*Cleaning and Storing Supply Chain Delivery Data*

DSCS6020 Collect/Store/Retrieve Data

Abhiram Paruchuri

100933135

Introduction

Project Proposal:

The proposed project idea is to clean a supply chain delivery data set, obtained from The United States President’s Emergency Plan for Aids Relief (pepfar). The organization’s main goal is to provide better health care and testing facilities to the places that are lacking them. The data consists of all the deliveries made by the pepfar delivery network over the past few years. Cleaning and preprocessing the data can help in using the data to observe trends in the medical needs and help in effectively predicting the inventory. As supply chain is a constantly evolving process, effectively forecasting the changes and trends can help in developing a better plan to handle the needs. In this case, it is the HIV medications and test kits. In addition, storing the dataset into a database can make visualizing the data easier.

In addition, I have also retrieved the HIV/AIDS prevalence rates, deaths caused by AIDS, number of people infected from the data on the size of HIV epidemic from WHO global health observatory data repository. The idea is to collect all the available data on the epidemic and combine it into a single repository. This will make analysis easier and improve the scope of the supply chain network by providing answers to the questions like, which is the most affected country?, how many are in need of medical aid? Etc.

Brief introduction to prepfar:

The U.S. President's Emergency Plan for AIDS Relief (PEPFAR) is the U.S. Government initiative to help save the lives of those suffering from HIV/AIDS around the world. This historic commitment is the largest by any nation to combat a single disease internationally, and PEPFAR investments also help alleviate suffering from other diseases across the global health spectrum. PEPFAR is driven by a shared responsibility among donor and partner nations and others to make smart investments to save lives.

PEPFAR is the cornerstone and largest component of the U.S. President’s Global Health Initiative. With a special focus on improving the health of women, newborns and children, the Global Health Initiative’s goal is to save the greatest number of lives by increasing and building upon what works and, then, supporting countries as they work to improve the health of their own people.

Brief introduction to SCMS:

The SCMS aim is to provide cost-effective, reliable, secure and sustainable supply chains for developing countries, which can be a huge thing for people suffering with HIV. For more than six years, the Supply Chain Management System (SCMS) has been saving lives through stronger supply chains. In collaboration with in-country and international partners, SCMS:

* Provides quality, best-value health care products to those who need them.
* Deploys innovative solutions to help programs enhance their supply chain capacity.
* Ensures accurate supply chain information is collected, shared and used.

Cleaning the Data

Dataset Overview:

There are two data sets that are being derived. They are:

1. The SCMS scheduled delivery data set
2. Dataset created from WHO country statistics.

The SCMS scheduled delivery dataset:

First, we deal with the SCMS scheduled delivery data set, which comprises of procurement transaction data from the Supply Chain Management System (SCMS), administered by the United States Agency for International Development (USAID), i.e. data about all the deliveries made, which are related to medical needs and facilities like, drugs, tests and suspension kits.

The dataset captures deliveries of antiretroviral (ARV)drugs, rapid diagnostic tests for HIV and malaria , and anti-malaria medicines, including prices and delivery destinations. This data is particularly valuable for understanding ranges and trends in pricing as well as

First we will review the glossary of the data set and talk about the problems we have with it.

|  |  |  |
| --- | --- | --- |
| **ID** | **FieldName** | **FieldDescription** |
| 1 | ID | Primary key identifier of the line of data in our analytical tool |
| 2 | Project Code | Project code |
| 3 | PQ # | Price quote (PQ) number |
| 4 | PO # | Order number: Purchase order (PO) for Direct Drop deliveries, or Sales Order (SO) for from Regional Delivery Center (RDC) deliveries |
| 5 | ASN/DN # | Shipment number: Advanced Shipment Note (ASN) for Direct Drop deliveries, or Delivery Note (DN) for from RDC deliveries |
| 6 | Country | Destination country |
| 7 | Managed By | SCMS managing office: either the Program Management Office (PMO) in the U.S. or the relevant SCMS field office |
| 8 | Fulfill Via | Method through which the shipment was fulfilled: via Direct Drop from vendor or from stock available in the RDCs |
| 9 | Vendor INCO Term | The vendor INCO term (also known as International Commercial Terms) for Direct Drop deliveries |
| 10 | Shipment Mode | Method by which commodities are shipped |
| 11 | PQ First Sent to Client Date | Date the PQ is first sent to the client |
| 12 | PO Sent to Vendor Date | Date the PO is first sent to the vendor |
| 13 | Scheduled Delivery Date | Current anticipated delivery date |
| 14 | Delivered to Client Date | Date of delivery to client |
| 15 | Delivery Recorded Date | Date on which delivery to client was recorded in SCMS information systems |
| 16 | Product Group | Product group for item, i.e. ARV, HRDT |
| 17 | Sub Classification | Identifies relevant product sub classifications, such as whether ARVs are pediatric or adult, whether a malaria product is an artemisinin-based combination therapy (ACT), etc. |
| 18 | Vendor | Vendor name |
| 19 | Item Description | Product name and formulation from Partnership for Supply Chain Management (PFSCM) Item Master |
| 20 | Molecule/Test Type | Active drug(s) or test kit type |
| 21 | Brand | Generic or branded name for the item |
| 22 | Dosage | Item dosage and unit |
| 23 | Dosage Form | Dosage form for the item (tablet, oral solution, injection, etc.). |
| 24 | Unit of Measure (Per Pack) | Pack quantity (pills or test kits) used to compute unit price |
| 25 | Line Item Quantity | Total quantity (packs) of commodity per line item |
| 26 | Line Item Value | Total value of commodity per line item |
| 27 | Pack Price | Cost per pack (i.e. month's supply of ARVs, pack of 60 test kits) |
| 28 | Unit Price | Cost per pill (for drugs) or per test (for test kits) |
| 29 | Manufacturing Site | Identifies manufacturing site for the line item for direct drop and from RDC deliveries |
| 30 | First Line Designation | Designates if the line in question shows the aggregated freight costs and weight associated with all items on the ASN/DN |
| 31 | Weight (Kilograms) | Weight for all lines on an ASN/DN |
| 32 | Freight Cost (USD) | Freight charges associated with all lines on the respective ASN/DN |
| 33 | Line Item Insurance (USD) | Line item cost of insurance, created by applying an annual flat rate (%) to commodity cost |

Problems faced with the dataset:

As this is the data that is made available for public, there are some key limitations to the data. Multiple deliveries are consolidated into single delivery, so freight cost is inaccurate relative to the line item. The dates are not properly documented. As supply chain data vary for different transactions and the data changes overtime. So NA values are a problem. Some dates are not applicable as the goods for those order are fulfilled out of stock available at RDC (Regional Development Center).The Line Item Value/Pack Price/Unit Price values have high variability as they can be sometimes low due to donations.

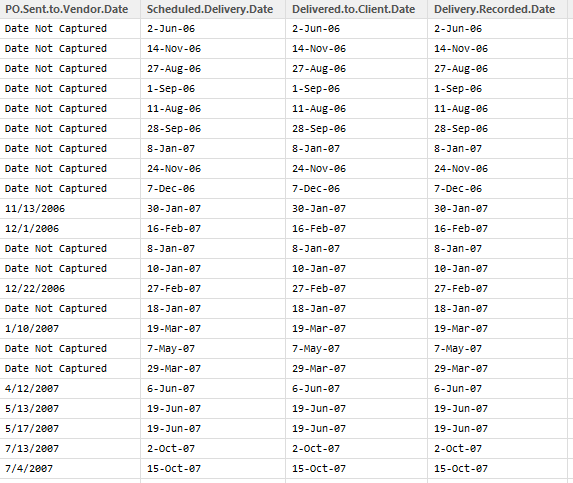
Cleaning the data:

First we clean the columns with the date values. U sing the lubridate package, we convert all the values in the date columns to date classes. The columns (PO.Sent.to.Venfor.Date, Sheduled.Delivery.Date, Delivered.to .Client.Date) are converted to numeric values, so as to find the delivery time and the delay time by subtracting the PO sent date and scheduled date from the delivered date. If the delayed time value is a negative vale, then the product arrived earlier than expected.

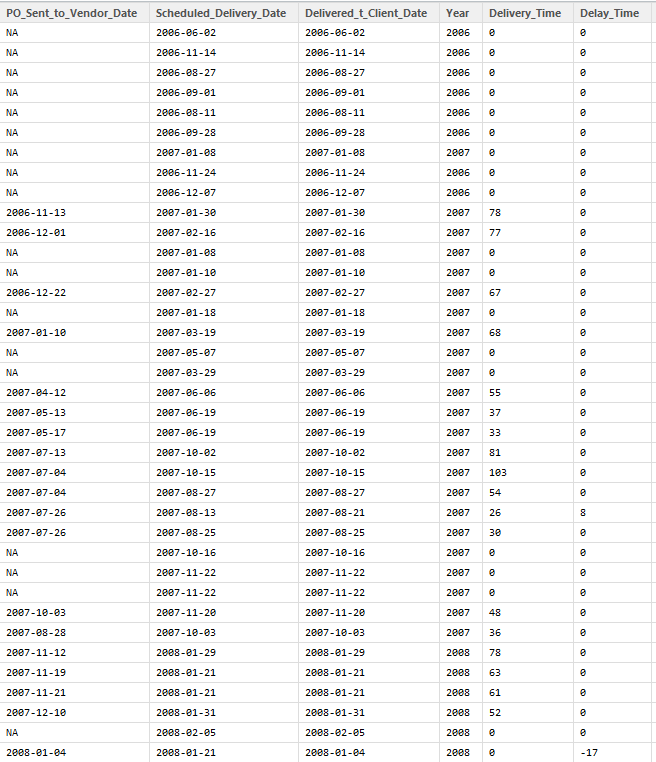
In the program, it is done by constructing a deriveDates() function

Screen shots:

Result:



*The dates before cleaning*

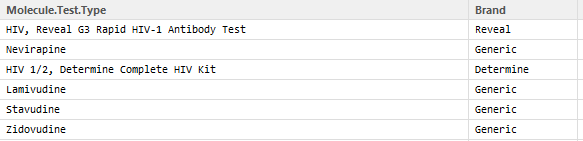
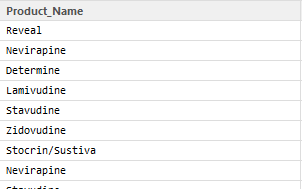


*The cleaned dataset for dates*

Assigning the Product name:

Now we assign a proper name to the product from the ‘Molecule Test Type’ column and the Brand name column. We parse through both the columns, and choose the brand name when it is present and choose the molecule name when the value is listed as generic as it means that the molecule name is generally the product name.

Result:

*A .The above figure has values before merging B. values after merging*

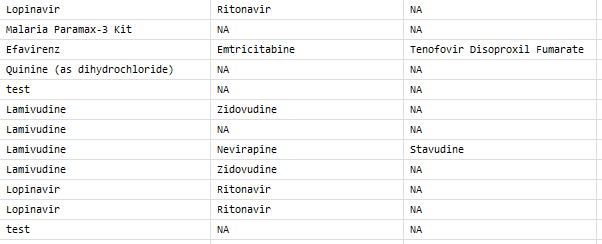
Splitting the drug into its parts:

Several drugs contain two or more types of molecules in them. By constructing the splitElements() function, we split the Molecules and dosages into separate columns. The use of splitting the drugs into their classes is to increase the scope of analysis on the drug type and combination.

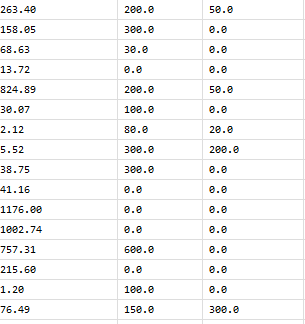
Result:



*The above table consists of molecule names and drug dosages before cleaning*



*Molecules split into their respective columns*



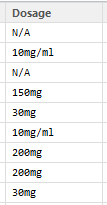
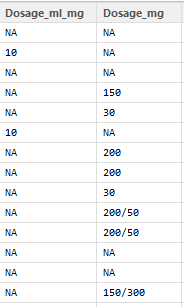
*Dosages split into their respective columns*

Assigning the dosages to their respective units & donation status:

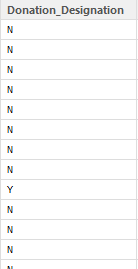
As many data analysis tools are sensitive to units of measure, we split the dosages in the column, which are jumbles into mg/ml, mg and g values to their respective unit columns

We also construct a new column to decide if the line item is a donation or not. The condition set is, if the pack price is less than 2.5 dollars, we designate it as a donation.

Result:

*Dosage Column before After*



*Donation Column*

Final note:

Finally the cleaned data is loaded into a data frame that can be loaded into a database. Some of the columns with redundant values and useless columns are left out while populating the new data frame as they are considered less important for data analysis or further inference.

Not considering the columns above, the other columns that are considered are, ID, project code, PO.SO, ASN, Country, Managed, Fulfill via, shipment mode, vendor, unit of measure per pack, Item Quantity, Pack price, first line designation, weight in kilograms, insurance.

The NA values are converted to zeros rather than omitting as they are very widely dispersed.

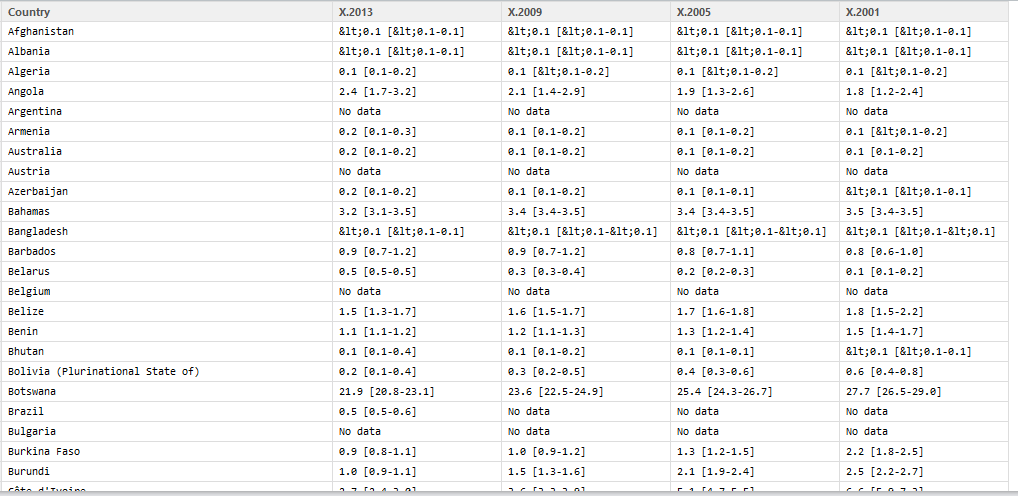
Second dataset retrieval:

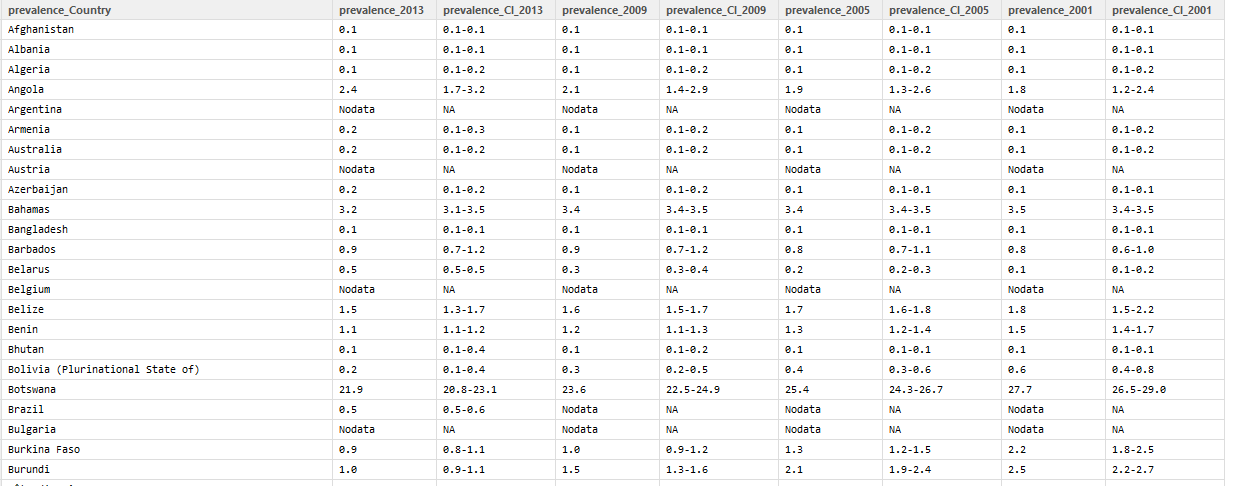
The second data set is constructed using three datasets that are downloaded in program from the WHO Global Health Observatory Data Repository. The cleaning is done in the following steps:

1. The data sets for country wise HIV prevalence, deaths and number of people infected are downloaded and opened in R in the form of a data frame.
2. We observe that all the columns are polluted with ‘&lt’ values, which might have occurred during retrieving.
3. We parse through each values in the three data frames and remove the ‘&lt’ values from the values.
4. Now we notice that the values for the prevalence, deaths and the number of people infected and their confidence interval values are present in the same columns.
5. We construct a function numSplit() to separate these values into two separate columns, while removing the white spaces and the square brackets present in the value.
6. Now we merge all the three datasets in to a single data set using merge() function. First prevalence and deaths data are merged on prevalence country column and then the merged data is merged with number of infected people data set with respect to the same prevalence country column so that we have a common binder.
7. NA values are removed using Na.omit() function as they are not dispersed , but are present in the form of rows.
8. All the data cleaning is done by the dataClean() function the code. The rename() function renames the data after cbind.

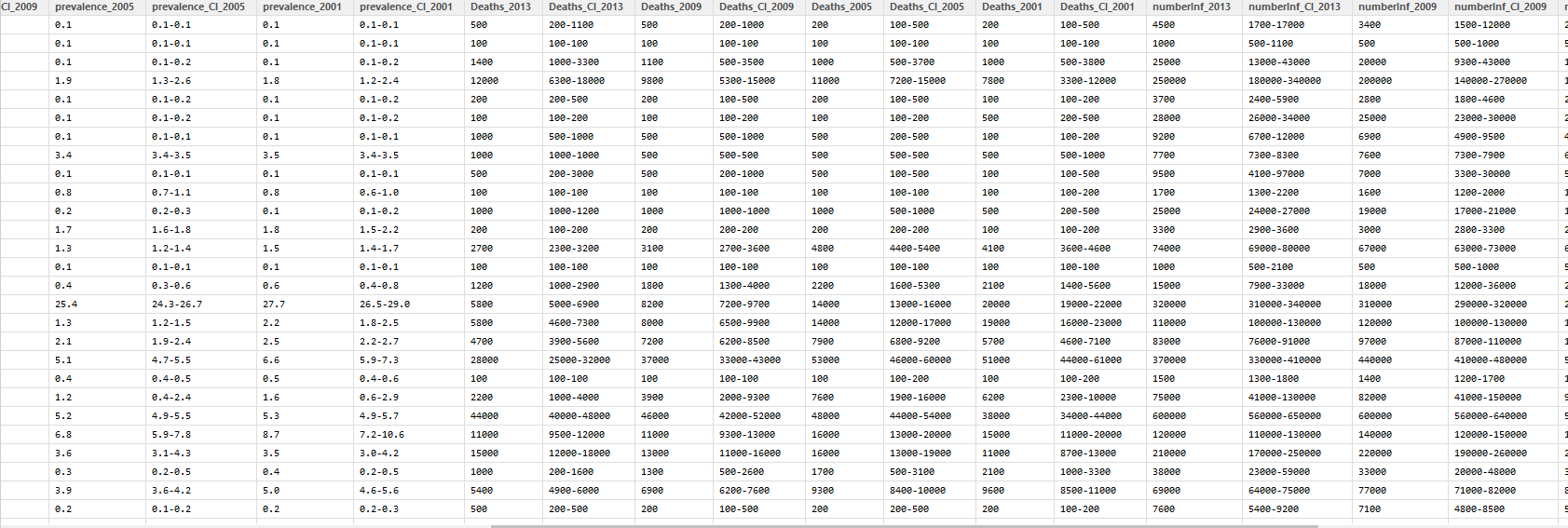
Screen shots:

The screen shots are provided only for the prevalence data set. It is the same for the other data sets also.

*Prevalence dataset before cleaning*



*Prevalence Dataset after cleaning*



*Sample of merged dataset (NA omitted)*

Creating a NoSQL database

A columnar type NoSQL database (mongoDB) is used to store the two retrieved data sets and we perform a set of queries on the databases to see that all the data is loaded correctly. Two data bases are used for storage as the two data sets are of a completely different purposes but have the same aim. So the data retrieved from the two databases simultaneously can give insight to new results as one has the medical supply data and the other has the statistical facts.

Code:

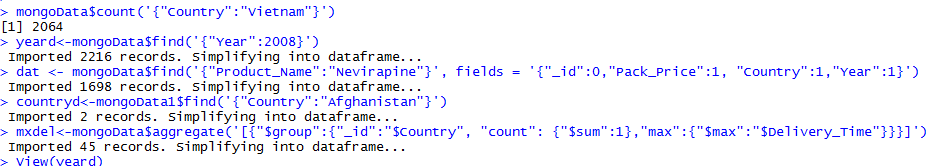
mongoData<-mongo("SCMS")

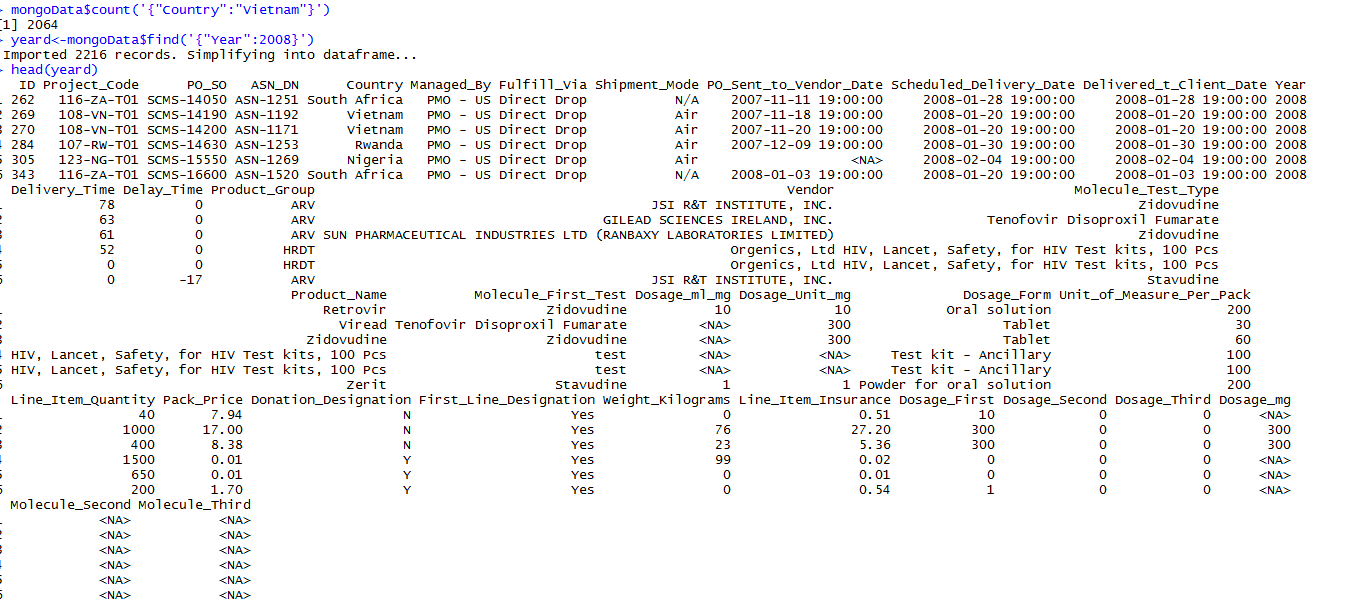
mongoData$insert(supplyData)

mongoData1<-mongo("HIVdata")

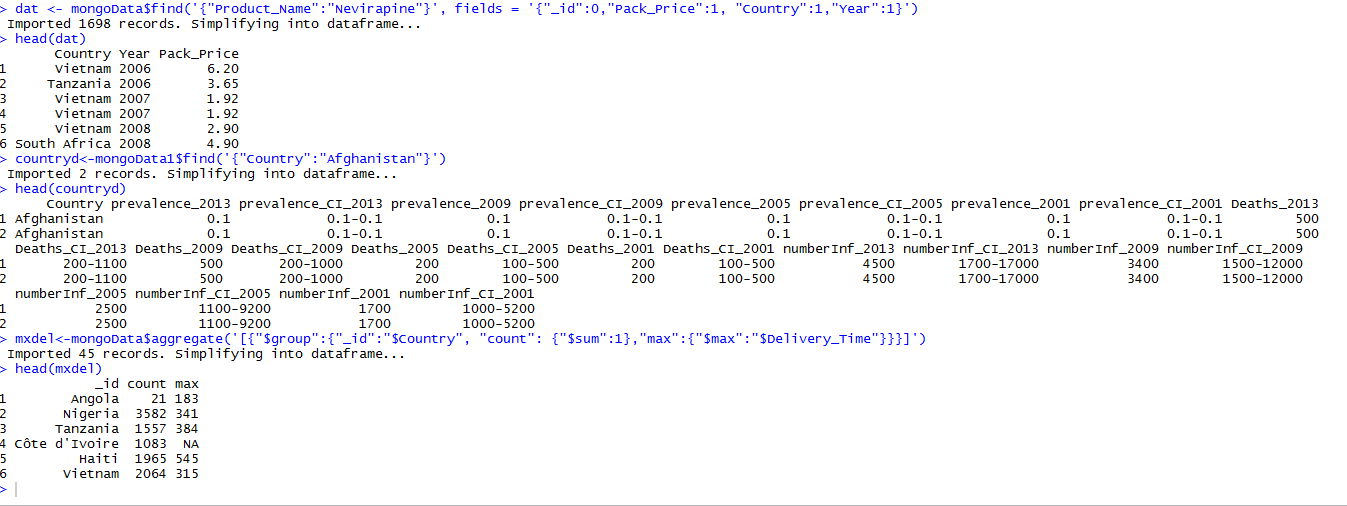
mongoData1$insert(finalData)

Queries:



Queries with head data:

*Queries being run to test the database*

 *Continuation of test Queries*

Problems Faced:

The main problems that I faced is when I tried to combine the two data sets that I retrieved. I have tried very hard to somehow embed into each other, but then I could only complete the process till the merging of the three datasets retrieved from WHO data. I think the main problem lies in the fact that a nested loop is necessary to embed the HIV statistics in to the supply chain data by inspecting each element and then merging them. The process took a lot of time and I finally abandoned that idea to create two separate datasets

Conclusion:

* Hence we can see that the supply chain data which is available publicly can be used for conducting effecting analytics if proper preprocessing is performed on the data.
* We also realize the supply chain data can also be linked to various factors in a given environment.
* If properly realized, the HIV medical supplies supply chain network can be properly optimized to the HIV prevalence rate of a country, so that the people who need help the most will receive it. But this cannot be achieve without further improvement in the data cleaning and acquisition systems.

References:

1. <https://cran.r-project.org>
2. <http://scms.pfscm.org/scms/about>
3. <http://www.pepfar.gov/>
4. <http://www.who.int/en/>
5. Class Notes ‘Collecting, Storing and Retrieving Data’-Yatish Jain and Martin Schedlbauer