CMS-Medicare

Title: Detecting Billing Anomalies in Medicare Data Using BigQuery

Project Statement

The project aims to identify unusual billing patterns from **inpatient_charges_2015** table within the Medicare system that could indicate errors or potential fraudulent activities. By analyzing the billing data available in the cms_medicare dataset, the project seeks to highlight discrepancies and outliers in charges from healthcare providers across different regions and services.

Research Questions

1. What are the common anomalies in billing across different medical procedures and geographic locations?

Tools

Google BigQuery, Google Data Studio, SQL

▼ **Steps**

1. Data Exploration
2. Define Anomalies
3. Conduct Detection Analysis
4. Create Visualizations
5. Generate Report

▼ **Data Info**

provider_id, provider_name, provider_street_address, provider_city, provider_state, provider_zipcode, drg_definition, hospital_referral_region_description, total_discharges, average_covered_charges, average_total_payments, average_medicare_payments

provider_id: String, Required

provider_name: String, Nullable

provider_street_address: String, Nullable

provider_city: String, Nullable

provider_state: String, Nullable

provider_zipcode: String, Nullable

drg_definition: String, Required

hospital_referral_region_description: String, Nullable

total_discharges: Integer, Nullable

average_covered_charges: Float, Nullable

average_total_payments: Float, Nullable

average_medicare_payments: Float, Nullable

▼ Step 1: Data Exploration

Checking Missing Value of Nullable columns

Code:

```
select
count(case when provider_name is null then 1 end)as missing_p
count(case when provider_street_address is null then 1 end)as
count(case when provider_city is null then 1 end)as missing_p
count(case when provider_state is null then 1 end)as missing_
count(case when provider_zipcode is null then 1 end)as missin
count(case when hospital_referral_region_description is null
```

```
count(case when total_discharges is null then 1 end)as missi
count(case when average_covered_charges is null then 1 end)as
count(case when average_total_payments is null then 1 end)as
count(case when average_medicare_payments is null then 1 end
from `bigquery-public-data.cms_medicare.inpatient_charges_20
```

Result: No missing value

Checking number of unique provider

Code:

```
select
count(distinct(provider_id)) as num_uni_provider
from `bigquery-public-data.cms_medicare.inpatient_charges_20
```

Result: 3231

Checking number of unique State

Code:

```
select
count(distinct(provider_state)) as num_uni_state
from `bigquery-public-data.cms_medicare.inpatient_charges_201
```

Result: 51

Checking number of unusual number in numerical features

Code:

```
select
count(case when total_discharges <0 then 1 end)as unusual_tot
count(case when average_covered_charges <0 then 1 end)as unus
count(case when average_total_payments <0 then 1 end)as unusu
```

```
count(case when average_medicare_payments <0 then 1 end)as un
from `bigquery-public-data.cms_medicare.inpatient_charges_201
```

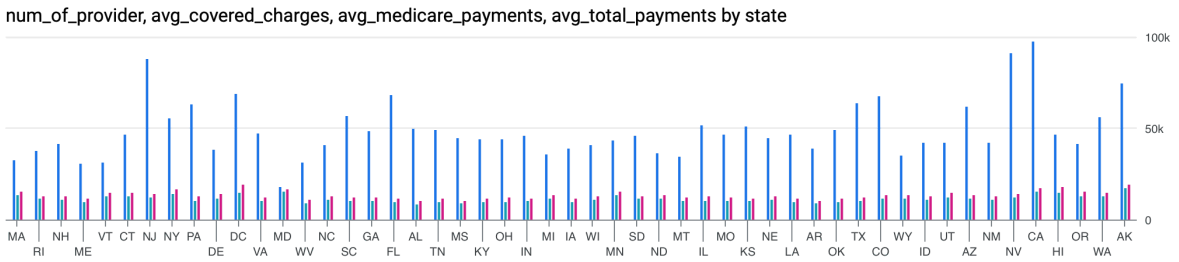
Result: 0

Checking Statistic value for each numerical features for each states

Code:

```
select
provider_state as state,
count(distinct(provider_id)) as num_of_provider,
round(avg(total_discharges),2) as average_discharges,
round(avg(average_covered_charges),2) as avg_covered_charges,
round(avg(average_medicare_payments),2) as avg_medicare_payme
round(avg(average_total_payments),2) as avg_total_payments,
round(min(total_discharges),2) as min_discharges,
round(min(average_covered_charges),2) as min_covered_charges,
round(min(average_medicare_payments),2) as min_medicare_payme
round(min(average_total_payments),2) as min_total_payments,
round(max(total_discharges),2) as max_discharges,
round(max(average_covered_charges),2) as max_covered_charges,
round(max(average_medicare_payments),2) as max_medicare_payme
round(max(average_total_payments),2) as max_total_payments,
round(STDDEV(total_discharges),2) as std_discharges,
round(stddev(average_covered_charges),2) as std_covered_charg
round(stddev(average_medicare_payments),2) as std_medicare_pa
round(stddev(average_total_payments),2) as std_total_payments
from `bigquery-public-data.cms_medicare.inpatient_charges_201
group by provider_state;
```

Result:



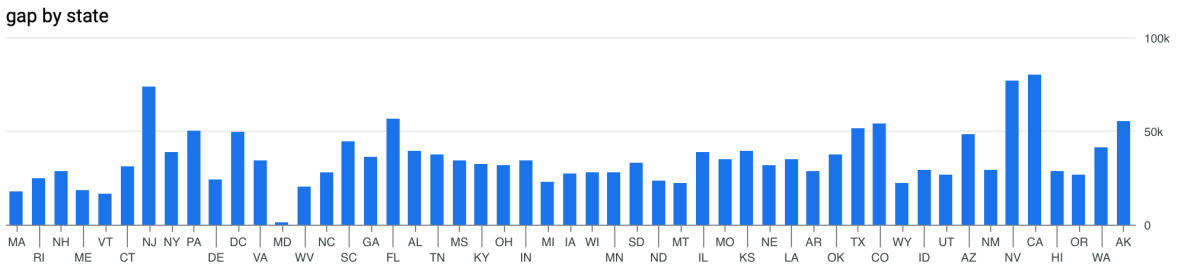
To better interpret these graph, it is very important to understand the relationship between covered payment, medicare payment, and total payment. Covered charges represent the requested amount by providers, medicare payments show what medicare agrees to pay, and total payments capture the total actual income that the provider receives for the services. So the covered payment is usually higher than other two payment and total payment is higher than medicare payment. Based on the understanding of these terms, we can then knowing which state should be further investigated to see whether there's anomalies exist.

The following states is needed to be further explored for following reason, **The range of gap between covered charge and total payment is too much**

Code:

```
select
provider_state as state,round(avg(average_covered_charges),2)
from `bigquery-public-data.cms_medicare.inpatient_charges_201
group by provider_state;
```

Results:



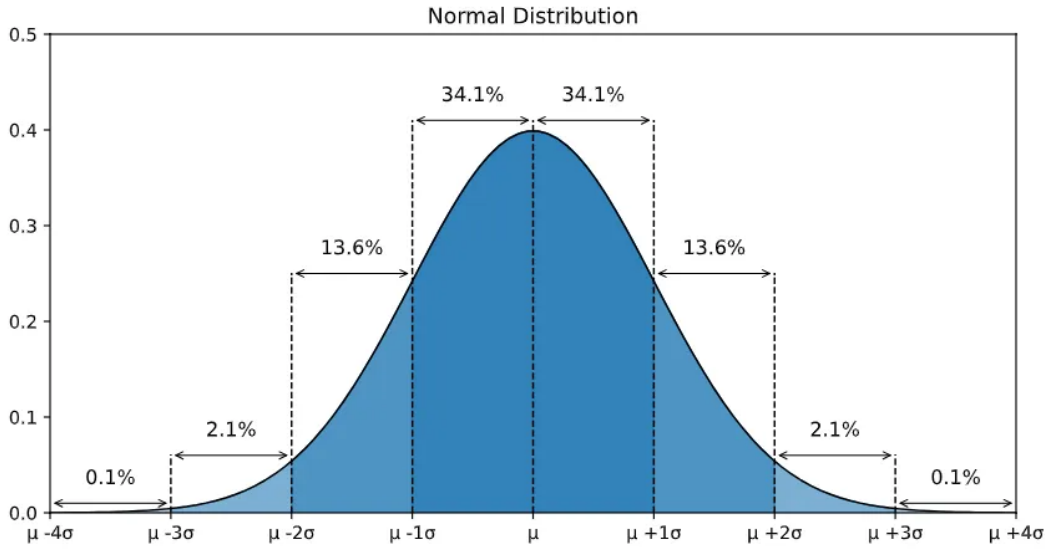
The states contains: NJ, FL, CO, NV, CA, AK

And to make it simple now, I'll demonstrate only CA which stands for the most wide range among all states' gap.

Step 2: Define Anomalies

I want to set up this gap between the initial charges, covered_charges, and the final payment that hospital end receives, total_payments, to be the items to looking into and see if there's any anomaly exist among the providers from CA. I'll demonstrate 2 different methods to examine the anomaly, 1. Standard Deviation 2. IQR method.

1. Standard Deviation



- Roughly 68.3% of the data is within 1 standard deviation of the average (from μ-1σ to μ+1σ)
- Roughly 95.5% of the data is within 2 standard deviations of the average (from μ-2σ to μ+2σ)
- Roughly 99.7% of the data is within 3 standard deviations of the average (from μ-3σ to μ+3σ)

We can follow the empirical rule, also called the 68–95–99.7 rule. The rule tells us that, for a normal distribution, there's a:

- 68 % chance a data point falls within 1 standard deviation of the mean
- 95 % chance a data point falls within 2 standard deviations of the mean
- 99.7 % chance a data point falls within 3 standard deviations of the mean

So I might think the data point that falls outside 3 standard deviations of the mean can be consider as anomalies in this case.

Code:

```
with na_stat as (
select
avg(average_covered_charges-average_total_payments)as avg_gap
stddev(average_covered_charges-average_total_payments) as std
from `bigquery-public-data.cms_medicare.inpatient_charges_201

select provider_name, avg(average_covered_charges-average_tot
case when (avg(average_covered_charges-average_total_payments
else 'Normal'
end as status
from `bigquery-public-data.cms_medicare.inpatient_charges_201
where provider_state ='CA'
group by provider_name;
```

Results:

The provider from CA and fall outside 3 standard deviations of the mean has only one which is Stanford Healthcare with 207789.49 gap value.

2. IQR

$$IQR = Q3 - Q1$$

1. **First Quartile (Q1):** Q1 is the value below which 25% of the data falls. Mathematically, it is the data point at the 25th percentile.
2. **Second Quartile (Q2):** Q2 is the median, representing the middle value of the dataset. 50% of the data falls below the median, and 50% falls

above.

3. **Third Quartile (Q3):** Q3 is the value below which 75% of the data falls. Mathematically, it is the data point at the 75th percentile.

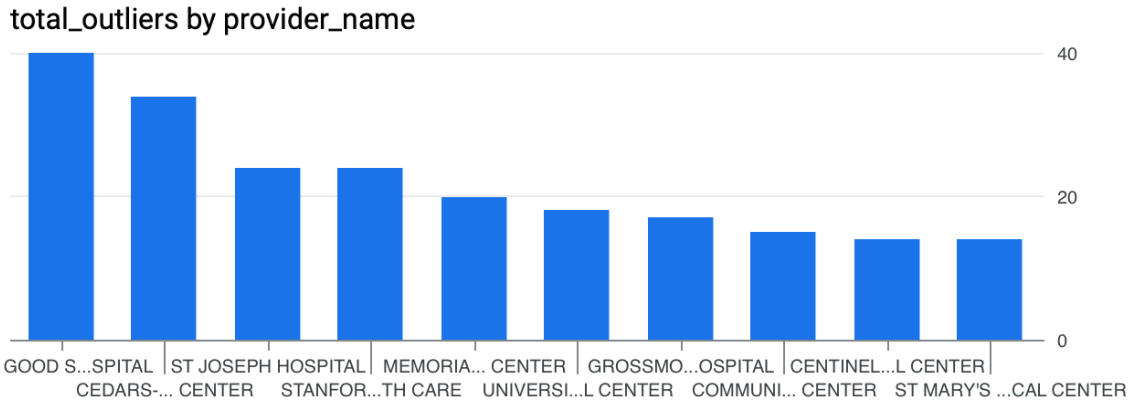
Code:

```
WITH gap_data AS (
  SELECT
    provider_state,
    provider_name,
    (average_covered_charges - average_total_payments) AS
  FROM
    `bigquery-public-data.cms_medicare.inpatient_charges_2
),
Q_stat AS (
  SELECT
    provider_state,
    provider_name,
    gap,
    PERCENTILE_CONT(gap, 0.25) OVER (PARTITION BY provider
    PERCENTILE_CONT(gap, 0.75) OVER (PARTITION BY provider
  FROM
    gap_data
),
IQR_stat AS (
  SELECT
    provider_state,
    provider_name,
    gap,
    Q1,
    Q3,
    (Q3 - Q1) AS IQR,
    (Q3 + 1.5 * (Q3 - Q1)) AS Upper_Bond
  FROM
    Q_stat
),
```

```
outlier_stat AS (
  SELECT
    provider_state,
    provider_name,
    gap,
    CASE
      WHEN gap > Upper_Bond THEN 'Outlier'
      ELSE 'Normal'
    END AS status
  FROM
    IQR_stat
  WHERE
    provider_state = 'CA'
)

SELECT
  provider_name,
  COUNT(*) AS total_outliers
FROM
  outlier_stat
WHERE
  status = 'Outlier'
GROUP BY
  provider_name
order by total_outliers desc
limit 10;
```

Results:



Conclusion

The project “Detecting Billing Anomalies in Medicare Data Using BigQuery” successfully identified potential discrepancies and outliers in billing patterns across various healthcare providers within the Medicare system. By leveraging the inpatient_charges_2015 table from the cms_medicare dataset, we explored and analyzed the relationships between covered charges, Medicare payments, and total payments. The findings revealed specific states—particularly California, New Jersey, Florida, Colorado, Nevada, and Alaska—where significant gaps between covered charges and total payments warranted further investigation.

Using statistical methods such as the Standard Deviation and Interquartile Range (IQR) techniques, we established thresholds to detect anomalies. For instance, the analysis identified Stanford Healthcare in California as having a notable gap of $207,789.49, indicating a potential billing irregularity. These insights highlight the importance of monitoring billing patterns in healthcare to mitigate errors and prevent fraudulent activities.

This analysis provides a foundation for ongoing scrutiny of Medicare billing practices. By employing tools like Google BigQuery and SQL, the project underscores the effectiveness of data-driven approaches in enhancing transparency and accountability in healthcare billing. Future work could expand on this analysis by incorporating additional years of data, exploring other regions, and implementing more sophisticated anomaly detection algorithms to refine our understanding of billing discrepancies in the Medicare

system. Overall, the project demonstrates the potential for data analysis to inform policy decisions and improve the integrity of healthcare financial practices.