Develop the python Shell Script to do the basic data cleanup on child labor and child marriage data.xlsx

a) Check duplicates and missing data
b) Eliminate mismatches
c) Cleans line breaks, spaces, and special characters.

PROCEDURE:

Step1: Import required libraries like pandas ,numpy and xlrd
Step2: Read data using read.csv method in pandas
Step3: Find basic info about data
Step4: check duplicates and missing data.
Step5: Eliminate mismatches
Step6: Clean line breaks, spaces and special characters
Step7: Detect outliers
Step 8: Normalize Casing

Code:

```
import pandas as pd
cl= pd.read_csv('National Child Labour.csv')
cl
cl.shape
cm= pd.read_csv('Child Marriage.csv')
cm
cm.shape
cl.isnull().sum()
cm.isnull().sum()
#to remove missing values from data
cl=cl.dropna()
cl.duplicated().sum()
cm.duplicated().sum()
cm['Male Married by 18'].duplicated().sum()
cm.loc[cm.duplicated()]
cl = cl.drop_duplicates()
cm=cm.drop_duplicates()
cl.shape
cl=cl.rename(columns={'Country Name':'Country'})
cl.columns
cm.isnull().sum()cm['Female Married by 18']=cm['Female Married by 18'].fillna(value=0)
cm.tail(30)
cm['Reference year']=cm['Reference year'].fillna(method='ffill')
cm.tail(30)
cm['Male Married by 18']=cm['Male Married by 18'].fillna(method='bfill')
cm.tail(30)
cm=cm.dropna()
cl[cl['Country Code']=='AND']
cl=cl.drop(cl[cl['Country Code']=='AND'].index)
new=pd.concat([cl,cm],axis=1)
cl.head()
new2=new.merge(cl,indicator=True,how='outer')
new2
#to remove line breaks columnswise
cl['Country'] = cl['Country'].str.replace('\n', ' ')
#to remove line breaks row-wise
```

```
cl.replace('\n','',regex=True)
#to display whitespaces
cm['Data source']=cm['Data source'].str.replace(' ','')
#to remove special characters from columns
cm['Female Married by 15'] = cm['Female Married by 15'].str.replace(r'\W', '',
regex=True)cm.head(20)
cm[cm['Female Married by 15'] == '1']
```

XLRD PROGRAM:

```
!pip install xlrd
import xlrd
import re
book = xlrd.open_workbook("SOWC 2014 Stat Tables_Table 9.xls")
sheet = book.sheet_by_name("Table 9 ")
data = {}
for i in range(14, sheet.nrows):
# Start at 14th row, because that is where the countries begin
row = sheet.row_values(i)
country = row[1]
data[country] = {
'child_labor': {
'total': [row[4], row[5]],
'male': [row[6], row[7]],
'female': [row[8], row[9]], },
'child_marriage': {
'married_by_15': [row[10], row[11]],
'married_by_18': [row[12], row[13]],
}
}
if country == "Zimbabwe":
break
import pprint
pprint.pprint(data)

# Extract data from the sheet
data = []
for row_idx in range(14, sheet.nrows): # Skip the header row
row = sheet.row_values(row_idx)
data.append(row)
# Check for duplicates
duplicates = []
seen = set()
for row in data:
row_str = ','.join(map(str, row)) if row_str in seen:
duplicates.append(row)
else:
seen.add(row_str)
if duplicates:
print("Duplicates found:")
for duplicate_row in duplicates:
print(duplicate_row)
else:
```

```
print("No duplicates found.")
# Check for missing data
missing_data_rows = []
for row_idx, row in enumerate(data):
if any(cell == '' or cell is None for cell in row):
missing_data_rows.append(row_idx)
if missing_data_rows:
print("Missing data found in rows:")
for row_idx in missing_data_rows:
print(f"Row {row_idx + 2}") # Adding 2 to account for header and 0-based indexing
else:
print("No missing data found.")
# Clean line breaks, spaces, and special characters
def clean_text(text):
cleaned_text = re.sub(r'\s+', ' ', str(text)) # Replace multiple spaces and line breaks with a single space
cleaned_text = re.sub(r'[^\w\s]', '', str(cleaned_text)) # Remove special characters
return cleaned_text.strip()
for row_idx in range(len(data)):
for col_idx in range(len(data[row_idx])):
data[row_idx][col_idx] = clean_text(data[row_idx][col_idx])
# Since xlrd doesn't support writing changes to Excel files, you can only display cleaned data
print("Data cleanup complete. Cleaned data:")
for row in data:
print(row)
```

Explain about line chart, bar chart, pie chart and develop a python script to draw both the charts

ChatGPT

ChatGPT

Certainly! Let's discuss each type of chart (line chart, bar chart, and pie chart) and then develop a Python script using the `matplotlib` library to draw these charts.

Line Chart:

A line chart is used to display data points over a continuous interval or time span. It is particularly useful for showing trends and patterns in data. Each data point is connected with a line, providing a visual representation of how the data changes.

Bar Chart:

A bar chart represents data using rectangular bars of lengths proportional to the values they represent. Bar charts are effective for comparing values across different categories or groups. They are commonly used to show the distribution of data or to compare discrete categories.

Pie Chart:

A pie chart is a circular statistical graphic that is divided into slices to illustrate numerical proportions. Each slice represents a proportion of the whole, and the size of each slice is proportional to the corresponding data value. Pie charts are useful for displaying the composition of a whole.

Python Script to Draw Line Chart, Bar Chart, and Pie Chart:

This script uses the `matplotlib` library to draw a line chart, bar chart, and pie chart for sample data. You can customize the `categories` and `values` lists with your own data.

```
import matplotlib.pyplot as plt

import numpy as np



# Sample data for demonstration

categories = ['Category A', 'Category B', 'Category C', 'Category D']

values = [25, 40, 30, 20]



# Line Chart

plt.figure(figsize=(10, 4))

plt.subplot(1, 3, 1)

plt.plot(categories, values, marker='o', color='b')

plt.title('Line Chart')

plt.xlabel('Categories')

plt.ylabel('Values')



# Bar Chart

plt.subplot(1, 3, 2)

plt.bar(categories, values, color='g')

plt.title('Bar Chart')

plt.xlabel('Categories')
plt.ylabel('Values')


# Pie Chart

plt.subplot(1, 3, 3)

plt.pie(values, labels=categories, autopct='%1.1f%%', startangle=90, colors=['orange', 'purple', 'pink', 'brown'])

plt.title('Pie Chart')


# Adjust layout and display

plt.tight_layout()

plt.show()
```

Make sure to install `matplotlib` if you haven't already:

bashCopy code

Run this script in a Python environment, and it will display all three charts side by side. Adjust the data and styling based on your specific requirements.

Explain about types of data with an example?

ChatGPT

ChatGPT

Data can be classified into different types based on its nature, characteristics, and the kind of values it represents. The commonly recognized types of data are:

1. **Nominal Data:**
    - Nominal data represents categories or labels.
    - It doesn't have a natural order or ranking.
    - Examples: Colors (e.g., red, blue, green), Gender (e.g., male, female), Marital Status (e.g., single, married, divorced).
2. **Ordinal Data:**
    - Ordinal data has a natural order or ranking, but the intervals between values are not meaningful.
    - It indicates the relative position or order of categories.
    - Examples: Education level (e.g., high school, bachelor's, master's, Ph.D.), Customer satisfaction ratings (e.g., poor, fair, good, excellent).
3. **Interval Data:**

    - Interval data has a meaningful order, and the intervals between values are equal, but it lacks a true zero point.
    - Arithmetic operations such as addition and subtraction make sense, but multiplication and division do not.
    - Examples: Temperature in Celsius or Fahrenheit (e.g., 20°C, 30°C), IQ scores, Likert scale with no true zero.
4. **Ratio Data:**
    - Ratio data has a meaningful order, equal intervals, and a true zero point.
    - All arithmetic operations are valid, including multiplication and division.
    - Examples: Height, weight, income, age, distance.
5. **Discrete Data:**
    - Discrete data consists of separate, distinct values with no intermediate values.
    - It often represents countable items.
    - Examples: Number of cars in a parking lot, Number of students in a classroom, Number of defects in a product.
6. **Continuous Data:**
    - Continuous data can take any value within a given range.
    - It has an infinite number of possible values within a specified interval.
    - Examples: Height, Weight, Temperature, Time.
7. **Categorical Data:**
    - Categorical data represents categories or labels and can be either nominal or ordinal.
    - It is qualitative rather than quantitative.
    - Examples: Gender, Colors, Marital Status.
8. **Numerical Data:**
    - Numerical data represents measurable quantities and can be either discrete or continuous.
    - It is quantitative and involves numerical values.
    - Examples: Height, Weight, Income, Temperature.

Understanding the type of data is crucial for selecting appropriate statistical methods, visualizations, and analysis techniques. Different types of data require different approaches for interpretation and modeling in various fields such as statistics, data science, and machine learning.