Machine Learning Mastery
Making Developers Awesome at Machine Learning

How to Use Statistics to Identify Outliers in Data

by **Jason Brownlee** on April 25, 2018 in **Statistics**

Last Updated on August 8, 2019

When modeling, it is important to clean the data sample to ensure that the observations best represent the problem.

Sometimes a dataset can contain extreme values that are outside the range of what is expected and unlike the other data. These are called outliers and often machine learning modeling and model skill in general can be improved by understanding and even removing these outlier values.

In this tutorial, you will discover more about outliers and two statistical methods that you can use to identify and filter outliers from your dataset.

After completing this tutorial, you will know:

- That an outlier is an unlikely observation in a dataset and may have one of many causes.
- That standard deviation can be used to identify outliers in Gaussian or Gaussian-like data.
- That the interquartile range can be used to identify outliers in data regardless of the distribution.

Discover statistical hypothesis testing, resampling methods, estimation statistics and nonparametric methods in my new book, with 29 step-by-step tutorials and full source code.

Let’s get started.

- **Update May/2018**: Fixed bug when filtering samples via outlier limits. Thanks Yishai E and peter.

How to Use Statistics to Identify Outliers in Data
Photo by Jeff Richardson, some rights reserved.

Tutorial Overview

This tutorial is divided into 4 parts; they are:

1. What are Outliers?
2. Test Dataset
3. Standard Deviation Method
4. Interquartile Range Method

What are Outliers?

An outlier is an observation that is unlike the other observations.

It is rare, or distinct, or does not fit in some way.

Outliers can have many causes, such as:

- Measurement or input error.
- Data corruption.
- True outlier observation (e.g. Michael Jordan in basketball).

There is no precise way to define and identify outliers in general because of the specifics of each dataset. Instead, you, or a domain expert, must interpret the raw observations and decide whether a value is an outlier or not.

Nevertheless, we can use statistical methods to identify observations that appear to be rare or unlikely given the available data.

This does not mean that the values identified are outliers and should be removed. But, the tools described in this tutorial can be helpful in shedding light on rare events that may require a second look.

A good tip is to consider plotting the identified outlier values, perhaps in the context of non-outlier values to see if there are any systematic relationship or pattern to the outliers. If there is, perhaps they are not outliers and can be explained, or perhaps the outliers themselves can be identified more systematically.

Test Dataset

Before we look at outlier identification methods, let’s define a dataset we can use to test the methods.

We will generate a population 10,000 random numbers drawn from a Gaussian distribution with a mean of 50 and a standard deviation of 5.

Numbers drawn from a Gaussian distribution will have outliers. That is, by virtue of the distribution itself, there will be a few values that will be a long way from the mean, rare values that we can identify as outliers.

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.

```
# generate gaussian data
from numpy.random import seed
from numpy.random import randn
from numpy import mean
from numpy import std
# seed the random number generator
seed(1)
# generate univariate observations
data = 5 * randn(10000) + 50
# summarize
print('mean=%.3f stdv=%.3f' % (mean(data), std(data)))
```

Running the example generates the sample and then prints the mean and standard deviation. As expected, the values are very close to the expected values.

```
mean=50.049 stdv=4.994
```

Standard Deviation Method

If we know that the distribution of values in the sample is Gaussian or Gaussian-like, we can use the standard deviation of the sample as a cut-off for identifying outliers.

The Gaussian distribution has the property that the standard deviation from the mean can be used to reliably summarize the percentage of values in the sample.

For example, within one standard deviation of the mean will cover 68% of the data.

So, if the mean is 50 and the standard deviation is 5, as in the test dataset above, then all data in the sample between 45 and 55 will account for about 68% of the data sample. We can cover more of the data sample if we expand the range as follows:

- 1 Standard Deviation from the Mean: 68%
- 2 Standard Deviations from the Mean: 95%
- 3 Standard Deviations from the Mean: 99.7%

A value that falls outside of 3 standard deviations is part of the distribution, but it is an unlikely or rare event at approximately 1 in 370 samples.

Three standard deviations from the mean is a common cut-off in practice for identifying outliers in a Gaussian or Gaussian-like distribution. For smaller samples of data, perhaps a value of 2 standard deviations (95%) can be used, and for larger samples, perhaps a value of 4 standard deviations (99.9%) can be used.

Let's make this concrete with a worked example.

Sometimes, the data is standardized first (e.g. to a Z-score with zero mean and unit variance) so that the outlier detection can be performed using standard Z-score cut-off values. This is a convenience and is not required in general, and we will perform the calculations in the original scale of the data here to make things clear.

We can calculate the mean and standard deviation of a given sample, then calculate the cut-off for identifying outliers as more than 3 standard deviations from the mean.

```
# calculate summary statistics
data_mean, data_std = mean(data), std(data)
# identify outliers
cut_off = data_std * 3
lower, upper = data_mean - cut_off, data_mean + cut_off
```

We can then identify outliers as those examples that fall outside of the defined lower and upper limits.

```
# identify outliers
outliers = [x for x in data if x < lower or x > upper]
```

Alternately, we can filter out those values from the sample that are not within the defined limits.

```
# remove outliers
outliers_removed = [x for x in data if x > lower and x < upper]
```

We can put this all together with our sample dataset prepared in the previous section.

The complete example is listed below.

```
# identify outliers with standard deviation
from numpy.random import seed
from numpy.random import randn
from numpy import mean
from numpy import std
# seed the random number generator
seed(1)
# generate univariate observations
data = 5 * randn(10000) + 50
# calculate summary statistics
data_mean, data_std = mean(data), std(data)
```

```
# identify outliers
cut_off = data_std * 3
lower, upper = data_mean - cut_off, data_mean + cut_off
# identify outliers
outliers = [x for x in data if x < lower or x > upper]
print('Identified outliers: %d' % len(outliers))
# remove outliers
outliers_removed = [x for x in data if x >= lower and x <= upper]
print('Non-outlier observations: %d' % len(outliers_removed))
```

Running the example will first print the number of identified outliers and then the number of observations that are not outliers, demonstrating how to identify and filter out outliers respectively.

```
Identified outliers: 29
Non-outlier observations: 9971
```

So far we have only talked about univariate data with a Gaussian distribution, e.g. a single variable. You can use the same approach if you have multivariate data, e.g. data with multiple variables, each with a different Gaussian distribution.

You can imagine bounds in two dimensions that would define an ellipse if you have two variables. Observations that fall outside of the ellipse would be considered outliers. In three dimensions, this would be an ellipsoid, and so on into higher dimensions.

Alternately, if you knew more about the domain, perhaps an outlier may be identified by exceeding the limits on one or a subset of the data dimensions.

Interquartile Range Method

Not all data is normal or normal enough to treat it as being drawn from a Gaussian distribution.

A good statistic for summarizing a non-Gaussian distribution sample of data is the Interquartile Range, or IQR for short.

The IQR is calculated as the difference between the 75th and the 25th percentiles of the data and defines the box in a box and whisker plot.

Remember that percentiles can be calculated by sorting the observations and selecting values at specific indices. The 50th percentile is the middle value, or the average of the two middle values for an even number of examples. If we had 10,000 samples, then the 50th percentile would be the average of the 5000th and 5001st values.

We refer to the percentiles as quartiles ("*quart*" meaning 4) because the data is divided into four groups via the 25th, 50th and 75th values.

The IQR defines the middle 50% of the data, or the body of the data.

The IQR can be used to identify outliers by defining limits on the sample values that are a factor *k* of the IQR below the 25th percentile or above the 75th percentile. The common value for the factor *k* is the value 1.5. A factor k of 3 or more can be used to identify values that are extreme outliers or “*far outs*” when described in the context of box and whisker plots.

On a box and whisker plot, these limits are drawn as fences on the whiskers (or the lines) that are drawn from the box. Values that fall outside of these values are drawn as dots.

We can calculate the percentiles of a dataset using the *percentile()* NumPy function that takes the dataset and specification of the desired percentile. The IQR can then be calculated as the difference between the 75th and 25th percentiles.

```
# calculate interquartile range
q25, q75 = percentile(data, 25), percentile(data, 75)
iqr = q75 - q25
```

We can then calculate the cutoff for outliers as 1.5 times the IQR and subtract this cut-off from the 25th percentile and add it to the 75th percentile to give the actual limits on the data.

```
# calculate the outlier cutoff
cut_off = iqr * 1.5
lower, upper = q25 - cut_off, q75 + cut_off
```

We can then use these limits to identify the outlier values.

```
# identify outliers
outliers = [x for x in data if x < lower or x > upper]
```

We can also use the limits to filter out the outliers from the dataset.

```
outliers_removed = [x for x in data if x > lower and x < upper]
```

We can tie all of this together and demonstrate the procedure on the test dataset.

The complete example is listed below.

```
# identify outliers with interquartile range
from numpy.random import seed
from numpy.random import randn
from numpy import percentile
# seed the random number generator
seed(1)
# generate univariate observations
data = 5 * randn(10000) + 50
# calculate interquartile range
q25, q75 = percentile(data, 25), percentile(data, 75)
iqr = q75 - q25
print('Percentiles: 25th=%.3f, 75th=%.3f, IQR=%.3f' % (q25, q75, iqr))
# calculate the outlier cutoff
```

```
cut_off = iqr * 1.5
lower, upper = q25 - cut_off, q75 + cut_off
# identify outliers
outliers = [x for x in data if x < lower or x > upper]
print('Identified outliers: %d' % len(outliers))
# remove outliers
outliers_removed = [x for x in data if x >= lower and x <= upper]
print('Non-outlier observations: %d' % len(outliers_removed))
```

Running the example first prints the identified 25th and 75th percentiles and the calculated IQR. The number of outliers identified is printed followed by the number of non-outlier observations.

```
Percentiles: 25th=46.685, 75th=53.359, IQR=6.674
Identified outliers: 81
Non-outlier observations: 9919
```

The approach can be used for multivariate data by calculating the limits on each variable in the dataset in turn, and taking outliers as observations that fall outside of the rectangle or hyper-rectangle.

Extensions

This section lists some ideas for extending the tutorial that you may wish to explore.

- Develop your own Gaussian test dataset and plot the outliers and non-outlier values on a histogram.
- Test out the IQR based method on a univariate dataset generated with a non-Gaussian distribution.
- Choose one method and create a function that will filter out outliers for a given dataset with an arbitrary number of dimensions.

If you explore any of these extensions, I'd love to know.

Further Reading

This section provides more resources on the topic if you are looking to go deeper.

Posts

- How to Identify Outliers in your Data

API

- seed() NumPy API
- randn() NumPy API
- mean() NumPy API

- std() NumPy API
- percentile() NumPy API

Articles

- Outlier on Wikipedia
- Anomaly detection on Wikipedia
- 68–95–99.7 rule on Wikipedia
- Interquartile range
- Box plot on Wikipedia

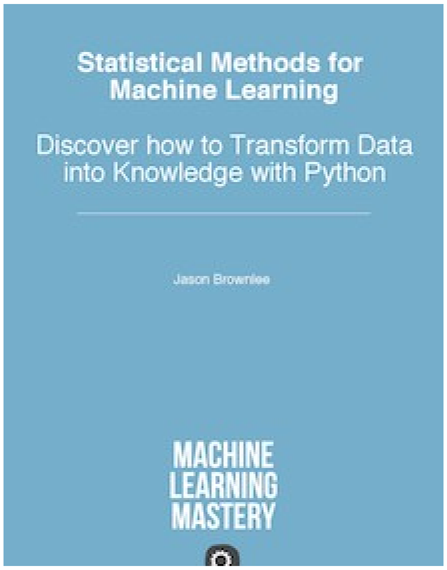
Summary

In this tutorial, you discovered outliers and two statistical methods that you can use to identify and filter outliers from your dataset.

Specifically, you learned:

- That an outlier is an unlikely observation in a dataset and may have one of many causes.
- That standard deviation can be used to identify outliers in Gaussian or Gaussian-like data.
- That the interquartile range can be used to identify outliers in data regardless of the distribution.

Do you have any questions?
Ask your questions in the comments below and I will do my best to answer.



Statistical Methods for Machine Learning

Discover how to Transform Data into Knowledge with Python

Jason Brownlee

MACHINE LEARNING MASTERY

Tweet Share Share

About Jason Brownlee

Jason Brownlee, PhD is a machine learning specialist who teaches developers how to get results with modern machine learning methods via hands-on tutorials.

View all posts by Jason Brownlee →

‹ Introduction to Random Number Generators for Machine Learning in Python

How to Calculate Correlation Between Variables in Python ›

64 Responses to *How to Use Statistics to Identify Outliers in Data*

Nitin Panwar April 25, 2018 at 5:05 pm # REPLY

Nicely explained. very well done.

Jason Brownlee April 26, 2018 at 6:21 am # REPLY

Thanks.

Haneesh June 27, 2019 at 2:32 am # REPLY

Hello, can you explain me in R, how to find out how many outliers exists in one variable using Q1-1.5*IQR & Q3+1.5*IQR. Please help me on this only in R as I’m new to analysis.

Jason Brownlee June 27, 2019 at 7:58 am # REPLY

Sorry, I don't have an example of this in R.

Marish April 26, 2018 at 11:06 am #

REPLY

Very helpful. Thank you

Jason Brownlee April 26, 2018 at 3:02 pm #

REPLY

I'm glad to hear that.

talha anwar April 27, 2018 at 3:48 am #

REPLY

Once i remove the outlier, how can i fill the space left by that outlier. Becuase in other features the length is more than in outlier removed features

Jason Brownlee April 27, 2018 at 6:09 am #

REPLY

The entire record could be removed.

Alternately, the value in the record could be removed, and then imputed: https://machinelearningmastery.com/handle-missing-data-python/

Sourav Maharana April 27, 2018 at 5:16 am #

REPLY

Jason's Brownlee articles and content are amazing as always

Jason Brownlee April 27, 2018 at 6:15 am #

REPLY

Thanks!

Vishesh sharma April 27, 2018 at 5:35 am #

REPLY

If suppose we have 50 features and we run this for each of the features then then the won’t the number of rows I would have to delete be a lot because of missing data?

Also would dbscan be preferable to this?

Jason Brownlee April 27, 2018 at 6:16 am #

REPLY ↩

It may be. It is a very simple/rough method, perhaps not suitable for large numbers of features.

Alternately, obs could be deleted and the missing values imputed.

jimmy April 27, 2018 at 10:20 am #

REPLY ↩

I liked your post, I think would be better with plotting.

Jason Brownlee April 27, 2018 at 2:27 pm #

REPLY ↩

Thanks for the suggestion.

Yishai E April 29, 2018 at 12:26 am #

REPLY ↩

Your code has a flaw – especially for the quantile example, which define the outlier borders based on data points from the dataset. If your outliers are >< from the border and your non-outliers are , then your borders are missing from both sets.

Jason Brownlee April 29, 2018 at 6:28 am #

REPLY ↩

What do you mean exactly, can you give a concrete example?

peter May 7, 2018 at 6:00 am #

REPLY ↩

I assume Yishai means that we need to add a ‘>=` and ‘<=' in the code to include samples that are equal to upper/lower.

Mukund April 30, 2018 at 11:38 pm #

REPLY ↰

Hi Dr.Jason.
Thanks for all the tips and I have been following your posts for a long time.

I don’t know, if this is the right forum to ask my following question. I am trying to evaluate various classifier algorithims, like decision tree , ADtree etc for a particular problem of detecting whether a candidate is Autistic or not, using very well known interview questionnaire ADI-R. Various literature claim to use A algorithim or B algorithim to show how they could use reduce the question sets (original 99 questions) and yet achieve great accuracy. Many literature state Adtree is best for this purposes. Yet, Adtree has its own limitation. I am confused. Could you kindly, explain what is the best way to proceed, given the complexity of this problem.

Jason Brownlee May 1, 2018 at 5:34 am #

REPLY ↰

What is the problem exactly?

Brad Smith May 16, 2018 at 5:43 am #

REPLY ↰

I’ve been thinking about the Standard Deviation method, and how some people have suggested that a very large outlier could skew the mean and standard deviation enough to interfere with outlier removal.

However, couldn’t this problem be mitigated by comparing each value to bounds that come from the mean and standard deviation of all the *other* values (leaving out the one value that you’re currently on in the list)? If the one value that you’re currently looking at is an outlier, then it will be left out of the mean and standard deviation calculations, making it much more likely to be deemed an outlier, even if it is a very large value.

This method may have a cost when it comes to efficiency, but the cost may be worth it depending on the application. Thanks!

Jason Brownlee May 16, 2018 at 6:10 am #

REPLY ↰

It might be easier to visually inspect plots of the data prior to calculating limits to ensure they make sense.

Kevin Arvai May 25, 2018 at 5:05 am # REPLY ↰

Thank you for the post, Jason. It inspired me to write a Kaggle kernel exploring the topic in more detail. I implemented your standard deviation and IQR methods 🙂
https://www.kaggle.com/kevinarvai/outlier-detection-practice-uni-multivariate

Jason Brownlee May 25, 2018 at 9:32 am # REPLY ↰

Well done! That is a very impressive kernel Kevin.

Tobi Adeyemi May 31, 2018 at 1:45 am # REPLY ↰

Hi Jason, are these methods covered in your new text; Statistical Methods for Machine Learning?

Jason Brownlee May 31, 2018 at 6:21 am # REPLY ↰

They are the methods I think you need to know how to use when working through an applied machine learning project.

Bhukya Neeharika June 5, 2018 at 8:02 pm # REPLY ↰

Respected sir,
i have issue with drawing boxplot in python using iqr method,where i know median,minimum,maximum,q1,q3.could you please him sir.

Jason Brownlee June 6, 2018 at 6:40 am # REPLY ↰

Perhaps the API will help:
https://matplotlib.org/api/_as_gen/matplotlib.pyplot.boxplot.html

Bhukya Neeharika June 6, 2018 at 6:06 pm # REPLY ↰

Respected sir,
i couldn’t understand that.could you please explain me in detail

Aman August 7, 2018 at 3:13 am # REPLY

Hi Jason,

I have data where the standard deviation is very close to the mean. So when I do the :
lower = mean – cutoff
it gives me a negative number. Is this alright? My data does not contain values less than 0.

Jason Brownlee August 7, 2018 at 6:33 am # REPLY

Perhaps this methods is not suitable for your data?

Matheus September 21, 2018 at 3:02 am # REPLY

Hi Jason,

When you say that the data needs to be standardized first, are you referring to data transformation (Normalization, StandartScaler, Box-cox)?

Jason Brownlee September 21, 2018 at 6:31 am # REPLY

Standardization explicitly, zero mean and unit standard deviation.

Felix October 10, 2018 at 1:16 am # REPLY

Hi Jason,
thank your for your expertise!

I get the following TypeError using your IRM code:

TypeError Traceback (most recent call last)
in ()
9 lower, upper = q25 – cutoff, q75 + cutoff
10 # identify outliers

```
—> 11 outliers = [x for x in dfg if x upper]
12 print('Identified outliers: %d' % len(outliers))
13 #remove outliers

in (.0)
9 lower, upper = q25 – cutoff, q75 + cutoff
10 # identify outliers
—> 11 outliers = [x for x in dfg if x upper]
12 print('Identified outliers: %d' % len(outliers))
13 #remove outliers

TypeError: '>' not supported between instances of 'numpy.ndarray' and 'str'
```

Jason Brownlee October 10, 2018 at 6:12 am # REPLY ↩

Sorry to hear that, I have some suggestions for you here:
https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me

Ravinder Ahuja November 12, 2018 at 6:55 am # REPLY ↩

Can you please put a post for replacing outlier with median using python..

Thanks

Jason Brownlee November 12, 2018 at 2:06 pm # REPLY ↩

Thanks for the suggestion.

Samuel November 25, 2018 at 4:25 pm # REPLY ↩

Thanks a lot, it is helpful.

Jason Brownlee November 26, 2018 at 6:15 am # REPLY ↩

I'm happy to hear that.

Srinivasa Rao Raghupatruni December 18, 2018 at 10:20 pm # REPLY ↩

Hi Jason,

Thank you for the wonderful article. I have implemented the above for my dataset. But when doing train_test split, I'm getting the below error:

ValueError: Found input variables with inconsistent numbers of samples: [459, 489].

Please suggest how to resolve the unequal shapes

Jason Brownlee December 19, 2018 at 6:34 am # REPLY ↩

I'm sorry to hear that ,I have some suggestions here:
https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me

Bijoy January 25, 2019 at 5:30 pm # REPLY ↩

Hi Jason,
Thanks for the wonderful work that you have been doing. I have just started working on ML to solve some problems we have.
Recently I have been trying to use ML to detect problems in machines(like motors) based on the vibration data collected from them. This will be a time series data. When the machine starts wearing out, the vibration data starts spiking. So ideally, the data would be all healthy and as the machine runs over a period of time, the vibration data would slowly start changing. The ML algo should find these deviations as they happen. What would you recommend to solve this kind of scenario? Statistical methods, ARIMA, NN ….Thanks in advance.

Jason Brownlee January 26, 2019 at 6:08 am # REPLY ↩

I recommend looking into "change detection" algorithms.

dongliang February 16, 2019 at 9:58 am # REPLY ↩

Very clear introduction to outliers and practical codes. Thanks

Jason Brownlee February 17, 2019 at 6:29 am # REPLY ↩

Thanks, I’m glad it helped.

Harriet April 8, 2019 at 8:01 pm # REPLY ↩

Hi,
How would you justify using the interquartile range method over other methods to identify outliers? I.e, does it have any particular strengths, and in what circumstances would we use it over others?
Thanks

Jason Brownlee April 9, 2019 at 6:23 am # REPLY ↩

It is simple and well understood.

Other methods may be complex and poorly understood.

Disha April 9, 2019 at 1:41 pm # REPLY ↩

Hi,
Can you please tell which method to choose – Z score or IQR for removing outliers from a dataset.
Dataset is a likert 5 scale data with around 30 features and 800 samples and I am trying to cluster the data in groups.
If I calculate Z score then around 30 rows come out having outliers whereas 60 outlier rows with IQR.
I am confused as which one to prefer.
Thanks.

Jason Brownlee April 9, 2019 at 2:43 pm # REPLY ↩

Perhaps try a suite of values, evaluate their effect on the data and choose a value that result in the desired effect.

You might want to plot the results, e.g outliers vs non-outliers.

Anna May 20, 2019 at 2:07 am #

REPLY

Hi,

Is it usefull to use this method when I only have 6 datapoints?
or what is the minimum I need?

Thank you!

Jason Brownlee May 20, 2019 at 6:33 am #

REPLY

Probably not. At least 30 points.

Srinivasa V June 15, 2019 at 4:43 pm #

REPLY

Well Explained!

In this case, we have removed the outliers
suppose we want to replace outliers with NAN how to do this

Could you explain the same

Thanks in Advance

Jason Brownlee June 16, 2019 at 7:11 am #

REPLY

You can get the indexes of the outlier values and set the values at those indexes to anything you wish, such as NaN.

You can also use the replace() function, I give an example here:
https://machinelearningmastery.com/handle-missing-data-python/

Artur October 20, 2019 at 5:12 am #

REPLY

Hello Jason,

as I understand, with
“outliers = [x for x in data if x upper]”
we get a list of the outlier-values (NOT the index).
Suppose that we have a multivariable DataFrame, how do we get the position of the outliers?
Meaning: Can we get a list with the indices of the outliers, so that we just drop them?

Many thanks for your interesting article!

Artur

Jason Brownlee October 20, 2019 at 6:25 am #

REPLY ↩

Perhaps use the where() numpy function?

Artur October 23, 2019 at 2:41 am #

REPLY ↩

Thx Jason,
thouht so too, but so far the np.where() func only gives me the position of outliers in my first column of interest. Maybe there is a problem with my loop..

But I figured an alternative 🙂

Jason Brownlee October 23, 2019 at 6:54 am #

REPLY ↩

Happy to hear that.

Kreecha November 12, 2019 at 11:40 am #

REPLY ↩

Do you have a reference for this of your statement: “ A good statistic for summarizing a non-Gaussian distribution sample of data is the Interquartile Range, or IQR for short.”. Personally, I am not sure IQR is suitable for all non Gaussian, but I would like to learn more if you can provide a reference.

Anyway thank for writing good articles.

Jason Brownlee November 12, 2019 at 2:06 pm #

REPLY ↩

It's a heuristic more than a rule, e.g. not in all cases.

Any good book on stats will describe this method.

Also see this:
https://en.wikipedia.org/wiki/Interquartile_range#Outliers

Shreya February 12, 2020 at 2:18 am #

REPLY

Can box plot or histogram be applied to find ouliers on whole dataset i.e. every data columns present in the dataset as a whole or we will have to apply it on every single column ?

Jason Brownlee February 12, 2020 at 5:50 am #

REPLY

Yes, but it is applied one column at a time.

Shreya February 18, 2020 at 2:24 am #

REPLY

Thank You !

Are there any ways in which instead of removing the outliers, we could replace them with some values so that the shape of our dataset will not be changed ?

Jason Brownlee February 18, 2020 at 6:22 am #

REPLY

Perhaps, but why?

Shreya February 18, 2020 at 3:48 pm #

Because if we remove the outliers then the number of data in all the columns of the dataset would be different which could create difficulty in training the model.

Jason Brownlee February 19, 2020 at 7:56 am #

Test and confirm.

Leave a Reply

Name (required)

Email (will not be published) (required)

Website

Welcome!

My name is *Jason Brownlee* PhD, and I **help developers** get results with **machine learning**.

Read more

Never miss a tutorial:

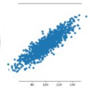
Picked for you:

A Gentle Introduction to k-fold Cross-Validation

Start Machine Learning

A Gentle Introduction to Normality Tests in Python

Statistical Significance Tests for Comparing Machine Learning Algorithms



How to Calculate Correlation Between Variables in Python

Statistics for Machine Learning (7-Day Mini-Course)