

Data Cleaning

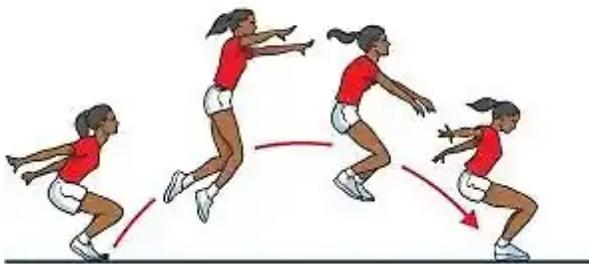
Section 3: Missing Data, Outlier Detection, and Principal Component Analysis (PCA)

LESSON 6: OUTLIER DETECTION

Lesson 6: Z-Scores

In statistics, there are times in which you transform data onto a different scale. If you have ever converted something like feet into meters, then you have performed a transformation. There is a particular transformation in statistics called a z-score. You will learn more about this transformation in the next course. For now, think of this value as indicating how extreme a value is. Typical values range from -3 to $+3$.

Here is an analogy of its use. Imagine you wanted to come up with a system for measuring when a room is unusually large, yet you do not have a measuring stick. Here is one way to do it. Imagine being in the middle of a typical room. You decide to measure the size of a room by how far you can jump before reaching a wall.



Girl jumping

Data Cleaning

Section 3: Missing Data, Outlier Detection, and Principal Component Analysis (PCA)



LESSON 6: OUTLIER DETECTION

If you think of a z-score as an average jump, then the idea for using this to detect outliers is as follows: Anything more than three jumps is unusual (i.e., outlier). Or another way of saying that: Any z-score less than -3 or bigger than $+3$ is considered unusual (i.e., outlier).

In the next course, you will learn how the z-score is calculated. For now, you will use technology to do this conversion.

Converting values into z-scores is as simple as using the scale function in R. In the following example, you will load the dplyr library to enable easy z-score transformations. You will also load the data sets library to load in some traditional data that you can experiment with. Assign the traditional "iris" data set to the name "data" and then create a new variable "ZScore_Sepal_Length" of the original "Sepal.Length" variable using the "scale" command. Finally, print out the original variable as well as the z-score. Based on the data displayed, none of these lengths appear to be outliers.>/p>



Data Cleaning

Section 3: Missing Data, Outlier Detection, and Principal Component Analysis (PCA)



LESSON 6: OUTLIER DETECTION

```
# R Output
Sepal.Length ZScore_Sepal_Length
1  5.1  -0.8976739
2  4.9  -1.1392005
3  4.7  -1.3807271
4  4.6  -1.5014904
5  5.0  -1.0184372
6  5.4  -0.5353840
```

R Code to determine Z-scores

[Download the R code and output \(opens new tab\)](#)

In Python, importing the SciPy package will be helpful for calculating the z-score and other normalizations. If you choose to not use this package, you can create your own function in Python with the functions provided by NumPy and pandas. You can see below where the SciPy package has a function to calculate the z-score simply titled `zscore`. You can see that these look like what was found with R. Differences can be noted to how they calculate the standard deviation in each language, with a minor



Data Cleaning

Section 3: Missing Data, Outlier Detection, and Principal Component Analysis (PCA)



LESSON 6: OUTLIER DETECTION

```
iris_df = pd.read_csv('C:/Users/Bernie/Desktop/iris.csv')
```

```
In [3]: iris_df.head()
```

```
Out[3]:
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
In [4]: iris_df['Zscore_Sepal_Length']=stats.zscore(iris_df.iloc[:,0])
```

```
In [5]: iris_df[['Sepal.Length', 'Zscore_Sepal_Length']].head()
```

```
Out[5]:
```

	Sepal.Length	Zscore_Sepal_Length
0	5.1	-0.900681
1	4.9	-1.143017
2	4.7	-1.385353
3	4.6	-1.506521
4	5.0	-1.021849

Using Python to import a dataset and calculate z-scores.

[PREVIOUS](#)[NEXT](#)