FINAL PROJECT REPORT.

Analyzing data from 2013 National Hospital Ambulatory Medical Care Survey (NHAMCS) data

Submitted by:

Shivam Ranjan [20017743]

Jimit Mehta

Shivani Patel

Aman Kumar

BIA 652

The 2013 National Hospital Ambulatory Medical Care Survey (NHAMCS) is a significant dataset for investigating the factors that influence patient wait times in emergency rooms. This study details an in-depth examination of the NHAMCS dataset, with an emphasis on the amount of time patients must wait to see healthcare providers in emergency departments, referred to as "WAITTIME." The study, conducted by the Centers for Disease Control and Prevention's National Centre for Health

Statistics, is a national probability sample survey that includes visits to hospital outpatient and emergency departments, as well as hospital-based ambulatory surgery center.

This study intends to find crucial insights into the factors that influence patient wait times, with a particular emphasis on three core research questions:

The Impact of Racial Minority Groups on Waiting Times: Determine whether patients from racial minority groups have greater wait times in emergency departments.

The Influence of Holiday seasons on Waiting Times: Determine whether waiting times vary during holiday seasons, offering insight into potential seasonal factors.

Predictive Variable Identification: Identify the five most relevant numeric and categorical variables that contribute significantly to predicting patient wait times.

Dataset:

The dataset, which is part of the National Health Care Surveys, provides a comprehensive assessment of healthcare utilization across multiple providers.

- The dataset's number of observations represents individual visits to emergency departments over the survey period. The dataset contains characteristics that span a wide range of aspects associated with each emergency room visit. "WAITTIME," which signifies the waiting time in minutes to see a healthcare professional, is a relevant variable along with others such as "ARRTIME", "LOV", etc. for this analysis.

- The dataset is constructed to include solely information from emergency department records, which aligns with the analysis's focus on patient wait times. Other components of the dataset description cover the survey scope, sample, field activities, data collecting and coding techniques, population estimates, codebook, and marginal data for chosen categories.

- NHAMCS is based on a nationally representative probability sample, ensuring a comprehensive picture of healthcare utilization in the United States. The dataset includes data from 2013, providing a snapshot of emergency department visits throughout that year.

After starting the Jupyter Notebook and creating a new file, I started with importing the necessary libraries as shown in the image below:

import pandas as pd: Pandas is a robust data manipulation and analysis package. We can utilise the pd prefix for Pandas functions by importing it as pd, making the code more compact.

import seaborn as sns: Seaborn is a Matplotlib-based data visualization library. It offers a high-level interface for creating visually appealing and informative statistical visuals. It is used to generate boxplots in this case.

import matplotlib.pyplot as plt: Matplotlib is a 2D plotting package, therefore import matplotlib.pyplot as plt. Seaborn is built on Matplotlib, and we leverage Matplotlib for extra plot customisation.

from sklearn.model_selection import train_test_split: Scikit-learn is a library for machine learning. To evaluate the model, we import train_test_split to split the dataset into training and testing sets.

from sklearn.linear_model import LinearRegression: To build a regression model, we import the Linear Regression model from scikit-learn.

from sklearn.metrics import mean_squared_error: This import is used to evaluate the model's performance using mean squared error.

In the next step, I used the code snippet shown below to read the dataset

pd.read_stata(): This Pandas function is used to read Stata datasets. 'ED2013-stata.dta' is the exact path to your dataset file in Jupyter Notebook.

convert_categoricals=False: This parameter is set to False to prevent categorical variables from being converted automatically. For better control, we might want to handle categorical variables ourselves.

convert_missing=True: When this parameter is set to True, Stata missing values are converted to NaN.

After this, I conducted **Exploratory Data Analysis**. EDA provided a full overview of the dataset, assisting analysts in comprehending the nature of the data with which they are working. EDA visualisation tools aided in identifying patterns, trends, and potential correlations within data, which

can then be used to direct further analysis. EDA also aided in the detection of outliers or anomalies in data that may influence the output of statistical models.

And it gave the following output –

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 24777 entries, 0 to 24776
Columns: 591 entries, VMONTH to EDWT
dtypes: int16(179), int32(37), int8(115), object(260)
memory usage: 64.0+ MB
None
              VMONTH         VDAYR      WAITTIME           LOV           AGE
count  24777.000000  24777.000000  24777.000000  24777.000000  24777.000000
mean       6.334383      3.937805     36.382694    205.344069     37.807765
std        3.453591      1.981420     70.247818    285.331943     23.962282
min        1.000000      1.000000     -9.000000     -9.000000      0.000000
25%        4.000000      2.000000      4.000000     75.000000     20.000000
50%        6.000000      4.000000     17.000000    140.000000     35.000000
75%        9.000000      6.000000     43.000000    240.000000     55.000000
max       12.000000      7.000000   1227.000000   5760.000000     93.000000

               AGER        AGEDAYS      RESIDNCE           SEX          ETHUN
count  24777.000000  24777.000000  24777.000000  24777.000000  24777.000000
mean       3.111676     -2.000000      0.818905      1.446987     -1.450377
std        1.470817     35.124036      1.620776      0.497192      4.983413
min        1.000000     -7.000000     -9.000000      1.000000     -9.000000
25%        2.000000     -7.000000      1.000000      1.000000     -9.000000
50%        3.000000     -7.000000      1.000000      1.000000      2.000000
75%        4.000000     -7.000000      1.000000      2.000000      2.000000
max        6.000000    364.000000      4.000000      2.000000      2.000000

       ...        BEDDATA          HLIST       HLISTED         REGION  \
count  ...  24777.000000  24777.000000  24777.000000  24777.000000
mean   ...      0.868830      0.971950     -0.303871      2.582677
std    ...      2.483475      1.140471      3.372949      1.025681
min    ...     -8.000000     -8.000000     -9.000000      1.000000
25%    ...      1.000000      1.000000      1.000000      2.000000
50%    ...      1.000000      1.000000      1.000000      3.000000
75%    ...      1.000000      1.000000      1.000000      3.000000
max    ...      6.000000      2.000000      2.000000      4.000000

              MSA  SETTYPE      YEAR       CSTRATM          CPSUM
count  24777.000000  24777.0  24777.0  2.477700e+04   24777.000000
mean       1.155669      3.0   2013.0  3.168823e+07   42947.443395
std        0.362548      0.0      0.0  9.899301e+06   49546.347418
min        1.000000      3.0   2013.0  2.011320e+07       1.000000
25%        1.000000      3.0   2013.0  2.031320e+07      21.000000
50%        1.000000      3.0   2013.0  4.010000e+07      36.000000
75%        1.000000      3.0   2013.0  4.030000e+07  100090.000000
max        2.000000      3.0   2013.0  4.040000e+07  100271.000000

             PATWT
count  24777.000000
mean    5261.051822
std     5053.340016
min       28.000000
25%     2132.000000
50%     3784.000000
75%     6473.000000
max    43788.000000

[8 rows x 331 columns]
VMONTH      0
VDAYR       0
ARRTIME     0
WAITTIME    0
LOV         0
           ..
YEAR        0
CSTRATM     0
CPSUM       0
PATWT       0
EDWT        0
Length: 591, dtype: int64
```

From the output obtained above, It can be interpreted that the dataset consists of 24,777 rows and 591 columns in a Pandas DataFrame. It has a combination of integer and object types (int16, int32, int8). The DataFrame's memory utilization is reported to be roughly 64.0 MB. The dataset includes patient demographics (e.g., age, gender, ethnicity), timing information (e.g., arrival time, wait time), and other categorical characteristics (e.g., Residence, Ethnicity). The descriptive statistics reveal the central tendency and variability of numerical variables, while the minimum and maximum values indicate the data range. There are no missing values reported, making future analysis easier. However, it is always necessary to validate data quality because missing values may be encoded differently (for example, as -9).

Now, we'll work on the given questions.

Question 1: Do patients from racial minority groups wait longer?

Method:
In the code below – **g = sns.FacetGrid(ed2013, col='RACER, col_wrap=4, height=4):**

sns.FacetGrid: In Seaborn, we can use this function to generate a grid of subplots based on the values of a categorical variable (in this case, 'RACER'). **ed2013:** The dataset is contained in the DataFrame.
col='RACER': Tells the grid to create a column for each unique value in the 'RACER' column. **col_wrap=4:** Reduces the number of columns in the grid to four, making it easier to visualize if there are a lot of different racial groups.
height=4: The height of each subplot is set to 4 inches.

g.map(plt.hist,'WAITTIME', bins=30, color='skyblue'):

g.map: Applies the provided graphing function (plt.hist in this case) to each FacetGrid-defined subset of data. **plt.hist:** Generates a histogram.
'WAITTIME': The variable plotted in the histogram. **bins=30:** The number of bins in the histogram is specified. **color='skyblue':** Changes the colour of the histogram bars to sky blue.

set_axis_labels('Wait Time (minutes),' 'Frequency'):

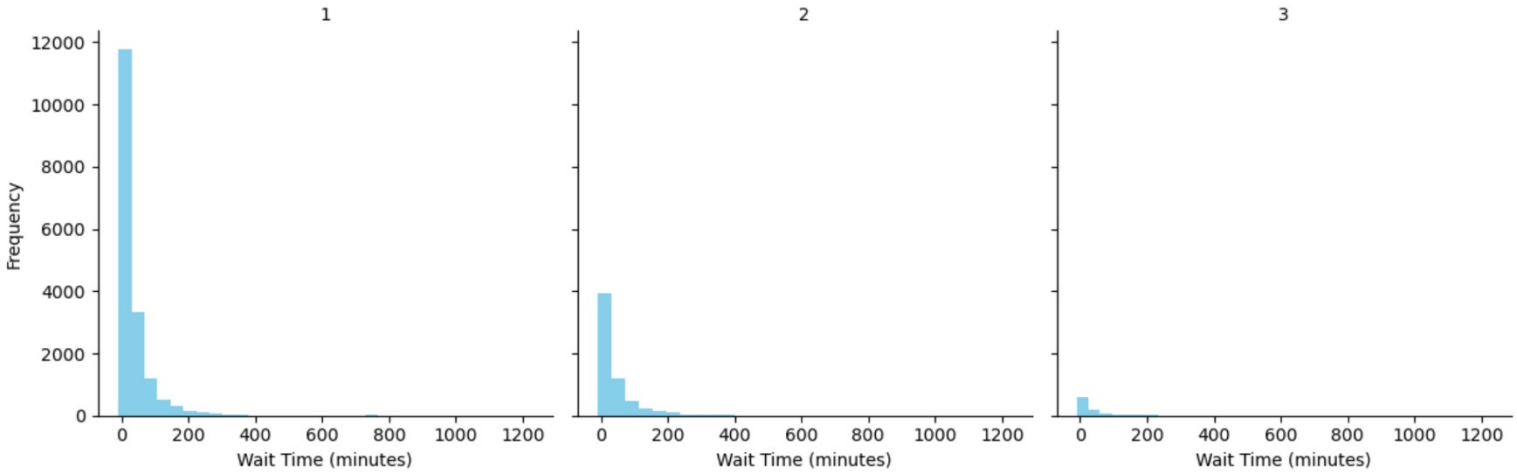
g.set_axis_labels: Sets the labels for each subplot's x and y axes.
'Wait Time (Minutes)': x-axis label.
'Frequency': the y-axis label.

g.set_titles(col_template='{col_name}'):

g.set_titles: Sets the titles for each subplot.
col_template='col_name': Uses the column name ('RACER') as the subplot title.

This code generates a grid of histograms, with each subplot representing the wait time distribution for a distinct racial group. The histograms show whether there are disparities in wait times between racial groupings. FacetGrid makes it simple to compare these distributions side by side. The labels and names have been added to help with the interpretation of the visualizations. The offered code's objective is to visually investigate and compare the distribution of emergency room wait times for different racial groups.

Result:



The histogram analysis indicates that wait times differ between racial groups, highlighting potential inequities in the healthcare system. A significant number of people in Group 1, who are mostly white, have very short or no wait periods, with a noticeable decrease as wait times grow. Group 2, which represents black people, exhibits a similar pattern, with greater frequencies for shorter wait times and a decline as wait times lengthen. Group 3, which includes various ethnicities, has a moderate proportion of people with short wait times, but the frequency decreases with increasing wait periods. While these data suggest potential discrepancies, more statistical research, including hypothesis testing, is required to determine the relevance of the observed differences.

Furthermore, various confounding factors and biases that may influence the results must be considered. The histogram data suggest that patients from racial minority groups, notably black people, may have longer wait times, but further statistical analysis is required for a more firm conclusion.

Question 2: If wait times increase during holiday periods?

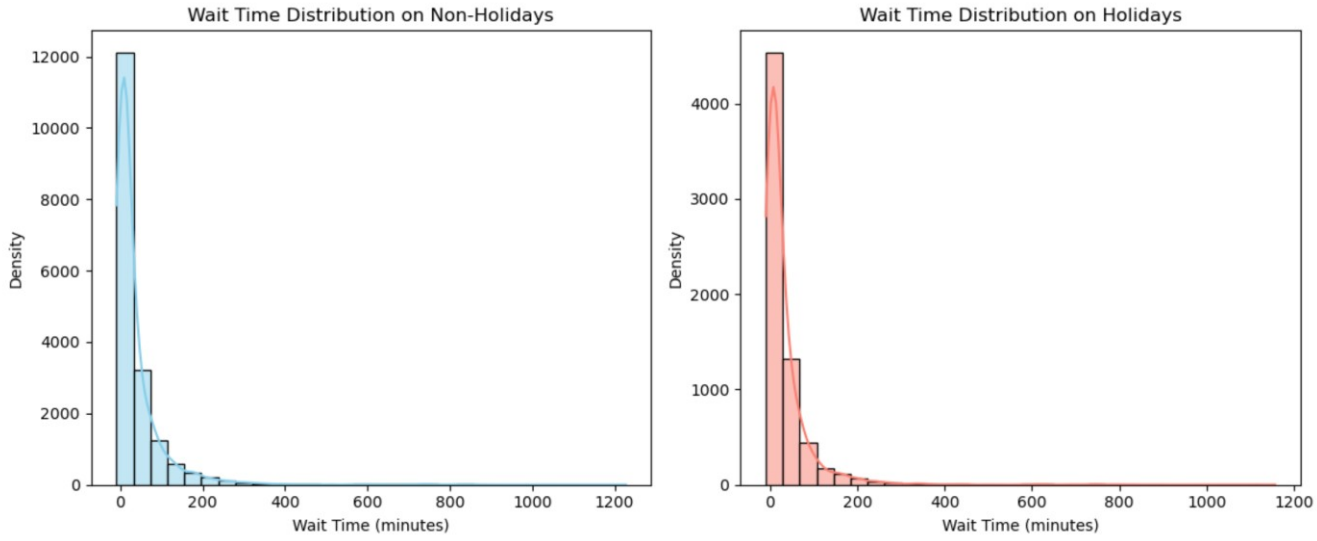
Method:

I wrote the code shown below where, holiday_days = [1, 7] generates a list to define the days that are designated holidays, with the integer 1 representing Sunday and the number 7 representing Saturday. This list will be used to classify weekend days as holidays in the ensuing study. The line ed2013['IS_HOLIDAY'] = ed2013['VDAYR'].isin(holiday_days) adds a new column to the dataset 'ed2013' called 'IS_HOLIDAY'. This column indicates whether or not a healthcare visit falls on a holiday. The method.isin(holiday_days) is used to determine whether the day represented by the dataset's 'VDAYR' column is included in the list of holiday days. As a result, the 'IS_HOLIDAY' column will contain Boolean values, with 'True' indicating that the day is a holiday and 'False' indicating that it is not.

Plt.subplots(nrows=1, ncols=2, figsize=(12, 5)) generates a figure with two subplots placed in a single row. The figsize param is used to specify the overall size of the figure, guaranteeing ideal dimensions for visual representation. Following that, the code plots histograms for both nonholidays and holidays. The non-holidays subplot is created using sns.histplot(...), which filters the data for entries corresponding to nonholidays, sets the variable 'WAITTIME' as the x-axis variable, and configures parameters such as the number of bins, kernel density estimate, and bar color. The plot is then linked to the first subplot with ax=axes[0], and titles, x-axis labels, and y-axis labels are specified to improve interpretability. Similarly, the code constructs a histogram for holidays, filters the data, and associates the plot with the second subplot. Titles and labels are also changed for the second subplot. Finally, the layout is changed with plt.tight_layout() to avoid overlap between subplots, and the final plot is displayed with plt.show(). The overall goal of this code is to visually compare the distribution of wait times during non-holidays versus holidays, with various subplots and colors used to improve clarity. The addition of kernel density estimates results in a smoother depiction of the data distribution, and carefully placed labels and titles offer important context, making the plot more useful.

In [13]:
```
#Question 2: If wait times increase during holiday periods?

#Creating a list of holidays that include Sunday and Saturday
holiday_days = [1, 7]  # Sunday and Saturday are considered holidays

# Creating a new column 'IS_HOLIDAY' indicating whether the visit day is a holiday
ed2013['IS_HOLIDAY'] = ed2013['VDAYR'].isin(holiday_days)

# Setting up the figure and axes
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 5))

# Plotting the histogram for wait times on non-holidays
sns.histplot(data=ed2013[ed2013['IS_HOLIDAY'] == False], x='WAITTIME', bins=30, kde=True, color='skyblue', ax=axes[0])
axes[0].set_title('Wait Time Distribution on Non-Holidays')
axes[0].set_xlabel('Wait Time (minutes)')
axes[0].set_ylabel('Density')

# Plotting the histogram for wait times on holidays
sns.histplot(data=ed2013[ed2013['IS_HOLIDAY'] == True], x='WAITTIME', bins=30, kde=True, color='salmon', ax=axes[1])
axes[1].set_title('Wait Time Distribution on Holidays')
axes[1].set_xlabel('Wait Time (minutes)')
axes[1].set_ylabel('Density')

# Adjusting layout to prevent overlap
plt.tight_layout()

plt.show()
```

Result



The histogram analysis of non-holiday and holiday emergency room wait times reveals distinct patterns in patient experiences. On non-holidays, the majority of patients have relatively low wait times, with a peak density of over 12,000 between 0 to 40 minutes. As wait time get longer, the density decreases, indicating a decrease in the number of patients who must endure longer delays. Notably, after 200 minutes, the density sharply declines to zero, showing uncommon cases of exceptionally long wait times during non-holidays. Holidays, on the other hand, show a similar pattern, with a peak density of roughly 5,000 for wait time ranging from 0 to 40 minutes. During the holidays, however, the density declines to zero if wait durations exceed 200 minutes, indicating a notable absence of instances with unusually lengthy wait times.

Q3 The five most important numeric variables and the five most important categorical variables that can predict how long a patient will wait.

Method:

I wrote the code that performs a feature importance analysis on the emergency room dataset using a Random Forest regression model to discover key factors influencing wait time. Initially, relevant features from the dataset are chosen, which include both numeric and categorical variables. To facilitate model training, the selected features are aggregated into a subset of the data, and categorical variables are turned into dummy variables. To assess the model's performance, the dataset is divided into training and testing sets. After that, a Random Forest model is created and fitted to the training data. The trained model is used to extract feature importance's, and a DataFrame is produced to connect each feature with its importance. Finally, the code determines and outputs the top five numerical and categorical variables in order of importance. This research sheds light on the most relevant elements influencing emergency room wait times, assisting in the comprehension of the dataset and identifying interesting areas for additional investigation.

Result:

```
Top 5 Numeric Variables:
   Feature  Importance
1      LOV    0.228665
0  ARRTIME    0.126288
5    PULSE    0.058547
2      AGE    0.056900
4    TEMPF    0.055358
```

```
Top 5 Categorical Variables:
   Feature  Importance
1    VDAYR    0.202424
0   VMONTH    0.171488
14  IMMEDR    0.095173
16  INJURY    0.087849
3      SEX    0.072852
```

The Random Forest regression model's feature importance analysis gives crucial insights into the factors of emergency room wait times. Among the top numerical variables, the Length of Visit (LOV) is the most influential, indicating that patients with lengthy visits

had longer wait times, possibly reflecting more complex medical demands. Arrival Time (ARRTIME) is also relevant, implying that wait durations are influenced by the time of day, with peak arrival periods associated with longer waits. Physiological parameters such as pulse rate and patient age help to explain wait time variances, emphasizing the significance of patient condition in this setting. Furthermore, the patient's body temperature (TEMPF) is highlighted as a numerical component that influences wait times. On the categorical side, the weekday (VDAYR) and month of the year (VMONTH) are important, indicating temporal differences in emergency room demand. Furthermore, the urgency (IMMEDR), the type of the injury (INJURY), and the gender of the patient (SEX) are identified as important categorical variables, emphasizing the larger impact of social and health-related factors on emergency room wait times.