# Part I: Research Question

## A1: Proposal of Question

This analysis aims to answer the question: How do readmission rates, defined as the number of patients who were readmitted divided by the total number of discharges, differ between the naturally occurring clusters in the hospital system’s population? To answer the question, agglomerative hierarchical clustering will be used.

## A2: Defined Goal

The primary goal of this data analysis is to split the patient population into naturally occurring clusters. This will allow the research question regarding the differences in readmission rates between the groups to be answered.

# Part II: Technique Justification

## B1: Explanation of Clustering Technique

Agglomerative hierarchical clustering works by sequentially combining similar samples. Starting with each observation in its own group, this method combines the two nearest neighboring groups, then reassesses the new set of groups. This process is repeated until the desired number of groups, or clusters, has been reached (Patlolla, 2018). There are several methods by which the distance between groups can be calculated. This analysis uses Ward’s method, which reduces the sum of the square of distances, much like how regression analysis reduces the sum of the squared error of residuals. The expected outcome is that the desired number of cluster labels will be assigned to each of the observations in the sample.

## B2: Summary of Technique Assumption

While agglomerative hierarchical clustering analysis does not have any specific statistical assumptions (dearC, 2019), the implementation of the algorithm has a space complexity of O(n2) and a time complexity of O(n3) (Patlolla, 2018). This means that for every additional observation fed into the algorithm, the computational space is squared and the amount of time for the algorithm to run is cubed. Because of this, hierarchical clustering should only be performed on relatively small sets of data.

## B3: Packages or Libraries List

To complete this analysis, the following Python 3.9 packages and libraries were used:

* **pandas**: The pandas library provides an interface for working with tabular data. It allows data manipulation that is similar to SQL. For this analysis, it is used to import and prepare the data for analysis.
* **Matplotlib**: The Matplotlib library provides an interface for creating visualizations based on data. In this analysis, it is used to create and annotate visualizations.
* **Seaborn**: The Seaborn library provides a simplified interface for creating visualizations based on data. It abstracts away much of the complexity of Matplotlib but does not offer the same flexibility. In this analysis, it is used to produce scatter plots.
* **SciPy**: The SciPy library provides an interface for various types of computing including scientific and statistical. For this analysis, it is used to assess the data for outliers as well as perform the hierarchical linkage and produce the dendrogram.
* **Scikit-learn**: The Scikit-learn library, also known as sklearn, provides an interface for the end-to-end application of machine learning algorithms. For this analysis it is used to both scale the numeric data prior to clustering as well as to visualize the t-distributed stochastic neighbor embedding or t-SNE.

# Part III: Data Preparation

## C1: Data Preprocessing

One goal of the data preprocessing step in this analysis is to scale the continuous variables. Because hierarchical clustering uses a distance metric to assess the nearness of clusters, it is important to ensure that the variables have all been brought down to a similar scale. Doing this prevents distortion when the algorithm assesses the proximity of groups.

## C2: Dataset Variables

The following variables will be used to perform the analysis and answer the research question:

|  |  |
| --- | --- |
| **Variable** | **Class** |
| ReAdmis | Categorical |
| Initial\_admin | Categorical |
| HighBlood | Categorical |
| Stroke | Categorical |
| Complication\_risk | Categorical |
| Overweight | Categorical |
| Arthritis | Categorical |
| Diabetes | Categorical |
| Hyperlipidemia | Categorical |
| BackPain | Categorical |
| Anxiety | Categorical |
| Allergic\_rhinitis | Categorical |
| Reflux\_esophagitis | Categorical |
| Asthma | Categorical |
| Services | Categorical |
| Initial\_days | Continuous |
| TotalCharge | Continuous |

## C3: Steps for Analysis

The following code segments were extracted from the file “task1.py”, which has been included with this submission.

First the required libraries were imported, and the output directory was specified.

**import** pandas **as** pd  
**import** matplotlib.pyplot **as** plt  
**import** seaborn **as** sns  
**from** scipy **import** stats  
**from** scipy.cluster.hierarchy **import** linkage, dendrogram, fcluster  
**from** sklearn.preprocessing **import** StandardScaler  
**from** sklearn.manifold **import** TSNE

**from** sklearn.metrics **import** silhouette\_score

OUT = **'./output1'**

Next, the data was loaded and previewed.

medical\_clean = pd.read\_csv(**'./medical\_clean.csv'**)print(**f'\n{**medical\_clean.head().to\_string()**}\n'**)

Then the data structures and types were evaluated, and the data was checked for missing values.

print(**f'\ncontinuous data:\n{**medical\_clean.describe().T.to\_string()**}'**)  
cat\_data = medical\_clean[[col **for** col **in** medical\_clean.columns  
 **if** medical\_clean[col].dtype **not in** [**'float64'**,   
 **'int64'**]]]  
print(**f'\ncategorical data:\n{**cat\_data.describe().T.to\_string()**}\n'**)medical\_clean.info()

Next, the data was checked for duplicates and outliers.

print(**f'\nduplicates:\n{**medical\_clean.duplicated().sum()**}'**)  
num\_cols = [col **for** col **in** medical\_clean.columns  
 **if** medical\_clean[col].dtype **in** [**'float64'**, **'int64'**]]  
df\_num = medical\_clean[num\_cols]  
df\_zscores = df\_num.apply(stats.zscore)  
df\_outliers = df\_zscores.apply(**lambda** x: (x > 3) | (x < -3))  
print(**f'\noutliers:\n{**df\_outliers.sum().to\_string()**}'**)

Once evaluation of the dataset was complete, the features of interest were selected.

features = [**'Initial\_admin'**, **'HighBlood'**, **'Stroke'**, **'Complication\_risk'**,  
 **'Overweight'**, **'Arthritis'**, **'Diabetes'**, **'Hyperlipidemia'**,  
 **'BackPain'**, **'Anxiety'**, **'Allergic\_rhinitis'**,   
 **'Reflux\_esophagitis'**, **'Asthma'**, **'Services'**, **'Initial\_days'**,   
 **'TotalCharge'**]  
df = medical\_clean[features]

Then the numeric features were normalized using the Scikit-learn class, StandardScaler.

num\_cols = [col **for** col **in** features  
 **if** df[col].dtype **in** [**'float64'**, **'int64'**]]  
scaler = StandardScaler()  
df.loc[:, num\_cols] = scaler.fit\_transform(df[num\_cols])  
print(df[num\_cols].head().to\_string())

Next, the categorical features were encoded. A few of the dummy features were dropped because they did not have a high value in prior analyses. The column names produced by the get\_dummies function were also updated to replace spaces with underscores.

df.to\_csv(os.path.join(OUT, **'cleaned\_dataset.csv'**))

df.replace({**'Yes'**: 1, **'No'**: 0}, inplace=**True**)  
df = pd.get\_dummies(df)  
df.drop([**'Initial\_admin\_Elective\_Admission'**,  
 **'Initial\_admin\_Observation\_Admission'**,  
 **'Complication\_risk\_High'**, **'Services\_Blood\_Work'**,  
 **'Services\_Intravenous'**],  
 axis=1, inplace=**True**)  
df.columns = [col.replace(**' '**, **'\_'**) **for** col **in** df.columns]  
print(df.head().to\_string())

Finally, the cleaned dataset was saved.

## C4: Cleaned Dataset

The cleaned dataset has been included with this submission as “cleaned\_dataset.csv”.

# Part IV: Analysis

## D1: Output and Intermediate Calculations

First, to decide how many clusters should be identified by the analysis, a t-distributed stochastic neighbor embedding (t-SNE) plot was produced. t-SNE plots flatten multidimensional observations so that the relative distances between observations can be viewed on a two-dimensional plane (Wilson, nd).

Chart, scatter chart

Description automatically generated

When viewing the t-SNE plot, two clusters are identifiable with some noise between them.

Next, the agglomerative hierarchical linkage was applied and a dendrogram was produced to review the applied linkage and validate the clusters seen in the t-SNE plot.

A picture containing diagram

Description automatically generated

Because the dataset contains 10,000 observations, the bottom of the dendrogram is not meaningful, nor are the labels on the x-axis. However, as was seen in the t-SNE plot, there are two clear clusters in the data. The case could be made for four clusters, but the analysis proceeded with two because of the agreement between both visuals.

Next, the clusters were labeled and assigned different colors on the t-SNE plot to visually assess the accuracy of the assignments.

Chart, scatter chart

Description automatically generated

The Silhouette Score was calculated next.



Then, the distribution of each feature between the two clusters was reviewed.

Text

Description automatically generated with medium confidenceText

Description automatically generatedText

Description automatically generated with low confidenceText, table

Description automatically generated Text

Description automatically generated with medium confidence Text

Description automatically generated with medium confidence Text

Description automatically generated Graphical user interface, text

Description automatically generated Text

Description automatically generated with medium confidence

Text

Description automatically generated with medium confidence Text

Description automatically generated with medium confidence Text

Description automatically generated Graphical user interface, text, application

Description automatically generated with medium confidence Table

Description automatically generated with medium confidence

For the continuous features, the unscaled means were compared between the two clusters.

Text

Description automatically generated

Finally, the readmission rates were calculated for each cluster to allow the research question to be answered.

Graphical user interface

Description automatically generated with medium confidence

## D2: Code Execution

The following code segments were extracted from the file “task1.py”, which has been included with this submission.

*# View flattened clusters*model = TSNE(learning\_rate=100)  
transformed = model.fit\_transform(df)  
xs = transformed[:, 0]  
ys = transformed[:, 1]  
sns.scatterplot(x=xs, y=ys)  
plt.title(**'t-SNE plot'**)   
plt.show()  
  
*# Perform hierarchical linkage and view dendrogram*mergings = linkage(df.values, method=**'ward'**)  
plt.subplots(figsize=(10, 6))  
dendrogram(mergings)  
plt.title(**'Ward Linkage'**)   
plt.show()  
  
*# Label the clusters and view them*df[**'cluster\_labels'**] = fcluster(mergings, 2, criterion=**'maxclust'**)  
sns.scatterplot(x=xs, y=ys, hue=df.cluster\_labels)  
plt.title(**'Labeled t-SNE plot'**)   
plt.show()  
  
*# Assess accuracy*  
score = silhouette\_score(df.loc[:, df.columns.\_\_ne\_\_(**'cluster\_labels'**)],  
 df.cluster\_labels)  
print(**'Silhouette Score:'**, score, **'\n'**)  
  
*# Summarize each of the clusters***for** f **in** features:  
 **if** f **not in** [**'TotalCharge'**, **'Initial\_days'**]:  
 ct = pd.crosstab(df.cluster\_labels, medical\_clean[f])  
 print(ct.to\_string(), **'\n'**)  
summary = pd.concat([df.cluster\_labels,  
 medical\_clean[[**'TotalCharge'**, **'Initial\_days'**]]],  
 axis=1).groupby(**'cluster\_labels'**).mean()  
print(summary.to\_string(), **'\n'**)  
  
*# Calculate the readmission rates*rr\_df = pd.concat([df.cluster\_labels, medical\_clean.ReAdmis], axis=1)  
rr\_df.replace({**'Yes'**: 1, **'No'**: 0}, inplace=**True**)  
readmission\_rate = rr\_df.groupby(**'cluster\_labels'**).mean()  
print(readmission\_rate.to\_string(), **'\n'**)

# Part V: Data Summary and Implications

## E1: Accuracy of Clustering Technique

Because there were no pre-defined labels associated with the clusters, they could not be assessed for accuracy using the traditional accuracy score metric. However, they could be anecdotally assessed using a t-SNE plot. The cluster labels were encoded as the color of each point to determine how well the visual clusters were captured. With the exception of about five to ten observations from cluster 1, all points appear to have been correctly labeled. Attempting to quantify this, 10,000 total observations minus 10 incorrectly labeled observations, then divided by 10,000 total observations produces a pseudo accuracy score of 0.999 or 99.9%.

The Silhouette Score, a metric used to measure how alike clustered observations are when compared to observations in other clusters (Ballard, 2020), was used to assess accuracy in the more formal sense. On a scale ranging from -1, meaning the clusters are incorrectly labeled, to 1, meaning the clusters are correctly labeled, the clusters scored 0.30. Given that the clusters are somewhat close to each other and that they are not neatly shaped or well divided, this score is decent. The Calinski Harabaz Index could have also been used, but the resulting score is not bounded making it difficult to interpret in isolation.

## E2: Results and Implications

When reviewing both the t-SNE plot and the hierarchical agglomerative linkage dendrogram, there were two clearly observable clusters in the data. Upon labeling the clusters, the labels applied to the linkage nearly perfectly aligned with the visible clusters in the t-SNE plot. On average cluster 1 had total charges that were $4,122.57 higher than cluster 2 and initial length of stay that was 50.37 days longer than cluster 2. When readmission rates were compared between the two clusters, cluster 2 had 0 readmissions while cluster 1 had a readmission rate of 73.24 readmissions per 100 admissions.

These results imply that there are two clearly separable clusters within the hospital chain’s patient population. The two clusters have very different total charges, lengths of stay, and readmission rates. Cluster 2 admissions tend to be shorter, cheaper, and do not result in readmission while cluster 1 admissions tend to be longer, more expensive, and more often than not, result in readmission within 30 days of discharge. Knowing these two groups exist and what their characteristics are is important information for the hospital chain’s administrators to be aware of as they work to reduce readmission rates across the system.

## E3: Limitation

The biggest limitation to this analysis was that there were no expected labels defined prior to analysis. It was purely exploratory in nature. The goal of the analysis was to discover the labels inherent in the data, which means there was no pre-defined way to measure accuracy. While there several ways to measure inter-cluster and intra-cluster variance exist, no method fits every scenario and most of them tend to bias towards equal variance within and/or between clusters. These tendencies often bias towards the expectation of well separated and evenly shaped clusters of similar size. Unfortunately, medical data rarely conforms to this norm, especially when investigating unbalanced scenarios like whether a patient will be readmitted to the hospital. This can be seen in the t-SNE plots which contain one fairly rounded cluster, and an oblong cluster which does not appear to take up as much space.

## E4: Course of Action

Based on the results of this analysis, two courses of action should be taken. First, the hospital chain should work to identify the predictors which place a patient in cluster 1 vs cluster 2. Knowing which combinations of predictors place a patient in either cluster will help rule out or predict the increased potential for readmission to the hospital. Second, cluster 1 should be analyzed further to see what sub-clusters exist. Because there are several sets of predictors which can likely label a patient as part of cluster 1, it would be beneficial to better understand what the different combinations look like and how they can be identified.

# Part VI: Demonstration

## F: Panopto Recording

The Panopto recording has been included with this submission as “task1\_panopto\_recording”.

## G: Sources for Third-Party Code

No sources for third-party code were used for this analysis.

## H: Sources

Ballard, C. (2020, May 15). *How to Evaluate Unsupervised Learning Models.* Retrieved from Towards Data Science: https://towardsdatascience.com/how-to-evaluate-unsupervised-learning-models-3aa85bd98aa2

dearC. (2019, August 26). *All the Annoying Assumptions.* Retrieved from Towards Data Science: https://towardsdatascience.com/all-the-annoying-assumptions-31b55df246c3

Patlolla, C. R. (2018, December 10). *Understanding the concept of Hierarchical clustering Technique.* Retrieved from Towards Data Science: https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec

Wilson, B. (nd). *Visualizing hierarchies | Python.* Retrieved from DataCamp: https://campus.datacamp.com/courses/unsupervised-learning-in-python/visualization-with-hierarchical-clustering-and-t-sne?ex=1