Overview

What is Correlation?
Pearson's Correlation
Spearman's Correlation
Co-variance

What is Correlation?

- Ever wondered how your data variables are linked to one another? Variables within a dataset can be related for lots of reasons.
  For example:
- One variable could cause or depend on the values of another variable.
- One variable could be lightly associated with another variable.
- Two variables could depend on a third unknown variable.
- All of these aspects of correlation and how data variables are dependent or can relate to one another get us thinking about their use. Correlation is very useful in data analysis and modelling to better understand the relationships between variables.

- The statistical relationship between two variables is referred to as their correlation.

- A correlation could be presented in different ways:
- Positive Correlation: both variables change in the same direction.
- Neutral Correlation: No relationship in the change of the variables.
- Negative Correlation: variables change in opposite directions.

- The performance of some algorithms can deteriorate if two or more variables are tightly related, called multicollinearity. An example is linear regression, where one of the offending correlated variables should be removed in order to improve the skill of the model.
- We may also be interested in the correlation between input variables with the output variable in order to provide insight into which variables may or may not be relevant as input for developing a model.
- The structure of the relationship may be known, e.g. it may be linear, or we may have no idea whether a relationship exists between two variables or what structure it may take. Depending on what is known about the relationship and the distribution of the variables, different correlation scores can be calculated.
- In this tutorial guide, we will delve into a correlation score tailored for variables with a Gaussian distribution and a linear relationship. We will also explore another score that does not rely on a specific distribution and captures any monotonic (either increasing or decreasing) relationship.

The Dataset for demo:-

- Before we go diving into correlation methods, let's define a dataset we can use to test the methods.

- We will generate 1,000 samples of two variables with a strong positive correlation. The first variable will be random numbers drawn from a Gaussian distribution with a mean of 100 and a standard deviation of 20. The second variable will be values from the first variable with Gaussian noise added with a mean of 50 and a standard deviation of 10.
- We will use the randn() function to generate random Gaussian values with a mean of 0 and a standard deviation of 1, then multiply the results by our own standard deviation and add the mean to shift the values into the preferred range.
- The pseudorandom number generator is seeded to ensure that we get the same sample of numbers each time the code is run.

Python command is .cov()

An complete example: code is as follows:-

```
# generate data1 and data2
from numpy.random import randn
from numpy.random import seed
from numpy import cov
# seed random number generator
seed(1)
# prepare data
data1 = 20 * randn(1000) + 100
data2 = data1 + (10 * randn(1000) + 50)
# calculate the covariance matrix
covariance = cov(data1, data2)
print(covariance)
```

- Running the example calculates and prints the covariance matrix.

```
[[385.33297729 389.7545618 ]
[389.7545618  500.38006058]]
```

- The covariance between the two variables is 389.75. We can see that it is positive, suggesting the variables change in the same direction as we expect.

- In the example above, since the dataset was contrived (generated manually - with each variable drawn from a Gaussian distribution and the variables linearly correlated)
  covariance is a reasonable method for describing the relationship.

Pearson's Correlation

- Named after Karl Pearson, The Pearson correlation coefficient can be used to summarize the strength of the linear relationship between two data samples.
- Pearson's correlation coefficient is calculated by dividing the covariance of the two variables by the product of their respective standard deviations.
- Formula

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

r = correlation coefficient
x_i = values of the x-variable in a sample
$\bar{x}$ = mean of the values of the x-variable
y_i = values of the y-variable in a sample
$\bar{y}$ = mean of the values of the y-variable

The coefficient returns a value between -1 and 1, symbolizing the full spectrum of correlation: from a complete negative correlation to a total positive correlation.

A value of 0 means no correlation.

The value must be interpreted, where often a value below -0.5 or above 0.5 indicates a notable correlation, and values below those values suggests a less notable correlation.

The pearsonr() function (from Scipy library) can be used to calculate the Pearson's correlation coefficient between two data samples with the same length.

An example of code - program :-

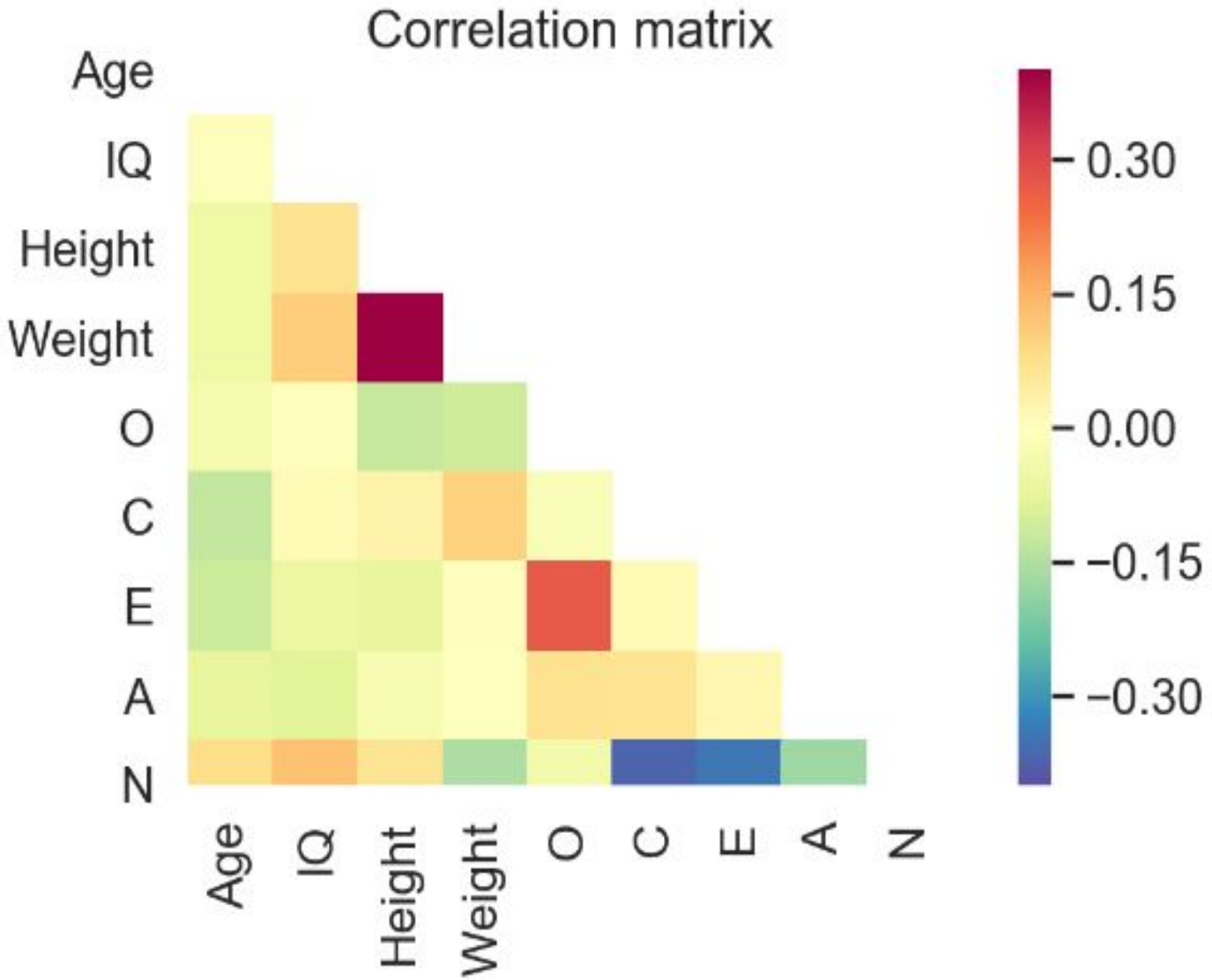
for only 2 columns (features) : data1 & data2,

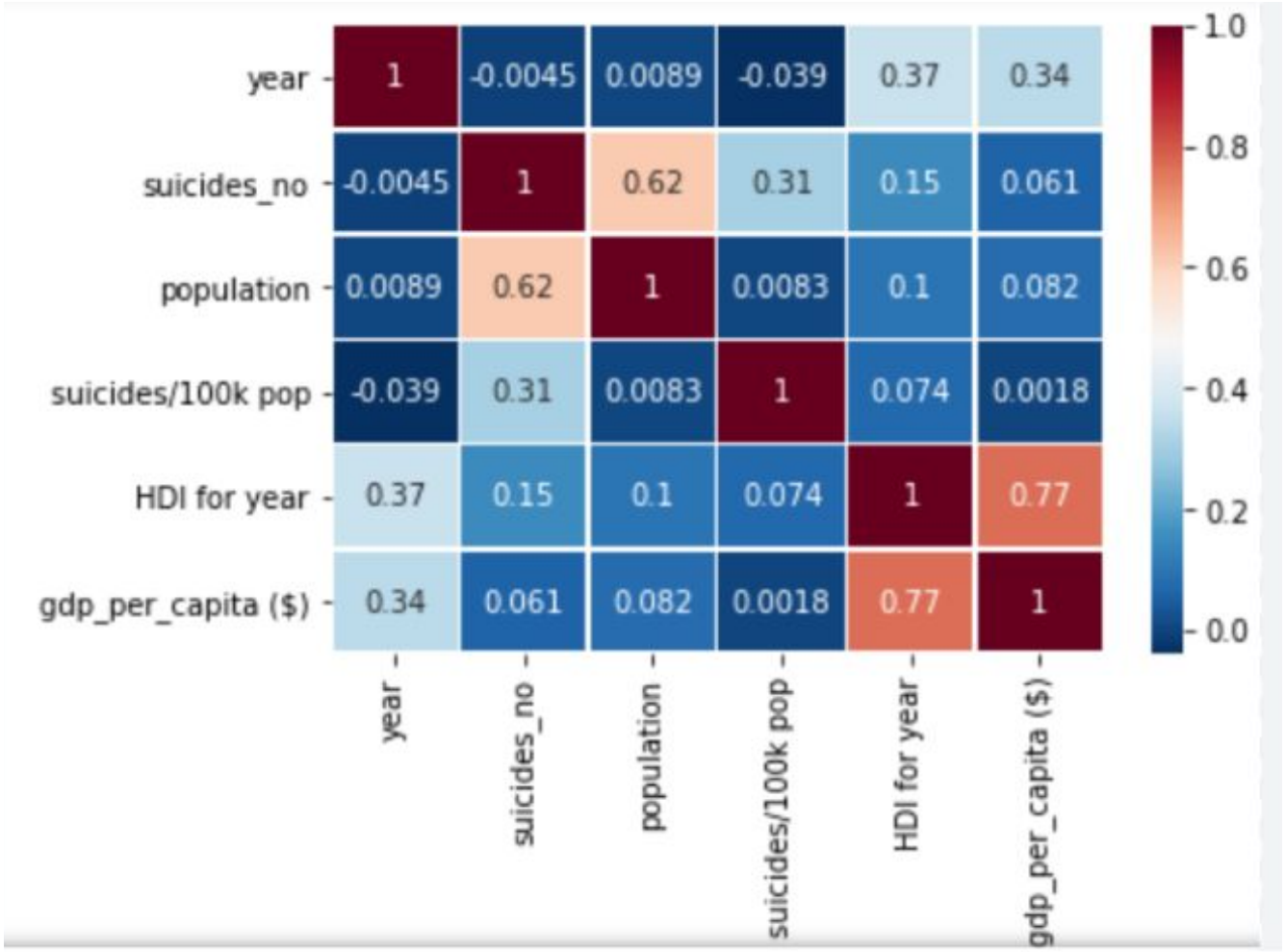
```
from numpy.random import randn
from numpy.random import seed
from scipy.stats import pearsonr
# seed random number generator
seed(1)
# prepare data
data1 = 20 * randn(1000) + 100
data2 = data1 + (10 * randn(1000) + 50)
# calculate Pearson's correlation
corr, _ = pearsonr(data1, data2)
print('Pearsons correlation: %.3f' % corr)
```

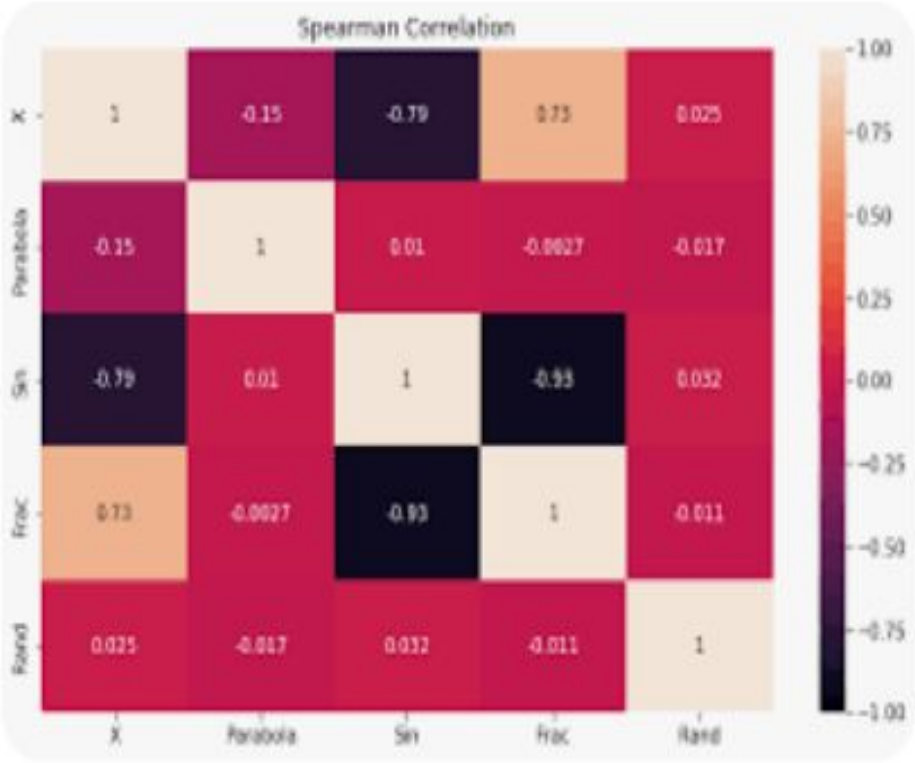
Upon execution, Pearson's correlation coefficient is determined and displayed.

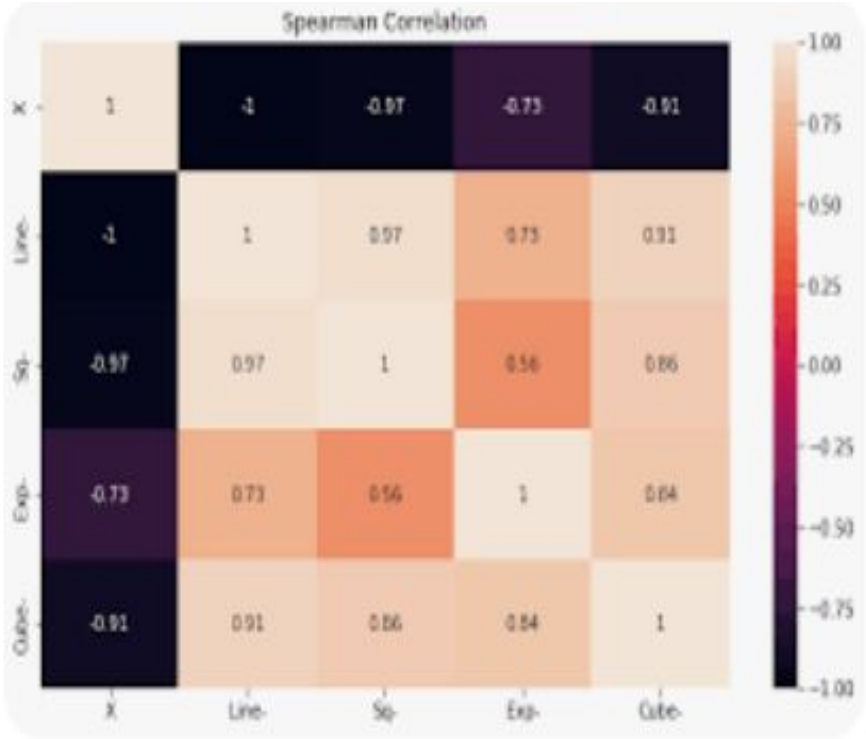
The evident positive correlation of 0.888 between the two variables is strong, as it surpasses the 0.5 threshold and approaches 1.0.

Pearsons correlation: 0.888


Correlation matrix
Age
IQ
Height
Weight
O
C
E
A
N
0.30
0.15
0.00
−0.15
−0.30


year
suicides_no
population
suicides/100k pop
HDI for year
gdp_per_capita ($)
1
-0.0045
0.0089
-0.039
0.37
0.34
-0.0045
1
0.62
0.31
0.15
0.061
0.0089
0.62
1
0.0083
0.1
0.082
-0.039
0.31
0.0083
1
0.074
0.0018
0.37
0.15
0.1
0.074
1
0.77
0.34
0.061
0.082
0.0018
0.77
1
1.0
0.8
0.6
0.4
0.2
0.0


Spearman Correlation
X
Parabola
Sin
Frac
Rand
1
-0.15
-0.79
0.73
0.025
-0.15
1
0.01
-0.0027
-0.017
-0.79
0.01
1
-0.93
0.032
0.73
-0.0027
-0.93
1
-0.011
0.025
-0.017
0.032
-0.011
1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


Spearman Correlation
X
Line
Sq
Exp
Cube
1
-1
-0.97
-0.73
-0.91
-1
1
0.97
0.73
0.91
-0.97
0.97
1
0.56
0.86
-0.73
0.73
0.56
1
0.84
-0.91
0.91
0.86
0.84
1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

Spearman's Correlation

- Named after Charles Spearman, Spearman's correlation coefficient can be used to summarize the strength between the two data samples. This test of relationship can also be used if there is a linear relationship between the variables but will have slightly less power (e.g. may result in lower coefficient scores).
- **As with the Pearson correlation coefficient, the scores are between -1 and 1 for perfectly negatively correlated variables and perfectly positively correlated respectively.**
- Instead of directly working with the data samples, it operates on the relative ranks of data values.

Formula

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

ρ = Spearman's rank correlation coefficient

d_i = difference between the two ranks of each observation

n = number of observations

- If you are unsure of the distribution and possible relationships between two variables, the Spearman correlation coefficient is a good tool to use.

- The spearmanr() SciPy function can be used to calculate the Spearman's correlation coefficient between two data samples with the same length.

- We can calculate the correlation between the two variables in our test problem.

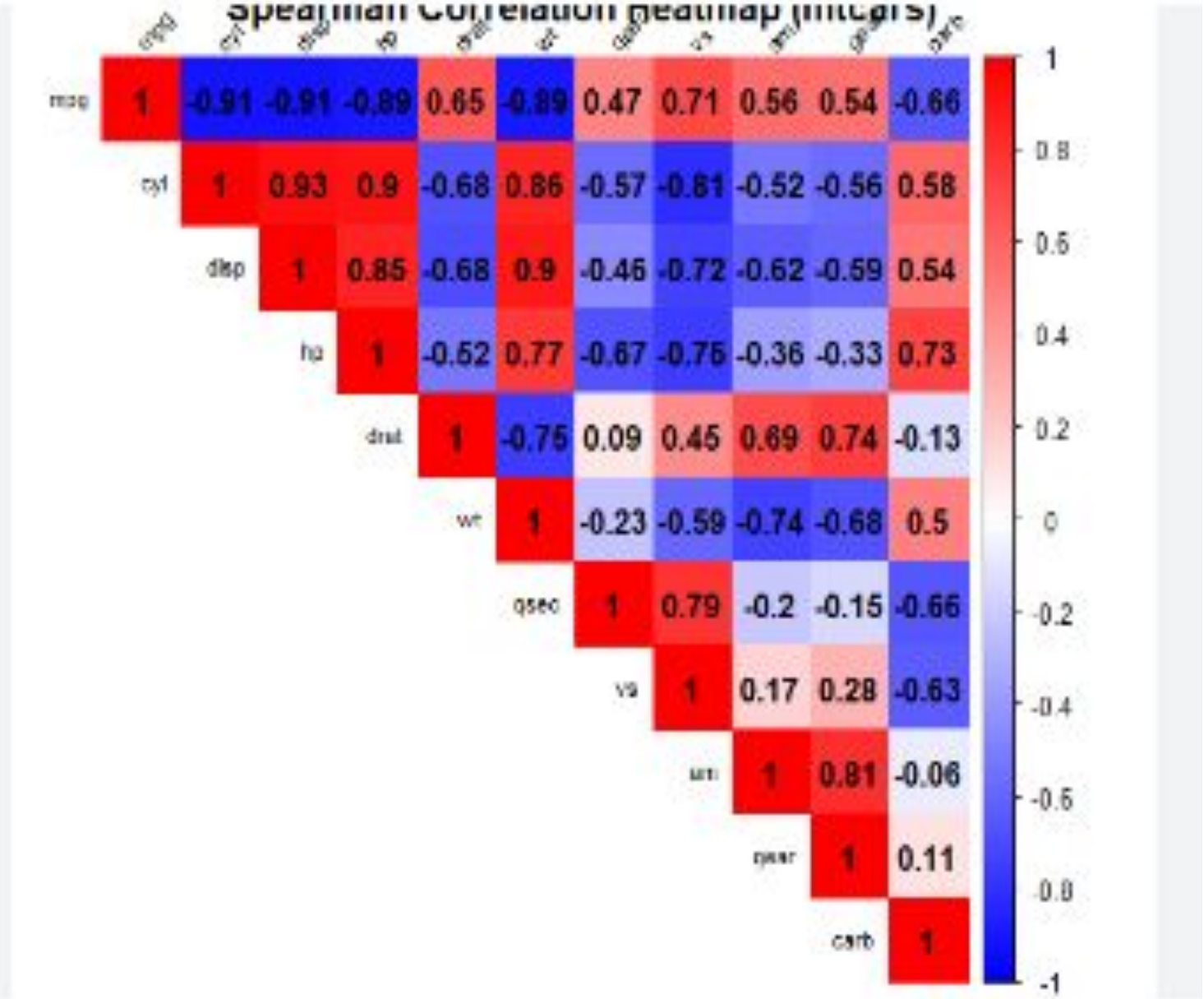
The complete example is listed below.

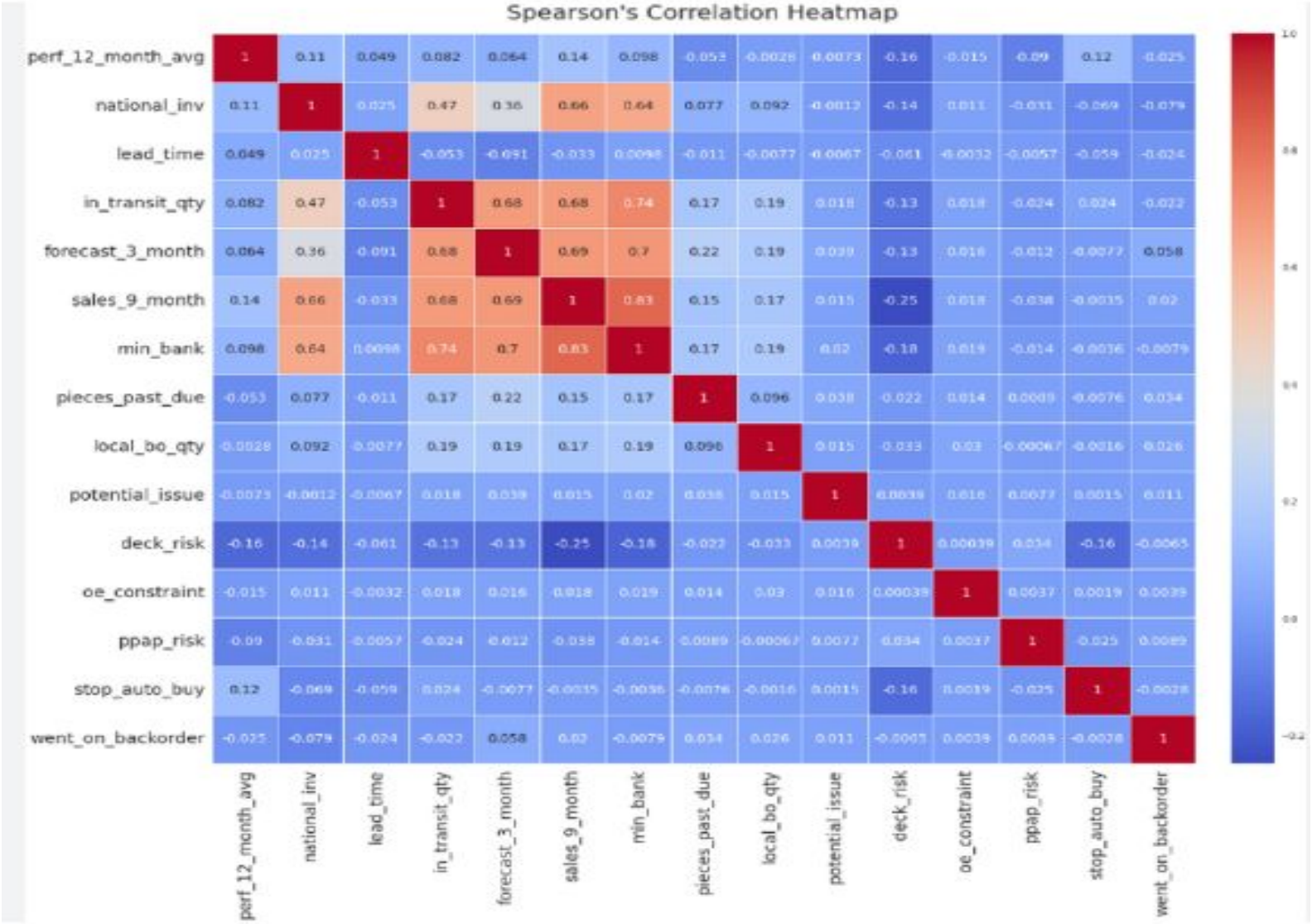
```
# calculate the spearmans's correlation between two variables
from numpy.random import randn
from numpy.random import seed
from scipy.stats import spearmanr
# seed random number generator
seed(1)
# prepare data
data1 = 20 * randn(1000) + 100
data2 = data1 + (10 * randn(1000) + 50)
# calculate spearman's correlation
corr, _ = spearmanr(data1, data2)
print('Spearmans correlation: %.3f' % corr)
```

- Running the example calculates and prints the Spearman's correlation coefficient.

Spearmans correlation: 0.872

- As with the Pearson's correlation coefficient, the coefficient can be calculated pair-wise for each variable in a dataset to give a correlation matrix for review.


Spearman Correlation Heatmap (integers)
mpg
cyl
disp
hp
drat
wt
qsec
vs
am
gear
carb
1 -0.91 -0.91 -0.89 0.65 -0.89 0.47 0.71 0.56 0.54 -0.66
1 0.93 0.9 -0.68 0.86 -0.57 -0.81 -0.52 -0.56 0.58
1 0.85 -0.68 0.9 -0.46 -0.72 -0.62 -0.59 0.54
1 -0.52 0.77 -0.67 -0.75 -0.36 -0.33 0.73
1 -0.75 0.09 0.45 0.69 0.74 -0.13
1 -0.23 -0.59 -0.74 -0.68 0.5
1 0.79 -0.2 -0.15 -0.66
1 0.17 0.28 -0.63
1 0.81 -0.06
1 0.11
1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


Spearson's Correlation Heatmap
perf_12_month_avg
national_inv
lead_time
in_transit_qty
forecast_3_month
sales_9_month
min_bank
pieces_past_due
local_bo_qty
potential_issue
deck_risk
oe_constraint
ppap_risk
stop_auto_buy
went_on_backorder
1 0.11 0.049 0.082 0.064 0.14 0.098 -0.053 -0.0028 0.0073 -0.16 -0.015 -0.09 0.12 -0.025
0.11 1 0.025 0.47 0.36 0.66 0.64 0.077 0.092 -0.0012 -0.14 0.011 -0.031 -0.069 -0.079
0.049 0.025 1 -0.053 -0.091 -0.033 0.0098 -0.011 -0.0077 -0.0067 -0.061 -0.0032 -0.0057 -0.059 -0.024
0.082 0.47 -0.053 1 0.68 0.68 0.74 0.17 0.19 0.018 -0.13 0.018 -0.024 0.024 -0.022
0.064 0.36 -0.091 0.68 1 0.69 0.7 0.22 0.19 0.039 -0.13 0.016 -0.012 -0.0077 0.058
0.14 0.66 -0.033 0.68 0.69 1 0.83 0.15 0.17 0.015 -0.25 0.018 -0.038 -0.0035 0.02
0.098 0.64 0.0098 0.74 0.7 0.83 1 0.17 0.19 0.02 -0.18 0.019 -0.014 -0.0036 -0.0079
-0.053 0.077 -0.011 0.17 0.22 0.15 0.17 1 0.096 0.038 -0.022 0.014 0.0089 -0.0076 0.034
-0.0028 0.092 -0.0077 0.19 0.19 0.17 0.19 0.096 1 0.015 -0.033 0.03 -0.00067 -0.0016 0.026
-0.0073 -0.0012 -0.0067 0.018 0.039 0.015 0.02 0.038 0.015 1 -0.0039 0.016 0.0077 0.0015 0.011
-0.16 -0.14 -0.061 -0.13 -0.13 -0.25 -0.18 -0.022 -0.033 0.0039 1 0.00039 0.034 -0.16 -0.0065
-0.015 0.011 -0.0032 0.018 0.016 0.018 0.019 0.014 0.03 0.016 0.00039 1 0.0037 0.0019 0.0039
0.09 -0.031 -0.0057 -0.024 -0.012 -0.038 -0.014 0.0089 -0.00067 0.0077 0.034 0.0037 1 -0.025 0.0089
0.12 -0.069 -0.059 0.024 -0.0077 -0.0035 -0.0036 -0.0076 -0.0016 0.0015 -0.16 0.0039 -0.025 1 -0.0028
-0.025 -0.079 -0.024 -0.022 0.058 0.02 -0.0079 0.034 0.026 0.011 -0.0065 0.0039 0.0089 -0.0028 1
perf_12_month_avg
national_inv
lead_time
in_transit_qty
forecast_3_month
sales_9_month
min_bank
pieces_past_due
local_bo_qty
potential_issue
deck_risk
oe_constraint
ppap_risk
stop_auto_buy
went_on_backorder
1.0
0.8
0.6
0.4
0.2
0.0
-0.2

Covariance

- At the core of many statistical analyses is the concept of linear relationships between variables. This is a relationship that is consistently additive across the two data samples.
- This relationship can be summarized between two variables, called the covariance. It is calculated as the average of the product between the values from each sample, where the values have been centred (by subtracting their respective means).

- The calculation of the sample covariance is as follows:

cov(X, Y) = (sum (x - mean(X)) * (y - mean(Y))) * 1/(n-1)

- The use of the mean in the calculation implies that each data should ideally adhere to a Gaussian or Gaussian-like distribution.
- The covariance sign can be interpreted as whether the two variables change in the same direction (positive) or change in different directions (negative). The magnitude of the covariance is not easily interpreted. A covariance value of zero indicates that both variables are completely independent.

- The cov() NumPy function can be used to calculate a covariance matrix between two or more variables.

covariance = cov(data1, data2)

- The diagonal of the matrix contains the covariance between each variable and itself. The other values in the matrix represent the covariance between the two variables; in this case, the remaining two values are the same given that we are calculating the covariance for only two variables.