

# A/B Testing

A/B Testing means analyzing two marketing strategies to choose the best marketing strategy that can convert more traffic into sales (or more traffic into your desired goal) effectively and efficiently

In A/B testing, we analyze the results of two marketing strategies to choose the best one for future marketing campaigns.

A/B testing means our goal can be to boost sales, followers, or traffic, but when we choose the best marketing strategy according to the results of our previous marketing campaigns, it is nothing but A/B testing.

In [7]: *# Importing important packages*

```
import pandas as pd
import numpy as np
import datetime as dt
import plotly
import plotly.graph_objects as go
import plotly.express as px
import plotly.io as pio
pio.template="plotly_dark"
```

In [9]: *# Reading CSV input files*

```
df1 = pd.read_csv(".\\Dataset\\CG.csv", sep= ";")
df2 = pd.read_csv(".\\Dataset\\TG.csv", sep= ";")
```

In [10]: *# Renaming columns and conversion on date object to date time format*

```
df1.rename(columns= {"Campaign Name":"Campaign Name",
                    "Date":"Date",
                    "Spend [USD]":"Amount Spent",
                    "# of Impressions":"No of Impressions",
                    "Reach":"Reach",
                    "# of Website Clicks":"Website Clicks",
                    "# of Searches":"Received Searches",
                    "# of View Content":"Content Viewed",
                    "# of Add to Cart":"Added to Cart",
                    "# of Purchase":"Purchases"}, inplace=True)
df2.rename(columns= {"Campaign Name":"Campaign Name",
                    "Date":"Date",
                    "Spend [USD]":"Amount Spent",
                    "# of Impressions":"No of Impressions",
                    "Reach":"Reach",
                    "# of Website Clicks":"Website Clicks",
                    "# of Searches":"Received Searches",
                    "# of View Content":"Content Viewed",
                    "# of Add to Cart":"Added to Cart",
                    "# of Purchase":"Purchases"}, inplace=True)

df1["Date"]=pd.to_datetime(df1["Date"], format="%d.%m.%Y")
df2["Date"]=pd.to_datetime(df2["Date"], format="%d.%m.%Y")
```

```
In [11]: # Data Cleaning
print(df1.isnull().sum()),print(df2.isnull().sum())
```

```
Campaign Name      0
Date               0
Amount Spent       0
No of Impressions  1
Reach              1
Website Clicks     1
Received Searches  1
Content Viewed     1
Added to Cart      1
Purchases          1
dtype: int64
Campaign Name      0
Date               0
Amount Spent       0
No of Impressions  0
Reach              0
Website Clicks     0
Received Searches  0
Content Viewed     0
Added to Cart      0
Purchases          0
dtype: int64
```

```
Out[11]: (None, None)
```

```
In [12]: # Replacing null values in of df1 each column with mean values of respective columns

df1["No of Impressions"].fillna(value=df1["No of Impressions"].mean(), inplace=True)
df1["Reach"].fillna(value=df1["Reach"].mean(), inplace=True)
df1["Website Clicks"].fillna(value=df1["Website Clicks"].mean(),inplace=True)
df1["Received Searches"].fillna(value=df1["Received Searches"].mean(),inplace=True)
df1["Content Viewed"].fillna(value=df1["Content Viewed"].mean(),inplace=True)
df1["Added to Cart"].fillna(value=df1["Added to Cart"].mean(), inplace=True)
df1["Purchases"].fillna(value=df1["Purchases"].mean(),inplace=True)
```

In [13]: *# Creating Merged Dataset for Analysis*

```
df3 = df1.merge(df2, how = "outer").sort_values(["Date"])
df3
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\reshape\merge.py:1113: UserWarning: You are merging on int and float columns where the float values are not equal to their
int representation
  warnings.warn(
```

Out[13]:

	Campaign Name	Date	Amount Spent	No of Impressions	Reach	Website Clicks	Received Searches	Content Viewed	Added to Cart	Purchases
0	Control Campaign	2019-08-01	2280	82702.000000	56930.000000	7016.000000	2290.000000	2159.000000	1819.0	618.000000
30	Test Campaign	2019-08-01	3008	39550.000000	35820.000000	3038.000000	1946.000000	1069.000000	894.0	255.000000
1	Control Campaign	2019-08-02	1757	121040.000000	102513.000000	8110.000000	2033.000000	1841.000000	1219.0	511.000000
31	Test Campaign	2019-08-02	2542	100719.000000	91236.000000	4657.000000	2359.000000	1548.000000	879.0	677.000000
2	Control Campaign	2019-08-03	2343	131711.000000	110862.000000	6508.000000	1737.000000	1549.000000	1134.0	372.000000
32	Test Campaign	2019-08-03	2365	70263.000000	45198.000000	7885.000000	2572.000000	2367.000000	1268.0	578.000000
3	Control Campaign	2019-08-04	1940	72878.000000	61235.000000	3065.000000	1042.000000	982.000000	1183.0	340.000000
33	Test Campaign	2019-08-04	2710	78451.000000	25937.000000	4216.000000	2216.000000	1437.000000	566.0	340.000000
4	Control Campaign	2019-08-05	1835	109559.758621	88844.931034	5320.793103	2221.310345	1943.793103	1300.0	522.793103
34	Test Campaign	2019-08-05	2297	114295.000000	95138.000000	5863.000000	2106.000000	858.000000	956.0	768.000000
35	Test Campaign	2019-08-06	2458	42684.000000	31489.000000	7488.000000	1854.000000	1073.000000	882.0	488.000000
5	Control Campaign	2019-08-06	3083	109076.000000	87998.000000	4028.000000	1709.000000	1249.000000	784.0	764.000000
36	Test Campaign	2019-08-07	2838	53986.000000	42148.000000	4221.000000	2733.000000	2182.000000	1301.0	890.000000
6	Control Campaign	2019-08-07	2544	142123.000000	127852.000000	2640.000000	1388.000000	1106.000000	1166.0	499.000000
37	Test Campaign	2019-08-08	2916	33669.000000	20149.000000	7184.000000	2867.000000	2194.000000	1240.0	431.000000
7	Control Campaign	2019-08-08	1900	90939.000000	65217.000000	7260.000000	3047.000000	2746.000000	930.0	462.000000
8	Control Campaign	2019-08-09	2813	121332.000000	94896.000000	6198.000000	2487.000000	2179.000000	645.0	501.000000
38	Test Campaign	2019-08-09	2652	45511.000000	31598.000000	8259.000000	2899.000000	2761.000000	1200.0	845.000000
39	Test Campaign	2019-08-10	2790	95054.000000	79632.000000	8125.000000	2312.000000	1804.000000	424.0	275.000000
9	Control Campaign	2019-08-10	2149	117624.000000	91257.000000	2277.000000	2475.000000	1984.000000	1629.0	734.000000
10	Control Campaign	2019-08-11	2490	115247.000000	95843.000000	8137.000000	2941.000000	2486.000000	1887.0	475.000000
40	Test Campaign	2019-08-11	2420	83633.000000	71286.000000	3750.000000	2893.000000	2617.000000	1075.0	668.000000
11	Control Campaign	2019-08-12	2319	116639.000000	100189.000000	2993.000000	1397.000000	1147.000000	1439.0	794.000000
41	Test Campaign	2019-08-12	2831	124591.000000	10598.000000	8264.000000	2081.000000	1992.000000	1382.0	709.000000
12	Control Campaign	2019-08-13	2697	82847.000000	68214.000000	6554.000000	2390.000000	1975.000000	1794.0	766.000000
42	Test Campaign	2019-08-13	1972	65827.000000	49531.000000	7568.000000	2213.000000	2058.000000	1391.0	812.000000
43	Test Campaign	2019-08-14	2537	56304.000000	25982.000000	3993.000000	1979.000000	1059.000000	779.0	340.000000
13	Control Campaign	2019-08-14	1875	145248.000000	118632.000000	4521.000000	1209.000000	1149.000000	1339.0	788.000000
44	Test Campaign	2019-08-15	2516	94338.000000	76219.000000	4993.000000	2537.000000	1609.000000	1090.0	398.000000
14	Control Campaign	2019-08-15	2774	132845.000000	102479.000000	4896.000000	1179.000000	1005.000000	1641.0	366.000000
45	Test Campaign	2019-08-16	3076	106584.000000	81389.000000	6800.000000	2661.000000	2594.000000	1059.0	487.000000
15	Control Campaign	2019-08-16	2024	71274.000000	42859.000000	5224.000000	2427.000000	2158.000000	1613.0	438.000000
16	Control Campaign	2019-08-17	2177	119612.000000	106518.000000	6628.000000	1756.000000	1642.000000	878.0	222.000000
46	Test Campaign	2019-08-17	1968	95843.000000	54389.000000	7910.000000	1995.000000	1576.000000	383.0	238.000000
47	Test Campaign	2019-08-18	1979	53632.000000	43241.000000	6909.000000	2824.000000	2522.000000	461.0	257.000000
17	Control Campaign	2019-08-18	1876	108452.000000	96518.000000	7253.000000	2447.000000	2115.000000	1695.0	243.000000
48	Test Campaign	2019-08-19	2626	22521.000000	10698.000000	7617.000000	2924.000000	2801.000000	788.0	512.000000
18	Control Campaign	2019-08-19	2596	107890.000000	81268.000000	3706.000000	2483.000000	2098.000000	908.0	542.000000

	Campaign Name	Date	Amount Spent	No of Impressions	Reach	Website Clicks	Received Searches	Content Viewed	Added to Cart	Purchases
49	Test Campaign	2019-08-20	2712	39470.000000	31893.000000	6050.000000	2061.000000	1894.000000	1047.0	730.000000
19	Control Campaign	2019-08-20	2675	113430.000000	78625.000000	2578.000000	1001.000000	848.000000	1709.0	299.000000
50	Test Campaign	2019-08-21	3112	133771.000000	109834.000000	5471.000000	1995.000000	1868.000000	278.0	245.000000
20	Control Campaign	2019-08-21	1803	74654.000000	59873.000000	5691.000000	2711.000000	2496.000000	1460.0	800.000000
51	Test Campaign	2019-08-22	2899	34752.000000	27932.000000	4431.000000	1983.000000	1131.000000	367.0	276.000000
21	Control Campaign	2019-08-22	2939	105705.000000	86218.000000	6843.000000	3102.000000	2988.000000	819.0	387.000000
52	Test Campaign	2019-08-23	2407	60286.000000	49329.000000	5077.000000	2592.000000	2004.000000	632.0	473.000000
22	Control Campaign	2019-08-23	2496	129880.000000	109413.000000	4410.000000	2896.000000	2496.000000	1913.0	766.000000
53	Test Campaign	2019-08-24	2078	36650.000000	30489.000000	7156.000000	2687.000000	2427.000000	327.0	269.000000
23	Control Campaign	2019-08-24	1892	72515.000000	51987.000000	4085.000000	1274.000000	1149.000000	1146.0	585.000000
24	Control Campaign	2019-08-25	1962	117006.000000	100398.000000	4234.000000	2423.000000	2096.000000	883.0	386.000000
54	Test Campaign	2019-08-25	2928	120576.000000	105978.000000	3596.000000	2937.000000	2551.000000	1228.0	651.000000
55	Test Campaign	2019-08-26	2311	80841.000000	61589.000000	3820.000000	2037.000000	1046.000000	346.0	284.000000
25	Control Campaign	2019-08-26	2233	124897.000000	98432.000000	5435.000000	2847.000000	2421.000000	1448.0	251.000000
56	Test Campaign	2019-08-27	2915	111469.000000	92159.000000	6435.000000	2976.000000	2552.000000	992.0	771.000000
26	Control Campaign	2019-08-27	2061	104678.000000	91579.000000	4941.000000	3549.000000	3249.000000	980.0	605.000000
57	Test Campaign	2019-08-28	2247	54627.000000	41267.000000	8144.000000	2432.000000	1281.000000	1009.0	721.000000
27	Control Campaign	2019-08-28	2421	141654.000000	125874.000000	6287.000000	1672.000000	1589.000000	1711.0	643.000000
58	Test Campaign	2019-08-29	2805	67444.000000	43219.000000	7651.000000	1920.000000	1240.000000	1168.0	677.000000
28	Control Campaign	2019-08-29	2375	92029.000000	74192.000000	8127.000000	4891.000000	4219.000000	1486.0	334.000000
29	Control Campaign	2019-08-30	2324	111306.000000	88632.000000	4658.000000	1615.000000	1249.000000	442.0	670.000000
59	Test Campaign	2019-08-30	1977	120203.000000	89380.000000	4399.000000	2978.000000	1625.000000	1034.0	572.000000

In [14]: 

```
# Counting samples of each campaign

df3["Campaign Name"].value_counts()
```

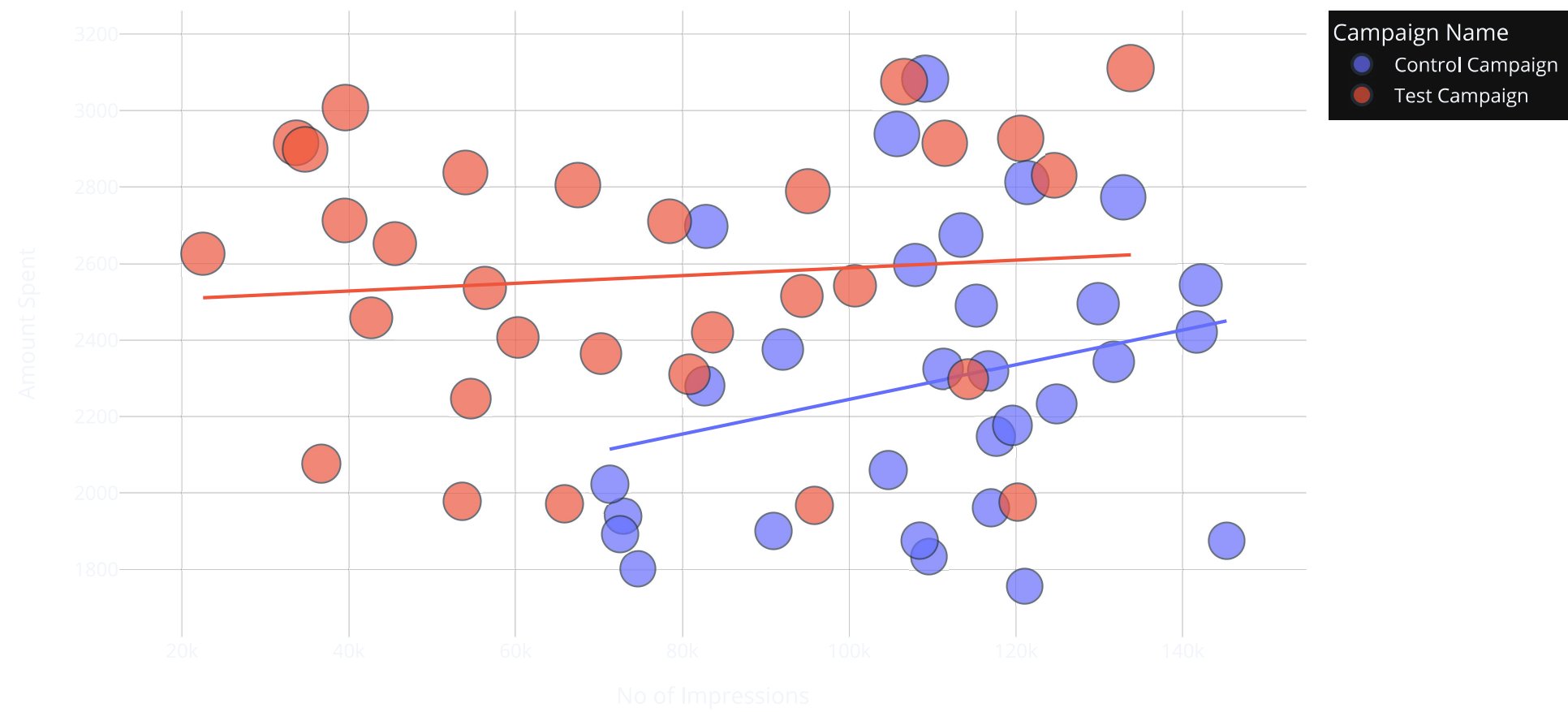
Out[14]: Test Campaign 30  
Control Campaign 30  
Name: Campaign Name, dtype: int64

```
In [15]: # A-B Testing

# 1) Analyzing the number of impressions vs Amount Spent from both campaigns

figure = plotly.express.scatter(data_frame = df3,
                                x="No of Impressions",
                                y="Amount Spent",
                                size="Amount Spent",
                                color= "Campaign Name",
                                trendline="ols", # ordinary least squares
                                template="plotly_dark")

figure.show()
```



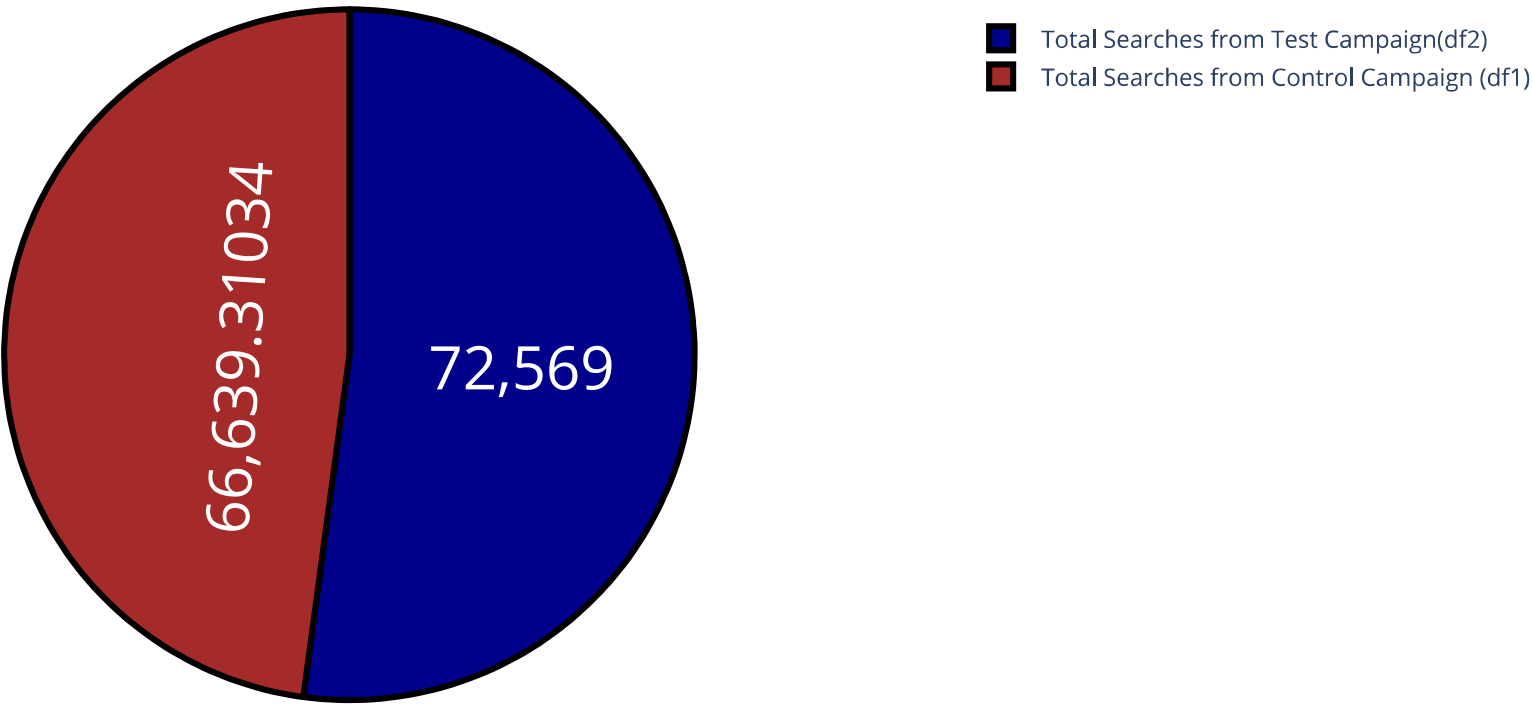
The control campaign generated a higher number of impressions relative to the amount spent on both campaigns.

```
In [16]: # 2) Total number of searches from both campaign

label = ["Total Searches from Control Campaign (df1)",
         "Total Searches from Test Campaign(df2)"]
counts = [sum(df1["Received Searches"]),
          sum(df2["Received Searches"])]
colors = ['brown', 'darkblue']
fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Control Vs Test: Searches')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
                  textfont_size=30,
                  marker=dict(colors=colors, line=dict(color='black', width=3)))

fig.show()
```

Control Vs Test: Searches

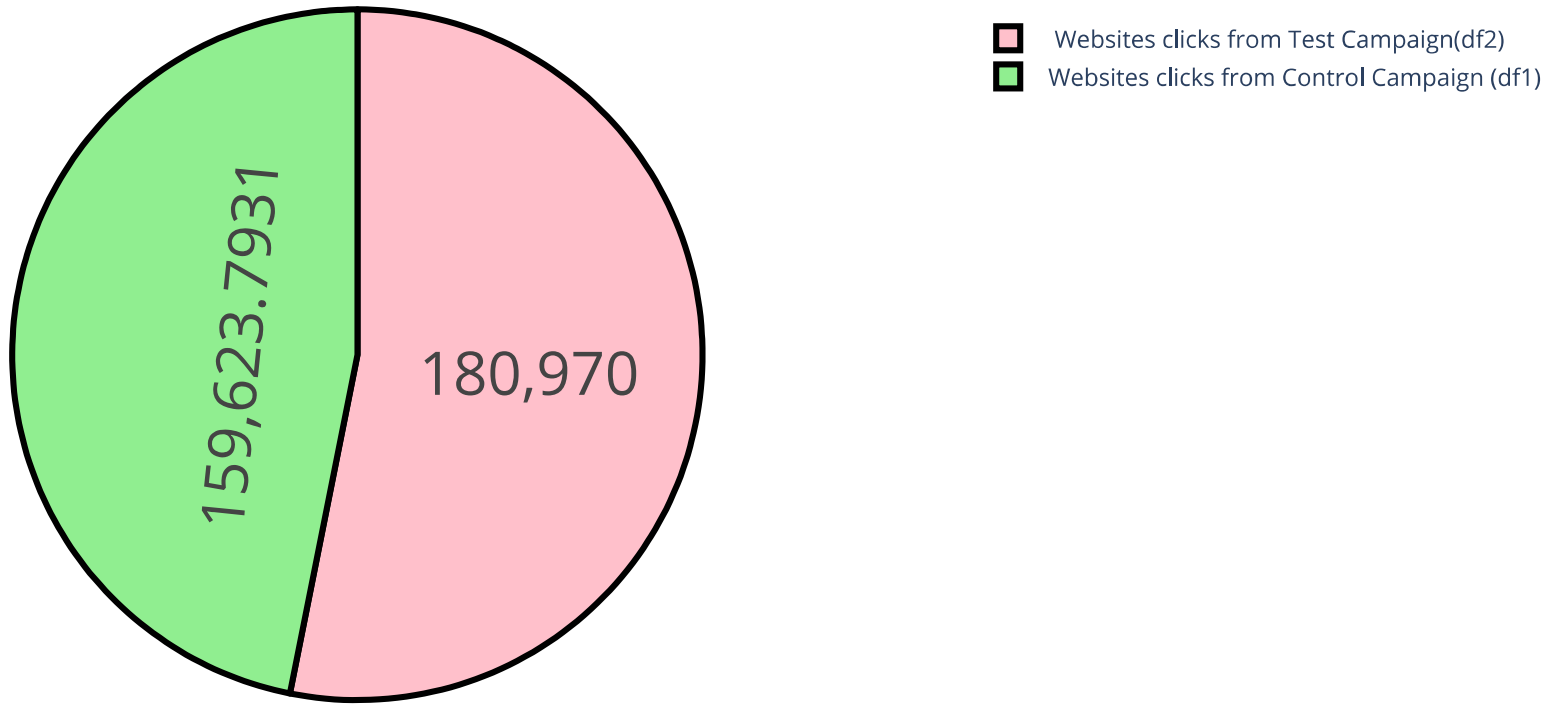


The test campaign yielded a higher number of searches on the website.

```
In [17]: # 3) Total number of website clicks from both campaign

label = ["Websites clicks from Control Campaign (df1)",
         " Websites clicks from Test Campaign(df2)"]
counts = [sum(df1["Website Clicks"]),
          sum(df2["Website Clicks"])]
colors = ['lightgreen', 'pink']
fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Control Vs Test: Searches')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
                  textfont_size=30,
                  marker=dict(colors=colors, line=dict(color='black', width=3)))
fig.show()
```

Control Vs Test: Searches



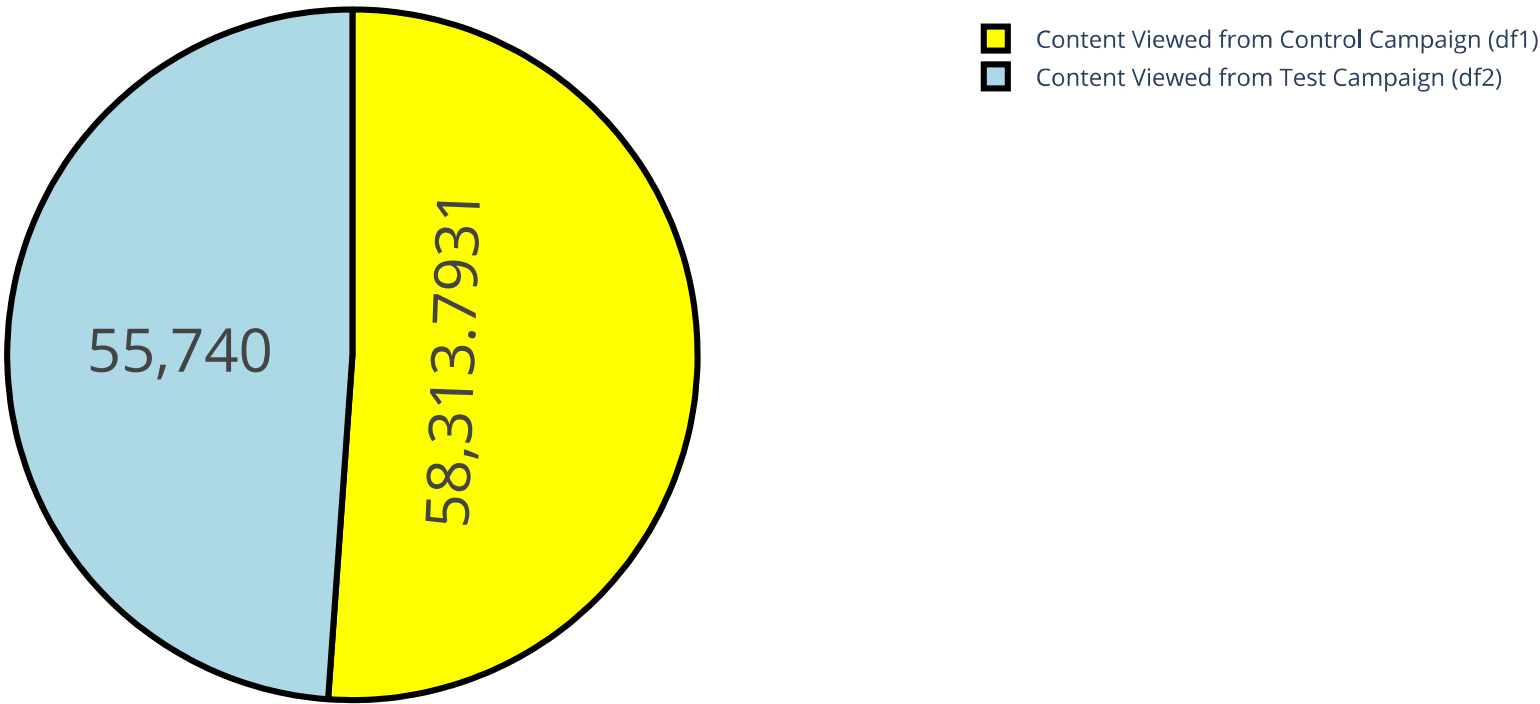
The test campaign wins in the number of website clicks.



In [18]: *# Amount of content viewed after reaching the website from both campaigns*

```
label = ["Content Viewed from Control Campaign (df1) ",
        "Content Viewed from Test Campaign (df2)"]
counts = [sum(df1["Content Viewed"]),
          sum(df2["Content Viewed"])]
colors = ['yellow', 'lightblue']
fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Control Vs Test: Content Viewed')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
                  textfont_size=30,
                  marker=dict(colors=colors,
                              line=dict(color='black', width=3)))
fig.show()
```

Control Vs Test: Content Viewed



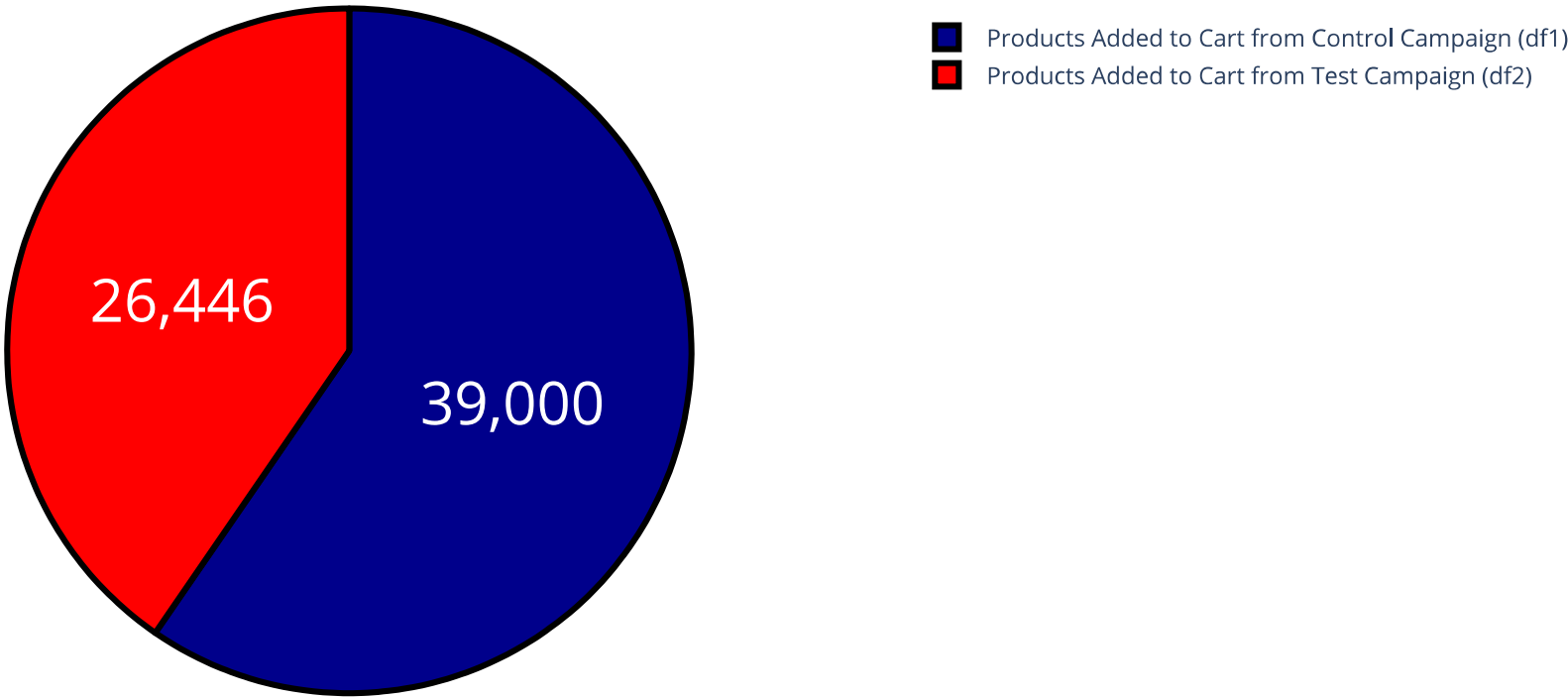
The audience of the control campaign exhibited a higher content viewing rate compared to the test campaign. Despite the marginal difference, the control campaign, with its lower website click count, demonstrated greater engagement on the website than the test campaign.

```
In [19]: # 3) Number of products added to the cart from both campaigns

label = ["Products Added to Cart from Control Campaign (df1)",
         "Products Added to Cart from Test Campaign (df2) "]
counts = [sum(df1["Added to Cart"]),
          sum(df2["Added to Cart"])]
colors = ['darkblue','red']
fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Control Vs Test: Added to Cart')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
                  textfont_size=30,
                  marker=dict(colors=colors,
                              line=dict(color='black', width=3)))

fig.show()
```

Control Vs Test: Added to Cart



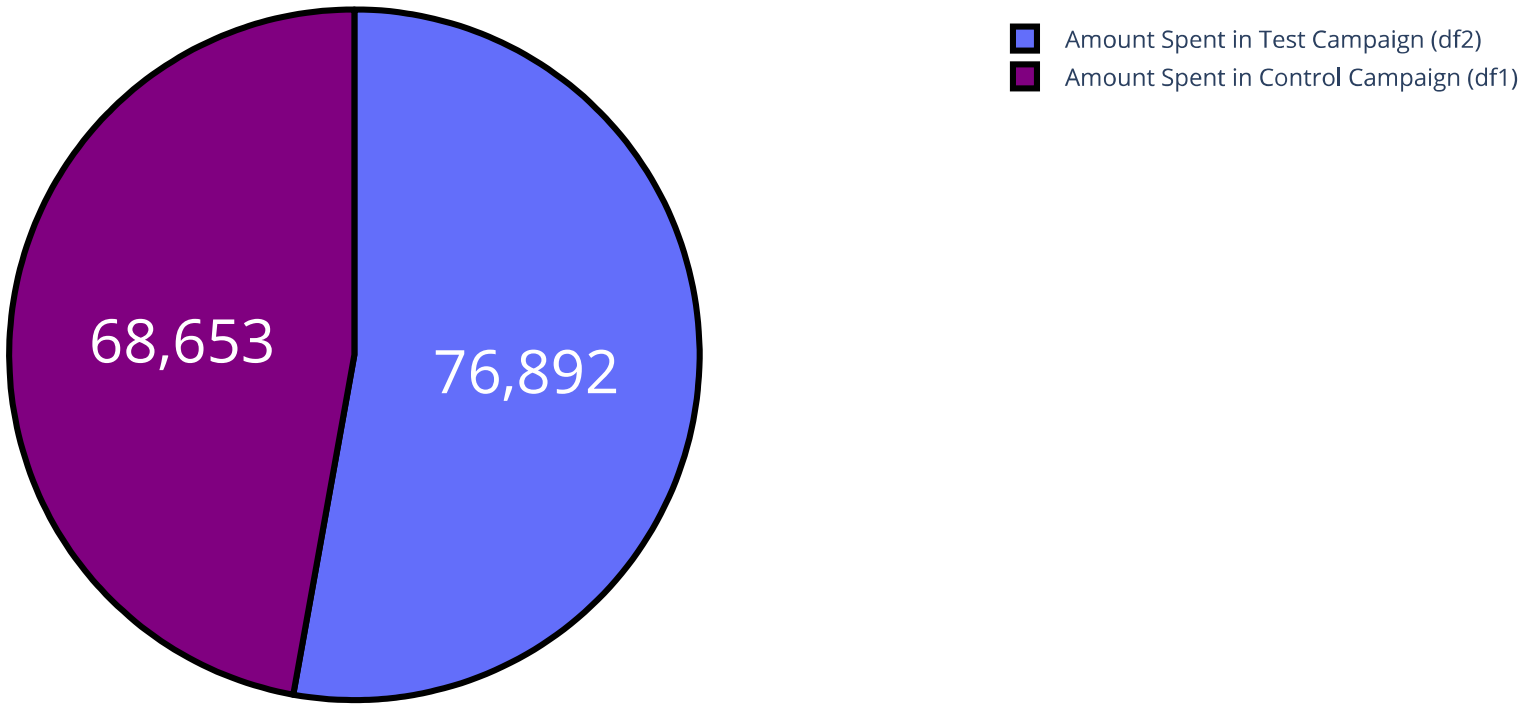
Despite experiencing fewer website clicks, the control campaign managed to accumulate more products in the cart.

```
In [20]: # 4)Amount spent on both campaigns

label = ["Amount Spent in Control Campaign (df1)",
         "Amount Spent in Test Campaign (df2)"]
counts = [sum(df1["Amount Spent"]),
          sum(df2["Amount Spent"])]
colors = ['purple', 'pastelgreen']
fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Control Vs Test: Amount Spent')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
                  textfont_size=30,
                  marker=dict(colors=colors,
                              line=dict(color='black', width=3)))

fig.show()
```

Control Vs Test: Amount Spent



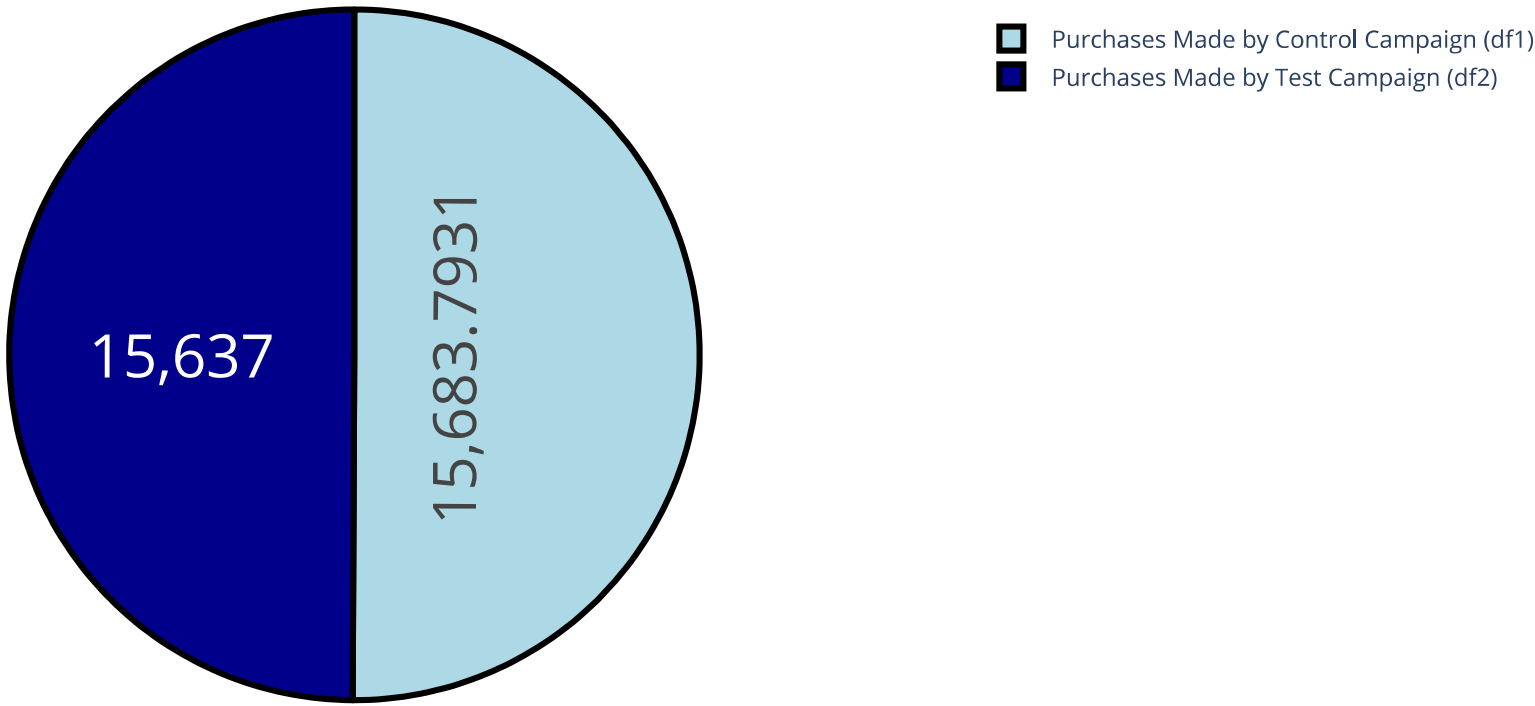
Although the test campaign incurred a higher expenditure compared to the control campaign, the latter yielded more content views and a greater number of products added to the cart. Therefore, the control campaign demonstrates greater efficiency than the test campaign.

```
In [21]: # 5) Purchases made by both campaigns

label = ["Purchases Made by Control Campaign (df1)",
         "Purchases Made by Test Campaign (df2) "]
counts = [sum(df1["Purchases"]),
          sum(df2["Purchases"])]
colors = ['lightblue', 'darkblue']
fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Control Vs Test: Purchases')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
                  textfont_size=30,
                  marker=dict(colors=colors,
                              line=dict(color='black', width=3)))

fig.show()
```

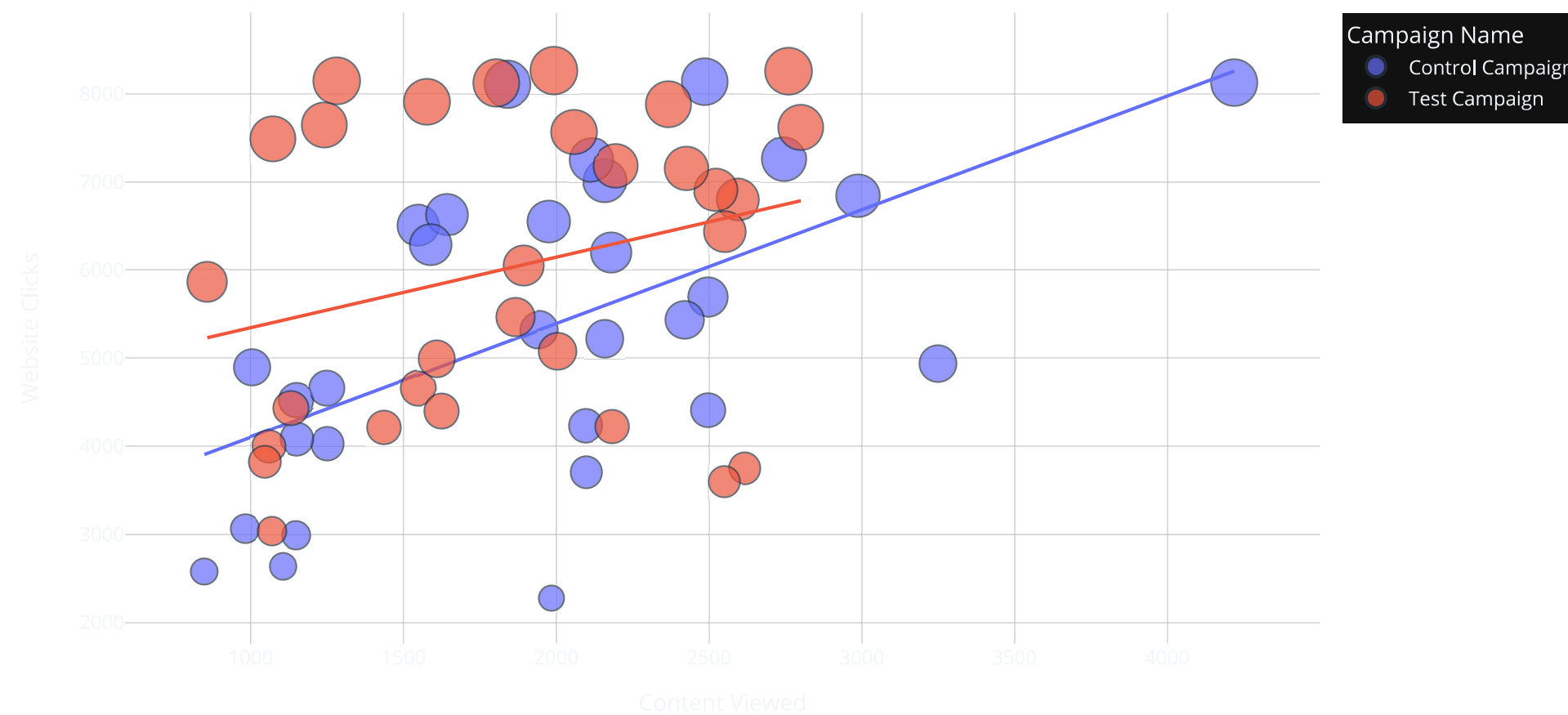
Control Vs Test: Purchases



The variance in purchases between both ad campaigns is merely approximately 1%. Since the control campaign generated more sales with a lower marketing expenditure, it prevails in this comparison

```
In [22]: # Analyzing which campaign converted more  
# The relationship between website clicks and content viewed from both campaigns
```

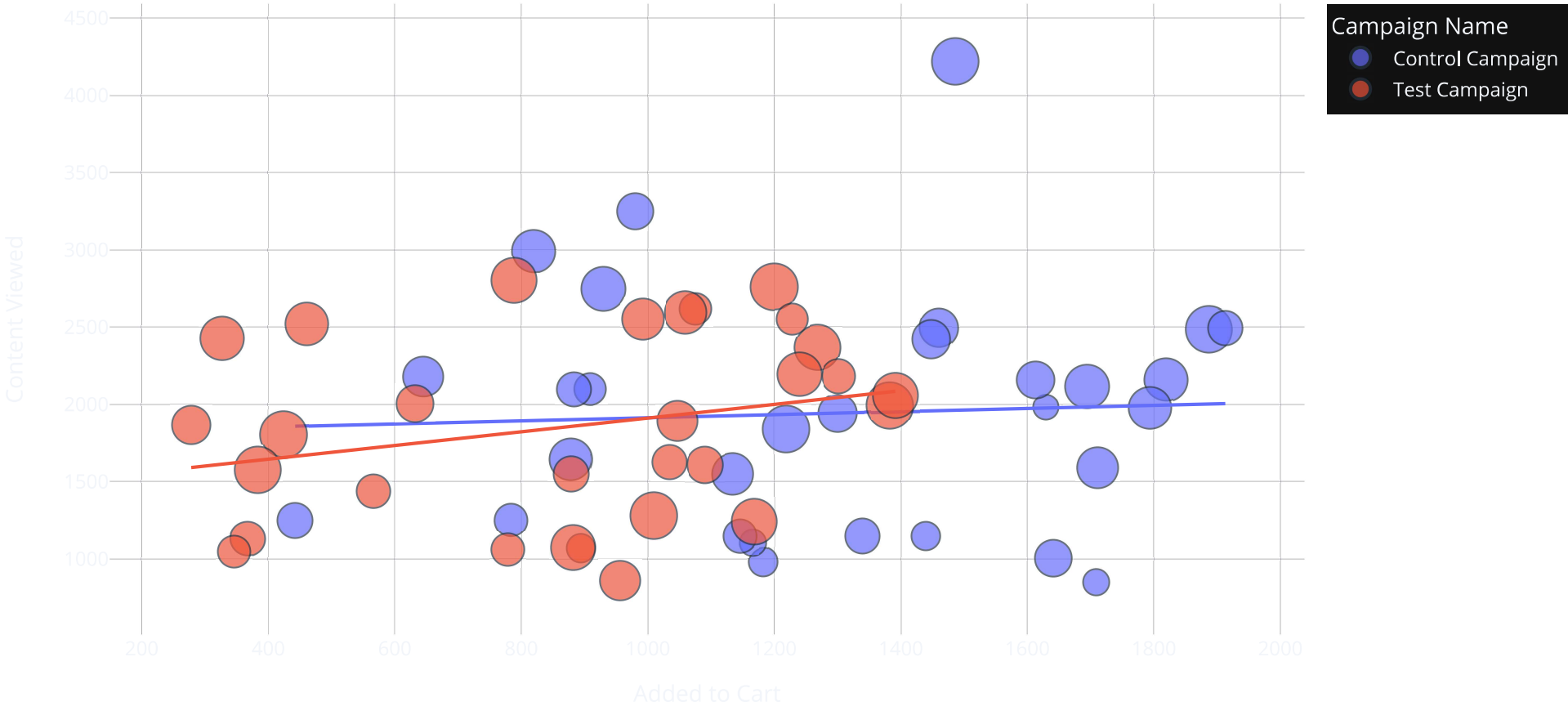
```
figure = px.scatter(data_frame = df3,  
                    x="Content Viewed",  
                    y="Website Clicks",  
                    size="Website Clicks",  
                    color= "Campaign Name",  
                    trendline="ols",  
                    template = "plotly_dark")  
figure.show()
```



Website clicks register higher numbers in the test campaign, yet engagement stemming from these clicks is greater in the control campaign. Therefore, the control campaign emerges as the winner.

In [23]: *# The relationship between the Amount of content viewed and the number of products added to the cart from both campaigns*

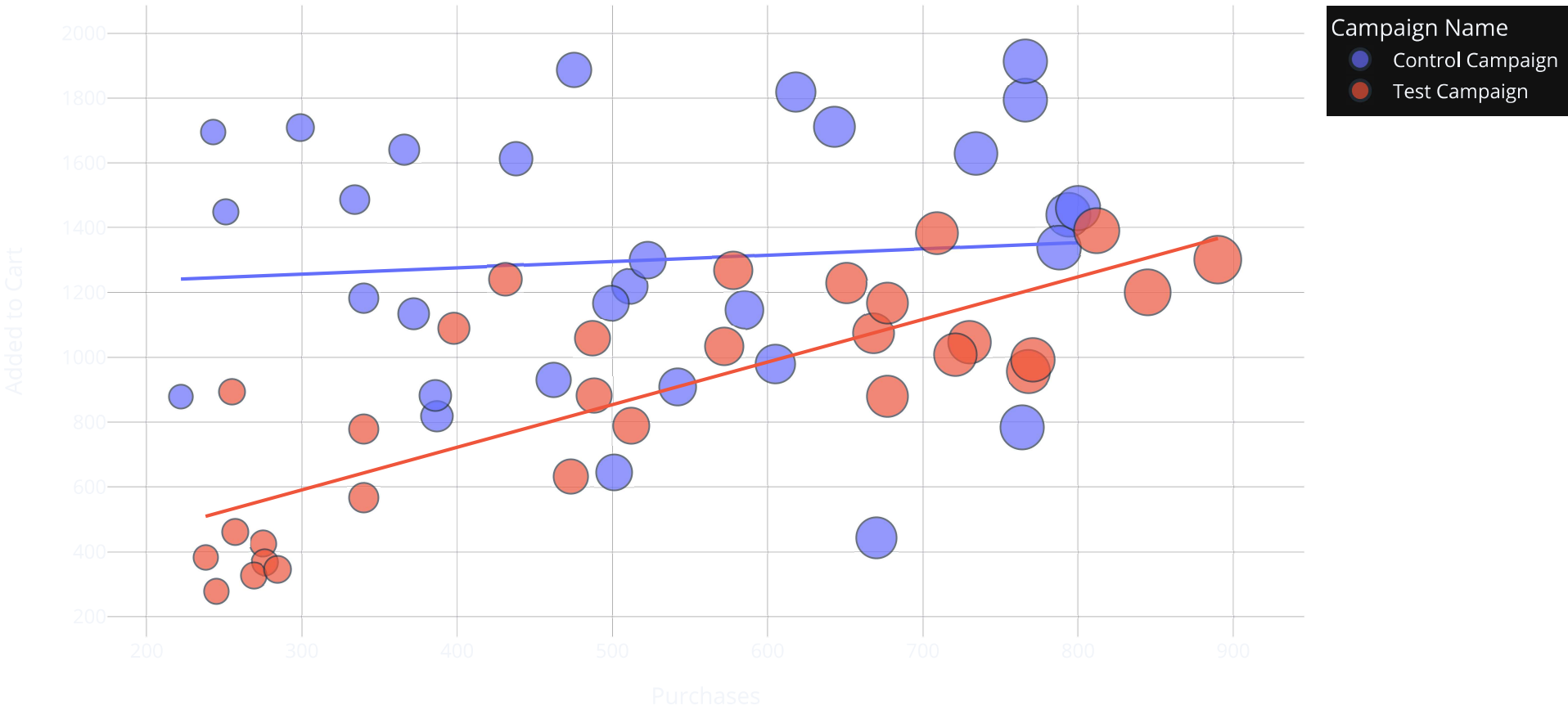
```
figure = px.scatter(data_frame = df3,
                    x="Added to Cart",
                    y="Content Viewed",
                    size="Website Clicks",
                    color= "Campaign Name",
                    trendline="ols",
                    template = "plotly_dark")
figure.show()
```



Once again, the control campaign emerges victorious

In [24]: *# The relationship between the number of products added to the cart and the number of sales from both campaigns*

```
figure = px.scatter(data_frame = df3,
                    x="Purchases",
                    y="Added to Cart",
                    size="Purchases",
                    color= "Campaign Name",
                    trendline="ols",
                    template = "plotly_dark")
figure.show()
```



Based on the graph above, it's clear that the control campaign achieved higher sales and had a greater number of products added to the cart. However, the conversion rate of the test campaign appears to be higher.

## Conclusion

Based on the results of the A/B tests conducted, it was observed that the control campaign generated higher sales and engagement among visitors. The control campaign led to a greater number of product views, resulting in more items being added to the cart and subsequently leading to increased sales. However, the conversion rate of products added to the cart was found to be higher in the test campaign. Despite this, the test campaign yielded more sales in relation to the number of products viewed and added to the cart. Overall, the control campaign exhibited higher sales performance. Consequently, the test campaign could be utilized for targeted marketing of specific products to a particular audience, while the control campaign is better suited for promoting multiple products to a broader audience.

