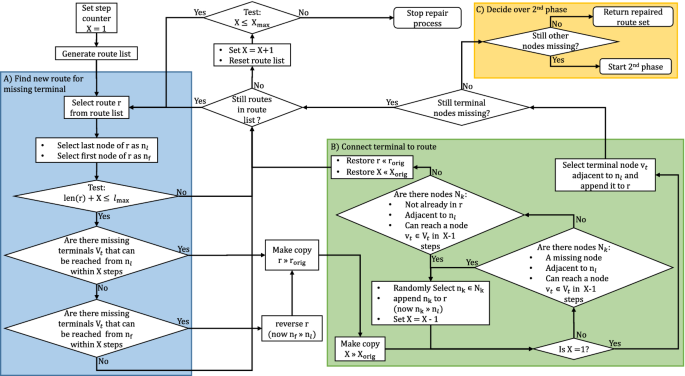
**Public Transportation Optimization**

***In order to be environment-friendly, relieve traffic congestion, reduce pollution, and be green and sustainable, the optimization and development of public transportation, as the subject of people's long-term research, has always been shining. With the emergence of shared transportation, public transportation systems face more challenges. In order to better connect with bike-sharing, car-sharing, and other modes of transportation, public transportation will carry out important reforms, among which the optimization of line network is one of the most important tasks. The traditional bus route design is mainly based on the “four-stage” method model, which is mainly based on the investigation and analysis of the existing traffic system and land use. Through the work flow of “evaluation, calibration, and verification,” the network balance optimization model is used to get the bus travel allocation prediction model. In this paper, the optimization problem of public transit network is studied from the point of view of the reliability of public transit network. It is proposed that public transit network can be abstracted into series-parallel system and parallel-series system model from the three states of normal, short-circuit failure, and open-circuit failure and is analyzed and discussed through the hypothesis experiment. The research of this paper will provide a new perspective for the optimization of public transit network, complement the traditional methods, and support the optimization and reliability improvement of urban public transit network. More reliable bus networks and other modes of transportation, such as walking, bike-sharing, and rail, will become more suitable for people to get around.***

FLOWCHART



**Data sheet for PTO**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Mode | Average Car | Compact Car | Electric Car | Van OR Pickup | Rideshare Passenger | Diesel Bus | | Average Occupancy | 1.5 | 1.5 | 1.5 | 1.5 | 1 | 8 | | Vehicle Ownership | 0.272 | 0.237 | 0.387 | 0.354 | 0 | 0 | | Vehicle Operation | 0.169 | 0.239 | 0.234 | 0.345 | 0.008 | 0 | | Operating Subsidy | 0 | 0.123 | 0.304 | 0.035 | 0.011 | 1.982 | | Travel Time | 0.234 | 0.456 | 0 | 0.087 | 0.201 | 2.675 | | Internet Crash | 0.23 | 0.45 | 0 | 0.509 | 0.022 | 0.564 | | External Crash | 0.245 | 0.453 | 0.345 | 0.303 | 0.452 | 0.345 | | Internal Health Ben | 0.089 | 0.098 | 0.879 | 0.345 | 0.562 | 0.786 | | External Health Ben | 0.344 | 0.034 | 0.341 | 0.562 | 0.709 | 0.431 | | Internal Parking | 0.786 | 0.304 | 0.234 | 0.345 | 0.034 | 0.109 | | External Parking | 0.346 | 0.345 | 0.555 | 0.987 | 0.345 | 0.054 | | Congestion | 0.445 | 0.478 | 0.658 | 0.056 | 0.785 | 0.0435 | | Road Facilities | 0.654 | 0.405 | 0.675 | 0.044 | 0.203 | 0.765 | | Land value | 0.124 | 0.452 | 0.456 | 0.34 | 0.098 | 0.085 | | Traffic service | 0.055 | 0.785 | 0.675 | 0.558 | 0.034 | 0.097 | | Transport Diversity | 0.323 | 0.421 | 0.278 | 0.054 | 0.352 | 0.453 | | Air Pollution | 0.345 | 0.321 | 0.513 | 0.095 | 0.766 | 0.0954 | | GHG | 0.476 | 0.702 | 0.712 | 0.088 | 0.456 | 0.564 | | Noise | 0.345 | 0.432 | 0.345 | 0.345 | 0.76 | 0.675 | | Resource Extemalitie | 0.123 | 0.457 | 0.675 | 0.675 | 0.705 | 0.543 | | Land Yse Impacts | 0.478 | 0.564 | 0.132 | 0.658 | 0.321 | 0.648 | | Water Pollution | 0.378 | 0.872 | 0.358 | 0.488 | 0.781 | 0.546 | | Waste | 0.0004 | 0.0004 | 0.0004 | 0.0004 | 0.0000 | 0.0004 | | ***TOTALS*** | 6.4614 | 8.6284 | 8.7564 | 7.2734 | 7.605 | 11.4613 |   Mode | Average Car | Compact Car | Electric Car | Van OR Pickup | Rideshare Passenger | Diesel Bus | Electric Trolley | Motor-cycle | Bicycle | Walk | Telework |
| Average Occupancy | 1.5 | 1.5 | 1.5 | 1.5 | 1 | 8 | 10 | 1 | 1 | 1 | 1 |
| Vehicle Ownership | 0.272 | 0.237 | 0.387 | 0.354 | 0 | 0 | 0 | 0.333 | 0.066 | 0 | 0.264 |
| Vehicle Operation | 0.169 | 0.239 | 0.234 | 0.345 | 0.008 | 0 | 0 | 0.264 | 0.066 | 0 | 0.264 |
| Operating Subsidy | 0 | 0.123 | 0.304 | 0.035 | 0.011 | 1.982 | 1.802 | 0.000 | 0.654 | 0.654 | 0.000 |
| Travel Time | 0.234 | 0.456 | 0 | 0.087 | 0.201 | 2.675 | 3.828 | 0.000 | 0.098 | 0.673 | 0.000 |
| Internet Crash | 0.23 | 0.45 | 0 | 0.509 | 0.022 | 0.564 | 0.098 | 0.000 | 0.43 | 0.613 | 0.000 |
| External Crash | 0.245 | 0.453 | 0.345 | 0.303 | 0.452 | 0.345 | 0.054 | 0.000 | 0.985 | 0.912 | 0.000 |
| Internal Health Ben | 0.089 | 0.098 | 0.879 | 0.345 | 0.562 | 0.786 | 0.986 | 0.000 | 0.543 | 0.431 | 0.000 |
| External Health Ben | 0.344 | 0.034 | 0.341 | 0.562 | 0.709 | 0.431 | 0.134 | 0.000 | 0.621 | 0.312 | 0.000 |
| Internal Parking | 0.786 | 0.304 | 0.234 | 0.345 | 0.034 | 0.109 | 0.431 | 0.000 | 0.342 | 0.412 | 0.000 |
| External Parking | 0.346 | 0.345 | 0.555 | 0.987 | 0.345 | 0.054 | 0.543 | 0.988 | 0.612 | 0.324 | 0.000 |
| Congestion | 0.445 | 0.478 | 0.658 | 0.056 | 0.785 | 0.0435 | 0.456 | 0.543 | 0.154 | 0.813 | 0.000 |
| Road Facilities | 0.654 | 0.405 | 0.675 | 0.044 | 0.203 | 0.765 | 0.123 | 0.903 | 0.041 | 0.512 | 0.001 |
| Land value | 0.124 | 0.452 | 0.456 | 0.34 | 0.098 | 0.085 | 0.034 | 0.043 | 0.812 | 0.432 | 0.000 |
| Traffic service | 0.055 | 0.785 | 0.675 | 0.558 | 0.034 | 0.097 | 0.564 | 0.000 | 0.564 | 0.442 | 0.000 |
| Transport Diversity | 0.323 | 0.421 | 0.278 | 0.054 | 0.352 | 0.453 | 0.564 | 0.094 | 0.513 | 0.221 | 0.000 |
| Air Pollution | 0.345 | 0.321 | 0.513 | 0.095 | 0.766 | 0.0954 | 0.435 | 0.654 | 0.543 | 0.012 | 0.000 |
| GHG | 0.476 | 0.702 | 0.712 | 0.088 | 0.456 | 0.564 | 0.054 | 0.765 | 0.754 | 0.412 | 0.004 |
| Noise | 0.345 | 0.432 | 0.345 | 0.345 | 0.76 | 0.675 | 0.453 | 0.432 | 0.342 | 0.812 | 0.000 |
| Resource Extemalitie | 0.123 | 0.457 | 0.675 | 0.675 | 0.705 | 0.543 | 0.321 | 0.675 | 0.614 | 0.221 | 0.000 |
| Land Yse Impacts | 0.478 | 0.564 | 0.132 | 0.658 | 0.321 | 0.648 | 0.45 | 0.000 | 0.142 | 0.612 | 0.000 |
| Water Pollution | 0.378 | 0.872 | 0.358 | 0.488 | 0.781 | 0.546 | 0.876 | 0.875 | 0.531 | 0.342 | 0.083 |
| Waste | 0.0004 | 0.0004 | 0.0004 | 0.0004 | 0.0000 | 0.0004 | 0.0004 | 0.000 | 0.0000 | 0 | 0.0000 |
| ***TOTALS*** | 6.4614 | 8.6284 | 8.7564 | 7.2734 | 7.605 | 11.4613 | 12.2064 | 6.5694 | 9.427 | 9.162 | 0.616 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Electric Trolley | Motor-cycle | Bicycle | Walk | Telework | Internet | Fixed | Market |
| 10 | 1 | 1 | 1 | 1 |  |  |  |
| 0 | 0.333 | 0.066 | 0 | 0.264 | 100% | 100% | 100% |
| 0 | 0.264 | 0.066 | 0 | 0.264 | 100% | 100% | 100% |
| 1.802 | 0.000 | 0.654 | 0.654 | 0.000 | 0% | 0% | 100% |
| 3.828 | 0.000 | 0.098 | 0.673 | 0.000 | 100% | 100% | 0% |
| 0.098 | 0.000 | 0.43 | 0.613 | 0.000 | 100% | 0% | 20% |
| 0.054 | 0.000 | 0.985 | 0.912 | 0.000 | 0% | 0% | 20% |
| 0.986 | 0.000 | 0.543 | 0.431 | 0.000 | 100% | 0% | 20% |
| 0.134 | 0.000 | 0.621 | 0.312 | 0.000 | 0% | 0% | 20% |
| 0.431 | 0.000 | 0.342 | 0.412 | 0.000 | 100% | 0% | 100% |
| 0.543 | 0.988 | 0.612 | 0.324 | 0.000 | 0% | 0% | 100% |
| 0.456 | 0.543 | 0.154 | 0.813 | 0.000 | 0% | 100% | 50% |
| 0.123 | 0.903 | 0.041 | 0.512 | 0.001 | 0% | 0% | 100% |
| 0.034 | 0.043 | 0.812 | 0.432 | 0.000 | 0% | 0% | 100% |
| 0.564 | 0.000 | 0.564 | 0.442 | 0.000 | 0% | 0% | 100% |
| 0.564 | 0.094 | 0.513 | 0.221 | 0.000 | 0% | 0% | 0% |
| 0.435 | 0.654 | 0.543 | 0.012 | 0.000 | 0% | 100% | 0% |
| 0.054 | 0.765 | 0.754 | 0.412 | 0.004 | 0% | 0% | 0% |
| 0.453 | 0.432 | 0.342 | 0.812 | 0.000 | 0% | 0% | 0% |
| 0.321 | 0.675 | 0.614 | 0.221 | 0.000 | 0% | 0% | 50% |
| 0.45 | 0.000 | 0.142 | 0.612 | 0.000 | 0% | 100% | 0% |
| 0.876 | 0.875 | 0.531 | 0.342 | 0.083 | 0% | 0% | 50% |
| 0.0004 | 0.000 | 0.0000 | 0 | 0.0000 | 0% | 0% | 0% |
| 12.2064 | 6.5694 | 9.427 | 9.162 | 0.616 |  |  |  |

**Program for pto**

|  |
| --- |
| #CreateTransportPlan |
|  | def transport\_plan(data, dict\_trucks, capacity\_dict): |
|  | # List of Stores per Truck for each DAY |
|  | df\_plan = pd.DataFrame(data.groupby(['Date', 'TruckID'])['Code'].apply(list)) |
|  | df\_plan.columns = ['List\_Code'] |
|  | # List of Box Quantity |
|  | df\_plan['List\_BOX'] = data.groupby(['Date', 'TruckID'])['BOX'].apply(list) |
|  | # Mean of FTL |
|  | df\_plan['FTL'] = data.groupby(['Date', 'TruckID'])['FTL'].mean() |
|  | df\_plan['Capacity(T)'] = df\_plan['FTL'].map(capacity\_dict) |
|  | df\_plan['List\_Loading'] = data.groupby(['Date', 'TruckID'])['Loading(T)'].apply(list) |
|  | df\_plan['Count'] = df\_plan['List\_Loading'].apply(lambda t: len(t)) |
|  | df\_plan['Total\_tons(T)'] = data.groupby(['Date', 'TruckID'])['Loading(T)'].sum() |
|  |  |
|  | # Distribute: one shipment per col |
|  | # Stores |
|  | d = df\_plan['List\_Code'].apply(pd.Series) |
|  | for col in d: |
|  | df\_plan["Store%d" % (col+1)] = d[col] |
|  | # Boxes number |
|  | d = df\_plan['List\_BOX'].apply(pd.Series) |
|  | for col in d: |
|  | df\_plan["Box%d" % (col+1)] = d[col] |
|  | # Shipments Tonnage |
|  | d = df\_plan['List\_Loading'].apply(pd.Series) |
|  | for col in d: |
|  | df\_plan["Tons%d" % (col+1)] = d[col] |
|  |  |
|  | # Fill NaN + Drop useless columns |
|  | df\_plan.fillna(0, inplace = True) |
|  | if 1 == 0: |
|  | df\_plan.drop(['List\_Code'], axis = 1, inplace = True) |
|  | df\_plan.drop(['List\_BOX'], axis = 1, inplace = True) |
|  | df\_plan.drop(['List\_Loading'], axis = 1, inplace = True) |
|  |  |
|  | return df\_plan |

**Html code**

|  |
| --- |
| #CreateTransportPlan |
|  | def transport\_plan(data, dict\_trucks, capacity\_dict): |
|  | # List of Stores per Truck for each DAY |
|  | df\_plan = pd.DataFrame(data.groupby(['Date', 'TruckID'])['Code'].apply(list)) |
|  | df\_plan.columns = ['List\_Code'] |
|  | # List of Box Quantity |
|  | df\_plan['List\_BOX'] = data.groupby(['Date', 'TruckID'])['BOX'].apply(list) |
|  | # Mean of FTL |
|  | df\_plan['FTL'] = data.groupby(['Date', 'TruckID'])['FTL'].mean() |
|  | df\_plan['Capacity(T)'] = df\_plan['FTL'].map(capacity\_dict) |
|  | df\_plan['List\_Loading'] = data.groupby(['Date', 'TruckID'])['Loading(T)'].apply(list) |
|  | df\_plan['Count'] = df\_plan['List\_Loading'].apply(lambda t: len(t)) |
|  | df\_plan['Total\_tons(T)'] = data.groupby(['Date', 'TruckID'])['Loading(T)'].sum() |
|  |  |
|  | # Distribute: one shipment per col |
|  | # Stores |
|  | d = df\_plan['List\_Code'].apply(pd.Series) |
|  | for col in d: |
|  | df\_plan["Store%d" % (col+1)] = d[col] |
|  | # Boxes number |
|  | d = df\_plan['List\_BOX'].apply(pd.Series) |
|  | for col in d: |
|  | df\_plan["Box%d" % (col+1)] = d[col] |
|  | # Shipments Tonnage |
|  | d = df\_plan['List\_Loading'].apply(pd.Series) |
|  | for col in d: |
|  | df\_plan["Tons%d" % (col+1)] = d[col] |
|  |  |
|  | # Fill NaN + Drop useless columns |
|  | df\_plan.fillna(0, inplace = True) |
|  | if 1 == 0: |
|  | df\_plan.drop(['List\_Code'], axis = 1, inplace = True) |
|  | df\_plan.drop(['List\_BOX'], axis = 1, inplace = True) |
|  | df\_plan.drop(['List\_Loading'], axis = 1, inplace = True) |
|  |  |
|  | return df\_plan |

***Explaination for Public Transportation Optimization***

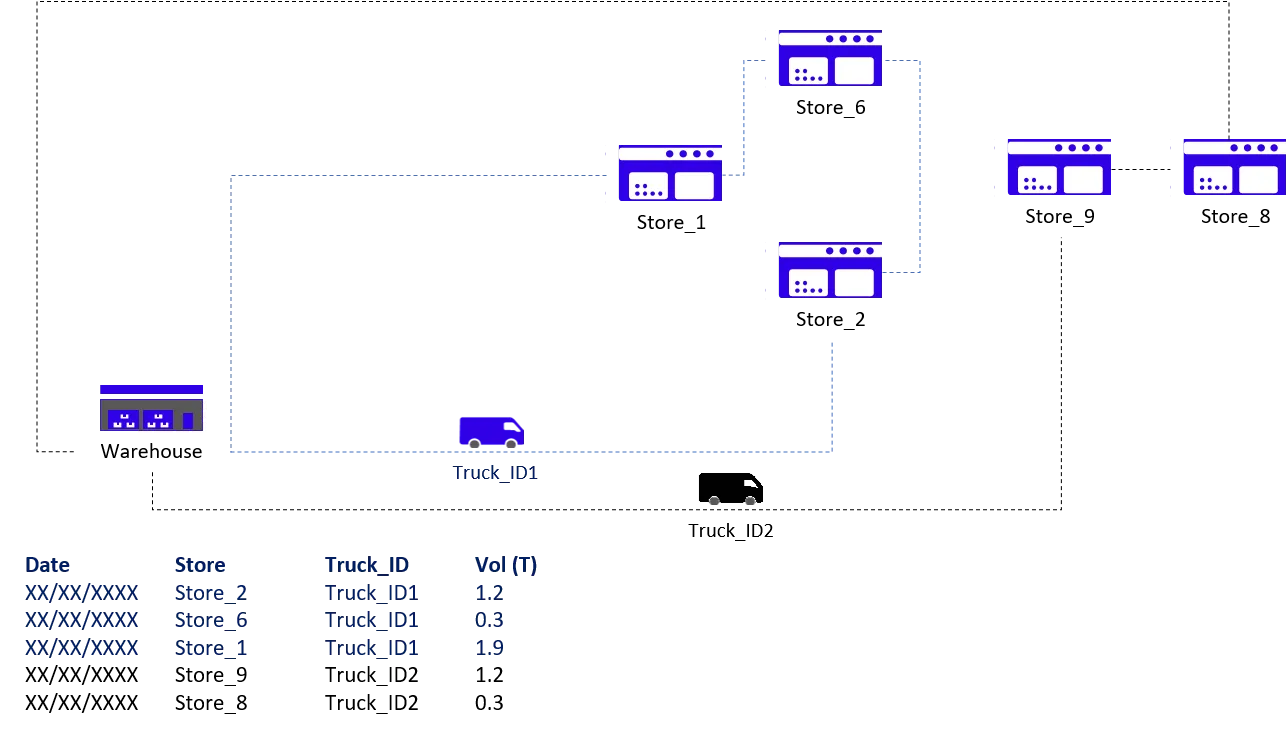


How do you make a transport plan with Python?

**1. Problem Statement**

Retail Stores Distribution with [**Full Truck Load (FTL)**](https://www.youtube.com/watch?v=PYkN24PMKd8)

* **1 Warehouse**delivering stores by using **three** types of Trucks  
  (3.5T, 5T, 8T)
* **49 Stores** delivered
* **12 Months** of Historical Data with **10,000 Deliveries**
* **7 days**a week of Operations
* **23 Cities**
* **84 Trucks** in your fleet



## 2. Objective: Reduce the Cost per Ton

Method: Shipment Consolidation

In this scenario, you are using 3rd party carriers that charge full trucks per destination:



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **s.no** | **City\_En** | **3.5T (Rmb)** | **5T (Rmb)** | **8T (Rmb)** | **3.5T (Rmb/Ton)** | **5T (Rmb/Ton)** | **8T (Rmb/Ton)** |
| **1** | City\_1 | 485 | 650 | 800 | 139 | 130 | 100 |
| **2** | City\_2 | 640 | 700 | 820 | 183 | 140 | 103 |
| 3 | City\_3 | 690 | 780 | 890 | 197 | 156 | 111 |
| 4 | City\_4 | 810 | 1,000 | 1,150 | 231 | 200 | 144 |
| 5 | City\_5 | 1,300 | 1,568 | 1,723 | 371 | 314 | 215 |
| 6 | City\_6 | 1,498 | 1,900 | 2,100 | 428 | 380 | 263 |
| 7 | City\_7 | 980 | 1,250 | 1,450 | 280 | 250 | 181 |
| 8 | City\_8 | 1,350 | 1,450 | 1,500 | 386 | 290 | 188 |
| 9 | City\_9 | 1,350 | 1,450 | 1,500 | 386 | 290 | 188 |
| 10 | City\_10 | 850 | 1,000 | 1,200 | 243 | 200 | 150 |

The table above shows rates applied by carriers for each city delivered for each type of truck. Observing**costs per ton are lower for larger trucks**, one lever of improvement is**maximizing shipments consolidation when building routes**.

Thus, the [**Route Transportation Planning Optimization's**](https://www.youtube.com/watch?v=lhDBTlsGDVc&t=1s) main target will be to cover a maximum number of stores per route.

# II. Data Processing: Understand the Current Situation

## 1. Import Datasets

Before starting to think about the [optimization model](https://www.youtube.com/watch?v=lhDBTlsGDVc&t=1s), your priority is to understand the current situation.

Starting with unstructured data coming from several sources, we’ll need to build a set of data frames to model our network and provide visibility on the loading rate and list of stores delivered for each route.

**Records of Deliveries per Store**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Truck\_ID** | **Store\_ID** | **FTL** | **Order** | **BOX** | **SKU** | **Loading (Tons)** |
| 9/1/2016 | Truck\_ID1 | Store\_ID1 | 3.5 | 16 | 311 | 83 | 2.404 |
| 9/1/2016 | Truck\_ID1 | Store\_ID2 | 3.5 | 18 | 178 | 83 | 1.668 |
| 9/1/2016 | Truck\_ID2 | Store\_ID3 | 3.5 | 10 | 74 | 54 | 0.81 |
| 9/1/2016 | Truck\_ID2 | Store\_ID4 | 3.5 | 19 | 216 | 88 | 2.413 |
| 9/1/2016 | Truck\_ID3 | Store\_ID5 | 3.5 | 10 | 117 | 54 | 1.119 |
| 9/1/2016 | Truck\_ID3 | Store\_ID6 | 3.5 | 15 | 294 | 92 | 2.962 |
| 9/1/2016 | Truck\_ID4 | Store\_ID7 | 3.5 | 5 | 42 | 19 | 0.421 |

### **Store Address**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Code** | **city** | **Long** | **Lat** | **address** | **Code** | **city** | **Long** |
| Store\_ID1 | City\_Store1 | 31.952792 | 118.8192708 | Address\_1 | Store\_ID1 | City\_Store1 | 31.952792 |
| Store\_ID2 | City\_Store2 | 31.952792 | 118.8192718 | Address\_2 | Store\_ID2 | City\_Store2 | 31.952792 |
| Store\_ID3 | City\_Store3 | 31.675948 | 120.7468221 | Address\_3 | Store\_ID3 | City\_Store3 | 31.675948 |
| Store\_ID4 | City\_Store4 | 31.664448 | 120.7700006 | Address\_4 | Store\_ID4 | City\_Store4 | 31.664448 |
| Store\_ID5 | City\_Store5 | 31.750971 | 119.9478857 | Address\_5 | Store\_ID5 | City\_Store5 | 31.750971 |
| Store\_ID6 | City\_Store6 | 31.791351 | 119.9232302 | Address\_6 | Store\_ID6 | City\_Store6 | 31.791351 |
| Store\_ID13 | City\_Store13 | 31.387863 | 121.2797154 | Address\_13 | Store\_ID13 | City\_Store13 | 31.387863 |

### **Transportation Costs**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **s.no** | **City\_En** | **3.5T (Rmb)** | **5T (Rmb)** | **8T (Rmb)** | **3.5T (Rmb/Ton)** | **5T (Rmb/Ton)** |
| **1** | City\_1 | 485 | 650 | 800 | 139 | 130 |
| **2** | City\_2 | 640 | 700 | 820 | 183 | 140 |
| 3 | City\_3 | 690 | 780 | 890 | 197 | 156 |
| 4 | City\_4 | 810 | 1,000 | 1,150 | 231 | 200 |
| 5 | City\_5 | 1,300 | 1,568 | 1,723 | 371 | 314 |
| 6 | City\_6 | 1,498 | 1,900 | 2,100 | 428 | 380 |
| 7 | City\_7 | 980 | 1,250 | 1,450 | 280 | 250 |

## 2. Listing of stores delivered by each route

Let us process the initial data frame to list all stores delivered for each route.

## 1 Route = 1 Truck ID + 1 Date

***IOT DEVICE***



**What is the Internet of Things?**

The Internet of Things, or IoT, refers to the billions of physical devices around the world that are now connected to the internet, all collecting and sharing data. Thanks to the arrival of super-cheap computer chips and the ubiquity of wireless networks, it's possible to turn anything, from something as small as [a pill](https://www.zdnet.com/article/how-sensors-enabled-eli-lilly-to-improve-the-patient-experience/)to something as big as [an aeroplane](https://www.zdnet.com/article/ten-examples-of-iot-and-big-data-working-well-together/), into a part of the IoT. Connecting up all these different objects and adding sensors to them adds a level of digital intelligence to devices that would be otherwise dumb, enabling them to communicate real-time data without involving a human being. The Internet of Things is making the fabric of the world around us more smarter and more responsive, merging the digital and physical universes.

* [**The Internet of Things? It's really a giant robot and we don't know how to fix it**](https://www.zdnet.com/article/the-internet-of-things-its-really-a-giant-robot-and-we-dont-know-how-to-fix-it/)

**What is an example of an Internet of Things device?**

Pretty much any physical object can be transformed into an IoT device if it can be connected to the internet to be controlled or communicate information.

[A lightbulb](https://www.zdnet.com/article/building-my-own-internet-of-things-ambient-experience-one-step-at-a-time/) that can be switched on using a smartphone app is an IoT device, as is a motion sensor or a [smart thermostat](https://www.zdnet.com/article/johnson-controls-cortana-powered-thermostat-is-up-for-preorder-in-march/) in your office or a connected streetlight. An IoT device could be as fluffy as [a child's toy](https://www.zdnet.com/article/fbi-to-parents-beware-your-kids-smart-toy-could-be-a-security-risk/) or as serious as [a driverless truck](https://www.zdnet.com/article/driverless-trucks-are-coming-but-for-now-adoption-is-in-the-slow-lane/). Some larger objects may themselves be filled with many smaller IoT components, such as a jet engine that's now filled with thousands of sensors collecting and transmitting data back to make sure it is operating efficiently. At an even bigger scale, [smart cities projects are filling entire regions with sensors](https://www.zdnet.com/article/las-vegas-announces-smart-city-plans-with-cisco/) to help us understand and control the environment.

**What is the history of the Internet of Things?**

The idea of adding sensors and intelligence to basic objects was discussed throughout the 1980s and 1990s (and there are arguably some [much earlier ancestors](https://innovateuk.blog.gov.uk/2017/07/03/the-history-of-internet-of-things-iot/)), but apart from some early projects -- including an internet-connected vending machine -- progress was slow simply because the technology wasn't ready. Chips were too big and bulky and there was no way for objects to communicate effectively.

Processors that were cheap and power-frugal enough to be all but disposable were needed before it finally became cost-effective to connect up billions of devices. The [adoption of RFID tags](https://www.zdnet.com/article/rfid-heralds-the-internet-of-things/) -- low-power chips that can communicate wirelessly -- solved some of this issue, along with the increasing availability of broadband internet and cellular and wireless networking. The [adoption of IPv6](https://www.zdnet.com/article/finally-ipv6s-killer-app-the-internet-of-things/)-- which, among other things, should provide enough IP addresses for every device the world (or indeed this galaxy) is ever likely to need -- was also a necessary step for the IoT to scale