**Table of Contents**

|  |  |
| --- | --- |
| **Topic** | **Page no.** |
| **Ch-1**  **Introduction** | **3** |
| **Ch-2**  **Litreture survey** | **4** |
| **Ch-3**  **Methodolgy and Implementation** | **5** |
| **Ch-4**  **Result and analysis** | **20** |
| **Ch-5**  **References** | **22** |

**Chapter 1**

**Introduction**

**Introduction** (of the project, and the flow of different chapters)

1. Consider a company whose shipping and transportation data of products from different plants and ports is given, the data given describes the number of plants available as a pickup source for the data and further shows the destination ports where the product is to be dropped off.
2. The additional data given is what is the crux of the discussion here, this is what applies the conditions on our primary problem of minimizing the transportation costs between the plants and the ports. The data set gives us the total demand at the destination points as well as the total supply present at the plants to fulfill these needs by devising a solution to help the company in minimizing the transportation costs. The data given in the data set is the data that gives us the data for a transportation simplex problem.
3. We’ll be solving the problem of this transportation, this will be implemented in python using multiple libraries like pandas, Numpy and PULP.
4. Following the above, we’ll proceed to The SAS Viya software, which is a visual dashboard and data representation tool, Using the services by platform, we’ll create multiple analytical graph, to recheck the solution found and predictions made using python to those results given by the graphs.
   1. **Problem statement:**

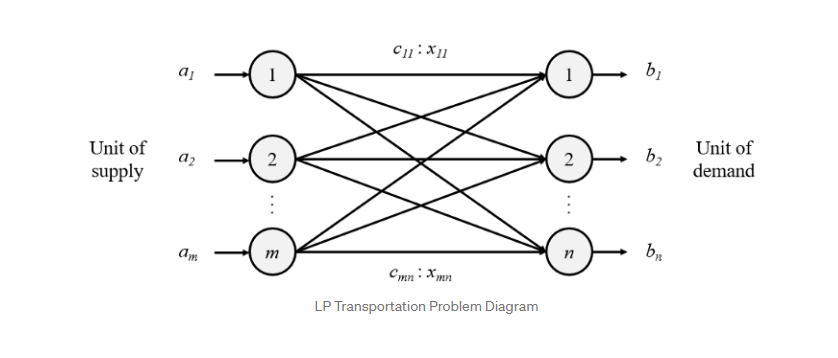
This project aims at providing the optimal solution for minimum transportation costing based on two decision variables along with a pictorial overview of the problem.

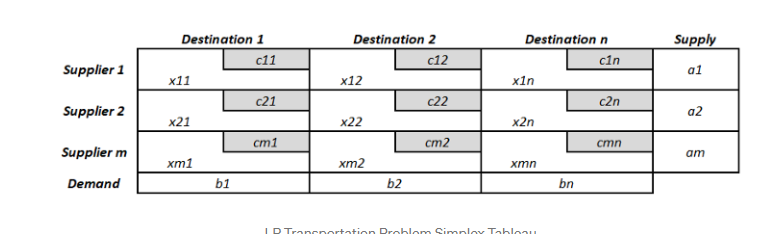
The transportation problem is a special type of linear programming problem where the objective consists in minimizing transportation costof a given commodity from a number of **PLANTS**to a number of **PORTS**. Each source has a limited supply (i.e. maximum number of products that can be sent from it) while each destination has a demand to be satisfied (i.e. minimum number of products that need to be shipped to it). The cost of shipping from a source to a destination is directly proportional to the number of units shipped.

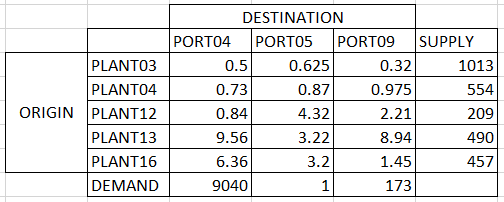
**Chapter 2**

**Literature survey**

The transportation problem is a special type of linear programming problem where the objective is to minimize the cost of distributing a product from a number of sources or origins to a number of destinations. Because of its special structure the usual simplex method is not suitable for solving transportation problems. These problems require a special method of solution. The origin of a transportation problem is the location from which shipments are dispatched. The destination of a transportation problem is the location to which shipments are transported. The unit transportation cost is the cost of transporting one unit of the consignment from an origin to a destination.

****

****

****

There are two different types of transportation problems based on the initial given information:

Balanced transportation problems are those in which the total supply and demand are equal.

Unbalanced Transportation Problems**:**cases where the total supply is not equal to the total demand.

In order to proceed with the solution of any given transportation problem, the first step consists in verifying if it is balanced. If it is not, it must be balanced accordingly.

**SAS VIYA:**

In order to mine, modify, organize, and retrieve data from many sources and conduct statistical analysis on it, SAS VIYA is used. Through the SAS language, SAS offers non-technical users a graphical point-and-click user interface in addition to other features. Declarative statements that give instructions on how to read a data set or change the look of the data are present in the DATA phase together with executable statements that cause the software to take an action.

**Python:**

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming. We shall be solving the above transportation simplex problem using the functions of the libraries PuLP, NUMPY and Pandas.

**Chapter 3**

**Methodology and Implementation**

**3.1 Implementation:**

The Implementation to this problem consists of three parts:

1.Data Cleaning and Exploration

2.Data Visualization

3.Data Modeling and Conclusions

**Phase 1:** Data Cleaning and Exploration

In this phase, we load the datasets in the colab notebook and use the csv library for cleaning and exploration. We check if the field titles and if all the data elements have been loaded successfully along with the modifications we require to solve our problem. We can also view data insights using functions like head(), info() or describe().Furthermore, we sliced our dataset with the columns we want to use and removed any unwanted columns. We also removed any blank values the dataset might have. Lastly, the demand for some origin plants was negligible compared to the others and we hence removed them.

**Phase 2:** Data Visualization

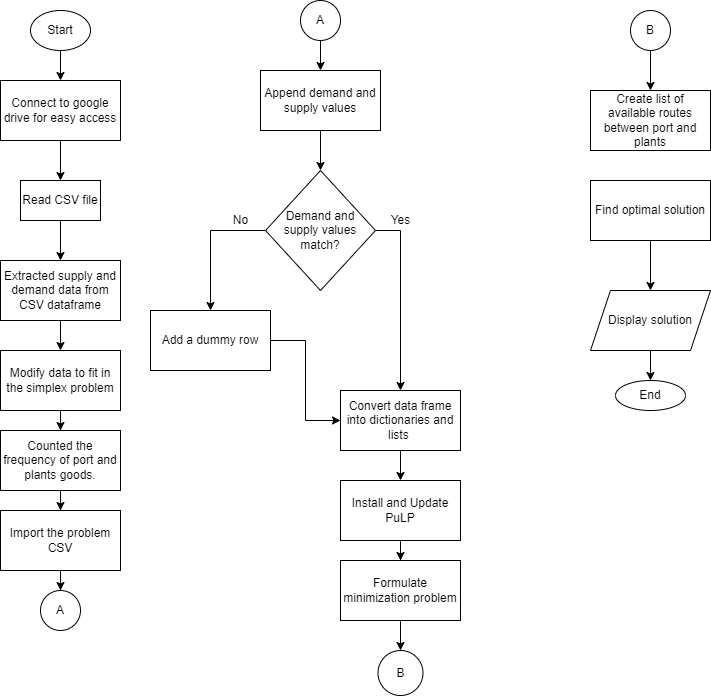
This phase of Data Visualization is implemented using SAS Visual Analytics Software. We load the dataset into the SAS Lasr Server and then create a dashboard. We can make meaningful graphs and charts to give us valuable insight into the transportation scenario of the company. We also need to keep in mind to implement the right color scheme to make the dashboard visually appealing.

Phase 3: Data Modeling and drawing Conclusions

Finally, the data is analyzed and worked on before it gives us an optimal solution to our transportation problem using python programming and business analytics and modeling techniques. Transportation problems are solved using linear programming concepts. We can conclude with the optimal solution for the given transportation problem.

**3.1 Blockdiagram/ flowchart / algorithm**

**Flowchart/blockdiagram:**

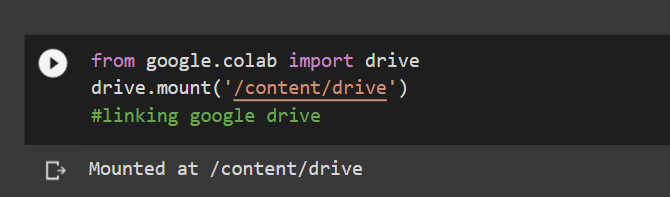
****

**Algorithm:**

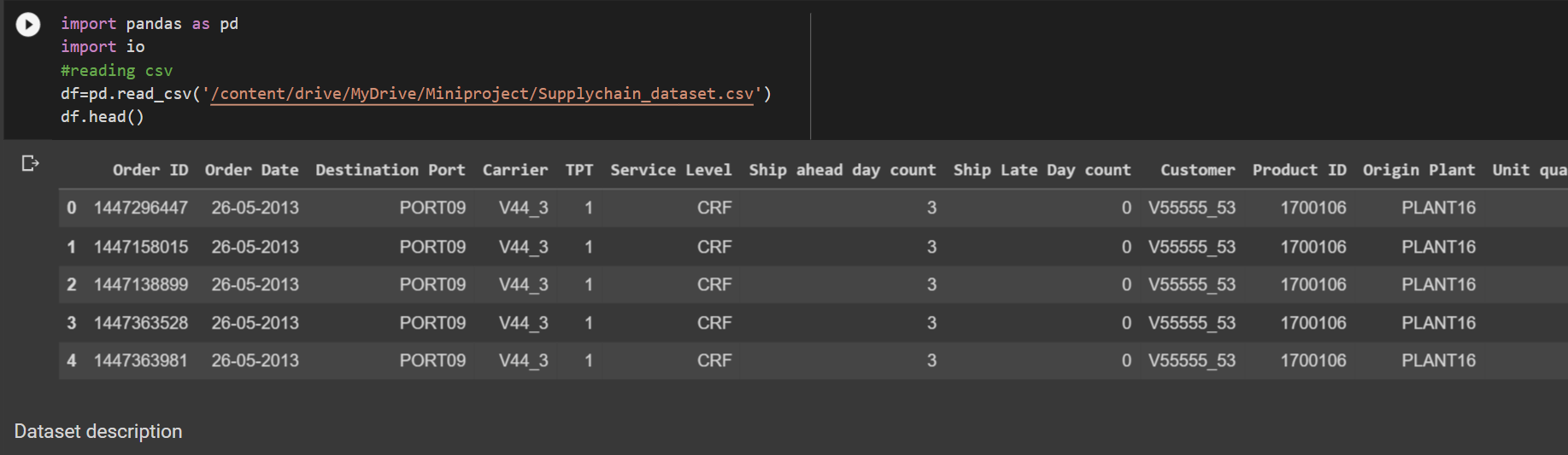
* Mounted google drive to obtain easy access to csv files
* Read csv file as a pandas dataframe
* Extracted supply and demand data from csv dataframe
* Arranged and modified supply and demand data to fit into the simplex problem
* Counted the frequency of port and plant goods
* Imported problem csv as pandas dataframe
* Appended the demand and supply values
* Modified the problem dataframe (axes, new columns, rows)
* Validation check: in case, a dummy row or column was needed if demand and supply values don’t match
* Converted dataframes into dictionaries and lists to be solved using PuLP library
* Installed and imported PuLP
* Formulated minimization problem
* Created list of available routes between ports and plants
* Defined decision variables, the objective function and the constraints
* Found optimal solution for all variables and minimisation answer

**CODE:**

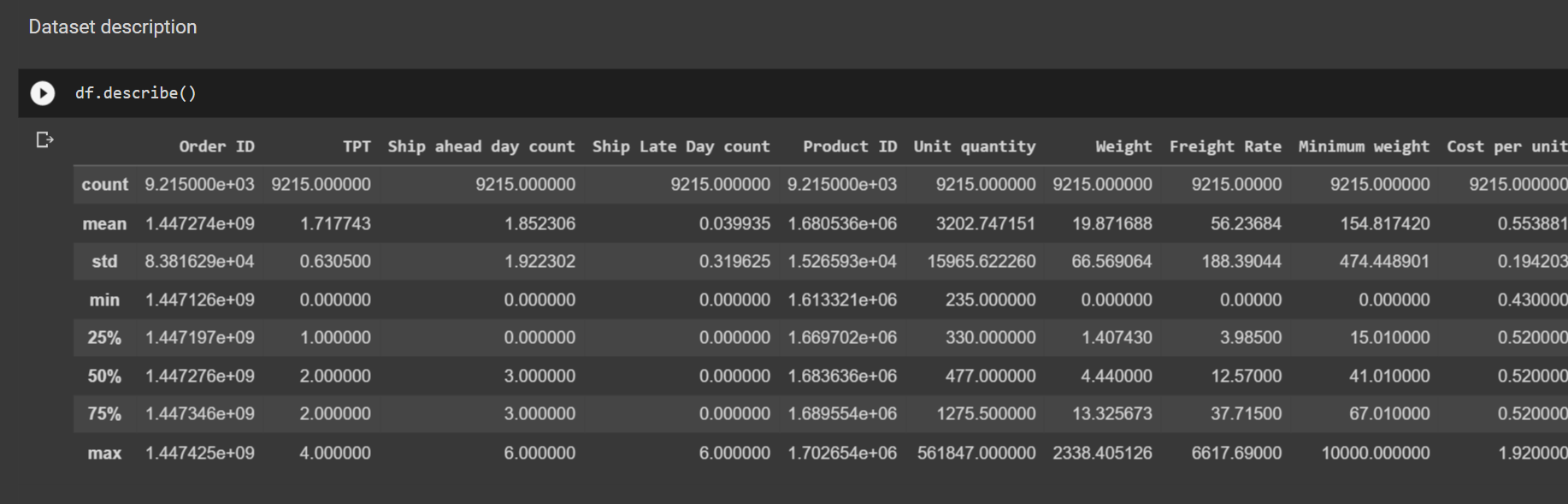
Linking the google drive to the project:

****

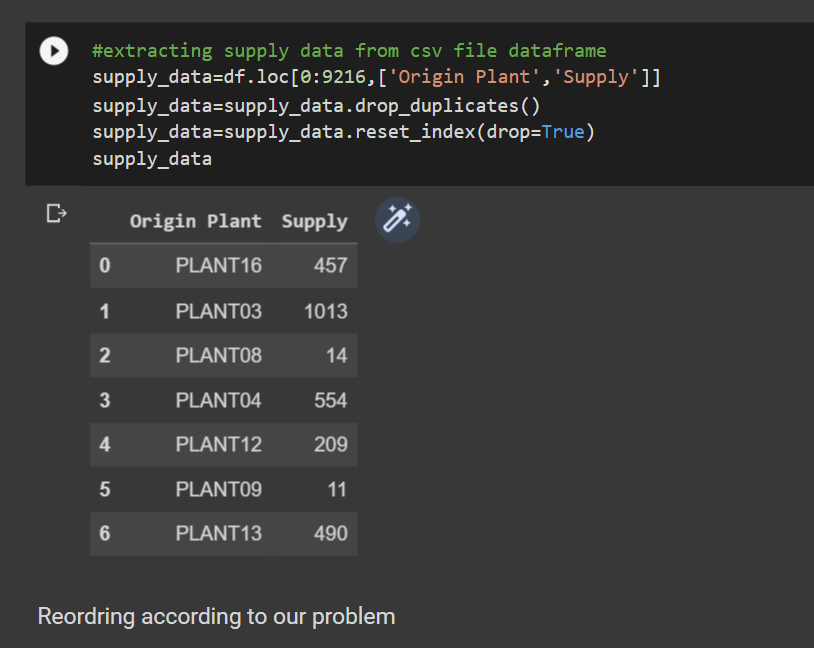
Importing the dataset from csv file as a pandas dataframe.



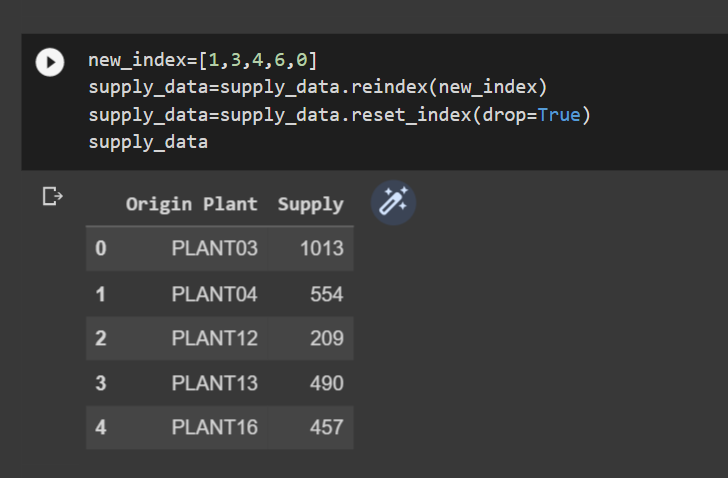
Describing the dataset using basic statistical tools:



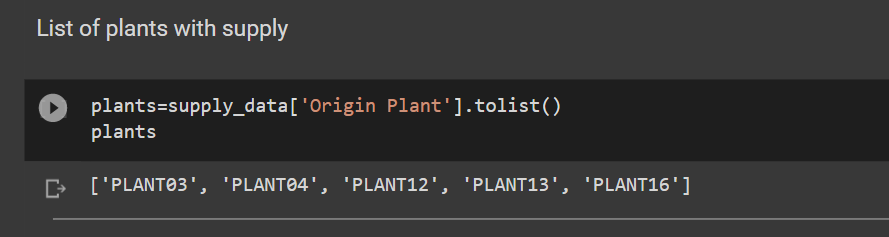
Using multiple functions here to extract the supply data only from the dataset. Manipulating this new supply dataframe by resetting the index and dropping duplicates.



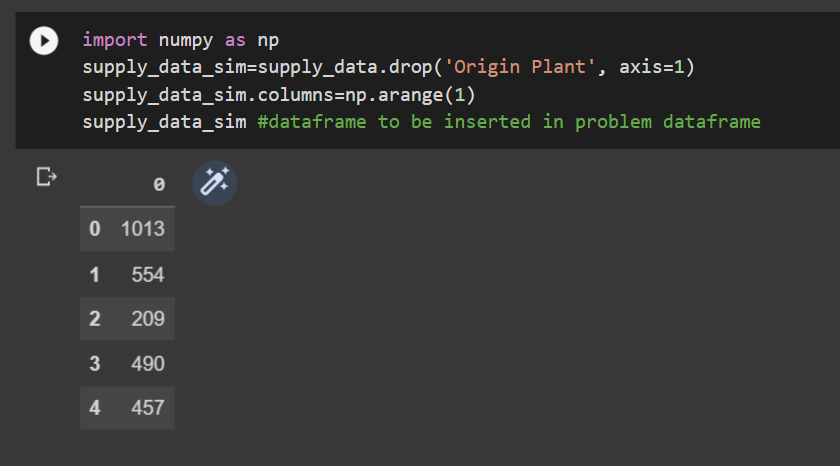
Reordering data according to our use in simplex:



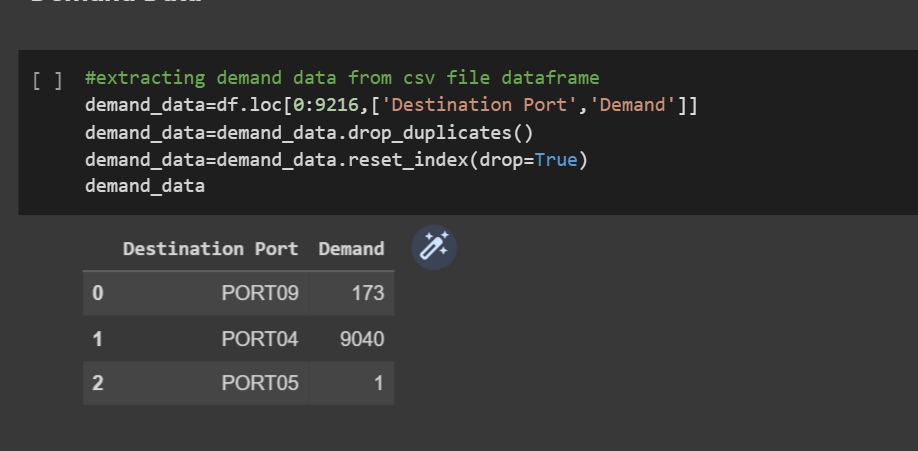
List of plants that supply:



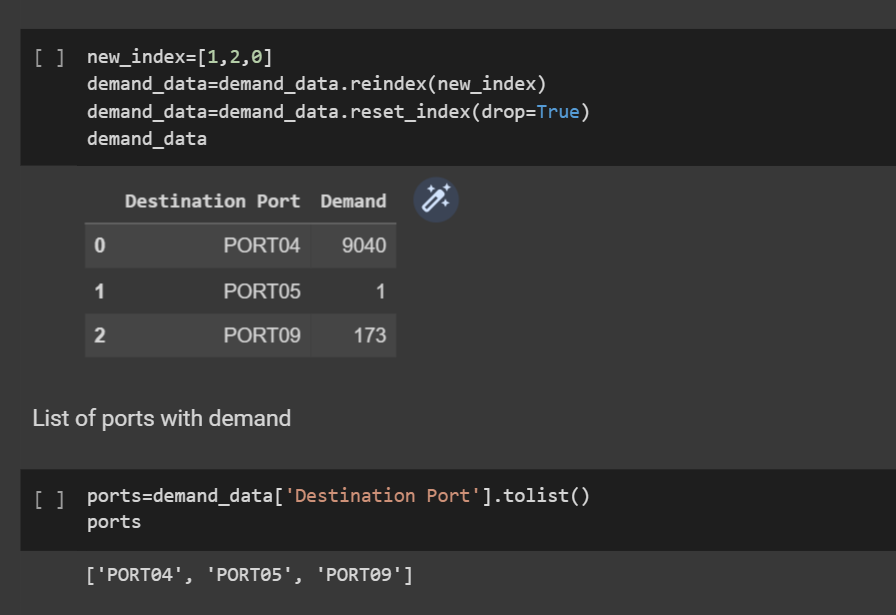
Importing numpy and using its functions to modify the data in a way which enables us to add this dataframe to the problem dataframe directly :



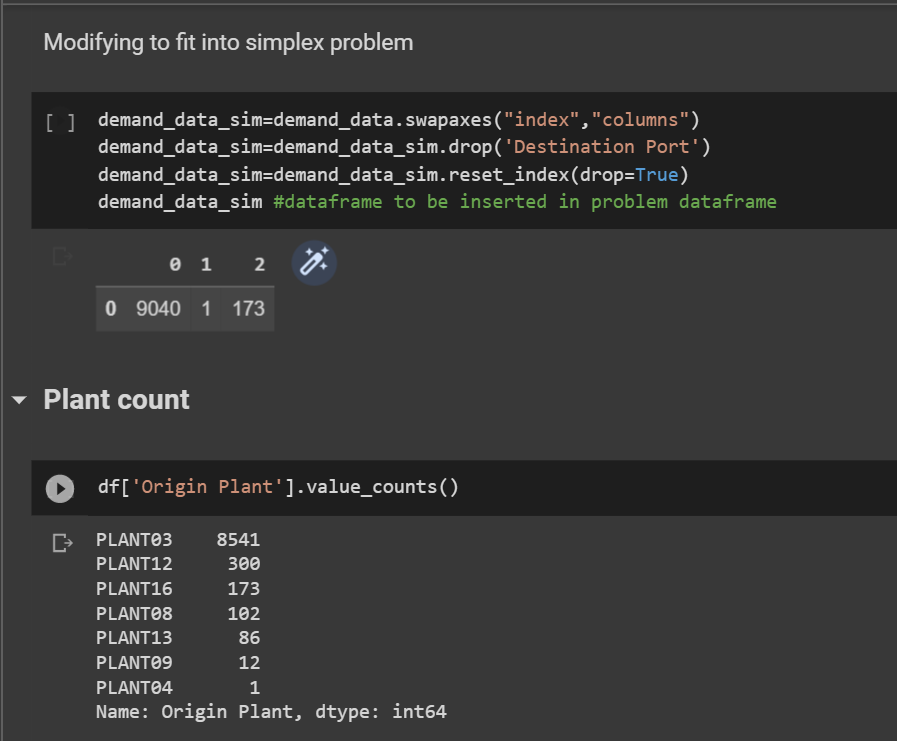
Extracting the demand data from the dataset:



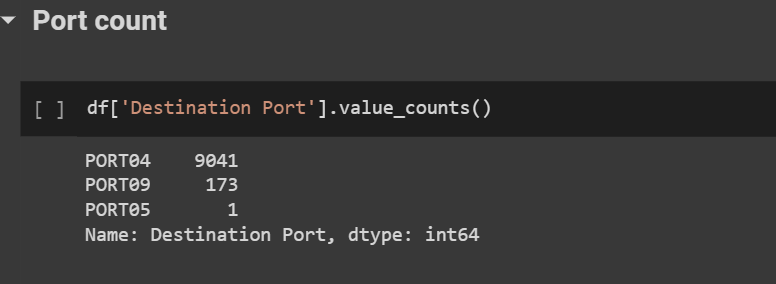
Modifying data to fit the simplex and removing duplicates:



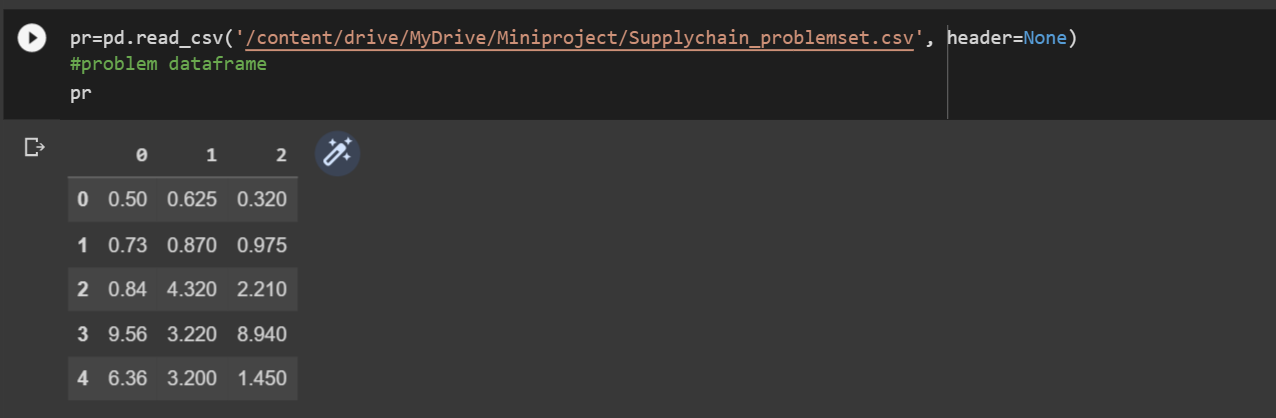
Modifying this new demand dataframe to be fit directly into the problem data:



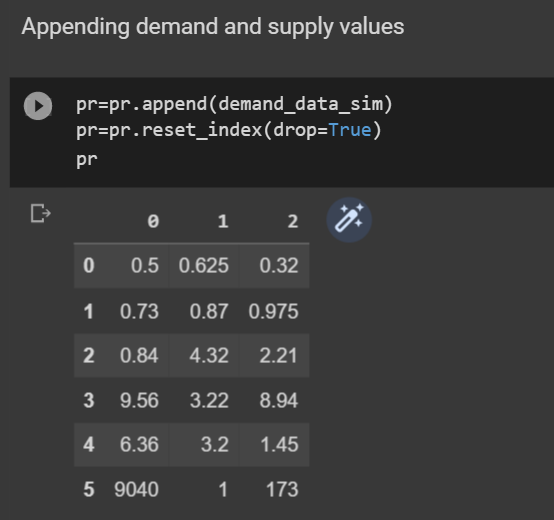
Counting the number of shipments to and from ports and plants respectively.

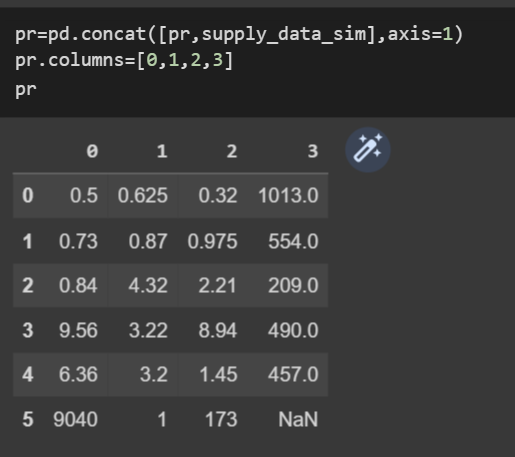


Reading the problem dataset from csv file:

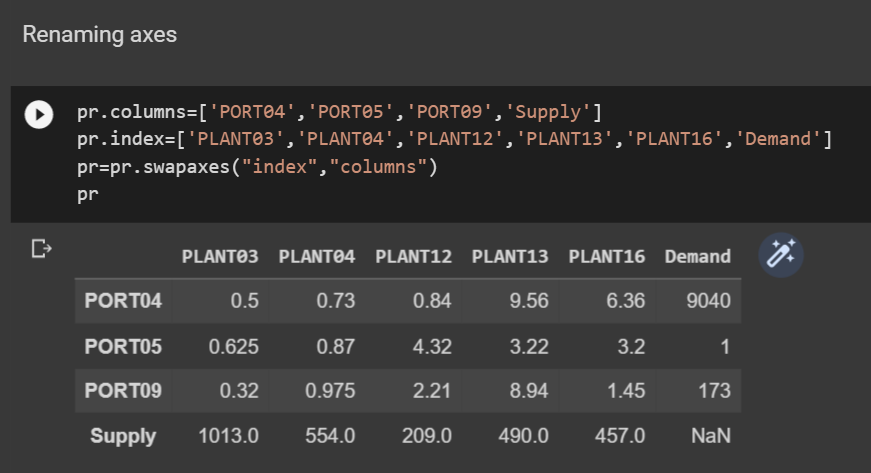


Adding the demand and supply values to our simplex problem

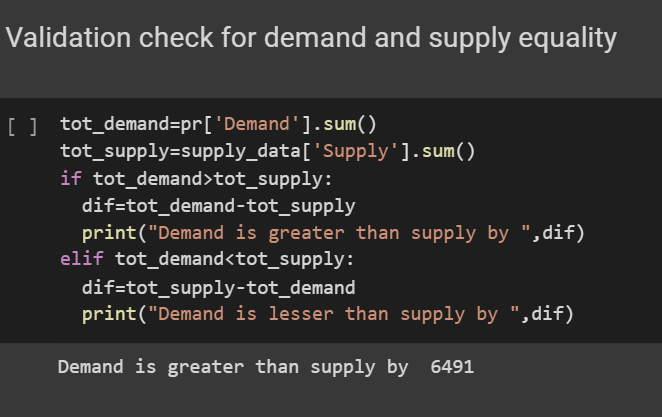




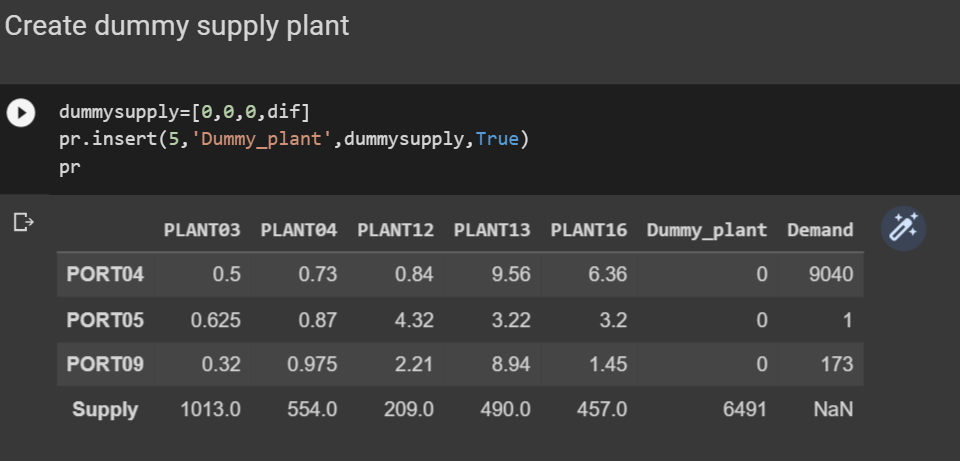
Labeling our problem dataframe:



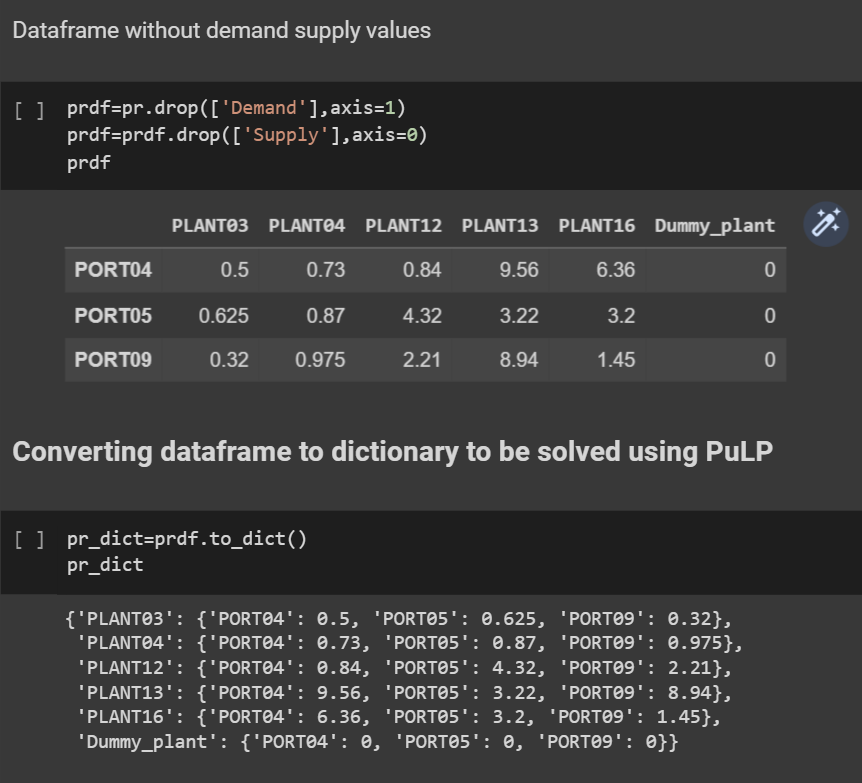
Checking demand and supply equality:



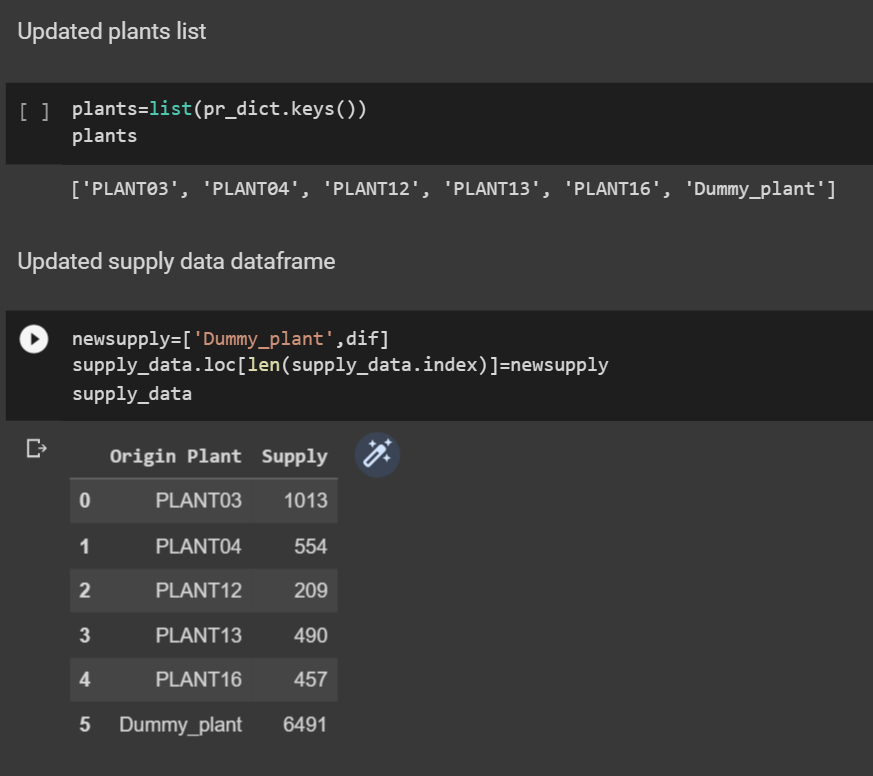
Since demand is higher we create a dummy supply plant:



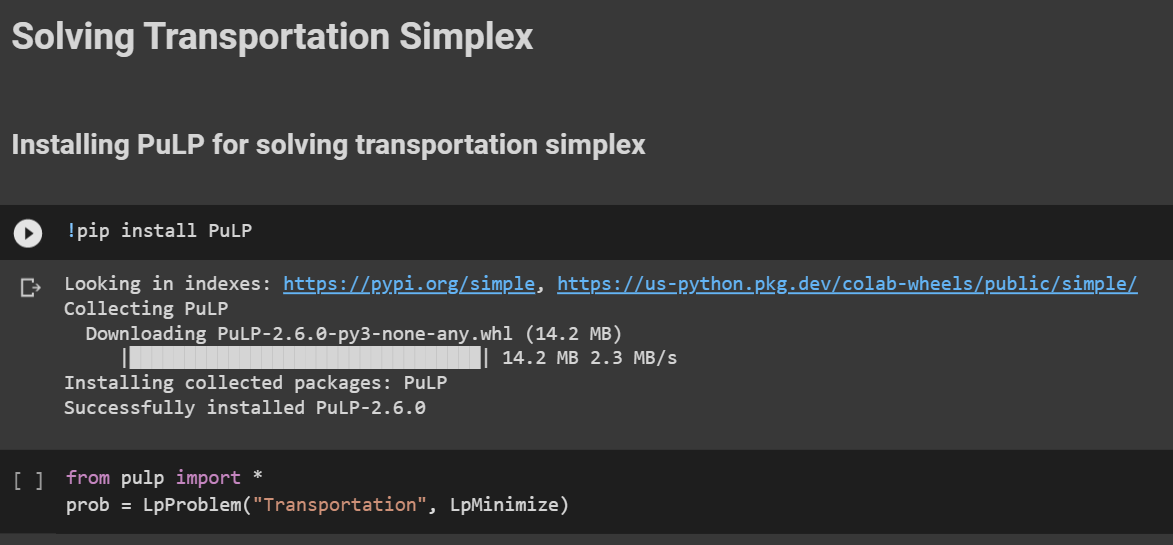
Removing demand and supply values to create a dictionary for the problem data:



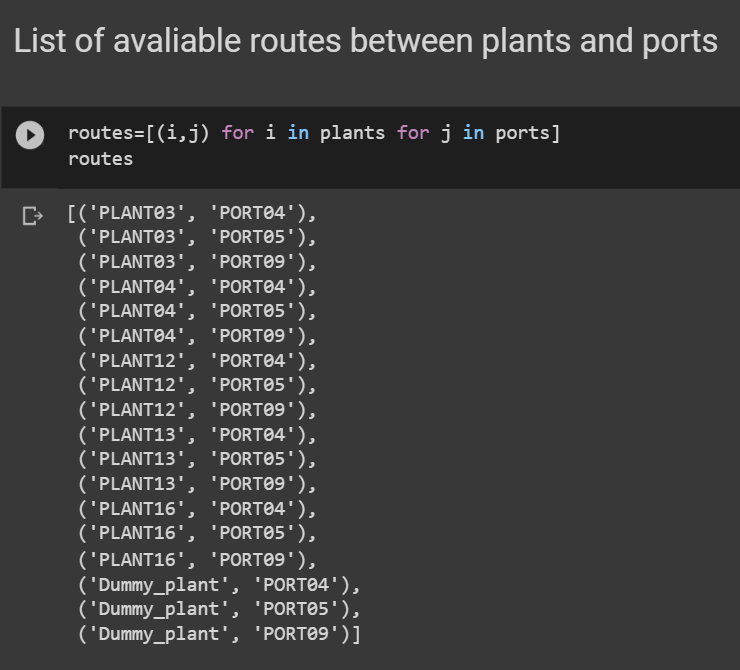
Updated the plants list and dataframe:



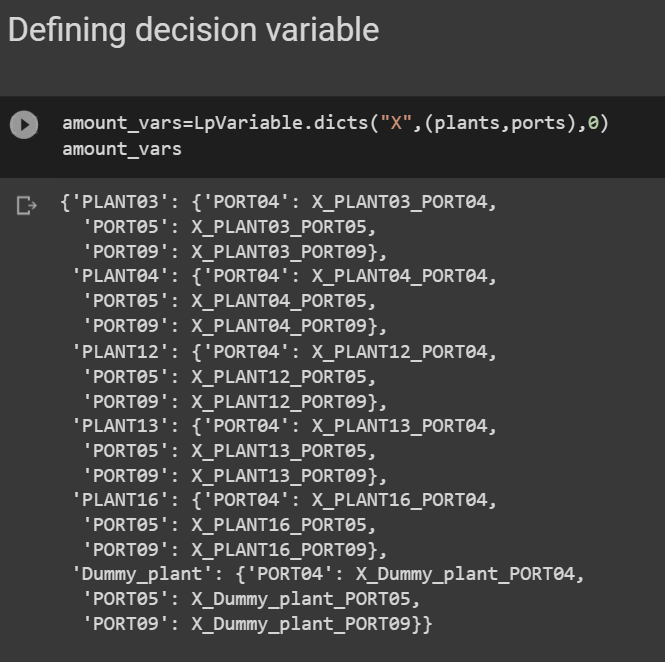
Importing PuLP to solve simplex and creating transportation minimization problem:



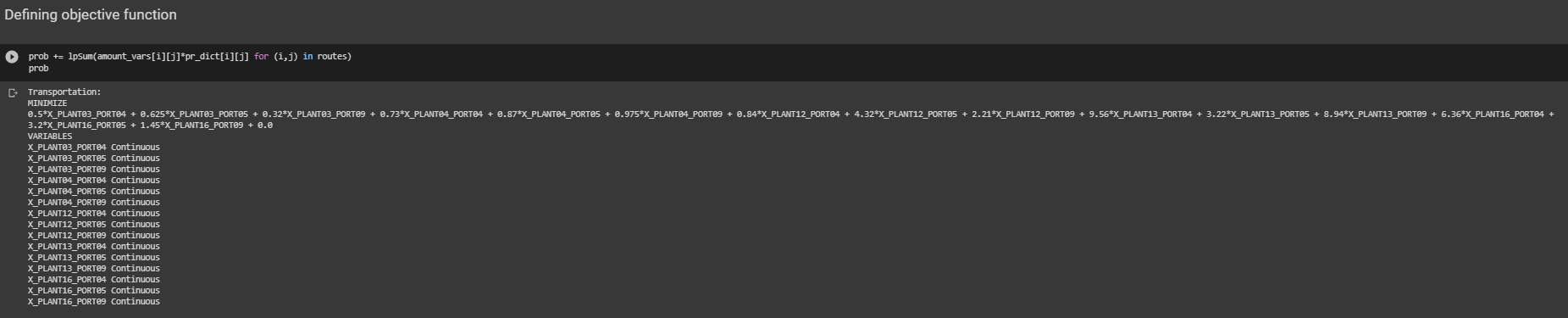
Create list of available routes between plants and ports



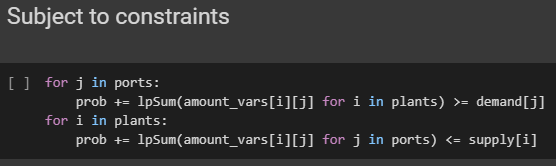
Creating decision variable:



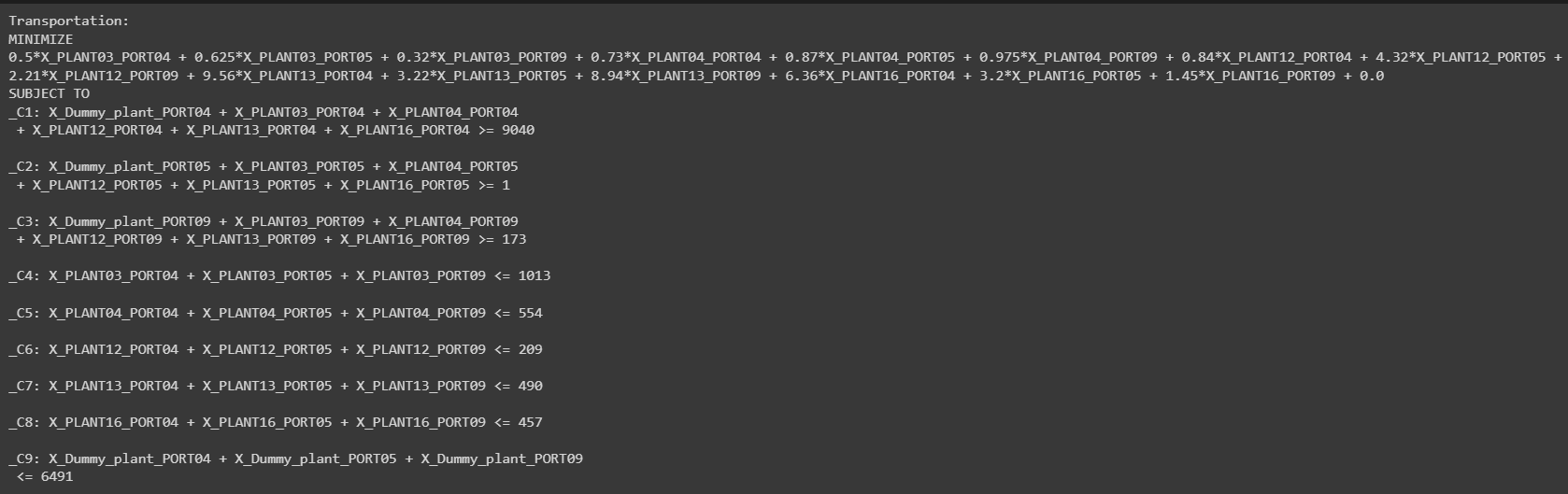
Creating objective function:

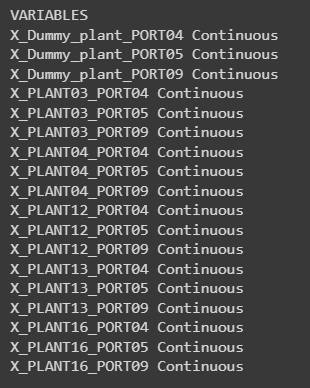


Adding constraints to problem:

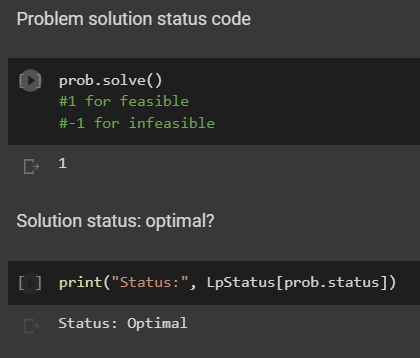


Full problem:

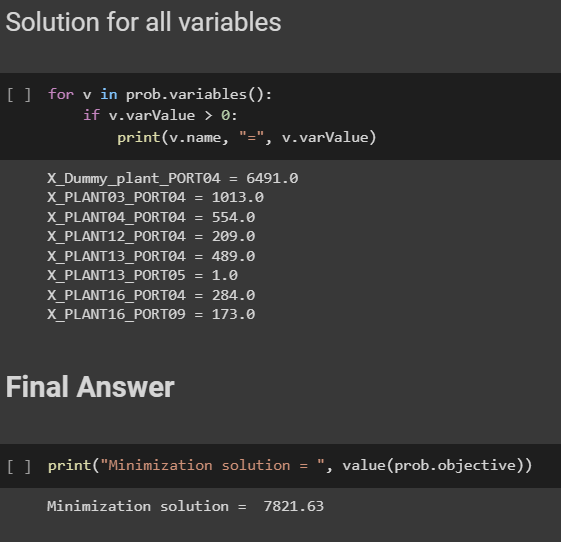




Checking if problem can be solved:

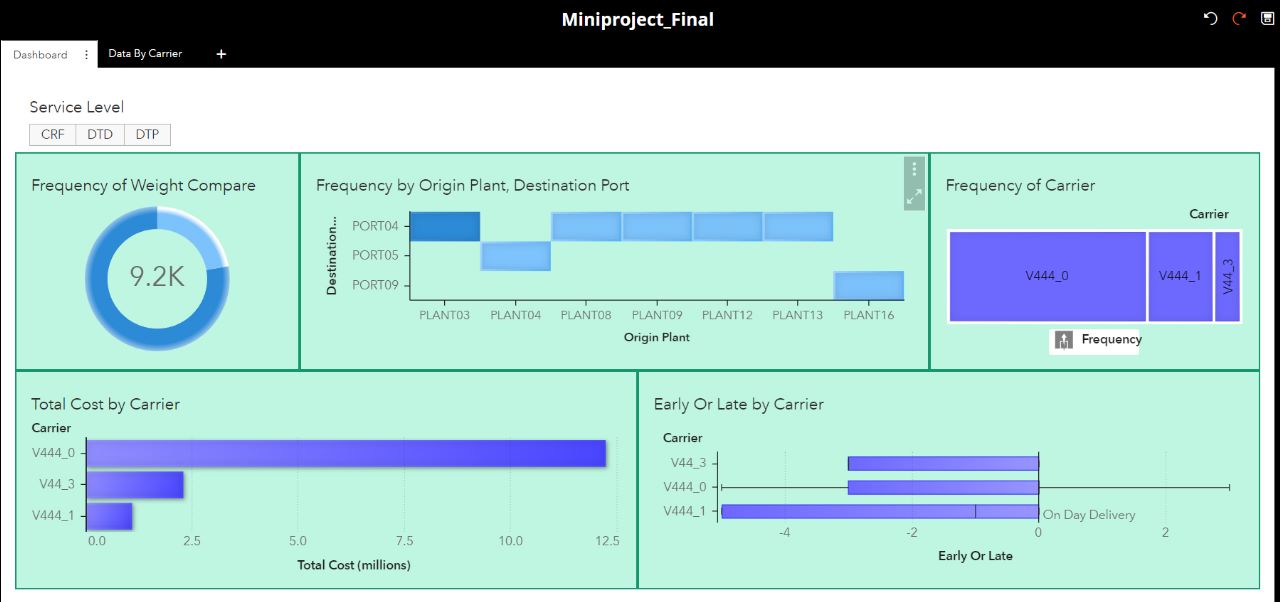


Solution to transportation simplex:

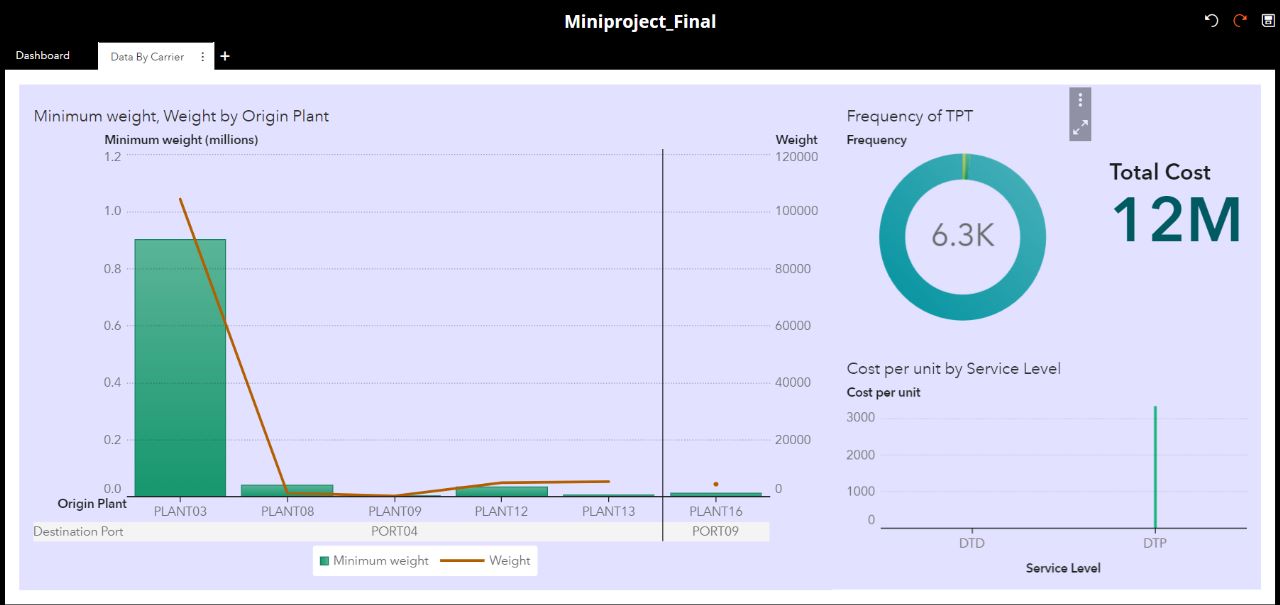


**DASHBOARDS AND GRAPHS:**

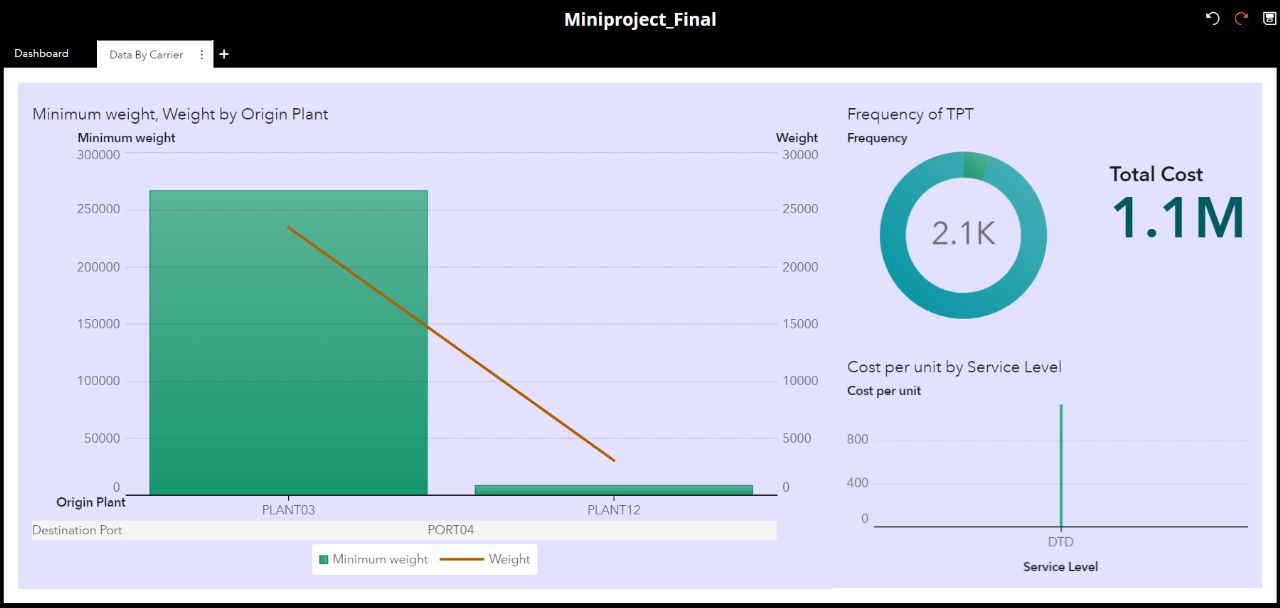
Main page:

****

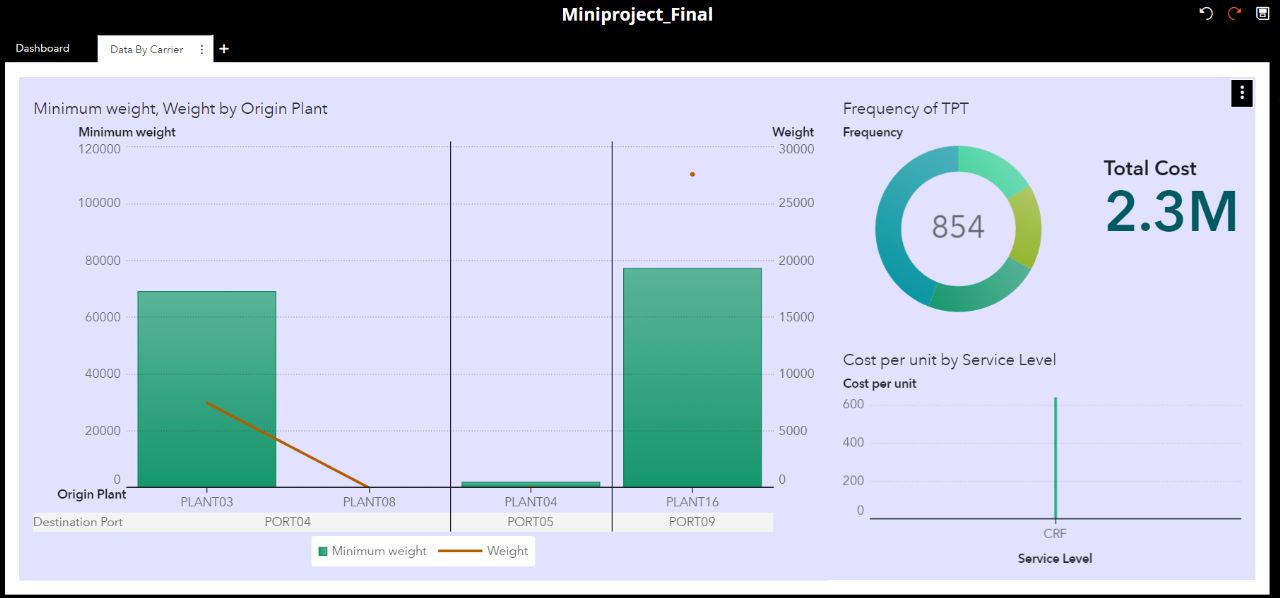
Data by carrier V444\_0:

****

Data by carrier V444\_1:

****

Data by carrier V44\_3:

****

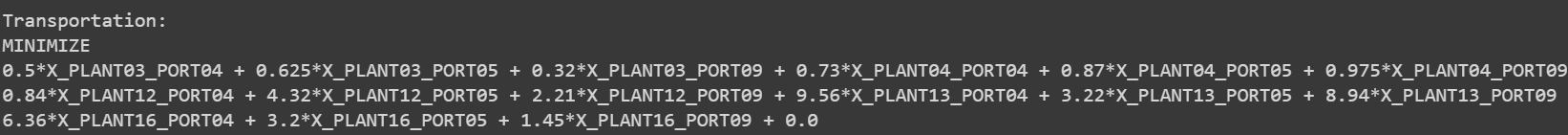
**Chapter 4**

**Results and Analysis**

Thus, we have formulated and solved our transportation problem and in addition, created a dashboard to represent important data regarding the same.

Formulated transportation simplex problem:

Objective function:



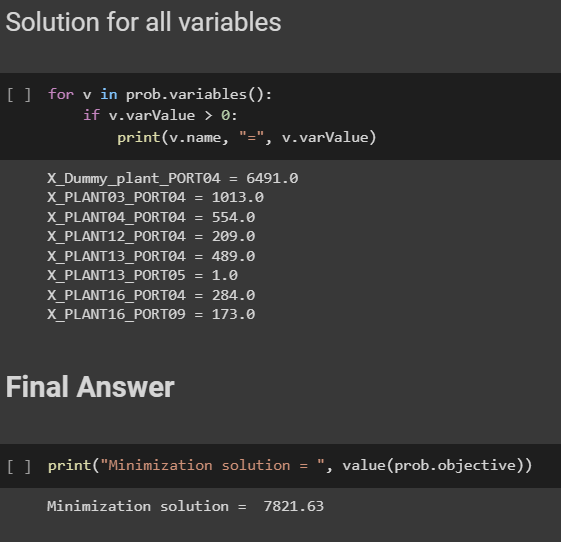
Constraints:



Variables:



Solution to transportation simplex:



**Chapter 5**

**References**

[**https://pydata.org/**](https://pydata.org/)

[**https://pandas.pydata.org/**](https://pandas.pydata.org/)

[**https://www.geeksforgeeks.org/python-pandas-dataframe/**](https://www.geeksforgeeks.org/python-pandas-dataframe/)

[**https://radzion.com/blog/operations/transportsimplex**](https://radzion.com/blog/operations/transportsimplex)

[**https://www.w3schools.com/python/pandas/default.asp**](https://www.w3schools.com/python/pandas/default.asp)

[**https://coin-or.github.io/pulp/**](https://coin-or.github.io/pulp/)

[**https://pypi.org/project/PuLP/**](https://pypi.org/project/PuLP/)

[**https://www.coin-or.org/PuLP/CaseStudies/a\_transportation\_problem.html**](https://www.coin-or.org/PuLP/CaseStudies/a_transportation_problem.html)

[**https://towardsdatascience.com/linear-programming-and-discrete-optimization-with-python-using-pulp-449f3c5f6e99**](https://towardsdatascience.com/linear-programming-and-discrete-optimization-with-python-using-pulp-449f3c5f6e99)