Georgia Tech College of Engineering
H. Milton Stewart School of Industrial and Systems Engineering

ISyE 6202 Supply Chain Facilities

Casework 2.1

HelpBots Supply Chain Facility Network Analysis, Planning, and Simulation

Guillaume Eymery - Louise Lacroix - Braden McCay Krohn

H. Milton Stewart School of Industrial and Systems Engineering

Contents

List of Figures

List of Tables

TASK 1

For this Task, we will suppose no demand seasonality as well as no demand uncertainty. The global market demand last year was 2,000,000 products, with the overall market in the USA growing about 7.5 % for the next year. For now, we sill not take into account the conservative and optimistic scenarios, that will be studied in Task 2. HelpBot has an actual 1.5 % market share.
We will write a code that for the 1-year planning horizon, takes into parameter the following aggregation threads and their combinations:

Temporal

1. Yearly
   No specific computation is expected here because the demand is already expressed yearly.

2. Monthly
   The idea is to divide the yearly demand by 12 and spread it equally in the 12 years.

3. Daily
   Same, than months but this time, the annual global demand is divided by 365.

Spatial

1. USA Territory
   No specific computation is expected, because the Global demand is already expressed nationally.

2. Market Type
   For the Market Type as well as State and 3 digit ZIP, we will be using the ZIP3-Market CSV file.

3. State

4. 3-digit ZIP

Measure

1. Dollars ($)
   For the market, let us assume that HelpBots average price for each robot is the

same as the competitors: 200 $.

2. Units (#)
   The strategy is to use the PMF associated to each ZIP code to compute the number of units ordered by each ZIP zone. For this we slightly modified the ZIP3 Market CSV file to take into consideration the associated PMF (in HelpBot ZIP3 Market PMF CSV file and by using the Excel VLOOKUP function) for each ZIP.

3. Volume (V)
   For the market, let us assume that HelpBots average volume for each robot is the same as the competitors.

4. Weight (W)
   For the market, let us assume that HelpBots average weight for each robot is the same as the competitors.

The planning horizon consists of various combinations of the threads listed above. Each of the combinations can be formulated as:

$$\left\{\begin{array}{ll}\text{Temporal:} & \text{Yearly, Monthly, Daily} \\ \text{Spatial:} & \text{USA Territory, Market Type, State, 3-digit ZIP} \\ \text{Measure:} & \text{Dollars (\$), Units (\#), Volume (V), Weight (W)}\end{array}\right\}$$

Thus, the combinations of aggregation threads for the 1-year planning horizon can be represented as:

$$(\text{Temporal} \times \text{Spatial} \times \text{Measure}) = 3 \times 4 \times 4 = 48 \text{ possible combinations.}$$

Now that the parameters for the function are set, we must define compute the amount of dollars ($), Volume (V) and Weight (W) that the market expects for the next year. We may then compute the expected Volume, Weights and Price: The following operations are performed on the columns of the DataFrame `table_main`:

$$\texttt{table\_main['Price']} = \texttt{table\_main['Global Demand']} \times 200$$
$$\texttt{table\_main['Volume']} = \texttt{table\_main['Global Demand']} \times 1.5 \times 2 \times 2$$
$$\texttt{table\_main['Weight']} = \texttt{table\_main['Global Demand']} \times 60$$

Where the global demand represents the yearly number of units that are to be sold the following year.

Now, we can write the Python function associated with Task 1. The code creates the expected Pivot Tables, function of the user's input.

The function is named: `create_pivot(number, letter, measure, table_main):`.

For example: `create_pivot(2, 'B', '#', table_main)`, returns the Monthly Pivot Table by Market.

Market Period	Primary	Secondary	Tertiary
Jan	133671	52451	2509
Feb	133671	52451	2509
Mar	133671	52451	2509
Apr	133671	52451	2509
May	133671	52451	2509
Jun	133671	52451	2509
Jul	133671	52451	2509
Aug	133671	52451	2509
Sep	133671	52451	2509
Oct	133671	52451	2509
Nov	133671	52451	2509
Dec	133671	52451	2509

Table 1: Market data for each period

The demand is the same each months, which is logical, since we are considering no variability, for now.

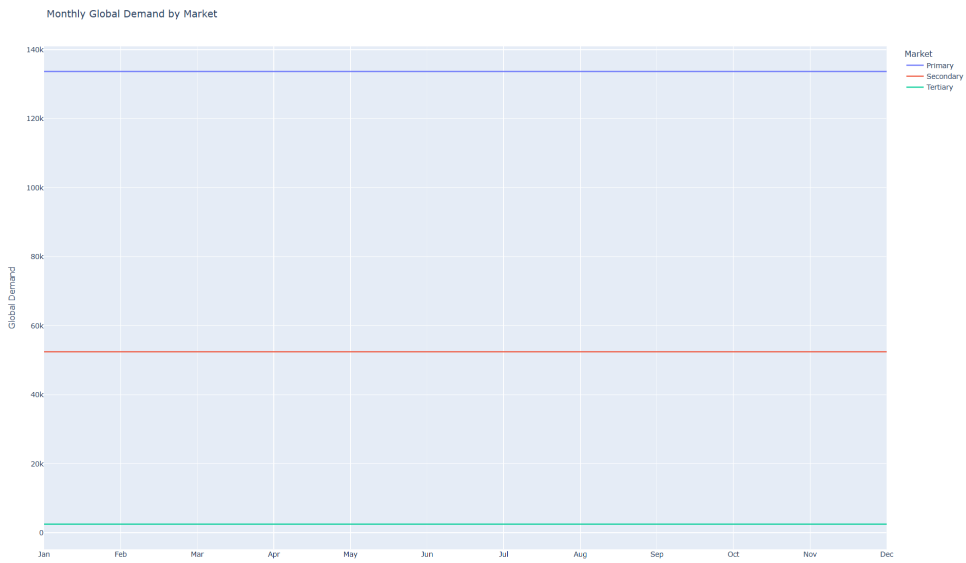

Figure 1: Global Demand, Monthly and per Market

Similarly, we can compute the annual market revenue per State (`create_pivot(1, 'C', '#', table_main)`).

State		AK	AL	...	WI	WV	WY
Period	Yearly	65000.0	3759600.0	...	5701400.0	2390600.0	729000.0

This representation can prove to be very useful if we want to visualise the dynamics in the home robot industry in the United States.

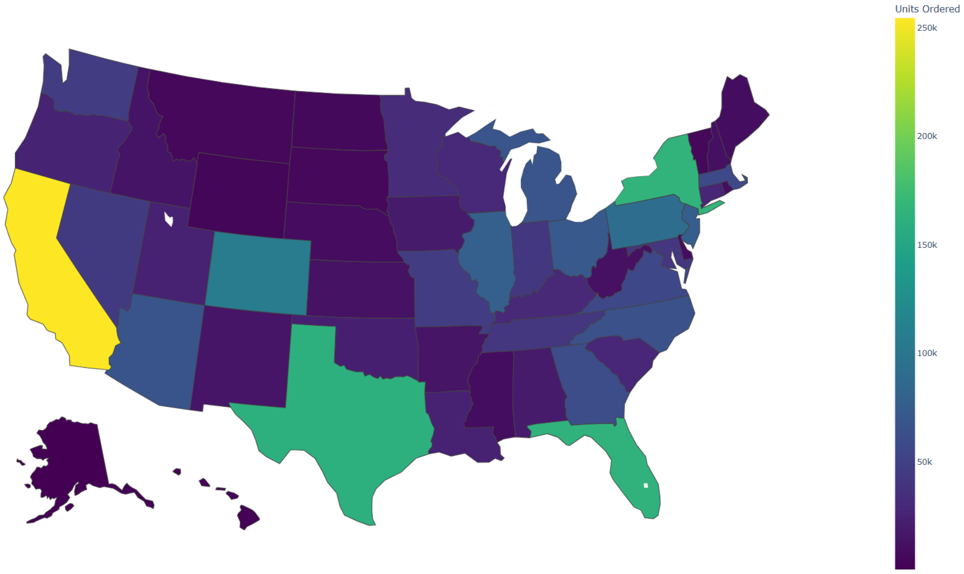

Figure 2: Heatmap of # of Units bought in a year, per State

TASK 2

Firstly, we will be looking at the demand seasonality, which can be found in the HelpBot seasonality CSV file. After that, by using the same Python function as previously - with small modifications- we can plot any kind of metric per area as wanted.
For example, by using the function `create_pivot1(3, 'B', '#', table_main).set_index ('Period')`, and for each day multiply the annual demand by:

$$Day_i = AnnualDemand \times WeeklySeasonality \times DailySeasonality$$

We obtain the following graph:

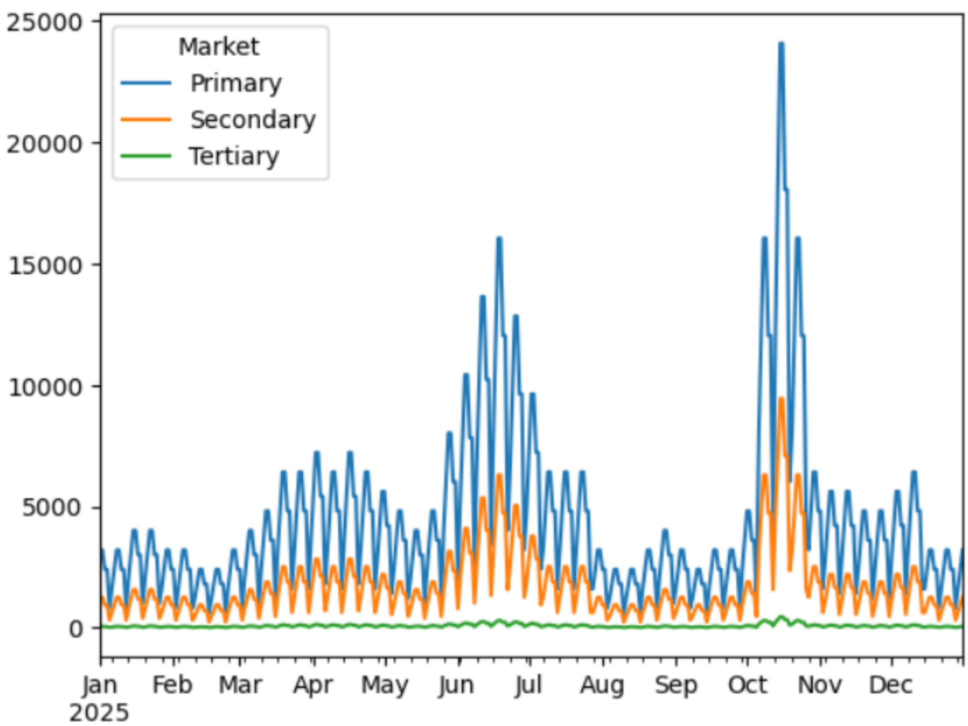

Figure 3: Global demand, per market, including seasonality

Let us note that we did not include uncertainty yet. We will start by focusing on the Global Market, and then study HelpBot more in detail.

We will be realizing a Monte Carlo Simulation to compute the demand for for HelpBots in 2025. This suggest that we will run multiple simulations for year 2025 with randomly selected parameters. These parameters represent the variations between:

1. X The uncertainty relative to the optimistic and conservative evolution of the mar-

ket demand in 2025. We will suppose:
$X \sim N(0.075, \sigma_1^2)$, with σ_1 the volatility.

2. YThe uncertainty relative to the optimistic and conservative evolution of the market share of HelpBots in 2025. We will suppose:
$Y \sim N(0.2, \sigma_2^2)$, with σ_2 the volatility.

3. Z_1The uncertainty due to the weekly demand in 2025 in 2025. We will suppose:
$Z_1 \sim N(0, \sigma_3^2)$

4. Z_2The uncertainty due to the daily demand in 2025 in 2025. We will suppose:
$Z_2 \sim N(0, \sigma_4^2)$

If a general method based on a multiple of the standard deviation $k\sigma$ is desired, the following formula can be applied:

$$\sigma = \frac{\text{Max} - \text{Min}}{2k}$$

where k is the multiple of the standard deviation σ that corresponds to the percentage of the distribution assumed to be covered by the interval $[\text{Min}, \text{Max}]$.
In our case: $k = 1, 2, 2.5$ Signifying that: 68%, 95% and 98% of the distribution is included in the interval.
With this method, we can determine $\sigma_1, \sigma_2, \sigma_3, \sigma_4,$. We than have all the necessary information to compute all the Monte Carlo Simulations.
We will do the following computations for each simulation.
Let:

- $D_{y-1,k}$: The demand for the year 2024.
- $x_{y,k}$: The market evolution in 2025, modeled as $x_{y,k} \sim \mathcal{N}(0.075, \sigma_1^2)$, where x_y is normally distributed with a mean of 0.075 and variance σ_1^2.
- $y_{y,k}$: The evolution of the market share for HelpBots in 2025, modeled as $y_y \sim \mathcal{N}(0.2, \sigma_2^2)$, where y_y is normally distributed with a mean of 0.2 and variance σ_2^2.
- $ms_{y-1,k}$: Market share last year (2024)
- $z1_{y,w,k}$: The annual demand seasonality, modeled as $z1_{y,w} \sim \mathcal{N}(0.2, \sigma_3^2)$, where $w \in [1, 52]$ represents the weeks of the year.
- $z2_{y,w,d,k}$: The weekly demand seasonality, modeled as $z2_{y,w,d} \sim \mathcal{N}(0.15, \sigma_4^2)$, where $d \in [0, 6]$ represents the days of the week.

Therefore, each day will have a demand of

$$d_{y,m,d,k}$$

:

$$\boxed{d_{y,m,d,k} = D_{y-1,k} \times ms_{y-1,k} \times (1 + x_{y,k}) \times (1 + y_{y,k}) \times z1_{y,w,k} \times z2_{y,w,d,k}}$$

Once we computed the daily demand of units for HelpBots, we can easily determine the total volume(V), price ($) and weight (W). Using the same code as in Task 1, we can also compute the simulation for each state, ZIP or Market. The following simulations results where computed considering the entire United States, but the reader can change this feature directly in the code.

When ran for 30 iterations per confidence interval $k\sigma$ (therefore 150 total iterations), we can observe the following graph:

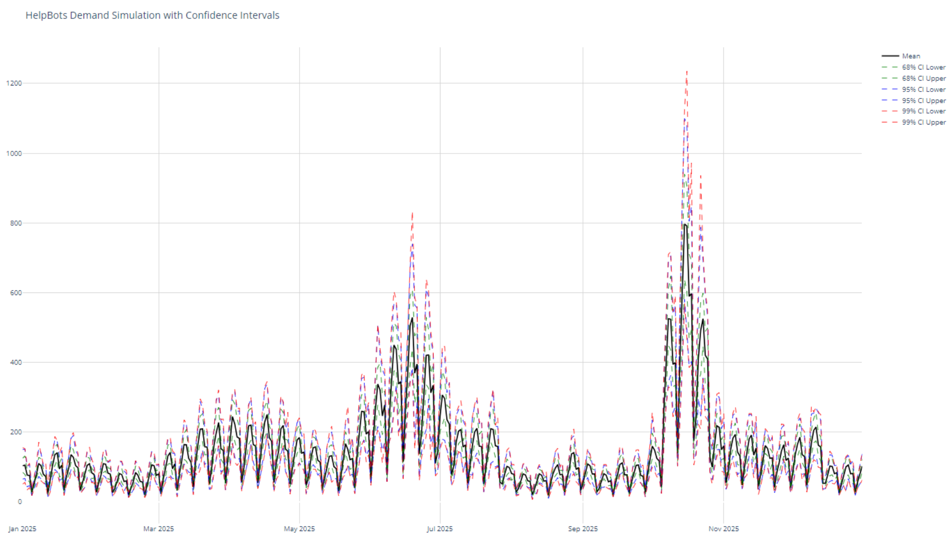

Figure 4: Monte Carlo Simulation for different confidence intervals, US, Number of Units #

In Green: Monte Carlo simulation for an interval of $+ - 1 \times \sigma$ In Blue: Monte Carlo simulation for an interval of $+ - 2 \times \sigma$ In Red: Monte Carlo simulation for an interval of $+ - 2.5 \times \sigma$ We can directly observe that for high demand days, the incertitude is very important. However, the values are not as spread out as expected. This is due to the fact that HelpBots Only controls a small part of the market.

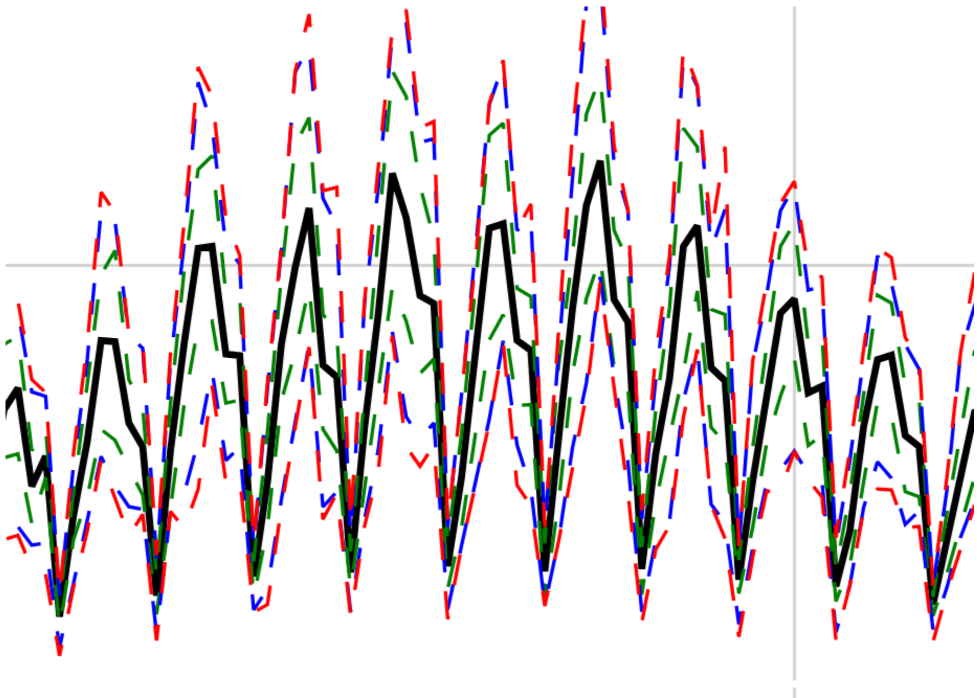

Figure 5: Zoom: Monte Carlo Simulation for different confidence intervals, US, Number of Units #

This Zoom enables US to better understand what is happening each day during the simulation. We can see that the demand regroups periodically at the same value. This is because one day of the week has a very low demand, therefore the dispersion around this value is small.

In the three next plots, we computed the results for a Monte Carlo Simulation:

- Under a triangular distribution
- With a National Growth estimated at 20% (min 15%, max 25%)
- By looking at the different Markets

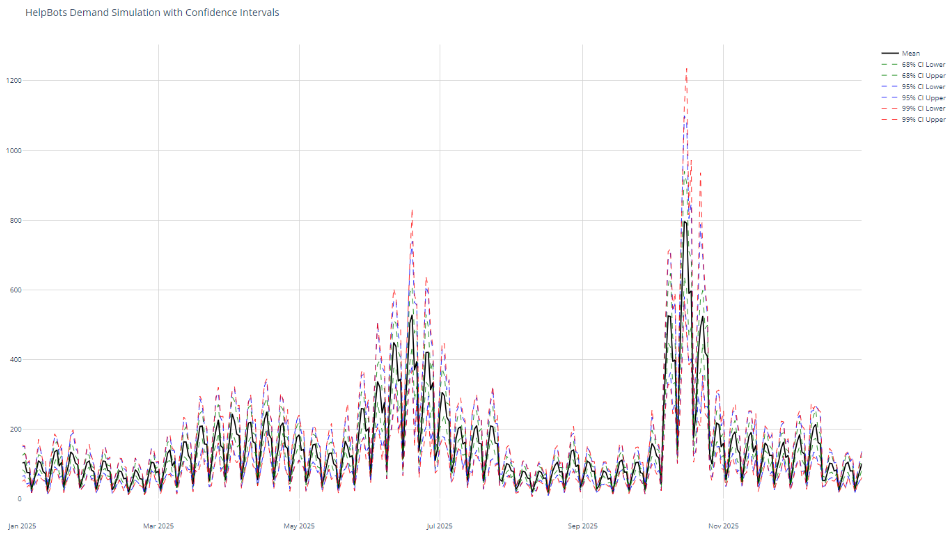

Figure 6: Monte Carlo Simulation for different market evolution scenarios, US, Number of Units #

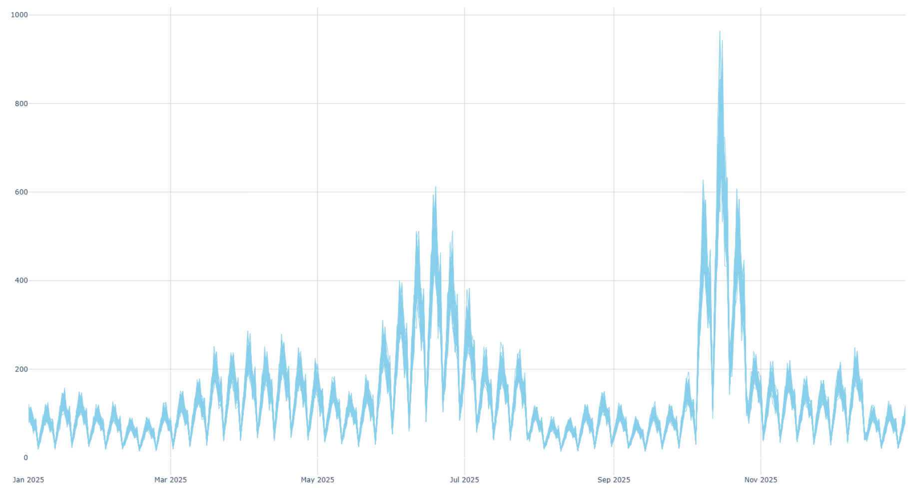

Figure 7: Monte Carlo Simulation for a Triangular Distribution, US, Number of Units #

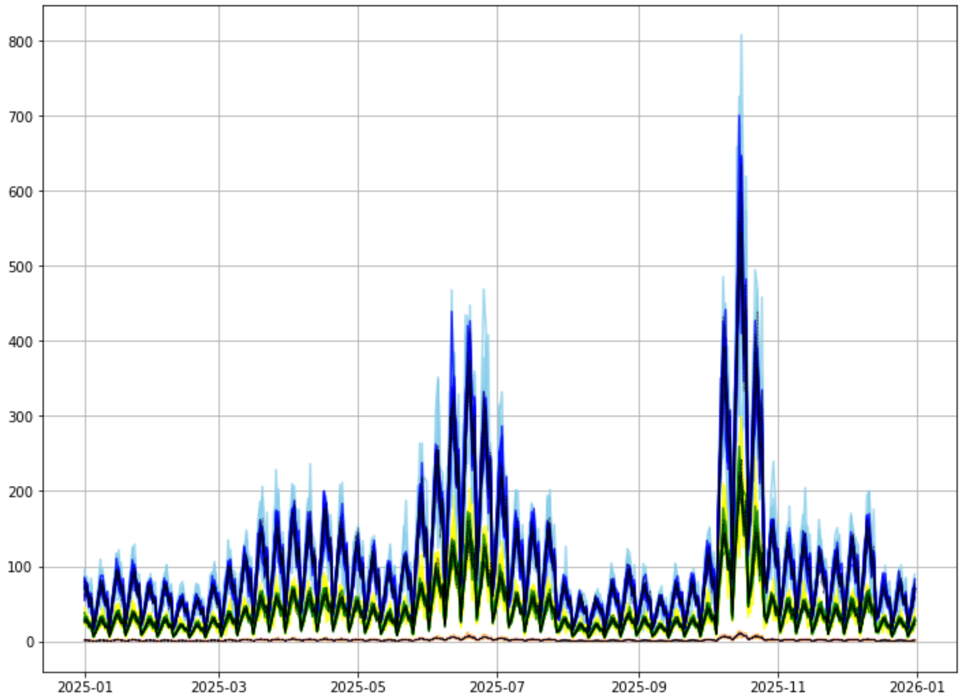

Figure 8: Monte Carlo Simulation for different Markets, Number of Units #

- in blue (sky blue): Primary Market
- in green (yellow): Secondary Market
- in pink (purple): Tertiary Market

TASK 3

Geographical plot

For this task, we will match every ZIP code to the closest fulfillment center (FC). We use the spreadsheet *HelpBots fc_zip3_distance* to do so. We will use Python to find, for each ZIP code, the closest FC. We will go through every line and assign every ZIP code to the closest FC, minimizing the distance to this FC. Then, we will group all ZIP codes that rely on the same FC and plot them on a US map using the same color. To achieve this, we shall use the dynamic Python library `plotly`, which has a US map feature.

However, we need to provide `plotly` with the coordinates for each ZIP code. To do this, we have used a government-issued spreadsheet that provides the latitude and longitude of all ZIP codes in the US. This spreadsheet had more ZIP codes than we need, using 5-digit ZIP codes. We will create a new spreadsheet from it, grouping all 5-digit ZIP codes into 3-digit ZIP codes. As a result, 01001, 01002, 01003, etc., will be combined under the ZIP code 010. Since all of these 5-digit ZIP codes are geographically close, the latitude and longitude for 010 will be the average of all the 010XX ZIP codes. We will follow this procedure for all ZIP codes and obtain the spreadsheet *US_Zip*, which we will use to plot the dots for each ZIP code related to a specific FC.

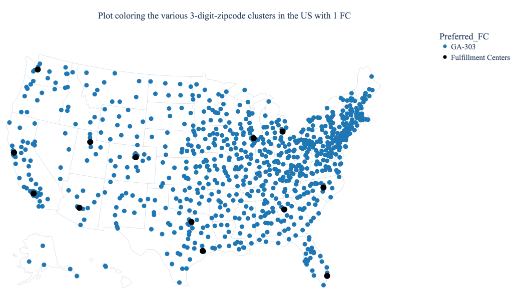

Figure 9: Plot coloring the various 3-digit-zipcode clusters in the US with 1 FC

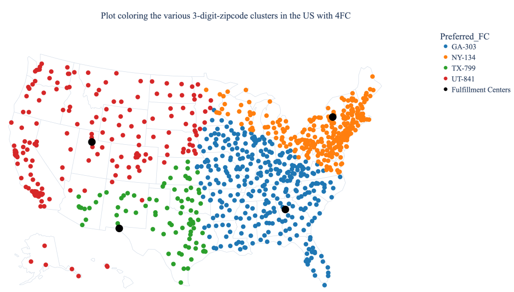

Figure 10: Plot coloring the various 3-digit-zipcode clusters in the US with 4 FC

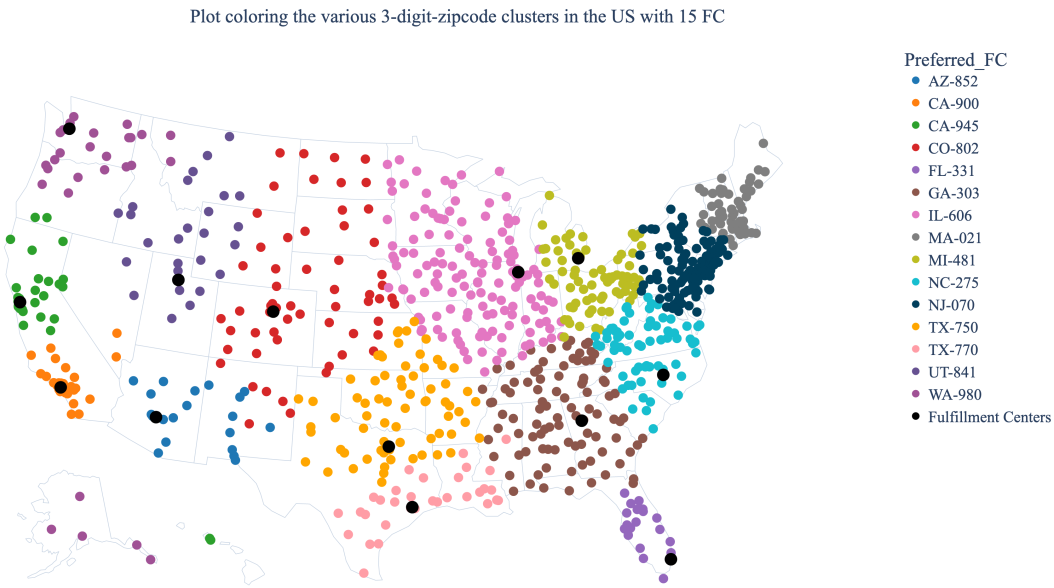

Figure 11: Plot coloring the various 3-digit-zipcode clusters in the US with 15 FC

We notice that FC are not equally spread accross the country. We have more FC on the East Cost, which makes sense since we have more ZIP codes to ship over here. This will help HelpBot to meet the demand as fast as possible over there. Other FC are shipping way wider areas as the Colorado FC that delivers up to the North of the country. But since the ZIP codes are more spread in the region, which means the density is lower, we may not need more FC here.

In a 4FC configuration as well as in a 15FC, we have indeed FC that ship to way larger areas than other. The is the case for the Utah one in the 4FC that ship to an area three times bigger than the one from New York.

Aggregated demand share

Aggregated demand share for 1FC network

In a 1-FC Network, the demand will be fulfilled completely by the single FC of the network. So plotting graphs with a 100% bars did not seem relevant here but we will keep in mind that all ZIP would be served by a unique FC, the one of Georgia, GA-303.

Aggregated demand share for 4FC network

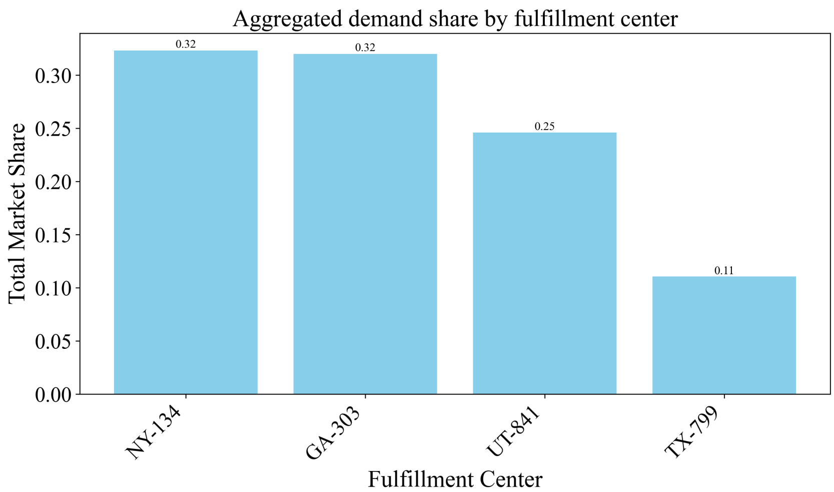

Figure 12: Aggregated demand share by fulfillment center in a 4FC Network

As in 15 FC, networks, we notice that it is not the FC with the biggest shipping area that handle the biggest proportion of demand. Indeed, figure 12 highlights that NY-134 handles more than 30% of the overall market while it only covers a seventh of the country surface. The Utah center only delivers to 11% of the demand while it covers about half the country.

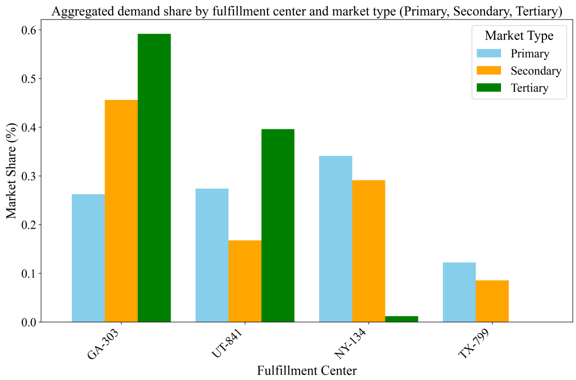

Figure 13: Aggregated demand share by fulfillment center and market type in a 4FC Network

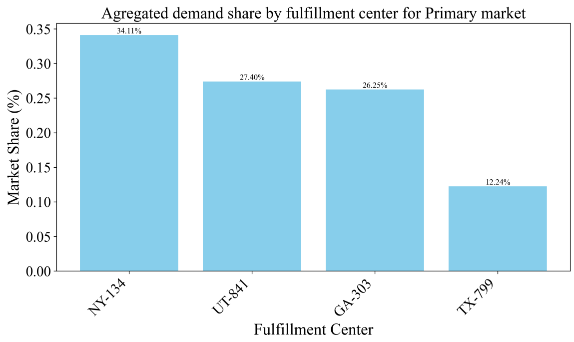

Figure 14: Aggregated demand share by fulfillment center for Primary market in a 4FC Network

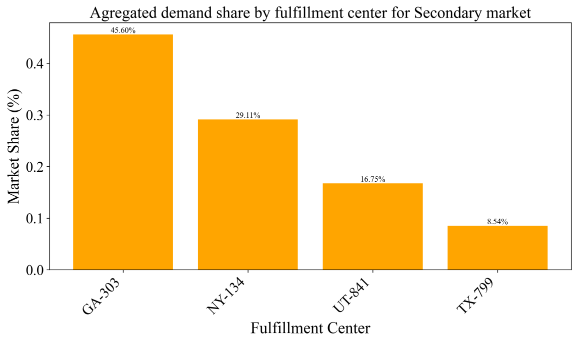

Figure 15: Aggregated demand share by fulfillment center for Secondary market in a 4FC Network

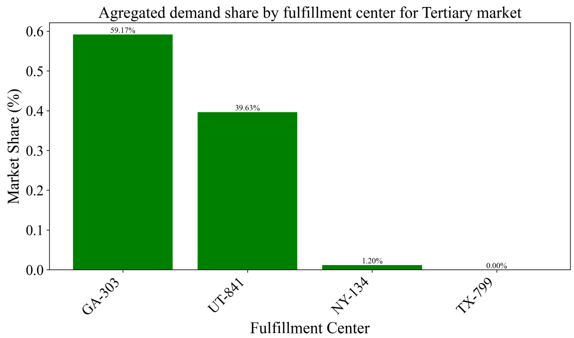

Figure 16: Aggregated demand share by fulfillment center for Tertiary market in a 4FC Network

We note the same trend as in 15FC networks with all types of market being handled by various FC. Nevertheless, one may notice that Georgia, a candidate for a single-FC network handles 45% of the Secondary and 59% of the Tertiary. Texas doesn't meet a significant part of the demand in neither market, which is consistent with the share it hold in figure 12.

Aggregated demand share for 15 FC network

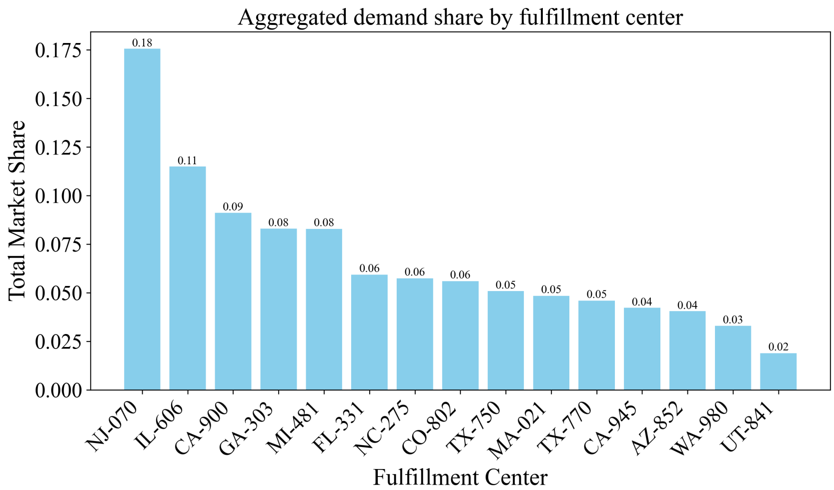

Figure 17: Aggregated demand share by fulfillment center in a 15FC Network

In a 15 FC network, the NJ-070 FC deals with 18% of the demand even if the area it ships is among the smallest in the country. On the other hand, CO-802 that covers a shipping area more than 5 times bigger deals with only 6% of the national demand. TX-770 deals with 5% of the demand and is actually close to another FC, TX-750. Maybe it is interesting to think about merging those two.

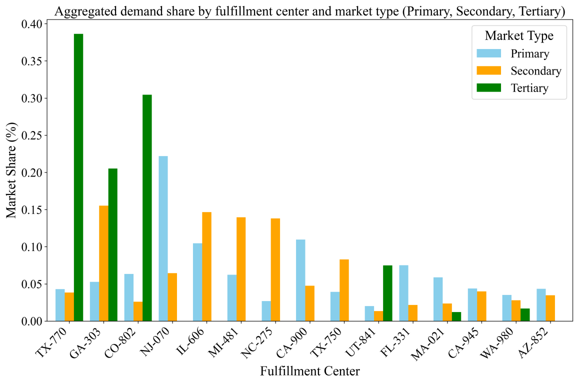

Figure 18: Aggregated demand share by fulfillment center and market type in a 15FC Network

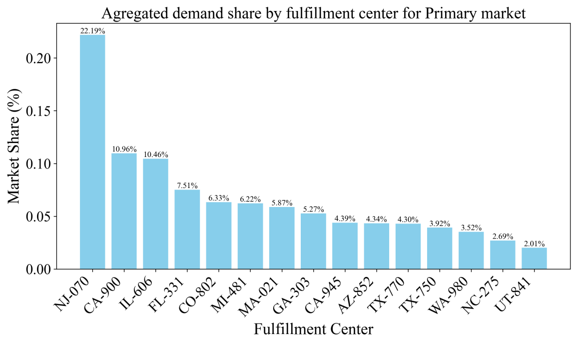

Figure 19: Aggregated demand share by fulfillment center for Primary market in a 15FC Network

In Figure 18, we have plotted the demand share for Primary, Secondary and Tertiary markets over the 15 FC. We notice that different FC deals with different types of market and that Tertiary is handled mostly by fex FC. Let's look more precisely at each market.

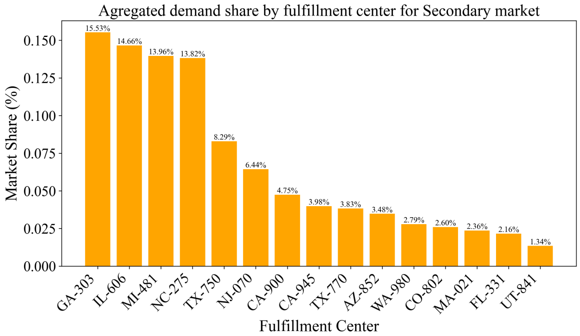

Figure 20: Aggregated demand share by fulfillment center for Secondary market in a 15FC Network

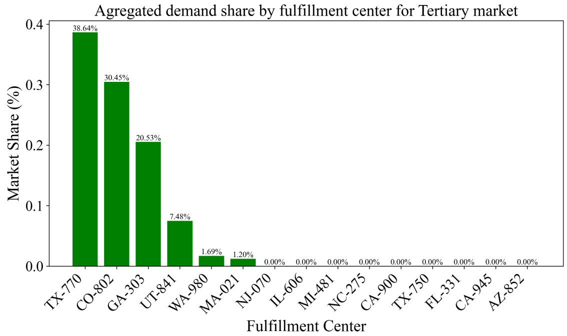

Figure 21: Aggregated demand share by fulfillment center for Tertiary market in a 15FC Network

In figure 19, we notice that NJ-070 deals with 22% of the primary market while UT-841 only 2%. This trend follows the same as in figure 17. In figure 20, we notice that more than 50% of the market is being handled by four FC (GA-303, IL-606, MI-481, NC-275). Figure 21 enhances that only 6 FC manage the Tertiary market, with more than 75% due to 3 FC actually. It is interesting to see that TX-770 that only handles 5% of the total market actually handles 38% of the Tertiary.

Demand distribution

Demand distribution in a 1FC Network

In a 1FC Network one would expect the average distance to travel from the FC to a ZIP bigger than in a multiple FC Network. This is indeed what the following graphs enhance.

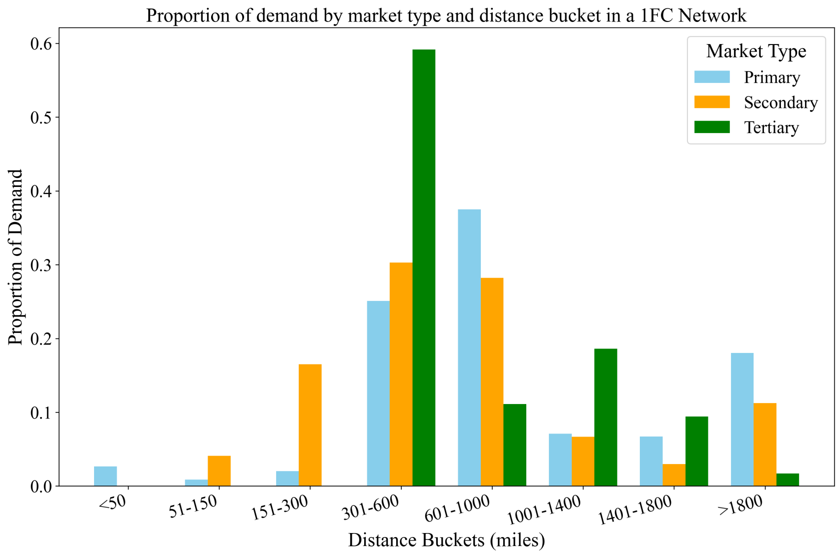

Figure 22: Proportion of demand by market and distance bucket in a 1-FC Network

We note that no less than 0.5% of the demand can be fulfilled under 50 miles and this is just fo rthe Primary market. Most of the products have to travel 301 to 1,000 miles to reach their destination in all three markets. We observe as well that about 20% of the products have to travel above a thousand miles to reach their destination. This makes sense as the center is not located in the middle of a country that remains 2,800 miles wide. We will have to evaluate the costs of all this later on to be able to compare efficiently this option with the other two.

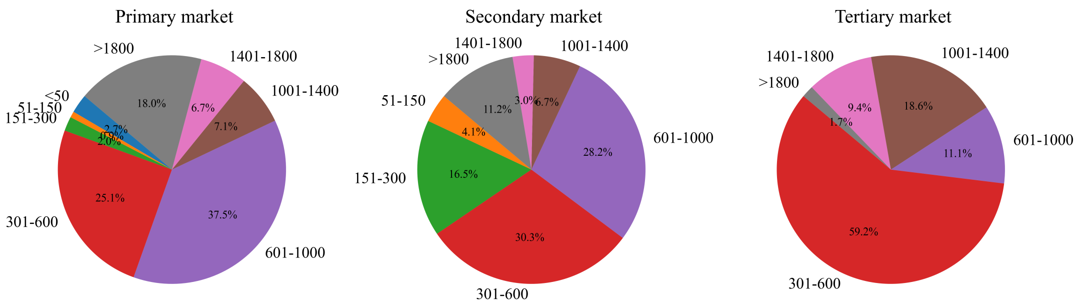

Figure 23: Proportion of demand by market and distance bucket in a 1-FC Network - Pie chart

In the Primary market, we notice that there is very little demand (5.6%) to fulfill in less than 300 miles. The majority (62%) of the demand is within the range [300 - 1000] miles. The Secondary market has a bigger proportion of demand (20%) that can be fulfilled under 300 miles. There is still more than half of the demand that can be fulfilled within the range [300 - 1000] miles. For the Tertiary market, there is no demand to meet under 300 miles. This market goes further since almost 30% of the demand is above 1000 miles.

Demand distribution in a 4FC Network

In a 4-FC network, we expect the distances to travel to be greater than in a 15-FC one. Indeed according to figure 24, there is very little demand to meet under 50 miles with most of the demand being meet-able between 301 to 600 miles. This would lead to bigger costs associated with travel but we would probably save on having fewer FC. We will need to look at this later on. One relevant point still is that we never have to travel more than a 1000 miles to meet the demand, as in a 15-FC Network.

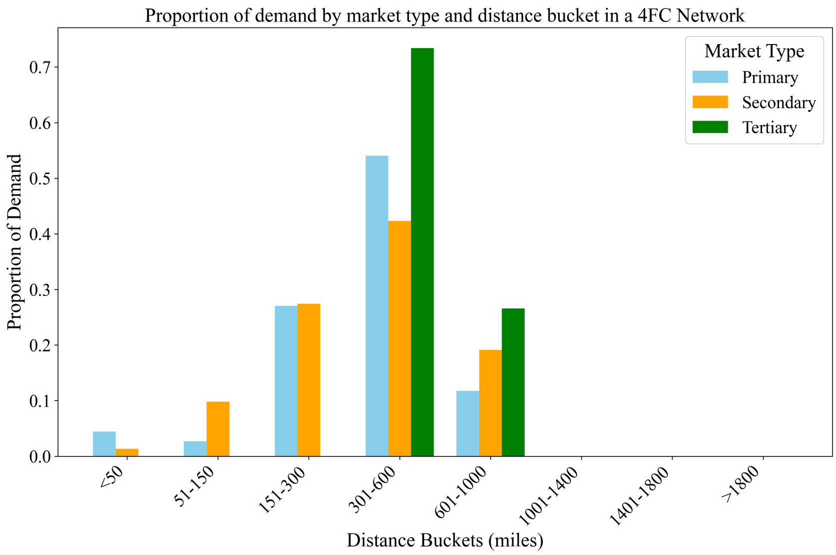

Figure 24: Proportion of demand by market and distance bucket in a 4-FC Network

Figure 25 provides a pie chart of the distance for each market.

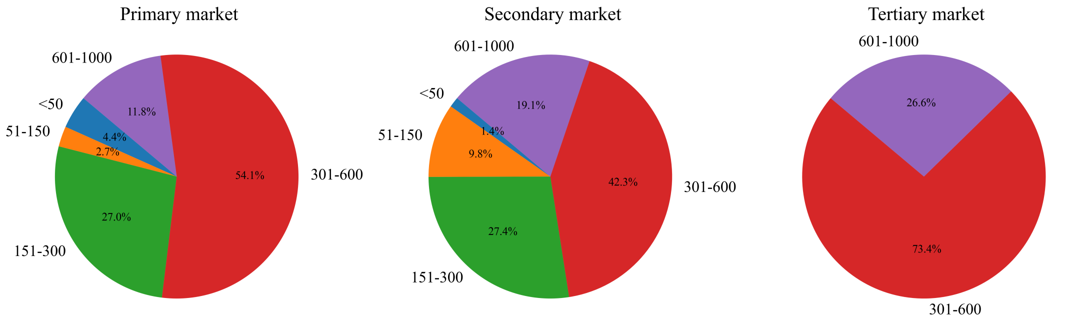

Figure 25: Proportion of demand by market and distance bucket in a 4-FC Network - Pie chart

The previous pie chart confirms that most demand (more than 50%) is to be met between the range 301-600 miles. We even notice than for Primary and Secondary markets about $\frac{3}{4}$ of the demand can be met below 600 miles. This numbers show, as expected, that we will need to travel longer distances to meet demand in this scenario because we could meet 100% of the demand below 600 miles for Primary and Secondary.

Demand distribution in a 15FC Network

In this part, we are plotting the proportion of demand of each demand depending on the distance bucket. Figure 26 provides the bar chart of the result of computation in a 15-FC Network. As a result we notice that there is no demand to meet more that 1000 miles away. On the spreadsheet, we see some of them being 2000 miles aways but since there is no PMF associated with such ZIP codes, we consider that there is no demand there. This means that the network is already pretty efficient as travel distances are not above 1000 miles.

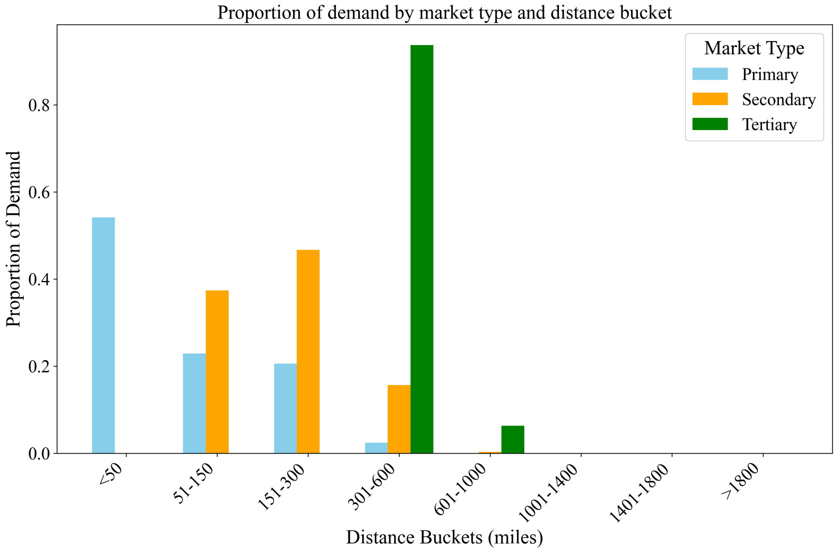

Figure 26: Proportion of demand by market and distance bucket in a 15-FC Network

Figure 27 provides a pie chart of the distance for each market.

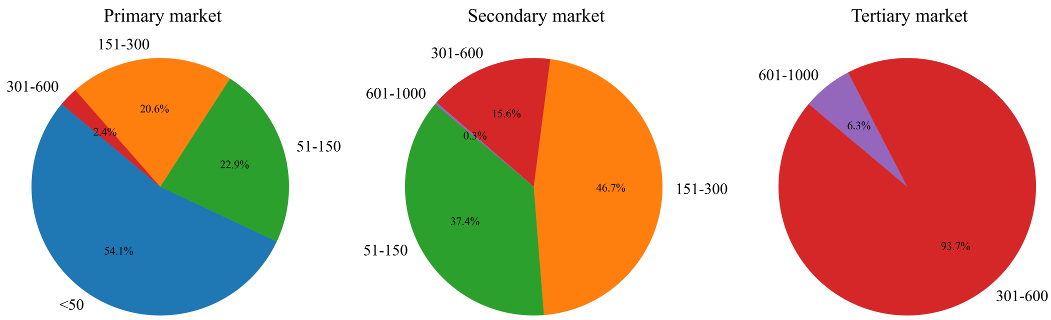

Figure 27: Proportion of demand by market and distance bucket in a 15-FC Network - Pie chart

The primary market is the most optimized since more than 50% of the demand can be fulfilled in less than 50 miles and in the worst case in 600 miles. Secondary market remains optimized as more than 80% of the demand can be fulfilled in less than 300 miles with 37% under 150. The tertiary needs more travel since there is not demand to meet in less than 301 miles, but still, 93% of the demand remains below 600 miles travel distance.

TASK 4

Geographical plot coloring the fulfillment clusters

Geographical plot coloring the fulfillment clusters in a 4FC Network

For this part, we will create fulfillment clusters that can deliver a ZIP code. To do so we will group FC that can serve the ZIP and are in the same distance bucket as the preferred FC or in the next higher one. To be sure we only treat the ZIP codes associated with a PMF we will use a new csv file named *case2_task4_4FC* in which we have removed the ZIP associated with no PMF and we will assign the correct PMF to each ZIP code. The distance buckets we use are (<50, 51-150, 151-300, 301-600, 601-1000, 1001-1400, 1401-1800, >1800). For instance, if a ZIP has a preferred FC being in the distance bucket 51-150, we will look at all the other FC that are in the same distance bucket and in the 151-300 one and they will form a FC cluster. We will group all the ZIP that belong to a same cluster and color them on the map. We will count the number of clusters, making sure that the order of the FC in the cluster is not considered, that is to say that (GA-303, UT-841, NY-134, TX-799) and (NY-134, GA-303, UT-841, TX-799) are two identical clusters. We count **11** different fulfillment clusters. They are plotted in various colors in figure 28.

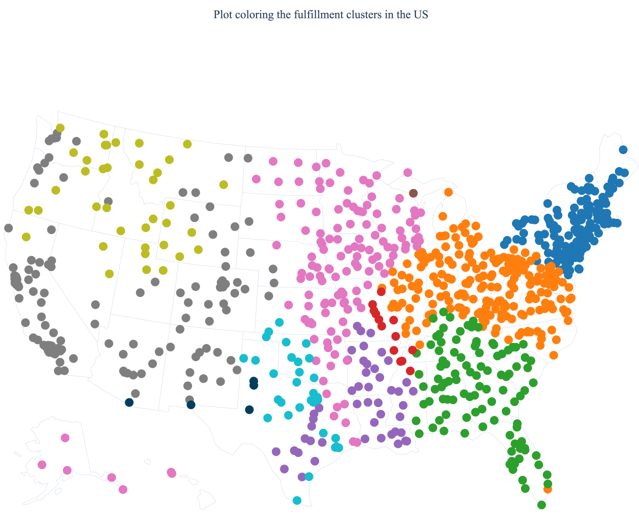

Figure 28: Geographical plot coloring the generated fulfillment clusters in a 4FC Network

Geographical plot coloring the fulfillment clusters in a 15FC Network

As in the 4FC Network, to be sure we only treat the ZIP codes associated with a PMF we will use a new csv file named *case_task4* in which we have removed the ZIP associated with no PMF and we will assign the correct PMF to each ZIP code. We count **99** different fulfillment clusters. They are plotted in various colors in figure 29. We note that we have 9 times more clusters in this situation than in a 4 FC.

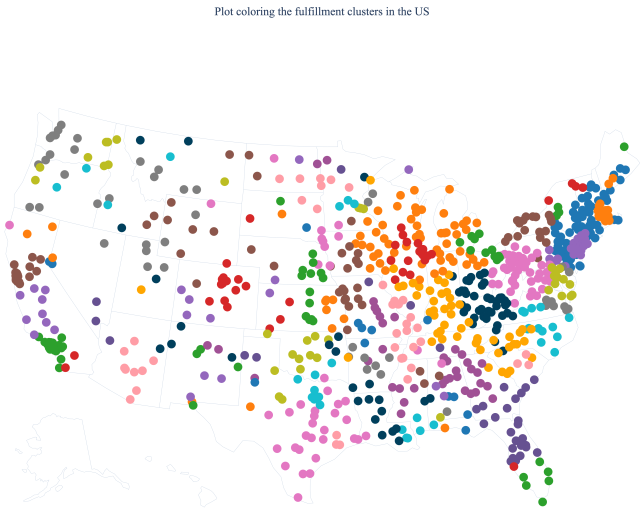

Figure 29: Geographical plot coloring the generated fulfillment clusters in a 15FC Network

Geographical plot coloring the various clusters from green to red

Geographical plot coloring the various clusters from green to red in a 4FC Network

We want to determine how accessible each ZIP code is, that is to say, compute the number of FC that can serve a single ZIP. We are trying to identify the most flexible clusters that are the ones with the most FC. We will plot in green the clusters that count the most FC (12 for the maximum) and shade the map down to red for the clusters that are made of one FC. This means that for this last one, a ZIP code relying on this cluster can only be served by one single FC. In order to do so, we will count on Python the numbers of FC in each cluster and then order them. We will then plot the ZIP in the proper color.

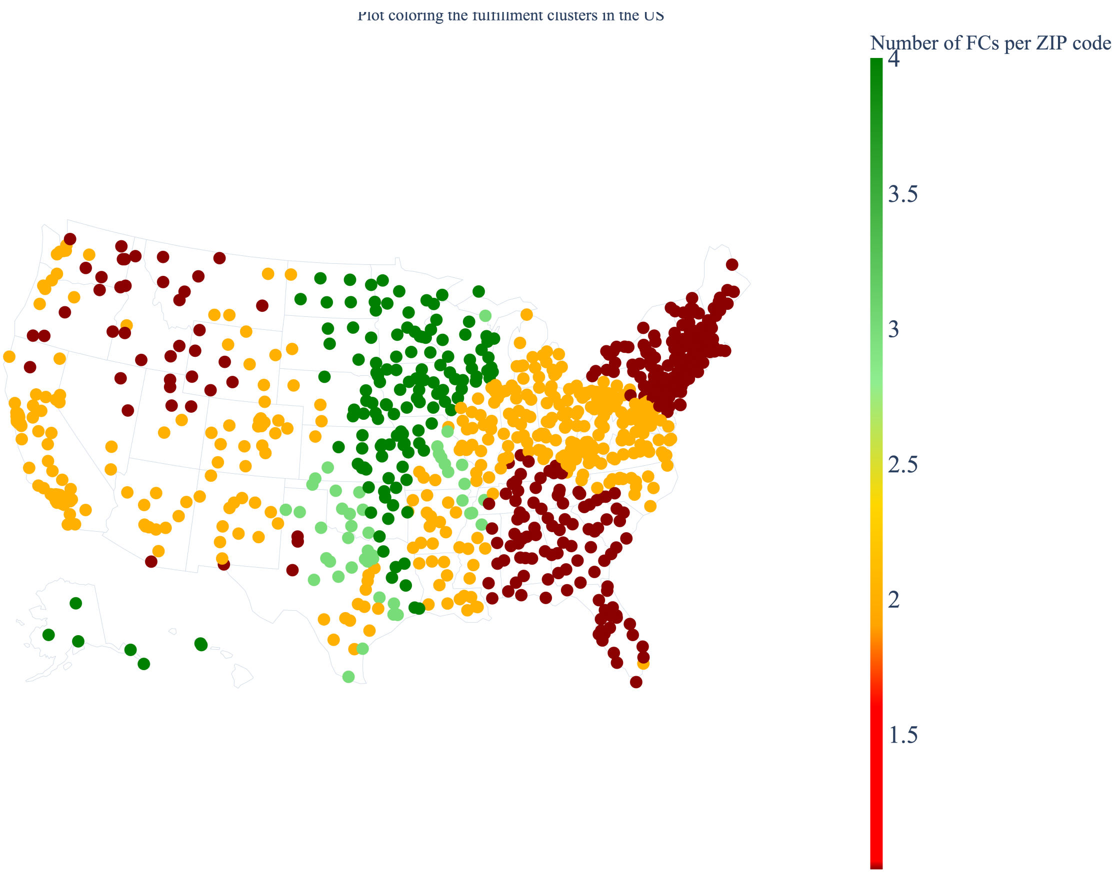

Figure 30: Geographical plot coloring the various clusters based on the number of FCs that can serve the cluster, in a red (lowest) – yellow – green (highest) color scheme in a 4FC Network

The Central U.S. shows a high density of ZIP codes that are served by multiple FCs, indicated by the deep green color. This suggests that the FC network in this area is highly resilient, with many overlapping service areas and more logistical flexibility.

The map highlights how major population centers, such as those near New York, Chicago, and Atlanta, are poorly covered with FC clusters. This is also due to the fact that the distance bucket scale is not linear (range of 50, than 100, than 300) so the further the Preferred FC is, the more likely we are to have several FC in the cluster and the opposite for closer Preferred FC. Since we have a higher density of FC on the coasts, the preferred FC tends to be in one of the closest distance bucket, reducing the probability for many other FC to be in such the same distance.

Geographical plot coloring the various clusters from green to red in a 15FC Network

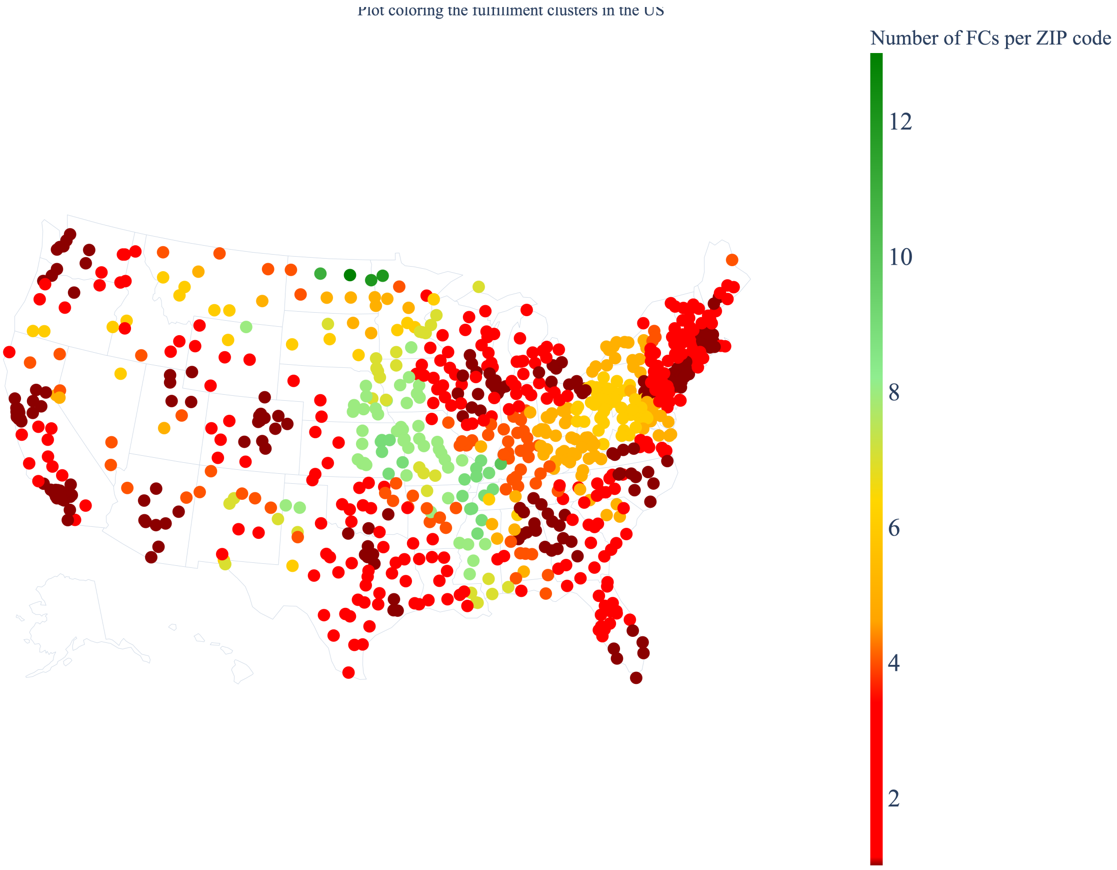

Figure 31: Geographical plot coloring the various clusters based on the number of FCs that can serve the cluster, in a red (lowest) – yellow – green (highest) color scheme in a 15FC Network

We note that this map is mostly red and this not due to the way we shaped the legend. Red corresponds to clusters of 4 or less FC. This means that most clusters are made of less than 4 FC. Nevertheless, we notice that in the center of the country, where we have less FC, we have a average of 8 FC per cluster. So event if they are far from the ZIP code, we have more flexibility on the FC to chose where to ship from to this area. On the cost, the clusters are smaller because we have more FC here so if we try to chose a FC that is further we will change of distance bucket. Let's note that distance buckets are not linear so the ranges are tighter for small distances than big ones.

Proportion of demand that can only be served by a single FC

Proportion of demand that can only be served by a single FC in a 1FC Network

In a 1FC Network there is no discussion to have, all ZIp can only be served by a single FC.

Proportion of demand that can only be served by a single FC in a 4FC Network

In order to answer this question we will first sort the cluster depending on whether they have several FC, looking for a coma in the `Preferred_FCs` column. If they don't, we will add the corresponding ZIP in each of these clusters into the list `single_fc_demand`; We will then count the number of items in this list, this will be the total number of ZIP that can only be served by a single FC. We will divide this quantity by the total number of demand (being 1 here since we are talking in PMF). Here is the code used in Python:

```python
single_fc_zip =
    fc_clusters[fc_clusters['Preferred_FCs'].str.count(',') == 0]

single_fc_demand = pd.merge(single_fc_zip, demand_data,
    on='ZIP3')
total_single_fc_demand = single_fc_demand['PMF'].sum()

total_demand = demand_data['PMF'].sum()

proportion_single_fc_demand = total_single_fc_demand /
    total_demand
```

Finally we find that 36% of the demand can only be served by one FC.

Proportion of demand that can only be served by a single FC in a 15FC Network

We proceed as in the previous part.

Finally we find that 39% of the demand can only be served by one FC.

This number is smaller than in a 4FC Network, which means that we have less flexibility in a 15FC Network.

Variation in demand distribution over shipping distance buckets

Variation in demand distribution over shipping distance buckets in a 1FC Network

In a 1-FC Network, we cannot count various preferred network. So we just plot the demand distribution over the shipping distance.

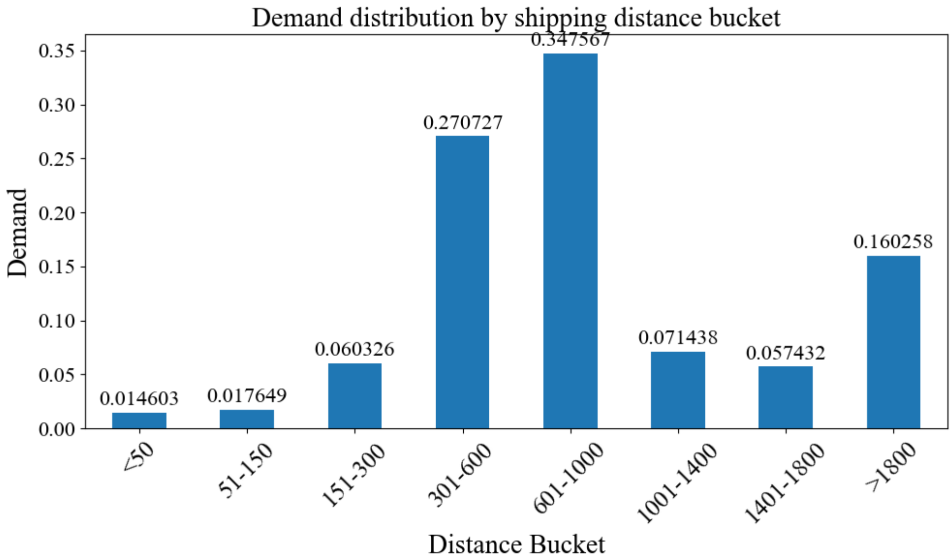

Figure 32: Variation in demand distribution over the shipping distance buckets in a 1FC Network

As acknowledged in a previous part, in a 1 FC Network, the shipping distance goes further than in a multiple FCs. We even have 16% of the demand being served above 1,800 miles.

Variation in demand distribution over shipping distance buckets in a 4FC Network

We will estimate the variation in demand distribution over the shipping distance buckets, assuming 90% demand satisfaction by closest FC, and the remaining 10% split equally among the other FCs serving the cluster. We will proceed as follows.

- Assign demand for ZIPs with only one FC in their cluster than remove the single FC rows from the original DataFrame.
- Assign ZIPs to their preferred FC until 90% of total demand is reached then remove rows that have been assigned.
- Assign the remaining 10% of demand among all FCs in the cluster.

Here is the final plot we get:

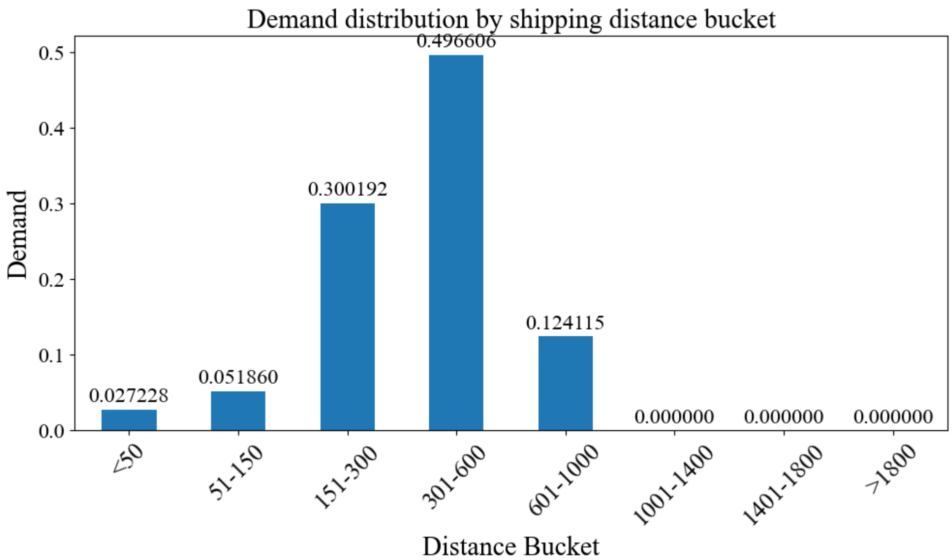

Figure 33: Variation in demand distribution over the shipping distance buckets in a 4FC Network

By adding more FC, we obviously reduced the distance between each ZIP code to its preferred FC, having no demand above a thousand miles.

Variation in demand distribution over shipping distance buckets in a 15FC Network

We proceed as in a 4FC Network. Here is the final plot we get:

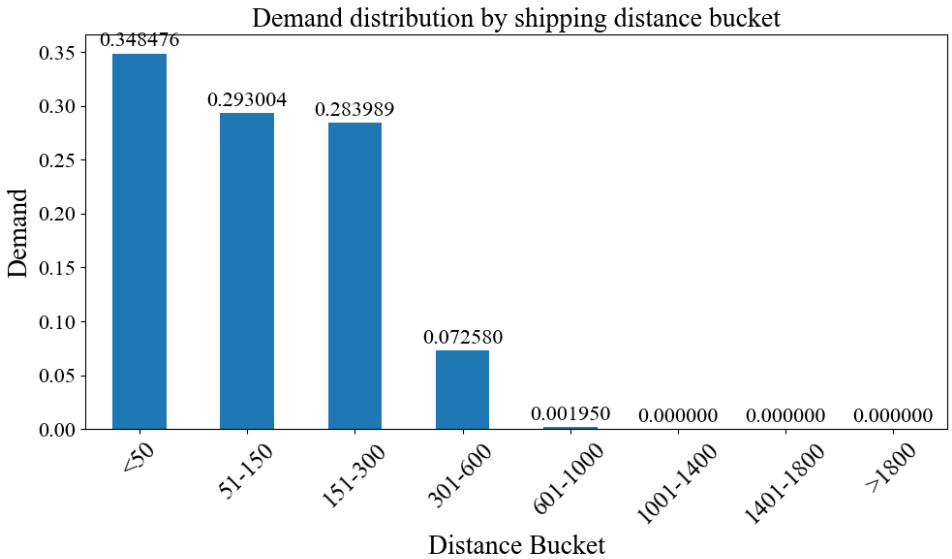

Figure 34: Variation in demand distribution over the shipping distance buckets in a 15FC Network

We observe that 92% of the demand will be fulfilled within less than 300 miles travel distance. This means that demand can be met for this ZIP by minimizing travel costs. Nevertheless, in this configuration, we have to set up 15 FC so we will need to consider the cost of setting these up and making them work. And we still have some demand to meet in the 601-1000 miles bucket (0.2%). Nevertheless, we managed to meet 32% of the demand in less than 50 miles. So if this is important to customer, we might consider this option when analyzing costs afterwards.

TASK 5

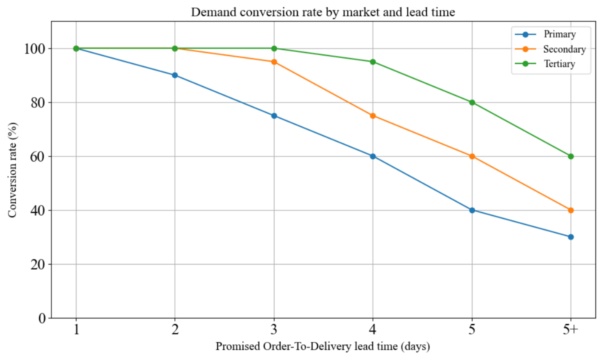

Figure 35: Maximum conversion rate in each market given the FCs assigned to it, for each potential OTD time promise options

We are given the various conversion rates for each market and each distance bucket. We note, as expected, that as the OTD grows, the conversion rate decreases. Indeed, if the customer can have the product in one day he is more likely to buy it than if he is to get it in 5 or more.

We will also need to consider the following shipment costs.

	Promised Order-To-Delivery Lead Time (days)					
Zone (*miles*)	1	2	3	4	5	5+
1 (*<50*)	$60.71	$35.30	$22.97	$13.86	$12.07	$10.25
2 (*51-150*)	$75.89	$44.13	$28.71	$17.33	$15.09	$12.81
3 (*151-300*)	$102.53	$58.52	$39.15	$23.77	$19.05	$15.97
4 (*301-600*)	$144.55	$79.37	$53.25	$31.57	$24.15	$19.81
5 (*601-1000*)	$207.84	$92.39	$65.49	$34.31	$28.73	$22.49
6 (*1001-1400*)	$269.21	$142.65	$89.50	$49.12	$34.00	$25.92
7 (*1401-1800*)	$284.08	$179.54	$120.24	$77.63	$36.17	$27.65
8 (*>1800*)	$291.21	$185.41	$133.04	$89.36	$38.81	$30.07

Figure 36: Shipment Costs (per product unit)

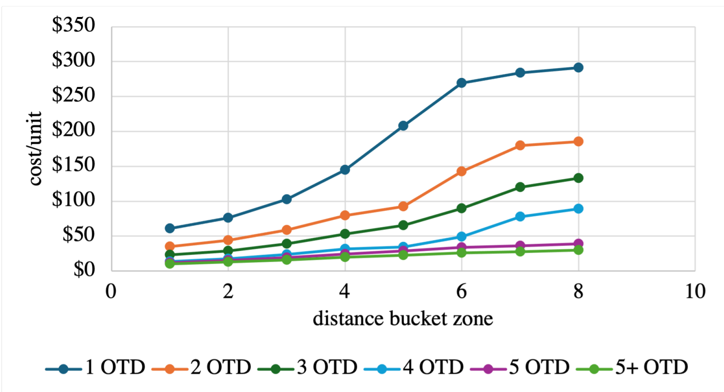

Figure 37: Shipments costs as a function of the distance for each OTD lead time

Annual profit for each market and each option in a 1FC Network

In the Python Code in task 4, we saved a csv file named *fc_clusters_with_zones* containing the following columns:

ZIP3	$Preferred_{FCs}$	$Preferred_{FC}$	Distance Market	State	PMF	num_{FCs}	Distance Zone

Table 3: Type of data in the fc_cluster_with_zones.csv file

We will save this file as *fc_clusters_with_zones_task5_1FC* and work on it, using three sheets, one for each type of market, to compute in this order :

- The shipment cost per unit: using the table in figure 36
- The total shipment cost $= 2,000,000 \times$ PMF$\times$shipment cost per unit$\times$conversion rate
- The total revenue $= 300 \times 2,000,000 \times$ PMF $\times$ conversion rate
- The net revenue = total revenue - total shipment cost
- The gross operating profit = net revenue - $75 \times 2,000,000 \times$ PMF $\times$ conversion rate

We will do this computation for each OTD lead time.

For each distance bucket zone and each OTD, we will plot the gross operating profit.

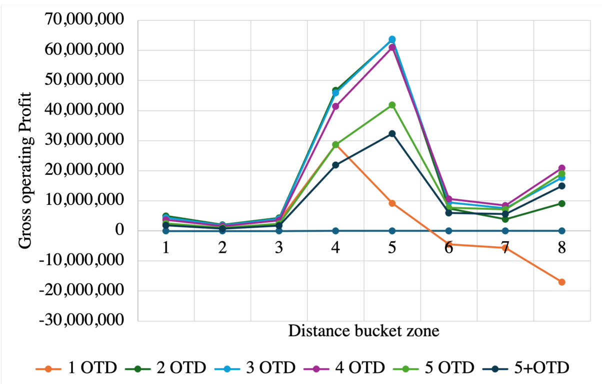

Figure 38: Annual profit for the Primary market for each option in a 1FC Network

In figure 38, representing the annual profit for the Primary market for each option in a 1FC Network, we note that the profit remains mostly the same (1 to 4 million of dollars) for all OTD lead times if the package has to be delivered in the range [0-300] miles. Then between 300 and 1400 miles, the profit explodes up to $65,000,000. This might be strange, since one might expect the profit to be bigger if the delivery is close to the FC. This is indeed true but we also need to take into account the demand and it appears, as shown in figure 23 that the majority of the demand in the primary market in a single FC network is to be fulfilled in the range 300-1400. So it makes sense that the profit grows for this distance bucket.

What about the most beneficial lead time? We could think that the biggest our conversion rate is, the more profit we make. Nevertheless, achieving a high conversion rate means delivering in a very short time which is pretty expensive. So we actually need to find the best combination between conversion rate and costs. In this case, it appears for distance buckets 3 to 6 that an OTD lead time of 3 days maximizes the profit.

Moreover, we realize that achieving a hundred percent conversion rate is not something we should go for since it definitely doesn't maximize profit and even results in loss for distances greater than 1,400 miles. This actually makes sense since achieving such rate is extremely expensive and since most of the demand is above 1,000 miles, shipments costs per unit are above \$250. When we add the \$75 COGS, we notice that the costs are above the selling price of \$300.

Finally, we note that we make bigger profit over the furthest shipping distance than the closest. This might seem counter-intuitive since shipping far is more expensive. However, in figure 37 we see that the shipment costs are not linear are and have a little slope for high OTD. Moreover, we realize more 10 times more sells in zone 8 than zone 1 and the revenue increases faster than the costs, which explains this difference of profit.

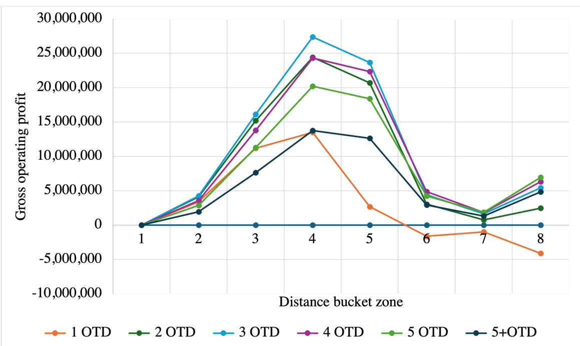

Figure 39: Annual profit for the Secondary market for each option in a 1FC Network

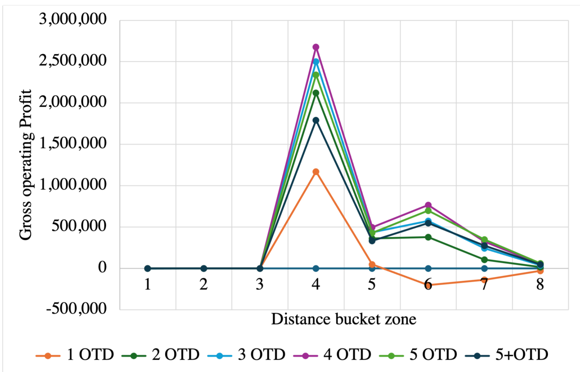

Figure 40: Annual profit for the Tertiary market for each option in a 1FC Network

Figures 39 and 40 plots the gross profit for each OTD lead time in function of the distance bucket zone for Secondary and Tertiary markets. As in the previous plot, we recognize some peak profit associated with the biggest demand. We also acknowledge as before that a one day OTD may result in loss and is never the best option to maximize profit for whatever distance bucket.

Annual profit for each market and each option in a 4FC Network

In order to deal with the 4FC Network case, we will use an Excel spreadsheet as in the previous case, derived from the Python code from task 4, called *fc_clusters_with_zones_task5_4FC*. We will proceed according to the same method as in a single FC Network.

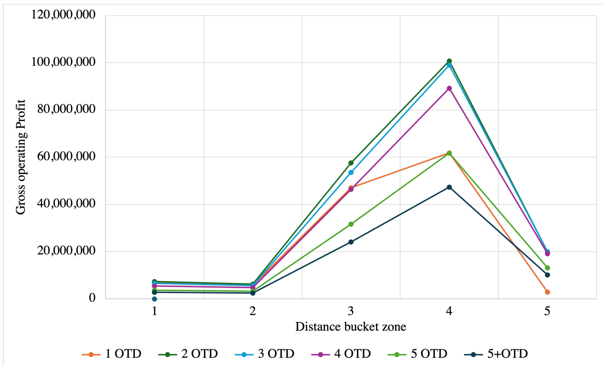

Figure 41: Annual profit for the Primary market for each option in a 4FC Network

In figure 41, we note that since we have multiplied by 4 the number of FC, we have reduced the maximum distance buckets which means we have reduced the maximum shipments costs. In a 4FC Network, the largest shipping distance is 1,000 mile. When we look at figure 37, we notice that we have two different parts in the graph, enhancing that the shipping cost/unit grows faster for distances after a 1,000 mile. This means we have indeed reduced the shipping costs.

This explains why profits are way greater than before. Indeed profit reaches $100,000,000 in the Primary market in the Network while it plateaued at $65,000,000 in the single network.

We notice furthermore, that we are never in deficit in such network even if the 1-OTD is still not the best option to maximize profit.

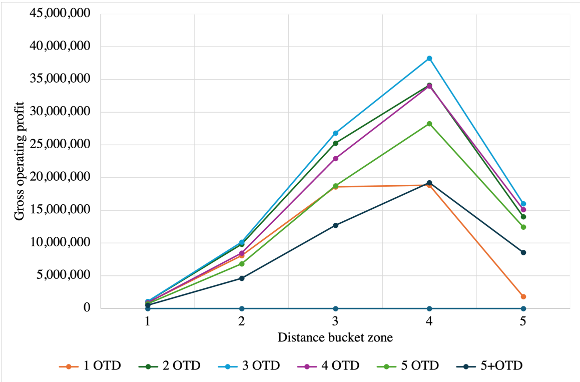

Figure 42: Annual profit for the Secondary market for each option in a 4FC Network

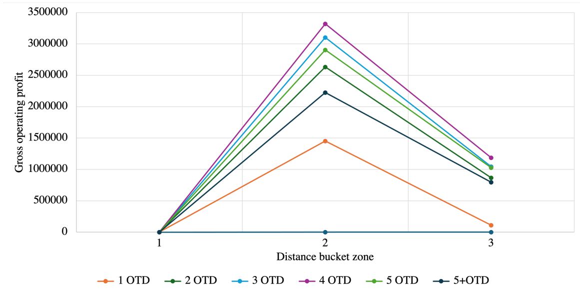

Figure 43: Annual profit for the Tertiary market for each option in a 4FC Network

Secondary and Tertiary markets also have higher gross operating profit in this case. We still have the same trend over which OTD lead times are the most beneficial, these being the 3-OTD for the secondary and 4-OTD for tertiary.

We acknowledge that there is indeed a compromise to find between maximizing conversion rates and minimizing shipments costs. Indeed, for a high conversion rate (1 OTD), the profit is not maximized since the shipment costs are too high. And for a minimal shipment rate (5+ OTD), the conversion rate is too low, we lack of customers, which means that the profit is not maximized. This is the reason why the optimal scenarios involve 3 or 4 OTD.

Annual profit for each market and each option in a 15FC Network

The spreadsheet used here is named *fc_clusters_with_zones_task5_15FC*.

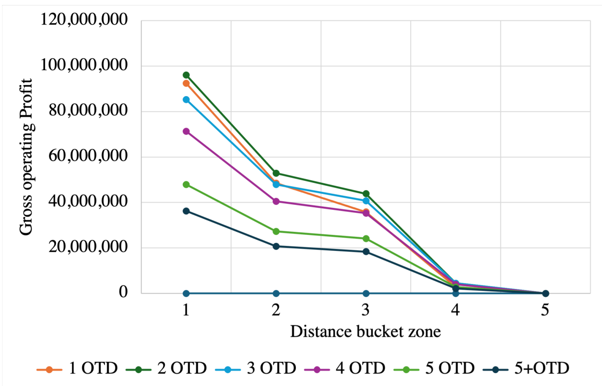

Figure 44: Annual profit for the Primary market for each option in a 15FC Network

In this scenario, more than 50% of the demand is served under 50 miles (zone 1). This explains why we get a higher profit for this zone. We acknowledge that we don't have to ship after zone 4, which might result in savings. Since we are mostly shipping to ZIP close to the FC, we see that for the first time, a 1-OTD can prove to be beneficial, just after the 2-OTD.

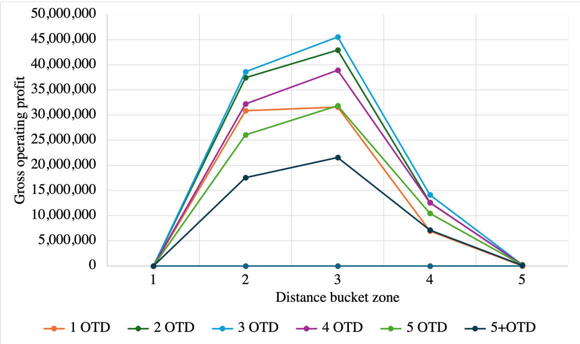

Figure 45: Annual profit for the Secondary market for each option in a 15FC Network

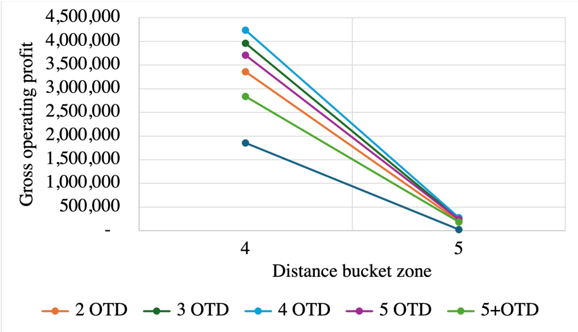

Figure 46: Annual profit for the Tertiary market for each option in a 15FC Network

The secondary market, displayed in figure 45 serves further ZIP than the primary which explains why, in order to maximize costs, we need to go 3-OTD. The tertiary market, displayed in figure 46 serves only zones 4 and 5 that are far from the FC and involves higher shipment costs.

TASK 6

For this task, we shall look at the most profitable configuration and lead time delivery to maximize profit. We will look at each type of market separately than combine them to look overall which FC configuration maximizes profit.
In order to do so, we use the spreadsheet *task6_overall* that assembles the total cumulative profit for each market, regardless of the zones. We have previously computed the sum of the profit for each option in the spreadsheets *fc_clusters_with_zones_task6_1FC*, *fc_clusters_with_zones_task6_4FC*, *fc_clusters_with_zones_task6_15FC*.

Annual profit for each market and each option in a 1FC Network

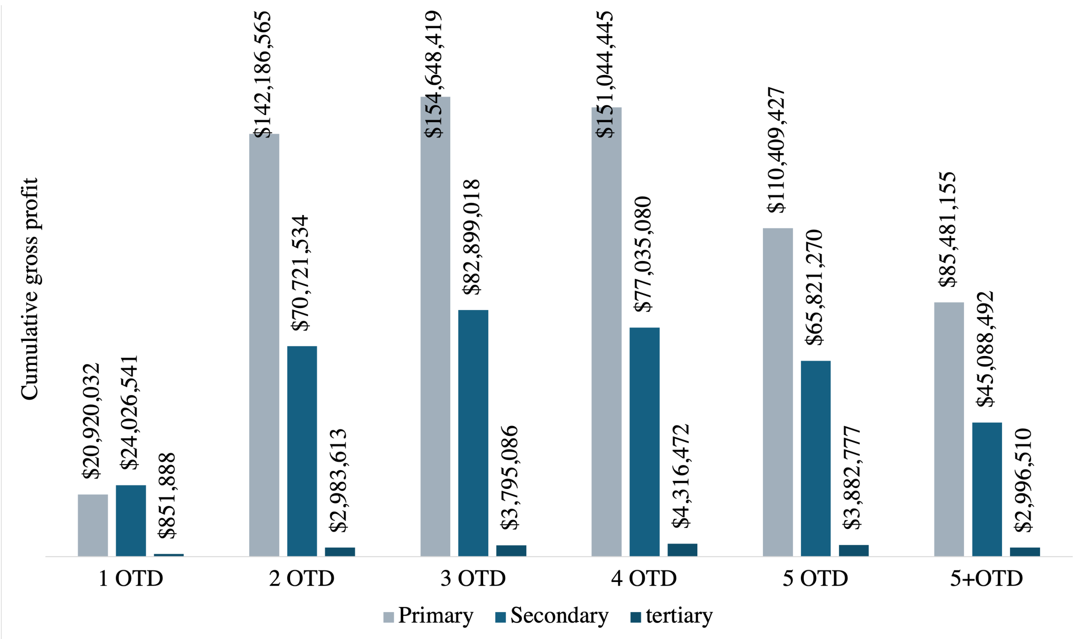

Figure 47: Cumulative profit for each market market and each option in a 1FC Network

The primary market is the most profitable overall, especially for the 3-OTD option where profit reaches more than $150,000,000. The secondary market also makes its biggest profit (more than $80,000,000) for this option. Since the Primary market serves further ZIP, it is more profitable for the 4-OTD option with a profit of more than $4,000.

Annual profit for each market and each option in a 4FC Network

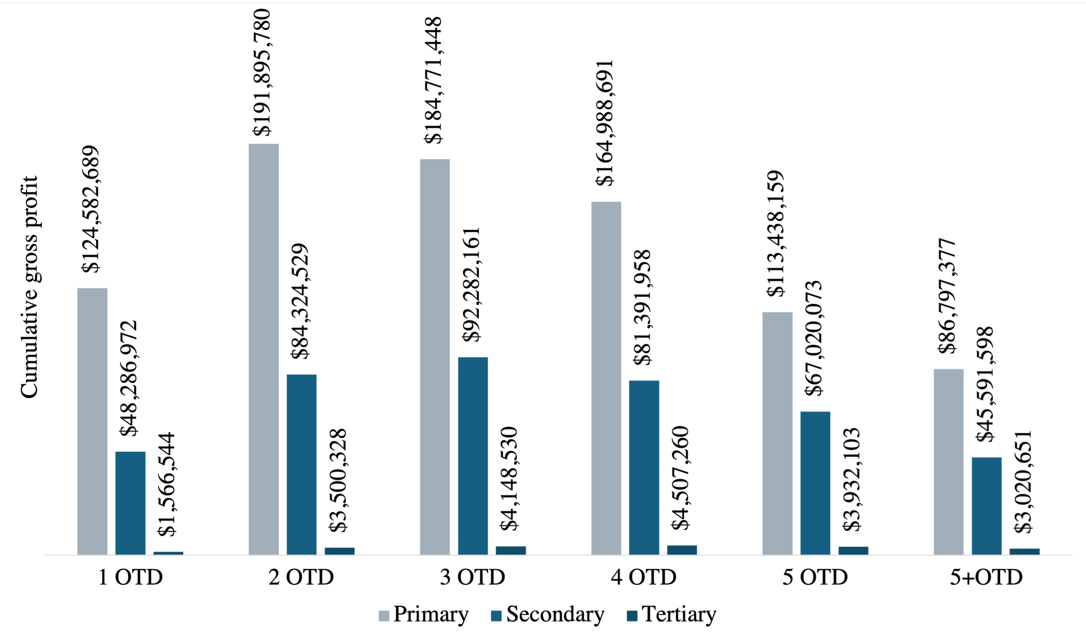

Figure 48: Cumulative profit for each market market and each option in a 4FC Network

The primary market is the most profitable overall, especially for the 2-OTD option where profit reaches more than $190,000,000. The secondary market on its side, makes its biggest profit (more than $90,000,000) for the 3-OTD option. Since the Primary market serves further ZIP, it is more profitable for the 4-OTD option with a profit of more than $4,000.

Annual profit for each market and each option in a 15FC Network

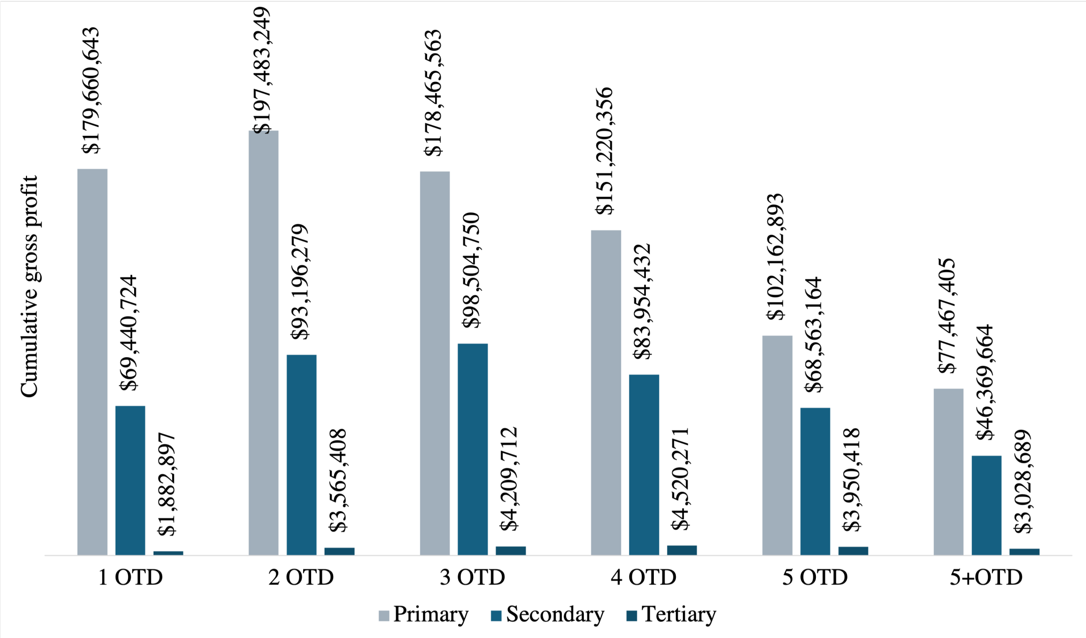

Figure 49: Cumulative profit for each market market and each option in a 15FC Network

The primary market is the most profitable overall, especially for the 2-OTD option where profit reaches almost $200,000,000. The secondary market also makes its biggest profit (more than $90,000,000) for this option. Since the Primary market serves further ZIP, it is more profitable for the 4-OTD option with a profit of more than $4,000. We note that this market is the most stable since it always reaches its maximum profit at the 4-OTD option.

Total cumulative profit for each option and each type of Network

In order to chosse the most profitable combination, we will group all markets into a single market portfolio, looking at the biggest profit in each network configuration.

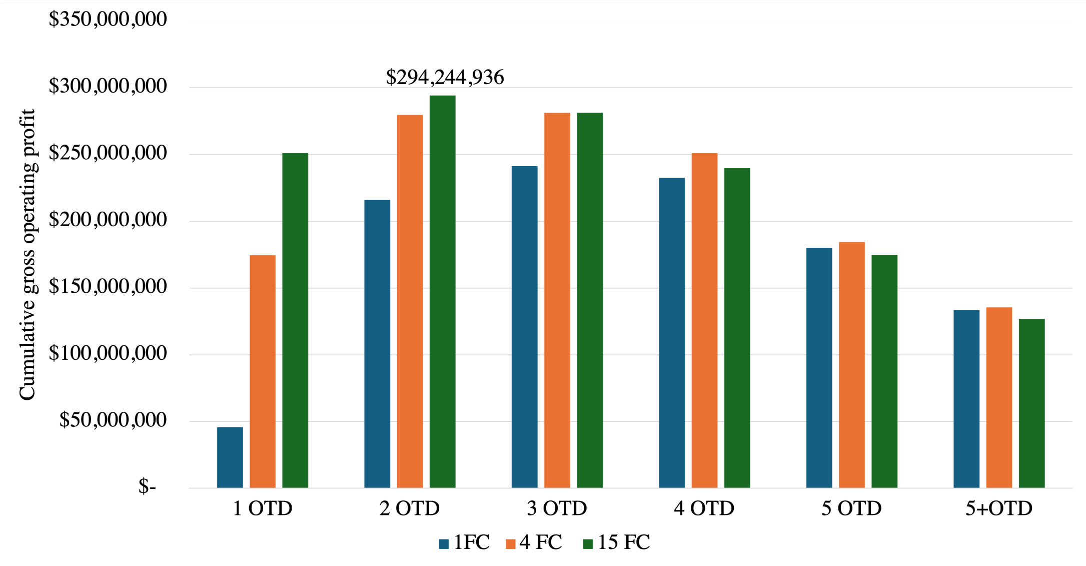

Figure 50: Total cumulative profit for each option and each type of Network

When we combine all the markets together, it ensues that that the most profitable configuration is the **15 FC network with the 2-OTD option** with a cumulative gross profit of almost $300,000,000. Nevertheless, here we did not consider the costs of having so many FC and the costs of replenishing them, so we might want to consider these costs before making a final decision. But so far, the 15FC network wins, which is actually pretty rational since we minimize shipment costs in this configuration. Let's still acknowledge that other configurations do not necessarily show huge profit disparities. Indeed when we compute the relative standard deviation, we obtain the following results:

option	1 OTD	2 OTD	3 OTD	4 OTD	5 OTD	5+OTD
relative standard deviation	66%	16%	9%	4%	3%	3%

As from option 3-OTD, this means that after observation of all remaining costs, we might go for a network with fewer FC.

TASK 7

Initial assumptions regarding overall demand and individual FC demand

We will be using a similar process to that seen in Task 2 in order to forecast demands for 99%, 95%, 68%, and 50% optimistic levels. For each day, an average of 30 trials are conducted to determine the demand on that day. This yields the countrywide demand which will be useful for examining the 1FC network. We will use the aggregated demand share by fulfillment center found in Task 3 to further break down the forecasted demand for the 4FC and 15FC networks.

Stock to be maintained for 1FC network

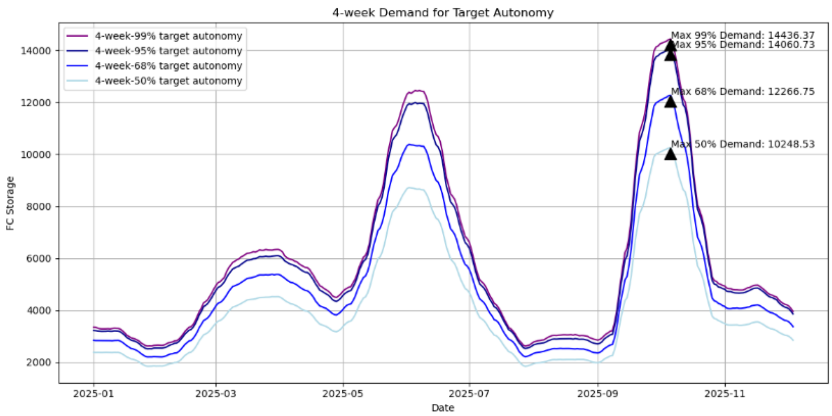

Figure 51: Required stock in single FC model for 4-week target autonomy at different levels of robustness

It can be seen that the max demand at the singular FC is about 41% greater or 4000 more units for a 99% robust fulfillment network compared to 50%. However, when there are not times of large growth, the gap between different robust models decreases in total units to a difference of about 1000 with a similar difference in percentage.

Stock to be maintained for 4FC network

For this section, we want to examine the required stock at each FC over the year to reach 4-week target autonomy at different robust levels. Previously in task 3, we calculated the percentage shares of demand within each of the distribution centers we can directly use this to examine the 4FC network since there is no location-specific seasonality and enough trials would reflect the true mean anyways.

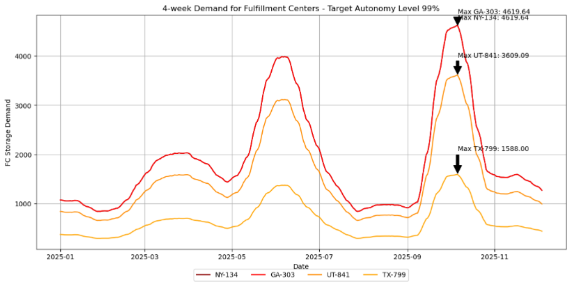

Figure 52: Required stock in 4FC model for 4-week target autonomy at 99% robustness

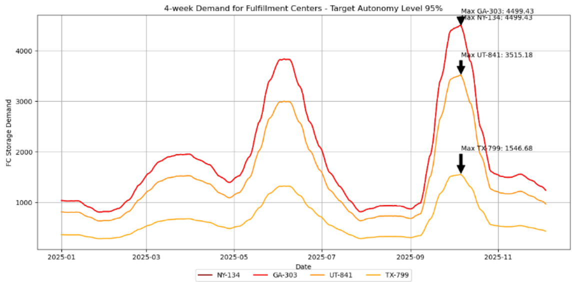

Figure 53: Required stock in 4FC model for 4-week target autonomy at 95% robustness

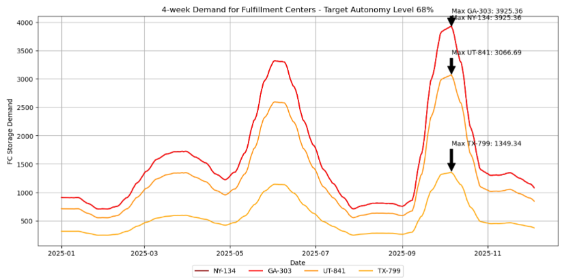

Figure 54: Required stock in 4FC model for 4-week target autonomy at 68% robustness

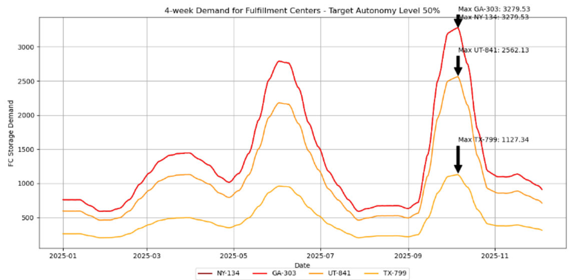

Figure 55: Required stock in 4FC model for 4-week target autonomy at 50% robustness

We can see that the max stock for any single FC within the 4FC network is 4620 for 99% or 3280 for 50%. This is a large reduction from the 1FC network which could be beneficial depending on the size and capacity constraints for a single FC.

Fulfillment Center	**99 percent**	**95 percent**	**68 percent**	**50 percent**
NY-134	4620	4499	3925	3280
GA-303	4620	4499	3925	3280
UT-841	3609	3515	3067	2562
TX-799	1588	1547	1349	1127
Overall	14437	14060	12267	10249

Table 4: Max demand at each FC for every robust level

Above, we can see all of the max demands at each of the FC.

Stock to be maintained for 15FC network

In this section, we examine the required stock at each FC for the same 4-week target autonomy at different levels within a 15FC network. Similar assumptions and general approach are used from the analysis in the 4FC network.

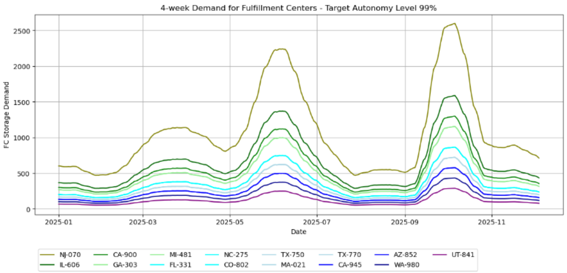

Figure 56: Required stock in 15FC model for 4-week target autonomy at 99% robustness

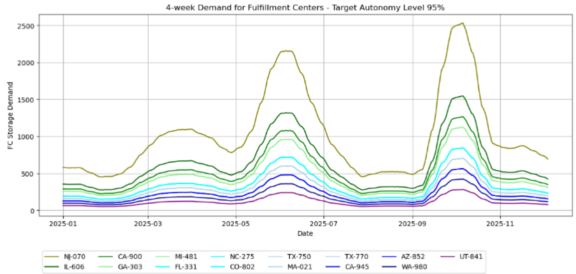

Figure 57: Required stock in 15FC model for 4-week target autonomy at 95% robustness

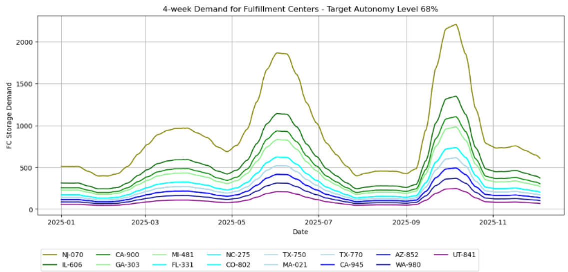

Figure 58: Required stock in 15FC model for 4-week target autonomy at 68% robustness

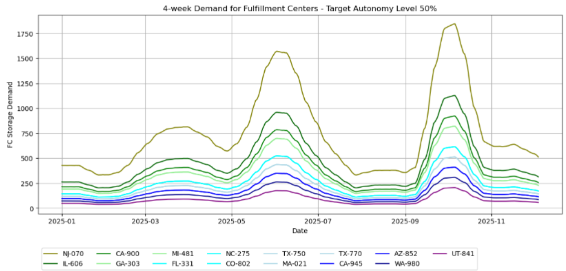

Figure 59: Required stock in 15FC model for 4-week target autonomy at 50% robustness

Above, we can see the stock required at each FC in order to reach 4-week target autonomy. The largest maximum stock for any one FC is about 2600 while the smallest maximum is roughly 300 units. This does a good job of spreading out the demand across multiple FC, however some FC may be underutilized like UT-841 if all centers are the same size.

Fulfillment Center	99 percent	95 percent	68 percent	50 percent
NJ-070	2599	2531	2208	1845
IL-606	1588	1547	1349	1127
CA-900	1299	1265	1104	922
GA-303	1155	1125	981	820
MI-481	1155	1125	981	820
FL-331	866	844	736	615
NC-275	866	844	736	615
CO-802	866	844	736	615
TX-750	722	703	613	512
MA-021	722	703	613	512
TX-770	722	703	613	512
CA-945	577	562	491	410
AZ-852	577	562	491	410
WA-980	433	422	368	307
UT-841	289	281	245	205
Overall	14437	14060	12267	10249

Table 5: Max demand at each FC for every robust level

From this table, we can see that some fulfillment centers require much larger stocks than others in order to maintain the 4-week target autonomy.

Stock to be maintained at DC

While we want to maintain a 4-week target autonomy at each FC, we also want to maintain an overall 6-week target autonomy where the additional units will be stored at the one DC.

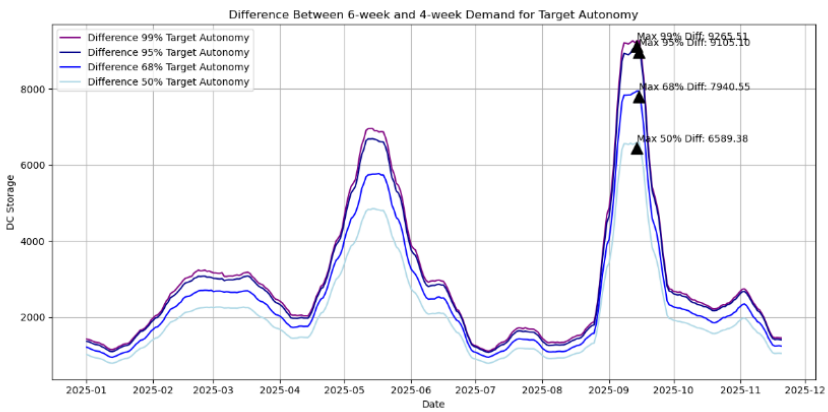

Figure 60: Required stock in single DC model for 6-week target autonomy overall at different levels of robustness

We can see that there are periods of time when the number of units stored at the DC can be up to 9266 for a 99% robustness or 6589 for a 50% robustness. Either way, this is very important because this additional space will have to be accounted for within the DC or rented nearby space.

TASK 8

Daily production from Assembly Factory to maintain 99% 6-week target autonomy at DC

Using the 6-week target autonomy stock needed in DC, 4-week target autonomy at FC, and daily demand, we can determine the daily production needs of the assembly factory (AF). We will assume that the FC and DC begin at the correct 4-week and 6-week stock. Additionally, stock from DC only decreases if stock in FC falls below the 4-week target autonomy. AF production only begins if the DC stock falls below the 6-week target autonomy

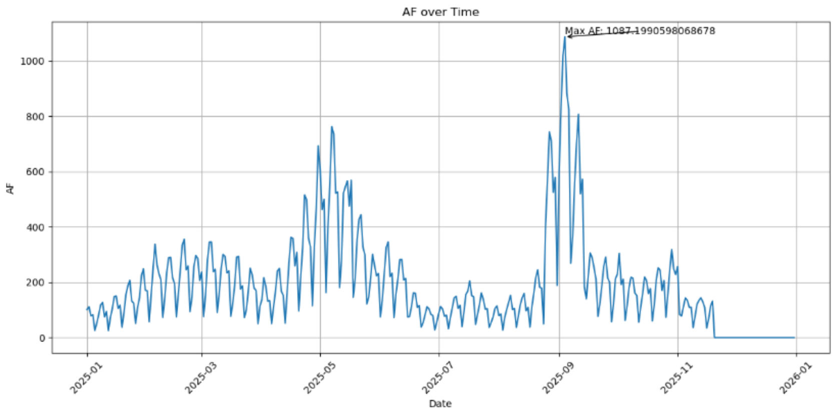

Figure 61: Required AF production per day in order to maintain 6-week target autonomy at DC

From this graph, we can see that there are times of large production and times of much less production. There is one day in particular, September 4th, 2025, where the required production for the day is 1087 units to maintain the 6-week target autonomy at the DC and 4-week autonomy in the FC. This value is likely infeasible and there will also be days when the factory is underutilized. For example, there are many days where less than 100 units are produced. Therefore, it will be crucial to see the smoothed production rate in order to diminish the peaks and increase utilization during the valleys.

Steady production rate from Assembly Factory to maintain 99% 6-week target autonomy at DC

In this scenario, we want to take a look at the steady rate of production needed at the AF in order to maintain at least the 6-week target autonomy at 99% robustness. This will smooth out the time of under and over-utilization within the AF but will result in more storage within the DC.

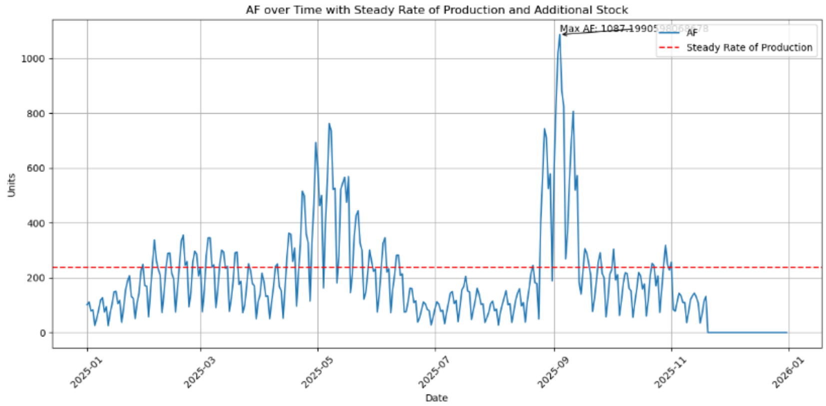

Figure 62: Required steady AF production per day in order to maintain 6-week target autonomy at DC compared to daily correction

It can be seen above that the steady rate in order to achieve 99% robust 6-week target autonomy is 238 units per day. This is a much more manageable rate of production than 1000 plus units per day. However, it is really important to look at the implications that this has on the overall number of units stored within the DC.

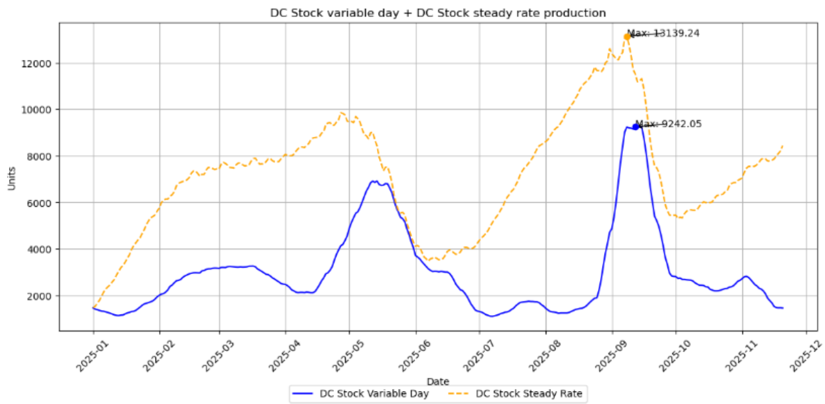

Figure 63: Required stock for steady AF production per day in order to maintain 6-week target autonomy at DC compared to daily correction

There is obviously a trade-off to having constant production. Since we produce more anticipating future growth, we must store much more additional stock throughout the year. Above, it can be seen that the maximum storage previously within the DC was 9242 units under the day of variable replenishment. However, under a steady state production, the maximum stock units at any one time is 13139. This is 42% higher. Additionally, there are periods when the stock required for the steady rate is more than 5 times the stock required for the variable day.

Financial considerations due to production rate from Assembly Factory to maintain 99% 6-week target autonomy at DC

There are two main financial costs associated with this process: production capacity and storage capacity. We will first take a look at the costs associated with the steady-state production. The calculation for production capacity costs. We will assume that the AF will be set up initially for the maximum output on any day. Therefore, the first day will experience an initial fixed cost of $140 per unit multiplied by the max output of all of the days. There is also a variable cost of $10 per unit per day. This results in the following graph.

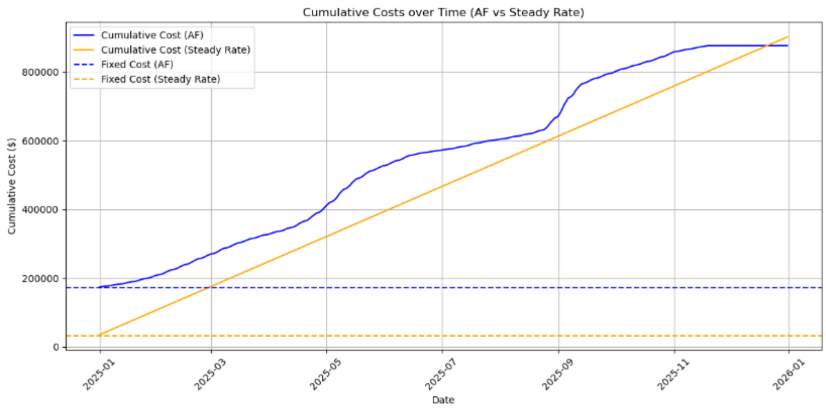

Figure 64: Cumulative cost per day for steady rate and variable day production

We can see that the initial investment is much higher for the variable production than the steady. The graphs are sometimes very similar in overall cost as seen in 2025-09. Assume that the graphs stop at 2025-11 since the production per day drops to zero due to there being no data available for 6 weeks in advance.

It is important to also consider the storage capacity financial costs. There is an initial set-up cost $4.62 per unit which will be calculated by multiplying the max projected stored units over the time interval by the per unit rate. Additionally, there is a variable cost which is $0.66 per unit per day. This results in the following graph.

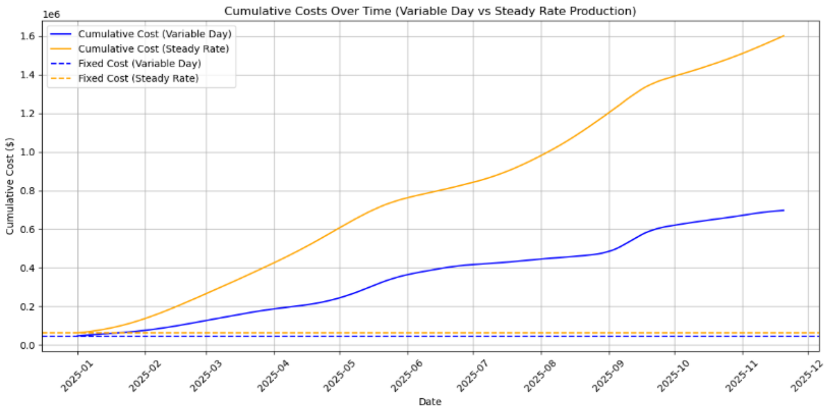

Figure 65: Cumulative cost per day for steady rate and variable day storage

We can see that the initial investment of the steady rate is slightly higher than that of the variable day production method. Additionally, the steady-state method holds more storage year-round and therefore results in a large difference in cost. Now we will combine these two costs to see the total costs over time for each method.

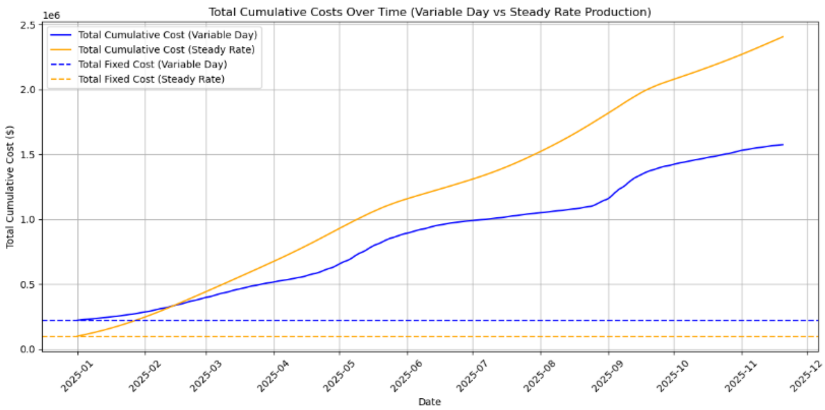

Figure 66: Cumulative cost per day for steady rate and variable day methods total

We can see that the overall costs for the variable day production and storage are less than that of the steady rate. This is due to the large amount of units to be stored in order to never fall below the 6-week 99% target autonomy. The actual production costs are less for a steady rate but the storage costs are much more. There is likely a steady rate that is the most cost-efficient but does not reach the 6-week target autonomy stock goal.

TASK 9

Algorithm computation

First of all, we will provide insights on the code used to solve task 9. The main function: algorithm(FC, replenishment_i, replenishment_t, r_a_days=14), takes in parameter the FC, the replenishment interval (`replenishment_i`), the replenishment time (`replenishment_t`) that is given in a CSV spreadsheet, and the `r_a_days`, which is the number of days in the robust autonomy threshold (not utilized for task 9 but will have some importance in task 10). This means that the algorithm implements a 7-day replenishment interval and a 14-day minimum robust autonomy threshold for each FC.
It tracks the FC's robust autonomy days daily, considering pipeline inventory (on-hand and en route) and the upper 99% demand forecast. A replenishment is triggered when:

- Robust autonomy days fall below 14, or
- 7 days have passed since the last replenishment

The replenishment quantity is the maximum of:

- (4-week-99% autonomy target inventory - pipeline inventory)
- 10% of a Full Truck Load (32 units)

In this basic case, the inventory is initialized at 0 but can be given any value.
The code generates inventory profiles for each FC over the planning horizon. The 'tricky' part is to compute the pipeline inventory. For that we had to create a `DataFrame()` object named `en_route_df` that stores the ongoing shipment.
To compute the pipeline inventory we take the inventory of last day, and we sum the shipment log over the next days (replenishment days):
This gives us the following computation:

$$\forall d : PI_d = Id - 1 + \sum ER_{d,d+rt}$$

With:

- PI_d: the pipeline inventory computed each day
- $Id - 1$ inventory the previous day
- $ER_{d,d+rt}$ the table of all shipments already departed to this facility

Let us dive into more detail about the replenishment process:
If: $\forall d, PI_d \leq RAT_{14d}$ or $d - lrd = 7$ it triggers an automatic replenishment shipment.
With:

- RAT_{14d} : robust autonomy threshold over 14 days
- lrd: last replenishment day (last day a replenishment figures in the shipment log)

Therefore:

$$A_d = d + rt$$

- A_d being the arrival day for next shipment
- rt being the replenishment time (specific to each FC)

The replenishment amount RA is then:

$$RA = RAT_{28d} - PI_d$$

We then add the maximum between the replenishment amount and 34 to the shipment log at the dedicated arrival date: A_D.
Ultimately: $lrd = d$.

Inventory Level Analysis

We can iterate through all the fulfillment centers and plot the inventory levels for each day of the year.

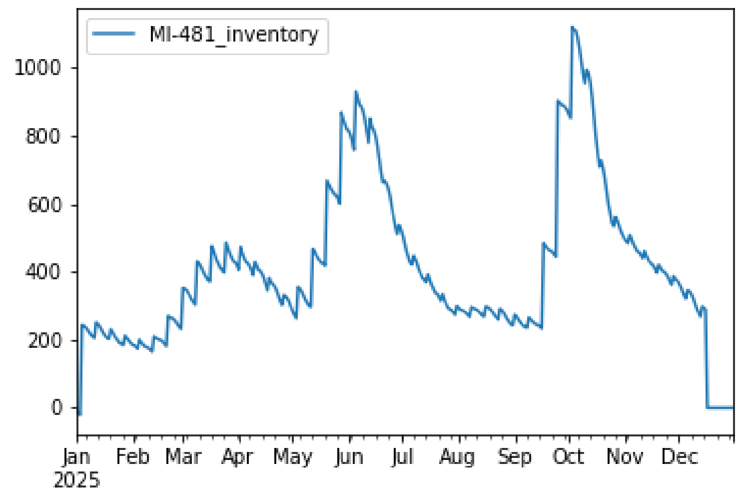

Figure 67: Inventory Level of the Minnesota Fulfillment Center

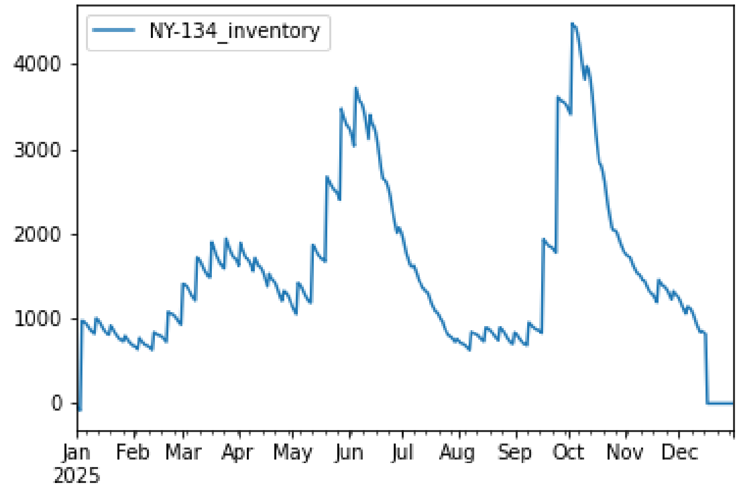

Figure 68: Inventory Level of the New York Fulfillment Center

The majority of the fulfillment centers have the same inventory profile, though with slight differences. At the beginning, we can observe the arrival of an important shipment. This is due to the fact that the inventory levels were initialized a 0 for all fulfillment centers. Therefore, the FC needs to accommodate rapidly to the demand. After a while, the replenishment amount of the fulfillment centers become more regular, just making slight adjustments. We can then observe two peaks, one in June and the other mid October. What is interesting to see is that the algorithm never sends all the product at once, bu spreads them across several delivery days.

Another graph that appears interesting to analyze would be the shipment log per day of year:

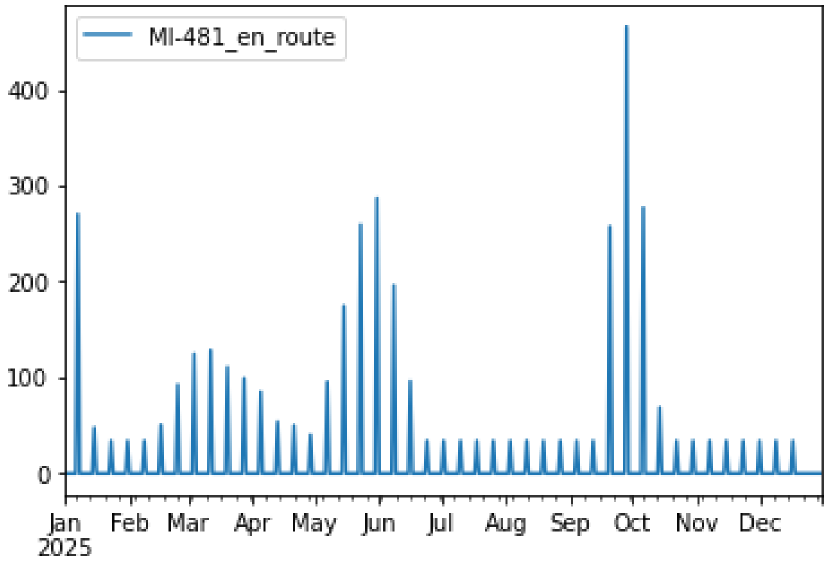

Figure 69: Shipment Log for the Minnesota Fulfillment Center

Throughout the year, we observe very steady shipment amounts (from July to September). This signifies that the shipments are LTL (10% of FTL: 34 units). This is what we will be trying to minimize in Task 10.

Some inventory levels, especially the ones with the lowest demand, have original plots: they rapidly attain a sufficient amount of inventory and the demand is so low that they only use 10% TL shipments to refill stock. This explains why they maintain a high level of inventory througout the year.

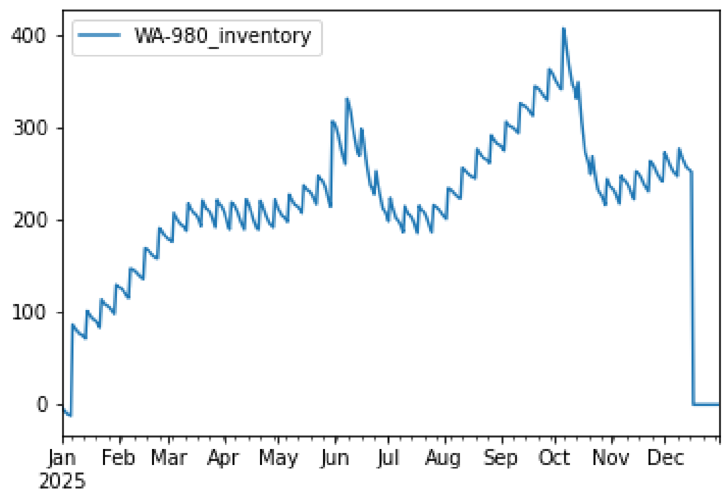

Figure 70: Shipment Log for the Wyoming Fulfillment Center

Throughput Capacity

The throughput capacity can be defined as:

$$TC_{d,FC} = \sum Inbound_{d,FC} + \sum Outbound_{d,FC}$$

We will be trying to compute a robust throughput capacity: max(TC).
For a 4 FC strategy, we obtain the following results:

GA-303	4608.0
NY-134	4515.0
TX-799	1540.0
UT-841	3473.0

(1)

For a 15 FC strategy, we obtain the following results:

Location	**Value**
AZ-852	556
GA-303	1178
CA-900	1250
CA-945	554
CO-802	840
UT-841	204
FL-331	847
IL-606	1552
MA-021	705
MI-481	1129
NC-275	855
NJ-070	2540
TX-750	705
TX-770	705
WA-980	416

(2)

We can see that for a 15 FC, the max throughput capacity is lower. This is not a surprise, since the amount of FCs are increased, the demand can be better spread out.
To compute the throughput capacity of the distribution center DC, we will proceed slightly differently: when an order is sent from a FC, we will consider it is sent right away by the DC. Therefore we can create the DC throughput log for each FC. We can group this demand for each FC and by using Excel, we find that for the 15 DC, the maximum throughput (which is only outbound for the DC) is **4,877**, and happens on **September 23**.
Similarly, for the 4 FC strategy, the maximum throughput for the DC is around **7,000 units**, it happens at the beginning of the year **January 2**. This value can be explained because there are no inventory at the beginning of the year so it is critical to replenish the inventory. Other than that, there are a few peaks where the demand rises between 5,000 units and 6,000 units during the year.
We can see that the DC throughput is not linearly correlated to the FC demand. However, it is less constraining for the Distribution Center it has to ship to the 15 FC, though owning 15 fulfillment centers appears more costly for the group.

TASK 10

Definition of the Optimization Problem

For this task, the goal is to optimize the following parameters in the `algorithm()` function:

- `replenishment_i`: the interval between two replenishment
- `r_a_days`: robust autonomy threshold days (number of days before the inventory is empty without replenishment - 95% certainty)

Do compute this optimization algorithm, we modified slightly the `algorithm()` function so that when:
`replenishment_amount` == 34 (10% TL), we add +1 to a counting value, in our case `number_low_refill` += 1.
We considered different ways to solve this optimization problem:

- by brute force: iterating through all of the fulfillment centers, iterating through all the replenishment interval and through all the desired days for a robust autonomy. It is easy to implement but extremely time consuming and requires a fair amount of computational power. Therefore, we ran this simulation for only a couple of given values for the replenishment interval and the robust autonomy threshold: [5,8,10,20,50,75,100].
- by using python optimization libraries, such as `scipy.optimize`. The benefits is that it requires way less computational power, but is harder to implement.

Brute force Method

For 4 FCs using the brute force method, we get the following results:

Table 6: Fulfillment Center Data

FC	replenishment_t	best_replenishment_interval	best_r_a_days	min_value
GA-303	0	5	5	0
NY-134	2	5	5	0
TX-799	3	5	5	0
UT-841	4	5	5	0

For 4 FCs, the flow is very important, therefore the LTL are reduced.
We find the same results for 15 FCs. This showcases the limits of the brute force method, which fais to give precise results.

`scipy.optimize` minimization method

For a 4FC strategy, the optimal results are:

- **Fulfillment Center: GA-303**
  - Optimal r_a_days: 30
  - Optimal replenishment_i: 7
  - Optimal value: 0
- **Fulfillment Center: NY-134**
  - Optimal r_a_days: 30
  - Optimal replenishment_i: 7
  - Optimal value: 0
- **Fulfillment Center: TX-799**
  - Optimal r_a_days: 30
  - Optimal replenishment_i: 7
  - Optimal value: 0
- **Fulfillment Center: UT-841**
  - Optimal r_a_days: 30
  - Optimal replenishment_i: 7
  - Optimal value: 0

We can see that the optimal parameters are the same for all the fulfillment centers. Let's note that they vary from the brute force method, but this can be explained that there are not one unique solution to this optimization problem. Furthermore, the fewer the FCs, the more they need restocking. Having few FCs tend to augment the flow, thus reducing the LTL amount.

The results for the 15 FCs are the same than for the 4FCs, and this for both optimization methods. It seems that a deeper analysis should be done (perhaps by comparing with another optimization algorithm), and simulations should be computed in order to determine the best model.