A PROJECT ON HOSPITAL EMERGENCY ROOM (ER) OPERATIONS OPTIMIZATION: A DATA-DRIVEN BUSINESS INTELLIGENCE SOLUTION

BY

GLADYS MUNYAZI DALLA

ABSTRACT

The Hospital Emergency Visits Business Intelligence Project aims to provide actionable insights into hospital emergency room operations by analyzing patient visit data. Using an end-to-end approach, this project leverages Python, SQL Server, and Power BI to transform raw data into meaningful visualizations that highlight key performance indicators (KPIs) such as patient wait times, satisfaction levels, admission rates, and department trends. This analysis not only enhances operational efficiency but also informs decision-making, ultimately improving patient care and resource allocation. A collaborative dataset sourced from Data.World serves as the foundation, providing detailed information about demographics, visit patterns, and departmental referrals. By designing and implementing a robust ETL pipeline and integrating data into a star schema database, the project ensures high-quality, scalable storage and visualization capabilities. The scope of the project acknowledges limitations such as sample size and demographic biases, but it emphasizes continuous data validation and a scalable BI infrastructure to address these challenges over time. Through this initiative, the project bridges the gap between raw data and actionable insights, delivering measurable value to healthcare stakeholders.

Table of Contents

INTRODUCTION

In the rapidly evolving landscape of healthcare delivery, emergency rooms (ERs) stand as critical frontlines of patient care, facing unprecedented challenges of operational complexity, resource constraints, and increasing patient expectations. The intersection of technological innovation and healthcare management presents a unique opportunity to revolutionize emergency room operations through advanced Business Intelligence (BI) solutions.

Modern healthcare institutions are increasingly recognizing that operational efficiency is not just about reducing costs, but about enhancing patient experience, optimizing resource allocation, and delivering high-quality care during critical moments. Emergency rooms, characterized by their unpredictable nature and high-stakes environment, represent a particularly complex ecosystem where data-driven insights can make a transformative difference.

This project emerges from the pressing need to address systemic challenges that have long plagued emergency room operations: prolonged wait times, inconsistent patient satisfaction, inefficient resource allocation, and limited understanding of patient demographics and referral patterns. By leveraging sophisticated data analytics and Business Intelligence technologies, we aim to convert raw operational data into strategic intelligence that can fundamentally reshape how emergency rooms function. The project seeks to empower healthcare administrators and medical professionals with unprecedented insights into their operational dynamics ultimately leading to enhanced patient experiences, more efficient operational processes, and a more responsive healthcare ecosystem.

1.1. Project Title

Hospital Emergency Room (ER) Operations Optimization: A Data-Driven Business Intelligence Solution

1.2. Project Objective

1.2.1 General Objective

To develop a comprehensive Business Intelligence (BI) solution that transforms hospital emergency room operations through advanced data analytics, enabling data-driven decision-making, improving operational efficiency, and enhancing patient care by leveraging integrated technologies and sophisticated analytical methodologies.

1.2.2 Specific Objectives

i. Data Integration and Management
   Develop a robust data pipeline that extracts, transforms, and loads emergency room visit data with high precision, ensuring comprehensive and reliable dataset preparation.
ii. Performance Analysis and Insights
   Conduct in-depth analysis of key performance indicators to identify operational bottlenecks, inefficiencies, and critical patterns in emergency room processes and patient interactions.
iii. Technological Integration
   Create an integrated analytical ecosystem using Python, SQL Server, and Power BI to transform complex healthcare data into interactive, actionable visualization tools.

iv. Stakeholder Empowerment
    Equip healthcare administrators with sophisticated decision-support tools that enable real-time performance monitoring and strategic resource allocation.
v. Continuous Improvement
    Establish a systematic, adaptive framework for ongoing operational assessment that supports predictive analytics and continuous organizational learning.
vi. Organizational Impact
    Drive meaningful improvements in patient experience and operational efficiency by leveraging data-driven insights to optimize emergency room performance and service delivery.

1.3. Business Case Definition

The business case centers on addressing critical challenges within hospital emergency room operations, specifically targeting inefficiencies in patient flow, satisfaction tracking, and resource allocation. By implementing a sophisticated BI solution, the project seeks to convert raw operational data into strategic intelligence that can drive meaningful improvements in healthcare service delivery. The solution will focus on analyzing patient demographics, wait times, satisfaction scores, and departmental referral patterns to identify opportunities for optimization and enhance overall patient care.

1.4. KPIs

Key Performance Indicators (KPIs) serve as critical metrics that transform raw operational data into meaningful insights. These carefully selected indicators provide a comprehensive view of emergency room performance, enabling healthcare administrators to make data-driven decisions and continuously improve patient care.

1. Average Patient Wait Time

Patient wait time represents a crucial metric in emergency room efficiency, measuring the duration from patient arrival to initial medical assessment. This KPI is calculated by computing the total wait time across all patients and dividing it by the total patient count, with calculations segmented by individual departments. The primary purpose is to optimize patient flow and systematically reduce waiting periods, ultimately improving patient experience and operational responsiveness. By analyzing wait times through the departmental lenses, healthcare administrators can identify specific bottlenecks and develop targeted strategies for improvement.

2. Patient Satisfaction Score

Patient satisfaction emerges as a critical indicator of service quality and overall patient experience. This KPI is determined by summing all satisfaction ratings and dividing by the number of rated encounters, deliberately excluding non-rated interactions to ensure accuracy. The analysis is further segmented by demographic groups, such as age and gender, to provide deeper insights. The satisfaction score provides a quantitative assessment of patient perceptions regarding their emergency room experience. By tracking this metric, healthcare institutions can continuously assess and enhance the quality of care, identifying areas of excellence and opportunities for improvement.

3. Departmental Referral Distribution

Departmental referral distribution offers insights into the internal dynamics of emergency room operations by tracking the number and percentage of referrals across different departments. By calculating the proportion of referrals to each specific department relative to total referrals, administrators can identify potential bottlenecks, understand departmental load, and recognize inefficiencies in patient routing. This KPI serves as a critical tool for resource allocation, staff deployment, and operational optimization.

4. Patient Demographics Analysis

Patient demographics analysis provides a nuanced understanding of the emergency room's patient population by breaking down patient counts across gender, age groups, and racial categories. By calculating the percentage of patients in each demographic segment, healthcare administrators can develop targeted strategies that address the specific needs of diverse patient populations. This approach enables more personalized care, helps identify potential disparities in service delivery, and supports more inclusive healthcare practices.

5. Total Patient Volume

Total patient volume serves as a fundamental metric tracking the overall utilization of emergency room services. By counting patient visits over specific time frames—daily, weekly, monthly, or annually—administrators can gauge healthcare facility engagement and identify trends in patient attendance. This metric provides critical insights into operational capacity, potential seasonal variations, and overall community healthcare needs.

6. Administrative Flag Analysis

Administrative flag analysis offers a sophisticated approach to understanding intervention rates and potential systemic risks within emergency room operations. By calculating the percentage of patients with administrative flags, this KPI provides insights into the frequency of administrative interventions and helps identify underlying issues that may require strategic attention. This metric serves as an early warning system, enabling proactive management of potential operational challenges.

1.5 Scope and Limitations

The scope of this project involves analyzing emergency room visit data to gain insights into key performance metrics such as patient wait times and satisfaction scores. Leveraging tools like Python, SQL Server, and Power BI, the analysis focuses on a specific time period to ensure targeted and relevant findings. However, the project is subject to several limitations. The dataset used represents a small sample size, which may impact the generalizability of the results. Additionally, there is potential for demographic representation bias, as the sample may not fully capture the diversity of the population served by the emergency room. The analysis is also constrained by limited historical data, which can restrict the ability to identify long-term trends. Furthermore, inconsistencies in data collection practices may affect the accuracy and completeness of the dataset, posing challenges to drawing comprehensive conclusions.

SOURCE SYSTEMS

2.1 Identification of Source Systems

The source system for this project was a dataset hosted on Data.World, a collaborative data platform that provides easy access to structured and curated datasets. The specific dataset, titled "Hospital ER," served as the foundation for analyzing emergency room visits. This system was chosen due to its open accessibility, well-organized data, and compatibility with Python's datadotworld library, which enabled seamless extraction of data through API integration. The dataset included key information such as patient demographics, visit dates, department referrals, satisfaction scores, and admission statuses, all essential for building a comprehensive analytical model. The choice of Data.World as the source system ensured that the data was reliable, structured, and ready for downstream ETL processes. Moreover, the platform's ability to integrate directly with Python reduced the complexity of the extraction phase, setting a strong foundation for the project's success.

ETL (EXTRACT, TRANSFORM, LOAD) PROCESS

3.1 ETL Plan Overview

The ETL (Extract, Transform, Load) process for analyzing hospital emergency visits using Python, SQL Server, and Power BI involves the following steps. **Extraction** begins with using the Data.World API to download the dataset into a pandas DataFrame, ensuring all necessary fields are captured. During **Transformation**, the dataset is cleaned by removing personally identifiable information (PII) like patient initials and last names, standardizing categorical values (e.g., replacing "None" in department referrals with "Unknown"), and handling missing values by replacing nulls in satisfaction scores with the median. Additional transformations include converting the date column to a datetime format, extracting date parts (day, month, year, day of the week), and ensuring data consistency. Next, foreign keys are established by deduplicating and generating IDs for dimensions like patient demographics, referral departments, and dates. In the **Loading** phase, the transformed data is mapped to the schema designed in SQL Server, starting with populating dimension tables (patient_dim, date_dim, and department_dim), followed by populating the er_visits_fact table, linking all dimensions using foreign keys. Python's libraries such as pymssql or pyodbc are used to connect to SQL Server and execute the necessary insert queries. Finally, Power BI is connected to the SQL Server database to build interactive dashboards for visualizing key metrics like patient wait times, satisfaction scores, admission rates, and department trends.

3.2 Extraction Process

The extraction process begins by pulling data from the source, in this case, a hospital emergency visits dataset hosted on Data.World. Using Python and the Data.World API, the dataset is accessed programmatically, ensuring a seamless connection to the data source. The API fetches the dataset as a pandas DataFrame, allowing for easy manipulation and analysis. During this stage, it is important to confirm that all necessary fields are included, such as patient demographics, department referrals, wait times, satisfaction scores, and admission statuses. The extraction process also includes validation checks to ensure that the dataset contains no corrupt

or incomplete downloads. Any issues identified at this stage, such as missing files or invalid API tokens, are addressed before proceeding further.

3.3 Transformation Process

The transformation process involves refining and preparing the extracted data for integration into the SQL Server database and subsequent analysis. Exploratory Data Analysis (EDA) was conducted in the Python script to gain insights into the dataset. For instance, the patient satisfaction scores were analyzed, revealing that 6,699 out of 9,216 visits (approximately 73%) had no recorded satisfaction ratings. This significant proportion of missing values indicates a gap in data collection, potentially stemming from patients not providing feedback or the hospital not consistently collecting it. While missing values were not handled during the transformation process, this decision was intentional, as excluding imputation allows for a more transparent analysis of the available data, ensuring that the final results reflect actual collected responses rather than assumptions or estimates.

Further transformations included creating a new column, ` patient_fullName `, by combining the patient's first initial and last name to retain identification consistency while ensuring a cleaner schema. Subsequently, the original columns `patient_first_initial` and `patient_last_name` were dropped to minimize redundancy and simplify the dataset. These transformations ensured that personally identifiable information (PII) was removed, aligning with data privacy standards while preserving enough information to maintain unique patient identifiers.

Additionally, the `date` column was standardized by converting it to a datetime format, enabling the extraction of specific components such as day, month, year, and day of the week. These components were stored in a separate `date_dim` table to support time-based analyses in the relational schema. To integrate the cleaned and transformed dataset with the MySQL Server database, SQLAlchemy was used as the database connection tool. This Python library facilitated seamless interaction between the script and the database, allowing for efficient data loading through structured insert operations. These steps ensured that the dataset was fully prepared for analysis and visualization in Power BI while maintaining high data quality and integrity.

3.4 Loading Process

The final step of the ETL process involved transferring the transformed dataset to the MySQL Server database and creating a relational data schema to support analytical queries. Using SQLAlchemy, the transformed dataset was first uploaded into a staging table in the MySQL Server. An SQL script was then executed to define and create the necessary dimension and fact tables in the database. Dimension tables, including patient_dim, date_dim, and department_dim, were created to organize and store distinct information about patients, dates, and department referrals. The fact table, er_visits_fact, was designed to store transactional data, linking the dimension tables through foreign key relationships.

Once the schema was in place, the SQL script populated the dimension tables with data from the hospital_er dataset. The patient_dim table contained deduplicated patient records with unique IDs, while the date_dim table held date components extracted from the visit_date column. Similarly, the department_dim table was populated with unique department referrals, providing

an organized structure for department-related data. Finally, the er_visits_fact table was populated with transactional data, including foreign keys referencing the dimension tables and measures such as patient satisfaction scores and wait times. This approach ensured a clean separation of data, optimized for querying and analysis.

By structuring the data in a star schema, the MySQL Server database was prepared for direct integration with Power BI. This allowed for the creation of interactive dashboards to visualize key performance metrics such as patient wait times, admission rates, satisfaction levels, and department referral trends. The structured loading process ensured that the data was both analytically ready and easily accessible for stakeholders, providing a robust foundation for decision-making.

3.5 Tools and Technologies Used in ETL

The ETL process for analyzing hospital emergency visits utilized a range of tools and technologies to ensure efficient and reliable data extraction, transformation, and loading. Python served as the primary scripting language, with libraries such as pandas for data manipulation and datadotworld for extracting the dataset via API. Exploratory Data Analysis (EDA) and data transformations, including cleaning, standardizing, and restructuring the dataset, were performed in Python. SQLAlchemy was employed to establish a seamless connection between Python and the MySQL Server database, enabling the transfer of transformed data into the database. In the loading phase, MySQL Server was used as the relational database management system, where SQL scripts were executed to create dimension and fact tables and populate them with data. Finally, Power BI was connected to the MySQL Server to build interactive dashboards for visualizing key metrics and insights. These tools and technologies provided a robust and scalable ETL pipeline, facilitating data-driven decision-making.

DATA STORAGE

4.1 Selection of Data Storage Solution

The selection of MySQL Server as the data storage solution for this project was driven by its scalability, performance, and compatibility with the tools used in the ETL process. MySQL Server is a widely used relational database management system known for its robust support for structured data and efficient handling of large datasets. Its ability to enforce relationships through foreign keys made it ideal for implementing the star schema design, which is essential for analytical purposes. Additionally, MySQL Server's compatibility with Python through libraries such as SQLAlchemy ensured a seamless integration during the data loading phase. Its reliability and support for SQL queries allowed for efficient storage and retrieval of data, making it well-suited for connecting with Power BI to build interactive dashboards. The choice of MySQL Server ensured a cost-effective, versatile, and high-performance data storage solution that met the analytical requirements of the project.

4.2 Database Structure

The database structure for the hospital emergency visits project was designed using a star schema, optimized for analytical queries and business intelligence reporting. The schema consists of one central fact table, er_visits_fact, and multiple supporting dimension tables, including patient_dim, date_dim, and department_dim. The fact table serves as the core of the

database, storing transactional data such as visit counts, satisfaction scores, wait times, and references to the dimension tables via foreign keys. The dimension tables provide detailed contextual information, with patient_dim capturing demographic details of patients, date_dim organizing visit dates into components such as day, month, and year, and department_dim describing referral departments. This structure ensures data normalization within the dimensions while maintaining simplicity in the fact table for quick aggregation and querying. The relational design supports efficient data retrieval and integration with Power BI, allowing stakeholders to analyze trends and performance metrics across different dimensions of the hospital's emergency visit operations.

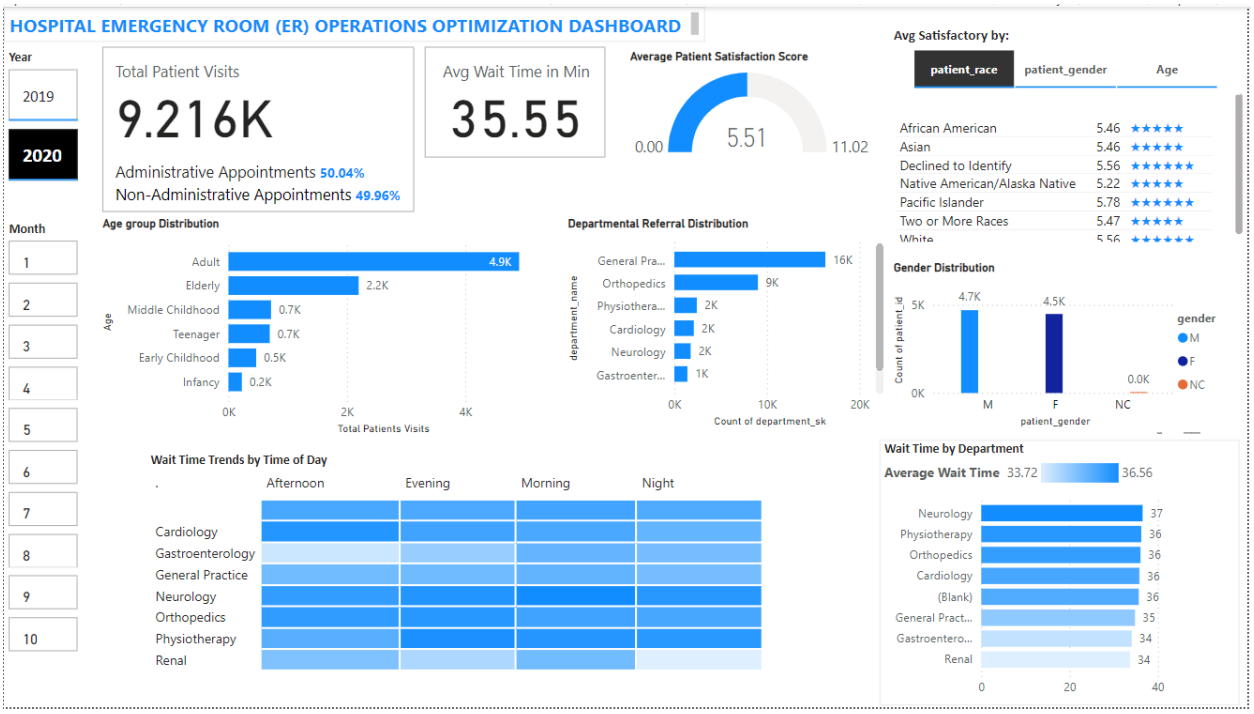
DATA VISUALIZATION

5.1 Choice of Data Visualization Tool

Power BI was chosen as the data visualization tool for this project due to its powerful capabilities in creating interactive, visually appealing, and insightful dashboards. Its seamless integration with MySQL Server allowed direct connectivity to the structured database, ensuring real-time data access and reducing manual intervention. Power BI's drag-and-drop interface made it intuitive to design complex visualizations, such as bar charts, heatmaps, and trend lines, to analyze key metrics like patient wait times, satisfaction ratings, and department performance. Its ability to handle large datasets efficiently ensured smooth performance, even when querying thousands of records. Additionally, Power BI's robust filtering and drill-down features enabled stakeholders to explore data at both high-level summaries and granular details, enhancing decision-making. The availability of cloud-based sharing through Power BI Service further streamlined collaboration, making it an ideal choice for presenting hospital emergency visit analytics to key stakeholders.

5.2 Report Design /Dashboard Development.

The project focuses on designing a comprehensive Emergency Room Performance Dashboard in Power BI to analyze key performance indicators (KPIs) and provide actionable insights for healthcare administrators. The dashboard incorporates intuitive visualizations to track critical metrics, such as Average Patient Wait Time, Patient Satisfaction Scores, Departmental Referral Distribution, Total Patient Volume, and Patient Demographics Analysis. By leveraging interactive charts, including bar graphs, tables, matrix, line charts, and slicers, the dashboard ensures seamless exploration of data across departments and demographic groups. Thoughtful design elements, such as card visuals for summarizing KPIs, multi-row cards for percentages, and slicers for interactivity, empower users to filter and drill down into specific data segments. The layout prioritizes clarity and ease of interpretation, with a focus on delivering both high-level summaries and granular insights. The use of calculated measures and DAX formulas ensures accuracy in aggregating patient counts, wait times, satisfaction scores, and demographic breakdowns. This well-structured dashboard serves as a decision-making tool, helping administrators optimize patient flow, identify bottlenecks, allocate resources effectively, and enhance overall emergency room efficiency and patient experience.



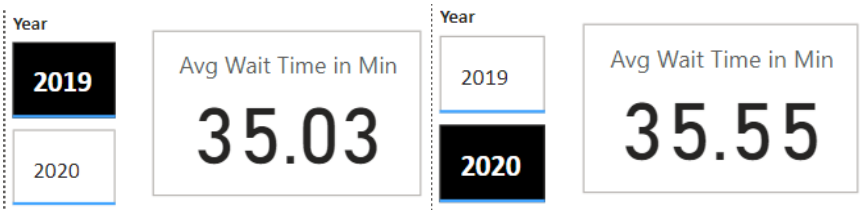
CONCLUSION AND SUMMARY

Data Visualization with annotations and summary

6.1. Key Findings

1. Average Wait Time findings:

a) An increase in average wait times from 35.03 minutes in 2019 to 35.55 in 2020 indicating a slight decline in operational efficiency, potentially due to higher patient volumes, resource constraints, or workflow bottlenecks. This trend highlights the need for targeted strategies to optimize patient flow, improve staff allocation, and address factors contributing to delays in the emergency room.

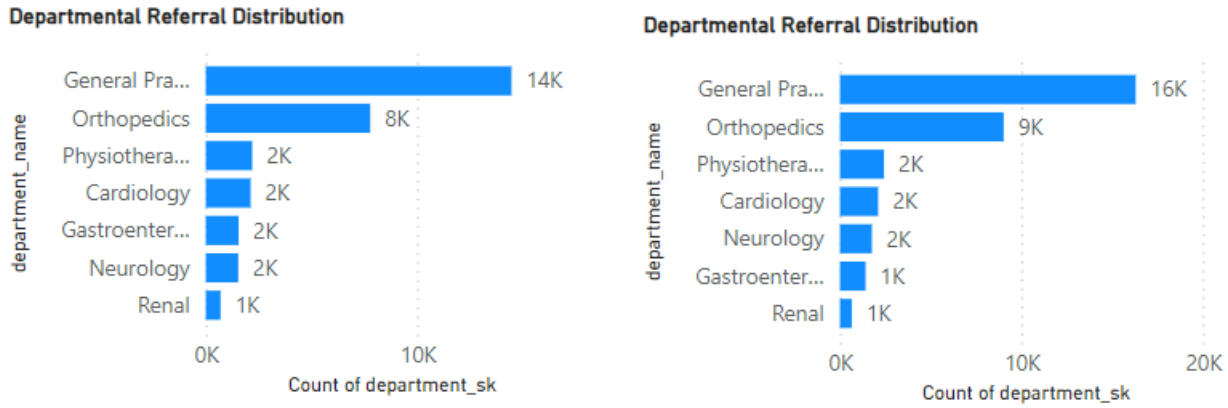


b) The average wait time by department decreased from 33.81 minutes in 2019 to 33.72 minutes in 2020. The Gastroenterology department, which recorded the highest average wait time of 37 minutes in 2019, showed notable improvement, reducing its wait time to 34 minutes in 2020. Conversely, the Neurology department experienced an increase in average wait time, rising from 36 minutes in 2019 to 37 minutes in 2020, highlighting the need for an operational review and resource optimization to address inefficiencies.

2. Patient Satisfaction Scores findings

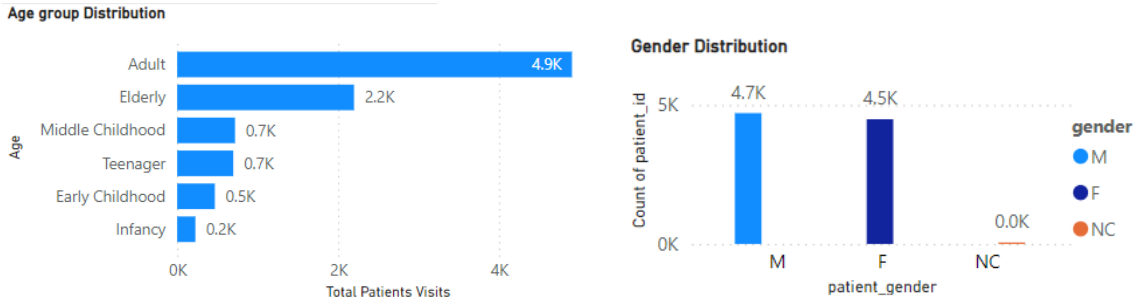
The Patient Satisfaction Score increased from 5.42 in 2019 to 5.51 in 2020, indicating a slight improvement in overall patient satisfaction. However, this positive trend is accompanied by a significant data gap: 6,699 out of 9,216 visits (approximately 73%) had no recorded satisfaction ratings. The lack of feedback from such a large percentage of visits presents a challenge in accurately assessing the true patient experience and identifying areas for improvement.

3. Departmental Referral Distribution findings



The General Practice department saw a 14% increase in referrals, rising from 14K in 2019 to 16K in 2020, while the Orthopedics department experienced a 12.5% increase, growing from 8K in 2019 to 9K in 2020. Cardiology, Physiotherapy, and Neurology departments remained stable at 2K referrals each for both years, showing no significant change in demand.

4. Patient Demographics Analysis



The Age Group Distribution graph reveals that the majority of patients fall into the Adult age group, accounting for 4.8% of the total. The Elderly and Middle Childhood age groups make up the next largest proportions, with 2.2% and 0.7% respectively. The Teenager, Early Childhood, and Infancy age groups have the smallest percentages, all below 0.2%. Turning to the Gender Distribution, the data shows that the patient population is predominantly Male, making up 5.0% of the total. Females account for 4.7% of the patients. There is a very small percentage (0.0%) of patients with an unspecified or non-conforming gender.

5. Administrative vs. Non-Administrative Appointments

The findings indicate a near-equal distribution between Administrative and Non-Administrative Appointments, with 50.04% of appointments categorized as Administrative and 49.96% as Non-Administrative. This balanced split suggests that both types of appointments are nearly equally represented, highlighting the importance of both operational functions in the organization.

6.2 Insights for Decision-Making

The gap in the Patient Satisfaction Score data collection highlights the need for improvements in the feedback collection process. Strategies such as simplifying the feedback process, increasing patient awareness, or implementing automated systems to prompt feedback could help capture a more representative sample of patient experiences. By addressing the missing data issue, the hospital could gain more accurate and actionable insights into patient satisfaction, enabling targeted improvements in service quality and patient care.

The notable increases in General Practice and Orthopedics reflect growing workloads in these departments, which may necessitate additional resources, staff, or process optimizations to effectively manage rising patient volumes. In contrast, the lack of significant change in smaller departments such as Cardiology, Gastroenterology, and Renal suggests consistent but lower patient demand, presenting opportunities for resource reallocation or targeted awareness programs to improve utilization and balance workloads across departments.

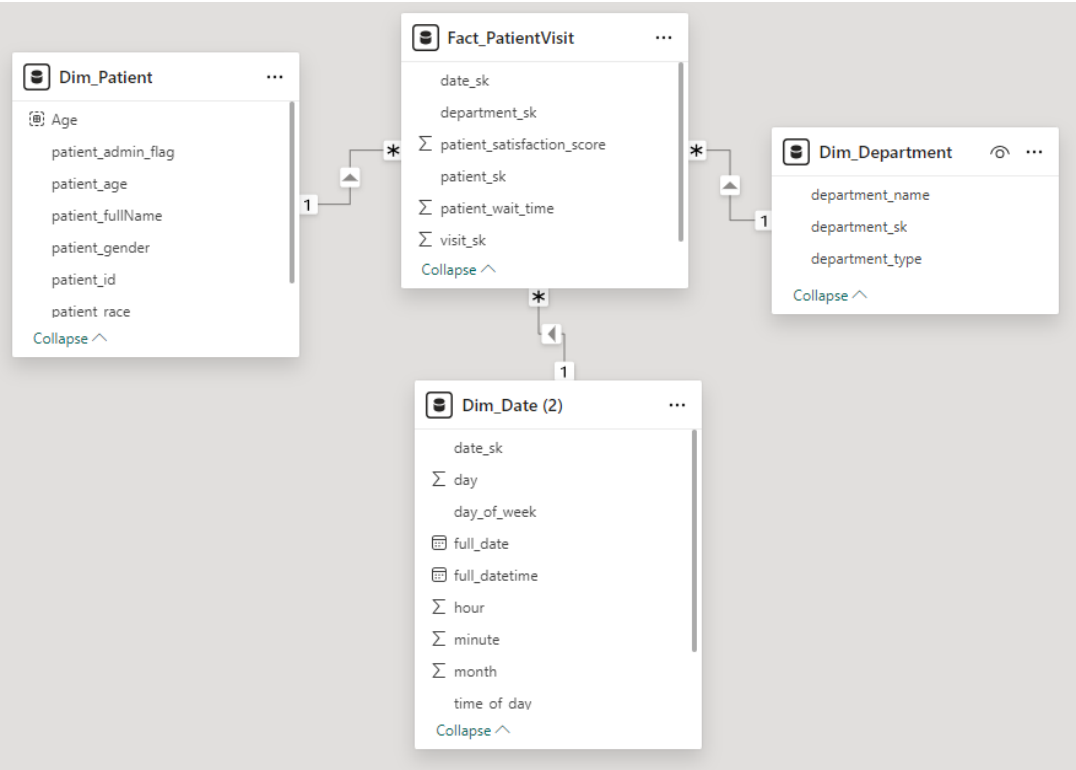
While the balance of administrative and non-administrative appointments may seem minor, its impact on hospital efficiency is substantial. Optimizing administrative processes not only shortens patient wait times but also contributes to better resource management, improved staff productivity, and enhanced patient satisfaction. Identifying and addressing inefficiencies in administrative tasks will have a ripple effect, enhancing the overall performance and quality of care within the hospital.

6.3 Reflection on Business Case Addressed

The business case successfully addresses key challenges in hospital emergency room operations by leveraging the insights derived from data analysis. Through the BI solution, critical inefficiencies in patient flow, satisfaction tracking, and resource allocation were identified and analyzed. Findings such as increased patient wait times in Neurology, improved efficiency in Gastroenterology, and rising referrals in General Practice and Orthopedics highlight areas requiring immediate attention. Additionally, the analysis of patient demographics sheds light on diverse patient needs, enabling more targeted strategies for equitable care delivery. The identification of gaps, such as the 73% missing satisfaction ratings, underscores opportunities to enhance data collection practices for more accurate performance measurement. Overall, these findings provide a solid foundation for hospital administrators to optimize processes, improve resource utilization, and ultimately elevate the quality of patient care in emergency room settings.

DOCUMENTATION

7.1 Data Modeling Diagram



7.2. Transformation Details

The transformation process involved refining and preparing the extracted data for integration into the SQL Server database and subsequent analysis. Key steps included:

1. Exploratory Data Analysis (EDA):
    - Patient satisfaction scores revealed 73% missing values (6,699 out of 9,216 visits), highlighting gaps in data collection. Missing values were intentionally left unhandled to ensure transparent analysis without imputation.
2. Data Cleaning and Column Refinement:
    - A new column, patient_fullName, was created by combining the first initial and last name to ensure identification consistency while meeting data privacy standards.
    - Redundant columns, patient_first_initial and patient_last_name, were dropped to simplify the schema and minimize redundancy.
3. Date Standardization:
    - The date column was standardized to datetime format, enabling the extraction of components like day, month, year, and day of the week. These were stored in a separate date_dim table to support time-based analysis.
4. Database Integration:
    - The Python SQLAlchemy library was used to facilitate efficient data loading into the SQL Server database through structured insert operations.

PROJECT IMPLEMENTATION CHALLENGES AND SOLUTIONS

During the implementation of the hospital emergency room BI project, several challenges were encountered, each requiring strategic solutions to ensure project success:

1. Data Quality Issues:
    - **Challenge:** A significant portion of the dataset, particularly patient satisfaction scores, had missing values (73% of visits had no recorded ratings). This posed a challenge in deriving reliable insights.
    - **Solution:** Instead of imputing missing values, which could introduce bias, the project team chose to analyze the available data transparently. This approach highlighted the data collection gap itself as a key finding, allowing stakeholders to focus on improving feedback collection processes.
2. Data Privacy and Security:
    - **Challenge:** Handling sensitive patient information such as names required adherence to strict privacy standards and compliance with data protection regulations.
    - **Solution:** Personally Identifiable Information (PII) was removed during the transformation phase. A new column, patient_fullName, was created using the first initial and last name to maintain unique identifiers without exposing sensitive data.
3. Integration of Large Datasets:
    - **Challenge:** Managing and integrating large volumes of emergency room data into the SQL Server database while ensuring efficiency and consistency was complex.
    - **Solution:** The SQLAlchemy library in Python was utilized for structured and seamless database connections, allowing efficient data loading operations. Batch insert methods were also used to optimize performance.
4. Standardizing Date and Time Data:
    - **Challenge:** The date column was inconsistently formatted, which could hinder time-based analysis.
    - **Solution:** The column was converted to a standardized datetime format, and specific time components (day, month, year, day of the week) were extracted and stored in a date_dim table. This transformation enabled precise time-based reporting in Power BI.
5. Complex Data Transformation Requirements:
    - **Challenge:** Transforming raw data into a clean and analysis-ready format required significant effort, especially when merging and dropping redundant columns.
    - **Solution:** Data cleaning processes were streamlined in Python, ensuring that unnecessary columns were removed while retaining key data attributes for analysis. This reduced schema complexity and enhanced overall data clarity.

FUTURE ENHANCEMENTS AND RECOMMENDATIONS

As the hospital emergency room BI project evolves, there are opportunities for future enhancements to further optimize operations, improve patient satisfaction, and ensure data-driven decision-making. Key recommendations include:

1. Improved Data Collection and Feedback Mechanisms:
    - **Enhancement:** Address the significant gap in patient satisfaction scores, where approximately 73% of visits lacked feedback. Implement automated feedback systems, such as digital surveys or follow-up emails, to encourage patient participation.
    - **Recommendation:** Incentivize patients to provide feedback and ensure data collection processes are consistently integrated into the patient discharge workflow
2. Resource Allocation Analysis:

- **Enhancement:** Develop advanced resource optimization models to identify underutilized departments (e.g., Cardiology, Renal) and allocate staff and resources more efficiently.
- **Recommendation:** Conduct periodic reviews of departmental workloads and refer patients to less burdened departments where appropriate to balance demand.

3. Patient Flow Analysis:

- **Enhancement:** Analyze patient flow from arrival to discharge to identify bottlenecks and delays in the emergency room process.
- **Recommendation:** Map the patient journey using process flow diagrams and optimize workflows to enhance throughput, reduce wait times, and improve overall efficiency.

REFERENCES

Gonçalves, C. T., Gonçalves, M. J. A., & Campante, M. I. (2023). Developing Integrated Performance Dashboards Visualisations Using Power BI as a Platform. *Information*, *14*(11), 614.

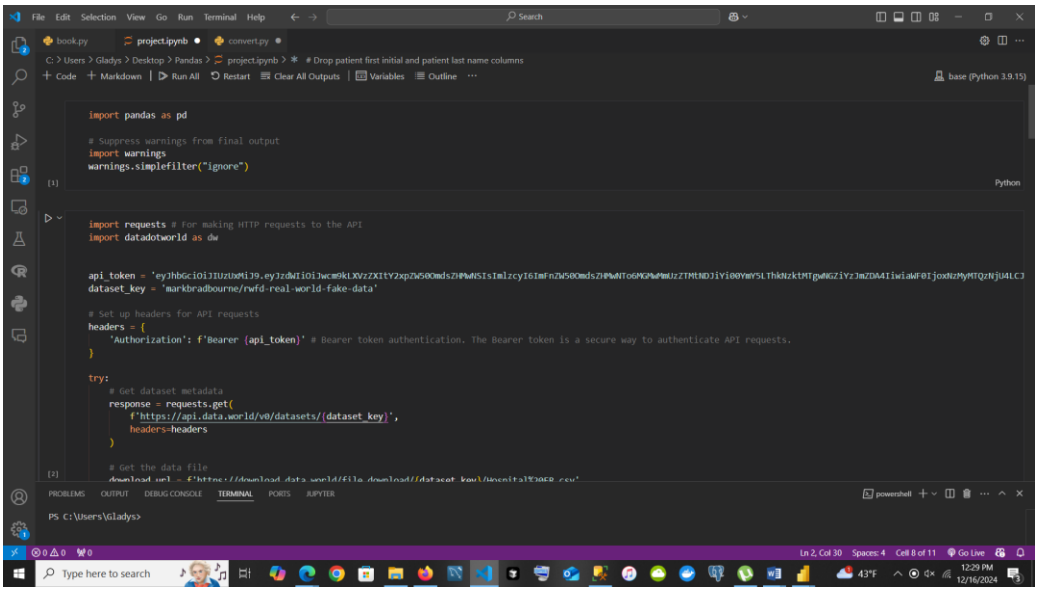
Lee, E. K., Atallah, H. Y., Wright, M. D., Post, E. T., Thomas IV, C., Wu, D. T., & Haley Jr, L. L. (2015). Transforming hospital emergency department workflow and patient care. Interfaces, 45(1), 58-82.

OpenAI. (2024). ChatGPT (Mar 14 version) [Large language model]. https://chat.openai.com/chat

Small, L. F. F. (2011). Determinants of physician utilization, emergency room use, and hospitalizations among populations with multiple health vulnerabilities. *Health:*, *15*(5), 491-516.

CODE SNIPPETS (IF APPLICABLE)

a. Snippet of python Jupiter notebook script used for extraction and transformation

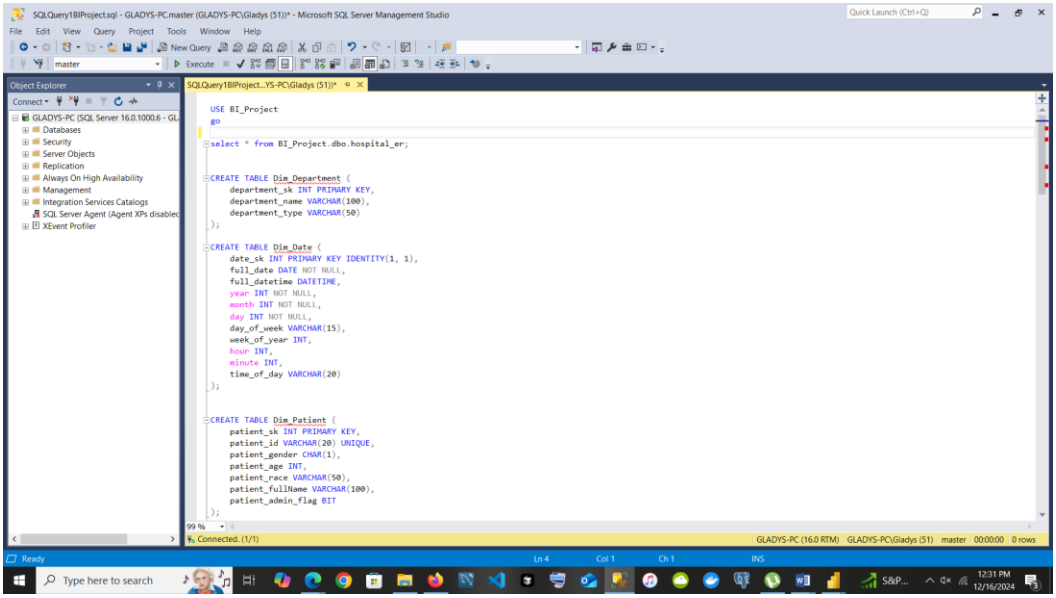


```
import pandas as pd

# Suppress warnings from final output
import warnings
warnings.simplefilter("ignore")
```

```
import requests # For making HTTP requests to the API
import datadotworld as dw


api_token = 'eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJwcm9kLXVzZXItY2xpZW50OmdsZHMwNSIsImlzcyI6ImFnZW50OmdsZHMwNTo6MGMwMWUzZTMtNDJiYi00YmY5LThkNzktMTgwNGZiYzJmZDA4IiwiaWF0IjoxNzMyMTQzNjU4LCJ
dataset_key = 'markbradbourne/rwfd-real-world-fake-data'

# Set up headers for API requests
headers = {
    'Authorization': f'Bearer {api_token}' # Bearer token authentication. The Bearer token is a secure way to authenticate API requests.
}

try:
    # Get dataset metadata
    response = requests.get(
        f'https://api.data.world/v0/datasets/{dataset_key}',
        headers=headers
    )

    # Get the data file
    download_url = f'https://download.data.world/file_download/{dataset_key}/Hospital%20ER.csv'
```

```
PS C:\Users\Gladys>
```

b. Snippet of SQL Server database as a staging area.



```
USE BI_Project
go

select * from BI_Project.dbo.hospital_er;


CREATE TABLE Dim_Department (
    department_sk INT PRIMARY KEY,
    department_name VARCHAR(100),
    department_type VARCHAR(50)
);

CREATE TABLE Dim_Date (
    date_sk INT PRIMARY KEY IDENTITY(1, 1),
    full_date DATE NOT NULL,
    full_datetime DATETIME,
    year INT NOT NULL,
    month INT NOT NULL,
    day INT NOT NULL,
    day_of_week VARCHAR(15),
    week_of_year INT,
    hour INT,
    minute INT,
    time_of_day VARCHAR(20)
);


CREATE TABLE Dim_Patient (
    patient_sk INT PRIMARY KEY,
    patient_id VARCHAR(20) UNIQUE,
    patient_gender CHAR(1),
    patient_age INT,
    patient_race VARCHAR(50),
    patient_fullName VARCHAR(100),
    patient_admin_flag BIT
);
```