Website Traffic Analysis

Importing required libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in c:\users\sanjay pk\anaconda3\lib\site-
packages (1.9.2)
Requirement already satisfied: pillow in c:\users\sanjay pk\anaconda3\lib\site-pac
kages (from wordcloud) (9.2.0)
Requirement already satisfied: matplotlib in c:\users\sanjay pk\anaconda3\lib\site
-packages (from wordcloud) (3.5.2)
Requirement already satisfied: numpy>=1.6.1 in c:\users\sanjay pk\anaconda3\lib\si
te-packages (from wordcloud) (1.21.5)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\sanjay pk\anaconda
3\lib\site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\sanjay pk\anaconda3\lib\si
te-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\sanjay pk\anaconda3\l
ib\site-packages (from matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\sanjay pk\anaconda3\l
ib\site-packages (from matplotlib->wordcloud) (1.4.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\sanjay pk\anaconda3\li
b\site-packages (from matplotlib->wordcloud) (3.0.9)
Requirement already satisfied: packaging>=20.0 in c:\users\sanjay pk\anaconda3\lib
\site-packages (from matplotlib->wordcloud) (21.3)
Requirement already satisfied: six>=1.5 in c:\users\sanjay pk\anaconda3\lib\site-p
ackages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
```

In [3]:
```
import os
from wordcloud import WordCloud
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
```

Importing the Dataset

In [4]:
```
data = pd.read_csv('Documents/daily-website-visitors.csv')
data
```

Out[4]:

	Row	Day	Day.Of.Week	Date	Page.Loads	Unique.Visits	First.Time.Visits	Retu
0	1	Sunday	1	9/14/2014	2,146	1,582	1,430	
1	2	Monday	2	9/15/2014	3,621	2,528	2,297	
2	3	Tuesday	3	9/16/2014	3,698	2,630	2,352	
3	4	Wednesday	4	9/17/2014	3,667	2,614	2,327	
4	5	Thursday	5	9/18/2014	3,316	2,366	2,130	
...	...	...	...	...	...	...	...	
2162	2163	Saturday	7	8/15/2020	2,221	1,696	1,373	
2163	2164	Sunday	1	8/16/2020	2,724	2,037	1,686	
2164	2165	Monday	2	8/17/2020	3,456	2,638	2,181	
2165	2166	Tuesday	3	8/18/2020	3,581	2,683	2,184	
2166	2167	Wednesday	4	8/19/2020	2,064	1,564	1,297	

2167 rows × 8 columns

Droping the columns

In [7]:
```
data = data.drop(columns=['Date'])
data
```

Out[7]:

	Row	Day	Day.Of.Week	Page.Loads	Unique.Visits	First.Time.Visits	Returning.Visits
0	1	Sunday	1	2,146	1,582	1,430	152
1	2	Monday	2	3,621	2,528	2,297	231
2	3	Tuesday	3	3,698	2,630	2,352	278
3	4	Wednesday	4	3,667	2,614	2,327	287
4	5	Thursday	5	3,316	2,366	2,130	236
...	...	...	...	...	...	...	...
2162	2163	Saturday	7	2,221	1,696	1,373	323
2163	2164	Sunday	1	2,724	2,037	1,686	351
2164	2165	Monday	2	3,456	2,638	2,181	457
2165	2166	Tuesday	3	3,581	2,683	2,184	499
2166	2167	Wednesday	4	2,064	1,564	1,297	267

2167 rows × 7 columns

Checking the dataset

In [8]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2167 entries, 0 to 2166
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Row                2167 non-null   int64 
 1   Day                2167 non-null   object
 2   Day.Of.Week        2167 non-null   int64 
 3   Page.Loads         2167 non-null   object
 4   Unique.Visits      2167 non-null   object
 5   First.Time.Visits  2167 non-null   object
 6   Returning.Visits   2167 non-null   object
dtypes: int64(2), object(5)
memory usage: 118.6+ KB
```

Checking the missing values

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

Out[9]:
```
Row                 0
Day                 0
Day.Of.Week         0
Page.Loads          0
Unique.Visits       0
First.Time.Visits   0
Returning.Visits    0
dtype: int64
```

In [13]:
```
data['Day.Of.Week'] = data['Page.Loads'].apply(lambda x: x.split('-')[0])
data
```

Out[13]:

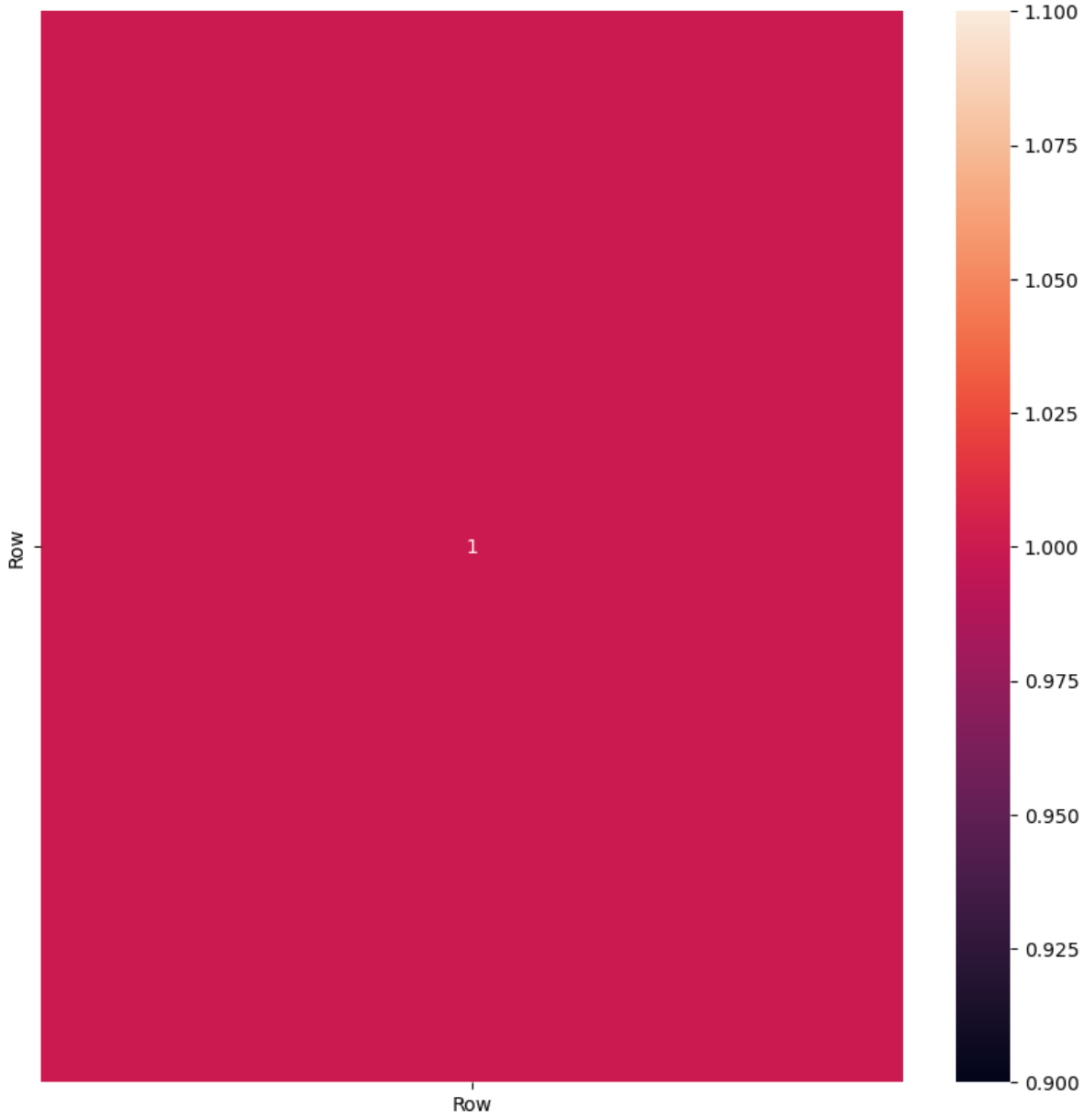
	Row	Day	Day.Of.Week	Page.Loads	Unique.Visits	First.Time.Visits	Returning.Visits
0	1	Sunday	2,146	2,146	1,582	1,430	152
1	2	Monday	3,621	3,621	2,528	2,297	231
2	3	Tuesday	3,698	3,698	2,630	2,352	278
3	4	Wednesday	3,667	3,667	2,614	2,327	287
4	5	Thursday	3,316	3,316	2,366	2,130	236
...	...	...	...	...	...	...	...
2162	2163	Saturday	2,221	2,221	1,696	1,373	323
2163	2164	Sunday	2,724	2,724	2,037	1,686	351
2164	2165	Monday	3,456	3,456	2,638	2,181	457
2165	2166	Tuesday	3,581	3,581	2,683	2,184	499
2166	2167	Wednesday	2,064	2,064	1,564	1,297	267

2167 rows × 7 columns

Visualizing the Datasets using various Graphs

In [14]:
```
plt.figure(figsize=(10,10))
sns.heatmap(data.corr(),annot=True)
```

Out[14]:
```
<AxesSubplot:>
```



In [15]:
```
data.corr().T
```

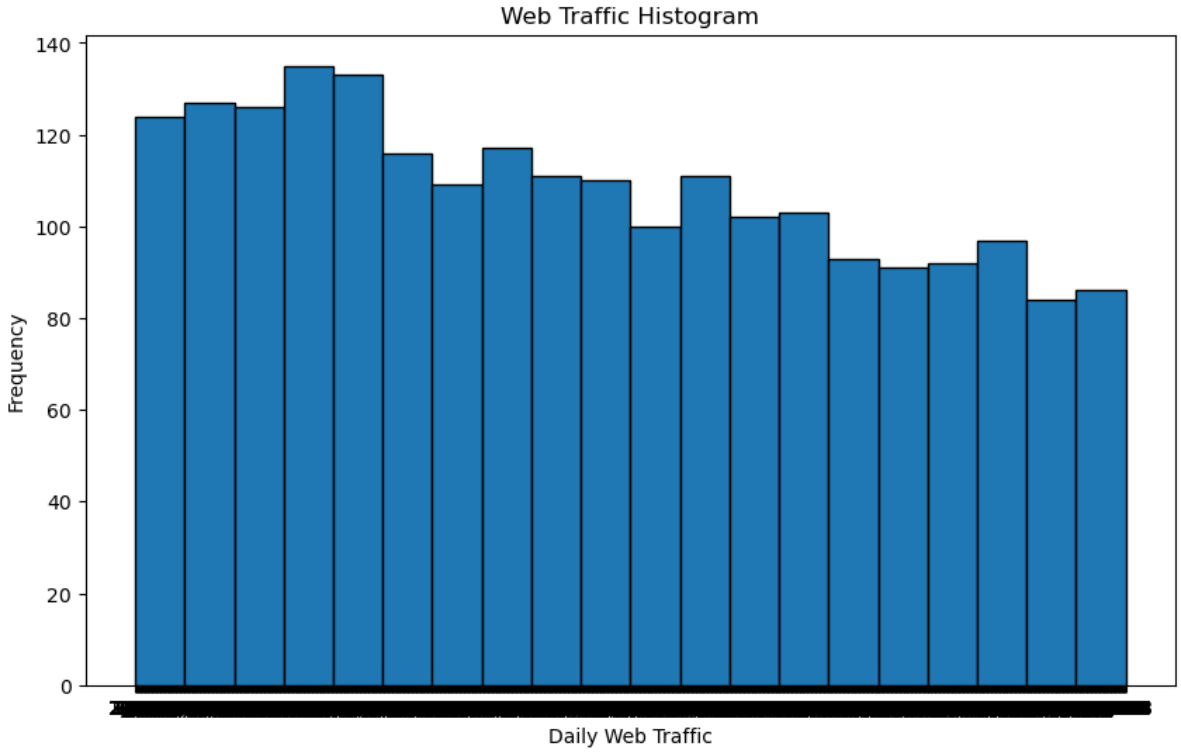
Out[15]:

	Row
Row	1.0

In [16]:
```
for i in data.columns:
    print(i,"---------",data[i].unique())
```

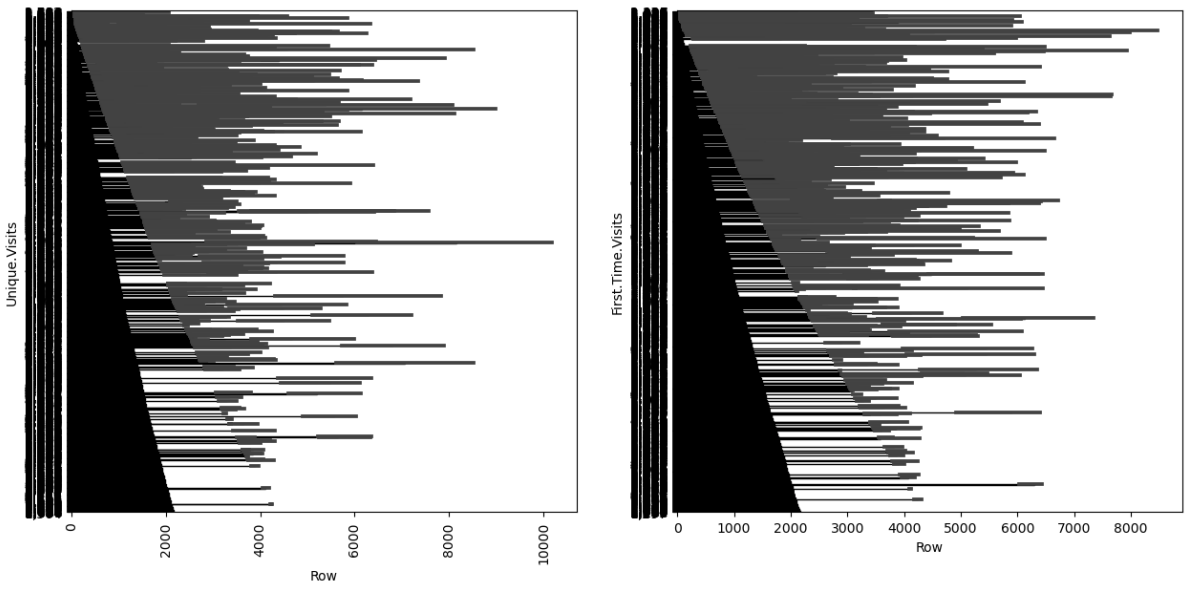
```
Row --------- [   1    2    3 ... 2165 2166 2167]
Day --------- ['Sunday' 'Monday' 'Tuesday' 'Wednesday' 'Thursday' 'Friday' 'Saturd
ay']
Day.Of.Week --------- ['2,146' '3,621' '3,698' ... '2,623' '3,581' '2,064']
Page.Loads --------- ['2,146' '3,621' '3,698' ... '2,623' '3,581' '2,064']
Unique.Visits --------- ['1,582' '2,528' '2,630' ... '2,638' '2,683' '1,564']
First.Time.Visits --------- ['1,430' '2,297' '2,352' ... '1,686' '2,184' '1,297']
Returning.Visits --------- ['152' '231' '278' '287' '236' '241' '133' '175' '274'
'268' '284' '286'
 '216' '140' '193' '296' '312' '308' '250' '162' '200' '328' '359' '351'
 '319' '282' '177' '233' '366' '387' '421' '376' '329' '203' '409' '434'
 '244' '290' '451' '423' '416' '384' '279' '226' '264' '428' '402' '403'
 '407' '343' '191' '292' '440' '410' '454' '436' '362' '208' '289' '490'
 '437' '488' '458' '389' '253' '469' '415' '406' '314' '337' '281' '560'
 '574' '561' '535' '377' '310' '546' '552' '594' '554' '442' '300' '391'
 '478' '477' '447' '320' '178' '321' '145' '197' '160' '183' '252' '182'
 '161' '258' '150' '223' '356' '385' '393' '350' '174' '220' '382' '427'
 '388' '288' '185' '221' '394' '429' '368' '270' '375' '411' '418' '464'
 '214' '297' '499' '501' '417' '242' '335' '494' '521' '509' '342' '471'
 '495' '500' '496' '380' '254' '361' '542' '570' '587' '483' '283' '556'
 '603' '533' '448' '325' '364' '578' '576' '596' '580' '486' '313' '424'
 '579' '581' '585' '551' '398' '618' '605' '571' '360' '408' '599' '604'
 '591' '504' '295' '371' '559' '648' '614' '633' '516' '318' '456' '698'
 '693' '686' '689' '566' '397' '475' '765' '715' '696' '754' '780' '760'
 '636' '502' '445' '709' '702' '679' '479' '695' '670' '668' '597' '345'
 '632' '506' '299' '354' '564' '651' '622' '466' '583' '455' '355' '589'
 '607' '639' '628' '255' '544' '545' '569' '468' '317' '553' '573' '541'
 '520' '432' '293' '522' '538' '363' '204' '229' '484' '523' '528' '467'
 '201' '276' '505' '548' '266' '525' '422' '213' '562' '503' '239' '285'
 '531' '511' '489' '401' '245' '473' '532' '524' '260' '515' '498' '470'
 '481' '536' '206' '259' '404' '235' '537' '482' '238' '311' '519' '563'
 '251' '333' '567' '584' '577' '256' '353' '547' '623' '645' '649' '661'
 '616' '331' '656' '652' '324' '657' '674' '646' '348' '419' '716' '660'
 '650' '678' '708' '704' '665' '621' '491' '461' '322' '753' '790' '749'
 '620' '833' '798' '800' '748' '617' '724' '658' '640' '598' '232' '309'
 '138' '176' '367' '326' '228' '457' '512' '426' '339' '624' '465' '301'
 '358' '575' '412' '588' '572' '392' '568' '609' '507' '613' '687' '555'
 '746' '810' '558' '370' '517' '731' '743' '736' '772' '346' '480' '737'
 '783' '744' '510' '792' '775' '694' '735' '634' '727' '681' '543' '782'
 '771' '725' '610' '400' '534' '836' '857' '837' '758' '662' '849' '877'
 '898' '830' '676' '625' '987' '924' '926' '832' '631' '907' '730' '606'
 '794' '755' '747' '680' '378' '444' '685' '787' '819' '856' '734' '803'
 '769' '675' '414' '796' '728' '673' '565' '707' '413' '677' '692' '667'
 '257' '365' '225' '269' '334' '647' '608' '611' '600' '529' '271' '381'
 '682' '672' '627' '653' '298' '476' '643' '699' '595' '275' '659' '513'
 '425' '729' '697' '294' '430' '776' '330' '742' '774' '349' '703' '714'
 '797' '841' '405' '899' '914' '815' '761' '872' '890' '897' '669' '853'
 '825' '845' '915' '814' '644' '946' '903' '919' '902' '690' '905' '918'
 '822' '487' '880' '969' '1,036' '961' '806' '784' '979' '998' '993' '957'
 '985' '895' '842' '369' '190' '439' '207' '396' '557' '590' '526' '373'
 '635' '341' '420' '684' '801' '763' '671' '705' '691' '344' '745' '717'
 '720' '700' '460' '664' '340' '721' '722' '443' '683' '719' '831' '710'
 '723' '602' '654' '452' '642' '374' '638' '463' '441' '323' '230' '527'
 '263' '492' '383' '234' '493' '518' '395' '198' '261' '435' '249' '497'
 '453' '450' '219' '265' '332' '438' '184' '291' '433' '472' '474' '446'
 '459' '305' '248' '327' '593' '372' '615' '592' '205' '199' '390' '302'
 '315' '462' '316' '399' '386' '619' '530' '641' '809' '586' '739' '767'
 '485' '706' '701' '629' '688' '711' '805' '881' '777' '663' '908' '913'
 '925' '866' '852' '848' '773' '751' '666' '612' '307' '549' '601' '306'
 '539' '626' '379' '582' '759' '732' '811' '752' '802' '630' '859' '838'
 '947' '850' '956' '937' '973' '243' '277' '550' '514' '766' '770' '823'
 '785' '778' '789' '791' '821' '817' '762' '824' '304' '181' '540' '202'
 '508' '637' '712' '157' '210' '188' '187' '246' '267' '303' '655' '834'
 '431' '280' '338']
```

In [18]:
```
plt.figure(figsize=(10, 6))
plt.hist(data['Unique.Visits'], bins=20, edgecolor='k')  # You can adjust the numbe


plt.xlabel('Daily Web Traffic')
plt.ylabel('Frequency')
plt.title('Web Traffic Histogram')


plt.show()
```



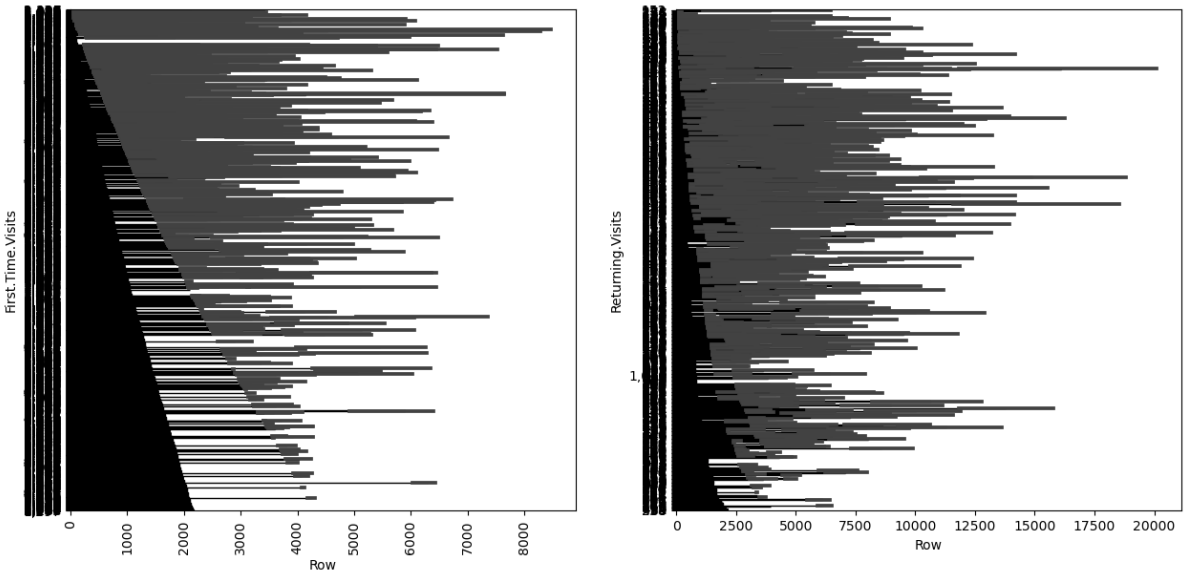
In [21]:
```
plt.figure(figsize=(15,15),dpi=100)
plt.subplot(2,2,1)
sns.barplot(x="Row",y="Unique.Visits",data=data,edgecolor="black",estimator=sum)
plt.xticks(rotation=90);
plt.subplot(2,2,2)
sns.barplot(x="Row",y="First.Time.Visits",data=data,edgecolor="black",estimator=sum
```

Out[21]:
```
<AxesSubplot:xlabel='Row', ylabel='First.Time.Visits'>
```



In [23]:
```
plt.figure(figsize=(15,15),dpi=100)
plt.subplot(2,2,1)
sns.barplot(x="Row",y="First.Time.Visits",data=data,edgecolor="black",estimator=sum
plt.xticks(rotation=90);
plt.subplot(2,2,2)
sns.barplot(x="Row",y="Returning.Visits",data=data,edgecolor="black",estimator=sum)
```

Out[23]:
```
<AxesSubplot:xlabel='Row', ylabel='Returning.Visits'>
```



In [29]:
```
plt.figure(figsize=(10, 6))
plt.scatter(data['Day'], data['First.Time.Visits'], s=20, alpha=0.5)
plt.xlabel('Day')
plt.ylabel('First.Time.Visits')
plt.title('Website Traffic Scatter Plot')
plt.show()
```