

# 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 program, administered by the U.S. federal government 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 and 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()  
# set the working directory according to the file location on your desktop  
setwd("C:/Users/Neha/Desktop")  
  
# check for the package and install new packages if required  
  
checkpackage <- function(pkg){  
  new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]  
  if (length(new.pkg))  
    install.packages(new.pkg, dependencies = TRUE)  
  sapply(pkg, require, character.only = TRUE)  
}  
  
packages <- c("lubridate", "dplyr")  
checkpackage(packages)  
  
# load the data into R using the read.csv command.  
HospitalDetails <- read.csv("HHCompare.State.csv", header = T, sep = ",")  
  
# convert the data into data frame using as.data.frame Command  
df <- as.data.frame(HospitalDetails)  
  
# get the required data from delete the rows that are not required.  
df <- df[ -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

12,248 observations of 24 variables

	State	Date.Certified	Type.of.Ownership	Offers.Nursing.Care.Services	Offers.Physical.Therapy.Services	Offers.Occupational.Therapy.Services	Offers.Speech.Pathology
1	AL	7/1/1966 0:00	Official Health Agency	Yes	Yes	Yes	Yes
2	AL	10/1/1972 0:00	Official Health Agency	Yes	Yes	Yes	Yes
3	AL	1/18/1973 0:00	Local	Yes	Yes	Yes	Yes
4	AL	7/24/1975 0:00	Official Health Agency	Yes	Yes	Yes	No
5	AL	9/4/1975 0:00	Local	Yes	Yes	Yes	Yes
6	AL	6/9/1976 0:00	Local	Yes	Yes	Yes	Yes
7	AL	3/26/1977 0:00	Local	Yes	Yes	Yes	Yes
8	AL	1/1/1977 0:00	Local	Yes	Yes	Yes	Yes
9	AL	5/23/1977 0:00	Local	Yes	Yes	Yes	Yes
10	AL	6/1/1978 0:00	Local	Yes	Yes	Yes	Yes
11	AL	12/28/1978 0:00	Local	Yes	Yes	Yes	Yes
12	AL	3/29/1979 0:00	Local	Yes	Yes	Yes	Yes
13	AL	7/25/1979 0:00	Local	Yes	Yes	Yes	Yes
14	AL	6/1/1979 0:00	Visiting Nurse Association	Yes	Yes	Yes	Yes
15	AL	5/26/1981 0:00	Local	Yes	Yes	Yes	Yes
16	AL	8/20/1981 0:00	Local	Yes	Yes	Yes	Yes

splayed 1000 rows of 12,248 (11,248 omitted)

## Loading the data into MongoDB:

```
##### Project on health care #####

rm(list = ls())

library(mongolite)

# Load data into mongoDB
loadData <- function (file) {
  df <- read.csv(file = file, header = TRUE, stringsAsFactors = FALSE)
  colnames(df) <- gsub("\\.", "", colnames(df))
  mongoData <- mongo(collection = "data", db = "Health")
  mongoData$insert(df)
  return (mongoData)
}

##### Analysis #####
file <- "HHCompare.State.csv"
mongoData <- loadData(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 ownerships
# 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:

```
1 ##### Project on health care #####
2
3 rm(list = ls())
4
5 library(mongolite)
6
7 # Load data into mongoDB
8 loadData <- function(file) {
9   df <- read.csv(file = file, header = TRUE, stringsAsFactors = FALSE)
10  colnames(df) <- gsub("\\.", "", colnames(df))
11  mongoData <- mongo(collection = "data", db = "Health")
12  mongoData$insert(df)
13  return(mongoData)
14 }
15
16 ##### Analysis #####
17 file <- "HHCCompare.State.csv"
18 mongoData <- loadData(file)
19
20 # Get distinct types of ownerships
21 mongoData$distinct(key = "TypeofOwnership")
22
23 # Get count of agencies offering physical therapy services
24 mongoData$count('{"OffersPhysicalTherapyServices": "Yes"}')
25
26 # Get count of agencies offering medical social services
27 mongoData$count('{"OffersMedicalSocialServices": "Yes"}')
28
29 # Get agencies offering occupational therapy service
30 mongoData$distinct(key = "TypeofOwnership", query = '{"OffersOccupationalTherapyServices": "Yes"}')
31
32 # Get distinct provider names
33 mongoData$distinct(key = "ProviderName")
34
35 # Get list of providers in MA
36 mongoData$distinct(key = "TypeofOwnership", query = '{"State": "MA"}')
```

Console:

```
+ df <- read.csv(file = file, header = TRUE, stringsAsFactors = FALSE)
+ colnames(df) <- gsub("\\.", "", colnames(df))
+ mongoData <- mongo(collection = "data", db = "Health")
+ mongoData$insert(df)
+ return(mongoData)
+ }
+ >
```

Total no. of rows processed:

```
1- ##### Project on health care #####
2 getwd()
3 setwd("~/Users/Arshiya/Downloads")
4 rm(list = ls())
5
6 library(mongolite)
7
8 # Load data into mongoDB
9 loadData <- function (file) {
10   df <- read.csv(file = file, header = TRUE, stringsAsFactors = FALSE)
11   colnames(df) <- gsub("\\.", "", colnames(df))
12   mongoData <- mongo(collection = "data", db = "Health")
13   mongoData$insert(df)
14   return (mongoData)
15 }
16
17- ##### Analysis #####
18 file <- "hospitalsdata.csv"
19 mongoData <- loadData(file)
20
21 # Get distinct types of ownerships
22 mongoData$distinct(key = "TypeofOwnership")
23
24 # Get count of agencies offering physical therapy services
25 mongoData$count('{"OffersPhysicalTherapyServices": "Yes"}')
26
27 # Get count of agencies offering medical social services
28 mongoData$count('{"OffersMedicalSocialServices": "Yes"}')
29
30 # Get agencies offering occupational therapy service
31 mongoData$distinct(key = "TypeofOwnership", query = '{"OffersOccupationalTherapyServices": "Yes"}')
32
33 # Get distinct provider names
34 mongoData$distinct(key = "ProviderName")
35
36 # Get list of providers in MA
37 mongoData$distinct(key = "TypeofOwnership", query = '{"State": "MA"}')
```

21:1 Analysis :

```
Console ~/Downloads/
> all works fine, but using read.csv:
In file(file, "rt") :
  cannot open file 'hospitals.csv': No such file or directory
> file <- "hospitalsdata.csv"
> mongoData <- loadData(file)
Complete! Processed total of 12248 rows.
>
```

Analysis:

```
17- ##### Analysis #####
18 file <- "hospitalsdata.csv"
19 mongoData <- loadData(file)
20
21 # Get distinct types of ownerships
22 mongoData$distinct(key = "TypeofOwnership")
23
24 # Get count of agencies offering physical therapy services
25 mongoData$count('{"OffersPhysicalTherapyServices": "Yes"}')
26
27 # Get count of agencies offering medical social services
28 mongoData$count('{"OffersMedicalSocialServices": "Yes"}')
29
30 # Get agencies offering occupational therapy service
31 mongoData$distinct(key = "TypeofOwnership", query = '{"OffersOccupationalTherapyServices": "Yes"}')
32
33 # Get distinct provider names
34 mongoData$distinct(key = "ProviderName")
35
36 # Get list of providers in MA
37 mongoData$distinct(key = "TypeofOwnership", query = '{"State": "MA"}')
```

24:1 Analysis :

```
Console ~/Downloads/
Complete! Processed total of 12248 rows.
> mongoData$distinct(key = "TypeofOwnership")
[1] "Official Health Agency"          "Local"
[3] "Visiting Nurse Association"      "Hospital Based Program"
[5] "Combination Government Voluntary" "Skilled Nursing Facility Based Program"
[7] "Rehabilitation Facility Based Program"
>
```

To get the count of agencies offering physical therapy services:

```
23
24 # Get count of agencies offering physical therapy services
25 mongoData$count({'OffersPhysicalTherapyServices': 'Yes'})
26
27 # Get count of agencies offering medical social services
28 mongoData$count({'OffersMedicalSocialServices': 'Yes'})
29
30 # Get agencies offering occupational therapy service
31 mongoData$distinct(key = "TypeofOwnership", query = '{"OffersOccupationalTherapyServices": "Yes"}')
32
33 # Get distinct provider names
34 mongoData$distinct(key = "ProviderName")
35
36 # Get list of providers in MA
37 mongoData$distinct(key = "TypeofOwnership", query = '{"State": "MA"}')
```

27:1 Analysis :

Console ~/Downloads/ ↗

1] Official Health Agency	Local
3] "Visiting Nurse Association"	"Hospital Based Program"
5] "Combination Government Voluntary"	"Skilled Nursing Facility Based Program"
7] "Rehabilitation Facility Based Program"	

```
mongoData$count({'OffersPhysicalTherapyServices': 'Yes'})
1] 11900
```

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