The **theft** package for R

Trent Henderson | 16 July 2022

Today's talk

01 Software landscape

02 The theft package for R

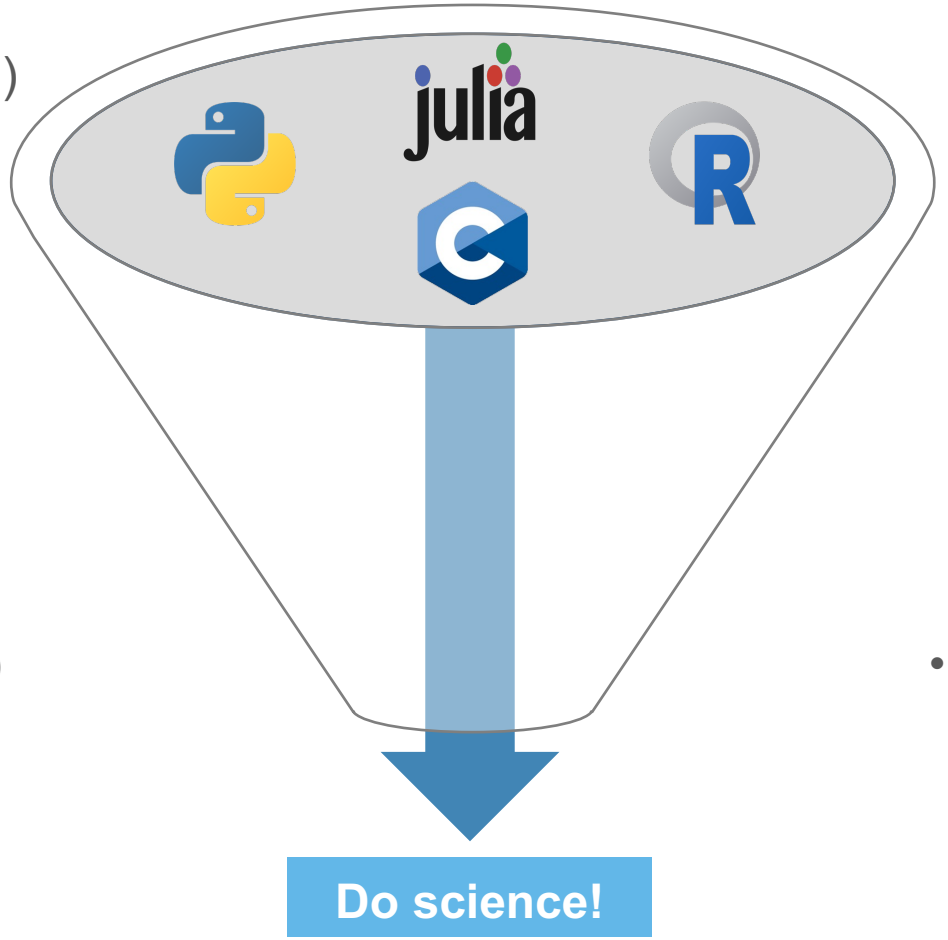
03 Future directions

04 Post-talk tutorial

Software landscape

Numerous open-source packages for feature extraction exist…

- **catch22** (pycatch22)
- **Kats**
- **TSFEL**
- **tsfresh**

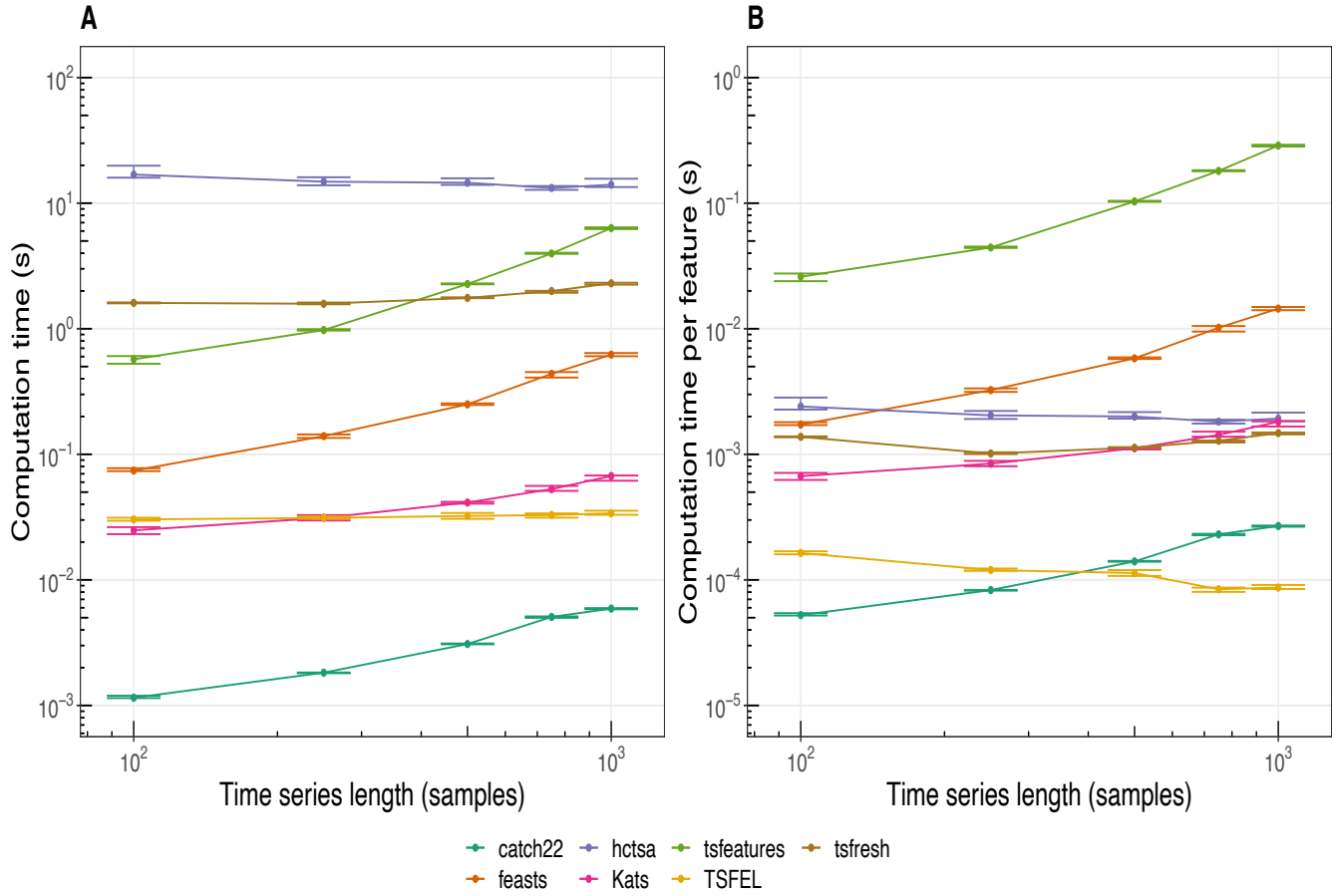


- **catch22**

- **catch22** (Rcatch22)
- **feasts**
- **tsfeatures**

- **catch22** (Catch22.jl)

…And they differ considerably…

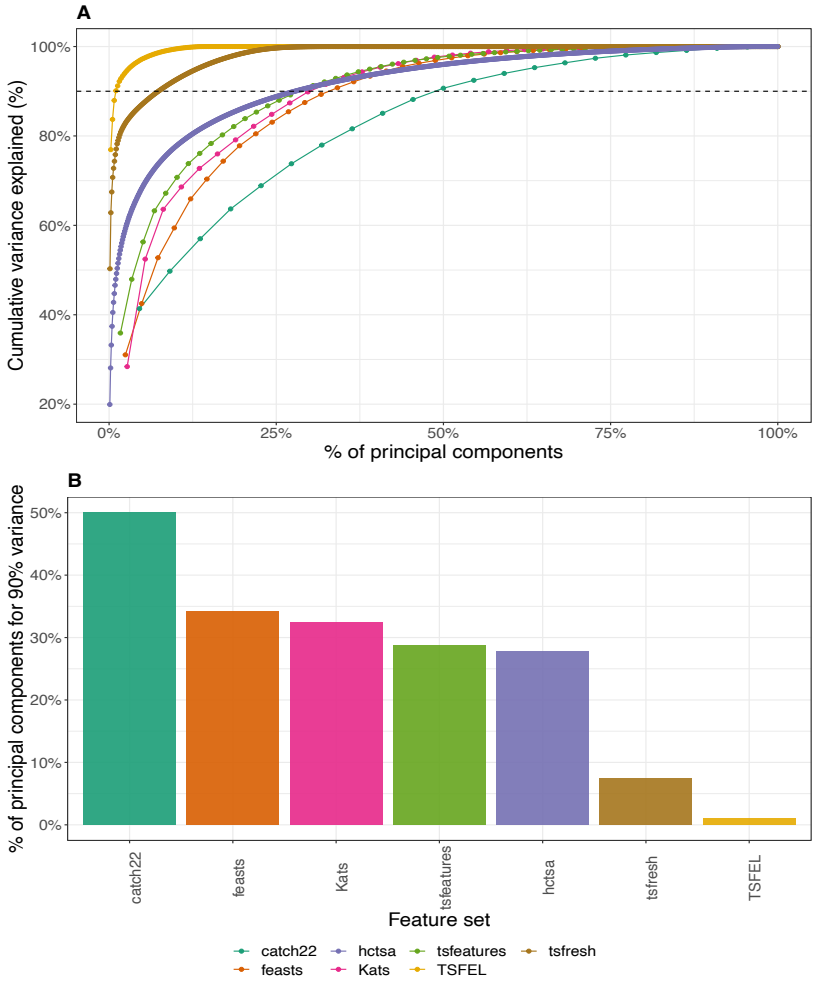


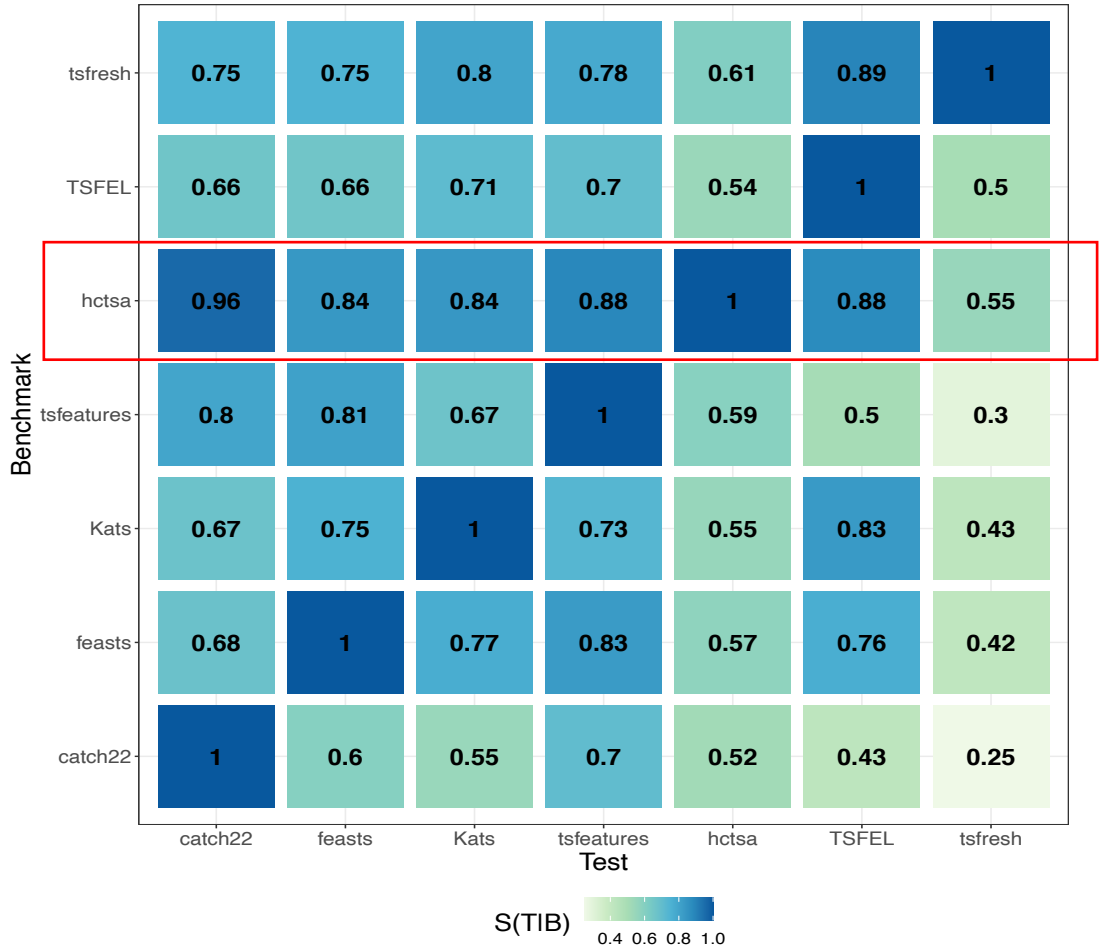
Key takeaways

- Computation time varies over orders of magnitudes
- Within-set redundancy is high for tsfresh and TSFEL
- Correlations were identified between feature sets, with tsfresh being the most “unique”

Henderson, T., & Fulcher, B. D. (2021). **An Empirical Evaluation of Time-Series Feature Sets**. 2021 International Conference on Data Mining Workshops, 1032-1038

…And they differ considerably…



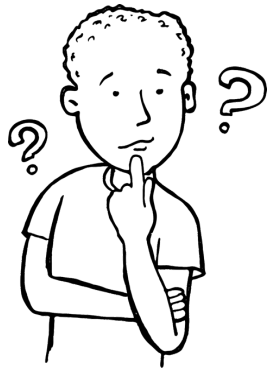


Henderson, T., & Fulcher, B. D. (2021). **An Empirical Evaluation of Time-Series Feature Sets**. 2021 International Conference on Data Mining Workshops, 1032-1038

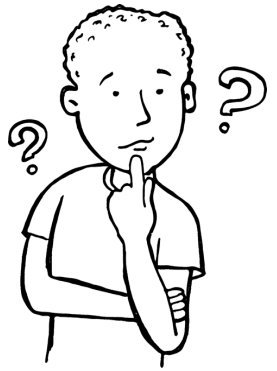
…Which raises many questions

I have a large fMRI dataset, what feature set do I use?

catch22 seems fast, does **feasts** add useful information on top of it?



Kats is a new feature set, how is it related to existing sets?



I don’t know what temporal dynamics best distinguish my classes, what feature set(s) should I use?

The theft package for R

theft

Tools for Handling Extraction of Features from Time series

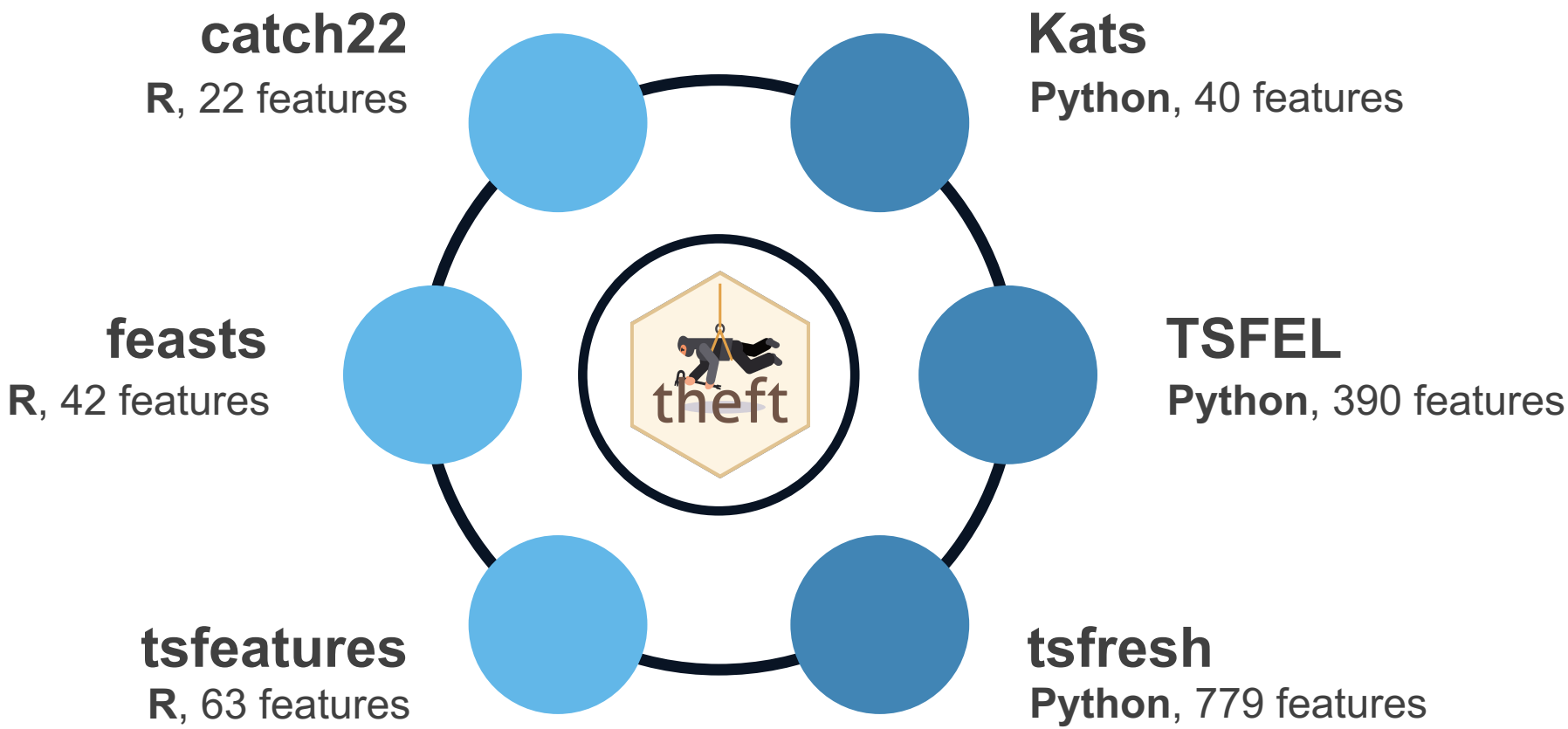
```
install.packages("theft")
```

The time series software package you never knew you needed

theft is a unified and extendable framework for computing features from six open-source sets from both R and Python.

It also includes a suite of functions for processing and interpreting the performance of extracted features, with extensive data-visualization templates, low-dimensional projections, and fitting and evaluation of feature-based classifiers.

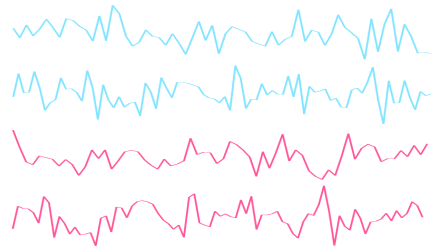
theft extracts features from six libraries in one convenient package



And provides an extensive workflow for feature-based time-series analysis

1. Load in raw time-series dataset

A



B

id	time	value	group
1	1	0.75	Control
2	1	1.24	Control
3	1	0.42	Treatment
.	.	.	.
.	.	.	.
.	.	.	.

2. Extract features for each unique time series

C

`init_theft` (If using Python feature sets)

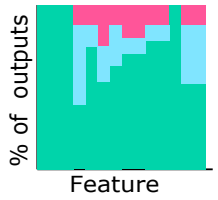
D

`calculate_features`

`catch22` `feasts` `tsfeatures` `tsfresh` `TSFEL` `Kats`

3. Analyze and visualize features

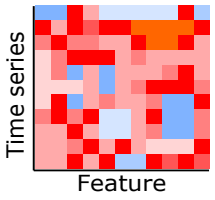
E



`plot_quality_matrix`

- Understand extraction quality
- Identify poor features to filter

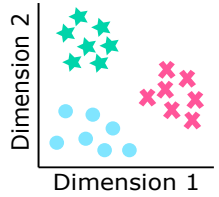
F



`plot_feature_matrix`

- Visualize extracted features
- Understand similarities

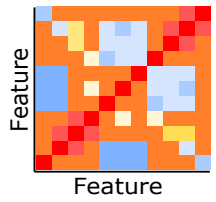
G



`plot_low_dimension`

- Reduce feature matrix size
- Classify time-series groups

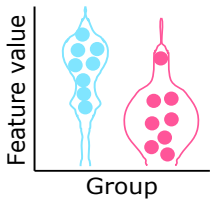
H



`plot_feature_correlations`

- Visualize time-series structure
- Visualize feature relationships

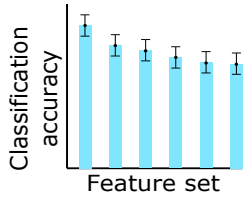
I



`compute_top_features`

- Classify time-series groups
- Visualize feature performance

J

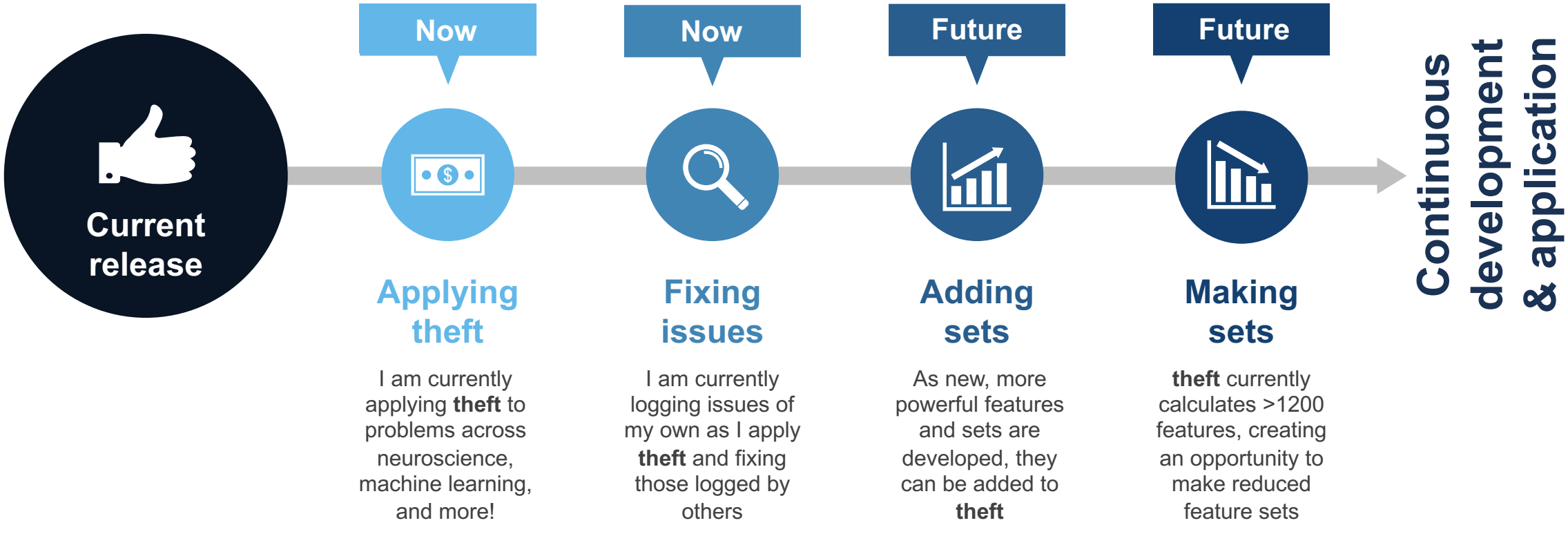


`fit_multi_feature_classifier`

- Classify time-series groups
- Visualize set/all performance

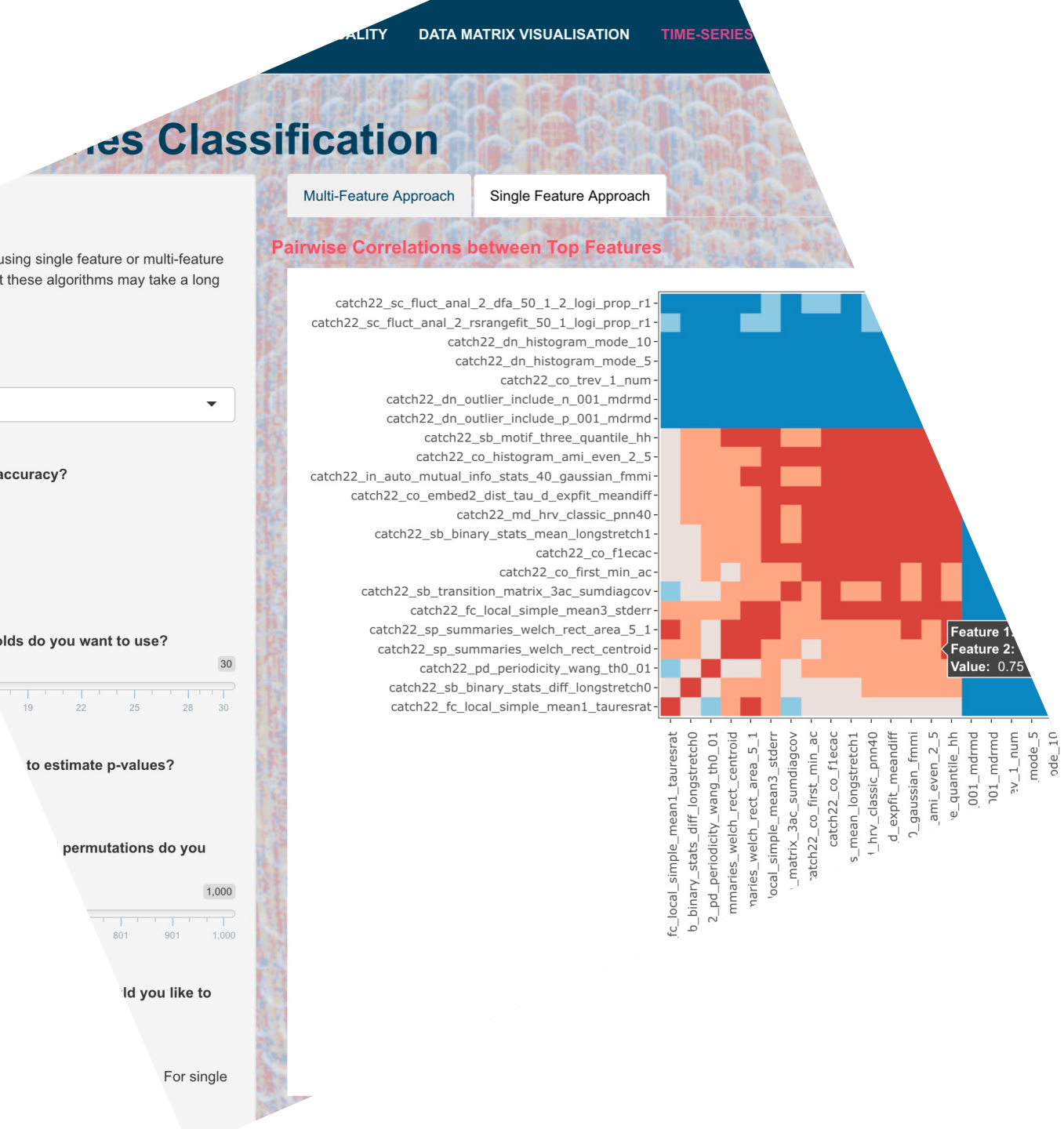
Future directions

theft is flexible and extensible



Please feel free to contribute!

You can find the **theft** source code on GitHub: https://github.com/hendersontrent/theft
A website with a rendered vignette of functionality is available: https://hendersontrent.github.io/theft/



See you later for the theft demo!

Resources for the demo:

- https://github.com/hendersontrent/CNS-2022-theft

Thanks for listening!

Feel free to get in touch:

then6675@uni.sydney.edu.au
https://github.com/hendersontrent
@trentlikesstats
https://www.linkedin.com/in/trent-henderson/

Some of my software packages: