Sampling Distribution

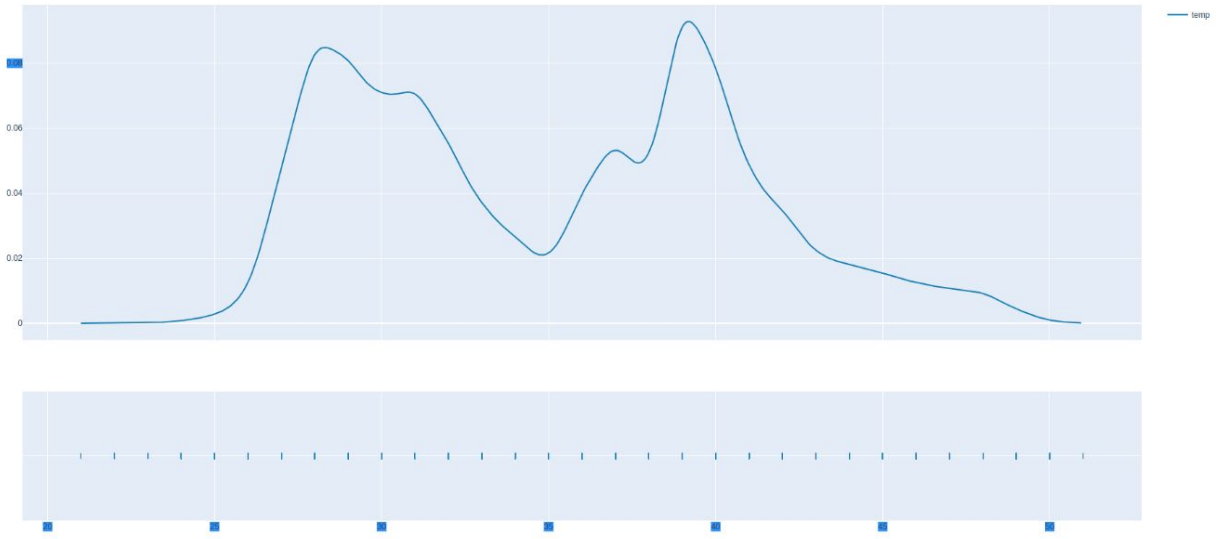
What we did:
In last class we learned about the properties of the normal distribution.

In this class we learned about the sampling distribution.

How we did it:

1. We got a data which didn't follow normal distribution and then we plotted it.

```
main.py                                    height-weight.py
1  import plotly.figure_factory as ff
2  import statistics
3  import random
4  import pandas as pd
5  import csv
6
7  df = pd.read_csv("data.csv")
8  data = df["temp"].tolist()
9  population_mean = statistics.mean(data)
10
11 fig = ff.create_distplot([data], ["temp"], show_hist=False)
12 fig.show()
13
```



2. We saw that the distribution was not normal.
3. We also saw the mean of the distribution and added trace of the mean.

```
df = pd.read_csv("data.csv")
data = df["temp"].tolist()
population_mean = statistics.mean(data)
std_deviation = statistics.stdev(data)
print("population mean:- ", population_mean)
print("std_deviation:- ", std_deviation)
```

```
#function to plot the mean on the graph
def show_fig(mean_list):
    df = mean_list
    fig = ff.create_distplot([df], ["temp"], show_hist=False)
    fig.add_trace(go.Scatter(x=[mean, mean], y=[0, 1], mode="lines", name="MEAN"))
    fig.show()
```

```
population mean:-  35.05393111079237
std_deviation:-  5.699825337585306
```

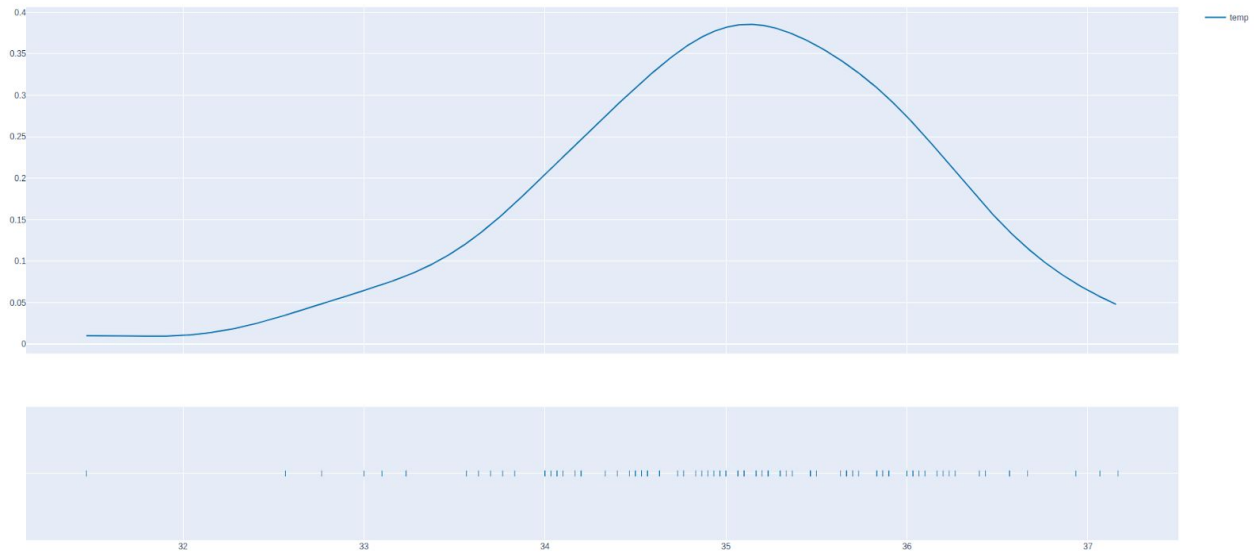
4. We took the sample of 100 data points and got its mean and standard deviation.

```
#code to find mean and std deviation of 100 data points
dataset = []
for i in range(0, 100):
    random_index= random.randint(0,len(data))
    value = data[random_index]
    dataset.append(value)
mean = statistics.mean(dataset)
std_deviation = statistics.stdev(dataset)

print("Mean of sample:- ",mean)
print("std_deviation of sample:- ",std_deviation)
```

```
Mean of sample:-  34.85
std_deviation of sample:-  5.768908304896432
```

5. Then we calculated the mean of the 100 random data points 1000 times and plotted it on the graph.

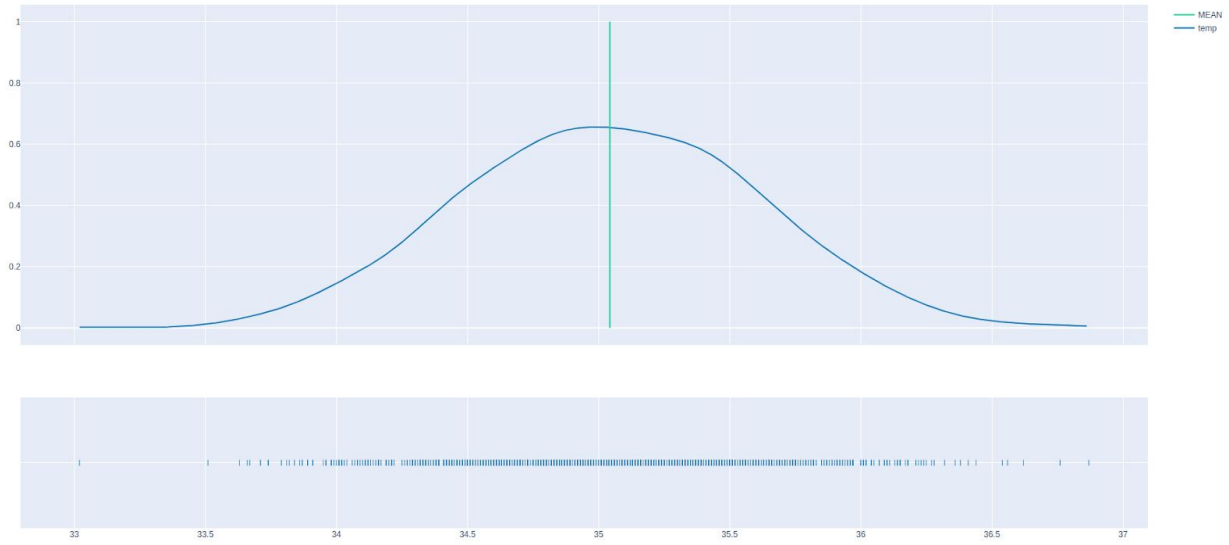
```
#function to get the mean of the given data samples
def random_set_of_mean(counter):
    dataset = []
    for i in range(0, counter):
        random_index= random.randint(0,len(data))
        value = data[random_index]
        dataset.append(value)
    mean = statistics.mean(dataset)
    return mean

#function to plot the mean on the graph
def show_fig(mean_list):
    df = mean_list
    fig = ff.create_distplot([df], ["temp"], show_hist=False)
    fig.show()

#function to get mean of 100 data points 1000 times and plot the graph
def setup():
    mean_list = []
    for i in range(0,1000):
        set_of_means= random_set_of_mean(100)
        mean_list.append(set_of_means)
    show_fig(mean_list)

setup()
```



6. Then we calculated the mean of the sampling data and added a trace for the mean.

```
#function to plot the mean on the graph
def show_fig(mean_list):
    df = mean_list
    mean = statistics.mean(mean_list)
    print("Mean of sampling distribution :-",mean )
    fig = ff.create_distplot([df], ["temp"], show_hist=False)
    fig.add_trace(go.Scatter(x=[mean, mean], y=[0, 1], mode="lines", name="MEAN"))
    fig.show()
```

```
Mean of sampling distribution :- 35.04286
```



7. Then we calculated the standard deviation of the sample mean.

```
def standard_deviation():
    mean_list = []
    for i in range(0,1000):
        set_of_means= random_set_of_mean(100)
        mean_list.append(set_of_means)

    std_deviation = statistics.stdev(mean_list)
    print("Standard deviation of sampling distribution:- ", std_deviation)

standard_deviation()
```

```
Standard deviation of sampling distribution:-  0.56694621050467
```

8. We concluded that the Standard deviation of the sampling mean = 1/10 of Population Standard deviation.

9. We tried the same with the different set of data.

What's next?

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