

 dhavalsays outliers z score

 History

 1 contributor

1149 lines (1149 sloc) | 58.8 KB



Outlier detection and removal using z-score and standard deviation in python pandas

```
In [162]: import pandas as pd
import matplotlib
from matplotlib import pyplot as plt
%matplotlib inline
matplotlib.rcParams['figure.figsize'] = (10,6)
```

We are going to use heights dataset from kaggle.com. Dataset has heights and weights both but I have removed weights to make it simple

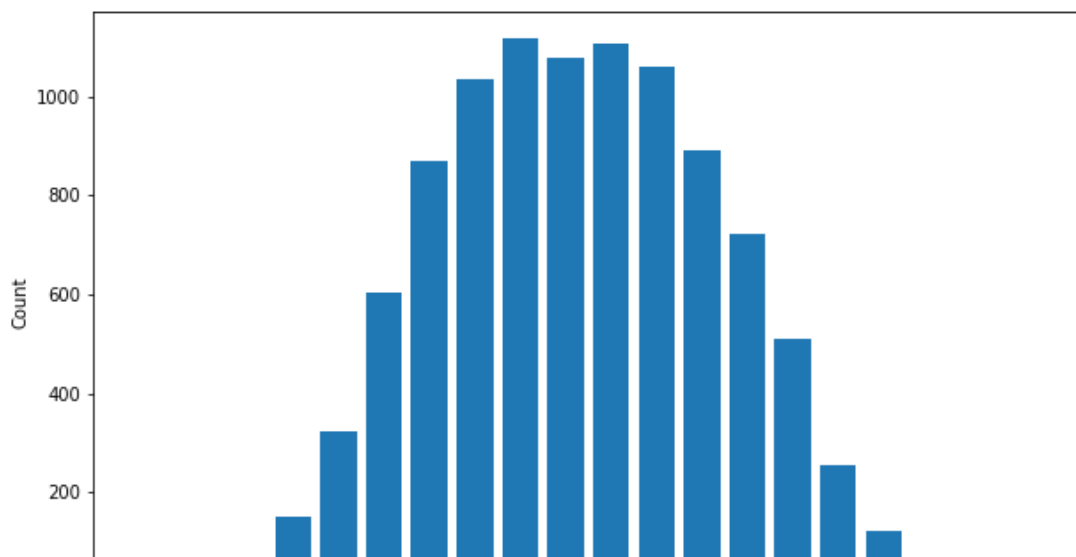
<https://www.kaggle.com/mustafaali96/weight-height> (<https://www.kaggle.com/mustafaali96/weight-height>)

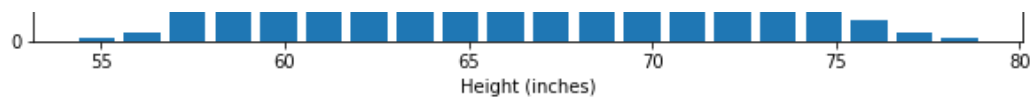
```
In [163]: df = pd.read_csv("heights.csv")
df.sample(5)
```

Out[163]:

	gender	height
1987	Male	65.478267
4478	Male	65.566101
5800	Female	66.258258
6054	Female	65.476903
2383	Male	71.505206

```
In [164]: plt.hist(df.height, bins=20, rwidth=0.8)
plt.xlabel('Height (inches)')
plt.ylabel('Count')
plt.show()
```





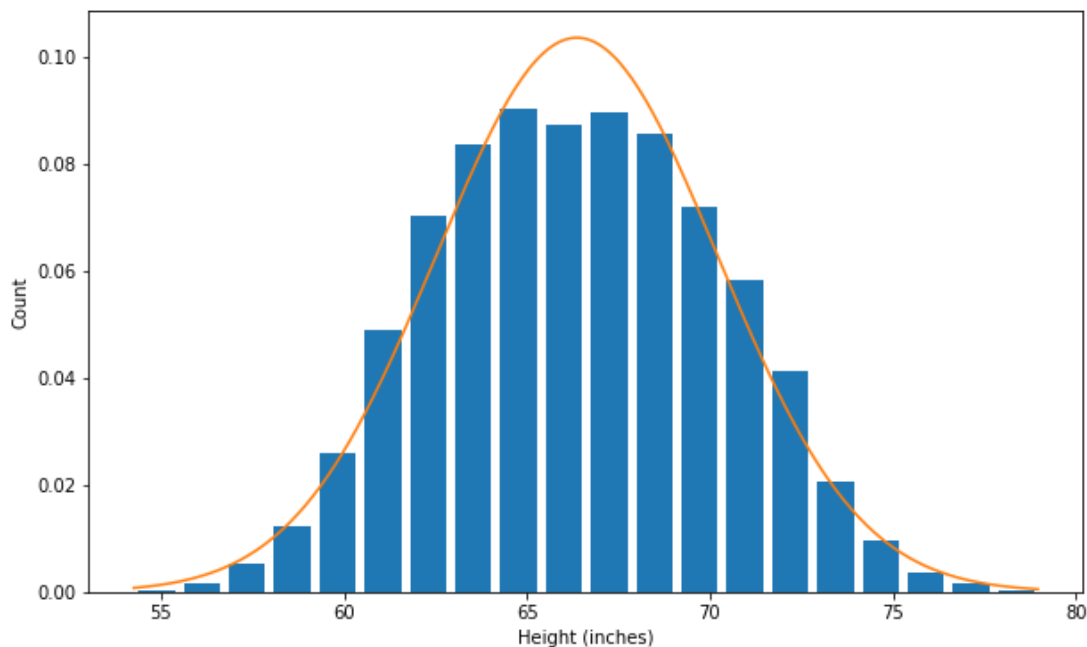
Read this awesome article to get your fundamentals clear on normal distribution, bell curve and standard deviation. <https://www.mathsisfun.com/data/standard-normal-distribution.html> (<https://www.mathsisfun.com/data/standard-normal-distribution.html>)

Plot bell curve along with histogram for our dataset

```
In [184]: from scipy.stats import norm
import numpy as np
plt.hist(df.height, bins=20, rwidth=0.8, density=True)
plt.xlabel('Height (inches)')
plt.ylabel('Count')

rng = np.arange(df.height.min(), df.height.max(), 0.1)
plt.plot(rng, norm.pdf(rng, df.height.mean(), df.height.std()))
```

Out[184]: [



```
In [168]: df.height.mean()
```

Out[168]: 66.3675597548656

```
In [169]: df.height.std()
```

Out[169]: 3.847528120795573

Here the mean is 66.37 and standard deviation is 3.84.

(1) Outlier detection and removal using 3 standard deviation

One of the ways we can remove outliers is remove any data points that are beyond **3 standard deviation** from mean. Which means we can come up with following upper and lower bounds

```
In [170]: upper_limit = df.height.mean() + 3*df.height.std()
          upper_limit
```

```
Out[170]: 77.91014411725232
```

```
In [171]: lower_limit = df.height.mean() - 3*df.height.std()
          lower_limit
```

```
Out[171]: 54.824975392478876
```

Here are the outliers that are beyond 3 std dev from mean

```
In [172]: df[(df.height>upper_limit) | (df.height<lower_limit)]
```

```
Out[172]:
```

	gender	height
994	Male	78.095867
1317	Male	78.462053
2014	Male	78.998742
3285	Male	78.528210
3757	Male	78.621374
6624	Female	54.616858
9285	Female	54.263133

Above the heights on higher end is **78 inch** which is around **6 ft 6 inch**. Now that is quite unusual height. There are people who have this height but it is very uncommon and it is ok if you remove those data points. Similarly on lower end it is **54 inch** which is around **4 ft 6 inch**. While this is also a legitimate height you don't find many people having this height so it is safe to consider both of these cases as outliers

Now remove these outliers and generate new dataframe

```
In [173]: df_no_outlier_std_dev = df[(df.height<upper_limit) & (df.height>lower_1
          df_no_outlier_std_dev.head()
```

```
Out[173]:
```

	gender	height
0	Male	73.847017
1	Male	68.781904
2	Male	74.110105
3	Male	71.730978

4	Male	69.881796
---	------	-----------

In [174]: `df_no_outlier_std_dev.shape`

Out[174]: (9993, 2)

In [175]: `df.shape`

Out[175]: (10000, 2)

Above shows original dataframe data 10000 data points. Out of that we removed 7 outliers (i.e. 10000-9993)

(2) Outlier detection and removal using Z Score

Z score is a way to achieve same thing that we did above in part (1)

Z score indicates how many standard deviation away a data point is.

For example in our case mean is 66.37 and standard deviation is 3.84.

If a value of a data point is 77.91 then Z score for that is 3 because it is 3 standard deviation away
(77.91 = 66.37 + 3 * 3.84)

Calculate the Z Score

$$Z = \frac{X - \mu}{\sigma}$$

μ = mean

σ = standard deviation

In [176]: `df['zscore'] = (df.height - df.height.mean()) / df.height.std()
df.head(5)`

Out[176]:

	gender	height	zscore
0	Male	73.847017	1.943964
1	Male	68.781904	0.627505
2	Male	74.110105	2.012343

3	Male	71.730978	1.393991
4	Male	69.881796	0.913375

Above for first record with height 73.84, z score is 1.94. This means 73.84 is 1.94 standard deviation away from mean

In [177]: $(73.84 - 66.37) / 3.84$

Out[177]: 1.9453124999999998

Get data points that has z score higher than 3 or lower than -3. Another way of saying same thing is get data points that are more than 3 standard deviation away

In [178]: `df[df['zscore']>3]`

Out[178]:

	gender	height	zscore
994	Male	78.095867	3.048271
1317	Male	78.462053	3.143445
2014	Male	78.998742	3.282934
3285	Male	78.528210	3.160640
3757	Male	78.621374	3.184854

In [179]: `df[df['zscore']<-3]`

Out[179]:

	gender	height	zscore
6624	Female	54.616858	-3.054091
9285	Female	54.263133	-3.146027

Here is the list of all outliers

In [180]: `df[(df.zscore<-3) | (df.zscore>3)]`

Out[180]:

	gender	height	zscore
994	Male	78.095867	3.048271
1317	Male	78.462053	3.143445
2014	Male	78.998742	3.282934
3285	Male	78.528210	3.160640
3757	Male	78.621374	3.184854
6624	Female	54.616858	-3.054091

9285	Female	54.263133	-3.146027
------	--------	-----------	-----------

Remove the outliers and produce new dataframe

```
In [181]: df_no_outliers = df[(df.zscore>-3) & (df.zscore<3)]  
df_no_outliers.head()
```

```
Out[181]:
```

	gender	height	zscore
0	Male	73.847017	1.943964
1	Male	68.781904	0.627505
2	Male	74.110105	2.012343
3	Male	71.730978	1.393991
4	Male	69.881796	0.913375

```
In [182]: df_no_outliers.shape
```

```
Out[182]: (9992, 3)
```