Topic	Single Sample z-tests	
Class Description	**Students apply the properties of normal distribution to formulate a single sample z-test. Students study the math scores for students in a school before and after an intervention to understand if the intervention had a significant impact.**	
Class	**C111**	
Class time	**45 mins**	
Goal	• Study the math score for the population • Create a sampling distribution for the population with sampling size = 100 • Use z-test to analyze if the intervention had a significant impact on the population	

Resources Required	• Teacher Resources
	  ○ Visual Studio Code with Live Share Plugin Installed
	  ○ Laptop with internet connectivity
	  ○ Earphones with mic
	  ○ Notebook and pen
	• Student Resources
	  ○ Visual Studio Code with Live Share Plugin Installed
	  ○ 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

- **Review the concepts learned in the earlier classes**

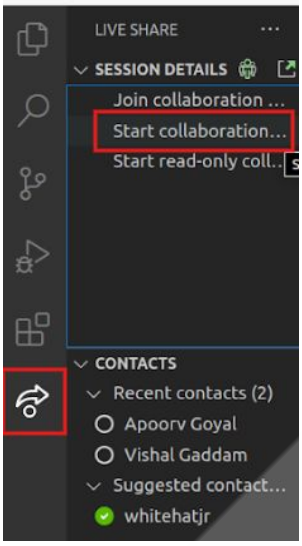
Class Steps	Teacher Action	Student Action

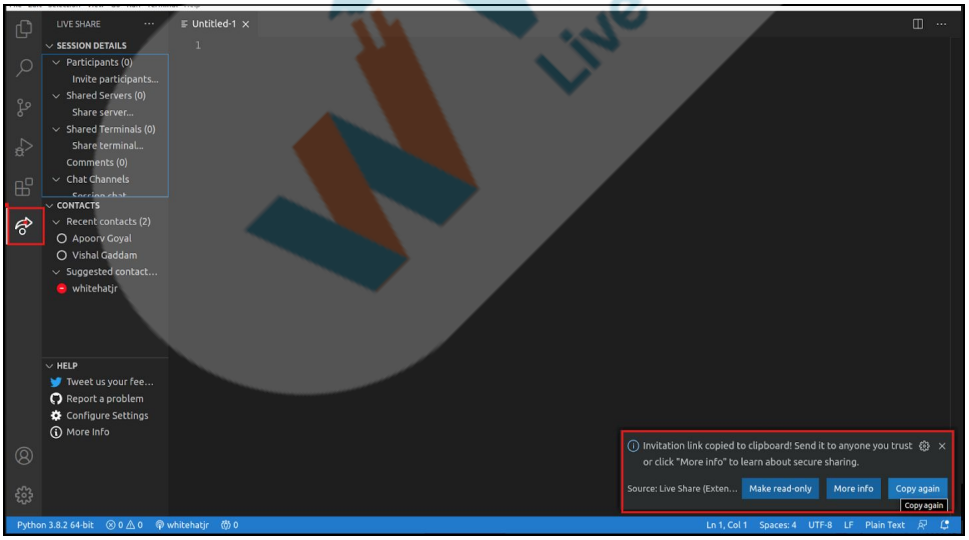
Step 1: Warm Up (5 mins)	Hi <Student Name>! Today, we will learn to apply all the fascinating things we learned about normal distributions in the last few classes to a practical real-life problem. But before we do that, can we re-capture what we learned in the last couple of classes about normal and sampling distributions.	**ESR:** - Most data in the world can be plotted as a normal distribution which looks like a bell-curve. - Mean of a population in a normal distribution is at its peak. - Most of the data (68%) in the population lie within one standard deviation from the mean. - 95 % of data lie within two standard deviations from the mean and 99% of data lie within three standard deviation from the mean. - Sampling distribution is the distribution created when we plot means of 1000s of samples with a fixed sampling size. - Mean of a sampling distribution is the same as the mean of the population. - Standard deviation of the sampling distribution (also called standard error of the mean) = standard deviation of the sampling distribution / sqrt (sampling size).
	Awesome. Let's learn how we can use these in a real-life scenario.	-
Teacher Initiates Screen Share		

CHALLENGE

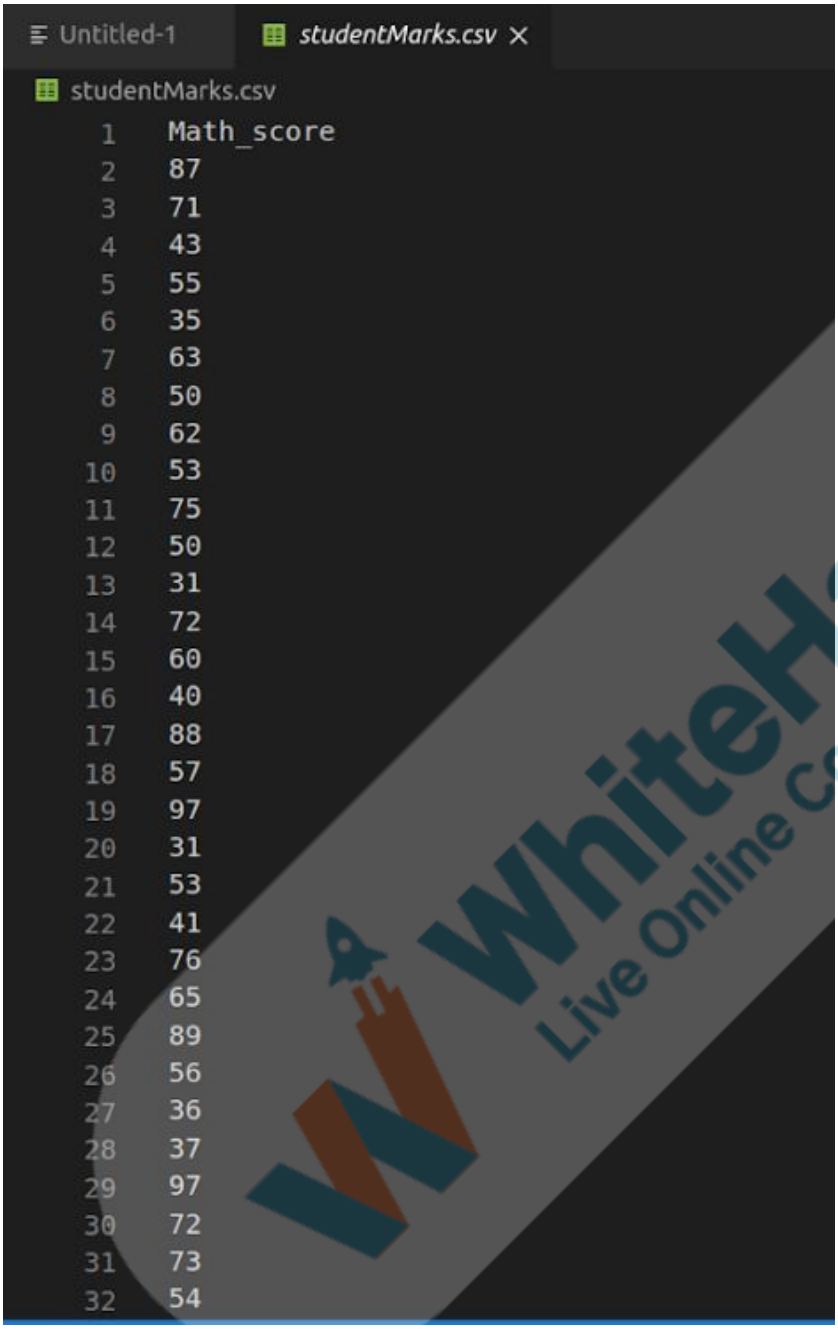
- **Study the math scores of students in a population**
- **Create a sampling distribution for the population with 100 sampling size**

Step 2: Teacher-led Activity (15 min)	*Ask the student to create a file in visual studio code and use a live sharing plugin to share the file link.*	*The student creates a python file and shares with the teacher using a live sharing plugin.*





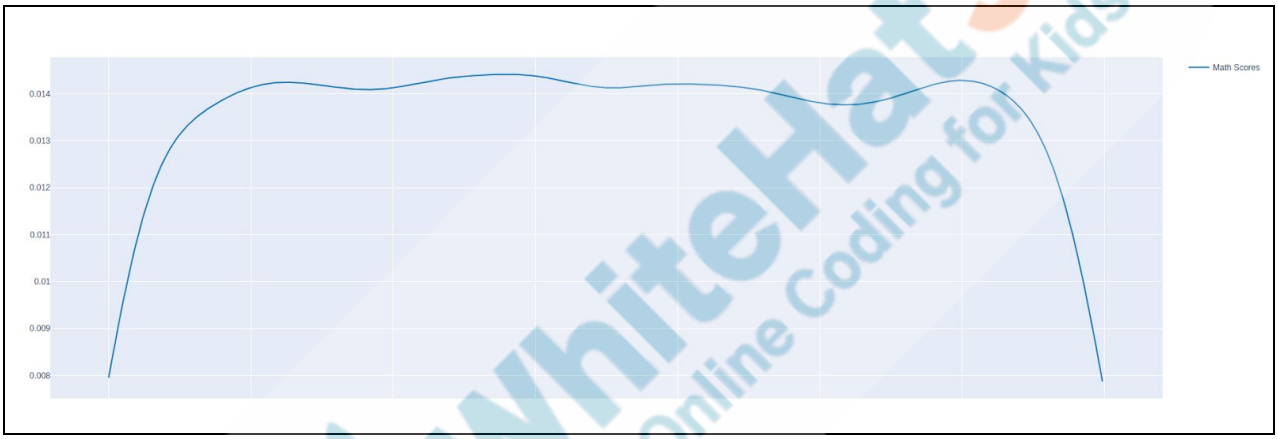
	Rita Prasad is a principal of a school. The school has 10000 students across Grades 1 to 12. The school students performed not so well in the recent math test in their half-yearly exams. Here is the Math score for all the students in the exam. **<Student Activity 1>** Do you want to open the link and look through the data?	*The student downloads the data and looks through the math score data for 10000 students.*



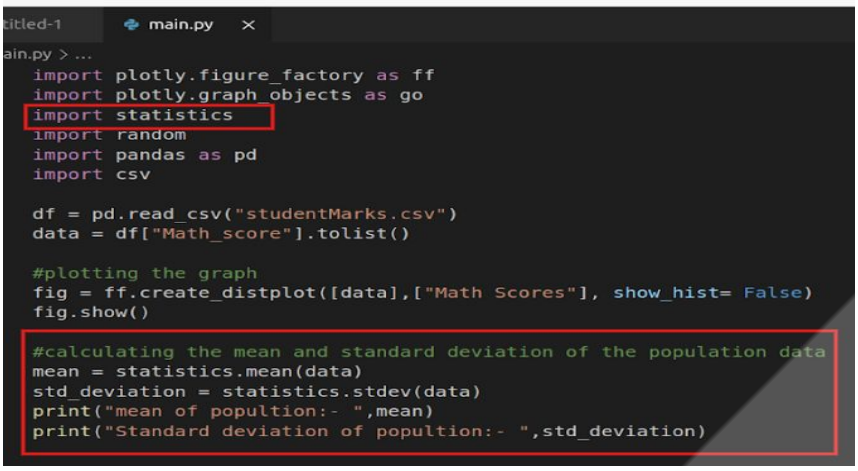
	What can we do to better understand the data and how the students have done?	**ESR:** - We can plot the distribution data. - We can calculate the mean and standard deviation for the data.

	Let us do it. Let us plot the distribution for the marks and see how the marks are distributed over the 10000 students. What is the shape of the distribution you expect?	**ESR:** Bell curve or normal distribution.
	Guide the student to write the code to plot the distribution for the math scores for all 10000 students. *<Help the student whenever they are stuck>* *Verify that it is a normal distribution.*	*The student writes code to plot the distribution of math scores for all the 10000 students.* *- Student imports plotly.figure_factory as ff.* *- The student imports pandas as pd.* *- Student imports csv.* *- The student uses pandas to read the csv file.* *- Converts the data into list and saves in a **data** variable.* *- Using plotly.figure_factory plots the data on the distplot.* *The student runs the code to see the normal distribution plot.*

```
main.py > mean
1  import plotly.figure_factory as ff
2  import plotly.graph_objects as go
3  import statistics
4  import random
5  import pandas as pd
6  import csv
7
8  df = pd.read_csv("studentMarks.csv")
9  data = df["Math_score"].tolist()
10
11 #plotting the graph
12 fig = ff.create_distplot([data],["Math Scores"], show_hist= False)
13 fig.show()
14
```



	Let us also find the mean and standard deviation for this data.	The student uses the statistics package to calculate the mean and standard deviation of the data.



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

df = pd.read_csv("studentMarks.csv")
data = df["Math_score"].tolist()

#plotting the graph
fig = ff.create_distplot([data],["Math Scores"], show_hist= False)
fig.show()

#calculating the mean and standard deviation of the population data
mean = statistics.mean(data)
std_deviation = statistics.stdev(data)
print("mean of popultion:- ",mean)
print("Standard deviation of popultion:- ",std_deviation)
```

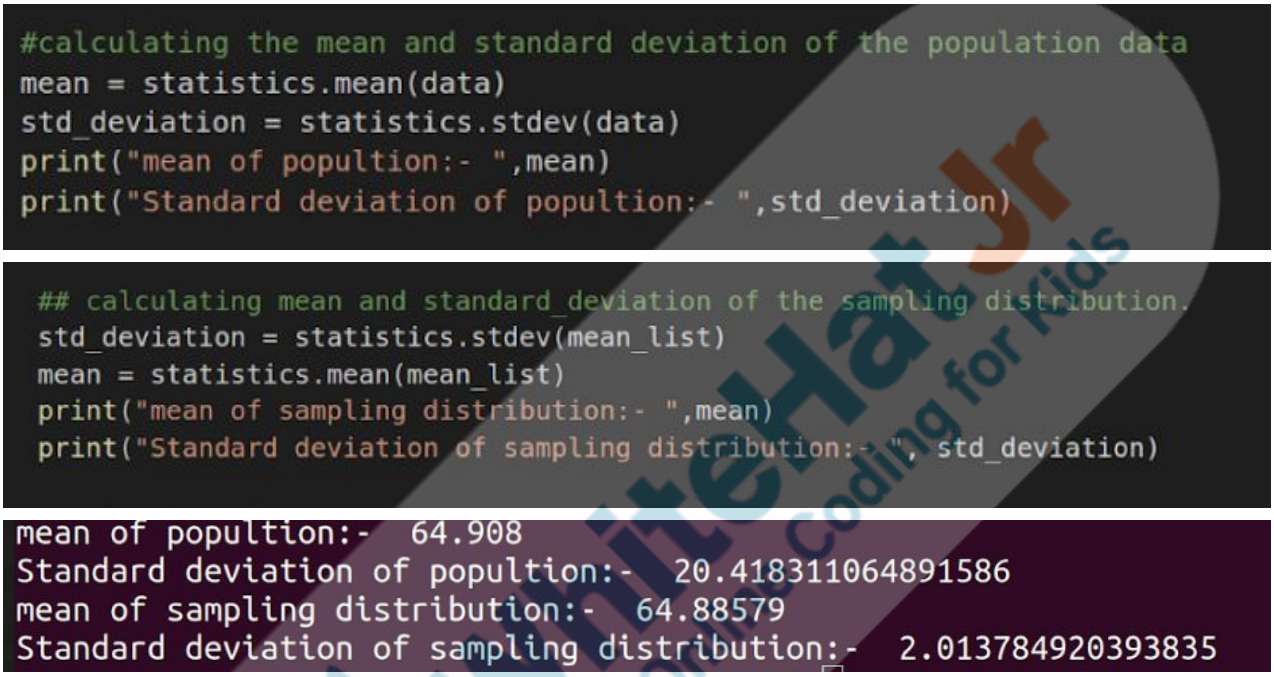
```
mean of popultion:-  64.908
Standard deviation of popultion:-  20.418311064891586
```

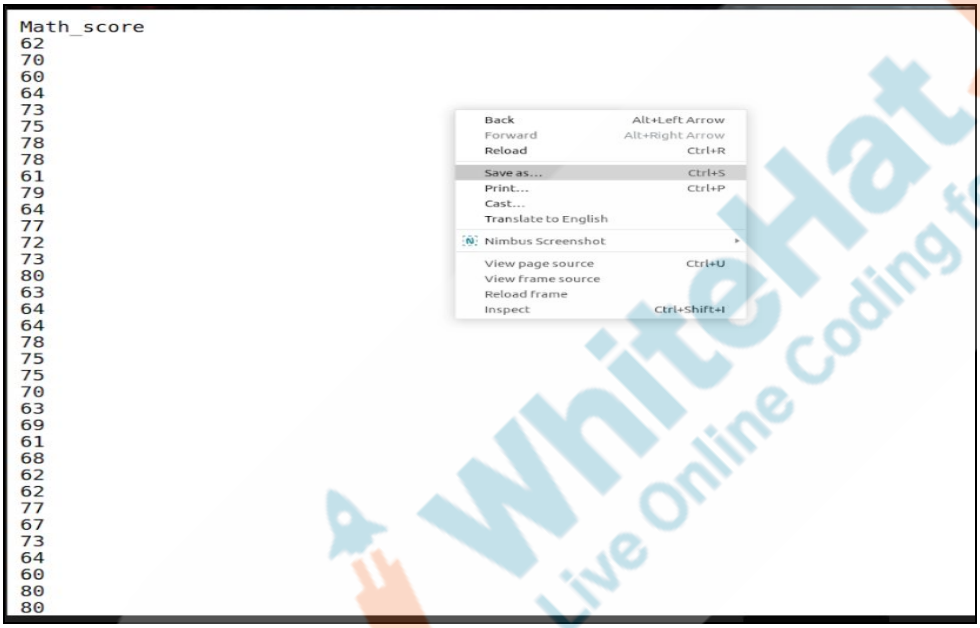
	What does this data tell you?	**ESR:** varied
	Alright. Rita Prasad, as the school principal decided to do something to improve the math scores of the students in her school. If you would be the principal, what are the things you would do?	**ESR:** - Extra classes - More practice problems - Workshops - Use of tech in classrooms
	Now, some of these things might be effective and some of these might not be effective in improving the Math scores. How do you think Rita Prasad should proceed with this?	**ESR:** varied
	Rita Prasad comes from a Data Science background. She decided to randomly select 3 groups of 100 students each and try different interventions on these groups. She then wants to understand	*The student asks questions about the approach Rita Prasad is taking.*

	the impact of each intervention and study if the effect of these interventions was positive enough so that it can be implemented throughout the school.	
	We would like to help Rita Prasad study the impact of different interventions on her random groups and also tell her which one is more effective. We will use the strategies and information which we have learned in the classes so far. Before that, let us draw a sampling distribution for the data with sampling size of 100. We can take around 1000 samples to create the distribution. *Guide the student to create the sampling distribution.* Check if the sample distribution is a bell shaped curve.	*Student codes to randomly generate 1000 samples of 100 sample size, find the mean and plot them on a distribution.*

```python
##  code to find the mean of 100 data points 1000 times
#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)-1)
        value = data[random_index]
        dataset.append(value)
    mean = statistics.mean(dataset)
    return mean


# Pass the number of time you want the mean of the data points as a parameter in range function in for loop
mean_list = []
for i in range(0,1000):
    set_of_means= random_set_of_mean(100)
    mean_list.append(set_of_means)


## calculating mean and standard_deviation of the sampling distribution.
std_deviation = statistics.stdev(mean_list)
mean = statistics.mean(mean_list)
print("mean of sampling distribution:- ",mean)

#plotting the mean of the sampling
fig = ff.create_distplot([mean_list], ["student marks"], show_hist=False)
fig.add_trace(go.Scatter(x=[mean, mean], y=[0, 0.20], mode="lines", name="MEAN"))
fig.show()
```

	What do you think the mean for this sampling distribution would be? What do you think would be the standard deviation of the sampling distribution?	**ESR:** Mean of sampling distribution should be equal to the mean of the population. Standard deviation of the sampling distribution = Standard deviation of the population / sqrt (100).

	Let's find the mean and standard error of the mean programmatically and check if they meet the expected values.	*The student writes code to calculate the standard error of the mean and mean values for sampling distribution.*
		
	Alright, remember what Rita Prasad decided to do with 3 random samples of 100 students in her school. She decided to put the three groups through 3 different interventions: 1. One group of 100 randomly selected students were given iPad tablets with math reading and learning materials in it. 2. Second group of 100 randomly selected students were given 2 hours of extra classes daily. 3. Third group of 100 randomly selected students were given math	*Student downloads and looks at the data set from Student Activity 2.*

	fun sheets to practice at home. After a month of intervention, all the three groups were tested again. The results of the score for all the three groups are in data1.csv, data2.csv and data3. csv <Student Activity 2>.	
		
	Can we use the data to evaluate if any of the three interventions worked, which interventions worked best and which one should be scaled with all the students in the school?	*The student thinks about it.*
	Let's try to see if we can use the score data for these sample students and statistically understand if the interventions have worked.	-
Teacher Stops Screen Share		

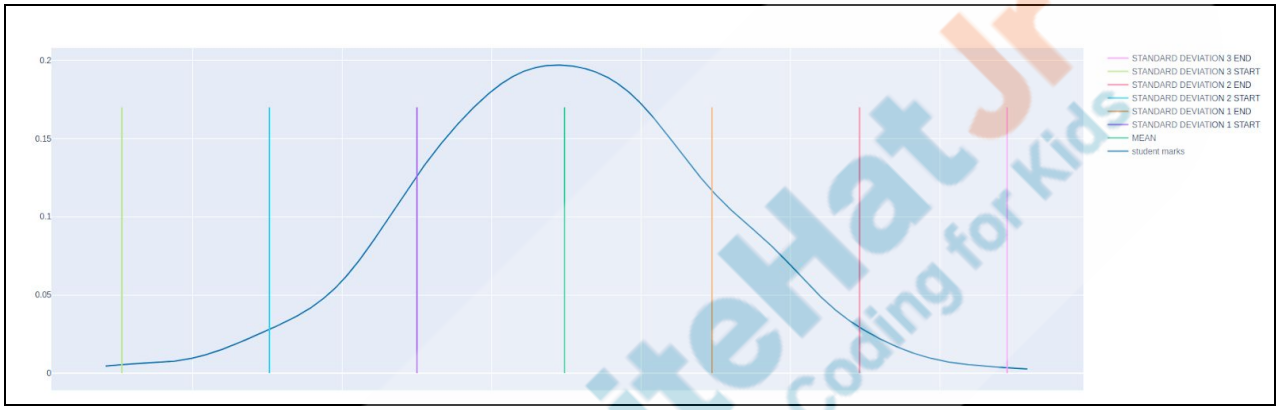
	Now it's your turn. Please share your screen with me.	
• **Ask Student to press ESC key to come back to panel** • **Guide Student to start Screen Share** • **Teacher gets into Fullscreen**		
ACTIVITY • **Calculate z-score of the intervention sample.** • **Estimate if the intervention had a significant impact on math learning and math scores of the students by calculating z score.**		
Step 3: Student-Led Activity (15 min)	Let's look at the sampling distribution again. Our 100 students sample could be anywhere on this distribution. It could be on the right of the mean or it could be on the left of the mean. We know that: 68% of data lie between one standard deviation from the mean. If the new sample mean lies within one standard deviation, we can say that not much might have changed because of the intervention. In statistics, we say that the change has not been significant. 95% of data lie between two standard deviations from the mean. If the sample mean lies beyond two standard deviations from the mean, we can say that there must have been a significant change because of the intervention.	*Student understands, asks questions and repeats his understanding of how to analyze the means of the three interventions.*

	99% of data lie between three standard deviations from the mean. If the sample mean lies even beyond the three standard deviations from the mean, it is highly probable that the intervention has an impact. If the intervention has been positive, the mean would lie to the right. If the intervention has been negative, the mean would lie to the left. *Get the student to understand and repeat these in their own words.*	
	Let us draw a traceline to mark one standard deviation, two standard deviation and three standard deviation in the figure.	*The student writes code to draw tracelines at one standard deviation, two standard deviation and at three standard deviation from the mean.*

```
## findig the standard deviation starting and ending 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)
third_std_deviation_start, third_std_deviation_end = mean-(3*std_deviation), mean+(3*std_deviation)
print("std1",first_std_deviation_start, first_std_deviation_end)
print("std2",second_std_deviation_start, second_std_deviation_end)
print("std3",third_std_deviation_start,third_std_deviation_end)

## plotting the graph with traces
fig = ff.create_distplot([mean_list], ["student marks"], show_hist=False)
fig.add_trace(go.Scatter(x=[mean, mean], y=[0, 0.17], mode="lines", name="MEAN"))
fig.add_trace(go.Scatter(x=[first_std_deviation_start, first_std_deviation_start], y=[0, 0.17], mode="lines
fig.add_trace(go.Scatter(x=[first_std_deviation_end, first_std_deviation_end], y=[0, 0.17], mode="lines", r
fig.add_trace(go.Scatter(x=[second_std_deviation_start, second_std_deviation_start], y=[0, 0.17], mode="lir
fig.add_trace(go.Scatter(x=[second_std_deviation_end, second_std_deviation_end], y=[0, 0.17], mode="lines",
fig.add_trace(go.Scatter(x=[third_std_deviation_start,third_std_deviation_start], y=[0,0.17], mode="lines",
fig.add_trace(go.Scatter(x=[third_std_deviation_end,third_std_deviation_end], y=[0,0.17], mode="lines", nam
fig.show()
```

Continued part of the code:

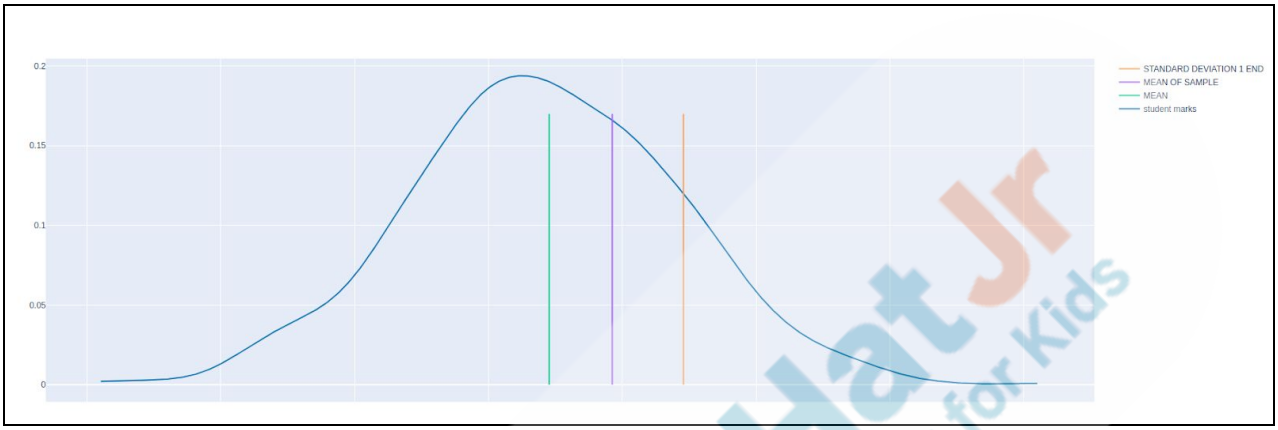
```
t marks"], show_hist=False)
  0.17], mode="lines", name="MEAN"))
n_start, first_std_deviation_start], y=[0, 0.17], mode="lines", name="STANDARD DEVIATION 1 START"))
n_end, first_std_deviation_end], y=[0, 0.17], mode="lines", name="STANDARD DEVIATION 1 END"))
on_start, second_std_deviation_start], y=[0, 0.17], mode="lines", name="STANDARD DEVIATION 2 START"))
on_end, second_std_deviation_end], y=[0, 0.17], mode="lines", name="STANDARD DEVIATION 2 END"))
n_start,third_std_deviation_start], y=[0,0.17], mode="lines", name="STANDARD DEVIATION 3 START"))
n_end,third_std_deviation_end], y=[0,0.17], mode="lines", name="STANDARD DEVIATION 3 END"))
```



	Let's find the means of scores for the three different interventions and plot them on the distribution to identify where they lie.	*Student calculates the means for the three samples representing three different interventions. The student draws tracelines to represent the means on the distribution graph.*

```
# finding the mean of the first data and plotting it on the plot.
df = pd.read_csv("data1.csv")
data = df["Math_score"].tolist()
mean_of_sample1 = statistics.mean(data)
print("Mean of sample1:- ",mean_of_sample1)
fig = ff.create_distplot([mean_list], ["student marks"], show_hist=False)
fig.add_trace(go.Scatter(x=[mean, mean], y=[0, 0.17], mode="lines", name="MEAN"))
fig.add_trace(go.Scatter(x=[mean_of_sample1, mean_of_sample1], y=[0, 0.17], mode="lines", name="MEAN OF SAM
fig.add_trace(go.Scatter(x=[first_std_deviation_end, first_std_deviation_end], y=[0, 0.17], mode="lines", r
fig.show()
```

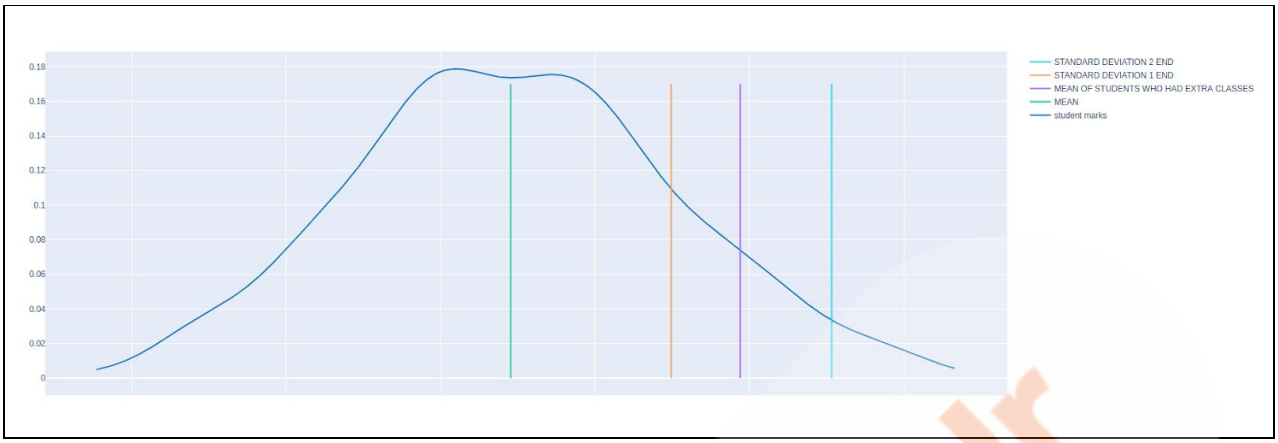
```
it marks"], show_hist=False)
 0.17], mode="lines", name="MEAN"))
ean_of_sample1], y=[0, 0.17], mode="lines", name="MEAN OF SAMPLE"))
n_end, first_std_deviation_end], y=[0, 0.17], mode="lines", name="STANDARD DEVIATION 1 END"))
```



	Where does the group which were given tablets with learning materials lie? Do you think this intervention was impactful?	**ESR:** Within one standard deviation from the mean. **ESR:** No

```
# finding the mean of the SECOND data (STUDENTS WHO HAD EXTRA CLASSES ) and plotting it on the plot.
df = pd.read_csv("data2.csv")
data = df["Math_score"].tolist()
mean_of_sample2 = statistics.mean(data)
print("mean of sample 2:- ",mean_of_sample2)
fig = ff.create_distplot([mean_list], ["student marks"], show_hist=False)
fig.add_trace(go.Scatter(x=[mean, mean], y=[0, 0.17], mode="lines", name="MEAN"))
fig.add_trace(go.Scatter(x=[mean_of_sample2, mean_of_sample2], y=[0, 0.17], mode="lines", name="MEAN OF SA
fig.add_trace(go.Scatter(x=[first_std_deviation_end, first_std_deviation_end], y=[0, 0.17], mode="lines", r
fig.add_trace(go.Scatter(x=[second_std_deviation_end, second_std_deviation_end], y=[0, 0.17], mode="lines",
fig.show()
```

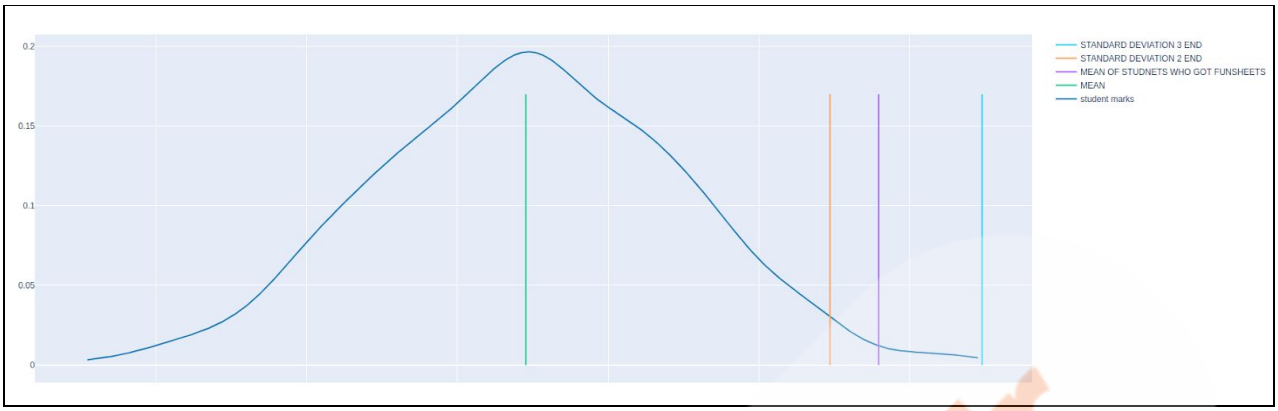
```
it marks"], show_hist=False)
 0.17], mode="lines", name="MEAN"))
ean_of_sample2], y=[0, 0.17], mode="lines", name="MEAN OF SAMPLE WHO GOT "))
n_end, first_std_deviation_end], y=[0, 0.17], mode="lines", name="STANDARD DEVIATION 1 END"))
on_end, second_std_deviation_end], y=[0, 0.17], mode="lines", name="STANDARD DEVIATION 2 END"))
```



	Where does the group which was given daily extra classes lie. Do you think this intervention was impactful?	**ESR:** Beyond two standard deviation from the mean. **ESR:** Yes, it is likely that the intervention was impactful.

```
# finding the mean of the THIRD data (STUDENTS WHO GOT FUNSHEET) and plotting it on the plot.
df = pd.read_csv("data3.csv")
data = df["Math_score"].tolist()
mean_of_sample3 = statistics.mean(data)
print("mean of sample3:- ",mean_of_sample3)
fig = ff.create_distplot([mean_list], ["student marks"], show_hist=False)
fig.add_trace(go.Scatter(x=[mean, mean], y=[0, 0.17], mode="lines", name="MEAN"))
fig.add_trace(go.Scatter(x=[mean_of_sample3, mean_of_sample3], y=[0, 0.17], mode="lines", name="MEAN OF STU
fig.add_trace(go.Scatter(x=[second_std_deviation_end, second_std_deviation_end], y=[0, 0.17], mode="lines",
fig.add_trace(go.Scatter(x=[third_std_deviation_end, third_std_deviation_end], y=[0, 0.17], mode="lines", r
fig.show()
```

```
t marks"], show_hist=False)
 0.17], mode="lines", name="MEAN"))
ean_of_sample3], y=[0, 0.17], mode="lines", name="MEAN OF STUDNETS WHO GOT FUNSHEETS"))
on_end, second_std_deviation_end], y=[0, 0.17], mode="lines", name="STANDARD DEVIATION 2 END"))
n_end, third_std_deviation_end], y=[0, 0.17], mode="lines", name="STANDARD DEVIATION 3 END"))
```



	Where does the group which was given fun sheets / worksheets to solve everyday lie. How impactful was this intervention?	**ESR:** Beyond three standard deviation from the mean. **ESR:** This intervention was the most impactful among all.
	Awesome. This analysis will only hold true if the 100 students selected in the groups are truly random and not biased in any way.	-
	Inspired by Rita Prasad's experiment, three other school principals collaborating in the same area tried to do an intervention in their school and study it. Each school's original math scores are listed here: School1.csv School2.csv School3.csv	*The student draws 3 different normal distributions (sampling size: 100) for each of the three school's math scores.* *Student calculates the mean and standard deviation of the data and draws tracelines at the mean, one sd, two sd and three sd on both the sides of the mean.*

	Each school randomly selected 100 students and ran an intervention with them. School 1 decided to give extra time to these students in the Maths Lab in their school. School 2 decided to use a certain popular Math Practice App for these students. School 3 decided to enforce use of Math Registers for daily problem solving for these students. After a month of intervention, each student in the groups were assessed. These are the scores of the students in the three groups: School_1_sample.csv School_2_sample.csv School_3_sample.csv Could you identify which of the three interventions has worked the best?	*The student calculates the score mean from each sample and plots them on the distribution graph.* *The student can visually estimate which intervention is most effective by calculating how many standard deviations away the new mean is from the sampling mean.*

```
#finding the mean of the STUDENTS WHO USED MATH PRACTISE APP and plotting it on the plot.
df = pd.read_csv("School_2_Sample.csv")
data = df["Math_score"].tolist()
mean_of_sample2 = statistics.mean(data)
print("mean of sample 2:- ",mean_of_sample2)
fig = ff.create_distplot([mean_list], ["student marks"], show_hist=False)
fig.add_trace(go.Scatter(x=[mean, mean], y=[0, 0.17], mode="lines", name="MEAN"))
fig.add_trace(go.Scatter(x=[mean_of_sample2, mean_of_sample2], y=[0, 0.17], mode="lines", name="MEAN OF STU
fig.add_trace(go.Scatter(x=[first_std_deviation_end, first_std_deviation_end], y=[0, 0.17], mode="lines", r
fig.add_trace(go.Scatter(x=[second_std_deviation_end, second_std_deviation_end], y=[0, 0.17], mode="lines",
fig.add_trace(go.Scatter(x=[third_std_deviation_end, third_std_deviation_end], y=[0, 0.17], mode="lines", r
fig.show()

# finding the mean of the STUDENTS WHO WERE ENFORCED WITH REGISTERS and plotting it on the plot.
df = pd.read_csv("School_3_Sample.csv")
data = df["Math_score"].tolist()
mean_of_sample3 = statistics.mean(data)
print("mean of sample3:- ",mean_of_sample3)
fig = ff.create_distplot([mean_list], ["student marks"], show_hist=False)
fig.add_trace(go.Scatter(x=[mean, mean], y=[0, 0.17], mode="lines", name="MEAN"))
fig.add_trace(go.Scatter(x=[mean_of_sample3, mean_of_sample3], y=[0, 0.17], mode="lines", name="MEAN OF STU
fig.add_trace(go.Scatter(x=[second_std_deviation_end, second_std_deviation_end], y=[0, 0.17], mode="lines",
fig.add_trace(go.Scatter(x=[third_std_deviation_end, third_std_deviation_end], y=[0, 0.17], mode="lines", r
fig.show()
```

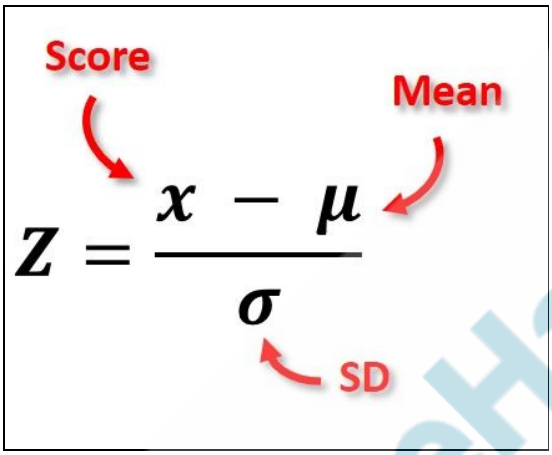
z-score.py > ...

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

#Change the School data here
df = pd.read_csv("School3.csv")
data = df["Math_score"].tolist()


##  code to find the mean of 100 data points 1000 times
#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)-1)
        value = data[random_index]
        dataset.append(value)
    mean = statistics.mean(dataset)
    return mean


# Function to get the mean of 100 data sets
mean_list = []
for i in range(0,1000):
    set_of_means= random_set_of_mean(100)
    mean_list.append(set_of_means)

```

```
## findig the standard deviation starting and ending 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)
third_std_deviation_start, third_std_deviation_end = mean-(3*std_deviation), mean+(3*std_deviation)
# print("std1",first_std_deviation_start, first_std_deviation_end)
# print("std2",second_std_deviation_start, second_std_deviation_end)
# print("std3",third_std_deviation_start,third_std_deviation_end)




# finding the mean of THE STUDENTS WHO GAVE EXTRA TIME TO MATH LAB and plotting on graph
df = pd.read_csv("School_1_Sample.csv")
data = df["Math_score"].tolist()
mean_of_sample1 = statistics.mean(data)
print("Mean of sample1:- ",mean_of_sample1)
fig = ff.create_distplot([mean_list], ["student marks"], show_hist=False)
fig.add_trace(go.Scatter(x=[mean, mean], y=[0, 0.17], mode="lines", name="MEAN"))
fig.add_trace(go.Scatter(x=[mean_of_sample1, mean_of_sample1], y=[0, 0.17], mode="lines", name="MEAN OF STU
fig.add_trace(go.Scatter(x=[first_std_deviation_end, first_std_deviation_end], y=[0, 0.17], mode="lines", r
fig.add_trace(go.Scatter(x=[second_std_deviation_end, second_std_deviation_end], y=[0, 0.17], mode="lines",
fig.add_trace(go.Scatter(x=[third_std_deviation_end, third_std_deviation_end], y=[0, 0.17], mode="lines", r
fig.show()



#finding the mean of the STUDENTS WHO USED MATH PRACTISE APP and plotting it on the plot.
# df = pd.read_csv("School_2_Sample.csv")
# data = df["Math_score"].tolist()
# mean_of_sample2 = statistics.mean(data)
# print("mean of sample 2:- ",mean_of_sample2)
```

Note:-The code will be the same to plot the school samples, you just have to change the data files.

	One way to find the impact of an intervention is by plotting the mean and comparing how many standard deviation away from the mean is the new sample mean by guesstimating. Another way to do is by mathematically calculating it: - Find the difference between the mean and the new sample mean: Sampling Distribution Mean - New sample mean - Calculate how many standard deviations away is the new sample mean from the sampling distribution mean:	*The student calculates the z score for each sample group by writing a program to calculate z score.* *The student compares z scores to arrive at which intervention worked best!* If z > 3 or z > 2 ; the new sample mean lies more than 3 standard deviations away from the sampling distribution mean. It also means that there has been a large impact of the

	(New Sample Mean - Sampling Distribution Mean) / standard deviation This is also called z-score.  z-score tells how many standard deviations away is the new sample mean from the sampling distribution mean. Can you calculate the z-scores of the samples?	intervention. We say that the change in math score is statistically significant. If z < 1 or z < 2; the impact of the intervention might not be statistically significant.

```
#finding the z score using the formula
z_score = (mean_of_sample1 - mean)/std_deviation
print("The z score is = ",z_score)
```

```
mean of sampling distribution:-  50.69924
Standard deviation of sampling distribution:-  2.879529182125215
Mean of sample1:-  50.41
The z score is =  -0.10044697646944323
```

```
#finding the z score using the formula
z_score = (mean_of_sample2 - mean)/std_deviation
print("The z score is = ",z_score)
```

```
mean of sampling distribution:-  49.75977
Standard deviation of sampling distribution:-  2.8632207529598865
mean of sample 2:-  55.33
The z score is =  1.9454420321037795
```

```
#finding the z score using the formula
z_score = (mean_of_sample3 - mean)/std_deviation
print("The z score is = ",z_score)
```

```
mean of sampling distribution:-  50.02569
Standard deviation of sampling distribution:-  2.9773998780704503
mean of sample3:-  57.29
The z score is =  2.4398167184408397
```

	This example was really important because it helped us identify which intervention worked best on different population sets. This type of testing to identify the impact of interventions is used by governments, financial institutions and non-profits to understand what works best with a population.	-

FEEDBACK

- **Appreciate the student for their efforts**
- **Identify 2 strengths and 1 area of progress for the student**

Step 4: Wrap-Up (5 min)	Can you summarize what we learned from today's class?	**ESR:** - We learned how to evaluate the impact of an intervention on a population by calculating z-score. - Z-score is a mathematical score which tells us how

		many standard deviation away is the new mean from the sampling population distribution.
	Perfect. You are fit to become a data analyst now - a person who analyzes data to give us more information. Next class, we will be creating a full-fledged data report.	-
	Up next, we have to dive into the capstone class. It's time to put your creative skills to test. In the next class, we will apply the concepts of mode, median and standard deviation to create a data story. Curious to know about the data story? Stay tuned! Please request your parents to join the 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?	*The student uses the markdown editor to write her/his reflection in a reflection journal.*

	• What have I learned about programming and developing games? • What aspects of the class helped me? What did I find difficult?	

Activity	Activity Name	Links
Student Activity 1	data of math scores of 1000 students.	https://raw.githubusercontent.com/whitehatjr/datasets/master/studentMarks.csv
Student Activity 2	data of interventions	1.https://raw.githubusercontent.com/whitehatjr/datasets/master/data1.csv 2.https://raw.githubusercontent.com/whitehatjr/datasets/master/data2.csv 3.https://raw.githubusercontent.com/whitehatjr/datasets/master/data3.csv
Student Activity 3	school interventions data	https://github.com/whitehatjr/datasets/tree/master/interventions_data
Teacher Activity 1	solution	https://github.com/whitehatjr/z-test