In [1]:
```python
#import the libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [2]:
```python
#import the data
file_path = r'C:\Users\Kavata Muia\Desktop\Data Analysis Practices\Instagram data.csv'
data = pd.read_csv(file_path, encoding = 'latin1')
#displays the first 5 rows
print(data.head())
```

```
   Impressions  From Home  From Hashtags  From Explore  From Other  Saves  \
0         3920       2586           1028           619          56     98
1         5394       2727           1838          1174          78    194
2         4021       2085           1188             0         533     41
3         4528       2700            621           932          73    172
4         2518       1704            255           279          37     96

   Comments  Shares  Likes  Profile Visits  Follows  \
0         9       5    162              35        2
1         7      14    224              48       10
2        11       1    131              62       12
3        10       7    213              23        8
4         5       4    123               8        0

                                             Caption  \
0  Here are some of the most important data visua...
1  Here are some of the best data science project...
2  Learn how to train a machine learning model an...
3  Here s how you can write a Python program to d...
4  Plotting annotations while visualizing your da...

                                            Hashtags
0  #finance #money #business #investing #investme...
1  #healthcare #health #covid #data #datascience ...
2  #data #datascience #dataanalysis #dataanalytic...
3  #python #pythonprogramming #pythonprojects #py...
4  #datavisualization #datascience #data #dataana...
```

In [3]:
```python
#check if the data has null values
data.isnull().sum()
```

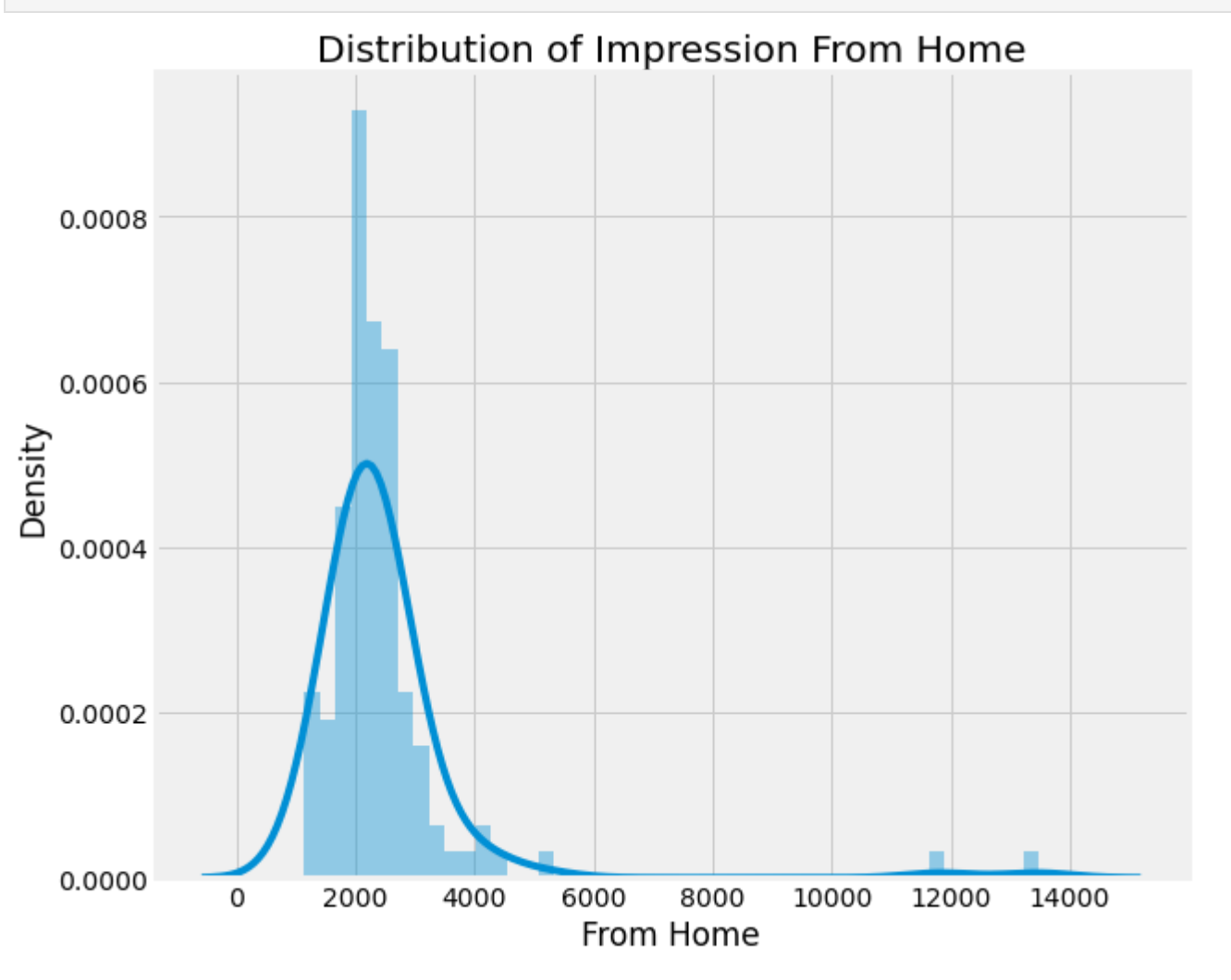
Out[3]:
```
Impressions       0
From Home         0
From Hashtags     0
From Explore      0
From Other        0
Saves             0
Comments          0
Shares            0
Likes             0
Profile Visits    0
Follows           0
Caption           0
Hashtags          0
dtype: int64
```

In [4]:
```python
#Check data type
data.info()
```

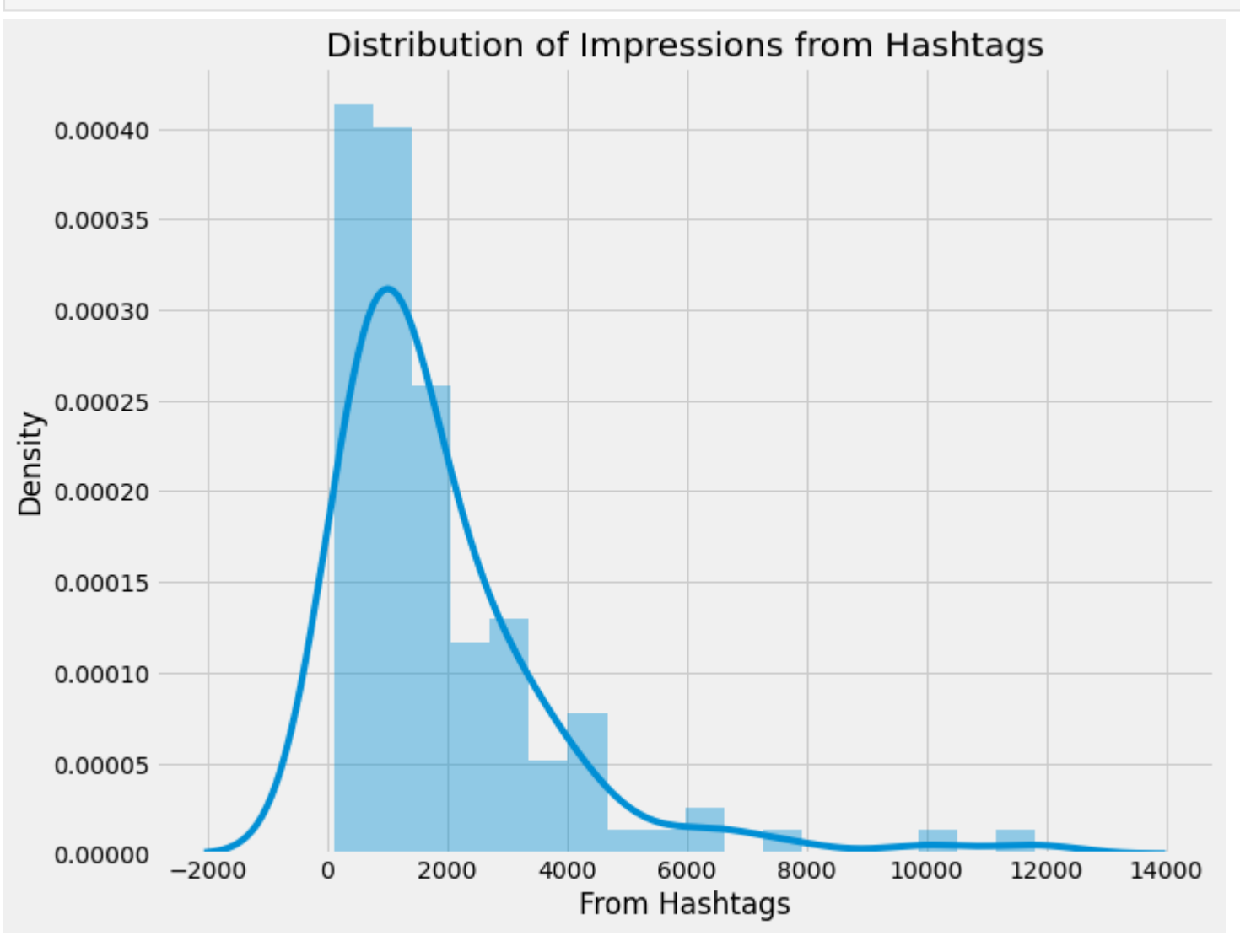
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119 entries, 0 to 118
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Impressions     119 non-null    int64
 1   From Home       119 non-null    int64
 2   From Hashtags   119 non-null    int64
 3   From Explore    119 non-null    int64
 4   From Other      119 non-null    int64
 5   Saves           119 non-null    int64
 6   Comments        119 non-null    int64
 7   Shares          119 non-null    int64
 8   Likes           119 non-null    int64
 9   Profile Visits  119 non-null    int64
 10  Follows         119 non-null    int64
 11  Caption         119 non-null    object
 12  Hashtags        119 non-null    object
dtypes: int64(11), object(2)
memory usage: 12.2+ KB
```

ANALYZING THE REACH OF MY INSTAGRAM POSTS

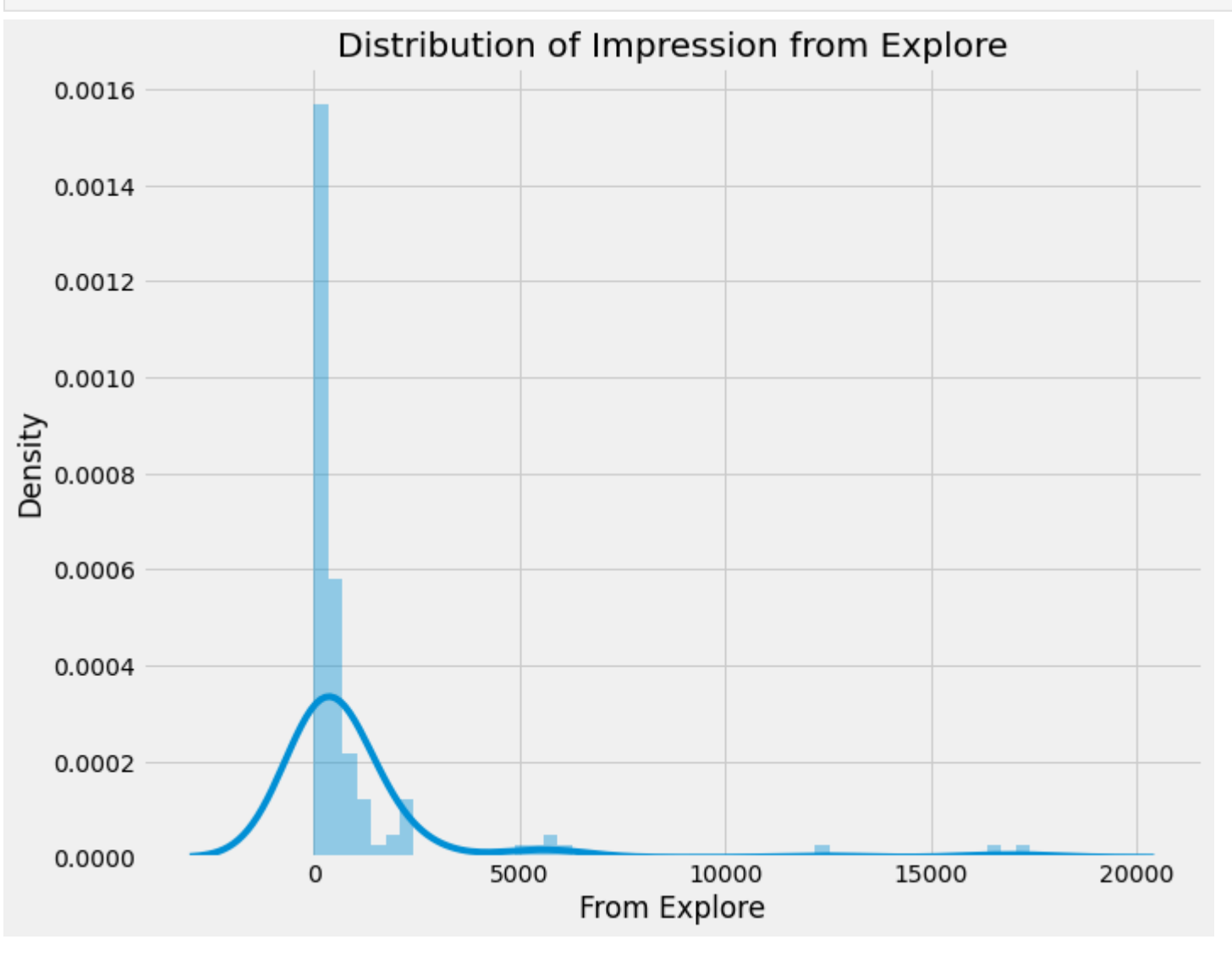
In [5]:
```python
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
#Analyzing impression from Home
plt.figure(figsize=(10,8))
plt.style.use('fivethirtyeight')
plt.title("Distribution of Impression From Home")
sns.distplot(data['From Home'])
plt.show()
```



In [6]:
```python
#Analyzing impression from Hashtags
plt.figure(figsize = (10,8))
plt.title("Distribution of Impressions from Hashtags")
sns.distplot(data['From Hashtags'])
plt.show()
```



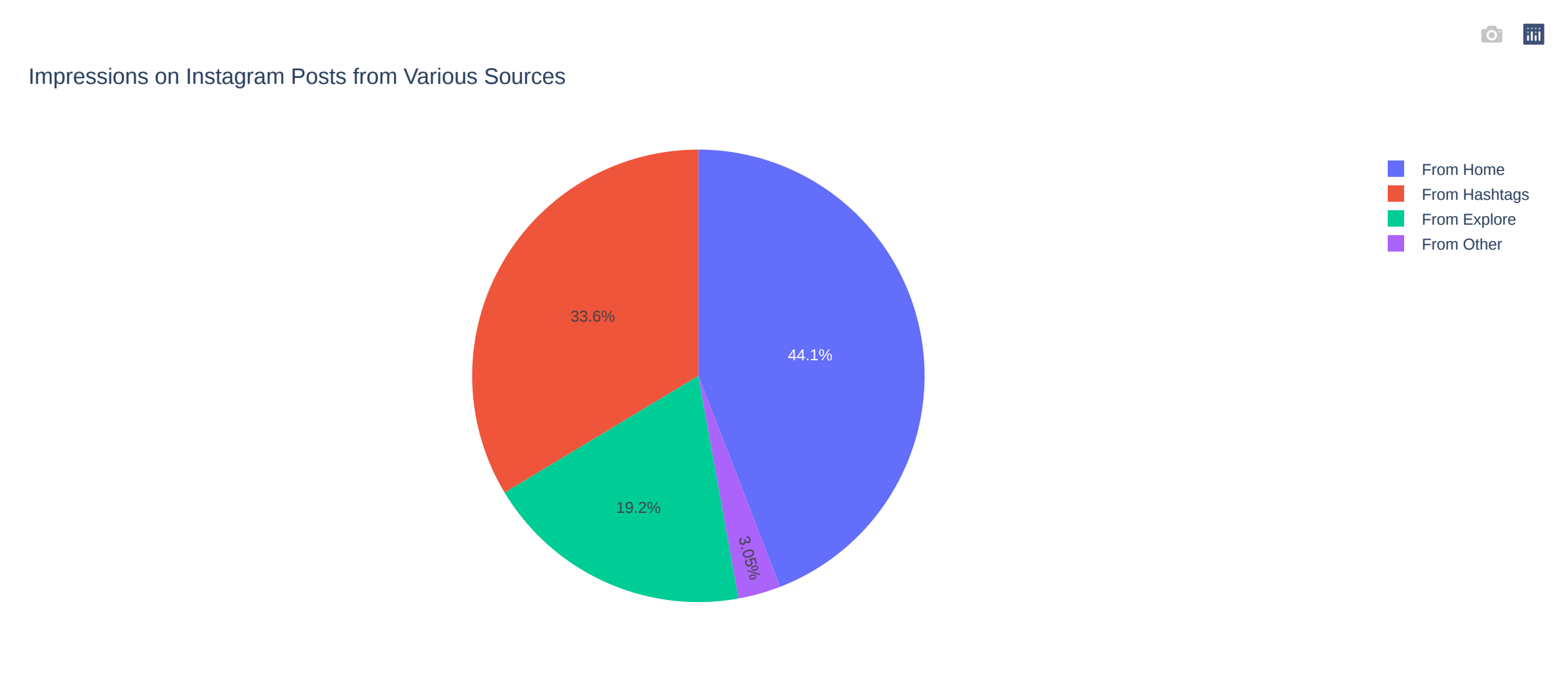
In [7]:
```python
#Analyzing impressions from explore section
plt.figure(figsize =(10,8))
plt.title("Distribution of Impression from Explore")
sns.distplot(data['From Explore'])
plt.show()
```



In [8]:
```python
#Check the percentage of impressions I get from various sources of Instagram
home = data["From Home"].sum()
hashtags = data["From Hashtags"].sum()
explore = data["From Explore"].sum()
other = data["From Other"].sum()

labels = ['From Home','From Hashtags','From Explore','From Other']
values = [home,hashtags,explore,other]

fig = px.pie(data, values=values, names=labels,
             title='Impressions on Instagram Posts from Various Sources')
fig.show()
```

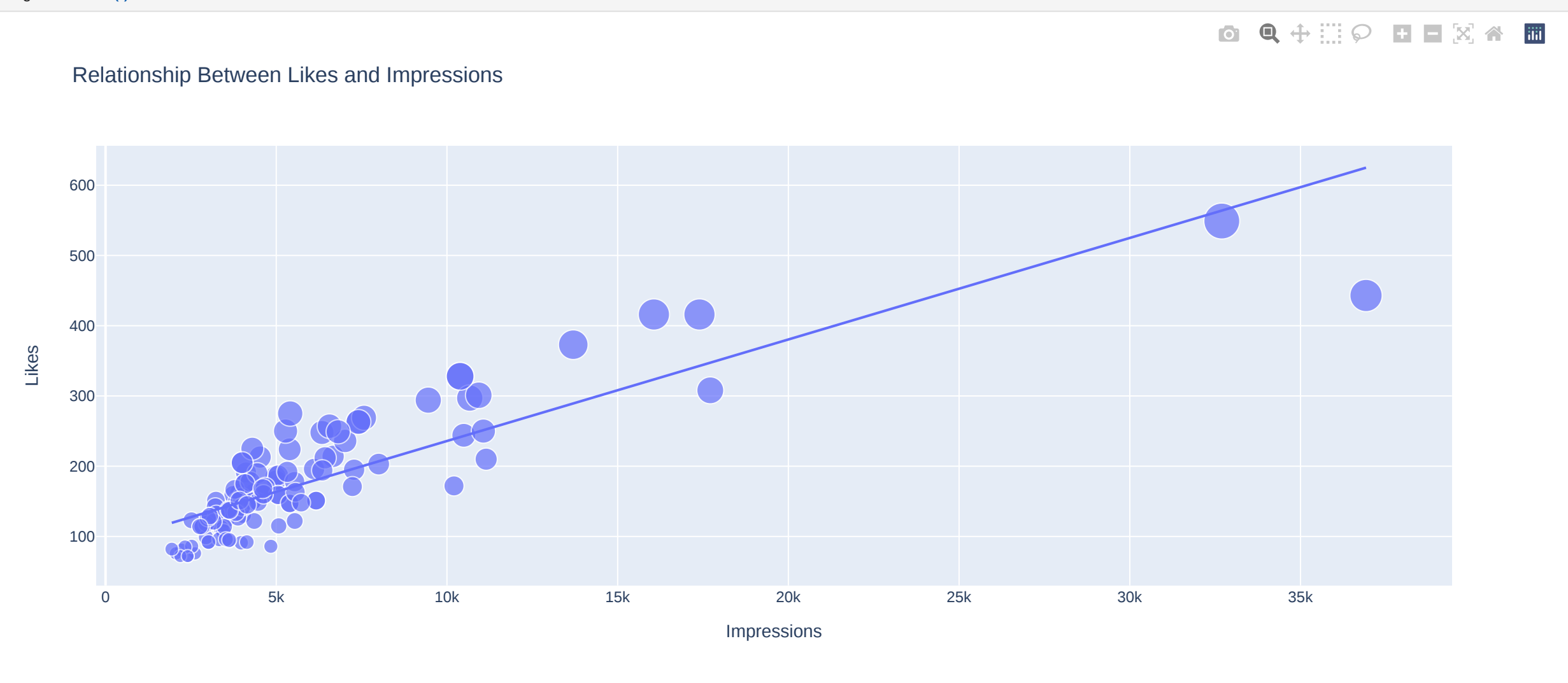


Observations:

- 44.1% of the reach is from my followers
- 33.6% is from hashtags
- 19.2% is from explore
- 3.05% is from other sources

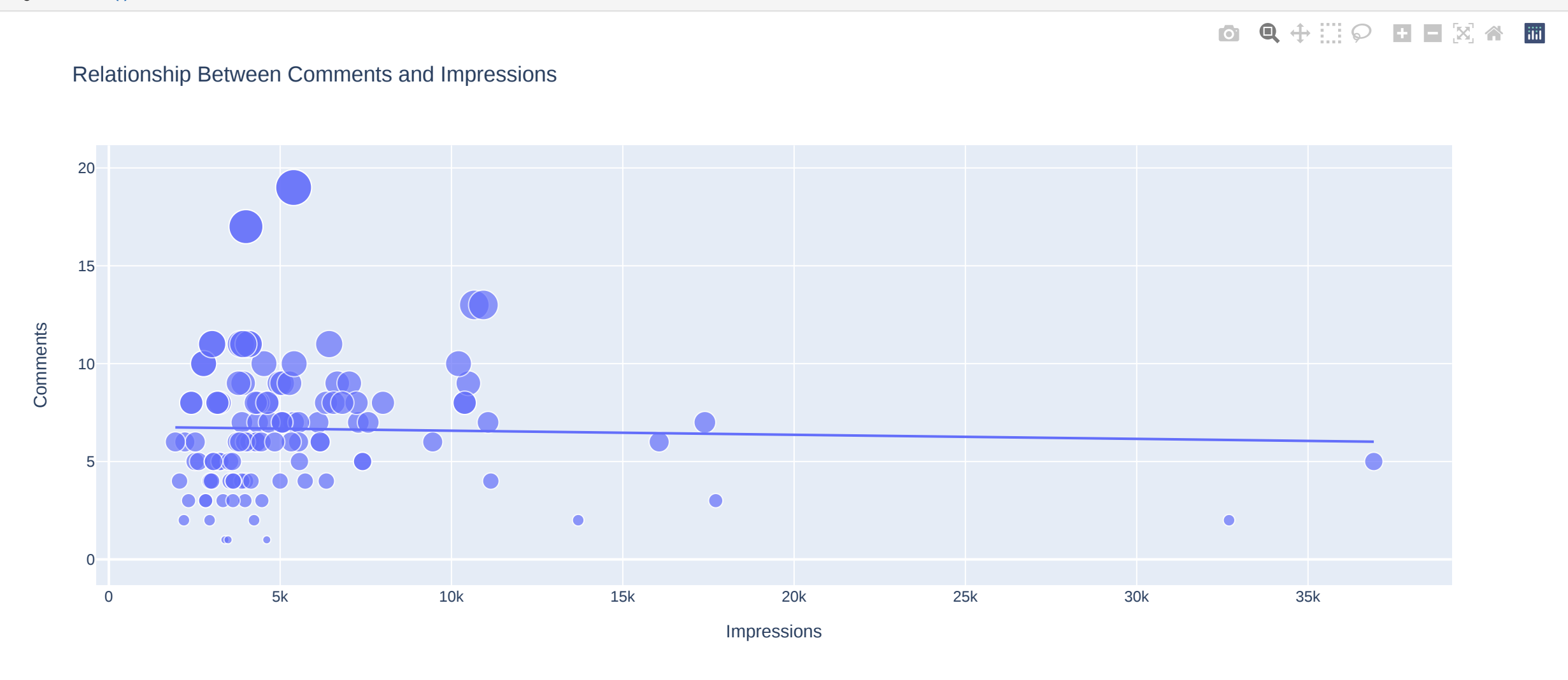
Analyzing Relationships

In [9]:
```python
#Relationship between the no of likes and impressions
figure = px.scatter(data_frame=data,x="Impressions",
                    y="Likes",size="Likes",trendline="ols",
                    title="Relationship Between Likes and Impressions")
figure.show()
```



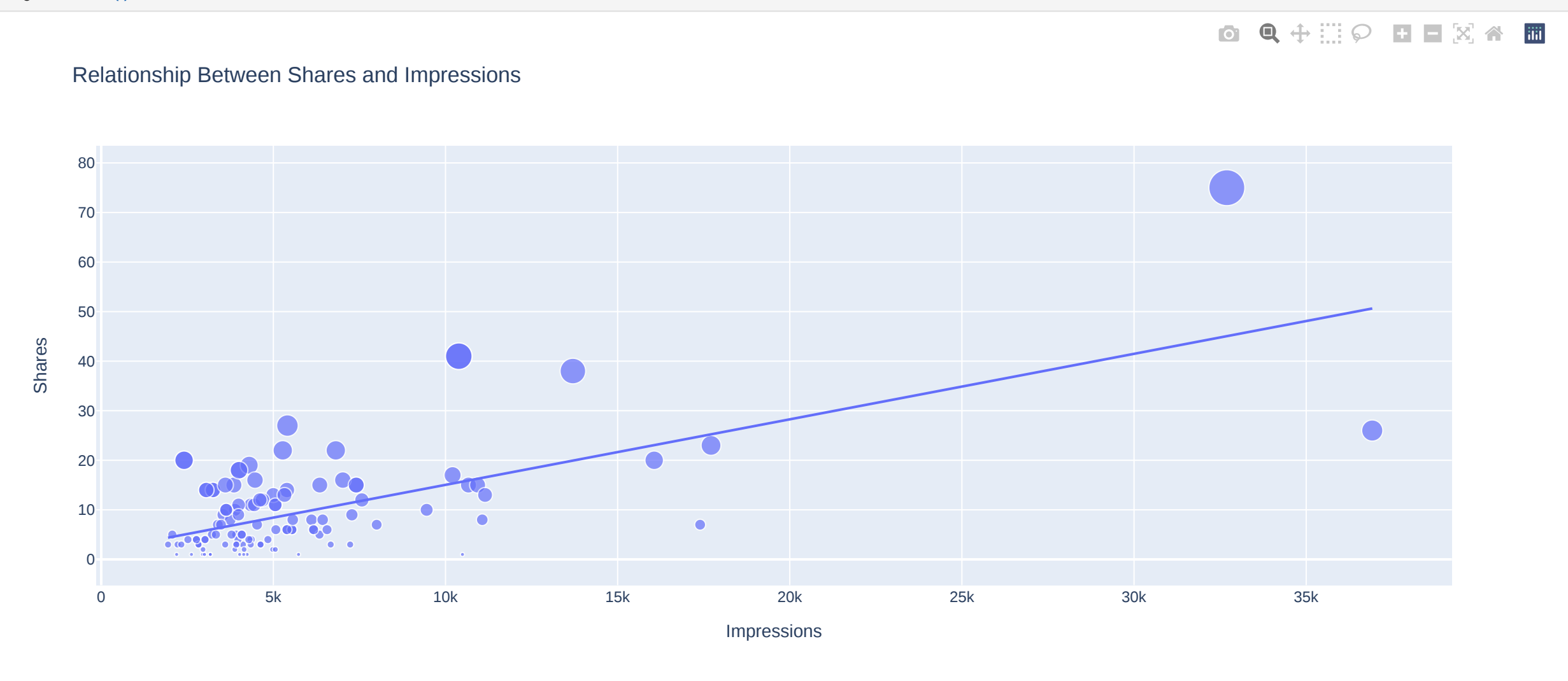
Observation: There is a linear relationship between the number of likes and impressions on my instagram posts

In [10]:
```python
#Relationship between comments and impressions
figure = px.scatter(data_frame=data,x="Impressions",
                    y="Comments",size="Comments",trendline="ols",
                    title="Relationship Between Comments and Impressions")
figure.show()
```



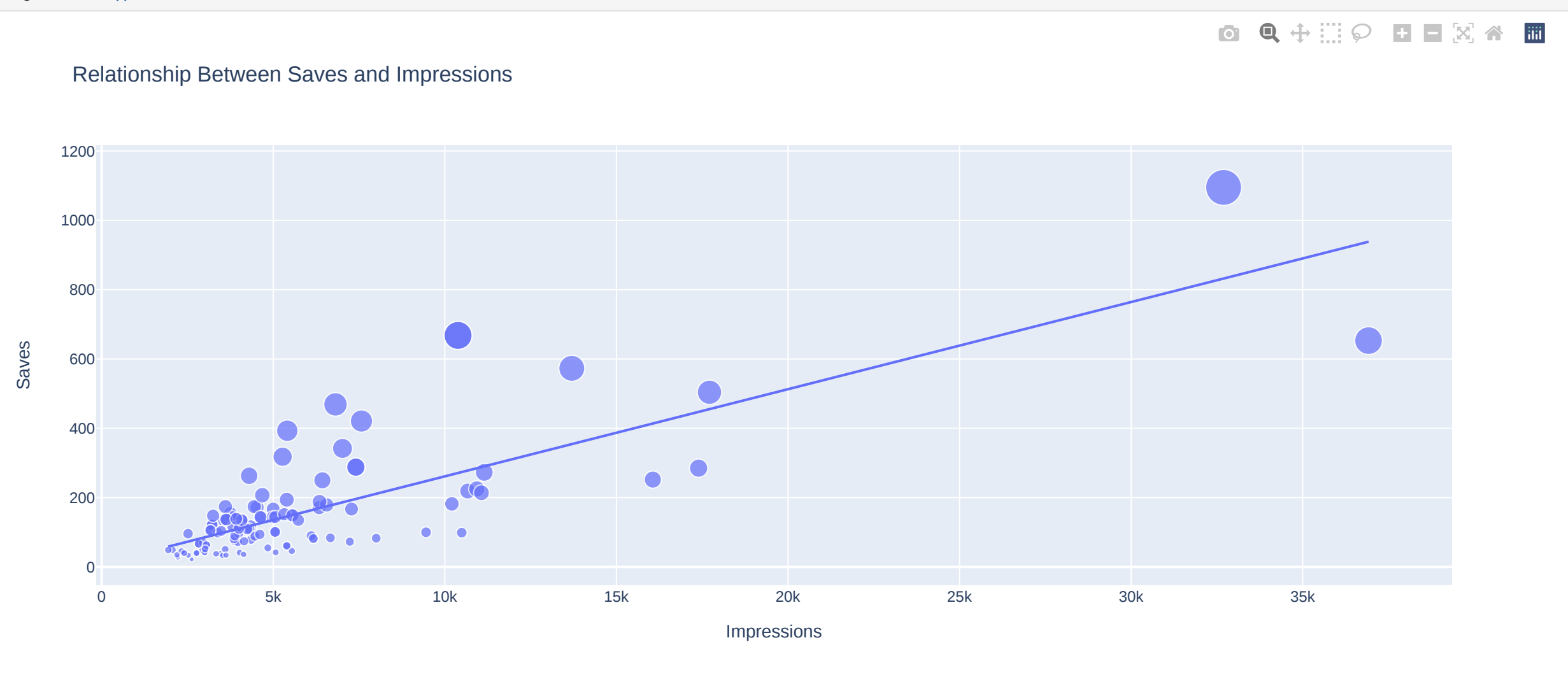
Observation: The number of comments does not affect its reach

In [11]:
```python
#Relationship between the no of shares and impression
figure = px.scatter(data_frame=data,x="Impressions",
                    y="Shares",size="Shares",trendline="ols",
                    title="Relationship Between Shares and Impressions")
figure.show()
```



Observation: The more number of shares will result to a higher reach but shares does not affect the number of reach as much as likes do

In [12]:
```python
#Relationship between saves and impressions
figure = px.scatter(data_frame=data,x="Impressions",
                    y="Saves",size="Saves",trendline="ols",
                    title="Relationship Between Saves and Impressions")
figure.show()
```



Observation: There is a linear relationship between the number of times my post is saved and the reach of my instagram post

In [13]:
```python
#Let's check the correlation of all columns with the impression columns
correlation = data.corr()
print(correlation["Impressions"].sort_values(ascending=False))
```

```
Impressions       1.000000
From Explore      0.893607
Follows           0.889363
Likes             0.849835
From Home         0.844698
Saves             0.779231
Profile Visits    0.760981
Shares            0.634675
From Other        0.592960
From Hashtags     0.560760
Comments         -0.028524
Name: Impressions, dtype: float64
```

Observation: The more likes and saves will help you get more reach on Instagram and the number of shares will not affect your reach

Analyzing Conversion Rates

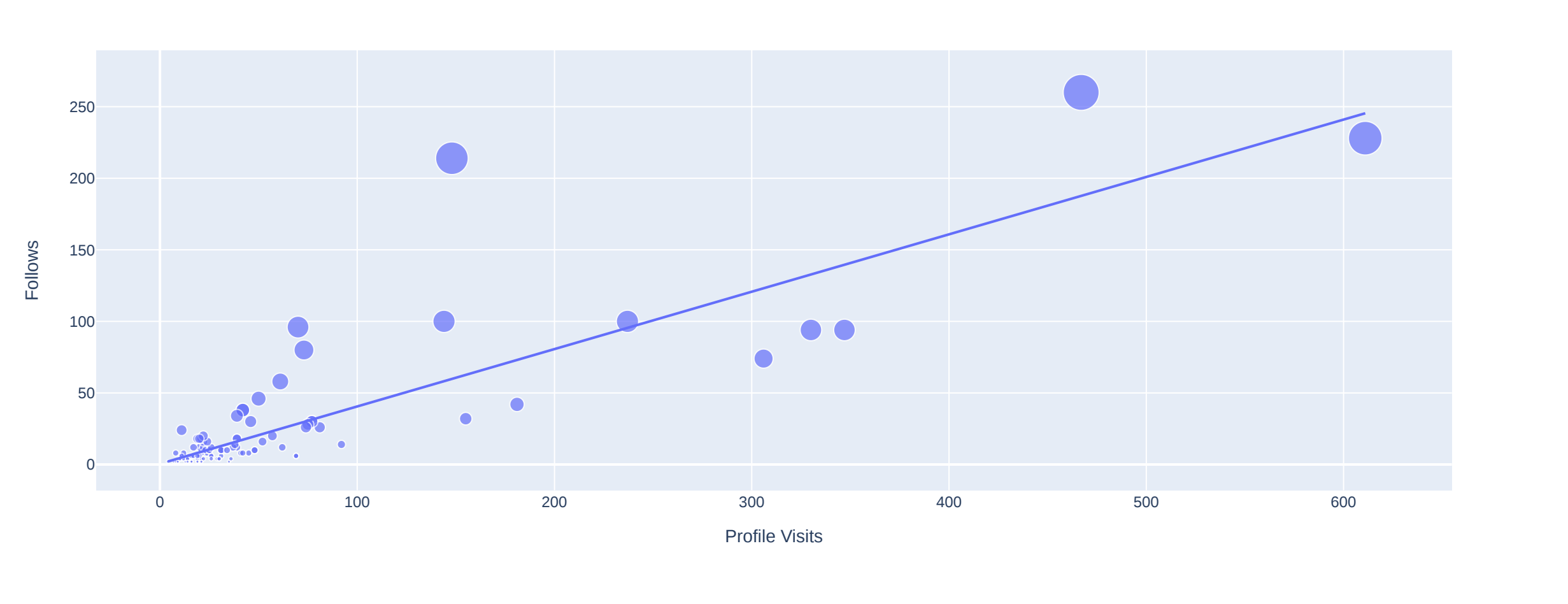
In instagram conversion rates means how many followers you are getting from the number of profile visits from a post. I calculated it by FOLLOWS/PROFILE VISITS * 100

In [14]:
```python
conversation_rate = (data["Follows"].sum()/data["Profile Visits"].sum())*100
print(conversation_rate)
```

```
41.00265604249668
```

Observation: So my conversation rate is 41% which sounds like a very good conversation rate!

In [15]:
```python
#We look at the relationship between the total profile visits and the number of followers gained from all profile visits
figure = px.scatter(data_frame = data, x = "Profile Visits", y = "Follows", size = "Follows", trendline = "ols",
                    title = "Relationship between Profile Visits and Followers Gained")
figure.show()
```



Observation: The relationship between Profile Visits and followers gained is also linear

Instagram Reach Prediction Model

In [16]:
```python
#Split the data into training and test sets before training the model
x = np.array(data[['Likes', 'Saves', 'Comments','Shares',
                   'Profile Visits','Follows']])
y = np.array(data["Impressions"])
xtrain, xtest, ytrain, ytest = train_test_split(x,y,
                                                test_size = 0.2,
                                                random_state = 42)
```

In [17]:
```python
model = LinearRegression()
model.fit(xtrain, ytrain)
model.score(xtest, ytest)
```

Out[17]:
```
0.8777977785012778
```

Lets predict the reach of an Instagram post by giving inputs to the model

In [18]:
```python
#Features = [['Likes','Saves', 'Comments', 'Shares', 'Profile Visits','Follows']]
features = np.array([[282.0, 233.0, 4.0, 9.0, 165.0, 54.0]])
model.predict(features)
```

Out[18]:
```
array([11139.58239766])
```

In [ ]: