Final Project : Time-series supervised models

The objective is to create Machine Learning models to predict the next month revenue of a company in order to reduce the time managers currently spend to calculate it. For example, if your provide the date April 15 to the model as an input, it is supposed to return the predicted revenue from April 15 to May 15 (next month)

Three strategies will be used to achieve this objective, that are:

1. The month number and day number are independent variables and the other characteristics collected and calculated, including the next month revenue, are dependent variables
2. The day of the year number (366 days) are independent variables and the other characteristics collected and calculated, including the next month revenue, are dependent variables
3. The day of the year number (366 days) and the other characteristics collected are independent variables and only the next month revenue is dependent variables

1-Extract Tranform Load the dataset

In [1]:
```
#Import the required local files
import functions
from functions import cslib, etl, viz, model
```

In [2]:
```
#Import the required libraries
import pandas as pd
import numpy as np
import os
```

In [3]:
```
#Load the dataset and save it into a pandas dataframe
#Fetch data and validate data format
print ("\nFETCH DATA")
#data_directory = "data/cs_train/"
data_directory = os.path.join("data","cs_train")
df1 = cslib.fetch_data(data_directory)

#Check teh shape of the dataset to confirm it was loaded or not
etl.get_df_shape(df1)
```

```
FETCH DATA
SHAPE OF THE DATA:
Number of rows: 815011
Number of columns: 10
```

In [4]:
```
#Observe data
print ("\nOBSERVE DATA")
etl.get_data_info(df1)
```

OBSERVE DATA

FIRST 10 ROWS OF THE DATA

	country	customer_id	day	invoice	month	price	stream_id	\
0	United Kingdom	13085.0	28	489434	11	6.95	85048	
1	United Kingdom	13085.0	28	489434	11	6.75	79323W	
2	United Kingdom	13085.0	28	489434	11	2.10	22041	
3	United Kingdom	13085.0	28	489434	11	1.25	21232	
4	United Kingdom	13085.0	28	489434	11	1.65	22064	
5	United Kingdom	13085.0	28	489434	11	1.25	21871	
6	United Kingdom	13085.0	28	489434	11	5.95	21523	
7	United Kingdom	13085.0	28	489435	11	2.55	22350	
8	United Kingdom	13085.0	28	489435	11	3.75	22349	
9	United Kingdom	13085.0	28	489435	11	1.65	22195	

	times_viewed	year	invoice_date
0	12	2017	2017-11-28
1	12	2017	2017-11-28
2	21	2017	2017-11-28
3	5	2017	2017-11-28
4	17	2017	2017-11-28
5	14	2017	2017-11-28
6	10	2017	2017-11-28
7	12	2017	2017-11-28
8	12	2017	2017-11-28
9	18	2017	2017-11-28

LAST 10 ROWS OF THE DATA

	country	customer_id	day	invoice	month	price	stream_id	\
815001	United Kingdom	16098.0	31	562271	7	3.75	22728	
815002	United Kingdom	16098.0	31	562271	7	3.75	22729	
815003	United Kingdom	16098.0	31	562271	7	9.95	23315	
815004	United Kingdom	16098.0	31	562271	7	9.95	23316	
815005	United Kingdom	16098.0	31	562271	7	3.75	22730	
815006	United Kingdom	16098.0	31	562271	7	3.75	22725	
815007	United Kingdom	16098.0	31	562271	7	3.75	22726	
815008	United Kingdom	16098.0	31	562271	7	3.75	22727	
815009	United Kingdom	14056.0	31	562269	7	2.95	22090	
815010	United Kingdom	15628.0	31	562163	7	1.65	22558	

	times_viewed	year	invoice_date
815001	1	2019	2019-07-31
815002	2	2019	2019-07-31
815003	1	2019	2019-07-31
815004	1	2019	2019-07-31
815005	4	2019	2019-07-31
815006	2	2019	2019-07-31
815007	12	2019	2019-07-31
815008	6	2019	2019-07-31
815009	2	2019	2019-07-31
815010	12	2019	2019-07-31

INFO ABOUT THE DATA

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 815011 entries, 0 to 815010
Data columns (total 10 columns):
```

```
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   country       815011 non-null  object
 1   customer_id   625249 non-null  float64
 2   day           815011 non-null  int64
 3   invoice       815011 non-null  object
 4   month         815011 non-null  int64
 5   price         815011 non-null  float64
 6   stream_id     815011 non-null  object
 7   times_viewed  815011 non-null  int64
 8   year          815011 non-null  int64
 9   invoice_date  815011 non-null  datetime64[s]
dtypes: datetime64[s](1), float64(2), int64(4), object(3)
memory usage: 62.2+ MB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 815011 entries, 0 to 815010
Data columns (total 10 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   country       815011 non-null  object
 1   customer_id   625249 non-null  float64
 2   day           815011 non-null  int64
 3   invoice       815011 non-null  object
 4   month         815011 non-null  int64
 5   price         815011 non-null  float64
 6   stream_id     815011 non-null  object
 7   times_viewed  815011 non-null  int64
 8   year          815011 non-null  int64
 9   invoice_date  815011 non-null  datetime64[s]
dtypes: datetime64[s](1), float64(2), int64(4), object(3)
memory usage: 62.2+ MB
None

MISSING DATA
189762 data are missing from the column named customer_id

DUPLICATED DATA ROWS
28844 rows are duplicated
First 5 duplicated rows:
            country  customer_id  day invoice  month  price stream_id  \
373  United Kingdom      16329.0   28  489517     11   3.75     21912

     times_viewed  year invoice_date
373             1  2017   2017-11-28
            country  customer_id  day invoice  month  price stream_id  \
382  United Kingdom      16329.0   28  489517     11   0.85     22130

     times_viewed  year invoice_date
382             6  2017   2017-11-28
            country  customer_id  day invoice  month  price stream_id  \
383  United Kingdom      16329.0   28  489517     11   0.65     22319

     times_viewed  year invoice_date
```

```
383            12  2017   2017-11-28
           country  customer_id  day invoice  month  price stream_id  \
384  United Kingdom      16329.0   28  489517     11   3.75     21913

     times_viewed  year invoice_date
384             1  2017   2017-11-28

DESCRIPTIVE STATISTICS ABOUT CATEGORICAL OBJECT COLUMNS
               country invoice stream_id
count           815011  815011    815011
unique              43   42646      5007
top     United Kingdom  537434    85123A
freq            751228    1350      5017
country : 43
customer_id : 5226
day : 31
invoice : 42646
month : 12
price : 2415
stream_id : 5007
times_viewed : 25
year : 3
invoice_date : 495

There is no column that includes a unique category of data
```

In [5]:
```python
def delete_duplicate_rows(df_source):
    #Delete duplicated rows
    df_target=df_source.drop_duplicates()
    #Calculate the number of rows deleted
    nbr_rows = df_source.shape[0] - df_target.shape[0]
    print (nbr_rows, "duplicate were removed")
    #Check teh shape of the dataset to confirm it was loaded or not
    print('PREVIOUS')
    etl.get_df_shape(df_source)
    print('CURRENT')
    etl.get_df_shape(df_target)
    #Return output
    return df_target

df2=delete_duplicate_rows(df1)
```

```
28844 duplicate were removed
PREVIOUS
SHAPE OF THE DATA:
Number of rows: 815011
Number of columns: 10
CURRENT
SHAPE OF THE DATA:
Number of rows: 786167
Number of columns: 10
```

In [6]:
```python
#Observe data
print ("\nOBSERVE DATA")
etl.get_data_info(df2)
```

OBSERVE DATA

FIRST 10 ROWS OF THE DATA

	country	customer_id	day	invoice	month	price	stream_id	\
0	United Kingdom	13085.0	28	489434	11	6.95	85048	
1	United Kingdom	13085.0	28	489434	11	6.75	79323W	
2	United Kingdom	13085.0	28	489434	11	2.10	22041	
3	United Kingdom	13085.0	28	489434	11	1.25	21232	
4	United Kingdom	13085.0	28	489434	11	1.65	22064	
5	United Kingdom	13085.0	28	489434	11	1.25	21871	
6	United Kingdom	13085.0	28	489434	11	5.95	21523	
7	United Kingdom	13085.0	28	489435	11	2.55	22350	
8	United Kingdom	13085.0	28	489435	11	3.75	22349	
9	United Kingdom	13085.0	28	489435	11	1.65	22195	

	times_viewed	year	invoice_date
0	12	2017	2017-11-28
1	12	2017	2017-11-28
2	21	2017	2017-11-28
3	5	2017	2017-11-28
4	17	2017	2017-11-28
5	14	2017	2017-11-28
6	10	2017	2017-11-28
7	12	2017	2017-11-28
8	12	2017	2017-11-28
9	18	2017	2017-11-28

LAST 10 ROWS OF THE DATA

	country	customer_id	day	invoice	month	price	stream_id	\
815001	United Kingdom	16098.0	31	562271	7	3.75	22728	
815002	United Kingdom	16098.0	31	562271	7	3.75	22729	
815003	United Kingdom	16098.0	31	562271	7	9.95	23315	
815004	United Kingdom	16098.0	31	562271	7	9.95	23316	
815005	United Kingdom	16098.0	31	562271	7	3.75	22730	
815006	United Kingdom	16098.0	31	562271	7	3.75	22725	
815007	United Kingdom	16098.0	31	562271	7	3.75	22726	
815008	United Kingdom	16098.0	31	562271	7	3.75	22727	
815009	United Kingdom	14056.0	31	562269	7	2.95	22090	
815010	United Kingdom	15628.0	31	562163	7	1.65	22558	

	times_viewed	year	invoice_date
815001	1	2019	2019-07-31
815002	2	2019	2019-07-31
815003	1	2019	2019-07-31
815004	1	2019	2019-07-31
815005	4	2019	2019-07-31
815006	2	2019	2019-07-31
815007	12	2019	2019-07-31
815008	6	2019	2019-07-31
815009	2	2019	2019-07-31
815010	12	2019	2019-07-31

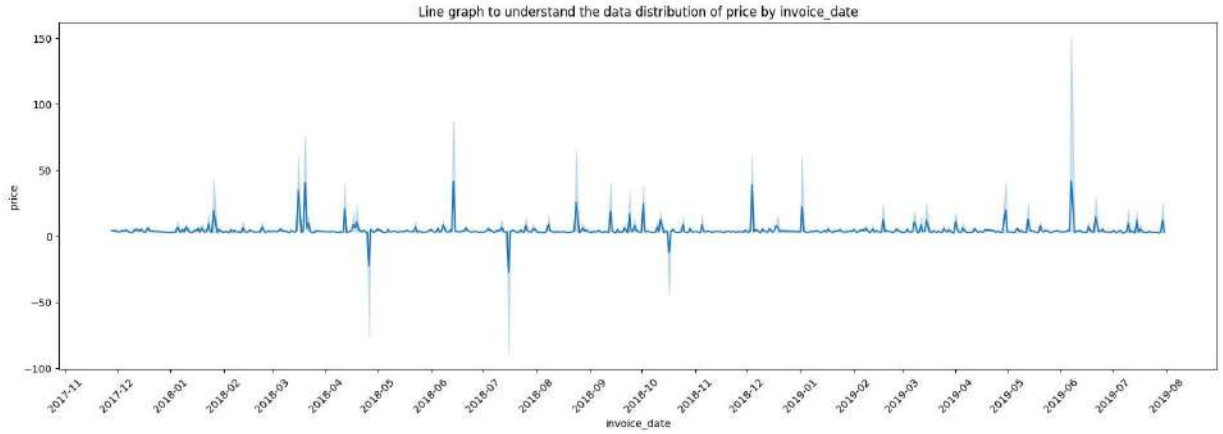
INFO ABOUT THE DATA

```
<class 'pandas.core.frame.DataFrame'>
Index: 786167 entries, 0 to 815010
Data columns (total 10 columns):
```

```
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   country       786167 non-null  object
 1   customer_id   603973 non-null  float64
 2   day           786167 non-null  int64
 3   invoice       786167 non-null  object
 4   month         786167 non-null  int64
 5   price         786167 non-null  float64
 6   stream_id     786167 non-null  object
 7   times_viewed  786167 non-null  int64
 8   year          786167 non-null  int64
 9   invoice_date  786167 non-null  datetime64[s]
dtypes: datetime64[s](1), float64(2), int64(4), object(3)
memory usage: 66.0+ MB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
Index: 786167 entries, 0 to 815010
Data columns (total 10 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   country       786167 non-null  object
 1   customer_id   603973 non-null  float64
 2   day           786167 non-null  int64
 3   invoice       786167 non-null  object
 4   month         786167 non-null  int64
 5   price         786167 non-null  float64
 6   stream_id     786167 non-null  object
 7   times_viewed  786167 non-null  int64
 8   year          786167 non-null  int64
 9   invoice_date  786167 non-null  datetime64[s]
dtypes: datetime64[s](1), float64(2), int64(4), object(3)
memory usage: 66.0+ MB
None

MISSING DATA
182194 data are missing from the column named customer_id

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT CATEGORICAL OBJECT COLUMNS
              country invoice stream_id
count          786167  786167    786167
unique             43   42646      5007
top     United Kingdom  558475    85123A
freq           723324     705      4869
country : 43
customer_id : 5226
day : 31
invoice : 42646
month : 12
price : 2415
stream_id : 5007
times_viewed : 25
```

```
year : 3
invoice_date : 495

There is no column that includes a unique category of data
```

In [7]:
```
#Display a line graph of daily revenue
viz.show_lineGraph(df2, 'invoice_date', 'price')
```



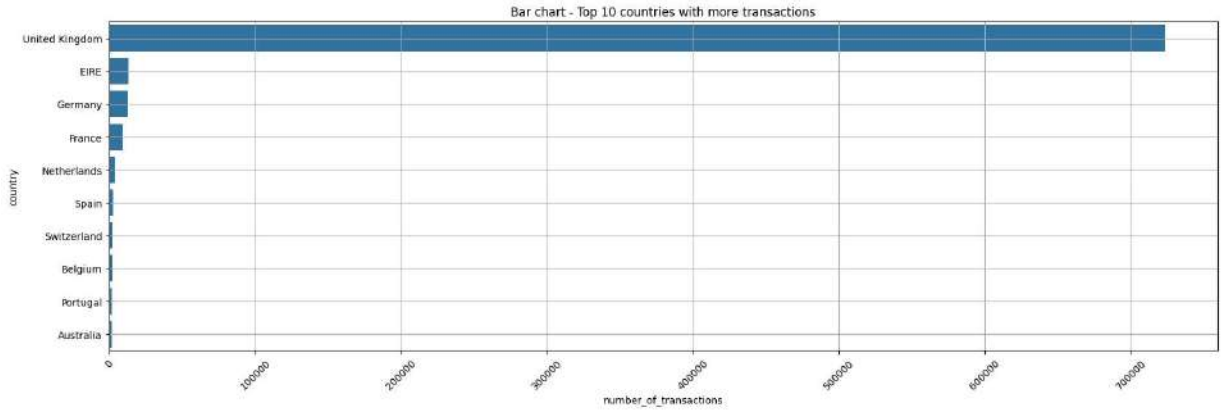
In [8]:
```
#Show the countries by their number of transactions from the highest to the lowest
df_count_country=df2[['invoice_date', 'country']].groupby(['country']).count()
df_count_country=df_count_country.rename(columns={"invoice_date": "number_of_transa
df_count_country=df_count_country.reset_index()
df_count_country=df_count_country.sort_values(by='number_of_transactions', ascendin
df_count_country['Weighting %']=(df_count_country['number_of_transactions']/df_coun
df_count_country=df_count_country.reset_index(drop=True)
df_count_country
```

Out[8]:

	country	number_of_transactions	Weighting %
0	United Kingdom	723324	92.006406
1	EIRE	13432	1.708543
2	Germany	13123	1.669238
3	France	9669	1.229891
4	Netherlands	4022	0.511596
5	Spain	2557	0.325249
6	Switzerland	2320	0.295103
7	Belgium	2159	0.274624
8	Portugal	1734	0.220564
9	Australia	1525	0.193979
10	Channel Islands	1270	0.161543
11	Sweden	1155	0.146915
12	Italy	1025	0.130379
13	Cyprus	898	0.114225
14	Finland	749	0.095272
15	Norway	736	0.093619
16	Austria	719	0.091456
17	Greece	627	0.079754
18	Denmark	617	0.078482
19	Unspecified	611	0.077719
20	Japan	501	0.063727
21	United Arab Emirates	462	0.058766
22	Poland	425	0.054060
23	Singapore	310	0.039432
24	Hong Kong	274	0.034853
25	USA	266	0.033835
26	Canada	223	0.028365
27	Malta	219	0.027857
28	Lithuania	173	0.022006
29	Iceland	167	0.021242

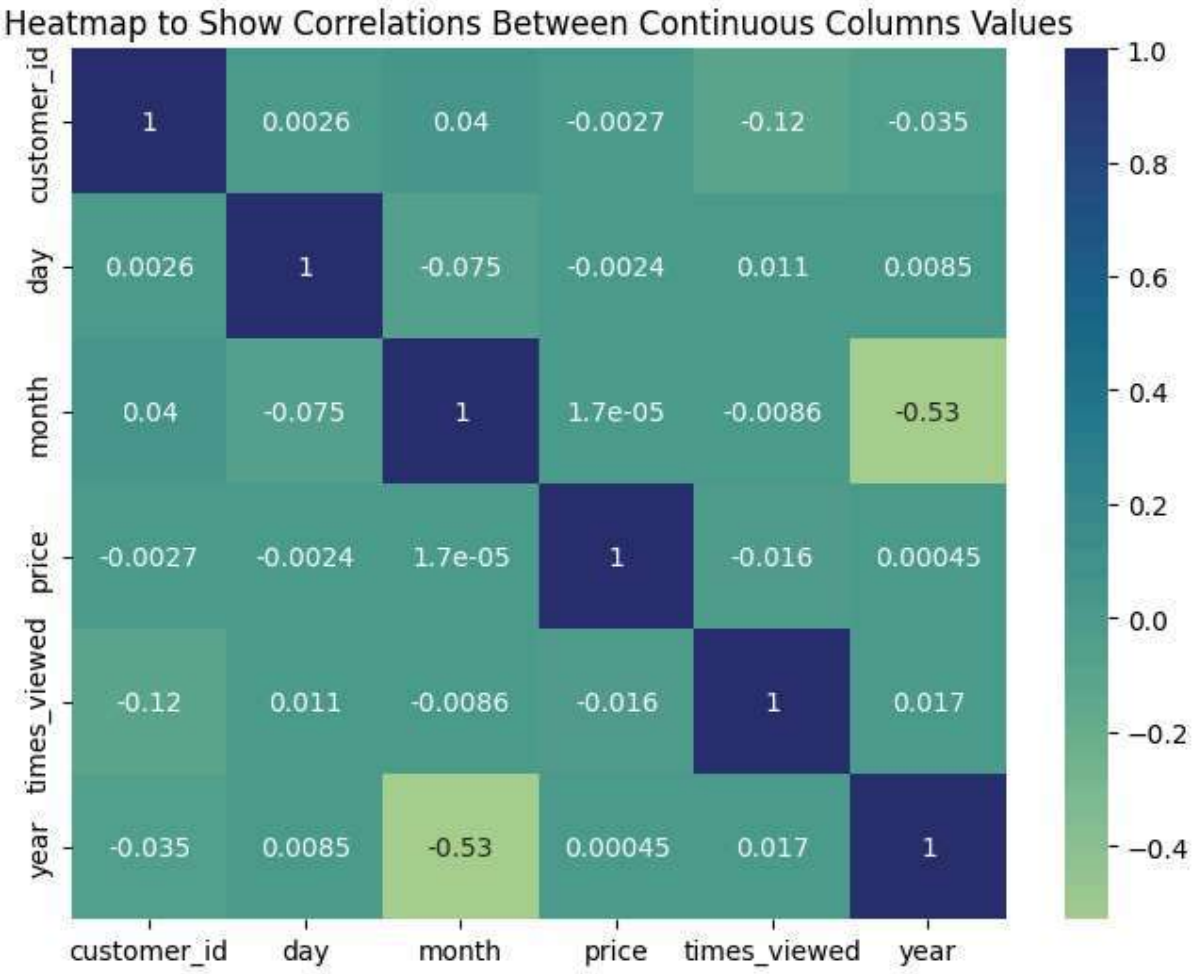
	country	number_of_transactions	Weighting %
30	Israel	139	0.017681
31	Bahrain	126	0.016027
32	RSA	111	0.014119
33	Brazil	94	0.011957
34	Thailand	76	0.009667
35	Korea	63	0.008014
36	European Community	61	0.007759
37	Lebanon	58	0.007378
38	West Indies	54	0.006869
39	Bermuda	34	0.004325
40	Nigeria	32	0.004070
41	Czech Republic	17	0.002162
42	Saudi Arabia	10	0.001272

In [9]:
```
#Plot top 10 contries
df_count_country=df_count_country.iloc[0: 10]
viz.show_barChart2(df_count_country)
```



In [10]:
```
#Display correlation
print ("\nDISPLAY CORRELATION BETWEEN CONTINUOUS COLUMN VALUES\n")
viz.show_heatmap(df2)
```

```
DISPLAY CORRELATION BETWEEN CONTINUOUS COLUMN VALUES

The following columns do not contain numerical data and then were removed:
#1 : country
#2 : invoice
#3 : stream_id
#4 : invoice_date
```



In [11]:
```
#Convert data to time series data and save them in csv files
ts_files_dir = "C:/Users/LSP/Documents/AI WorkFlow Capstone/data/ts_data"
#cslib.main(dt_dir, ts_df_dir)
cslib.fetch_ts(df2, ts_files_dir, clean=False)
```

```
... loading ts data from files
```

Out[11]:
```
{'all':           date  purchases  unique_invoices  unique_streams  total_views
\
 0    2017-11-01          0                0               0            0
 1    2017-11-02          0                0               0            0
 2    2017-11-03          0                0               0            0
 3    2017-11-04          0                0               0            0
 4    2017-11-05          0                0               0            0
 ..          ...        ...              ...             ...          ...
 602  2019-06-26       1357               67             999         6419
 603  2019-06-27       1613               80             944         9425
 604  2019-06-28       1011               70             607         5518
 605  2019-06-29          0                0               0            0
 606  2019-06-30        581               27             423         2508

      revenue
 0       0.00
 1       0.00
 2       0.00
 3       0.00
 4       0.00
 ..       ...
 602  4902.32
 603  5477.33
 604  3535.50
 605     0.00
 606  1741.25

 [607 rows x 6 columns],
 'eire':           date  purchases  unique_invoices  unique_streams  total_views
\
 0    2017-11-01          0                0               0            0
 1    2017-11-02          0                0               0            0
 2    2017-11-03          0                0               0            0
 3    2017-11-04          0                0               0            0
 4    2017-11-05          0                0               0            0
 ..          ...        ...              ...             ...          ...
 602  2019-06-26         80                3              78          709
 603  2019-06-27         30                1              30          237
 604  2019-06-28         55                3              54          440
 605  2019-06-29          0                0               0            0
 606  2019-06-30          0                0               0            0

      revenue
 0       0.00
 1       0.00
 2       0.00
 3       0.00
 4       0.00
 ..       ...
 602   253.67
 603    92.16
 604   295.21
 605     0.00
 606     0.00

 [607 rows x 6 columns],
```

```
 'france':           date  purchases  unique_invoices  unique_streams  total_view
s  \
 0    2017-11-01          0                0               0           0
 1    2017-11-02          0                0               0           0
 2    2017-11-03          0                0               0           0
 3    2017-11-04          0                0               0           0
 4    2017-11-05          0                0               0           0
 ..          ...        ...              ...             ...         ...
 602  2019-06-26          0                0               0           0
 603  2019-06-27          1                1               1           4
 604  2019-06-28         26                2              25         252
 605  2019-06-29          0                0               0           0
 606  2019-06-30          0                0               0           0

      revenue
 0       0.00
 1       0.00
 2       0.00
 3       0.00
 4       0.00
 ..       ...
 602     0.00
 603     1.25
 604    79.75
 605     0.00
 606     0.00

 [607 rows x 6 columns],
 'germany':           date  purchases  unique_invoices  unique_streams  total_vie
ws  \
 0    2017-11-01          0                0               0           0
 1    2017-11-02          0                0               0           0
 2    2017-11-03          0                0               0           0
 3    2017-11-04          0                0               0           0
 4    2017-11-05          0                0               0           0
 ..          ...        ...              ...             ...         ...
 602  2019-06-26          0                0               0           0
 603  2019-06-27         62                3              59         614
 604  2019-06-28         10                2              10          52
 605  2019-06-29          0                0               0           0
 606  2019-06-30          0                0               0           0

      revenue
 0       0.00
 1       0.00
 2       0.00
 3       0.00
 4       0.00
 ..       ...
 602     0.00
 603   159.78
 604    51.89
 605     0.00
 606     0.00

 [607 rows x 6 columns],
```

```
 'hong_kong':            date  purchases  unique_invoices  unique_streams  total_v
iews  \
 0    2018-04-01          0                0               0            0
 1    2018-04-02          0                0               0            0
 2    2018-04-03          0                0               0            0
 3    2018-04-04          0                0               0            0
 4    2018-04-05          0                0               0            0
 ..          ...        ...              ...             ...          ...
 421  2019-05-27          0                0               0            0
 422  2019-05-28          0                0               0            0
 423  2019-05-29          0                0               0            0
 424  2019-05-30          0                0               0            0
 425  2019-05-31          0                0               0            0

      revenue
 0        0.0
 1        0.0
 2        0.0
 3        0.0
 4        0.0
 ..       ...
 421      0.0
 422      0.0
 423      0.0
 424      0.0
 425      0.0

 [426 rows x 6 columns],
 'netherlands':            date  purchases  unique_invoices  unique_streams  total
_views  \
 0    2017-11-01          0                0               0            0
 1    2017-11-02          0                0               0            0
 2    2017-11-03          0                0               0            0
 3    2017-11-04          0                0               0            0
 4    2017-11-05          0                0               0            0
 ..          ...        ...              ...             ...          ...
 602  2019-06-26          0                0               0            0
 603  2019-06-27          0                0               0            0
 604  2019-06-28          0                0               0            0
 605  2019-06-29          0                0               0            0
 606  2019-06-30          0                0               0            0

      revenue
 0        0.0
 1        0.0
 2        0.0
 3        0.0
 4        0.0
 ..       ...
 602      0.0
 603      0.0
 604      0.0
 605      0.0
 606      0.0

 [607 rows x 6 columns],
```

```
 'norway':            date  purchases  unique_invoices  unique_streams  total_view
s  \
 0    2017-12-01          0                0               0            0
 1    2017-12-02          0                0               0            0
 2    2017-12-03          0                0               0            0
 3    2017-12-04         21                1              21          175
 4    2017-12-05          0                0               0            0
 ..          ...        ...              ...             ...          ...
 572  2019-06-26          0                0               0            0
 573  2019-06-27          0                0               0            0
 574  2019-06-28          0                0               0            0
 575  2019-06-29          0                0               0            0
 576  2019-06-30          0                0               0            0

      revenue
 0       0.00
 1       0.00
 2       0.00
 3     123.28
 4       0.00
 ..       ...
 572     0.00
 573     0.00
 574     0.00
 575     0.00
 576     0.00

 [577 rows x 6 columns],
 'portugal':            date  purchases  unique_invoices  unique_streams  total_vi
ews  \
 0    2017-11-01          0                0               0            0
 1    2017-11-02          0                0               0            0
 2    2017-11-03          0                0               0            0
 3    2017-11-04          0                0               0            0
 4    2017-11-05          0                0               0            0
 ..          ...        ...              ...             ...          ...
 602  2019-06-26          0                0               0            0
 603  2019-06-27          0                0               0            0
 604  2019-06-28          0                0               0            0
 605  2019-06-29          0                0               0            0
 606  2019-06-30          0                0               0            0

      revenue
 0        0.0
 1        0.0
 2        0.0
 3        0.0
 4        0.0
 ..       ...
 602      0.0
 603      0.0
 604      0.0
 605      0.0
 606      0.0

 [607 rows x 6 columns],
```

```
 'singapore':           date  purchases  unique_invoices  unique_streams  total_v
iews  \
 0    2018-04-01          0                0               0            0
 1    2018-04-02          0                0               0            0
 2    2018-04-03          0                0               0            0
 3    2018-04-04          0                0               0            0
 4    2018-04-05          0                0               0            0
 ..          ...        ...              ...             ...          ...
 451  2019-06-26          0                0               0            0
 452  2019-06-27          0                0               0            0
 453  2019-06-28          0                0               0            0
 454  2019-06-29          0                0               0            0
 455  2019-06-30          0                0               0            0

      revenue
 0        0.0
 1        0.0
 2        0.0
 3        0.0
 4        0.0
 ..       ...
 451      0.0
 452      0.0
 453      0.0
 454      0.0
 455      0.0

 [456 rows x 6 columns],
 'spain':           date  purchases  unique_invoices  unique_streams  total_views
\
 0    2017-11-01          0                0               0            0
 1    2017-11-02          0                0               0            0
 2    2017-11-03          0                0               0            0
 3    2017-11-04          0                0               0            0
 4    2017-11-05          0                0               0            0
 ..          ...        ...              ...             ...          ...
 602  2019-06-26          0                0               0            0
 603  2019-06-27          0                0               0            0
 604  2019-06-28          0                0               0            0
 605  2019-06-29          0                0               0            0
 606  2019-06-30          0                0               0            0

      revenue
 0        0.0
 1        0.0
 2        0.0
 3        0.0
 4        0.0
 ..       ...
 602      0.0
 603      0.0
 604      0.0
 605      0.0
 606      0.0

 [607 rows x 6 columns],
```

```
 'united_kingdom':            date  purchases  unique_invoices  unique_streams  to
tal_views  \
 0    2017-11-01          0                0                0               0
 1    2017-11-02          0                0                0               0
 2    2017-11-03          0                0                0               0
 3    2017-11-04          0                0                0               0
 4    2017-11-05          0                0                0               0
 ..          ...        ...              ...              ...             ...
 602  2019-06-26       1273               62              965            5701
 603  2019-06-27       1480               72              897            8220
 604  2019-06-28        907               62              565            4653
 605  2019-06-29          0                0                0               0
 606  2019-06-30        581               27              423            2508

      revenue
 0       0.00
 1       0.00
 2       0.00
 3       0.00
 4       0.00
 ..       ...
 602  4610.15
 603  5105.97
 604  3061.10
 605     0.00
 606  1741.25

 [607 rows x 6 columns]}
```

In [12]:
```
#Display the path, including names, of the files created
import re
file_list = [os.path.join(ts_files_dir,f) for f in os.listdir(ts_files_dir) if re.s
file_list
```

Out[12]:
```
['C:/Users/LSP/Documents/AI WorkFlow Capstone/data/ts_data\\ts-all.csv',
 'C:/Users/LSP/Documents/AI WorkFlow Capstone/data/ts_data\\ts-eire.csv',
 'C:/Users/LSP/Documents/AI WorkFlow Capstone/data/ts_data\\ts-france.csv',
 'C:/Users/LSP/Documents/AI WorkFlow Capstone/data/ts_data\\ts-germany.csv',
 'C:/Users/LSP/Documents/AI WorkFlow Capstone/data/ts_data\\ts-hong_kong.csv',
 'C:/Users/LSP/Documents/AI WorkFlow Capstone/data/ts_data\\ts-netherlands.csv',
 'C:/Users/LSP/Documents/AI WorkFlow Capstone/data/ts_data\\ts-norway.csv',
 'C:/Users/LSP/Documents/AI WorkFlow Capstone/data/ts_data\\ts-portugal.csv',
 'C:/Users/LSP/Documents/AI WorkFlow Capstone/data/ts_data\\ts-singapore.csv',
 'C:/Users/LSP/Documents/AI WorkFlow Capstone/data/ts_data\\ts-spain.csv',
 'C:/Users/LSP/Documents/AI WorkFlow Capstone/data/ts_data\\ts-united_kingdom.cs
v']
```

In [13]:
```
#Create a dataframe with each time serie dataset
list_ts_df = ['df_ts_all','df_ts_eire','df_ts_france','df_ts_germany','df_ts_hong_k
library_ts_df = {}

if len(file_list)==len(list_ts_df):
    size = len(file_list)
    for i in range (0,size):
        df_name = list_ts_df[i]
        library_ts_df[df_name] = pd.read_csv(file_list[i])
```

```
        #Check whether the dataset was loaded or not
        print(i+1, '- Dataframe', df_name , 'was created successfully!')
        etl.get_df_shape(library_ts_df[df_name])
        print('\n')
```

```
1 - Dataframe df_ts_all was created successfully!
SHAPE OF THE DATA:
Number of rows: 607
Number of columns: 6


2 - Dataframe df_ts_eire was created successfully!
SHAPE OF THE DATA:
Number of rows: 607
Number of columns: 6


3 - Dataframe df_ts_france was created successfully!
SHAPE OF THE DATA:
Number of rows: 607
Number of columns: 6


4 - Dataframe df_ts_germany was created successfully!
SHAPE OF THE DATA:
Number of rows: 607
Number of columns: 6


5 - Dataframe df_ts_hong_kong was created successfully!
SHAPE OF THE DATA:
Number of rows: 426
Number of columns: 6


6 - Dataframe df_ts_netherlands was created successfully!
SHAPE OF THE DATA:
Number of rows: 607
Number of columns: 6


7 - Dataframe df_ts_norway was created successfully!
SHAPE OF THE DATA:
Number of rows: 577
Number of columns: 6


8 - Dataframe df_ts_portugal was created successfully!
SHAPE OF THE DATA:
Number of rows: 607
Number of columns: 6


9 - Dataframe df_ts_singapore was created successfully!
SHAPE OF THE DATA:
Number of rows: 456
Number of columns: 6


10 - Dataframe df_ts_spain was created successfully!
SHAPE OF THE DATA:
```

```
Number of rows: 607
Number of columns: 6


11 - Dataframe df_ts_united_kingdom was created successfully!
SHAPE OF THE DATA:
Number of rows: 607
Number of columns: 6
```

In [14]:

```python
#Observe data in the time serie datasets
print ("\nOBSERVE DATA")
for i in range (0,size):
    df_name = list_ts_df[i]
    print(i+1, '- Dataframe', df_name , 'data:')
    etl.get_data_info(library_ts_df[df_name])
    print('\n')
```

OBSERVE DATA
1 - Dataframe df_ts_all data:

FIRST 10 ROWS OF THE DATA

	date	purchases	unique_invoices	unique_streams	total_views	revenue
0	2017-11-01	0	0	0	0	0.0
1	2017-11-02	0	0	0	0	0.0
2	2017-11-03	0	0	0	0	0.0
3	2017-11-04	0	0	0	0	0.0
4	2017-11-05	0	0	0	0	0.0
5	2017-11-06	0	0	0	0	0.0
6	2017-11-07	0	0	0	0	0.0
7	2017-11-08	0	0	0	0	0.0
8	2017-11-09	0	0	0	0	0.0
9	2017-11-10	0	0	0	0	0.0

LAST 10 ROWS OF THE DATA

	date	purchases	unique_invoices	unique_streams	total_views	revenue
597	2019-06-21	1035	63	652	6401	15245.14
598	2019-06-22	0	0	0	0	0.00
599	2019-06-23	696	36	557	2724	2565.32
600	2019-06-24	1172	61	724	5140	4653.83
601	2019-06-25	1033	84	647	6770	5245.46
602	2019-06-26	1357	67	999	6419	4902.32
603	2019-06-27	1613	80	944	9425	5477.33
604	2019-06-28	1011	70	607	5518	3535.50
605	2019-06-29	0	0	0	0	0.00
606	2019-06-30	581	27	423	2508	1741.25

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606

```
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             607 non-null    object
 1   purchases        607 non-null    int64
 2   unique_invoices  607 non-null    int64
 3   unique_streams   607 non-null    int64
 4   total_views      607 non-null    int64
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             607 non-null    object
 1   purchases        607 non-null    int64
 2   unique_invoices  607 non-null    int64
 3   unique_streams   607 non-null    int64
 4   total_views      607 non-null    int64
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT CATEGORICAL OBJECT COLUMNS
              date
count          607
unique         607
top     2019-06-30
freq             1
date : 607
purchases : 421
unique_invoices : 141
unique_streams : 358
total_views : 462
revenue : 469

There is no column that includes a unique category of data


2 - Dataframe df_ts_eire data:

FIRST 10 ROWS OF THE DATA
         date  purchases  unique_invoices  unique_streams  total_views  \
0  2017-11-01          0                0               0            0
```

```
1  2017-11-02          0                0              0           0
2  2017-11-03          0                0              0           0
3  2017-11-04          0                0              0           0
4  2017-11-05          0                0              0           0
5  2017-11-06          0                0              0           0
6  2017-11-07          0                0              0           0
7  2017-11-08          0                0              0           0
8  2017-11-09          0                0              0           0
9  2017-11-10          0                0              0           0

   revenue
0      0.0
1      0.0
2      0.0
3      0.0
4      0.0
5      0.0
6      0.0
7      0.0
8      0.0
9      0.0

LAST 10 ROWS OF THE DATA
           date  purchases  unique_invoices  unique_streams  total_views  \
597  2019-06-21         73                5              71          646
598  2019-06-22          0                0               0            0
599  2019-06-23          0                0               0            0
600  2019-06-24         46                2              42          327
601  2019-06-25          0                0               0            0
602  2019-06-26         80                3              78          709
603  2019-06-27         30                1              30          237
604  2019-06-28         55                3              54          440
605  2019-06-29          0                0               0            0
606  2019-06-30          0                0               0            0

     revenue
597  1325.09
598     0.00
599     0.00
600   178.44
601     0.00
602   253.67
603    92.16
604   295.21
605     0.00
606     0.00

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             607 non-null    object
 1   purchases        607 non-null    int64
 2   unique_invoices  607 non-null    int64
```

```
 3   unique_streams   607 non-null    int64
 4   total_views      607 non-null    int64
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             607 non-null    object
 1   purchases        607 non-null    int64
 2   unique_invoices  607 non-null    int64
 3   unique_streams   607 non-null    int64
 4   total_views      607 non-null    int64
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT CATEGORICAL OBJECT COLUMNS
             date
count         607
unique        607
top     2019-06-30
freq            1
date : 607
purchases : 113
unique_invoices : 11
unique_streams : 102
total_views : 227
revenue : 277

There is no column that includes a unique category of data


3 - Dataframe df_ts_france data:

FIRST 10 ROWS OF THE DATA
         date  purchases  unique_invoices  unique_streams  total_views  \
0  2017-11-01          0                0               0            0
1  2017-11-02          0                0               0            0
2  2017-11-03          0                0               0            0
3  2017-11-04          0                0               0            0
4  2017-11-05          0                0               0            0
5  2017-11-06          0                0               0            0
6  2017-11-07          0                0               0            0
```

```
7  2017-11-08          0               0               0           0
8  2017-11-09          0               0               0           0
9  2017-11-10          0               0               0           0

   revenue
0      0.0
1      0.0
2      0.0
3      0.0
4      0.0
5      0.0
6      0.0
7      0.0
8      0.0
9      0.0

LAST 10 ROWS OF THE DATA
           date  purchases  unique_invoices  unique_streams  total_views  \
597  2019-06-21         31                1              31          317
598  2019-06-22          0                0               0            0
599  2019-06-23          0                0               0            0
600  2019-06-24         10                1              10           98
601  2019-06-25          8                1               8           84
602  2019-06-26          0                0               0            0
603  2019-06-27          1                1               1            4
604  2019-06-28         26                2              25          252
605  2019-06-29          0                0               0            0
606  2019-06-30          0                0               0            0

     revenue
597    84.15
598     0.00
599     0.00
600    33.75
601    26.62
602     0.00
603     1.25
604    79.75
605     0.00
606     0.00

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             607 non-null    object
 1   purchases        607 non-null    int64
 2   unique_invoices  607 non-null    int64
 3   unique_streams   607 non-null    int64
 4   total_views      607 non-null    int64
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None
```

```
BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             607 non-null    object
 1   purchases        607 non-null    int64
 2   unique_invoices  607 non-null    int64
 3   unique_streams   607 non-null    int64
 4   total_views      607 non-null    int64
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT CATEGORICAL OBJECT COLUMNS
              date
count          607
unique         607
top     2019-06-30
freq             1
date : 607
purchases : 92
unique_invoices : 8
unique_streams : 84
total_views : 216
revenue : 265

There is no column that includes a unique category of data


4 - Dataframe df_ts_germany data:

FIRST 10 ROWS OF THE DATA
         date  purchases  unique_invoices  unique_streams  total_views  \
0  2017-11-01          0                0               0            0
1  2017-11-02          0                0               0            0
2  2017-11-03          0                0               0            0
3  2017-11-04          0                0               0            0
4  2017-11-05          0                0               0            0
5  2017-11-06          0                0               0            0
6  2017-11-07          0                0               0            0
7  2017-11-08          0                0               0            0
8  2017-11-09          0                0               0            0
9  2017-11-10          0                0               0            0

   revenue
0      0.0
```

```
1      0.0
2      0.0
3      0.0
4      0.0
5      0.0
6      0.0
7      0.0
8      0.0
9      0.0

LAST 10 ROWS OF THE DATA
           date  purchases  unique_invoices  unique_streams  total_views  \
597  2019-06-21          9                1               9           73
598  2019-06-22          0                0               0            0
599  2019-06-23          1                1               1           15
600  2019-06-24          0                0               0            0
601  2019-06-25          0                0               0            0
602  2019-06-26          0                0               0            0
603  2019-06-27         62                3              59          614
604  2019-06-28         10                2              10           52
605  2019-06-29          0                0               0            0
606  2019-06-30          0                0               0            0

     revenue
597    48.89
598     0.00
599     2.95
600     0.00
601     0.00
602     0.00
603   159.78
604    51.89
605     0.00
606     0.00

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             607 non-null    object
 1   purchases        607 non-null    int64
 2   unique_invoices  607 non-null    int64
 3   unique_streams   607 non-null    int64
 4   total_views      607 non-null    int64
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
```

```
---  ------           --------------  -----
 0   date             607 non-null    object
 1   purchases        607 non-null    int64
 2   unique_invoices  607 non-null    int64
 3   unique_streams   607 non-null    int64
 4   total_views      607 non-null    int64
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT CATEGORICAL OBJECT COLUMNS
              date
count          607
unique         607
top     2019-06-30
freq             1
date : 607
purchases : 100
unique_invoices : 11
unique_streams : 98
total_views : 254
revenue : 334

There is no column that includes a unique category of data


5 - Dataframe df_ts_hong_kong data:

FIRST 10 ROWS OF THE DATA
         date  purchases  unique_invoices  unique_streams  total_views  \
0  2018-04-01          0                0               0            0
1  2018-04-02          0                0               0            0
2  2018-04-03          0                0               0            0
3  2018-04-04          0                0               0            0
4  2018-04-05          0                0               0            0
5  2018-04-06          0                0               0            0
6  2018-04-07          0                0               0            0
7  2018-04-08          0                0               0            0
8  2018-04-09          0                0               0            0
9  2018-04-10          0                0               0            0

   revenue
0      0.0
1      0.0
2      0.0
3      0.0
4      0.0
5      0.0
6      0.0
```

```
7      0.0
8      0.0
9      0.0

LAST 10 ROWS OF THE DATA
           date  purchases  unique_invoices  unique_streams  total_views  \
416  2019-05-22          0                0               0            0
417  2019-05-23          0                0               0            0
418  2019-05-24          0                0               0            0
419  2019-05-25          0                0               0            0
420  2019-05-26          0                0               0            0
421  2019-05-27          0                0               0            0
422  2019-05-28          0                0               0            0
423  2019-05-29          0                0               0            0
424  2019-05-30          0                0               0            0
425  2019-05-31          0                0               0            0

     revenue
416      0.0
417      0.0
418      0.0
419      0.0
420      0.0
421      0.0
422      0.0
423      0.0
424      0.0
425      0.0

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 426 entries, 0 to 425
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             426 non-null    object
 1   purchases        426 non-null    int64
 2   unique_invoices  426 non-null    int64
 3   unique_streams   426 non-null    int64
 4   total_views      426 non-null    int64
 5   revenue          426 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 20.1+ KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 426 entries, 0 to 425
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             426 non-null    object
 1   purchases        426 non-null    int64
 2   unique_invoices  426 non-null    int64
 3   unique_streams   426 non-null    int64
 4   total_views      426 non-null    int64
```

```
 5   revenue          426 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 20.1+ KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT CATEGORICAL OBJECT COLUMNS
              date
count          426
unique         426
top     2019-05-31
freq             1
date : 426
purchases : 7
unique_invoices : 3
unique_streams : 7
total_views : 9
revenue : 9

There is no column that includes a unique category of data


6 - Dataframe df_ts_netherlands data:

FIRST 10 ROWS OF THE DATA
         date  purchases  unique_invoices  unique_streams  total_views  \
0  2017-11-01          0                0               0            0
1  2017-11-02          0                0               0            0
2  2017-11-03          0                0               0            0
3  2017-11-04          0                0               0            0
4  2017-11-05          0                0               0            0
5  2017-11-06          0                0               0            0
6  2017-11-07          0                0               0            0
7  2017-11-08          0                0               0            0
8  2017-11-09          0                0               0            0
9  2017-11-10          0                0               0            0

   revenue
0      0.0
1      0.0
2      0.0
3      0.0
4      0.0
5      0.0
6      0.0
7      0.0
8      0.0
9      0.0

LAST 10 ROWS OF THE DATA
           date  purchases  unique_invoices  unique_streams  total_views  \
```

```
597  2019-06-21          0                0              0            0
598  2019-06-22          0                0              0            0
599  2019-06-23          0                0              0            0
600  2019-06-24          0                0              0            0
601  2019-06-25        103                3             99         1244
602  2019-06-26          0                0              0            0
603  2019-06-27          0                0              0            0
604  2019-06-28          0                0              0            0
605  2019-06-29          0                0              0            0
606  2019-06-30          0                0              0            0

     revenue
597     0.00
598     0.00
599     0.00
600     0.00
601   248.66
602     0.00
603     0.00
604     0.00
605     0.00
606     0.00

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             607 non-null    object
 1   purchases        607 non-null    int64
 2   unique_invoices  607 non-null    int64
 3   unique_streams   607 non-null    int64
 4   total_views      607 non-null    int64
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             607 non-null    object
 1   purchases        607 non-null    int64
 2   unique_invoices  607 non-null    int64
 3   unique_streams   607 non-null    int64
 4   total_views      607 non-null    int64
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

MISSING DATA
```

There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT CATEGORICAL OBJECT COLUMNS

	date
count	607
unique	607
top	2019-06-30
freq	1

date : 607
purchases : 62
unique_invoices : 6
unique_streams : 62
total_views : 121
revenue : 131

There is no column that includes a unique category of data

7 - Dataframe df_ts_norway data:

FIRST 10 ROWS OF THE DATA

	date	purchases	unique_invoices	unique_streams	total_views	\
0	2017-12-01	0	0	0	0	
1	2017-12-02	0	0	0	0	
2	2017-12-03	0	0	0	0	
3	2017-12-04	21	1	21	175	
4	2017-12-05	0	0	0	0	
5	2017-12-06	0	0	0	0	
6	2017-12-07	0	0	0	0	
7	2017-12-08	0	0	0	0	
8	2017-12-09	0	0	0	0	
9	2017-12-10	0	0	0	0	

	revenue
0	0.00
1	0.00
2	0.00
3	123.28
4	0.00
5	0.00
6	0.00
7	0.00
8	0.00
9	0.00

LAST 10 ROWS OF THE DATA

	date	purchases	unique_invoices	unique_streams	total_views	\
567	2019-06-21	0	0	0	0	
568	2019-06-22	0	0	0	0	
569	2019-06-23	0	0	0	0	
570	2019-06-24	0	0	0	0	
571	2019-06-25	0	0	0	0	
572	2019-06-26	0	0	0	0	

```
573  2019-06-27          0                0              0           0
574  2019-06-28          0                0              0           0
575  2019-06-29          0                0              0           0
576  2019-06-30          0                0              0           0

     revenue
567      0.0
568      0.0
569      0.0
570      0.0
571      0.0
572      0.0
573      0.0
574      0.0
575      0.0
576      0.0

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 577 entries, 0 to 576
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             577 non-null    object
 1   purchases        577 non-null    int64
 2   unique_invoices  577 non-null    int64
 3   unique_streams   577 non-null    int64
 4   total_views      577 non-null    int64
 5   revenue          577 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 27.2+ KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 577 entries, 0 to 576
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             577 non-null    object
 1   purchases        577 non-null    int64
 2   unique_invoices  577 non-null    int64
 3   unique_streams   577 non-null    int64
 4   total_views      577 non-null    int64
 5   revenue          577 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 27.2+ KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT CATEGORICAL OBJECT COLUMNS
```

```
              date
count          577
unique         577
top     2019-06-30
freq             1
date : 577
purchases : 18
unique_invoices : 5
unique_streams : 16
total_views : 19
revenue : 19

There is no column that includes a unique category of data


8 - Dataframe df_ts_portugal data:

FIRST 10 ROWS OF THE DATA
         date  purchases  unique_invoices  unique_streams  total_views  \
0  2017-11-01          0                0               0            0
1  2017-11-02          0                0               0            0
2  2017-11-03          0                0               0            0
3  2017-11-04          0                0               0            0
4  2017-11-05          0                0               0            0
5  2017-11-06          0                0               0            0
6  2017-11-07          0                0               0            0
7  2017-11-08          0                0               0            0
8  2017-11-09          0                0               0            0
9  2017-11-10          0                0               0            0

   revenue
0      0.0
1      0.0
2      0.0
3      0.0
4      0.0
5      0.0
6      0.0
7      0.0
8      0.0
9      0.0

LAST 10 ROWS OF THE DATA
           date  purchases  unique_invoices  unique_streams  total_views  \
597  2019-06-21          1                1               1            4
598  2019-06-22          0                0               0            0
599  2019-06-23          0                0               0            0
600  2019-06-24          0                0               0            0
601  2019-06-25          0                0               0            0
602  2019-06-26          0                0               0            0
603  2019-06-27          0                0               0            0
604  2019-06-28          0                0               0            0
605  2019-06-29          0                0               0            0
606  2019-06-30          0                0               0            0

     revenue
```

```
597     4.95
598     0.00
599     0.00
600     0.00
601     0.00
602     0.00
603     0.00
604     0.00
605     0.00
606     0.00

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             607 non-null    object
 1   purchases        607 non-null    int64
 2   unique_invoices  607 non-null    int64
 3   unique_streams   607 non-null    int64
 4   total_views      607 non-null    int64
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             607 non-null    object
 1   purchases        607 non-null    int64
 2   unique_invoices  607 non-null    int64
 3   unique_streams   607 non-null    int64
 4   total_views      607 non-null    int64
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT CATEGORICAL OBJECT COLUMNS
              date
count          607
unique         607
top     2019-06-30
freq             1
date : 607
```

```
purchases : 39
unique_invoices : 5
unique_streams : 40
total_views : 60
revenue : 66

There is no column that includes a unique category of data


9 - Dataframe df_ts_singapore data:

FIRST 10 ROWS OF THE DATA
         date  purchases  unique_invoices  unique_streams  total_views  \
0  2018-04-01          0                0               0            0
1  2018-04-02          0                0               0            0
2  2018-04-03          0                0               0            0
3  2018-04-04          0                0               0            0
4  2018-04-05          0                0               0            0
5  2018-04-06          0                0               0            0
6  2018-04-07          0                0               0            0
7  2018-04-08          0                0               0            0
8  2018-04-09          0                0               0            0
9  2018-04-10          0                0               0            0

   revenue
0      0.0
1      0.0
2      0.0
3      0.0
4      0.0
5      0.0
6      0.0
7      0.0
8      0.0
9      0.0

LAST 10 ROWS OF THE DATA
           date  purchases  unique_invoices  unique_streams  total_views  \
446  2019-06-21          0                0               0            0
447  2019-06-22          0                0               0            0
448  2019-06-23          0                0               0            0
449  2019-06-24          0                0               0            0
450  2019-06-25          0                0               0            0
451  2019-06-26          0                0               0            0
452  2019-06-27          0                0               0            0
453  2019-06-28          0                0               0            0
454  2019-06-29          0                0               0            0
455  2019-06-30          0                0               0            0

     revenue
446      0.0
447      0.0
448      0.0
449      0.0
450      0.0
451      0.0
```

```
452      0.0
453      0.0
454      0.0
455      0.0

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 456 entries, 0 to 455
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             456 non-null    object
 1   purchases        456 non-null    int64
 2   unique_invoices  456 non-null    int64
 3   unique_streams   456 non-null    int64
 4   total_views      456 non-null    int64
 5   revenue          456 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 21.5+ KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 456 entries, 0 to 455
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             456 non-null    object
 1   purchases        456 non-null    int64
 2   unique_invoices  456 non-null    int64
 3   unique_streams   456 non-null    int64
 4   total_views      456 non-null    int64
 5   revenue          456 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 21.5+ KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT CATEGORICAL OBJECT COLUMNS
              date
count          456
unique         456
top     2019-06-30
freq             1
date : 456
purchases : 6
unique_invoices : 4
unique_streams : 6
total_views : 6
revenue : 6
```

There is no column that includes a unique category of data

10 - Dataframe df_ts_spain data:

FIRST 10 ROWS OF THE DATA

	date	purchases	unique_invoices	unique_streams	total_views	revenue
0	2017-11-01	0	0	0	0	0.0
1	2017-11-02	0	0	0	0	0.0
2	2017-11-03	0	0	0	0	0.0
3	2017-11-04	0	0	0	0	0.0
4	2017-11-05	0	0	0	0	0.0
5	2017-11-06	0	0	0	0	0.0
6	2017-11-07	0	0	0	0	0.0
7	2017-11-08	0	0	0	0	0.0
8	2017-11-09	0	0	0	0	0.0
9	2017-11-10	0	0	0	0	0.0

LAST 10 ROWS OF THE DATA

	date	purchases	unique_invoices	unique_streams	total_views	revenue
597	2019-06-21	0	0	0	0	0.0
598	2019-06-22	0	0	0	0	0.0
599	2019-06-23	5	1	5	49	33.6
600	2019-06-24	0	0	0	0	0.0
601	2019-06-25	0	0	0	0	0.0
602	2019-06-26	0	0	0	0	0.0
603	2019-06-27	0	0	0	0	0.0
604	2019-06-28	0	0	0	0	0.0
605	2019-06-29	0	0	0	0	0.0
606	2019-06-30	0	0	0	0	0.0

INFO ABOUT THE DATA

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   date             607 non-null    object 
 1   purchases        607 non-null    int64  
 2   unique_invoices  607 non-null    int64  
 3   unique_streams   607 non-null    int64  
 4   total_views      607 non-null    int64  
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   date             607 non-null    object 
 1   purchases        607 non-null    int64  
 2   unique_invoices  607 non-null    int64  
 3   unique_streams   607 non-null    int64  
 4   total_views      607 non-null    int64  
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT CATEGORICAL OBJECT COLUMNS
              date
count          607
unique         607
top     2019-06-30
freq             1
date : 607
purchases : 47
unique_invoices : 5
unique_streams : 49
total_views : 86
revenue : 96

There is no column that includes a unique category of data


11 - Dataframe df_ts_united_kingdom data:

FIRST 10 ROWS OF THE DATA
```

```
         date  purchases  unique_invoices  unique_streams  total_views  \
0  2017-11-01          0                0               0            0
1  2017-11-02          0                0               0            0
2  2017-11-03          0                0               0            0
3  2017-11-04          0                0               0            0
4  2017-11-05          0                0               0            0
5  2017-11-06          0                0               0            0
6  2017-11-07          0                0               0            0
7  2017-11-08          0                0               0            0
8  2017-11-09          0                0               0            0
9  2017-11-10          0                0               0            0

   revenue
0      0.0
1      0.0
2      0.0
3      0.0
4      0.0
5      0.0
6      0.0
7      0.0
8      0.0
9      0.0

LAST 10 ROWS OF THE DATA
           date  purchases  unique_invoices  unique_streams  total_views  \
597  2019-06-21        903               54             602         5207
598  2019-06-22          0                0               0            0
599  2019-06-23        663               33             534         2397
600  2019-06-24       1116               58             708         4715
601  2019-06-25        922               80             611         5442
602  2019-06-26       1273               62             965         5701
603  2019-06-27       1480               72             897         8220
604  2019-06-28        907               62             565         4653
605  2019-06-29          0                0               0            0
606  2019-06-30        581               27             423         2508

      revenue
597  13706.03
598      0.00
599   2431.59
600   4441.64
601   4970.18
602   4610.15
603   5105.97
604   3061.10
605      0.00
606   1741.25

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             607 non-null    object
```

```
 1   purchases        607 non-null    int64
 2   unique_invoices  607 non-null    int64
 3   unique_streams   607 non-null    int64
 4   total_views      607 non-null    int64
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             607 non-null    object
 1   purchases        607 non-null    int64
 2   unique_invoices  607 non-null    int64
 3   unique_streams   607 non-null    int64
 4   total_views      607 non-null    int64
 5   revenue          607 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 28.6+ KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT CATEGORICAL OBJECT COLUMNS
              date
count          607
unique         607
top     2019-06-30
freq             1
date : 607
purchases : 428
unique_invoices : 132
unique_streams : 361
total_views : 460
revenue : 469

There is no column that includes a unique category of data
```

In [15]:
```
#Convert data to the correct datatype
for i in range (0,size):
    df_name = list_ts_df[i]
    (library_ts_df[df_name])['date'] = pd.to_datetime((library_ts_df[df_name])['dat
    (library_ts_df[df_name])['purchases'] = (library_ts_df[df_name])['purchases'].a
    (library_ts_df[df_name])['unique_invoices'] = (library_ts_df[df_name])['unique_
    (library_ts_df[df_name])['unique_streams'] = (library_ts_df[df_name])['unique_s
    (library_ts_df[df_name])['total_views'] = (library_ts_df[df_name])['total_views
```

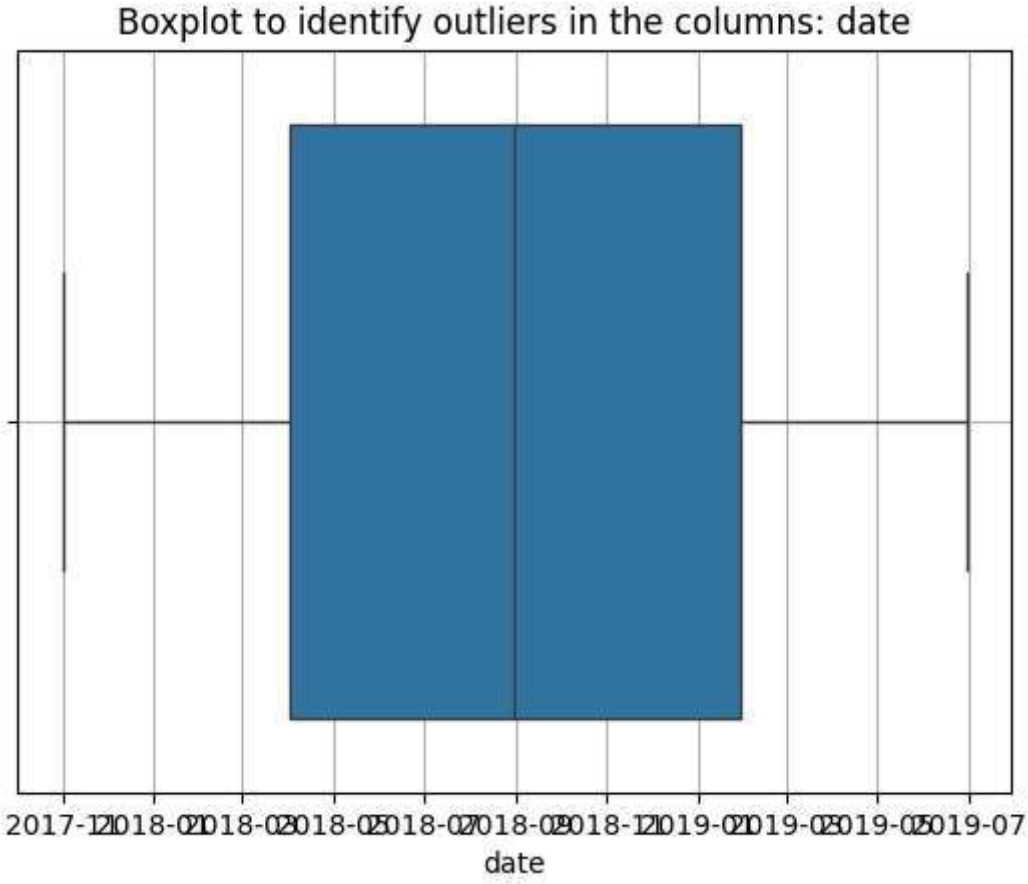
```
#Check dtypes
for i in range (0,size):
    df_name = list_ts_df[i]
    print((library_ts_df[df_name]).dtypes)
```

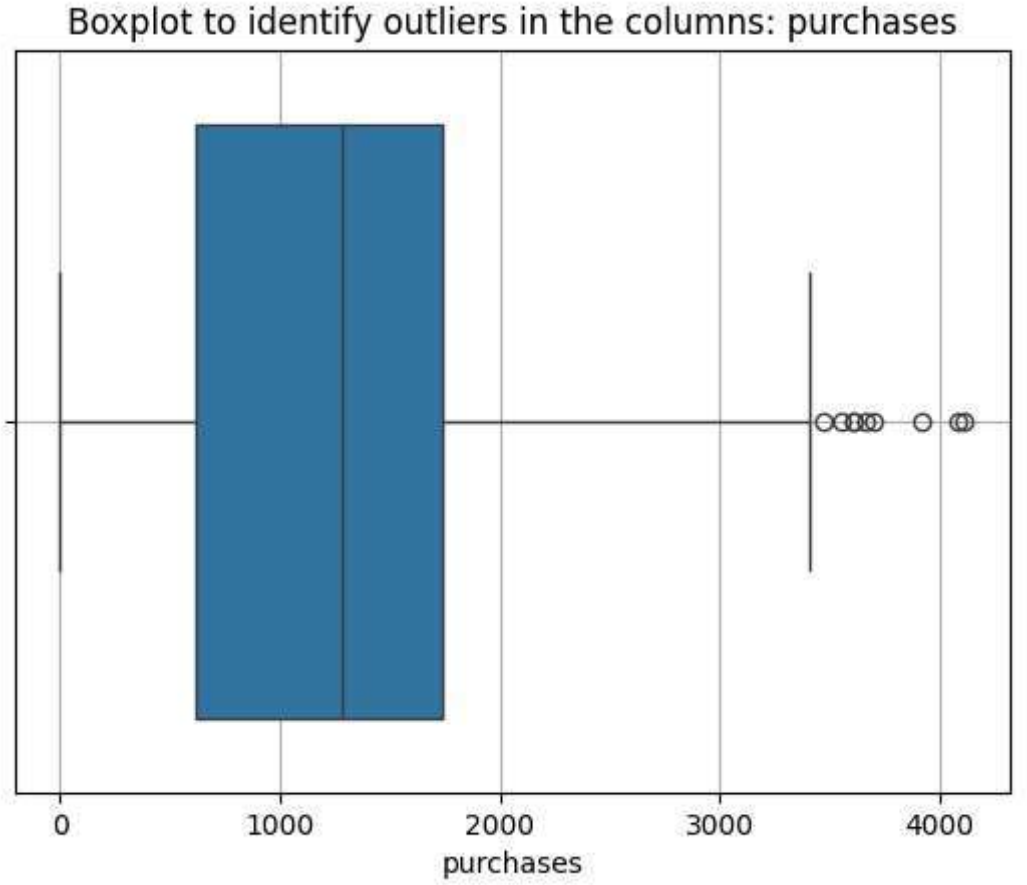
```
date               datetime64[ns]
purchases                 float64
unique_invoices           float64
unique_streams            float64
total_views               float64
revenue                   float64
dtype: object
date               datetime64[ns]
purchases                 float64
unique_invoices           float64
unique_streams            float64
total_views               float64
revenue                   float64
dtype: object
date               datetime64[ns]
purchases                 float64
unique_invoices           float64
unique_streams            float64
total_views               float64
revenue                   float64
dtype: object
date               datetime64[ns]
purchases                 float64
unique_invoices           float64
unique_streams            float64
total_views               float64
revenue                   float64
dtype: object
date               datetime64[ns]
purchases                 float64
unique_invoices           float64
unique_streams            float64
total_views               float64
revenue                   float64
dtype: object
date               datetime64[ns]
purchases                 float64
unique_invoices           float64
unique_streams            float64
total_views               float64
revenue                   float64
dtype: object
date               datetime64[ns]
purchases                 float64
unique_invoices           float64
unique_streams            float64
total_views               float64
revenue                   float64
dtype: object
date               datetime64[ns]
purchases                 float64
unique_invoices           float64
unique_streams            float64
total_views               float64
revenue                   float64
dtype: object
```

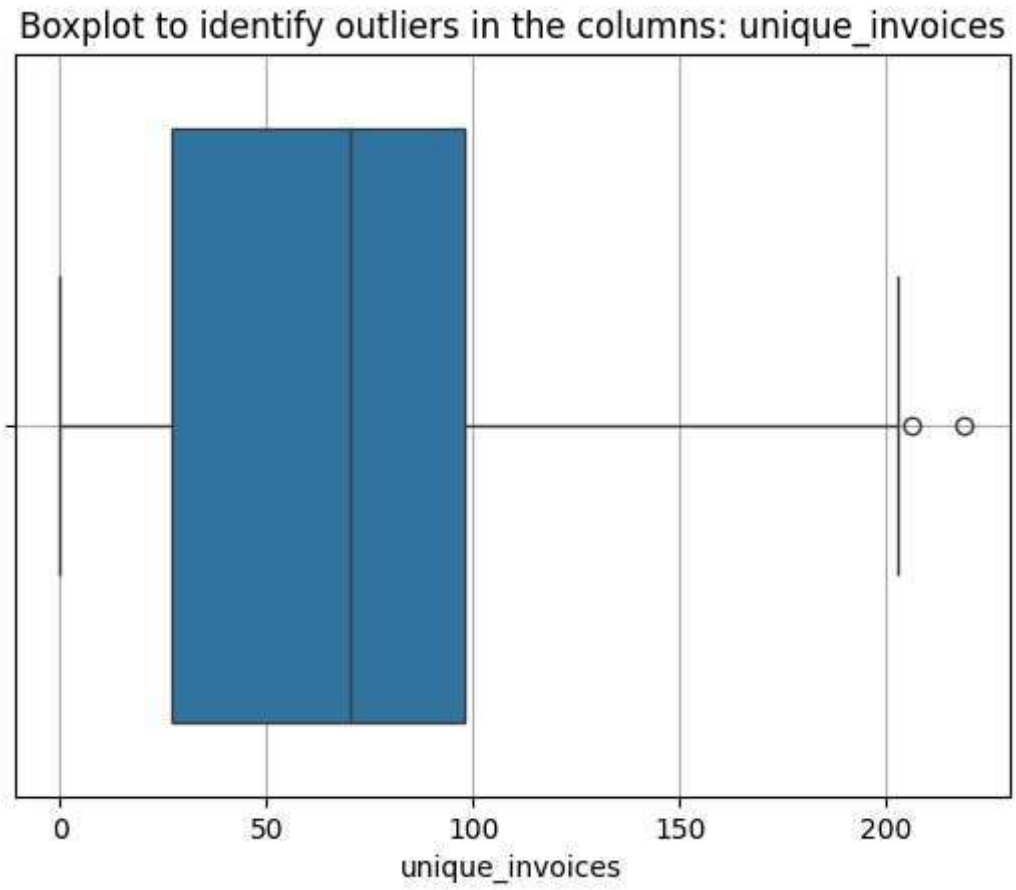
```
date                datetime64[ns]
purchases                  float64
unique_invoices            float64
unique_streams             float64
total_views                float64
revenue                    float64
dtype: object
date                datetime64[ns]
purchases                  float64
unique_invoices            float64
unique_streams             float64
total_views                float64
revenue                    float64
dtype: object
date                datetime64[ns]
purchases                  float64
unique_invoices            float64
unique_streams             float64
total_views                float64
revenue                    float64
dtype: object
```

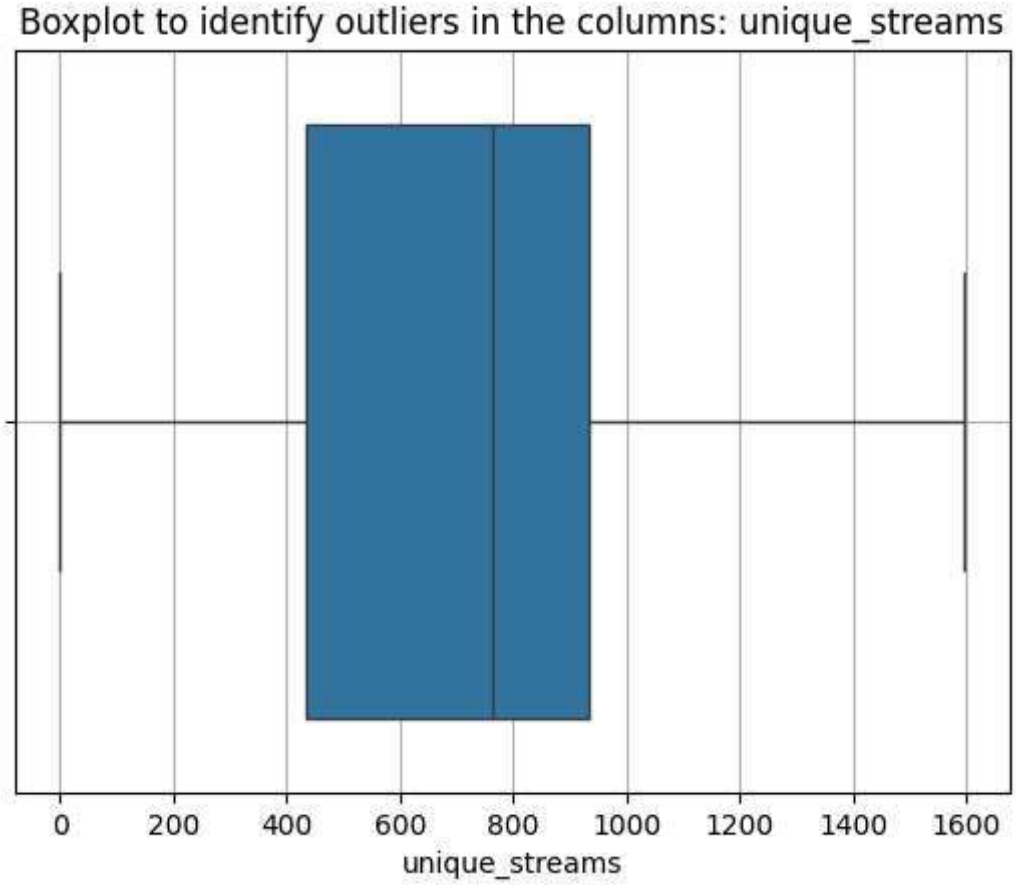
In [16]:
```
#Check for outliers
list_col_name = library_ts_df['df_ts_all'].columns
size_list_col_name = len (list_col_name)
for i in range (0,size):
    df_name = list_ts_df[i]
    print(i+1, '- Dataframe', df_name , 'data:')
    for j in range (size_list_col_name):
        viz.show_boxplot((library_ts_df[df_name]), list_col_name[j])
        print('\n')
```

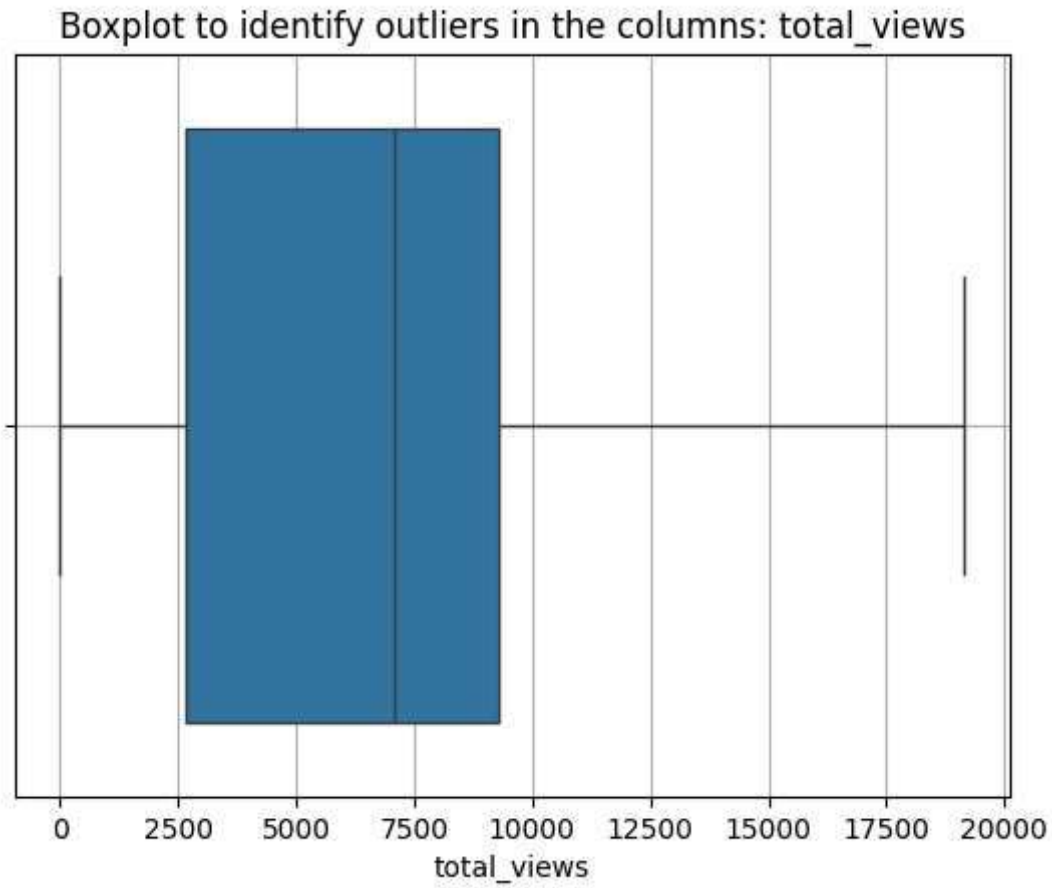
```
1 - Dataframe df_ts_all data:
```

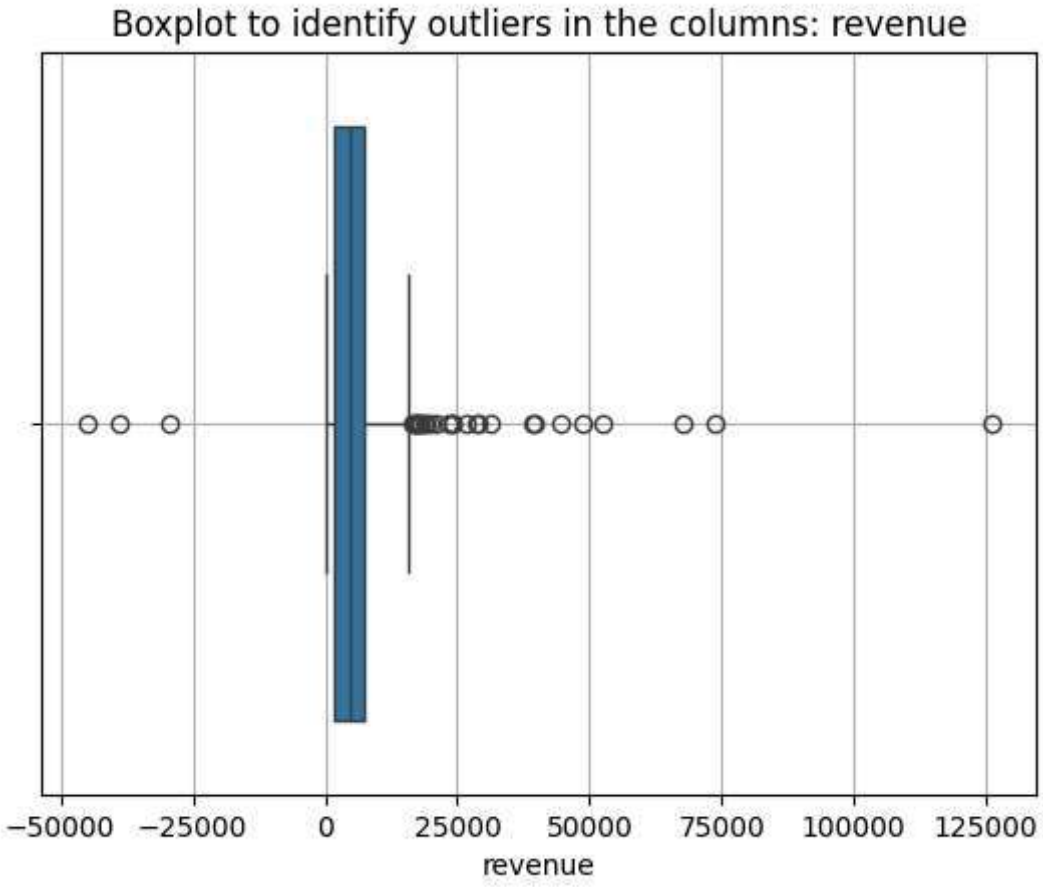

Boxplot to identify outliers in the columns: date
2017-12
2018-02
2018-03
2018-05
2018-07
2018-09
2018-12
2019-02
2019-03
2019-05
2019-07
date


Boxplot to identify outliers in the columns: purchases
0
1000
2000
3000
4000
purchases

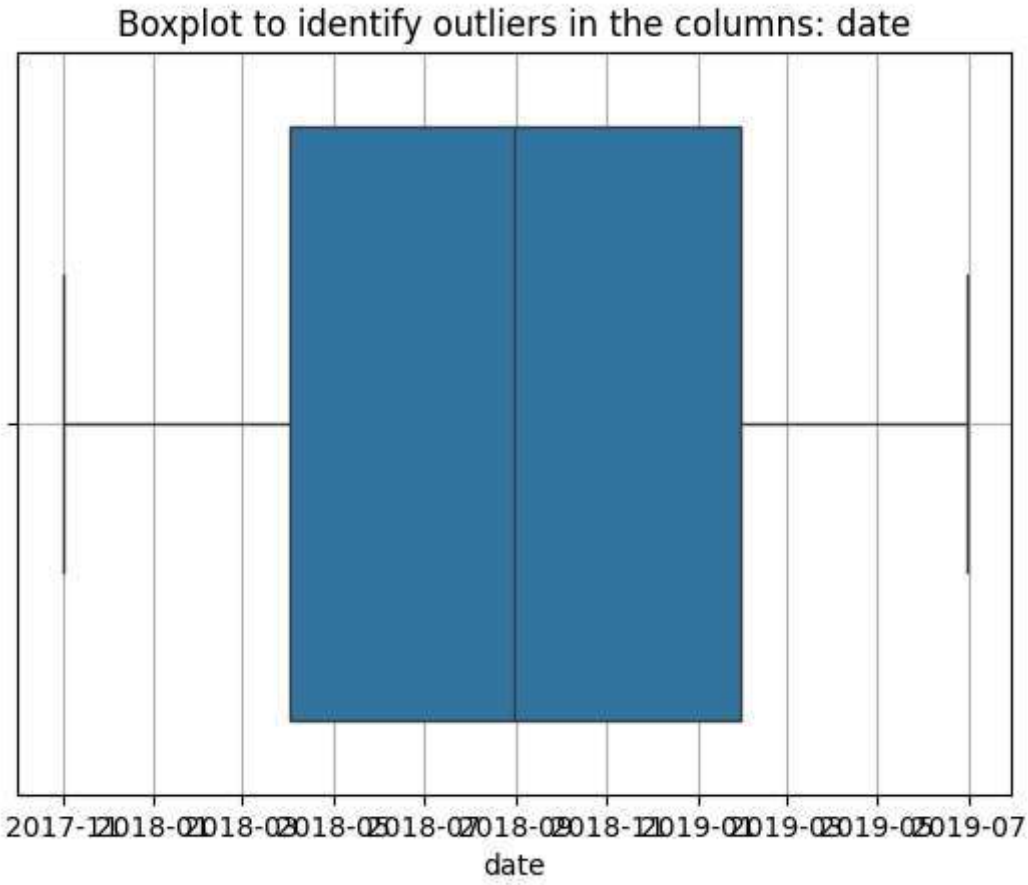

Boxplot to identify outliers in the columns: unique_invoices
0
50
100
150
200
unique_invoices

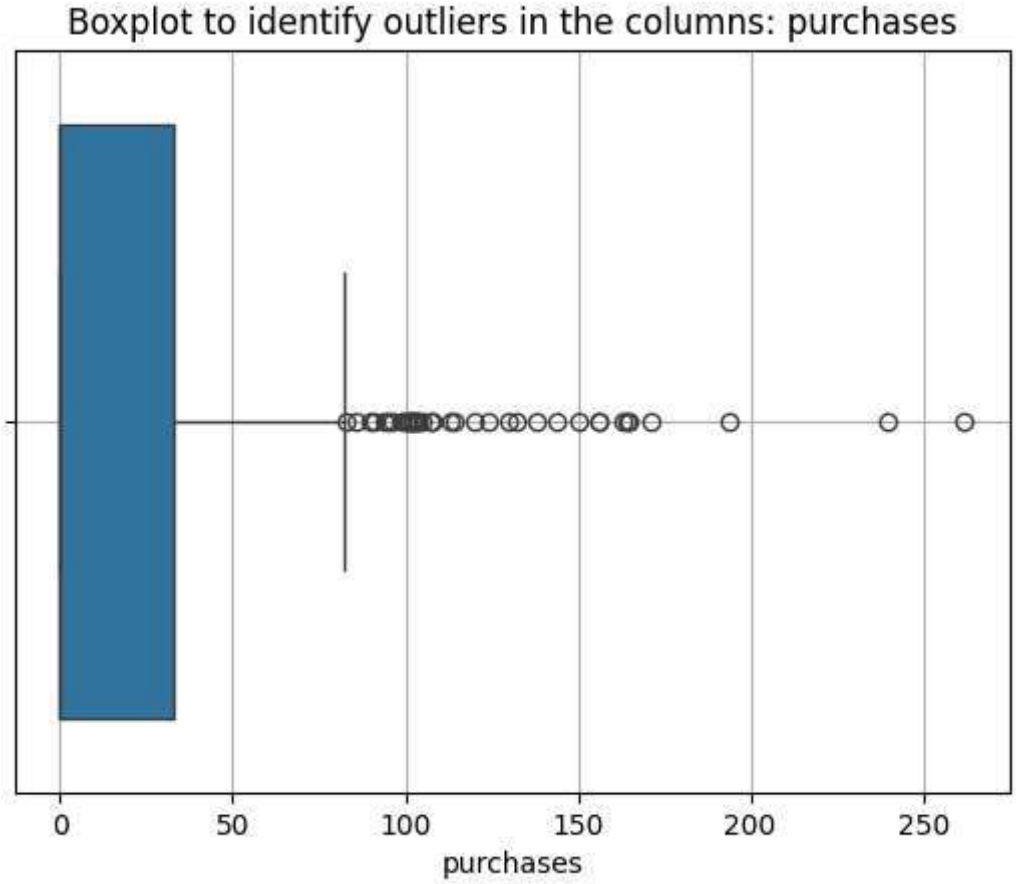

Boxplot to identify outliers in the columns: unique_streams
0
200
400
600
800
1000
1200
1400
1600
unique_streams

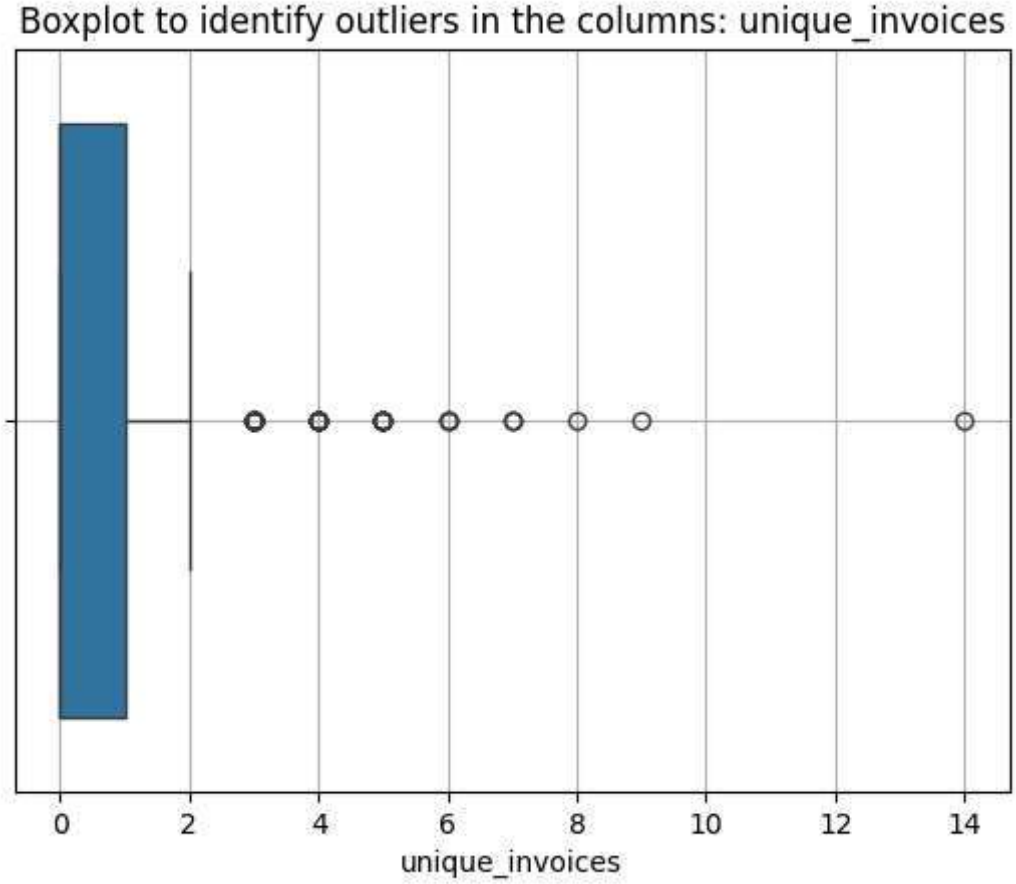

Boxplot to identify outliers in the columns: total_views
0
2500
5000
7500
10000
12500
15000
17500
20000
total_views

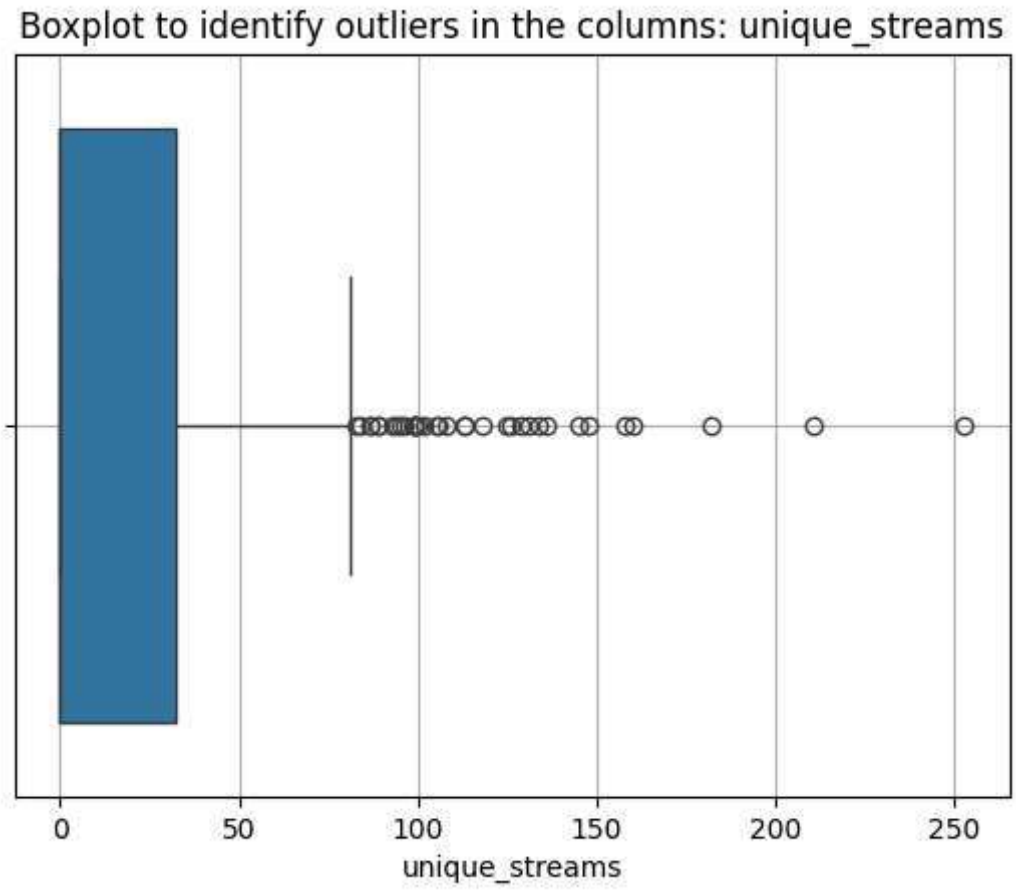

Boxplot to identify outliers in the columns: revenue
−50000
−25000
0
25000
50000
75000
100000
125000
revenue

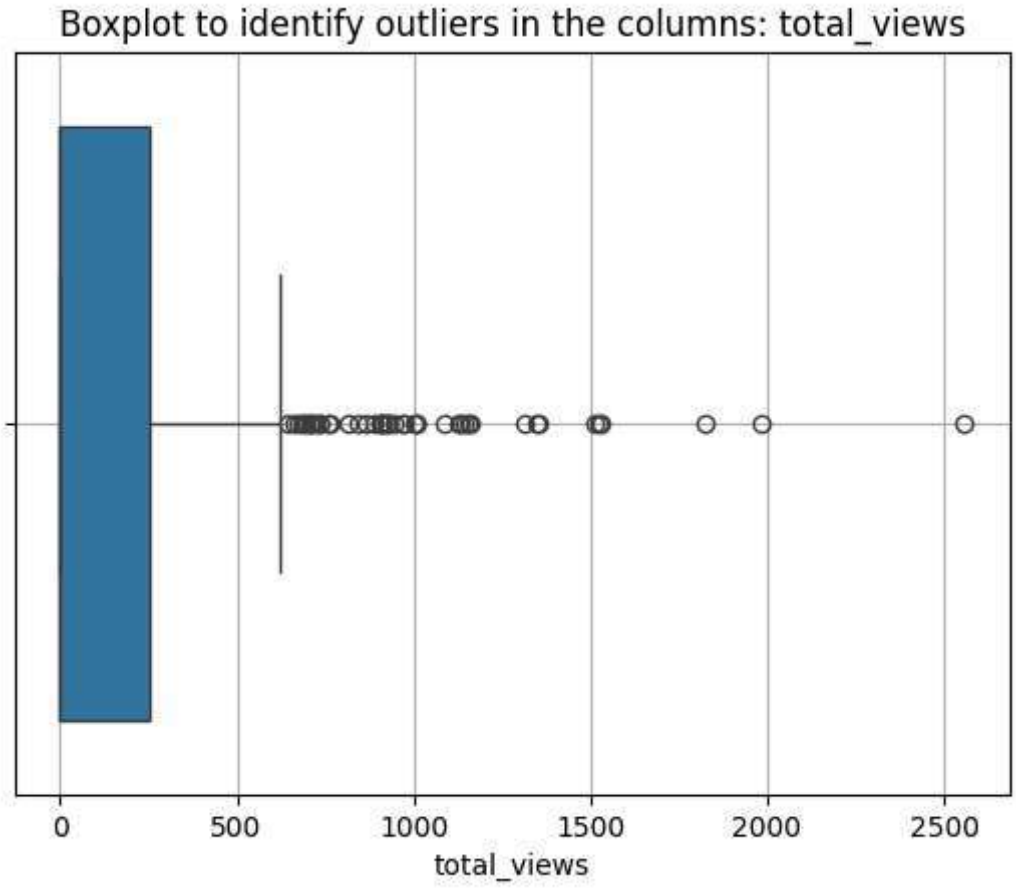
2 - Dataframe df_ts_eire data:

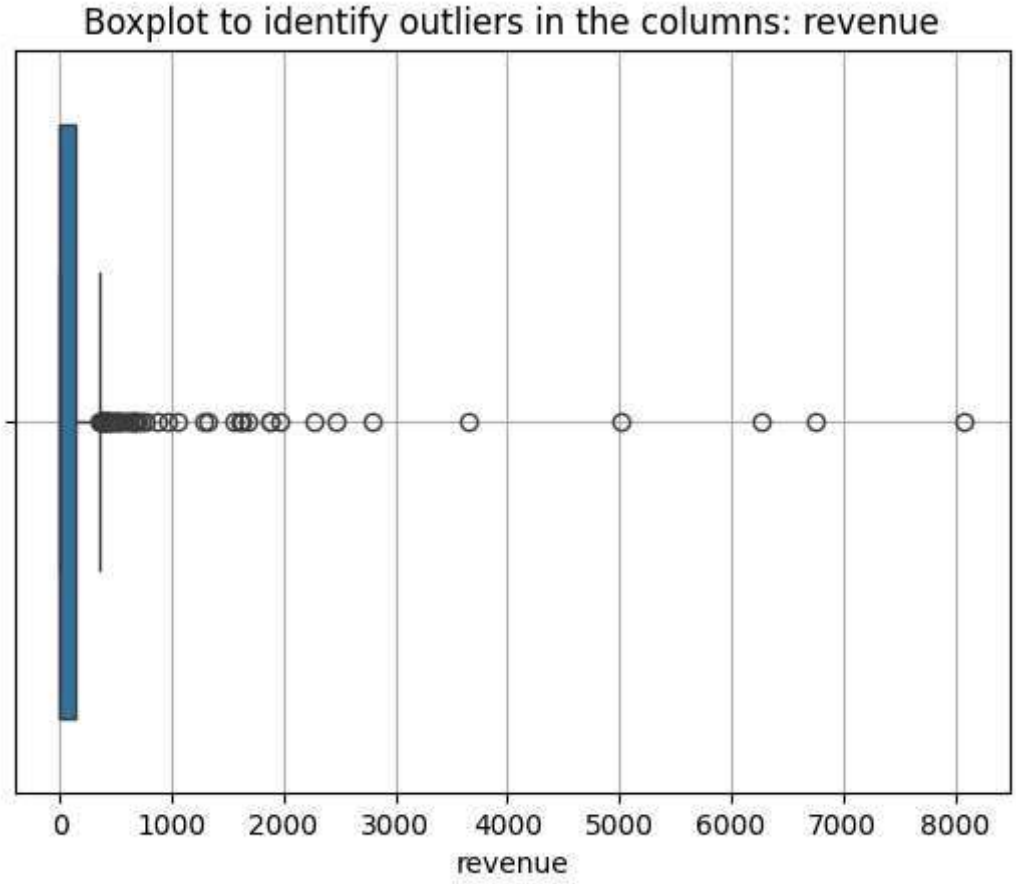



Boxplot to identify outliers in the columns: purchases
0
50
100
150
200
250
purchases

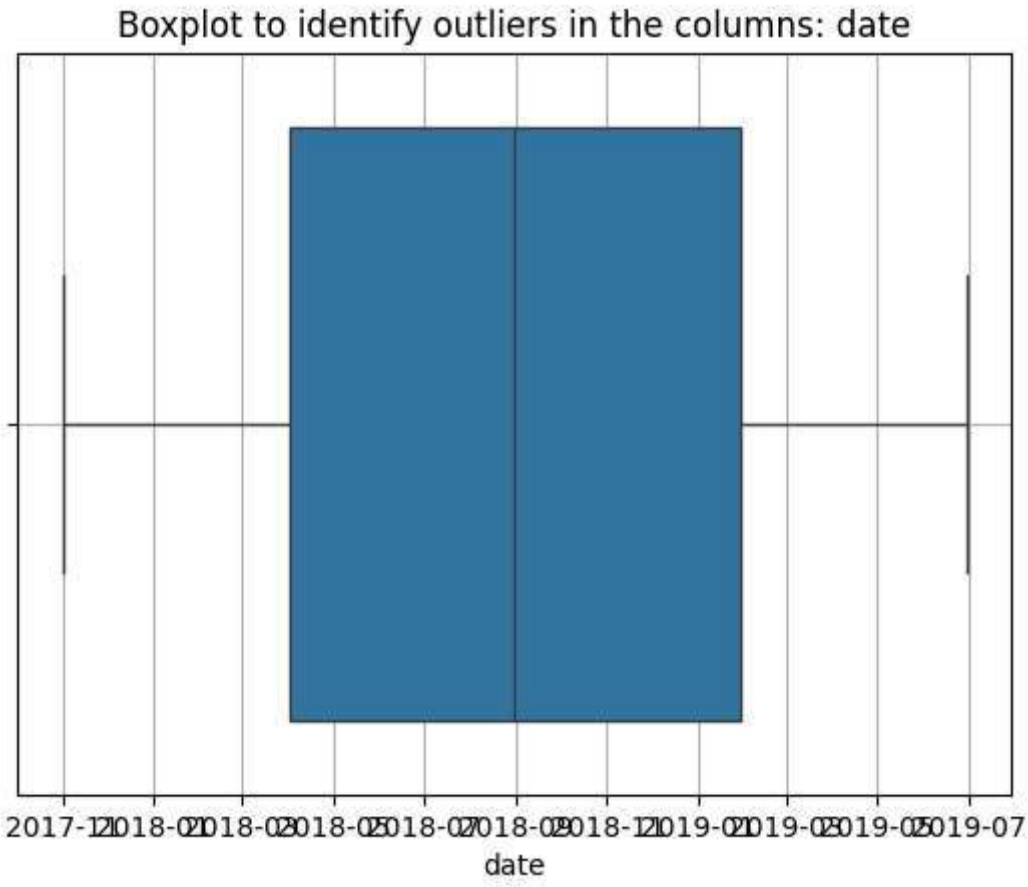

Boxplot to identify outliers in the columns: unique_invoices
0
2
4
6
8
10
12
14
unique_invoices

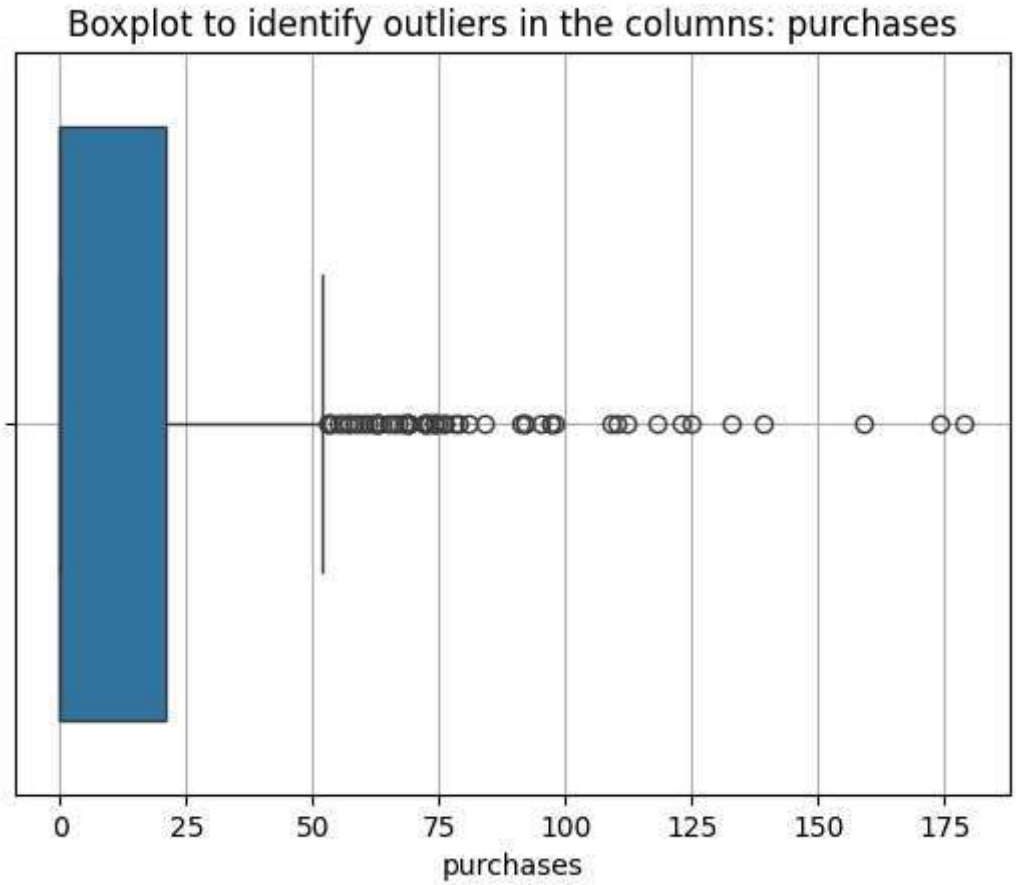

Boxplot to identify outliers in the columns: unique_streams
0
50
100
150
200
250
unique_streams

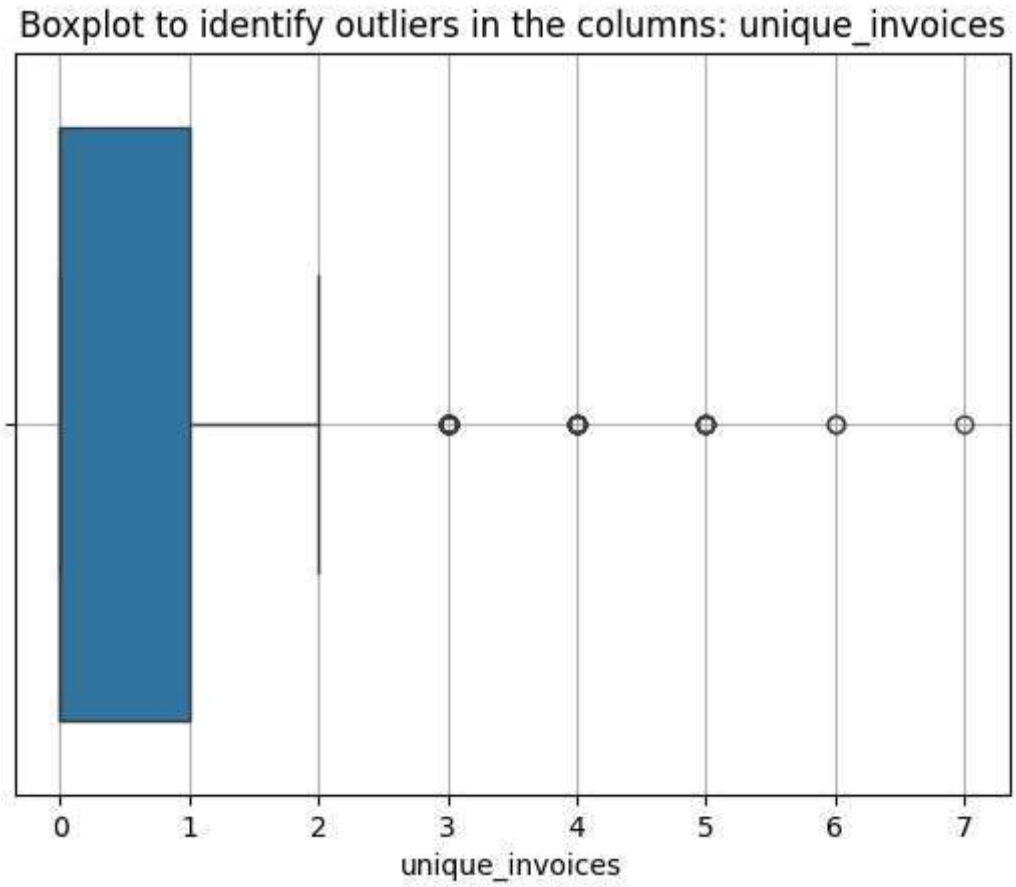

Boxplot to identify outliers in the columns: total_views
0
500
1000
1500
2000
2500
total_views

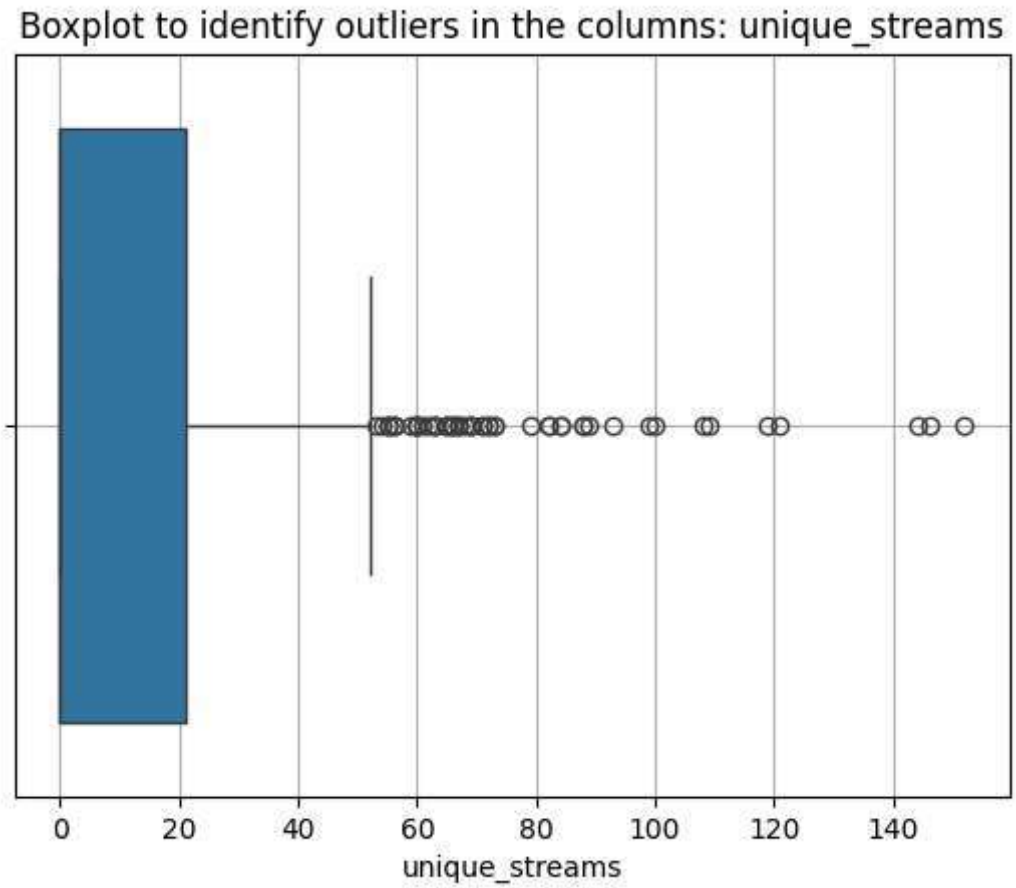


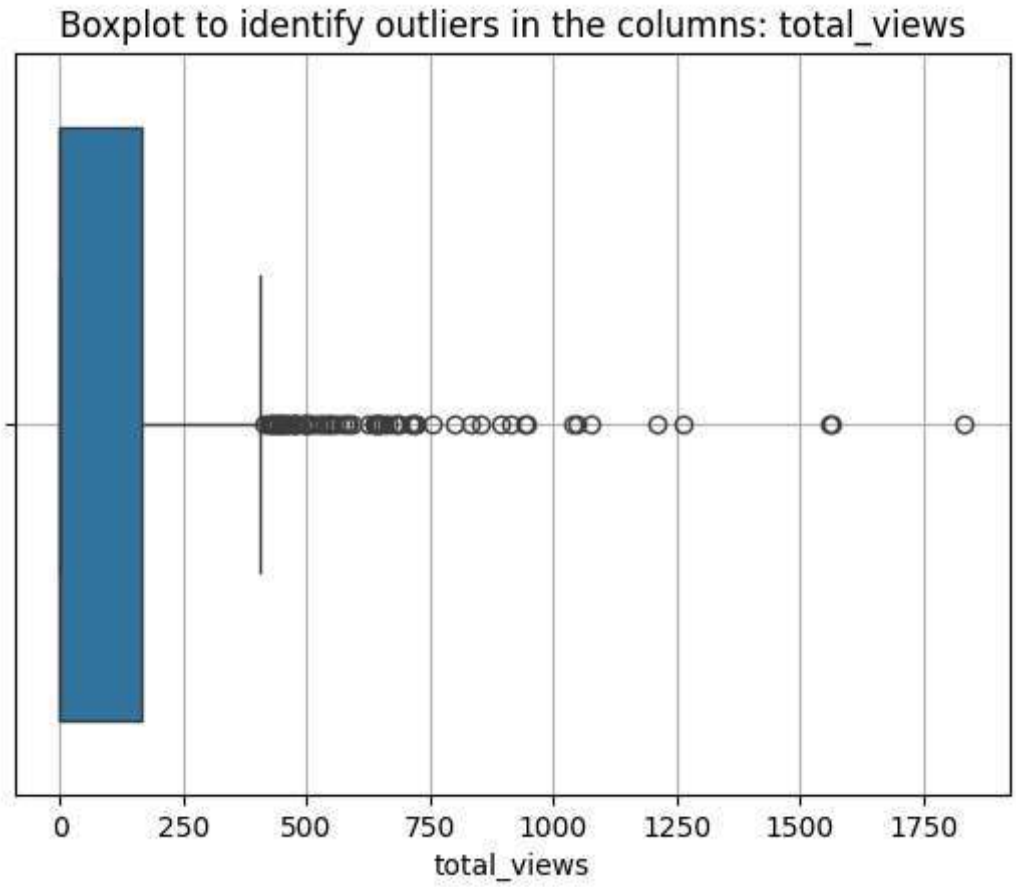
3 - Dataframe df_ts_france data:

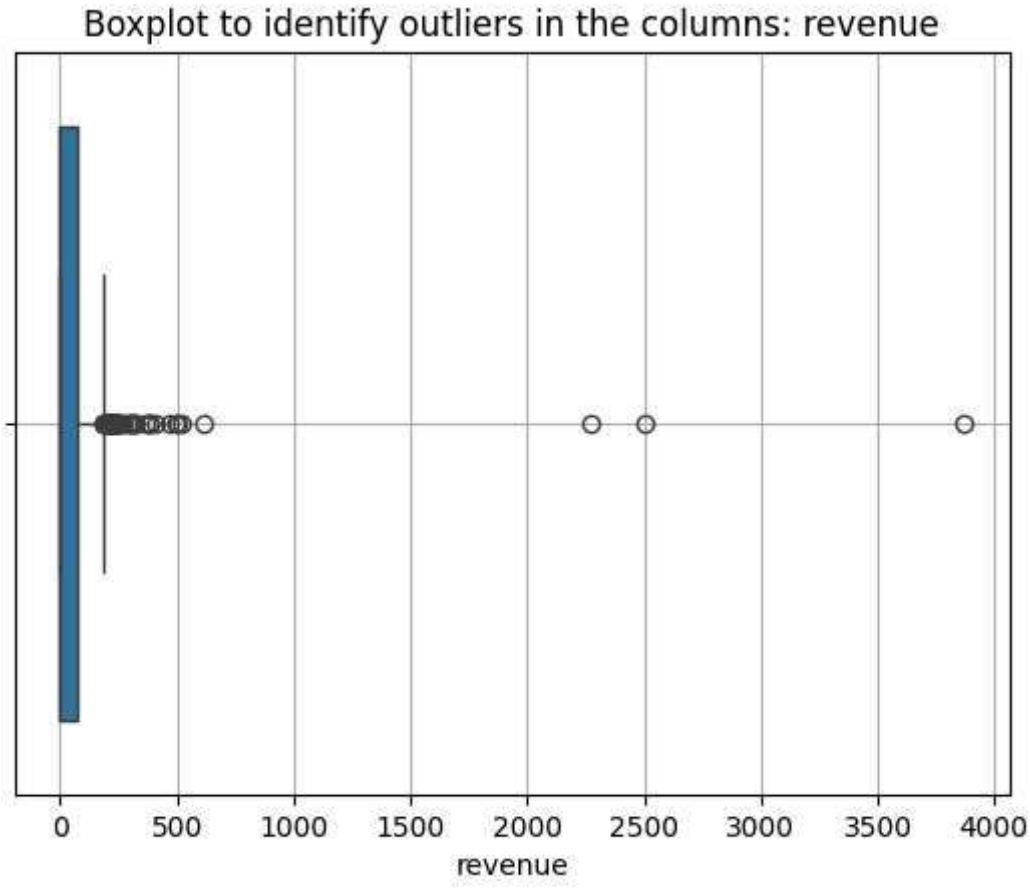



Boxplot to identify outliers in the columns: purchases
0
25
50
75
100
125
150
175
purchases

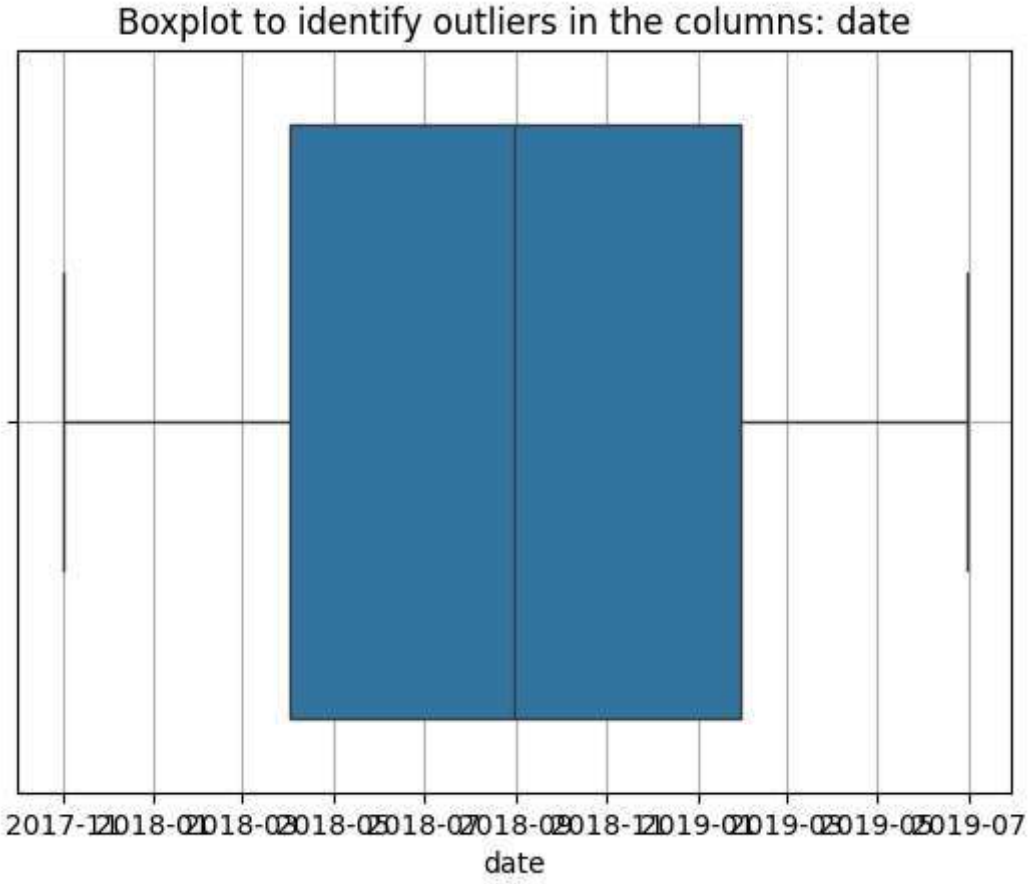

Boxplot to identify outliers in the columns: unique_invoices
0
1
2
3
4
5
6
7
unique_invoices

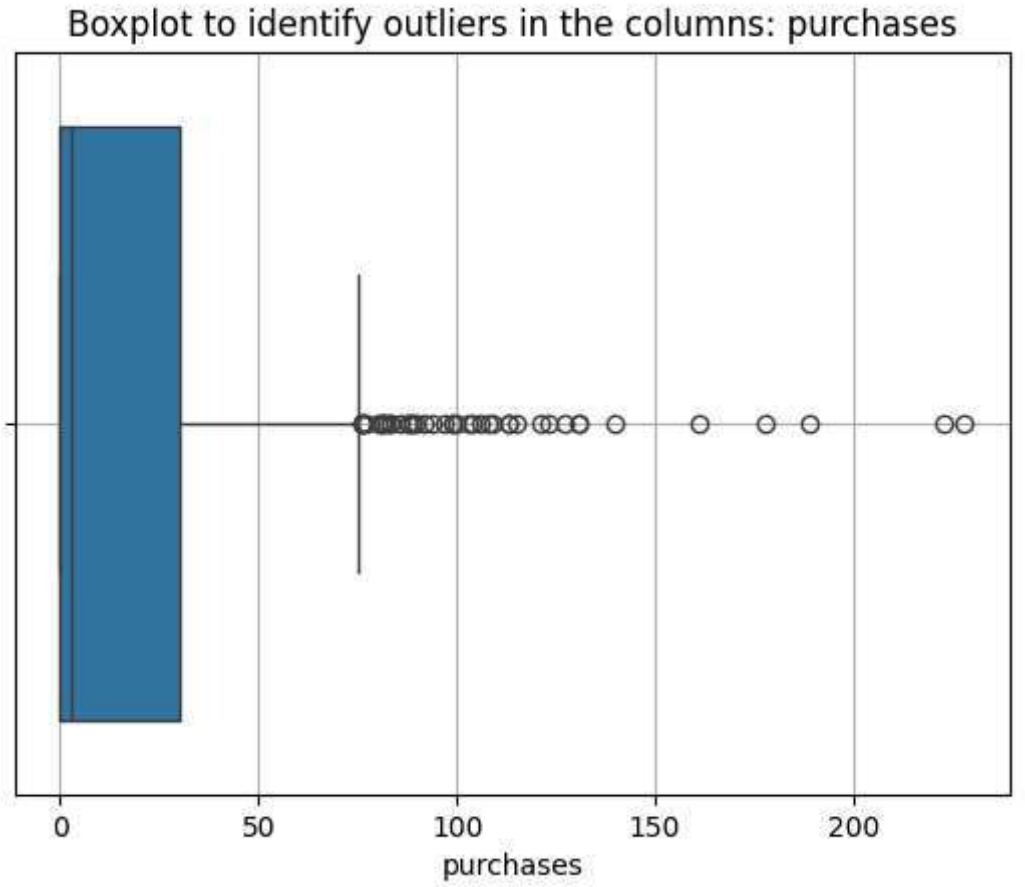

Boxplot to identify outliers in the columns: unique_streams
0
20
40
60
80
100
120
140
unique_streams

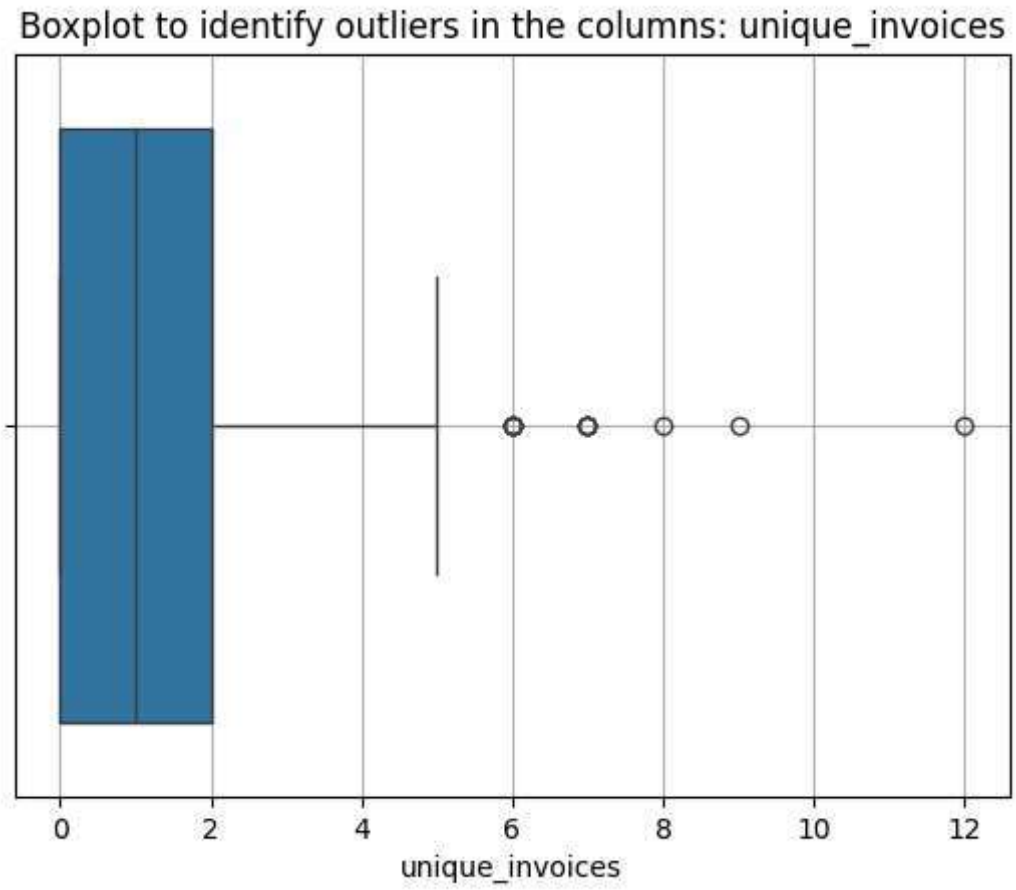

Boxplot to identify outliers in the columns: total_views
0
250
500
750
1000
1250
1500
1750
total_views

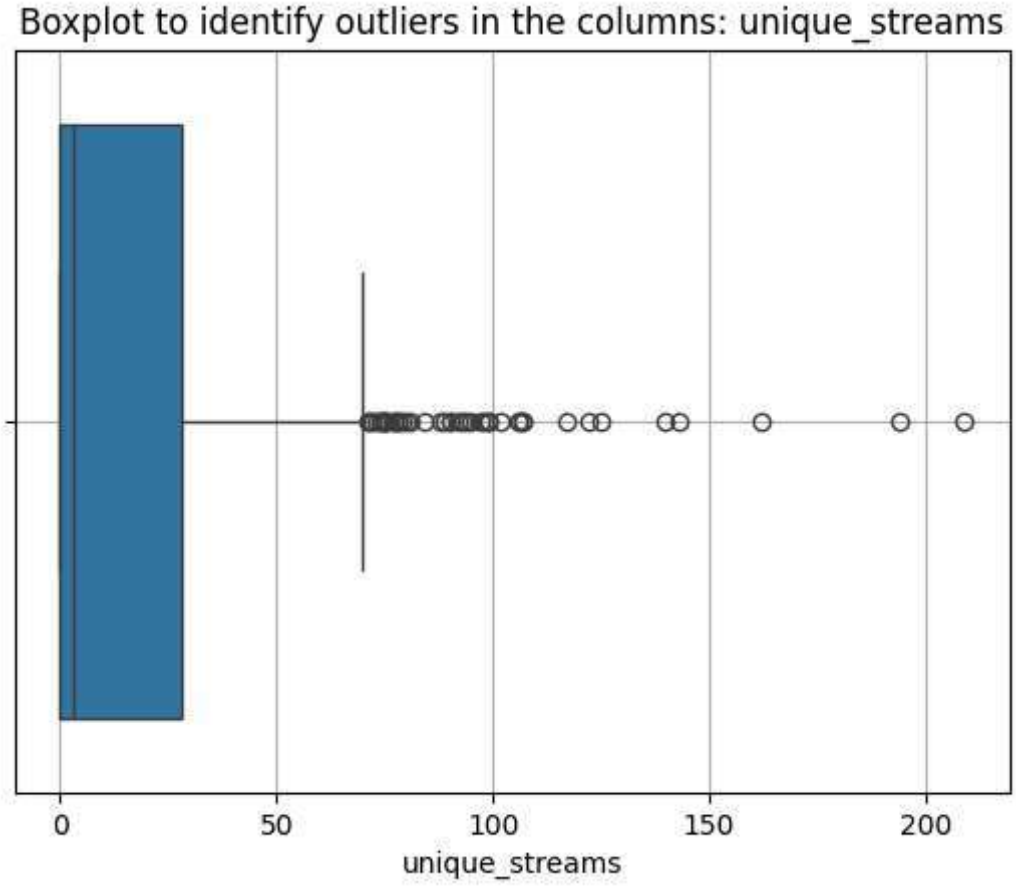


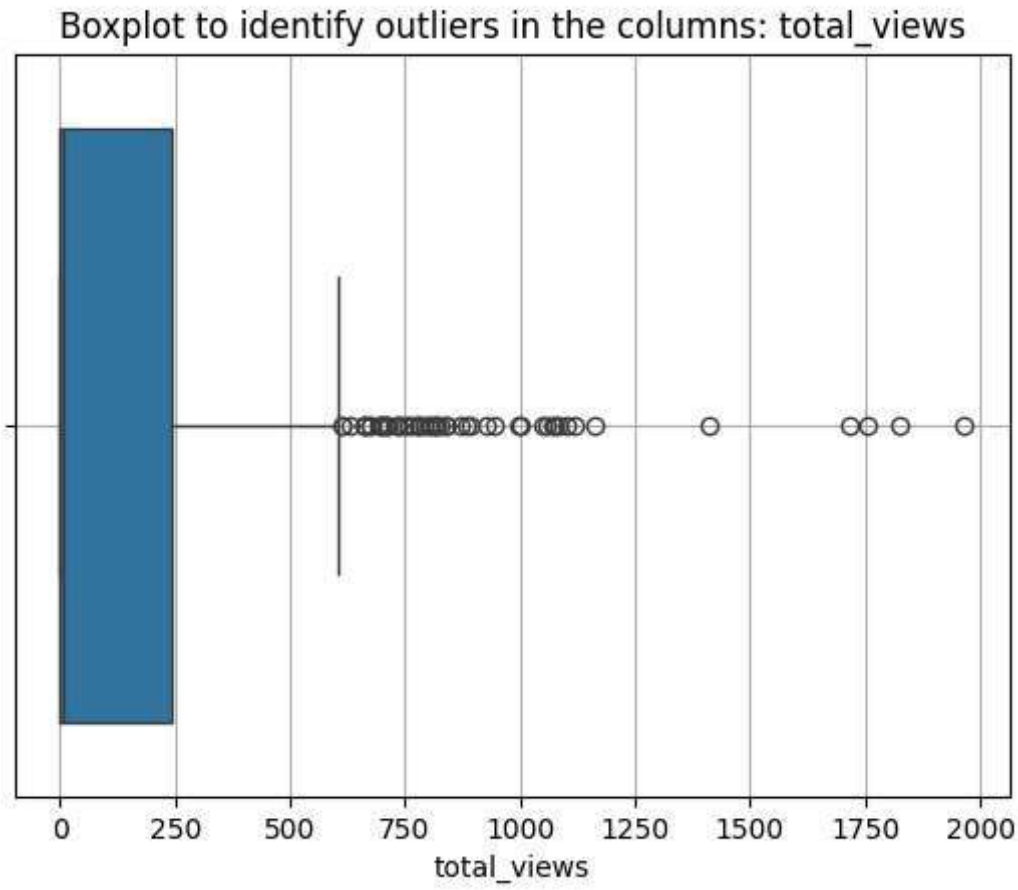
4 - Dataframe df_ts_germany data:

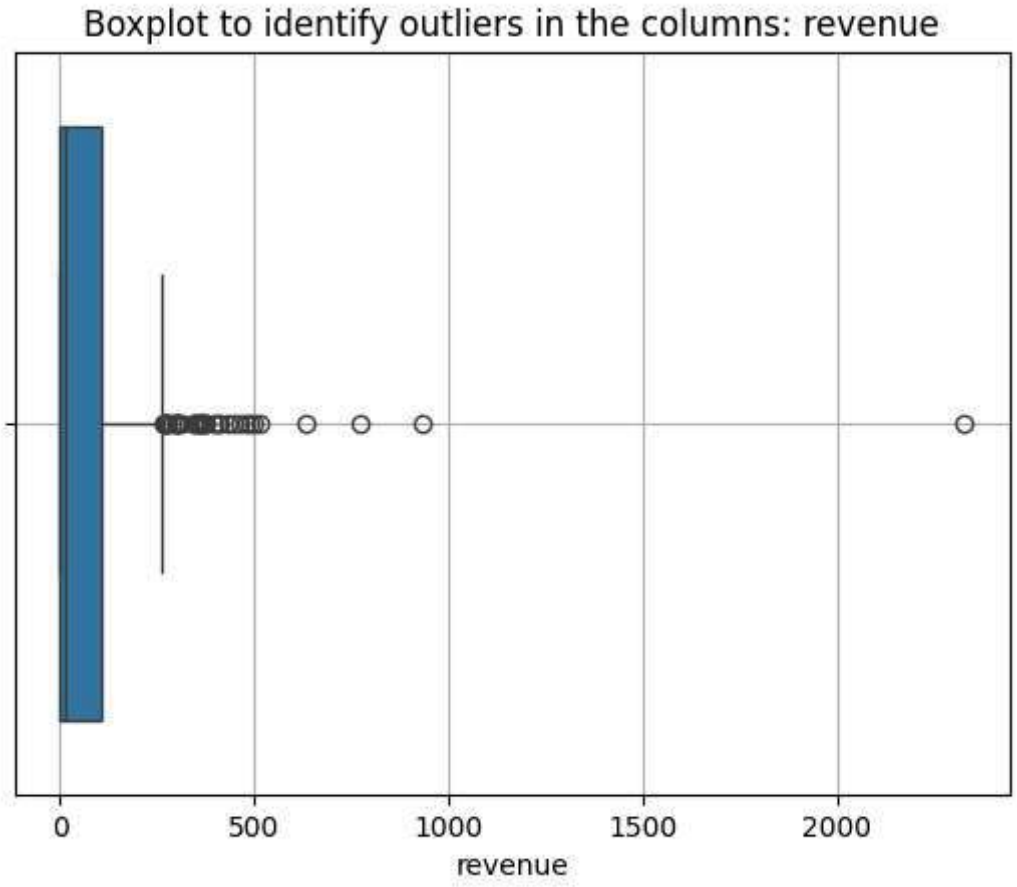

Boxplot to identify outliers in the columns: date
2017-12
2018-02
2018-03
2018-05
2018-07
2018-09
2018-12
2019-02
2019-03
2019-05
2019-07
date


Boxplot to identify outliers in the columns: purchases
0
50
100
150
200
purchases

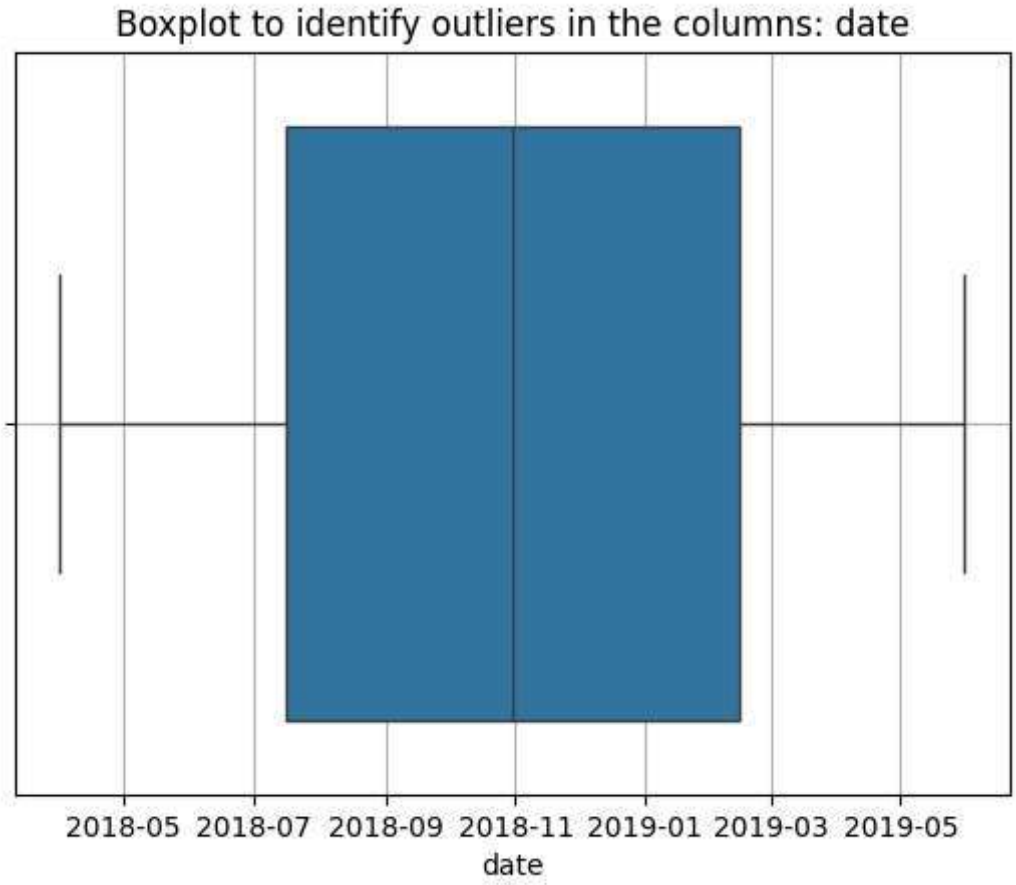

Boxplot to identify outliers in the columns: unique_invoices
0
2
4
6
8
10
12
unique_invoices

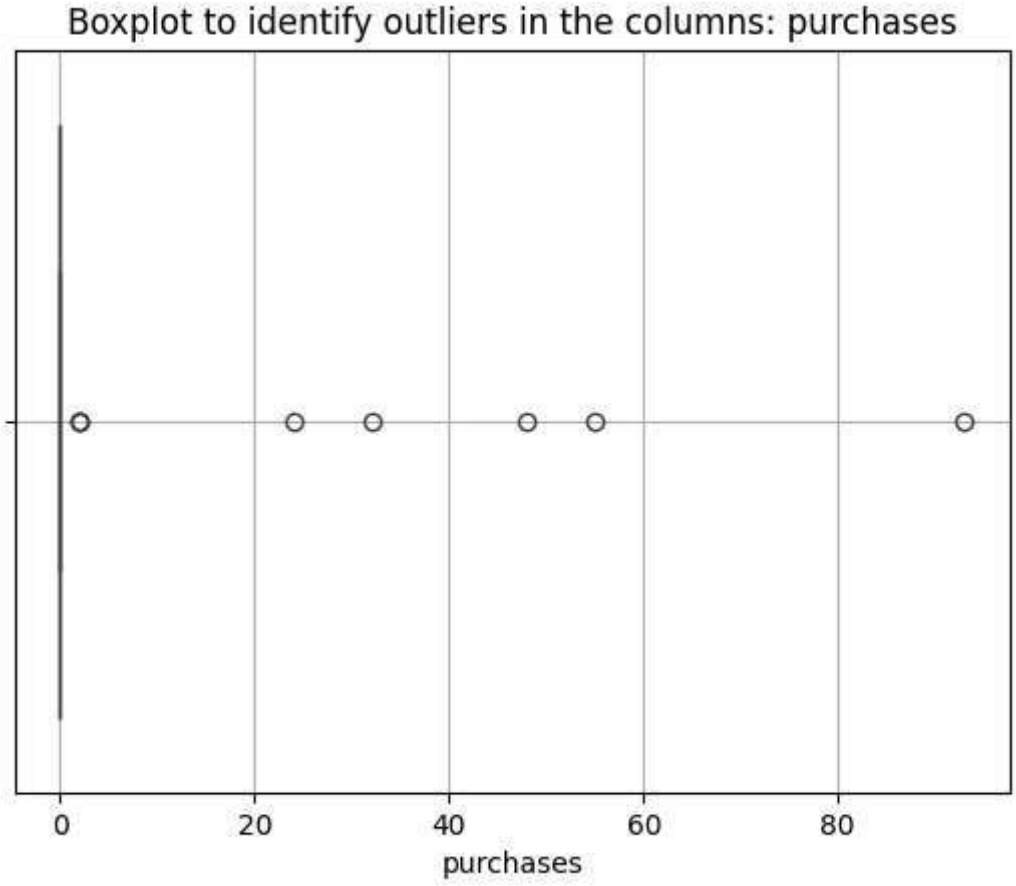

Boxplot to identify outliers in the columns: unique_streams
0
50
100
150
200
unique_streams

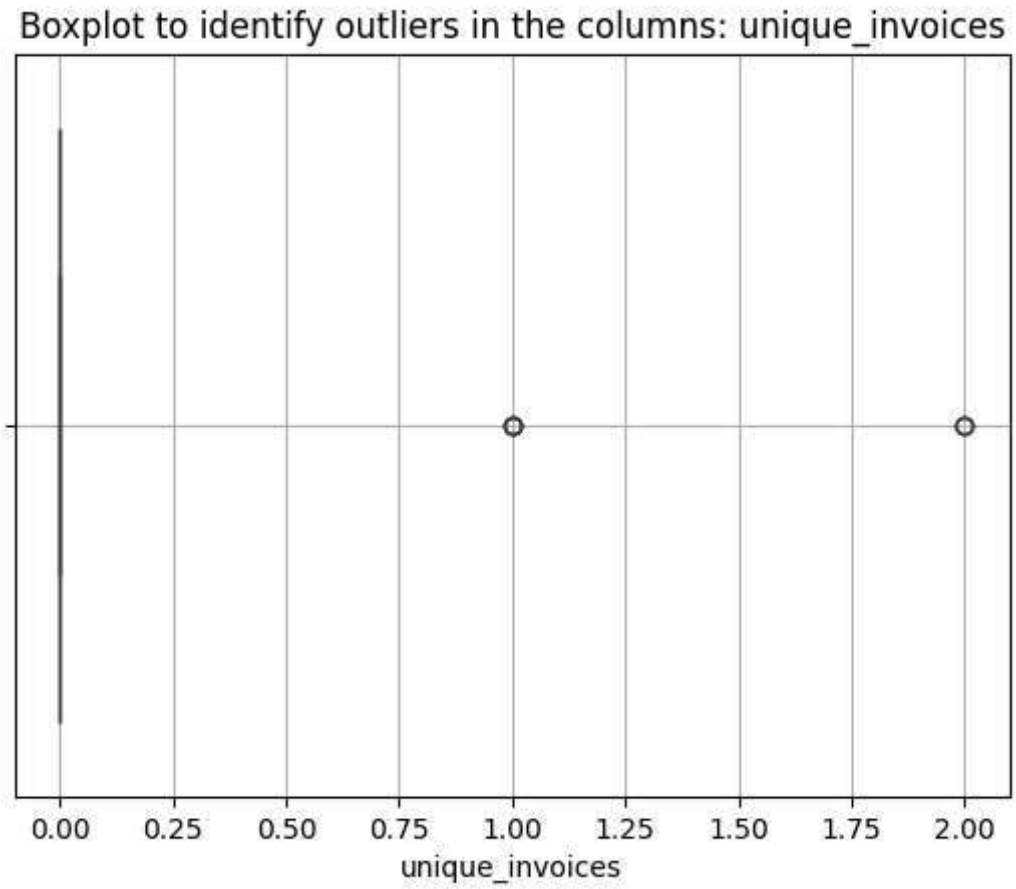

Boxplot to identify outliers in the columns: total_views
0
250
500
750
1000
1250
1500
1750
2000
total_views

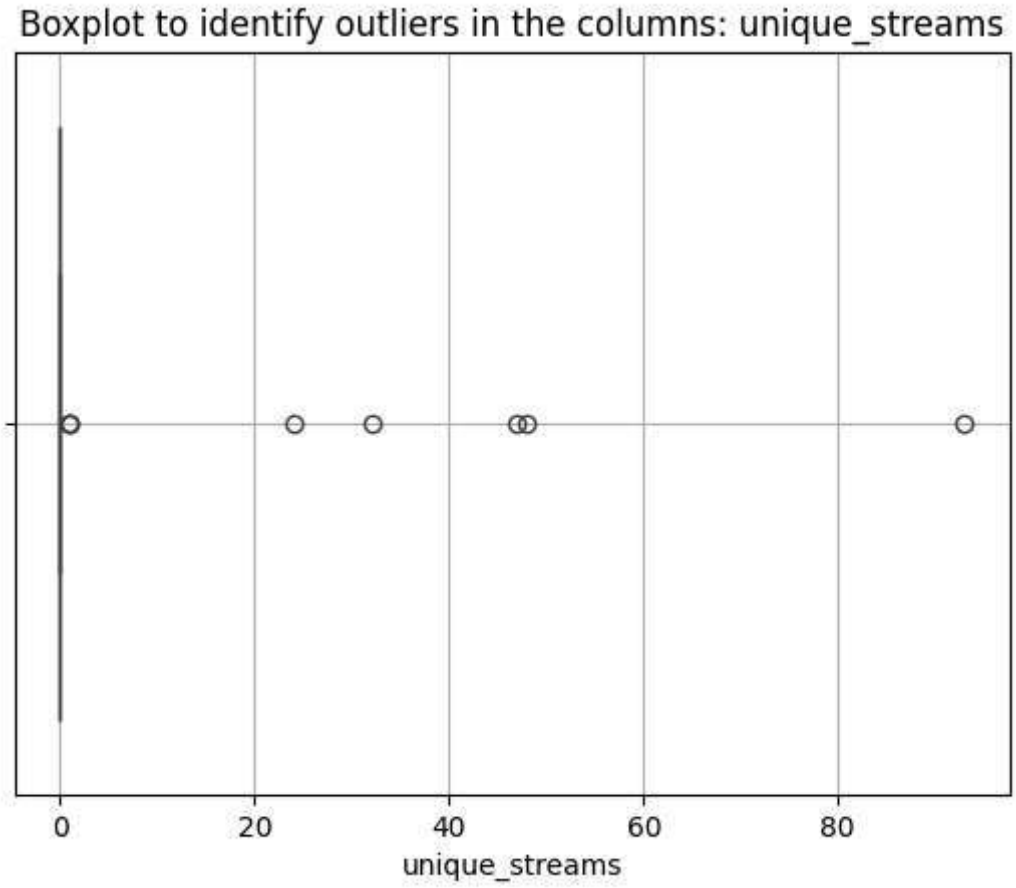

Boxplot to identify outliers in the columns: revenue
0
500
1000
1500
2000
revenue

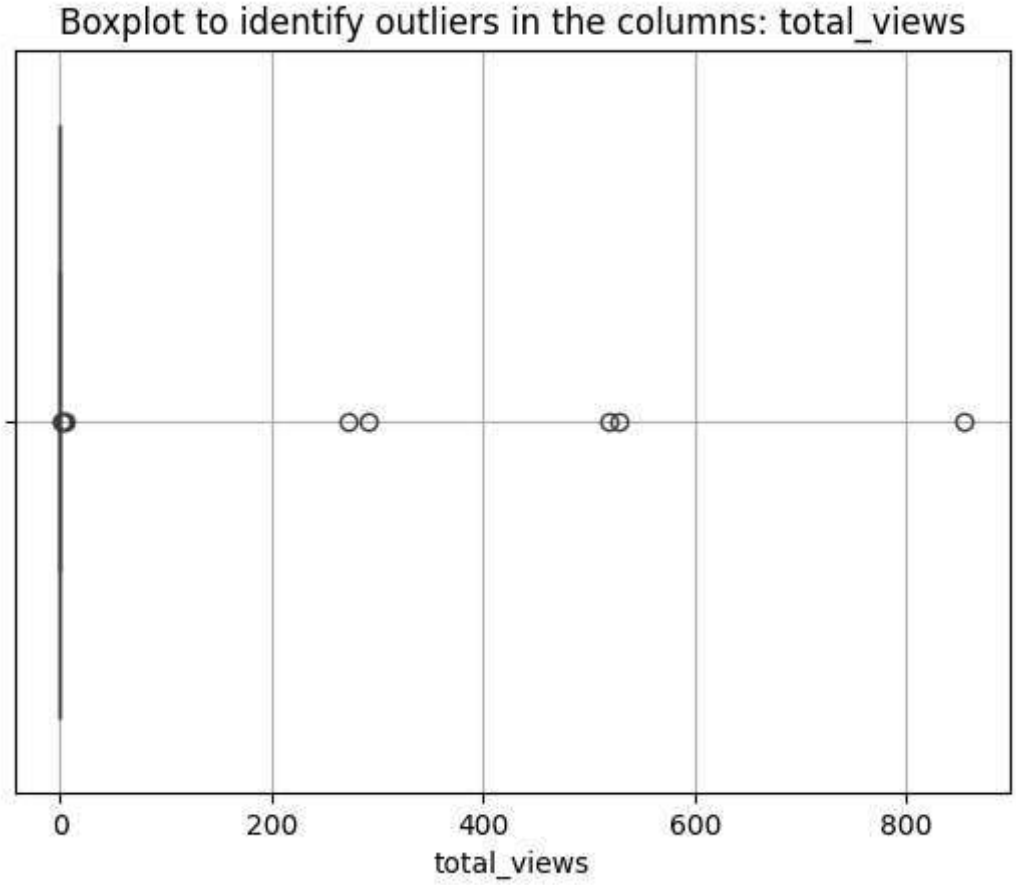
5 - Dataframe df_ts_hong_kong data:

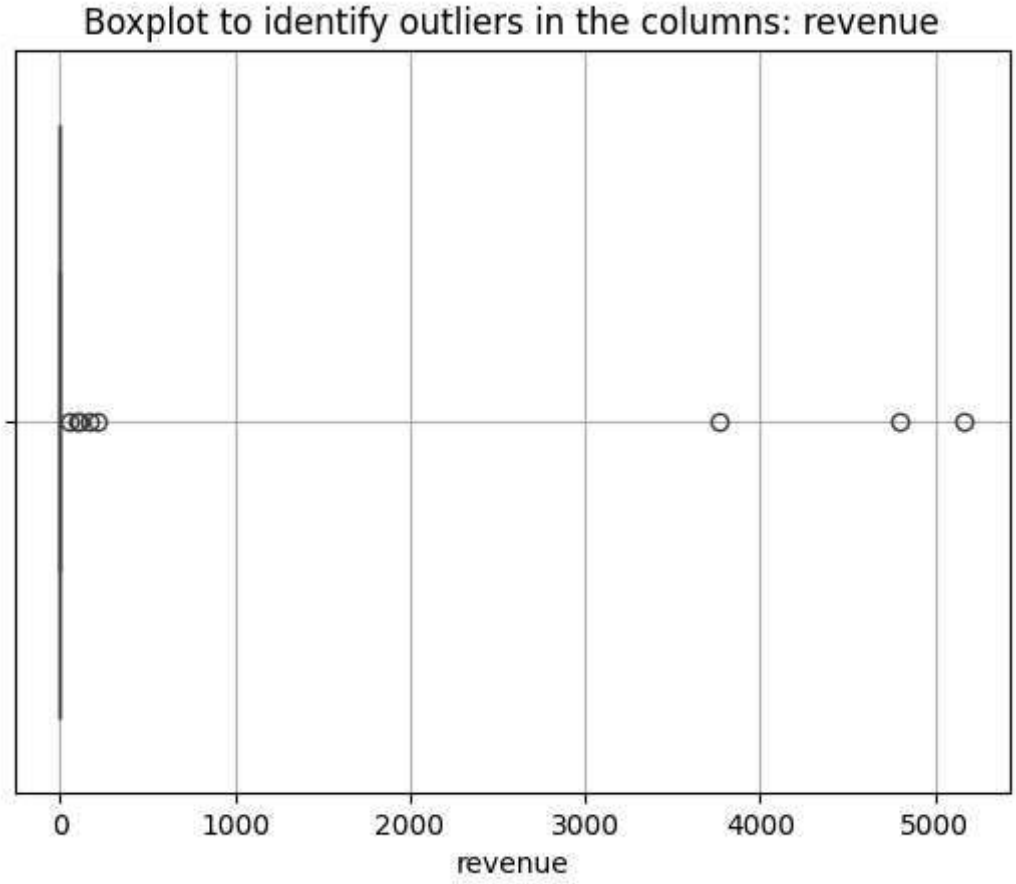



Boxplot to identify outliers in the columns: purchases
0
20
40
60
80
purchases

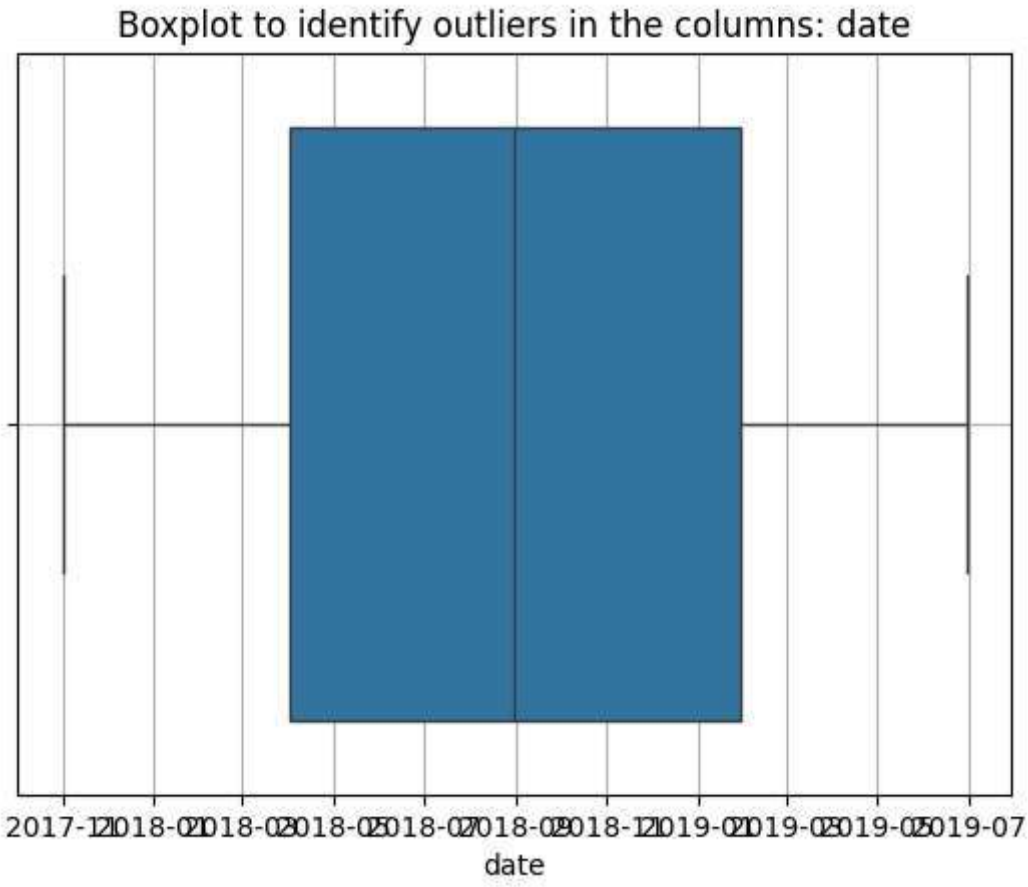

Boxplot to identify outliers in the columns: unique_invoices
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
unique_invoices

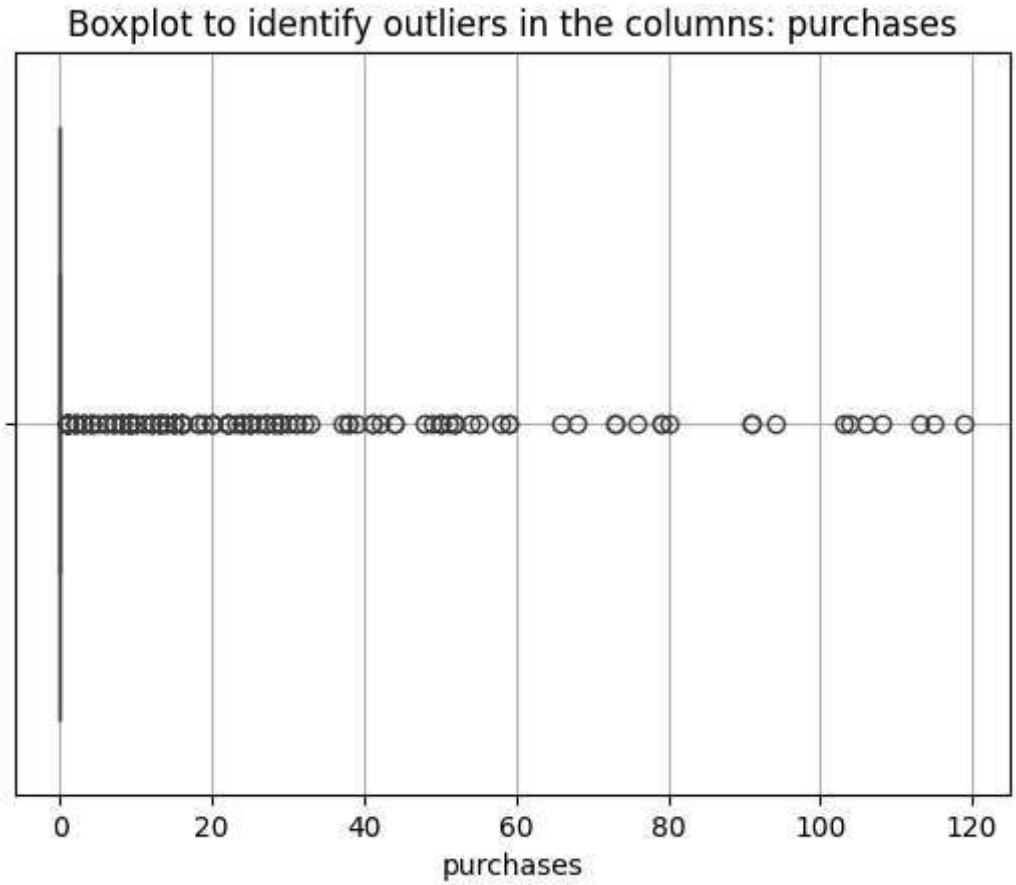

Boxplot to identify outliers in the columns: unique_streams
0
20
40
60
80
unique_streams

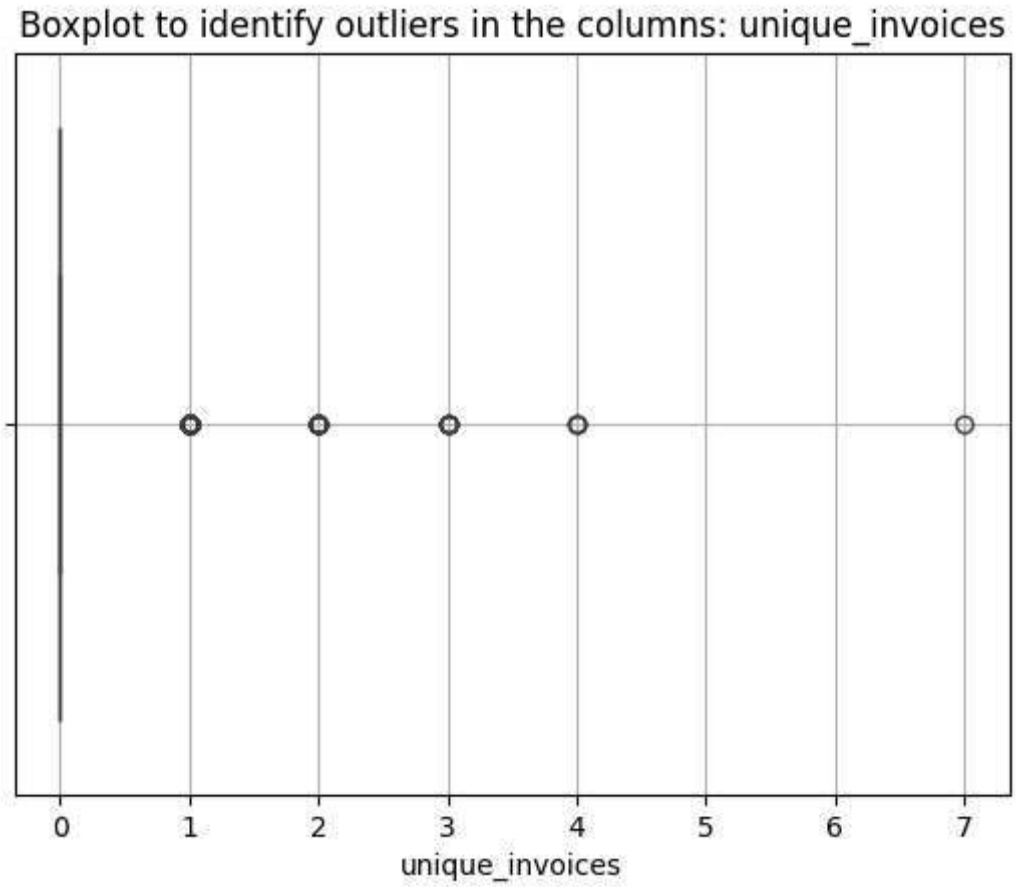

Boxplot to identify outliers in the columns: total_views
0
200
400
600
800
total_views

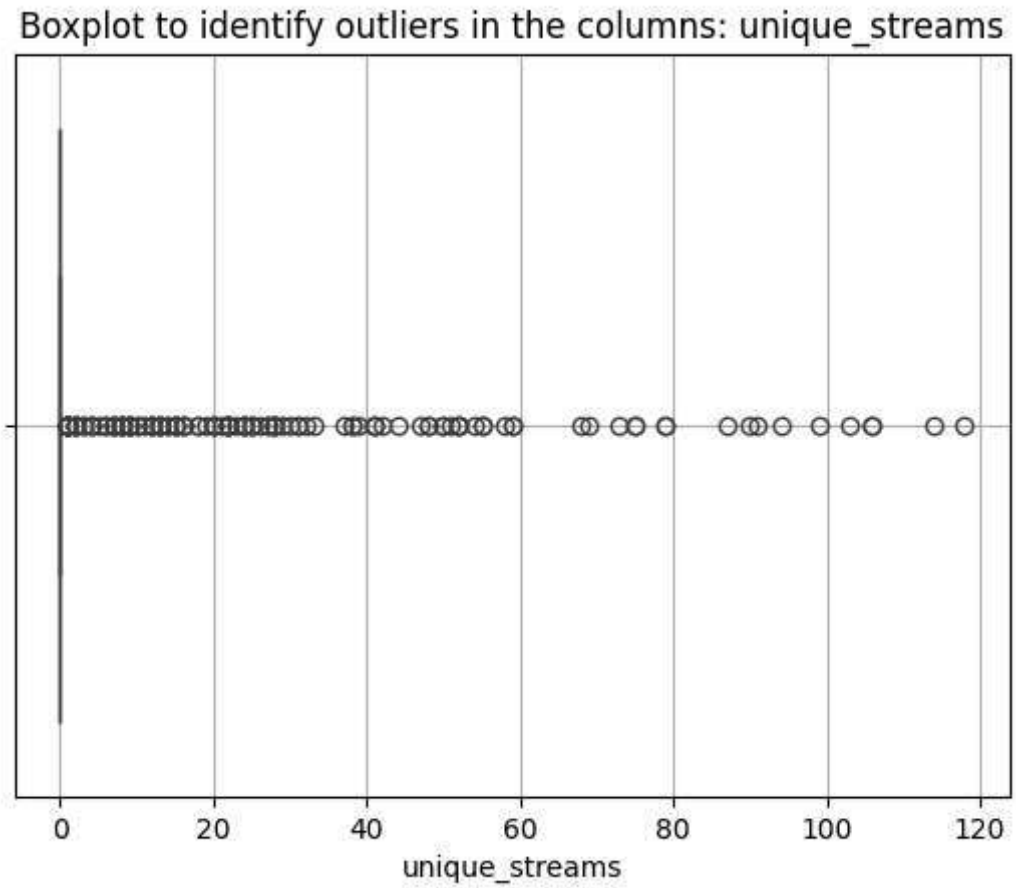


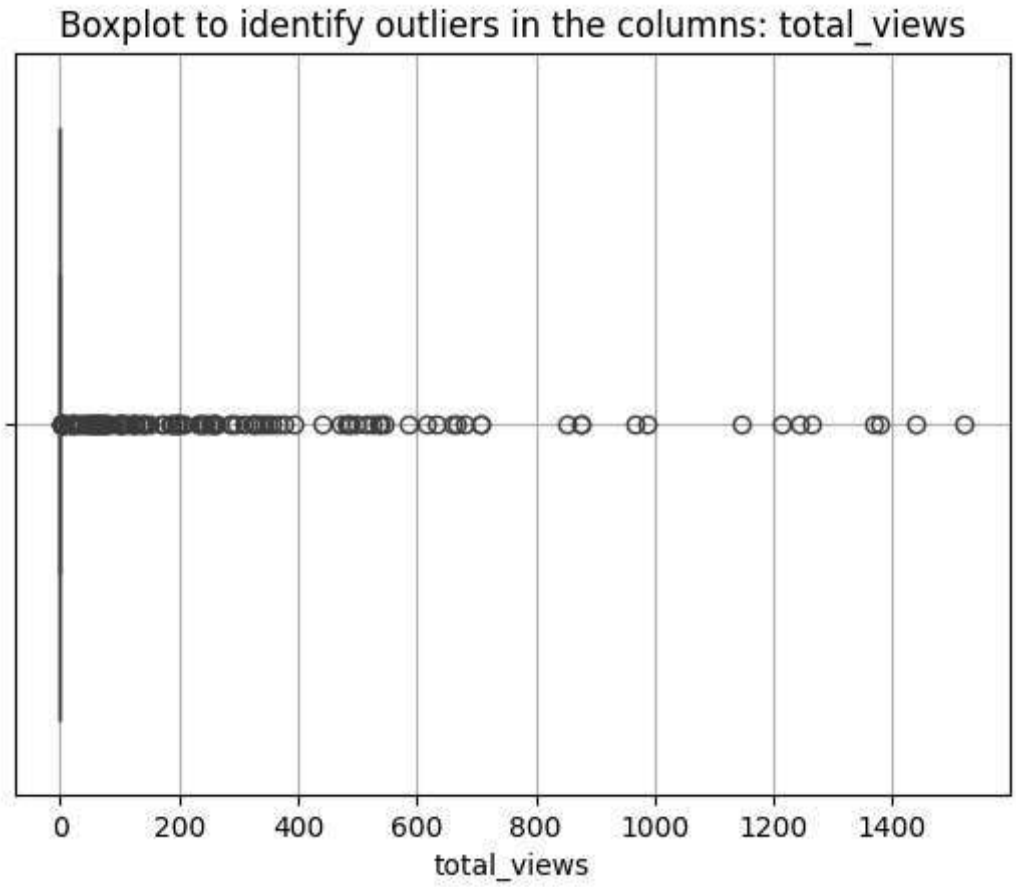
6 - Dataframe df_ts_netherlands data:

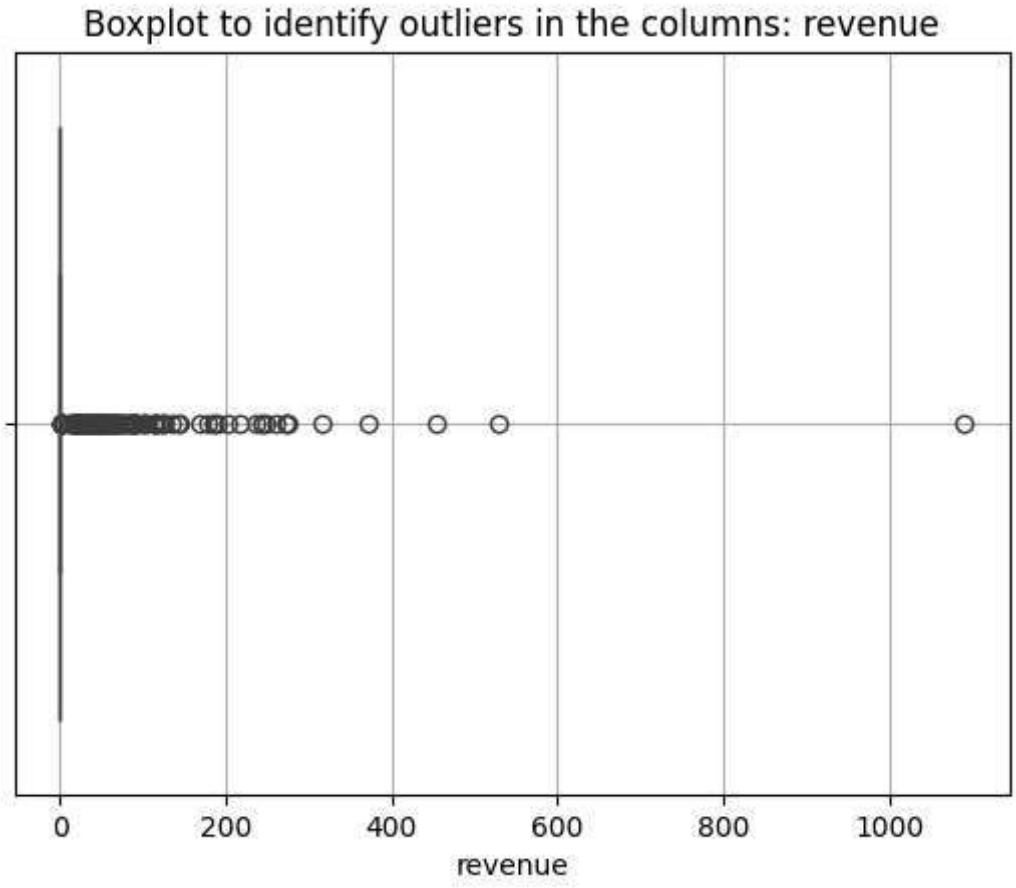



Boxplot to identify outliers in the columns: purchases
0
20
40
60
80
100
120
purchases

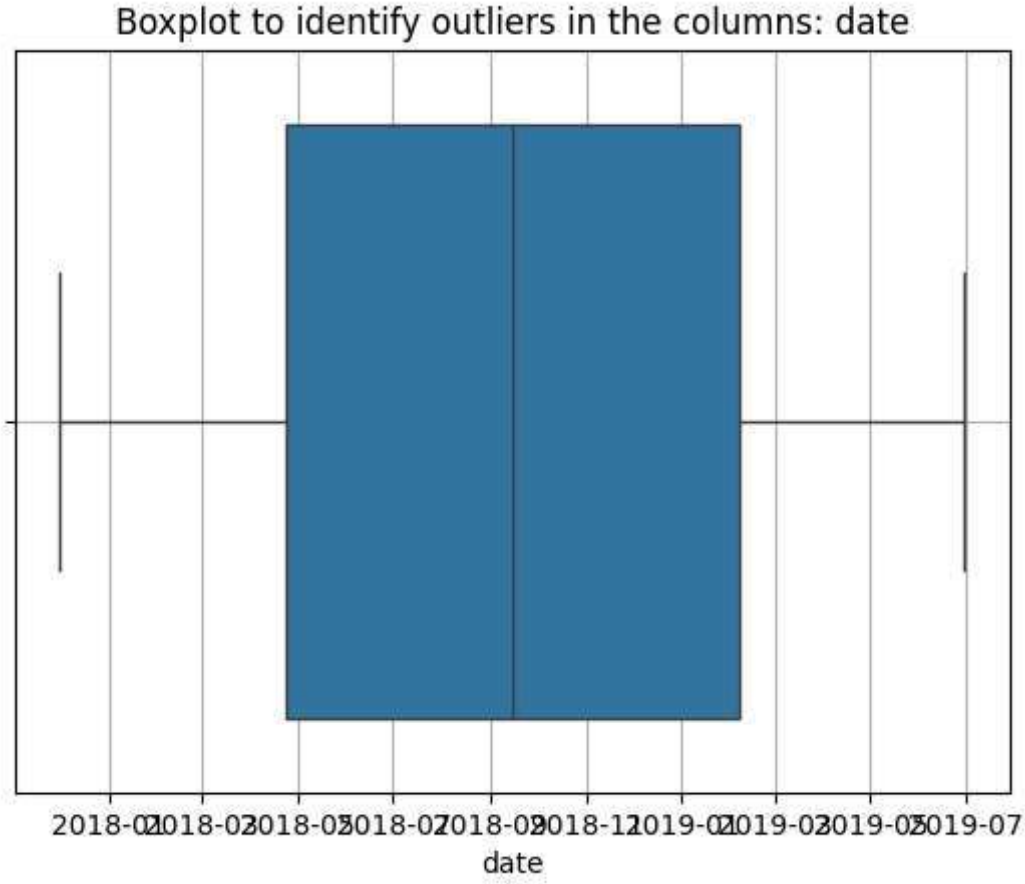

Boxplot to identify outliers in the columns: unique_invoices
0
1
2
3
4
5
6
7
unique_invoices

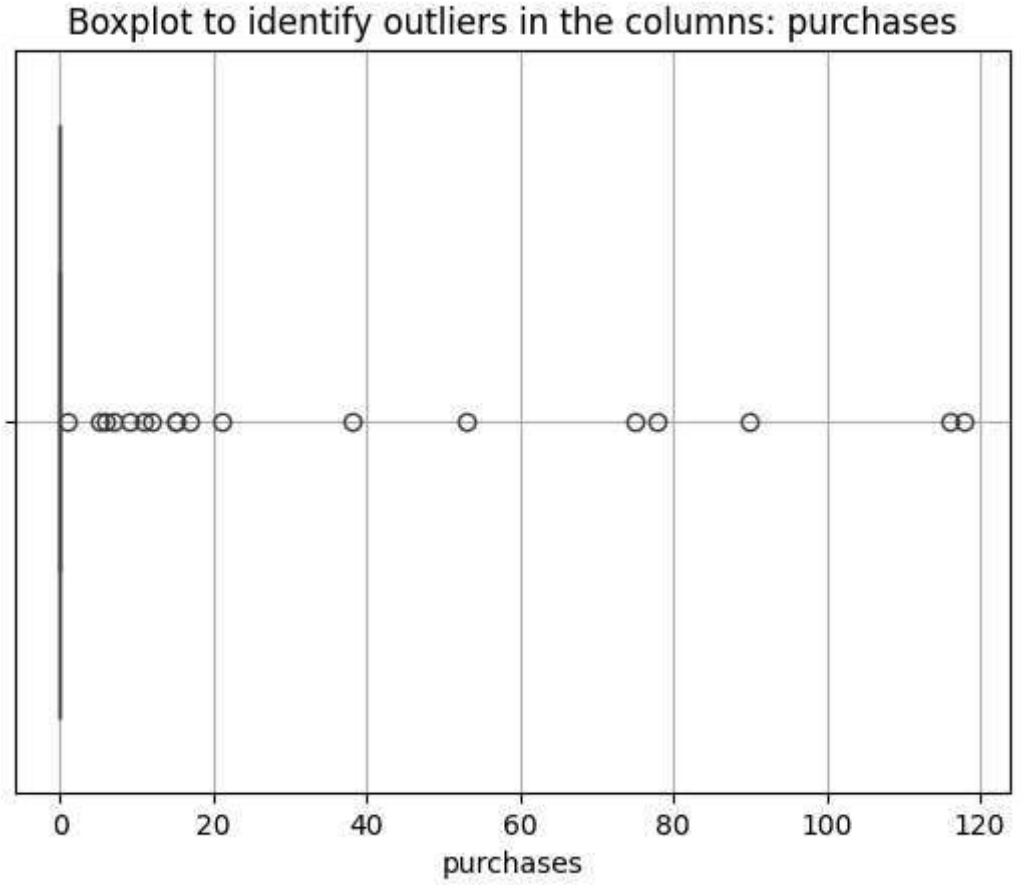

Boxplot to identify outliers in the columns: unique_streams
0
20
40
60
80
100
120
unique_streams

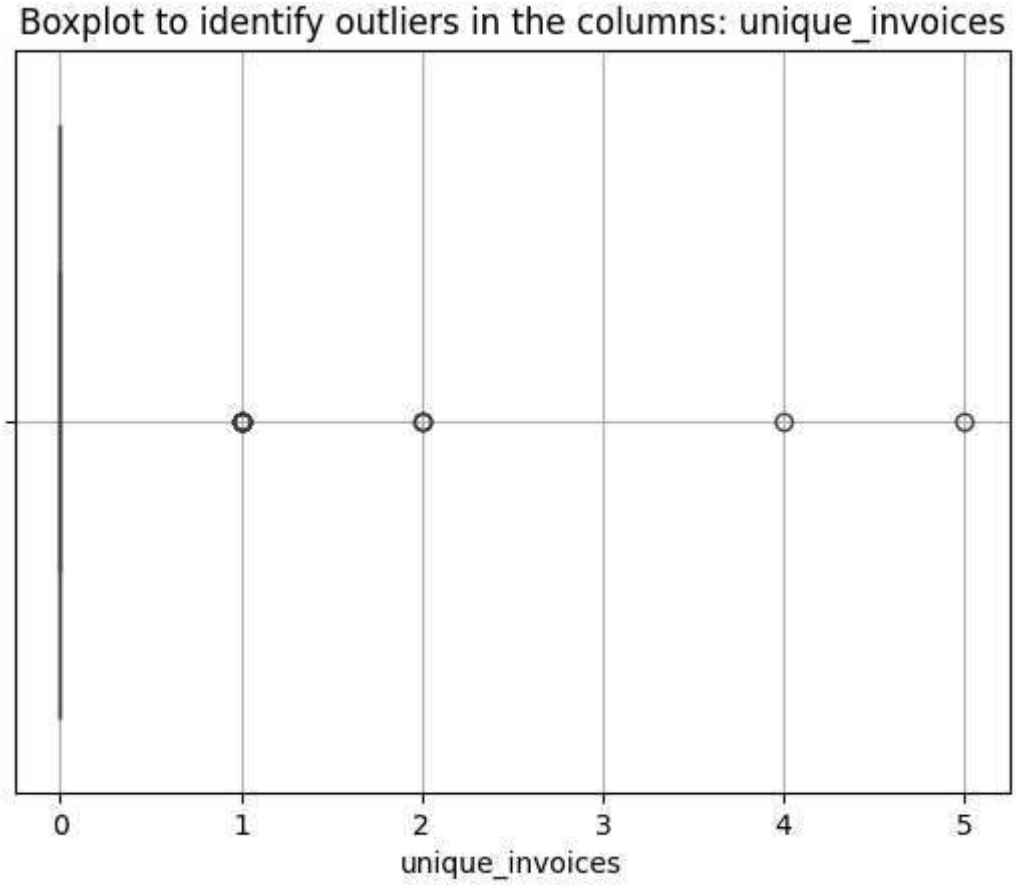

Boxplot to identify outliers in the columns: total_views
0
200
400
600
800
1000
1200
1400
total_views

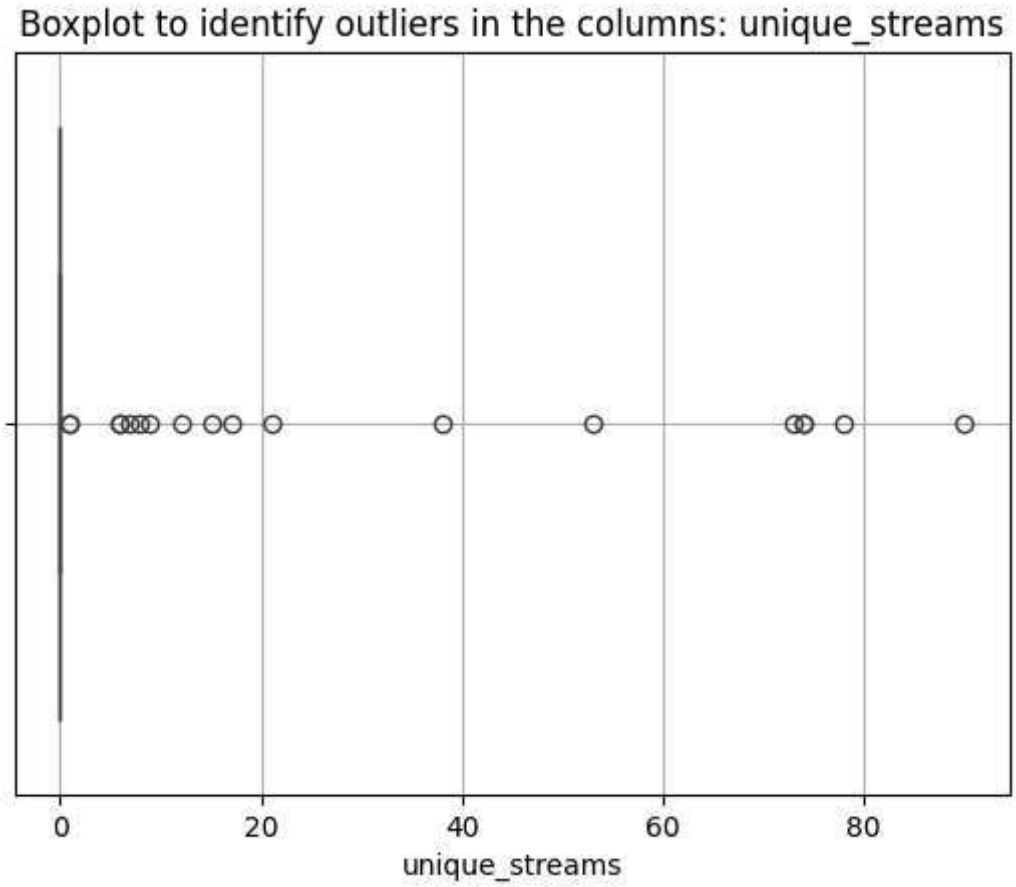


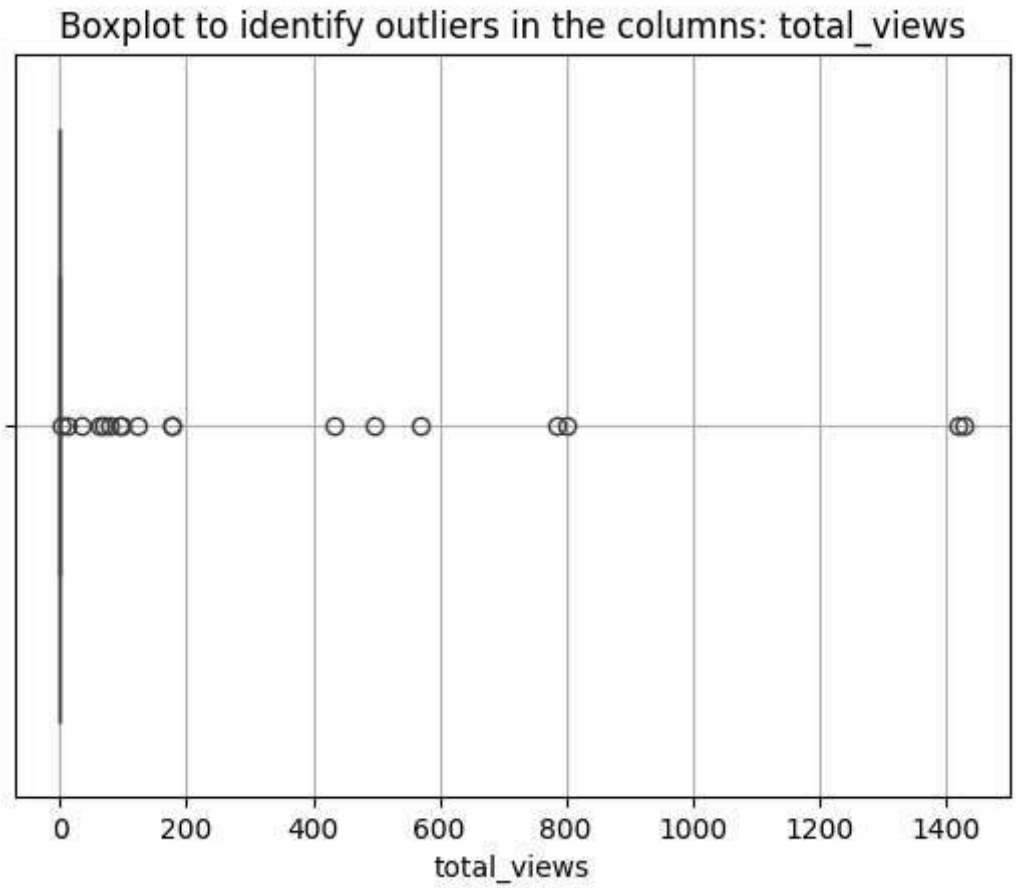
7 - Dataframe df_ts_norway data:

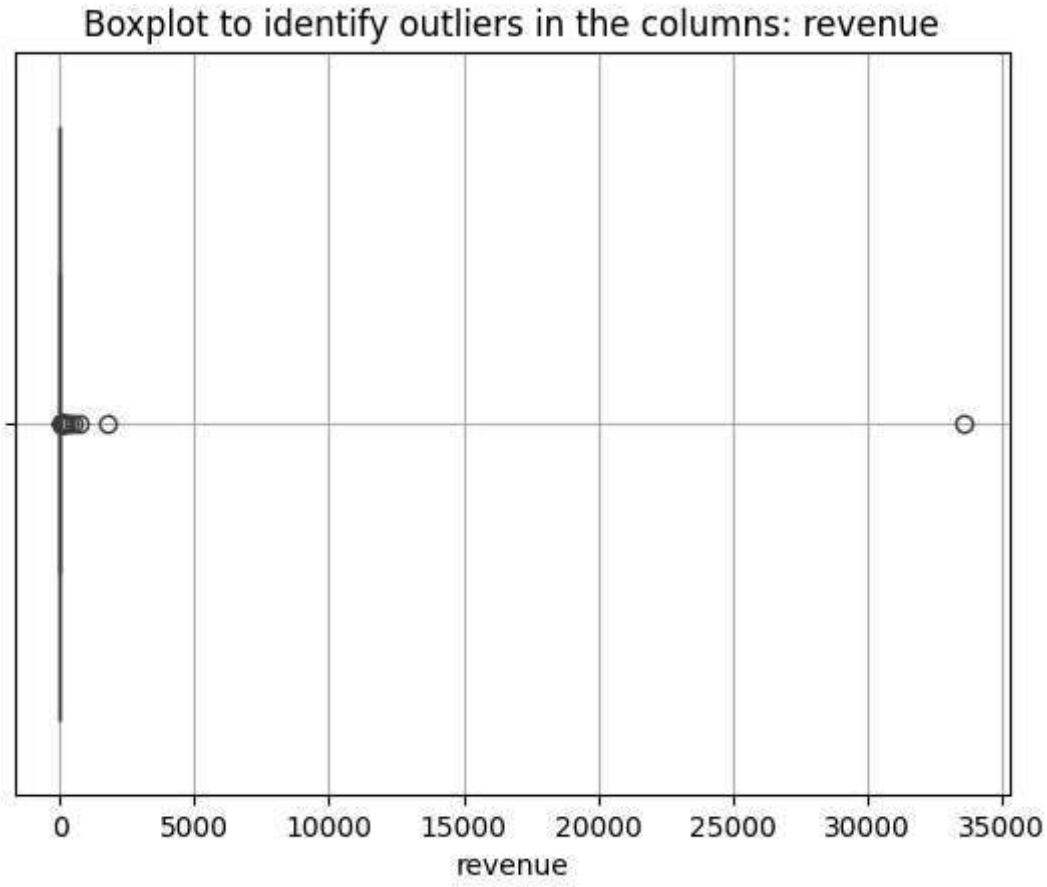

Boxplot to identify outliers in the columns: date
2018-01
2018-03
2018-05
2018-07
2018-09
2018-11
2019-01
2019-03
2019-05
2019-07
date


Boxplot to identify outliers in the columns: purchases
0
20
40
60
80
100
120
purchases

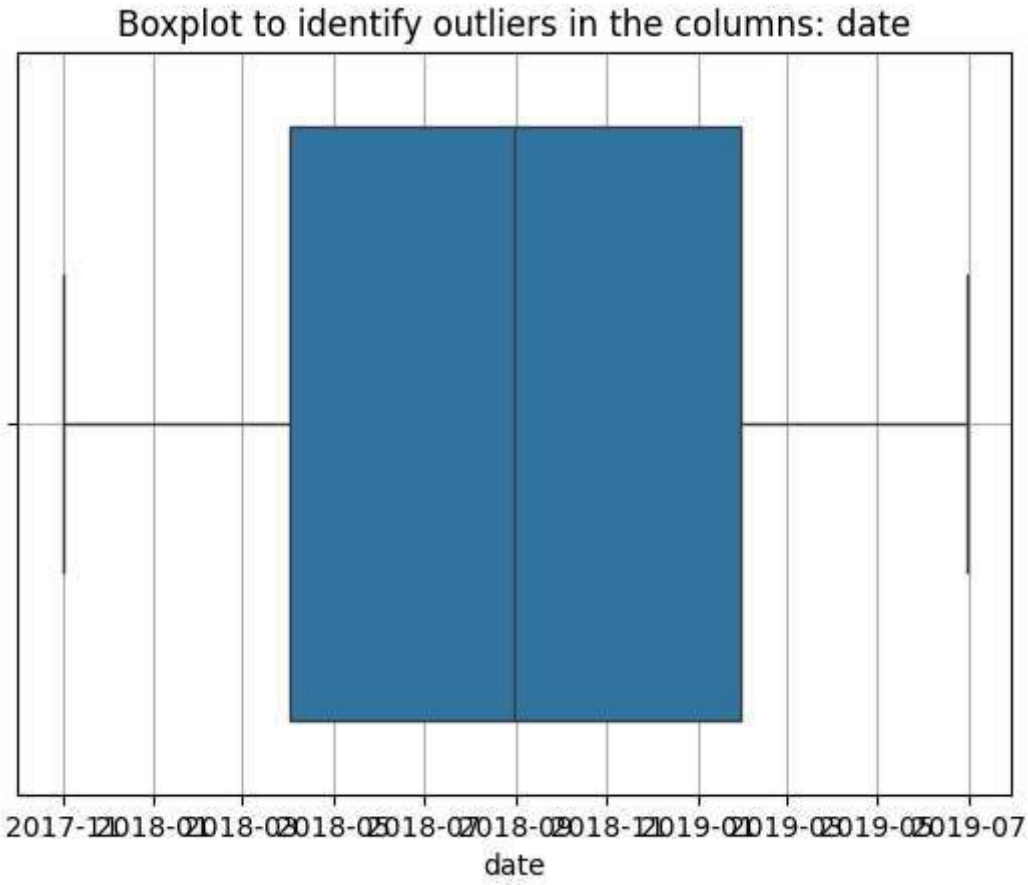

Boxplot to identify outliers in the columns: unique_invoices
0
1
2
3
4
5
unique_invoices

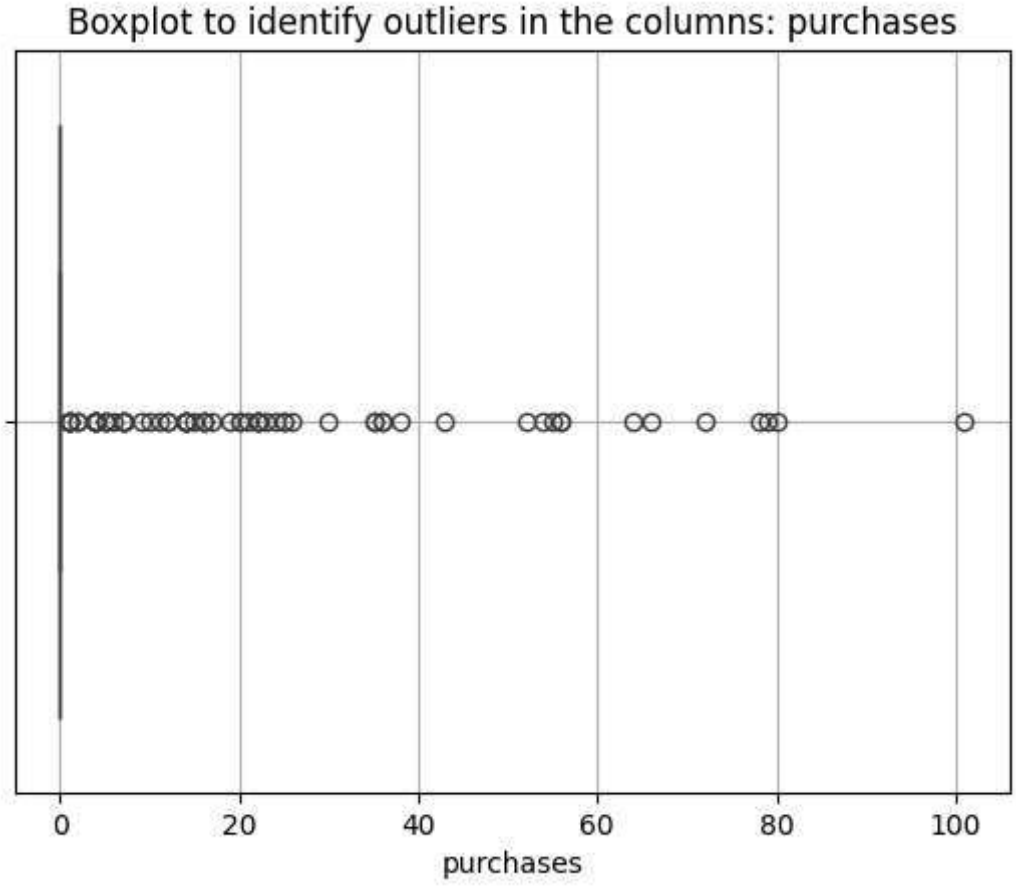

Boxplot to identify outliers in the columns: unique_streams
0
20
40
60
80
unique_streams

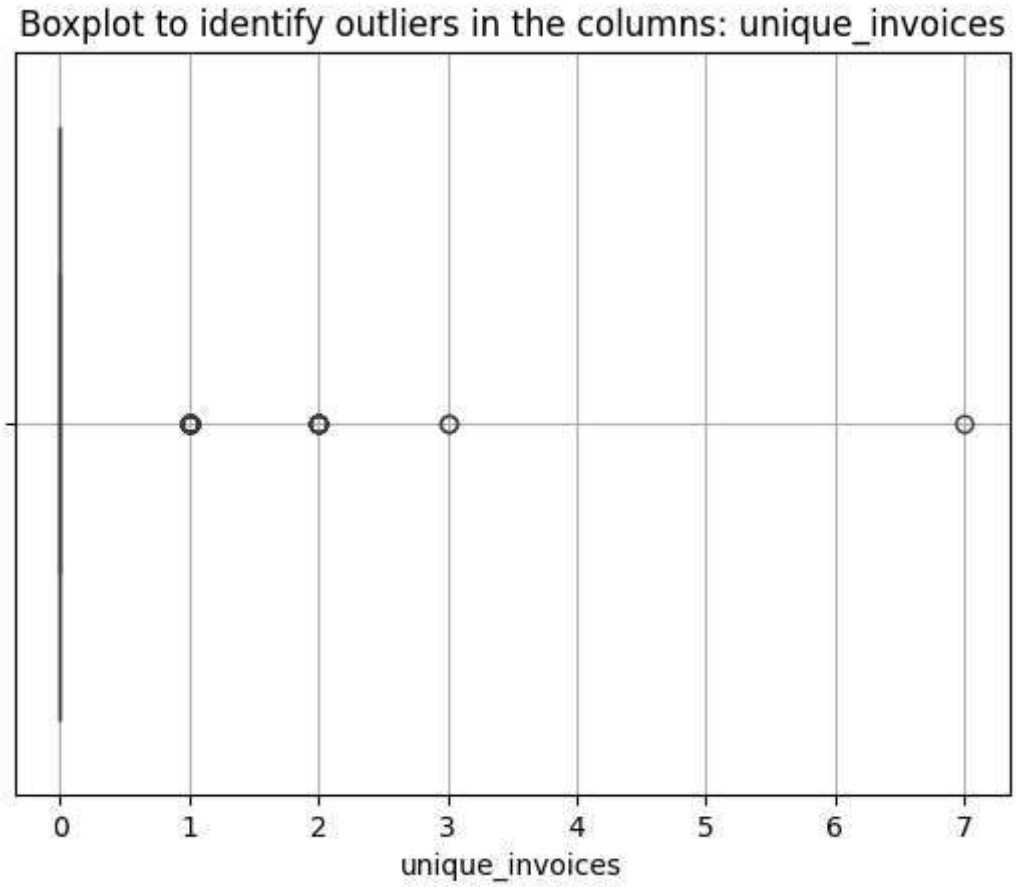

Boxplot to identify outliers in the columns: total_views
0
200
400
600
800
1000
1200
1400
total_views

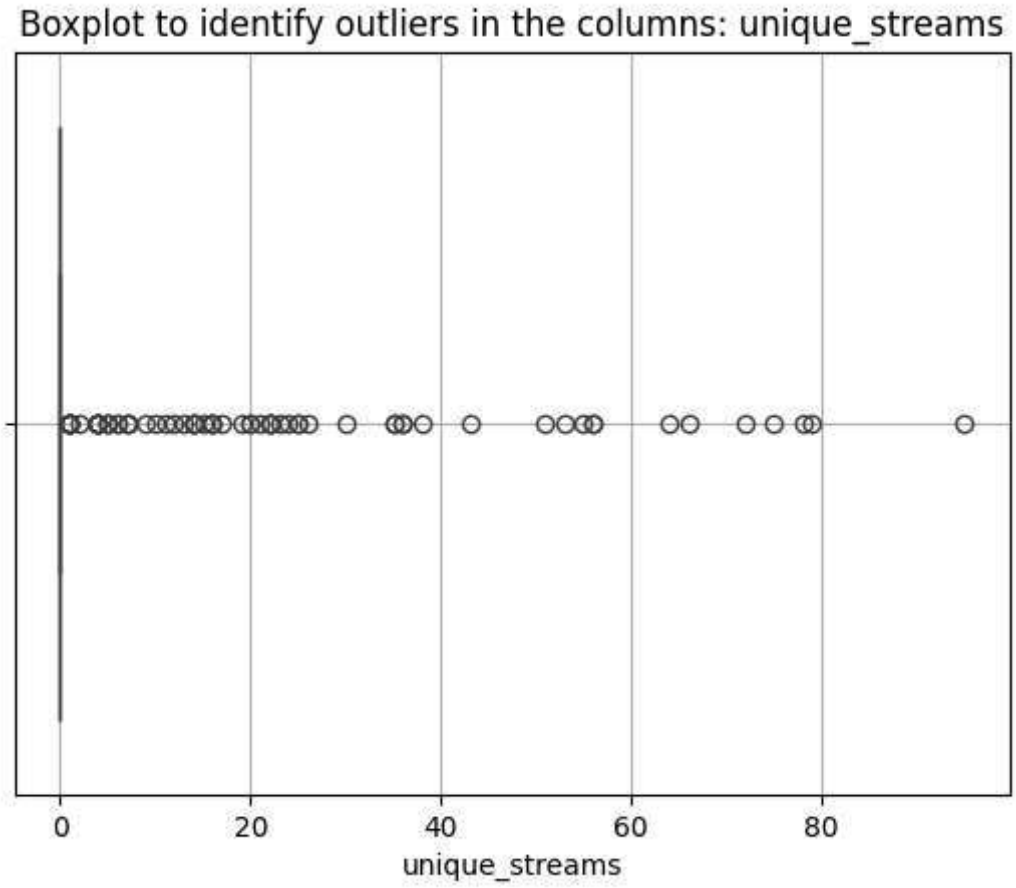

Boxplot to identify outliers in the columns: revenue
0
5000
10000
15000
20000
25000
30000
35000
revenue

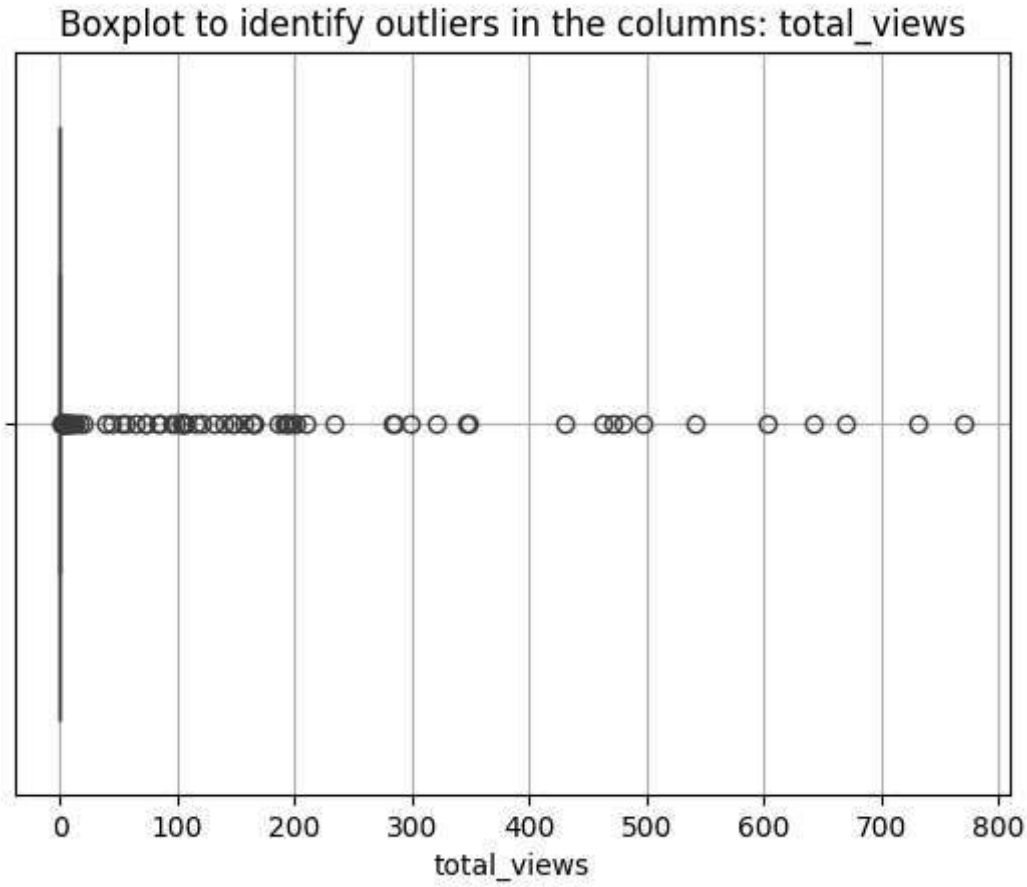
8 - Dataframe df_ts_portugal data:

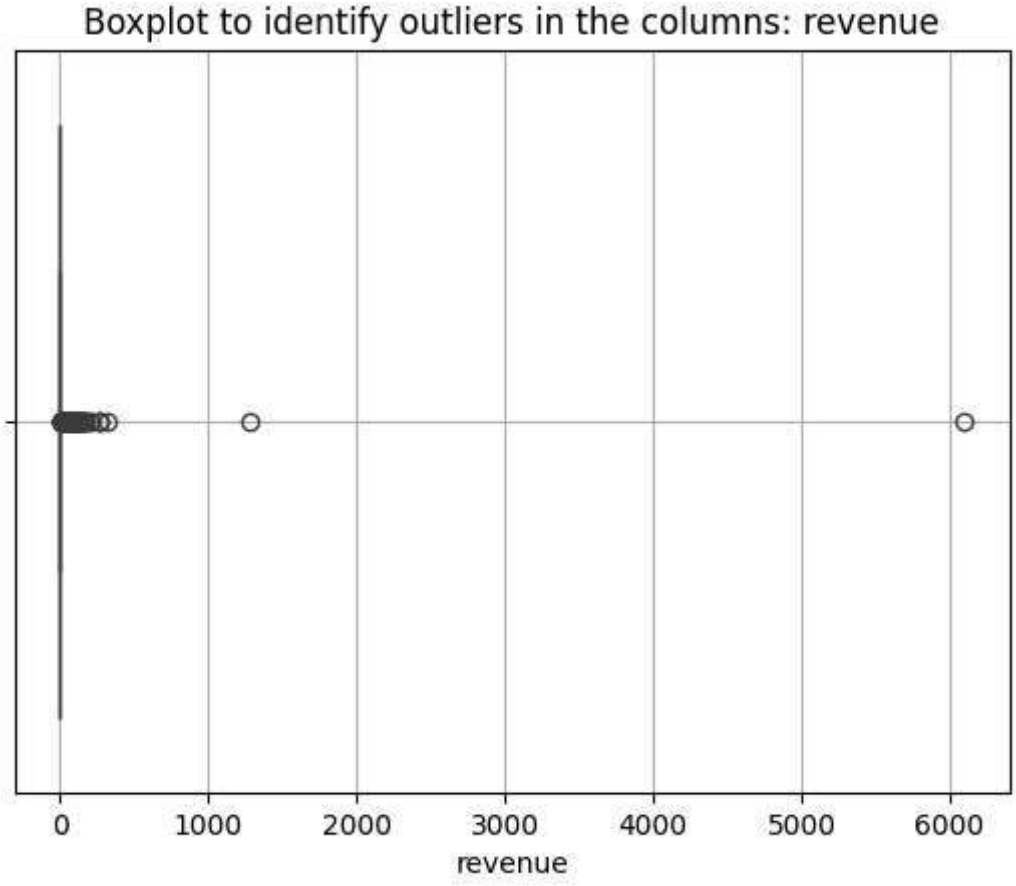



Boxplot to identify outliers in the columns: purchases
0
20
40
60
80
100
purchases

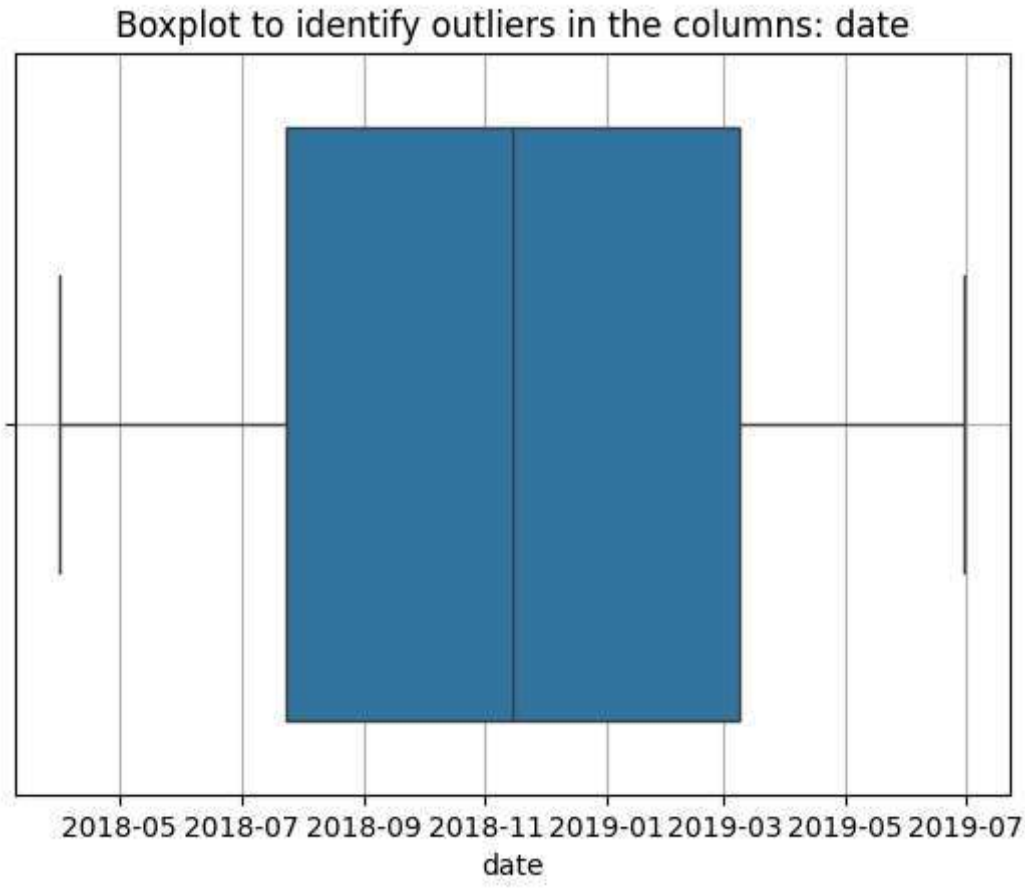

Boxplot to identify outliers in the columns: unique_invoices
0
1
2
3
4
5
6
7
unique_invoices

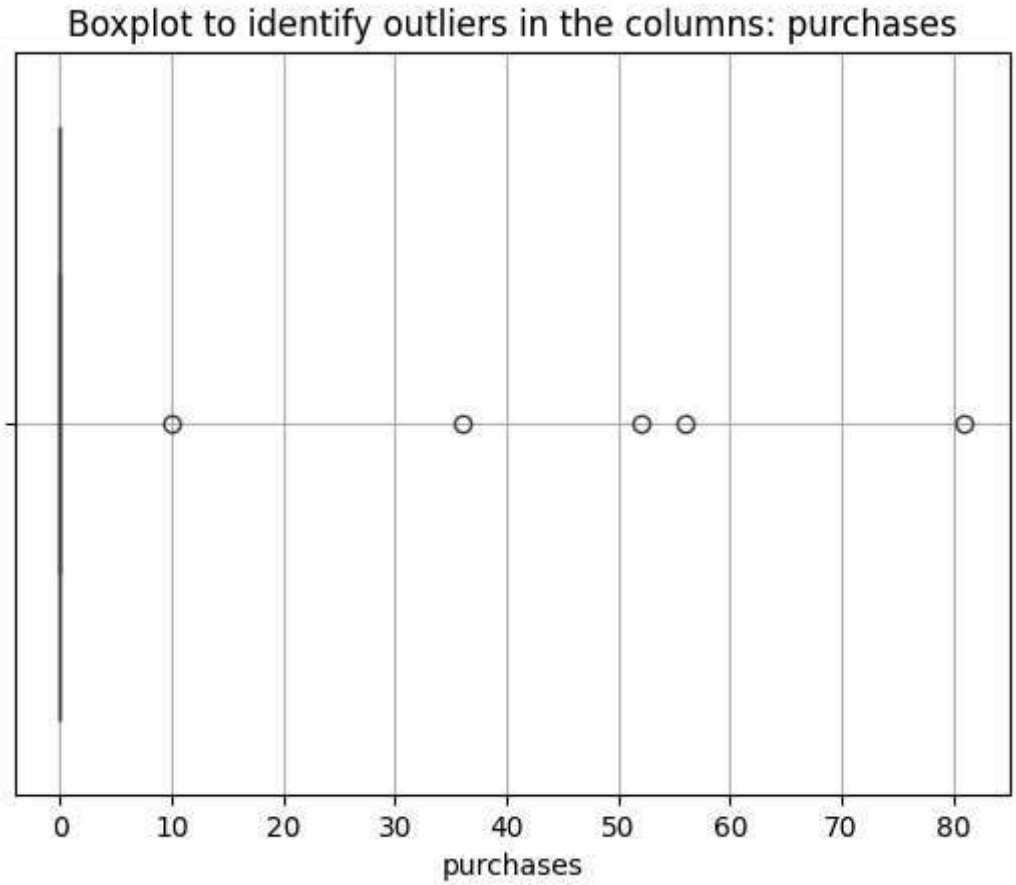

Boxplot to identify outliers in the columns: unique_streams
0
20
40
60
80
unique_streams

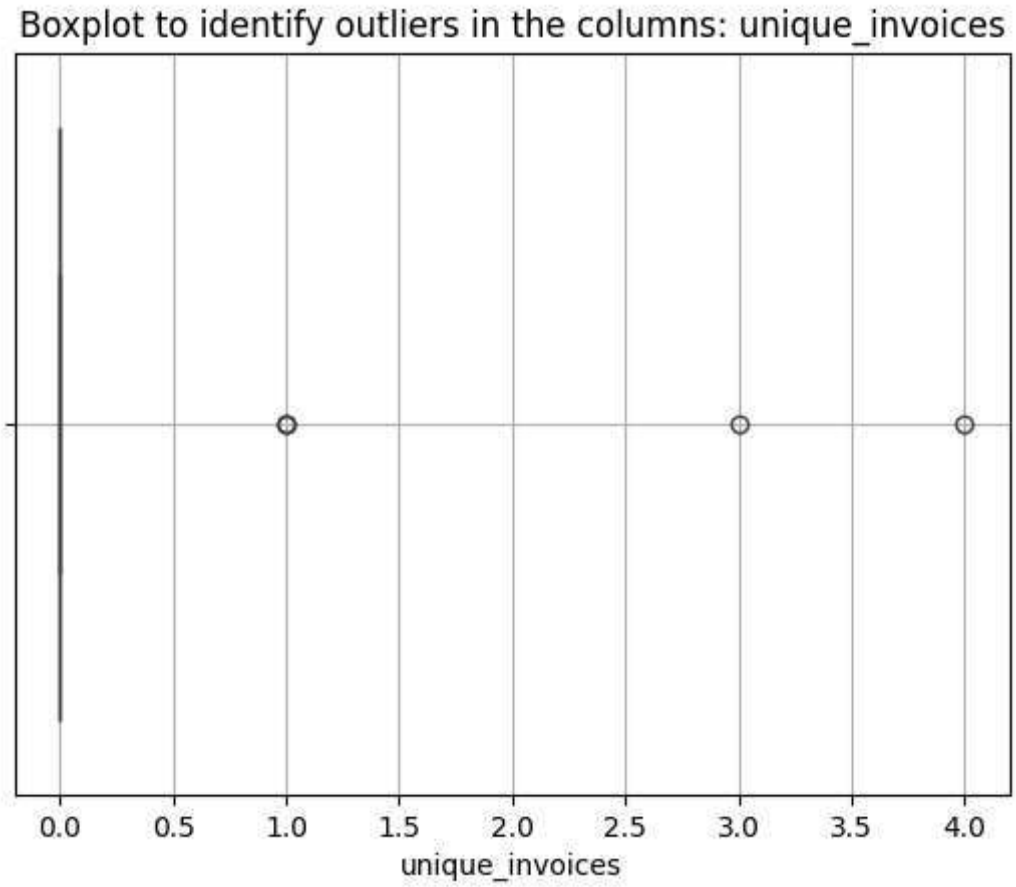

Boxplot to identify outliers in the columns: total_views
0
100
200
300
400
500
600
700
800
total_views

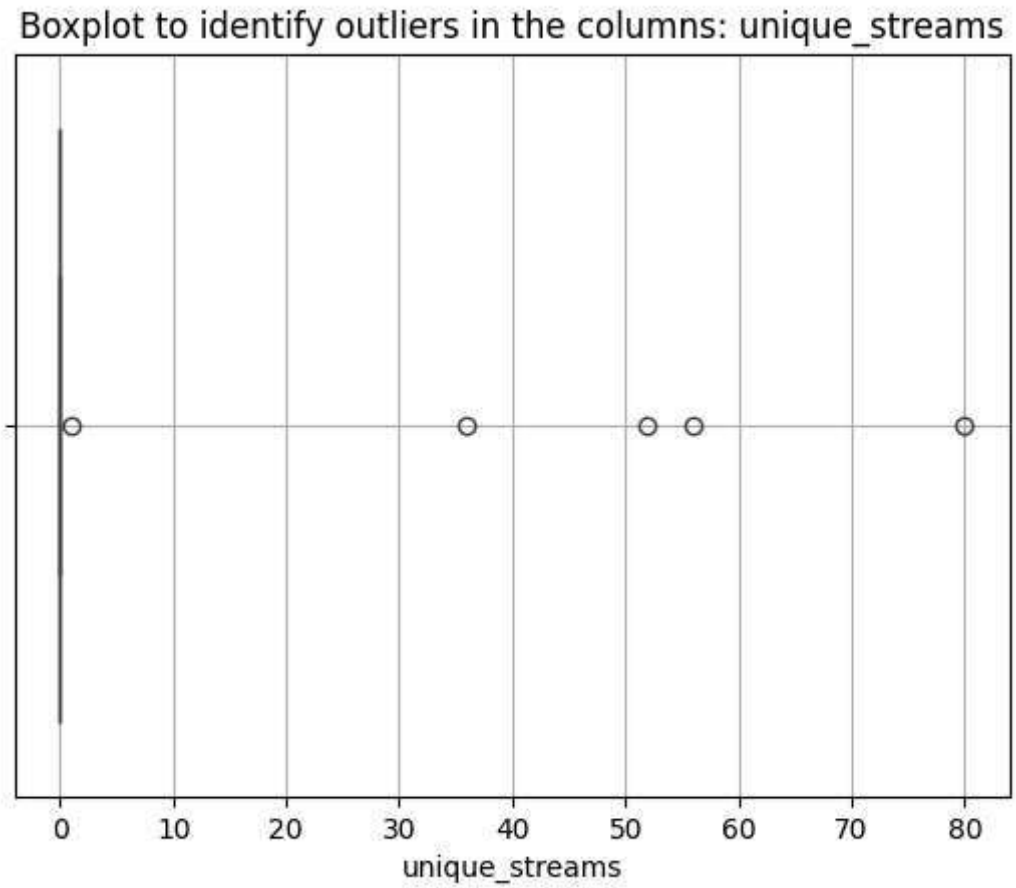


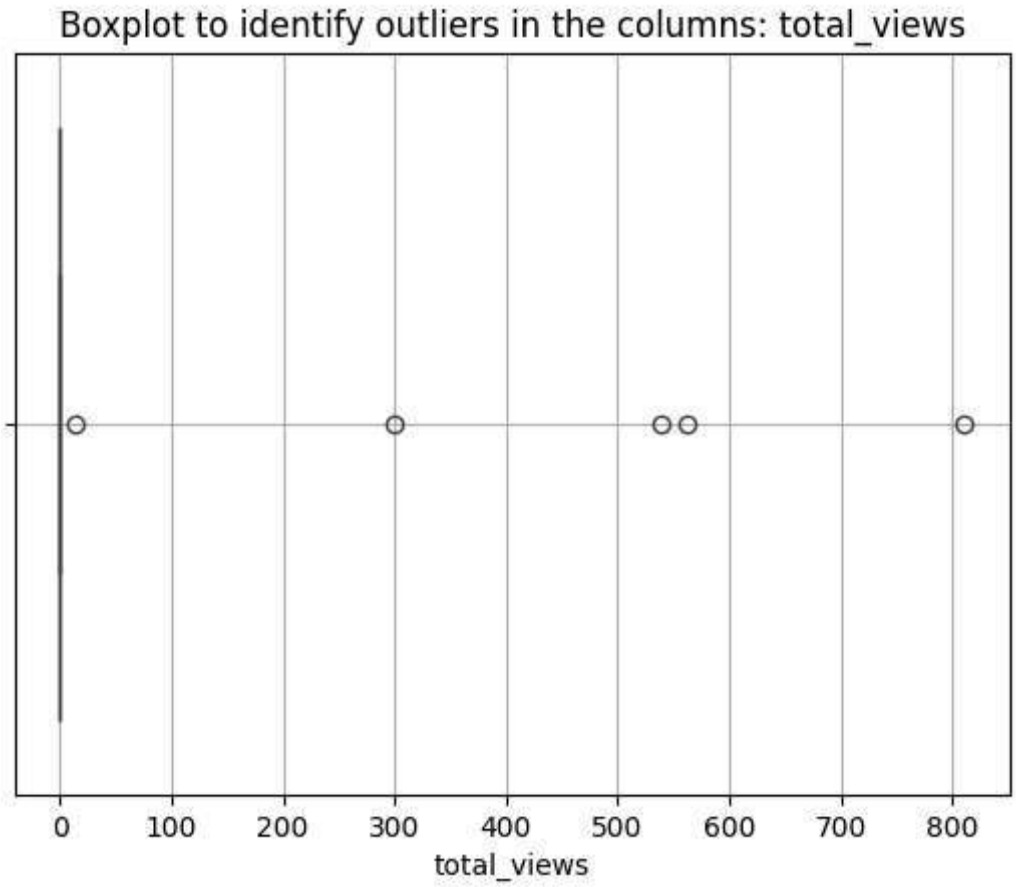
9 - Dataframe df_ts_singapore data:

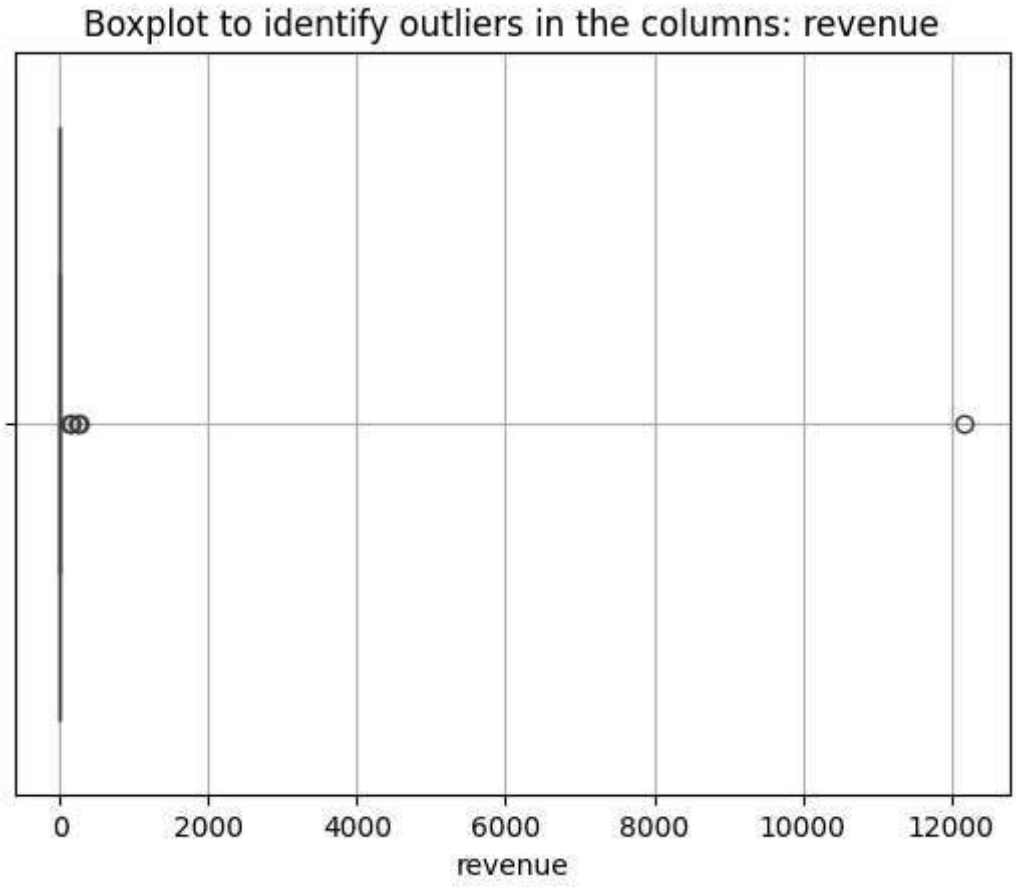



Boxplot to identify outliers in the columns: purchases
0
10
20
30
40
50
60
70
80
purchases

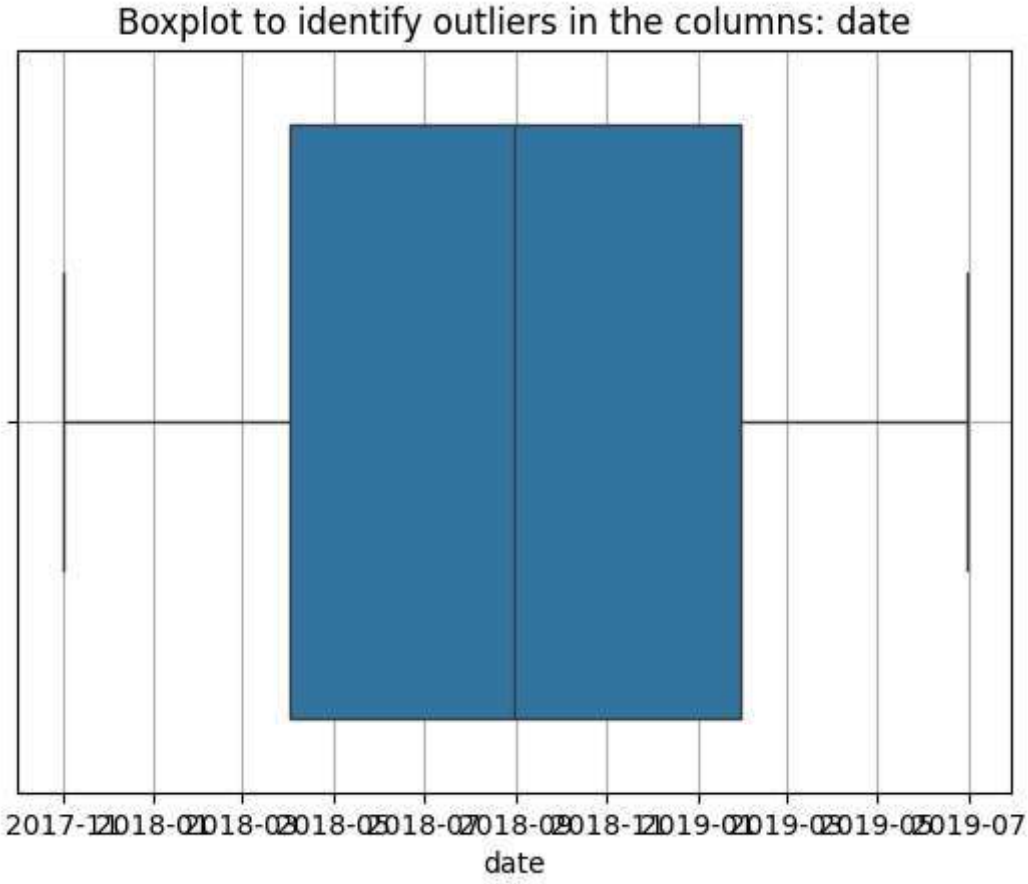

Boxplot to identify outliers in the columns: unique_invoices
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
unique_invoices

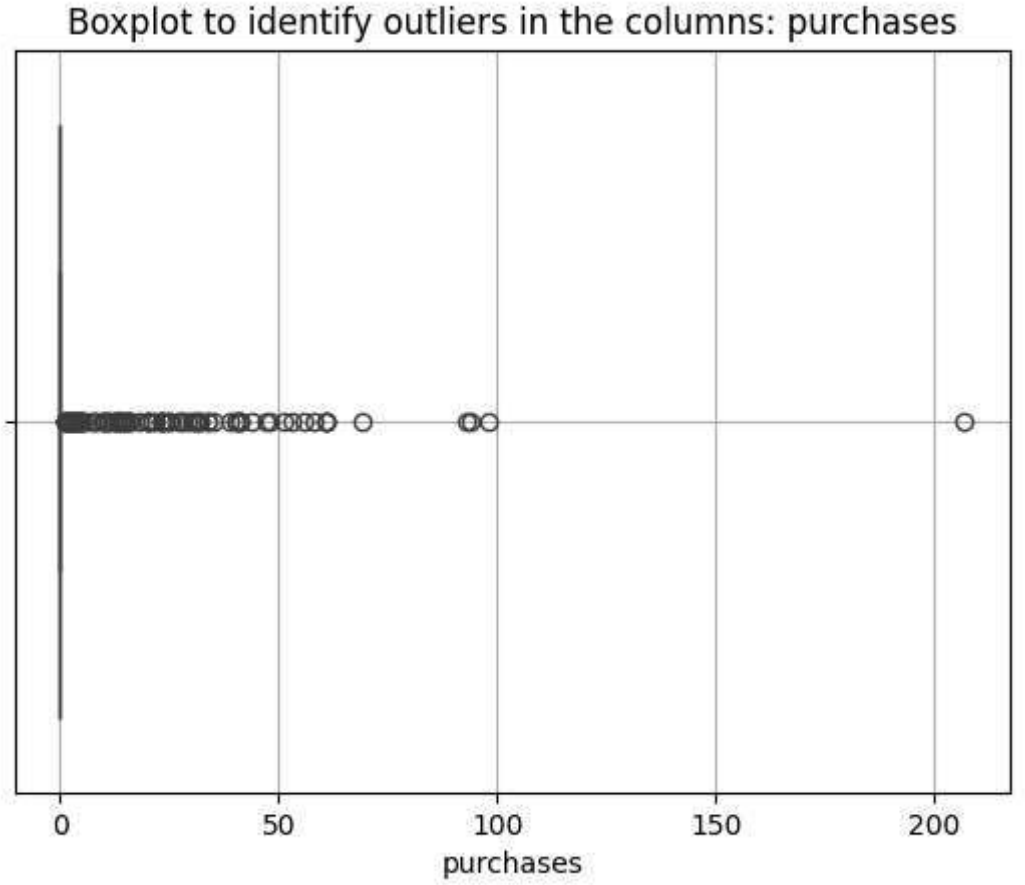

Boxplot to identify outliers in the columns: unique_streams
0
10
20
30
40
50
60
70
80
unique_streams

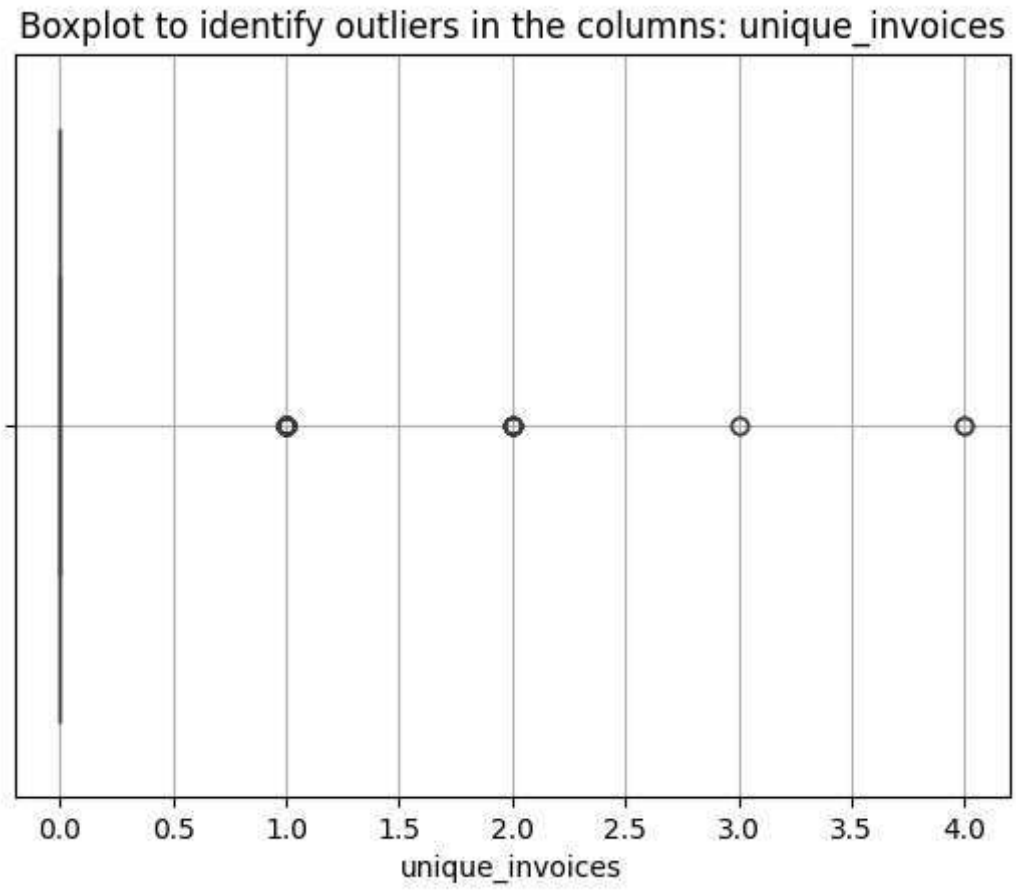

Boxplot to identify outliers in the columns: total_views
0
100
200
300
400
500
600
700
800
total_views

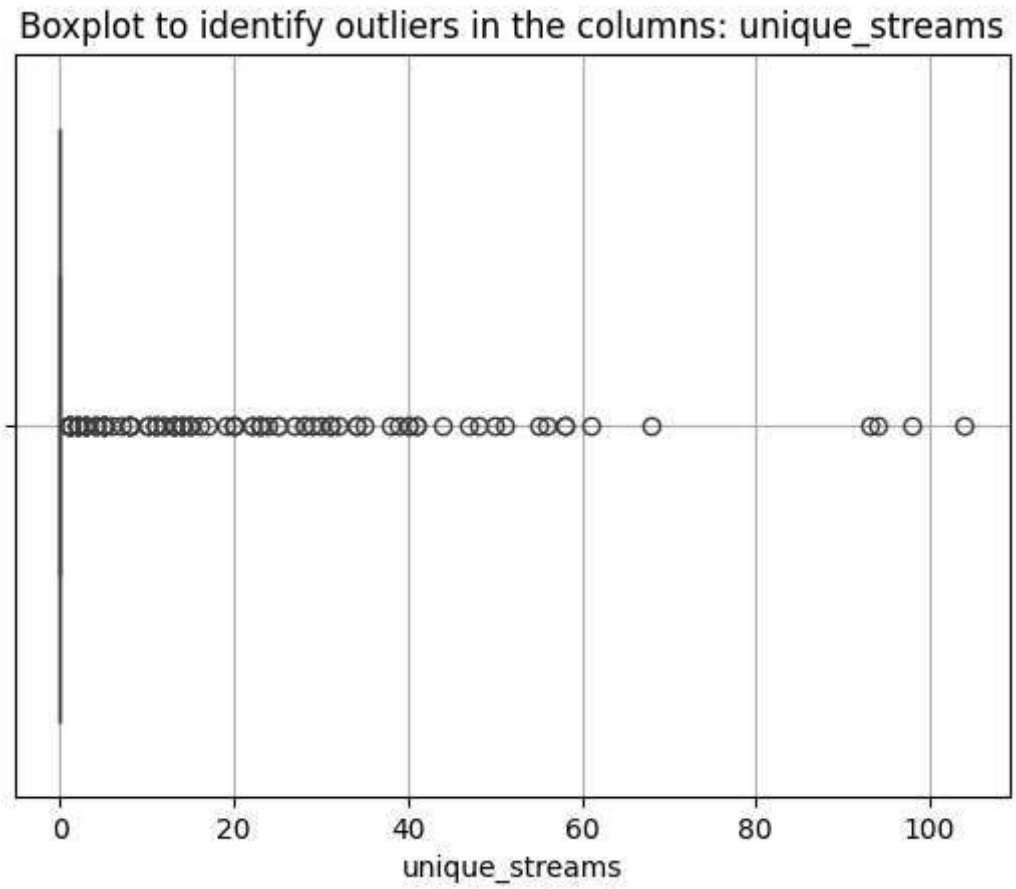


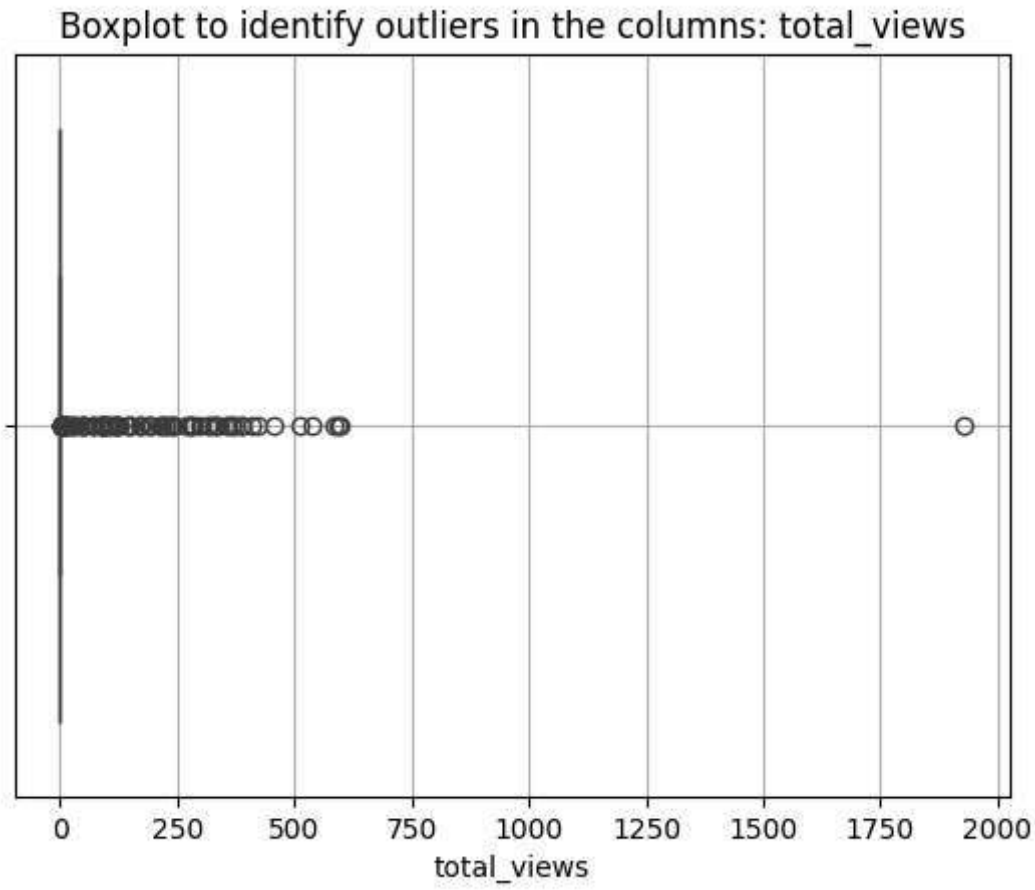
10 - Dataframe df_ts_spain data:

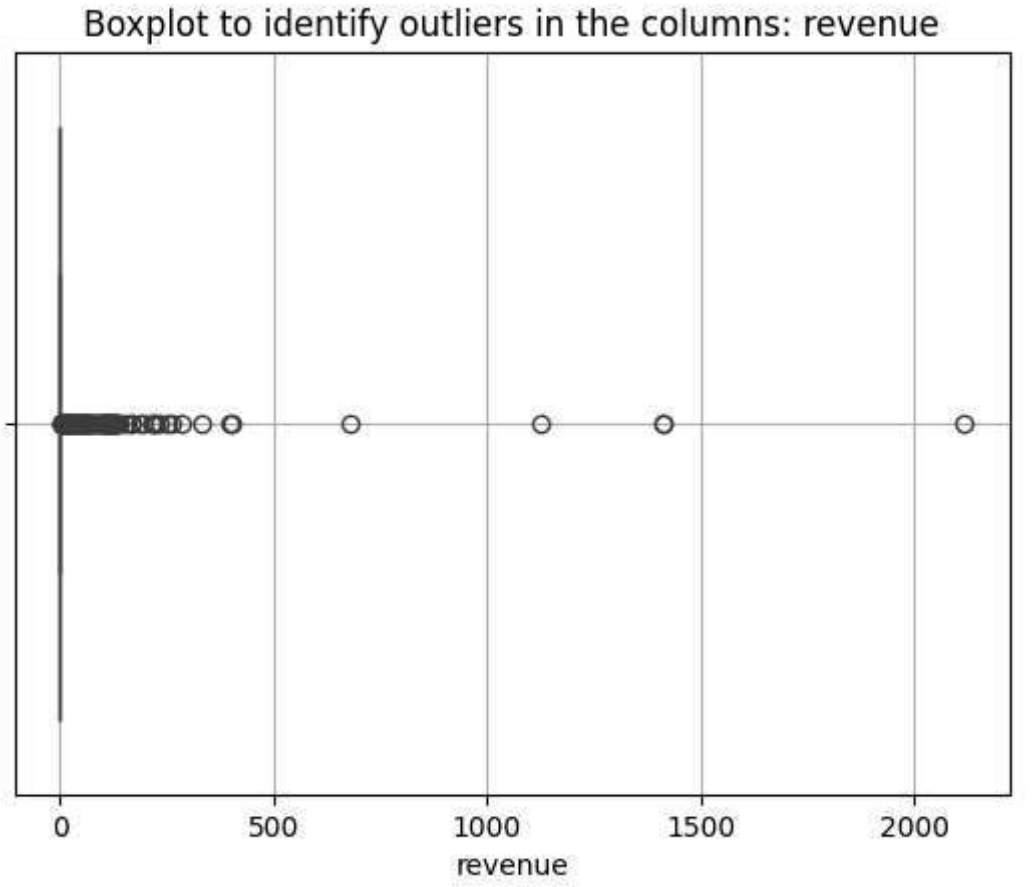

Boxplot to identify outliers in the columns: date
2017-12
2018-01
2018-02
2018-03
2018-05
2018-07
2018-09
2018-11
2019-01
2019-03
2019-05
2019-07
date


Boxplot to identify outliers in the columns: purchases
0
50
100
150
200
purchases

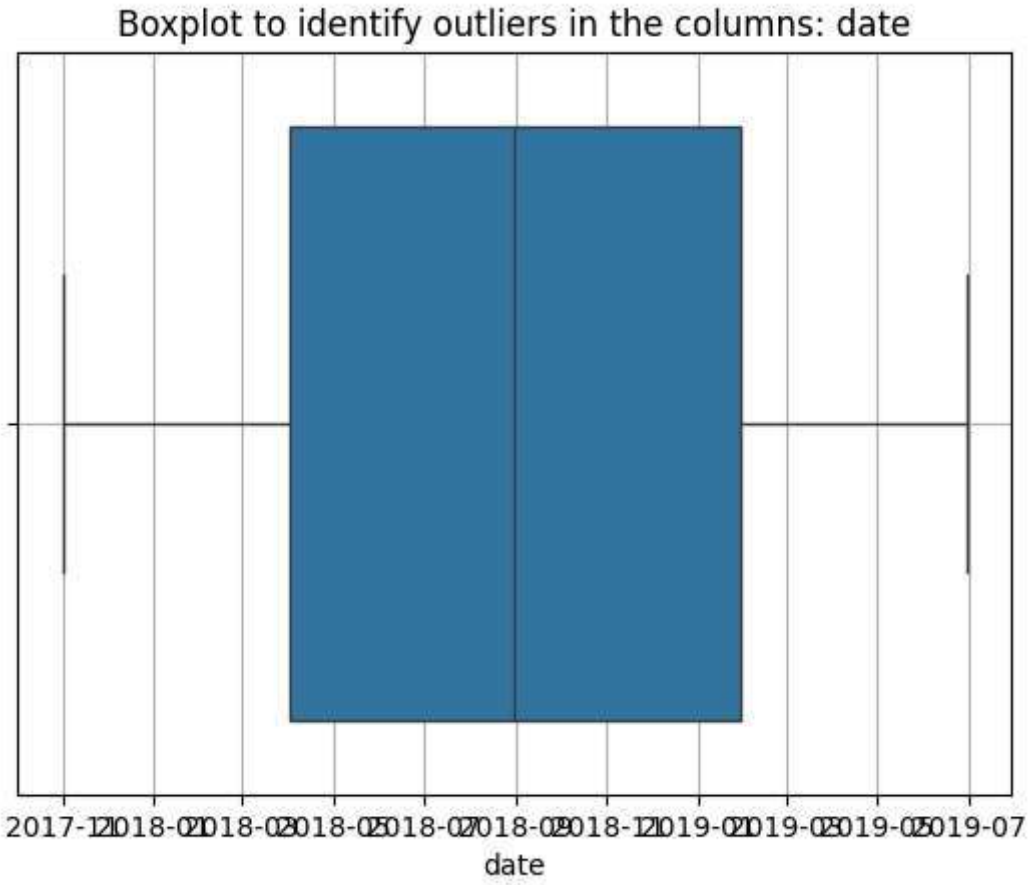

Boxplot to identify outliers in the columns: unique_invoices
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
unique_invoices

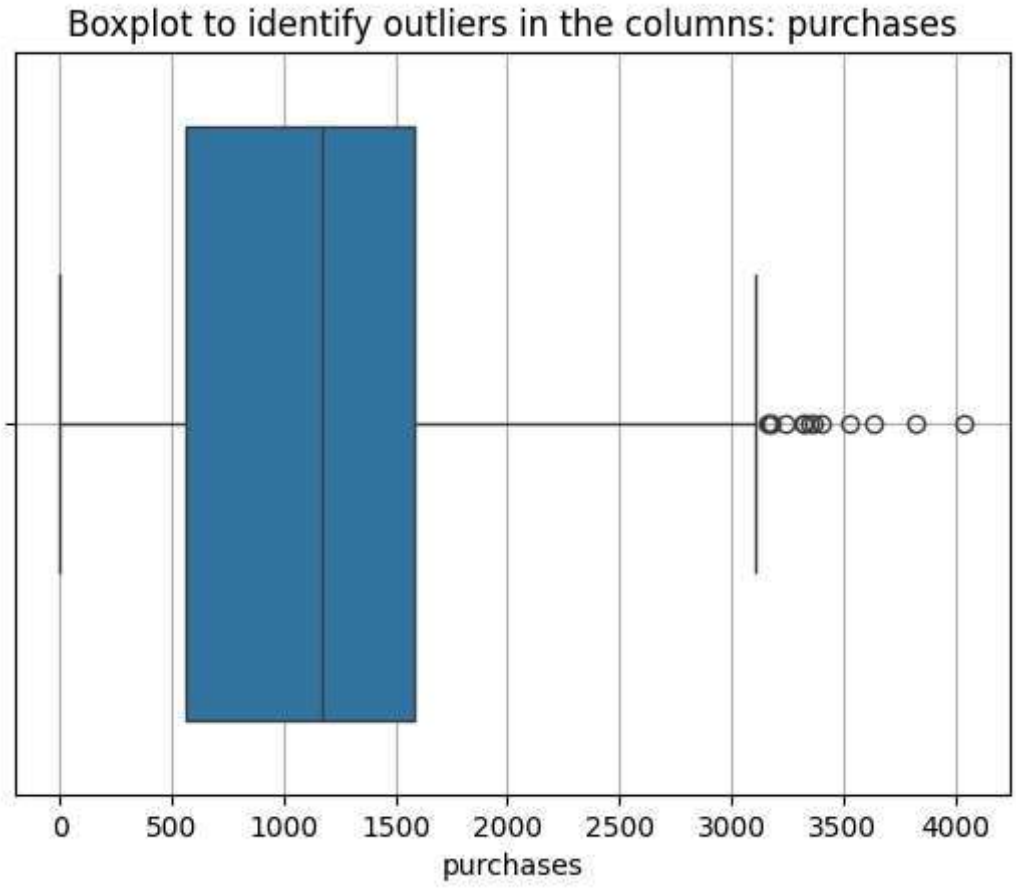

Boxplot to identify outliers in the columns: unique_streams
0
20
40
60
80
100
unique_streams

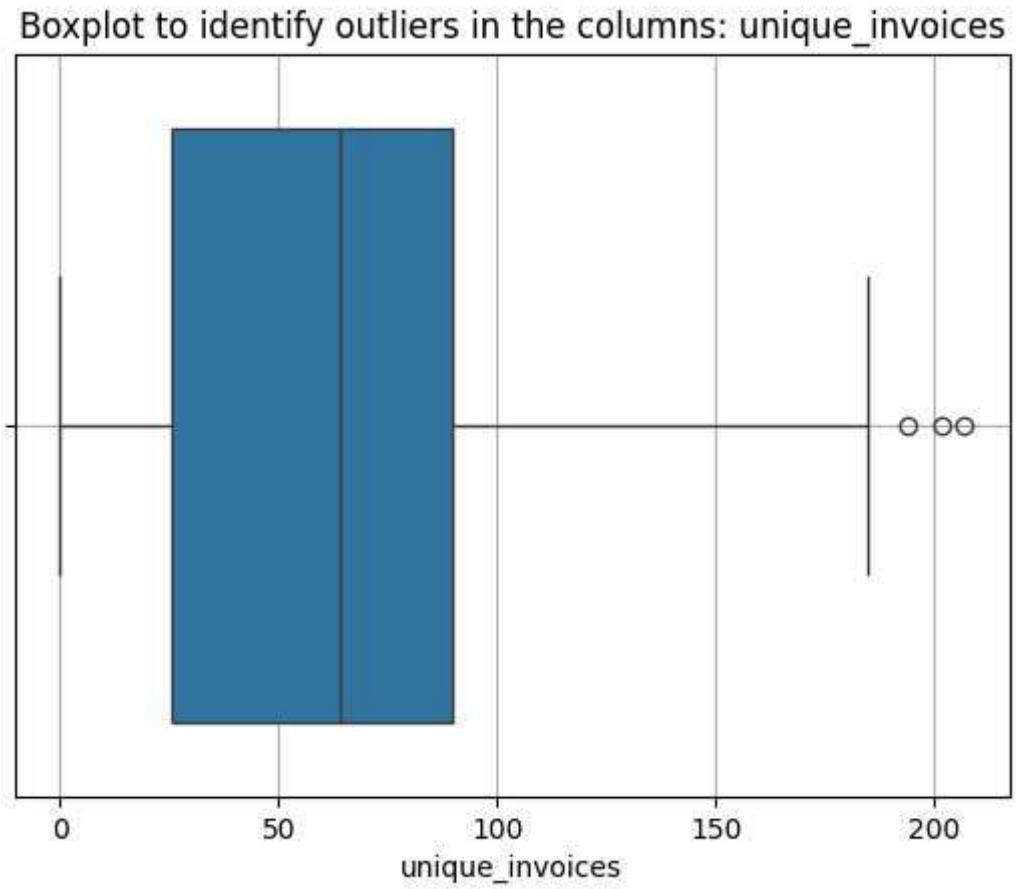

Boxplot to identify outliers in the columns: total_views
0
250
500
750
1000
1250
1500
1750
2000
total_views

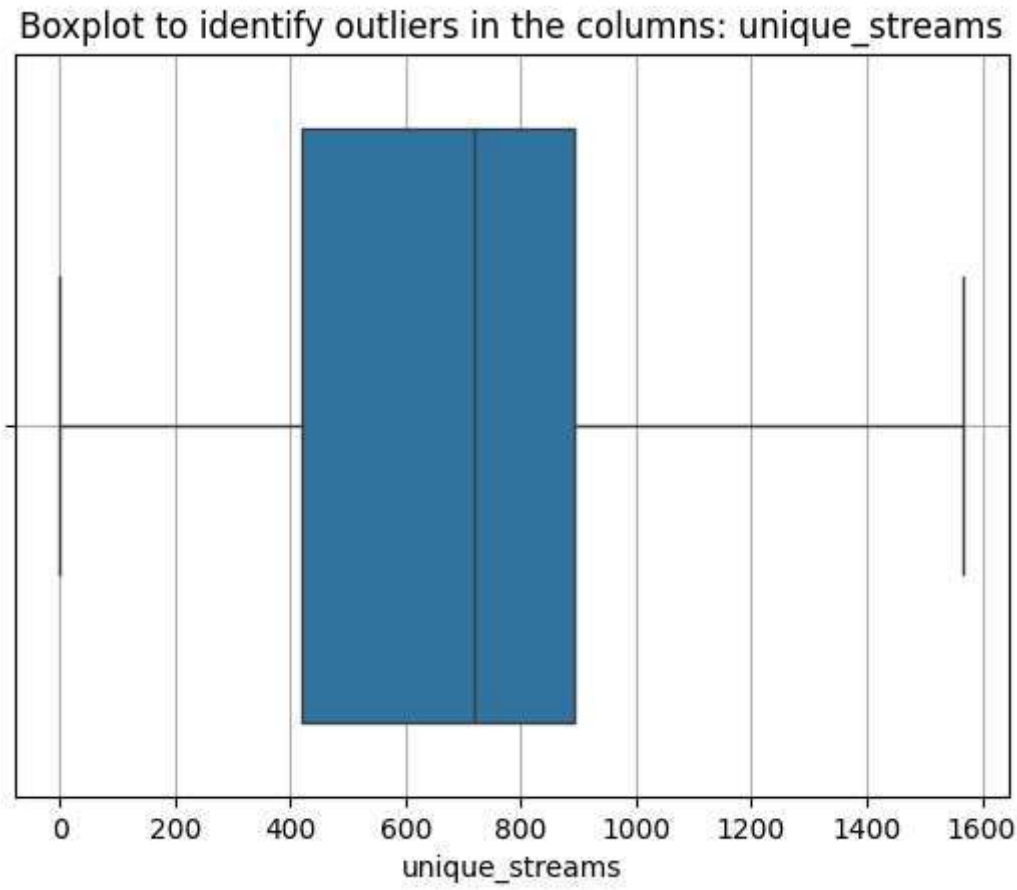

Boxplot to identify outliers in the columns: revenue
0
500
1000
1500
2000
revenue

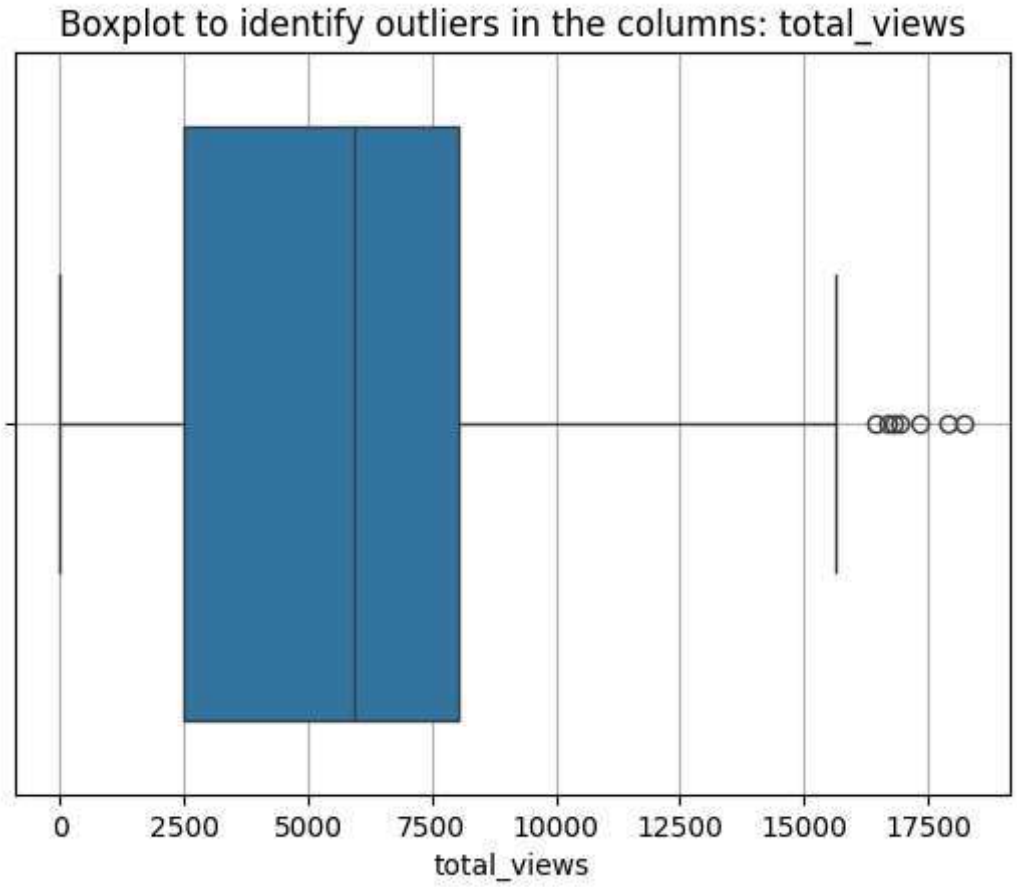
11 - Dataframe df_ts_united_kingdom data:

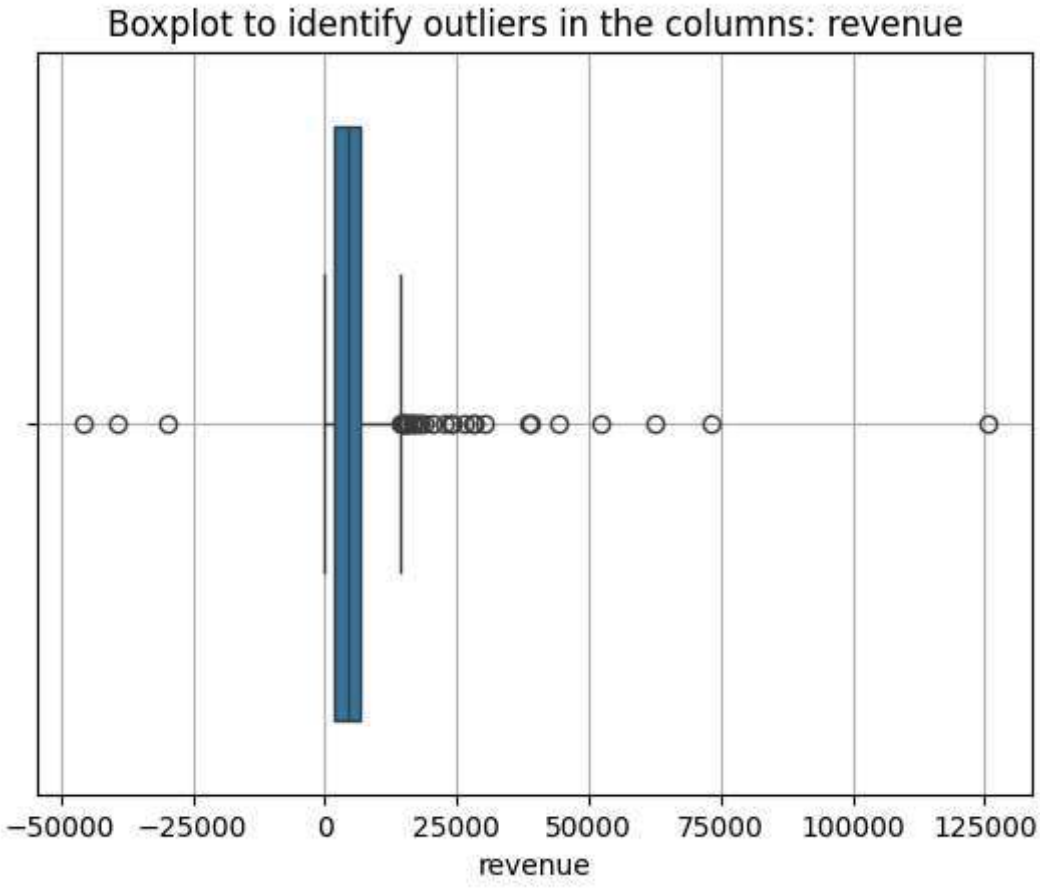



Boxplot to identify outliers in the columns: purchases
0
500
1000
1500
2000
2500
3000
3500
4000
purchases


Boxplot to identify outliers in the columns: unique_invoices
0
50
100
150
200
unique_invoices


Boxplot to identify outliers in the columns: unique_streams
0
200
400
600
800
1000
1200
1400
1600
unique_streams


Boxplot to identify outliers in the columns: total_views
0
2500
5000
7500
10000
12500
15000
17500
total_views

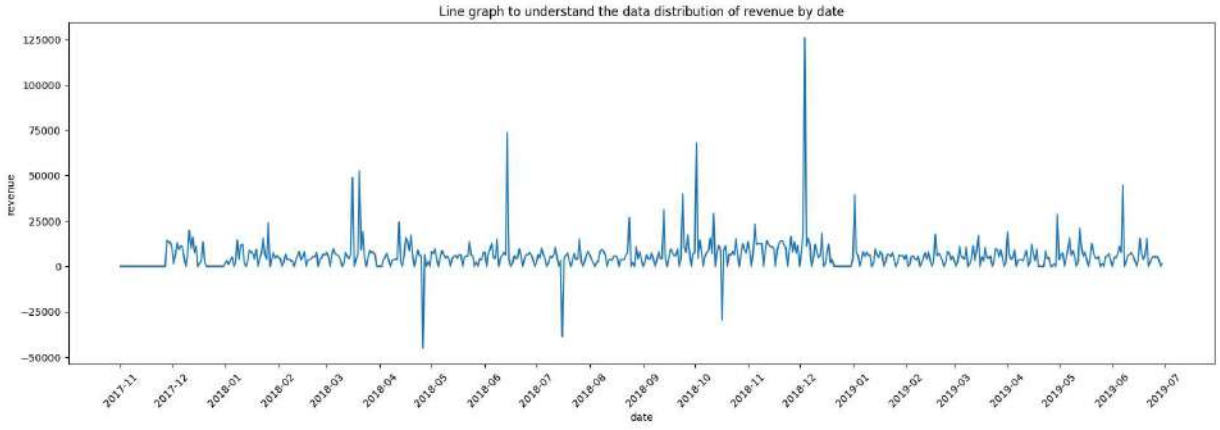


In [17]:
```
library_ts_df['df_ts_all'].columns
```

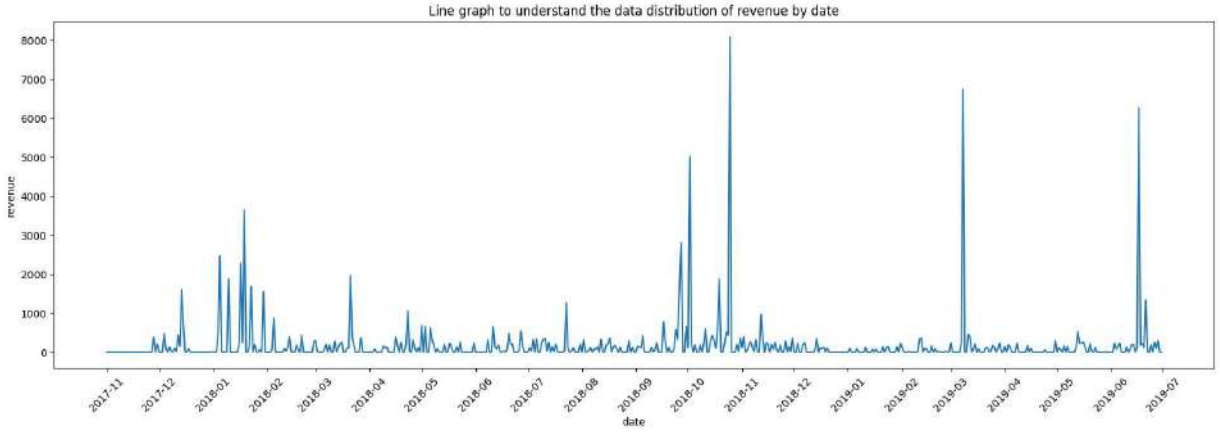
Out[17]:
```
Index(['date', 'purchases', 'unique_invoices', 'unique_streams', 'total_views',
       'revenue'],
      dtype='object')
```

In [18]:
```
#Display a line graph of daily revenue the time serie datasets
for i in range (0,size):
    df_name = list_ts_df[i]
    print(i+1, '- Dataframe', df_name , 'data:')
    viz.show_lineGraph((library_ts_df[df_name]), 'date', 'revenue')
    print('\n')
```

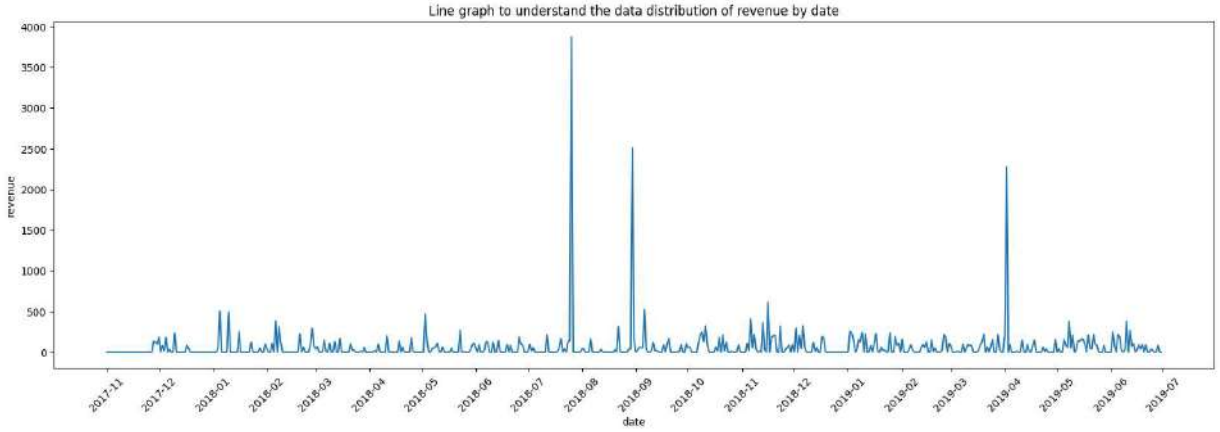
```
1 - Dataframe df_ts_all data:
```



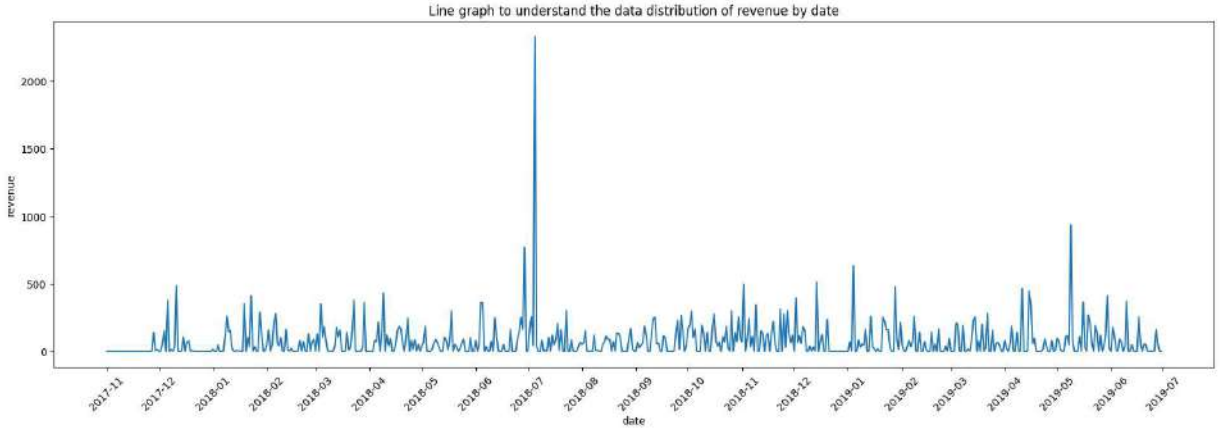
2 - Dataframe df_ts_eire data:



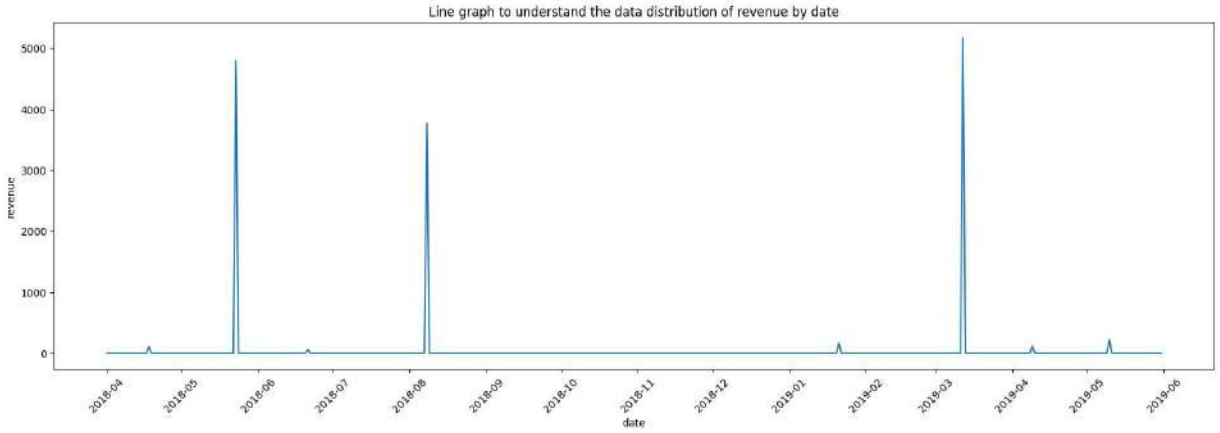
3 - Dataframe df_ts_france data:



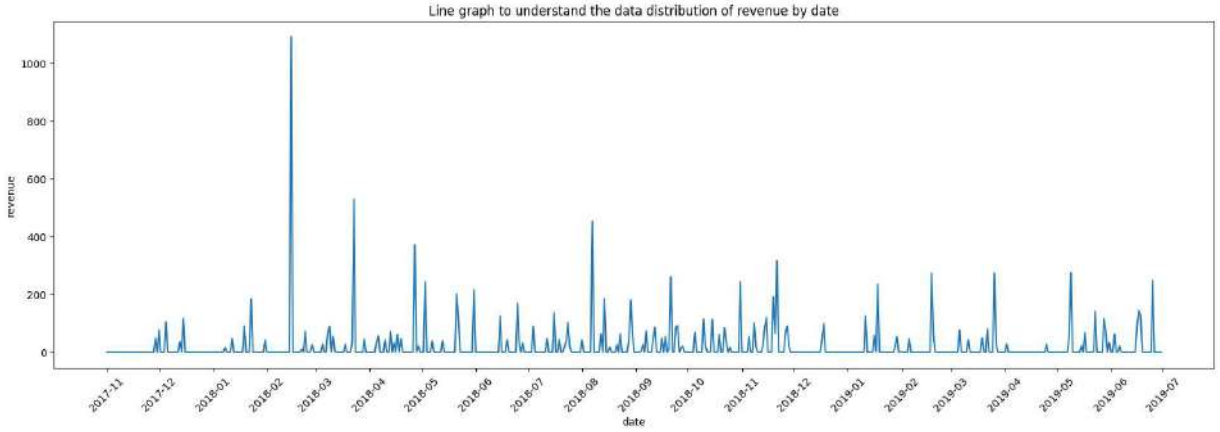
4 - Dataframe df_ts_germany data:



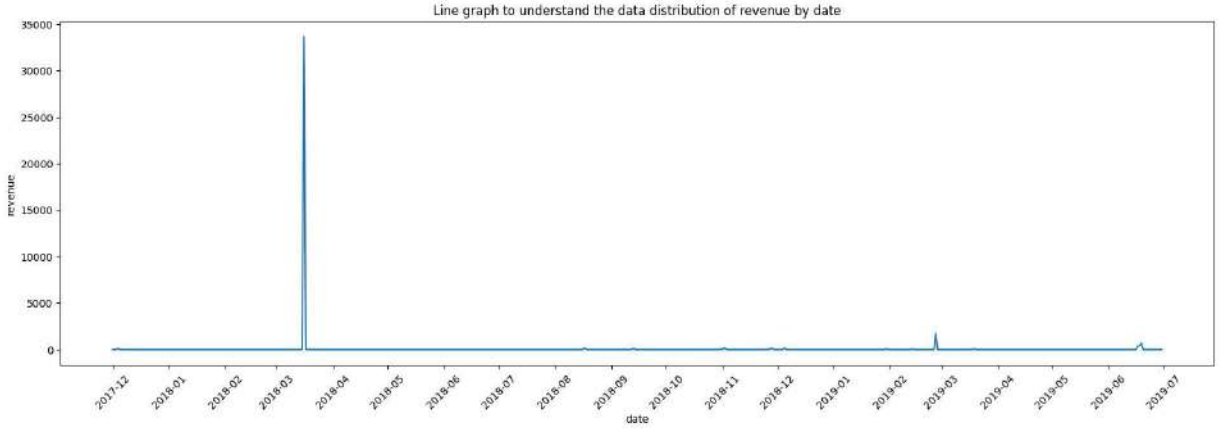
5 - Dataframe df_ts_hong_kong data:



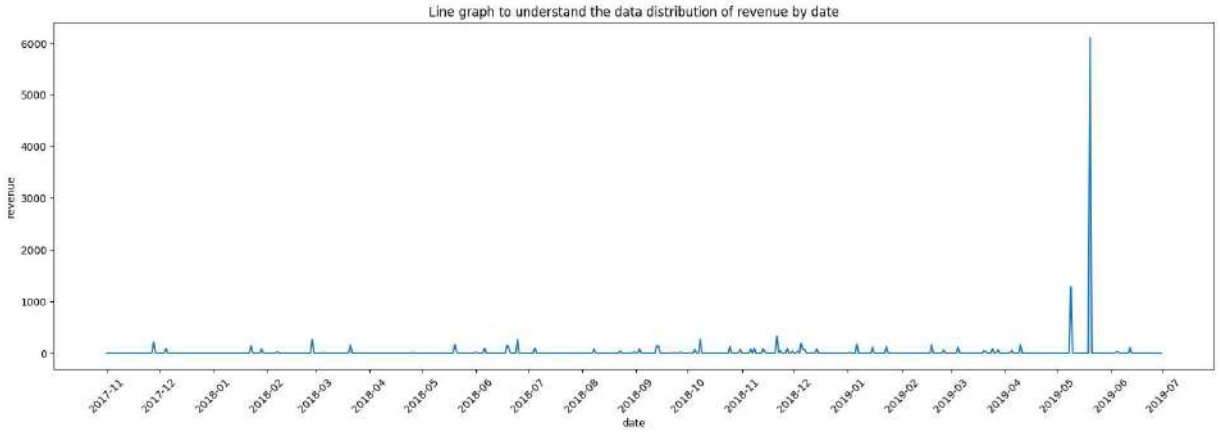
6 - Dataframe df_ts_netherlands data:



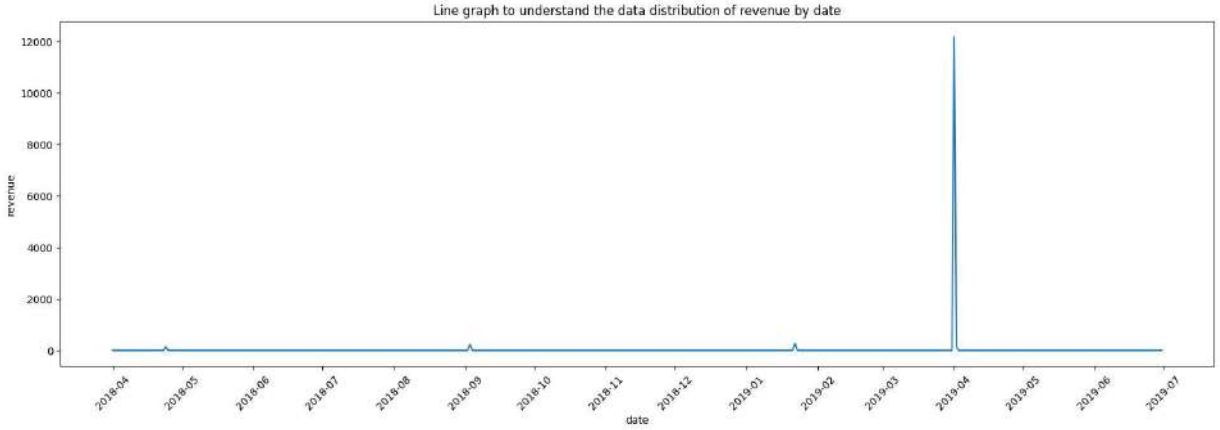
7 - Dataframe df_ts_norway data:



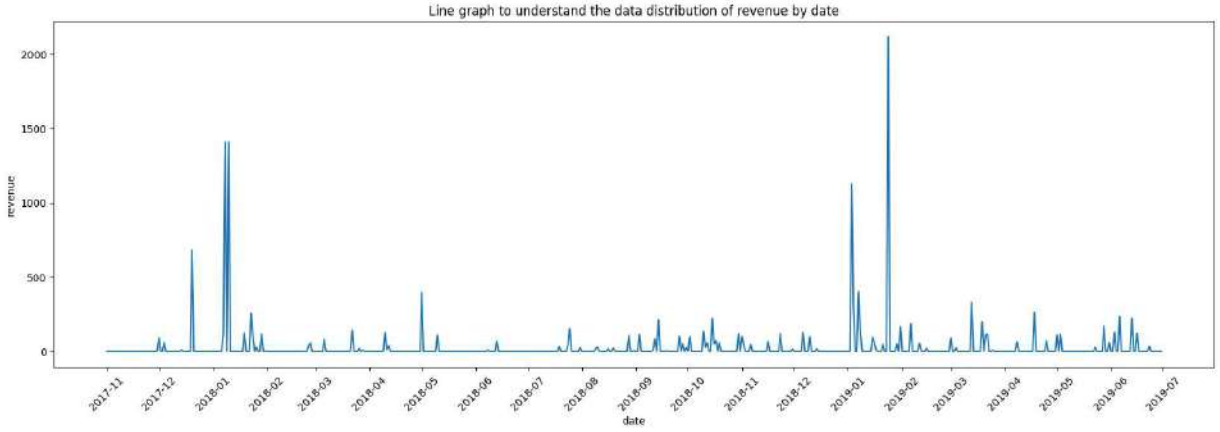
8 - Dataframe df_ts_portugal data:



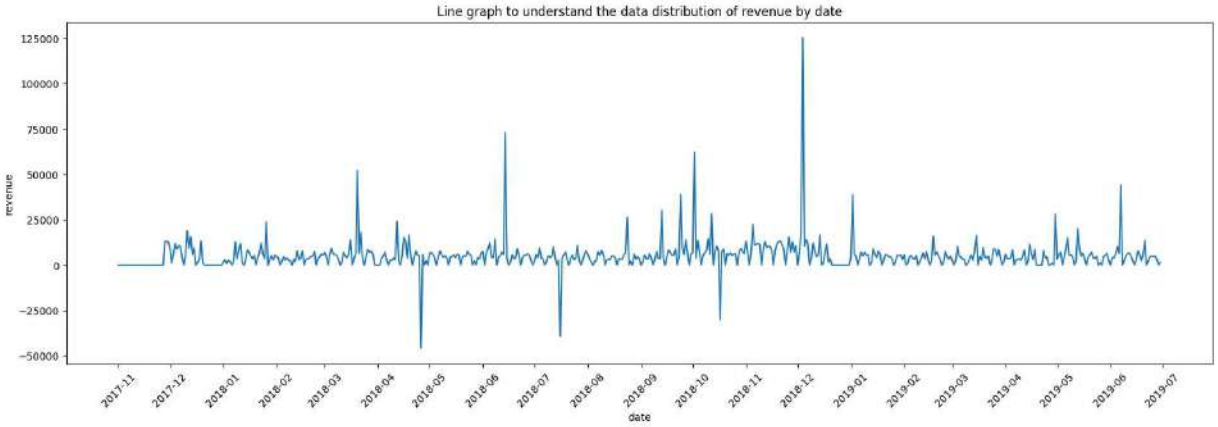
9 - Dataframe df_ts_singapore data:



10 - Dataframe df_ts_spain data:



11 - Dataframe df_ts_united_kingdom data:



2-Create and Test 3 Multi target regression Models using 2 columns month and day as independent variables

This strategy consists of fitting one regressor per target. This is a simple strategy for extending regressors that do not natively support multi-target regression.

2.1.-Preprocessing

In [19]:
```
#Feature engineering: aggregate the data by month and day
library_ts_df_monthly = {}
for i in range (0,size):
    df_name = list_ts_df[i]
    library_ts_df_monthly[df_name] = model.preprocess_data(library_ts_df[df_name])
    print(i+1, '- Dataframe', df_name , 'was aggregated by month and day.')
```

```
1 - Dataframe df_ts_all was aggregated by month and day.
2 - Dataframe df_ts_eire was aggregated by month and day.
3 - Dataframe df_ts_france was aggregated by month and day.
4 - Dataframe df_ts_germany was aggregated by month and day.
5 - Dataframe df_ts_hong_kong was aggregated by month and day.
6 - Dataframe df_ts_netherlands was aggregated by month and day.
7 - Dataframe df_ts_norway was aggregated by month and day.
8 - Dataframe df_ts_portugal was aggregated by month and day.
9 - Dataframe df_ts_singapore was aggregated by month and day.
10 - Dataframe df_ts_spain was aggregated by month and day.
11 - Dataframe df_ts_united_kingdom was aggregated by month and day.
```

In [20]:
```
#Observe data aggregated by month and day
print ("\nOBSERVE DATA")
for i in range (0,size):
    df_name = list_ts_df[i]
    print(i+1, '- Dataframe', df_name , 'data:')
    etl.get_data_info(library_ts_df_monthly[df_name])
    print('\n')
```

OBSERVE DATA
1 - Dataframe df_ts_all data:

FIRST 10 ROWS OF THE DATA

	current_month	current_day	purchases	unique_invoices	unique_streams
0	1	1	918.0	63.5	677.0
1	1	2	1335.0	61.0	828.5
2	1	3	1060.5	46.0	635.0
3	1	4	1427.5	72.5	865.0
4	1	5	356.0	24.5	252.5
5	1	6	546.0	25.5	325.5
6	1	7	1334.0	49.0	799.5
7	1	8	2067.5	79.0	1019.0
8	1	9	1444.5	63.5	929.5
9	1	10	1535.5	57.5	940.5

	total_views	current_revenue	next_month_revenue	next_month	next_day
0	4301.0	3118.810	190490.752	2	1
1	7017.0	21114.805	189528.237	2	2
2	6103.0	4449.060	168808.037	2	3
3	9310.5	4616.005	168161.137	2	4
4	2727.5	2477.890	169749.702	2	5
5	2908.5	1713.620	171288.422	2	6
6	6853.0	4934.665	173333.327	2	7
7	8296.5	9819.200	172630.137	2	8
8	6698.5	5662.135	164311.447	2	9
9	6272.0	8619.230	159563.167	2	10

LAST 10 ROWS OF THE DATA

	current_month	current_day	purchases	unique_invoices	unique_streams
355	12	22	0.0	0.0	0.0
356	12	23	0.0	0.0	0.0
357	12	24	0.0	0.0	0.0
358	12	25	0.0	0.0	0.0
359	12	26	0.0	0.0	0.0
360	12	27	0.0	0.0	0.0
361	12	28	0.0	0.0	0.0
362	12	29	0.0	0.0	0.0
363	12	30	0.0	0.0	0.0
364	12	31	0.0	0.0	0.0

	total_views	current_revenue	next_month_revenue	next_month	next_day
355	0.0	0.0	129155.4005	1	22
356	0.0	0.0	139456.7905	1	23
357	0.0	0.0	146383.0955	1	24
358	0.0	0.0	150259.3055	1	25
359	0.0	0.0	162235.9760	1	26
360	0.0	0.0	163377.9160	1	27
361	0.0	0.0	168410.0210	1	28
362	0.0	0.0	174878.7065	1	29
363	0.0	0.0	180112.4965	1	30
364	0.0	0.0	184987.5220	1	31

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364

```
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT NUMERICAL COLUMNS
                    count           mean           std       min         25%  \
current_month       365.0       6.526027      3.452584      1.00       4.000
current_day         365.0      15.720548      8.808321      1.00       8.000
purchases           365.0    1251.982192    714.770363      0.00     749.500
unique_invoices     365.0      68.671233     39.739072      0.00      43.000
unique_streams      365.0     670.215068    329.179009      0.00     442.000
total_views         365.0    6723.249315   3897.530256      0.00    3847.000
current_revenue     365.0    5945.519214   7712.140511 -38727.48    2773.125
next_month_revenue  365.0  190256.614838  57352.263842  94646.52  150261.410
next_month          365.0       6.526027      3.452584      1.00       4.000
next_day            365.0      15.720548      8.808321      1.00       8.000
```

```
                          50%         75%         max
current_month           7.000      10.000      12.000
current_day            16.000      23.000      31.000
purchases            1315.500    1659.000    3611.000
unique_invoices        71.000      93.000     206.000
unique_streams        732.500     892.500    1384.000
total_views          7126.500    8952.000   18500.000
current_revenue      5288.340    7499.005   69502.710
next_month_revenue 180276.181  217124.620  356034.091
next_month              7.000      10.000      12.000
next_day               16.000      23.000      31.000
current_month : 12
current_day : 31
purchases : 317
unique_invoices : 179
unique_streams : 295
total_views : 329
current_revenue : 331
next_month_revenue : 364
next_month : 12
next_day : 31

There is no column that includes a unique category of data


2 - Dataframe df_ts_eire data:

FIRST 10 ROWS OF THE DATA
   current_month  current_day  purchases  unique_invoices  unique_streams  \
0              1            1        1.0              0.5             1.0
1              1            2        7.5              0.5             7.5
2              1            3        0.0              0.0             0.0
3              1            4       81.5              2.5            79.0
4              1            5       16.5              2.0            15.5
5              1            6        6.5              0.5             6.5
6              1            7        0.5              0.5             0.5
7              1            8        0.5              0.5             0.5
8              1            9        0.0              0.0             0.0
9              1           10        1.5              1.5             0.5

   total_views  current_revenue  next_month_revenue  next_month  next_day
0          2.0            7.000              8076.660           2         1
1         20.0           44.885              8069.660           2         2
2          0.0            0.000              8024.775           2         3
3        913.0          233.275              8057.415           2         4
4        130.0         1231.835              8266.740           2         5
5         42.5           41.850              7034.905           2         6
6          1.5            1.875              6993.055           2         7
7          4.0            5.475              6991.180           2         8
8          0.0            0.000              6985.705           2         9
9          4.5          937.210              6985.705           2        10

LAST 10 ROWS OF THE DATA
     current_month  current_day  purchases  unique_invoices  unique_streams  \
355             12           22        0.0              0.0             0.0
356             12           23        0.0              0.0             0.0
```

```
357             12           24        0.0              0.0             0.0
358             12           25        0.0              0.0             0.0
359             12           26        0.0              0.0             0.0
360             12           27        0.0              0.0             0.0
361             12           28        0.0              0.0             0.0
362             12           29        0.0              0.0             0.0
363             12           30        0.0              0.0             0.0
364             12           31        0.0              0.0             0.0

     total_views  current_revenue  next_month_revenue  next_month  next_day
355          0.0              0.0            5972.495           1        22
356          0.0              0.0            6875.765           1        23
357          0.0              0.0            6945.725           1        24
358          0.0              0.0            7036.985           1        25
359          0.0              0.0            7036.985           1        26
360          0.0              0.0            7036.985           1        27
361          0.0              0.0            7067.385           1        28
362          0.0              0.0            7137.565           1        29
363          0.0              0.0            7912.230           1        30
364          0.0              0.0            8023.635           1        31

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
```

```
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT NUMERICAL COLUMNS
                    count         mean          std       min      25%  \
current_month       365.0     6.526027     3.452584     1.000     4.00
current_day         365.0    15.720548     8.808321     1.000     8.00
purchases           365.0    22.841096    32.300690     0.000     0.00
unique_invoices     365.0     1.065753     1.328494     0.000     0.00
unique_streams      365.0    21.708219    29.727271     0.000     0.00
total_views         365.0   188.047945   288.215178     0.000     0.00
current_revenue     365.0   190.619438   607.363285     0.000     0.00
next_month_revenue  365.0  6099.822027  4512.525951  2013.395  3049.11
next_month          365.0     6.526027     3.452584     1.000     4.00
next_day            365.0    15.720548     8.808321     1.000     8.00

                        50%       75%       max
current_month         7.000    10.000     12.00
current_day          16.000    23.000     31.00
purchases            11.000    33.000    240.00
unique_invoices       0.500     1.500      8.00
unique_streams       10.500    32.500    211.00
total_views          74.500   273.000   2558.00
current_revenue      55.595   160.515   8086.08
next_month_revenue 4038.325  7028.660  25412.51
next_month            7.000    10.000     12.00
next_day             16.000    23.000     31.00
current_month : 12
current_day : 31
purchases : 122
unique_invoices : 14
unique_streams : 122
total_views : 202
current_revenue : 235
next_month_revenue : 327
next_month : 12
next_day : 31

There is no column that includes a unique category of data


3 - Dataframe df_ts_france data:

FIRST 10 ROWS OF THE DATA
   current_month  current_day  purchases  unique_invoices  unique_streams  \
0              1            1        0.0              0.0             0.0
1              1            2       38.0              2.0            33.0
2              1            3       23.5              1.5            22.0
```

```
3              1           4       37.5             1.5            36.0
4              1           5       62.5             1.5            60.5
5              1           6        4.5             0.5             4.5
6              1           7       25.0             0.5            25.0
7              1           8       15.0             2.0            13.5
8              1           9       31.5             2.0            30.0
9              1          10       22.0             1.5            22.0

   total_views  current_revenue  next_month_revenue  next_month  next_day
0          0.0            0.000             2159.170           2         1
1        327.5          127.640             2159.170           2         2
2        157.5          117.775             2031.530           2         3
3        395.5          105.900             1966.830           2         4
4        605.0          250.595             1876.225           2         5
5         39.0           10.100             1860.710           2         6
6        233.0           74.365             1864.965           2         7
7        126.5           64.070             1943.175           2         8
8        266.0          120.635             1935.350           2         9
9        153.0          245.310             1814.715           2        10

LAST 10 ROWS OF THE DATA
     current_month  current_day  purchases  unique_invoices  unique_streams  \
355             12           22        0.0              0.0             0.0
356             12           23        0.0              0.0             0.0
357             12           24        0.0              0.0             0.0
358             12           25        0.0              0.0             0.0
359             12           26        0.0              0.0             0.0
360             12           27        0.0              0.0             0.0
361             12           28        0.0              0.0             0.0
362             12           29        0.0              0.0             0.0
363             12           30        0.0              0.0             0.0
364             12           31        0.0              0.0             0.0

     total_views  current_revenue  next_month_revenue  next_month  next_day
355          0.0              0.0            1608.715           1        22
356          0.0              0.0            1669.545           1        23
357          0.0              0.0            1672.270           1        24
358          0.0              0.0            1792.170           1        25
359          0.0              0.0            1792.170           1        26
360          0.0              0.0            1792.170           1        27
361          0.0              0.0            1909.510           1        28
362          0.0              0.0            1950.535           1        29
363          0.0              0.0            2004.780           1        30
364          0.0              0.0            2054.700           1        31

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
```

```
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT NUMERICAL COLUMNS
                    count         mean          std     min      25%  \
current_month       365.0     6.526027     3.452584    1.00     4.00
current_day         365.0    15.720548     8.808321    1.00     8.00
purchases           365.0    15.254795    22.314363    0.00     0.00
unique_invoices     365.0     0.831507     0.932307    0.00     0.00
unique_streams      365.0    14.221918    19.810990    0.00     0.00
total_views         365.0   127.552055   193.825049    0.00     0.00
current_revenue     365.0    68.035301   253.125726    0.00     0.00
next_month_revenue  365.0  2177.129644  1067.425153  609.77  1551.87
next_month          365.0     6.526027     3.452584    1.00     4.00
next_day            365.0    15.720548     8.808321    1.00     8.00

                      50%      75%      max
current_month       7.000    10.00    12.00
current_day        16.000    23.00    31.00
purchases           5.500    24.50   179.00
unique_invoices     0.500     1.50     6.00
unique_streams      5.500    23.50   144.00
total_views        44.000   205.00  1563.00
current_revenue    18.205    80.22  3871.40
```

```
next_month_revenue  1850.130  2159.17  5011.77
next_month             7.000    10.00    12.00
next_day              16.000    23.00    31.00
current_month : 12
current_day : 31
purchases : 105
unique_invoices : 10
unique_streams : 101
total_views : 194
current_revenue : 228
next_month_revenue : 321
next_month : 12
next_day : 31

There is no column that includes a unique category of data


4 - Dataframe df_ts_germany data:

FIRST 10 ROWS OF THE DATA
   current_month  current_day  purchases  unique_invoices  unique_streams  \
0              1            1        1.5              0.5             1.5
1              1            2        8.5              1.0             8.5
2              1            3        0.5              0.5             0.5
3              1            4      121.0              3.0           104.0
4              1            5        0.0              0.0             0.0
5              1            6        0.0              0.0             0.0
6              1            7       11.0              1.5            10.5
7              1            8       21.5              2.5            21.0
8              1            9       50.5              3.0            46.0
9              1           10       23.5              1.5            23.5

   total_views  current_revenue  next_month_revenue  next_month  next_day
0          4.0            8.145            2620.935           2         1
1         16.5           35.745            2690.205           2         2
2          2.0            0.825            2654.460           2         3
3       1016.0          341.745            2693.430           2         4
4          0.0            0.000            2498.340           2         5
5          0.0            0.000            2653.900           2         6
6         83.0           41.795            2718.295           2         7
7        150.0           70.700            2826.400           2         8
8        472.5          155.200            2807.610           2         9
9        202.5           95.285            2652.410           2        10

LAST 10 ROWS OF THE DATA
     current_month  current_day  purchases  unique_invoices  unique_streams  \
355             12           22        0.0              0.0             0.0
356             12           23        0.0              0.0             0.0
357             12           24        0.0              0.0             0.0
358             12           25        0.0              0.0             0.0
359             12           26        0.0              0.0             0.0
360             12           27        0.0              0.0             0.0
361             12           28        0.0              0.0             0.0
362             12           29        0.0              0.0             0.0
363             12           30        0.0              0.0             0.0
364             12           31        0.0              0.0             0.0
```

```
     total_views  current_revenue  next_month_revenue  next_month  next_day
355          0.0              0.0            1570.530           1        22
356          0.0              0.0            1854.770           1        23
357          0.0              0.0            1934.435           1        24
358          0.0              0.0            1964.725           1        25
359          0.0              0.0            1964.725           1        26
360          0.0              0.0            1964.725           1        27
361          0.0              0.0            2348.090           1        28
362          0.0              0.0            2457.845           1        29
363          0.0              0.0            2465.695           1        30
364          0.0              0.0            2574.270           1        31

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

MISSING DATA
There is no missing data
```

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT NUMERICAL COLUMNS

	count	mean	std	min	25%	50%	75%	max
current_month	365.0	6.526027	3.452584	1.000	4.000	7.000	10.00	12.00
current_day	365.0	15.720548	8.808321	1.000	8.000	16.000	23.00	31.00
purchases	365.0	20.493151	24.853457	0.000	0.000	12.000	32.00	121.00
unique_invoices	365.0	1.293151	1.308922	0.000	0.000	1.000	2.00	8.00
unique_streams	365.0	19.256164	22.865006	0.000	0.000	11.500	29.00	106.00
total_views	365.0	173.835616	223.460107	0.000	0.000	89.000	257.50	1100.00
current_revenue	365.0	76.629700	144.423917	0.000	0.000	46.150	111.32	2329.68
next_month_revenue	365.0	2452.150400	770.255900	1247.475	1964.725	2351.095	2654.46	4844.17
next_month	365.0	6.526027	3.452584	1.000	4.000	7.000	10.00	12.00
next_day	365.0	15.720548	8.808321	1.000	8.000	16.000	23.00	31.00

current_month : 12
current_day : 31
purchases : 116
unique_invoices : 13
unique_streams : 116
total_views : 218
current_revenue : 259
next_month_revenue : 342
next_month : 12
next_day : 31

There is no column that includes a unique category of data

5 - Dataframe df_ts_hong_kong data:

FIRST 10 ROWS OF THE DATA

	current_month	current_day	purchases	unique_invoices	unique_streams
0	1	1	0.0	0.0	0.0
1	1	2	0.0	0.0	0.0
2	1	3	0.0	0.0	0.0
3	1	4	0.0	0.0	0.0
4	1	5	0.0	0.0	0.0
5	1	6	0.0	0.0	0.0
6	1	7	0.0	0.0	0.0
7	1	8	0.0	0.0	0.0
8	1	9	0.0	0.0	0.0
9	1	10	0.0	0.0	0.0

```
   total_views  current_revenue  next_month_revenue  next_month  next_day
0          0.0              0.0              167.66           2         1
1          0.0              0.0              167.66           2         2
2          0.0              0.0              167.66           2         3
3          0.0              0.0              167.66           2         4
4          0.0              0.0              167.66           2         5
5          0.0              0.0              167.66           2         6
6          0.0              0.0              167.66           2         7
7          0.0              0.0              167.66           2         8
8          0.0              0.0              167.66           2         9
9          0.0              0.0              167.66           2        10

LAST 10 ROWS OF THE DATA
     current_month  current_day  purchases  unique_invoices  unique_streams  \
355             12           22        0.0              0.0             0.0
356             12           23        0.0              0.0             0.0
357             12           24        0.0              0.0             0.0
358             12           25        0.0              0.0             0.0
359             12           26        0.0              0.0             0.0
360             12           27        0.0              0.0             0.0
361             12           28        0.0              0.0             0.0
362             12           29        0.0              0.0             0.0
363             12           30        0.0              0.0             0.0
364             12           31        0.0              0.0             0.0

     total_views  current_revenue  next_month_revenue  next_month  next_day
355          0.0              0.0              167.66           1        22
356          0.0              0.0              167.66           1        23
357          0.0              0.0              167.66           1        24
358          0.0              0.0              167.66           1        25
359          0.0              0.0              167.66           1        26
360          0.0              0.0              167.66           1        27
361          0.0              0.0              167.66           1        28
362          0.0              0.0              167.66           1        29
363          0.0              0.0              167.66           1        30
364          0.0              0.0              167.66           1        31

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None
```

```
BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT NUMERICAL COLUMNS
                    count         mean          std  min  25%   50%      75%  \
current_month       365.0     6.526027     3.452584  1.0  4.0   7.0    10.00
current_day         365.0    15.720548     8.808321  1.0  8.0  16.0    23.00
purchases           365.0     0.467123     4.233762  0.0  0.0   0.0     0.00
unique_invoices     365.0     0.023288     0.178147  0.0  0.0   0.0     0.00
unique_streams      365.0     0.438356     3.961131  0.0  0.0   0.0     0.00
total_views         365.0     4.491781    41.152339  0.0  0.0   0.0     0.00
current_revenue     365.0    32.245219   356.681660  0.0  0.0   0.0     0.00
next_month_revenue  365.0  1031.847014  1753.107180  0.0  0.0  54.6  2396.61
next_month          365.0     6.526027     3.452584  1.0  4.0   7.0    10.00
next_day            365.0    15.720548     8.808321  1.0  8.0  16.0    23.00

                       max
current_month         12.000
current_day           31.000
purchases             55.000
unique_invoices        2.000
unique_streams        47.000
total_views          528.000
current_revenue     5167.520
next_month_revenue  5224.225
next_month            12.000
next_day              31.000
current_month : 12
current_day : 31
purchases : 7
unique_invoices : 4
unique_streams : 7
```

total_views : 9
current_revenue : 9
next_month_revenue : 14
next_month : 12
next_day : 31

There is no column that includes a unique category of data

6 - Dataframe df_ts_netherlands data:

FIRST 10 ROWS OF THE DATA

	current_month	current_day	purchases	unique_invoices	unique_streams
0	1	1	0.0	0.0	0.0
1	1	2	0.0	0.0	0.0
2	1	3	0.0	0.0	0.0
3	1	4	0.0	0.0	0.0
4	1	5	0.0	0.0	0.0
5	1	6	0.0	0.0	0.0
6	1	7	0.0	0.0	0.0
7	1	8	1.0	0.5	1.0
8	1	9	0.0	0.0	0.0
9	1	10	0.0	0.0	0.0

	total_views	current_revenue	next_month_revenue	next_month	next_day
0	0.0	0.00	438.785	2	1
1	0.0	0.00	438.785	2	2
2	0.0	0.00	438.785	2	3
3	0.0	0.00	438.785	2	4
4	0.0	0.00	461.920	2	5
5	0.0	0.00	461.920	2	6
6	0.0	0.00	461.920	2	7
7	3.5	7.82	461.920	2	8
8	0.0	0.00	454.100	2	9
9	0.0	0.00	454.100	2	10

LAST 10 ROWS OF THE DATA

	current_month	current_day	purchases	unique_invoices	unique_streams
355	12	22	0.0	0.0	0.0
356	12	23	0.0	0.0	0.0
357	12	24	0.0	0.0	0.0
358	12	25	0.0	0.0	0.0
359	12	26	0.0	0.0	0.0
360	12	27	0.0	0.0	0.0
361	12	28	0.0	0.0	0.0
362	12	29	0.0	0.0	0.0
363	12	30	0.0	0.0	0.0
364	12	31	0.0	0.0	0.0

	total_views	current_revenue	next_month_revenue	next_month	next_day
355	0.0	0.0	286.845	1	22
356	0.0	0.0	378.965	1	23
357	0.0	0.0	378.965	1	24
358	0.0	0.0	378.965	1	25
359	0.0	0.0	378.965	1	26
360	0.0	0.0	378.965	1	27

```
361          0.0              0.0             390.535           1        28
362          0.0              0.0             417.310           1        29
363          0.0              0.0             417.310           1        30
364          0.0              0.0             438.785           1        31

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT NUMERICAL COLUMNS
                   count        mean         std     min     25%      50%  \
current_month      365.0    6.526027    3.452584    1.00    4.00    7.000
current_day        365.0   15.720548    8.808321    1.00    8.00   16.000
purchases          365.0    6.990411   15.754575    0.00    0.00    0.000
```

```
unique_invoices     365.0    0.353425    0.638526    0.00    0.00    0.000
unique_streams      365.0    6.852055   15.397883    0.00    0.00    0.000
total_views         365.0   68.865753  174.780317    0.00    0.00    0.000
current_revenue     365.0   21.207685   54.001669    0.00    0.00    0.000
next_month_revenue  365.0  678.645918  218.967898  149.05  545.77  696.005
next_month          365.0    6.526027    3.452584    1.00    4.00    7.000
next_day            365.0   15.720548    8.808321    1.00    8.00   16.000

                        75%       max
current_month        10.000    12.000
current_day          23.000    31.000
purchases             6.000   115.000
unique_invoices       0.500     4.000
unique_streams        6.000   114.000
total_views          40.000  1368.000
current_revenue      20.400   545.765
next_month_revenue  838.905  1194.630
next_month           10.000    12.000
next_day             23.000    31.000
current_month : 12
current_day : 31
purchases : 69
unique_invoices : 8
unique_streams : 65
total_views : 114
current_revenue : 127
next_month_revenue : 216
next_month : 12
next_day : 31

There is no column that includes a unique category of data


7 - Dataframe df_ts_norway data:

FIRST 10 ROWS OF THE DATA
   current_month  current_day  purchases  unique_invoices  unique_streams  \
0              1            1        0.0              0.0             0.0
1              1            2        0.0              0.0             0.0
2              1            3        0.0              0.0             0.0
3              1            4        0.0              0.0             0.0
4              1            5        0.0              0.0             0.0
5              1            6        0.0              0.0             0.0
6              1            7        0.0              0.0             0.0
7              1            8        0.0              0.0             0.0
8              1            9        0.0              0.0             0.0
9              1           10        0.0              0.0             0.0

   total_views  current_revenue  next_month_revenue  next_month  next_day
0          0.0              0.0               32.65           2         1
1          0.0              0.0               32.65           2         2
2          0.0              0.0               32.65           2         3
3          0.0              0.0               32.65           2         4
4          0.0              0.0               32.65           2         5
5          0.0              0.0               32.65           2         6
6          0.0              0.0               32.65           2         7
```

```
7          0.0             0.0              32.65          2         8
8          0.0             0.0              32.65          2         9
9          0.0             0.0              32.65          2        10

LAST 10 ROWS OF THE DATA
     current_month  current_day  purchases  unique_invoices  unique_streams  \
355             12           22        0.0              0.0             0.0
356             12           23        0.0              0.0             0.0
357             12           24        0.0              0.0             0.0
358             12           25        0.0              0.0             0.0
359             12           26        0.0              0.0             0.0
360             12           27        0.0              0.0             0.0
361             12           28        0.0              0.0             0.0
362             12           29        0.0              0.0             0.0
363             12           30        0.0              0.0             0.0
364             12           31        0.0              0.0             0.0

     total_views  current_revenue  next_month_revenue  next_month  next_day
355          0.0              0.0                0.00           1        22
356          0.0              0.0                0.00           1        23
357          0.0              0.0                0.00           1        24
358          0.0              0.0                0.00           1        25
359          0.0              0.0                0.00           1        26
360          0.0              0.0                0.00           1        27
361          0.0              0.0                0.00           1        28
362          0.0              0.0                0.00           1        29
363          0.0              0.0               32.65           1        30
364          0.0              0.0               32.65           1        31

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
```

```
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT NUMERICAL COLUMNS
                    count         mean          std  min  25%     50%     75%  \
current_month       365.0     6.526027     3.452584  1.0  4.0    7.00   10.00
current_day         365.0    15.720548     8.808321  1.0  8.0   16.00   23.00
purchases           365.0     1.371233     9.224750  0.0  0.0    0.00    0.00
unique_invoices     365.0     0.049315     0.251983  0.0  0.0    0.00    0.00
unique_streams      365.0     1.167123     7.498957  0.0  0.0    0.00    0.00
total_views         365.0    14.100000   102.645315  0.0  0.0    0.00    0.00
current_revenue     365.0    53.319740   881.390757  0.0  0.0    0.00    0.00
next_month_revenue  365.0  1706.231671  4830.176642  0.0  0.0  139.74  348.62
next_month          365.0     6.526027     3.452584  1.0  4.0    7.00   10.00
next_day            365.0    15.720548     8.808321  1.0  8.0   16.00   23.00

                        max
current_month          12.000
current_day            31.000
purchases             118.000
unique_invoices         2.500
unique_streams         78.000
total_views          1430.000
current_revenue     16814.275
next_month_revenue  17730.135
next_month             12.000
next_day               31.000
current_month : 12
current_day : 31
purchases : 18
unique_invoices : 5
unique_streams : 17
total_views : 19
current_revenue : 19
next_month_revenue : 32
next_month : 12
next_day : 31

There is no column that includes a unique category of data
```

8 - Dataframe df_ts_portugal data:

FIRST 10 ROWS OF THE DATA

	current_month	current_day	purchases	unique_invoices	unique_streams
0	1	1	0.0	0.0	0.0
1	1	2	2.0	0.5	2.0
2	1	3	0.0	0.0	0.0
3	1	4	0.0	0.0	0.0
4	1	5	0.0	0.0	0.0
5	1	6	26.0	1.0	25.5
6	1	7	0.0	0.0	0.0
7	1	8	0.0	0.0	0.0
8	1	9	0.0	0.0	0.0
9	1	10	0.0	0.0	0.0

	total_views	current_revenue	next_month_revenue	next_month	next_day
0	0.0	0.000	319.875	2	1
1	1.0	6.700	319.875	2	2
2	0.0	0.000	313.175	2	3
3	0.0	0.000	313.175	2	4
4	0.0	0.000	313.175	2	5
5	235.5	87.405	313.175	2	6
6	0.0	0.000	240.830	2	7
7	0.0	0.000	240.830	2	8
8	0.0	0.000	240.830	2	9
9	0.0	0.000	240.830	2	10

LAST 10 ROWS OF THE DATA

	current_month	current_day	purchases	unique_invoices	unique_streams
355	12	22	0.0	0.0	0.0
356	12	23	0.0	0.0	0.0
357	12	24	0.0	0.0	0.0
358	12	25	0.0	0.0	0.0
359	12	26	0.0	0.0	0.0
360	12	27	0.0	0.0	0.0
361	12	28	0.0	0.0	0.0
362	12	29	0.0	0.0	0.0
363	12	30	0.0	0.0	0.0
364	12	31	0.0	0.0	0.0

	total_views	current_revenue	next_month_revenue	next_month	next_day
355	0.0	0.0	151.195	1	22
356	0.0	0.0	278.090	1	23
357	0.0	0.0	278.090	1	24
358	0.0	0.0	278.090	1	25
359	0.0	0.0	278.090	1	26
360	0.0	0.0	278.090	1	27
361	0.0	0.0	278.090	1	28
362	0.0	0.0	319.875	1	29
363	0.0	0.0	319.875	1	30
364	0.0	0.0	319.875	1	31

INFO ABOUT THE DATA

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
```

```
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT NUMERICAL COLUMNS
                    count        mean         std  min     25%      50%  \
current_month       365.0    6.526027    3.452584  1.0    4.00    7.000
current_day         365.0   15.720548    8.808321  1.0    8.00   16.000
purchases           365.0    2.787671    9.151697  0.0    0.00    0.000
unique_invoices     365.0    0.150685    0.405025  0.0    0.00    0.000
unique_streams      365.0    2.747945    9.025652  0.0    0.00    0.000
total_views         365.0   21.617808   77.164622  0.0    0.00    0.000
current_revenue     365.0   19.461904  169.093526  0.0    0.00    0.000
next_month_revenue  365.0  622.780932  949.125436  0.0  240.83  325.135
next_month          365.0    6.526027    3.452584  1.0    4.00    7.000
next_day            365.0   15.720548    8.808321  1.0    8.00   16.000
```

```
                      75%       max
current_month       10.00    12.000
current_day         23.00    31.000
purchases            0.00    80.000
unique_invoices      0.00     4.000
unique_streams       0.00    78.000
total_views          0.00   731.000
current_revenue      0.00  3130.155
next_month_revenue  500.04  3865.405
next_month          10.00    12.000
next_day            23.00    31.000
current_month : 12
current_day : 31
purchases : 43
unique_invoices : 6
unique_streams : 42
total_views : 57
current_revenue : 61
next_month_revenue : 117
next_month : 12
next_day : 31

There is no column that includes a unique category of data


9 - Dataframe df_ts_singapore data:

FIRST 10 ROWS OF THE DATA
   current_month  current_day  purchases  unique_invoices  unique_streams  \
0              1            1        0.0              0.0             0.0
1              1            2        0.0              0.0             0.0
2              1            3        0.0              0.0             0.0
3              1            4        0.0              0.0             0.0
4              1            5        0.0              0.0             0.0
5              1            6        0.0              0.0             0.0
6              1            7        0.0              0.0             0.0
7              1            8        0.0              0.0             0.0
8              1            9        0.0              0.0             0.0
9              1           10        0.0              0.0             0.0

   total_views  current_revenue  next_month_revenue  next_month  next_day
0          0.0              0.0              263.34           2         1
1          0.0              0.0              263.34           2         2
2          0.0              0.0              263.34           2         3
3          0.0              0.0              263.34           2         4
4          0.0              0.0              263.34           2         5
5          0.0              0.0              263.34           2         6
6          0.0              0.0              263.34           2         7
7          0.0              0.0              263.34           2         8
8          0.0              0.0              263.34           2         9
9          0.0              0.0              263.34           2        10

LAST 10 ROWS OF THE DATA
     current_month  current_day  purchases  unique_invoices  unique_streams  \
355             12           22        0.0              0.0             0.0
356             12           23        0.0              0.0             0.0
```

```
357             12           24        0.0              0.0             0.0
358             12           25        0.0              0.0             0.0
359             12           26        0.0              0.0             0.0
360             12           27        0.0              0.0             0.0
361             12           28        0.0              0.0             0.0
362             12           29        0.0              0.0             0.0
363             12           30        0.0              0.0             0.0
364             12           31        0.0              0.0             0.0

     total_views  current_revenue  next_month_revenue  next_month  next_day
355          0.0              0.0              263.34           1        22
356          0.0              0.0              263.34           1        23
357          0.0              0.0              263.34           1        24
358          0.0              0.0              263.34           1        25
359          0.0              0.0              263.34           1        26
360          0.0              0.0              263.34           1        27
361          0.0              0.0              263.34           1        28
362          0.0              0.0              263.34           1        29
363          0.0              0.0              263.34           1        30
364          0.0              0.0              263.34           1        31

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
```

```
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT NUMERICAL COLUMNS
                    count        mean          std  min  25%   50%     75%  \
current_month       365.0    6.526027     3.452584  1.0  4.0   7.0   10.00
current_day         365.0   15.720548     8.808321  1.0  8.0  16.0   23.00
purchases           365.0    0.509589     5.403300  0.0  0.0   0.0    0.00
unique_invoices     365.0    0.019178     0.198661  0.0  0.0   0.0    0.00
unique_streams      365.0    0.494521     5.357287  0.0  0.0   0.0    0.00
total_views         365.0    4.928767    53.956832  0.0  0.0   0.0    0.00
current_revenue     365.0   18.395274   319.264197  0.0  0.0   0.0    0.00
next_month_revenue  365.0  588.648767  1736.312516  0.0  0.0   0.0  219.75
next_month          365.0    6.526027     3.452584  1.0  4.0   7.0   10.00
next_day            365.0   15.720548     8.808321  1.0  8.0  16.0   23.00

                        max
current_month         12.000
current_day           31.000
purchases             81.000
unique_invoices        3.000
unique_streams        80.000
total_views          811.000
current_revenue     6090.840
next_month_revenue  6231.185
next_month            12.000
next_day              31.000
current_month : 12
current_day : 31
purchases : 6
unique_invoices : 5
unique_streams : 6
total_views : 6
current_revenue : 6
next_month_revenue : 8
next_month : 12
next_day : 31

There is no column that includes a unique category of data


10 - Dataframe df_ts_spain data:

FIRST 10 ROWS OF THE DATA
   current_month  current_day  purchases  unique_invoices  unique_streams  \
0              1            1        0.0              0.0             0.0
1              1            2        0.0              0.0             0.0
2              1            3        0.5              0.5             0.5
```

```
3              1           4       49.0             0.5           49.0
4              1           5        0.0             0.0            0.0
5              1           6        0.0             0.0            0.0
6              1           7       60.5             1.0           60.5
7              1           8       18.5             2.0           18.0
8              1           9        0.0             0.0            0.0
9              1          10        2.0             2.0            1.0

   total_views  current_revenue  next_month_revenue  next_month  next_day
0          0.0            0.000            4030.310           2         1
1          0.0            0.000            4030.310           2         2
2          1.5          563.000            4030.310           2         3
3        268.0          143.320            3467.310           2         4
4          0.0            0.000            3323.990           2         5
5          0.0            0.000            3415.955           2         6
6        415.0          256.070            3415.955           2         7
7        185.0          771.165            3159.885           2         8
8          0.0            0.000            2388.720           2         9
9          3.5          705.495            2388.720           2        10

LAST 10 ROWS OF THE DATA
     current_month  current_day  purchases  unique_invoices  unique_streams  \
355             12           22        0.0              0.0             0.0
356             12           23        0.0              0.0             0.0
357             12           24        0.0              0.0             0.0
358             12           25        0.0              0.0             0.0
359             12           26        0.0              0.0             0.0
360             12           27        0.0              0.0             0.0
361             12           28        0.0              0.0             0.0
362             12           29        0.0              0.0             0.0
363             12           30        0.0              0.0             0.0
364             12           31        0.0              0.0             0.0

     total_views  current_revenue  next_month_revenue  next_month  next_day
355          0.0              0.0            2609.815           1        22
356          0.0              0.0            2736.815           1        23
357          0.0              0.0            3848.635           1        24
358          0.0              0.0            3848.635           1        25
359          0.0              0.0            3863.260           1        26
360          0.0              0.0            3863.260           1        27
361          0.0              0.0            3863.260           1        28
362          0.0              0.0            3946.525           1        29
363          0.0              0.0            3946.525           1        30
364          0.0              0.0            4030.310           1        31

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
```

```
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

MISSING DATA
There is no missing data

DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT NUMERICAL COLUMNS
                    count        mean         std   min      25%     50%  \
current_month       365.0    6.526027    3.452584   1.0    4.000    7.00
current_day         365.0   15.720548    8.808321   1.0    8.000   16.00
purchases           365.0    3.976712   11.641810   0.0    0.000    0.00
unique_invoices     365.0    0.216438    0.447525   0.0    0.000    0.00
unique_streams      365.0    3.750685   10.546706   0.0    0.000    0.00
total_views         365.0   30.764384   93.238804   0.0    0.000    0.00
current_revenue     365.0   23.444849   92.881515   0.0    0.000    0.00
next_month_revenue  365.0  750.235178  859.187926  16.8  301.325  502.29
next_month          365.0    6.526027    3.452584   1.0    4.000    7.00
next_day            365.0   15.720548    8.808321   1.0    8.000   16.00

                     75%      max
current_month      10.00    12.00
current_day        23.00    31.00
purchases           0.00   103.50
unique_invoices     0.00     2.00
unique_streams      0.00    74.50
total_views         0.00   964.50
current_revenue     0.00  1111.82
```

```
next_month_revenue  626.74  4030.31
next_month           10.00    12.00
next_day             23.00    31.00
current_month : 12
current_day : 31
purchases : 48
unique_invoices : 5
unique_streams : 48
total_views : 79
current_revenue : 88
next_month_revenue : 156
next_month : 12
next_day : 31

There is no column that includes a unique category of data


11 - Dataframe df_ts_united_kingdom data:

FIRST 10 ROWS OF THE DATA
   current_month  current_day  purchases  unique_invoices  unique_streams  \
0              1            1      908.5             61.5           674.0
1              1            2     1277.5             55.5           804.0
2              1            3     1008.5             41.5           617.5
3              1            4     1136.5             64.5           744.0
4              1            5      263.0             20.5           190.0
5              1            6      509.0             23.5           306.5
6              1            7     1229.5             44.5           756.5
7              1            8     1979.5             69.5           992.5
8              1            9     1338.5             57.0           885.0
9              1           10     1451.0             50.0           905.0

   total_views  current_revenue  next_month_revenue  next_month  next_day
0       4224.0         3054.915          168776.787           2         1
1       6649.5        20895.180          167800.752           2         2
2       5639.0         3709.820          147300.177           2         3
3       6716.0         3778.740          146777.642           2         4
4       1892.5          934.760          148002.447           2         5
5       2591.5         1574.265          150454.892           2         6
6       6074.0         4530.165          152522.767           2         7
7       7476.5         8829.985          151858.897           2         8
8       5746.0         5201.190          144420.842           2         9
9       5586.0         6530.655          140133.507           2        10

LAST 10 ROWS OF THE DATA
     current_month  current_day  purchases  unique_invoices  unique_streams  \
355             12           22        0.0              0.0             0.0
356             12           23        0.0              0.0             0.0
357             12           24        0.0              0.0             0.0
358             12           25        0.0              0.0             0.0
359             12           26        0.0              0.0             0.0
360             12           27        0.0              0.0             0.0
361             12           28        0.0              0.0             0.0
362             12           29        0.0              0.0             0.0
363             12           30        0.0              0.0             0.0
364             12           31        0.0              0.0             0.0
```

```
     total_views  current_revenue  next_month_revenue  next_month  next_day
355          0.0              0.0         115011.4205           1        22
356          0.0              0.0         123458.7855           1        23
357          0.0              0.0         128804.7105           1        24
358          0.0              0.0         132304.2905           1        25
359          0.0              0.0         144266.3360           1        26
360          0.0              0.0         145408.2760           1        27
361          0.0              0.0         149842.3560           1        28
362          0.0              0.0         154971.1015           1        29
363          0.0              0.0         159256.1515           1        30
364          0.0              0.0         163649.0620           1        31

INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

BASIC INFO ABOUT THE DATA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   current_month       365 non-null    int32
 1   current_day         365 non-null    int32
 2   purchases           365 non-null    float64
 3   unique_invoices     365 non-null    float64
 4   unique_streams      365 non-null    float64
 5   total_views         365 non-null    float64
 6   current_revenue     365 non-null    float64
 7   next_month_revenue  365 non-null    float64
 8   next_month          365 non-null    int64
 9   next_day            365 non-null    int64
dtypes: float64(6), int32(2), int64(2)
memory usage: 25.8 KB
None

MISSING DATA
There is no missing data
```

```
DUPLICATED DATA ROWS
There is no duplicated rows

DESCRIPTIVE STATISTICS ABOUT NUMERICAL COLUMNS
                     count           mean           std       min          25%  \
current_month        365.0       6.526027      3.452584      1.00       4.0000
current_day          365.0      15.720548      8.808321      1.00       8.0000
purchases            365.0    1150.898630    666.326010      0.00     682.5000
unique_invoices      365.0      63.172603     36.841457      0.00      39.5000
unique_streams       365.0     636.382192    315.825621      0.00     419.0000
total_views          365.0    5868.190411   3459.484531      0.00    3250.5000
current_revenue      365.0    5306.972075   7389.310806 -39116.64    2412.0400
next_month_revenue   365.0  169823.106411  55246.220501  80580.90  132616.3575
next_month           365.0       6.526027      3.452584      1.00       4.0000
next_day             365.0      15.720548      8.808321      1.00       8.0000

                          50%         75%         max
current_month           7.000       10.00      12.000
current_day            16.000       23.00      31.000
purchases            1200.000     1537.00    3371.000
unique_invoices        65.000       86.50     194.000
unique_streams        698.000      841.00    1316.000
total_views          6179.500     7837.00   17336.000
current_revenue      4760.260     6467.66   68821.780
next_month_revenue 158874.736   195567.84  329592.561
next_month              7.000       10.00      12.000
next_day               16.000       23.00      31.000
current_month : 12
current_day : 31
purchases : 317
unique_invoices : 177
unique_streams : 296
total_views : 330
current_revenue : 331
next_month_revenue : 364
next_month : 12
next_day : 31

There is no column that includes a unique category of data
```

2.2.-Split data into predictors (X) and target (y)

In [21]:

```python
# Split the data into features and target
# X = X[df_name] and y = [df_name]
X = {}
y = {}
for i in range (0,size):
    df_name = list_ts_df[i]
    X[df_name], y[df_name] = model.split_into_predictors_targets(library_ts_df_mont
    print(i+1, '- Dataframe', df_name , 'was successfully splitted')
    print("SHAPE OF THE DATA: ")
    print("Number of (rows, column) X : " + str((X[df_name]).shape))
```

```
    print("Number of (row, columns) y : " + str((y[df_name]).shape))
    print('\n')
```

```
1 - Dataframe df_ts_all was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 2)
Number of (row, columns) y : (365, 6)


2 - Dataframe df_ts_eire was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 2)
Number of (row, columns) y : (365, 6)


3 - Dataframe df_ts_france was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 2)
Number of (row, columns) y : (365, 6)


4 - Dataframe df_ts_germany was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 2)
Number of (row, columns) y : (365, 6)


5 - Dataframe df_ts_hong_kong was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 2)
Number of (row, columns) y : (365, 6)


6 - Dataframe df_ts_netherlands was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 2)
Number of (row, columns) y : (365, 6)


7 - Dataframe df_ts_norway was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 2)
Number of (row, columns) y : (365, 6)


8 - Dataframe df_ts_portugal was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 2)
Number of (row, columns) y : (365, 6)


9 - Dataframe df_ts_singapore was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 2)
Number of (row, columns) y : (365, 6)


10 - Dataframe df_ts_spain was successfully splitted
SHAPE OF THE DATA:
```

```
Number of (rows, column) X : (365, 2)
Number of (row, columns) y : (365, 6)


11 - Dataframe df_ts_united_kingdom was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 2)
Number of (row, columns) y : (365, 6)
```

2.2.-Split data into training (X_train, y_train) and test (X_test, y_test) sets

In [22]:

```
# Randomly split the data into 80% for training and 20% for test sets
# X_train=X_train[df_name] , y_train=y_train[df_name], X_test=X_test[df_name] and y
X_train, X_test , y_train , y_test = {}, {}, {}, {}
for i in range (0,size):
    df_name = list_ts_df[i]
    X_train[df_name], X_test[df_name], y_train[df_name], y_test[df_name] = model.sp
    print(i+1, '- Dataframe', df_name , 'was successfully splitted')
    print("SHAPE OF THE DATA: ")
    print("Number of (rows, column) Xtrain : " + str((X_train[df_name]).shape))
    print("Number of (row, columns) Xtest : " + str((X_test[df_name]).shape))
    print("Number of (rows, column) ytrain : " + str((y_train[df_name]).shape))
    print("Number of (row, columns) ytest : " + str((y_test[df_name]).shape))
    print('\n')
```

```
1 - Dataframe df_ts_all was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 2)
Number of (row, columns) Xtest : (37, 2)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


2 - Dataframe df_ts_eire was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 2)
Number of (row, columns) Xtest : (37, 2)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


3 - Dataframe df_ts_france was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 2)
Number of (row, columns) Xtest : (37, 2)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


4 - Dataframe df_ts_germany was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 2)
Number of (row, columns) Xtest : (37, 2)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


5 - Dataframe df_ts_hong_kong was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 2)
Number of (row, columns) Xtest : (37, 2)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


6 - Dataframe df_ts_netherlands was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 2)
Number of (row, columns) Xtest : (37, 2)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


7 - Dataframe df_ts_norway was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 2)
Number of (row, columns) Xtest : (37, 2)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)
```

```
8 - Dataframe df_ts_portugal was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 2)
Number of (row, columns) Xtest : (37, 2)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


9 - Dataframe df_ts_singapore was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 2)
Number of (row, columns) Xtest : (37, 2)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


10 - Dataframe df_ts_spain was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 2)
Number of (row, columns) Xtest : (37, 2)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


11 - Dataframe df_ts_united_kingdom was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 2)
Number of (row, columns) Xtest : (37, 2)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)
```

In [23]:
```
#Sort in ascending to be able to plot appropiate time-series visualizationabs
for i in range (0,size):
    df_name = list_ts_df[i]
    X_train[df_name]=X_train[df_name].sort_index(ascending=True)
    X_test[df_name] =X_test [df_name].sort_index(ascending=True)
    y_train[df_name]=y_train[df_name].sort_index(ascending=True)
    y_test[df_name] =y_test [df_name].sort_index(ascending=True)
```

2.3.-Train Predict and Evaluate 3 models

In [24]:
```
# Train, predict, and evaluate each model
y_pred_lr , y_pred_rf , y_pred_xgb , lr_model , rf_model, xgb_model = {}, {}, {}, {
for i in range (0,size):
    df_name = list_ts_df[i]
    print(i+1, '- The model of Dataframe', df_name , 'was successfully trained, use
    y_pred_lr[df_name], y_pred_rf[df_name], y_pred_xgb[df_name], lr_model[df_name],
    print('\n')
```

1 - The model of Dataframe df_ts_all was successfully trained, used to predict and e
valuated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 15113.29
Mean Squared Error (MSE) : 1294475847.27
Root Mean-Square Error (RMSE) : 35978.82
Coefficient of Determination (R2) : -0.48

Random Forest Model:
Mean Absolute Error (MAE): 19545.20
Mean Squared Error (MSE) : 2341630784.42
Root Mean-Square Error (RMSE) : 48390.40
Coefficient of Determination (R2) : -1.05

XGBoost Model:
Mean Absolute Error (MAE): 19331.28
Mean Squared Error (MSE) : 2286804480.00
Root Mean-Square Error (RMSE) : 47820.54
Coefficient of Determination (R2) : -1.19

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	15113.293798	1.294476e+09	35978.824985	-0.483398
Random Forest	19545.198085	2.341631e+09	48390.399713	-1.049131
XGBoost	19331.283203	2.286804e+09	47820.544539	-1.186588

2 - The model of Dataframe df_ts_eire was successfully trained, used to predict and
evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 874.97
Mean Squared Error (MSE) : 4187745.55
Root Mean-Square Error (RMSE) : 2046.40
Coefficient of Determination (R2) : -2.39

Random Forest Model:
Mean Absolute Error (MAE): 379.47
Mean Squared Error (MSE) : 1051110.57
Root Mean-Square Error (RMSE) : 1025.24
Coefficient of Determination (R2) : -9.51

XGBoost Model:
Mean Absolute Error (MAE): 391.03
Mean Squared Error (MSE) : 1343493.62
Root Mean-Square Error (RMSE) : 1159.09
Coefficient of Determination (R2) : -19.48

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	874.971760	4.187746e+06	2046.398189	-2.393377
Random Forest	379.473528	1.051111e+06	1025.236837	-9.511772
XGBoost	391.033905	1.343494e+06	1159.091724	-19.480516

3 - The model of Dataframe df_ts_france was successfully trained, used to predict an
d evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 233.06
Mean Squared Error (MSE) : 240743.08
Root Mean-Square Error (RMSE) : 490.66
Coefficient of Determination (R2) : -4.09

Random Forest Model:
Mean Absolute Error (MAE): 115.73
Mean Squared Error (MSE) : 55572.87
Root Mean-Square Error (RMSE) : 235.74
Coefficient of Determination (R2) : -1.58

XGBoost Model:
Mean Absolute Error (MAE): 112.77
Mean Squared Error (MSE) : 57160.69
Root Mean-Square Error (RMSE) : 239.08
Coefficient of Determination (R2) : -1.93

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	233.058620	240743.076100	490.655761	-4.093454
Random Forest	115.725481	55572.872287	235.738992	-1.581300
XGBoost	112.774971	57160.687500	239.083014	-1.933189

4 - The model of Dataframe df_ts_germany was successfully trained, used to predict a
nd evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 203.24
Mean Squared Error (MSE) : 173161.74
Root Mean-Square Error (RMSE) : 416.13
Coefficient of Determination (R2) : -1.30

Random Forest Model:
Mean Absolute Error (MAE): 148.96
Mean Squared Error (MSE) : 83310.70
Root Mean-Square Error (RMSE) : 288.64
Coefficient of Determination (R2) : -0.91

XGBoost Model:
Mean Absolute Error (MAE): 151.73
Mean Squared Error (MSE) : 87349.57
Root Mean-Square Error (RMSE) : 295.55
Coefficient of Determination (R2) : -1.31

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	203.240907	173161.744737	416.127078	-1.303295
Random Forest	148.961496	83310.697833	288.635926	-0.913134
XGBoost	151.731705	87349.570312	295.549607	-1.307158

5 - The model of Dataframe df_ts_hong_kong was successfully trained, used to predict and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 27.16
Mean Squared Error (MSE) : 4142.49
Root Mean-Square Error (RMSE) : 64.36
Coefficient of Determination (R2) : -0.53

Random Forest Model:
Mean Absolute Error (MAE): 9.16
Mean Squared Error (MSE) : 1501.51
Root Mean-Square Error (RMSE) : 38.75
Coefficient of Determination (R2) : -0.07

XGBoost Model:
Mean Absolute Error (MAE): 8.31
Mean Squared Error (MSE) : 1392.82
Root Mean-Square Error (RMSE) : 37.32
Coefficient of Determination (R2) : -0.07

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	27.164393	4142.491214	64.362188	-0.531159
Random Forest	9.157770	1501.513603	38.749369	-0.071601
XGBoost	8.308639	1392.815918	37.320449	-0.070510

6 - The model of Dataframe df_ts_netherlands was successfully trained, used to predict and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 107.44
Mean Squared Error (MSE) : 45753.36
Root Mean-Square Error (RMSE) : 213.90
Coefficient of Determination (R2) : -6.47

Random Forest Model:
Mean Absolute Error (MAE): 63.86
Mean Squared Error (MSE) : 20348.58
Root Mean-Square Error (RMSE) : 142.65
Coefficient of Determination (R2) : -4.05

XGBoost Model:
Mean Absolute Error (MAE): 63.22
Mean Squared Error (MSE) : 22380.40
Root Mean-Square Error (RMSE) : 149.60
Coefficient of Determination (R2) : -6.03

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	107.443806	45753.355157	213.900339	-6.471344
Random Forest	63.861771	20348.584071	142.648463	-4.050297
XGBoost	63.215790	22380.400391	149.600803	-6.025217

7 - The model of Dataframe df_ts_norway was successfully trained, used to predict and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 315.73
Mean Squared Error (MSE) : 533391.53
Root Mean-Square Error (RMSE) : 730.34
Coefficient of Determination (R2) : -48.77

Random Forest Model:
Mean Absolute Error (MAE): 70.33
Mean Squared Error (MSE) : 30302.45
Root Mean-Square Error (RMSE) : 174.08
Coefficient of Determination (R2) : -1.82

XGBoost Model:
Mean Absolute Error (MAE): 71.70
Mean Squared Error (MSE) : 32855.49
Root Mean-Square Error (RMSE) : 181.26
Coefficient of Determination (R2) : -2.22

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	315.733440	533391.534969	730.336590	-48.774811
Random Forest	70.330258	30302.451395	174.075993	-1.815316
XGBoost	71.697090	32855.492188	181.260840	-2.217873

8 - The model of Dataframe df_ts_portugal was successfully trained, used to predict and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 63.78
Mean Squared Error (MSE) : 17828.10
Root Mean-Square Error (RMSE) : 133.52
Coefficient of Determination (R2) : -1.78

Random Forest Model:
Mean Absolute Error (MAE): 70.95
Mean Squared Error (MSE) : 24966.99
Root Mean-Square Error (RMSE) : 158.01
Coefficient of Determination (R2) : -3.56

XGBoost Model:
Mean Absolute Error (MAE): 70.70
Mean Squared Error (MSE) : 25434.49
Root Mean-Square Error (RMSE) : 159.48
Coefficient of Determination (R2) : -4.68

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	63.775048	17828.098839	133.521904	-1.779286
Random Forest	70.950056	24966.992313	158.009469	-3.562833
XGBoost	70.697411	25434.486328	159.481931	-4.682113

9 - The model of Dataframe df_ts_singapore was successfully trained, used to predict and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 93.97
Mean Squared Error (MSE) : 50922.68
Root Mean-Square Error (RMSE) : 225.66
Coefficient of Determination (R2) : -3.54

Random Forest Model:
Mean Absolute Error (MAE): 12.42
Mean Squared Error (MSE) : 3137.76
Root Mean-Square Error (RMSE) : 56.02
Coefficient of Determination (R2) : -0.06

XGBoost Model:
Mean Absolute Error (MAE): 11.86
Mean Squared Error (MSE) : 3123.67
Root Mean-Square Error (RMSE) : 55.89
Coefficient of Determination (R2) : -0.06

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	93.965340	50922.679793	225.660541	-3.540825
Random Forest	12.421250	3137.763548	56.015744	-0.061870
XGBoost	11.862251	3123.671143	55.889813	-0.061720

10 - The model of Dataframe df_ts_spain was successfully trained, used to predict and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 368.15
Mean Squared Error (MSE) : 1027385.87
Root Mean-Square Error (RMSE) : 1013.60
Coefficient of Determination (R2) : -0.56

Random Forest Model:
Mean Absolute Error (MAE): 327.27
Mean Squared Error (MSE) : 830994.22
Root Mean-Square Error (RMSE) : 911.59
Coefficient of Determination (R2) : -0.48

XGBoost Model:
Mean Absolute Error (MAE): 325.16
Mean Squared Error (MSE) : 828068.56
Root Mean-Square Error (RMSE) : 909.98
Coefficient of Determination (R2) : -0.51

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	368.145371	1.027386e+06	1013.600448	-0.556843
Random Forest	327.268168	8.309942e+05	911.588846	-0.476884
XGBoost	325.157379	8.280686e+05	909.982726	-0.510828

```
11 - The model of Dataframe df_ts_united_kingdom was successfully trained, used to p
redict and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 14233.09
Mean Squared Error (MSE) : 1143262133.08
Root Mean-Square Error (RMSE) : 33812.16
Coefficient of Determination (R2) : -0.37

Random Forest Model:
Mean Absolute Error (MAE): 19572.44
Mean Squared Error (MSE) : 2377811041.60
Root Mean-Square Error (RMSE) : 48762.80
Coefficient of Determination (R2) : -1.01

XGBoost Model:
Mean Absolute Error (MAE): 19615.31
Mean Squared Error (MSE) : 2362238976.00
Root Mean-Square Error (RMSE) : 48602.87
Coefficient of Determination (R2) : -1.13

Evaluation Summary :
                           MAE           MSE          RMSE        R2
Linear         14233.092209  1.143262e+09  33812.159545 -0.373074
Random Forest  19572.435455  2.377811e+09  48762.803873 -1.013690
XGBoost        19615.314453  2.362239e+09  48602.870039 -1.134821
```

2.4.-Plot test data observed and predicted with the 3 models

In [25]:
```
#Convert month and day to day of year to plot the test data observed and predicted
X_test_day_of_year = {}
for i in range (0,size):
    df_name = list_ts_df[i]
    print(i+1, '- Dataframe', df_name , 'data:')
    X_test_day_of_year[df_name] = model.to_day_of_year_df(X_test[df_name])
    print(X_test_day_of_year[df_name])
```

1 - Dataframe df_ts_all data:

	day_of_year	current_month	current_day
328	330	11	25
329	331	11	26
330	332	11	27
331	333	11	28
332	334	11	29
333	335	11	30
334	336	12	1
335	337	12	2
336	338	12	3
337	339	12	4
338	340	12	5
339	341	12	6
340	342	12	7
341	343	12	8
342	344	12	9
343	345	12	10
344	346	12	11
345	347	12	12
346	348	12	13
347	349	12	14
348	350	12	15
349	351	12	16
350	352	12	17
351	353	12	18
352	354	12	19
353	355	12	20
354	356	12	21
355	357	12	22
356	358	12	23
357	359	12	24
358	360	12	25
359	361	12	26
360	362	12	27
361	363	12	28
362	364	12	29
363	365	12	30
364	366	12	31

2 - Dataframe df_ts_eire data:

	day_of_year	current_month	current_day
328	330	11	25
329	331	11	26
330	332	11	27
331	333	11	28
332	334	11	29
333	335	11	30
334	336	12	1
335	337	12	2
336	338	12	3
337	339	12	4
338	340	12	5
339	341	12	6
340	342	12	7
341	343	12	8
342	344	12	9

```
343          345             12           10
344          346             12           11
345          347             12           12
346          348             12           13
347          349             12           14
348          350             12           15
349          351             12           16
350          352             12           17
351          353             12           18
352          354             12           19
353          355             12           20
354          356             12           21
355          357             12           22
356          358             12           23
357          359             12           24
358          360             12           25
359          361             12           26
360          362             12           27
361          363             12           28
362          364             12           29
363          365             12           30
364          366             12           31
3 - Dataframe df_ts_france data:
     day_of_year  current_month  current_day
328          330             11           25
329          331             11           26
330          332             11           27
331          333             11           28
332          334             11           29
333          335             11           30
334          336             12            1
335          337             12            2
336          338             12            3
337          339             12            4
338          340             12            5
339          341             12            6
340          342             12            7
341          343             12            8
342          344             12            9
343          345             12           10
344          346             12           11
345          347             12           12
346          348             12           13
347          349             12           14
348          350             12           15
349          351             12           16
350          352             12           17
351          353             12           18
352          354             12           19
353          355             12           20
354          356             12           21
355          357             12           22
356          358             12           23
357          359             12           24
358          360             12           25
359          361             12           26
```

```
360          362             12           27
361          363             12           28
362          364             12           29
363          365             12           30
364          366             12           31
4 - Dataframe df_ts_germany data:
     day_of_year  current_month  current_day
328          330             11           25
329          331             11           26
330          332             11           27
331          333             11           28
332          334             11           29
333          335             11           30
334          336             12            1
335          337             12            2
336          338             12            3
337          339             12            4
338          340             12            5
339          341             12            6
340          342             12            7
341          343             12            8
342          344             12            9
343          345             12           10
344          346             12           11
345          347             12           12
346          348             12           13
347          349             12           14
348          350             12           15
349          351             12           16
350          352             12           17
351          353             12           18
352          354             12           19
353          355             12           20
354          356             12           21
355          357             12           22
356          358             12           23
357          359             12           24
358          360             12           25
359          361             12           26
360          362             12           27
361          363             12           28
362          364             12           29
363          365             12           30
364          366             12           31
5 - Dataframe df_ts_hong_kong data:
     day_of_year  current_month  current_day
328          330             11           25
329          331             11           26
330          332             11           27
331          333             11           28
332          334             11           29
333          335             11           30
334          336             12            1
335          337             12            2
336          338             12            3
337          339             12            4
```

```
338          340             12             5
339          341             12             6
340          342             12             7
341          343             12             8
342          344             12             9
343          345             12            10
344          346             12            11
345          347             12            12
346          348             12            13
347          349             12            14
348          350             12            15
349          351             12            16
350          352             12            17
351          353             12            18
352          354             12            19
353          355             12            20
354          356             12            21
355          357             12            22
356          358             12            23
357          359             12            24
358          360             12            25
359          361             12            26
360          362             12            27
361          363             12            28
362          364             12            29
363          365             12            30
364          366             12            31
6 - Dataframe df_ts_netherlands data:
     day_of_year  current_month  current_day
328          330             11            25
329          331             11            26
330          332             11            27
331          333             11            28
332          334             11            29
333          335             11            30
334          336             12             1
335          337             12             2
336          338             12             3
337          339             12             4
338          340             12             5
339          341             12             6
340          342             12             7
341          343             12             8
342          344             12             9
343          345             12            10
344          346             12            11
345          347             12            12
346          348             12            13
347          349             12            14
348          350             12            15
349          351             12            16
350          352             12            17
351          353             12            18
352          354             12            19
353          355             12            20
354          356             12            21
```

```
355          357             12           22
356          358             12           23
357          359             12           24
358          360             12           25
359          361             12           26
360          362             12           27
361          363             12           28
362          364             12           29
363          365             12           30
364          366             12           31
7 - Dataframe df_ts_norway data:
     day_of_year  current_month  current_day
328          330             11           25
329          331             11           26
330          332             11           27
331          333             11           28
332          334             11           29
333          335             11           30
334          336             12            1
335          337             12            2
336          338             12            3
337          339             12            4
338          340             12            5
339          341             12            6
340          342             12            7
341          343             12            8
342          344             12            9
343          345             12           10
344          346             12           11
345          347             12           12
346          348             12           13
347          349             12           14
348          350             12           15
349          351             12           16
350          352             12           17
351          353             12           18
352          354             12           19
353          355             12           20
354          356             12           21
355          357             12           22
356          358             12           23
357          359             12           24
358          360             12           25
359          361             12           26
360          362             12           27
361          363             12           28
362          364             12           29
363          365             12           30
364          366             12           31
8 - Dataframe df_ts_portugal data:
     day_of_year  current_month  current_day
328          330             11           25
329          331             11           26
330          332             11           27
331          333             11           28
332          334             11           29
```

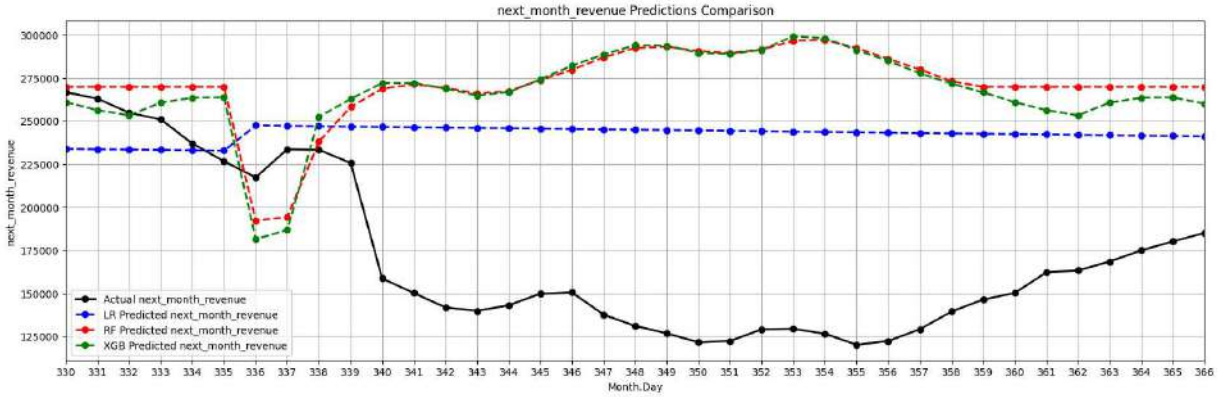
```
333          335             11           30
334          336             12            1
335          337             12            2
336          338             12            3
337          339             12            4
338          340             12            5
339          341             12            6
340          342             12            7
341          343             12            8
342          344             12            9
343          345             12           10
344          346             12           11
345          347             12           12
346          348             12           13
347          349             12           14
348          350             12           15
349          351             12           16
350          352             12           17
351          353             12           18
352          354             12           19
353          355             12           20
354          356             12           21
355          357             12           22
356          358             12           23
357          359             12           24
358          360             12           25
359          361             12           26
360          362             12           27
361          363             12           28
362          364             12           29
363          365             12           30
364          366             12           31
9 - Dataframe df_ts_singapore data:
     day_of_year  current_month  current_day
328          330             11           25
329          331             11           26
330          332             11           27
331          333             11           28
332          334             11           29
333          335             11           30
334          336             12            1
335          337             12            2
336          338             12            3
337          339             12            4
338          340             12            5
339          341             12            6
340          342             12            7
341          343             12            8
342          344             12            9
343          345             12           10
344          346             12           11
345          347             12           12
346          348             12           13
347          349             12           14
348          350             12           15
349          351             12           16
```

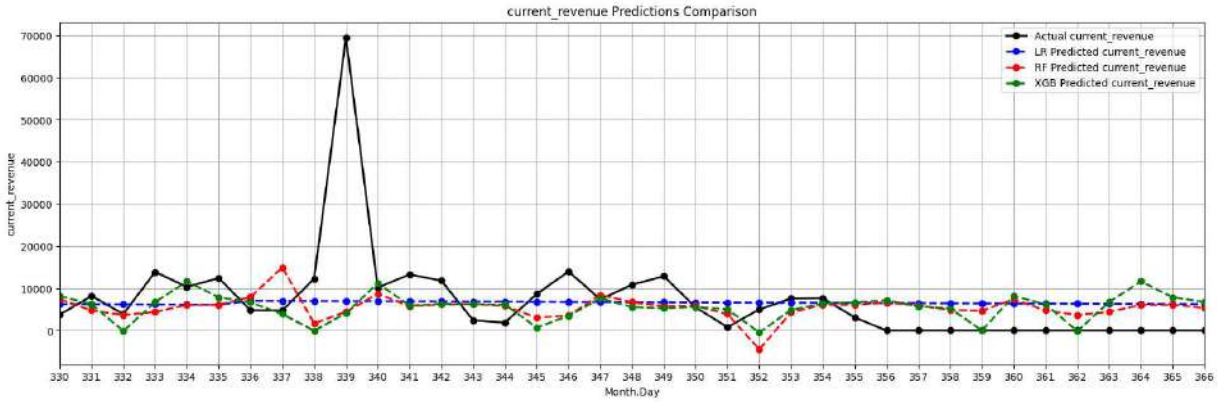
```
350          352             12           17
351          353             12           18
352          354             12           19
353          355             12           20
354          356             12           21
355          357             12           22
356          358             12           23
357          359             12           24
358          360             12           25
359          361             12           26
360          362             12           27
361          363             12           28
362          364             12           29
363          365             12           30
364          366             12           31
10 - Dataframe df_ts_spain data:
     day_of_year  current_month  current_day
328          330             11           25
329          331             11           26
330          332             11           27
331          333             11           28
332          334             11           29
333          335             11           30
334          336             12            1
335          337             12            2
336          338             12            3
337          339             12            4
338          340             12            5
339          341             12            6
340          342             12            7
341          343             12            8
342          344             12            9
343          345             12           10
344          346             12           11
345          347             12           12
346          348             12           13
347          349             12           14
348          350             12           15
349          351             12           16
350          352             12           17
351          353             12           18
352          354             12           19
353          355             12           20
354          356             12           21
355          357             12           22
356          358             12           23
357          359             12           24
358          360             12           25
359          361             12           26
360          362             12           27
361          363             12           28
362          364             12           29
363          365             12           30
364          366             12           31
11 - Dataframe df_ts_united_kingdom data:
     day_of_year  current_month  current_day
```

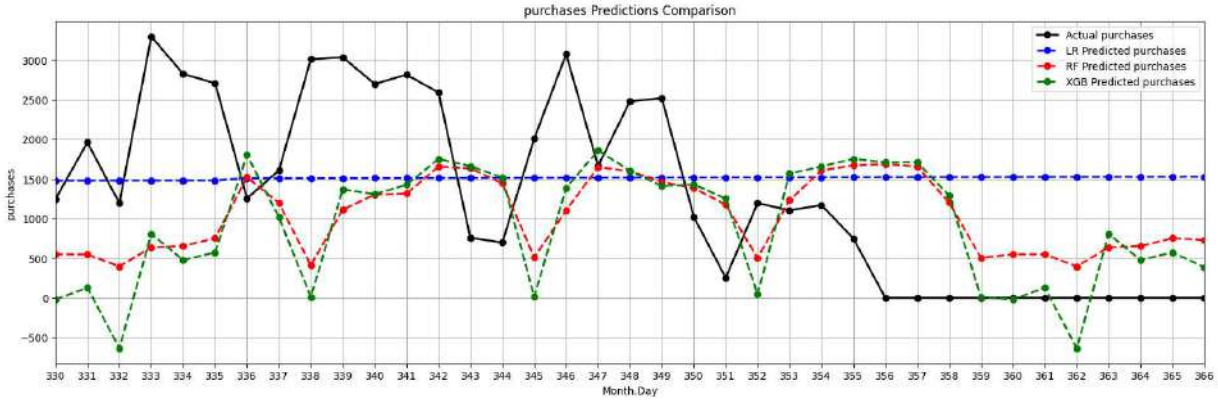
328	330	11	25
329	331	11	26
330	332	11	27
331	333	11	28
332	334	11	29
333	335	11	30
334	336	12	1
335	337	12	2
336	338	12	3
337	339	12	4
338	340	12	5
339	341	12	6
340	342	12	7
341	343	12	8
342	344	12	9
343	345	12	10
344	346	12	11
345	347	12	12
346	348	12	13
347	349	12	14
348	350	12	15
349	351	12	16
350	352	12	17
351	353	12	18
352	354	12	19
353	355	12	20
354	356	12	21
355	357	12	22
356	358	12	23
357	359	12	24
358	360	12	25
359	361	12	26
360	362	12	27
361	363	12	28
362	364	12	29
363	365	12	30
364	366	12	31

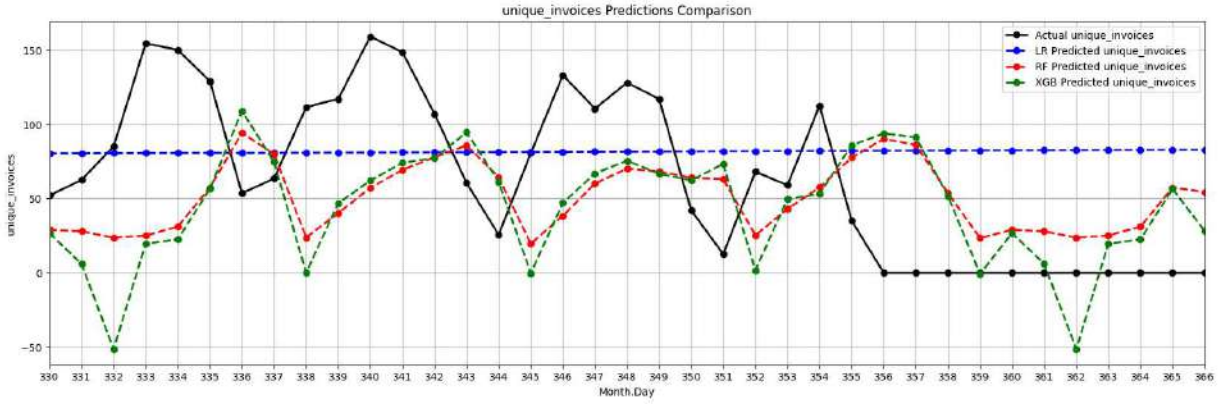
In [26]:
```
# Plot results for each target
for i in range (0,size):
    df_name = list_ts_df[i]
    print(i+1, '- Dataframe', df_name , 'data:')
    for target in y_test[df_name].columns:
        viz.plot_test_data(X_test_day_of_year[df_name]['day_of_year'], y_test[df_na
```

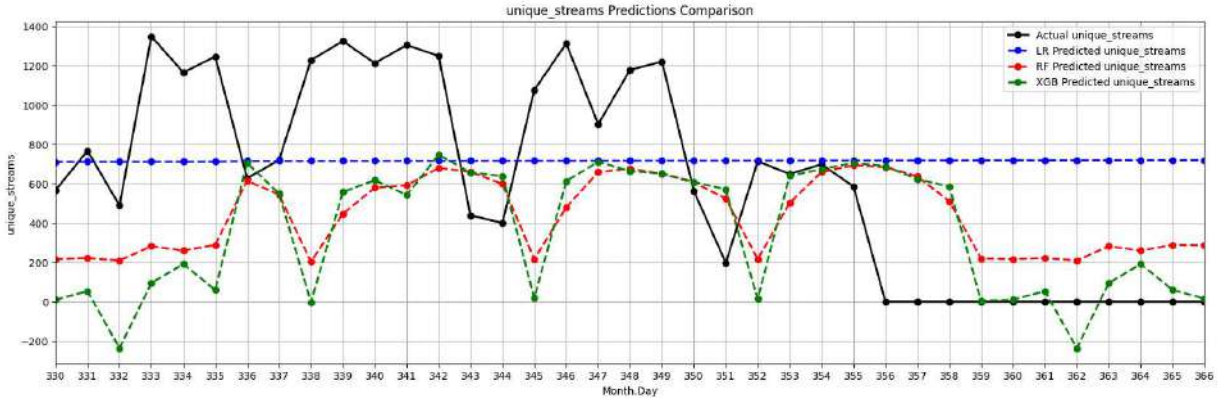
```
1 - Dataframe df_ts_all data:
```

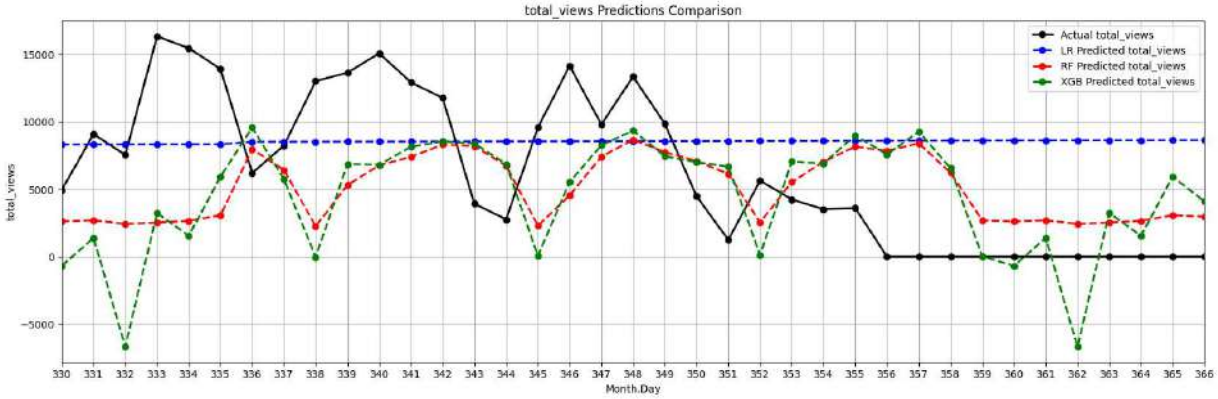

next_month_revenue Predictions Comparison
next_month_revenue
Month.Day
Actual next_month_revenue
LR Predicted next_month_revenue
RF Predicted next_month_revenue
XGB Predicted next_month_revenue


current_revenue Predictions Comparison
current_revenue
Month.Day
Actual current_revenue
LR Predicted current_revenue
RF Predicted current_revenue
XGB Predicted current_revenue

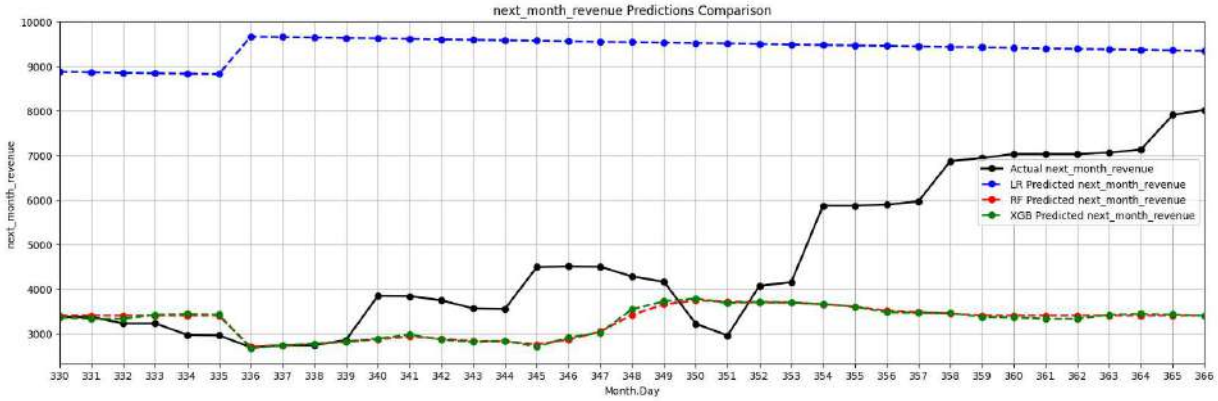

purchases Predictions Comparison
purchases
Month.Day
Actual purchases
LR Predicted purchases
RF Predicted purchases
XGB Predicted purchases

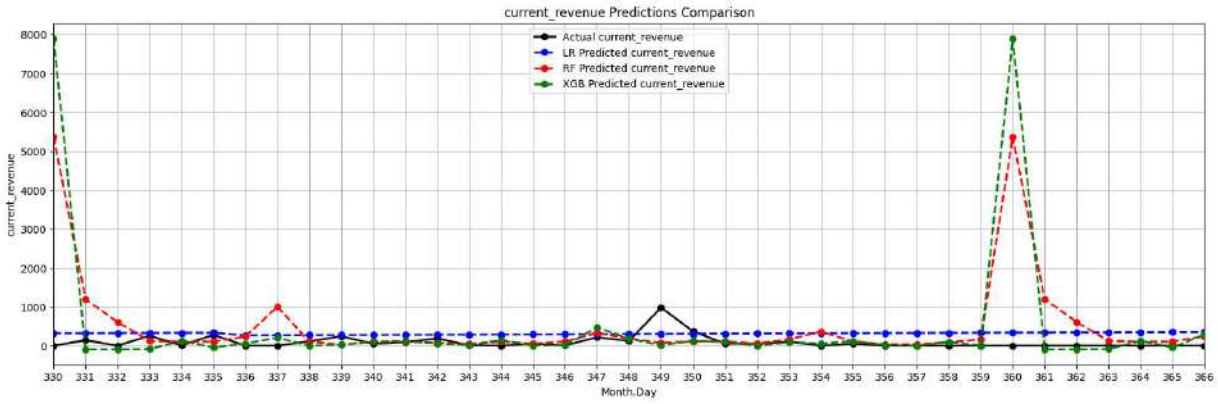

unique_invoices Predictions Comparison
unique_invoices
Month.Day
Actual unique_invoices
LR Predicted unique_invoices
RF Predicted unique_invoices
XGB Predicted unique_invoices

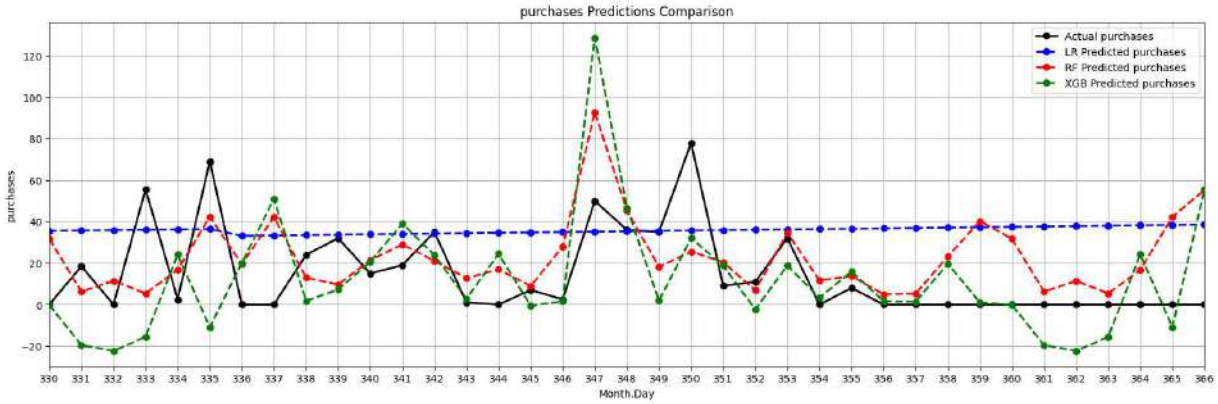


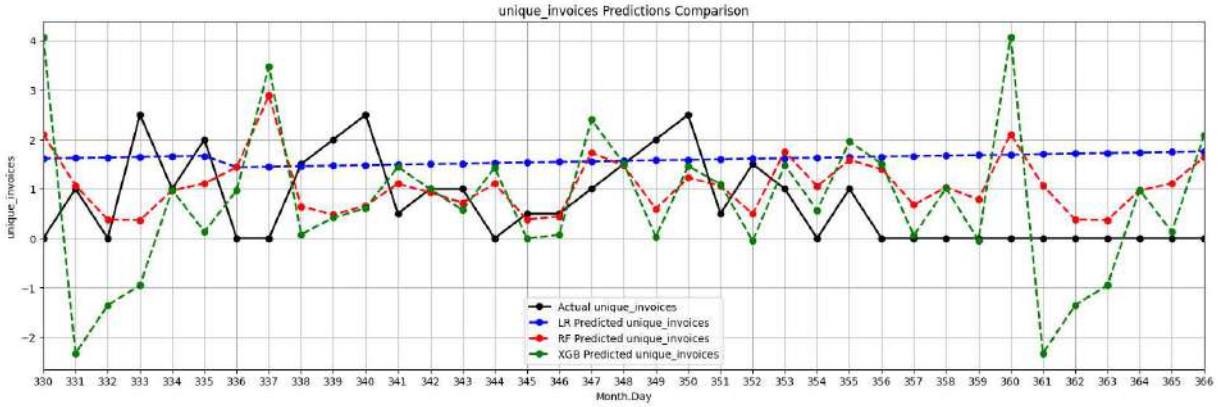


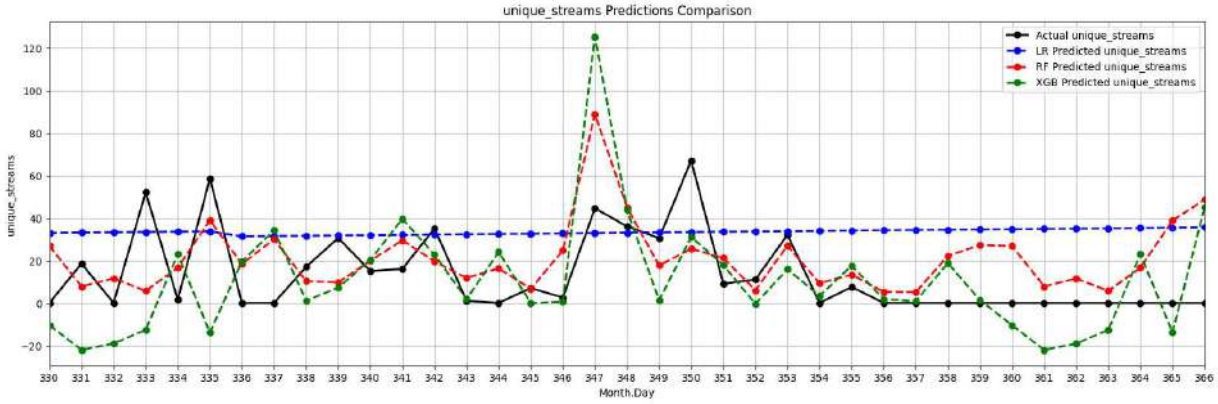
2 - Dataframe df_ts_eire data:

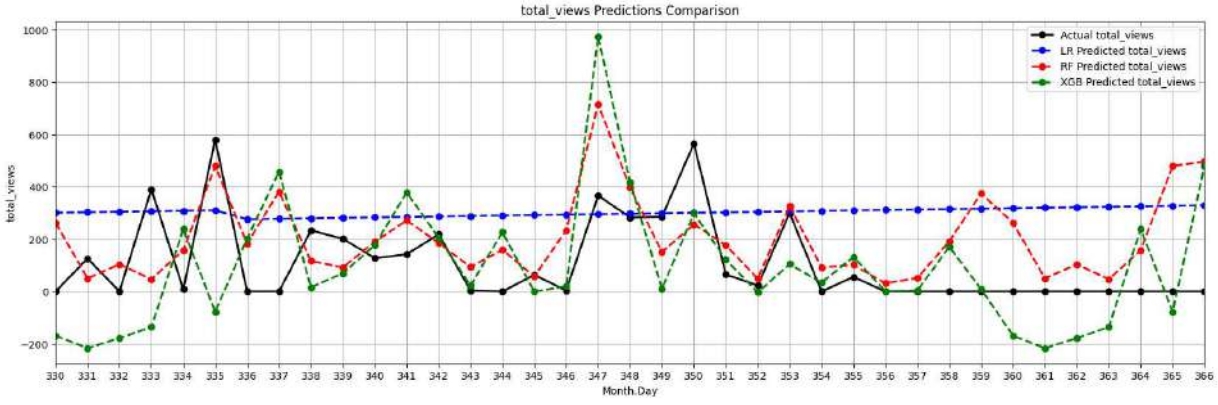




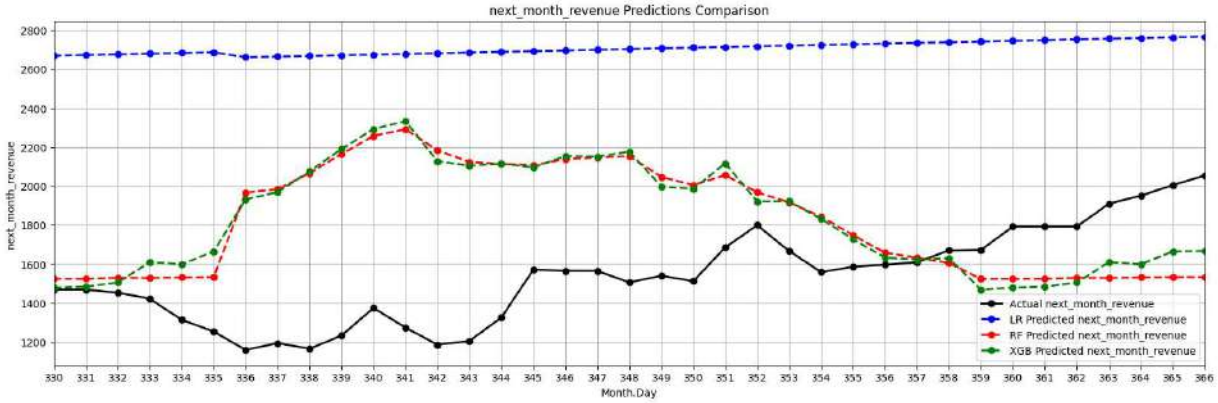


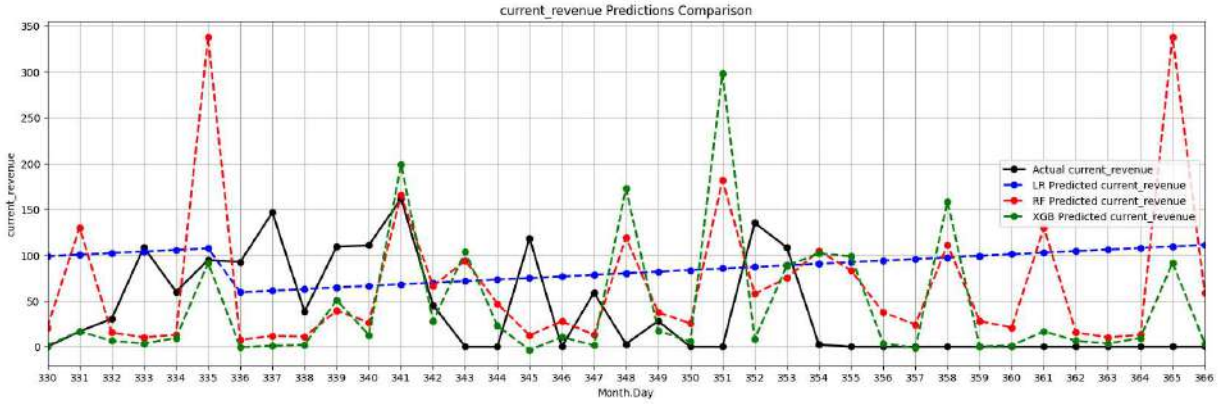


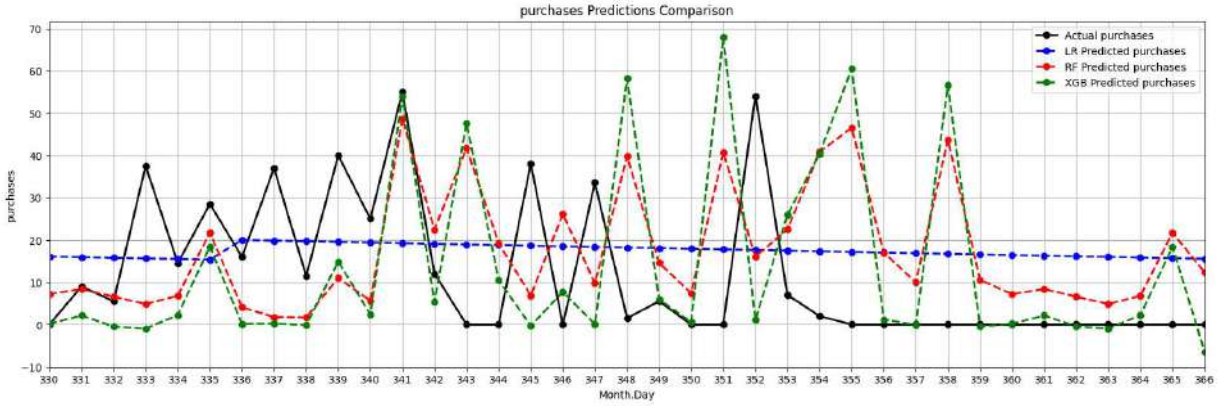


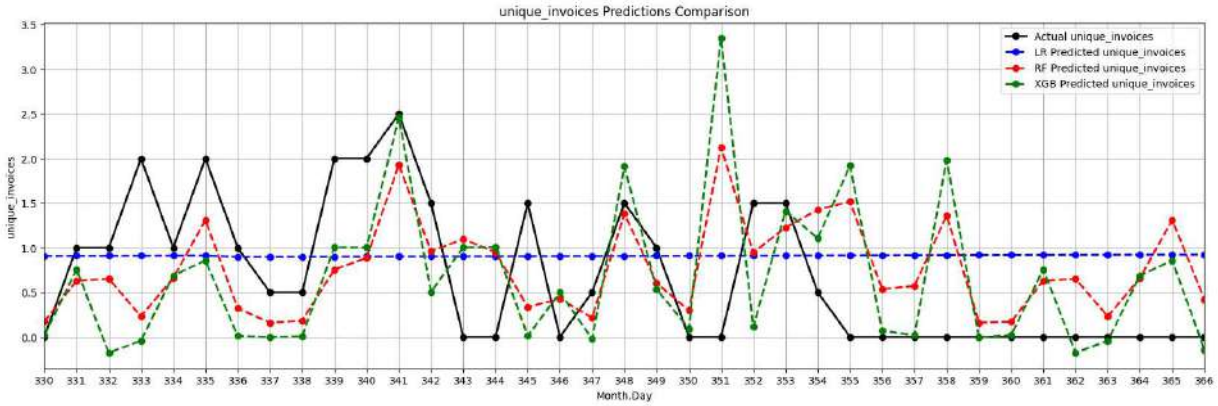


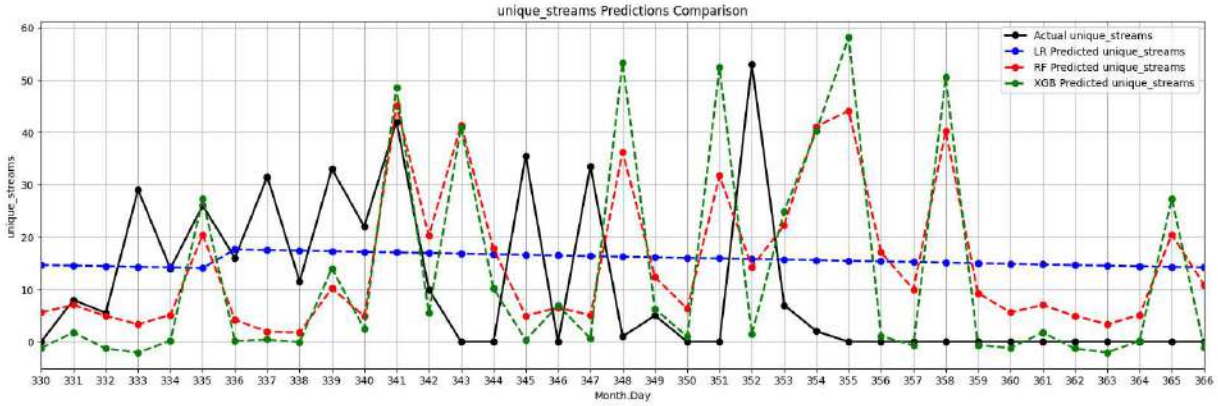
3 - Dataframe df_ts_france data:

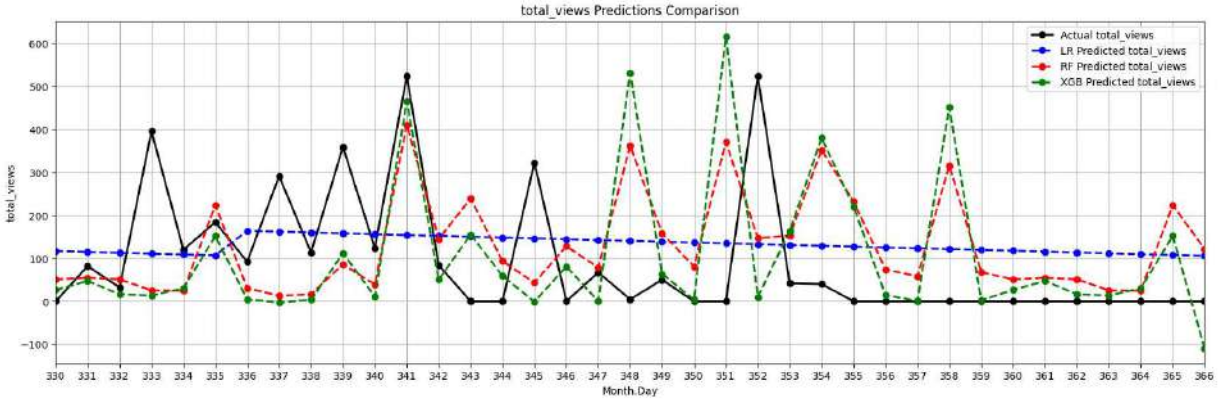

next_month_revenue Predictions Comparison
next_month_revenue
Month.Day
Actual next_month_revenue
LR Predicted next_month_revenue
RF Predicted next_month_revenue
XGB Predicted next_month_revenue


current_revenue Predictions Comparison
current_revenue
Month.Day
Actual current_revenue
LR Predicted current_revenue
RF Predicted current_revenue
XGB Predicted current_revenue

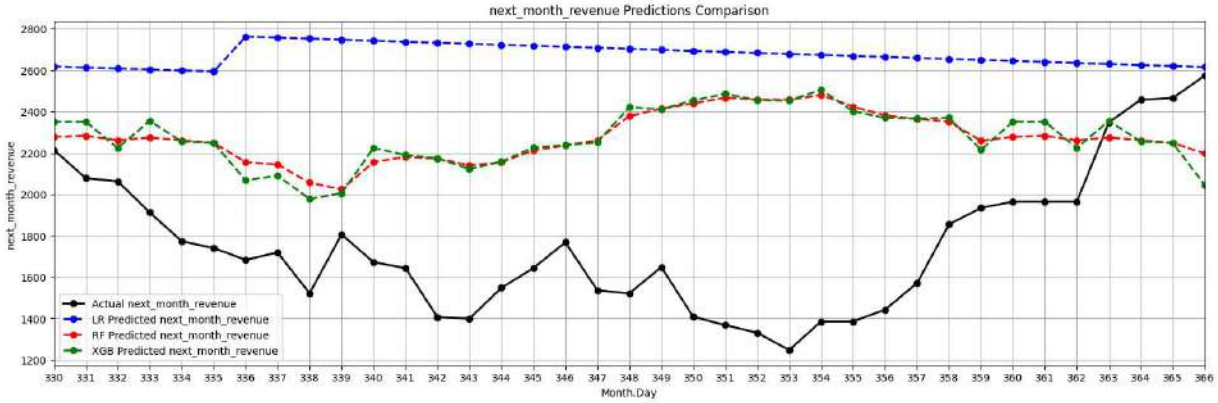

purchases Predictions Comparison
purchases
Month.Day
Actual purchases
LR Predicted purchases
RF Predicted purchases
XGB Predicted purchases

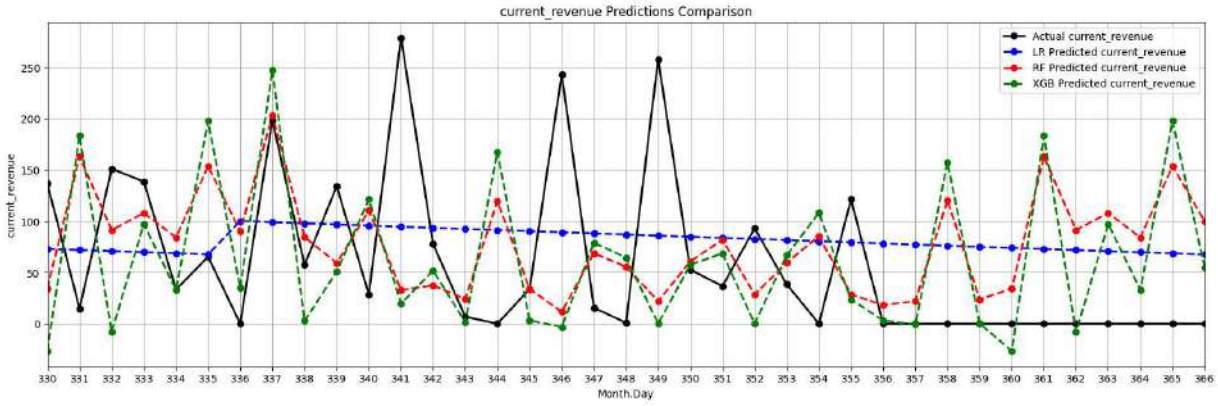

unique_invoices Predictions Comparison
unique_invoices
Month.Day
Actual unique_invoices
LR Predicted unique_invoices
RF Predicted unique_invoices
XGB Predicted unique_invoices

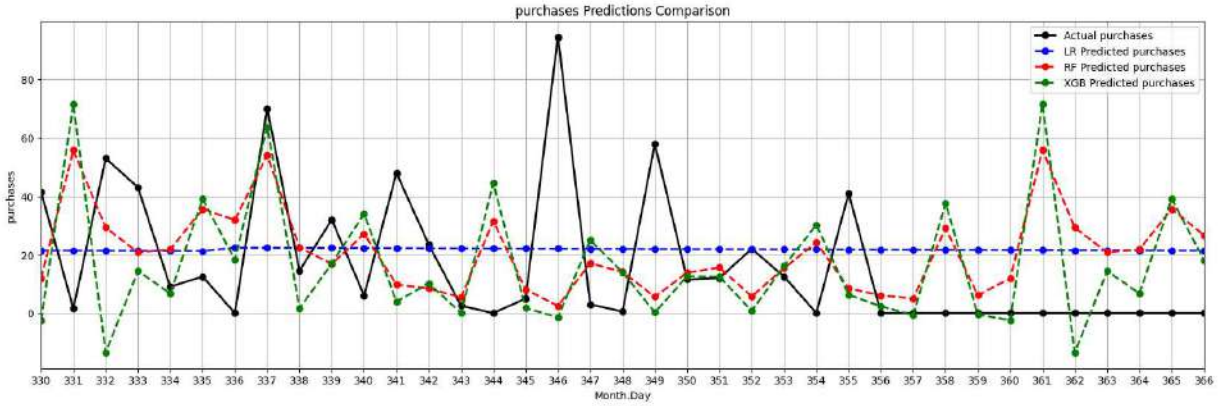


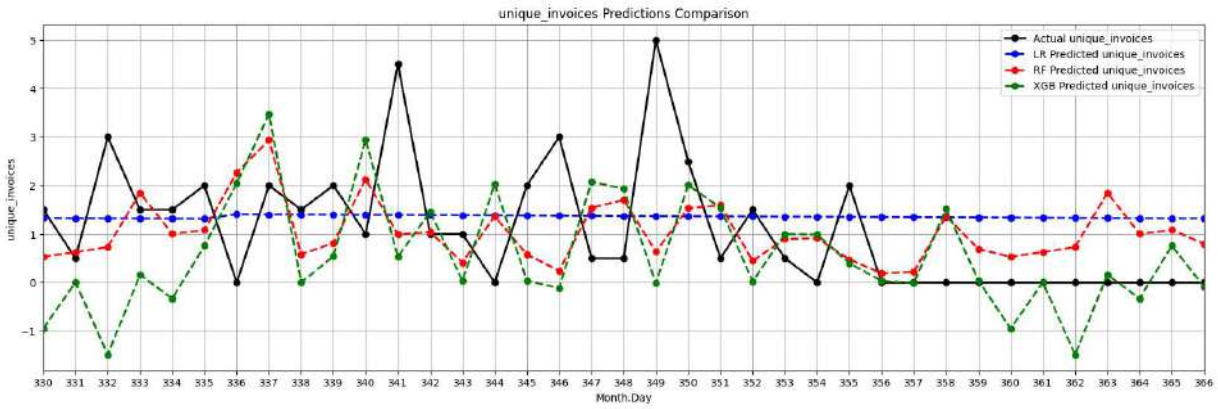


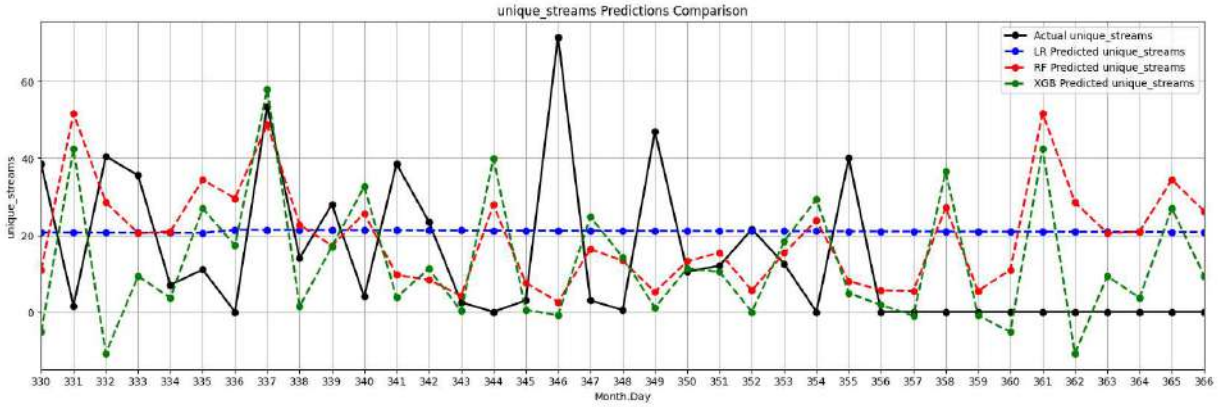
4 - Dataframe df_ts_germany data:

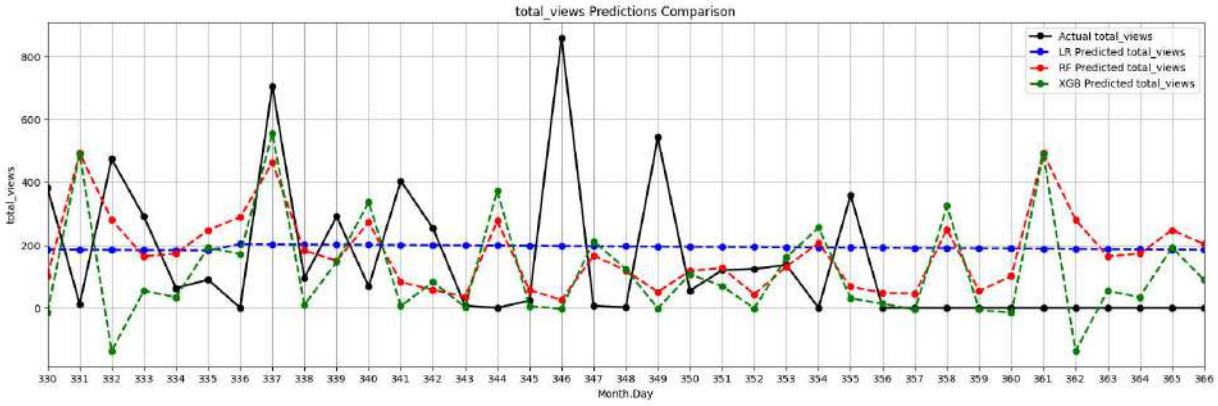




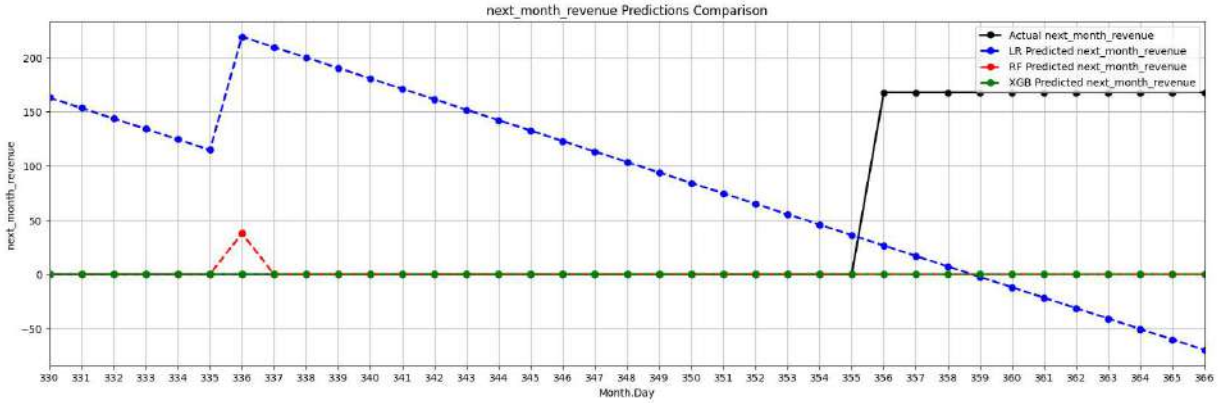


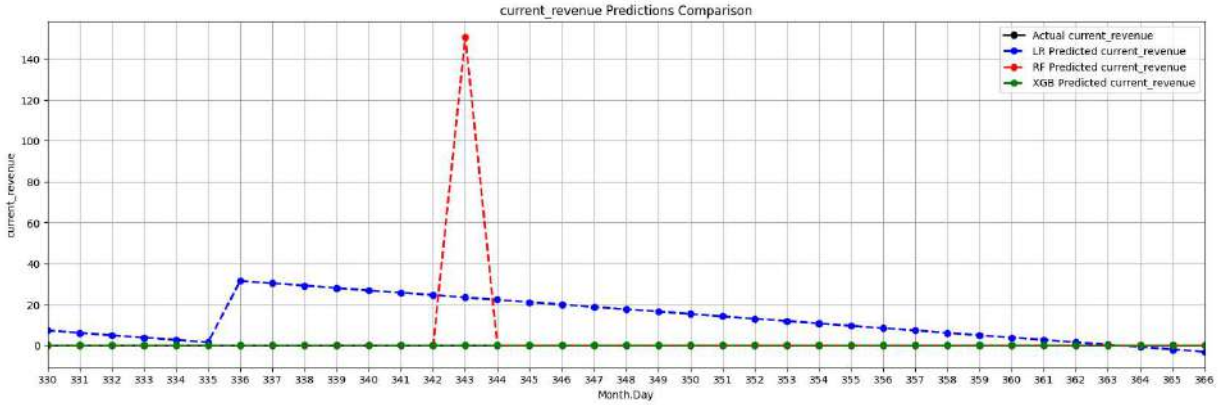


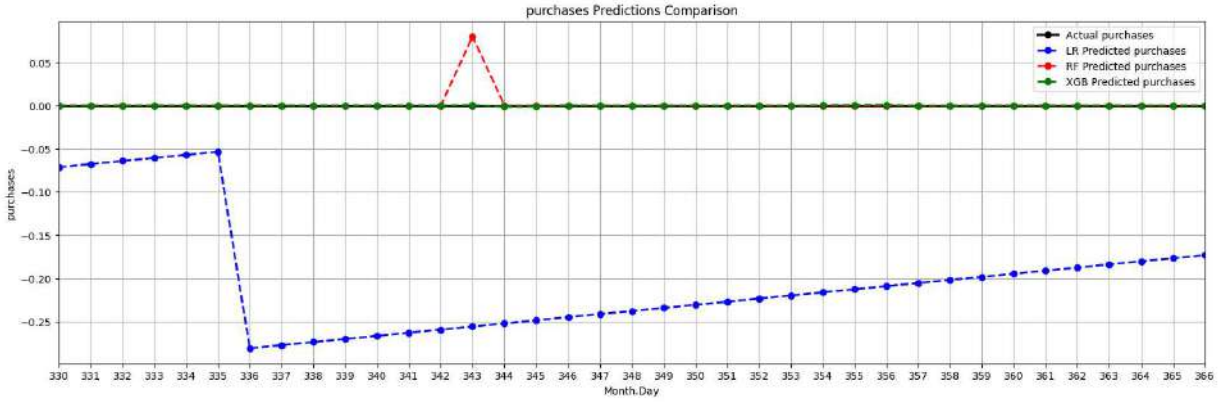


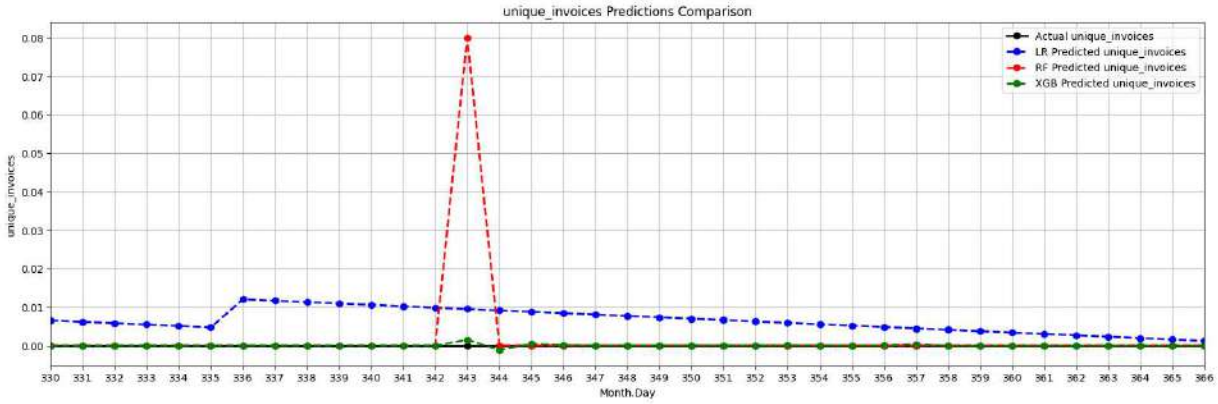


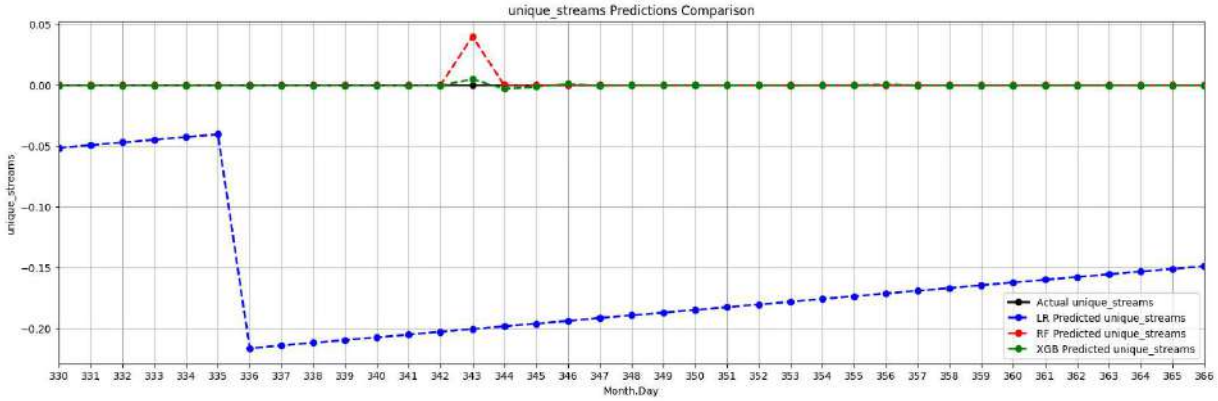
5 - Dataframe df_ts_hong_kong data:

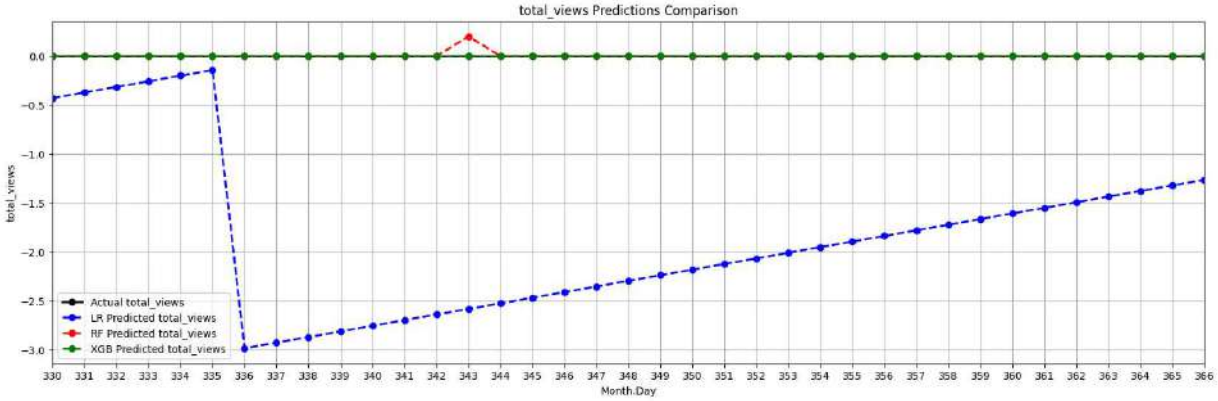

next_month_revenue Predictions Comparison
Actual next_month_revenue
LR Predicted next_month_revenue
RF Predicted next_month_revenue
XGB Predicted next_month_revenue
next_month_revenue
Month.Day


current_revenue Predictions Comparison
Actual current_revenue
LR Predicted current_revenue
RF Predicted current_revenue
XGB Predicted current_revenue
current_revenue
Month.Day

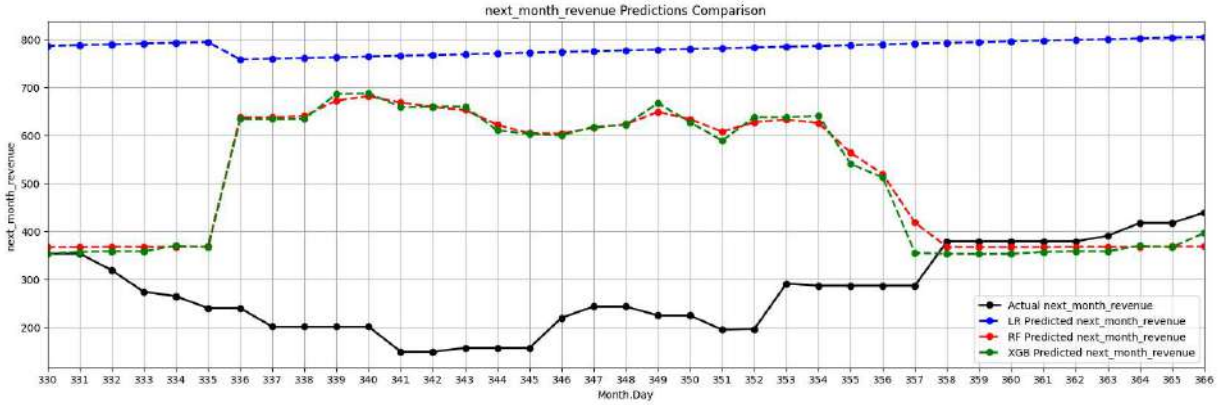

purchases Predictions Comparison
Actual purchases
LR Predicted purchases
RF Predicted purchases
XGB Predicted purchases
purchases
Month.Day

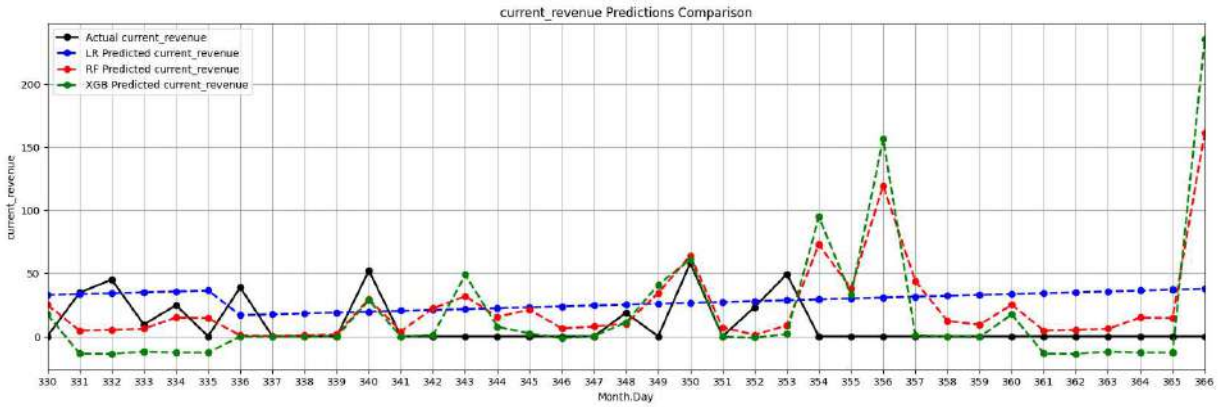

unique_invoices Predictions Comparison
Actual unique_invoices
LR Predicted unique_invoices
RF Predicted unique_invoices
XGB Predicted unique_invoices
unique_invoices
Month.Day

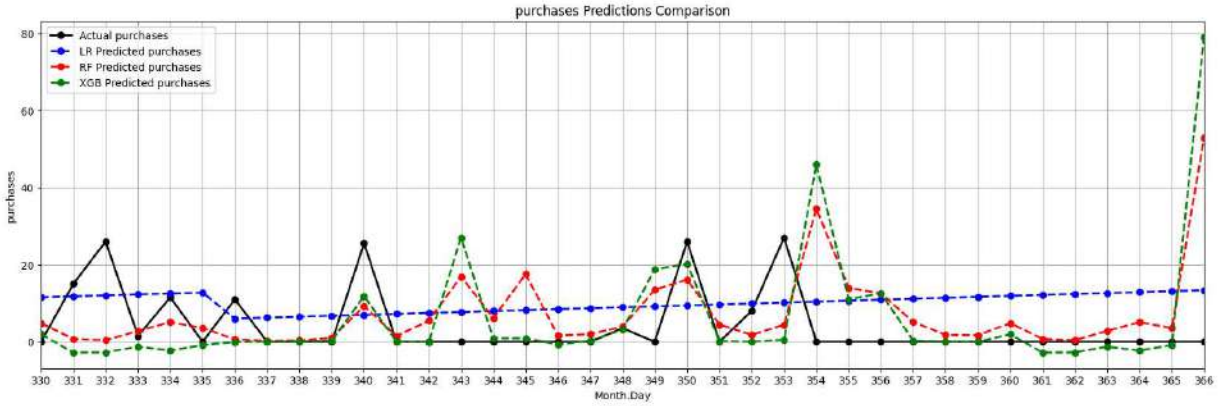


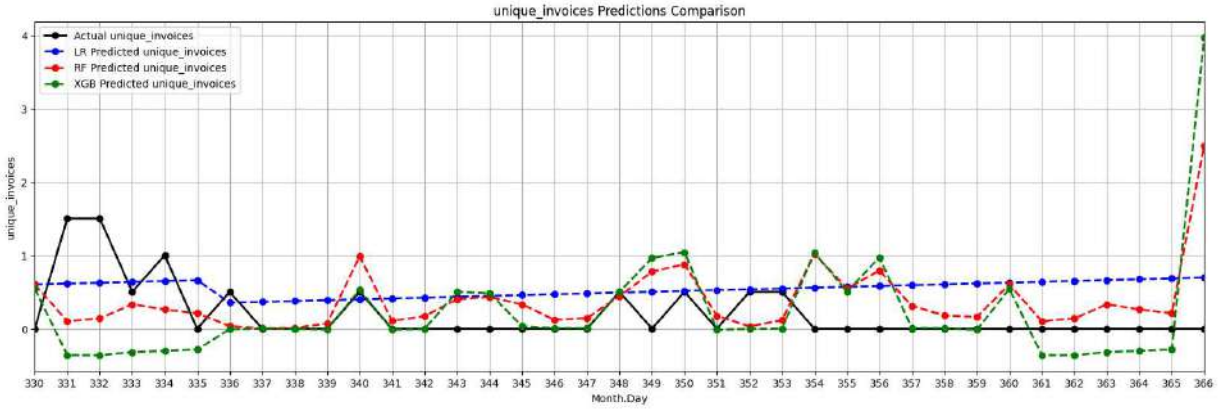


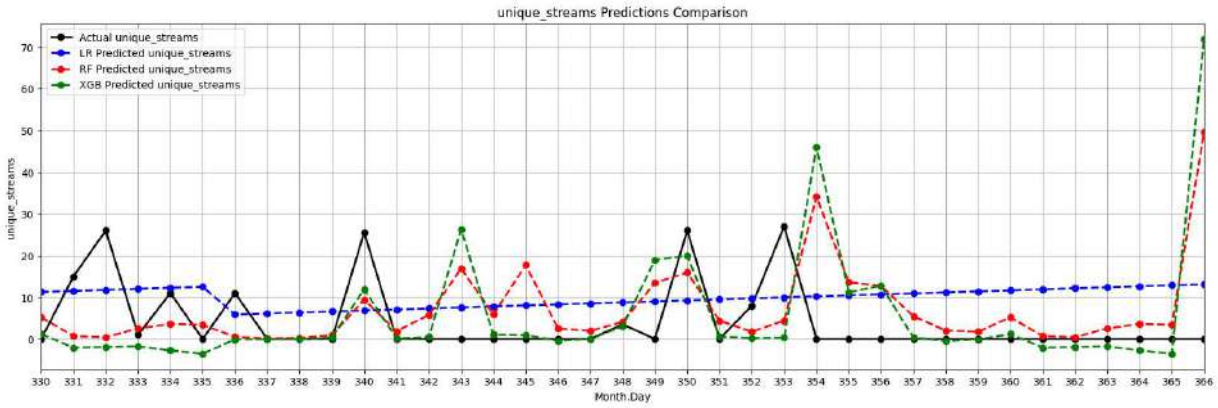
6 - Dataframe df_ts_netherlands data:

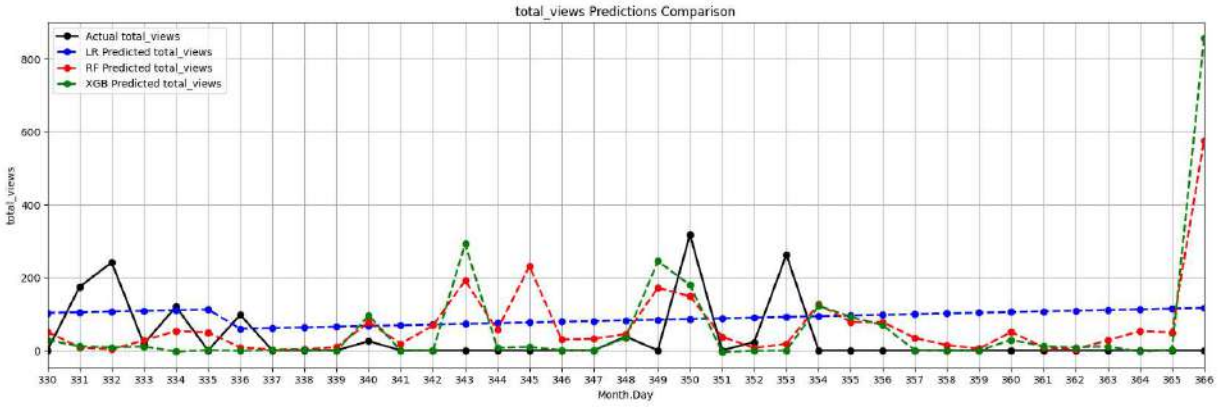




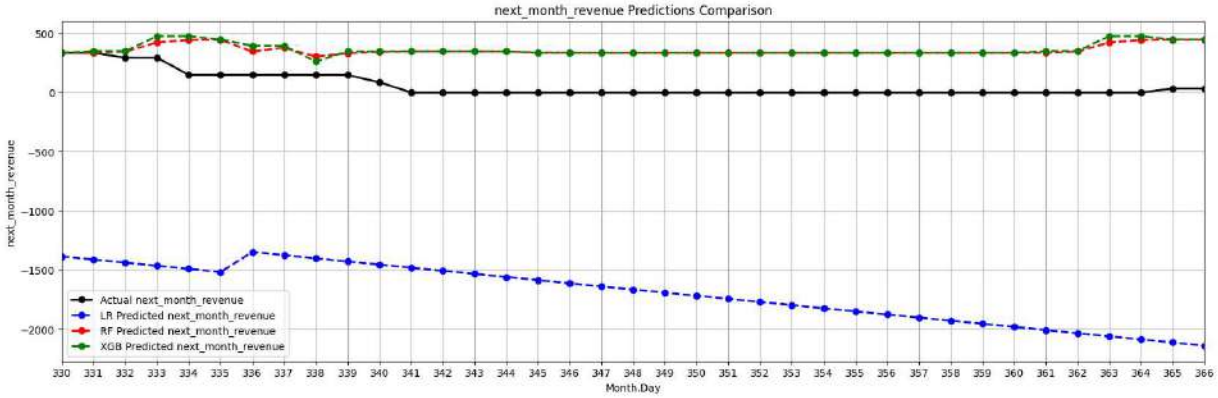


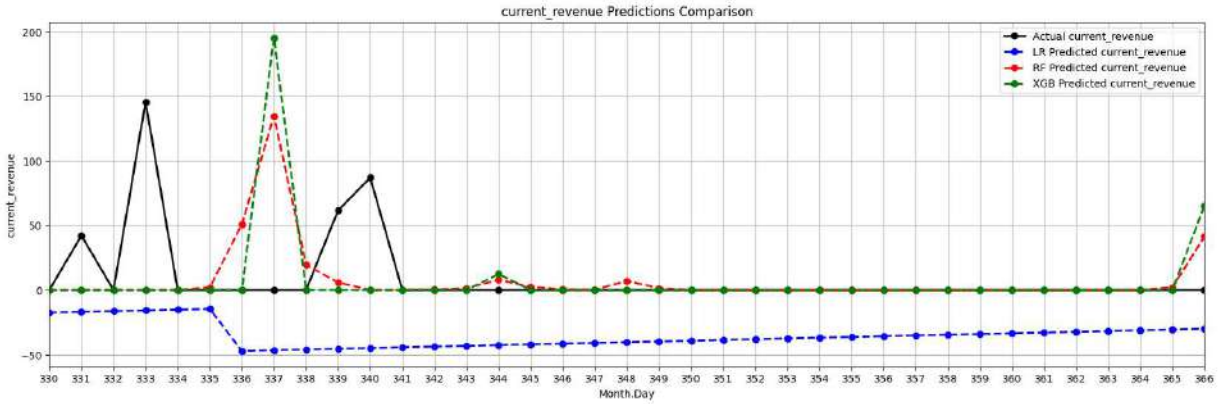


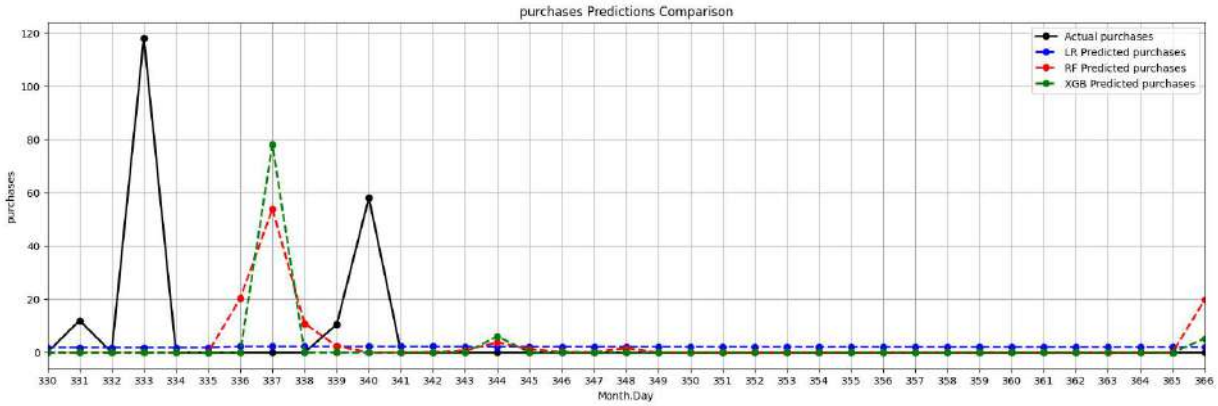


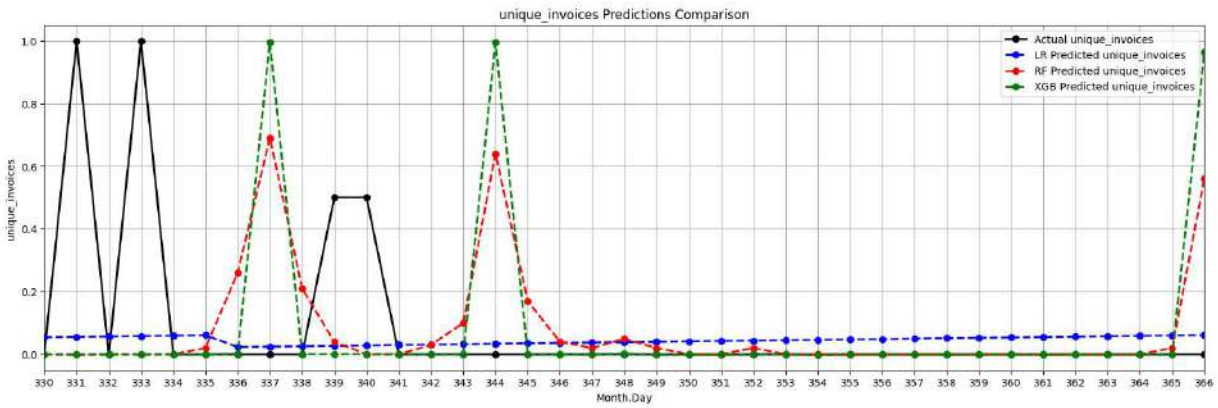


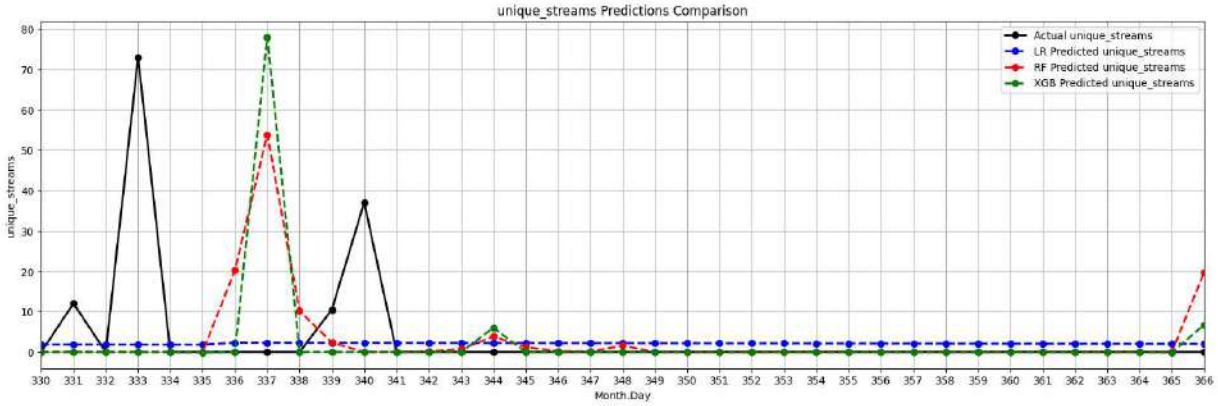
7 - Dataframe df_ts_norway data:

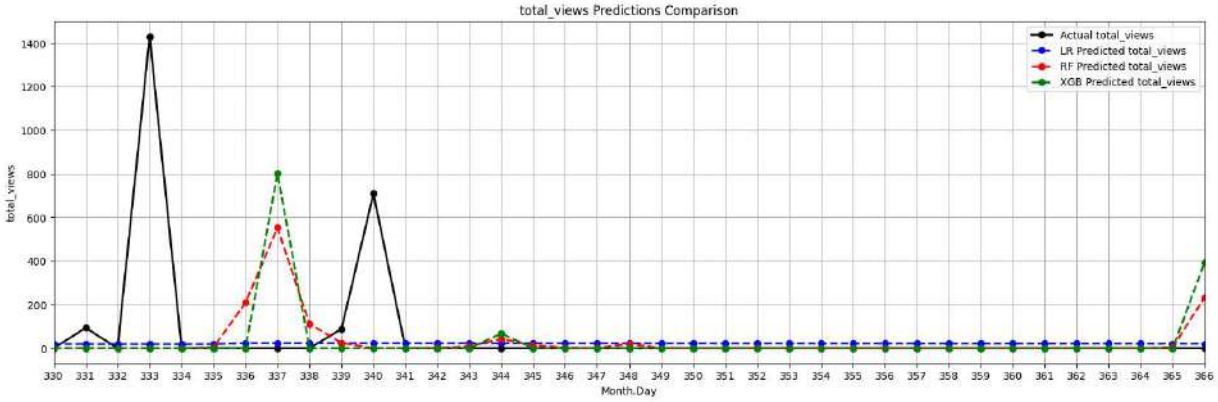

next_month_revenue Predictions Comparison
next_month_revenue
Month.Day
Actual next_month_revenue
LR Predicted next_month_revenue
RF Predicted next_month_revenue
XGB Predicted next_month_revenue


current_revenue Predictions Comparison
current_revenue
Month.Day
Actual current_revenue
LR Predicted current_revenue
RF Predicted current_revenue
XGB Predicted current_revenue

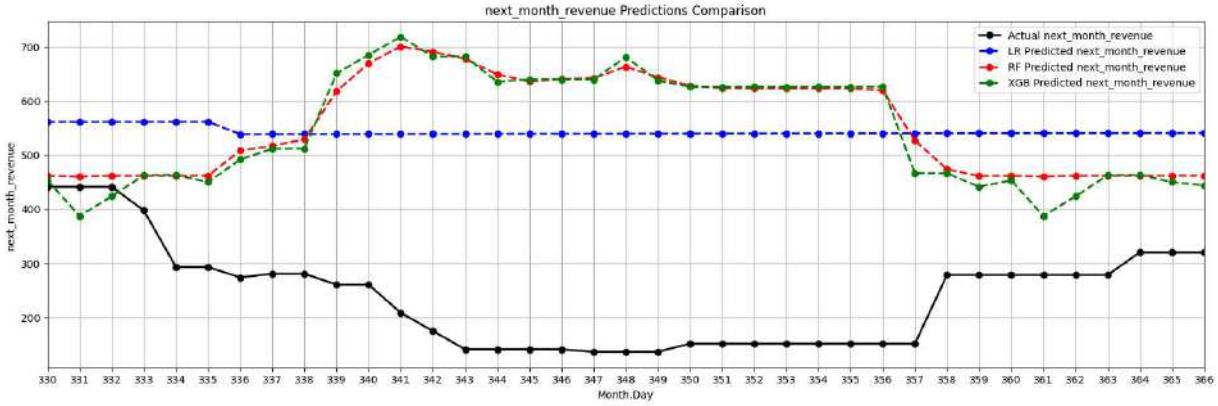

purchases Predictions Comparison
purchases
Month.Day
Actual purchases
LR Predicted purchases
RF Predicted purchases
XGB Predicted purchases

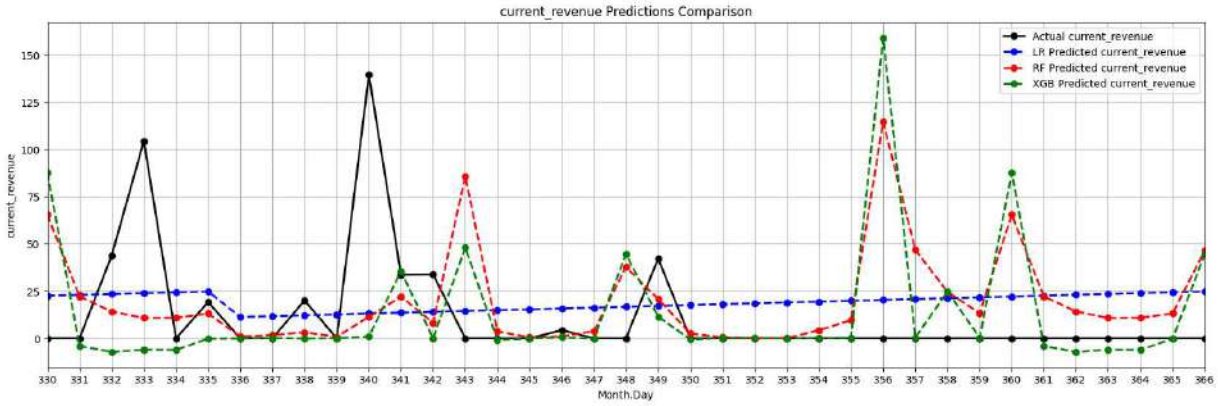

unique_invoices Predictions Comparison
unique_invoices
Month.Day
Actual unique_invoices
LR Predicted unique_invoices
RF Predicted unique_invoices
XGB Predicted unique_invoices

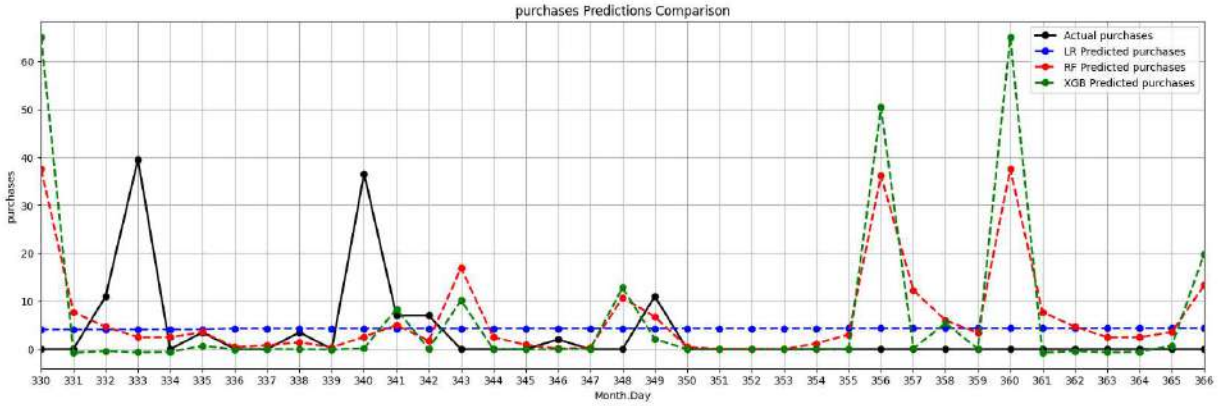


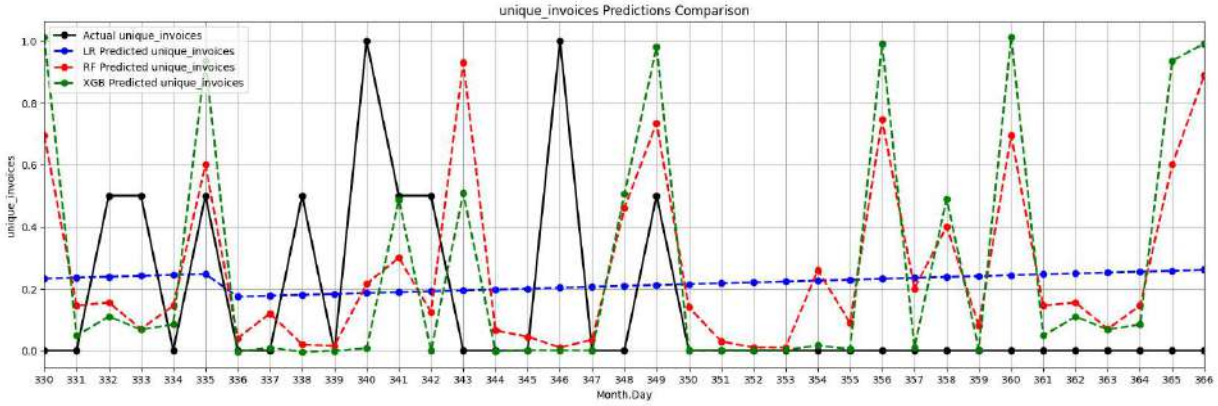


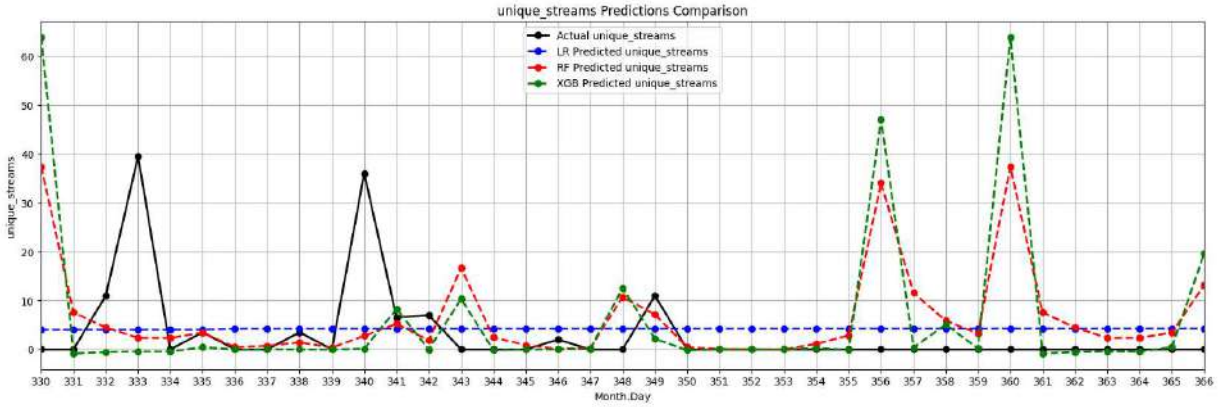
8 - Dataframe df_ts_portugal data:

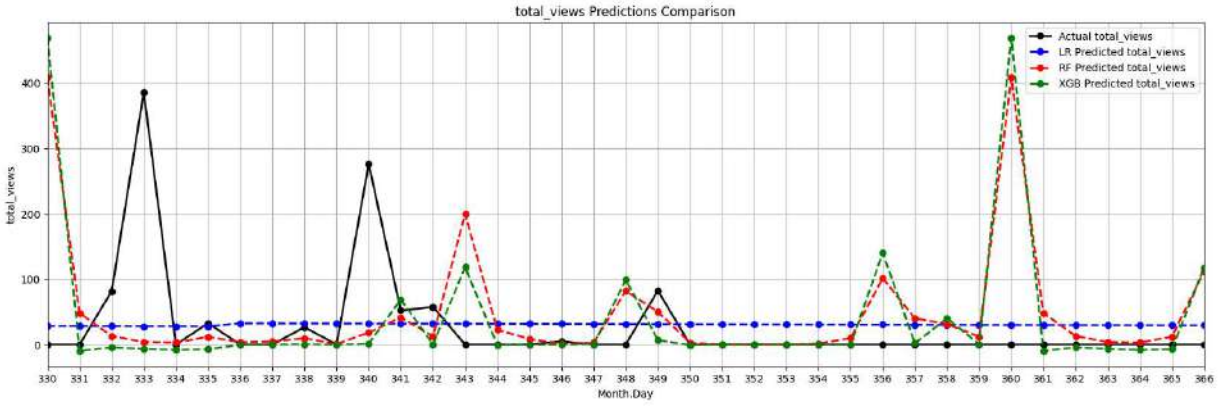




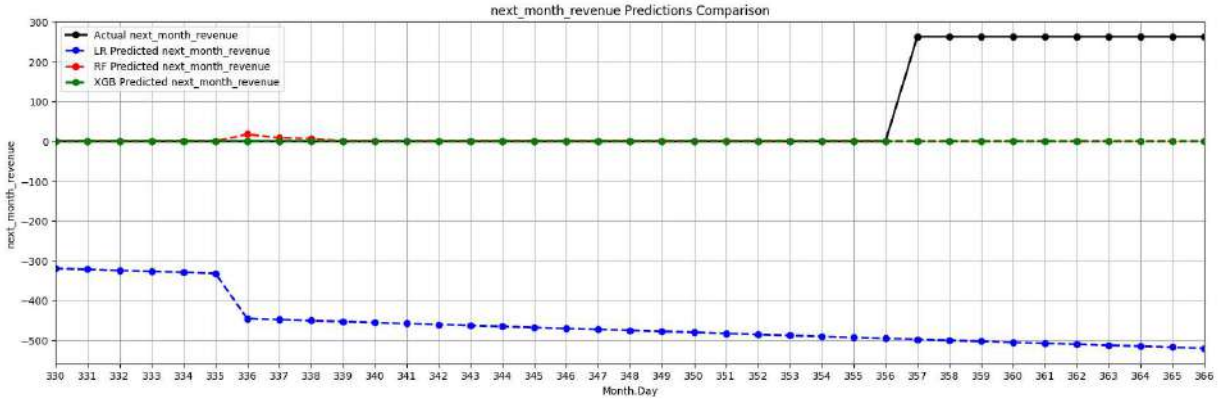


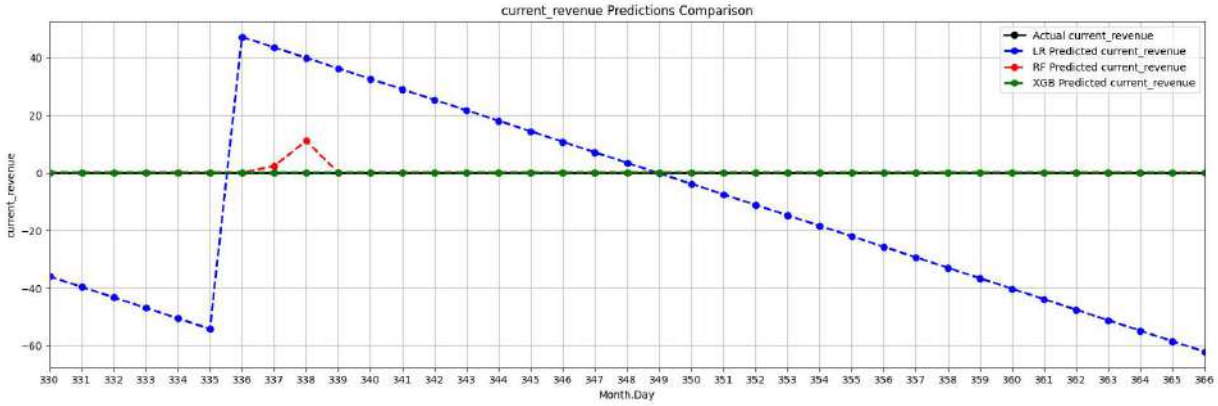


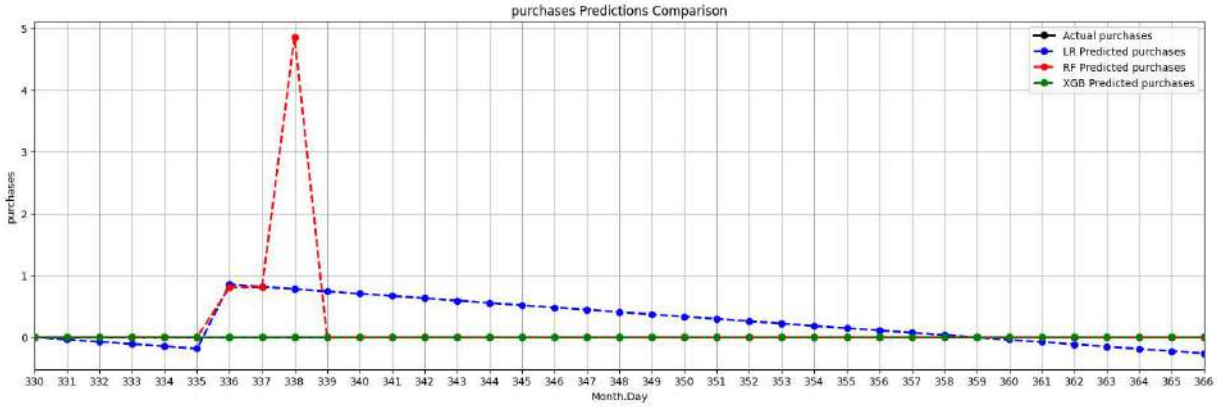


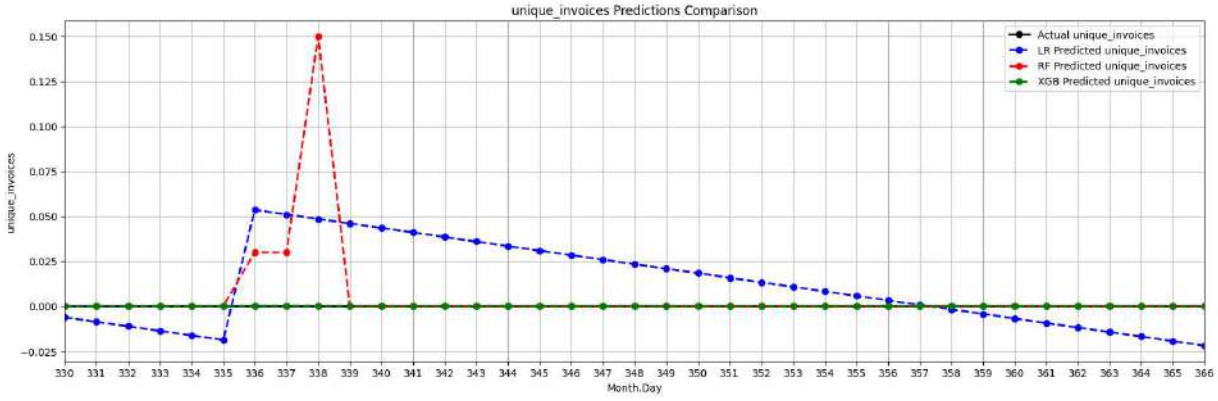


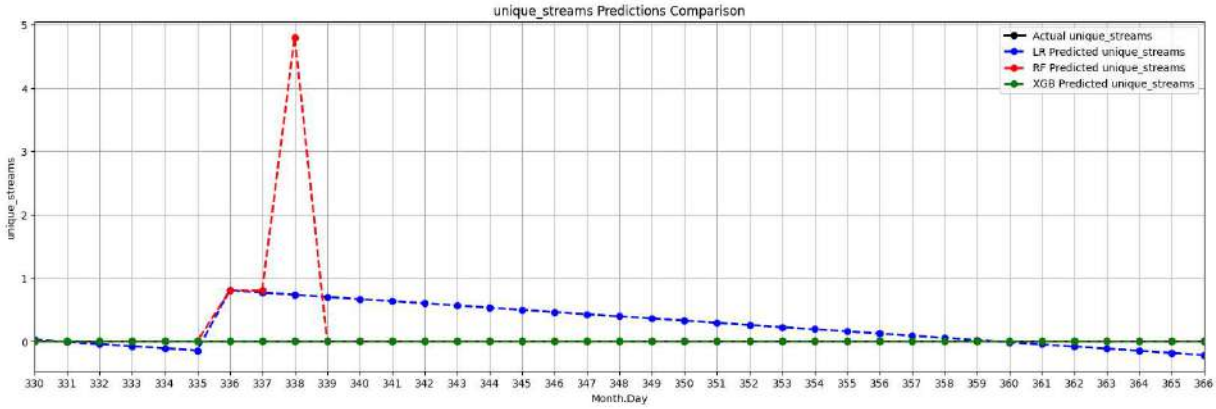
9 - Dataframe df_ts_singapore data:

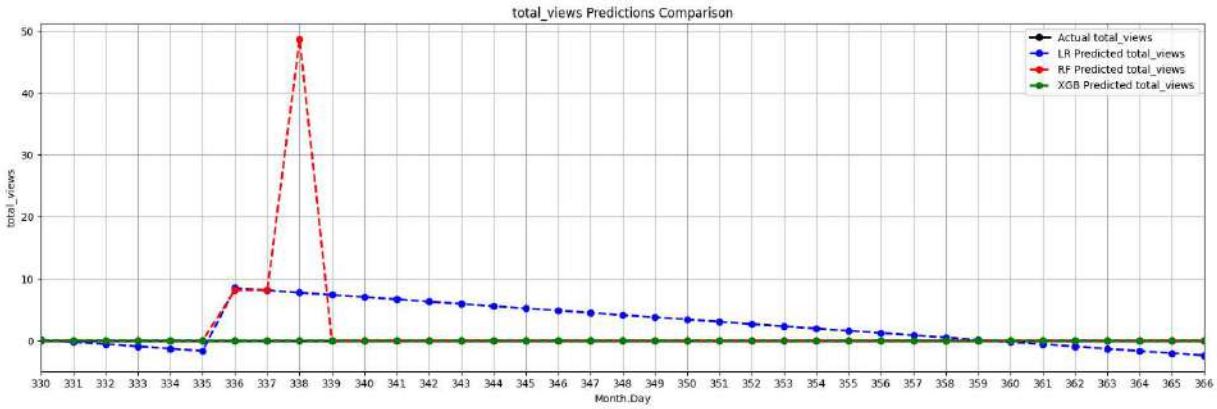

next_month_revenue Predictions Comparison
Actual next_month_revenue
LR Predicted next_month_revenue
RF Predicted next_month_revenue
XGB Predicted next_month_revenue
next_month_revenue
Month.Day


current_revenue Predictions Comparison
Actual current_revenue
LR Predicted current_revenue
RF Predicted current_revenue
XGB Predicted current_revenue
current_revenue
Month.Day

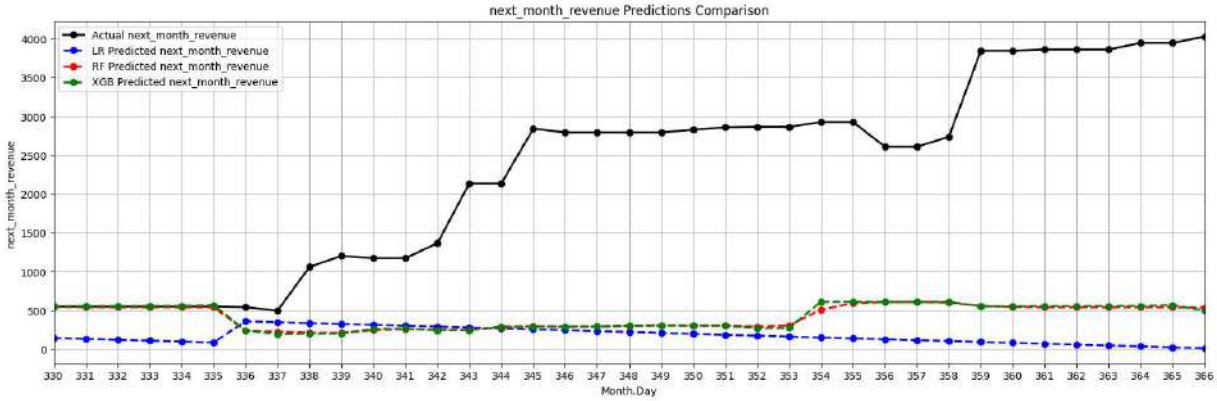

purchases Predictions Comparison
Actual purchases
LR Predicted purchases
RF Predicted purchases
XGB Predicted purchases
purchases
Month.Day

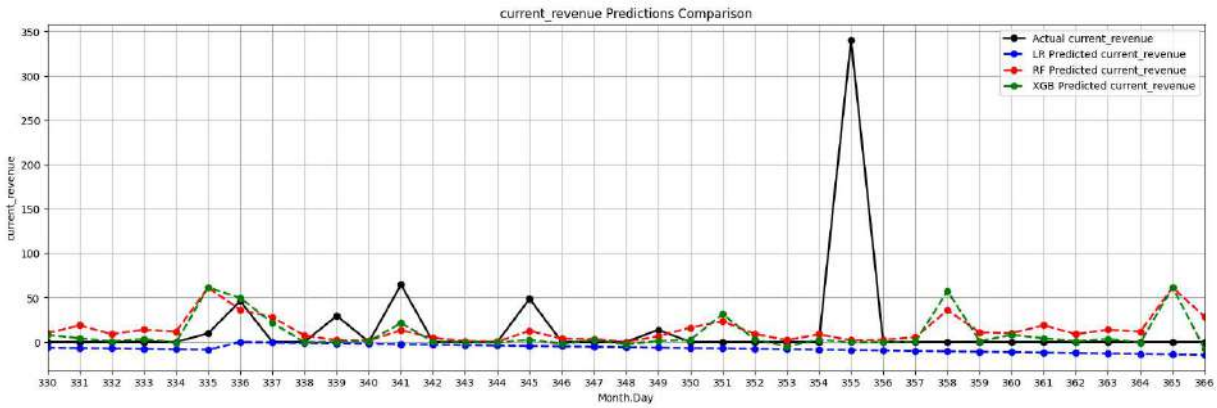

unique_invoices Predictions Comparison
Actual unique_invoices
LR Predicted unique_invoices
RF Predicted unique_invoices
XGB Predicted unique_invoices
unique_invoices
Month.Day

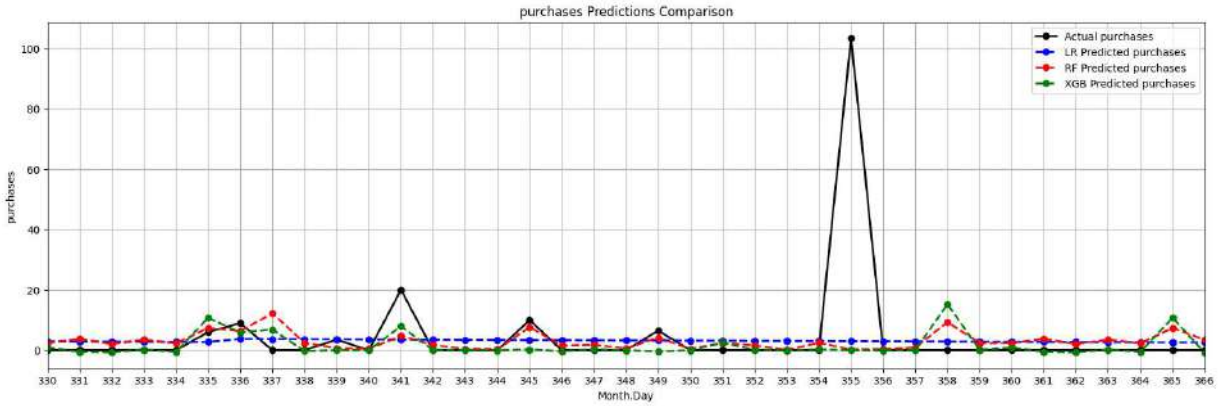


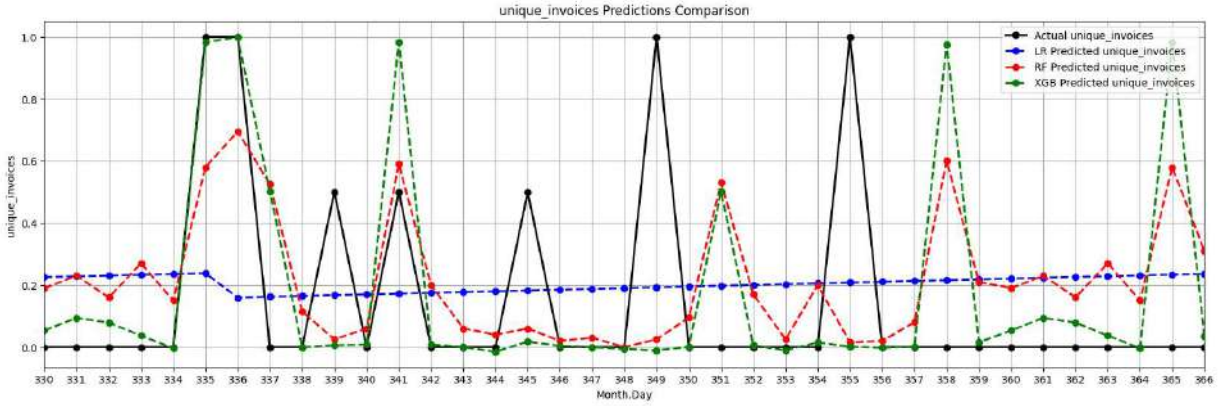


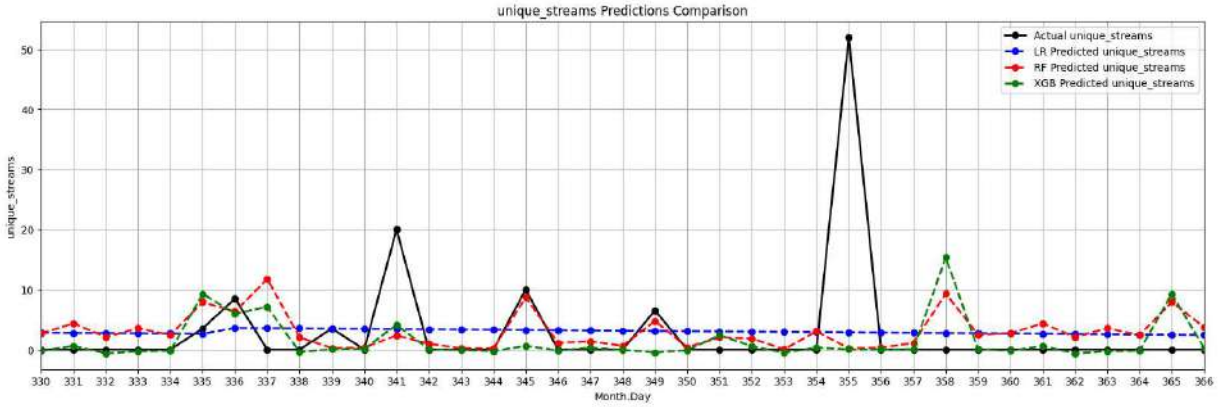
10 - Dataframe df_ts_spain data:

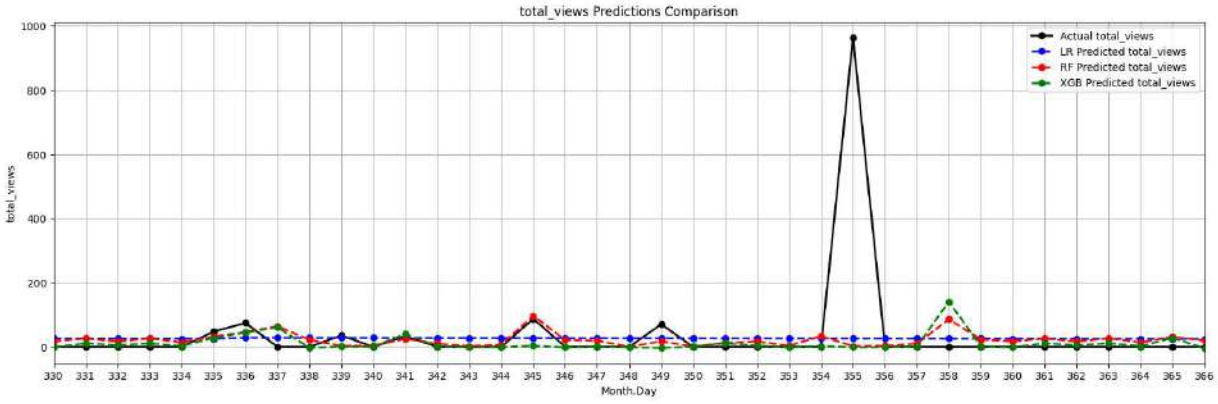




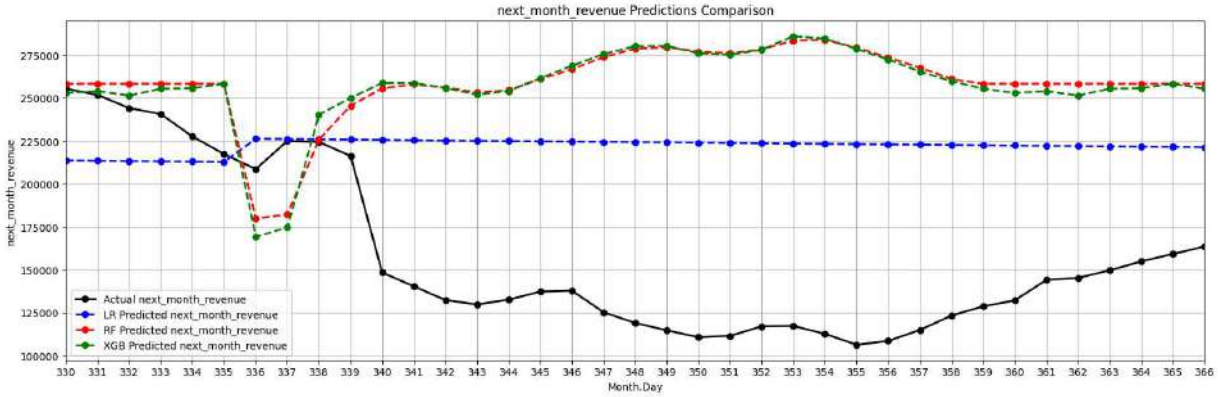


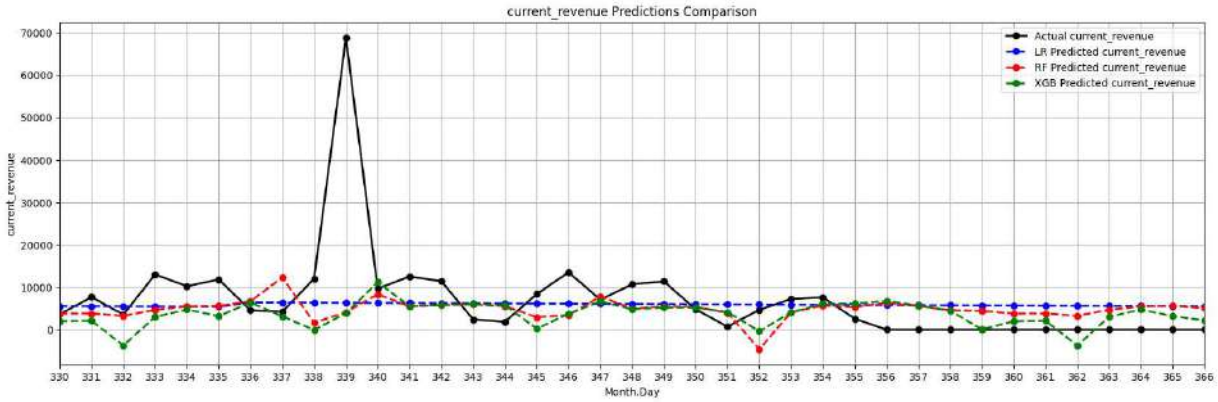


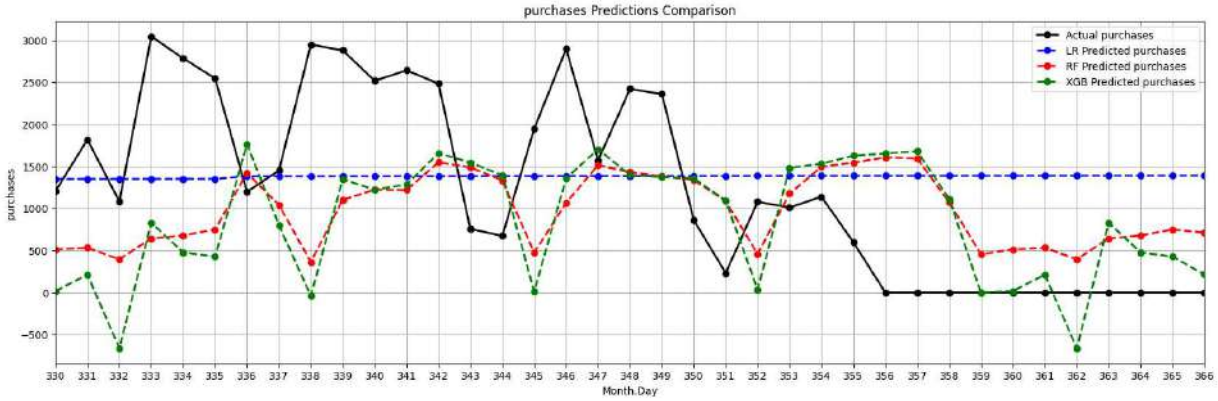


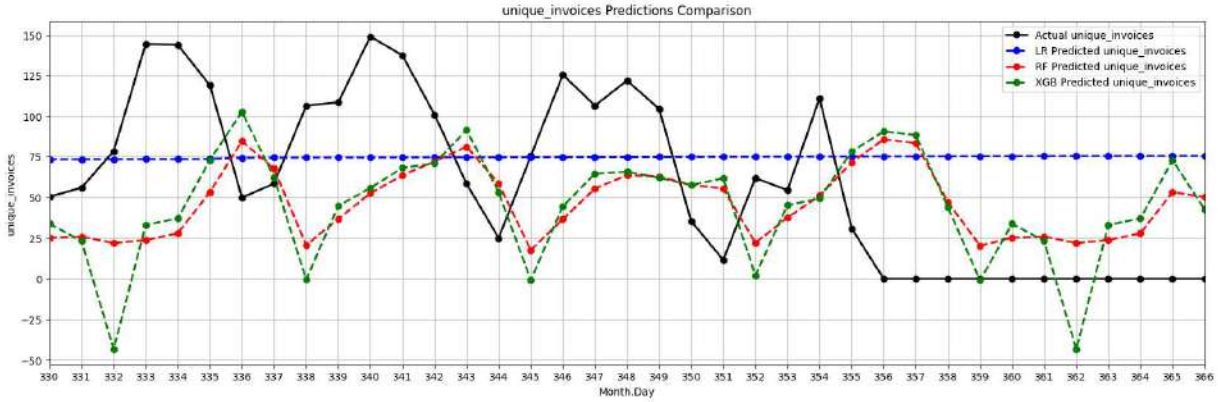


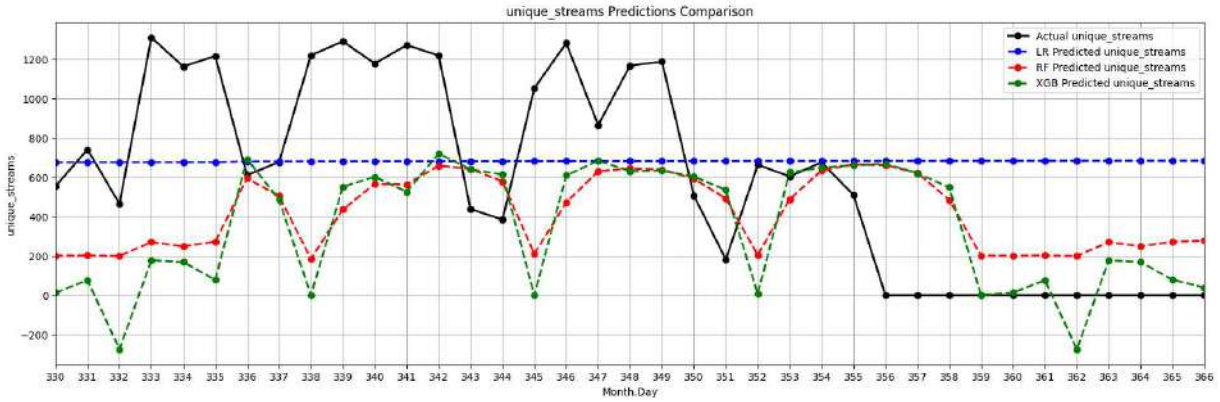
11 - Dataframe df_ts_united_kingdom data:

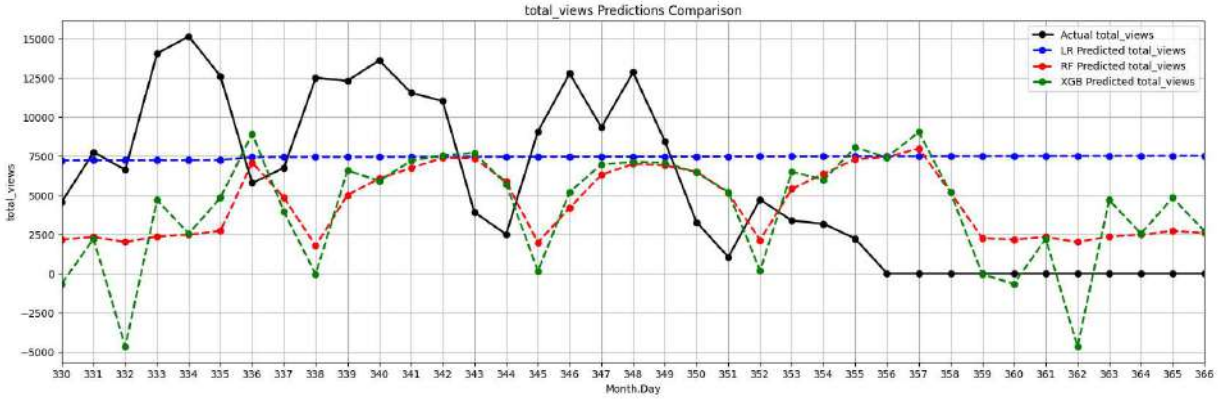

next_month_revenue Predictions Comparison
next_month_revenue
Month.Day
Actual next_month_revenue
LR Predicted next_month_revenue
RF Predicted next_month_revenue
XGB Predicted next_month_revenue


current_revenue Predictions Comparison
current_revenue
Month.Day
Actual current_revenue
LR Predicted current_revenue
RF Predicted current_revenue
XGB Predicted current_revenue


purchases Predictions Comparison
purchases
Month.Day
Actual purchases
LR Predicted purchases
RF Predicted purchases
XGB Predicted purchases


unique_invoices Predictions Comparison
unique_invoices
Month.Day
Actual unique_invoices
LR Predicted unique_invoices
RF Predicted unique_invoices
XGB Predicted unique_invoices





2.5-Test the Models with an additionnal single data

Scenario :

In [27]:
```
#calcualte predicted data using the models for January 3
#Data to use the model
single_month = 1
single_day = 3
test_date = {'current_month': single_month, 'current_day': single_day}
new_test_df = pd.DataFrame(data=test_date, index=([0]))
print (new_test_df)
```

```
   current_month  current_day
0              1            3
```

In [28]:
```
#Use the models for an additionnal prediction
y_test_pred_lr, y_test_pred_rf, y_test_pred_xgb = {}, {}, {}
for i in range (0,size):
    df_name = list_ts_df[i]
    y_test_pred_lr[df_name] = lr_model[df_name].predict(new_test_df)
    y_test_pred_rf[df_name] = rf_model[df_name].predict(new_test_df)
    y_test_pred_xgb[df_name] = xgb_model[df_name].predict(new_test_df)

for i in range (0,size):
    df_name = list_ts_df[i]
    #Compare results
    print(i+1, '- Dataframe', df_name , 'data:')

    print('\nObserved data')
    print(library_ts_df_monthly[df_name]
```

```
        [(library_ts_df_monthly[df_name]['current_month']==test_date['current_month
          (library_ts_df_monthly[df_name]['current_day']==test_date['current_day'])

    print('\nPredicted')
    d1 = {
        'purchases'         : [y_test_pred_lr[df_name][0][0],  y_test_pred_rf[df_na
        'unique_invoices'   : [y_test_pred_lr[df_name][0][1],  y_test_pred_rf[df_na
        'unique_streams'    : [y_test_pred_lr[df_name][0][2],  y_test_pred_rf[df_na
        'total_views'       : [y_test_pred_lr[df_name][0][3],  y_test_pred_rf[df_na
        'current_revenue'   : [y_test_pred_lr[df_name][0][4],  y_test_pred_rf[df_na
        'next_month_revenue': [y_test_pred_lr[df_name][0][5],  y_test_pred_rf[df_na
        }

    report = pd.DataFrame(data=d1, index=(['Linear Regression', 'Random Forest.', '
    print (report.T)
    print('\n')
```

1 - Dataframe df_ts_all data:

Observed data

	2
current_month	1.000
current_day	3.000
purchases	1060.500
unique_invoices	46.000
unique_streams	635.000
total_views	6103.000
current_revenue	4449.060
next_month_revenue	168808.037
next_month	2.000
next_day	3.000

Predicted

	Linear Regression	Random Forest.	XGBoost Model
purchases	152955.602540	176706.91040	168907.593750
unique_invoices	5428.800250	6975.95065	4469.812500
unique_streams	1013.914878	992.91000	1049.506104
total_views	58.055699	38.15500	46.163815
current_revenue	639.909731	614.85500	632.859253
next_month_revenue	5254.989702	5533.56000	6136.808594

2 - Dataframe df_ts_eire data:

Observed data

	2
current_month	1.000
current_day	3.000
purchases	0.000
unique_invoices	0.000
unique_streams	0.000
total_views	0.000
current_revenue	0.000
next_month_revenue	8024.775
next_month	2.000
next_day	3.000

Predicted

	Linear Regression	Random Forest.	XGBoost Model
purchases	3745.246640	7572.18345	8031.564453
unique_invoices	81.773403	37.88610	5.394011
unique_streams	11.782674	5.62500	0.840271
total_views	0.575494	0.21000	0.071242
current_revenue	12.233350	5.42500	0.355781
next_month_revenue	89.873029	85.67000	6.225188

3 - Dataframe df_ts_france data:

Observed data

	2
current_month	1.000
current_day	3.000

purchases	23.500
unique_invoices	1.500
unique_streams	22.000
total_views	157.500
current_revenue	117.775
next_month_revenue	2031.530
next_month	2.000
next_day	3.000

Predicted

	Linear Regression	Random Forest.	XGBoost Model
purchases	1829.881548	2060.92675	2030.419312
unique_invoices	38.952721	113.94955	116.576935
unique_streams	15.724192	26.31000	23.479624
total_views	0.789208	1.40500	1.487928
current_revenue	15.018424	23.27500	21.999340
next_month_revenue	151.998912	197.81500	158.808563

4 - Dataframe df_ts_germany data:

Observed data

	2
current_month	1.000
current_day	3.000
purchases	0.500
unique_invoices	0.500
unique_streams	0.500
total_views	2.000
current_revenue	0.825
next_month_revenue	2654.460
next_month	2.000
next_day	3.000

Predicted

	Linear Regression	Random Forest.	XGBoost Model
purchases	2461.409485	2664.93465	2655.275879
unique_invoices	87.375566	25.61550	1.061122
unique_streams	20.547388	6.09500	0.524652
total_views	1.309123	0.52000	0.493359
current_revenue	19.095695	6.45500	0.665101
next_month_revenue	170.832779	44.85000	4.969006

5 - Dataframe df_ts_hong_kong data:

Observed data

	2
current_month	1.00
current_day	3.00
purchases	0.00
unique_invoices	0.00
unique_streams	0.00
total_views	0.00
current_revenue	0.00
next_month_revenue	167.66

next_month	2.00
next_day	3.00

Predicted

	Linear Regression	Random Forest.	XGBoost Model
purchases	2125.171305	167.66	167.661255
unique_invoices	67.336802	0.00	-0.001383
unique_streams	1.082274	0.00	-0.000056
total_views	0.045898	0.00	0.000227
current_revenue	1.004338	0.00	0.000173
next_month_revenue	10.084095	0.00	0.000123

6 - Dataframe df_ts_netherlands data:

Observed data

	2
current_month	1.000
current_day	3.000
purchases	0.000
unique_invoices	0.000
unique_streams	0.000
total_views	0.000
current_revenue	0.000
next_month_revenue	438.785
next_month	2.000
next_day	3.000

Predicted

	Linear Regression	Random Forest.	XGBoost Model
purchases	658.814235	439.01635	439.144379
unique_invoices	10.280321	0.00000	-0.122444
unique_streams	2.414478	0.00000	-0.277074
total_views	0.109038	0.01500	-0.010367
current_revenue	2.353695	0.43500	-0.044680
next_month_revenue	37.172858	0.00000	-1.320354

7 - Dataframe df_ts_norway data:

Observed data

	2
current_month	1.00
current_day	3.00
purchases	0.00
unique_invoices	0.00
unique_streams	0.00
total_views	0.00
current_revenue	0.00
next_month_revenue	32.65
next_month	2.00
next_day	3.00

Predicted

	Linear Regression	Random Forest.	XGBoost Model
purchases	5158.607434	32.65	32.639015

unique_invoices	130.017392	0.00	-0.000567
unique_streams	0.039008	0.00	-0.000435
total_views	0.033048	0.00	-0.000164
current_revenue	-0.007286	0.00	-0.000649
next_month_revenue	-1.235935	0.00	-0.001146

8 - Dataframe df_ts_portugal data:

Observed data

	2
current_month	1.000
current_day	3.000
purchases	0.000
unique_invoices	0.000
unique_streams	0.000
total_views	0.000
current_revenue	0.000
next_month_revenue	313.175
next_month	2.000
next_day	3.000

Predicted

	Linear Regression	Random Forest.	XGBoost Model
purchases	768.147217	315.721	313.247406
unique_invoices	16.697371	2.077	0.474245
unique_streams	1.421564	0.620	0.038717
total_views	0.062225	0.140	0.007698
current_revenue	1.409698	0.620	0.002967
next_month_revenue	14.285521	0.280	-0.252822

9 - Dataframe df_ts_singapore data:

Observed data

	2
current_month	1.00
current_day	3.00
purchases	0.00
unique_invoices	0.00
unique_streams	0.00
total_views	0.00
current_revenue	0.00
next_month_revenue	263.34
next_month	2.00
next_day	3.00

Predicted

	Linear Regression	Random Forest.	XGBoost Model
purchases	1593.683533	263.34	263.340088
unique_invoices	87.100123	0.00	0.000488
unique_streams	1.236344	0.00	0.000183
total_views	0.056100	0.00	0.000160
current_revenue	1.170311	0.00	0.000192
next_month_revenue	11.869002	0.00	0.000205

10 - Dataframe df_ts_spain data:

Observed data

	2
current_month	1.00
current_day	3.00
purchases	0.50
unique_invoices	0.50
unique_streams	0.50
total_views	1.50
current_revenue	563.00
next_month_revenue	4030.31
next_month	2.00
next_day	3.00

Predicted

	Linear Regression	Random Forest.	XGBoost Model
purchases	1031.102461	4000.7268	4030.559570
unique_invoices	55.916530	321.0128	562.078796
unique_streams	5.066293	2.7300	0.535561
total_views	0.214602	0.3000	0.497171
current_revenue	5.014270	2.7300	0.484819
next_month_revenue	34.452262	11.5600	1.934026

11 - Dataframe df_ts_united_kingdom data:

Observed data

	2
current_month	1.000
current_day	3.000
purchases	1008.500
unique_invoices	41.500
unique_streams	617.500
total_views	5639.000
current_revenue	3709.820
next_month_revenue	147300.177
next_month	2.000
next_day	3.000

Predicted

	Linear Regression	Random Forest.	XGBoost Model
purchases	130304.731993	154526.1724	147285.500000
unique_invoices	4816.608577	6399.7440	3742.060303
unique_streams	936.787598	924.6650	1012.339355
total_views	53.704941	35.8200	41.088558
current_revenue	605.116418	601.9000	620.478271
next_month_revenue	4577.784282	5356.9200	5671.266602

3-Create and Test 3 Multi target regression Models using 1 column day_of_week as independent

variables

This strategy consists of fitting one regressor per target. This is a simple strategy for extending regressors that do not natively support multi-target regression.

In [29]:
```
#Convert month and day to day of year in the time-series dataset
library_ts_df_dayofyear = {}
for i in range (0,size):
    df_name = list_ts_df[i]
    print(i+1, '- Dataframe', df_name , 'data:')
    library_ts_df_dayofyear[df_name] = model.to_day_of_year_df(library_ts_df_monthl
    print(library_ts_df_dayofyear[df_name])
```

```
1 - Dataframe df_ts_all data:
     day_of_year  current_month  current_day  purchases  unique_invoices  \
0              1              1            1      918.0             63.5
1              2              1            2     1335.0             61.0
2              3              1            3     1060.5             46.0
3              4              1            4     1427.5             72.5
4              5              1            5      356.0             24.5
..           ...            ...          ...        ...              ...
360          362             12           27        0.0              0.0
361          363             12           28        0.0              0.0
362          364             12           29        0.0              0.0
363          365             12           30        0.0              0.0
364          366             12           31        0.0              0.0

     unique_streams  total_views  current_revenue  next_month_revenue  \
0             677.0       4301.0         3118.810         190490.7520
1             828.5       7017.0        21114.805         189528.2370
2             635.0       6103.0         4449.060         168808.0370
3             865.0       9310.5         4616.005         168161.1370
4             252.5       2727.5         2477.890         169749.7020
..              ...          ...              ...                ...
360             0.0          0.0            0.000         163377.9160
361             0.0          0.0            0.000         168410.0210
362             0.0          0.0            0.000         174878.7065
363             0.0          0.0            0.000         180112.4965
364             0.0          0.0            0.000         184987.5220

     next_month  next_day
0             2         1
1             2         2
2             2         3
3             2         4
4             2         5
..          ...       ...
360           1        27
361           1        28
362           1        29
363           1        30
364           1        31

[365 rows x 11 columns]
2 - Dataframe df_ts_eire data:
     day_of_year  current_month  current_day  purchases  unique_invoices  \
0              1              1            1        1.0              0.5
1              2              1            2        7.5              0.5
2              3              1            3        0.0              0.0
3              4              1            4       81.5              2.5
4              5              1            5       16.5              2.0
..           ...            ...          ...        ...              ...
360          362             12           27        0.0              0.0
361          363             12           28        0.0              0.0
362          364             12           29        0.0              0.0
363          365             12           30        0.0              0.0
364          366             12           31        0.0              0.0

     unique_streams  total_views  current_revenue  next_month_revenue  \
```

```
0              1.0          2.0            7.000            8076.660
1              7.5         20.0           44.885            8069.660
2              0.0          0.0            0.000            8024.775
3             79.0        913.0          233.275            8057.415
4             15.5        130.0         1231.835            8266.740
..             ...          ...              ...                 ...
360            0.0          0.0            0.000            7036.985
361            0.0          0.0            0.000            7067.385
362            0.0          0.0            0.000            7137.565
363            0.0          0.0            0.000            7912.230
364            0.0          0.0            0.000            8023.635

     next_month  next_day
0             2         1
1             2         2
2             2         3
3             2         4
4             2         5
..          ...       ...
360           1        27
361           1        28
362           1        29
363           1        30
364           1        31

[365 rows x 11 columns]
3 - Dataframe df_ts_france data:
     day_of_year  current_month  current_day  purchases  unique_invoices  \
0              1              1            1        0.0              0.0
1              2              1            2       38.0              2.0
2              3              1            3       23.5              1.5
3              4              1            4       37.5              1.5
4              5              1            5       62.5              1.5
..           ...            ...          ...        ...              ...
360          362             12           27        0.0              0.0
361          363             12           28        0.0              0.0
362          364             12           29        0.0              0.0
363          365             12           30        0.0              0.0
364          366             12           31        0.0              0.0

     unique_streams  total_views  current_revenue  next_month_revenue  \
0               0.0          0.0            0.000             2159.170
1              33.0        327.5          127.640             2159.170
2              22.0        157.5          117.775             2031.530
3              36.0        395.5          105.900             1966.830
4              60.5        605.0          250.595             1876.225
..              ...          ...              ...                 ...
360             0.0          0.0            0.000             1792.170
361             0.0          0.0            0.000             1909.510
362             0.0          0.0            0.000             1950.535
363             0.0          0.0            0.000             2004.780
364             0.0          0.0            0.000             2054.700

     next_month  next_day
0             2         1
1             2         2
```

```
2             2         3
3             2         4
4             2         5
..          ...       ...
360           1        27
361           1        28
362           1        29
363           1        30
364           1        31

[365 rows x 11 columns]
4 - Dataframe df_ts_germany data:
     day_of_year  current_month  current_day  purchases  unique_invoices  \
0              1              1            1        1.5              0.5
1              2              1            2        8.5              1.0
2              3              1            3        0.5              0.5
3              4              1            4      121.0              3.0
4              5              1            5        0.0              0.0
..           ...            ...          ...        ...              ...
360          362             12           27        0.0              0.0
361          363             12           28        0.0              0.0
362          364             12           29        0.0              0.0
363          365             12           30        0.0              0.0
364          366             12           31        0.0              0.0

     unique_streams  total_views  current_revenue  next_month_revenue  \
0               1.5          4.0            8.145             2620.935
1               8.5         16.5           35.745             2690.205
2               0.5          2.0            0.825             2654.460
3             104.0       1016.0          341.745             2693.430
4               0.0          0.0            0.000             2498.340
..              ...          ...              ...                  ...
360             0.0          0.0            0.000             1964.725
361             0.0          0.0            0.000             2348.090
362             0.0          0.0            0.000             2457.845
363             0.0          0.0            0.000             2465.695
364             0.0          0.0            0.000             2574.270

     next_month  next_day
0             2         1
1             2         2
2             2         3
3             2         4
4             2         5
..          ...       ...
360           1        27
361           1        28
362           1        29
363           1        30
364           1        31

[365 rows x 11 columns]
5 - Dataframe df_ts_hong_kong data:
     day_of_year  current_month  current_day  purchases  unique_invoices  \
0              1              1            1        0.0              0.0
1              2              1            2        0.0              0.0
```

```
2              3             1             3        0.0              0.0
3              4             1             4        0.0              0.0
4              5             1             5        0.0              0.0
..           ...           ...           ...        ...              ...
360          362            12            27        0.0              0.0
361          363            12            28        0.0              0.0
362          364            12            29        0.0              0.0
363          365            12            30        0.0              0.0
364          366            12            31        0.0              0.0

     unique_streams  total_views  current_revenue  next_month_revenue  \
0               0.0          0.0              0.0              167.66
1               0.0          0.0              0.0              167.66
2               0.0          0.0              0.0              167.66
3               0.0          0.0              0.0              167.66
4               0.0          0.0              0.0              167.66
..              ...          ...              ...                 ...
360             0.0          0.0              0.0              167.66
361             0.0          0.0              0.0              167.66
362             0.0          0.0              0.0              167.66
363             0.0          0.0              0.0              167.66
364             0.0          0.0              0.0              167.66

     next_month  next_day
0             2         1
1             2         2
2             2         3
3             2         4
4             2         5
..          ...       ...
360           1        27
361           1        28
362           1        29
363           1        30
364           1        31

[365 rows x 11 columns]
6 - Dataframe df_ts_netherlands data:
     day_of_year  current_month  current_day  purchases  unique_invoices  \
0              1              1            1        0.0              0.0
1              2              1            2        0.0              0.0
2              3              1            3        0.0              0.0
3              4              1            4        0.0              0.0
4              5              1            5        0.0              0.0
..           ...            ...          ...        ...              ...
360          362             12           27        0.0              0.0
361          363             12           28        0.0              0.0
362          364             12           29        0.0              0.0
363          365             12           30        0.0              0.0
364          366             12           31        0.0              0.0

     unique_streams  total_views  current_revenue  next_month_revenue  \
0               0.0          0.0              0.0             438.785
1               0.0          0.0              0.0             438.785
2               0.0          0.0              0.0             438.785
3               0.0          0.0              0.0             438.785
```

```
4               0.0          0.0              0.0             461.920
..              ...          ...              ...                 ...
360             0.0          0.0              0.0             378.965
361             0.0          0.0              0.0             390.535
362             0.0          0.0              0.0             417.310
363             0.0          0.0              0.0             417.310
364             0.0          0.0              0.0             438.785

     next_month  next_day
0             2         1
1             2         2
2             2         3
3             2         4
4             2         5
..          ...       ...
360           1        27
361           1        28
362           1        29
363           1        30
364           1        31

[365 rows x 11 columns]
7 - Dataframe df_ts_norway data:
     day_of_year  current_month  current_day  purchases  unique_invoices  \
0              1              1            1        0.0              0.0
1              2              1            2        0.0              0.0
2              3              1            3        0.0              0.0
3              4              1            4        0.0              0.0
4              5              1            5        0.0              0.0
..           ...            ...          ...        ...              ...
360          362             12           27        0.0              0.0
361          363             12           28        0.0              0.0
362          364             12           29        0.0              0.0
363          365             12           30        0.0              0.0
364          366             12           31        0.0              0.0

     unique_streams  total_views  current_revenue  next_month_revenue  \
0               0.0          0.0              0.0                32.65
1               0.0          0.0              0.0                32.65
2               0.0          0.0              0.0                32.65
3               0.0          0.0              0.0                32.65
4               0.0          0.0              0.0                32.65
..              ...          ...              ...                  ...
360             0.0          0.0              0.0                 0.00
361             0.0          0.0              0.0                 0.00
362             0.0          0.0              0.0                 0.00
363             0.0          0.0              0.0                32.65
364             0.0          0.0              0.0                32.65

     next_month  next_day
0             2         1
1             2         2
2             2         3
3             2         4
4             2         5
..          ...       ...
```

```
360           1        27
361           1        28
362           1        29
363           1        30
364           1        31

[365 rows x 11 columns]
8 - Dataframe df_ts_portugal data:
     day_of_year  current_month  current_day  purchases  unique_invoices  \
0              1              1            1        0.0              0.0
1              2              1            2        2.0              0.5
2              3              1            3        0.0              0.0
3              4              1            4        0.0              0.0
4              5              1            5        0.0              0.0
..           ...            ...          ...        ...              ...
360          362             12           27        0.0              0.0
361          363             12           28        0.0              0.0
362          364             12           29        0.0              0.0
363          365             12           30        0.0              0.0
364          366             12           31        0.0              0.0

     unique_streams  total_views  current_revenue  next_month_revenue  \
0               0.0          0.0              0.0             319.875
1               2.0          1.0              6.7             319.875
2               0.0          0.0              0.0             313.175
3               0.0          0.0              0.0             313.175
4               0.0          0.0              0.0             313.175
..              ...          ...              ...                 ...
360             0.0          0.0              0.0             278.090
361             0.0          0.0              0.0             278.090
362             0.0          0.0              0.0             319.875
363             0.0          0.0              0.0             319.875
364             0.0          0.0              0.0             319.875

     next_month  next_day
0             2         1
1             2         2
2             2         3
3             2         4
4             2         5
..          ...       ...
360           1        27
361           1        28
362           1        29
363           1        30
364           1        31

[365 rows x 11 columns]
9 - Dataframe df_ts_singapore data:
     day_of_year  current_month  current_day  purchases  unique_invoices  \
0              1              1            1        0.0              0.0
1              2              1            2        0.0              0.0
2              3              1            3        0.0              0.0
3              4              1            4        0.0              0.0
4              5              1            5        0.0              0.0
..           ...            ...          ...        ...              ...
```

```
360          362             12           27        0.0              0.0
361          363             12           28        0.0              0.0
362          364             12           29        0.0              0.0
363          365             12           30        0.0              0.0
364          366             12           31        0.0              0.0

     unique_streams  total_views  current_revenue  next_month_revenue  \
0               0.0          0.0              0.0              263.34
1               0.0          0.0              0.0              263.34
2               0.0          0.0              0.0              263.34
3               0.0          0.0              0.0              263.34
4               0.0          0.0              0.0              263.34
..              ...          ...              ...                 ...
360             0.0          0.0              0.0              263.34
361             0.0          0.0              0.0              263.34
362             0.0          0.0              0.0              263.34
363             0.0          0.0              0.0              263.34
364             0.0          0.0              0.0              263.34

     next_month  next_day
0             2         1
1             2         2
2             2         3
3             2         4
4             2         5
..          ...       ...
360           1        27
361           1        28
362           1        29
363           1        30
364           1        31

[365 rows x 11 columns]
10 - Dataframe df_ts_spain data:
     day_of_year  current_month  current_day  purchases  unique_invoices  \
0              1              1            1        0.0              0.0
1              2              1            2        0.0              0.0
2              3              1            3        0.5              0.5
3              4              1            4       49.0              0.5
4              5              1            5        0.0              0.0
..           ...            ...          ...        ...              ...
360          362             12           27        0.0              0.0
361          363             12           28        0.0              0.0
362          364             12           29        0.0              0.0
363          365             12           30        0.0              0.0
364          366             12           31        0.0              0.0

     unique_streams  total_views  current_revenue  next_month_revenue  \
0               0.0          0.0             0.00            4030.310
1               0.0          0.0             0.00            4030.310
2               0.5          1.5           563.00            4030.310
3              49.0        268.0           143.32            3467.310
4               0.0          0.0             0.00            3323.990
..              ...          ...              ...                 ...
360             0.0          0.0             0.00            3863.260
361             0.0          0.0             0.00            3863.260
```

```
362             0.0          0.0             0.00           3946.525
363             0.0          0.0             0.00           3946.525
364             0.0          0.0             0.00           4030.310

     next_month  next_day
0             2         1
1             2         2
2             2         3
3             2         4
4             2         5
..          ...       ...
360           1        27
361           1        28
362           1        29
363           1        30
364           1        31

[365 rows x 11 columns]
11 - Dataframe df_ts_united_kingdom data:
     day_of_year  current_month  current_day  purchases  unique_invoices  \
0              1              1            1      908.5             61.5
1              2              1            2     1277.5             55.5
2              3              1            3     1008.5             41.5
3              4              1            4     1136.5             64.5
4              5              1            5      263.0             20.5
..           ...            ...          ...        ...              ...
360          362             12           27        0.0              0.0
361          363             12           28        0.0              0.0
362          364             12           29        0.0              0.0
363          365             12           30        0.0              0.0
364          366             12           31        0.0              0.0

     unique_streams  total_views  current_revenue  next_month_revenue  \
0             674.0       4224.0         3054.915         168776.7870
1             804.0       6649.5        20895.180         167800.7520
2             617.5       5639.0         3709.820         147300.1770
3             744.0       6716.0         3778.740         146777.6420
4             190.0       1892.5          934.760         148002.4470
..              ...          ...              ...                ...
360             0.0          0.0            0.000         145408.2760
361             0.0          0.0            0.000         149842.3560
362             0.0          0.0            0.000         154971.1015
363             0.0          0.0            0.000         159256.1515
364             0.0          0.0            0.000         163649.0620

     next_month  next_day
0             2         1
1             2         2
2             2         3
3             2         4
4             2         5
..          ...       ...
360           1        27
361           1        28
362           1        29
363           1        30
```

```
364           1        31

[365 rows x 11 columns]
```

3.1.-Split data into predictors (X) and target (y)

In [30]:
```python
# Split the data into features and target
# X = X[df_name] and y = [df_name]
X = {}
y = {}
for i in range (0,size):
    df_name = list_ts_df[i]
    X[df_name], y[df_name] = model.split_into_predictors_targets(library_ts_df_dayo
    print(i+1, '- Dataframe', df_name , 'was successfully splitted')
    print("SHAPE OF THE DATA: ")
    print("Number of (rows, column) X : " + str((X[df_name]).shape))
    print("Number of (row, columns) y : " + str((y[df_name]).shape))
    print('\n')
```

```
1 - Dataframe df_ts_all was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 1)
Number of (row, columns) y : (365, 6)


2 - Dataframe df_ts_eire was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 1)
Number of (row, columns) y : (365, 6)


3 - Dataframe df_ts_france was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 1)
Number of (row, columns) y : (365, 6)


4 - Dataframe df_ts_germany was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 1)
Number of (row, columns) y : (365, 6)


5 - Dataframe df_ts_hong_kong was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 1)
Number of (row, columns) y : (365, 6)


6 - Dataframe df_ts_netherlands was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 1)
Number of (row, columns) y : (365, 6)


7 - Dataframe df_ts_norway was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 1)
Number of (row, columns) y : (365, 6)


8 - Dataframe df_ts_portugal was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 1)
Number of (row, columns) y : (365, 6)


9 - Dataframe df_ts_singapore was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 1)
Number of (row, columns) y : (365, 6)


10 - Dataframe df_ts_spain was successfully splitted
SHAPE OF THE DATA:
```

```
Number of (rows, column) X : (365, 1)
Number of (row, columns) y : (365, 6)


11 - Dataframe df_ts_united_kingdom was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 1)
Number of (row, columns) y : (365, 6)
```

3.2.-Split data into training (X_train, y_train) and test (X_test, y_test) sets

In [31]:

```
# Randomly split the data into 80% for training and 20% for test sets
# X_train=X_train[df_name] , y_train=y_train[df_name], X_test=X_test[df_name] and y
X_train, X_test , y_train , y_test = {}, {}, {}, {}
for i in range (0,size):
    df_name = list_ts_df[i]
    X_train[df_name], X_test[df_name], y_train[df_name], y_test[df_name] = model.sp
    print(i+1, '- Dataframe', df_name , 'was successfully splitted')
    print("SHAPE OF THE DATA: ")
    print("Number of (rows, column) Xtrain : " + str((X_train[df_name]).shape))
    print("Number of (row, columns) Xtest : " + str((X_test[df_name]).shape))
    print("Number of (rows, column) ytrain : " + str((y_train[df_name]).shape))
    print("Number of (row, columns) ytest : " + str((y_test[df_name]).shape))
    print('\n')
```

```
1 - Dataframe df_ts_all was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 1)
Number of (row, columns) Xtest : (37, 1)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


2 - Dataframe df_ts_eire was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 1)
Number of (row, columns) Xtest : (37, 1)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


3 - Dataframe df_ts_france was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 1)
Number of (row, columns) Xtest : (37, 1)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


4 - Dataframe df_ts_germany was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 1)
Number of (row, columns) Xtest : (37, 1)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


5 - Dataframe df_ts_hong_kong was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 1)
Number of (row, columns) Xtest : (37, 1)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


6 - Dataframe df_ts_netherlands was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 1)
Number of (row, columns) Xtest : (37, 1)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


7 - Dataframe df_ts_norway was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 1)
Number of (row, columns) Xtest : (37, 1)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)
```

```
8 - Dataframe df_ts_portugal was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 1)
Number of (row, columns) Xtest : (37, 1)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


9 - Dataframe df_ts_singapore was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 1)
Number of (row, columns) Xtest : (37, 1)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


10 - Dataframe df_ts_spain was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 1)
Number of (row, columns) Xtest : (37, 1)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)


11 - Dataframe df_ts_united_kingdom was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 1)
Number of (row, columns) Xtest : (37, 1)
Number of (rows, column) ytrain : (328, 6)
Number of (row, columns) ytest : (37, 6)
```

In [32]:
```python
#Sort in ascending to be able to plot appropiate time-series visualizationabs
for i in range (0,size):
    df_name = list_ts_df[i]
    X_train[df_name]=X_train[df_name].sort_index(ascending=True)
    X_test[df_name] =X_test [df_name].sort_index(ascending=True)
    y_train[df_name]=y_train[df_name].sort_index(ascending=True)
    y_test[df_name] =y_test [df_name].sort_index(ascending=True)
```

3.3.-Train Predict and Evaluate 3 models

In [33]:
```python
# Train, predict, and evaluate each model
y_pred_lr , y_pred_rf , y_pred_xgb , lr_model , rf_model, xgb_model = {}, {}, {}, {
for i in range (0,size):
    df_name = list_ts_df[i]
    print(i+1, '- The model of Dataframe', df_name , 'was successfully trained, use
    y_pred_lr[df_name], y_pred_rf[df_name], y_pred_xgb[df_name], lr_model[df_name],
    print('\n')
```

```
1 - The model of Dataframe df_ts_all was successfully trained, used to predict and e
valuated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 15055.39
Mean Squared Error (MSE) : 1302430798.37
Root Mean-Square Error (RMSE) : 36089.21
Coefficient of Determination (R2) : -0.49

Random Forest Model:
Mean Absolute Error (MAE): 18822.45
Mean Squared Error (MSE) : 2035952991.29
Root Mean-Square Error (RMSE) : 45121.54
Coefficient of Determination (R2) : -1.19

XGBoost Model:
Mean Absolute Error (MAE): 19009.75
Mean Squared Error (MSE) : 2106656384.00
Root Mean-Square Error (RMSE) : 45898.33
Coefficient of Determination (R2) : -0.88

Evaluation Summary :
                           MAE           MSE           RMSE        R2
Linear         15055.390531  1.302431e+09  36089.206120 -0.492575
Random Forest  18822.450697  2.035953e+09  45121.535782 -1.186454
XGBoost        19009.750000  2.106656e+09  45898.326593 -0.880224


2 - The model of Dataframe df_ts_eire was successfully trained, used to predict and
evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 883.28
Mean Squared Error (MSE) : 4184627.07
Root Mean-Square Error (RMSE) : 2045.64
Coefficient of Determination (R2) : -2.35

Random Forest Model:
Mean Absolute Error (MAE): 294.33
Mean Squared Error (MSE) : 761698.48
Root Mean-Square Error (RMSE) : 872.75
Coefficient of Determination (R2) : -0.28

XGBoost Model:
Mean Absolute Error (MAE): 293.39
Mean Squared Error (MSE) : 748813.50
Root Mean-Square Error (RMSE) : 865.34
Coefficient of Determination (R2) : -0.20

Evaluation Summary :
                         MAE           MSE         RMSE        R2
Linear         883.275591  4.184627e+06  2045.636104 -2.345484
Random Forest  294.333129  7.616985e+05   872.753390 -0.279377
XGBoost        293.390778  7.488135e+05   865.340107 -0.204127
```

3 - The model of Dataframe df_ts_france was successfully trained, used to predict an
d evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 231.92
Mean Squared Error (MSE) : 239370.81
Root Mean-Square Error (RMSE) : 489.26
Coefficient of Determination (R2) : -4.03

Random Forest Model:
Mean Absolute Error (MAE): 62.55
Mean Squared Error (MSE) : 14029.56
Root Mean-Square Error (RMSE) : 118.45
Coefficient of Determination (R2) : -0.02

XGBoost Model:
Mean Absolute Error (MAE): 70.20
Mean Squared Error (MSE) : 14870.68
Root Mean-Square Error (RMSE) : 121.95
Coefficient of Determination (R2) : -0.10

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	231.919844	239370.814685	489.255368	-4.030185
Random Forest	62.550072	14029.556787	118.446430	-0.021867
XGBoost	70.203972	14870.682617	121.945408	-0.102624

4 - The model of Dataframe df_ts_germany was successfully trained, used to predict a
nd evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 205.22
Mean Squared Error (MSE) : 171817.91
Root Mean-Square Error (RMSE) : 414.51
Coefficient of Determination (R2) : -1.31

Random Forest Model:
Mean Absolute Error (MAE): 130.90
Mean Squared Error (MSE) : 73043.25
Root Mean-Square Error (RMSE) : 270.27
Coefficient of Determination (R2) : -0.55

XGBoost Model:
Mean Absolute Error (MAE): 140.13
Mean Squared Error (MSE) : 81413.70
Root Mean-Square Error (RMSE) : 285.33
Coefficient of Determination (R2) : -0.52

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	205.219703	171817.908786	414.509238	-1.310125
Random Forest	130.895074	73043.251460	270.265150	-0.549001
XGBoost	140.129257	81413.695312	285.330852	-0.516521

5 - The model of Dataframe df_ts_hong_kong was successfully trained, used to predict and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 24.37
Mean Squared Error (MSE) : 3132.73
Root Mean-Square Error (RMSE) : 55.97
Coefficient of Determination (R2) : -0.36

Random Forest Model:
Mean Absolute Error (MAE): 8.31
Mean Squared Error (MSE) : 1392.83
Root Mean-Square Error (RMSE) : 37.32
Coefficient of Determination (R2) : 0.76

XGBoost Model:
Mean Absolute Error (MAE): 8.31
Mean Squared Error (MSE) : 1392.83
Root Mean-Square Error (RMSE) : 37.32
Coefficient of Determination (R2) : -0.07

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	24.372291	3132.732178	55.970815	-0.361235
Random Forest	8.307477	1392.831674	37.320660	0.762821
XGBoost	8.307540	1392.830688	37.320647	-0.070513

6 - The model of Dataframe df_ts_netherlands was successfully trained, used to predict and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 106.72
Mean Squared Error (MSE) : 45624.62
Root Mean-Square Error (RMSE) : 213.60
Coefficient of Determination (R2) : -6.35

Random Forest Model:
Mean Absolute Error (MAE): 27.69
Mean Squared Error (MSE) : 3887.62
Root Mean-Square Error (RMSE) : 62.35
Coefficient of Determination (R2) : -0.28

XGBoost Model:
Mean Absolute Error (MAE): 25.75
Mean Squared Error (MSE) : 3619.28
Root Mean-Square Error (RMSE) : 60.16
Coefficient of Determination (R2) : -0.32

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	106.724150	45624.620263	213.599205	-6.352660
Random Forest	27.693533	3887.618032	62.350766	-0.284840
XGBoost	25.745386	3619.278320	60.160438	-0.324969

7 - The model of Dataframe df_ts_norway was successfully trained, used to predict an
d evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 313.89
Mean Squared Error (MSE) : 522823.49
Root Mean-Square Error (RMSE) : 723.07
Coefficient of Determination (R2) : -47.82

Random Forest Model:
Mean Absolute Error (MAE): 59.12
Mean Squared Error (MSE) : 26108.38
Root Mean-Square Error (RMSE) : 161.58
Coefficient of Determination (R2) : -1.23

XGBoost Model:
Mean Absolute Error (MAE): 59.12
Mean Squared Error (MSE) : 26108.38
Root Mean-Square Error (RMSE) : 161.58
Coefficient of Determination (R2) : -1.23

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	313.891334	522823.491729	723.065344	-47.815285
Random Forest	59.117432	26108.382182	161.580884	-1.230998
XGBoost	59.117580	26108.375000	161.580862	-1.230891

8 - The model of Dataframe df_ts_portugal was successfully trained, used to predict
and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 63.10
Mean Squared Error (MSE) : 17619.00
Root Mean-Square Error (RMSE) : 132.74
Coefficient of Determination (R2) : -1.74

Random Forest Model:
Mean Absolute Error (MAE): 47.63
Mean Squared Error (MSE) : 10793.99
Root Mean-Square Error (RMSE) : 103.89
Coefficient of Determination (R2) : -0.93

XGBoost Model:
Mean Absolute Error (MAE): 46.43
Mean Squared Error (MSE) : 10604.10
Root Mean-Square Error (RMSE) : 102.98
Coefficient of Determination (R2) : -0.92

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	63.096018	17619.003219	132.736593	-1.735730
Random Forest	47.632565	10793.993899	103.894148	-0.933873
XGBoost	46.433033	10604.101562	102.976218	-0.922862

9 - The model of Dataframe df_ts_singapore was successfully trained, used to predict and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 92.25
Mean Squared Error (MSE) : 53774.51
Root Mean-Square Error (RMSE) : 231.89
Coefficient of Determination (R2) : -3.76

Random Forest Model:
Mean Absolute Error (MAE): 11.86
Mean Squared Error (MSE) : 3123.78
Root Mean-Square Error (RMSE) : 55.89
Coefficient of Determination (R2) : 0.77

XGBoost Model:
Mean Absolute Error (MAE): 11.86
Mean Squared Error (MSE) : 3123.78
Root Mean-Square Error (RMSE) : 55.89
Coefficient of Determination (R2) : -0.06

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	92.254237	53774.509888	231.893316	-3.763953
Random Forest	11.862162	3123.781784	55.890802	0.771605
XGBoost	11.862220	3123.781006	55.890795	-0.061728

10 - The model of Dataframe df_ts_spain was successfully trained, used to predict and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 365.77
Mean Squared Error (MSE) : 1003388.19
Root Mean-Square Error (RMSE) : 1001.69
Coefficient of Determination (R2) : -0.53

Random Forest Model:
Mean Absolute Error (MAE): 304.10
Mean Squared Error (MSE) : 750011.53
Root Mean-Square Error (RMSE) : 866.03
Coefficient of Determination (R2) : -0.34

XGBoost Model:
Mean Absolute Error (MAE): 306.10
Mean Squared Error (MSE) : 738516.94
Root Mean-Square Error (RMSE) : 859.37
Coefficient of Determination (R2) : -0.39

Evaluation Summary :

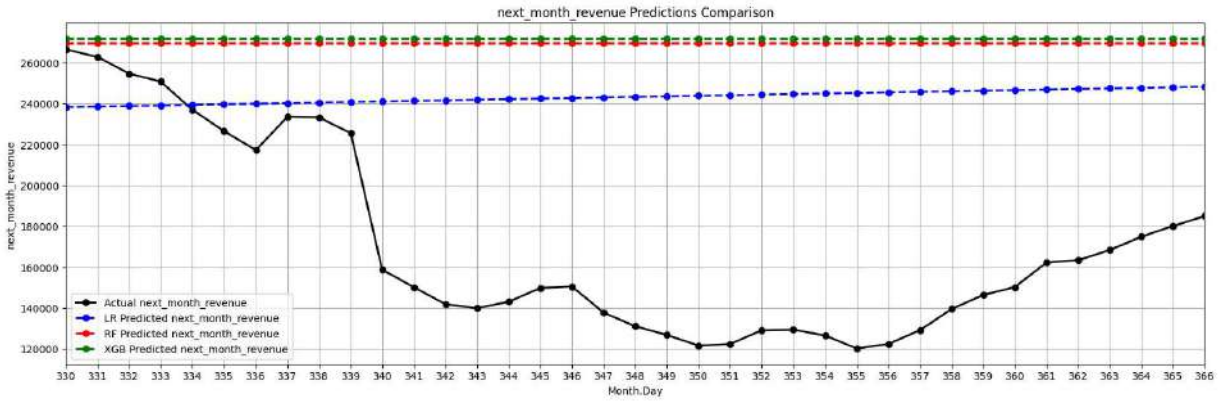
	MAE	MSE	RMSE	R2
Linear	365.767921	1.003388e+06	1001.692662	-0.534859
Random Forest	304.099005	7.500115e+05	866.032059	-0.338750
XGBoost	306.097595	7.385169e+05	859.370082	-0.394217

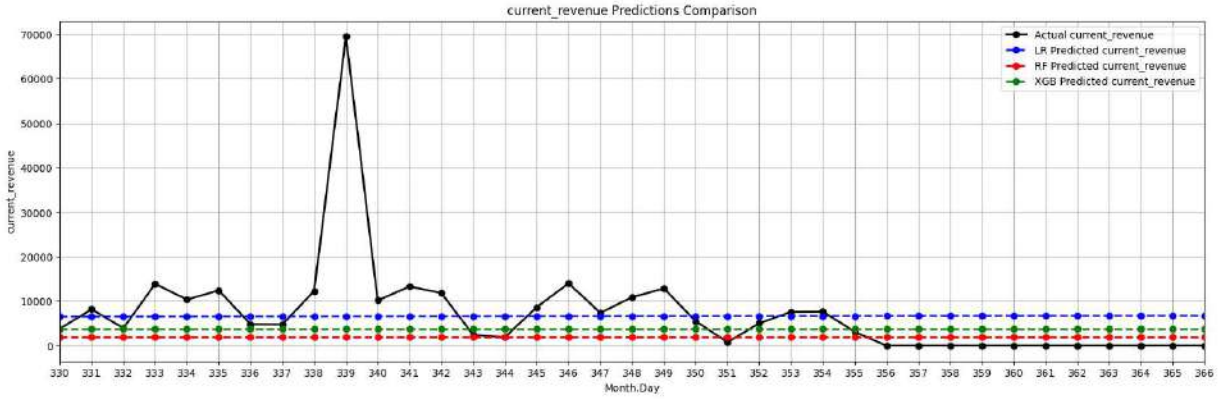
```
11 - The model of Dataframe df_ts_united_kingdom was successfully trained, used to p
redict and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 14207.85
Mean Squared Error (MSE) : 1152309026.68
Root Mean-Square Error (RMSE) : 33945.68
Coefficient of Determination (R2) : -0.38

Random Forest Model:
Mean Absolute Error (MAE): 18931.56
Mean Squared Error (MSE) : 2101484937.79
Root Mean-Square Error (RMSE) : 45841.96
Coefficient of Determination (R2) : -1.17

XGBoost Model:
Mean Absolute Error (MAE): 19098.79
Mean Squared Error (MSE) : 2164363776.00
Root Mean-Square Error (RMSE) : 46522.72
Coefficient of Determination (R2) : -0.86

Evaluation Summary :
                           MAE           MSE          RMSE        R2
Linear          14207.847684  1.152309e+09  33945.677585 -0.384215
Random Forest   18931.557344  2.101485e+09  45841.956086 -1.172768
XGBoost         19098.787109  2.164364e+09  46522.723222 -0.861873
```

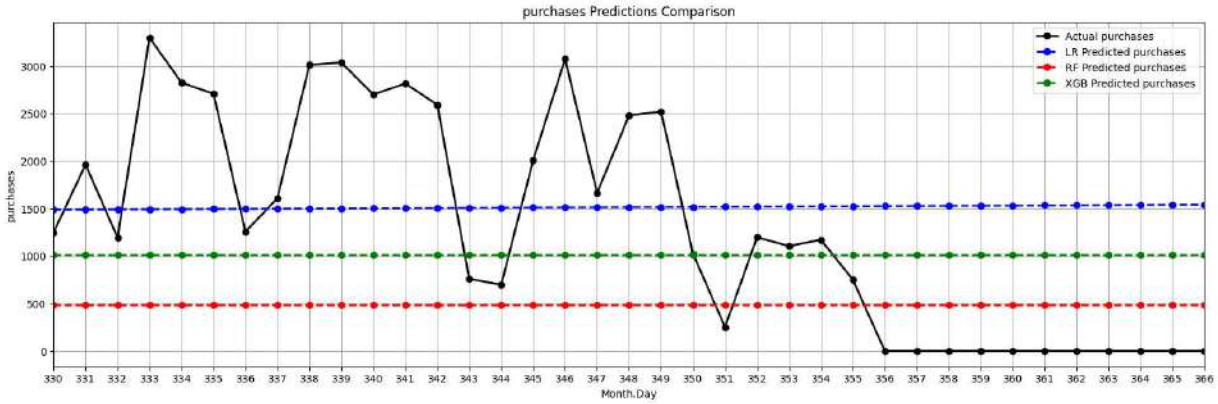
3.4.-Plot test data observed and predicted with the 3 models

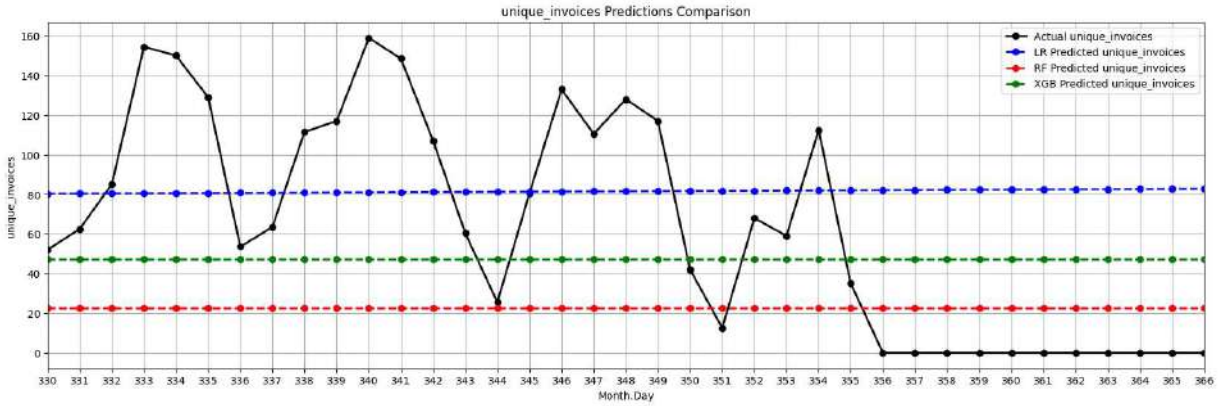
In [34]:
```
# Plot results for each target
for i in range (0,size):
    df_name = list_ts_df[i]
    print(i+1, '- Dataframe', df_name , 'data:')
    for target in y_test[df_name].columns:
        viz.plot_test_data(X_test[df_name]['day_of_year'], y_test[df_name], y_pred_
```

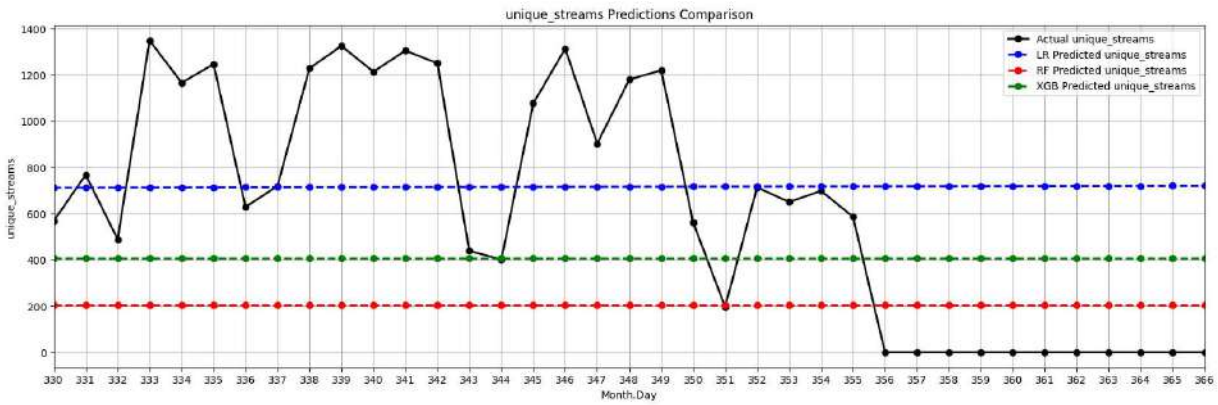
```
1 - Dataframe df_ts_all data:
```

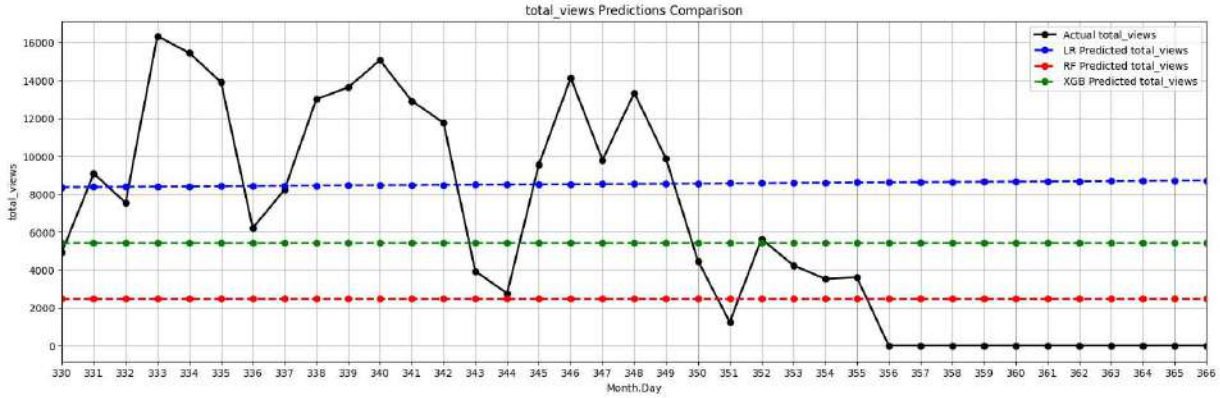



current_revenue Predictions Comparison
Actual current_revenue
LR Predicted current_revenue
RF Predicted current_revenue
XGB Predicted current_revenue
current_revenue
Month.Day

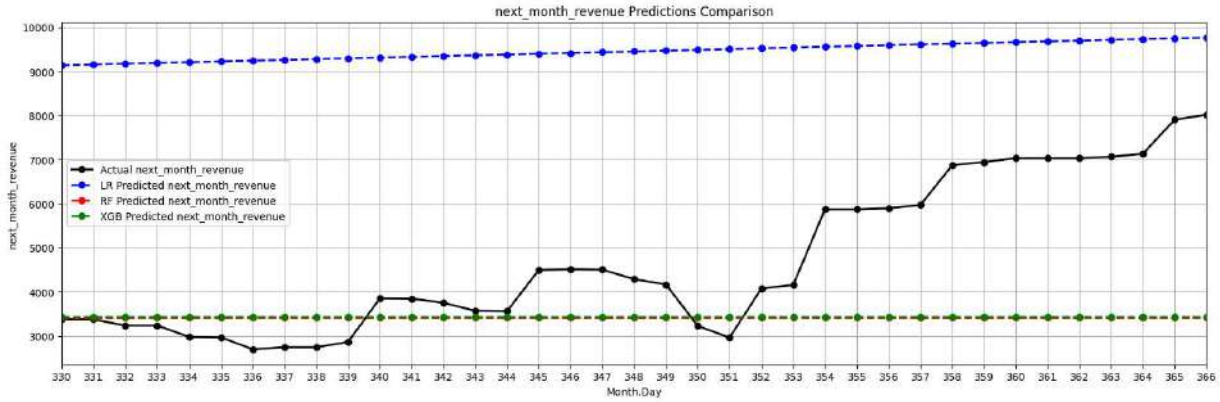

purchases Predictions Comparison
Actual purchases
LR Predicted purchases
RF Predicted purchases
XGB Predicted purchases
purchases
Month.Day

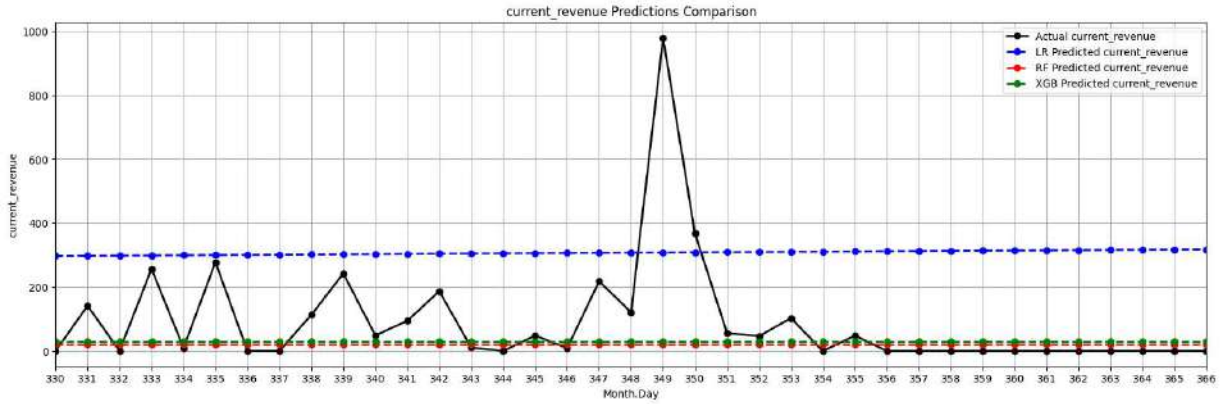

unique_invoices Predictions Comparison
Actual unique_invoices
LR Predicted unique_invoices
RF Predicted unique_invoices
XGB Predicted unique_invoices
unique_invoices
Month.Day

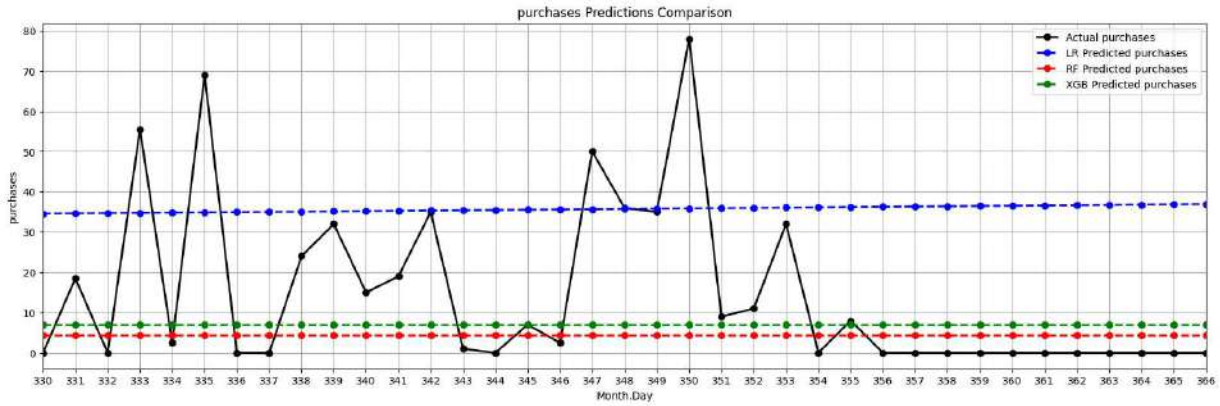

unique_streams Predictions Comparison
Actual unique_streams
LR Predicted unique_streams
RF Predicted unique_streams
XGB Predicted unique_streams
unique_streams
Month.Day

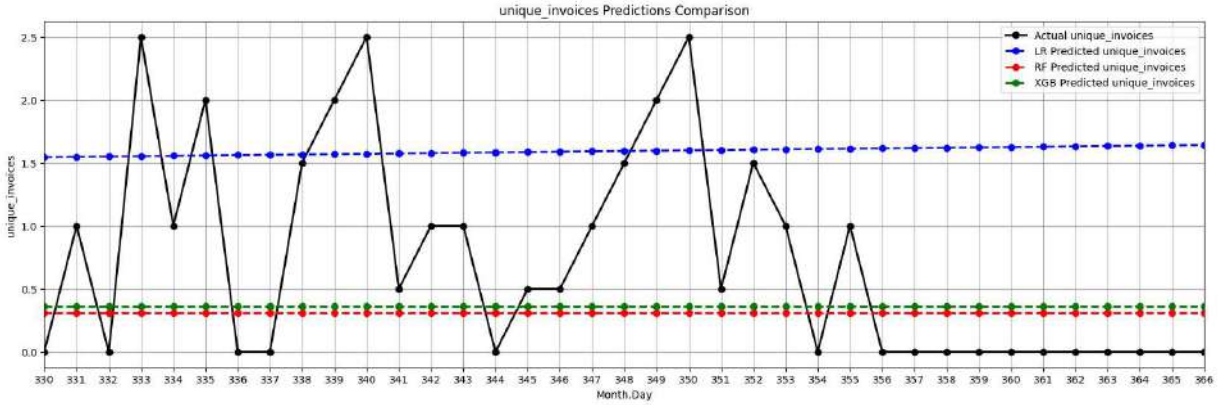


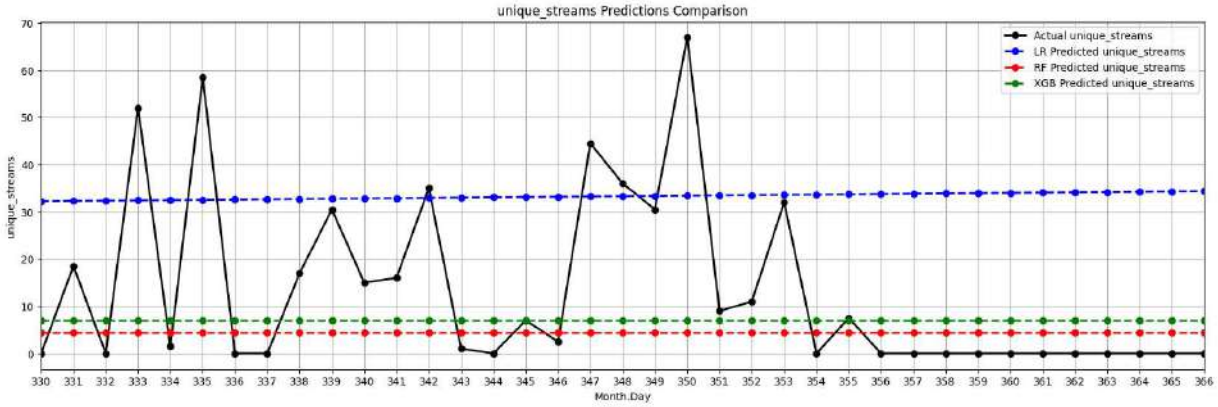
2 - Dataframe df_ts_eire data:

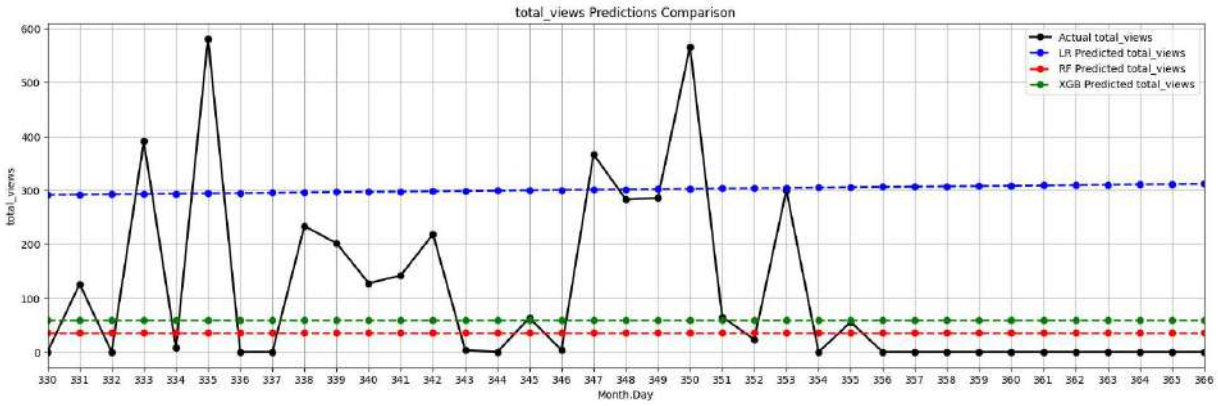




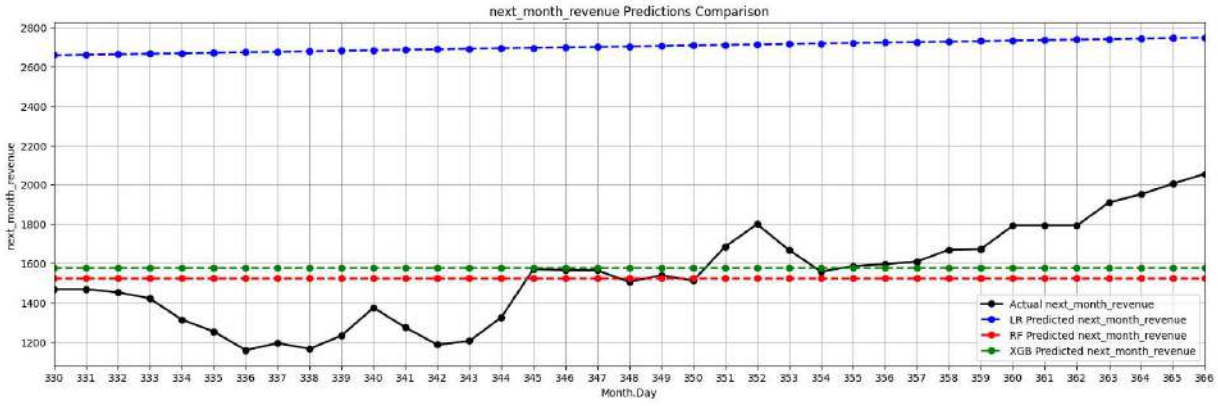


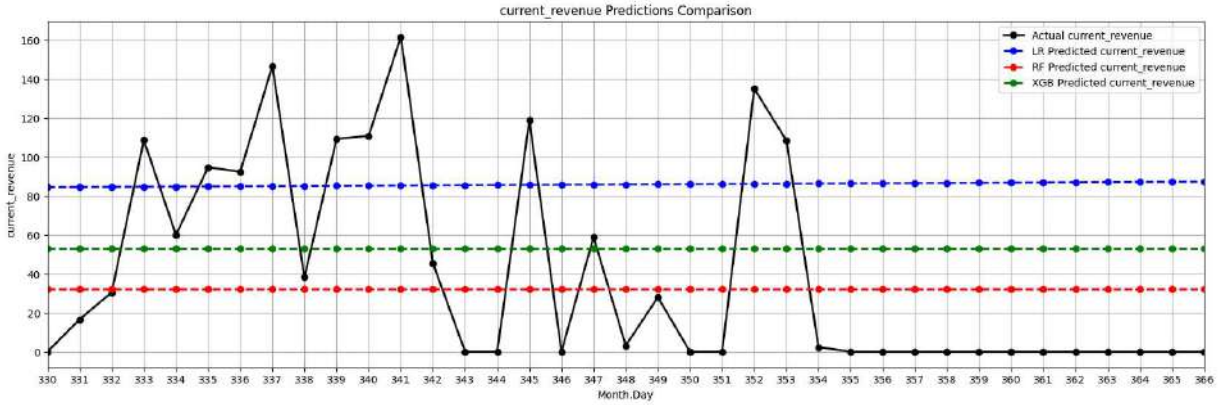


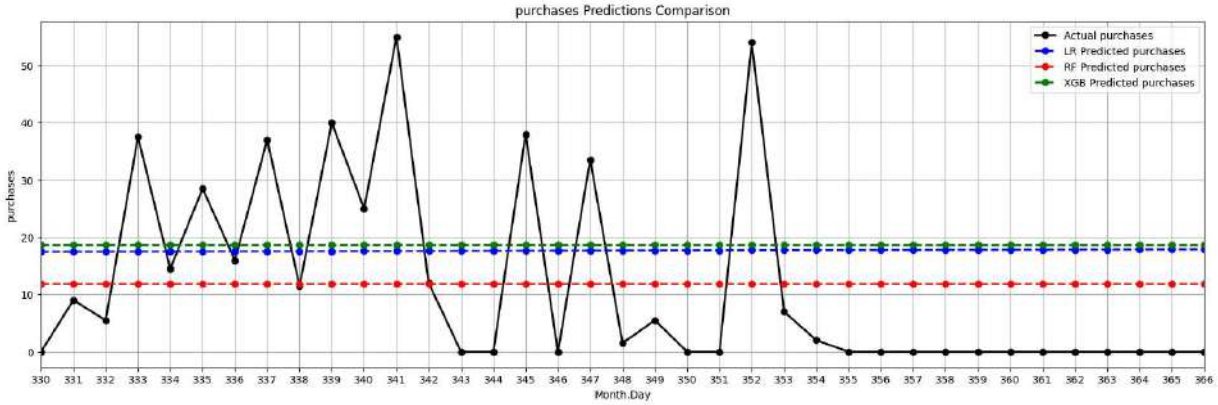


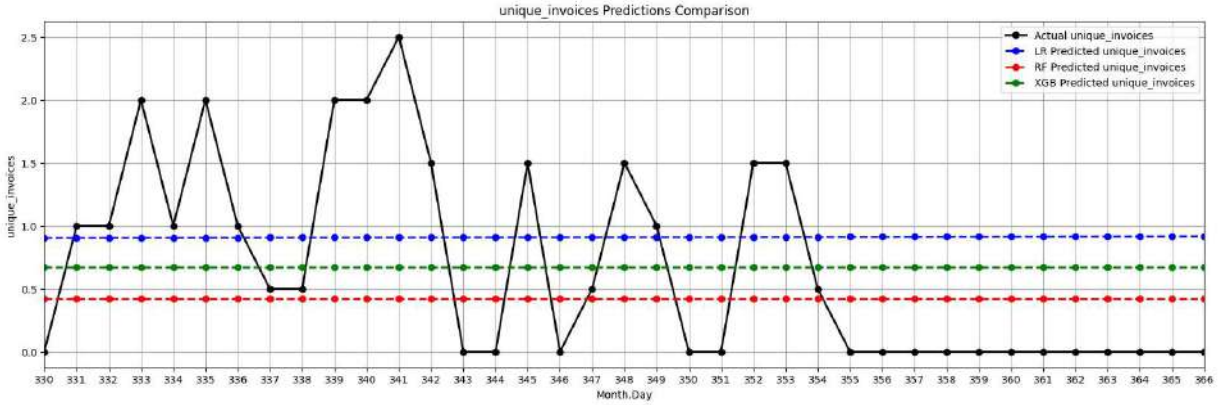


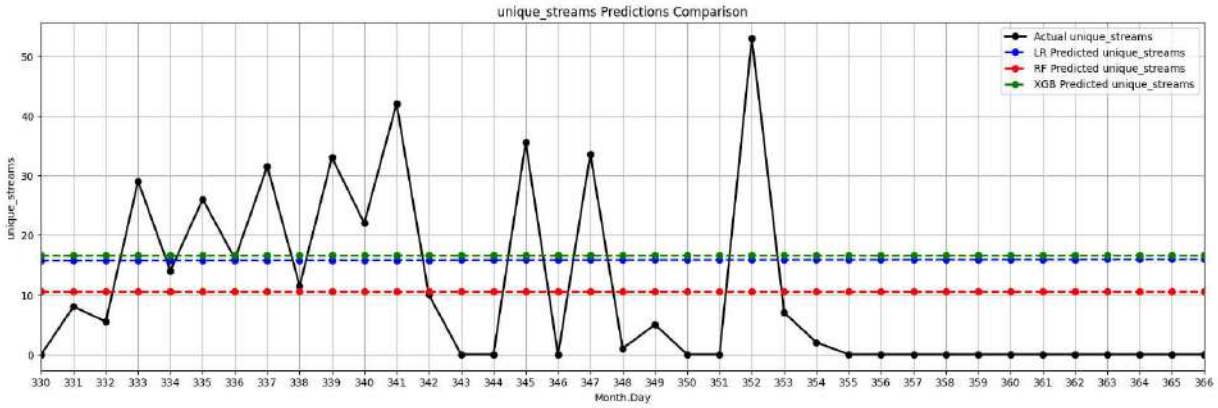
3 - Dataframe df_ts_france data:

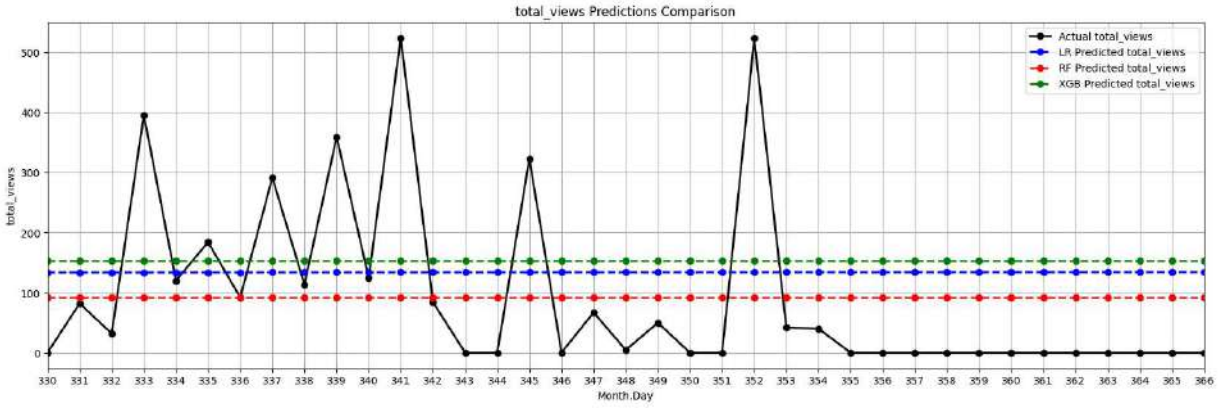



current_revenue Predictions Comparison
Actual current_revenue
LR Predicted current_revenue
RF Predicted current_revenue
XGB Predicted current_revenue
current_revenue
Month.Day

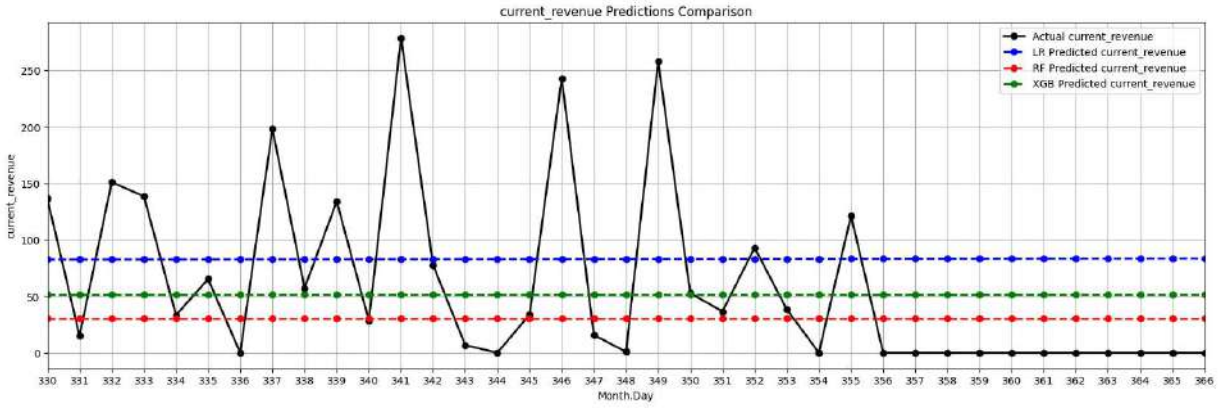

purchases Predictions Comparison
Actual purchases
LR Predicted purchases
RF Predicted purchases
XGB Predicted purchases
purchases
Month.Day

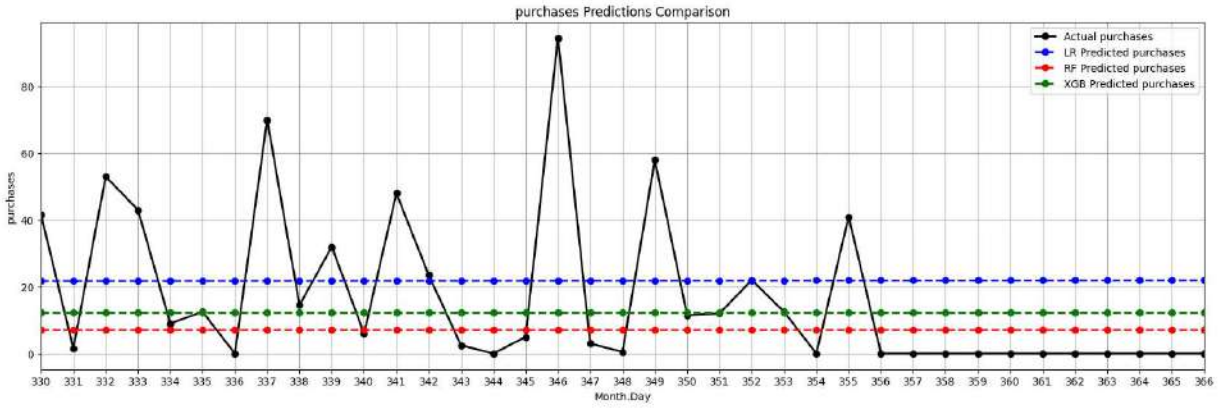

unique_invoices Predictions Comparison
Actual unique_invoices
LR Predicted unique_invoices
RF Predicted unique_invoices
XGB Predicted unique_invoices
unique_invoices
Month.Day

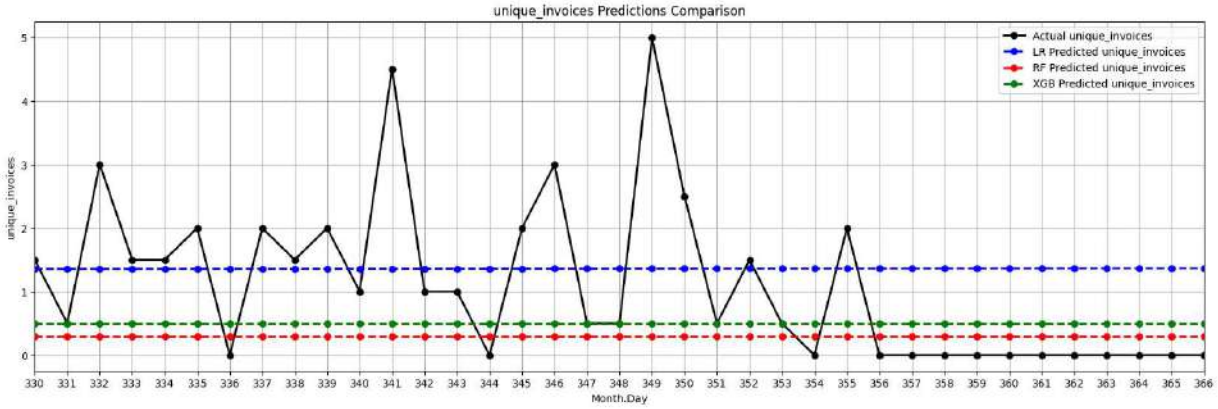

unique_streams Predictions Comparison
Actual unique_streams
LR Predicted unique_streams
RF Predicted unique_streams
XGB Predicted unique_streams
unique_streams
Month.Day

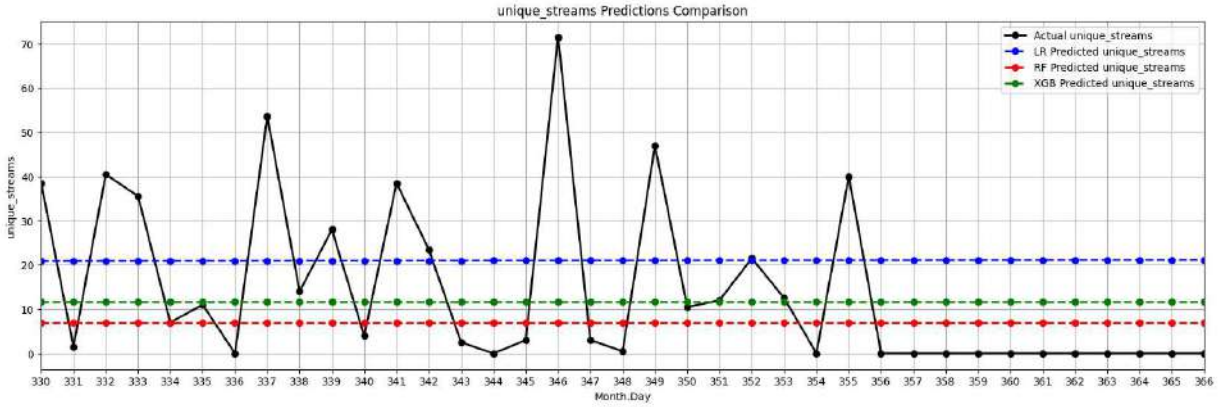


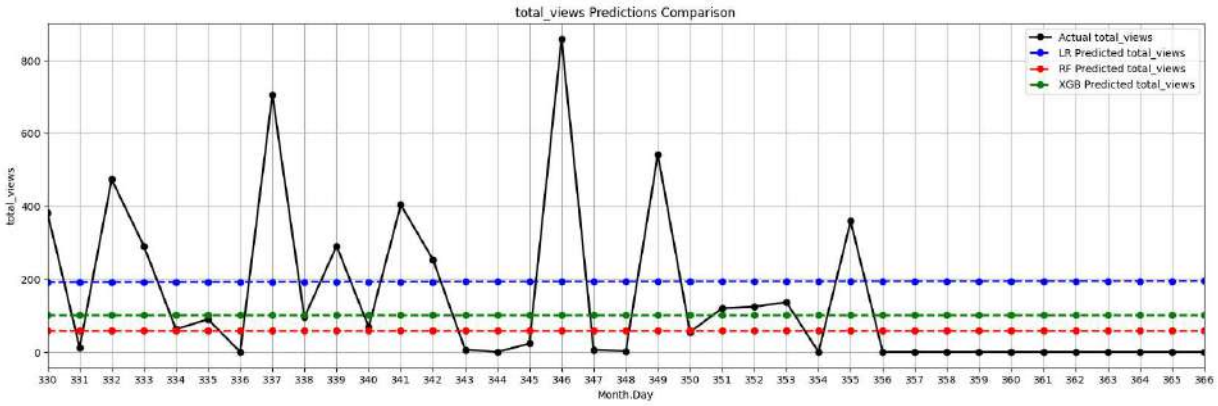
4 - Dataframe df_ts_germany data:



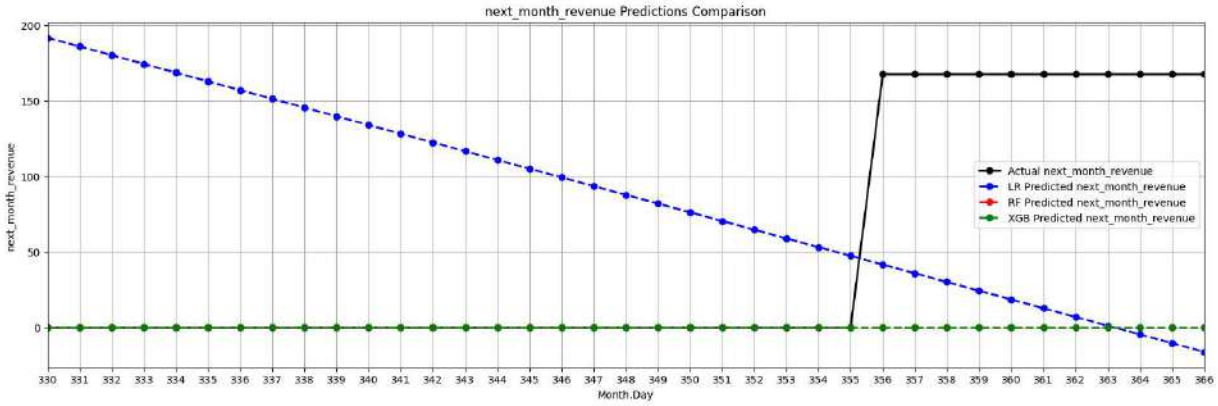


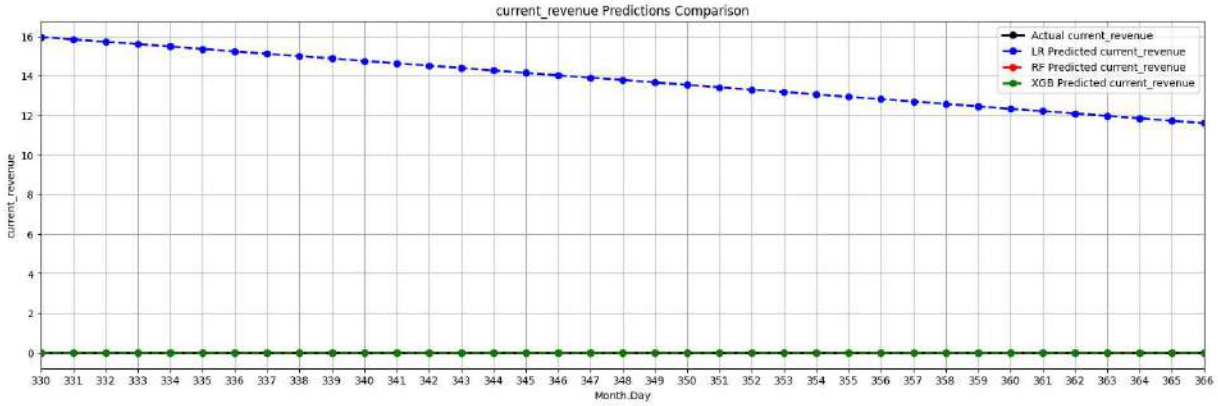


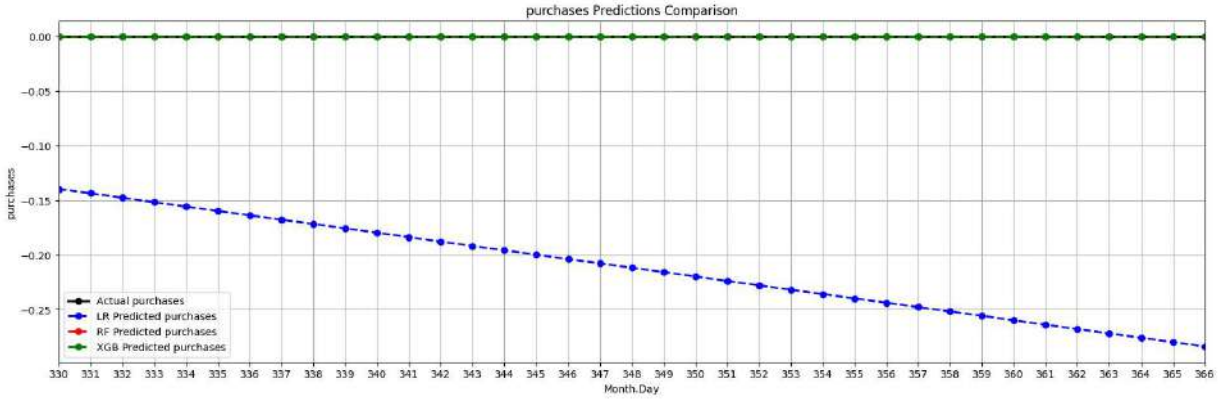


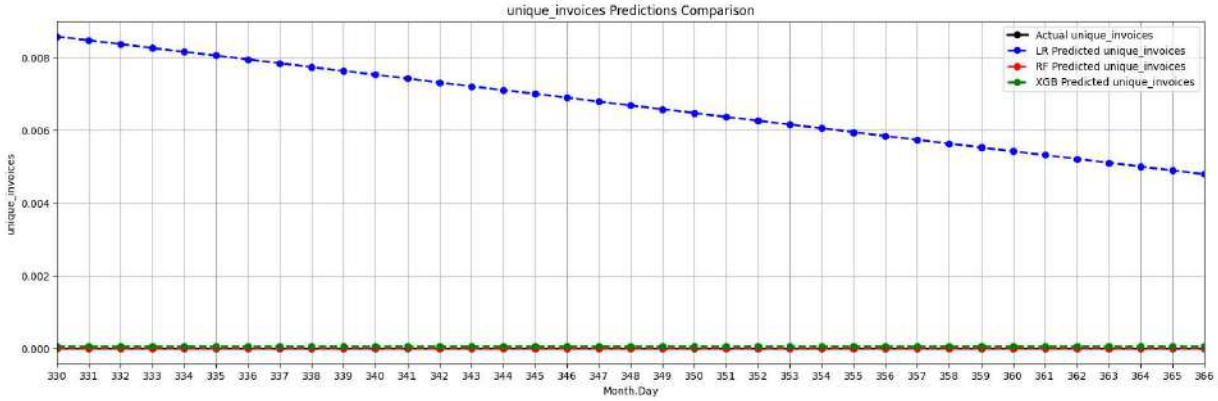


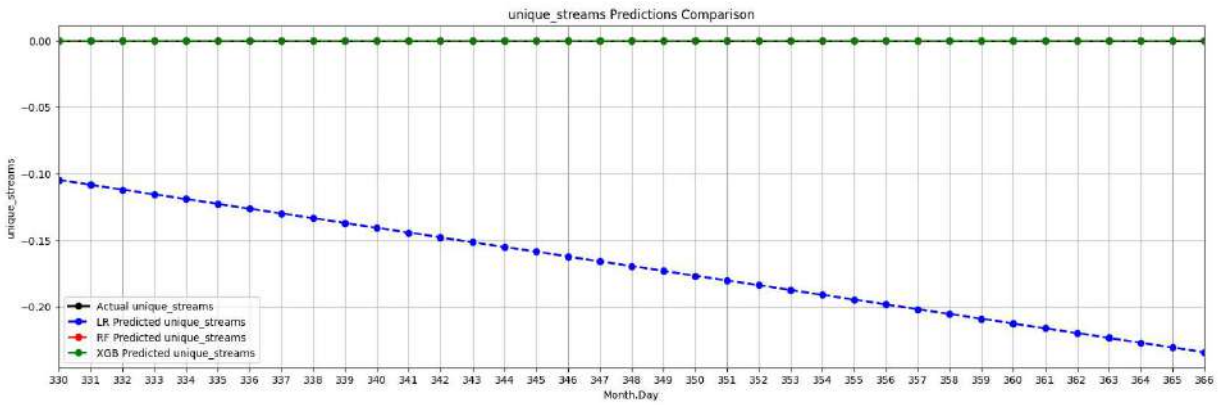
5 - Dataframe df_ts_hong_kong data:

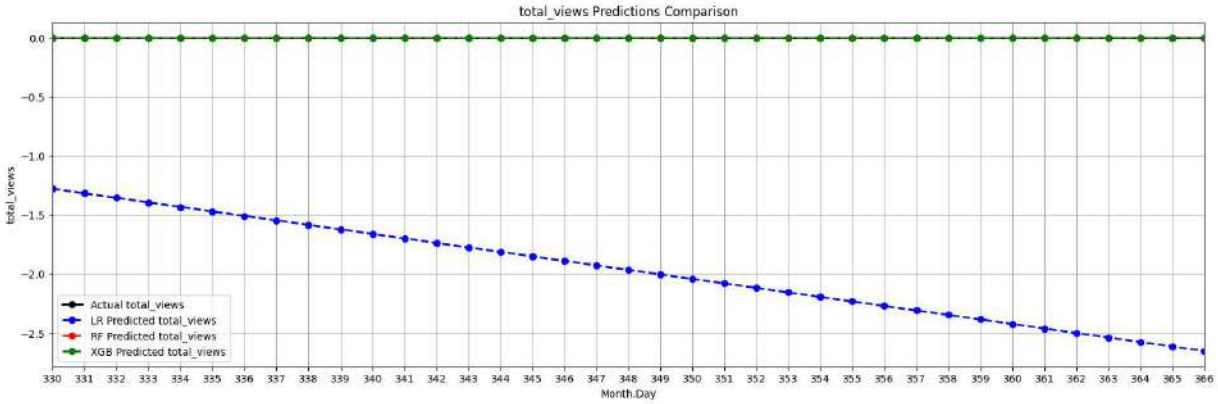



current_revenue Predictions Comparison
Actual current_revenue
LR Predicted current_revenue
RF Predicted current_revenue
XGB Predicted current_revenue
current_revenue
Month.Day

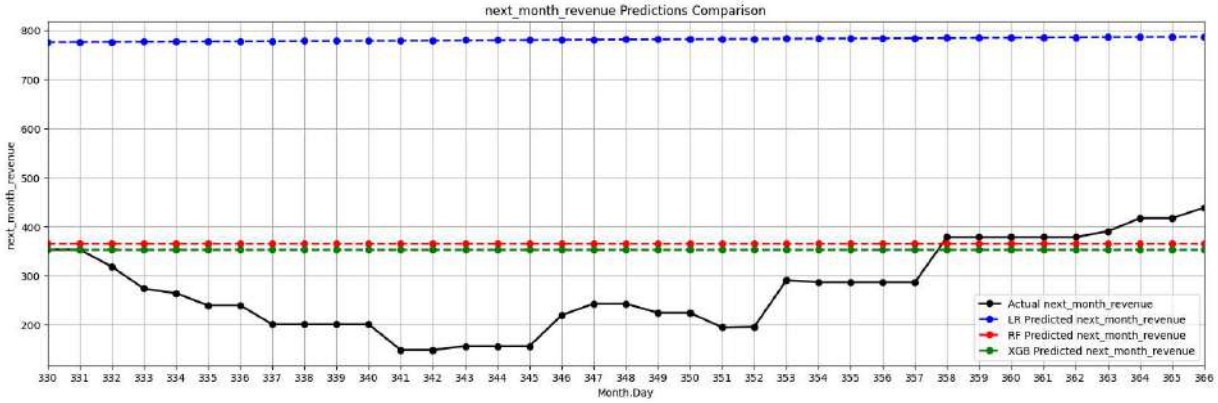

purchases Predictions Comparison
Actual purchases
LR Predicted purchases
RF Predicted purchases
XGB Predicted purchases
purchases
Month.Day

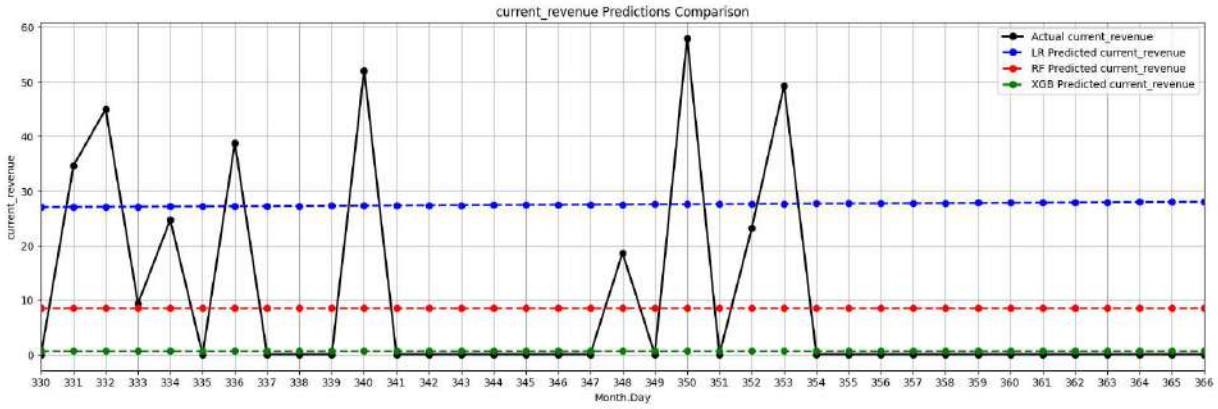

unique_invoices Predictions Comparison
Actual unique_invoices
LR Predicted unique_invoices
RF Predicted unique_invoices
XGB Predicted unique_invoices
unique_invoices
Month.Day

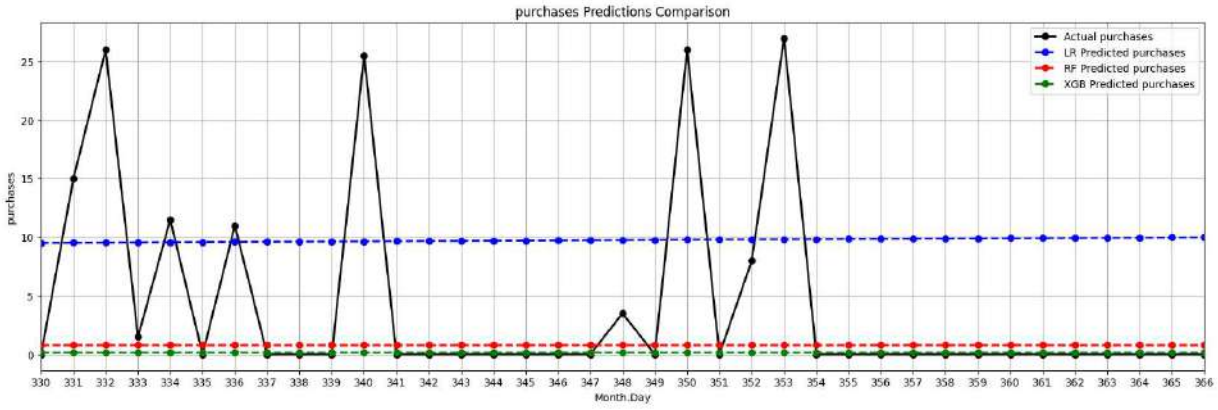

unique_streams Predictions Comparison
Actual unique_streams
LR Predicted unique_streams
RF Predicted unique_streams
XGB Predicted unique_streams
unique_streams
Month.Day

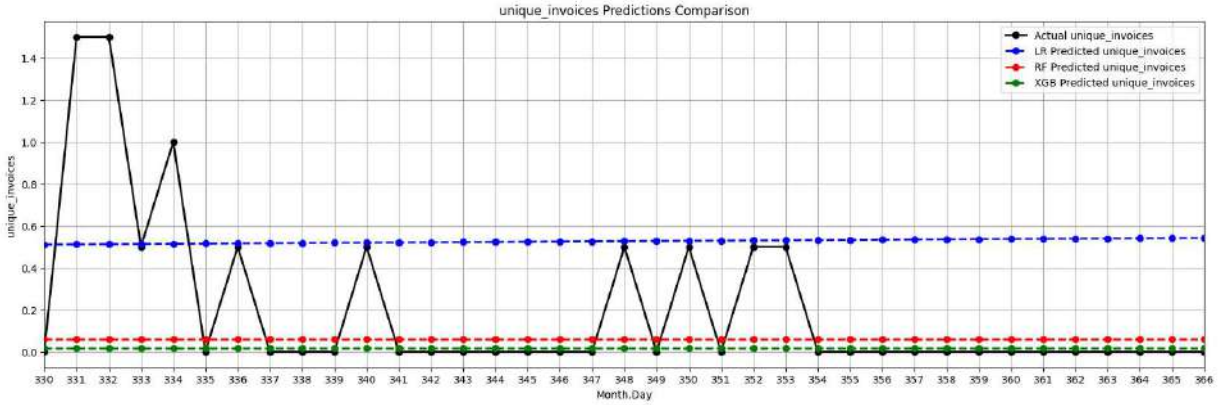


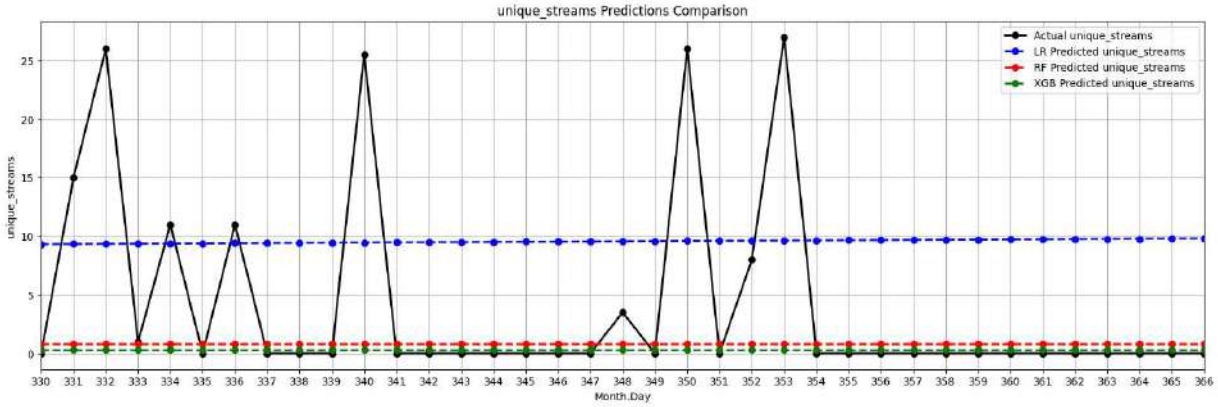
6 - Dataframe df_ts_netherlands data:

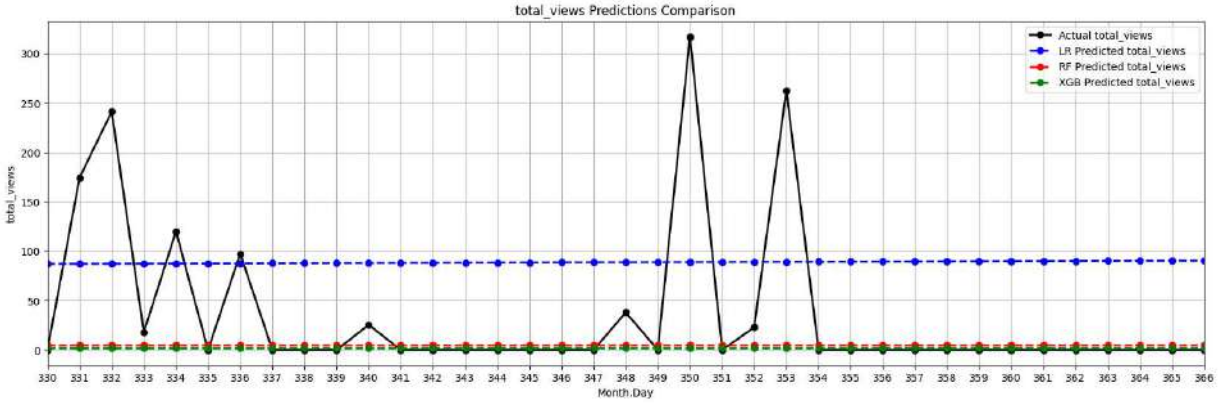




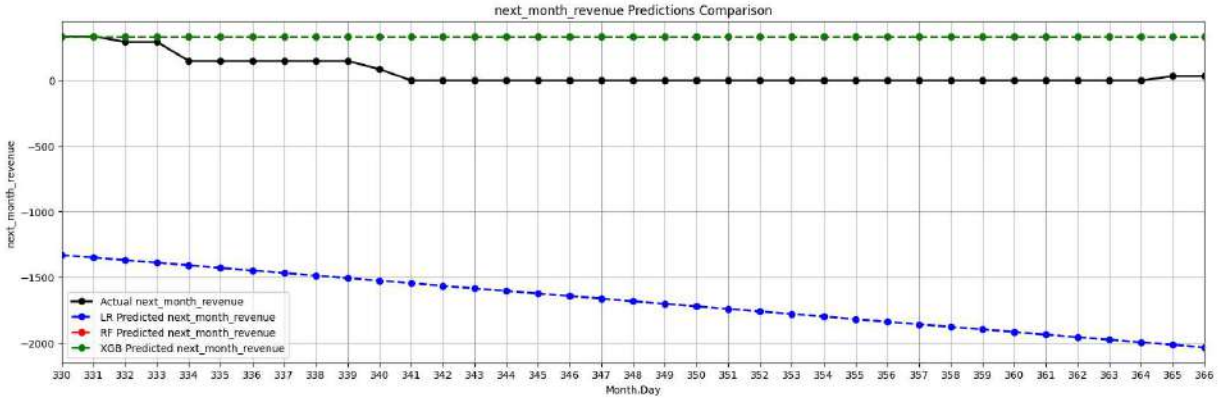


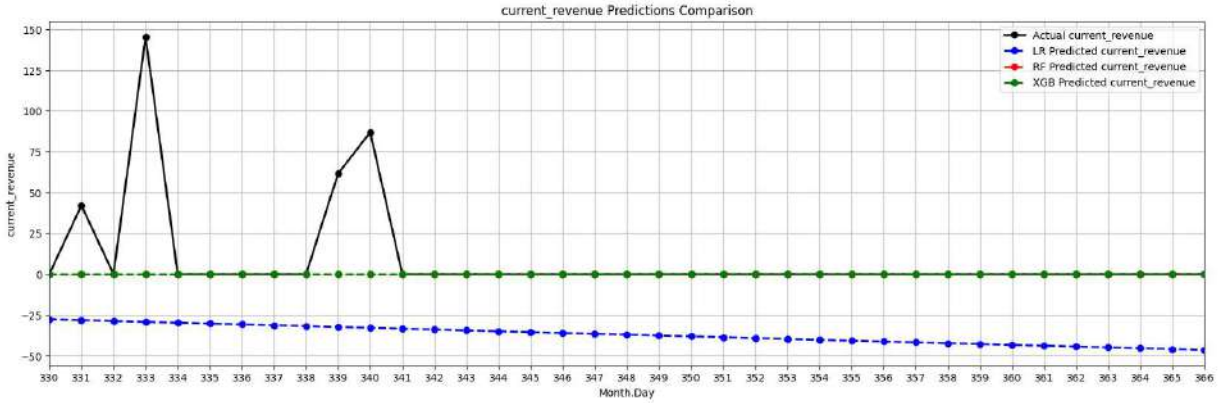


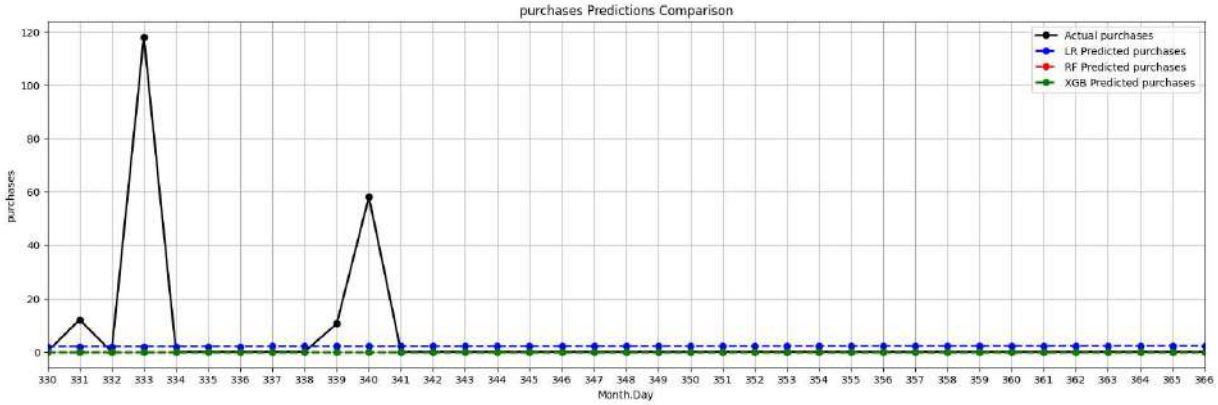


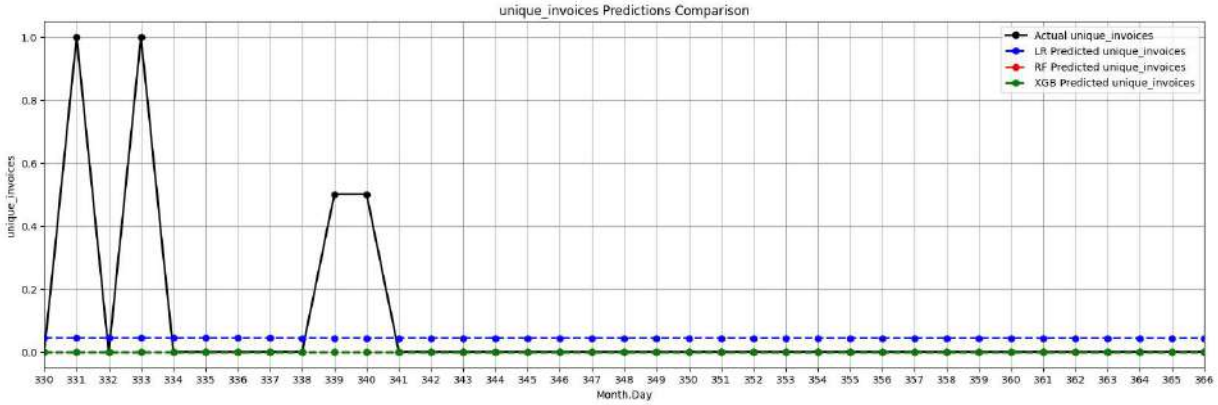


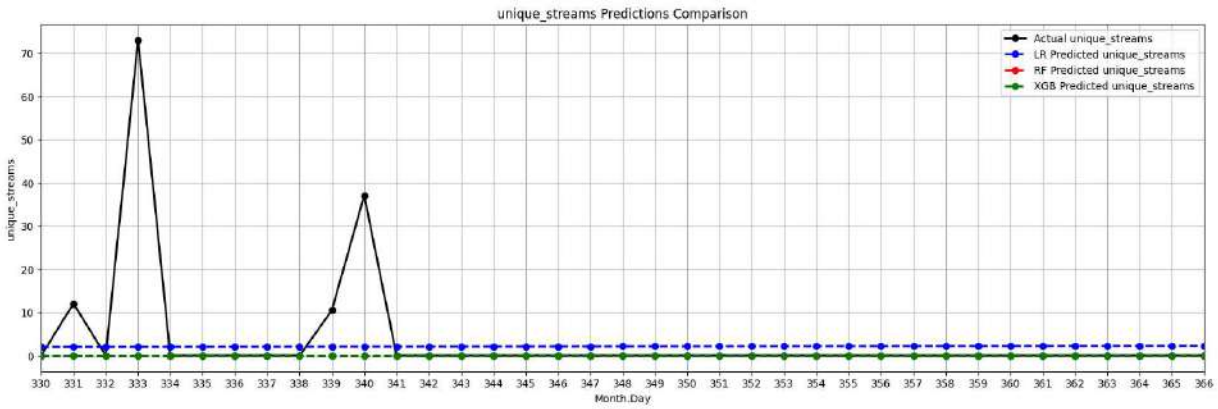
7 - Dataframe df_ts_norway data:

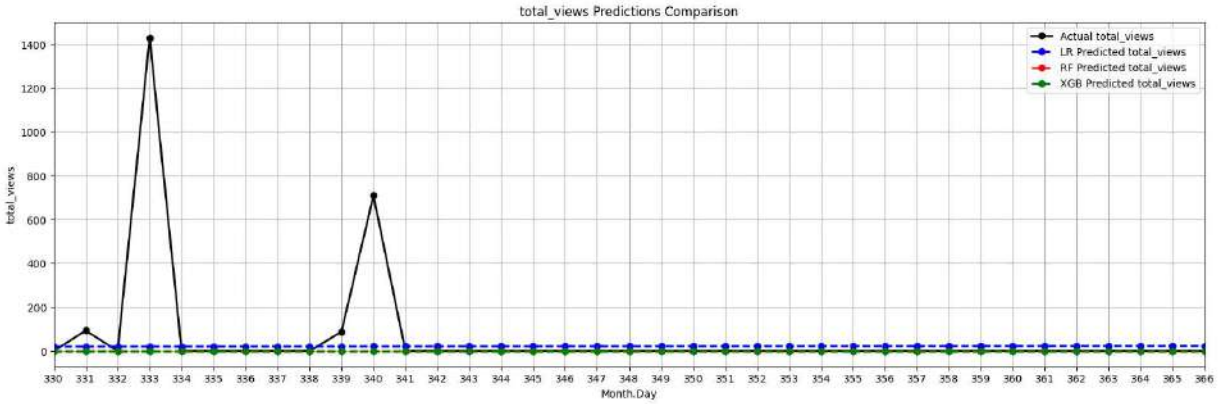



current_revenue Predictions Comparison
Actual current_revenue
LR Predicted current_revenue
RF Predicted current_revenue
XGB Predicted current_revenue
current_revenue
Month.Day

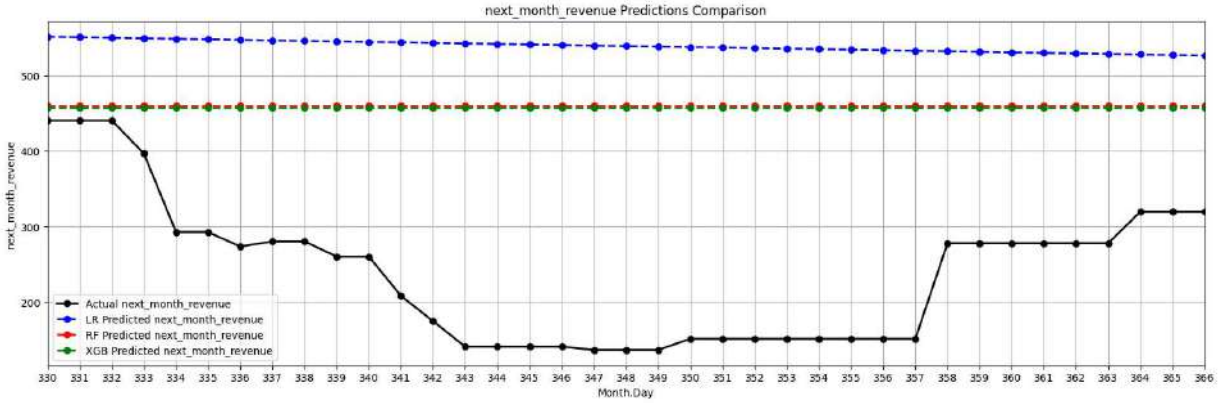

purchases Predictions Comparison
Actual purchases
LR Predicted purchases
RF Predicted purchases
XGB Predicted purchases
purchases
Month.Day

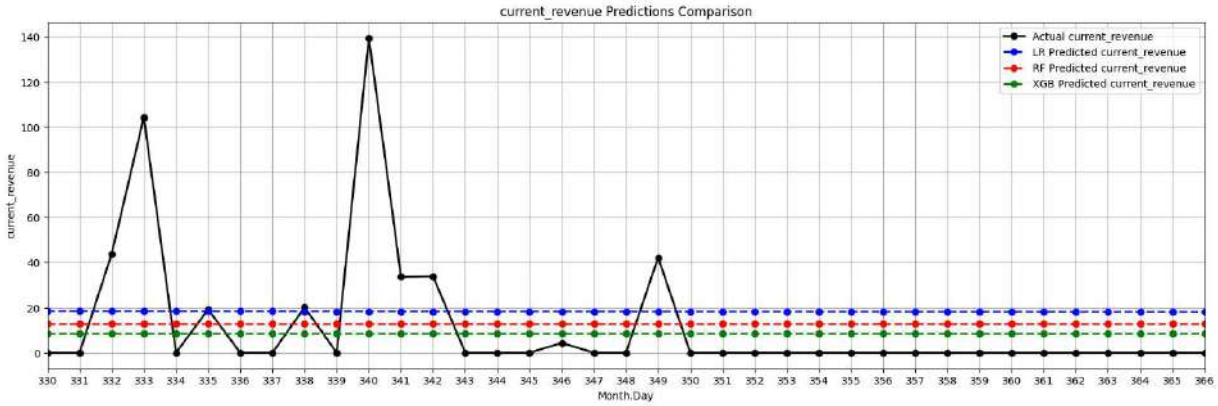

unique_invoices Predictions Comparison
Actual unique_invoices
LR Predicted unique_invoices
RF Predicted unique_invoices
XGB Predicted unique_invoices
unique_invoices
Month.Day

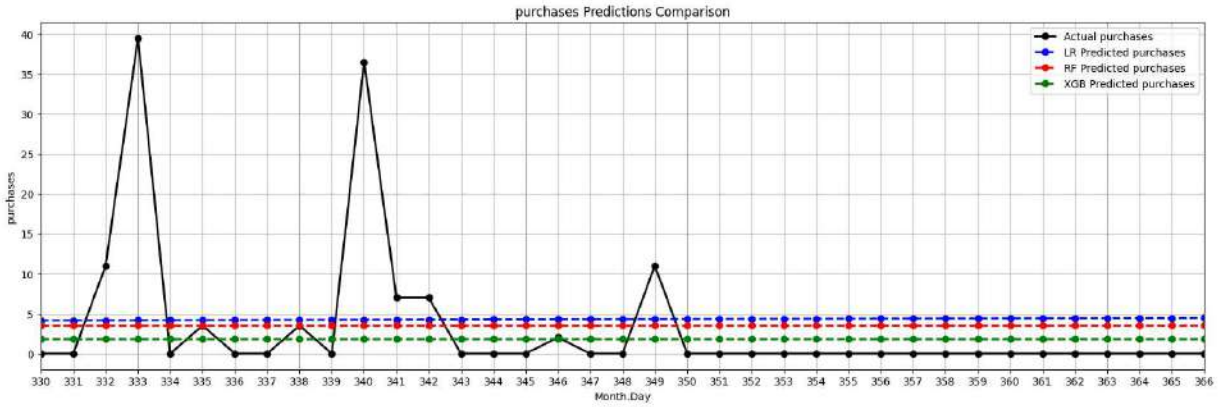

unique_streams Predictions Comparison
Actual unique_streams
LR Predicted unique_streams
RF Predicted unique_streams
XGB Predicted unique_streams
unique_streams
Month.Day

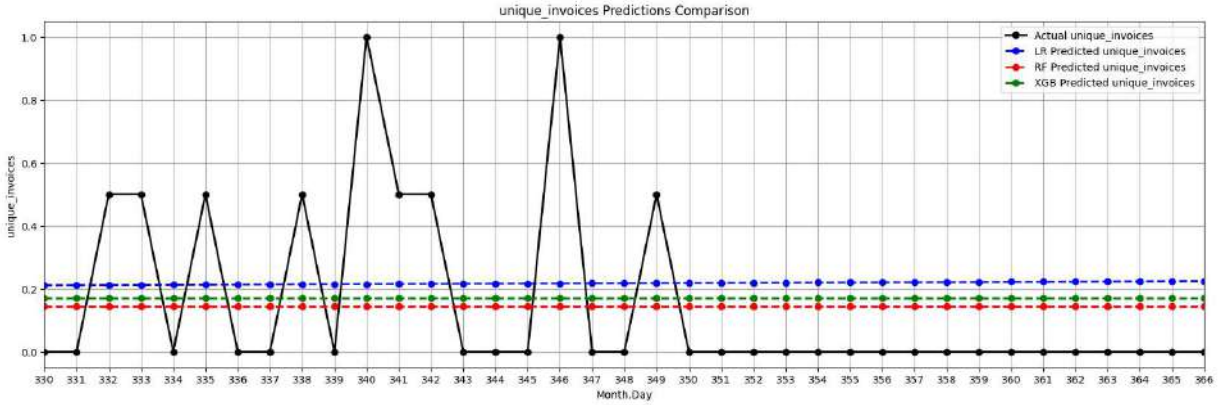


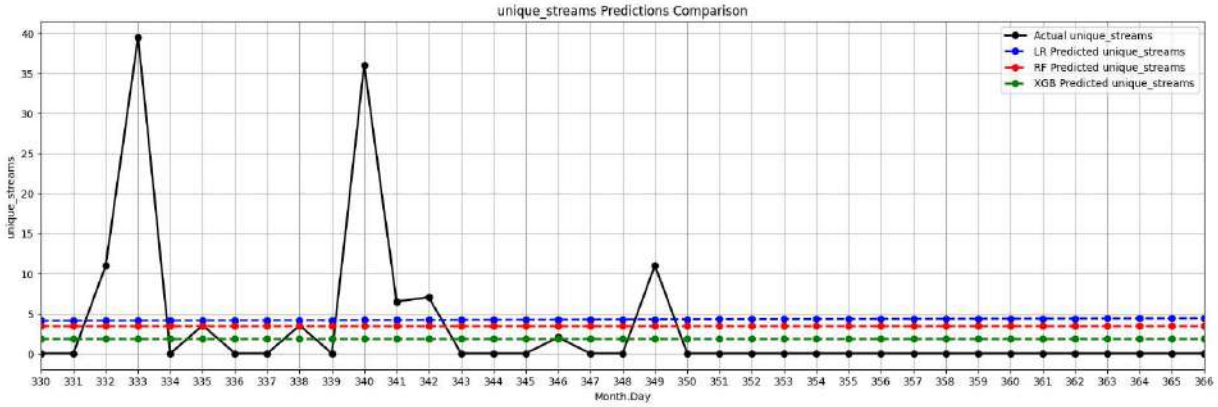
8 - Dataframe df_ts_portugal data:

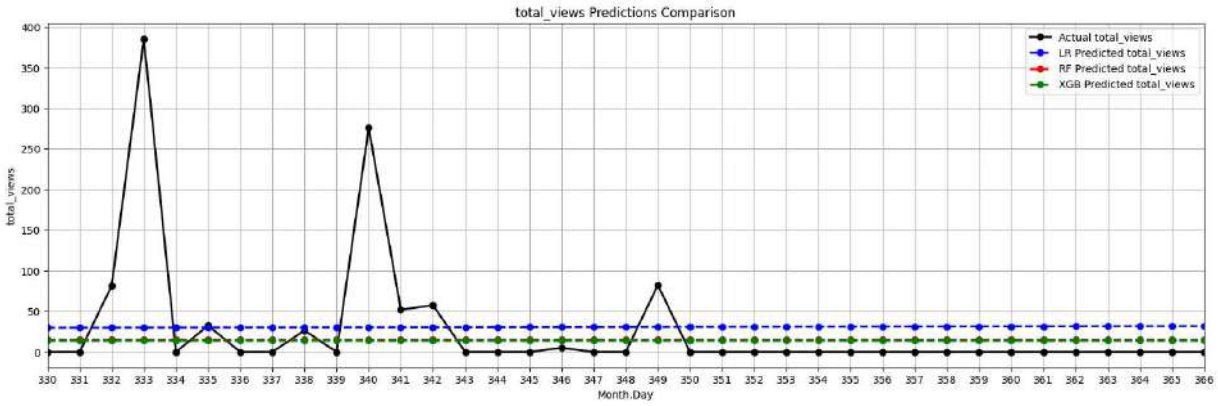




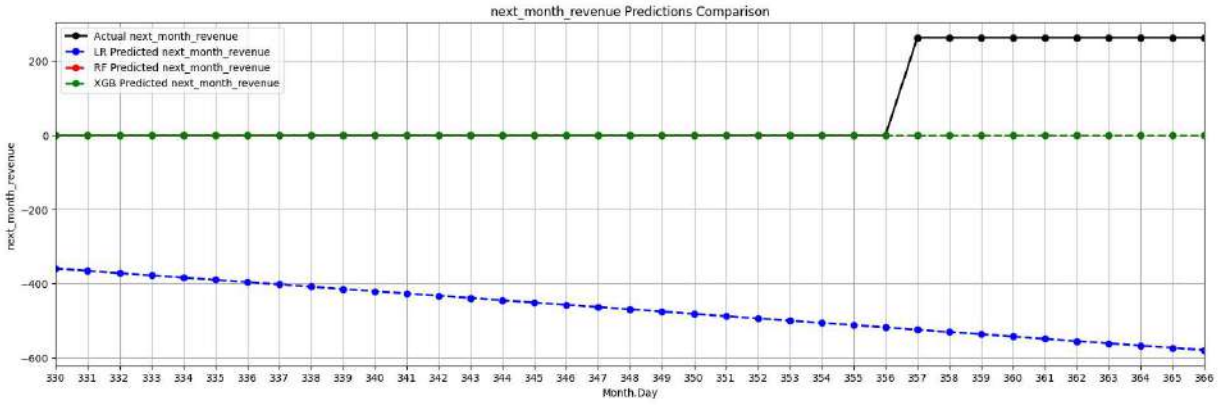


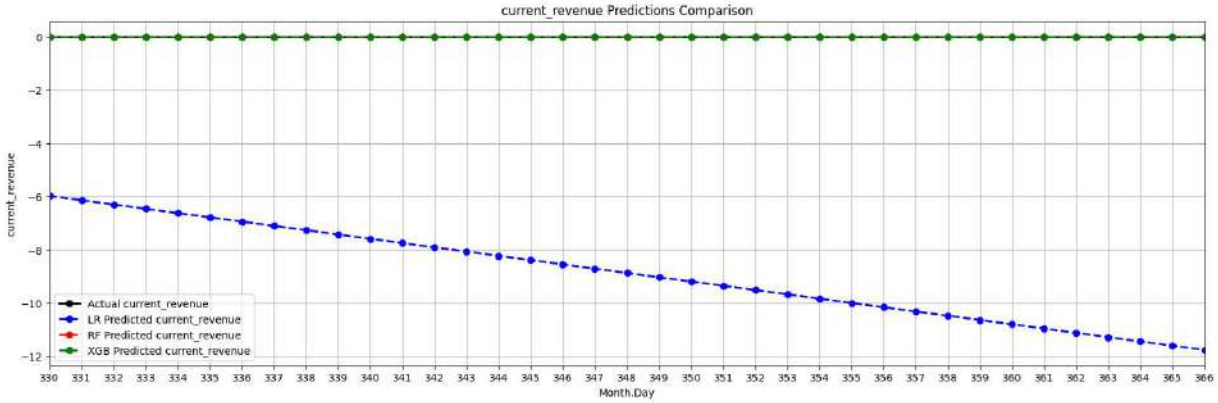


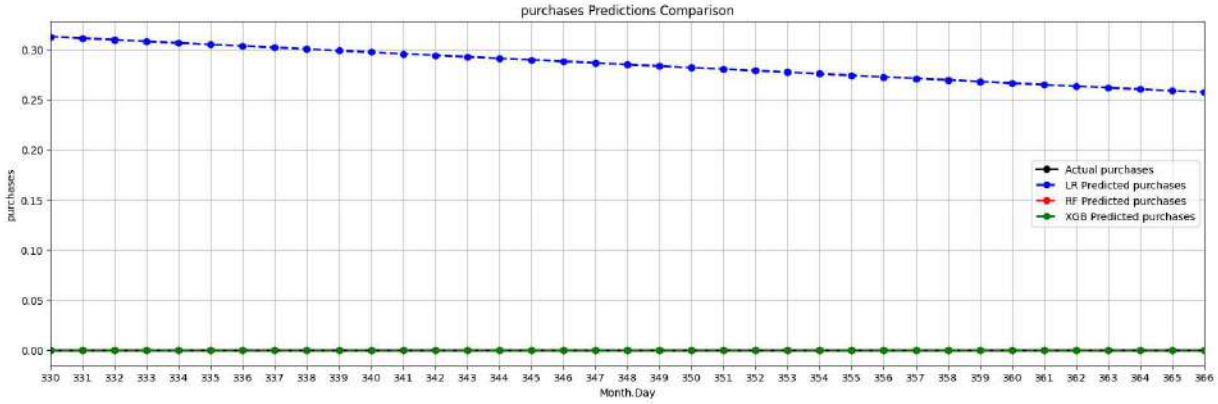


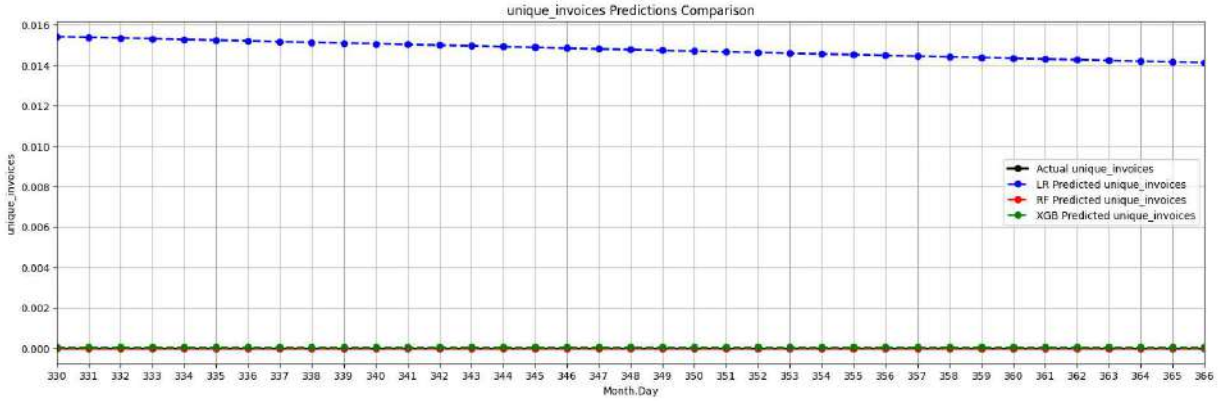


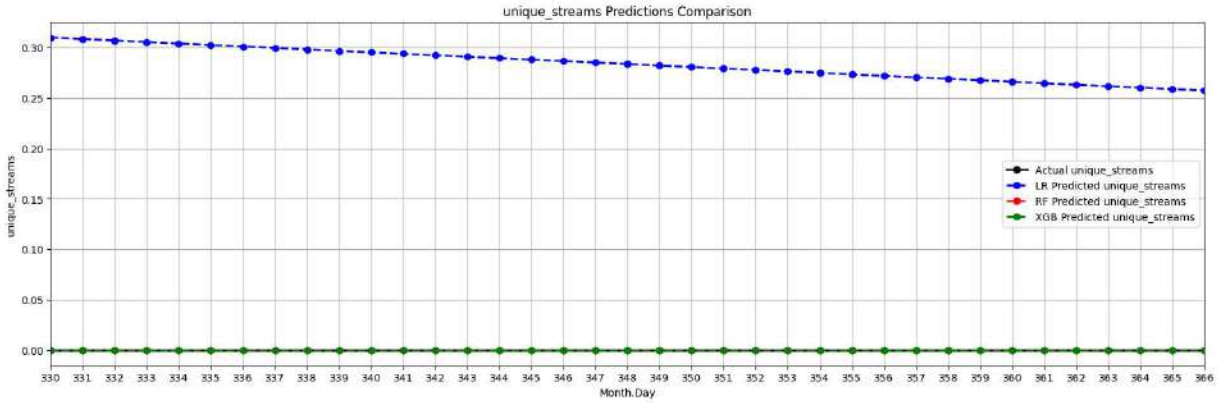
9 - Dataframe df_ts_singapore data:

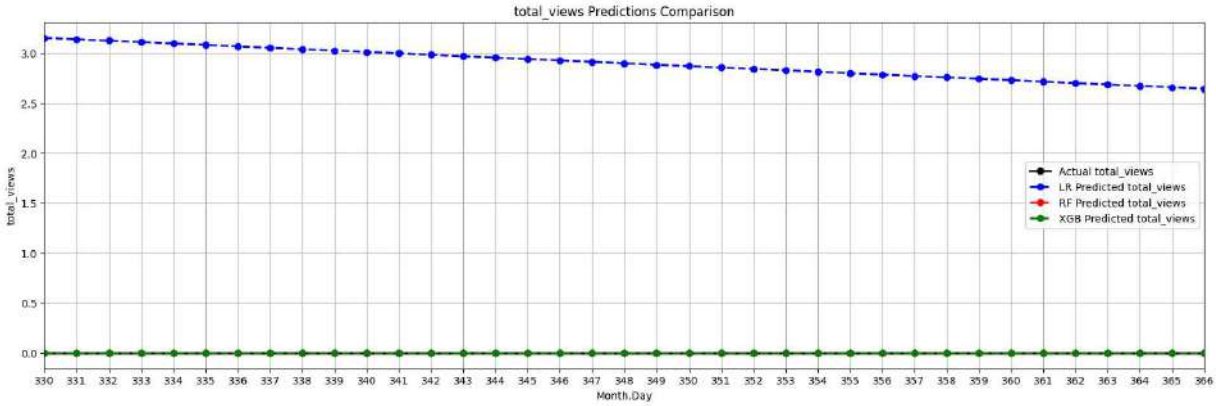



current_revenue Predictions Comparison
current_revenue
Month.Day
Actual current_revenue
LR Predicted current_revenue
RF Predicted current_revenue
XGB Predicted current_revenue

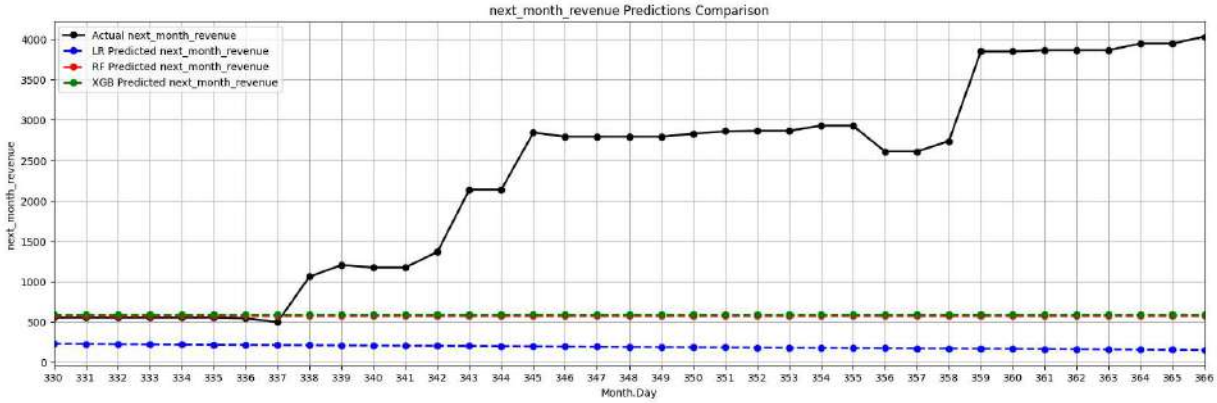

purchases Predictions Comparison
purchases
Month.Day
Actual purchases
LR Predicted purchases
RF Predicted purchases
XGB Predicted purchases

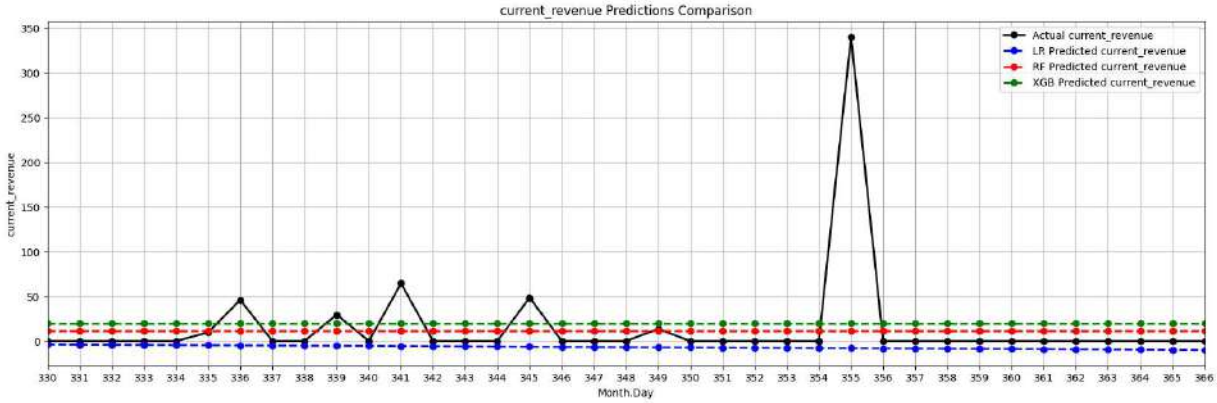

unique_invoices Predictions Comparison
unique_invoices
Month.Day
Actual unique_invoices
LR Predicted unique_invoices
RF Predicted unique_invoices
XGB Predicted unique_invoices

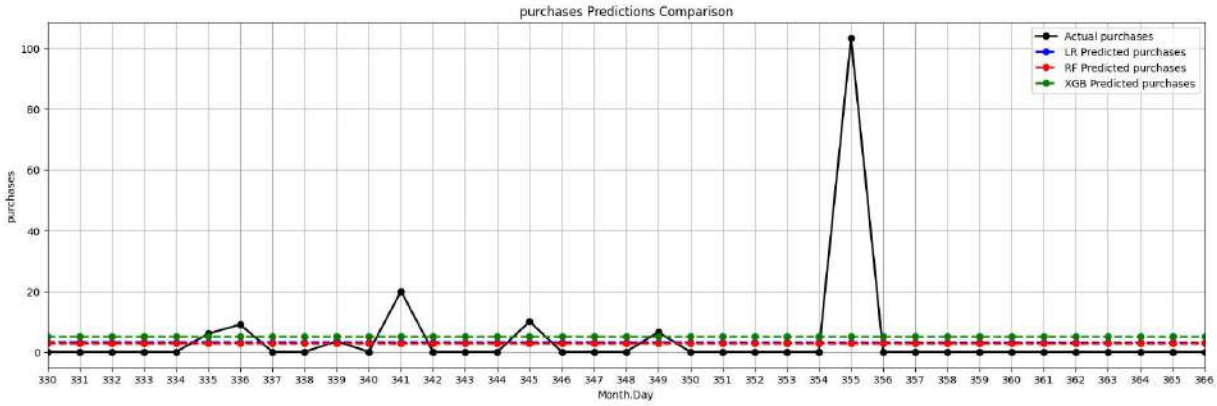

unique_streams Predictions Comparison
unique_streams
Month.Day
Actual unique_streams
LR Predicted unique_streams
RF Predicted unique_streams
XGB Predicted unique_streams

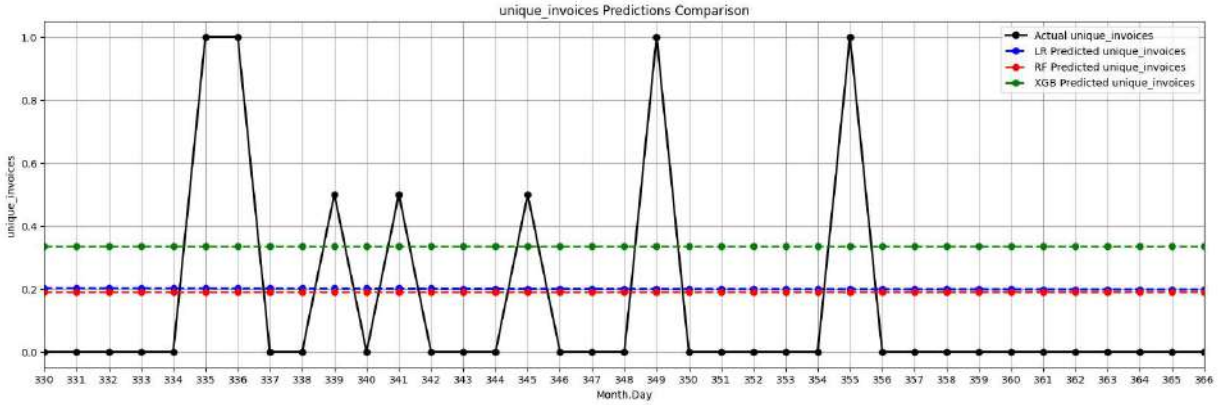


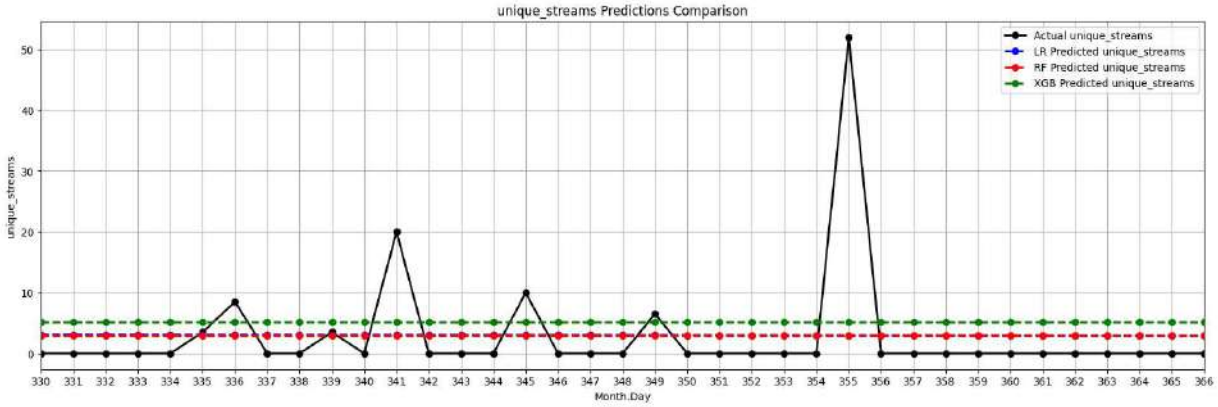
10 - Dataframe df_ts_spain data:

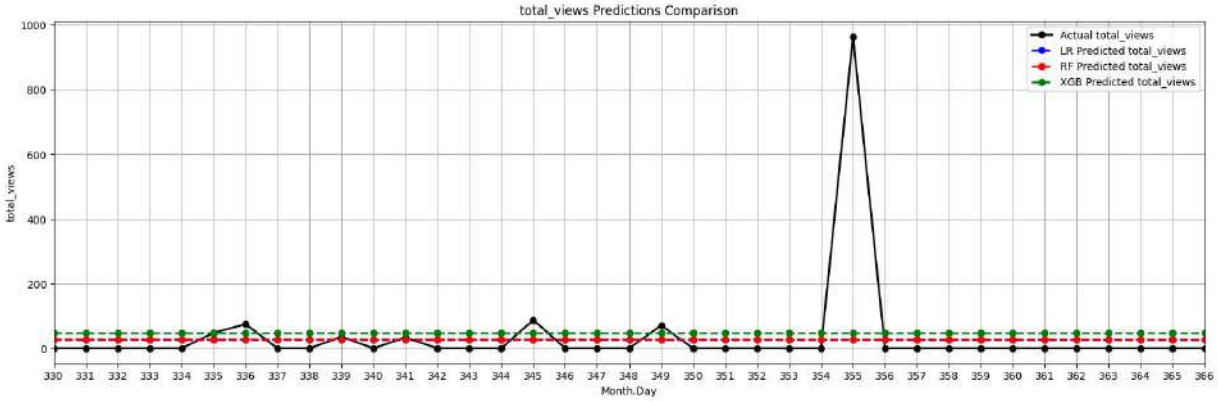




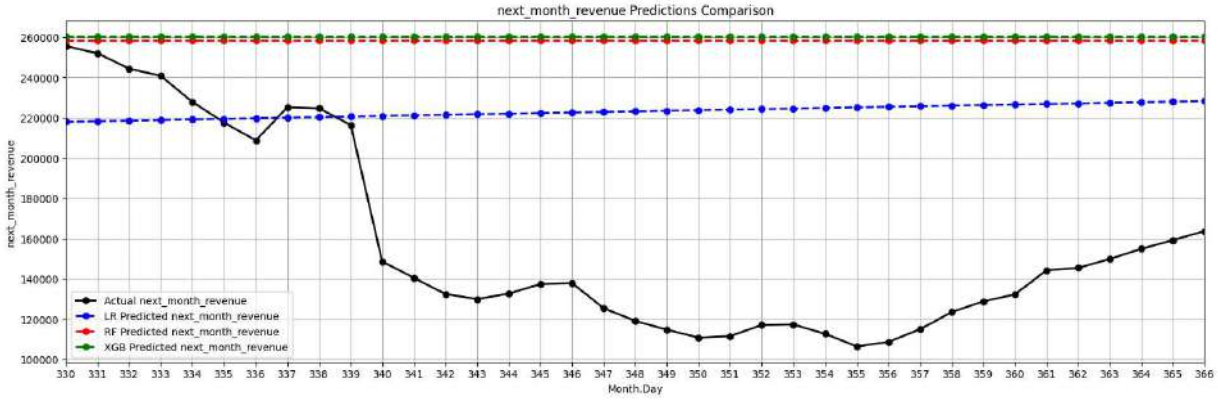


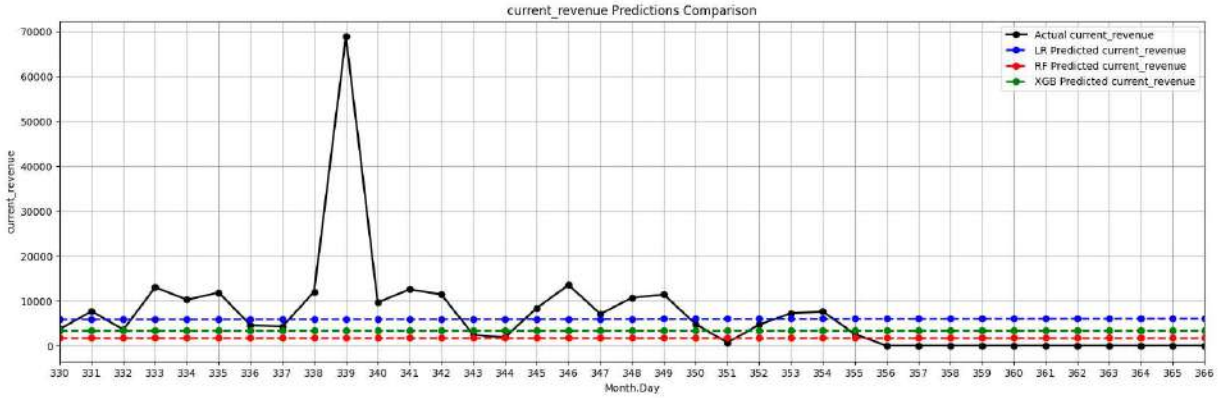


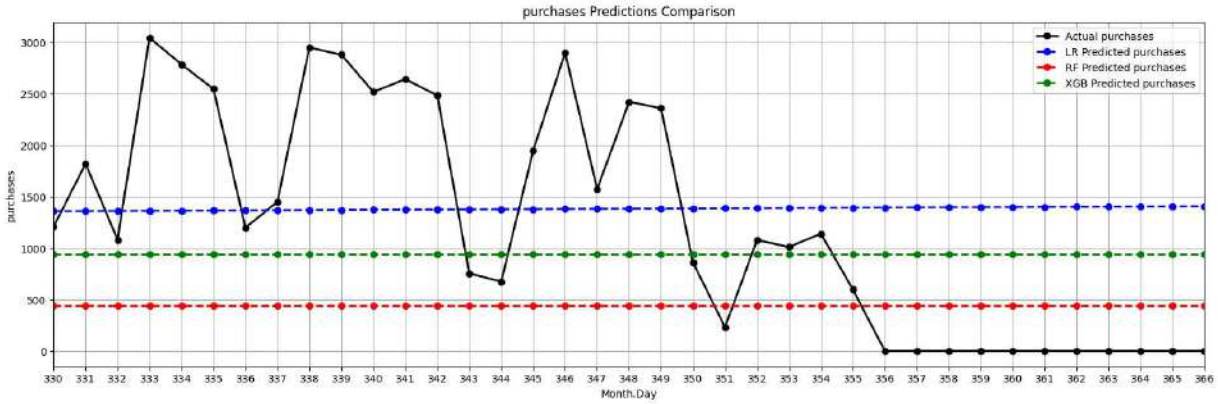


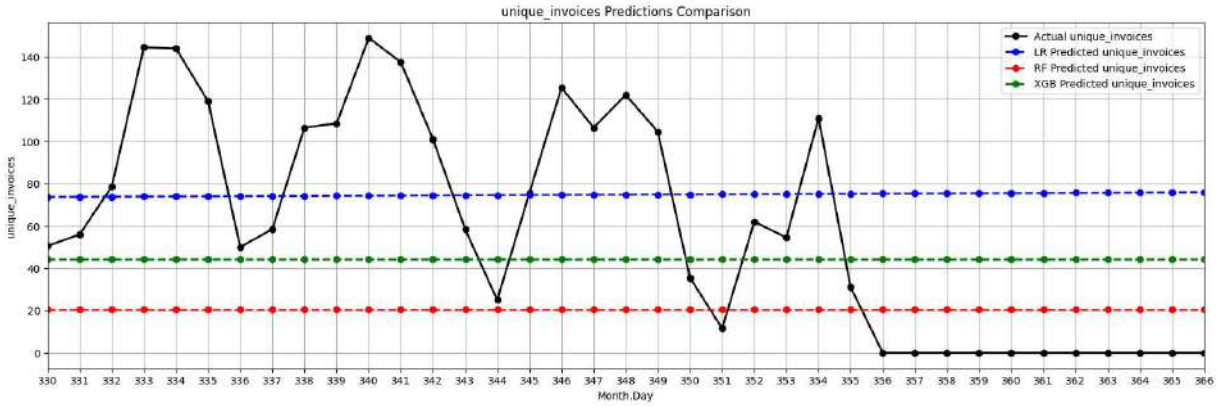


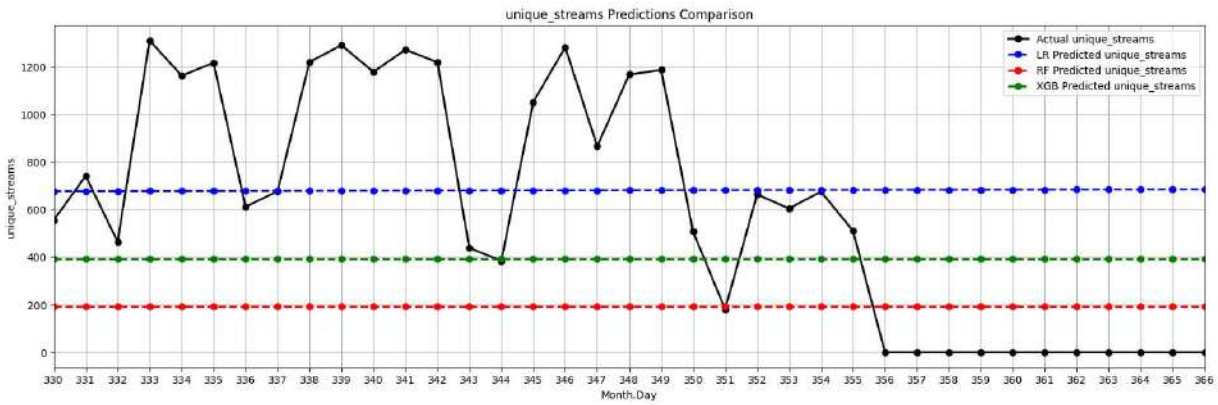
11 - Dataframe df_ts_united_kingdom data:

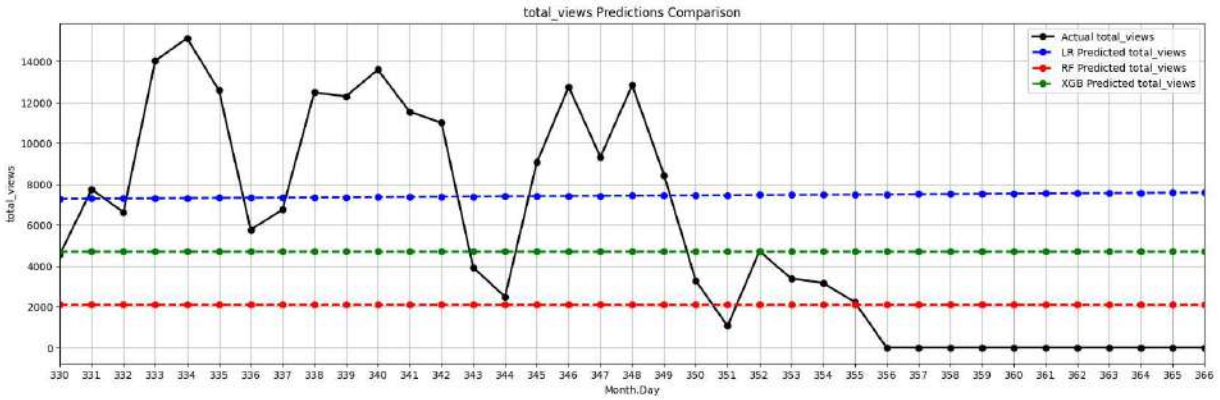



current_revenue Predictions Comparison
Actual current_revenue
LR Predicted current_revenue
RF Predicted current_revenue
XGB Predicted current_revenue
current_revenue
Month.Day


purchases Predictions Comparison
Actual purchases
LR Predicted purchases
RF Predicted purchases
XGB Predicted purchases
purchases
Month.Day


unique_invoices Predictions Comparison
Actual unique_invoices
LR Predicted unique_invoices
RF Predicted unique_invoices
XGB Predicted unique_invoices
unique_invoices
Month.Day


unique_streams Predictions Comparison
Actual unique_streams
LR Predicted unique_streams
RF Predicted unique_streams
XGB Predicted unique_streams
unique_streams
Month.Day



3.5-Test the Models with an additionnal single data

In [35]:
```
#calcualte predicted data using the models for January 3
#Data to use the model
single_month = 1
single_day = 3
day_rank = model.calculate_day_of_year(single_month, single_day)
test_date = {'day_of_year': day_rank}
new_test_df = pd.DataFrame(data=test_date, index=([0]))
print (new_test_df)
```

```
   day_of_year
0            3
```

In [36]:
```
#Use the models for an additionnal prediction
y_test_pred_lr, y_test_pred_rf, y_test_pred_xgb = {}, {}, {}
for i in range (0,size):
    df_name = list_ts_df[i]
    y_test_pred_lr[df_name] = lr_model[df_name].predict(new_test_df)
    y_test_pred_rf[df_name] = rf_model[df_name].predict(new_test_df)
    y_test_pred_xgb[df_name] = xgb_model[df_name].predict(new_test_df)

for i in range (0,size):
    df_name = list_ts_df[i]
    #Compare results
    print(i+1, '- Dataframe', df_name , 'data:')

    print('\nObserved data')
    print(library_ts_df_dayofyear[df_name]
        [(library_ts_df_dayofyear[df_name]['day_of_year']==test_date['day_of_year']

    print('\nPredicted data')
    d1 = {
        'purchases'         : [y_test_pred_lr[df_name][0][0],  y_test_pred_rf[df_na
        'unique_invoices'   : [y_test_pred_lr[df_name][0][1],  y_test_pred_rf[df_na
        'unique_streams'    : [y_test_pred_lr[df_name][0][2],  y_test_pred_rf[df_na
        'total_views'       : [y_test_pred_lr[df_name][0][3],  y_test_pred_rf[df_na
        'current_revenue'   : [y_test_pred_lr[df_name][0][4],  y_test_pred_rf[df_na
        'next_month_revenue': [y_test_pred_lr[df_name][0][5],  y_test_pred_rf[df_na
        }

    report = pd.DataFrame(data=d1, index=(['Linear Regression', 'Random Forest.', '
```

```
    print (report.T)
    print('\n')
```

1 - Dataframe df_ts_all data:

Observed data

	2
day_of_year	3.000
current_month	1.000
current_day	3.000
purchases	1060.500
unique_invoices	46.000
unique_streams	635.000
total_views	6103.000
current_revenue	4449.060
next_month_revenue	168808.037
next_month	2.000
next_day	3.000

Predicted data

	Linear Regression	Random Forest.	XGBoost Model
purchases	147259.410148	176920.5814	168845.265625
unique_invoices	5075.648794	10035.8669	4534.338867
unique_streams	1004.659186	1154.3400	1060.989258
total_views	58.165349	52.8200	50.616684
current_revenue	638.820125	708.2650	643.253296
next_month_revenue	5194.934973	6418.2050	6088.471680

2 - Dataframe df_ts_eire data:

Observed data

	2
day_of_year	3.000
current_month	1.000
current_day	3.000
purchases	0.000
unique_invoices	0.000
unique_streams	0.000
total_views	0.000
current_revenue	0.000
next_month_revenue	8024.775
next_month	2.000
next_day	3.000

Predicted data

	Linear Regression	Random Forest.	XGBoost Model
purchases	3412.690284	8046.35540	8026.586426
unique_invoices	108.171185	37.26025	9.649257
unique_streams	13.132271	6.02500	1.105011
total_views	0.668551	0.31500	0.074389
current_revenue	13.204520	5.91500	1.457390
next_month_revenue	103.969517	44.72000	11.676043

3 - Dataframe df_ts_france data:

Observed data

	2

day_of_year	3.000
current_month	1.000
current_day	3.000
purchases	23.500
unique_invoices	1.500
unique_streams	22.000
total_views	157.500
current_revenue	117.775
next_month_revenue	2031.530
next_month	2.000
next_day	3.000

Predicted data

	Linear Regression	Random Forest.	XGBoost Model
purchases	1846.003624	2077.16855	2033.576660
unique_invoices	58.234122	116.09355	116.128906
unique_streams	13.866264	28.20500	23.580143
total_views	0.795097	1.59500	1.484936
current_revenue	13.597110	25.58500	22.154858
next_month_revenue	129.182155	221.42000	164.431686

4 - Dataframe df_ts_germany data:

Observed data

	2
day_of_year	3.000
current_month	1.000
current_day	3.000
purchases	0.500
unique_invoices	0.500
unique_streams	0.500
total_views	2.000
current_revenue	0.825
next_month_revenue	2654.460
next_month	2.000
next_day	3.000

Predicted data

	Linear Regression	Random Forest.	XGBoost Model
purchases	2395.044955	2664.93465	2655.133545
unique_invoices	74.536296	26.69235	5.502501
unique_streams	20.103566	8.08500	1.911935
total_views	1.272764	0.76500	0.561665
current_revenue	18.813592	7.40500	1.806597
next_month_revenue	162.843882	47.57000	8.231142

5 - Dataframe df_ts_hong_kong data:

Observed data

	2
day_of_year	3.00
current_month	1.00
current_day	3.00
purchases	0.00

unique_invoices	0.00
unique_streams	0.00
total_views	0.00
current_revenue	0.00
next_month_revenue	167.66
next_month	2.00
next_day	3.00

Predicted data

	Linear Regression	Random Forest.	XGBoost Model
purchases	2079.832258	167.66	167.659866
unique_invoices	55.516379	0.00	0.000285
unique_streams	1.170003	0.00	0.000056
total_views	0.043001	0.00	0.000171
current_revenue	1.071951	0.00	0.000051
next_month_revenue	11.187175	0.00	0.000044

6 - Dataframe df_ts_netherlands data:

Observed data

	2
day_of_year	3.000
current_month	1.000
current_day	3.000
purchases	0.000
unique_invoices	0.000
unique_streams	0.000
total_views	0.000
current_revenue	0.000
next_month_revenue	438.785
next_month	2.000
next_day	3.000

Predicted data

	Linear Regression	Random Forest.	XGBoost Model
purchases	673.611686	439.01635	438.820496
unique_invoices	18.101838	0.00000	0.240068
unique_streams	5.126951	0.00000	0.081720
total_views	0.231655	0.00000	0.006801
current_revenue	5.011430	0.00000	0.136768
next_month_revenue	58.309195	0.00000	0.996260

7 - Dataframe df_ts_norway data:

Observed data

	2
day_of_year	3.00
current_month	1.00
current_day	3.00
purchases	0.00
unique_invoices	0.00
unique_streams	0.00
total_views	0.00
current_revenue	0.00

next_month_revenue	32.65
next_month	2.00
next_day	3.00

Predicted data

	Linear Regression	Random Forest.	XGBoost Model
purchases	5067.544921	32.65	32.650166
unique_invoices	143.046484	0.00	0.000223
unique_streams	-0.133247	0.00	0.000137
total_views	0.047944	0.00	0.000144
current_revenue	-0.200324	0.00	0.000112
next_month_revenue	-3.118463	0.00	0.000110

8 - Dataframe df_ts_portugal data:

Observed data

	2
day_of_year	3.000
current_month	1.000
current_day	3.000
purchases	0.000
unique_invoices	0.000
unique_streams	0.000
total_views	0.000
current_revenue	0.000
next_month_revenue	313.175
next_month	2.000
next_day	3.000

Predicted data

	Linear Regression	Random Forest.	XGBoost Model
purchases	778.322704	315.788	313.158813
unique_invoices	22.162783	2.278	0.287646
unique_streams	1.347769	0.680	0.060133
total_views	0.091639	0.170	0.006284
current_revenue	1.326852	0.680	0.081194
next_month_revenue	12.426632	0.340	0.463364

9 - Dataframe df_ts_singapore data:

Observed data

	2
day_of_year	3.00
current_month	1.00
current_day	3.00
purchases	0.00
unique_invoices	0.00
unique_streams	0.00
total_views	0.00
current_revenue	0.00
next_month_revenue	263.34
next_month	2.00
next_day	3.00

Predicted data

	Linear Regression	Random Forest.	XGBoost Model
purchases	1639.551560	263.34	263.339905
unique_invoices	46.540388	0.00	0.000094
unique_streams	0.817568	0.00	0.000036
total_views	0.027174	0.00	0.000110
current_revenue	0.787169	0.00	0.000036
next_month_revenue	7.781318	0.00	0.000036

10 - Dataframe df_ts_spain data:

Observed data

	2
day_of_year	3.00
current_month	1.00
current_day	3.00
purchases	0.50
unique_invoices	0.50
unique_streams	0.50
total_views	1.50
current_revenue	563.00
next_month_revenue	4030.31
next_month	2.00
next_day	3.00

Predicted data

	Linear Regression	Random Forest.	XGBoost Model
purchases	918.858260	4000.7268	4030.159668
unique_invoices	52.256152	321.0128	559.973816
unique_streams	4.664694	2.2400	0.563329
total_views	0.245691	0.3000	0.496744
current_revenue	4.610817	2.2400	0.530001
next_month_revenue	33.181306	11.5600	1.852428

11 - Dataframe df_ts_united_kingdom data:

Observed data

	2
day_of_year	3.000
current_month	1.000
current_day	3.000
purchases	1008.500
unique_invoices	41.500
unique_streams	617.500
total_views	5639.000
current_revenue	3709.820
next_month_revenue	147300.177
next_month	2.000
next_day	3.000

Predicted data

	Linear Regression	Random Forest.	XGBoost Model
purchases	125046.751506	155330.32430	147319.937500
unique_invoices	4407.648972	9495.10335	3761.890381

unique_streams	925.274922	1092.62500	1018.405396
total_views	53.472213	47.97000	44.803173
current_revenue	604.144652	684.52000	630.789246
next_month_revenue	4512.808102	5917.43500	5601.660156

4-Create and Test 3 One target regression Models

This strategy consists of a multiple regression that predicts only one target based on multiple independent variables. This is the most common way to perform regression.

4.1.-Split data into predictors (X) and target (y)

In [37]:
```
# Split the data into features and target
# X = X[df_name] and y = [df_name]
X = {}
y = {}
for i in range (0,size):
    df_name = list_ts_df[i]
    X[df_name], y[df_name] = model.split_into_predictors_targets(library_ts_df_dayo
    print(i+1, '- Dataframe', df_name , 'was successfully splitted')
    print("SHAPE OF THE DATA: ")
    print("Number of (rows, column) X : " + str((X[df_name]).shape))
    print("Number of (row, columns) y : " + str((y[df_name]).shape))
    print('\n')
```

```
1 - Dataframe df_ts_all was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 6)
Number of (row, columns) y : (365, 1)


2 - Dataframe df_ts_eire was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 6)
Number of (row, columns) y : (365, 1)


3 - Dataframe df_ts_france was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 6)
Number of (row, columns) y : (365, 1)


4 - Dataframe df_ts_germany was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 6)
Number of (row, columns) y : (365, 1)


5 - Dataframe df_ts_hong_kong was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 6)
Number of (row, columns) y : (365, 1)


6 - Dataframe df_ts_netherlands was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 6)
Number of (row, columns) y : (365, 1)


7 - Dataframe df_ts_norway was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 6)
Number of (row, columns) y : (365, 1)


8 - Dataframe df_ts_portugal was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 6)
Number of (row, columns) y : (365, 1)


9 - Dataframe df_ts_singapore was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 6)
Number of (row, columns) y : (365, 1)


10 - Dataframe df_ts_spain was successfully splitted
SHAPE OF THE DATA:
```

```
Number of (rows, column) X : (365, 6)
Number of (row, columns) y : (365, 1)


11 - Dataframe df_ts_united_kingdom was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) X : (365, 6)
Number of (row, columns) y : (365, 1)
```

4.2.-Split data into training (X_train, y_train) and test (X_test, y_test) sets

In [38]:

```
# Randomly split the data into 80% for training and 20% for test sets
# X_train=X_train[df_name] , y_train=y_train[df_name], X_test=X_test[df_name] and y
X_train, X_test , y_train , y_test = {}, {}, {}, {}
for i in range (0,size):
    df_name = list_ts_df[i]
    X_train[df_name], X_test[df_name], y_train[df_name], y_test[df_name] = model.sp
    print(i+1, '- Dataframe', df_name , 'was successfully splitted')
    print("SHAPE OF THE DATA: ")
    print("Number of (rows, column) Xtrain : " + str((X_train[df_name]).shape))
    print("Number of (row, columns) Xtest : " + str((X_test[df_name]).shape))
    print("Number of (rows, column) ytrain : " + str((y_train[df_name]).shape))
    print("Number of (row, columns) ytest : " + str((y_test[df_name]).shape))
    print('\n')
```

```
1 - Dataframe df_ts_all was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 6)
Number of (row, columns) Xtest : (37, 6)
Number of (rows, column) ytrain : (328, 1)
Number of (row, columns) ytest : (37, 1)


2 - Dataframe df_ts_eire was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 6)
Number of (row, columns) Xtest : (37, 6)
Number of (rows, column) ytrain : (328, 1)
Number of (row, columns) ytest : (37, 1)


3 - Dataframe df_ts_france was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 6)
Number of (row, columns) Xtest : (37, 6)
Number of (rows, column) ytrain : (328, 1)
Number of (row, columns) ytest : (37, 1)


4 - Dataframe df_ts_germany was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 6)
Number of (row, columns) Xtest : (37, 6)
Number of (rows, column) ytrain : (328, 1)
Number of (row, columns) ytest : (37, 1)


5 - Dataframe df_ts_hong_kong was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 6)
Number of (row, columns) Xtest : (37, 6)
Number of (rows, column) ytrain : (328, 1)
Number of (row, columns) ytest : (37, 1)


6 - Dataframe df_ts_netherlands was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 6)
Number of (row, columns) Xtest : (37, 6)
Number of (rows, column) ytrain : (328, 1)
Number of (row, columns) ytest : (37, 1)


7 - Dataframe df_ts_norway was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 6)
Number of (row, columns) Xtest : (37, 6)
Number of (rows, column) ytrain : (328, 1)
Number of (row, columns) ytest : (37, 1)
```

```
8 - Dataframe df_ts_portugal was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 6)
Number of (row, columns) Xtest : (37, 6)
Number of (rows, column) ytrain : (328, 1)
Number of (row, columns) ytest : (37, 1)


9 - Dataframe df_ts_singapore was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 6)
Number of (row, columns) Xtest : (37, 6)
Number of (rows, column) ytrain : (328, 1)
Number of (row, columns) ytest : (37, 1)


10 - Dataframe df_ts_spain was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 6)
Number of (row, columns) Xtest : (37, 6)
Number of (rows, column) ytrain : (328, 1)
Number of (row, columns) ytest : (37, 1)


11 - Dataframe df_ts_united_kingdom was successfully splitted
SHAPE OF THE DATA:
Number of (rows, column) Xtrain : (328, 6)
Number of (row, columns) Xtest : (37, 6)
Number of (rows, column) ytrain : (328, 1)
Number of (row, columns) ytest : (37, 1)
```

In [39]:
```
#Sort in ascending to be able to plot appropiate time-series visualizationabs
for i in range (0,size):
    df_name = list_ts_df[i]
    X_train[df_name]=X_train[df_name].sort_index(ascending=True)
    X_test[df_name] =X_test [df_name].sort_index(ascending=True)
    y_train[df_name]=y_train[df_name].sort_index(ascending=True)
    y_test[df_name] =y_test [df_name].sort_index(ascending=True)
```

4.3.-Train Predict and Evaluate 3 models

In [40]:
```
# Train, predict, and evaluate each model
y_pred_lr , y_pred_rf , y_pred_xgb , lr_model , rf_model, xgb_model = {}, {}, {}, {
for i in range (0,size):
    df_name = list_ts_df[i]
    print(i+1, '- The model of Dataframe', df_name , 'was successfully trained, use
    y_pred_lr[df_name], y_pred_rf[df_name], y_pred_xgb[df_name], lr_model[df_name],
    print('\n')
```

```
1 - The model of Dataframe df_ts_all was successfully trained, used to predict and e
valuated as follows:
```

```
C:\Users\MDA\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.
py:1389: DataConversionWarning: A column-vector y was passed when a 1d array was exp
ected. Please change the shape of y to (n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
Linear Regression Model:
Mean Absolute Error (MAE): 71991.24
Mean Squared Error (MSE) : 6361149625.34
Root Mean-Square Error (RMSE) : 79756.82
Coefficient of Determination (R2) : -1.96

Random Forest Model:
Mean Absolute Error (MAE): 96847.74
Mean Squared Error (MSE) : 11654151659.46
Root Mean-Square Error (RMSE) : 107954.40
Coefficient of Determination (R2) : -4.42

XGBoost Model:
Mean Absolute Error (MAE): 100682.98
Mean Squared Error (MSE) : 12174680064.00
Root Mean-Square Error (RMSE) : 110338.93
Coefficient of Determination (R2) : -4.67

Evaluation Summary :
                           MAE           MSE            RMSE        R2
Linear           71991.239701  6.361150e+09   79756.815542 -1.960992
Random Forest    96847.737235  1.165415e+10  107954.396203 -4.424783
XGBoost         100682.984375  1.217468e+10  110338.932676 -4.667078


2 - The model of Dataframe df_ts_eire was successfully trained, used to predict and
evaluated as follows:

C:\Users\MDA\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.
py:1389: DataConversionWarning: A column-vector y was passed when a 1d array was exp
ected. Please change the shape of y to (n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
```

```
Linear Regression Model:
Mean Absolute Error (MAE): 4127.11
Mean Squared Error (MSE) : 20897239.02
Root Mean-Square Error (RMSE) : 4571.35
Coefficient of Determination (R2) : -6.36

Random Forest Model:
Mean Absolute Error (MAE): 1529.65
Mean Squared Error (MSE) : 4478246.42
Root Mean-Square Error (RMSE) : 2116.19
Coefficient of Determination (R2) : -0.58

XGBoost Model:
Mean Absolute Error (MAE): 1531.69
Mean Squared Error (MSE) : 4461023.50
Root Mean-Square Error (RMSE) : 2112.11
Coefficient of Determination (R2) : -0.57

Evaluation Summary :
                          MAE           MSE         RMSE        R2
Linear         4127.113978  2.089724e+07  4571.349802 -6.359982
Random Forest  1529.654673  4.478246e+06  2116.186764 -0.577233
XGBoost        1531.687744  4.461024e+06  2112.113515 -0.571167


3 - The model of Dataframe df_ts_france was successfully trained, used to predict an
d evaluated as follows:

C:\Users\MDA\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.
py:1389: DataConversionWarning: A column-vector y was passed when a 1d array was exp
ected. Please change the shape of y to (n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
```

```
Linear Regression Model:
Mean Absolute Error (MAE): 1175.51
Mean Squared Error (MSE) : 1419004.95
Root Mean-Square Error (RMSE) : 1191.22
Coefficient of Determination (R2) : -23.12

Random Forest Model:
Mean Absolute Error (MAE): 206.29
Mean Squared Error (MSE) : 64971.81
Root Mean-Square Error (RMSE) : 254.90
Coefficient of Determination (R2) : -0.10

XGBoost Model:
Mean Absolute Error (MAE): 216.50
Mean Squared Error (MSE) : 70479.23
Root Mean-Square Error (RMSE) : 265.48
Coefficient of Determination (R2) : -0.20

Evaluation Summary :
                          MAE           MSE         RMSE         R2
Linear         1175.506101  1.419005e+06  1191.219941 -23.123521
Random Forest   206.288259  6.497181e+04   254.895678  -0.104541
XGBoost         216.501419  7.047923e+04   265.479239  -0.198169


4 - The model of Dataframe df_ts_germany was successfully trained, used to predict a
nd evaluated as follows:

C:\Users\MDA\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.
py:1389: DataConversionWarning: A column-vector y was passed when a 1d array was exp
ected. Please change the shape of y to (n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
```

Linear Regression Model:
Mean Absolute Error (MAE): 910.10
Mean Squared Error (MSE) : 951463.45
Root Mean-Square Error (RMSE) : 975.43
Coefficient of Determination (R2) : -7.29

Random Forest Model:
Mean Absolute Error (MAE): 560.83
Mean Squared Error (MSE) : 385803.00
Root Mean-Square Error (RMSE) : 621.13
Coefficient of Determination (R2) : -2.36

XGBoost Model:
Mean Absolute Error (MAE): 614.12
Mean Squared Error (MSE) : 460624.16
Root Mean-Square Error (RMSE) : 678.69
Coefficient of Determination (R2) : -3.01

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	910.104197	951463.452744	975.429881	-7.290390
Random Forest	560.830705	385803.002567	621.130423	-2.361619
XGBoost	614.116089	460624.156250	678.692976	-3.013558

5 - The model of Dataframe df_ts_hong_kong was successfully trained, used to predict and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 124.32
Mean Squared Error (MSE) : 17724.72
Root Mean-Square Error (RMSE) : 133.13
Coefficient of Determination (R2) : -2.02

Random Forest Model:
Mean Absolute Error (MAE): 49.84
Mean Squared Error (MSE) : 8356.99
Root Mean-Square Error (RMSE) : 91.42
Coefficient of Determination (R2) : -0.42

XGBoost Model:
Mean Absolute Error (MAE): 49.84
Mean Squared Error (MSE) : 8356.99
Root Mean-Square Error (RMSE) : 91.42
Coefficient of Determination (R2) : -0.42

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	124.316062	17724.724102	133.134233	-2.018269
Random Forest	49.844865	8356.990043	91.416574	-0.423077
XGBoost	49.844879	8356.986328	91.416554	-0.423076

6 - The model of Dataframe df_ts_netherlands was successfully trained, used to predict and evaluated as follows:

```
C:\Users\MDA\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.
py:1389: DataConversionWarning: A column-vector y was passed when a 1d array was exp
ected. Please change the shape of y to (n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
C:\Users\MDA\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.
py:1389: DataConversionWarning: A column-vector y was passed when a 1d array was exp
ected. Please change the shape of y to (n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
```

```
Linear Regression Model:
Mean Absolute Error (MAE): 496.37
Mean Squared Error (MSE) : 253601.97
Root Mean-Square Error (RMSE) : 503.59
Coefficient of Determination (R2) : -33.68

Random Forest Model:
Mean Absolute Error (MAE): 122.42
Mean Squared Error (MSE) : 20265.65
Root Mean-Square Error (RMSE) : 142.36
Coefficient of Determination (R2) : -1.77

XGBoost Model:
Mean Absolute Error (MAE): 108.07
Mean Squared Error (MSE) : 15471.57
Root Mean-Square Error (RMSE) : 124.38
Coefficient of Determination (R2) : -1.12

Evaluation Summary :
                         MAE             MSE        RMSE         R2
Linear         496.372028  253601.965091  503.589084 -33.681493
Random Forest  122.423462   20265.651635  142.357478  -1.771442
XGBoost        108.073219   15471.572266  124.384775  -1.115824


7 - The model of Dataframe df_ts_norway was successfully trained, used to predict an
d evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 3487.33
Mean Squared Error (MSE) : 73490259.36
Root Mean-Square Error (RMSE) : 8572.65
Coefficient of Determination (R2) : -6838.79

Random Forest Model:
Mean Absolute Error (MAE): 274.63
Mean Squared Error (MSE) : 85933.19
Root Mean-Square Error (RMSE) : 293.14
Coefficient of Determination (R2) : -7.00

XGBoost Model:
Mean Absolute Error (MAE): 276.77
Mean Squared Error (MSE) : 86401.57
Root Mean-Square Error (RMSE) : 293.94
Coefficient of Determination (R2) : -7.04

Evaluation Summary :
                          MAE           MSE         RMSE           R2
Linear         3487.332398  7.349026e+07  8572.645995 -6838.792941
Random Forest   274.629668  8.593319e+04   293.143627    -6.997865
XGBoost         276.770416  8.640157e+04   293.941440    -7.041458


8 - The model of Dataframe df_ts_portugal was successfully trained, used to predict
and evaluated as follows:
```

```
C:\Users\MDA\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.
py:1389: DataConversionWarning: A column-vector y was passed when a 1d array was exp
ected. Please change the shape of y to (n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
C:\Users\MDA\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.
py:1389: DataConversionWarning: A column-vector y was passed when a 1d array was exp
ected. Please change the shape of y to (n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
```

Linear Regression Model:
Mean Absolute Error (MAE): 291.84
Mean Squared Error (MSE) : 94954.78
Root Mean-Square Error (RMSE) : 308.15
Coefficient of Determination (R2) : -9.83

Random Forest Model:
Mean Absolute Error (MAE): 223.35
Mean Squared Error (MSE) : 58619.03
Root Mean-Square Error (RMSE) : 242.11
Coefficient of Determination (R2) : -5.68

XGBoost Model:
Mean Absolute Error (MAE): 219.99
Mean Squared Error (MSE) : 56925.79
Root Mean-Square Error (RMSE) : 238.59
Coefficient of Determination (R2) : -5.49

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	291.840686	94954.778967	308.147333	-9.825517
Random Forest	223.347445	58619.026923	242.113665	-5.682984
XGBoost	219.988846	56925.785156	238.591251	-5.489943

9 - The model of Dataframe df_ts_singapore was successfully trained, used to predict and evaluated as follows:

Linear Regression Model:
Mean Absolute Error (MAE): 542.81
Mean Squared Error (MSE) : 324246.53
Root Mean-Square Error (RMSE) : 569.43
Coefficient of Determination (R2) : -22.71

Random Forest Model:
Mean Absolute Error (MAE): 71.17
Mean Squared Error (MSE) : 18742.69
Root Mean-Square Error (RMSE) : 136.90
Coefficient of Determination (R2) : -0.37

XGBoost Model:
Mean Absolute Error (MAE): 71.17
Mean Squared Error (MSE) : 18742.69
Root Mean-Square Error (RMSE) : 136.90
Coefficient of Determination (R2) : -0.37

Evaluation Summary :

	MAE	MSE	RMSE	R2
Linear	542.813484	324246.534213	569.426496	-22.70726
Random Forest	71.172973	18742.690703	136.903947	-0.37037
XGBoost	71.172974	18742.685547	136.903928	-0.37037

10 - The model of Dataframe df_ts_spain was successfully trained, used to predict and evaluated as follows:

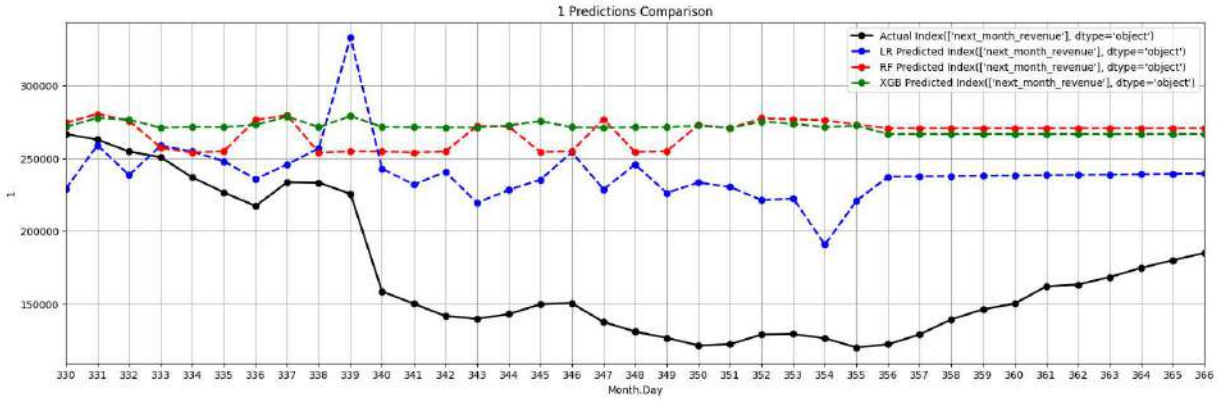
```
C:\Users\MDA\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.
py:1389: DataConversionWarning: A column-vector y was passed when a 1d array was exp
ected. Please change the shape of y to (n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
C:\Users\MDA\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.
py:1389: DataConversionWarning: A column-vector y was passed when a 1d array was exp
ected. Please change the shape of y to (n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
Linear Regression Model:
Mean Absolute Error (MAE): 2177.96
Mean Squared Error (MSE) : 6820162.51
Root Mean-Square Error (RMSE) : 2611.54
Coefficient of Determination (R2) : -3.58

Random Forest Model:
Mean Absolute Error (MAE): 1739.02
Mean Squared Error (MSE) : 4471607.27
Root Mean-Square Error (RMSE) : 2114.62
Coefficient of Determination (R2) : -2.00

XGBoost Model:
Mean Absolute Error (MAE): 1729.53
Mean Squared Error (MSE) : 4421824.50
Root Mean-Square Error (RMSE) : 2102.81
Coefficient of Determination (R2) : -1.97

Evaluation Summary :
                         MAE           MSE         RMSE        R2
Linear         2177.963858  6.820163e+06  2611.544086 -3.580034
Random Forest  1739.024689  4.471607e+06  2114.617523 -2.002878
XGBoost        1729.529297  4.421824e+06  2102.813472 -1.969447


11 - The model of Dataframe df_ts_united_kingdom was successfully trained, used to p
redict and evaluated as follows:

C:\Users\MDA\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.
py:1389: DataConversionWarning: A column-vector y was passed when a 1d array was exp
ected. Please change the shape of y to (n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
```

```
Linear Regression Model:
Mean Absolute Error (MAE): 70926.42
Mean Squared Error (MSE) : 6184747542.29
Root Mean-Square Error (RMSE) : 78643.17
Coefficient of Determination (R2) : -1.74

Random Forest Model:
Mean Absolute Error (MAE): 90637.04
Mean Squared Error (MSE) : 10578003857.86
Root Mean-Square Error (RMSE) : 102849.42
Coefficient of Determination (R2) : -3.68

XGBoost Model:
Mean Absolute Error (MAE): 88207.73
Mean Squared Error (MSE) : 10034798592.00
Root Mean-Square Error (RMSE) : 100173.84
Coefficient of Determination (R2) : -3.44

Evaluation Summary :
                        MAE           MSE            RMSE        R2
Linear         70926.417361  6.184748e+09   78643.165897 -1.738337
Random Forest  90637.039213  1.057800e+10  102849.423226 -3.683480
XGBoost        88207.726562  1.003480e+10  100173.841855 -3.442972
```

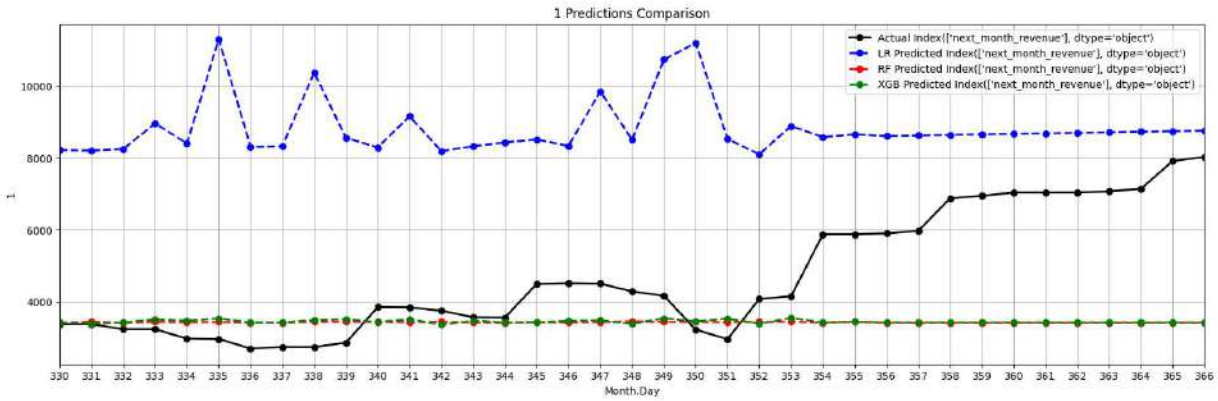
3.4.-Plot test data observed and predicted with the 3 models

In [41]:
```
# Plot results for only day of year and next month revenue
for i in range (0,size):
    df_name = list_ts_df[i]
    print(i+1, '- Dataframe', df_name , 'data:')
    viz.plot_test_data(X_test[df_name]['day_of_year'], y_test[df_name], y_pred_lr[d
```

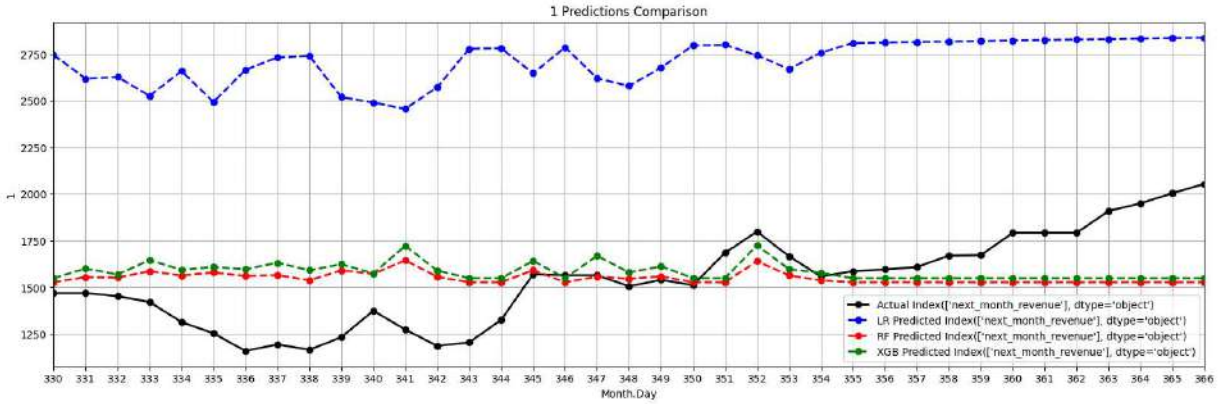
```
1 - Dataframe df_ts_all data:
```



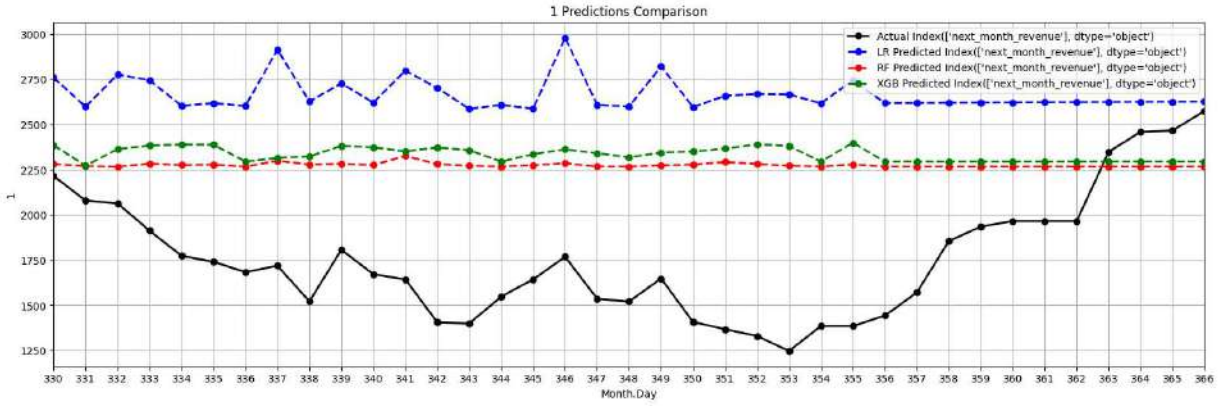

```
2 - Dataframe df_ts_eire data:
```



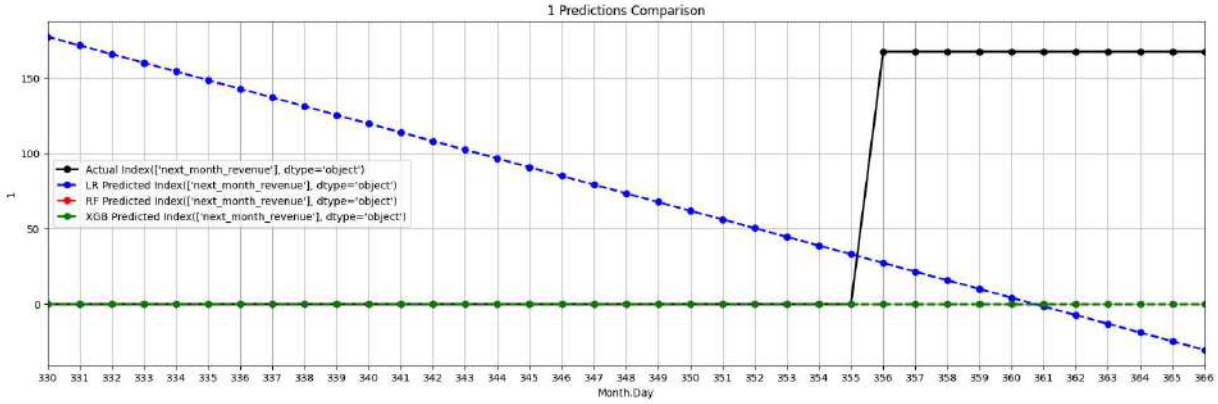
3 - Dataframe df_ts_france data:



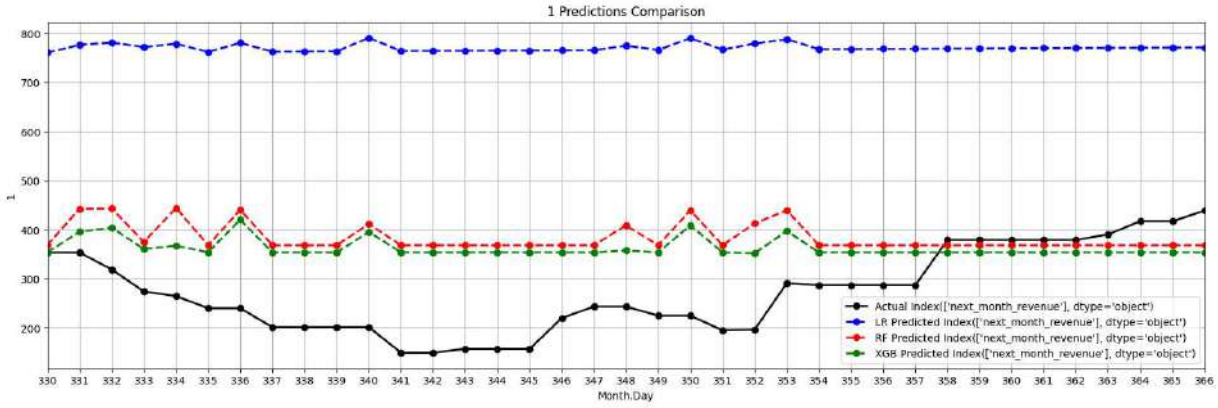
4 - Dataframe df_ts_germany data:



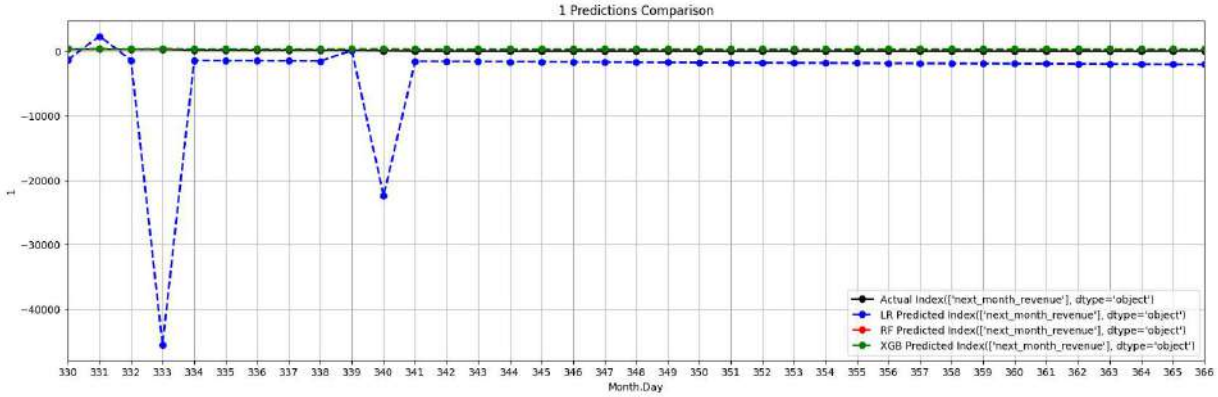
5 - Dataframe df_ts_hong_kong data:



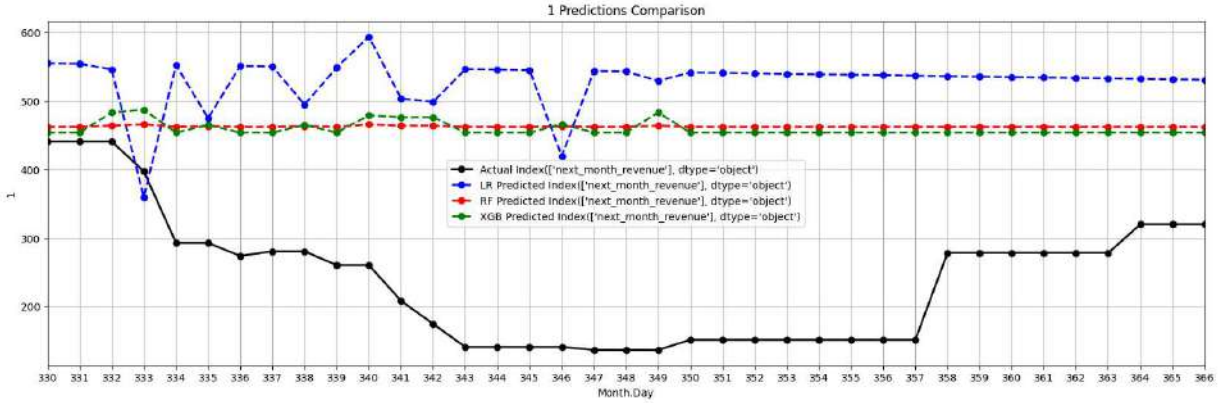
6 - Dataframe df_ts_netherlands data:



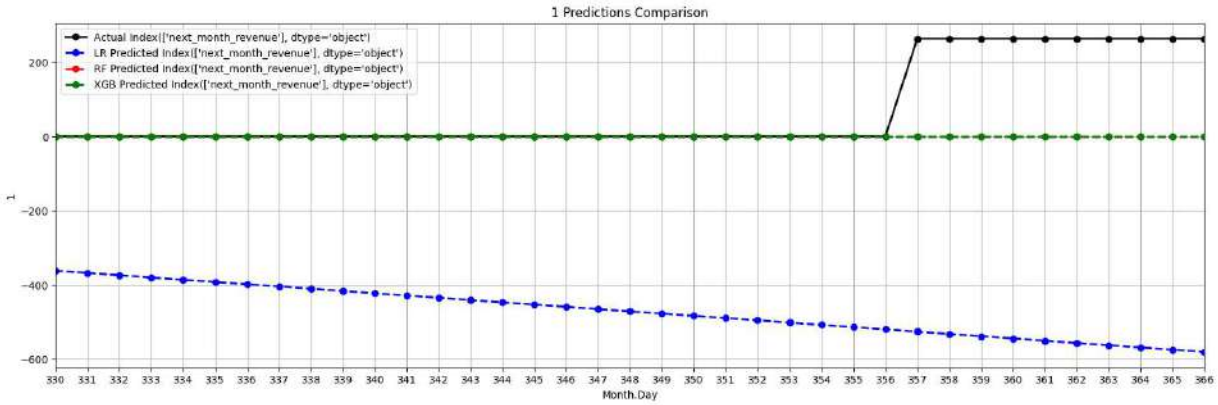
7 - Dataframe df_ts_norway data:



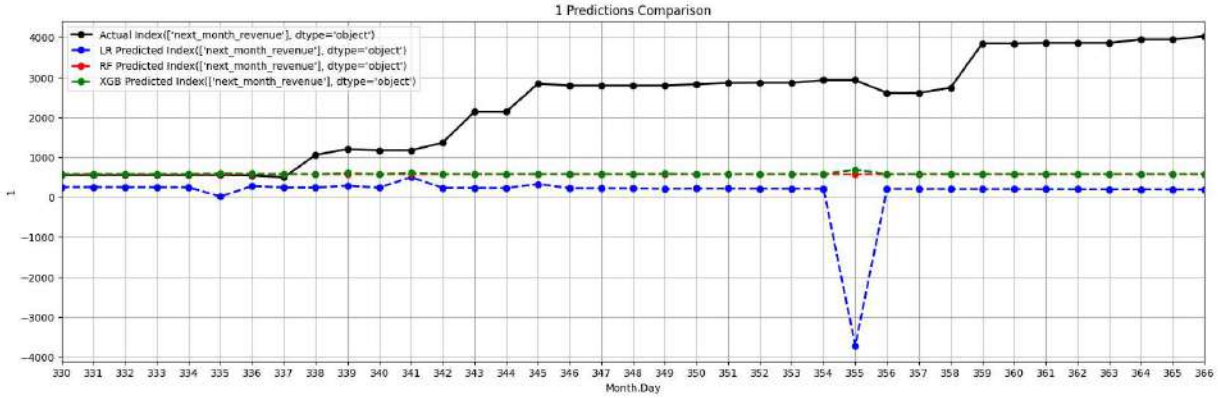
8 - Dataframe df_ts_portugal data:



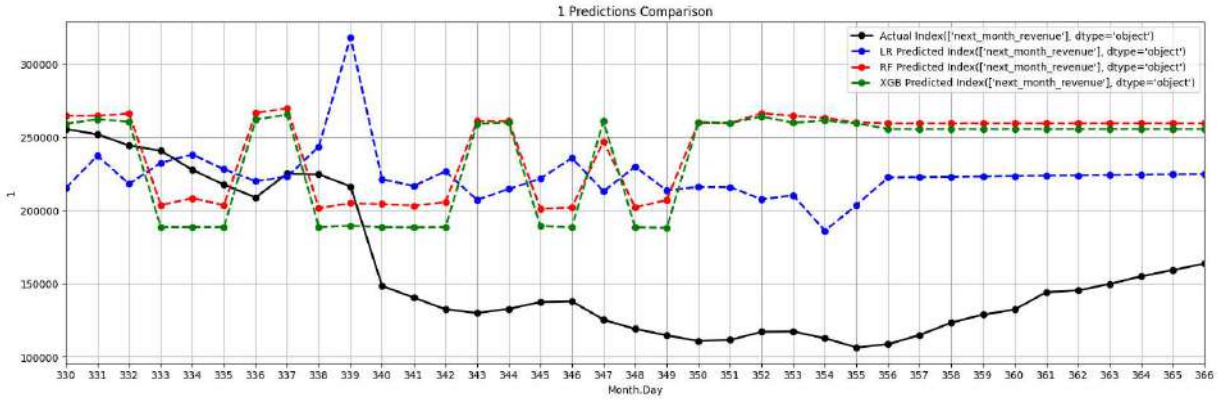
9 - Dataframe df_ts_singapore data:



10 - Dataframe df_ts_spain data:



11 - Dataframe df_ts_united_kingdom data:



In [ ]: