# Central Limit theorem

The Central Limit Theorem states that the sampling distribution of the sample means approaches a [normal distribution](http://www.statisticshowto.com/probability-and-statistics/normal-distributions/) as the [sample size](http://www.statisticshowto.com/probability-and-statistics/find-sample-size/) gets larger — no matter what the shape of the population distribution. This fact holds especially true for sample sizes over 30. All this is saying is that as you take more samples, especially large ones, your graph of the [*sample means*](http://www.statisticshowto.com/sample-mean/) will look more like a normal distribution.

# Law of Large umbers

A "law of large numbers" is one of several theorems expressing the idea that as the number of trials of a random process increases, the percentage difference between the expected and actual values goes to zero.

**Interview question**

**What do you know about Central Limit Theorem? Can you prove CLT? Applications?**

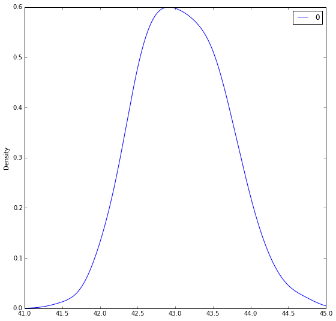
The central limit theorem is one of the most important results of probability theory and serves as the foundation of many methods of statistical analysis. At a high level, the theorem states the distribution of many sample means, known as a sampling distribution, will be normally distributed. This rule holds even if the underlying distribution itself is not normally distributed. As a result we can treat the sample mean as if it were drawn normal distribution.

**# constructing the population**

|  |
| --- |
| np.random.seed(10)  population\_ages1 = stats.poisson.rvs(loc=18, mu=35, size=150000)  population\_ages2 = stats.poisson.rvs(loc=18, mu=10, size=100000)  population\_ages = np.concatenate((population\_ages1, population\_ages2))  population\_ages.mean() |

|  |
| --- |
| np.random.seed(10)  point\_estimates = [] *# Make empty list to hold point estimates*  **for** x **in** range(200): *# Generate 200 samples*  sample = np.random.choice(a= population\_ages, size=500)  point\_estimates.append( sample.mean() )    # Plotting my data  pd.DataFrame(point\_estimates).plot(kind="density", *# Plot sample mean density*  figsize=(9,9),  xlim=(41,45)) |

<matplotlib.axes.\_subplots.AxesSubplot at 0xa664f98>

****

The sampling distribution appears to be roughly normal, despite the bimodal population distribution that the samples were drawn from. In addition, the mean of the sampling distribution approaches the true population mean: