**Task 1: Data Cleaning and Profiling**

Joseph Emmanuel Cayetano

Western Governors University

D599: Data Preparation and Exploration Task 1

October 8, 2024

**A1a. Describe the general characteristics of the initial dataset**

The dataset to be cleaned in this task is named “Employee Turnover Dataset.xlsx.” This dataset contains demographic information, job roles, job performances and salaries for all employees working in a large multinational technology firm. The data has 10319 employee records and 35 variables. The variable names are: Age, Turnover, BusinessTravel, DailyRate, Department, DistanceFromHome, Education, EducationField, EmployeeCount, EmployeeNumber, EnvironmentSatisfaction, Gender, HourlyRate, JobInvolvement, JobLevel, JobRole, JobSatisfaction, MaritalStatus, MonthlyIncome, MonthlyRate, NumCompaniesWorked, Over18, OverTime, PercentSalaryHike, PerformanceRating, RelationshipSatisfaction, StandardHours, StockOptionLevel, TotalWorkingYears, TrainingTimesLastYear, WorkLifeBalance, YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion, and YearsWithCurrManager. These are the variables that will be observed and cleaned for future analysis and model building.

**A1b. Indicate the data type and data subtype for each variable**

A screenshot of a computer

Description automatically generated

The dataset has 35 variables. Age is a numeric data type that has continuous data values with a float64 subtype. Turnover is a categorical data type that has nominal data values with an object subtype. BusinessTravel is categorical data type that has nominal data values with an object subtype. DailyRate is a numeric data type that has discrete data values with an int64 subtype. Department is a categorical data type that has nominal data values with an object subtype. DistanceFromHome is a numeric data type that has discrete data values with an int64 subtype. Education is a categorical data type that has ordinal data values with an int64 subtype. EducationField is a categorical data type that has nominal data values with an object subtype. EmployeeCount is a numeric data type that has discrete data values with an int64 subtype. EmployeeNumber is a numeric data type that has discrete data values with an int64 subtype. EnvironmentSatisfaction is a categorical data type that has ordinal data values with an int64 subtype. Gender is a categorical data type that has nominal data values with an object subtype. HourlyRate is a numeric data type that has discrete data values with an int64 subtype. JobInvolvement is a categorical data type that has ordinal data values with an int64 subtype. JobLevel is a categorical data type that has ordinal data values with an int64 subtype. JobRole is a categorical data type that has nominal data values with an object subtype. JobSatisfaction is a categorical data type that has ordinal data values with an int64 subtype. MaritalStatus is a categorical data type that has nominal data values with an object subtype. MonthlyIncome is a numeric data type that has continuous data values with a float64 subtype. MonthlyRate is a numeric data type that has continuous data values with a float64 subtype. NumCompaniesWorked is a numeric data type that has continuous data values with a float64 subtype. Over18 is a categorical data type with nominal data values with an object subtype. OverTime is a categorical data type with nominal data values with an object subtype. PercentSalaryHike is a numeric data type that has discrete data values with an int64 subtype. PerformanceRating is a categorical data type that has ordinal data values with an int64 subtype. RelationshipSatisfaction is a categorical data type that has ordinal data values with an int64 subtype. StandardHours is a numeric data type that has discrete data values with an int64 subtype. StockOptionLevel is a categorical data type that has ordinal data values with an int64 subtype. TotalWorkingYears is a numeric data type that has continuous data values with a float64 subtype. TrainingTimesLastYear is a numeric data type that has continuous data values with a float 64 subtype. WorkLifeBalance is a categorical data type that has ordinal data values with an int64 subtype. YearsAtCompany is a numeric data type that has discrete data values with an int64 subtype. YearsInCurrentRole is a numeric data type that has discrete data values with an int64 subtype. YearsSinceLastPromotion is a numeric data type that has continuous data values with a float64 subtype. YearsWithCurrManager is a numeric data type that has discrete data values with an object subtype.

**A1c. Provide a sample of observable values for each variable**

I will be providing a sample of observable values for each variable. The Age variable has sample values of 27, 31, 19, 30, 21. The Turnover variable has sample values of ‘No’, ‘Yes’, ‘Yes’, ‘Yes’, ‘No.’ The BusinessTravel variable has sample values of ‘Non-Travel’, ‘Travel\_Frequently’, ‘Travel\_Rarely, ‘Non-Travel’, ‘Non-Travel.’ The DailyRate variable has sample values of 1444, 459, 1259, 1153, 339. The Department variable has sample values of ‘Sales’, ‘Hardware’, ‘Support’, ‘Software’, ‘Human Resources.’ The DistanceFromHome variable has sample values of 40, 22, 20, 34, 10. The Education variable has sample values of 5 (post-graduate), 1, 3, 2, 4. The EducationField variable has sample values of ‘Technical Degree’, ‘Other’, ‘Human Resources’, ‘Life Sciences’, ‘Marketing.’ The EmployeeCount variable has sample values of 1, -1, 3, 1, 1. The EmployeeNumber variable has sample values of 5504, 105, 7602, 257, 8434. The EnvironmentSatisfaction variable has sample values of 4 (highly satisfied), 1, 3, 2, 4. The Gender variable has sample values of ‘Male’, ‘Male’, ‘Female’, ‘Female’, ‘Male.’ The HourlyRate variable has sample values of 97, 35, 101, 97, 132. The JobInvolvement variable has sample values of 4 (very involved), 3, 2, 1, 4. The JobLevel variable has sample values of 1 (entry level), 2, 4, 3, 5. The JobSatisfaction variable has sample values of 4 (highly satisfied), 2, 3, 1, 4. The Marital Status variable has sample values of ‘single’, ‘divorced’, ‘married’, ‘single’, ‘divorced’. The MonthlyIncome variable has sample values of 48289, 39266, 23402, 29244, 28109. The MonthlyRate variable has sample values of 1374462, 527440, 1359570, 61029, 277944. The NumCompaniesWorked variable has sample values of 3, 0, 6, 7, 4. The Over18 variable has sample values of ‘Y’, ‘Y’, ‘Y’, ‘Y’, ‘Y’. The OverTime variable has sample values of ‘No’, ‘No’, ‘Yes’, ‘No’, ‘Yes’. The PercentSalaryHike variable has sample values of 11, 30, 30, 33, 35. The PerformanceRating variable has sample values of 4 (exceeds expectation), 2, 1, 2, 1. The RelationshipSatisfaction variable has sample values of 4 (highly satisfied), 2,1,3, 4. The StandardHours variable has sample values of 80, 80, 80, 80, 80. The StockOptionLevel variable has sample values of 4 (maximum), 3, 2, 1, 4. The TotalWorkingYears variable has sample values of 29, 11, 37, 21, 25. The TrainingTimesLastYear has sample values of 4, 5, 6, 3, 2. The WorkLifeBalance variable has sample values of 4 (very satisfactory), 4, 2, 4, 1. The YearsAtCompany variable has sample values of 8, 8, 3, 17, 26. The YearsInCurrentRole variable has sample values of 10, 4, 1, 12, 3. The YearsSinceLastPromotion has sample values of 1, 1 ,1, 1, 5. The YearsWithCurrManager has sample values of 17, 1, 1, 20, 3.

**B1. Explain how you inspected the dataset for each of the quality issues listed in part B**

I will be providing an image of both the code and output results for each quality issue, and explaining how I inspected each quality issue in the dataset.

A screenshot of a computer

Description automatically generated

For the quality issue of duplicate entries in the dataset, I used the duplicated() function from pandas to return 'True' if the row is a duplicate. Then, I filtered the data to output only the rows that are 'True'. There are 298 duplicate rows in the dataset, and they need to be removed to ensure data integrity.

A screenshot of a computer screen

Description automatically generated

For the quality issue of missing values, I used the isnull() function from pandas to return 'True' if the cell contains 'NaN' or a missing value. I also used the sum() function to count the total number of 'True' values or missing values in each column. There are 9 variables with missing values, which are: Age, EducationField, Gender, MonthlyIncome, MonthlyRate, NumCompaniesWorked, TotalWorkingYears, TrainingTimesLastYear, and YearsSinceLastPromotion. The column TrainingTimesLastYear has the highest number of missing values with 418.

A screenshot of a computer

Description automatically generated

For the quality issue of formatting errors, I used the head() function for each column in the data to check a sample of its values and whether its data type make sense with its values. After inspecting, there are columns in the data that should have an int data type instead of a float. For example, the Age column should have an int data type because age is discrete and is usually represented as a whole number. We won't be measuring values precisely, so decimal values won't be needed. This also applies to these columns: NumCompaniesWorked, TotalWorkingYears, TrainingTimesLastYear, and YearsSinceLastPromotion. These columns will be converted to an int data type.

I used the unique() function from pandas to return the unique values in the BusinessTravel column and check whether it contains yes or no values like it is stated in the data dictionary. The column did not contain any yes or no values but instead, it contains travel frequency values. The values for the BusinessTravel will be changed to yes or no, since that is what stated in the data dictionary.

A screenshot of a computer program

Description automatically generated

For the quality issue of inconsistent entries, there are values from multiple columns in the data that do not make sense. For example, looking at the TotalWorkingYears and YearsAtCompany columns, I created a condition to return rows where the TotalWorkingYears is less than YearsAtCompany. It returned a single row where it has a negative number of TotalWorkingYears and is less than YearsAtCompany. It is impossible to have a greater number of years working at a single company than the total number of working years. This is an inconsistency in the data, and it must be fixed.

There are also two columns in the data that are inconsistent. The first column is EmployeeCount. The EmployeeCount value for each row should be 1. I created a condition where it returns rows if the EmployeeCount values is not equal to 1. Any returned rows indicate an inconsistency. Another inconsistent column is MonthlyIncome. The values in the MonthlyIncome column should always be positive because there is no such thing as negative income. I created a condition where it returns rows if the MonthlyIncome values is less than 0. Any returned rows indicate an inconsistency.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

For the quality issue of outliers, there are multiple columns in the data that need to be examined for outliers, and they are Age, DistanceFromHome, and TotalWorkingYears columns. For these columns, I used a box plot and the interquartile range (IQR) method to identify outliers. The IQR is a very useful method, as it specifically tells the value and the row where the outlier is from.

**B2. List your findings for each quality issue listed in part B**

I will be providing an image of both the code and output results for each quality issue and explaining the output results for each quality issue.

A screenshot of a computer

Description automatically generated

After executing the code with the duplicated() function, it returns all the duplicate entries in the data. There are 298 duplicate rows, and they need to be removed to have consistent data results.

A screenshot of a computer screen

Description automatically generated

After executing the code with the isnull() function and sum function(), it returned the total missing values for each column. There are 9 columns with missing values, which are: Age, EducationField, Gender, MonthlyIncome, MonthlyRate, NumCompaniesWorked, TotalWorkingYears, TrainingTimesLastYear, and YearsSinceLastPromotion. The column TrainingTimesLastYear has the highest number of missing values with 418. These columns will be filled with values.

A screenshot of a computer

Description automatically generated

Using the head() function, we get to see sample values for the columns Age, NumCompaniesWorked, TotalWorkingYears, TrainingTimesLastYear, and YearsSinceLastPromotion. In the Age column, its values have decimal points because it is a float data type. Since we are not going to measure age values precisely, decimals won't be needed, which is why the Age column needs to be converted to an int data type. In the NumCompaniesWorked, its values have decimal points since it is a float data type. Counting the total number of companies worked is usually represented as a whole number, which is why it will be converted to an int data type. In the TotalWorkingYears column, decimals are present in its values. Counting the total number of working years is usually represented as a whole number, which is why the column will be converted to an int data type. In the column TrainingTimesLastYear, its values have decimal points. This column should be a discrete variable, which is why it will be converted to an int data type. In the YearsSinceLastPromotion, decimals points are present in its values. YearsSinceLastPromotion is usually represented as a whole number, so it will be converted to an int data type.

Using the unique() function for the BusinessTravel column, we get to see its unique values. As you can see, the column contains string and numeric values, which is a bit confusing. We will be changing its values to yes or no for simplicity.

A screenshot of a computer program

Description automatically generated

After creating a condition where TotalWorkingYears is less than YearsAtCompany, it returns a row that has a negative value in the TotalWorkingYears. This indicates inconsistency since it is not possible to have a negative total number of working years, and it is less than YearsAtCompany. In the EmployeeCount column, another inconsistency is found. All values in the EmployeeCount columns should be 1, but it returned rows with -1 and 3 values. In the MonthlyIncome column, it returned a row with a negative value. Since it is impossible to have a negative monthly income, this indicates inconsistency.

A screenshot of a computer

Description automatically generated

Looking at the boxplot for the Age column, there are two significant outliers. Using the IQR method, we can see the specific values of these outliers, and they are 148 and 96. These values of ages do not seem right.

A screenshot of a computer

Description automatically generated

As for the boxplot for the DistanceFromHome, there are three significant outliers. Using the IQR method, we see that the values of these outliers are 3737, 3535, and 978. These values are outliers, and they need to be fixed.

A screenshot of a computer screen

Description automatically generated

As for the boxplot for the TotalWorkingYears column, there is one significant outlier. With the help of the IQR method, we identified that the value of the outlier is 222. It is impossible to have 222 total working years. This outlier will need to be fixed.

**C1. Describe how you modified the dataset after identifying each quality issue listed in B**

A screenshot of a computer

Description automatically generated

I modified the dataset by dropping duplicate rows using the drop\_duplicates() function. By default, this function keeps the first occurrence of each duplicate row in the data. After running the duplicated() function, the data does not contain duplicate rows anymore.

A screenshot of a computer code

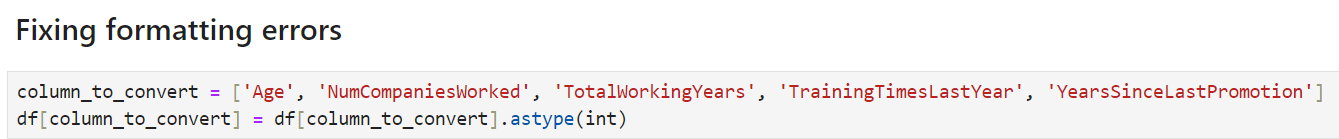
Description automatically generated

I modified the dataset by filling in missing values with mean, median, and mode. For the Age column, I replaced its missing values with the median age, which is 39. I was able to replace it using the fillna() function and inserting median age into the function as a parameter. For the EducationField, I replaced its missing values with the most frequent value, which is marketing. I used the fillna() function and inserted the most frequent EducationField value into the function as a parameter. For the Gender column, I replaced its missing values with the most frequent gender value, which is male. I used the fillna() function and inserted the most frequent Gender value into the function as a parameter. For the MonthlyIncome column, I replaced its missing values with the mean of all MonthlyIncome values. I used the fillna() function and inserted the mean into the function as a parameter. For the MonthlyRate column, I replaced its missing values with the mean of all MonthlyRate values. I used the fillna() function and inserted the mean into the function as a parameter. For the NumCompaniesWorked column, I replaced its missing values with the median of all NumCompaniesWorked values. I used the fillna() function and inserted the median into the function as a parameter. For the TotalWorkingYears column, I replaced its missing values with the median of all TotalWorkingYears values. I used the fillna() function and inserted the median into the function as a parameter. For the TrainingTimesLastYear, I replaced its missing values with the median of all TrainingTimesLastYear values. I used the fillna() function and inserted the median into the function as a parameter. For the YearsSinceLastPromotion, I replaced its missing values with the median of all YearsSinceLastPromotion values. I used the fillna() function and inserted the median into the function as a parameter.

A screenshot of a computer

Description automatically generated

Checking the total number of missing values in the dataset again, we can see that every column is now filled, and there are no more missing values.



A screenshot of a computer

Description automatically generated

I modified the dataset by converting the data types of columns Age, NumCompaniesWorked, TotalWorkingYears, TrainingTimesLastYear, and YearsSinceLastPromotion from float to int using the astype() function and inserting int to the function as a parameter. These columns now have an int data type that prevents decimal values.

A screenshot of a computer program

Description automatically generated

A close up of a computer screen

Description automatically generated

I modified the values of the BusinessTravel column by replacing the original values 'Travel\_Rarely' and 'Travel\_Frequently to 'Yes’ and replacing the rest of the values to 'No'. This was done using the map() function.

A screenshot of a computer

Description automatically generated

EmployeeNumber 9994 has a negative value of TotalWorkingYears, which is an inconsistent entry in the data because an employee can never have a negative total amount of working years. If we try to replace the value with the mean of TotalWorkingYears, the value will be higher than the Age value, which is also an inconsistent entry. Therefore, we will drop this row instead.

A screenshot of a computer

Description automatically generated

Since there were multiple rows that did not have an EmployeeCount value of 1, their EmployeeCount values must be updated. I was able to update their values by using the .loc indexer which accesses and modifies column values of specific rows. The EmployeeCount value was updated to 1 for employees with numbers 8115, 21, and 194.

A screenshot of a computer code

Description automatically generated

I modified this inconsistent row with a negative MonthlyIncome value by replacing it with the mean of all the MonthlyIncome values. I used the .loc indexer to access and modify the MonthlyIncome value of a specific EmployeeNumber.

A screenshot of a computer program

Description automatically generated

Before modifying the outliers, there were outliers in the Age column with values 148 and 96. Instead of removing the outliers completely, I replaced their values with the median of all the Age values. I was able to replace it by using the .loc indexer to access and modify the rows where the Age values 148 and 96 were from and replaced it with the median age.

A screenshot of a computer

Description automatically generated

Before modifying the outliers, there were outliers in the DistanceFromHome column with values 3737, 3535, and 978. Instead of removing these outliers, I replaced the values with the median of all DistanceFromHome values. I used the .loc indexer to modify the rows where the DistanceFromHome values 3737, 3535, and 978 were from, and replaced it with the DistanceFromHome median value.

A screenshot of a computer program

Description automatically generated

Before modifying the outliers, there was an outlier in the TotalWorkingYears column with the value 222. Instead of removing this outlier, I replaced the value with the median of all TotalWorkingYears values. I used the .loc indexer to modify the row where the TotalWorkingYears value 222 was from and replaced it with the TotalWorkingYears median value.

**C2. Discuss why you chose the specific data cleaning techniques you used to clean the quality issues listed in part B**

A screenshot of a computer

Description automatically generated

I removed all the duplicate rows in the dataset to prevent skewed analysis and incorrect statistical conclusions, and to improve the querying performance of the dataset. Duplicate rows should always be removed to have a reliable and accurate data result.

A screenshot of a computer code

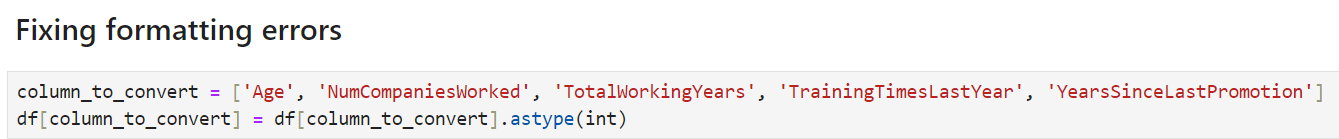
Description automatically generated

I replaced the missing values in the Age column with the median value of all Age values because the median is less affected by outliers, and the median value returns a whole number. If I used the mean instead, the missing values would get values with decimal places, which is not ideal when measuring age. In the EducationField column, I replaced its missing values with the mode, which is the most frequently occurring value in the EducationField column. I used the mode because it is ideal for categorical column like the EducationField column, and it helps maintain the original distribution of the dataset. For the Gender column, I used mode to replace its missing values. I replaced the missing values with the value 'male' in the Gender column since it is the value that appears frequently, and using the most frequent value is perfect for a categorical variable like Gender. In the MonthlyIncome column, I replaced its missing values with the mean value of all MonthlyIncome values because using the mean value is ideal for columns that are normally distributed like the MonthlyIncome column. Also, the MonthlyIncome column is a float data type so having a mean value with decimals is acceptable. For the MonthlyRate column, I used the mean value to replace its missing values because the column is normally distributed, and it is usually ideal to use the mean value for columns like income. In the NumCompaniesWorked column, its missing values were replaced with the median value of all NumCompaniesWorked values to maintain the column's original distribution. Also, if the mean value was used instead, it would return a value with decimals, which is not ideal for a discrete variable like NumCompaniesWorked. For the TotalWorkingYears column, its missing values were replaced with the median value of all TotalWorkingYears values. I used the median value because the TotalWorkingYears column is a discrete variable that usually represents whole numbers. If I used the mean value, I would get a value with decimals, which is not ideal for discrete variables. In the TrainingTimesLastYear column, I used the median value to replace its missing values to preserve the column's original distribution, and since it is a discrete variable, it is more ideal to use the median value. For the YearsSinceLastPromotion column, I replaced its missing values with the median value because it is a discrete variable that usually represents whole numbers, and decimal values is not required in the context of this column.

A screenshot of a computer

Description automatically generated

Essentially, I replaced all the missing values in the dataset with mean, mode, and median values rather than deleting the missing rows completely to preserve the dataset's size, avoid biases, and maintain data integrity.



A screenshot of a computer

Description automatically generated

I converted the Age, NumCompaniesWorked, TotalWorkingYears, TrainingTimesLastYear, and YearsSinceLastPromotion columns from float to an int data type because these columns typically represent whole numbers, which align with their real-world interpretation. Having decimal values in discrete columns do not make sense. Also, it is much easier to understand and visualize discrete values in the context of these columns.

A screenshot of a computer program

Description automatically generated

A close up of a computer screen

Description automatically generated

I changed the values of the BusinessTravel column to 'yes' or 'no' because the column had both string and numeric values, which was confusing. By using only, the values 'yes' or 'no', it makes the data simpler and easier to understand.

A screenshot of a computer

Description automatically generated

I decided to remove the row that contains a negative total amount of working years because keeping an inconsistent entry like this one can compromise the overall quality of the data. By dropping inconsistent rows, it prevents analytical errors and facilitates accurate insights. Also, replacing the negative value with the mean, mode or median won't work because Age must be taken into consideration.

A screenshot of a computer

Description automatically generated

I changed the EmployeeCount value to 1 for employee numbers 8115, 21, and 194 because each row in the dataset has an EmployeeCount value of 1. It is important to have consistent data to help make accurate insights. I did not drop the rows because the modification is not that significant, and the rows may contain useful information.

A screenshot of a computer code

Description automatically generated

I replaced the row with a negative MonthlyIncome value with the mean of all MonthlyIncome values to prevent bias in the analysis and preserve the overall average of the dataset. It is impossible to have a negative monthly income value, so it had to be fixed. I did not drop the row because it may contain useful information in the analysis.

A screenshot of a computer program

Description automatically generated

In the Age column, I fixed the outliers by replacing its Age values with the median value of all Age values. The reason for replacement is not only to remove the outliers in the Age column, but to also preserve the original distribution of the data. I also avoided dropping the outliers completely, as the rows with outliers still have valid data for the rest of the columns in the dataset. Additionally, using the median value for a discrete variable like age is more ideal than using the mean value.

A screenshot of a computer

Description automatically generated

In the DistanceFromHome column, I fixed the outliers by replacing its DistanceFromHome values with the median value of all DistanceFromHome values. I used the median value to maintain the data distribution of the column and for consistency. I avoided dropping the rows where the outliers were from because they still contain useful data for other columns.

A screenshot of a computer program

Description automatically generated

In the TotalWorkingYears column, I fixed the outlier by replacing its TotalWorkingYears value with the median value of all TotalWorkingYears values. I used the median value because it is ideal for discrete variables like TotalWorkingYears. By replacing the value, we got rid of the outlier and still retained the row which the outlier was from.

**C3. Describe two or more advantages to your data cleaning approach specified in part C1**

There are multiple advantages to my data cleaning approach from part C1. The first advantage is I retained most data points by replacing missing values with mean, median, and mode values instead of just removing them completely. This imputation method helped “preserve valuable information contained in the missing values, leading to more accurate and reliable analysis” (Chandrikasai, 2023, par. 3). The second advantage is I improved the reliability of the data and prevented bias by eliminating duplicate rows and outliers. Both duplicate rows and outliers can negatively impact the quality of data, which is why it is important to remove them. The third advantage is I improved data representation by converting columns that usually represent whole values into an integer data type. This method also enhances data performance and accuracy.

**C4. Discuss two or more limitations to your data cleaning approach specified in part C1**

There are two limitations regarding my data cleaning approach from part C1. The first limitation is I may have distorted the relationship between variables. For example, since I have replaced some monthly income values with the median value, I may have ignored the relationship between MonthlyIncome and another variable where MonthlyIncome is correlated with. This could lead to inaccurate estimates. The second limitation is I may have flagged some values as outliers and replaced them with the median value, even though the original values were legitimate and real. This can be seen where I have replaced the values of multiple outliers in the DistanceFromHome column with the median value, and not considering that they may be remote workers, which explains why their DistanceFromHome values were very high from the average.

**D1. Provide a data cleaning report as a document file that includes responses to task prompts**

This Word document file “D599 Task 1 Final” is the data cleaning report that includes responses to task prompts.

**D2. Provide the annotated code you used to detect and mitigate the data quality as an executable script file**

I have attached the .ipynb file where I have performed all my data cleaning techniques. The file is called “D599 Task 1”, and it is a Jupyter Notebook file.

**D3. Provide a copy of the cleaned dataset as a CSV file**

I have attached the cleaned dataset to my submission. The CSV file is called “cleaned\_employee\_turnover\_dataset.”

**D4. Panopto Recording**

Here is my Panopto link: https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=da9ac042-4034-4dd7-b4b6-b20400f471d3

References

Chandrikasai. (2023, April 11). *Imputing missing values is another technique used to handle missing data in a dataset*. Medium. https://medium.com/@chandrikasai9997/imputing-missing-values-is-another-technique-used-to-handle-missing-data-in-a-dataset-824957ce71b4