Web scraping and analysis ¶

Scraping data from Skytrax

If you visit [https://www.airlinequality.com] (https://www.airlinequality.com%5D) you can see that there is a lot of data there. For this task, we are only interested in reviews related to British Airways and the Airline itself.

If you navigate to this link: [https://www.airlinequality.com/airline-reviews/british-airways] (https://www.airlinequality.com/airline-reviews/british-airways%5D) you will see this data. Now, we can use `Python` and `BeautifulSoup` to collect all the links to the reviews and then to collect the text data on each of the individual review links.

#import the libraries

In [1]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd
```

In [2]:

```
base_url = "https://www.airlinequality.com/airline-reviews/british-airways"
pages = 10
page_size = 100

reviews = []

# for i in range(1, pages + 1):
for i in range(1, pages + 1):

    print(f"Scraping page {i}")

    # Create URL to collect links from paginated data
    url = f"{base_url}/page/{i}/?sortby=post_date%3ADesc&pagesize={page_size}"

    # Collect HTML data from this page
    response = requests.get(url)

    # Parse content
    content = response.content
    parsed_content = BeautifulSoup(content, 'html.parser')
    for para in parsed_content.find_all("div", {"class": "text_content"}):
        reviews.append(para.get_text())

    print(f"   ---> {len(reviews)} total reviews")
```

```
Scraping page 1
   ---> 100 total reviews
Scraping page 2
   ---> 200 total reviews
Scraping page 3
   ---> 300 total reviews
Scraping page 4
   ---> 400 total reviews
Scraping page 5
   ---> 500 total reviews
Scraping page 6
   ---> 600 total reviews
Scraping page 7
   ---> 700 total reviews
Scraping page 8
   ---> 800 total reviews
Scraping page 9
   ---> 900 total reviews
Scraping page 10
   ---> 1000 total reviews
```

In [3]:

```
df = pd.DataFrame()
df["reviews"] = reviews
df.head()
```

Out[3]:

	reviews
0	✅ Trip Verified | Couldn’t book in online. Ar...
1	✅ Trip Verified | London Heathrow to Mumbai in...
2	✅ Trip Verified | Keflavík, Iceland to London ...
3	✅ Trip Verified | Terrible Experience with Bri...
4	✅ Trip Verified | An airline that lives in the...

In [4]:

```
directory='.csv'
```

In [5]:
```
df.to_csv("britishairways_reviews12.csv")
```

In [6]:
```
len(df)
```

Out[6]:
```
1000
```

Congratulations! Now you have your dataset for this task! The loops above collected 1000 reviews by iterating through the paginated pages on the website. However, if you want to collect more data, try increasing the number of pages!

The next thing that you should do is clean this data to remove any unnecessary text from each of the rows. For example, "✅ Trip Verified" can be removed from each row if it exists, as it's not relevant to what we want to investigate.

In [7]:
```
df["reviews"] = df["reviews"].str.replace("✅ Trip Verified", "")
df["reviews"] = df["reviews"].str.replace("❌ Not Verified", "")
df["reviews"] = df["reviews"].str.replace("|", "")
df.head()
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_26484\1144794447.py:3: FutureWarning: The default value of regex
will change from True to False in a future version. In addition, single character regular expressions will *
not* be treated as literal strings when regex=True.
  df["reviews"] = df["reviews"].str.replace("|", "")
```

Out[7]:

	reviews
0	Couldn’t book in online. Arrived at check i...
1	London Heathrow to Mumbai in a Boeing 787-8 ...
2	Keflavík, Iceland to London Heathrow on an A...
3	Terrible Experience with British Airways. I ...
4	An airline that lives in their past glory an...

In [8]:
```
len(df)
```

Out[8]:
```
1000
```

#generating word cloud

In [9]:
```
from wordcloud import WordCloud
```

In [10]:
```
reviews_combined = " ".join(df.reviews.values)
```

In [11]:
```
reviews_combined
```

Out[11]:
```
'   Couldn’t book in online. Arrived at check in to find we had been bumped off due to overselling. No BA staff available. Very helpful Gatwick staff got us a bus to LHR and a flight to Toulouse. Had knock in effect on our car booking and sharing as the rest of family had been able yo board original flight. Airlines should be legally stopped from selling seats twice.   London Heathrow to Mumbai in a Boeing 787-8 in Business Class. The lounge near Terminal 5, Gate B36 at Heathrow was outstanding in its service and offerings. It provides us just the right frame to relax in before boarding as the departure was delayed by almost 2 hours. The 787-8 on our flight featured the older Club World seating. Not the best in class but comfortable enough. I hear that the new Club Suites configuration is far superior. British Airways onboard service was outstanding in every respect. All in all, a very comfortable flight. One minor irritant: for some reason this aircraft was not fitted with WiFi. We got into Mumbai at 8 am, a civilized time to arrive.   Keflavík, Iceland to London Heathrow on an A320 in Business Class. The journey got off on an unpleasant note - the Business Class line at Keflavík was so long that it looked like an Economy Class check-in. It took over 30 mins to get through. There was no lounge access offered. The boarding process was well handled. British Airways Business Class seats for the Club Europe product are terrible - exactly the same as Economy with the middle seat left vacant. You don\'t even get extra pitch. What made the overall product tolerable were the good onboard service and the inflight WiFi. Also the fact that the flight leaves at a convenient mid-morning time of 10:40 am.   Terrible Experience with British Airways. I booked a flight with BA to travel from Gibraltar to London Heathrow on May 10, 2023. My flight was scheduled to leave at 4:00 p.m. in the afternoon. I had originally planned on leaving my luggage at Heathrow upon arrival and travel
```

In [12]:
```
import re
reviews_combined = re.sub(r"\.",". ",reviews_combined)
reviews_combined = re.sub(r"[^\w\s]+"," ",reviews_combined)
```

In [13]:
```
1 reviews_combined
```

Out[13]:
```
'   Couldn t book in online   Arrived at check in to find we had been bumped off due to overselling   No
BA staff available   Very helpful Gatwick staff got us a bus to LHR and a flight to Toulouse   Had knock
in effect on our car booking and sharing as the rest of family had been able yo board original flight   A
irlines should be legally stopped from selling seats twice     London Heathrow to Mumbai in a Boeing 787
8 in Business Class   The lounge near Terminal 5  Gate B36 at Heathrow was outstanding in its service and
offerings   It provides us just the right frame to relax in before boarding as the departure was delayed
by almost 2 hours   The 787 8 on our flight featured the older Club World seating   Not the best in class
but comfortable enough   I hear that the new Club Suites configuration is far superior   British Airways
onboard service was outstanding in every respect   All in all  a very comfortable flight   One minor irri
tant  for some reason this aircraft was not fitted with WiFi   We got into Mumbai at 8 am  a civilized ti
me to arrive     Keflavík  Iceland to London Heathrow on an A320 in Business Class   The journey got off
on an unpleasant note   the Business Class line at Keflavík was so long that it looked like an Economy Cl
ass check in   It took over 30 mins to get through   There was no lounge access offered   The boarding pr
ocess was well handled   British Airways Business Class seats for the Club Europe product are terrible
exactly the same as Economy with the middle seat left vacant   You don t even get extra pitch   What made
the overall product tolerable were the good onboard service and the inflight WiFi   Also the fact that th
e flight leaves at a convenient mid morning time of 10 40 am     Terrible Experience with British Airways
I booked a flight with BA to travel from Gibraltar to London Heathrow on May 10  2023   My flight was sch
eduled to leave at 4 00 p  m   in the afternoon   I had originally planned on leaving my luggage at Heath
```

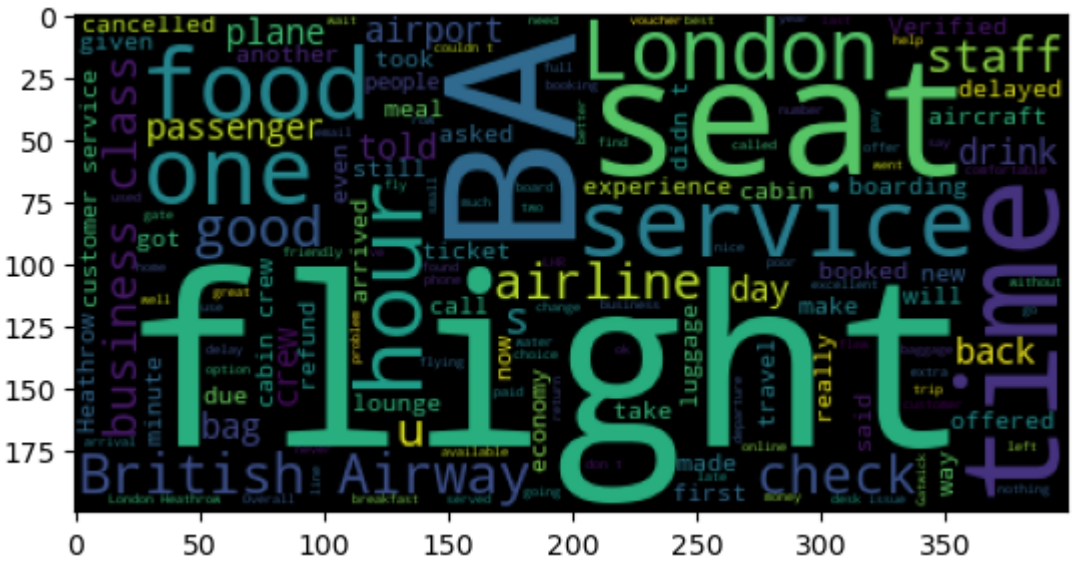
In [14]:
```
1 #Total words in reviews
2 len(reviews_combined)
```

Out[14]:
```
810567
```

In [15]:
```
1 word_cloud = WordCloud().generate_from_text(reviews_combined)
```

In [16]:
```
1 import matplotlib.pyplot as plt
```

In [17]:
```
plt.figure()
plt.imshow(word_cloud)
plt.show()
```



In [18]:
```
word_cloud = WordCloud(width=800,height=800,background_color='white',max_words=50,random_state = 46333, s
generate_from_text(reviews_combined)
```

In [19]:

```
plt.figure(figsize=[8,8])
plt.imshow(word_cloud)
plt.show()
```


Verified
British Airway
service
lounge
plane
hour
meal
business
class
even
London
good
passenger
cancelled
made
airport
flight
drink
first
s
got
booked
aircraft
Heathrow
bag
minute
another
food
u
cabin
BA
offered
boarding
really
seat
one
luggage
check
experience
now
will
told
crew
back
staff
cabin crew
customer service
time
delayed
airline
day
0
100
200
300
400
500
600
700

In [20]:
```
all_terms = []
fdist = {}
all_terms = reviews_combined.split(" ")
for word in all_terms:
    fdist[word] = fdist.get(word,0) + 1
```

In [21]:
```
fdist["will"]
```

Out[21]: 217

In [22]:
```
fdist["good"]
```

Out[22]: 386

In [23]:
```
fdist["bad"]
```

Out[23]: 83

In [24]:
```
print(all_terms)
```

```
['', '', '', 'Couldn', 't', 'book', 'in', 'online', '', '', 'Arrived', 'at', 'check', 'in', 'to', 'find',
'we', 'had', 'been', 'bumped', 'off', 'due', 'to', 'overselling', '', '', 'No', 'BA', 'staff', 'availabl
e', '', '', 'Very', 'helpful', 'Gatwick', 'staff', 'got', 'us', 'a', 'bus', 'to', 'LHR', 'and', 'a', 'fli
ght', 'to', 'Toulouse', '', '', 'Had', 'knock', 'in', 'effect', 'on', 'our', 'car', 'booking', 'and', 'sh
aring', 'as', 'the', 'rest', 'of', 'family', 'had', 'been', 'able', 'yo', 'board', 'original', 'flight',
'', '', 'Airlines', 'should', 'be', 'legally', 'stopped', 'from', 'selling', 'seats', 'twice', '', '',
'', '', 'London', 'Heathrow', 'to', 'Mumbai', 'in', 'a', 'Boeing', '787', '8', 'in', 'Business', 'Class',
'', '', 'The', 'lounge', 'near', 'Terminal', '5', '', 'Gate', 'B36', 'at', 'Heathrow', 'was', 'outstandin
g', 'in', 'its', 'service', 'and', 'offerings', '', '', 'It', 'provides', 'us', 'just', 'the', 'right',
'frame', 'to', 'relax', 'in', 'before', 'boarding', 'as', 'the', 'departure', 'was', 'delayed', 'by', 'al
most', '2', 'hours', '', '', 'The', '787', '8', 'on', 'our', 'flight', 'featured', 'the', 'older', 'Clu
b', 'World', 'seating', '', '', 'Not', 'the', 'best', 'in', 'class', 'but', 'comfortable', 'enough', '',
'', 'I', 'hear', 'that', 'the', 'new', 'Club', 'Suites', 'configuration', 'is', 'far', 'superior', '',
'', 'British', 'Airways', 'onboard', 'service', 'was', 'outstanding', 'in', 'every', 'respect', '', '',
'All', 'in', 'all', '', 'a', 'very', 'comfortable', 'flight', '', '', 'One', 'minor', 'irritant', '', 'fo
r', 'some', 'reason', 'this', 'aircraft', 'was', 'not', 'fitted', 'with', 'WiFi', '', '', 'We', 'got', 'i
nto', 'Mumbai', 'at', '8', 'am', '', 'a', 'civilized', 'time', 'to', 'arrive', '', '', '', '', 'Keflaví
k', '', 'Iceland', 'to', 'London', 'Heathrow', 'on', 'an', 'A320', 'in', 'Business', 'Class', '', '', 'Th
e', 'journey', 'got', 'off', 'on', 'an', 'unpleasant', 'note', '', '', 'the', 'Business', 'Class', 'lin
```

In [25]:
```
freq = {"words":list(fdist.keys()),"freq":list(fdist.values())}
df_dist = pd.DataFrame(freq)
```

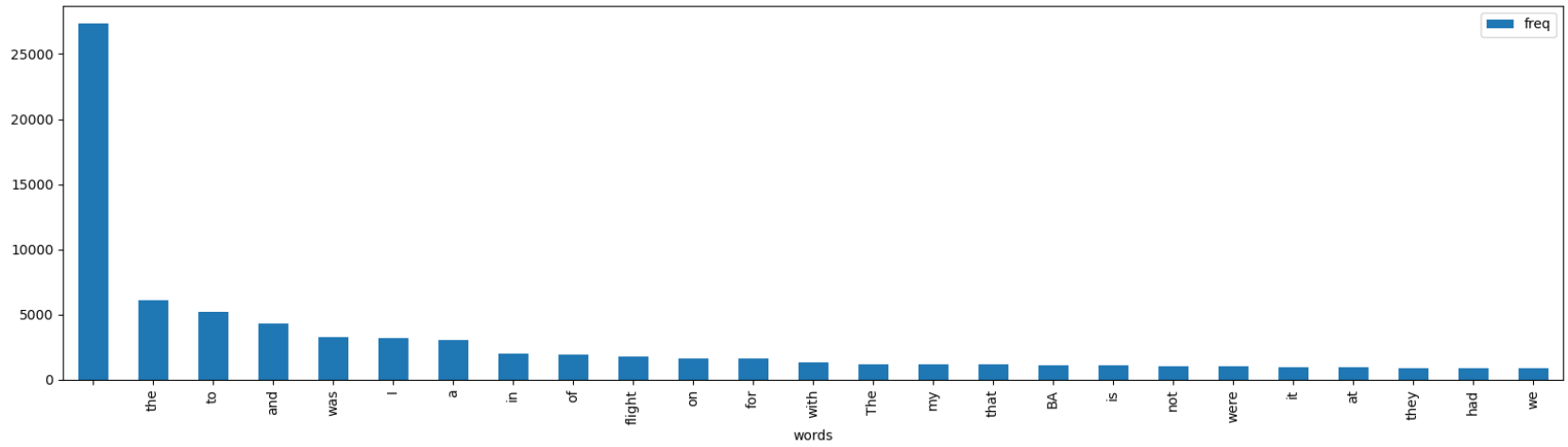
In [26]:
```
df_dist.head(10)
```

Out[26]:

	words	freq
0		27332
1	Couldn	9
2	t	602
3	book	67
4	in	2024
5	online	92
6	Arrived	18
7	at	933
8	check	305
9	to	5189

In [27]:
```
%matplotlib inline
df_dist.sort_values(ascending=False, by="freq").head(25).\
plot.bar(x= "words", y= "freq",figsize=(20,5))
```

Out[27]: <AxesSubplot:xlabel='words'>



Problems with the above visuals 1.case is non uniform 2. punctuations and stop words are present

In [28]:
```
#1. case normalization and tokenizing
```

In [29]:
```
df.reviews[:10]
```

Out[29]:
```
0       Couldn’t book in online. Arrived at check i...
1      London Heathrow to Mumbai in a Boeing 787-8 ...
2      Keflavík, Iceland to London Heathrow on an A...
3      Terrible Experience with British Airways. I ...
4      An airline that lives in their past glory an...
5       Check-in Desk rude and dismissive. Flight l...
6      I chose British Airways especially because I...
7    Not Verified  I booked Premium Economy from IN...
8      A simple story with an unfortunate outcome t...
9       Flight was delayed due to the inbound fligh...
Name: reviews, dtype: object
```

In [30]:
```
from nltk.tokenize import word_tokenize
```

In [31]:
```
#All Reviews tokenized and in lower case
all_terms = word_tokenize(reviews_combined.lower())
```

In [32]:
```
print(all_terms)
```

```
['couldn', 't', 'book', 'in', 'online', 'arrived', 'at', 'check', 'in', 'to', 'find', 'we', 'had', 'bee
n', 'bumped', 'off', 'due', 'to', 'overselling', 'no', 'ba', 'staff', 'available', 'very', 'helpful', 'ga
twick', 'staff', 'got', 'us', 'a', 'bus', 'to', 'lhr', 'and', 'a', 'flight', 'to', 'toulouse', 'had', 'kn
ock', 'in', 'effect', 'on', 'our', 'car', 'booking', 'and', 'sharing', 'as', 'the', 'rest', 'of', 'famil
y', 'had', 'been', 'able', 'yo', 'board', 'original', 'flight', 'airlines', 'should', 'be', 'legally', 's
topped', 'from', 'selling', 'seats', 'twice', 'london', 'heathrow', 'to', 'mumbai', 'in', 'a', 'boeing',
'787', '8', 'in', 'business', 'class', 'the', 'lounge', 'near', 'terminal', '5', 'gate', 'b36', 'at', 'he
athrow', 'was', 'outstanding', 'in', 'its', 'service', 'and', 'offerings', 'it', 'provides', 'us', 'jus
t', 'the', 'right', 'frame', 'to', 'relax', 'in', 'before', 'boarding', 'as', 'the', 'departure', 'was',
'delayed', 'by', 'almost', '2', 'hours', 'the', '787', '8', 'on', 'our', 'flight', 'featured', 'the', 'ol
der', 'club', 'world', 'seating', 'not', 'the', 'best', 'in', 'class', 'but', 'comfortable', 'enough',
'i', 'hear', 'that', 'the', 'new', 'club', 'suites', 'configuration', 'is', 'far', 'superior', 'british',
'airways', 'onboard', 'service', 'was', 'outstanding', 'in', 'every', 'respect', 'all', 'in', 'all', 'a',
'very', 'comfortable', 'flight', 'one', 'minor', 'irritant', 'for', 'some', 'reason', 'this', 'aircraft',
'was', 'not', 'fitted', 'with', 'wifi', 'we', 'got', 'into', 'mumbai', 'at', '8', 'am', 'a', 'civilized',
'time', 'to', 'arrive', 'keflavík', 'iceland', 'to', 'london', 'heathrow', 'on', 'an', 'a320', 'in', 'bus
iness', 'class', 'the', 'journey', 'got', 'off', 'on', 'an', 'unpleasant', 'note', 'the', 'business', 'cl
ass', 'line', 'at', 'keflavík', 'was', 'so', 'long', 'that', 'it', 'looked', 'like', 'an', 'economy', 'cl
ass', 'check', 'in', 'it', 'took', 'over', '30', 'mins', 'to', 'get', 'through', 'there', 'was', 'no', 'l
```

In [33]:
```
len(all_terms)
```

Out[33]: 147087

In [34]:
```
len(set(all_terms))
```

Out[34]: 7623

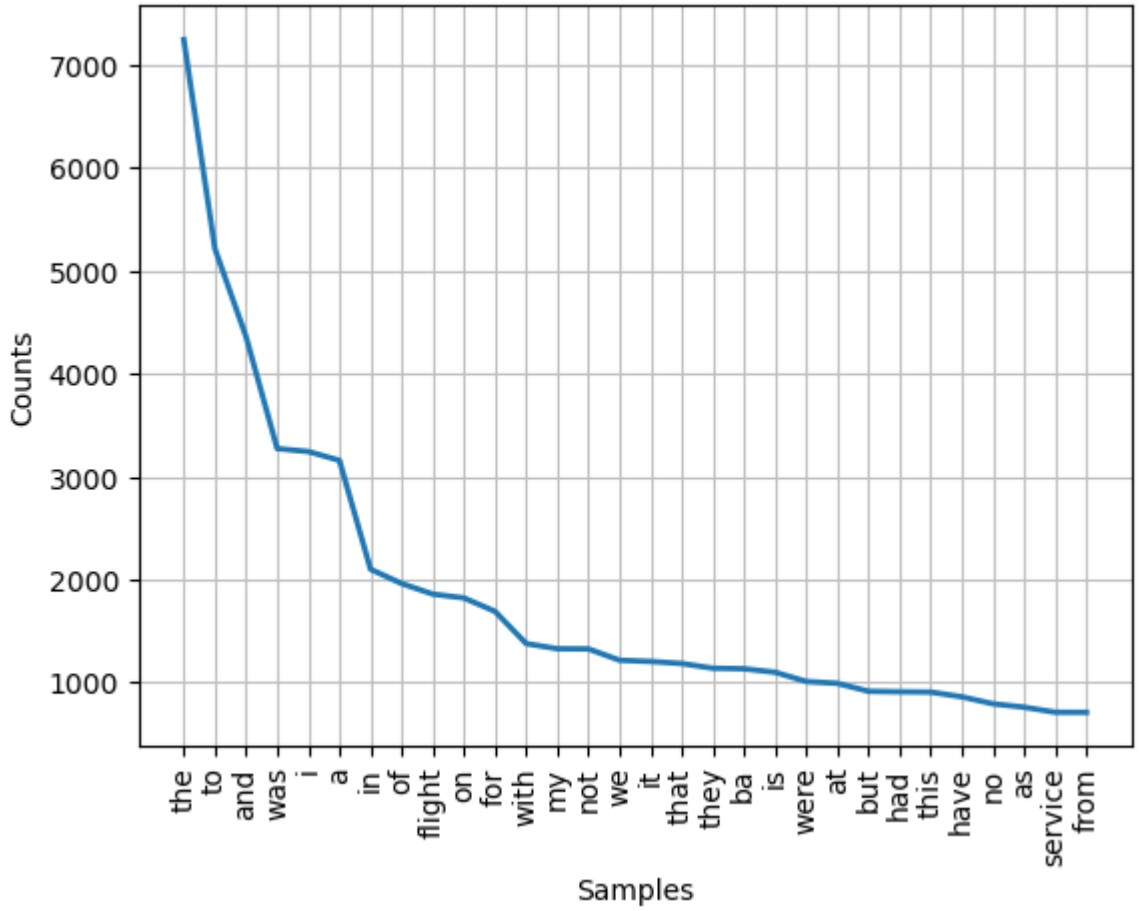
In [35]:
```
#visualizing the frequency distribution
```

In [36]:
```
from nltk.probability import FreqDist
```

In [37]:
```
fdist = FreqDist(all_terms)
fdist
```

Out[37]:
```
FreqDist({'the': 7250, 'to': 5216, 'and': 4360, 'was': 3272, 'i': 3243, 'a': 3157, 'in': 2098, 'of': 1959,
'flight': 1856, 'on': 1819, ...})
```

In [38]:
```
fdist.plot(30,cumulative=False)
plt.show()
```



In [39]:
```
from string import punctuation
from nltk.corpus import stopwords
```

In [40]:
```
stop_nltk = stopwords.words("english")
```

In [41]:
```
print(stop_nltk)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd",
'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'he
rself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'wh
o', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'be
ing', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'o
r', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'int
o', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on',
'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how',
'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'o
wn', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "shoul
d've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn',
"didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma',
'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'was
n', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

In [42]:
```
import re
def clean_text(text):
    # Remove special characters and symbols (keep only alphabets, numbers, and spaces)
    cleaned_text = re.sub(r"[^a-zA-Z0-9\s]", "", text)

    # Convert to lowercase
    cleaned_text = cleaned_text.lower()

    return cleaned_text

# Clean the reviews using the clean_text function
df["Cleaned_Review"] = df["reviews"].apply(clean_text)
```

In [43]:
```
df["Cleaned_Review"]
```

Out[43]:
```
0         couldnt book in online arrived at check in ...
1        london heathrow to mumbai in a boeing 7878 i...
2        keflavk iceland to london heathrow on an a32...
3        terrible experience with british airways i b...
4        an airline that lives in their past glory an...
                             ...
995    not verified  dublin to london i was trying to...
996      london  pisa return i fly this route often a...
997       i was in prague flying british airways back...
998      \r\nba34 kullhr 6 sept return ba11 lhrsinkul...
999      we flew from los angeles to leeds bradford v...
Name: Cleaned_Review, Length: 1000, dtype: object
```

In [44]:
```
len(df)
```

Out[44]:
```
1000
```

In [45]:
```
def classify_sentiment(review):
    positive_keywords = ["good", "fantastic", "wonderful", "amazing","nice","pleasure","delight","lovely"
    negative_keywords = ["bad", "negative", "improve service","worse","bad quality","improve","cancelled"

    review = review.lower()

    if any(keyword in review for keyword in positive_keywords):
        return 1  # Positive sentiment
    elif any(keyword in review for keyword in negative_keywords):
        return 0  # Negative sentiment
    else:
        return -1  # neutral

# Apply the classify_sentiment function to each row to get the sentiment classification
df["Sentiment"] = df["Cleaned_Review"].apply(classify_sentiment)
```

In [46]:
```
1 df["Sentiment"].value_counts()
```

Out[46]:
```
 1    535
 0    349
-1    116
Name: Sentiment, dtype: int64
```

In [47]:
```
1 terms_updated = [word for word in all_terms if word not in stop_nltk and len(word) > 1]
```

In [48]:
```
1 print(terms_updated[:20])
```

```
['book', 'online', 'arrived', 'check', 'find', 'bumped', 'due', 'overselling', 'ba', 'staff', 'available',
'helpful', 'gatwick', 'staff', 'got', 'us', 'bus', 'lhr', 'flight', 'toulouse']
```

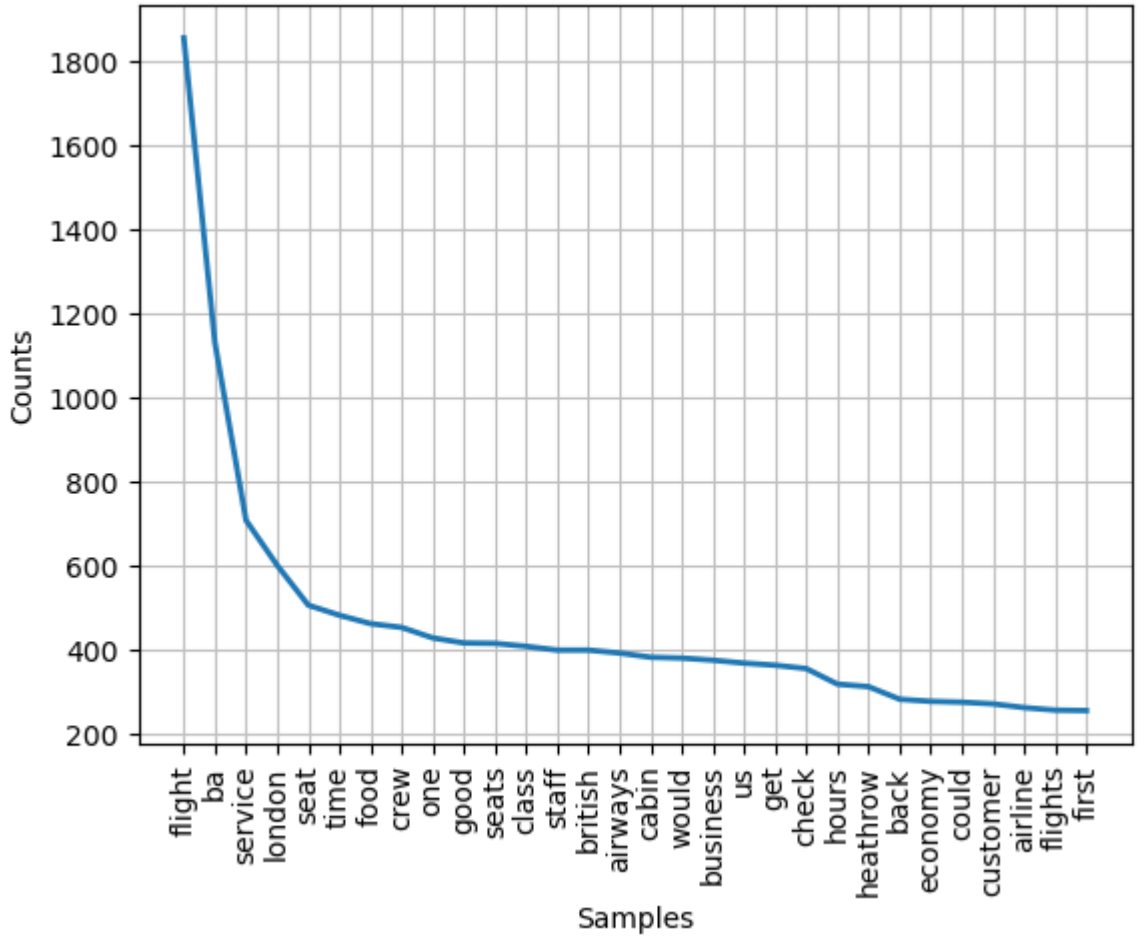
In [49]:
```
1 len(set(terms_updated))
```

Out[49]:
```
7457
```

In [50]:
```
1 fdist = FreqDist(terms_updated)
2 fdist
```

Out[50]:
```
FreqDist({'flight': 1856, 'ba': 1130, 'service': 707, 'london': 600, 'seat': 505, 'time': 481, 'food': 461,
'crew': 452, 'one': 427, 'good': 415, ...})
```

In [51]:
```
1 fdist.plot(30,cumulative=False)
2 plt.show()
```



In [52]:
```
1 df["Sentiment"].value_counts()
```

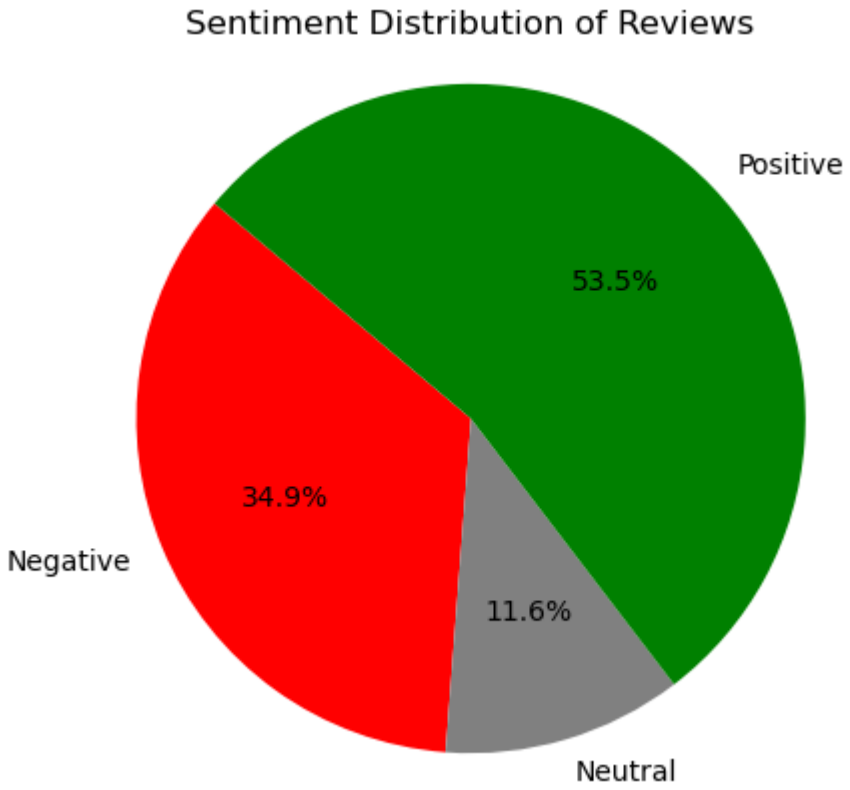
Out[52]:
```
 1    535
 0    349
-1    116
Name: Sentiment, dtype: int64
```

In [53]:

```
new_df = df[["Cleaned_Review", "Sentiment"]]

# Plot the pie chart to show the distribution of sentiments
sentiment_counts = new_df["Sentiment"].value_counts()
labels = ['Negative', 'Neutral', 'Positive']
sizes = [sentiment_counts[0], sentiment_counts[-1], sentiment_counts[1]]
colors = ['red', 'gray', 'green']
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)
plt.axis('equal')
plt.title('Sentiment Distribution of Reviews')
plt.show()


```



In [54]:
```
# Calculate the total percentage of positive, negative, and neutral reviews
total_reviews = len(new_df)
positive_percentage = (sentiment_counts[1] / total_reviews) * 100
negative_percentage = (sentiment_counts[0] / total_reviews) * 100
neutral_percentage = (sentiment_counts[-1] / total_reviews) * 100


```

In [55]:
```
print(f"Positive Reviews: {positive_percentage:.2f}%")
print(f"Negative Reviews: {negative_percentage:.2f}%")
print(f"Neutral Reviews: {neutral_percentage:.2f}%")
```

```
Positive Reviews: 53.50%
Negative Reviews: 34.90%
Neutral Reviews: 11.60%
```

In [ ]:
```

```