Hands-on Lab 7 Removing Duplicates_v2

March 22, 2025

1 Removing Duplicates

Estimated time needed: **30** minutes

1.1 Introduction

In this lab, you will focus on data wrangling, an important step in preparing data for analysis. Data wrangling involves cleaning and organizing data to make it suitable for analysis. One key task in this process is removing duplicate entries, which are repeated entries that can distort analysis and lead to inaccurate conclusions.

1.2 Objectives

In this lab you will perform the following:

1. Identify duplicate rows in the dataset.
2. Use suitable techniques to remove duplicate rows and verify the removal.
3. Summarize how to handle missing values appropriately.
4. Use ConvertedCompYearly to normalize compensation data.

1.2.1 Install the Required Libraries

[1]:
```
!pip install pandas
```

```
Collecting pandas
  Downloading
pandas-2.2.3-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(89 kB)
Collecting numpy>=1.26.0 (from pandas)
  Downloading
numpy-2.2.4-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(62 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in
/opt/conda/lib/python3.12/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.12/site-
packages (from pandas) (2024.2)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2025.1-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-
packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

```
Downloading
pandas-2.2.3-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.7
MB)
                           12.7/12.7 MB
130.3 MB/s eta 0:00:00
Downloading
numpy-2.2.4-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.1 MB)
                           16.1/16.1 MB
149.3 MB/s eta 0:00:00
Downloading tzdata-2025.1-py2.py3-none-any.whl (346 kB)
Installing collected packages: tzdata, numpy, pandas
Successfully installed numpy-2.2.4 pandas-2.2.3 tzdata-2025.1
```

1.2.2 Step 1: Import Required Libraries

[2]:
```python
import pandas as pd
```

1.2.3 Step 2: Load the Dataset into a DataFrame

load the dataset using pd.read_csv()

[17]:
```python
# Define the URL of the dataset
import pandas as pd

file_path = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
↪n01PQ9pSmiRX6520flujwQ/survey-data.csv"

try:

    df = pd.read_csv(file_path, encoding='utf-8')
    print("Data loaded successfully!")
    print(df.head())
except Exception as e:
    print("Error loading data:", e)
```

```
Data loaded successfully!
   ResponseId                      MainBranch                 Age  \
0           1  I am a developer by profession  Under 18 years old
1           2  I am a developer by profession     35-44 years old
2           3  I am a developer by profession     45-54 years old
3           4           I am learning to code     18-24 years old
4           5  I am a developer by profession     18-24 years old

            Employment RemoteWork   Check  \
0  Employed, full-time     Remote  Apples
1  Employed, full-time     Remote  Apples
2  Employed, full-time     Remote  Apples
3   Student, full-time        NaN  Apples
```

```
4   Student, full-time        NaN  Apples

                                      CodingActivities  \
0                                                Hobby
1  Hobby;Contribute to open-source projects;Other…
2  Hobby;Contribute to open-source projects;Other…
3                                                  NaN
4                                                  NaN

                                               EdLevel  \
0                           Primary/elementary school
1       Bachelor's degree (B.A., B.S., B.Eng., etc.)
2    Master's degree (M.A., M.S., M.Eng., MBA, etc.)
3  Some college/university study without earning …
4  Secondary school (e.g. American high school, G…

                                             LearnCode  \
0                                Books / Physical media
1  Books / Physical media;Colleague;On the job tr…
2  Books / Physical media;Colleague;On the job tr…
3  Other online resources (e.g., videos, blogs, f…
4  Other online resources (e.g., videos, blogs, f…

                                       LearnCodeOnline  … JobSatPoints_6  \
0                                                  NaN  …            NaN
1  Technical documentation;Blogs;Books;Written Tu…  …            0.0
2  Technical documentation;Blogs;Books;Written Tu…  …            NaN
3  Stack Overflow;How-to videos;Interactive tutorial  …            NaN
4  Technical documentation;Blogs;Written Tutorial…  …            NaN

  JobSatPoints_7 JobSatPoints_8 JobSatPoints_9 JobSatPoints_10  \
0            NaN            NaN            NaN             NaN
1            0.0            0.0            0.0             0.0
2            NaN            NaN            NaN             NaN
3            NaN            NaN            NaN             NaN
4            NaN            NaN            NaN             NaN

  JobSatPoints_11           SurveyLength SurveyEase ConvertedCompYearly JobSat
0             NaN                    NaN        NaN                 NaN    NaN
1             0.0                    NaN        NaN                 NaN    NaN
2             NaN  Appropriate in length       Easy                 NaN    NaN
3             NaN               Too long       Easy                 NaN    NaN
4             NaN              Too short       Easy                 NaN    NaN

[5 rows x 114 columns]
```

Note: If you are working on a local Jupyter environment, you can use the URL directly in the pandas.read_csv() function as shown below:

#df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/n01PQ9pSmiRX6520fluj
data.csv")

1.2.4 Step 3: Identifying Duplicate Rows

Task 1: Identify Duplicate Rows 1. Count the number of duplicate rows in the dataset. 2. Display the first few duplicate rows to understand their structure.

[8]:
```python
# Count the number of duplicate rows
num_duplicates = df.duplicated().sum()
print("Number of duplicate rows:", num_duplicates)

# Display the first few duplicate rows
duplicate_rows = df[df.duplicated()]
print("First few duplicate rows:")
print(duplicate_rows.head())
```

```
Number of duplicate rows: 0
First few duplicate rows:
Empty DataFrame
Columns: [ResponseId, MainBranch, Age, Employment, RemoteWork, Check,
CodingActivities, EdLevel, LearnCode, LearnCodeOnline, TechDoc, YearsCode,
YearsCodePro, DevType, OrgSize, PurchaseInfluence, BuyNewTool, BuildvsBuy,
TechEndorse, Country, Currency, CompTotal, LanguageHaveWorkedWith,
LanguageWantToWorkWith, LanguageAdmired, DatabaseHaveWorkedWith,
DatabaseWantToWorkWith, DatabaseAdmired, PlatformHaveWorkedWith,
PlatformWantToWorkWith, PlatformAdmired, WebframeHaveWorkedWith,
WebframeWantToWorkWith, WebframeAdmired, EmbeddedHaveWorkedWith,
EmbeddedWantToWorkWith, EmbeddedAdmired, MiscTechHaveWorkedWith,
MiscTechWantToWorkWith, MiscTechAdmired, ToolsTechHaveWorkedWith,
ToolsTechWantToWorkWith, ToolsTechAdmired, NEWCollabToolsHaveWorkedWith,
NEWCollabToolsWantToWorkWith, NEWCollabToolsAdmired, OpSysPersonal use,
OpSysProfessional use, OfficeStackAsyncHaveWorkedWith,
OfficeStackAsyncWantToWorkWith, OfficeStackAsyncAdmired,
OfficeStackSyncHaveWorkedWith, OfficeStackSyncWantToWorkWith,
OfficeStackSyncAdmired, AISearchDevHaveWorkedWith, AISearchDevWantToWorkWith,
AISearchDevAdmired, NEWSOSites, SOVisitFreq, SOAccount, SOPartFreq, SOHow,
SOComm, AISelect, AISent, AIBen, AIAcc, AIComplex, AIToolCurrently Using,
AIToolInterested in Using, AIToolNot interested in Using, AINextMuch more
integrated, AINextNo change, AINextMore integrated, AINextLess integrated,
AINextMuch less integrated, AIThreat, AIEthics, AIChallenges, TBranch, ICorPM,
WorkExp, Knowledge_1, Knowledge_2, Knowledge_3, Knowledge_4, Knowledge_5,
Knowledge_6, Knowledge_7, Knowledge_8, Knowledge_9, Frequency_1, Frequency_2,
Frequency_3, TimeSearching, TimeAnswering, Frustration, ProfessionalTech,
ProfessionalCloud, ProfessionalQuestion, …]
Index: []

[0 rows x 114 columns]
```

1.2.5 Step 4: Removing Duplicate Rows

Task 2: Remove Duplicates 1. Remove duplicate rows from the dataset using the drop_duplicates() function. 2. Verify the removal by counting the number of duplicate rows after removal .

[9]:
```python
## Write your code here
# Remove duplicate rows
df = df.drop_duplicates()

# Verify the removal by counting the number of duplicate rows
num_duplicates = df.duplicated().sum()
print("Number of duplicate rows after removal:", num_duplicates)
```

```
Number of duplicate rows after removal: 0
```

1.2.6 Step 5: Handling Missing Values

Task 3: Identify and Handle Missing Values 1. Identify missing values for all columns in the dataset. 2. Choose a column with significant missing values (e.g., EdLevel) and impute with the most frequent value.

[10]:
```python
## Write your code here
# Identify missing values for all columns
missing_values = df.isnull().sum()
print("Missing values per column:")
print(missing_values)

# Impute missing values in the 'EdLevel' column with the most frequent value
most_frequent = df['EdLevel'].mode()[0]
df['EdLevel'].fillna(most_frequent, inplace=True)

# Verify the imputation
print("\nMissing values in 'EdLevel' after imputation:")
print(df['EdLevel'].isnull().sum())
```

```
Missing values per column:
ResponseId                0
MainBranch                0
Age                       0
Employment                0
RemoteWork            10631
                      …
JobSatPoints_11       35992
SurveyLength           9255
SurveyEase             9199
ConvertedCompYearly   42002
JobSat                36311
Length: 114, dtype: int64
```

```
Missing values in 'EdLevel' after imputation:
0

/tmp/ipykernel_300/462093557.py:9: FutureWarning: A value is trying to be set on
a copy of a DataFrame or Series through chained assignment using an inplace
method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  df['EdLevel'].fillna(most_frequent, inplace=True)
```

1.2.7 Step 6: Normalizing Compensation Data

Task 4: Normalize Compensation Data Using ConvertedCompYearly 1. Use the ConvertedCompYearly column for compensation analysis as the normalized annual compensation is already provided. 2. Check for missing values in ConvertedCompYearly and handle them if necessary.

[11]:
```python
## Write your code here
# Check for missing values in ConvertedCompYearly
missing_comp = df['ConvertedCompYearly'].isnull().sum()
print("Missing values in 'ConvertedCompYearly':", missing_comp)

# Handle missing values by imputing with the median
median_comp = df['ConvertedCompYearly'].median()
df['ConvertedCompYearly'].fillna(median_comp, inplace=True)

# Verify the imputation
missing_comp_after = df['ConvertedCompYearly'].isnull().sum()
print("Missing values in 'ConvertedCompYearly' after imputation:",
 ↪missing_comp_after)
```

```
Missing values in 'ConvertedCompYearly': 42002
Missing values in 'ConvertedCompYearly' after imputation: 0

/tmp/ipykernel_300/2462540287.py:8: FutureWarning: A value is trying to be set
on a copy of a DataFrame or Series through chained assignment using an inplace
method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
```

instead, to perform the operation inplace on the original object.

```
df['ConvertedCompYearly'].fillna(median_comp, inplace=True)
```

1.2.8 Step 7: Summary and Next Steps

In this lab, you focused on identifying and removing duplicate rows.

- You handled missing values by imputing the most frequent value in a chosen column.
- You used ConvertedCompYearly for compensation normalization and handled missing values.
- For further analysis, consider exploring other columns or visualizing the cleaned dataset.

[14]:
```
pip install matplotlib seaborn
```

```
Collecting matplotlib
  Downloading matplotlib-3.10.1-cp312-cp312-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (11 kB)
Collecting seaborn
  Downloading seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.3.1-cp312-cp312-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.4 kB)
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.56.0-cp312-cp312-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl.metadata (101 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Downloading kiwisolver-1.4.8-cp312-cp312-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.2 kB)
Requirement already satisfied: numpy>=1.23 in /opt/conda/lib/python3.12/site-
packages (from matplotlib) (2.2.4)
Requirement already satisfied: packaging>=20.0 in
/opt/conda/lib/python3.12/site-packages (from matplotlib) (24.2)
Collecting pillow>=8 (from matplotlib)
  Downloading pillow-11.1.0-cp312-cp312-manylinux_2_28_x86_64.whl.metadata (9.1
kB)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Downloading pyparsing-3.2.1-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: python-dateutil>=2.7 in
/opt/conda/lib/python3.12/site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: pandas>=1.2 in /opt/conda/lib/python3.12/site-
packages (from seaborn) (2.2.3)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.12/site-
packages (from pandas>=1.2->seaborn) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.12/site-
```

```
packages (from pandas>=1.2->seaborn) (2025.1)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-
packages (from python-dateutil>=2.7->matplotlib) (1.17.0)
Downloading
matplotlib-3.10.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(8.6 MB)
                          8.6/8.6 MB
140.0 MB/s eta 0:00:00
Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
Downloading
contourpy-1.3.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (323
kB)
Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.56.0-cp312-cp312-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (4.9 MB)
                          4.9/4.9 MB
114.8 MB/s eta 0:00:00
Downloading
kiwisolver-1.4.8-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.5
MB)
                          1.5/1.5 MB
93.8 MB/s eta 0:00:00
Downloading pillow-11.1.0-cp312-cp312-manylinux_2_28_x86_64.whl (4.5 MB)
                          4.5/4.5 MB
157.4 MB/s eta 0:00:00
Downloading pyparsing-3.2.1-py3-none-any.whl (107 kB)
Installing collected packages: pyparsing, pillow, kiwisolver, fonttools, cycler,
contourpy, matplotlib, seaborn
Successfully installed contourpy-1.3.1 cycler-0.12.1 fonttools-4.56.0
kiwisolver-1.4.8 matplotlib-3.10.1 pillow-11.1.0 pyparsing-3.2.1 seaborn-0.13.2
Note: you may need to restart the kernel to use updated packages.
```

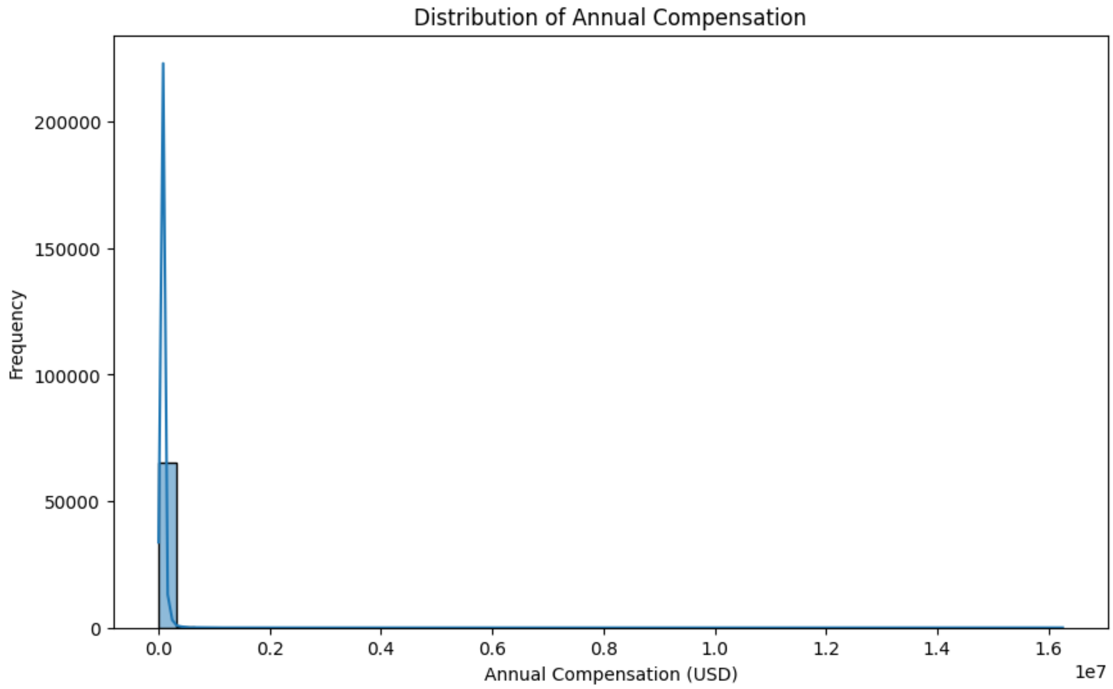
[15]:
```python
import matplotlib.pyplot as plt
import seaborn as sns
```

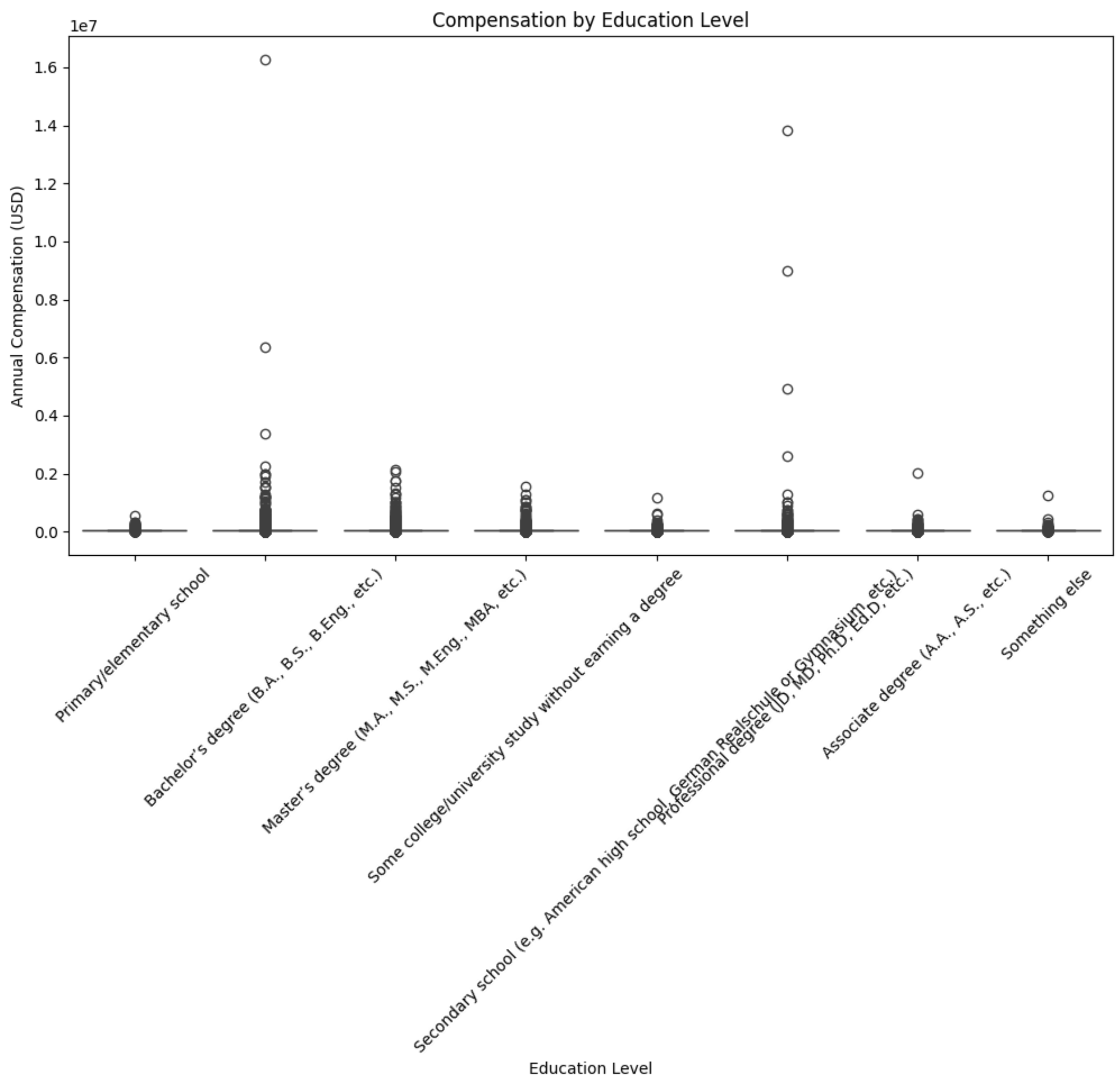
[16]:
```python
## Write your code here

print("Summary statistics:")
print(df.describe())

# Visualize the distribution of annual compensation
plt.figure(figsize=(10, 6))
sns.histplot(df['ConvertedCompYearly'], bins=50, kde=True)
plt.title("Distribution of Annual Compensation")
plt.xlabel("Annual Compensation (USD)")
plt.ylabel("Frequency")
```

```python
plt.show()

# Visualize the relationship between education level and compensation
plt.figure(figsize=(12, 6))
sns.boxplot(data=df, x='EdLevel', y='ConvertedCompYearly')
plt.title("Compensation by Education Level")
plt.xlabel("Education Level")
plt.ylabel("Annual Compensation (USD)")
plt.xticks(rotation=45)
plt.show()
```

```
Summary statistics:
         ResponseId      CompTotal       WorkExp  JobSatPoints_1  \
count  65437.000000   3.374000e+04  29658.000000    29324.000000
mean   32719.000000  2.963841e+145     11.466957       18.581094
std    18890.179119  5.444117e+147      9.168709       25.966221
min        1.000000   0.000000e+00      0.000000        0.000000
25%    16360.000000   6.000000e+04      4.000000        0.000000
50%    32719.000000   1.100000e+05      9.000000       10.000000
75%    49078.000000   2.500000e+05     16.000000       22.000000
max    65437.000000  1.000000e+150     50.000000      100.000000

       JobSatPoints_4  JobSatPoints_5  JobSatPoints_6  JobSatPoints_7  \
count    29393.000000    29411.000000    29450.000000     29448.00000
mean         7.522140       10.060857       24.343232        22.96522
std         18.422661       21.833836       27.089360        27.01774
min          0.000000        0.000000        0.000000         0.00000
25%          0.000000        0.000000        0.000000         0.00000
50%          0.000000        0.000000       20.000000        15.00000
75%          5.000000       10.000000       30.000000        30.00000
max        100.000000      100.000000      100.000000       100.00000

       JobSatPoints_8  JobSatPoints_9  JobSatPoints_10  JobSatPoints_11  \
count    29456.000000    29456.000000     29450.000000     29445.000000
mean        20.278165       16.169432        10.955713         9.953948
std         26.108110       24.845032        22.906263        21.775652
min          0.000000        0.000000         0.000000         0.000000
25%          0.000000        0.000000         0.000000         0.000000
50%         10.000000        5.000000         0.000000         0.000000
75%         25.000000       20.000000        10.000000        10.000000
max        100.000000      100.000000       100.000000       100.000000

       ConvertedCompYearly        JobSat
count         6.543700e+04  29126.000000
mean          7.257636e+04      6.935041
std           1.122207e+05      2.088259
min           1.000000e+00      0.000000
25%           6.500000e+04      6.000000
```

```
50%            6.500000e+04      7.000000
75%            6.500000e+04      8.000000
max            1.625660e+07     10.000000
```





<!– ## Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2024-11-05	1.2	Madhusudhan Moole	Updated lab
2024-09-24	1.1	Madhusudhan Moole	Updated lab
2024-09-23	1.0	Raghul Ramesh	Created lab

–!>