

Rediscover the known Universe with NASA datasets

Horacio Gonzalez

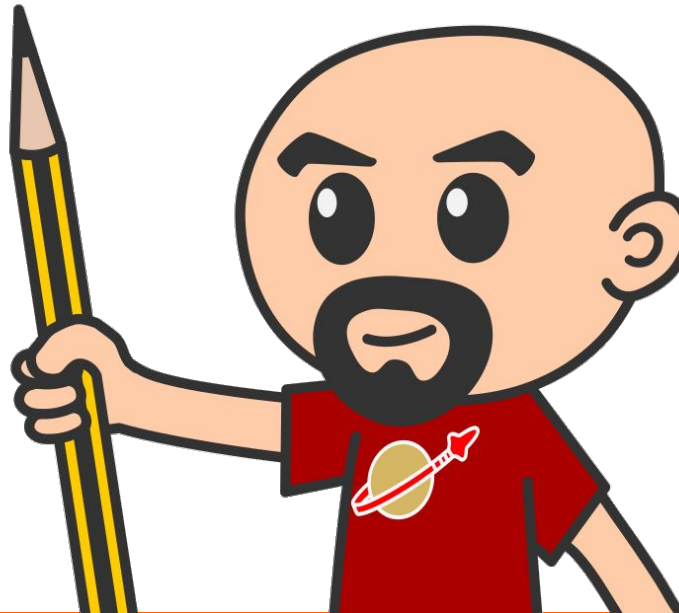
@LostInBrittany



Horacio Gonzalez

@LostInBrittany

Spaniard lost in Brittany, developer, dreamer and all-around geek



HelloExoWorld



Looking for exoplanets in NASA datasets



HelloExoWorld

Once upon a time...



An amateur astronomer



Pierre Zemb, DevOps OVH



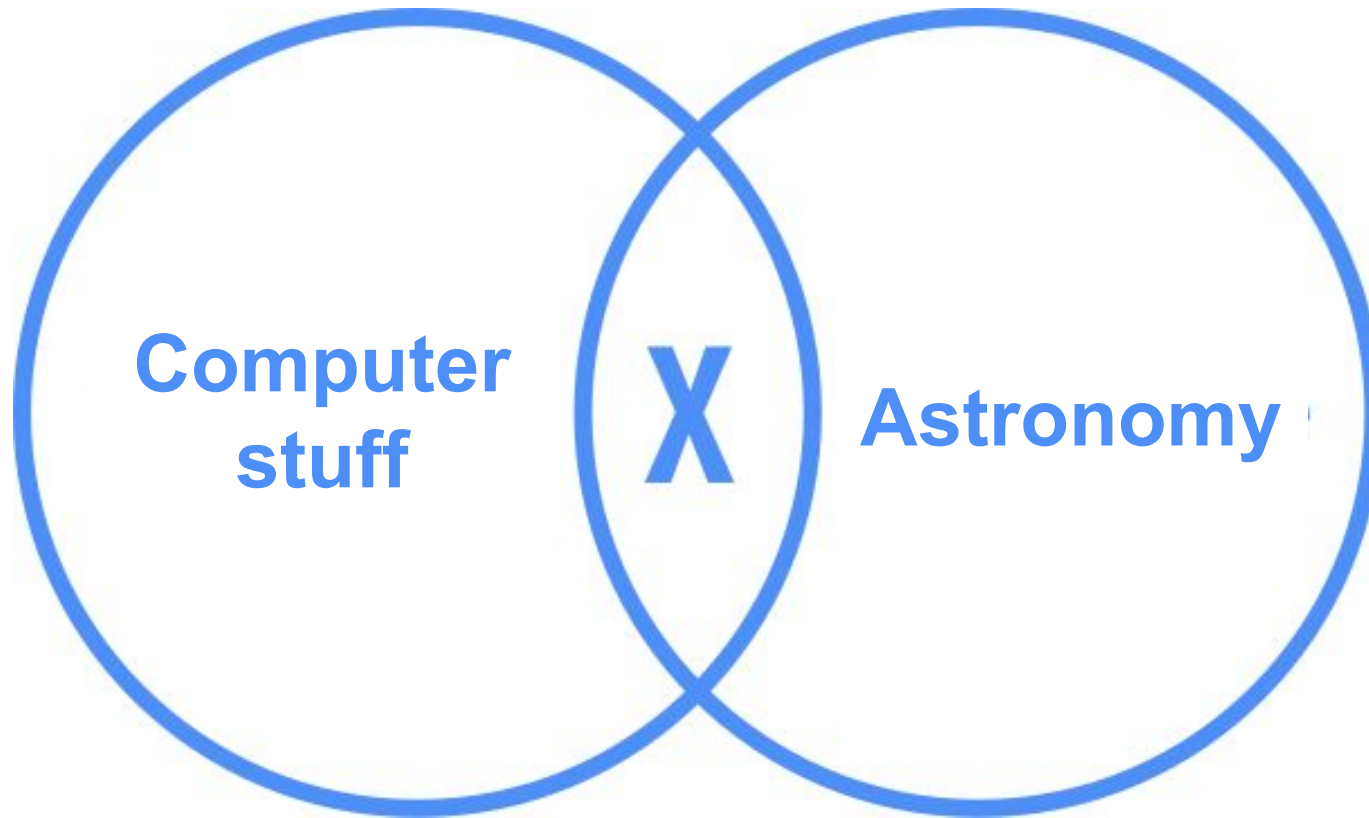
What not to do if you love astronomy



Live in Brest



Looking for solutions



Mixing passions



Google is your friend...



time series astro

time series **astronomy**
time series **analysis in astronomy limits and potentialities**
astroml.time series
astronomical time series **analysis**
random time series **in astronomy**
astrophysical time series

Google Search I'm Feeling Lucky

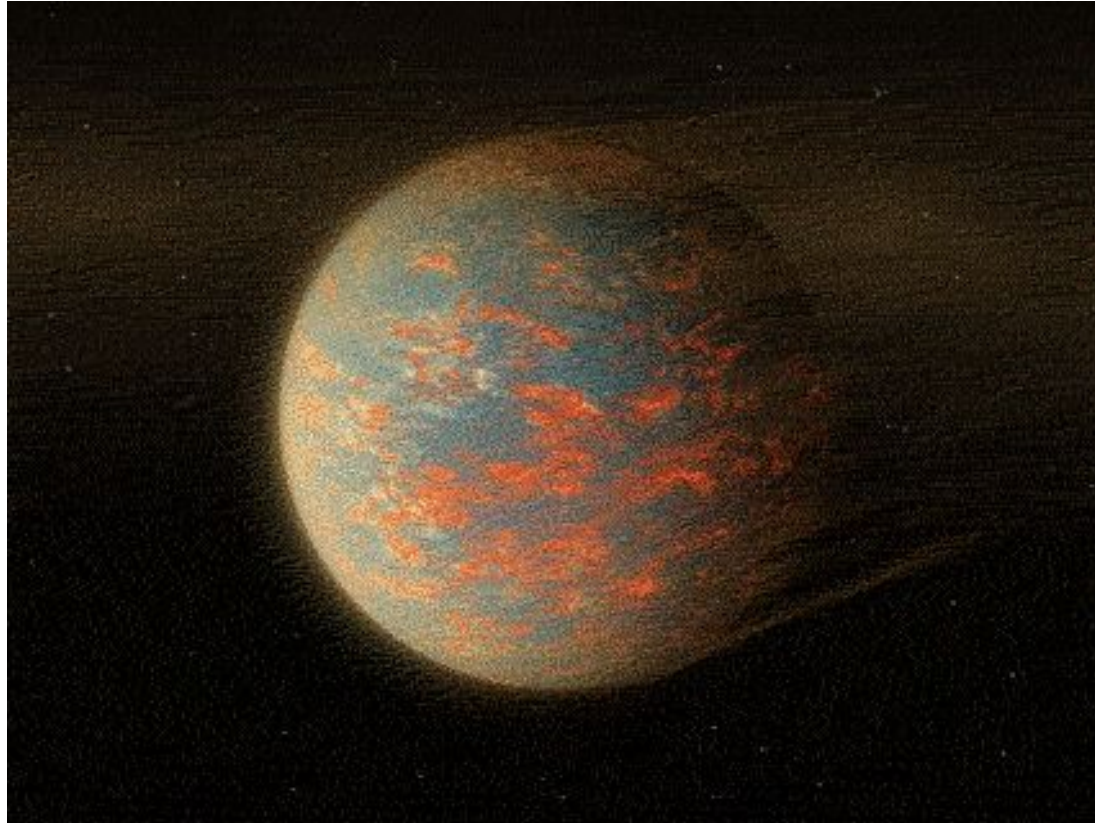
[Learn more](#)

[Report inappropriate predictions](#)

Let's find a project



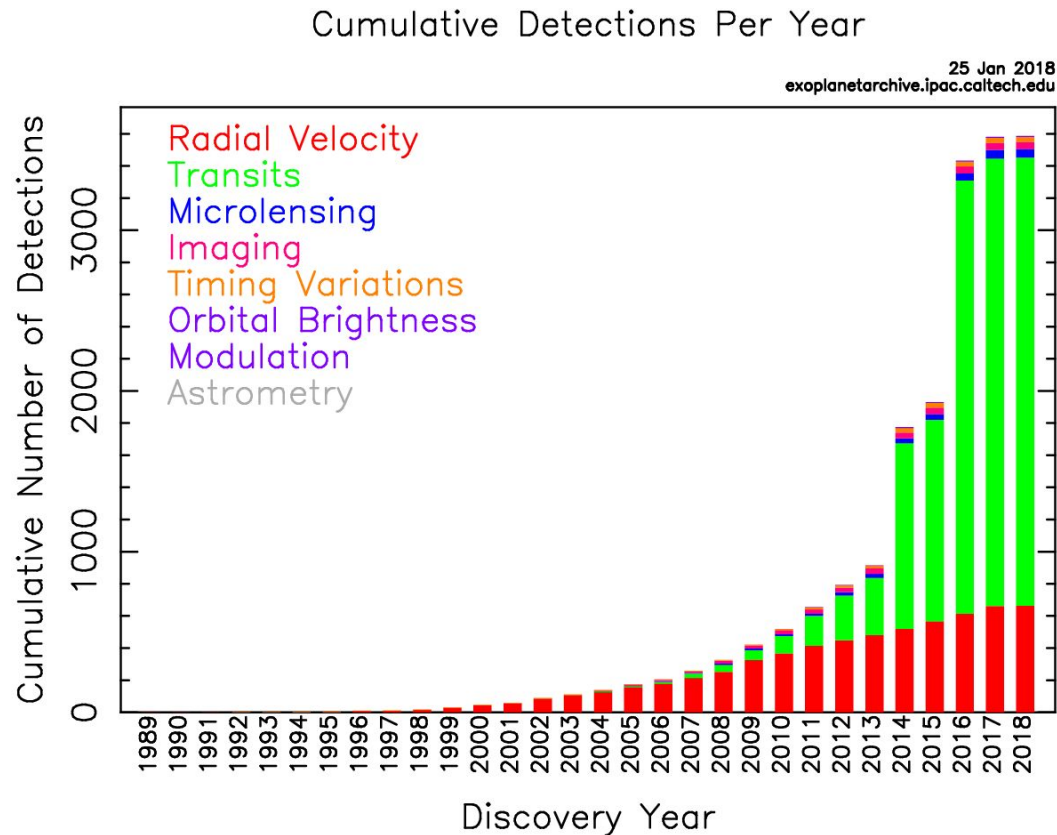
Exoplanets?



Planets orbiting stars far away



How do we find them?



The transit method seems the best



The transit method



Credits: NASA's Goddard Space Flight Center



How do we look for transits?

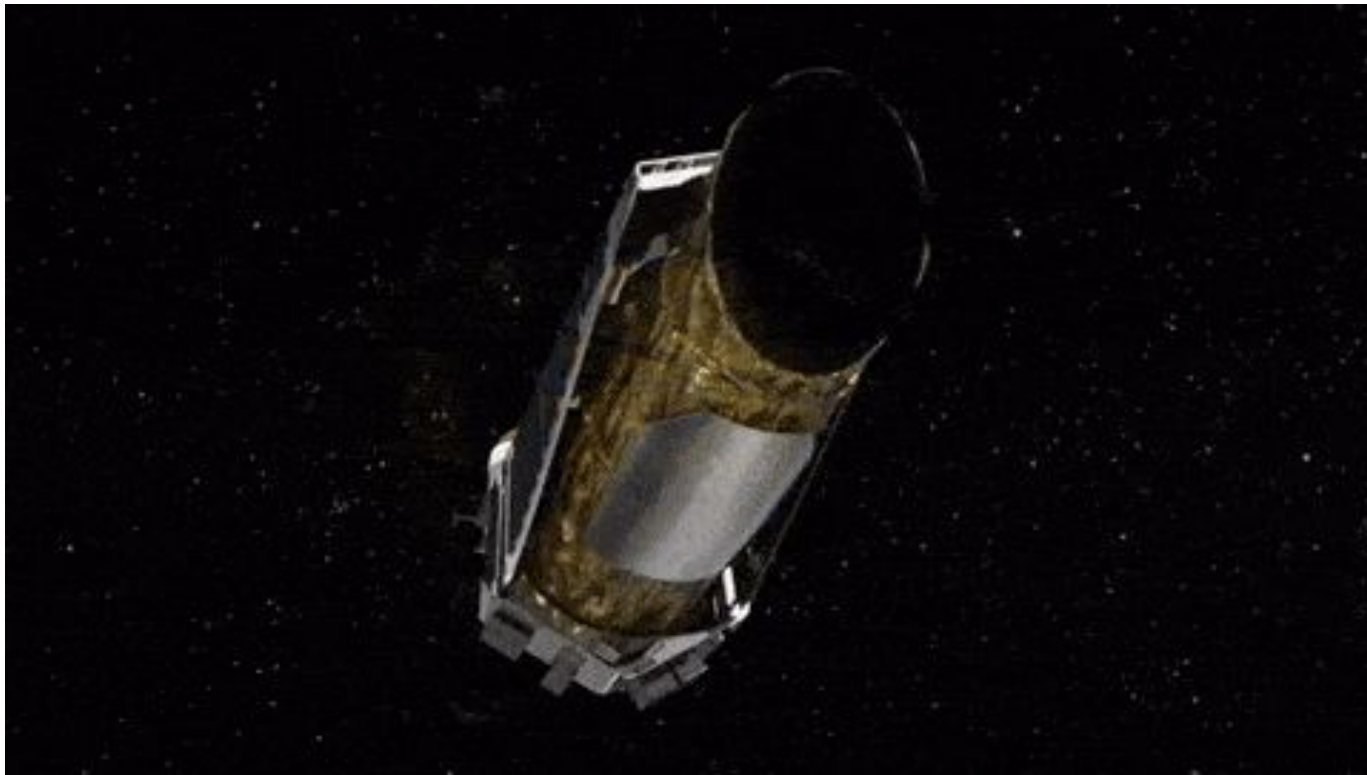
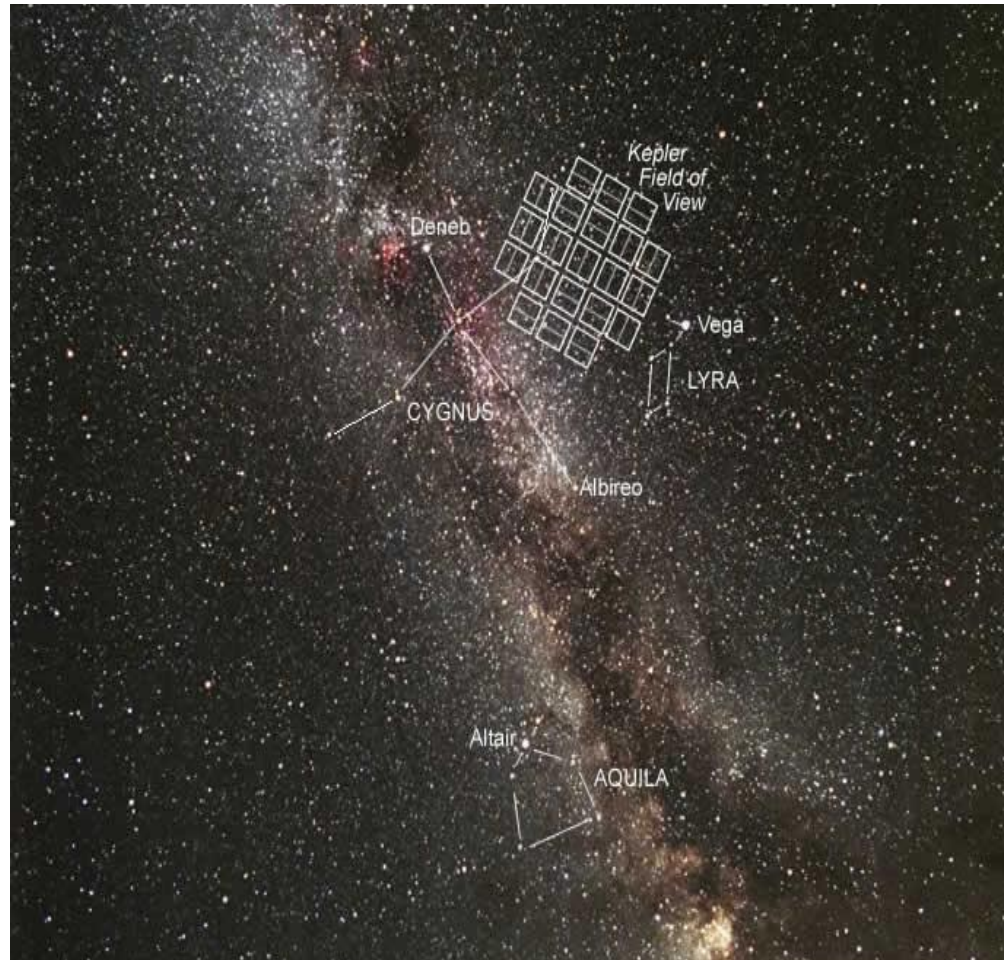


Image credits : NASA

Kepler



Watching the sky



By Carter Roberts [Public domain], via Wikimedia Commons



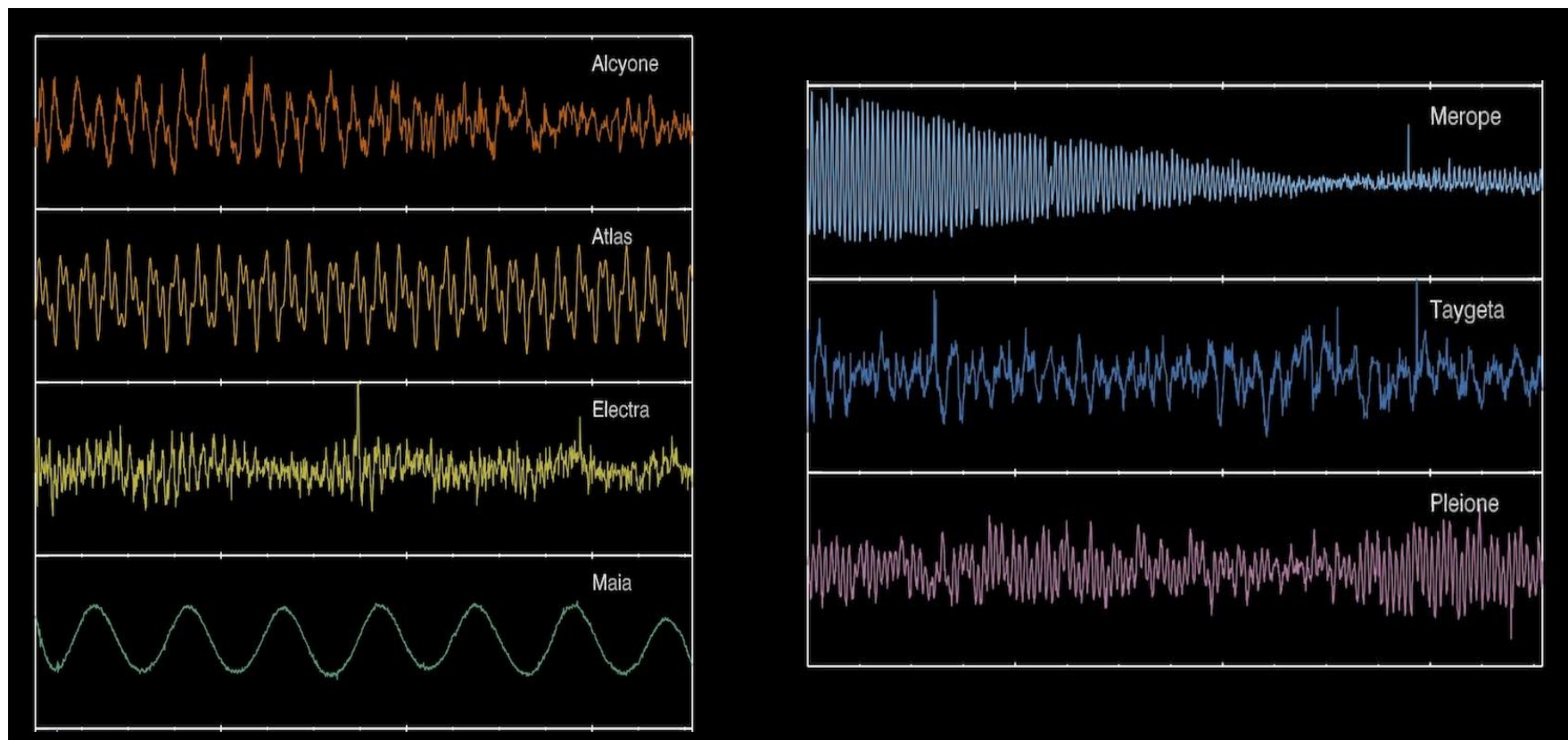
And what kind of data we get?



Pleiades By NASA, ESA, AURA/Caltech, Palomar Observatory. Via Wikimedia Common



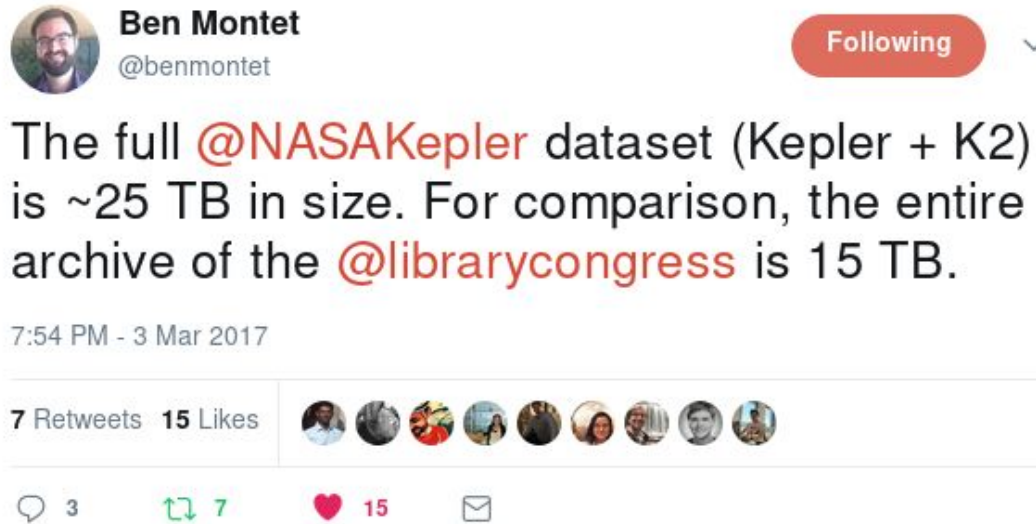
Well, that's the problem



Seven stars, seven different profiles



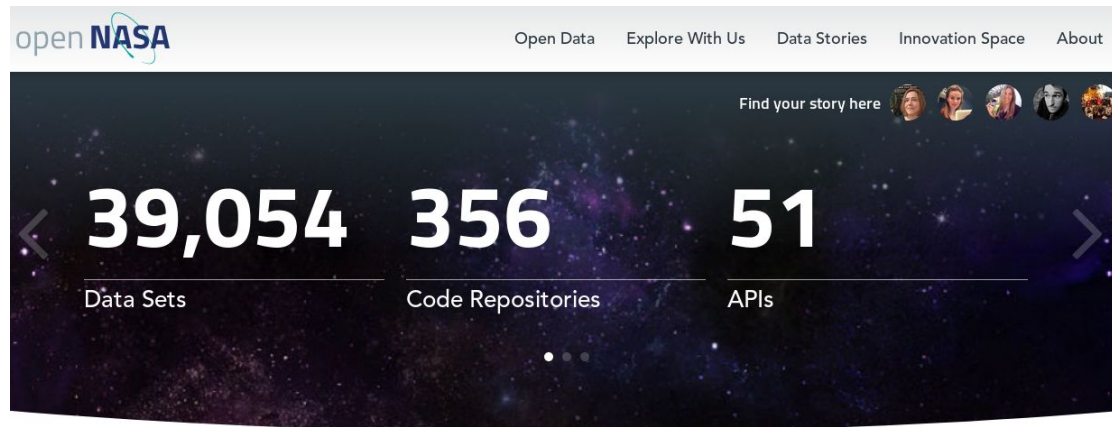
Kinda big data



Over 40 million light curves



Big AND open data



What describes you best?



Citizen Scientist



Developer



Citizen Activist



Govvie



Curious

Lots of datasets in [#opendata](#)



And we can help with that!



Let's use our tools to analyse the data



Time Series

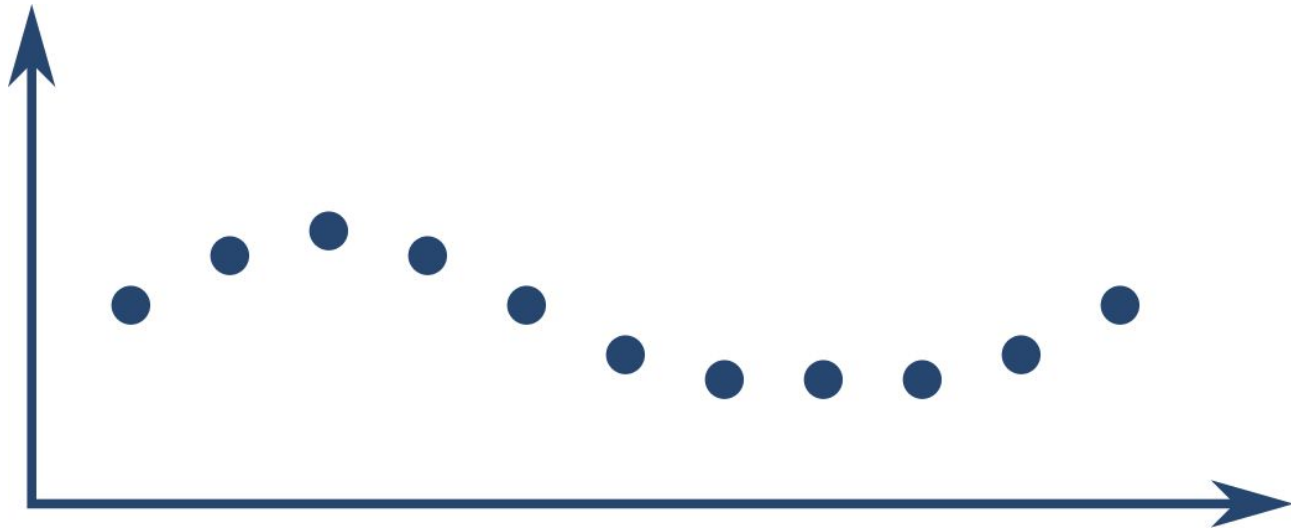
To analyse Kepler datasets



Kepler: spatial Time Series

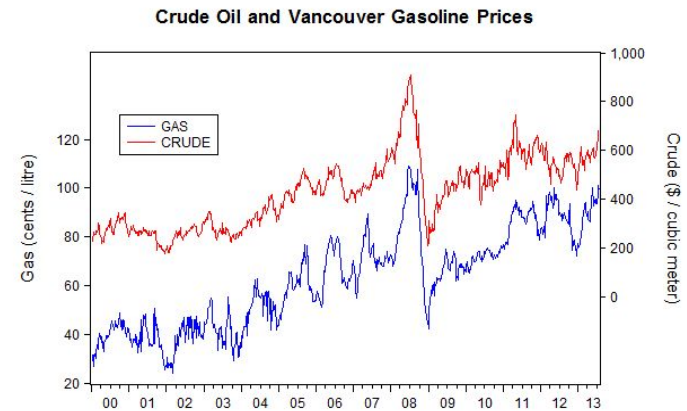
Definition of Time Series:

A series of data points indexed in time order



Time Series

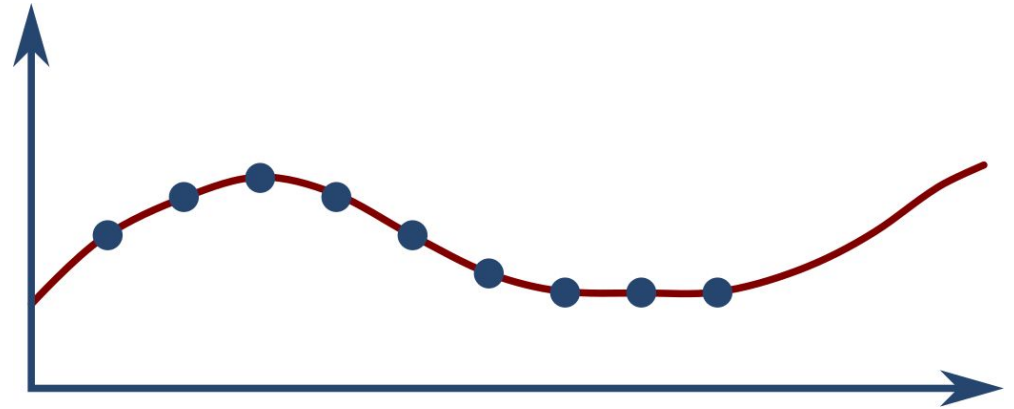
- Stock Market Analysis
- Economic Forecasting
- Budgetary Analysis
- Process and Quality Control
- Workload Projections
- Census Analysis
- ...



Time Series

Applications:

- Understanding the data
- Fit a model
 - Monitoring
 - Forecasting



Time Series

Stock market Analytics
Economic Forecasting



\$\$\$



Study & Research



Time Series

Many specific analytical tools:

- Moving average
- ARMA (AutoRegressive Moving Average)
- Multivariate ARMA models
- ARCH (AutoRegressive Conditional Heteroscedasticity)
- Dynamic time warping
- ...



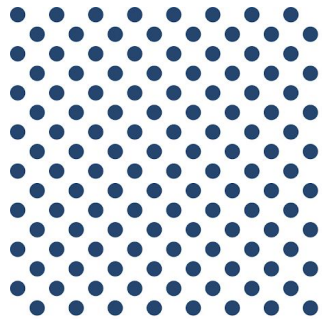
Time Series

Specific application of general tools

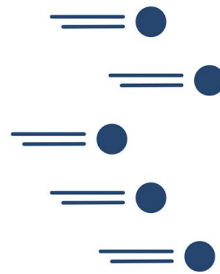
- Artificial neural networks
- Hidden Markov model
- Fourier & Wavelets transforms
- Entropy encoding
- ...



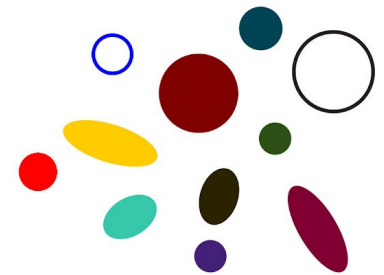
Dealing with Time Series



Volume



Velocity



Variety

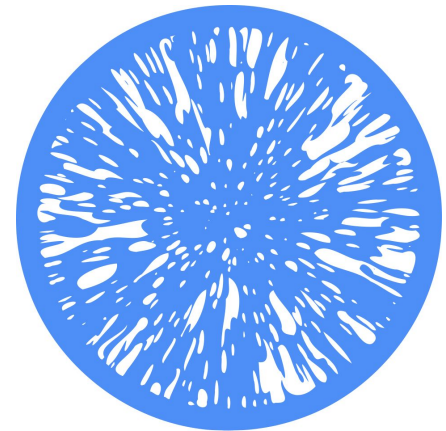
The 3 'v'



A match made in heaven

Warp 10, OVH Observability and HelloExoWorld

METRICS



OVH Observability Data Platform

Some of OVH Observability metrics:

- 1.5M datapoints/s, 24/7
- Peaks at ~10M datapoints/s
- 500M unique series



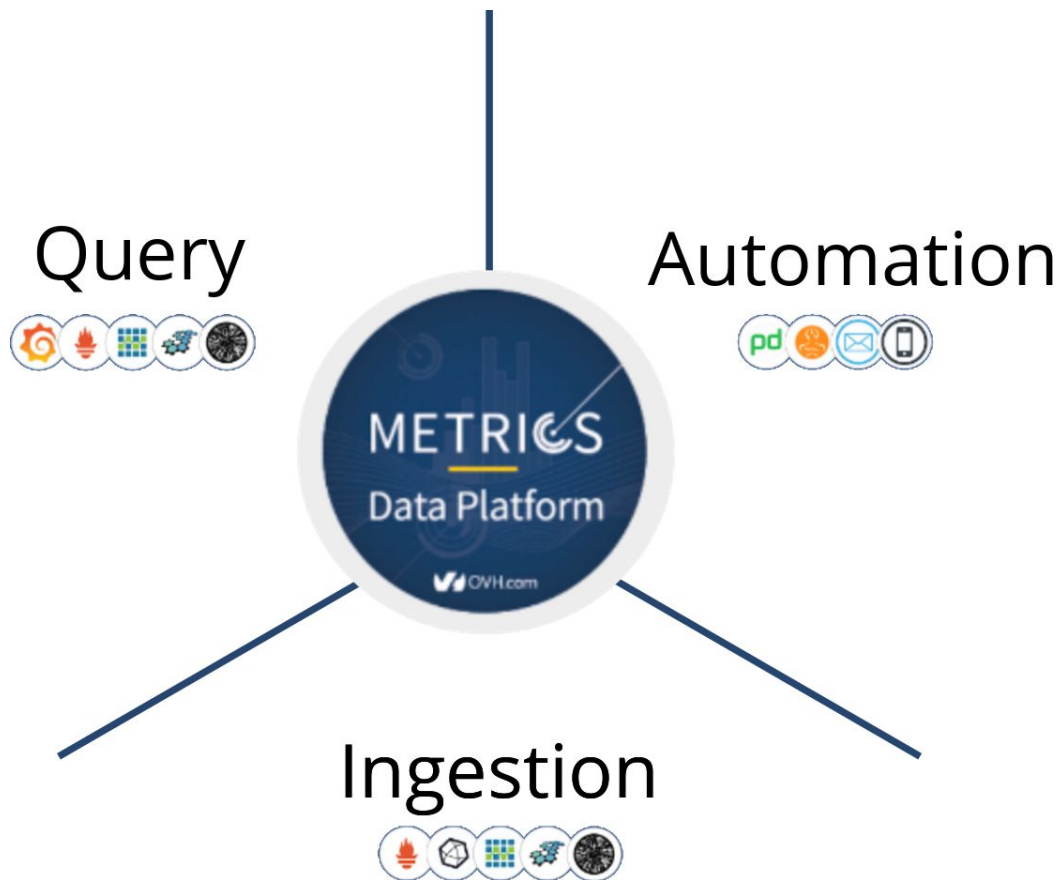
Tools to deal with Time Series



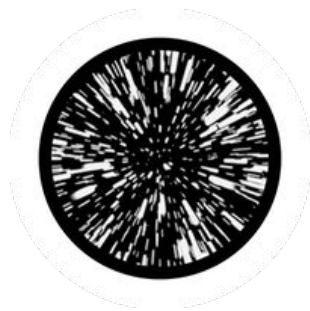
Many options



Metrics Data Platform



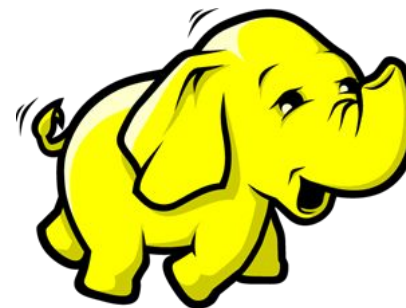
Metrics Data Platform



+



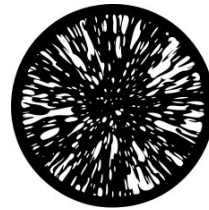
+



Why Warp 10?

Warp 10 is a software platform that

- Ingests and stores time series
- Manipulates and analyzes time series



WARP 10



Analytics is the key to success



Fetching data is only the tip of the iceberg



Manipulating Time Series with Warp 10

A true Time Series analysis toolbox

- Hundreds of functions
- Manipulation frameworks
- Analysis workflow



What we have done

- Downloaded and parsed 40 millions of FITS files
- Pushed it to OVH Metrics
- Select a cool subset as training set
- Verified we could find the same planets as NASA



Choosing a star: Kepler 11

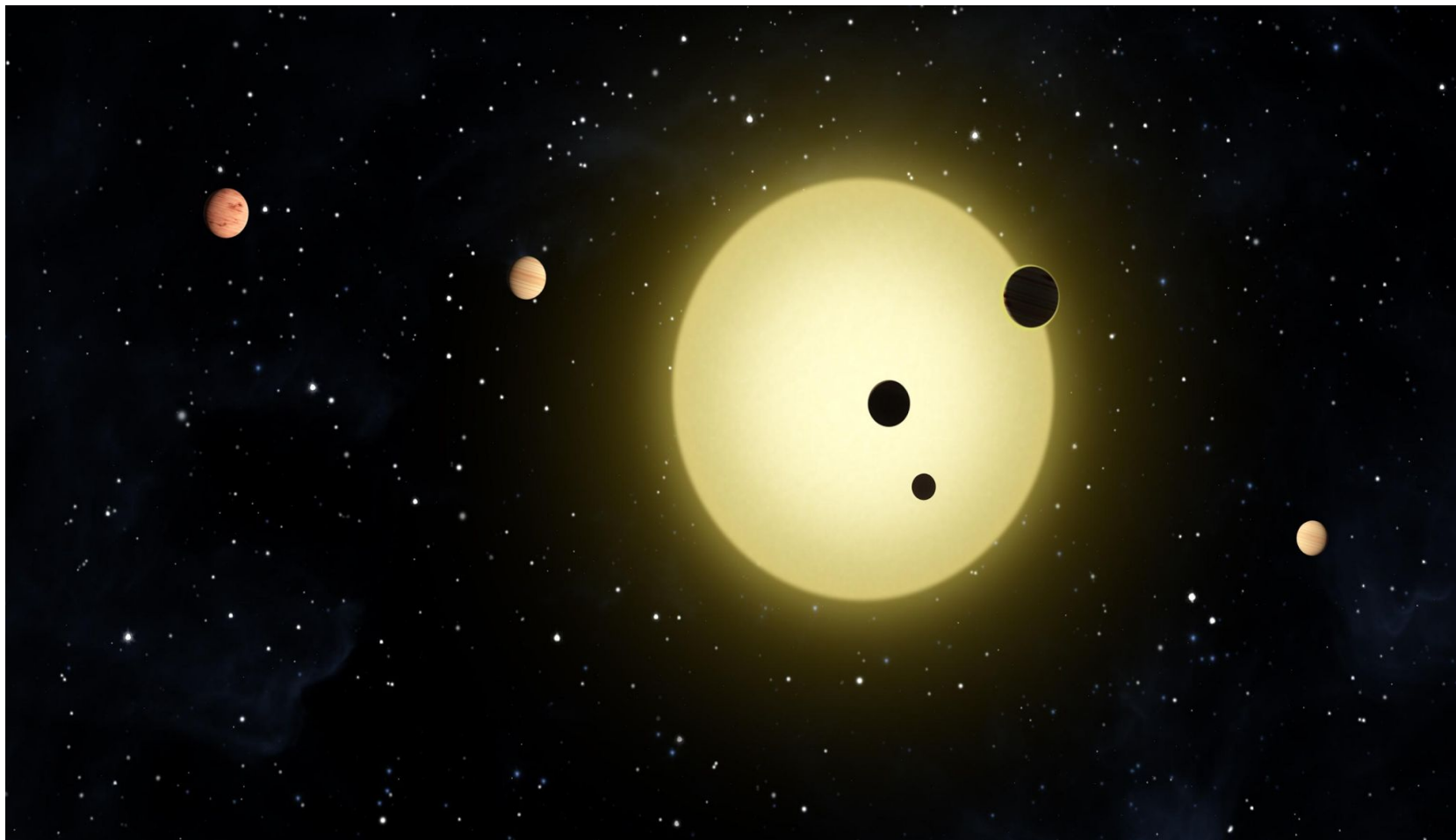
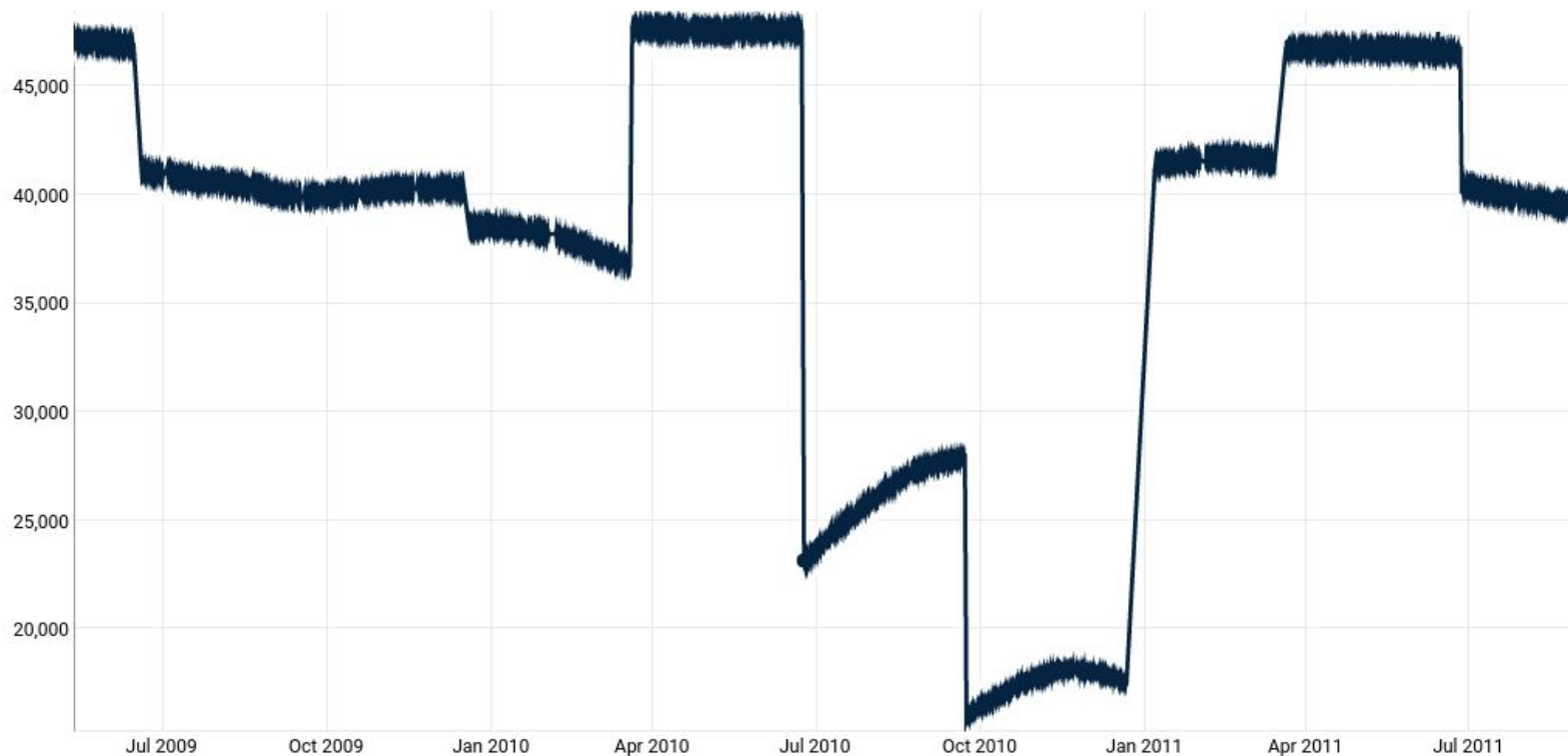


Image credit: NASA/Tim Pyle



Looking at the raw signal...

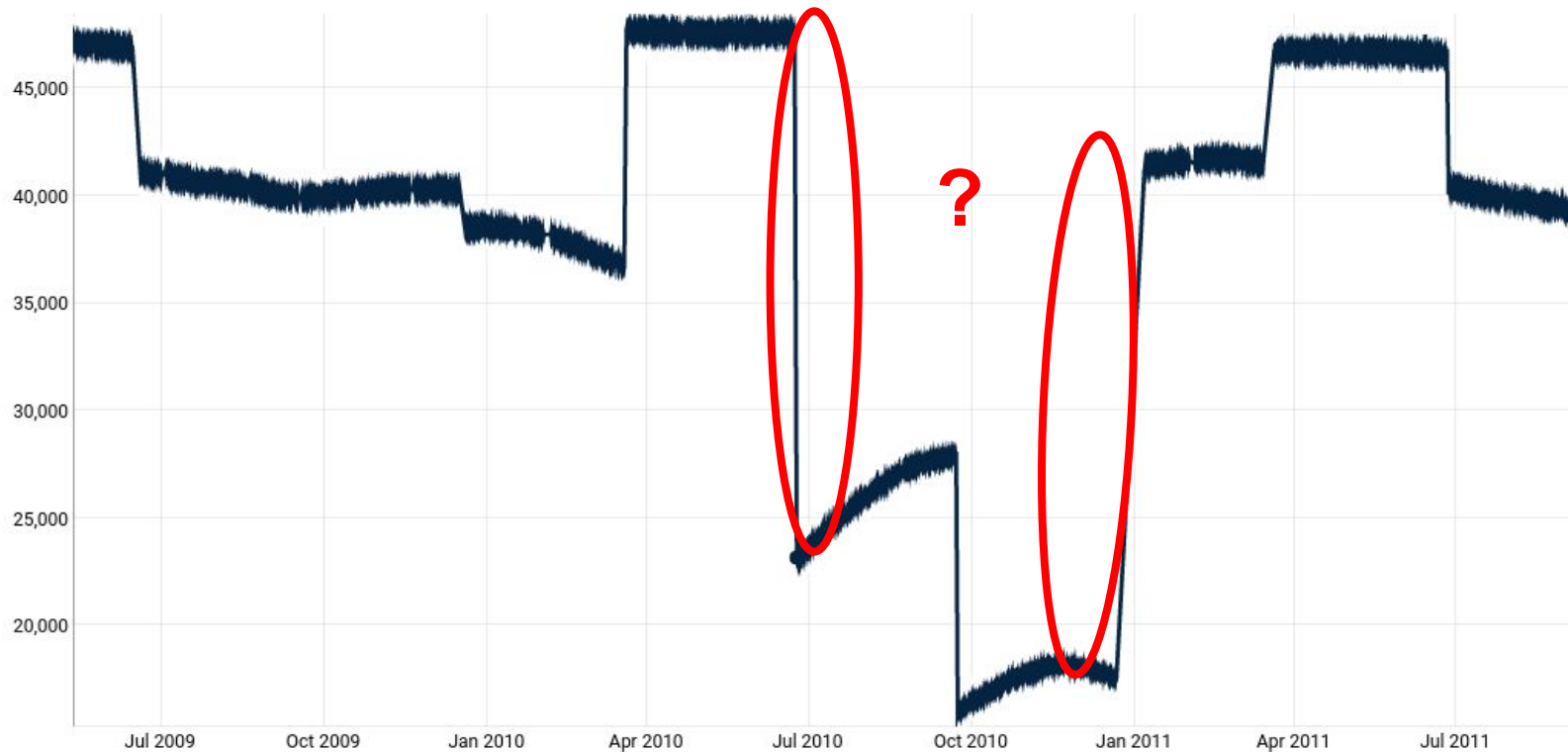


SAP_FLUX:

The flux in units of electrons per second contained in the optimal aperture pixels collected by the spacecraft.



Looking at the raw signal...

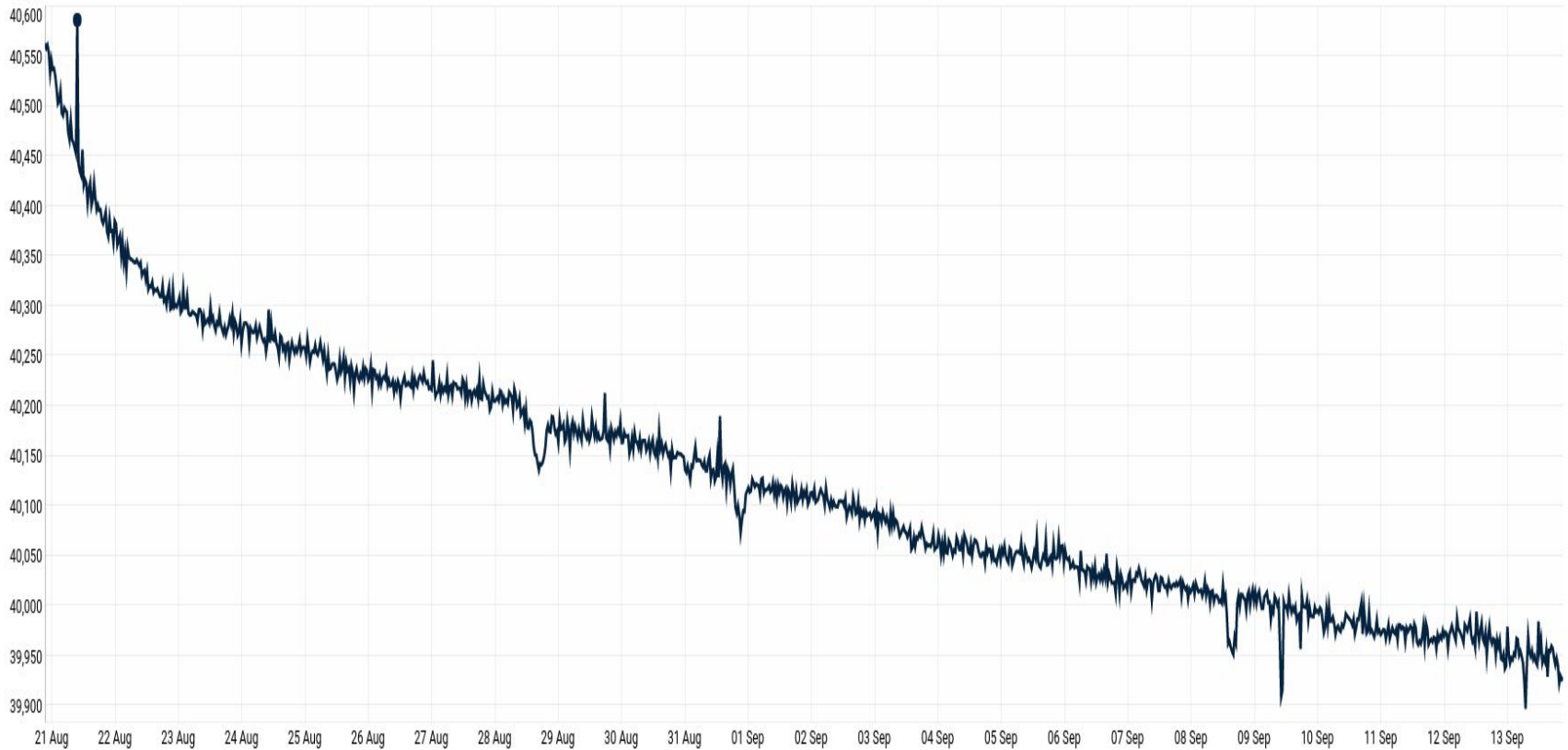


SAP_FLUX:

The flux in units of electrons per second contained in the optimal aperture pixels collected by the spacecraft.



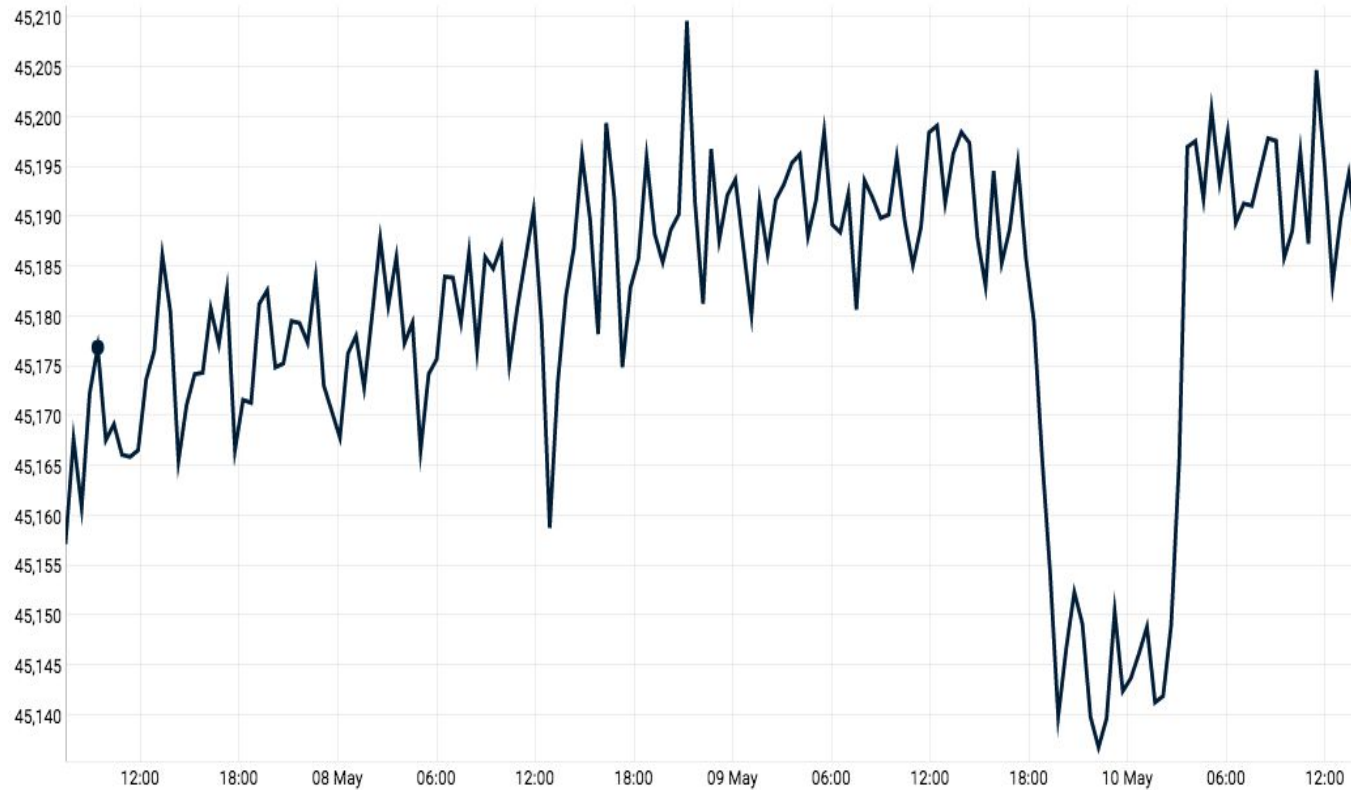
Looking at one record



Perturbations in dirty signals



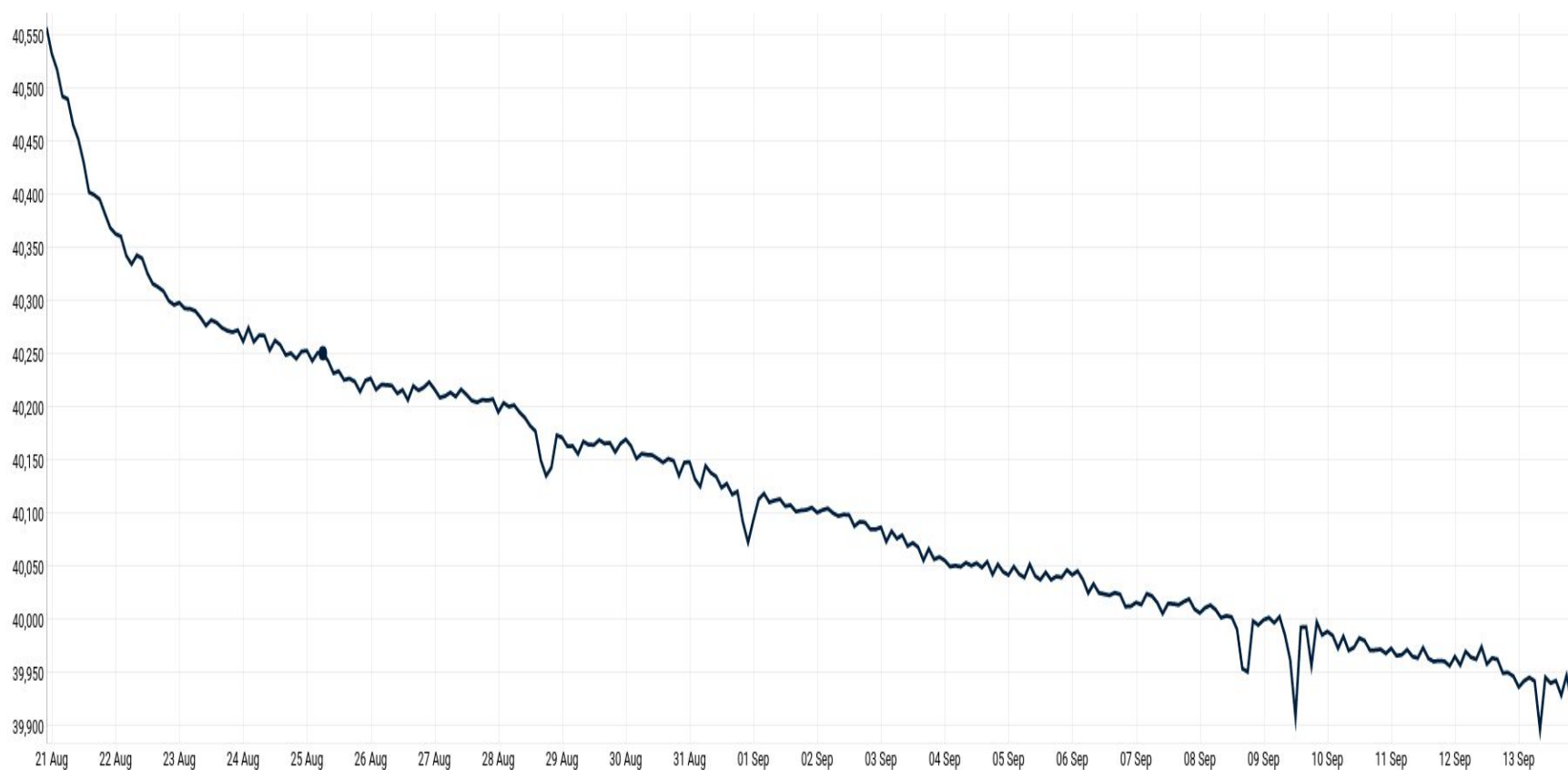
Transits are tiny



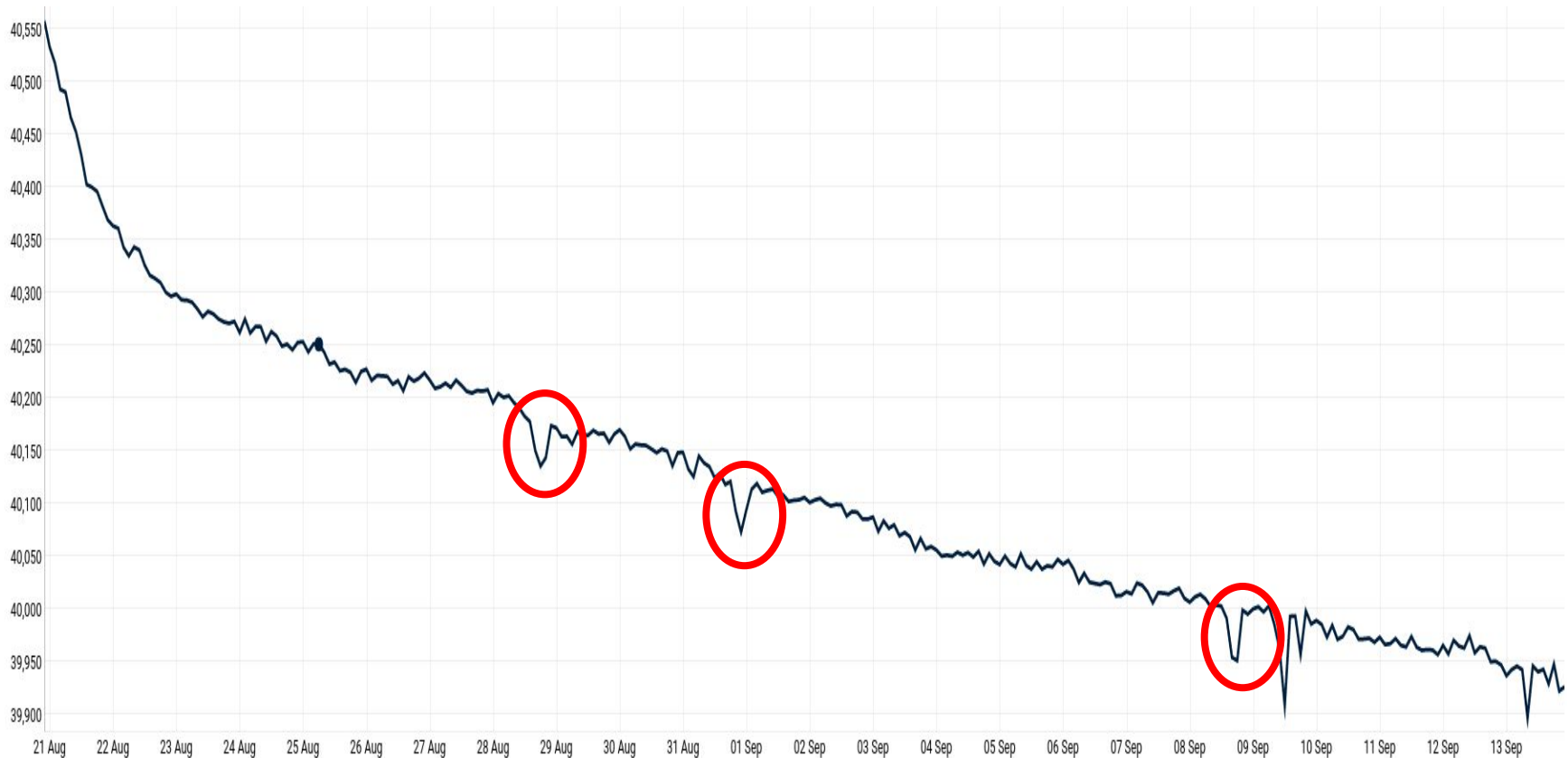
~40 electrons per second



First step: downsampling



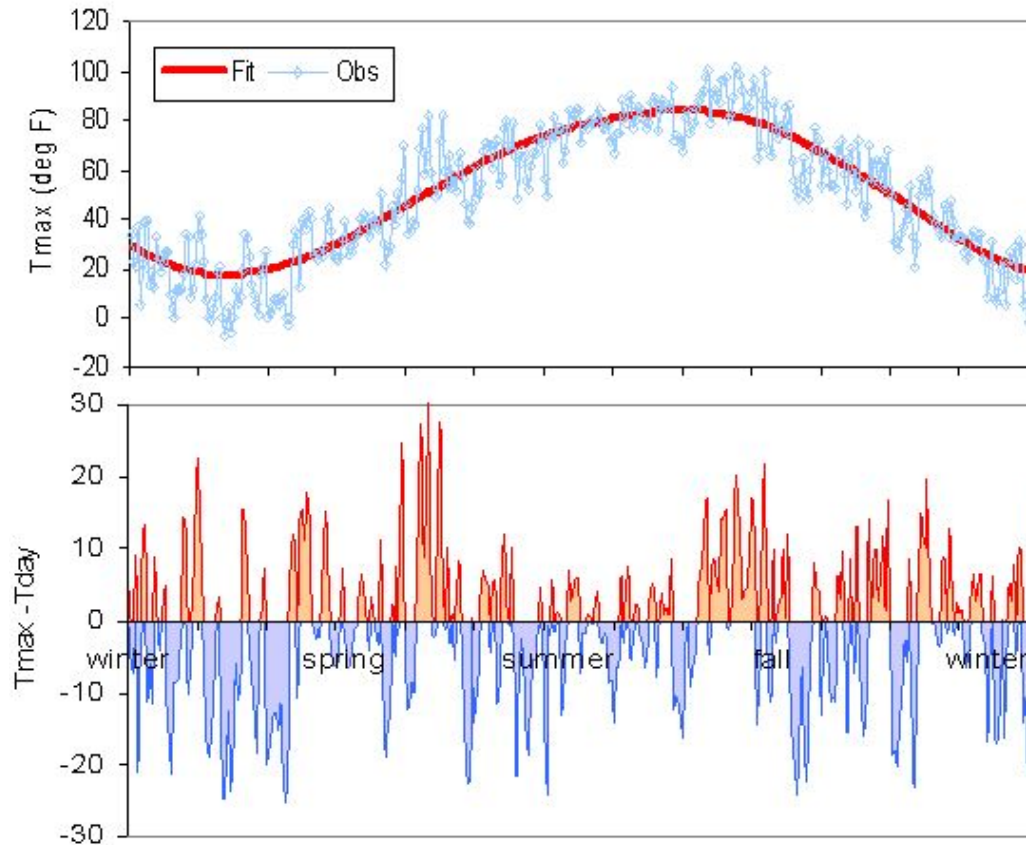
First step: downsampling



You can see the transit candidates...
but how can we teach the computer to see them?



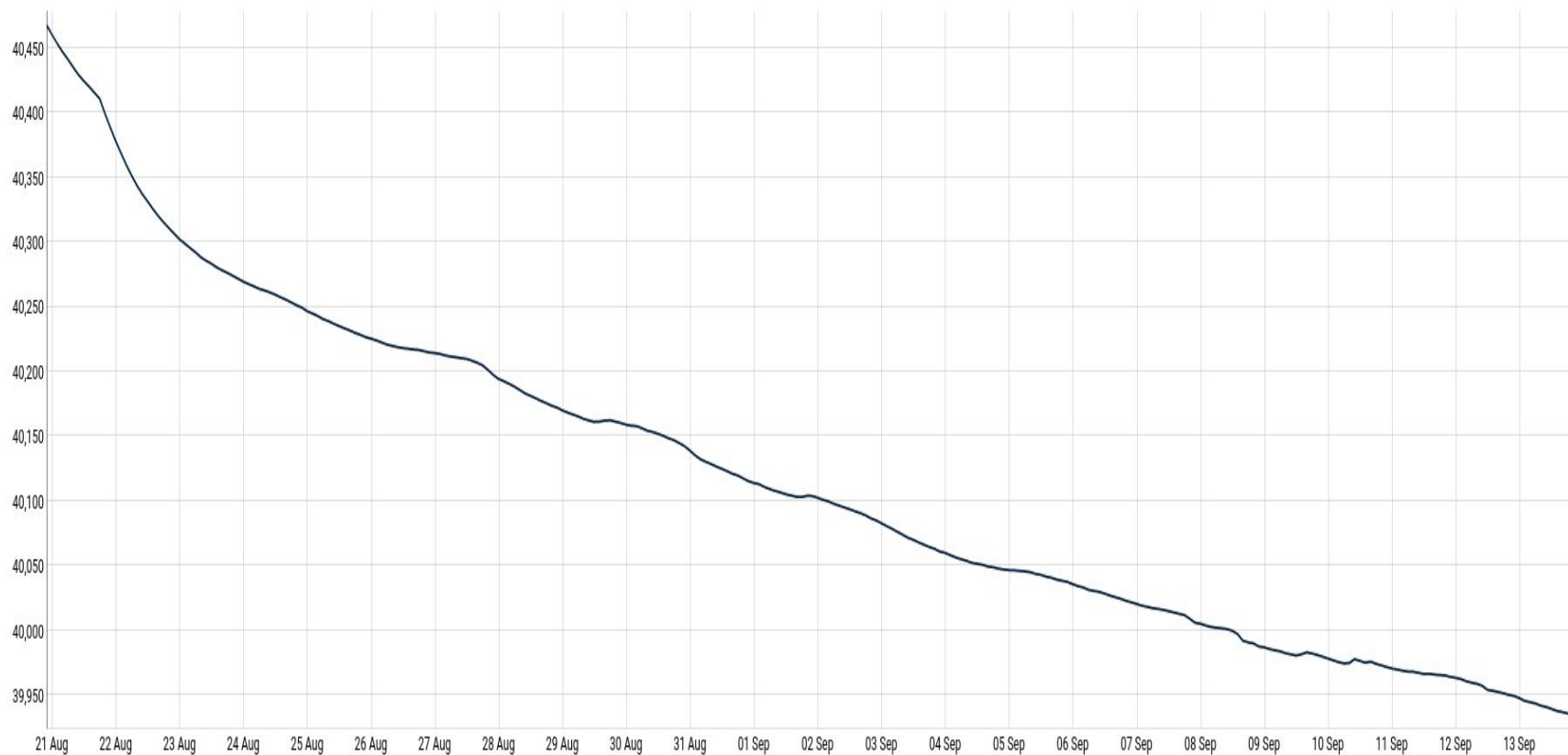
If you ♥ signal processing



High pass filter



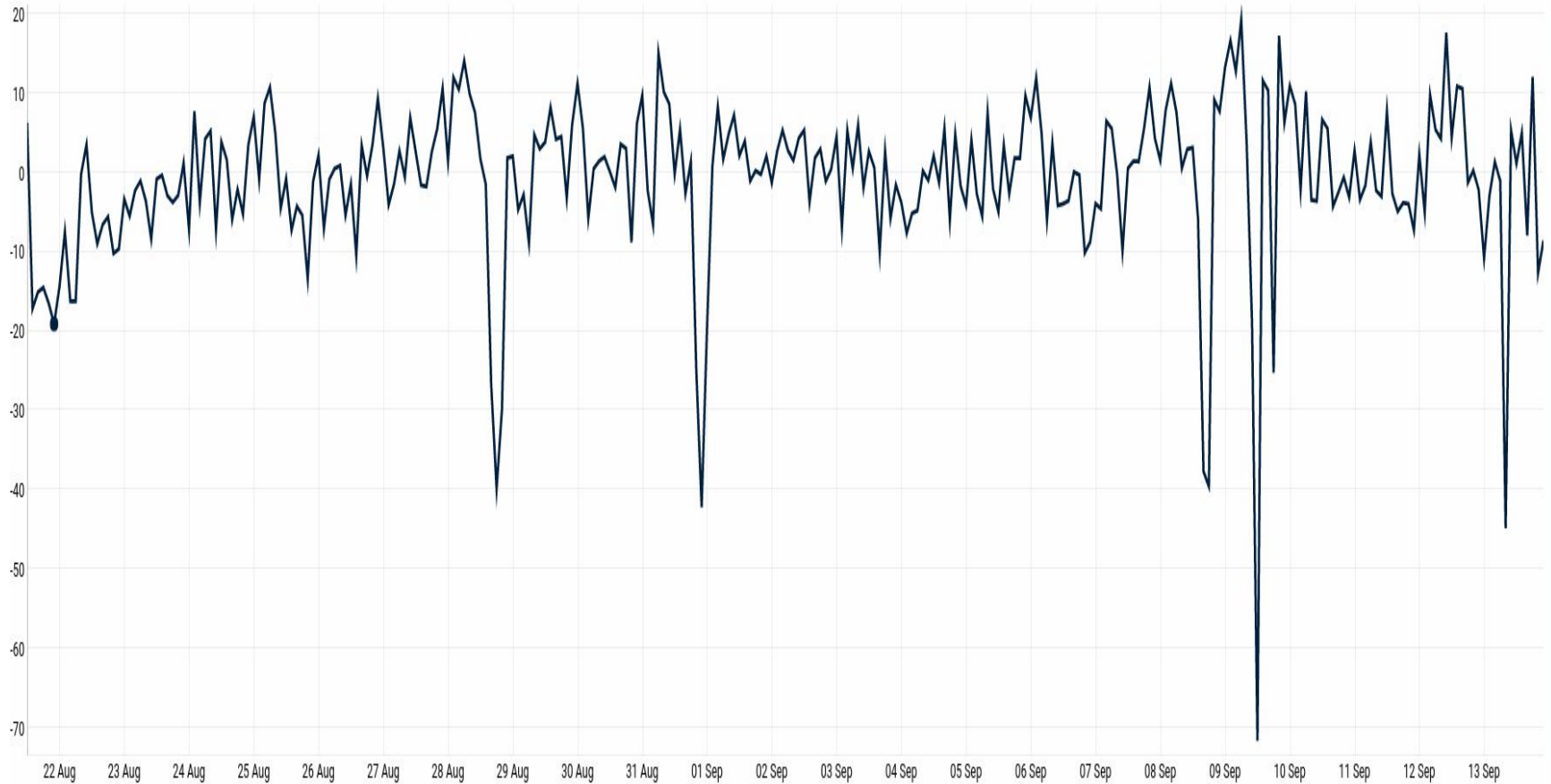
Poor person's high pass filter



Using the trend



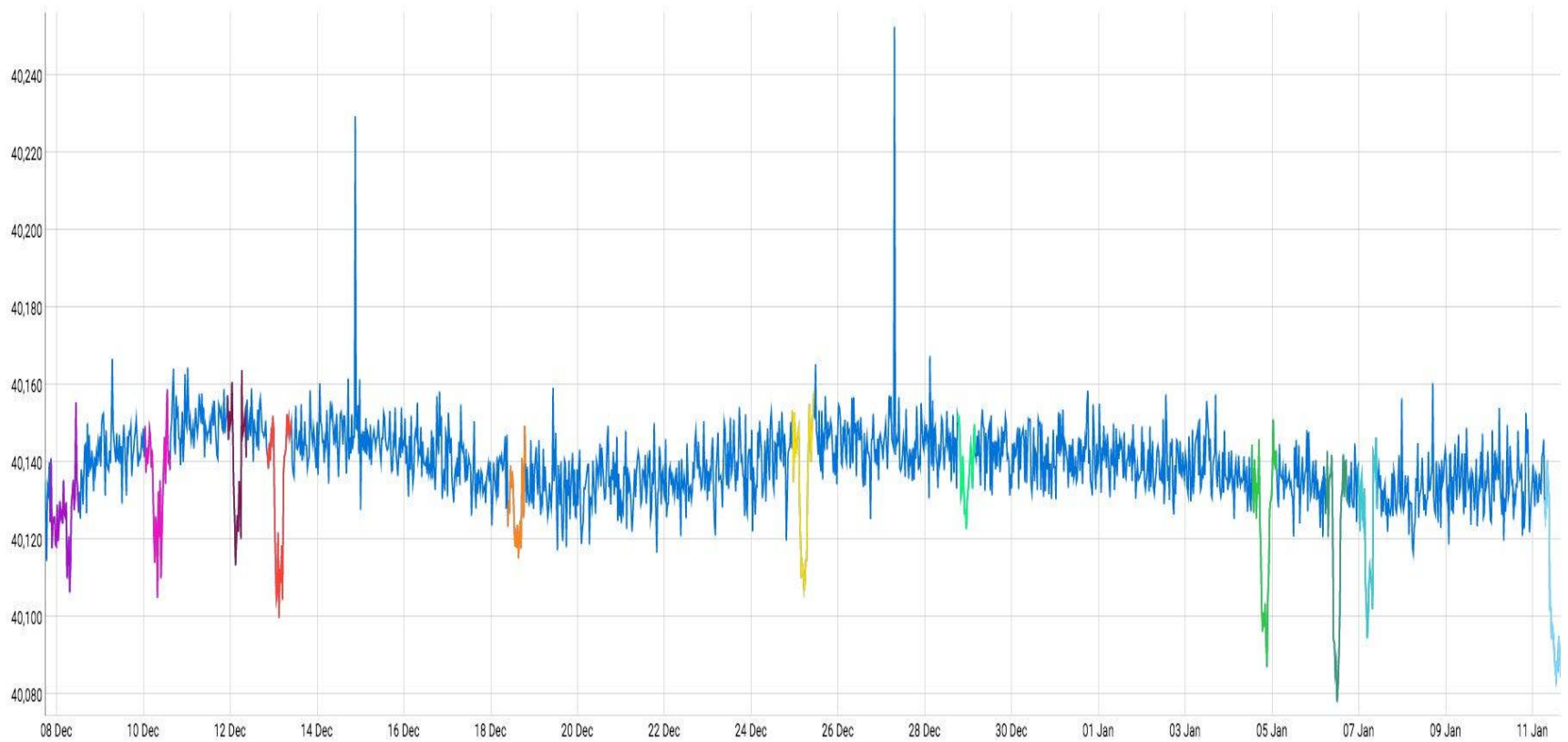
Signal - Trend



Now you can see them well



After some tuning



We have our transit candidates



What's next?

Where do we go from here?

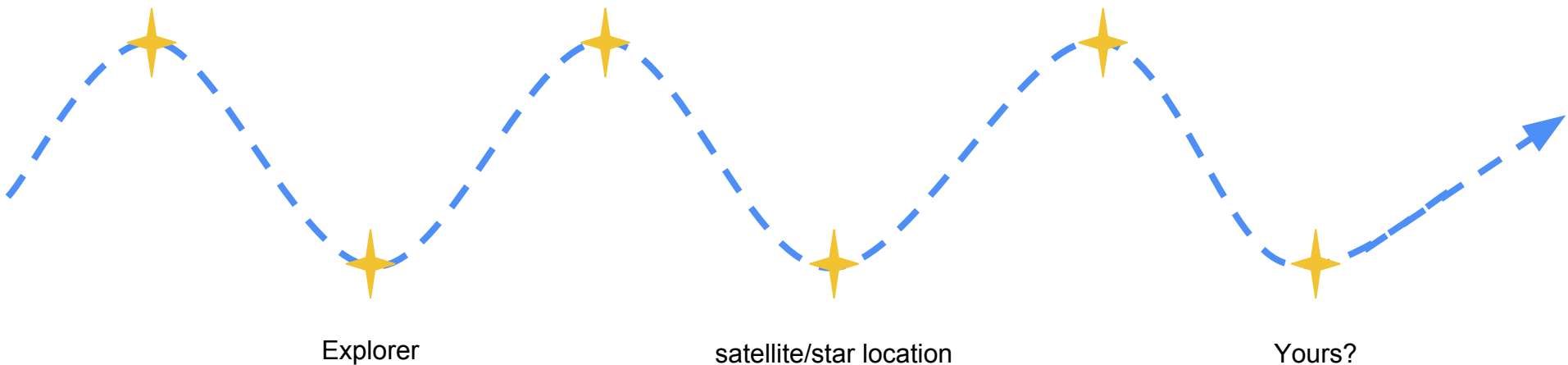


Only the beginning

New import method

Better detection

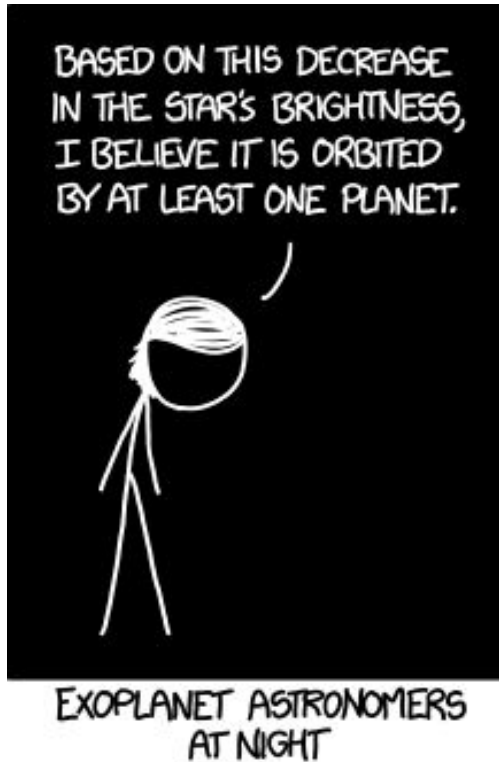
Deep learning



A growing team



And you!



Join us!
<https://helloexo.world>

<https://xkcd.com/1371/>



Want to know more?

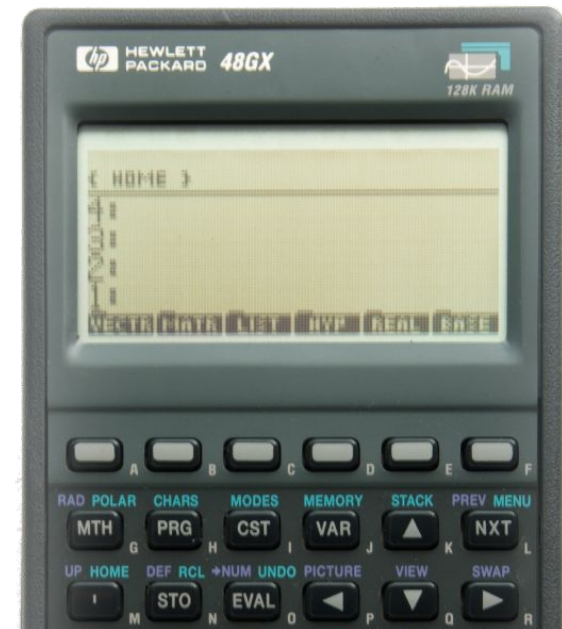
Analysing with WarpScript



WarpScript

Reverse Polish Notation

Input	2	3	add	11	mul	1	add
Stack		3		11		1	
	2	2	5	5	55	55	56



Variables

'hello, world!'

'exo' STORE

\$exo

// Push Hello World String on the Stack

// Store it in a variable called exo

// Then push back exo variable on the stack



What are the available series?

```
[
  $readToken           // Application authentication
  '~.*'               // selector for classname
  {}                  // Selector for labels
]
FIND
```

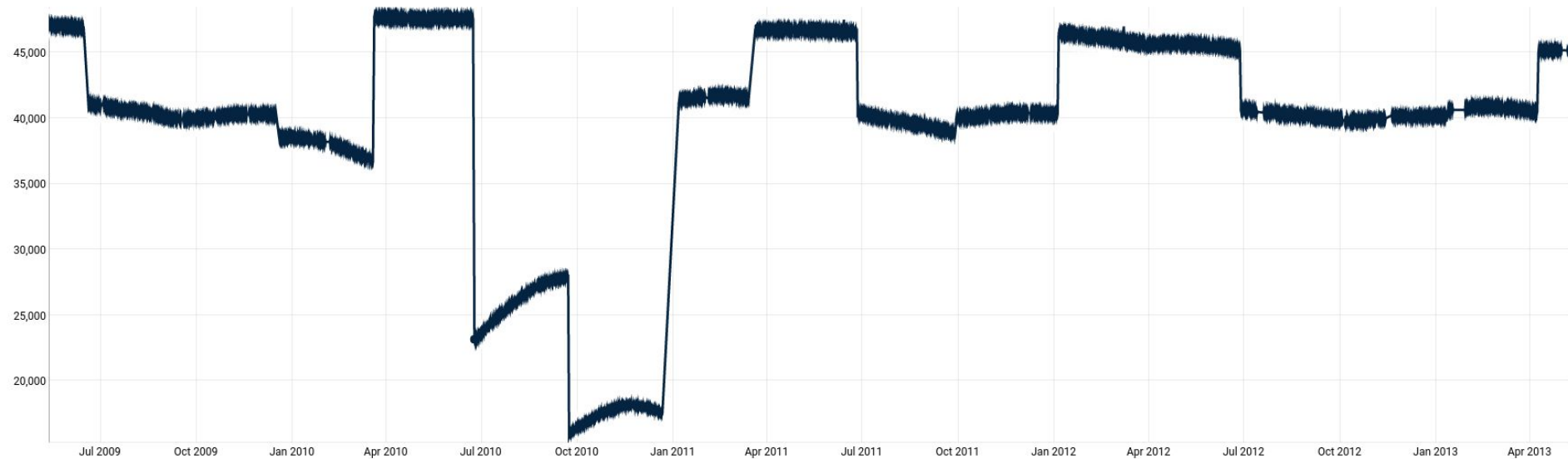


Get raw data

```
[
  $readToken // Application authentication
  'sap.flux' // selector for classname
  { 'KEPLERID' '6541920' } // Selector for labels
  '2009-05-02T00:56:10.000000Z' // Start date
  '2013-05-11T12:02:06.000000Z' // End date
]
FETCH
```



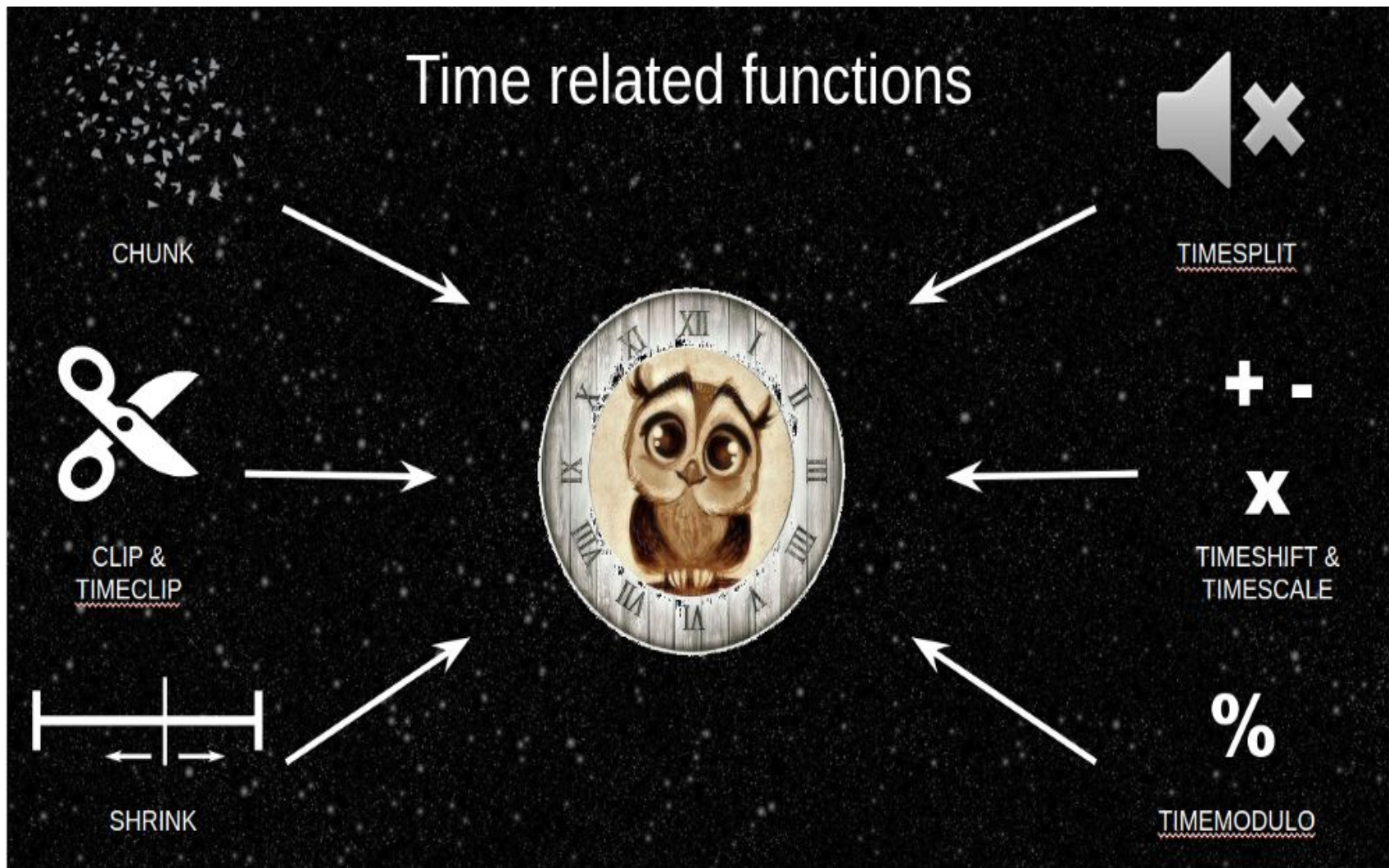
Kepler-11: Raw data



Time manipulation



Time related functions



How to split a Time series

`$gts`

`6 h`

`100`

`'record'`

TIMESPLIT

`// Singleton (or list of) GTS`

`// Minimum of time without data-points`

`// Minimum of data-points required`

`// New labels to subdivide the result`

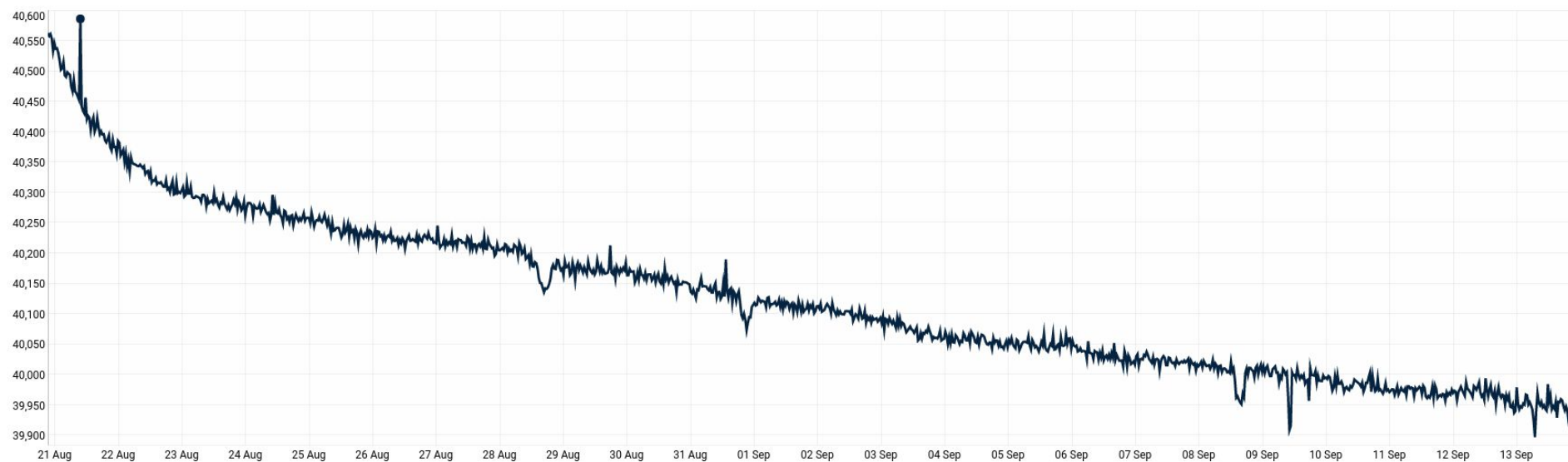


Filtering

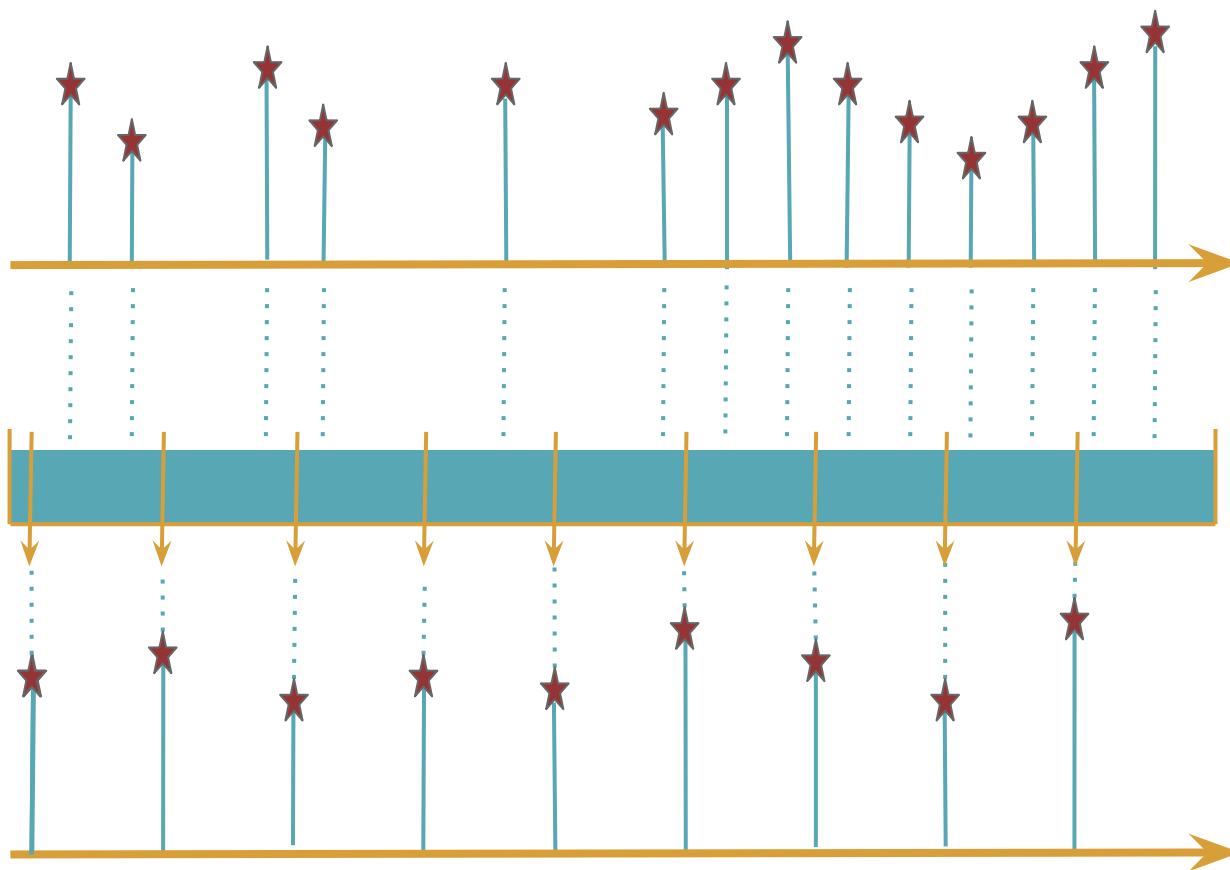
```
[  
  $gts // Singleton (or list of) GTS  
  [] // Equivalence classes  
  { 'record' '5' } // Labels to select  
  filter.bylabels // Type of filter  
]  
FILTER
```



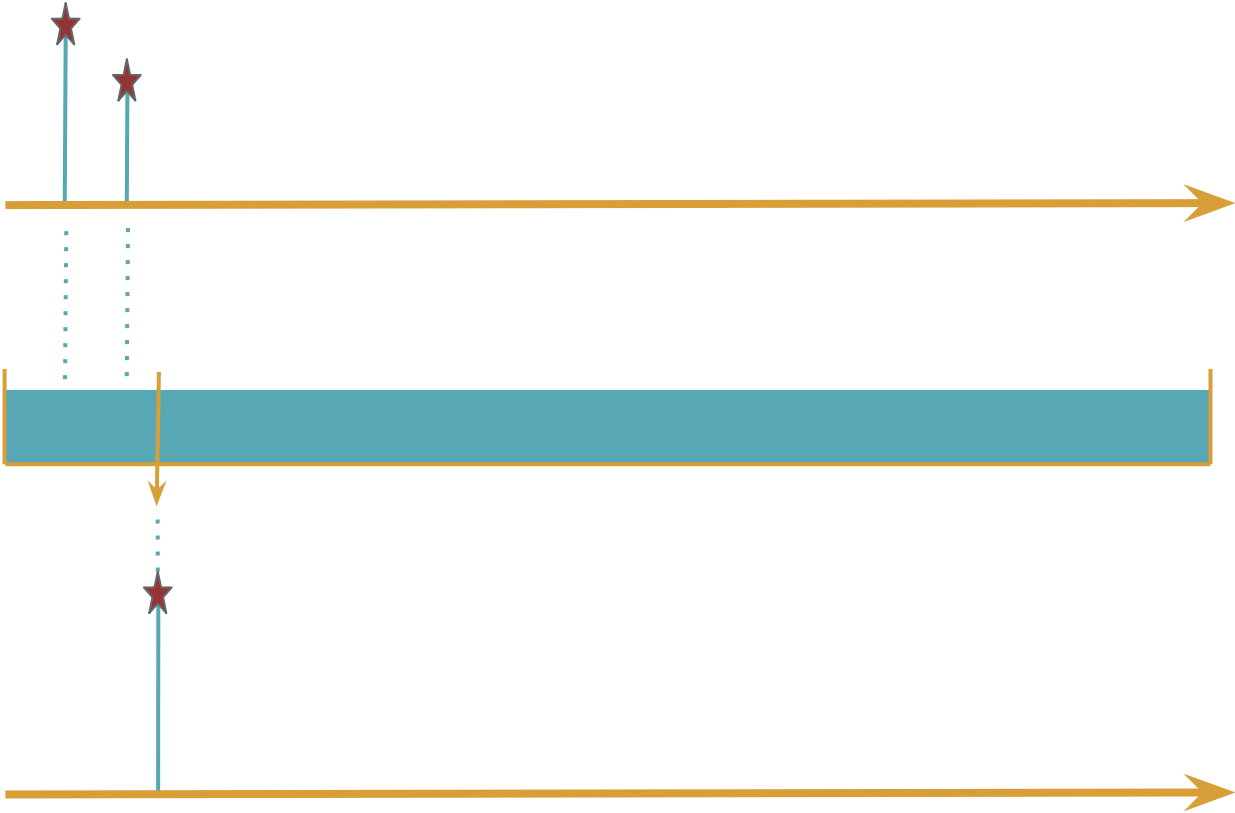
Reference record: 5



Downsampling

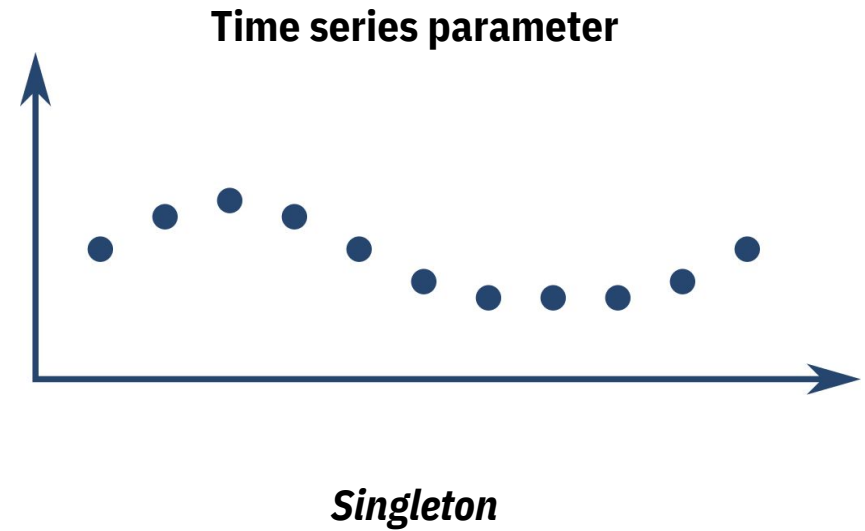


Bucketize



Syntax

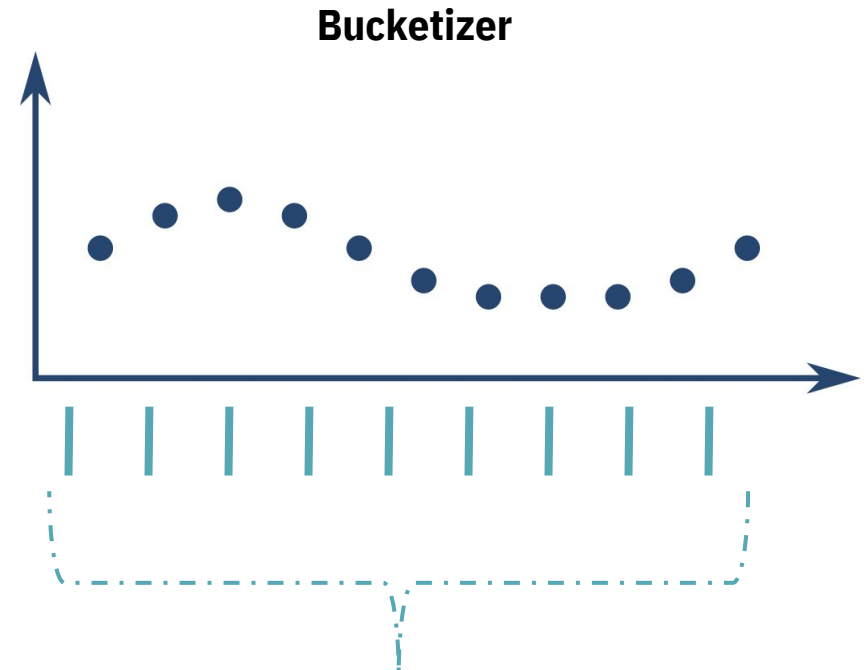
```
[  
  $gts  
  bucketizer.min  
  0  
  2 h  
  0  
]  
BUCKETIZE
```



Syntax

```
[  
  $gts  
  bucketizer.min  
  0  
  2 h  
  0  
]
```

BUCKETIZE

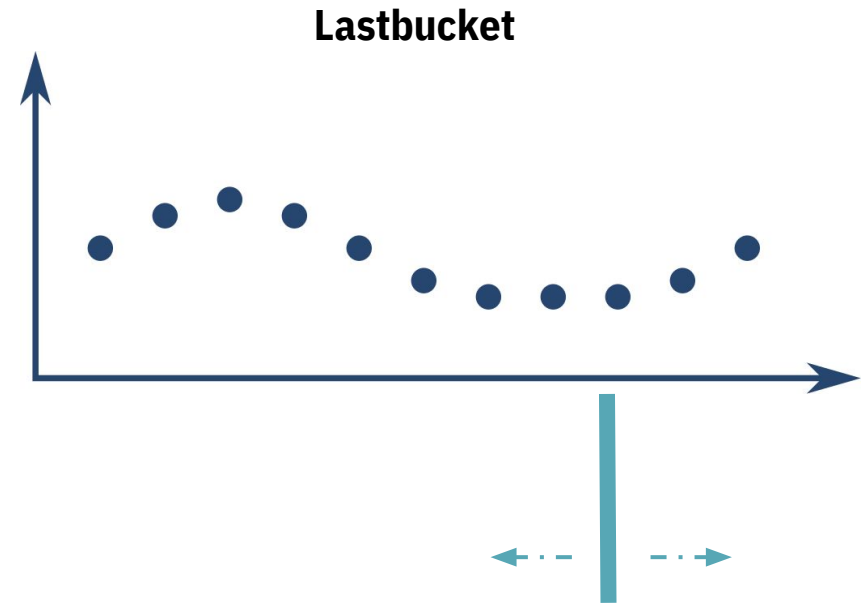


*Type of operator to apply on each bucket
last, max, mean, and, count ...*



Syntax

```
[  
  $gts  
  bucketizer.min  
  0  
  2 h  
  0  
]  
BUCKETIZE
```

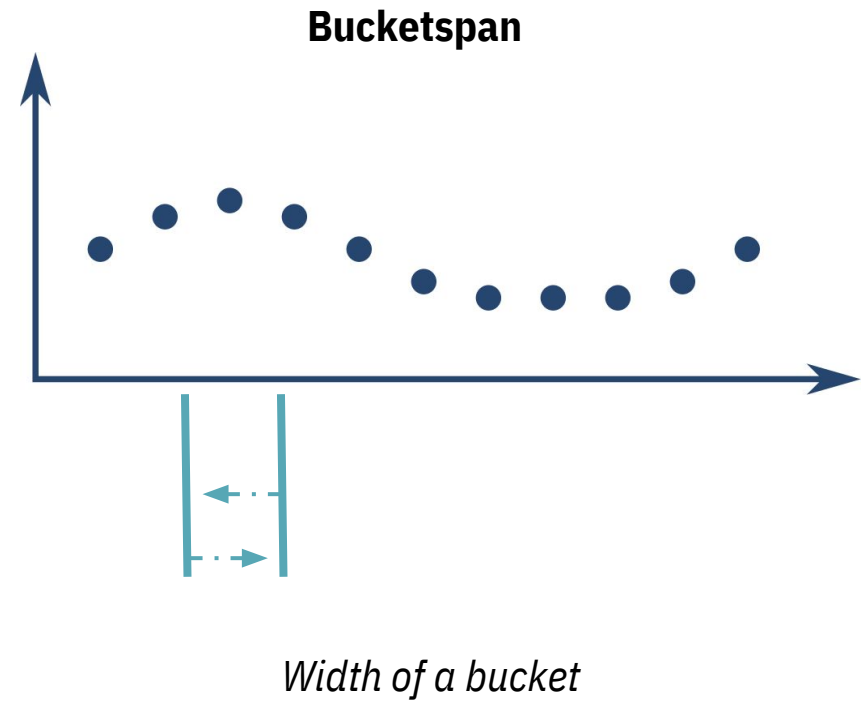


End timestamp of the more recent bucket



Syntax

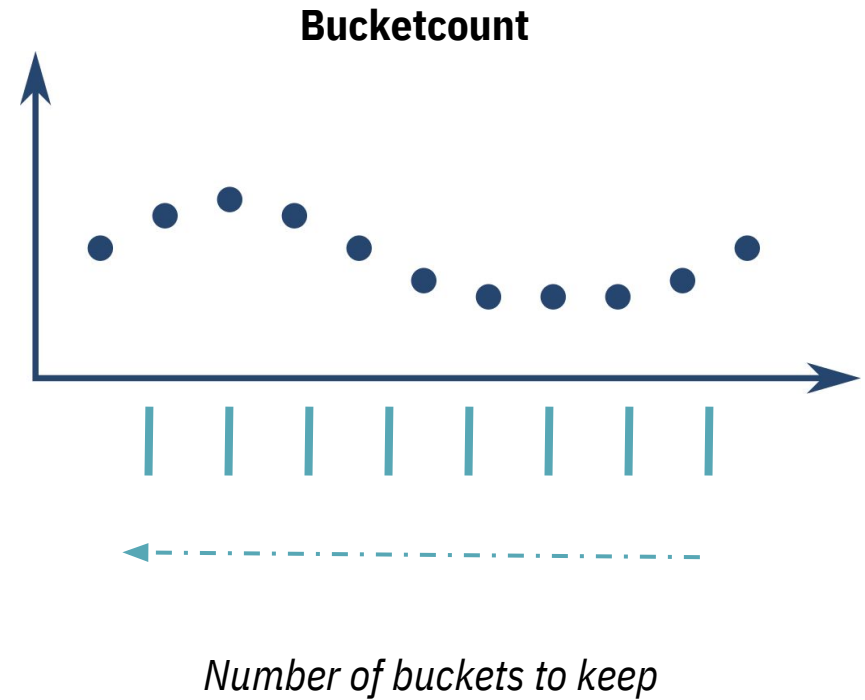
```
[  
  $gts  
  bucketizer.min  
  0  
  2 h  
  0  
]  
BUCKETIZE
```



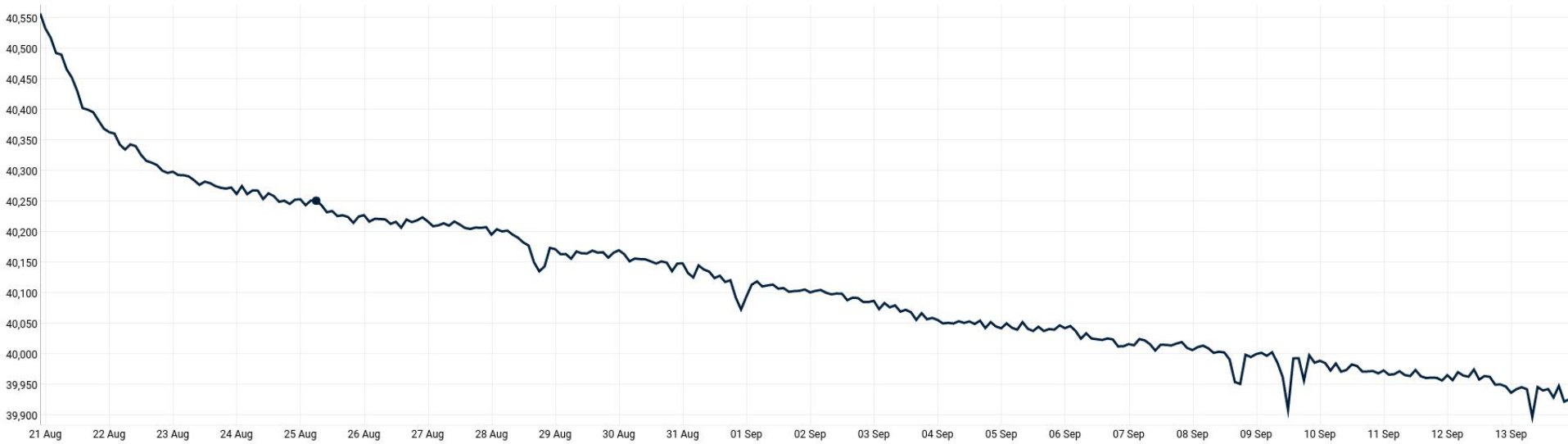
Syntax

```
[  
  $gts  
  bucketizer.min  
  0  
  2 h  
  0  
]
```

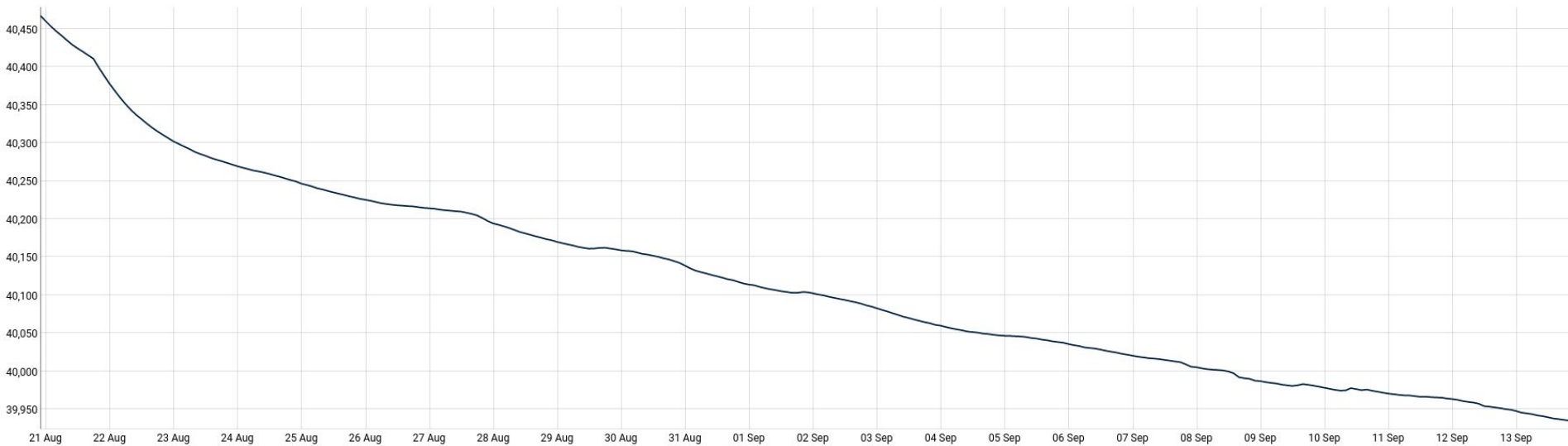
BUCKETIZE



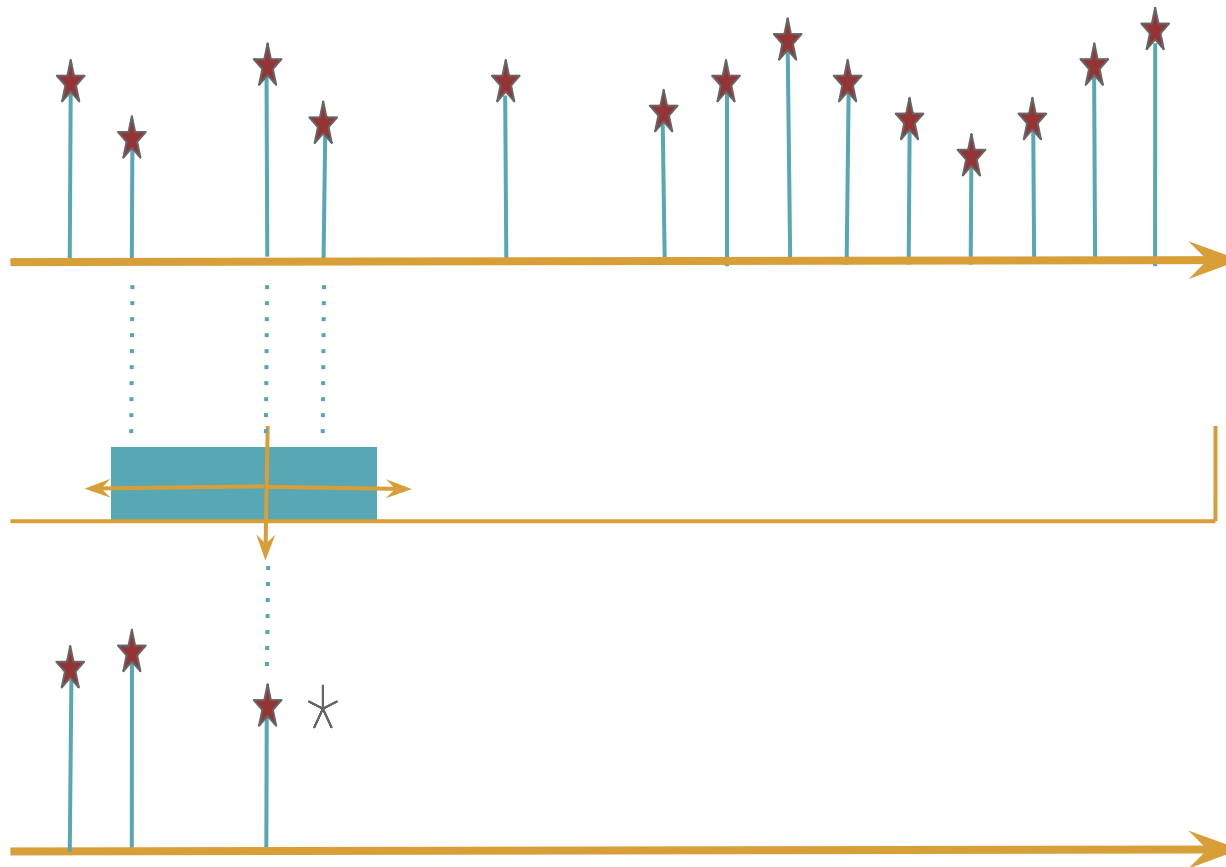
Actual



Trend



Mapper

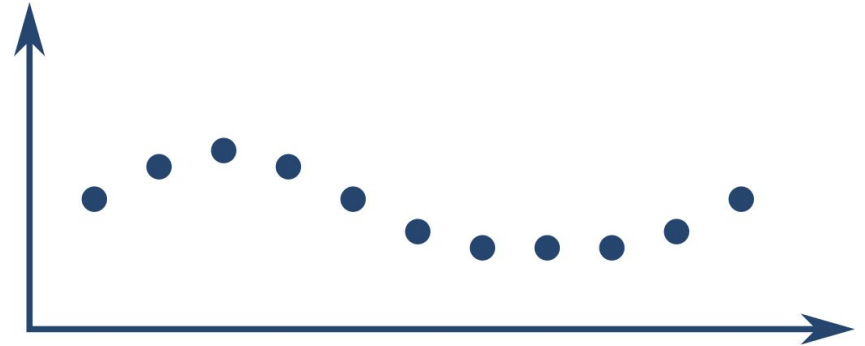


Syntax

```
[  
  $gts  
  mapper.mean  
  2  
  2  
  0  
]  
MAP
```

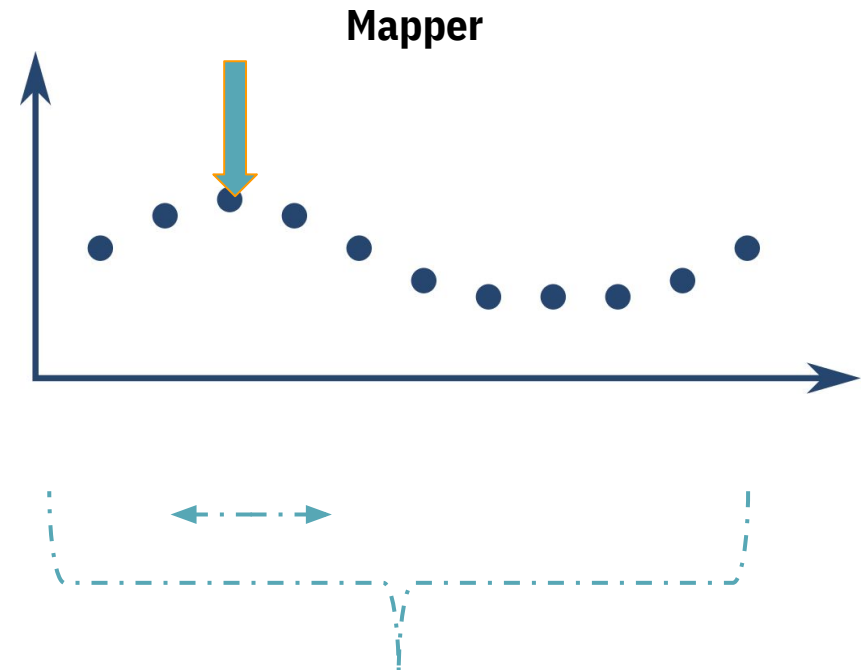


Time series parameter



Syntax

```
[  
  $gts  
  mapper.mean  
  2  
  2  
  0  
]  
MAP
```

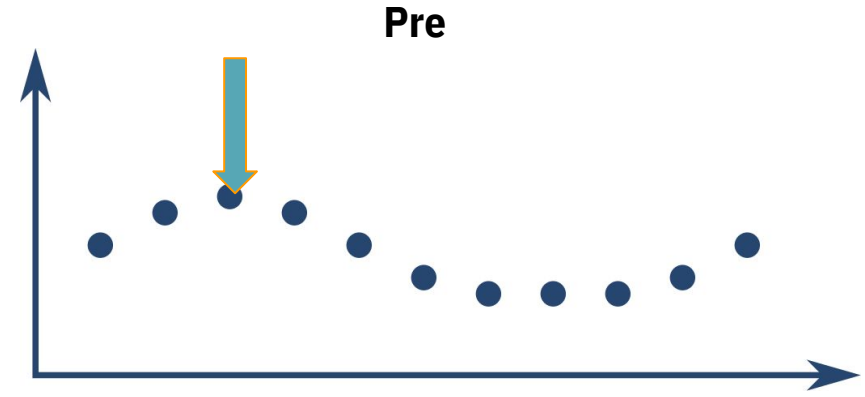


Type of operator to apply on each *window*
add, gt, rate, and, count...



Syntax

```
[  
  $gts  
  mapper.mean  
  2  
  2  
  0  
]  
MAP
```

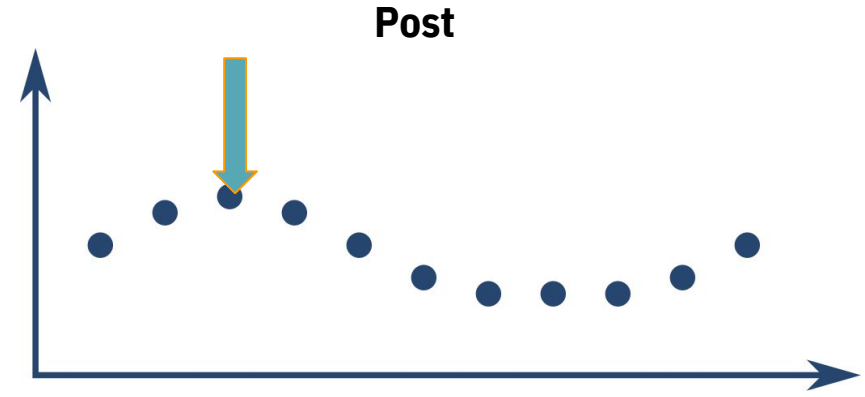


Number of data-points before



Syntax

```
[  
  $gts  
  mapper.mean  
  2  
  2  
  0  
]  
MAP
```

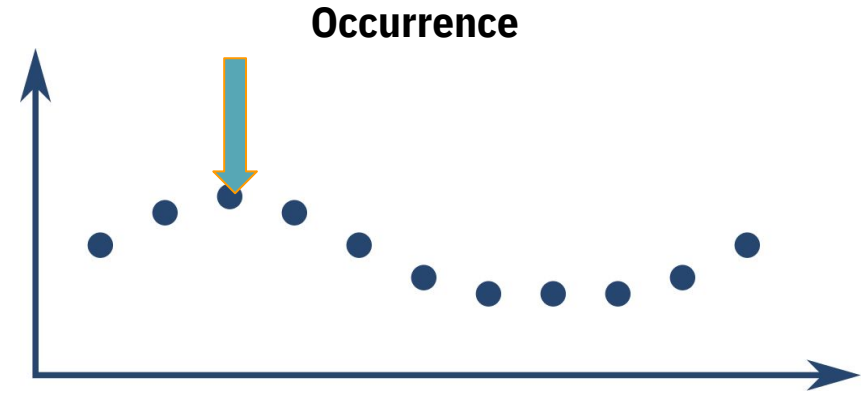


Number of data-points after



Syntax

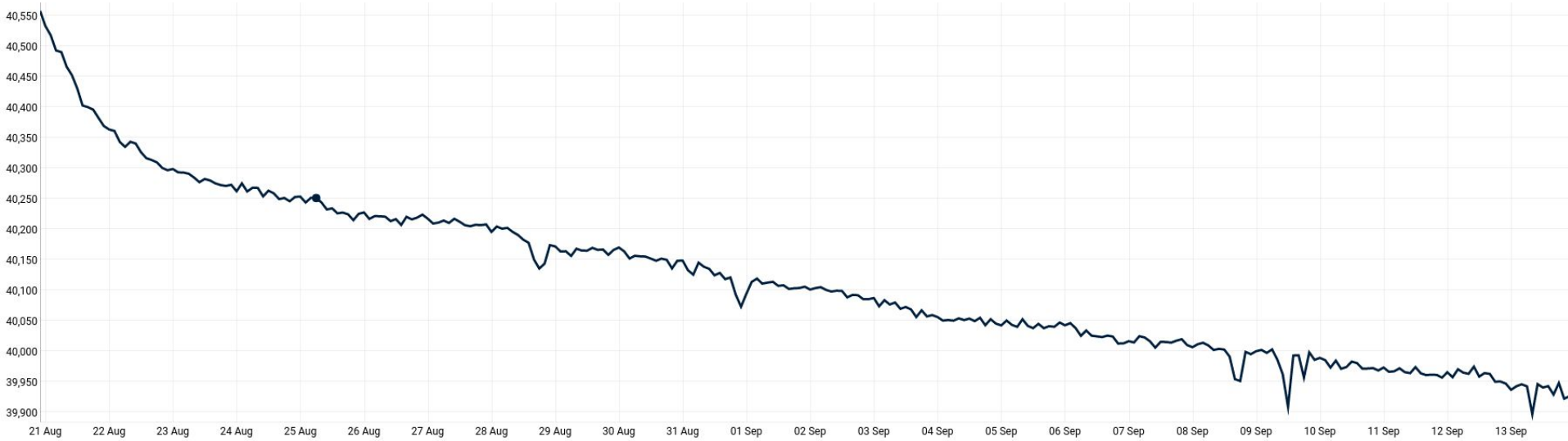
```
[  
  $gts  
  mapper.mean  
  2  
  2  
  0  
]  
MAP
```



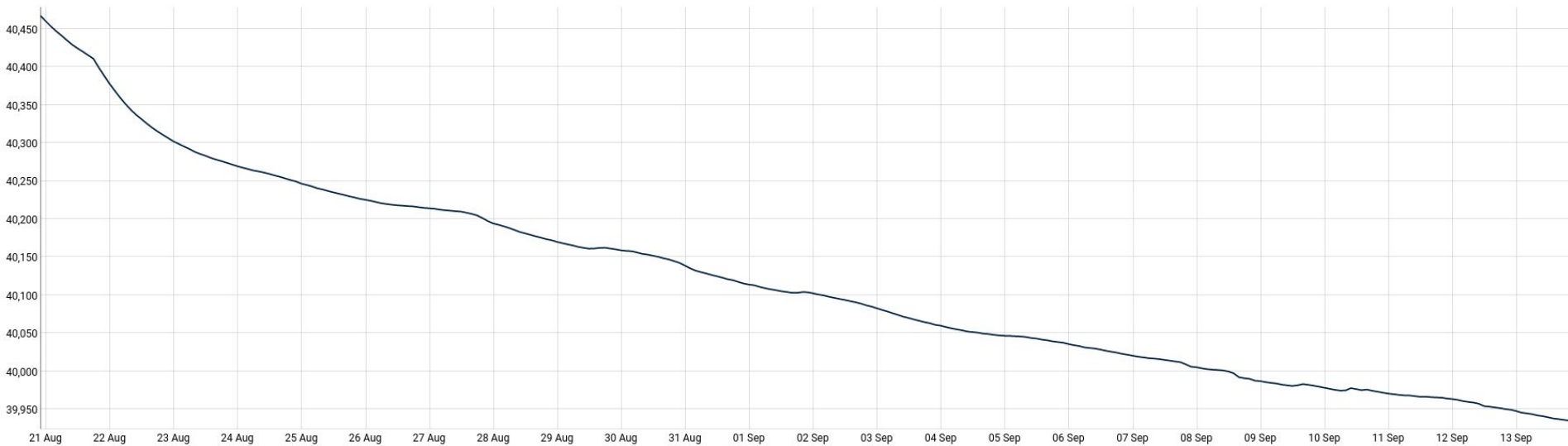
Maximal number of calculation for a data-point



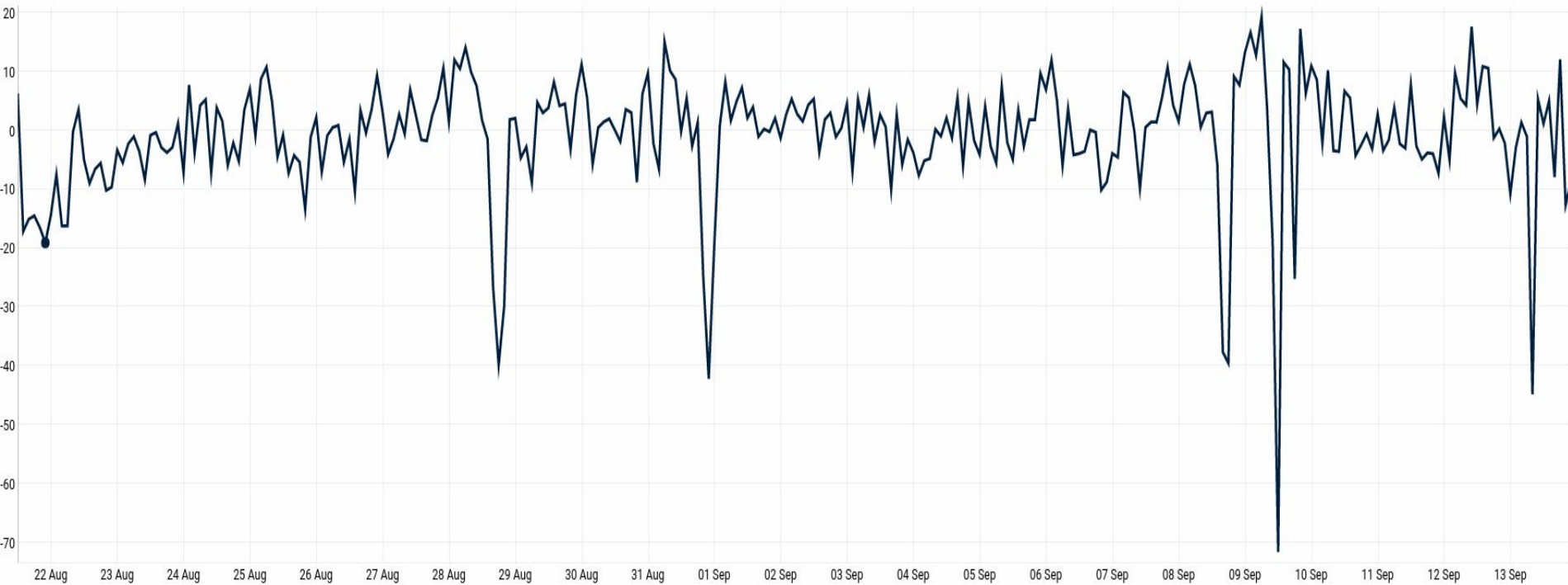
Actual



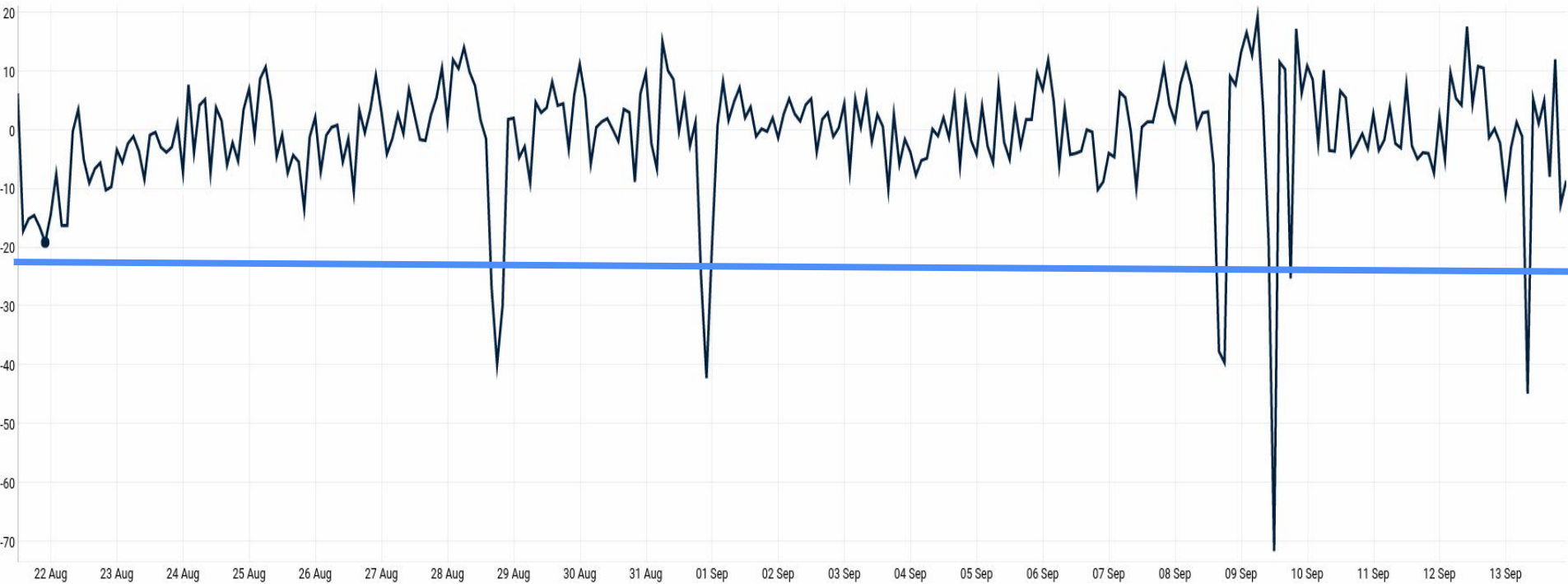
Trend



Actual - trend



Actual - trend



Time to level-up!



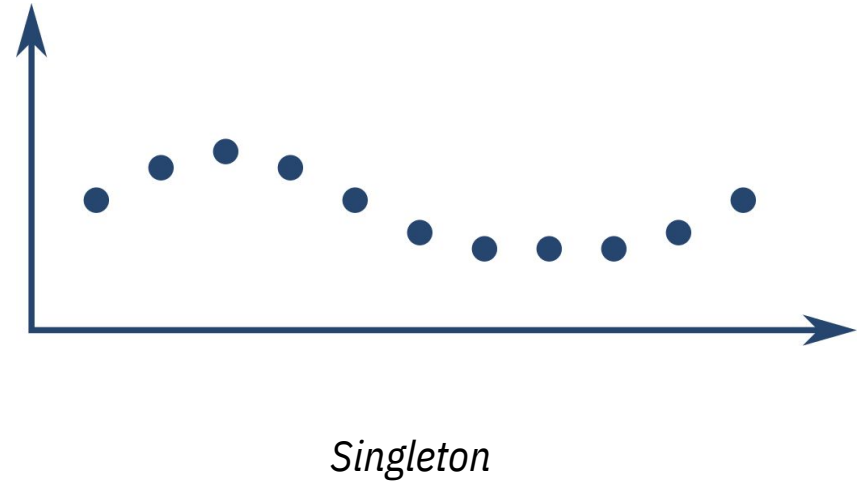
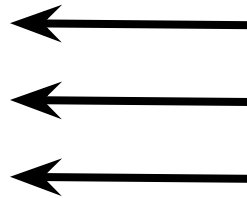
Time series operation

```
[  
    $gts0           // First series pull  
    ...           // ...  
    $gtsN         // N series pull  
    [ 'record' ]  // Key labels list  
    op.add        // Type of operator  
]  
APPLY
```



Syntax

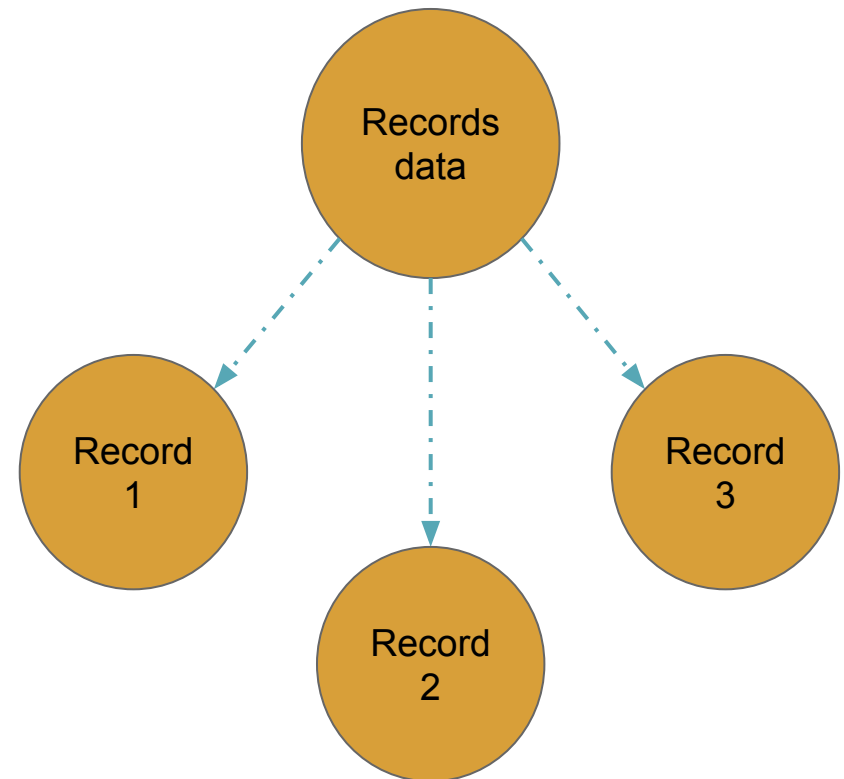
```
[  
  $gts0  
  ...  
  $gtsN  
  [ 'record' ]  
  op.add  
]  
APPLY
```



Syntax

Equivalence class

```
[  
  $gts0  
  ...  
  $gtsN  
  ['record']  
  op.add  
]  
APPLY
```

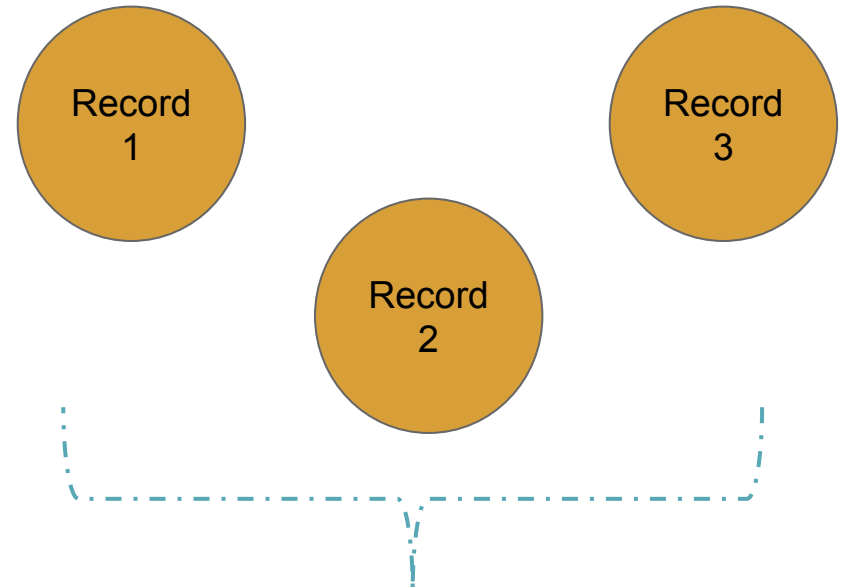


Syntax

```
[  
  $gts0  
  ...  
  $gtsN  
  [ 'record' ]  
  op.add  
]  
APPLY
```



Operator



*Type of operator to apply on each **class**
sub, gt, mask, and, mul ...*



Final result

