**D599 - Part 1 - Data Profiling  
Andrew Marchese**

**Contents**

[A. Profile Data 1](#_Toc175499903)

[B. Indicate the data type and data subtype for each variable. 1](#_Toc175499904)

[C. Provide a sample of observable values for each variable. 3](#_Toc175499905)

[B. Data Cleaning and Plan 4](#_Toc175499906)

[B1. Dataset Quality Issues 4](#_Toc175499907)

[B2. List of Quality Issues Found 4](#_Toc175499908)

[C. Discuss Cleaning 5](#_Toc175499909)

[C1. Dataset Modifications 5](#_Toc175499910)

[C2. Data Cleansing Techniques 5](#_Toc175499911)

[C3. Technique Advantages 5](#_Toc175499912)

[C4. Technique Limitations 6](#_Toc175499913)

## **A. Profile Data**

a. Describe the general characteristics of the initial dataset (e.g., rows, columns).  
The data dictionary for this dataset describes a table containing over 30 different columns, each representing another characteristic about the employee. Each employee is recorded on a different row, capturing many aspects relating to that employee, such as education level, pay rate, department, job satisfaction, and tenure with the company. This dataset contains almost all demographic information, job-related factors, and work-life balance metrics that may influence employee turnover. This structured data is of immense help while analyzing the trends and patterns of retention and turnover of employees in the firm. Informed decision-making with regard to human resource strategies would therefore be possible.

## **B. Indicate the data type and data subtype for each variable.**

**Age**: Integer (Numeric)

**Turnover**: Categorical (String - Binary)

**BusinessTravel**: Categorical (String - Binary)

**DailyRate**: Integer (Numeric)

**Department**: Categorical (String - Nominal)

**DistanceFromHome**: Integer (Numeric)

**Education**: Ordinal (Numeric - Categorical)

**EducationField**: Categorical (String - Nominal)

**EmployeeCount**: Constant (Integer)

**EmployeeNumber**: Identifier (Integer)

**EnvironmentSatisfaction**: Ordinal (Numeric - Categorical)

**Gender**: Categorical (String - Binary)

**HourlyRate**: Integer (Numeric)

**JobInvolvement**: Ordinal (Numeric - Categorical)

**JobLevel**: Ordinal (Numeric - Categorical)

**JobRole**: Categorical (String - Nominal)

**JobSatisfaction**: Ordinal (Numeric - Categorical)

**MaritalStatus**: Categorical (String - Nominal)

**MonthlyIncome**: Integer (Numeric)

**MonthlyRate**: Integer (Numeric)

**NumCompaniesWorked**: Integer (Numeric)

**Over18**: Categorical (String - Binary)

**OverTime**: Categorical (String - Binary)

**PercentSalaryHike**: Integer (Numeric)

**PerformanceRating**: Ordinal (Numeric - Categorical)

**RelationshipSatisfaction**: Ordinal (Numeric - Categorical)

**StandardHours**: Integer (Numeric - Constant)

**StockOptionalLevel**: Ordinal (Numeric - Categorical)

**TotalWorkingYears**: Integer (Numeric)

**TrainingTimesLastYear**: Integer (Numeric)

**WorkLifeBalance**: Ordinal (Numeric - Categorical)

**YearsAtCompany**: Integer (Numeric)

**YearsInCurrentRole**: Integer (Numeric)

**YearsSinceLastPromotion**: Integer (Numeric)

**YearsWithCurrManager**: Integer (Numeric)

## **C. Provide a sample of observable values for each variable.**

**Age**: 33, 35, 27, 44, 56

**Turnover**: Yes, Yes, Yes, No, No

**BusinessTravel**: Non-Travel, Non-Travel, Travel\_Frequently, Travel\_Rarely, Travel\_Rarely

**DailyRate**: 241, 679, 359, 1133, 118

**Department**: Hardware, Support, Hardware, Software, Software

**DistanceFromHome**: 16, 7, 50, 12, 43

**Education**: 3, 2, 1, 5, 2

**EducationField**: Technical Degree, Life Sciences, Life Sciences, Life Sciences, Human Resources

**EmployeeCount**: 1 (constant across all samples)

**EmployeeNumber**: 3505, 1129, 6305, 4595, 7203

**EnvironmentSatisfaction**: 1, 3, 4, 2, 2

**Gender**: Female, Male, Female, Female, Female

**HourlyRate**: 67, 122, 199, 150, 115

**JobInvolvement**: 3, 2, 1, 2, 3

**JobLevel**: 3, 5, 1, 4, 1

**JobRole**: Manufacturing Director, Research Director, Sales Representative, Research Director, Developer

**JobSatisfaction**: 1, 2, 3, 3, 2

**MaritalStatus**: Married, Single, Single, Divorced, Married

**MonthlyIncome**: 36809, 1690, 50883, 11166, 42537

**MonthlyRate**: 294472, 32110, 865011, 89328, 1063425

**NumCompaniesWorked**: 1, 6, 8, 5, 6

**Over18**: Y, Y, Y, Y, Y

**OverTime**: Yes, Yes, Yes, No, Yes

**PercentSalaryHike**: 18, 5, 7, 34, 49

**PerformanceRating**: 1, 4, 2, 3, 2

**RelationshipSatisfaction**: 1, 1, 4, 4, 2

**StandardHours**: 80 (constant across all samples)

**StockOptionLevel**: 4, 1, 4, 3, 1

**TotalWorkingYears**: 35, 5, 10, 19, 3

**TrainingTimesLastYear**: 4, 1, 4, 4, 3

**WorkLifeBalance**: 4, 1, 2, 1, 4

**YearsAtCompany**: 13, 4, 4, 14, 1

**YearsInCurrentRole**: 2, 3, 2, 7, 1

**YearsSinceLastPromotion**: 8, 3, 4, 12, 1

**YearsWithCurrManager**: 11, 4, 2, 2, 1

## **B. Data Cleaning and Plan**

## **B1. Dataset Quality Issues**

I performed two main steps to detect inconsistencies in the dataset. First, I converted all missing values and NoneType entries to NaN throughout the entire DataFrame, then I applied the drop\_duplicates function to drop all duplicate rows. Second, I profiled each column individually using a custom Python script that provided me with a display of useful details like the data types, number of unique values, count of NaN entries, and count of blank entries. The script also used the .describe() function from pandas to locate any outlying positive or negative values. Using that information, I then created data cleaning Python code as needed for each column in the Jupyter notebook cell in a manner that aligned with the columns real world business use and data type.

## **B2. List of Quality Issues Found**

**DataFrame Duplicate Entries:** Found 298 duplicate rows

**DataFrame Blank Values:** Found six blank cells [‘’]

**Age:** Has incorrect age value of 148

**BusinessTravel:** Has these three uncategorical values [-1, 1, '00',]

**DistanceFromHome:** Has a 3737 mile commute value

**EmployeeCount:** Has incorrect values [-1, 3]

**MonthlyIncome:** Has a negative value of -38005

**MonthlyRate:** Has an outlying scientific notation value of [8.722149e+11]

**TotalWorkingYears:** Has a negative value of [-1.0,]

**TrainingTimesLastYear:** Has 418 NaN values but no zero values

**YearsWithCurrManager:** Has negative value of -1000 and a unmatching value of ['na']

## **C. Discuss Cleaning**

## **C1. Dataset Modifications**

**Duplicate Entries:** Removed duplicate rows to ensure each employee record is unique.

**Missing Values:** Converted NoneType and blank values to NaN for consistent handling of missing data.

**Age:** Set ages outside the range of 16-90 to NaN to maintain realistic age values.

**BusinessTravel:** Standardized the "BusinessTravel" column by setting non-matching categories to NaN.

**DistanceFromHome:** Changed distances greater than 90 miles to NaN to reflect plausible commuting distances.

**EmployeeCount:** Ensured "EmployeeCount" was consistently set to 1 across all entries.

**MonthlyIncome:** Converted negative income values to their absolute values to ensure only positive incomes.

**MonthlyRate:** Set abnormally high values in "MonthlyRate" to NaN to prevent distortion of analysis.

**TotalWorkingYears:** Converted negative values in "TotalWorkingYears" to NaN for accurate work experience data.

**TrainingTimesLastYear:** Replaced NaN values with 0 in "TrainingTimesLastYear" to indicate no training.

**YearsWithCurrManager:** Standardized "YearsWithCurrManager" by converting non-integer and out-of-range values to NaN.

## **C2. Data Cleansing Techniques**

I chose specific data cleansing methodologies to ensure the dataset was accurate, consistent, and ready for meaningful analysis. For example, I converted erroneous values like negative figures or improbable entries—a 148-year-old age or a 3,737-mile commute—to NaN to minimize skewed results. Removal of duplicate entries was done to avoid over-representation, and missing or NoneType values were transformed to NaN for consistency across the dataset. I also applied logical thresholds and used domain knowledge—for example, age ranges and feasible lengths of commutes—to identify and correct outliers.

## **C3. Technique Advantages**

The greatest advantage of this data cleaning procedure is that it can preserve the consistency and accuracy of the dataset, hence making it reliable for analysis. It is easier to handle cases of missing or inaccurate data represented as NaN during analysis without bias. Removal of duplicates prevents repetition of the same employee, hence avoiding distortion in the analysis, which could have been caused by many entries of the same employee.

## **C4. Technique Limitations**

Another limitation is that replacement of invalid or inconstant values by NaN may incur huge data loss, in case of the effecting of several records, thus loss of analytical statistical power. Another limitation is that these thresholds set for outlier detection—for instance, setting a maximum commute distance of 90 miles—might inadvertently remove certain valid data points that are very rare and thus probably miss the key trends.