Identifying Reimbursement Opportunities for Healthcare Providers and Professionals

Mark Roach
Rishabh Sareen
Alexander Sietsema
Siqiang Wang

April 28, 2022

Abstract Healthcare providers receive reimbursement from insurance companies and the federal government. The goal of this project is to assist The Rybar Group in identifying reimbursement opportunities for their clients by analyzing the DSH patient percentage. This report develops a graphical user interface to automate the generation of summarized cost reports in the future. It also analyzes the trends of DSH payments in similar sized hospitals according to the state they are located in and their categorization as urban or rural hospitals.

Work done in partial fulfilment of the requirements of Michigan State University MTH 844; advised by Mr. Jesse Parker, Mr. Rick Reid, and Ms. Brooke Yowell, The Rybar Group; Dr. David Bramer and Dr. Peiru Wu, Michigan State University.

Table of Contents

Introduction

The Rybar Group (TRG) was founded in 1989 in Michigan with a mission to provide high-quality services to healthcare providers. Focusing exclusively on the healthcare industry, they offer their expertise in improving financial operations of healthcare systems, hospitals, and physicians' practices, whilst also assisting management in ensuring accurate financial reporting. Their team of reimbursement professionals includes CPAs, former hospital CFOs and former Medicare auditors.

The target healthcare market for The Rybar Group is small to medium-sized hospitals. Every hospital is required to file a Medicare cost report annually with the Centers for Medicare & Medicaid Services (CMS). Such cost reports are publicly available on the CMS website and contain a wide variety of demographic and operational data, such as the type of hospital (urban or rural), procedures undertaken, costs charged, and total patient days in hospital. The goal of this project is to assist The Rybar Group in using these cost reports to identify potential hospitals and healthcare agencies that may be able to increase their Medicare reimbursements.

The initial objective of this project is to create a system that can collect and summarize the raw data in a format useful for The Rybar Group. The next objective involves identifying the reimbursement opportunities for the hospitals by analyzing their Disproportionate Share Hospital (DSH) payments. The DSH payments are paid by the state to qualifying hospitals that serve a large number of Medicaid and uninsured individuals. Payment eligibility is measured by the hospital's DSH patient percentage. The hospital will receive federal reimbursement when the DSH percentage is higher than 15%. Projecting hospitals that have DSH percentages near 15% threshold will assist The Rybar Group in finding target clients which need help improving their reports.

This report begins with the Data section which outlines the contents of a sample cost report collected by the CMS. The Analysis section illustrates the calculation of the DSH patient percentage. The Methods section demonstrates the process for generating a summarized cost report, while the Results section discusses the variation in DSH payments across rural and urban hospitals in different states. It is followed by the Discussion section, which is involved with projecting DSH payments for an individual hospital. The Conclusions section summarizes the report while the Recommendations section details areas for potential improvement. The Appendices contain technical details, including in-depth descriptions of the data, code, and modeling techniques.

Data

The data was downloaded from the database of the CMS. Each year, the Medicare-certified institutional providers are required to submit the CMS-2552-10 annual cost report form.

The provided dataset contains cost reports from the 2010 to 2020 fiscal years. The data itself contains the raw data fields found in each cost report separated into three tables: the Report table,

the Numeric table, and the Alpha table. The relationship between the three tables and a sample report is shown in Appendix A.

The Report table covers the general information about each provider, such as the type of report, the time period covered, the location of the hospital, etc. The first column of the Report table is the report record number (RPT_REC_NUM), which is a unique identifier for each hospital that is shared across the Numeric and Alpha tables. This record number will be the key variable for concatenating the tables in the next steps.

The name, description, and example of each variable are included in three tables. In total, there are 26 variables. It includes information such as fiscal year, addresses, vendors, providers, level of medical utilization, etc. A sample column and its description are given below in Table 1.

Table 1. Sample Column Names and Descriptions.

Column Name	Description	Example
ADR_VNDR_CD	Vendor for Fiscal Intermediary	4
ALPHNMRC_ITM_TXT	Provider reported alpha data	2600DRUGS
CLMN_NUM	Valid Column Number defined	1000
FI_CREAT_DT	Date the FI created the HCRIS file	7/15/2021
FI_NUM	Fiscal Intermediary Number in effect at the time of cost report filing	11001
FI_RCPT_DT	Date cost report was received by Fiscal Intermediary	7/14/2021
FY_BGN_DT	Cost Report Fiscal Year beginning date	1/1/2020
FY_END_DT	Cost Report Fiscal Year ending date	12/31/2020
INITL_RPT_SW	Y or N, Y = the first cost report filed for this provider	N

The remaining raw data of the cost report is shared between the Numeric table and the Alpha table. Both tables are in a flattened format, where each row provides the row and column number of the location of a cell from the report, as well as the value of that cell. The Numeric table stores the locations and values of all the cells from the cost report with purely numeric values, while the Alpha table stores alphanumeric cells. There are approximately 3 million rows of data in the Alpha table and 19 million rows of data in the Numeric table across the whole dataset.

Each hospital is assigned a unique provider number by the CMS. This number is used in the cost report to uniquely identify the hospital. It is generally a six-digit number, where the first two numbers classify the state or territory. For state codes 01 to 09, the provider number will be a five-digit number where the first digit will indicate the state code with the 0 disregarded. While all states are assigned a number, additional territories/countries are assigned a code. Also, occasionally states have multiple state codes. Table 2 gives an example of a few CMS state codes that are used. The complete state codes table is shown in Appendix A.

Table 2. Sample CMS state codes.

State	Code	State	Code	State	Code	State	Code
Alabama	01	Iowa	16,76	New Jersey	31	Utah	46
Alaska	02	Kansas	17,70	New Mexico	32	Vermont	47
Arizona	03	Kentucky	18	New York	33	Virgin Islands	48

The last four digits of the provider number indicate the type of health care facilities. able 3 below gives a sample of few facility codes that are used. For instance, if the code is 3152, between 3100 and 3199, this indicates it is a home health agency.

Table 3. Sample CMS health care facility codes.

Facility	Code	Facility	Code
Medical Assistance Facilities	1225-1299	Home Health Agencies	3100-3199
Critical Access Hospitals	1300-1399	Community Mental Health Centers	4600-4799
Hospices	1500-1799	Outpatient Physical Therapy Services	6500-6989

It is then possible to apply both the state code and facility code to identify any 5 or 6-digit provider number. For example, the six-digit provider number 181664 may be broken down into the state code 18 (Kentucky from Table 2) and the facility code 1664 (Hospice from Table 3), which would indicate that the provider is a hospice located in Kentucky.

Analysis

The Disproportionate Share Hospital patient percentage is used as a metric to determine which hospitals are eligible to receive payments from the government. Hospitals whose DSH patient percentage exceeds the 15% threshold can receive a payment adjustment and those with percentages between 12 and 18 are important to The Rybar Group, as their cost report can be altered to receive more reimbursement. It is defined as:

$$DSH\ Patient\ Percent = \frac{Medicaid\ Patient\ Days}{Total\ Patient\ Days} + \frac{Medicare\ SSI\ Days}{Total\ Medicare\ Days}.$$

Total Patient Days means the total number of patients occupying beds in a long-term care facility for all days in the calendar period for which an assessment is being reported and paid. For example, if 2 patients occupy beds for 3 days, the Total Patient Days will be 6 days. The Medicaid Patient days are calculated based on the case that the insurance cost be apportioned to Medicaid. The Total Medicare Patient Days is the sum of three parts: Health Maintenance Organization (HMO), Total Hospital, and Labor and Delivery Days. Medicare SSI days means the patient days that are entitled to the supplemental security income (SSI) benefits [1].

The total charges do not reflect the payment received by the hospital, and do not reflect the actual cost to the hospital of providing patient care. Instead, hospitals typically compare their total charges to their cost using a cost-to-charge ratio determination.

The cost-to-charge ratio is the ratio between a hospital's expenses and what hospital charge to the patients. The closer the cost-to-charge ratio is to 1, the less difference there is between the actual costs incurred and the hospital's charges. Multiplying each hospital's overall cost-to-charge ratio by total charges provides an estimate of the hospital's costs. The cost-to-charge ratio can be used to estimate the cost of some specific procedures or to compare hospital costs between different facilities in the same local area or in other areas of the country.

Table 4. Comparison of cost-to-charge Ratio for Hospital A and Hospital B.

	Hospital A	**Hospital B**
Number of Knee Replacements per month	20	10
Total Charges	$800,000	$500,000
Average Charge Per procedure	$40,000	$50,000
Hospital Cost to Charge Ratio	40%	35%
Estimated Cost Per Procedure	$20,000	$17,500

The example in Table 4 compares the costs of two hospitals providing knee replacements located in the same city. Based on average charges per procedure, Hospital B appears more expensive for knee replacements. Hospital B's lower cost-to-charge ratio, however, means that it performed each of the hip replacements at a lower average estimated cost than Hospital A.

Having an estimate of DSH patient percentage over the next couple of years can help The Rybar Group to accurately file the Medicaid reimbursement claims for their clients. In the given dataset, the DSH percentage in the current year is not related to the previous year hence a linear regression model is used to extrapolate the percentage. Thus, when the residuals are distributed randomly and not influenced by the residuals in previous year, linear regression gives a good estimation. It essentially models the relationship between two variables by fitting a linear equation to the observed data.

The formula for the model is given by $y_i = \alpha + \beta x_i + \varepsilon_i$, where the dependent variable y_i represents the DSH patient percentage, the independent x_i represents the year (ranging from 2010 to 2020), and ε_i represents error by year.

The regression line is fitted to the data by minimizing the sum of the squares of the residuals. The visualization of linear regression is shown in the Results section.

Methods

This section explains the process followed to extract certain columns and rows from the Report table, Numeric table, and Alpha table for each year and combine them into a summary report that can provide direct insight into DSH payments.

The Report table, the Numeric table, and the Alpha table are related to each other using the same report record number. The first task was to create a process that could convert the raw data into a single summarized table for both internal and external use. In particular, the rows of the summarized data correspond to individual and summary fields specified by the company for each hospital in the dataset.

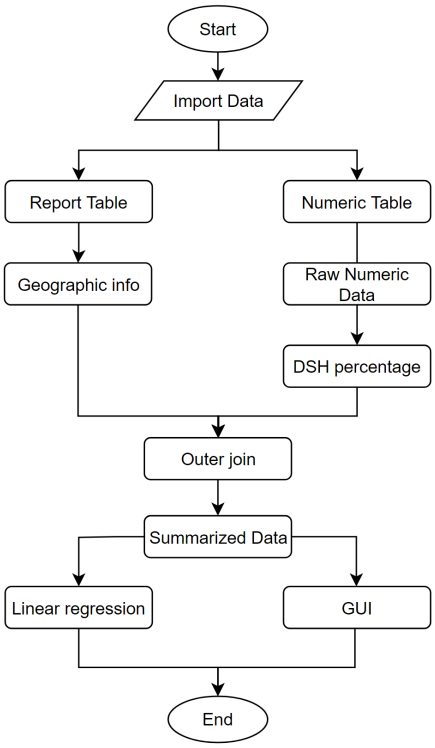


Figure 1. Workflow chart describing the merging and slicing of columns across different files.

The workflow chart of the data processing steps is shown in Figure 1. There are two tables used in the first process after importing the data to programming software. The report table contains the report number and provider number uniquely for each report. The report table also includes geographic information like which state a hospital is located in and whether it is in an urban or rural area. The numeric table has the raw fields used to calculate the Medicaid utilization, HMO, and DSH patient percentage as described in the Analysis section.

The summary table is created by joining these two types of information together for each hospital. When performing the outer join, missing values in the numeric table are assigned to be zero. The data was then used for linear regression analysis.

Results

This section illustrates the correspondence of DSH payments with the location of hospitals across different states and their categorization as urban or rural hospitals. It also looks at applying the regression model to the yearly average to estimate the 2021 average DSH patient percentage.

After merging the columns using the process outlined in the previous section, the two ratios MED_UTIL and SSI_PER are calculated based on the division formula in the analysis section and are demonstrated in Table 5. The DSH patient percentage is then the sum of the Medicaid utilization ratio and the SSI percentage.

Table 5. Example of calculating DSH Patient Percentage for 2019 fiscal year.

RPT_REC_NUM	HMO	TOT_HOSP	LAB_DEL_DAYS	TOT_HOSP	MED_UTIL	SSI_PER	DSH_PAT_PER
649071	nan	nan	nan	481	nan	0.1529	nan
649407	1056	2965	50	31535	0.1291	0.0268	0.1559
649741	854	nan	18	6287	0.1387	0.0434	0.1821
650373	18	46	7	707	0.1004	0.0465	0.1469
650435	nan	nan	nan	9738	nan	0.0942	nan
650477	262	nan	10	1701	0.1599	0.0765	0.2364
650826	180	nan	2	1129	0.1612	nan	nan
651192	557	1434	22	1329	0.1514	0.0209	0.1723
651366	nan	nan	nan	137	nan	nan	nan

Table 5 above gives a few example rows of the DSH patient percentage calculation for the 2019 fiscal year. The ‘nan’ values refer to instances in the cost report where no value was given.

The ‘nan’ values are assumed to be circumstances when the healthcare provider did not enter a value for the relevant part of the cost. These 'nan’ values are thus assumed to be 0 and are replaced in the data as such.

Figure 2 below demonstrates the generated histogram of the DSH patient percentage values for all the providers in that fiscal year. While the general cut-off point for claiming DSH reimbursement is 15%, for healthcare providers close to that value (12% or above), the Rybar Group can consult with the provider to find ways to increase that percentage.

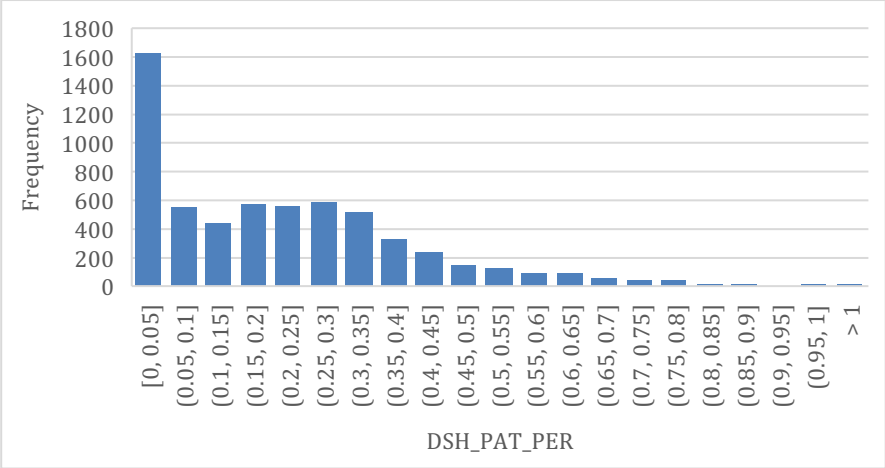


Figure 2. DSH Patient Percentage Histogram for HOSP10FY2019.

Figure 3 shows the average DSH patient percentage in the United States by year 2010 to 2020 (11 years). Fiscal year 2021 is not included since available data remains limited.

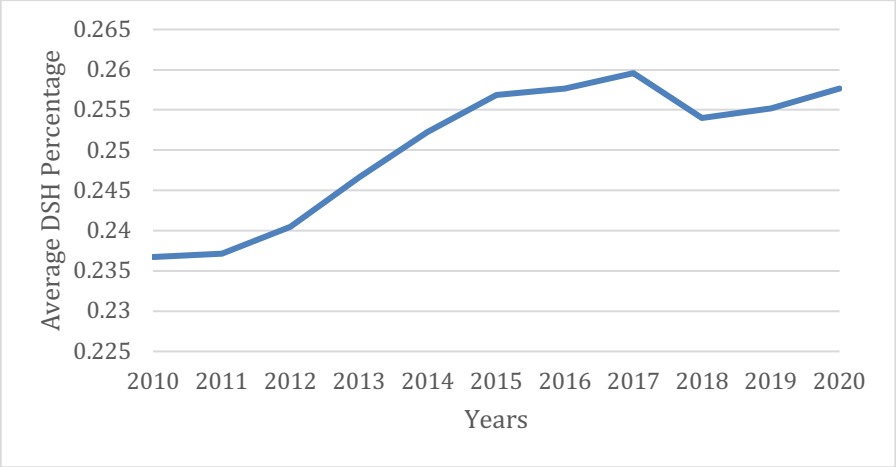


Figure 3. The Average DSH Patient Percentage in United States.

The slope of the curve demonstrates that there was a significant growth in average DSH percentage between 2011 and 2015. This is due to the expiration of the American Recovery and Reinvestment Act. From fiscal year 2009 to fiscal year 2011, there was increased federal funding available for Medicaid. The loss of this funding caused reduced enrollment in Medicaid [2]. The general upward trend observed is due to the aging of the baby boom population who have begun to be enrolled into Medicare.

In the United States, urban and rural Hospitals are primarily distinguished based on the following criterion. The first of which is the geographical designation. The U.S. Census Bureau provides the labels of urban or rural. Compared to urban hospitals, rural hospitals are much smaller

in terms of staff numbers, available beds, etc. Another criterion is the official status as rural referral centers [3].

Rural hospitals, although about equal in number nationally with urban hospitals, differ markedly from urban hospitals in their characteristics and utilization. They are much smaller, less than one-third the size of urban hospitals in average bed size, and they have about one-fourth the total number of Medicare discharges. About 32 states in the US have a higher proportion of rural hospitals than urban hospitals. Figure 4 below shows the average DSH patient percentage in the rural and urban hospitals of the United States from 2010 to 2020.

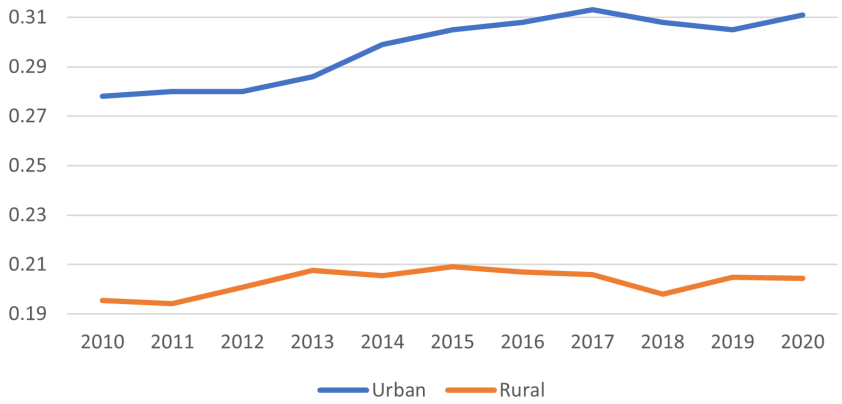
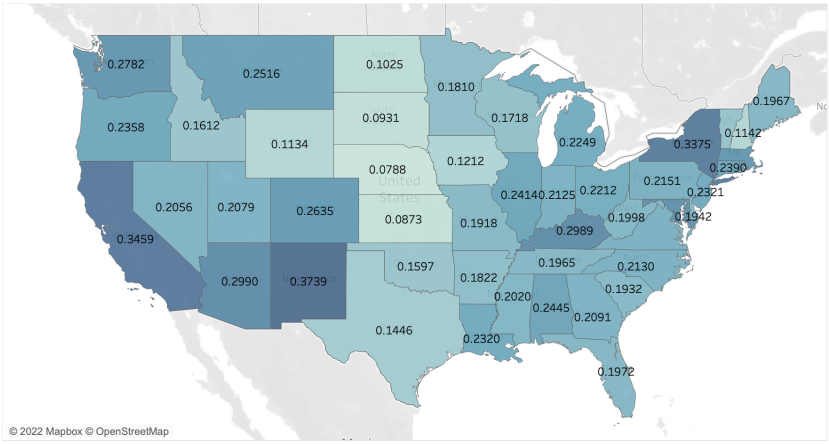


Figure 4. The Average DSH Patient Percentage of rural and urban hospitals in the US.

It is evident from Figure 4 that DSH patient percentage change in rural hospitals is around 1.5% while that of urban hospitals is above 3%. This indicates that although the Medicare Prescription Drug, Improvement and Modernization Act (MMA), 2003, which was enacted to address the inequity in hospital payments, has not been as successful. The MMA established a 12 percent (of total Prospective Payment System) cap on DSH payments for urban hospitals with up to 100 beds and all rural hospitals with up to 500 beds, except for rural referral centers.

Thus, rural hospitals with 100 to 500 beds are treated differently than their urban counterparts (the latter are eligible for DSH payments with no cap). The MMA resolved some equity issues in DSH payment by applying the same percentage formulas across urban and rural hospitals (same reference) but created an inequity by applying a cap differently based on urban-rural status.

According to the North Carolina Rural Health Research Program at the University of North Carolina [4], 161 rural hospitals have shut their doors since 2005. This is due in part to the lack of Medicaid eligibility in certain states.



Map based on Longitude (generated) and Latitude (generated). Color shows average of Dsh Pat Per. The marks are labeled by average of Dsh Pat Per. Details are shown for States. The view is filtered on States and Latitude (generated). The States filter keeps 55 of 55 members. The Latitude (generated) filter keeps non-Null values only.

Figure 5. The Average DSH Patient Percentage per state in the US in 2019.

Figure 5 above gives the average DSH patient percentage per state. California, New Mexico, and New York have the highest average DSH patient percentage whereas states such as Nebraska, Kansas, and South Dakota have the lowest average DSH patient Percentage.

Under the Affordable Care Act (ACA) passed in 2010, states were required by law to expand Medicaid, offering eligibility to any adult earning up to 138% of the federal poverty line. In 2012, the supreme court upheld ACA, but the Medicaid expansion requirement was changed so that it was optional for states. Figure 6 below identifies whether each state had expanded by 2019 [5].

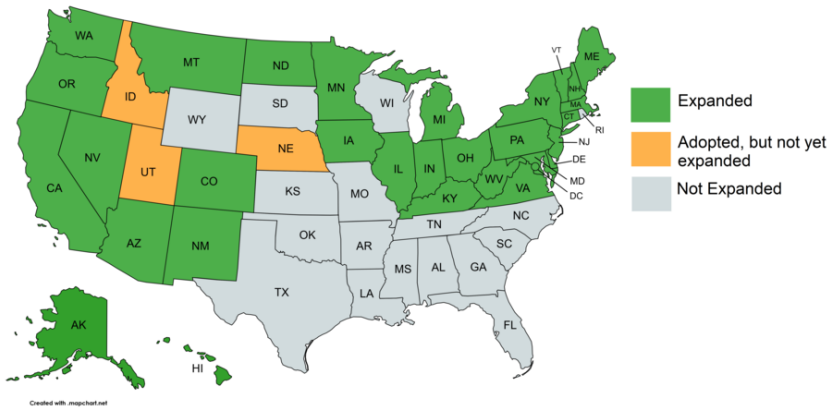


Figure 6. Map of states which had expanded Medicaid in 2019.

Figure 6 above shows that there is a correlation between the states which had chosen to expand the Medicaid program and those that have a higher average DSH Patient Percentage.

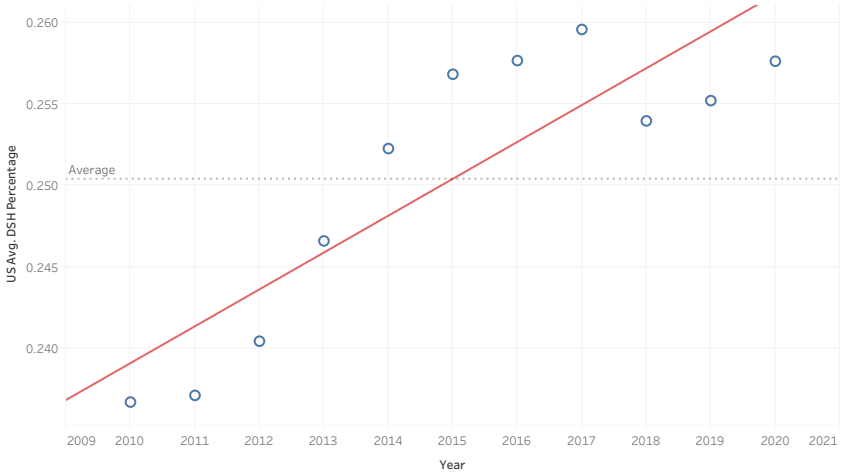


Figure 7. Scatter plot and regression line of DSH percentage.

Linear regression was performed to the DSH patient percentage. Figure 7 shows the fit linear regression line, given by the equation $\hat{y} = 0.0022597 \cdot x - 4.3029$. The predicted value of DSH patient percentage of 2021 is 26.4%. The error in fit seems to be somewhat periodic by year, which may imply that a different model may do a better job of representing the data. Access to more years of data would significantly clarify these potential correlations.

Discussion

While trends of the DSH patient percentage can be considered in aggregate across all hospitals, making meaningful predictions for individual hospitals is not possible. From 2010 to 2020, the mean change in DSH patient percentage in consecutive years is effectively zero.

This implies that, for a given hospital, the best prediction for the next year's DSH percentage is just the current percentage. Of course, this assumes that all hospitals are functionally identical, which is certainly not true. Including other information in a predictive model may be able to uncover more useful trends.

However, there are interesting insights to be found in the aggregate change in DSH percentage between consecutive years. Figure 8 above shows a histogram of these values for all hospitals across the first four years present in the data. Surprisingly, the values are not normally distributed. Future work may be able to predict the overall distribution of DSH percentage change, even if doing so for individual hospitals is impossible.

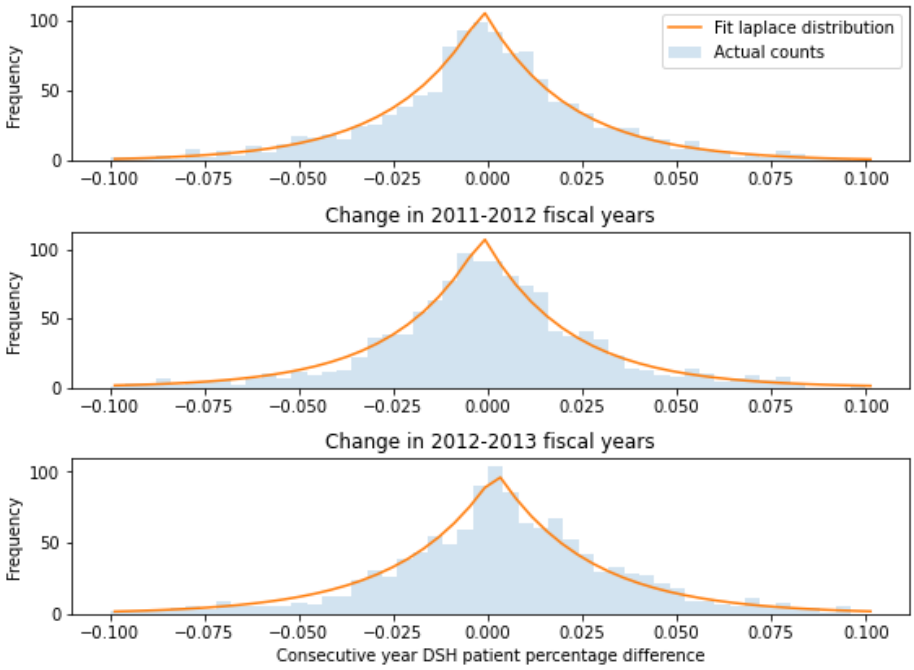


Figure 8. DSH patient percentage differences for consecutive years.

This may allow for general predictions of, for example, what percent of hospitals should expect a DSH increase in future years. A technical description of this phenomenon from a probabilistic perspective and a definition of the Laplace distribution is provided in Appendix C.

Conclusions

Based on the results, the following conclusions are supported:

- A GUI is an effective tool to perform data summarization tasks without the need for technical expertise.
- The regression model predicts that the average DSH percentage in 2021 will be 26.4% indicating that a larger proportion of Medicaid funds are diverted towards DSH payments.
- The states that did not adopt Medicaid expansion had the percentage below the national average level, which was 25.52%.
- The urban hospitals show a 10% higher DSH patient percentage than the rural hospitals.

Recommendations

The team has the following recommendations for The Rybar Group:

- The hospitals in rural areas should be supported to meet the 15% threshold of receiving reimbursements.
- Tools from financial mathematics may be useful for the problem of projection of DSH percentages, particularly by interpreting the DSH patient percentage as a random variable.
- Analyze how the distribution of low-income patient burden is shared by the hospitals within a market area and whether it varies by type of payer and type of service.
- Machine learning algorithms like support vector machine (SVM), artificial neural network (ANN), random forest (RF), can be used to analyze the demand on various branches of medicine like cardiology in hospitals across different states.
- Time series analysis can be utilized to determine if it would be beneficial for some hospitals to expand their outpatient care delivery.

Acknowledgements

Parker, Jesse	Mr. Jesse Parker is the Director of Reimbursement at The Rybar group. We would like to thank him for preparing the data for the project and for his approachability and flexibility. He was always available to answer all our questions and provided a deep insight into the world of medical reimbursement.
Reid, Rick	Mr. Rick Reid is the CEO of The Rybar Group. We would like to express our sincere gratitude towards him for believing in our ability and giving us this opportunity to work on the project. We would also like to thank him for providing us with his valuable support.
Yowell, Brooke	Ms. Brooke Yowell is a Medicaid Reimbursement Manager at The Rybar Group. We would like to thank her for helping us gather most of the subject matter information and clearly defining the desired outcomes of the project.
Bramer, David	Dr. David Bramer is a professor in the Department of Mathematics at Michigan State University. We would like to thank him for his constant advice and encouragement which helped us develop a coherence between different ideas and approaches to solve the problem statement.
Wu, Peiru	Dr. Peiru Wu is a professor in the Department of Mathematics at Michigan State University and the Director of the Professional Science Master's Program in Industrial Mathematics. We would like to sincerely express our heartfelt gratitude and appreciation to Dr. Wu for providing us with her expert guidance and counsel at each step of the project. Her relentless feedback and invaluable suggestions helped us understand the intricate details of the project.

References

[1] *Neuhausen K, Davis AC, Needleman J, Brook RH, Zingmond D, Roby DH* "Disproportionate-share hospital payment reductions may threaten the financial stability of safety-net hospitals." 2014

[2] *Michal Sparer* "Medicaid And The Limits of State Health Reform" 2010

[3] *Hatten JM, Connerton RE*. Urban and rural hospitals: how do they differ?. Health Care Financ Rev. 1986;8(2):77-85.

[4] *C Alfero* "Evaluating the change of policy changes" Rural Policy Research Institute (RUPRI) February 2018.

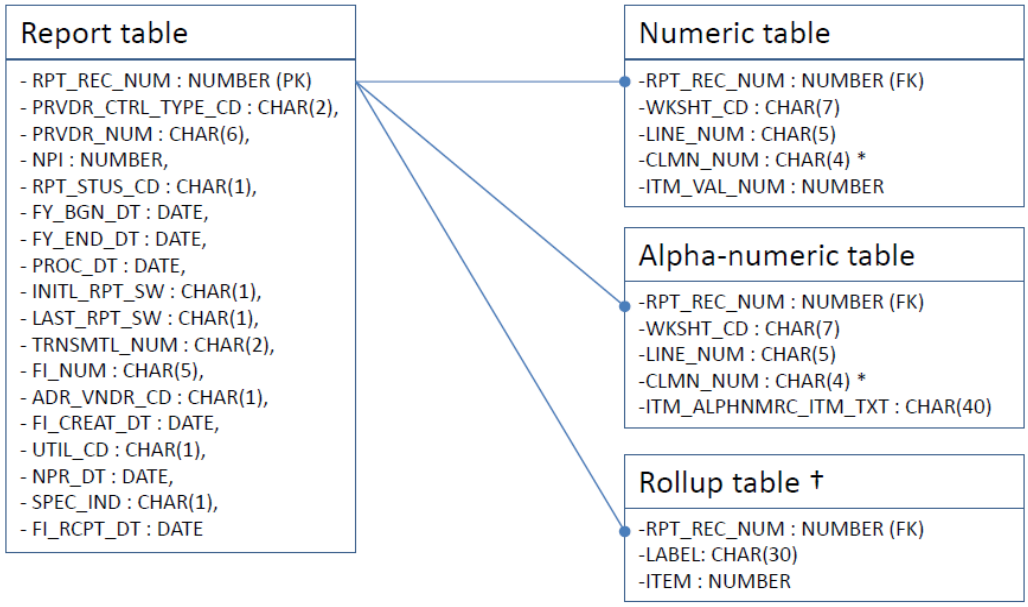
[5] *Chris Lee* "Kaiser finds Medicaid spending is down." October 2021. https://ccf.georgetown.edu/2012/10/30/kaiser-survey-finds-medicaid-spending-is-down/

Appendix A

This appendix describes the relation and correspondence of various columns of the given raw data to the fields of a printed cost report.

For each healthcare institution, there are four tables for each year. They are RPT(report), ALPHA(alpha), NMRC(numeric), and ROLLUP table. The ALPHA contains all entries into the cost report forms that are alphanumeric, while NMRC contains only the numeric values from the forms. The RPT data is identified by a unique report record number which is used to link to all the individual values in the ALPHA and NMRC files. The ROLLUP table is not used in this project.

Table A1. The breakdown of the report data into three separate tables.



As shown in Figure A1, the report data is further broken down into three separate files, the Numeric Table (NMRC), the Alpha-Numeric Table (ALPHA), and the Rollup Table (ROLLUP or RLP). There is one common column in all the tables, which is RPT_REC_NUM. It is the Report Record Number, a cost report specific number assigned by HCRIS.

An example was analyzed to demonstrate how the data given in the Report table, Alpha table, and Numeric table relate to the original cost report submitted by the health care provider.

Table A2. A demonstration of how the numerical tables link to the submitted cost report.

Cost Report

Health Financial Systems | PEARL RIVER COUNTY HOSPITAL | In Lieu of Form CMS-2552-10

ADJUSTMENTS TO EXPENSES | Provider CCN: 25-1333 | Period: From 10/01/2019 To 01/31/2020 | Worksheet A-8 | Date/Time Prepared: 10/15/2020 8:13 pm

	Cost Center Description	Basis/Code (2)	Amount	Expense Classification on Worksheet A To/From Which the Amount is to be Adjusted: Cost Center	Line #	Wkst. A-7 Ref.	
		1.00	2.00	3.00	4.00	5.00	
34.00	INTEREST PAID BY COUNTY	A	37,791	CAP REL COSTS-BLDG & FIXT	1.00	11	34.00
35.00	MEDICAID NH BED ASSESSMENT	A	-4,608	OTHER LONG TERM CARE	46.00	0	35.00
36.00	LOBBYING COSTS - DUES	A	-612	ADMINISTRATIVE & GE	5.01	14	36.00
37.00	MHA EXPENSE	A	4,196	ADMINISTRATIVE & GE	5.01	0	37.00
38.00	IOP TRANSPORTATION & MEALS	A	-504	IOP	76.00	0	38.00
39.00	LATE FEES	A	-100	ADMINISTRATIVE & GE	5.01	0	39.00
40.00	LATE FEES	A	-135	CAP REL COSTS-MVBLE EQUIP	2.00	11	40.00
40.01	PART B BENEFITS	A	-21,150	EMPLOYEE BENEFITS DEPARTMENT	4.00	0	40.01
50.00	TOTAL (sum of lines 1 thru 49) (Transfer to Worksheet A, column 6, line 200.)		-159,685				50.00

Report Table

673097	9	251333	1	10/01/2019	01/31/2020	08/31/2020	N	N	N	7001	4	08/31/2020	F	08/21/2020

Numeric Table

673097	A800000	3400	200	37791
673097	A800000	3400	400	1
673097	A800000	3400	500	11
673097	A800000	3500	200	-4608
673097	A800000	3500	400	46
673097	A800000	3600	200	-612
673097	A800000	3600	400	5.01
673097	A800000	3600	500	14
673097	A800000	3700	200	4196
673097	A800000	3700	400	5.01
673097	A800000	3800	200	-504
673097	A800000	3800	400	76
673097	A800000	3900	200	-100
673097	A800000	3900	400	5.01
673097	A800000	4000	200	-135
673097	A800000	4000	400	2
673097	A800000	4000	500	11
673097	A800000	4001	200	-21150
673097	A800000	4001	400	4
673097	A800000	5000	200	-159685

Alpha-numeric Table

673097	A800000	3400	0	INTEREST PAID BY COUNTY
673097	A800000	3400	100	A
673097	A800000	3400	300	CAP REL COSTS-BLDG & FIXT
673097	A800000	3500	0	MEDICAID NH BED ASSESSMENT
673097	A800000	3500	100	A
673097	A800000	3500	300	OTHER LONG TERM CARE
673097	A800000	3600	0	LOBBYING COSTS - DUES
673097	A800000	3600	100	A
673097	A800000	3600	300	ADMINISTRATIVE & GE
673097	A800000	3700	0	MHA EXPENSE
673097	A800000	3700	100	A
673097	A800000	3700	300	ADMINISTRATIVE & GE
673097	A800000	3800	0	IOP TRANSPORTATION & MEALS
673097	A800000	3800	100	A
673097	A800000	3800	300	IOP
673097	A800000	3900	0	LATE FEES
673097	A800000	3900	100	A
673097	A800000	3900	300	ADMINISTRATIVE & GE
673097	A800000	4000	0	LATE FEES
673097	A800000	4000	100	A
673097	A800000	4000	300	CAP REL COSTS-MVBLE EQUIP
673097	A800000	4001	0	PART B BENEFITS
673097	A800000	4001	100	A
673097	A800000	4001	300	EMPLOYEE BENEFITS DEPARTMENT

Table A2, shows a sample cost report and selected rows from the numeric tables. The first column in the Numeric and Alpha-numeric tables lists the Report Number and the Report Table links the Report Number with the Provider Number and Fiscal Year Dates which uniquely identifies the cost report. The second column lists the worksheet code, which identifies the pages of the cost report that the given data comes from. The third and fourth columns identify line and column numbers which relate to the row and column numbers in the cost report. The last column provides the data which is given at that location.

Table A3. CMS state codes.

State	Code	State	Code	State	Code	State	Code
Alabama	01	Iowa	16,76	New Jersey	31	Utah	46
Alaska	02	Kansas	17,70	New Mexico	32	Vermont	47
Arizona	03	Kentucky	18	New York	33	Virgin Islands	48
Arkansas	04	Louisiana	19,71	North Carolina	34	Virginia	49
California	05, 55	Maine	20	North Dakota	35	Washington	50
Colorado	06	Maryland	21,80	Ohio	36,72	West Virginia	51
Connecticut	07	Massachusetts	22	Oklahoma	37	Wisconsin	52
Delaware	08	Michigan	23	Oregon	38	Wyoming	53
Colombia	09	Minnesota	24,77	Pennsylvania	39,73	Canada	56
Florida	10, 68	Mississippi	25	Puerto Rico	40	Mexico	59
Georgia	11	Missouri	26	Rhode Island	41	American Samoa	64
Hawaii	12	Montana	27	South Carolina	42	Guam	65
Idaho	13	Nebraska	28	South Dakota	43	Northern Marinas	66
Illinois	14,78	Nevada	29	Tennessee	44		
Indiana	15	New Hampshire	30	Texas	45,67		

All the states and regions with their codes are listed in Table A3. The table includes the 50 states, some US territories, and some border countries.

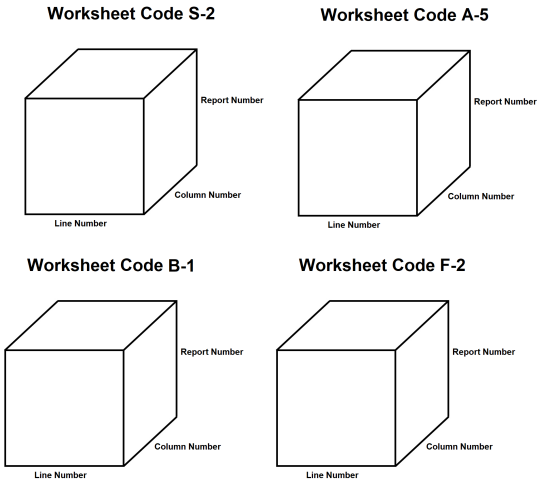


Figure A1. Figurative diagram of data storage.

Figure A1 illustrates how the data is being stored after merging. When a worksheet code is selected, a vertical slice can be generated, either by length or by width, which will produce all the values given for that specific row or column in the worksheet for all the reports for that year. This will generate a matrix which allows for easier analysis.

Rows in the two tables are combined using an outer join using the RPT_REC_NUM as an index. Unlike the normal merge operation, the outer join returns the unmatched rows in one or both tables.

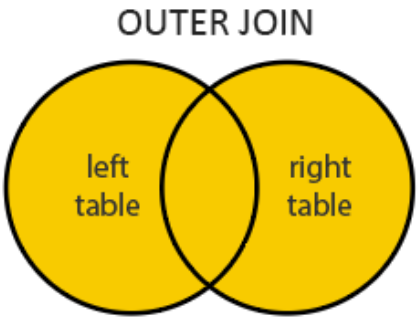


Figure A2. Venn diagram of Outer join.

Figure A2 provides a Venn diagram demonstrating that the outer join returns the value that is either in the left or the right table.

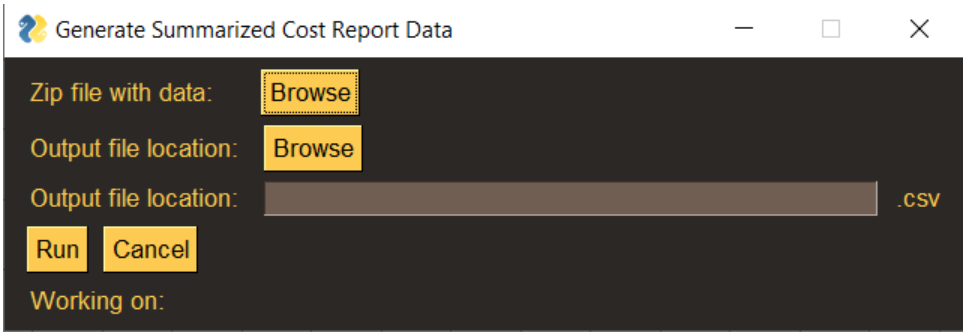


Figure A3. Image of GUI window.

For practical use, the GUI depicted in Figure A3 was developed to allow for creation of the summary table without the need to run functions directly in a Python console. Using the PyInstaller package, this GUI and supporting functions could be wrapped into a standalone executable file for use completely independent of Python.

Appendix B

This Appendix contains the source code used to reformat and summarize the data using the techniques described in the report. This code relies heavily on the Pandas module for data loading and manipulation. For implementation details, please refer to the comments in the source code.

Table B1. Code developed to open, extract, and load compressed data files.

```
import pandas as pd
from zipfile import ZipFile
import PySimpleGUI as sg
#May need to install!
import time
import os
import matplotlib.pyplot as plt
import seaborn as sns

def open_zip(zipname, full=False):
    """Extracts a given zip file and inner merges the contents into a dataframe."""
    if full:
        # If full file path (C:/...), use that path directly. Otherwise (if
        # being run in same directory as "MSU - HCRIS Project",
        # use the default path)
        zipfile = zipname
    else:
        zipfile = f"MSU - HCRIS Project/{zipname}.ZIP"

    with ZipFile(zipfile, "r") as zipf:
        # Generates the names of the contents of the zip file from the given
        # name of the zip file.
        fname = zipfile[zipfile.rfind("/")+1:-4].replace("FY", "_")
        # Opens the CSV files as Pandas dataframes and gives the columns the
        # appropriate names.
        alpha = pd.read_csv(zipf.open(f"{fname}_ALPHA.CSV"),
names=["RPT_REC_NUM","WKSHT_CD","LINE_NUM","CLMN_NUM","ALPHNMRC_ITM_TXT"])

        nmrc = pd.read_csv(zipf.open(f"{fname}_NMRC.CSV"), names=
                      ["RPT_REC_NUM","WKSHT_CD","LINE_NUM","CLMN_NUM","ITM_VAL_NUM"])
        rpt = pd.read_csv(zipf.open(f"{fname}_RPT.CSV"), names=
                          ["RPT_REC_NUM","PRVDR_CTRL_TYPE_CD","PRVDR_NUM","NPI",
                           "RPT_STUS_CD","FY_BGN_DT","FY_END_DT","PROC_DT",
                           "INITL_RPT_SW","LAST_RPT_SW","TRNSMTL_NUM","FI_NUM",
                           "ADR_VNDR_CD","FI_CREAT_DT","UTIL_CD","NPR_DT",
                           "SPEC_IND","FI_RCPT_DT"])
        return rpt, alpha, nmrc
```

As shown in Table B1, a function was defined to extract the zip file containing the data, and returns three raw data tables discussed in Appendix A.

After the alpha, numeric, and report tables are extracted, the next step is to merge the tables and perform the data analysis.

Table B2. Code developed to merge, slice, and display data files.

```
def reformat(table):
    """Reformats a dataframe to drop the worksheet code column and unflatten
    the rows of the table."""
    table = table.drop(["WKSHT_CD"], axis=1)
    # Sets three (row) indices on the table, then unstacks the CLMN_NUM index
    # into separate columns. The resulting table has multiindexed rows
    # (RPT_REC_NUM, then LINE_NUM), and columns given by the value of CLMN_NUM.
    return table.set_index(["RPT_REC_NUM", "LINE_NUM", "CLMN_NUM"]).unstack()

def get_worksheet(code, alpha, nmrc):
    """Selects entries from the alpha and nmrc tables corresponding to a given
    worksheet code"""
    alpha = alpha[alpha["WKSHT_CD"] == code]
    nmrc = nmrc[nmrc["WKSHT_CD"] == code]
    # Reformats both tables and joins them together.
    joined = reformat(alpha).join(reformat(nmrc), how="outer")

    # The resulting joined table has a unneeded multiindex, so it is dropped.
    joined.columns = joined.columns.droplevel()

    return joined

def select(df, rows, cols):
    """Returns the slice of a joined dataframe corresponding to all RPT_REC_NUMs
    at a given row and column. Also drops the unneeded RPT_REC_NUM multiindex."""
    # slice(None) instructs df.loc to select all of the uppper index.
    return df.loc[(slice(None), rows), cols]

def get_medicaid_utilization(alpha, nmrc):
    """Constructs medicaid utilization as outlined by company."""
    joined = get_worksheet("S300001", alpha, nmrc)

    # droplevel(1)'s are needed in all cases where row single row is selected.
    # TODO: detect this case and roll into select function.
    to_columns = select(joined,(200, 1400, 3200), '00700').unstack(level=1)\
        .join(select(joined, 1400, '00800'), how="outer").droplevel(1).fillna(0)
    to_columns["medicaid_utilization"]   =   (to_columns[200]   +   to_columns[1400]   +
to_columns[3200]) / to_columns["00800"]

    return to_columns

def get_ssi_percentage(alpha, nmrc):
    """Gets SSI percentage from field specified by company."""
    joined = get_worksheet("E00A18A", alpha, nmrc)
    return select(joined, 3000, "00100").droplevel(1)
```

```
def get_geography(alpha, nmrc):
    joined = get_worksheet("S200001", alpha, nmrc)
    out = select(joined, 2600, "00100").droplevel(1)
    out.name = "GEO"
    return out

def get_states_codes(rpt):
    prv_num = rpt["PRVDR_NUM"]
    prv_num = prv_num.to_frame()   # series to df
    prv_num['init_char'] = (prv_num["PRVDR_NUM"] / 10000).astype(int)

    dict = {1: 'Alabama', 2: 'Alaska', 3: 'Arizona', 4: 'Arkansas', 5: 'California',
55: 'California', 75: 'California', 6: 'Colorado', 7: 'Connecticut', 8: 'Delaware', 9:
'District of Columbia', 10: 'Florida', 68: 'Florida', 69: 'Florida', 11: 'Georgia', 12:
'Hawaii', 13: 'Idaho', 14: 'Illinois', 78: 'Illinois', 15: 'Indiana', 16: 'Iowa', 76:
'Iowa', 17: 'Kansas', 70: 'Kansas', 18: 'Kentucky', 19: 'Louisiana', 71: 'Louisiana',
20: 'Maine', 21: 'Maryland', 80: 'Maryland', 22: 'Massachusetts', 30: 'New Hampshire',
31: 'New Jersey', 32: 'New Mexico', 33: 'New York', 34: 'North Carolina', 35: 'North
Dakota', 36: 'Ohio', 72: 'Ohio', 37: 'Oklahoma', 38: 'Oregon', 39: 'Pennsylvania', 73:
'Pennsylvania', 40: 'Puerto Rico', 41: 'Rhode Island', 42: 'South Carolina', 43: 'South
Dakota', 44: 'Tennessee', 45: 'Texas', 67: 'Texas', 74: 'Texas', 46: 'Utah', 47:
'Vermont', 48: 'Virgin Islands', 49: 'Virginia', 50: 'Washington', 51: 'West Virginia',
23: 'Michigan', 24: 'Minnesota', 77: 'Minnesota', 25: 'Mississippi', 26: 'Missouri',
27: 'Montana', 28: 'Nebraska', 29: 'Nevada', 52: 'Wisconsin', 53: 'Wyoming', 56:
'Canada', 59: 'Mexico', 64: 'American Samoa', 65: 'Guam', 66: 'Commonwealth of the
Northern Marianas Islands '}
    prv_num['states'] = prv_num['init_char'].map(dict)
    joined  = prv_num[['PRVDR_NUM', 'states']]
    return joined

def get_s_fields(alpha, nmrc):
    """Gets fields from Worksheet S specified by company."""
    joined = get_worksheet("S100000", alpha, nmrc)

    # Since the goal is to select and join many individual fields, use
    # pd.concat since each object is a series, not a dataframe.
    ctc = select(joined, 100, "00100").droplevel(1)
    medicaid_charges = select(joined, 600, "00100").droplevel(1)
    medicaid_cost = select(joined, 700, "00100").droplevel(1)
    charity_charges = select(joined, 2000, "00300").droplevel(1)
    charity_cost = select(joined, 2300, "00300").droplevel(1)
    total_unreimbursed_uncompensated = select(joined, 3100, "00100").droplevel(1)
    # TODO: Verify that this concat works as expected (outer join).
    out = pd.concat([ctc, medicaid_charges, medicaid_cost,
                     charity_charges, charity_cost,
                     total_unreimbursed_uncompensated], axis=1)
    out.columns = ["CST_TO_CHG", "MED_CHG", "MED_CST",
                   "CHAR_CHG", "CHAR_CST", "TOT_UNR_UNC"]

    return out
```

```
def display(string, window):
    """Prints action to console or GUI depending on whether GUI window object
    is passed into function."""
    if window:
        window.update(string)
    else:
        print(string)
```

The functions defined in Table B2 are primarily merging and selection utility functions, which are used for constructing the summarized data for output.

The data can be summarized as described in the Methods section using the functions defined above.

Table B3. Code developed to perform data summarization.

```
def make_summarized_data(zipname, window=None, response=None):
    """Constructs the summarized data in the format specified by the company."""

    if response:
        rpt, alpha, nmrc = response
    else:
        display("Opening files...", window)
        rpt, alpha, nmrc = open_zip(zipname)
    # Gets provider numbers and fiscal year end dates.
    id_cols = rpt[["RPT_REC_NUM", "PRVDR_NUM", "FY_END_DT"]].set_index("RPT_REC_NUM")

    display("Getting Medicaid utilization...",window)
    medicaid = get_medicaid_utilization(alpha, nmrc)

    display("Getting SSI percentage...", window)
    ssi = get_ssi_percentage(alpha, nmrc)

    display("Getting S100000 worksheet fields...", window)
    s_wksht = get_s_fields(alpha, nmrc)

    display("Getting geography...", window)
    geo = get_geography(alpha, nmrc)

    display("Getting states info...", window)
    states = get_states_codes(rpt)

    display("Joining fields...", window)
    summarized = id_cols.join([medicaid, ssi, s_wksht, geo], how="outer")
    # add states
    summarized = summarized.merge(states, on = 'PRVDR_NUM', how = 'left')
    summarized = summarized.drop_duplicates()

    summarized.columns = ["PRVDR_NUM",
```

```
                            "FY_END_DT",
                            "HMO",
                            "TOT_HOSP",
                            "LAB_DEL_DAYS",
                            "TOT_HOSP",
                            "MED_UTIL",
                            "SSI_PER",
                            "CST_TO_CHG",
                            "MED_CHG",
                            "MED_CST",
                            "CHAR_CHG",
                            "CHAR_CST",
                            "TOT_UNR_UNC",
                            "URBAN_RURAL",
                            "STATES"]
    # Replaces all NaN values with zeroes.
    # TODO: verify that this is the intended behavior.
    summarized.fillna(0, inplace=True)
    summarized["DSH_PAT_PER"] = summarized["MED_UTIL"] + summarized["SSI_PER"]

    return summarized
```

The function in Table B3 serves as the end-to-end function, which takes in the name of a zip file and returns the summarized data.

While the above code is sufficient alone to perform the data handling, the company wishes to have a piece of software that can perform these tasks without the use of a Python editor. The code below defines a GUI (Graphical User Interface) which can be turned into a standalone executable file.

Table B4. Code developed to define layout and basic functions of the GUI.

```
def run_GUI():
    sg.theme('DarkAmber')

    # Creates layout of GUI with given button keys
    layout = [
                [sg.Text("Zip file with data:    "),
sg.FileBrowse(key="in_filename")],
                [sg.Text("Output file location: "),
sg.FolderBrowse(key="out_location")],
                [sg.Text("Output file name: "), sg.InputText(key="out_filename"),
sg.Text(".csv")],
                [],
                [sg.Button('Run'), sg.Button('Cancel')],
                [],
                [sg.Text("Working on: "), sg.Text("", key="working_on")],
            ]


    # Instantiates window
    window = sg.Window('Generate Summarized Cost Report Data', layout, finalize=True)
```

```
    # Loops until window is closed or cancelled
    while True:
        # Reads button events
        event, values = window.read()

        # If user closes window or presses cancel
        if event == sg.WIN_CLOSED or event == 'Cancel':
            break

        # Opening the zip files takes long enough that the computer declares the
        # window unresponsive. window.perform_long_operation ensures that this
        # does not occur. The result of the long operation (the opened files)
        # are stored in the values dictionary at the given key.

        # When run button is pressed
        elif event == 'Run':
            window["working_on"].update("Opening files...")
            window.perform_long_operation(lambda : open_zip(values["in_filename"],
                                full=True), 'FILES OPENED')

        # The event 'FILES OPENED' occurs once the files are opened, as above.
        elif event == "FILES OPENED":

            summarized = make_summarized_data(None, window["working_on"],
values[event])
            # Saves the summarized dataframe to the desired location.

summarized.to_csv(f"{values['out_location']}/{values['out_filename']}.csv")
            window["working_on"].update("Finished!")

        time.sleep(0.1)

    window.close()
```

The code in Table B4 defines a layout for the window, instantiates the window, and allows the window to be closed, and allows the run button to open, summarize, and save the data.

Appendix C

This Appendix will describe some of the observations about the DSH percentage distribution mentioned in the Discussion section from a probabilistic perspective, drawing on some topics from financial mathematics. Some familiarity with these topics is assumed. This section will not offer any explanations about the underlying reasons for these phenomena, rather, the intention is to provide context that may motivate future work.

As mentioned in the Discussion section, the mean change in the DSH percentage for a particular hospital in consecutive years is zero. Treating the DSH percentage over a period of years as a sequence of random variables, this condition may be described as a discrete-time martingale. Unlike traditional martingales, the probability distribution is non-normal, in particular, it is a Laplace distribution. A Laplace distribution is parameterized by a mean μ, a variance b, as given in the formula

$$\mathrm{L}(x|\,\mu, b) = \frac{1}{2b} e^{-|x-\mu|/b},$$

for a particular difference value x. Figure C1 below demonstrates the actual distribution of the DSH percentage differences as well as a Laplace distribution fit to the data. Fitting was performed with functional optimization with L2 loss.

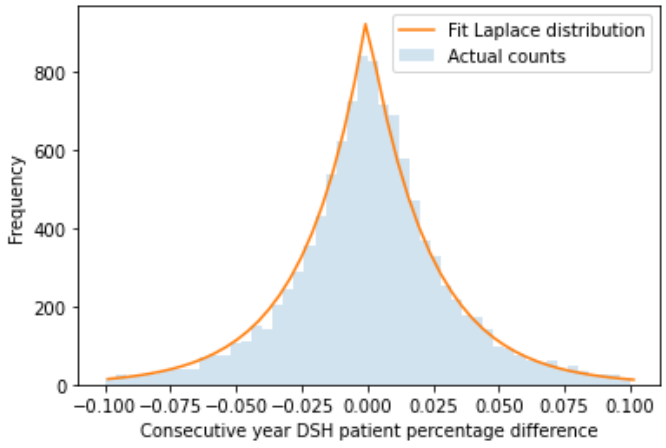


Figure C1. DSH percentage following Laplace Distribution.

Laplace distributions arise from a variety of natural processes, and a natural explanation of why the Laplace distribution appears here may inform future research. As alluded to in the Discussion section, functional regression can be used to approximate the distribution for each pair of years. There is potential for a yearly projection of this distribution by predicting the value of each parameter. However, because of the limited number of years available in the current data, this idea is left as future work as well.