Лабораторна робота 3
Кореляційний аналіз у Python

Мета: продемонструвати свої навички кореляційного аналізу даних, використовуючи заданий набір даних та вказані інструменти

Передумови / сценарій

У цій лабораторній роботі ви дізнаєтесь, як використовувати Python для обчислення кореляції. У частині 1 ви налаштуєте набір даних. У частині 2 ви дізнаєтеся, як визначити, чи змінні в даному наборі даних є корельованими. У частині 3 ви будете використовувати Python для обчислення кореляції між двома наборами змінних. Нарешті, у частині 4 ви здійсните візуалізацію результатів дослідження.

Завдання до лабораторної роботи знаходиться за посиланням:

https://static-course-assets.s3.amazonaws.com/IoTFBDA201/en/course/files/3.1.5.5%20Lab%20-%20Correlation%20Analysis%20in%20Python.html

Необхідні ресурси

- 1 ПК з доступом до Інтернету
- Бібліотеки Python: pandas, numpy, matplotlib, seaborn
- Файли даних: brainsize.txt

Частина 1: Набір даних

Використайте набір даних, який містить зразок 40 студентів у Southwestern university. Було взято чотири субтести (словниковий запас, схожість, дизайн блоків та доповнення картинок) переглянутої Wechsler Adult Intelligence Scale (1981).

https://en.wikipedia.org/wiki/Wechsler_Adult_Intelligence_Scale

Дослідники використовували магнітно-резонансну томографію (МРТ) для визначення розміру мозку досліджуваних. Також включена інформація про стать та розмір тіла (зріст та вага). З міркувань конфіденційності дослідники утримували ваги двох предметів та зростання одного предмета.

До набору даних застосовано дві прості модифікації:

1. Заміна знаків питання, що використовуються для представлення утриманих точок даних, описаних вище, рядком 'NaN'. Заміна була здійснена, оскільки Pandas неправильно обробляє знаки питання.
2. Заміна усіх символів табуляції комами, перетворивши набір даних у набір даних CSV.

Підготовлений набір даних зберігається як **brainsize.txt.**

Крок 1: Завантаження набору даних із файлу.

Перш ніж набір даних можна використовувати, його потрібно завантажити в пам'ять.

У наведеному нижче коді перший рядок імпортує модулі pandas та визначається pd як дескриптор, який посилається на модуль.

Другий рядок завантажує CSV-файл набору даних у змінну з назвою brainFile.

Третій рядок read_csv(), це метод pandas для перетворення CSV набору даних, збережений в brainFile в dataframe. Потім фрейм даних зберігається у змінній brainFrame.

Запустіть клітинку коду нижче, щоб виконати описані функції.

```
# Code cell 1
import pandas as pd
brainFile = './Data/brainsize.txt'
brainFrame = pd.read_csv(brainFile)
```

Крок 2: Перевірка дата фрейму.

Щоб переконатися, що фрейм даних правильно завантажений і створений, скористайтеся методом head(). Метод pandas head() відображає перші п'ять записів кадру даних.

```
# Code cell 2
brainFrame.head()
```

Частина 2: Діаграми розсіювання та корельовані змінні

Крок 1: Метод describe().

Модуль pandas включає метод describe(), який виконує однакові загальні обчислення щодо даного набору даних. На додаток до загальних результатів, включаючи кількість, середнє, стандартне відхилення, мінімальне та максимальне, describe() також є чудовим способом швидкого тестування достовірності значень у фреймі даних.

Виконайте код нижче, щоб виводити результати методу describe(), обчислені по відношенню до дата фрейму brainFrame.

```
# Code cell 3
brainFrame.describe()
```

Крок 2: Графіки діаграм розсіювання

Графіки діаграм розсіювання важливі при роботі з кореляціями, оскільки вони дозволяють швидко візуально перевірити природу взаємозв'язку між змінними.

У цій лабораторній роботі використовується коефіцієнт кореляції Пірсона, який чутливий лише до лінійного співвідношення між двома змінними (навести формули для обчислення кореляції Пірсона). Існують інші більш надійні методи кореляції, але в цій лабораторній роботі вони не розглядаються.

a. Завантажте необхідні модулі.

Перш ніж побудувати графіки, необхідно імпортувати кілька модулів, а саме numpy та matplotlib. Запустіть код нижче, щоб завантажити ці модулі.

```
# Code cell 4
import numpy as np
import matplotlib.pyplot as plt
```

b. Відокремте дані.

Щоб результати не спотворювались через різницю в чоловічому та жіночому тілах, дата фрейм розділений на два кадри даних: один, що містить усі записи чоловіків, а інший - лише жіночі випадки.

Запуск коду нижче створює два нових кадри даних, menDf і womenDf, кожен з яких містить відповідні записи.

```
# Code cell 5
menDf = brainFrame[(brainFrame.Gender == 'Male')]
womenDf = brainFrame[(brainFrame.Gender == 'Female')]
```

c. Побудуйте графіки.

Оскільки набір даних включає три різні показники інтелекту (PIQ, FSIQ та VIQ), перший рядок коду нижче використовує метод mean() для обчислення середнього значення між трьома та збереження результату у змінній menMeanSmarts. Зверніть увагу, що перший рядок також стосується menDf, відфільтрованого кадру даних, що містить лише чоловічі записи.

Другий рядок використовує matplotlib метод scatter() для створення діаграми розсіювання між menMeanSmarts змінною та MRI_Count атрибутом. MRI_Count у цьому наборі даних можна вважати мірою фізичного розміру мозку суб'єктів.

Третій рядок відображає графік.

Четвертий рядок використовується для відображення графіку.

```
# Code cell 6
menMeanSmarts = menDf[["PIQ", "FSIQ", "VIQ"]].mean(axis=1)
plt.scatter(menMeanSmarts, menDf["MRI_Count"])
plt.show()
%matplotlib inline
```

Наведений нижче код створює графік розбіжності для відфільтрованого кадру даних лише для жінок.

```
# Code cell 7
# Graph the women-only filtered dataframe
#womenMeanSmarts = ?
#plt.scatter(?, ?)

plt.show()
%matplotlib inline
```

Частина 3: Обчислення кореляції в Python

Крок 1: Обчисліть співвідношення між brainFrame.

Метод pandas corr() забезпечує простий спосіб обчислення кореляції щодо кадру даних. Викликавши метод для фрейму даних, можна отримати кореляцію між усіма змінними одночасно.

```
# Code cell 8
brainFrame.corr(method='pearson')
```

Зверніть увагу на діагональ зліва направо у таблиці кореляцій, сформованій вище. Чому діагональ заповнена 1s? Це випадковість? Поясніть.

Не дивлячись на таблицю кореляцій вище, зауважте, що значення відображаються дзеркально; значення нижче 1 діагоналі мають дзеркальний аналог вище 1 діагоналі. Це випадковість? Поясніть.

Використовуючи метод corr(), можна розрахувати кореляцію змінних, що містяться в кадрі даних лише для жінок:

```
# Code cell 9
womenDf.corr(method='pearson')
```

І те саме можна зробити для кадру даних, призначеного лише для чоловіків:

```
# Code cell 10
# Use corr() for the male-only dataframe with the pearson method
#?.corr(?)
```

Частина 4: Візуалізація

Крок 1: Встановіть Seaborn.

Для спрощення візуалізації кореляцій даних можна використовувати графіки теплових карт. На основі кольорових квадратів графіки теплових карт можуть допомогти швидко виявити кореляційні зв’язки.

За допомогою модуля Python seaborn дуже легко побудувати графіки теплових карт.

Спочатку запустіть код нижче, щоб завантажити та встановити модуль seaborn.

```
# Code cell 11
!pip install seaborn
```

Крок 2: Побудуйте графік кореляційної теплової карти.

Тепер, коли кадри даних готові, можна побудувати теплові карти.

Рядок 1: Створює таблицю кореляції на основі women NoGenderDf кадру даних та зберігає її в wcorr.

Рядок 2: Використовує seaborn метод heatmap() для формування та побудови графіку теплової карти. Зверніть увагу', що heatmap() приймає wcorr в якості параметра.

Рядок 3: Використовуйте для експорту та збереження сформованої теплової карти як зображення PNG. Поки рядок 3 не активний (перед ним є #символ коментаря, що змушує інтерпретатор ігнорувати його), він зберігався з інформаційною метою.

```
# Code cell 12
import seaborn as sns

wcorr = womenDf.corr()
sns.heatmap(wcorr)
#plt.savefig('attribute_correlations.png', tight_layout=True)
```

Подібним чином, наведений нижче код створює і складає теплову карту для кадру даних лише для чоловіків.

```
# Code cell 14
mcorr = menDf.corr()
sns.heatmap(mcorr)
#plt.savefig('attribute_correlations.png', tight_layout=True)
```

Багато пар змінних мають кореляцію, близьку до нуля. Що це означає?

Навіщо розділяти стать?

Які змінні мають сильнішу кореляцію з розміром мозку (MRI_Count)? Це очікується? Поясніть.