Properties of Normal Distribution

What we did:
In last class we learned about the bell curve - normal distribution and how to plot it.

In this class we learned about the properties of the normal distribution..

How we did it:

1. We got the data by rolling the dice 1000 times

```
#Creating a list of sum of 2 dice, rolled 1000 times
dice_result = []
for i in range(0, 1000):
    dice1 = random.randint(1, 6)
    dice2 = random.randint(1, 6)
    dice_result.append(dice1 + dice2)
```

2. We then calculated the mean,median and mode of the data using the statistics library.

```
import statistics
```

```
#Calculating the mean and the standard deviation
mean = sum(dice_result) / len(dice_result)
std_deviation = statistics.stdev(dice_result)
```

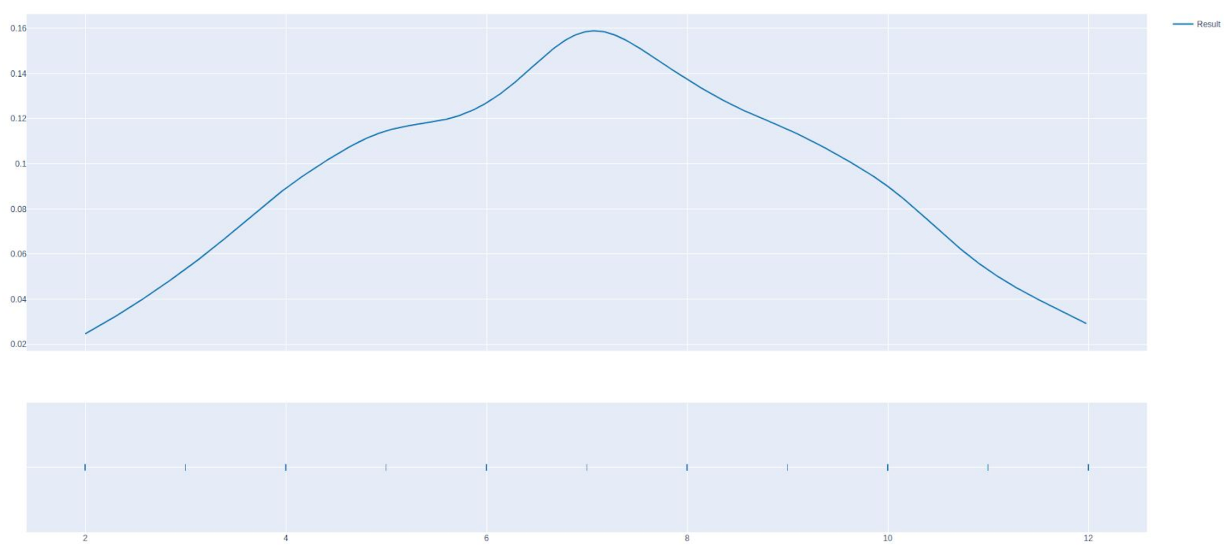
```
median = statistics.median(dice_result)
mode = statistics.mode(dice_result)
```

```
Mean of this data is 7.006
```

```
Median of this data is 7.0
Mode of this data is 7
```

4. Then we plotted the normal distribution plot.

```
fig = ff.create_distplot([dice_result], ["Result"], show_hist=False)
fig.show()
```



5. We coded to find the standard deviation for the results.

```
std_deviation = statistics.stdev(dice_result)
```

```
Standard deviation of this data is 2.314150230925229
```

6. Then we found the count of data points between mean - sd and mean + sd and it's percentage for first , second and third deviation respectively.

```
#Finding 1 standard deviation stard and end values, and 2 standard deviations stard and end values
first_std_deviation_start, first_std_deviation_end = mean-std_deviation, mean+std_deviation
second_std_deviation_start, second_std_deviation_end = mean-(2*std_deviation), mean+(2*std_deviation)
```

```
_3_std_deviation = [result for result in dice_result if result > third_std_deviation_start and result < third_std_deviation_end]
```

```
print("{}% of data lies within 1 standard deviation".format(len(list_of_data_within_1_std_deviation)*100.0/len(dice_result)))
```

```
print("{}% of data lies within 2 standard deviations".format(len(list_of_data_within_2_std_deviation)*100.0/len(dice_result)))
```

```
print("{}% of data lies within 3 standard deviations".format(len(list_of_data_within_3_std_deviation)*100.0/len(dice_result)))
```

```
68.7% of data lies within 1 standard deviation
95.5% of data lies within 2 standard deviations
100.0% of data lies within 3 standard deviations
```

We found that
99% of data lie between mean - 3sd and mean + 3sd
95% of data lie between mean -2sd and mean + 2sd
68% of data lie between mean - sd and mean + sd

7.We calculated the same for the second set of data to check if it’s true for all types of data.
8. We used the height and weight data and found mean ,median and mode.

```
import pandas as pd
import statistics
import csv
df = pd.read_csv("height-weight.csv")
height_list = df["Height(Inches)"].to_list()
weight_list = df["Weight(Pounds)"].to_list()
#Mean for height and Weight
height_mean = statistics.mean(height_list)
weight_mean = statistics.mean(weight_list)
#Median for height and weight
height_median = statistics.median(height_list)
weight_median = statistics.median(weight_list)
#Mode for height and weight
height_mode = statistics.mode(height_list)
weight_mode = statistics.mode(weight_list)
#Printing mean, median and mode to validate
print("Mean, Median and Mode of height is {}, {} and {} respectively".format(height_mean, height_median, height_mode))
print("Mean, Median and Mode of weight is {}, {} and {} respectively".format(weight_mean, weight_median, weight_mode))
```

9. We calculated the % of data points that lie in 1st,2nd and 3rd deviation.

```python
#1, 2 and 3 Standard Deviations for height
height_first_std_deviation_start, height_first_std_deviation_end = height_mean-height_std_deviation, height_mean+height_std_deviation
height_second_std_deviation_start, height_second_std_deviation_end = height_mean-(2*height_std_deviation), height_mean+(2*height_std_deviation)
height_third_std_deviation_start, height_third_std_deviation_end = height_mean-(3*height_std_deviation), height_mean+(3*height_std_deviation)
#1, 2 and 3 Standard Deviations for weight
weight_first_std_deviation_start, weight_first_std_deviation_end = weight_mean-weight_std_deviation, weight_mean+weight_std_deviation
weight_second_std_deviation_start, weight_second_std_deviation_end = weight_mean-(2*weight_std_deviation), weight_mean+(2*weight_std_deviation)
weight_third_std_deviation_start, weight_third_std_deviation_end = weight_mean-(3*weight_std_deviation), weight_mean+(3*weight_std_deviation)
#Percentage of data within 1, 2 and 3 Standard Deviations for Height
height_list_of_data_within_1_std_deviation = [result for result in height_list if result > height_first_std_deviation_start and result < height
height_list_of_data_within_2_std_deviation = [result for result in height_list if result > height_second_std_deviation_start and result < heigh
height_list_of_data_within_3_std_deviation = [result for result in height_list if result > height_third_std_deviation_start and result < height
#Percentage of data within 1, 2 and 3 Standard Deviations for Weight
weight_list_of_data_within_1_std_deviation = [result for result in weight_list if result > weight_first_std_deviation_start and result < weight
weight_list_of_data_within_2_std_deviation = [result for result in weight_list if result > weight_second_std_deviation_start and result < weigh
weight_list_of_data_within_3_std_deviation = [result for result in weight_list if result > weight_third_std_deviation_start and result < weight
#Printing data for height and weight (Standard Deviation)
print("{}% of data for height lies within 1 standard deviation".format(len(height_list_of_data_within_1_std_deviation)*100.0/len(height_list)))
print("{}% of data for height lies within 2 standard deviations".format(len(height_list_of_data_within_2_std_deviation)*100.0/len(height_list)))
print("{}% of data for height lies within 3 standard deviations".format(len(height_list_of_data_within_3_std_deviation)*100.0/len(height_list)))
print("{}% of data for weight lies within 1 standard deviation".format(len(weight_list_of_data_within_1_std_deviation)*100.0/len(weight_list)))
print("{}% of data for weight lies within 2 standard deviations".format(len(weight_list_of_data_within_2_std_deviation)*100.0/len(weight_list)))
print("{}% of data for weight lies within 3 standard deviations".format(len(weight_list_of_data_within_3_std_deviation)*100.0/len(weight_list)))
```

```
68.356% of data for height lies within 1 standard deviation
```

```
68.52% of data for weight lies within 1 standard deviation
```

2nd deviation.

```python
#1, 2 and 3 Standard Deviations for height
height_first_std_deviation_start, height_first_std_deviation_end = height_mean-height_std_deviation, height_mean+height_std_deviation
height_second_std_deviation_start, height_second_std_deviation_end = height_mean-(2*height_std_deviation), height_mean+(2*height_std_deviation)
height_third_std_deviation_start, height_third_std_deviation_end = height_mean-(3*height_std_deviation), height_mean+(3*height_std_deviation)
#1, 2 and 3 Standard Deviations for weight
weight_first_std_deviation_start, weight_first_std_deviation_end = weight_mean-weight_std_deviation, weight_mean+weight_std_deviation
weight_second_std_deviation_start, weight_second_std_deviation_end = weight_mean-(2*weight_std_deviation), weight_mean+(2*weight_std_deviation)
weight_third_std_deviation_start, weight_third_std_deviation_end = weight_mean-(3*weight_std_deviation), weight_mean+(3*weight_std_deviation)
#Percentage of data within 1, 2 and 3 Standard Deviations for Height
height_list_of_data_within_1_std_deviation = [result for result in height_list if result > height_first_std_deviation_start and result < height
height_list_of_data_within_2_std_deviation = [result for result in height_list if result > height_second_std_deviation_start and result < heigh
height_list_of_data_within_3_std_deviation = [result for result in height_list if result > height_third_std_deviation_start and result < height
#Percentage of data within 1, 2 and 3 Standard Deviations for Weight
weight_list_of_data_within_1_std_deviation = [result for result in weight_list if result > weight_first_std_deviation_start and result < weight
weight_list_of_data_within_2_std_deviation = [result for result in weight_list if result > weight_second_std_deviation_start and result < weigh
weight_list_of_data_within_3_std_deviation = [result for result in weight_list if result > weight_third_std_deviation_start and result < weight
#Printing data for height and weight (Standard Deviation)
print("{}% of data for height lies within 1 standard deviation".format(len(height_list_of_data_within_1_std_deviation)*100.0/len(height_list)))
print("{}% of data for height lies within 2 standard deviations".format(len(height_list_of_data_within_2_std_deviation)*100.0/len(height_list)))
print("{}% of data for height lies within 3 standard deviations".format(len(height_list_of_data_within_3_std_deviation)*100.0/len(height_list)))
print("{}% of data for weight lies within 1 standard deviation".format(len(weight_list_of_data_within_1_std_deviation)*100.0/len(weight_list)))
print("{}% of data for weight lies within 2 standard deviations".format(len(weight_list_of_data_within_2_std_deviation)*100.0/len(weight_list)))
print("{}% of data for weight lies within 3 standard deviations".format(len(weight_list_of_data_within_3_std_deviation)*100.0/len(weight_list)))
```

```
95.46% of data for height lies within 2 standard deviations
95.284% of data for weight lies within 2 standard deviations
```

3rd deviation.

```python
#1, 2 and 3 Standard Deviations for height
height_first_std_deviation_start, height_first_std_deviation_end = height_mean-height_std_deviation, height_mean+height_std_deviation
height_second_std_deviation_start, height_second_std_deviation_end = height_mean-(2*height_std_deviation), height_mean+(2*height_std_deviation)
height_third_std_deviation_start, height_third_std_deviation_end = height_mean-(3*height_std_deviation), height_mean+(3*height_std_deviation)
#1, 2 and 3 Standard Deviations for weight
weight_first_std_deviation_start, weight_first_std_deviation_end = weight_mean-weight_std_deviation, weight_mean+weight_std_deviation
weight_second_std_deviation_start, weight_second_std_deviation_end = weight_mean-(2*weight_std_deviation), weight_mean+(2*weight_std_deviation)
weight_third_std_deviation_start, weight_third_std_deviation_end = weight_mean-(3*weight_std_deviation), weight_mean+(3*weight_std_deviation)
#Percentage of data within 1, 2 and 3 Standard Deviations for Height
height_list_of_data_within_1_std_deviation = [result for result in height_list if result > height_first_std_deviation_start and result < height_
height_list_of_data_within_2_std_deviation = [result for result in height_list if result > height_second_std_deviation_start and result < heigh
height_list_of_data_within_3_std_deviation = [result for result in height_list if result > height_third_std_deviation_start and result < height_
#Percentage of data within 1, 2 and 3 Standard Deviations for Weight
weight_list_of_data_within_1_std_deviation = [result for result in weight_list if result > weight_first_std_deviation_start and result < weight_
weight_list_of_data_within_2_std_deviation = [result for result in weight_list if result > weight_second_std_deviation_start and result < weigh
weight_list_of_data_within_3_std_deviation = [result for result in weight_list if result > weight_third_std_deviation_start and result < weight_
#Printing data for height and weight (Standard Deviation)
print("{}% of data for height lies within 1 standard deviation".format(len(height_list_of_data_within_1_std_deviation)*100.0/len(height_list)))
print("{}% of data for height lies within 2 standard deviations".format(len(height_list_of_data_within_2_std_deviation)*100.0/len(height_list))
print("{}% of data for height lies within 3 standard deviations".format(len(height_list_of_data_within_3_std_deviation)*100.0/len(height_list))
print("{}% of data for weight lies within 1 standard deviation".format(len(weight_list_of_data_within_1_std_deviation)*100.0/len(weight_list)))
print("{}% of data for weight lies within 2 standard deviations".format(len(weight_list_of_data_within_2_std_deviation)*100.0/len(weight_list))
print("{}% of data for weight lies within 3 standard deviations".format(len(weight_list_of_data_within_3_std_deviation)*100.0/len(weight_list))
```

```
99.796% of data for height lies within 3 standard deviations
```

```
99.724% of data for weight lies within 3 standard deviations
```

What's next?

In the next class, we will learn more about normal distribution.