FPT Software

FRESHER ACADEMY

AI Fresher

Assignment 4

Pandas

Viet Tran

VERSION 1.0

TABLE OF CONTENTS

EXERCISE 01 (SHORT)

Create a DataFrame from a list.

EXERCISE 02 (SHORT)

Create a DataFrame from a Numpy array.

EXERCISE 03 (SHORT)

Create a DataFrame with multiple columns from several lists. Store the DataFrame into a csv file with headers.

EXERCISE 04 (SHORT)

Create a DataFrame with multiple columns from several lists. Store the DataFrame into a csv file without headers.

EXERCISE 05 (SHORT)

Read the csv file from the output of Exercise 03 into a DataFrame with a defined index column. Check the information of the DataFrame with info() method. Use the loc() method to retrieve rows by using index values.

EXERCISE 06 (SHORT)

Read the csv file from the output of Exercise 04 into a DataFrame with a defined index column. Check the information of the DataFrame with info() method. Use the loc() method to retrieve rows by using index values.

EXERCISE 07 (SHORT)

Read the sales.csv file with the 'Date' column as the index of the DataFrame. Use info(), head(), tail(), loc() and iloc() to check the information of the DataFrame.

EXERCISE 08 (SHORT)

Read the sales.csv file with the 'Date' column as the index of the DataFrame. Create a new column Remaining so Units + Remaining = 100. Add another column SaleMan as 'David Beckham'. Store the DataFrame with headers into a csv file named sales_08.

EXERCISE 09 (MEDIUM)

Create a DataFrame from 3 different lists names, scoreA and score B having the same lengths. Store the DataFrame in to a csv file.

Read the csv file into another DataFrame then:

- Find people having the highest score A.
- Find people having the lowest score B.
- Sort people in descending order of score A and score B respectively.

EXERCISE 10 (MEDIUM)

From 5 names David, Peter, Ethan, Jason and Micheal, create a list of random 1000 names. Moreover, create another list of random 1000 numbers of births from 0 to 100. Create a DataFrame of Names and Births from these two lists. Store the DataFrame into a csv file without header and no index.

EXERCISE 11 (MEDIUM)

Create a DataFrame with columns:

- id: from 1 to 100
- Exam: 5
- Year: 2018
- Scores are read from scores.csv

From this DataFrame:

- Count numbers of ids/examinees by scores.
- Add a new column Group which Group1 for ids from 1 to 50 and Group 2 for ids from 51 to 100
    - Calculate mean of scores each group
    - Calculate sum of scores each group

EXERCISE 12 (MEDIUM)

Read file euro_2012.csv into a DataFrame and check information by using info().

From the DataFrame:

- Filter 5 columns Team, Goals, Yellow Cards, Red Cards and Players Used.

- Sort and display teams in the ascending order of 'Player Used'.
- Sort and display teams in the ascending orders of 'Yellow Cards' and 'Red Cards' respectively
- Calculate the mean of 'Yellow Cards' of all teams in Euro 2012
- Filter teams having more than 9 goals and sort them in the descending order of 'Passes'
- Create a new column 'Shooting Accuracy' as ['Shots on target'] / (['Shots on target'] + ['Shots off target']) * 100
- Sort and display teams, goals and shooting accuracy in the descending order of 'Shooting Accuracy'
- Store the DataFrame into euro_12.csv with headers

EXERCISE 13 (MEDIUM)

Read file users.txt into a DataFrame and check information by using info().

From the DataFrame:

- Calculate mean of ages for each occupation
- Create a new column 'gender_n' to convert categorical genders to numeric genders (M:1, F:0)
- Calculate male ratio (%) of each occupation
- Sort male ratios in descending order
- Find and display min and max ages of each occupation
- Calculate and display mean of ages of male and female for each occupation
- Count and display occupations by gender (male and female)

EXERCISE 14 (MEDIUM)

Read file OfficeSupplies.csv into a DataFrame (A) with the column 'OrderDate' as the index.

- Check information by using info(), head() and tail().
- Create the DataFrame (B): total sold units of each representative in each region.
- Calculate maximum of total units in each region to store in a DataFrame (C)
- Drop index of C.
- Merge B and C into a DataFrame (D).

- Transform the DataFrame A and store it into a csv file similar to sales_min_max.csv

EXERCISE 15 (MEDIUM)

Read mart_list_of_orders.csv and parse date columns into a DataFrame with 'Order ID' as the index:

- Create a new column 'Ship Days' as the shipping duration (use 'Ship Date' and 'Order Date').
- Filter to find fast orders with 'Ship Days' less than 2 days.
- Make a report to display numbers of orders each month (For instance, there were 257 orders in January).

Read mart_order_details.csv into another DataFrame:

- Join two DataFrame on 'Order ID' to create a DataFrame having full order information.
- Use the info() method to check.

EXERCISE 16 (MEDIUM)

Analyze data in the file 'US Cities Population.csv' and make at least 3 reports of population.

EXERCISE 17 (LONG)

Write a program to generate data similar to the file sale_data.csv. Analyze data columns then use Numpy and Pandas to create random data.

EXERCISE 18 (LONG)

Analyze data in the file 'UK Bank Customers.csv' and make at least 5 reports of UK banking customers.

EXERCISE 19 (LONG)

Analyze data in the file 'Unemployment Statistics.xlsx' and make at least 5 reports of unemployment.

EXERCISE 20 (LONG)

Analyze data in the file 'Startups.xlsx' and make at least 5 reports of startups.