A/B Testing Case Study

A/B Testing Case Study A/B testing helps in finding a better approach to finding customers, marketing products, getting a higher reach, or anything that helps a business convert most of its target customers into actual customers.

Here is a dataset based on A/B testing submitted by İlker Yıldız on Kaggle. Below are all the features in the dataset:

- Campaign Name: The name of the campaign
- Date: Date of the record
- Spend: Amount spent on the campaign in dollars
- of Impressions: Number of impressions the ad crossed through the campaign
- Reach: The number of unique impressions received in the ad
- of Website Clicks: Number of website clicks received through the ads
- of Searches: Number of users who performed searches on the website
- of View Content: Number of users who viewed content and products on the website
- of Add to Cart: Number of users who added products to the cart
- of Purchase: Number of purchases

Two campaigns were performed by the company:
1. Control Campaign
2. Test Campaign
Perform A/B testing to find the best campaign for the company to get more customers

In [1]:
```
import plotly.graph_objects as go
import plotly.express as px
import plotly.io as pio
pio.templates.default = "plotly_white"
import datetime
import pandas as pd
```

In [2]:
```
control_data = pd.read_csv('Data/control_group.csv',sep=";")
test_data = pd.read_csv('Data/test_group.csv',sep=';')
```

In [3]:
```
control_data.head()
```
Out[3]:

	Campaign Name	Date	Spend [USD]	# of Impressions	Reach	# of Website Clicks	# of Searches	# of View Content	# of Add to Cart	# of Purchase
0	Control Campaign	1.08.2019	2280	82702.0	56930.0	7016.0	2290.0	2159.0	1819.0	618.0
1	Control Campaign	2.08.2019	1757	121040.0	102513.0	8110.0	2033.0	1841.0	1219.0	511.0
2	Control Campaign	3.08.2019	2343	131711.0	110862.0	6508.0	1737.0	1549.0	1134.0	372.0
3	Control Campaign	4.08.2019	1940	72878.0	61235.0	3065.0	1042.0	982.0	1183.0	340.0
4	Control Campaign	5.08.2019	1835	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [4]:
```
test_data.head()
```
Out[4]:

	Campaign Name	Date	Spend [USD]	# of Impressions	Reach	# of Website Clicks	# of Searches	# of View Content	# of Add to Cart	# of Purchase
0	Test Campaign	1.08.2019	3008	39550	35820	3038	1946	1069	894	255
1	Test Campaign	2.08.2019	2542	100719	91236	4657	2359	1548	879	677
2	Test Campaign	3.08.2019	2365	70263	45198	7885	2572	2367	1268	578
3	Test Campaign	4.08.2019	2710	78451	25937	4216	2216	1437	566	340
4	Test Campaign	5.08.2019	2297	114295	95138	5863	2106	858	956	768

Data Preparation

The datasets have some errors in column names. Let's give new column names before moving forward:

In [5]:
```
control_data.columns = ["Campaign Name", "Date", "Amount Spent",
                        "Number of Impressions", "Reach", "Website Clicks",
                        "Searches Received", "Content Viewed", "Added to Cart",
                        "Purchases"]
test_data.columns = ["Campaign Name", "Date", "Amount Spent",
                        "Number of Impressions", "Reach", "Website Clicks",
                        "Searches Received", "Content Viewed", "Added to Cart",
                        "Purchases"]
```

Now let's see if the datasets have null values or not:

In [6]:
```
control_data.isnull().sum()
```

Out[6]:
```
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
```

In [7]:
```
test_data.isnull().sum()
```

Out[7]:
```
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
```

The dataset of the control campaign has missing values in a row. Let’s fill in these missing values by the mean value of each column:

In [8]:
```
control_data['Number of Impressions'].fillna(value=control_data['Number of Impressions'].mean(),inplace=True)
control_data['Reach'].fillna(value=control_data['Reach'].mean(),inplace=True)
control_data['Website Clicks'].fillna(value=control_data['Website Clicks'].mean(),inplace=True)
control_data['Searches Received'].fillna(value=control_data['Searches Received'].mean(),inplace=True)
control_data['Content Viewed'].fillna(value=control_data['Content Viewed'].mean(),inplace=True)
control_data['Added to Cart'].fillna(value=control_data['Added to Cart'].mean(),inplace=True)
control_data['Purchases'].fillna(value=control_data['Purchases'].mean(),inplace=True)
```

Now I will create a new dataset by merging both datasets:

In [9]:
```
ab_data = control_data.merge(test_data,how='outer').sort_values(['Date'])
ab_data = ab_data.reset_index(drop=True)
print(ab_data)
```

	Campaign Name	Date	Amount Spent	Number of Impressions	\
0	Control Campaign	1.08.2019	2280	82702.000000	
1	Test Campaign	1.08.2019	3008	39550.000000	
2	Test Campaign	10.08.2019	2790	95054.000000	
3	Control Campaign	10.08.2019	2149	117624.000000	
4	Test Campaign	11.08.2019	2420	83633.000000	
5	Control Campaign	11.08.2019	2490	115247.000000	
6	Test Campaign	12.08.2019	2831	124591.000000	
7	Control Campaign	12.08.2019	2319	116639.000000	
8	Test Campaign	13.08.2019	1972	65827.000000	
9	Control Campaign	13.08.2019	2697	82847.000000	
10	Test Campaign	14.08.2019	2537	56304.000000	
11	Control Campaign	14.08.2019	1875	145248.000000	
12	Test Campaign	15.08.2019	2516	94338.000000	
13	Control Campaign	15.08.2019	2774	132845.000000	
14	Control Campaign	16.08.2019	2024	71274.000000	
15	Test Campaign	16.08.2019	3076	106584.000000	
16	Control Campaign	17.08.2019	2177	119612.000000	
17	Test Campaign	17.08.2019	1968	95843.000000	
18	Control Campaign	18.08.2019	1876	108452.000000	
19	Test Campaign	18.08.2019	1979	53632.000000	
20	Control Campaign	19.08.2019	2596	107890.000000	
21	Test Campaign	19.08.2019	2626	22521.000000	
22	Control Campaign	2.08.2019	1757	121040.000000	
23	Test Campaign	2.08.2019	2542	100719.000000	
24	Control Campaign	20.08.2019	2675	113430.000000	
25	Test Campaign	20.08.2019	2712	39470.000000	
26	Control Campaign	21.08.2019	1803	74654.000000	
27	Test Campaign	21.08.2019	3112	133771.000000	
28	Control Campaign	22.08.2019	2939	105705.000000	
29	Test Campaign	22.08.2019	2899	34752.000000	
30	Control Campaign	23.08.2019	2496	129880.000000	
31	Test Campaign	23.08.2019	2407	60286.000000	
32	Control Campaign	24.08.2019	1892	72515.000000	
33	Test Campaign	24.08.2019	2078	36650.000000	
34	Control Campaign	25.08.2019	1962	117006.000000	
35	Test Campaign	25.08.2019	2928	120576.000000	
36	Control Campaign	26.08.2019	2233	124897.000000	
37	Test Campaign	26.08.2019	2311	80841.000000	
38	Control Campaign	27.08.2019	2061	104678.000000	
39	Test Campaign	27.08.2019	2915	111469.000000	
40	Control Campaign	28.08.2019	2421	141654.000000	
41	Test Campaign	28.08.2019	2247	54627.000000	

42	Test Campaign	29.08.2019	2805	67444.000000
43	Control Campaign	29.08.2019	2375	92029.000000
44	Test Campaign	3.08.2019	2365	70263.000000
45	Control Campaign	3.08.2019	2343	131711.000000
46	Control Campaign	30.08.2019	2324	111306.000000
47	Test Campaign	30.08.2019	1977	120203.000000
48	Test Campaign	4.08.2019	2710	78451.000000
49	Control Campaign	4.08.2019	1940	72878.000000
50	Test Campaign	5.08.2019	2297	114295.000000
51	Control Campaign	5.08.2019	1835	109559.758621
52	Test Campaign	6.08.2019	2458	42684.000000
53	Control Campaign	6.08.2019	3083	109076.000000
54	Test Campaign	7.08.2019	2838	53986.000000
55	Control Campaign	7.08.2019	2544	142123.000000
56	Test Campaign	8.08.2019	2916	33669.000000
57	Control Campaign	8.08.2019	1900	90939.000000
58	Control Campaign	9.08.2019	2813	121332.000000
59	Test Campaign	9.08.2019	2652	45511.000000

	Reach	Website Clicks	Searches Received	Content Viewed	\
0	56930.000000	7016.000000	2290.000000	2159.000000	
1	35820.000000	3038.000000	1946.000000	1069.000000	
2	79632.000000	8125.000000	2312.000000	1804.000000	
3	91257.000000	2277.000000	2475.000000	1984.000000	
4	71286.000000	3750.000000	2893.000000	2617.000000	
5	95843.000000	8137.000000	2941.000000	2486.000000	
6	10598.000000	8264.000000	2081.000000	1992.000000	
7	100189.000000	2993.000000	1397.000000	1147.000000	
8	49531.000000	7568.000000	2213.000000	2058.000000	
9	68214.000000	6554.000000	2390.000000	1975.000000	
10	25982.000000	3993.000000	1979.000000	1059.000000	
11	118632.000000	4521.000000	1209.000000	1149.000000	
12	76219.000000	4993.000000	2537.000000	1609.000000	
13	102479.000000	4896.000000	1179.000000	1005.000000	
14	42859.000000	5224.000000	2427.000000	2158.000000	
15	81389.000000	6800.000000	2661.000000	2594.000000	
16	106518.000000	6628.000000	1756.000000	1642.000000	
17	54389.000000	7910.000000	1995.000000	1576.000000	
18	96518.000000	7253.000000	2447.000000	2115.000000	
19	43241.000000	6909.000000	2824.000000	2522.000000	
20	81268.000000	3706.000000	2483.000000	2098.000000	
21	10698.000000	7617.000000	2924.000000	2801.000000	
22	102513.000000	8110.000000	2033.000000	1841.000000	

23	91236.000000	4657.000000	2359.000000	1548.000000
24	78625.000000	2578.000000	1001.000000	848.000000
25	31893.000000	6050.000000	2061.000000	1894.000000
26	59873.000000	5691.000000	2711.000000	2496.000000
27	109834.000000	5471.000000	1995.000000	1868.000000
28	86218.000000	6843.000000	3102.000000	2988.000000
29	27932.000000	4431.000000	1983.000000	1131.000000
30	109413.000000	4410.000000	2896.000000	2496.000000
31	49329.000000	5077.000000	2592.000000	2004.000000
32	51987.000000	4085.000000	1274.000000	1149.000000
33	30489.000000	7156.000000	2687.000000	2427.000000
34	100398.000000	4234.000000	2423.000000	2096.000000
35	105978.000000	3596.000000	2937.000000	2551.000000
36	98432.000000	5435.000000	2847.000000	2421.000000
37	61589.000000	3820.000000	2037.000000	1046.000000
38	91579.000000	4941.000000	3549.000000	3249.000000
39	92159.000000	6435.000000	2976.000000	2552.000000
40	125874.000000	6287.000000	1672.000000	1589.000000
41	41267.000000	8144.000000	2432.000000	1281.000000
42	43219.000000	7651.000000	1920.000000	1240.000000
43	74192.000000	8127.000000	4891.000000	4219.000000
44	45198.000000	7885.000000	2572.000000	2367.000000
45	110862.000000	6508.000000	1737.000000	1549.000000
46	88632.000000	4658.000000	1615.000000	1249.000000
47	89380.000000	4399.000000	2978.000000	1625.000000
48	25937.000000	4216.000000	2216.000000	1437.000000
49	61235.000000	3065.000000	1042.000000	982.000000
50	95138.000000	5863.000000	2106.000000	858.000000
51	88844.931034	5320.793103	2221.310345	1943.793103
52	31489.000000	7488.000000	1854.000000	1073.000000
53	87998.000000	4028.000000	1709.000000	1249.000000
54	42148.000000	4221.000000	2733.000000	2182.000000
55	127852.000000	2640.000000	1388.000000	1106.000000
56	20149.000000	7184.000000	2867.000000	2194.000000
57	65217.000000	7260.000000	3047.000000	2746.000000
58	94896.000000	6198.000000	2487.000000	2179.000000
59	31598.000000	8259.000000	2899.000000	2761.000000

	Added to Cart	Purchases
0	1819.0	618.000000
1	894.0	255.000000
2	424.0	275.000000
3	1629.0	734.000000

4	1075.0	668.000000
5	1887.0	475.000000
6	1382.0	709.000000
7	1439.0	794.000000
8	1391.0	812.000000
9	1794.0	766.000000
10	779.0	340.000000
11	1339.0	788.000000
12	1090.0	398.000000
13	1641.0	366.000000
14	1613.0	438.000000
15	1059.0	487.000000
16	878.0	222.000000
17	383.0	238.000000
18	1695.0	243.000000
19	461.0	257.000000
20	908.0	542.000000
21	788.0	512.000000
22	1219.0	511.000000
23	879.0	677.000000
24	1709.0	299.000000
25	1047.0	730.000000
26	1460.0	800.000000
27	278.0	245.000000
28	819.0	387.000000
29	367.0	276.000000
30	1913.0	766.000000
31	632.0	473.000000
32	1146.0	585.000000
33	327.0	269.000000
34	883.0	386.000000
35	1228.0	651.000000
36	1448.0	251.000000
37	346.0	284.000000
38	980.0	605.000000
39	992.0	771.000000
40	1711.0	643.000000
41	1009.0	721.000000
42	1168.0	677.000000
43	1486.0	334.000000
44	1268.0	578.000000
45	1134.0	372.000000
46	442.0	670.000000

```
47          1034.0  572.000000
48           566.0  340.000000
49          1183.0  340.000000
50           956.0  768.000000
51          1300.0  522.793103
52           882.0  488.000000
53           784.0  764.000000
54          1301.0  890.000000
55          1166.0  499.000000
56          1240.0  431.000000
57           930.0  462.000000
58           645.0  501.000000
59          1200.0  845.000000
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_7908\2576005690.py:1: UserWarning: You are merging on int and flo
at columns where the float values are not equal to their int representation.
  ab_data = control_data.merge(test_data,how='outer').sort_values(['Date'])
```

Before moving forward, let’s have a look if the dataset has an equal number of samples about both campaigns:

In [10]:
```
print(ab_data['Campaign Name'].value_counts())
```

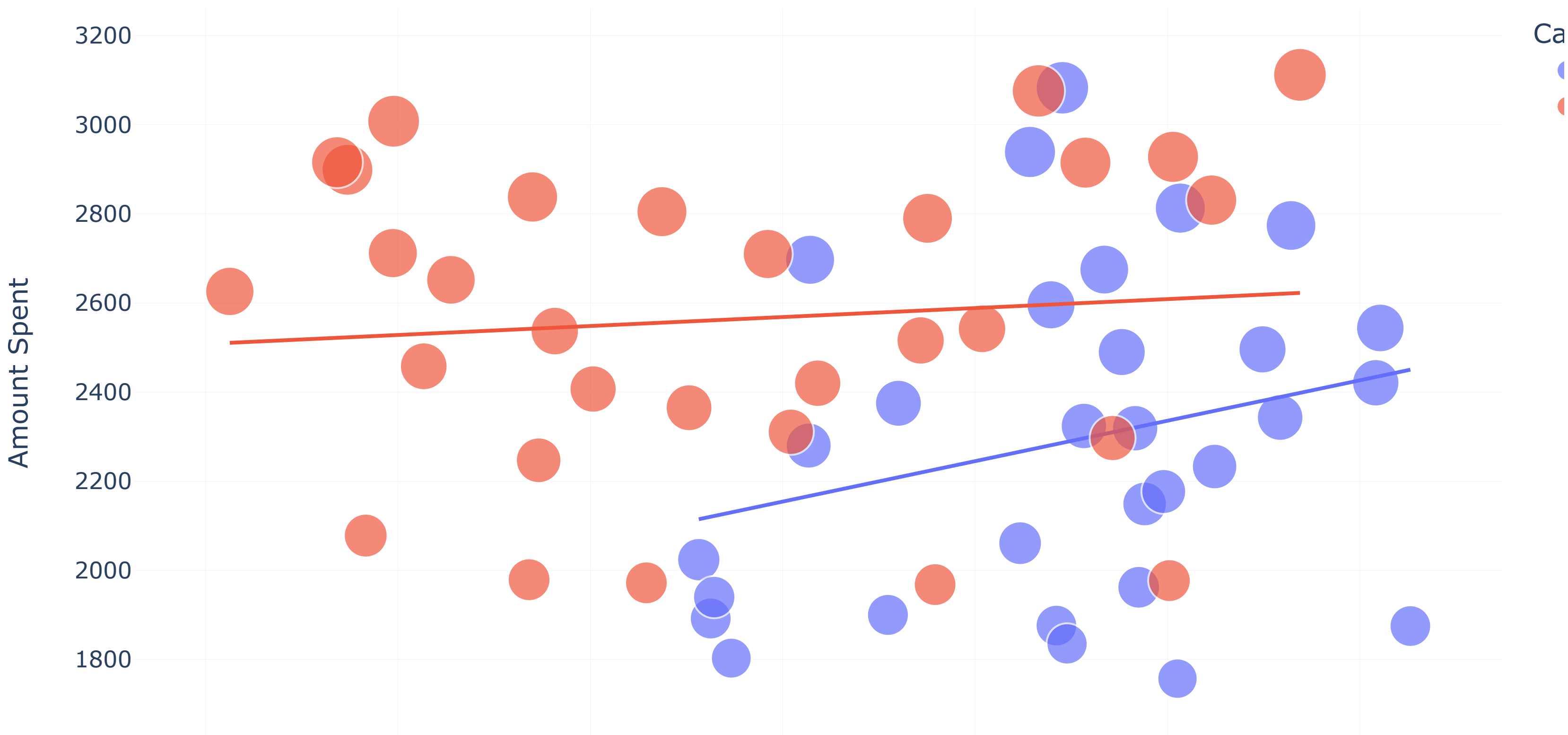
```
Control Campaign    30
Test Campaign       30
Name: Campaign Name, dtype: int64
```

The dataset has 30 samples for each campaign. Now let’s start with A/B testing to find the best marketing strategy.

A/B Testing to Find the Best Marketing Strategy

To get started with A/B testing, I will first analyze the relationship between the number of impressions we got from both campaigns and the amount spent on both campaigns:

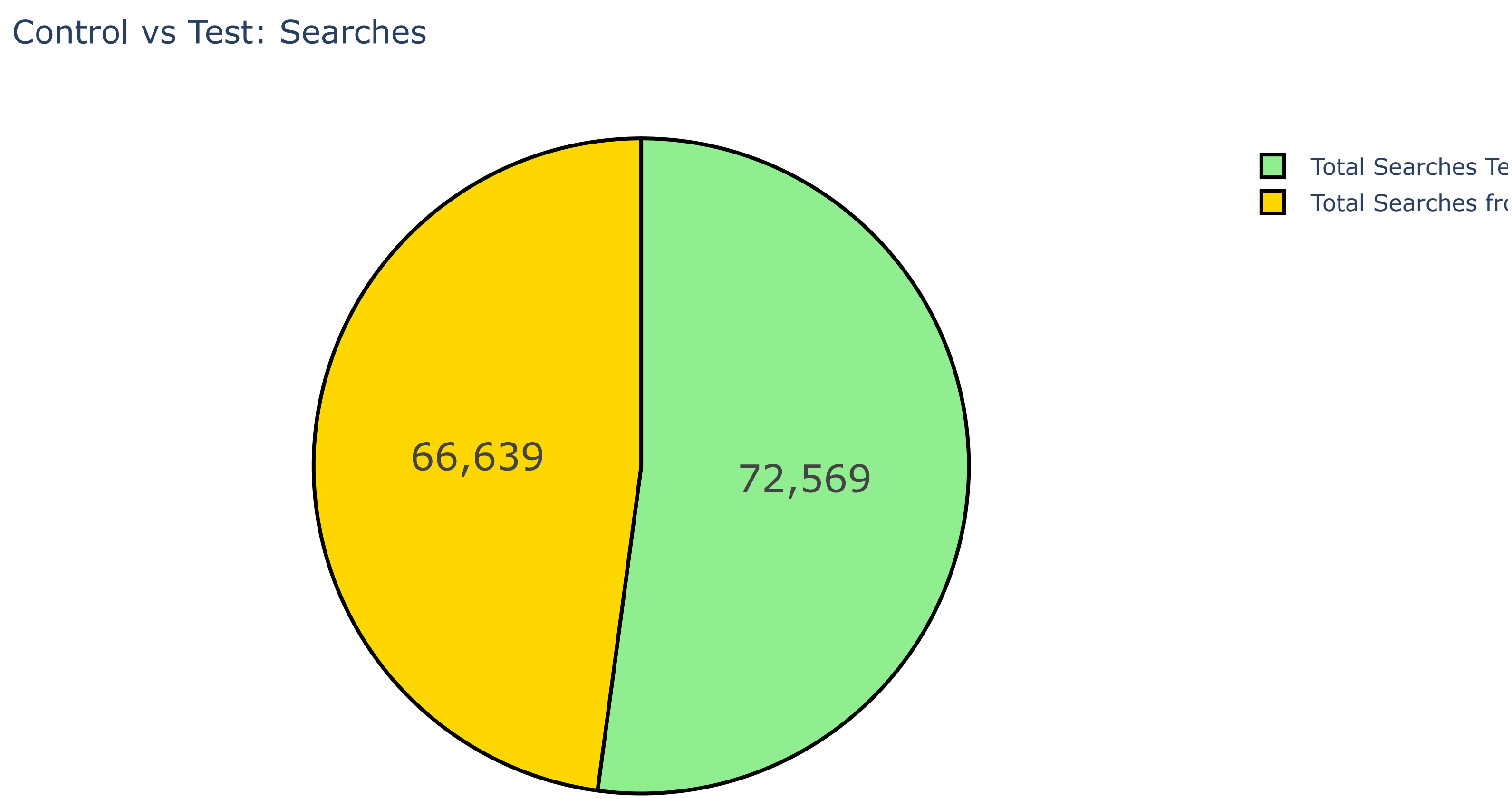
In [11]:
```
figure = px.scatter(data_frame=ab_data,
                    x = 'Number of Impressions',
                    y = 'Amount Spent',
                    size = 'Amount Spent',
                    color = 'Campaign Name',
                    trendline = 'ols')
figure.show()
```



The control campaign resulted in more impressions according to the amount spent on both campaigns. Now let’s have a look at the number of searches performed on the website from both campaigns

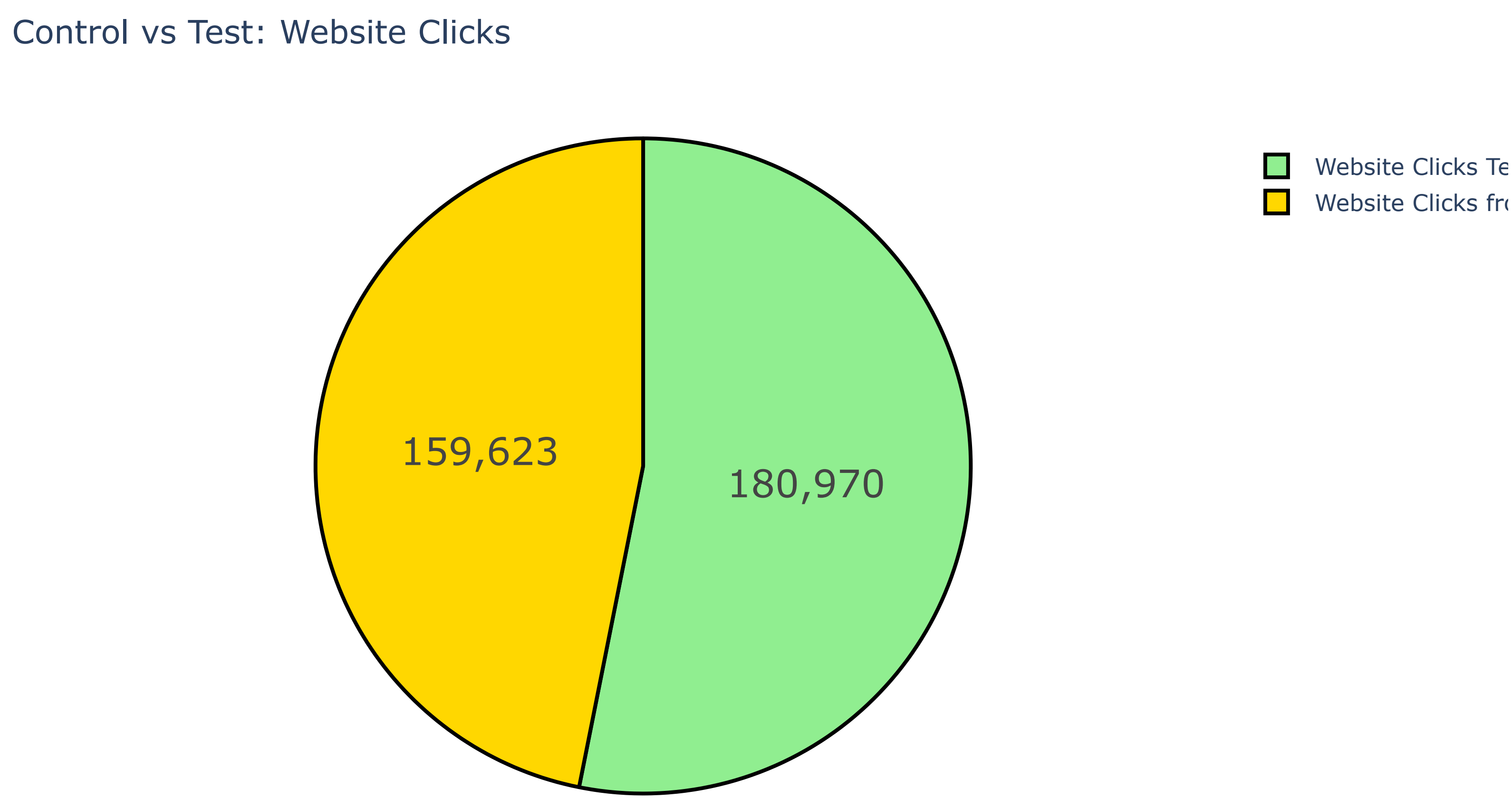
In [12]:

```
label = ['Total Searches from Control Campaign',
         'Total Searches Test Campaign']
Counts = [int(sum(control_data['Searches Received'])),
         sum(test_data['Searches Received'])]
colors = ['gold' , 'lightgreen']
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=20,
                 marker = dict(colors = colors, line =  dict(color = 'black',width = 2)))
fig.show()
```



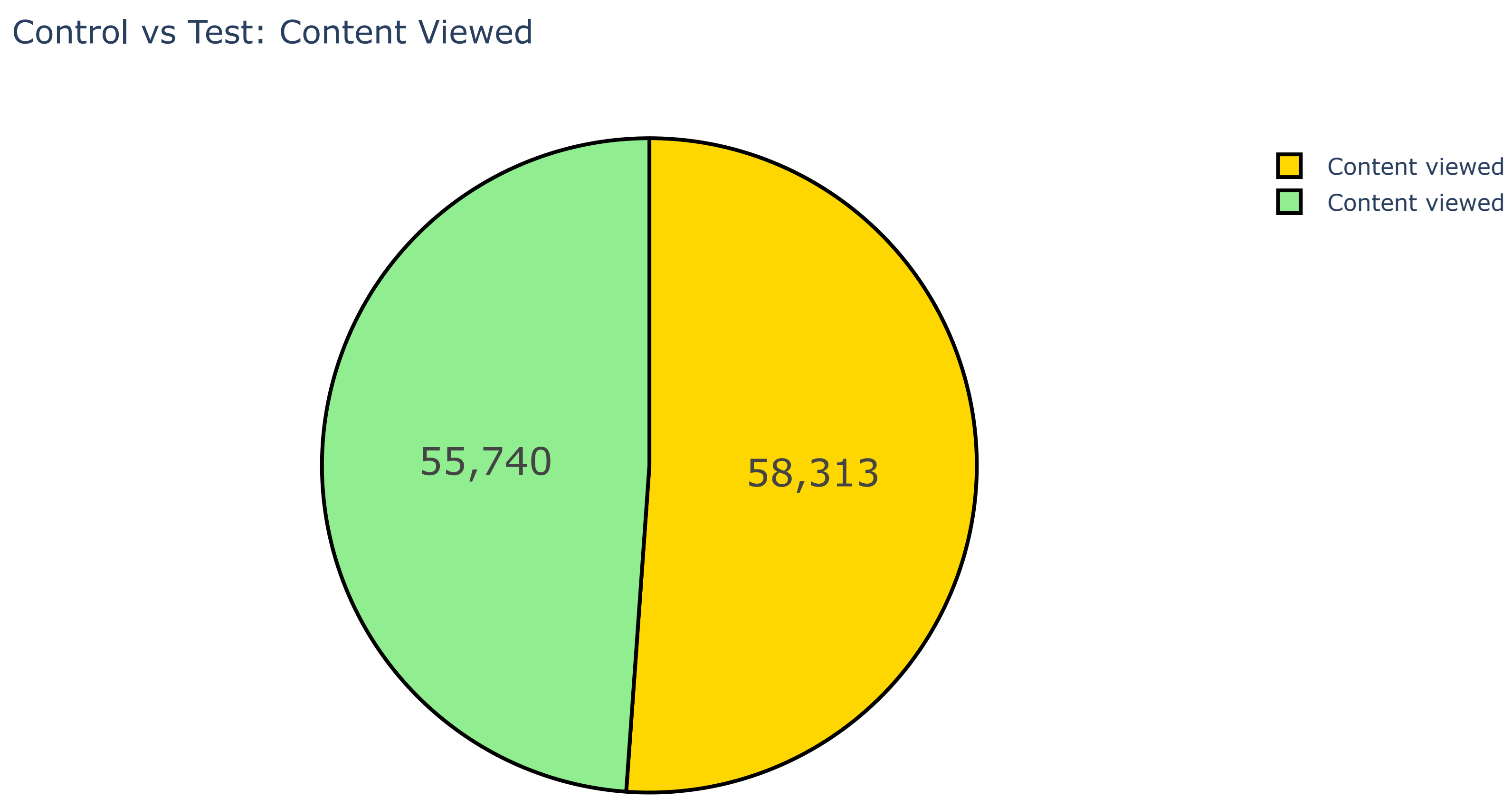
The test campaign resulted in more searches on the website. Now let’s have a look at the number of website clicks from both campaigns:

In [13]:
```
label = ['Website Clicks from Control Campaign',
         'Website Clicks Test Campaign']
Counts = [int(sum(control_data['Website Clicks'])),
         sum(test_data['Website Clicks'])]
colors = ['gold' , 'lightgreen']
fig = go.Figure(data = [go.Pie(labels = label, values = Counts)])
fig.update_layout(title_text = 'Control vs Test: Website Clicks')
fig.update_traces(hoverinfo = 'label+percent',
                 textinfo = 'value',
                 textfont_size=20,
                 marker = dict(colors = colors, line =  dict(color = 'black',width = 2)))
fig.show()
```



The test campaign wins in the number of website clicks. Now let’s have a look at the amount of content viewed after reaching the website from both campaigns:

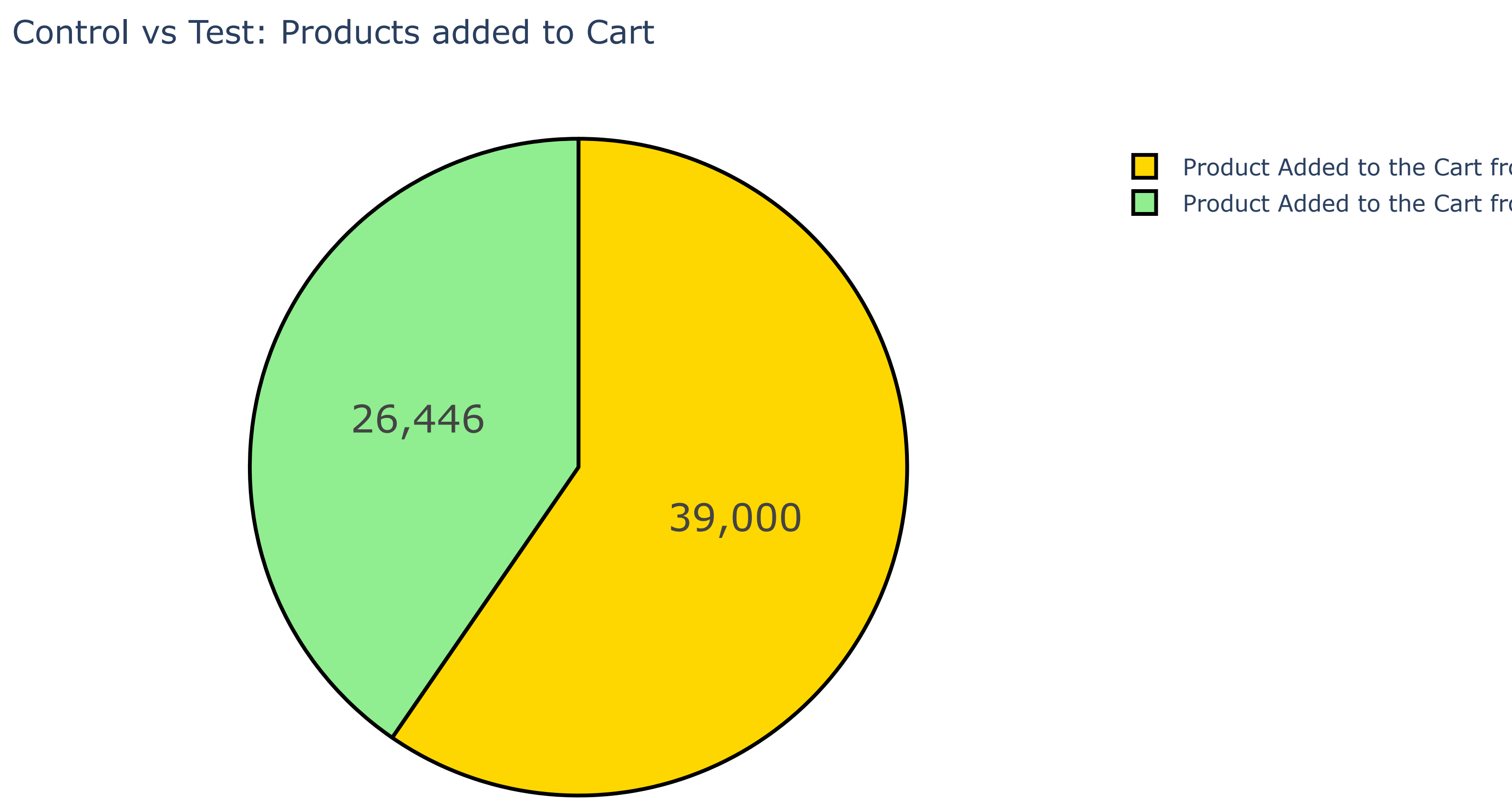
In [14]:
```
label = ['Content viewed by Control Campaign',
         'Content viewed by Test Campaign']
Counts = [int(sum(control_data['Content Viewed'])),
         sum(test_data['Content Viewed'])]
colors = ['gold' , 'lightgreen']
fig = go.Figure(data = [go.Pie(labels = label, values = Counts)])
fig.update_layout(title_text = 'Control vs Test: Content Viewed')
fig.update_traces(hoverinfo = 'label+percent',
                 textinfo = 'value',
                 textfont_size=20,
                 marker = dict(colors = colors, line =  dict(color = 'black',width = 2)))
fig.show()
```



The audience of the control campaign viewed more content than the test campaign. Although there is not much difference, as the website clicks of the control campaign were low, its engagement on the website is higher than the test campaign.

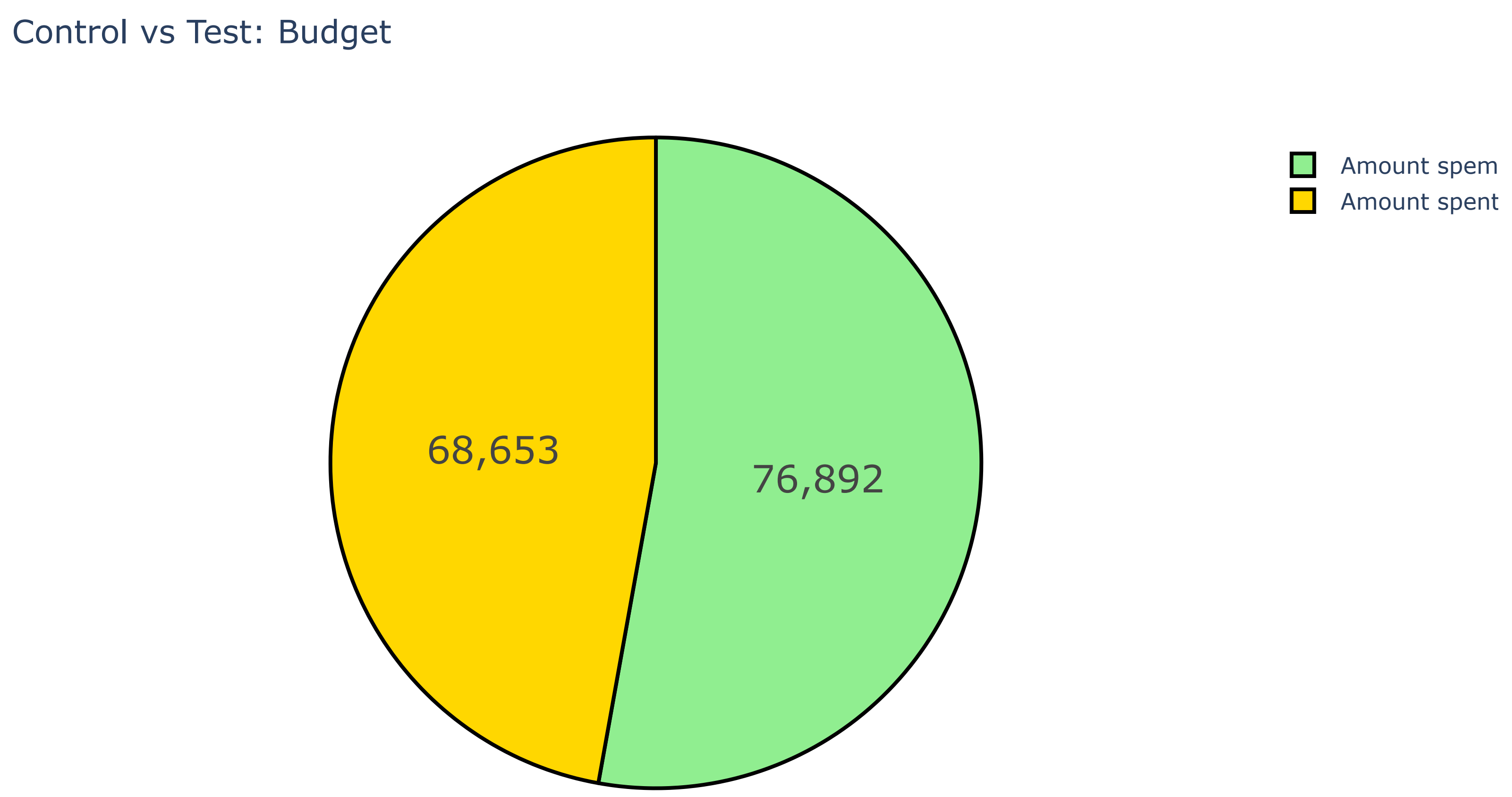
Now let’s have a look at the number of products added to the cart from both campaigns:

In [15]:
```
label = ['Product Added to the Cart from Control Campaign',
         'Product Added to the Cart from Test Campaign']
Counts = [sum(control_data['Added to Cart']),
         sum(test_data['Added to Cart'])]
colors = ['gold' , 'lightgreen']
fig = go.Figure(data = [go.Pie(labels = label, values = Counts)])
fig.update_layout(title_text = 'Control vs Test: Products added to Cart')
fig.update_traces(hoverinfo = 'label+percent',
                 textinfo = 'value',
                 textfont_size=20,
                 marker = dict(colors = colors, line =  dict(color = 'black',width = 2)))
fig.show()
```



Despite low website clicks more products were added to the cart from the control campaign. Now let’s have a look at the amount spent on both campaigns:

In [16]:
```
label = ['Amount spent in Control Campaign',
         'Amount spemt in Test Campaign']
Counts = [sum(control_data['Amount Spent']),
         sum(test_data['Amount Spent'])]
colors = ['gold' , 'lightgreen']
fig = go.Figure(data = [go.Pie(labels = label, values = Counts)])
fig.update_layout(title_text = 'Control vs Test: Budget')
fig.update_traces(hoverinfo = 'label+percent',
                 textinfo = 'value',
                 textfont_size=20,
                 marker = dict(colors = colors, line =  dict(color = 'black',width = 2)))
fig.show()
```

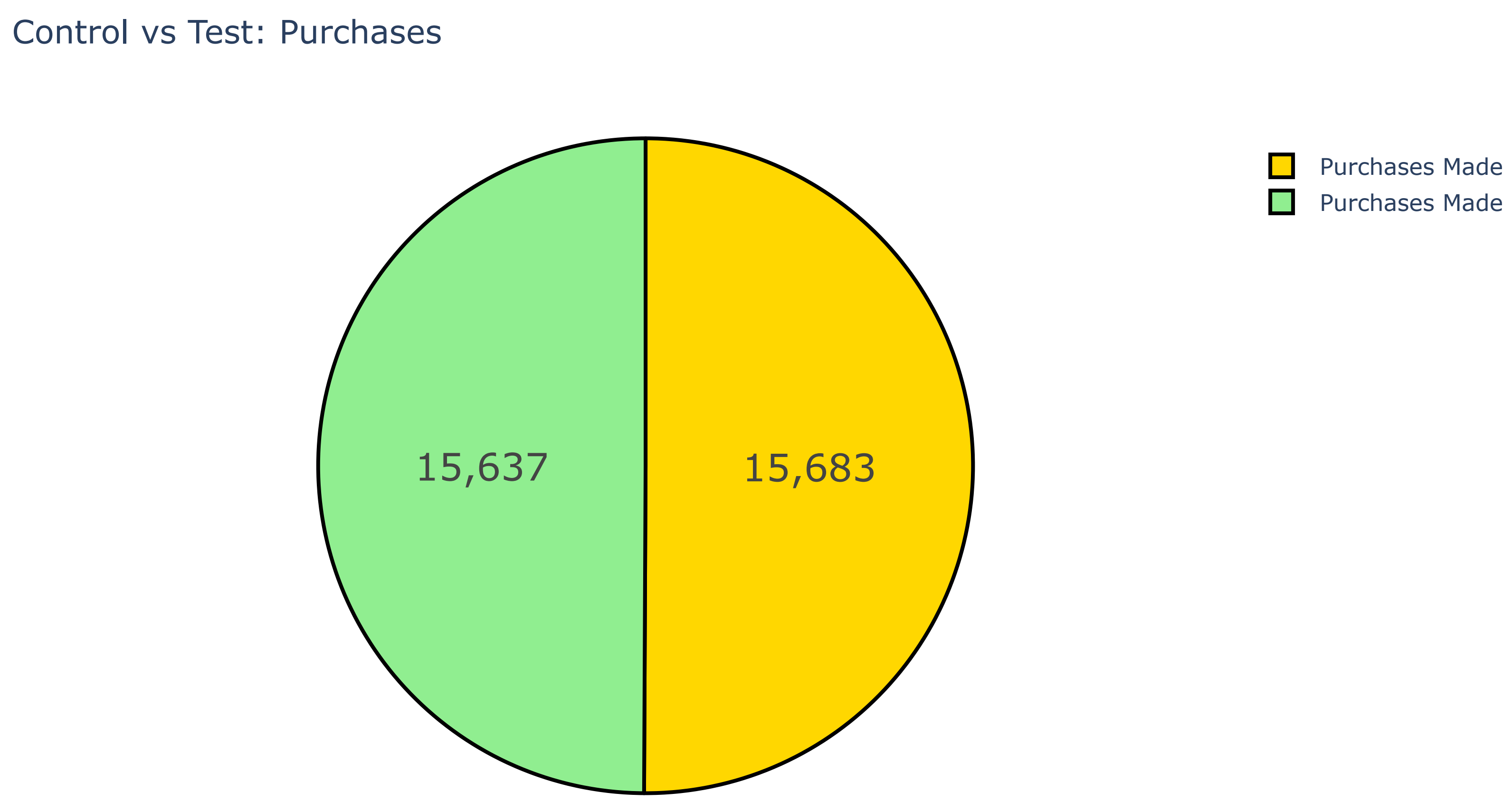


The amount spent on the test campaign is higher than the control campaign. But as we can see that the control campaign resulted in more content views and more products in the cart, the control campaign is more efficient than the test campaign.

Now let’s have a look at the purchases made by both campaigns

In [17]:

```
label = ['Purchases Made by Control Campaign',
         'Purchases Made by Test Campaign']
Counts = [int(sum(control_data['Purchases'])),
         sum(test_data['Purchases'])]
colors = ['gold' , 'lightgreen']
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=20,
                 marker = dict(colors = colors, line =  dict(color = 'black',width = 2)))
fig.show()
```

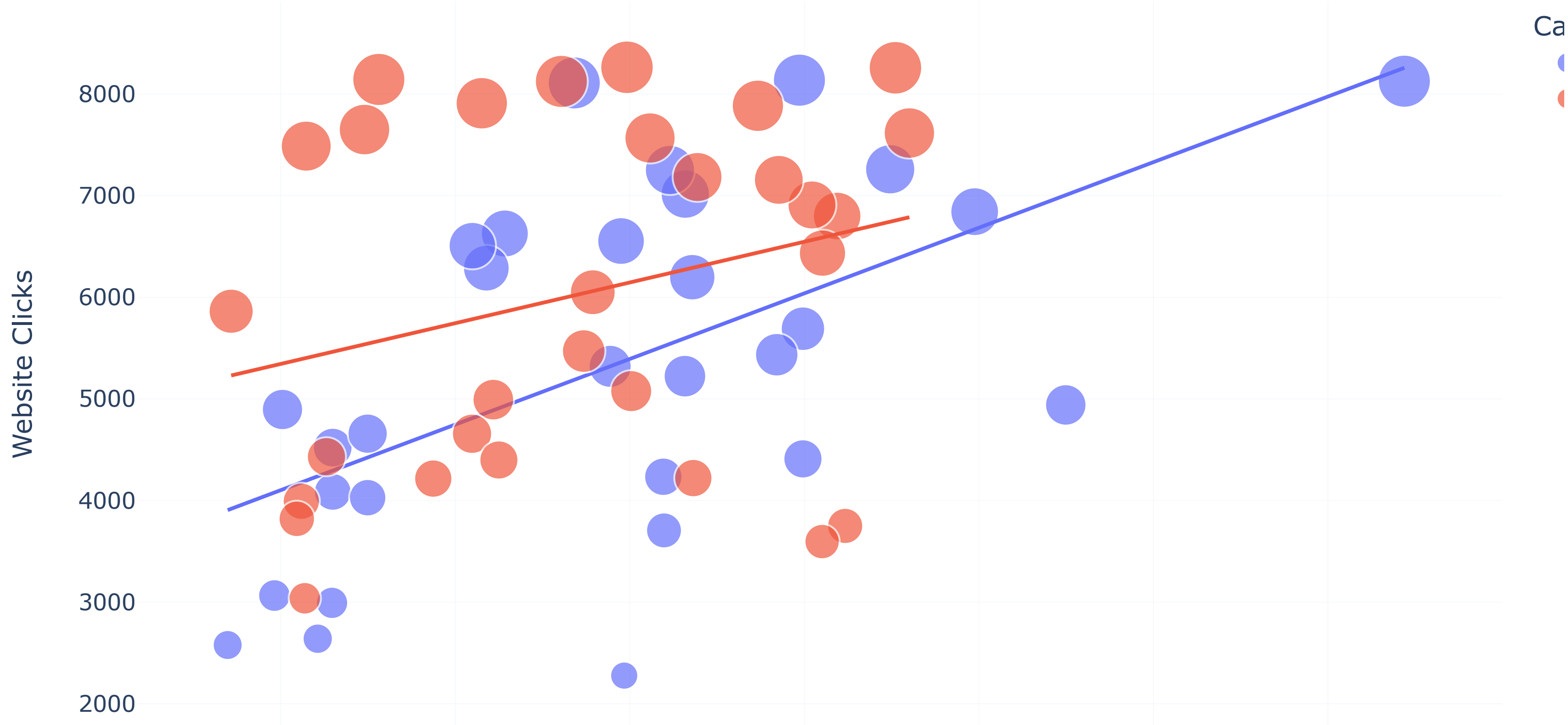


There’s only a difference of around 1% in the purchases made from both ad campaigns. As the Control campaign resulted in more sales in less amount spent on marketing, the control campaign wins here!

Now let’s analyze some metrics to find which ad campaign converts more. I will first look at the relationship between the number of website clicks and content viewed from both campaigns:

In [18]:

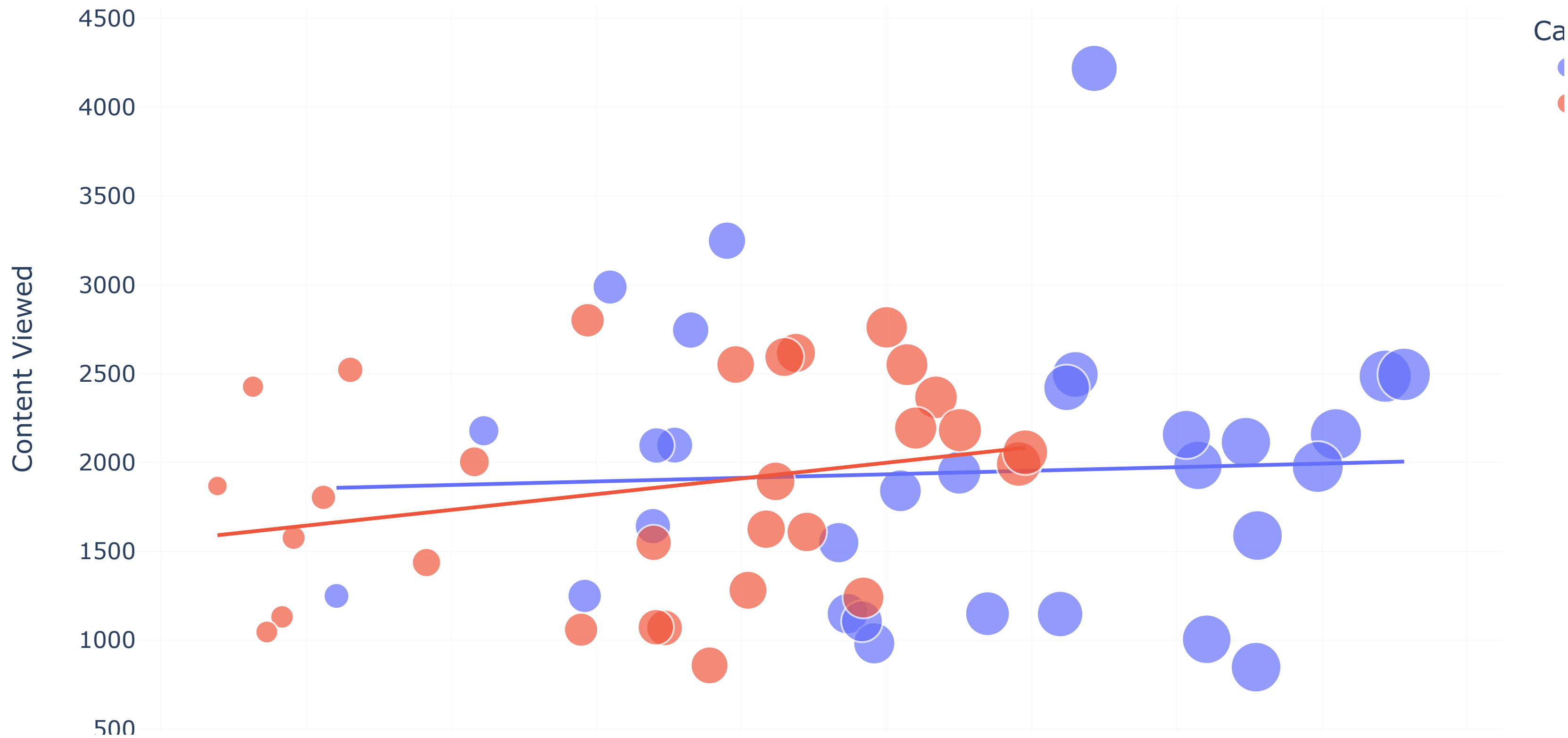
```
figure = px.scatter(data_frame=ab_data,
                    x = 'Content Viewed',
                    y = 'Website Clicks',
                    size = 'Website Clicks',
                    color = 'Campaign Name',
                    trendline = 'ols')
figure.show()
```



The website clicks are higher in the test campaign, but the engagement from website clicks is higher in the control campaign. So the control campaign wins!

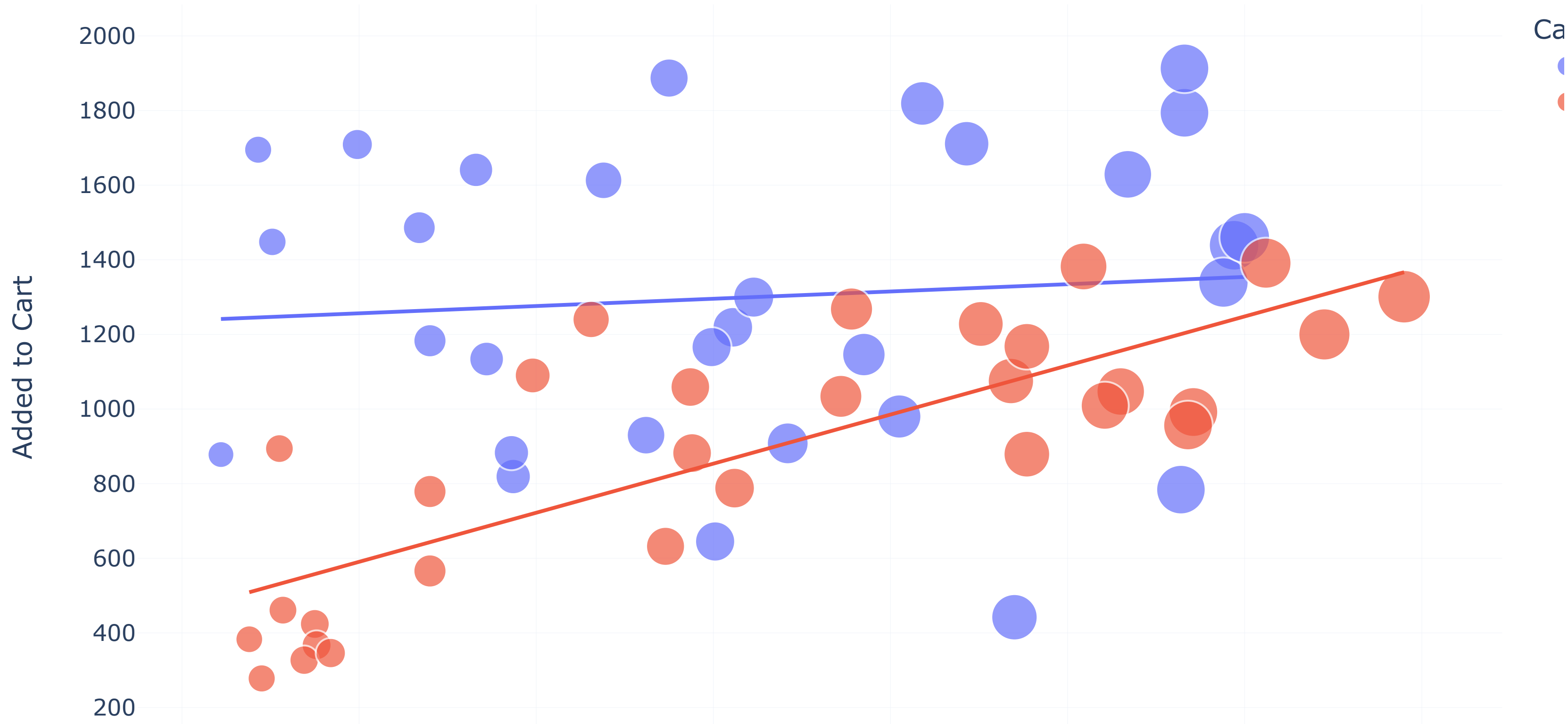
Now I will analyze the relationship between the amount of content viewed and the number of products added to the cart from both campaigns:

In [19]:
```
figure = px.scatter(data_frame=ab_data ,
                    x = 'Added to Cart',
                    y = 'Content Viewed',
                    size= 'Added to Cart',
                    color = 'Campaign Name',
                    trendline = 'ols')
figure.show()
```



Again, the control campaign wins! Now let's have a look at the relationship between the number of products added to the cart and the number of sales from both campaigns:

In [20]:
```python
figure = px.scatter(data_frame = ab_data,
                    x = 'Purchases',
                    y = 'Added to Cart',
                    size = 'Purchases',
                    color = 'Campaign Name',
                    trendline = 'ols')
figure.show()
```



Although the control campaign resulted in more sales and more products in the cart, the conversation rate of the test campaign is higher.

Conclusion

From the above A/B tests, we found that the control campaign resulted in more sales and engagement from the visitors. More products were viewed from the control campaign, resulting in more products in the cart and more sales. But the conversation rate of products in the cart is higher in the test campaign. The test campaign resulted in more sales according to the products viewed and added to the cart. And the control campaign results in more sales overall. So, the Test campaign can be used to market a specific product to a specific audience, and the Control campaign can be used to market multiple products to a wider audience.