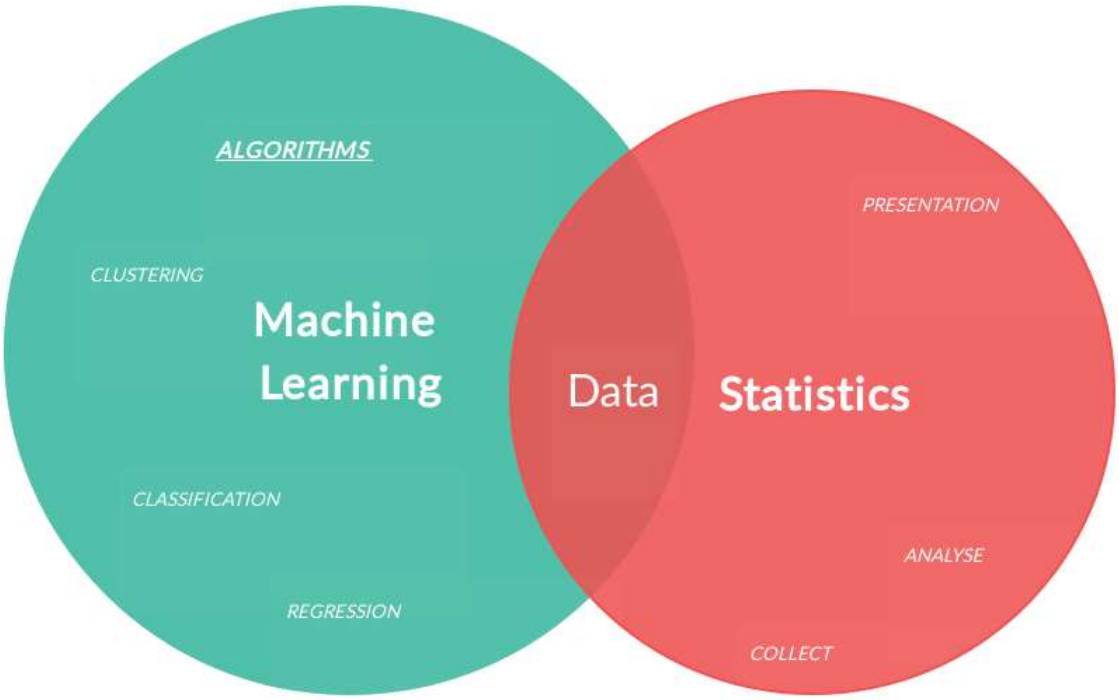
Statistical Hypothesis Tests in Python



Notebook Content:

In [52]:
```
import warnings
warnings.filterwarnings('ignore')
```

In order to carry out any Machine learning Projects, Probability and Statistics plays a major role.Probability deals with predicting the likelihood of future events however Statistics deals with analyse frequency of past events.

Normality Tests

Main obejctive of performing Normality Tests is to validate the Gaussian distribution of data.

Shapiro-Wilk Test

Tests whether a data sample has a Gaussian distribution.

Assumption

Observations in each sample are independent and distributed identically.

Hypothesis

- H0: the sample has a Gaussian distribution.
- H1: the sample does not have a Gaussian distribution.

Scipy Ref ->

In [53]:
```
from scipy.stats import shapiro
data = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
stat, p = shapiro(data)
print('stat={0:.3f}, p={0:.3f}' .format(stat, p))
if p > 0.05:
    print('Probably Gaussian')
else:
    print('Probably not Gaussian')
```

```
stat=0.895, p=0.895
Probably Gaussian
```

D'Agostino's K^2 Test

Tests whether a data sample has a Gaussian distribution.

Assumption

Observations in each sample are independent and distributed identically.

Hypothesis

- H0: the sample has a Gaussian distribution.
- H1: the sample does not have a Gaussian distribution.

Scipy Ref ->

In [54]:
```
# Example of the D'Agostino's K^2 Normality Test
from scipy.stats import normaltest
data = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
stat, p = normaltest(data)
print('stat={0:.3f}, p={0:.3g}'.format(stat, p))
if p > 0.05:
    print('Probably Gaussian')
else:
    print('Probably not Gaussian')
```

```
stat=3.392, p=3.39
Probably Gaussian
```

Anderson-Darling Test

Tests whether a data sample has a Gaussian distribution.

Assumption

Observations in each sample are independent and distributed identically.

Hypothesis

- H0: the sample has a Gaussian distribution.
- H1: the sample does not have a Gaussian distribution.

Scipy Ref ->

In [55]:
```python
# Example of the Anderson-Darling Normality Test
from scipy.stats import anderson
data = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
result = anderson(data)
print('stat={0:.3g}'.format(result.statistic))
for i in range(len(result.critical_values)):
    sl, cv = result.significance_level[i], result.critical_values[i]
    if result.statistic < cv:
        print('Probably Gaussian at the %.1f%% level' % (sl))
    else:
        print('Probably not Gaussian at the %.1f%% level' % (sl))
```

```
stat=0.424
Probably Gaussian at the 15.0% level
Probably Gaussian at the 10.0% level
Probably Gaussian at the 5.0% level
Probably Gaussian at the 2.5% level
Probably Gaussian at the 1.0% level
```

Correlation Tests

Correlation Tests are used to check the correlation between two independent features or variables.

Pearson's Correlation Coefficient

Tests whether a data sample is linearly separable.

Assumption

a) Observations in each sample are independent and distributed identically.
b) Observations are normally distributed.
c) Similar variance between independent variables

Hypothesis

- H0: the samples are correlated.
- H1: the sample does not have any correlation.

Scipy Ref ->

In [56]:
```
# Example of the Pearson's Correlation test
from scipy.stats import pearsonr
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [0.353, 3.517, 0.125, -7.545, -0.555, -1.536, 3.350, -1.578, -3.537, -1.579]
stat, p = pearsonr(data1, data2)
print('stat={0:.3f}, p={0:.3f}'.format(stat, p))
if p > 0.05:
    print('Probably independent')
else:
    print('Probably dependent')
```

```
stat=0.688, p=0.688
Probably dependent
```

Spearman's Rank Correlation

Tests whether a data sample is montonically separable.

Assumption

a) Observations in each sample are independent and distributed identically.
b) Observations in each sample are ranked .

Hypothesis

- H0: the samples are correlated.
- H1: the sample does not have any correlation.

Scipy Ref ->

In [57]:
```
# Example of the Spearman's Rank Correlation Test
from scipy.stats import spearmanr
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [0.353, 3.517, 0.125, -7.545, -0.555, -1.536, 3.350, -1.578, -3.537, -1.579]
stat, p = spearmanr(data1, data2)
```

```
print('stat={0:.3g}, p={0:.3f}'.format(stat, p))
if p > 0.05:
    print('Probably independent')
else:
    print('Probably dependent')
```

```
stat=0.855, p=0.855
Probably dependent
```

Kendall's Rank Correlation

Tests whether a data sample is montonically separable.

Assumption

a) Observations in each sample are independent and distributed identically.

b) Observations in each sample are ranked .

Hypothesis

- H0: the samples are correlated.
- H1: the sample does not have any correlation.

Scipy Ref ->

In [58]:
```
# Example of the Kendall's Rank Correlation Test
from scipy.stats import kendalltau
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [0.353, 3.517, 0.125, -7.545, -0.555, -1.536, 3.350, -1.578, -3.537, -1.579]
stat, p = kendalltau(data1, data2)
print('stat={0:.3f}, p={0:.3f}'.format(stat, p))
if p > 0.05:
    print('Probably independent')
else:
    print('Probably dependent')
```

```
stat=0.733, p=0.733
Probably dependent
```

Chi-Squared Test

Tests whether two categorical variables are related to each other.

Assumption

a) Observations in used in contengency table are Independent.

b) There are more than 25 examples in contengency table .

Hypothesis

- H0: the samples are correlated.
- H1: the sample does not have any correlation.

Scipy Ref ->

In [59]:
```
# Example of the Chi-Squared Test
from scipy.stats import chi2_contingency
table = [[10, 20, 30],[6,  9,  17]]
stat, p, dof, expected = chi2_contingency(table)
print('stat={0:.3g}, p={0:.3f}' .format(stat, p))
if p > 0.05:
    print('Probably independent')
else:
    print('Probably dependent')
```

```
stat=0.272, p=0.272
Probably independent
```

Stationary Tests

Used for Validating the Time series data trends(Stationary/Not-Stationary).

Augmented Dickey-Fuller Unit Root Test

Tests whether a Time series data has autoregressive trend.

Assumption

Data Instance have temporality.

Hypothesis

- H0: the unit root is present.
- H1: the unit root not present.

Stats-Model Ref ->

In [60]:
```
# Example of the Augmented Dickey-Fuller unit root test
from statsmodels.tsa.stattools import adfuller
data = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
stat, p, lags, obs, crit, t = adfuller(data)
print('stat={0:.3f}, p={0:.3f}'.format(stat, p))
if p > 0.05:
    print('Probably not Stationary')
else:
    print('Probably Stationary')
```

```
stat=0.992, p=0.992
Probably not Stationary
```

Kwiatkowski-Phillips-Schmidt-Shin

Tests whether a Time series trend is stationary or not.

Assumption

Data Instance have temporality.

Hypothesis

- H0: the stationarity is present.
- H1: the stationarity not present.

Stats-Model Ref ->

In [61]:
```
# Example of the Kwiatkowski-Phillips-Schmidt-Shin test
from statsmodels.tsa.stattools import kpss
data = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
stat, p, lags, crit = kpss(data)
print('stat={0:.3g}, p={0:.3g}'.format(stat, p))
if p > 0.05:
    print('Probably not Stationary')
else:
    print('Probably Stationary')
```

```
stat=0.41, p=0.41
Probably not Stationary
```

Parametric Statistical Hypothesis Tests

Statistical Test for comaparison between data samples.

Student's t-test

Average between two data samples are significantly different.

Assumption

a)Each data sample's observation are independent and distributed.
b)Observations are normally distributed.
c)Observations have same variance between each other.

Hypothesis

- H0: the mean between two samples are equal .
- H1: the mean between two samples are not equal.

Scipy Ref ->

In [62]:
```
# Example of the Student's t-test
from scipy.stats import ttest_ind
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169
stat, p = ttest_ind(data1, data2)
print('stat={0:.3f}, p={0:.3f}'.format(stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

```
stat=-0.326, p=-0.326
Probably the same distribution
```

Paired Student's t-test

Average between two data samples are significantly different.

Assumption

a)Each data sample's observation are independent and distributed.
b)Observations are normally distributed.
c)Observations have same variance between each other.
d)Observations are paired.

Hypothesis

- H0: the mean between two samples are equal .
- H1: the mean between two samples are not equal.

Scipy Ref ->

In [63]:
```
# Example of the Paired Student's t-test
from scipy.stats import ttest_rel
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169
stat, p = ttest_rel(data1, data2)
print('stat={0:.3f}, p={0:.3f}'.format(stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

```
stat=-0.334, p=-0.334
Probably the same distribution
```

Analysis of Variance Test (ANOVA)

Average between two data samples are significantly independent and different.

Assumption

a)Each data sample's observation are independent and distributed.
b)Observations are normally distributed.
c)Observations have same variance between each other.

Hypothesis

- H0: the mean between two samples are equal .
- H1: the mean between two samples are not equal.

Scipy Ref ->

In [64]:
```
# Example of the Analysis of Variance Test
from scipy.stats import f_oneway
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169
data3 = [-0.208, 0.696, 0.928, -1.148, -0.213, 0.229, 0.137, 0.269, -0.870, -1.204]
stat, p = f_oneway(data1, data2, data3)
print('stat={0:.3g}, p={0:.3g}'.format(stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

```
stat=0.0964, p=0.0964
Probably the same distribution
```

Nonparametric Statistical Hypothesis Tests

Mann-Whitney U Test

Distribution of two data samples are equal or not.

Assumption

a)Each data sample's observation are independent and distributed.
b)Observations in each data samples can be ranked.

Hypothesis

- H0: the distribution of two samples are equal .
- H1: the distribution of two samples are not equal.

Scipy Ref ->

In [65]:
```
# Example of the Mann-Whitney U Test
from scipy.stats import mannwhitneyu
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169
stat, p = mannwhitneyu(data1, data2)
print('stat={0:.3g}, p={0:.3g}'.format(stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

```
stat=40, p=40
Probably the same distribution
```

Wilcoxon Signed-Rank Test

Distribution between two paired samples are significantly equal or not.

Assumption

a)Each data sample's observation are independent and distributed.
b)Observations can be ranked.
c)Observations are paired.

Hypothesis

- H0: the distribution of two samples are equal .
- H1: the distribution of two samples are not equal.

Scipy Ref ->

In [66]:
```
# Example of the Wilcoxon Signed-Rank Test
from scipy.stats import wilcoxon
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169
stat, p = wilcoxon(data1, data2)
print('stat={0:.3g}, p={0:.3g}' .format (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

```
stat=21, p=21
Probably the same distribution
```

Kruskal-Wallis H Test

Distribution between two independent samples are significantly equal or not.

Assumption

a)Each data sample's observation are independent and distributed.
b)Observations can be ranked.

Hypothesis

- H0: the distribution of samples are equal .
- H1: the distribution of samples are not equal.

Scipy Ref ->

In [67]:
```
# Example of the Kruskal-Wallis H Test
from scipy.stats import kruskal
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169
stat, p = kruskal(data1, data2)
print('stat={0:.3g}, p={0:.3g}'.format(stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

```
stat=0.571, p=0.571
Probably the same distribution
```

Friedman Test

Distribution between two paired samples are significantly equal or not.

Assumption

a)Each data sample's observation are independent and distributed.
b)Observations can be ranked.
c)Observations can be paired.

Hypothesis

- H0: the distribution of all samples are equal .

- H1: the distribution of one or more samples are not equal.

Scipy Ref ->

In [68]:
```
# Example of the Friedman Test
from scipy.stats import friedmanchisquare
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169
data3 = [-0.208, 0.696, 0.928, -1.148, -0.213, 0.229, 0.137, 0.269, -0.870, -1.204]
stat, p = friedmanchisquare(data1, data2, data3)
print('stat={0:.3g}, p={0:.3f}'.format(stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

```
stat=0.8, p=0.800
Probably the same distribution
```

Above some of the key Statistical Tests that can be used in any Machine learning Projects.These test can be used in normality validation, establishing relationships between variables, and differences between samples.

-> Weblink

In [ ]: