

a-b-testing

April 2, 2024

```
[65]: import pandas as pd
import numpy as np
from datetime import *
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[9]: control_df=pd.read_csv('/content/control_group.csv',sep=';')
test_df=pd.read_csv('/content/test_group.csv',sep=';')
```

```
[10]: control_df.head()
```

```
[10]:
```

	Campaign Name	Date	Spend [USD]	# of Impressions	Reach \
0	Control Campaign	1.08.2019	2280	82702.0	56930.0
1	Control Campaign	2.08.2019	1757	121040.0	102513.0
2	Control Campaign	3.08.2019	2343	131711.0	110862.0
3	Control Campaign	4.08.2019	1940	72878.0	61235.0
4	Control Campaign	5.08.2019	1835	NaN	NaN

	# of Website Clicks	# of Searches	# of View Content	# of Add to Cart \
0	7016.0	2290.0	2159.0	1819.0
1	8110.0	2033.0	1841.0	1219.0
2	6508.0	1737.0	1549.0	1134.0
3	3065.0	1042.0	982.0	1183.0
4	NaN	NaN	NaN	NaN

	# of Purchase
0	618.0
1	511.0
2	372.0
3	340.0
4	NaN

```
[11]: test_df.head()
```

```
[11]:
```

	Campaign Name	Date	Spend [USD]	# of Impressions	Reach \
0	Test Campaign	1.08.2019	3008	39550	35820
1	Test Campaign	2.08.2019	2542	100719	91236

2	Test Campaign	3.08.2019	2365	70263	45198
3	Test Campaign	4.08.2019	2710	78451	25937
4	Test Campaign	5.08.2019	2297	114295	95138

	# of Website Clicks	# of Searches	# of View Content	# of Add to Cart	\
0	3038	1946	1069	894	
1	4657	2359	1548	879	
2	7885	2572	2367	1268	
3	4216	2216	1437	566	
4	5863	2106	858	956	

	# of Purchase
0	255
1	677
2	578
3	340
4	768

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

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

```
[17]: control_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Campaign Name          30 non-null    object
1   Date                   30 non-null    object
2   Amount Spent           30 non-null    int64
3   Number of Impressions  29 non-null    float64
4   Reach                  29 non-null    float64
5   Website Clicks         29 non-null    float64
6   Searches Received      29 non-null    float64
7   Content Viewed         29 non-null    float64
8   Added to Cart          29 non-null    float64
9   Purchases              29 non-null    float64
dtypes: float64(7), int64(1), object(2)
memory usage: 2.5+ KB
```

```
[16]: control_df.isnull().sum()
```

```
[16]: 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
```

```
[18]: test_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   Campaign Name         30 non-null    object
 1   Date                  30 non-null    object
 2   Amount Spent          30 non-null    int64
 3   Number of Impressions 30 non-null    int64
 4   Reach                 30 non-null    int64
 5   Website Clicks        30 non-null    int64
 6   Searches Received     30 non-null    int64
 7   Content Viewed        30 non-null    int64
 8   Added to Cart         30 non-null    int64
 9   Purchases             30 non-null    int64
dtypes: int64(8), object(2)
memory usage: 2.5+ KB
```

```
[19]: test_df.isnull().sum()
```

```
[19]: 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
```

```
[22]: # Fill the null Values by taking mean of the attributes.
control_df['Number of Impressions'].fillna(control_df['Number of Impressions'].
    ↳mean(),inplace=True)
control_df['Reach'].fillna(control_df['Reach'].mean(),inplace=True)
control_df['Website Clicks'].fillna(control_df['Website Clicks'].
    ↳mean(),inplace=True)
control_df['Searches Received'].fillna(control_df['Searches Received'].
    ↳mean(),inplace=True)
control_df['Content Viewed'].fillna(control_df['Content Viewed'].
    ↳mean(),inplace=True)
control_df['Added to Cart'].fillna(control_df['Added to Cart'].
    ↳mean(),inplace=True)
control_df['Purchases'].fillna(control_df['Purchases'].mean(),inplace=True)
```

```
[23]: control_df.isnull().sum().sum()
```

```
[23]: 0
```

```
[24]: control_df.describe(include=[int,float])
```

```
[24]:
```

	Amount Spent	Number of Impressions	Reach	Website Clicks	\
count	30.000000	30.000000	30.000000	30.000000	
mean	2288.433333	109559.758621	88844.931034	5320.793103	
std	367.334451	21311.695472	21452.627592	1726.803732	
min	1757.000000	71274.000000	42859.000000	2277.000000	
25%	1945.500000	95191.250000	75300.250000	4122.250000	
50%	2299.500000	112368.000000	91418.000000	5272.396552	
75%	2532.000000	121259.000000	101958.750000	6609.500000	
max	3083.000000	145248.000000	127852.000000	8137.000000	

	Searches Received	Content Viewed	Added to Cart	Purchases
count	30.000000	30.000000	30.000000	30.000000
mean	2221.310345	1943.793103	1300.000000	522.793103
std	851.025795	764.021907	400.371207	181.810508
min	1001.000000	848.000000	442.000000	222.000000
25%	1629.250000	1249.000000	942.500000	375.500000
50%	2340.000000	1979.500000	1319.500000	506.000000
75%	2655.000000	2360.500000	1638.000000	663.250000
max	4891.000000	4219.000000	1913.000000	800.000000

```
[25]: test_df.describe(include=[int,float])
```

```
[25]:
```

	Amount Spent	Number of Impressions	Reach	Website Clicks	\
count	30.000000	30.000000	30.000000	30.000000	
mean	2563.066667	74584.800000	53491.566667	6032.333333	
std	348.687681	32121.377422	28795.775752	1708.567263	
min	1968.000000	22521.000000	10598.000000	3038.000000	

25%	2324.500000	47541.250000	31516.250000	4407.000000
50%	2584.000000	68853.500000	44219.500000	6242.500000
75%	2836.250000	99500.000000	78778.750000	7604.750000
max	3112.000000	133771.000000	109834.000000	8264.000000

	Searches Received	Content Viewed	Added to Cart	Purchases
count	30.000000	30.000000	30.000000	30.000000
mean	2418.966667	1858.000000	881.533333	521.233333
std	388.742312	597.654669	347.584248	211.047745
min	1854.000000	858.000000	278.000000	238.000000
25%	2043.000000	1320.000000	582.500000	298.000000
50%	2395.500000	1881.000000	974.000000	500.000000
75%	2801.250000	2412.000000	1148.500000	701.000000
max	2978.000000	2801.000000	1391.000000	890.000000

From above descriptive stats by comparing both control and test group , If you see carefully where the average amount spent in test group is Rs.2563 is greater than the average amount spent in control group is Rs.2288.

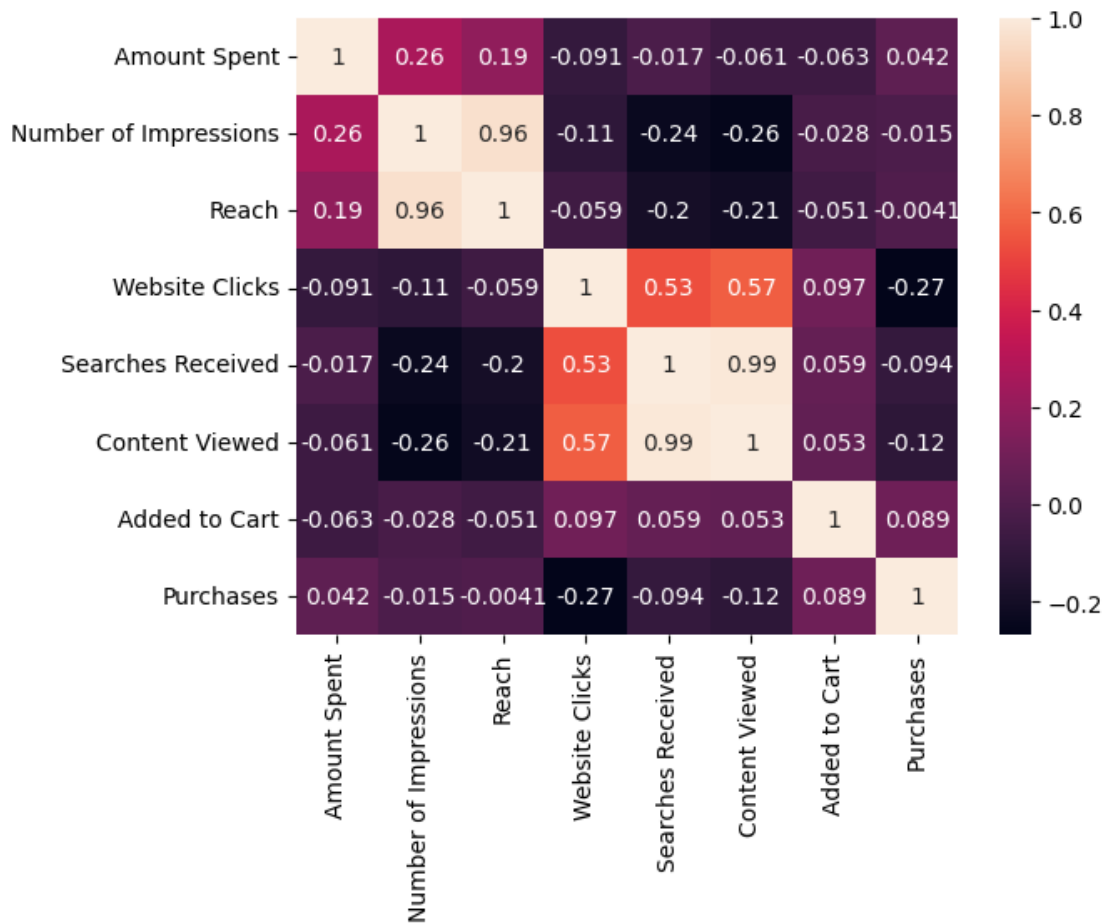
Average Website Clicks in test group is 6032,Where in control group is 5320.

The Mean purchases of both groups are same.But if you see maximum purchases in test group is 890 is greater than maximum purchases in control group which is only 800.So the conversion rate is higher in test group.

```
[41]: sns.heatmap(control_df.corr(),annot=True)
plt.show()
```

```
<ipython-input-41-db305e0d8fce>:1: FutureWarning:
```

The default value of `numeric_only` in `DataFrame.corr` is deprecated. In a future version, it will default to `False`. Select only valid columns or specify the value of `numeric_only` to silence this warning.

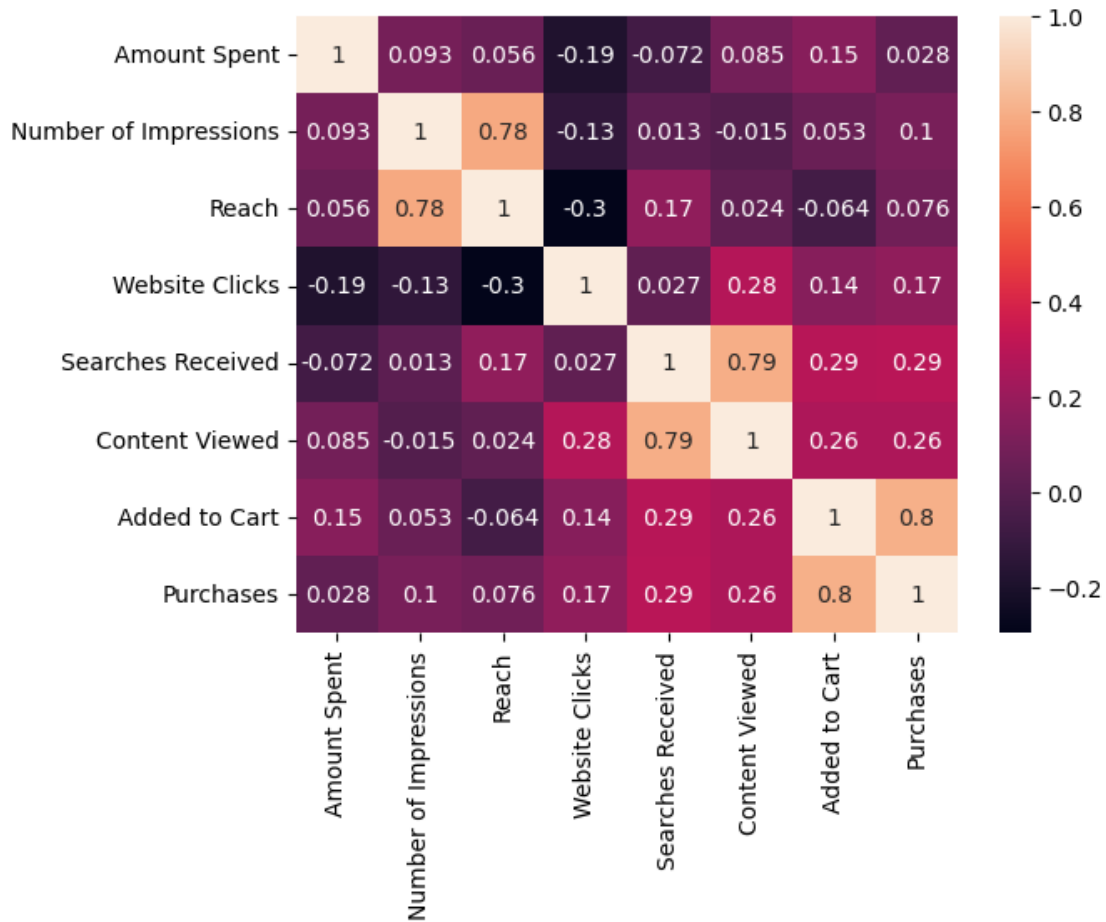


From the above heatmap of control group if you see that website clicks ,content viewed and searches received attributes are partially postively correlated.

```
[42]: sns.heatmap(test_df.corr(),annot=True)
plt.show()
```

<ipython-input-42-15a6942cfe5b>:1: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.



Here a surprise that the persons who mostly adding the products in cart is more likely to purchase the products (High Positive Correlation of 0.8).Where we does not the see the customer behaviour in control group.

Number of impressions and reach are positively correlated.

```
[62]: ab_test_df=pd.concat([control_df,test_df]).sort_values('Date').reset_index()
ab_test_df.drop('index',inplace=True,axis=1)
```

```
[64]: ab_test_df.head()
```

```
[64]:
```

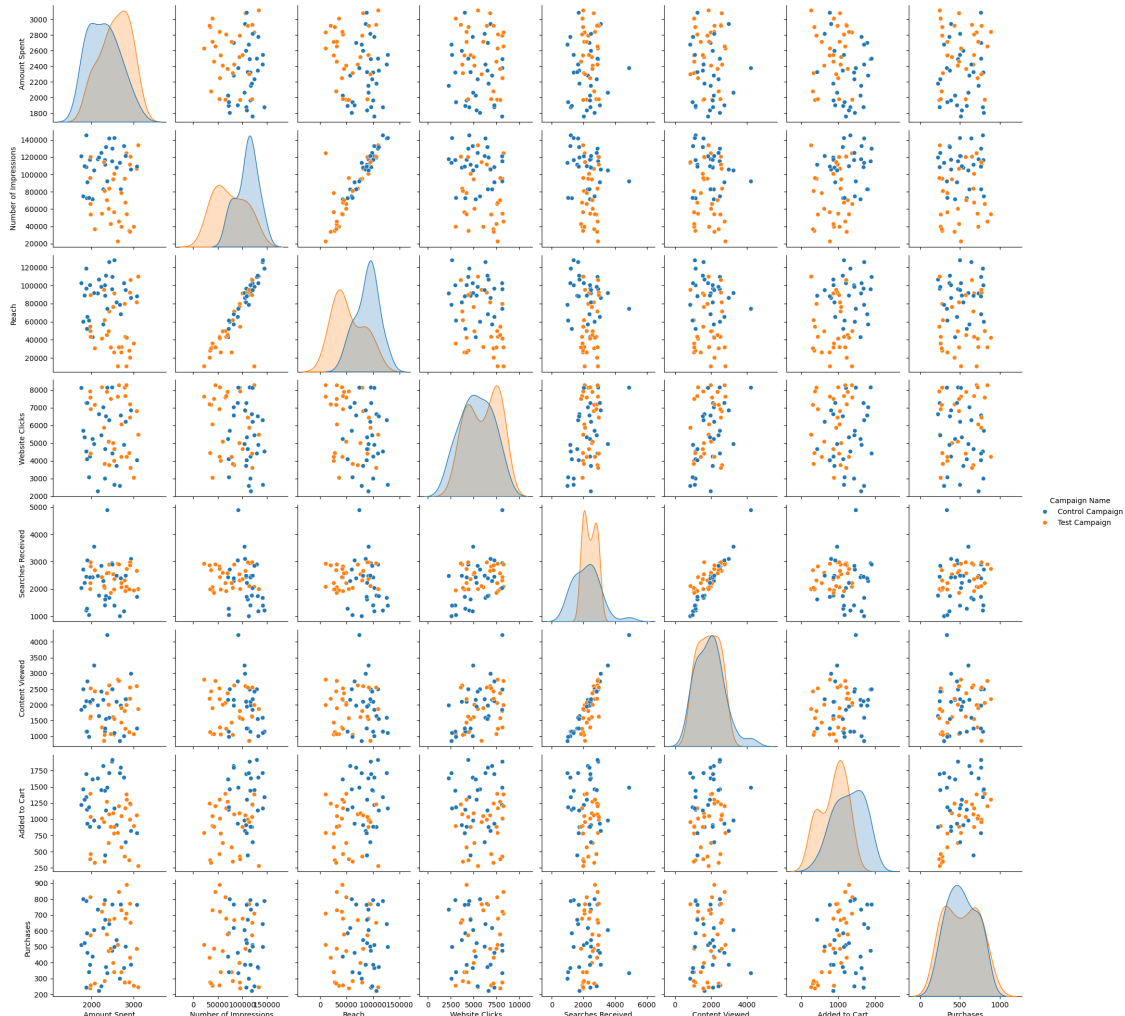
	Campaign Name	Date	Amount Spent	Number of Impressions	Reach
0	Control Campaign	1.08.2019	2280	82702.0	56930.0
1	Test Campaign	1.08.2019	3008	39550.0	35820.0
2	Test Campaign	10.08.2019	2790	95054.0	79632.0
3	Control Campaign	10.08.2019	2149	117624.0	91257.0
4	Test Campaign	11.08.2019	2420	83633.0	71286.0

Website Clicks Searches Received Content Viewed Added to Cart Purchases

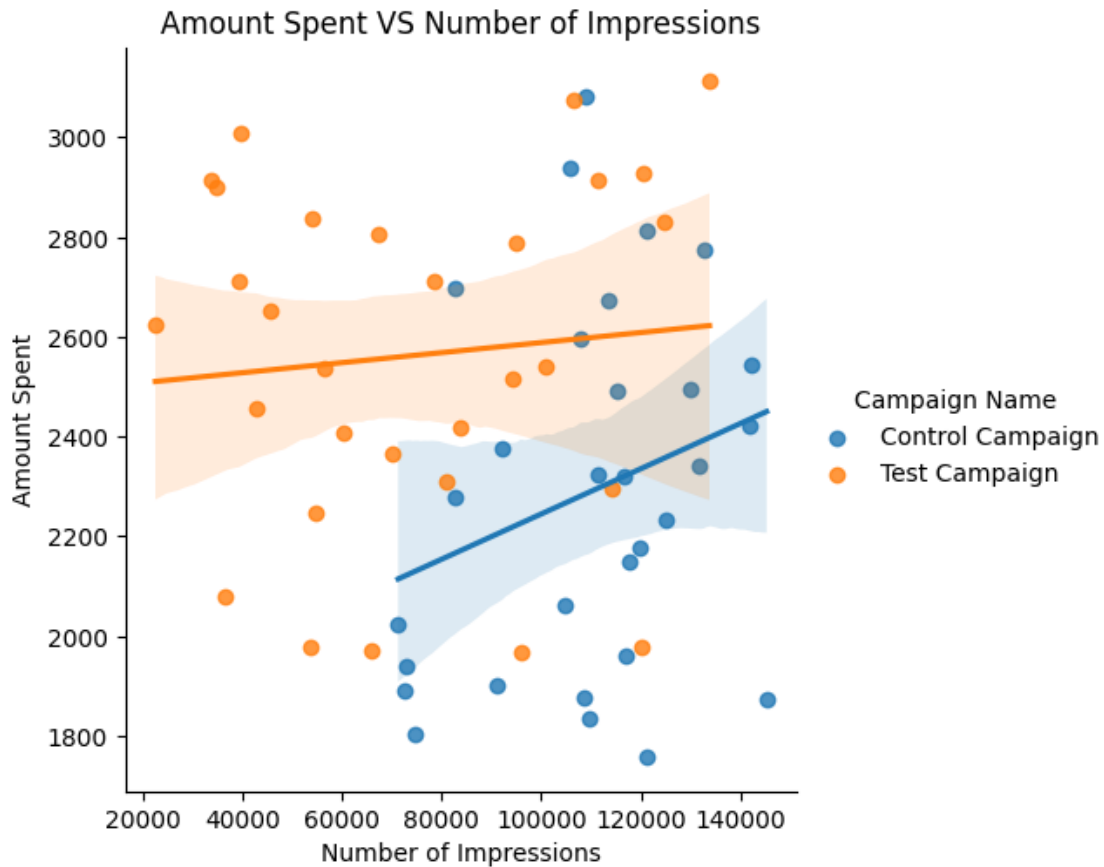
0	7016.0	2290.0	2159.0	1819.0	618.0
1	3038.0	1946.0	1069.0	894.0	255.0
2	8125.0	2312.0	1804.0	424.0	275.0
3	2277.0	2475.0	1984.0	1629.0	734.0
4	3750.0	2893.0	2617.0	1075.0	668.0

```
[67]: sns.pairplot(ab_test_df, hue='Campaign Name')
```

```
[67]: <seaborn.axisgrid.PairGrid at 0x7a401f213640>
```



```
[108]: sns.lmplot(data=ab_test_df, x='Number of Impressions', y='Amount Spent', hue='Campaign Name')
plt.title("Amount Spent VS Number of Impressions")
plt.show()
```

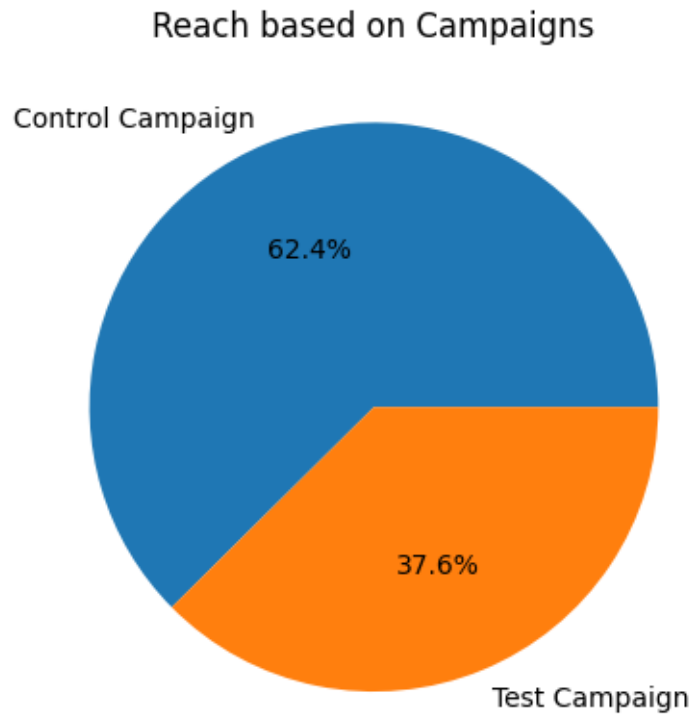



In Control Campaign the Number of impressions are higher than Test Campaign. In Control Campaign we can see that Amount they spent increases, Where the number of impressions also somewhat increases.

```
[88]: reach=ab_test_df[['Campaign Name','Reach']].groupby('Campaign Name').
      ↪sum('Reach').reset_index()
```

```
[88]:      Campaign Name      Reach
0  Control Campaign  2.665348e+06
1   Test Campaign  1.604747e+06
```

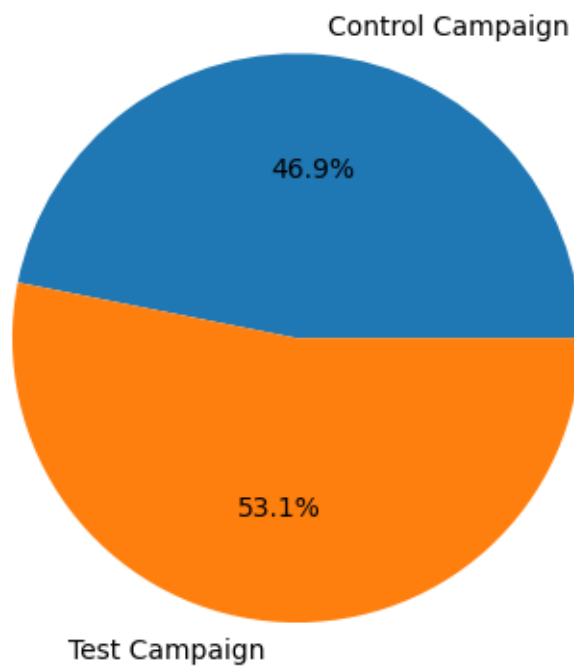
```
[92]: plt.pie(x=reach['Reach'].tolist(),labels=reach['Campaign Name'].
      ↪tolist(),autopct='%1.1f%%')
plt.title('Reach based on Campaigns')
plt.show()
```



```
[94]: clicks=ab_test_df[['Campaign Name','Website Clicks']].groupby('Campaign Name').  
      ↪sum('Website Clicks').reset_index()
```

```
[96]: plt.pie(x=clicks['Website Clicks'].tolist(),labels=clicks['Campaign Name'].  
      ↪tolist(),autopct='%1.1f%%')  
plt.title('Website Clicks based on Campaigns')  
plt.show()
```

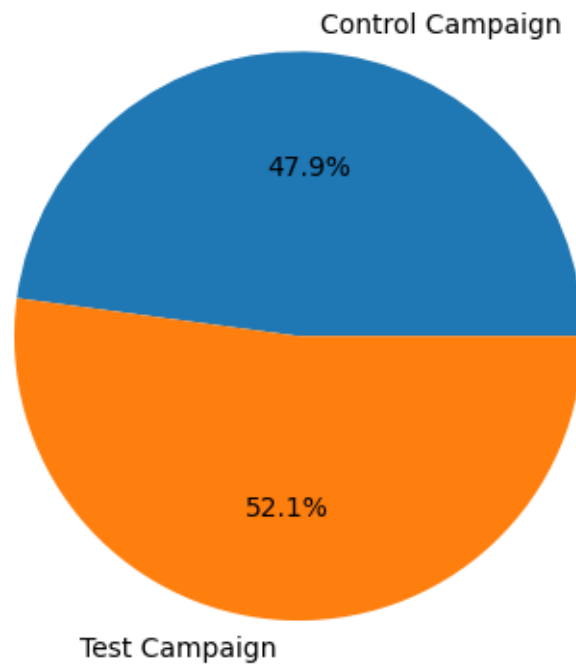
Website Clicks based on Campaigns



```
[98]: search=ab_test_df[['Campaign Name','Searches Received']].groupby('Campaign_↵  
↵Name').sum('Searches Received').reset_index()
```

```
[100]: plt.pie(x=search['Searches Received'].tolist(),labels=search['Campaign Name'].  
↵tolist(),autopct='%1.1f%%')  
plt.title('Searches Received based on Campaigns')  
plt.show()
```

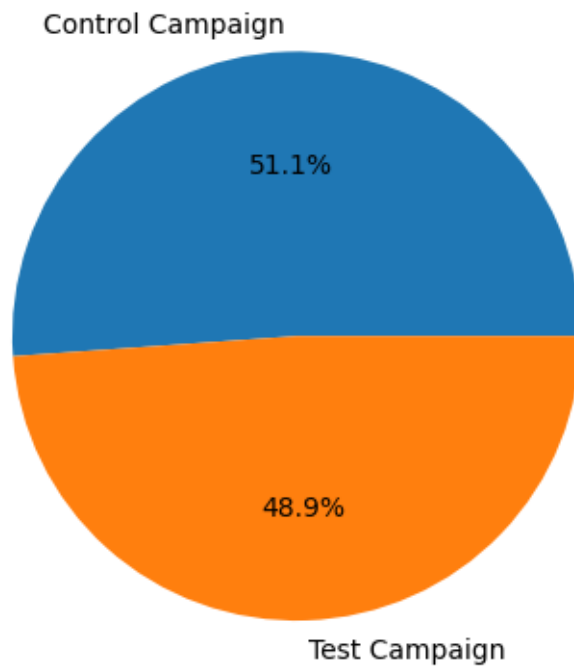
Searches Received based on Campaigns



```
[101]: content=ab_test_df[['Campaign Name','Content Viewed']].groupby('Campaign Name').  
        ↳sum('Content Viewed').reset_index()
```

```
[103]: plt.pie(x=content['Content Viewed'].tolist(),labels=content['Campaign Name'].  
        ↳tolist(),autopct='%1.1f%%')  
plt.title('Content Viewed based on Campaigns')  
plt.show()
```

Content Viewed based on Campaigns



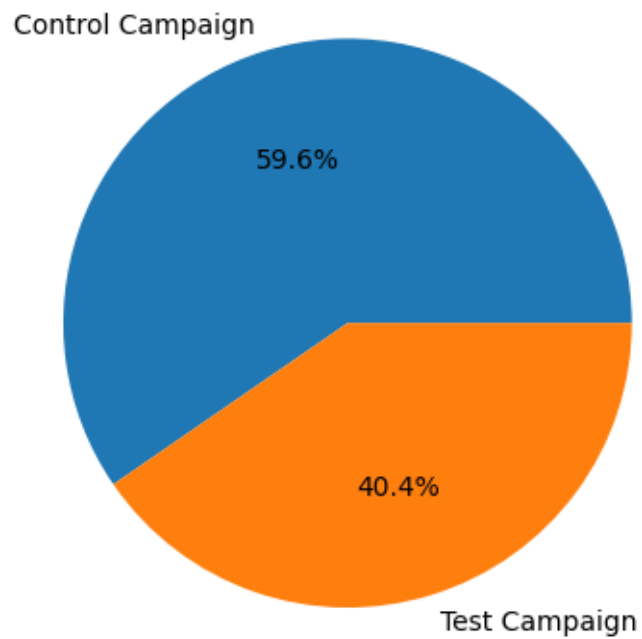
```
[104]: cart=ab_test_df[['Campaign Name','Added to Cart']].groupby('Campaign Name').  
        ↪sum('Added to Cart').reset_index()
```

```
[104]:
```

	Campaign Name	Added to Cart
0	Control Campaign	39000.0
1	Test Campaign	26446.0

```
[105]: plt.pie(x=cart['Added to Cart'].tolist(),labels=cart['Campaign Name'].  
        ↪tolist(),autopct='%1.1f%%')  
plt.title('Added to Cart based on Campaigns')  
plt.show()
```

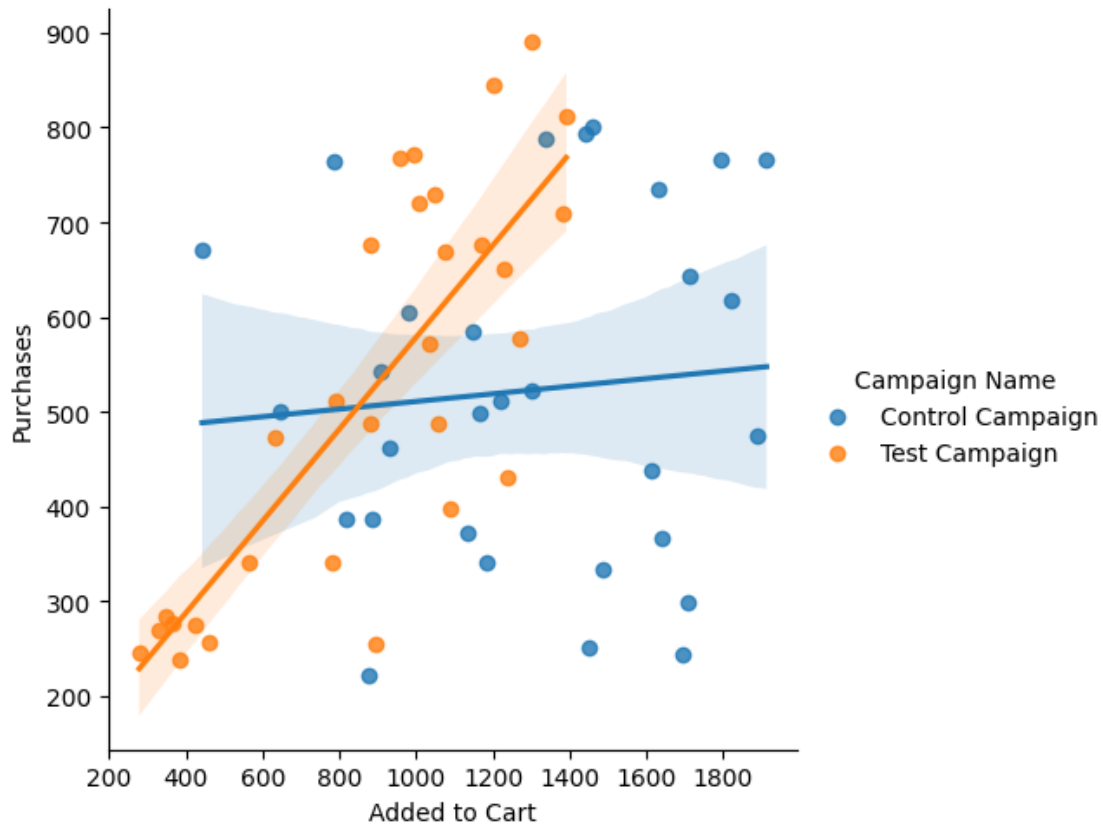
Added to Cart based on Campaigns



If we see that 60% of products added to cart in Control Campaign. Only 40% of products added to cart in Test Campaign.

```
[109]: sns.lmplot(data=ab_test_df, x='Added to Cart', y='Purchases', hue='Campaign Name')
```

```
[109]: <seaborn.axisgrid.FacetGrid at 0x7a40192ac640>
```



From the above plot we can able to understand .In Control Campaign even though more products are added to cart where it is not purchased.

In Test Campaign less products are added to cart , Where products are purchased. The persons who mostly adding the products in cart is more likely to purchase the products.Because you see linearity of the plot.Highly positively correlated

Conclusion : In Test Campaign Where the **Conversion Rate** is higher than Control Campaign.Control Campaign drives more number of impressions,reach and more people added to cart.

At the end of day Purchaes / Sales are more important because it drives revenue & profit.Test Campaign can be used to market specific products to specific audiences (Personalized Marketing / recommendations) based on the customer behaviours & demographics.While Control Campaign are used to market products for wider audiences (General Marketing).