KS-Test

[ ] ↳ *11 cells hidden*

T-test & 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
```

```
--2022-05-19 15:48:44--  https://docs.google.com/uc?export=download&id=1Hl96n6
Resolving docs.google.com (docs.google.com)... 173.194.218.139, 173.194.218.10
Connecting to docs.google.com (docs.google.com)|173.194.218.139|:443... connec
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.
--2022-05-19 15:48:45--  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

2022-05-19 15:48:45 (122 MB/s) - 'ab_test_data.csv' saved [887610/887610]
```

```
!ls -lrt
```

```
total 872
drwxr-xr-x 1 root root   4096 May 17 13:39 sample_data
-rw-r--r-- 1 root root 887610 May 19 12:07 ab_test_data.csv
```

```
!cat ab_test_data.csv
```

```
2018-07-26,499,0,0.9485966082564069,control
2018-05-18,164,0,1.8795977915646052,control
2018-05-07,844,0,4.164459695951402,treatment
2018-07-28,30,0,2.597267595449251,control
2018-10-22,51,0,3.0921452227127206,control
2018-06-20,532,0,2.250200897332297,treatment
2018-04-27,555,0,0.8846015100833784,treatment
2018-07-28,201,0,1.3060720892263125,control
2018-07-30,503,1,5.268129198848328,treatment
2018-07-31,752,0,2.1013009965234732,treatment
2018-07-21,111,0,1.795113710002761,control
2018-05-17,768,0,2.726962366461604,treatment
2018-08-12,416,0,3.80262145422116,control
2018-08-11,139,1,1.2131887871749831,control
2018-03-27,308,0,2.1737724004449883,control
2018-06-17,970,0,5.820765858258524,treatment
2018-09-05,376,0,5.066769890914933,control
2018-09-07,489,0,1.3846492630418181,control
2018-12-29,308,0,10.795949274699835,control
2018-05-30,736,0,3.7897019408791426,treatment
2018-03-06,301,0,2.699121326947254,control
2018-08-03,547,1,6.200809700041885,treatment
2018-07-21,436,1,1.3518607837222931,control
2018-11-14,507,0,4.3198214955554874,treatment
2018-07-11,807,1,1.1820394825584641,treatment
2018-08-07,771,0,1.6956154028825516,treatment
2018-07-21,704,0,1.1660962947526128,treatment
2018-05-10,620,1,1.1703272186648386,treatment
2018-04-30,681,1,2.2900538409710434,treatment
2018-01-17,761,0,1.7702776878978959,treatment
2018-12-19,951,0,2.66771754179069,treatment
2018-08-01,988,0,3.1589984606381125,treatment
2018-05-11,86,1,2.1666533893975766,control
2018-04-18,752,0,1.856403982596956,treatment
2018-06-08,301,0,0.6557341058208296,control
2018-08-30,686,0,2.995242572147309,treatment
2018-09-13,770,1,0.8376611814111453,treatment
2018-05-05,819,0,2.7851217721216868,treatment
2018-10-30,93,0,5.236038247515851,control
2018-07-05,716,0,3.0702712416461013,treatment
2018-07-04,711,0,2.7671672868091024,treatment
2018-11-10,409,0,3.663753664506593,control
2018-12-12,626,0,3.4690048360831014,treatment
2018-04-21,303,0,2.497720726919897,control
2018-10-01,610,0,12.71433207126069,treatment
2018-08-18,767,0,4.3734687765449936,treatment
2018-05-21,46,0,0.8180819239118137,control
2018-03-31,142,0,2.976420578309185,control
2018-06-26,493,0,1.9497968104617955,control
```

```
2018-05-23,664,0,2.2007621539261053,treatment
2018-12-12,385,0,2.2548042204823537,control
2018-07-10,851,1,5.263484969696276,treatment
2018-03-14,165,0,1.0715842439019196,control
2018-12-01,335,0,3.5389298305442685,control
2018-06-21,892,0,2.5789716946294967,treatment
2018-11-23,885,0,4.262445013483545,treatment
2018-09-15,603,0,2.808984233326774,treatment
2018-06-12,368,0,1.8120152663344664,control
2018-01-22,963,0,5.083732249974792,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
3269	2018-08-03	438	1	5.759168	control
6562	2018-03-30	275	0	2.534567	control
9975	2018-01-06	381	0	2.232723	control
1042	2018-12-02	621	1	3.279372	treatment
1860	2018-09-07	505	0	7.124559	treatment
...	...	...	...	...	...
4403	2018-05-18	644	0	3.742603	treatment
355	2018-01-09	915	0	2.659615	treatment
16787	2018-11-04	695	0	0.943913	treatment
16841	2018-02-02	475	0	2.246929	control
7323	2018-04-25	976	1	4.504133	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

```
# 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
```

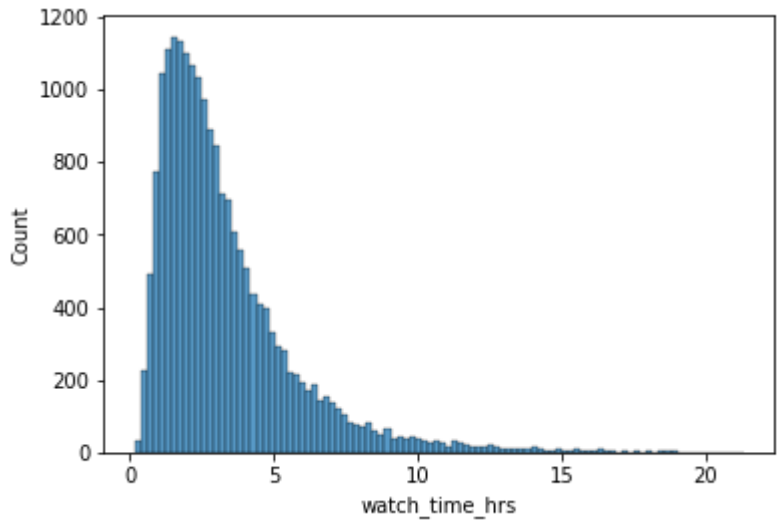
```
26.036198684124518
```

```
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)
```

```
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
```

```
0.5556665488445294
```

```
#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)
```

```
# 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
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: Futur
  import pandas.util.testing as tm
(15.96034913022092, 2.4137738128170024e-57)
```

✓ 0s completed at 21:18