**Final Report**

Sankhepo Ndhlovu

Southern New Hampshire University

DAD 215: Introduction to SAS

Lee Edwards

August 13, 2023

* **Introduction**

Citi Bike is a bike sharing system in New York that makes its trip data available their trip data for anyone seeking insights into its operational systems. The data files contain a record for each ride, detailing start and stop station names and times, station and bike IDs, station latitude and longitude, as well as trip duration. Limited rider information, including gender, year of birth, and user type (subscriber or customer), is also provided.

* **Study Focus and Objectives**

This study aims to address several key questions: Which are the most frequently used stations? What is the average age of riders? How does the distribution of genders, age groups, and user types look? Mainly, this paper will analyze if the Utilization of the Citi Bike program has remained steady every July for the past two years in Brooklyn but not in Manhattan. Breakouts of the analysis will be shown by gender, age groups, and user subscription type.

The study outcomes will offer significant insights for decision-making regarding the expansion or reduction of the Citi Bike program across various geographical zones, including college campuses.

1. **INTRODUCTION TO THE SOFTWARE INTERFACE**

**Creation Of a Data Library**

The initial step for this analysis was to create a library name. This name is repeatedly used throughout the code to refer to the storage location of the datasets. This step ensures easy access to the datasets whenever needed within the code.

A screen shot of a computer program

Description automatically generated

1. WORKING WITH THE DATA

* **Importing And Converting the Data**

The Citi Bike data is first saved in Microsoft Excel Spreadsheets. For this project, we focused on datasets from 2013, 2014, and 2015. These datasets were brought into permanent SAS datasets by converting them from CSV files. This was done using the PROC IMPORT procedure. This procedure is designed to read and transform data from outside files, such as CSV, into SAS datasets. We also used the DBMS option to specify that the file type is CSV. Moreover, the OUT option was employed to define the name and place where the new SAS dataset will be stored.A close up of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

* **Formats and Functions Used to Transform Data:**

The first step involved cleaning the data by removing missing or inaccurate entries, such as birth years marked as '\N' or empty. In this step, variables were converted into their appropriate SAS formats of either numerical, categorical, or date formats.A white background with black text

Description automatically generated

Next, we computed the age of Citi Bike users by subtracting their birth year from the current year, using the ‘today()’ format.

A close-up of a text

Description automatically generated

The IF-ELSE conditional statements were then used to create the ‘AgeGroup’ variable, which allowed for individuals to be grouped according to their ages.

A close-up of a number

Description automatically generated

The SUBSTR function was used to extract the date and time components from the datetime strings. Then, the INPUT function was later applied to convert these extracted components into SAS date values. This enabled the creation of the two new variables—year and month.

A screenshot of a computer program

Description automatically generated

The first five observations of the dataset can be viewed using the PROC PRINT procedure.

A close up of a message

Description automatically generated

A screenshot of a computer screen

Description automatically generated

This was done for each year’s dataset (2013, 2014, 2015)

* **Syntax Or Logic Errors**

In the 2015 dataset, I encountered date values that were in the format ‘mm/dd/yyyy’ as opposed to the ‘mm-dd-yyyy’ formats in the years 2013 and 2014. I needed to extract the month and year from these dates for analysis, but the different formats were causing errors in my code as the ‘mm/dd/yyyy’ format was not recognized properly by the INPUT function due to the presence of slashes.

To address this issue, I modified the code to first replace the slashes in the date values with hyphens, and then used the modified date values for calculations. This gave me the ‘mm-dd-yyyy’ date format which was recognized by the INPUT function, which allowed me to extract the time and date parts and then create the month and year variables as required.

* **Sorting and Merging the Datasets**

Before merging, the three datasets were each sorted by their key identifiers using the PROC SORT procedure. The key ID variables used to merge the datasets were ‘usertype’, ‘birth\_year, ‘gender’.

A screenshot of a computer code

Description automatically generated

The datasets were merged using the MERGE statements by their common key IDs, which is important for the analysis of trends across each year. A KEEP statement was used to keep only relevant variables.A screen shot of a computer code

Description automatically generated

Within the dataset, the CATS function was used to merge two variables (start\_station\_name and end\_station\_name) to create a new variable (start\_end\_station).



Next, the IF-ELSE conditional formatting statement was used to create a new variable named ‘trip\_time’, based on trip duration. Arithmetic symbols were used to calculate and convert the trip duration from seconds to minutes, ensuring the new variable was created based on duration in minutes.A screenshot of a computer program

Description automatically generated

The first five observations of the merged dataset can be viewed using the PROC PRINT procedure.

A close-up of a text

Description automatically generated

A screenshot of a computer

Description automatically generated

* **Creating Multiple Datasets**

1. TripsCount: The dataset TripsCount was created to count the number of rides for each combination of the start and end station names of the bike rides. This was essential in identifying the most popular routes. The analysis was however narrowed down to only the first top 20 trips for each year. This enabled me to create breakouts of the dataset by borough type.
2. Top20Trips: This dataset was created to identify and store information about the top 20 most frequent trips based on the combination of start and end station names. It sorts the TripsCount dataset (which contains the count of rides for each start-end station combination) in descending order based on the ride count. Then it selects the first 20 observations within each year, effectively identifying the most popular trips in terms of ride counts for each year.
3. Stations\_boroughs: The stations\_borough dataset was created to map each station name to a borough. It is created using a SELECT statement to assign borough names to specific station names. The dataset is used to replace station names with their corresponding boroughs in the **merged\_datasets** dataset, which allows for the data to be combined at the borough level.
4. Merged\_dataset\_with\_boroughs: This dataset was created to merge the ‘merged\_datasets’ with the ‘stations\_borough’ dataset, which allows for the data to be analyzed and visualized at the borough level.
5. Summary\_counts: The summary\_counts dataset was used to store the counts of observations for each combination of Year, usertype, gender, age group, and borough that were calculated using the PROC SUMMARY procedure.
6. Summary percentages: This dataset calculates and stores percentages for the categories, based on the counts in the ‘summary\_counts’ dataset. It calculates the percentage of each count relative to the total count for each year. The percentages are used for creating grouped bar charts to understand how different breakouts (usertype, gender, age group) contribute to bike utilization over time.
7. **UNDERSTANDING THE DATA**

* **Analysis Done to Summarize The Data**

Mean Age in each year:

The PROC MEANS procedure was used to calculate the average age for each year.

A screenshot of a computer code

Description automatically generated

A close-up of a receipt

Description automatically generatedA screenshot of a computer screen

Description automatically generated

A close-up of a receipt

Description automatically generated

The findings suggest that the average age was 48 years in 2013, 47 years in 2014, and 46 years in 2015. The youngest riders were 26 years old in 2013, 25 years old in 2014, and 24 years old in 2015. On the other end, there were some extreme values, like ages of 124 in 2013 and 2014, and 138 in 2015. These high ages seem improbable, indicating potential data errors. It's important to note that these ages were calculated by subtracting the birth year from the current year. The extremely high ages suggest that these individuals might not be alive, given that these ages surpass a century.

Means of breakout groups within Merged Datasets:

The PROC MEAN procedure was also used to calculate statistical measures of both numerical and categorical variables. This was done using a CLASS statement to specify categories and a VAR statement to specify numerical variables.

A computer code with black text

Description automatically generated

A table with numbers and letters

Description automatically generated

The most chosen subscription type for all genders was the subscription plan (Annual members). Among customer subscribers (24-hour pass or single ride user), there were 28 instances in the unknown gender group, 10 instances among males, and just 3 instances among females. Within the unknown gender category, most of these users fell within the 36-49 age range. For males (Gender = 1), the majority were aged between 36 and 49, as well as those aged 50 and above. Among females (Gender = 2), the majority were also in the 36-49 age range. Overall, there were more male users who utilized the Citi Bike program than females and those with unknown gender.

* **Creating Graphs**

Top 20 Trips Bar Chart:

This chart was created using the TripsCount and Top20Trips datasets to identify the top 20 most popular trip routes, based on the number of rides for each year. PROC GCHART was used to create the horizontal bar chart in descending order of ride count, with start\_station\_name on the y-axis and ride\_count on the x-axis. Each bar represents a trip route.

A close up of a computer screen

Description automatically generated

A graph of a number of trips

Description automatically generated

The most traveled route with the highest ride counts in 2013 originated from the Pershing Square North station, followed by W 21 St & 6 Ave station, and then West Thames St station.

A graph of a trip

Description automatically generated

The most traveled route with the highest ride counts in 2014 originated from the West Thames St station, followed by Avenue D & E 12 St station, and then E 13 St & Avenue A station.

A graph with blue and black lines

Description automatically generated

The most traveled route with the highest ride counts in 2013 originated from the 1 Ave & E 44 St station, followed by 11 Ave & W 59 St station, and then W 45 St & 6 Ave station.

Station to Borough Bar Chart:

Station names were categorized into boroughs using a SELECT statement from the top 20 trips taken each year. While this method works, it is worth noting that it might not be convenient for larger datasets. The PROC GCHART procedure was used to create a vertical bar chart for the stations\_boroughs dataset. These charts were used to compare the number of rides taken in the two boroughs for each year.

A graph with a bar and a number of columns

Description automatically generatedA graph of a bar

Description automatically generatedA graph showing a bar

Description automatically generated

From these graphs, it's evident that Manhattan consistently recorded the highest number of rides each year.

Grouped Bar Charts by Category (Age Group, Gender, User Type):

Three stacked bar charts were created using PROC GCHART using the summary\_percentage datasets. These charts were used to show the utilization patterns of Citi Bike in the boroughs of Brooklyn and Manhattan, based on the categories of age group, gender, and user type for each year. The x-axis represents the boroughs, the y-axis represents the percentage of utilization, and bars are grouped by the chosen category.

Borough vs. Age Group and YearA graph of a bicycle race

Description automatically generated

Borough vs. Gender and Year, where 0 = unknown, 1 = Male, 2 = Female.

A graph of a number of people

Description automatically generated

Borough vs. User Type and Year

A graph of a number of people

Description automatically generated

It's evident that Manhattan led in the percentage of rides for all three years. In 2013, it recorded the highest number of rides, and this trend continued with a relatively closer percentage to Brooklyn’s rides in 2014 and 2015, although still higher. In contrast, Brooklyn consistently had a lower percentage of rides compared to Manhattan.

In the year 2013, riders aged 36–49 took the most trips, just slightly surpassing those in the 50+ age group. Moreover, the male gender group (gender 1) made the most use of Citi Bike rides in both 2013 and 2014. Interestingly, all user types remained subscribers across all three years. It's important to highlight that this analysis is based on the top 20 trips within each respective year.

* **Summary**

1. Station Popularity and Utilization:

The analysis revealed that Manhattan consistently dominated in terms of bike ride utilization across the years 2013 to 2015. This indicates a higher demand and utilization in Manhattan. Further exploration can be conducted to understand the specific routes or areas within Manhattan that contribute to this high utilization, but it could be potentially due to its urban nature and tourist attractions. This insight can inform station placement strategies for more equitable distribution to boost Brooklyn's utilization.

1. Demographic Insights:

Gender-wise, the data indicated a significant gender gap, with males utilizing the Citi Bike program more than females. Understanding the different preferences and needs of male and female riders can lead to more inclusive and effective bike-sharing services.

The analysis of age groups revealed that the age category of 36-49 consistently had the highest ridership across the years. This age group's utilization for the program might suggest that it caters well to the needs of the working population, possibly as a convenient mode of transportation for their daily commutes.

1. User Type Insights:

Almost all riders had a subscription type of 'subscriber' throughout the three years analyzed. This finding highlights the program's success in having loyal subscribers, who likely find it more convenient and cost-effective to subscribe for the whole year. The limited 'customer' users suggests an opportunity for targeted marketing for the program to attract more casual or occasional riders.

1. Temporal Utilization Trends:

It was observed that while Brooklyn's utilization remained relatively steady, Manhattan's utilization showed fluctuations, as it was high in 2013 and was relatively lower in 2014 and 2015, despite still being higher than Brooklyn. This trend could be attributed to various factors like weather conditions or changing tourism dynamics.

* **Conclusion**

Based on the analysis outcomes, it can be concluded that colleges could indeed benefit from the addition or expansion of bike-sharing installations near their campuses. There is a demand for the Citi Bike utilization in both boroughs and given the fact that many college campuses are located in denser urban environments like Manhattan, there is a higher probability that installing the bike-sharing program would be used frequently. Additionally, given that utilization is spread out across different age groups, it is most likely that Citi Bike would be attractive to students, faculty, and staff.

* **Errors or Difficulties Encountered**

During the coding process, several errors and difficulties were encountered while trying to manipulate date and time values to extract the date part and time part to create the variables ‘Month’ and ‘Year’, especially when using the INPUT function. This issue was resolved by employing the SUBSTR function to extract the specific dates and time strings and then use the INPUT function to convert the parts into numeric values.