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)
```

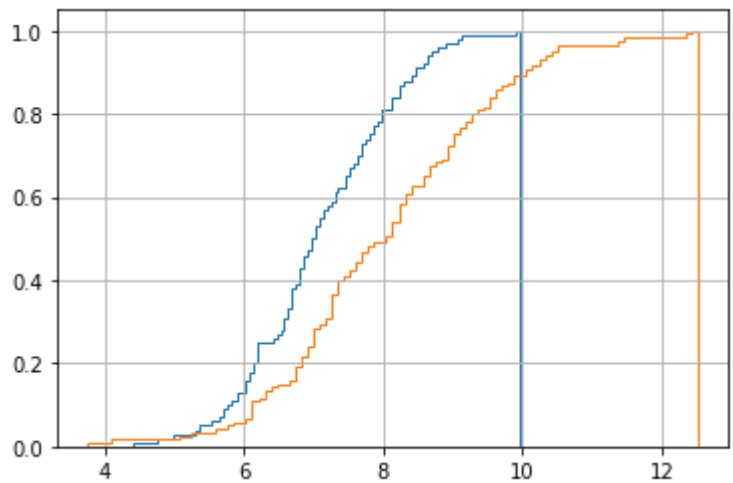
```
    KstestResult(statistic=0.3233333333333333, pvalue=1.5163387982131127e-05)
```

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)

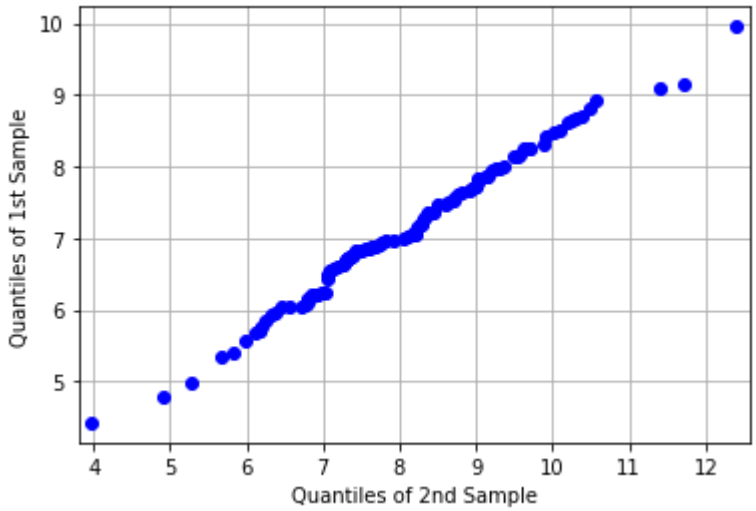
```
plt.grid()
a = plt.hist(d1, bins=100, cumulative=True, label='CDF', density=True, histtype='st
b = plt.hist(d2, bins=100, cumulative=True, label='CDF', density=True, histtype='st
plt.show()
```




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

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: Futur
  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 comapre 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/1Hl96n6BWdl3ruJgCo_gaAWEb0kEYg__H/view?usp=sharin
id = "1Hl96n6BWdl3ruJgCo_gaAWEb0kEYg__H"
path = "https://docs.google.com/uc?export=download&id=" + id
print(path)
```

```
https://docs.google.com/uc?export=download&id=1Hl96n6BWdl3ruJgCo_gaAWEb0kEYg__
```

```
!wget "https://docs.google.com/uc?export=download&id=1Hl96n6BWdl3ruJgCo_gaAWEb0kEYg
```

```
--2022-07-15 09:58:22--  https://docs.google.com/uc?export=download&id=1Hl96n6
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.googleuserconte
Connecting to doc-00-ag-docs.googleusercontent.com (doc-00-ag-docs.googleuserc
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
```

```
Streaming output truncated to the last 5000 lines.
2018-12-11,282,0,7.605139253709626,control
2018-09-19,42,0,2.935637149241187,control
2018-12-03,186,0,1.4865915060433719,control
2018-03-07,858,0,3.077709767971149,treatment
2018-03-12,961,0,5.477459458663748,treatment
2018-08-05,795,1,0.692514874129173,treatment
2018-05-16,894,0,5.120439877884421,treatment
2018-12-17,235,1,3.359689350528685,control
2018-01-16,1,0,2.481073013731408,control
2018-06-16,673,0,2.204182726873531,treatment
2018-03-13,974,0,1.0660611626400247,treatment
2018-08-03,358,1,1.159103405278024,control
2018-08-21,617,0,1.3828799214591214,treatment
2018-03-06,464,0,1.2201656323762258,control
2018-01-07,950,0,1.541418132845083,treatment
2018-02-22,386,0,0.7704302714023268,control
2018-05-25,822,0,2.2250176586222103,treatment
2018-01-14,166,0,1.7474447050896362,control
2018-01-19,159,1,3.9461830039276027,control
2018-11-22,320,0,1.5204168949544836,control
2018-02-08,915,1,1.9463796819077597,treatment
2018-01-11,812,0,5.811112715817566,treatment
2018-01-30,655,1,2.285621781613707,treatment
2018-10-01,325,0,3.4950416745360693,control
2018-11-13,405,0,2.2415860475504177,control
2018-07-05,286,0,3.376674573461322,control
2018-04-27,958,0,3.8910152531511457,treatment
2018-05-10,877,1,3.5523509536439937,treatment
2018-05-08,487,0,1.6956352695921468,control
2018-01-21,579,0,1.7983680286387858,treatment
```

```
2018-12-31,375,1,0.5658757425115898,control
2018-10-02,454,0,1.4223777430046,control
2018-04-01,571,0,1.701038496967191,treatment
2018-09-29,291,0,1.7466780774364734,control
2018-11-21,911,0,5.120313643829219,treatment
2018-06-27,122,0,3.661661824701089,control
2018-08-18,416,0,2.167692989673024,control
2018-03-26,62,0,3.1335231808338233,control
2018-01-23,377,0,1.1708621456693034,control
2018-02-25,847,1,4.835258977150139,treatment
2018-01-18,107,0,4.044797637817295,control
2018-06-08,895,1,0.3875093720676122,treatment
2018-05-26,261,0,4.62352375849384,control
2018-11-28,108,0,2.6113191939037854,control
2018-07-25,346,1,4.286072642575083,control
2018-07-12,464,0,4.934722866754722,control
2018-01-08,545,0,2.8932071720309405,treatment
2018-04-01,584,0,2.3654486560181707,treatment
2018-04-08,755,0,12.924859105271475,treatment
2018-07-07,597,0,4.181364412920746,treatment
2018-02-22,403,0,2.0178176840734063,control
2018-12-21,665,0,1.7074820186103967,treatment
2018-11-25,467,1,0.9615658998163824,control
2018-09-07,86,0,1.103975609252748,control
2018-12-24,462,0,1.6265017901627483,control
2018-12-05,977,0,3.1812231926737753,treatment
2018-05-04,360,0,3.350953607160751,control
```

```
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)
```