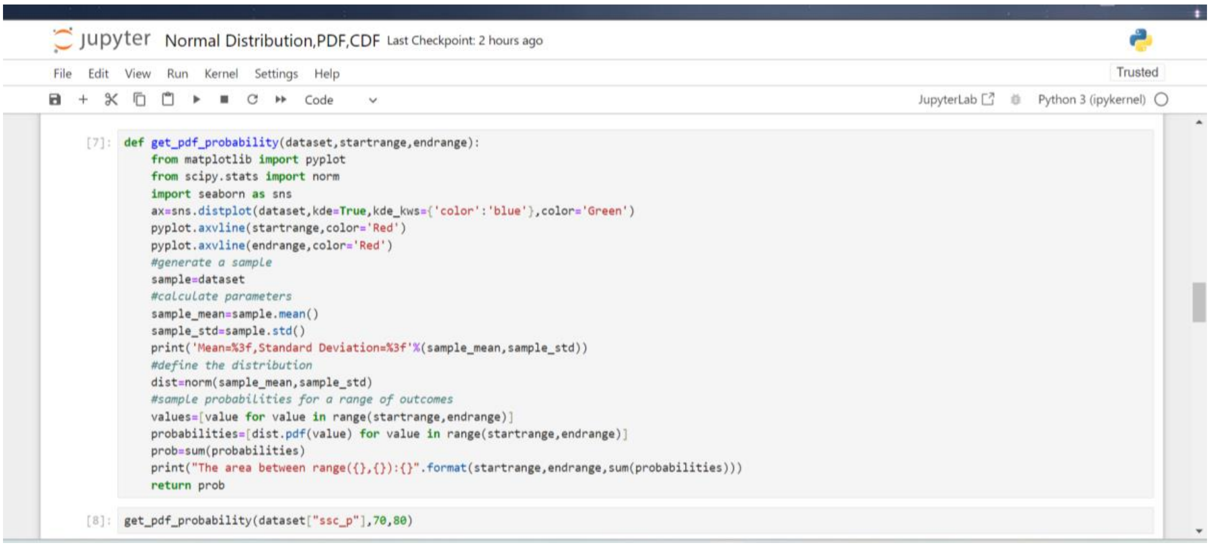
PROBABILITY DENSITY FUNCTION

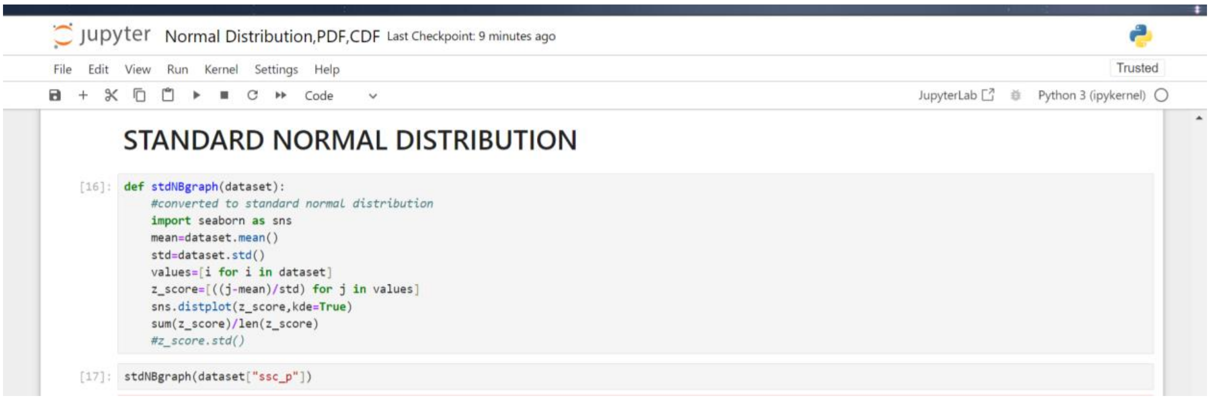

Jupyter Normal Distribution,PDF,CDF Last Checkpoint: 2 hours ago
File Edit View Run Kernel Settings Help
Code
Trusted
JupyterLab
Python 3 (ipykernel)

```
[7]: def get_pdf_probability(dataset,startrange,endrange):
         from matplotlib import pyplot
         from scipy.stats import norm
         import seaborn as sns
         ax=sns.distplot(dataset,kde=True,kde_kws={'color':'blue'},color='Green')
         pyplot.axvline(startrange,color='Red')
         pyplot.axvline(endrange,color='Red')
         #generate a sample
         sample=dataset
         #calculate parameters
         sample_mean=sample.mean()
         sample_std=sample.std()
         print('Mean=%3f,Standard Deviation=%3f'%(sample_mean,sample_std))
         #define the distribution
         dist=norm(sample_mean,sample_std)
         #sample probabilities for a range of outcomes
         values=[value for value in range(startrange,endrange)]
         probabilities=[dist.pdf(value) for value in range(startrange,endrange)]
         prob=sum(probabilities)
         print("The area between range({},{}):{}".format(startrange,endrange,sum(probabilities)))
         return prob

[8]: get_pdf_probability(dataset["ssc_p"],70,80)
```

1. Create a function for Probability density function
2. Import required libraries – matplotlib, scipy.stats & seaborn
3. Using distplot function, set colours for the required curves as blue and green respectively.
4. Using axvline function, draw vertical lines for start range and end range.
5. Generate a sample as dataset
6. Calculate mean and standard deviation for the required column from the dataset.
7. Print those mean and standard deviation values.
8. Using norm function, calculate the normal distribution with calculated mean and standard deviation.
9. Generate a list of values from start range to end range.
10. Finding the pdf value from start range to end range.
11. Sum the total pdf values and print it.
12. Call the function for the required range of start and end.

STANDARD NORMAL DISTRIBUTION


jupyter
Normal Distribution,PDF,CDF Last Checkpoint: 9 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

STANDARD NORMAL DISTRIBUTION

[16]:
```
def stdNBgraph(dataset):
    #converted to standard normal distribution
    import seaborn as sns
    mean=dataset.mean()
    std=dataset.std()
    values=[i for i in dataset]
    z_score=[((j-mean)/std) for j in values]
    sns.distplot(z_score,kde=True)
    sum(z_score)/len(z_score)
    #z_score.std()
```

[17]:
```
stdNBgraph(dataset["ssc_p"])
```

1. Create a function named stdNBgraph.
2. Import required libraries – seaborn
3. Calculate mean and standard deviation for the required column in the dataset.
4. Generate a list of values from the specific column in the dataset.
5. Calculate the z-score value using (X-Mean)/(Standard Deviation) formula.
6. Using distplot function, a graph is plotted.
7. Now z_score is calculated.
8. Resultant graph will be within defined set of values in the x-axis.
9. Curve shape remains the same.
10. X-axis range will be converted to defined det of values.