From Data to Decisions: Analyzing Medicare Utilization and Cost Patterns.

Author : Aman Prajapati

Dataset : Hospital Service Area 2023 (Centers for Medicaid and Medicare Services)

Introduction

Understanding how healthcare services are used—and how much they cost—is essential for building a fair and efficient healthcare system. This project was motivated by the growing concern around rising medical costs, uneven access to care, and regional disparities in how patients are treated. By analyzing real Medicare data, we aimed to uncover hidden patterns that could inform better policy decisions and improve how resources are allocated.

We chose the Hospital Service Area 2023 dataset (Centers for Medicaid and Medicare services) because it offers detailed information on Medicare patients, including the total number of cases, care days, and charges across ZIP codes and provider facilities. This made it a strong foundation for identifying which regions or hospitals may be overburdened, underserved, or inefficient in their operations.

The scope of this project includes cleaning the raw data, engineering useful metrics like average cost and duration per case, and using both Python and Tableau to visualize trends. Our analysis focuses on three key questions: which areas have the highest costs, which hospitals are most efficient, and where care might be falling short. The goal is to turn complex healthcare data into clear, actionable insights for healthcare professionals and decision-makers.

Overview of the Data

The dataset used in this project comes from the Hospital Service Area (HSA) file obtained from https://data.cms.gov/provider-summary-by-type-of-service/medicare-inpatient-hospitals/hospital-service-area/data . It summarizes real Medicare inpatient hospital claims collected each year, showing how many patients were treated, how long they stayed, and how much was charged.

Each record in the dataset represents a combination of a hospital and the ZIP code where the patient lives. This helps us understand how care is delivered across different regions and which hospitals are serving which communities.

Key data fields include:

- **MEDICARE_PROV_NUM:** A unique ID for each hospital that provided care.
- **ZIP_CD_OF_RESIDENCE:** Where the patient lives, based on Medicare records.

- **TOTAL_DAYS_OF_CARE :** How many total days patients from a ZIP code spent in a hospital.
- **TOTAL_CHARGES :** The total billing amount (in whole dollars) for those hospital visits.
- **TOTAL_CASES :** The number of separate patient visits or discharges recorded.

This data allowed us to analyze patterns in healthcare usage and cost at a very local level—making it possible to see which areas are overburdened, underserved, or experiencing unusually high charges.

Methodology

Data Preprocessing

We started by cleaning the dataset to fix suppressed values marked with *, converting them into blanks so we could work with numbers properly. Key columns like total charges, care days, and patient cases were cleaned and converted to numeric format.

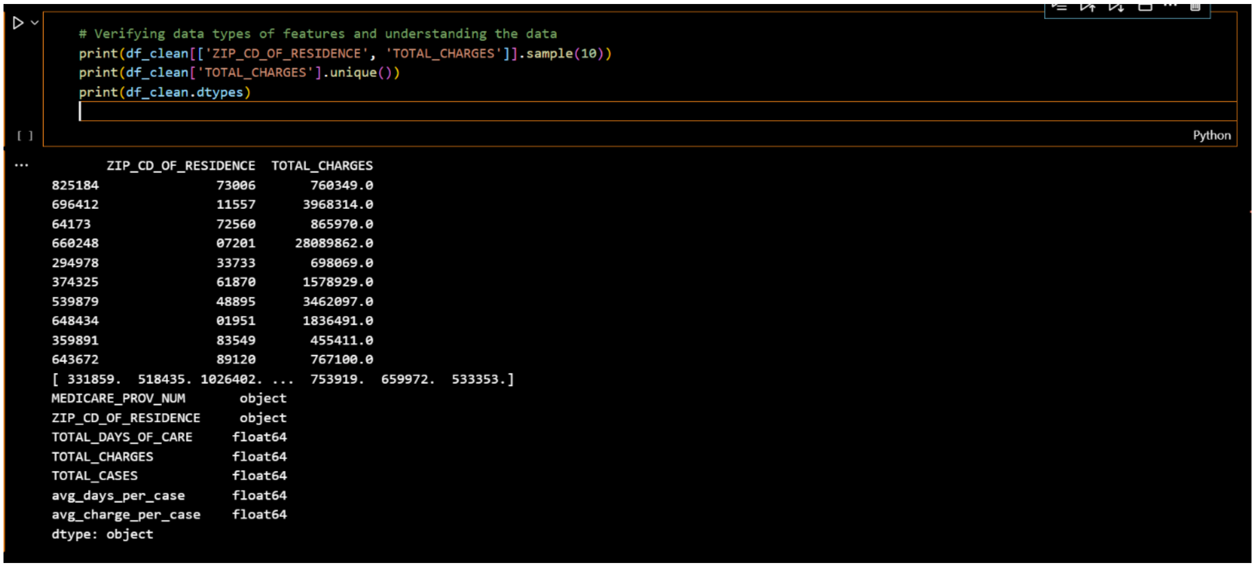
Rows with missing values were removed to keep the analysis accurate. We then created two new columns — average days per case and average charge per case — to help compare hospitals and ZIP codes more fairly.

This cleaned and enriched dataset was then saved for further analysis and visualization.

Feature Engineering

To make the dataset more meaningful and ready for analysis, I created two new features that helped measure hospital efficiency and cost patterns more effectively. The first was avg_days_per_case, calculated by dividing the total number of care days by the number of cases. This gave insight into how long patients typically stayed per visit, which is important for understanding hospital resource usage. The second was avg_charge_per_case, derived by dividing total Medicare charges by total cases. This allowed me to standardize costs across providers and ZIP codes, making it easier to spot unusually high or low charge patterns regardless of patient volume. These engineered features helped transform raw billing data into practical healthcare performance metrics.

Analysis :

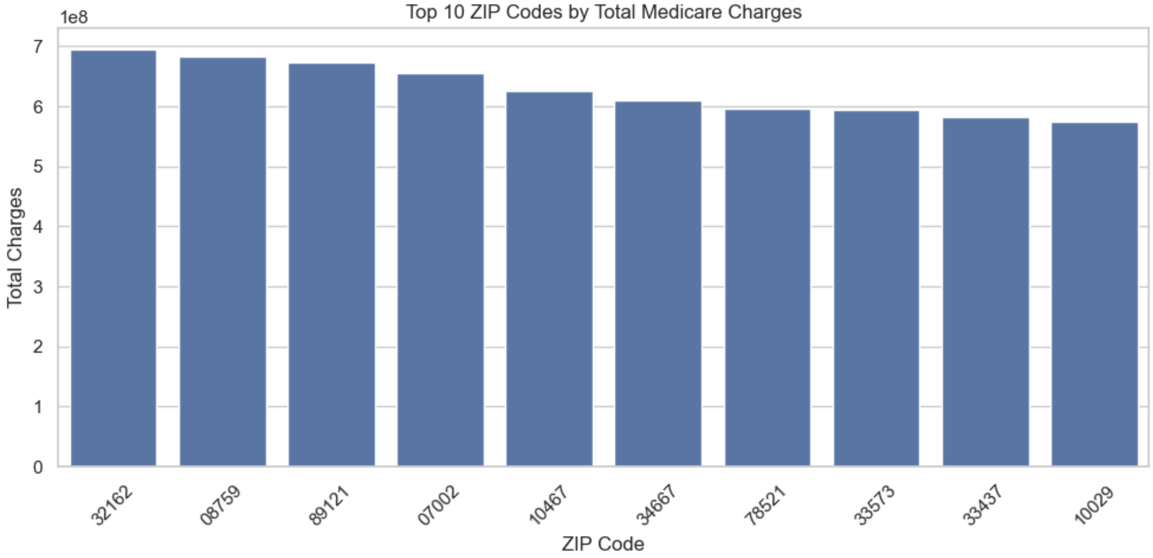


```python
# Verifying data types of features and understanding the data
print(df_clean[['ZIP_CD_OF_RESIDENCE', 'TOTAL_CHARGES']].sample(10))
print(df_clean['TOTAL_CHARGES'].unique())
print(df_clean.dtypes)
```

```
        ZIP_CD_OF_RESIDENCE  TOTAL_CHARGES
825184                73006       760349.0
696412                11557      3968314.0
64173                 72560       865970.0
660248                07201     28089862.0
294978                33733       698069.0
374325                61870      1578929.0
539879                48895      3462097.0
648434                01951      1836491.0
359891                83549       455411.0
643672                89120       767100.0
[ 331859.  518435. 1026402. ...  753919.  659972.  533353.]
MEDICARE_PROV_NUM       object
ZIP_CD_OF_RESIDENCE     object
TOTAL_DAYS_OF_CARE     float64
TOTAL_CHARGES          float64
TOTAL_CASES            float64
avg_days_per_case      float64
avg_charge_per_case    float64
dtype: object
```

Before diving into analysis, I verified the data types and took a sample of values from key columns like TOTAL_CHARGES and ZIP_CD_OF_RESIDENCE. This step helped ensure that the data was correctly cleaned and that the numeric columns were properly interpreted as floating-point numbers, not strings. It also allowed me to spot any potential issues early on (e.g., outliers, unusual formatting, or leftover non-numeric entries). Seeing a spread of realistic charge values and confirming all cost-related features (TOTAL_CHARGES, TOTAL_DAYS_OF_CARE, TOTAL_CASES, and the derived averages) as float64 gave me confidence that the dataset was ready for aggregation, visualization, and further analysis.

Top 10 ZIP Codes by Total Medicare Charges



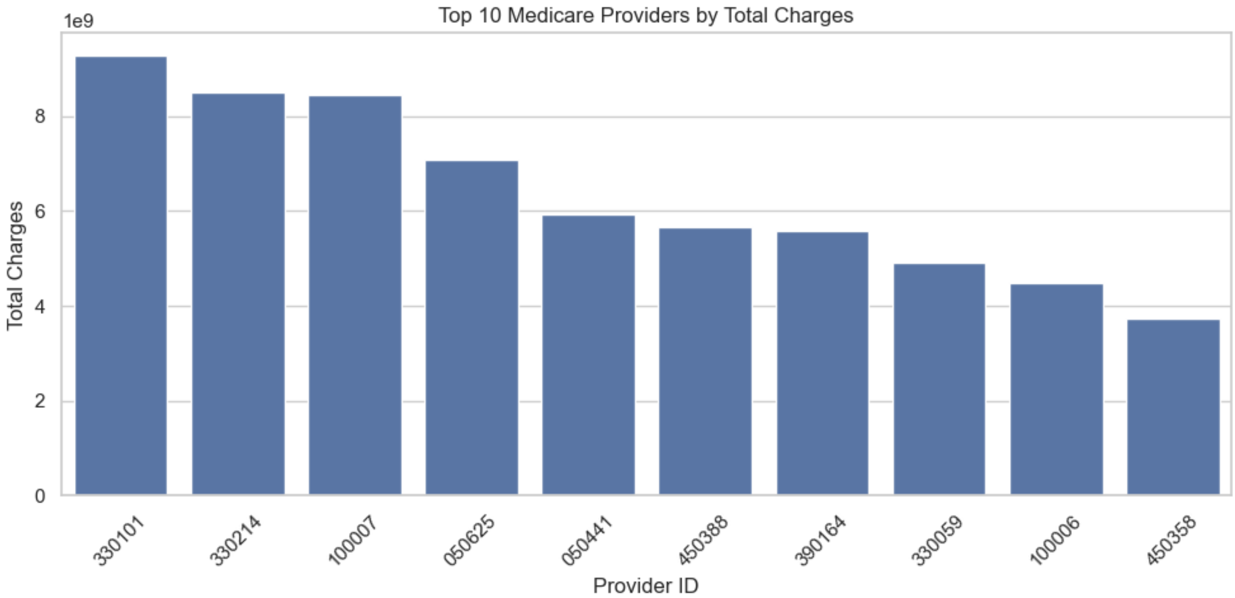
This bar chart shows the **top 10 ZIP codes** in the U.S. that generated the **highest total Medicare charges** in 2023. Each bar represents the combined amount billed to Medicare for patients residing in that ZIP code, regardless of which hospital or provider treated them.

From the analysis, ZIP code **32162** tops the list, followed closely by **08759** and **89121**. These areas likely have a combination of:

- A higher number of Medicare beneficiaries,
- A concentration of elderly populations,
- Or greater healthcare service utilization (e.g., frequent admissions, longer stays, or costlier procedures).

This insight helps highlight **regional hotspots** where Medicare spending is the highest — which can be useful for policymakers, analysts, or hospital administrators looking to investigate healthcare usage, costs, or even potential overbilling patterns.

Top 10 Medicare Providers by Total Charges



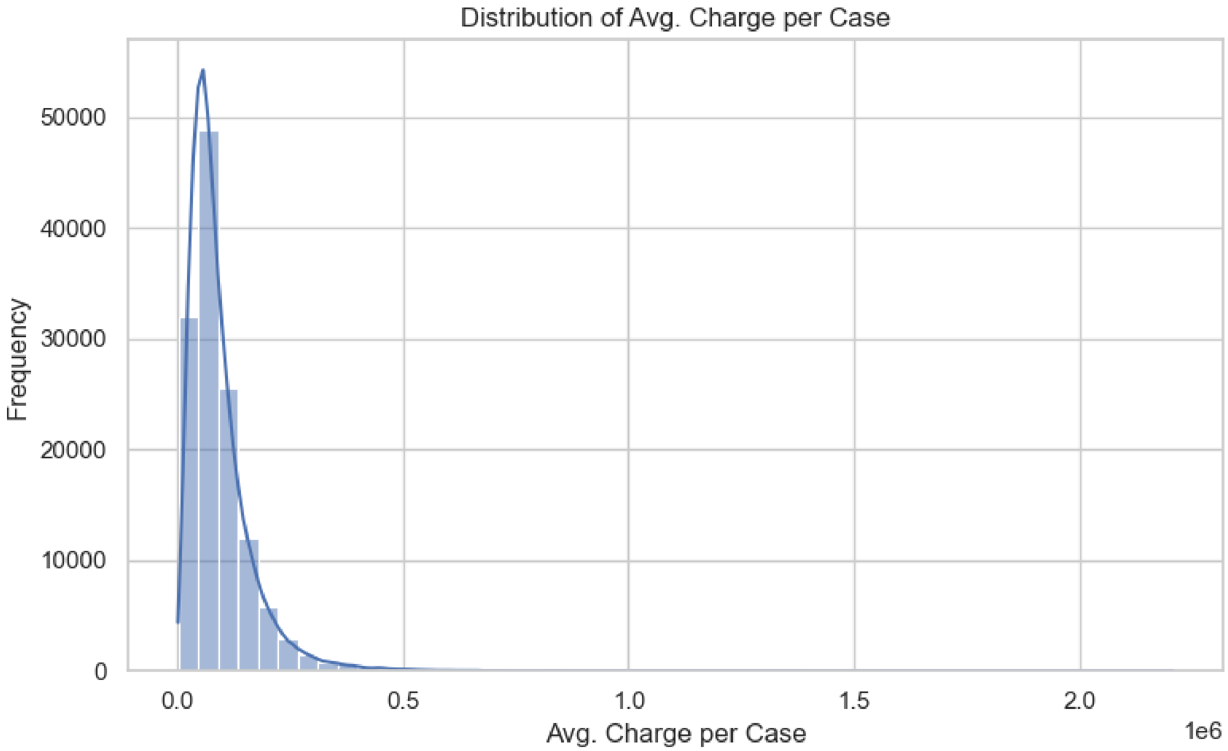
This bar chart highlights the **top 10 healthcare providers** that billed the **highest total charges** to Medicare in 2023. Each provider is represented by their unique Medicare Provider ID, and the height of each bar reflects the overall amount they charged Medicare throughout the year.

From the analysis, provider **330101** stands out as the highest biller, followed closely by providers **330214** and **100007**. These high figures may indicate:

- Large hospital systems with high patient volume
- Specialized treatment centers (e.g., surgical or oncology centers)
- Or locations with complex, high-cost procedures

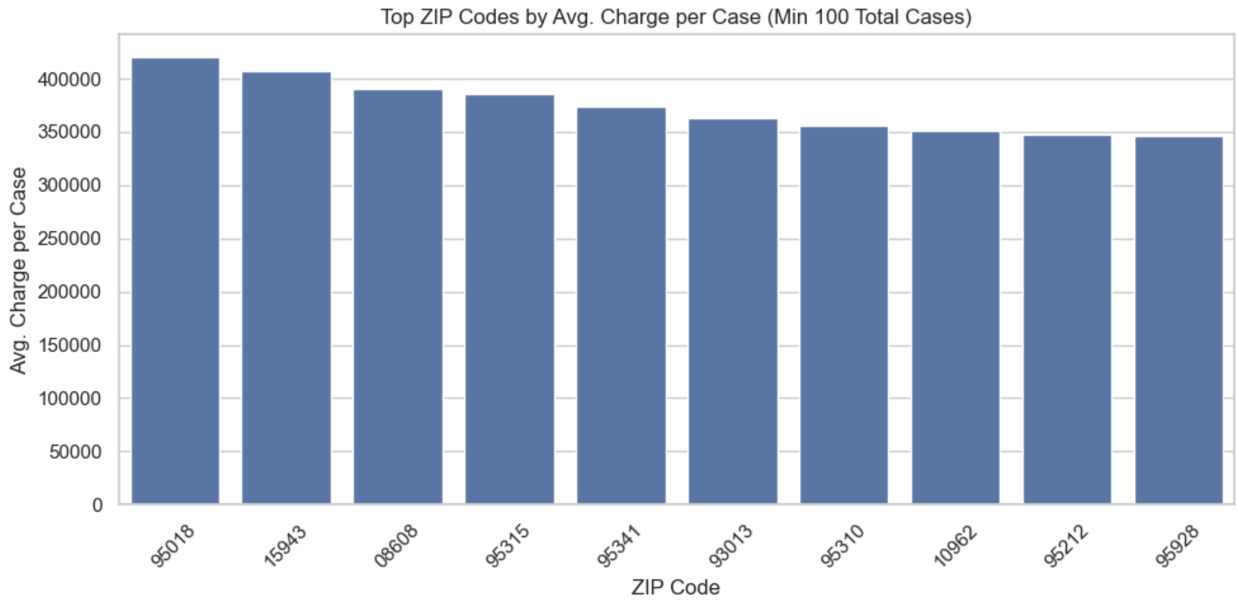
Understanding which providers generate the highest Medicare charges can help identify **cost drivers** in the healthcare system, and may support audits, funding allocation, or policy-making focused on improving cost-efficiency and transparency.

Distribution of Average charge per case



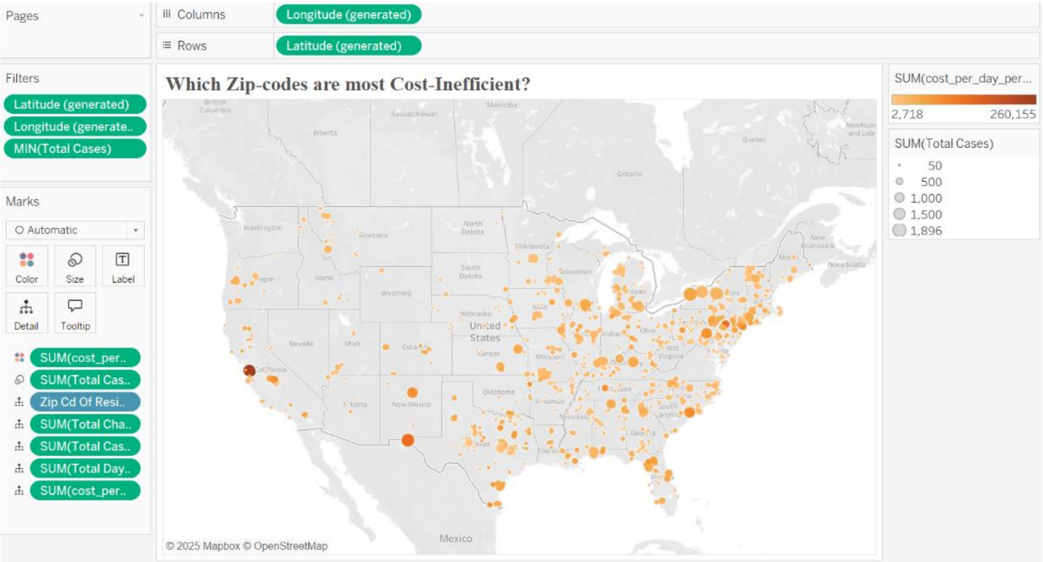
This chart displays how much providers charge on average for each Medicare case. Most providers fall into a lower cost range, but there are a few with significantly higher charges, which creates a long right tail in the distribution. Including this helped me identify **cost outliers** and understand how healthcare charges vary across the system. It supports my project's goal of uncovering potential inefficiencies and pricing disparities in Medicare billing.

Top 10 Zip-codes by Avg. Charge per case



This chart highlights the ZIP codes with the highest average Medicare charges per case, considering only areas with at least 100 total cases. By filtering out low-volume ZIPs, the goal was to focus on regions where the data is more reliable. From the analysis, ZIP code 95018 had the highest average charge per case. This helped me identify high-cost areas that consistently bill more per patient, which could signal either higher complexity of care or potential inefficiencies in how services are delivered.

Highlight ZIPs where costs are unusually high per care day even if the patient count is not justifying it — flagging potential inefficiencies.



Columns: Longitude (generated)

Rows: Latitude (generated)

Which Zip-codes are most Cost-Inefficient?

Filters: Latitude (generated), Longitude (generate.., MIN(Total Cases)

Marks: Automatic

SUM(cost_per.., SUM(Total Cas.., Zip Cd Of Resi.., SUM(Total Cha.., SUM(Total Cas.., SUM(Total Day.., SUM(cost_per..

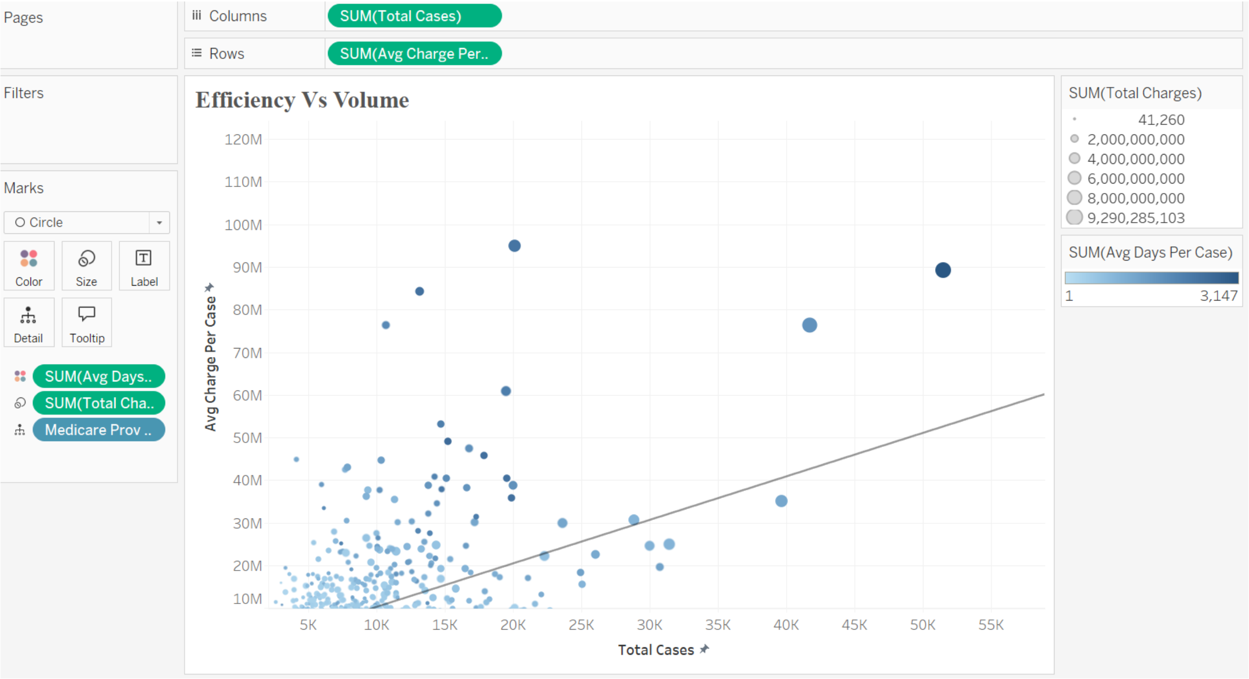
SUM(cost_per_day_per... 2,718 – 260,155

SUM(Total Cases): 50, 500, 1,000, 1,500, 1,896

© 2025 Mapbox © OpenStreetMap

This interactive map visualizes Medicare cost inefficiency by comparing cost per care-day per case across ZIP codes. The darker the color, the higher the cost per day; the larger the bubble, the higher the number of cases. The calculated field cost_per_day_per_case was engineered from the cleaned dataset by dividing total Medicare charges by the total number of care days. This helped identify ZIP codes where Medicare spending is unusually high relative to how long care is actually provided. By also filtering out ZIPs with very low case volume, this visualization focuses on patterns that are statistically meaningful. The goal here was to flag potential inefficiencies, such as overcharging or costly short stays, and help healthcare analysts or policymakers prioritize areas for review.

Correlate provider volume with efficiency — is there a sweet spot where providers manage both volume and cost better?



Columns: SUM(Total Cases)

Rows: SUM(Avg Charge Per..

Marks: Circle

SUM(Avg Days..

SUM(Total Cha..

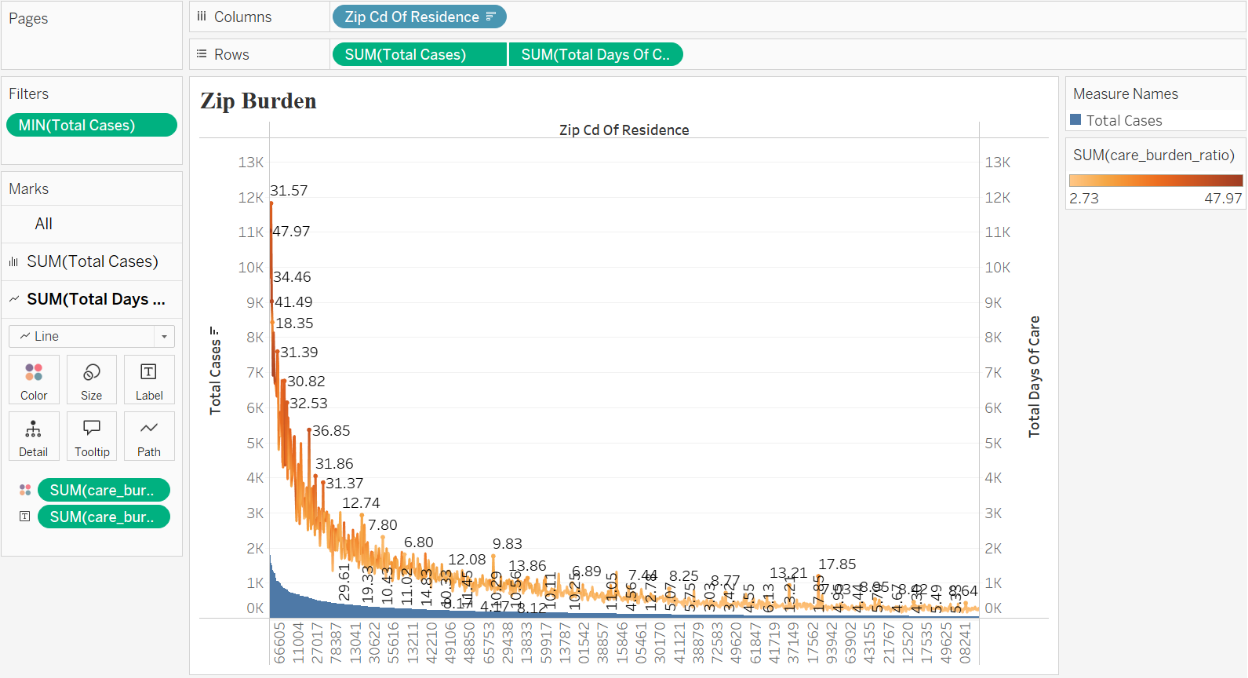
Medicare Prov ..

SUM(Total Charges): 41,260; 2,000,000,000; 4,000,000,000; 6,000,000,000; 8,000,000,000; 9,290,285,103

SUM(Avg Days Per Case): 1 – 3,147

This scatter plot explores the relationship between provider volume and efficiency by comparing the total number of Medicare cases (X-axis) against the average charge per case (Y-axis). Each point represents a provider, sized by total charges and colored by average length of stay (avg_days_per_case). The data comes directly from the cleaned dataset and the feature engineering steps I performed earlier. By adding a trendline, the chart visually tests whether providers handling more cases are also more cost-efficient. Interestingly, the pattern shows that higher volume doesn't always mean lower cost per case — some high-volume providers still have high average charges, while others are far more efficient. This insight is important for questioning assumptions about economies of scale in healthcare delivery.

ZIP Code-Level Burden Analysis: Comparing Patient Volume vs. Care Resources



This dual-axis chart compares total patient cases (bars) with total days of care (line) across ZIP codes, helping uncover potential mismatches in resource allocation. A calculated metric, care_burden_ratio (days of care per case), is used to measure how much care is provided relative to demand. The data comes from the feature-engineered dataset, where this ratio was created to reflect potential service pressure. ZIP codes with high case counts but lower care days may indicate underserved areas, while those with a high care burden ratio could be experiencing resource strain. This visualization helps flag ZIPs that may require policy attention or healthcare support planning.

Challenges and Solutions

One of the main challenges in this project was dealing with masked or incomplete data. Many rows had key values like TOTAL_CASES, TOTAL_CHARGES, and TOTAL_DAYS_OF_CARE replaced with asterisks (*) due to CMS privacy rules, making those rows unusable for numeric analysis. Instead of attempting to impute or guess these values, I chose to remove them entirely to ensure the accuracy and integrity of the insights. Another challenge was transforming basic billing data into meaningful performance metrics. This was addressed through thoughtful feature engineering—specifically calculating metrics like avg_charge_per_case, avg_days_per_case, and care_burden_ratio—which helped turn raw numbers into actionable insights about cost efficiency and care distribution.

Conclusion

This project helped me turn a large Medicare dataset into something meaningful by analyzing hospital service usage, costs, and care patterns across different ZIP codes and providers. Through data cleaning and feature creation, I was able to highlight areas where healthcare spending seems unusually high, providers that may be less efficient, and ZIP codes that could be underserved or overburdened. While the project has its limitations, the insights gained can be valuable for healthcare leaders who want to better understand where resources are being used well—and where improvements might be needed. Projects like this show how data analysis can support smarter decisions and potentially lead to better, fairer care for more people.