Visualisation Function Use Guideline and Example Analysis

Aaron

May 2024

1 Introduction

This document serves as a comprehensive guide for utilizing a suite of visualization functions specifically designed for analyzing the Health Insurance Marketplace's BenefitsCostSharing dataset. These functions are developed to enhance the interpretative power of cluster analysis. Each function is tailored to address different aspects of data visualization, ranging from plotting average values to exploring data distributions and categorical proportions across clusters. These facilitate a deeper understanding of complex datasets through graphical representations

The visualization toolkit provided herein is intended to support stakeholders, analysts, and researchers in making data-driven decisions and uncovering insightful patterns within the healthcare insurance sector. By implementing these functions, users can efficiently communicate findings, compare metrics across various dimensions, and thereby, contribute to the strategic planning and optimization of health insurance offerings.

In the following sections, we will explore five key visualization functions: `plot_average_by_cluster`, `plot_boxplot_cluster`, `plot_pie_chart`, `plot_proportion_by_cluster`, and `plot_heatmap_means`. Each section includes detailed usage instructions, scenarios for application, and example code snippets to demonstrate the practical implementation of these tools in a real-world context.

2 Function `plot_average_by_cluster`

2.1 Function Description:

The `plot_average_by_cluster` function is designed to create customized bar plots that display the average values of a specified numeric variable across different clusters in your dataset. This visualization is particularly useful for comparing the central tendencies of a variable across categorically segmented data.

2.2 Guidelines for Using the Function:

The full guideline of using `plot_average_by_cluster` function is structured as below:

Purpose:

- To visually compare the average values of a specific numeric variable across different clusters or groups.
- Useful in identifying patterns, outliers, or inconsistencies across categorized data.

Appropriate Data Types:

- **data**: A dataframe that contains at least one categorical column (for clustering) and multiple numeric columns (for calculating means).
- **cluster_col**: The column in your data frame that contains categorical data representing different groups or clusters.
- **numeric_col**: The numeric column for which you want to calculate and visualize the mean across each cluster. Ensure that this column contains numeric data such as integers or floating-point numbers.

Usage Scenarios:

- **Comparative Analysis**: When you need to compare metrics like sales, scores, rates, or any quantitative measure across different categorical groups defined by some clustering process or segmentation criterion.
- **Quality Control**: Checking consistency or performance metrics across different production batches or operational groups.
- **Market Analysis**: Understanding customer behavior across different market segments or demographic clusters.

Steps to Use the Function:

1. **Prepare Your Data**: Ensure your dataframe is set up with the correct types for the clustering column and the numeric column. Convert the cluster column to factor and other columns to numeric types if necessary.
2. **Call the Function**: Provide the dataframe, the name of the cluster column, and the numeric column as arguments. Optionally, customize the plot title, x-label, and y-label to fit your analysis context.
3. **Visualize and Interpret**: Use the output plot to discuss differences and trends among clusters. The colors in the bar chart provide a quick visual cue about the relative magnitude of the averages. Text labels on each bar offer precise values for more detailed insights.

Example Usage:

```
plot_average_by_cluster(data = recluster_profile, "cluster col = recluster",
                        "numeric col = coins_inn_tier1_num_mean",
                        plot_title = "Average Coinsurance In-Network Tier 1 by Cluster",
                        x_label = "Cluster", y_label = "Average Coinsurance")
```

2.3 Example Plot and Analysis:

The bar chart (Figure 1) produced by the `plot_average_by_cluster` function with the example codes shows the average coinsurance in-network tier 1 by cluster. This visualization allows us to quickly identify which clusters have higher or lower average coinsurance rates.

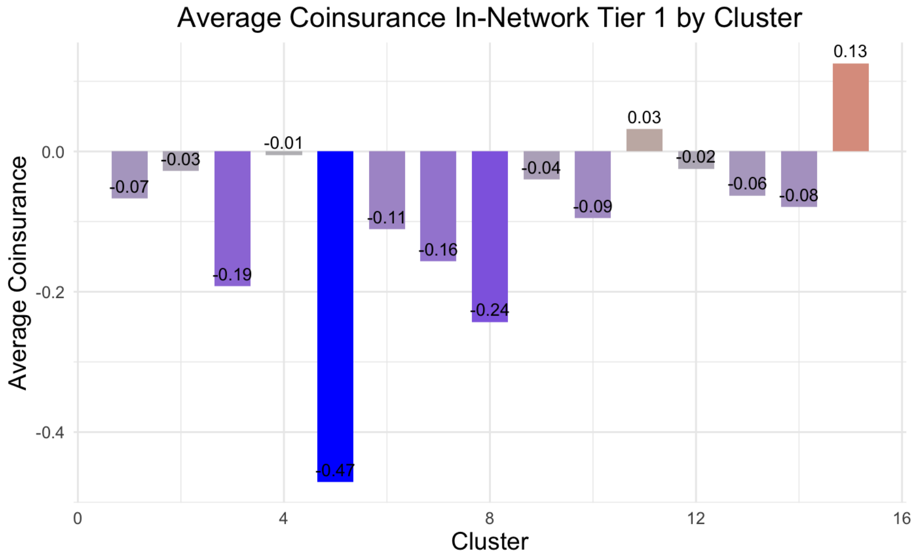

Figure 1: Average Coinsurance In-Network Tier 1 by Cluster. This bar chart visually represents the mean coinsurance rates across clusters.

From the chart, we observe significant variability in coinsurance rates across clusters. For instance, cluster 4 displays a notably lower average coinsurance (-0.47), suggesting that plans in this cluster generally offer more favorable cost-sharing terms for in-network tier 1 services compared to other clusters. In contrast, cluster 15 shows a higher average (0.13), which may indicate less favorable coinsurance terms.

Such visualizations are essential for stakeholders in the health insurance sector as they provide a clear and immediate understanding of how coinsurance responsibilities vary across different clusters. This may potentially affect consumer affordability and access.

3 Function `plot_boxplot_cluster`

3.1 Function Description:

The `plot_boxplot_cluster` function is designed to create customized boxplots for visualizing the distribution of a numeric variable across different clusters. The function leverages ggplot2 to generate boxplots, which are useful for identifying median, quartiles, and outliers within each group. It also uses RColorBrewer to handle color schemes efficiently, especially when dealing with multiple clusters.

3.2 Guidelines for Using the Function:

The full guideline of using `plot_boxplot_cluster` function is structured as below:

Purpose:

- To visualize the distribution of a numeric variable across clusters, helping identify trends, outliers, and the spread within each group.
- Useful for exploratory data analysis to quickly assess variability and central tendencies within segmented groups.

Appropriate Data Types:

- **data**: A dataframe that includes at least one categorical column for clustering and one numeric column for the distribution analysis.
- **cluster_col**: The categorical column in your dataframe that denotes clustering. Each unique value in this column represents a different cluster.
- **numeric_col**: The numeric column you wish to analyze. This should be a column containing quantitative data.

Usage Scenarios:

- **Understanding Benefit Distribution**: When you need to analyze how different types of benefits (e.g., emergency services, routine care) are distributed across different states or plan types.
- **Investigating Outliers**: To identify states or plans where the insurance costs or benefits are significantly different from others, which may warrant further investigation for risk assessment or policy adjustments.
- **Comparative Analysis**: Comparing health plan features across different segments to guide policy decisions or marketing strategies.

Steps to Use the Function:

1. **Prepare Your Data**: Ensure your dataframe contains the cluster and numeric columns needed. Convert the cluster column to factor and other columns to numeric types if necessary.

2. **Call the Function**: Provide the dataframe, the name of the cluster column, and the numeric column as arguments. Optionally, customize the plot title, x-label, and y-label to fit your analysis context.
3. **Visualize and Interpret**: Use the output plot to discuss distribution patterns like range, median, and presence of outliers across clusters. The use of colors helps distinguish between clusters.

Example Usage:

```
plot_boxplot_cluster(
  data = benefits_norm,
  cluster_col = "cluster",
  numeric_col = "coins_inn_tier1_num",
  plot_title = "Distribution of Coinsurance In-Network Tier 1 by Cluster",
  x_label = "Cluster",
  y_label = "Coinsurance In-Network Tier 1 (%)"
)
```

3.3 Example Plot and Analysis:

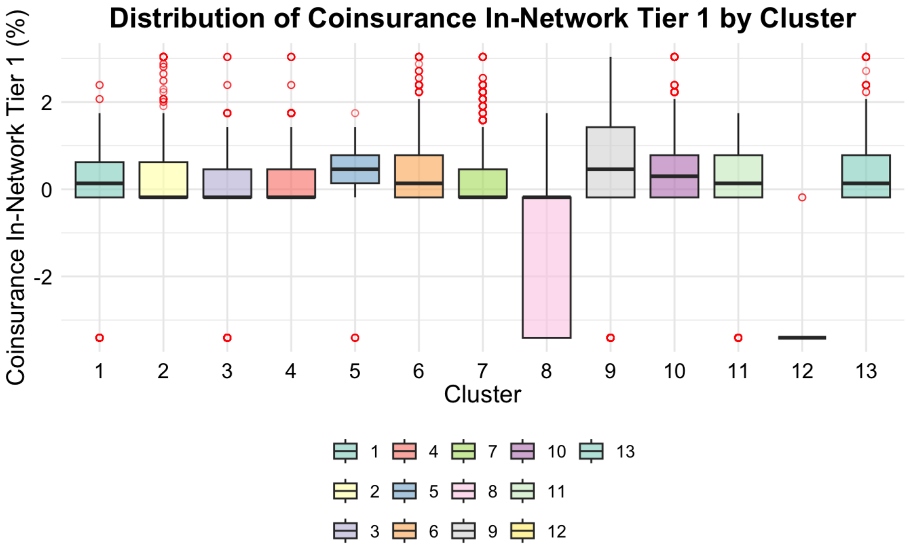

Figure 2: Distribution of Coinsurance In-Network Tier 1 by Cluster. Each boxplot shows the spread and central value of coinsurance rates. Outliers are marked in red, highlighting exceptions or anomalies in the data.

The boxplot produced by the `plot_boxplot_cluster` function (Figure 2) provides a detailed visualization of the distribution of coinsurance rates for in-network Tier 1 services across various clusters. This plot is instrumental in assessing the spread and central tendencies of coinsurance percentages. We can observe the following key findings from the box plot:

Key observations: - Outliers are present in several clusters, such as 1, 3, and 13, where individual plans may offer unusually high or low coinsurance rates compared to others within the same cluster. - Clusters such as 2 and 12 show relatively low variability and central medians close to zero, suggesting these clusters might consist of plans with moderate coinsurance demands. - The negative values in clusters like 6 and 11 could be indicative of data entry errors or specific incentives within those plans, requiring further investigation.

This type of analysis is critical for stakeholders looking to optimize insurance offerings or for consumers aiming to choose plans that align with their financial and health needs.

4 Function `plot_pie_chart`

4.1 Function Description:

The `plot_pie_chart` function creates a polar bar chart (essentially a pie chart) for visualizing the distribution of categorical variables across different clusters. This kind of visualization is particularly useful for assessing the proportions of categories within each cluster, making it easier to identify dominant or rare categories within each segment.

4.2 Guidelines for Using the Function:

The full guideline of using `plot_pie_chart` function is structured as below:

Purpose:

- To provide a visual representation of the proportion of different categories within each cluster, highlighting differences in category distribution across clusters.

Appropriate Data Types:

- **data**: A dataframe that must include at least one categorical column for the cluster and another categorical column for the category distribution.
- **cluster_col**: The column that denotes different clusters or groups. Each unique value in this column represents a different cluster.
- **category_col**: The categorical variable you want to analyze. This column contains the data that will be summarized and visualized in the pie chart.

Usage Scenarios:

- **Analyzing Cost Variability**: When interested in how certain costs or benefits, such as coinsurance or copayments, vary across different clusters like states or plan types.

- **Insurance Plan Analysis**: To visualize the distribution of various insurance plan types (like HMO, PPO) across different clusters such as geographic regions or consumer demographics.

Steps to Use the Function:

1. **Prepare Your Data**: Ensure your DataFrame contains the necessary cluster and category columns.
2. **Call the Function**: Provide the dataframe, cluster column, and category column as arguments. Optionally, customize the plot title and legend title to fit your analytical context.
3. **Visualize and Interpret**: Use the output plot to discuss the proportions of categories within each cluster. This can help in understanding how different categories are distributed across the clusters.

Example Usage:

```
plot_pie_chart(
  data = benefits_norm,
  cluster_col = "cluster",
  category_col = "is_subj_to_ded_tier1_num",
  plot_title = "Distribution of Deductible Subject by Cluster",
  legend_title = "Deductible Status"
)
```

4.3 Example Plot and Analysis:

As shown in the Figure 4, you could use the function `plot_pie_chart` to examine the distribution of deductible statuses across different clusters. This visual analysis can highlight differences in policy terms between clusters, such as the prevalence of deductible or no-deductible policies. We can draw the following observations and interpretations with the figure shown:

- **Clusters with predominant "Yes" status:** Clusters such as 9 and 11 predominantly show a higher proportion of plans with deductibles. This suggests that plans in these clusters are likely structured to include deductibles, possibly reflecting higher coverage options or plans designed for consumers expecting higher medical usage.
- **Clusters with significant "No" status:** Clusters 4 and 12 demonstrate a higher proportion of plans without deductibles. These clusters might be indicative of premium plans where direct out-of-pocket costs from deductibles are minimized, appealing to consumers seeking straightforward coverage.
- **Presence of "NA" status:** Clusters 1, 6, and 13 show a notable presence of "NA" status, indicating plans where the deductible component is

not applicable. This could reflect special types of insurance plans, such as catastrophic plans, where traditional deductibles are not a primary feature.

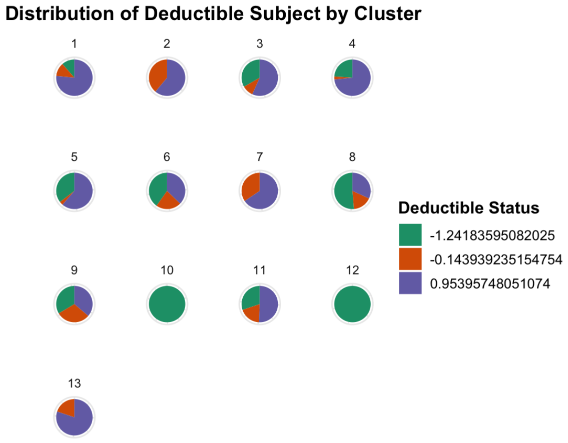

Figure 3: Distribution of Deductible Subject by Cluster using the `plot_pie_chart` function. Each pie chart shows the percentage of deductible statuses within each cluster, with "NA" assigned to the smallest numerical value, "No" to the next, and "Yes" to the largest.

5 Function `plot_proportion_by_cluster`

The `plot_proportion_by_cluster` function is designed to visualize the proportions of categories within each cluster, using a stacked bar plot where the height of each segment corresponds to the proportion of the category within the cluster. Please refer to guidance of `plot_pie_chart` function for function `plot_proportion_by_cluster`, as they are the same but with different presentation format.

Example Usage:

```
plot_pie_chart(
  data = benefits_norm,
  cluster_col = "cluster",
  category_col = "is_subj_to_ded_tier1_num",
  plot_title = "Distribution of Deductible Subject by Cluster",
  legend_title = "Deductible Status"
)
```

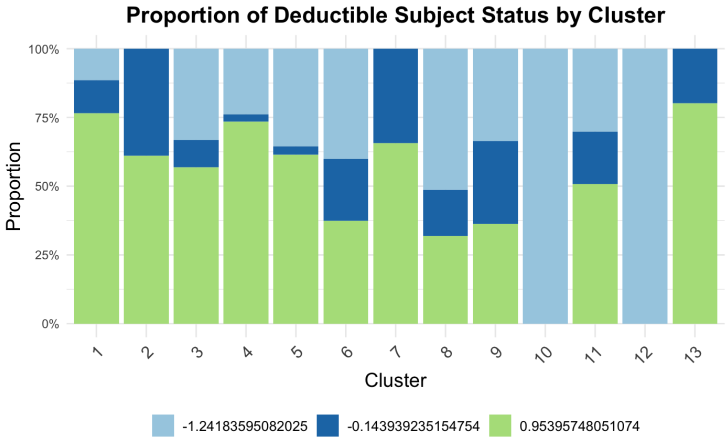

Figure 4: Distribution of Deductible Subject by Cluster using the `plot_proportion_by_cluster` function. Each pie chart shows the percentage of deductible statuses within each cluster, with "NA" assigned to the smallest numerical value, "No" to the next, and "Yes" to the largest.

6 Function `plot_heatmap_means`

6.1 Function Description:

The `plot_heatmap_means` function is specifically designed to create a heatmap visualization of the mean values for numeric variables across different clusters. This visualization is excellent for spotting significant average of an an interested variable among all clusters.

6.2 Guidelines for Using the Function:

The full guideline of using `plot_heatmap_means` function is structured as below:

Purpose:

- To visualize the average values of numeric variables across different clusters in a concise and visually appealing manner. Heatmaps allow quick identification of high and low values across multiple dimensions. This simplifies the comparisons across clusters.

Appropriate Data Types:

- **data**: A dataframe that contains at least one categorical column (for clustering) and multiple numeric columns (for calculating means).
- **cluster_col**: The column in your dataframe that denotes the cluster. Each unique value represents a different cluster.

Usage Scenarios:

- **Comparing Cost Structures**: Analyzing how average costs or premiums differ across states or plan categories.
- **Policy Impact Review**: Assessing the impact of policy changes by comparing pre and post-policy change averages across key metrics like beneficiary costs, copayments, or coinsurance rates.
- **Evaluating Service Utilization**: Understanding how different service utilization rates (like hospital stays or doctor visits) vary across different insurance plan clusters.

Steps to Use the Function:

1. **Prepare Your Data**: Your DataFrame should have clearly defined cluster columns and numeric columns that are clean and formatted correctly (no missing values or non-numeric types where not expected).
2. **Call the Function**: Provide the dataframe, and the name of the cluster column name as arguments. The function will then automatically calculate the means of all numeric columns, pivots the data, and then creates a heatmap.
3. **Visualize and Interpret**: The `plot_average_by_cluster` function generates a heatmap that visually represents the mean values of various variables across different clusters. The heatmap is structured with clusters labeled on the y-axis and metrics on the x-axis. Each cell in the heatmap represents the mean value of a metric for a specific cluster. This visualization is beneficial for quickly identifying patterns, or significant differences across clusters.

Example Usage:

```
plot_heatmap_means(data = benefits_norm, cluster_col = "cluster",
                   plot_title = "Heatmap of Mean Values by Cluster")
```

6.3 Example Plot and Analysis:

Understanding the Heatmap

- **Color Intensity:** The color of each cell reflects the magnitude of the mean value relative to others in the same metric, with a color gradient from light (low absolute values) to dark (high absolute values). This gradient helps in quickly spotting which clusters are above or below the average for specific metrics.
- **Reading the Heatmap:** To interpret the heatmap:
  - **Horizontally:** Reviewing across a row allows you to understand how a particular cluster compares across all metrics. A consistently dark or light row indicates overall higher or lower mean values for that cluster.

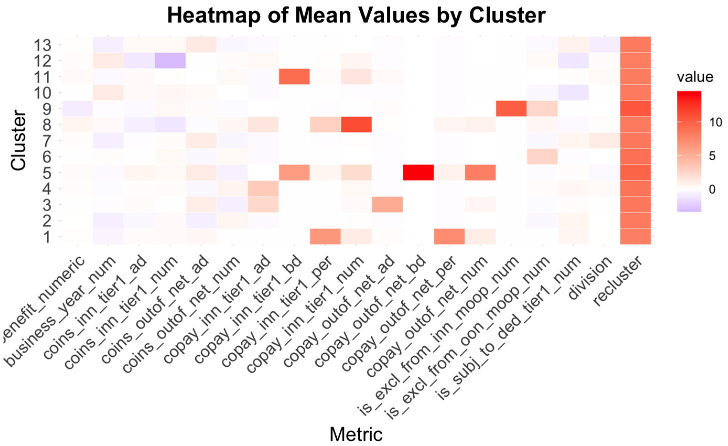

Figure 5: Heatmap of Mean Values by Cluster: Each cell represents the mean value of a metric for a cluster, with color intensity reflecting the magnitude. Darker shades indicate the means have a higher absolute value

  - **Vertically:** Analyzing down a column shows how all clusters perform on a specific metric. This is useful to identify if a metric generally scores high or low across clusters or if any cluster is an outlier.

- **Significance of Colors in Cells:** A block that is significantly darker or lighter than others in the same row or column highlights that the cluster is performing notably different from others in that metric. This can indicate a cluster with unique characteristics.

Example Interpretation

Consider a heatmap where the cluster ‘5‘ has darker cells in metrics related to ‘coins_inn_tier1_num‘ and ‘moop_inn_num‘. This suggests that cluster ‘5‘ might have benefits with notably higher coinsurance rates and maximum out-of-pocket costs for in-network services compared to other clusters. Such insights are critical for stakeholders aiming to adjust or market plans effectively within that cluster.