Topic	Sampling Distribution	
Class Description	**Student learns to draw random samples from a population to create a sampling distribution of a population. Student discovers that sampling distribution for any population is a normal distribution. Student also identifies the relationship between the means and standard deviation of a population and of a sample distribution.**	
Class	**C110**	
Class time	**45 mins**	
Goal	● Draw a distribution for temperature in a room ● Draw random samples from the population to create a sampling (normal distribution) ● Identify the relationship between mean and standard deviation of the population vs mean and standard deviation of the sampling distribution	
Resources Required	● Teacher Resources ○ VCS ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ● Student Resources ○ VCS ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen	
Class structure	**Warm Up Teacher-led Activity Student-led Activity Wrap up**	**5 mins 15 min 15 min 5 min**

CONTEXT

- **How do we analyze data which is not a normal distribution**

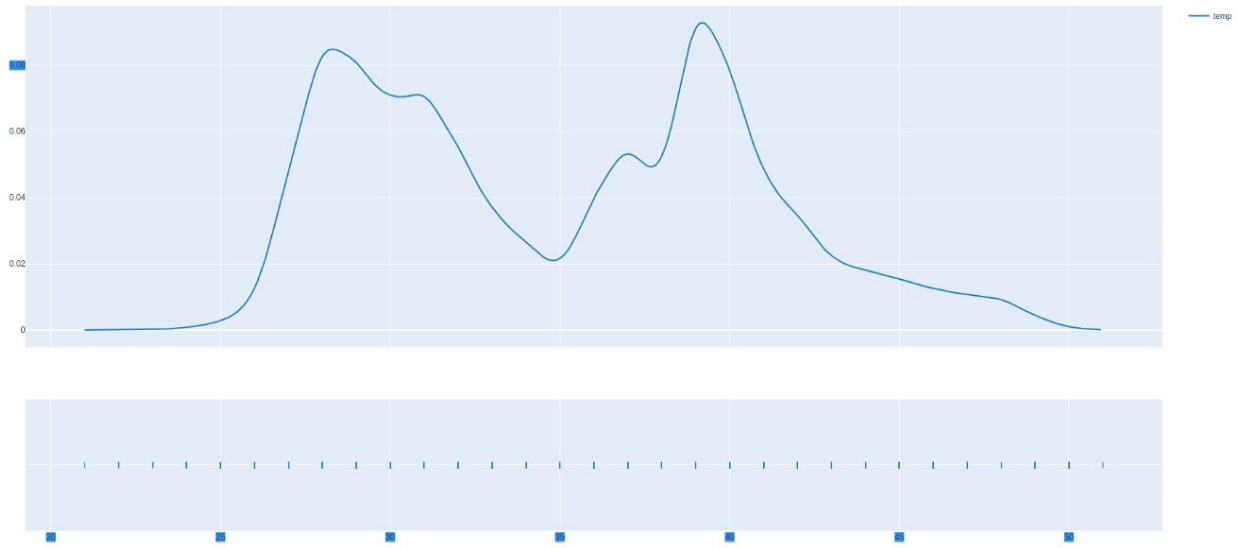
Class Steps	Teacher Action	Student Action
Step 1: Warm Up (5 mins)	Hey <Student name> Remember what we did in the last class?	ESR: We learned about the properties of the normal distribution
	Can you detail the properties of normal distribution we discovered in the last class? All these properties of normal distribution will be super-useful when we explore machine learning algorithms to predict data.	ESR: We have learned that normal distributions can be seen as probability distributions. Mean = Median = Mode in a normal distribution and corresponds to the peak value Normal distribution is symmetric around the peak value. 68% of all data lie within one standard deviation of the mean 95% of all the data lie within two standard deviation of the mean 99% of all the data lie within three standard deviation of the mean
	We have discovered that most of the data in the natural world follows a normal distribution. But there are some data sets which do not follow a normal distribution. We're going to explore and analyze this kind of data in today's class. Let's get started.	-

<table>
<tr><th colspan="3">Teacher Initiates Screen Share</th></tr>
<tr><td colspan="3"><u>CHALLENGE</u>
• Define population and sample population
• Introduce the concept of sampling and creating a sampling distribution of the population
• Identify the relationship between mean and standard deviation of population vs mean and standard deviation of sample population</td></tr>
<tr><td>Step 2:
Teacher-led Activity
(15 min)</td><td>Let's start with the temperature data in the room we had started exploring in the last class.
Can you help me draw a plot for the temperature data?

Teacher writes the code to draw the distribution for temperature data collected by the sensors in a room</td><td>Student helps the teacher in writing the code to draw the distribution for the temperature data</td></tr>
<tr><td colspan="3">main.py • height-weight.py

```
import plotly.figure_factory as ff
import statistics
import random
import pandas as pd
import csv

df = pd.read_csv("data.csv")
data = df["temp"].tolist()
population_mean = statistics.mean(data)

fig = ff.create_distplot([data], ["temp"], show_hist=False)
fig.show()
```

</td></tr>
</table>



	Is the distribution normal? So clearly, there are datasets which do not follow normal distribution.	ESR: No!
	Let us try to find the mean and standard deviation of the data we have. Can you help me find the mean and standard deviation of the data we have? Teacher can also draw a traceline at the mean.	Student helps the teacher import the statistics package and use mean() and stdev() to find the mean and standard deviation of the data set

```
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
```

	We know the mean and standard deviation of the data. We also know that the data does not have a normal distribution. Now, let's do a small experiment. Let's takes out 100 random data points from the raw temperature data. 100 random data points is called "sample data" and raw temperature data is called "population data" We will collect 100 random data points, calculate the mean & standard deviation and print it on the console. Can you help me write the code for this?	Student helps the teacher randomly select data points from the population. ESR: We can import random and use random.randint to generate random indices from which we can collect the data.

	How do we do the random selection from the raw data list? Teacher writes code to collect ONE sample of random 100 data points , find its mean & standard deviation and print it.	

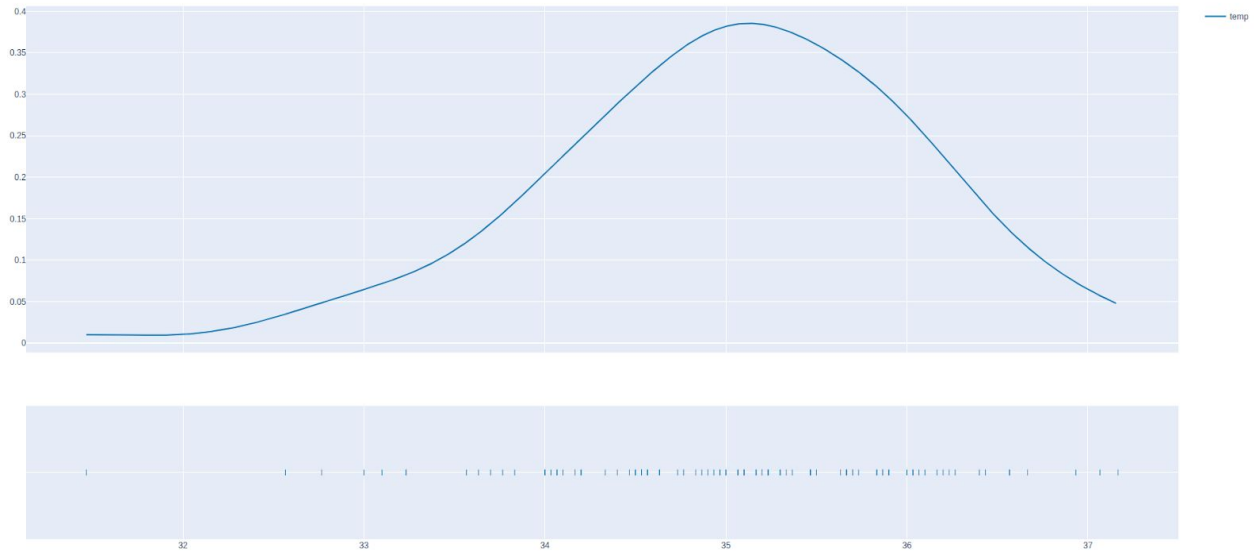
```
#code to find mean and std deviation of 100 data points
dataset = []
for i in range(0, 100):
    random_index= random.randint(0,len(data))
    value = data[random_index]
    dataset.append(value)
mean = statistics.mean(dataset)
std_deviation = statistics.stdev(dataset)

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

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

	What is the mean of the 100 random data points from the population? Is it the same or is it different from the population mean? What is the standard deviation of the 100 random data points from the population? Is it the same or is it different from the population standard deviation? Most likely both the sample mean and	ESR: Mean of sample :- 34.85 Sample Standard Deviation:-5.768 They are different from the population mean and standard deviation

	the standard deviation are different from the population mean and standard deviation	
	Now, let's do this a 1000 times. Let us collect 1000 random samples of 100 data points from the population. Let us find the mean of each sample and store them in a list. Let's then plot the distribution of all the sample means. Can you help me write the code for this? Teacher writes the code and runs it.	Student helps the teacher to write code to collect 1000 random samples of 100 data points from the population and plot the distribution for all the sample means.

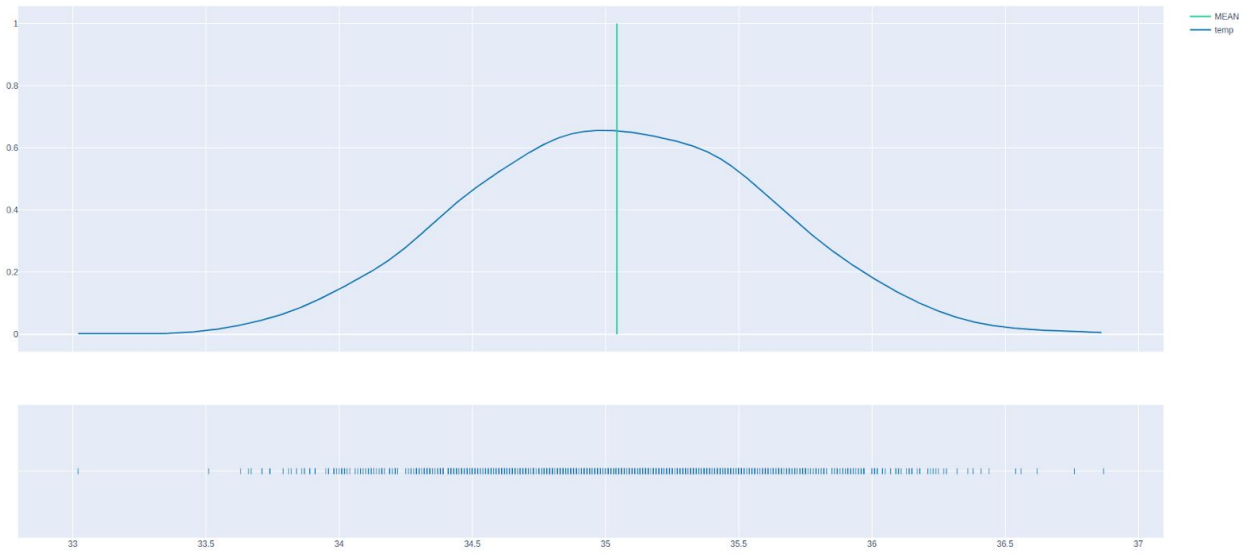
```python
#function to get the mean of the given data samples
def random_set_of_mean(counter):
    dataset = []
    for i in range(0, counter):
        random_index= random.randint(0,len(data))
        value = data[random_index]
        dataset.append(value)
    mean = statistics.mean(dataset)
    return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



	What do you observe about the distribution graph for the sampling means?	ESR: Student observes the sampling mean distribution to be a normal distribution!
	This kind of distribution is called sampling mean distribution. And provided we take sufficient number of samples, the sampling mean distribution for ANY kind of data will always be a normal distribution! Isn't that amazing? Let's find the mean of the sampling distribution. Teacher with the help of student writes the code for finding the mean of the sampling distribution Teacher can print the mean of the sample distribution and also draw a traceline at the mean	ESR: Yes! Student helps the teacher write code for finding the mean of the sample distributions

```
#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
```



	What do you observe about the mean ?	ESR: Mean for the sampling distribution is the same as the mean for the population.
	Yes! This is true for all sampling distributions for all kinds of data. Mean for sampling distribution = Population Mean	-
	Let's try to find the standard deviation for the sampling mean distribution. Can you help me write the code for it. Teacher writes the code to calculate the standard deviation for the sampling mean distribution.	Student helps the teacher write the code for calculating the standard deviation for the sampling mean distribution

```
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
```

	Teacher runs the code to show standard deviation of the sampling mean distribution. What is the standard deviation of the sampling mean distribution? Do you observe any relation with the population standard deviation?	ESR: standard deviation of sampling mean = 0.5669 Standard deviation of the sampling mean = 1/10 of Population Standard deviation
	Great observation! In fact the relationship between standard deviation of population and standard deviation of sampling mean distribution is given by: Standard deviation of sampling mean distribution = Standard Deviation of Population / sqrt (number of data in each sample) Here, the number of data in each sample was 100. So, we found the standard deviation of sampling mean	Student asks questions to clarify the relationship.

	distribution = 1/10 * standard deviation of the population	
	Standard deviation of the sampling mean distribution is also called standard error of the mean (SE) The relationship is important as it allows us to get information about the population data from the sampling mean data. We'll learn how to use this in our next class where we will try to understand if a given random sample belongs to a particular population!! Interesting?	ESR: Yes!
	Meanwhile, do you want to try different size of samples and check if the relationship still holds true.	-
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	

- **Ask Student to press ESC key to come back to panel**
- **Guide Student to start Screen Share**
- **Teacher gets into Fullscreen**

ACTIVITY

- **Experiment with different numbers of samples to create different sampling distributions.**
- **Validate the relationship between mean and standard deviation of population and sampling distribution.**

Step 3: Student-Led Activity (15 min)	Guide the student to download the sample and write code to create a sample of 100 data points	Student writes code to create sample of 100 data points

```
#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)
```

	Guide the student to create 1000 such samples and plot their means (sampling mean distribution).	Student creates 1000 samples - each having 100 data points. Student finds the means of each sample, stores in a list and plots their distribution

```
#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()
```

	Guide the student to find the mean of the sampling mean distribution and observe the relationship between sampling mean and population mean	Student finds the mean of the samples. Student observes the relation ship Sampling Mean = Population Mean

```
population_mean = statistics.mean(data)
print("population mean:- ", population_mean)
mean = statistics.mean(mean_list)
print("Mean of sampling distribution :-",mean )
```

	Guide the student to find the standard deviation of the sampling mean (sampling error of the mean) and identify the relationship between population standard deviation and sampling error of the mean.	Student finds the sampling error of the mean and verifies the relationship. Sampling error of the mean = Standard deviation of population / sqrt (30)

```
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()
```

	Guide the student to use different sampling sizes to verify the relationship.	Student uses the different sampling sizes to verify the relationship between the standard deviation of the population and standard error of the mean

```
#function to get the mean of the given data samples
# pass the number of data points you want  as counter
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.add_trace(go.Scatter(x=[mean, mean], y=[0, 1], mode="lines", name="MEAN"))
    fig.show()


# Pass the number of time you want the mean of the data points as a parameter in range function in for loop
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()
```

	Guide the student to use different number of samples while plotting the sampling mean distribution. Get them to observe how the distribution shape is not close to normal when the sample sizes are small.	Student experiments plotting the sampling mean distribution with different sample sizes.

	We performed the different analysis with the temperature data collected by sensors in a room. But you can try this with any other data which follows or does not follow a normal distribution!	Student uses the alternative dataset to test out the statistical insights collected in the current lesson and verifies it.
Teacher Guides Student to Stop Screen Share		
FEEDBACK ● **Appreciate the student for their efforts** ● **Identify 2 strengths and 1 area of progress for the student**		
Step 4: Wrap-Up (5 min)	Let's quickly summarize what we learned in today's class.	ESR: - We learned how to create a sampling mean distribution of a population for different sample sizes and different sampling sizes - We observed how sampling mean distribution for large sample sizes are always normal distribution - Sampling Mean = Population Mean - Sampling Error = Population Standard Deviation / sqrt (sampling size)

	We will be using these statistical insights and what we learned in the last class to make predictions and conclusions about data sets in the next class!	-
Teacher Clicks 		
Additional Activities	Encourage the student to write reflection notes in their reflection journal using markdown. Use these as guiding questions: ● What happened today? - Describe what happened - Code I wrote ● How did I feel after the class? ● What have I learned about programming and developing games? ● What aspects of the class helped me? What did I find difficult?	The student uses the markdown editor to write her/his reflection in a reflection journal.

Activity	Activity Name	Links
Teacher Activity 1	Solution link	https://github.com/whitehatjr/Sampling-distribution
Student Activity 1	Data link	https://raw.githubusercontent.com/whitehatjr/datasets/master/newdata.csv