Demographic Data Analysis Of Physician Fee Schedule On Insurance

By

Hang Yang

Submitted to:

Computer Science Department, University of Illinois at Chicago

For the Degree of Science of Computer Science

Under the Supervision of:
Dr. Plamen Petrov,
Dr. Mark Boris Shapiro,
Dr. Ugo Buy

Graduation Project Report for 2020 Spring

UIN: 6711181689

Abstract

To estimate physician fee for a particular medical procedure denoted by CPT/HCPCT code, this report has raised a hypothesis to compute the estimated physician fee schedule based on weighted insurance fee schedules. Assume that physicians get paid depending on the insurance coverage of their patients, so on average, physicians get paid for a particular procedure that they perform depending on the proportion of patients with each insurance type. The report has included three main insurance types in the United States insurance market: Medicaid (a wide-ranging health insurance program for low-income individuals for all ages), Medicare (a federal government program that provides health care coverage for elders 65+ or disability), and Private Insurance (like UnitedHealth, Anthem). Each of them has different prices for each procedure and the prices are published as fee schedule. To determine a fee schedule that a physician would be willing to accept from a new patient or an uninsured patient, this report calculates the weighted fee based on fee schedules from above three insurance types. The weights are determined as the proportion of the population, such as Medicaid, or Medicare beneficiaries in each state or county. The report mainly specifies two geographic levels of weighted fee schedules: state-level fee schedule and county-level fee schedule. The general state-level fee schedule is based on weights of each insurance type over a particular state, especially to identify fee schedules for counties that do not have Medicaid or Medicare beneficiaries, such as newly Medicaid program expanded counties. The county-level fee schedules are based on weights of each insurance type over certain county population.

Table of Contents

Introduction

The jupyter script mentioned in the project is here in [the GitHub repository script].

Problem introduction

It is normal to go to hospital or visit a physician in a clinic when people do not feel well, especially when humans are experiencing an unexpected hardship during the COVID-19 quarantine. If people go to visit a physician, the medical staff or physician usually will ask them a question: can you please tell me what your medical insurance is? Bad story is that some hospitals or clinics might decline a patient who does not have any medical insurance coverage. That is sad because their condition may get worse if they could not get cured in time. People might ask why hospitals and clinics could be so indifferent to decline to accept new patients. Don't new patients mean profit to hospitals? Let us simply analyze what the profit is to a hospital. Simply speaking, profit is the amount of income that the hospitals earned from patients after accounting for all expenses, debts, additional income streams and operating costs. The cost could so huge that hospitals would like to charge as much as they can under the law to keep it open. Not mention if hospitals want to hire more experts and to invest on frontend research [1]. In that case, what is a reasonable bill to patients for a certain procedure service?

The answer is not easy. "Reasonable" means that the fee for a procedure should cover most of the costs that satisfy the physicians, meanwhile, it also should not drive new patients away by an outstanding bill. The report proposes an idea to get the estimated fee schedule based on the proportion of Medicaid, Medicare and private insurance beneficiaries over the state/county population. In the following report, it specifies two geographic levels of estimated fee schedules: state level and county level. Nation level is ignored because it is helpless for a specific area physician to refer to if the area conducts insurance policies from local state governments. This analysis report involves crawling all states/counties population and insurance enrollment datasets, collecting insurance fee schedule data files, calculating weights (i.e. proportions) of each insurance type and average estimation of final adjusted fee schedules [2].

Assumptions

Because of limitations of accessing confidential information from private providers, we have some assumptions around the private insurance enrollment and fee schedule data.

1. The physician fee schedules of private providers are seen same as Medicare insurance
2. All current proportion in a state/county except Medicaid and Medicare insurance beneficiaries is considered to be covered by private insurance. That is to say, the proportion of private insurance is equal to the difference of 1 subtracted by Medicaid and Medicare rate.

The project also makes below optimizations to make project results more accurate and practical.

3. The actual costs or expenses of each insurance type have little difference from their physician fee schedule data.
4. Select Non-Facility Fee or Max Fee from published fee schedules as our insurance fee because the project supposes that physicians would like to be paid more for their work.

Chapter 1 Input data

- Get population data

United States Census Bureau does census investigation annually and its duty is to publish the demographic statistics of population lives in states and counties. I have searched U.S. census official website and downloaded the all states data table for fiscal year 2018. The reason that I choose statistics data of fiscal year 2018 because data of fiscal year 2019 are still under construction. The downloaded dataset is a CSV file and I have stored the data into dataset **df_pop_alldata_2018**.

- Get State Medicaid enrollment data

Centers for Medicare & Medicaid Services (CMS) is a federal agency within the United States Department of Health and Human Services. It releases and updates state Medicaid and CHIP enrollment data every month. The downloaded dataset is a CSV file and stored as dataset **df_mdcaid_2018Bs** [3].

- Get County Medicaid enrollment data

County Medicaid enrollment data is only published by each state itself. Without loss of generality, the project takes three states (Connecticut, Indiana, New York) as examples. The name convention for this report is that dataset named with "_CT_" is for state Connecticut. "_IN_" is for Indiana and "_NY_" is for New York. The three states' county enrollment data is stored in **df_CT_mdcaid_2018Bc, df_IN_mdcaid_2018Bc, df_NY_mdcaid_2018Bc**. However, **df_CT_mdcaid_2018Bc** is processed separately because it could not be downloaded directly. Connecticut only offers township level Medicaid enrollment.

- Get Medicare enrollment data

CMS publishes monthly Medicare enrollment data as well as Public Use File (PUF) which enables us to take use of the demographic, enrollment information in both state level and county level. It is downloaded as an Excel file and stored in script as **df_mdcare_2018Bsc**.

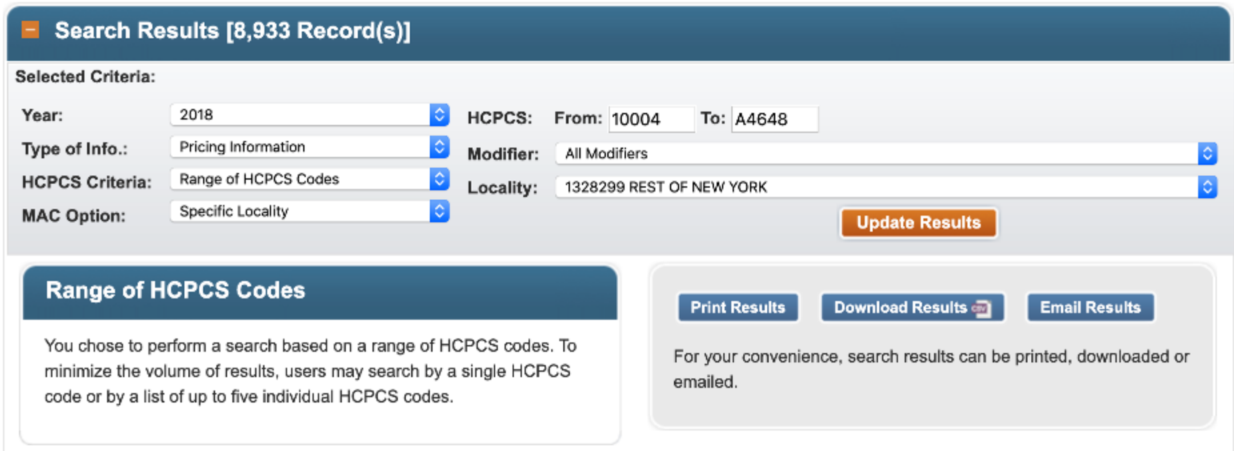
- Get Medicaid Physician Fee Schedule (PFS) data

Medicaid programs are mostly running under state-level governments. Each state could have their own characterized policies, name conventions, and format of the physician fee schedules. In the project, I take three states as samples and they are Connecticut, Indiana, and New York. The rest could be conducted following similar procedures.

5. Connecticut Medicaid PFS is downloaded as CSV file and stored as **df_CT_mdcaid_ASCPFS_2018**.
6. Indiana Medicaid PFS is obtained as excel file and stored as **df_IN_mdcaid_OPFS_2018**.
7. New York Medicaid PFS is obtained as excel file and stored as **df_NY_mdcaid_MPFS_2018**.

- Get Medicare Physician Fee Schedule data

Medicare PFS is organized and managed by Federal, it can only be accessed through a federal managed online search engine. Below is an example of the search engine:



Currently, I have download Medicare PFS reports of Connecticut, Indiana and New York state for fiscal year 2018 and HCPCS range from Medicaid datasets. They are named separately as **df_CT_mdcare_pfs_2018, df_IN_mdcare_pfs_2018, and df_NY_mdcare_pfs_2018**.

Readers can get above datasets except **df_CT_mdcaid_2018Bc** in bold fonts *in our jupyter script cell[2]*.

Chapter 2 EDA

It is noticeable that we have inputs from different sources, like federal bureau, state departments, and private company. How to combine those datasets together is our first task? As the report is studying the state-level and county-level demographic data, the keys to connect different datasets are supposed to geographic information, i.e. state names, county names, state name abbreviations, or county name abbreviations. In order to get the geographic information, let us take a quick look at our input datasets using Exploratory Data Analysis (EDA) techniques.

EDA on population data

The first overview on the population data **df_pop_alldata_2018** is implemented *in the jupyter script cell[3] and cell[4]*. There are totally 3193 entries and 15 columns in this dataset, in which "STNAME" is the state name, "CTYNAME" is the county name" and population estimation from year 2010 to 2018.

Granularity: Each entry in the dataset means either the total population in a state or total population in a county. Rows with "COUNTY" equal to zero mean state counts, otherwise, rows with non-zero values stand for county counts. *It is executed at Jupyter script cell[5]*.

Scope: The data contains both state level and county level record. The report separates those records into two datasets based on the attribute "COUNTY": **df_POP_alldata_2018Bs** and **df_POP_alldata_2018Bc**.

Temporality: The whole sheet contains population estimation from year 2010 – 2018. I only choose population counts which belongs to year 2018 for further research.

Faithfulness: Statistics data types are all integers. They are valuable and computable. CMS has the high quality data which is responsible for all states statistics[4].

EDA on Medicaid enrollment data

- Overview on State-Level enrollment

The Initial view on state Medicaid enrollment data **df_mdcaid_2018Bs** tells us there are 51 entries and 29 columns. *It is executed at jupyter cell[8] and cell[9].* We choose attributes "State Abbreviation", "State Name" and "Total Medicaid and CHIP Enrollment" and have them renamed to "STNAME", "STATE", "MDCAID_CNT_2018BS" for further study.

- Overview on County-Level enrollment

Without loss of generality, the report still uses state Connecticut, Indiana, and New York as examples. To get the county-level Medicaid enrollment data for above three states, I investigate their local health care and insurance program. The datasets has already been stored in the "Input Data" chapter, they are **df_CT_mdcaid_2018Bc, df_IN_mdcaid_2018Bc, and df_NY_mdcaid_2018Bc**.

Connecticut Medicaid county-level enrollment

As specified in the chapter "Input data", we could only load the Connecticut township Medicaid enrollment data **df_CT_DSS_2018Bt** and towns-county **df_CT_Town_County** mapping relationship. Firstly, I filter programs Medicaid and CHIP in DSS table and then sum them up as total Medicaid enrollment data. Secondly, I take use of mapping relationship of towns and counties to insert county geographic information attribute into DSS table. Thirdly, GroupBy object properties and aggregate function sum is used to compute the county-level Medicaid enrollment counts. Finally, I store the county and count information into a new dataset **df_CT_MDcaid_2018Bc**. *It is executed in the jupyter script cell[22-24].*

Indiana Medicaid county-level enrollment

The overview and preprocessing are conducted at jupyter script cell[18-19]. I find out that the index of the dataset is county names connected with an ID and "County Total" is the enrollment data. In order to get our geographic information key – county name, I use "reset_index" function to add the existing index as a new column for the dataset, and then separate the new column "index" as "COUNTY_ID" and "CTYNAME". Finally we generate a new dataset **df_IN_MDcaid_2018Bc** with desired geographic key "CTYNAME" and its corresponding enrollment data.

New York Medicaid county-level enrollment

The overview and preprocessing are conducted at jupyter script cell[25-27]. The file from New York state is organized in a pivot table like below:

	County	Plan Name	Roster Enrollment	Roster Enrollment.1	Roster Enrollment.2	Enrollment	Unnamed: 6
0	Albany	TOTALS:	4152.0	822.0	3405.0	36066.0	44445.0
1	Mandatory	Capital District Physicians Health Plan	2574.0	467.0	2223.0	20249.0	25513.0
2	Eff. Oct 1997	Fidelis Care	1063.0	249.0	834.0	11645.0	13791.0
3	NaN	MVP Health Plan	285.0	50.0	96.0	2184.0	2615.0
4	NaN	United Healthcare Plan of NY	91.0	20.0	74.0	749.0	934.0
5	NaN	Wellcare of New York	139.0	36.0	178.0	1239.0	1592.0
6	Allegany	TOTALS:	730.0	247.0	615.0	6528.0	8120.0
7	Mandatory	Fidelis Care	321.0	122.0	210.0	3340.0	3993.0
8	Eff. Feb 2007	HealthNow	169.0	55.0	222.0	1271.0	1717.0
9	NaN	YourCare Health Plan	240.0	70.0	183.0	1917.0	2410.0

The "County" column contains dirty data, like "Nan", "Mandatory", which are not valid county name information. But we could filter the county names out by value "TOTALS:" in column "Plan Name". The total enrollment data is stored in column "Unamed:6" which has weak readability. Also, all the counts are of quantitative type but floating decimal format is not a good idea to represent the counts. Finally, I convert the counts into numbers and store above information into new dataset **df_NY_MDcaid_2018Bc**.

EDA on Medicare enrollment data

The dataset df_mdcare_2018Bsc is *executed at jupyter cell[11-12]*. It contains 3250 entries and 248 columns. Here are some findings of the dataset:

- 1, The first row means the national total counts of Medicare beneficiaries with "County" equal to "NATIONAL TOTAL". The row with "County" equal to "STATE TOTAL" means state total Medicare enrollment counts. Otherwise, it stands for county Medicare

enrollment counts. For each state, the row statistics starts from state total counts of Medicare beneficiaries following by each county
- 2, Some counties have missing values in total counts of Medicare beneficiaries, denoted by "*". The count attributes are categorical value instead of quantitative values.
- 3, The counts in column "Beneficiaries with Part A and Part B" are the sum of counts of column "FFS Beneficiaries" and "MA Beneficiaries".

To fix the second finding, we first rename the count attributes "Beneficiaries with Part A and Part B", "FFS Beneficiaries", and "MA Beneficiaries" labels as "MDCARE_CNT_2018", "FFS_CNT", "MA_CNT", and then convert them into quantitative values if the value is not "*". The third finding helps us to fill in the missing values in "MDCARE_CNT_2018" by adding up counts in "FFS_CNT" and "MA_CNT". *It is executed at jupyter script cell[13]*.

Based on first finding, the report separates state-level and county-level data by filtering the values in "County and generates two new dataframes named **df_MDcare_2018Bs** and **df_MDcare_2018Bc**. *This is executed at jupyter script cell[15]*.

EDA on PFS data

Both Medicaid and Medicare PFS are specified for each state. We choose **df_IN_mdcaid_OPFS_2018, df_CT_mdcaid_ASCPFS_2018, df_NY_mdcaid_MPFS_2018** as our research data. They have some common properties:

- Each row stands for a fee schedule of a procedure service noted by procedure code/HCPCT code
- To maintain the validity and temporality of PFS data, the report will only include procedure code which is effective in fiscal year 2018.
- I decided to drop rows that do not contain valid price value in the PFS tables, because the CPT codes could have huge gap for different procedures, like heart operation and flus.
- Amount is usually represented in "$123, 45.00" format.

Utilizing the range of CPT code after cleansed and the CMS online Medicare search engine, the Medicare PFS could be accessed and downloaded. To handle Medicare datasets, I create a function called ***handle_medicare_pfs*** *which is shown in the script cell[32].*

Under the first assumption, the report suggest that private insurance fee schedule could be seen as similar as Medicare fee schedule. Commonly speaking, private insurance fee is higher than Medicare fee schedule, and Medicare fee schedule is higher than Medicaid fee schedule for a particular medical procedure [5]. Therefore, the report could include desired columns "MDCAID_PFS_AMT", "MDCARE_PFS_AMT", "PP_PFS_AMT" into uniform PFS datasets **df_CT_PFS_2018, df_IN_PFS_2018, and df_NY_PFS_2018** which could be applied to compute weighted fee schedule later.

Chapter 3 Calculate weights

The goal of the project is to estimate weighted fee schedule for a particular procedure service. The weights are determined by the proportion of each insurance (Medicaid, Medicare, and Private Insurance) over state or county population. Below are the formulas:

Formulas

$$State\ Medicaid\ Rate = \frac{state\ Medicaid\ enrollment\ data}{state\ total\ population}$$

$$State\ Medicare\ Rate = \frac{state\ Medicare\ enrollment\ data}{state\ total\ population}$$

$$State\ Private\ Insurance\ Rate = 1 - \ state\ Medicaid\ Rate - state\ Medicare\ Rate$$

$$County\ Medicaid\ Rate = \frac{county\ Medicaid\ enrollment\ data}{county\ total\ population}$$

$$County\ Medicare\ Rate = \frac{county\ Medicare\ enrollment\ data}{county\ total\ population}$$

$$County\ Private\ Insurance\ Rate = 1 - \ County\ Medicaid\ Rate - County\ Medicare\ Rate$$

Calculate state level weights of each insurance type

In conclusion, state level weights table **df_POPMM_2018Bs** *is executed in the jupyter script cell[45-46].* We know that Utah state has the lowest rate of Medicaid enrollment which is 9.12%. DC has the highest rate of Medicaid is 36.90%. While DC has the lowest Medicare rate is 11.09%, Maine has the highest Medicare rate is 23.48%. New Mexico has the lowest private insurance rate is 47.02%, Utah has the highest private insurance rate is 79.6%. Our private insurance rate is under second assumption that all non-Medicaid and Medicare covered population has bought private insurance.

Calculate county level weights of each insurance type

To compute each state's county level weights, I create a function ***calculate_county_level_weights*** in the script cell[47]. The main logic of the function is to compute the county level weights which is the proportion of each insurance type in the county. To obtain the proportion, it first filters out all population data and Medicare enrollment data of counties in the specific state. After that, it combines county level population dataset with county level Medicaid and Medicare enrollment dataset. Finally, we apply above formula and vector arithmetic operation into calculating weights based on county-level population, Medicaid, and Medicare enrollment data.

To test on our examples, I have computed out the result for Connecticut, Indiana, and New York state. Among them, Connecticut county level weights table **df_CT_MMP_Weights_2018Bc** has 8 counties. Indiana county level weights table **df_IN_MMP_Weights_2018Bc** has 88 counties. New York county level weights table **df_NY_MMP_Weights_2018Bc** has 46 counties. They are *executed from jupyter script cell[47-50].*

Chapter 4 Calculate weighted fee for each procedure code

To calculate weighted fee for each procedure code, I develop two functions: one is called ***compute_estimated_state_pfs***, the other is called ***compute_estimated_county_pfs***. The first function is *defined in jupyter cell[51]* and its main logic is to access the weights in the state weights table by the state abbreviation name and compute the weighted fee schedule. The formula to calculate state level weighted fee is as below:

$$
\begin{aligned}
&Weighted\ State\ Physician\ Fee\ Schedule \\
&\qquad = state\ Medicaid\ rate * Medicaid\ Fee\ Schedule + state\ Medicare\ rate \\
&\qquad * Medicare\ Fee\ Schedule + state\ private\ insurance\ rate \\
&\qquad * private\ insurance\ fee\ schedule
\end{aligned}
$$

For our county level function ***compute_estimated_county_pfs*** whch is defined *in the jupyter script cell[52].* It first creates a county list with all the county names in that state. Following that, it populates each member of the county list as an attribute in the county level physician fee schedule and initialize them as zeros. Then it uses a similar formula to calculate county level weighted fee as below:

$$
\begin{aligned}
&Weighted\ County\ Physician\ Fee\ Schedule \\
&\qquad = county\ Medicaid\ rate * Medicaid\ Fee\ Schedule \\
&\qquad + county\ Medicare\ rate * Medicare\ Fee\ Schedule \\
&\qquad + county\ private\ insurance\ rate * private\ insurance\ fee\ schedule
\end{aligned}
$$

After implementing above two functions, the final weighted physician fee schedules are generated *in the jupyter script cell[54], cell[55], and cell[56]* and the report also exports a copy of the weighted fee schedules which have been uploaded into project GitHub repository.

Conclusion

To solve the problem that how much a physician should get paid for a particular medical procedure service from uninsured patients, the project has assumed that physicians get paid by the insurance coverage of their patients. That is to say, the physician can get paid for a medical procedure based on the weighted insurance fee schedule. To get the weighted It conducts the experiment on collecting different data sources, like population data, different insurance enrollment data, and different insurance physician fee schedule of fiscal year 2018. From the EDA on these datasets, we conclude that these datasets could be unified by the geographic information. Because some states have just expanded their Medicaid program, to give a state level overview of the weighted physician fee, the project calculates weights for insurance programs based on state level enrollment data and its population. However, in this research we also know that many state governments have released their own practitioner fee schedule besides the federal office. It means that state level PFS might not be accurate or practicable enough for a physician to refer to. To solve it, the project brings up the idea to generate county level weighted fee schedules for each state. Without loss of generality, we take Connecticut, Indiana, and New York as experiment states. In the final result datasets, we find out that each state's county level weighted fee schedule is fluctuated from state level weighted fee schedule. And different county's weighted fee schedule is also varying from CPT code to CPT code.

Future work

On the one hand, the report does not include the weighted national fee schedule because it is too general to be applied to specific states, not mentioning specific county or town. On the other hand, there could be some improvements for future research. Firstly, the actual costs of a patient visit for a particular procedure code could be more valuable when physicians take them into consideration for above new patient's billing case. Secondly, it is highly expected that we can cooperate with private insurance companies with their enrollment data and physician fee schedule database. Thirdly, further study could focus on town, Hospital Referral Region (HRR) level or ZIP code-level fee schedules could be quite prospective. The procedures in the report of how to calculate the weighted fee schedule in this report could be applicable for future study.

References

[1] Jordon Rosenfeld, “Billing strategies for cash-pay patients”, *MEDICAL ECONOMICS*, Mar 29 2018, https://www.medicaleconomics.com/revenue-cycle-management/billing-strategies-cash-pay-patients. [Accessed 17 Apr 2020]
[2] Kaitlyn Houseman, “Fee Schedule Dos and Don’ts”, *REVELEMD*, Mar 17, 2016, https://www.revelemd.com/blog/fee-schedule-dos-and-donts. [Accessed 17 Apr 2020]
[3] KFF, “Analysis of Recent Declines in Medicaid and CHIP enrollment”, *KFF*, Nov 25, 2019, https://www.kff.org/medicaid/fact-sheet/analysis-of-recent-declines-in-medicaid-and-chip-enrollment/. [Accessed 23 Apr 2020]
[4] Charles E. Lenoard, et. al., “The quality of Medicaid and Medicare data obtained from CMS and its contractors: implications for pharmacoepidemiology “, *NCBI*, Apr 26, 2017, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5406992/. [Accessed 23 Apr 2020]
[5] Stephen Zuckerman et. al., “Trends in Medical Physician Fees”, *HEALTH AFFAIRS*, 28 April 2009, https://www.healthaffairs.org/doi/full/10.1377/hlthaff.28.3.w510. [Accessed 28 Apr 2020]