R de Pearson, cortesía de Sonic Visualiser

Edgar Delgado Vega

24 de setiembre de 2024
Compilado el 7 de agosto de 2025, v.0.0.1

Resumen

En este artículo te suelto un par de versos relajados para entender el coeficiente de correlación de Pearson desde su esencia matemática hasta su implementación en código con TypeScript.

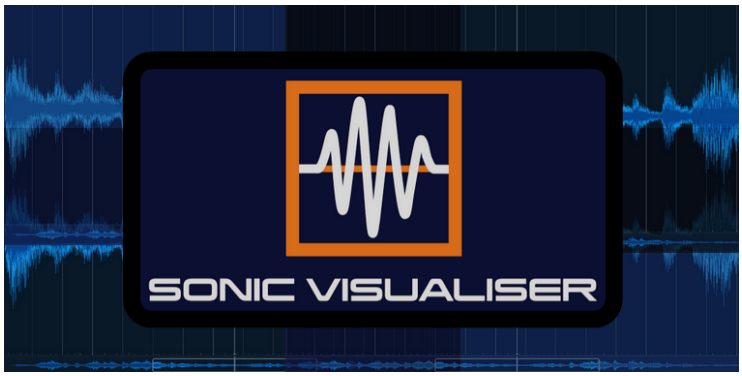

Figura 1: Despertando a Sonic Visualiser.

Índice

1. Introducción

En esta primera aproximación a la estadística quiero compartir algo súper **chévere** que me he encontrado indagando en las **representaciones gráficas** que generamos a partir de análisis de grabaciones con **Sonic Visualiser**, mientras escuchaba el tema de los *Thundercats*.

Lo que vamos a ver es una forma sencilla y relajada de entender un concepto estadístico clave. Después, lo divertido viene con la implementación en **TypeScript**, que es donde le ponemos el *sabor*. Aquí, no obstante, hay fórmulas porque son elegantes (salió rimando).

Entonces, ¿cuál es el norte en esta publicación?

> Queremos conocer la correlación entre dos performances de una misma pieza musical. Es decir, queremos ver más allá de lo evidente, como si tuviésemos la *espada del augurio*.

2. ¿Qué es correlación?

Iniciamos prodigando una definición brumosa, como ciertas mañanas de verano al avistar el mar desde los acantilados de la costa limeña:

Definición 2.1. La correlación nos indica si dos variables (o más) presentan una dependencia lineal y cuán fuerte es el grado de dicha dependencia.

Lo *lineal* debe entenderse como un cambio simultáneo y a la misma velocidad (más adelante volveremos a este punto). Sin embargo, es importante aclarar que la correlación se refiere más a una **asociación** entre variables y no implica ni pretende establecer una relación de **causa y efecto**.

3. Con las formulitas macizas

Las fórmulas matemáticas son sumamente bellas porque abstraen una infinidad de ideas en un par de símbolos. Parte de su belleza radica en su concisión. Seguro estás pensando: "¡Qué tanto bla, bla! Quiero ver ya el contenido."

Vamos al grano, sin más preámbulo:

$$r_{xy} = \frac{\sum_{i=0}^{N-1}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N-1}(x_i - \bar{x})^2 \cdot \sum_{i=0}^{N-1}(y_i - \bar{y})^2}}. \tag{1}$$

La expresión (1) es una de las más recurridas correlaciones: la **r** de Pearson. Volviéndonos personas de cultura, el nombre completo es *coeficiente de producto-momento de Karl Pearson*. Vaya nombre macizo.

Observación 3.1. He adecuado un poco el estilo de la definición para aludir a cómo la escribiremos en el código; es mi alegato.

El coeficiente **r** de Pearson es una medida que normaliza la correlación lineal entre las variables **x** e **y** en el intervalo $[-1, 1]$. Seguro estás preguntándote:

1. ¿Qué son las variables **x** e **y**?
2. ¿Qué es normalización?
3. ¿Qué son las demás cosas?
4. ¿Por qué sigo leyendo esto? :/

En palabras llanas, las variables son secuencias de números y la normalización es el proceso que asegura que el coeficiente solo produzca valores comprendidos entre -1 y $+1$. Más formalmente,

$$-1 \leq r \leq 1, \quad r \in \mathbb{R}.$$

Nota, además, que no he mencionado las variables independiente y dependiente. En el contexto de una asociación, este floro no sería el más adecuado.

Observación 3.2. ¿Qué significan los valores que puede tomar r? Si $r = 1$, hay una correlación positiva perfecta: al aumentar una variable, la otra también aumenta de manera proporcional. Si $r = -1$, existe una correlación negativa perfecta: cuando una variable aumenta, la otra disminuye proporcionalmente. Por otro lado, si $r = 0$, no hay correlación lineal entre las variables, aunque podría existir algún otro tipo de relación no lineal.

Observación 3.3. Hay más condiciones para aplicar la correlación de Pearson (**r**), como que los datos presenten distribución normal y la ausencia de valores atípicos. Por el momento, nos centraremos únicamente en descifrar la fórmula.

Para dar respuesta a la tercera pregunta, nos centraremos de lleno en la implementación en `TypeScript`.

4. El codiguillo explayado

Para esta sección, te recomiendo tener a la mano la fórmula de Pearson más arriba descrita.

Observación 4.1. No buscaremos la optimización en primera instancia.

4.1. Algo de notación

Las variables `x` y `y` son secuencias de números. El primer *rack* rectangular que elegiremos para ordenar las cosas serán los **arreglos**. Los arreglos son *estructuras de datos lineales, simples pero poderosas*. Son un caballo ganador para cientos y cientos de batallas.

En TypeScript, los escribimos con sus tipos bien tranquilos así:

```
const x: number[] = [4, 7, 10, 2];
const y: number[] = [3, 5, 8, 4];
```

Continuando con la explicación de la fórmula, seguro observas una N, que representa el número de muestras o longitud de los arreglos. Desde un punto de vista programador, tendríamos que $N = 4$ para x e y.

¿Qué es la letra i en la fórmula? Más fácil decir que es el **índice** del arreglo y comienza en 0.

Quedamos entonces armando las sumatorias:

$$\sum_{i=0}^{N-1} x_i.$$

Parece difícil, pero en realidad solo tendrás que sumar todos los elementos del arreglo x, desde el índice i hasta el último índice.

4.2. Tu libreta de notas sin correr

Finalmente, el ingrediente faltante es el **promedio**; el que dejaba roja o azul tu libreta. Siendo más refinados, se le llama *media aritmética*, y la definimos como la suma de todos los elementos entre el número de elementos que tiene el arreglo (N). Se denota como sigue:

$$\bar{x} = \frac{\sum_{i=0}^{N-1} x_i}{N}. \tag{2}$$

¿Cómo funcionan los índices? Tomamos el valor de $x = [4, 7, 10, 3]$ y sumamos los elementos accediendo mediante índices:

$$x[0] + x[1] + x[2] + x[3]$$
$$4 + 7 + 10 + 3 = 24.$$

Contamos el número de elementos en el arreglo x:

$$N = 4.$$

Por último dividimos

$$\bar{x} = \frac{24}{4} = 6.$$

En TypeScript, la media (2) luce como:

```
function mean(variable: number[]): number {
  const N = variable.length;
  let sum = 0;
  for (let i = 0; i < N; i++) {
    sum += variable[i];
  }
  return sum / N;
}
```

Nada nuevo bajo el sol de noviembre en Lima. Sumas todas tus calificaciones y te esperas lo mejor.

4.3. Lo que va arriba en la `r` de Pearson

Vamos ahora al ingrediente numerador de la `r` en (1). Aunque es posible escribirlo en un estilo más funcional con `.reduce()`, preferí el toque clásico con el bucle `for`, puesto que le pone más semejanza a la notación matemática.

```
function pearsonProductMoment(xVariable: number[],
    yVariable: number[]): number {
  const meanXVariable = mean(xVariable);
  const meanYVariable = mean(yVariable);
  const N = xVariable.length;
  let sumProductDeviations = 0;
  for (let i = 0; i < N; i++) {
    sumProductDeviations += (xVariable[i] - meanXVariable)
        * (yVariable[i] - meanYVariable);
  }
  return sumProductDeviations;
}
```

Observa que le hemos puesto el nombre de **producto de momentos de Pearson** para darle más promesa al asunto; así también se le llama. Lo que he hecho es únicamente sumar el producto de las desviaciones de ambas variables respecto a sus medias.

La `sumProductDeviations` es la partecita de la fórmula que retorna:

$$\sum_{i=0}^{N-1}(x_i - \bar{x})(y_i - \bar{y}). \tag{3}$$

Explayando los cálculos tenemos lo siguiente para el índice 0:

$$(x[0] - \bar{x})(y[0] - \bar{y})$$
$$(4-6) \times (3-5) = (-2) \times (-2) = 4$$

Nuestro acumulador `sumProductDeviations` se inicia en cero y ahora suma 4.

Para el índice 1 repetimos lo mismo:

$$(7-6) \times (5-5) = 1 \times 0 = 0,$$

y así sucesivamente hasta el último índice.

4.4. El piso de abajo en la r de Pearson:

Solo nos faltaría el último pasajero, el denominador en la fracción de Pearson (1).

$$\sqrt{\sum_{i=0}^{N-1}(x_i - \bar{x})^2 \cdot \sum_{i=0}^{N-1}(y_i - \bar{y})^2} \tag{4}$$

También es conocido como producto de desviaciones estándar o, para simplificar, el factor normalizador (el que mencionamos más arriba).

```
function standardDeviationsProduct(xVariable: number[],
    yVariable: number[]): number {
  const meanXVariable = mean(xVariable)
  const meanYVariable = mean(yVariable)
  const N = xVariable.length
  let sumX = 0;
  let sumY = 0
```

```
  for (let i = 0; i < N; i++) {
      sumX += (xVariable[i] - meanXVariable) ** 2;
      sumY += (yVariable[i] - meanYVariable) ** 2;
  }
}
```

¿Pero qué me estás haciendo aquí? Sumamos el cuadrado de las desviaciones de cada variable por separado y luego las multiplicamos.

El `sumX` que está en el bucle tiene esta pinta en la fórmula:

$$\sum_{i=0}^{N-1}(x_i - \bar{x})^2.$$

Igual se hace para el valor de `sumY`,

$$\sum_{i=0}^{N-1}(y_i - \bar{y})^2.$$

Por otra parte, ¿recuerdas que preguntabas para qué jugar con las propiedades de raíces y potencias? Estás viendo que el término de normalización está elevado a un exponente fraccionario 1/2. Ni me lo recuerdes.

4.5. Llamando a los términos prohibidos

Finalmente, solo queda llamar a las funciones para armar la fracción con numerador (3) y denominador (4) tan esperada:

```
function rPearsonCoef(x: number[], y: number[]){
    return pearsonProductMoment(x, y) /
        standardDeviationsProduct(x, y);
}
```

- **Observación**: ¿Se puede refactorizar el código para evitar repetir el cálculo de la media aritmética con muestras enormes?

Estás en lo correcto, citemos al griego:

> La duda es el principio de la sabiduría. (Aristóteles)

¿Qué tal si optimizas el código, cambias a otro estilo, lo haces más semántico utilizando `Math.sqrt()`? Diviértete un par de minutos.

5. Una alternativa más yuca

Seguro habrás leído por otro lugar que la correlación de Pearson se expresa así:

$$r_{XY} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}. \tag{5}$$

Es decir, estamos hablando de la popular covarianza y las desviaciones estándar. Lo que significan está aquí para salvarte:

$$\operatorname{cov}(X,Y) = \frac{1}{N-1}\sum_{i=0}^{N-1}(X_i-\bar{X})(Y_i-\bar{Y}),$$

$$\sigma_X = \sqrt{\frac{1}{N-1}\sum_{i=0}^{N-1}(X_i-\bar{X})^2}, \quad \sigma_Y = \sqrt{\frac{1}{N-1}\sum_{i=0}^{N-1}(Y_i-\bar{Y})^2}.$$

¿Y cómo llego a la fórmula 1 del inicio? Hay que manipular un poquito las cosas y ya.

Demostración. Antes que nada, omitimos la escritura de los índices del arreglo y tenemos

$$r_{XY} = \frac{\frac{1}{N-1}\sum(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\frac{1}{N-1}\sum(X_i-\bar{X})^2}\cdot\sqrt{\frac{1}{N-1}\sum(Y_i-\bar{Y})^2}}.$$

Ahora extraemos el factor $\frac{1}{N-1}$ de las raíces:

$$\sqrt{\frac{1}{N-1}\sum(X_i-\bar{X})^2} = \frac{1}{\sqrt{N-1}}\sqrt{\sum(X_i-\bar{X})^2},$$

$$\sqrt{\frac{1}{N-1}\sum(Y_i-\bar{Y})^2} = \frac{1}{\sqrt{N-1}}\sqrt{\sum(Y_i-\bar{Y})^2}.$$

Por lo tanto,

$$\sigma_X\sigma_Y = \frac{1}{N-1}\sqrt{\sum(X_i-\bar{X})^2}\cdot\sqrt{\sum(Y_i-\bar{Y})^2}.$$

Toca sustituir en r:

$$r_{XY} = \frac{\frac{1}{N-1}\sum(X_i-\bar{X})(Y_i-\bar{Y})}{\frac{1}{N-1}\sqrt{\sum(X_i-\bar{X})^2}\cdot\sqrt{\sum(Y_i-\bar{Y})^2}}$$

Finalmente, al cancelar el factor común $\frac{1}{N-1}$, regresamos a la fórmula 1:

$$r_{XY} = \frac{\sum(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum(X_i-\bar{X})^2}\cdot\sqrt{\sum(Y_i-\bar{Y})^2}}.$$

Con esto queda el rollo demostrado. □

6. ¿Qué viene ahora?

Lo que viene es un *misterio*: algún gráfico loco o el tutorial de cómo dibujar una maceta para una secuoya.

Y hablando de árboles, hasta la próxima, ¡mándale saludos al recordado Hebaristo de Valdelomar!