

IO-AVSTATS - User Manual Release 23.06.01

WORK IN PROGRESS

IO Aeronautical Autonomy Labs, LLC



© IO-Aero 2023

1.		Intro	oduction	5
2.		Gett	ing Started	6
	a.	A	pplication Access	6
	b.	G	eneral Usability	8
		i.	Data frame	8
3.		Арр	lication Association Rule Analysis	10
	a.	In	troduction	10
	b.	Pı	rocessing Options	10
		i.	Run the Algorithms	10
		ii.	Apriori Algorithm	. 11
		iii.	ECLAT Algorithm	12
		iv.	FP-Growth Algorithm	13
		V.	FP-Max Algorithm	. 14
		vi.	Item Selection	. 14
		vii.	Filtered Raw Data	. 15
		Fi	ltered Raw Data Profile	16
		viii.	Transaction Data	. 18
		ix.	Binary Data	. 19
		x.	One-Hot Encoded Data	. 20
		xi.	Frequent Item Sets	. 21
		xii.	Frequent Item Sets TreeMap	. 21
		xiii.	Association Rules	. 23
	c.	Fi	lter Events Sequence	. 23
	d.	Fi	lter Findings	. 24
	e.	Fi	lter Other Criteria	. 24
		i.	Aircraft categories	. 24
		ii.	Aircraft involved	. 25
4.		Арр	lication Aviation Event Analysis	26
	a.	In	rtroduction	26
	b.	Pı	rocessing Options	26
		i.	Run the Data Analysis	26
		ii.	Extended Version	27
		iii.	Map	27
		iv.	Data Graphs - Years	27

	Fatalities per Year by FAR Operations Parts	27
	Fatalities per Year by Selected FAR Operations Parts	27
	Events per Year by CICTT Codes	27
	Events per Year by Event Types	27
	Events per Year by Highest Injury Levels	27
	Events per Year by Main Phases of Flight	27
	Events per Year by Nearest Airport	27
	Events per Year by Phases of Flight	27
	Events per Year by Safety Systems	27
	Events per Year by Top Level Logical Parameters	27
٧	n. Data Graphs - Totals	27
	Total Fatalities by FAR Operations Parts	27
	Total Fatalities by Selected FAR Operations Parts	27
	Total Events by CICTT Codes	27
	Total Events by Event Types	27
	Total Events by Highest Injury Levels	27
	Total Events by Main Phases of Flight	27
	Total Events by Nearest Airport	27
	Total Events by Phases of Flight	27
	Total Events by Safety Systems	27
	Total Events by Top Level Logical Parameters	27
٧	vi. Data Graphs - Distances	27
	Distance to the Nearest Airport	27
٧	/ii. Data Profile	27
٧	/iii. Detailed Data	27
c.	Filter Options	27
i.	. Standard Version	27
ii	i. Extended Version	27
Δ	Application Database Profiling	28
a.	Introduction	28
b.	Processing Options	28
i.	. Data Profile	28
ii	i. Detailed Data	31
c.	Filter Options	31

5.

6.	Α	pplication US Aviation Fatal Accident Analysis	33
а		Introduction	33
b).	Processing Options	33
	i.	Map	33
	ii.	. Fatality-based Charts	34
		Selected FAR Operations Parts	35
	iii	i. Accident-based Charts	37
		Preventable Accidents by Safety Systems	38
c		Filter Options	40
7.	С	hange Log	41
а		Release 23.05.15	41
b).	Release 23.05.08	41
c		Release 23.05.01	41
C	l.	Release 23.04.22	41
e	١.	Release 23.04.15	41
f.		Release 23.04.08	42
g	.	Release 23.04.01	42
8	R	eferences	43

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.

Release 23.06.01 Page 5 of 43

2. Getting Started

a. 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.

Release 23.06.01 Page 6 of 43

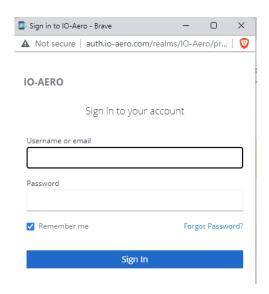


Figure 2: Authentication dialog

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.

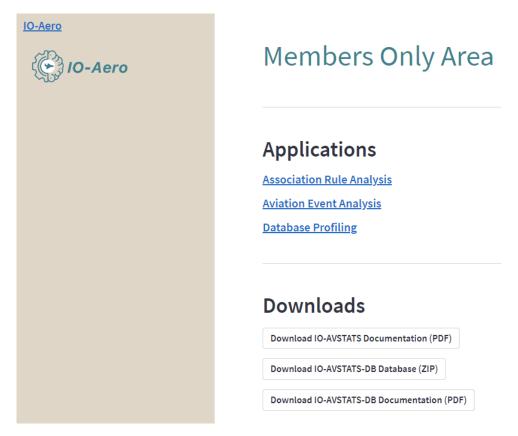


Figure 3: Sample menu in the members only area

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

Release 23.06.01 Page 7 of 43



Figure 4: Application standard controls top left

b. 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 is 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.

i. Data frame

A data frame contains detailed data in an interactive table format similar to MS Excel.

Release 23.06.01 Page 8 of 43



Figure 5: Data frame example

The following interactive options are available for data frames:

- Column resizing: resize columns by dragging and dropping column header borders.
- Column sorting: sort columns by clicking on their headers.
- **Copy to clipboard**: select one or multiple cells, copy them to clipboard, and paste them into your favorite spreadsheet software.
- **Search**: search through data by clicking a table, using hotkeys (\Re Cmd + F or Ctrl + F) to bring up the search bar, and using the search bar to filter data.
- Table (height, width) resizing: resize tables by dragging and dropping the bottom right corner of tables

As a rule, the data contained in the data frame can be downloaded to the local system as a CSV file if requested.

Release 23.06.01 Page 9 of 43

3. Application Association Rule Analysis

a. 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.

This application gives an opportunity to experiment with association rule analysis algorithms. The content of the "basket" can be filled either with the occurrence or finding codes of an event or their components.

b. Processing Options

i. Run the Algorithms

Streamlit-based applications immediately recalculate each time a parameter is changed. However, since the desired setting can consist of a whole row of parameters, this checkbox allows you to explicitly trigger the recalculation after all desired settings have been made.



Figure 6: Checkbox to run the algorithms

Release 23.06.01 Page 10 of 43

ii. Apriori Algorithm

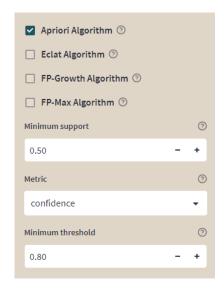


Figure 7: Settings for the Apriori algorithm

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

The <u>implementation</u> of the Apriori algorithm used here is from the MLXTEND library. The following parameters are available:

- **Minimum support**: A float between 0 and 1 for minimum support of the item sets returned. The support is computed as the fraction transactions_where_item(s)_occur / total_transactions.
- **Metric:** Metric to evaluate if a rule is of interest **A** stands for antecedent and **C** stands for consequent:
 - o support(A->C) = support(A+C) [aka 'support'], range: [0, 1]
 - confidence(A->C) = support(A+C) / support(A), range: [0, 1]
 - lift(A->C) = confidence(A->C) / support(C), range: [0, inf]
 - leverage(A->C) = support(A->C) support(A)*support(C), range: [-1, 1]
 - conviction = [1 support(C)] / [1 confidence(A->C)], range: [0, inf]
 - zhangs_metric(A->C) = leverage(A->C) / max(support(A->C)*(1-support(A)), support(A)*(support(C)-support(A->C))) range: [-1,1]
- **Minimum threshold**: Minimal threshold for the evaluation metric, via the metric parameter, to decide whether a candidate rule is of interest.

Release 23.06.01 Page 11 of 43

iii. ECLAT Algorithm

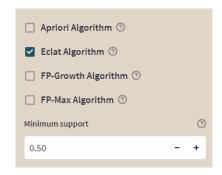


Figure 8: Settings for the ECLAT algorithm

The ECLAT (Equivalence Class Clustering and Bottom-Up Lattice Traversal) 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

The <u>implementation</u> of the ECLAT algorithm used here is from the pyECLAT library. The following parameter is available:

• Minimum support: A float between 0 and 1 for minimum support of the item sets returned. The support is computed as the fraction transactions_where_item(s)_occur / total_transactions.

Release 23.06.01 Page 12 of 43

iv. FP-Growth Algorithm

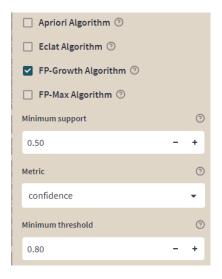


Figure 9: Settings for the FP-Growth algorithm

Here is a simple explanation of how the FP-Growth algorithm 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.

The <u>implementation</u> of the FP-Growth algorithm used here is from the MLXTEND library. The same parameters are available as for the Apriori algorithm.

Release 23.06.01 Page 13 of 43

v. FP-Max Algorithm

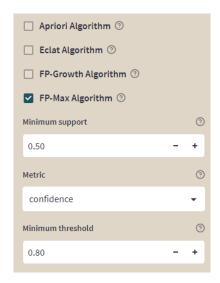


Figure 10: Settings for the FP-Max algorithm

The FP-Max 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.

The <u>implementation</u> of the FP-Max algorithm used here is from the MLXTEND library. The same parameters are available as for the Apriori algorithm.

vi. Item Selection

Here the content of the "basket" can be determined. Available are the occurrence codes or their components from the database table events_sequence as well as the finding codes or their components from the database table findings.

The following rules must be adhered to when making the selection:

- For database table events_sequence, either Occurrence code or any number of its components can be selected.
- For database table events_sequence, either All or any of Defining event or No defining event can be selected.
- For database table **findings**, either **Finding code** or any number of its components can be selected.

Release 23.06.01 Page 14 of 43

• For database table findings, either All or any of Cause, Factor Or Neither can be selected

Association Rule Analysis - Year 2008 until 2022

☐ Show Active Filter(s) ③	About this Appli	cation ③	☐ User Guide: Application ③		
Items from database ta	ble events_se	quence	User Guide: Ite	ems from events_sequence** ③	
Occurrence code:	✓ All (3	☐ Defining event ③	☐ No defining event ②	
Eventsoe no:	☐ All (3	☐ Defining event ③	☐ No defining event ②	
Phase no:	☐ All (3	☐ Defining event ③	☐ No defining event ③	
Items from database ta	ble findings		User Guide: Ite	ems from findings ③	
Finding code:	✓ All ③	☐ Cause ③	☐ Factor ⑦	☐ Neither ③	
Category no:	☐ All ⑦	☐ Cause ②	☐ Factor ⑦	☐ Neither ②	
Subcategory no:	☐ All ③	☐ Cause ②	☐ Factor ③	☐ Neither ⑦	
Section no:	☐ All ③	☐ Cause ②	☐ Factor ③	☐ Neither ③	
Subsection no:	☐ All ③	☐ Cause ③	☐ Factor ③	☐ Neither ③	
Modifier no:	☐ All ②	☐ Cause ②	☐ Factor ③	☐ Neither ③	

A finer selection of Occurrence code and Findings code can be made via their components at Filter Events Sequence and Filter Findings.

vii. Filtered Raw Data

With this checkbox the detailed raw data can be displayed.

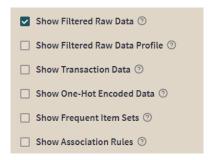
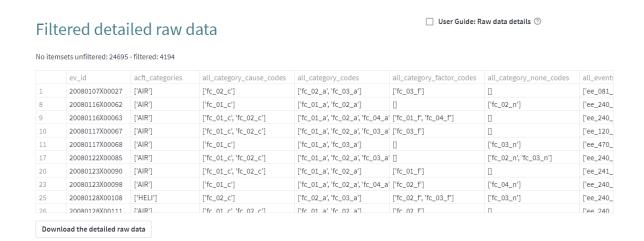


Figure 11: Checkbox to show the filtered raw data

The data is displayed in a table format, the so-called data frame. The data frame shows only the data that is still available after applying the selected filter options. The data in the data frame can also be downloaded as a CSV file using the 'Download the detailed raw data' button.

Release 23.06.01 Page 15 of 43



More detailed information on how to use the data frame can be found in section '2.b General Usability'.

Filtered Raw Data Profile

This checkbox triggers profiling of the filtered raw data and creating a report.

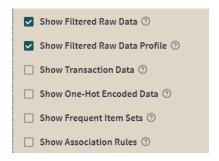


Figure 12: Checkbox to show the raw data profile

The Overview shows mostly global details about the dataset (number of records, number of variables, overall missingness and duplicates, memory footprint).

Profile of the filtered raw data

☐ User Guide: Raw data profile ②

Overview



Figure 13: Raw data profile summary

Release 23.06.01 Page 16 of 43

The Overview is followed by the Variables section, which contains the detailed information for each column. The Select Columns select box at the top allows you to select a specific column.

Variables

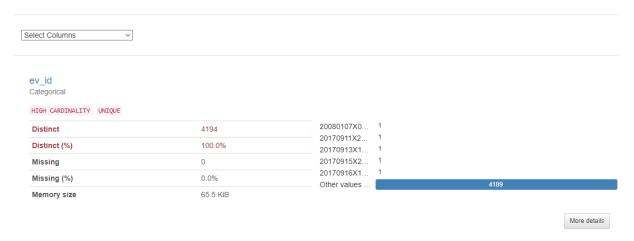


Figure 14: Simple view of column profile

More information about a specific column can be retrieved by clicking the More Details button. The additional information shown is located in the two tabs Overview and Categories.

Variables

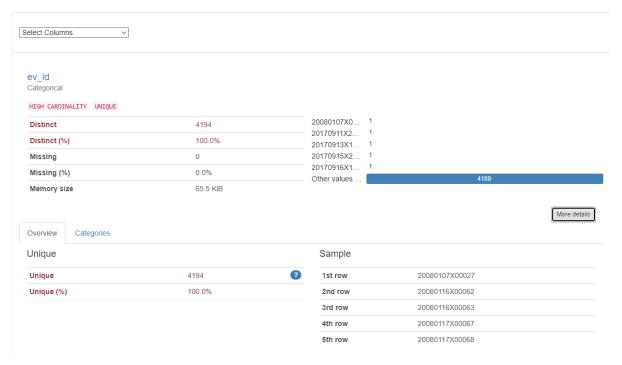


Figure 15: Extended view of column profile

Columns whose contents cannot be evaluated are shown as follows:

Release 23.06.01 Page 17 of 43



Figure 16: Example of an unsupported column

The Download the profile report button at the end of the report can be used to download an HTML version of the profiling report to the local computer.



Figure 17: Download button

Further details can be found in the <u>ydata-profiling documentation</u>.

viii. Transaction Data

XXXX

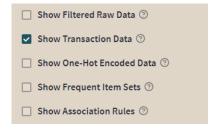


Figure 18: Checkbox to show the transaction data

XXX

Detailed transaction data Non-Eclat Algorithm

☐ User Guide: Transaction data details ③

	items
1	['eo_153081_a', 'ff_0204151039_a', 'ff_0303602099_a']
8	['eo_506240_a', 'eo_650470_a', 'ff_0106201020_a', 'ff_02031
9	['eo_350240_a', 'eo_350241_a', 'eo_350330_a', 'eo_650470_
10	['eo_502120_a', 'ff_0106201220_a', 'ff_0202202544_a', 'ff_03
11	['eo_450470_a', 'ff_0106201220_a', 'ff_0302101099_a']
17	['eo_350240_a', 'ff_0106202220_a', 'ff_0201302044_a', 'ff_02
20	['eo_500241_a', 'eo_500270_a', 'eo_650470_a', 'ff_01061035
23	['eo_402240_a', 'eo_650337_a', 'ff_0103000001_a', 'ff_01062
25	['eo_450240_a', 'eo_650470_a', 'ff_0206102044_a', 'ff_02063
26	['en 501240 a' 'en 650402 a' 'en 650470 a' 'ff 01062000

Release 23.06.01 Page 18 of 43

Detailed transaction data Eclat Algorithm

☐ User Guide: Transaction data details ②

No transactions: 4194

	0	1	2	3	4	5	6	7
0	eo_153081_a	ff_0204151039_a	ff_0303602099_a					
1	eo_506240_a	eo_650470_a	ff_0106201020_a	ff_0203100044_a				
2	eo_350240_a	eo_350241_a	eo_350330_a	eo_650470_a	ff_0103523013_a	ff_0106201020_a	ff_0206201541_a	ff_04021010
3	eo_502120_a	ff_0106201220_a	ff_0202202544_a	ff_0303602083_a	ff_0304203083_a			
4	eo_450470_a	ff_0106201220_a	ff_0302101099_a					
5	eo_350240_a	ff_0106202220_a	ff_0201302044_a	ff_0202201544_a	ff_0303602083_a			
6	eo_500241_a	eo_500270_a	eo_650470_a	ff_0106103508_a	ff_0106201020_a	ff_0206304044_a		
7	eo_402240_a	eo_650337_a	ff_0103000001_a	ff_0106200020_a	ff_0201201044_a	ff_0201201544_a	ff_0403201070_a	
8	eo_450240_a	eo_650470_a	ff_0206102044_a	ff_0206304044_a	ff_0302100091_a	ff_0303401599_a		
9	en 501240 a	en 650402 a	en 650470 a	ff 0106200020 a	ff 0201203044 a	ff 0204101544 a		

Download the detailed transaction data

ix. Binary Data

XXXX

XXX

☐ Show Filtered Raw Data ②
☐ Show Transaction Data ②
Show Binary Data ②
☐ Show Frequent Itemsets ②
☐ Show Frequent Itemsets TreeMap ③
☐ Show Association Rules ③

Release 23.06.01 Page 19 of 43

Binary data Eclat Algorithm

☐ User Guide: Binary data ②

No transactions: 4194 - no columns: 2508

	eo_403337_a		eo_553490_a	eo_350282_a	ff_0102256301_a	eo_402241_a	ff_0106204221_a	ff_0102340001_a	ff_010353470
0	0	1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0
5	0	1	0	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0	0
7	0	1	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0	0
9	0	1	0	0	0	0	0	n	n

Download the binary data

x. One-Hot Encoded Data

xxxx

☐ Show Filtered Raw Data ③
☐ Show Transaction Data ②
Show One-Hot Encoded Data ③
☐ Show Frequent Itemsets ③
☐ Show Association Rules ③

XXX

Detailed one-hot encoded data Non-Eclat Algorithm

☐ User Guide: Binary data one-hot encoded ③

No transactions: 4194 - no columns: 2507

	eo_100000_a	eo_100010_a	eo_100020_a	eo_100030_a	eo_100040_a	eo_100050_a	eo_100080_a	eo_100081_a	eo_100110_a	eo_100193_a
1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0

Download the one-hot encoded data

Release 23.06.01 Page 20 of 43

xi. Frequent Item Sets

XXXX

☐ Show Filtered Raw Data ②
☐ Show Transaction Data ③
☐ Show One-Hot Encoded Data ②
Show Frequent Itemsets ②
☐ Show Association Rules ③

XXX

Detailed frequent itemsets Eclat Algorithm

User Guide: Frequent itemset details ②

No frequent itemsets: 2507 - max. combinations: 23

	item	item_description	transactions
805	eo_650470_a	occurrence code Uncontroll	1382
1440	ff_0206304044_a	finding code Personnel issue	1381
941	ff_0204152044_a	finding code Personnel issue	734
471	ff_0106201020_a	finding code Aircraft - Aircraft	530
836	ff_0106200020_a	finding code Aircraft - Aircraft	454
1520	ff_0500000000_a	finding code Not determined	445
1786	ff_0106201220_a	finding code Aircraft - Aircraft	360
1914	eo_350240_a	occurrence code Initial climi	318
872	eo_450240_a	occurrence code Maneuverin	279
198	ff 0202202544 a	finding code Personnel issue	250

Download the frequent itemsets

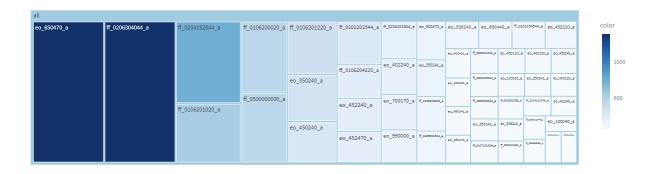
xii. Frequent Item Sets TreeMap

Release 23.06.01 Page 21 of 43

☐ Show Filtered Raw Data ②
☐ Show Transaction Data ③
☐ Show Binary Data ③
☐ Show Frequent Itemsets ②
Show Frequent Itemsets TreeMap ②
☐ Show Association Rules ②

Frequent itemsets tree map Eclat Algorithm

 $\hfill \square$ User Guide: Frequent itemsets tree map $\ensuremath{\mathfrak{D}}$



Detailed frequent itemsets Apriori Algorithm

No frequent itemsets: 7

	support	itemsets	itemsets_description
0	0.3295	frozenset({'eo_650470_a'})	['occurrence code Uncontrol
4	0.3293	frozenset({'ff_0206304044_a'	['finding code Personnel issu
6	0.1803	frozenset({'ff_0206304044_a',	['finding code Personnel issu
3	0.1750	frozenset({'ff_0204152044_a'	['finding code Personnel issu
2	0.1264	frozenset({'ff_0106201020_a'	['finding code Aircraft - Aircra
1	0.1082	frozenset({'ff_0106200020_a'	['finding code Aircraft - Aircra
5	0.1061	frozenset({'ff_0500000000_a'	['finding code Not determine

Download the frequent itemsets

Release 23.06.01 Page 22 of 43

xiii. Association Rules

XXXX

☐ Show Filtered Raw Data ③
☐ Show Transaction Data ③
☐ Show One-Hot Encoded Data ②
☐ Show Frequent Itemsets ③
Show Association Rules ③

XXX



Detailed association rules Eclat Algorithm

User Guide: Association rule details ③

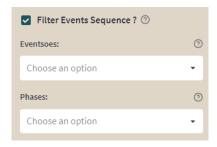
No association rules: 7

Download the association rules

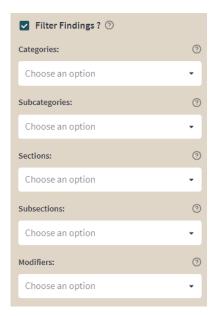
	Item	Item_description	Support
1	& eo_650470_a	['occurrence code Uncontro	0.3295
4	& ff_0206304044_a	['finding code Personnel issu	0.3293
6	eo_650470_a & ff_020630404	['occurrence code Uncontro	0.1803
3	& ff_0204152044_a	['finding code Personnel issu	0.1750
0	& ff_0106201020_a	['finding code Aircraft - Aircr	0.1264
2	&ff_0106200020_a	['finding code Aircraft - Aircr	0.1082
5	&ff_0500000000_a	['finding code Not determine	0.1061

c. Filter Events Sequence

Release 23.06.01 Page 23 of 43

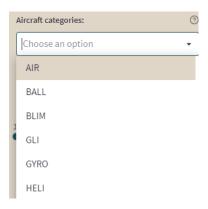


d. Filter Findings



e. Filter Other Criteria

i. Aircraft categories



Release 23.06.01 Page 24 of 43

ii. Aircraft involved



Release 23.06.01 Page 25 of 43

4. Application Aviation Event Analysis

a. 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.

- b. Processing Options
- i. Run the Data Analysis

XXX



Figure 19: Checkbox to run the data analysis

XXX

Release 23.06.01 Page 26 of 43

- ii. Extended Version
- iii. Map
- iv. Data Graphs-Years

Fatalities per Year by FAR Operations Parts

Fatalities per Year by Selected FAR Operations Parts

Events per Year by CICTT Codes

Events per Year by Event Types

Events per Year by Highest Injury Levels

Events per Year by Main Phases of Flight

Events per Year by Nearest Airport

Events per Year by Phases of Flight

Events per Year by Safety Systems

Events per Year by Top Level Logical Parameters

v. Data Graphs-Totals

Total Fatalities by FAR Operations Parts

Total Fatalities by Selected FAR Operations Parts

Total Events by CICTT Codes

Total Events by Event Types

Total Events by Highest Injury Levels

Total Events by Main Phases of Flight

Total Events by Nearest Airport

Total Events by Phases of Flight

Total Events by Safety Systems

Total Events by Top Level Logical Parameters

vi. Data Graphs- Distances

Distance to the Nearest Airport

- vii. Data Profile
- viii. Detailed Data
- c. Filter Options
- i. Standard Version
- ii. Extended Version

Release 23.06.01 Page 27 of 43

5. Application Database Profiling

a. Introduction



Figure 20: Header Database Profiling

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.

<u>ydata-profiling</u> is a Python library that generates an interactive report from a Pandas data frame. 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. ydata-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.

b. Processing Options

i. Data Profile

This checkbox triggers profiling of the previously selected database table or database view and creating a report.



Figure 21: Checkbox to show the data profile

The Overview shows mostly global details about the dataset (number of records, number of variables, overall missingness and duplicates, memory footprint).

Release 23.06.01 Page 28 of 43

Profiling Database Table events

☐ User Guide: Show data profile ②

Overview

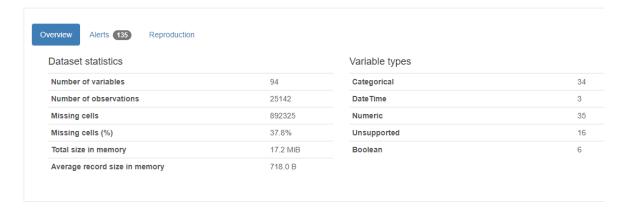


Figure 22: Data profile summary

The Overview is followed by the Variables section, which contains the detailed information for each column. The Select Columns select box at the top allows you to select a specific column.

Variables



Figure 23: Simple view of column profile

More information about a specific column can be retrieved by clicking the More Details button. The additional information shown is located in the two tabs Overview and Categories.

Release 23.06.01 Page 29 of 43

Variables

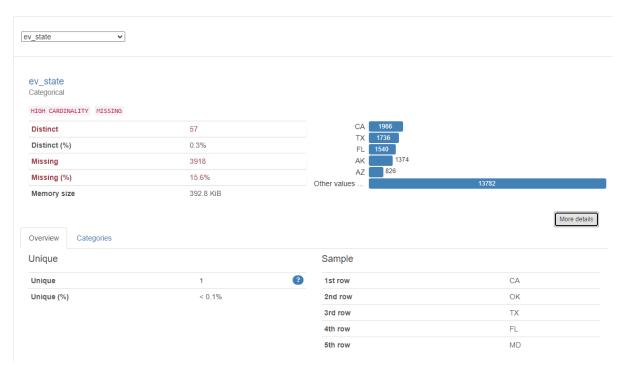


Figure 24: Extended view of column profile

Columns whose contents cannot be evaluated are shown as follows:



Figure 25: Example of an unsupported column

The Download the profile report button at the end of the report can be used to download an HTML version of the profiling report to the local computer.



Figure 26: Download button

Further details can be found in the ydata-profiling documentation.

Release 23.06.01 Page 30 of 43

ii. Detailed Data

With this checkbox the detailed data for the selected database table or view can be displayed.



Figure 27: Checkbox to show the detailed data

The data is displayed in a table format, the so-called data frame. The data frame shows only the data that is still available after applying the selected filter options. The data in the data frame can also be downloaded as a CSV file using the 'Download all data as CSV file' button.



Figure 28: Data frame

More detailed information on how to use the data frame can be found in section '2.b General Usability'.

c. Filter Options

Database tables and views: A database table or database view to be analyzed can be selected here. The default value is the database table event.



Figure 29: Radio button to select a database table or database view

Release 23.06.01 Page 31 of 43

Event year(s): A range of years between 1982 and today can be selected here via a slider widget. However, the selection made here only has an effect on those database tables and database views that either contain a column ev_year or contain a foreign key to the database table event.



Figure 30: Slider to select an interval of event years

Release 23.06.01 Page 32 of 43

6. Application US Aviation Fatal Accident Analysis

a. 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.

b. Processing Options

i. Map



Figure 31: Functional control

Release 23.06.01 Page 33 of 43

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 32: US map

ii. 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.

Release 23.06.01 Page 34 of 43

Selected FAR Operations Parts

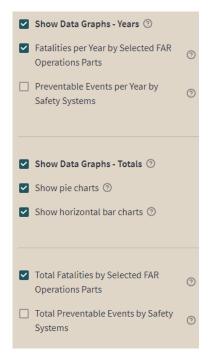


Figure 33: Functional controls

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.

Release 23.06.01 Page 35 of 43

Number of Fatalities per Year by Selected FAR Operations Parts

☐ User guide: Years chart ②

Number of Fatalities per Year by Selected FAR Operations Parts

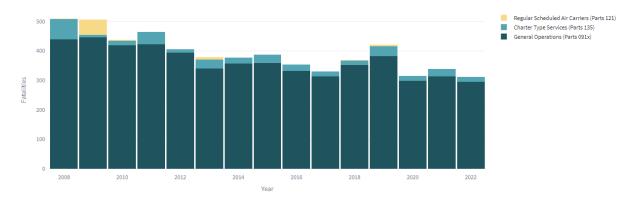


Figure 34: Bar chart representation of annual values

Total Number of Fatalities by Selected FAR Operations Parts

☐ User guide: Totals chart ③

Total Number of Fatalities by Selected FAR Operations Parts

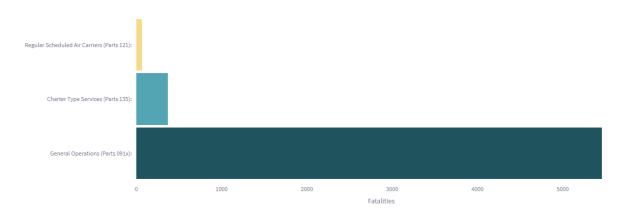


Figure 35: Horizontal bar chart representation of totals

Release 23.06.01 Page 36 of 43

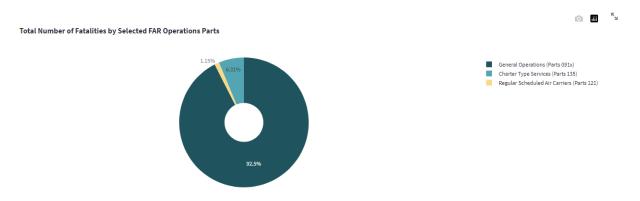


Figure 36: Pie chart representation of totals

iii. 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.

Release 23.06.01 Page 37 of 43

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.

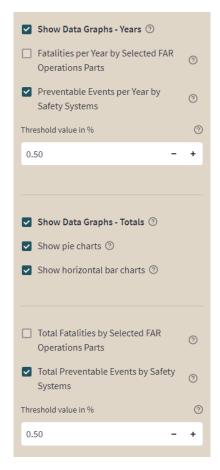


Figure 37: Functional controls

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	<pre>is_rss_forced_landing i.e., is_attitude controllable and is_emergency_landing</pre>
•	Spin / stall	is_rss_spin_stall_prevention_and_recovery i.e.,
		is_attitude_controllable and is_spin_stall
•	Terrain collision	<pre>is_rss_terrain_collision_avoidance i.e.,</pre>
		${ t is_attitude_controllable}$ and ${ t is_altitude_low}$ 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.

Release 23.06.01 Page 38 of 43

Number of Preventable Accidents per Year by Safety Systems

☐ User guide: Years chart ③

Number of Preventable Accidents per Year by Safety Systems

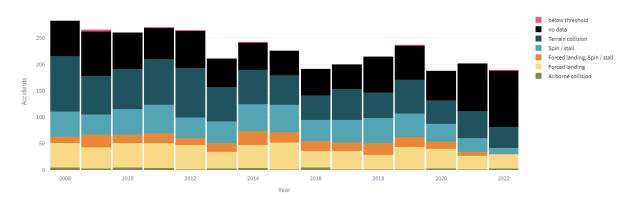


Figure 38: Bar chart representation of annual values

Total Number of Preventable Accidents by Safety Systems

☐ User guide: Totals chart ③



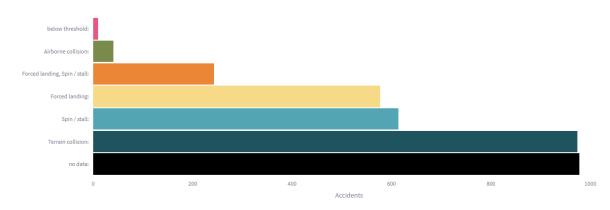


Figure 39: Horizontal bar chart representation of totals

Release 23.06.01 Page 39 of 43

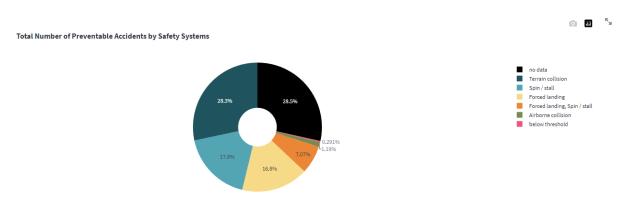


Figure 40: Pie chart representation of totals

c. Filter Options

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



Figure 41: Slider to select an interval of event years

Release 23.06.01 Page 40 of 43

7. Change Log

```
a. Release 23.06.01
```

• NTSB files included:

```
o avall.zip - 6/ 1/2023 6:07:07 AM o up01JUN.zip - 6/ 1/2023 3:00:19 AM
```

- ZIP Code Database file included:
 - o June 2023
- b. Release 23.05.22
- NTSB file included:

```
o up22MAY.zip - 5/22/2023 3:00:10 AM
```

- c. Release 23.05.15
- NTSB file included:

```
o up15MAY.zip - 5/15/2023 3:00:19 AM
```

- d. Release 23.05.08
- NTSB file included:

```
o up08MAY.zip - 5/ 8/2023 3:00:17 AM
```

- e. Release 23.05.01
- FAA Airport file included:

```
o 2023.04.20_Airports.csv (20. April 2023)
```

- FAA Runway file included:
 - o 2023.04.20_Runways.csv (20. April 2023)
- NTSB files included:
 - o Pre2008.zip 9/30/2020 12:51:56 PM
 - o avall.zip 5/ 1/2023 5:58:06 AM
 - o up01MAY.zip 5/ 1/2023 3:00:26 AM
- f. Release 23.04.22
- National Plan of Integrated Airport Systems (NPIAS) included:
 - o NPIAS-2023-2027-Appendix-A.xlsx
- NTSB file included:
 - o Up22APR.zip 4/22/2023 3:30 AM
- g. Release 23.04.15
- FAA Airports file included:
 - o 2023.02.23_Airports.csv (23. February 2023)
- NTSB file included:
 - o Up15APR.zip 4/15/2023 3:00:12 AM

Release 23.06.01 Page 41 of 43

- simplemaps files included:
 - o simplemaps_uscities_basicv1.76
 - o simplemaps_uszips_basicv1.82
- United States Zip Codes.org file included:
 - o zip_code_database.xls (42735 entries)

h. Release 23.04.08

- NTSB files included:
 - o Pre2008.zip 9/30/2020 12:51:56 PM
 - o avall.zip 4/ 3/2023 8:13:22 AM
 - o up08APR.zip 4/ 8/2023 3:01:20 AM
- i. Release 23.04.01
- NTSB files included:
 - o Pre2008.zip 9/30/2020 12:51:56 PM
 - o avall.zip 3/ 1/2023 6:01:23 AM
 - o up08MAR.zip 3/ 8/2023 3:00:22 AM
 - o up15MAR.zip 3/15/2023 3:00:11 AM
 - o up22MAR.zip 3/22/2023 3:00:14 AM
 - o up01APR.zip 4/ 1/2023 3:00:31 AM

Release 23.06.01 Page 42 of 43

8. References

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

Dias, J.R. (2023). pyECLAT. [online] GitHub. Available at:

https://github.com/jeffrichardchemistry/pyECLAT [Accessed 14 May 2023].

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 Revolut ionize Aviation Safety

rasbt.github.io. (n.d.). Apriori - mlxtend. [online] Available at:

https://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/ [Accessed 14 May 2023].

rasbt.github.io. (n.d.). Fpgrowth - mlxtend. [online] Available at:

https://rasbt.github.io/mlxtend/user_guide/frequent_patterns/fpgrowth/ [Accessed 14 May 2023].

rasbt.github.io. (n.d.). Fpmax - mlxtend. [online] Available at:

https://rasbt.github.io/mlxtend/user_guide/frequent_patterns/fpmax/ [Accessed 14 May 2023].

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

Release 23.06.01 Page 43 of 43