DATA VISUALISATION COURSEWORK

ICAO Flight Accident Report

Contents

List of Figures

List of Tables

1.0 Introduction

1.1 Recent press reports highlighted that despite a 41% fall in flight departures due to the Covid 19 pandemic in 2020, the number of fatalities from commercial flights had increased. A total of 299 lives were lost in 40 accidents involving large commercial passenger airlines. This compares to 2019 when there were almost double the number of accidents and 274 fatalities, suggesting that air travel was relatively less safe in 2020 [1].

1.2 There have been significant increases in safety procedures over the last fifty years and air travel is generally considered to be a very safe means of travel [2]. Accidents that occur are usually attributed to human error, weather conditions and maintenance issues [3]. Whilst fatal air accidents garner the most press attention, many non-fatal accidents occur varying widely in type and severity, including injuries due to turbulence, air contamination in the cabin, passengers falling off the airstairs when embarking and airport staff being hurt by aircraft when manoeuvring.

1.3 Airline accidents can dramatically affect the public perception of safety and therefore the reputation and profitability of an airline operator. In 2014, Malaysia Airlines suffered a 40% reduction in market value after flight MH370 went missing on 8 March while flying from Kuala Lumpur to Beijing [4]. In 2008, a Qantas flight from Singapore to Perth, Australia, experienced severe turbulence. There were no fatalities, but 110 passengers and 9 crew were hurt, 12 seriously and 39 hospitalised, attracting significant press attention [5].

1.4 This report compares 2020 to previous years air accident data to determine whether it can be considered to be more or less safe. Relationships between the number of air accidents and other reported factors are also investigated to develop a profile associated with higher accident rates. Sections 2 and 3 outline the dataset and cleaning process and Section 4 establishes the research objectives and visualisation methodology. Findings are contained in Section 5 with conclusions in Section 6.

2.0 Dataset

2.1 Data was gathered from the International Civil Aviation Organisation (ICAO), a UN agency involved with air safety. The ICAO collects data relating to civil aviation from 193 member states and provides statistical reports designed to guide the development of policies and standards throughout the industry [6].

2.2 The ICAO defines an air accident as covering the duration of time from when a person boards a flight to travel until all have disembarked, and involving instances where an aircraft:

- is missing or inaccessible,
- is damaged so as to adversely affect the flight and require major repair,
- causes the death or serious injury of a person including those on the ground [7].

2.3 All available accident and departure data was downloaded from the API (2008 to 2020). Other data was gathered to create new comparator variables. Appendix C lists all variables used with references.

3.0 Data Cleaning and Exploration

3.1 The dataset was filtered to include all flights commercial aircraft above 5.7 tonnes. Craft below this weight are considered light aircraft such as crop-dusters and small private jets. Helicopters and military craft were also removed.

3.2 Preliminary examination identified two large incidents were omitted in 2020, both instances where the planes had been 'shot-down'. Previous ICAO data included similar incidents and although they are not attributable to an operator's actions, they are part of the safety profile likely to impact passenger confidence and are included.

3.3 The data had a large number of missing entries, inconsistent labelling, categorisation errors and duplicates. Incident data was checked where possible to accident reports filed online [16][17]. More extensive and time-consuming data cleaning was required than initially anticipated to produce a reliable dataset, which was surprising given the provenance and usage of the dataset by global stakeholders.

3.4 Basic plots and statistics were prepared to help to identify areas of interest and three potential indicators of flight safety identified including the number of:

- accidents/incidents,
- fatal accidents and
- fatalities.

4.0 Research Questions and Approach

4.1 The following research questions were investigated with visualisations:

- How does the number of total, fatal accidents and fatalities in 2020 compare to the previous 12 years of available aircraft accident data?
- Is there any relationship between the number of total and fatal accidents and the:
    - time that an accident occurs,
    - plane size, model, and engine type,
    - flight stage,
    - airline operator,
    - safety certification status,
    - state of registration and
    - state of occurrence?
- Is the number of air accidents related to the average age and size of fleet and the income level per capita of a country?
- What insights can be drawn from the findings to inform the safety of air travel?

4.2 The visualisations are based on three key principles of design - trustworthiness, accessibility, and elegance [18] and the Gestalt laws of visual perception applicable to visualisations – simplicity, proximity, similarity, continuity, closure and connectiveness [19]. Key information is emphasised by focussing on the pre-attentive attributes of colour, position, and form (including dimension, orientation, size, shape, markings, and grouping) to encode the relevant categorical and quantitative aspects in the data [20][21].

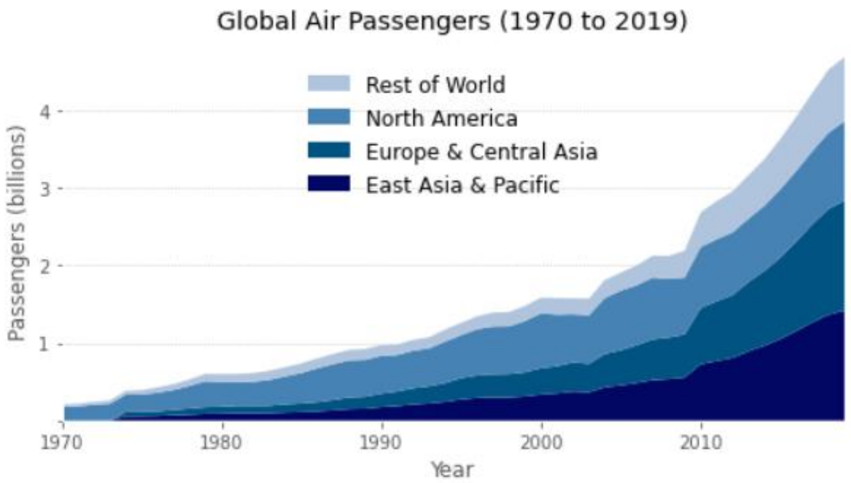
4.3 Charts are produced with the Matplotlib library in Python with a non-technical audience in mind. A consistent, minimalist chart style is used with muted colours to reduce the non-data ink [22] and designed to emulate a business report in the airline industry. The colour palette was also checked for the most common colour blindness compatibility [23]. Visualisations are constructed from the primary chart components (lines, bars, and points) and secondary components (annotations, trend lines, tick-marks, scale, and legends) and each is accompanied by the source data, explanatory notes and an italicised caption providing a brief summary of the key points. An explanation as to why that chart type was chosen is included below the caption.

5.0 Findings

5.1 Global Air Trends

5.1.1 Global air travel has increased exponentially over the last two decades and every hour, around 400 commercial flights, transporting passengers and cargo take off somewhere in the world [24]. In 2019 4.4 billion passengers were transported on 37 million flights [8]. Figure 1 shows the trend in global passenger numbers by region from 1970 to 2019.

Figure 1 - Air Passenger Numbers By Region (1970-2019)



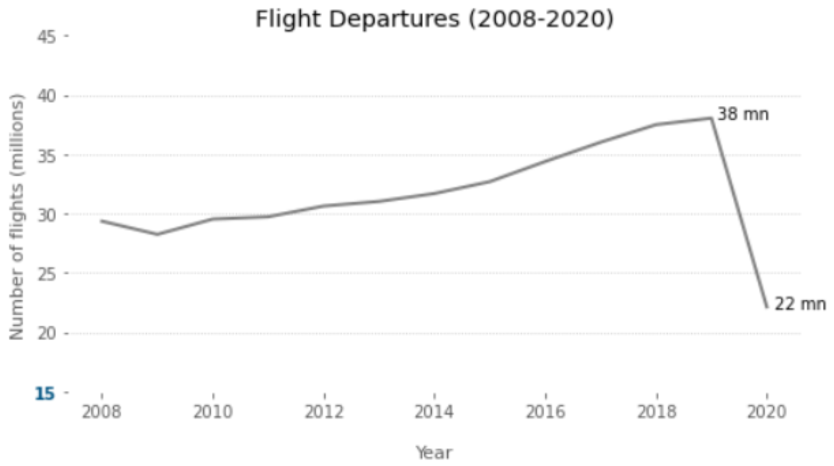
Source: World Bank - figures available to 2019
Note: Rest of World is Latin America, Caribbean, Middle East, North Africa, Sub-Saharan Africa, and South Asia

Global air travel has grown rapidly over the last 50 years and most notably in the East Asia & Pacific and European & Central Asian regions over the last decade.

5.1.2 A stacked area plot allows both the trend in flights over time and the whole to part relationships amongst the main regions to be shown simultaneously. An aspect ratio (width/height) of 2.0, commonly used for time series emphasises the horizontal over the vertical and all borders are removed as the law of closure completes the picture. The colour-mix delineates the regions and using a white background and unboxed legend stops attention being drawn from the data. Reversing the legend to match the order of the data and faded grid lines aid interpretation without dominating the chart and both figure titles and a chart title are used, so that if the chart is used elsewhere, the information is not lost.

5.1.3 Flight numbers have increased to service this global passenger demand. Figure 2 shows the trend in flight numbers from 2008 to 2019.

Figure 2 – Global Departures By Year (2008-2020)



Source: ICAO 2020
Note 1 : Includes data for the years available only 2008-2020.
Note 2: The y-axis (Number of Flights) has a non-zero start point.

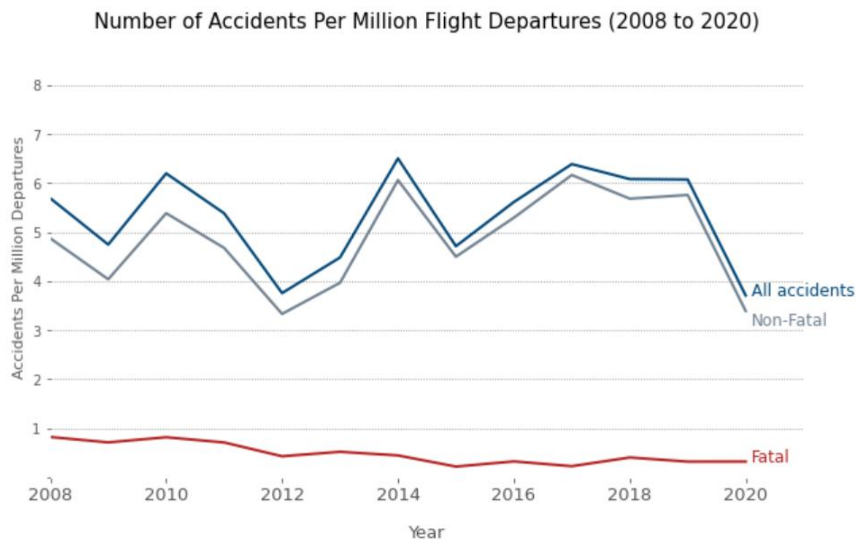
A significant drop in 2020 departures reflects the huge impact that travel restrictions due to the SARS COV-19 pandemic had on worldwide air movements.

5.1.4 A line plot exploits the law of simplicity and clearly shows the key point, which is the 2020 drop in flights. The two most recent years are emphasised with annotation in close proximity to the points of reference. The baseline does not start at zero to focus the eye on the data not blank space. This is acceptable as the line focuses on change not comparison of length [25][26] but the non-zero start point is marked blue, and an explanatory note added.

5.2 Number of Fatal and Non-Fatal Accidents

5.2.1 Against this backdrop of global demand, air accident numbers are shown in Figure 3.

Figure 3 - Air Accident Trends (2008-2020)



Source: ICAO 2020

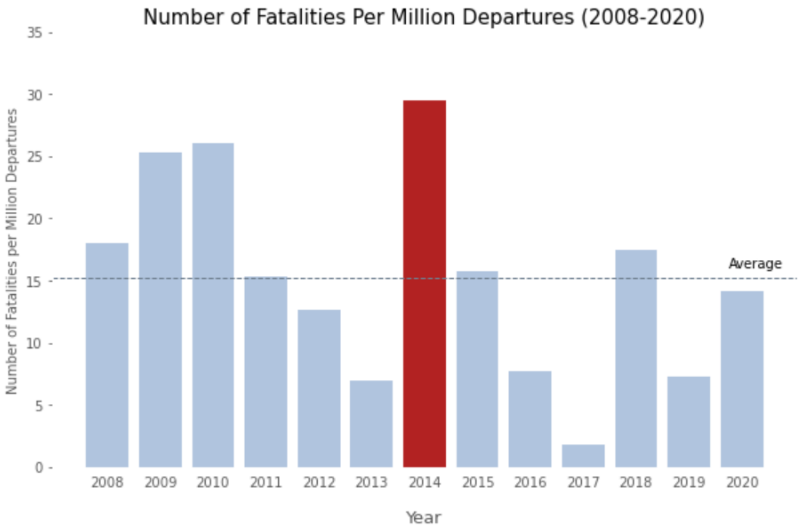
The air accident rate peaked in 2014 with the lowest rate in 2020. The vast majority of accidents are non-fatal with fatal accidents remaining well below 1 accident per million flight departures and trending downwards over the period.

5.2.2 A line plot shows the variability in the accident rate over time as well as allowing comparison between the fatal and non-fatal accident rates with each line interpreted as a single group of points through the law of connectiveness. Proximity and similarity are exploited through labels being close to the point of reference and color-coded to match the lines. Y-axis grid lines help interpretation of the lines at each time point.

5.3 Number of Fatalities

5.3.1 Although most accidents are do not result in fatalities, large loss of life in a single accident can be very damaging to an airline's reputation. The sheer numbers of lives lost from large airliner crashes are usually viewed as more significant particularly in terms of news attention than many fatal incidents that occur each year involving smaller jets and cargo planes. Figure 4 shows the number of fatalities across the period expressed per million departures.

Figure 4 - Fatalities By Year (2008-2020)



Source: ICAO 2020
Note 1: The loss of Malaysia Airlines MH370 and MH17 and an Asia Air flight represented 80% of the fatalaties in the year.

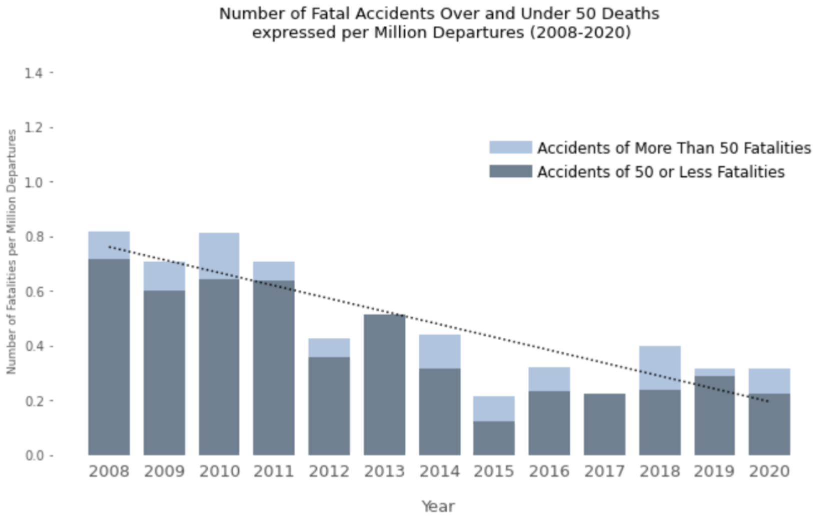
Over the last twelve years, the largest number of fatalities occurred in 2014 when there were 1,112 fatalities mostly attributable to three major accidents. Fatalities in 2020 (314) were a little higher than 2019 (278) but when expressed per million departures, 2020 appears significantly worse, although still below the average for the period.

5.3.2 A bar chart focuses attention on the variance in individual values rather than the change over time. Although 2014 is clearly the largest bar, colour as a powerful pre-attentive attribute provides a focal point to emphasise the large loss of life in this year and provides a reference point against which the other bars can be compared. An average line draws attention back to the fact that 2020 although higher than the previous year is still below the average for the period. The bars share a common base line (law of continuity) so x-ticks are not required, and grid lines are removed as the bars can easily be compared.

5.4 Number of Accidents with More or Less than 50 Fatalities

5.4.1 The number of fatal accidents split between larger accidents recording over 50 deaths and those of 50 and under is shown in Figure 5.

Figure 5 – Fatal Accidents Involving More or Less Than 50 Fatalities (2008-2020)



Source: ICAO 2020

Fatal accidents as a whole have trended downwards, possibly due to continuous improvements in air safety standards [3]. Incidents involving 50 or less deaths constitute the majority of the total, with larger accidents being comparatively rare anomalous events. Whilst 2019 and 2020 had a similar number of fatal accidents overall, two large accidents in 2020 – the 'downing' of a Ukrainian plane over Iran and the crashing of a Pakistani airliner – accounting for 80% of the deaths in the year [16].

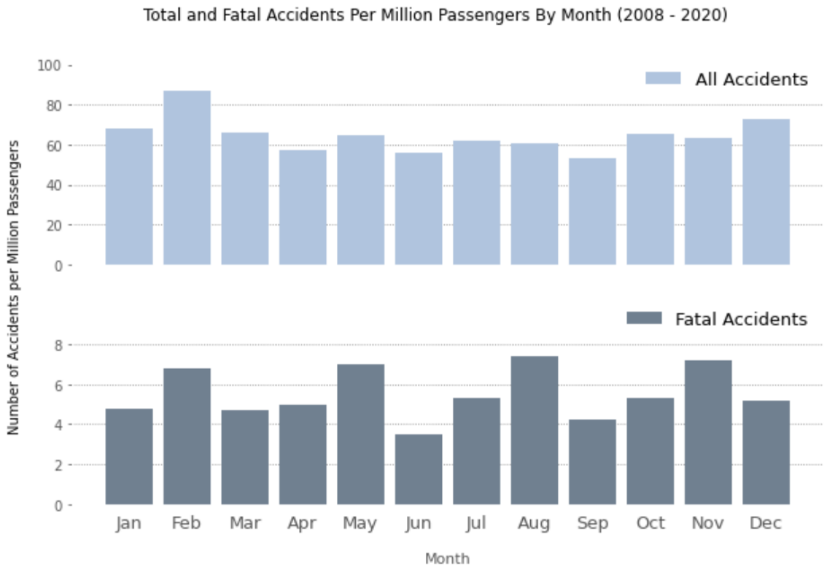
5.4.2 A stacked bar chart shows both the split and movement over time, aided by a line emphasising this decrease. The darker colour is used to focus attention on the downwards trend in accidents below 50 deaths which is an important insight. The legend is reversed and placed close to the data to aid interpretation.

5.4.3 The number of fatalities per accident is directly related to the size of aircraft involved and somewhat random in nature. In comparison the number of accidents could be related to many factors and these are discussed below.

5.5 When do Most Accidents Occur?

5.5.1 Certain months can experience worse weather or have more traffic congestion, which might impact on the relative safety of a flight. Figures 6 and 7 chart total and fatal accidents per million passengers by month and day.

Figure 6 – Total and Fatal Accident Rates By Month

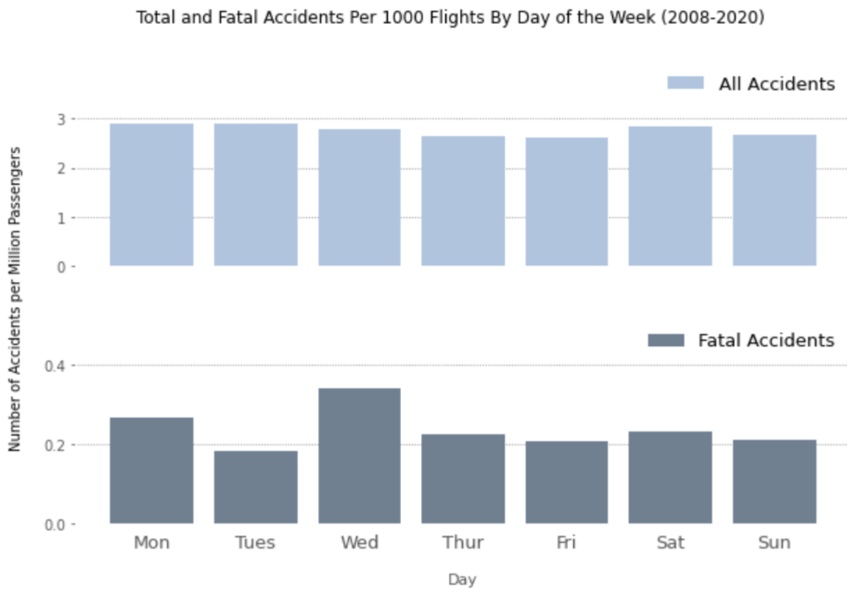


Source: ICAO Statistics 2020 and WorldPop 2020

Note: Average monthly passenger numbers over the period 2010 to 2018 are used since data was not available for flights or for the whole period. This is a simplistic assumption but allows comparison between months.

The accident rate is highest in February, dipping over the busy northern hemisphere summer season from June through to September, generally recognised as a safer time to fly due to weather [27]. The highest fatal accident rate is in August when airspace and airports can be congested [28]. Rates peak each quarter, possibly due to reporting points but further research is needed to explain this.

Figure 7 - Total and Fatal Accidents By Weekday



Source: ICAO 2020 and FlightRader24.com (Daily Flight Data 2019)

Note: Daily flight data for 2019 is used as the base for this metric as data for the period was not available. This is a simplistic assumption but allows comparison between days.

The total accident rate is slightly higher on Mondays and Tuesdays and fatal accidents peak on Wednesdays. Weekends and Mondays are usually the busiest flying days, with midweek the least busy [29][30], so other factors rather than busy airspace are likely to be involved.

5.5.2 Subplots with a shared x-axis allow direct comparisons between total and fatal accident rates. These are shown separately due to different y-axis scales and although clustered bar charts are more visually appealing they would require confusing dual axes. As the bars are very similar in length, grid lines are used for interpretation and a consistent colour scheme separates the accident categories.

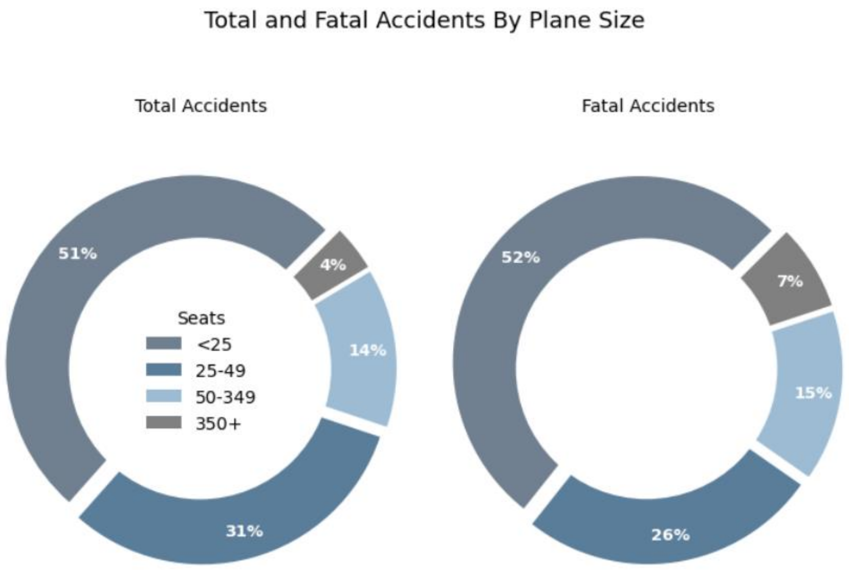
5.6 What Sized Planes have the most Accidents?

5.6.1 Operators run different sized aircraft according to their routes and passengers and they can be classified by seat numbers:

- Very small - <25
- Small – 25-49
- Medium – 50-349
- Large – 350+

Figure 8 shows the proportion of total and fatal accidents accounted for by each seat capacity.

Figure 8 – Total and Fatal Accidents Size of Aircraft



Source: ICAO Statistics 2020 and The Travel Insider

The vast majority of accidents and fatal accidents occur in small planes with less than 50 seats. Larger planes with more than 350 seats account for less than 10% of the total for all accidents and fatal accidents alike.

5.6.2 Although pie chart/donut charts can distort information, they are used here since there are only four slices, the proportions are quite different, and it enables easy comparison. Exploding the slices, rounded percentages and setting the text to white rather than black helps legibility. The blank space is used for the legend as per the law of proximity.

5.7 What Models of Plane have the most Accidents?

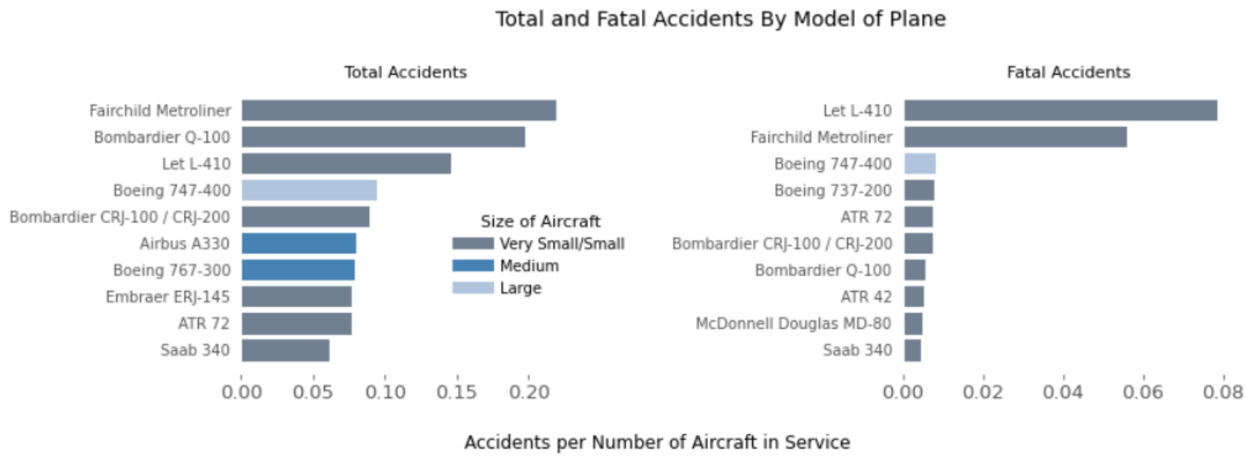
5.7.1 Aircraft with more than 50 accidents are shown in Table 1. Based on counts, the Boeing 737-800, a medium sized jet reported the most accidents and all but two are Boeing and Airbus models.

Table 1 – Accidents By Aircraft Model for Models with more than 50 Accidents (2008 -2020)

Model	Size(seats)	Engine	Accidents
Boeing 737-800	**Medium**	**Jet**	**211**
Airbus A320	Medium	Jet	178
Bombardier Q-100	Small	Turboprop	106
Airbus A330	Medium	Jet	95
Airbus A321	Medium	Jet	92
Airbus A319	Small	Jet	91
ATR 72	Small	Turboprop	72
Boeing 767-300	Medium	Jet	59
Boeing 737-700	Small	Jet	56
Boeing 737-300	Small	Jet	55

When the accident rate for all aircraft is expressed as a proportion of the number of planes in service (actively flying), this reveals a different picture shown by Figure 9.

Figure 9 – Total and Fatal Accidents per Number of Aircraft in Service



Source: ICAO Statistics 2020, number of planes in service from DVB Commercial Aircraft Report, Airfleets.net and FlightGlobal.com census.

Note 1: The number of planes in service is the best figure available from the sources above and assumed to be unchanged over the period. This is a simplistic assumption but allows a comparison between years.

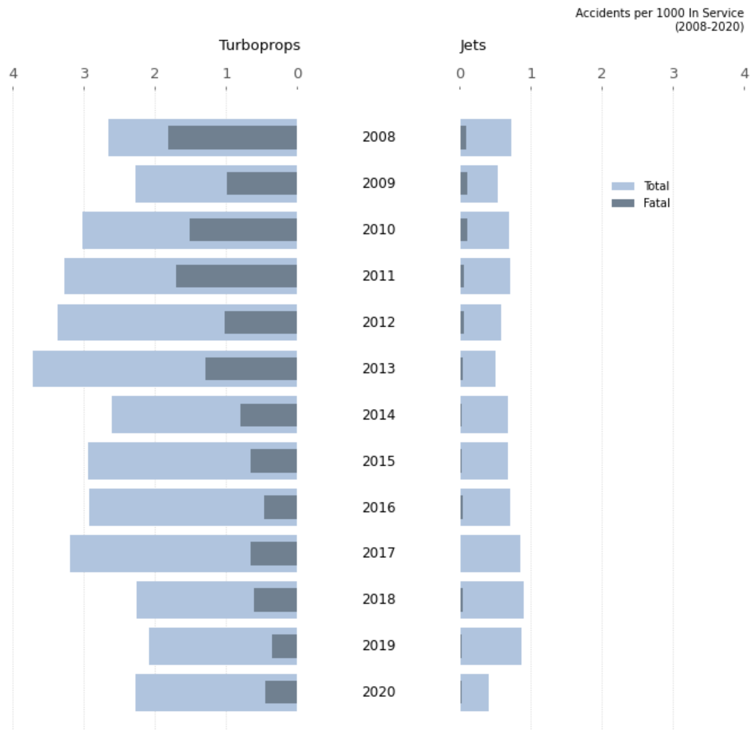
The Boeing 737-800, whilst being at the top of Table 1 does not appear in the chart once the number of planes flying is accounted for. Only one large jet is shown, the Boeing 747-400, an older model first flown in 1989 [12]. The top 3 planes for total and top 2 for fatal accidents are 19 seat turboprops.

5.7.2 Horizontal bar subplots enable long aircraft model names to be accommodated . The y-axis is inverted to draw attention to the highest values at the top left where the eye naturally rests before moving across to the other chart. Gridlines are removed to reduce clutter and colour is used to group similar planes by size. The legend is placed in open space on the first chart with the most categories to interpret and close to the point of reference.

5.8 What Engine Types have the most Accidents?

5.8.1 Plane engine types include jets, turboprops, and a small number of piston engine aircraft. Jets and turboprops (turbines) are regarded as much safer than the older piston engines commonly found in smaller aircraft [3]. Turboprops are often used for shorter distances as they are cheaper, more fuel efficient, able to land safely on shorter runways and have lower maintenance and insurance costs. Jets are primarily used for longer flights at higher altitudes where fuel efficiencies can be gained [31]. Figure 10 compares the total and fatal accident rate for the main categories of turboprops and jets.

Figure 10 - Total and Fatal Accidents by Engine Type



Source: ICAO Statistics 2020 and number of planes in service from DVB Commercial Aircraft Report, Airfleets.net and FlightGlobal.com census.

Note 1: A very small number of piston engines are excluded from the data for simplicity.
Note 2: The average accident rate in a year is the number of accidents for each model of aircraft as a proportion of the number of that model in service, averaged. The number in service is the best figure available and assumed to be unchanged over the period which is a simplistic assumption but allows a comparison between years.

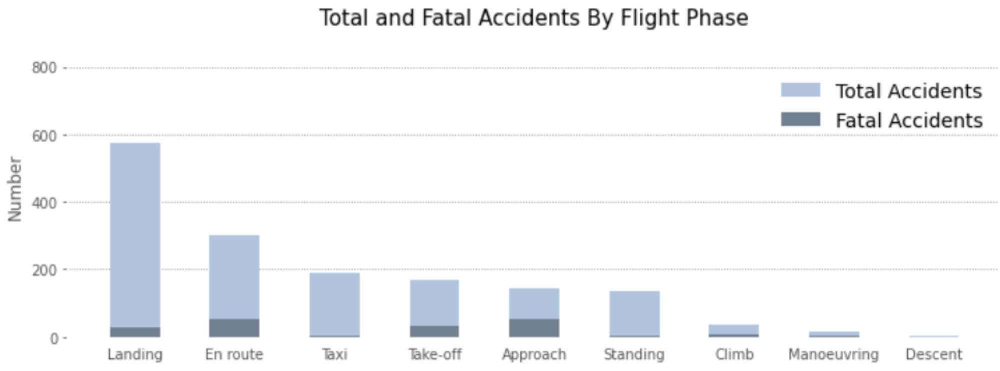
Turboprops have a higher accident and fatal accident rate than jets with total and fatal accidents highest in 2013 and 2008, respectively. Jet accident rates are lowest in 2020 and remain below 1 per thousand craft in service over the period.

5.8.2 The tornado chart exploits the law of simplicity through symmetry to show the relative importance of variables over time. The order is not sorted largest to smallest as usually found this in this type of chart since the years run consecutively. The order of years is reversed for ease of reading and the x-axis is moved to the top where the eye naturally rests. The x-axis scale is kept constant for both categories to allow quick visual comparison. Vertical grid lines are faded to stop them dominating the chart .

5.9 At what Stage of Flight do Accidents Occur?

5.9.1 Flights can encounter a wide range of problems, including bird strikes, cabin depressurisation, turbulence, and accidents on the tarmac when manoeuvring. Figure 11 shows the number of total accidents and fatal accidents split by stage of flight.

Figure 11 – Total and Fatal Accidents By Flight Stage



Source: ICAO Statistics 2020

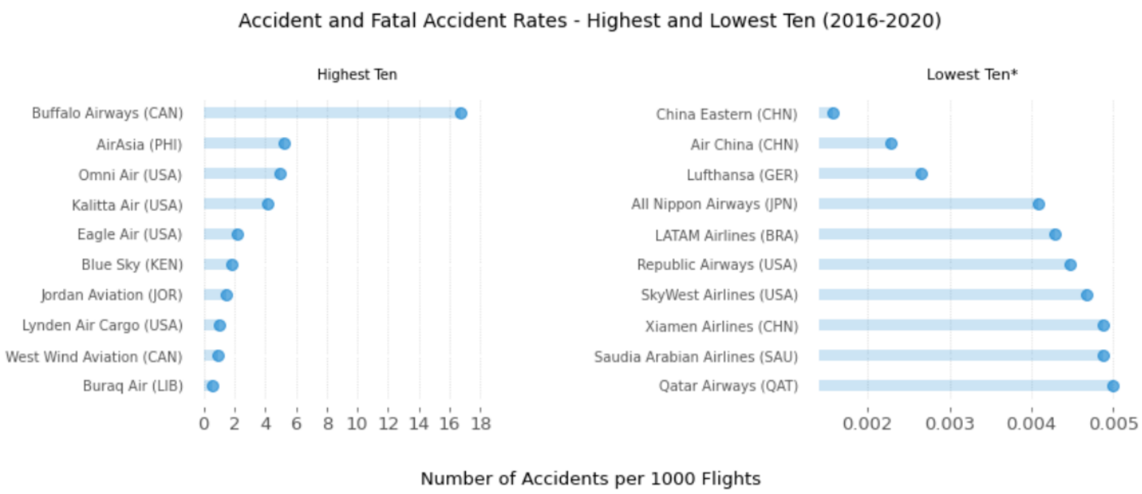
Most accidents occur on landing and fatal accidents mainly occur en-route and on approach to landing. Of interest is the number of accidents whilst planes are stationary.

5.9.2 The human eye naturally follows lines and sequences to create pathways (law of continuity) so a vertical stacked bar chart sorted by size enables the eye to note the largest value and move naturally left to right. The actual values are not as important as the general picture, so although y-axis grid lines are used, they are set at large intervals to reduce clutter.

5.10 Which Operator has the most Accidents?

5.10.1 Within each country, operators, run the commercial passenger and cargo transport on domestic and international routes. Figure 12 shows total accidents expressed per 100,000 international and domestic flights for the top and bottom ten operators and Figure 13, shows all operators reporting fatal accidents.

Figure 12 – Total Accidents by Operator – Highest and Lowest Ten (2016-2020)

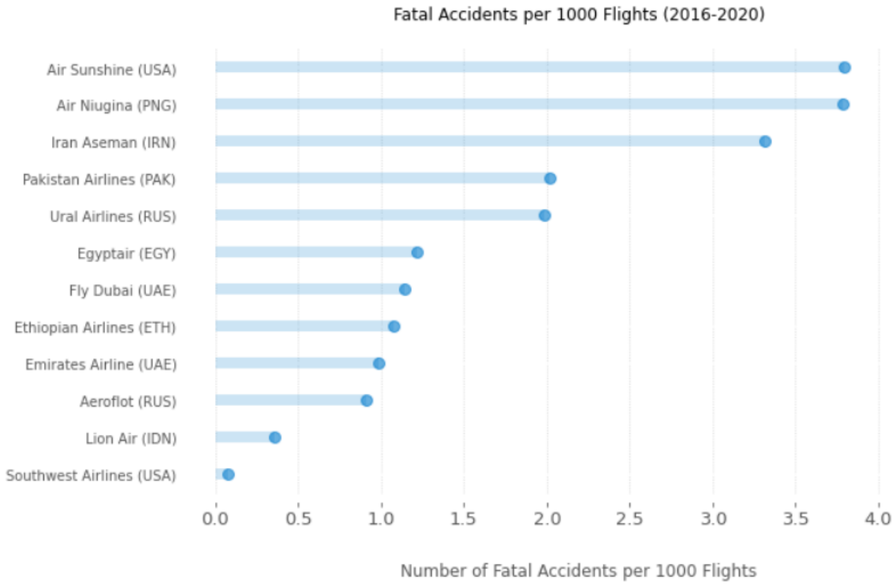


Source: ICAO Statistics 2020

Note 1: The number of accidents and fatal accidents by operator is for the last five years as per the ICAO Operator Risk Profile.
Note 2: The average number of flights for the five-year period is used as a base to allow comparison between operators and only includes those operators reporting annual flight data.

Operators with the highest accident rates are mostly small, domestic charter or cargo airlines. In comparison, the lowest ten operators reporting at least one accident tend to be larger international passenger airlines, with China Eastern, being the best of the group.

Figure 13 – Fatal Accident Rate By Operator (2016-2020)



Source: ICAO Statistics 2020

Note 1: Chart excludes two operators – West Wind Aviation (CAN) and Angara Airlines (RUS) with accident rates of 98 and 32 per 1000 flights, respectively. Each had just one fatal accident but very low annual flights skewing the data and both had safety certificates withdrawn.
Note 2: The number of accidents and fatal accidents by operator is for the last five years as per the ICAO Operator Risk Profile and shows all the operators recording fatal accidents.
Note 3: The average number of flights for the five-year period is used as a base to allow comparison between operators and chart only includes those operators reporting annual flight data.

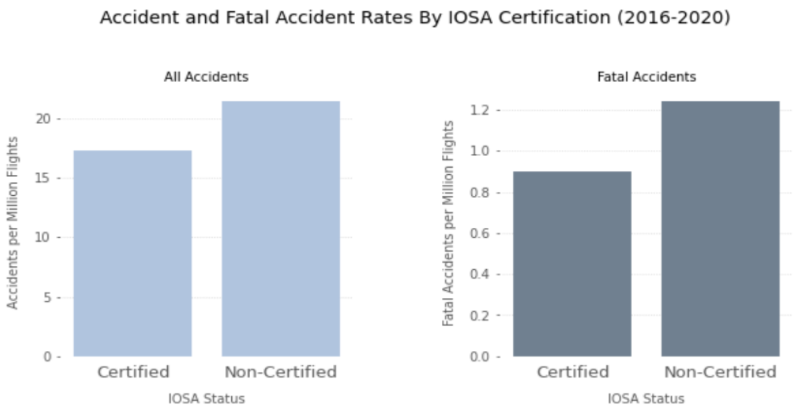
Fatal accidents involve a mix of smaller domestic and charter airlines and some state airlines operating internationally. Two airlines were excluded from the chart with very high accident rates due to exceptionally low annual flight miles (note 1).

5.10.2 In Fig. 12, horizontal side-by-side bar charts accommodate the operator names and enable comparison between groups. The highest ten are inverted to draw the eye to the top of the chart showing the worst airline and allow the eye to easily move right to the best airline. The use of lines instead of bars reduces the data/ink ratio and the design is also intended to represent planes in flight.

5.11 Are Accident Rates Lower for Safety Certified Operators?

5.11.1 The IOSA is a non-mandatory international management and control audit. IOSA certified airlines were reported to have a crash rate three times less than those airlines not on the IOSA registry, although many smaller airlines do not participate due to cost [32]. Figure 14 compares the proportion of operators with and without certification against the number of accidents in the last 5 years.

Figure 14 – Accident Rates By IOSA certification



Source: ICAO Statistics 2020

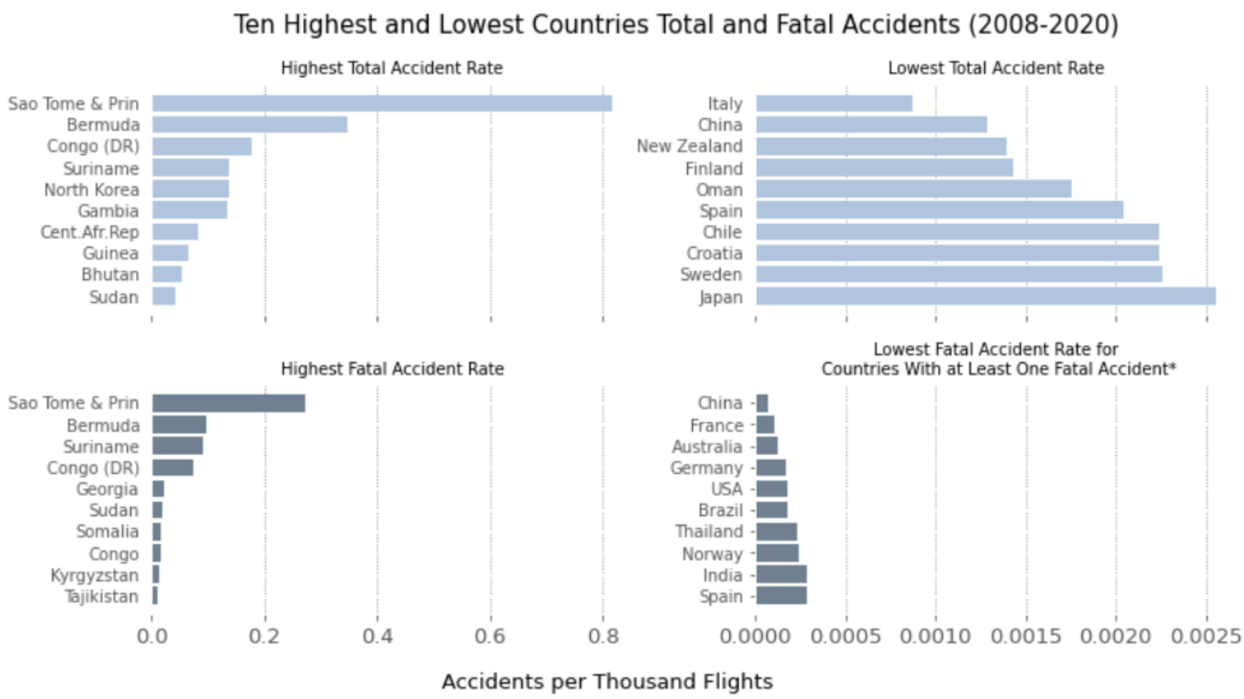
Accident rates are lower for certified operators.

5.11.2 Side by side bar charts are the simplest way to compare the two categories split between those certified and those not. Different scales on a single chart would require dual axes or a log scale, which can be confusing. Colour separates the two categories and gridlines are faded to the background to reduce clutter.

5.12 Which State of Registration has the most Accidents?

5.12.1 The convention on International Civil Aviation requires that all aircraft be registered with a national aviation authority which sets the procedures and standards for operation of those aircraft. The total and fatal accident rates by registering country states (top and bottom ten) per 1000 flights is shown in Figure 15.

Figure 15 – Total and Fatal Accident Rates By State of Registration



Source: ICAO Statistics 2020
*Note: Of the 193 countries reporting, 72 had no fatal accidents

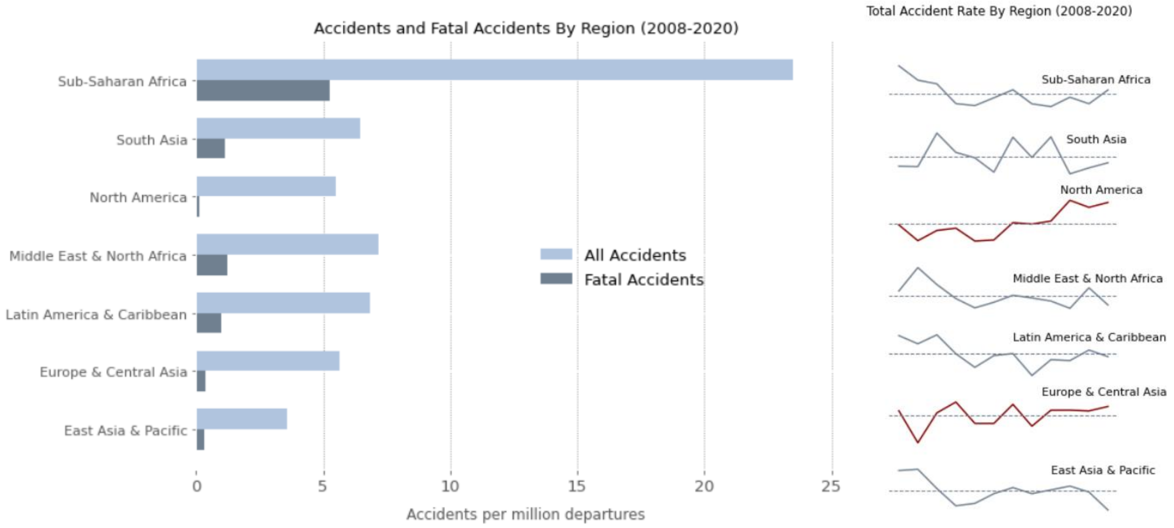
Most of the states with the worst accident rates are less-developed countries. In comparison, states with the lowest total and fatal accident rates, where at least one accident was recorded, are mainly developed countries. However, there may be many factors contributing to these differences including the respective age and models in the fleet, levels of maintenance, quality, and amount of training of crew, airport, and landing facilities.

5.12.2 A two-by-two matrix allows all the relevant comparisons to be made without plotting four separate charts. Shared X-axes, and colours clearly split the fatal from total accidents. Charts for the highest countries in each category are reversed to draw the eye to the top of the chart and allow natural eye-movement to the lowest accident rates.

5.13 Where do Accidents Occur?

5.13.1 Figure 16 summarises the number of accidents by region of occurrence expressed per million flights in total and over the period.

Figure 16 – Accidents By Region of Occurrence (2008-2020)



Source: ICAO Statistics 2020 and World Bank
Note1: Trend for all accidents is shown as fatal accident data not available for all years.

Sub-Saharan Africa has a significantly worse total and fatal rate to the rest of the regions, although it has trended downwards. This compares to North America and Europe & Central Asia where total accidents occurring within their borders have increased.

5.13.2 A clustered horizontal bar plot shows total and fatal accident rates by region and accommodates the longer names. The bars use the pre-attentive attribute of form to allow a quick assessment of the worst and best region in each category. The legend is ordered the same as the bars and placed close to the data for easy reference. A series of small multiples in the form of sparklines alongside the plot shows the general trend for each region since a chart showing all the regions would be overcrowded [33]. Sparklines are in the same order as the bar chart but labelled to avoid confusion and colour is used to highlight worsening regions.

5.14 Relationship Between Accident Rate, Fleet Age and Size and Income Band

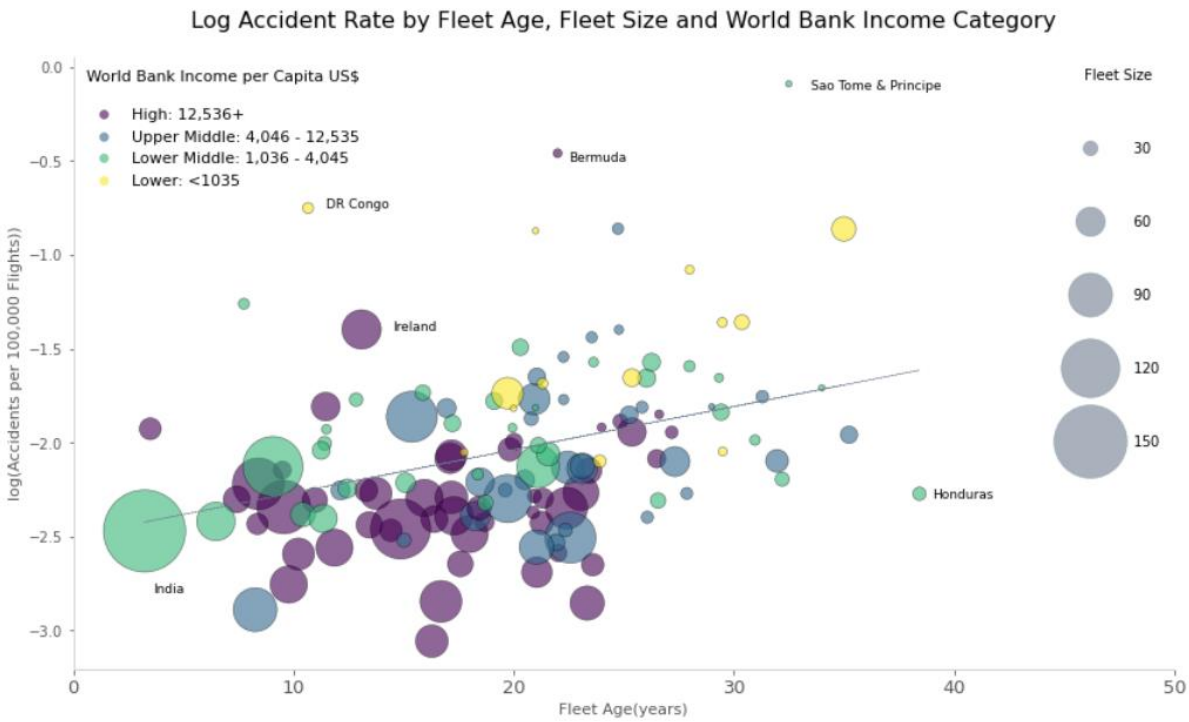
5.14.1 Older planes are more likely to be less airworthy [34] and therefore more likely to be involved in accidents. Table 2 summarises the fleet ages and sizes for states with the worst and best accident rates.

Table 2 – Accident Rates, Fleet Ages and Sizes forTen Worst and Best States of Registration

Accident Rate Group	Average Fleet Age	Average Fleet Size
Worst Ten States	23.6	4
Best Ten States	16.3	29
Worst Ten Fatal States	25.0	3
Best Ten Fatal States	15.0	46

5.14.2 The ten worst states have an average fleet age 45% higher than the best ten and far smaller fleet sizes. The difference is even more noticeable for fatal accidents. These states are predominantly developing countries (Fig. 15) suggesting a relationship between GDP and accident rates. To examine this further, the data was split into the four World Bank Income per Capita bands - Upper, Upper Middle, Lower Middle and Lower and Figure 17 compares fleet age, accident rate and Income per capita by country.

Figure 17 – Log Accident Rate By Fleet Age and Size and Income per Capita (2008-2020)



Source: ICAO Statistics 2020 and World Bank
Note1: A log scale is used for accident rates as the data points are far apart and this helps to visualise the points on a single axis.
Note2: Some bubbles of interest are annotated, including India, a lower-middle income country with a large and young fleet and low accident rate, Ireland, a high-income country with a young fleet and relatively high accident rate and Honduras with the oldest fleet but not the highest accident rate.

There is a weak positive relationship overall between fleet age and the log accident rate shown by the trend line. In general, lower income countries tend to have smaller and older fleets and higher accident rates although some exceptions are annotated on the chart and included in the note. The relationship between the accident rate and fleet age for high income countries is weak or absent, as the bubbles are clustered horizontally, suggesting this factor has little or no influence on accidents. Other factors such as pilot error, weather or better *reporting standards may be more important. For low-income countries, running older fleets, mechanical failures or poor maintenance schedules may be more important issues.*

5.14.3 This chart was inspired by the Gap Minder Lifespan chart [35]. Each bubble allows four variables to be shown simultaneously and relies on the laws of proximity and similarity to group the data. World Bank categories are grouped by colour, using the Viridis colour scheme which is designed to be colour-blind friendly [36]. Bubble size indicates fleet size and bubbles are increased by a constant and edged in black to enable smaller income countries to be seen. Transparency is also used due to some overplotting of bubbles. The trend line shows the relationship of the x and y variables, faded so as not to dominate the chart and some of the more interesting outliers are annotated.

6.0 Conclusions

6.1 In terms of total air accidents per million departures, 2020 was the safest year over the period. The number of fatal accidents were also the third lowest for the period and virtually the same as 2019. Actual fatalities whilst higher than 2019, were still below the period average. Large fatal accidents are rare events with the majority of fatal incidents involving 50 or less deaths. These smaller fatal accidents have been trending downwards – possibly due to increased safety standards - and were lower in 2020 compared to 2019. In light of this 2020 does not appear to be a less safe year for air travel.

6.2 The relationship between air accidents and a range of factors was examined and findings are summarised in Table 3. Results should be interpreted with care, for example turboprops and smaller aircraft are associated with the highest accident rates, but this could reflect greater use of these planes in developing regions. Likewise, worsening trends for accidents occurring in North America, a region with high reporting standards, may simply reflect increasing traffic from country operators with poor safety records. Further analysis of inter-relationships between factors is required before drawing firm conclusions.

Table 3 – Profile of Factors Associated with the Highest Accident Rates

Factors	All Accidents	Fatal Accidents
Worst flying month	February	August
Worst flying day	Tuesday	Wednesday
Size of aircraft	<25 seats	<25 seats
Model	Fairchild Metroliner	Let-410
Engine Type	Turboprop	Turboprop
Flight Phase	Landing	Approach
Worst Operator	Buffalo Airways (Canada)	Air Sunshine (USA)
IOSA Certification	No certification	No certification
Worst Country of Registration	Sao Tome & Principe	Sao Tome & Principe
Worst Location By Region	Sub-Saharan Africa	Sub-Saharan Africa
Worst Trending Region	North America	n/a
Worst Ten Operators Average Fleet Age	24 years	25 years
Worst Ten Operators Average Fleet Size	4	3
World Bank Income per Capita Band	Low	Low

6.3 There is a weak positive relationship between fleet age, size, income category and the accident rate in relation to poorer countries but the relationship is not so clear for high-income countries. It is likely that the picture is complicated involving a wide range of contributing factors including standards of pilot training, maintenance schedules, safety and reporting standards, flying, and landing conditions, weather conditions and whether a country is wartorn to explain the differences in accident rates. However, this report has highlighted some interesting insights to inform the building of a predictive model.

Word count excluding appendices, tables, headings, numbering, notes, sources, references, header, figure, table, and contents pages – 3997

References

[1] “Plane crash deaths rose in 2020 despite pandemic", *the Guardian*, 2021. [Online]. Available: https://www.theguardian.com/world/2021/jan/02/plane-crash-deaths-rose-in-2020-despite-pandemic. [Accessed: 23- Mar- 2021].

[2] N. Insights and E. Articles, "How aviation safety has improved", *AGCS Global*, 2021. [Online]. Available: https://www.agcs.allianz.com/news-and-insights/expert-risk-articles/how-aviation-safety-has-improved.html. [Accessed: 23- Mar- 2021].

[3] "The Most Common Causes of Aviation Accidents", *The National Law Review*, 2021. [Online]. Available:https://www.natlawreview.com/article/most-common-causes-aviation-accidents#:~:text=Pilot%20error%20refers%20to%20an,can%20contribute%20to%20pilot%20error. [Accessed: 16- Apr- 2021].

[4] "Malaysia Airlines losses worsen on MH370 disappearance", *BBC News*, 2021. [Online]. Available: https://www.bbc.co.uk/news/business-27435455. [Accessed: 23- Mar- 2021].

[5] "Investigation: AO-2008-070 - In-flight upset - Airbus A330-303, VH-QPA, 154 km west of Learmonth, WA, 7 October 2008", *Atsb.gov.au*, 2021. [Online]. Available: https://www.atsb.gov.au/publications/investigation_reports/2008/aair/ao-2008-070/. [Accessed: 16- Apr- 2021].

[6] "API Data Service", *Icao.int*, 2021. [Online]. Available: https://www.icao.int/safety/iStars/Pages/API-Data-Service-new.aspx. [Accessed: 16- Apr- 2021].

[7] Annex 13 - Aircraft Accident And Incident Investigation 12th Edition, July 2020", International Civil Aviation Organisation, 2020.

[8] "Air transport, passengers carried | Data", *Data.worldbank.org*, 2021. [Online]. Available: https://data.worldbank.org/indicator/IS.AIR.PSGR. [Accessed: 23- Mar- 2021].

[10] "Live Flight Tracker - Real-Time Flight Tracker Map | Flightradar24", *Flightradar24*, 2021. [Online]. Available: https://www.flightradar24.com/data/statistics. [Accessed: 16- Apr- 2021].

[11] “Airplane Types", *The Travel Insider*, 2021. [Online] Available: https://blog.thetravelinsider.info/airplane-types [Accessed: 09- Apr- 2021].

[12] *Dvbbank.com*, 2021. [Online]. Available: https://www.dvbbank.com/~/media/Files/D/dvbbank-corp/aviation/dvb-overview-of-commercial-aircraft-2018-2019.pdf. [Accessed: 16- Apr- 2021].

[13] *Airfleets.net,* 2021 [Online]. Available: https://www.airfleets.net/recherche/country.htm. [Accessed: Accessed: 16- Apr- 2021].

[14] "GDP per capita (current US$) | Data", *Data.worldbank.org*, 2021. [Online]. Available: https://data.worldbank.org/indicator/NY.GDP.PCAP.CD. [Accessed: 16- Apr- 2021].

[15] "World Bank Country and Lending Groups – World Bank Data Help Desk", *Datahelpdesk.worldbank.org*, 2021. [Online]. Available: https://datahelpdesk.worldbank.org/knowledgebase/articles/906519-world-bank-country-and-lending-groups. [Accessed: 16- Apr- 2021].

[16] A. Harro Ranter, "Aviation Safety Network >", *Aviation-safety.net*, 2021. [Online]. Available: https://aviation-safety.net/. [Accessed: 08- Apr- 2021].

[17] "The Aviation Herald", *Avherald.com*, 2021. [Online]. Available: https://avherald.com/. [Accessed: 08- Apr- 2021].

[18] A. KIRK, *DATA VISUALISATION*. [S.l.]: SAGE PUBLICATIONS, 2019.

[19] H. Laws and W. Craig, "How to Make Data Visualization Better with Gestalt Laws", *WebFX Blog*, 2021. [Online]. Available: https://www.webfx.com/blog/web-design/data-visualization-gestalt-laws/. [Accessed: 22- Apr- 2021].

[20] S. Few, *Show me the numbers*. Burlingame, Calif.: Analytics Press, 2012.

[21] C. Ware, *Information Visualization, 3rd Edition*.

[22] "Tufte's Principles | thedoublethink", *thedoublethink*, 2021. [Online]. Available: https://thedoublethink.com/tuftes-principles-for-visualizing-quantitative-information/#:~:text=Tufte%20claims%20that%20good%20graphical,without%20loss%20of%20data%2Dinformation. [Accessed: 23- Apr- 2021].

[23] D. Nichols, "Coloring for Colour-blindness'", *Davidmathlogic.com*, 2021. [Online]. Available: https://davidmathlogic.com/colorblind/#%23D81B60-%231E88E5-%23FFC107-%23004D40-%239DBCD4. [Accessed: 16- Apr- 2021].

[24] "Future of Aviation", *Icao.int*, 2021. [Online]. Available: https://www.icao.int/Meetings/FutureOfAviation/Pages/default.aspx. [Accessed: 23- Mar- 2021].

[25] "Edward Tufte forum: baseline for amount scale", *Edwardtufte.com*, 2021. [Online]. Available: https://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=00003q. [Accessed: 18- Apr- 2021].

[26] O. Digital, "Does the axis have to start at zero? (Part 1 – line charts) | ONS Digital", Digitalblog.ons.gov.uk, 2021. [Online]. Available: https://digitalblog.ons.gov.uk/https://digitalblog.ons.gov.uk/2016/06/27/does-the-axis-have-to-start-at-zero-part-1-line-charts/#/#:~:text=Data%20in%20a%20line%20chart,that%20they,that%20they%20are%20always%20misleading. [Accessed: 05- May- 2021].

[27] "Ten air travel tips that could save your life", *Motley Rice*, 2021. [Online]. Available: https://www.motleyrice.com/blogpost/ten-air-travel-tips-could-save-your-life. [Accessed: 24- Mar- 2021].

[28] "It's Official, August 9th Is The Busiest Air Travel Day Of The Year...", *God Save The Points*, 2021. [Online]. Available: https://www.godsavethepoints.com/busiest-travel-day-of-the-year-august-9th/. [Accessed: 09- Apr- 2021].

[29] "This is the worst day of the week to fly on this summer", *The Independent*, 2021. [Online]. Available: https://www.independent.co.uk/travel/news-and-advice/airports-busiest-day-summer-2017-heathrow-gatwick-manchester-luton-birmingham-edinburgh-aberdeen-a7843536.html. [Accessed: 09- Apr- 2021].

[30] "What Are The Busiest & Quietest Days In The Week To Fly? - Simple Flying", *Simple Flying*, 2021. [Online]. Available: https://simpleflying.com/best-days-to-fly/. [Accessed: 17- Apr- 2021].

[31] "Turboprop vs Jet: Learn Their Advantages and Disadvantages", *PilotMall.com*, 2021. [Online]. Available: https://www.pilotmall.com/blogs/news/turboprop-vs-jet-learn-their-advantages-and-disadvantages. [Accessed: 09- Apr- 2021].

[32] "Frequently Asked Questions about USOAP", *Icao.int*, 2021. [Online]. Available: https://www.icao.int/safety/CMAForum/Pages/FAQ.aspx#:~:text=What%20is%20a%20USOAP%20Audit,material%2C%20and%20best%20safety%20practices. [Accessed: 23- Apr- 2021].

[33] E. Tufte, *Beautiful Evidence*. Cheshire: Graphics Press, 2006, p46-63.

[34] "Ageing Aircraft - Structural Failure - SKYbrary Aviation Safety", *Skybrary.aero*, 2021. [Online]. Available: https://www.skybrary.aero/index.php/Ageing_Aircraft_-_Structural_Failure. [Accessed: 17- Apr- 2021].

[35] "World Health Chart | Gapminder", *Gapminder.org*, 2021. [Online]. Available: https://www.gapminder.org/fw/world-health-chart/. [Accessed: 27- Apr- 2021].

[36]]N. Bob Rudis, "Introduction to the viridis color maps", *Cran.r-project.org*, 2021. [Online]. Available: https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html. [Accessed: 27- Apr- 2021].

Appendix A – Source Code

```
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.transforms as transforms
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D

# Load files
data = pd.read_excel('AccidentsData.xlsx', sheet_name = "PassengerNumbers", engine='openpyxl')
data_departs = pd.read_excel("AccidentsData.xlsx",sheet_name = "Departures",engine = 'openpyxl')
data_accident = pd.read_excel("AccidentsData.xlsx",sheet_name = "AccidentsClean",engine =
'openpyxl')
monthly_passengers = pd.read_excel("AccidentsData.xlsx",sheet_name =
"MonthlyPassengers",engine = 'openpyxl')
flights_day = pd.read_excel("AccidentsData.xlsx",sheet_name = "FlightsByDay",engine = 'openpyxl')
operator_flights = pd.read_excel("AccidentsData.xlsx",sheet_name = "OperatorRisk",engine =
'openpyxl')
deps = pd.read_excel("AccidentsData.xlsx",sheet_name = "WBDepsReg",index_col = 0,engine =
'openpyxl')
gdp_cap = pd.read_excel("AccidentsData.xlsx",sheet_name = "GDP per Capita",engine = 'openpyxl')
income_cat = pd.read_excel("AccidentsData.xlsx",sheet_name = "WorldBankIncome",engine =
'openpyxl')
fleet_age = pd.read_excel("AccidentsData.xlsx",sheet_name = "OperatorRisk",engine = 'openpyxl')


# GLOBAL PASSENGERS

# Create data sets of interest
df1 = data.drop(columns =
['1960',"1961","1962","1963","1964","1965","1966","1967","1968","1969","2020"],axis = 1)
world = df1[df1["Country Name"] == "World"]
LA = df1[df1["Country Name"] == "Latin America & Caribbean"]
EA = df1[df1["Country Name"] == "East Asia & Pacific"]
E =  df1[df1["Country Name"] == "Europe & Central Asia"]
ME = df1[df1["Country Name"] == "Middle East & North Africa"]
NA = df1[df1["Country Name"] == "North America"]
SA = df1[df1["Country Name"] == "South Asia"]
SS = df1[df1["Country Name"] == "Sub-Saharan Africa"]

# Express as billions
def billions(df):
   df_new = pd.DataFrame(df.iloc[:,4:].sum(axis=0)/1000000000,
                   columns = ["Passengers (billions)"])
   return df_new

# Run function
world_pass = billions(world)
LA_pass = billions(LA)
EA_pass = billions(EA)
E_pass = billions(E)
```

```
ME_pass = billions(ME)
NA_pass = billions(NA)
SA_pass = billions(SA)
SS_pass = billions(SS)

# Combine into one dataframe
combined_pass = pd.concat([LA_pass,EA_pass,E_pass,ME_pass,NA_pass,SA_pass,SS_pass],axis = 1)
combined_pass.columns = ["Latin America & Carribean","East Asia & Pacific","Europe & Central
Asia", "Middle East & North Africa",
                "North America", "South Asia", "Sub-Saharan Africa"]
```

 # Figure 1 - Air Passenger Numbers By Region (1970-2020)

```
def global_pass():

   # Set variables
   X = np.arange(1970,2020)
   labels = ["East Asia & Pacific","Europe & Central Asia","North America","Rest of World"]
   eap = combined_pass["East Asia & Pacific"]
   nam = combined_pass["North America"]
   eca = combined_pass["Europe & Central Asia"]
   row = combined_pass["Latin America & Carribean"] + combined_pass["Middle East & North
Africa"] + combined_pass["South Asia"] + combined_pass["Sub-Saharan Africa"]

   # Plot chart
   fig, ax = plt.subplots(figsize = (8,4))
   pal = ["xkcd:darkblue",'xkcd:deep sea blue','steelblue','lightsteelblue']
   ax.yaxis.grid(True,linestyle = ":",color = "grey", alpha = 0.5)
   ax.stackplot(X,eap,eca,nam,row, labels = labels, colors = pal)
   ax.set_facecolor(color = "white")

   # Titles, axes and annotations
   ax.set_title("Global Air Passengers (1970 to 2019)")
   ttl = ax.title
   ttl.set_position([.5, 1.5])
   handles, labels = ax.get_legend_handles_labels()
   ax.legend(reversed(handles), reversed(labels),loc = "upper center",facecolor = "white",
          frameon = False, prop = {'size':12}, bbox_to_anchor = (0.5,0.96))
   ax.set_xlim(1970,2019)
   yticks = ax.yaxis.get_major_ticks()
   yticks[0].label1.set_visible(False)
   ax.set_xlabel("Year",fontsize = 12)
   ax.set_ylabel("Passengers (billions)",fontsize = 12)

   plt.show()

# DEPARTURES

# Set departures to millions
departs = data_departs.groupby("Year")["Departures"].sum()/1000000
```

 # Figure 2 - Global Departures by Year (2008-2020)

```
def departures():

    # Set variables
    X = np.arange(2008,2021)

    # Plot chart
    plt.style.use('ggplot')
    fig, ax = plt.subplots(figsize = (8,4))
    ax.set_facecolor(color = "white")
    ax.yaxis.grid(True,linestyle = ":",color = "grey", alpha = 0.5)
    ax.plot(X,departs, color = "gray",linewidth = 2)

    # Titles, axes and annotations
    ax.set_title("Flight Departures (2008-2020)")
    ax.set_ylim(15,45)
    ax.set_xlabel("Year",fontsize = 11,labelpad = 15)
    ax.set_ylabel("Number of flights (millions)",fontsize = 11)
    ax.text(2020.15,22,"22 mn")
    ax.text(2019.1,38,"38 mn")
    ax.get_yticklabels()[0].set_fontweight("bold")
    ax.get_yticklabels()[0].set_color('xkcd:deep sea blue')

    plt.show()

 #   ACCIDENT DATA

# Set year as string and fill na with zero
data_accident.Year = data_accident.Year.astype('object')
data_accident["Fatalities"].fillna(0,inplace = True)

# Filter to get commercial large plane data over 5.7 tonnes
commercial = data_accident[(data_accident["Airplane"] == 1.0) &
                  (data_accident["Over5700"] == 1.0) &
                  (data_accident["Operator"] != "private") & (data_accident["ScheduledCommercial"]
== 1.0)]

# Get incidents numbers and number of fatalities an put into dataframe
all_accidents = commercial.groupby("Year")["Date"].count()
fatal_accidents =commercial[commercial["Fatalities"] >0].groupby("Year")["Date"].count()
fatalities = commercial[commercial["Fatalities"]>0].groupby("Year")["Fatalities"].sum()
all_accidents_df = pd.concat([all_accidents,fatal_accidents,fatalities],axis = 1)
all_accidents_df.columns = ["Number Accidents","Accidents With Fatalities","Number Fatalities"]
all_accidents_df['Accidents No Fatalities'] = all_accidents_df['Number Accidents'] -
all_accidents_df['Accidents With Fatalities']

# Group departures by year
departures_year = data_departs.groupby("Year",as_index = False)["Departures"].agg("sum")
departures_year.set_index('Year',inplace = True)

# Concat the departures to the incidents dataframe and express each accident measure over the
same base of number of departures
accidents_per_departures = pd.concat([all_accidents_df,departures_year],axis = 1)
```

```
accidents_per_departures["Accidents_per_departures"] = accidents_per_departures["Number Accidents"]/accidents_per_departures["Departures"]*1000000
accidents_per_departures["Fatal_per_departures"] = accidents_per_departures["Accidents With Fatalities"]/accidents_per_departures["Departures"]*1000000
accidents_per_departures["Non_fatal_per_departures"] = accidents_per_departures["Accidents No Fatalities"]/accidents_per_departures["Departures"]*1000000
accidents_per_departures["Fatalities_per_departures"] = accidents_per_departures["Number Fatalities"]/accidents_per_departures["Departures"]*1000000
accidents_per_departures["% Fatal Accidents"] = round(accidents_per_departures["Accidents With Fatalities"]/accidents_per_departures["Number Accidents"]*100,0)
```

 # Figure 3 - Air Accident Trends (2008-2020)

```
def accidents():

    # Set variables
    X = accidents_per_departures.index
    a = accidents_per_departures["Accidents_per_departures"]
    b = accidents_per_departures["Non_fatal_per_departures"]
    c = accidents_per_departures["Fatal_per_departures"]

    # Plot Chart
    fig, ax = plt.subplots(figsize = (10,6))
    ax.plot(X,a, linewidth = 2,color = "xkcd:prussian blue")
    ax.plot(X,b, linewidth = 2, color = "slategrey")
    ax.plot(X,c, linewidth = 2, color = "firebrick")

    # Titles and Axes
    ax.set_title("Number of Accidents Per Million Flight Departures (2008 to 2020)", fontsize=15)
    ax.yaxis.grid(True,linestyle = ":",color = "grey")
    ax.set_xlabel("Year",fontsize = 13,labelpad = 15)
    ax.set_xlim(2008,2021)
    ax.set_yticks(np.arange(0, 9,1))
    ax.set_ylim(0,9)
    xticks = ax.xaxis.get_major_ticks()
    yticks = ax.yaxis.get_major_ticks()
    xticks[7].set_visible(False)
    yticks[0].label1.set_visible(False)
    ax.set_facecolor(color = "white")
    top = ax.spines["top"]
    right_side = ax.spines["right"]
    left_side = ax.spines["left"]
    right_side.set_visible(False)
    left_side.set_visible(False)
    top.set_visible(False)

    # Annotations
    plt.text(2020.10,0.3,"Fatal",fontsize = 12,color = "firebrick")
    plt.text(2020.10,3.1,"Non-Fatal",fontsize = 12,color = "slategrey")
    plt.text(2020.10,3.7,"All accidents",fontsize = 12,color = "xkcd:prussian blue")
    ax.set_ylabel("Accidents Per Million Departures",fontsize = 11)
    plt.show()
```

 # Figure 4 - Number of Fatalities by Year (2008-2020)

```
def fatalities():
   # Set variables
   X = all_accidents.index
   d = accidents_per_departures["Fatalities_per_departures"]
   d_mean = d.mean() # median and mean very similar
   color = ["lightsteelblue","lightsteelblue","lightsteelblue","lightsteelblue",
         "lightsteelblue","lightsteelblue","firebrick","lightsteelblue","lightsteelblue",
         "lightsteelblue","lightsteelblue","lightsteelblue","lightsteelblue"]

   # Plot chart
   fig, ax = plt.subplots(figsize = (10,6))
   ax.bar(X,d, color = color)
   ax.set_title("Number of Fatalities Per Million Departures (2008-2020)", fontsize=15)
   ax.set_facecolor(color = "white")
   ax.set_xlabel("Year",fontsize = 13,labelpad = 15)

   # Titles, Axes and Annotations
   ax.text(2019.75,16,"Average",fontsize = 10)
   ax.set_ylim(0,35)
   ax.axhline(y=d_mean, linewidth=1, linestyle = "--", color='slategrey')
   plt.xticks(accidents_per_departures.index,fontsize = 10)
   ax.set_ylabel("Number of Fatalities per Million Departures",fontsize = 10)
   plt.tick_params(axis='x', which='both', bottom=False, top=False, labelbottom=True)

   plt.show()

# Create dataframes where the fatalities were greater than 50 deaths and less than or equal to 50
deaths
above_50_fatalities = commercial[commercial["Fatalities"]>50]
below_50_fatalities = commercial[(commercial["Fatalities"]<=50) & (commercial["Fatalities"]>0)]

# Groupby year and concat to new dataframe - add in 2013 & 2017 > 50 deaths which was missed
from the groupby as zero
above_50_df = above_50_fatalities.groupby("Year",as_index = False)["Fatalities"].count()
new_row1 = {'Year':2017,'Fatalities':0}
new_row2 = {'Year':2013,'Fatalities':0}
above_50_df = above_50_df.append(new_row1,ignore_index = True).sort_values(by = "Year")
above_50_df = above_50_df.append(new_row2,ignore_index = True).sort_values(by = "Year")
above_50_df.set_index("Year",inplace = True)
below_50_df = below_50_fatalities.groupby("Year",as_index = False)["Fatalities"].count()
below_50_df.set_index("Year",inplace = True)
combined_50 = pd.concat([below_50_df,above_50_df,accidents_per_departures["Departures"]],axis
= 1)
combined_50.columns = ["<=50 deaths",">50 deaths","Departures"]
combined_50["<=50 per depart"] = combined_50["<=50
deaths"]/combined_50["Departures"]*1000000
combined_50[">50 per depart"] = combined_50[">50
deaths"]/combined_50["Departures"]*1000000
combined_50.columns = ["<=50 deaths",">50 deaths","Departures","<=50 per depart",">50 per
depart"]
```

```
combined_50["Total Number Fatal"] = combined_50["<=50 deaths"] + combined_50[">50 deaths"]
```

 # Figure 5 - Fatal Accidents Involving More or Less Than 50 Fatalities (2008-2020)

```
def above_50():
   # Set variables
   X = combined_50.index
   f = combined_50["<=50 per depart"]
   g = combined_50[">50 per depart"]
   h = f+g
   width = 0.75
   z = np.polyfit(X,h,1)
   p = np.poly1d(z)

   # Plot chart
   fig, ax = plt.subplots(figsize = (11,6))
   ax.set_facecolor(color = "white")
   ax.bar(X,f,width, color = "slategrey")
   ax.bar(X,g,width,color = "lightsteelblue",bottom = f)
   ax.plot(X,p(X),"black",linestyle = ":")

   # Titles and Axes
   ax.set_ylabel("Number of Fatalities per Million Departures", fontsize = 9)
   ax.set_xlabel("Year",fontsize = 13,labelpad = 15)
   ax.set_xticks(X)
   plt.tick_params(axis='x', which='both', bottom=False, top=False, labelbottom=True)
   ax.set_ylim(0,1.5)
   ax.set_title("Number of Fatal Accidents Over and Under 50 Deaths\n expressed per Million
Departures (2008-2020)",fontsize = 13)
   custom_lines = [Line2D([0], [1], color="lightsteelblue", lw=10),
             Line2D([0], [1], color="slategrey", lw=10)]

   ax.legend(custom_lines, ["Accidents of More Than 50 Fatalities","Accidents of 50 or Less
Fatalities"],
          facecolor = "white",frameon = False,bbox_to_anchor=(1,0.80),fontsize = 12)

   plt.show()


# WHEN DO ACCIDENTS OCCUR?

# Get average monthly passenger totals
monthly_passenger_totals = pd.DataFrame(monthly_passengers.groupby("Month",as_index =
False)["Total_OS"].sum()).set_index("Month")
monthly_passenger_totals.columns = ["Av_passengers"]
monthly_passenger_totals["Av_passengers(million)"] =
monthly_passenger_totals["Av_passengers"]/1000000
pd.options.mode.chained_assignment = None

# Tidy data and extract months and days from the timestamp column for accidents
commercial.Date = commercial.Date.apply(lambda x: x.replace('"', ''))
commercial["Date"] = commercial["Date"].apply(lambda x: pd.Timestamp(x))
commercial["Month"] = commercial["Date"].apply(lambda x: x.month)
```

```
commercial["Dayofweek"] = commercial["Date"].apply(lambda x: x.dayofweek)

# Create monthly dataframes for total and fatal accidents
month = pd.DataFrame(commercial.groupby("Month").count())
fatal_month =
pd.DataFrame(commercial[commercial["Fatalities"]>0].groupby("Month")["Date"].count())
```

 # Figure 6 - Total and Fatal Accidents by Month

```
def mon():
  # Set variables
   x = np.arange(len(month.index))
   labels = ['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']
   m = month["Date"]/monthly_passenger_totals["Av_passengers(million)"]
   fm = fatal_month["Date"]/monthly_passenger_totals["Av_passengers(million)"]

   # Plot chart
   fig, (ax1,ax2) = plt.subplots(2,sharex = True,figsize = (10,6))
   fig.suptitle("Total and Fatal Accidents Per Million Passengers By Month (2008 - 2020)", fontsize =
12)

   # Plot first chart
   ax1.bar(x,m, label='Total Accidents',color = "lightsteelblue")
   ax1.set_facecolor("white")
   ax1.set_ylim(0,100)
   ax1.xaxis.grid(False)
   ax1.yaxis.grid(True,linestyle = ":",color = "grey")
   ax1.legend(loc = "upper right",fontsize = 13,facecolor = "white",frameon =
False,bbox_to_anchor=(1,1.05))
   ax1.tick_params(axis='x',bottom = False)

   # Plot second chart
   ax2.bar(x,fm, label='Fatal Accidents',color = "slategrey")
   ax2.set_facecolor("white")
   ax2.set_yticks(np.arange(0, 10,2))
   ax2.set_ylim(0,10)
   ax2.xaxis.grid(False)
   ax2.yaxis.grid(True,linestyle = ":",color = "grey")
   ax2.set_xticks(x)
   plt.tick_params(axis='x', which='both', bottom=False, top=False, labelbottom=True)
   ax2.set_xticklabels(labels)
   ax2.legend(loc = "upper right",fontsize = 13,facecolor = "white",frameon =
False,bbox_to_anchor=(1,1.05))
   ax2.set_xlabel("Month",fontsize = 11,labelpad = 15)
   fig.text(0.06, 0.5, 'Number of Accidents per Million Passengers', va='center', rotation='vertical')

   plt.show()

# Get the flight data for all accidents and fatal accidents by day of the week
flights = pd.DataFrame(flights_day.groupby("Day")["2019 Number of flights"].mean()).reset_index()
flights = pd.DataFrame(flights_day.groupby("Day")["2019 Number of flights"].mean()).reset_index()
flights.columns = ["Day_name","Flights 2019"]

# Label with daynames, get counts by day and merge with df of number of flights
```

```
day_name = ["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"]
day = pd.DataFrame(commercial.Dayofweek.value_counts()).reset_index()
day.columns = ["Day_name","Accidents"]
day.Day_name = day_name
day_merged = day.merge(flights, on = "Day_name")
day_merged["Acc_per_000_flights"] = (day_merged["Accidents"]/day_merged["Flights 2019"])*1000
day_name = ["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"]

# Repeat above for fatal accidents
day_fatal =
pd.DataFrame(commercial[commercial["Fatalities"]>0].groupby("Dayofweek")["Date"].count()).reset
_index()
day_fatal["Day_name"] = day_name
day_fatal.drop(columns = "Dayofweek",inplace = True)
day_fatal.columns = ["Fatal Accidents","Day_name"]
day_fatal_merged = day_fatal.merge(flights, on = "Day_name")
day_fatal_merged["Fatal_per_000_flights"] = (day_fatal_merged["Fatal
Accidents"]/day_merged["Flights 2019"])*1000
```

#Figure 7 - Total and Fatal Accidents by Weekday

```
def day():
   # Set variables
   x = [0,1,2,3,4,5,6]
   labels = ['Mon','Tues','Wed','Thur','Fri','Sat','Sun']
   fig, (ax1,ax2) = plt.subplots(2,sharex = True,figsize = (10,6))
   fig.suptitle("Total and Fatal Accidents Per 1000 Flights By Day of the Week (2008-2020)", fontsize =
12)
   plt.subplots_adjust(left=None, bottom=None, right=None, top=None, wspace=None, hspace=0.3)
   d = day_merged["Acc_per_000_flights"]
   fd = day_fatal_merged["Fatal_per_000_flights"]

   # Plot first chart
   ax1.bar(x,d, label='Total Accidents',color = "lightsteelblue")
   ax1.set_facecolor("white")
   ax1.set_yticks(np.arange(0, 4,1))
   ax1.set_ylim(0,4)
   ax1.xaxis.grid(False)
   ax1.yaxis.grid(True,linestyle = ":",color = "grey")
   ax1.legend(loc = "upper right",fontsize = 13,facecolor = "white",frameon = False,bbox_to_anchor =
(1,1.05))
   ax1.tick_params(axis='x',bottom = False)

   # Plot second chart
   ax2.bar(x,fd, label='Fatal Accidents',color = "slategrey")
   ax2.set_facecolor("white")
   ax2.set_yticks(np.arange(0,0.5,0.2))
   ax2.set_ylim(0,0.5)
   ax2.yaxis.grid(True,linestyle = ":",color = "grey")
   ax2.set_xticks(x)
   plt.tick_params(axis='x', which='both', bottom=False, top=False, labelbottom=True)
   ax2.set_xticklabels(labels)
   ax2.legend(loc = "upper right",fontsize = 13,facecolor = "white",frameon = False,bbox_to_anchor =
(1,1.05))
```

```
    ax2.set_xlabel("Day",fontsize = 11, labelpad = 15)
    fig.text(0.06, 0.5, 'Number of Accidents per Million Passengers', va='center', rotation='vertical')

    plt.show()

# ACCIDENTS BY SIZE AND MODEL OF PLANE

# Dataframes of all accidents by size and model
plane_type = commercial[["Model","Size","EngineType","In Service","Year"]].reset_index()
planes_sort = pd.DataFrame(plane_type.groupby(["Model","In Service","Size"],as_index =
False)["index"].count().sort_values(by = "index",ascending = False))
planes_sort.columns = ["Model","In Service","Size","Accidents"]
planes_sort["Accidents/InService"] = planes_sort["Accidents"]/planes_sort["In Service"]
planes_sorted = planes_sort.sort_values(by = "Accidents/InService",ascending = False)

# Dataframes of fatal accidents by size and model
plane_type_fatal = commercial[commercial["Fatalities"]>0].reset_index()
planes_sort_fatal = pd.DataFrame(plane_type_fatal.groupby(["Model","In Service","Size"],as_index
= False)["index"].count().sort_values(by = "index",ascending = False))
planes_sort_fatal.columns = ["Model","In Service","Size","Accidents"]
planes_sort_fatal["Accidents/InService"] = planes_sort_fatal["Accidents"]/planes_sort_fatal["In
Service"]
planes_sorted_fatal = planes_sort_fatal.sort_values(by = "Accidents/InService",ascending = False)

# Size of plane by total and fatal accidents
size_count_fatal = pd.DataFrame(planes_sort_fatal.Size.value_counts())
seats = pd.Series(["<25","25-49","50-349","350+"])
size_count = pd.DataFrame(planes_sort.Size.value_counts())
size_count.set_index(seats,inplace = True)
```

 # Figure 8 - Total and Fatal Accidents by Size of Aircraft

```
def size():

    # Set variables
    labels1 = size_count.index[0:4]
    planes1 = size_count.Size[0:4]
    planes2 = size_count_fatal.Size[0:4]
    colors = ["slategrey","xkcd:steel blue","xkcd:light grey blue","grey"]
    explode = (0.05,0.05,0.05,0.05)

    # Plot chart
    fig,(ax1,ax2) = plt.subplots(1,2,figsize = (10,10))
    fig.suptitle("Total and Fatal Accidents By Plane Size", fontsize=18,y =0.88)
    ax1.pie(planes1, labels=['','','',''],autopct='%1.0f%%',startangle=45,colors =
colors,pctdistance=0.85, explode = explode,
        textprops={'fontsize': 13, 'color':"white",'weight':'heavy'})

    # Chart 1
    ax1.set_title("Total Accidents",fontsize = 14, y = 0.8)
    centre_circle = plt.Circle((0,0),0.70,fc='white')
    ax1.add_artist(centre_circle)
    ax1.axis('equal')
```

```
    legend1 = ax1.legend(loc = "center",labels = labels1,facecolor = "white",frameon = False,fontsize =
14,title = "Seats")
    plt.setp(legend1.get_title(),fontsize=14)
    plt.tight_layout()

    # Chart 2
    ax2.pie(planes2, labels=['','','',''],autopct='%1.0f%%',startangle=45,colors =
colors,pctdistance=0.85, explode = explode,
        textprops={'fontsize': 13,'color':'white','weight':'heavy'})
    ax2.set_title("Fatal Accidents",fontsize = 14,y = 0.8)
    centre_circle = plt.Circle((0,0),0.70,fc='white')
    ax2.add_artist(centre_circle)
    ax2.axis('equal')
    plt.tight_layout()

    plt.show()
```

 # Figure 9 - Total Accidents By Plane Model

```
def model():
    # Set variables
    X1 = planes_sorted.Model.head(10)
    y1 = planes_sorted["Accidents/InService"].head(10)
    X2 = planes_sorted_fatal.Model.head(10)
    y2 = planes_sorted_fatal["Accidents/InService"].head(10)
    colour_list1 =
["slategrey","slategrey","slategrey","lightsteelblue","slategrey","steelblue","steelblue","slategrey","s
lategrey"]
    colour_list2 =
["slategrey","slategrey","lightsteelblue","slategrey","slategrey","slategrey","slategrey","slategrey","s
lategrey","slategrey"]

    # Plot chart
    fig, (ax1,ax2) = plt.subplots(1,2,figsize = (12,4))
    fig.subplots_adjust(wspace = 1,top = 0.8)
    fig.text(0.45, -0.05, 'Accidents per Number of Aircraft in Service', ha = "center",fontsize = 12)
    fig.suptitle("Total and Fatal Accidents By Model of Plane",fontsize = 14)


    # Chart 1
    ax1.barh(X1,y1,color = colour_list1)
    ax1.tick_params(axis='y', which='both', left=False, right=False, labelleft=True)
    ax1.invert_yaxis()
    ax1.set_facecolor("white")
    ax1.set_title("Total Accidents",fontsize = 11)

    custom_lines = [Line2D([0], [1], color="slategrey", lw=8),
                Line2D([0], [1], color="steelblue", lw=8),
                Line2D([0], [1], color="lightsteelblue", lw=8)
                ]

    legend1 = ax1.legend(custom_lines, ['Very Small/Small','Medium','Large'],facecolor =
"white",frameon = False,
          bbox_to_anchor=(1.2,0.6),title = "Size of Aircraft", fontsize = 10)
```

```
    plt.setp(legend1.get_title(),fontsize=11)


    # Chart 2
    ax2.barh(X2,y2,color = colour_list2)
    ax2.tick_params(axis='y', which='both', left=False, right=False, labelleft=True)
    ax2.invert_yaxis()
    ax2.set_facecolor("white")
    ax2.set_title("Fatal Accidents",fontsize = 11)

        plt.show()


# ACCIDENTS BY ENGINE TYPE

# Engine types for all accidents and fatal accidents
engine_type = commercial[["Model","EngineType","Year","In Service","Region"]].reset_index()
engine_accidents = pd.DataFrame(engine_type.groupby(["EngineType","Model","Year","In
Service","Region"],as_index = False)["index"].count())
engine_accidents.columns = ["EngineType","Model","Year","In Service","Region","Accidents"]
engines = engine_accidents.sort_values(by = "Accidents",ascending = False)

engine_type_fatal = commercial[commercial["Fatalities"]>0]
engine_type_fatal = engine_type_fatal[["Model","EngineType","Year"]].reset_index()
engine_accidents_fatal =
pd.DataFrame(engine_type_fatal.groupby(["EngineType","Model","Year"],as_index =
False)["index"].count())
engine_accidents_fatal.columns = ["EngineType","Model","Year","Fatal"]
engines_fatal = engine_accidents_fatal.sort_values(by = "Fatal",ascending = False)

# By jet engines
jet_engines = pd.DataFrame(engines[engines["EngineType"] ==
"Jet"].groupby("Year")["Accidents"].sum()).reset_index()
jets_in_service = engine_accidents[engine_accidents["EngineType"] ==
"Jet"].groupby("Year",as_index = False)["In Service"].sum()
jet_engines= jet_engines.merge(jets_in_service)
jet_engines["Accidents/In Service/000"] = jet_engines["Accidents"]/jet_engines["In Service"]*1000

# jet engine fatal
jet_fatal = engines_fatal[engines_fatal["EngineType"] ==
"Jet"].groupby(["Year"])["Fatal"].sum().reset_index()
jet_fatal = jet_fatal.merge(jet_engines, on = "Year")
jet_fatal.drop(columns = ["Accidents","Accidents/In Service/000"],inplace = True)
jet_fatal["Fatal/In Service/000"] = jet_fatal["Fatal"]/jet_fatal["In Service"]*1000

# By turboprop engines
turbo_engines = pd.DataFrame(engines[engines["EngineType"] ==
"Turboprop"].groupby("Year")["Accidents"].sum()).reset_index()
turbo_in_service = engine_accidents[(engine_accidents["EngineType"] == "Turboprop") |
(engine_accidents["EngineType"] == "Piston")].groupby("Year",as_index = False)["In Service"].sum()
turbo_engines = turbo_engines.merge(turbo_in_service)
turbo_engines["Accidents/In Service/000"] = turbo_engines["Accidents"]/turbo_engines["In
Service"]*1000
```

```
# Turboprop fatal
turbo_fatal = engines_fatal[(engines_fatal["EngineType"] ==
"Turboprop")].groupby("Year")["Fatal"].sum().reset_index()
turbo_fatal = turbo_fatal.merge(turbo_engines, on = "Year")
turbo_fatal.drop(columns = ["Accidents","Accidents/In Service/000"],inplace = True)
turbo_fatal["Fatal/In Service/000"] = turbo_fatal["Fatal"]/turbo_fatal["In Service"]*1000
```

 # Figure 10 - Total and Fatal Accidents by Engine Type

```
def engine():
    # Set variables
    years =
["2008","2009","2010","2011","2012","2013","2014","2015","2016","2017","2018","2019","2020"]
    num_years = len(years)
    jets = jet_engines["Accidents/In Service/000"]
    turbos = turbo_engines["Accidents/In Service/000"]
    jets_engines_fatal = jet_fatal["Fatal/In Service/000"]
    turbo_engines_fatal = turbo_fatal["Fatal/In Service/000"]
    pos = np.arange(num_years)+.5

    # Plot Chart
    fig = plt.figure(facecolor='white', edgecolor='none',figsize = (10,10))
    ax1 = fig.add_axes([0.05, 0.1, 0.35, 0.8])
    ax2 = fig.add_axes([0.6, 0.1, 0.35, 0.8])
    ax1.set_facecolor("white")
    ax2.set_facecolor("white")

    ax1.set_xticks(np.arange(0,5,1))
    ax2.set_xticks(np.arange(0,5,1))
    ax1.set_xlim(0,4)
    ax2.set_xlim(0,4)
    ax1.xaxis.set_ticks_position('top')
    ax2.xaxis.set_ticks_position('top')

    for loc, spine in ax1.spines.items():
        if loc!='right':
            spine.set_color('none')

    for loc, spine in ax2.spines.items():
        if loc!='left':
            spine.set_color('none')

    ax1.barh(pos,turbos, align='center',color = "lightsteelblue", edgecolor='None')
    ax1.barh(pos, turbo_engines_fatal, align='center', facecolor='slategrey',height=0.5,
edgecolor='None')
    ax1.set_yticks([])
    ax1.xaxis.grid(True,linestyle = ":",color = "lightgrey")
    ax1.invert_xaxis()
    ax1.invert_yaxis()

    ax2.barh(pos, jets, align='center', color = "lightsteelblue",edgecolor='None')
    ax2.barh(pos, jets_engines_fatal, align='center', facecolor='slategrey',height=0.5,
edgecolor='None')
```

```
    ax2.set_yticks([])
    ax2.xaxis.grid(True,linestyle = ":",color = "lightgrey")
    ax2.spines['left'].set_visible(True)
    ax2.invert_yaxis()

    transform = transforms.blended_transform_factory(fig.transFigure, ax2.transData)
    for i, label in enumerate(years):
        ax2.text(0.5, i+0.5, label, ha='center', va='center',transform=transform,fontsize = 14)

    # Titles and Legend
    ax2.set_title('Jets', x=0.05, y=1.05, fontsize=14)
    ax1.set_title('Turboprops', x=0.87, y=1.05, fontsize=14)
    fig.suptitle('Accidents per 1000 In Service\n(2008-2020)', x=0.95, y = 1,ha='right',fontsize = 14)
    labels = ["Total Accidents","Fatal Accidents"]
    ax2.legend(labels = labels,facecolor = "white",frameon = False,ncol = 1, loc = 'lower left',
            bbox_to_anchor=(0.5,0.8),fontsize = 13)

    plt.show()

# ACCIDENTS BY FLIGHT STAGE

# find stage by total and fatal accidents
stage = pd.DataFrame(commercial.FlightPhase.value_counts()).reset_index()
stage.columns = ["Flight_phase","Number"]
fatal_stage =commercial[commercial["Fatalities"]
>0].groupby("FlightPhase")["Date"].count().reset_index()
fatal_stage.columns = ["Flight_phase","Number"]
fatal_stage = fatal_stage.sort_values(by = "Number",ascending = False)
new_row = {'Flight_phase':"Tow",'Number':0}
fatal_stage = fatal_stage.append(new_row,ignore_index = True)

# Create combined categories to reduce complexity of data
stage['Flight_phase'] = stage['Flight_phase'].replace(['Tow','Manoeuvring'],'Manoeuvring')
fatal_stage['Flight_phase'] =
fatal_stage['Flight_phase'].replace(['Tow','Manoeuvring'],'Manoeuvring')
stage['Flight_phase'] = stage['Flight_phase'].replace(['Initial Climb'],'Climb')
stage = stage[stage["Flight_phase"]!= "Unknown"]
fatal_stage['Flight_phase'] = fatal_stage['Flight_phase'].replace(['Initial Climb'],'Climb')
fatal_stage = fatal_stage[fatal_stage["Flight_phase"]!= "Unknown"]
```

 # Figure 11 - Total and Fatal Accidents by Flight Stage

```
def stage_flight():
    # Set variables
    width = 0.5
    fig, ax = plt.subplots(figsize = (12,4),edgecolor = None)
    ax.set_facecolor(color = "white")

    # Plot chart
    ax.bar(stage.Flight_phase,stage.Number,width, color = "lightsteelblue")
    ax.bar(fatal_stage.Flight_phase,fatal_stage.Number,width,color = "slategrey")
    ax.tick_params(axis='x', which='both', bottom=False, top=False, labelbottom=True)
    plt.xticks(fontsize=10 )
    ax.set_ylabel("Number", fontsize = 12)
```

```
    ax.set_yticks(np.arange(0, 1000, step=200))
    ax.set_ylim(0,900)
    ax.yaxis.grid(True,linestyle = ":",color = "grey")
    ax.set_title("Total and Fatal Accidents By Flight Phase",fontsize =15)
    ax.legend(labels = ["Total Accidents","Fatal Accidents"],facecolor = "white",frameon = False,
            bbox_to_anchor=(1,0.9),fontsize = 14)
    plt.show()

# ACCIDENTS BY OPERATOR

# Get total accidents by operator
operator_flights = operator_flights[operator_flights['annual_flights']>0]
operator_flights = operator_flights[operator_flights['accidents_5y']>0]
operator_flights["Accident_rate"] =
operator_flights["accidents_5y"]/operator_flights["annual_flights"]*1000
operator_flights = operator_flights.sort_values(by = "Accident_rate",ascending = False)
```

 # Figure 12 - Total Accidents By Operator

```
def operator():
    # Set variables
    operator_top = operator_flights.head(10)
    operator_bot = operator_flights.tail(10)
    my_list = [0,1,2,3,4,5,6,7,8,9]
    labels1 = ["Buffalo Airways (CAN)","AirAsia (PHI)","Omni Air (USA)","Kalitta Air (USA)",
            "Eagle Air (USA)","Blue Sky (KEN)","Jordan Aviation (JOR)","Lynden Air Cargo (USA)",
            "West Wind Aviation (CAN)", "Buraq Air (LIB)"]

    labels2 = ["Qatar Airways (QAT)","Saudia Arabian Airlines (SAU)","Xiamen Airlines
(CHN)","SkyWest Airlines (USA)",
          "Republic Airways (USA)","LATAM Airlines (BRA)","All Nippon Airways (JPN)","Lufthansa
(GER)",
          "Air China (CHN)","China Eastern (CHN)"]

    # Plot charts
    fig, (ax1,ax2) = plt.subplots(1,2,figsize = (12,4))
    plt.suptitle("Total and Fatal Accident Rates - Highest and Lowest Ten (2016-2020)",fontsize = 14, x
= 0.45,y = 1.1)
    plt.subplots_adjust(hspace=0.1,wspace = 1)
    fig.text(0.45, -0.09, 'Number of Accidents per 1000 Flights', ha='center',fontsize = 13)

    # Chart 1
    ax1.set_facecolor("white")
    ax1.set_title("Highest Ten",fontsize = 10,pad = 15)
    ax1.tick_params(axis='y', which='both', left=False, right=False, labelleft=True)
    ax1.hlines(y = my_list,xmin = 0, xmax = operator_top['Accident_rate'],color = "#007acc",alpha =
0.2, linewidth = 8)
    ax1.xaxis.grid(True,linestyle = ":",color = "lightgrey")
    ax1.plot(operator_top["Accident_rate"],labels1,"o",markersize = 8, color = "#007acc",alpha = 0.6)
    ax1.spines['top'].set_visible(False)
    ax1.set_xticks(np.arange(0, 20,2))
    ax1.spines['left'].set_bounds(1,len(my_list))
    ax1.set_xlim(0,20)
    ax1.spines['left'].set_position(('outward',8))
```

```
    ax1.spines['bottom'].set_position(('outward',5))
    ax1.invert_yaxis()

    # Chart 2
    ax2.set_facecolor("white")
    ax2.hlines(y = my_list,xmin = 0, xmax = operator_bot['Accident_rate'],color = "#007acc",alpha =
0.2, linewidth = 8)
    ax2.tick_params(axis='y', which='both', left=False, right=False, labelleft=True)
    ax2.xaxis.grid(True,linestyle = ":",color = "lightgrey")
    ax2.plot(operator_bot["Accident_rate"],labels2,"o",markersize = 8, color = "#007acc",alpha = 0.6)
    ax2.set_title("Lowest Ten*",fontsize = 11,pad = 15)
    ax2.spines['top'].set_visible(False)
    ax2.spines['left'].set_bounds(1,len(my_list))
    ax2.spines['left'].set_position(('outward',8))
    ax2.spines['bottom'].set_position(('outward',5))

    plt.show()
```

 # Figure 13 - Fatal Accidents By Operator

```
# Fatal accidents by operator
operator_flights_fatal = operator_flights[operator_flights['fatalaccidents_5y']>0]
operator_flights_fatal["Accident_rate_fatal"] =
operator_flights_fatal["fatalaccidents_5y"]/operator_flights["annual_flights"]*100000
operator_flights_fatal = operator_flights_fatal.sort_values(by = "Accident_rate_fatal",ascending =
False)
operator_flights_fatal = operator_flights_fatal[2:]

def fatal_op():
    # Set variables
    my_list3 = list(range(0,len(operator_flights_fatal.operatorName)))
    labels3 = ["Air Sunshine (USA)","Air Niugina (PNG)",
           "Iran Aseman (IRN)","Pakistan Airlines (PAK)","Ural Airlines (RUS)","Egyptair (EGY)","Fly Dubai
(UAE)",
           "Ethiopian Airlines (ETH)","Emirates Airline (UAE)","Aeroflot (RUS)","Lion Air
(IDN)","Southwest Airlines (USA)"]

    # Plot charts
    fig, ax3 = plt.subplots(figsize = (10,6))

    # Chart 3
    ax3.set_facecolor("white")
    ax3.set_facecolor("white")
    ax3.hlines(y = my_list3,xmin = 0, xmax = operator_flights_fatal['Accident_rate_fatal'],color =
"#007acc",alpha = 0.2, linewidth = 8)
    ax3.plot(operator_flights_fatal["Accident_rate_fatal"],labels3,"o",markersize = 8, color =
"#007acc",alpha = 0.6)
    ax3.xaxis.grid(True,linestyle = ":",color = "lightgrey")
    ax3.invert_yaxis()

    # Titles and Axes
    ax3.set_title("Fatal Accidents per 1000 Flights (2016-2020)",fontsize = 12, pad = 20)
    ax3.tick_params(axis='y', which='both', left=False, right=False, labelleft=True)
    ax3.spines['top'].set_visible(False)
```

```
    ax3.set_xlabel("Number of Fatal Accidents per 1000 Flights")
    ax3.xaxis.labelpad = 25
    ax3.spines['left'].set_bounds(1,len(my_list3))
    ax3.spines['left'].set_position(('outward',8))
    ax3.spines['bottom'].set_position(('outward',5))
    ax3.set_xlim(0,5)
    ax3.set_xticks(np.arange(0,5,0.5))
    xticks = ax3.xaxis.get_major_ticks()
    xticks[9].set_visible(False)

    plt.show()

# ACCIDENTS BY SAFETY CERTIFICATION

# Get certified operators
cert = operator_flights[operator_flights["is_iosa_certified"] == 1]
cert_rate = cert["accidents_5y"].sum()/cert["annual_flights"].sum()*1000000
cert_rate_fatal = cert["fatalaccidents_5y"].sum()/cert["annual_flights"].sum()*1000000

# Non-Certified operators
non_cert = operator_flights[operator_flights["is_iosa_certified"] == 0]
non_cert_rate = non_cert["accidents_5y"].sum()/non_cert["annual_flights"].sum()*1000000
non_cert_rate_fatal =
non_cert["fatalaccidents_5y"].sum()/non_cert["annual_flights"].sum()*1000000
```

 # Figure 14 - Accident Rates by IOSA Certification

```
def safety():
    # Set variables
    y = ["Certified","Non-Certified"]
    rate = [cert_rate,non_cert_rate]
    fatal = [cert_rate_fatal,non_cert_rate_fatal]

    # Plot Chart
    fig, (ax1,ax2) = plt.subplots(1,2,figsize = (10,4))
    plt.suptitle("Total and Fatal Accident Rates By IOSA Certification (2016-2020)",fontsize = 14, x =
0.5,y = 1.1)
    plt.subplots_adjust(wspace = 0.5)

    ax1.bar(y,rate,color = "lightsteelblue")
    ax1.set_title("Total Accidents",fontsize = 10)
    ax1.yaxis.grid(True,linestyle = ":",color = "lightgrey")
    ax1.tick_params(axis='x', which='both', bottom=False, top=False, labelbottom=True)
    ax1.set_facecolor("white")
    ax1.set_ylabel("Accidents per Million Flights",fontsize = 10,labelpad = 10)
    ax1.set_xlabel("IOSA Status",fontsize = 10,labelpad = 10)

    ax2.bar(y,fatal,color = "slategrey")
    ax2.set_title("Fatal Accidents",fontsize = 10)
    ax2.tick_params(axis='x', which='both', bottom=False, top=False, labelbottom=True)
    ax2.yaxis.grid(True,linestyle = ":",color = "lightgrey")
    ax2.set_facecolor("white")
    ax2.set_ylabel("Fatal Accidents per Million Flights",fontsize = 10,labelpad = 10)
    ax2.set_xlabel("IOSA Status",fontsize = 10,labelpad = 10)
```

```
   plt.show()


# ACCIDENTS BY STATE OF REGISTRATION

# Get incidents numbers and number of fatalities an put into dataframe
state = pd.DataFrame(commercial.groupby("StateOfRegistry",as_index = False)["Date"].count())
state.columns = ["State","Total_acc"]
fatal_state = pd.DataFrame(commercial[commercial["Fatalities"]
>0].groupby("StateOfRegistry",as_index = False)["Date"].count())
fatal_state.columns = ["State","Fatal_acc"]
all_state = state.merge(fatal_state, how = "left",on = "State")
all_state["Fatal_acc"] = all_state["Fatal_acc"].fillna(0)
all_state["Acc_no_fat"] = all_state["Total_acc"] - all_state["Fatal_acc"]

# Departures by state
state_departures = data_departs.groupby("State",as_index =
False)["Departures"].sum().sort_values(by = "Departures",
                                                    ascending = False)

# Merge the dataframes and create a new columns to show the accidents by state operator
state_combined = all_state.merge(state_departures,how = "left",on = "State")
state_combined["state_acc_thoudeps"] =
(state_combined["Total_acc"]/state_combined["Departures"])*1000
state_combined["fatal_acc_thoudeps"] =
(state_combined["Fatal_acc"]/state_combined["Departures"])*1000

# Extract best and worst states
state_worst = state_combined.sort_values(by = "state_acc_thoudeps",ascending = False).head(10)
state_best = state_combined.sort_values( by = "state_acc_thoudeps",ascending = False)
state_best = state_best[state_best["state_acc_thoudeps"] > 0].tail(10)
fatal_worst = state_combined.sort_values(by = "fatal_acc_thoudeps",ascending = False).head(10)
fatal_best = state_combined.sort_values( by = "fatal_acc_thoudeps",ascending = False)
fatal_best = fatal_best[fatal_best["fatal_acc_thoudeps"] > 0].tail(10)
```

 # Figure 15 - Total and Fatal Accident Rates by State of Registration

```
def by_state():
   # Set variables
   label1 = ["Sao Tome & Prin","Bermuda","Congo (DR)","Suriname","North
Korea","Gambia","Cent.Afr.Rep","Guinea","Bhutan",
        "Sudan"]
   label2 = ["Japan","Sweden","Croatia","Chile","Spain","Oman","Finland","New
Zealand","China","Italy"]
   label3 = ["Sao Tome & Prin", "Bermuda","Suriname","Congo
(DR)","Georgia","Sudan","Somalia","Congo","Kyrgyzstan","Tajikistan"]
   label4 =
["Spain","India","Norway","Thailand","Brazil","USA","Germany","Australia","France","China"]

   # Plot charts
   fig, ax = plt.subplots(2,2,sharex = "col",figsize = (12,6))
   fig.text(0.45, 0.02, 'Accidents per Thousand Flights', ha='center',fontsize = 13)
```

```
    plt.suptitle("Ten Highest and Lowest Countries Total and Fatal Accidents (2008-2020)",fontsize =
15)
    plt.subplots_adjust(hspace=0.3,wspace = 0.25)

    ax[0,0].barh(label1,state_worst["state_acc_thoudeps"], align='center', color='lightsteelblue')
    ax[0,0].set_title("Highest Total Accident Rate",fontsize = 10)
    ax[0,0].set_facecolor("White")
    ax[0,0].xaxis.grid(True,linestyle = ":",color = "grey")
    ax[0,0].tick_params(axis='y', which='both', left=False, right=False, labelleft=True)
    ax[0,0].invert_yaxis()

    ax[0,1].barh(label2, state_best["state_acc_thoudeps"], align='center', color='lightsteelblue')
    ax[0,1].tick_params(axis='y', which='both', left=False, right=False, labelleft=True)
    ax[0,1].set_title("Lowest Total Accident Rate",fontsize = 10)
    ax[0,1].set_facecolor("White")
    ax[0,1].xaxis.grid(True,linestyle = ":",color = "grey")

    ax[1,0].barh(label3, fatal_worst["fatal_acc_thoudeps"], align='center', color='slategrey')
    ax[1,0].tick_params(axis='y', which='both', left=False, right=False, labelleft=True)
    ax[1,0].set_title("Highest Fatal Accident Rate",fontsize = 10)
    ax[1,0].set_facecolor("White")
    ax[1,0].xaxis.grid(True,linestyle = ":",color = "grey")
    ax[1,0].invert_yaxis()

    ax[1,1].barh(label4, fatal_best["fatal_acc_thoudeps"], align='center', color='slategrey')
    ax[1,0].tick_params(axis='y', which='both', left=False, right=False, labelleft=True)
    ax[1,1].set_title("Lowest Fatal Accident Rate for\n Countries With at Least One Fatal
Accident*",fontsize = 10)
    ax[1,1].set_facecolor("White")
    ax[1,1].xaxis.grid(True,linestyle = ":",color = "grey")

    plt.show()

# ACCIDENTS BY REGION of OCCURRENCE

# Put to millions departures
reg_deps = deps/1000000

# Get locations of accidents, drop ocean incidents and express per departures
locations = pd.DataFrame(commercial.groupby("Region",as_index =
False)["Date"].count()).set_index("Region")
locations.drop(["Enroute/Oceans/Other"],axis = 0,inplace = True)
locations["Deps"] = reg_deps["Total"]
locations["Acc_deps"] = locations["Date"]/locations["Deps"]
locations.columns = ["Accidents","Deps","Acc_deps"]

# Repeat above for fatal accidents
locations_fatal = pd.DataFrame(commercial[commercial["Fatalities"]>0].groupby("Region",as_index
= False)["Date"].count()).set_index("Region")
locations_fatal.drop(["Enroute/Oceans/Other"],axis = 0,inplace = True)
locations_fatal["Deps"] = reg_deps["Total"]
locations_fatal["Fat_deps"] = locations_fatal["Date"]/locations_fatal["Deps"]
locations_fatal.columns = ["Fatal","Deps","Fat_deps"]
```

 # Figure 16 (a) - Total and Fatal Accident Rates by Region of Occurrence

```
def region():
    labels = ['East Asia & Pacific', 'Europe & Central Asia',
      'Latin America & Caribbean', 'Middle East & North Africa',
      'North America', 'South Asia', 'Sub-Saharan Africa']
    accidents = locations.Acc_deps
    fatal = locations_fatal.Fat_deps

    x = np.arange(len(labels))  # the label locations
    width = 0.35  # the width of the bars

    fig, ax = plt.subplots(figsize = (10,6))
    ax.barh(x + width/2, accidents, width, label='Total Accidents',color = "lightsteelblue")
    ax.barh(x - width/2, fatal, width, label='Fatal Accidents',color = "slategrey")

    # Add some text for labels, title and custom x-axis tick labels, etc.
    ax.set_facecolor("White")
    ax.set_xlabel('Accidents per million departures',labelpad = 10)
    ax.set_title("Total and Fatal Accidents By Region (2008-2020)",fontsize = 13)
    ax.set_xlim(0,26)
    plt.yticks(fontsize=11)
    ax.xaxis.grid(True,linestyle = ":",color = "grey")
    ax.set_yticks(x)
    ax.set_yticklabels(labels)
    ax.legend(fontsize = 13,facecolor = "white",frameon = False,bbox_to_anchor=(0.5,0.55))

    fig.tight_layout()

    plt.show()

# Get trend in total accidents over the period
# Group by region and year
by_year = pd.DataFrame(commercial.groupby(["Region","Year"],as_index = False)["Date"].count())
by_year = by_year[by_year["Region"]!= 'Enroute/Oceans/Other']

# Function to get accidents per regional departures
def trans(name):
    reg = by_year[by_year["Region"] == name][0:-1][:]
    reg["Acc"] = reg_deps.T[name][0:-1].tolist()
    reg["Acc_deps"] = reg["Date"]/reg["Acc"]
    mean_val = reg["Acc_deps"].mean()
    return reg,mean_val

EAP,EAP_mean = trans("East Asia & Pacific")
ECA,ECA_mean = trans("Europe & Central Asia")
LAC,LAC_mean = trans("Latin America & Caribbean")
MEN,MEN_mean = trans("Middle East & North Africa")
NA,NA_mean = trans("North America")
SA,SA_mean = trans("South Asia")
SSA,SSA_mean = trans("Sub-Saharan Africa")
```

 # Figure 16 (b) - Accident Trend By Region of Occurrence

```
# Note this chart is combined with Figure 16 to produce one figure in the report.

def spark():
    # Set Variables
    def trans(name):
        reg = by_year[by_year["Region"] == name][0:][0:-1]
        reg["Acc"] = reg_deps.T[name][0:-1].tolist()
        reg["Acc_deps"] = reg["Date"]/reg["Acc"]
        mean_val = reg["Acc_deps"].mean()
        return reg,mean_val

    # Values
    EAP,EAP_mean = trans("East Asia & Pacific")
    ECA,ECA_mean = trans("Europe & Central Asia")
    LAC,LAC_mean = trans("Latin America & Caribbean")
    MEN,MEN_mean = trans("Middle East & North Africa")
    NA,NA_mean = trans("North America")
    SA,SA_mean = trans("South Asia")
    SSA,SSA_mean = trans("Sub-Saharan Africa")

    labels = EAP.Year

    # Plot Chart
    fig, axs = plt.subplots(7,1,figsize = (4,8))
    fig.suptitle("Total Accident Trend By Region (2008-2020)",fontsize = 12)
    plt.subplots_adjust(left=None, bottom=None, right=None, top=None, wspace=None, hspace=0.5)

    axs[6].plot(labels,EAP["Acc_deps"],color = "slategrey")
    axs[6].axhline(y=EAP_mean, linewidth=1, linestyle = "--", color='slategrey')
    axs[5].plot(labels,ECA["Acc_deps"],color = "maroon")
    axs[5].axhline(y=ECA_mean, linewidth=1, linestyle = "--", color='slategrey')
    axs[4].plot(labels,LAC["Acc_deps"],color = "slategrey")
    axs[4].axhline(y=LAC_mean, linewidth=1, linestyle = "--", color='slategrey')
    axs[3].plot(labels,MEN["Acc_deps"],color = "slategrey")
    axs[3].axhline(y=MEN_mean, linewidth=1, linestyle = "--", color='slategrey')
    axs[2].plot(labels,NA["Acc_deps"],color = "maroon")
    axs[2].axhline(y=NA_mean, linewidth=1, linestyle = "--", color='slategrey')
    axs[1].plot(labels,SA["Acc_deps"],color = "slategrey")
    axs[1].axhline(y=SA_mean, linewidth=1, linestyle = "--", color='slategrey')
    axs[0].plot(labels,SSA["Acc_deps"],color = "slategrey")
    axs[0].axhline(y=SSA_mean, linewidth=1, linestyle = "--", color='slategrey')
    ax_array = [axs[0],axs[1],axs[2],axs[3],axs[4],axs[5],axs[6]]

    axs[6].text(2016,5,"East Asia & Pacific",fontsize = 11)
    axs[5].text(2015.5,7.5,"Europe & Central Asia",fontsize = 11)
    axs[4].text(2014,10,"Latin America & Caribbean",fontsize = 11)
    axs[3].text(2014,12,"Middle East & North Africa",fontsize = 11)
    axs[2].text(2016.5,9,"North America",fontsize = 11)
    axs[1].text(2016.8,8.5,"South Asia",fontsize = 11)
    axs[0].text(2015.5,35,"Sub-Saharan Africa",fontsize = 11)

    # Remove chart junk
    for axes in ax_array:
        axes.tick_params(axis = "x", which = "both", bottom = False, top = False)
```

```
        axes.tick_params(axis = "y", which = "both", left = False, right = False)
        axes.set_yticklabels([])
        axes.set_xticklabels([])
        axes.set_facecolor("white")

    # Annotations

    plt.show()


# ACCIDENTS BY FLEET SIZE, FLEET AGE AND INCOME PER CAPITA

# Fleet age by Country
fleet_age = fleet_age[["countryName","countryCode","av_fleet_age","aircraft"]]
fleet_age.columns = ["Country Name","Country Code","Fleet Age","Fleet Size"]
fleet_age_country_av = pd.DataFrame(fleet_age.groupby("Country Code",as_index = False)["Fleet
Age"].mean())
fleet_size_country_av = pd.DataFrame(fleet_age.groupby("Country Code",as_index = False)["Fleet
Size"].mean())

# GDP per Capita
gdp_cap = gdp_cap[["Country Name","Country Code","Average (000s)"]]
gdp_cap.sort_values(by = "Average (000s)",ascending = False).head(10)

# Merge into one dataframe
mergeddf = gdp_cap.merge(fleet_age_country_av,on = "Country Code")
mergeddf = mergeddf.merge(fleet_size_country_av, on = "Country Code")
mergeddf2 = mergeddf.merge(state_combined,left_on = "Country Code",right_on = "State")

# Merge with world bank income category and set categories to numerical values
mergeddf2_combined = mergeddf2.merge(income_cat, how = 'left', on = "Country Name")
mergeddf2_combined["Income_cat"] = mergeddf2_combined['Income'].replace(['High','Upper
Middle','Lower Middle','Lower'],[1,2,3,4])
mergeddf2_combined["Income_cat"].astype(float)
```

 # Figure 17 - Accident Rate by Fleet Age, Fleet Size and Income per Capita

```
def bubble():
    fl = mergeddf2_combined["Fleet Age"]
    ar = np.log10(mergeddf2_combined["state_acc_thoudeps"])
    c = mergeddf2_combined["Income_cat"]
    size = mergeddf2_combined["Fleet Size"]*20
    fig, ax = plt.subplots(figsize = (14,8))
    ax.set_facecolor("White")
    z = np.polyfit(fl, ar, 1)
    p = np.poly1d(z)

    # Plot chart
    sc = ax.scatter(fl,ar,c = c,alpha = 0.6,s=size,cmap="viridis",edgecolors = "black",linewidth = 0.5)
    ax.plot(fl,p(fl),"slategrey",linestyle = "--",linewidth = 0.5,alpha = 1)
    ax.set_title("Log Accident Rate by Fleet Age, Fleet Size and World Bank Income Category",fontsize
= 16,pad = 20)
    ax.set_xlim(0,50)
    ax.spines['left'].set_visible(True)
```

```
    ax.spines['bottom'].set_visible(True)
    ax.spines["left"].set_color("lightgrey")
    ax.spines["bottom"].set_color("lightgrey")
    ax.set_xlabel("Fleet Age(years)",fontsize =11)
    ax.set_ylabel("log(Accidents per 100,000 Flights))",fontsize = 11)
    ax.text(22.5,-0.5,"Bermuda",fontsize = 9)
    ax.text(14.5,-1.4,"Ireland",fontsize = 9)
    ax.text(39,-2.3,"Honduras",fontsize = 9)
    #ax.text(4.5,-1.95,"Brunei",fontsize = 9)
    ax.text(33.5,-0.12,"Sao Tome & Principe",fontsize = 9)
    ax.text(3.6,-2.8,"India",fontsize = 9)
    ax.text(11.5,-0.75,"DR Congo",fontsize = 9)
    #ax.text(35.8,-0.87,"N.Korea",fontsize = 9)


    labels = {'High: 12,536+':1,'Upper Middle: 4,046 - 12,535':2, 'Lower Middle: 1,036 - 4,045':3,
'Lower: <1035':4}
    legend1 = ax.legend(handles=sc.legend_elements(num=[1,2,3,4])[0], labels=labels.keys(), fontsize
= 11,
                    facecolor = "white",frameon = False,title = "World Bank Income per Capita US$\n",loc =
"upper left")

    plt.setp(legend1.get_title(),fontsize=11)
    ax.add_artist(legend1)

    g = lambda s: np.sqrt(s/12)*10
    legend2 = ax.legend(*sc.legend_elements("sizes",num = 6, func = g,c = 'slategrey',alpha =
0.6),facecolor = "white",frameon = False, title = "Fleet Size",
          labelspacing = 4, loc = "upper right",handletextpad = 2)
    plt.setp(legend2.get_title(),fontsize=10)

    plt.show()


# RUN PLOTS

global_pass()
departures()
accidents()
fatalities()
above_50()
mon()
day()
size()
model()
engine()
stage_flight()
operator()
fatal_op()
safety()
by_state()
region()
spark()
bubble()
```

Appendix B – Accidents Dataset Link

Dataset Link

Appendix C – Variables and Sources

Variables	Measure	Datatype	Source	Reference
Accidents Data:				
Total Accidents	Number	Numerical	ICAO	[6]
Fatal Accidents	"	"		
Fatalities	"	"		
Operator	Name	Categorical Nominal		
State of Registry	Country	" "		
Flight Phase	ICAO stages	" "		
Plane Model	Model Type	" "		
Engine Type	Engine Type	" "		
Fleet Age	Years	Numerical		
Fleet Numbers	Numbers	Numerical		
Supporting Data*				
Passenger Numbers	Number	Numerical	World Bank	[8]
Departures by Year	"	"	ICAO	[6]
Passengers by Month	"	"	World Pop	[9]
Daily Flights	"	"	Flight Tracker	[10]
Seat Numbers	"	"	The Travel Insider	[11]
Number in Service	"	"	Dvbbank, Airfleets.net	[12][13]
GDP per Capita	"	"	World Bank	[14]
World Bank Income	"	"	ICAO Country Codes	[6]
World Region	Region	Categorical Nominal	World Bank	[15]

*Supporting data is used to create variables that allow valid comparisons, for example – accidents per million departures, per thousand daily flights or accidents per number of planes in service.