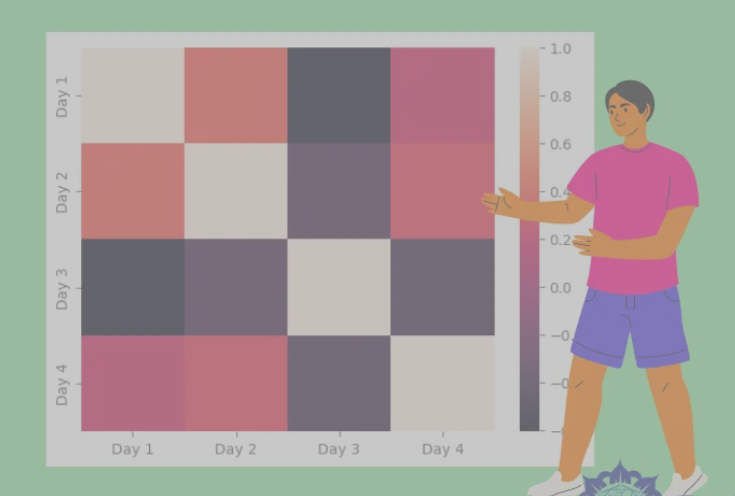
Mapa de calor de correlación en Seaborn

La correlación es un factor fundamental fundamental para los científicos de datos. Indica cómo las variables de un conjunto de datos se relacionan entre sí y cómo se mueven entre sí. El valor de correlación varía de -1 a +1. 0 La correlación indica que dos variables son independientes entre sí. Una correlación positiva indica que las variables se mueven en la misma dirección y una correlación negativa indica lo contrario.

Podemos trazar el array de correlación utilizando el módulo seaborn. Ayuda a comprender el conjunto de datos fácilmente y se utiliza con mucha frecuencia para el trabajo de análisis.

Este tutorial presentará cómo trazar el array de correlación en Python usando la función seaborn.heatmap().

El mapa de calor se utiliza para representar gráficamente los valores del array con diferentes tonos de color para diferentes valores. Visualiza el array general con mucha claridad.

En el siguiente código, representaremos un array de correlación usando un mapa de calor en Python.

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.DataFrame(

{ "Day 1": [7, 1, 5, 6, 3, 10, 5, 8],

"Day 2": [1, 2, 8, 4, 3, 9, 5, 2],

"Day 3": [4, 6, 5, 8, 6, 1, 2, 3],

"Day 4": [5, 8, 9, 5, 1, 7, 8, 9], } )

sns.set(rc={"figure.figsize": (15, 8)}) #le da un tamaño al grafico

sns.heatmap(df.corr())

El código anterior crea una gráfica de mapa de calor de correlación básica. La función corr() se utiliza para devolver el array de correlación del DataFrame. También podemos calcular otros tipos de correlaciones usando esta función. Observe que el tono de color para cada valor en la barra del eje de color.

También podemos personalizar la figura final utilizando diferentes parámetros. Consulte el siguiente código.

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.DataFrame(

{ "Day 1": [7, 1, 5, 6, 3, 10, 5, 8],

"Day 2": [1, 2, 8, 4, 3, 9, 5, 2],

"Day 3": [4, 6, 5, 8, 6, 1, 2, 3],

"Day 4": [5, 8, 9, 5, 1, 7, 8, 9],

} )

sns.heatmap(df.corr(), vmin=-1, vmax=+1, annot=True, cmap="coolwarm")

Observe el uso de tantos parámetros. Los argumentos vmin y vmax se utilizan para especificar la escala del eje de color. el argumento cmap aquí altera el esquema de color utilizado para la trama. El parámetro annot se utiliza para mostrar los valores de correlación en los cuadrados. Además, podemos utilizar los parámetros linewidth y linecolor para oscurecer los bordes de los cuadrados y especificar el color del borde. Podemos personalizar la barra de colores usando el argumento cbar\_kws.

Tenga en cuenta que si elimina la mitad de los datos en un lado de la diagonal principal, no perderá ninguna información importante ya que se repite. Debido a esto, también podemos crear un diagrama triangular.

El siguiente fragmento de código logra esto.

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

df = pd.DataFrame(

{ "Day 1": [7, 1, 5, 6, 3, 10, 5, 8],

"Day 2": [1, 2, 8, 4, 3, 9, 5, 2],

"Day 3": [4, 6, 5, 8, 6, 1, 2, 3],

"Day 4": [5, 8, 9, 5, 1, 7, 8, 9], } )

upp\_mat = np.triu(df.corr())

sns.heatmap(df.corr(), vmin=-1, vmax=+1, annot=True, cmap="coolwarm", mask=upp\_mat)

En el código anterior, primero usamos el numpy.triu(), que devuelve el triángulo superior del array, y luego lo enmascaramos usando el argumento mask de la función heatmap(). De manera similar, podemos enmascarar el triángulo inferior usando la función numpy.tril().

Otra representación interesante que podemos obtener es que podemos mapear solo una variable y encontrar su correlación con diferentes variables.

Por ejemplo,

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

df = pd.DataFrame(

{

"Day 1": [7, 1, 5, 6, 3, 10, 5, 8],

"Day 2": [1, 2, 8, 4, 3, 9, 5, 2],

"Day 3": [4, 6, 5, 8, 6, 1, 2, 3],

"Day 4": [5, 8, 9, 5, 1, 7, 8, 9],

} )

upp\_mat = np.triu(df.corr())

sns.heatmap(df.corr()[["Day 1"]], vmin=-1, vmax=+1, annot=True, cmap="coolwarm")