

[Start Here](#)[Blog](#)[Books](#)[About](#)[Contact](#)

Need help with machine learning? [Take the FREE Crash-Course.](#)

How To Implement Simple Linear Regression From Scratch With Python

by **Jason Brownlee** on October 26, 2016 in **Algorithms From Scratch**



Linear regression is a prediction method that is more than 200 years old.

[Simple linear regression](#) is a great first machine learning algorithm to implement as it requires you to estimate properties from your training dataset, but is simple enough for beginners to understand.

In this tutorial, you will discover how to implement the simple linear regression algorithm from scratch in Python.

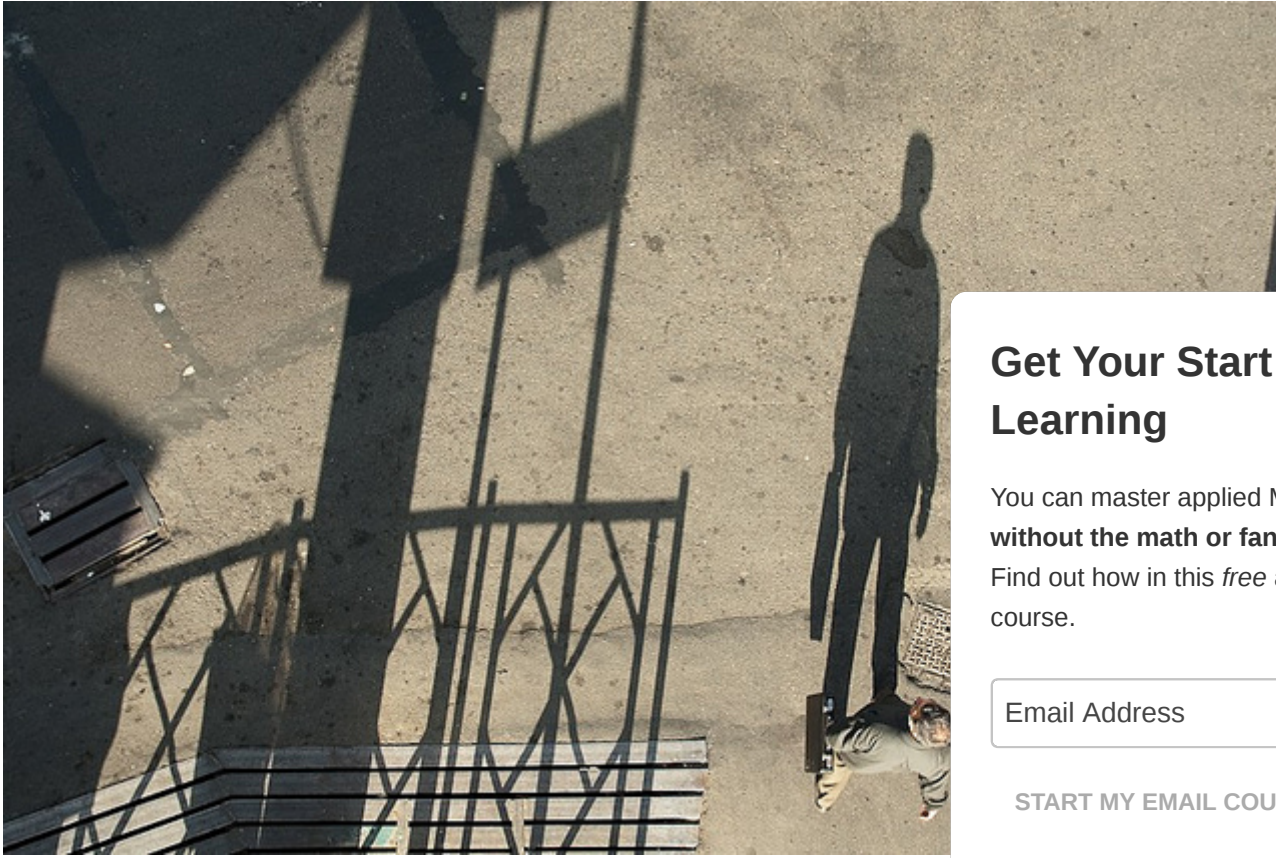
After completing this tutorial you will know:

- How to estimate statistical quantities from training data.

[Get Your Start in Machine Learning](#)

- How to estimate linear regression coefficients from data.
- How to make predictions using linear regression for new data.

Let's get started.



How To Implement Simple Linear Regression From Scratch With Python
Photo by [Kamyar Adl](#), some rights reserved.

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Description

This section is divided into two parts, a description of the simple linear regression technique and a description of the dataset to which we will later apply it.

Simple Linear Regression

Get Your Start in Machine Learning

Linear regression assumes a linear or straight line relationship between the input variables (X) and the single output variable (y).

More specifically, that output (y) can be calculated from a linear combination of the input variables (X). When there is a single input variable, the method is referred to as a simple linear regression.

In simple linear regression we can use statistics on the training data to estimate the coefficients required by the model to make predictions on new data.

The line for a simple linear regression model can be written as:

$$y = b_0 + b_1 * x$$

where b_0 and b_1 are the coefficients we must estimate from the training data.

Once the coefficients are known, we can use this equation to estimate output values for y given new

It requires that you calculate statistical properties from the data such as mean, variance and covariance.

All the algebra has been taken care of and we are left with some arithmetic to implement to estimate

Briefly, we can estimate the coefficients as follows:

$$\begin{aligned} B_1 &= \frac{\sum((x(i) - \text{mean}(x)) * (y(i) - \text{mean}(y)))}{\sum((x(i) - \text{mean}(x))^2)} \\ B_0 &= \text{mean}(y) - B_1 * \text{mean}(x) \end{aligned}$$

where the i refers to the value of the ith value of the input x or output y.

Don't worry if this is not clear right now, these are the functions will implement in the tutorial.

Swedish Insurance Dataset

We will use a real dataset to demonstrate simple linear regression.

The dataset is called the "Auto Insurance in Sweden" dataset and involves predicting the total payment for all the claims in thousands of Swedish Kronor (y) given the total number of claims (x).

This means that for a new number of claims (x) we will be able to predict the total payment of claims

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

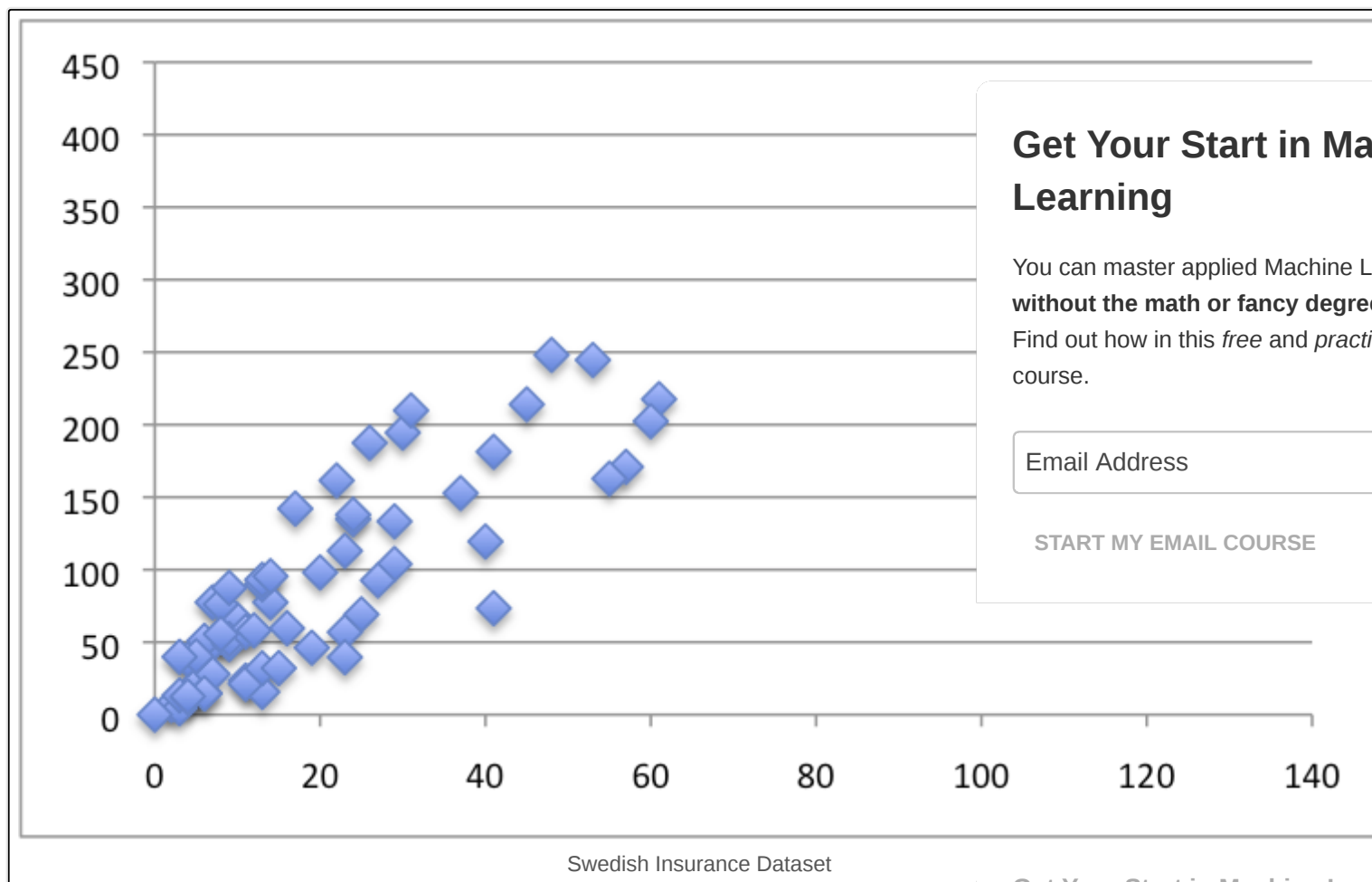
Get Your Start in Machine Learning

Here is a small sample of the first 5 records of the dataset.

```
1 108,392.5
2 19,46.2
3 13,15.7
4 124,422.2
5 40,119.4
```

Using the Zero Rule algorithm (that predicts the mean value) a Root Mean Squared Error or RMSE of about 72.251 (thousands of Kronor) is expected.

Below is a scatter plot of the entire dataset.



Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

You can download the raw dataset from [here](#) or [here](#).

Save it to a CSV file in your local working directory with the name “**insurance.csv**”.

Note, you may need to convert the European “,” to the decimal “.”. You will also need change the file from white-space-separated variables to CSV format.

Tutorial

This tutorial is broken down into five parts:

1. Calculate Mean and Variance.
2. Calculate Covariance.
3. Estimate Coefficients.
4. Make Predictions.
5. Predict Insurance.

These steps will give you the foundation you need to implement and train simple linear regression models.

1. Calculate Mean and Variance

The first step is to estimate the mean and the variance of both the input and output variables from the dataset.

The mean of a list of numbers can be calculated as:

```
1 mean(x) = sum(x) / count(x)
```

Below is a function named **mean()** that implements this behavior for a list of numbers.

```
1 # Calculate the mean value of a list of numbers
2 def mean(values):
3     return sum(values) / float(len(values))
```

The variance is the sum squared difference for each value from the mean value.

Variance for a list of numbers can be calculated as:

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**. Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

```
1 variance = sum( (x - mean(x))^2 )
```

Below is a function named **variance()** that calculates the variance of a list of numbers. It requires the mean of the list to be provided as an argument, just so we don't have to calculate it more than once.

```
1 # Calculate the variance of a list of numbers
2 def variance(values, mean):
3     return sum([(x-mean)**2 for x in values])
```

We can put these two functions together and test them on a small contrived dataset.

Below is a small dataset of x and y values.

NOTE: delete the column headers from this data if you save it to a .CSV file for use with the final code example.

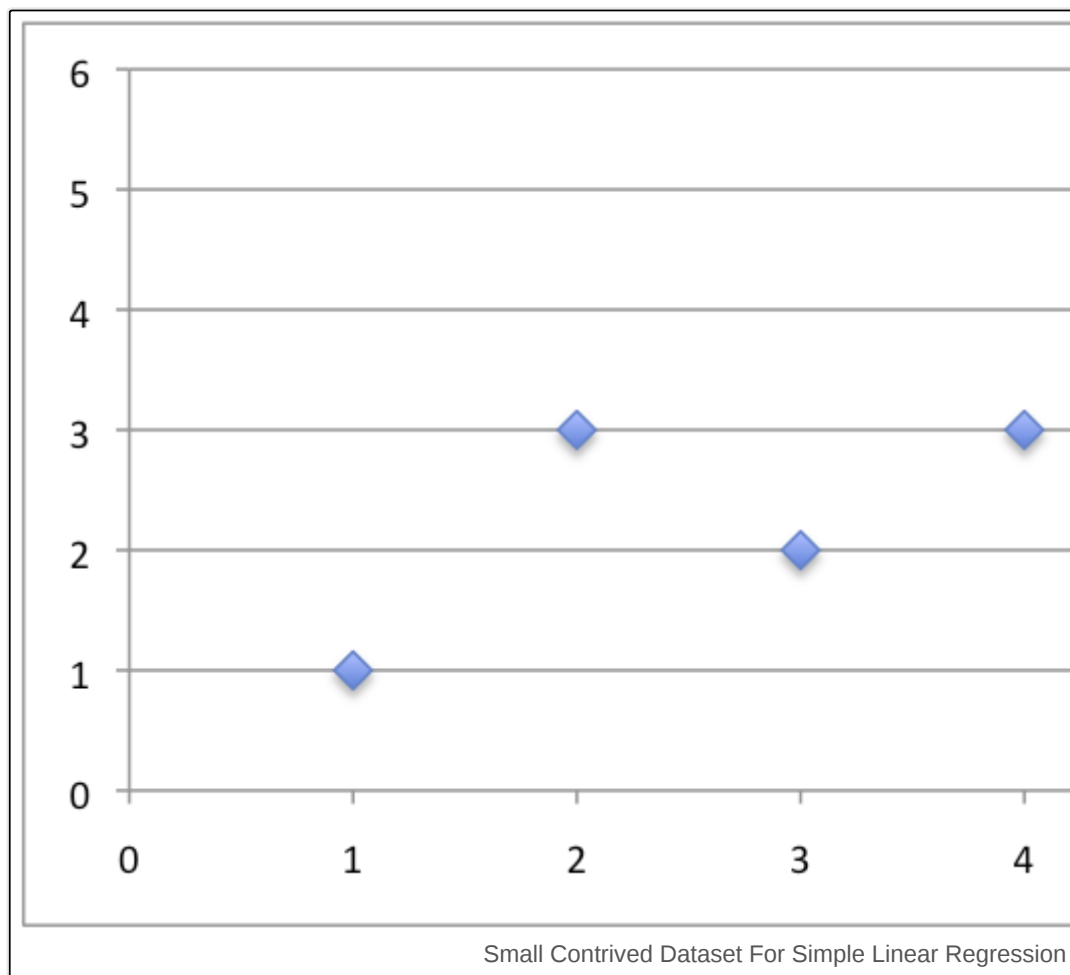
```
1 x, y
2 1, 1
3 2, 3
4 4, 3
5 3, 2
6 5, 5
```

We can plot this dataset on a scatter plot graph as follows:

Get Your Start in Machine Learning ×

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

We can calculate the mean and variance for both the x and y values in the example below.

```
1 # Estimate Mean and Variance
2
3 # Calculate the mean value of a list of numbers
4 def mean(values):
5     return sum(values) / float(len(values))
6
7 # Calculate the variance of a list of numbers
8 def variance(values, mean):
9     return sum([(x-mean)**2 for x in values])
10
11 # calculate mean and variance
```

Get Your Start in Machine Learning

```

12 dataset = [[1, 1], [2, 3], [4, 3], [3, 2], [5, 5]]
13 x = [row[0] for row in dataset]
14 y = [row[1] for row in dataset]
15 mean_x, mean_y = mean(x), mean(y)
16 var_x, var_y = variance(x, mean_x), variance(y, mean_y)
17 print('x stats: mean=%.3f variance=%.3f' % (mean_x, var_x))
18 print('y stats: mean=%.3f variance=%.3f' % (mean_y, var_y))

```

Running this example prints out the mean and variance for both columns.

```

1 x stats: mean=3.000 variance=10.000
2 y stats: mean=2.800 variance=8.800

```

This is our first step, next we need to put these values to use in calculating the covariance.

2. Calculate Covariance

The covariance of two groups of numbers describes how those numbers change together.

Covariance is a generalization of correlation. Correlation describes the relationship between two groups of numbers. Covariance describes the relationship between two or more groups of numbers.

Additionally, covariance can be normalized to produce a correlation value.

Nevertheless, we can calculate the covariance between two variables as follows:

```

1 covariance = sum((x[i] - mean(x)) * (y[i] - mean(y)))

```

Below is a function named **covariance()** that implements this statistic. It builds upon the previous step by taking the mean of these values as arguments.

```

1 # Calculate covariance between x and y
2 def covariance(x, mean_x, y, mean_y):
3     covar = 0.0
4     for i in range(len(x)):
5         covar += (x[i] - mean_x) * (y[i] - mean_y)
6     return covar

```

We can test the calculation of the covariance on the same small contrived dataset as in the previous section.

Get Your Start in Machine Learning ×

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Putting it all together we get the example below.

```

1 # Calculate Covariance
2
3 # Calculate the mean value of a list of numbers
4 def mean(values):
5     return sum(values) / float(len(values))
6
7 # Calculate covariance between x and y
8 def covariance(x, mean_x, y, mean_y):
9     covar = 0.0
10    for i in range(len(x)):
11        covar += (x[i] - mean_x) * (y[i] - mean_y)
12    return covar
13
14 # calculate covariance
15 dataset = [[1, 1], [2, 3], [4, 3], [3, 2], [5, 5]]
16 x = [row[0] for row in dataset]
17 y = [row[1] for row in dataset]
18 mean_x, mean_y = mean(x), mean(y)
19 covar = covariance(x, mean_x, y, mean_y)
20 print('Covariance: %.3f' % (covar))

```

Running this example prints the covariance for the x and y variables.

```

1 Covariance: 8.000

```

We now have all the pieces in place to calculate the coefficients for our model.

3. Estimate Coefficients

We must estimate the values for two coefficients in simple linear regression.

The first is B1 which can be estimated as:

```

1 B1 = sum((x(i) - mean(x)) * (y(i) - mean(y))) / sum( (x(i) - mean(x))^2 )

```

We have learned some things above and can simplify this arithmetic to:

```

1 B1 = covariance(x, y) / variance(x)

```

We already have functions to calculate **covariance()** and **variance()**.

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

Next, we need to estimate a value for B_0 , also called the intercept as it controls the starting point of the line where it intersects the y-axis.

```
1 B0 = mean(y) - B1 * mean(x)
```

Again, we know how to estimate B_1 and we have a function to estimate `mean()`.

We can put all of this together into a function named `coefficients()` that takes the dataset as an argument and returns the coefficients.

```
1 # Calculate coefficients
2 def coefficients(dataset):
3     x = [row[0] for row in dataset]
4     y = [row[1] for row in dataset]
5     x_mean, y_mean = mean(x), mean(y)
6     b1 = covariance(x, x_mean, y, y_mean) / variance(x, x_mean)
7     b0 = y_mean - b1 * x_mean
8     return [b0, b1]
```

We can put this together with all of the functions from the previous two steps and test out the calculation.

```
1 # Calculate Coefficients
2
3 # Calculate the mean value of a list of numbers
4 def mean(values):
5     return sum(values) / float(len(values))
6
7 # Calculate covariance between x and y
8 def covariance(x, mean_x, y, mean_y):
9     covar = 0.0
10    for i in range(len(x)):
11        covar += (x[i] - mean_x) * (y[i] - mean_y)
12    return covar
13
14 # Calculate the variance of a list of numbers
15 def variance(values, mean):
16    return sum([(x-mean)**2 for x in values])
17
18 # Calculate coefficients
19 def coefficients(dataset):
20    x = [row[0] for row in dataset]
21    y = [row[1] for row in dataset]
22    x_mean, y_mean = mean(x), mean(y)
23    b1 = covariance(x, x_mean, y, y_mean) / variance(x, x_mean)
24    b0 = y_mean - b1 * x_mean
25    return [b0, b1]
26
```

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

```

27 # calculate coefficients
28 dataset = [[1, 1], [2, 3], [4, 3], [3, 2], [5, 5]]
29 b0, b1 = coefficients(dataset)
30 print('Coefficients: B0=%.3f, B1=%.3f' % (b0, b1))

```

Running this example calculates and prints the coefficients.

```

1 Coefficients: B0=0.400, B1=0.800

```

Now that we know how to estimate the coefficients, the next step is to use them.

4. Make Predictions

The simple linear regression model is a line defined by coefficients estimated from training data.

Once the coefficients are estimated, we can use them to make predictions.

The equation to make predictions with a simple linear regression model is as follows:

```

1 y = b0 + b1 * x

```

Below is a function named **simple_linear_regression()** that implements the prediction equation to use together the estimation of the coefficients on training data from the steps above.

The coefficients prepared from the training data are used to make predictions on the test data, which

```

1 def simple_linear_regression(train, test):
2     predictions = list()
3     b0, b1 = coefficients(train)
4     for row in test:
5         yhat = b0 + b1 * row[0]
6         predictions.append(yhat)
7     return predictions

```

Let's pull together everything we have learned and make predictions for our simple contrived dataset.

As part of this example, we will also add in a function to manage the evaluation of the predictions called **evaluate_algorithm()** and another function to estimate the Root Mean Squared Error of the predictions called **rmse_metric()**.

The full example is listed below.

Get Your Start in Machine Learning ×

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

```

1 # Standalone simple linear regression example
2 from math import sqrt
3
4 # Calculate root mean squared error
5 def rmse_metric(actual, predicted):
6     sum_error = 0.0
7     for i in range(len(actual)):
8         prediction_error = predicted[i] - actual[i]
9         sum_error += (prediction_error ** 2)
10    mean_error = sum_error / float(len(actual))
11    return sqrt(mean_error)
12
13 # Evaluate regression algorithm on training dataset
14 def evaluate_algorithm(dataset, algorithm):
15     test_set = list()
16     for row in dataset:
17         row_copy = list(row)
18         row_copy[-1] = None
19         test_set.append(row_copy)
20     predicted = algorithm(dataset, test_set)
21     print(predicted)
22     actual = [row[-1] for row in dataset]
23     rmse = rmse_metric(actual, predicted)
24     return rmse
25
26 # Calculate the mean value of a list of numbers
27 def mean(values):
28     return sum(values) / float(len(values))
29
30 # Calculate covariance between x and y
31 def covariance(x, mean_x, y, mean_y):
32     covar = 0.0
33     for i in range(len(x)):
34         covar += (x[i] - mean_x) * (y[i] - mean_y)
35     return covar
36
37 # Calculate the variance of a list of numbers
38 def variance(values, mean):
39     return sum([(x-mean)**2 for x in values])
40
41 # Calculate coefficients
42 def coefficients(dataset):
43     x = [row[0] for row in dataset]
44     y = [row[1] for row in dataset]
45     x_mean, y_mean = mean(x), mean(y)
46     b1 = covariance(x, x_mean, y, y_mean) / variance(x, x_mean)

```

Get Your Start in Machine Learning ×

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
47     b0 = y_mean - b1 * x_mean
48     return [b0, b1]
49
50 # Simple linear regression algorithm
51 def simple_linear_regression(train, test):
52     predictions = list()
53     b0, b1 = coefficients(train)
54     for row in test:
55         yhat = b0 + b1 * row[0]
56         predictions.append(yhat)
57     return predictions
58
59 # Test simple linear regression
60 dataset = [[1, 1], [2, 3], [4, 3], [3, 2], [5, 5]]
61 rmse = evaluate_algorithm(dataset, simple_linear_regression)
62 print('RMSE: %.3f' % (rmse))
```

Running this example displays the following output that first lists the predictions and the RMSE of the

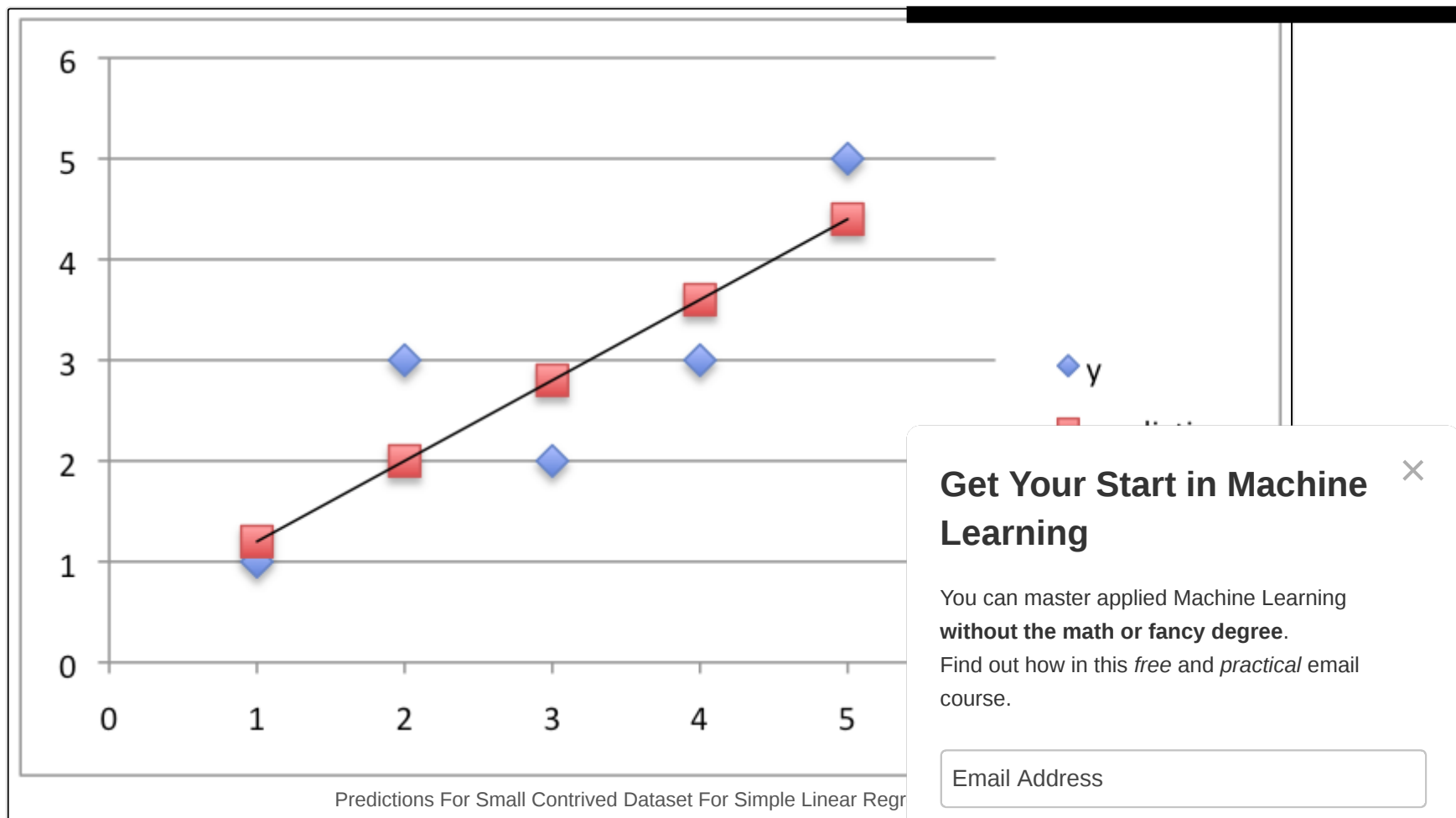
```
1 [1.1999999999999995, 1.9999999999999996, 3.5999999999999996, 2.8, 4.3999999999999995]
2 RMSE: 0.693
```

Finally, we can plot the predictions as a line and compare it to the original dataset.

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**. Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

5. Predict Insurance

We now know how to implement a simple linear regression model.

Let's apply it to the Swedish insurance dataset.

This section assumes that you have downloaded the dataset to the file **insurance.csv** and it is available in the current working directory.

We will add some convenience functions to the simple linear regression from the previous steps.

Get Your Start in Machine Learning

Specifically a function to load the CSV file called `load_csv()`, a function to convert a loaded dataset to numbers called `str_column_to_float()`, a function to evaluate an algorithm using a train and test set called `train_test_split()` a function to calculate RMSE called `rmse_metric()` and a function to evaluate an algorithm called `evaluate_algorithm()`.

The complete example is listed below.

A training dataset of 60% of the data is used to prepare the model and predictions are made on the remaining 40%.

```
1 # Simple Linear Regression on the Swedish Insurance Dataset
2 from random import seed
3 from random import randrange
4 from csv import reader
5 from math import sqrt
6
7 # Load a CSV file
8 def load_csv(filename):
9     dataset = list()
10    with open(filename, 'r') as file:
11        csv_reader = reader(file)
12        for row in csv_reader:
13            if not row:
14                continue
15            dataset.append(row)
16    return dataset
17
18 # Convert string column to float
19 def str_column_to_float(dataset, column):
20     for row in dataset:
21         row[column] = float(row[column].strip())
22
23 # Split a dataset into a train and test set
24 def train_test_split(dataset, split):
25     train = list()
26     train_size = split * len(dataset)
27     dataset_copy = list(dataset)
28     while len(train) < train_size:
29         index = randrange(len(dataset_copy))
30         train.append(dataset_copy.pop(index))
31     return train, dataset_copy
32
33 # Calculate root mean squared error
34 def rmse_metric(actual, predicted):
35     sum_error = 0.0
36     for i in range(len(actual)):
```

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

```

37     prediction_error = predicted[i] - actual[i]
38     sum_error += (prediction_error ** 2)
39     mean_error = sum_error / float(len(actual))
40     return sqrt(mean_error)
41
42 # Evaluate an algorithm using a train/test split
43 def evaluate_algorithm(dataset, algorithm, split, *args):
44     train, test = train_test_split(dataset, split)
45     test_set = list()
46     for row in test:
47         row_copy = list(row)
48         row_copy[-1] = None
49         test_set.append(row_copy)
50     predicted = algorithm(train, test_set, *args)
51     actual = [row[-1] for row in test]
52     rmse = rmse_metric(actual, predicted)
53     return rmse
54
55 # Calculate the mean value of a list of numbers
56 def mean(values):
57     return sum(values) / float(len(values))
58
59 # Calculate covariance between x and y
60 def covariance(x, mean_x, y, mean_y):
61     covar = 0.0
62     for i in range(len(x)):
63         covar += (x[i] - mean_x) * (y[i] - mean_y)
64     return covar
65
66 # Calculate the variance of a list of numbers
67 def variance(values, mean):
68     return sum([(x-mean)**2 for x in values])
69
70 # Calculate coefficients
71 def coefficients(dataset):
72     x = [row[0] for row in dataset]
73     y = [row[1] for row in dataset]
74     x_mean, y_mean = mean(x), mean(y)
75     b1 = covariance(x, x_mean, y, y_mean) / variance(x, x_mean)
76     b0 = y_mean - b1 * x_mean
77     return [b0, b1]
78
79 # Simple linear regression algorithm
80 def simple_linear_regression(train, test):
81     predictions = list()
82     b0, b1 = coefficients(train)
83     for row in test:

```

Get Your Start in Machine Learning ×

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning


```

84         yhat = b0 + b1 * row[0]
85         predictions.append(yhat)
86     return predictions
87
88 # Simple linear regression on insurance dataset
89 seed(1)
90 # load and prepare data
91 filename = 'insurance.csv'
92 dataset = load_csv(filename)
93 for i in range(len(dataset[0])):
94     str_column_to_float(dataset, i)
95 # evaluate algorithm
96 split = 0.6
97 rmse = evaluate_algorithm(dataset, simple_linear_regression, split)
98 print('RMSE: %.3f' % (rmse))

```

Running the algorithm prints the RMSE for the trained model on the training dataset.

A score of about 38 (thousands of Kronor) was achieved, which is much better than the Zero Rule algorithm (of Kronor) on the same problem.

```
1 RMSE: 38.339
```

Extensions

The best extension to this tutorial is to try out the algorithm on more problems.

Small datasets with just an input (x) and output (y) columns are popular for demonstration in statistics and are available online.

Seek out some more small datasets and make predictions using simple linear regression.

Did you apply simple linear regression to another dataset?

Share your experiences in the comments below.

Review

In this tutorial, you discovered how to implement the simple linear regression algorithm from scratch

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

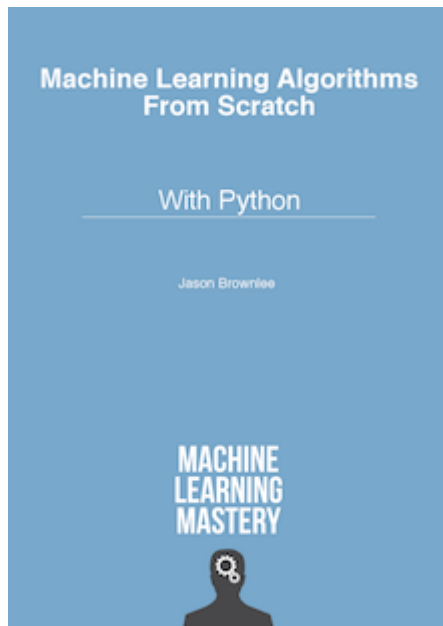
Specifically, you learned:

- How to estimate statistics from a training dataset like mean, variance and covariance.
- How to estimate model coefficients and use them to make predictions.
- How to use simple linear regression to make predictions on a real dataset.

Do you have any questions?

Ask your question in the comments below and I will do my best to answer.

Want to Code Algorithms in Python With



Code Your First Algorithm

...with step-by-step tutorials on r

Discover how in my new
[Machine Learning Algorithms](#)

It covers **18 tutorials** with all the code for
Linear Regression, k-Nearest Neighbors, Stochastic

Finally, Pull Back the Machine Learning A

Skip the Academics. Just Results.

[Click to learn more.](#)

Get Your Start in Machine Learning

You can master applied Machine Learning
without the math or fancy degree.
Find out how in this *free* and *practical* email
course.

START MY EMAIL COURSE

Get Your Start in Machine Learning



About Jason Brownlee

Dr. Jason Brownlee is a husband, proud father, academic researcher, author, professional developer and a machine learning practitioner. He is dedicated to helping developers get started and get good at applied machine learning. [Learn more.](#)

[View all posts by Jason Brownlee](#) →

< [How To Create an Algorithm Test Harness From Scratch With Python](#)

[How to Implement Linear Regression With Stochastic Gradient Descent From Scratch With Python](#) >

68 Responses to *How To Implement Simple Linear Regression From Scratch With Python*



Vineeth October 27, 2016 at 7:28 pm #

Hi Jason,

i have downloaded the csv file, but when i try to run the script against the file, i get the following error

” could not convert string to float: ‘X’ ”

this script stops at function def train_test_split(dataset, split)

can you confirm how your csv file is structured ?

Regards

Vineeth

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



Jason Brownlee October 28, 2016 at 9:08 am #

REPLY ↩

Get Your Start in Machine Learning

Sorry to hear that Vineeth.

Totally my error, do not include the column headers in the small contrived dataset. Delete the first row.

I will update the example.



En-wai October 30, 2016 at 7:58 am #

REPLY ↩

Hi Jason.....i have deleted the column headers X and Y along with all other descriptive info in the file but i kee getting this error:

" ValueError: could not convert string to float: i"

here are the first 5 values in my csv file after removing the white space(replacing it with commas)

108,392.5

19,46.2

13,15.7

124,422.2

40,119.4

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



Jason Brownlee October 30, 2016 at 8:59 am #

Your file looks perfect.

Confirm that you do not have any empty rows on the end of the file.



Adrian Moldovan October 27, 2016 at 9:20 pm #

REPLY ↩

This is brilliant!

Thanks for talking the time to go through all the steps and explain literally... everything.

Get Your Start in Machine Learning



Jason Brownlee October 28, 2016 at 9:09 am #

REPLY ↩

You're welcome Adrian, I'm glad you found it valuable.



Nelson Silva October 28, 2016 at 2:11 am #

REPLY ↩

Hello Jason,
great tutorial!
It would be great if you also provided the code for the respective plots in python!
Especially the plot for the dataset 😊
Thank you.



Jason Brownlee October 28, 2016 at 9:16 am #

Great suggestion Nelson, thanks.

I was aiming to keep the use of libs to a minimum (e.g. no matplotlib or seaborn).



Rahul Sharma June 13, 2017 at 5:46 am #

Hi Nelson, You can use pyplotlib library to create this kind of scatter plot:

Pls use this code to implement scatter plot:

```
import pyplotlib.pyplot as py
py.scatter(x_axis_value,y_axis_value,color='black')
py.show()
```

I hope this helps !

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.**
Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning



venkat dabbara October 28, 2016 at 4:58 am #

REPLY ↩

`predicted = algorithm(dataset, test_set)`

where is algorithm defined???



Jason Brownlee October 28, 2016 at 9:18 am #

REPLY ↩

Great question Venkat.

The “algorithm” argument in the `evaluate_algorithm()` function is a name of a function. We pass in the name of the function. This means that when we execute `algorithm()` to make predictions in `evaluate_algorithm()`, we are in fact calling the function.

I did this to separate algorithm evaluation from algorithm implementation, so that the same test harness can be used to evaluate different algorithms.

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**. Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



En-wai October 28, 2016 at 9:14 pm #

under section 2. Calculating covariance i think the two meaning there is not quiet a clear. Pls clarify.

“In fact, covariance is a generalization of correlation that is limited to two variables. Whereas correlation is a measure of the relationship between two variables.”????????



Jason Brownlee October 29, 2016 at 7:42 am #

REPLY ↩

Thanks En-wai, I have updated the language.

I was trying to comment on how covariance is an abstraction of correlation to go from 2 groups of numbers to more than 2 groups of numbers.



Ram October 29, 2016 at 1:06 am #

REPLY

Hi,

I got clear idea on linear regression. Thank You.

We do calculate linear regression with SciPi library as below.

```
regr = linear_model.LinearRegression()
```

```
regr.fit(X_train, y_train).
```

Please clarify whether all this calculation will happen behind the scenes when we call the above code.



Jason Brownlee October 29, 2016 at 9:25 am #

Hi Ram,

There are more efficient approaches to implement these algorithms using linear algebra. I expect this to happen behind the scenes.

Implementing algorithms is great for learning how they work, but it is not a good idea to use these from scratch.

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



Aliyu A. Aziz October 29, 2016 at 6:03 pm #

Hi Jason,

Many thanks for this easy to follow LR from scratch. I have noticed Line 9

```
file = open(filename, "rb")
```

is opening the file in text mode and causing the "Error: iterator should return strings, not bytes (did you open the file in text mode?)"

Changing 'rb' to 'rt' or 'r'

```
file = open(filename, "rt")
```

fixes the error.

Get Your Start in Machine Learning

Best regards



Jason Brownlee October 30, 2016 at 8:54 am #

REPLY ↩

Great, thanks Aliyu.

It does work on my platform, but I will make the example more portable.



saimadhu November 3, 2016 at 6:06 pm #

REPLY ↩

Hi,

Jason Brownlee

Thanks a lot for such an amazing post on simple linear regression. This post is the best tutorial to get the and I felt this post is the must read before learning the multi-regression analysis.

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



Jason Brownlee November 4, 2016 at 9:06 am #

Thanks saimadhu, I'm glad you found it useful.



Johnny December 13, 2016 at 4:45 am #

REPLY ↩

Another great one and I love these foundation ones. Also, you get right into the steps/meat of it and you do not leave out cosmetics – just wrap those up neatly at the end. Thank you sir.

I would like to see/study this same type of process for datasets pertaining to the basic types of business. Specifically, how to produce good dataset and properly frame up problem areas, for business. Do you recommend any books?

Get Your Start in Machine Learning



Jason Brownlee December 13, 2016 at 8:09 am #

REPLY ↩

Thanks Johnny.

Sorry, I don't know of good books like that. It is an empirical pursuit – more of a craft. The best education is practice.



Aslam March 12, 2017 at 4:39 am #

REPLY ↩

I am a beginner and found this very useful.

Thank you sir !



Jason Brownlee March 12, 2017 at 8:28 am #

I'm glad to hear it!



Girish March 25, 2017 at 2:58 am #

How do we plot the graph using code



Jason Brownlee March 25, 2017 at 7:39 am #

REPLY ↩

You can use matplotlib:

```
1 from matplotlib import pyplot
2 pyplot.plot(x, y)
3 pyplot.show()
```

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning



Nemanja April 2, 2017 at 5:33 am #

REPLY ↩

Hy, how can we plot a line of regression on our graph? And what we can do to reduce a rmse?Thanks



Jason Brownlee April 2, 2017 at 6:33 am #

REPLY ↩

You can evaluate the RMSE each epoch/iteration, save the RMSE values in an array and plot the array using matplotlib.



Sean April 3, 2017 at 4:36 am #

what is the relationship between `numpy.cov()` , `numpy.var()` methods and your `covariance()` , `var()` between the two.

Thanks



Abhishek April 28, 2017 at 3:25 pm #

Its a great article thankyou for helping us...



Jason Brownlee April 29, 2017 at 7:21 am #

REPLY ↩

Thanks Abhishek, I'm glad that you found it useful.



Nuwan C May 4, 2017 at 1:30 pm #

REPLY ↩

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

Hi Jason,

Thank you for another great tutorial.

What does the Zero Based algorithm do and why it use in her?

Thank you



Jason Brownlee May 5, 2017 at 7:28 am #

REPLY ↩

Do you mean the Zero Rule algorithm?

See this post for a description and worked example:

<http://machinelearningmastery.com/implement-baseline-machine-learning-algorithms-scratch-python/>



John David Kromkowski May 5, 2017 at 2:28 am #

Nice work

Maybe tiny typo:

$\text{covariance} = \sum((x(i) - \text{mean}(x)) * (y - \text{mean}(y)))$

should be

$\text{covariance} = \sum((x(i) - \text{mean}(x)) * (y(i) - \text{mean}(y)))$

You have it correct in the actual code

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



Jason Brownlee May 5, 2017 at 7:32 am #

REPLY ↩

Thanks John. Fixed.

Get Your Start in Machine Learning



Etienne May 20, 2017 at 6:35 pm #

REPLY

Good day Jason

My model is $y = b_0 + (b_1 * x) - (b_2 / (b_3 + x))$, which gives an asymptotic approach in a flocculation process. While I get a good data fit using the scipy curve_fit routine, I do not know how to get the leverage, the diagonal elements of the hat matrix H. Whereas in your model, the X system matrix would be formulated as:

$\hat{y} = H.y$

and H is $X(X^T.X)^{-1}.X^T$, where X^T is the transpose of X

In your model $X.\hat{b}$ would be:

$\begin{bmatrix} 1 & x_0 \end{bmatrix} \begin{bmatrix} b_0 \end{bmatrix}$

$\begin{bmatrix} 1 & x_1 \end{bmatrix} \begin{bmatrix} b_1 \end{bmatrix}$

$\begin{bmatrix} 1 & x_2 \end{bmatrix} .$

$\begin{bmatrix} 1 & x_3 \end{bmatrix}$

$\begin{bmatrix} \dots \end{bmatrix}$

But what would it be in my case?

Another problem is how to solve for H, so I can get the diagonal elements hii.

Any help would be greatly appreciated.

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



suguna May 24, 2017 at 4:45 am #

I removed columns header from csv file(Insurance CSV)

then I am getting this following error:

ValueError: could not convert string to float: female

Rahul Sharma June 13, 2017 at 5:47 am #

Get Your Start in Machine Learning



suguna , you need to remove all the empty cells in your csv, if any are present. That is what is causing this error



Rahul Sharma June 14, 2017 at 2:21 am #

REPLY ↩

Hi Jason,

As per the derivation : https://en.wikipedia.org/wiki/Standard_deviation

Variance = Avg $(xi - xMean)^2$

But here in algorithm you have used it as : $sum([(x-mean)**2 \text{ for } x \text{ in values}])$

which is not average but only some of squared difference. Is this some kind of modification?



Rahul Sharma June 15, 2017 at 10:12 pm #

Hi Jason. Can you please clarify this doubt.



Digvijay Rana June 15, 2017 at 5:10 am #

Thankyou very much Sir,

I had been looking for someplace to start implenting algos myself. This is best tutorial i have read by far.

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



Jason Brownlee June 15, 2017 at 8:53 am #

REPLY ↩

Thanks, I'm glad to hear that.

I have many right here:

<https://machinelearningmastery.com/machine-learning-algorithms-from-scratch/>



noname June 24, 2017 at 3:13 pm #

REPLY ↩

too late to board the ML bus ..Digvijay..



Jason Brownlee June 25, 2017 at 5:58 am #

REPLY ↩

Never too late.



Vaibhav June 17, 2017 at 11:08 pm #

Thanks a lot sir ! . Its a best description so far .



Jason Brownlee June 18, 2017 at 6:31 am #

I'm glad to hear it.



Kris July 6, 2017 at 11:20 pm #

I'm confused about your definition of covariance. Generally it's finally divided by $(n - 1)$ where n is the number of samples, where as there is no such operation carried out through out the code. Can you please clarify ?



Soumik Rakshit July 13, 2017 at 1:45 am #

REPLY ↩

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

I am unable to download the dataset as a csv file. Can someone please help me???



Jason Brownlee July 13, 2017 at 9:57 am #

REPLY ↩

Here is the raw file:

```
1 Auto Insurance in Sweden
2
3 In the following data
4 X = number of claims
5 Y = total payment for all the claims in thousands of Swedish Kronor
6 for geographical zones in Sweden
7 Reference: Swedish Committee on Analysis of Risk Premium in Motor Insurance
8 http://college.hmco.com/mathematics/brase/understandable\_statistics/7e/students/datasets/slr/frames/frame.html
9
10
11 X    Y
12 108  392,5
13 19   46,2
14 13   15,7
15 124  422,2
16 40   119,4
17 57   170,9
18 23   56,9
19 14   77,5
20 45   214
21 10   65,3
22 5    20,9
23 48   248,1
24 11   23,5
25 23   39,6
26 7    48,8
27 2    6,6
28 24   134,9
29 6    50,9
30 3    4,4
31 23   113
32 6    14,8
33 9    48,7
34 9    52,1
35 3    13,2
36 29   103,9
37 7    77,5
```

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
38 4 11,8
39 20 98,1
40 7 27,9
41 4 38,1
42 0 0
43 25 69,2
44 6 14,6
45 5 40,3
46 22 161,5
47 11 57,2
48 61 217,6
49 12 58,1
50 4 12,6
51 16 59,6
52 13 89,9
53 60 202,4
54 41 181,3
55 37 152,8
56 55 162,8
57 41 73,4
58 11 21,3
59 27 92,6
60 8 76,1
61 3 39,9
62 17 142,1
63 13 93
64 13 31,9
65 15 32,1
66 8 55,6
67 29 133,3
68 30 194,5
69 24 137,9
70 9 87,4
71 31 209,8
72 14 95,5
73 53 244,6
74 26 187,5
```

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

You will need to convert the “,” to “.” and replace the space between columns with “,”.



uma maheswari July 16, 2017 at 9:09 pm #

hi jason

REPLY ↩

Get Your Start in Machine Learning

can you tell how do we implement the linear regression on image dataset



Jason Brownlee July 17, 2017 at 8:46 am #

REPLY ↩

Perhaps linear regression is a bad fit image data.

Convolutional neural networks are very popular for image data:

<http://machinelearningmastery.com/crash-course-convolutional-neural-networks/>



Pierce Ng July 31, 2017 at 1:39 am #

Hi Jason,

Great stuff! Thanks for the exposition.

I implemented a no-shuffling version of `train_test_split` which always takes the first 38 entries as training program gives RMSE of 45.23.

Your RMSE of 38.339 is from the randomization in `train_test_split` with `seed(1)`. If I try with `seed(2)` then

What's the next step with different values of RMSE?

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



Jason Brownlee July 31, 2017 at 8:16 am #

This is the variance of the method.

Ideally, we would evaluate the algorithm multiple times and report the mean and standard deviation of the model.

Does that help?

Get Your Start in Machine Learning



Pierce Ng August 6, 2017 at 3:20 pm #

REPLY ↩

It does, thanks.

I ported your Python code to Pharo Smalltalk and wrote a blog post. See <http://www.samadhiweb.com/blog/2017.08.06.dataframe.html>.



Jason Brownlee August 7, 2017 at 8:39 am #

REPLY ↩

Very cool Pierce. Nice work!

I used to work with a dev who was a massive small talk fan.



Eoin Kenny August 11, 2017 at 3:14 am #

That is NOT the formula for variance... you're supposed to divide by n or $n-1$, what is going on?



Jason Brownlee August 11, 2017 at 6:43 am #

Might be population vs sample variance.



Tanmay September 1, 2017 at 2:42 am #

REPLY ↩

Hi Jason, why this `*args` parameter in the `evaluate_algorithm` function?

Jason Brownlee September 1, 2017 at 6:49 am #

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning



So we can provide a variable number parameters for the algorithm to the `evaluate_algorithm` function.

It is generic for different algorithms.



vedang September 15, 2017 at 2:19 am #

REPLY ↩

How do we predict the value of y, given x. Also how obtain the accuracy?



Jason Brownlee September 15, 2017 at 12:16 pm #

REPLY ↩

This is all of applied machine learning.

Perhaps you might be best starting with Weka:

<https://machinelearningmastery.com/start-here/#weka>

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



Barrys September 15, 2017 at 1:27 pm #

Hi Jason,

Thanks for your great articles. They are very helpful to me.

Something is not clear to me. Why we calculated RMSE and what it means exactly? I was expecting you test data as you did in your KNN post.

Thanks.



Jason Brownlee September 16, 2017 at 8:36 am #

REPLY ↩

RMSE is root mean squared error and is the error of the predictions in the same units as the

Get Your Start in Machine Learning

You cannot calculate accuracy for a regression problem. Accuracy refers to the percentage of correct label predictions out of all label predictions made. We do not predict a label in regression, we predict a quantity.



Barrys September 16, 2017 at 1:28 pm #

REPLY ↩

So we need to run another algorithm to predict its label? How can I use y value for a given x without knowing its label (that's linear regression as i understood)? What kind of problems does the linear regression solve?

It would be better to explain RMSE in the document and why we calculate it? it is mentioned but not explained.



Jason Brownlee September 17, 2017 at 5:25 am #

Linear regression is for problems where we want to predict a quantity, called regression.

If you have a dataset where you need to predict a label, you cannot use linear regression. You need a classification algorithm.

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



Rizvi October 11, 2017 at 9:23 pm #

Hi Jason,

This is an awesome post for beginners.

Currently, we are fitting a polynomial with 2 coefficients to the data. If we extend this approach to higher order polynomials, will that come under the scope of non-linear regression? Also, will increasing the polynomial order improve the estimation accuracy?



Jason Brownlee October 12, 2017 at 5:28 am #

REPLY ↩

It may improve accuracy, or it may over fit the data. Try it and see.

Get Your Start in Machine Learning

Use a robust test harness to ensure you do not trick yourself.



Melina October 17, 2017 at 12:08 am #

REPLY ↩

Great tutorial! Thank you 😊

I am preparing a demo for simple linear regression and I plan to show the code using sklearn and compare it to “own” regression algorithm code, tweaked version of yours!

I am stuck at one thing in your code and that is the variance formula/equation.

* You are using: $\text{Variance} = \text{Sum}((x - \text{mean}(x))^2)$

* Should it not be: $\text{Variance} = \text{Sum}((x - \text{mean}(x))^2) / N$

I am probably just confused so please correct me to the right thinking if I am.



Jason Brownlee October 17, 2017 at 5:49 am #

It is not normalized. See here for more information:

https://en.wikipedia.org/wiki/Simple_linear_regression

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Leave a Reply

Get Your Start in Machine Learning

Name (required)

Email (will not be published) (required)

Website

Welcome to Machine Learning Mastery



Hi, I'm Dr. Jason Brownlee.
My goal is to make practitioners like YOU awesome at applied machine learning.

[Read More](#)

Get Your Start in Machine Learning ×

You can master applied Machine Learning **without the math or fancy degree.**
Find out how in this *free* and *practical* email course.

Code Algorithms From Scratch in Python

Discover how to code top machine learning algorithms from first principles

[Get Your Start in Machine Learning](#)



Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

POPULAR



Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras
JULY 21, 2016



Your First Machine Learning Project in Python Step-By-Step
JUNE 10, 2016



Develop Your First Neural Network in Python With Keras Step-By-Step
MAY 24, 2016



Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras
JULY 26, 2016

How to Setup a Python Environment for Machine Learning and Deep Learning with Anaconda

Get Your Start in Machine Learning



MARCH 13, 2017



Time Series Forecasting with the Long Short-Term Memory Network in Python

APRIL 7, 2017



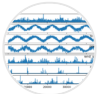
Multi-Class Classification Tutorial with the Keras Deep Learning Library

JUNE 2, 2016



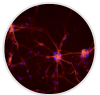
Regression Tutorial with the Keras Deep Learning Library in Python

JUNE 9, 2016



Multivariate Time Series Forecasting with LSTMs in Keras

AUGUST 14, 2017



How to Implement the Backpropagation Algorithm From Scratch In Python

NOVEMBER 7, 2016

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

[START MY EMAIL COURSE](#)

© 2017 Machine Learning Mastery. All Rights Reserved.

[Privacy](#) | [Contact](#) | [About](#)

Get Your Start in Machine Learning