TP: Data Wrangling with Pandas

Your Name

December 18, 2024

Introduction

In this practical session, we will explore various data wrangling techniques using the pandas library in Python. Each mini-project will focus on a specific task, and solutions will be provided for each.

1 Mini-Project 1: Titanic Survivor Analysis with Visualization

Description

You have a dataset of Titanic passengers. Your task is to create a pandas DataFrame from the data, then calculate the number of survivors and non-survivors by passenger class. Next, visualize this data using a bar chart.

Solution

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the data
url = 'https://raw.githubusercontent.com/chrisalbon/sim_data/master/titanic.csv'
dataframe = pd.read_csv(url)

# Calculate the number of survivors and non-survivors by class
survivors_by_class = dataframe[dataframe['Survived'] == 1].groupby('PClass').size()
non_survivors_by_class = dataframe[dataframe['Survived'] == 0].groupby('PClass').size()
```

```python

# Visualize the data
classes = survivors_by_class.index
survivors_counts = survivors_by_class.values
non_survivors_counts = non_survivors_by_class.values

plt.bar(classes, survivors_counts, label='Survivors')
plt.bar(classes, non_survivors_counts, bottom=survivors_counts, label='Non-Survivors')
plt.xlabel('Class')
plt.ylabel('Number of Passengers')
plt.title('Survivors and Non-Survivors by Class')
plt.legend()
plt.show()
```

2 Mini-Project 2: Handling Missing Values and Age Analysis

Description

You have a DataFrame containing data on Titanic passengers, but some age values are missing. Your task is to replace these missing values with the mean age. Then, analyze the age distribution by creating a histogram.

Solution

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Load the data
url = 'https://raw.githubusercontent.com/chrisalbon/sim_data/master/titanic.csv'
dataframe = pd.read_csv(url)

# Replace missing values with the mean age
mean_age = dataframe['Age'].mean()
dataframe['Age'].fillna(mean_age, inplace=True)

# Display a histogram of the ages
plt.hist(dataframe['Age'], bins=20, color='blue', edgecolor='black')
plt.xlabel('Age')
plt.ylabel('Number of Passengers')
plt.title('Age Distribution of Passengers')
plt.show()
```

3 Mini-Project 3: Renaming Columns and Adding a New Feature

Description

You have a DataFrame with unclear column names. Your task is to rename the columns to make them more descriptive. Then, add a new column that indicates whether a passenger is a child (age $<$ 18) or an adult.

Solution

```python
import pandas as pd

# Load the data
url = 'https://raw.githubusercontent.com/chrisalbon/sim_data/master/titanic.csv'
dataframe = pd.read_csv(url)

# Rename the columns
dataframe.rename(columns={'PClass': 'PassengerClass', 'Sex': 'Gender'}, inplace=True)

# Add a new column to indicate if the passenger is a child or an adult
dataframe['IsChild'] = dataframe['Age'] < 18

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

4 Mini-Project 4: Grouping by Class and Calculating Detailed Statistics

Description

You have a DataFrame containing information about Titanic passengers, grouped by class (1st, 2nd, 3rd). Your task is to calculate the mean, median, minimum, and maximum age for each class. Then, display these statistics in a DataFrame.

Solution

```python
import pandas as pd

# Load the data
url = 'https://raw.githubusercontent.com/chrisalbon/sim_data/master/titanic.csv'
dataframe = pd.read_csv(url)

# Group by class and calculate detailed statistics
stats_by_class = dataframe.groupby('PClass')['Age'].agg(['mean', 'median', 'min', 'max'])

print(stats_by_class)
```

5 Mini-Project 5: Merging DataFrames and Sales Analysis

Description

You have two DataFrames, one containing information about employees and the other containing their sales. Your task is to merge these two DataFrames on a common column (e.g., employee ID). Then, calculate the total sales by employee and display the top 5 sellers.

Solution

```python
import pandas as pd

# Create the DataFrames
employee_data = {'employee_id': ['1', '2', '3', '4'], 'name': ['Amy Jones', 'Allen Keys', 'Alice Bees', 'Tim Horton']}
sales_data = {'employee_id': ['3', '4', '5', '6'], 'total_sales': [23456, 2512, 2345, 1455]}

dataframe_employees = pd.DataFrame(employee_data)
dataframe_sales = pd.DataFrame(sales_data)

# Merge the DataFrames
merged_dataframe = pd.merge(dataframe_employees, dataframe_sales, on='employee_id', how='inner')

# Calculate the total sales by employee
```

```python
total_sales_by_employee = merged_dataframe.groupby('name')['total_sales'].sum().reset_index()

# Display the top 5 sellers
top_5_sellers = total_sales_by_employee.sort_values(by='total_sales', ascending=False).head(5)

print(top_5_sellers)
```