Analyzing and Predicting the hospitalization Costs using Data Science - DSC520_FinalProject_Part2

Guruprasad Velikadu Krishnamoorthy

2023-02-19

Contents

1 Introduction

Problem Statement: With the tremendous advancement that has been made in technology in the last few decades, things that used to be considered luxuries such as smartphones or 50-inch television are not very expensive anymore. However, even with all the technological progress one thing that has not changed much is the cost of medical treatment. In fact, many medical procedures have gotten so expensive over the years that some are not affordable, even after having Health Insurance and patients have to go for external sources such as GoFundMe to collect funds. When the doctor suggests a medical treatment and when we call the hospital to ask how much the treatment might cost, there is a good chance the hospital may not be willing to tell the actual amount, considering how complicated the health care system in the US is. Many people get sucked into this mess without completely understanding all the nuances of the healthcare system and end up with a massive bill that they may not be able to pay. An article from CNBC states that two-thirds of all bankruptcies in the US are tied to issues with Medical Bills. With such a huge problem in hand, there is no better solution to get an idea of how much an essential but non-emergency procedure would cost in a hospital. (Konish n.d.)

Why would someone be interested in the research? Medical bills can be very expensive, especially if you don't have insurance or if you get treatment from a hospital that is out of the health insurance network. Most hospitals are not transparent about sharing the treatment costs and they redirect to reach out to health insurance instead. However, not many people know that medical bills can be negotiated with hospitals. If we have the right stats and numbers, the hospitals may be willing to negotiate the bills and it can save thousands of dollars and sometimes can even save bankruptcies for some families. This research focuses on studying the pattern behind the medical costs across various states and cities in the US and identifying a pattern in the costs. Patients can use the data to choose care wisely and avoid expensive hospitals if they have a choice. Also, these numbers can help negotiate the bills with the hospitals.

How Data Science can help with the research? Data science can be used to study patterns and derive statistics and compare prices over a period. Also, data from different datasets can be combined to derive the relationship between the variables and a model can be built to predict the prices in the coming years.

2 Research Questions

1. What is the average cost for treatment, say, for example, Chemotherapy, in each state across the US?
2. What is the total number of inpatient stays across the states for a specific medical procedure during a

given period?

3. How much has the average cost of a medical procedure (for Chemotherapy, as an example) varied over the years across a state, Vermont for instance?
4. What is the average price of Chemotherapy treatment across various cities in Vermont?
5. Given a zip code in Vermont, if multiple hospitals offer the same treatment for Chemotherapy, how different are the costs?
6. Are the hospitals operating for Profit or Non-Profit? Is there a substantial difference in the costs for the same medical procedures between the For-Profit and Non-Profit hospitals? By what percentage?
7. How many hospital beds are available across a city or a state? What are their inpatient occupancy rates? Do "For-Profit" hospitals provide more beds in the state than "Non-Profit"? By how much percent?
8. Is there insight into the Profit margin and other financials of the "for-profit" and "non-profit" hospitals for a given period? How profitable are the "For-Profit" hospitals?
9. Is there any correlation between the number of uninsured people across the state/ region to the average cost of medical treatments across the region?
10. Is there a pattern in the variation in the cost of a medical procedure over a period? Can it predict the prices of treatment in the future?

3 Approach

The problem that this paper address is to study the disparity in hospitalization costs across the country. Though the costs may differ from state to state, many times they are so much different even within the same city or county. Also, hospitals are not transparent about sharing the medical costs, so it can be very frustrating to undergo a medical procedure not knowing how much it costs and if the hospital is overcharging for the procedure or not. The solution addresses this problem by studying the trends in medical costs (for a medical procedure as an example) across the state or region and over a period and provides an overall idea about the differences in the cost. The results will be plotted in graphs and plots for visualization. The study also provides results of costs that are grouped based on a state or city or a region for a medical procedure. This can also be used by the patients to compare the costs with the explanation of benefits to getting a picture of whether the costs look normal or inflated.

4 How the approach addresses (fully or partially) the problem

The approach includes importing the data from different periods and comparing the trends in the costs of medical treatment. We use the average costs in the state as a benchmark to compare the costs and identify how the hospital's costs compare with others in the same region. We study the occupancy rates of hospitals and compare them with the state's average and discover the pattern between inpatient occupancy and their costs. Also, the Tables and plots in R will be used to visualize the results for easy understanding. Also, the costs of treatments may vary between counties and if the difference is substantial, one can choose to undergo treatment from a cheaper hospital from a neighboring county or a neighboring state in some cases.

This project may partially address the problem statement of identifying the affordable hospitals in the region as the study does not include the health insurance data in the research. Also this study uses Medicare data for research and sometimes hospitals may charge a different price for treatments than the medicare costs. Hence the study from the research can be used a baseline for further research. The rates are compared across hospitals with the assumption that they are all within the same network, which may not be true in all cases. This project is to identify the pattern of how hospitals charge different rates for the same treatments in the same region and to predict costs in the future. Also, visualization of the data using tools such as Tableau or Power BI can be a great enhancement for this research which is not included in this project. Visualization can add more features such as the ability to select multiple hospitals and select procedures in the hospitals and compare the costs.

Below is the high-level approach for addressing the research project:

Identifying the datasets : There are a variety of sources to find the datasets such as www.data.cms.gov, www.cdc.gov, www.who.int, www.eia.gov, Kaggle, etc. However, the real datasets on hospitals and the costs of procedures may be a challenge to find. So, the first step is to invest time to find the right dataset.

Exploring the dataset : Once the datasets are found, the second step is to study the data and understand each field in the dataset. We may not find all the relevant data in the same dataset. So, we may have to identify more than one dataset and join them to extract meaningful insights. Hence understanding the data is very important.

Draft a high-level plan : The next step is to come up with a high-level plan (algorithm) for the approach to address the problem. This step will require identifying the fields that we are planning to join across different datasets and analyzing the number of records in the original data to see if a subset of data is required for analysis.

Identifying the technical requirements : The next step is technical where we identify the list of R packages and plots required to address the problem. In this step, we will perform a very high-level analysis of the technical requirements as more packages can be added later during the development stage. Further steps of cleansing data and transforming the data will be done during the development phase.

Development and exploration : After the steps mentioned in the above question are complete, the next steps involve loading the data into R and cleansing and transforming the data. Depending upon the size of the dataset, a subset of data can be used for the research. In this stage, we can identify new fields required to be used in the plots to join the datasets.

5 Datasets

Seven datasets are used in this research and the details are given below.

i. **MUP_IHP_RY21_P02_V10_DY18_PrvSvc.csv**
ii. **MUP_IHP_RY21_P02_V10_DY19_PrvSvc.csv**
iii. **MUP_IHP_RY22_P02_V10_DY20_PrvSvc.csv**

(“Medicare Inpatient Hospitals - by Provider and Service” 2022)

iv. **MUP_IHP_RY21_P02_V10_Dy18_Geo.csv**
v. **MUP_IHP_RY21_P02_V10_Dy19_Geo.csv**
vi. **MUP_IHP_RY22_P02_V10_Dy20_Geo.csv**

(“Medicare Inpatient Hospitals - by Geography and Service” 2022)

- ***Description:*** The Datasets 1,2 and 3 contain data about the use, hospital charges, and payments for more than 3000 US hospitals that received IPPS payments (Inpatient Prospective Payment System). Each dataset is for the years 2020, 2019, and 2018 respectively. The Datasets 4,5, and 6 are like the ones above except the data is summarized at state and nation for each category of the medical procedure. The 3 datasets are for the years 2020, 2019, and 2018 respectively.

- ***Description of the fields:***

  – ***Rndrng_Prvdr_CCN*** : This describes the Rendering provider’s CMS certification number, which is a unique Identifier to identify a hospital.

  - ***Rndrng_Prvdr_Org_Name*** : Denotes the Rendering Provider organization name which is the name of the hospital.
  - ***Rndrng_Prvdr_St, Rndrng_Prvdr_City, Rndrng_Prvdr_State_Abrvtn, and Rndrng_Prvdr_Zip5*** : These fields indicate the street address, City name, State name, and the Zip codes of the hospital.
  - ***DRG_Cd***: Classification code for the Diagnosis Related Group that indicates unique code for each type of medical procedure.
  - ***DRG_Desc*** : Description of the Diagnosis Related Group code. This indicates the description of the medical procedure.
  - ***Tot_Dschrgs*** : This indicates the total number of all discharges billed by all the providers.
  - ***Avg_Tot_Pymt_Amt*** : This indicates the average total payment made to all providers which include the copays, deductibles, etc.
  - ***Rndrng_Prvdr_Geo_Lvl*** : This indicates if the data is summarized at the National level or State level.
  - ***Rndrng_Prvdr_Geo_Desc*** : This indicates the name of the state.

- **Observations about the data**

  - The data in these datasets do not have any empty or missing values.
  - However, to distinguish the data in the datset from each other a new column for Year should be added.
  - The Filed CCN has some values that are of length 5 and some are of length 6. We may need to standardize the length by adding a leading 0 to join the datasets that are used in this project.
  - Some of the datasets are huge with up to 200,000 rows and have up to 15 columns each, though only the columns described above will be utilized in this project.

- ***Period when data was collected:*** Dec, 2022

vii. **NASHP_HCT_Data_2022_November.csv**

(“Hospital Provider Cost Report” 2022)

- ***Description:*** This contains the Healthcare Coset report details collected by the NASHP team (National Academy for State Health Policy) in collaboration with rice university. It contains the Healthcare cost report details such as hospital details, Bed size, Occupancy Rates, and lots of financials such as net revenue, net income, etc. It contains data for approximately 5100 hospitals across the US for the period between 2010 to 2021.

- ***Description of the fields:***
  - ***CCN#*** : This indicates the Unique identifier for each hospital (CMS Certification Number). This denotes a Medicare Provider Number.
  - ***Facility Type*** : This indicates if the hospital is a Critical Access Hospital or a Short-Term Facility.
  - ***Year*** : Indicates the reporting year.
  - ***Hospital Name, Hospital Abbreviated Name, Address, Zip Code, City, State*** : Indicates the name and address of the hospital facility.
  - ***Hospital Ownership Type*** : This indicates the classification of the hospital if it is "For-Profit" or "Non-Profit" or "Governmental".
  - ***Bed Size*** : Indicates the number of beds available for Inpatient care and does not include the beds for emergency rooms or Outpatient care.
  - ***Inpatient Occupancy*** : Percentage of beds occupied by inpatient care during the given year.
  - ***Net Income (Loss) and Net Profit Margin*** : Indicates the financial metrics that will be used in this project.
- **Observations about the data** :
  - The dataset has 85 fields and 51894 rows of data, though in this project about 12 fields will be used.
  - This dataset requires quite a bit of cleansing. The field "Hospital Name" and "Hospital Abbreviated name" has invalid numeric values that should be excluded.
  - The Address field and City have invalid values such as ".", "X"," Y", etc. should be excluded.
  - The field's Bed Size, Occupancy rates, Net Revenue, and Net Profit margin all have invalid values such as "-99999" and "." Should be excluded or defaulted to 0.
- ***Period when data was collected:*** 2022

6 Required Packages

- ***ggplot2:*** This is one of the most important packages used for data visualization and is used to create Aesthetic visualization plots.

- ***lubridate:*** The dataset contains date fields and hence lubridate is used to extract individual date components if needed.
- ***dplyr and plyr:*** This is used for various type of data manipulation.
- ***stringr:*** This is used for String Manipulations.
- ***tidyr:*** This is used to tidy up the data.
- ***purrr:*** This is used to apply the function to a list and also to use the map functions.
- ***reshape2:*** This package is used for Data munging to convert the rows to columns or for data Melting operations.
- ***Magrittr :*** This package will be used to utilize the pipe functionality
- ***DT:*** This is for an interactive tabular experience through DataTables JavaScript library
- ***rmarkdown and knitr:*** These are required to knit the documents as pdf using RMarkdown.

7 Plots and Table needs

- ***Plots***
  - *Histograms*: This is to show the distribution of values for a variable.
  - *Scatterplots*: This is to show two variables in comparison with each other
  - *Boxplots*: This is to identify the outliers in the data.
  - *Density plots*: This is to show the density distribution
  - *Facet_wrap*: This takes data in one variable and splits data in it and creates separate panes for each set to fit them in the plot.
  - *Facet_grid*: This is like facet wrap but it applies all levels of a variable to a row and column and fits them in a plot.
  - *Violin Plot* : These are similar to the box plots but can explain the density of the data.
  - *line plot*: Line plot to show the relation between two variables.
  - *qq-plot* : These can be used to explain how close is the data to Normal distribution.
  - *Heat Map* : Heat map can be used to explain the density in the pattern of data based on the Heat color schemes.

- ***Tables:***
    - *kableExtra* : Tabular data can be printed using kableExtra
    - *Datatable*: From DT package

8 How to Import and Clean my data:

Importing data:

This project has seven datasets of which the first six datasets have similar structure and all these datasets are in CSV format. Hence Importing the data and cleansing will be similar for the first 6 datasets. The seventh dataset is in Excel format.

To import the CSV datasets, ***read.csv()*** in R will be used and for importing Excel datasets, ***read_excel()*** function from the ***readxl*** package will be used.

Transforming Datatypes:

- After importing the data, the structure of the data is analyzed to see if the fields are of appropriate data types. Some of the fields that have fewer unique values can be converted to factors.
- Some of the fields such as ***total amount, number of beds, number of hospitalizations, etc.*** can be numeric fields. If they are not numeric, datatype conversion should be done.
- Fields such as ***DRG_Cd*** which represents the code for medical procedures need not be numeric instead, they should be a Character field.

Renaming Columns and limiting the number of columns:

- The columns should be renamed to shorter and easy-to-understand names. Some of the fields have spaces in them, which are replaced with an underscore so that they are easy to use.
- Some of the datasets have about 85 columns, and as we may not require all of them, only a few columns will be loaded into our final data frame by using the select function.
- The column names in the final data frame can be validated by using ***names()*** function.

Cleansing the data:

- Some of the fields like Hospital names have invalid characters such as Double quotes, which should be removed.

- The Medical procedures are identified with the DRG_Cd (Unique code for a medical procedure), however, while combing the datasets it is important to make sure the Description of the procedures are the same across the datasets. For example, the below descriptions of the same procedure code will yield different results while plotting the graphs. The names should be cleansed after combining the datasets.

TRACH ***W*** *MV >96 HRS OR PDX EXC FACE, MOUTH & NECK* ***W/O*** *MAJ O.R.*

TRACH ***WITH*** *MV >96 HRS OR PDX EXC FACE, MOUTH & NECK* ***WITHOUT*** *MAJ O.R.*

- The Hospital code, CCN, which uniquely identifies a medical facility should have the same length across all datasets. As these are read from CSV and excel files, the codes may be interpreted as Numbers. They should be converted to Characters and 0's should be padded in the front if the length is not the same for all values. While joining data frames, this can come in handy.
- Some values in Hospital names and city fields have only numbers that are invalid values, and they should be excluded. Some of the address fields have invalid values such as only numbers and one or two characters. Though these are invalid values, they will not be excluded because the address is not a critical field in this analysis and the invalid values can be ignored.
- Also, some character fields have invalid values such as period (".") in them and can be excluded. This should be done cautiously. For instance, "ST. VINCENTS EAST" is valid, while "9.2" is an invalid hospital name. Hence the length of the field along with the presence of periods should be analyzed before excluding them.
- Some of the important fields such as Bed size, and city names have "NA" and they can be excluded.
- Some numeric fields have invalid values such as -99999 for Bed size and they can be excluded. As the dataset have many thousands of records, excluding a few invalid values will be reasonable

Importing and Cleansing commands in R

```r
# Calling the Libraries used
library(readr)
library(magrittr)
library(dplyr)
library(ggplot2)
library(ggthemes)
library(lubridate)
library(dplyr)
library(tidyr)
library(DT)
library(scales)
library(stringr)
library(ggalt)
library(treemap)
library(readxl)
library(knitr)
library(lemon)
library(kableExtra)
```

```
## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : ’length(x) = 2 > 1’ in coercion to ’logical(1)’
```

```
library(reshape2)
```

```
# Exporting Dataset #1
# MUP_IHP_RY22_P02_V10_DY20_PrvSvc.csv
medic_prvsvc_2020_df_orig <- read.csv("MUP_IHP_RY22_P02_V10_DY20_PrvSvc.csv")
# Exporting the Dataset #2-
# MUP_IHP_RY21_P02_V10_DY19_PrvSvc.csv
medic_prvsvc_2019_df_orig <- read.csv("MUP_IHP_RY21_P02_V10_DY19_PrvSvc.csv")
# Exporting the Dataset #3-
# MUP_IHP_RY21_P02_V10_DY18_PrvSvc.csv
medic_prvsvc_2018_df_orig <- read.csv("MUP_IHP_RY21_P02_V10_DY18_PrvSvc.csv")
# Validating the structure of one of the imported dataset
str(medic_prvsvc_2020_df_orig)
```

```
## ’data.frame’:    165281 obs. of  15 variables:
##  $ Rndrng_Prvdr_CCN          : int  10001 10001 10001 10001 10001 10001 10001 10001 10001 10001 ...
##  $ Rndrng_Prvdr_Org_Name     : chr  "Southeast Health Medical Center" "Southeast Health Medical Cente
##  $ Rndrng_Prvdr_St           : chr  "1108 Ross Clark Circle" "1108 Ross Clark Circle" "1108 Ross Clar
##  $ Rndrng_Prvdr_City        : chr  "Dothan" "Dothan" "Dothan" "Dothan" ...
##  $ Rndrng_Prvdr_State_Abrvtn: chr  "AL" "AL" "AL" "AL" ...
##  $ Rndrng_Prvdr_State_FIPS  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Rndrng_Prvdr_Zip5        : int  36301 36301 36301 36301 36301 36301 36301 36301 36301 36301 ...
##  $ Rndrng_Prvdr_RUCA        : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Rndrng_Prvdr_RUCA_Desc   : chr  "Metropolitan area core: primary flow within an urbanized area of
##  $ DRG_Cd                   : int  23 24 25 27 38 39 56 57 64 65 ...
##  $ DRG_Desc                 : chr  "CRANIOTOMY W MAJOR DEVICE IMPLANT OR ACUTE COMPLEX CNS PDX W MCC
##  $ Tot_Dschrgs              : int  51 16 30 11 25 50 14 23 101 99 ...
##  $ Avg_Submtd_Cvrd_Chrg     : num  166626 105888 182686 117293 112529 ...
##  $ Avg_Tot_Pymt_Amt         : num  37033 26209 29492 15641 11399 ...
##  $ Avg_Mdcr_Pymt_Amt        : num  33908 22531 27776 14357 9559 ...
```

```r
# Creating a new dataframe with a limited set of fields
medic_prvsvc_2020_df <- medic_prvsvc_2020_df_orig %>%
    select(CCN = Rndrng_Prvdr_CCN, hosp_name = Rndrng_Prvdr_Org_Name,
        street = Rndrng_Prvdr_St, city = Rndrng_Prvdr_City,
        state = Rndrng_Prvdr_State_Abrvtn, zip = Rndrng_Prvdr_Zip5,
        DRG_Cd, DRG_Desc, Tot_Dschrgs, Avg_Tot_Pymt_Amt)

medic_prvsvc_2019_df <- medic_prvsvc_2019_df_orig %>%
    select(CCN = Rndrng_Prvdr_CCN, hosp_name = Rndrng_Prvdr_Org_Name,
        street = Rndrng_Prvdr_St, city = Rndrng_Prvdr_City,
        state = Rndrng_Prvdr_State_Abrvtn, zip = Rndrng_Prvdr_Zip5,
        DRG_Cd, DRG_Desc, Tot_Dschrgs, Avg_Tot_Pymt_Amt)

medic_prvsvc_2018_df <- medic_prvsvc_2018_df_orig %>%
    select(CCN = Rndrng_Prvdr_CCN, hosp_name = Rndrng_Prvdr_Org_Name,
        street = Rndrng_Prvdr_St, city = Rndrng_Prvdr_City,
        state = Rndrng_Prvdr_State_Abrvtn, zip = Rndrng_Prvdr_Zip5,
        DRG_Cd, DRG_Desc, Tot_Dschrgs, Avg_Tot_Pymt_Amt)

# Adding a new column for the Year to each Dataframe
```

```
medic_prvsvc_2020_df$year <- factor(2020)
medic_prvsvc_2019_df$year <- factor(2019)
medic_prvsvc_2018_df$year <- factor(2018)

# Cleansing the dataset by converting a few columns as
# Factors
medic_prvsvc_2020_df$DRG_Cd <- factor(medic_prvsvc_2020_df$DRG_Cd)
medic_prvsvc_2019_df$DRG_Cd <- factor(medic_prvsvc_2019_df$DRG_Cd)
medic_prvsvc_2018_df$DRG_Cd <- factor(medic_prvsvc_2018_df$DRG_Cd)

# Cleansing by removing invalid characters
medic_prvsvc_2020_df$DRG_Desc <- str_replace(medic_prvsvc_2020_df$DRG_Desc,
    "\"", "")
medic_prvsvc_2019_df$DRG_Desc <- str_replace(medic_prvsvc_2019_df$DRG_Desc,
    "\"", "")
medic_prvsvc_2018_df$DRG_Desc <- str_replace(medic_prvsvc_2018_df$DRG_Desc,
    "\"", "")

# Padding a 0 to standardize the length of CCN field
medic_prvsvc_2020_df$CCN <- str_pad(medic_prvsvc_2020_df$CCN,
    6, pad = "0")
medic_prvsvc_2019_df$CCN <- str_pad(medic_prvsvc_2019_df$CCN,
    6, pad = "0")
medic_prvsvc_2018_df$CCN <- str_pad(medic_prvsvc_2018_df$CCN,
    6, pad = "0")
```

```
# Combining the First 3 datasets into 1 Dataset
medic_prvsvc_combined_df <- rbind(medic_prvsvc_2018_df, medic_prvsvc_2019_df,
    medic_prvsvc_2020_df)

# Validating the counts of individual dataset and the
# combined one
nrow(medic_prvsvc_2018_df)
```

```
## [1] 193003
```

```r
nrow(medic_prvsvc_2019_df)
```

```
## [1] 188806
```

```r
nrow(medic_prvsvc_2020_df)
```

```
## [1] 165281
```

```r
nrow(medic_prvsvc_combined_df)
```

```
## [1] 547090
```

```r
identical(nrow(medic_prvsvc_combined_df), nrow(medic_prvsvc_2018_df) +
    nrow(medic_prvsvc_2019_df) + nrow(medic_prvsvc_2020_df))
```

```
## [1] TRUE
```

```r
# Cleansing the combined dataframe by Standardizing the
# DRG_Desc Values by eliminating the duplicates
unique_drg_cd_prvsvc <- medic_prvsvc_combined_df %>%
    count(DRG_Cd, DRG_Desc) %>%
    select(DRG_Cd, DRG_Desc) %>%
    distinct(DRG_Cd, .keep_all = TRUE)
names(unique_drg_cd_prvsvc) <- c("unique_DRG_Cd", "Unique_DRG_Desc")
# Using Merge function to standardize the Values
medic_prvsvc_combined_df <- merge(x = medic_prvsvc_combined_df,
    y = unique_drg_cd_prvsvc, by.x = "DRG_Cd", by.y = "unique_DRG_Cd") %>%
    select(CCN, hosp_name, street, city, state, zip, DRG_Cd,
        DRG_Desc = Unique_DRG_Desc, Tot_Dschrgs, Avg_Tot_Pymt_Amt,
        year)
```

```r
# Importing 4th dataset MUP_IHP_RY22_P02_V10_Dy20_Geo.csv
medic_geo_2020_df_orig <- read.csv("MUP_IHP_RY22_P02_V10_Dy20_Geo.csv")
# Importing 5th Dataset MUP_IHP_RY21_P02_V10_Dy19_Geo.csv
medic_geo_2019_df_orig <- read.csv("MUP_IHP_RY21_P02_V10_Dy19_Geo.csv")
# Importing 6th Dataset MUP_IHP_RY21_P02_V10_Dy18_Geo.csv
medic_geo_2018_df_orig <- read.csv("MUP_IHP_RY21_P02_V10_Dy18_Geo.csv")

# Creating a new dataframe with a limited set of fields
medic_geo_2020_df <- medic_geo_2020_df_orig %>%
    select(Geo_lvl = Rndrng_Prvdr_Geo_Lvl, Geo_cd = Rndrng_Prvdr_Geo_Cd,
        Geo_desc = Rndrng_Prvdr_Geo_Desc, DRG_Cd, DRG_Desc,
        Tot_Dschrgs, Avg_Tot_Pymt_Amt)
medic_geo_2019_df <- medic_geo_2019_df_orig %>%
    select(Geo_lvl = Rndrng_Prvdr_Geo_Lvl, Geo_cd = Rndrng_Prvdr_Geo_Cd,
        Geo_desc = Rndrng_Prvdr_Geo_Desc, DRG_Cd, DRG_Desc,
        Tot_Dschrgs, Avg_Tot_Pymt_Amt)
medic_geo_2018_df <- medic_geo_2019_df_orig %>%
    select(Geo_lvl = Rndrng_Prvdr_Geo_Lvl, Geo_cd = Rndrng_Prvdr_Geo_Cd,
        Geo_desc = Rndrng_Prvdr_Geo_Desc, DRG_Cd, DRG_Desc,
        Tot_Dschrgs, Avg_Tot_Pymt_Amt)

# Adding a new column for the Year to each Dataframe
medic_geo_2020_df$year <- factor(2020)
medic_geo_2019_df$year <- factor(2019)
medic_geo_2018_df$year <- factor(2018)

# Cleansing the dataset by converting a few columns as
# Factors
medic_geo_2020_df$DRG_Cd <- factor(medic_geo_2020_df$DRG_Cd)
medic_geo_2019_df$DRG_Cd <- factor(medic_geo_2019_df$DRG_Cd)
medic_geo_2018_df$DRG_Cd <- factor(medic_geo_2018_df$DRG_Cd)
```

```
# Cleansing by removing invalid characters
medic_geo_2020_df$DRG_Desc <- str_replace(medic_geo_2020_df$DRG_Desc,
    "\"", "")
medic_geo_2019_df$DRG_Desc <- str_replace(medic_geo_2019_df$DRG_Desc,
    "\"", "")
medic_geo_2018_df$DRG_Desc <- str_replace(medic_geo_2018_df$DRG_Desc,
    "\"", "")
```

```
# Combining the Datasets 4,5 and 6 into one DataFrame
medic_geo_combined_df <- rbind(medic_geo_2018_df, medic_geo_2019_df,
    medic_geo_2020_df)
nrow(medic_geo_combined_df)
```

```
## [1] 82831
```

```
# Validating the counts of the combined Dataframe
identical(nrow(medic_geo_combined_df), nrow(medic_geo_2018_df) +
    nrow(medic_geo_2019_df) + nrow(medic_geo_2020_df))
```

```
## [1] TRUE
```

```
# Cleansing the combined dataframe by Standardizing the
# DRG_Desc Values by eliminating the duplicates
unique_drg_cd_geo <- medic_geo_combined_df %>%
    count(DRG_Cd, DRG_Desc) %>%
    select(DRG_Cd, DRG_Desc) %>%
    distinct(DRG_Cd, .keep_all = TRUE)
names(unique_drg_cd_geo) <- c("unique_DRG_Cd", "Unique_DRG_Desc")
# Using Merge function to standardize the Values
medic_geo_combined_df <- merge(x = medic_geo_combined_df, y = unique_drg_cd_geo,
    by.x = "DRG_Cd", by.y = "unique_DRG_Cd") %>%
    select(DRG_Cd, DRG_Desc = Unique_DRG_Desc, Geo_lvl, Geo_cd,
        Geo_desc, Tot_Dschrgs, Avg_Tot_Pymt_Amt, year)
```

```r
# Importing the 7th dataset
# NASHP_HCT_Data_2022_November.xlsx
NASHP_excel <- "NASHP_HCT_Data_2022_November.xlsx"
NASHP_orig_df <- suppressWarnings(read_excel(NASHP_excel, sheet = 2))
# Creating a new dataframe with limited columns and also
# renaming the fields
NASHP_df <- NASHP_orig_df %>%
    select(CCN = `CCN#`, facility_type = `Facility Type`, year = Year,
        hosp_name = `Hospital Name`, hosp_abbr_name = `Hospital Abbreviated Name`,
        address = Address, zip = `Zip Code`, city = City, state = State,
        health_system = `Health System`, hosp_ownership = `Hospital Ownership Type`,
        bed_size = `Bed Size`, inpatient_occupancy = `Inpatient Occupancy`,
        net_income = `Net Income (Loss)`, net_profit_margin = `Net Profit Margin`)
# Examining the structure of the new dataframe
str(NASHP_df)
```

```
## tibble [51,894 x 15] (S3: tbl_df/tbl/data.frame)
##  $ CCN                : chr [1:51894] "010001" "010001" "010001" "010001" ...
##  $ facility_type      : chr [1:51894] "Short Term (General and Specialty) Hospitals" "Short Term (Ge
##  $ year               : num [1:51894] 2011 2012 2013 2014 2015 ...
##  $ hosp_name          : chr [1:51894] "SOUTHEAST ALABAMA MEDICAL CENTER" "SOUTHEAST ALABAMA MEDICAL
##  $ hosp_abbr_name     : chr [1:51894] "SE AL MC" "SE AL MC" "SE AL MC" "SE AL MC" ...
##  $ address            : chr [1:51894] "1108 ROSS CLARK CIRCLE" "1108 ROSS CLARK CIRCLE" "1108 ROSS C
##  $ zip                : chr [1:51894] "36301" "36301" "36301" "36301" ...
##  $ city               : chr [1:51894] "DOTHAN" "DOTHAN" "DOTHAN" "DOTHAN" ...
##  $ state              : chr [1:51894] "AL" "AL" "AL" "AL" ...
##  $ health_system      : chr [1:51894] "Southeast Health AL" "Southeast Health AL" "Southeast Health
##  $ hosp_ownership     : chr [1:51894] "Governmental" "Governmental" "Governmental" "Governmental" ..
##  $ bed_size           : num [1:51894] 408 408 408 398 400 387 387 327 342 332 ...
##  $ inpatient_occupancy: num [1:51894] 0.607 0.578 0.633 0.683 0.614 ...
##  $ net_income         : chr [1:51894] "4089394" "6672718" "12739814" "1125576" ...
##  $ net_profit_margin  : chr [1:51894] "1.6318108370371089E-2" "2.6184635846616799E-2" "4.86257836126
```

```r
# Cleansing the dataset by converting a few columns as
# Factors and some as Numeric
NASHP_df$CCN <- as.factor(NASHP_df$CCN)
NASHP_df$facility_type <- as.factor(NASHP_df$facility_type)
NASHP_df$year <- as.factor(NASHP_df$year)
NASHP_df$state <- as.factor(NASHP_df$state)
NASHP_df$health_system <- as.factor(NASHP_df$health_system)
NASHP_df$hosp_ownership <- as.factor(NASHP_df$hosp_ownership)
NASHP_df$net_income <- as.numeric(NASHP_df$net_income)
```

```
## Warning: NAs introduced by coercion
```

```r
NASHP_df$net_profit_margin <- as.numeric(NASHP_df$net_profit_margin)
```

```
## Warning: NAs introduced by coercion
```

```r
# Creating a reusable function to check for Number only
# values
numbers_only <- function(var1) {
    !grepl("\\D", var1)
}

# Checking the Number of rows that have only Numbers in
# the Hospital name field and filtering them as the count
# is low.
nrow(NASHP_df[numbers_only(NASHP_df$hosp_name), ])
```

```
## [1] 5
```

```r
NASHP_df <- NASHP_df[!numbers_only(NASHP_df$hosp_name), ]
# Validating if the cleanup is done correctly
NASHP_df[numbers_only(NASHP_df$hosp_name), ]
```

```
## # A tibble: 0 x 15
## # ... with 15 variables: CCN <fct>, facility_type <fct>, year <fct>,
## #   hosp_name <chr>, hosp_abbr_name <chr>, address <chr>, zip <chr>,
## #   city <chr>, state <fct>, health_system <fct>, hosp_ownership <fct>,
## #   bed_size <dbl>, inpatient_occupancy <dbl>, net_income <dbl>,
## #   net_profit_margin <dbl>
```

```
# Checking other fields such as City and State to see if
# they have 'only Number' values and there is none.
sum(numbers_only(NASHP_df$city))
```

```
## [1] 0
```

```
sum(numbers_only(NASHP_df$state))
```

```
## [1] 0
```

```
# Validating To find if any city has name that is lesser
# than or 2 characters. Results none.
NASHP_df[nchar(NASHP_df$city) <= 2, ]
```

```
## # A tibble: 0 x 15
## # ... with 15 variables: CCN <fct>, facility_type <fct>, year <fct>,
## #   hosp_name <chr>, hosp_abbr_name <chr>, address <chr>, zip <chr>,
## #   city <chr>, state <fct>, health_system <fct>, hosp_ownership <fct>,
## #   bed_size <dbl>, inpatient_occupancy <dbl>, net_income <dbl>,
## #   net_profit_margin <dbl>
```

```
# Creating a function that checks if a Period is present
contains_period <- function(var1) {
    grepl("\\.", var1)
}
# Checking if there are any periods in the city name and
# if the length is lesser than or equal to 5. Results 1
# row
```

```r
NASHP_df[contains_period(NASHP_df$city) & nchar(NASHP_df$city) <=
    5, ]
```

```
## # A tibble: 1 x 15
##   CCN    facil~1 year  hosp_~2 hosp_~3 address zip   city  state healt~4 hosp_~5
##   <fct>  <fct>   <fct> <chr>   <chr>   <chr>   <chr> <chr> <fct> <fct>   <fct>
## 1 280030 Short ~ 2011  CREIGH~ CREIGH~ 30      68131 9.01  NE    Indepe~ For-Pr~
## # ... with 4 more variables: bed_size <dbl>, inpatient_occupancy <dbl>,
## #   net_income <dbl>, net_profit_margin <dbl>, and abbreviated variable names
## #   1: facility_type, 2: hosp_name, 3: hosp_abbr_name, 4: health_system,
## #   5: hosp_ownership
```

```r
# Excluding 1 row from above
NASHP_df <- NASHP_df[!(contains_period(NASHP_df$city) & nchar(NASHP_df$city) <=
    5), ]
# finding invalid values in bed size and excluding them
nrow(NASHP_df[is.na(NASHP_df$bed_size), ])
```

```
## [1] 71
```

```r
NASHP_df <- NASHP_df[!is.na(NASHP_df$bed_size), ]
# Excluding the -99999 values from bed size
NASHP_df <- NASHP_df[-c(which(NASHP_df$bed_size < 0)), ]
# Checking for nulls in other numeric fields
sum(is.na(NASHP_df$net_profit_margin))
```

```
## [1] 1004
```

```r
sum(is.na(NASHP_df$net_income))
```

```
## [1] 1005
```

```
NASHP_df <- NASHP_df[!is.na(NASHP_df$net_profit_margin), ]
NASHP_df <- NASHP_df[!is.na(NASHP_df$net_income), ]
# Converting the net_profit_margin into percentage
NASHP_df$net_profit_margin <- round(NASHP_df$net_profit_margin *
    100, digits = 2)
# Finding invalid values in the Net Income and
# eliminating them
sum(NASHP_df$net_income == ".")
```

```
## [1] 0
```

```
NASHP_df <- NASHP_df[!NASHP_df$net_income == ".", ]
# Displaying Final row count in the data frame after all
# Cleanups
nrow(NASHP_df)
```

```
## [1] 50796
```

9 What does the final data set look like?

1. The datasets medic_prvsvc_2018_df, medic_prvsvc_2019_df, and medic_prvsvc_2020_df will be combined to form a final combined dataset for Provider Services- ***medic_prvsvc_combined_df***

```
kbl(head(medic_prvsvc_combined_df[1:6, c(1:6)]), caption = "Medical Provider Services DataFrame",
    booktabs = T) %>%
    kable_styling(latex_options = c("striped", "hold_position"))
```

2. The datasets medic_geo_2018_df, medic_geo_2019_df, and medic_geo_2020_df will be combined to form the final combined dataset- ***medic_geo_combined_df***

```
kbl(head(medic_geo_combined_df[1:6, c(1, 3:8)]), caption = "Medical Geographics DataFrame",
    booktabs = T) %>%
    kable_styling(latex_options = c("striped", "hold_position"))
```

Table 1: Medical Provider Services DataFrame

CCN	hosp_name	street	city	state	zip
220116	Tufts Medical Center	800 Washington Street	Boston	MA	2111
220110	Brigham And Women's Hospital	75 Francis Street	Boston	MA	2115
520098	University Of Wi Hospitals & Clinics Authority	600 Highland Avenue	Madison	WI	53792
060024	University Of Colorado Hospital Authority	12605 E 16th Ave	Aurora	CO	80045
100128	Tampa General Hospital	1 Tampa General Cir	Tampa	FL	33606
490009	University Of Virginia Medical Center	1215 Lee Street	Charlottesville	VA	22908

Table 2: Medical Geographics DataFrame

DRG_Cd	Geo_lvl	Geo_cd	Geo_desc	Tot_Dschrgs	Avg_Tot_Pymt_Amt	year
1	National	NA	National	2228	276916.9	2018
1	State	20	Kansas	21	186416.0	2020
1	State	36	New York	177	318296.2	2018
1	State	55	Wisconsin	32	284826.5	2018
1	State	17	Illinois	93	303856.4	2020
1	State	1	Alabama	12	208240.9	2019

3. The 7th dataset after cleansing will look like the below: ***NASHP_df***

```r
kbl(head(NASHP_df[1:6, c(1, 3, 5:8)]), caption = "NASHP DataFrame",
    booktabs = T) %>%
    kable_styling(latex_options = c("striped", "hold_position"))
```

Table 3: NASHP DataFrame

CCN	year	hosp_abbr_name	address	zip	city
010001	2011	SE AL MC	1108 ROSS CLARK CIRCLE	36301	DOTHAN
010001	2012	SE AL MC	1108 ROSS CLARK CIRCLE	36301	DOTHAN
010001	2013	SE AL MC	1108 ROSS CLARK CIRCLE	36301	DOTHAN
010001	2014	SE AL MC	1108 ROSS CLARK CIRCLE	36301	DOTHAN
010001	2015	SE AL MC	1108 ROSS CLARK CIRCLE	36301	DOTHAN
010001	2016	SE AL MC	1108 ROSS CLARK CIRCLE	36301	DOTHAN

10 What information is not self-evident?

- Each individual dataset (after cleansing) for Medical Provider services- ***medic_prvsvc_combined_df*** contains data such as Hospital details, Details of the medical procedures, and the Average payment amount for Medicare for each service each year. However, what is not self-evident is the combined metrics of the cost of a medical procedure, such as the average cost of a procedure for a city or state, or zip code.

- Also, each dataset contains the details of medical procedures only for a year, combining the data for each year helps us study the trends in the costs across a geographic location, which is not self-evident.
- All the above points also apply to the datasets ***medic_geo_combined_df***. This dataset contains data at the state and National level but not at the city or Zip code. Hence, combining the data for each year helps us study the trends at the state/ National level for changes in the cost of medical procedures over the years, as it is not self-evident.
- The dataset ***NASHP_df*** contains the details of the hospital such as the address, type of hospital, Ownership type, and a lot of financial information such as Net Revenue, Profit margin, etc. However, it only presents the raw data if the hospital was profitable or not. Combining the data from the other Medicare datasets and studying the patterns in the payments made for Medicare helps us understand why the hospitals were profitable for a certain period. This information is not self-evident unless the datasets are combined, and the underlying patterns are studied.
- In some cities that share borders between states, the medical costs can vary significantly across states. For instance, for someone that lives in ***Boise City, OK*** (which is in Oklahoma's panhandle), the states of ***Texas, Colorado, New Mexico, and Kansas*** are all within 30 minutes of driving distance. This research can help a patient in Boise City compare the costs across the hospitals and choose the appropriate one. This information is not self-evident from the raw data.

11 What are different ways you could look at this data?

This study contains two broader types of research as described below:

1. The first category is to study the data at ***the national level and compare them with the state averages***. This includes analysis such as comparing the cost of the National average for a medical procedure versus the state averages, Identifying the top 10 medical procedures that had the most hospitalizations across the nation, etc.

2. The second type of study is focused on a relatively ***smaller geographic area such as a city or State or Zip code***. For instance, identifying the top 10 profitable hospitals in a state, Identifying the cities that offer treatment for medical procedures in the state. This also includes comparing the costs of medical procedures in a city that shares borders across states.

All the cities and states included in this study are from states that are not frequently discussed or researched, such as Mid-western states, and sparsely populated states like Wyoming, and Alaska. Popular states such as California, Florida, Texas, New York, etc. are deliberately not discussed in this study for this purpose.

12 How do you plan to slice and dice the data?

- As indicated above, the data will be sliced based on the state or City. For some comparison, data will be filtered based on the Zip-code.
- For the study that focuses on the cost of a medical Procedure, the data will be sliced based on the Procedure code (DRG_Cd) or the description (DRG_Desc).
- For the study that involves the medical costs for a given period, filters will be applied to the Year column and required years will be selected.
- In the NASHP_df dataset, apart from the state and city filters, the data will be sliced based on the Hospital ownership type(hosp_ownership). This dataset also has other fields such as health systems and data will be sliced on this field as well.
- The NASHP_df dataset has data for the years 2011 to 2021, though most of the study is focused on the years 2018-2020, hence filter of the year field will be applied to these years.

13 How could you summarize your data to answer key questions?

1. The data can be grouped based on the ***Medical procedure code and/or description*** and then ***summarized based on Average payments, total payments, and Total and Average hospitalizations***. This is shown below:

```r
# creating re-usable function for display purposes
topN <- function(df_name, var, N = 10) {
    df_name %>%
        arrange(desc(var)) %>%
        head(N)
}


# Summarizing based on Medical Procedures done and using
# the function topN to display 10 rows
medic_prvsvc_summary_df1 <- medic_prvsvc_2020_df %>%
    group_by(DRG_Cd, DRG_Desc) %>%
    dplyr::summarise(total_amt_paid = sum(Avg_Tot_Pymt_Amt),
        mean_amt_paid = mean(Avg_Tot_Pymt_Amt), total_discharges = sum(Tot_Dschrgs)) %>%
    topN(., "total_amt_paid", 10)
kbl(head(medic_prvsvc_summary_df1), caption = "Results of Summary Dataframe 1",
    booktabs = T) %>%
    kable_styling(full_width = F, latex_options = c("striped",
        "hold_position")) %>%
    column_spec(2, width = "20em")
```

Table 4: Results of Summary Dataframe 1

DRG_Cd	DRG_Desc	total_amt_paid	mean_amt_paid	total_discharges
1	HEART TRANSPLANT OR IMPLANT OF HEART ASSIST SYSTEM WITH MCC	20535801.3	297620.31	1343
3	ECMO OR TRACHEOSTOMY WITH MV >96 HOURS OR PRINCIPAL DIAGNOSIS EXCEP	57683736.0	171677.79	9186
4	TRACH W MV >96 HRS OR PDX EXC FACE, MOUTH & NECK W/O MAJ O.R.	38420005.1	103279.58	7219
5	LIVER TRANSPLANT WITH MCC OR INTESTINAL TRANSPLANT	6918481.0	130537.38	1025
6	LIVER TRANSPLANT W/O MCC	371192.7	46399.09	100
7	LUNG TRANSPLANT	3768959.9	144960.00	498

2. The state can be included in the grouping above and then filters can be applied to a specific state to study the numbers for that state. In this example, Medicare payments in South Dakota are studied based on the grouping done in the states.

```r
# Creating a summary data frame that summarizes the Total
# payments made in each state per medical procedure
prvsvc_2020_summary_df1 <- medic_prvsvc_2020_df %>%
    group_by(DRG_Cd, DRG_Desc, state) %>%
    dplyr::summarise(total_amt_paid = sum(Avg_Tot_Pymt_Amt),
        mean_amt_paid = mean(Avg_Tot_Pymt_Amt), total_discharges = sum(Tot_Dschrgs)) %>%
```

Table 5: Results of Summary Dataframe 2

DRG_Cd	DRG_Desc	state	total_amt_paid	mean_amt_paid	total_discharges
460	SPINAL FUSION EXCEPT CERVICAL WITHOUT MCC	SD	223023.9	27877.99	225
470	MAJOR HIP AND KNEE JOINT REPLACEMENT OR REATTACHMENT OF LOWER EXTREM	SD	187349.3	14411.49	2452
870	SEPTICEMIA OR SEVERE SEPSIS WITH MV >96 HOURS	SD	175364.8	58454.92	60
483	MAJOR JOINT/LIMB REATTACHMENT PROCEDURE OF UPPER EXTREMITIES	SD	172643.0	17264.30	563
454	COMBINED ANTERIOR AND POSTERIOR SPINAL FUSION WITH CC	SD	162167.4	40541.86	76
207	RESPIRATORY SYSTEM DIAGNOSIS WITH VENTILATOR SUPPORT >96 HOURS	SD	154435.6	51478.52	47

```r
    arrange(desc(total_amt_paid))

# Top 8 Medicate Payments in the state of South Dakota

# with the number of Hospitalizations.

SD_2020_df <- prvsvc_2020_summary_df1 %>%

    filter(state == "SD") %>%

    topN(., "total_discharges", 8)

kbl(head(SD_2020_df), caption = "Results of Summary Dataframe 2",

    booktabs = T) %>%

    kable_styling(full_width = F, latex_options = c("striped")) %>%

    column_spec(2, width = "10em")
```

3. In the example below, the intention is to study the top ten hospitalizations in Mobile, Alabama in the year 2020. Hence the filter for the state and city will be applied first, and then data is grouped based on the Procedure code & its description. Then the summary of total Hospitalization is extracted and top 10 results are displayed.

```r
# Top 10 Hospitalizations in Mobile, AL in 2020 and using
# Reusable function topN View(medic_prvsvc_2020_df)
mobileAL_hosp_df_2020 <- medic_prvsvc_2020_df %>%
    filter(state == "AL" & city == "Mobile") %>%
    group_by(DRG_Cd, DRG_Desc) %>%
    dplyr::summarise(Hospitalizations = sum(Tot_Dschrgs)) %>%
    topN(., "Hospitalizations")

kbl(head(mobileAL_hosp_df_2020), caption = "Results of Summary Dataframe 3",
    booktabs = T) %>%
    kable_styling(full_width = F, latex_options = c("striped",
        "hold_position")) %>%
    column_spec(2, width = "20em")
```

Table 6: Results of Summary Dataframe 3

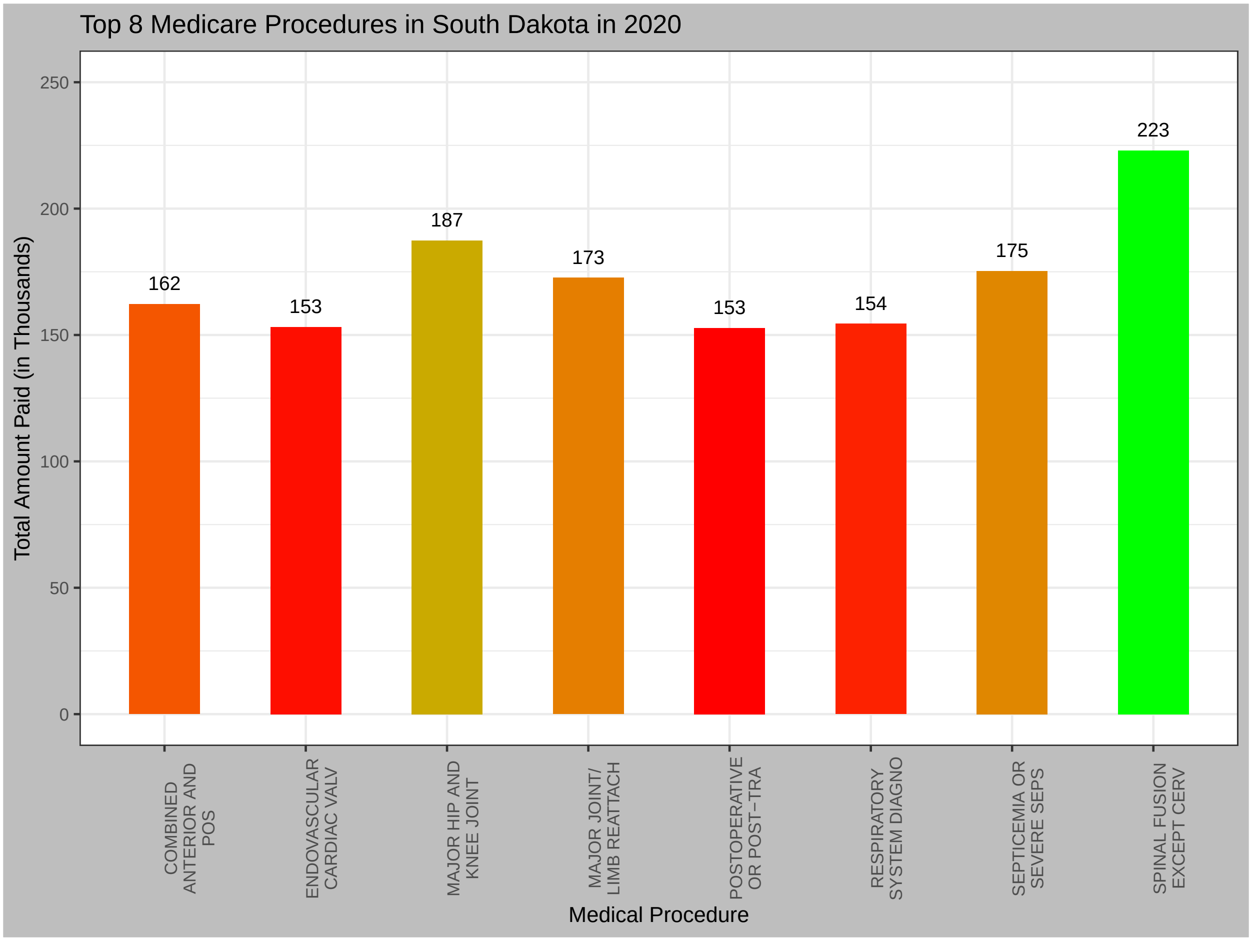
DRG_Cd	DRG_Desc	Hospitalizations
3	ECMO OR TRACHEOSTOMY WITH MV >96 HOURS OR PRINCIPAL DIAGNOSIS EXCEP	20
4	TRACH W MV >96 HRS OR PDX EXC FACE, MOUTH & NECK W/O MAJ O.R.	20
23	CRANIOTOMY W MAJOR DEVICE IMPLANT OR ACUTE COMPLEX CNS PDX W MCC OR	15
24	CRANIOTOMY WITH MAJOR DEVICE IMPLANT OR ACUTE COMPLEX CNS PRINCIPAL	14
25	CRANIOTOMY & ENDOVASCULAR INTRACRANIAL PROCEDURES W MCC	38
36	CAROTID ARTERY STENT PROCEDURES WITHOUT CC/MCC	13

4. Many such summarizations are explained in the next section along with the plots and graphs.

14 What types of plots and tables will help you to illustrate the findings to your questions?

- *Plot 1: Top 8 Medicare Procedures in South Dakota in 2020*

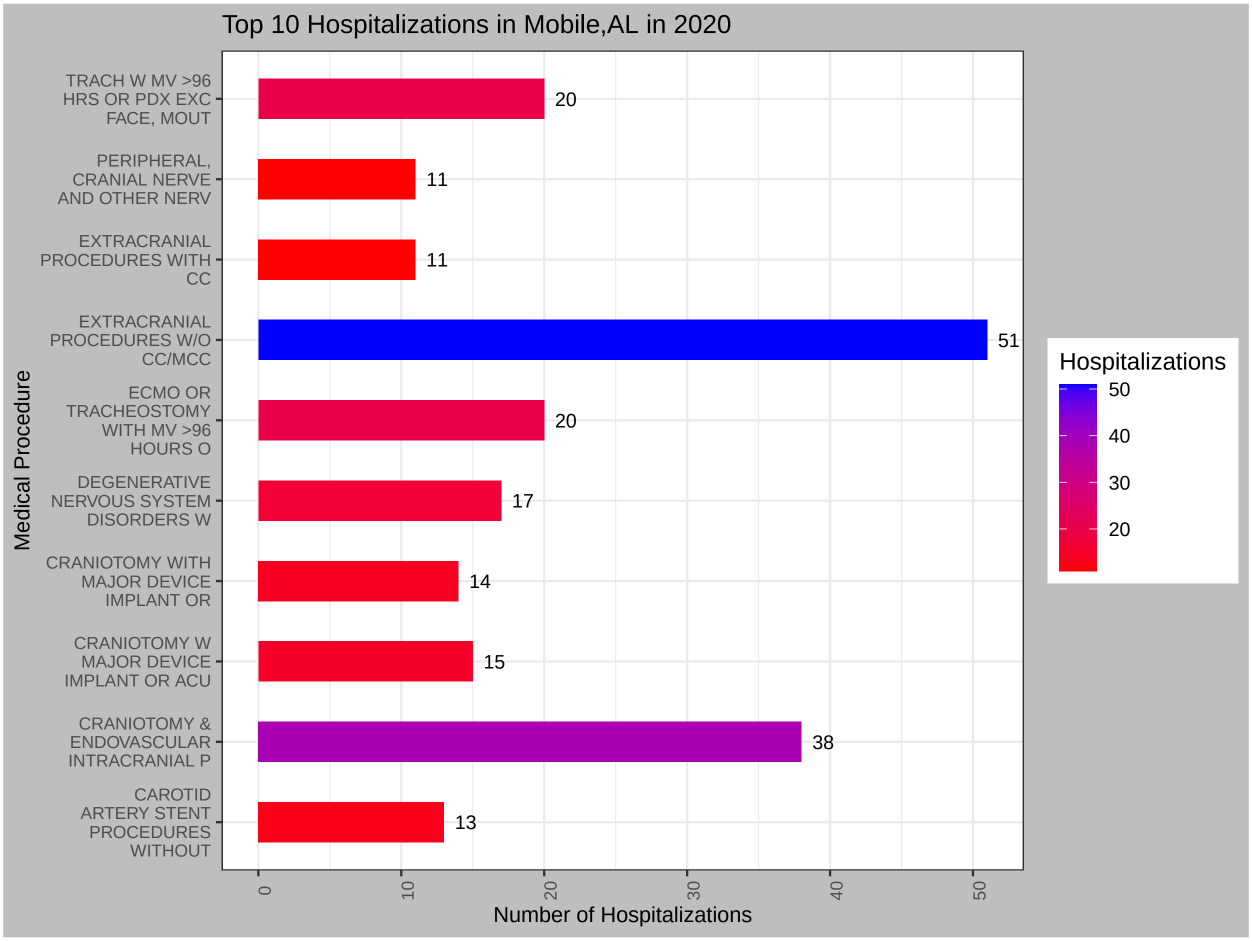
```r
# Top 8 Medicare Procedures in South Dakota in 2020
ggplot(data = SD_2020_df, aes(str_wrap(str_sub(DRG_Desc, 1,
    25), 15), total_amt_paid/1000, fill = total_amt_paid)) +
    geom_bar(stat = "identity", width = 0.5, position = "dodge") +
    geom_text(aes(label = round(total_amt_paid/1000)), vjust = -1,
        size = 3) + scale_fill_gradient(low = "Red", high = "Green") +
    labs(x = "Medical Procedure", y = "Total Amount Paid (in Thousands)",
        title = "Top 8 Medicare Procedures in South Dakota in 2020") +
    scale_y_continuous(labels = comma) + ylim(0, 250) + theme_bw() +
    theme(plot.title = element_text(size = 12), axis.text.x = element_text(size = 8,
        angle = 90), axis.text.y = element_text(size = 8),
        axis.title = element_text(size = 10), plot.background = element_rect(fill = "Grey"),
        legend.position = "none")
```



Analysis of Plot 1 Results: The results indicate Top payments made in South Dakota in 2020 by Medicare was for ***Spinal fusion procedures*** followed by ***Hip & Knee joint treatments***.

- ***Plot 2: Top 10 Hospitalizations in Mobile, Alabama in 2020***

```r
# Top 10 Hospitalizations in Mobile, Alabama in 2020
ggplot(data = mobileAL_hosp_df_2020, aes(str_wrap(str_sub(DRG_Desc,
    1, 40), 15), Hospitalizations)) + geom_bar(stat = "identity",
    width = 0.5, aes(fill = Hospitalizations)) + scale_fill_gradient(low = "red",
    high = "blue") + coord_flip() + labs(x = "Medical Procedure",
    y = "Number of Hospitalizations", title = "Top 10 Hospitalizations in Mobile,AL in 2020") +
    geom_text(aes(label = Hospitalizations), hjust = -0.5,
        size = 3) + theme_bw() + theme(plot.title = element_text(size = 12),
    axis.text.x = element_text(size = 8, angle = 90), axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10), plot.background = element_rect(fill = "Grey"))
```



Analysis of Plot 2 Results: Results indicate that Most hospitalization stays in Mobile,Alabama in 2020 was for ***Extra cranial Procedures*** followed by ***Craniotomy & endovascular procedures***

- ***Plot 3: Chemotherapy treatment in West Virginia***

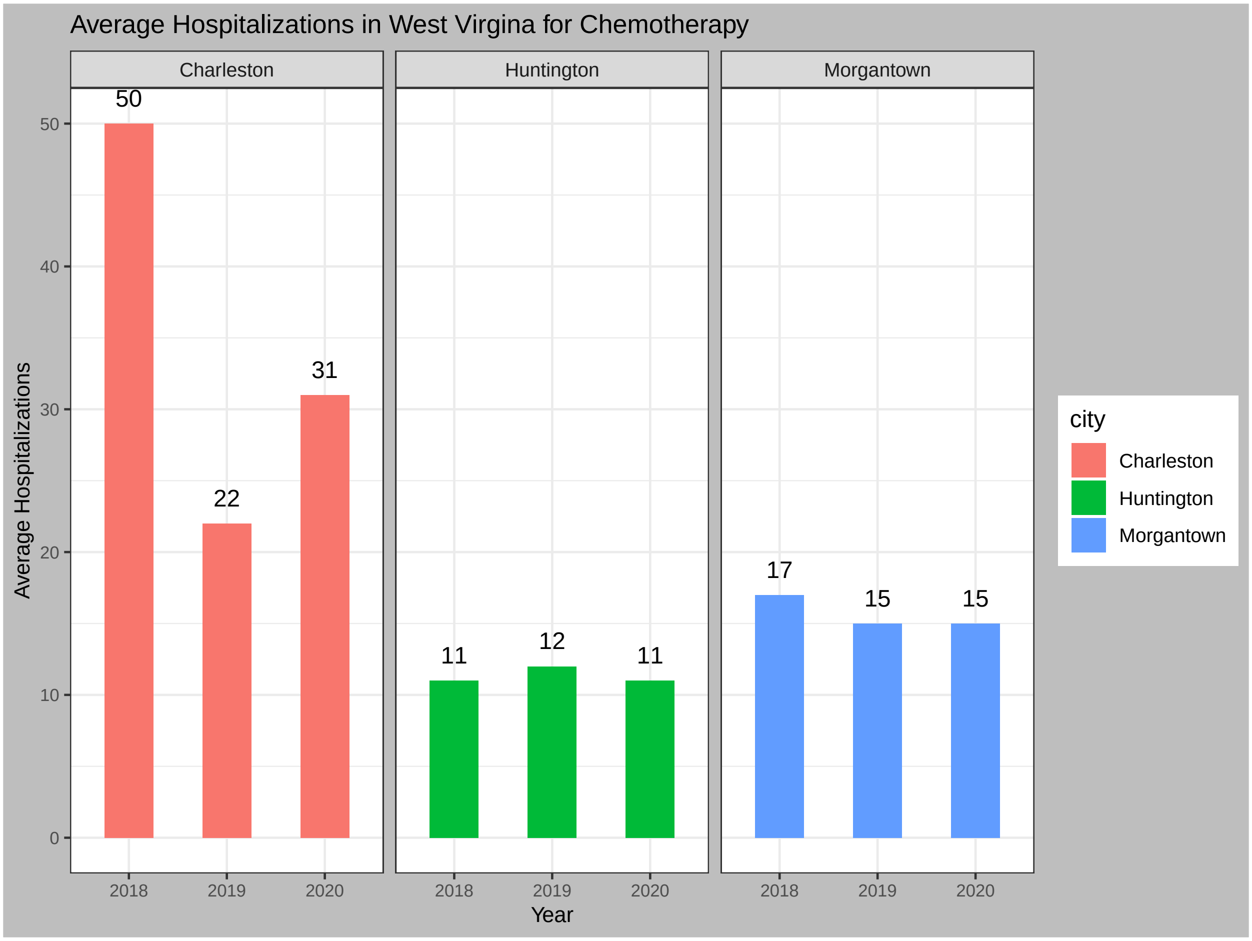
```
# Creating a Summary dataframe from combined data based
# on total & average payments, total and average
# hospitalizations based on the variables Year, Medical
# procedure, State and City
prvsvc_comb_summary_df1 <- medic_prvsvc_combined_df %>%
    group_by(year, DRG_Cd, DRG_Desc, state, city) %>%
    dplyr::summarise(Total_payments = sum(Avg_Tot_Pymt_Amt),
        Average_payment = mean(Avg_Tot_Pymt_Amt), total_Hospitalization = sum(Tot_Dschrgs),
        avg_Hospitalization = mean(Tot_Dschrgs))
WV_Chemo_df <- prvsvc_comb_summary_df1 %>%
```

Table 7: Chemotherapy treatment in West Virginia

year	DRG_Cd	state	city	total_Hospitalization	avg_Hospitalization
2018	847	WV	Charleston	50	50
2018	847	WV	Huntington	11	11
2018	847	WV	Morgantown	34	17
2019	847	WV	Charleston	22	22
2019	847	WV	Huntington	12	12
2019	847	WV	Morgantown	15	15

```r
    filter(DRG_Cd == "847" & state == "WV") %>%
    topN(., "city")
# Printing The results of the WV_Chemo_df
kbl(head(WV_Chemo_df[, c(1:2, 4:5, 8:9)]), caption = "Chemotherapy treatment in West Virginia",
    booktabs = T) %>%
    kable_styling(full_width = F, latex_options = c("striped"))
```

```r
# Chemotherapy treatment in West Virginia
ggplot(WV_Chemo_df, aes(year, avg_Hospitalization)) + geom_bar(stat = "identity",
    width = 0.5, aes(fill = city)) + geom_text(aes(label = avg_Hospitalization),
    vjust = -1) + facet_wrap(~city) + theme_bw() + labs(x = "Year",
    y = "Average Hospitalizations", title = "Average Hospitalizations in West Virgina for Chemotherapy")
    theme(plot.title = element_text(size = 12), axis.text.x = element_text(size = 8),
        axis.text.y = element_text(size = 8), axis.title = element_text(size = 10),
        plot.background = element_rect(fill = "Grey"))
```



Analysis of Plot 3 Results: Results indicate most number of Hospitalization stays for ***Chemotherapy treatment*** in WV was in ***Charleston*** between 2018 to 2020 followed by ***Morgantown***.

- ***Plot 4: Hospitalizations in Kansas City, MO vs Kansas City,KS for Liver Ailments***

```
# Liver related procedures in Kansas City MO and KS
kansasCity_df <- medic_prvsvc_combined_df %>%
    filter(city %in% "Kansas City" & state %in% c("MO", "KS") &
        grepl("Liver", medic_prvsvc_combined_df$DRG_Desc, ignore.case = TRUE)) %>%
    group_by(year, state) %>%
    dplyr::summarise(Tot_payment = sum(Avg_Tot_Pymt_Amt), Avg_payment = mean(Avg_Tot_Pymt_Amt),
        Total_discharges = sum(Tot_Dschrgs), Avg_discharges = mean(Tot_Dschrgs)) %>%
    arrange(year)

# Printing The results of the kansasCity_df
```

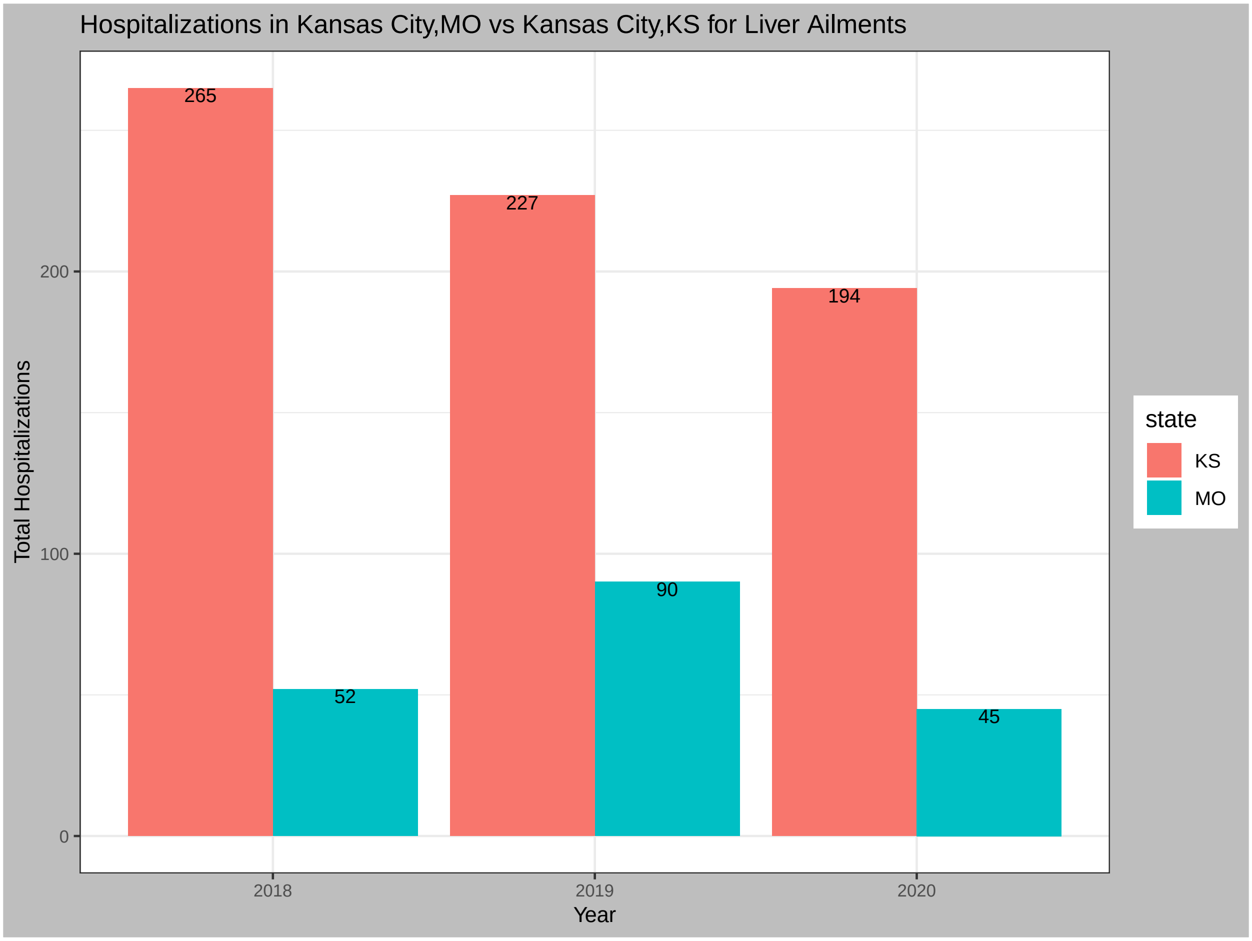
Table 8: Liver treatment in Kansas City

year	state	Tot_payment	Avg_payment	Total_discharges	Avg_discharges
2018	KS	256876.00	42812.67	265	44.16667
2018	MO	59911.77	19970.59	52	17.33333
2019	KS	226576.14	32368.02	227	32.42857
2019	MO	134934.12	26986.82	90	18.00000
2020	KS	186266.70	31044.45	194	32.33333
2020	MO	65865.98	21955.33	45	15.00000

```
kbl(head(kansasCity_df[, 1:6]), caption = "Liver treatment in Kansas City",
    booktabs = T) %>%
    kable_styling(full_width = F, latex_options = c("striped")) %>%
    column_spec(2, width = "15em")
```

```
# Hospitalizations in Kansas City, MO vs KS for Liver
# Ailments
ggplot(kansasCity_df, aes(year, Total_discharges, fill = state)) +
    geom_col(position = "dodge") + geom_text(aes(label = Total_discharges,
    fill = state), color = "black", size = 3, vjust = 1, position = position_dodge(0.9)) +
    labs(x = "Year", y = "Total Hospitalizations", title = "Hospitalizations in Kansas City,MO vs Kansa
    theme_bw() + theme(plot.title = element_text(size = 12),
    axis.text.x = element_text(size = 8), axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10), plot.background = element_rect(fill = "Grey"))
```

```
## Warning in geom_text(aes(label = Total_discharges, fill = state), color =
## "black", : Ignoring unknown aesthetics: fill
```



Analysis of Plot 4 Results: Results indicate that most of the patients went ***chose KansasCity, KS hospitals than Kansascity, MO hospitals*** for inpatient services. Most research to be done to analyze the cost of treatments across these states to understand the pattern.

- ***Plot 5: National Average vs Wyoming for Cost of Heart related Ailments treatment***

```
# National Average vs Treatment costs in Wyoming for
# Heart related Ailments
wyoming_heart_df <- medic_geo_combined_df %>%
    filter(year %in% c("2020", "2019", "2018") & Geo_desc %in%
        c("National", "Wyoming") & DRG_Cd %in% c("269", "291",
        "292", "293")) %>%
    select(DRG_Cd, DRG_Desc, Geo_desc, Avg_Tot_Pymt_Amt, year)
# Using Spread function to create new columns for
# National average and Wyoming
```

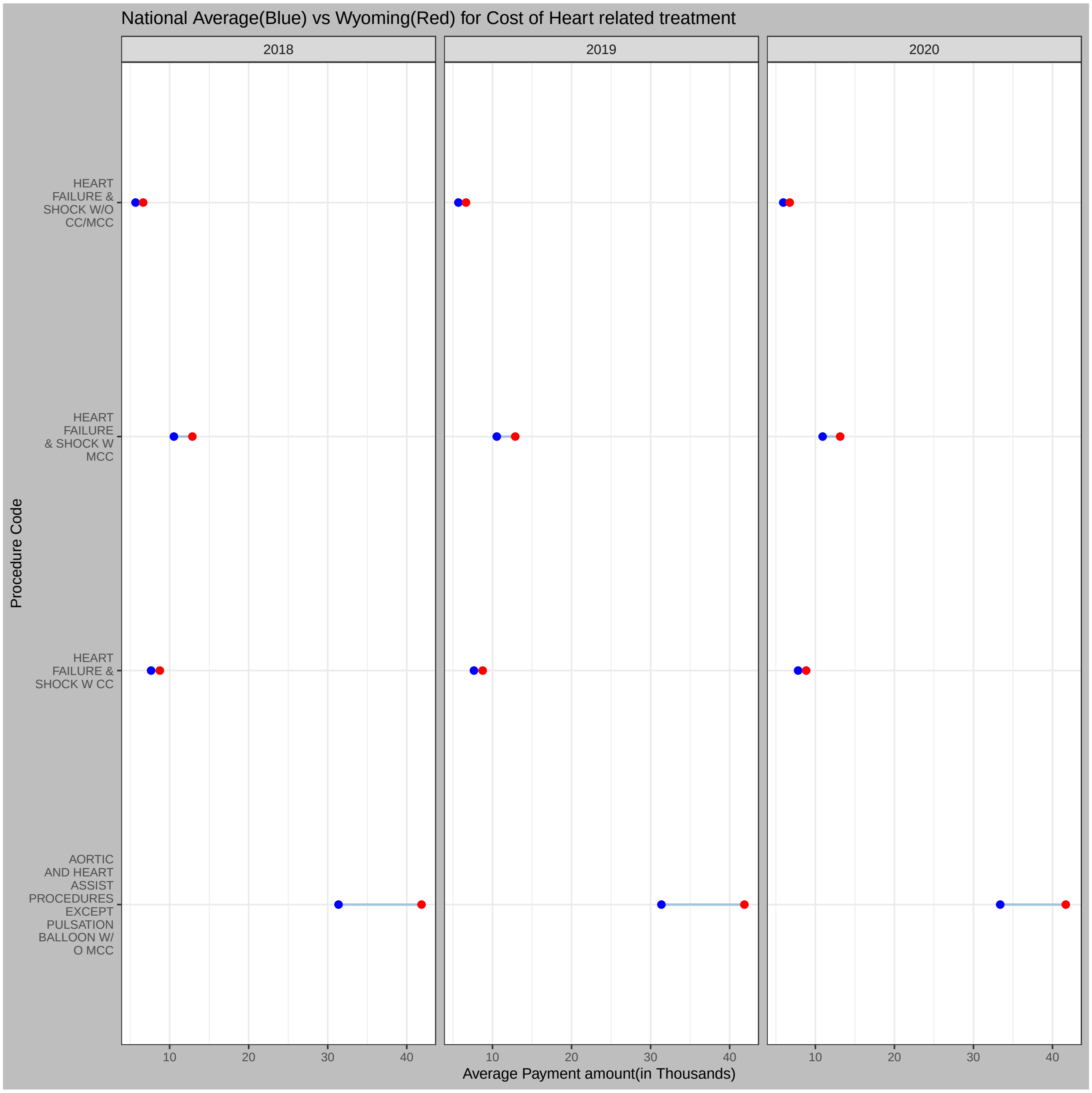
Table 9: National Average vs Wyoming for Cost of Heart related Ailments treatment

DRG_Cd	DRG_Desc	year	National	Wyoming
269	AORTIC AND HEART ASSIST PROCEDURES EXCEPT PULSATION BALLOON W/O MCC	2018	31366.63	41865.42
269	AORTIC AND HEART ASSIST PROCEDURES EXCEPT PULSATION BALLOON W/O MCC	2019	31366.63	41865.42
269	AORTIC AND HEART ASSIST PROCEDURES EXCEPT PULSATION BALLOON W/O MCC	2020	33384.54	41676.46
291	HEART FAILURE & SHOCK W MCC	2018	10538.62	12875.81
291	HEART FAILURE & SHOCK W MCC	2019	10538.62	12875.81
291	HEART FAILURE & SHOCK W MCC	2020	10918.66	13140.10

```
wyoming_heart_df_spread <- wyoming_heart_df %>%
    spread(key = Geo_desc, value = Avg_Tot_Pymt_Amt)



# Printing The results of the wyoming_heart_df_spread
kbl(head(wyoming_heart_df_spread), caption = "National Average vs Wyoming for Cost of Heart related Ail
    booktabs = T) %>%
    kable_styling(full_width = F, latex_options = c("striped")) %>%
    column_spec(2, width = "15em")
```

```
# National Average vs Wyoming for Heart related Ailments
ggplot(wyoming_heart_df_spread, aes(x = National/1000, xend = Wyoming/1000,
    y = DRG_Desc, group = DRG_Cd)) + geom_dumbbell(color = "#a3c4dc",
    size = 0.75, colour_x = "Blue", colour_xend = "Red", size_x = 2,
    size_xend = 2) + facet_wrap(~year) + labs(x = "Average Payment amount(in Thousands) ",
    y = "Procedure Code", title = "National Average(Blue) vs Wyoming(Red) for Cost of Heart related tre
    color = "Legend") + theme_bw() + theme(plot.title = element_text(size = 12),
    axis.text.x = element_text(size = 8), axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10), plot.background = element_rect(fill = "Grey"),
    legend.position = "bottom") + scale_y_discrete(labels = function(x) str_wrap(x,
    width = 10)) + scale_color_manual(name = "Regression Model",
    breaks = c("Linear", "Quadratic", "Cubic"), values = c(Cubic = "pink",
        Quadratic = "blue", Linear = "purple"))
```

```
## Warning: Using the `size` aesthietic with geom_segment was deprecated in ggplot2 3.4.0.
## i Please use the `linewidth` aesthetic instead.
```



Analysis of Plot 5 Results: Results indicate that the Average cost of heart related treatment in ***Wyoming is slightly above the national average*** except for the ***Aortic and heart assist procedures***.

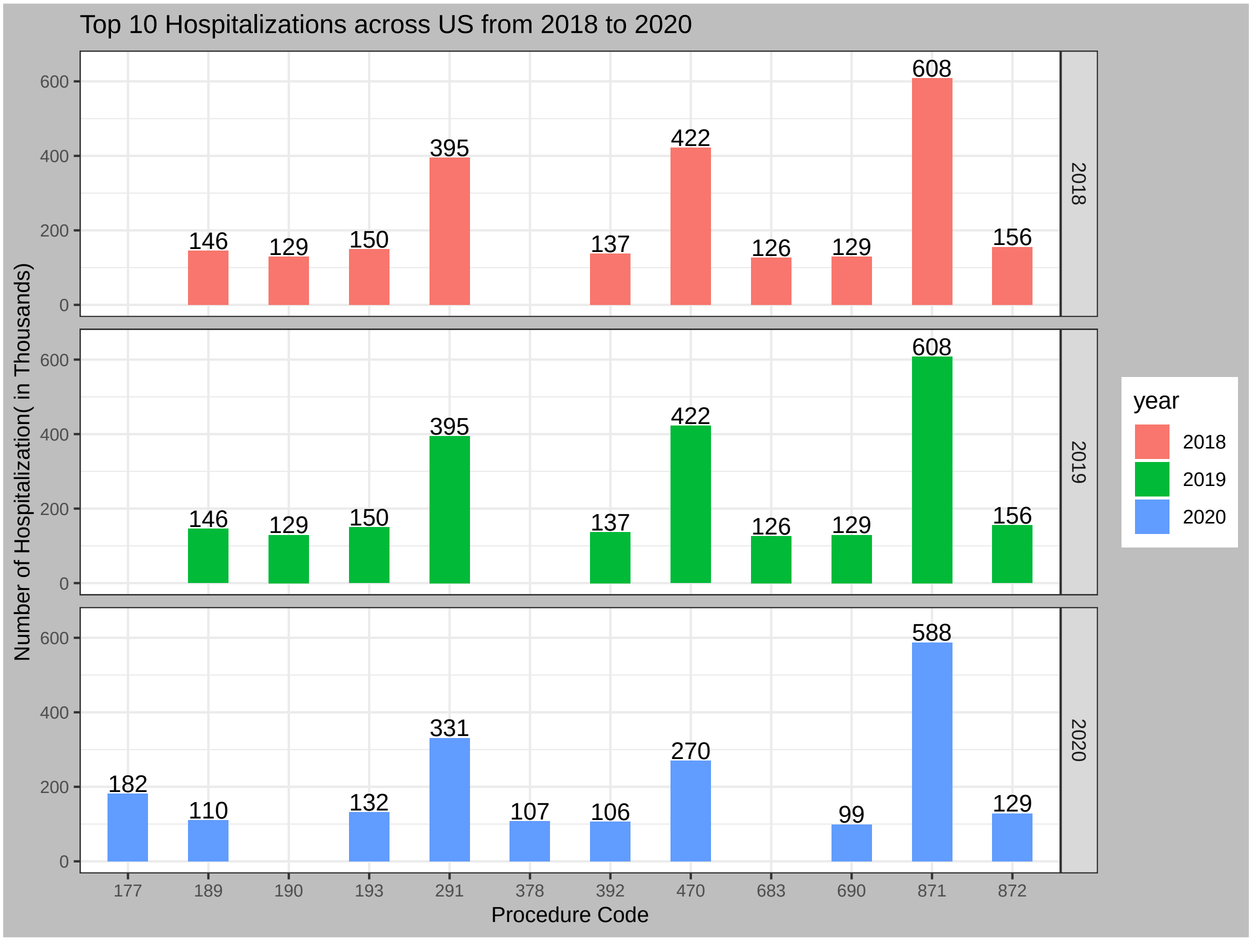
- ***Plot 6 : Top 10 Hospitalizations in each year across the U.S***

```r
# Top 10 Hospitalizations in each year across the country


Top10_hosp_geo_df_2020 <- medic_geo_combined_df %>%
    filter(Geo_desc == "National" & year %in% c("2020")) %>%
    arrange(desc(Tot_Dschrgs)) %>%
    head(n = 10)
Top10_hosp_geo_df_2019 <- medic_geo_combined_df %>%
    filter(Geo_desc == "National" & year %in% c("2019")) %>%
    arrange(desc(Tot_Dschrgs)) %>%
    head(n = 10)
Top10_hosp_geo_df_2018 <- medic_geo_combined_df %>%
    filter(Geo_desc == "National" & year %in% c("2018")) %>%
    arrange(desc(Tot_Dschrgs)) %>%
    head(n = 10)
# Creating Top10_hosp_geo_df by combining data from above
# 3 Dataframes
Top10_hosp_geo_df <- rbind(Top10_hosp_geo_df_2020, Top10_hosp_geo_df_2019,
    Top10_hosp_geo_df_2018) %>%
    select(DRG_Cd, DRG_Desc, Tot_Dschrgs, year)

# Printing The results of the Top10_hosp_geo_df
kbl(head(Top10_hosp_geo_df), caption = "Top 10 Hospitalizations in each year across U.S",
    booktabs = T) %>%
    kable_styling(full_width = F, latex_options = c("striped")) %>%
    column_spec(2, width = "15em")
```

Table 10: Top 10 Hospitalizations in each year across U.S

DRG_Cd	DRG_Desc	Tot_Dschrgs	year
871	SEPTICEMIA OR SEVERE SEPSIS W/O MV >96 HOURS W MCC	587611	2020
291	HEART FAILURE & SHOCK W MCC	331381	2020
470	MAJOR HIP AND KNEE JOINT REPLACEMENT OR REATTACHMENT OF LOWER EXTREM	270402	2020
177	RESPIRATORY INFECTIONS & INFLAMMATIONS W MCC	181981	2020
193	SIMPLE PNEUMONIA & PLEURISY W MCC	131572	2020
872	SEPTICEMIA OR SEVERE SEPSIS W/O MV >96 HOURS W/O MCC	128651	2020

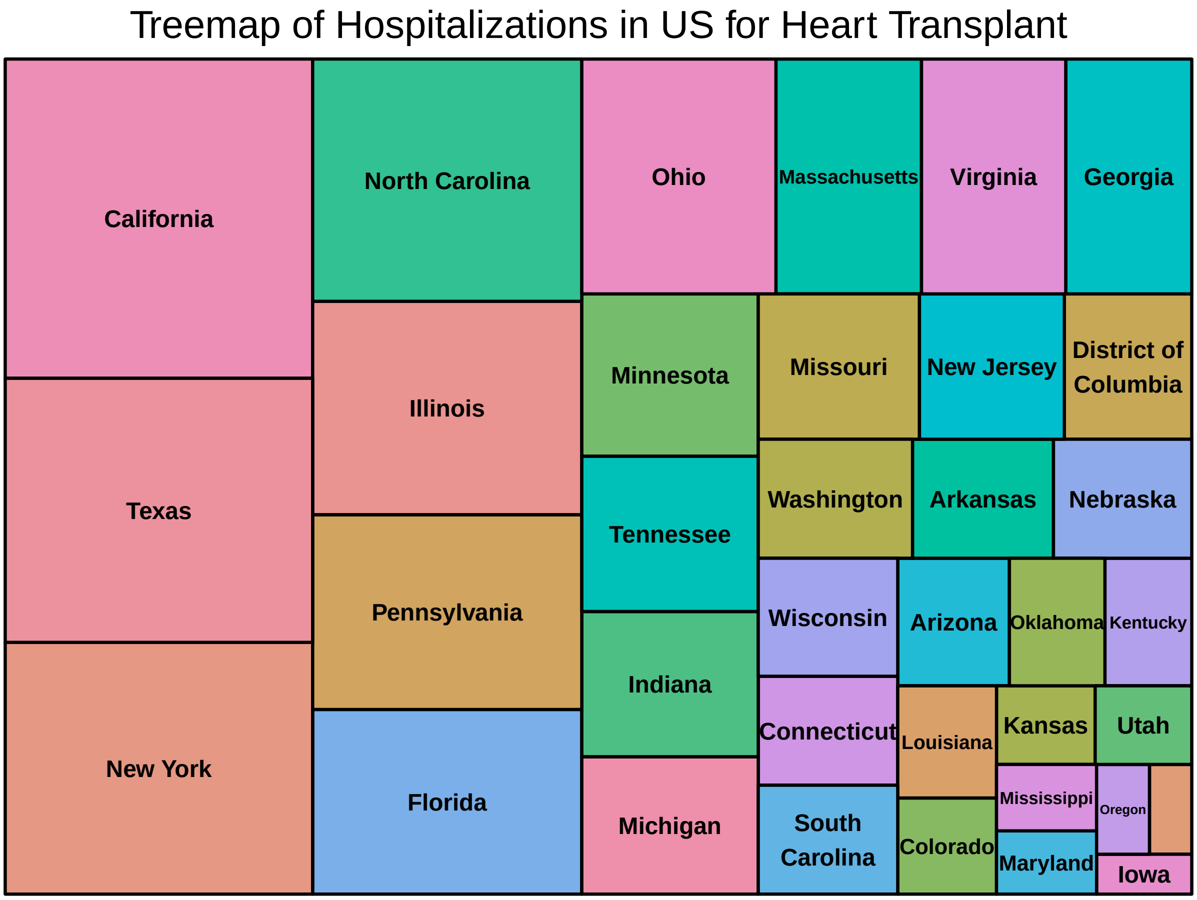
```r
# Top 10 Hospitalizations in each year across the U.S
ggplot(Top10_hosp_geo_df, aes(DRG_Cd, Tot_Dschrgs/1000)) +
    geom_bar(stat = "Identity", width = 0.5, aes(fill = year)) +
    labs(x = "Procedure Code", y = "Number of Hospitalization( in Thousands)",
        title = "Top 10 Hospitalizations across US from 2018 to 2020") +
    facet_grid(year ~ .) + ylim(0, 650) + theme_bw() + theme(plot.title = element_text(size = 12),
    axis.text.x = element_text(size = 8), axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10), plot.background = element_rect(fill = "Grey")) +
    geom_text(aes(label = (round(Tot_Dschrgs/1000)), vjust = -0.1))
```



Analysis of Plot 6 Results: Results are shown based on the Procedure code due to space constraints. The table can be used to identify the name of the Procedure description which indicates that maximum hospitalization across US from 2018 to 2020 was for ***SEPTICEMIA OR SEVERE SEPSIS*** followed by ***MAJOR HIP AND KNEE JOINT REPLACEMENT*** and ***HEART FAILURES***

- ***Plot 7: Creating a Treemap of Hospitalizations in the US for Heart Transplant Procedures from 2018 to 2020***

```
# Treemap of Hospitalizations in the US for Heart
# Transplant Procedures from 2018 to 2020
treemap(medic_geo_combined_df %>%
    filter(Geo_lvl == "State" & DRG_Cd == "1"), index = "Geo_desc",
    vSize = "Tot_Dschrgs", type = "index", title = "Treemap of Hospitalizations in US for Heart Transpla
    fontsize.title = 18)
```



Analysis of Plot 7 Results: The Treemap indicates that most hospitalizations in the US for Heart transplant was in ***California followed by Texas, New York, North Carolina, Illinois, Pennsylvania and others***. Surprisingly, ***Washington DC*** with population less than a million has more hospitalizations than states like ***Colorado, Arizona ,South Carolina*** which are more populated than DC.

- ***Plot 8: Most Profitable hospitals in Nebraska***

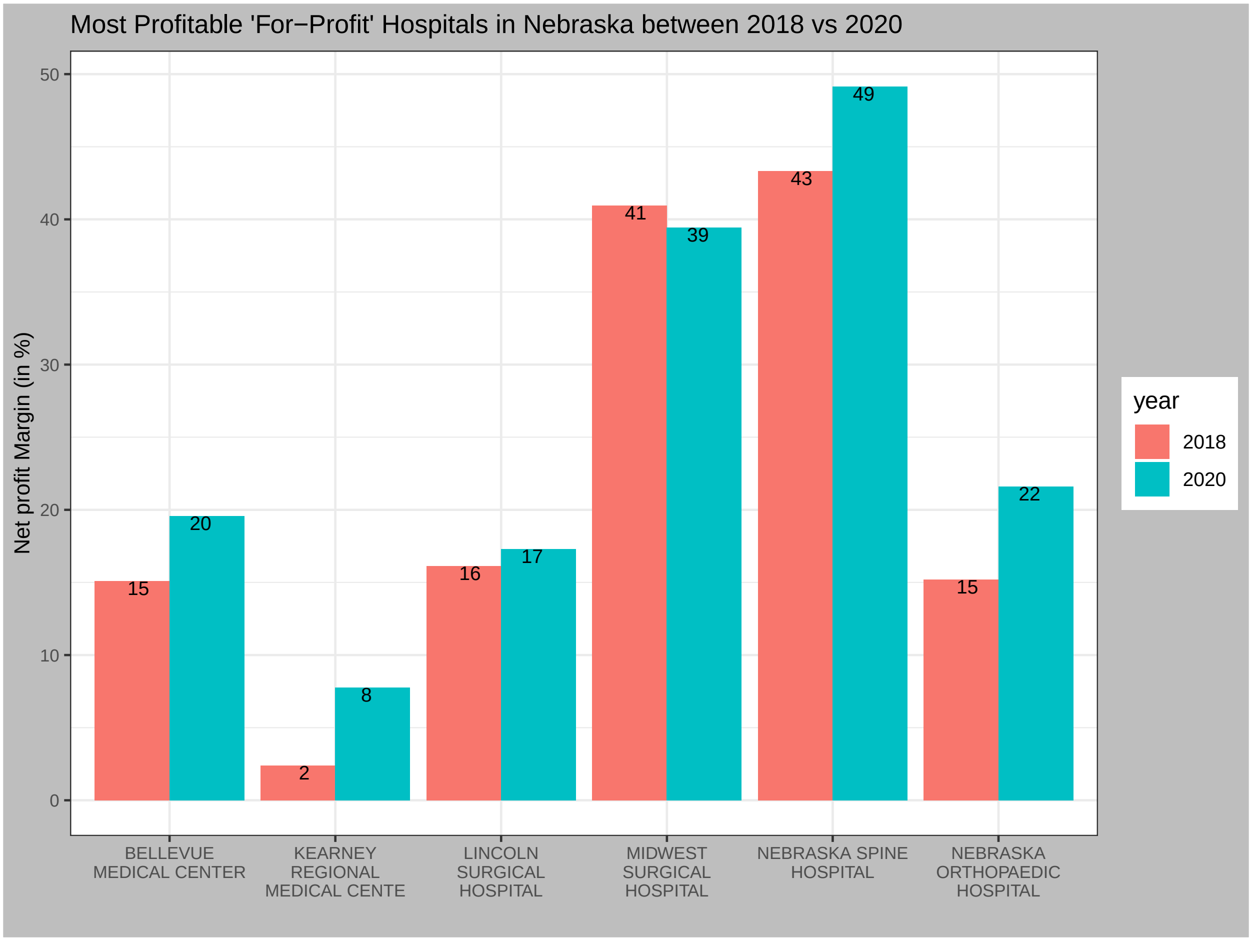
```
nebraska_hosp_df <- NASHP_df %>%
    filter(state == "NE" & hosp_ownership == "For-Profit" &
        year %in% c("2018", "2020") & net_income > 0) %>%
    group_by(year) %>%
    arrange(year, desc(net_profit_margin)) %>%
    select(CCN, year, hosp_name, city, state, health_system,
        net_income, net_profit_margin)
```

Table 11: Most Profitable 'For-Profit' Hospitals in Nebraska between 2018 vs 2020

year	hosp_name	city	state	net_income	net_profit_margin
2018	NEBRASKA SPINE HOSPITAL	OMAHA	NE	19942302	43.33
2018	MIDWEST SURGICAL HOSPITAL	OMAHA	NE	21566392	40.95
2018	LINCOLN SURGICAL HOSPITAL	LINCOLN	NE	10025338	16.14
2018	NEBRASKA ORTHOPAEDIC HOSPITAL LLC	OMAHA	NE	12222977	15.21
2018	BELLEVUE MEDICAL CENTER	BELLEVUE	NE	10999128	15.09
2018	KEARNEY REGIONAL MEDICAL CENTER	KEARNEY	NE	2831353	2.40

```
# Printing The results of the nebraska_hosp_df
kbl(head(nebraska_hosp_df[1:6, c(2:5, 7:8)]), caption = "Most Profitable 'For-Profit' Hospitals in Nebra
    booktabs = T) %>%
    kable_styling(full_width = F, latex_options = c("striped"))
```

```
# Most Profitable hospitals in Nebraska
ggplot(nebraska_hosp_df, aes(str_wrap(str_sub(hosp_name, 1,
    30), 15), net_profit_margin, fill = year)) + geom_col(position = "dodge") +
    geom_text(aes(label = round(net_profit_margin)), color = "black",
        size = 3, vjust = 1, position = position_dodge(0.75)) +
    labs(x = "", y = "Net profit Margin (in %)", title = "Most Profitable 'For-Profit' Hospitals in Neb
    theme_bw() + theme(plot.title = element_text(size = 12),
    axis.text.x = element_text(size = 8), axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10), plot.background = element_rect(fill = "Grey"))
```



Analysis of Plot 8 Results: Results indicate ***Nebraska Spine Hospital*** in Omaha is the most profitable across the state followed by ***Midwest Surgical hospital*** in Omaha (again!) for the period 2018 and 2020. Also these results indicate the most profitable hospitals are located on the ***eastern part of Nebraska*** which is more populated than other regions of the state(except Kearney, NE).

- ***Plot 9: Net Profit margin vs payments per patient 'for-Profit' hospital in Anchorage,Alaska***

```
# How many 'For-profit' Hospitals operate in Alaska
NASHP_df %>%
    filter(state == "AK" & hosp_ownership == "For-Profit") %>%
    distinct(CCN, city)
```

```
## # A tibble: 3 x 2
##    CCN     city
##   <fct>  <chr>
## 1 020006 PALMER
## 2 020017 ANCHORAGE
## 3 021307 CORDOVA
```

```
# checking the profit margin of 'For-Profit' hospital in
# Anchorage, Alaska
NASHP_Anchorage_df <- NASHP_df %>%
    filter(CCN %in% c("020017") & year %in% c("2018", "2019",
        "2020")) %>%
    select(CCN, hosp_name, city, state, year, net_income, net_profit_margin)



# Extracting the results from other dataframe
# medic_prvsvc_combined_df for the same Anchorage
# hospital 020017
medic_prvsvc_Anchorage_df <- medic_prvsvc_combined_df %>%
    filter(CCN %in% c("020017")) %>%
    group_by(CCN, year) %>%
    dplyr::summarise(total_payment = sum(Avg_Tot_Pymt_Amt),
        total_discharges = sum(Tot_Dschrgs), payment_per_patient = total_payment/total_discharges)
medic_prvsvc_Anchorage_df
```

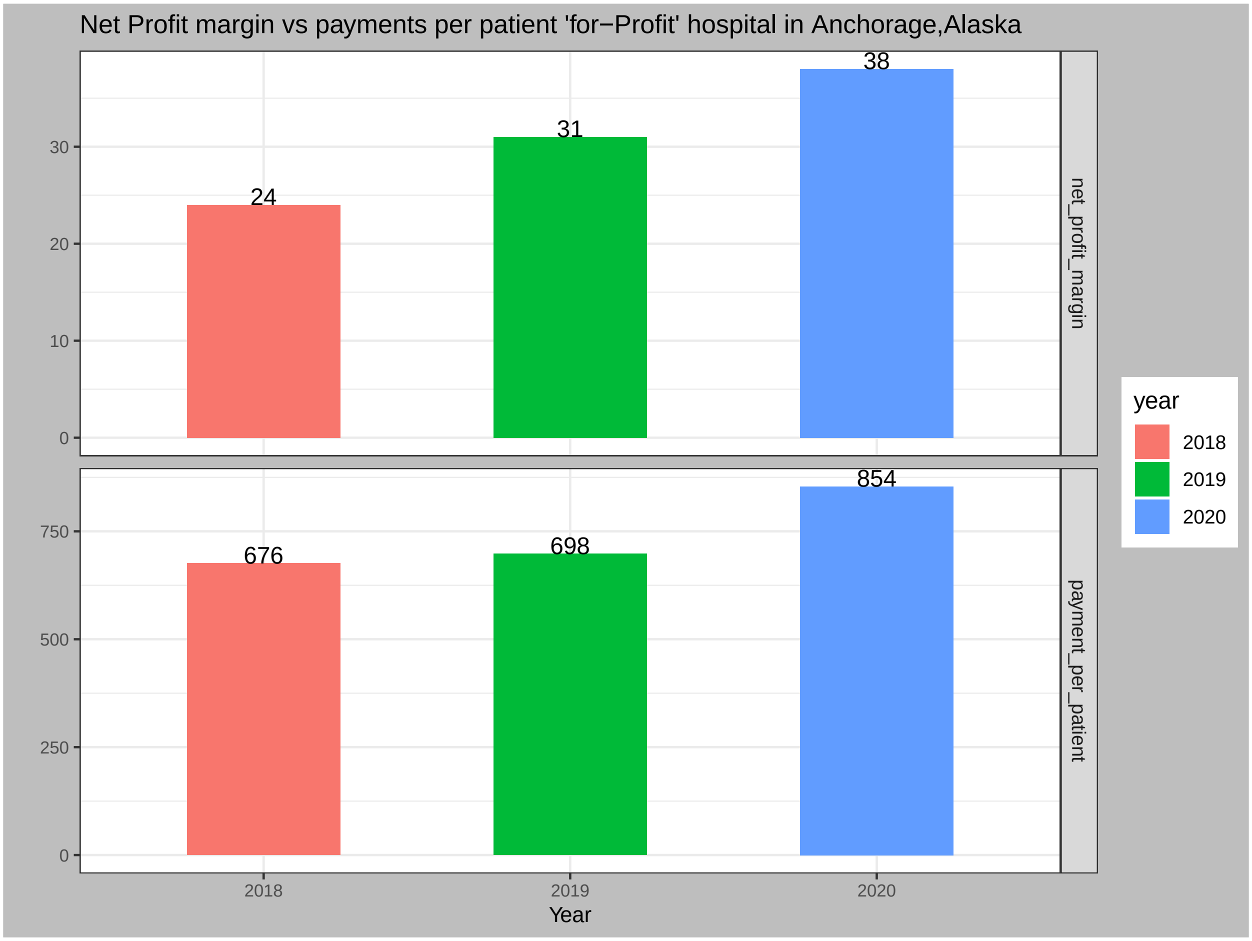
```
## # A tibble: 3 x 5
## # Groups:   CCN [1]
##   CCN    year  total_payment total_discharges payment_per_patient
##   <chr>  <fct>         <dbl>            <int>               <dbl>
## 1 020017 2018        990021.             1465                676.
## 2 020017 2019       1347698.             1931                698.
## 3 020017 2020       1302531.             1526                854.
```

Table 12: Most Profitable Hospitals in Anchorage Alaska between 2018 to 2020

CCN	year	hosp_name	city	state	net_income	total_discharges	Type
020017	2018	ALASKA REGIONAL HOSPITAL	ANCHORAGE	AK	62828656	1465	net_profit_margin
020017	2019	ALASKA REGIONAL HOSPITAL	ANCHORAGE	AK	96080416	1931	net_profit_margin
020017	2020	ALASKA REGIONAL HOSPITAL	ANCHORAGE	AK	123580707	1526	net_profit_margin
020017	2018	ALASKA REGIONAL HOSPITAL	ANCHORAGE	AK	62828656	1465	payment_per_patie
020017	2019	ALASKA REGIONAL HOSPITAL	ANCHORAGE	AK	96080416	1931	payment_per_patie
020017	2020	ALASKA REGIONAL HOSPITAL	ANCHORAGE	AK	123580707	1526	payment_per_patie

```r
# Joining both the dataframes based on the Hospital code
# and year

Anchorage_df <- merge(x = NASHP_Anchorage_df, y = medic_prvsvc_Anchorage_df,
    by = c("CCN", "year")) %>%
    select(-total_payment) %>%
    gather(key = Type, value = Stats, net_profit_margin, payment_per_patient)
Anchorage_df$Stats <- round(Anchorage_df$Stats)

# Printing The results of the Anchorage_df
kbl(head(Anchorage_df), caption = "Most Profitable Hospitals in Anchorage Alaska between 2018 to 2020",
    booktabs = T) %>%
    kable_styling(full_width = F, latex_options = c("striped")) %>%
    column_spec(c(3, 8), width = "10em")
```

```r
ggplot(Anchorage_df, aes(x = year, y = Stats, fill = year)) +
    geom_bar(stat = "identity", position = "dodge", width = 0.5) +
    facet_grid(Type ~ ., scales = "free_y") + labs(x = "Year",
    y = "", title = "Net Profit margin vs payments per patient 'for-Profit' hospital in Anchorage,Alask
    geom_text(aes(label = Stats), vjust = 0) + theme_bw() +
    theme(plot.title = element_text(size = 12), axis.text.x = element_text(size = 8),
        axis.text.y = element_text(size = 8), axis.title = element_text(size = 10),
        plot.background = element_rect(fill = "Grey"))
```



Analysis of Plot 9 Results: The results indicate same pattern in net profit margin vs Payments made by Medicare per patient between 2018 to 2020 for the ***Alaska regional Hospital***. Also the table indicates that the hospital ***treated lesser patients*** in 2020 than in 2019, but ***made more profit*** than in 2019. More analysis is required to understand the reasoning behind it.

- ***Plot 10: Comparing Cost of Spinal Procedure - National Average vs State***

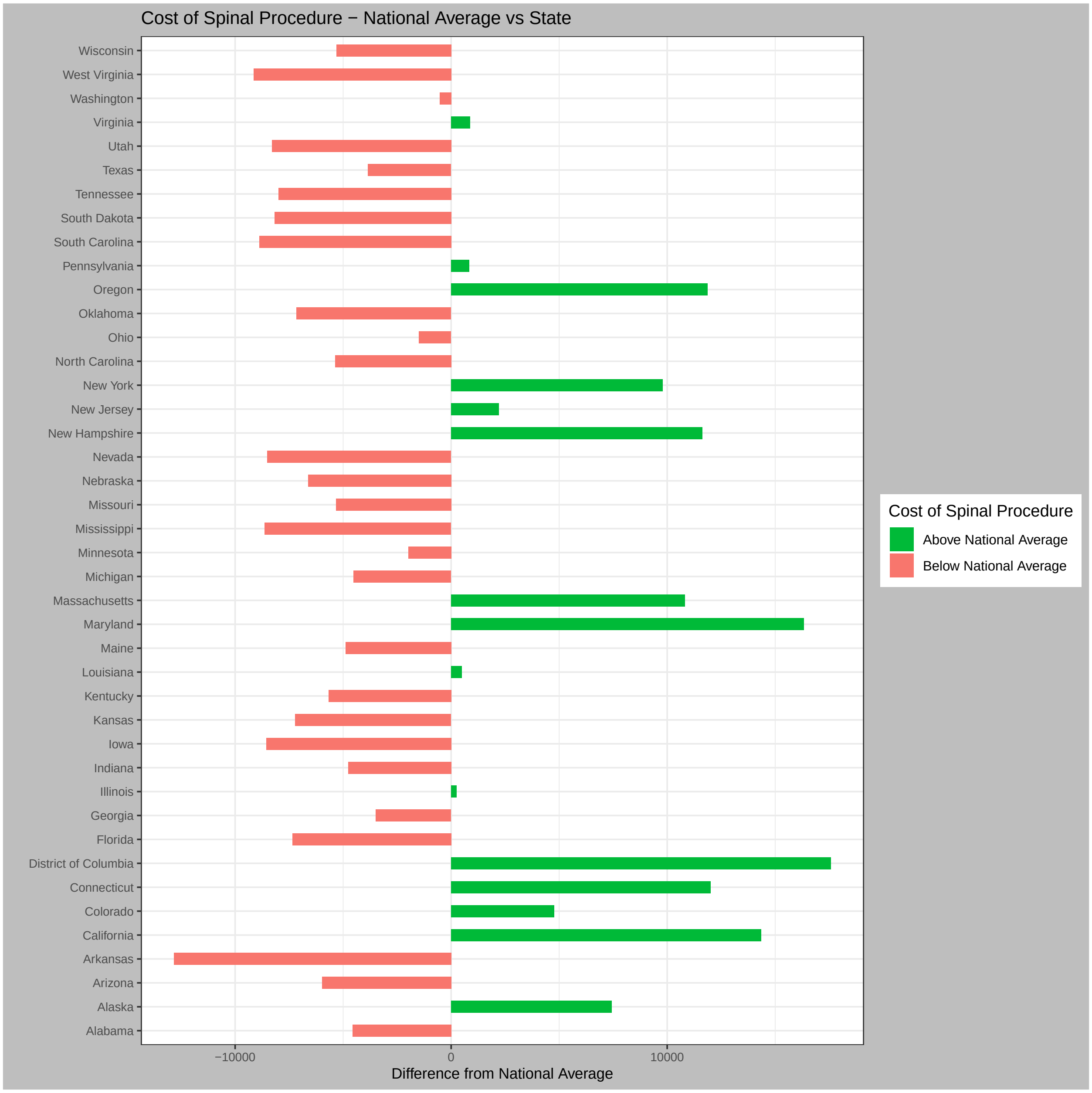
```
spinal_2020_national_avg <- medic_geo_combined_df %>%
    filter(Geo_lvl == "National" & DRG_Cd == "28" & year ==
        "2020") %>%
    select(Avg_Tot_Pymt_Amt)
spinal_2020_df <- medic_geo_combined_df %>%
    filter(Geo_lvl == "State" & DRG_Cd == "28" & year == "2020") %>%
```

Table 13: Cost of Spinal Procedure- National Average vs State

	Geo_desc	Avg_Tot_Pymt_Amt	difference	comparePrice
7	Arkansas	35743.00	-12830.020	below
24	West Virginia	39427.00	-9146.020	below
38	South Carolina	39695.48	-8877.543	below
15	Mississippi	39941.20	-8631.820	below
42	Iowa	40014.26	-8558.759	below
23	Nevada	40064.00	-8509.020	below

```
    select(Geo_desc, Avg_Tot_Pymt_Amt) %>%
    mutate(difference = (Avg_Tot_Pymt_Amt - as.numeric(spinal_2020_national_avg)))
spinal_2020_df$comparePrice <- ifelse(spinal_2020_df$difference <
    0, "below", "above")
spinal_2020_df <- spinal_2020_df[order(spinal_2020_df$difference),
    ]

kbl(head(spinal_2020_df), caption = "Cost of Spinal Procedure- National Average vs State",
    booktabs = T) %>%
    kable_styling(full_width = F, latex_options = c("striped"))
```

```
ggplot(spinal_2020_df, aes(x = Geo_desc, y = difference)) +
    geom_bar(stat = "identity", aes(fill = comparePrice), width = 0.5) +
    scale_fill_manual(name = "Cost of Spinal Procedure", labels = c("Above National Average",
        "Below National Average"), values = c(above = "#00ba38",
        below = "#f8766d")) + coord_flip() + theme_bw() + theme(plot.title = element_text(size = 12),
    axis.text.x = element_text(size = 8), axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10), plot.background = element_rect(fill = "Grey")) +
    labs(x = "", y = "Difference from National Average", title = "Cost of Spinal Procedure - National A
```



15 Do you plan on incorporating any machine learning techniques to answer your research questions? Explain.

- Clustering algorithms such as ***K-means clustering*** can be used in this research to identify the clusters by studying the patterns of hospitals that operate for profit and non-profit. This can help us ***identify which region of the US has the maximum number of hospitals*** that operates “for-profit” and try to find the reasoning behind it.

- Also, the clustering algorithms can be used to ***study the patterns of the number of hospitalizations*** for a medical procedure across the country and try to find the cluster where it is more common and then find out why.
- Another interesting thing to demonstrate is to ***find the clusters of hospitals*** located across the country and try to ***match them with the population distribution of the US***, though this may require additional datasets and integration with Maps.

16 Questions for future steps

So far in this paper, various plots have been built to illustrate the hidden data within the datasets. In the future steps, the focus will be to put everything together and illustrate how this research addresses the problem statement and answer the ten questions that we began with in Part 1. Also, a regression model will be built to study the pattern and predict the Payment amount for a medical procedure.

Also, a new column called "regions" will be included to help identify the clusters in the data, and Machine learning algorithms will be applied to find segments and derive conclusions based on that.

17 Session Info

```r
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22621)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] reshape2_1.4.4   kableExtra_1.3.4 lemon_0.4.6      knitr_1.41
##  [5] readxl_1.4.1     treemap_2.4-3    ggalt_0.4.0      stringr_1.4.1
##  [9] scales_1.2.1     DT_0.27          tidyr_1.2.1      lubridate_1.9.0
## [13] timechange_0.1.1 ggthemes_4.2.4   ggplot2_3.4.0    dplyr_1.0.10
## [17] magrittr_2.0.3   readr_2.1.3
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.4         maps_3.4.1         viridisLite_0.4.1  shiny_1.7.3
##  [5] assertthat_0.2.1   highr_0.9          cellranger_1.1.0   yaml_2.3.6
##  [9] Rttf2pt1_1.3.12    pillar_1.8.1       lattice_0.20-45    glue_1.6.2
## [13] extrafontdb_1.0    digest_0.6.30      RColorBrewer_1.1-3 promises_1.2.0.1
## [17] rvest_1.0.3        colorspace_2.0-3   htmltools_0.5.3    httpuv_1.6.6
## [21] plyr_1.8.8         pkgconfig_2.0.3    purrr_0.3.5        xtable_1.8-4
## [25] webshot_0.5.4      svglite_2.1.1      later_1.3.0        tzdb_0.3.0
## [29] tibble_3.1.8       farver_2.1.1       generics_0.1.3     ellipsis_0.3.2
## [33] withr_2.5.0        cli_3.4.1          mime_0.12          evaluate_0.18
## [37] ash_1.0-15         fansi_1.0.3        MASS_7.3-58.1      xml2_1.3.3
## [41] tools_4.2.2        data.table_1.14.4  hms_1.1.2          formatR_1.12
## [45] lifecycle_1.0.3    gridBase_0.4-7     munsell_0.5.0      compiler_4.2.2
## [49] systemfonts_1.0.4  rlang_1.0.6        grid_4.2.2         rstudioapi_0.14
## [53] htmlwidgets_1.5.4  igraph_1.3.5       labeling_0.4.2     rmarkdown_2.18
## [57] proj4_1.0-12       gtable_0.3.1       DBI_1.1.3          R6_2.5.1
## [61] gridExtra_2.3      fastmap_1.1.0      extrafont_0.19     utf8_1.2.2
## [65] KernSmooth_2.23-20 stringi_1.7.8      Rcpp_1.0.9         vctrs_0.5.0
## [69] tidyselect_1.2.0   xfun_0.34
```

References

"Hospital Provider Cost Report." 2022. Centers for Medicare & Medicaid Services. https://catalog.data.gov/dataset/hospital-provider-cost-report.

Konish, Lorie. n.d. "This Is the Real Reason Most Americans File for Bankruptcy." *CNBC*. Accessed February 11, 2023. https://www.cnbc.com/2019/02/11/this-is-the-real-reason-most-americans-file-for-bankruptcy.html.

"Medicare Inpatient Hospitals - by Geography and Service." 2022. Centers for Medicare & Medicaid Services. https://catalog.data.gov/dataset/medicare-inpatient-hospitals-by-geography-and-service-1bbc8.

"Medicare Inpatient Hospitals - by Provider and Service." 2022. Centers for Medicare & Medicaid Services. https://catalog.data.gov/dataset/medicare-inpatient-hospitals-by-provider-and-service-9af02.