We can use this test, if we observe two independent samples from the same or different population, e.g. exam scores of boys and girls or of two ethnic groups. The test measures whether the average (expected) value differs significantly across samples. If we observe a large p-value, for example larger than 0.05 or 0.1, then we cannot reject the null hypothesis of identical average scores. If the p-value is smaller than the threshold, e.g. 1%, 5% or 10%, then we reject the null hypothesis of equal averages.

The test results in **bold are not significant**  because the pvalue is greater than 0.05

The null hypothesis is that the average value for a hit is the same as that in the population

p **=** stats.ttest\_ind(df\_clean\_control['popularity'], df\_clean['popularity'])

p

Out[14]:

Ttest\_indResult(statistic=6.018939611631809, pvalue=5.201097005648276e-09)

In [15]:

k **=** stats.ttest\_ind(df\_clean['key'], df\_clean\_control['key'])

k

Out[15]:

**Ttest\_indResult(statistic=0.10743349944316466, pvalue=0.9145181075229887)**

In [16]:

d **=** stats.ttest\_ind(df\_clean\_control['danceability'], df\_clean['danceability'])

d

Out[16]:

Ttest\_indResult(statistic=-4.643781590685511, pvalue=5.1574884301576395e-06)

In [17]:

e **=** stats.ttest\_ind(df\_clean['energy'], df\_clean\_control['energy'])

e

Out[17]:

**Ttest\_indResult(statistic=0.6260636007941561, pvalue=0.5317573213293452)**

In [18]:

l **=** stats.ttest\_ind(df\_clean\_control['loudness'], df\_clean['loudness'])

l

Out[18]:

Ttest\_indResult(statistic=-3.6349232828783524, pvalue=0.0003279651339519008)

In [19]:

m **=** stats.ttest\_ind(df\_clean['mode'], df\_clean\_control['mode'])

m

Out[19]:

**Ttest\_indResult(statistic=-0.6036969470111998, pvalue=0.5465090637984527)**

In [20]:

s **=** stats.ttest\_ind(df\_clean\_control['speechiness'], df\_clean['speechiness'])

s

Out[20]:

**Ttest\_indResult(statistic=-0.2598935269514805, pvalue=0.7951271962680867)**

In [21]:

​

a **=** stats.ttest\_ind(df\_clean['acousticness'], df\_clean\_control['acousticness'])

a

Out[21]:

Ttest\_indResult(statistic=-2.5331307173185147, pvalue=0.011822947071933121)

In [22]:

po **=** stats.ttest\_ind(df\_clean\_control['popularity'], df\_clean['popularity'])

po

Out[22]:

**Ttest\_indResult(statistic=6.018939611631809, pvalue=5.201097005648276e-09)**

In [23]:

i **=** stats.ttest\_ind(df\_clean['instrumentalness'], df\_clean\_control['instrumentalness'])

i

Out[23]:

Ttest\_indResult(statistic=-2.585243901911141, pvalue=0.010210891224437897)

In [24]:

li **=** stats.ttest\_ind(df\_clean\_control['liveness'], df\_clean['liveness'])

li

Out[24]:

Ttest\_indResult(statistic=2.3598910661309223, pvalue=0.018931388415099608)

In [25]:

v **=** stats.ttest\_ind(df\_clean['valence'], df\_clean\_control['valence'])

v

Out[25]:

**Ttest\_indResult(statistic=1.0664188201315805, pvalue=0.28710634012291836)**

In [26]:

t **=** stats.ttest\_ind(df\_clean\_control['tempo'], df\_clean['tempo'])

t

Out[26]:

**Ttest\_indResult(statistic=1.4642766390382267, pvalue=0.14418318769064847**)

In [27]:

du **=** stats.ttest\_ind(df\_clean['duration\_ms'], df\_clean\_control['duration\_ms'])

du

Out[27]:

**Ttest\_indResult(statistic=-0.10502979433636993, pvalue=0.9164235432175388)**

In [28]:

​

ti **=** stats.ttest\_ind(df\_clean['time\_signature'], df\_clean\_control['time\_signature'])

ti

Out[28]:

**Ttest\_indResult(statistic=1.6912004279629755, pvalue=0.0918546429775687)**