

▼ KS-Test

```
from scipy import stats
import numpy as np
import matplotlib.pyplot as plt
```

```
# recovery times of patients who took medicine-1
```

```
r1 = [8.82420842, 7.47774471, 7.55712098, 7.98131439, 6.82771606,
      7.48566433, 9.15385732, 5.84040502, 8.26124313, 8.4728876 ,
      6.82582186, 7.00490974, 8.43423058, 6.72099932, 6.97495982,
      5.93748053, 5.40707847, 6.16385557, 6.71421056, 4.42396183,
      6.87285228, 8.00313581, 6.69035041, 7.83622942, 8.70984957,
      5.56284584, 9.08093437, 4.98165193, 7.67769408, 6.04738478,
      7.64921582, 7.31051639, 6.74463303, 7.27356973, 8.16787232,
      6.90990965, 7.06439167, 6.62921957, 6.08283539, 6.2458137 ,
      8.65173634, 5.76080646, 6.20573219, 8.91561004, 6.22560201,
      5.67542104, 6.97412435, 8.31354697, 8.14172701, 8.26099345,
      7.87612791, 6.24835109, 9.95324783, 6.59504627, 6.17365145,
      6.05676895, 7.23030223, 7.71311809, 7.37163804, 5.69798738,
      5.71056902, 7.94556876, 7.47234105, 6.85346234, 4.77892053,
      6.92631063, 6.10681151, 7.06277198, 7.18023164, 7.78285327,
      7.85500885, 6.54349161, 8.25949958, 6.44289198, 7.16705977,
      6.03517015, 7.61274786, 7.032845 , 6.78161745, 7.07917968,
      6.21549342, 5.34267439, 6.73039933, 7.70562561, 8.15117049,
      6.72564324, 6.68220904, 8.50359274, 7.52912703, 7.34572493,
      5.95734283, 6.58259396, 6.49394335, 8.68069592, 8.60547125,
      6.8905056 , 7.72575925, 6.84801609, 7.96999724, 7.10420915]
```

```
# recovery times of patietnts who took medicine-2
```

```
r2 = [ 9.56597358, 7.49291458, 8.73841824, 7.63523452, 4.12559277,
      7.3679259 , 9.87873565, 6.14516559, 8.19923821, 7.30169992,
      10.24606417, 6.83814477, 7.01611267, 6.15716049, 8.29590714,
      12.3333305 , 8.22144016, 6.06830071, 3.75820649, 6.69220157,
      10.08721618, 9.70580422, 7.31050006, 11.40145721, 5.64818498,
      7.38914449, 8.43740074, 6.3451435 , 7.05694361, 8.1997151 ,
      9.03059061, 7.76904679, 6.92375578, 5.78318543, 8.99027781,
      7.56186529, 5.27095372, 8.32896688, 11.52935757, 7.08119961,
      9.48825066, 9.14072759, 7.30357663, 8.62183754, 10.40999814,
      8.70096763, 7.04645384, 6.378799 , 10.5098363 , 7.36078888,
      7.33403615, 8.07396248, 6.18309499, 7.24668404, 9.03430611,
      8.99016584, 6.78606416, 8.436418 , 6.85877947, 10.10405772,
      6.74943076, 7.57812376, 7.12920671, 9.38065269, 9.57139966,
      6.4484012 , 6.93877043, 9.22141667, 8.34815638, 7.73980671,
      7.17840767, 9.27913457, 6.49963224, 9.92287292, 7.63978639,
      9.53931977, 9.02602273, 6.79374185, 8.59715131, 8.37747338,
      8.78161815, 6.78716383, 8.28473394, 8.20283798, 12.50518811,
      10.19772574, 8.93758457, 8.9540311 , 8.28927558, 6.28935098,
      7.69447559, 9.66777701, 10.33898342, 8.71199578, 5.12781581,
      9.70954569, 9.13685031, 7.28989718, 8.0868909 , 7.42937556,
      7.31356749, 9.92345816, 8.60211814, 9.33228465, 8.14132658,
```

```
6.17871495, 10.28358242, 7.31898597, 7.95085527, 6.20331719,
9.19119762, 6.98600628, 7.05314883, 10.57921482, 6.83637574,
7.86199283, 8.23350975, 5.87625665, 7.78945364, 8.83612492]
```

```
d1 = np.array(r1)
d2 = np.array(r2)
```

```
n1 = len(d1)
n2 = len(d2)
```

```
n1
```

```
100
```

```
n2
```

```
120
```

```
#2-sample KS Test
stats.ks_2samp(d1, d2)
```

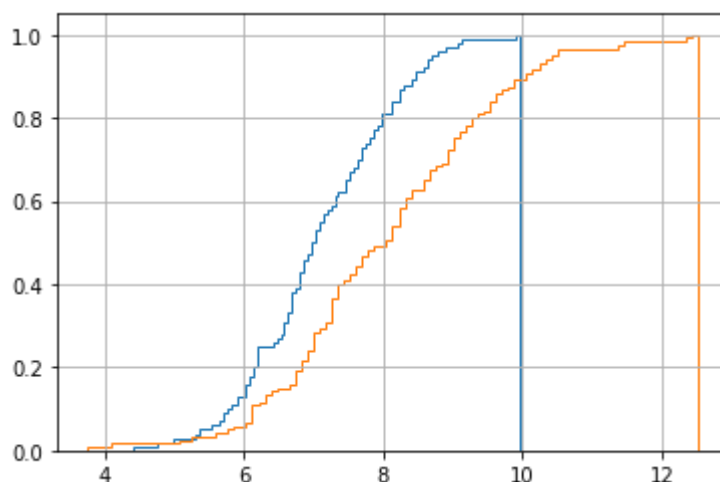
```
KstestResult(statistic=0.3233333333333333, pvalue=1.5163387982131127e-05)
```

p-value = 0.00001516 < 0.001 = 0.1% = alpha

=> Reject H0 (r1 and r2 same distribution)

=> Accept Ha (the distributions of r1 and r2 are different)

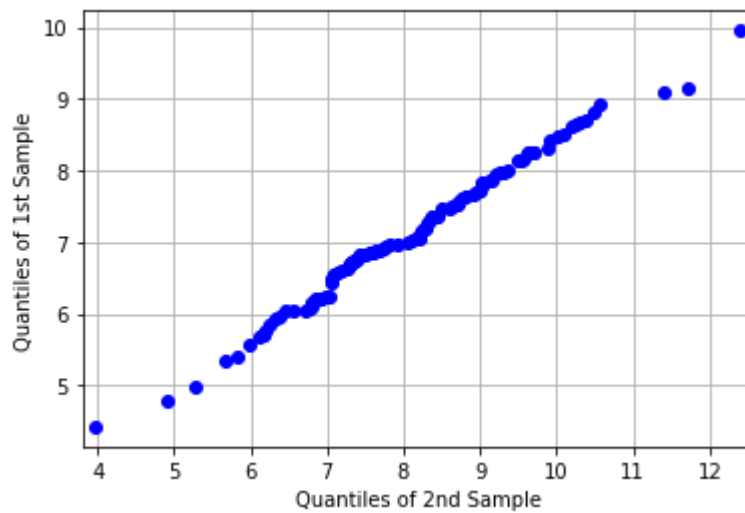
```
plt.grid()
a = plt.hist(d1, bins=100, cumulative=True, label='CDF', density=True, histtype='st
b = plt.hist(d2, bins=100, cumulative=True, label='CDF', density=True, histtype='st
plt.show()
```



```
from statsmodels.graphics.gofplots import qqplot_2samples
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning:
import pandas.util.testing as tm
```

```
qqplot_2samples(d1, d2)
plt.grid()
```



Double-click (or enter) to edit

▼ Z-test

```
# Group A --> Treatment Group shown 2 ads per ad-break
# Group B --> Control Group shown only 1 ad per ad break
# Let us compare mean watch-times per group
# H0: mu1 = mu2
# H1: mu1 != mu2
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
import scipy
```

```
# Download data
# https://drive.google.com/file/d/1H196n6BWdl3ruJgCo_gaAWEb0kEYg__H/view?usp=sharing
id = "1H196n6BWdl3ruJgCo_gaAWEb0kEYg__H"
path = "https://docs.google.com/uc?export=download&id=" + id
print(path)
```

https://docs.google.com/uc?export=download&id=1H196n6BWdl3ruJgCo_gaAWEb0kEYg__H

```
!wget "https://docs.google.com/uc?export=download&id=1H196n6BWdl3ruJgCo_gaAWEb0kEYg__H"
```

--2022-07-15 09:58:22-- https://docs.google.com/uc?export=download&id=1H196n6BWdl3ruJgCo_gaAWEb0kEYg__H

```

Resolving docs.google.com (docs.google.com)... 142.250.98.138, 142.250.98.139,
Connecting to docs.google.com (docs.google.com)|142.250.98.138|:443... connect
HTTP request sent, awaiting response... 303 See Other
Location: https://doc-00-ag-docs.googleusercontent.com/docs/securesc/ha0ro937c
Warning: wildcards not supported in HTTP.
--2022-07-15 09:58:23-- https://doc-00-ag-docs.googleusercontent.com/docs/sec
Resolving doc-00-ag-docs.googleusercontent.com (doc-00-ag-docs.googleusercontent.com)...
Connecting to doc-00-ag-docs.googleusercontent.com (doc-00-ag-docs.googleusercontent.com)...
HTTP request sent, awaiting response... 200 OK
Length: 887610 (867K) [text/csv]
Saving to: 'ab_test_data.csv'

```

```

ab_test_data.csv      100%[=====>] 866.81K  --.-KB/s    in 0.007s

```

```

2022-07-15 09:58:23 (120 MB/s) - 'ab_test_data.csv' saved [887610/887610]

```

```
!ls -lrt
```

```

total 872
drwxr-xr-x 1 root root 4096 Jul 13 13:43 sample_data
-rw-r--r-- 1 root root 887610 Jul 15 09:58 ab_test_data.csv

```

```
!cat ab_test_data.csv
```

```

2018-04-24,672,0,1.204038745515143,treatment
2018-10-27,735,0,1.720859204825507,treatment
2018-03-21,756,0,1.6596750020799753,treatment
2018-08-09,665,0,1.6993040851610293,treatment
2018-05-31,662,0,2.156321897846803,treatment
2018-05-28,267,0,1.2101014751056542,control
2018-12-05,157,1,3.262553148000151,control
2018-06-20,921,0,2.227408455175863,treatment
2018-08-14,143,0,3.190808386607947,control
2018-08-29,17,0,1.8798011760876796,control

2018-01-16,628,0,0.67496075296494,treatment
2018-08-30,621,0,10.644234566020007,treatment
2018-02-11,832,0,1.0737883660596945,treatment
2018-01-09,166,0,6.257587343737802,control
2018-03-21,570,0,2.692056020667736,treatment
2018-03-06,205,0,3.9752558540897067,control
2018-12-22,782,0,2.0935524794099583,treatment
2018-08-22,282,1,2.7272375275111767,control
2018-01-20,835,0,1.6120508745368631,treatment
2018-05-26,410,0,1.485549342880035,control
2018-09-29,175,0,4.27566654286646,control
2018-08-14,203,0,7.08196864590587,control
2018-12-10,151,0,2.0858808654427015,control
2018-07-11,6,0,0.9401265462317687,control
2018-11-05,279,0,3.31996513840014,control
2018-02-13,771,0,1.580129170313505,treatment
2018-11-23,586,0,1.685259132296609,treatment
2018-09-22,232,1,1.1782476633851746,control
2018-12-19,72,0,3.398055397889533,control
2018-07-20,656,1,3.9332518639031524,treatment
2018-06-07,260,0,13.682553899635908,control
2018-06-16,749,0,1.1309293957414697,treatment
2018-06-27,978,0,2.1416438727707994,treatment
2018-12-09,181,0,2.9626500338042097,control

```

```

2018-07-27,56,0,9.21023302816482,control
2018-02-13,314,0,1.4622995603566717,control
2018-04-02,251,1,4.149166635566898,control
2018-08-11,505,0,1.2945839762107971,treatment
2018-11-09,706,0,1.8298824763546142,treatment
2018-12-26,640,0,3.6040346787836106,treatment
2018-04-12,977,0,7.027417795521305,treatment
2018-04-18,41,0,6.8274027837706095,control
2018-05-16,410,0,2.750157888591269,control
2018-04-06,167,0,6.006471278223235,control
2018-08-01,159,0,0.918504336113582,control
2018-03-15,239,0,2.436639260223427,control
2018-10-12,483,0,5.743235539470613,control
2018-01-24,248,0,2.934808120889822,control
2018-09-15,15,1,7.460048995483625,control
2018-10-24,261,0,2.2228976531659312,control
2018-11-20,404,0,5.388538537060598,control
2018-01-10,856,0,4.707772718491784,treatment
2018-03-03,763,1,9.88400616644684,treatment
2018-10-09,794,0,6.157143537539619,treatment
2018-05-21,479,0,4.319866305895602,control
2018-07-20,547,0,2.1242890454930574,treatment
2018-10-23,431,0,0.4521129780822763,control
2018-07-13,538,0,0.43455836190494873,treatment

```

```

ab_test_data = pd.read_csv("ab_test_data.csv")
ab_test_data.sample(100)

```

	date	customer_id	premium	watch_time_hrs	customer_segmnt
15560	2018-12-09	298	0	1.181086	control
11054	2018-12-17	562	1	2.537064	treatment
6255	2018-09-04	732	0	5.251925	treatment
9187	2018-08-24	521	0	9.019050	treatment
12888	2018-07-19	88	0	1.495338	control
...
6432	2018-06-15	612	0	5.108603	treatment
6727	2018-07-27	62	0	2.944847	control
17074	2018-01-11	244	0	7.780596	control
15845	2018-12-22	618	0	1.459799	treatment
10471	2018-12-08	948	0	2.855943	treatment

100 rows × 5 columns

```
ab_test_data.shape
```

```
(20000, 5)
```



```
ab_test_data['customer_segmnt'].value_counts()
# n1=n2=10000 => we can do t-test or z-test to compare means.
```

```
control      10000
treatment    10000
Name: customer_segmnt, dtype: int64
```

```
ab_test_data.describe()
```

	customer_id	premium	watch_time_hrs
count	20000.000000	20000.000000	20000.000000
mean	499.001650	0.176750	9.362542
std	288.223444	0.381467	244.884839
min	0.000000	0.000000	0.160268
25%	249.000000	0.000000	1.678066
50%	500.000000	0.000000	2.670953
75%	747.000000	0.000000	4.204673
max	999.000000	1.000000	10007.648185

```
# remove extreme values as we dont want them to impact means
ab_test_data["watch_time_hrs"].quantile(0.999)
```

```
# NOTE: only 24 hrs in a day
```

```
26.036198684124518
```

```
ab_test_data["watch_time_hrs"].quantile(0.998)
```

```
21.356607722117484
```

```
q998 = ab_test_data["watch_time_hrs"].quantile(0.998)
ab_test_data_no_out = ab_test_data[~(ab_test_data["watch_time_hrs"] > q998)]
```

```
# disb of watch-time
sns.histplot(ab_test_data_no_out['watch_time_hrs'], bins=100)
plt.show()
```



```
#split the data
```

```
ab_test_control_data = ab_test_data_no_out[ab_test_data_no_out["customer_segmnt"] =  
ab_test_treatment_data = ab_test_data_no_out[ab_test_data_no_out["customer_segmnt"]
```



```
ab_test_control_data.shape
```

```
(9973, 5)
```

```
ab_test_treatment_data.shape
```

```
(9987, 5)
```

```
# 2-sample z-test as n1 nad n2 are large.
```

```
# Refer: https://www.statsmodels.org/dev/generated/statsmodels.stats.weightstats.zt
```

```
from statsmodels.stats.weightstats import ztest as ztest
```

```
ztest(ab_test_control_data["watch_time_hrs"], ab_test_treatment_data["watch_time_hr
```

```
(15.96034913022092, 2.4137738128170024e-57)
```

▼ T-Test

```
#T-Test
```

```
dof = ab_test_control_data.shape[0] + ab_test_treatment_data.shape[0] - 2
```

```
dof
```

```
19958
```

```
diff_means = ab_test_control_data["watch_time_hrs"].mean() - ab_test_treatment_data
```

```
diff_means
```

```
0.555666548844524
```

```
#2 sample t-test
```

```
stats.ttest_ind(ab_test_control_data["watch_time_hrs"], ab_test_treatment_data["wat
```

```
Ttest_indResult(statistic=15.96034913022092, pvalue=5.438408586231319e-57)
```

 0s completed at 15:30