

# How can bivariate analysis be performed in Python, and what are some examples of its application?

Authored by  
**stats writer**

July 1, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can bivariate analysis be performed in Python, and what are some examples of its application?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165432>

Bivariate analysis is a statistical technique used to explore the relationship between two variables. In Python, bivariate analysis can be performed using various libraries such as Pandas, NumPy, and Matplotlib. These libraries provide functions and methods to calculate correlation, create scatter plots, and perform regression analysis.

Some examples of bivariate analysis in Python include studying the correlation between a person's age and their income level, analyzing the relationship between temperature and ice cream sales, and examining the association between education level and job satisfaction. Bivariate analysis in Python allows for visualizing and understanding the relationship between two variables, making it a useful tool for data analysis and decision making in various fields such as business, healthcare, and social sciences.

## **Perform Bivariate Analysis in Python (With Examples)**

**The term bivariate analysis refers to the analysis of two variables. You can remember this because the prefix "bi" means "two."**

**The purpose of bivariate analysis is to understand the relationship between two variables**

**There are three common ways to perform bivariate analysis:**

**1. Scatterplots**

**2. Correlation Coefficients**

**3. Simple Linear Regression**

The following example shows how to perform each of these types of bivariate analysis in Python using the following pandas DataFrame that contains information about two variables: (1) Hours spent studying and (2) Exam score received by 20 different students:

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'hours': ,  
'score': })
```

```
#view first five rows of DataFrame
```

```
df.head()
```

```
hours score
```

```
0 1 75
```

```
1 1 66
```

```
2 1 68
```

```
3 2 74
```

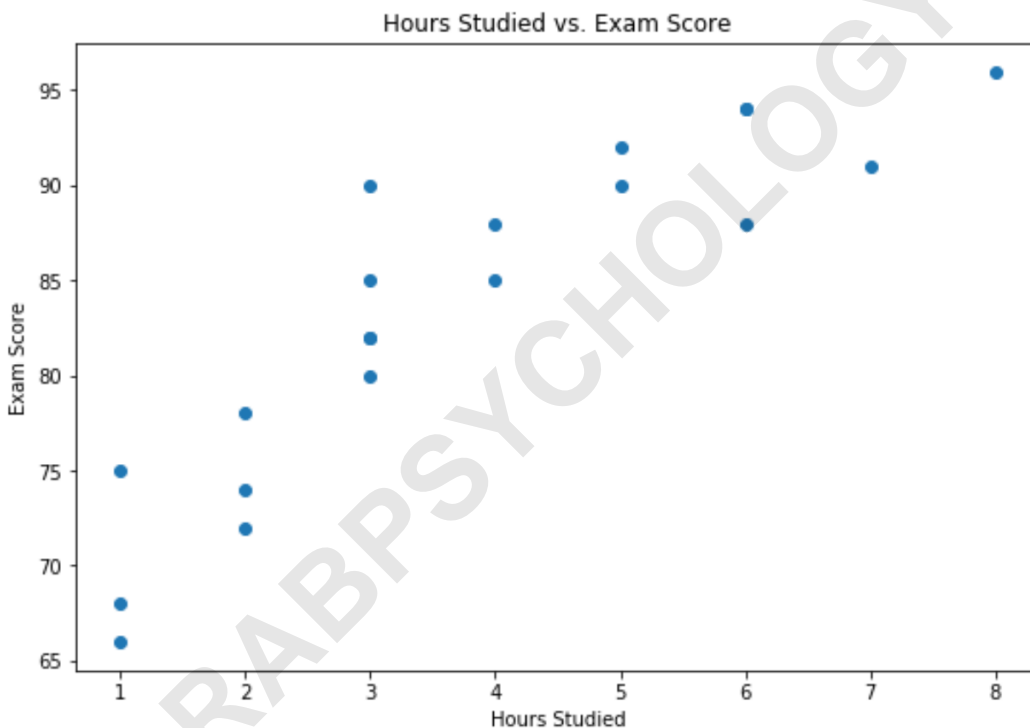
```
4 2 78
```

### 1. Scatterplots

We can use the following syntax to create a scatterplot of hours studied vs. exam score:

```
import matplotlib.pyplot as plt

#create scatterplot of hours vs. score
plt.scatter(df.hours, df.score)
plt.title('Hours Studied vs. Exam Score')
plt.xlabel('Hours Studied')
plt.ylabel('Exam Score')
```



The x-axis shows the hours studied and the y-axis shows the exam score received.

From the plot we can see that there is a positive relationship between the two variables: As hours

**studied increases, exam score tends to increase as well.**

## 2. Correlation Coefficients

**A Pearson Correlation Coefficient is a way to quantify the linear relationship between two variables.**

**We can use the `corr()` function in pandas to create a correlation matrix:**

```
#create correlation matrix  
df.corr()
```

```
hours score  
hours 1.000000 0.891306  
score 0.891306 1.000000
```

**The correlation coefficient turns out to be 0.891. This indicates a strong positive correlation between hours studied and exam score received.**

## 3. Simple Linear Regression

**Simple linear regression is a statistical method we can use to quantify the relationship between two variables.**

We can use the `OLS()` function from the `statsmodels` package to quickly fit a for hours studied and exam score received:

```
import statsmodels.api as sm

#define response variable
y = df

#define explanatory variable
x = df]

#add constant to predictor variables
x = sm.add_constant(x)

#fit linear regression model
model = sm.OLS(y, x).fit()

#view model summary
print(model.summary())
```

## OLS Regression Results

```
=====
=====
```

**Dep. Variable: score R-squared: 0.794**

**Model: OLS Adj. R-squared: 0.783**

**Method: Least Squares F-statistic: 69.56**

**Date: Mon, 22 Nov 2021 Prob (F-statistic): 1.35e-07**

**Time: 16:15:52 Log-Likelihood: -55.886**

**No. Observations: 20 AIC: 115.8**

**Df Residuals: 18 BIC: 117.8**

**Df Model: 1**

**Covariance Type: nonrobust**

=====  
=====

**coef std err t P>|t|**

-----

**const 69.0734 1.965 35.149 0.000 64.945 73.202**

**hours 3.8471 0.461 8.340 0.000 2.878 4.816**

=====  
=====

**Omnibus: 0.171 Durbin-Watson: 1.404**

**Prob(Omnibus): 0.918 Jarque-Bera (JB): 0.177**

**Skew: 0.165 Prob(JB): 0.915**

**Kurtosis: 2.679 Cond. No. 9.37**

=====  
=====

**The fitted regression equation turns out to be:**

**Exam Score = 69.0734 + 3.8471\*(hours studied)**

**This tells us that each additional hour studied is associated with an average increase of 3.8471 in exam score.**

**We can also use the fitted regression equation to predict the score that a student will receive based on their total hours studied.**

**For example, a student who studies for 3 hours is predicted to receive a score of 81.6147:**

**Exam Score = 69.0734 + 3.8471\*(hours studied)**  
**Exam Score = 69.0734 + 3.8471\*(3)**  
**Exam Score = 81.6147**

**Additional Resources**

**The following tutorials provide additional information about bivariate analysis:**