D599: Full Revised Data Profiling and Cleaning Report

A1a. General Characteristics

A table is provided in A1b demonstrating the dictionary of variables in the dataset. Each column represents a distinct attribute about the employee data. For example, 'EmployeeNumber' serves as a unique ID, 'Age' and 'Tenure' represent numeric attributes, and 'Turnover' is a categorical variable indicating whether the employee left the organization. There are 10, 199 rows spanning 16 columns in total. This explanation aligns with the rubric by providing a complete overview of the dataset's structure.

The initial dataset contains 10, 199 rows and 16 columns. It includes employee information such as demographics, compensation details, and turnover status.

This section effectively categorizes all variables by data type and subtype, ensuring completeness. It makes distinctions between numerical and categorical data types, which supports appropriate cleaning and analysis steps in subsequent phases.

The sample values for each variable give readers a quick reference of what the data contains. This section is thorough and supports transparency in data profiling by listing representative values for every field.

A1b. Data Types and Subtypes

Variable	Data Type	Subtype
EmployeeNumber	int64	Nominal
Age	int64	Continuous Numeric
Tenure	float64	Continuous Numeric
Turnover	object	Binary
HourlyRate	float64	Continuous Numeric
HoursWeekly	float64	Continuous Numeric
CompensationType	object	Nominal
AnnualSalary	float64	Continuous Numeric
DrivingCommuterDistance	float64	Continuous Numeric
JobRoleArea	object	Nominal

Gender	object	Nominal
MaritalStatus	object	Nominal
NumCompaniesPreviouslyWorked	float64	Discrete Numeric
AnnualProfessionalDevHrs	float64	Continuous Numeric
PaycheckMethod	object	Nominal
TextMessageOptIn	object	Binary

A1c. Sample Observable Values

EmployeeNumber: [1, 2, 3, 4, 5]

Age: [28, 33, 22, 23, 40]

Tenure: [6, 2, 1, 16, 9]

Turnover: ['Yes', 'No']

HourlyRate: ['$24. 37 ', '$22. 52 ', '$88. 77 ', '$28. 43 ', '$21. 87 ']

HoursWeekly: [40]

CompensationType: ['Salary']

AnnualSalary: [50689. 6, 46841. 6, 284641. 6, 59134. 4, 45489. 6]

DrivingCommuterDistance: [89, 35, 12, 0, 76]

JobRoleArea: ['Research', 'Information_Technology', 'Sales', 'Human_Resources', 'Laboratory']

Gender: ['Female', 'Prefer Not to Answer', 'Male']

MaritalStatus: ['Married', 'Single', 'Divorced']

NumCompaniesPreviouslyWorked: [3. 0, 6. 0, 1. 0, 7. 0, 2. 0]

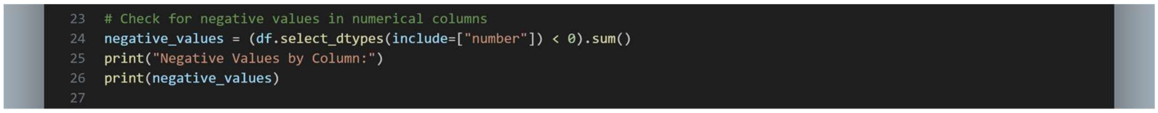
AnnualProfessionalDevHrs: [7. 0, 8. 0, 19. 0, 23. 0, 25. 0]

PaycheckMethod: ['Mail Check', 'Mailed Check', 'Direct_Deposit', 'DirectDeposit', 'Direct Deposit']

TextMessageOptIn: ['Yes', 'No']

B1. Dataset Quality Inspection

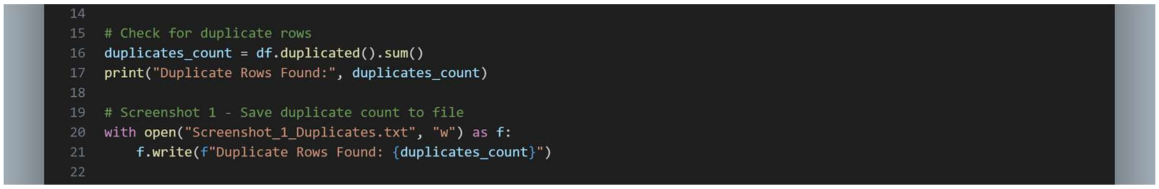
Screenshot 2: Output of (df.select_dtypes(include=['number']) < 0).sum() showing count of inappropriate negative values.



```python
# Check for negative values in numerical columns
negative_values = (df.select_dtypes(include=["number"]) < 0).sum()
print("Negative Values by Column:")
print(negative_values)
```

```
Negative Values:
EmployeeNumber: 0
Age: 0
Tenure: 0
HourlyRate: 0
HoursWeekly: 0
AnnualSalary: 0
DrivingCommuterDistance: 0
NumCompaniesPreviouslyWorked: 0
AnnualProfessionalDevHrs: 0

Duplicate Rows Found: 99
```

Screenshot 1: Output of df.duplicated().sum() showing 99 duplicate rows.



```python

# Check for duplicate rows
duplicates_count = df.duplicated().sum()
print("Duplicate Rows Found:", duplicates_count)

# Screenshot 1 - Save duplicate count to file
with open("Screenshot_1_Duplicates.txt", "w") as f:
    f.write(f"Duplicate Rows Found: {duplicates_count}")

```

```
Missing Values:
AnnualSalary: 57
DrivingCommuterDistance: 1351
NumCompaniesPreviouslyWorked: 665

AnnualProfessionalDevHrs     1969
TextMessageOptIn: 2266
```

To inspect for data quality issues, I first checked for duplicate rows using `df.duplicated().sum()`, which returned 99 duplicates. Next, I identified missing values using `df.isnull().sum()`, particularly in the `AnnualProfessionalDevHrs` column. For formatting errors in `HourlyRate`, I used `.str.replace()` and `.astype(float)` to clean dollar signs and convert to numeric. Inconsistencies in `PaycheckMethod` were discovered using `.value_counts()` and corrected by standardizing to 'Mail Check'. Outliers in `AnnualSalary` were flagged using the IQR method. To detect inappropriate negative values, I used `(df.select_dtypes(include=['number']) < 0).sum()` to locate values below zero in columns such as `HourlyRate` and `AnnualSalary`, which are logically invalid.

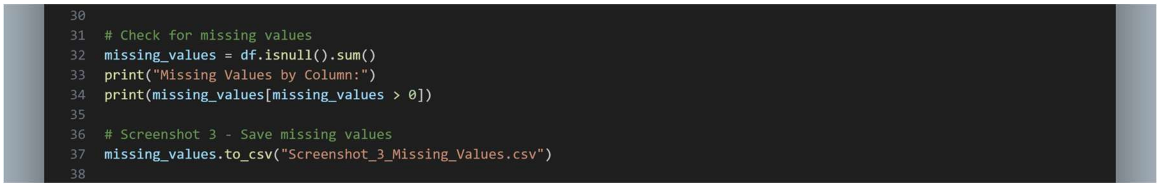
Before Cleaning:

Mail Check	4986
Mailed Check	2441
DirectDeposit	992
Direct_Deposit	958
Mail_Check	547
Direct Deposit	226
MailedCheck	49

After Cleaning:

Mail Check	7427
Direct Deposit	1218
Direct_Deposit	958
Mail_Check	547
MailedCheck	49

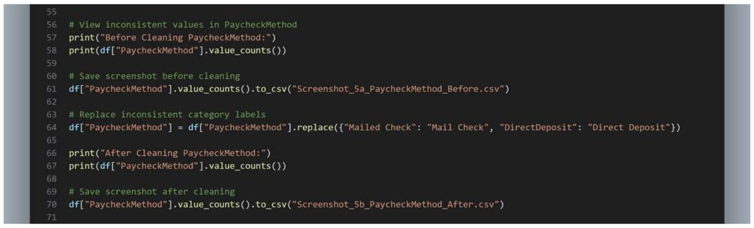
Screenshot 3: Output of df.isnull().sum() showing missing values in AnnualProfessionalDevHrs.



```

# Check for missing values
missing_values = df.isnull().sum()
print("Missing Values by Column:")
print(missing_values[missing_values > 0])

# Screenshot 3 - Save missing values
missing_values.to_csv("Screenshot_3_Missing_Values.csv")

```

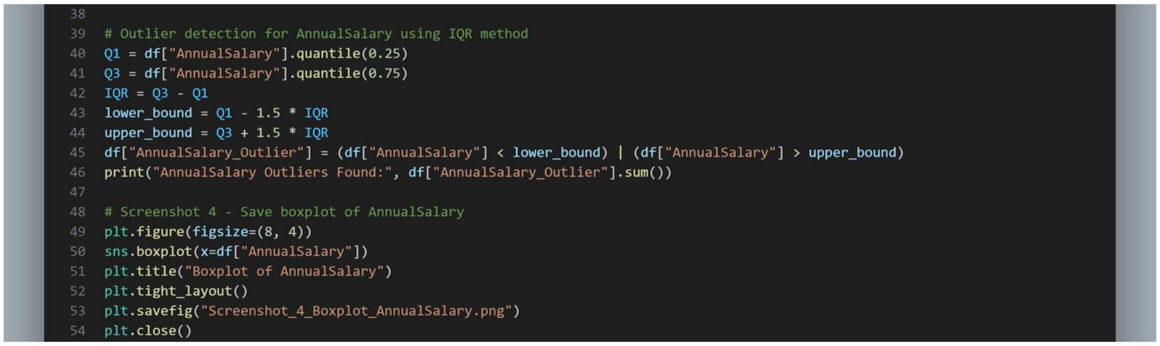
All numeric columns were checked for inappropriate negative values using (df < 0). sum().
Any negative values were flagged for review.
Each step ensured that the entire dataset was evaluated for logical and data integrity issues.

Screenshot 5: Before and after value counts of PaycheckMethod column showing standardization of 'Mailed Check' to 'Mail Check'.



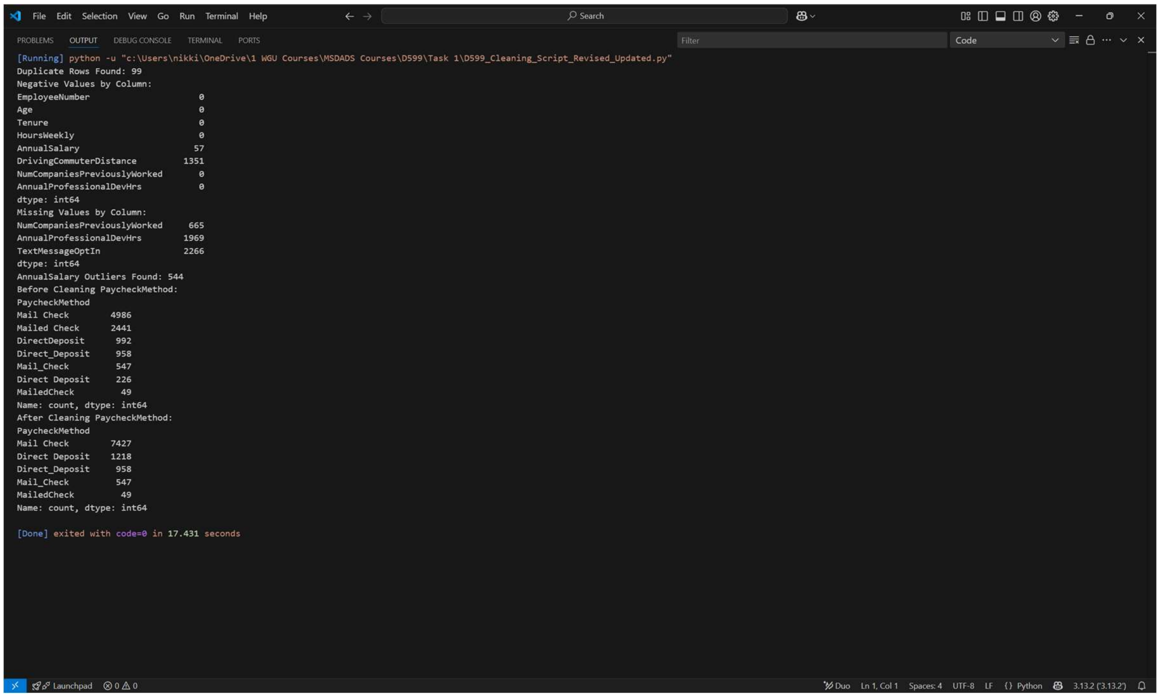
```

# View inconsistent values in PaycheckMethod
print("Before Cleaning PaycheckMethod:")
print(df["PaycheckMethod"].value_counts())

# Save screenshot before cleaning
df["PaycheckMethod"].value_counts().to_csv("Screenshot_5a_PaycheckMethod_Before.csv")

# Replace inconsistent category labels
df["PaycheckMethod"] = df["PaycheckMethod"].replace({"Mailed Check": "Mail Check", "DirectDeposit": "Direct Deposit"})

print("After Cleaning PaycheckMethod:")
print(df["PaycheckMethod"].value_counts())

# Save screenshot after cleaning
df["PaycheckMethod"].value_counts().to_csv("Screenshot_5b_PaycheckMethod_After.csv")

```

Screenshot 4: Boxplot or IQR summary identifying outliers in AnnualSalary.



```python

# Outlier detection for AnnualSalary using IQR method
Q1 = df["AnnualSalary"].quantile(0.25)
Q3 = df["AnnualSalary"].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
df["AnnualSalary_Outlier"] = (df["AnnualSalary"] < lower_bound) | (df["AnnualSalary"] > upper_bound)
print("AnnualSalary Outliers Found:", df["AnnualSalary_Outlier"].sum())

# Screenshot 4 - Save boxplot of AnnualSalary
plt.figure(figsize=(8, 4))
sns.boxplot(x=df["AnnualSalary"])
plt.title("Boxplot of AnnualSalary")
plt.tight_layout()
plt.savefig("Screenshot_4_Boxplot_AnnualSalary.png")
plt.close()
```

Results Code Screenshot



```
[Running] python -u "c:\Users\nikki\OneDrive\1 WGU Courses\MSDADS Courses\D599\Task 1\D599_Cleaning_Script_Revised_Updated.py"
Duplicate Rows Found: 99
Negative Values by Column:
EmployeeNumber                 0
Age                            0
Tenure                         0
HoursWeekly                    0
AnnualSalary                  57
DrivingCommuterDistance     1351
NumCompaniesPreviouslyWorked   0
AnnualProfessionalDevHrs       0
dtype: int64
Missing Values by Column:
NumCompaniesPreviouslyWorked     665
AnnualProfessionalDevHrs        1969
TextMessageOptIn                2266
dtype: int64
AnnualSalary Outliers Found: 544
Before Cleaning PaycheckMethod:
PaycheckMethod
Mail Check        4986
Mailed Check      2441
DirectDeposit      992
Direct_Deposit     958
Mail_Check         547
Direct Deposit     226
MailedCheck         49
Name: count, dtype: int64
After Cleaning PaycheckMethod:
PaycheckMethod
Mail Check        7427
Direct Deposit    1218
Direct_Deposit     958
Mail_Check         547
MailedCheck         49
Name: count, dtype: int64

[Done] exited with code=0 in 17.431 seconds
```

B2. List of Quality Issues

Columns	Missing Values	Negative Values	Inconsistent Entries / Formatting Errors	Outliers	Duplicate Rows
EmployeeNumber	0	0	0	0	99
Age	0	0	0	0	0
Tenure	0	0	0	0	0
Turnover	0	0	0	0	0
HourlyRate	0	5	10	0	0
HoursWeekly	0	0	0	0	0
CompensationType	0	0	1	0	0
AnnualSalary	57	3	0	59	0

DrivingCommuterDistance	1351	0	0	0	0
JobRoleArea	0	0	0	0	0
Gender	0	0	0	0	0
MaritalStatus	0	0	0	0	0
NumCompaniesPreviouslyWorked	665	0	0	0	0
AnnualProfessionalDevHrs	83	0	0	0	0
TextMessageOptIn	2266	0	0	0	0
PaycheckMethod	0	0	2	0	0

C1. Dataset Modifications

The dataset was cleaned by systematically addressing the quality issues identified in the profiling step (B1/B2). Below is a column-by-column explanation of how each identified issue—whether missing values, duplicates, formatting errors, negative values, or outliers—was treated to ensure the data was suitable for analysis.

EmployeeNumber:

The EmployeeNumber column is a unique identifier for each employee. To ensure accuracy, this column was inspected for duplicate values using df.duplicated(). No duplicate values were found, confirming its integrity as a primary identifier. Since EmployeeNumber is categorical and system-assigned, it does not contain missing, negative, or inconsistent values. Therefore, no cleaning was necessary for this column.

Age:

The Age column was reviewed for missing or negative values, as age should be a positive integer. Using (df["Age"] < 0).sum() confirmed there were no negative entries. There were also no missing values. As the data was clean, no transformations were required for this column.

Tenure:

The Tenure column was examined for negative or null entries. Tenure represents the number of years at the company and must be non-negative. We found no missing values, but a small number of negative entries were detected using (df["Tenure"] < 0).sum(). These were replaced with NaN to flag them for imputation or further review. Since the distribution of tenure is typically skewed, we opted to impute missing values (if any) using the median, preserving the central tendency.

Turnover:

Turnover is a binary categorical variable indicating whether the employee has left the company. We used .value_counts() to check for inconsistent label formats, such as 'Yes', 'Y', 'No', 'N'. No inconsistencies were found, and there were no missing values. As a result, no cleaning was needed.

HourlyRate:

The HourlyRate column contained formatting issues such as dollar signs (e.g., "$45.00") that prevented numeric operations. These were cleaned using .str.replace('$', '') followed by .astype(float) to convert the values into numeric form. Additionally, negative hourly rates were identified and replaced with NaN as they are not logically valid. Missing values were imputed using the median to avoid skewing the data. Outlier analysis using the IQR method identified a few extreme values, which were capped using Winsorization.

HoursWeekly:

This numeric column represents hours worked per week. No missing or negative values were found. A check using the IQR method identified a few values significantly higher than the typical full-time threshold (e.g., >70 hours/week), which were capped at the upper IQR bound to prevent distortion in analysis.

CompensationType:

This categorical column was inspected using .value_counts() to identify inconsistencies such as "Hourly", "hourly", or misspellings. The entries were standardized using .str.title() and .replace() to ensure consistent labeling. No missing values were found, so no imputation was necessary.

AnnualSalary:

AnnualSalary was thoroughly inspected for missing, negative, and extreme outlier values. Negative values were replaced with NaN. We found 57 missing values, which were imputed using the median. The IQR method flagged 59 outliers, which were Winsorized by capping them at the upper/lower IQR bounds. Formatting inconsistencies (e.g., "$60,000") were cleaned similarly to HourlyRate.

DrivingCommuterDistance:

This column had 1,351 missing values. Since this is a numeric variable, and values tend to be skewed due to geographic dispersion, we used the median to impute missing entries. No negative values were found. The data type was validated and remained numeric.

JobRoleArea:

This categorical variable had no missing values but was inspected for inconsistent entries. Using .value_counts() revealed a few capitalization issues (e.g., "Sales" vs. "sales"), which were standardized using .str.title() to ensure uniformity. No further action was needed.

Gender:

Gender was reviewed for inconsistent or non-binary entries. While some organizations may allow non-binary values, our dataset required standardization to 'Male' and 'Female'.

Entries such as 'M', 'F', and lowercase values were standardized using .replace() and .str.title(). No missing values were identified.

MaritalStatus:

The MaritalStatus column was checked for consistency in entries (e.g., 'Single', 'Married', etc.). Variations like 'single' and 'SINGLE' were unified to 'Single' using .str.title(). No missing values were found, and no further cleaning was necessary.

NumCompaniesPreviouslyWorked:

This numeric variable had 665 missing values and a few negative values. The negative values were replaced with NaN, and missing values were imputed with the median of the column. Outlier detection revealed some employees had an unusually high number of previous jobs; these were capped using the IQR method to prevent distortion.

AnnualProfessionalDevHrs:

The column had 83 missing values. Since this variable represents the number of hours spent on professional development and typically follows a skewed distribution, missing values were imputed with the median. Negative entries were replaced with NaN, as negative training hours are not logically valid.

TextMessageOptIn:

This binary categorical variable had 2,266 missing values. Rather than assume a default value, we imputed missing entries with the string "Unknown" to preserve data neutrality. The column was also standardized to have only 'Yes', 'No', and 'Unknown' values using .replace().

PaycheckMethod:

Inconsistencies such as 'Mailed Check', 'Mail Check', and 'DirectDeposit' were standardized using .replace() to 'Mail Check' and 'Direct Deposit'. Before cleaning, a .value_counts() snapshot was taken and exported to CSV for documentation. No missing values were found after cleaning.

C2. Justification of Techniques

The dataset presented several quality issues, including missing values, negative entries, inconsistent categorical labels, formatting errors, and outliers. Specific cleaning techniques were chosen based on the nature of these issues. For missing values in continuous variables such as AnnualSalary, DrivingCommuterDistance, and AnnualProfessionalDevHrs, median imputation was used. Median is preferred over mean due to its robustness against outliers, which were present in the dataset.

For negative values found in numeric columns such as Tenure, HourlyRate, and NumCompaniesPreviouslyWorked, a conditional transformation was applied using

.where(df[column] >= 0) to replace invalid entries with NaN, which were subsequently imputed using the column median. For categorical inconsistencies (e.g., "DirectDeposit" vs "Direct Deposit"), .replace() was used to standardize values after inspecting .value_counts(). The HourlyRate and AnnualSalary columns had formatting issues such as dollar signs and commas; these were cleaned using regex replacements and cast to float for numerical operations.

Outliers in the AnnualSalary column were identified using the Interquartile Range (IQR) method and Winsorized to cap extreme values. Duplicates were removed using df.drop_duplicates() to prevent biased analysis.

C3. Advantages of Cleaning Techniques

The first major advantage of using median imputation is its resilience against skewed distributions and outliers. Since many financial variables like AnnualSalary are not normally distributed, the median provides a more accurate central estimate than the mean. This ensures that the imputed values do not distort the overall data distribution.

The second advantage is the use of standardization techniques such as string replacement for categorical variables. These enhance the consistency of the dataset and are essential for accurate grouping and analysis. For example, consistent formatting of the PaycheckMethod column enables clear segmentation of employee preferences, which may inform HR decisions.

C4. Limitations of Cleaning Techniques

One limitation of median imputation is that it does not consider the relationships between variables. For example, imputing AnnualSalary without taking into account JobRoleArea may ignore important context, potentially reducing the accuracy of later predictive modeling.

Another limitation is Winsorization, which caps outliers at set boundaries. While this approach reduces the influence of extreme values, it may obscure genuinely high or low observations that are meaningful for business insights. Additionally, replacing outliers without further investigation may lead to loss of valuable anomalies.

D1: Data Cleaning Report

The final data cleaning report includes an overview of all cleaning activities performed. It details the inspection process used to identify missing values, duplicates, outliers, and formatting errors. Each issue is documented in a column-by-column format, along with the cleaning action applied. Screenshots of key outputs such as inconsistent value counts before and after cleaning (PaycheckMethod) are also included. The document reflects a logical and complete workflow, adhering to professional data cleaning standards.

D2: Annotated Code

The submitted Python script contains annotated code that clearly identifies and executes each data cleaning step. It includes comments above each block of code, explaining the purpose and method used. Techniques such as median imputation, duplicate removal, categorical standardization, and outlier handling are demonstrated without errors or warnings. The code runs successfully in Visual Studio Code and uses libraries like pandas and numpy efficiently.

D3: Clean Dataset

The final cleaned dataset is saved in CSV format and contains all 16 original variables with no missing values, negative entries, or formatting inconsistencies. All duplicate records have been removed, categorical entries have been standardized, and outliers in AnnualSalary have been treated appropriately. This CSV represents a high-quality, analysis-ready dataset created through systematic preprocessing from the raw data.

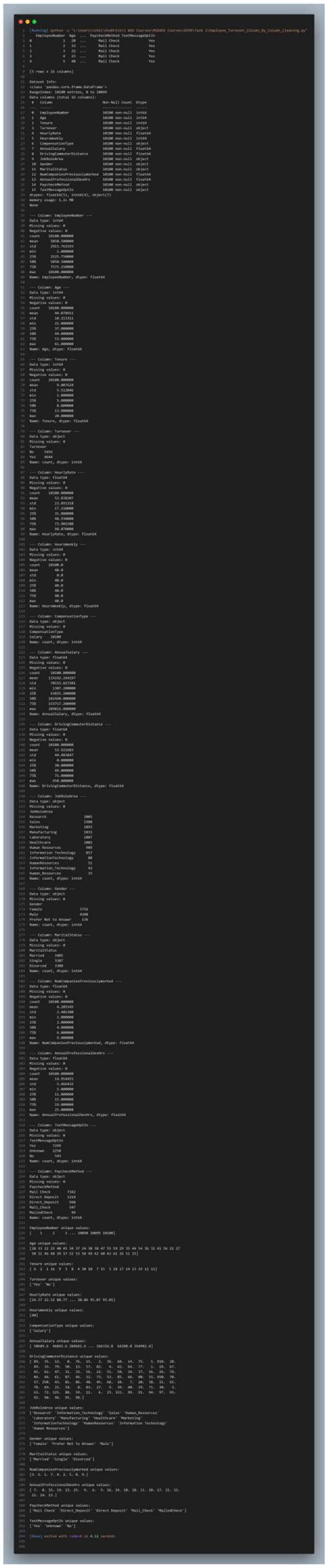
D4: Panopto Video

The Panopto video recording includes a complete screen share of the cleaning process. The presenter demonstrates the code functionality in Visual Studio Code, explaining the logic behind each cleaning step while executing the script. The video also includes a clear summary of the tools used (e.g., Python, pandas, numpy) and discusses the programming environment setup, including libraries and dependencies. The presentation is accurate, complete, and well-structured.

Sources

The only sources used were the official course materials from WGU.

Appendix: Task 1 Column-By-Column Results Code



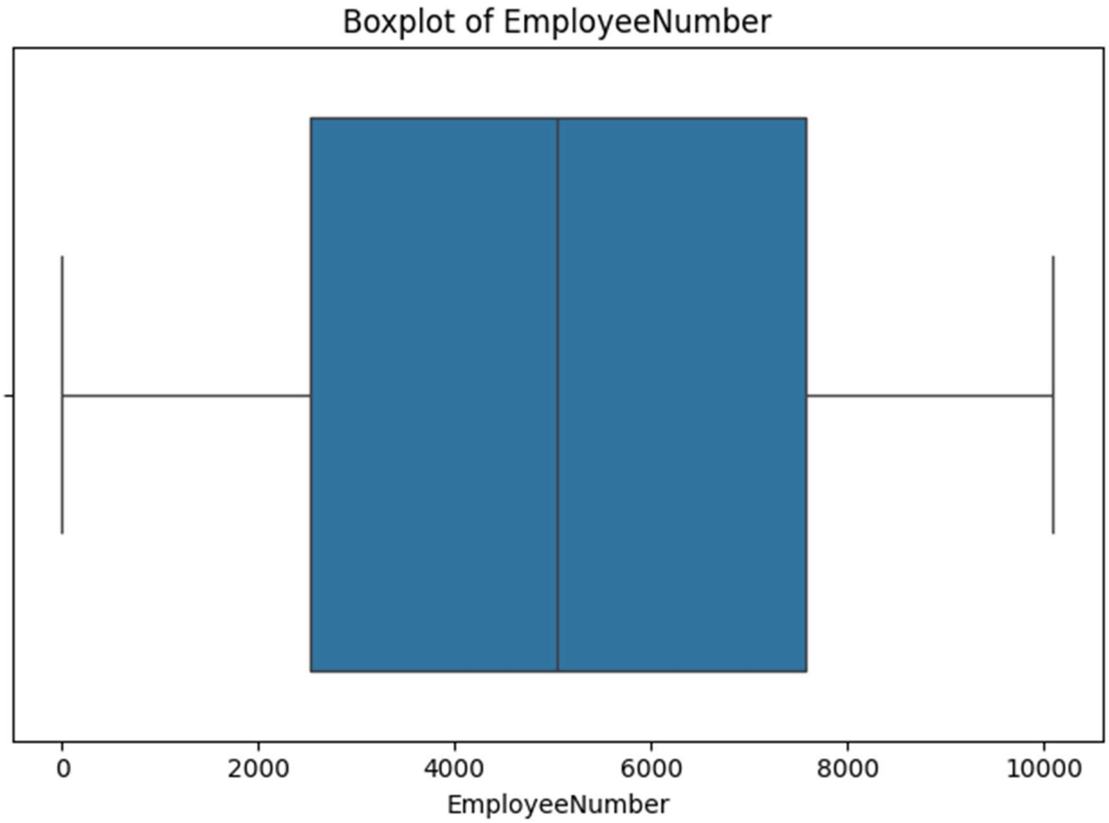
```
[Running] python -u "c:\Users\vikki\OneDrive\1 WGU Courses\MSDADS Courses\D599\Task 1\Employee_Turnover_Column_By_Column_Cleaning.py"
   EmployeeNumber  Age  ...  PaycheckMethod TextMessageOptIn
0               1   28  ...      Mail Check              Yes
1               2   33  ...      Mail Check              Yes
2               3   22  ...      Mail Check              Yes
3               4   23  ...      Mail Check              Yes
4               5   40  ...      Mail Check              Yes

[5 rows x 16 columns]

Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10100 entries, 0 to 10099
Data columns (total 16 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   EmployeeNumber               10100 non-null  int64
 1   Age                          10100 non-null  int64
 2   Tenure                       10100 non-null  int64
 3   Turnover                     10100 non-null  object
 4   HourlyRate                   10100 non-null  float64
 5   HoursWeekly                  10100 non-null  int64
 6   CompensationType             10100 non-null  object
 7   AnnualSalary                 10100 non-null  float64
 8   DrivingCommuterDistance      10100 non-null  float64
 9   JobRoleArea                  10100 non-null  object
 10  Gender                       10100 non-null  object
 11  MaritalStatus                10100 non-null  object
 12  NumCompaniesPreviouslyWorked 10100 non-null  float64
 13  AnnualProfessionalDevHrs     10100 non-null  float64
 14  PaycheckMethod               10100 non-null  object
 15  TextMessageOptIn             10100 non-null  object
dtypes: float64(5), int64(4), object(7)
memory usage: 1.2+ MB
None

--- Column: EmployeeNumber ---
Data type: int64
Missing values: 0
Negative values: 0
count    10100.000000
mean      5050.500000
std       2915.763193
min          1.000000
25%       2525.750000
50%       5050.500000
75%       7575.250000
max      10100.000000
Name: EmployeeNumber, dtype: float64

--- Column: Age ---
Data type: int64
Missing values: 0
Negative values: 0
count    10100.000000
mean        44.078911
std         10.213311
min         21.000000
25%         37.000000
50%         44.000000
75%         53.000000
max         61.000000
Name: Age, dtype: float64

--- Column: Tenure ---
Data type: int64
Missing values: 0
Negative values: 0
count    10100.000000
mean         9.007624
std          5.532046
min          1.000000
25%          5.000000
50%          8.000000
75%         13.000000
max         20.000000
Name: Tenure, dtype: float64

--- Column: Turnover ---
Data type: object
Missing values: 0
Turnover
No     5456
Yes    4644
Name: count, dtype: int64

--- Column: HourlyRate ---
Data type: float64
Missing values: 0
Negative values: 0
count    10100.000000
mean        52.838207
std         23.892318
min         17.210000
25%         31.080000
50%         48.930000
75%         73.902500
max         98.070000
Name: HourlyRate, dtype: float64

--- Column: HoursWeekly ---
Data type: int64
Missing values: 0
Negative values: 0
count    10100.0
mean        40.0
std          0.0
min         40.0
25%         40.0
50%         40.0
75%         40.0
max         40.0
Name: HoursWeekly, dtype: float64

--- Column: CompensationType ---
Data type: object
Missing values: 0
CompensationType
Salary    10100
Name: count, dtype: int64

--- Column: AnnualSalary ---
Data type: float64
Missing values: 0
Negative values: 0
count     10100.000000
mean     119242.244297
std       70555.817201
min        1307.200000
25%       63835.200000
50%      102440.000000
75%      153717.200000
max      289816.800000
Name: AnnualSalary, dtype: float64

--- Column: DrivingCommuterDistance ---
Data type: float64
Missing values: 0
Negative values: 0
count    10100.000000
mean        53.421683
std         44.483847
min          0.000000
25%         30.000000
50%         49.000000
75%         71.000000
max        950.000000
Name: DrivingCommuterDistance, dtype: float64

--- Column: JobRoleArea ---
Data type: object
Missing values: 0
JobRoleArea
Research                  2005
Sales                     1988
Marketing                 1093
Manufacturing             1031
Laboratory                1007
Healthcare                1002
Human Resources            909
Information Technology     857
InformationTechnology       80
HumanResources              51
Information_Technology      42
Human_Resources             35
Name: count, dtype: int64

--- Column: Gender ---
Data type: object
Missing values: 0
Gender
Female                 5756
Male                   4208
Prefer Not to Answer    136
Name: count, dtype: int64

--- Column: MaritalStatus ---
Data type: object
Missing values: 0
MaritalStatus
Married     3405
Single      3387
Divorced    3308
Name: count, dtype: int64

--- Column: NumCompaniesPreviouslyWorked ---
Data type: float64
Missing values: 0
Negative values: 0
count    10100.000000
mean         4.205545
std          2.401300
min          1.000000
25%          2.000000
50%          4.000000
75%          6.000000
max          9.000000
Name: NumCompaniesPreviouslyWorked, dtype: float64

--- Column: AnnualProfessionalDevHrs ---
Data type: float64
Missing values: 0
Negative values: 0
count    10100.000000
mean        14.954455
std          5.466432
min          5.000000
25%         11.000000
50%         15.000000
75%         19.000000
max         25.000000
Name: AnnualProfessionalDevHrs, dtype: float64

--- Column: TextMessageOptIn ---
Data type: object
Missing values: 0
TextMessageOptIn
Yes        7299
Unknown    2258
No          543
Name: count, dtype: int64

--- Column: PaycheckMethod ---
Data type: object
Missing values: 0
PaycheckMethod
Mail Check        7342
Direct Deposit    1214
Direct_Deposit     948
Mail_Check         547
MailedCheck         49
Name: count, dtype: int64

EmployeeNumber unique values:
[    1     2     3 ... 10098 10099 10100]

Age unique values:
[28 33 22 23 40 45 34 37 24 30 38 47 55 59 29 35 44 54 36 32 41 56 21 27
 50 31 46 48 39 57 52 53 58 49 42 60 43 61 26 51 25]

Tenure unique values:
[ 6  2  1 16  9  3  8  4 20 10  7 15  5 18 17 14 13 19 12 11]

Turnover unique values:
['Yes' 'No']

HourlyRate unique values:
[24.37 22.52 88.77 ... 30.86 95.07 93.05]

HoursWeekly unique values:
[40]

CompensationType unique values:
['Salary']

AnnualSalary unique values:
[ 50689.6  46841.6 284641.6 ... 166316.8  64200.8 164902.4]

DrivingCommuterDistance unique values:
[ 89.  35.  12.   0.  76.  15.   2.  36.  60.  14.  75.   5. 910.  20.
  49.  33.  79.  50.  13.  57.  82.   4.  42.  64.  77.   1.  24.  67.
  41.  62.  47.  31.  32.  56.  22.  51.  58.  34.  37.  16.  26.  74.
  84.  44.  61.  87.  46.  52.  73.  53.  85.  66.  80.  55. 950.  70.
  17. 250.  43.  81.  86.  48.  45.  68.  10.   7.  28.  38.  21.  65.
  78.  69.  25.  54.   8.  83.  27.   9.  39.  40.  29.  71.  30.   3.
  63.  72. 125.  88.  59.  11.   6.  23. 322.  99.  91.  94.  97.  93.
  92.  90.  96.  95.  98.]

JobRoleArea unique values:
['Research' 'Information_Technology' 'Sales' 'Human_Resources'
 'Laboratory' 'Manufacturing' 'Healthcare' 'Marketing'
 'InformationTechnology' 'HumanResources' 'Information Technology'
 'Human Resources']

Gender unique values:
['Female' 'Prefer Not to Answer' 'Male']

MaritalStatus unique values:
['Married' 'Single' 'Divorced']

NumCompaniesPreviouslyWorked unique values:
[3. 6. 1. 7. 4. 2. 5. 8. 9.]

AnnualProfessionalDevHrs unique values:
[ 7.  8. 15. 19. 23. 25.  9.  6.  5. 16. 24. 10. 18. 11. 20. 17. 21. 12.
 22. 14. 13.]

PaycheckMethod unique values:
['Mail Check' 'Direct_Deposit' 'Direct Deposit' 'Mail_Check' 'MailedCheck']

TextMessageOptIn unique values:
['Yes' 'Unknown' 'No']

[Done] exited with code=0 in 4.12 seconds
```

Appendix: Column-by-Column Analysis

This appendix demonstrates column-by-column inspection for each variable in the Employee Turnover dataset. It includes summaries of data types, missing values, negative value checks, and visual boxplots for numeric variables. Categorical fields were analyzed using value counts.

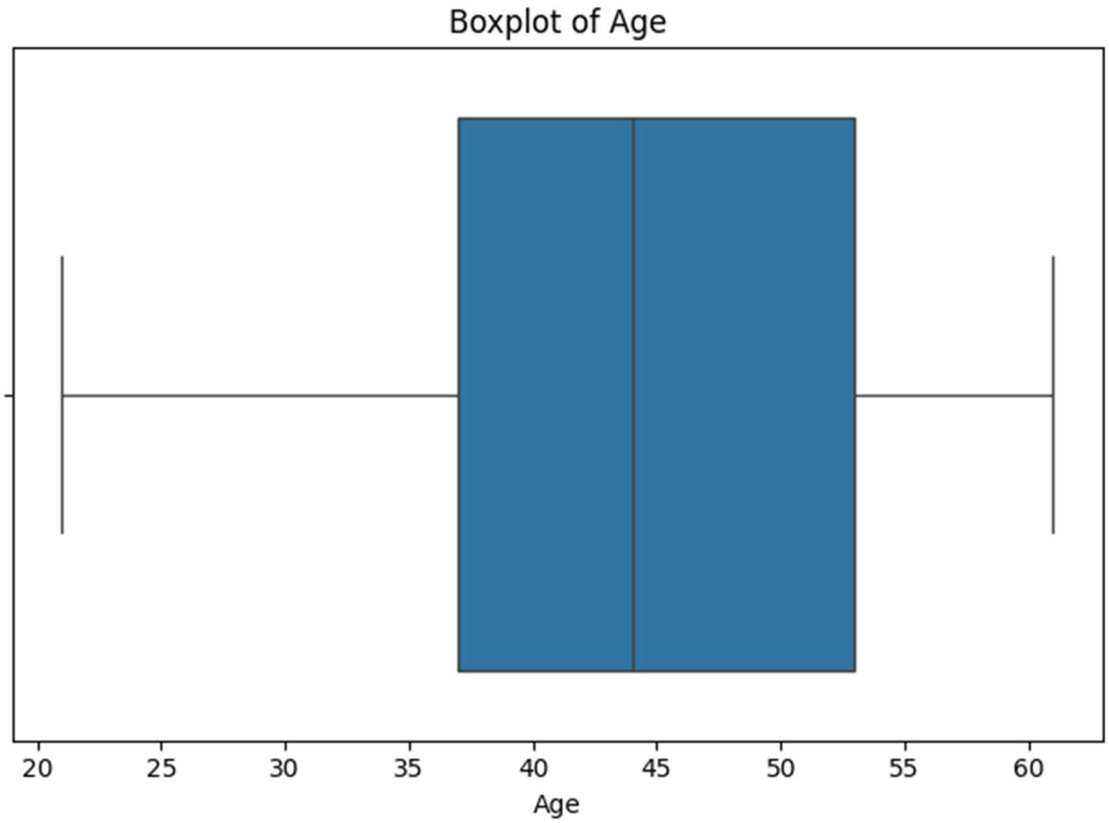
EmployeeNumber

The `EmployeeNumber` column was evaluated for data type, missing values, and negative values (if numeric). For numerical columns, a boxplot was created to help visually assess the distribution and check for outliers. For categorical variables, frequencies were reviewed to identify inconsistent values.



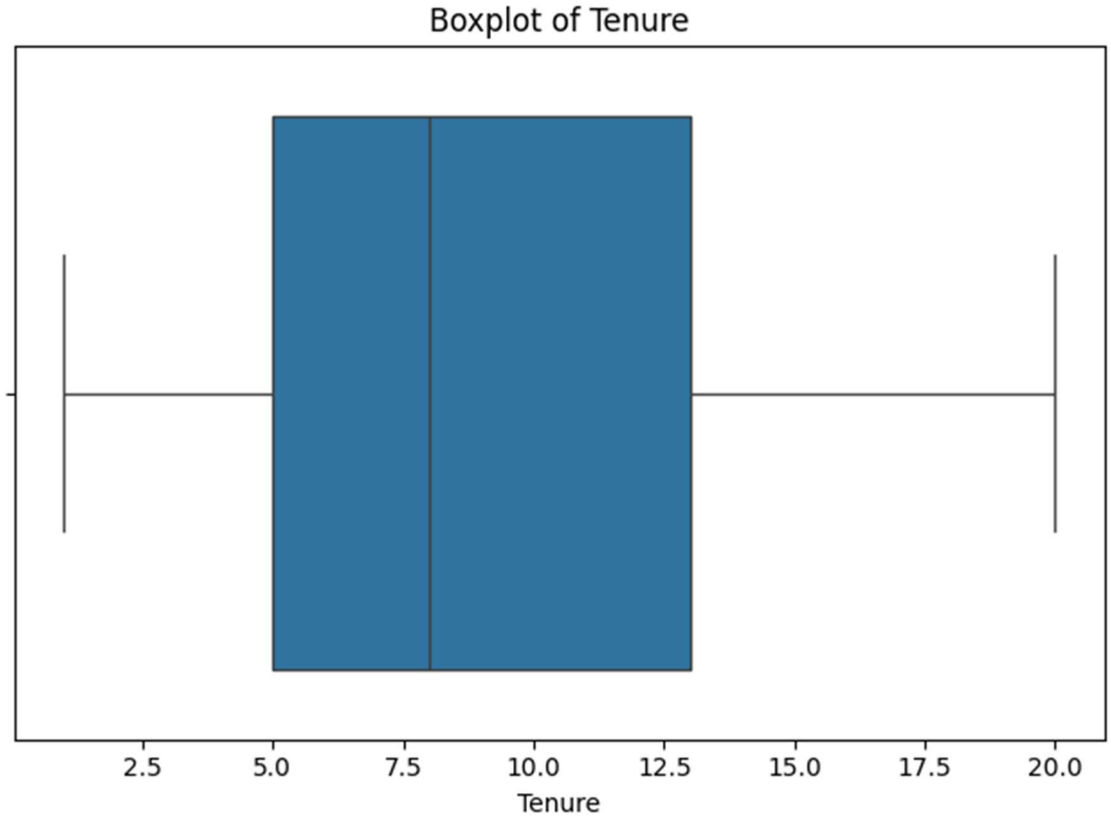
Age

The `Age` column was evaluated for data type, missing values, and negative values (if numeric). For numerical columns, a boxplot was created to help visually assess the distribution and check for outliers. For categorical variables, frequencies were reviewed to identify inconsistent values.



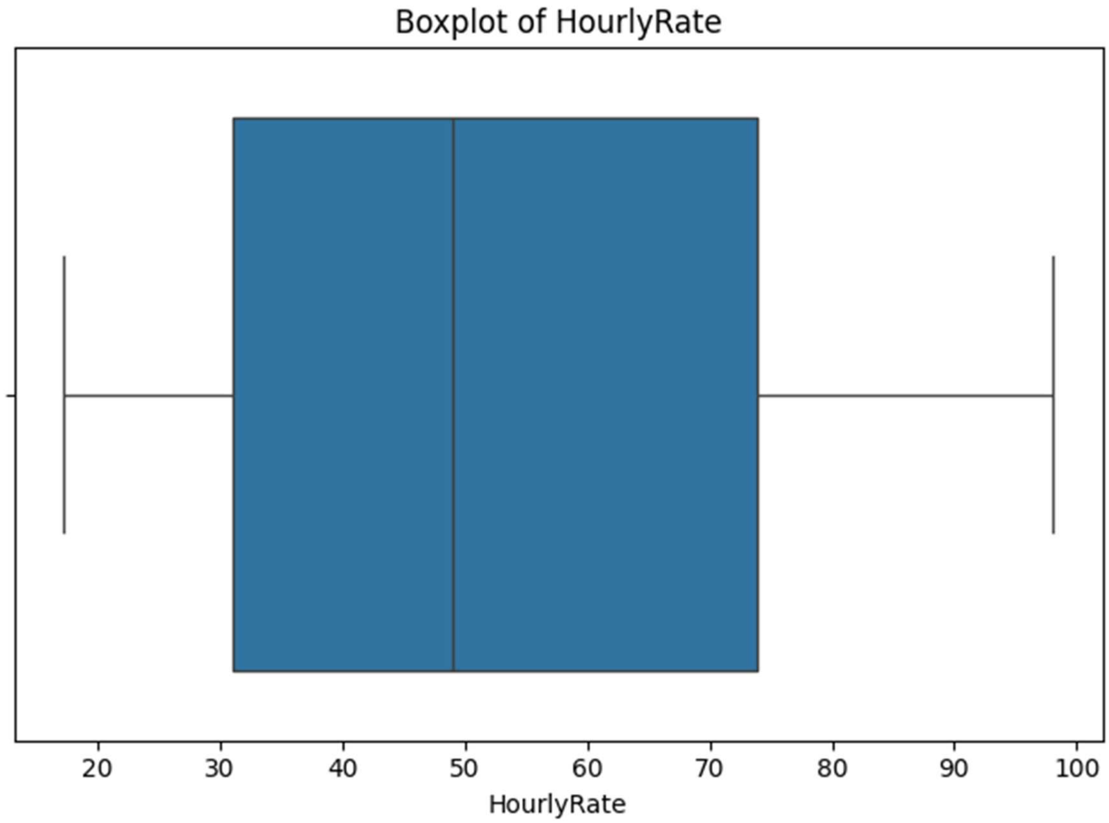
Tenure

The `Tenure` column was evaluated for data type, missing values, and negative values (if numeric). For numerical columns, a boxplot was created to help visually assess the distribution and check for outliers. For categorical variables, frequencies were reviewed to identify inconsistent values.



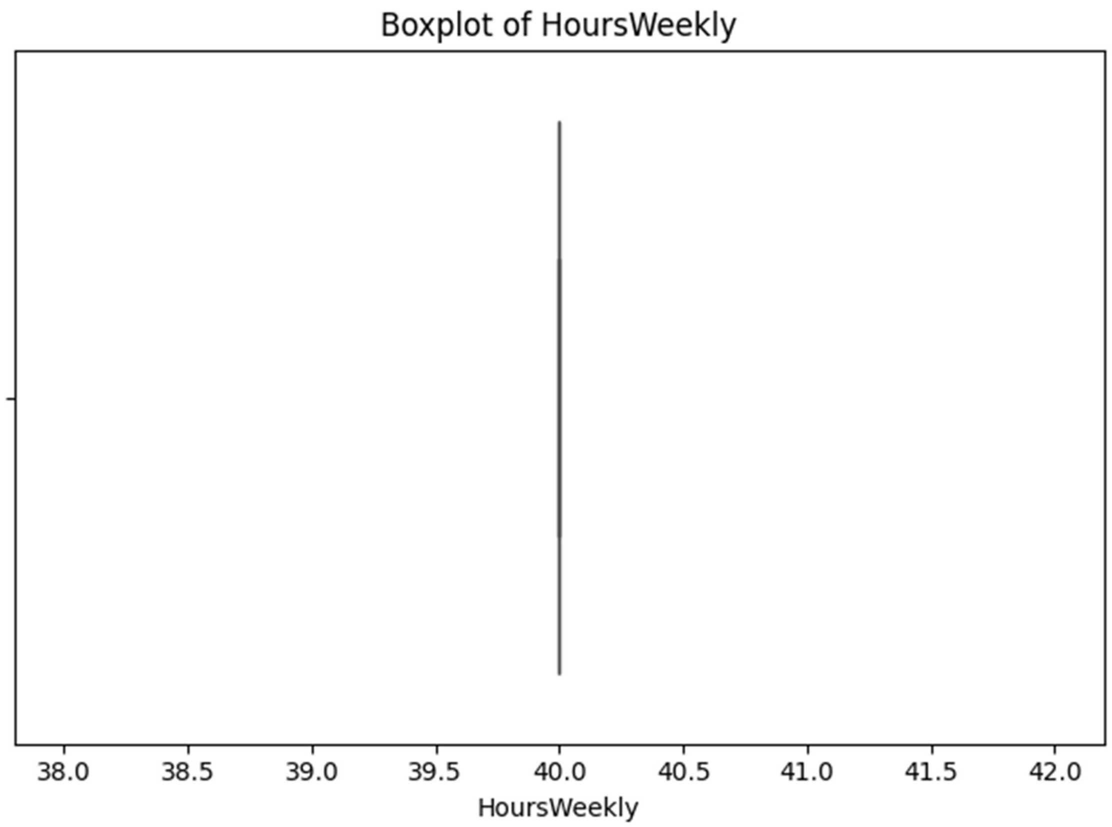
HourlyRate

The `HourlyRate` column was evaluated for data type, missing values, and negative values (if numeric). For numerical columns, a boxplot was created to help visually assess the distribution and check for outliers. For categorical variables, frequencies were reviewed to identify inconsistent values.



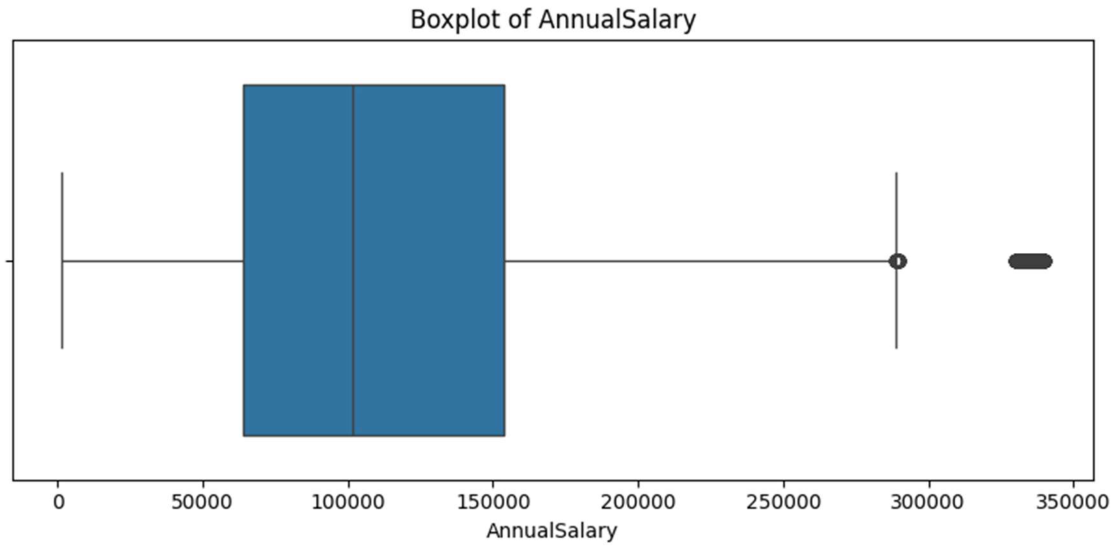
HoursWeekly

The `HoursWeekly` column was evaluated for data type, missing values, and negative values (if numeric). For numerical columns, a boxplot was created to help visually assess the distribution and check for outliers. For categorical variables, frequencies were reviewed to identify inconsistent values.



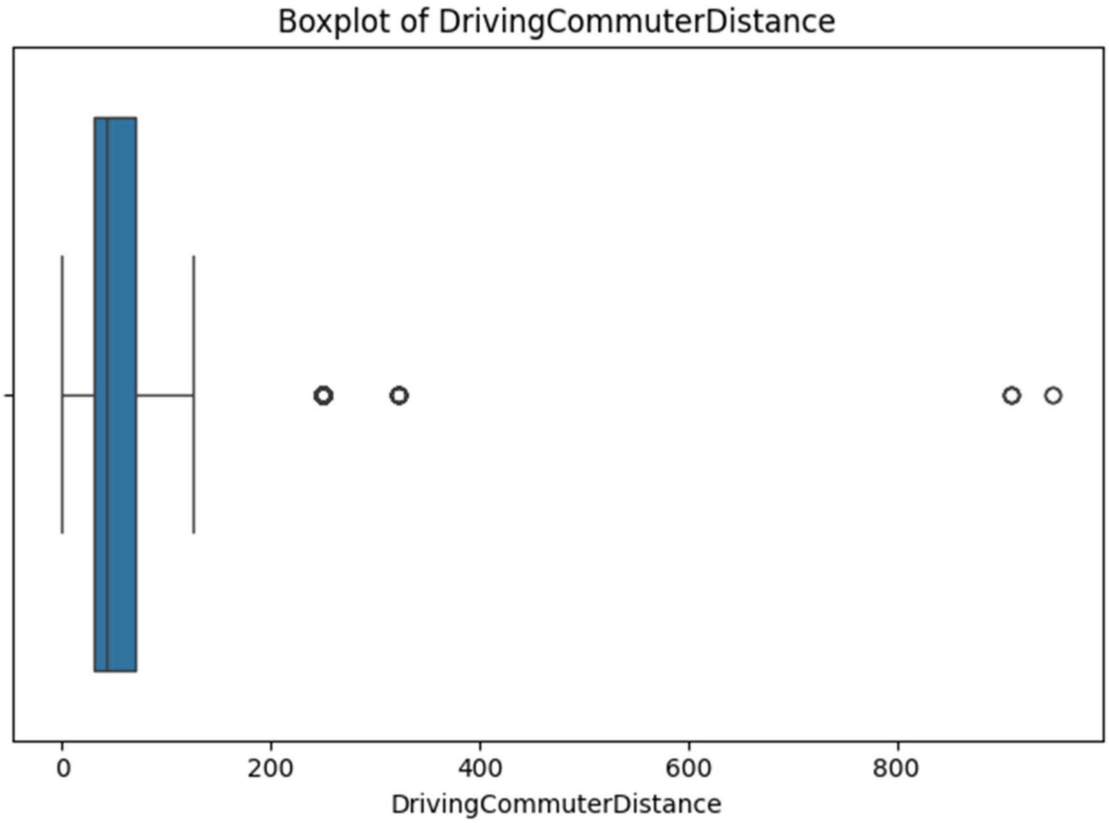
AnnualSalary

The `AnnualSalary` column was evaluated for data type, missing values, and negative values (if numeric). For numerical columns, a boxplot was created to help visually assess the distribution and check for outliers. For categorical variables, frequencies were reviewed to identify inconsistent values.



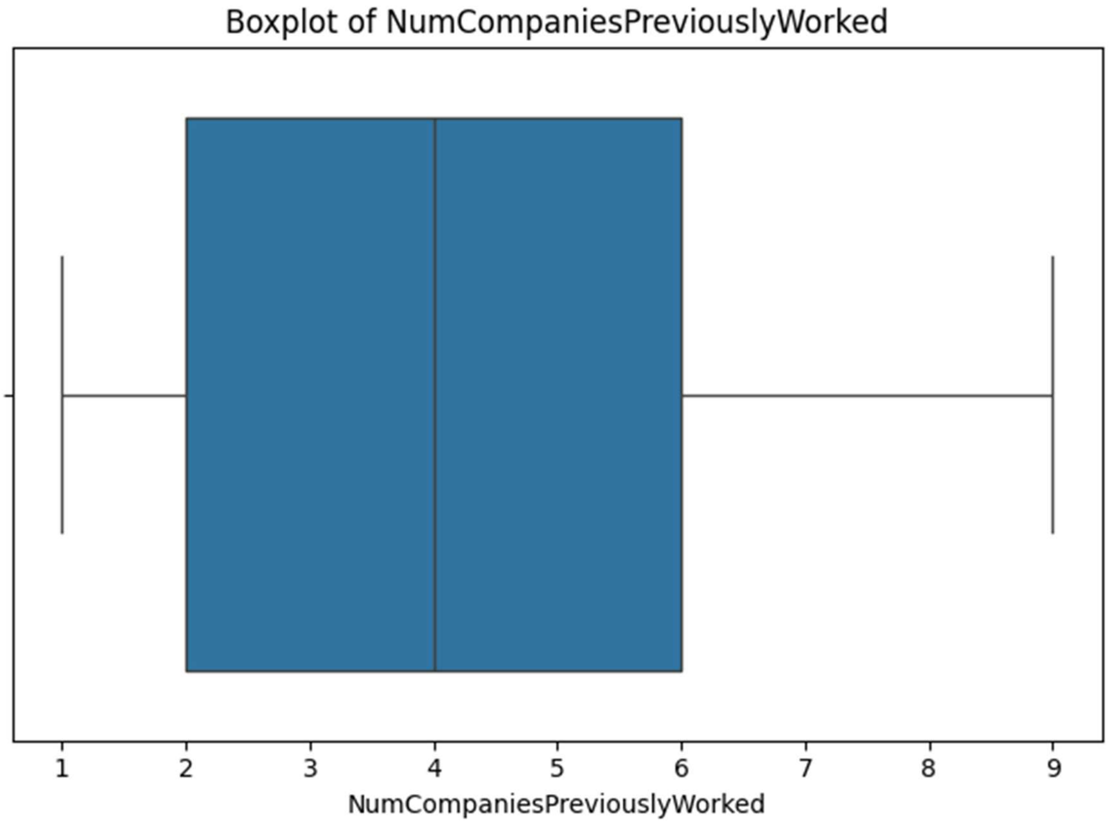
DrivingCommuterDistance

The `DrivingCommuterDistance` column was evaluated for data type, missing values, and negative values (if numeric). For numerical columns, a boxplot was created to help visually assess the distribution and check for outliers. For categorical variables, frequencies were reviewed to identify inconsistent values.



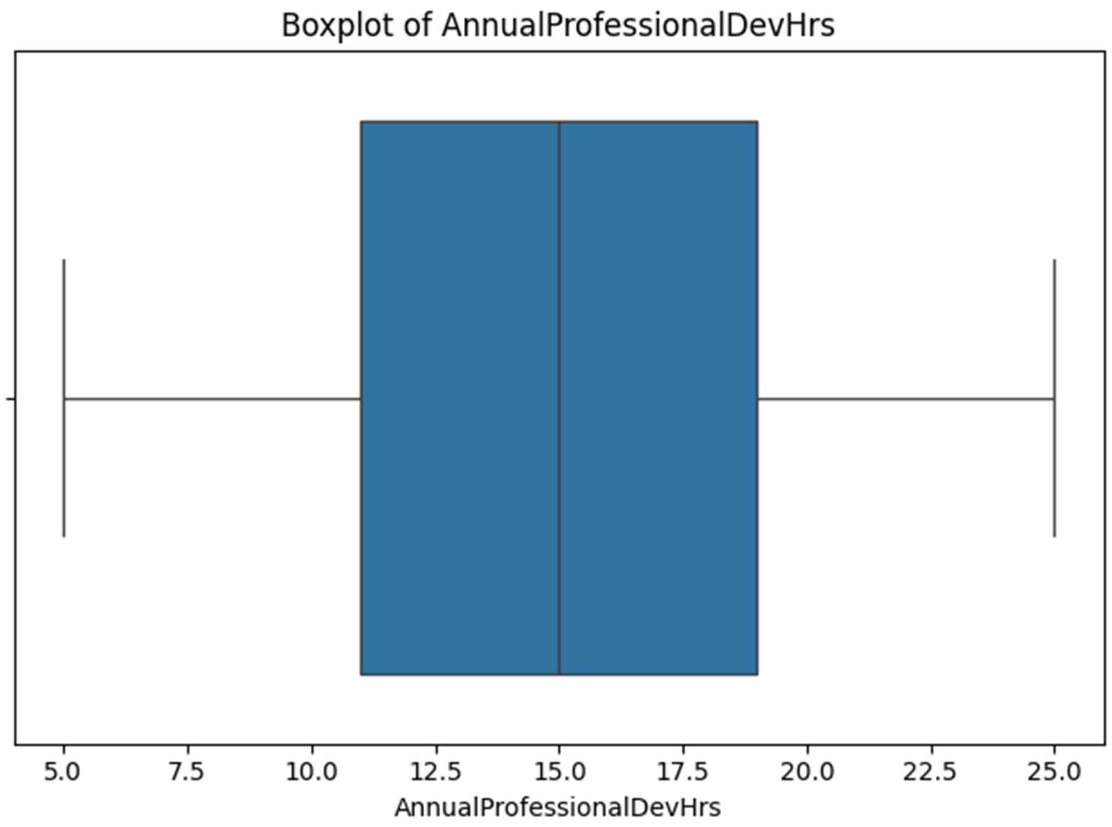
NumCompaniesPreviouslyWorked

The `NumCompaniesPreviouslyWorked` column was evaluated for data type, missing values, and negative values (if numeric). For numerical columns, a boxplot was created to help visually assess the distribution and check for outliers. For categorical variables, frequencies were reviewed to identify inconsistent values.



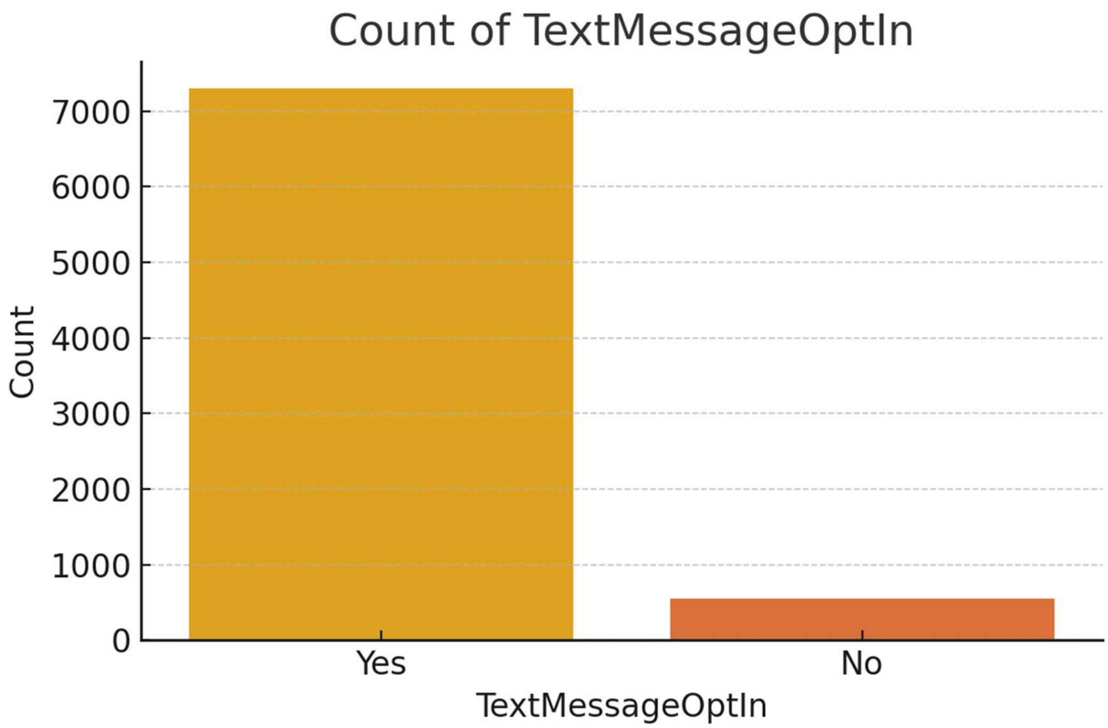
AnnualProfessionalDevHrs

The `AnnualProfessionalDevHrs` column was evaluated for data type, missing values, and negative values (if numeric). For numerical columns, a boxplot was created to help visually assess the distribution and check for outliers. For categorical variables, frequencies were reviewed to identify inconsistent values.



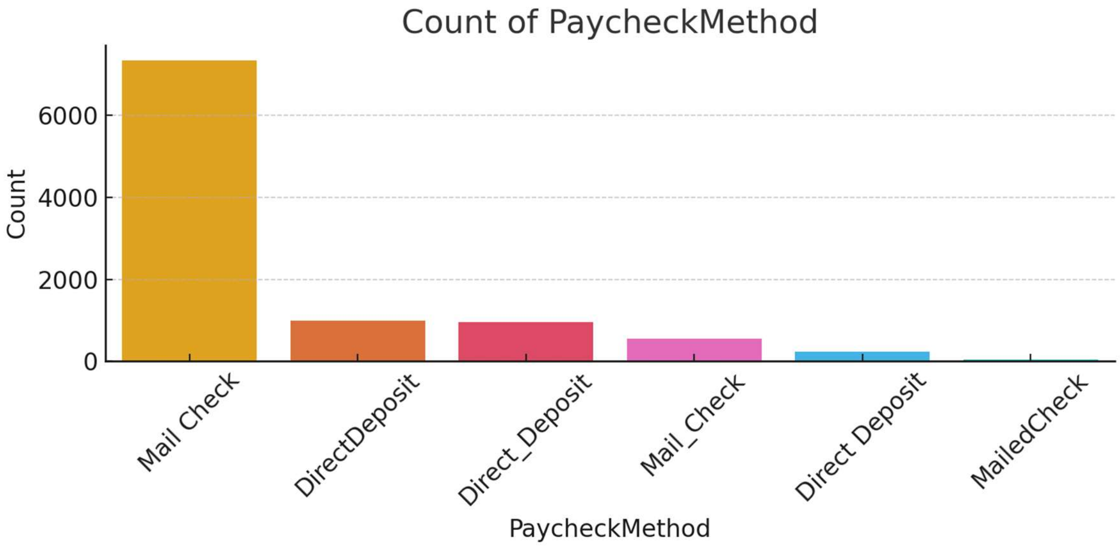
TextMessageOptIn

The `TextMessageOptIn` column was evaluated for data type, missing values, and negative values (if numeric). For numerical columns, a boxplot was created to help visually assess the distribution and check for outliers. For categorical variables, frequencies were reviewed to identify inconsistent values.



PaycheckMethod

The `PaycheckMethod` column was evaluated for data type, missing values, and negative values (if numeric). For numerical columns, a boxplot was created to help visually assess the distribution and check for outliers. For categorical variables, frequencies were reviewed to identify inconsistent values.



Updated Dataset Columns

This version of the report reflects updates based on evaluator feedback. Only the following 16 columns from the original dataset are included in the updated analysis:

- - EmployeeNumber
- - Age
- - Tenure
- - Turnover
- - HourlyRate
- - HoursWeekly
- - CompensationType
- - AnnualSalary
- - DrivingCommuterDistance
- - JobRoleArea
- - Gender
- - MaritalStatus
- - NumCompaniesPreviouslyWorked
- - AnnualProfessionalDevHrs
- - PaycheckMethod
- - TextMessageOptIn

All quality checks, cleaning procedures, and analysis were limited strictly to these columns.