index

January 11, 2022

1 The Standard Normal Distribution - Lab

1.1 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.

1.2 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

1.3 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.

1.4 Load, visualize and give general comments about the dataset

Use pandas for loading and inspecting the data.

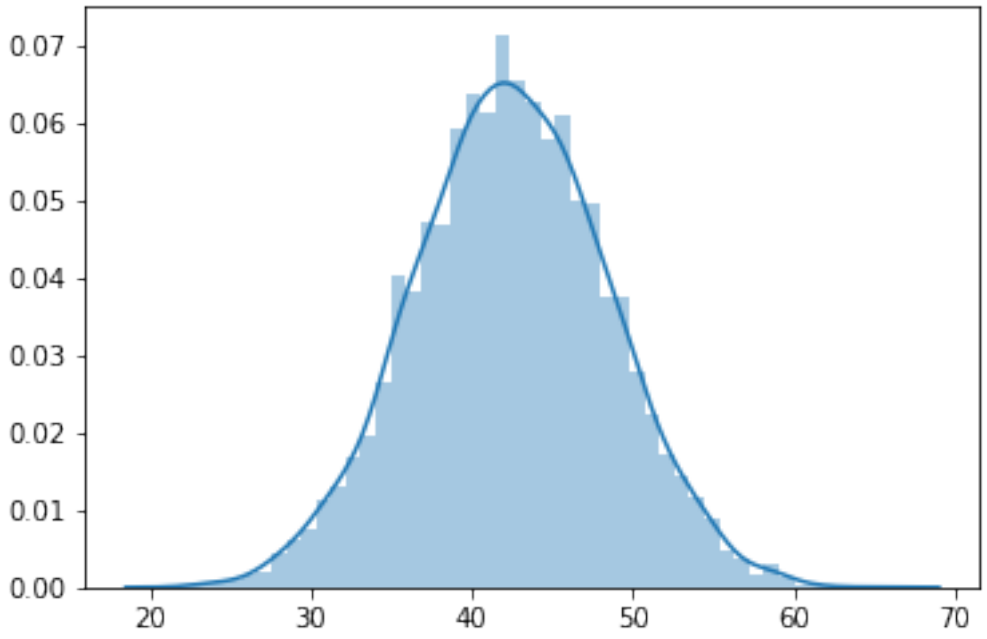
[22]:
```
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter

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

```
df.columns
```

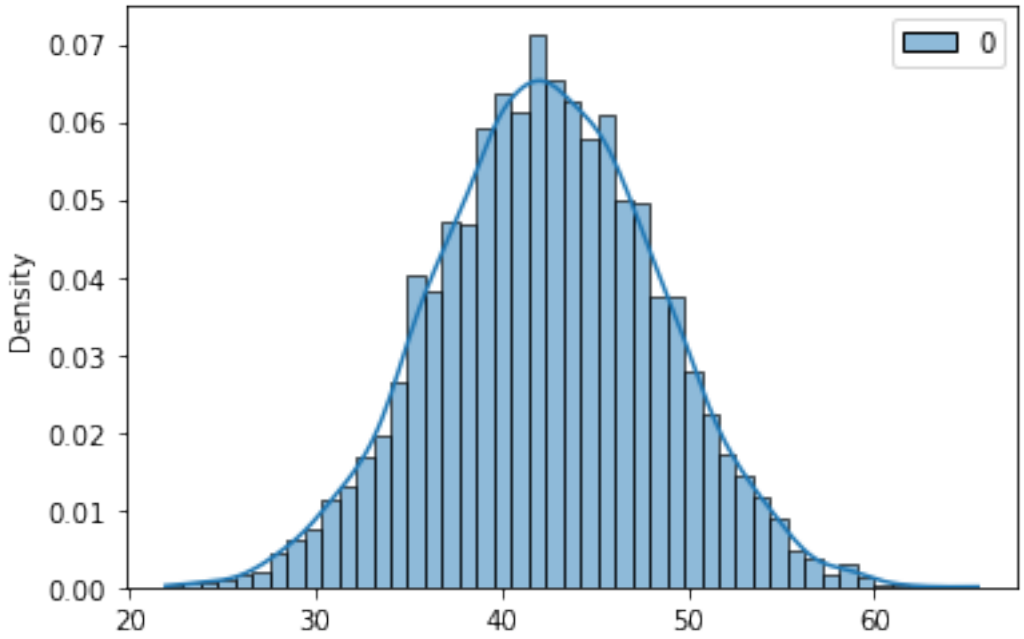
[22]:
```
Index(['0'], dtype='object')
```

[7]:
```
# Create a plot
sns.distplot(df);
```

```
/opt/anaconda3/envs/learn-env/lib/python3.8/site-
packages/seaborn/distributions.py:2551: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



[11]:
```
## Or by using histplot

sns.histplot(df, stat = "density", kde = True);
```



[12]:
```
# Your comments about the data here
# It can be seen that the data is normally distributed
```

1.5 Briefly explain what each value represents in this data set

[35]:
```
# Your answer here
df.agg(["mean", "std"])
mean = df.mean()
sigma = df.std()
mean
# # Or
# df.mean()
# df.std()
```

[35]:
```
0    42.407624
dtype: float64
```

1.6 Define the interval bounds which contain 99% of the observations

Hint: Recall the empirical rule related to 3σ.

[31]:
```
# Perform any calculations necessary here
maxi = list(df.mean() + 3*df.std())[0]
mini = list(df.mean() - 3*df.std())[0]
print(maxi)
```

```
print(mini)

df_3sigma = df[(df["0"] >= mini) & (df["0"] <= maxi)]
df_3sigma
```

```
60.418713372301596
24.39653434322378
```

[31]:

	0
0	39.741234
1	39.872055
2	44.331164
3	46.600623
4	40.694984
…	…
4995	39.037750
4996	51.861086
4997	36.441352
4998	42.549280
4999	34.798407

[4990 rows x 1 columns]

[ ]:
```
# Write your answer here
# 3 sigma means that around 99 % of the data is in the interval
# (mu - 3*sigma, mu + 3*sigma). In this problem, the number of data falling in
# this interval is 4990.
```

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

[36]:
```
# Calculate z
z = (35 - mean) / (sigma)
z
```

[36]:
```
0   -1.233844
dtype: float64
```

[8]:
```
# Interpret the result
# 35 is -1.233844 standard deviation under the mean value
```

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

[9]:
```
# Interpret the z score
# z scores give the position of a data in the standardized normal curve.
# so z = 1.85 means that data is 1.85 standard deviation away from mean value
```

```
# and since it is positive, we conclude that the data is greater than zero
# which is the mean value of the standardized normal curve.
```

[39]:
```
# Calculate yield
z = 1.85
x = mean + z*sigma
x
```

[39]:
```
0    53.514462
dtype: float64
```

[11]:
```
# What is the yield ?
# z = 1.85 gives x = 53.51 and this value is 1.85 standard deviation away from
# the mean valu of the data frame.
```

1.9 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?

[43]:
```
# Give your solution here
df["standard"] = (df["0"] - df["0"].mean()) / df["0"].std()
df
```

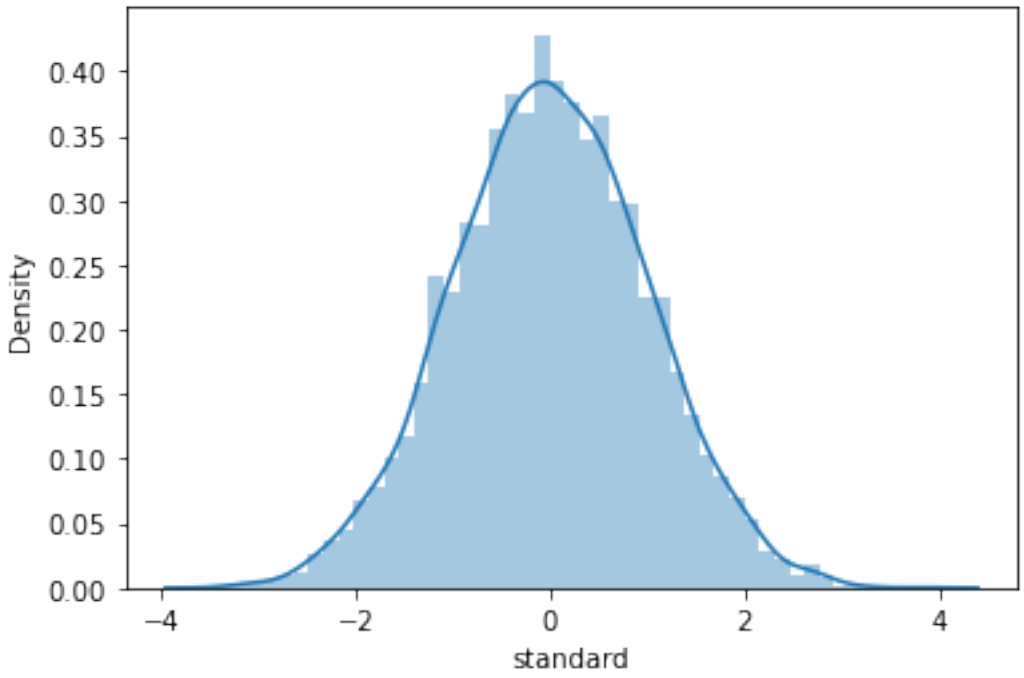
[43]:
```
              0  standard
0     39.741234 -0.444125
1     39.872055 -0.422335
2     44.331164  0.320393
3     46.600623  0.698403
4     40.694984 -0.285264
...         ...       ...
4995  39.037750 -0.561300
4996  51.861086  1.574607
4997  36.441352 -0.993766
4998  42.549280  0.023595
4999  34.798407 -1.267422

[5000 rows x 2 columns]
```

[45]:
```
# Your observations
df["standard"].agg(["mean", "std"])
sns.distplot(df["standard"]);
```

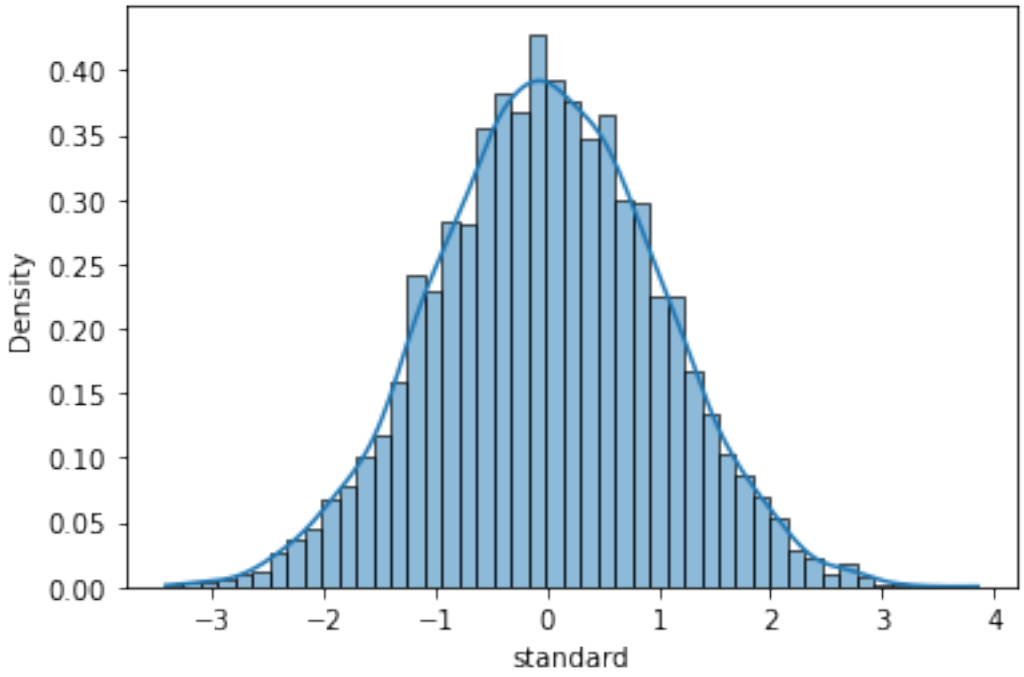
```
/opt/anaconda3/envs/learn-env/lib/python3.8/site-
```

```
packages/seaborn/distributions.py:2551: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

[45]: <AxesSubplot:xlabel='standard', ylabel='Density'>



[46]:
```
## Or

sns.histplot(df["standard"], stat = "density", kde = True);
```

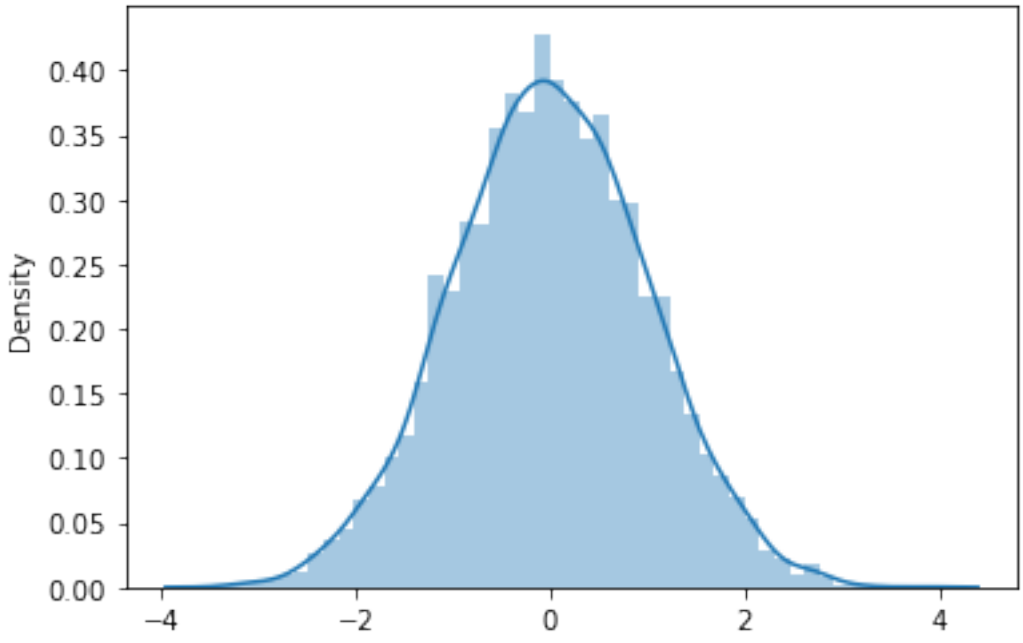


[48]:
```
# This is a standard normal curve with mean and standard deviation equal to
# 0 and 1 respectively.
```

[49]:
```
## From GitHub Solution

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))
```

```
/opt/anaconda3/envs/learn-env/lib/python3.8/site-
packages/seaborn/distributions.py:2551: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

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



1.10 Summary

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