

Western Kentucky University

TopSCHOLAR®

Masters Theses & Specialist Projects

Graduate School

Fall 2020

Supply Chain Analysis to Determine E-Commerce Distribution Center Locations

Fatima Chebchoub

Western Kentucky University, fatima.chebchoub768@topper.wku.edu

Follow this and additional works at: <https://digitalcommons.wku.edu/theses>



Part of the [Computer and Systems Architecture Commons](#), [Data Storage Systems Commons](#), [E-Commerce Commons](#), [Management Information Systems Commons](#), and the [Operations and Supply Chain Management Commons](#)

Recommended Citation

Chebchoub, Fatima, "Supply Chain Analysis to Determine E-Commerce Distribution Center Locations" (2020). *Masters Theses & Specialist Projects*. Paper 3465.
<https://digitalcommons.wku.edu/theses/3465>

This Thesis is brought to you for free and open access by TopSCHOLAR®. It has been accepted for inclusion in Masters Theses & Specialist Projects by an authorized administrator of TopSCHOLAR®. For more information, please contact topscholar@wku.edu.

SUPPLY CHAIN ANALYSIS TO DETERMINE E-COMMERCE DISTRIBUTION
CENTER LOCATIONS

A Thesis
Presented to
The Faculty of School of Engineering and Applied Sciences
Western Kentucky University
Bowling Green, Kentucky

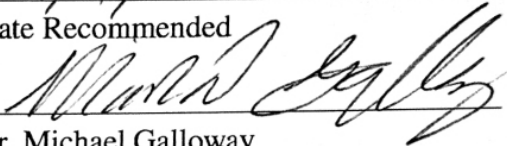
In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

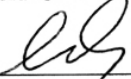
By
Fatima Chebchoub

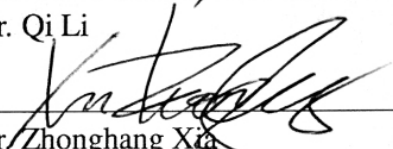
December 2020

SUPPLY CHAIN ANALYSIS TO DETERMINE E-COMMERCE DISTRIBUTION
CENTER LOCATIONS

11/4/2020
Date Recommended


Dr. Michael Galloway


Dr. Qi Li


Dr. Zhonghang Xia



Associate Provost for Research and Graduate Education

I dedicate my dissertation work to my family and my friends. A special feeling of gratitude to my loving husband Lhousseine Guerwane for his support and words of encouragement. Without him I would not have been able to complete this research, and without him I would not have made it through my Master's degree! I will always appreciate all he had done and being my best cheerleader. To my parents, my sister, and my brothers who have supported me throughout my Master's Degree studies.

ACKNOWLEDGMENTS

I wish to thank my committee members who were more than generous with their expertise and precious time.

A special thanks to Dr. Michael Galloway, my committee chairman, for his countless hours of reflecting, reading, encouraging, providing guidance and feedback, and most of all his patience throughout the entire project. Also, for being my mentor throughout my Master's Degree. I have never had a professor like Dr. Galloway, his dedication to his students is contagious; he helped me to apply for conferences and grants. In addition, Dr. Galloway created the Cloud lab to provide computing resources for his students' projects, we often meet weekly to discuss research paper and computer science topics. Thank you very much Dr. Galloway for all the sacrifices you made to meet with me with my crazy work and school schedules. The Students at the Computer Science Department are lucky to have you as their professor.

Thank you also to Dr. Li Qi and Dr. Zhonghang Xia for agreeing to serve on my committee.

Finally, I would like to thank the rest of my professors in the Computer Science Department at Western Kentucky University for everything you have done for me.

CONTENTS

1	INTRODUCTION	1
2	BACKGROUND AND LITERATURE REVIEW	4
2.1	Components	4
2.1.1	Big Data and Hadoop	5
2.1.2	Docker Containers	8
2.1.3	Docker vs. Virtual Machines	10
2.1.4	Kubernetes	11
2.2	Commercial Software for Supply Chain	14
2.3	Supply Chain	15
2.3.1	Supply Chain Introduction	15
2.3.2	Supply Chain and the Pandemic	20
3	EXPERIMENTAL DESIGN	21
3.0.1	Data Set	21
3.0.2	Methodology:	23
3.0.3	R implementation	24
4	EXPERIMENTAL RESULT	25
4.1	CASE STUDY OF VISUALIZATION USING MAPS	25
4.2	CASE STUDY OF VISUALIZATION USING GRAPHS	33
4.2.1	Bar graphs	33
4.2.2	Pie graphs	47
4.2.3	Dots graphs	51
5	NETWORK	53

5.1	Network Modeling	53
5.1.1	What is Network?	53
5.1.2	Represent a network in R	55
6	CONCLUSION AND FUTURE WORK	61
6.1	CONCLUSION	61
6.2	FUTURE WORK	62
	REFERENCES	63
	APPENDICES	65
A	APPENDIX A	66
A.1	R and RStudio	66
A.1.1	Install and Download R	66
A.1.2	Install and Download RStudio	67
B	APPENDIX B	73
B.1	Datasets	73
B.1.1	First Dataset	73
B.1.2	Second Dataset	74
B.2	Scripts	75

LIST OF FIGURES

2.1	MapReduce Architecture. [Shvachko, Kuang, Radia, and Chansler, 2010]	7
2.2	MapReduce Architecture. [M. Dhavapriya, 2016]	7
2.3	Docker Architecture. [Container, 2013]	8
2.4	Virtual Machines. [Container, 2013]	11
2.5	High-level diagram of a master. [Schenker, 2018]	13
2.6	List of Available Commercial Software Tools for Supply Chain Management. [Funaki, 2009]	16
2.7	Supply Chain Network. [Niki Matinrad, 2013]	18
4.1	United States of America Map	26
4.2	USA Map with Facilities	27
4.3	USA Map with Quantities	28
4.4	DC's with Quantities	29
4.5	Top 100 Customer locations from one DC	30
4.6	Low 100 Customer locations from one DC	32
4.7	Total Order for each State	34
4.8	Low 100 Customer locations from one DC	35
4.9	Bar graph with more clear data	36
4.10	Total Orders for Top 20 States	37
4.11	Total Orders for Top States Sorted	38
4.12	Total Orders for Low 20 States	39
4.13	Total Orders for Low 20 States Sorted	40
4.14	Total Quantity for each State	41

4.15	Total Quantity for each State	42
4.16	Total Quantity for Top 20 States	43
4.17	Total Quantity for Top 20 States Sorted	44
4.18	Total Quantity for Low 20 States	45
4.19	Total Quantity for Low 20 States Sorted	46
4.20	Pie Graph of the Top 10 states	48
4.21	Pie Graph of the Top 10 states (Percentages)	49
4.22	3D Pie Graph of the Top 10 states	50
4.23	Distances between one DC and each State	52
5.1	An undirected graph with 10 and 11 edges. [graph, 2019]	54
5.2	A directed graph with 10 vertices (or nodes) and 13 edges. [graph, 2019]	54
5.3	Network Between DCs and Cities	56
5.4	Network Between DCs and Cities with States	57
5.5	Network Between DCs and States	58
5.6	Network Between DCs and States	59
5.7	Network Between DCs and States	60
A.1	Download R	66
A.2	Download R	66
A.3	Download R	67
A.4	Download R	67
A.5	Download R	67
A.6	Install R	68

A.7	Install R	68
A.8	Install R	69
A.9	Install R	69
A.10	Download RStudio	70
A.11	Download RStudio	70
A.12	Download RStudio	71
A.13	Download RStudio	71
A.14	Download RStudio	72
B.1	Data Set Script	75
B.2	Data Set Script	76
B.3	USA Map Script	76
B.4	USA Map with Facilities Script	77
B.5	USA Map with Quantities script	77
B.6	DC's with Quantities Script	77
B.7	Top 100 Customer locations from One DC's script	78
B.8	Low 100 Customer locations from one DC	79
B.9	Low 100 Customer locations from one DC	80
B.10	Top 20 and Low 20 states orders script	81
B.11	Total Quantity for each State Script	82
B.12	Total Quantity for each State Script	82
B.13	Pie Graphs Script	83
B.14	Distances between one DC and each State Script	84

B.15 Network Between DCs and Cities Script 85

SUPPLY CHAIN ANALYSIS TO DETERMINE E-COMMERCE DISTRIBUTION
CENTER LOCATIONS

Fatima Chebchoub

December 2020

86 Pages

Directed by: Dr. Michael Galloway, Dr. Li Qi, Dr. Zhonghang Xia

School of Engineering and Applied Sciences

Western Kentucky University

Supply chain management is the key success for each business. Having a robust supply chain will help the business to improve service, quality, reduce the costs, improve the speed, and be more flexible. Organizations need to look at data and deploy plans to move the product from operation to logistics to manage a global chain. In this paper we will use E-Comm shipment data to identify the best locations to build a new distribution center (DC).

Chapter 1

INTRODUCTION

Businesses and consumers will agree that getting the right order, the right time and the right place is the dream. To achieve this goal, we use supply chain management. The businesses will have strategies, plans and goals to make the customers happy. From a consumer point of view, we have options to where we go to get our needs which makes it more comparative for most of businesses, that is why companies will look for best choices for their customer when they buy their suppliers. When it comes to best choice it hard to define, some customers will look for the best quality product, some will just need the least expensive one. So, the biggest challenge for a company is to define what is the best choice for their customers based on their product they sell. After the company buys their suppliers, now it is the time to produce a product, the big goal of companies is to make the operation operate every day quickly and operate the right way. Businesses have to make revenue by selling more, to sell more they have to keep the operation moving and delivering goods to customers. The transportation plays a big role on a company supply chain, after making the product/service, now is the time to deliver it to the consumers/customers. Companies start by packaging the product, load it to the transportation. Also, companies will make a decision if they need to transport the good or the consumer will pick up the good, or maybe there is a third party that will take care of the transportation. It is all based on how fast they want to deliver the product. Is it a product that can be stored in a warehouse to be

delivered later? How much the company should produce? How long should they keep it? All these questions and more will lead to a good supply chain management. One supply management can be great to one company and it may not work for a different one, which makes the topic of supply chain management very hard to discuss and analyze because each business is doing things differently, even within the same type of business. Also, businesses are very sensitive when it comes to this topic of supply chain management, they want to keep their tricks to themselves, and don't want to share their way of doing things because of the massive competition between the businesses. In this project, I will address the issues of supply management, and how to make the supply management more efficient. In this project I will try to help with a problem with a major apparel manufacturing company. The Company want to build more Distribution centers around US but not sure where will be the best place to build a new distribution center using shipment data. First, in this project I am going to analyze the shipment data, I will do a data cleaning, which include correcting or deleting corrupt records on my data set. Second, we will create an algorithm that will solve or answer the question of this project with is "where is the best location to build a distribution center for a major apparel manufacturing company".

After that I will divide the data set to a training data set to train the algorithm and a validation data set to validate and test the result of the algorithm. After analyzing, training and validating the algorithm, I will move to implement the project. To do that, I will create an application that will have an interface, which will make it easy for the users to use. They will have to load a new data set of shipment records, also some basic information such as, entering the current location of the distribution center, it can be one or multiple distribution centers across the United States. The application will take the information the user enters,

and the data set file, then it will show a map with the current DC's location, and will predict distribution center(s) simultaneously, based on the shipment data.

The aim of this project is to identify the best locations for distribution centers using a data analysis, which will lead to determine the best location for a business to have an e-Commerce DC based on shipment data.

Scope

Identify the best place to build a new distribution center using shipment data to meet the future needs of the business. So, in this project I will have two parts. The first part is the analysis part and the second part will be the implementation where I will create a tool for the end users to use to do analysis with other data sets, anytime they want. The end user want a tool that looks like IImasoft [IImasoft, 1998] and ArcGIS [ArcGIS, 1999]. To achieve these requirements, I will need to read work related to the topic of supply chain management, and network optimization algorithms. Moreover, I will also discuss a couple of architectures and software's tools that I will compare in this paper to decide which one to use and which one will be more beneficial for the users to use. First, let us discuss the architectures component of the project, what hardware / network components are available to use. Also, which software to use that will help solve the problem. Second, I will move to related works.

Chapter 2

BACKGROUND AND LITERATURE REVIEW

To perform a large-scale analysis, we often require the availability of a massive number of computers. To address this need I will be using The Cloud Computing Lab at Western Kentucky University which I will have the option to use a Cluster. Cloud computing provides scientists with a completely new model of utilizing the computing infrastructure. Compute resources, storage resources, as well as applications, can be dynamically provisioned. Cloud computing, the long-held dream of computing as a utility has gradually made software even more attractive as a service and shaping the way IT hardware is designed and purchased [Patil, Navada, Peshave, and Borole, 2011].

2.1 Components

After we listed the background, the scope, and what hardware we have, the next part of this literature study project, will be to decide which packages and components I will need to install in to the Cluster. I read a couple of research papers that talk about different architectures, which I will be analyzing, surveying and comparing those architectures to each other. When analyzing Big data most people will prefer to use Hadoop. However, Docker also becoming a hot topic and a lot of people will use it. So, which one is the best solution for this project? and how are they similar and different? To answer these questions, I have two research papers one talks about Big data Analytic using Hadoop

[Shvachko et al., 2010] and the other one is an Introduction to docker and analysis of its performance [Baback Bashari Rad, 2017].

2.1.1 Big Data and Hadoop

The authors of [Bijesh Dhyani, 2014] present the basic understanding of big data and how to organize data from performance prescriptive. Big data is a collection of data that can be processed using computers. The term big data is referring to where we have big size volume of data, the data is complex, and it grows fast. Data is always a big thing that companies and people try to collect. We wanted to store as much as we could. That is why the companies Like Oracle and Microsoft were always trying to create tools to manage these data like relational databases. Recently, we see massive data available to us to use. All these data were generated using smart phones, social media websites, machines and other resources.

So, switching from collecting data to data management is a big deal. The ability to extract information from data will help everybody perform better or learn quicker from this information. So, to extract the information we need tools that will handle the massive Data. Before we talk about the tools, what is Big data? and what was considered a big data five years ago, is it still considered big data now? The research paper [Bijesh Dhyani, 2014] define big data as “Big data is the availability of a large amount of data which become defect to store process and mine using a traditional database primarily because of the data available is large, complex, unstructured and rapidly changing”. There are a couple of challenges that will be phasing anyone using big data. [Bijesh Dhyani, 2014] defines those challenges as below:

- **Volume:** the amount of data which is the size of the dataset.
- **Variety:** which presents the format of the dataset. Some data will be text, audio, log files...etc. for my project the dataset will be a text format.
- **Velocity:** speed of data processing (rate of growth) it makes it hard to manage and process. This will be the main issue for my project, my data set will have a thousand of shipping data everyday which will make it difficult to capture and analyse.
- **Value:** The value of the data is huge to the companies. Data will be a great source for businesses to help with increasing their performance and learn from their mistakes.
- **Veracity:** when we are dealing with a high volume there is always the chance to have noise. In my case the data that I have is a shipment data that contains transactions, I will do a cleaning phase which I will delete all the corrupt transactions and inaccurate records that do not represent the data set.

After we define big data, and its challenges, what are the tools that will help reduce these challenges and help process and analysis the data? the answer to this question is Hadoop.

Hadoop provides a distributed file system and a framework for the analysis and transformation of very large data sets using the MapReduce paradigm. An important characteristic of Hadoop is the partitioning of data and computation across many (thousands) of hosts and executing application computations in parallel close to their data.

Since we are using large data sets, Hadoop will divide the data to small chunks and map to different computers, then run algorithms (MapReduce) to process the data and

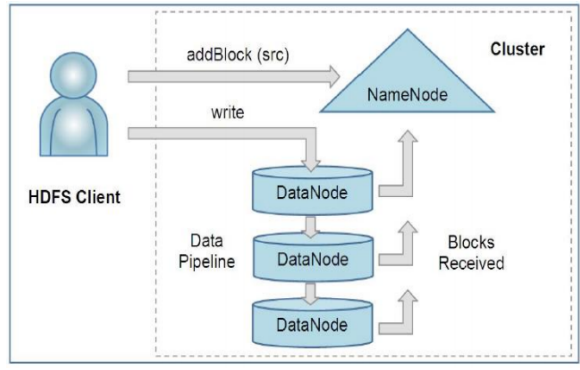


Figure 2.1: MapReduce Architecture. [Shvachko et al., 2010]

return the results from all computers all together to produce one result. MapReduce is widely been used for the efficient analysis of Big Data. Traditional DBMS techniques like Joins and Indexing and other techniques like graph search is used for classification and clustering of Big Data [M. Dhavapriya, 2016].

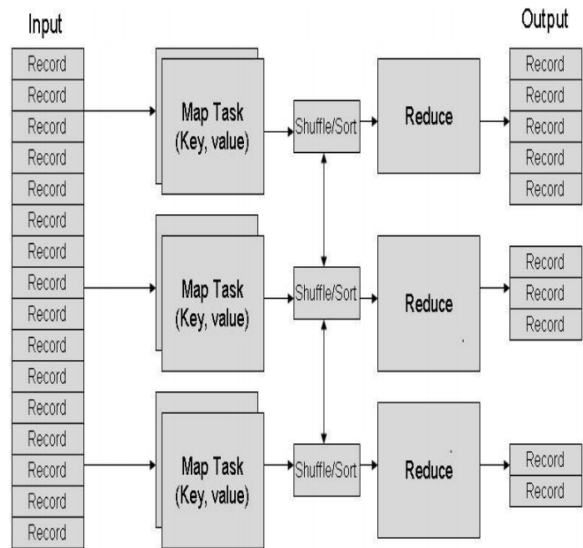


Figure 2.2: MapReduce Architecture. [M. Dhavapriya, 2016]

So, we are processing data in parallel so the result will be faster instead of running a chunk after another one which will take time to process.

2.1.2 Docker Containers

Docker is an open source platform that runs applications and makes the process easier to develop and distribute. The applications that are built in the docker are packaged with all the supporting dependencies into a standard form called a container [Baback Bashari Rad, 2017].

Based on [Container, 2013], containers are isolated from each other and bundle their own tools, libraries and configuration files. They can communicate with each other through well-defined channels. All containers are run by a single operating system kernel and are thus more lightweight than virtual machines. Containers are created from images that specify their precise contents. Images are often created by combining and modifying standard images downloaded from public repositories.

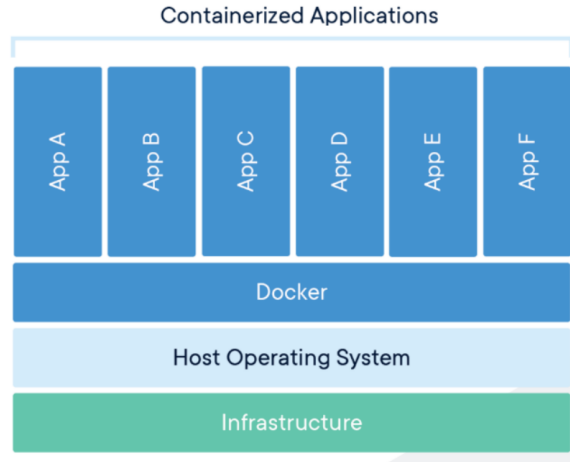


Figure 2.3: Docker Architecture. [Container, 2013]

Docker was designed to give a quick and lightweight environment to run and test the code more efficient before the code is being moved to production.

[Baback Bashari Rad, 2017] listed the four main internal components of Docker:

1. **Docker Client and Server:** docker can function as a Client/Server application. The server side will get a request from the client side and then process. The Client/ Server can be run on the same machine or different machines and connected with a remote server.
2. **Docker Images:** As I listed before we use the image to create the docker container. The image can be created using two methods. First method is the Base image, the image requirement can be added to the image by modifying it. The second method is the docker file, an automatic way to create an image. The docker file is just steps of command to follow to create the image. the commands will run from the bash terminal.
3. **Docker Registries:** we place the docker images in the docker registries. It's like a code repository where we can push and pull images from. Docker Hub is the public registry to where everyone can push images and pull available images from.
4. **Docker Container:** Containers hold the whole kit requirement for an application, so we can isolate and run the application.

[Baback Bashari Rad, 2017] also listed the advantages and disadvantages of Docker:

Advantages:

- **Speed:** speed is the main advantage of docker (highlighted). It's faster to build a container because it's too small.
- **Portability:** All the applications that are built inside the docker are portable and easy to move without affecting the performance.

- **Scalability:**Docker can be deployed in several platforms, like cloud, physical and data server. Moreover, docker can be moved easily between cloud and local host.
- **Rapid Delivery:**Docker container can work in every environment as they have the dependency applications which make docker easy to deliver. So, the programmer can predict a good result when moving the code from test to prod.
- **Density:**Docker does not use the hypervisor that is why it is more efficient. Also, the performance is higher.

Disadvantages:

- There is no complete virtualization because docker uses Linux Kernel.
- Docker does not run on an old machine.
- The performance and integration of the docker is not good with Windows and Mac Operating Systems.
- Security issue.

Docker uses a single host compared to virtual machines. What are other similarities and differences between docker and virtual machine?

2.1.3 Docker vs. Virtual Machines

To make an operating system easy to be deployed, we use virtual machine, which is an image copy of a specific operating system. Virtual machine uses an extra layer between the host operating system and guest operating system. This layer is known as a Hypervisor 2.4. However, docker does not use that extra layer, it uses another layer known as Docker engine and also docker has no guest operating 2.3.

When it comes to performance docker perform better than virtual machine in term of time and speed [Baback Bashari Rad, 2017]. However, virtual machine seems to be faster when it comes to execution. Virtual machine can help with making an elastic system and share resources among guest operating systems but virtual machine struggle with running big data workload [Zhang, Liu, Pu, Dou, Wu, and Zhou, 2018].

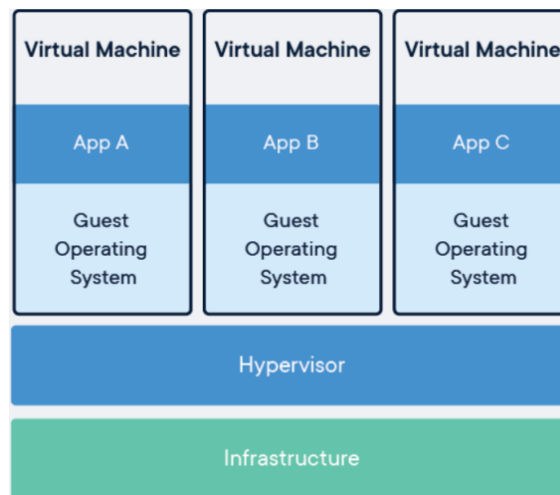


Figure 2.4: Virtual Machines. [Container, 2013]

2.1.4 Kubernetes

Kubernetes is a tool to install and manage Docker containers, Kubernetes was developed by Google, it uses the Docker images and deploys applications into the containers. The advantage of using Kubernetes first is that the containers are easily scaled up, we can eliminate and remade. The second reason is when compare Kubernetes with virtual machines, Kubernetes containers are deployed faster, more efficiently and reliably [Schenker, 2018].

The paper [Dongmin Kim, 2019] talks about how the organizations made the decision to transmit from doing all the work by building and managing their own comput-

ing facility to cloud computing have been benefiting from maximized capacity and cost-efficiency. One of the tools that they use is Kubernetes because it is an open-source container orchestration and easy to use because you need to feed some configurations to Kubernetes and the Kubernetes services will take these configurations and run the configurations. The service for example can create Docker images containers and Kubernetes can deploy and manage the components and their relationships.

The service for example can create Docker images containers and Kubernetes can deploy and manage the components and their relationships. the paper [Dongmin Kim, 2019] listed the elements of Kubernetes:

Kubernetes pod: this is an essential building block of Kubernetes, and it's the smallest unit of deployment. Usually containing one or multiple Docker containers.

Kubernetes node: this represents a VM or physical machine where the Kubernetes pods are run.

Kubernetes cluster: this consists of a set of worker nodes that cooperate to run applications as a single unit. Its master node coordinates all activities within the cluster.

The paper [Schenker, 2018] explains in details Kubernetes master nodes, which are used to manage a Kubernetes cluster. The following 2.5 is a high-level diagram of such a master:

The master nodes only run on Linux such as RHEL, CentOS, and Ubuntu. On the Linux machine, we need to have the following four Kubernetes services running [Schenker, 2018]:

API server: This is the gateway to Kubernetes. The API server is the embodi-

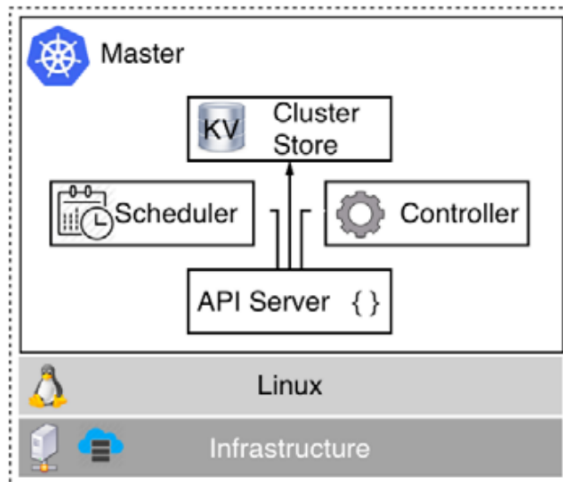


Figure 2.5: High-level diagram of a master. [Schenker, 2018]

ment of the Kubernetes control plane. This service will list, create, modify, or delete any resources in the cluster of all the requests.

Controller: The controller, or more precisely the controller manager. The controller manager contains the replication controller, the pod controller and services controller. All these controllers try to control loop that observes the state of the cluster via the API server and their job, then will make changes, attempting to move the cluster or effective state toward the desired state.

Scheduler:The scheduler is a service that tries its best to schedule pods on worker nodes considering various boundary conditions, such as the following:

- Resource requirements
- Service requirements
- Hardware/software policy constraints
- Quality-of-service requirements

- Data locality

Cluster store: This is an instance of etcd that is used to store all information about the state of the cluster. The Etcd is a highly reliable distributed data store. Kubernetes uses it to store the entire cluster state [Sayfan, 2017].

As a conclusion of the related work for a different architectures' components, we have decided to use Docker containers to implement the project. Docker is a free software used to create virtual containers and Docker container is more lightweight than virtual machines. Docker container will guarantee that my tool will run the same way in any environment. The Docker image will have all the dependencies that the tool will need to run, which will make it easy to implement on major apparel manufacturing company hardware. Also, Docker container allows isolation of component within one system, so only the components that need to access the other component will. As a result, we will have an easy way to manage the security of the system.

2.2 Commercial Software for Supply Chain

Before I start the modeling and algorithms that supply chain management used, let us see what tools are available for companies today that will help them with the supply chain. The paper [Funaki, 2009] listed the commercial software tools for supply chain design and their vendors. Figure 2.6 shows the list of currently available commercial software tools for supply chain design and their vendors together with years of release and web sites. These commercial softwares are available in the US market and they are considered the top software for supply chain management. The list included the tools that does not need to support their main software such as ERP, SCP and APS. The only two exceptions are "In-

for's Strategic Network Design and i2's Supply Chain Strategist, because they can provide supply chain design solution independently from their main products." [Funaki, 2009].

The paper [Funaki, 2009] was done in 2009, therefore I had to check if they still exist now. All the commercial software still exists, except the first software "CAST" which is now part of the LLamasoft software now. The website links changed for some of the software. Most of these software vendors charge licenses, consulting fees when selling their product. Which is the norm for most software companies [Funaki, 2009]. Compared to other software, we see a slow growth, because the companies do not use or buy supply chain management tools. It is mainly due to the expensive supply chain management software, and not many companies believe that it will solve their problems, but it will always need other dependencies, so that is why they cut the expense by not buy the tool.

To solve that problem many companies, start to charge by services, as [Funaki, 2009] listed "Some vendors who recognized this gap between the license based schema and the user's perception are starting pay-per-use services".

2.3 Supply Chain

2.3.1 Supply Chain Introduction

In this section, we will try to understand the Supply chain management and how it is important to the companies and the world. So, what is Supply chain? According the research paper [Nelson Dzipire, 2014] supply chain is "A set of facilities, suppliers, customers, products and methods of controlling inventory, purchasing and distribution" the goal of the supply chain is to "enhance the operational efficiency, profitability and competitive position of a firm and its supplier chain partners" [Nelson Dzipire, 2014].

Name	Vendor	Year released	Web site
CAST	Barloworld Optimus	1989	http://www.barloworldoptimus.com/home.aspx
4flow vista	4flow	2001	http://www.4flow.de/logistikberatung/4flow-vista.html
LogicNet Plus	ILOG/IBM	1995	http://www.ilog.com/products/logicnet-plus-xe/
LOPTIS	Optimal Software	N/A	http://www.ketronms.com/loptis.shtml
NETWORK	Supply Chain Associates	1968	http://SupplychainAssoc.com/NETWORK.htm
Opti-Net	TechnoLogix Decision Sciences	1993	http://www.technologix.ca/solutions/optinet_supplychain.htm
PowerChain Network Design	Optiant	2000	http://www.optiant.com/content/blogcategory/72/119/
PRODISI SCO	Prologos	1985	http://www.prologos.de/English/Prodisi.htm
SAILS	Insight	1984	http://www.insight-mss.com/_products/_sails/
SITELINK	CGR Management Consultants	1995	http://www.cgrmc.com/index.html
Strategic Network Design	Infor	N/A	http://www.infor.com/solutions/scm/strategicnetworkdesign/
Supply Chain Guru	LLamasoft	1998	http://www.llamasoft.com/index.html
Supply Chain Strategist	i2 Technologies	N/A	http://www.i2.com/solutions/solution_library/supply_chain_strategist.cfm

Figure 2.6: List of Available Commercial Software Tools for Supply Chain Management. [Funaki, 2009]

The paper [Niki Matinrad, 2013] describes the four entities of the supply chain. Suppliers, distribution networks, manufacturers, and customers are the four entities of a supply chain. The connection between each entity is not easy given today's competition even between these entities. So, supply chain must consider the "interactions and limitations between these elements and also consider operating factors, constraints and the dynamics in the market, such as changes in demand" [Niki Matinrad, 2013]. Companies need to know when to buy supplies, where to store it. What happens if there is no demand and the company ordered many supplies, and does not have a place to store it? How about order supplies only when customer make an order, in that case will the company has

enough time to produce the product and ship it to the customer in a timely manner? The better the connection between the suppliers, distribution networks, and manufacturers the happier the customers will be, which means the customer is satisfied and becomes loyal to the company and will make more orders again from the company.

The figure below shows general supply chain network that includes three various levels of enterprises: retailers, distribution centers and plants [Niki Matinrad, 2013].

- **The supplier:** where the company buys the supplies. The supplier provides raw material, energy, services ...etc.
- **The Factory:** where the company transfer the raw material to a product or service the customer will use.
- **The Distribution Center:** where the products (goods) will be stocked in a warehouse or specialized building so we can redistribute to retailers or wholesalers, or directly to consumers.
- **Retailer:** where the product will be sold to the customer. It can be a store or online store like (Amazon, eBay...).
- **Customer:** a person or company that will receive the final product. It's the one that buy and use the product or service.

Each one of us is part of a supply chain every day. We buy products and services, we decide where to buy it from, and when to buy it. And because there are billions of other customers doing the same thing it makes it hard for companies and make the level of competition between companies are very high. In today's market no matter how good

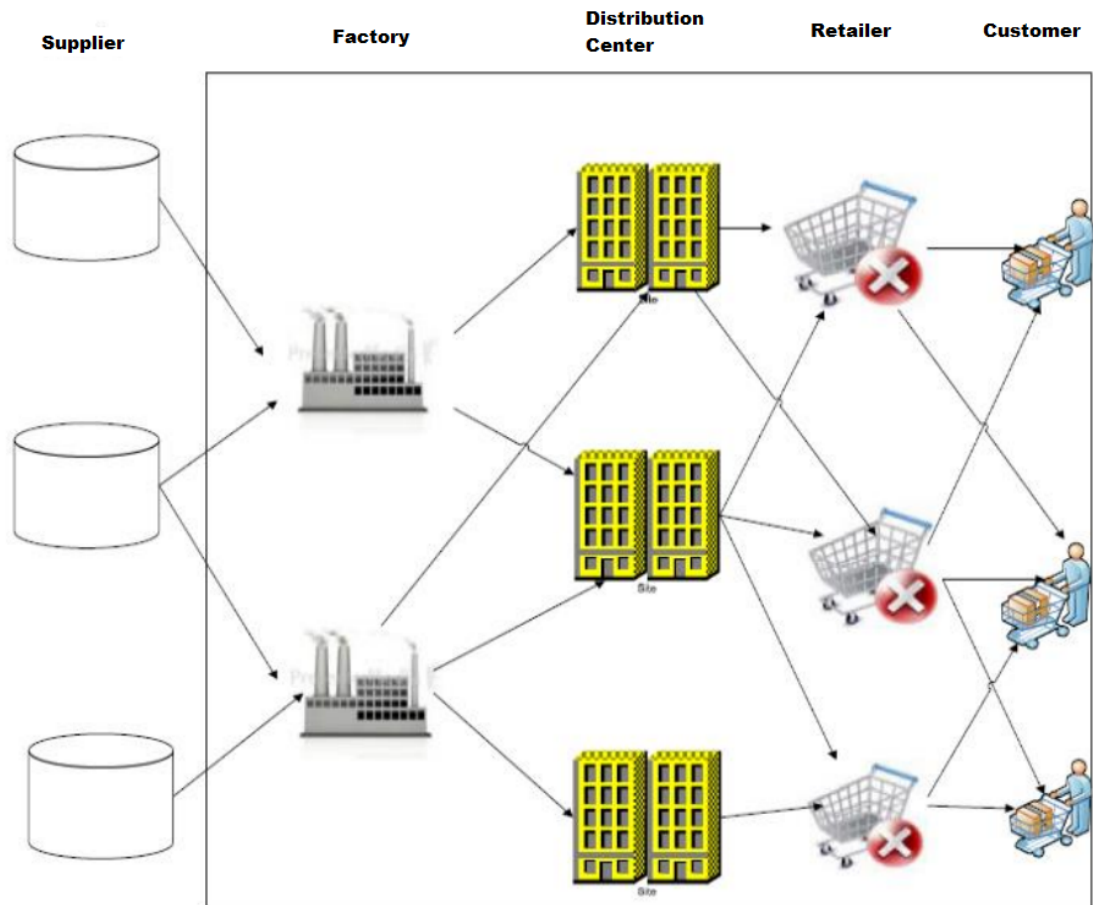


Figure 2.7: Supply Chain Network. [Niki Matinrad, 2013]

the product the companies make, it is only valuable until it gets to the customer and the customer likes it. So, supply chain has become extremely complex.

Companies struggle when they have to choose their suppliers. It is a huge responsibility, because the company is not relying on just the suppliers but also the suppliers of their suppliers. In today's global market, where materials come from all over the world, it's hard to manage the global chain of supplies without collaborations with all companies. When a company develops a plan on where to buy supplies and how many employees to hire, it is all based on the suppliers' companies that deal with and the relationship with these

companies. The stronger the relationship, the more the company can succeed on its supply chain. The companies will benefit a lot from supply chain management when they start aligning production strategy with all of the other supply chain entities. However, a good supply chain strategy must also align with the company strategy and help the company with decision making and its mission.

The paper [Nag, Han, and qing Yao, 2014] discussed how supply chain is a critical component of business strategy and listed the three-supply chain inventory strategy:

Fisher's model and its variations: This supply chain inventory strategy is based on supply lead time variability, product characteristics and demand variability. This model is based on stable product and predictable demand, high competition and low-profile margin [Nag et al., 2014].

Lean and agile paradigms: This supply chain inventory strategy combined lean and agile paradigm; lean manufacturing is used with level scheduling upstream and agile paradigm is adopted to meet market demand [Nag et al., 2014]. The main difference between lean and agile is that lean will aim for low cost while agile will aim for high availability and responsiveness to changes in the product mix and volume [Nag et al., 2014].

Push, pull and push-pull systems: This supply chain inventory strategy is based on long term forecasts of market demand. The push system aims for desired customer level, on the other hand, pull system aims to respond to real time market demand [Nag et al., 2014]. So, for the pull supply chain, the company chooses not to produce anything until the customer make an order, in this case the customer will have the opportunity to request something that they really like, for example, the customer can choose the color, the materials, etc. However, push supply chain strategy knows the customer will need/use

a product/service so the company will produce the product / service before the customer makes any order. So, the product will be stored and waiting for the customer to make an order. The advantage of push supply chain strategy that will be fast to ship the product, but sometimes customers may not like the product and the company will end up with a product that they produce a high volume for it, but customers did not like it. Some companies choose to combine both push and pull systems when it comes to the supply chain strategy of their company. If they have more than one product, they will produce some products and have it ready for the customers, they will also have the option for the customers to request special product(s).

2.3.2 Supply Chain and the Pandemic

When we put all supply chain entities together, supply chain can be very complicated and expensive. When we add Pandemics like the one we experiencing now (Covid19) to the equation of all the entities, we can have a major influence on how supply chain work or at least it shows some companies that their supply chain strategy does not work or is not efficient enough to adapter to the changes or challenges that may face. Many companies during the pandemic experienced a broken supply chain, with people who could not get the basic needs , like food.

Chapter 3

EXPERIMENTAL DESIGN

People now prefer to buy everything online. That is why companies have to keep track of where their eComm customers are often. Where the customers are located, how far is that from their Distribution centers, how long it usually takes to deliver the goods to the customers? Do any customers cancel orders because it was not delivered on time? All these questions will show companies where they stand now when it comes to their eComm customers. Due to the fast growing of eComm websites, in this project I will do an analysis to determine the best location for a business to have an eComm DC based on shipment data.

Hypothesis: Determine the best location for a business to have an eComm DC that is based on shipment data, will reduce the cost of delivering goods to the customer and high service.

3.0.1 Data Set

3.0.1.1 First data set

My dataset in the appendix B.1.1 is a collection of information about retail and eComm transactions for a major apparel manufacturing company, with details information about each transaction, like date, quantity and customer location. The data is for all the transactions for the year of 2017.

The dataset in the appendix B.1.1 is for multiple brands for the same major apparel

manufacturing company. The dataset will have the data about all the brands and each brand will have different items they will ship. Each distribution center for the major apparel manufacturing company ships specific items. So, if DC1 ships item1 then we can only ship that item from the DC1. This is the way the company is doing the shipment of the items now, but there are some exceptions sometimes, they will ship the same item from two distribution centers for special events. The company also used Third-party logistics (3PL) to outsource elements of its distribution, warehousing, and fulfillment services. Because of the confidentiality of the dataset, I have created a program that will generate random zip codes for each customer, and also generate random quantity ships.

Because the data is a very sensitive data, I created an C Sharp script that will generate random data that will look the same as real data.

As you see in the appendix B.1, the script generate a random five digit number to present a zip code, the second col is for a random number between 1 and 100 to present the quantity that was ordered.

3.0.1.2 Second data set

To map the location of the customer based on the random zip code we need to convert the five random digits to a lat and long. To do that I create a R script that will take the data set that was created using the script on the the appendix B.1 and other data set B.1.2 for the zip code information and create a new dataset that will have more information about the location than just the zip code.

The data for the information about the zip code B.1.2 was downloaded from this website: [ZipCode, 2012]. The data is a free zip code database, the size of the data is 4.2 MB. The database was last Updated on 1/22/2012 and it is the primary location only.

3.0.1.3 *Final data set*

To merge the data from the first data set and the second data set, I created an R script. The script on the Appendix B.2 lists the script commands to read the two data set, one excel and one csv files, then it convert the data set to data frame, to merge the data we use the zip code col, after merging the data we will have a data set with both cols from the first data set one and the second data set. After merging the two data sets the final data can be saved as .Rdata or a csv file.

3.0.2 *Methodology:*

All these elements will make creating a large analysis, to determine the best location, hard. The project is limited to only ship some items (SKUs) from one Distribution center and the company also used 3PL. The challenge is to create a supply chain tool that will take data about customers location, item they ordered and quantity, plugin the data and show the best location to have a distribution center based on our customer location and demands (eCommerce and Retails). The tool at the end can be use by any company, they only need a dataset with the zip codes of their customers, the quantities and current DCs.

The main question I will answer on the first round of the implementation is the following:

- Ecommerce based on the number of shipment and customers location, where will be a good location for a new distribution center?

3.0.2.1 *Baseline:*

Current demand, shipping patterns, and network flows.

3.0.2.2 *Design and Scenarios:*

- Distribution network
- Optimized flows

3.0.3 R implementation

I will use the programming language R which is free software for statistics and graphics. I will be using R language to develop the statistical part of the project and also help with data analysis.

To visualize the data I will be using leaflet library for R, Leaflet is one of the most popular open-source JavaScript libraries for interactive maps.

For the other analysis parts of the project I will use many R libraries That will allows to build high charts/graphs in R.

Chapter 4

EXPERIMENTAL RESULT

4.1 CASE STUDY OF VISUALIZATION USING MAPS

Data visualization is using that powerful processes/mechanisms by create, manipulate and interact with representations of the data in a graphical way so we can get information from the data by visualizing the data and get better understanding in order to gain insight into the data instead of keeping the data as a form of tables, that will not give us the big picture that the users' needs.

In today's world companies have a lot of data stored about their business, customers, and products. All of that data does not make any sense for most people in the company that does not deal with that data or even the team that works with the data if it is big data. People's brains can process some data but when it comes to big data our brain has limited availability to process all that data at the same time. Using mathematical equations to understand the data can hide a lot of details and can lead to misleading or even wrong answers about the data. Also, we live in a visualize society where everyone processes data as a form of vitalization and not data in a form of numbers and database tables.

A good data visualization helps us tell a story to our broad audience and get more efficient and interaction feedback from our audience that they can see something in the data that they were not able to see before. A good data visualization with a good story will educate, motivate and make the user engage with the story and ask more questions

which will lead to better understanding of the data. A good story about the findings leads to an action that will benefit the user of the data by better understanding their customer, increasing sales . . . etc.

The best way to visualize the data of this project is by maps. Maps provide a powerful way to visualize data. They allow us to quickly place data in the context of geographic areas. Maps give us a great visualize image about the location of our customers, which area of the map has more sales.

The datasets we are using contain data about shipment and location where they were shipped. We also have two more datasets one contains the data about the facilities and another one has the location for the DC's.

By run the script B.3 we get the USA map below



Figure 4.1: United States of America Map

Figure B.4 shows the script to install and load the leaflet package, the script also maps the Lat and Long attributes for all the facilities using the leaflet library.

By run the script B.4 we get the USA map with Facilities

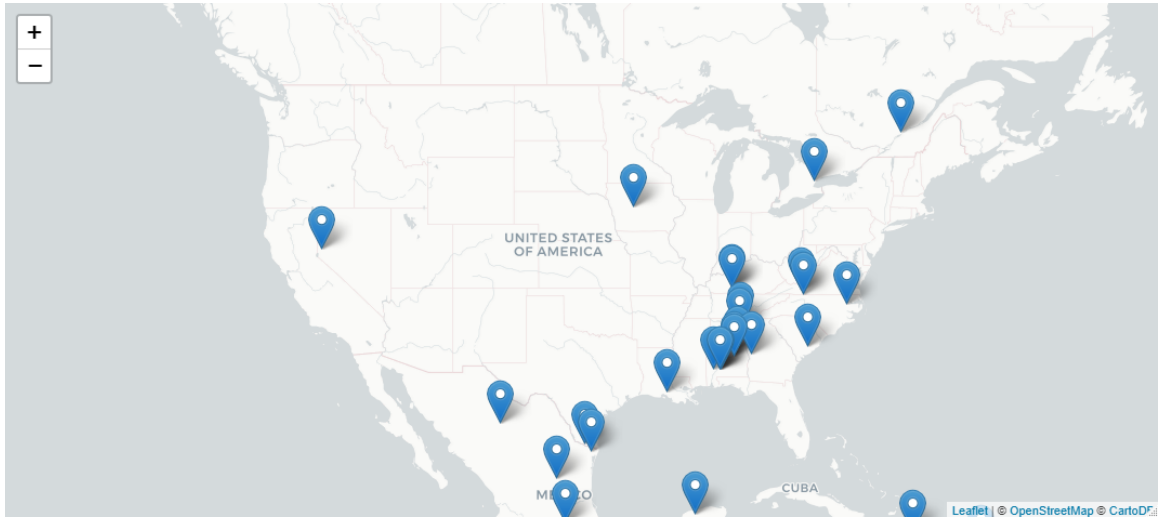


Figure 4.2: USA Map with Facilities

Now we have an idea where our facilities are located.

Now let show where most of the quantities are located. Figure B.5 shows the script to map all the orders locations with the quantities.

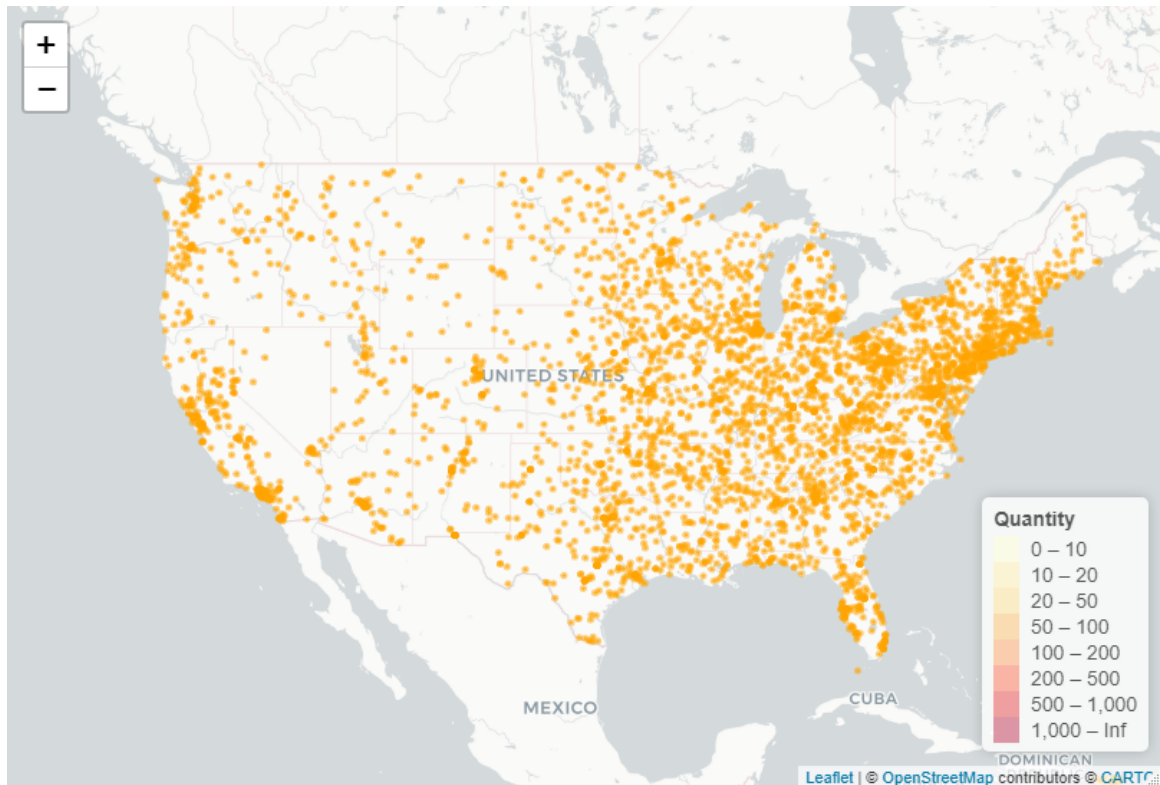


Figure 4.3: USA Map with Quantities

The figure 4.3 shows that most of the orders come from the East Coast of the United States.

To get a better understanding let add the current Dc's to the map B.6 and see where they are located. By adding DC's to the map we can see how far the Dc's from our customers.

The figure 4.4 shows that most of the DC's are in the Southeast of the country and most the orders for the products are coming from the Northeast and fellow by the Southeast. That means that the current Dc's are close to the region where we have most of the customers but not very close to the main region where we have the largest orders.

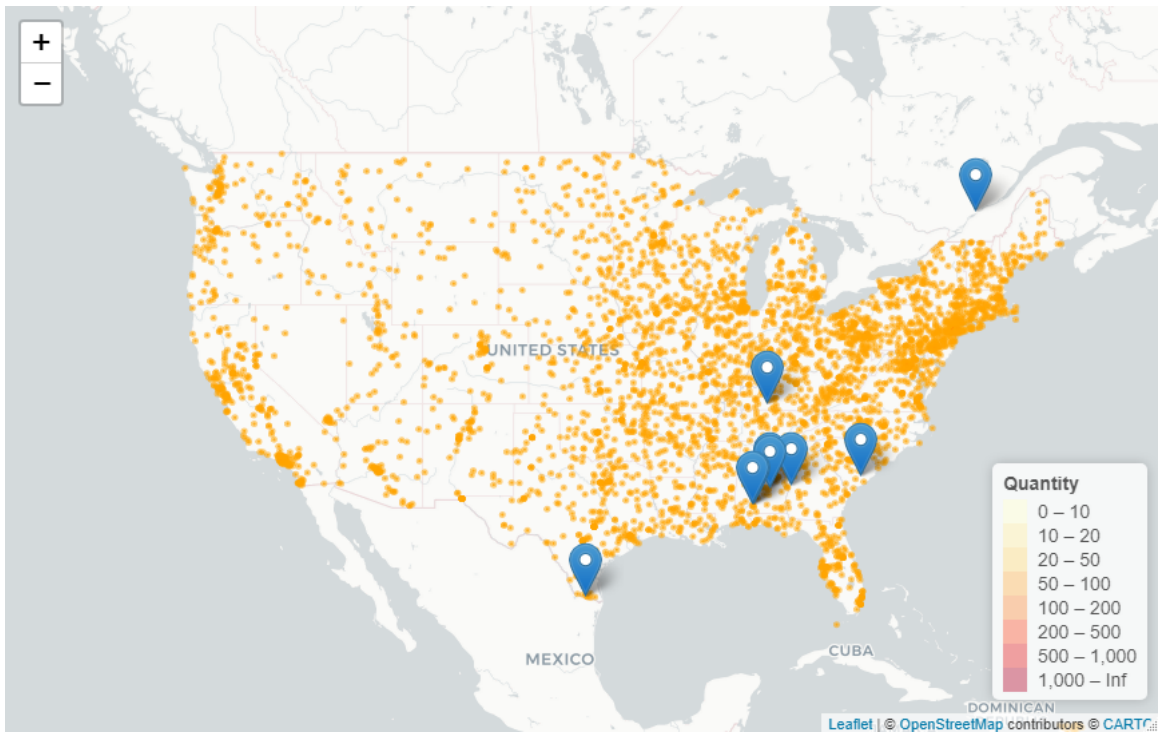


Figure 4.4: DC's with Quantities

In figure B.7 we have the code that will create lines between one DC and the top 100 customers (the customer that orders the biggest quantities) to see how far the DC is from our top customers.

The figure 4.5 shows how far the top 100 customers are from one of the main DC's. Many customers are in the Northeast of the country like how figure 4.4 showed we also see that there are few customers from the Northwest and southwest of the country.

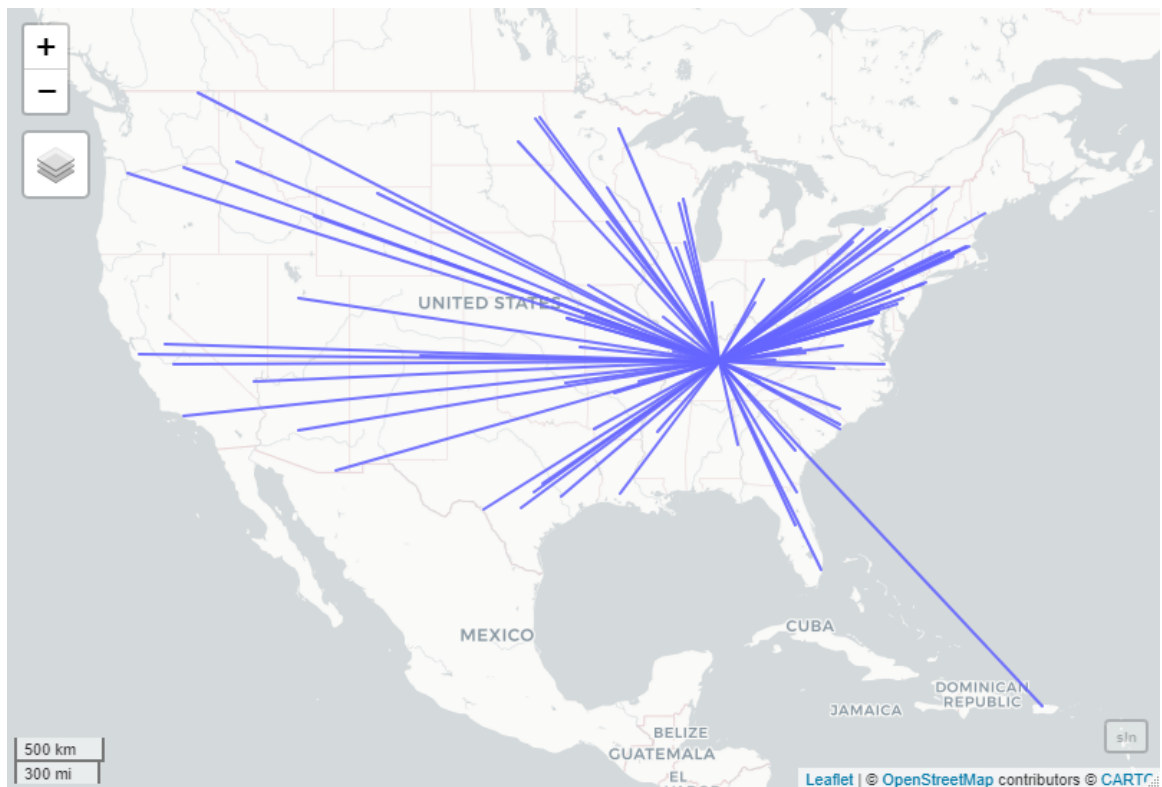


Figure 4.5: Top 100 Customer locations from one DC

Now we know where our top 100 customers are, let's see where our low quantity customers are? We will use the same script to generate the map but this time we will switch to a new DC and get the 100 customers with the lower quantities.

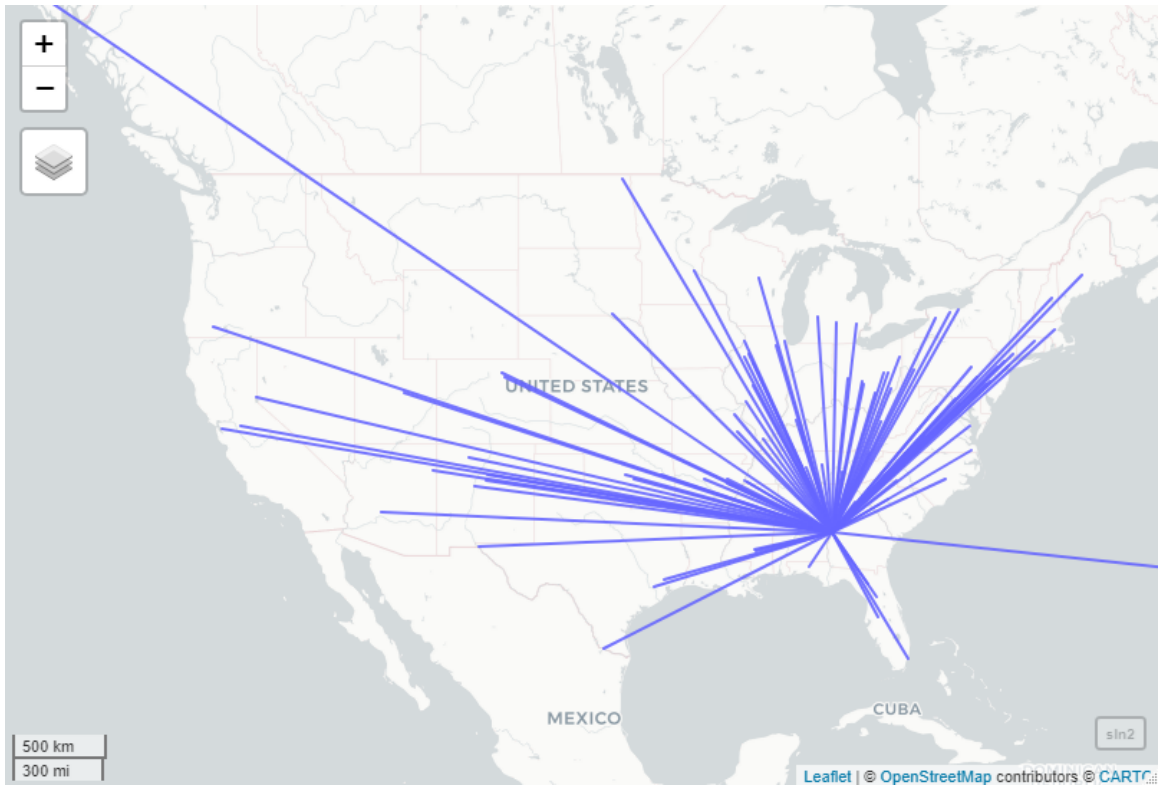


Figure 4.6: Low 100 Customer locations from one DC

The figure 4.6 shows the same result like the top 100 customers that our lower customers are also located in the Northeast and Southeast with fewer customers from the southwest (West Coast) this time, but we also notice that there are more customers close to the DC this time.

4.2 CASE STUDY OF VISUALIZATION USING GRAPHS

4.2.1 Bar graphs

In this case study we will use a variety of graphs. The first graph we will use is the Bar graph. Bar graphs are another common data visualization tool. In GG plot, we use two different functions to create bar graphs. `Geom bar` creates a bar graph where we specify the value that appears on the X axis and use the count of the number of rows matching that value as the Y axis value. `Geom col` creates a column graph, which, like bar graphs, allows us to specify the value on the X axis, but also allows us to specify a Y axis value manually. These two functions are very similar. They only differ in the fact that bars use count as the Y axis value, while columns allow us to choose the Y axis value. Let's try some examples in R.

In the example B.8 we have the code that will use the library “`ggplote2`” to create a graph with the total orders for each state. Before we plot the data, we start by grouping by the state and get the number of the orders.

By run the script B.8 we get the graph 4.7

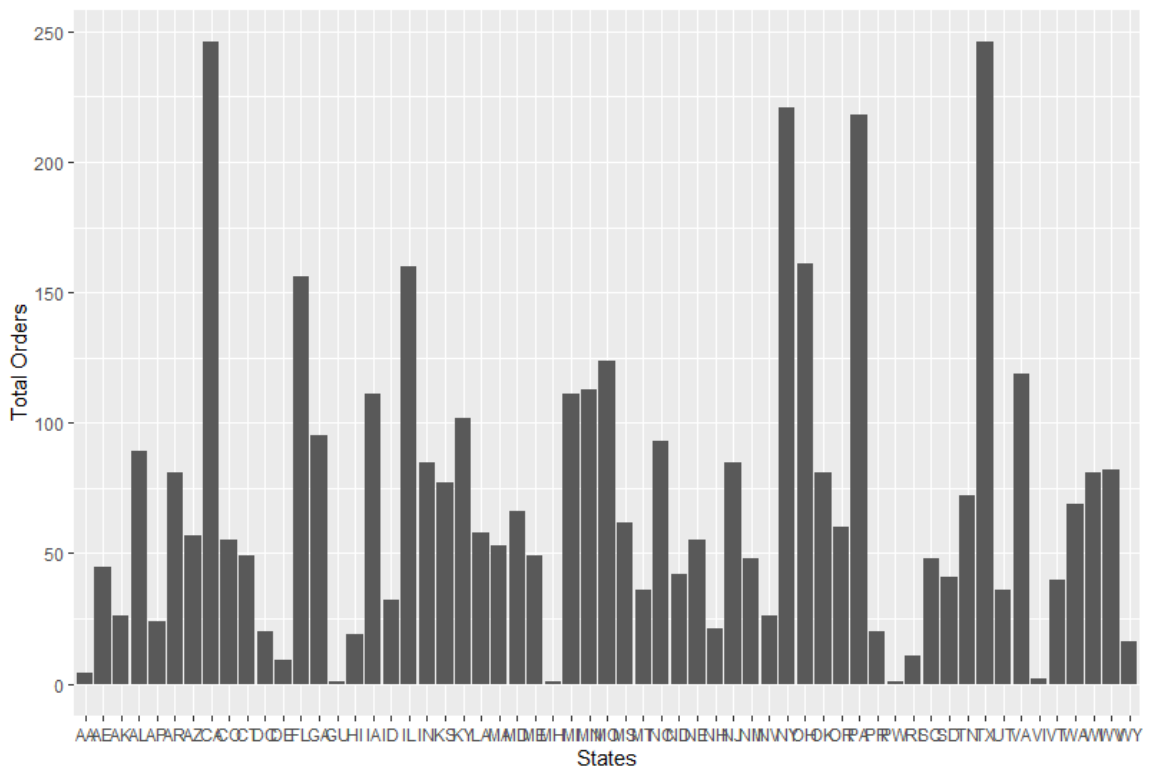


Figure 4.7: Total Order for each State

To have more representing graphs I will add the colors by adding the property “fill=State” to the geom col function. The graph shows that California and Texas are the top 2 states that order the products.

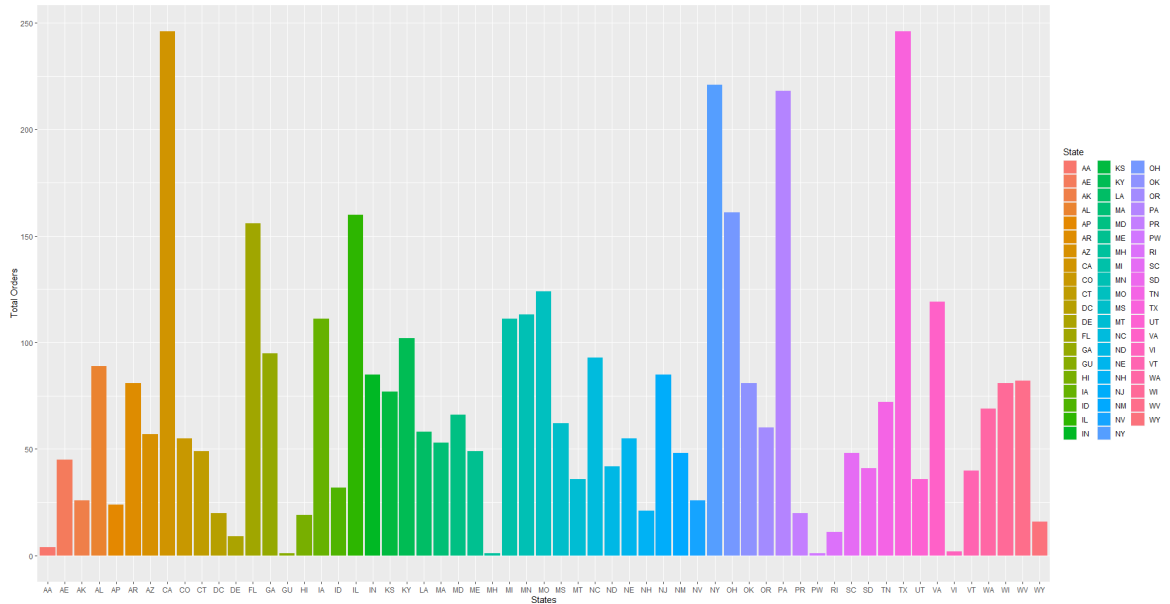


Figure 4.8: Low 100 Customer locations from one DC

Another way to show the Bar graph with more clear data is to create the graph as a circle. The script B.9 shows the script that was used to create the graph.

By run the script B.9 we get the graph 4.9



Figure 4.9: Bar graph with more clear data

How about the top 20/low 20 states? To get the top 20/low 20 states we will use the script in figure B.10 the script will sort the data frame to get the top 20 and low 20, then we will use that new dataframe with the top and low 20 to create the bars graphs.

Figure 4.10 and figure 4.11 show the top 20 states are California, Texas, New York, Pennsylvania are the top and most of the States are located on the Northeast (East Coast), Southwest and Southeast of the country.

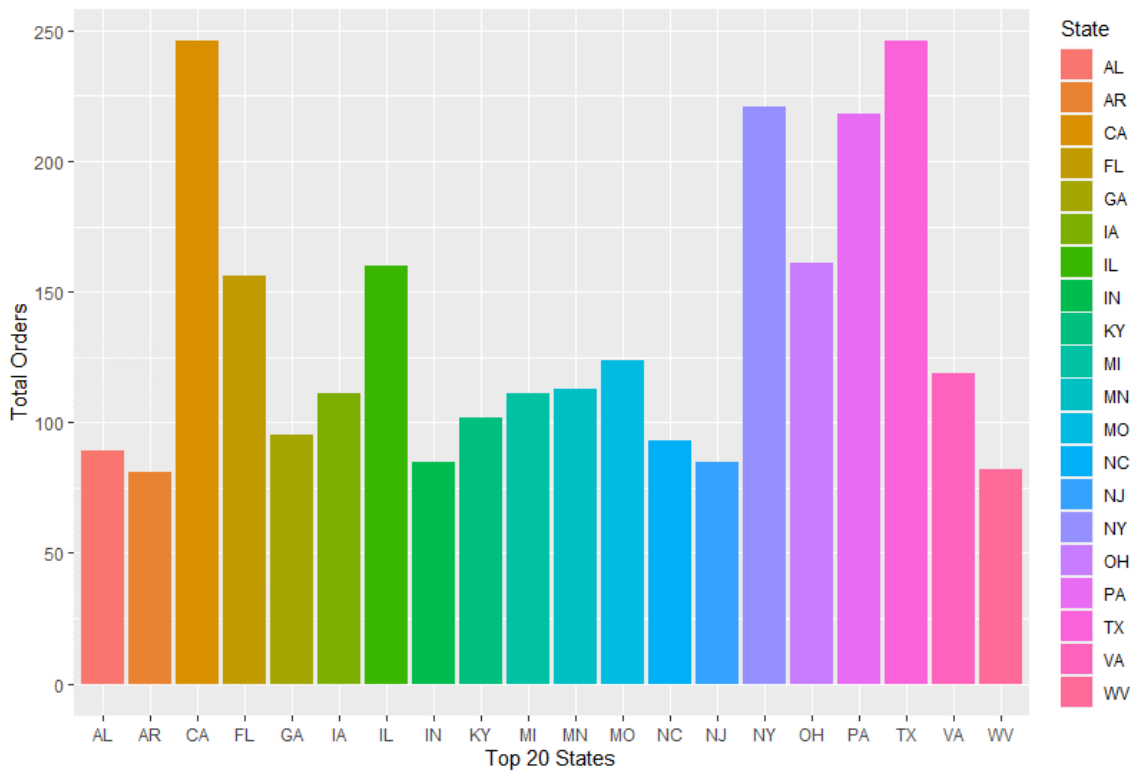


Figure 4.10: Total Orders for Top 20 States

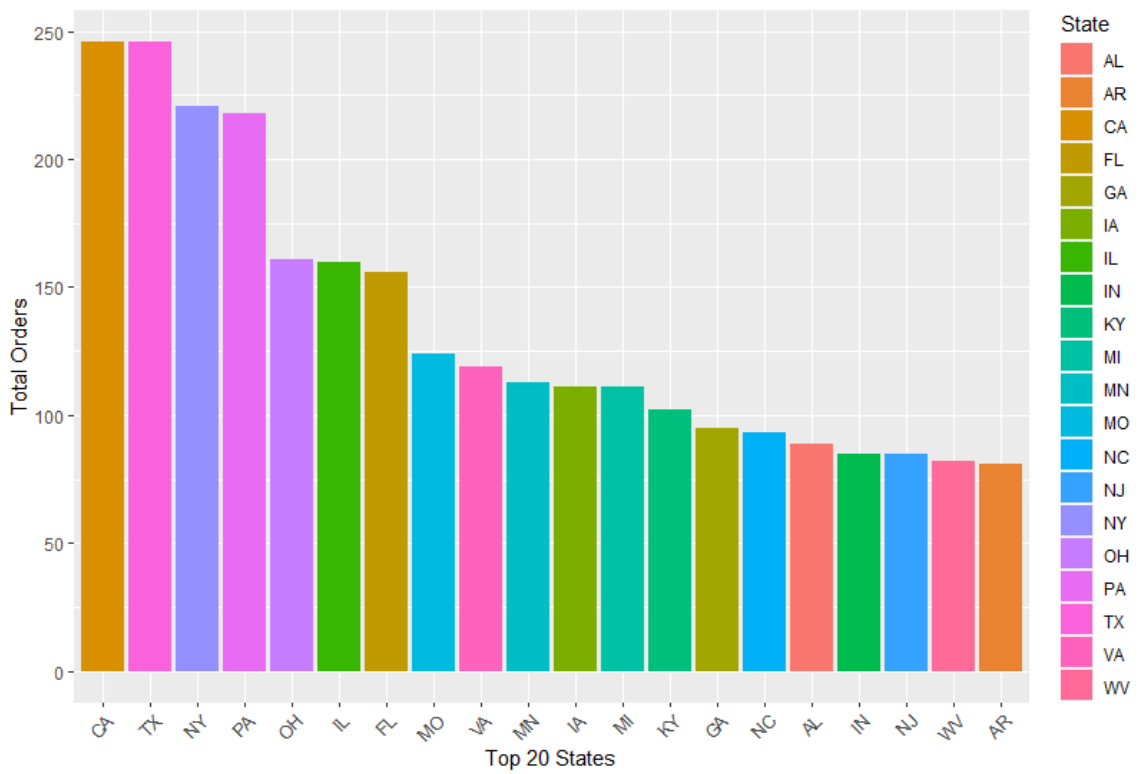


Figure 4.11: Total Orders for Top States Sorted

On the other hand, the low 20 states as shown in figure 4.12 and 4.13 show are South Dakota, Vermont, Utah, Idaho, Alaska, Nevada . . . Most of these states are located on the Northwest, Central and West coast of the country.

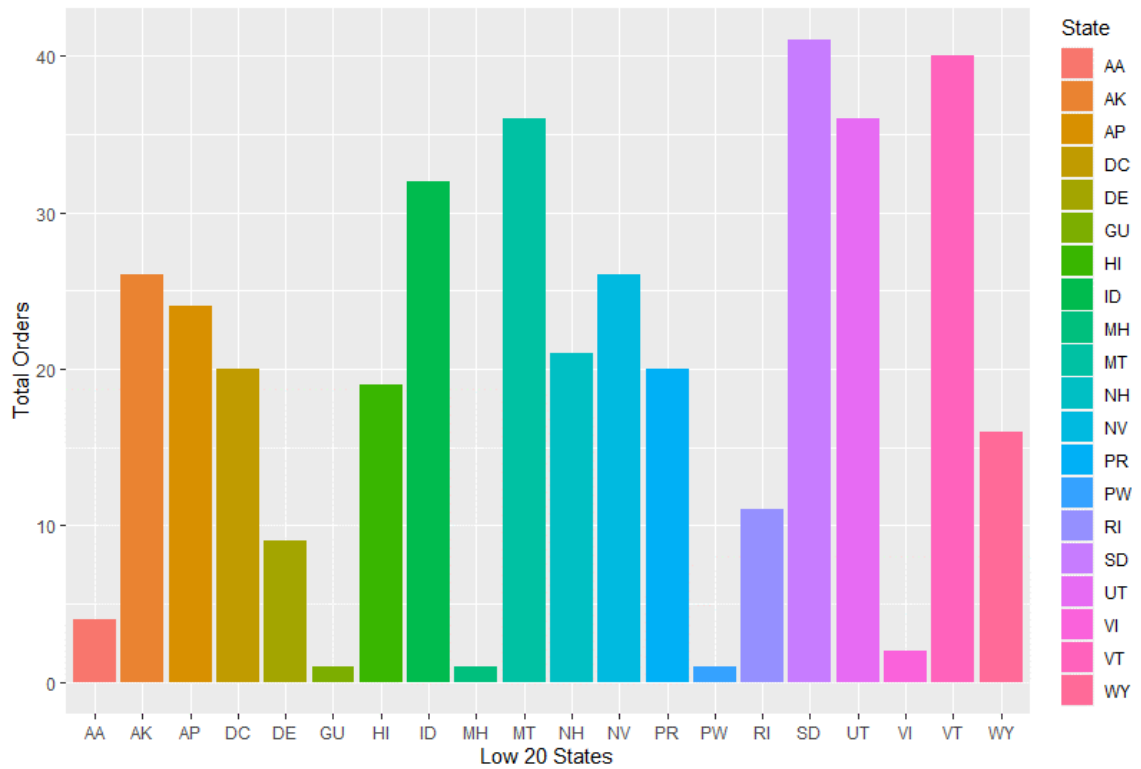


Figure 4.12: Total Orders for Low 20 States

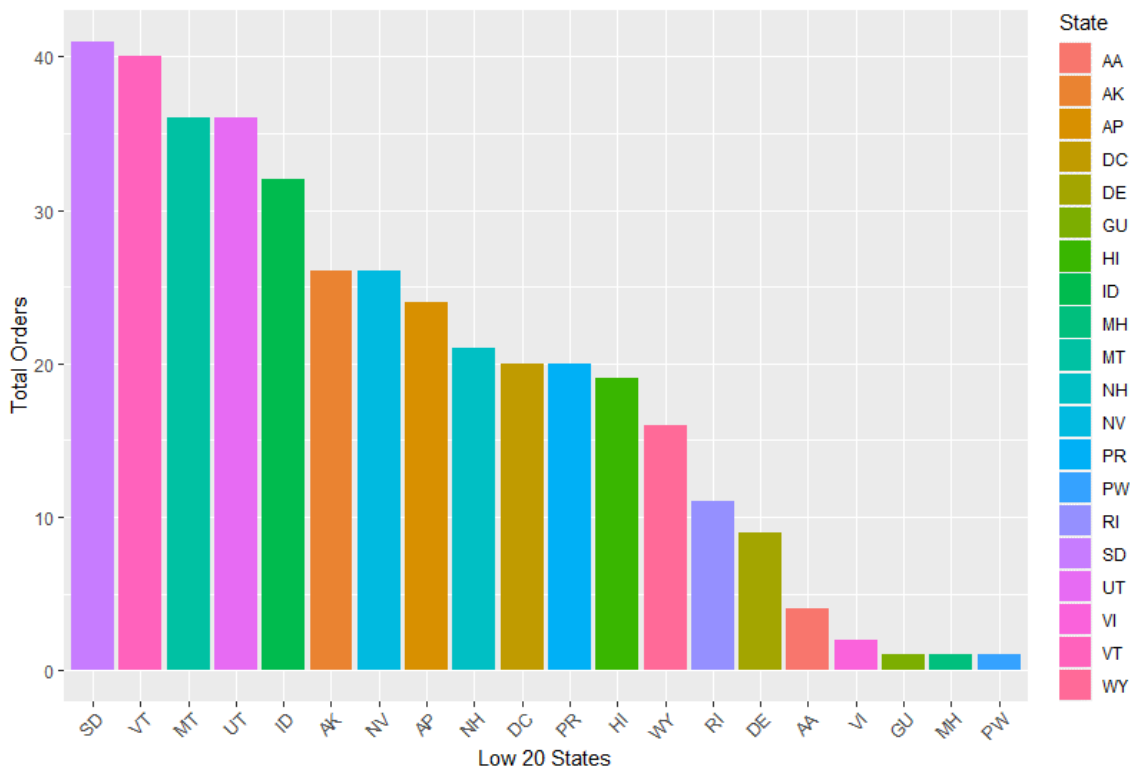


Figure 4.13: Total Orders for Low 20 States Sorted

After visualizing the total orders for each States with the top and low 20 states let do the same but this time with total quantities. The script B.11 group by the state and sum the quantity number. Using the bar graph, we get the graph in figure 4.14 and 4.15

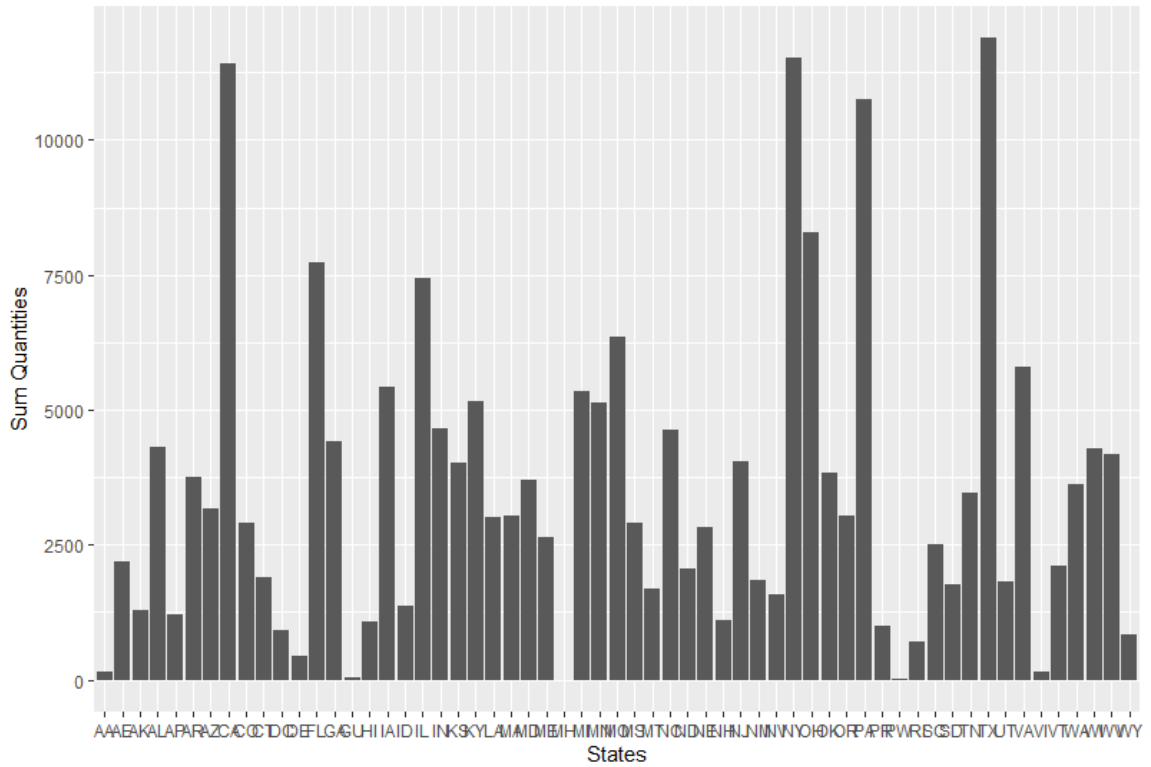


Figure 4.14: Total Quantity for each State

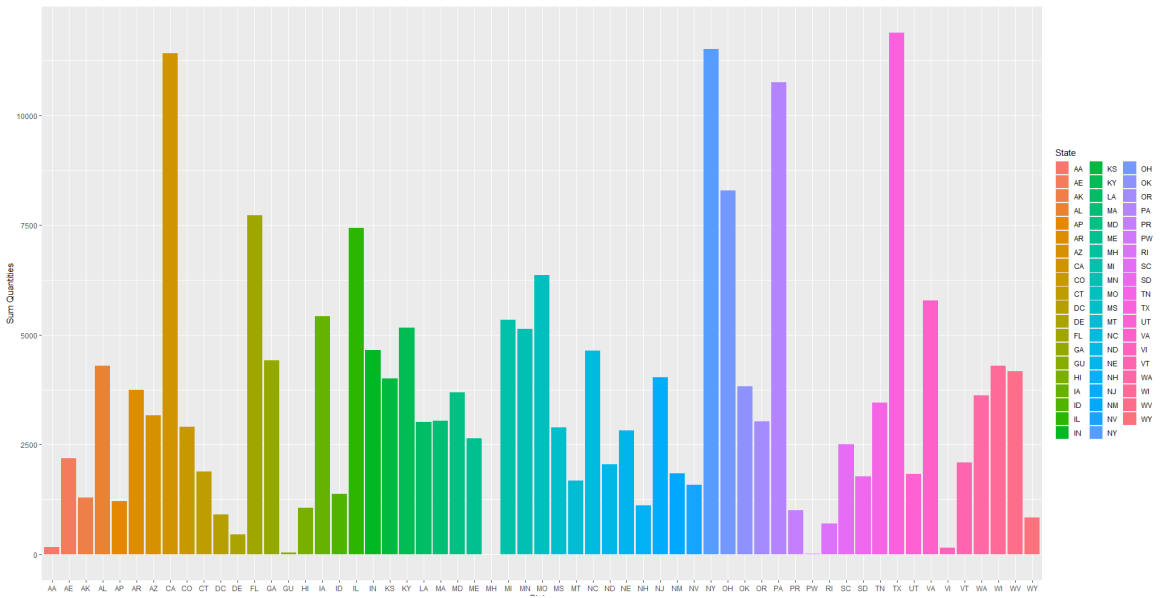


Figure 4.15: Total Quantity for each State

How about the top 20/low 20 states with quantity? To get the top 20/low 20 states we will use the script in figure B.12 The script will sort the data frame to get the top 20 and low 20, then we will use that new data frame with the top and low 20 to create the bars graphs.

Figure 4.16 and figure 4.17 show the top 20 states are Texas, New York, Pennsylvania, California, Ohio, Florida ... are the top and most of the States are located on the Northeast (East Coast), Southwest and Southeast of the country.

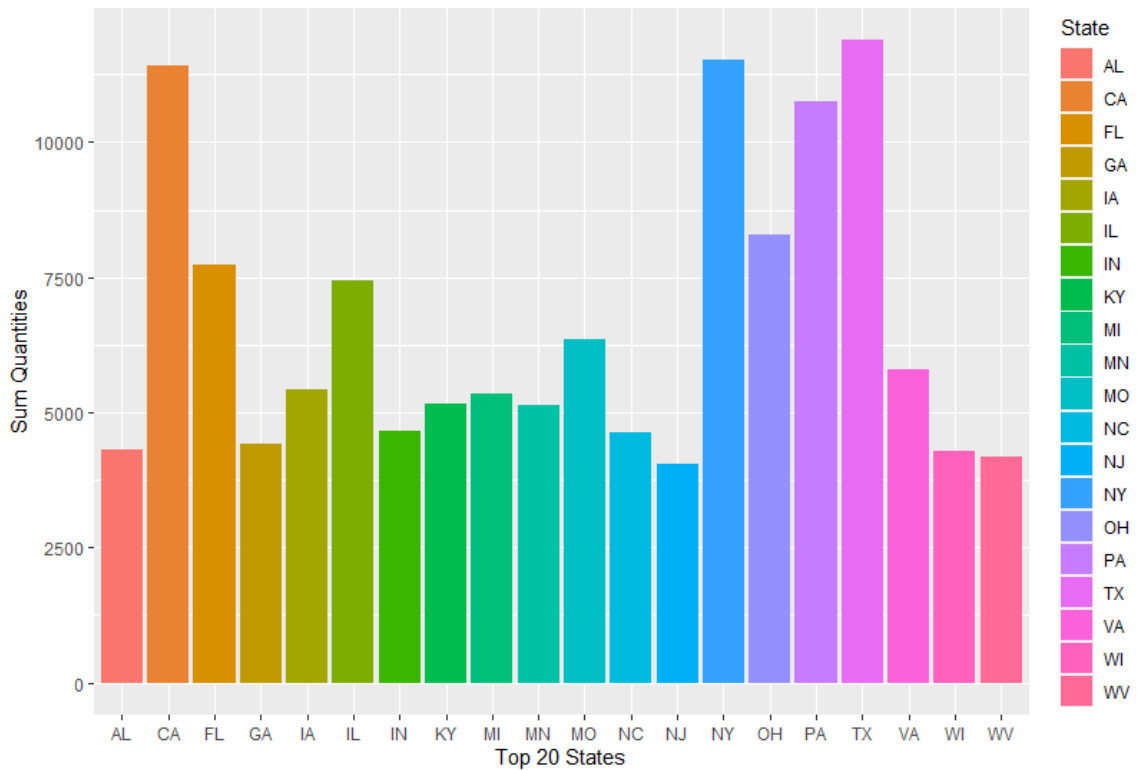


Figure 4.16: Total Quantity for Top 20 States

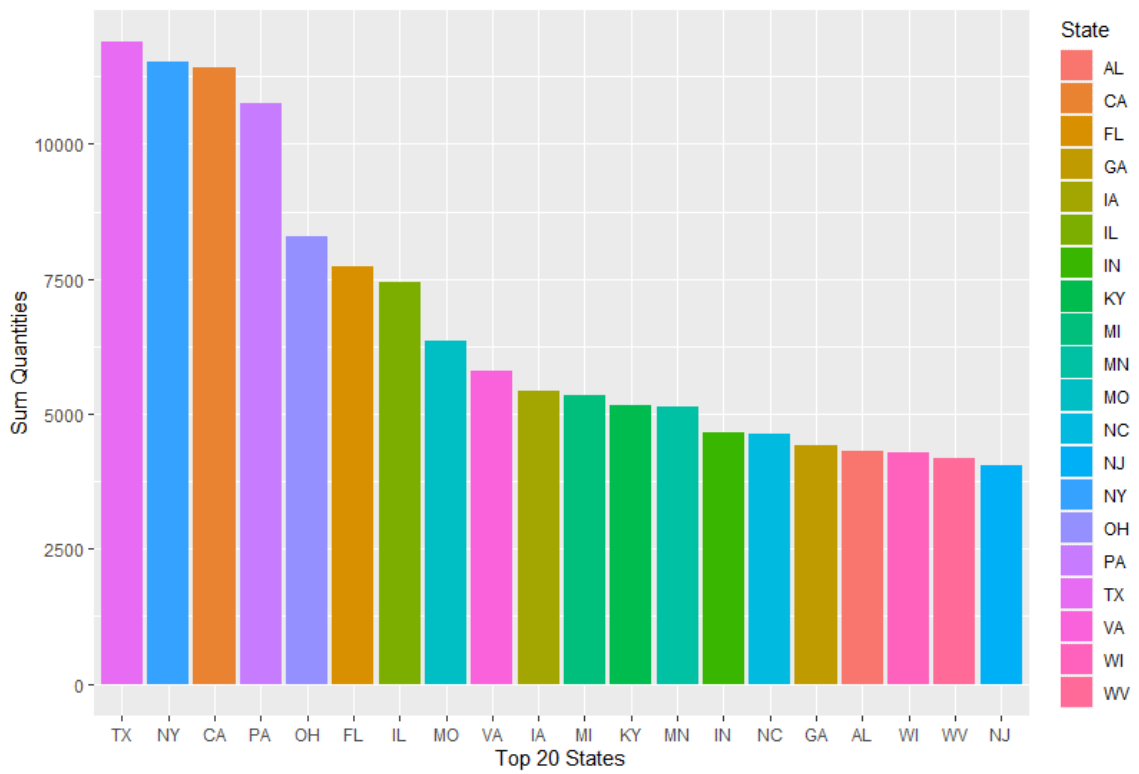


Figure 4.17: Total Quantity for Top 20 States Sorted

On the other hand, the low 20 states as shown in figure 4.18 and 4.19 show are Utah, South Dakota, New Mexico, Montana, Vermont, Utah, Idaho, Alaska Most of these states are located on the Northwest, Central and West coast of the country.

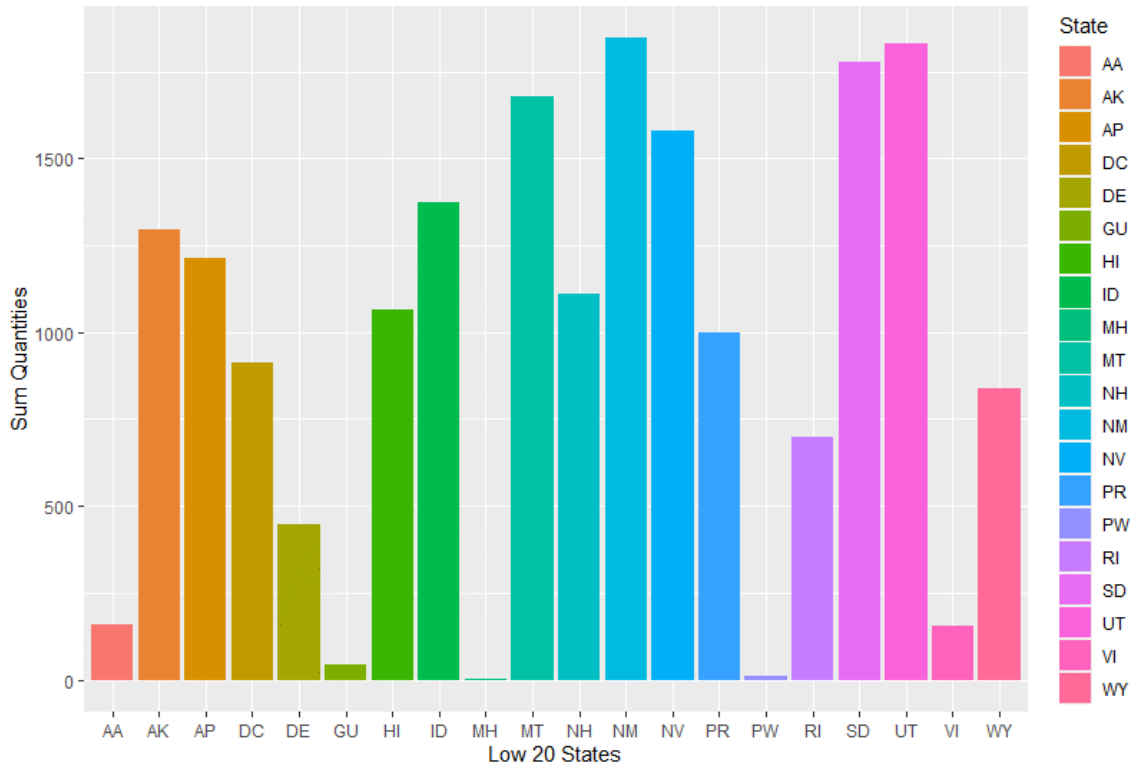


Figure 4.18: Total Quantity for Low 20 States

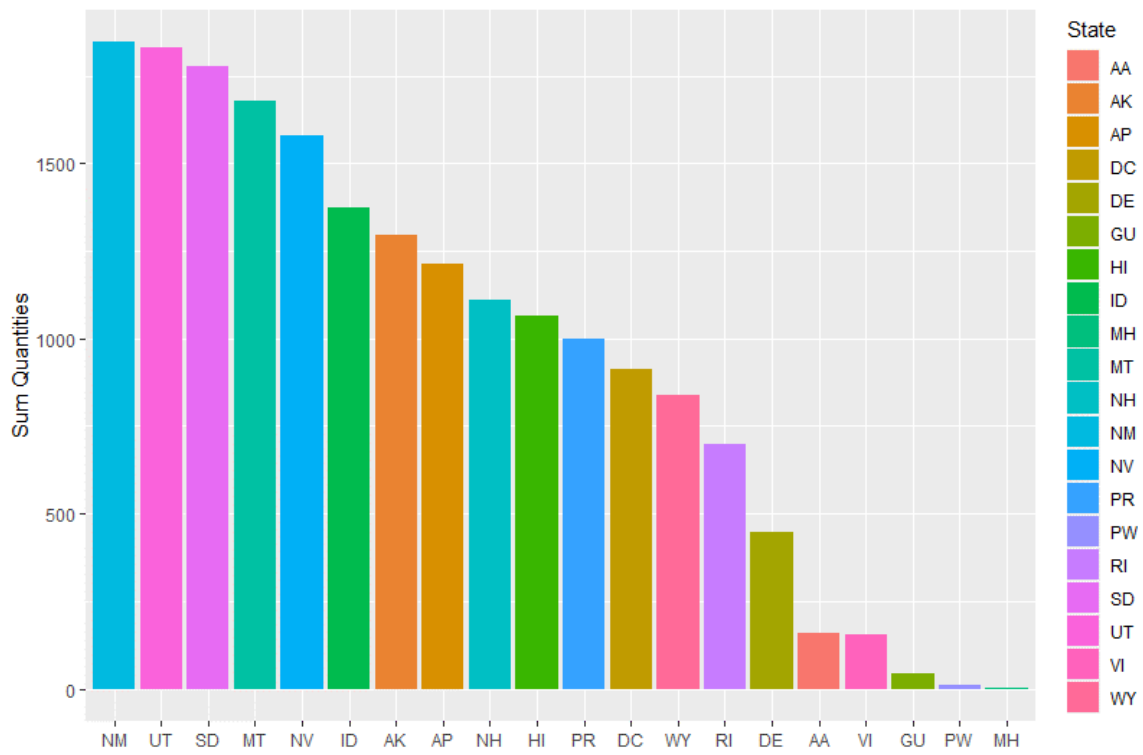


Figure 4.19: Total Quantity for Low 20 States Sorted

4.2.2 Pie graphs

Pie charts are typically used to tell a story about the parts-to-whole aspect of a set of data. That is, how big part A is in relation to part B, C, and so on. To create a pie we will use the Library "plotrix" that will help us with labelling and color. In the script B.13 we will create a simple pie and a 3D pie. The data we will need for the pie is the same we used for the example earlier, we will use the total number of quantities for the top 10 states and how the total quantities percentages for each state.

Figure 4.20 shows a simple pie graph listing the top 10 states. Texas is the one dominating the pie. After adding the percentage to get a better visualization with percentage we get the figure 4.21

Pie Chart of States

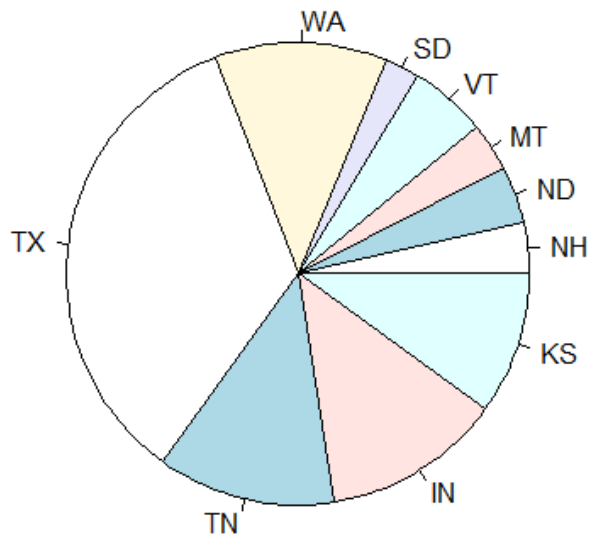


Figure 4.20: Pie Graph of the Top 10 states

Figures 4.21 and 4.22 shows that Texas is ordering 34 percent of the products that the company ships that year, followed by Tennessee and Indiana by 13 Percent

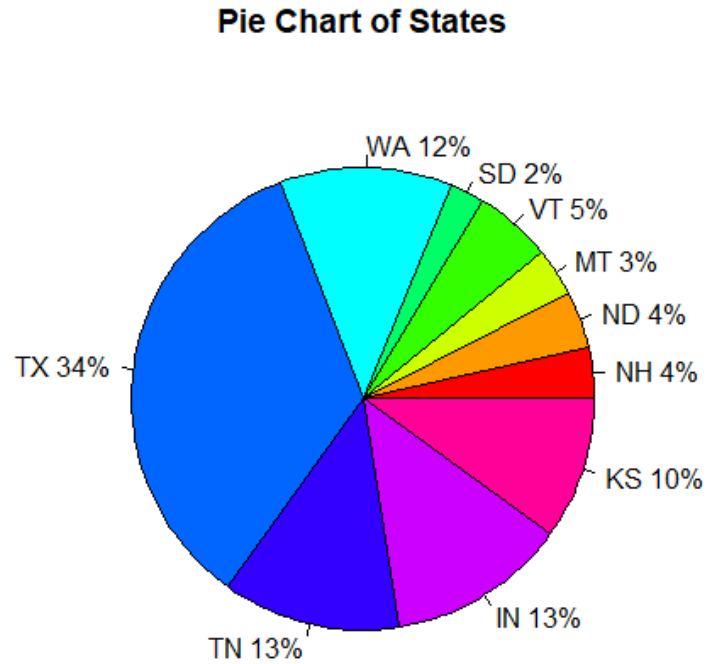


Figure 4.21: Pie Graph of the Top 10 states (Percentages)

Pie Chart of States

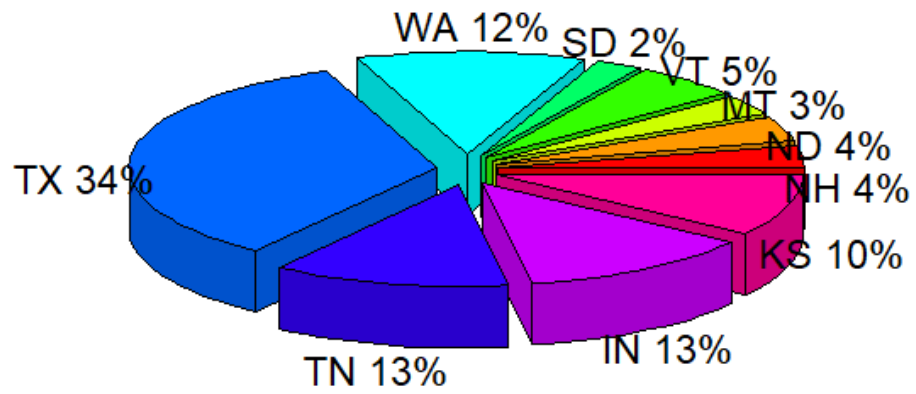


Figure 4.22: 3D Pie Graph of the Top 10 states

4.2.3 Dots graphs

To calculate the distance between two points on our data set we will be using the package "geosphere" [package, 2019] is spherical trigonometry for geographic applications.

That is a compute distance and related measures for angular (longitude/latitude) locations. Also, this package implements functions that compute various aspects of distance, direction, area, etc. for geographic (geodetic) coordinates. Some of the functions are based on an ellipsoid (spheroid) model of the world, other functions use a (simpler, but less accurate) spherical model.

Figure B.14 calculates the distance between one DC and each location in our dataset. The distance will be generated on Meters and we have to convert it to Miles. Figure 4.23 shows the distance between the one DCs and each location in the data set.

From the data we gather before about the top/low 20 total orders and quantity, it looks like distance is playing a big role in that, as figure 4.23 shows that most of the top 20 states are the closest to the DC and the low 20 states are the farthest from the DC.

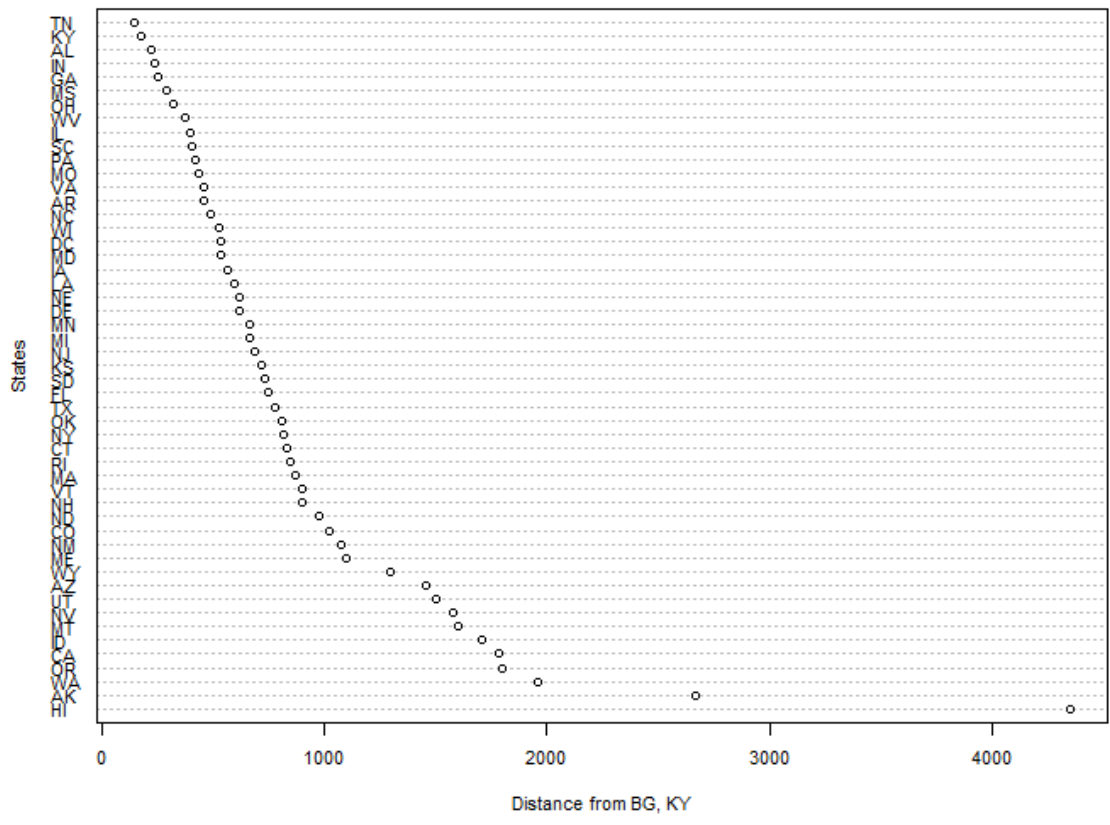


Figure 4.23: Distances between one DC and each State

Chapter 5

NETWORK

5.1 Network Modeling

5.1.1 What is Network?

Network is a collection of entities (nodes and edges) together with a set of relations on these entities.

Nodes are vertices that correspond to objects.

Edges are the connections between objects.

What is a relationship? Relationship is the property of two or more entities compose to properties of the entities alone(attributes).

So in the network analysis we study the relational data.

what is Density: the density of a network is defined as a ratio of the number of edges to the number of possible edges.

$$D = E / \text{Possible } E$$

Possible E depends on if we have an undirected or directed network.

[graph, 2019] An Undirected graph is graph, i.e., a set of objects (called vertices or nodes) that are connected together, where all the edges are bidirectional. An undirected graph is sometimes called an undirected network.

One can formally define an undirected graph as $G=(N,E)$, consisting of the set N of nodes and the set E of edges, which are unordered pairs of elements of N.

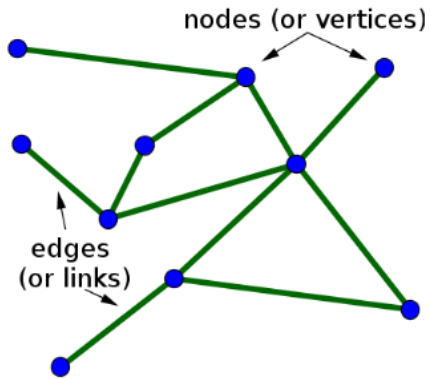


Figure 5.1: An undirected graph with 10 and 11 edges. [graph, 2019]

[graph, 2019] A directed graph is graph, i.e., a set of objects (called vertices or nodes) that are connected together, where all the edges are directed from one vertex to another. A directed graph is sometimes called a digraph or a directed network.

One can formally define a directed graph as $G=(N,E)$, consisting of the set N of nodes and the set E of edges, which are ordered pairs of elements of N .

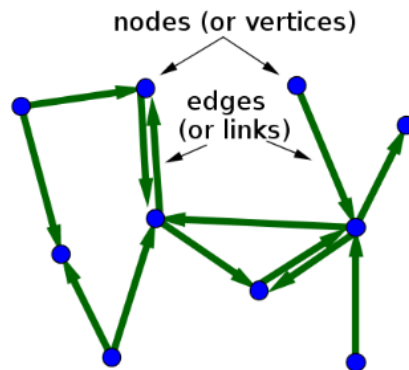


Figure 5.2: A directed graph with 10 vertices (or nodes) and 13 edges. [graph, 2019]

5.1.2 Represent a network in R

[graph and its representations, 2019] We can represent a graph using :

1. Adjacency Matrix: Adjacency Matrix is a 2D array of size $V \times V$ where V is the number of vertices in a graph.

2. Adjacency List: An array of lists is used. The size of the array is equal to the number of vertices.

To represent a network in R we will use the package "igraph", "igraph" is a library and R package for network analysis. Figure B.15 shows the script to create the network between the current DCs and the cities that shipped to, since this is a sensitive data that the company did not want me to show, I will be generating a random DCs list and assign it to cities and states, the goal of this is to show how network graph can give us an idea where the DCs ship the most of their products.

Figure 5.3 shows the result of the script B.15 and create a network between each DC and the states.

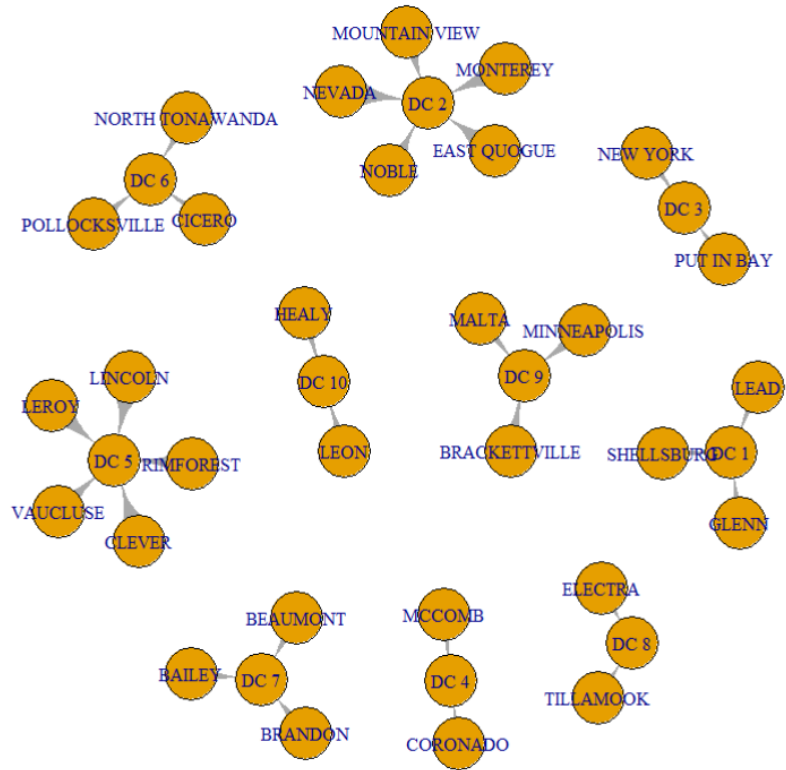


Figure 5.3: Network Between DCs and Cities

It is hard to tell which state the city is located in the graph, so let's add the state's name to the city column.

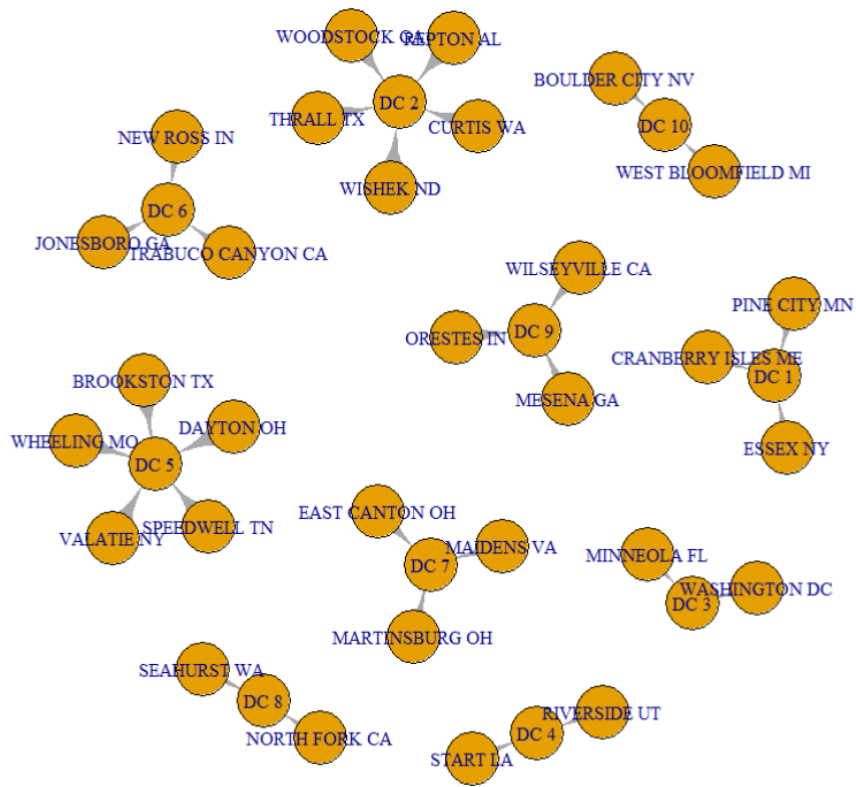


Figure 5.4: Network Between DCs and Cities with States

What Does the network of the states in the U.S look like? Figure 5.5 shows the network between the current DCs and the states.

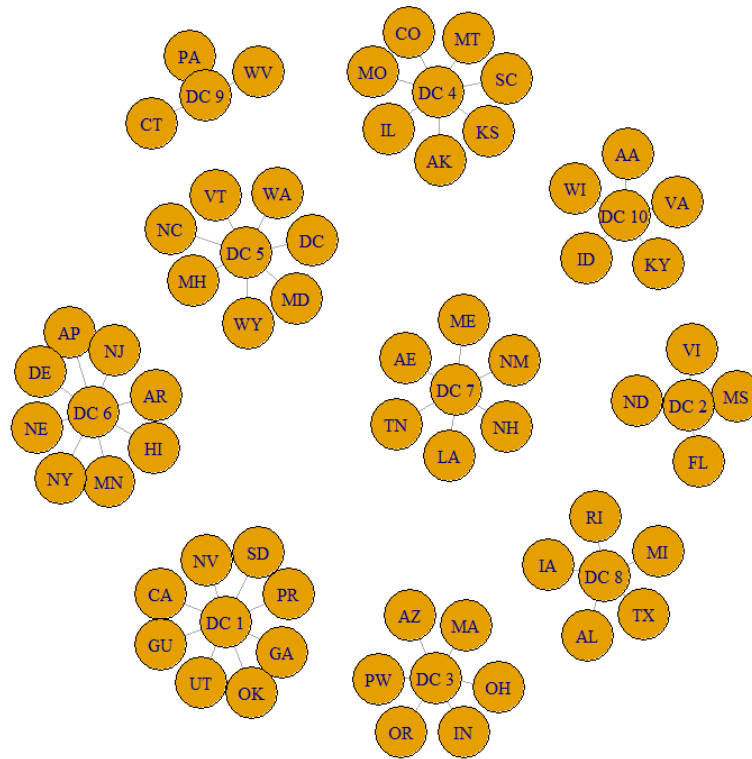


Figure 5.5: Network Between DCs and States

The network data will make more sense when the company will apply the real data.

There are many ways to show this network graph by changing the layout of the network from sphere, circle, random and fruchterman.reingold

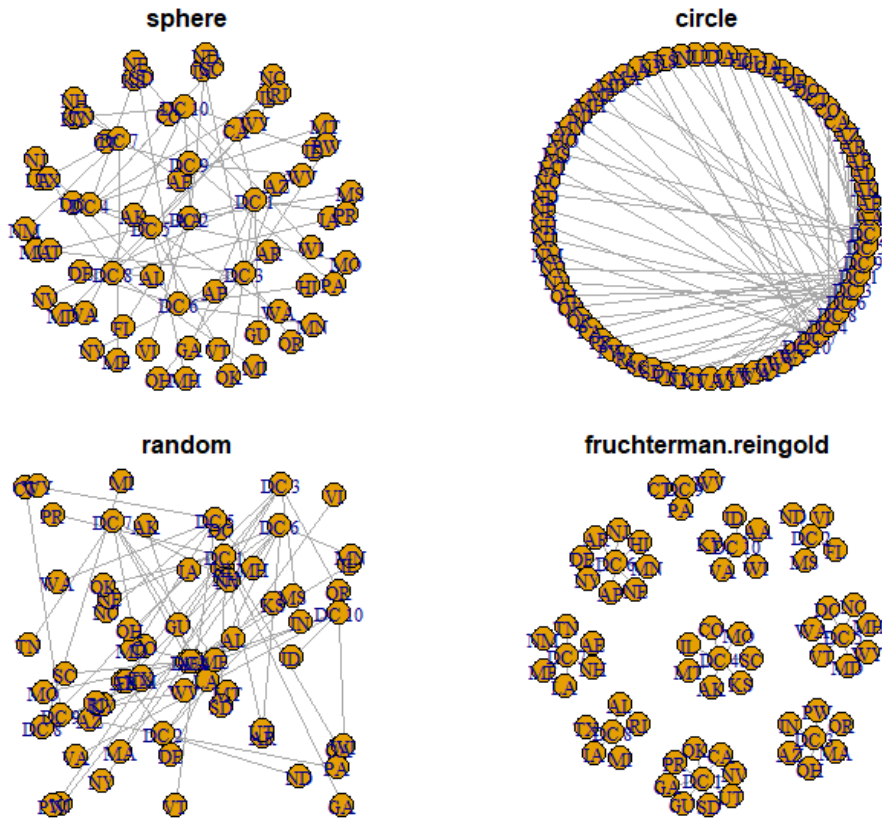


Figure 5.6: Network Between DCs and States

Not All the DCs ship the same amount of products to states and to differentiate between the DCs that ship more items we need to change the node size depending on the flow from that DC.

Figure 5.7 shows that DC 6, DC1, DC4 and DC5 shipped more items than other DCs.

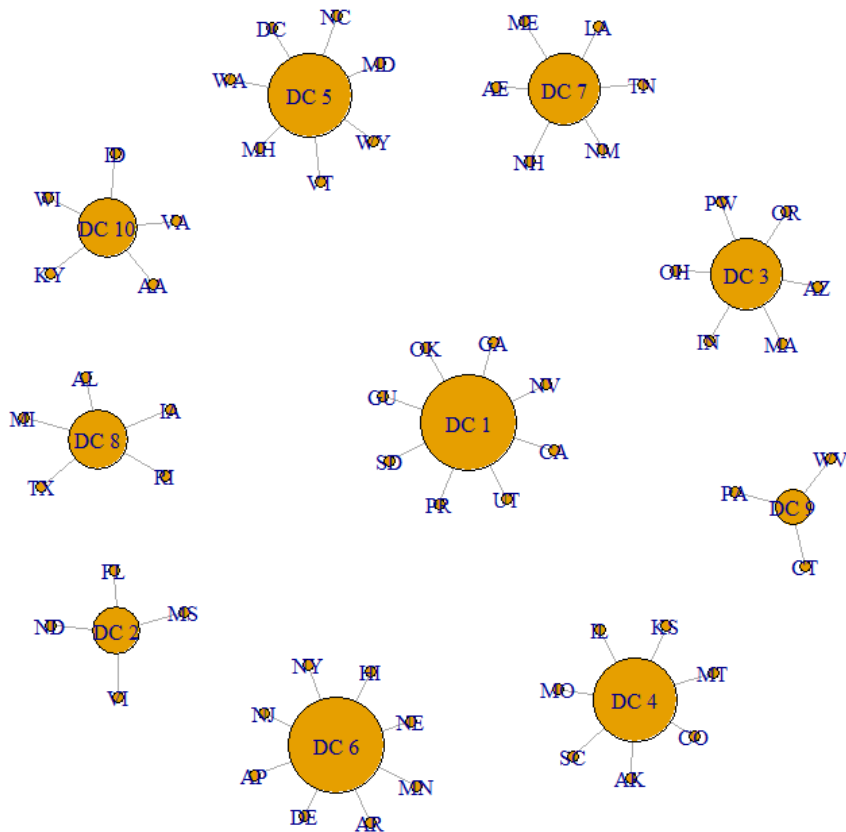


Figure 5.7: Network Between DCs and States

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In total word managing supply chains become complicated due to the delivery system becoming more global. However, Visualization can help us manage the supply chain for Ecommerce shipping data especially if it is within one country like the USA. Graphs like Maps, Bars, Networks ... can clearly give us some advantage to visualize and track the data, to have a better idea where are the products delivered now and how is that compared to our current DCs location. Also, how to make that more efficient and is there is any changes to the current DCs location that will improve our supply chain and make the company more profit and make the customer happy. From the Visualization we did in this project we conclude the below:

- Most of the customers are in the Northeast (East Coast and Great Lakes) of the country, fellow by Southeast and Southwest of the country. So, most of the users are located in the haft east of the country.
- Few Customers are located on the Northwest, Central and West coast of the country.
- Distance is playing a big role in the total quantity ordered from each state. The top 20 states are the closest to the DC and the low 20 states are the farthest from the DC.

As a result, the current DCs are great for the current demand but we need a new

DCs in Northeast (East Coast) of the country, where we have a lot of customers with the biggest total quantity shipped.

6.2 FUTURE WORK

Incorporate how to combine existing DC's and new DC's and create network optimizations. Also, implement the project, to do that, I will create an application that will have an interface, which will make it easy for the users to use. They will have to load a new data set of shipment records, also some basic information such as, entering the current location of the distribution center, it can be one or multiple distribution centers across the United States. The application will take the information the user enters, and the data set file, then it will show a map with the current DC's location, and will predict distribution center(s) simultaneously, based on the shipment data.

REFERENCES

- ArcGIS (1999). Arcgis. <https://www.arcgis.com/index.html>. [Online].
- Baback Bashari Rad, Harrison John Bhatti, M. A. (2017). An introduction to docker and analysis of its performance. *International Journal of Computer Science and Network Security*.
- Bijesh Dhyani, A. B. (2014). Big data analytics using hadoop. *International Journal of Computer Applications*.
- Container (2013). What is a container? <https://www.docker.com/resources/what-container>. [Online].
- CRANRProject (2020). Cran r project. <https://cran.r-project.org>. [Online].
- documentation, R. (2019). ndtv package. <https://cran.r-project.org/web/packages/ndtv/vignettes/ndtv.pdf>. [Online].
- Dongmin Kim, Hanif Muhammad, E. K. S. H. C. L. (2019). Tosca-based and federation-aware cloud orchestration for kubernetes container platform. *Applied Sciences*.
- Funaki, K. (2009, August). State of the art survey of commercial software for supply chain design.
- graph and its representations (2019). graph and its representations. <https://www.geeksforgeeks.org/graph-and-its-representations/>. [Online].
- graph, U. (2019). Undirected graph definition. https://mathinsight.org/definition/undirected_graph. [Online].
- llamasoft (1998). llamasoft. <https://www.llamasoft.com/>. [Online].
- M. Dhavapriya, N. Y. (2016). Big data analytics: Challenges and solutions using hadoop, map reduce and big table. *International Journal of Computer Science Trends and Technology*.
- Nag, B., C. Han, and D. qing Yao (2014). Mapping supply chain strategy: an industry analysis. *Journal of Manufacturing Technology Management* 25(3), 351–370.

- Nelson Dzipire, Y. N.-G. (2014). A multi-stage supply chain network optimization using genetic algorithms. *Mathematical Theory and Modeling*.
- Niki Matinrad, Emad Roghanian, Z. R. (2013). Supply chain network optimization: A review of classification, models, solution techniques and future research. *Growing Science*.
- package, G. (2019). geosphere. <https://cran.r-project.org/web/packages/geosphere/vignettes/geosphere.pdf>. [Online].
- Patil, S., A. Navada, A. Peshave, and V. Borole (2011, July). Online c/c++ compiler using cloud computing. In *2011 International Conference on Multimedia Technology*, pp. 3591–3594.
- Sayfan, G. (2017). *Mastering Kubernetes: Automating container deployment and management*. Packet.
- Schenker, G. (2018). *Containerize your Apps with Docker and Kubernetes*. Packt Publishing.
- Shinyapp (2017). Shiny apps. <https://www.shinyapps.io/>. [Online].
- Shvachko, K., H. Kuang, S. Radia, and R. Chansler (2010, May). The hadoop distributed file system. In *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, pp. 1–10.
- Zhang, Q., L. Liu, C. Pu, Q. Dou, L. Wu, and W. Zhou (2018). A comparative study of containers and virtual machines in big data environment. *CoRR abs/1807.01842*.
- ZipCode (2012). Zip code. <http://federalgovernmentzipcodes.us/>. [Online].

Appendices

Appendix A APPENDIX A

A.1 R and RStudio

A.1.1 Install and Download R

First open the URL : <https://www.r-project.org/>

Click on the link Download R

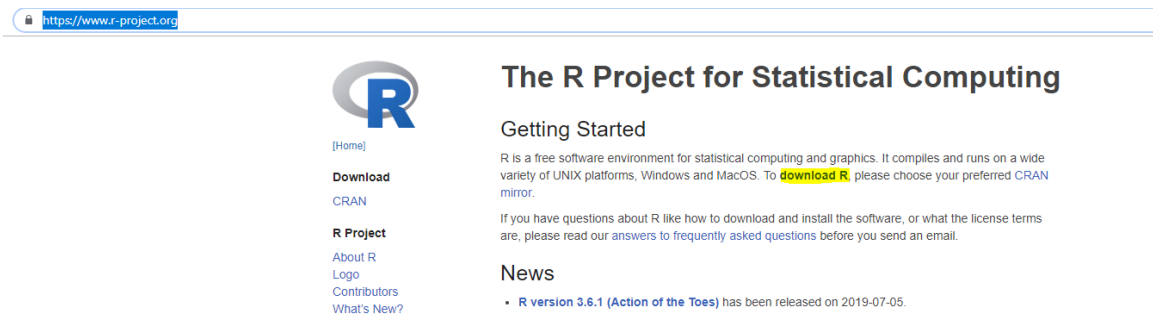


Figure A.1: Download R

Select one of the closest universities to your location

UK	https://www.stats.bris.ac.uk/R/ http://www.stats.bris.ac.uk/R/ https://cran.ma.imperial.ac.uk/ http://cran.ma.imperial.ac.uk/	University of Bristol University of Bristol Imperial College London Imperial College London
USA	https://cran.cnr.berkeley.edu/ http://cran.cnr.berkeley.edu/ https://mirror.las.iastate.edu/CRAN/ http://mirror.las.iastate.edu/CRAN/ https://ftp.usg.iu.edu/CRAN/ http://ftp.usg.iu.edu/CRAN/ https://rweb.crmda.ku.edu/cran/ http://rweb.crmda.ku.edu/cran/ https://cran.mtu.edu/ http://cran.mtu.edu/ https://repo.miserver.it.umich.edu/cran/ http://cran.wustl.edu/ http://archive.linux.duke.edu/cran/ https://cran.case.edu/ http://cran.case.edu/ https://ftp.osuosl.org/pub/cran/ http://ftp.osuosl.org/pub/cran/ http://lib.stat.cmu.edu/R/CRAN/ http://cran.mirrors.hoobly.com/ https://mirrors.nics.utk.edu/cran/ http://mirrors.nics.utk.edu/cran/ https://cran.revolutionanalytics.com/ http://cran.revolutionanalytics.com/	University of California, Berkeley, CA University of California, Berkeley, CA Iowa State University, Ames, IA Iowa State University, Ames, IA Indiana University Indiana University University of Kansas, Lawrence, KS University of Kansas, Lawrence, KS Michigan Technological University, Houghton, MI Michigan Technological University, Houghton, MI MBNI, University of Michigan, Ann Arbor, MI Washington University, St. Louis, MO Duke University, Durham, NC Case Western Reserve University, Cleveland, OH Case Western Reserve University, Cleveland, OH Oregon State University Oregon State University Statlib, Carnegie Mellon University, Pittsburgh, PA Hoobly Classifieds, Pittsburgh, PA National Institute for Computational Sciences, Oak Ridge, TN National Institute for Computational Sciences, Oak Ridge, TN Revolution Analytics, Dallas, TX Revolution Analytics, Dallas, TX

Figure A.2: Download R

Select one of the operating systems where you want to install R

Choose "Base" if this is the first time you are installing R

Finally click Download link to download R

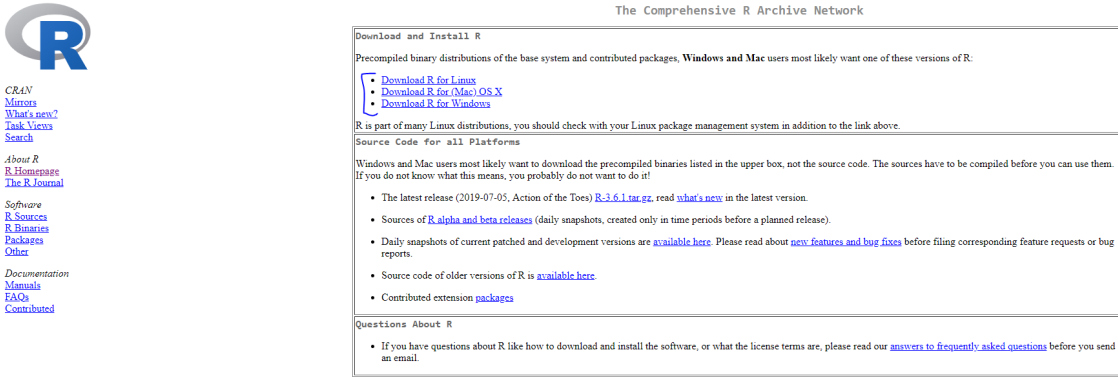


Figure A.3: Download R

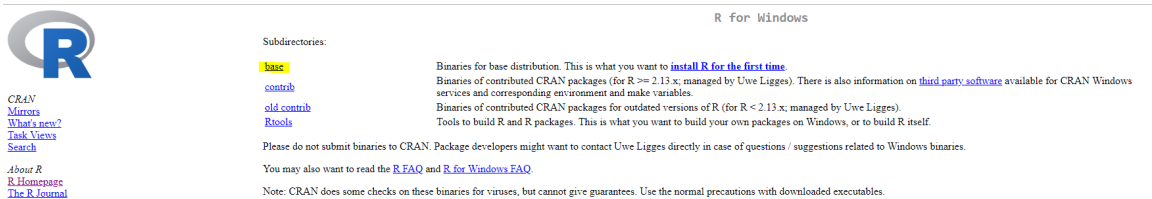


Figure A.4: Download R

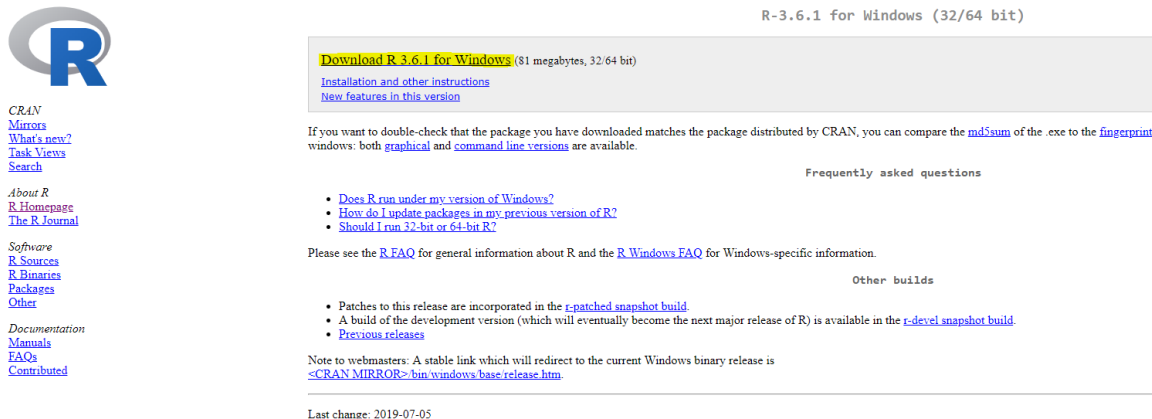


Figure A.5: Download R

Open the download application and start the install of R by choosing a language and Click next

Choose the default option "NO" and click next.

R is installed now!

A.1.2 Install and Download RStudio

First open the URL : <https://www.rstudio.com/products/rstudio/download/> click on the link Download

Choose one platform.

Open the downloaded application and start installing RStudio by clicking "Next" RStudio is installed now!

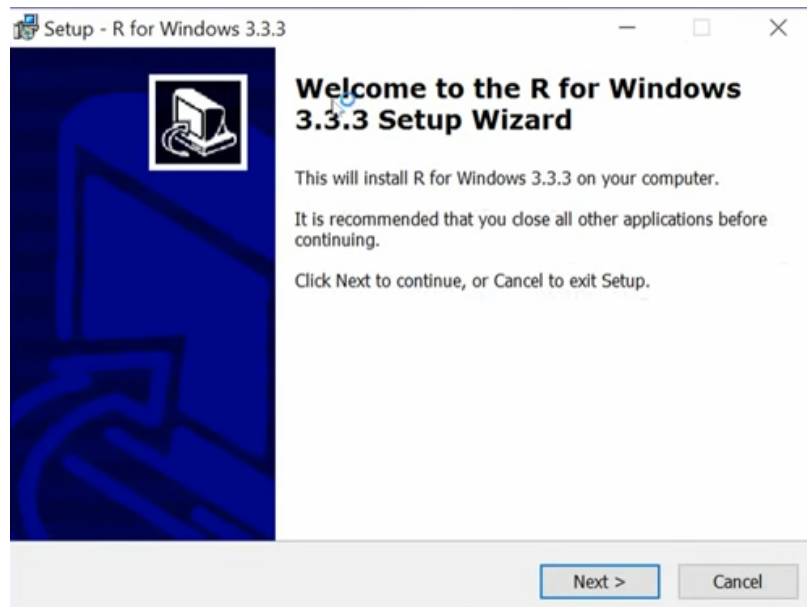


Figure A.6: Install R

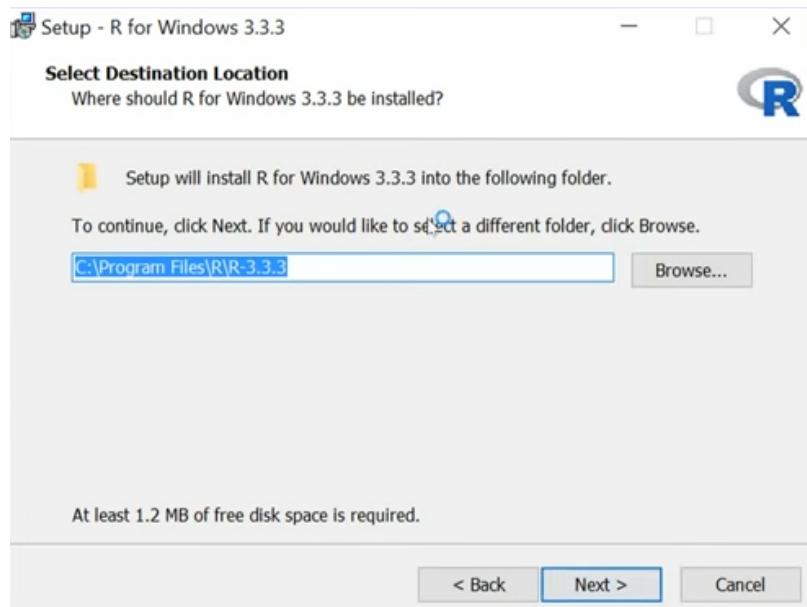


Figure A.7: Install R

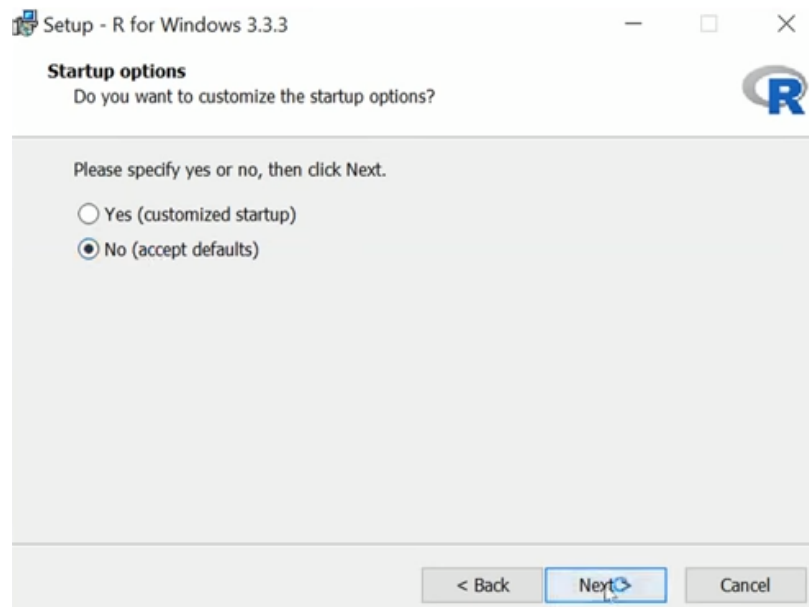


Figure A.8: Install R



Figure A.9: Install R

<https://www.rstudio.com/products/rstudio/download/>

R Studio

Products Resources Pricing About Us Blogs

Choose Your Version of RStudio

RStudio is a set of integrated tools designed to help you be more productive with R. It includes a console, syntax-highlighting editor that supports direct code execution, and a variety of robust tools for plotting, viewing history, debugging and managing your workspace. [Learn More](#) about RStudio features.

RStudio's new solution for every professional data science team. RStudio Team includes RStudio Server Pro, RStudio Connect and RStudio Package Manager. [LEARN MORE](#)

RStudio Desktop Open Source License	RStudio Desktop Commercial License	RStudio Server Open Source License	RStudio Server Pro Commercial License
FREE	\$995 per year	FREE	\$4,975 per year (5 Named Users)
DOWNLOAD	BUY	DOWNLOAD	BUY
Learn More	Learn More	Learn More	Evaluation Learn More
Integrated Tools for R	●	●	●
Priority Support	●		●
Access via Web Browser		●	●
Enterprise Security			●

Figure A.10: Download RStudio

R Studio

Products Resources Pricing About Us Blogs

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.2.1335 - Windows 7+ (64-bit)	126.9 MB	2019-04-08	d0e2470f1f8ef4cd35a669aa323a2136
RStudio 1.2.1335 - macOS 10.12+ (64-bit)	121.1 MB	2019-04-08	6c570b0e2144583f7c48c284ce299eeef
RStudio 1.2.1335 - Ubuntu 14/Debian 8 (64-bit)	92.2 MB	2019-04-08	c1b07d0511469abfe582919b183eee83
RStudio 1.2.1335 - Ubuntu 16 (64-bit)	99.3 MB	2019-04-08	c142d69c210257fb10d18c045fff13c7
RStudio 1.2.1335 - Ubuntu 18/Debian 10 (64-bit)	100.4 MB	2019-04-08	71a8d1990c0d97939804b46cfb0aea75
RStudio 1.2.1335 - Fedora 19/RedHat 7 (64-bit)	114.1 MB	2019-04-08	296b6ef88969a91297fab6545f256a7a
RStudio 1.2.1335 - Debian 9 (64-bit)	100.6 MB	2019-04-08	1e32d4d6f6e216f086a81ca82ef65a91
RStudio 1.2.1335 - OpenSUSE 15 (64-bit)	101.6 MB	2019-04-08	2795a63c7ef8e2aa2dae86ba09a81e5
RStudio 1.2.1335 - SLES/OpenSUSE 12 (64-bit)	94.4 MB	2019-04-08	c65424b06ef6737279d982db9eeFcae1

Zip/Tarballs

Zip/tar archives	Size	Date	MD5
RStudio 1.2.1335 - Windows 7+ (64-bit)	186.6 MB	2019-04-08	f1e013ade0c241969400507cf258e0ad
RStudio 1.2.1335 - Ubuntu 14/Debian 8 (64-bit)	137.6 MB	2019-04-08	e3e1ea2dd113fd9cf4d40bc5035ef6fde
RStudio 1.2.1335 - Ubuntu 18/Debian 10 (64-bit)	147.8 MB	2019-04-08	5ee7dd7b501675f0a631c62d403ea1b6
RStudio 1.2.1335 - Debian 9 (64-bit)	148.1 MB	2019-04-08	8090451cb7d520633eba80fd355ad4c1
RStudio 1.2.1335 - Fedora 19/RedHat 7 (64-bit)	147.2 MB	2019-04-08	34630cd7c66c3429879bd79982349380

Source Code

A tarball containing source code for RStudio v1.2.1335 can be downloaded from [here](#)

Figure A.11: Download RStudio

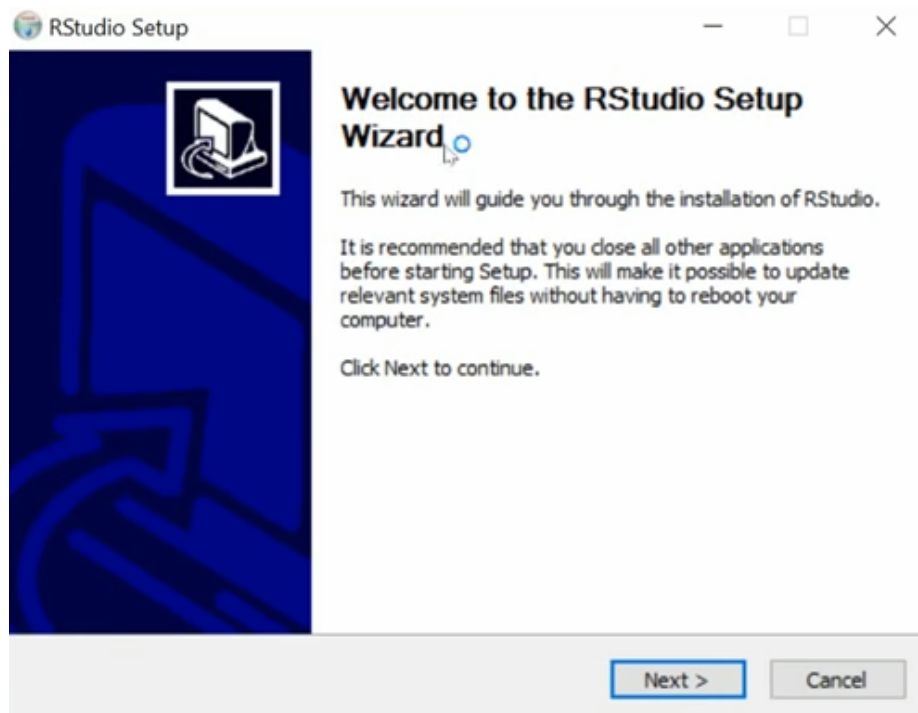


Figure A.12: Download RStudio

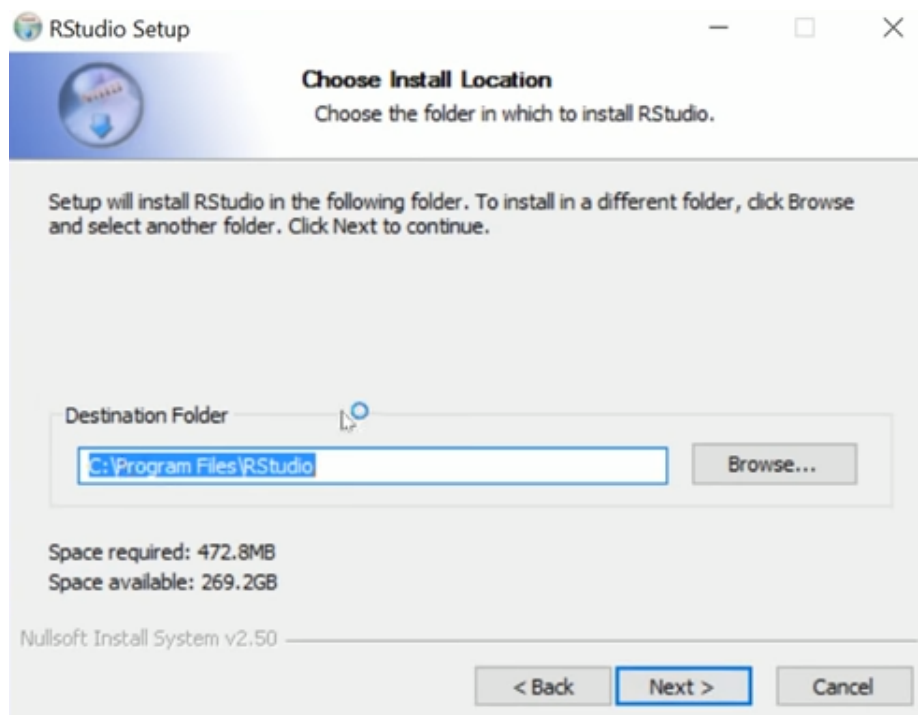


Figure A.13: Download RStudio

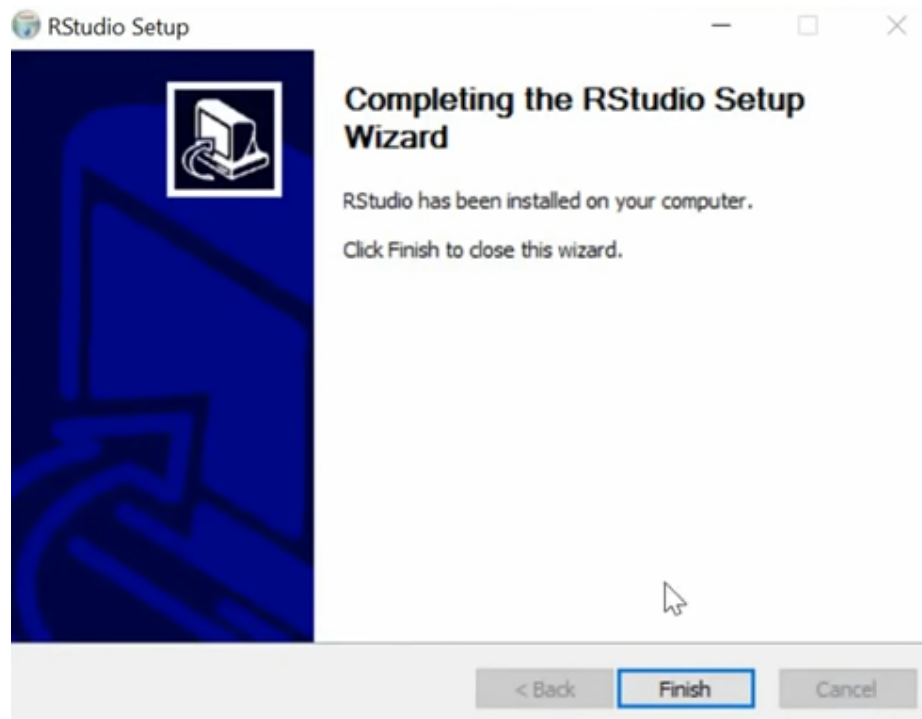


Figure A.14: Download RStudio

Appendix B APPENDIX B

B.1 Datasets

B.1.1 First Dataset

The Dataset contain the following:

- **Origin DC:** The Distribution Center the item was originally send from.
- **Customer Ship to:** The name of the customer that the item was ship to.
- **Customer Ship to City:** Customer city.
- **Customer Ship to Country:** Customer Country.
- **Customer Ship to State:** Customer State if the country is the USA.
- **Customer Ship to Province:** Customer Province (Canada ...)
- **Customer Ship to Zip:** Customer ship to zip code.
- **Customer Ship to Type:** Whether it's an organization or not.
- **Shipment date:** The date the DC ship the item.
- **Style:** The style of the item.
- **Color:** The color of the item.
- **Size:** The size of the item (S, M, L, XL...).
- **Shipped Unit of Measure:** The unit is either "Each" unit or a "Pack" unit. The pack unit is from Two to Twenty-Two items in a Pack.
- **Ship Method Code:** Not sure what it is used for and if I am going to use it in my analysis.
- **Carrier Name:** The Name of the Carriers (or FEDEX or UPS or ...) plus if it is an LTL (Less than truckload shipping or less than load is the transportation of relatively small freight) or it is a full truckload or PICK UP or PARCEL or ...
- **Freight Terms:** Collect, Paid, Due, THIRD PARTY
- **Customer Sold to:** ID, Name, Type
- **Customer Ship to:** ID, Name, Type, Zip
- **Customer Deliver to:** ID, Type, Zip (the most important data in the Dataset).
- **Delivery shipped quantity:** The quantity that was shipped. Based on this quantity we will decide what is the best location for a Distribution center.

B.1.2 Second Dataset

The Dataset [ZipCode, 2012] contains fields like:

- **Zipcode:** 5 digit Zipcode or military postal code(FPO/APO) .
- **ZipCodeType:** Standard, PO BOX Only, Unique, Military(implies APO or FPO).
- **City:** USPS official city name(s) .
- **State:** USPS official state, territory, or quasi-state (AA, AE, AP) abbreviation code.
- **LocationType:** Primary, Acceptable, Not Acceptable.
- **Lat:** Decimal Latitude, if available.
- **Long:** Decimal Longitude, if available.
- **Location:** Standard Display (eg Phoenix, AZ ; Pago Pago, AS ; Melbourne, AU) .
- **Decommisioned:** If Primary location, Yes implies historical Zipcode, No Implies current Zipcode; If not Primary, Yes implies Historical Placename.
- **TaxReturnsFiled:** Number of Individual Tax Returns Filed in 2008.
- **EstimatedPopulation:** Tax returns filed + Married filing jointly + Dependents.
- **TotalWages:** Total of Wages Salaries and Tips

B.2 Scripts

The Script B.1 generate a random five digit number to present a zip code, the second col is for a random number between 1 and 100 to present the quantity that was ordered.

```
Program.cs* [X]
Script
1  using System;
2  using System.Collections.Generic;
3  using System.IO;
4  using OfficeOpenXml;
5
6  namespace Script
7  {
8      class Program
9      {
10         static void Main(string[] args)
11         {
12             using (ExcelPackage excel = new ExcelPackage())
13             {
14                 excel.Workbook.Worksheets.Add("Data");
15                 var headerRow = new List<string>()
16                 {
17                     new string[] { "ZipCode", "Quantity" }
18                 };
19
20                 // Determine the header range (e.g. A1:D1)
21                 string headerRange = "A1:" + Char.ConvertFromUtf32(headerRow[0].Length + 64) + "1";
22
23                 // Target a worksheet
24                 var worksheet = excel.Workbook.Worksheets["Data"];
25
26                 // Popular header row data
27                 worksheet.Cells[headerRange].LoadFromArrays(headerRow);
28
29                 int[,] a = new int[100, 100];
30                 int i, j;
31                 Random generator = new Random();
32
33                 for (i = 2; i < 100; i++)
34                 {
35                     String zipcode = generator.Next(0, 99999).ToString("D5");
36                     String quantity = generator.Next(1, 99).ToString();
37                     worksheet.Cells[i, 1].Value = Convert.ToInt32(zipcode);
38                     worksheet.Cells[i, 2].Value = Convert.ToInt32(quantity);
39                 }
40
41                 FileInfo excelFile = new FileInfo(@"C:\Users\fcheb0\Desktop\DataSet.xlsx");
42                 excel.SaveAs(excelFile);
43             }
44         }
45     }
46 }
47
48
```

Figure B.1: Data Set Script

The Script B.2 lists the script commands to read the two data set, one excel and one csv files, then it convert the data set to data frame, to merge the data we use the zip code col, after merging the data we will have a data set with both cols from the first data set one and the second data set. After merging the two data sets the final data can be save as .Rdata or a csv file.

```
install.packages("readxl")
library(readxl)

#Load all the datasets:
DataSet <- read_excel("C:/Users/Fatima Chebchoub/Desktop/Thesis/Dataset/DatasetMoreData.xlsx")
DataSetZip <- read.csv("C:/Users/Fatima Chebchoub/Desktop/Thesis/dataset/free-zipcode-database-Primary.csv")
Facilities <- read.csv("C:/Users/Fatima Chebchoub/Desktop/Thesis/Dataset/Facilities.csv")
DCs <- read.csv("C:/Users/Fatima Chebchoub/Desktop/Thesis/Dataset/DCs.csv")

#Create a dataframe for the datasets and use STR() to provides information about the structure of some object.:
df1 <- as.data.frame(DataSet)
str(df1)

df2 <- as.data.frame(DataSetZip)
str(df2)

head(df1)
head(df2)

#Merge the two datasets using the Zipcode
df <- merge(df1, df2, by.x = c("Zipcode"),by.y = c("Zipcode"))

#show the data
head(df)

#Save the file for future use
save(df, file = "C:/Users/Fatima Chebchoub/Desktop/CS 595/Dataset/df.Rdata")
write.csv(df, file = "C:/Users/Fatima Chebchoub/Desktop/CS 595/Dataset/df.csv", row.names = FALSE)
```

Figure B.2: Data Set Script

The Script B.3 Load the USA map using the Leaflet library.

```
# USA map
leaflet(Facilities) %>%
  addProviderTiles("CartoDB.Positron") %>%
  setView(-98.483330, 38.712046, zoom = 4)
```

Figure B.3: USA Map Script

Script B.4 shows the script to install and load the leaflet package, the script also maps the Lat and Long attributes for all the facilities using the leaflet library.

```
# USA map with the current Facilities:
leaflet(Facilities) %>%
  addProviderTiles("CartoDB.Positron") %>%
  setView(-98.483330, 38.712046, zoom = 4)%>%
  addMarkers(~POINT_X, ~POINT_Y, popup = ~as.character(Name), label = ~as.character(Name))
```

Figure B.4: USA Map with Facilities Script

Script B.5 shows the script to map all the orders locations with the quantities.

```
# USA map with the quantity for each part of the country:
leaflet(df) %>%
  addProviderTiles("CartoDB.Positron") %>%
  setView(-98.483330, 38.712046, zoom = 4)%>%
  addCircles(~Long, ~Lat, weight = 3, radius=40,
            color="#ffa500", stroke = TRUE, fillopacity = 0.8, labeloptions = ~City ) %>%
  addLegend(pal = pal,
            title = "Quantity",
            values = df$Quantity,
            opacity = 0.4,
            position = "bottomright")
```

Figure B.5: USA Map with Quantities script

Script B.6 shows the scrip to add DC's to the map we can see how far the Dc's from our customers.

```
# USA map with the current DC's with the quantity :
leaflet(df) %>%
  addProviderTiles("CartoDB.Positron") %>%
  setView(-98.483330, 38.712046, zoom = 4) %>%
  addMarkers(data =DCs, ~POINT_X, ~POINT_Y, popup = ~as.character(Name), label = ~as.character(Name)) %>%
  addCircles(~Long, ~Lat, weight = 3, radius=40, popup=~City,
            color="#ffa500", stroke = TRUE, fillopacity = 0.8 ) %>%
  addLegend(pal = pal,
            title = "Quantity",
            values = df$Quantity,
            opacity = 0.4,
            position = "bottomright")
```

Figure B.6: DC's with Quantities Script

In figure B.7 we have the code that will create lines between one DC and the top 100 customers (the customer that orders the biggest quantities) to see how far the DC is from our top customers.

```
install.packages("mapview")
install.packages("raster")
library(mapview)
library(raster)

## start point Bowling Green KY
root <- matrix(c(-86.40895504, 36.95536442), ncol = 2)
colnames(root) <- c("Long", "Lat")

## end points

#Sort the dataframe bu quantity (descending)
dfLocation <- df[order(-df$Quantity),]

# After sorting geting the top 20 locations
dfLocation <- dfLocation[1:100,]

#create a new data frame for the locations with just Lt and Long
locations <- data.frame(Long = dfLocation$Long, Lat = dfLocation$Lat)

#delete all the NA if we had any
locations <- locations[complete.cases(locations), ]

## create and append spatial lines
lst <- lapply(1:nrow(locations), function(i) {
  spatialLines(list(Lines(list(Line(rbind(root, locations[i, ]))), ID = i)),
    proj4string = CRS("+init=epsg:4326"))
})

sln <- do.call("bind", lst)

## display data
mapview(sln)
```

Figure B.7: Top 100 Customer locations from One DC's script

The code is using “mapview” package, MapView is an interactive viewing of spatial objects in R. The package provides functionality to view spatial objects interactively. The intention is to provide interactivity for easy and quick visualization during spatial data analysis. It is not intended for fine-tuned presentation quality map production. [CRANR-Project, 2020]

Script B.8 we have the code that will use the library “ggplot2” to create a graph with the total orders for each state. Before we plot the data, we start by grouping by the state and get the number of the orders.

```
library(ggplot2)

##### How many products orders for each state #####
states_orders <- df %>%
  group_by(State) %>%
  summarize(total=n())

#delete all the NA if we have in the dataset
states_orders <- states_orders[complete.cases(states_orders), ]

#using points:
ggplot(states_orders,aes(x=State,y=total)) + geom_point()

# How many customers we have in each state?

#using bars:
ggplot(states_orders,aes(x=State,y=total)) + geom_bar(stat="identity") + labs(y="Total Orders", x = "States")

# using bars with color for each state!
ggplot(data=states_orders) +
  geom_col(mapping=aes(x=State, y= total, fill=State)) + labs(y="Total Orders", x = "States")
```

Figure B.8: Low 100 Customer locations from one DC

The script B.9 show the Bar graph with more clear data is to create the graph as a circle.

```
#####Show the Bar graph as circle

states_orders_state <- df %>%
  group_by(State) %>%
  summarize(total=n())

#delete all the NA if we have in the dataset
states_orders_state <- states_orders_state[complete.cases(states_orders_state), ]

# Set a number of 'empty bar'
empty_bar <- 10

# Add lines to the initial dataset
to_add <- matrix(NA, empty_bar, ncol(states_orders_state))
colnames(to_add) <- colnames(states_orders_state)
states_orders_state <- rbind(states_orders_state, to_add)
states_orders_state$id <- seq(1, nrow(states_orders_state))

#delete all the NA if we have in the dataset
states_orders_state <- states_orders_state[complete.cases(states_orders_state), ]

# Order data:
states_orders_state = states_orders_state %>% arrange(total)

# Get the name and the y position of each label
label_data <- states_orders_state
number_of_bar <- nrow(label_data)
angle <- 90 - 360 * (label_data$id-0.5) /number_of_bar
# I subtract 0.5 because the letter must have the angle of the center of the bars. Not extreme right(1) or extreme left (0)
label_data$hjust <- ifelse( angle < -90, 1, 0)
label_data$angle <- ifelse(angle < -90, angle+180, angle)

# Make the plot
ggplot(states_orders_state, aes(x=as.factor(id), y=total)) +
  # Note that id is a factor. If x is numeric, there is some space between the first bar
  geom_bar(stat="identity", fill=alpha("green", 0.3)) +
  ylim(-100,120) +
  theme_minimal() +
  theme(
    axis.text = element_blank(),
    axis.title = element_blank(),
    panel.grid = element_blank(),
    plot.margin = unit(rep(-1,4), "cm")
  ) +
  coord_polar(start = 0) +
  geom_text(data=label_data, aes(x=id, y=total+10, label=State, hjust=hjust),
    color="black", fontface="bold",alpha=0.6, size=2.5, angle= label_data$angle, inherit.aes = FALSE )
```

Figure B.9: Low 100 Customer locations from one DC

The script in figure B.10 The script will sort the data frame to get the top 20 and low 20, then we will use that new dataframe with the top and low 20 to create the bars graphs.

```

##### Top States
#Sort the dataframe by total so we can get the top 20 states
topStates <- states_orders[order(-states_orders$total),]

# After sorting getting the top 20 states
topStates <- topStates[1:20,]

#using bars:
ggplot(data=topStates) +
  geom_col(mapping=aes(x=State, y= total, fill=State)) + labs(y="Total Orders", x = "Top 20 States")

#Plot bars with order the total customer in each state:
ggplot(data=topStates) +
  geom_col(mapping=aes(x=reorder(State, -total), y= total, fill=State)) +labs(y="Total Orders", x = "Top 20 States")
+theme(axis.text.x=element_text(angle=45, hjust=1))

##### Low States
#Sort the dataframe by total so we can get the top 20 states
lowStates <- states_orders[order(states_orders$total),]

# After sorting getting the top 20 states
lowStates <- lowStates[1:20,]

#using bars:
ggplot(data=lowStates) +
  geom_col(mapping=aes(x=State, y= total, fill=State))+ labs(y="Total Orders", x = "Low 20 States")

#Plot bars with order the total customer in each state:
ggplot(data=lowStates) +
  geom_col(mapping=aes(x=reorder(State, -total), y= total, fill=State)) + labs(y="Total orders", x = "Low 20 States")
+ theme(axis.text.x=element_text(angle=45, hjust=1))

```

Figure B.10: Top 20 and Low 20 states orders script

The script B.11 group by the state and sum the quantity number.

```
##### How much Quantities in each state #####
states_QTS <- df %>%
  group_by(State) %>%
  summarize(total=sum(Quantity))

#delete all the NA if we have in the dataset
states_QTS <- states_QTS[complete.cases(states_QTS), ]

#using points:
ggplot(states_QTS,aes(x=State,y=total)) + geom_point()

# How much quantity deliver for each state?

#using bars:
ggplot(states_QTS,aes(x=State,y=total)) + geom_bar(stat="identity") + labs(y="Sum Quantities", x = "States")

# using bars with color for each state!
ggplot(data=states_QTS) +
  geom_col(mapping=aes(x=State, y= total, fill=State)) + labs(y="Sum Quantities", x = "States")
```

Figure B.11: Total Quantity for each State Script

The script B.12 will sort the data frame to get the top 20 and low 20, then we will use that new data frame with the top and low 20 to create the bars graphs.

```
#####Top States
#Sort the dataframe by quantity so we can get the top 20 states
topQTStates <- states_QTS[order(-states_QTS$total),]

# After sorting getting the top 20 states
topQTStates <- topQTStates[1:20,]

#using bars:
ggplot(data=topQTStates) +
  geom_col(mapping=aes(x=State, y= total, fill=State)) + labs(y="Sum Quantities", x = "Top 20 States")

#Plot bars with order the total customer in each state:
ggplot(data=topQTStates) +
  geom_col(mapping=aes(x=reorder(State, -total), y= total, fill=State)) + labs(y="Sum Quantities", x = "Top 20 States")
+theme(axis.text.x=element_text(angle=45, hjust=1))

#####Low States
#Sort the dataframe by quantity so we can get the top 20 states
lowQTStates <- states_QTS[order(states_QTS$total),]

# After sorting getting the top 20 states
lowQTStates <- lowQTStates[1:20,]

#using bars:
ggplot(data=lowQTStates) +
  geom_col(mapping=aes(x=State, y= total, fill=State)) + labs(y="Sum Quantities", x = "Low 20 States")

#Plot bars with order the total customer in each state:
ggplot(data=lowQTStates) +
  geom_col(mapping=aes(x=reorder(State, -total), y= total, fill=State)) + labs(y="Sum Quantities", x = "Low 20 States")
+ theme(axis.text.x=element_text(angle=45, hjust=1))
```

Figure B.12: Total Quantity for each State Script

The script B.13 uses the Library "plotrix" that will help us with labelling and color. We will use the total number of quantities for the top 10 states and how the total quantities percentages for each state.

```

# install the package "plotrix" that will help various labeling, axis and color scaling functions.
install.packages("plotrix")
library(plotrix)

#delete all the NA if we had any
dfClean <- df[complete.cases(df), ]

states_QTSPie <- dfClean %>%
  group_by(State) %>%
  summarize(total=sum(Quantity))

# Get random 10 states
states_QTSPie <- sample_n(states_QTSPie, 10)

# Simple Pie Chart
slices <- states_QTSPie$total
lbls <- states_QTSPie$State
pie(slices, labels = lbls, main="Pie Chart of Countries")

# Pie Chart with Percentages
pct <- round(slices/sum(slices)*100)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(slices,labels = lbls, col=rainbow(length(lbls)),
    main="Pie Chart of Countries")

# 3D Exploded Pie Chart
pie3D(slices,labels=lbls,explode=0.1,
      main="Pie Chart of Countries ")

```

Figure B.13: Pie Graphs Script

The Script B.14 calculates the distance between one DC and each location in our dataset. The distance will be generated on Meters and we have to convert it to Miles.

```
#copy the dataset to a new dataset and create a new col distance
df_final <- df
df_final$distance <- 0

## start point root Bowling Green KY
root <- data.frame(Long = -86.40895504, Lat = 36.95536442)
colnames(root) <- c("Long", "Lat")

#loop through all the dataset and add value to the distance col(convert the unit from meters to miles)
for (i in 1:nrow(df)) {
  row <- df_final[i,]
  n <- distm(c(root$Long, root$Lat), c(row$Long, row$Lat), fun = distHaversine)[,1] / 1609
  df_final$distance[i] <- n
}

#delete all the NA if we had any
df_final <- df_final[complete.cases(df_final), ]

#show the result
df_final
|

# Get random unique rows for each state
df_final <- ddply(df_final,.(df_final$State),
                 function(x) {
                   x[sample(nrow(x),size=1),]
                 })
#set the state as the index name for the dataset so dochart will show the Yaxis as the states
rownames(df_final) <- df_final$State

#Sort the dataframe bu distance (descending)
df_final <- df_final[order(-df_final$distance),]

#Plot the graph
dotchart(df_final$distance,labels=row.names(df_final),cex=.7,
         ylab="States",
         xlab="Distance from BG, KY")
```

Figure B.14: Distances between one DC and each State Script

The script B.15 shows hoe to create the network between the current DCs and the cities that shipped to.


```

# library
library(igraph)

#create a dataframe with only the name of the DC
dc <- as_data_frame(DCs$Name)

#duplicate the data for the dataframe DCS
dc <- rbind(dc, dc[rep(sample(nrow(dc), 10), 10), ])

#create a dataframe with only the city
dfNetwork <- as_data_frame(df$City)

#delete all the NA if we had any
dfNetwork <- dfNetwork[complete.cases(dfNetwork), ]

#Get random 30 Dcs and cities
dc <- sample_n(dc, 30)
dfNetwork <- sample_n(dfNetwork, 30)

#Merge the two data frames|
dfNetworkFinal <- data.frame(cbind(S=dfNetwork, T=dc))

# create the network object
network <- graph_from_data_frame(d=dfNetworkFinal, directed=TRUE)

# plot it
plot(network)

```

Figure B.15: Network Between DCs and Cities Script

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
3PL	Third-party logistics
API	Application programming interface
APO	Army Post Office
APS	Advanced planning and scheduling
DC	Distribution Center
E-comm	e-Commerce
ERP	Enterprise resource planning
FPO	Fleet Post Office
IT	Information Technology
LTL	Less-than-truckload shipping
RHEL	Red Hat Enterprise Linux
SCP	Special Containment Procedures or Secure
SKU	Stock keeping unit
US	United States
VM	Virtual Machine