Data Visualization of MIMIC Demo Dataset

JESUS MINJARES

UT AUSTIN: AI IN HEALTHCARE

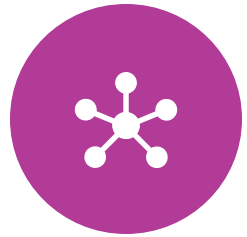
Overview

GRAPH 1: MOST COMMON MICROORGANISM IN BLOOD CULTURES

GRAPH 2: TREND OF PRESCRIPTIONS OVER TIME

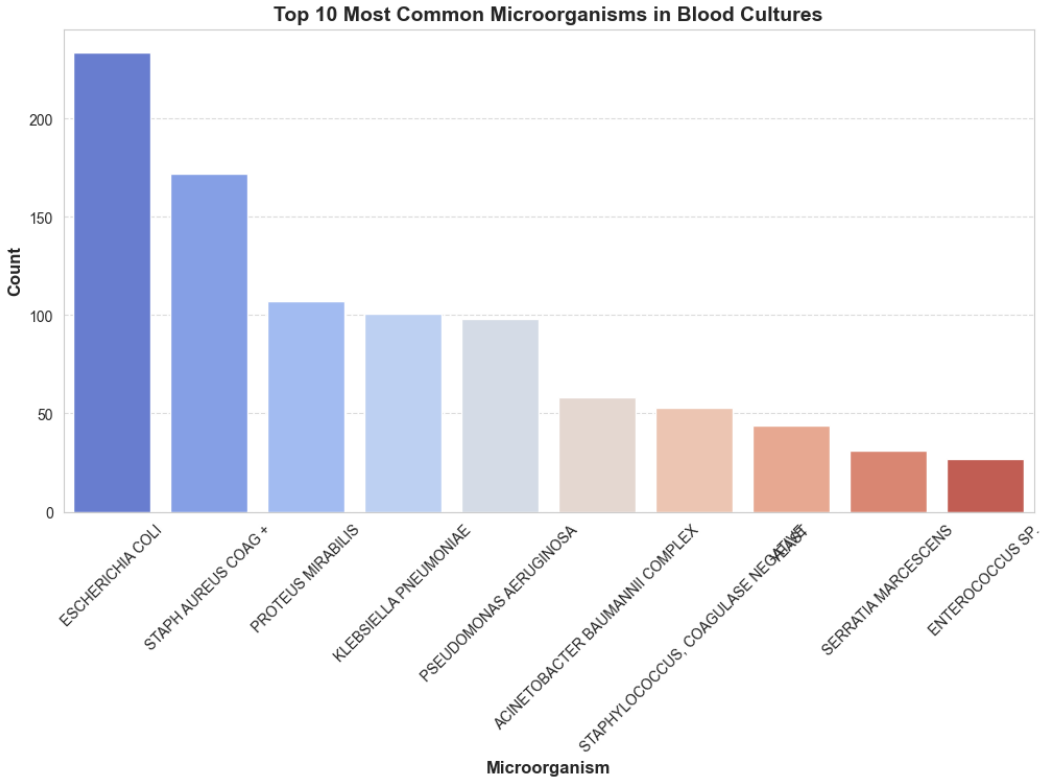
GRAPH 3: TOP 10 PRESCRIBED DRUGS

GRAPH 4: ANTIBIOTIC RESISTANCE PATTERNS



GRAPH 5: DIAGNOSIS-BASED PRESCRIPTION NETWORK GRAPH

Graph 1: Most Common Microorganisms in Blood Cultures



- This visualization highlights the most frequently detected microorganisms in hospital blood cultures. Identifying these microorganisms is critical for guiding treatment strategies and antibiotic stewardship. By analyzing this data, hospitals can proactively prepare for common infections and mitigate the spread of resistant bacteria.

1

Load the dataset: Read the **MICROBIOLOGYEVENTS.csv** file using *pandas*.

2

Process the data: Extract microorganism names and count their occurrences.

3

Select the top 10: Identify the **10** most frequently occurring microorganisms.

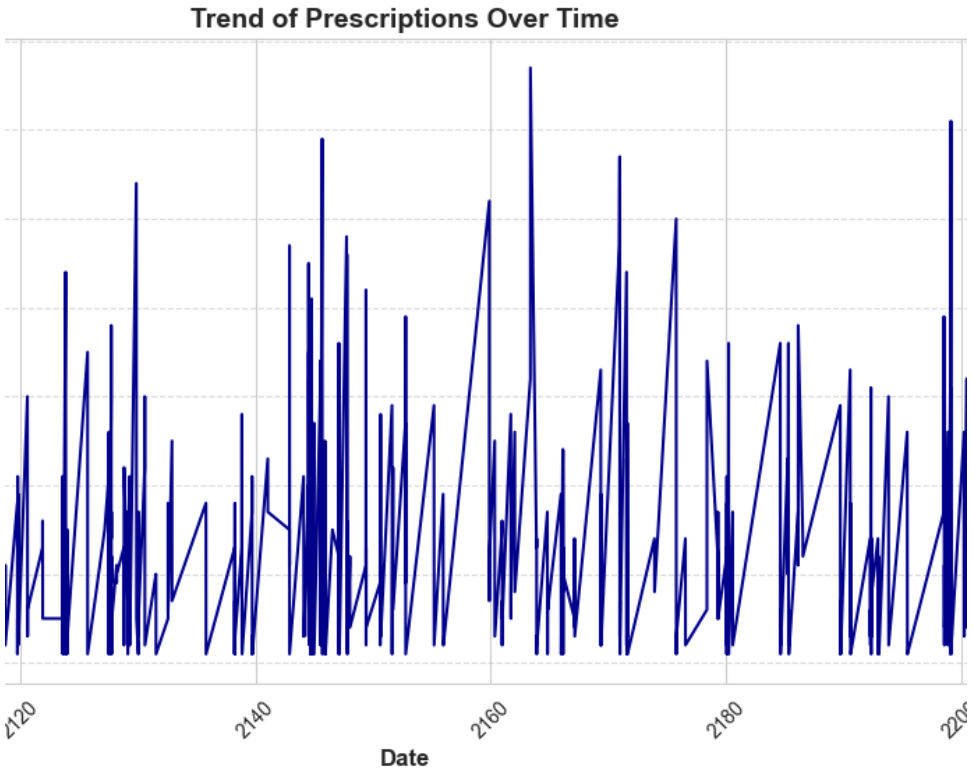
4

Generate the bar chart:

- Use sns.barplot() to plot the data.
- Set appropriate labels and a title.
- Rotate x-axis labels for readability.

Graph 1: Step by Step

GRAPH 2: TREND OF PRESCRIPTIONS OVER TIME



- This line graph presents the variation in prescription rates over time. Seasonal trends, public health interventions, and hospital protocols can influence these fluctuations. Understanding these trends assists in inventory management and policy adjustments to optimize patient care.

Graph 2: Step by Step

1

Load the dataset: Read the **PRESCRIPTIONS.csv** file.

2

Convert time format: Ensure the startdate column is in datetime format.

3

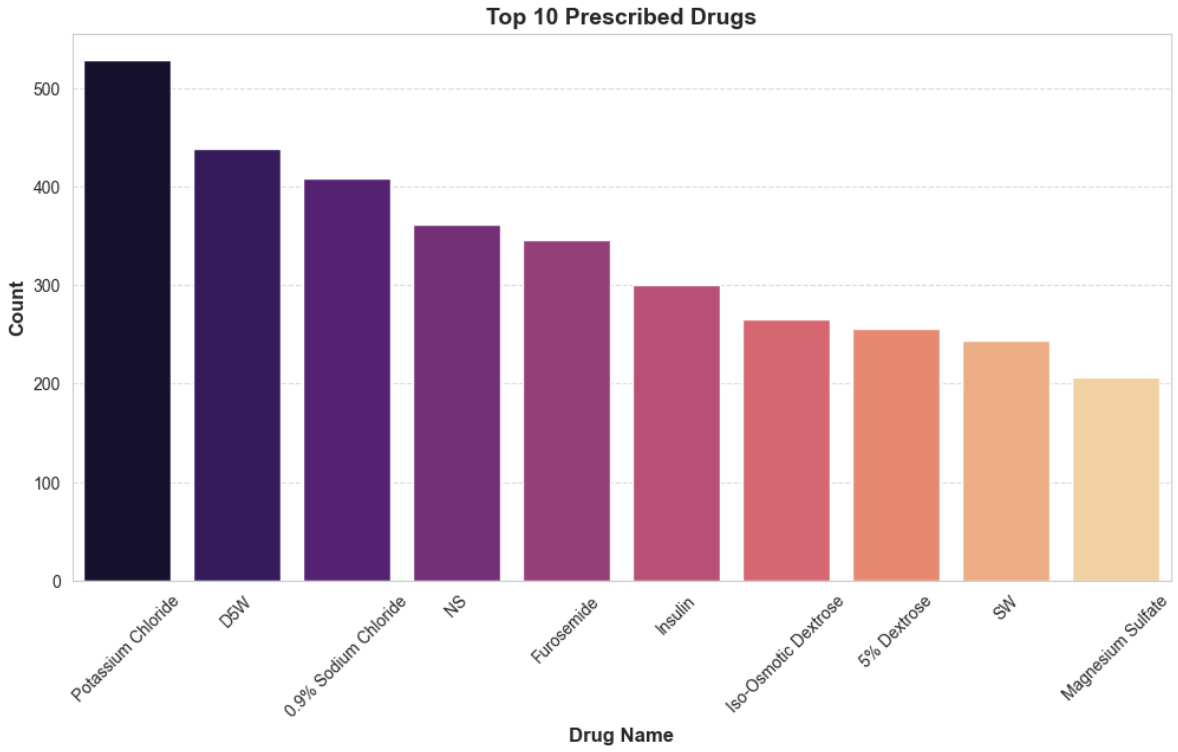
Aggregate prescription counts: Group data by date to compute daily prescription totals.

4

Generate the line plot:

- Use sns.lineplot() to plot the trend over time.
- Format the x-axis and y-axis labels appropriately.
- Rotate x-axis labels for clarity.

GRAPH 3: TOP 10 PRESCRIBED DRUGS



- This bar chart illustrates the most frequently prescribed medications. Identifying these drugs helps hospitals manage stock levels and analyze treatment patterns for prevalent conditions.

1

Load the dataset: Read the **PRESCRIPTIONS.csv** file.

2

Extract prescription data: Identify unique drug names and their frequency.

3

Select the top 10 drugs: Find the most frequently prescribed medications.

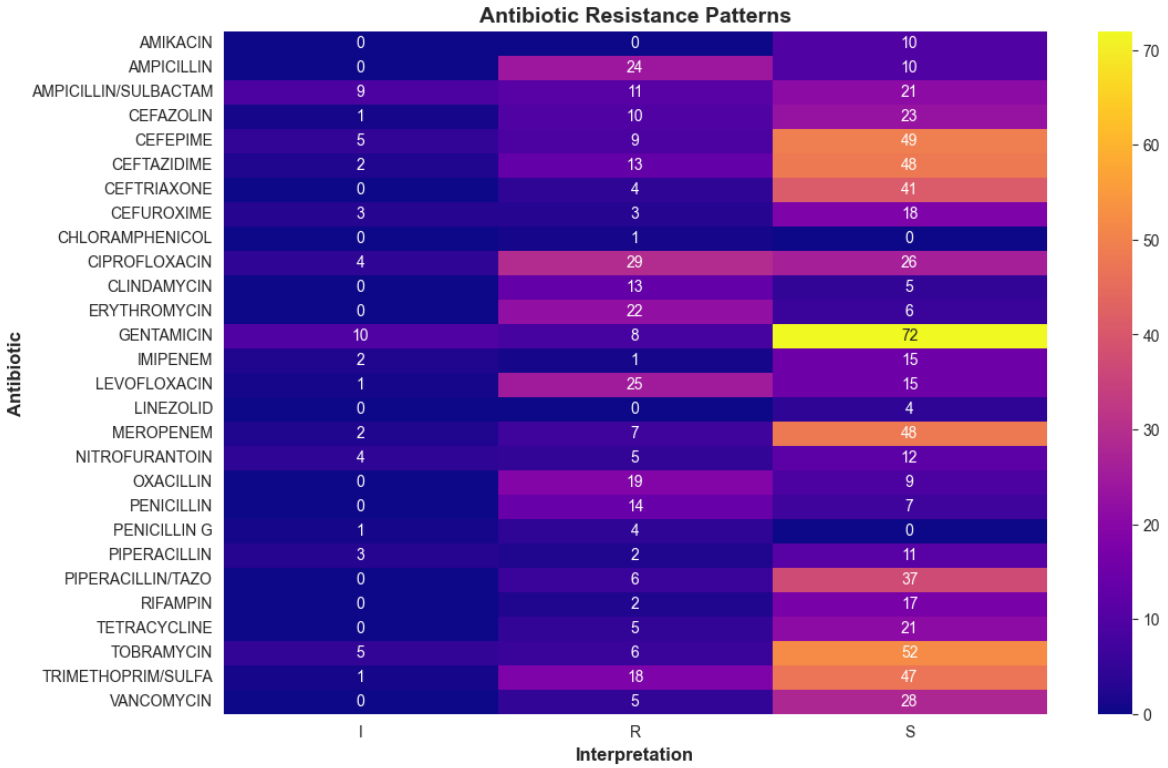
4

Generate the bar chart:

- Use sns.barplot() to plot the drug frequencies.
- Format the visualization with proper labels and title.
- Rotate x-axis labels for clarity.

Graph 3: Step by Step

Graph 4: Antibiotic Resistance Patterns



- This heatmap visualizes antibiotic resistance patterns by displaying interactions between different antibiotics and bacterial resistance interpretations. Hospitals use this data to optimize treatment protocols and mitigate antibiotic resistance

Graph 4: Step by Step

1

Load the dataset: Read the **MICROBIOLOGYEVENTS.csv** file.

2

Filter relevant columns: Keep only antibiotic names and resistance classifications.

3

Create a pivot table:

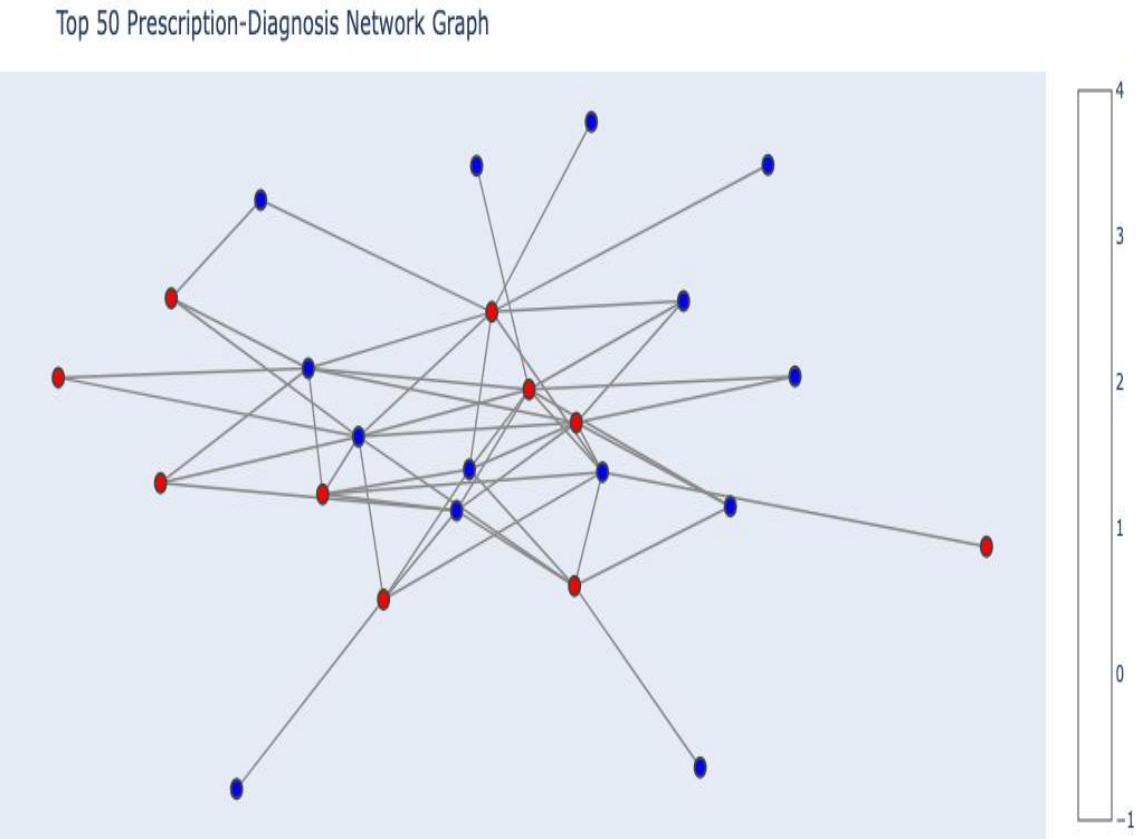
- Summarize resistance occurrences.
- Index by antibiotic name and classify resistance on columns.

4

Generate the heatmap:

- Use sns.heatmap() to visualize the data.
- Annotate the heatmap for better clarity.
- Apply an appropriate color scheme.

Graph 5: Diagnosis-Based Prescription Network Graph



- A network graph shows drug relationships, co-prescriptions, and interactions, highlighting common treatment patterns. Red nodes represent diagnoses, blue nodes are prescriptions, and thicker lines indicate stronger links. Some drugs treat many conditions, while others are for specific cases.

Graph 5: Step by Step

1

Load the dataset: Read the **PRESCRIPTIONS.csv** file using pandas.

2

Extract relevant data: Identify co-prescribed drugs within patient stays.

3

Create an adjacency list: Construct a list of drug pairs that are frequently prescribed together.

4

Generate the network graph:

- Use networkx to construct the graph structure.
- Apply nx.spring_layout() for node positioning.
- Use nx.draw() to visualize the connections.