Importing Libraries

In [53]:
```
import pandas as pd
import datetime
from datetime import date, timedelta
import plotly.graph_objects as go
import plotly.express as px
import plotly.io as pio
pio.templates.default = "plotly_white"
```

A/B testing is used for experimenting on a new marketing strategy, a new design or even a new game. To make an experiment, two different strategy is tried on two seperate group. One group is called test group and the experimental version is shown to them. Then, the remaining users are called control group and they see the default vesion.

According to the results, the better performing version is selected. To decide which version performed better, data analysis can be used to determine.

Data Preprocessing

In [54]:
```
control_group = pd.read_csv("control_group.csv", sep = ";")
test_group = pd.read_csv("test_group.csv", sep = ";")
```

In [55]:
```
control_group.head()
```

Out[55]:

	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 [56]:
```
test_group.head()
```

Out[56]:

	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

As it is seen from the dataframes, the column names are a little bit confusing. They will be replaced with a clearer headings.

In [57]:
```
control_group.columns = ["Campaign Name", "Date", "Amount Spent",
                         "Number of Impressions", "Reach", "Website Clicks",
                         "Searches Received", "Content Viewed", "Added to Cart",
                         "Purchases"]

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

In [58]:
```
control_group.head(10)
```

Out[58]:

	Campaign Name	Date	Amount Spent	Number of Impressions	Reach	Website Clicks	Searches Received	Content Viewed	Added to Cart	Purchases
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
5	Control Campaign	6.08.2019	3083	109076.0	87998.0	4028.0	1709.0	1249.0	784.0	764.0
6	Control Campaign	7.08.2019	2544	142123.0	127852.0	2640.0	1388.0	1106.0	1166.0	499.0
7	Control Campaign	8.08.2019	1900	90939.0	65217.0	7260.0	3047.0	2746.0	930.0	462.0
8	Control Campaign	9.08.2019	2813	121332.0	94896.0	6198.0	2487.0	2179.0	645.0	501.0
9	Control Campaign	10.08.2019	2149	117624.0	91257.0	2277.0	2475.0	1984.0	1629.0	734.0

In [59]:
```
control_group.describe()
```

Out[59]:

	Amount Spent	Number of Impressions	Reach	Website Clicks	Searches Received	Content Viewed	Added to Cart	Purchases
count	30.000000	29.000000	29.000000	29.000000	29.000000	29.000000	29.000000	29.000000
mean	2288.433333	109559.758621	88844.931034	5320.793103	2221.310345	1943.793103	1300.000000	522.793103
std	367.334451	21688.922908	21832.349595	1757.369003	866.089368	777.545469	407.457973	185.028642
min	1757.000000	71274.000000	42859.000000	2277.000000	1001.000000	848.000000	442.000000	222.000000
25%	1945.500000	92029.000000	74192.000000	4085.000000	1615.000000	1249.000000	930.000000	372.000000
50%	2299.500000	113430.000000	91579.000000	5224.000000	2390.000000	1984.000000	1339.000000	501.000000
75%	2532.000000	121332.000000	102479.000000	6628.000000	2711.000000	2421.000000	1641.000000	670.000000
max	3083.000000	145248.000000	127852.000000	8137.000000	4891.000000	4219.000000	1913.000000	800.000000

In [60]:
```
test_group.head(10)
```

Out[60]:

	Campaign Name	Date	Amount Spent	Number of Impressions	Reach	Website Clicks	Searches Received	Content Viewed	Added to Cart	Purchases
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
5	Test Campaign	6.08.2019	2458	42684	31489	7488	1854	1073	882	488
6	Test Campaign	7.08.2019	2838	53986	42148	4221	2733	2182	1301	890
7	Test Campaign	8.08.2019	2916	33669	20149	7184	2867	2194	1240	431
8	Test Campaign	9.08.2019	2652	45511	31598	8259	2899	2761	1200	845
9	Test Campaign	10.08.2019	2790	95054	79632	8125	2312	1804	424	275

In [61]:
```
test_group.describe()
```

Out[61]:

	Amount Spent	Number of Impressions	Reach	Website Clicks	Searches Received	Content Viewed	Added to Cart	Purchases
count	30.000000	30.000000	30.000000	30.000000	30.000000	30.000000	30.000000	30.000000
mean	2563.066667	74584.800000	53491.566667	6032.333333	2418.966667	1858.000000	881.533333	521.233333
std	348.687681	32121.377422	28795.775752	1708.567263	388.742312	597.654669	347.584248	211.047745
min	1968.000000	22521.000000	10598.000000	3038.000000	1854.000000	858.000000	278.000000	238.000000
25%	2324.500000	47541.250000	31516.250000	4407.000000	2043.000000	1320.000000	582.500000	298.000000
50%	2584.000000	68853.500000	44219.500000	6242.500000	2395.500000	1881.000000	974.000000	500.000000
75%	2836.250000	99500.000000	78778.750000	7604.750000	2801.250000	2412.000000	1148.500000	701.000000
max	3112.000000	133771.000000	109834.000000	8264.000000	2978.000000	2801.000000	1391.000000	890.000000

The number of tests for both control group and test group are hopefully equal. Let's check the emty values.

In [62]:
```
control_group.isnull().sum()
```

Out[62]:
```
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 [63]:
```
test_group.isnull().sum()
```

Out[63]:
```
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
```

In [64]:
```
control_group[control_group.isna().any(axis=1)]
```

Out[64]:

	Campaign Name	Date	Amount Spent	Number of Impressions	Reach	Website Clicks	Searches Received	Content Viewed	Added to Cart	Purchases
4	Control Campaign	5.08.2019	1835	NaN	NaN	NaN	NaN	NaN	NaN	NaN

One row value is missing in our control group dataframe. We can drop it as its effect will not be too much on the whole data. However, we should delete the corresponding row from the test data to compare both accurately.

In [65]:
```
control_group.dropna(how='any', inplace=True)
```

In [66]:
```
control_group.describe()
```

Out[66]:

	Amount Spent	Number of Impressions	Reach	Website Clicks	Searches Received	Content Viewed	Added to Cart	Purchases
count	29.000000	29.000000	29.000000	29.000000	29.000000	29.000000	29.000000	29.000000
mean	2304.068966	109559.758621	88844.931034	5320.793103	2221.310345	1943.793103	1300.000000	522.793103
std	363.534822	21688.922908	21832.349595	1757.369003	866.089368	777.545469	407.457973	185.028642
min	1757.000000	71274.000000	42859.000000	2277.000000	1001.000000	848.000000	442.000000	222.000000
25%	1962.000000	92029.000000	74192.000000	4085.000000	1615.000000	1249.000000	930.000000	372.000000
50%	2319.000000	113430.000000	91579.000000	5224.000000	2390.000000	1984.000000	1339.000000	501.000000
75%	2544.000000	121332.000000	102479.000000	6628.000000	2711.000000	2421.000000	1641.000000	670.000000
max	3083.000000	145248.000000	127852.000000	8137.000000	4891.000000	4219.000000	1913.000000	800.000000

Let's locate the corresponding row from the test data and delete it.

In [67]:
```
test_group.loc[test_group['Date'] == '5.08.2019']
```

Out[67]:

	Campaign Name	Date	Amount Spent	Number of Impressions	Reach	Website Clicks	Searches Received	Content Viewed	Added to Cart	Purchases
4	Test Campaign	5.08.2019	2297	114295	95138	5863	2106	858	956	768

In [68]:
```
test_group = test_group.drop(4)
```

In [69]:
```
test_group.describe()
```

Out[69]:

	Amount Spent	Number of Impressions	Reach	Website Clicks	Searches Received	Content Viewed	Added to Cart	Purchases
count	29.000000	29.000000	29.000000	29.000000	29.000000	29.000000	29.000000	29.000000
mean	2572.241379	73215.482759	52055.482759	6038.172414	2429.758621	1892.482759	878.965517	512.724138
std	351.155100	31786.355952	28190.975729	1738.505086	391.022986	577.063355	353.446953	209.480633
min	1968.000000	22521.000000	10598.000000	3038.000000	1854.000000	1046.000000	278.000000	238.000000
25%	2365.000000	45511.000000	31489.000000	4399.000000	2037.000000	1437.000000	566.000000	284.000000
50%	2626.000000	67444.000000	43241.000000	6435.000000	2432.000000	1894.000000	992.000000	488.000000
75%	2838.000000	95843.000000	76219.000000	7617.000000	2824.000000	2427.000000	1168.000000	677.000000
max	3112.000000	133771.000000	109834.000000	8264.000000	2978.000000	2801.000000	1391.000000	890.000000

It is time to merge the two campaigns to compare the results easily.

In [70]:
```
ab_data = control_group.merge(test_group,
                              how="outer").sort_values(["Date"])
ab_data = ab_data.reset_index(drop=True)
ab_data.head(10)
```

Out[70]:

	Campaign Name	Date	Amount Spent	Number of Impressions	Reach	Website Clicks	Searches Received	Content Viewed	Added to Cart	Purchases
0	Control Campaign	1.08.2019	2280	82702.0	56930.0	7016.0	2290.0	2159.0	1819.0	618.0
1	Test Campaign	1.08.2019	3008	39550.0	35820.0	3038.0	1946.0	1069.0	894.0	255.0
2	Test Campaign	10.08.2019	2790	95054.0	79632.0	8125.0	2312.0	1804.0	424.0	275.0
3	Control Campaign	10.08.2019	2149	117624.0	91257.0	2277.0	2475.0	1984.0	1629.0	734.0
4	Test Campaign	11.08.2019	2420	83633.0	71286.0	3750.0	2893.0	2617.0	1075.0	668.0
5	Control Campaign	11.08.2019	2490	115247.0	95843.0	8137.0	2941.0	2486.0	1887.0	475.0
6	Test Campaign	12.08.2019	2831	124591.0	10598.0	8264.0	2081.0	1992.0	1382.0	709.0
7	Control Campaign	12.08.2019	2319	116639.0	100189.0	2993.0	1397.0	1147.0	1439.0	794.0
8	Test Campaign	13.08.2019	1972	65827.0	49531.0	7568.0	2213.0	2058.0	1391.0	812.0
9	Control Campaign	13.08.2019	2697	82847.0	68214.0	6554.0	2390.0	1975.0	1794.0	766.0

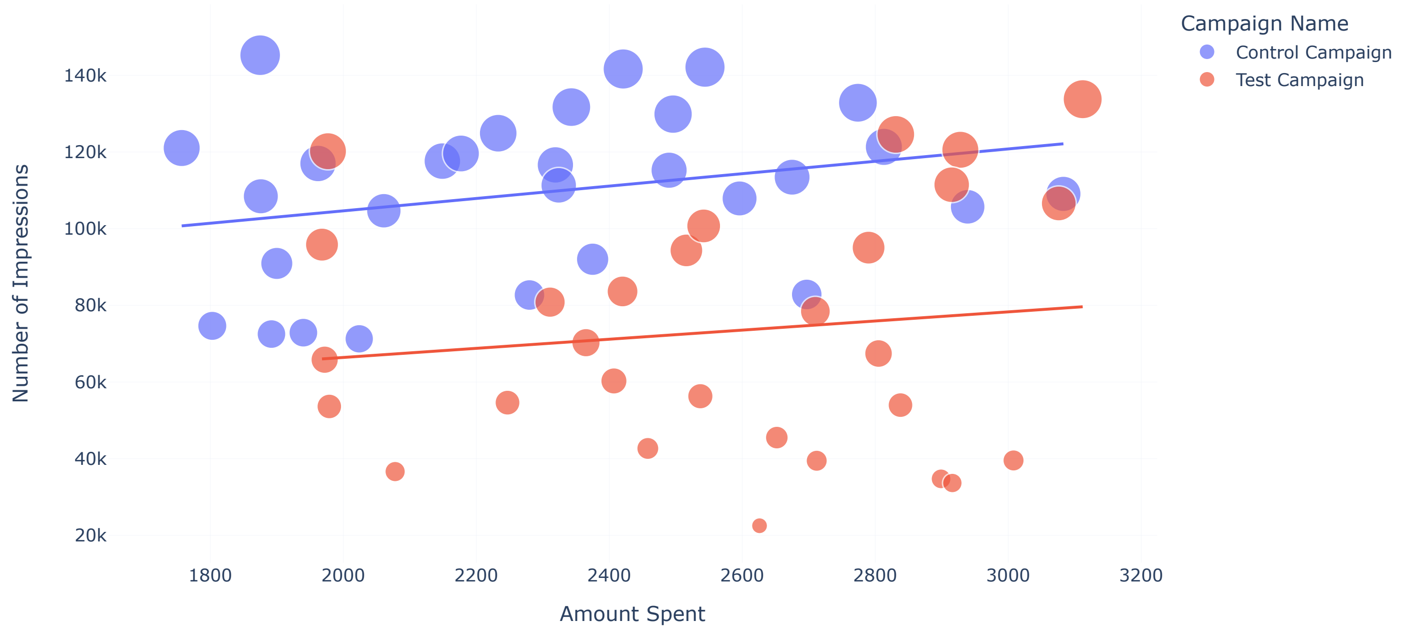
In [71]:
```
ab_data["Campaign Name"].value_counts()
```

Out[71]:
```
Control Campaign    29
Test Campaign       29
Name: Campaign Name, dtype: int64
```

Great, now in our A/B test data, there are 29 test cases for a control group and test group.

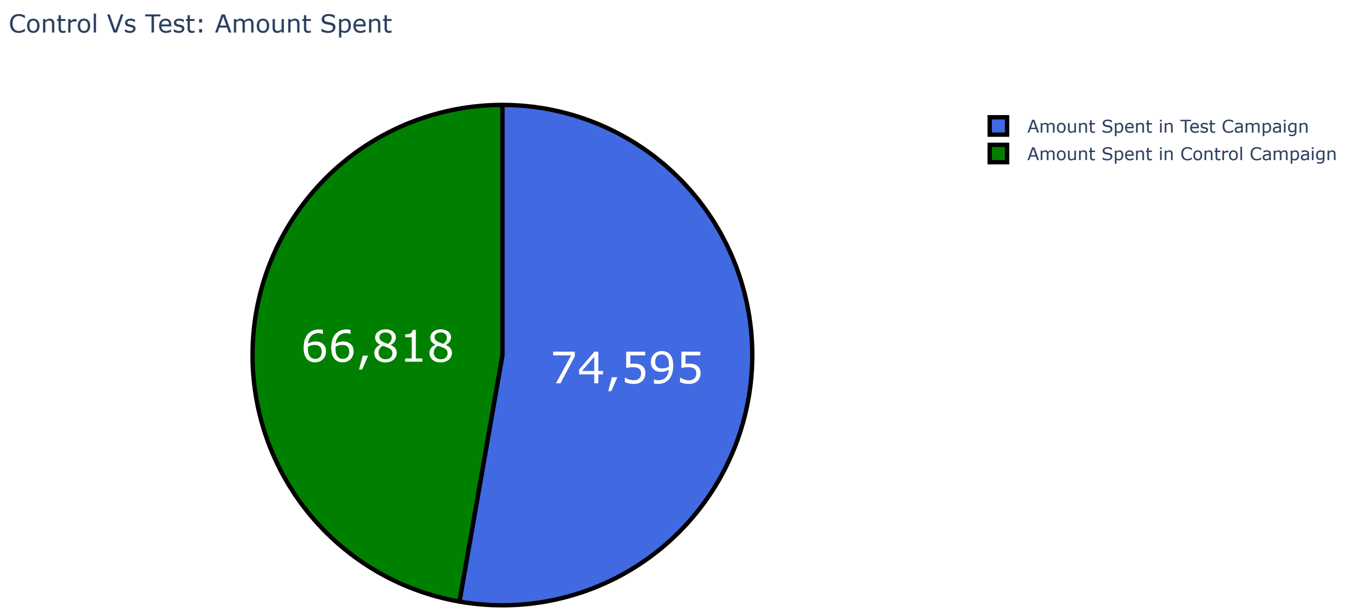
Data Visualization

In [72]:
```python
figure = px.scatter(data_frame = ab_data,
                    x="Amount Spent",
                    y="Number of Impressions",
                    size="Number of Impressions",
                    color= "Campaign Name",
                    trendline="ols")
figure.show("notebook")
```



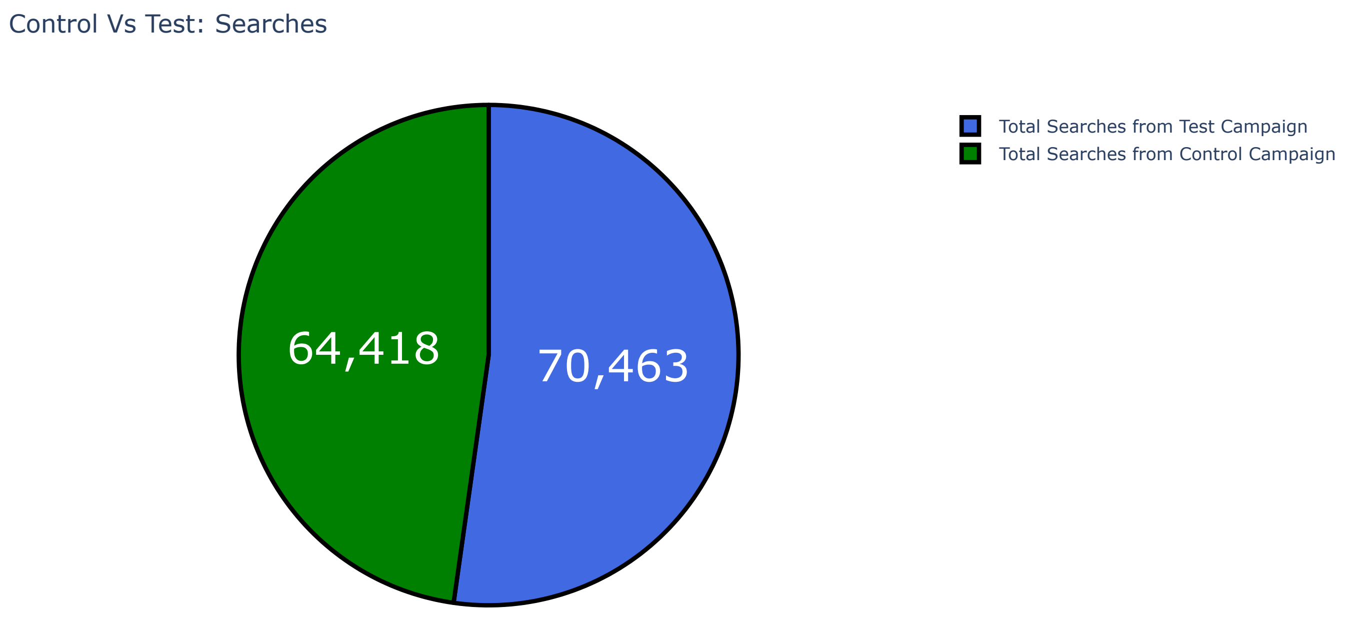
Control campaign worked better for the number of impression per amount spent during the campaign.

```
In [73]: label = ["Amount Spent in Control Campaign",
                  "Amount Spent in Test Campaign"]
         counts = [sum(control_group["Amount Spent"]),
                   sum(test_group["Amount Spent"])]
         colors = ['green','royalblue']
         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("notebook", width=400, height=300)
```



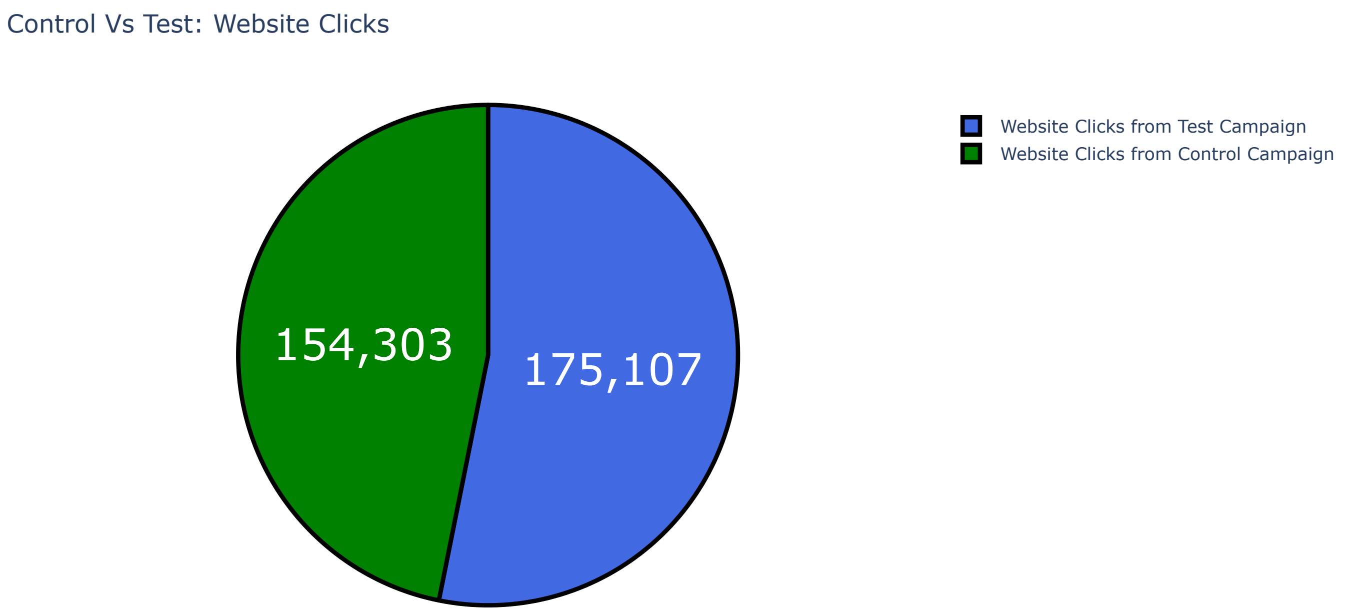
More amount of money spent for the test campaign. Thus, we expect more results from the test campaign.

```
In [74]: label = ["Total Searches from Control Campaign",
                  "Total Searches from Test Campaign"]
         counts = [sum(control_group["Searches Received"]),
                   sum(test_group["Searches Received"])]
         colors = ['green','royalblue']
         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("notebook", width=400, height=300)
```



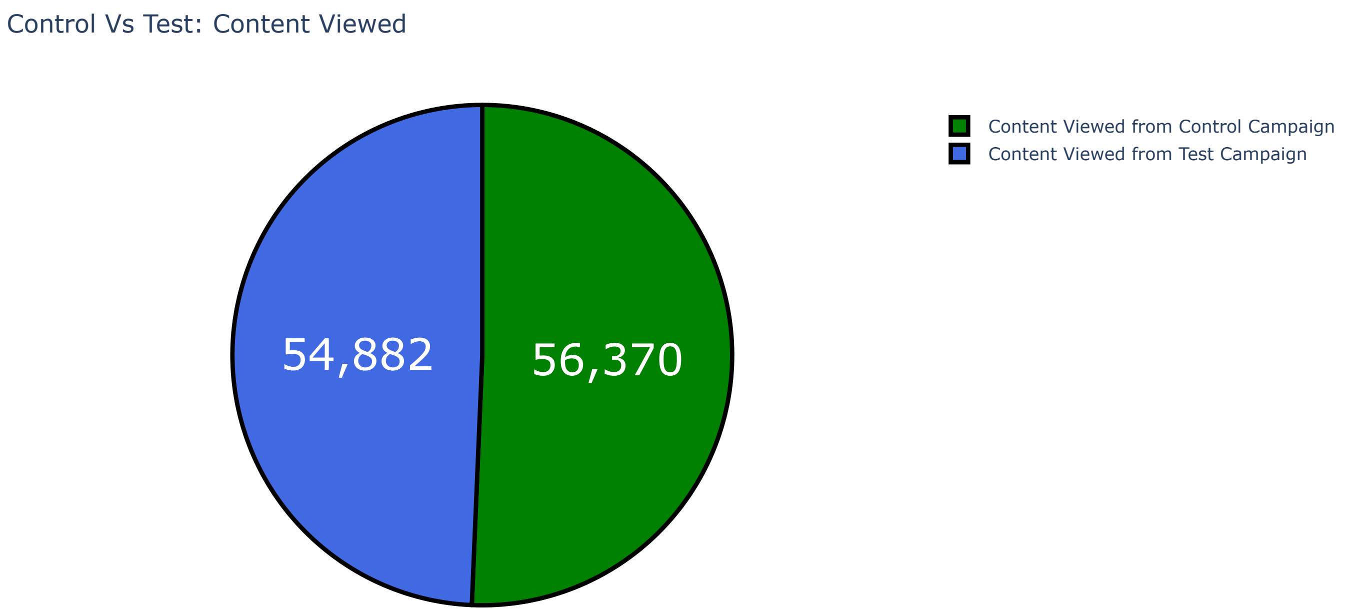
Test campaign got more searches.

```
In [75]: label = ["Website Clicks from Control Campaign",
                  "Website Clicks from Test Campaign"]
         counts = [sum(control_group["Website Clicks"]),
                   sum(test_group["Website Clicks"])]
         colors = ['green','royalblue']
         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=30,
                           marker=dict(colors=colors,
                                       line=dict(color='black', width=3)))
         fig.show("notebook", width=400, height=300)
```



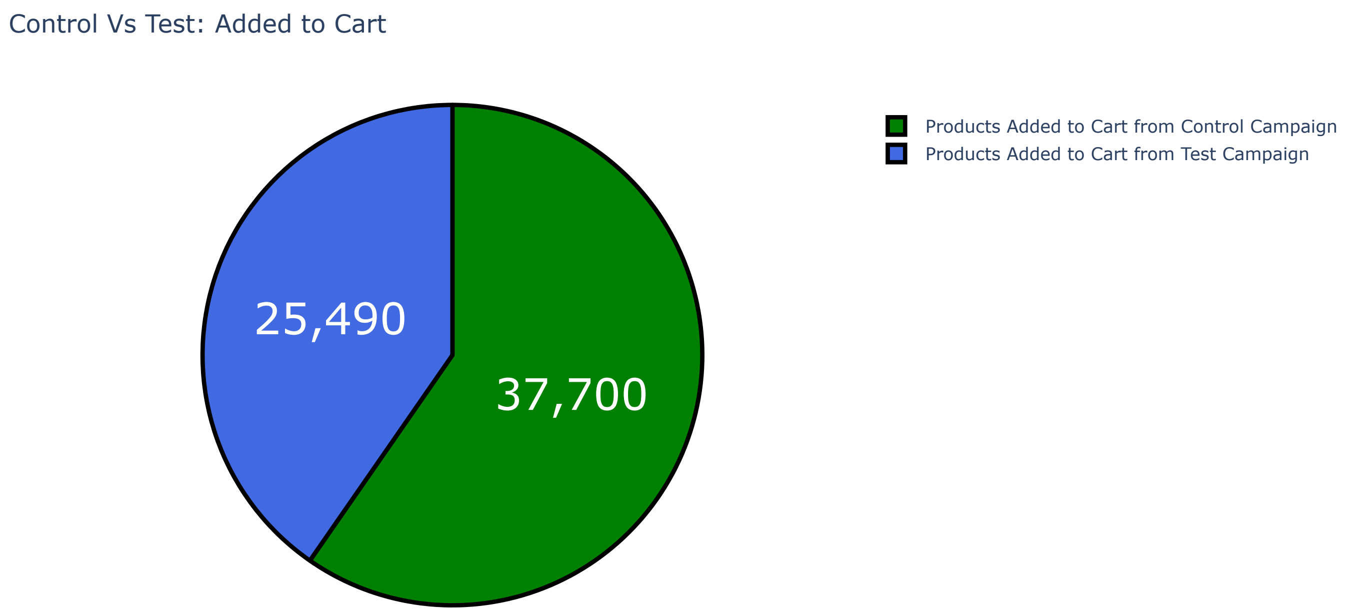
Test campaign got more website click.

```
In [76]: label = ["Content Viewed from Control Campaign",
                  "Content Viewed from Test Campaign"]
         counts = [sum(control_group["Content Viewed"]),
                   sum(test_group["Content Viewed"])]
         colors = ['green','royalblue']
         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("notebook", width=400, height=300)
```



Control campaign resulted in more content viewed by customers.

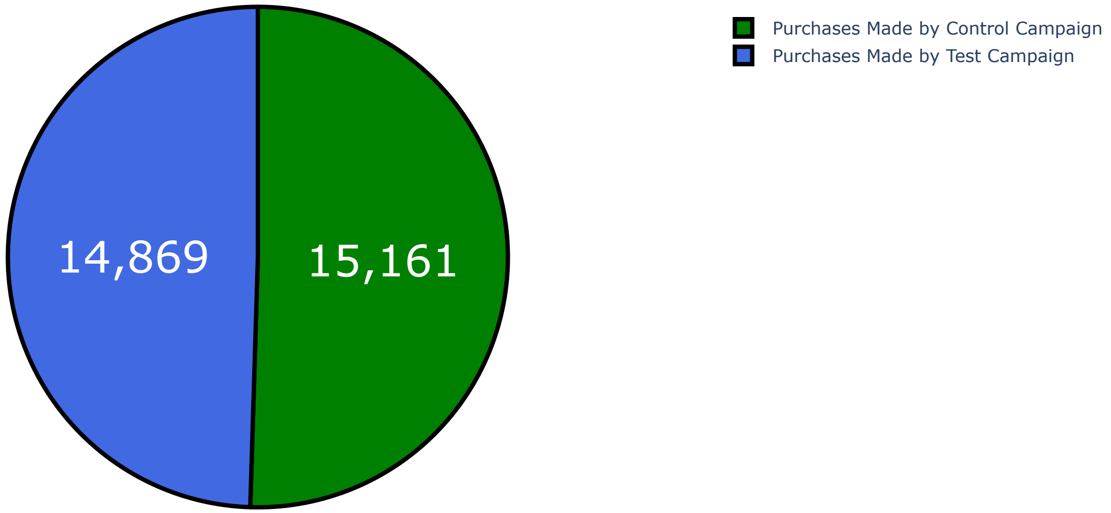
```
In [77]: label = ["Products Added to Cart from Control Campaign",
                  "Products Added to Cart from Test Campaign"]
         counts = [sum(control_group["Added to Cart"]),
                   sum(test_group["Added to Cart"])]
         colors = ['green','royalblue']
         fig = go.Figure(data=[go.Pie(labels=label, 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("notebook", width=400, height=300)
```



Customers which are in control group added more products to their charts.

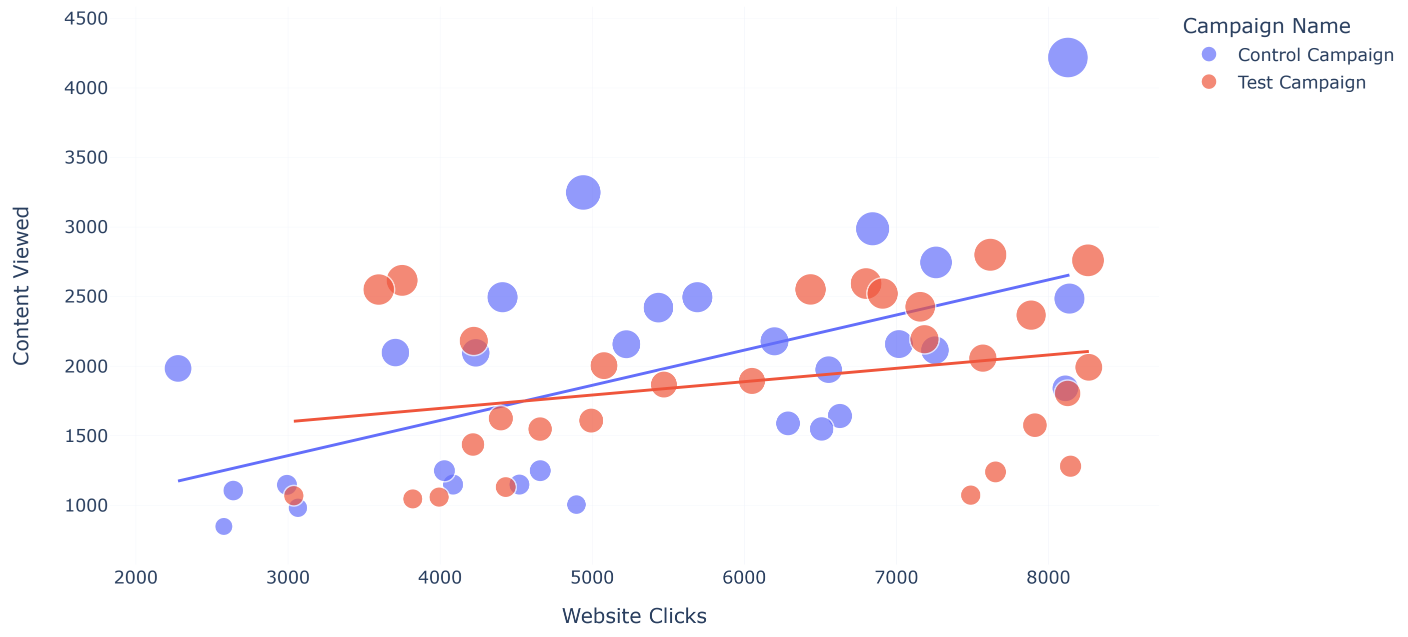
```
In [78]: label = ["Purchases Made by Control Campaign",
                  "Purchases Made by Test Campaign"]
         counts = [sum(control_group["Purchases"]),
                   sum(test_group["Purchases"])]
         colors = ['green','royalblue']
         fig = go.Figure(data=[go.Pie(labels=label, 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("notebook", width=400, height=300)
```

Control Vs Test: Purchases



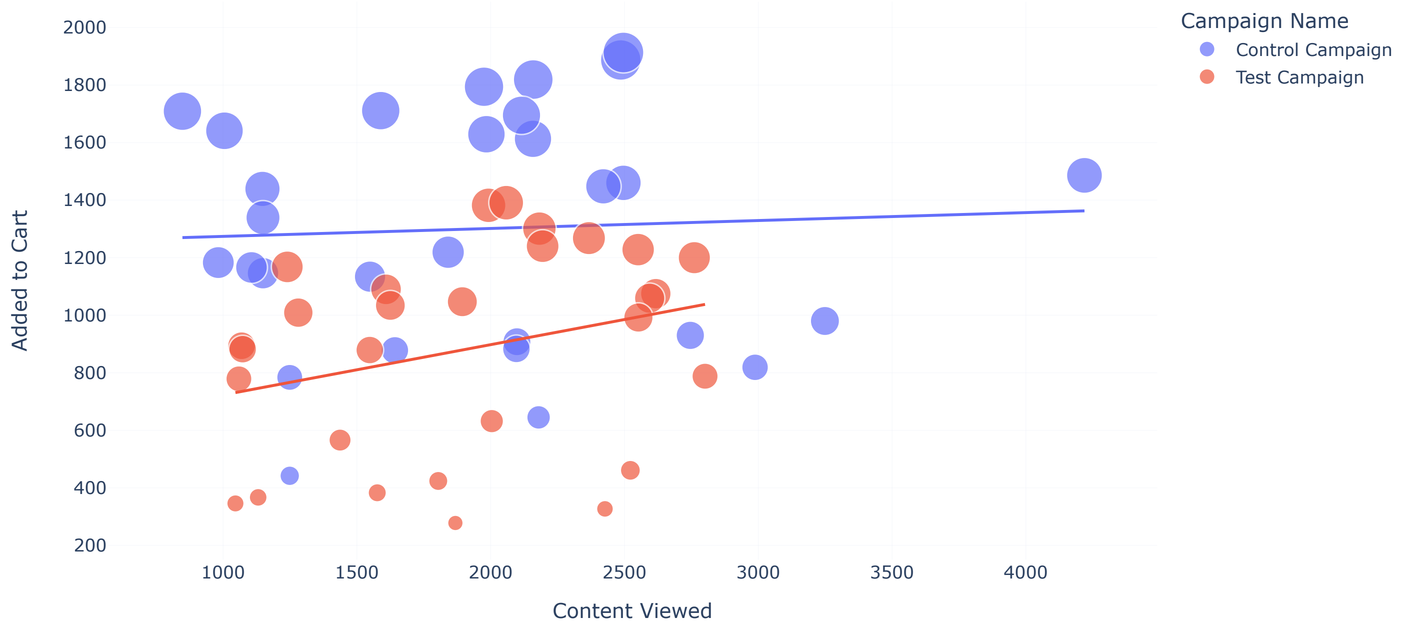
Customers in control group purchased sligtly more than customers in test group, eventhough, customers in the control group added way more products to their chart.

```
In [79]: figure = px.scatter(data_frame = ab_data,
                             x="Website Clicks",
                             y="Content Viewed",
                             size="Content Viewed",
                             color= "Campaign Name",
                             trendline="ols")
         figure.show("notebook")
```



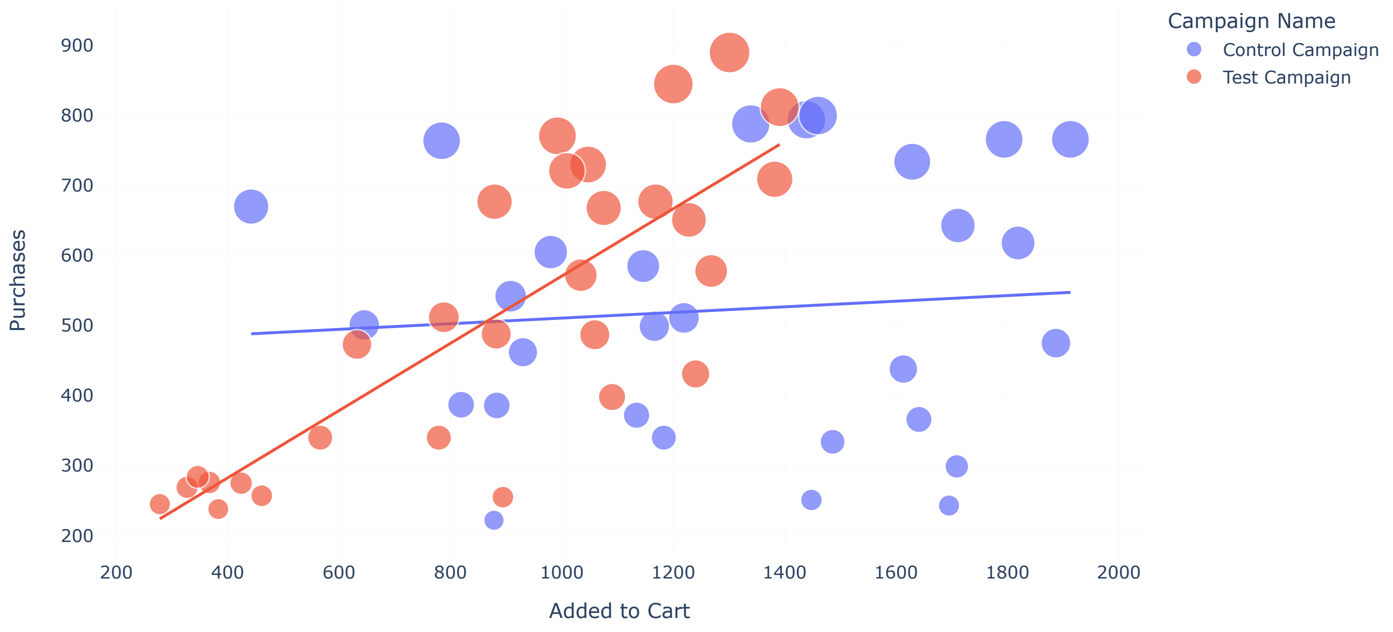
Control campaign contributed to more content views per website clicks.

```
In [80]: figure = px.scatter(data_frame = ab_data,
                             x="Content Viewed",
                             y="Added to Cart",
                             size="Added to Cart",
                             color= "Campaign Name",
                             trendline="ols")
         figure.show("notebook")
```



Customers in test group added more products to their charts respect to the content views.

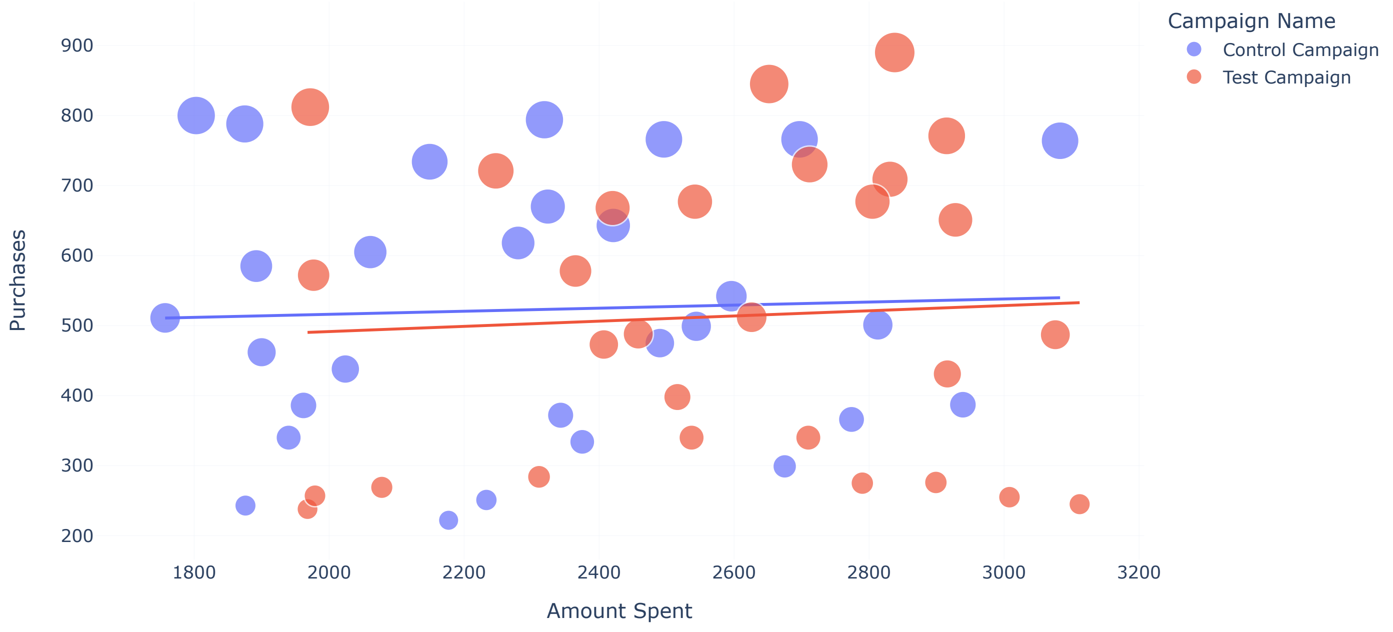
```
In [81]: figure = px.scatter(data_frame = ab_data,
                             x="Added to Cart",
                             y="Purchases",
                             size="Purchases",
                             color= "Campaign Name",
                             trendline="ols")
         figure.show("notebook")
```



Test group bought more products than control group according to the number of products in their carts.

Let's check how the customers in both group purchase according to the amount spent for the campaigns.

In [82]:
```
figure = px.scatter(data_frame = ab_data,
                    x="Amount Spent",
                    y="Purchases",
                    size="Purchases",
                    color= "Campaign Name",
                    trendline="ols")
figure.show("notebook")
```



Conclusion

The overall purchase according to amount of money spent on campaigns are similar for both campaigns. Thus, it is hard to say which one is better for revenue increase. However, it can be said that, the control group campaign resulted in more traffic and more interactions while test group campaign resulted in higher conversion rates. For marketing specific product, the test campaign can work better but for increasing brand awareness, the control campaign would be better.