In [2]:
```
#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
from sklearn.model_selection import train_test_split
from sklearn.linear_model import PassiveAggressiveRegressor
```

In [3]:
```
#reading the data from dataset
ds = pd.read_csv("C:\\Dataset\\Instagram Reach Analysis\\Instagram_data.csv", encoding = 'latin1')
```

In [4]:
```
#printing the dataset
print(ds.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 [5]:
```
#checking whether dataset contains null values or not
ds.isnull().sum()
```

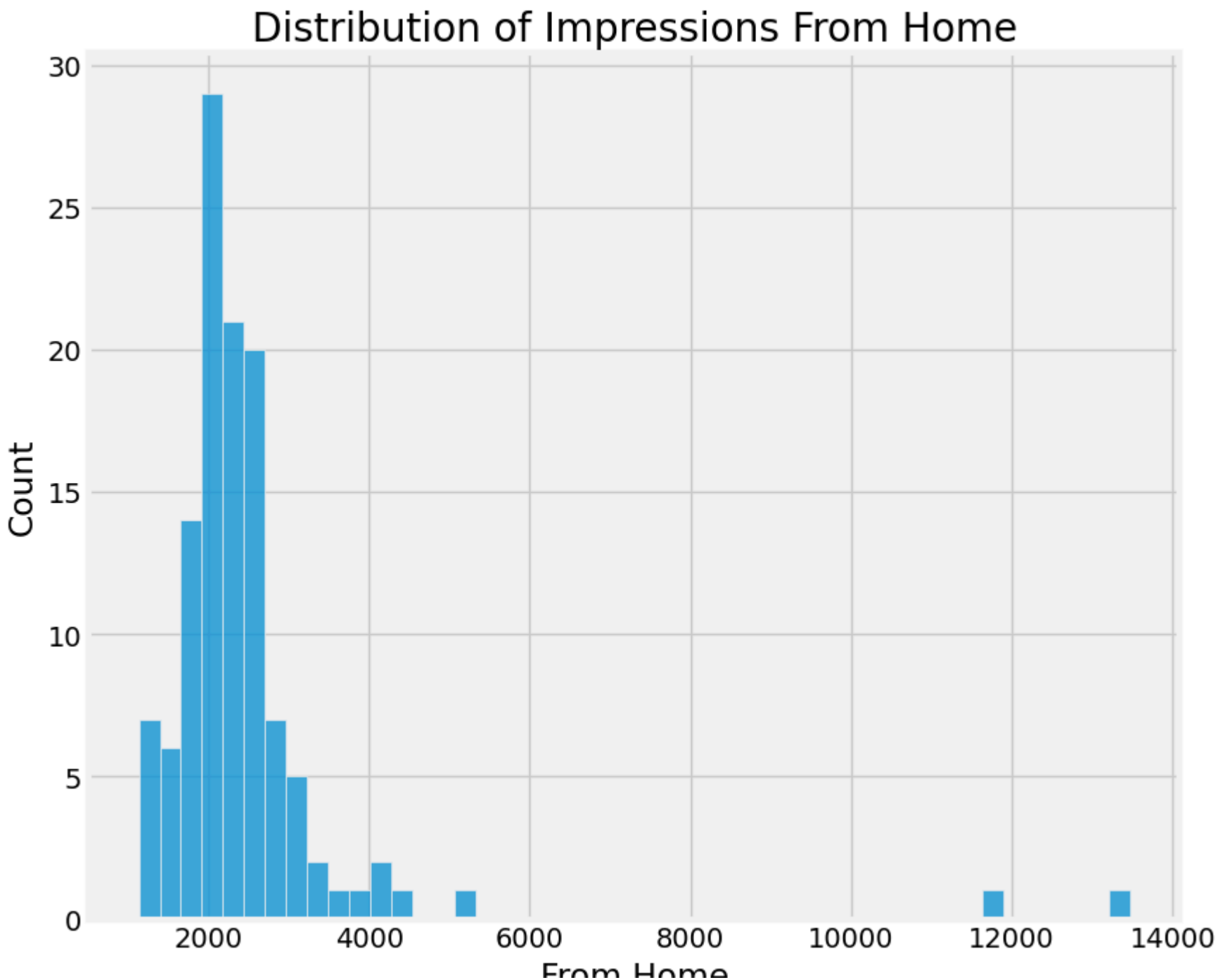
Out[5]:
```
Impressions       0
From Home         0
From Hashtags     0
From Explore      0
From Other        0
Saves             0
Comments          0
Shares            0
Likes             0
Profile Visits    0
Follows           0
Caption           0
Hashtags          0
dtype: int64
```

In [6]:
```
ds = ds.dropna()
```

In [7]:
```
#knowing the datatypes of cloumns
ds.info()
```

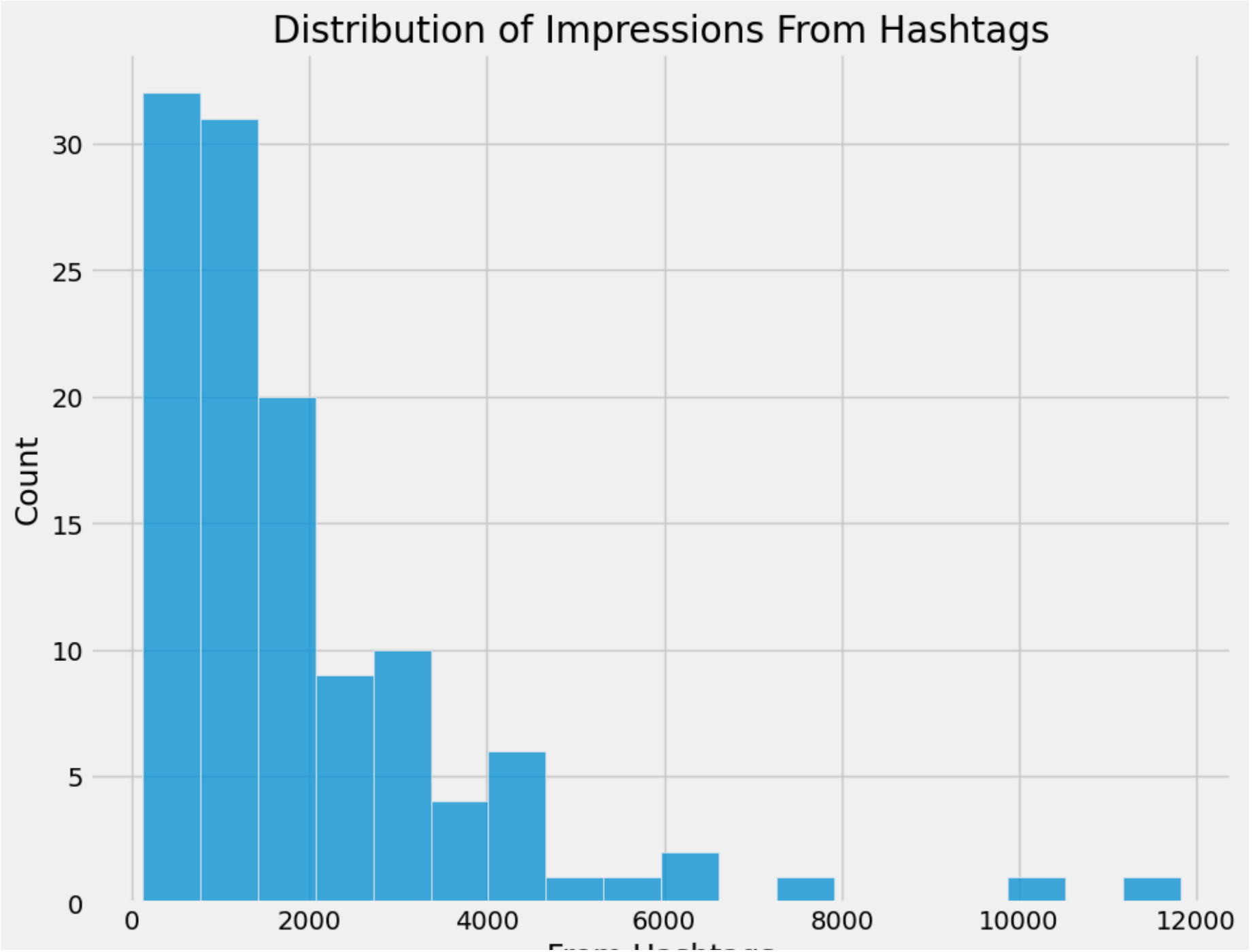
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119 entries, 0 to 118
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Impressions     119 non-null    int64
 1   From Home       119 non-null    int64
 2   From Hashtags   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
```

In [8]:
```
#starting the analysis by first looking at the distribution of impressions from home
plt.figure(figsize=(10, 8))
plt.style.use('fivethirtyeight')
plt.title("Distribution of Impressions From Home")
sns.histplot(ds['From Home'])
plt.show()
```


Distribution of Impressions From Home
Count
0
5
10
15
20
25
30
2000
4000
6000
8000
10000
12000
14000
From Home

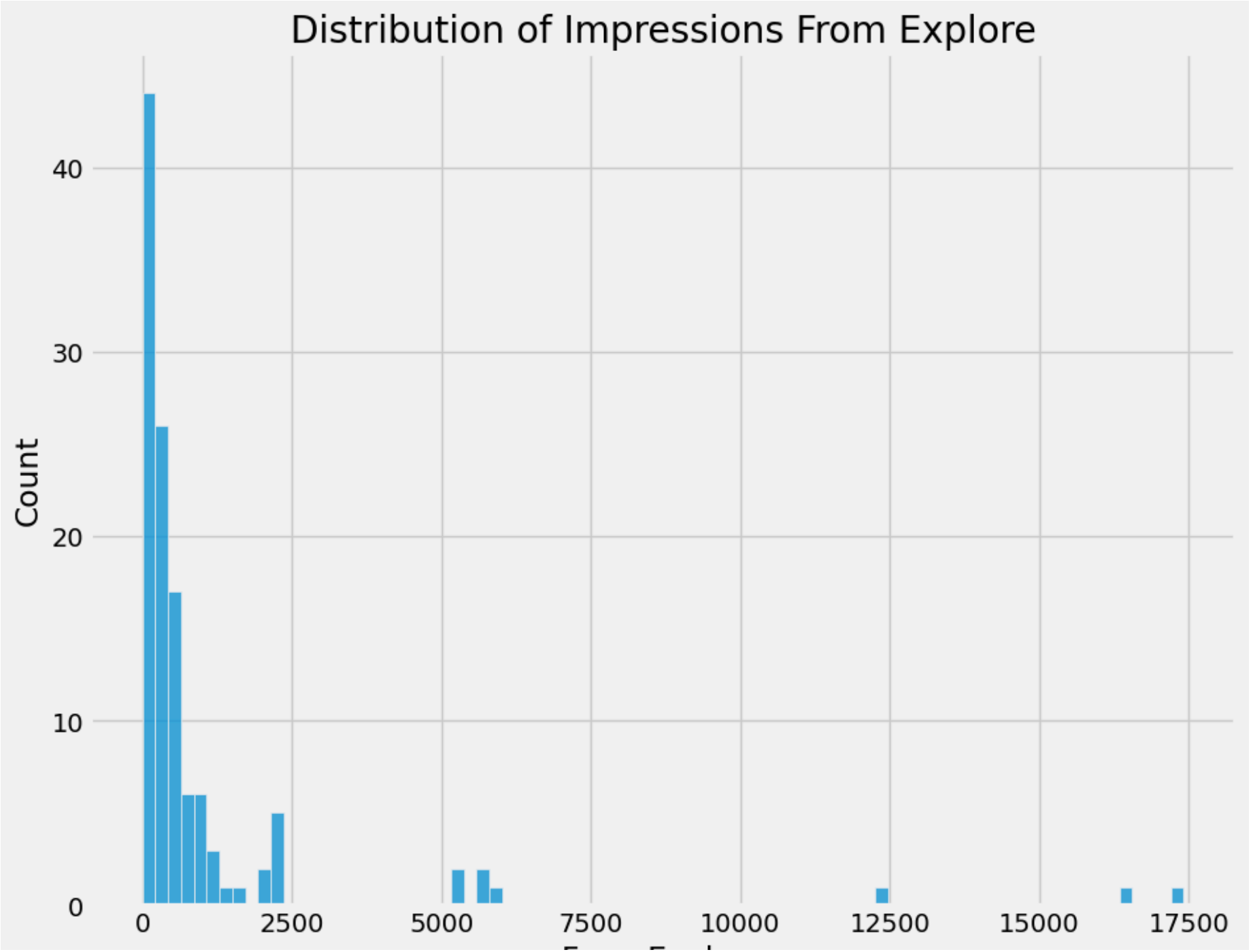
From Home

In [9]:
```
#now the analysis of impression by hashtags
plt.figure(figsize=(10, 8))
plt.title("Distribution of Impressions From Hashtags")
sns.histplot(ds['From Hashtags'])
plt.show()
```


Distribution of Impressions From Hashtags
Count
0
5
10
15
20
25
30
0
2000
4000
6000
8000
10000
12000

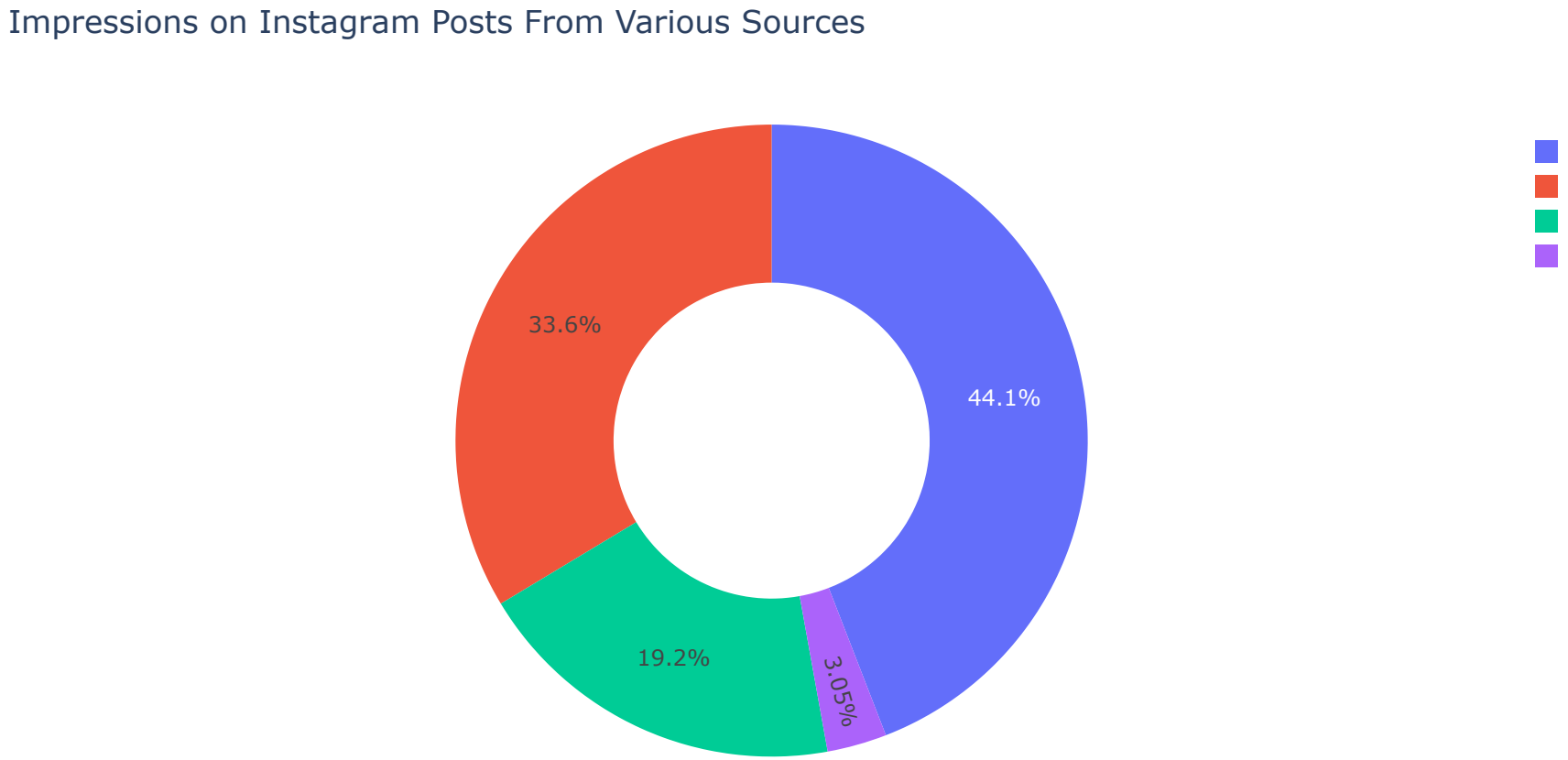
From Hashtags

In [10]:
```
#now the analysis of distribution of expression from explore
plt.figure(figsize=(10, 8))
plt.title("Distribution of Impressions From Explore")
sns.histplot(ds['From Explore'])
plt.show()
```


Distribution of Impressions From Explore
Count
0
10
20
30
40
0
2500
5000
7500
10000
12500
15000
17500

From Explore

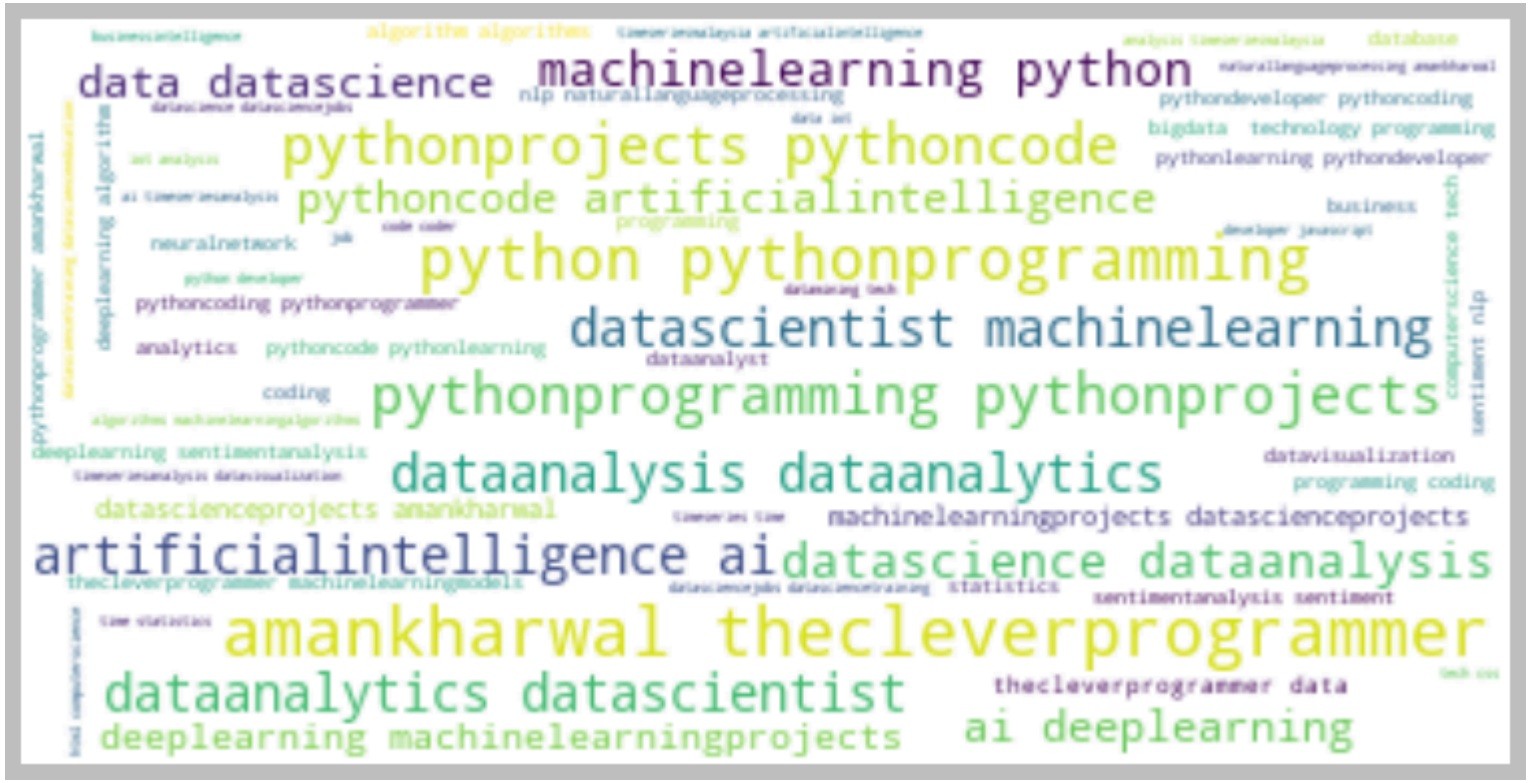
In [11]:
```
#Looking at the percentage of impressions getting on instagram
home = ds["From Home"].sum()
hashtags = ds["From Hashtags"].sum()
explore = ds["From Explore"].sum()
other = ds["From Other"].sum()

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

fig = px.pie(ds, values=values, names=labels,
             title='Impressions on Instagram Posts From Various Sources', hole=0.5)
fig.show()
```



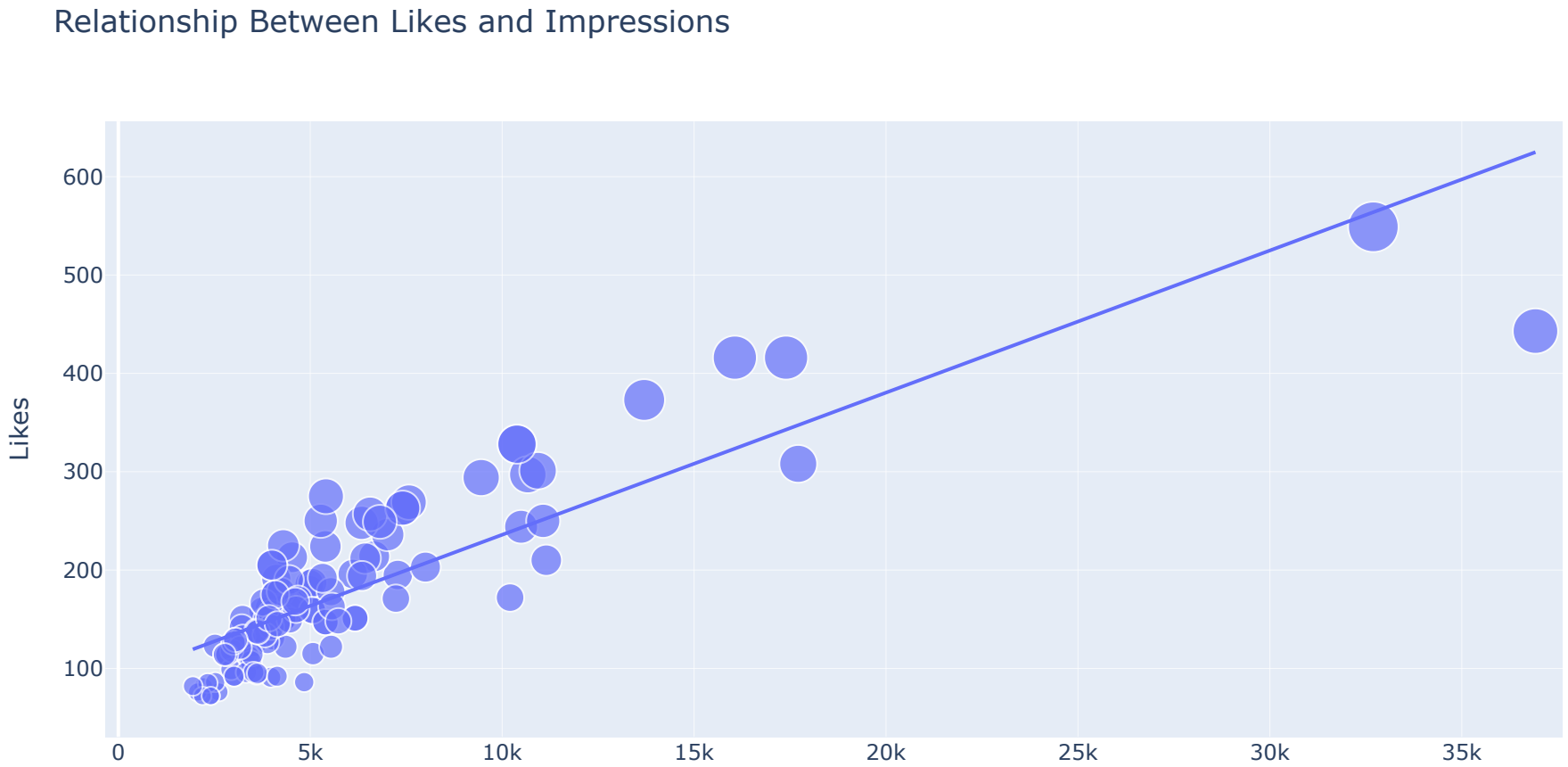
In [12]:
```
#creating a wordcloud of the caption column to look at the most used words in the caption
text = " ".join(i for i in ds.Caption)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords, background_color="white").generate(text)
plt.style.use('classic')
plt.figure( figsize=(12,10))
plt.imshow(wordcloud, interpolation='bilinear')
```

```
plt.axis("off")
plt.show()
```

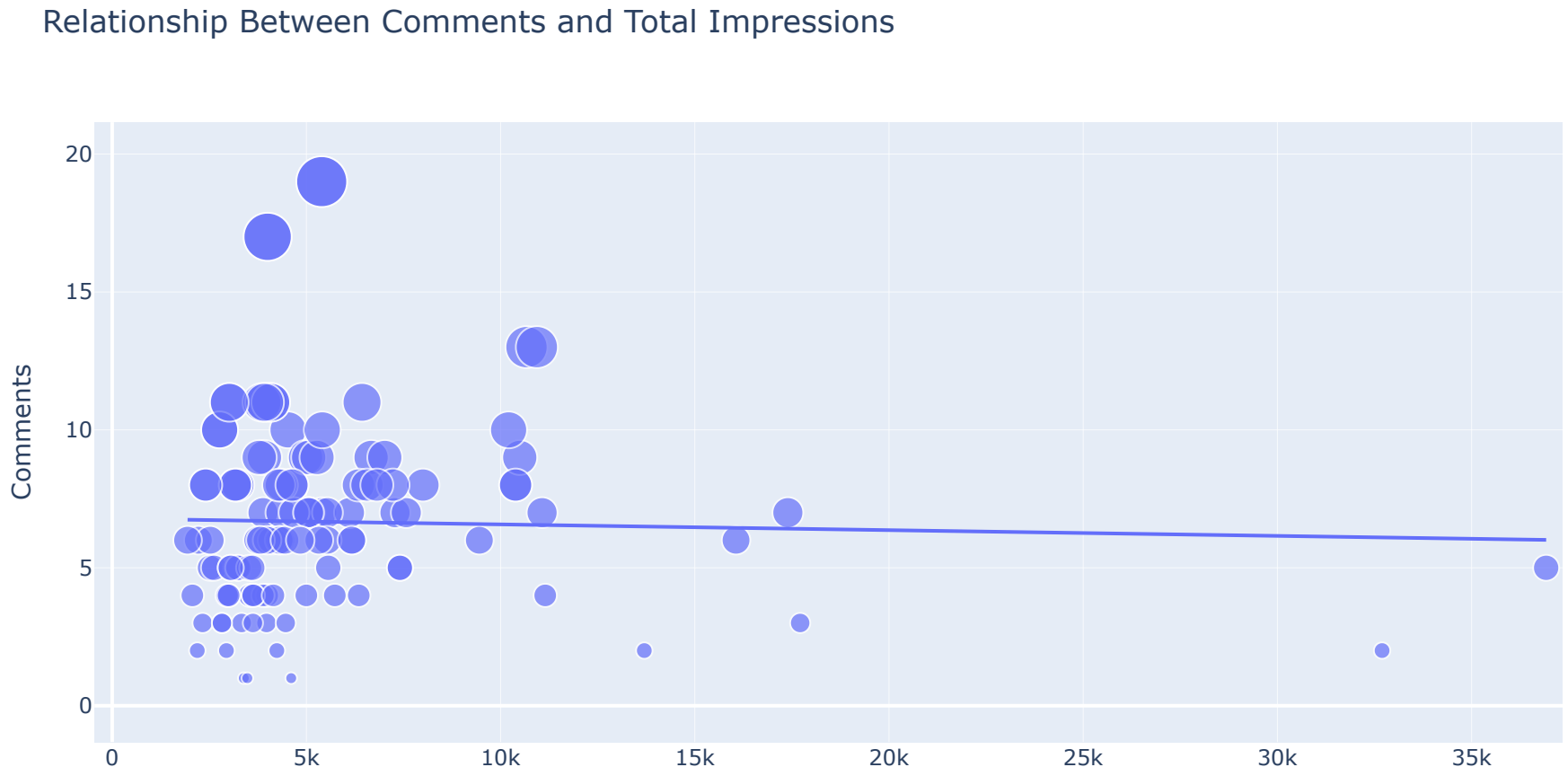
In [13]:
```
#creating a wordcloud of the hastags column to look at the most used hastags in the caption
text = " ".join(i for i in ds.Hashtags)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords, background_color="white").generate(text)
plt.figure( figsize=(12,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



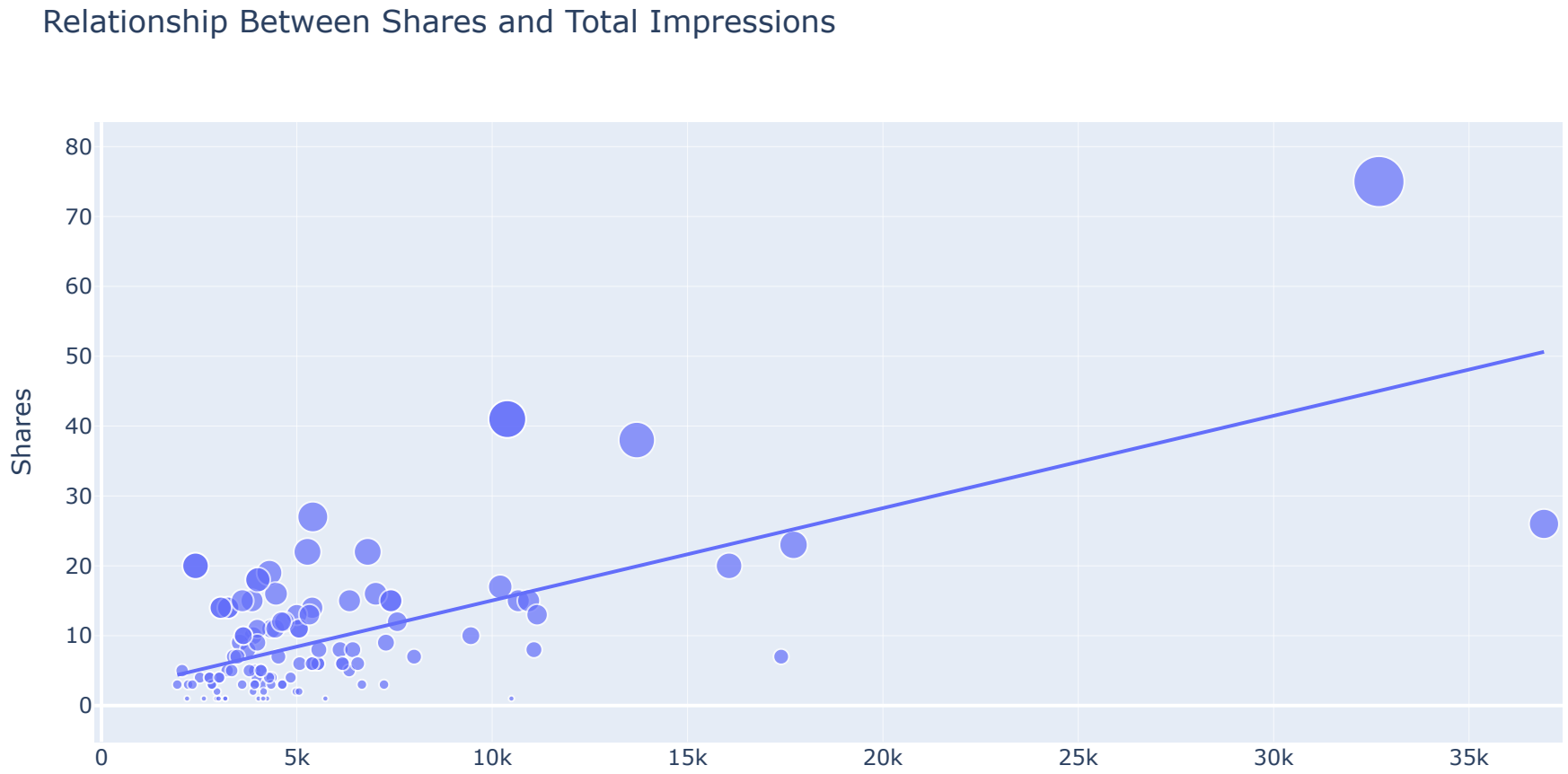
In [14]:
```
#analyzing the relationship between the number of likes and the number of impressions
figure = px.scatter(data_frame = ds, x="Impressions",
                    y="Likes", size="Likes", trendline="ols",
                    title = "Relationship Between Likes and Impressions")
figure.show()
```



In [15]:
```
#analyzing the relationship between the number of comments and the number of impressions
figure = px.scatter(data_frame = ds, x="Impressions",
                    y="Comments", size="Comments", trendline="ols",
                    title = "Relationship Between Comments and Total Impressions")
figure.show()
```



In [16]:
```
#analyzing the relationship between the number of shares and the number of impressions
figure = px.scatter(data_frame = ds, x="Impressions",
                    y="Shares", size="Shares", trendline="ols",
                    title = "Relationship Between Shares and Total Impressions")
figure.show()
```



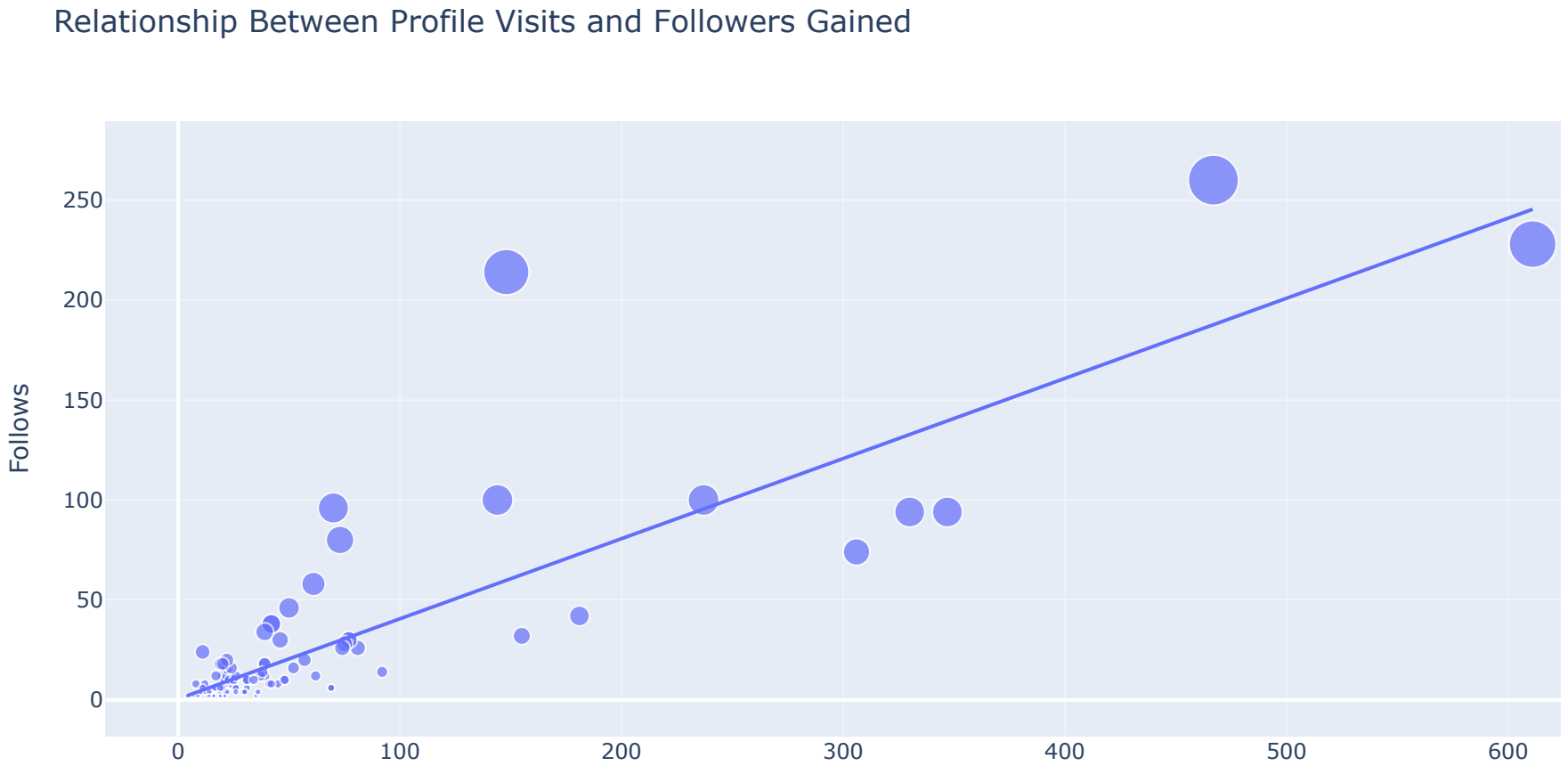
In [17]:
```
#looking at the correlation of all columns with impressions columns
numeric_ds = ds.select_dtypes(include=['number'])
correlation = numeric_ds.corr()
print(correlation["Impressions"].sort_values(ascending=False))
```

```
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
```

In [18]:
```
#analyzing converson rate
conversion_rate = (ds["Follows"].sum() / ds["Profile Visits"].sum()) * 100
print(conversion_rate)
```

```
41.00265604249668
```

In [19]:
```
#relationship between total profile visits and number of followers gained
figure = px.scatter(data_frame = ds, x="Profile Visits",
                    y="Follows", size="Follows", trendline="ols",
                    title = "Relationship Between Profile Visits and Followers Gained")
figure.show()
```



In [20]:
```
#The relationship between profile visits and followers gained is also linear.
```

In [45]:
```
#prediction model
x = np.array(ds[['Likes', 'Saves', 'Comments', 'Shares',
                 'Profile Visits', 'Follows']])
y = np.array(ds["Impressions"])
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=42)
```

In [53]:
```
model = PassiveAggressiveRegressor()
model.fit(xtrain, ytrain)
model.score(xtest, ytest)
```

Out[53]:
```
0.9075823172559758
```

In [54]:
```
#predicting the reach of an instagram post
# 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[54]:
```
array([12281.29385189])
```

In [24]:
```
#this is how we can analyze and predict the reach of instagram posts with machine learning using python.
```