

A Tool for Automated Identification and Reconstruction of Special Physical Phenomena

Yu (Claire) Chen

Center for Space Plasma and Aeronomics Research

The University of Alabama in Huntsville

07/08/2024

Sun



Spacecraft



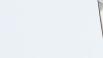
Earth

From Data to Event Identification

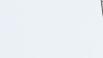
Daily Example

For a smoothie store without CCTV where we only have records of smoothie ingredient usages, how do we know if any strawberry smoothies were being made during a specific time period?

Theoretically Amount Needed

 Strawberries: 150g	 Milk: 120 ml	 Ice Cubes: 4-5 cubes
 Banana: 100g	 Honey: 1-2 tsps	
 Yogurt: 125g	 Blender: Must have	

Recorded Usage 12:00-12:10 1/1/2024

 Strawberries: 152g	 Milk: 125 ml	 Ice Cubes: 3 cubes
 Banana: 102g	 Honey: 1 tsp	
 Yogurt: 123g	 Blender: Used	

Event Confirmation & Follow-up

- Recorded usage ≈ Theoretical amount needed
 - Very likely for a strawberry smoothie to be made during 12:00-12:10 on 1/1/2024
- Can derive additional/nutritional values as follow-up:
 - protein content, sweetness level, etc.

From Data to Event Identification

Automated Tool in This Project

Automatically extract & process data, identify special physical phenomena (events), conduct data analyses, derive event characteristics, and output files.

Theoretically Amount Needed

- 🍓 Strawberries: 150g
- 🍌 Banana: 100g
- ...

Targeted events follow theoretical patterns, which can be described by a series of complex formulas

Theoretical Prediction

Recorded Usage 12:00-12:10 1/1/2024

- 🍓 Strawberries: 152g
- 🍌 Banana: 102g
- ...

Raw Data 📈: few TB+, i.e., equivalent to databases.
Processing: Take segments only and transform to high-level data products

Processed Observational Data

Event Confirmation & Follow-up

- Recorded usage ≈ Theoretical...
- Very likely for a strawberry...
- Can derive additional values...
- protein content, etc.

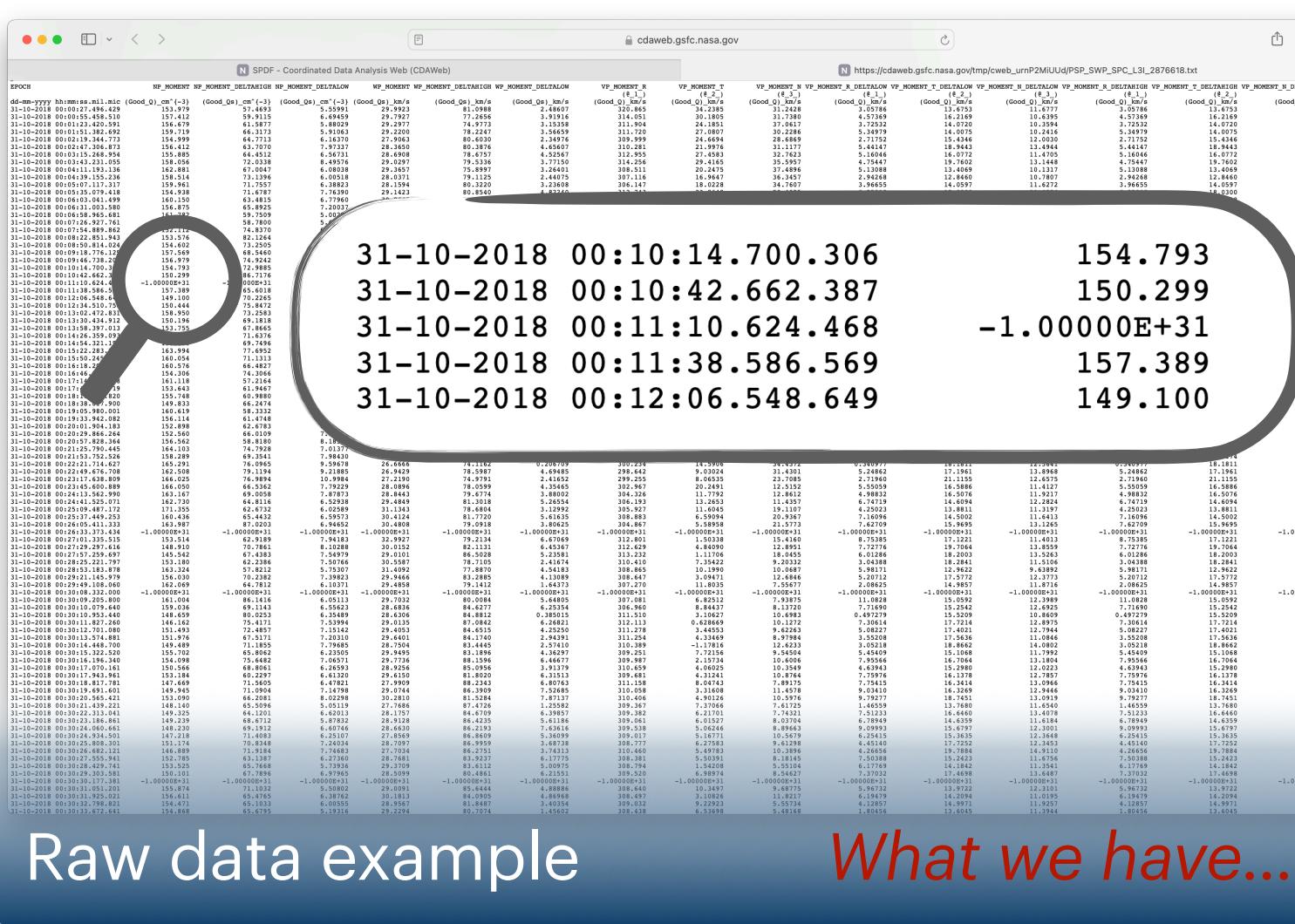
Event identification by checking Observation ≈ Theoretical prediction?
Derive parameters for each identified event & build **database**

Event Identification & Database

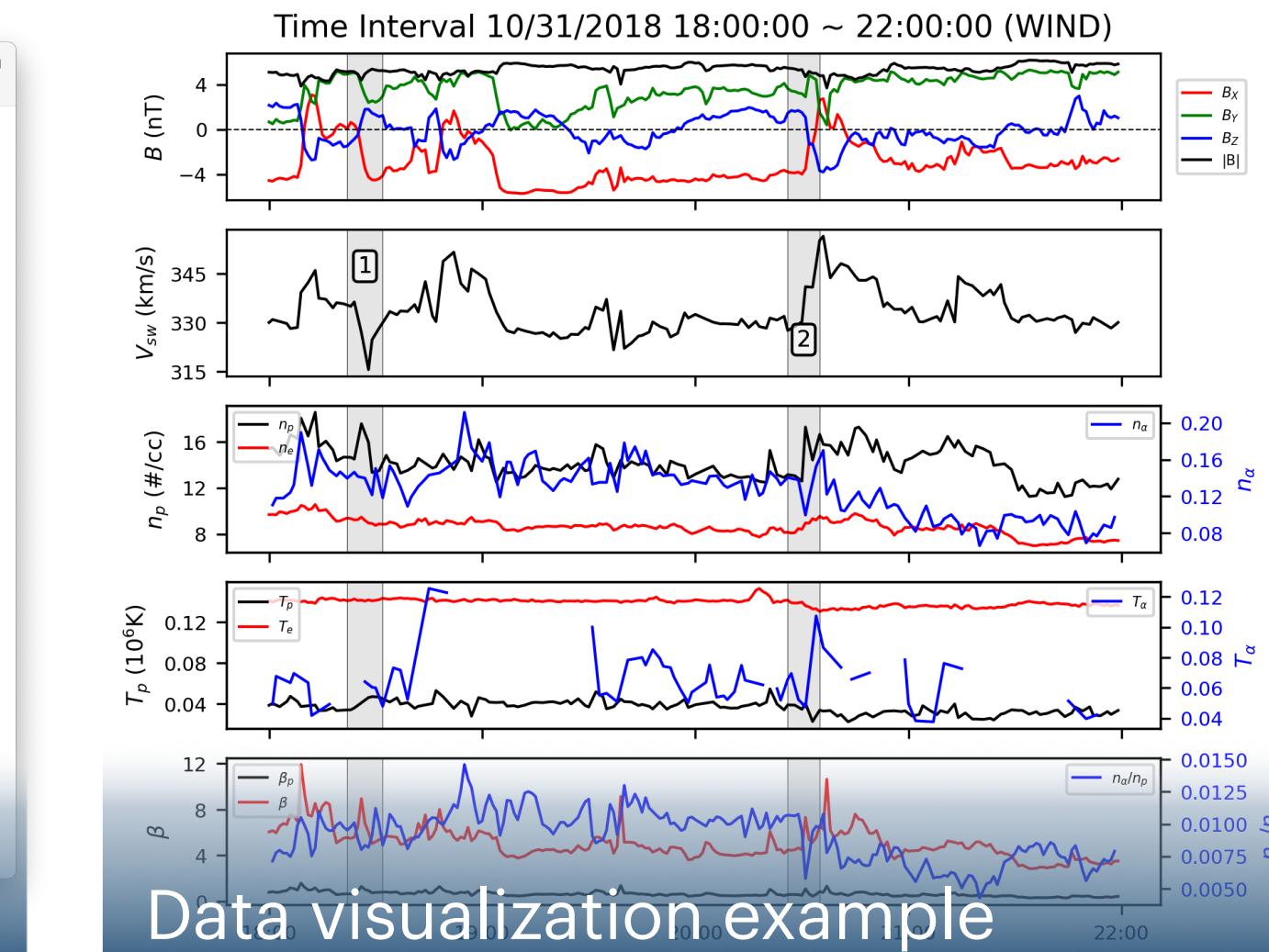
What can we get with this tool?

`detection(rootDir, spacecraftID='WIND', timeStart=datetime(2018,10,31,18,0,0), timeEnd=datetime(2018,10,31,22,0,0), duration=(10,30), includeTe=True, includeNe=False, Search=True, CombineRawResult=True, GetMoreInfo=True, LabelFluxRope=True, B_mag_threshold=5.0, shockList_DF_path=shockList, allowIntvOverlap=False)`

👉 What we use...



Time-series spacecraft data with outliers.
Data volumes depend on length of time period and data resolution.



Automatically generate 20+ different figures for comprehensive analysis.

Identified events example

What we want...

The final list including 2 identified events.
Proactive step: Each entry is supplied with ~100 derived parameters.

Automated Tool: Development

Major Functions: pull & process data, identify special events, extract characteristics, and output files.

Choices of Programming

Data Extract, Transform, & Load

Data Performance & Data Security

Automation Settings

Documentation

Version Control

1. Choices About Programming/Software:

Software/Programming	Pros	Cons
Python	Open source and free; resourceful community providing multiple tools	Requires learning and manual installation; not all tools are reliable
Matlab	Powerful for mathematical computations	License is not free; very slow for large-scale calculations
Excel	Widely used for daily tasks	Limited capacity and slow for large files; unable for complex 2D & 3D computations

Final Choices:

- Python: all data processing steps, calculations, analytics, and visualization
- Excel: share event lists per user requests
- SQL (PostgreSQL): for efficient data storage and database management (internal use)

Automated Tool: Development

Major Functions: pull & process data, identify special events, extract characteristics, and output files.

Choices of Programming

Data Extract, Transform, & Load

Data Performance & Data Security

Automation Settings

Documentation

Version Control

2. Data Extract, Transform, & Load (ETL) Pipelines:

- Data Extraction: extract data from sources (database, file, and online website, etc.)
 - This tool: Python + supported tool to extract data segments from data archives
- Data Transformation/Processing: prepare data for further analysis
 - This tool: Common transformations (1-4)

1. Convert Data Types

1.02 blender to 1 blender,
convert float to integer

2. Drop Duplicates

One & only one record per
timestamp, remove repeated

3. Identify & Clean Outliers

Beyond expected range...
Data integrity issue

4. Derive New Products

E.g., Pressure = Density *
Temperature * Factor

3.1 Inserting NaN/Null

Have records at 12:00 & 12:10 only
=> Insert NaN/Null at 12:05

3.2 Interpolation

Used 2 🍩 from 12:00 to 12:10
=> Possibly 1 🍩 by the middle at 12:05

3.3 Downsampling

Have records at 12:00 & 12:10 only
=> Analyze broader intervals > 10 mins

3.4 Replacement

Honey is out of stock, replace it
with sweetener

Automated Tool: Development

Major Functions: pull & process data, identify special events, extract characteristics, and output files.

Choices of Programming

Data Extract, Transform, & Load

Data Performance & Data Security

Automation Settings

Documentation

Version Control

2. Data Extract, Transform, & Load (ETL) Pipelines:

- Data Extraction: extract data from sources (database, file, and online website, etc.)
 - This tool: Python + supported tool to extract data segments from data archives
- Data Transformation/Processing: prepare data for further analysis
 - This tool: Common transformations (1-4)

1. Convert Data Types

2. Drop Duplicates

3. Identify & Clean Outliers

4. Derive New Products

- Data Loading: load transformed/processed data into databases or saved as files
 - This tool: output final event lists as CSV files to share with the others and/or load into databases

3. Data Performance:

- This tool: establish checkpoints to check efficiency, accuracy, etc. & show warnings in execution window

4. Data Security:

- This tool: N/A since files and results are intended to be public, e.g., publish identified events to online database
- Feasible if needs to be confidential: add access limit, require users to login, etc.

Automated Tool: Development

Major Functions: pull & process data, identify special events, extract characteristics, and output files.

Choices of Programming

Data Extract, Transform, & Load

Data Performance & Data Security

Automation Settings

Documentation

Version Control

2. Data Extract, Transform, & Load (ETL) Pipelines:

- Data Extraction: extract data from sources (database, file, and online website, etc.)
 - This tool: Python + supported tool to extract data segments from data archives
- Data Transformation/Processing: prepare data for further analysis
 - This tool: Common transformations (1-4)

1. Convert Data Types

2. Drop Duplicates

3. Identify & Clean Outliers

4. Derive New Products

- Data Loading: load transformed/processed data into databases or saved as files
 - This tool: output final event lists as CSV files to share with the others and/or load into databases

3. Data Performance:

- This tool: establish checkpoints to check efficiency, accuracy, etc. & show warnings in execution window

4. Data Security:

- This tool: N/A since files and results are intended to be public, e.g., publish identified events to online database
- Feasible if needs to be confidential: add access limit, require users to login, etc.

Automated Tool: Development

Major Functions: pull & process data, identify special events, extract characteristics, and output files.

Choices of Programming

Data Extract, Transform, & Load

Data Performance & Data Security

Automation Settings

Documentation

Version Control

5. Automation Settings:

```
detection(rootDir, spacecraftID='WIND', timeStart=datetime(2018,10,31,18,0,0), timeEnd=datetime(2018,10,31,22,0,0),  
duration=(10,30), includeTe=True, includeNe=False, Search=True, CombineRawResult=True, GetMoreInfo=True,  
LabelFluxRope=True, B_mag_threshold=5.0, shockList_DF_path=shockList, allowIntvOverlap=False)
```

- **With an automated tool:** only need to run 1~3 lines to get final results & figures
- **Optional automation settings**
 - Controller: syntax like `Search=True`
 - Setting it to be `True`: activate the corresponding modules
 - Setting it to be `False`: skip the corresponding modules or finish this round
 - Different combinations result in different outputs — ensuring flexible usage to meet different needs
 - Users also opt to edit pre-written script files — customizing adjustments and extending

Automated Tool: Development

Major Functions: pull & process data, identify special events, extract characteristics, and output files.

Choices of Programming

Data Extract, Transform, & Load

Data Performance & Data Security

Automation Settings

Documentation

Version Control

6. Documentation: guide users on how to use the tool and understand its functions

- Documentation & instructions within code files & on GitHub (left two figures); video introduction available
- Built-in real-time tips & guidance on Python execution window (right figure)

The screenshot shows a GitHub repository page for 'PyGSDR / PyGS'. A file named 'instruction_gsr_examples.md' is selected. The content of the file is a step-by-step instruction for Grad-Shafranov Reconstruction (GSR). It includes a code snippet for the 'reconstruction' function and a note at the bottom: "Note: It may be better to copy all these lines into a script file and run python3 script.py".

This screenshot is similar to the one above, showing the same README file. An 'Outline' panel is open on the right side, listing sections such as '1. Initializing', '2. Obtaining the flux rope axis', etc. Below the outline, there is some explanatory text and a code block.

This screenshot shows a terminal window titled 'PyGS --zsh -- 86x40'. It displays a command-line interface for the GSR reconstruction. The user has typed 'reconstruction(...)' and is prompted to 'Start the reconstruction...'. The terminal then lists various parameters for the Grad-Shafranov reconstruction, such as polynomial order (3), grid points (15, 131), and boundary selection (True). It also shows results like maximum axial magnetic field (7.403238594695631 nT) and estimated axial current (64515916.57123201 A). At the bottom, it says 'Done.' followed by the user's name and the command.

Automated Tool: Development

Major Functions: pull & process data, identify special events, extract characteristics, and output files.

Choices of Programming

Data Extract, Transform, & Load

Data Performance & Data Security

Automation Settings

Documentation

Version Control

7. Version Control: Track changes, allow reversion to previous states, and backup

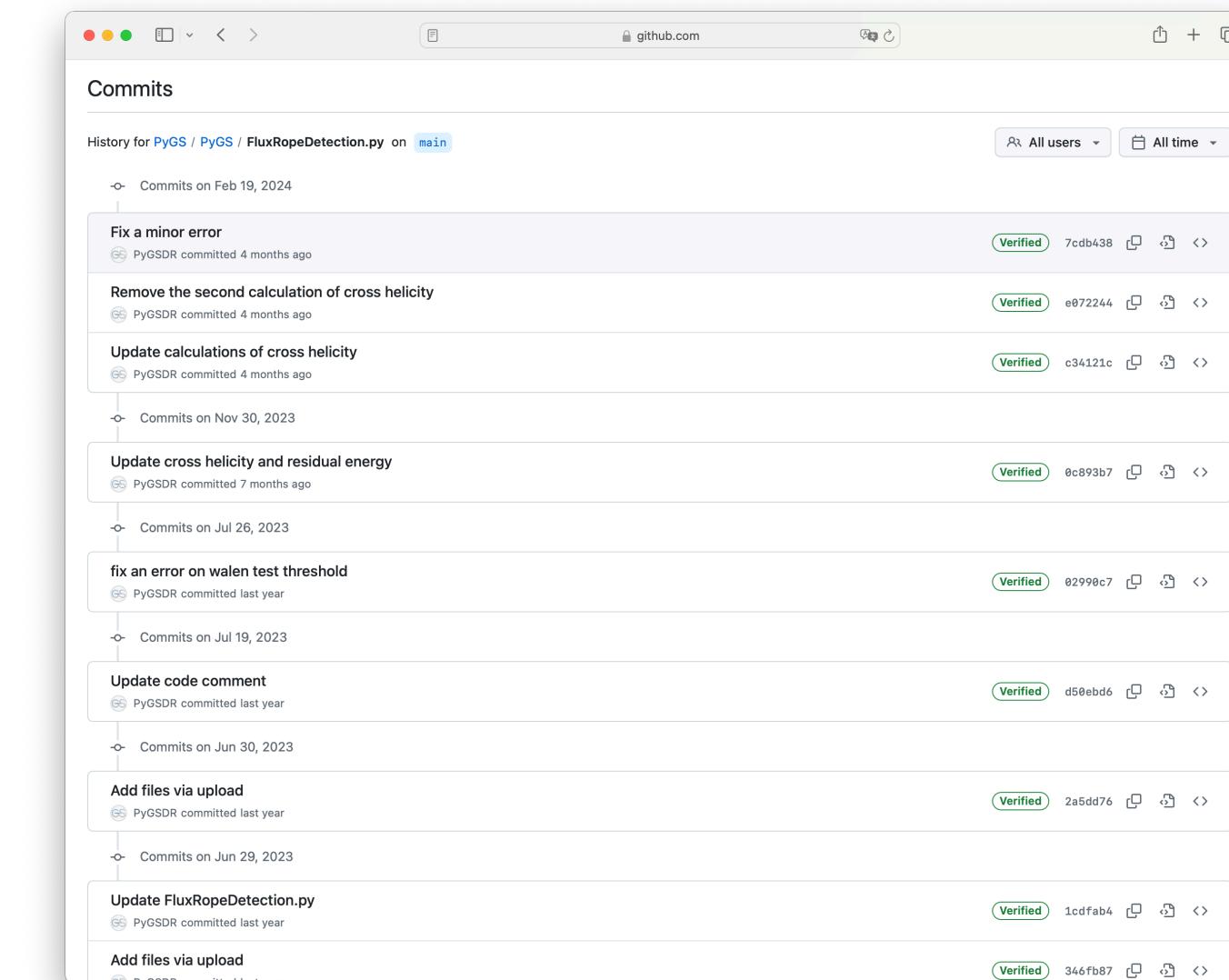
- Before launching: manage local files & document each version change with a brief description (middle figure)
- After launching: from 0.0.1 (beta/public test) to 1.0.0 (official); changes are available on GitHub (right figure)

Software Version

“x.y.z”

- x: major update
- y: minor update
 - e.g., adding new features
- z: fix bugs
 - generally do not impact the major use

```
25
26 2017-02-07
27 version 2.1
28 1) Combine three version into one.(For macbook, blueshark, and bladerunner)
29 2) Remove the command to get Np.DataFrame, Np is not used in this code.
30 3) Create and destroy pool in inner for loop. Since we want to save result in file when we finish one
31 4) Add searching time range printing in searchFluxRopeInWindow() function to indicate procedure.
32
33 2017-02-07
34 version 2.2
35 1) Add one more command line argument to specify the time range list. Command format is:
36     python GS_detectFluxRope_multiprocessing_v2.2.py 1996 '((20,30),(30,40),(40,50))'
37
38 2017-02-09
39 version 2.3
40 1) In version 2.2, we used a outer for loop to iterate duration range, and a inner for loop to iterate
41
42 2017-02-10
43 version 2.4
44 1) Change the residue calculating formula. Divided by N within the square root.
45
46 2017-02-11
47 version 3.0
48 1) A new major version. Retrun more informations.
49
50 2017-06-10
51 version 3.1
52 1) In this version, we change the A value smoothing method. Firstly, downsample A to 20 points, then ap
53 2) Improve sliding window generating method.
54 3) Use a new duration tuple specify method. User provide the min and max duration, and window size ran
55
56 2019-10-01
57 version 3.1.1
58 1) Apply the full expression for Pt = NKT + Bz^2/2mu0.
59
60 2020-04
61 version 3.1.2
62 1) Apply the extended GS-based equation (Teh 2018, EP&S), which has new Pt'(A'):
63     A'(x,θ) ~ -(1-a)By, a = MA^2
64     Pt' = (1-a)^2 * Bz^2/2mu0 + (1-a)p + a(1-a)B^2/2mu0
65
66 2020-04
67 version 3.1.4
68 1) Add Te, such that p = npk(Tp + Te)
```



Automated Tool: Post Evaluation & Future Work

Did we meet our objectives?

- This is a NASA-funded project, and thus needs to adhere to our proposal
- We delivered the tool/package 5 months earlier than deadline
- Met the community's evaluating criteria and was listed on website

Transferable to Data Management Role at Columbia University

Does the tool perform well?

- All-in-one package to conduct various analyses - convenient
- Successfully run on different OS/environments

- Keep key elements such as automated ETL

What has the user feedback been like?

- Before the public testing: tested by the internal users whose feedbacks helped refine the tool
- User-friendly features, flexibility, & support contributions
- Positive feedbacks & our work has been acknowledged in several journal publications

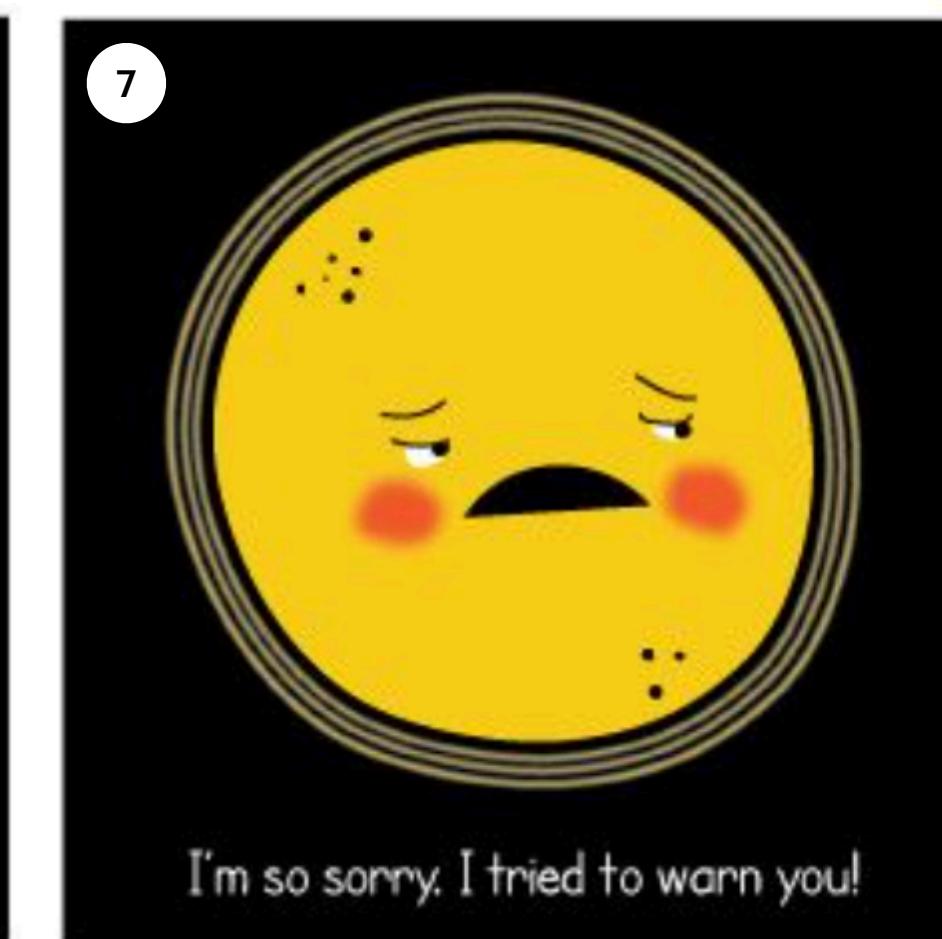
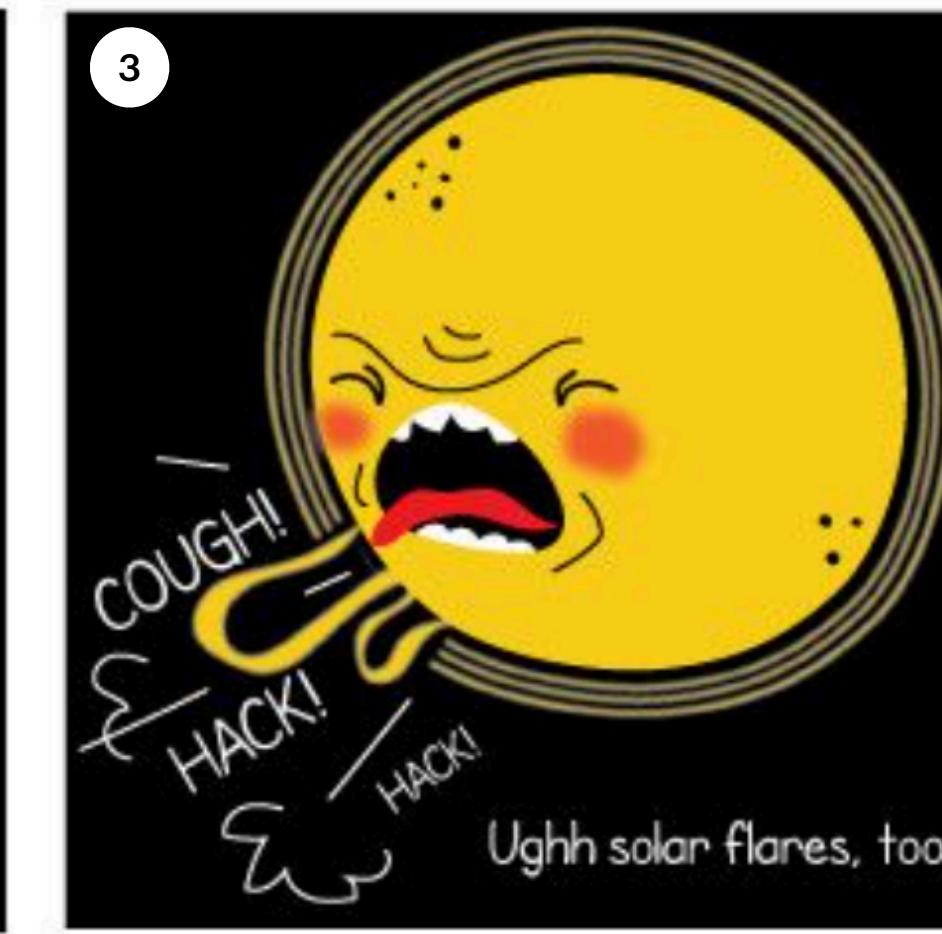
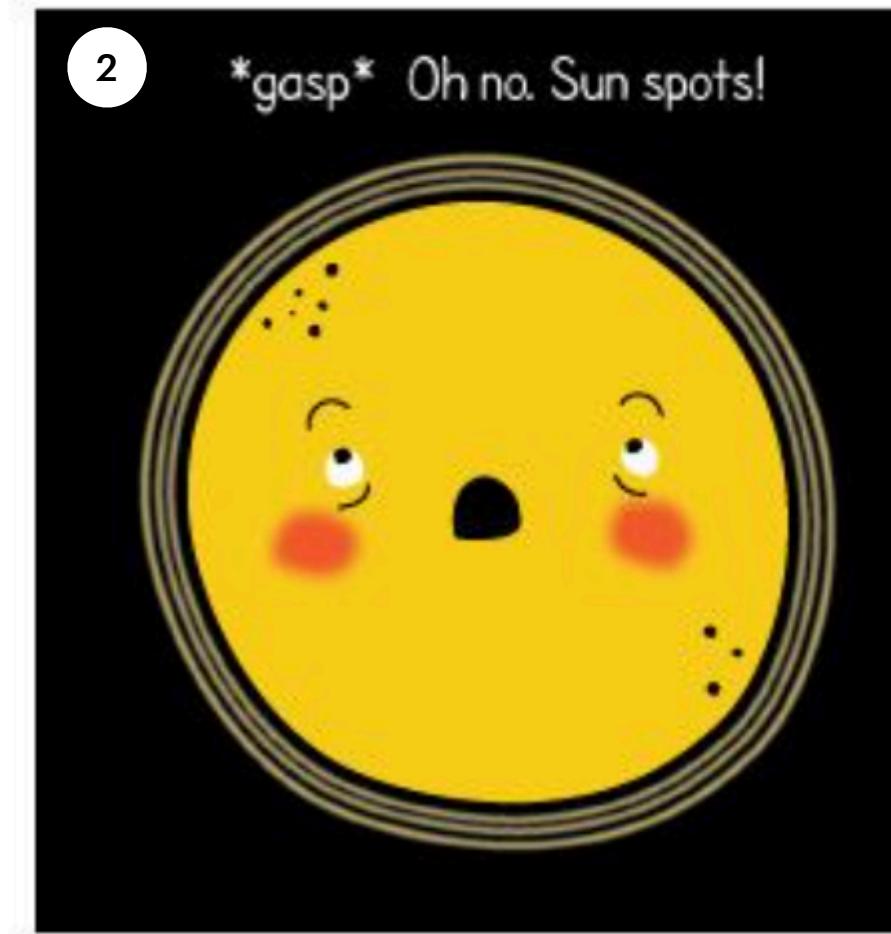
- Proactive approach to minimize frequent data requests
 - Different strategies for non-technical users such as website request

Possible Next Steps

- Maintenance & Add new features if time permits

Possible Next Steps

QUARK! COMICS



©2015 Hadria Beth Hermle

Thanks!