Лабораторная работа - Корреляционный анализ в Python

Цели

Часть 1: Набор данных

Часть 2: Диаграммы рассеяния и корреляционные переменные

Часть 3: Расчёт корреляции с Python

Часть 4: Визуализация

История / Cценарий

Корреляция является важной статистической зависимостью, которая может указывать, связаны ли линейные значения переменных.

В этой лабораторной работе вы узнаете, как использовать Python для расчета корреляции. В Части 1 Вы подготовите набор данных. В Части 2 Вы узнаете, как определить, являются ли переменные в данном наборе данных взаимозависимыми. Наконец, в Части 3, Вы будете использовать Python для вычисления корреляции между двумя наборами переменных.

Необходимо:

Библиотеки Python: pandas, numpy, matplotlib, seaborn

Файл данных: brainsize.txt

Часть 1: Набор данных

Вы будете использовать набор данных, который содержит выборку 40 студентов правшей с вводного курса по Психологии из университета Southwestern. Испытуемые прошли четыре субтеста (Vocabulary, Similarities, Block Design, и Picture Completion) для расчёта шкалы интеллекта взрослых по Векслеру. Исследователи использовали магнитно-резонансную томографию (МРТ) для определения размера мозга у субъектов. Также включены сведения о гендерных показателях и размере тела (рост и вес). Исследователи скрыли вес двух испытуемых и высоту одного из них по причинам конфиденциальности. Для набора данных были применены две простые модификации:

Заменены вопросительные знаки, используемые для представления скрытых данных, описанных выше, строкой «NaN». Замена была выполнена, потому что Pandas не обрабатывает вопросительные знаки правильно.

Заменены все символы табуляции запятыми, преобразуя набор данных в набор данных CSV.

Готовый набор данных сохранён как brainsize.txt.

Шаг 1: Загрузка набора данных из файла.

До того, как набор данных можно использовать, он должен быть загружен в память.

В приведенном ниже коде первая строка импортирует модули pandas и определяет pd как дескриптор, который ссылается на один из модулей.

Вторая строка загружает CSV-файл набора данных в переменную с именем brainFile.

Третья строка использует метод read\_csv() из библиотеки pandas, чтобы преобразовать набор данных CSV, хранящийся в brainFile в кадр данных. Затем кадр данных хранится в переменной brainFrame.

Запустите ячейку ниже, чтобы выполнить описанные функции.

import pandas as pd

brainFrame = pd.read\_csv(?, delimiter='\t')

Ответ:

import pandas as pd

brainFrame = pd.read\_csv('brain\_data.csv', delimiter='\t')

Шаг 2. Проверка кадра данных.

Чтобы убедиться, что кадр данных правильно загружен и создан, используйте метод head(). Метод head() отображает первые пять записей в кадре данных.

brainFrame.?()

Ответ:

brainFrame.head()

Метод head() принимает параметр - число строк, отобразите первые 10 строк таблицы

brainFrame.head(?)

Ответ:

brainFrame.head(10)

Также Dataframe имеет метод tail(), который возвращает последние 5 строк кадра данных. Аналогично head() метод tail() принимает аргументом число строк, которые он вернет. Отобразите последние 8 строк таблицы.

Ответ:

print(df.tail(8))

Часть 2: Диаграммы рассеяния и корреляционные переменные

Шаг 1: метод describe() в библиотеке pandas.

Библиотека pandas включает в себя метод describe(), который выполняет одни и те же общие вычисления на предложенном наборе данных. Помимо базовых операций таких, как подсчёт, расчёт среднего значения, расчёт среднеквадратического отклонения, расчёт минимума, и максимума, describe() также отличный способ быстро проверить достоверность значений в кадре данных.

Ниже используйте функцию describe, для вывода статистики по данным

brainFrame.?()

Ответ:

brainFrame.describe()

Шаг 2: Диаграммы рассеяния

Диаграммы рассеяния важны при работе с корреляциями, поскольку они позволяют быстро визуально проверить природу связи между переменными. В этой лабораторной работе используется коэффициент корреляции Пирсона, который чувствителен только к линейной зависимости между двумя переменными. Также существуют другие более надежные методы корреляции, но они выходят за рамки этой лабораторной работы.

a. Загрузите необходимые модули.

Прежде чем строить графики, необходимо импортировать несколько модулей, а именно numpy и matplotlib. Запустите ячейку ниже, чтобы загрузить эти модули.

import numpy as np

import matplotlib.pyplot as plt

b. Разделите данные.

Чтобы гарантировать, что результаты не будут искажены из-за различий в мужских и женских телах, кадр данных разбит на два: один содержит все мужские записи, а другой - только женские экземпляры.

Создайте два новых кадра данных: menDf и womenDf, каждый из которых содержит соответствующие записи.

menDf = brainFrame[?]

womenDf = brainFrame[?]

Ответ:

menDf = brainFrame[brainFrame['Gender'] == 'Male']

womenDf = brainFrame[brainFrame['Gender'] == 'Female']

c. Постройте графики.

Поскольку набор данных включает в себя три различных измерения интеллекта (PIQ, FSIQ и VIQ), первая строка ниже использует метод mean() из библиотеки Pandas для вычисления среднего значения между тремя измерениями и сохранения результата в переменной menMeanSmarts. Обратите внимание, что первая строка относится к menDf, отфильтрованному кадру данных, содержащему только мужские записи.

Вторая строка использует метод scatter() библиотеки matplotlib для создания графика диаграммы рассеяния между переменной menMeanSmarts и атрибутом MRI\_Count. MRI\_Count в этом наборе данных можно рассматривать как меру физического размера мозга испытуемых.

Третья строка просто отображает график.

Четвертая строка используется для гарантированного отображения графика в этом блокноте.

# Ячейка для кода № 6

menMeanSmarts = menDf[["PIQ", "FSIQ", "VIQ"]].mean(axis=1)

plt.scatter(menMeanSmarts, menDf["MRI\_Count"])

plt.show()

%matplotlib inline

Ответ:

womenMeanSmarts = womenDf[["PIQ", "FSIQ", "VIQ"]].mean(axis=1)

plt.scatter(womenMeanSmarts, womenDf["MRI\_Count"])

plt.show()

%matplotlib inline

Аналогичным образом создайте график диаграммы рассеяния для кадра данных, содержащему только женские записи.

# Ячейка для кода № 7

# Постройка графика диаграммы рассеяния для кадра данных с женскими записями

womenMeanSmarts = ???

plt.scatter(???)

#

#

Ответ:

womenMeanSmarts = womenDf[["PIQ", "FSIQ", "VIQ"]].mean(axis=1)

plt.scatter(womenMeanSmarts, womenDf["MRI\_Count"])

plt.xlabel('Среднее значение интеллекта')

plt.ylabel('MRI\_Count')

plt.title('Диаграмма рассеяния для женских записей')

plt.show()

%matplotlib inline

Часть 3: Вычисление корреляции с Python

Шаг 1: Рассчитайте корреляцию для brainFrame.

Метод corr() библиотеки pandas обеспечивает простой способ вычисления корреляции для кадра данных. Просто применяя метод на кадре данных, можно получить корреляцию между всеми переменными одновременно.

brainFrame.?(method='pearson')

Ответ:

brainFrame.corr(method='pearson')

Обратите внимание на диагональ слева направо в таблице корреляции, сгенерированной выше. Почему диагональ заполнена значениями 1? Это совпадение? Объясните.

(ответ)

Корреляция переменной с самой собой всегда равна 1, так как переменная полностью коррелирует с самой собой. Это не совпадение, а математическое следствие определения корреляции.

Продолжая смотреть на таблицу корреляции выше, обратите внимание, что значения зеркалируются; значения под диагональю имеют зеркальный аналог над ней. Это совпадение? Объясните.

(ответ)

Нет, зеркальное отражение значений в таблице корреляции не является совпадением, это естественное свойство корреляционной матрицы. Поскольку корреляция между переменной X и Y такая же, как и между переменной Y и X, значения в таблице корреляции зеркалируются относительно главной диагонали.

Используя тот же метод corr(), вычислите корреляцию переменных, содержащихся в кадре с женскими записями:

womenDf.?(method='pearson')

Ответ:

correlation\_matrix = womenDf.corr()

И то же самое можно сделать для переменных, содержащихся в кадре с мужскими записями:

# Используйте corr() для расчёта критерия корреляции Пирсона для кадра данных с мужчинами

Ответ:

correlation\_matrix\_men = menDf.corr()

Часть 4: Визуализация

Шаг 1: Установите Seaborn.

Чтобы упростить визуализацию корреляций данных, можно использовать тепловую карту. На основе цветных квадратов тепловая карта может помочь выявить корреляции с первого взгляда.

Модуль Python с именем seaborn очень упрощает построение тепловых карт.

Сначала запустите ячейку ниже, чтобы загрузить и установить модуль seaborn. (закомментируйте, если библиотека установлена)

# Ячейка для кода № 11

!pip install seaborn

Ответ:

import seaborn as sns

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 8))

sns.heatmap(correlation\_matrix\_men, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Heatmap for Men Data')

plt.show()

Шаг 2: Нарисуйте корреляционную тепловую карту.

Теперь, когда кадры данных готовы, можно отобразить тепловые карты. Далее приведен анализ кода из ячейки ниже:

Строка 1: Создает таблицу корреляции, основанную на фрейме данных womenNoGenderDf и хранит его в wcorr.

Строка 2: Использует метод heatmap() библиотеки seaborn для генерации и построения тепловой карты. Обратите внимание, что heatmap() принимает wcorr, как параметр.

Строка 3: используется для экспорта и сохранения созданной тепловой карты в виде PNG-изображения. Хотя строка 3 не активна (перед ней стоит символ #, заставляя интерпретатор игнорировать ее), она сохранена в информационных целях.

# Ячейка для кода № 12

import seaborn as sns

wcorr = womenDf.corr()

sns.heatmap(wcorr)

#plt.savefig('attribute\_correlations.png', tight\_layout=True)

Аналогичным образом создайте и отобразите тепловую карту для кадра данных с мужчинами.

# Ячейка для кода № 14

mcorr = ???

#

#

Ответ:

mcorr = menDf.corr()

plt.figure(figsize=(12, 10))

sns.heatmap(mcorr, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Heatmap for Men')

plt.show()

У многих пар переменных корреляция близка к нулю. Что это значит?

(ответ)

Это означает, что между этими парами переменных нет явной статистической связи или взаимосвязи. В других словах, изменение одной переменной не влияет на изменение другой переменной.

Зачем делать разделение по полу?

(ответ)

Разделение по полу может быть полезным в некоторых случаях, например, при проведении медицинских исследований, при определении демографических характеристик аудитории или при разработке товаров и услуг, ориентированных на конкретную гендерную группу.

Какие переменные имеют более сильную корреляцию с размером мозга (MRI\_Count)? Это ожидалось? Объясните.

(ответ)

образование, интеллект и физическая активность имеют положительную корреляцию с размером мозга, в то время как возраст, раса и наличие болезней могут иметь отрицательную корреляцию.

Таким образом, переменные, которые представляют физиологические, генетические и образовательные особенности человека, вероятно, будут иметь более сильную корреляцию с размером мозга