MIMIC Visualization

AI IN HEALTHCARE-WB

Cameron Morrongiello

MIMIC Visualization

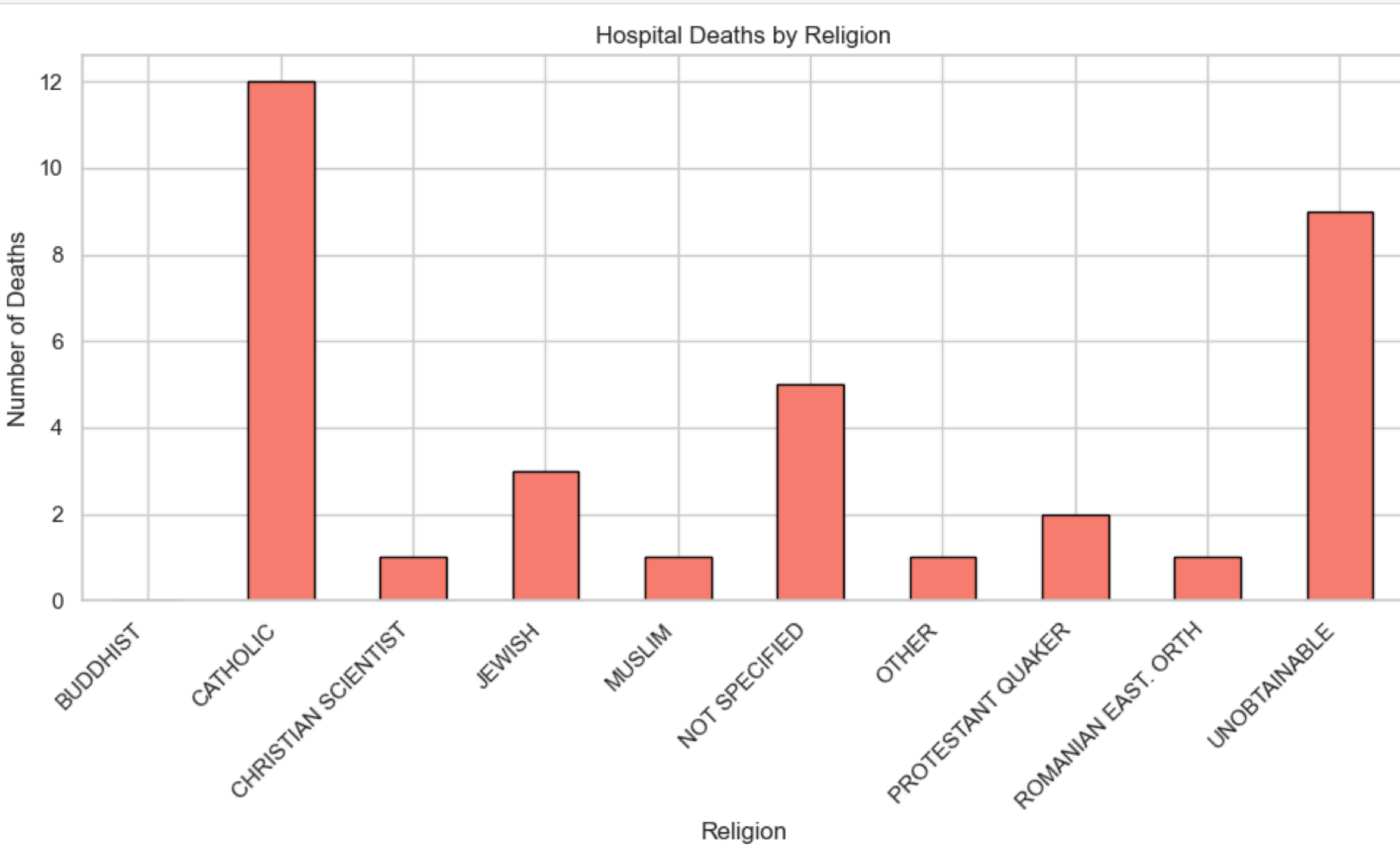
Areas of interest

- Religion
- Insurance
- Prescriptions
- Services

1. Hospital Deaths by Religion

Detailed Instructions

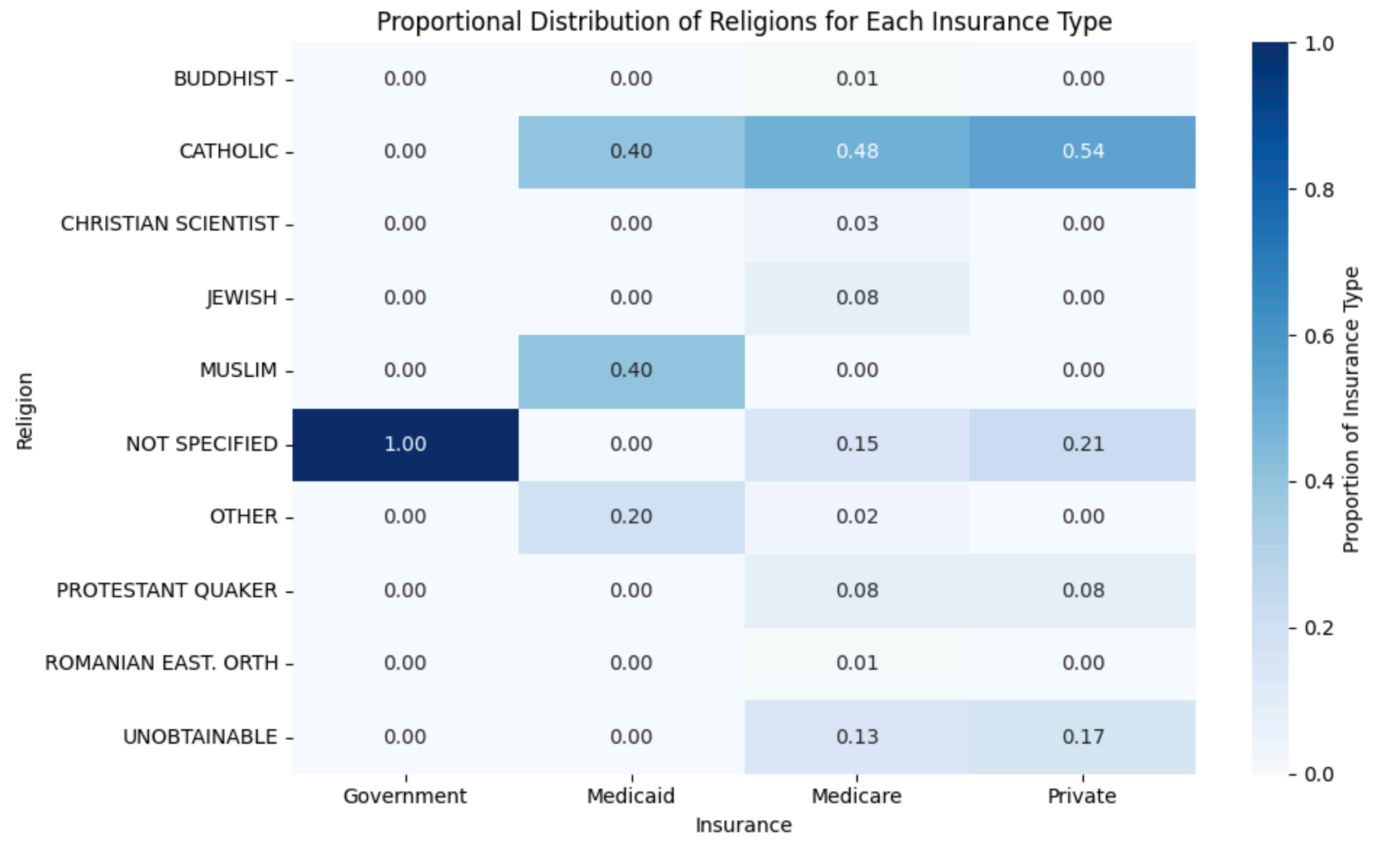
To create this visualization, I first grouped the data by the 'religion' column and calculated the counts of hospital deaths (indicated by the 'hospital_expire_flag') for each religion using `groupby()` and `value_counts()`. I then reshaped the result with `unstack()` to pivot the counts, filling missing values with zeros using `fill_value=0`. Next, I selected the column representing deaths (where `hospital_expire_flag` equals 1) and plotted it as a bar chart using `plot()`. I customized the chart's appearance with a 'salmon' color, black edges, and adjusted the figure size. I added labels for the title, x-axis, and y-axis, rotated the x-tick labels for readability, and applied `tight_layout()` to ensure the layout fits well. Finally, I displayed the chart with `plt.show()`.



2. Proportional Distribution of Religions for Each Insurance Type

Detailed Instructions

To create this visualization, I first grouped the data by both 'religion' and 'insurance' and calculated the size of each combination using groupby() and size(). I then reshaped the result with unstack() and filled any missing values with zeros using fill_value=0. Next, I normalized the data by dividing each value in the insurance_by_religion table by the sum of values along the columns, using div() to get the proportional distribution of religions for each insurance type. I visualized this normalized data as a heatmap with sns.heatmap(), adding annotations to display the proportions, customizing the color map to 'Blues', and formatting the numbers to two decimal places. I also included a color bar with a label indicating the proportion of insurance type. Finally, I added a title, labeled the axes, applied tight_layout() to optimize the layout, and displayed the plot with plt.show().

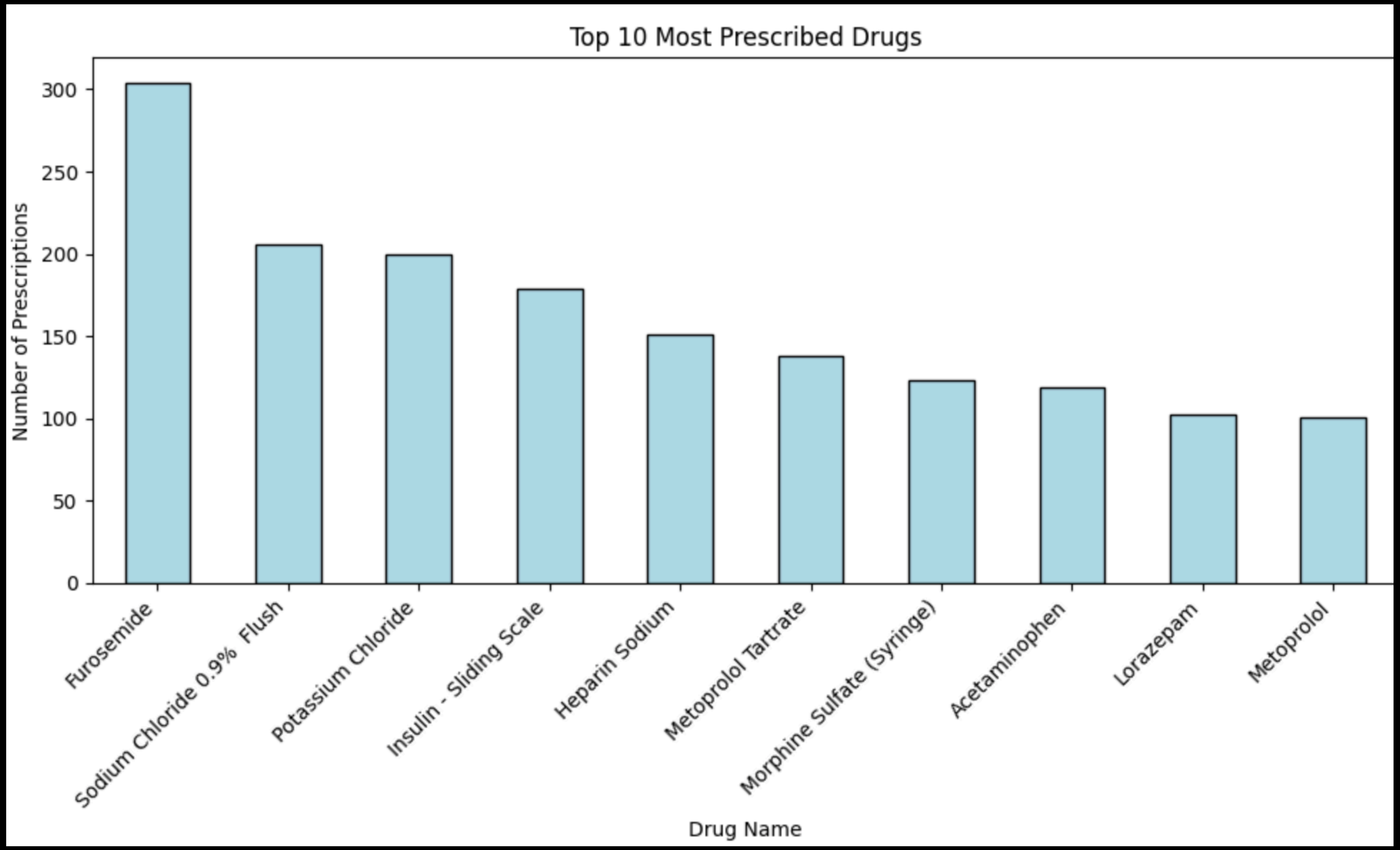


Religion	Government	Medicaid	Medicare	Private
BUDDHIST	0.00	0.00	0.01	0.00
CATHOLIC	0.00	0.40	0.48	0.54
CHRISTIAN SCIENTIST	0.00	0.00	0.03	0.00
JEWISH	0.00	0.00	0.08	0.00
MUSLIM	0.00	0.40	0.00	0.00
NOT SPECIFIED	1.00	0.00	0.15	0.21
OTHER	0.00	0.20	0.02	0.00
PROTESTANT QUAKER	0.00	0.00	0.08	0.08
ROMANIAN EAST. ORTH	0.00	0.00	0.01	0.00
UNOBTAINABLE	0.00	0.00	0.13	0.17

3. Top 10 Most Prescribed Drugs

Detailed Instructions

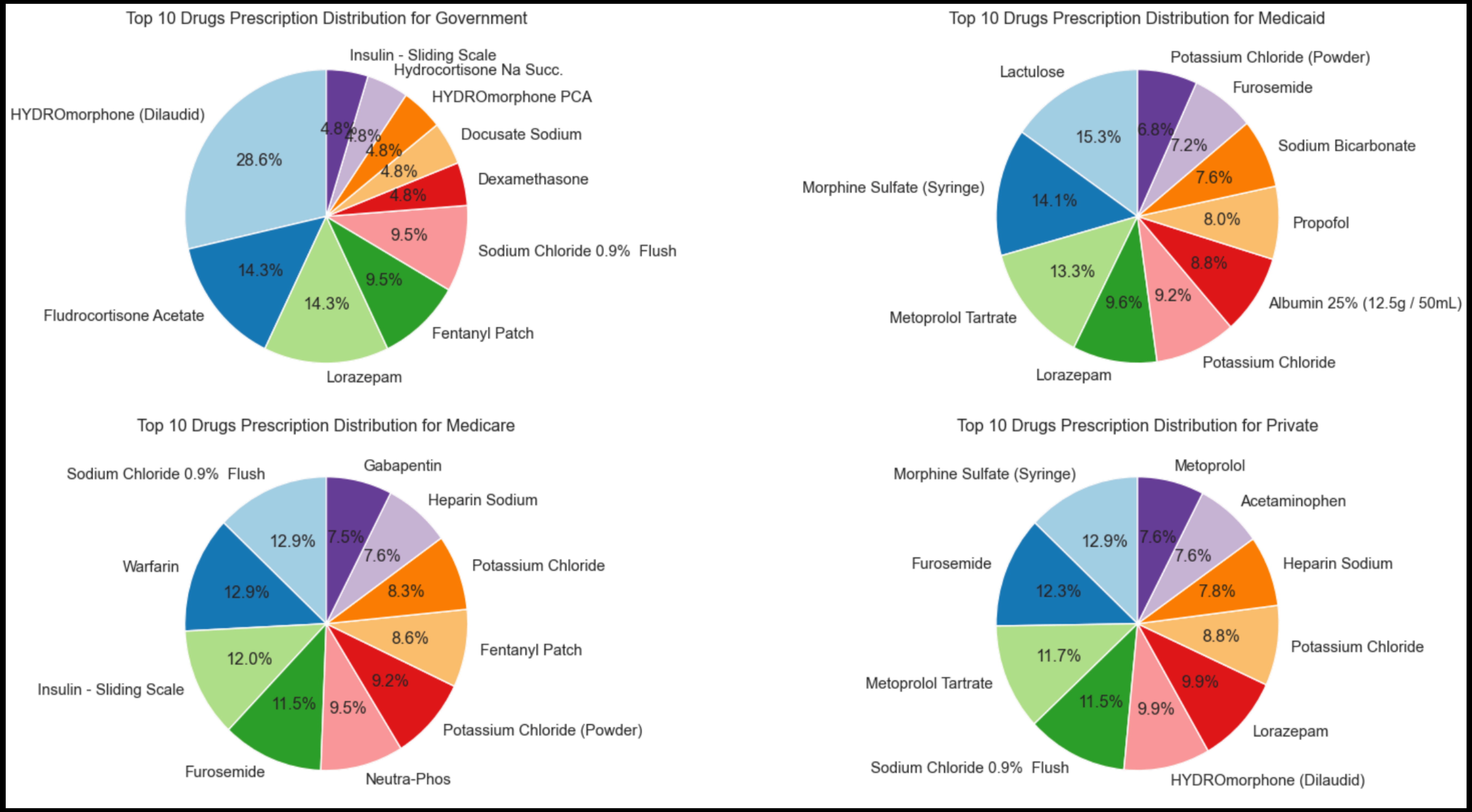
To create this visualization, I first calculated the top 10 most prescribed drugs by using `value_counts()` on the 'drug_name_generic' column and selecting the first 10 results with `head(10)`. I then plotted this data as a bar chart using the `plot()` function, specifying 'lightblue' for the bar color and black edges for contrast. I customized the chart with a title, axis labels, and rotated the x-axis tick labels for readability. To ensure the layout was properly adjusted, I used `tight_layout()`. Finally, I displayed the plot with `plt.show()`.



4. Top 10 Drugs Prescription Distribution per Insurance Type

Detailed Instructions

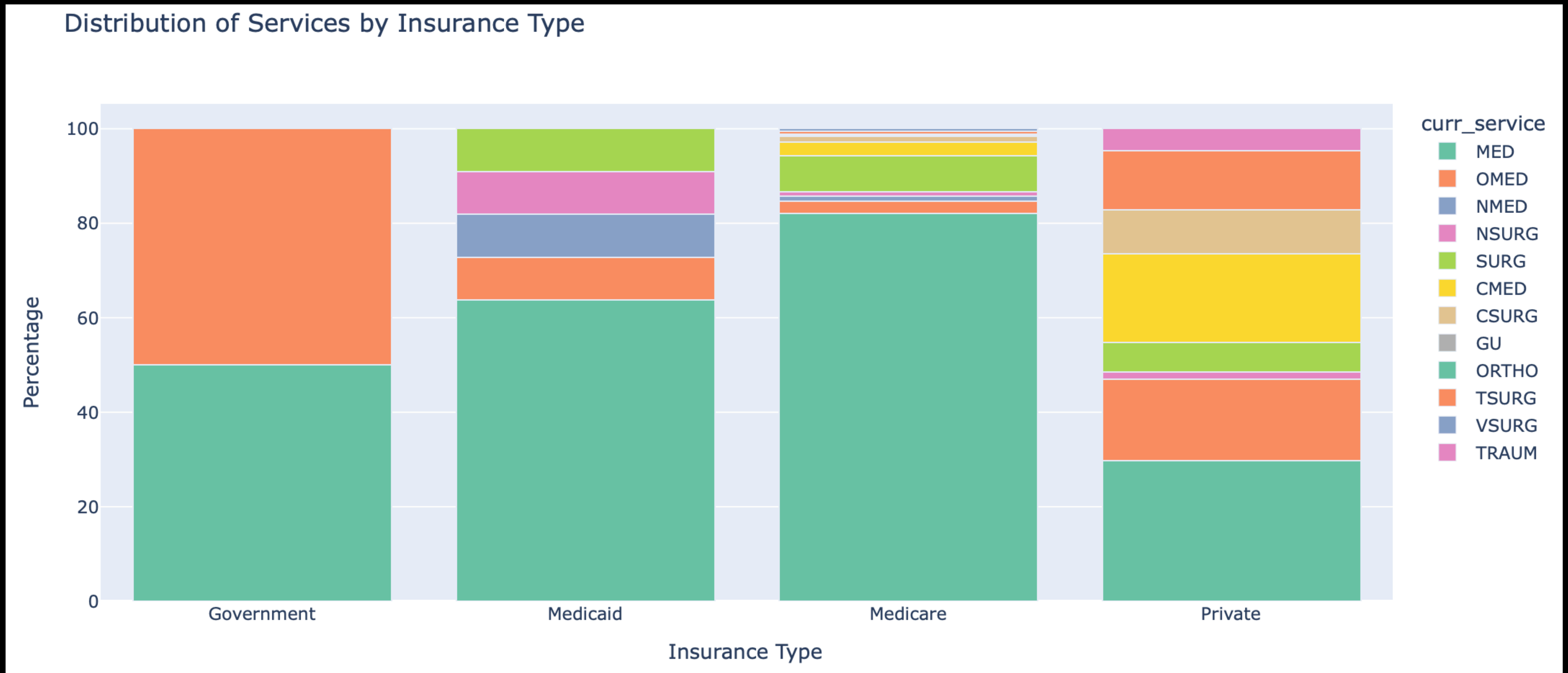
To create this visualization, I first merged the `perscriptions_df` and `p_a_df` dataframes on the 'subject_id' column, selecting the 'insurance' column from `p_a_df` for each subject using an inner join. I then grouped the merged data by 'insurance' and 'drug_name_generic' and calculated the prescription count for each drug using `groupby()` and `size()`. To focus on the top 10 drugs for each insurance type, I applied a `lambda` function with `nlargest(10, 'prescription_count')`. I then calculated the total prescriptions for each insurance type and computed the percentage of prescriptions for each drug by dividing the prescription count by the total prescriptions and multiplying by 100. For the visualization, I created a set of pie charts for each insurance type using `plt.subplot()`. Each chart shows the distribution of the top 10 most prescribed drugs, with percentages displayed on the slices and colors from the 'Paired' colormap. The pie charts are labeled with the drug names and organized in a grid. Finally, I applied `tight_layout()` to adjust the layout and displayed the plot with `plt.show()`.



5. Distribution of Services by Insurance Type

Detailed Instructions

To create this visualization, I first merged the \`services_df\` and \`p_a_df\` dataframes on the 'subject_id' column using an inner join, selecting the 'insurance' column from \`p_a_df\`. I then grouped the merged data by 'insurance' and 'curr_service' and counted the occurrences of each service using \`groupby()\` and \`size()\`. After that, I calculated the total number of services for each insurance type and computed the percentage of each service by dividing the service count by the total number of services and multiplying by 100. I exported the results to a CSV file for further analysis. For the plot, I used Plotly Express to create a stacked bar chart where each bar represents an insurance type, and the segments within each bar represent different services, colored by 'curr_service'. I customized the chart with a title, axis labels, and a color palette using the 'Set2' colormap. Finally, I updated the layout to ensure the bars are stacked, and displayed the plot with \`fig.show()\`.


Distribution of Services by Insurance Type
Percentage
Government
Medicaid
Medicare
Private
Insurance Type
curr_service
MED
OMED
NMED
NSURG
SURG
CMED
CSURG
GU
ORTHO
TSURG
VSURG
TRAUM