IO-AVSTATS - User Manual

Release 23.05.01

IO Aeronautical Autonomy Labs, LLC

1. Introduction

The **IO-AVSTATS-DB** database contains not only the NTSB's aviation accident data, but also a large number of supplementary data from a wide variety of data sources. To enable the most comprehensive analysis of this data, **IO-Aero** provides the following cloud-based applications as tools:

- **Association Rule Analysis**: to apply various association rule algorithms to selected aspects of event causes, such as phase of flight or cause of accident, and more,
- **Aviation Event Analysis**: allows detailed analysis of selectable event data using data profiling, maps, various chart types and more,
- **Database Profiling**: allows exploratory data analysis of all tables and views in the database **IO-AVSTATS-DB**,
- **US Aviation Fatal Accident Analysis**: this is a freely available but in terms of data and functionality very limited version of the **Aviation Event Analysis** application.

2. Getting Started

2.1. Application Access

The **US Aviation Fatal Accidents** application is freely available and can be accessed either through the IO-Aero website at www.io-aero.com under **Products** or directly through the link stats.io-aero.com.

The remaining functionality is only available to IO-Aero customers either via the IO-Aero website www.io-aero.com under **Members** or directly via one of the following links:

Association Rule Analysis:	slara.io-aero.com
Aviation Event Analysis:	ae1982.io-aero.com
Database Profiling:	pd1982.io-aero.com
Members Only Area:	members.io-aero.com

To use this functionality, you must first authenticate yourself using the **Sign in** button:

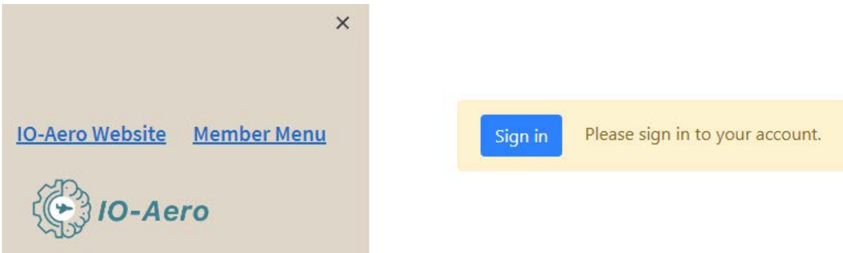

Figure 1: Application access lock

Authentication is done with username or email address and password. A process for changing or new password is available.

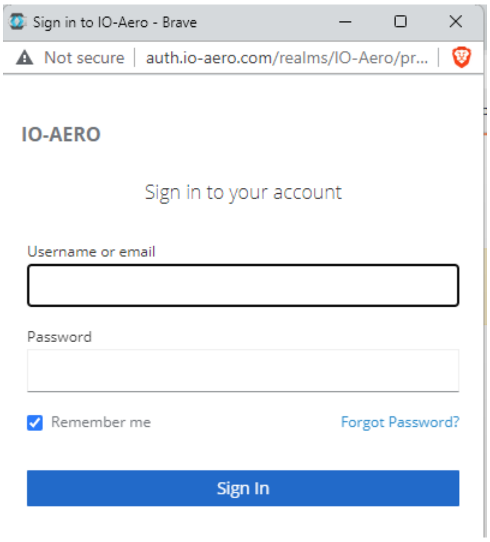

Figure 2: Authentication dialog

Depending on the personal permissions, the member menu provides the links to the applications under **`Applications`** and the database and user documentation in PDF format under **`Downloads`**. At the top left, access to the IO-Aero website is available.

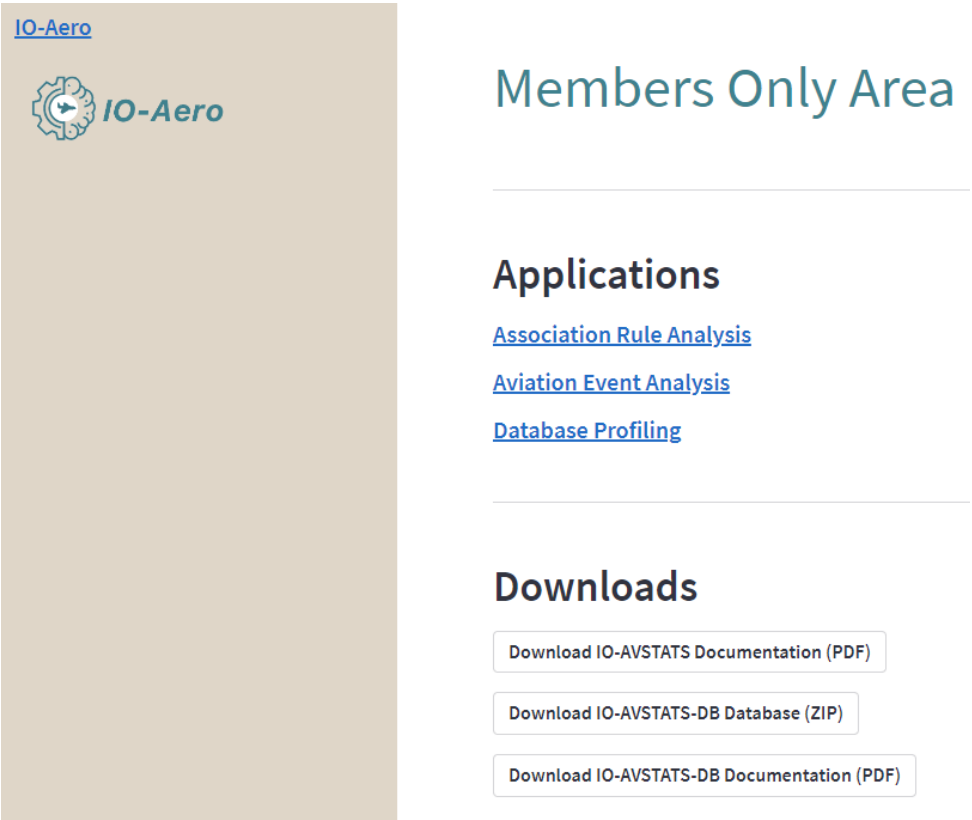

Figure 3: Sample menu in the members only area

In all applications, access to the IO-Aero website and access to the **`Members Only Area`** is available in the upper left corner.

Figure 4: Application standard controls top left

2.2. General Usability

All applications are based on Streamlit. Streamlit is a powerful and versatile library that makes it easy to build and deploy data-driven web apps with Python. The main idea behind Streamlit is to provide a fast and efficient way to build and deploy data science and machine learning projects, without the need for extensive coding knowledge or web development skills. Streamlit comes with pre-built widgets and tools that allow you to create interactive elements such as sliders, text inputs, and dropdowns, as well as data visualizations like charts and graphs.

The user interface of a Streamlit application is composed of multiple elements that can be arranged in various ways depending on the design of the application. Here are some of the key elements that we use to create a Streamlit user interface:

- Sidebar: The sidebar is a container on the left-hand side of the screen that can hold various widgets, such as sliders, drop-down menus, and checkboxes. The sidebar can be used to allow users to interact with the application by adjusting input parameters.
- **Title**: The title is the text element that appears at the top of the page and typically describes the purpose of the application.
- Main page: The main page is the primary area of the screen where the results of the application are displayed. This can include charts, tables, or any other visualizations that the application generates.
- Buttons: Buttons are clickable elements that perform a specific action when clicked. These can be used to initiate computations or navigate between different views of the application.
- Text elements: Text elements can be used to provide instructions, descriptions, or context for the application. These can be placed throughout the interface to help guide users through the application.
- Layout components: Streamlit provides various layout components, such as columns and containers, that can be used to organize the user interface in a structured and intuitive way.

Overall, the structure of a Streamlit application should be designed to provide a clear and intuitive user experience, with a focus on making it easy for users to interact with the application and understand its results.

3. Application Association Rule Analysis

3.1. Introduction

Association Rule Analysis, also known as Market Basket Analysis or Affinity Analysis, is a data mining technique that aims to identify the relationships between variables in a large dataset. Specifically, it focuses on finding the association between different items that are frequently bought together by customers in a transactional dataset.

In other words, it looks for patterns or rules that explain the co-occurrence of items in a dataset. For example, if a customer buys bread, there is a high likelihood that they will also buy butter, as these two items are often bought together. By identifying such patterns, businesses can understand the buying behavior of their customers and make better decisions about product placement, promotions, and pricing strategies.

Association Rule Analysis uses measures such as support, confidence, and lift to identify the most relevant patterns. Support refers to the frequency of occurrence of a particular itemset, confidence measures the strength of association between two items, and lift indicates the degree of association between two items beyond chance. The results of Association Rule Analysis are typically presented as a set of rules that express the relationship between items in terms of if-then statements.

3.2. Filter Options

3.3. Apriori Algorithm

The Apriori algorithm is a popular technique in data mining and machine learning used for finding frequent itemsets in a dataset. It is based on the idea that if an itemset is frequent, then all of its subsets must also be frequent.

The algorithm works by scanning the dataset to identify the support (frequency) of each item, and then using that information to find frequent itemsets. The support of an itemset is defined as the number of transactions in which the itemset appears. The algorithm then iterates through the dataset, looking for sets of items that appear together more often than a minimum support threshold.

The Apriori algorithm is called "Apriori" because it uses prior knowledge of frequent itemsets to prune the search space and make the algorithm more efficient. In other words, the algorithm starts by looking for frequent itemsets of length 1, and then uses those itemsets to generate candidate sets of length 2. It continues this process, generating candidate itemsets of increasing length until no more frequent itemsets can be found.

Overall, the Apriori algorithm is a useful tool for identifying frequently occurring patterns in large datasets and has been applied in a wide range of applications.

3.4. ECLAT Algorithm

The ECLAT (Equivalence Class Clustering and Bottom-Up Lattice Traversal) algorithm is a popular data mining technique used for finding frequent item sets in a transactional dataset.

The algorithm works by first finding all the individual items that occur frequently in the dataset. It then uses a depth-first search approach to efficiently find all the combinations of items that occur frequently together (i.e., the frequent item sets).

To achieve this, ECLAT creates a vertical representation of the dataset, where each column represents an item and each row represents a transaction. It then uses this representation to determine the support (i.e., frequency of occurrence) of each itemset.

The algorithm recursively combines frequent item sets of increasing size, and the resulting item sets are stored in a lattice structure. The lattice structure provides an efficient way to store and search for frequent item sets.

Overall, the ECLAT algorithm is a fast and memory-efficient method for finding frequent item sets in large transactional datasets

3.5. FP Growth Algorithm

The FP-Growth algorithm is a popular data mining technique used to find frequent patterns in large datasets.

Here is a simple explanation of how it works:

1. First, the algorithm scans the dataset and calculates the frequency of each item (such as a product in a store) in the dataset.

2. Then, it builds a tree structure called an FP-Tree based on the frequent items identified in step 1. Each path in the tree represents a frequent pattern in the dataset.

3. Next, the algorithm uses the FP-Tree to mine the dataset for frequent item sets. It does this by recursively mining conditional sub-trees that are built from the original tree.

4. Finally, the algorithm returns a list of frequent item sets along with their corresponding support values (i.e., the frequency of occurrence in the dataset).

One of the benefits of the FP-Growth algorithm is that it can be faster than other algorithms that use an apriori approach, which require multiple passes over the dataset to identify frequent item sets. By building a tree structure, FP-Growth can reduce the number of scans needed to identify frequent patterns in the dataset, making it an efficient and effective method for large-scale data mining.

3.6. FP Max Algorithm

The FP-max algorithm is a popular algorithm for frequent pattern mining in data mining and machine learning. The algorithm is used to find frequent patterns, which are patterns that appear frequently in a given set of data.

The algorithm works by first scanning the data set to find all of the frequent items. It then uses these frequent items to build a tree structure called an FP-tree. Each branch in the tree represents a frequent item, and each node in the branch represents a transaction that contains that item.

Once the FP-tree is constructed, the algorithm recursively mines it to find all the frequent patterns. It does this by finding all the conditional patterns of each frequent item in the tree, starting with the most frequent item. A conditional pattern is a sub-pattern of a frequent pattern that appears in the same transactions as the frequent item.

The algorithm continues this process until no more frequent patterns can be found. The result is a set of frequent patterns that can be used to identify common trends and relationships in the data.

4. Application Aviation Event Analysis

4.1. Introduction

US Aviation Fatal Accident Analysis is an analysis of fatal aviation accidents that have occurred in the United States. The analysis involves examining data related to fatal accidents, including the number of accidents, fatalities, and the causes of accidents.

The analysis is important because it helps to identify trends and patterns in aviation accidents, which can be used to improve aviation safety. The analysis can also be used to identify areas where additional research or training may be needed to prevent future accidents.

Aviation safety is a critical concern, and the US Aviation Fatal Accident Analysis plays an important role in improving safety by identifying areas of concern and developing strategies to prevent future accidents.

4.2. Filter Options

4.2.1. Standard Version

4.2.2. Extended Version

4.3. Distance-based Charts

4.3.1. Nearest Airport

4.4. Event-based Charts

4.4.1. CICTT Codes

4.4.2. Event Types

4.4.3. Highest Injury Levels

4.4.4. Main Phases of Flight

4.4.5. Nearest Airport

4.4.6. Phases of Flight

4.4.7. Safety Systems

4.4.8. Top Level Logical Parameters

4.5. Fatality-based Charts

4.5.1. FAR Operations Parts

4.5.2. Selected FAR Operations Parts

4.6. Data Profile

4.7. Detailed Data

4.8. Map

5. Application Database Profiling

5.1. Introduction

Exploratory Data Analysis (EDA) is the process of examining and analyzing data to summarize its main characteristics. The goal of EDA is to understand the underlying patterns, relationships, and trends within the data.

EDA typically involves a range of techniques, such as data visualization, statistical analysis, and data mining, to identify patterns and insights in the data. The main objectives of EDA are to identify anomalies, missing values, outliers, and to check the assumptions of statistical models.

EDA is a crucial step in the data analysis process because it provides insights into the data that can guide further analysis and modeling. EDA is often used in data science and machine learning projects, as well as in business and other fields where data analysis is important.

Data profiling is the process of examining and analyzing data from various sources to understand the structure, content, relationships, and quality of the data. The goal of data profiling is to create a comprehensive understanding of the data so that it can be used effectively for analysis, decision-making, and other purposes.

Data profiling involves collecting information about the data, such as its size, format, data types, completeness, uniqueness, patterns, and outliers. This information can be used to identify data quality issues, such as missing values, inconsistent data, or data that does not conform to expected standards.

Data profiling can also help to identify relationships between data elements, such as primary keys, foreign keys, and dependencies. This information is critical for data modeling and database design.

Overall, data profiling is an important step in the data management process, as it helps organizations to better understand and manage their data, and to ensure that it is accurate, reliable, and consistent.

Pandas profiling is a Python library that generates an interactive report from a pandas DataFrame. The report includes various statistics and visualizations that provide insights into the data. It helps in understanding the data distribution, identifying missing values, detecting outliers, and much more.

Pandas profiling offers a fast and easy way to get an overview of a dataset. It provides a comprehensive report that includes details about the data types, distribution, missing values, correlations, and much more. The report is generated in HTML format and can be easily exported to different formats.

Pandas profiling can be used for various data analysis tasks such as exploratory data analysis, data cleaning, and data preprocessing. It can be integrated with Jupyter notebooks, making it easy to generate reports on the go.

Overall, Pandas profiling is a powerful tool that can help data scientists, analysts, and researchers gain insights into their data quickly and easily.

Pandas profiling is a Python library that allows for automatic EDA (exploratory data analysis) of a pandas DataFrame. It generates a comprehensive report containing information about the data type of each column, missing values, statistical summaries, and other insights.

The library provides an easy way to understand the data without much effort. With the help of Pandas profiling, you can generate interactive reports that contain visualizations, such as histograms, correlation matrices, and scatter plots, that help you to get a better understanding of the data.

Pandas profiling is a very useful tool for data scientists and analysts who need to quickly understand the structure and content of a dataset. It can save a lot of time and effort by automating the process of EDA, which can be time-consuming and repetitive.

5.2. Filter Options

5.3. Data Profile

5.4. Detailed Data

6. Application US Aviation Fatal Accident Analysis

6.1. Introduction

US Aviation Fatal Accident Analysis is an analysis of fatal aviation accidents that have occurred in the United States. The analysis involves examining data related to fatal accidents, including the number of accidents, fatalities, and the causes of accidents.

The analysis is important because it helps to identify trends and patterns in aviation accidents, which can be used to improve aviation safety. The analysis can also be used to identify areas where additional research or training may be needed to prevent future accidents.

Aviation safety is a critical concern, and the US Aviation Fatal Accident Analysis plays an important role in improving safety by identifying areas of concern and developing strategies to prevent future accidents.

This is a data and functionally limited version of the 'Aviation Event Analysis' application.

The database view **`io_app_ae1982`** is used for the data. The data limitations concern:

- only events of type accident,
- only operations of type charter services (parts 135), regular scheduled air carriers (parts 121), or general aviation (parts 091x),
- only U.S. related fatal accidents from 2008 to present, i.e., accidents on U.S. soil, departure or planned arrival in the U.S., U.S. owner, U.S. operator, or U.S. registration.

Functionality is limited to:

- total and annual views on fatalities by selected FAR operations parts,
- total and annual views on preventable accidents by safety system, and
- the fatal accident map.

The application is divided into two parts: On the left side there is the so-called sidebar and on the right side the results are displayed. The sidebar has the functional controls in the upper part and the controls for filtering the data in the lower part. Each time the functional or filtering controls are changed, all the results on the right side are recalculated.

6.2. Filter Options

A range of years between 2008 and today can be selected here via this slider widget:

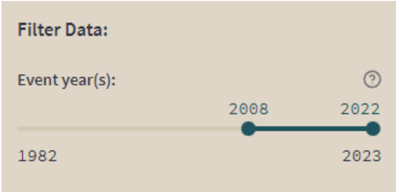

Figure 5: Filter control - interval of event years

6.3. Functional Control: Accident-based Charts

These analyses are based on fatal accidents in U.S. aviation.

The selected data can be displayed in up to three different chart types:

- The vertical bar chart shows the annual values,
- Pie chart and horizontal bar chart show the total values.

6.3.1. Preventable Accidents by Safety Systems

The percentage threshold limits the individual display of the affected safety systems. The safety systems whose percentage is below the threshold are grouped in the **below threshold** category.

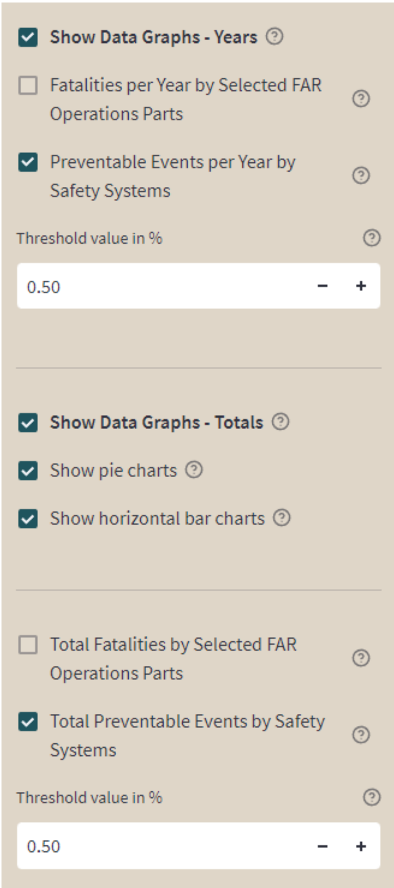

Figure 6: Functional controls

The high-level security system requirements defined in [Hook, Loyd & Sizoo, David & Fuller, Justin. (2022)] are mapped to the data in **IO-AVSTATS-DB** as follows:

- Airbone collision **is_midair_collision**
- Forced landing **is_rss_forced_landing** i.e., **is_attitude controllable** and **is_emergency_landing**
- Spin / stall **is_rss_spin_stall_prevention_and_recovery** i.e., **is_attitude_controllable** and **is_spin_stall**

- Terrain collision **`is_rss_terrain_collision_avoidance`** i.e., **`is_attitude_controllable`** and **`is_altitude_low`** and **`is_altitude_controllable`**

This data is mainly based on the database tables **`event_sequence`** and **`findings`**. However, since these data are not unique per event and moreover, several aircraft can be involved in an event, combinations of safety systems can occur, e.g.: **`Forced landing, Spin / stall`**.

Number of Preventable Accidents per Year by Safety Systems

☐ User guide: Years chart

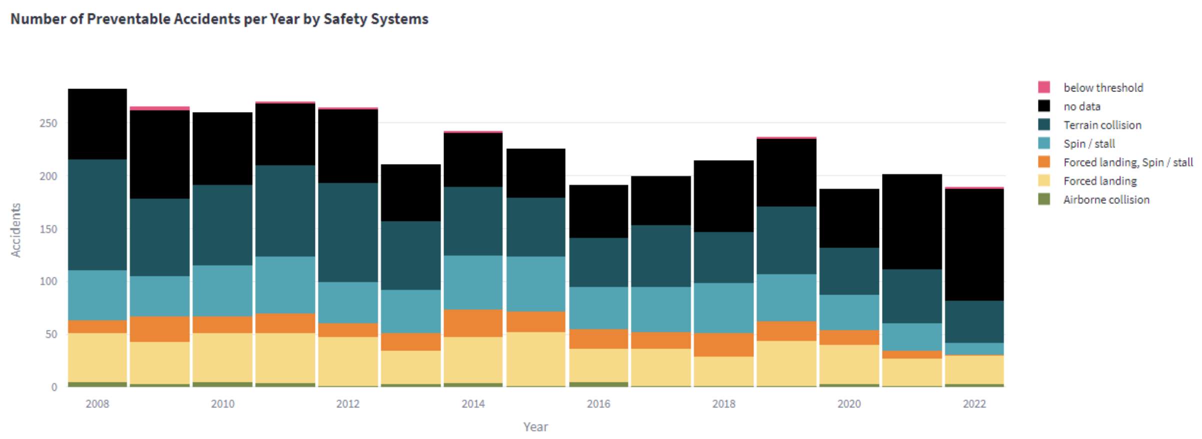

Figure 7: Bar chart representation of annual values

Total Number of Preventable Accidents by Safety Systems

☐ User guide: Totals chart

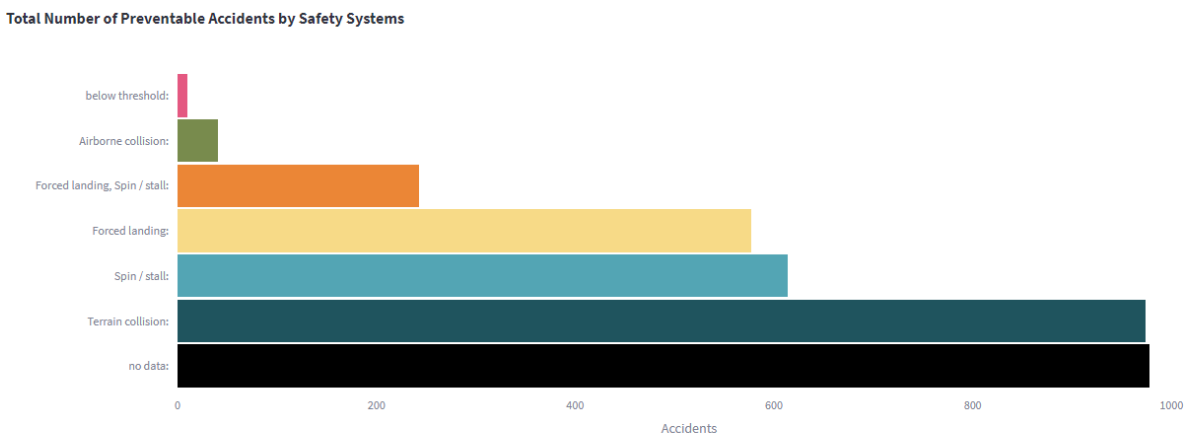

Figure 8: Horizontal bar chart representation of totals

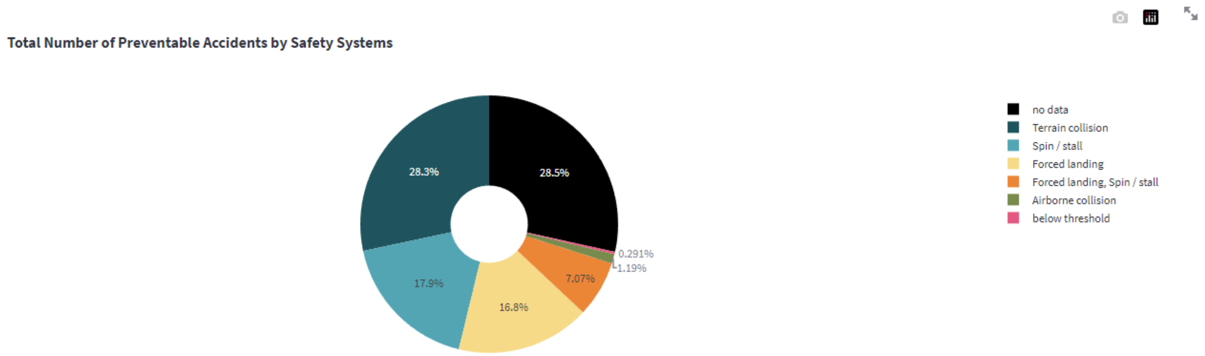

Figure 9: Pie chart representation of totals

6.4. Functional Control: Fatality-based Charts

These analyses are based on U.S. aviation fatalities.

The selected data can be displayed in up to three different chart types:

- The vertical bar chart shows the annual values,
- Pie chart and horizontal bar chart show the total values.

6.4.1. Selected FAR Operations Parts

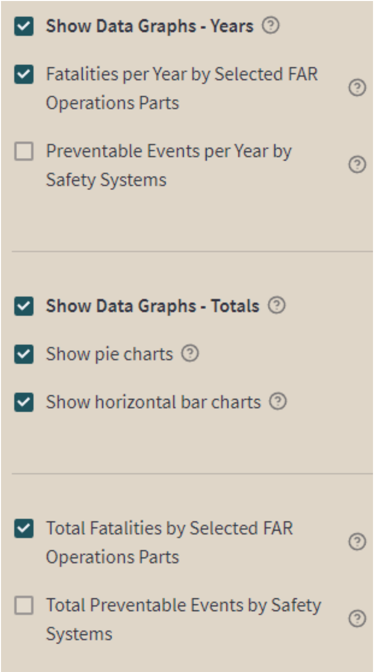

Figure 10: Functional controls

The fatalities processed here result exclusively from accidents that can be assigned to one of the following FAR operations parts:

- Parts 091x General operations
- Parts 121 Regular scheduled air carriers
- Parts 135 Charter type services

Since the assignment is made at the aircraft, accidents with multiple aircraft involved and different FAR Operations Parts may result in multiple fatality counts for the accidents involved. However, this only affects a negligible number of accidents.

Number of Fatalities per Year by Selected FAR Operations Parts

☐ User guide: Years chart

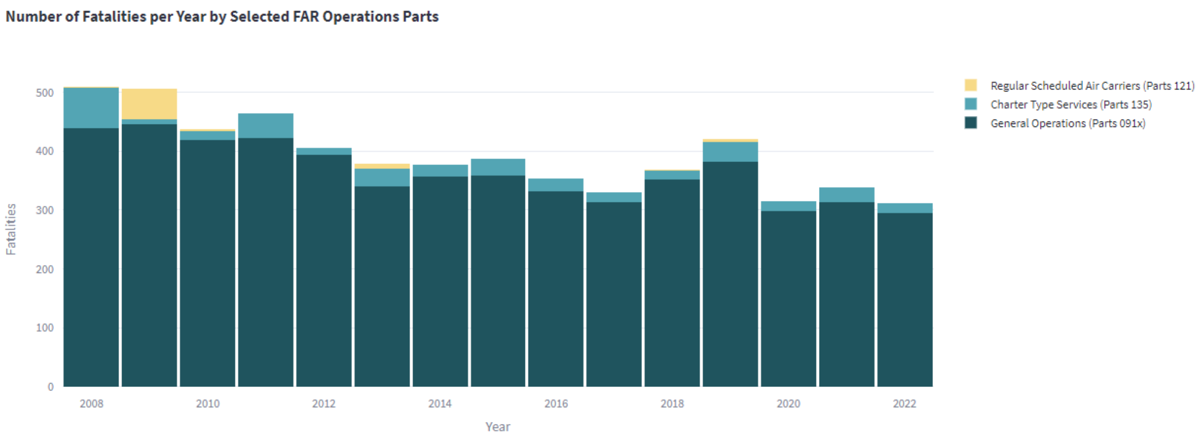

Figure 11: Bar chart representation of annual values

Total Number of Fatalities by Selected FAR Operations Parts

☐ User guide: Totals chart

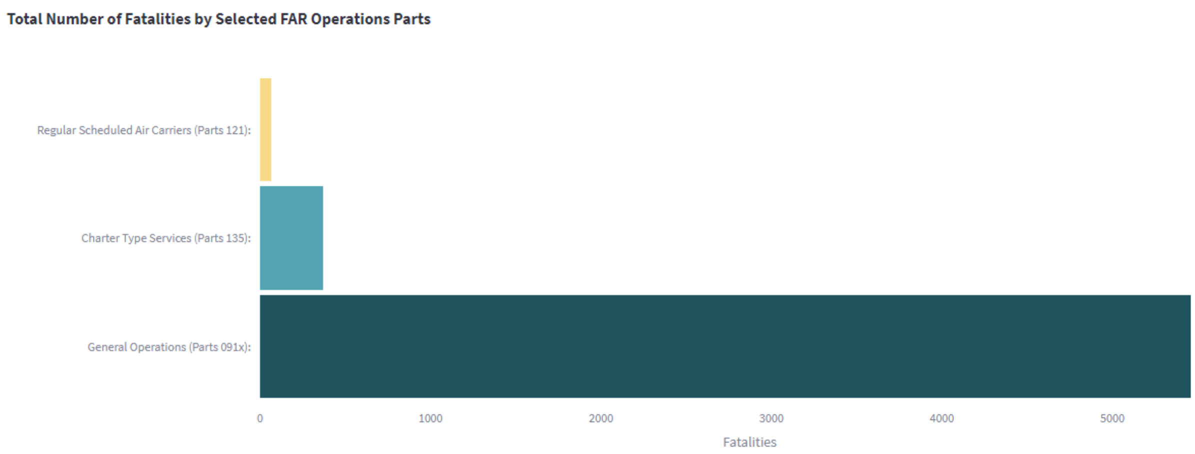

Figure 12: Horizontal bar chart representation of totals

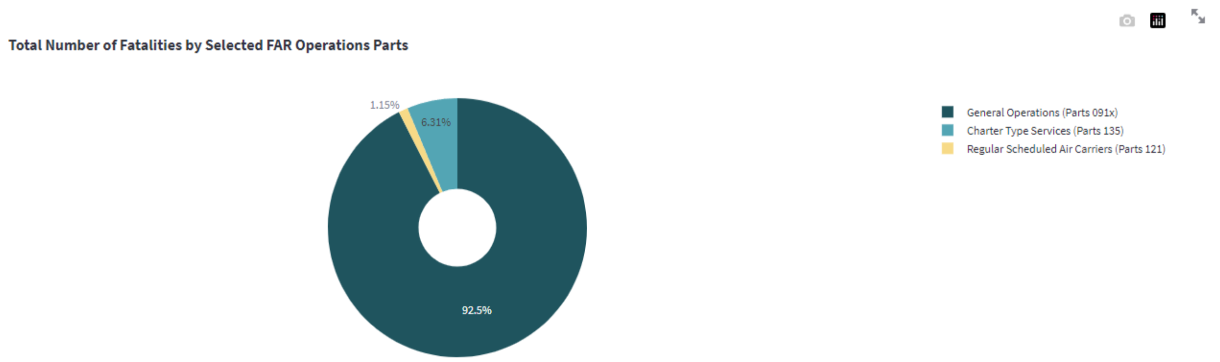

Figure 13: Pie chart representation of totals

6.5. Functional Control: Map

Figure 14: Functional control

The map shows the places where fatal aviation accidents occurred in the selected period. Each point represents at least one accident with fatalities. If you hover the mouse over such a point, you will get detailed information about the accident behind it. However, only those accidents can be shown here for which a decimal longitude and latitude have been entered. The map can be zooming in and out as needed. Furthermore, the map can also be printed out.

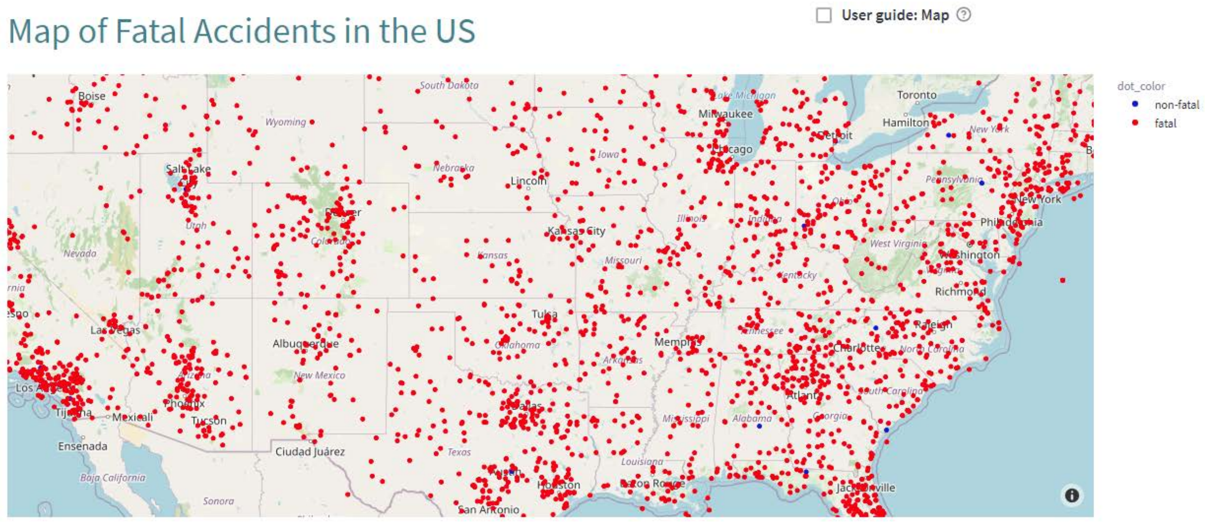

Figure 15: US map

7. Change Log

7.1. Release 23.05.01

- **FAA Airport file included:**
    - 2023.04.20_Airports.csv (20. April 2023)
- **FAA Runway file included:**
    - 2023.04.20_Runways.csv (20. April 2023)
- **NTSB files included:**
    - Pre2008.zip - 9/30/2020 12:51:56 PM
    - avall.zip - 5/ 1/2023 5:58:06 AM
    - up01MAY.zip - 5/ 1/2023 3:00:26 AM

7.2. Release 23.04.22

- **National Plan of Integrated Airport Systems (NPIAS) included:**
    - NPIAS-2023-2027-Appendix-A.xlsx
- **NTSB file included:**
    - Up22APR.zip - 4/22/2023 3:30 AM

7.3. Release 23.04.15

- **FAA Airports file included:**
    - 2023.02.23_Airports.csv (23. February 2023)
- **NTSB file included:**
    - Up15APR.zip - 4/15/2023 3:00:12 AM
- **simplemaps files included:**
    - simplemaps_uscities_basicv1.76
    - simplemaps_uszips_basicv1.82
- **United States Zip Codes.org file included:**
    - zip_code_database.xls (42735 entries)

7.4. Release 23.04.08

- **NTSB files included:**
    - Pre2008.zip - 9/30/2020 12:51:56 PM
    - avall.zip - 4/ 3/2023 8:13:22 AM
    - up08APR.zip - 4/ 8/2023 3:01:20 AM

7.5. Release 23.04.01

- **NTSB files included:**
    - Pre2008.zip - 9/30/2020 12:51:56 PM
    - avall.zip - 3/ 1/2023 6:01:23 AM
    - up08MAR.zip - 3/ 8/2023 3:00:22 AM
    - up15MAR.zip - 3/15/2023 3:00:11 AM
    - up22MAR.zip - 3/22/2023 3:00:14 AM
    - up01APR.zip - 4/ 1/2023 3:00:31 AM

8. References

data.ntsb.gov. (n.d.). MDB Download Directory - NTSB.ADMS.DataTransfer.Web. [online] Available at: https://data.ntsb.gov/avdata.

Hook, Loyd & Sizoo, David & Fuller, Justin. (2022). How Digital Safety Systems Could Revolutionize Aviation Safety. pp. 1-9. 10.1109/DASC55683.2022.9925863. Available at: https://www.researchgate.net/publication/365100080_How_Digital_Safety_Systems_Could_Revolutionize_Aviation_Safety

streamlit.io. (n.d.). *Streamlit • The fastest way to build and share data apps*. [online] Available at: https://streamlit.io/.