exponential gravity model with a fixed β.
Steps:
1. Locate where β is defined in the notebook (by default it should be `beta = -0.035` or similar).
2. Leave that value unchanged for this activity.
3. Run all remaining cells:
   - the gravity model iterations (balancing),
   - creation of the trip matrix T_{ij},
   - any plots (heatmaps, convergence curves),
   - calculation of average trip length.
4. When the notebook finishes, note down:
   - the final trip matrix,
   - average trip length,
   - how many iterations were needed to converge (if shown).

Comparison with Excel:
- Compare the Python T_{ij} with the Excel T_{ij} (for the same P, A and C).
- Is the difference small (only rounding)? If yes, the two implementations are consistent.

Reflection & Insight

If Python and Excel give the same numbers, you have successfully moved from a "white box" Excel model to a programmable model you can extend and repeat quickly.

4 Activity 3 – Interactive Gravity Model (Fixed P, A, C)

The notebook `Interactive_Gravity_Model_Lab.ipynb` keeps the OD system fixed but lets you change β with a slider to see how trip patterns and convergence respond.

Task 5: Explore β Slider with Preloaded Data

Goal: Understand qualitatively how different β values affect trip distribution.
Steps:
1. Upload `Interactive_Gravity_Model_Lab.ipynb` to Colab.
2. Run all cells from top to bottom.
3. Find the interactive widget (slider) for β; typical range might be from -0.01 to -0.20.
4. Try at least three values, for example:
   - $\beta = -0.01$ (low impedance),
   - $\beta = -0.035$ (medium),
   - $\beta = -0.10$ (high impedance).
5. For each value, observe:
   - the trip matrix (are trips more spread out or concentrated?),
   - the convergence plot (do iterations converge faster or slower?),
   - the friction curve $F(c) = e^{-\beta c}$.

Record briefly (1–2 sentences each):
- What happens to average trip length as β becomes more negative?
- Does convergence become easier or harder for extreme β values?

Reflection & Insight

Intuition check: a more negative β means a stronger penalty on long trips, so flows should concentrate on short-distance OD pairs and average trip length should shrink.

5 Activity 4 – Student Input + β Slider (Exponential Deterrence)

This activity combines *your own data* with an interactive β slider. You will work with `Gravity_Model_Student_Input(Exponential_Deterrence_Function).ipynb`.

Task 6: Define Your Own OD System

Goal: Create a small OD system and analyse it under different β values.
Steps:
1. Upload the notebook to Colab.
2. Run the import/setup cells.
3. In the input section, specify:
   - number of zones (4 or 5 is recommended),
   - Productions P_i for each zone,
   - Attractions A_j,
   - travel cost matrix C_{ij} (minutes or generalised cost).
4. Make sure your P and A values are realistic and roughly balanced (same total or close).

Example choices (you can change them):
- Zone 1: large residential area (high production),
- Zone 2: CBD (high attraction),
- Zone 3: university (medium production and attraction),
- Zone 4: industrial area, etc.

Task 7: Run Model for Several β Values

Goal: Perform a small sensitivity analysis on your OD system.
Steps:
1. Use the β slider in the notebook to select at least three different values, e.g.:
   - $\beta_1 = -0.01$,
   - $\beta_2 = -0.035$,
   - $\beta_3 = -0.10$.
2. For each β:
   - run the gravity model,
   - record the trip matrix,
   - note the average trip length,
   - look at the heatmap of T_{ij},
   - examine the convergence / error plot and friction curve.
3. Save screenshots or downloaded images of key plots for each β (you will need them for the homework report).

Questions:
- How does trip distribution shift when you move from β_1 to β_3?
- Which OD pairs lose/gain trips as the deterrence strengthens?

6 Activity 5 – Manual Friction Factor Matrix

The notebook `Gravity_Model_Manual_Friction.ipynb` allows you to directly supply the friction matrix F_{ij} instead of using a specific formula.

Task 8: Run Gravity Model with Custom F_ij

Goal: See how an arbitrary friction matrix changes trip patterns.
Steps:
1. Upload `Gravity_Model_Manual_Friction.ipynb` to Colab.
2. Run the setup cells.
3. Enter the same Productions P, Attractions A and cost matrix C you used earlier (if required).
4. In the section where the manual friction matrix F_{ij} is defined, create:
   - a version that strongly favours certain OD pairs (e.g. CBD links),
   - or a version that penalises one particular zone pair.
5. Run the gravity model with your manual F_{ij} and examine:
   - the resulting trip matrix,
   - any changes to convergence,
   - differences compared to the exponential friction function case.

Comment briefly:
- What happens if one F_{ij} is much larger/smaller than others?
- Does your manual matrix behave similarly to some exponential β, or is it more extreme?

Reflection & Insight

Manual F_{ij} gives you full control: you can mimic tolls, restricted areas or highly attractive corridors by tweaking specific entries in the matrix.

7 Activity 6 – Transport Demand Prediction (Regression)

The notebook `Transport_Demand_Regression_Lab.ipynb` demonstrates how to use regression to predict passenger demand based on route features such as distance and fare.

Task 9: Run Regression Notebook

Goal: Fit a simple regression model to transport demand data.
Steps:
1. Upload `Transport_Demand_Regression_Lab.ipynb` to Colab.
2. Run all cells from top to bottom:
   - loading the dataset,
   - exploratory data analysis (EDA),
   - splitting into training and testing sets,
   - training a linear regression model,
   - evaluating performance (e.g. R^2, error metrics).
3. Identify the key features used in the regression (e.g. distance, fare, route type).

Questions:
- Which variables have the strongest relationship with passenger demand?
- Is the model underpredicting or overpredicting any interesting routes?

Reflection & Insight

This regression model connects back to trip distribution: good demand forecasts help you set realistic productions and attractions for your zones.

8 Homework – Gravity Model with Sensitivity Exploration

Your homework is based on the gravity model with exponential friction and β sensitivity, as described in the Lab 6 slides (Section 6).

Task 10: Define Your Own OD System (4–5 Zones)

Goal: Build a small but meaningful OD system for analysis.
Steps:
1. Choose either a 4-zone or 5-zone system representing a simple city or region.
2. For each zone, define:
   - Production P_i (e.g. number of origin trips),
   - Attraction A_j (e.g. jobs, schools, services).
3. Create a travel cost matrix C_{ij} (in minutes or generalised cost).
4. Make sure your P and A are realistic and that total production roughly matches total attraction.

Task 11: Run Gravity_Model_Student_Input (Exponential) for Multiple β

Goal: Analyse how different β values affect your OD system.
Steps:
1. Open `Gravity_Model_Student_Input(Exponential_Deterrence_Function).ipynb`.
2. Input your P, A, and C.
3. Run the model for at least three different β values, for example:
   - $\beta_1 = -0.01$,
   - $\beta_2 = -0.035$,
   - $\beta_3 = -0.10$.
4. For each β, record:
   - the final trip matrix T_{ij},
   - the convergence behaviour (iterations, error plot),
   - the average trip length,
   - the heatmap of trips,
   - the friction curve $F(c)$.
5. Save your modified `.ipynb` file and export plots (screenshots or images) for the report.

Task 12: Analyse and Report

Goal: Summarise and interpret your findings in a short report.
Minimum content:
- A short description of your OD system (zones, P, A, C).
- A section comparing the three β scenarios:
  - how trip distribution changed,
  - how convergence speed and errors changed,
  - how average trip length and friction curves changed.
- 2–3 paragraphs of discussion linking your observations to transport planning:
  - why stronger deterrence leads to more local trips,
  - when a planner might want a "high-β" vs "low-β" model.
- Conclusion with your main insights.
- Attach (or embed) selected trip matrices, heatmaps and plots as figures or appendices.

Deliverables:
- Your modified `.ipynb` file.
- A PDF report (structure can follow the Lab 6 slide suggestion: title, objectives, introduction, methodology, β results, discussion, conclusion, references, appendices).

Reflection & Insight

Focus on the *story*: how does changing β reshape travel patterns in your small "city" and what does that imply for planning decisions (e.g. where to invest in capacity or new services)?