rev for	<pre>soup = BeautifulSoup(response.text, 'html.parser') review_rating_header = set() for i in soup.find_all('td', class_='review-rating-header'): review_rating_header.add(i.text) lists_dict = {} for h in review_rating_header: lists_dict[h] = [] print(review_rating_header) print(lists_dict)</pre>	
ls_ ls_ ls_ 'Se 'Se		
	<pre>for page in range(1, max_page+1): url = "https://www.airlinequality.com/airline-reviews/british-airways/page/"+str(page)+"/?sortby=post_date%3ADesc&pagesize=100" response = requests.get(url) soup = BeautifulSoup(response.text, 'html.parser') print(url) for content in soup.find_all('article', itemprop="review"): ratingValue = content.find('span', itemprop="ratingValue") if ratingValue != None:</pre>	
#	ratingValue = int(ratingValue.text) else: ratingValue = np.nan ls_ratingValue.append(ratingValue) # print('Score:', ratingValue) user_loc = content.find('h3', class_="text_sub_header").text if "(" in user_loc: user_loc = user_loc.split("(")[1].split(")")[0] else: user_loc = np.nan # ls_user_loc.append(user_loc) print('User Location:', user_loc)	
<pre>trip_verified_tag = soup.find('a', href=lambda href: href and 'verified-reviews' in href) if trip_verified_tag: trip_verified = trip_verified_tag.text.strip() ls_trip_verified.append(trip_verified) else: ls_trip_verified.append(np.nan) # print("") for lists_dict_element in lists_dict: flag=1 for i in content.find_all('td', class_='review-rating-header'): if(lists_dict_element == i.text):</pre>		
#		
df df df	<pre>df = pd.DataFrame() df['ratingValue'] = ls_ratingValue df['user_loc'] = ls_user_loc df['trip_verified'] = ls_trip_verified for i in review_rating_header: df[i] = lists_dict[i]</pre> EDA	
df	rating Value user_loc trip_verified Seat Comfort Recommended Wiff & Connectivity Food & Beverages Ground Service Seat Type Aircraft Inflight Entertainment Value For Money Date Flow Cabin Staff Service 1 9.0 Iceland Trip Verified 3.0 yes NaN Yer Seat Office Office Seat	Gatwick to Montpelier NaN Couple Leisure NaN London to Mumbai NaN Couple Leisure NaN
361 361 361	1.0 Qatar Trip Verified 1.0 no NaN NaN 1.0 Business Class A321 NaN 1 July 2023 1.0	NaN
cla at = -0 12345678910112345617 115617 pp	Calums (core.frame.DataFrame'> Angelindex: 3615 entries, 0 to 3614	
1 3 4 2 5 N/A df Tr: Ve No	df['Inflight Entertainment'].value_counts() 1 768 3 541 4 461 2 427 5 312 N/A 10 Name: Inflight Entertainment, dtype: int64 df.replace('N/A', np.nan, inplace=True) df['trip_verified'].value_counts() Trip Verified 900 Verified Review 900 Not Verified 500 Name: trip_verified, dtype: int64 df['trip_verified'].replace('Verified Review', 'Trip Verified', inplace=True)	
df	ratingValue user_loc trip_verified Seat Comfort Recommended Wifi & Connectivity Food & Beverages Ground Service Seat Type Aircraft Inflight Entertainment Value For Money Date Flow Cabin Staff Service 1 9.0 United Kingdom Trip Verified 3.0 yes NaN 9.0 NaN 1.0 Na	Gatwick to Montpelier Couple Leisure London to Mumbai Couple Leisure
361 361 361	1.0 Qatar Trip Verified 1.0 no NaN NaN 1.0 Business Class A321 NaN 1.0 July 2023 1.0	NaN NaN NaN NaN NaN NaN NaN NaN NaN
df. ra us tr Se Re Wi Fo Gr Se Ai In Va Da Ca Ro Ty	df.isna().sum() ratingValue 5 user_loc 2 trip_verified 1315 Seat Comfort 108 Recommended 0 Wifi & Connectivity 3038 Food & Beverages 366 Ground Service 842 Seat Type 2 Aircraft 1739 Inflight Entertainment 1106 Value For Money 1 Inflight Entertainment 1106 Value For Money 1 Date Flown 778 Cabin Staff Service 118 Route 775 Type 0f Traveller 776	
Ty dt dt df df claangata	Type of Traveller 770 dtype: int64 for i in df.columns: if i in ['ratingValue', 'Seat Comfort', 'Wifi & Connectivity', 'Food & Beverages', 'Ground Service', 'Inflight Entertainment', 'Value For Money', 'Cabin Staff Service']: df[i].fillna(0, inplace=True) df[i] = df[i].astype('int') df.info() class 'pandas.core.frame.DataFrame'> angeIndex: 3615 entries, 0 to 3614 ata columns (total 16 columns):	
# 0 1 2 3 4 5 6 7 8 9 1 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 1 1 1	# Column Non-Null Count Dtype	
Sns plt plt	Data Visualization sns.kdeplot(df['ratingValue'], shade=True) plt.title('Distribution of Ratings') plt.show() Distribution of Ratings 0.175	
	0.150 - 0.125 - 0.100 -	
plt plt	-2 0 2 4 6 8 10 12 ratingValue plt.pie(x=df['trip_verified'].value_counts().tolist(), labels=df['trip_verified'].value_counts().keys().tolist()) plt.title('verified vs Non-verified users') plt.show() Verified vs Non-verified users rip Verified	
plt	Not Verified Sns.barplot(data=df, x='Seat Type', y='ratingValue') plt.title('Average Ratings by Type of Seats') plt.show()	
6 5 5 4 4 A	Average Ratings by Type of Seats 6 - 5 - 4 - 4 - 4 - 4 - 4 - 4 - 4 - 4 - 4	
2 1 0 plt my_ p=r p.(plt.pie(x=df['Recommended'].value_counts().tolist(), labels=df['Recommended'].value_counts().keys().tolist()) my_circl=plt.circle((0,0), 0.7, color='white') p=plt.gef() p, gca().add_artist(my_circle)	
p.g plt		
plt plt	sns.boxplot(data=df[df['ratingValue']==9], x='trip_verified', y='ratingValue') plt.title('Distribution of Ratings by Verified and Non-verified passengers') plt.show() Distribution of Ratings by Verified and Non-verified passengers	
latingvalue	8 - Box 6 - A - A - A - A - A - A - A - A - A -	
plt plt	Sums. countplot(data=df, x='Type Of Traveller') plt.title('Number of Types of Traveller') plt.show() Number of Types of Traveller 800 - 600 -	
sns plt	Couple Leisure Solo Leisure Family Leisure Business Type Of Traveller sns.boxplot(data=df, x='Type Of Traveller', y='ratingValue') plt.title('bistribution of Ratings by Type of traveller') plt.show()	
plt 1	Distribution of Ratings by Type of traveller 10	
df	Couple Leisure Solo Leisure Family Leisure Business ratingValue user_loc trip_verified Seat Comfort Recommended Wifi & Connectivity Food & Beverages Ground Service Seat Type Aircraft Inflight Entertainment Value For Money Date Flow Cabin Staff Service O 1 United Kingdom Trip Verified 3 no 0 1 1 1 Economy Class NaN 0 1 1 July 2023 3 1 9 Iceland Trip Verified 4 yes 0 5 4 Business Class Boeing 787-8 3 4 July 2023 5 5 2 6 Iceland Trip Verified 3 yes 4 4 4 4 4 5 8 Business Class A320 0 3 3 July 2023 5 5	Gatwick to Montpelier Couple Leisure London to Mumbai Couple Leisure Keflavik to London Couple Leisure
361 361 361	3 1 Canada Trip Verified 1 no 0 0 1 Economy Class NaN 0 1 May 2023 1 4 1 Qatar Trip Verified 1 no 0 0 1 Business Class A321 0 1 July 2023 1	Gibraltar to London Heathrow Solo Leisure Madrid to London Family Leisure NaN NaN NaN NaN NaN NaN NaN NaN NaN
361 61 sns	3613 4 United States NaN	
nfli		
plt plt	sns.heatmap(df.corr()['ratingvalue'].iloc[i:].to_frame(), annot=True) alt_title('Correlation of variables for Rating') alt_show() Correlation of variables for Rating Seat Comfort - 0.74 - 0.8 Wifi & Connectivity - 0.051 - 0.7 Food & Beverages - 0.74 - 0.6 Ground Service - 0.36 - 0.5 Afflight Entertainment - 0.4 Value For Money - 0.87	
sns plt plt 9	Cabin Staff Service - 0.73 ratingValue sns.lineplot(data=df, x='Seat Comfort', y='ratingValue') plt.title('Correlation of Rating with Seat Comfort') plt.show() Correlation of Rating with Seat Comfort 9 8 - 7- 6 6 -	
3 2 1 for	for i in ['Seat Comfort', 'Wifi & Connectivity', 'Food & Beverages', 'Ground Service', 'Inflight Entertainment', 'Value For Money', 'Cabin Staff Service']: sns.lineplot(data=df, x=i, y='ratingValue', label=i) plt.xlabel('All variables')	
plt	plt. xiabel('All variables') plt. show() Correlation of Rating with all variables 9	
4	3 -	