D206: Data Cleaning

Performance Assessment

Logan Rosemeyer

Western Governor's University

D206: Data Cleaning

Dr. Eric Straw

December 18, 2022

Contents

A: Question or Decision

Is income a predictor for the means by which the patient was admitted into the hospital initially?

B: Required Variables

Name	Type	Description	Example
CaseOrder	Nominal	Order of the data	1
Customer_id	Nominal	ID of the patient	C412403
Interaction	Nominal	ID for unique interaction	8cd49b13-f45a-4b47-a2bd-173ffa932c2f
UID	Nominal	ID for unique transaction	3a83ddb66e2ae73798bdf1d705dc0932
City	Nominal	Patient's city	Eva
State	Nominal	Patients's state	AL
County	Nominal	Patient's county	Morgan
Zip	Nominal	Patient's Zip Code	35621
Lat	Continuous	Patient's Latitude	34.3
Lng	Continuous	Patient's Longitude	-86.7
Population	Discrete	Population of patient's city	2951
Area	Nominal	Type of area of patient's city	Suburban

Timezone	Nominal	Time zone of patient's city	America/Chicago
Job	Nominal	Patient's Job	Psychologist, sport and exercise
Children	Discrete	Number of children the patient has	1
Age	Discrete	Age of patient in years	53
Education	Ordinal	Patient's level of education	Some College, Less than 1 Year
Employment	Nominal	Patient's employment status	Full Time
Income	Discrete	Patient's yearly income	86576
Marital	Nominal	Patient's marital status	Divorced
Gender	Nominal	Patient's identified gender	Male
ReAdmis	Nominal	Patient readmitted within one month of original hospitalization	0

VitD_Levels	Continuous	Patient's Vitamin D levels	17.8
Doc_visits	Discrete	Number of times the doctor visited the patient	6
Full_meals_eaten	Discrete	Number of meals eaten by the patient while hospitalized	2
VitD_supp	Discrete	Times the patient received vitamin D supplements	1
Soft_drink	Nominal	Patient drinks soda 3 or more times a day	0
Initial_admin	Nominal	How the patient was initially admitted to the hospital	Emergency Admission
HighBlood	Nominal	Patient has high blood pressure	1
Stroke	Nominal	Patient has had a stroke	1

Complication_risk	Nominal	Patient's complication risk level	2
Overweight	Nominal	Patient is overweight	1
Arthritis	Nominal	Patient has arthritis	0
Diabetes	Nominal	Patient has diabetes	1
Hyperlipidemia	Nominal	Patient has hyperlipidemia	0
BackPain	Nominal	Patient has chronic back pain	1
Anxiety	Nominal	Patient has Anxiety	0
Allergic_rhinitis	Nominal	Patient has allergic rhinitis	1
Reflux_esophagitis	Nominal	Patient has reflux esophagitis	0
Asthma	Nominal	Patient has Asthma	1
Services	Nominal	Service received by the patient	Blood Work

Initial_days	Continuous	Length of initial hospitalization in days	10.59
TotalCharge	Discrete	Amount charged to patient per day	3191
Additional_charges	Discrete	Additional charges for specialized treatment	17939
Item1	Ordinal	How important quick admission was to patient	1
Item2	Ordinal	How important quick treatment was to patient	2
Item3	Ordinal	How important quick visits were to patient	3
Item4	Ordinal	How important reliability is to patient	4
Item5	Ordinal	How important different options are to patient	5

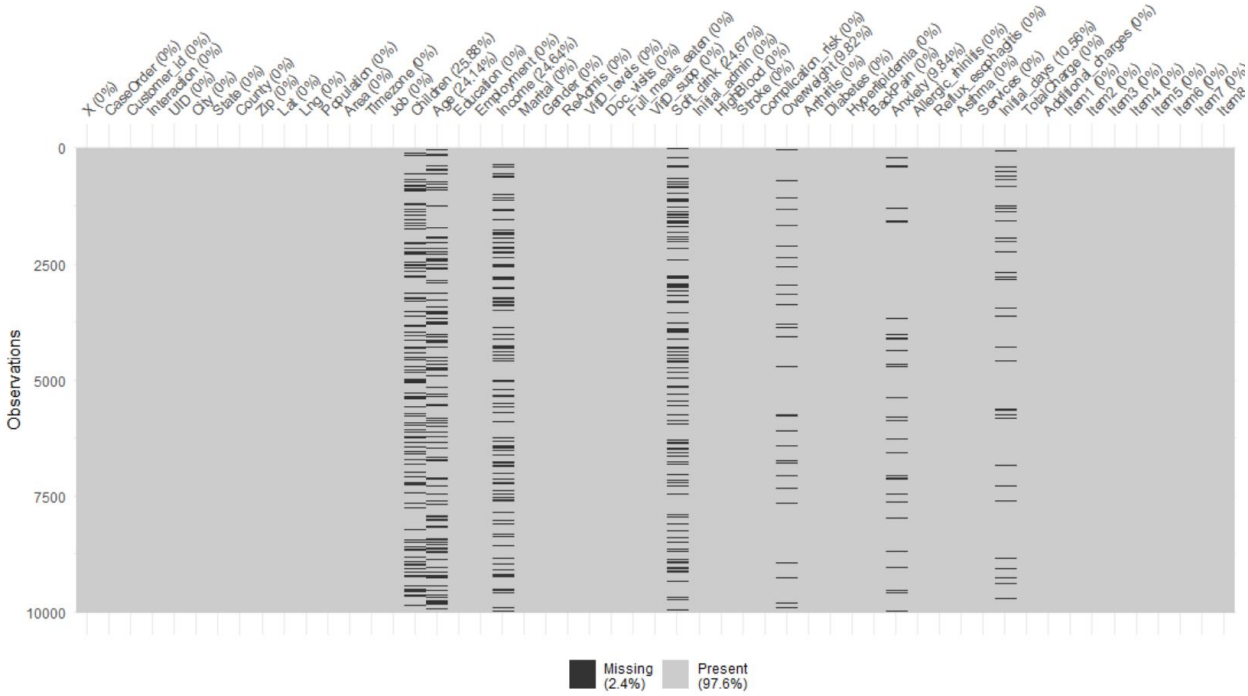
Item6	Ordinal	How important hours of treatment are to patient	6
Item7	Ordinal	How important having courteous staff is to patient	7
Item8	Ordinal	How important active listening from doctor is to patient	8
Education_numeric	Ordinal	Patient's level of education represented numerically	12

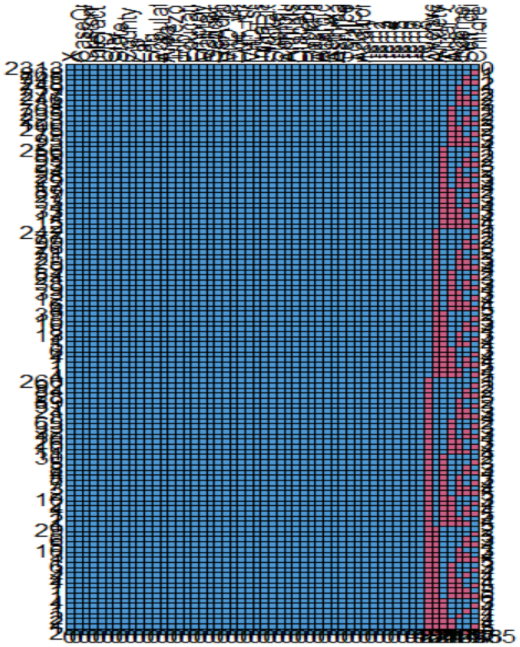
C1: Plan to Find Anomalies

The first thing I wanted to do was to find any duplicate records. To do this I used the sum function with the duplicated function. Next, I wanted to find any missing values. To do this, I used three methods. First, I took a sum of the NAs in each column. Second, I used the vis_miss function to visualize where there was missing data in my dataframe. Last, I used the md.pattern function which is a very similar visualization as the vis_miss function. To check for outliers, I used boxplots. I also re-expressed all categorical variables that where yes/no to 1/0 using recode and as.numeric functions.

C2: Justification of Approach

I selected the sum function with the duplicated function because it gave me exactly how many rows were duplicated. I could also have just used the duplicated function and duplicated rows would have returned TRUE. I used three different methods to find NA values. The first was the colSums function with the is.na function. I used this because it gave me a count of how many NA values were in each column. The second method I used was the vis_miss function. The vis_miss function colors cells with NA values black and cells that do have values grey (vis_miss, n.d.). The third method I used to find NA values was the md.pattern function. This function actually does a good job of combining the two previous methods. There is a visual with blue meaning data is present and pink means data is missing. Rows that have the same columns missing are grouped together. The right side of the visualization gives a count of how many columns are missing. The bottom gives a count of how many NA values are in each column. The left side gives a count of how many rows are grouped together with the same missing columns (Van Buuren, 2018). This method does work better with smaller datasets because, as you can see, everything is pretty crowded. To find outliers, I used boxplots. This worked really well because the boxplot function automatically calculates Q1-1.5*IQR and Q3+1.5*IQR, and plots points outside of those fences as individual points. I also used a histogram to look for outliers, specifically for the total charges column. There was a gap in the histogram that I wasn't able to see with the boxplot that led me to find more outliers.





C3: Justification of Tools

I used R as my programing language and RStudio for my environment. I chose R over Python because I already know Python fairly well and wanted to start learning R. I felt like now was the best time to do it

because I got to apply and learn from actually doing it. I used the visdat library to use the vis_miss visualization. I used the MICE library for imputation. I felt like using MICE would have been better than imputing the mean or median of the dataset. With 10000 observations, I felt like even though I had missing data, there was still enough observations to know what the distribution of each variable should look like. The MICE package does this very well. I used the dplyr package to re-express most of my categorical variables from yes/no to 1/0. I did this because I feel like expressing yes/no numerically can make analysis easier later on while still keeping the necessary information. The last package I used was the factoextra package. I used this package to make the PCA.

C4: Provide the Code

Code for detection is listed below. Code has also been uploaded as a separate file.

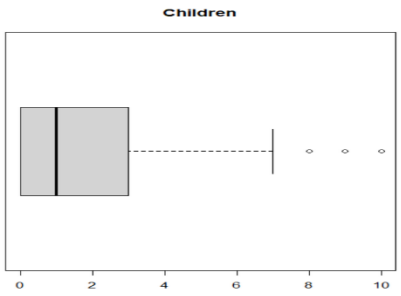
```
#duplicated(medical)
sum(duplicated(medical))

#load visdat and check for missing values
library(visdat)
colSums(is.na(medical))
vis_miss(medical)

#Check for outliers in numeric data
str(imputed_medical)
boxplot(imputed_medical$Children) #Has reasonable outliers
boxplot(imputed_medical$Age)
boxplot(imputed_medical$Income) #Has reasonable outliers
boxplot(imputed_medical$Doc_visits)
boxplot(imputed_medical$VitD_levels) #Has Outliers
boxplot(imputed_medical$Full_meals_eaten) #Has some outliers
boxplot(imputed_medical$VitD_supp) #Has some outliers
boxplot(imputed_medical$Initial_days)
boxplot(imputed_medical$TotalCharge) #Has outliers/should dive deeper
boxplot(imputed_medical$Additional_charges) #Has reasonable outliers
table(imputed_medical$Item1)
table(imputed_medical$Item2)
table(imputed_medical$Item3)
table(imputed_medical$Item4)
table(imputed_medical$Item5)
```

```
table(imputed_medical$Item6)
table(imputed_medical$Item7)
table(imputed_medical$Item8)

#Dive deeper into VitD_levels
#https://www.health.harvard.edu/blog/vitamin-d-whats-right-level-
2016121910893
hist(imputed_medical$VitD_levels, breaks=20)
summary(imputed_medical$VitD_levels)

#Dive deeper into Full_meals_eaten
table(imputed_medical$Full_meals_eaten)
summary(imputed_medical$Full_meals_eaten)

#Dive deeper into VitD_supp
#stackoverflow.com/questions/24332534/r-hist-function-aggregates-zero-
and-1-values-into-one-bin
hist(imputed_medical$VitD_supp, breaks=5, right = F, labels = TRUE)
summary(imputed_medical$VitD_supp)
table(imputed_medical$VitD_supp)


#Dive deeper into Total Charge (per day)
hist(imputed_medical$TotalCharge)
summary(imputed_medical$TotalCharge)

#Replace values to the right of the gap in the histogram with the
median TotalCharge
sort(boxplot(imputed_medical$TotalCharge)$out, decreasing=T)
hist(imputed_medical$TotalCharge)
imputed_medical$TotalCharge[imputed_medical$TotalCharge>11000] <-
median(imputed_medical$TotalCharge)
boxplot(imputed_medical$TotalCharge)
hist(imputed_medical$TotalCharge)

#Dive deeper into Additional_charges
hist(imputed_medical$Additional_charges)

#Start re-expressing categorical variables
library(dplyr)
str(imputed_medical)

#Re-express education
unique(imputed_medical$Education)
```

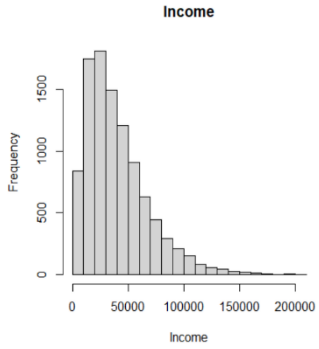
```
edu.num <- recode(imputed_medical$Education, "Some College, Less than
1 Year"=12, "Some College, 1 or More Years, No Degree"=13, "GED or
Alternative Credential"=12, "Regular High School Diploma"=12,
"Bachelor's Degree"=16, "Master's Degree"=18, "Nursery School to 8th
Grade"=8, "9th Grade to 12th Grade, No Diploma"=11, "Doctorate
Degree"=22, "Associate's Degree"=14, "Professional School Degree"=12,
"No Schooling Completed"=0)
imputed_medical$Education_numeric <- as.numeric(edu.num)
str(imputed_medical)
unique(imputed_medical$Education_numeric)

#Re-express ReAdmis
unique(imputed_medical$ReAdmis)
readmis.num <- recode(imputed_medical$ReAdmis, "No"=0, "Yes"=1)
imputed_medical$ReAdmis <- as.numeric(readmis.num)
unique(imputed_medical$ReAdmis)

#Re-express Soft_drink
unique(imputed_medical$Soft_drink)
soft.num <- recode(imputed_medical$Soft_drink, "No"=0, "Yes"=1)
imputed_medical$Soft_drink <- as.numeric(soft.num)
unique(imputed_medical$Soft_drink)

#Re-express HighBlood
unique(imputed_medical$HighBlood)
highblood.num <- recode(imputed_medical$HighBlood, "No"=0, "Yes"=1)
imputed_medical$HighBlood <- as.numeric(highblood.num)
unique(imputed_medical$HighBlood)

#Re-express Stroke
unique(imputed_medical$Stroke)
stroke.num <- recode(imputed_medical$Stroke, "No"=0, "Yes"=1)
imputed_medical$Stroke <- as.numeric(stroke.num)
unique(imputed_medical$Stroke)

#Re-express Arthritis
unique(imputed_medical$Arthritis)
arth.num <- recode(imputed_medical$Arthritis, "No"=0, "Yes"=1)
imputed_medical$Arthritis <- as.numeric(arth.num)
unique(imputed_medical$Arthritis)

#Re-express Diabetes
unique(imputed_medical$Diabetes)
diab.num <- recode(imputed_medical$Diabetes, "No"=0, "Yes"=1)
imputed_medical$Diabetes <- as.numeric(diab.num)
```

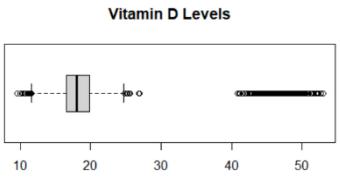
```
unique(imputed_medical$Diabetes)

#Re-express Hyperlipidemia
unique(imputed_medical$Hyperlipidemia)
hyp.num <- recode(imputed_medical$Hyperlipidemia, "No"=0, "Yes"=1)
imputed_medical$Hyperlipidemia <- as.numeric(hyp.num)
unique(imputed_medical$Hyperlipidemia)

#Re-express BackPain
unique(imputed_medical$BackPain)
back.num <- recode(imputed_medical$BackPain, "No"=0, "Yes"=1)
imputed_medical$BackPain <- as.numeric(back.num)
unique(imputed_medical$BackPain)

#Re-express Allergic_rhinitis
unique(imputed_medical$Allergic_rhinitis)
allrhi.num <- recode(imputed_medical$Allergic_rhinitis, "No"=0,
"Yes"=1)
imputed_medical$Allergic_rhinitis <- as.numeric(allrhi.num)
unique(imputed_medical$Allergic_rhinitis)

#Re-express Reflux_esophagitis
unique(imputed_medical$Reflux_esophagitis)
reflux.num <- recode(imputed_medical$Reflux_esophagitis, "No"=0,
"Yes"=1)
imputed_medical$Reflux_esophagitis <- as.numeric(reflux.num)
unique(imputed_medical$Reflux_esophagitis)

#Re-express Asthma
unique(imputed_medical$Asthma)
asthma.num <- recode(imputed_medical$Asthma, "No"=0, "Yes"=1)
imputed_medical$Asthma <- as.numeric(asthma.num)
unique(imputed_medical$Asthma)

#Re-express Complication_risk
unique(imputed_medical$Complication_risk)
comp.num <- recode(imputed_medical$Complication_risk, "Low"=1,
"Medium"=2, "High"=3)
imputed_medical$Complication_risk <- as.numeric(comp.num)
unique(imputed_medical$Complication_risk)
```

D1: Cleaning Findings



I did not find any duplicated observations. I found 2588 missing values in the children variable, 2414 missing values in the age variable, 2464 missing values in the income variable, 2467 missing values in the soft drink variable, 982 missing values in the overweight variable, 984 missing values in the anxiety variable, and 1056 missing values in the initial days variable. I found 415 outliers in the children column. They had values of 8, 9, and 10. I did not find any outliers in the age column. I found 329 outliers in the income column. They ranged from 106220.5 to 207249.1. I didn't find any outliers in the doctor visits. I found 534 outliers in the vitD levels. The lower outliers ranged from 9.5 to 11.6 while the upper outliers ranged from 24.8 to 53.0. I found 8 outliers in the full meals eaten. They had values of 6 and 7. I found 70 outliers in the vitD supplements column. They had values of 3, 4, and 5. I found no outliers in the initial days column. Using a histogram, I found 500 outliers in the total charges column. They range in value from 13090.41 to 21524.22. I found 424 outliers in the additional charges column. They ranged in value from 27088.15 to 30566.07.

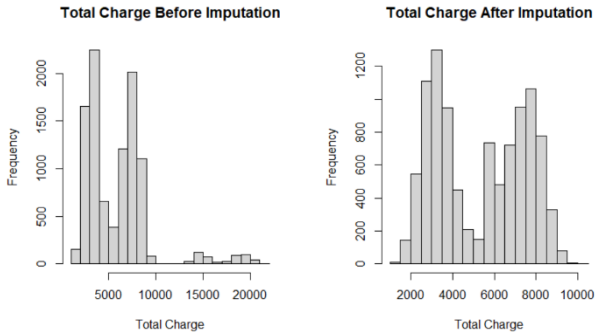




D2: Justification of Mitigation Methods

I did not have any duplicated values so I did not have to do any cleaning for that part. For the missing values, I used the MICE method of imputation. I used this method because I wanted to keep the distributions of each variable relatively similar and MICE will do that for me (*Getting Started With Multiple Imputation in R | University of Virginia Library Research Data Services + Sciences*, 2020). I decided that the VitD level outliers were within a normal a normal range that someone might have

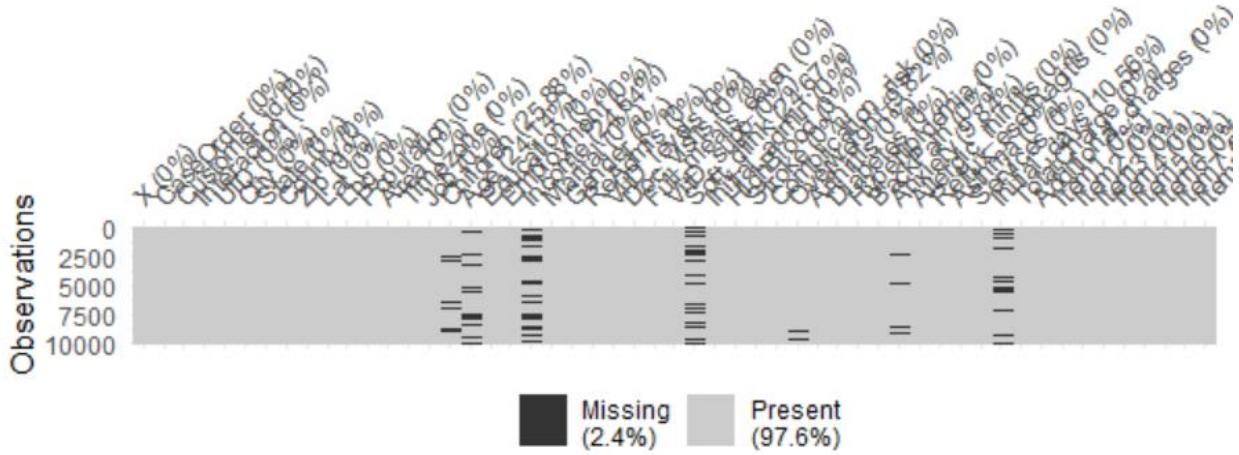
(Tello, 2020). I also kept outliers from the full meals eaten column because it is possible for a few people to have actually eaten more than 3 meals in a day. I also thought the outliers for VitD supplements were reasonable enough to keep. The total charges outliers needed to be changed. There was a big gap in between 11000 and around 13000. This led me to believe that there was probably some sort of error. My guess would be that the error occurred because the total charges column is total charges per day. The outliers were probably values that were total charges overall, and not total charges per day. I decided the best way to fix this was to impute the median where the outliers were. I decided to keep

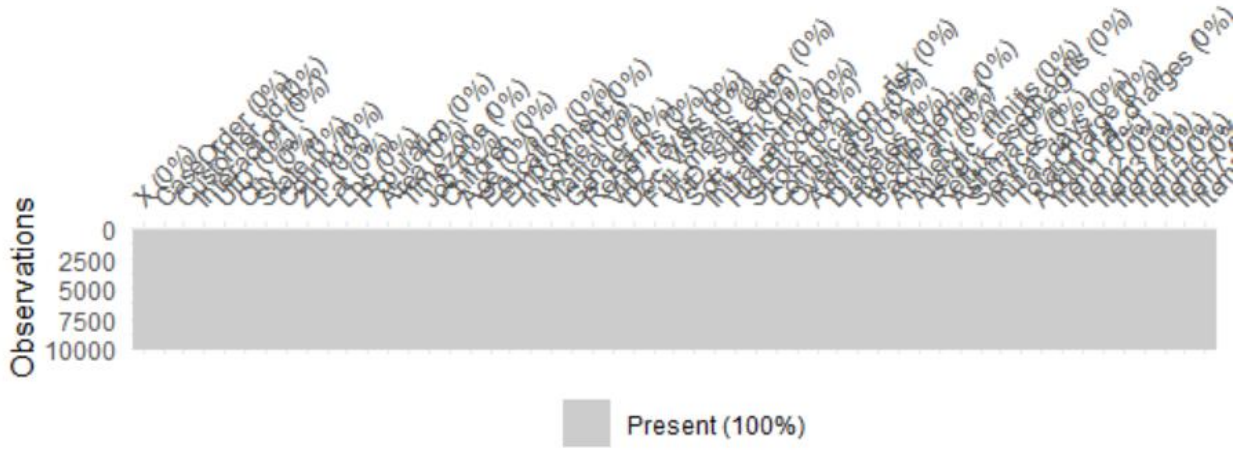


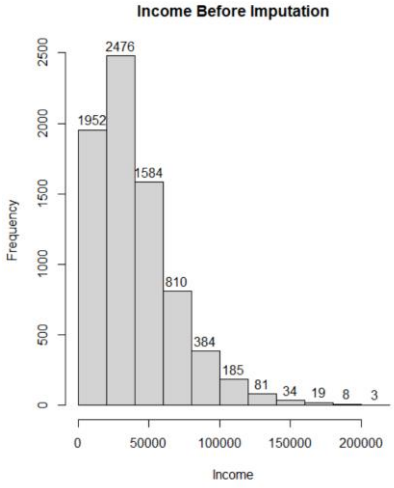
the outliers in the additional charges category. Even though some of them were fairly large, there was no gap in the data and medical expenses can be extremely large, which led me to believe that they are accurate.

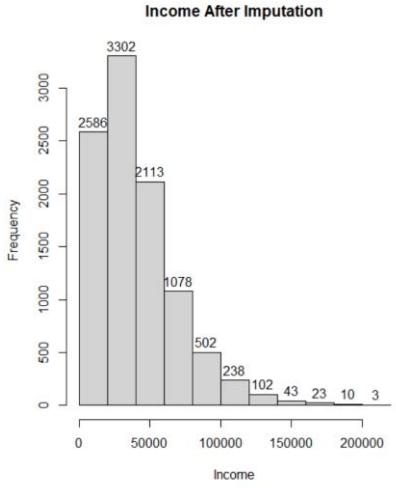
D3: Summary of the Outcomes

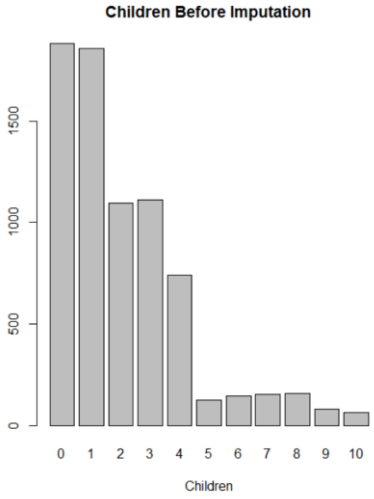
Overall, I ended up using MICE for imputation of missing values. I also used univariate imputation with the median for total charges. Below are before and after visualizations for columns with imputed data.

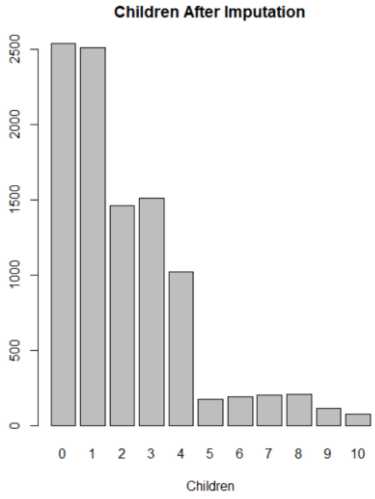

Observations
0
2500
5000
7500
10000
Missing
(2.4%)
Present
(97.6%)

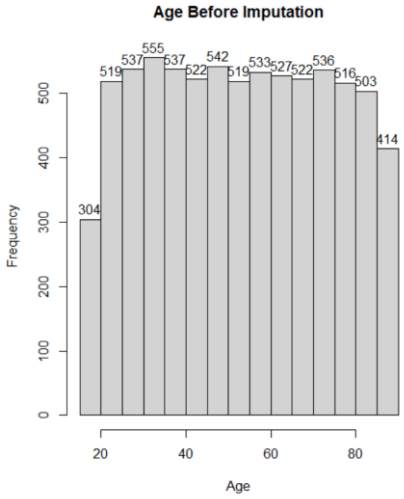

Observations
0
2500
5000
7500
10000
Present (100%)

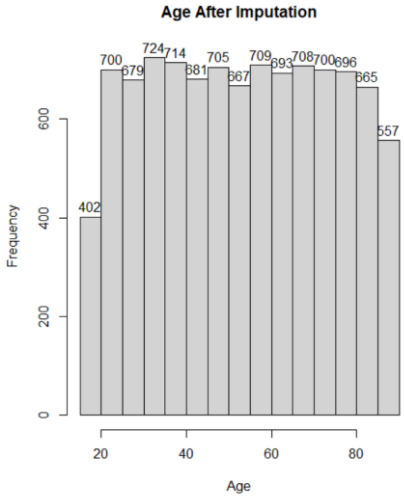

Income Before Imputation
Frequency
0
500
1000
1500
2000
2500
1952
2476
1584
810
384
185
81
34
19
8
3
0
50000
100000
150000
200000
Income

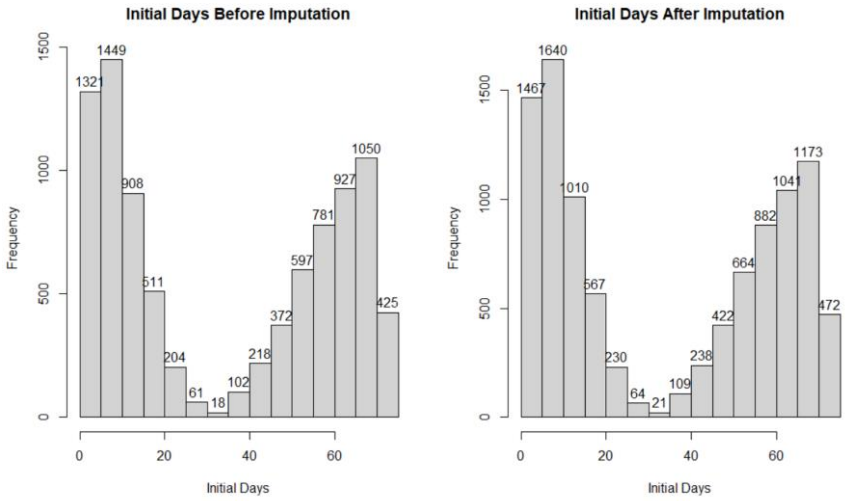

Income After Imputation
Frequency
0
500
1000
1500
2000
2500
3000
2586
3302
2113
1078
502
238
102
43
23
10
3
0
50000
100000
150000
200000
Income

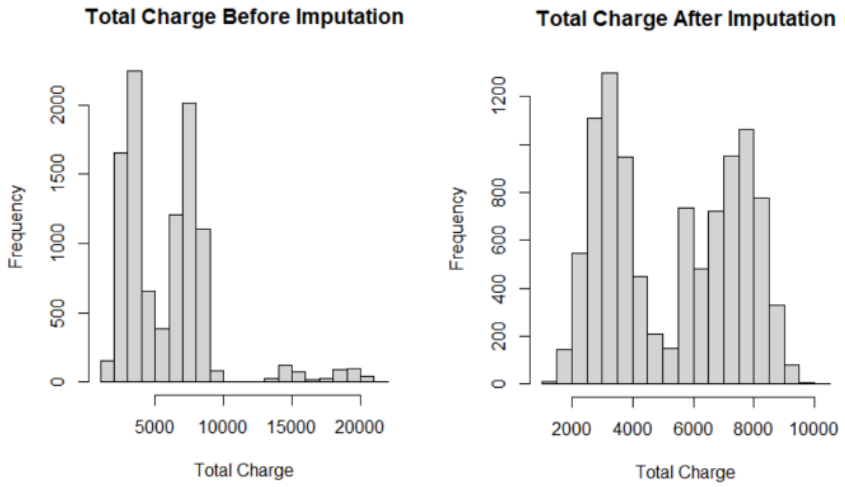

Children Before Imputation
0
500
1000
1500
0 1 2 3 4 5 6 7 8 9 10
Children


Children After Imputation
0
500
1000
1500
2000
2500
0 1 2 3 4 5 6 7 8 9 10
Children


Age Before Imputation
Frequency
0
100
200
300
400
500
304
519
537
555
537
522
542
519
533
527
522
536
516
503
414
20 40 60 80
Age


Age After Imputation
Frequency
0
200
400
600
402
700
679
724
714
681
705
667
709
693
708
700
696
665
557
20 40 60 80
Age





D4: Mitigation Code

Code for mitigation is listed below. Code has also been uploaded as a separate file.

```
#Use MICE to fill in missing values and keep distributions similar
library(mice)
medical$Soft_drink <- as.factor(medical$Soft_drink)
md.pattern(medical,rotate.names = TRUE)
imputed_medical <- mice(medical)
summary(imputed_medical)
imputed_medical$imp$Income
```

```
imputed_medical <- complete(imputed_medical)


#Replace values to the right of the gap in the histogram with the
median TotalCharge
sort(boxplot(imputed_medical$TotalCharge)$out, decreasing=T)
hist(imputed_medical$TotalCharge)
imputed_medical$TotalCharge[imputed_medical$TotalCharge>11000] <-
median(imputed_medical$TotalCharge)
boxplot(imputed_medical$TotalCharge)
hist(imputed_medical$TotalCharge)
```

D5: Clean Data

Clean data set has been submitted as a separate .csv file.

D6: Limitations

I did not have to clean any duplicates. The limitations of using the MICE method of imputation is that the imputations are still just guesses. There is a lot of information in the columns that do have missing data so keeping the same distribution could be accurate, but there is still a chance that the missing data would have impacted the distribution of the data. If that is the case, the MICE method would not be accurate. The disadvantages of boxplots was shown when I tried to use them to find outliers in the total charges column. In the boxplot, the outliers were relatively close to the upper fence. When I looked at a histogram of the same data, there was a clear gap that was not shown in the boxplot. I also used histograms, but that was only to verify outliers from boxplots. As far as replacing outliers, I used the median of the column as a replacement for the outliers. There are definitely disadvantages to imputing the median for outliers. The median is better than the mean because the mean can be influenced by

outliers. Using the median is not the best though because I am only imputed one number. If there are a large amount of missing values, imputing just one number can greatly affect the distribution of the data.

D7: Impact of the Limitations

One challenge a data analyst would face by using my now cleaned data set would be that the values imputed for missing data are just a guesstimate. They are not the real values for the real people we are looking at. Some may be right. Some may be close, but there will definitely be some imputed values that are not close to what the reality is. Another challenge a data analyst may face is with the total charges column. There will be a large amount of values that are the same as the median. While this may not affect the summary of the data, this will likely affect the mode and the distribution of the data.

E1: Principal Components

I decided to use the following variables: population, children, age, income, vitamin D levels, doctor visits, full meals eaten, vitamin D supplements, initial days hospitalized, total charge, and additional charges.

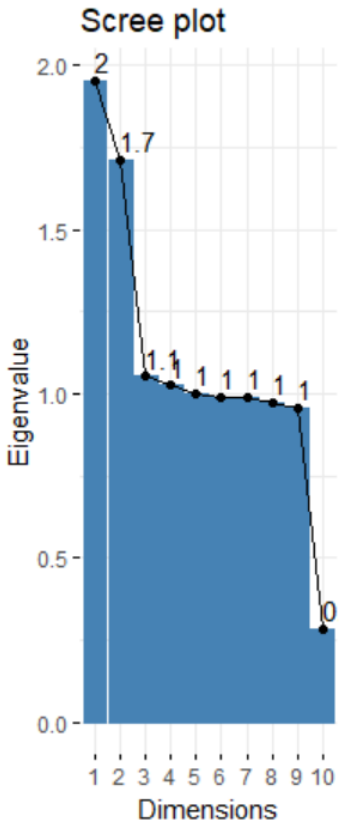
	PC1	PC2	PC3	PC4
Population	0.025621308	-0.02664111	-3.463223e-01	0.460813087
Children	0.024046122	-0.01241976	-3.567508e-02	-0.384514777
Age	0.110495171	0.69716572	2.013593e-05	0.026896407
Income	-0.013796011	-0.02296165	-5.134637e-01	-0.393994727
VitD_levels	0.065564240	0.02747095	-2.327195e-01	-0.517975058
Doc_visits	-0.006759935	0.02238413	-3.224326e-01	0.083457582
Full_meals_eaten	-0.022010060	0.03475919	4.414762e-01	-0.446467438
VitD_supp	0.030599256	0.01286068	-5.097892e-01	-0.090881842
Initial_days	0.693981933	-0.11693822	3.969520e-02	0.026428779
TotalCharge	0.698044713	-0.10307147	2.150795e-02	-0.005139786
Additional_charges	0.108274429	0.69687957	-1.405333e-02	0.016233121

	PC5	PC6	PC7	PC8
Population	0.003480527	-0.359530151	0.59541672	0.028460532
Children	0.563688902	-0.690249179	-0.03129845	-0.182835620
Age	0.017552419	0.002240294	-0.00532412	-0.022351246
Income	-0.111836998	0.171428086	-0.24761843	-0.443917921
VitD_levels	-0.244820661	0.157274796	0.64942575	0.003657675
Doc_visits	-0.641698451	-0.506764382	-0.35021714	0.112590131
Full_meals_eaten	-0.286156079	-0.238501190	0.07172997	0.439857044
VitD_supp	0.338342643	0.162225058	-0.17109211	0.748764729
Initial_days	-0.006175634	-0.004989619	-0.06355694	-0.008546109
TotalCharge	-0.028868308	0.011099601	-0.01498347	-0.004108625
Additional_charges	0.021919979	-0.002854512	-0.00828069	-0.035996725

	PC9	PC10	PC11
Population	-0.426142129	0.0119227800	-0.0012518774
Children	0.146546122	0.0048080697	0.0068381733
Age	0.001146466	0.7067622608	0.0259252168
Income	-0.528583701	0.0268964303	0.0028915161
VitD_levels	0.404173938	-0.0058039070	-0.0515770194
Doc_visits	0.290808819	-0.0061423271	-0.0014799121
Full_meals_eaten	-0.516332607	0.0092633226	0.0005344644
VitD_supp	-0.006097207	-0.0004548712	-0.0015897998
Initial_days	-0.036919803	0.0311585417	-0.7042235849
TotalCharge	-0.008426052	-0.0327319024	0.7065978356
Additional_charges	-0.039091276	-0.7052650259	-0.0373449258

E2: Criteria Used

I believe that PCA1 and PCA2 should be kept. They have by far the two highest eigenvalues out of the 11 PCAs. The variables that are most highly correlated also make sense. PCA1 shows a correlation between initial days hospitalized and total charge. This makes sense because the longer you spend in the hospital, the more serious the condition probably is, and the more you will have to pay. PCA2 also makes sense to me because age and additional charges are positively correlated. I think this is because as you get older, the more likely it is that you will need some sort of miscellaneous procedure, treatment, or medicine.



E3: Benefits

The benefits to PCA are numerous, but the biggest is being able to reduce the dimensionality of the dataset. This helps with machine learning models. One example from my results would be PCA1 which

has a correlation between initial days hospitalized and total charge. By reducing this dimensionality, we can improve the performance of our models. By reducing our dimenstionality, we can also use less resources to run our models, which can save the company money. (Arbel, n.d.)

F: Video

Video has been uploaded in the submission folder.

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=836cff9f-fe00-4c27-b468-af7300f23aed

G: Sources for Third-Party Code

Vis_miss: Visualise a data.frame to display missingness. RDocumentation. (n.d.). Retrieved December 18, 2022, from https://www.rdocumentation.org/packages/visdat/versions/0.5.3/topics/vis_miss

H: Sources

Daniel T. Larose, & Chantal D. Larose. (2019). *Data Science Using Python and R.* Wiley.

Stef Van Buuren. (2018). *Flexible imputation of missing data*. Crc Press, Taylor & Francis Group.

Getting Started with Multiple Imputation in R | University of Virginia Library Research Data Services + Sciences. (2020, May 18). https://data.library.virginia.edu/getting-started-with-multiple-imputation-in-r/

Tello, M., MD. (2020, April 16). *Vitamin D: What's the "right" level?* Harvard Health. https://www.health.harvard.edu/blog/vitamin-d-whats-right-level-2016121910893

Iftach Arbel. (n.d.). *What is Principal Component Analysis (PCA) & How to use it?.* Bigabid. https://www.bigabid.com/what-is-pca-and-how-can-i-use-it/