Analysis of Healthcare Provided by Medicare

Project By:

Neha Parulekar

DSCS6020 17374 Collect/Store/Retrieve Data SEC 01 Fall 2015

Project Proposal.

The main aim of the project is to analyze the no. of Hospitals that are covered under the Medicare throughout the states in US and what facilities they provide. The dataset is based on the survey that was done in the year of 2014, where patients were asked to give there feedback based on the facilities they visited and the services they took here at the facilities. The dataset also covers the extra services that are provided by these hospitals.

The dataset is a CSV file. I first loaded this dataset into R to sort out data and remove a few unwanted data present and then loaded it into MongoDB to get some quires from the dataset.

About the Data:

In the United States, Medicare is a national [social insurance](https://en.wikipedia.org/wiki/Social_insurance) program, administered by the [U.S. federal government](https://en.wikipedia.org/wiki/United_States_Department_of_Health_and_Human_Services) since 1966, currently using about 30 private insurance companies across the United States. Medicare provides health insurance for Americans aged 65 and older who have worked and paid into the system. It also provides health insurance to younger people with disabilities, [end stage renal disease](https://en.wikipedia.org/wiki/End_Stage_Renal_Disease_Program) and [amyotrophic lateral sclerosis](https://en.wikipedia.org/wiki/Amyotrophic_lateral_sclerosis).

Medicare.gov provides the data to analyze and compare. The website is The Socrata Open Data API (SODA) allows software developers to access data hosted in Socrata sites such as Data.Medicare.gov. In this platform, every dataset is automatically provided with a simple Application Programming Interface, or API (i.e. API endpoint), ensuring access to every dataset. The SODA API supports a number of different formats, including JSON, XML, CSV, and RDF. To change the output format of a request, simply change the extension you use on the resource. The JSON format is the most commonly used format for API responses, as it is the most compact and efficient format provided by Socrata. The SODA API supports a number of different formats, including JSON, XML, CSV, and RDF. To change the output format of a request, simply change the extension you use on the resource. The JSON format is the most commonly used format for API responses, as it is the most compact and efficient format provided by Socrata.

The dataset HHCompare.state is a plain CSV file that contains information about Hospitals across the United States that support Medicare. It gives a detailed information about the location of the hospitals, with there address, the CMS number, the facilities they provide along with the verification of the facilities by the patients. It also gives information about the type of ownership of the hospital along with the address and the phone number.

The data can be used by the patient seeking the information about different hospitals in their area. As the data covers all the states and provides detailed information, patient can look into it with the zip codes to get to the nearest to their Hospitals. The data can also be used to compare different hospitals for the facilities they provided according to the need.

Loading the data:

Loading the data into R:

*#################################################################################*

*# FINAL PROJECT SUBMISSION*

*# Neha Parulekar*

*################################################################################*

*# get the working directory*

[**getwd**](http://inside-r.org/r-doc/base/getwd)()

*# set the working directory according to the file location on your desktop*

[**setwd**](http://inside-r.org/r-doc/base/setwd)("C:/Users/Neha/Desktop")

*# check for the package and istall new packages if required*

checkpackage <- [**function**](http://inside-r.org/r-doc/base/function)(pkg){

new.pkg <- pkg[!(pkg %in% [**installed.packages**](http://inside-r.org/r-doc/utils/installed.packages)()[, "Package"])]

**if** ([**length**](http://inside-r.org/r-doc/base/length)(new.pkg))

[**install.packages**](http://inside-r.org/r-doc/utils/install.packages)(new.pkg, dependencies = **TRUE**)

[**sapply**](http://inside-r.org/r-doc/base/sapply)(pkg, [**require**](http://inside-r.org/r-doc/base/require), character.only = **TRUE**)

}

packages <- [**c**](http://inside-r.org/r-doc/base/c)( "lubridate", "dplyr")

checkpackage(packages)

*# load the data into R using the read.csv command.*

HospitalDetails <- [**read.csv**](http://inside-r.org/r-doc/utils/read.csv)("HHCompare.State.csv", header = T, sep = ",")

*# convert the data into data frame using as.data.frame Command*

[**df**](http://inside-r.org/r-doc/stats/df) <- [**as.data.frame**](http://inside-r.org/r-doc/base/as.data.frame)(HospitalDetails)

*# get the required data from delete the rows that are not required.*

[**df**](http://inside-r.org/r-doc/stats/df) <- [**df**](http://inside-r.org/r-doc/stats/df)[ -[**c**](http://inside-r.org/r-doc/base/c)(11, 13, 15, 17, 19, 21) ] *# removing unwanted columns from the data frame*

Steps for the code:

*# get the working directory*

*# set the working directory according to the file location on your desktop*

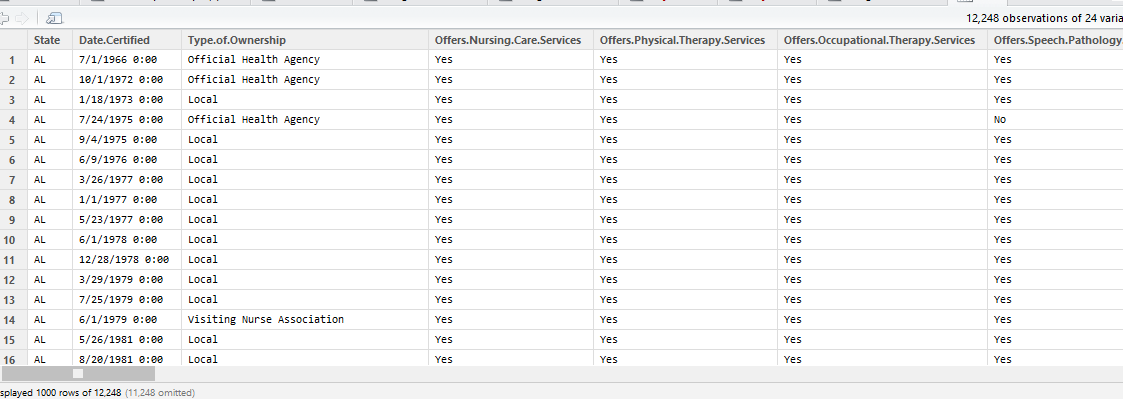
*# Check for the package and install new packages if required*

*# load the data into R using the read.csv command.*

*# convert the data into data frame using as.data.frame Command*

*# get the required data from delete the rows that are not required.*

Screenshot of the Dataset in a DataFrame



Loading the data into MongoDB:

*##### Project on health care #####*

[**rm**](http://inside-r.org/r-doc/base/rm)([**list**](http://inside-r.org/r-doc/base/list) = [**ls**](http://inside-r.org/r-doc/base/ls)())

[**library**](http://inside-r.org/r-doc/base/library)(mongolite)

*# Load data into mongoDB*

loadData <- [**function**](http://inside-r.org/r-doc/base/function) ([**file**](http://inside-r.org/r-doc/base/file)) {

[**df**](http://inside-r.org/r-doc/stats/df) <- [**read.csv**](http://inside-r.org/r-doc/utils/read.csv)([**file**](http://inside-r.org/r-doc/base/file) = [**file**](http://inside-r.org/r-doc/base/file), header = **TRUE**, stringsAsFactors = **FALSE**)

[**colnames**](http://inside-r.org/r-doc/base/colnames)([**df**](http://inside-r.org/r-doc/stats/df)) <- [**gsub**](http://inside-r.org/r-doc/base/gsub)("**\\**.", "", [**colnames**](http://inside-r.org/r-doc/base/colnames)([**df**](http://inside-r.org/r-doc/stats/df)))

mongoData <- mongo(collection = "data", db = "Health")

mongoData$insert([**df**](http://inside-r.org/r-doc/stats/df))

[**return**](http://inside-r.org/r-doc/base/return) (mongoData)

}

*##### Analysis #####*

[**file**](http://inside-r.org/r-doc/base/file) <- "HHCompare.State.csv"

mongoData <- loadData([**file**](http://inside-r.org/r-doc/base/file))

*# Get distinct types of ownersips*

mongoData$distinct(key = "TypeofOwnership")

*# Get count of agencies offering physical therapy services*

mongoData$count('{"OffersPhysicalTherapyServices": "Yes"}')

*# Get count of agencies offering medical social services*

mongoData$count('{"OffersMedicalSocialServices":"Yes"}')

*# Get agencies offering occupational therapy service*

mongoData$distinct(key = "TypeofOwnership", query = '{"OffersOccupationalTherapyServices":"Yes"}')

*# Get distinct provider names*

mongoData$distinct(key = "ProviderName")

*# Get list of providers in MA*

mongoData$distinct(key = "TypeofOwnership", query = '{"State": "MA"}')

steps for the code:

*# Load data into mongoDB*

*##### Analysis #####*

*# Get distinct types of ownersips*

*# Get count of agencies offering physical therapy services*

*# Get count of agencies offering medical social services*

*# Get agencies offering occupational therapy service*

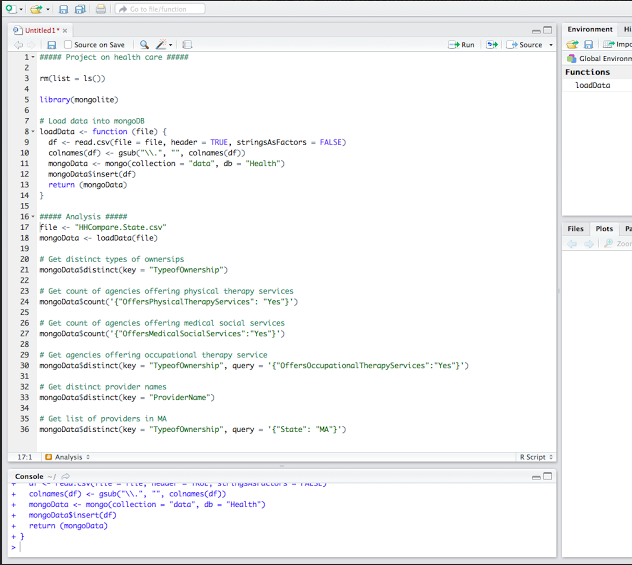
*# Get distinct provider names*

*# Get list of providers in MA*

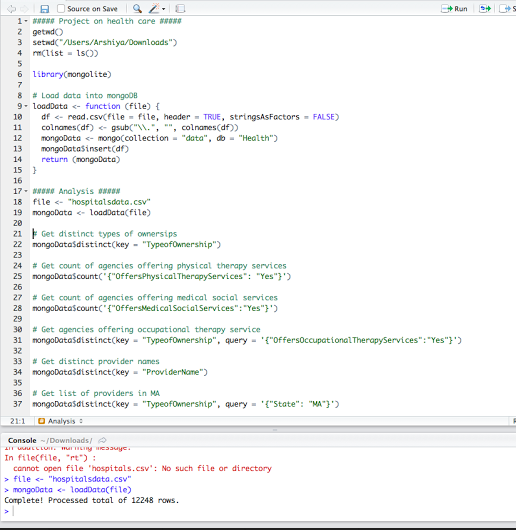
Screenshots of the program:

Below attached are a few screen shots of the outputs obtained

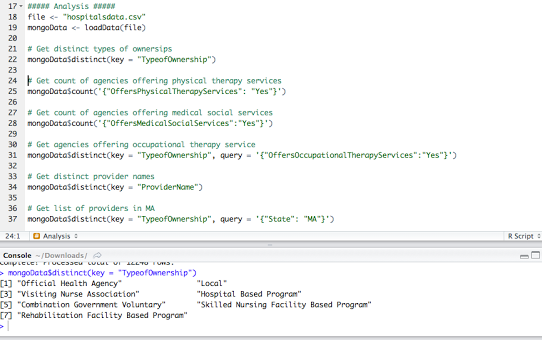
Loading the data into mongoDB:



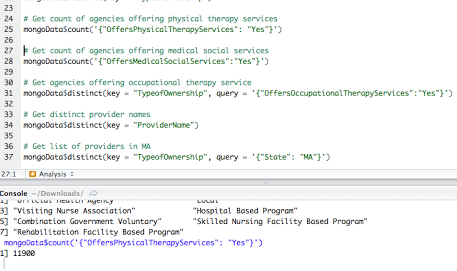
Total no. of rows processed:



Analysis:



To get the count of agensies offering physical therapy services:



These are a few quires obtained using the databases. These quires can be done state wise, area wise, or throughout the country.