Introduction

We reglarly hear in the news and social media that illegal migrants and asylum seekers to the UK are mainly men of fighting age and that most are arriving by small boats across the channel from France.

This analysis investigates the UK official statistics on migration from the UK Government to identify whether these assertions are indeed true.

Source Data

Irregular Migration Statistics Data

The data is from Quarter 1 from 2018 to Quarter 1 2024 which is the latest available at the time of the analysis.

Notes

1. Data on small boat arrivals relates to any individual who is either (a) detected on arrival to the UK, or (b) detected in the Channel by UK authorities and subsequently brought to the UK, having travelled across the English channel in a small boat.
2. Data on inadequately documented air arrivals relates to recorded detections of individuals who intentionally attempt to arrive in the UK through air routes either without adequate documentation or using fraudulent documentation. This does not include passengers who have genuinely made a mistake, lost their documents, or have arrived inadequately documented for reasons outside of their control. It does not include those who were prevented from boarding at their port of embarkation.
3. Data on recorded detections in the UK relates to recorded detections of individuals outside of the controlled environment of a port, who when encountered are believed by authorities to have evaded border controls to enter the UK irregularly, up to 72 hours before being detected. This includes detections at the ports serviced by the juxtaposed controls (Dover, Cheriton/Longport and St Pancras).
4. Data on recorded detections at UK ports relates to recorded detections of individuals attempting to enter the UK irregularly at ports. This includes, for example, those detected at ports in lorries, and shipping containers. It does not include detections at ports that are serviced by the juxtaposed controls (Dover, Cheriton/Longport and St Pancras) or those prevented from leaving their port of embarkation en route to the UK, such as those detected at the juxtaposed controls in France and Belgium.

Import libraries and data

In [12]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly
```

```
import plotly.express as px
import warnings

warnings.filterwarnings("ignore")
```

In [13]:
```
df = pd.read_csv(r'C:\Users\imoge\AllMLProjects\Data\UK_irregular_migration_updated.csv
```

In [14]:
```
df.shape
```

Out[14]:
```
(13406, 10)
```

In [15]:
```
df.head()
```

Out[15]:

	Year	Quarter	Method of entry	Nationality	Region	Sex	Age Group	Number of detections	Unnamed: 8	Unnamed: 9
0	2018	2018 Q1	Inadequately documented air arrivals	Afghanistan	Asia Central	Female	17 and under	14	NaN	NaN
1	2018	2018 Q1	Inadequately documented air arrivals	Afghanistan	Asia Central	Female	18 to 24	4	NaN	NaN
2	2018	2018 Q1	Inadequately documented air arrivals	Afghanistan	Asia Central	Female	25 to 39	14	NaN	NaN
3	2018	2018 Q1	Inadequately documented air arrivals	Afghanistan	Asia Central	Female	40 and over	22	NaN	NaN
4	2018	2018 Q1	Inadequately documented air arrivals	Afghanistan	Asia Central	Male	17 and under	21	NaN	NaN

In [16]:
```
df.tail()
```

Out[16]:

	Year	Quarter	Method of entry	Nationality	Region	Sex	Age Group	Number of detections	Unnamed: 8	Unnamed: 9
13401	2025	2025 Q1	Small boat arrivals	Yemen	Middle East	Female	40 and over	3	NaN	NaN
13402	2025	2025 Q1	Small boat arrivals	Yemen	Middle East	Male	17 and under	3	NaN	NaN
13403	2025	2025 Q1	Small boat arrivals	Yemen	Middle East	Male	18 to 24	59	NaN	NaN

	Year	Quarter	Method of entry	Nationality	Region	Sex	Age Group	Number of detections	Unnamed: 8	Unnamed: 9
13404	2025	2025 Q1	Small boat arrivals	Yemen	Middle East	Male	25 to 39	173	NaN	NaN
13405	2025	2025 Q1	Small boat arrivals	Yemen	Middle East	Male	40 and over	28	NaN	NaN

In [17]:
```
# Drop empty columns
df.drop(columns = ['Unnamed: 8','Unnamed: 9'],axis = 1, inplace = True)
```

In [18]:
```
# Check datatypes
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13406 entries, 0 to 13405
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   Year                  13406 non-null  int64 
 1   Quarter               13406 non-null  object
 2   Method of entry       13406 non-null  object
 3   Nationality           13406 non-null  object
 4   Region                13406 non-null  object
 5   Sex                   13406 non-null  object
 6   Age Group             13406 non-null  object
 7   Number of detections  13406 non-null  object
dtypes: int64(1), object(7)
memory usage: 838.0+ KB
```

There are some features that we need to convert to different datatypes for analysis

In [19]:
```
# Confirm nulls
df.isnull().sum()
```

Out[19]:
```
Year                  0
Quarter               0
Method of entry       0
Nationality           0
Region                0
Sex                   0
Age Group             0
Number of detections  0
dtype: int64
```

There are no null values

In [20]:
```
print(df['Year'].unique())
```

```
[2018 2019 2020 2021 2022 2023 2024 2025]
```

We have data from 2018 to 2025

In [21]:
```
print(df['Quarter'].unique())
```

```
['2018 Q1' '2018 Q2' '2018 Q3' '2018 Q4' '2019 Q1' '2019 Q2' '2019 Q3'
 '2019 Q4' '2020 Q1' '2020 Q2' '2020 Q3' '2020 Q4' '2021 Q1' '2021 Q2'
 '2021 Q3' '2021 Q4' '2022 Q1' '2022 Q2' '2022 Q3' '2022 Q4' '2023 Q1'
 '2023 Q2' '2023 Q3' '2023 Q4' '2024 Q1' '2024 Q2' '2024 Q3' '2024 Q4'
 '2025 Q1']
```

The data is from first quarter of 2018 to first quarter of 2025. We will discount the first quarter of 2025 so that we have seven full years of data.

In [22]:
```
df = df[df['Year']!=2025]
```

In [23]:
```
print(df['Method of entry'].unique())
```

```
['Inadequately documented air arrivals' 'Recorded detections at UK ports'
 'Recorded detections in the UK' 'Small boat arrivals']
```

There are four categories of irregular arrivals

In [24]:
```
print(df['Nationality'].unique())
```

```
['Afghanistan' 'Albania' 'Algeria' 'Angola' 'Brazil'
 'British overseas citizens' 'Burundi' 'Cameroon' 'Canada' 'Chad' 'Chile'
 'China' 'Colombia' 'Congo' 'Congo (Democratic Republic)'
 'Dominican Republic' 'Egypt' 'Eritrea' 'Ethiopia' 'Gambia, The' 'Georgia'
 'Ghana' 'Greece' 'Guinea-Bissau' 'India' 'Iran' 'Iraq' 'Italy' 'Kuwait'
 'Latvia' 'Lebanon' 'Moldova' 'Nepal' 'Netherlands' 'Nigeria'
 'North Macedonia' 'Occupied Palestinian Territories' 'Other and unknown'
 'Pakistan' 'Poland' 'Russia' 'Sierra Leone' 'Slovenia' 'Somalia'
 'South Africa' 'South Korea' 'Sri Lanka' 'Stateless' 'Sudan' 'Syria'
 'Taiwan' 'Tanzania' 'Turkey' 'Ukraine' 'United States' 'Vietnam' 'Yemen'
 'Libya' 'Morocco' 'Guinea' 'Ivory Coast' 'Kyrgyzstan' 'Liberia'
 'South Sudan' 'Spain' 'Uganda' 'Argentina' 'Bangladesh'
 'Bosnia and Herzegovina' 'Denmark' 'France' 'Israel' 'Jamaica' 'Kenya'
 'Kosovo' 'Lithuania' 'Mali' 'Mexico' 'Portugal' 'Refugee' 'Romania'
 'Senegal' 'Thailand' 'Trinidad and Tobago' 'Tunisia' 'Zambia' 'Zimbabwe'
 'Western Sahara' 'Comoros' 'Hong Kong' 'Jordan' 'Qatar' 'Saudi Arabia'
 'Serbia' 'Niger' 'Germany' 'Kazakhstan' 'Azerbaijan' 'Belgium'
 'Central African Republic' 'Costa Rica' 'Cyprus (Northern part of)'
 'Mauritania' 'Mauritius' 'Nicaragua' 'Peru' 'Sweden' 'Croatia' 'Hungary'
 'Rwanda' 'Singapore' 'Suriname' 'Burkina Faso' 'Equatorial Guinea'
 'Grenada' 'Philippines' 'United Arab Emirates' 'Bhutan' 'Malawi'
 'Ecuador' 'Djibouti' 'Virgin Islands (British)' 'Tajikistan' 'Belarus'
 'Benin' 'Japan' 'Uzbekistan' 'Pitcairn Islands (British)' 'Guatemala'
 'Namibia' 'Togo' 'Venezuela' 'Bulgaria' 'Norway' 'Australia' 'St Lucia'
 'Czechia' 'Cambodia' 'Malaysia' 'Myanmar (Burma)' 'Armenia' 'Slovakia'
 'Not currently recorded' 'St Vincent and the Grenadines' 'Dominica'
 'Greenland' 'Haiti' 'Mozambique' 'Andorra' 'Oman' 'Reunion (French)'
 'Botswana' 'Fiji' 'Honduras' 'Bahrain' 'Cuba' 'Marshall Islands'
 'Ireland' 'Turkmenistan' 'East Timor' 'Finland' 'Laos' 'Bolivia'
 'Iceland' 'Guyana' 'Indonesia' 'Montenegro' 'Bahamas, The' 'Eswatini'
 'Malta' 'Mongolia' 'French Guiana (French)' 'Papua New Guinea' 'Cyprus'
 'Austria' 'Barbados' 'Paraguay' 'El Salvador' 'Saint Barthelemy (French)']
```

In [25]:
```
print(df['Region'].unique())
```

```
['Asia Central' 'Europe Other' 'Africa North' 'Africa Sub-Saharan'
 'America Central and South' 'Other' 'America North' 'Asia East' 'EU 14'
 'Asia South' 'Middle East' 'EU 8' 'Asia South East' 'EU 2' 'EU Other'
 'Oceania' 'Not currently recorded']
```

People are from many countries and regions around the world

In [26]:
```
print(df['Sex'].unique())
```

```
['Female' 'Male' 'Unknown' 'Not currently recorded']
```

In [27]:
```
print(df['Age Group'].unique())
```

```
['17 and under' '18 to 24' '25 to 39' '40 and over' 'Unknown'
 'Not currently recorded']
```

In [28]:
```
df['Number of detections']
```

Out[28]:
```
0         14
1          4
2         14
3         22
4         21
        ...
12857      5
12858     32
12859    278
12860    473
12861     74
Name: Number of detections, Length: 12862, dtype: object
```

In [29]:
```
df['Number of detections'].unique()
```

Out[29]:
```
array(['14', '4', '22', '21', '8', '20', '18', '3', '12', '5', '25', '37',
       '7', '1', '2', '11', '6', '28', '15', '9', '71', '85', '31', '16',
       '10', '47', '24', '17', '19', '30', '13', '50', '55', '38', '23',
       '57', '91', '112', '162', '60', '59', '96', '27', '35', '58', '29',
       '99', '51', '34', '105', '98', '123', '33', '87', '69', '78',
       '122', '79', '36', '42', '43', '119', '41', '32', '165', '125',
       '116', '107', '54', '92', '120', '146', '188', '113', '75', '110',
       '46', '210', '74', '82', '102', '88', '127', '171', '45', '142',
       '267', '249', '72', '157', '26', '97', '137', '182', '128', '214',
       '233', '64', '66', '40', '63', '70', '73', '106', '160', '205',
       '126', '115', '173', '68', '61', '117', '83', '129', '143', '130',
       '84', '39', '304', '44', '48', '161', '238', '224', '53', '202',
       '198', '80', '52', '89', '49', '136', '100', '144', '338', '62',
       '104', '153', '81', '560', '134', '264', '56', '138', '231', '362',
       '108', '151', '86', '65', '174', '77', '167', '168', '281', '67',
       '178', '95', '164', '278', '118', '226', '402', '163', '145',
       '154', '180', '141', '251', '308', '93', '200', '159', '419',
       '852', '475', '1,061', '1,182', '256', '548', '628', '76', '204',
       '172', '302', '306', '177', '90', '179', '221', '243', '492',
       '1,580', '1,493', '187', '149', '193', '363', '1,299', '1,158',
       '430', '375', '329', '356', '364', '121', '309', '227', '103',
       '284', '313', '396', '892', '894', '150', '109', '170', '213',
       '183', '431', '152', '207', '303', '139', '158', '562', '1,206',
       '781', '397', '257', '459', '1,025', '3,082', '2,999', '626',
       '195', '94', '276', '474', '720', '175', '244', '320', '332',
       '245', '450', '374', '232', '169', '114', '833', '1,981', '1,143',
       '370', '434', '135', '191', '255', '340', '441', '779', '312',
       '394', '101', '335', '272', '323', '229', '268', '310', '212',
       '285', '379', '176', '166', '131', '181', '378', '147', '247',
       '524', '1,165', '780', '201', '197', '405', '519', '421', '435',
       '240', '330', '282', '186', '481', '460', '156', '194', '328',
       '341', '321', '208', '499', '349', '185', '222', '429', '345',
```

```
       '388', '155', '287', '307', '343', '184', '266', '933', '760',
       '336', '479', '710', '291', '598', '504', '133', '199', '827',
       '665', '520', '578', '326', '440', '275', '674', '708', '111',
       '473'], dtype=object)
```

We have some values with commas for thousands that we need to strip out

In [30]:
```
# Strip out commas and set to integer
df['Number of detections'] = (df['Number of detections'].str.replace(',', '', regex=Tru
```

Arrivals by Year

In [31]:
```
# Get arrivals by year and look at unique values
arr_year = df[['Year','Quarter','Number of detections']]
```

In [32]:
```
# Check the info
arr_year.columns = ['Year','Quarter','Number']
arr_year.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 12862 entries, 0 to 12861
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Year     12862 non-null  int64 
 1   Quarter  12862 non-null  object
 2   Number   12862 non-null  int32 
dtypes: int32(1), int64(1), object(1)
memory usage: 351.7+ KB
```

In [33]:
```
# Get arrivals by year
sum_arrivals = arr_year.groupby(['Year'],as_index = False).sum()
```

In [34]:
```
# Arrivals by year
sum_arrivals
```

Out[34]:

	Year	Number
0	2018	13377
1	2019	16281
2	2020	17100
3	2021	36813
4	2022	54702
5	2023	36699
6	2024	43630

In [35]:
```
# Total number of arrivals
sum_arrivals['Number'].sum()
```

Out[35]: 218602

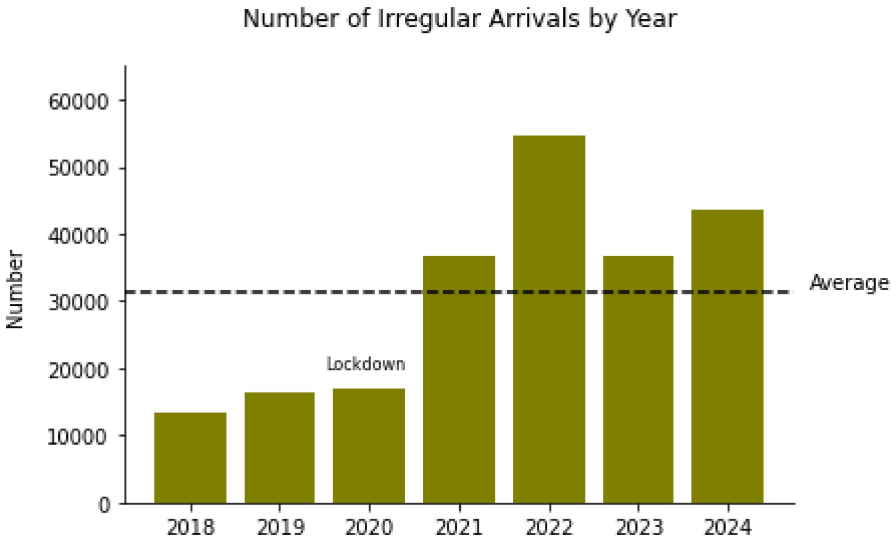
In [36]:
```
# Average arrivals
sum_arrivals['Number'].mean()
```

Out[36]: 31228.85714285714

In [37]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 12862 entries, 0 to 12861
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   Year                  12862 non-null  int64 
 1   Quarter               12862 non-null  object
 2   Method of entry       12862 non-null  object
 3   Nationality           12862 non-null  object
 4   Region                12862 non-null  object
 5   Sex                   12862 non-null  object
 6   Age Group             12862 non-null  object
 7   Number of detections  12862 non-null  int32 
dtypes: int32(1), int64(1), object(6)
memory usage: 854.1+ KB
```

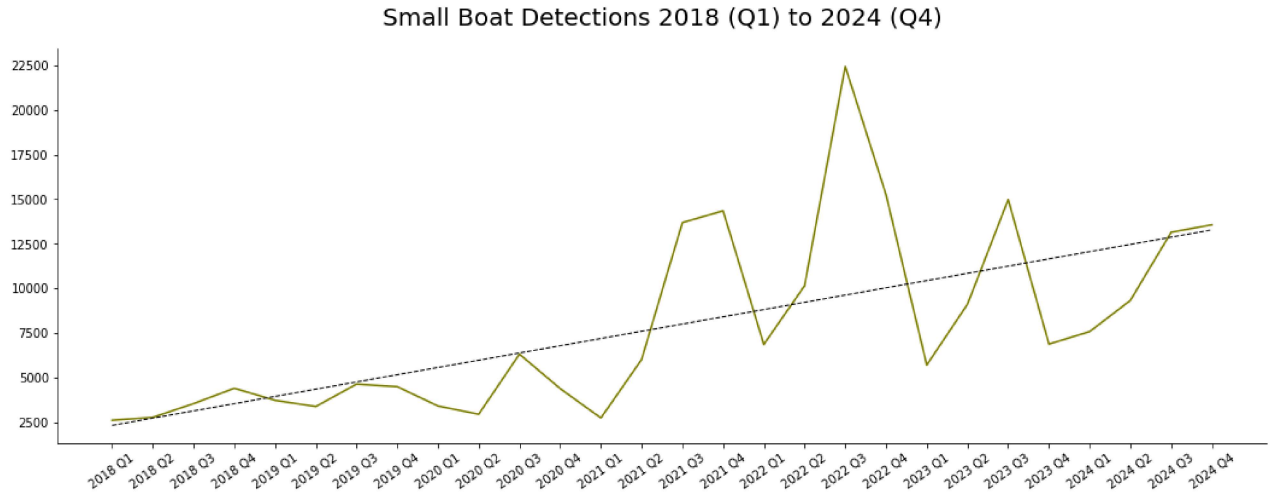
In [38]:
```
# Plot Arrivals by Year
fig, ax = plt.subplots(figsize = (6,4))
plt.bar(x = sum_arrivals.Year, height = sum_arrivals.Number,color = 'olive')
plt.title("Number of Irregular Arrivals by Year",fontsize = 12, pad = 20)
plt.text(2019.5,20000,'Lockdown', fontsize = 8)
plt.text(2024.9,31900,'Average')
plt.ylabel('Number',labelpad = 10)
plt.axhline(y=sum_arrivals['Number'].mean(), color='k', linestyle='--')
plt.ylim(0,65000)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False);
```



- The trend is generally upwards over the period

- The jump up in 2021 and into 2022 follows a period of lockdown for Covid 19 in 2020
- The numbers fall back in 2023 but still higher than prior to covid and above the mean value as the dotted line

In [39]:
```
# Plot by quarter over the period
quarterly = df.groupby('Quarter')['Number of detections'].sum()
ax = quarterly.plot(figsize = (18,6), color = 'olive')

# Set x-ticks and labels using the index
ax.set_xticks(range(len(quarterly)))
ax.set_xticklabels(quarterly.index, rotation=35)

# Trend line
x = np.arange(len(quarterly))  # numerical x-values
y = quarterly.values           # detection counts

# Fit a 1st degree polynomial (linear trend)
z = np.polyfit(x, y, 1)
p = np.poly1d(z)

# Plot trend line
ax.plot(x, p(x), color='black', linestyle='--', linewidth=1, label='Trend')

# Add axis labels and title if desired
ax.set_xlabel('')
ax.set_ylabel('')
ax.set_title('Small Boat Detections 2018 (Q1) to 2024 (Q4)',fontsize = 20, pad = 20)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
```



Arrivals by Method of Entry

In [40]:
```
# Get arrivals by method
sum_method = df.groupby(['Method of entry'],as_index = False)['Number of detections'].s
sum_method.columns = ['Method','Number']
sum_method['%'] = (sum_method['Number']/sum_method['Number'].sum())*100
sum_method
```

Out[40]:

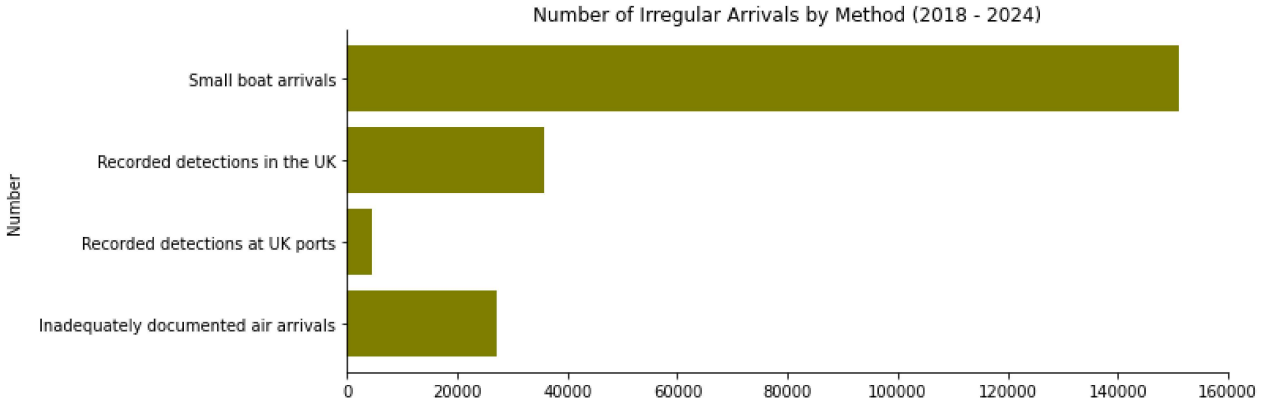
	Method	Number	%
0	Inadequately documented air arrivals	27229	12.455970
1	Recorded detections at UK ports	4483	2.050759
2	Recorded detections in the UK	35729	16.344315
3	Small boat arrivals	151161	69.148956

In [282…

```
# Plot Arrivals by Year
fig, ax = plt.subplots(figsize = (10,4))
bars = ax.barh(sum_method.Method, sum_method.Number,color = 'olive')
plt.title("Number of Irregular Arrivals by Method (2018 - 2024)",fontsize = 12)
plt.ylabel('Number',labelpad = 10)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.xlim(0,160000)
#for bar in bars:
   # width = bar.get_width()
    #ax.text(width + 1000,
           # bar.get_y() + bar.get_height() / 2,
           # f'{int(width):,}',
           # va='center');
```

Out[282…

```
(0.0, 160000.0)
```



Most irregular arrivals are by small boats

Arrivals by Method by Year

In [218…

```
# Get arrivals by method and number of detections and convert to pivot table
arr_meth_year = df.groupby(['Year','Method of entry'],as_index = False)['Number of dete
```

In [220…

```
arr_meth_year
```

Out[220…

	Year	Method of entry	Number of detections
0	2018	Inadequately documented air arrivals	4769
1	2018	Recorded detections at UK ports	1052

	Year	Method of entry	Number of detections
2	2018	Recorded detections in the UK	7257
3	2018	Small boat arrivals	299
4	2019	Inadequately documented air arrivals	5237
5	2019	Recorded detections at UK ports	962
6	2019	Recorded detections in the UK	8239
7	2019	Small boat arrivals	1843
8	2020	Inadequately documented air arrivals	2328
9	2020	Recorded detections at UK ports	841
10	2020	Recorded detections in the UK	5465
11	2020	Small boat arrivals	8466
12	2021	Inadequately documented air arrivals	2561
13	2021	Recorded detections at UK ports	665
14	2021	Recorded detections in the UK	5061
15	2021	Small boat arrivals	28526
16	2022	Inadequately documented air arrivals	5130
17	2022	Recorded detections at UK ports	310
18	2022	Recorded detections in the UK	3488
19	2022	Small boat arrivals	45774
20	2023	Inadequately documented air arrivals	3854
21	2023	Recorded detections at UK ports	327
22	2023	Recorded detections in the UK	3081
23	2023	Small boat arrivals	29437
24	2024	Inadequately documented air arrivals	3350
25	2024	Recorded detections at UK ports	326
26	2024	Recorded detections in the UK	3138
27	2024	Small boat arrivals	36816

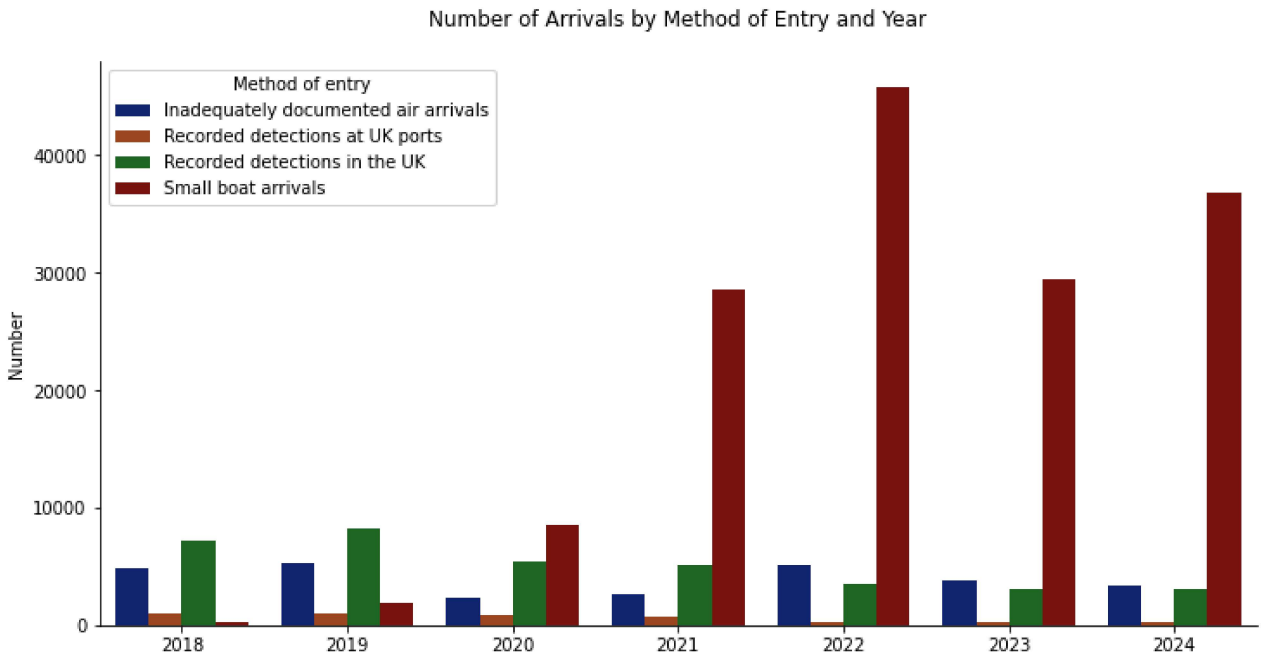
In [563…

```
# Plot the arrivals by method and year
fig, ax = plt.subplots(figsize = (12,6))
g = sns.barplot(data = arr_meth_year ,
                x='Year',
                y='Number of detections',
                hue='Method of entry',
                palette = 'dark')
plt.title("Number of Arrivals by Method of Entry and Year", pad = 20)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.ylabel("Number")
```

```
plt.xlabel("")
;
```

Out[563…

```
''
```



We can see that detections at ports and in the UK have trended downwards, air arrivals have fluctuated over the period but the largest category of small boat arrivals is the largest element and peaked in 2022

Analysis by Quarter

In [223…

```
df['Quarter'].head()
```

Out[223…

```
0    2018 Q1
1    2018 Q1
2    2018 Q1
3    2018 Q1
4    2018 Q1
Name: Quarter, dtype: object
```

We would like to split off the quarter from the year and then just groupby quarter

In [224…

```
# Select the columns we want, strip off the Q1 etc and groupby this new column
quarter = df[['Quarter','Number of detections']]
quarter['Q'] = quarter['Quarter'].str[-2:]
quarter_grouped = quarter.groupby('Q',as_index =False)['Number of detections'].sum()
quarter_grouped
```

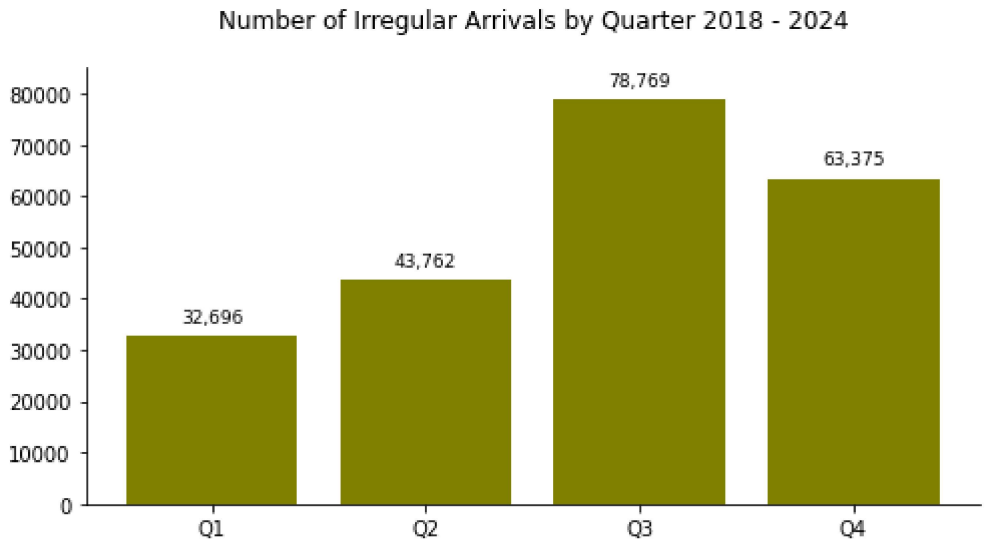
Out[224…

	Q	Number of detections
0	Q1	32696
1	Q2	43762
2	Q3	78769

	Q	Number of detections
3	Q4	63375

In [266…

```
# Plot Arrivals by Quarter
fig, ax = plt.subplots(figsize = (8,4))
bars = plt.bar(x = quarter_grouped.Q, height = quarter_grouped['Number of detections'],
plt.title("Number of Irregular Arrivals by Quarter 2018 - 2024",fontsize = 12, pad = 20
plt.ylabel('Number',labelpad = 10)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.ylabel("")
plt.ylim(0,85000)

#for bar in bars:
   # height = bar.get_height()
    #ax.text(
       # bar.get_x() + bar.get_width() / 2,  # center of the bar
        #height + 2000,                      # slightly above the bar
       # f'{int(height):,}',                 # formatted label
        #ha='center',
        #va='bottom',
        #fontsize=9
   # );
```



Quarter 3 has almost 2.5 times the amount in Q1. This is July to September, the summer period and the weather is better for the small boats to make the crossing

What age group are they?

In [267…

```
df['Age Group'].unique()
```

Out[267…

```
array(['17 and under', '18 to 24', '25 to 39', '40 and over',
       'Not available'], dtype=object)
```

In [271…

```
# Replace these values
df['Age Group'].replace({'Not currently recorded':'Not available','Unknown':'Not availa
```

The age categories are as we might expect but there are also some that are not recorded and some that are unknown. It is not clear what the difference is between these but we will combine these into one category of 'not available'.

In [272…

```
# Create dataframe of number of detections by age group
age_sum = df.groupby(['Age Group'],as_index = False)['Number of detections'].sum()
age_sum
```

Out[272…

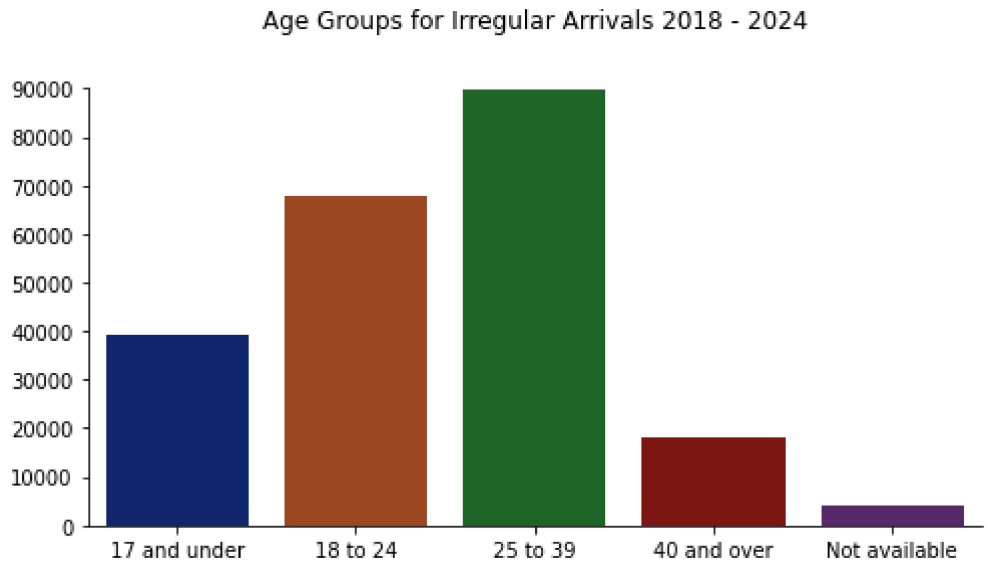
	Age Group	Number of detections
0	17 and under	39255
1	18 to 24	67607
2	25 to 39	89537
3	40 and over	18063
4	Not available	4140

In [283…

```
# Plot the age data
fig, ax = plt.subplots(figsize = (8,4))
ax = sns.barplot(data = age_sum,
                 x = 'Age Group',
                 y = 'Number of detections',
                # errorbar = None,
                 palette = 'dark')
plt.title("Age Groups for Irregular Arrivals 2018 - 2024", pad = 30)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.ylabel("Number", labelpad = 10)
plt.ylim(0,90000)
plt.xlabel("")
plt.ylabel("")
#for bar in ax.patches:
  #  height = bar.get_height()
    #ax.text(
      #  bar.get_x() + bar.get_width() / 2,
      # height + 1000,
       # f'{int(height):,}',
       # ha='center',
       # va='bottom',
       # fontsize=9
   # )
```

Out[283…

```
Text(0, 0.5, '')
```



As commonly noted in the press, the main age group is 25 to 39. We can investigate if this has changed over the six year period by adding in year to the plot.

In [234…

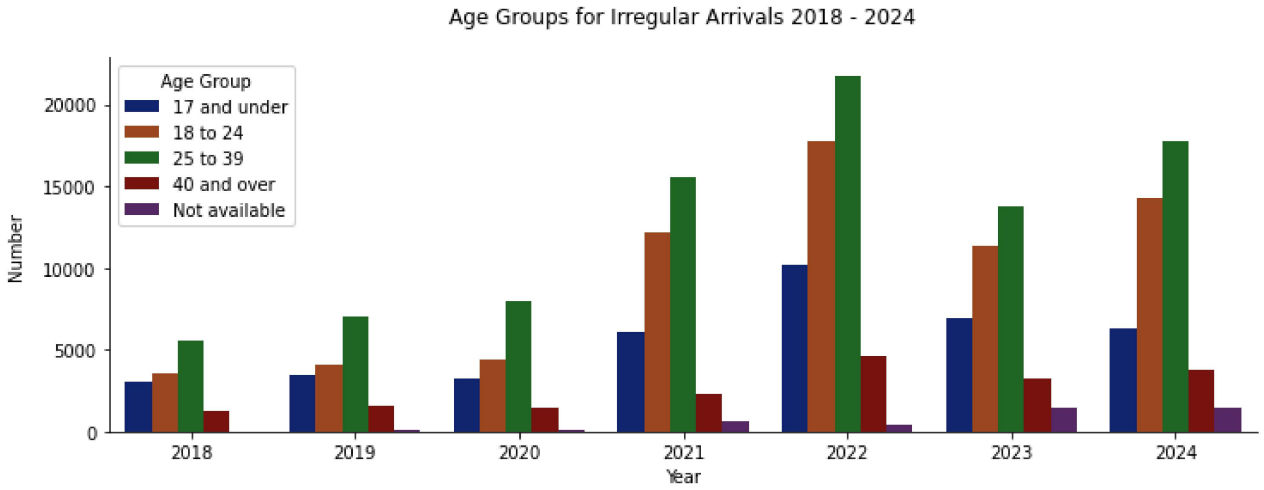
```
# Create dataframe number of detections by year and age groupo
age_year = df.groupby(['Year','Age Group'], as_index = False)['Number of detections'].s
age_year.head()
```

Out[234…

	Year	Age Group	Number of detections
0	2018	17 and under	2991
1	2018	18 to 24	3598
2	2018	25 to 39	5592
3	2018	40 and over	1193
4	2018	Not available	3

In [564…

```
# Plot the arrivals by age group per year
fig, ax = plt.subplots(figsize = (12,4))
ax = sns.barplot(data = age_year,
                 x ='Year',
                 y = 'Number of detections',
                 hue = 'Age Group',
                 #errorbar = None,
                 palette = 'dark')
plt.title("Age Groups for Irregular Arrivals 2018 - 2024", pad = 20)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.ylabel("Number", labelpad = 10);
```



We can see that the largest age group remains 25 to 39 over the period

In [236…

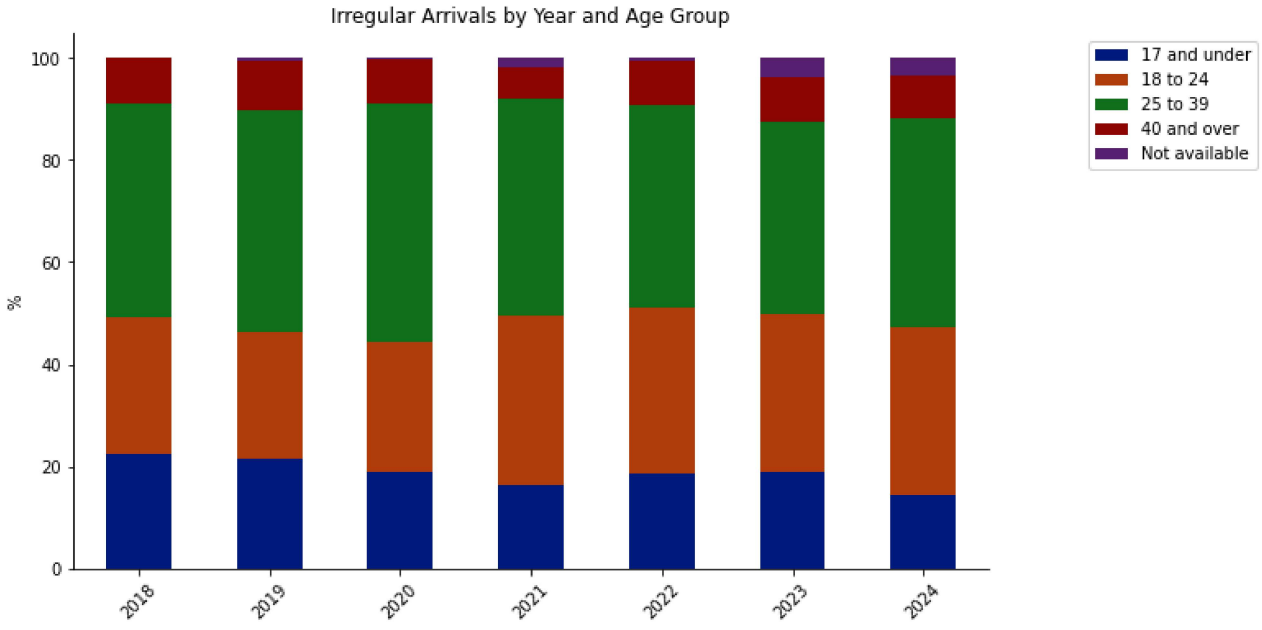
```
# Create a crosstab table for the values we want
age_cross = pd.crosstab(index = age_year['Year'], columns = age_year['Age Group'], valu
age_cross = (age_cross*100).reset_index()
age_cross
```

Out[236…

Age Group	Year	17 and under	18 to 24	25 to 39	40 and over	Not available
0	2018	22.359273	26.896913	41.803095	8.918293	0.022427
1	2019	21.436030	24.967754	43.387998	9.532584	0.675634
2	2020	18.836257	25.438596	46.836257	8.491228	0.397661
3	2021	16.483308	33.097004	42.302991	6.367316	1.749382
4	2022	18.662937	32.402837	39.808417	8.403715	0.722094
5	2023	18.989618	30.981771	37.415188	8.703234	3.910188
6	2024	14.455650	32.809993	40.779280	8.551455	3.403621

In [565…

```
# Plot the crosstablulated data
sns.set_palette("dark")
ax = age_cross.plot(
    x = 'Year',
    kind = 'bar',
    stacked = True,
    title = 'Irregular Arrivals by Year and Age Group',
    mark_right = True,
    figsize = (10,6)
    )

plt.xticks(rotation=45)
plt.xlabel("")
plt.ylabel("%")
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.legend(bbox_to_anchor=(1.1, 1));
```



- We can see that the category for age unavailable has increased as a proportion over the period
- The category for 25 to 39 has decreased as a proportion of the total compared to the 18 to 24 age group.

What sex are the arrivals?

In [243…

```
sex = df.groupby(['Sex'],as_index = False)['Number of detections'].sum()
sex['%'] = round(sex['Number of detections']/(sex['Number of detections'].sum())*100,2)
sex
```

Out[243…

	Sex	Number of detections	%
0	Female	31753	14.53
1	Male	182932	83.68
2	Not currently recorded	3275	1.50
3	Unknown	642	0.29

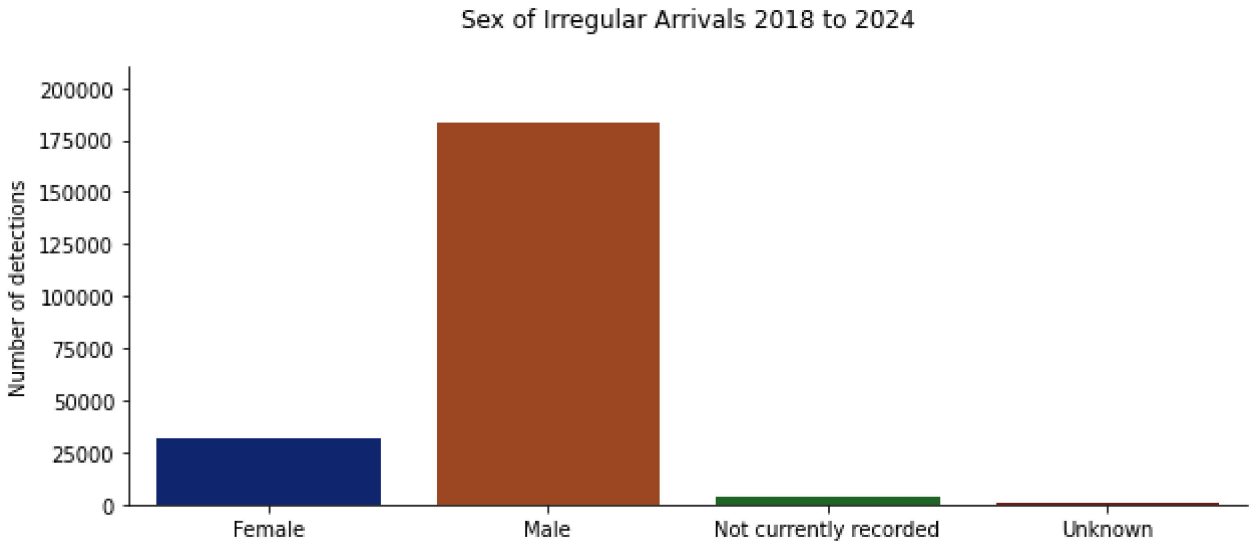
Irregular arrivals are predominantly male

In [566…

```
fig, ax = plt.subplots(figsize = (10,4))
sns.set_palette("dark")
ax = sns.barplot(data = sex, x = 'Sex', y = 'Number of detections')
plt.title('Sex of Irregular Arrivals 2018 to 2024',pad = 20)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.xlabel("")
plt.ylim(0,210000)
#for index, row in sex.iterrows():
    #plt.text(index, row['Number of detections'] + 3000, f"{row['%']:.1f}%",
             #color='black', ha="center")
;
```

Out[566…

''



In [91]:

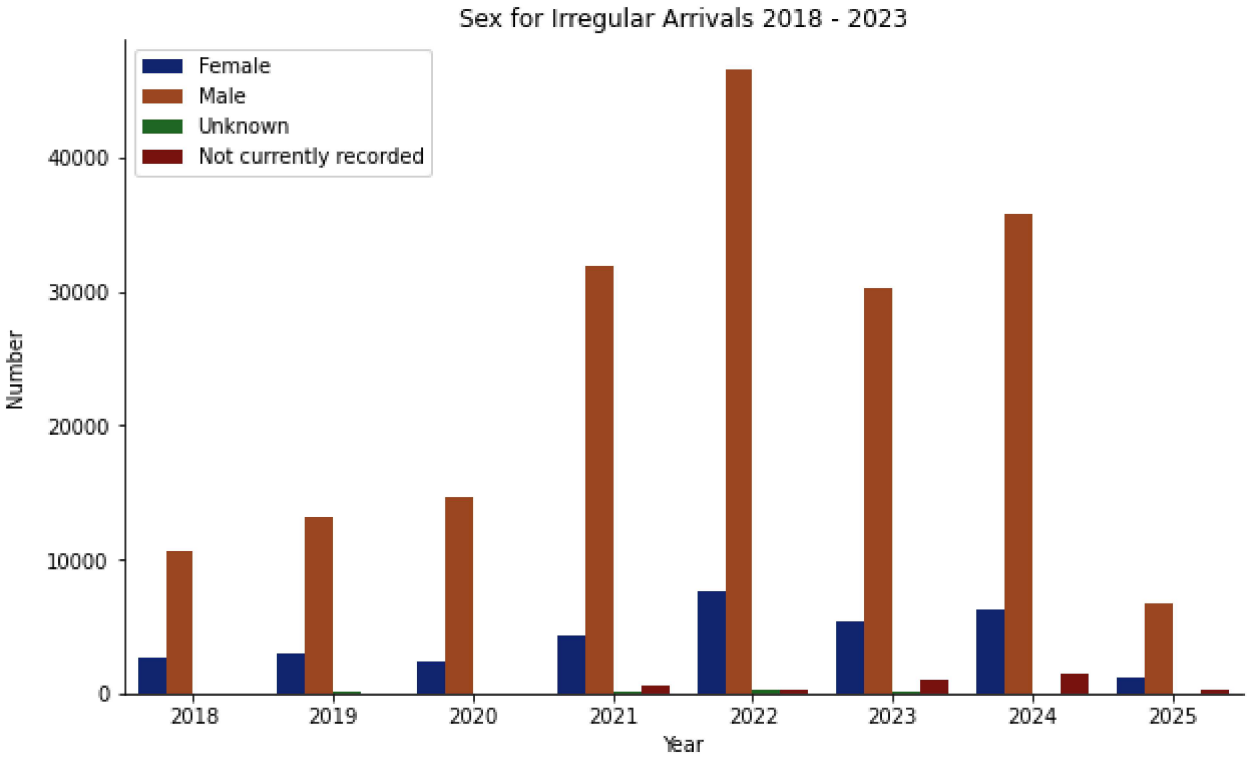
```
# Create dataframe number of detections by year and age groupo
sex_year = df.groupby(['Year','Sex'], as_index = False)['Number of detections'].sum()
sex_year.head()
```

Out[91]:

	Year	Sex	Number of detections
0	2018	Female	2704
1	2018	Male	10634
2	2018	Unknown	39
3	2019	Female	3023
4	2019	Male	13196

In [285…

```
# Plot the arrivals by sex per year
plt.figure(figsize=(10, 6))
ax = sns.barplot(data = sex_year,
                 x ='Year',
                 y = 'Number of detections',
                 hue = 'Sex',
                 #errorbar = None,
                 palette = 'dark')
plt.title("Sex for Irregular Arrivals 2018 - 2023")
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.ylabel("Number", labelpad = 10)
ax.legend(loc = 'upper left');
```



In [286…

```
# Create a crosstab table for the values we want
sex_cross = pd.crosstab(index = sex_year['Year'], columns = sex_year['Sex'], values = s
sex_cross = (sex_cross*100).reset_index()
sex_cross
```

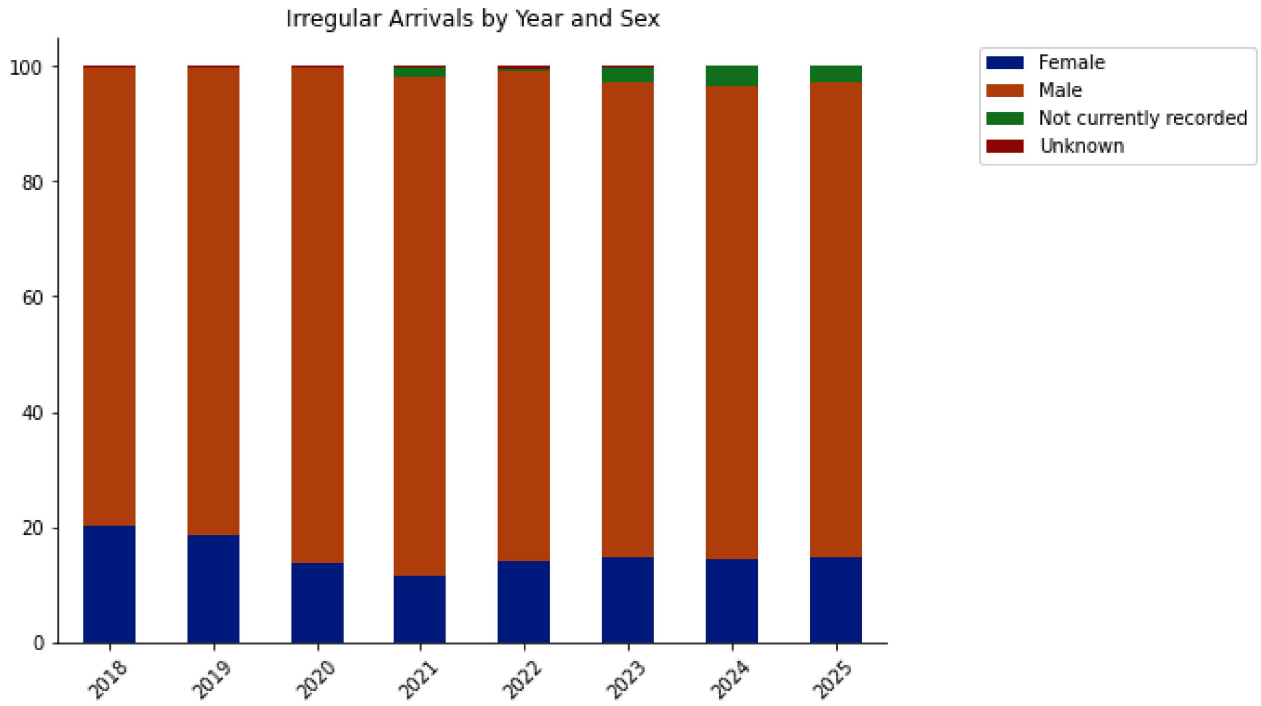
Out[286…

Sex	Year	Female	Male	Not currently recorded	Unknown
0	2018	20.213800	79.494655	0.000000	0.291545
1	2019	18.567656	81.051532	0.000000	0.380812
2	2020	13.789474	86.017544	0.000000	0.192982
3	2021	11.593730	86.472170	1.580963	0.353136
4	2022	14.025081	85.091953	0.365617	0.517349
5	2023	14.809668	82.195700	2.822965	0.171667
6	2024	14.423562	82.163649	3.339445	0.073344
7	2025	14.690594	82.537129	2.747525	0.024752

In [287…

```
# Plot the crosstablulated data
sns.set_palette("dark")

ax = sex_cross.plot(
    x = 'Year',
    kind = 'bar',
    stacked = True,
    title = 'Irregular Arrivals by Year and Sex',
    mark_right = True,
     figsize=(8, 6)
    )
```

```
plt.xticks(rotation=45)
plt.xlabel("")
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.legend(bbox_to_anchor=(1.1, 1));
```



We can see that the proportion of females being recorded initially fell from 2018 but increased again from 2022

Where do they come from?

In [288…

```
df['Region'].unique()
```

Out[288…

```
array(['Asia Central', 'Europe Other', 'Africa North',
       'Africa Sub-Saharan', 'America Central and South', 'Other',
       'America North', 'Asia East', 'EU 14', 'Asia South', 'Middle East',
       'EU 8', 'Asia South East', 'EU 2', 'EU Other', 'Oceania',
       'Not currently recorded'], dtype=object)
```

We have a number of regions where the irregular arrivals come from. We can look at the countries contained within these to get more of an idea

In [289…

```
# Function to display each region
def nation(reg):
    result = df[df['Region']==reg]['Nationality'].value_counts().to_frame()
    print('Region: ',i)
    display(result)
```

In [290…

```
# run the function for each region in the unique list
for i in list(df['Region'].unique()):
    nation(i)
```

Region: Asia Central

Nationality	
Afghanistan	677
Tajikistan	27
Kazakhstan	17
Kyrgyzstan	13
Uzbekistan	6
Turkmenistan	3

Region: Europe Other

Nationality	
Albania	675
Turkey	394
Georgia	266
Ukraine	113
Russia	112
Azerbaijan	51
Kosovo	31
Moldova	16
Cyprus (Northern part of)	13
Belarus	11
North Macedonia	8
Serbia	8
Norway	5
Bosnia and Herzegovina	4
Montenegro	3
Armenia	3
Andorra	2
Iceland	1

Region: Africa North

Nationality	
Sudan	545
Libya	258
Algeria	238
Morocco	234
Egypt	220

	Nationality
Tunisia	91
Mauritania	16
Western Sahara	11

Region: Africa Sub-Saharan

	Nationality
Eritrea	620
Ethiopia	376
Somalia	324
Chad	197
Nigeria	185
Guinea	124
Gambia, The	105
South Sudan	103
Ghana	98
Congo (Democratic Republic)	74
Tanzania	70
Cameroon	64
Senegal	59
Sierra Leone	58
Mali	48
Ivory Coast	42
South Africa	38
Angola	34
Zimbabwe	34
Niger	30
Congo	25
Kenya	17
Burkina Faso	16
Central African Republic	15
Guinea-Bissau	15
Liberia	15
Benin	11
Burundi	10
Namibia	10

	Nationality
Uganda	9
Togo	7
Mauritius	7
Botswana	6
Djibouti	5
Equatorial Guinea	4
Rwanda	3
Mozambique	2
Malawi	2
Zambia	2
Eswatini	1
Reunion (French)	1
Comoros	1

Region: America Central and South

	Nationality
Brazil	77
Mexico	14
Colombia	13
Dominica	12
Dominican Republic	10
Peru	10
Jamaica	7
Argentina	6
Bolivia	6
Venezuela	5
Cuba	4
Chile	4
Costa Rica	4
Ecuador	3
Trinidad and Tobago	3
French Guiana (French)	2
Nicaragua	2
Guatemala	2
Grenada	2

Nationality	
Guyana	1
El Salvador	1
Bahamas, The	1
St Vincent and the Grenadines	1
Honduras	1
Virgin Islands (British)	1
Haiti	1
Paraguay	1
Barbados	1
Suriname	1
St Lucia	1
Saint Barthelemy (French)	1

Region: Other

Nationality	
Stateless	381
Other and unknown	221
Refugee	105
British overseas citizens	20
Pitcairn Islands (British)	1

Region: America North

Nationality	
United States	29
Canada	3
Greenland	2

Region: Asia East

Nationality	
China	157
Hong Kong	5
Taiwan	4
Japan	3
South Korea	2
Mongolia	1

Region: EU 14

	Nationality
Italy	39
Spain	27
France	26
Portugal	22
Germany	15
Denmark	10
Greece	9
Netherlands	7
Sweden	5
Belgium	5
Ireland	4
Finland	2
Austria	2

Region: Asia South

	Nationality
India	353
Pakistan	267
Sri Lanka	256
Bangladesh	103
Nepal	32
Bhutan	1

Region: Middle East

	Nationality
Iran	777
Iraq	720
Syria	582
Kuwait	365
Yemen	230
Occupied Palestinian Territories	215
Lebanon	78
Jordan	42
Saudi Arabia	20
Israel	13
United Arab Emirates	9

	Nationality
Qatar	3
Bahrain	1
Oman	1

Region: EU 8

	Nationality
Poland	23
Lithuania	11
Hungary	10
Czechia	6
Slovakia	6
Latvia	6
Slovenia	2

Region: Asia South East

	Nationality
Vietnam	365
Philippines	11
Myanmar (Burma)	9
Thailand	6
Indonesia	5
Malaysia	2
Singapore	2
Laos	1
Cambodia	1
East Timor	1

Region: EU 2

	Nationality
Romania	48
Bulgaria	13

Region: EU Other

	Nationality
Cyprus	1
Malta	1
Croatia	1

Region: Oceania

	Nationality
Fiji	1
Australia	1
Marshall Islands	1
Papua New Guinea	1

```
Region:  Not currently recorded
```

	Nationality
Not currently recorded	15

Category Analysis

- EU2 - Romania and Bulgaria
- EU14 - Italy, Spain, France, Portugal, Germany, Denmark, Greece, Netherlands, Belgium, Sweden, Ireland, Finland, Austria
- EU8 - Poland, Lithuania, Hungary, Latvia, Czechia, Slovakia, Slovenia
- Europe Other - Albania, Turkey, Georgia, Ukraine, Russia, Azerbaijan, Kosovo, Cyprus(North), Moldova, Belarus, North Macedonia, Norway, Servia, Bosnia and Herzegovina, Andorra, Montenegro, Armenia, Iceland
- Other - Stateless, Other and Unknown, Refugee, British Overseas Citizens, Pitcairn Islands, South Georgia and South Sandwich Islands

In [291…

```
# Groupby region and sort in descending order
df.groupby('Region',as_index = False)['Number of detections'].sum().sort_values(by = 'N
```

Out[291…

	Region	Number of detections
13	Middle East	87661
12	Europe Other	32816
1	Africa Sub-Saharan	27152
4	Asia Central	26454
0	Africa North	20568
7	Asia South East	7945
6	Asia South	7653
16	Other	3365
14	Not currently recorded	3275
5	Asia East	1086
2	America Central and South	248
8	EU 14	190
9	EU 2	77

	Region	Number of detections
10	EU 8	67
3	America North	38
15	Oceania	4
11	EU Other	3

- We can see that most come from the Middle East
- We can probably clean up the data a little so we can visualise it better
- EU14, EU2, EU8 and EU Other into one category

In [292…
```
df_copy = df.copy()

df_copy['Region'].replace({'EU 14':'EU','EU 2':'EU','EU 8':'EU','EU Other':'EU'}, inpla
```

In [293…
```
df_copy.groupby('Region',as_index = False)['Number of detections'].sum().sort_values(by
```

Out[293…

	Region	Number of detections
10	Middle East	87661
9	Europe Other	32816
1	Africa Sub-Saharan	27152
4	Asia Central	26454
0	Africa North	20568
7	Asia South East	7945
6	Asia South	7653
13	Other	3365
11	Not currently recorded	3275
5	Asia East	1086
8	EU	337
2	America Central and South	248
3	America North	38
12	Oceania	4

In [294…
```
# Drop those not currently recorded
df_copy = df_copy[df_copy['Region']!='Not currently recorded']
```

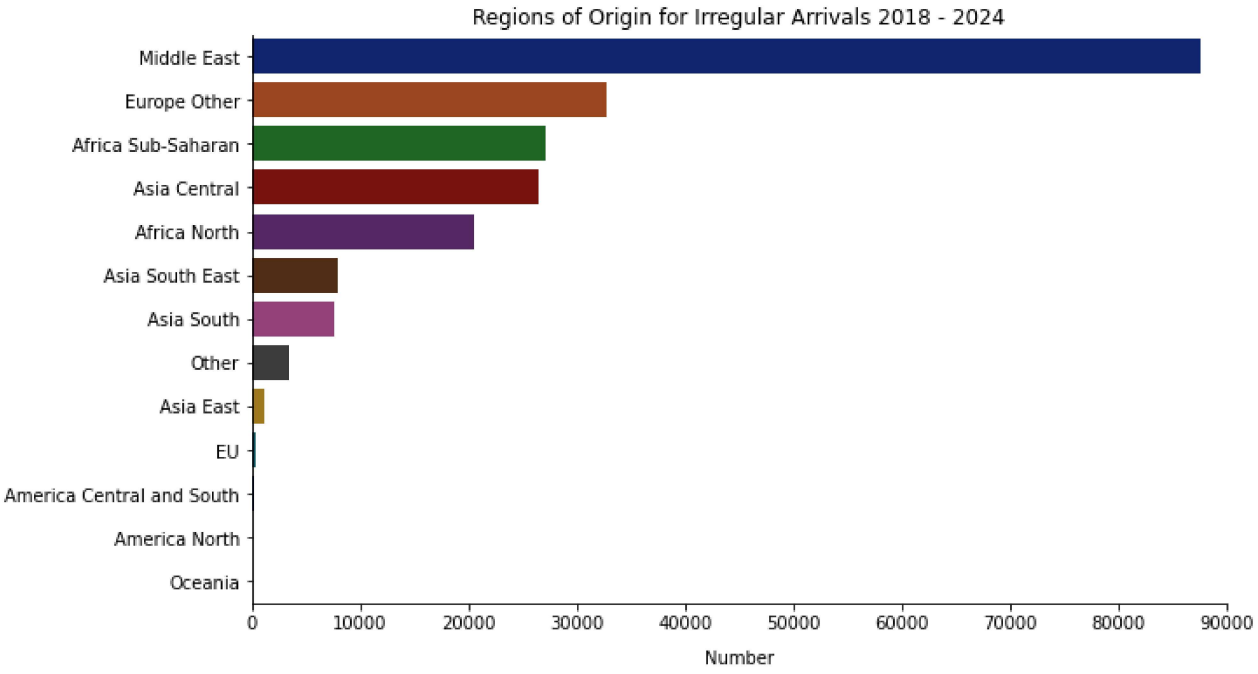
In [295…
```
df_region = df_copy.groupby(['Region'], as_index = False)['Number of detections'].sum()
df_region.head()
```

Out[295…

	Region	Number of detections
10	Middle East	87661
9	Europe Other	32816
1	Africa Sub-Saharan	27152
4	Asia Central	26454
0	Africa North	20568

In [299…

```
# Plot the age data
plt.figure(figsize=(10, 6))
ax = sns.barplot(data = df_region,
                 y = 'Region',
                 x = 'Number of detections',
                 #errorbar = None,
                 palette = 'dark',
                )
plt.title("Regions of Origin for Irregular Arrivals 2018 - 2024")
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.xlim(0,90000)
plt.ylabel("")
plt.xlabel("Number", labelpad = 10);
```



The Middle East is by far and away the source of the largest number of irregular arrivals. We can see if this has changed in pattern over the years or if this has been the main source for the whole six year period.

In [300…

```
# Groupby region and year
region_year = df_copy.groupby(['Year','Region'], as_index = False)['Number of detection
region_year
```

Out[300…

	Year	Region	Number of detections
0	2018	Africa North	1484
1	2018	Africa Sub-Saharan	2290
2	2018	America Central and South	31
3	2018	America North	15
4	2018	Asia Central	741
…	…	…	…
81	2024	Asia South East	3803
82	2024	EU	57
83	2024	Europe Other	3711
84	2024	Middle East	15318
85	2024	Other	607

86 rows × 3 columns

In [306…

```
region_year.sort_values(by = ["Year","Number of detections"],ascending = False)
```

Out[306…

	Year	Region	Number of detections
84	2024	Middle East	15318
78	2024	Asia Central	6360
75	2024	Africa Sub-Saharan	6142
74	2024	Africa North	4477
81	2024	Asia South East	3803
…	…	…	…
11	2018	Other	205
7	2018	Asia South East	196
8	2018	EU	45
2	2018	America Central and South	31
3	2018	America North	15

86 rows × 3 columns

In [301…

```
# Create a crosstab table for the values we want
region_cross = pd.crosstab(index = region_year['Year'], columns = region_year['Region']
region_cross = region_cross.reset_index()
region_cross
```

Out[301…

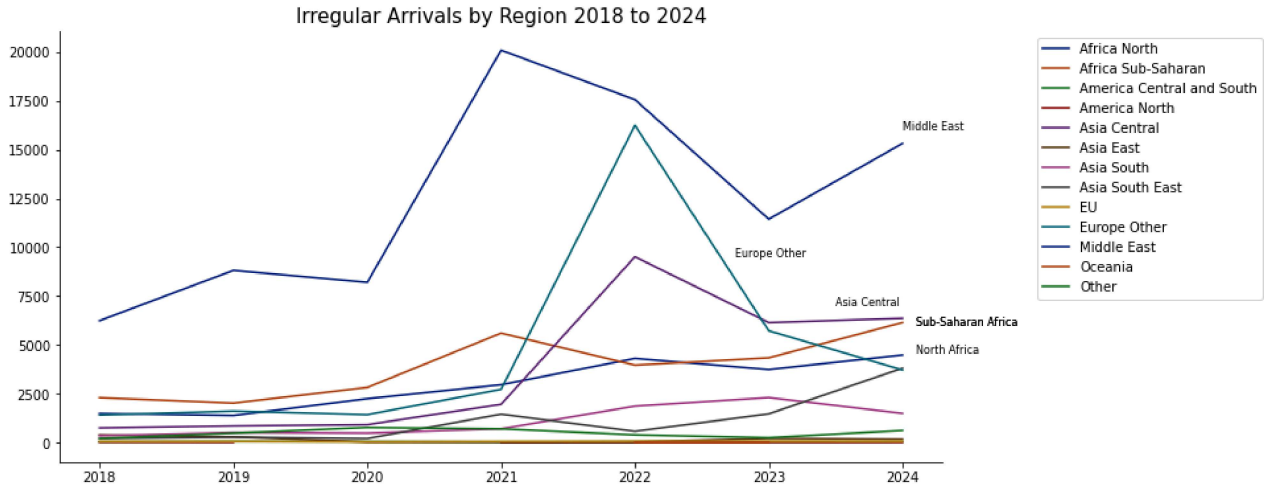
Region	Year	Africa North	Africa Sub-Saharan	America Central and South	America North	Asia Central	Asia East	Asia South	Asia South East	EU	Europe Other	Middle East
0	2018	1484.0	2290.0	31.0	15.0	741.0	388.0	342.0	196.0	45.0	1402.0	6238.0
1	2019	1376.0	2014.0	45.0	5.0	845.0	283.0	503.0	274.0	48.0	1603.0	8813.0
2	2020	2240.0	2814.0	43.0	NaN	910.0	14.0	472.0	199.0	24.0	1421.0	8201.0
3	2021	2953.0	5599.0	23.0	4.0	1950.0	12.0	696.0	1445.0	47.0	2717.0	20093.0
4	2022	4303.0	3960.0	34.0	5.0	9508.0	20.0	1858.0	566.0	57.0	16243.0	17561.0
5	2023	3735.0	4333.0	33.0	4.0	6140.0	201.0	2296.0	1462.0	59.0	5719.0	11437.0
6	2024	4477.0	6142.0	39.0	5.0	6360.0	168.0	1486.0	3803.0	57.0	3711.0	15318.0

In [567…

```
# Plot a line chart
sns.set_palette("dark")
fig, ax = plt.subplots(figsize = (12,6))
region_cross.set_index('Year').plot(ax = ax)
ax.legend(bbox_to_anchor=(1.1, 1))
plt.title('Irregular Arrivals by Region 2018 to 2024', fontsize = 15)
plt.text(2024,16000,'Middle East', fontsize = 8)
plt.text(2022.75,9500,'Europe Other', fontsize = 8)
plt.text(2023.5,7000,'Asia Central', fontsize = 8)
plt.text(2024.1,6000,'Sub-Saharan Africa', fontsize = 8)
plt.text(2024.1,6000,'Sub-Saharan Africa', fontsize = 8)
plt.text(2024.1,4600,'North Africa', fontsize = 8)
plt.xlabel("")
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
;
```

Out[567…

''



- The chart clearly shows the peaks in 2022 amongst the biggest source regions
- The Middle East peaks in 2021

- Europe Other region includes Ukraine, where we might expect larger numbers of people to originate from due to the war but we need to investigate this a bit further
- Asia Central peaks in 2022
- Since then the numbers have been falling but remain higher than the period prior to the Covid lockdown
- We can look at the proportions of the top five regions over the period

In [320…
```
df_copy['Region'].unique()
```

Out[320…
```
array(['Asia Central', 'Europe Other', 'Africa North',
       'Africa Sub-Saharan', 'America Central and South', 'Other',
       'America North', 'Asia East', 'EU', 'Asia South', 'Middle East',
       'Asia South East', 'Oceania'], dtype=object)
```

In [321…
```
# Filter for top 5 regions
regions = ['Middle East','Europe Other', 'Asia Central', 'Africa Sub-Saharan', 'Africa
top5 = df_copy[df_copy['Region'].isin(regions)]

# Groupby year
top5_year = top5.groupby(['Year','Region'], as_index = False)['Number of detections'].s

# Create a crosstab table for the values we want
top5_cross = pd.crosstab(index = top5_year['Year'],
                         columns = top5_year['Region'],
                         values = top5_year['Number of detections'],
                         aggfunc = 'sum',
                         )
top5_cross
```

Out[321…

Region	Africa North	Africa Sub-Saharan	Asia Central	Europe Other	Middle East
Year					
2018	1484	2290	741	1402	6238
2019	1376	2014	845	1603	8813
2020	2240	2814	910	1421	8201
2021	2953	5599	1950	2717	20093
2022	4303	3960	9508	16243	17561
2023	3735	4333	6140	5719	11437
2024	4477	6142	6360	3711	15318

Analysis of some of the peaks on the chart

In [610…
```
# 2022 analysis of Europe Other
df[(df['Region']=='Europe Other') & (df['Year']==2022)].groupby('Nationality')['Number
```

Out[610…
```
Nationality
Albania                    13602
Azerbaijan                    18
Belarus                        1
```

```
Bosnia and Herzegovina           1
Cyprus (Northern part of)        7
Georgia                       1214
Iceland                          1
Kosovo                          19
Moldova                          4
Montenegro                       2
North Macedonia                  4
Russia                          16
Turkey                        1341
Ukraine                         13
Name: Number of detections, dtype: int32
```

In [614…

```
# 2021 analysis of the Middle East
df[(df['Region']=='Middle East') & (df['Year']==2021)].groupby('Nationality')['Number o
```

Out[614…

```
Nationality
Iran                                9455
Iraq                                7029
Jordan                                 7
Kuwait                               573
Lebanon                               15
Occupied Palestinian Territories     124
Oman                                   1
Qatar                                  1
Saudi Arabia                           2
Syria                               2729
United Arab Emirates                   2
Yemen                                155
Name: Number of detections, dtype: int32
```

In [615…

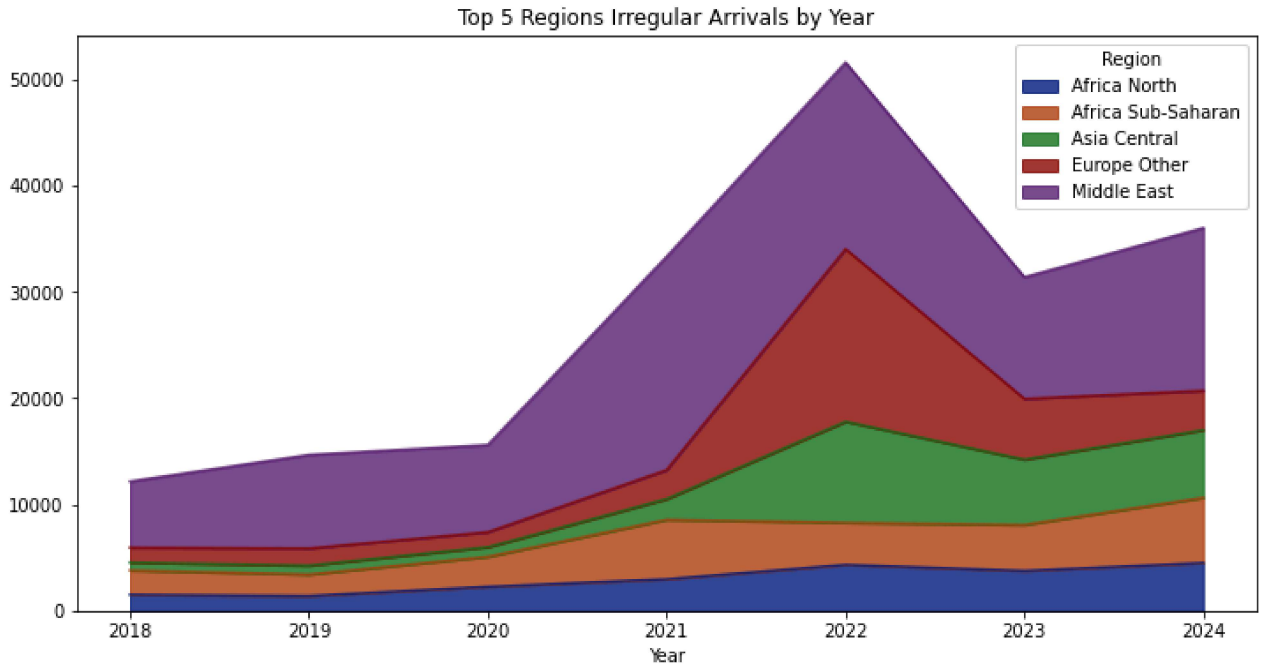
```
# 2022 analysis of Asia Central
df[(df['Region']=='Asia Central') & (df['Year']==2022)].groupby('Nationality')['Number
```

Out[615…

```
Nationality
Afghanistan     9495
Kazakhstan         4
Tajikistan         8
Turkmenistan       1
Name: Number of detections, dtype: int32
```

In [322…

```
# Plot an area chart
sns.set_palette("dark")
fig, ax = plt.subplots(figsize = (12,6))
top5_cross.plot.area(ax = ax, alpha = 0.8)
plt.title("Top 5 Regions Irregular Arrivals by Year");
```



In [323…

```
# Create a percent crosstab table for the values we want
top5_cross_percent = pd.crosstab(index = top5_year['Year'],
                         columns = top5_year['Region'],
                         values = top5_year['Number of detections'],
                         normalize = 'index',
                         aggfunc = 'sum'
                         )
top5_cross_percent = (top5_cross_percent*100).reset_index()
top5_cross_percent
```

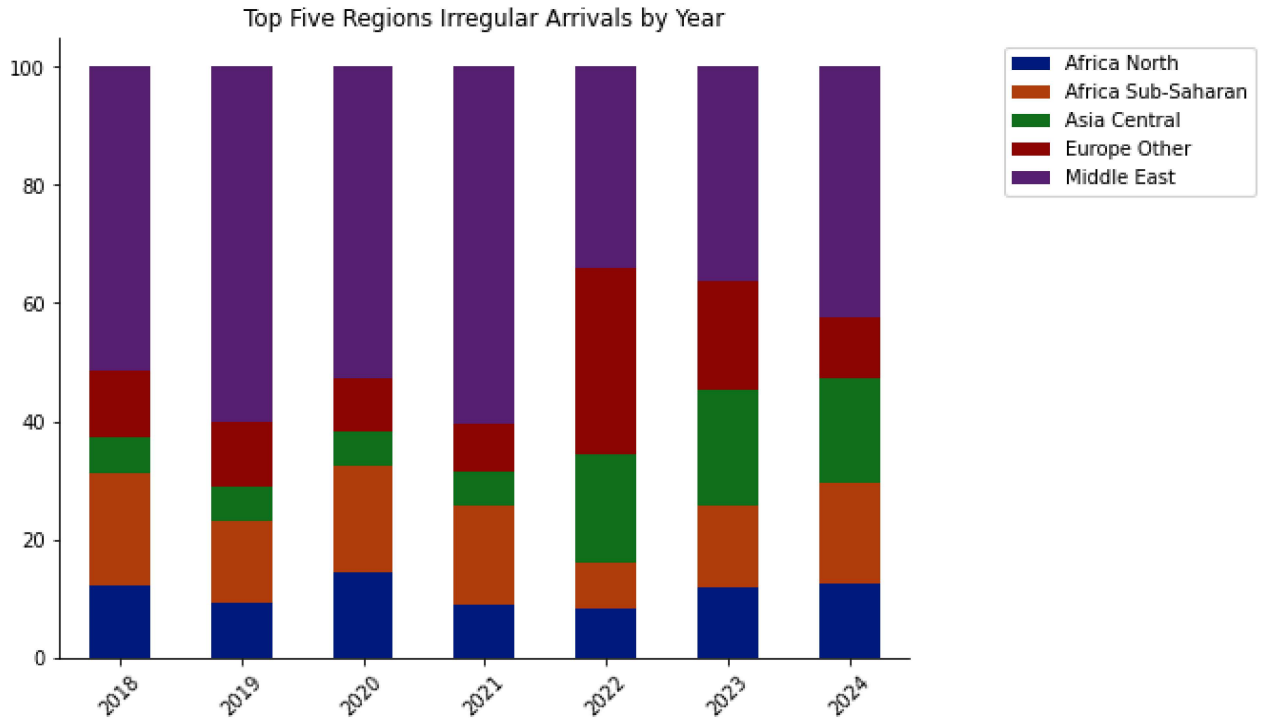
Out[323…

	Region	Year	Africa North	Africa Sub-Saharan	Asia Central	Europe Other	Middle East
0	2018	12.208968	18.839984	6.096257	11.534348	51.320444
1	2019	9.391850	13.746502	5.767524	10.941233	60.152891
2	2020	14.371872	18.054664	5.838573	9.117156	52.617734
3	2021	8.864673	16.807757	5.853746	8.156220	60.317603
4	2022	8.343190	7.678139	18.435288	31.493941	34.049443
5	2023	11.908558	13.815202	19.576585	18.234281	36.465374
6	2024	12.433348	17.057321	17.662742	10.306043	42.540547

In [327…

```
# Plot the crosstablulated data as percentage stacked bar
sns.set_palette("dark")
ax = top5_cross_percent.plot(
    x = 'Year',
    kind = 'bar',
    stacked = True,
    title = 'Top Five Regions Irregular Arrivals by Year',
    mark_right = True,
    figsize = (8,6)
    )
```

```
plt.xticks(rotation=45)
plt.xlabel("")
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.legend(bbox_to_anchor=(1.1, 1));
```



The area chart and stacked bar show the proportions by each of the top 5 regions over the period. The stacked chart is a bit more clear and shows how the proportion from the Middle East increased in 2021 relative to the other regions and then dropped back in 2022 and 2023, with Europe Other increasing significantly in 2022, with increases also from Asia Central the 'stans'.

Analysis by Country

We will continue to look at the top 5 regions in the rest of this analysis

In [328…

```
regions
```

Out[328…

```
['Middle East',
 'Europe Other',
 'Asia Central',
 'Africa Sub-Saharan',
 'Africa North']
```

In [329…

```
# Create separate dataframes

middle = df_copy[df_copy['Region']=='Middle East'].groupby('Nationality',as_index = Fal
europe_other = df_copy[df_copy['Region']=='Europe Other'].groupby('Nationality',as_inde
asia_cent = df_copy[df_copy['Region']=='Asia Central'].groupby('Nationality',as_index =
africa_sub = df_copy[df_copy['Region']=='Africa Sub-Saharan'].groupby('Nationality',as_
africa_north = df_copy[df_copy['Region']=='Africa North'].groupby('Nationality',as_inde
```

In [330…

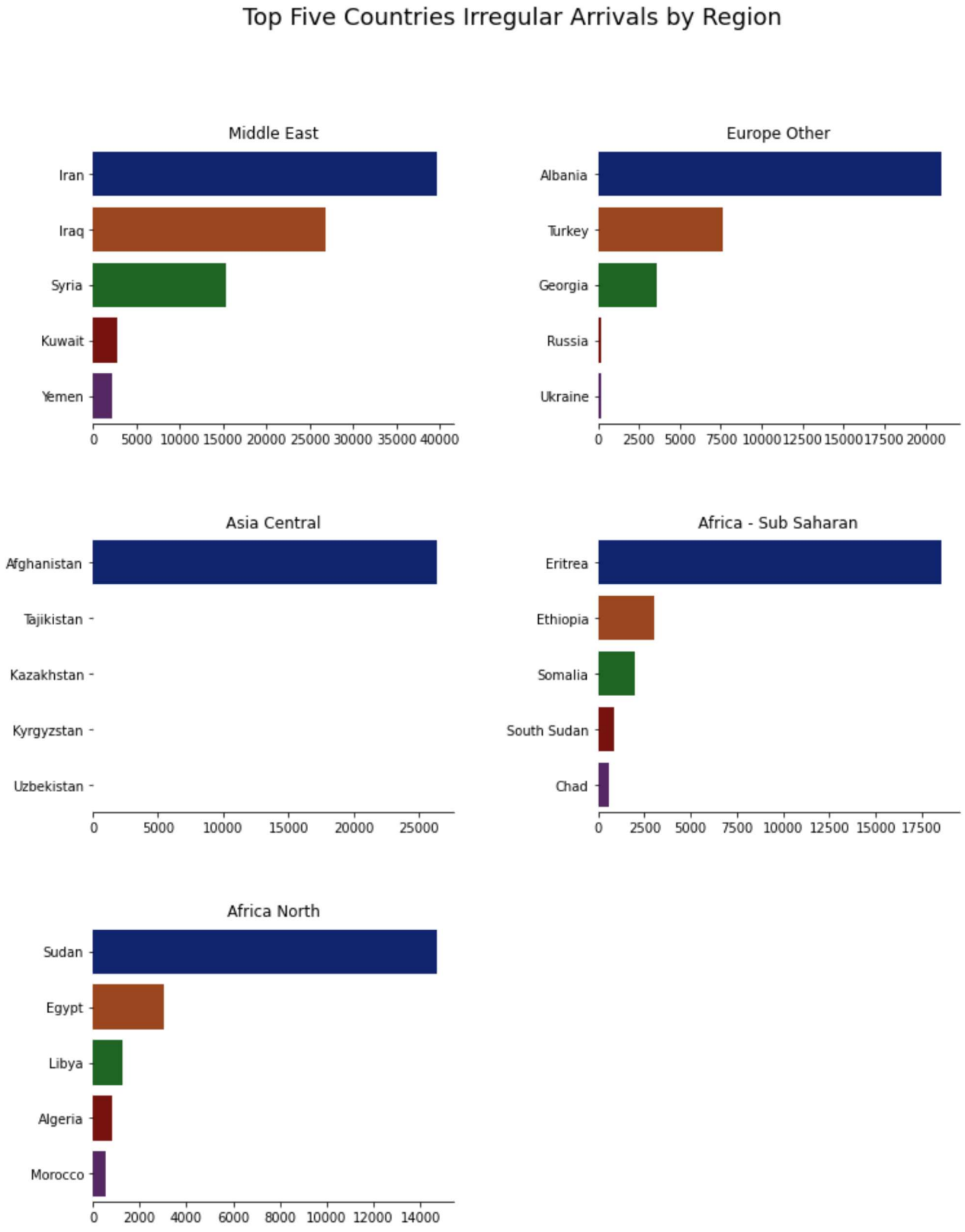
```
africa_north
```

Out[330…

	Nationality	Number of detections
5	Sudan	14703
1	Egypt	3032
2	Libya	1269
0	Algeria	808
4	Morocco	532

In [332…

```
# Plot the countries and regions
fig, axs = plt.subplots(3,2, figsize = (12,15))
fig.delaxes(axs[2,1])

fig.suptitle("Top Five Countries Irregular Arrivals by Region", fontsize=18)
plt.subplots_adjust(wspace=0.4,hspace = 0.4)

x = 'Number of detections'

# Plot barplots
a = sns.barplot(y=middle.Nationality, x=x, data=middle, ax = axs[0,0],palette = 'dark')
b = sns.barplot(y=europe_other.Nationality, x=x, data=europe_other, ax = axs[0,1],palet
c = sns.barplot(y=asia_cent.Nationality, x=x, data=asia_cent, ax = axs[1,0],palette = '
d = sns.barplot(y=africa_sub.Nationality, x=x, data=africa_sub, ax = axs[1,1],palette =
e = sns.barplot(y=africa_north.Nationality, x=x, data=africa_north, ax = axs[2,0],palet

plots = [a,b,c,d,e]

# Cycle over the plots and remove axis labels and set facecolour
for p in plots:
    p.set(xlabel = None)
    p.set(ylabel = None)
    p.set_facecolor('white')

# Add titles to the plots
axs[0,0].set_title("Middle East", fontsize = 12)
axs[0,1].set_title("Europe Other", fontsize = 12)
axs[1,0].set_title("Asia Central", fontsize = 12)
axs[1,1].set_title("Africa - Sub Saharan", fontsize = 12)
axs[2,0].set_title("Africa North", fontsize = 12)

# Remove frames
list_frames = [axs[0,0],axs[0,1],axs[1,0],axs[1,1],axs[2,0]]
for l in list_frames:
    l.spines['top'].set_visible(False)
    l.spines['right'].set_visible(False)
    l.spines['left'].set_visible(False)
```



In [333…

```
# Get the top countries by number of arrivals
all_countries = df_copy.groupby('Nationality',as_index = False)['Number of detections']
all_countries['Average'] = all_countries['Number of detections']/6
all_countries.sort_values(by = 'Average',ascending = False).round(0)
```

Out[333…

	Nationality	Number of detections	Average
77	Iran	39598	6600.0

	Nationality	Number of detections	Average
78	Iraq	26823	4470.0
0	Afghanistan	26365	4394.0
1	Albania	21017	3503.0
52	Eritrea	18577	3096.0
...	...	...	...
136	Saint Barthelemy (French)	1	0.0
40	Cyprus	1	0.0
38	Croatia	1	0.0
34	Comoros	1	0.0
124	Paraguay	1	0.0

178 rows × 3 columns

In [572…

```
# Create interactive map of average migration
fig = px.choropleth(locationmode="country names",
                    locations = all_countries.Nationality,
                    title = 'UK Irregular Arrivals Average 2018 to 2024',
                    color = all_countries['Number of detections'],
                    color_continuous_scale='Ice_r',

                    )
fig.update_layout(
    title_x=0.5,
    title_y=0.85)

fig.update_layout(coloraxis_colorbar_title_text = '')
fig.show()
```

Boat Arrivals Claiming Refugee Status

Asylum Claims and Initial Decisions Data

Data

In [489…
```
ref = pd.read_csv(r'C:\Users\imoge\AllMLProjects\Data\AsylumClaimsUpdated.csv')
```

In [490…
```
ref.shape
```

Out[490…
```
(5980, 10)
```

In [491…
```
ref.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5980 entries, 0 to 5979
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Year            5980 non-null   int64 
 1   Quarter         5980 non-null   object
 2   Asylum claim    5980 non-null   object
 3   Nationality     5980 non-null   object
 4   Region          5980 non-null   object
 5   Applicant type  5980 non-null   object
 6   UASC            5980 non-null   object
 7   Sex             5980 non-null   object
 8   Age Group       5980 non-null   object
 9   Arrivals        5980 non-null   int64 
dtypes: int64(2), object(8)
memory usage: 467.3+ KB
```

In [492…
```
ref.head()
```

Out[492…

	Year	Quarter	Asylum claim	Nationality	Region	Applicant type	UASC	Sex	Age Group	Arrivals
0	2018	2018 Q2	Asylum claim raised	Ukraine	Europe Other	Dependant	Non-UASC	Female	17 and under	2
1	2018	2018 Q2	Asylum claim raised	Ukraine	Europe Other	Dependant	Non-UASC	Female	25 to 39	1

	Year	Quarter	Asylum claim	Nationality	Region	Applicant type	UASC	Sex	Age Group	Arrivals
2	2018	2018 Q2	Asylum claim raised	Ukraine	Europe Other	Main applicant	Non-UASC	Male	40 and over	1
3	2018	2018 Q3	Asylum claim raised	Afghanistan	Asia Central	Main applicant	Non-UASC	Male	18 to 24	1
4	2018	2018 Q3	Asylum claim raised	Afghanistan	Asia Central	Main applicant	Non-UASC	Male	25 to 39	1

In [493…

```
ref.tail()
```

Out[493…

	Year	Quarter	Asylum claim	Nationality	Region	Applicant type	UASC	Sex	Age Group	Arrivals
5975	2024	2024 Q4	No asylum claim raised	Turkey	Europe Other	N/A - No asylum claim	Non-UASC	Male	18 to 24	3
5976	2024	2024 Q4	No asylum claim raised	Turkey	Europe Other	N/A - No asylum claim	Non-UASC	Male	25 to 39	3
5977	2024	2024 Q4	No asylum claim raised	Turkey	Europe Other	N/A - No asylum claim	Non-UASC	Male	40 and over	3
5978	2024	2024 Q4	No asylum claim raised	Vietnam	Asia South East	N/A - No asylum claim	Non-UASC	Male	25 to 39	2
5979	2024	2024 Q4	No asylum claim raised	Yemen	Middle East	N/A - No asylum claim	Non-UASC	Male	25 to 39	1

In [494…

```
ref['Year'].unique()
```

Out[494…

```
array([2018, 2019, 2020, 2021, 2022, 2023, 2024], dtype=int64)
```

In [495…

```
ref['Quarter'].unique()
```

Out[495…

```
array(['2018 Q2', '2018 Q3', '2018 Q4', '2019 Q1', '2019 Q2', '2019 Q3',
       '2019 Q4', '2020 Q1', '2020 Q2', '2020 Q3', '2020 Q4', '2021 Q1',
       '2021 Q2', '2021 Q3', '2021 Q4', '2022 Q1', '2022 Q2', '2022 Q3',
       '2022 Q4', '2023 Q1', '2023 Q2', '2023 Q3', '2023 Q4', '2024 Q1',
       '2024 Q2', '2024 Q3', '2024 Q4'], dtype=object)
```

We do not have any entries for Q1 of 2018

In [498…

```
ref['Asylum claim'].unique()
```

Out[498…
```
array(['Asylum claim raised', 'No asylum claim raised'], dtype=object)
```

In [496…
```
ref['UASC'].unique()
```

Out[496…
```
array(['Non-UASC', 'UASC'], dtype=object)
```

USAC - Whether the asylum applicant was an unaccompanied asylum seeking child (UASC).

In [921…
```
# Strip out commas and set to integer for the applications
#ref['Applications'] = (ref['Applications'].str.replace(',', '', regex=True).astype(int
```

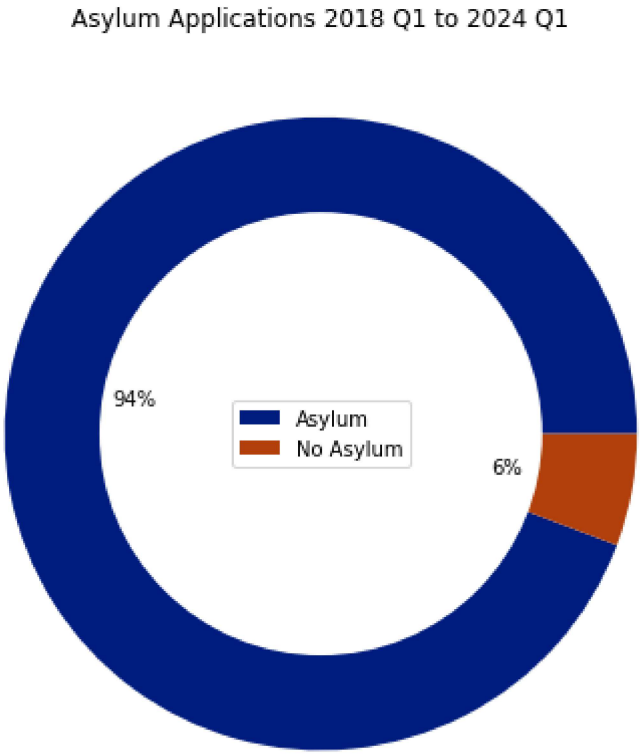
What Proportion of Small Boat Arrivals Raise a Refugee Application?

In [500…
```
apps = ref.groupby(['Asylum claim'], as_index = False)['Arrivals'].sum()
apps
```

Out[500…

	Asylum claim	Arrivals
0	Asylum claim raised	139465
1	No asylum claim raised	8469

In [505…
```
# Draw pie
plt.figure(figsize=(6, 6))

plt.pie(x = apps['Arrivals'], autopct='%.0f%%')

#draw circle
centre_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

# Add title and legend
plt.title("Asylum Applications 2018 Q1 to 2024 Q1")
labels = ['Asylum','No Asylum']
plt.legend(labels = labels, bbox_to_anchor=[0.5, 0.5], loc='center')

# Show
plt.tight_layout()
plt.show()
```



The vast majority of those arriving by small boat from 2018 Q1 to 2024 Q1 claimed asylum

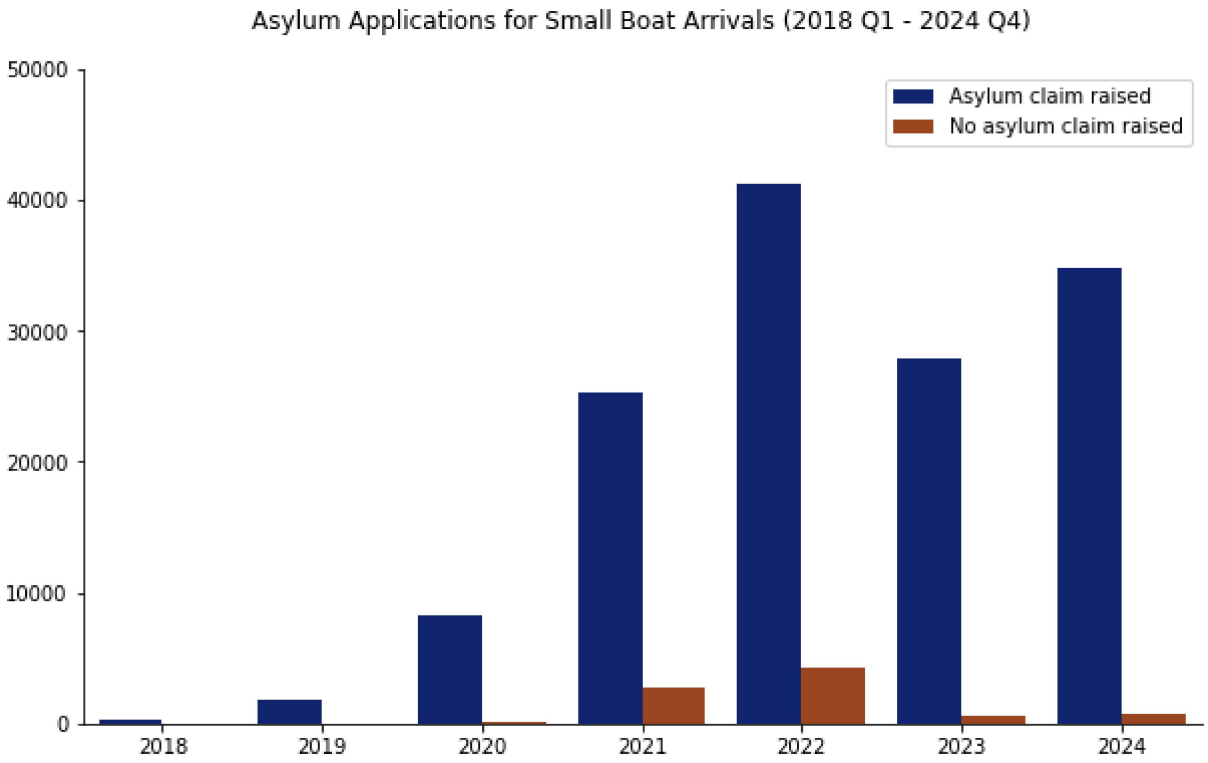
How has this changed over time?

In [507…

```
# Group by year and applications
asy_year = ref.groupby(['Year','Asylum claim'], as_index = False)['Arrivals'].sum()
```

In [568…

```
# Plot the claims by year
plt.figure(figsize=(10, 6))

ax = sns.barplot(data = asy_year,
                 x ='Year',
                 y = 'Arrivals',
                 hue = 'Asylum claim',
                 #errorbar = None,
                 palette = 'dark')
plt.title("Asylum Applications for Small Boat Arrivals (2018 Q1 - 2024 Q4)", pad = 20)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.ylabel("Number", labelpad = 10)
plt.xlabel("")
plt.ylabel("")
plt.ylim(0,50000)
ax.legend(title=None) ;
```



In [571...

```
asy_pivot = asy_year.pivot_table(index='Year', columns='Asylum claim', values='Arrivals
asy_pivot['Total'] = asy_pivot['Asylum claim raised'] + asy_pivot['No asylum claim rais
asy_pivot['% raised'] = round(asy_pivot['Asylum claim raised']/asy_pivot['Total']*100,1
asy_pivot
```

Out[571...

Asylum claim Year	Asylum claim raised	No asylum claim raised	Total	% raised
2018	280	5	285	98.2
2019	1813	20	1833	98.9
2020	8276	130	8406	98.5
2021	25276	2747	28023	90.2
2022	41188	4290	45478	90.6
2023	27819	607	28426	97.9
2024	34813	670	35483	98.1

Seems to have been fairly consistent over the period with the exception of 2021 and 2022 where the proportion of those arriving by small boat not claiming asylum increased relative to those that did. It is not clear why.

Albanian Returns Agreement: The agreement between the UK and Albania in late 2022 facilitated the return of many Albanian nationals whose asylum claims were refused or withdrawn. This agreement significantly impacted the overall composition of asylum claims from small boat arrivals.

We don't need to go through all the other features such as age, sex, nationality etc as we know that the vast majority of arrivals were from small boats and the vast majority of those claim asylum and

we have already looked at the other features in relation to method of entry.