Correlation, Bivariate, and Regression Analysis

Rafiq Islam

2024-12-18

Table of contents

## Introduction

Correlation and regression are two fundamental concepts in statistic, often used to study relationships between variables. While correlation measures the strength and direction of a linear relationship between two variables, regression goes further by modeling the relationship to predict or explain one variable based on another. This blog explores the mathematical underpinnings of both concepts, illustrating their significance in data analysis.

## Correlation

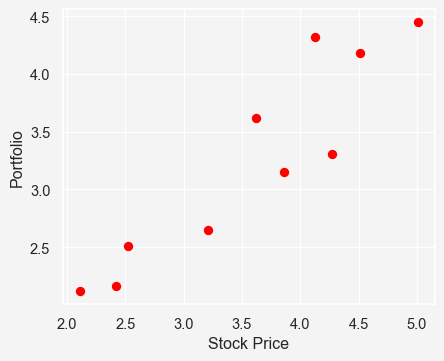
To better explain, we will use the following hypothetical stock data of 10 companies with stock price and their corresponding proportion in the portfolio.

import pandas as pd  
  
df = pd.DataFrame({  
 'Stock': ['Apple', 'Citi', 'MS', 'WF', 'GS', 'Google', 'Amazon', 'Tesla', 'Toyota', 'SPY'],  
 'StockPrice': [2.11, 2.42, 2.52, 3.21, 3.62, 3.86, 4.13, 4.27, 4.51, 5.01],   
 'Portfolio': [2.12, 2.16, 2.51, 2.65, 3.62, 3.15, 4.32, 3.31, 4.18, 4.45]  
})  
  
df.set\_index('Stock', inplace=True)  
  
df.T

| Stock | Apple | Citi | MS | WF | GS | Google | Amazon | Tesla | Toyota | SPY |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| StockPrice | 2.11 | 2.42 | 2.52 | 3.21 | 3.62 | 3.86 | 4.13 | 4.27 | 4.51 | 5.01 |
| Portfolio | 2.12 | 2.16 | 2.51 | 2.65 | 3.62 | 3.15 | 4.32 | 3.31 | 4.18 | 4.45 |

The scatterplot of the data looks like this

from mywebstyle import plot\_style  
plot\_style('#f4f4f4')  
import matplotlib.pyplot as plt  
plt.scatter(df.StockPrice, df.Portfolio, color='red')  
plt.xlabel('Stock Price')  
plt.ylabel('Portfolio')  
plt.show()



We can see from the graph that there appears to be a linear relationship between the and values in this case. To find the relationship mathematically we define the followings

Similarly,

The sample correlation coefficient is then given as

You may have seen a different formula to calculate this quantity which often looks a bit different

The sample correlation coefficient, , is an estimator of the population correlation coefficient, , in the same way as is an estimator of or is an estimator of . Now the question is what does this values mean?

| Value | Meaning |
| --- | --- |
|  | The two variables move together in the same direction in a perfect linear relationship. |
|  | The two variables tend to move together in the same direction but there is NOT a direct relationship. |
|  | The two variables can move in either direction and show no linear relationship. |
|  | The two variables tend to move together in opposite directions but there is not a direct relationship. |
|  | The two variables move together in opposite directions in a perfect linear relationship. |

Let’s calculate the correlation of our stock data.

import math  
x = df.StockPrice.values  
y = df.Portfolio.values  
  
n = len(x)  
  
x\_sum, y\_sum =0,0  
s\_xx, s\_yy, s\_xy = 0,0,0  
for i in range(n):  
 x\_sum += x[i]  
 s\_xx += x[i]\*\*2  
 y\_sum += y[i]  
 s\_yy += y[i]\*\*2  
 s\_xy += x[i]\*y[i]   
  
s\_xx = s\_xx - (x\_sum)\*\*2/n  
s\_yy = s\_yy - (y\_sum)\*\*2/n  
s\_xy = s\_xy - (x\_sum \* y\_sum)/n  
  
r = s\_xy/math.sqrt(s\_xx \* s\_yy)  
  
# Print with formatted labels  
print(f"Sₓₓ: {s\_xx:.2f}")  
print(f"Sᵧᵧ: {s\_yy:.2f}")  
print(f"Sₓᵧ: {s\_xy:.2f}")  
print(' ')  
print(f"r : {r:.2f}")

Sₓₓ: 8.53  
Sᵧᵧ: 6.97  
Sₓᵧ: 7.13  
   
r : 0.92

## Bivariate Analysis

The joint probability density function for and in the bivariate normal distribution is given by:

When , the denominator in the PDF becomes zero, which might appear problematic. However, what happens in this case is that the joint distribution degenerates into a **one-dimensional structure** (a line) rather than being a two-dimensional probability density.

To see why, consider the quadratic term inside the exponential:

When , this quadratic expression simplifies, as shown next.

Start with the simplified when :

This is a **perfect square** because the “cross term” cancels out all independent variability of and when .

For the quadratic term to have any non-zero probability density (since it appears in the exponent of the PDF), it must be equal to zero:

Rearranging this equation:

Multiply through by :

Thus:

This is the equation of a straight line in the -plane. The slope of the line is , and the line passes through the point . When , all the joint probability mass collapses onto this line, meaning and are perfectly linearly dependent.

// Import necessary libraries  
Plot = require('@observablehq/plot')  
d3 = require('d3@7')  
  
// Define the bivariate normal PDF function  
function bivariateNormalPDF(x, y, muX, muY, sigmaX, sigmaY, rho) {  
 const z =  
 ((x - muX) \*\* 2) / sigmaX \*\* 2 -  
 (2 \* rho \* (x - muX) \* (y - muY)) / (sigmaX \* sigmaY) +  
 ((y - muY) \*\* 2) / sigmaY \*\* 2;  
 const denominator = 2 \* Math.PI \* sigmaX \* sigmaY \* Math.sqrt(1 - rho \*\* 2);  
 return Math.exp(-z / (2 \* (1 - rho \*\* 2))) / denominator;  
}  
  
// Parameters  
const muX = 0,  
 muY = 0,  
 sigmaX = 1,  
 sigmaY = 1;  
  
// Create a slider for rho  
viewof rho = Inputs.range([-0.99, 0.99], { step: 0.01, value: 0, label: 'Correlation (ρ)' })  
  
// Generate grid data  
const x = d3.range(-3, 3.1, 0.1);  
const y = d3.range(-3, 3.1, 0.1);  
const grid = x.flatMap((xi) => y.map((yi) => ({ x: xi, y: yi, z: bivariateNormalPDF(xi, yi, muX, muY, sigmaX, sigmaY, rho) })));  
  
// Create the contour plot  
Plot.plot({  
 marks: [  
 Plot.contour(grid, {  
 x: 'x',  
 y: 'y',  
 z: 'z',  
 stroke: 'steelblue',  
 strokeWidth: 1,  
 thresholds: 10,  
 }),  
 ],  
 x: {  
 label: 'X',  
 },  
 y: {  
 label: 'Y',  
 },  
 color: {  
 legend: true,  
 label: 'Density',  
 },  
 width: 600,  
 height: 600,  
})