```
from google.colab import files
uploaded = files.upload()
```

Choose Files adult.csv

- **adult.csv**(text/csv) - 5326368 bytes, last modified: 3/3/2025 - 100% done

```
Saving adult.csv to adult.csv
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OrdinalEncoder, OneHotEncoder
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from scipy import stats
```

```
import pandas as pd

# Replace 'your_file.csv' with the name of the file you just uploaded
df = pd.read_csv('adult.csv')
df.head()  # Display the first few rows
```

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United States
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United States

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.head(10)
```

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United States
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United States
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United States
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United States
4	18	?	103497	Some-	10	Never-	?	Own-child	White	Female	0	0	30	United

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.shape
```

```
(48842, 15)
```

```
print(df.describe())
```

```
                age        fnlwgt  educational-num  capital-gain  \
count  48842.000000  4.884200e+04     48842.000000  48842.000000
mean      38.643585  1.896641e+05        10.078089   1079.067626
std       13.710510  1.056040e+05         2.570973   7452.019058
min       17.000000  1.228500e+04         1.000000      0.000000
25%       28.000000  1.175505e+05         9.000000      0.000000
50%       37.000000  1.781445e+05        10.000000      0.000000
```

```
75%       48.000000  2.376420e+05        12.000000      0.000000
max       90.000000  1.490400e+06        16.000000  99999.000000

       capital-loss  hours-per-week
count  48842.000000    48842.000000
mean      87.502314       40.422382
std      403.004552       12.391444
min        0.000000        1.000000
25%        0.000000       40.000000
50%        0.000000       40.000000
75%        0.000000       45.000000
max     4356.000000       99.000000
```

```
#Code to Find Missing Values
# Check for missing values in each column
missing_values = df.isnull().sum()

# Display columns with missing values
print(missing_values[missing_values > 0])
```

```
Series([], dtype: int64)
```

```
import numpy as np

# Introduce missing values at specific locations
df.loc[5, 'educational-num'] = np.nan  # Set missing value for 'AGE' at row index 5
df.loc[7, 'age'] = np.nan  # Set missing value for 'BMI' at row index 10

# Display the first 10 rows to check the changes
print(df.head(10))
df
```

```
0  25.0          Private  226802         11th              7.0
1  38.0          Private   89814      HS-grad              9.0
2  28.0        Local-gov  336951   Assoc-acdm             12.0
3  44.0          Private  160323  Some-college            10.0
4  18.0                ?  103497  Some-college            10.0
5  34.0          Private  198693         10th              NaN
6  29.0                ?  227026      HS-grad              9.0
7   NaN  Self-emp-not-inc  104626   Prof-school            15.0
8  24.0          Private  369667  Some-college            10.0
9  55.0          Private  104996       7th-8th              4.0

       marital-status         occupation   relationship   race  gender  \
0       Never-married  Machine-op-inspct      Own-child  Black    Male
1  Married-civ-spouse    Farming-fishing        Husband  White    Male
2  Married-civ-spouse    Protective-serv        Husband  White    Male
3  Married-civ-spouse  Machine-op-inspct        Husband  Black    Male
4       Never-married                  ?      Own-child  White  Female
5       Never-married      Other-service  Not-in-family  White    Male
6       Never-married                  ?      Unmarried  Black    Male
7  Married-civ-spouse     Prof-specialty        Husband  White    Male
8       Never-married      Other-service      Unmarried  White  Female
9  Married-civ-spouse       Craft-repair        Husband  White    Male

   capital-gain  capital-loss  hours-per-week native-country income
0             0             0              40  United-States  <=50K
1             0             0              50  United-States  <=50K
2             0             0              40  United-States   >50K
3          7688             0              40  United-States   >50K
4             0             0              30  United-States  <=50K
5             0             0              30  United-States  <=50K
6             0             0              40  United-States  <=50K
7          3103             0              32  United-States   >50K
8             0             0              40  United-States  <=50K
9             0             0              10  United-States  <=50K
```

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week
0	25.0	Private	226802	11th	7.0	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40
1	38.0	Private	89814	HS-grad	9.0	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50
2	28.0	Local-gov	336951	Assoc-acdm	12.0	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40
3	44.0	Private	160323	Some-college	10.0	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40
4	18.0	?	103497	Some-college	10.0	Never-married	?	Own-child	White	Female	0	0	30
...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	27.0	Private	257302	Assoc-	12.0	Married-civ-	Tech-	Wife	White	Female	0	0	38

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
print(df.describe())
```

```
                age        fnlwgt  educational-num  capital-gain  \
count  48841.000000  4.884200e+04     48841.000000  48842.000000
mean      38.643087  1.896641e+05        10.078172   1079.067626
std       13.710207  1.056040e+05         2.570933   7452.019058
min       17.000000  1.228500e+04         1.000000      0.000000
25%       28.000000  1.175505e+05         9.000000      0.000000
50%       37.000000  1.781445e+05        10.000000      0.000000
75%       48.000000  2.376420e+05        12.000000      0.000000
max       90.000000  1.490400e+06        16.000000  99999.000000

       capital-loss  hours-per-week
count  48842.000000    48842.000000
mean      87.502314       40.422382
std      403.004552       12.391444
min        0.000000        1.000000
25%        0.000000       40.000000
50%        0.000000       40.000000
75%        0.000000       45.000000
max     4356.000000       99.000000
```

```
#Code to Find Missing Values
# Check for missing values in each column
missing_values = df.isnull().sum()

# Display columns with missing values
print(missing_values[missing_values > 0])
```

```
age               1
educational-num   1
dtype: int64
```

```
#Set the values to some value (zero, the mean, the median, etc.).
# Step 1: Create an instance of SimpleImputer with the median strategy for Age and mean stratergy for Salary
imputer1 = SimpleImputer(strategy="median")
imputer2 = SimpleImputer(strategy="mean")

df_copy=df

# Step 2: Fit the imputer on the "Age" and "Salary"column
# Note: SimpleImputer expects a 2D array, so we reshape the column
imputer1.fit(df_copy[["educational-num"]])
imputer2.fit(df_copy[["age"]])

# Step 3: Transform (fill) the missing values in the "Age" and "Salary"c column
df_copy["educational-num"] = imputer1.transform(df[["educational-num"]])
df_copy["age"] = imputer2.transform(df[["age"]])

# Verify that there are no missing values left
print(df_copy["educational-num"].isnull().sum())
print(df_copy["age"].isnull().sum())
```

```
0
0
```

```
import pandas as pd
from sklearn.preprocessing import OrdinalEncoder, OneHotEncoder

# Normalize the Gender column to be consistent (uppercase in this case)
df['gender'] = df['gender'].str.upper()  # Convert to uppercase
df['race'] = df['race'].str.upper()
# Initialize OrdinalEncoder for 'Gender' column
ordinal_encoder = OrdinalEncoder(categories=[["FEMALE", "MALE"]])  # Encoding 'F' as 0, 'M' as 1

# Fit and transform the data in the 'Gender' column
df["gender_Encoded"] = ordinal_encoder.fit_transform(df[["gender"]])

# Initialize OneHotEncoder for the 'City' column (if the column exists)
# You should replace "City" with the actual column name in your dataset
onehot_encoder = OneHotEncoder()

# Fit and transform the "City" column (replace 'City' with the actual name of the column)
if 'race' in df.columns:
    encoded_data = onehot_encoder.fit_transform(df[["race"]])

    # Convert the sparse matrix to a dense array
    encoded_array = encoded_data.toarray()

    # Convert to DataFrame for better visualization
    encoded_df = pd.DataFrame(encoded_array, columns=onehot_encoder.get_feature_names_out(["race"]))

    # Concatenate the one-hot encoded columns with the original DataFrame
    df_encoded = pd.concat([df, encoded_df], axis=1)

    # Drop the original 'City' column as it is now encoded
    df_encoded.drop("race", axis=1, inplace=True)

# If there is no 'City' column, proceed with just encoding 'Gender'
else:
    df_encoded = df.copy()

# Drop the original 'Gender' column
df_encoded.drop("gender", axis=1, inplace=True)

# Display the first few rows of the encoded dataframe
```

```
print(df_encoded.head())
df_encoded
```

```
    age workclass  fnlwgt     education  educational-num      marital-status  \
0  25.0   Private  226802          11th              7.0       Never-married
1  38.0   Private   89814       HS-grad              9.0  Married-civ-spouse
2  28.0 Local-gov  336951    Assoc-acdm             12.0  Married-civ-spouse
3  44.0   Private  160323  Some-college             10.0  Married-civ-spouse
4  18.0         ?  103497  Some-college             10.0       Never-married

          occupation relationship  capital-gain  capital-loss  hours-per-week  \
0  Machine-op-inspct    Own-child             0             0              40
1    Farming-fishing      Husband             0             0              50
2    Protective-serv      Husband             0             0              40
3  Machine-op-inspct      Husband          7688             0              40
4                  ?    Own-child             0             0              30

  native-country income  gender_Encoded  race_AMER-INDIAN-ESKIMO  \
0  United-States  <=50K             1.0                      0.0
1  United-States  <=50K             1.0                      0.0
2  United-States   >50K             1.0                      0.0
3  United-States   >50K             1.0                      0.0
4  United-States  <=50K             0.0                      0.0

   race_ASIAN-PAC-ISLANDER  race_BLACK  race_OTHER  race_WHITE
0                      0.0         1.0         0.0         0.0
1                      0.0         0.0         0.0         1.0
2                      0.0         0.0         0.0         1.0
3                      0.0         1.0         0.0         0.0
4                      0.0         0.0         0.0         1.0
```

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	capital-gain	capital-loss	hours-per-week	native-country	incom
0	25.0	Private	226802	11th	7.0	Never-married	Machine-op-inspct	Own-child	0	0	40	United-States	<=50I
1	38.0	Private	89814	HS-grad	9.0	Married-civ-spouse	Farming-fishing	Husband	0	0	50	United-States	<=50I
2	28.0	Local-gov	336951	Assoc-acdm	12.0	Married-civ-spouse	Protective-serv	Husband	0	0	40	United-States	>50I
3	44.0	Private	160323	Some-college	10.0	Married-civ-spouse	Machine-op-inspct	Husband	7688	0	40	United-States	>50I
4	18.0	?	103497	Some-college	10.0	Never-married	?	Own-child	0	0	30	United-States	<=50I
...	...	...	...	...	...	...	...	...	...	...	...	...	.
48837	27.0	Private	257302	Assoc-acdm	12.0	Married-civ-spouse	Tech-support	Wife	0	0	38	United-States	<=50I
48838	40.0	Private	154374	HS-grad	9.0	Married-civ-spouse	Machine-op-inspct	Husband	0	0	40	United-States	>50I
48839	58.0	Private	151910	HS-grad	9.0	Widowed	Adm-clerical	Unmarried	0	0	40	United-States	<=50I
48840	22.0	Private	201490	HS-grad	9.0	Never-married	Adm-clerical	Own-child	0	0	20	United-States	<=50I

Next steps: Generate code with df_encoded | View recommended plots | New interactive sheet

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

# Initialize the MinMaxScaler
normalizer = MinMaxScaler()

# Apply the MinMaxScaler to the 'AGE' column only
df_encoded[['hours-per-week']] = normalizer.fit_transform(df_encoded[['hours-per-week']])

# Display the first few rows to verify the transformation
print(df_encoded.head())
df_encoded
```

```
4  18.0          ?  103497  Some-college            10.0       Never-married

         occupation relationship  capital-gain  capital-loss  hours-per-week  \
0  Machine-op-inspct    Own-child             0             0        0.397959
1    Farming-fishing      Husband             0             0        0.500000
2    Protective-serv      Husband             0             0        0.397959
3  Machine-op-inspct      Husband          7688             0        0.397959
4                  ?    Own-child             0             0        0.295918

  native-country income  gender_Encoded  race_AMER-INDIAN-ESKIMO  \
0  United-States  <=50K             1.0                      0.0
1  United-States  <=50K             1.0                      0.0
2  United-States   >50K             1.0                      0.0
3  United-States   >50K             1.0                      0.0
4  United-States  <=50K             0.0                      0.0

   race_ASIAN-PAC-ISLANDER  race_BLACK  race_OTHER  race_WHITE
0                      0.0         1.0         0.0         0.0
1                      0.0         0.0         0.0         1.0
2                      0.0         0.0         0.0         1.0
3                      0.0         1.0         0.0         0.0
4                      0.0         0.0         0.0         1.0
```

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	capital-gain	capital-loss	hours-per-week	native-country	inco
0	25.0	Private	226802	11th	7.0	Never-married	Machine-op-inspct	Own-child	0	0	0.397959	United-States	<=5
1	38.0	Private	89814	HS-grad	9.0	Married-civ-spouse	Farming-fishing	Husband	0	0	0.500000	United-States	<=5
2	28.0	Local-gov	336951	Assoc-acdm	12.0	Married-civ-spouse	Protective-serv	Husband	0	0	0.397959	United-States	>5
3	44.0	Private	160323	Some-college	10.0	Married-civ-spouse	Machine-op-inspct	Husband	7688	0	0.397959	United-States	>5
4	18.0	?	103497	Some-college	10.0	Never-married	?	Own-child	0	0	0.295918	United-States	<=5
...	...	...	...	...	...	...	...	...	...	...	...	...	
48837	27.0	Private	257302	Assoc-acdm	12.0	Married-civ-spouse	Tech-support	Wife	0	0	0.377551	United-States	<=5
48838	40.0	Private	154374	HS-grad	9.0	Married-civ-spouse	Machine-op-inspct	Husband	0	0	0.397959	United-States	>5
48839	58.0	Private	151910	HS-grad	9.0	Widowed	Adm-clerical	Unmarried	0	0	0.397959	United-States	<=5
48840	22.0	Private	201490	HS-grad	9.0	Never-married	Adm-clerical	Own-child	0	0	0.193878	United-States	<=5
48841	52.0	Self-emp-inc	287927	HS-grad	9.0	Married-civ-spouse	Exec-managerial	Wife	15024	0	0.397959	United-States	>5

Next steps: Generate code with df_encoded | View recommended plots | New interactive sheet

```
# Standardization (mean=0, variance=1)
#Pros: Works well for normally distributed data; suitable for many models.
#Cons: Sensitive to outliers.
scaler = StandardScaler()
df_encoded[['age']] = scaler.fit_transform(df_encoded[['age']])
df_encoded.head()
```

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	capital-gain	capital-loss	hours-per-week	native-country	incom
0	-0.995125	Private	226802	11th	7.0	Never-married	Machine-op-inspct	Own-child	0	0	0.397959	United-States	<=50I
1	-0.046907	Private	89814	HS-grad	9.0	Married-civ-spouse	Farming-fishing	Husband	0	0	0.500000	United-States	<=50I
2	-0.776305	Local-gov	336951	Assoc-acdm	12.0	Married-civ-spouse	Protective-serv	Husband	0	0	0.397959	United-States	>50I
3	0.390732	Private	160323	Some-college	10.0	Married-civ-spouse	Machine-op-inspct	Husband	7688	0	0.397959	United-States	>50I

Next steps: Generate code with df_encoded View recommended plots New interactive sheet

```
#Removing Outliers
# Outlier Detection and Treatment using IQR
#Pros: Simple and effective for mild outliers.
#Cons: May overly reduce variation if there are many extreme outliers.
df_encoded_copy1=df_encoded
df_encoded_copy2=df_encoded
df_encoded_copy3=df_encoded

Q1 = df_encoded_copy1['fnlwgt'].quantile(0.25)
Q3 = df_encoded_copy1['fnlwgt'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
df_encoded_copy1['fnlwgt'] = np.where(df_encoded_copy1['fnlwgt'] > upper_bound, upper_bound,
                    np.where(df_encoded_copy1['fnlwgt'] < lower_bound, lower_bound, df_encoded_copy1['fnlwgt']))

print(df_encoded_copy1.head())
```

```
        age  workclass    fnlwgt     education  educational-num  \
0 -0.995125    Private  226802.0          11th              7.0
1 -0.046907    Private   89814.0       HS-grad              9.0
2 -0.776305  Local-gov  336951.0    Assoc-acdm             12.0
3  0.390732    Private  160323.0  Some-college             10.0
4 -1.505704          ?  103497.0  Some-college             10.0

       marital-status         occupation relationship  capital-gain  \
0       Never-married  Machine-op-inspct    Own-child             0
1  Married-civ-spouse    Farming-fishing      Husband             0
2  Married-civ-spouse    Protective-serv      Husband             0
3  Married-civ-spouse  Machine-op-inspct      Husband          7688
4       Never-married                  ?    Own-child             0

   capital-loss  hours-per-week native-country income  gender_Encoded  \
0             0        0.397959  United-States  <=50K             1.0
1             0        0.500000  United-States  <=50K             1.0
2             0        0.397959  United-States   >50K             1.0
3             0        0.397959  United-States   >50K             1.0
4             0        0.295918  United-States  <=50K             0.0

   race_AMER-INDIAN-ESKIMO  race_ASIAN-PAC-ISLANDER  race_BLACK  race_OTHER  \
0                      0.0                      0.0         1.0         0.0
1                      0.0                      0.0         0.0         0.0
2                      0.0                      0.0         0.0         0.0
3                      0.0                      0.0         1.0         0.0
4                      0.0                      0.0         0.0         0.0

   race_WHITE
0         0.0
1         1.0
2         1.0
3         0.0
4         1.0
```

```
#Removing Outliers
# Z-score method
#Pros: Good for normally distributed data.
#Cons: Not suitable for non-normal data; may miss outliers in skewed distributions.

df_encoded_copy2['fnlwgt_zscore'] = stats.zscore(df_encoded_copy2['fnlwgt'])
```

```
df_encoded_copy2['fnlwgt'] = np.where(df_encoded_copy2['fnlwgt_zscore'].abs() > 3, np.nan, df_encoded_copy2['fnlwgt'])  # Replace outliers wi
print(df_encoded_copy2.head())
```

```
        age  workclass    fnlwgt     education  educational-num  \
0 -0.995125    Private  226802.0          11th              7.0
1 -0.046907    Private   89814.0       HS-grad              9.0
2 -0.776305  Local-gov  336951.0    Assoc-acdm             12.0
3  0.390732    Private  160323.0  Some-college             10.0
4 -1.505704          ?  103497.0  Some-college             10.0

       marital-status         occupation relationship  capital-gain  \
0       Never-married  Machine-op-inspct    Own-child             0
1  Married-civ-spouse    Farming-fishing      Husband             0
2  Married-civ-spouse    Protective-serv      Husband             0
3  Married-civ-spouse  Machine-op-inspct      Husband          7688
4       Never-married                  ?    Own-child             0

   capital-loss  hours-per-week native-country income  gender_Encoded  \
0             0        0.397959  United-States  <=50K             1.0
1             0        0.500000  United-States  <=50K             1.0
2             0        0.397959  United-States   >50K             1.0
3             0        0.397959  United-States   >50K             1.0
4             0        0.295918  United-States  <=50K             0.0

   race_AMER-INDIAN-ESKIMO  race_ASIAN-PAC-ISLANDER  race_BLACK  race_OTHER  \
0                      0.0                      0.0         1.0         0.0
1                      0.0                      0.0         0.0         0.0
2                      0.0                      0.0         0.0         0.0
3                      0.0                      0.0         1.0         0.0
4                      0.0                      0.0         0.0         0.0

   race_WHITE  fnlwgt_zscore
0         0.0       0.419934
1         1.0      -1.017089
2         1.0       1.575412
3         0.0      -0.277440
4         1.0      -0.873553
```

```
#Removing Outliers
# Median replacement for outliers
#Pros: Keeps distribution shape intact, useful when capping isn't feasible.
#Cons: May distort data if outliers represent real phenomena.
df_encoded_copy3['fnlwgt_zscore'] = stats.zscore(df_encoded_copy3['fnlwgt'])
```