Healthcare Data Analysis

Introduction:

The purpose of this project is to analyse patient data from a healthcare dataset to extract meaningful insights using big data techniques and Databricks tools. The analysis focuses on understanding various patterns in patient demographics, medical conditions, hospital admissions and billing amounts to help healthcare organisations make informed decisions. We aim to use the data to identify trends in patient admissions, the frequency of medical conditions and the cost distribution associated with different treatments and hospitals.

Data Loading and Schema Inspection:

The Kaggle healthcare dataset is loaded from DBFS into a PySpark DataFrame. The schema is inspected to understand the structure and data types, ensuring the data is ready for further analysis.

Data Transformation:

Columns like "Name" are transformed to standardize the formatting (e.g., capitalizing the first letter).

3

```
df = spark.read.csv("/FileStore/tables/healthcare_dataset.csv", header=True, inferSchema=True)
display(df)
```

Table

	Name	Age	Gender	Blood Type	Medical Condition	Date of Admission	Doct
1	Bobby JacksOn	30	Male	B-	Cancer	2024-01-31	Matthew
2	LesLie TErRy	62	Male	A+	Obesity	2019-08-20	Samanth
3	DaNnY sMitH	76	Female	A-	Obesity	2022-09-22	Tiffany M
4	andrEw waTtS	28	Female	O+	Diabetes	2020-11-18	Kevin We
5	adrIENNE bEll	43	Female	AB+	Cancer	2022-09-19	Kathleen
6	EMILY JOHNSOn	36	Male	A+	Asthma	2023-12-20	Taylor Ne
7	edwArD EDWaRDs	21	Female	AB-	Diabetes	2020-11-03	Kelly Ols
8	CHrisTInA MARtinez	20	Female	A+	Cancer	2021-12-28	Suzanne
9	JASmINe aGullaR	82	Male	AB+	Asthma	2020-07-01	Daniel Fe
10	ChRISTopher BerG	58	Female	AB-	Cancer	2021-05-23	Heather
11	mIchElLe daniELs	72	Male	O+	Cancer	2020-04-19	John Du
12	aaRon MARtiNeZ	38	Female	A-	Hypertension	2023-08-13	Douglas
13	connOR HANsEn	75	Female	A+	Diabetes	2019-12-12	Kenneth
14	rObeRt bAuer	68	Female	AB+	Asthma	2020-05-22	Theresa
15	bROOkE brady	44	Female	AB+	Cancer	2021-10-08	Roberta

10,000+ rows | Truncated data due to row limit

4

```
# Checking the schema of the DataFrame
df.printSchema()
```

```
root
 |-- Name: string (nullable = true)
 |-- Age: integer (nullable = true)
 |-- Gender: string (nullable = true)
 |-- Blood Type: string (nullable = true)
 |-- Medical Condition: string (nullable = true)
 |-- Date of Admission: date (nullable = true)
 |-- Doctor: string (nullable = true)
 |-- Hospital: string (nullable = true)
 |-- Insurance Provider: string (nullable = true)
 |-- Billing Amount: double (nullable = true)
 |-- Room Number: integer (nullable = true)
 |-- Admission Type: string (nullable = true)
 |-- Discharge Date: date (nullable = true)
 |-- Medication: string (nullable = true)
 |-- Test Results: string (nullable = true)
```

5

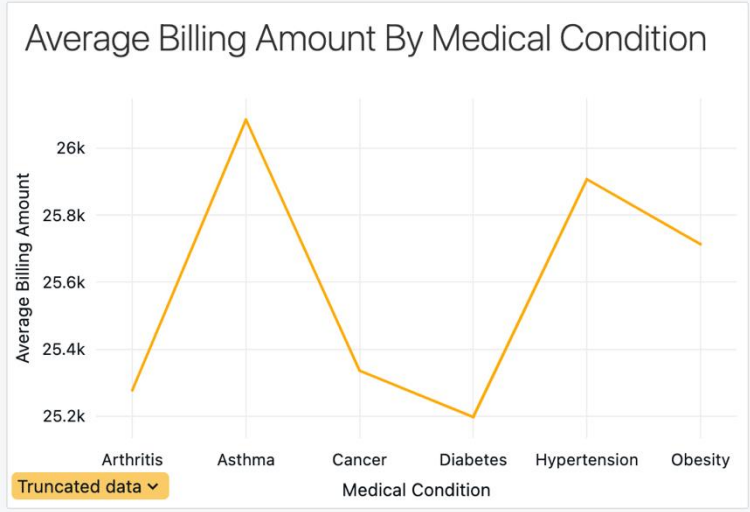
```
#Capitalising first letter of Name
from pyspark.sql.functions import initcap

df = df.withColumn("Name", initcap("Name"))
display(df)
```

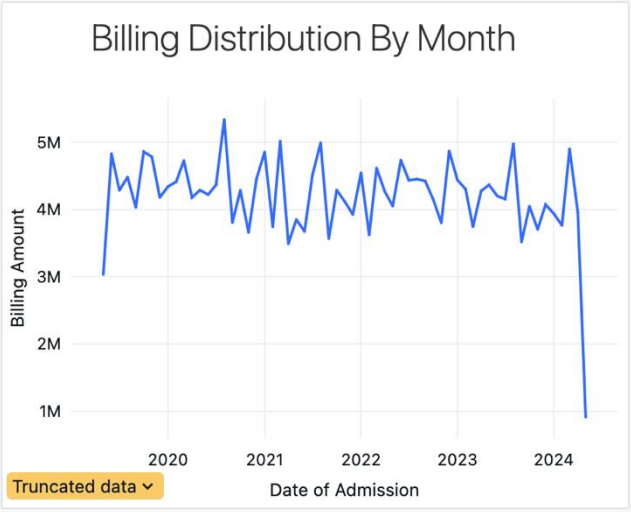
Before Transformation

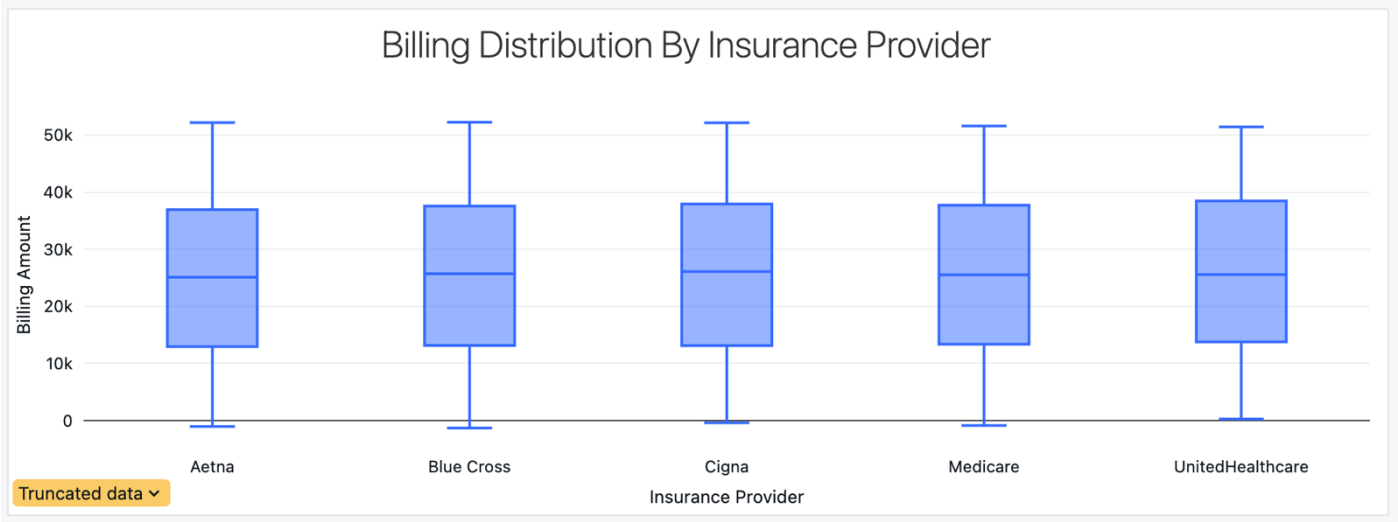
	Name	Age	Gender	Blood Ty
1	Bobby JacksOn	30	Male	B-
2	LesLie TErRy	62	Male	A+
3	DaNnY sMitH	76	Female	A-
4	andrEw waTtS	28	Female	O+
5	adrIENNE bEll	43	Female	AB+
6	EMILY JOHNSOn	36	Male	A+

After Transformation

	Name	Age	Gender	Blood Type
1	Bobby Jackson	30	Male	B-
2	Leslie Terry	62	Male	A+
3	Danny Smith	76	Female	A-
4	Andrew Watts	28	Female	O+
5	Adrienne Bell	43	Female	AB+
6	Emily Johnson	36	Male	A+







Analysis:

Average Billing Amount by Medical Condition:

The lowest average billing is for Diabetes at 25,198, while the highest is for Asthma at 26,086. This suggests that billing costs vary slightly by medical condition, with asthma treatments or procedures generally costing more on average than those for diabetes.

Billing Distribution by Insurance Provider:

The box plot shows a nearly normal distribution for billing amounts across all insurance providers, with the median values ranging from 25,000 to over 26,000. This indicates a consistent range of billing costs regardless of the insurance provider.

Billing Distribution by Month:

The trend in billing distribution shows a decline from May 2019 (highest at 3,036,708) to May 2024 (lowest at 912,944). The highest billing is observed in August 2020 at 5,338,718. This suggests significant variability in monthly billing amounts over time, possibly due to external factors such as healthcare demand, policy changes or seasonal variations.

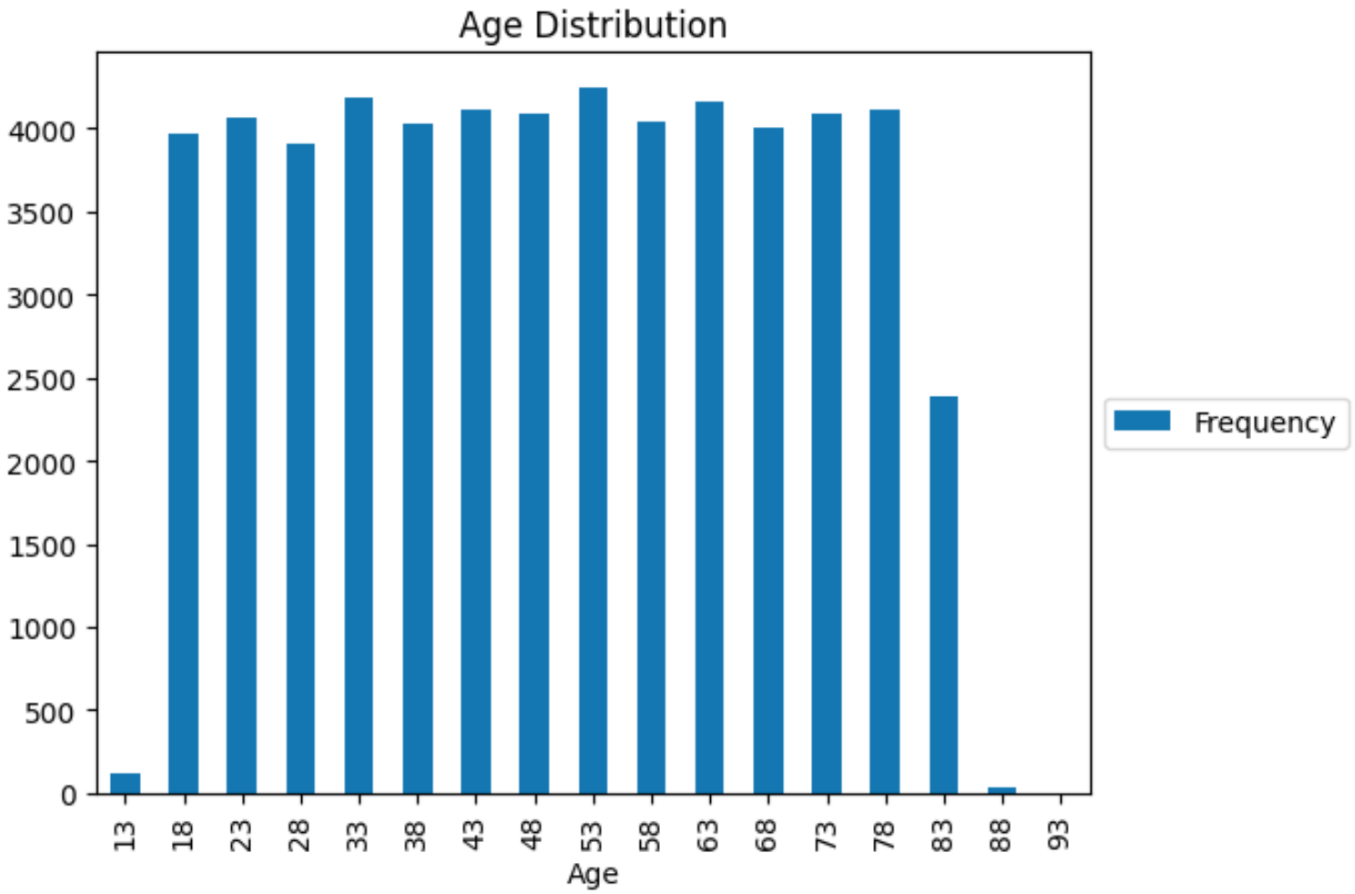
Descriptive Analysis:

Histograms are generated to analyze the distribution of patient ages and billing amounts. The frequency of medical conditions is analyzed based on admission type and blood type.

8

```
%pip install matplotlib pandas numpy

import matplotlib.pyplot as plt
import pandas as pd
import numpy as np


# Adjusting bins to have a wider range for Age Distribution
age_histogram = df.select("Age").rdd.flatMap(lambda x: [x]).histogram(list(np.arange(start=min(df.select("Age").rdd.flatMap(lambda x: [x]).min()),
                                                                                     stop=max(df.select("Age").rdd.flatMap(lambda x: [x]).max()) + 10,
                                                                                     step=5)))
age_pd = pd.DataFrame(list(zip(*age_histogram)), columns=['Age', 'Frequency'])

# Plotting with frequency legend outside graph area
fig, ax = plt.subplots()
age_pd.plot(kind='bar', x='Age', y='Frequency', title="Age Distribution", ax=ax)
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.close(fig)  # Prevents the automatic display of the figure in Jupyter-like environments
display(fig)
```

```
Note: you may need to restart the kernel using dbutils.library.restartPython() to use updated packages.
Requirement already satisfied: matplotlib in /databricks/python3/lib/python3.10/site-packages (3.7.0)
Requirement already satisfied: pandas in /databricks/python3/lib/python3.10/site-packages (1.5.3)
Requirement already satisfied: numpy in /databricks/python3/lib/python3.10/site-packages (1.23.5)
Requirement already satisfied: pyparsing>=2.3.1 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: cycler>=0.10 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: pillow>=6.2.0 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: contourpy>=1.0.1 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (1.0.5)
Requirement already satisfied: fonttools>=4.22.0 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: packaging>=20.0 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (23.2)
Requirement already satisfied: pytz>=2020.1 in /databricks/python3/lib/python3.10/site-packages (from pandas) (2022.7)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel using dbutils.library.restartPython() to use updated packages.
```



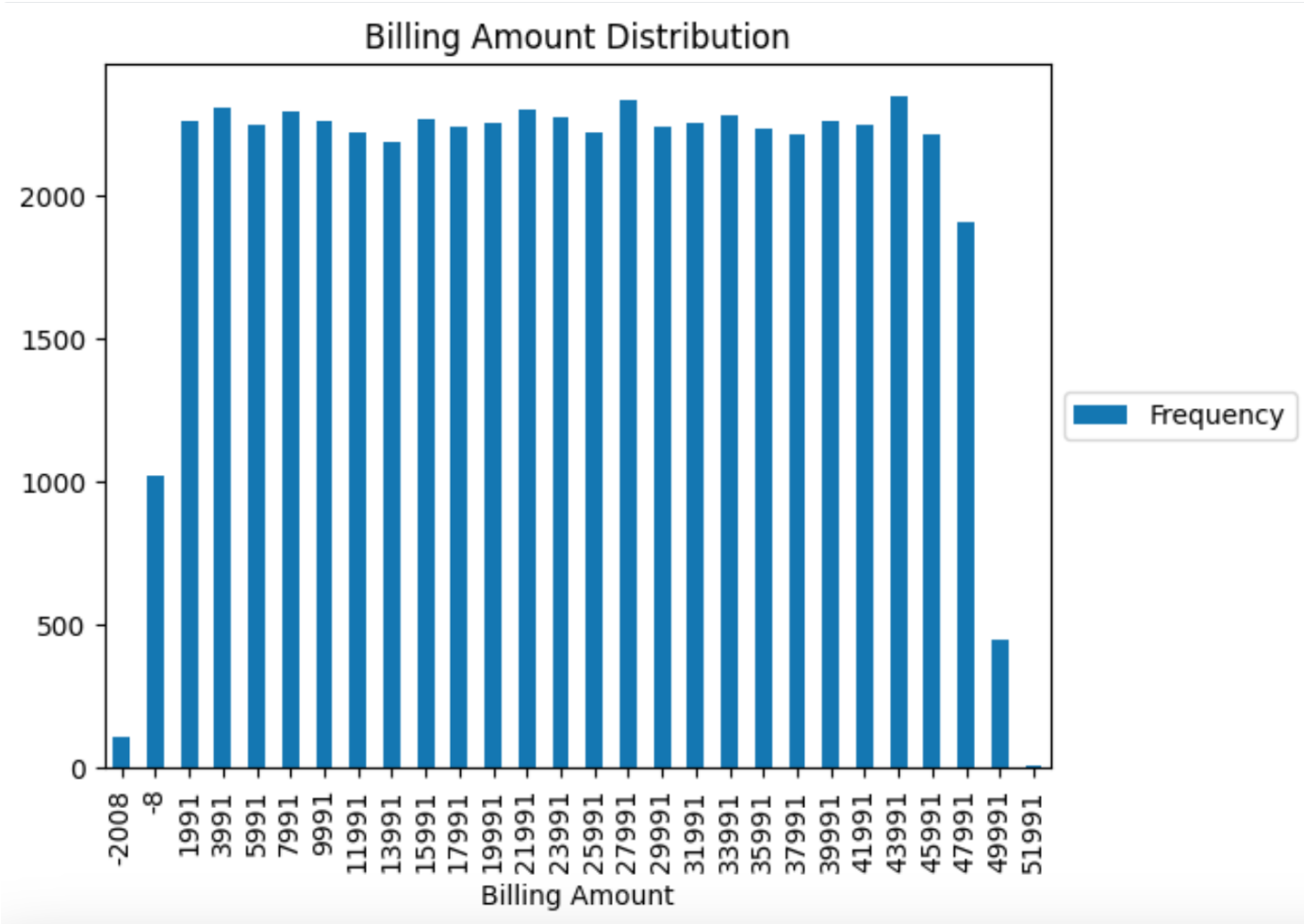
Analysis:

- Most patients are within the age range of 18 to 78, showing a relatively uniform distribution in this age group.
- The frequency drops significantly for ages below 18 and above 78, indicating fewer admissions in these age groups.

10

```
%pip install matplotlib pandas

import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# Adjusting bins for Billing Amount Distribution in whole numbers and increasing bin range in thousands
billing_amount_min = df.select("Billing Amount").rdd.flatMap(lambda x: x).min()
billing_amount_max = df.select("Billing Amount").rdd.flatMap(lambda x: x).max()
bin_range = np.arange(start=billing_amount_min, stop=billing_amount_max + 2000, step=2000).tolist()  # Adjust step for bin size

billing_histogram = df.select("Billing Amount").rdd.flatMap(lambda x: [x[0]]).histogram(bin_range)
billing_pd = pd.DataFrame(list(zip(*billing_histogram)), columns=['Billing Amount', 'Frequency'])

# Remove decimals from Billing Amount
billing_pd['Billing Amount'] = billing_pd['Billing Amount'].astype(int)

# Plotting with frequency legend outside graph area
fig, ax = plt.subplots()
billing_pd.plot(kind='bar', x='Billing Amount', y='Frequency', title="Billing Amount Distribution", ax=ax)
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.close(fig)  # Prevents the automatic display of the figure in Jupyter-like environments
display(fig)
```

```
Note: you may need to restart the kernel using dbutils.library.restartPython() to use updated packages.
Requirement already satisfied: matplotlib in /databricks/python3/lib/python3.10/site-packages (3.7.0)
Requirement already satisfied: pandas in /databricks/python3/lib/python3.10/site-packages (1.5.3)
Requirement already satisfied: kiwisolver>=1.0.1 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: fonttools>=4.22.0 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: cycler>=0.10 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: contourpy>=1.0.1 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (1.0.5)
Requirement already satisfied: python-dateutil>=2.7 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: numpy>=1.20 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (1.23.5)
Requirement already satisfied: packaging>=20.0 in /databricks/python3/lib/python3.10/site-packages (from matplotlib) (23.2)
Requirement already satisfied: pytz>=2020.1 in /databricks/python3/lib/python3.10/site-packages (from pandas) (2022.7)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel using dbutils.library.restartPython() to use updated packages.
```

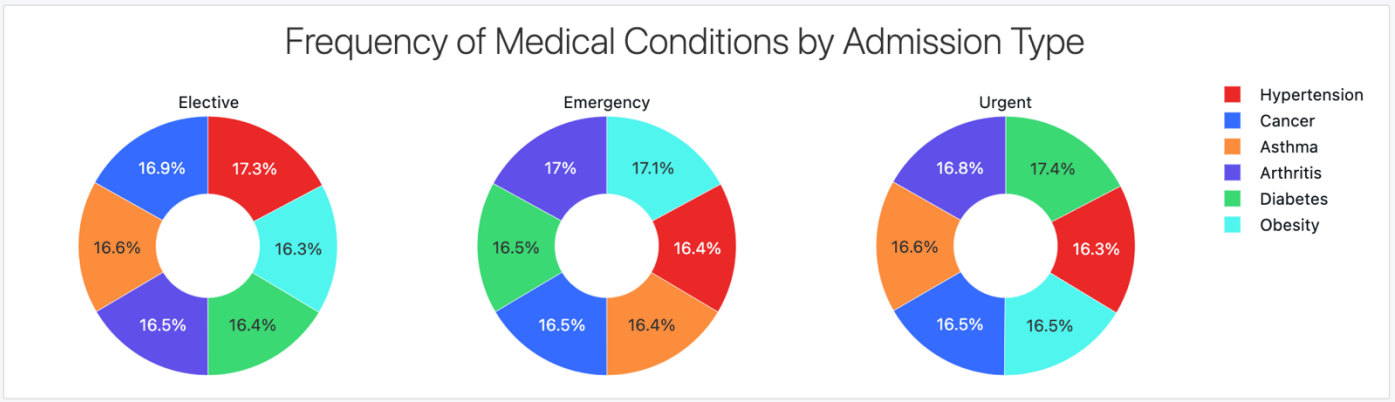


Analysis:

- Billing amounts between 1991 and 47,991 have relatively uniform frequencies, indicating a consistent range of treatment costs, with billing amounts up to 45991 have a consistently high frequency, i.e., above 2000
- Extremely low (< 1991) or high billing amounts (> 47991) show significantly lower frequencies, suggesting these are less common. Negative billing amounts are retained in the dataset as they may represent refunds, adjustments or corrections in the billing records.

12

```
#Creating a graph of Medical Condition by Admission Type
from pyspark.sql.functions import count

# Grouping data by Medical Condition and Admission Type, then counting occurrences
histogram_data = df.groupBy("Medical Condition", "Admission Type").agg(count("*").alias("Frequency"))

display(histogram_data)
```



	Medical Condition	Admission Type	Frequency
1	Cancer	Emergency	3015
2	Hypertension	Emergency	3001
3	Cancer	Urgent	3064
4	Diabetes	Emergency	3017
5	Arthritis	Emergency	3108
6	Hypertension	Urgent	3023
7	Obesity	Elective	3043
8	Obesity	Urgent	3062
9	Diabetes	Urgent	3229
10	Arthritis	Elective	3083
11	Asthma	Emergency	3002
12	Asthma	Elective	3102
13	Arthritis	Urgent	3117

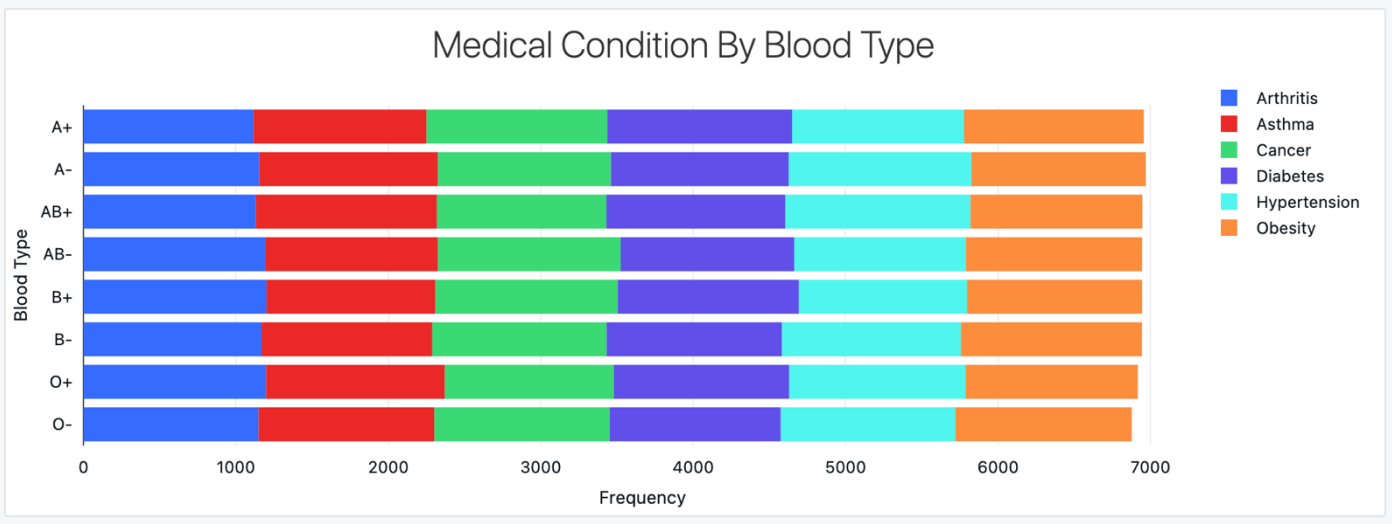
Analysis:

The nearly uniform distribution of chronic conditions across elective, urgent, and emergency care suggests these issues are consistently critical regardless of care type, necessitating comprehensive and ongoing management.

- Hypertension is more prevalent in elective care, suggesting patients regularly manage this condition through planned medical visits.
- Diabetes is notable in urgent care, pointing to acute complications requiring prompt attention.
- Obesity is higher in emergency care, reflecting its potential to lead to severe, immediate health crises.

14

```
#Creating a graph of Medical Condition by Blood Type
from pyspark.sql.functions import count

# Grouping by Blood Type and Medical Condition
blood_type_condition_histogram = df.groupBy("Blood Type", "Medical Condition").agg(count("*").alias("Frequency"))

display(blood_type_condition_histogram)
```



	Blood Type	Medical Condition	Frequency
1	O+	Diabetes	1151
2	AB-	Diabetes	1139
3	A+	Asthma	1135
4	O-	Asthma	1154
5	B-	Arthritis	1169
6	A-	Asthma	1173
7	O+	Arthritis	1198
8	O+	Asthma	1173
9	B-	Hypertension	1173
10	AB-	Hypertension	1125
11	O+	Obesity	1130
12	AB+	Arthritis	1130
13	O-	Arthritis	1149

Analysis:

- Diabetes is most frequent in A+ (1213), suggesting this blood type might have a higher prevalence for this condition.
- Asthma has the highest frequency in AB+ (1189), indicating a greater impact on this blood type.
- Arthritis is most common in B+ (1201), showing a significant occurrence in this group.
- Hypertension is highest in AB+ (1215), highlighting a particular vulnerability for this blood type.
- Obesity is more frequent in B- (1188), suggesting a notable correlation.
- Cancer shows a high occurrence in AB- (1198), indicating a significant presence within this blood type.

Exploratory SQL Queries:

SQL queries are used to identify patterns, such as the most common medical conditions by hospital and the medical conditions associated with the highest average billing amounts.

Visualization and Advanced Analysis:

Visualizations are created to graphically represent the distribution of billing amounts and the frequency of medical conditions across different hospitals.

Complex SQL queries are executed to determine the costliest medical conditions by hospital and to understand the impact of age on billing amounts.

17

```
#Finding frequency of admissions of all hospitals
hospital_histogram = df.groupBy("Hospital").agg(count("*").alias("Frequency"))

display(hospital_histogram)
```

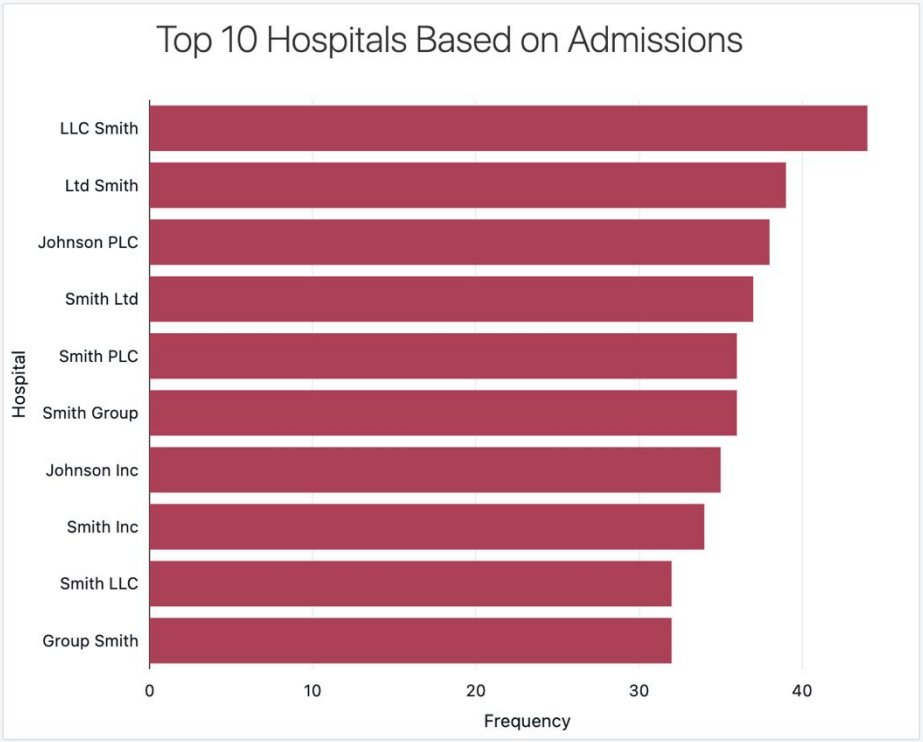
Table

	Hospital	Frequency
1	Ramirez-Robinson	3
2	Foster Lamb, Graham and	1
3	LLC Massey	2
4	Coleman-Aguilar	1
5	Group Stein	2
6	Smith PLC	36
7	Alvarado-Martin	1
8	Hall Group	8
9	and Mayo Chen, Murray	1
10	Lopez-Wilson	2
11	Harris-Farrell	1
12	Dawson-Williams	2
13	Holmes Reed and Johnson,	1
14	and Lee Rodriguez Morris,	1
15	Watkins, and Young Perry	1

10,000+ rows | Truncated data due to row limit

18

```
#Filtering Top 10 Hospital based on number of admissions
hospital_histogram.createOrReplaceTempView("hospital_histogram")
display(spark.sql("select * from hospital_histogram order by Frequency desc limit 10"))
```



	Hospital	Frequency
1	LLC Smith	44
2	Ltd Smith	39
3	Johnson PLC	38
4	Smith Ltd	37
5	Smith PLC	36
6	Smith Group	36
7	Johnson Inc	35
8	Smith Inc	34
9	Smith LLC	32
10	Group Smith	32

Analysis:

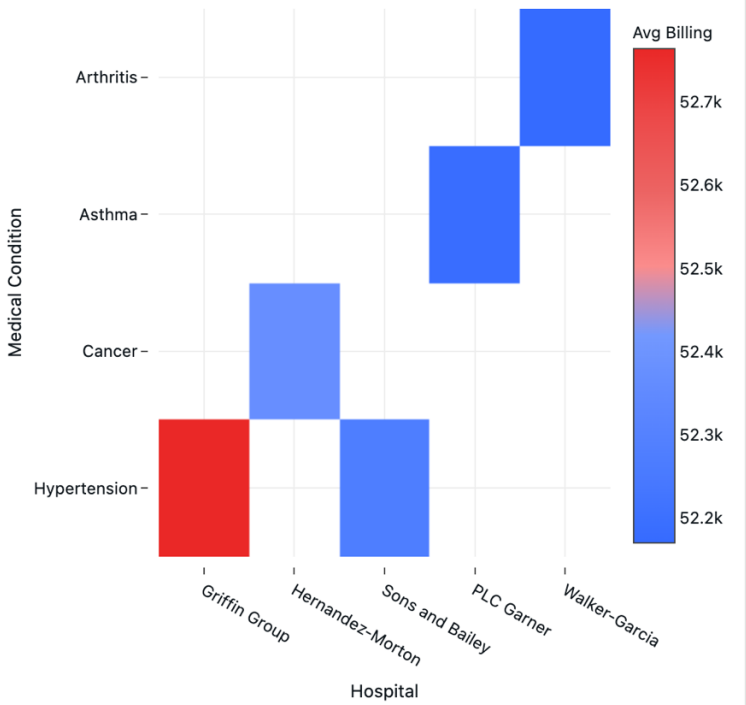
- LLC Smith dominates the top 10 hospitals by admissions, reflecting a significant market presence.
- Admissions are relatively balanced across the top 10, with a narrow range between 32 and 44.

20

```
#Creating a temporary view for running SQL queries
df.createOrReplaceTempView("medical_data")
```

21

```
#Using Spark SQL to find the top 10 hospitals with the highest average billing by medical coniditon
display(spark.sql("SELECT `Medical Condition`, Hospital, AVG(`Billing Amount`) AS Avg_Billing FROM medical_data GROUP BY `
```



	Medical Condition	Hospital	Avg_Billing
1	Hypertension	Griffin Group	52764.276736
2	Cancer	Hernandez-Mort...	52373.032374
3	Hypertension	Sons and Bailey	52271.66374
4	Asthma	PLC Garner	52181.8377923
5	Arthritis	Walker-Garcia	52170.036853

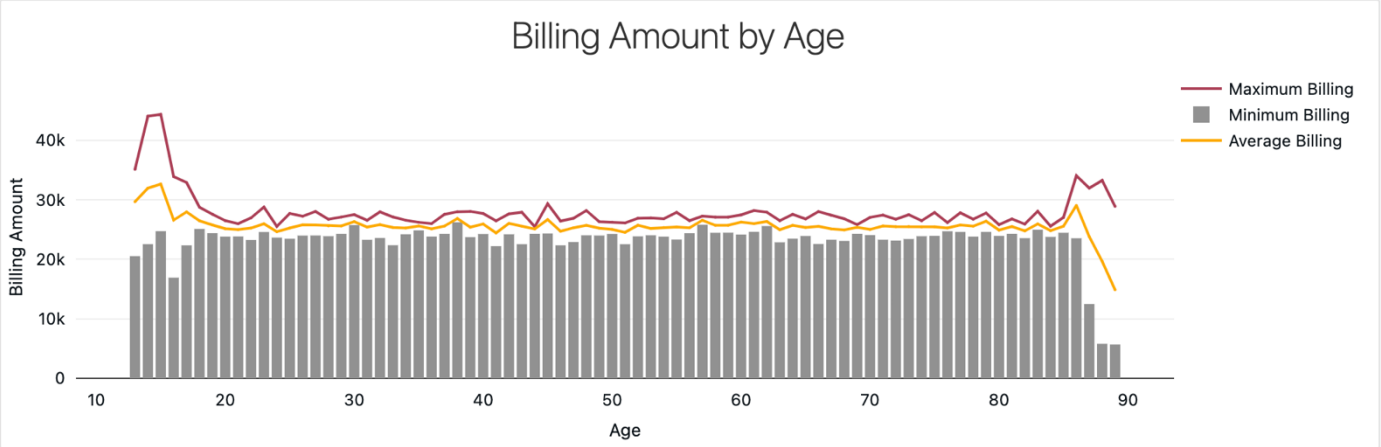
Analysis:

- Griffin Group has the highest average billing for Hypertension at 52,764.28.
- Hernandez-Morton ranks second with Cancer treatments averaging 52,373.03.
- Sons and Bailey comes in third with Hypertension treatments averaging 52,271.66.
- PLC Garner is fourth, with an average billing for Asthma at 52,181.84.
- Walker-Garcia is fifth, having an average billing for Arthritis at 52,170.04.

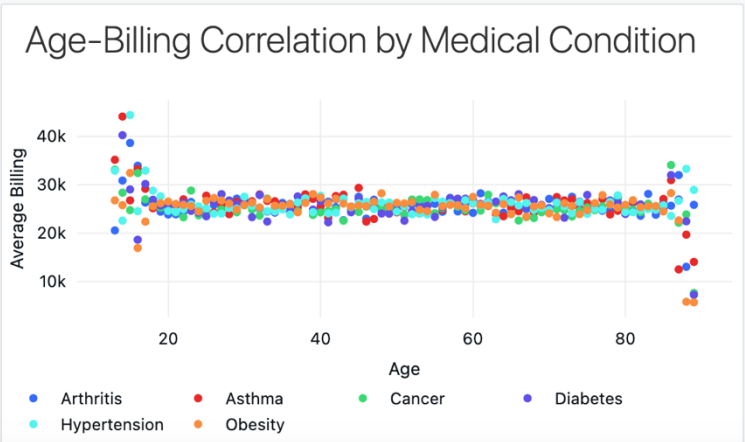
These findings highlight that Hypertension and Cancer are among the most expensive conditions to treat across different hospitals.

23

```
%sql
SELECT * from
(select Age, `Medical Condition`, AVG(`Billing Amount`) as Avg_Billing
FROM medical_data
GROUP BY Age, `Medical Condition`
) subquery
ORDER BY Avg_Billing DESC;
```



	Age	Medical Condition	Avg_Billing
1	15	Hypertension	44389.8148672875
2	14	Asthma	44079.14582336665
3	14	Diabetes	40229.17807496129
4	15	Arthritis	38629.0420109481
5	13	Asthma	35138.8030769479
6	86	Cancer	34062.61328021160
7	16	Arthritis	33914.41708901
8	88	Hypertension	33273.8991729484
9	16	Asthma	33232.7525928473
10	13	Cancer	33125.6981138031
11	17	Hypertension	32924.5144074835
12	13	Hypertension	32900.0444781830



Analysis:

1. Average, Minimum, and Maximum Billing Amount by Age:

- Age 15 shows the highest maximum billing amount of 44,389.8, the highest minimum billing of 24,743.6, and the highest average billing of 32,660.
- Average billing generally decreases and stabilizes within the range of 24,000 to 26,000 across most ages.
- At age 86, there is a slight increase in the average billing amount to 29,073.1.
- The lowest average billing amount is observed at age 89, at 14,892.

2. Age-Billing Correlation by Medical Condition:

There is no apparent correlation between age and billing amount across different medical conditions, indicating that billing is not strongly influenced by age for specific conditions.

Conclusion:

The analysis reveals key insights into the healthcare dataset, including billing trends, the prevalence of medical conditions and how these factors vary across different hospitals. The visualizations and SQL queries provide a clear understanding of the data, enabling data-driven decision-making for healthcare management and policy formulation.