```
# Packages imports
import numpy as np
import pandas as pd
import scipy.stats as stats
import statsmodels.stats.api as sms
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
from math import ceil

%matplotlib inline


import warnings
warnings.filterwarnings("ignore")
warnings.simplefilter(action='ignore', category=FutureWarning)

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.options.display.float_format = '{:.4f}'.format


df = pd.read_excel('assignment.xlsx', sheet_name='AB_Test')


df.head()
```

	Date	Address	Variations	DeviceType	Clicks	Visitors
0	2023-04-09	fyttlyf/business/campaign/campaigns-business/c...	Control	Desktop	146	1189
1	2023-04-09	fyttlyf/business/campaign/get-paid///	Treatment	Mobile	19	1389
2	2023-04-09	fyttlyf/business/campaign/campaigns-business/s...	Control	Desktop	134	5752
	2023-	fyttlyf/business/product/business-				

```
df.shape
```

```
(18099, 6)
```

```
df['Variations'].value_counts()
```

```
Control      10547
Treatment     7552
Name: Variations, dtype: int64
```

```
df['Address'].value_counts()
```

```
fyttlyf/business/product/mitigate-risk///                                  44
fyttlyf/enterprise/product/enterprise-manage-risk///                       44
fyttlyf/business/product/business-marketplace-payments///                  43
fyttlyf/business/product/business-raise-support-for-business///            43
fyttlyf/personal/campaign/signup10///                                      39
fyttlyf/enterprise//enterprise-advanced-fraud-security///                  38
fyttlyf/business/enterprise/digital-gambling-campaign///                   34
fyttlyf/business//travel-business-payment-solutions///                     34
fyttlyf/enterprise/product/enterprise-product-spotlight///                 33
fyttlyf/general//account-sign-up///                                        30
fyttlyf/personal//home///                                                  28
fyttlyf/personal//signup-via-app-terms///                                  28
main/consappdownload/xsell////APP_REDIRECT/                                27
fyttlyf/enterprise/campaigns/global-payment-processing-solutions///        25
fyttlyf/business/product/all-in-one-solution///                            24
fyttlyf/personal/campaign/qip2r059n///                                     17
fyttlyf/personal/campaign/kmspr057t///                                     17
fyttlyf/personal/homepage/home///                                          17
fyttlyf/personal/campaign/udspr120w///                                     16
fyttlyf/personal/campaign/opt-in-eng///                                    15
fyttlyf///fyttlyf-and-your-data///                                         14
fyttlyf/personal/campaign/epp2r07sf///                                     12
fyttlyf/personal/campaign/eyspr15mj///                                     11
fyttlyf/personal/campaign/euspr07hl///                                     10
fyttlyf/business//digital-grocery-payment-solutions-report///               9
fyttlyf//////                                                               8
fyttlyf/business/campaign/accept-more-payments///                           7
fyttlyf/personal//country-worldwide///                                      6
fyttlyf/business/campaign/campaigns-business/integration-update///          5
fyttlyf/both/product/account-selection///                                   3
fyttlyf/personal/buy/security-safety///                                     2
fyttlyf/business//solution-providers///                                     2
fyttlyf/personal/product/digital-wallet-ways-to-pay/add-payment-method///   2
fyttlyf/personal/product/digital-wallet-send-receive-money/fyttlyf-me///    1
fyttlyf/personal/product/digital-wallet-ways-to-pay/buy-now-pay-later///    1
Name: Address, dtype: int64
```

```
df['DeviceType'].value_counts()
```

```
Desktop    6330
Mobile     6181
Tablet     3687
Others     1901
Name: DeviceType, dtype: int64
```

```
df['Date'].value_counts()
```

```
2023-03-31    1100
2023-04-03    1046
2023-04-09    1035
2023-03-30    1029
2023-03-29    1025
2023-04-04    1025
2023-04-14    1012
2023-04-01    1010
2023-04-11    1010
2023-04-05     989
2023-04-07     988
2023-04-02     986
2023-04-12     984
2023-04-10     983
2023-04-06     978
2023-04-08     978
2023-04-13     970
2023-04-15     951
Name: Date, dtype: int64
```

```
df['Clicks'].value_counts(ascending=False)
```

```
478         1
590         1
857         1
157         1
622         1
3497        1
1174        1
2173        1
1160        1
238         1
322         1
5984        1
333         1
673         1
2206        1
549         1
2326        1
579         1
3614        1
1218        1
357         1
876         1
754         1
566         1
5724        1
985         1
834         1
1107        1
391         1
2928        1
335         1
733         1
3438        1
474         1
1346        1
403         1
883         1
485         1
481         1
966         1
1210        1
1775        1
1784        1
1137        1
288         1
369         1
572         1
2907        1
554         1
Name: Clicks, dtype: int64
```

```
df['Clicks'].nunique()
```

```
722
```

```
df['Clicks'].unique()
```

```
  327,   322,   248,   238,  1160,   151,  2173,  1174,  3497,
  331,   245,   622,   157,   857,   566,   590,   324,  5724,
  834,  2907,   312,   103,   227,   175,   572,   369,   288,
 1137,  1784,  1775,   159,  1210,   125,   966,   481,   485,
  883,   511,   403,  1346,   474,   304,   522,   217,   213,
 3438,   733,   152,  1126,   651,   335,  2928,   273,   391,
 1107,   985,  1163,   504,   471,   218,   695,  1158,  2401,
 1215,   280,   548,   291,   756,   544,   697,  1070,   143,
10680,   347,   221,   191,  6525,  2613,   436,  1226,  1220,
  987,   210,  1214,   448,  6384,   741,  2170,   465,   942,
  491,   153,  1209,   685,  2478,  1014,   379,  3617,   196,
 1212,   380,   150,   543,  1006,   386,  3581,  1189,  8994,
  898,   270,  1023,   405,   702,   140,  3925,   627,  1076,
  219,   694,  9987,  2271,   204,  1185,   469,   666,   626,
  496,  1094,   456,  6576,   211,  2266,   796,  3088,  1243,
  761,   259,  1266,  4112,   819,  2057,   409,   444,   198,
  422,   581,  1524,  1247,  3076,   254,   441,   726,   362,
  250,   182,   537,  5414,  1180,   795,   309,   212,  1052,
  401,  2162,   567,   224,   429,   918,   500,  1149,   299,
 2077,   569,  3782,   289,   346,  2830,  1527,  2523,   410,
 5468,   364,   681,  1263,  7094,   404,   775,   574,  2698,
  236,   633,   301,   484,  2248,   650,   428,   426,   329,
  653,  1154,   222, 11190,   679,   470,   297,  1110,   294,
 3993,  6212,   656,  1011,  3606,   138,  1248,  6434,  1127,
  641,   455,   338,  1120,  2177,  1159,   593,   753,   925,
  607,   730,   505,   411,   305, 10475,  1148,   565,  6261,
 2276,   439,   269,   296,  4897,  3910,   216,  6122,   306,
  763, 10805,  2329,  6501,  3938,  1086,  1101,   542,  1176,
  460,   744,   509,  2425,   794,  5123,   344,  1124,   226,
  573,   698,  1105,  6471,  1175,   317,   412,  2311,   731,
 2190,  1201,   534,   461,   892,   510,   283,  3676,   476,
 4889,   689,   494,   599,   718,   517,   669,  5739,   676,
  601,   360,   423,   586,  3274,  6070,  5062,   268,  1235,
  521,   453,   714,  4085,   201,   503,   251,  1061,  1092,
 1975,   849,   447,   552,  2194,  2350,   424,   383,   553,
  406,   644,   154,  1870,   413,  3099,   865,  4402,  2745,
 1339,   554])
```

```
df["Clicks"] = np.where(df["Clicks"] >1000, 0, 1)
```

```
df['Visitors'].value_counts()
```

```
6637         1
1375         1
5846         1
5266         1
1479         1
335          1
3644         1
7008         1
2735         1
2387         1
2034         1
6707         1
893          1
2932         1
47639        1
1556         1
2098         1
1456         1
5676         1
4678         1
960          1
Name: Visitors, dtype: int64
```

```
df['Visitors'].nunique()
```

```
1705
```

```
df['Visitors'].unique().tolist()
```

```
        664,
        440,
        254,
        2268,
        11274,
        298,
        5609,
        ...]
```

```
df["Visitors"] = np.where(df["Visitors"] >5000, 0, 1)
```

```
df["Visitors"]
```

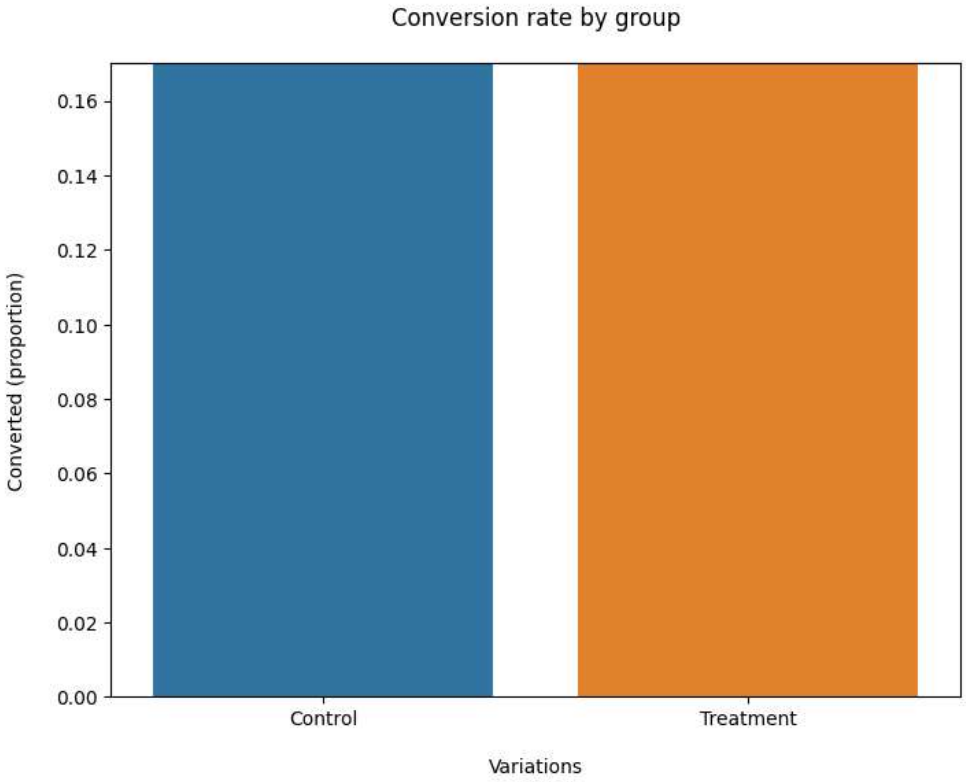
```
18042    1
18043    1
18044    1
18045    1
18046    1
18047    1
18048    1
18049    1
18050    1
18051    1
18052    1
18053    1
18054    1
18055    1
18056    1
18057    1
18058    1
18059    1
18060    1
18061    1
18062    1
18063    1
18064    0
18065    1
18066    1
18067    1
18068    1
18069    1
18070    1
18071    1
18072    1
18073    1
18074    1
18075    1
18076    1
18077    1
18078    1
18079    1
18080    1
18081    1
18082    1
18083    1
18084    1
18085    1
18086    1
18087    1
18088    1
18089    1
18090    1
18091    1
18092    1
18093    1
18094    1
18095    1
18096    1
18097    1
18098    1
Name: Visitors, dtype: int64
```

```
conversion_rates = df.groupby('Variations')['Visitors']

std_p = lambda x: np.std(x, ddof=0)              # Std. deviation of the proportion
se_p = lambda x: stats.sem(x, ddof=0)            # Std. error of the proportion (std / sqrt(n))
```

```
conversion_rates = conversion_rates.agg([np.mean, std_p, se_p])
conversion_rates.columns = ['conversion_rate', 'std_deviation', 'std_error']


conversion_rates.style.format('{:.3f}')
```

Variations	conversion_rate	std_deviation	std_error
Control	0.976	0.152	0.001
Treatment	0.997	0.055	0.001

```
plt.figure(figsize=(8,6))

sns.barplot(x=df['Variations'], y=df['Visitors'], ci=False)

plt.ylim(0, 0.17)
plt.title('Conversion rate by group', pad=20)
plt.xlabel('Variations', labelpad=15)
plt.ylabel('Converted (proportion)', labelpad=15);
```




```
from statsmodels.stats.proportion import proportions_ztest, proportion_confint


control_results = df[df['Variations'] == 'Control']['Visitors']
treatment_results = df[df['Variations'] == 'Treatment']['Visitors']


n_con = control_results.count()
n_treat = treatment_results.count()
successes = [control_results.sum(), treatment_results.sum()]
nobs = [n_con, n_treat]

z_stat, pval = proportions_ztest(successes, nobs=nobs)
(lower_con, lower_treat), (upper_con, upper_treat) = proportion_confint(successes, nobs=nobs, alpha=0.05)

print(f'z statistic: {z_stat:.2f}')
print(f'p-value: {pval:.3f}')
```

```
print(f'ci 95% for control group: [{lower_con:.3f}, {upper_con:.3f}]')
print(f'ci 95% for treatment group: [{lower_treat:.3f}, {upper_treat:.3f}]')
```

```
z statistic: -11.27
p-value: 0.000
ci 95% for control group: [0.973, 0.979]
ci 95% for treatment group: [0.996, 0.998]
```

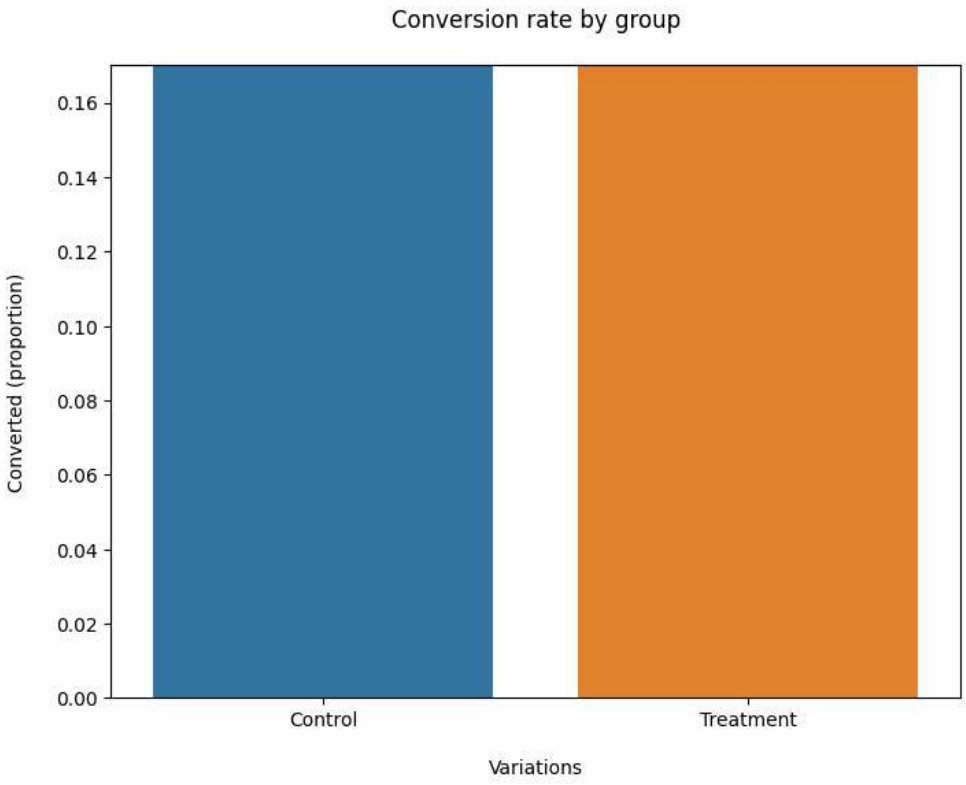
```
conversion_rates = df.groupby('Variations')['Clicks']

std_p = lambda x: np.std(x, ddof=0)              # Std. deviation of the proportion
se_p = lambda x: stats.sem(x, ddof=0)            # Std. error of the proportion (std / sqrt(n))

conversion_rates = conversion_rates.agg([np.mean, std_p, se_p])
conversion_rates.columns = ['conversion_rate', 'std_deviation', 'std_error']
```

```
conversion_rates.style.format('{:.3f}')
```

Variations	conversion_rate	std_deviation	std_error
Control	0.986	0.116	0.001
Treatment	0.995	0.072	0.001

```
plt.figure(figsize=(8,6))

sns.barplot(x=df['Variations'], y=df['Clicks'], ci=False)

plt.ylim(0, 0.17)
plt.title('Conversion rate by group', pad=20)
plt.xlabel('Variations', labelpad=15)
plt.ylabel('Converted (proportion)', labelpad=15)
```

```
Text(0, 0.5, 'Converted (proportion)')
```




```
from statsmodels.stats.proportion import proportions_ztest, proportion_confint

control_results = df[df['Variations'] == 'Control']['Clicks']
```

```
treatment_results = df[df['Variations'] == 'Treatment']['Clicks']

n_con = control_results.count()
n_treat = treatment_results.count()
successes = [control_results.sum(), treatment_results.sum()]
nobs = [n_con, n_treat]

z_stat, pval = proportions_ztest(successes, nobs=nobs)
(lower_con, lower_treat), (upper_con, upper_treat) = proportion_confint(successes, nobs=nobs, alpha=0.05)

print(f'z statistic: {z_stat:.2f}')
print(f'p-value: {pval:.3f}')
print(f'ci 95% for control group: [{lower_con:.3f}, {upper_con:.3f}]')
print(f'ci 95% for treatment group: [{lower_treat:.3f}, {upper_treat:.3f}]')
```

```
z statistic: -5.58
p-value: 0.000
ci 95% for control group: [0.984, 0.989]
ci 95% for treatment group: [0.993, 0.996]
```

✓ 0s completed at 10:22 AM