

A/B Testing using Python

- 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
- Add to Cart: Number of users who added products to the cart
- # of Purchase: Number of purchases

Two campaigns were performed by the company:

- Control Campaign
- Test Campaign

```
In [13]: import pandas as pd
import numpy as np
import datetime
from datetime import date, timedelta
import plotly.graph_objs as go
import plotly.express as px
import plotly.io as pio
pio.templates.default = "plotly_white"

control_data = pd.read_csv("control_group.csv", sep=";")
test_data = pd.read_csv("test_group.csv", sep=";")
```

```
In [2]: 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	2290	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	1163.0	343.0
4	Control Campaign	5.08.2019	1835	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
In [3]: 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	3028	39550	35820	3038	1946	1069	894	255
1	Test Campaign	2.08.2019	2642	100719	91236	4657	2369	1548	879	477
2	Test Campaign	3.08.2019	2885	70283	45198	7885	2972	2367	1268	578
3	Test Campaign	4.08.2019	2710	79451	25937	4216	2216	1437	966	340
4	Test Campaign	5.08.2019	2297	114295	95138	5863	2106	858	956	768

Data Preparation

```
In [4]: control_data.columns = ["Campaign Name", "Date", "Amount Spent",
                             "Number of Impressions", "Reach", "Website Clicks",
                             "Searches Received", "Content Viewed", "Added to Cart",
                             "Purchases"]
```

```
In [5]: control_data.head()
```

	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	2290	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	1163.0	343.0
4	Control Campaign	5.08.2019	1835	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
In [6]: test_data.columns = ["Campaign Name", "Date", "Amount Spent",
                             "Number of Impressions", "Reach", "Website Clicks",
                             "Searches Received", "Content Viewed", "Added to Cart",
                             "Purchases"]
```

```
In [7]: test_data.head()
```

	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	3028	39550	35820	3038	1946	1069	894	255
1	Test Campaign	2.08.2019	2642	100719	91236	4657	2369	1548	879	477
2	Test Campaign	3.08.2019	2885	70283	45198	7885	2972	2367	1268	578
3	Test Campaign	4.08.2019	2710	79451	25937	4216	2216	1437	966	340
4	Test Campaign	5.08.2019	2297	114295	95138	5863	2106	858	956	768

```
In [8]: control_data.isnull().sum()
```

```
Out[8]: Campaign Name      0
Date                      0
Amount Spent              0
Number of Impressions     0
Reach                    1
Website Clicks            1
Searches Received        1
Content Viewed           1
Added to Cart            1
Purchases                1
dtype: int64
```

```
In [9]: test_data.isnull().sum()
```

```
Out[9]: 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 [10]: control_data.shape
```

```
Out[10]: (50, 10)
```

```
In [11]: test_data.shape
```

```
Out[11]: (50, 10)
```

```
In [12]: 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)
```

```
In [13]: control_data["Number of Impressions"] = control_data["Number of Impressions"].astype(np.int64)
control_data["Reach"] = control_data["Reach"].astype(np.int64)
control_data["Website Clicks"] = control_data["Website Clicks"].astype(np.int64)
control_data["Searches Received"] = control_data["Searches Received"].astype(np.int64)
control_data["Content Viewed"] = control_data["Content Viewed"].astype(np.int64)
control_data["Added to Cart"] = control_data["Added to Cart"].astype(np.int64)
control_data["Purchases"] = control_data["Purchases"].astype(np.int64)
```

```
In [14]: control_data.isnull().sum()
```

```
Out[14]: 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 [15]: ab_data = control_data.merge(test_data, how="outer", sort_values(["Date"])
ab_data = ab_data.reset_index(drop=True)
ab_data.head()
```

	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	2290	82702	56930	7016	2290	2159	1819	618
1	Test Campaign	1.08.2019	3028	39550	35820	3038	1946	1069	894	255
2	Test Campaign	10.08.2019	2790	90504	79632	8125	2312	1804	424	275
3	Control Campaign	10.08.2019	2149	117024	81257	2277	2475	1984	1628	734
4	Test Campaign	11.08.2019	2420	83633	71286	3750	2893	2617	1076	668

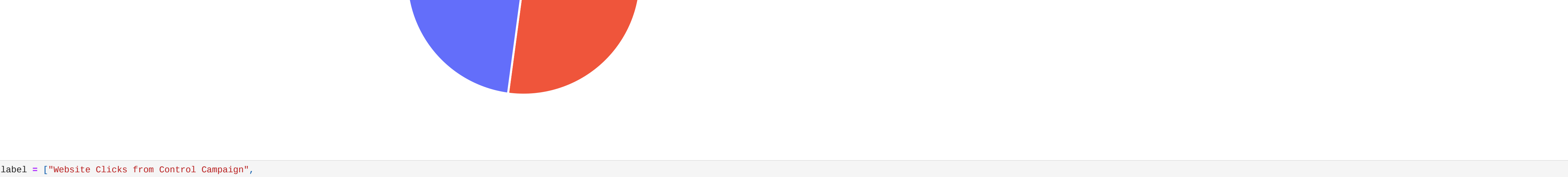
```
In [16]: ab_data["Campaign Name"].value_counts()
```

```
Out[16]: Campaign Name
Control Campaign    38
Test Campaign       12
Name: count, dtype: int64
```

A/B Testing to Find the Best Marketing Strategy

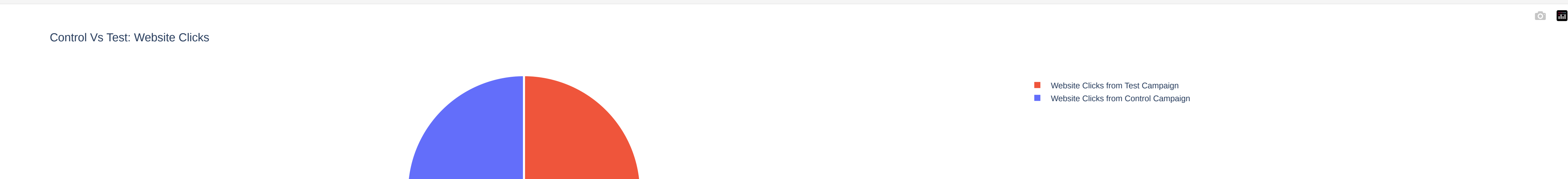
```
In [17]: figure = px.scatter(data_frame = ab_data,
                             x="Number of Impressions",
                             y="Amount Spent",
                             size="Amount Spent",
                             color="Campaign Name",
                             trendLine="ols")
```

```
figure.show()
```



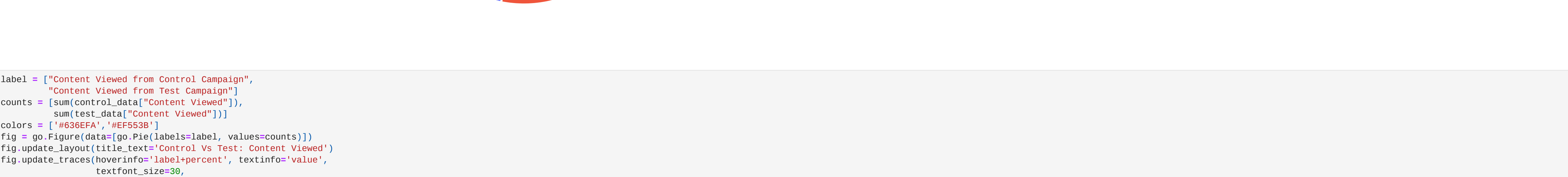
```
In [18]: label = ["Total Searches from Control Campaign",
                 "Total Searches from Test Campaign"]
counts = [sum(control_data["Searches Received"]),
          sum(test_data["Searches Received"])]
colors = ["#808080", "#FF5555"]
fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text="Control vs Test: Searches")
fig.update_traces(hoverinfo="label+percent", textinfo="value",
                  textfont_size=30,
                  marker=dict(colors=colors,
                              linedict(color="white", width=3)))
fig.show()
```

```
Control Vs Test: Searches
```



```
In [19]: label = ["Website Clicks from Control Campaign",
                 "Website Clicks from Test Campaign"]
counts = [sum(control_data["Website Clicks"]),
          sum(test_data["Website Clicks"])]
colors = ["#808080", "#FF5555"]
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,
                              linedict(color="white", width=3)))
fig.show()
```

```
Control Vs Test: Website Clicks
```



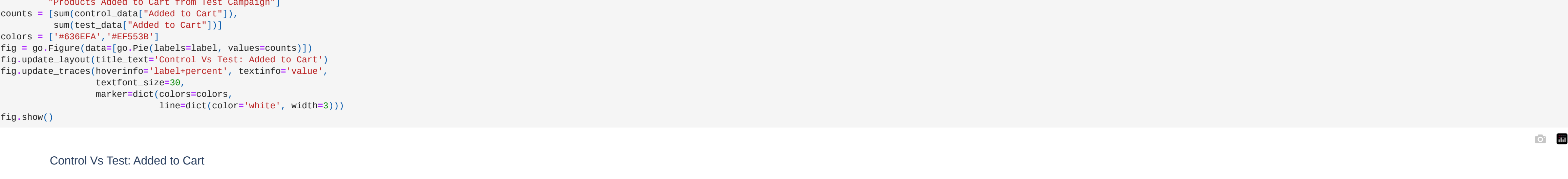
```
In [20]: label = ["Content Viewed from Control Campaign",
                 "Content Viewed from Test Campaign"]
counts = [sum(control_data["Content Viewed"]),
          sum(test_data["Content Viewed"])]
colors = ["#808080", "#FF5555"]
fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text="Control vs Test: Content Viewed")
fig.update_traces(hoverinfo="label+percent", textinfo="value",
                  textfont_size=30,
                  marker=dict(colors=colors,
                              linedict(color="white", width=3)))
fig.show()
```

```
Control Vs Test: Content Viewed
```



```
In [21]: 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 = ["#808080", "#FF5555"]
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,
                              linedict(color="white", width=3)))
fig.show()
```

```
Control Vs Test: Added to Cart
```



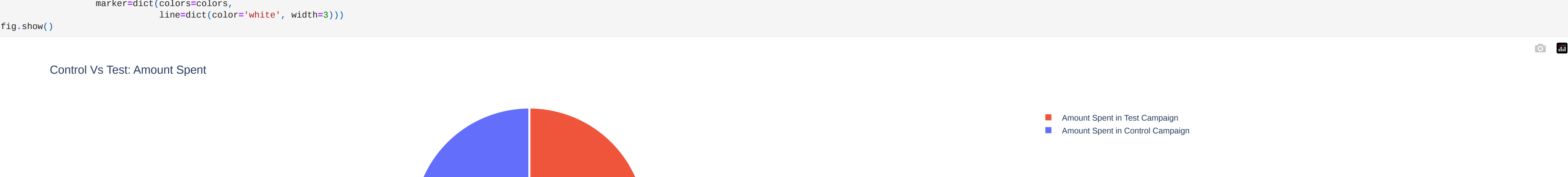
```
In [22]: label = ["Amount Spent in Control Campaign",
                 "Amount Spent in Test Campaign"]
counts = [sum(control_data["Amount Spent"]),
          sum(test_data["Amount Spent"])]
colors = ["#808080", "#FF5555"]
fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text="Control vs Test: Amount Spent")
fig.update_traces(hoverinfo="label+percent", textinfo="value",
                  textfont_size=30,
                  marker=dict(colors=colors,
                              linedict(color="white", width=3)))
fig.show()
```

```
Control Vs Test: Amount Spent
```



```
In [23]: label = ["Purchases Made by Control Campaign",
                 "Purchases Made by Test Campaign"]
counts = [sum(control_data["Purchases"]),
          sum(test_data["Purchases"])]
colors = ["#808080", "#FF5555"]
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,
                              linedict(color="white", width=3)))
fig.show()
```

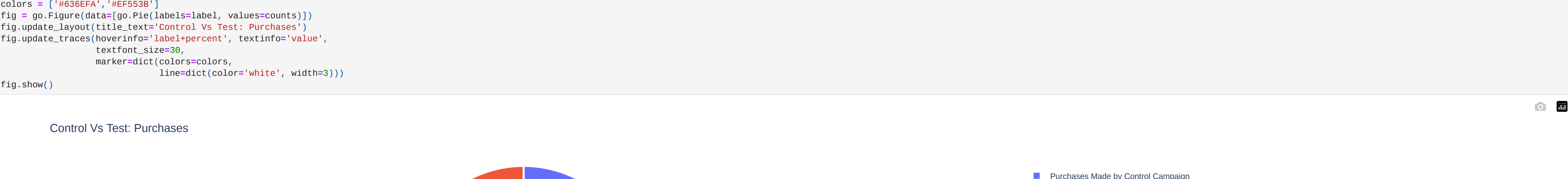
```
Control Vs Test: Purchases
```



```
In [24]: figure = px.scatter(data_frame = ab_data,
                             x="Content Viewed",
                             y="Website Clicks",
                             size="Website Clicks",
                             color="Campaign Name",
                             trendLine="ols")
```



```
In [25]: figure = px.scatter(data_frame = ab_data,
                             x="Added to Cart",
                             y="Content Viewed",
                             size="Added to Cart",
                             color="Campaign Name",
                             trendLine="ols")
```



```
In [26]: figure = px.scatter(data_frame = ab_data,
                             x="Purchases",
                             y="Added to Cart",
                             size="Purchases",
                             color="Campaign Name",
                             trendLine="ols")
```



