CONTENTS

Introduction to Correlation

Ruslana Dalinina | 01.31.17

Prerequisites

Experience with the specific topic: Novice

Professional experience: No industry experience

To follow this article, the reader should be familiar with Python syntax and have some understanding of basic statistical concepts (e.g. average, standard deviation).

Introduction: What Is Correlation and Why Is It Useful?

Correlation is one of the most widely used – and widely misunderstood – statistical concepts. In this overview, we provide the definitions and intuition behind several types of correlation and illustrate how to calculate correlation using the Python **pandas** library.

The term "correlation" refers to a mutual relationship or association between quantities. In almost any business, it is useful to express one quantity in terms of its relationship with others. For example, sales might increase when the marketing department spends more on TV advertisements, or a customer's average purchase amount on an e-commerce website might depend on a number of factors related to that customer. Often, correlation is the first step to understanding these relationships and subsequently building better business and statistical models.

So, why is correlation a useful metric?

- Correlation can help in predicting one quantity from another
- Correlation can (but often does not, as we will see in some examples below) indicate the presence of a causal relationship
- Correlation is used as a basic quantity and foundation for many other modeling

More formally, correlation is a statistical measure that describes the association between random variables. There are several methods for calculating the correlation coefficient, each measuring different types of strength of association. Below we summarize three of the most widely used methods.

Types of Correlation

CONTENTS

Before we go into the details of how correlation is calculated, it is important to introduce the concept of *covariance*. Covariance is a statistical measure of association between two variables X and Y. First, each variable is centered by subtracting its mean. These centered scores are multiplied together to measure whether the increase in one variable is associated with the increase in another. Finally, expected value (E) of the product of these centered scores is calculated as a summary of association. Intuitively, the product of centered scores can be thought of as the area of a rectangle with each point's distance from the mean describing a side of the rectangle:

$$Cov(X, Y) = E[(X - E[X])(Y - E[Y])]$$

If both variables tend to move in the same direction, we expect the "average" rectangle connecting each point (X_i, Y_i) to the means (X_bar, Y_bar) to have a large and positive diagonal vector, corresponding to a larger positive product in the equation above. If both variables tend to move in opposite directions, we expect the average rectangle to have a diagonal vector that is large and negative, corresponding to a larger negative product in the equation above. If the variables are unrelated, then the vectors should, on average, cancel out – and the total diagonal vector should have a magnitude near 0, corresponding to a product near 0 in the equation above.

If you are wondering what "expected value" is, it is another way of saying the average, or mean μ, of a random variable. It is also referred to as "expectation." In other words, we can write the following equation to express the same quantity in a different way:

$$E(Y) = \bar{Y} = \mu_Y$$

The problem with covariance is that it keeps the scale of the variables X and Y, and therefore can take on any value. This makes interpretation difficult and comparing covariances to each other impossible. For example, $Cov(X, Y) = 5.2$ and $Cov(Z, Q) = 3.1$ tell us that these pairs are positively associated, but it is difficult to tell whether the relationship between X and Y is stronger than Z and Q without looking at the means and distributions of these variables. This is where correlation becomes useful – by standardizing covariance by some measure of variability in the data, it produces a quantity that has intuitive interpretations and consistent scale.

SCROLL TO TOP

Pearson Correlation Coefficient

Pearson is the most widely used correlation coefficient. Pearson correlation measures the linear association between continuous variables. In other words, this coefficient quantifies the degree to which a relationship between two variables can be described by a line. Remarkably, while correlation can have many interpretations, the same formula developed by Karl Pearson over 120 years ago is still the most widely used today.

CONTENTS

In this section, we will introduce several popular formulations and intuitive interpretations for Pearson correlation (referred to as ρ).

The original formula for correlation, developed by Pearson himself, uses raw data and the means of two variables, X and Y:

$$\rho_{X,\,Y} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$

In this formulation, raw observations are centered by subtracting their means and re-scaled by a measure of standard deviations.

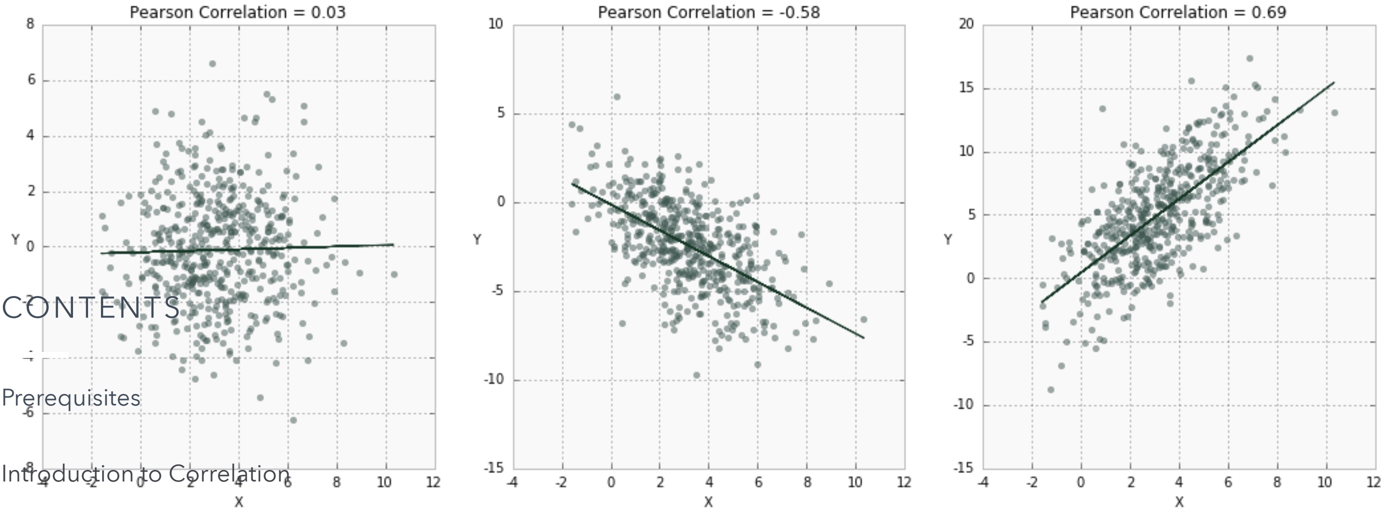
A different way to express the same quantity is in terms of expected values, means μ_X, μ_Y, and standard deviations σ_X, σ_Y:

$$\rho_{X,\,Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

Notice that the numerator of this fraction is identical to the above definition of covariance, since mean and expectation can be used interchangeably. Dividing the covariance between two variables by the product of standard deviations ensures that correlation will always fall between -1 and 1. This makes interpreting the correlation coefficient much easier.

The figure below shows three examples of Pearson correlation. The closer ρ is to 1, the more an increase in one variable associates with an increase in the other. On the other hand, the closer ρ is to -1, the increase in one variable would result in decrease in the other. Note that if X and Y are independent, then ρ is close to 0, but not vice versa! In other words, Pearson correlation can be small even if there is a strong relationship between two variables. We will see shortly how this can be the case.

SCROLL TO TOP



CONTENTS

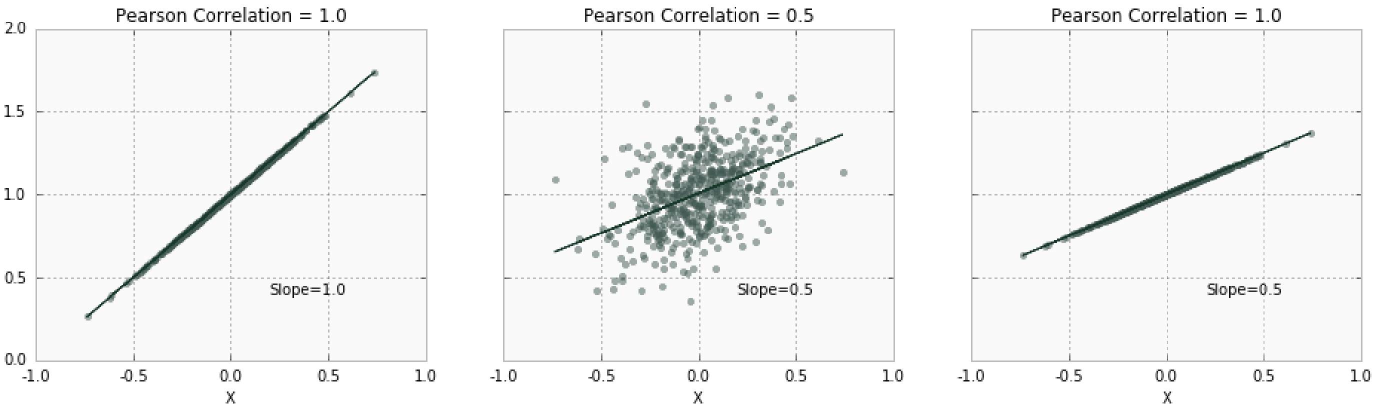
So, how can we interpret Pearson correlation?Turns out, there is a clear connection between Pearson correlation and the slope of a line. In the above figure, a regression line through each scatter plot is shown. The regression line is optimal, as it minimizes the distance of all points to itself. Because of this property, the slope of the regression line of Y and X is mathematically equivalent to correlation between X and Y, standardized by the ratio of their standard deviations:

$$\rho = b\frac{s_x}{s_y}$$

where b is the slope of the regression line of Y from X.In other words, correlation reflects the association and amount of variability between the two variables.This relationship with the slope of the line has two important implications:

1. It makes it more clear why Pearson correlation describes linear relationships
2. It also shows why correlation is important and so widely used in predictive modeling

However, note that in the above equation for ρ, correlation **does not equal slope** – rather, it is standardized by a measure of data variability. For example, it is possible to have a very small magnitude of slope but large correlations between variables. In the figure below, the line describing this relationship is relatively flat, but correlation is 1 since variability s_y is very small:



Note that, so far, we have not made any assumptions about the distribution of X and Y. The only restriction is that Pearson ρ assumes a linear relationship between the two variables. Pearson correlation relies on means and standard deviations, which means it is only defined for distributions where those statistics are finite, making the coefficient sensitive to outliers.Another way to interpret Pearson correlation is to use the coefficient of determination, also knows as R^2. While ρ is unitless, its square is interpreted at the proportion of variance of Y explained by X. In the above example, ρ = -0.65 implies that $(-0.65^2)*100$ = 42% of variation in Y can be explained by X.There are many other ways to interpret ρ. Check out the classic paper " Thirteen ways to look at the correlation coefficient" if you are interested in connections between correlation and vectors, ellipses and more.

Spearman's Correlation Coefficient

SCROLL TO TOP

Spearman's rank correlation coefficient can be defined as a special case of Pearson ρ applied to ranked (sorted) variables. Unlike Pearson, Spearman's correlation is not restricted to linear relationships. Instead, it measures **monotonic association** (only strictly increasing or decreasing, but not mixed) between two variables and relies on the rank order of values. In other words, rather than comparing means and variances, Spearman's coefficient looks at the relative order of values for each variable. This makes it appropriate to use with both continuous and discrete data.

CONTENTS

The formula for Spearman's coefficient looks very similar to that of Pearson, with the distinction of being computed on ranks instead of raw scores:

$$\rho_{rank_X,\ rank_Y} = \frac{cov(rank_X,\ rank_Y)}{\sigma_{rank_X}\sigma_{rank_Y}}$$

If all ranks are unique (i.e. there are no ties in ranks), you can also use a simplified version:

$$\rho_s = 1 - \frac{6\sum d_i^2}{N(N^2-1)}$$

where $d_i = rank(X_i)$ - $rank(Y_i)$ is the difference between the two ranks of each observation and N is the number of observations.

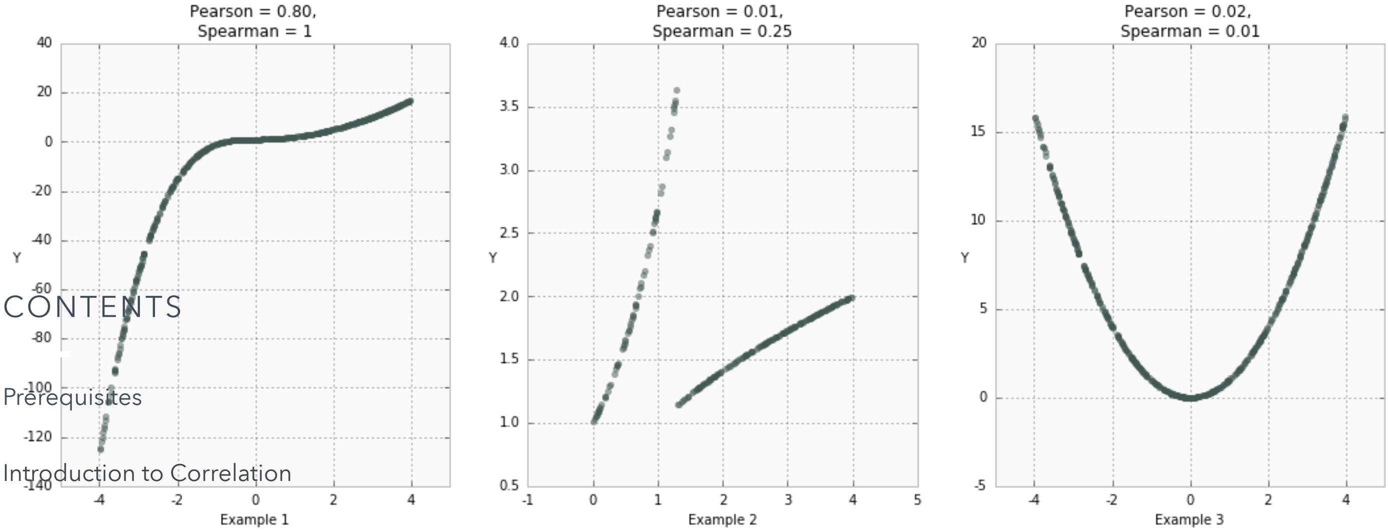
The difference between Spearman and Pearson correlations is best illustrated by example. In the below figure, there are three scenarios with both correlation coefficients shown. In the first example, there is a clear monotonic (always increasing) and non-linear relationship. Since ranks of values perfectly align in this case, the Spearman's coefficient is 1. Pearson correlation is weaker in this case, but it is still showing a very strong association due to the partial linearity of the relationship.

The data in Example 2 shows clear groups in X and a strong, although non-monotonic, association for both groups with Y. In this case, the Pearson correlation is almost 0 since the data is very non-linear. The Spearman rank correlation shows a weak association since the data is non-monotonic.

Finally, Example 3 shows a nearly perfect quadratic relationship centered around 0. However, both correlation coefficients are almost 0 due to the non-monotonic, non-linear, and symmetric nature of the data.

These hypothetical examples illustrate that correlation is by no means an exhaustive summary of relationships within the data. Weak or no correlation does not imply a lack of association, as seen in Example 3, and even a strong correlation coefficient might not fully capture the nature of the relationship. It is always a good idea to use visualization techniques and multiple statistical data summaries to get a better picture of how your variables relate to each other.

SCROLL TO TOP



CONTENTS

Kendall's Tau Coefficient

The third correlation coefficient we will discuss is also based on variable ranks. However, unlike Spearman's coefficient, Kendall's τ does not take into account the difference between ranks – only directional agreement. Therefore, this coefficient is more appropriate for discrete data.

Formally, Kendall's τ coefficient is defined as:

$$\tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{N(N-1)/2}$$

As an example, consider a simple dataset consisting of five observations. In practice, such a small number of data points would not be sufficient nor reliable to draw any conclusions. But here, we consider it for the sake of the simplicity of calculation:

	X	Y
a	1	7
b	2	5
c	3	1
d	4	6
e	5	9

Concordant pairs (x_1, y_1), (x_2, y_2) are pairs of values in which ranks coincide: $x_1 < x_2$ and $y_1 < y_2$ or $x_1 > x_2$ and $y_1 > y_2$. In our mini example, (4,6) and (5,9) in rows d and e is a concordant pair. A discordant pair would be one that does not satisfy this condition, such as (1, 7) and (2, 5). To calculate the numerator of τ, we compare all possible pairs in the dataset and count number of concordant pairs; 6 in this case:

- (1,7) and (5,9)
- (2,5) and (4,6)
- (2,5) and (5,9)
- (3,1) and (4,6)
- (3,1) and (5,9)

SCROLL TO TOP

- (4,6) and (5,9)

and discordant pairs:

- (1,7) and (2,5)
- (1,7) and (3,1)
- (1,7) and (4,6)
- (2,5) and (3,1)

The denominator of Kendall's τ is just the number of possible combinations of pairs, which ensures that τ varies between 1 and -1. With five data points, there are $5 * 4/2 = 10$ possible combinations, making $\tau = (6\text{-}4) / 10 = 0.2$ in this example. Kendall's correlation is particularly useful for discrete data, where the relative position of data points is more important that difference between them.

```
# fake kendall
k = pd.DataFrame()
k['X'] = np.arange(5)+1
k['Y'] = [7, 5,1, 6, 9]
print k.corr(method='kendall')
```

```
     X    Y
X  1.0  0.2
Y  0.2  1.0
```

CONTENTS

Calculating Correlation in Pandas

Below, we show how to calculate correlation for an example problem using a Python library. We will be using a dataset on vehicle fuel efficiency from University of California, Irvine. Let's say it is of interest to see what vehicle characteristics can help explain fuel consumption (mpg) of a vehicle. We begin by reading the dataset from the UCI online data repository and examining first few rows. Dataset documentation states that a special character is used for missing values (`?`), which can be used as one of the parameters to pandas `read_csv()` function:

```
import pandas as pd
path = 'http://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data'

mpg_data = pd.read_csv(path, delim_whitespace=True, header=None,
            names = ['mpg', 'cylinders', 'displacement','horsepower',
            'weight', 'acceleration', 'model_year', 'origin', 'name'],
            na_values='?')
```

SCROLL TO TOP

Upon inspecting the dataset, we see that **`horsepower`** has six missing values, which pandas' correlation method will automatically drop. Since the number of missing values is small, this setting is acceptable for our illustrative example. However, always make sure that dropping missing values is appropriate for your use case. If that is not the case, there are many existing methods for filling in and handling missing values, such as simple mean imputation.

```python
mpg_data.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
mpg             398 non-null float64
cylinders       398 non-null int64
displacement    398 non-null float64
horsepower      392 non-null float64
weight          398 non-null float64
acceleration    398 non-null float64
model_year      398 non-null int64
origin          398 non-null int64
name            398 non-null object
dtypes: float64(5), int64(3), object(1)
memory usage: 28.1+ KB
```

pandas provides a convenient one-line method **`corr()`** for calculating correlation between data frame columns. In our fuel efficiency example, we can check whether heavier vehicles tend to have lower **`mpg`** by passing the method to specific columns:

```python
mpg_data['mpg'].corr(mpg_data['weight'])
```

```
-0.8317409332443354
```

As expected, there seems to be a strong negative correlation between vehicle **`weight`** and **`mpg`**. But what about **`horsepower`** or **`displacement`**? Conveniently, pandas can quickly calculate correlation between all columns in a dataframe. The user can also specify the correlation method: Spearman, Pearson, or Kendall. If no method is specified, Pearson is used by default. Here, we drop model year and origin variables and calculate Pearson correlation between all remaining columns of the data frame:

In [ ]:

```python
# pairwise correlation
mpg_data.drop(['model_year', 'origin'], axis=1).corr(method='spearman')
```

SCROLL TO TOP

Out[ ]:

	mpg	cylinders	displacement	horsepower	weight	acceleration
mpg	1.000000	-0.821864	-0.855692	-0.853616	-0.874947	0.438677
cylinders	-0.821864	1.000000	0.911876	0.816188	0.873314	-0.474189
displacement	-0.855692	0.911876	1.000000	0.876171	0.945986	-0.496512
horsepower	-0.853616	0.816188	0.876171	1.000000	0.878819	-0.658142
weight	-0.874947	0.873314	0.945986	0.878819	1.000000	-0.404550
acceleration	0.438677	-0.474189	-0.496512	-0.658142	-0.404550	1.000000

CONTENTS

pandas also supports highlighting methods for tables, so it is easier to see high and low correlations. It is important to understand possible correlations in your data, especially when building a regression model. Strongly correlated predictors, phenomenon referred to as multicollinearity, will cause coefficient estimates to be less reliable. Below is an example of calculating Pearson correlation on our data and using a color gradient to format the resulting table:

```
model_year', 'origin'], axis=1).corr(method='pearson').style.format("{:.2}").background_gradient(cmap=plt.g
```

	mpg	cylinders	displacement	horsepower	weight	acceleration
mpg	1.0	-0.78	-0.8	-0.78	-0.83	0.42
cylinders	-0.78	1.0	0.95	0.84	0.9	-0.51
displacement	-0.8	0.95	1.0	0.9	0.93	-0.54
horsepower	-0.78	0.84	0.9	1.0	0.86	-0.69
weight	-0.83	0.9	0.93	0.86	1.0	-0.42
acceleration	0.42	-0.51	-0.54	-0.69	-0.42	1.0

Finally, to visually inspect the relationship between **mpg**, **weight**, **horsepower**, and **acceleration**, we can plot these values and calculate Pearson and Spearman coefficients. The dataset at hand consists of less than 400 points, which can be easily displayed on a scatter plot. If you are dealing with much larger datasets, consider taking a sample of your data first to speed up the process and produce more readable plots.

In this case, Spearman's coefficient is higher than Pearson for **horsepower** and **weight**, since relationship is non-linear. For **acceleration**, both coefficients are close since the relationship is not as clearly defined:

```
# plot correlated values
plt.rcParams['figure.figsize'] = [16, 6]

fig, ax = plt.subplots(nrows=1, ncols=3)

ax=ax.flatten()
```

```python

cols = ['weight', 'horsepower', 'acceleration']
colors=['#415952', '#f35134', '#243AB5', '#243AB5']
j=0

for i in ax:
    if j==0:
        i.set_ylabel('MPG')
    i.scatter(mpg_data[cols[j]], mpg_data['mpg'],  alpha=0.5, color=colors[j])
    i.set_xlabel(cols[j])
    i.set_title('Pearson: %s'%mpg_data.corr().loc[cols[j]]['mpg'].round(2)+' Spearman: %s'%mpg_data.corr(me
    j+=1

plt.show()
```

Types of Correlation

Pearson Correlation Coefficient

Spearman's Correlation Coefficient

Kendall's Tau Coefficient

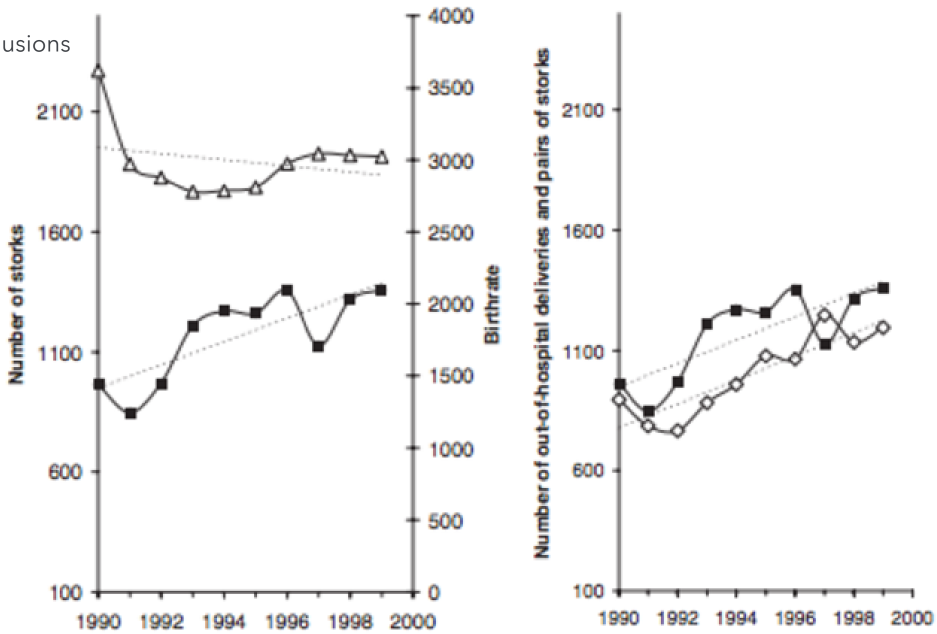
Calculating Correlation in Pandas

Correlation and Causation

Conclusions

Correlation and Causation

The relationships between variables in our fuel efficiency example were very intuitive and explainable through vehicle mechanics. However, things are not always this straightforward. It is a well-known fact that correlation does not imply causation, and therefore, any strong correlation should be thought of critically. For example, German researchers used the concept of correlation in this humorous paper to support a theory that babies are delivered by storks. This figure shows the correlation between the number of storks and baby deliveries:

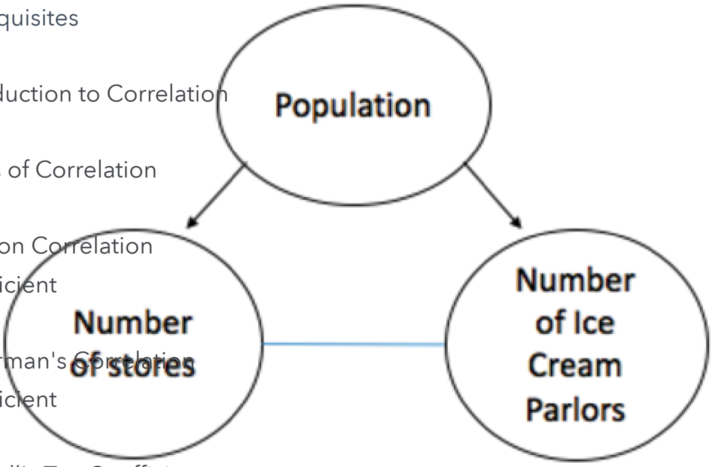


The chart on the left shows an increasing trend in the number of storks (black line) and a decreasing trend in the number of clinical deliveries. On the other hand, the chart on the right shows that a number of out-of-hospital deliveries (white square marks) follow the increasing pattern in the number of storks. Looking at the correlation between these series, the authors suggest that the increase in out-of-hospital deliveries paired with the increase in the number of storks and the simultaneous decrease in hospital d SCROLL TO TOP t that more and more babies in Germany are being delivered by storks.

Of course, this is a silly example. Nonetheless, it demonstrates an important point: Spurious statistical associations can be found in a multitude of quantities, simply due to chance.

Often, a relationship may appear to be causal through high correlation due to some unobserved variables. For example, the number of grocery stores in a city can be strongly correlated with the number of ice cream creameries. However, there is an obvious hidden variable here – the population size of the city:

CONTENTS



These examples show how correlation is only one data summary statistic that by no means tells the complete story of relationships in the data.

Conclusions

This overview is a primer of correlation types and interpretations. We have introduced three popular correlation methods and demonstrated how to calculate them using **pandas**. Correlation is a useful quantity in many applications, especially when conducting a regression analysis. While the methods listed here are widely used and cover most use cases, there are other measures of association not covered here, such phi coefficient for binary data or mutual information.

References

Rodgers, J. L., & Nicewander, W. A. (1988). Thirteen Ways to Look at the Correlation Coefficient. The American Statistician, 42(1), 59-66.

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Hofer, T., Przyrembel, H., & Verleger, S. (2004). New evidence for the Theory of the Stork. Paediatric and Perinatal Epidemiology, 18(1), 88-92.

Data Science

SUBSCRIBE TO OUR NEWSLETTER →

Technology Solutions Resources Tools Company

Conclusions

SCROLL TO TOP