1. **Introduction and Objectives**

IBM has confront a seriously commercial challenge in terms of staff churn. Understanding the primary variables driving attrition is critical for designing successful retention tactics.

This article describes the data cleaning process used to prepare the provided IBM HR dataset (PAVANSUBHASH, 2017) for the proposed analytical and visualization purposes. The purpose is to guarantee that the data is reliable, consistent, and appropriate.A nine-step cleaning method was created using Python and the pandas and scikit-learn libraries.

1. **Data Loading and Initial Inspection**

Loading raw IBM HR data and understanding its initial structure, in line with the dataset overview in the proposal.

Code implementation in this section is as follows: For initial investigation, the pandas library was utilized, including loading using pd.read\_csv(), df.head(), and df.info().

**Finding:** The dataset loaded successfully, validating the 1470 records and 35 columns indicated in the proposal, which included a combination of quantitative and categorical data important to HR research (for example, MonthlyIncome, JobRole, Age, YearsAtCompany).

**Justification:** This first step ensures that the dataset matches the one outlined in the proposal and serves as a baseline for determining the necessary cleaning operations to support the planned attrition factor analysis.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 1: Output result for step 1

1. **Handling Missing Values**

First off, we checked the data carefully to make sure it was accurate and didn't have any missing pieces. Gaps in data can really skew comparisons, especially when looking at attrition rates across different teams or roles. We used df.isnull().sum() to hunt down any missing values.

A black screen with colorful text

AI-generated content may be incorrect.

Figure 2: Output result for step 2

It turned out the result that the dataset was already completed itself, with no missing values found! We don't have to fill in any gaps or remove any data.

This is great because any charts comparing attrition or looking at things like satisfaction will use data from the entire team, giving us a solid for the insights we require.

1. **Handling Duplicates**

Next, we used df.duplicated() to look for duplicate employee records.sum(). This was required to avoid mistakenly counting persons twice and skewing our attrition statistics.

Resulting that we found no duplicate entries in the data. That implies no cleaning was necessary on this step.

A black rectangular object with white text

AI-generated content may be incorrect.

Figure 3: Output result for step 3

With different records established, we can be confident that the data is trustworthy and that our planned study of attrition rates and correlations will be accurate.

1. **Handling Columns with Single Unique Values**

The next stage in cleaning the data set was removing the columns that provided no new or unneeded information. We have used df.nunique() and d.drop() to address the same information and eliminated. If a value remains constant, it cannot explain why some employees depart while others remain.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 4: Output result for step 4

We found three columns like this: EmployeeCount, Over18, and StandardHours. Because these were constant for everyone, they weren't useful for understanding attrition. So, I removed these columns from the dataset.

This makes the data simpler and lets us focus our analysis and visuals, like the bar charts and heatmaps we plan to make, on the columns that might actually show differences, such as JobRole or JobSatisfaction

1. **Outlier Detection (Example: MonthlyIncome)**

An important part of preparing the data was looking for outliers, which are extremely high or low values, in the number-based columns. We focused on MonthlyIncome since understanding its link to attrition is key to my proposal.   
These extreme values could potentially skew our results or visualizations like scatter plots, so it's good to identify them. We used the Interquartile Range (IQR) method to do this for MonthlyIncome. This involved calculating the 25th percentile (Q1) and 75th percentile (Q3) of income, finding the difference (IQR), and then setting limits to see which income values fell too far below Q1 or too far above Q3

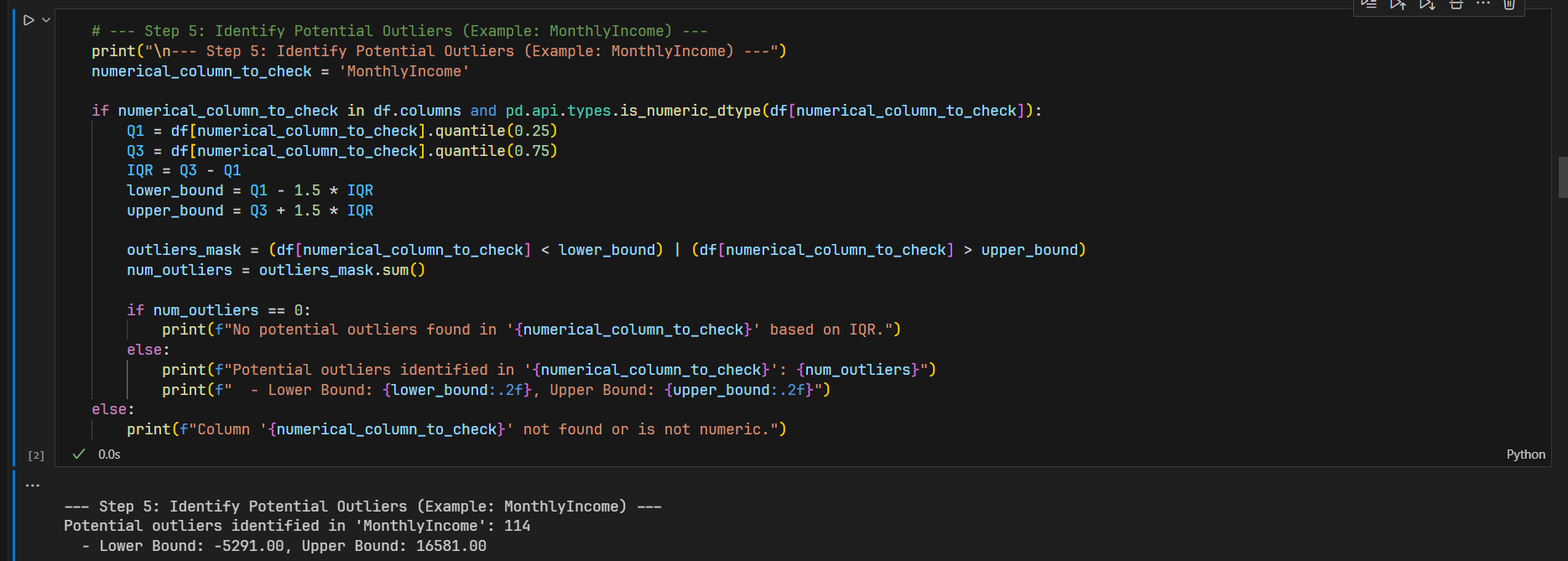


Figure 5: Output result for step 5

**Result:**

* Potential high-income outliers were detected (values somewhat greater than 16581).  
  Action & Justification: Outliers were detected, but not automatically deleted. This is consistent with the proposal's purpose of studying linkages (for example income vs attrition).
* Extreme values may indicate legit high-earning executives or unique positions with significant turnover trends.
* Handling requires careful attention throughout the visualization step (for example, using log scales, robust statistics, charts like the suggested scatter plot), than blank removal during the cleaning stage.

1. **Data Transformation: Encoding Categorical Variables**

Our main goal here was to get the key categorical details about employees ready for analysis. We're talking about factors identified earlier like Job Role, Department, Marital Status, Business Travel, Gender, Education Field, and Overtime status. To run correlations, generate heatmaps, or use these for filtering in the dashboard, they needed to be in a numerical format.

**Code implementation:** identified the text-based columns using df.select\_dtypes() and converted these using One-Hot Encoding (pd.get\_dummies(..., drop\_first=True)), which creates binary (0/1) columns for each category and also mapped the 'Attrition' column ('Yes'/'No') directly to 1/0 integers, making it ready for calculations and use as our target variable. This help to transform the necessary text columns into numbers.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 6: Output result for step 6

This step is so essential because it makes possible all of the quantitative analysis and visualizations that we described latter on. By treating factors like "Department" or "Job Role" numerically, we can produce comparison bar charts that display attrition rates or investigate combinations of factors in heatmaps that directly support the project's goals.

1. **Data Transformation: Normalization/Scaling (Optional Consideration)**

The goal is to consider scaling numerical features if necessary for specific advanced visualizations or modeling techniques that are not explicitly detailed as static charts in the proposal but could be useful for deeper correlation analysis.  
Sklearn.preprocessing was used to demonstrate code implementation with Min-Max scaling.MinMaxScaler applied to a copy of the data.

A computer screen shot of a computer

AI-generated content may be incorrect.

Figure 7: Output result for step 7

**Action and justification:** The primary visualizations suggested in proposal (bar charts comparing rates, scatter plots showing income relationships, and heatmaps) will be easier for the target HR audience to comprehend if the original scale and units (MonthlyIncome in dollars, for example) are used, as the saved dataset was not scaled.   
Since the cleaned data currently prioritizes direct interpretability for the intended visualizations, we must conclude that scaling is still an option for more complex analyses in the future if necessary.

1. **Saving the Cleaned Data**

Purpose of this stage is to store the processed, analysis-ready dataset.  
With code Implementation at the final step, the final cleaned DataFrame was saved using df.to\_csv('cleaned\_WA\_Fn-UseC\_-HR-Employee-Attrition.csv', index=False).  
A screenshot of a computer

AI-generated content may be incorrect.

Figure 8: Output result for step 8

Findings & justification : The cleaned dataset was successfully saved and providing the essential output and the basis for creating the visualizations and interactive dashboard proposed and required for subsequent hurdles.

A black screen with white text

AI-generated content may be incorrect.

Figure 9: Output result for the re-check step

To ensure that the process meets the task requirements, I have included a final step in the code to check that everything has been completed.

1. **Conclusion**

In conclusion, we've thoroughly cleaned up the IBM HR dataset now. We went through and tackled the main data quality problems – filling in missing info, ditching duplicate records and columns we didn't need, flagging any outliers, and getting all the categorical data properly coded into numbers.  
The result demonstrates that the dataset is now well-structured, consistent, and ready for the next higher stage. It's configured exactly for the data visualization tasks outlined in the proposal, so we can get started on creating static charts (bars, scatters, heatmaps) and an interactive dashboard to delve into and demonstrate what's driving employee attrition at IBM.

1. **Reference:**

PAVANSUBHASH. (2017). IBM HR Analytics Employee Attrition & Performance. Www.kaggle.com. https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset