

```
In [1]: 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"
```

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

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

	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	1787	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	6806.0	1737.0	1549.0	1134.0	372.0
3	Control Campaign	4.08.2019	1940	72879.0	61236.0	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()
```

	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	956	340
4	Test Campaign	5.08.2019	2297	114295	95138	5863	2106	858	956	768

## Data preparation

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

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

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

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

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

```
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 recieved"].fillna(value=control_data["Searches recieved"].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)
```

## Merge 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.head())

   Campaign Name      Date  Amount Spend  Number of Impressions  Reach \
0  Control Campaign  1.08.2019          2288             82702.0  56930.0
1    Test Campaign  1.08.2019          3008             39550.0  35820.0
2    Test Campaign  10.08.2019          2790             95554.0  79632.0
3  Control Campaign  10.08.2019          2149             117624.0  91257.0
4    Test Campaign  11.08.2019          2428             83633.0  71286.0
```

```
   Website Clicks  Searches recieved  Content Viewed  Added to Cart  Purchases
0             7016.0             2290.0             2159.0             1819.0             618.0
1             8110.0             2033.0             1841.0             1219.0             511.0
2             6806.0             1737.0             1549.0             1134.0             372.0
3             3065.0             1042.0             982.0             1183.0             340.0
4             2277.0             2475.0             1984.0             1029.0             734.0
4             3759.0             2893.0             2617.0             1875.0             668.0
```

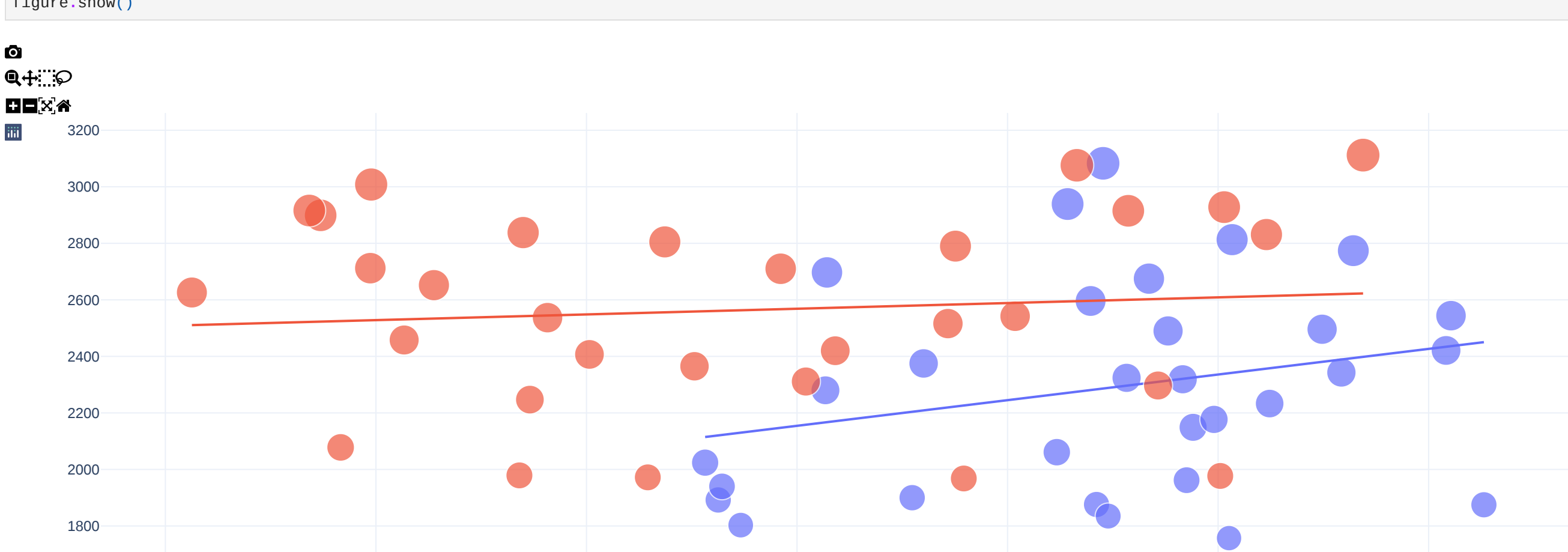
C:\bra\lib\site-packages\pandas\core\reshape\merge.py:1285: UserWarning: You are merging on int and float columns where the float values are not equal to their int representation.
warnings.warn(

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

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

A/B testing is to find the best strategy to get started with ab testing we will start with first analyzing relationship no of impressions and amount spent on both the campaigns

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



the control campaign result in more impression according to the amount spend on both the campaigns

now we will look for number of searches performed on the website from both the campaigns

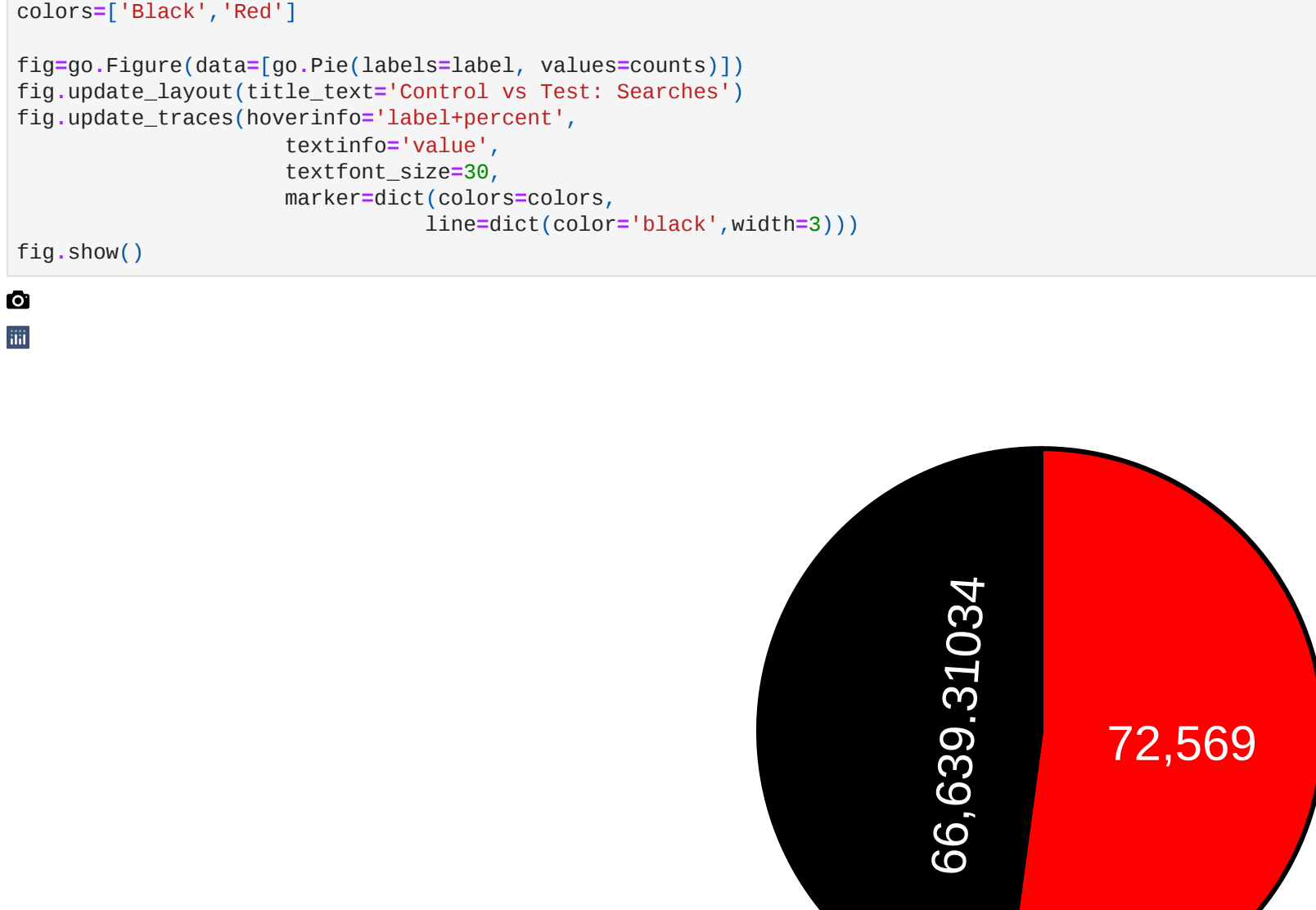
```
In [12]: label=["Total Searches from Control campaign",
            "Total searches from Test campaign"]

counts=[sum(control_data["Searches recieved"]),
        sum(test_data["Searches recieved"])]

colors=["Black","Red"]

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()
```



The test campaign resulted in more searches on the website

have a look at the number of website clicks on both website

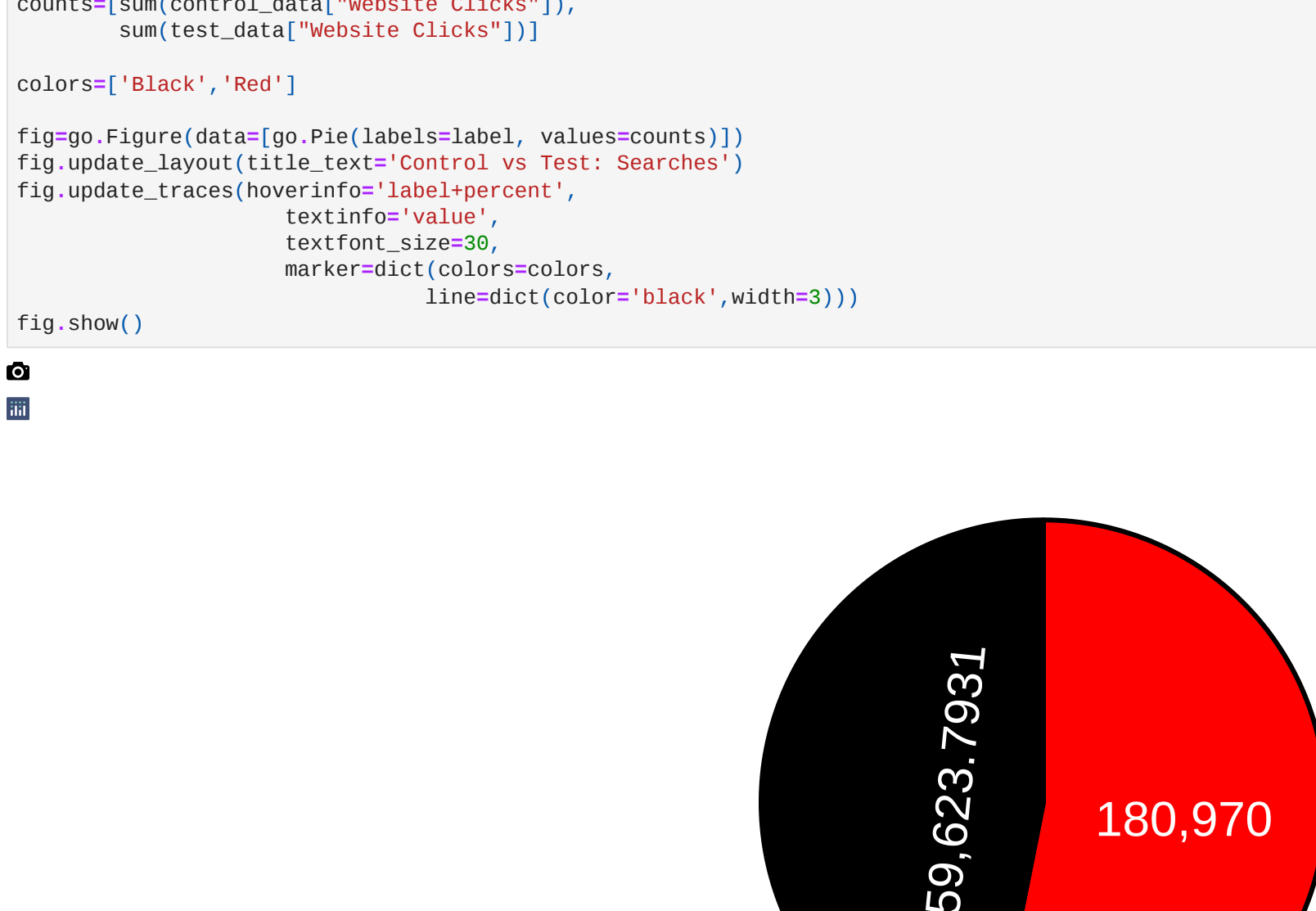
```
In [13]: label=["Website Clicks from Control campaign",
            "Website Clicks from Test campaign"]

counts=[sum(control_data["Website Clicks"]),
        sum(test_data["Website Clicks"])]

colors=["Black","Red"]

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()
```



the test campaign wins in the number of websites clicked

have a look at the content viewed after reaching the website from both campaigns

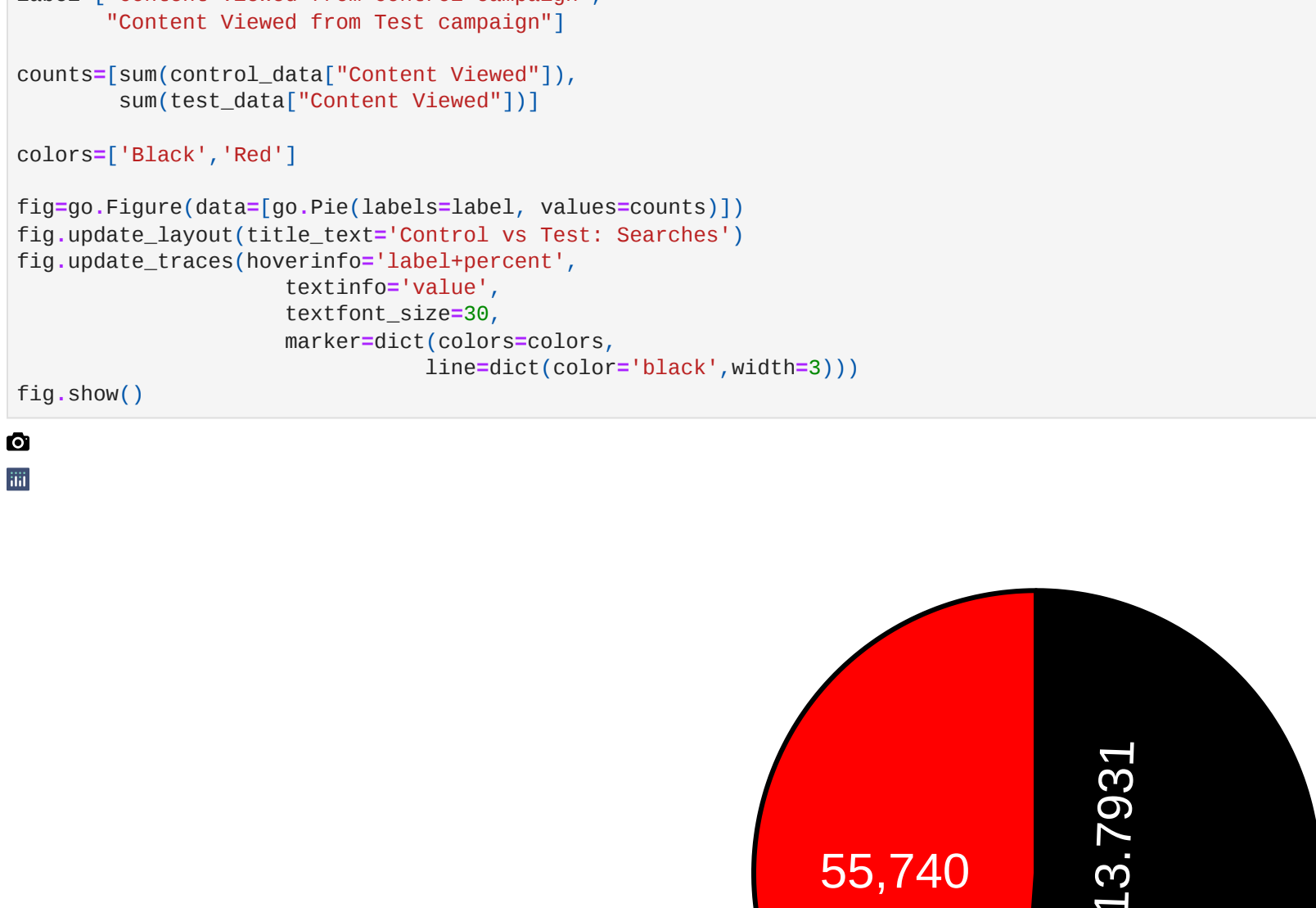
```
In [14]: label=["Content Viewed from Control campaign",
            "Content Viewed from Test campaign"]

counts=[sum(control_data["Content Viewed"]),
        sum(test_data["Content Viewed"])]

colors=["Black","Red"]

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()
```



the audience of control campaign viewed more content than test campaign although there is no much difference , as website clicks of control campaign were low its engagement on website is higher than test campaign

now have a look at products added to cart

```
In [15]: label=["Products Added to cart from Control campaign",
            "Products Added to cart from Test campaign"]

counts=[sum(control_data["Added to Cart"]),
        sum(test_data["Added to Cart"])]

colors=["Black","Red"]

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()
```



despite low website clicks the number of products added to cart from control campaign is more then test campaign

now have a look at amount spend on both campaigns

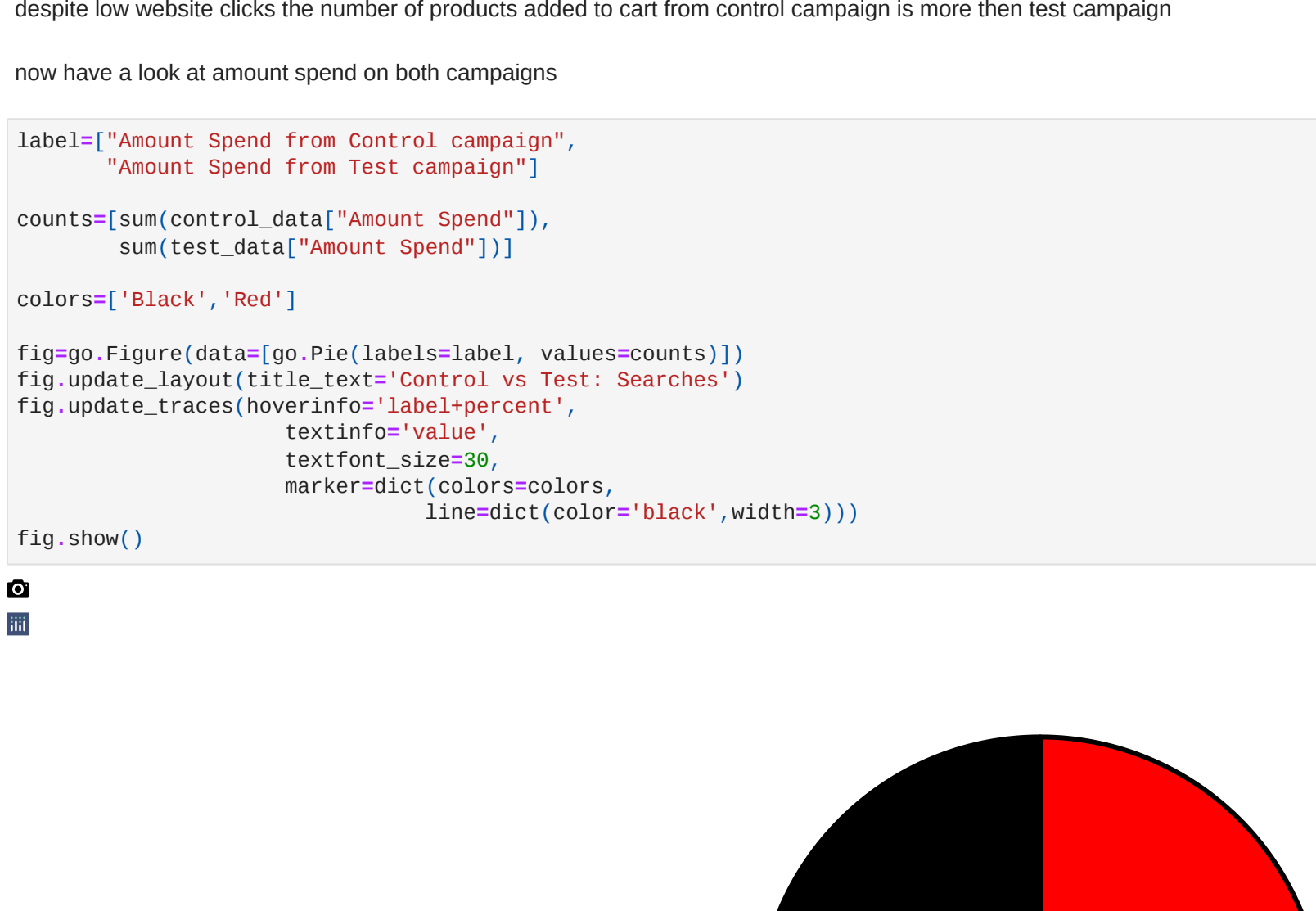
```
In [16]: label=["Amount Spend from Control campaign",
            "Amount Spend from Test campaign"]

counts=[sum(control_data["Amount Spend"]),
        sum(test_data["Amount Spend"])]

colors=["Black","Red"]

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()
```



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.

have a look at purchases made on both websites

```
In [17]: label=["Purchases made from Control campaign",
            "Purchases made from Test campaign"]

counts=[sum(control_data["Purchases"]),
        sum(test_data["Purchases"])]

colors=["Black","Red"]

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)))

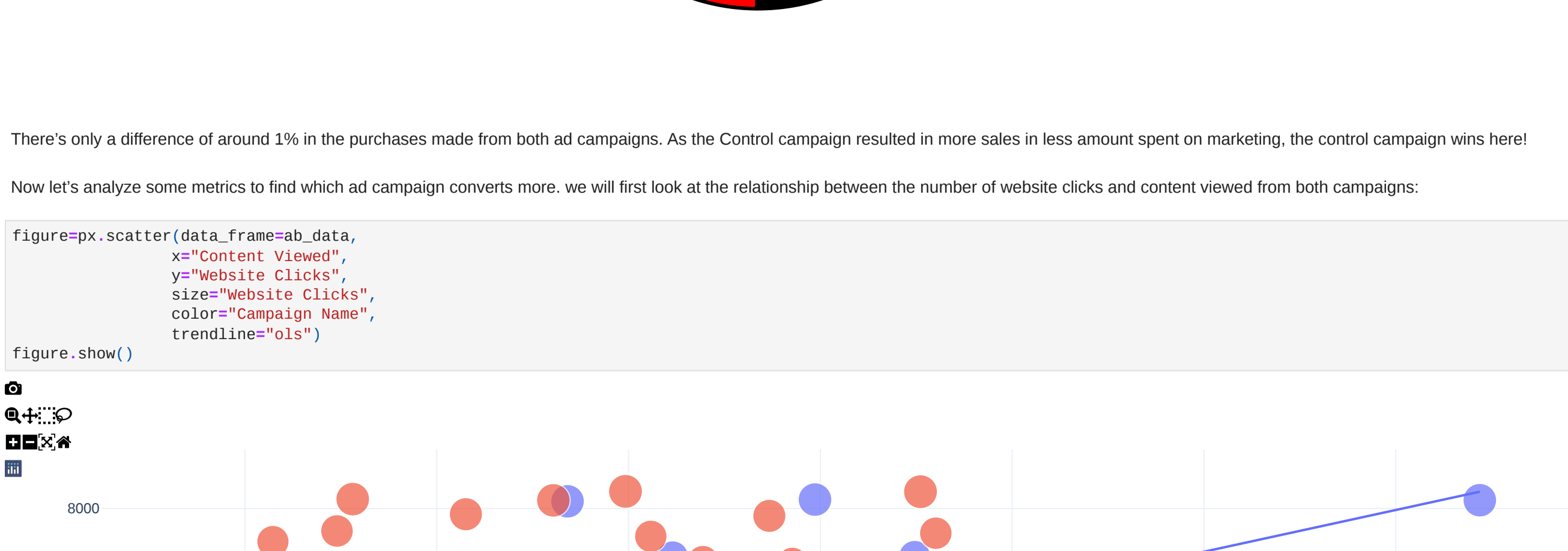
fig.show()
```



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 in which ad campaign converts more. we will first look at the relationship between the number of website clicks and content viewed from both campaigns:

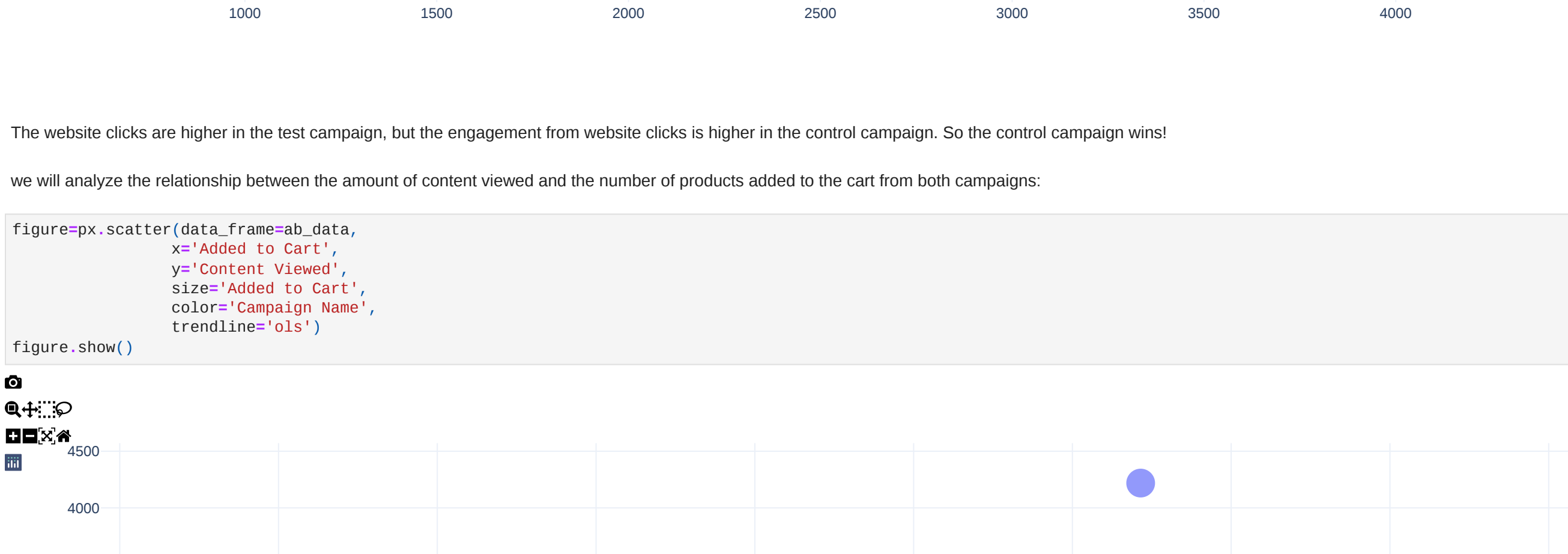
```
In [18]: figure=px.scatter(data_frame=ab_data,
                        x="Content Viewed",
                        y="Website Clicks",
                        size="Website Clicks",
                        color="Campaign Name",
                        trendline="ols")
figure.show()
```



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!

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

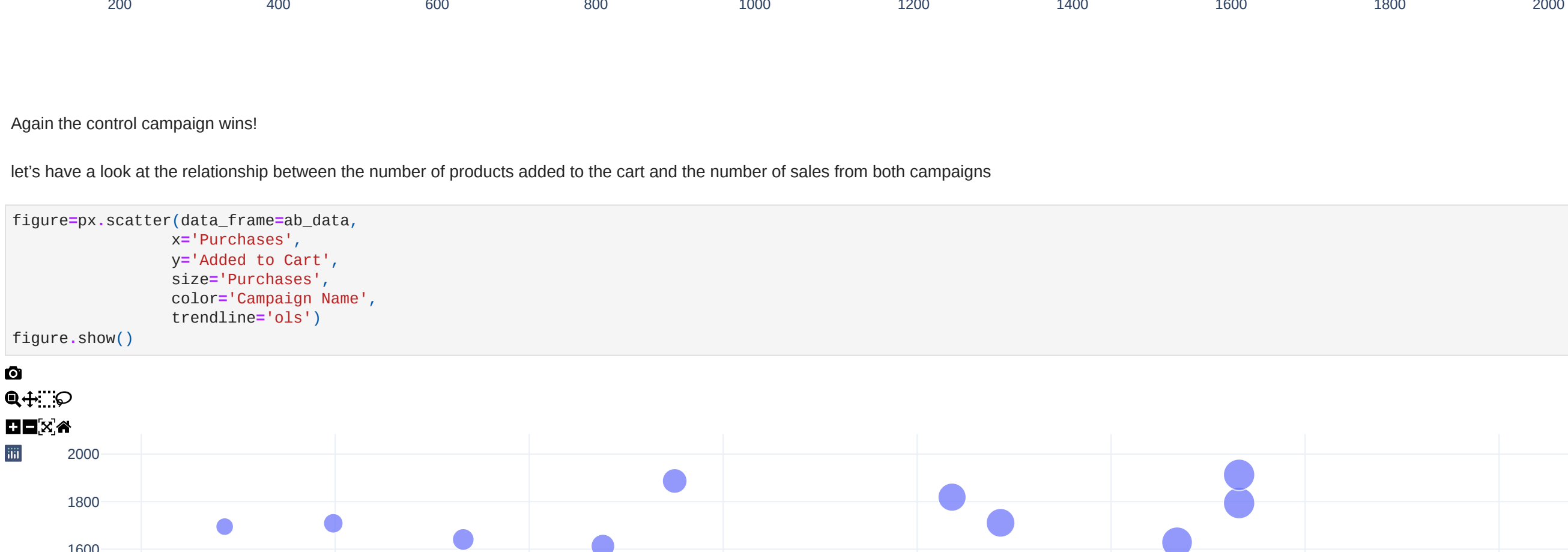
```
In [19]: figure=px.scatter(data_frame=ab_data,
                        x="Added to Cart",
                        y="Content Viewed",
                        size="Added to Cart",
                        color="Campaign Name",
                        trendline="ols")
figure.show()
```



Again the control campaign wins!

let's have a look at the number of products added to the cart and the number of sales from both campaigns

```
In [20]: figure=px.scatter(data_frame=ab_data,
                        x="Purchases",
                        y="Website Clicks",
                        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 conversion 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.

```
In [ ]:
```