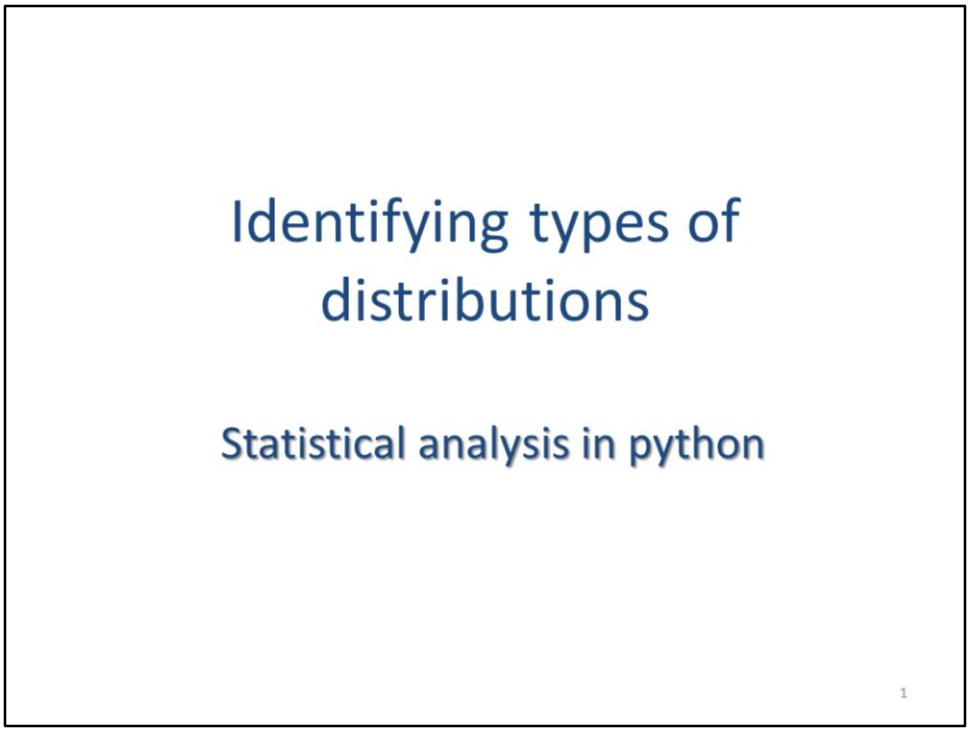
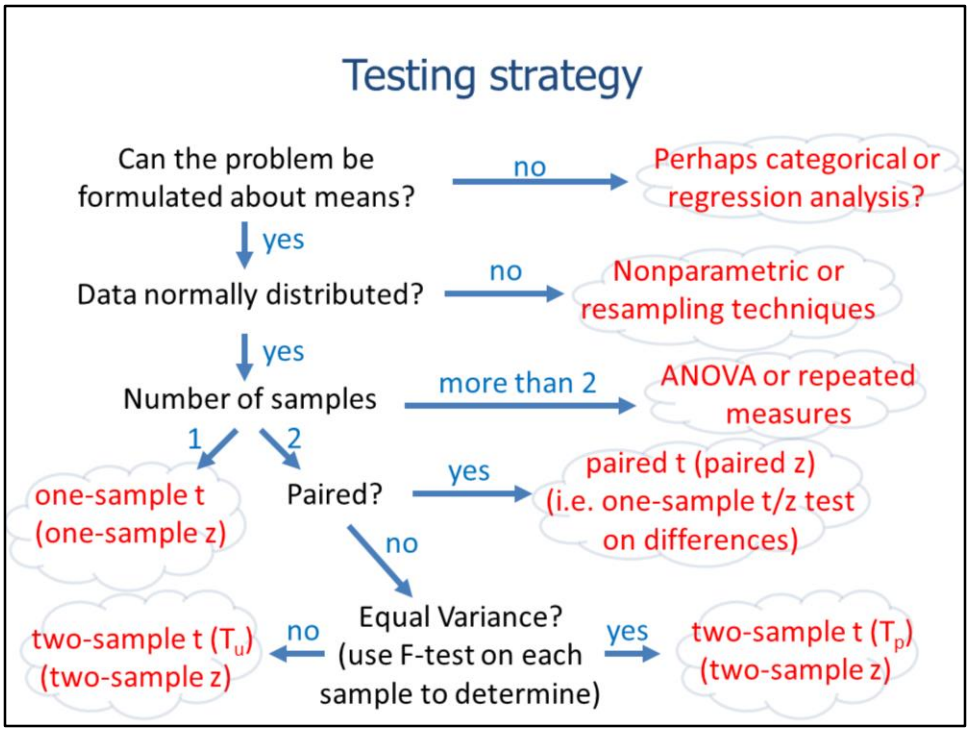


This video will explore scipy's tools for creating random data distributions as well as identifying the type of distribution that corresponds to a given dataset.



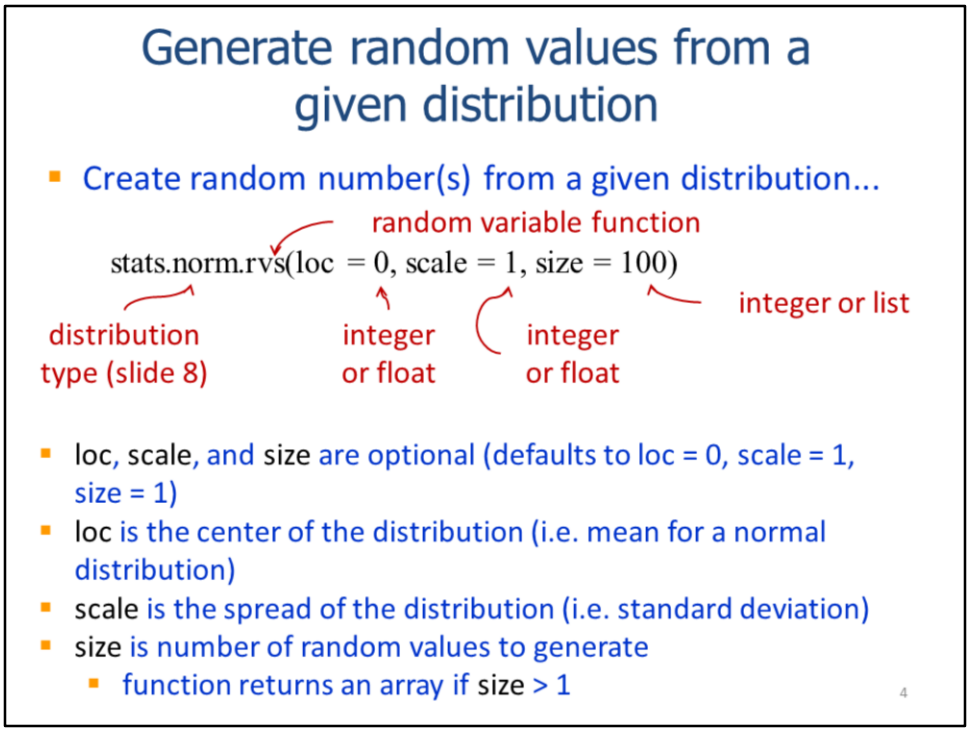
This slide shows a general strategy for determining an appropriate statistical test to use for an analysis. The appropriate test will depend on the type of distribution, the number of samples, and whether the samples are independent. The remainder of this video will discuss scipy's tools for determining the type of distribution that a sample belongs to.

Data distributions in stats module

Distribution	Function
• Normal	• norm
• Uniform	• uniform
• Student's T	• t
• F	• f
• Cauchy	• cauchy
• Chi-squared	• chi2
• Exponential	• expon
• Bernoulli	• bernoulli
• Binomial	• binom
• Geometric	• geom
• Negative binomial	• nbinom
• Poisson	• poisson

3

The stats module can work with data distributions listed on this slide. The name of the toolsets in the stats submodule that correspond to a particular distribution is provided in the right-hand column.



This statement shows how to create a random dataset with a normal distribution.

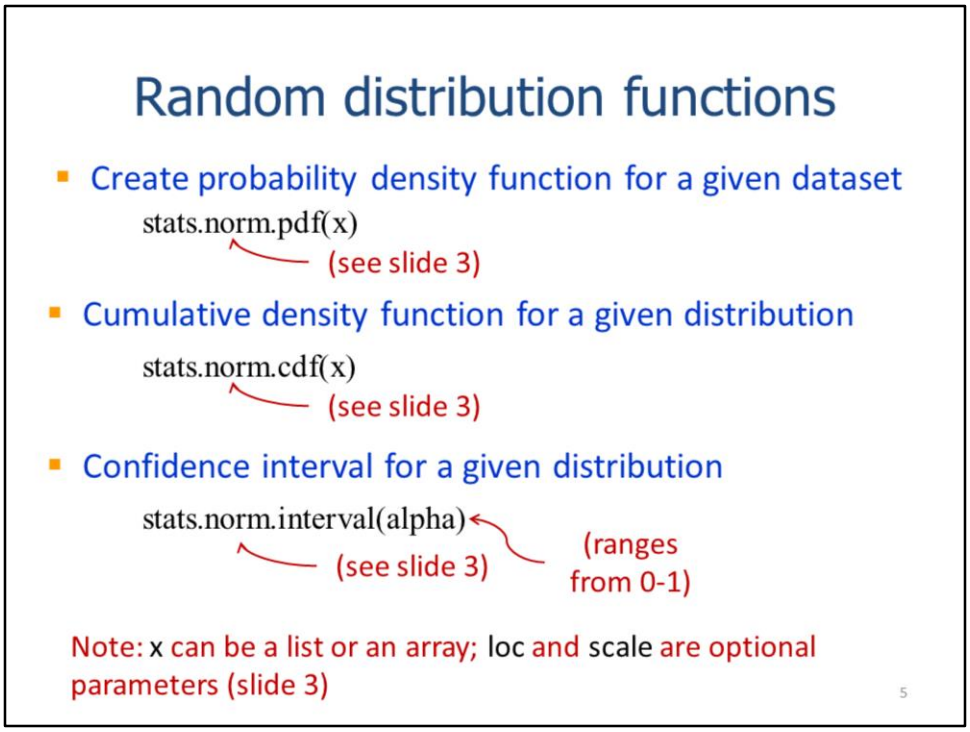
Norm indicates the distribution type.

The **rvs** function creates random values within the specified distribution.

The **loc** parameter specifies the center of the distribution – the center of a normal distribution is the mean.

The **scale** is the spread of the distribution – for a normal distribution, the scale is the standard deviation.

The **size** parameter specifies the number of random values that will be generated.

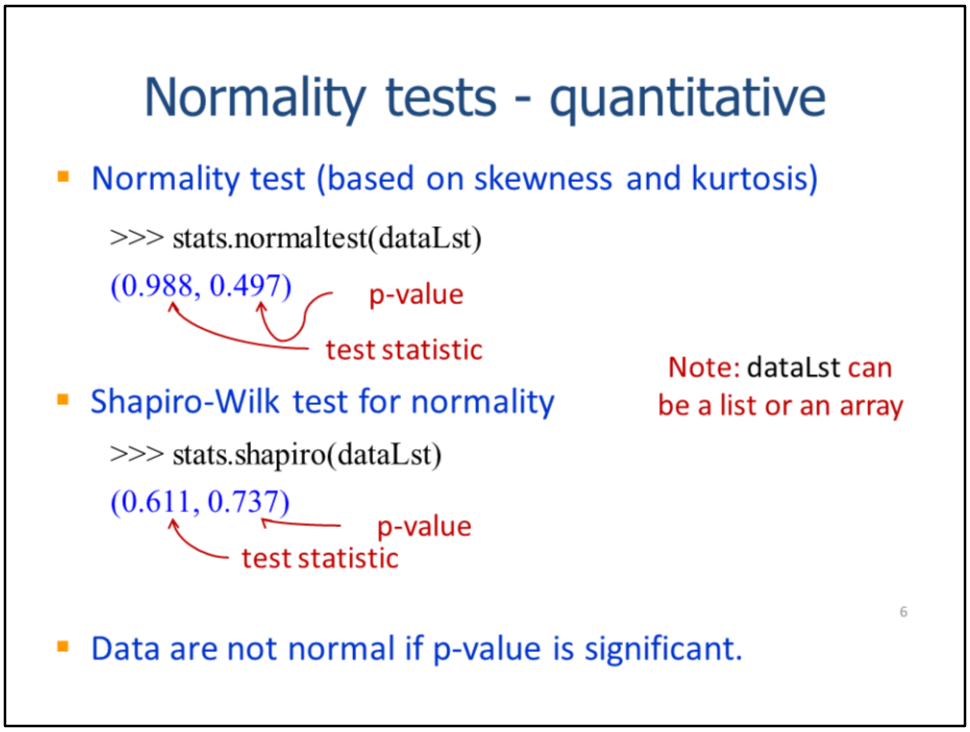


The examples on this slide will use a normal distribution; however, the functions can be used for any distribution listed on slide 3.

The **pdf** tool calculates the probability density function for a list of data points. This indicates the probability that a given value will be randomly picked from the specified distribution. In other words, the probability indicates how likely a value is to belong to the distribution. The further the data point is from the mean, the lower its probability of belonging to the distribution.

The **cdf** tool calculates the cumulative density function for the specified distribution. The cdf indicates the probability that a value from the distribution will be less than the specified value.

The **interval** tool calculates a confidence interval, around the mean, that contains the percentage of the distribution specified by **alpha**. For example, if we have a normal distribution with a mean of 0 and a standard deviation of 1, then a confidence interval for an alpha of 0.95 would be -1.96 to 1.96. This interval contains 95% of the distribution and there is only a 5% chance that a data point outside this interval will belong to the distribution.

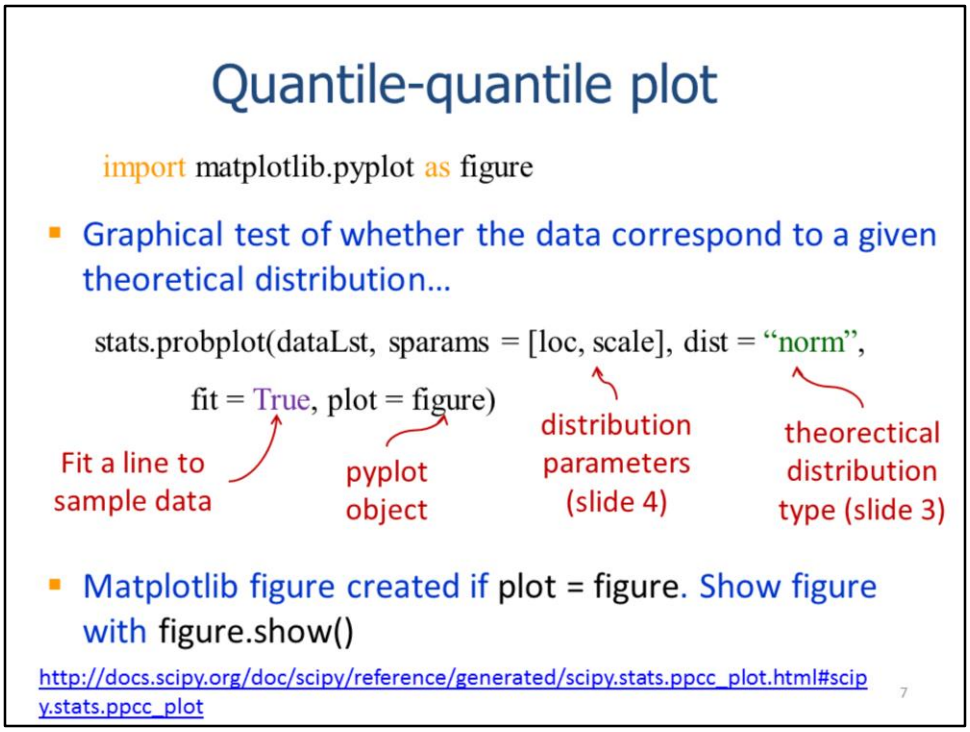


Parametric tests require the sample data to have a normal distribution in order to be valid. The stats module provides several tests for normality. Confidence in the normalcy of the data can be increased by applying multiple tests – if the tests consistently find the data are normal, then the data are most likely normal.

This slide presents two stats module methods for testing normality - the **normaltest** and the **shapiro** test. Both methods return a tuple containing the test statistic and the corresponding p-value.

Note that either a list or an array can be used as the input for these methods.

For both the normaltest and the shapiro test, the distributions are found to be not normal if the p-value is significant. P-values are typically considered significant if they are less than 0.05.

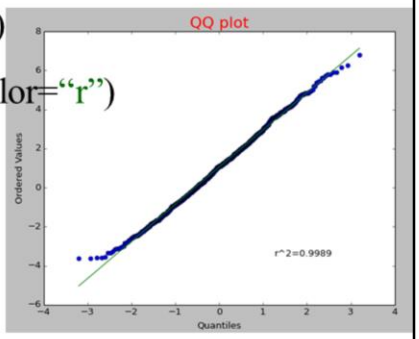


The quantile-quantile plot provides a graphical method for testing the normality of a data distribution. This plot can be created using by importing matplotlib's pyplot module and then using the stat's module's probplot function.

The probplot function will create the plot if the **plot** parameter is set to the pyplot figure. The **figure.show** statement will be needed to display the figure.

Example script: quantile-quantile plot

```
import matplotlib.pyplot as figure
import scipy.stats as stats
dataLst = stats.norm.rvs(scale = 2, size=1000)
stats.probplot(dataLst, plot=figure)
figure.title("QQ plot", size=20, color="r")
figure.show()
pValue=stats.shapiro(dataLst)[1]
if pValue > 0.05:
    print "Dataset is normal at 95% confidence level"
```



8

This slide shows an example script that creates a quantile-quantile plot.

The pyplot and stats modules are imported.

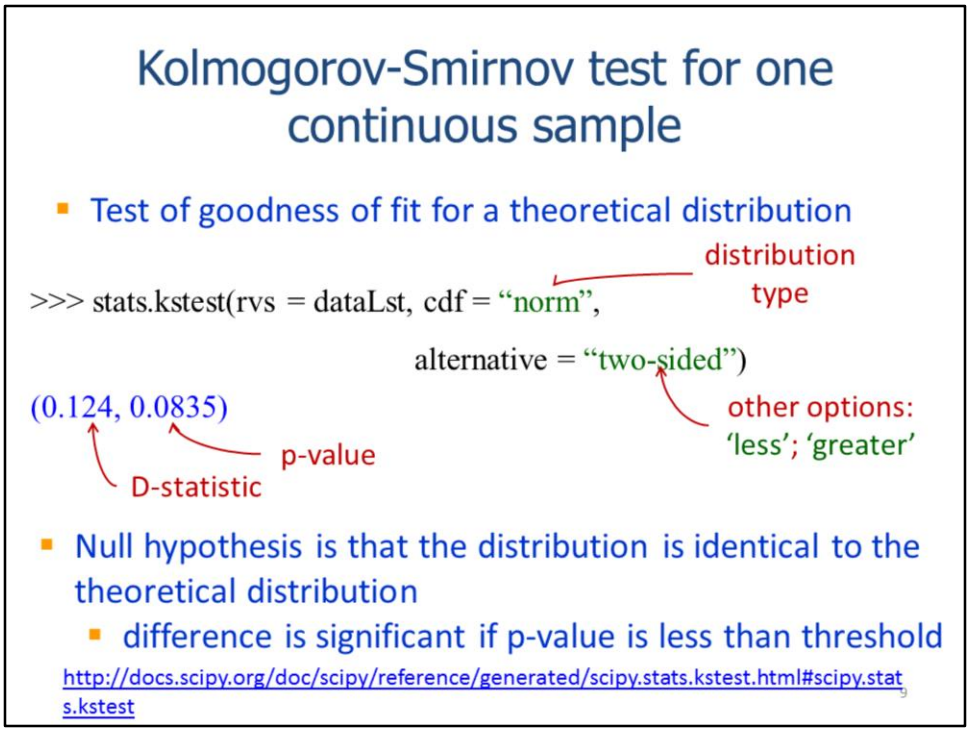
The stats module is used to create an array of 1000 random values sampled from a normal distribution that has a mean of 0 and a standard deviation of 2.

The **probplot** method is used to create the plot in the pyplot figure.

The figure title is set and the figure is shown.

The **shapiro** test is calculated to confirm the quantile-quantile test.

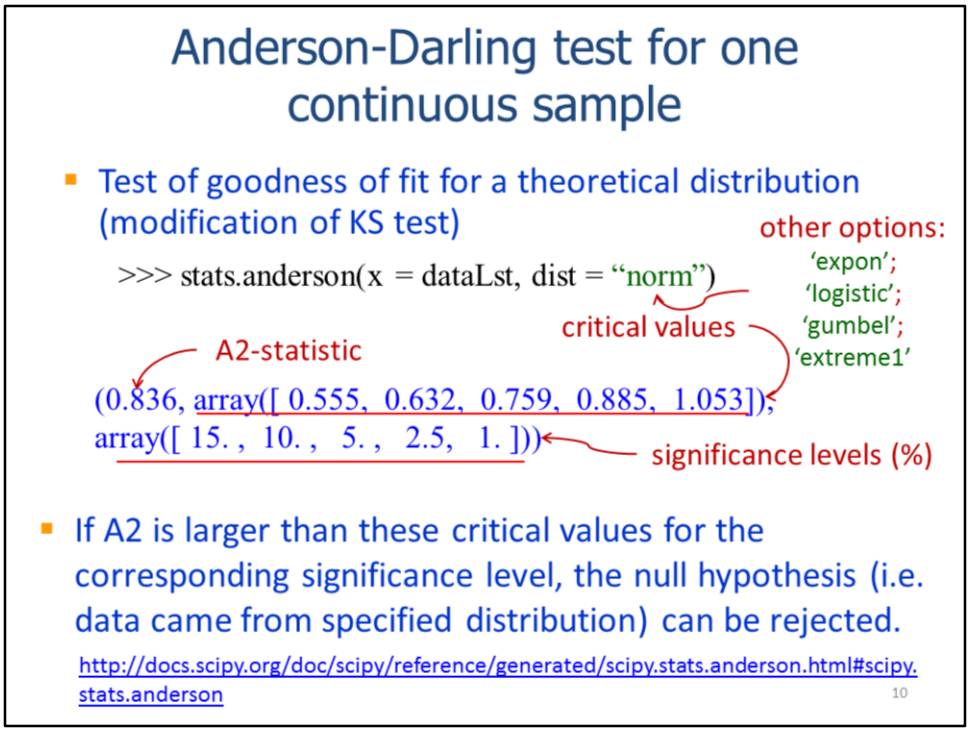
If the shapiro test p-value is > 0.05, then the result is not significant and the dataset is found to be normal.



The KS test will test how well a dataset fits a given theoretical distribution.

The stats module's **kstest** can test the fit of the dataLst to any of the distributions listed on slide 3. In this example, a normal distribution was specified so the **kstest** serves as an additional normality test.

The null hypothesis for the kstest is that the distribution is identical to the theoretical distribution so a significant p-value indicates the distribution does not match the theoretical distribution.



The **Anderson** test also tests the fit of a dataset to a specified theoretical distribution. This test can be used for the distribution types listed here.

The result from the test is a tuple that contains the A2 test statistic, an array with critical values, and an array with significance levels.

If the test statistic is larger than a given critical value, then the test is significant at that level. A significant test indicates that the dataset does not fit the theoretical distribution. In the results for the example statement, the A2 test statistic of 0.836 is larger than the critical value (0.759) at the 95% confidence level but smaller than the critical value (0.885) at the 97.5% confidence level. Therefore, we can conclude with 95% certainty that the dataset does not match the theoretical distribution.