```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
import warnings
warnings.filterwarnings('ignore')

control_df =
pd.read_csv('/kaggle/input/ab-testing-dataset/control_group.csv', sep
=";")
test_df =
pd.read_csv('/kaggle/input/ab-testing-dataset/test_group.csv', sep
=";")

control_df.head()
```

```
      Campaign Name       Date  Spend [USD]  # of Impressions
Reach  \
0  Control Campaign  1.08.2019         2280           82702.0
56930.0
1  Control Campaign  2.08.2019         1757          121040.0
102513.0
2  Control Campaign  3.08.2019         2343          131711.0
110862.0
3  Control Campaign  4.08.2019         1940           72878.0
61235.0
4  Control Campaign  5.08.2019         1835               NaN
NaN

   # of Website Clicks  # of Searches  # of View Content  # of Add to
Cart  \
0               7016.0         2290.0             2159.0
1819.0
1               8110.0         2033.0             1841.0
1219.0
2               6508.0         1737.0             1549.0
1134.0
3               3065.0         1042.0              982.0
1183.0
4                  NaN            NaN                NaN
NaN

   # of Purchase
0          618.0
1          511.0
2          372.0
3          340.0
4            NaN
```

```
test_df.head()
```

```
   Campaign Name       Date  Spend [USD]  # of Impressions  Reach  \
0  Test Campaign  1.08.2019         3008             39550  35820
1  Test Campaign  2.08.2019         2542            100719  91236
2  Test Campaign  3.08.2019         2365             70263  45198
3  Test Campaign  4.08.2019         2710             78451  25937
4  Test Campaign  5.08.2019         2297            114295  95138

   # of Website Clicks  # of Searches  # of View Content  # of Add to
Cart  \
0                 3038           1946               1069
894
1                 4657           2359               1548
879
2                 7885           2572               2367
1268
3                 4216           2216               1437
566
4                 5863           2106                858
956

   # of Purchase
0            255
1            677
2            578
3            340
4            768
```

```
control_df.columns = ["Campaign Name", "Date", "Amount Spent",
                      "Number of Impressions", "Reach", "Website
Clicks",
                      "Searches Received", "Content Viewed", "Added
to Cart",
                      "Purchases"]

test_df.columns = ["Campaign Name", "Date", "Amount Spent",
                      "Number of Impressions", "Reach", "Website
Clicks",
                      "Searches Received", "Content Viewed", "Added
to Cart",
                      "Purchases"]

control_df.describe()
```

```
       Amount Spent  Number of Impressions          Reach  Website
Clicks  \
count     30.000000              29.000000      29.000000
29.000000
mean    2288.433333          109559.758621   88844.931034
```

```
5320.793103
std      367.334451           21688.922908   21832.349595
1757.369003
min     1757.000000           71274.000000   42859.000000
2277.000000
25%     1945.500000           92029.000000   74192.000000
4085.000000
50%     2299.500000          113430.000000   91579.000000
5224.000000
75%     2532.000000          121332.000000  102479.000000
6628.000000
max     3083.000000          145248.000000  127852.000000
8137.000000

       Searches Received  Content Viewed  Added to Cart   Purchases
count          29.000000       29.000000      29.000000   29.000000
mean         2221.310345     1943.793103    1300.000000  522.793103
std           866.089368      777.545469     407.457973  185.028642
min          1001.000000      848.000000     442.000000  222.000000
25%          1615.000000     1249.000000     930.000000  372.000000
50%          2390.000000     1984.000000    1339.000000  501.000000
75%          2711.000000     2421.000000    1641.000000  670.000000
max          4891.000000     4219.000000    1913.000000  800.000000

test_df.describe()

       Amount Spent  Number of Impressions          Reach  Website
Clicks  \
count     30.000000              30.000000      30.000000
30.000000
mean    2563.066667           74584.800000   53491.566667
6032.333333
std      348.687681           32121.377422   28795.775752
1708.567263
min     1968.000000           22521.000000   10598.000000
3038.000000
25%     2324.500000           47541.250000   31516.250000
4407.000000
50%     2584.000000           68853.500000   44219.500000
6242.500000
75%     2836.250000           99500.000000   78778.750000
7604.750000
max     3112.000000          133771.000000  109834.000000
8264.000000

       Searches Received  Content Viewed  Added to Cart   Purchases
count          30.000000       30.000000      30.000000   30.000000
mean         2418.966667     1858.000000     881.533333  521.233333
std           388.742312      597.654669     347.584248  211.047745
min          1854.000000      858.000000     278.000000  238.000000
```

```
25%          2043.000000     1320.000000     582.500000  298.000000
50%          2395.500000     1881.000000     974.000000  500.000000
75%          2801.250000     2412.000000    1148.500000  701.000000
max          2978.000000     2801.000000    1391.000000  890.000000
```

```
control_df.isnull().sum()
```

```
Campaign Name             0
Date                      0
Amount Spent              0
Number of Impressions     1
Reach                     1
Website Clicks            1
Searches Received         1
Content Viewed            1
Added to Cart             1
Purchases                 1
dtype: int64
```

```
control_df.iloc[:, 3:] = control_df.iloc[:,
3:].fillna(control_df.iloc[:, 3:].median())
```

```
test_df.isnull().sum()
```

```
Campaign Name             0
Date                      0
Amount Spent              0
Number of Impressions     0
Reach                     0
Website Clicks            0
Searches Received         0
Content Viewed            0
Added to Cart             0
Purchases                 0
dtype: int64
```

```
control_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Campaign Name          30 non-null     object
 1   Date                   30 non-null     object
 2   Amount Spent           30 non-null     int64
 3   Number of Impressions  30 non-null     float64
 4   Reach                  30 non-null     float64
 5   Website Clicks         30 non-null     float64
 6   Searches Received      30 non-null     float64
 7   Content Viewed         30 non-null     float64
```

```
 8   Added to Cart          30 non-null     float64
 9   Purchases              30 non-null     float64
dtypes: float64(7), int64(1), object(2)
memory usage: 2.5+ KB
```

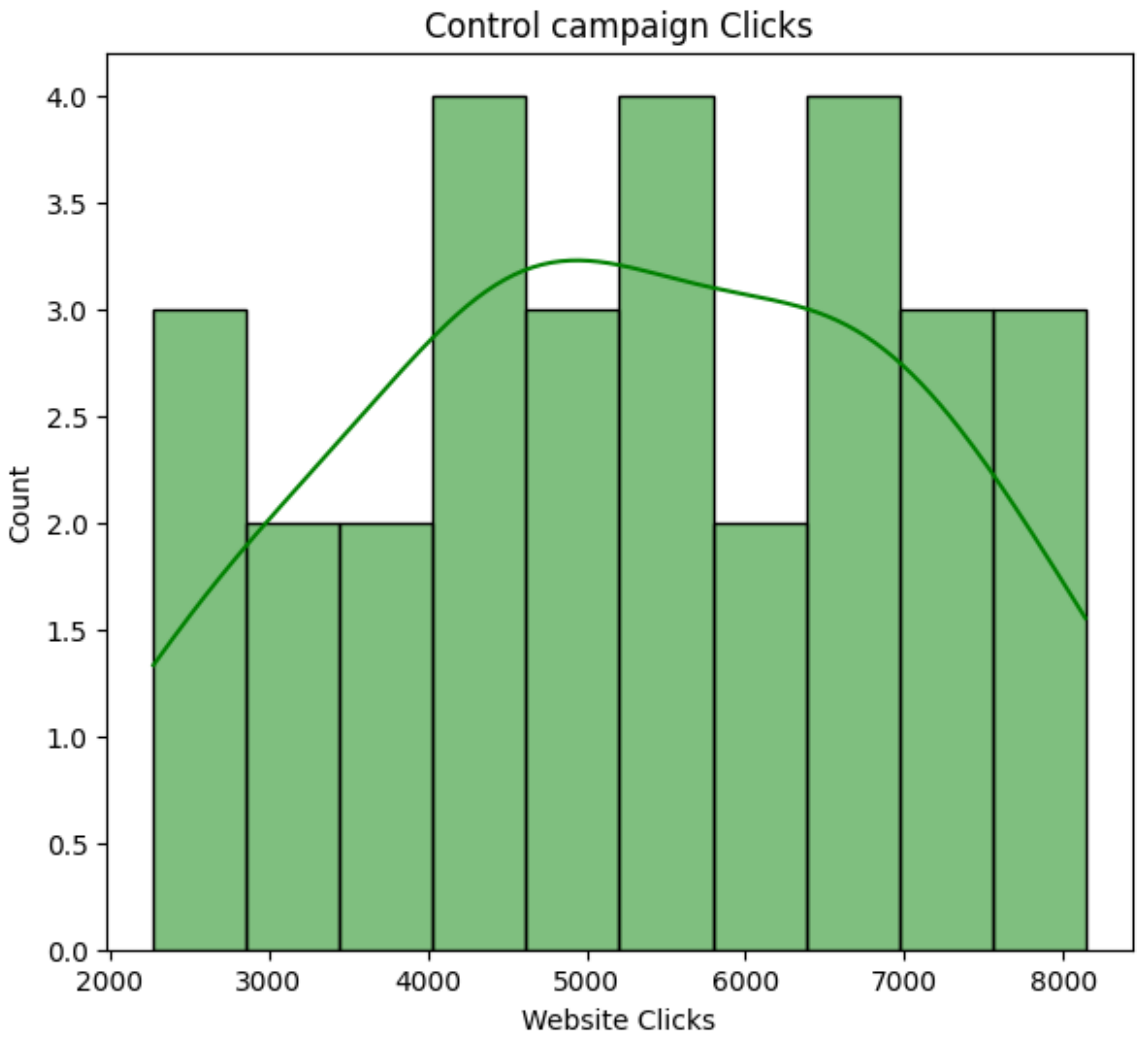
```
test_df.info()
```

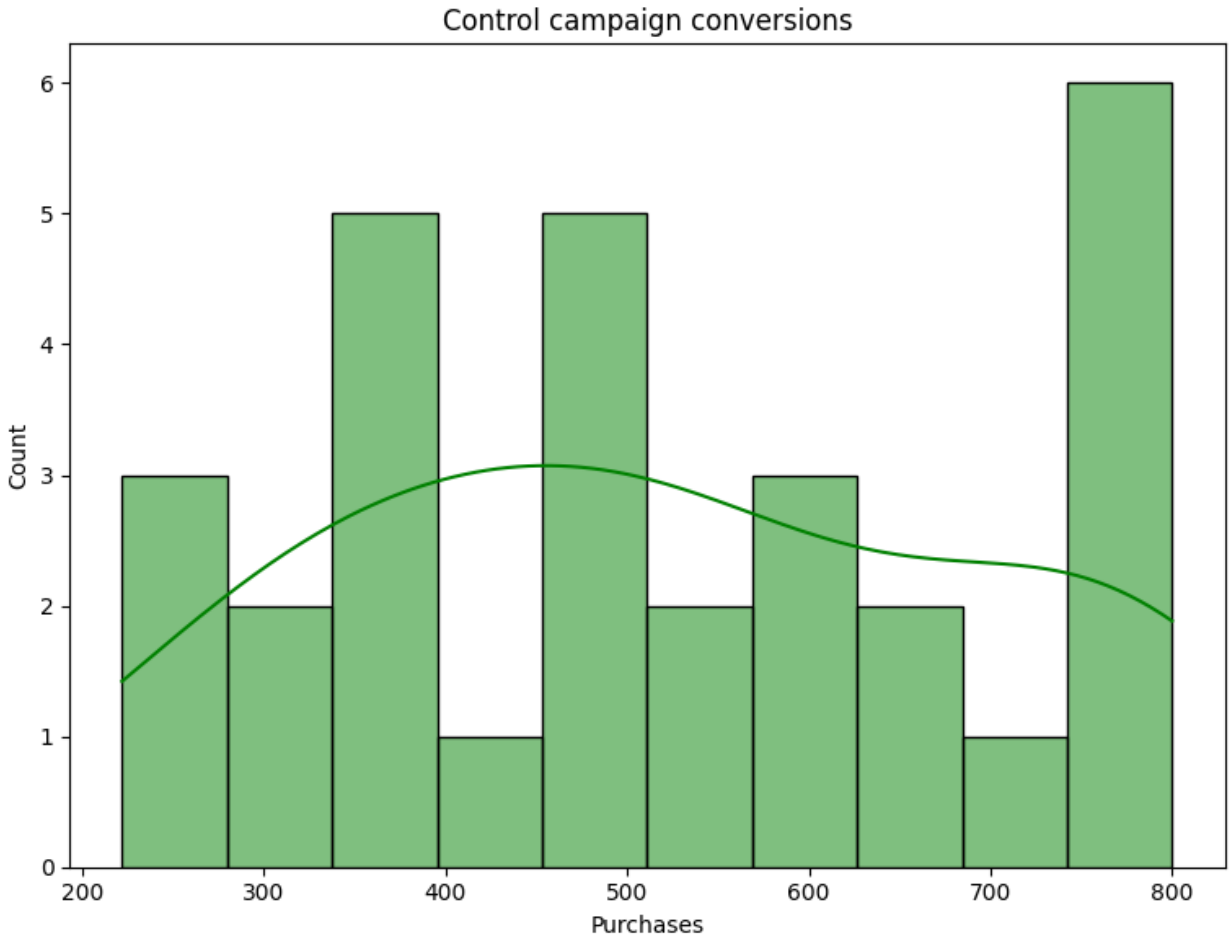
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Campaign Name          30 non-null     object
 1   Date                   30 non-null     object
 2   Amount Spent           30 non-null     int64
 3   Number of Impressions  30 non-null     int64
 4   Reach                  30 non-null     int64
 5   Website Clicks         30 non-null     int64
 6   Searches Received      30 non-null     int64
 7   Content Viewed         30 non-null     int64
 8   Added to Cart          30 non-null     int64
 9   Purchases              30 non-null     int64
dtypes: int64(8), object(2)
memory usage: 2.5+ KB
```

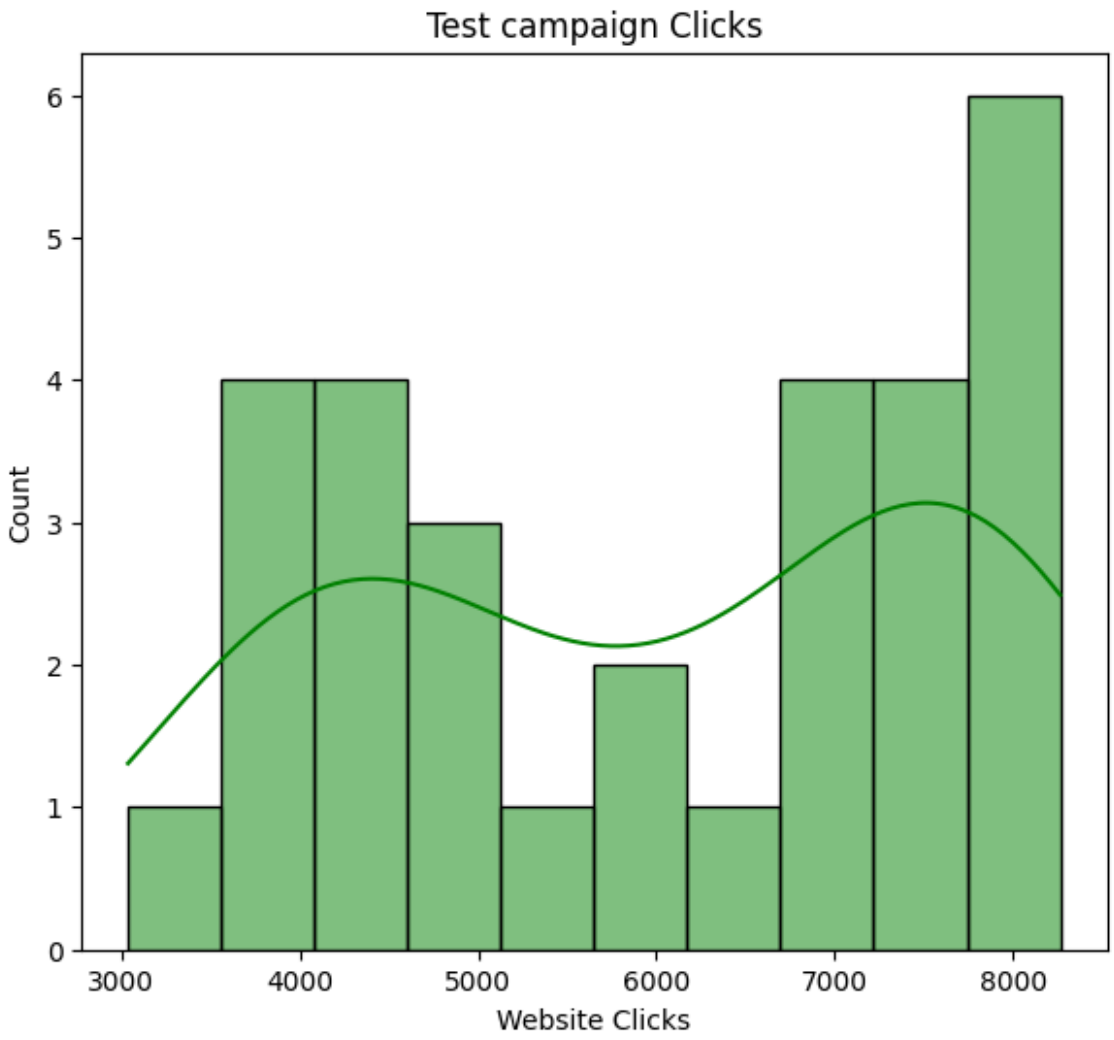
```
control_df['Date']=pd.to_datetime(control_df['Date'], format="%d.%m.
%Y")
test_df['Date']=pd.to_datetime(test_df['Date'], format="%d.%m.%Y")

columns_to_convert=["Amount Spent",
                    "Number of Impressions", "Reach", "Website
Clicks",
                    "Searches Received", "Content Viewed", "Added
to Cart",
                    "Purchases"]
control_df[columns_to_convert]=control_df[columns_to_convert].astype('
int64')

control_df['Click Through Rate'] = round(100*control_df['Website
Clicks'] / control_df['Number of Impressions'],2)
control_df['Conversion Rate'] =round(100* control_df['Purchases'] /
control_df['Website Clicks'],2)

test_df['Click Through Rate'] = round(100*test_df['Website Clicks'] /
test_df['Number of Impressions'],2)
test_df['Conversion Rate'] =round(100* test_df['Purchases'] /
test_df['Website Clicks'],2)
```

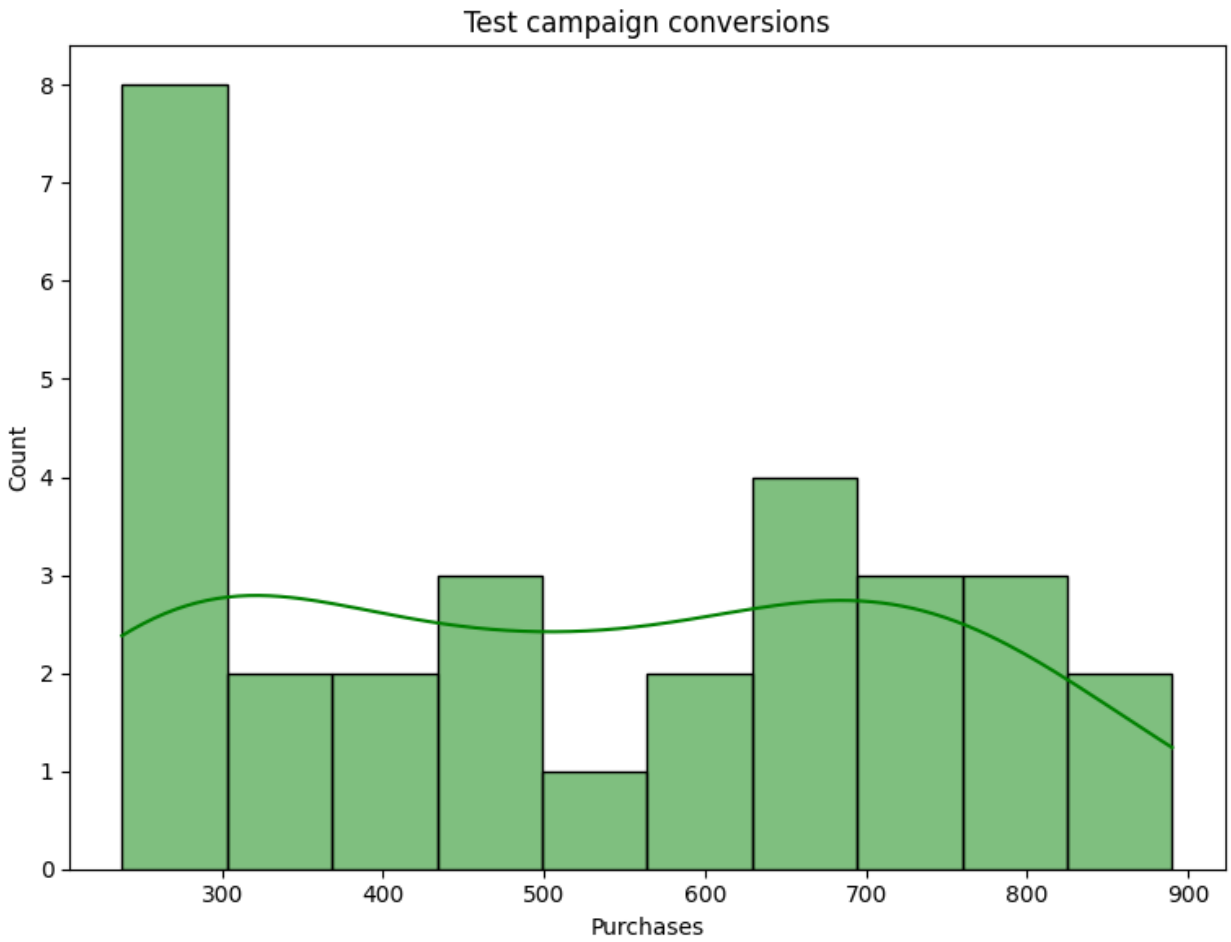
Which Campaign platform is more effective interms of clicks and conversions

```
#distribution of clicks ad purchases for Control Campaign
plt.figure(figsize=(15,6))
plt.subplot(1,2,1)
plt.title("Control campaign Clicks")
sns.histplot(control_df['Website
Clicks'],bins=10,edgecolor='k',color="green",kde=True)
plt.figure(figsize=(15,6))
plt.subplot(1,2,2)
plt.title("Control campaign conversions")
sns.histplot(control_df['Purchases'],bins=10,edgecolor='k',color="gree
n",kde=True)
plt.tight_layout()
plt.show()


#distribution of clicks ad purchases for Test campaign
plt.figure(figsize=(15,6))
plt.subplot(1,2,1)
plt.title("Test campaign Clicks")
sns.histplot(test_df['Website
Clicks'],bins=10,edgecolor='k',color="green",kde=True)
plt.figure(figsize=(15,6))
plt.subplot(1,2,2)
plt.title("Test campaign conversions")
sns.histplot(test_df['Purchases'],bins=10,edgecolor='k',color="green",
kde=True)
plt.tight_layout()
plt.show()
```


Control campaign Clicks
Count
Website Clicks
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
2000
3000
4000
5000
6000
7000
8000


Control campaign conversions
Count
Purchases
0
1
2
3
4
5
6
200
300
400
500
600
700
800


Test campaign Clicks
Count
Website Clicks
0
1
2
3
4
5
6
3000
4000
5000
6000
7000
8000



Does Campaign Control or Campaign Test achieve a higher click-through rate?

```
sum1= control_df['Click Through Rate']
sum2=test_df['Click Through Rate']
average_metrics1 = pd.DataFrame({
    'Metric': ['Control_df'],
    'Value': [sum1.sum()]
})

average_metrics2 = pd.DataFrame({
    'Metric': ['test_df'],
    'Value': [sum2.sum()]
})

combined_metrics = pd.concat([average_metrics1, average_metrics2],
ignore_index=True)
combined_metrics
```

```
       Metric   Value
0  Control_df  152.38
1     test_df  307.25
```

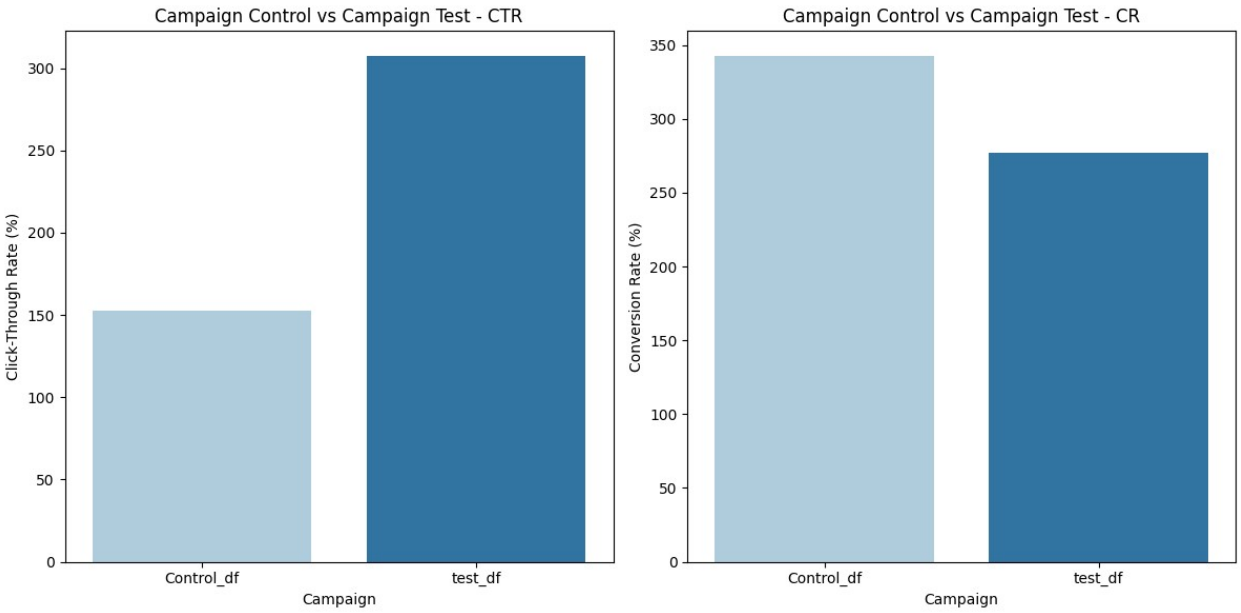
Does Campaign Control or Campaign Test achieve a higher conersion rate?

```
Total1= control_df['Conversion Rate']
Total2=test_df['Conversion Rate']
average_metrics1 = pd.DataFrame({
    'Metric': ['Control_df'],
    'Value': [Total1.sum()]
})

average_metrics2 = pd.DataFrame({
    'Metric': ['test_df'],
    'Value': [Total2.sum()]
})

combined_metrics2 = pd.concat([average_metrics1, average_metrics2],
ignore_index=True)
combined_metrics2
```

```
       Metric   Value
0  Control_df  342.44
1     test_df  276.92
```

Distribution of CTR and CR

```
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.barplot(x='Metric', y='Value', data=combined_metrics,
palette='Paired')
plt.title('Campaign Control vs Campaign Test - CTR')
plt.ylabel('Click-Through Rate (%)')
plt.xlabel('Campaign')


plt.subplot(1, 2, 2)
sns.barplot(x='Metric', y='Value', data=combined_metrics2,
palette='Paired')
plt.title('Campaign Control vs Campaign Test - CR')
plt.ylabel('Conversion Rate (%)')
plt.xlabel('Campaign')

plt.tight_layout()
plt.show()
```



The sum of CTRs of Campaign Test is greater than Campaign Control, it suggests that individual CTRs for Campaign Test are higher than those for Campaign Control.

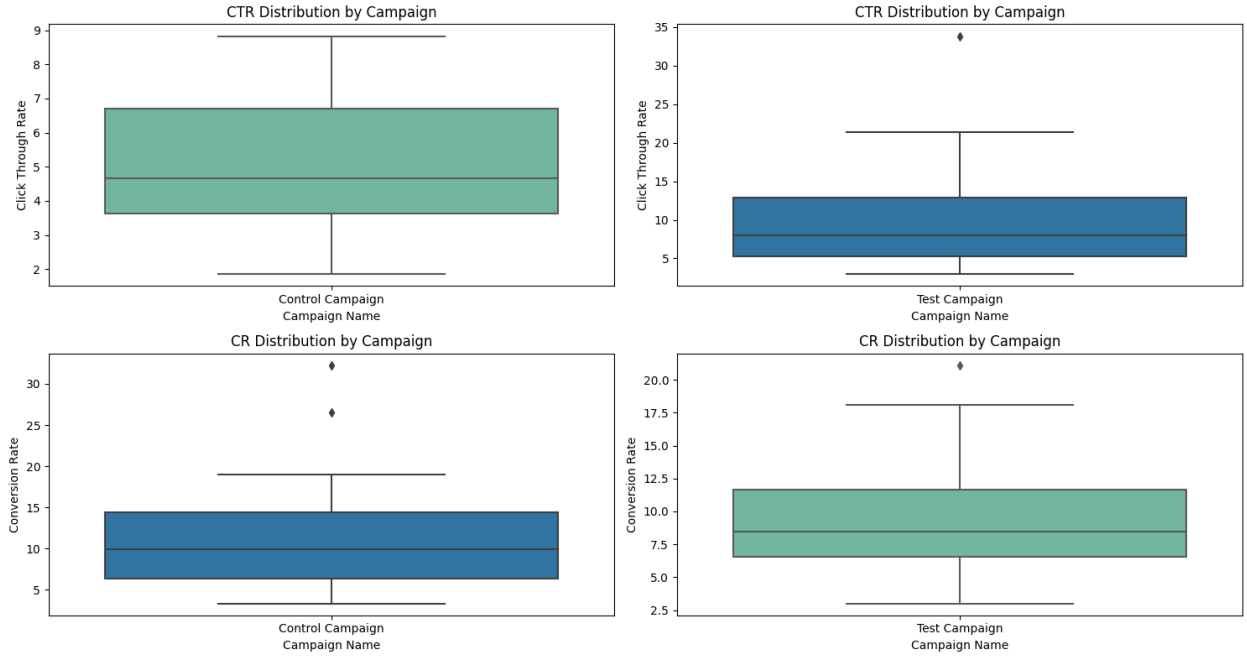
However, the sum of CR (Conversion Rate) for Campaign COntrol is higher, it indicates that, despite having a lower CTR, the users from Campaign Control are converting at a higher rate (i.e., completing desired actions like purchases).

High CTR but low CR: This may suggest that while Campaign Test attracts more users (due to high CTR), they are not converting as effectively.

Lower CTR but high CR: This suggests that Campaign Control might attract fewer users, but the users they attract are more likely to convert.

```
plt.figure(figsize=(15,8))
plt.subplot(2,2,1)
sns.boxplot(x='Campaign Name',y='Click Through
Rate',data=control_df,hue='Campaign Name',dodge=False,palette="Set2")
plt.title('CTR Distribution by Campaign')
plt.legend([], [], frameon=False)
plt.subplot(2,2,2)
sns.boxplot(x='Campaign Name',y='Click Through
Rate',data=test_df,hue='Campaign Name',dodge=False)
plt.title('CTR Distribution by Campaign')
plt.legend([], [], frameon=False)


plt.subplot(2,2,3)
sns.boxplot(x='Campaign Name',y='Conversion
Rate',data=control_df,hue='Campaign Name',dodge=False)
plt.title('CR Distribution by Campaign')
plt.legend([], [], frameon=False)
plt.subplot(2,2,4)
```

```
sns.boxplot(x='Campaign Name',y='Conversion
Rate',data=test_df,hue='Campaign Name',dodge=False,palette="Set2")
plt.title('CR Distribution by Campaign')
plt.legend([], [], frameon=False)
plt.tight_layout()
plt.show()
```



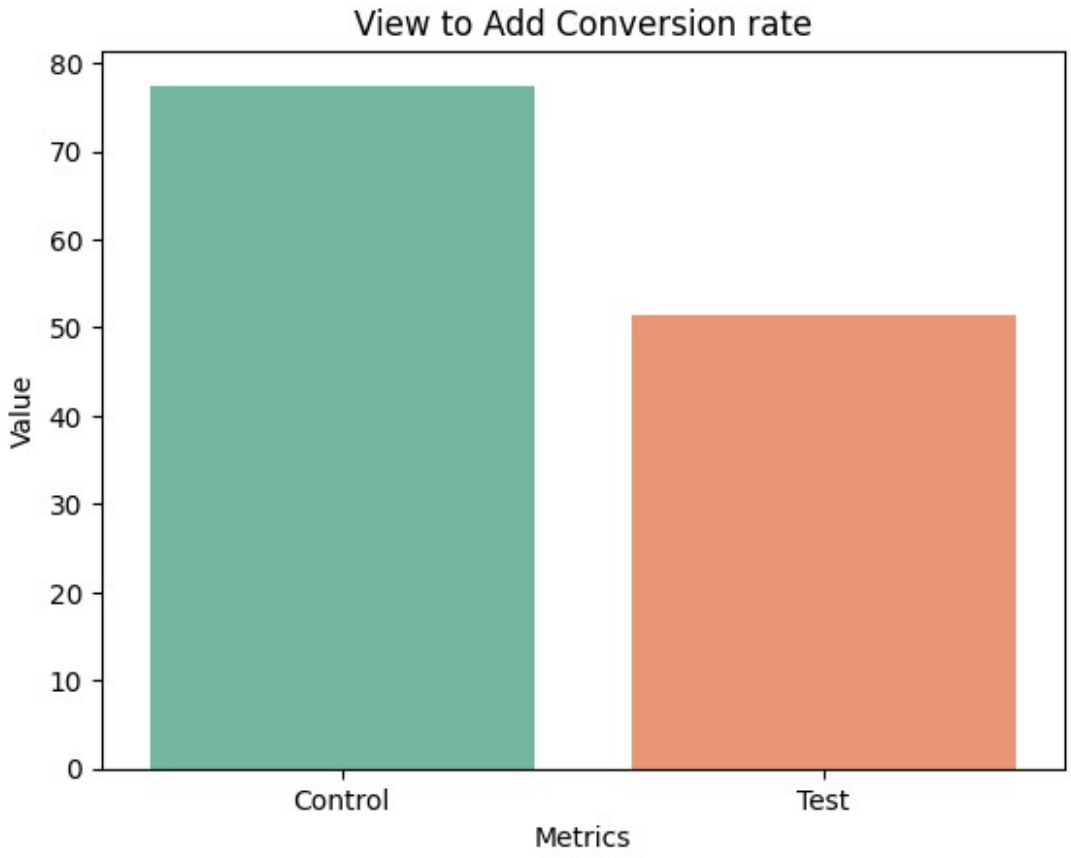

```
#View to Cart Conversion Rate- Control
VCR_C=round((100*control_df['Added to Cart']/control_df['Content
Viewed']).mean(),2)

#View to Cart Conversion Rate- Test
VCR_T=round((100*test_df['Added to Cart']/test_df['Content
Viewed']).mean(),2)

print(VCR_C,VCR_T)
```

```
77.44 51.51
```

```
df1= pd.DataFrame({'Metrics':["Control"],
                                'Value':[VCR_C]})
df2=pd.DataFrame({'Metrics':["Test"],
                                'Value':[VCR_T]})
combined_metrics2=pd.concat([df1,df2],ignore_index= True)
combined_metrics2
```

```
   Metrics  Value
0  Control  77.44
1     Test  51.51
```

```
sns.barplot(x='Metrics',y="Value",data=combined_metrics2,palette="Set2
")
plt.title("View to Add Conversion rate")
plt.show()
```



From this visualization we can interpret, the control campaign demonstrates a higher View-to-Cart Conversion Rate compared to the test campaign. This indicates that users exposed to the control campaign are more likely to add products to their cart after viewing the ad.

```
merged_df=pd.concat([control_df,test_df],ignore_index=True)
merged_df
```

```
       Campaign Name       Date  Amount Spent  Number of Impressions
Reach  \
0   Control Campaign 2019-08-01          2280                  82702
56930
1   Control Campaign 2019-08-02          1757                 121040
102513
2   Control Campaign 2019-08-03          2343                 131711
110862
3   Control Campaign 2019-08-04          1940                  72878
61235
```

```
4   Control Campaign 2019-08-05          1835                  113430
91579
5   Control Campaign 2019-08-06          3083                  109076
87998
6   Control Campaign 2019-08-07          2544                  142123
127852
7   Control Campaign 2019-08-08          1900                   90939
65217
8   Control Campaign 2019-08-09          2813                  121332
94896
9   Control Campaign 2019-08-10          2149                  117624
91257
10  Control Campaign 2019-08-11          2490                  115247
95843
11  Control Campaign 2019-08-12          2319                  116639
100189
12  Control Campaign 2019-08-13          2697                   82847
68214
13  Control Campaign 2019-08-14          1875                  145248
118632
14  Control Campaign 2019-08-15          2774                  132845
102479
15  Control Campaign 2019-08-16          2024                   71274
42859
16  Control Campaign 2019-08-17          2177                  119612
106518
17  Control Campaign 2019-08-18          1876                  108452
96518
18  Control Campaign 2019-08-19          2596                  107890
81268
19  Control Campaign 2019-08-20          2675                  113430
78625
20  Control Campaign 2019-08-21          1803                   74654
59873
21  Control Campaign 2019-08-22          2939                  105705
86218
22  Control Campaign 2019-08-23          2496                  129880
109413
23  Control Campaign 2019-08-24          1892                   72515
51987
24  Control Campaign 2019-08-25          1962                  117006
100398
25  Control Campaign 2019-08-26          2233                  124897
98432
26  Control Campaign 2019-08-27          2061                  104678
91579
27  Control Campaign 2019-08-28          2421                  141654
125874
28  Control Campaign 2019-08-29          2375                   92029
```

```
74192
29  Control Campaign 2019-08-30          2324                  111306
88632
30     Test Campaign 2019-08-01          3008                   39550
35820
31     Test Campaign 2019-08-02          2542                  100719
91236
32     Test Campaign 2019-08-03          2365                   70263
45198
33     Test Campaign 2019-08-04          2710                   78451
25937
34     Test Campaign 2019-08-05          2297                  114295
95138
35     Test Campaign 2019-08-06          2458                   42684
31489
36     Test Campaign 2019-08-07          2838                   53986
42148
37     Test Campaign 2019-08-08          2916                   33669
20149
38     Test Campaign 2019-08-09          2652                   45511
31598
39     Test Campaign 2019-08-10          2790                   95054
79632
40     Test Campaign 2019-08-11          2420                   83633
71286
41     Test Campaign 2019-08-12          2831                  124591
10598
42     Test Campaign 2019-08-13          1972                   65827
49531
43     Test Campaign 2019-08-14          2537                   56304
25982
44     Test Campaign 2019-08-15          2516                   94338
76219
45     Test Campaign 2019-08-16          3076                  106584
81389
46     Test Campaign 2019-08-17          1968                   95843
54389
47     Test Campaign 2019-08-18          1979                   53632
43241
48     Test Campaign 2019-08-19          2626                   22521
10698
49     Test Campaign 2019-08-20          2712                   39470
31893
50     Test Campaign 2019-08-21          3112                  133771
109834
51     Test Campaign 2019-08-22          2899                   34752
27932
52     Test Campaign 2019-08-23          2407                   60286
49329
```

```
53     Test Campaign 2019-08-24          2078                   36650
30489
54     Test Campaign 2019-08-25          2928                  120576
105978
55     Test Campaign 2019-08-26          2311                   80841
61589
56     Test Campaign 2019-08-27          2915                  111469
92159
57     Test Campaign 2019-08-28          2247                   54627
41267
58     Test Campaign 2019-08-29          2805                   67444
43219
59     Test Campaign 2019-08-30          1977                  120203
89380

    Website Clicks  Searches Received  Content Viewed  Added to
Cart  \
0              7016                2290            2159            1819

1              8110                2033            1841            1219

2              6508                1737            1549            1134

3              3065                1042             982            1183

4              5224                2390            1984            1339

5              4028                1709            1249             784

6              2640                1388            1106            1166

7              7260                3047            2746             930

8              6198                2487            2179             645

9              2277                2475            1984            1629

10             8137                2941            2486            1887

11             2993                1397            1147            1439

12             6554                2390            1975            1794

13             4521                1209            1149            1339

14             4896                1179            1005            1641

15             5224                2427            2158            1613

16             6628                1756            1642             878
```

17	7253	2447	2115	1695
18	3706	2483	2098	908
19	2578	1001	848	1709
20	5691	2711	2496	1460
21	6843	3102	2988	819
22	4410	2896	2496	1913
23	4085	1274	1149	1146
24	4234	2423	2096	883
25	5435	2847	2421	1448
26	4941	3549	3249	980
27	6287	1672	1589	1711
28	8127	4891	4219	1486
29	4658	1615	1249	442
30	3038	1946	1069	894
31	4657	2359	1548	879
32	7885	2572	2367	1268
33	4216	2216	1437	566
34	5863	2106	858	956
35	7488	1854	1073	882
36	4221	2733	2182	1301
37	7184	2867	2194	1240
38	8259	2899	2761	1200
39	8125	2312	1804	424
40	3750	2893	2617	1075
41	8264	2081	1992	1382
42	7568	2213	2058	1391

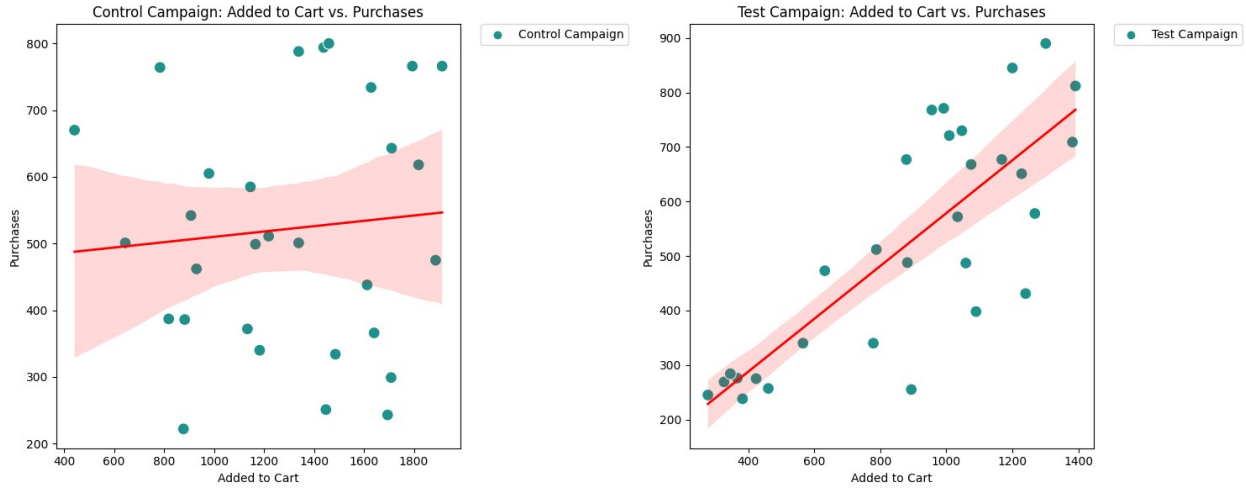
43	3993	1979	1059	779
44	4993	2537	1609	1090
45	6800	2661	2594	1059
46	7910	1995	1576	383
47	6909	2824	2522	461
48	7617	2924	2801	788
49	6050	2061	1894	1047
50	5471	1995	1868	278
51	4431	1983	1131	367
52	5077	2592	2004	632
53	7156	2687	2427	327
54	3596	2937	2551	1228
55	3820	2037	1046	346
56	6435	2976	2552	992
57	8144	2432	1281	1009
58	7651	1920	1240	1168
59	4399	2978	1625	1034

	Purchases	Click Through Rate	Conversion Rate
0	618	8.48	8.81
1	511	6.70	6.30
2	372	4.94	5.72
3	340	4.21	11.09
4	501	4.61	9.59
5	764	3.69	18.97
6	499	1.86	18.90
7	462	7.98	6.36
8	501	5.11	8.08
9	734	1.94	32.24
10	475	7.06	5.84
11	794	2.57	26.53
12	766	7.91	11.69
13	788	3.11	17.43
14	366	3.69	7.48

```
15         438                  7.33              8.38
16         222                  5.54              3.35
17         243                  6.69              3.35
18         542                  3.43             14.62
19         299                  2.27             11.60
20         800                  7.62             14.06
21         387                  6.47              5.66
22         766                  3.40             17.37
23         585                  5.63             14.32
24         386                  3.62              9.12
25         251                  4.35              4.62
26         605                  4.72             12.24
27         643                  4.44             10.23
28         334                  8.83              4.11
29         670                  4.18             14.38
30         255                  7.68              8.39
31         677                  4.62             14.54
32         578                 11.22              7.33
33         340                  5.37              8.06
34         768                  5.13             13.10
35         488                 17.54              6.52
36         890                  7.82             21.09
37         431                 21.34              6.00
38         845                 18.15             10.23
39         275                  8.55              3.38
40         668                  4.48             17.81
41         709                  6.63              8.58
42         812                 11.50             10.73
43         340                  7.09              8.51
44         398                  5.29              7.97
45         487                  6.38              7.16
46         238                  8.25              3.01
47         257                 12.88              3.72
48         512                 33.82              6.72
49         730                 15.33             12.07
50         245                  4.09              4.48
51         276                 12.75              6.23
52         473                  8.42              9.32
53         269                 19.53              3.76
54         651                  2.98             18.10
55         284                  4.73              7.43
56         771                  5.77             11.98
57         721                 14.91              8.85
58         677                 11.34              8.85
59         572                  3.66             13.00

merged_df.columns = merged_df.columns.str.strip()  # Removes
leading/trailing spaces
print(merged_df.columns)  # Verify column names again
```

```
Index(['Campaign Name', 'Date', 'Amount Spent', 'Number of
Impressions',
       'Reach', 'Website Clicks', 'Searches Received', 'Content
Viewed',
       'Added to Cart', 'Purchases', 'Click Through Rate', 'Conversion
Rate'],
      dtype='object')

plt.figure(figsize=(15, 6))

# Control Campaign
plt.subplot(1, 2, 1)
sns.scatterplot(
    data=control_df,
    x="Added to Cart",
    y="Purchases",
    hue="Campaign Name",
    palette="viridis",
    s=100,
)
sns.regplot(
    data=control_df,
    x="Added to Cart",
    y="Purchases",
    scatter=False,
    color="red",
    line_kws={"linewidth": 2},
)
plt.title("Control Campaign: Added to Cart vs. Purchases")
plt.legend(bbox_to_anchor=(1.05, 1), loc="upper left",
borderaxespad=0)

# Test Campaign
plt.subplot(1, 2, 2)
sns.scatterplot(
    data=test_df,
    x="Added to Cart",
    y="Purchases",
    hue="Campaign Name",
    palette="viridis",
    s=100,
)
sns.regplot(
    data=test_df,
    x="Added to Cart",
    y="Purchases",
    scatter=False,
    color="red",
    line_kws={"linewidth": 2},
)
```

```
plt.title("Test Campaign: Added to Cart vs. Purchases")
plt.legend(bbox_to_anchor=(1.05, 1), loc="upper left",
borderaxespad=0)

plt.tight_layout()
plt.show()
```



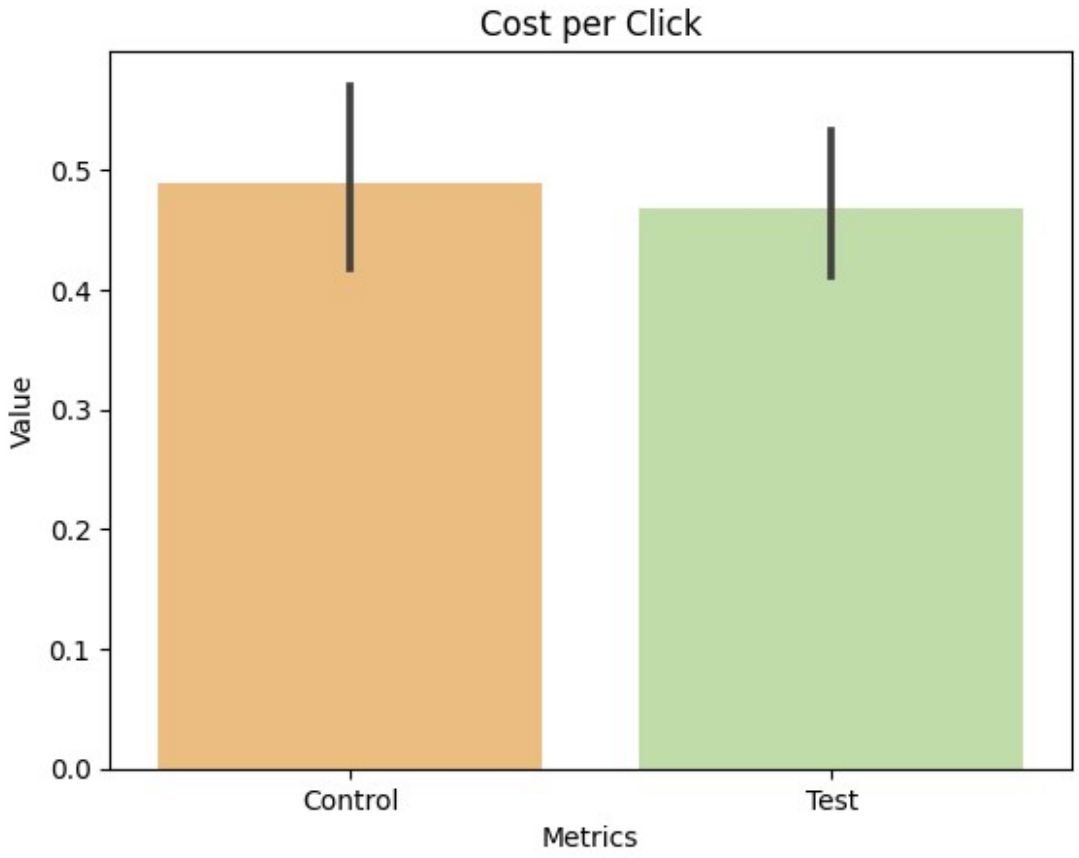
From the visulization, the test campaign is having higher positive correlation than control campaign.

```
# Calculate CPC for each campaign
control_df['CPC'] = round(control_df['Amount Spent'] /
control_df['Website Clicks'], 2)
test_df['CPC'] = round(test_df['Amount Spent'] / test_df['Website
Clicks'], 2)

df3 = pd.DataFrame({'Metrics': ["Control"] * len(control_df), 'Value':
control_df['CPC']})
df4 = pd.DataFrame({'Metrics': ["Test"] * len(test_df), 'Value':
test_df['CPC']})

combined_metrics3 = pd.concat([df3, df4], ignore_index=True)

print(combined_metrics3)
```

```
    Metrics  Value
0   Control   0.32
1   Control   0.22
2   Control   0.36
3   Control   0.63
4   Control   0.35
5   Control   0.77
6   Control   0.96
7   Control   0.26
```

```
8   Control   0.45
9   Control   0.94
10  Control   0.31
11  Control   0.77
12  Control   0.41
13  Control   0.41
14  Control   0.57
15  Control   0.39
16  Control   0.33
17  Control   0.26
18  Control   0.70
19  Control   1.04
20  Control   0.32
21  Control   0.43
22  Control   0.57
23  Control   0.46
24  Control   0.46
25  Control   0.41
26  Control   0.42
27  Control   0.39
28  Control   0.29
29  Control   0.50
30     Test   0.99
31     Test   0.55
32     Test   0.30
33     Test   0.64
34     Test   0.39
35     Test   0.33
36     Test   0.67
37     Test   0.41
38     Test   0.32
39     Test   0.34
40     Test   0.65
41     Test   0.34
42     Test   0.26
43     Test   0.64
44     Test   0.50
45     Test   0.45
46     Test   0.25
47     Test   0.29
48     Test   0.34
49     Test   0.45
50     Test   0.57
51     Test   0.65
52     Test   0.47
53     Test   0.29
54     Test   0.81
55     Test   0.60
56     Test   0.45
```

```
57     Test   0.28
58     Test   0.37
59     Test   0.45

sns.barplot(x='Metrics',y="Value",data=combined_metrics3,palette="Spectral")
plt.title("Cost per Click")
plt.show()
```



From above chart we can observe that the Cotrol campaign CPC is slightly higher than the Test campaign CPC. This indicates the control campaign might be spending more amount per click or might be having fewer clicks when compared to test campaign.

Hypothesis Testing

Hypothesis: Performing Control campaign result in higher number of ad conversions compared to performing Test Campaign

H0: There is no difference in number of conversions between Cotrol campaign and test Campaign

H1: There is a difference in number of conversions between Cotrol campaign and test Campaign

```
CR_C= round(Total1.mean(),2)
CR_T= round(Total2.mean(),2)
CTR_C=round(sum1.mean(),2)
CTR_T=round(sum2.mean(),2)

print("Mean Conversion Rate_____________________________")
print("Conversion Rate For Control:", CR_C)
print("Conversion Rate For Test:",CR_T)
ttest,p_value=stats.ttest_ind(a=control_df['Conversion
Rate'],b=test_df['Conversion Rate'],equal_var=False)
print(ttest)
print(p_value)
if p_value<0.05:
    print("We are rejecting the Null Hypothesis")
else:
    print('We are failed to reject the Null Hypothesis')
```

```
Mean Conversion Rate_____________________________
Conversion Rate For Control: 11.41
Conversion Rate For Test: 9.23
1.4829705689156862
0.14432230844288618
We are failed to reject the Null Hypothesis
```

The mean conversion rate of Control campaign is more or less similar to the mean conversion rate of test campaign. This suggests, on average, both the campaigns are having similar conversion rate.

Hypothesis: Performing Control campaign result in higher number of ad conversions compared to performing Test Campaign

H0: There is no difference in Click Through Rate between Cotrol campaign and test Campaign

H1: There is a difference in Click Through Rate between Cotrol campaign and test Campaign

```
print("Mean Click Through Rate_____________________________")
print("Click Through Rate For Control:",CTR_C)
print("Click Through Rate For Test:",CTR_T)
ttest,p_value=stats.ttest_ind(a=control_df['Click Through
Rate'],b=test_df['Click Through Rate'],equal_var=False)
print(ttest)
print(p_value)
if p_value<0.05:
    print("We are rejecting the Null Hypothesis")
else:
    print('We are failed to reject the Null Hypothesis')
```

```
Mean Click Through Rate_____________________________
Click Through Rate For Control: 5.08
Click Through Rate For Test: 10.24
```

```
-4.001968004017592
0.00032079103247097454
We are rejecting the Null Hypothesis
```

Here, we can observe that the T_test value is in negative which explains control campaign CTR is lesser than test campaign CTR.

The mean click through rate of test campaign(10.24) is higher than the mean click through rate of control camapign(5.08). This indicates the test campaign attracts more number of clicks compared to the control campaign.