learn-co-curriculum / dsc-standard-normal-distribution-lab Public

View license

0 stars 179 forks

Star Watch

Code Issues 1 Pull requests Actions Projects Security Insights

solution

This branch is 10 commits ahead, 9 commits behind master.

lmcm18 fixed learning goals ... on Oct 28, 2019 14

View code

README.md

The Standard Normal Distribution - Lab

Introduction

In the previous lesson, you learned about the formula of the z-score, and looked at a few toy examples to explain an observation's standard score for normally distributed data. In this lab, you'll practice by standardizing and visualize some normal distributions.

Objectives

You will be able to:

- Calculate and interpret the z-score (standard score) for an observation from normally distributed data
- Visualize data before and after standardization to visually inspect the results

Let's get started

> A z-score can help identify how many standard deviations above or below the mean a certain observation is. Every time you obtain a z-score, use "above" or "below" in your phrasing.

The yields of apple trees in an orchard have been recorded in the file `yield.csv`. Each observation is recorded by weighing apples from trees (in pounds) and adding their weights. There are 5000 observations in total for this data.

Load, visualize and give general comments about the dataset

Use pandas for loading and inspecting the data.

```
# Import libraries
import numpy as np
import seaborn as sns
import pandas as pd

# Read the yield data as a dataframe
df = pd.read_csv('yield.csv')
df.head()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

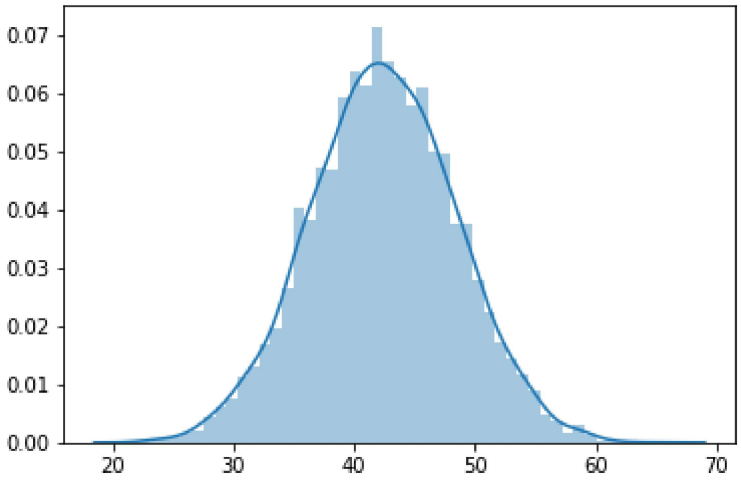
```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	0
0	39.741234
1	39.872055
2	44.331164

	0
3	46.600623
4	40.694984

```
sns.distplot(df);
```




```
# Your comments about the data here

# The data is normally distributed as shown by the density curve
```

Briefly explain what each value represents in this data set

```
# Your answer here

# Each value represents the yield from a single tree in terms of total weight of app
# that were obtained from this tree
```

Define the interval bounds which contain 99% of the observations

> **Hint**: Recall the empirical rule related to 3σ.

```
# Perform any calculations necessary here
mean = df.mean()
```

```
sd = df.std()
mean,sd
```

```
(0    42.407624
 dtype: float64, 0    6.003697
 dtype: float64)
```

```
# Write your answer here

# the mean value is 42.4 and the standard deviation is around 6
# 68% of tree yields have weight between (42.4 - 6) 36.4 and (42.4 - 6) 48.4 pounds;
# 95% between 30.4 and 54.4;
# Almost all between 24.4 and 60.4 pounds
```

Compute and interpret the z-score for a tree yielding 35 pounds of apples

```
# Calculate z
z = (35 - mean)/sd
z
```

```
0   -1.233844
dtype: float64
```

```
# Interpret the result

# This tree’s yield is 1.23 standard deviations below the mean yield.
```

Suppose a tree has a z-score of 1.85. Interpret this z-score. What is the yield of this tree?

```
# Interpret the z score

# This tree’s yield is 1.85 standard deviations above the mean
```

```
X = mean + 1.85*sd
X
```

```
0    53.514462
dtype: float64
```

```
# What is the yield ?

# Yield of this tree is 53.5 pounds.
```

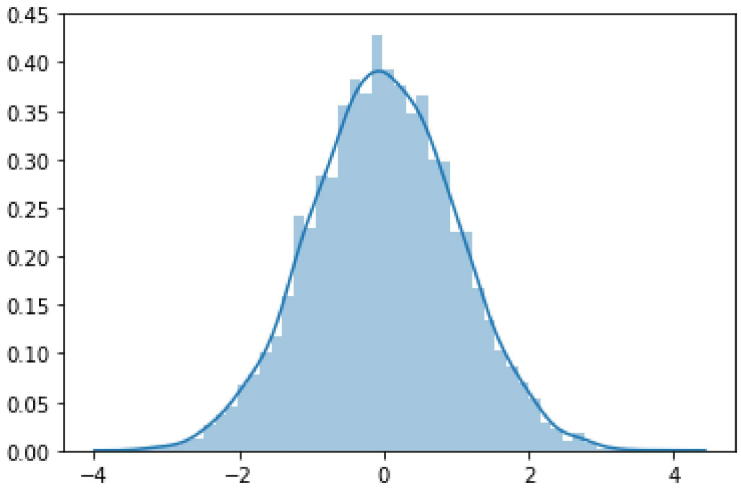
Convert each tree's yield to a z-score so the new variable is the "z-score for weight"

The units are still the apple trees. For the data set of all z-scores:

- What is the shape?
- The mean?
- The standard deviation?

```
z_data = [(x - df['0'].mean())/df['0'].std() for x in df['0']]
sns.distplot(z_data)
mean = np.mean(np.array(z_data))
sd = np.std((np.array(z_data)))
print ('Mean:', round(mean,2))
print ('SD:', round(sd,2))
```

```
Mean: 0.0
SD: 1.0
```




```
# Your observations

# It is a standard normal distribution
# Mean is 0 (it is a very small figure that rounds off to 0)
# SD is 1
# This is obvious because we standardised the whole distribution
```

Summary

In this lab, you practiced your knowledge of the standard normal distribution!

Releases

No releases published

Packages

No packages published

Contributors 6

Languages

- **Jupyter Notebook** 100.0%