Correlation, Bivariate, and Regression Analysis

Rafiq Islam

2024-12-18

Table of contents

## Introduction

Correlation and regression are two fundamental concepts in statistics, 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 Analysis

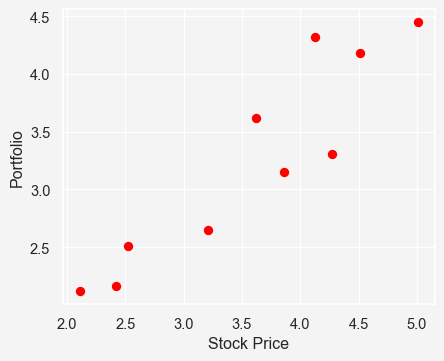
To better explain, we will use the following hypothetical stock data of 10 companies with stock price and their corresponding proportion in the portfolio.

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"Sum x: {x\_sum:.2f}")  
print(f"Sum y: {y\_sum:.2f}")  
print(f"Sₓₓ: {s\_xx:.2f}")  
print(f"Sᵧᵧ: {s\_yy:.2f}")  
print(f"Sₓᵧ: {s\_xy:.2f}")  
print(' ')  
print(f"r : {r:.2f}")

Sum x: 35.66  
Sum y: 32.47  
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:

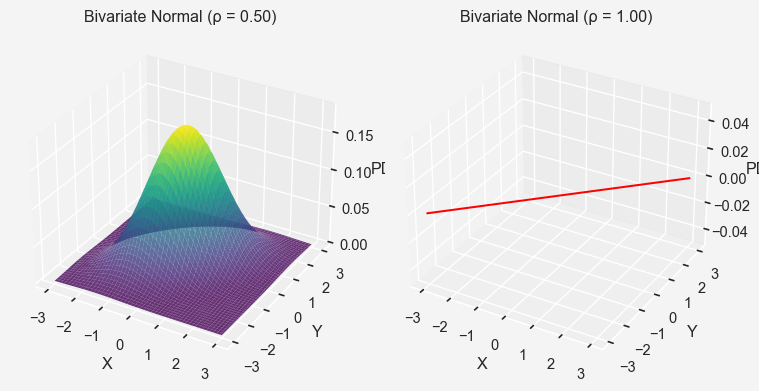
Rearranging this equation:

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 numpy as np  
  
from mpl\_toolkits.mplot3d import Axes3D  
  
# Define the bivariate normal PDF  
def bivariate\_normal\_pdf(x, y, mu\_x, mu\_y, sigma\_x, sigma\_y, rho):  
 z = (  
 ((x - mu\_x) \*\* 2) / sigma\_x\*\*2  
 - 2 \* rho \* (x - mu\_x) \* (y - mu\_y) / (sigma\_x \* sigma\_y)  
 + ((y - mu\_y) \*\* 2) / sigma\_y\*\*2  
 )  
 denominator = 2 \* np.pi \* sigma\_x \* sigma\_y \* np.sqrt(1 - rho\*\*2)  
 return np.exp(-z / (2 \* (1 - rho\*\*2))) / denominator  
  
# Parameters  
x = np.linspace(-3, 3, 100)  
y = np.linspace(-3, 3, 100)  
X, Y = np.meshgrid(x, y)  
  
# Function to plot the bivariate normal distribution and a line for rho = 1 or -1  
def plot\_bivariate\_and\_line\_side\_by\_side(rho1, rho2):  
 fig = plt.figure(figsize=(8, 4))  
  
 # Plot for the first rho  
 ax1 = fig.add\_subplot(121, projection='3d')  
 if abs(rho1) == 1:  
 # Degenerate case: Straight line  
 line\_x = np.linspace(-3, 3, 100)  
 line\_y = line\_x # Since rho = 1 implies y = x (perfect correlation)  
 ax1.plot(line\_x, line\_y, np.zeros\_like(line\_x), label=f'Degenerate Line (ρ = {rho1})', color='red')  
 else:  
 # General bivariate normal distribution  
 Z = bivariate\_normal\_pdf(X, Y, 0, 0, 1, 1, rho1)  
 ax1.plot\_surface(X, Y, Z, cmap='viridis', edgecolor='none', alpha=0.8)  
  
 ax1.set\_title(f'Bivariate Normal (ρ = {rho1:.2f})')  
 ax1.set\_xlabel('X')  
 ax1.set\_ylabel('Y')  
 ax1.set\_zlabel('PDF')  
  
 # Plot for the second rho  
 ax2 = fig.add\_subplot(122, projection='3d')  
 if abs(rho2) == 1:  
 # Degenerate case: Straight line  
 line\_x = np.linspace(-3, 3, 100)  
 line\_y = line\_x # Since rho = 1 implies y = x (perfect correlation)  
 ax2.plot(line\_x, line\_y, np.zeros\_like(line\_x), label=f'Degenerate Line (ρ = {rho2})', color='red')  
 else:  
 # General bivariate normal distribution  
 Z = bivariate\_normal\_pdf(X, Y, 0, 0, 1, 1, rho2)  
 ax2.plot\_surface(X, Y, Z, cmap='viridis', edgecolor='none', alpha=0.8)  
  
 ax2.set\_title(f'Bivariate Normal (ρ = {rho2:.2f})')  
 ax2.set\_xlabel('X')  
 ax2.set\_ylabel('Y')  
 ax2.set\_zlabel('PDF')  
  
 plt.tight\_layout()  
 plt.show()  
  
# Plot examples side by side  
plot\_bivariate\_and\_line\_side\_by\_side(0.5, 1) # Example with rho = 0.5 and rho = 1



### Statistic

Under the null hypothesis, where has a distribution with degree of freedom.

### Fisher’s Transformation of

If , then has approximately a normal distribution with mean and standard deviation .

For our stock data:

**Null Hypothesis :** There is no association between stock prices and the portfolio values, i.e.,   
**Alternative Hypothesis :** There is some association between the stock price and portfolio values, i.e.,

If is true, then the test statistic has a distribution. The observed value is much greater than the critical value of at level which is .

So, we reject the null hypothesis at the level and conclude that there is a very strong evidence that .

Alternatively, if we want to use the Fisher’s test:

If is true, then the test statistic has a distribution.

The observed value of this statistic is , which corresponds to a value of on the distribution. This is much greater than , the upper point of the standard normal distribution.

So, we reject at the level and conclude that there is very strong evidence that ie that there is a positive linear correlation between the stock price and portfolio value.

## Regression Analysis

Given a set of points for a simple linear regression of the form

$$
Y\_i = \alpha +\beta x\_i + \epsilon\_i; \hspace{4mm} i=1,2,\cdots,n
$$

with and .

### Model Fitting

We can estimate the parameters from the method of least squares but that’s not the goal in this case. Fitting the model involves finding and and the estimating the variance .

where, and

is the observed value of a statistic whose sampling distribution has the following properties

$$
\mathbb{E}[\hat{B}]=\beta, \hspace{4mm} var[\hat{B}]=\frac{\sigma^2}{S\_{xx}}
$$

And the estimate of the error variance

### Goodness of fit

To better understand the goodness of fit of the model for the data at hand, we can study the total variation in the responses, as given by

Let’s see how:

In the case that the data are “close” to a line ( high- a strong linear relationship) the model fits well, the fitted responses (the values on the fitted line) are close to the observed responses, and so is relatively high with relatively low.

In the case that the data are not “close” to a line ( low - a weak linear relationship) the model does not fit so well, the fitted responses are not so close to the observed responses, and so is relatively low and relatively high.

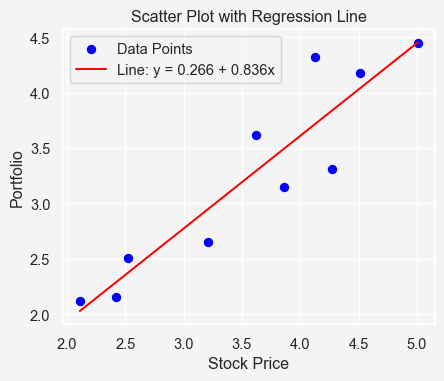
The proportion of the total variability of the responses “explained” by a model is called the coefficient of determination, denoted .

which takes value between 0 to 1, inclusive. The higher , the better fitting.

For our data, we have:

Therefore, the fitted line would be . Now we see the other metrics

# Parameters for the line  
alpha = 0.266   
beta = 0.836   
  
# Line values  
line\_x = np.linspace(min(df.StockPrice), max(df.StockPrice), 100)   
line\_y = alpha + beta \* line\_x   
  
# Plot  
plt.scatter(df.StockPrice, df.Portfolio, color='blue', label='Data Points')  
plt.plot(line\_x, line\_y, color='red', label=f'Line: y = {alpha} + {beta}x')  
  
# Labels and title  
plt.xlabel('Stock Price')  
plt.ylabel('Portfolio')  
plt.title('Scatter Plot with Regression Line')  
plt.legend()  
plt.show()



### Inference on

We can rewrite , as

Now we recall that is the random variable that has as its realization. Therefore, . We also recall that . Putting these together we obtain,

Now the fact that s are uncorrelated. Therefore, and we have . Therefore,

Since and so

and the observed variance has the property

Since and are independent, it follows that

In other words:

Now the big question is what’s the use of this mathematical jargon that we have learned so far? Let’s use our regression problem on stock data to explain.

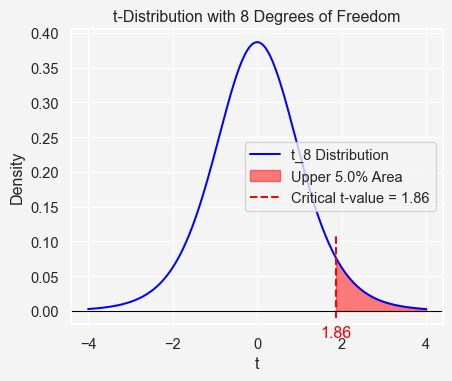
**, there is no linear relationship**  
vs  
**, there is a linear relationship**

Based on our data we have and , and . Therefore, under , the test statistic

But the observed value of this statistic

which is way higher than the critical value at significance level.

from scipy.stats import t  
  
# Parameters  
df = 8 # Degrees of freedom  
alpha = 0.05 # Upper tail probability  
t\_critical = t.ppf(1 - alpha, df) # Critical t-value at the 95th percentile  
  
# Generate x values for the t-distribution  
x = np.linspace(-4, 4, 500)  
y = t.pdf(x, df)  
  
# Plot the t-distribution  
plt.plot(x, y, label=f't\_{df} Distribution', color='blue')  
plt.fill\_between(x, y, where=(x >= t\_critical), color='red', alpha=0.5, label=f'Upper {alpha\*100}% Area')  
  
# Annotate the critical t-value on the x-axis  
plt.axvline(t\_critical, ymin=0.02, ymax=0.30,color='red', linestyle='--', label=f'Critical t-value = {t\_critical:.2f}')  
plt.text(t\_critical, -0.02, f'{t\_critical:.2f}', color='red', ha='center', va='top')  
  
# Add a horizontal line at y = 0  
plt.axhline(0, color='black', linestyle='-', linewidth=0.8)  
  
# Labels, title, and legend  
plt.title(f"t-Distribution with {df} Degrees of Freedom")  
plt.xlabel("t")  
plt.ylabel("Density")  
plt.legend()  
  
# Adjust plot limits  
  
  
# Show plot  
plt.show()



So, we reject the null hypothesis at the level and conclude that there is a very strong evidence that , i.e., the portfolio value is increasing over stock price.

**Alternatively,** let’s put our analysis in a different approach. We claim that

**, there is a linear relationship**  
vs

In this case,

Therefore, the confidence interval for is

The two-sided confidence interval contains the value , so the two-sided test conducted at level results in being accepted.

### Mean Response and Individual Response

#### Mean Response

If is the expected (mean) response for a value of the predictor variable, that is , then is an unbiased estimator given by

and the variance of the estimator is given by

Therefore,

#### Individual Response

The actual estimate of an individual response

However, the uncertainty associated with this estimator, as indicated by its variance, is higher compared to the mean estimator because it relies on the value of an individual response rather than the more stable mean. To account for the additional variability of an individual response relative to the mean, an extra term, , must be included in the variance expression for the estimator of a mean response.

Thus,

Let’s put this two idea through our example. If we want to find a confidence interval or the expected portfolio value on stock price of say, 360. In that case,

and

So, the CI

That is for a stock price of , the expected portfolio value would be in the range of

Similarly, the CI for the predicted actual portfolio value

or

### Model Accuracy

The residual from the fit at is the estimated error which is defined by

Scatter plots of residuals versus the explanatory variable (or the fitted response values) are particularly insightful. A lack of random scatter in the residuals, such as the presence of a discernible pattern, indicates potential shortcomings in the model.

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]  
})  
x = df.StockPrice.values  
y = df.Portfolio.values   
  
y\_hat = [0.266+0.836\*i for i in x]  
plt.scatter(x, y-y\_hat)  
plt.axhline(0)  
plt.ylabel('Residuals')  
plt.xlabel('Stock Price')  
plt.title('Scatter plot of the residuals from the fitted line')  
plt.show()



In this plot, we can see that the residuals tend to increase as increases, indicates that the error variance is not bounded, but increasing with . So, the model is not the best one. A transformation of the responses may stabilize the error variance. In certain case, for some growth models, the appropriate model is that the expected response is related to the exploratory variable through an exponential relationship, i.e.,

x = df.StockPrice.values  
y = np.log(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"Sum x: {x\_sum:.2f}")  
print(f"Sum y: {y\_sum:.2f}")  
print(f"Sₓₓ: {s\_xx:.2f}")  
print(f"Sᵧᵧ: {s\_yy:.2f}")  
print(f"Sₓᵧ: {s\_xy:.2f}")  
print(' ')  
print(f"r : {r:.2f}")

Sum x: 35.66  
Sum y: 11.43  
Sₓₓ: 8.53  
Sᵧᵧ: 0.70  
Sₓᵧ: 2.29  
   
r : 0.94

Now we have:

import numpy as np  
z\_hat = [np.log(0.1873)+0.268\*i for i in x]  
z = np.log(y)  
plt.scatter(x, z-z\_hat)  
plt.axhline(np.mean(z-z\_hat))  
plt.ylabel('Residuals')  
plt.xlabel('Stock Price')  
plt.title('Scatter plot of the residuals from the fitted line')  
plt.show()



Now the residuals look good, that is no special pattern or increasing the error variance.

Thanks for reading.

## References

1. **Montgomery, D. C., & Runger, G. C.** (2014). *Applied Statistics and Probability for Engineers*. Wiley.
2. **Casella, G., & Berger, R. L.** (2002). *Statistical Inference*. Duxbury.
3. **Cohen, J., Cohen, P., West, S. G., & Aiken, L. S.** (2003). *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. Routledge.
4. **Seber, G. A. F., & Lee, A. J.** (2003). *Linear Regression Analysis*. Wiley.
5. **Neter, J., Kutner, M. H., Nachtsheim, C. J., & Wasserman, W.** (1996). *Applied Linear Statistical Models*. Irwin.
6. **Hastie, T., Tibshirani, R., & Friedman, J.** (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
7. **Weisberg, S.** (2005). *Applied Linear Regression*. Wiley.
8. **Bivariate Normal Distribution Explanation**:
   * Rice, J. A. (2006). *Mathematical Statistics and Data Analysis*. Thomson Brooks/Cole.
   * A detailed exploration of the bivariate normal distribution and its properties.
9. **Fisher’s Transformation of Correlation Coefficients**:
   * Fisher, R. A. (1921). *On the probable error of a coefficient of correlation*. Metron, 1, 3-32.
   * The foundational paper describing Fisher’s transformation and its use in hypothesis testing.

**Share on**

**You may also like**