1. AirBNB:

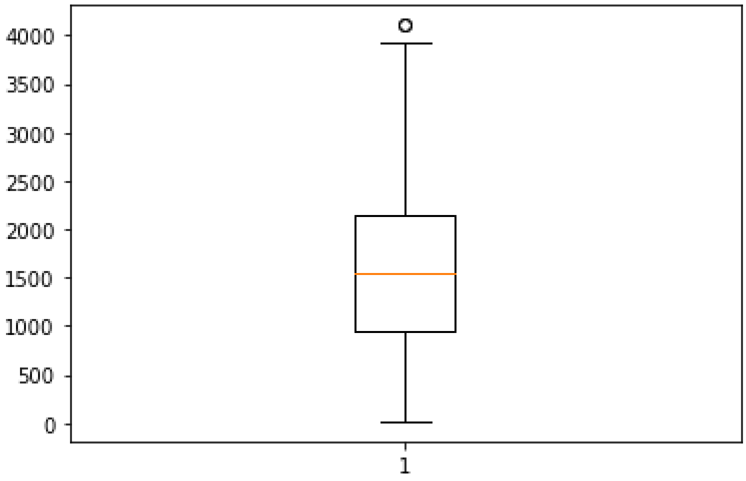
Q1.1 Boxplot

From the boxplot, we can usually infer Min, Max, Q1, Median, Q3, and Outliers information. From below boxplot I can see the Min vale is close to Zero, Max value is Close to 4000, Q1 is close to 1000, Median is close to 1500, Q3 is close to 2100, and there is an Outlier above 4000

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt

listing_host_start_df = pd.read_csv("C:/Users/satya/OneDrive - Texas State University/ShareKnowledge/Courses/QMST5336-ANA/Assignment/1/listing_host_start.csv",header = 0, delimiter = ",")
listing_host_start_df_null_dropped = listing_host_start_df.dropna(subset=['host_duration'])
plt.boxplot(listing_host_start_df_null_dropped['host_duration'])
plt.show()
```



Q1.2

Yes, there is an outlier based on TUKEY Method.

Step i: The Q1 value is 954.0 and Q3 value is 2156.0

Step ii: using step i calculate IQR(Q3-Q1) value 1202.0

Step iii: using step ii calculate upper boundary(Q3 + 1.5_IQR) and lower boundary(Q1 - 1.5_IQR)

Step iv: using step iii outliers are detected

In [2]:

```
from scipy import stats
import numpy as np

iqr_value = stats.iqr(listing_host_start_df_null_dropped['host_duration'], axis = 0)
Q1 = np.percentile(listing_host_start_df_null_dropped['host_duration'], 25, axis = 0)
Q3 = np.percentile(listing_host_start_df_null_dropped['host_duration'], 75, axis = 0)
upper_boundary_value = Q3 + 1.5 * iqr_value
lower_boundary_value = Q1 - (1.5 * iqr_value)
print(F"iqr_value value is: {iqr_value} \nQ1 value is: {Q1} \nQ3 val is {Q3} \nupper_boundary value is:{upper_boundary_value} \nlower_boundary value is:{lower_boundary_value}")
outliers = listing_host_start_df_null_dropped[(listing_host_start_df_null_dropped['host_duration'] > upper_boundary_value) | (listing_host_start_df_null_dropped['host_duration'] < lower_boundary_value)]
print("\n\nbelow are the outliers")
display(outliers)
```

```
iqr_value value is: 1202.0
Q1 value is: 954.0
Q3 val is 2156.0
upper_boundary value is:3959.0
lower_boundary value is:-849.0


below are the outliers
```

	host_id	host_since	host_is_superhost	update	start_month	host_duration
7720	23	2008-03-03	f	2019-05-31	3.0	4106.0
10827	23	2008-03-03	f	2019-05-31	3.0	4106.0

Q1.3

Mean and Standard deviation after removal of outlier.

In [3]:

```
import numpy as np

listing_host_start_df_null_dropped_without_outliers = listing_host_start_df_null_droppe
d[(listing_host_start_df_null_dropped['host_duration'] < upper_boundary_value) | (listi
ng_host_start_df_null_dropped['host_duration'] > lower_boundary_value)]

listing_host_start_df_null_dropped_without_outliers_mean = np.mean(listing_host_start_d
f_null_dropped_without_outliers['host_duration'])
print(F"The mean of hosting days :{listing_host_start_df_null_dropped_without_outliers_
mean}")

listing_host_start_df_null_dropped_without_outliers_std = np.std(listing_host_start_df_
null_dropped_without_outliers['host_duration'])
print(F"The standard deviation of hosting days :{listing_host_start_df_null_dropped_wit
hout_outliers_std}")
```

```
The mean of hosting days :1562.3848502841631
The standard deviation of hosting days :794.317815382655
```

Q1.4

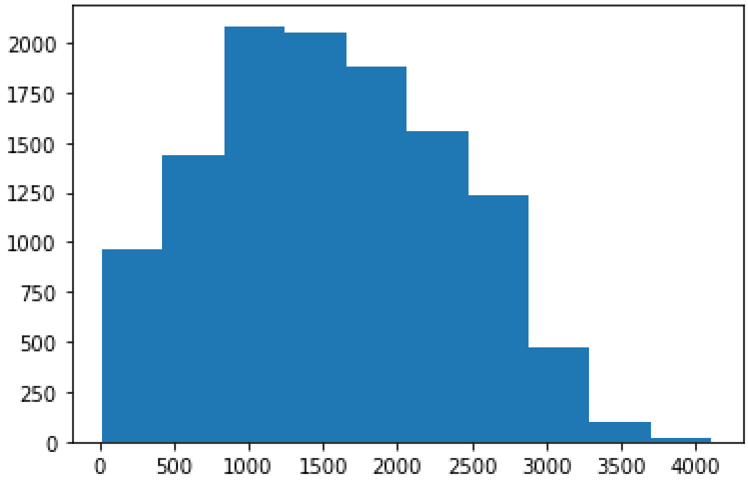
From the Histogram we can see a rightly skeweness (the values are spread to right)

In [4]:

```
import  matplotlib.pyplot  as plt
plt.hist(listing_host_start_df_null_dropped_without_outliers['host_duration'])
```

Out[4]:

```
(array([ 965., 1437., 2085., 2050., 1884., 1554., 1233.,  467.,   96.,
          18.]),
 array([  18. ,  426.8,  835.6, 1244.4, 1653.2, 2062. , 2470.8, 2879.6,
        3288.4, 3697.2, 4106. ]),
 <BarContainer object of 10 artists>)
```

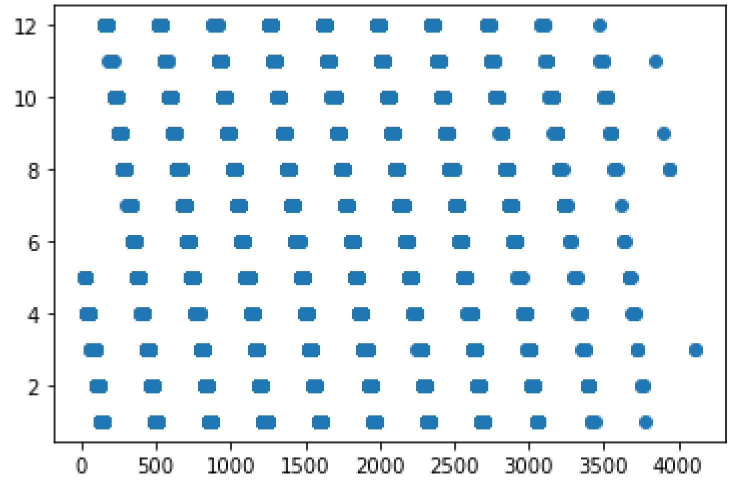


Q1.5 Scatterplot

From the Scatterplot we cannot see any association(linear or non-linear) and strength of relationship(positive or negative)

In [5]:

```
import  matplotlib.pyplot  as plt
plt.scatter(listing_host_start_df_null_dropped_without_outliers['host_duration'], listi
ng_host_start_df_null_dropped_without_outliers['start_month'])
plt.show()
```



2 Game of Thorns

Q2.1 Top 8 killers

In [6]:

```
import pandas as pd
game_of_thorns_df = pd.read_csv("C:/Users/satya/OneDrive - Texas State University/Share
Knowledge/Courses/QMST5336-ANA/Assignment/1/game-of-thrones-deaths-data.csv",header = 0
, delimiter = ",",encoding = "latin1")
top_8_killers_df = game_of_thorns_df.groupby('killer',as_index=False).count()[['killer'
,'character_killed']].sort_values(by='character_killed', axis=0, ascending=False)
top_8_killers_df = top_8_killers_df[top_8_killers_df['killer'] !='None']
print(top_8_killers_df.head(8))
```

```
               killer  character_killed
138             Wight              1602
29             Drogon              1426
5          Arya Stark              1278
103           Rhaegal               273
20    Cersei Lannister              199
57           Jon Snow               112
118     Stark soldier                96
14     Bolton soldier                91
```

Q2.2 word cloud of locations regarding the frequencies characters die

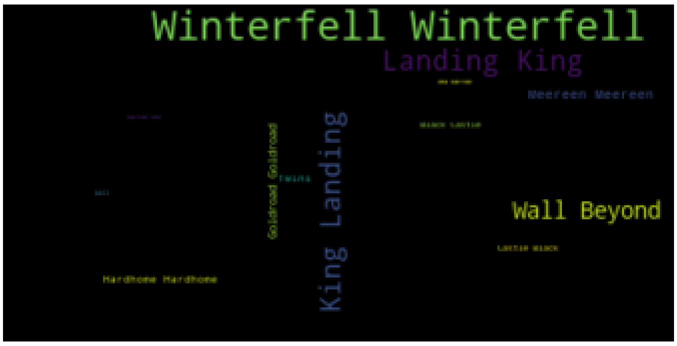
In [7]:

```
import os
from os import path
from wordcloud import WordCloud
import matplotlib.pyplot as plt

wordcloud_location = WordCloud().generate(game_of_thorns_df['location'].str.strip(to_st
rip=None).to_string( header=False, index=False))
plt.imshow(wordcloud_location, interpolation='bilinear')
plt.axis("off")
```

Out[7]:

```
(-0.5, 399.5, 199.5, -0.5)
```



In [ ]:

```

```