Rediscover the known Universe with NASA datasets

Horacio Gonzalez
@LostInBrittany

Kevin Georges
@0xd33d33

Horacio Gonzalez

web2day
DIGITAL SPRINGBREAK

@LostInBrittany

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

OVH
Team DevRel

Breizh C@mp
Mix de technologies

la cantine BREST
Finist Devs

WARP 10
MEETUP

Google Developers Experts 2015
Web Technologies GDE

Kevin Georges

web2day
DIGITAL SPRINGBREAK

@0xd33d33

Tech Leader
OVH Metrics

OVH
Innovation is Freedom

HelloExoWorld

web2day
DIGITAL SPRINGBREAK

Looking for exoplanets in NASA datasets

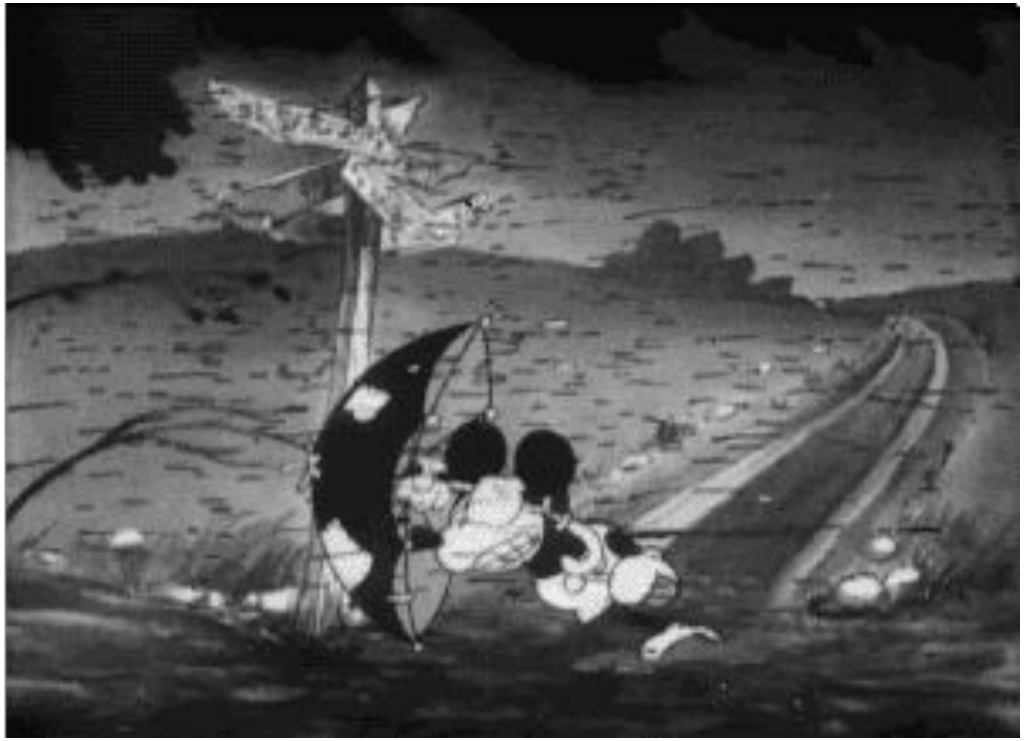
HelloExoWorld

Once upon a time...

An amateur astronomer

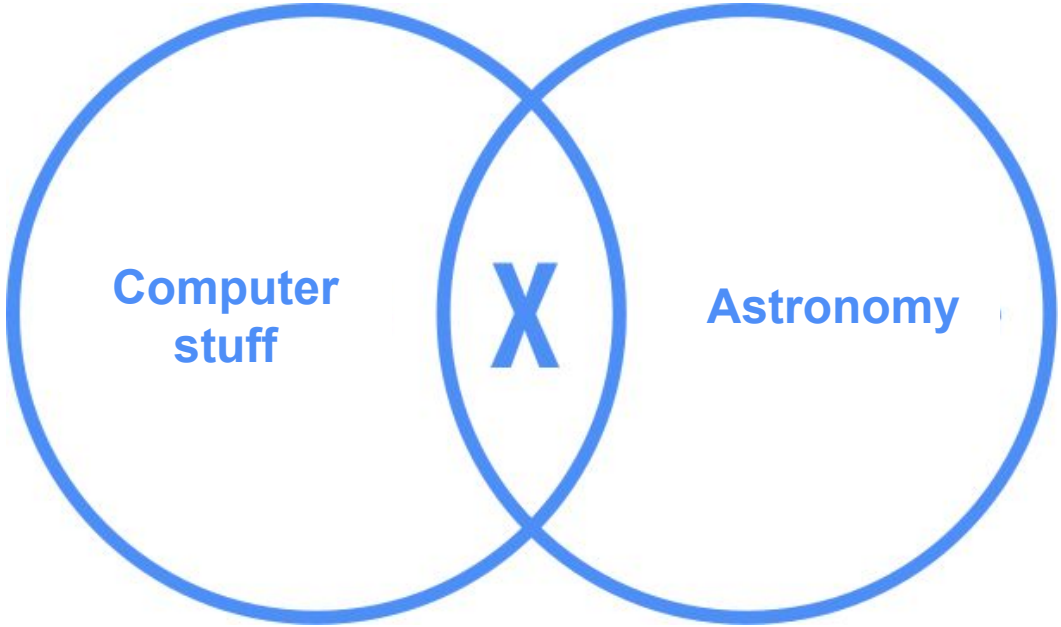
Pierre Zemb, DevOps OVH

What not to do if you love astronomy



Live in Brest

Looking for solutions

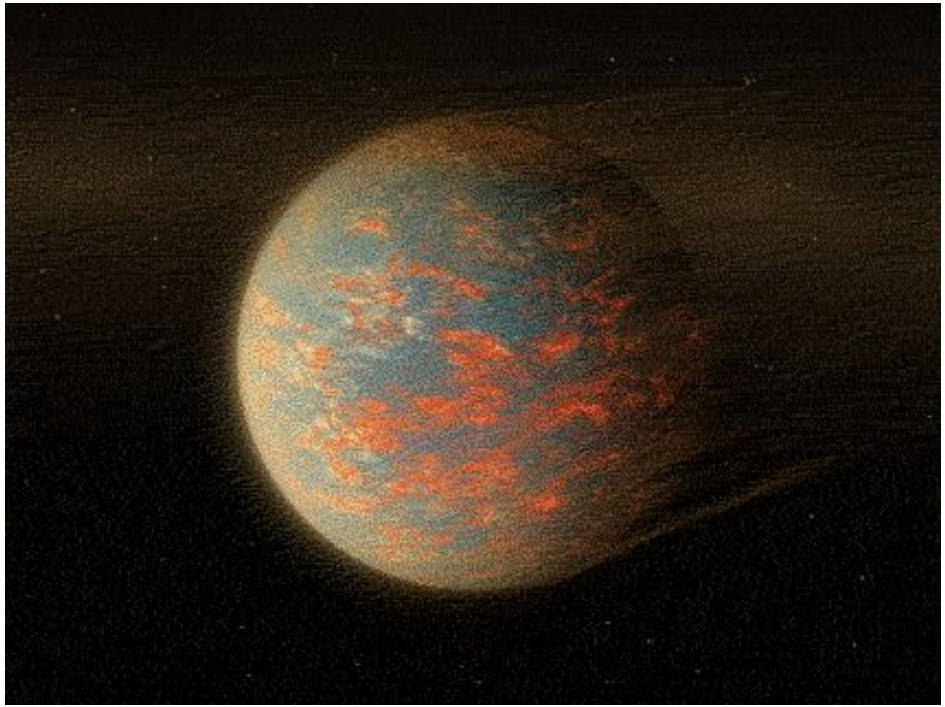


Mixing passions

Google is your friend...

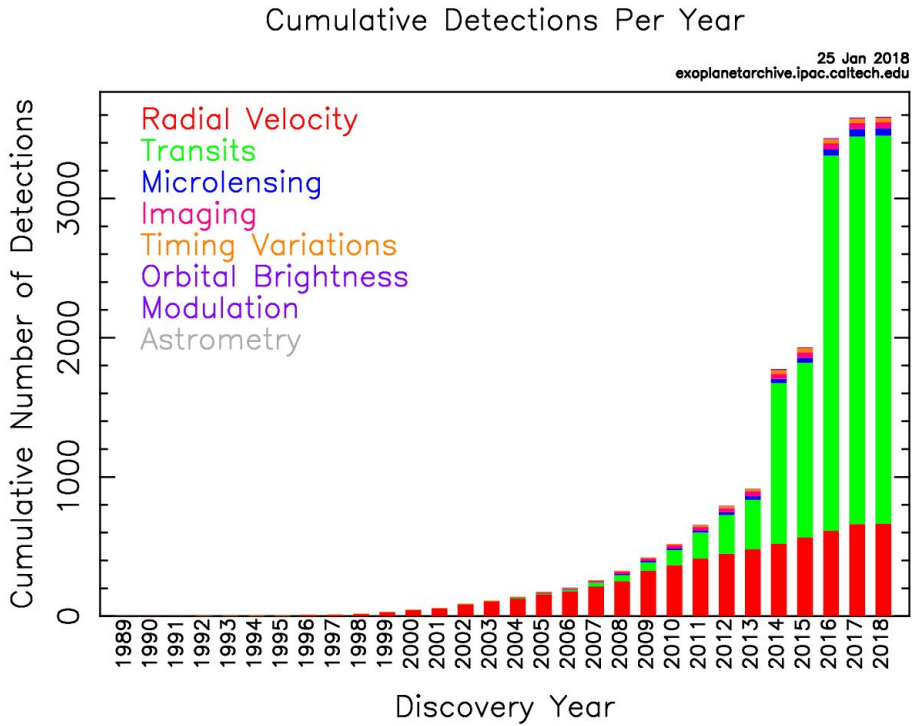
Let's find a project

Exoplanets?



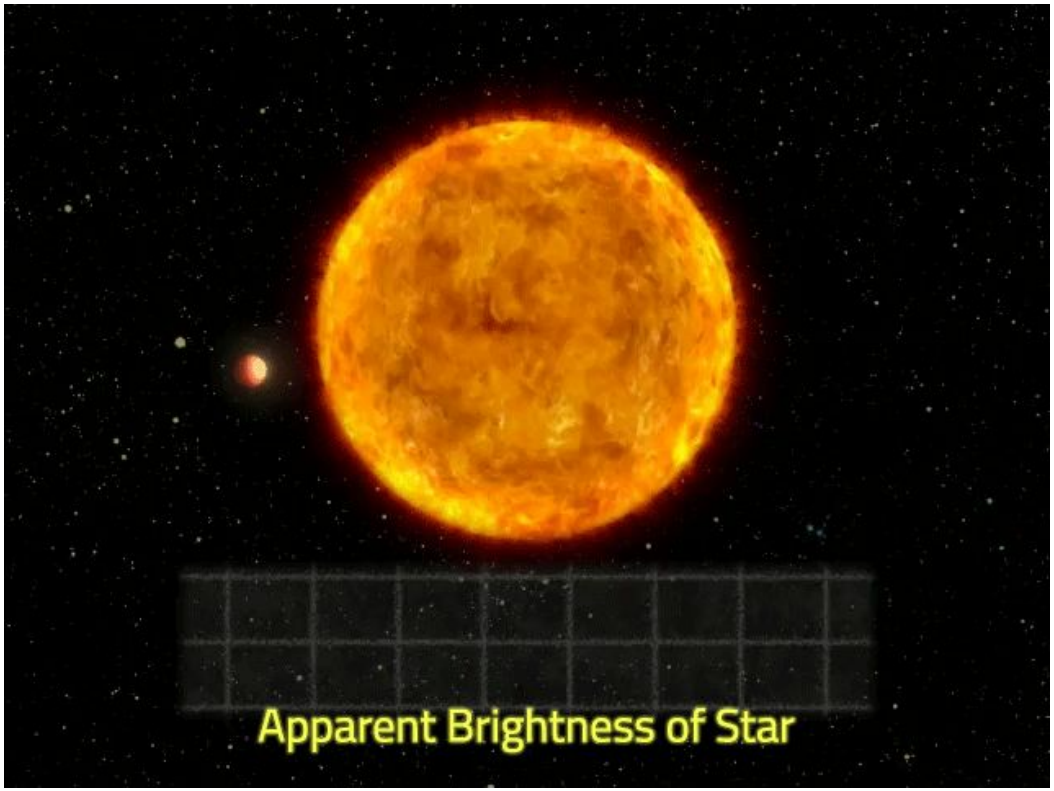
Planets orbiting stars far away

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?

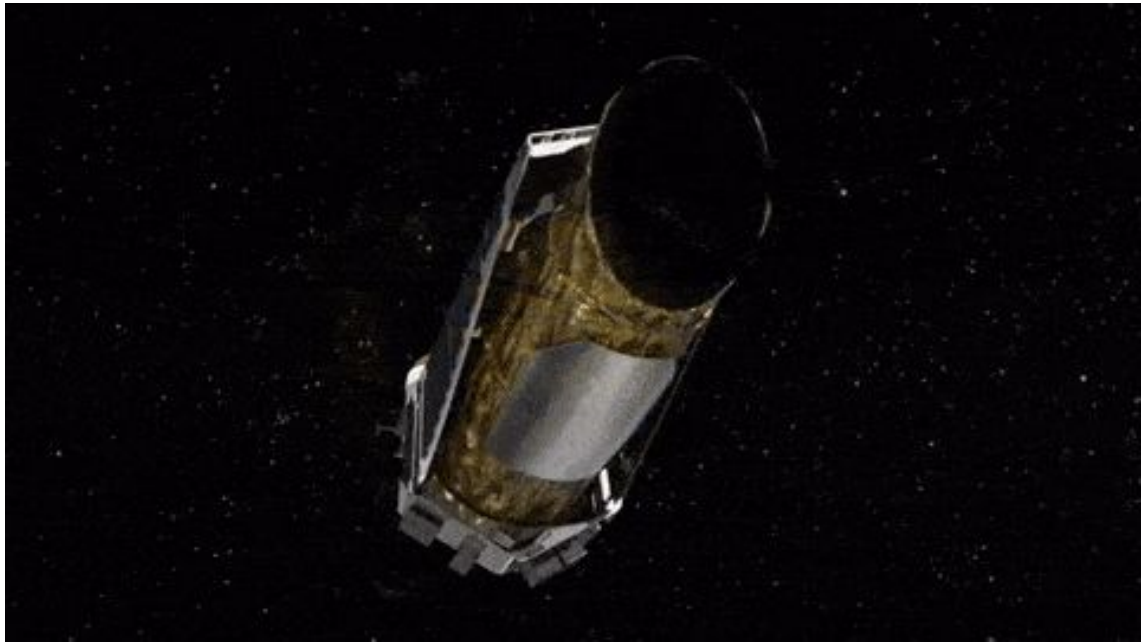
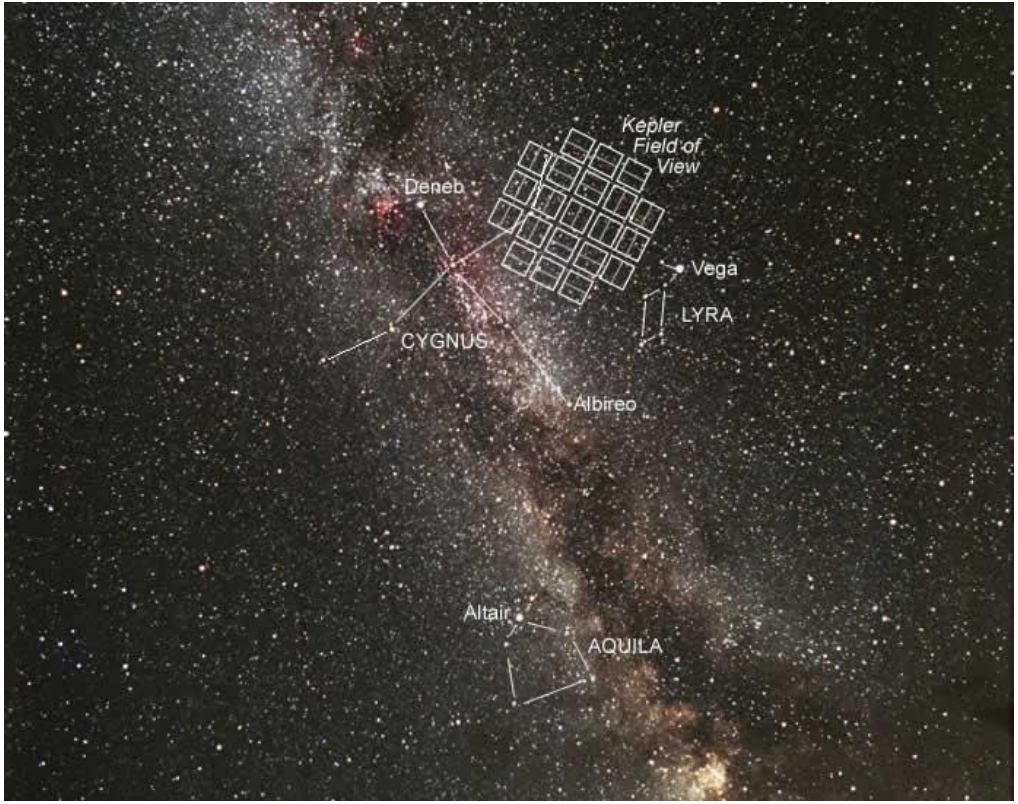


Image credits : NASA

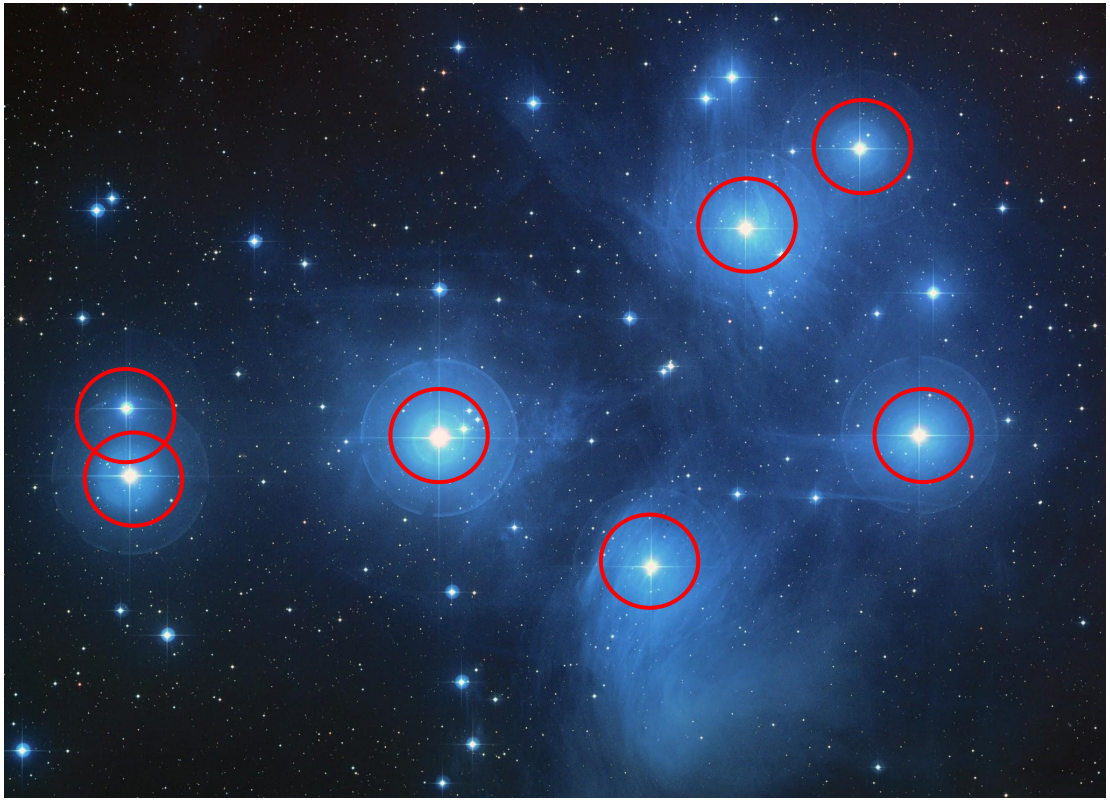
Kepler

Watching the sky



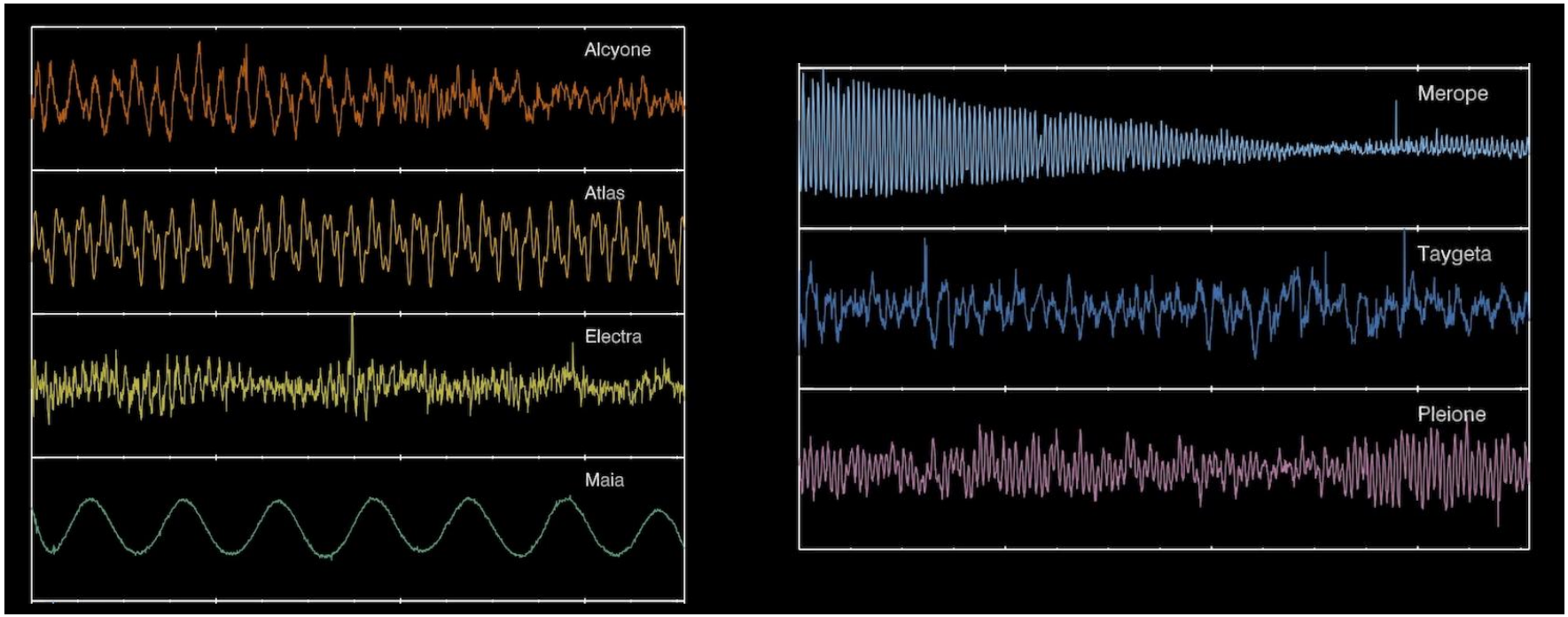
By Carter Roberts [Public domain], via Wikimedia Commons

And what kind of data we get?



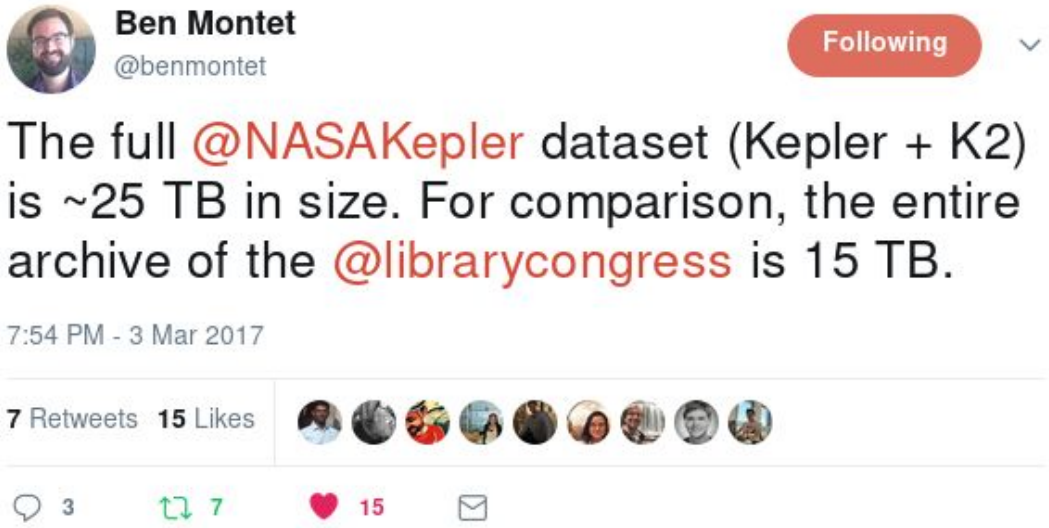
Pleiades By NASA, ESA, AURA/Caltech, Palomar Observatory. Via Wikimedia Common

Well, that's the problem



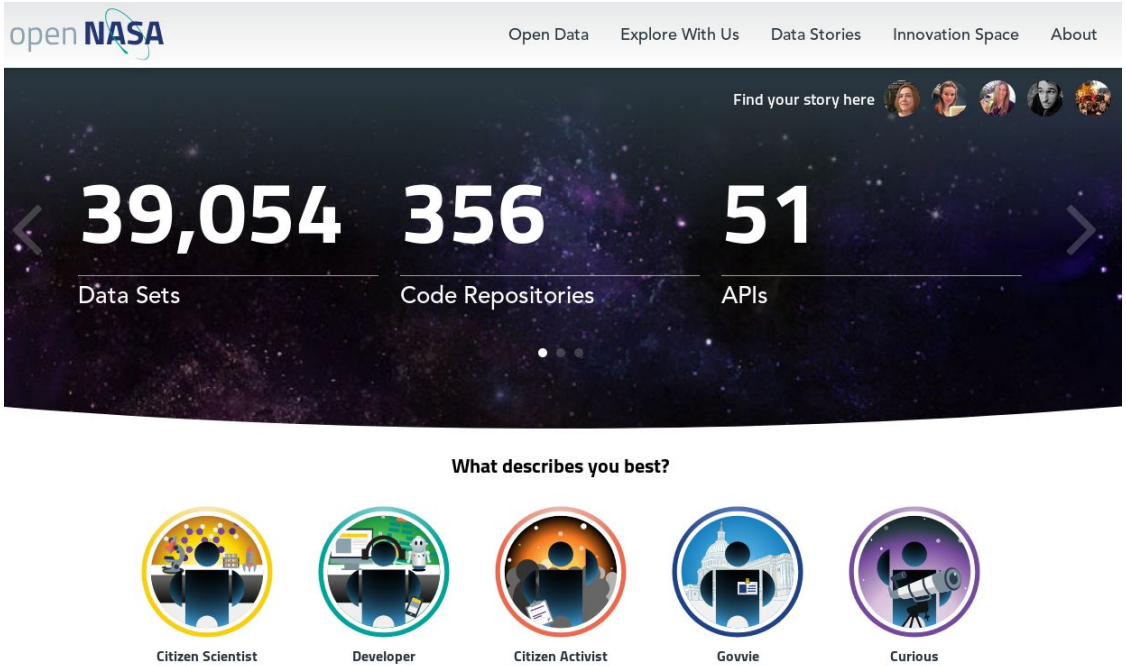
Seven stars, seven different profiles

Kinda big data



Over 40 million light curves

Big AND open data



Lots of datasets in #opendata

And we can help with that!

Let's use our tools to analyse the data

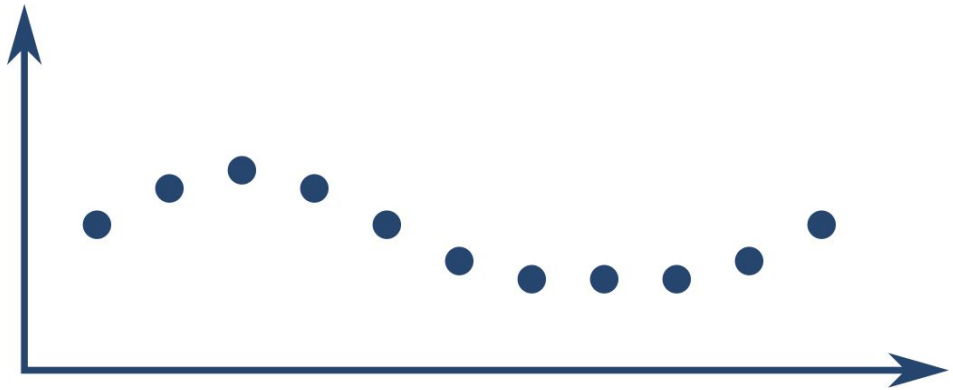
Time Series

To analyse Kepler datasets

Kepler: spatial Time Series

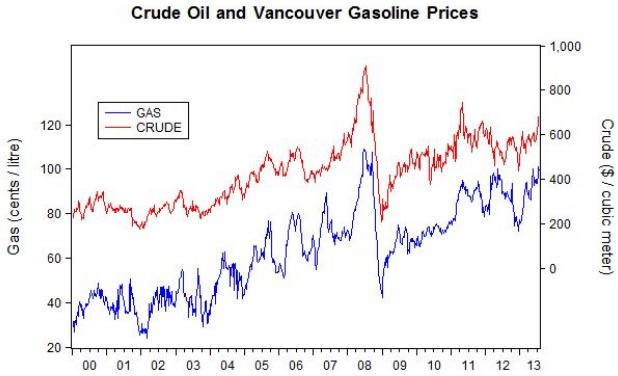
Definition of Time Series:

A series of data points indexed in time order



Time Series

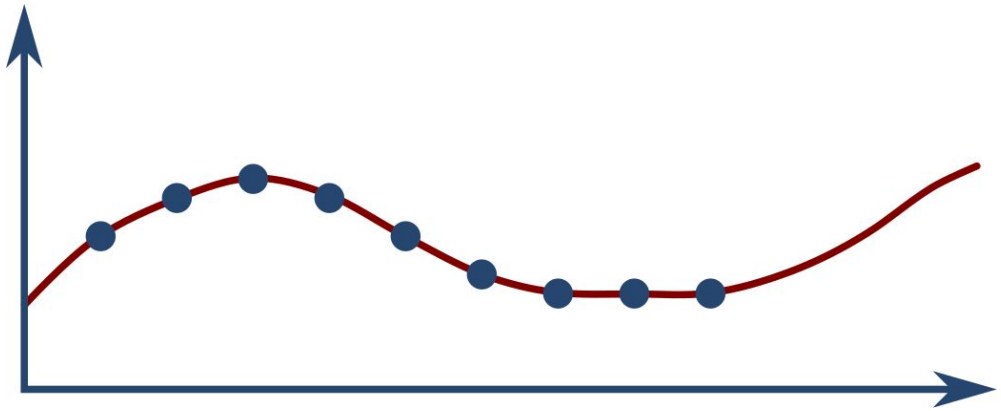
- Stock Market Analysis
- Economic Forecasting
- Budgetary Analysis
- Process and Quality Control
- Workload Projections
- Census Analysis
- ...



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:

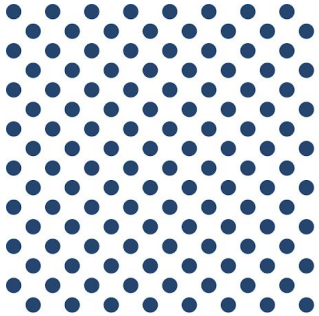
- Moving average
- ARMA (AutoRegressive Moving Average)
- Multivariate ARMA models
- ARCH (AutoRegressive Conditional Heteroscedasticity)
- Dynamic time warping
- ...

Time Series

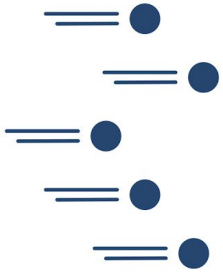
Specific application of general tools

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

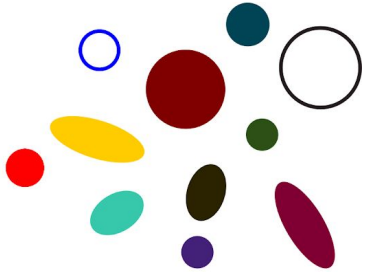
Dealing with Time Series



Volume



Velocity

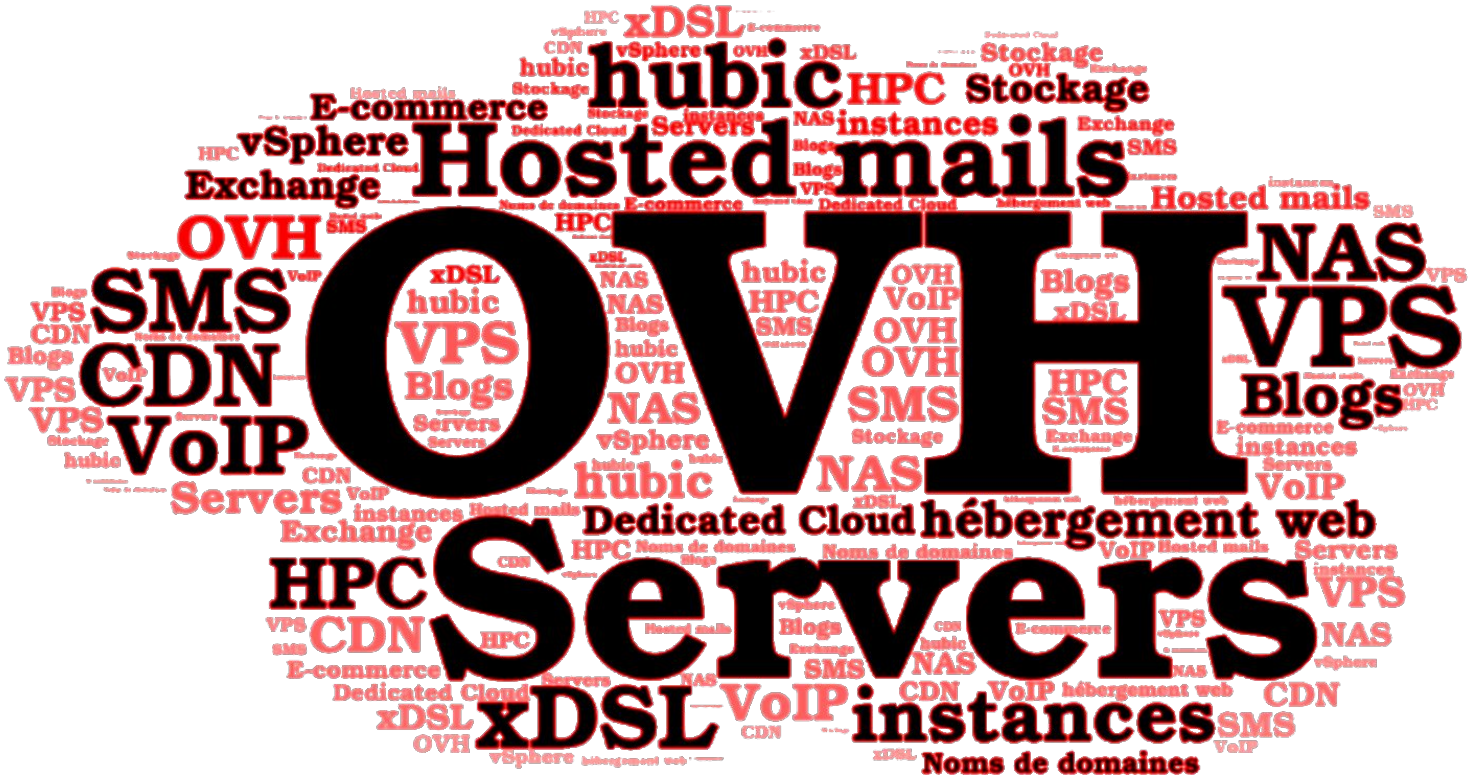


Variety

The 3 'v'

Monitoring OVH with Time Series

web2day
DIGITAL SPRINGBREAK



Metrics Data Platform

web2day DIGITAL SPRINGBREAK

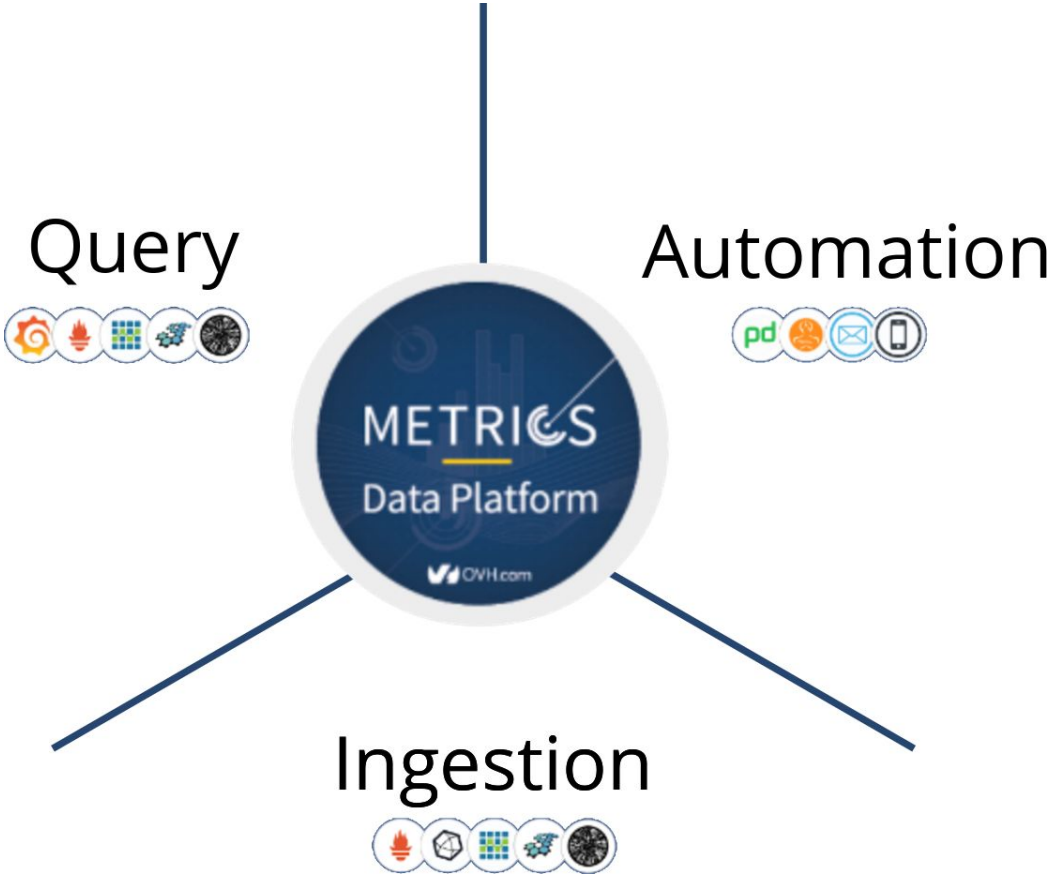
Some of Metrics' metrics:

- 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

Analytics is the key to success

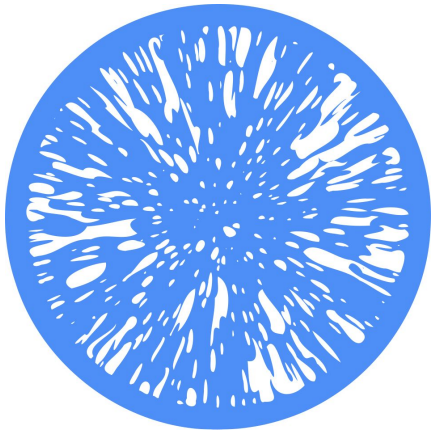
Fetching data is only the tip of the iceberg

A true Time Series analysis toolbox

- Hundreds of functions
- Manipulation frameworks
- Analysis workflow

A match made in heaven

Warp 10, OVH Metrics and HelloExoWorld



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

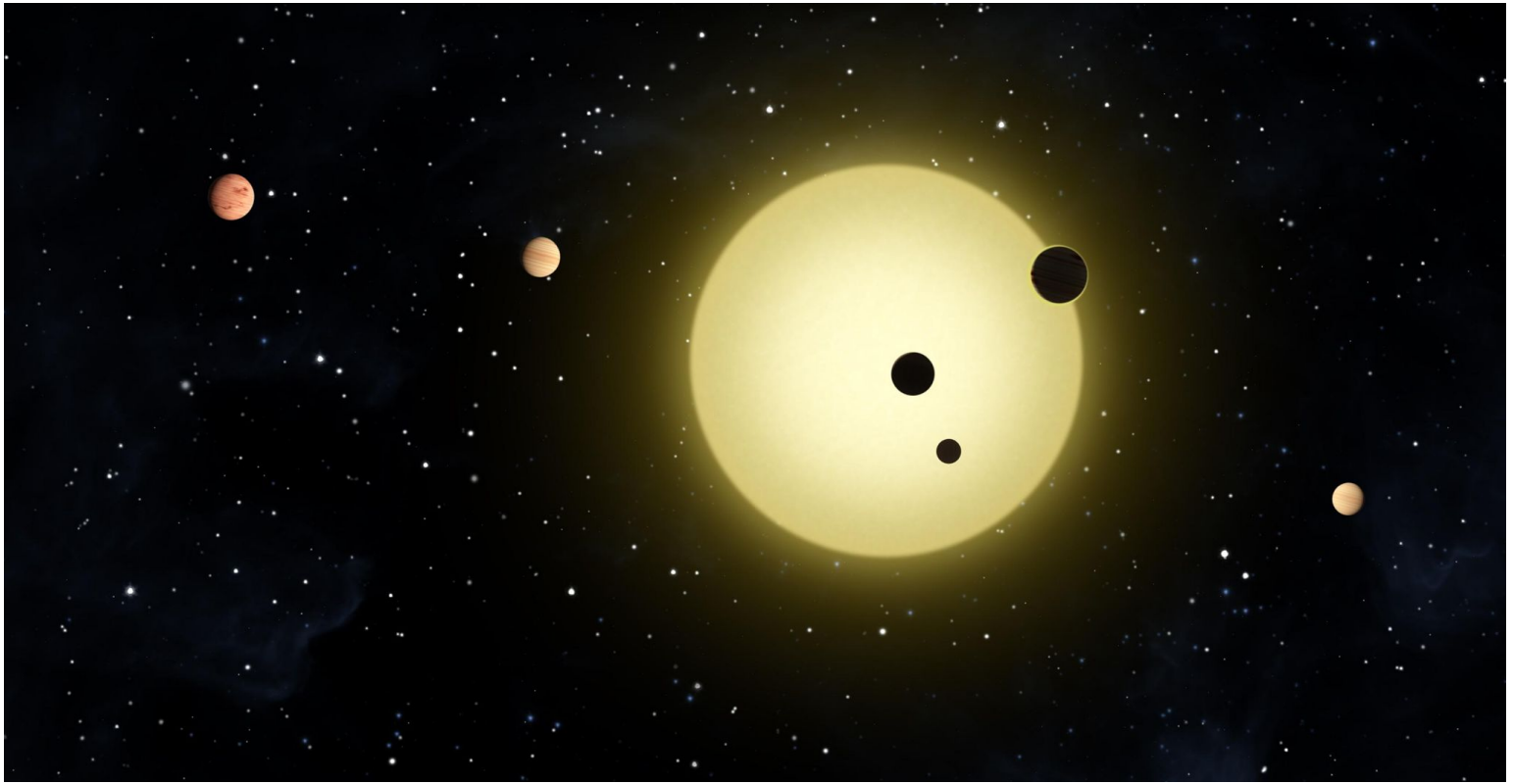
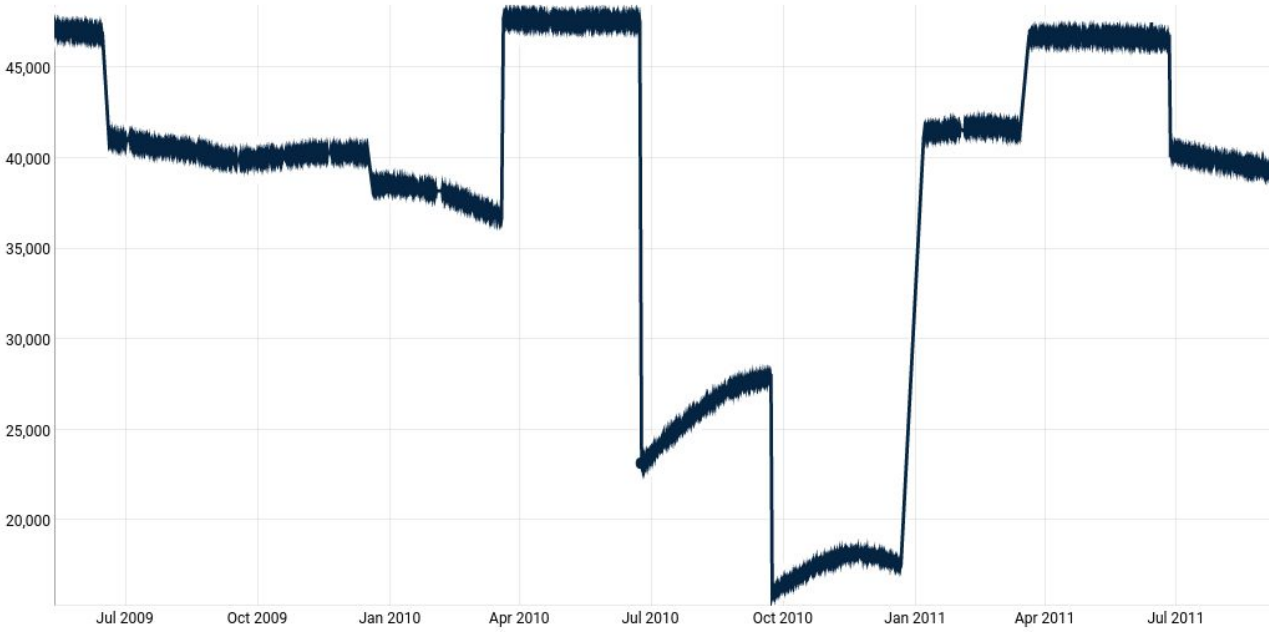


Image credit: NASA/Tim Pyle

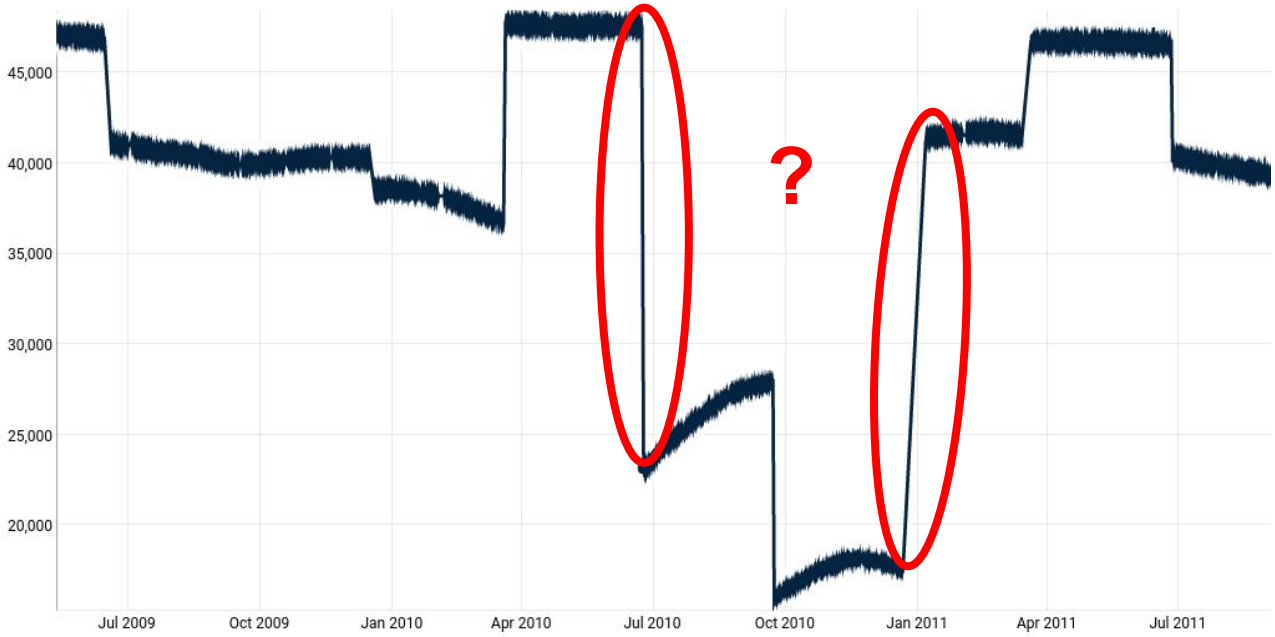
Looking at the raw signal...



SAP_FLUX:

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

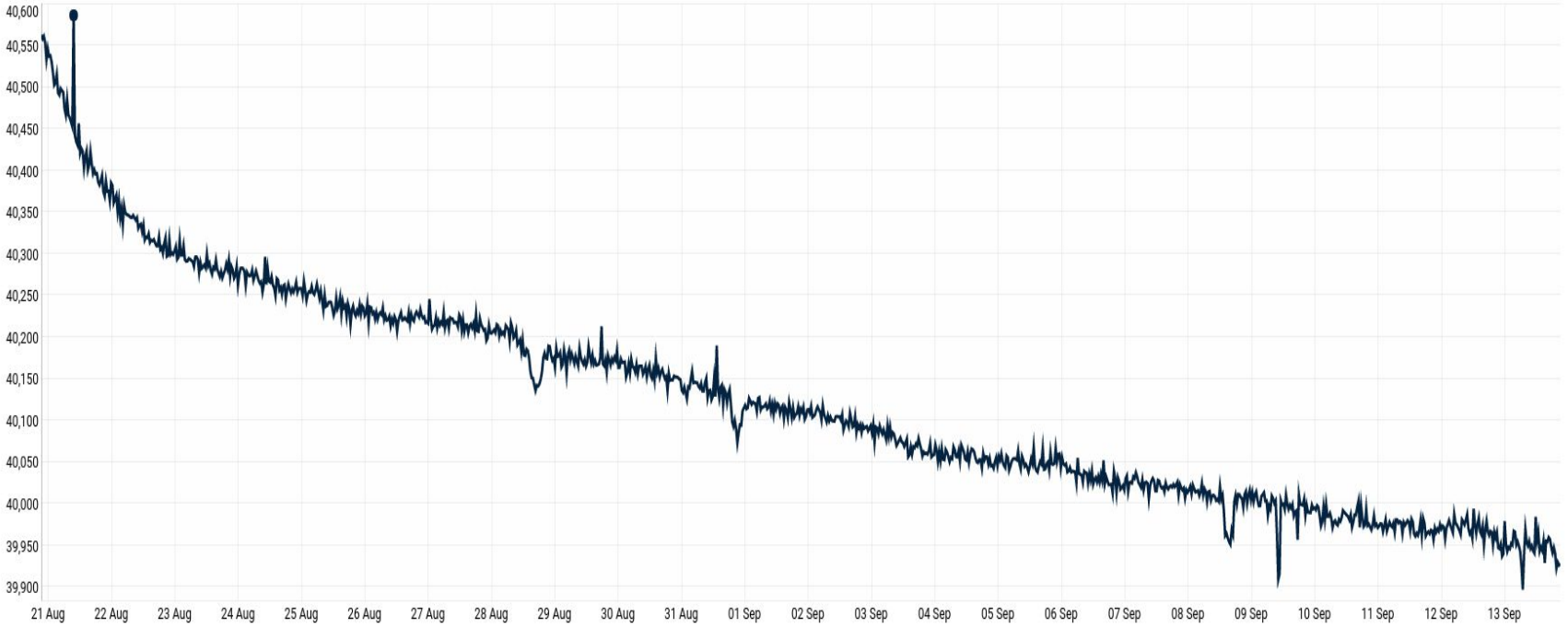
Looking at the raw signal...



SAP_FLUX:

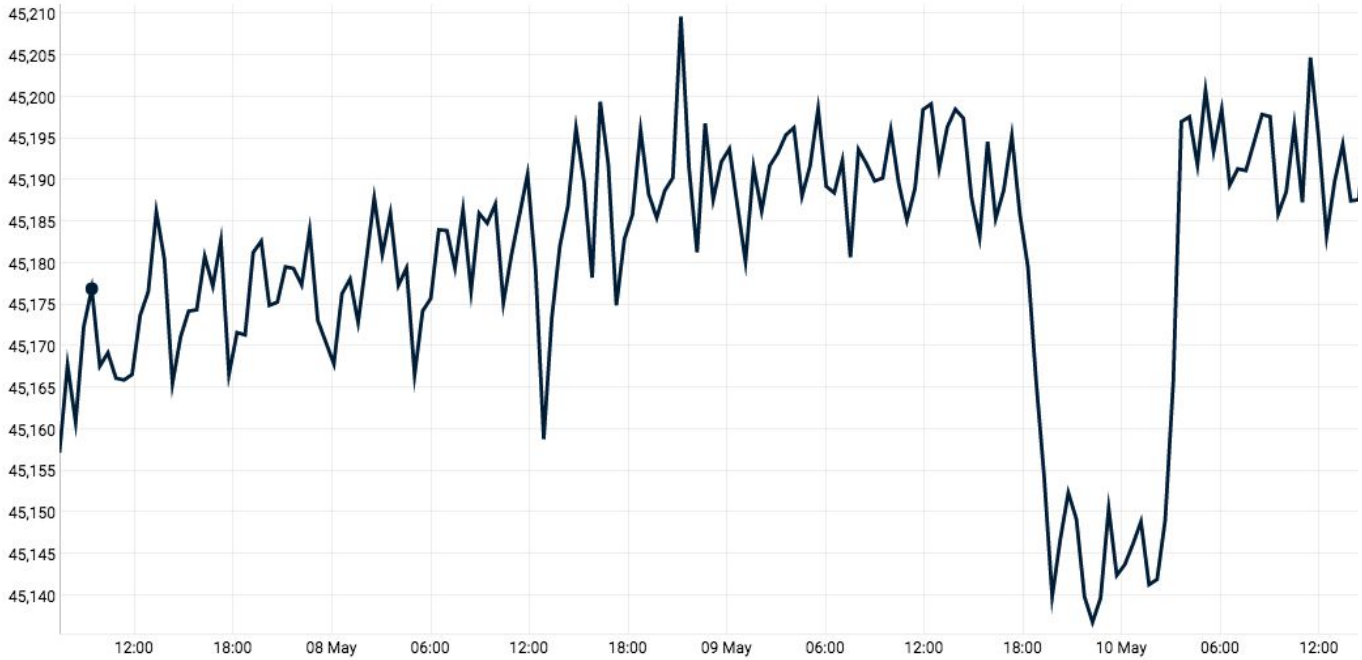
The flux in units of electrons per second contained in the optimal aperture pixels collected by the spacecraft.

Looking at one record



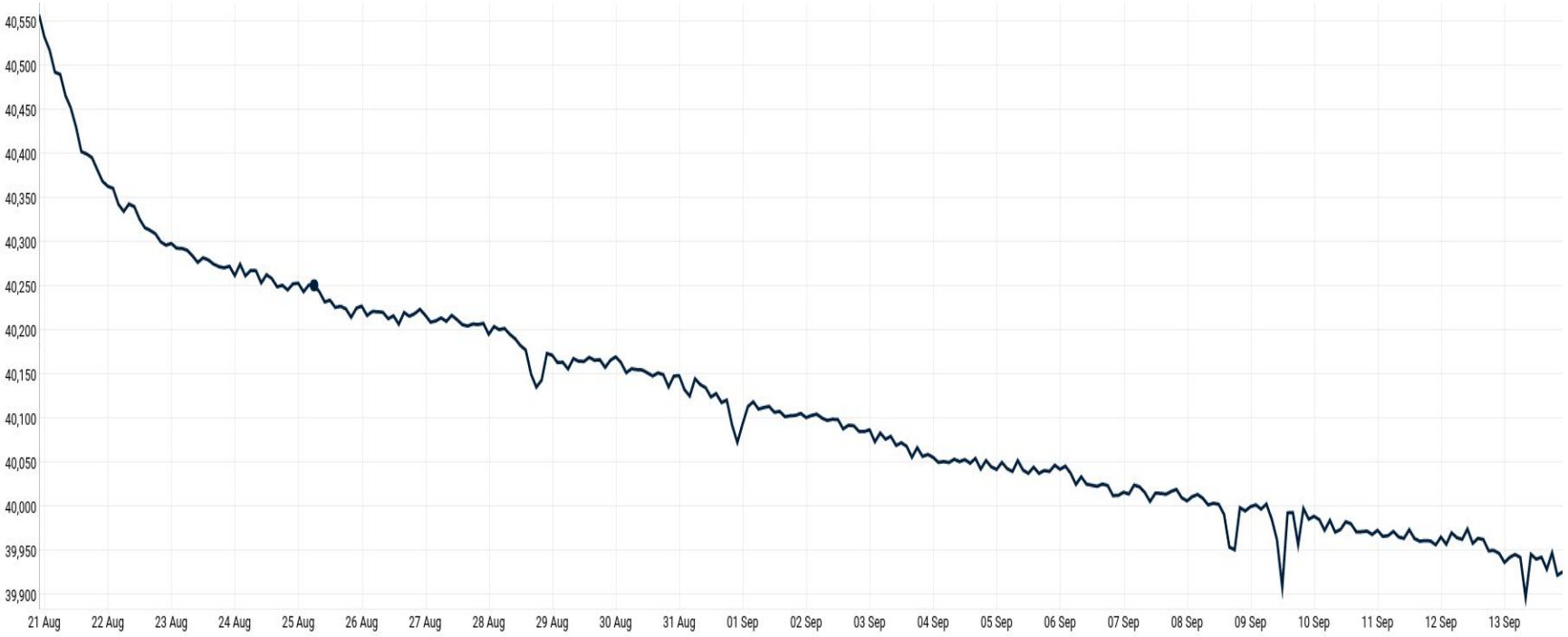
Perturbations in dirty signals

Transits are tiny

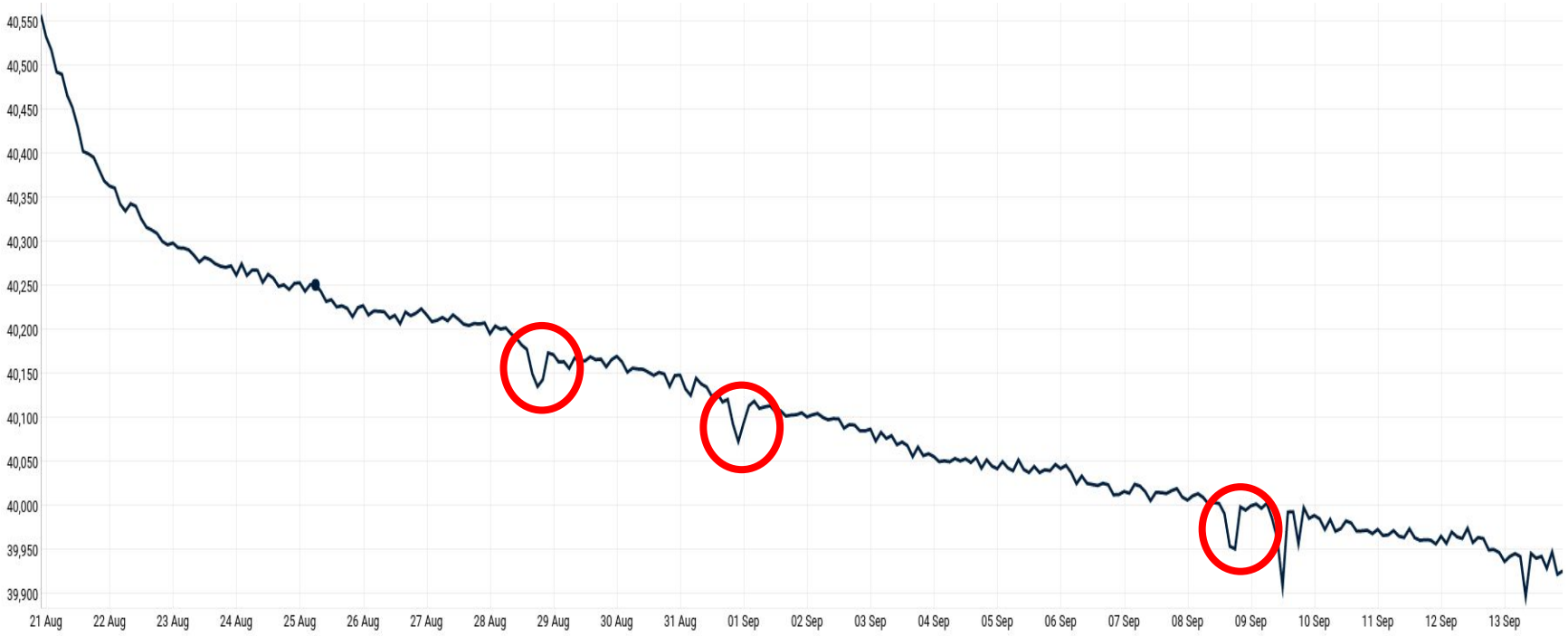


~40 electrons per second

First step: downsampling

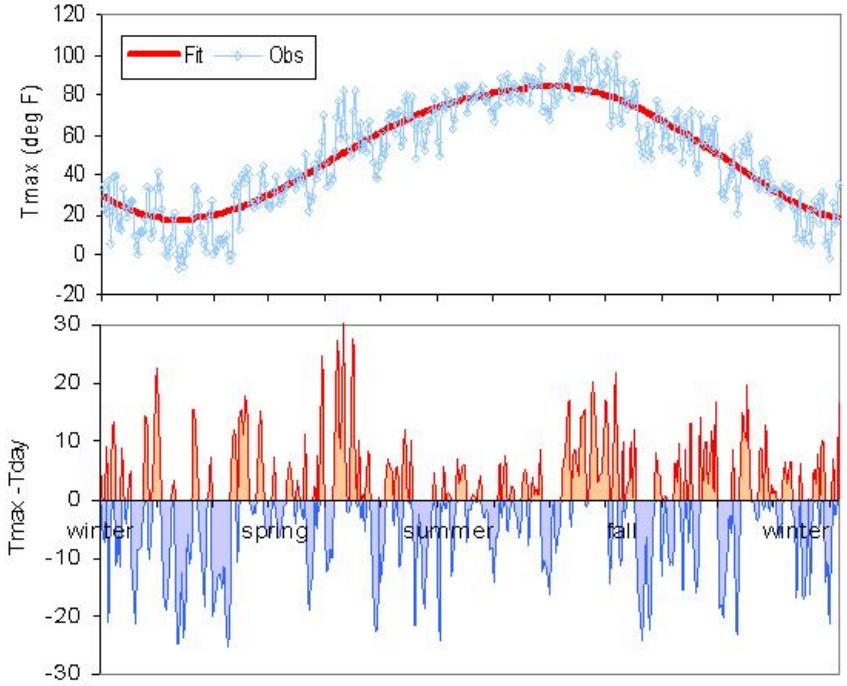


First step: downsampling



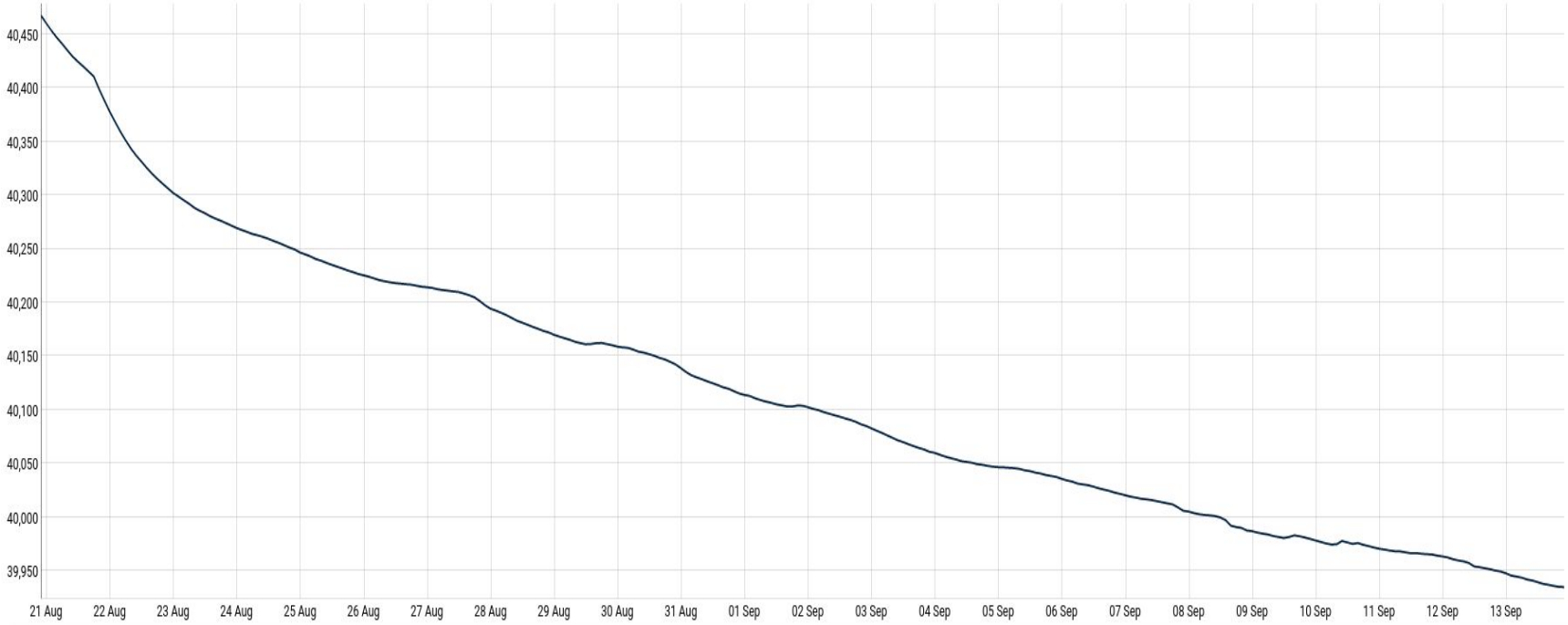
You can see the transit candidates…
but how can we teach the computer to see them?

If you ♥ signal processing



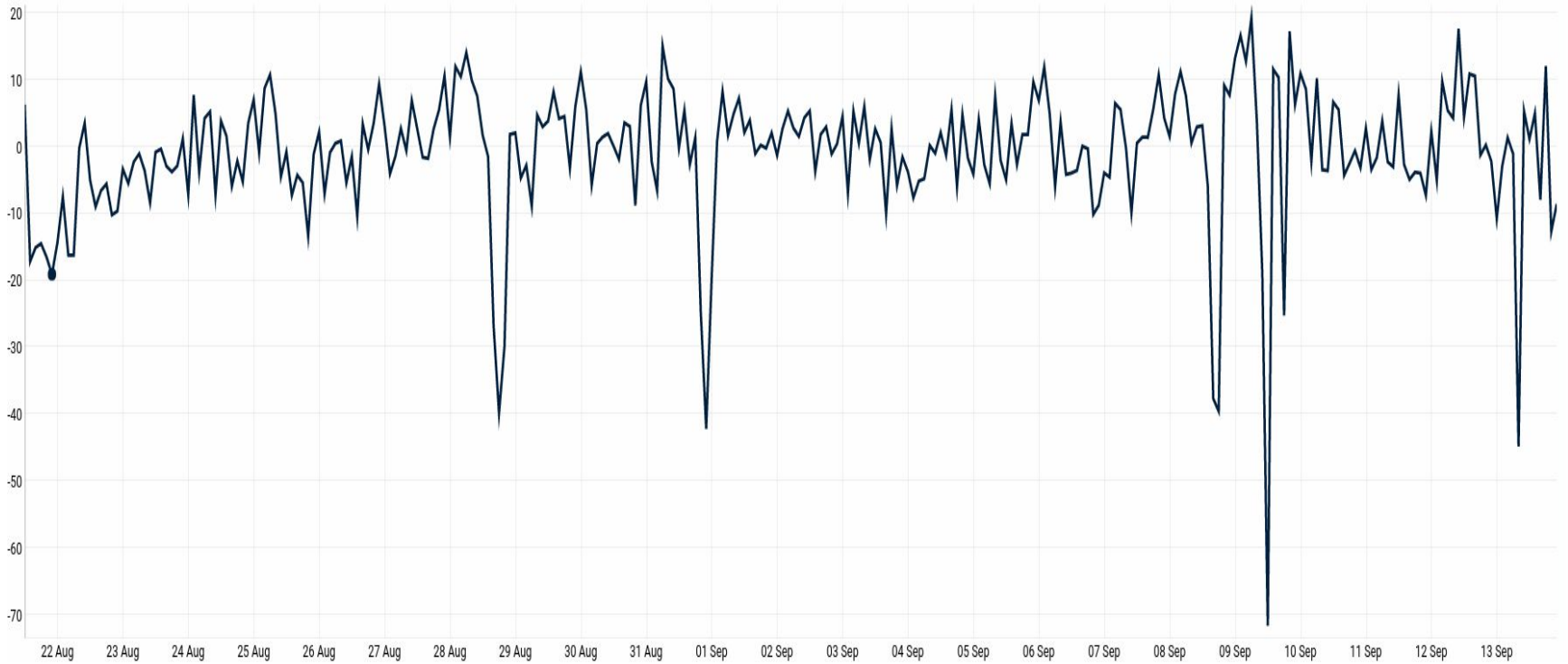
High pass filter

Poor person's high pass filter



Using the trend

Signal - Trend



Now you can see them well

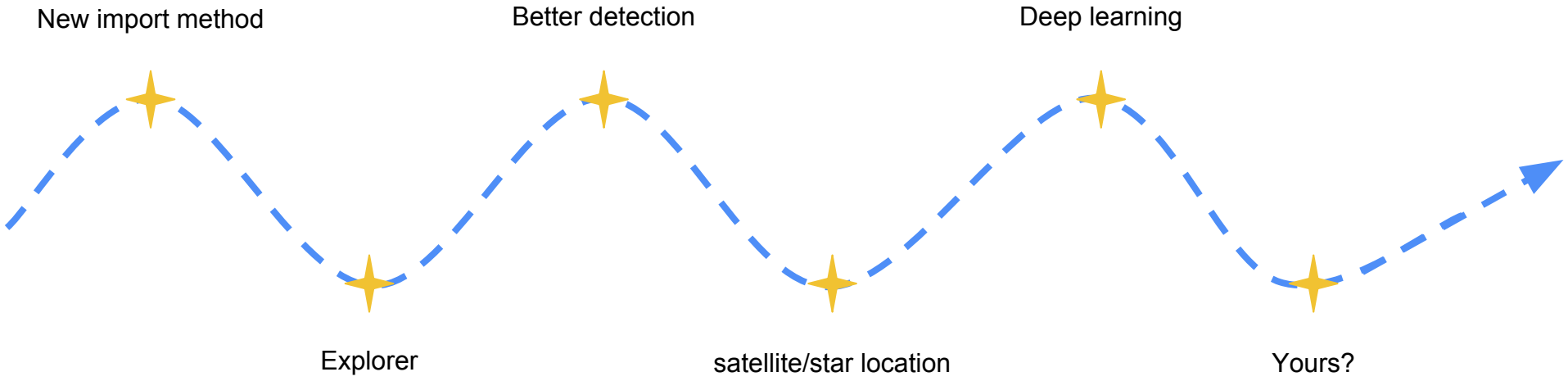
After some tuning

We have our transit candidates

What's next?

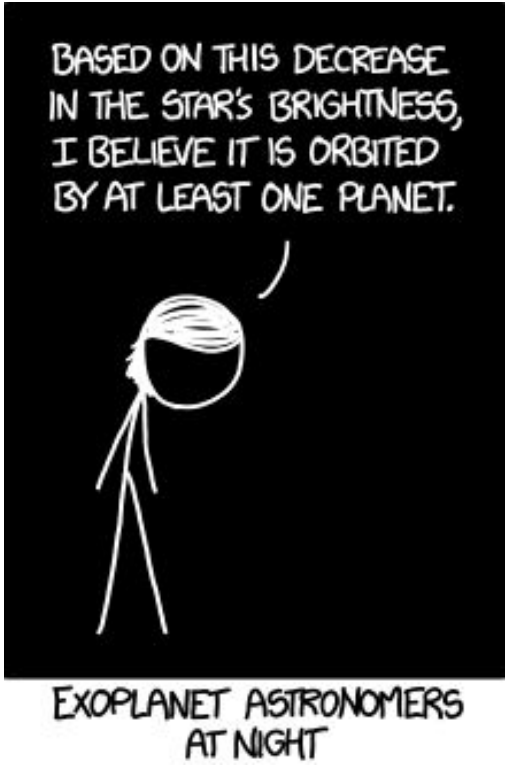
Where do we go from here?

Only the beginning



A growing team

And you!



https://xkcd.com/1371/

Join us!

https://helloexo.world

Thank you!

web2day
DIGITAL SPRINGBREAK

Want to know more?

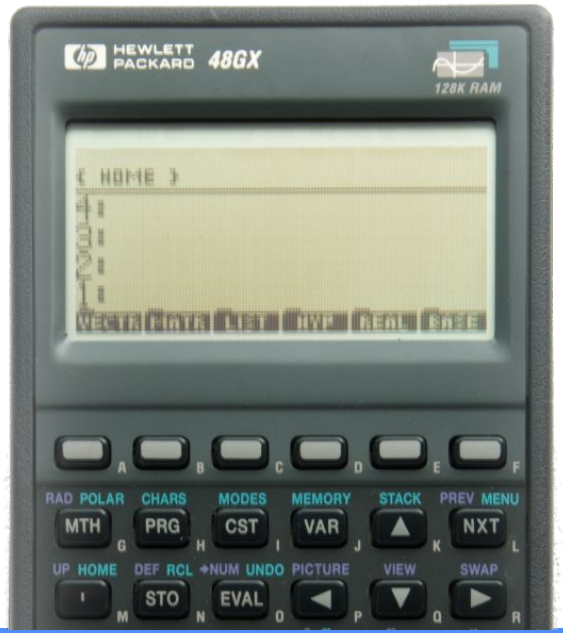
Analysing with WarpScript

WarpScript

web2day
DIGITAL SPRINGBREAK

Reverse Polish Notation

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



Variables

```
'hello, world!'          // Push Hello World String on the Stack
'exo' STORE              // Store it in a variable called exo
$exo                     // Then push back exo variable on the stack
```

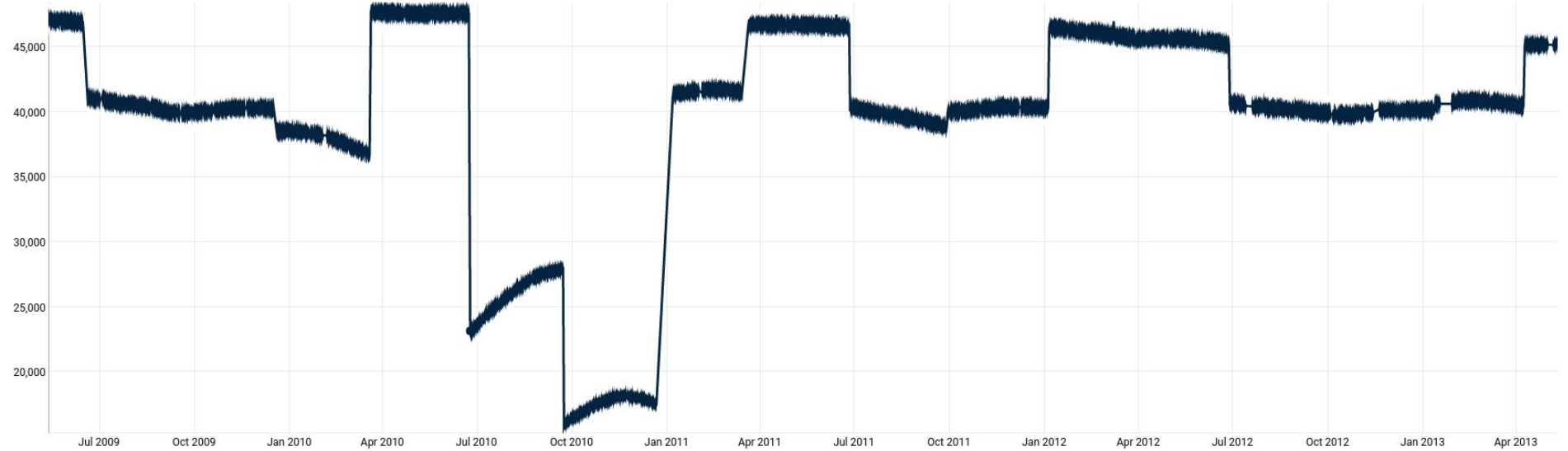

What are the available series?

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

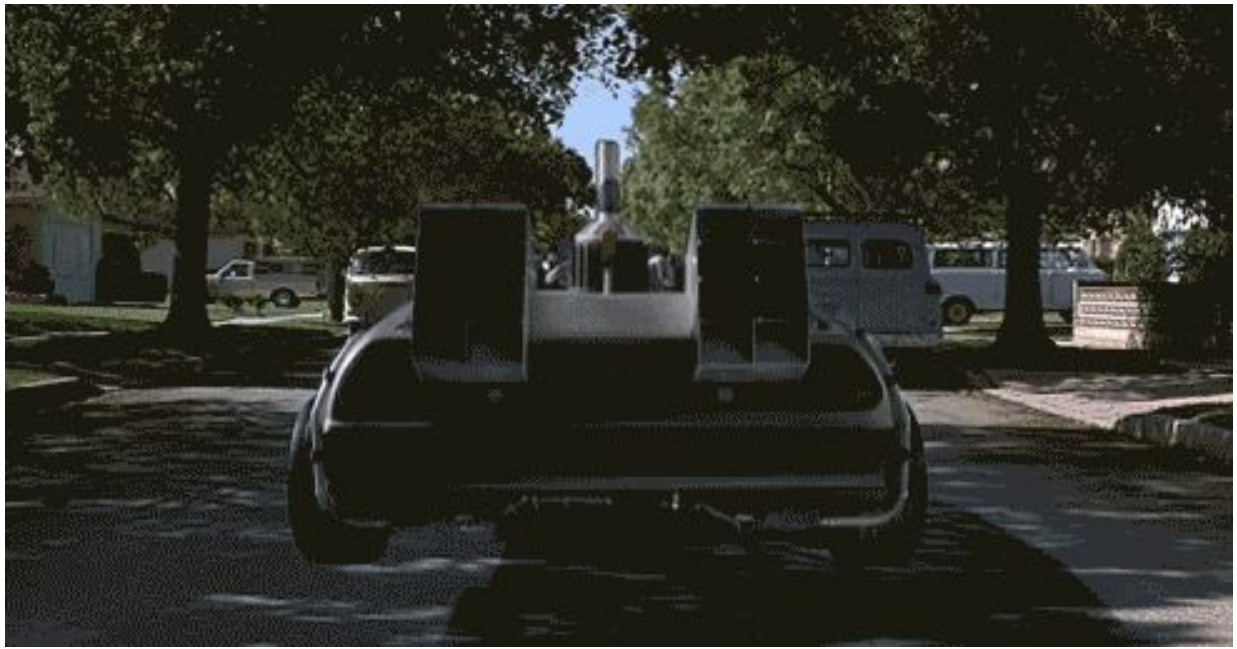

Get raw data

```
[
        $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
```

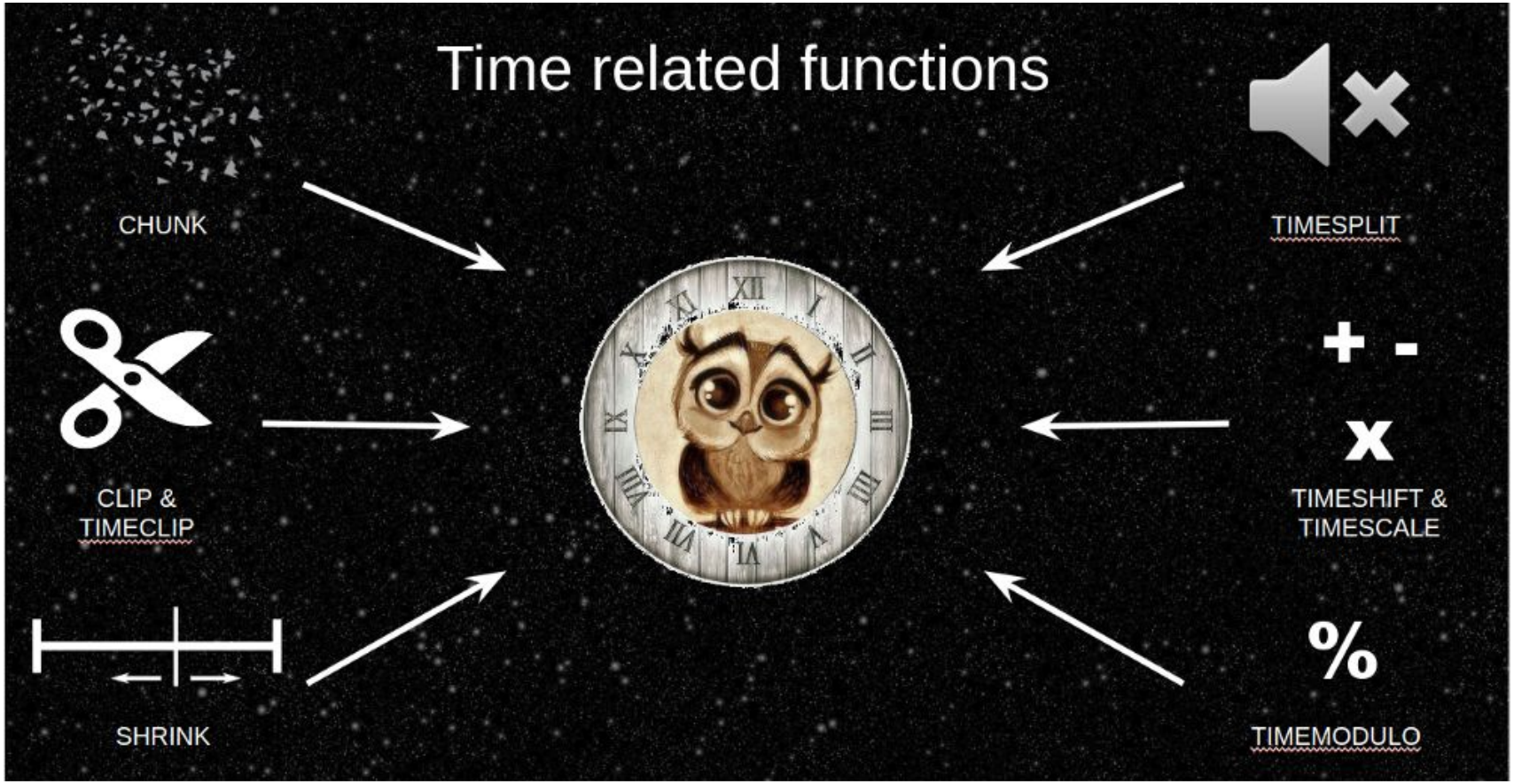

Kepler-11: Raw data



Time manipulation



Time related functions



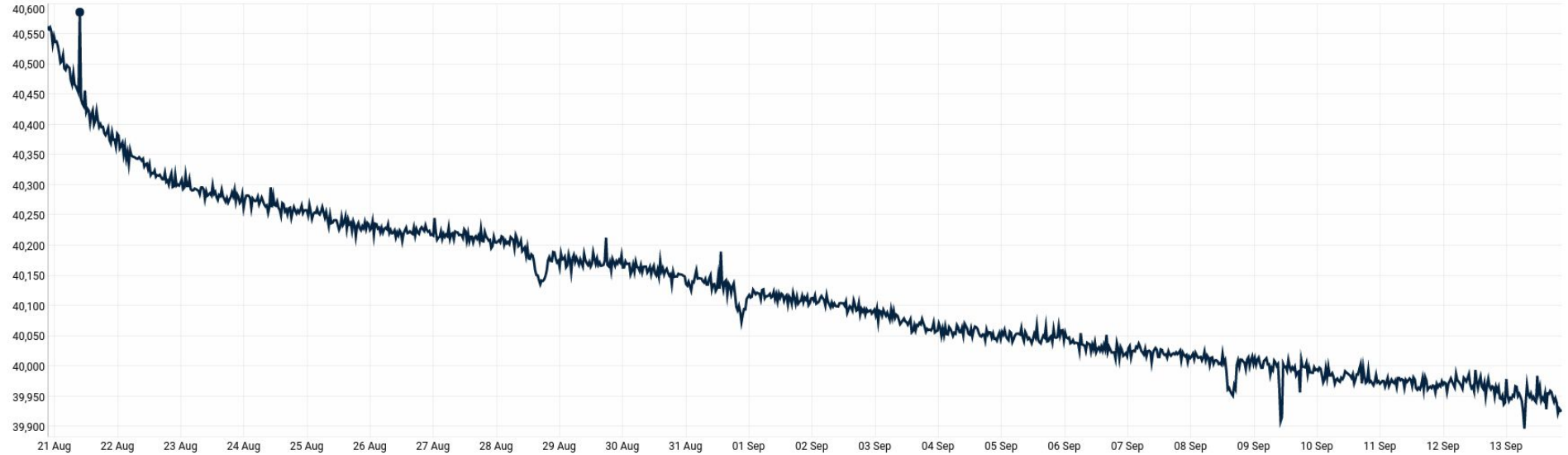
How to split a Time series

```
$gts                    // Singleton (or list of) GTS
6 h                     // Minimum of time without data-points
100                     // Minimum of data-points required
'record'                // New labels to subdivide the result
TIMESPLIT
```

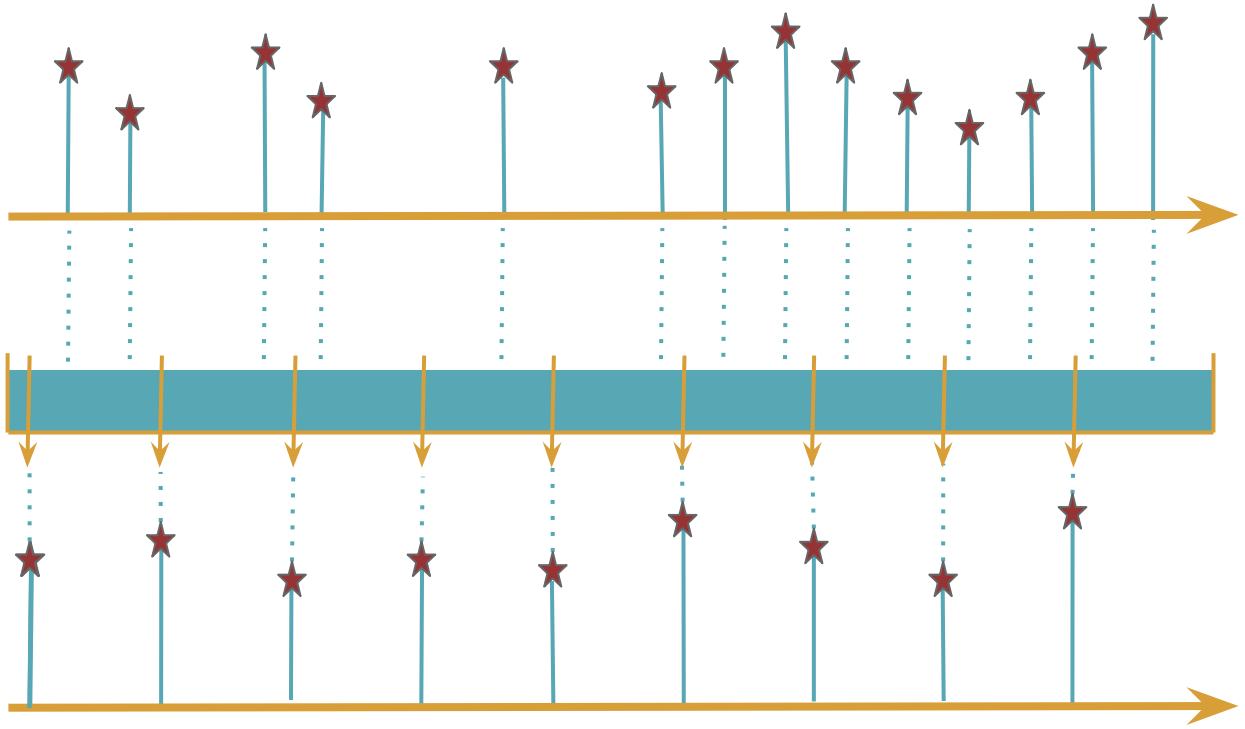

Filtering

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

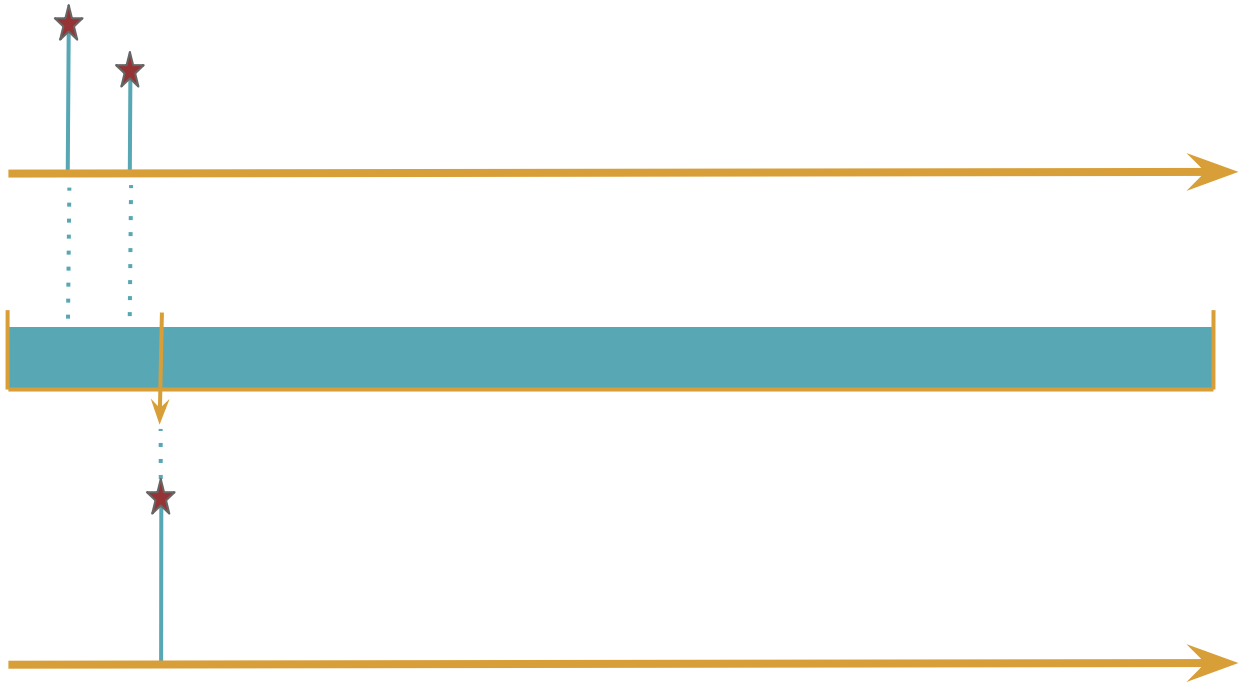

Reference record: 5



Downsampling

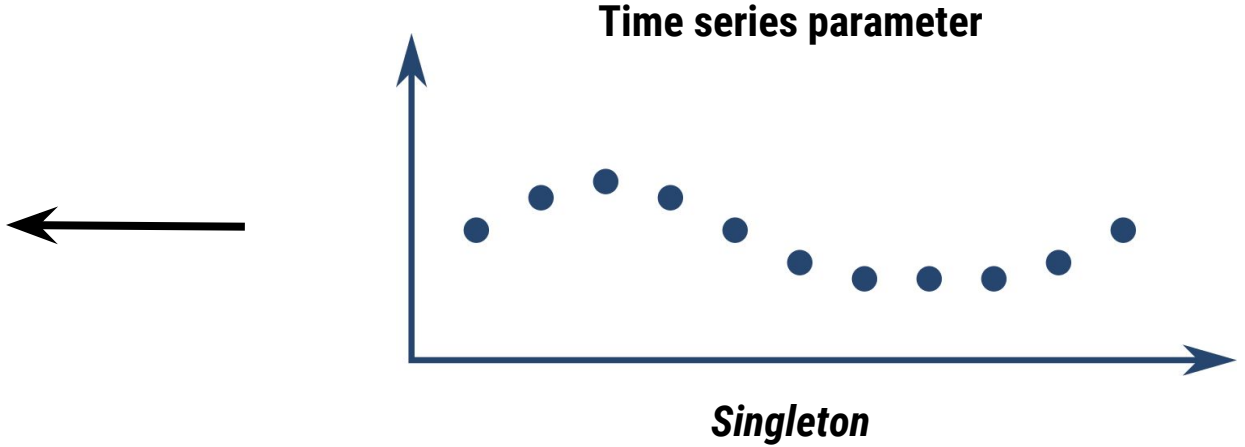


Bucketize



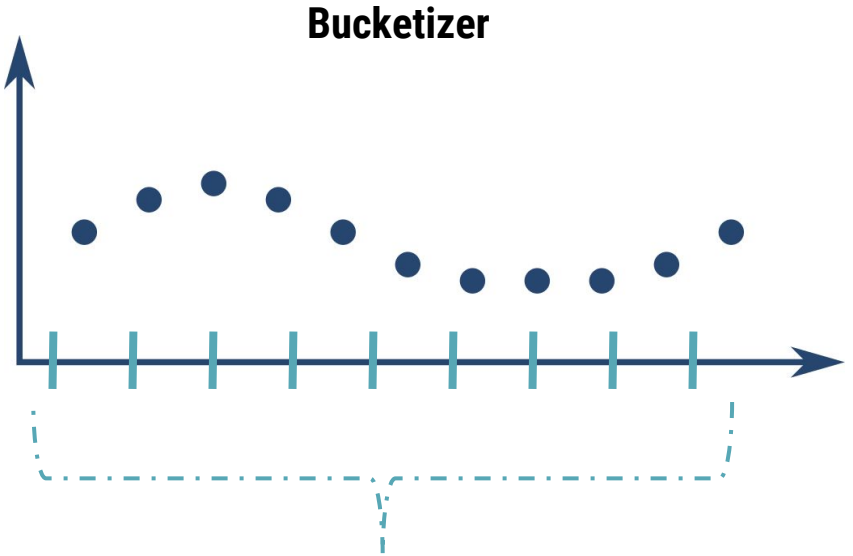
Syntax

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



Syntax

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

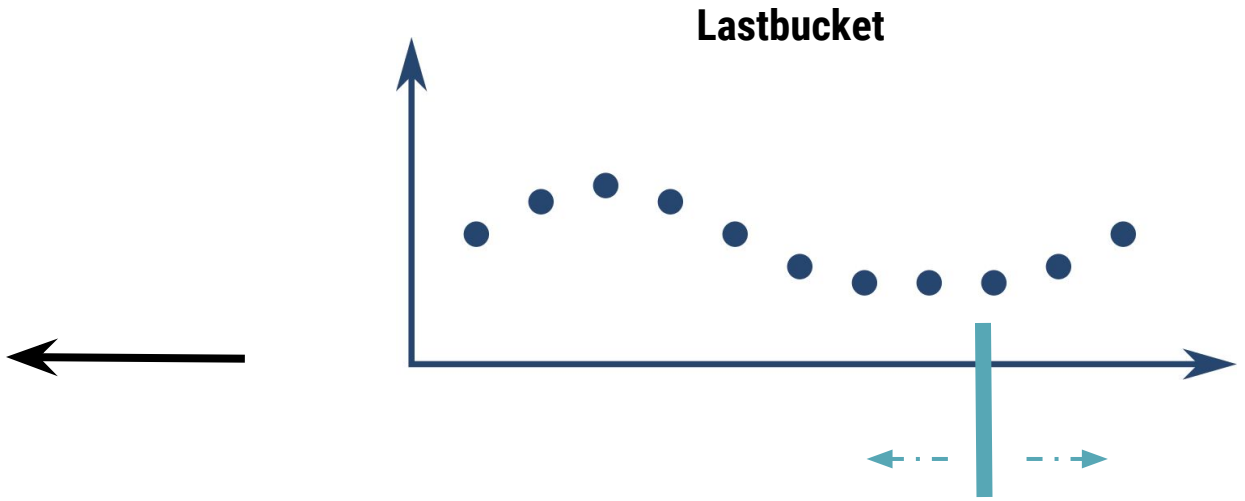


Type of operator to apply on each bucket

last, max, mean, and, count ...

Syntax

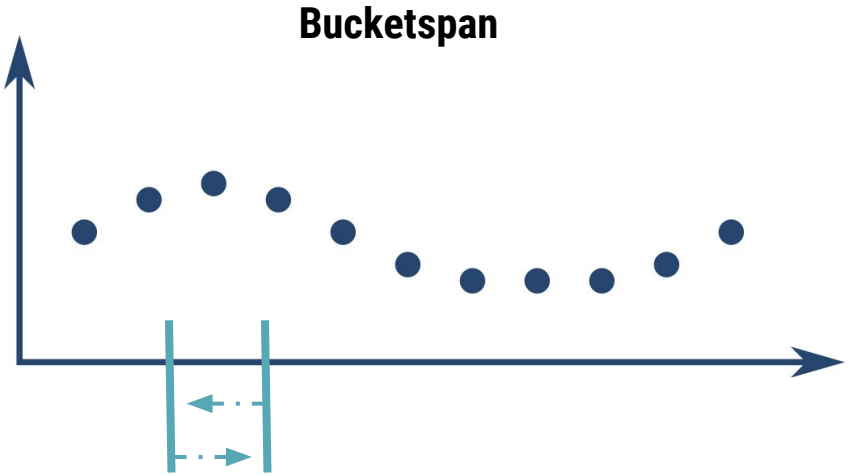
```
[
    $gts
    bucketizer.min
    0
    2 h
    0
]
BUCKETIZE
```



End timestamp of the more recent bucket

Syntax

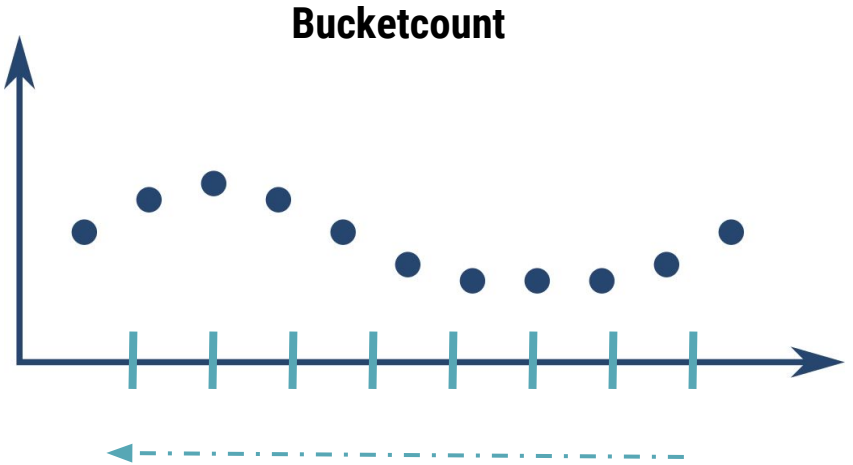
```
[
        $gts
        bucketizer.min
        0
        2 h
        0
]
BUCKETIZE
```



Width of a bucket

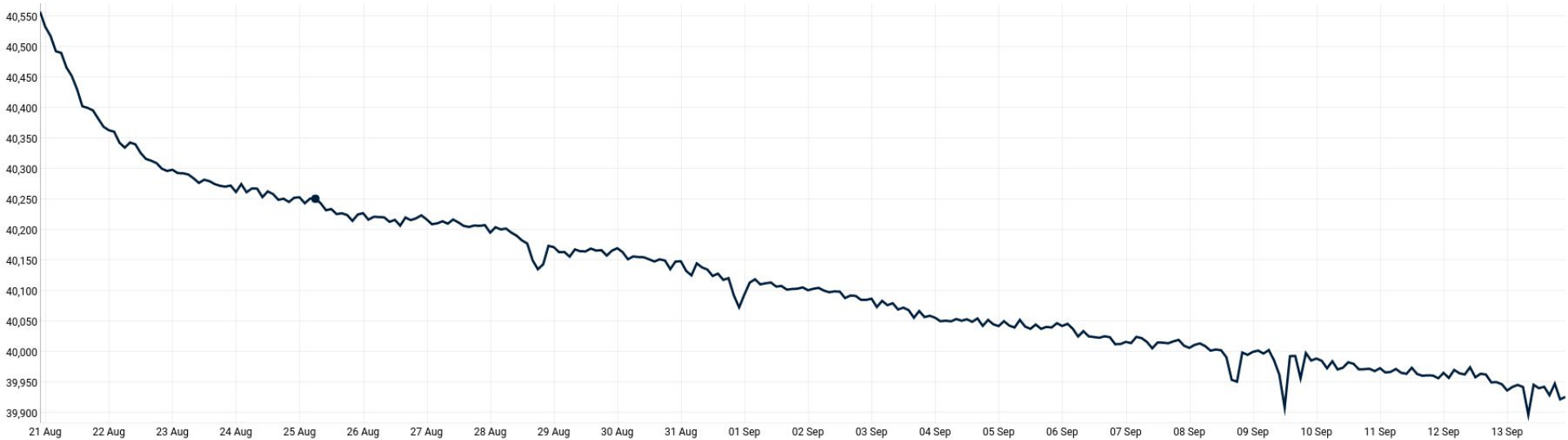
Syntax

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

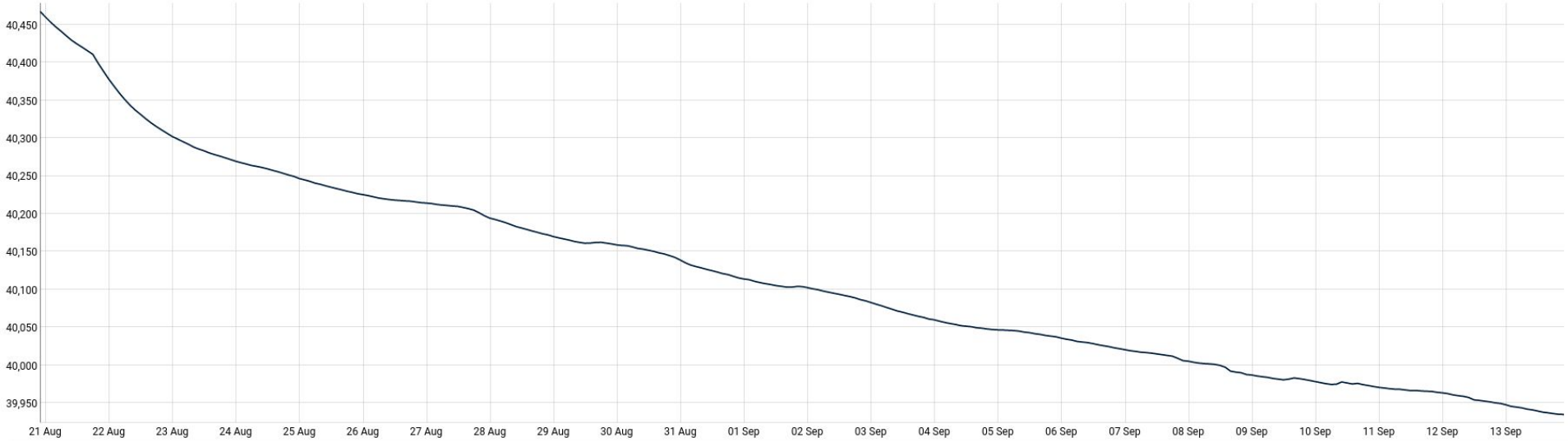


Number of buckets to keep

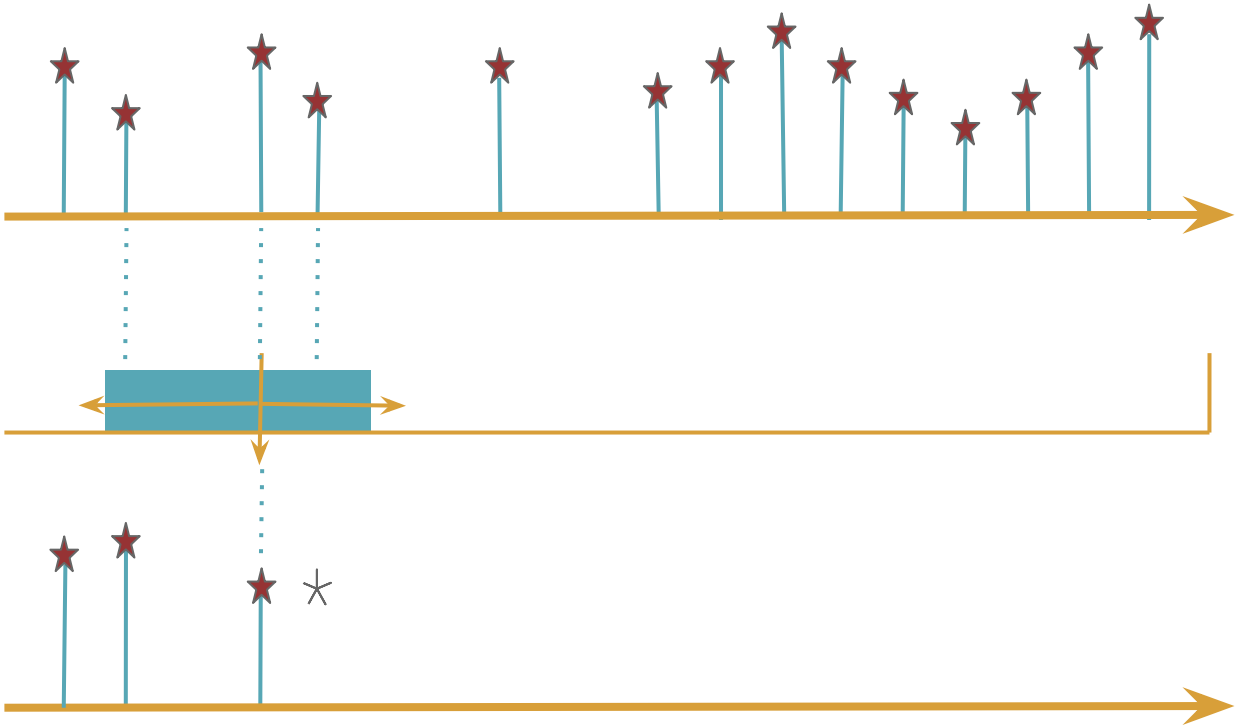
Actual


40,550
40,500
40,450
40,400
40,350
40,300
40,250
40,200
40,150
40,100
40,050
40,000
39,950
39,900
21 Aug
22 Aug
23 Aug
24 Aug
25 Aug
26 Aug
27 Aug
28 Aug
29 Aug
30 Aug
31 Aug
01 Sep
02 Sep
03 Sep
04 Sep
05 Sep
06 Sep
07 Sep
08 Sep
09 Sep
10 Sep
11 Sep
12 Sep
13 Sep

Trend

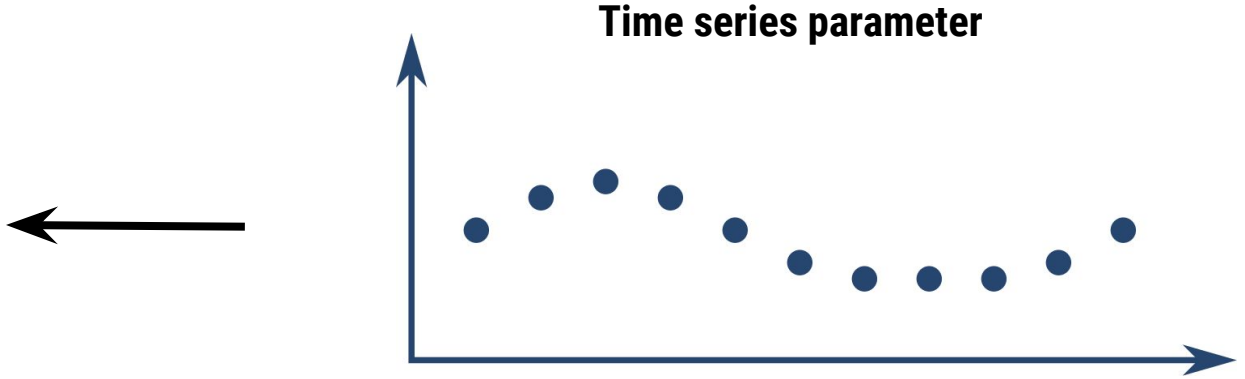

40,450
40,400
40,350
40,300
40,250
40,200
40,150
40,100
40,050
40,000
39,950
21 Aug
22 Aug
23 Aug
24 Aug
25 Aug
26 Aug
27 Aug
28 Aug
29 Aug
30 Aug
31 Aug
01 Sep
02 Sep
03 Sep
04 Sep
05 Sep
06 Sep
07 Sep
08 Sep
09 Sep
10 Sep
11 Sep
12 Sep
13 Sep

Mapper



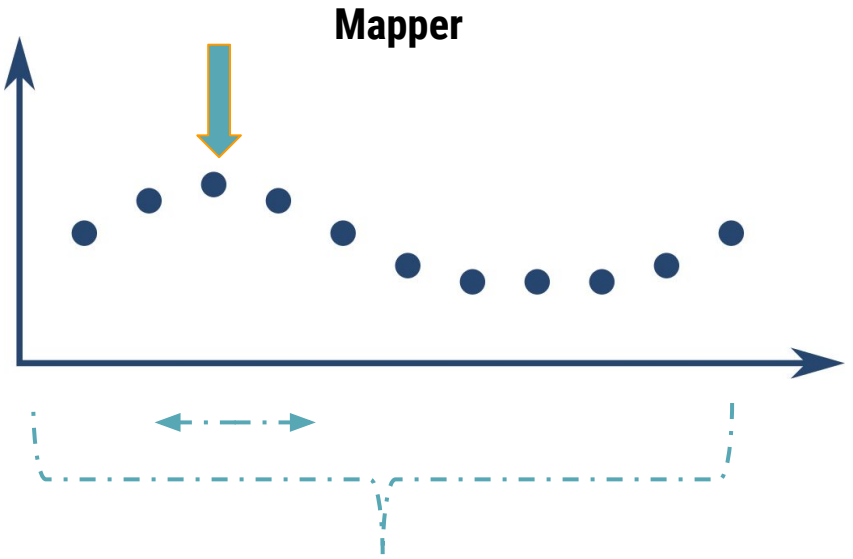
Syntax

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



Syntax

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

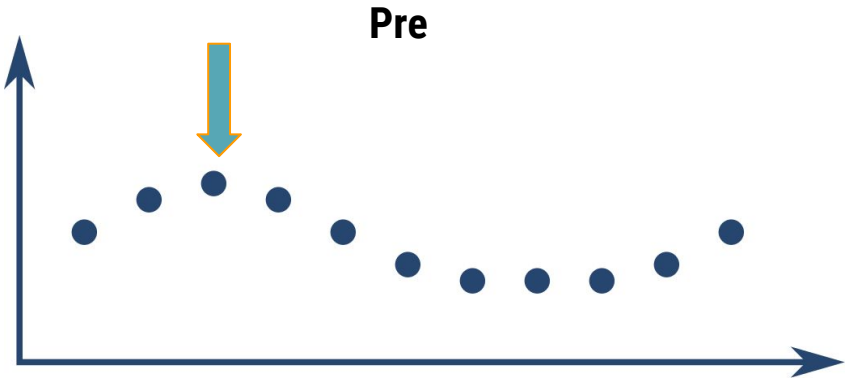


Type of operator to apply on each window

add, gt, rate, and, count...

Syntax

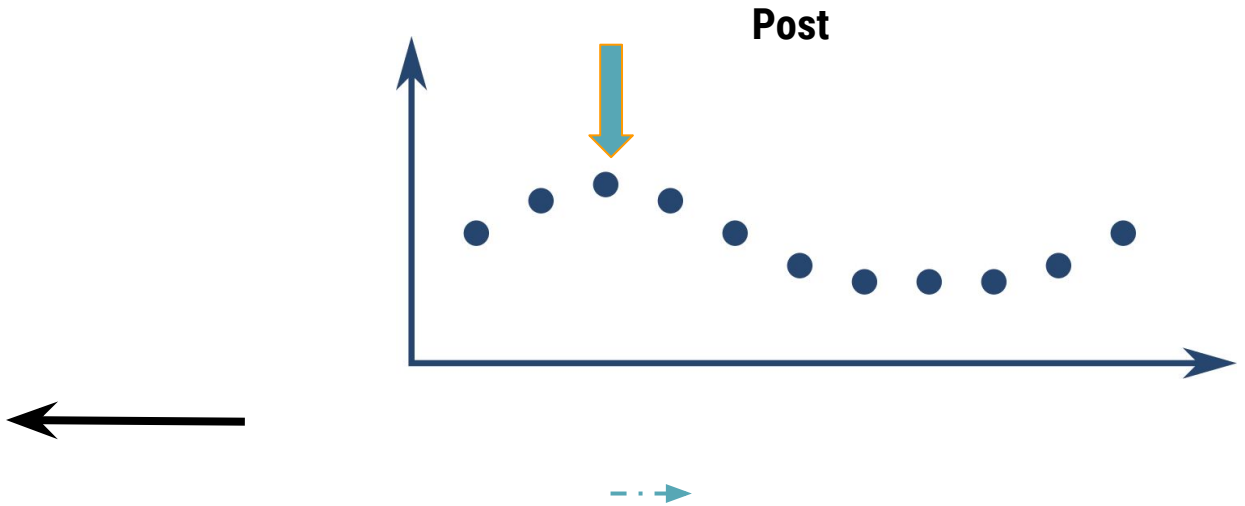
```
[
        $gts
        mapper.mean
        2
        2
        0
]
MAP
```



Number of data-points before

Syntax

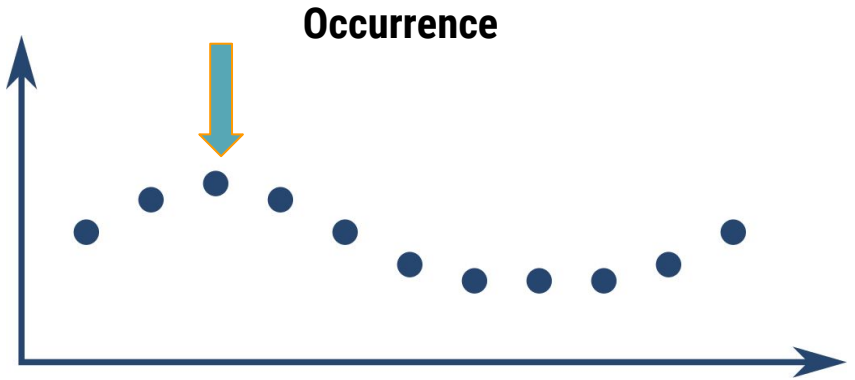
```
[
        $gts
        mapper.mean
        2
        2
        0
]
MAP
```



Number of data-points after

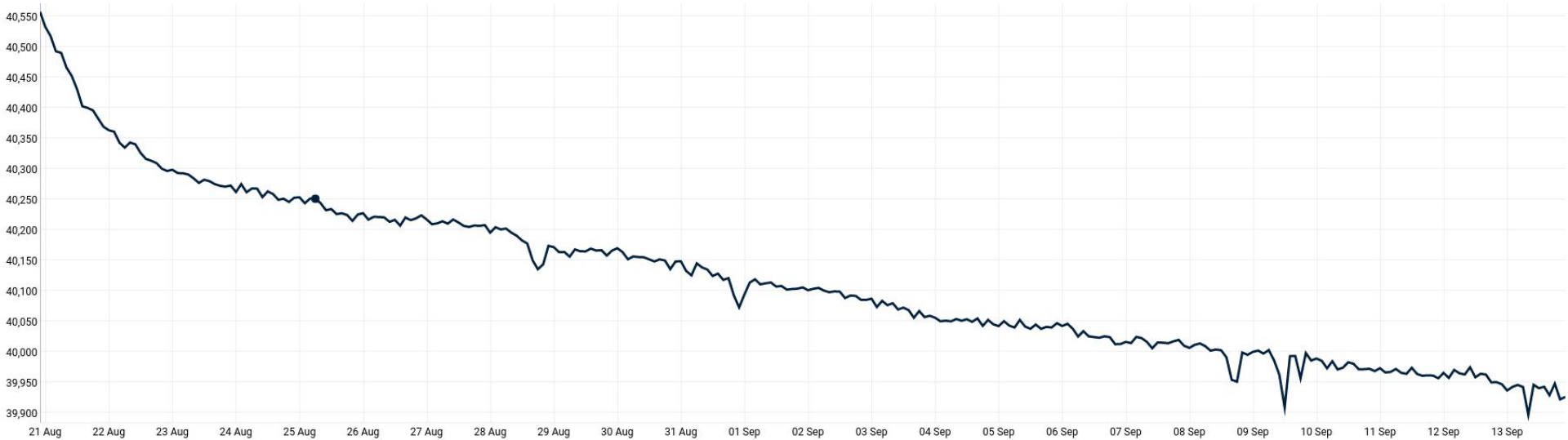
Syntax

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

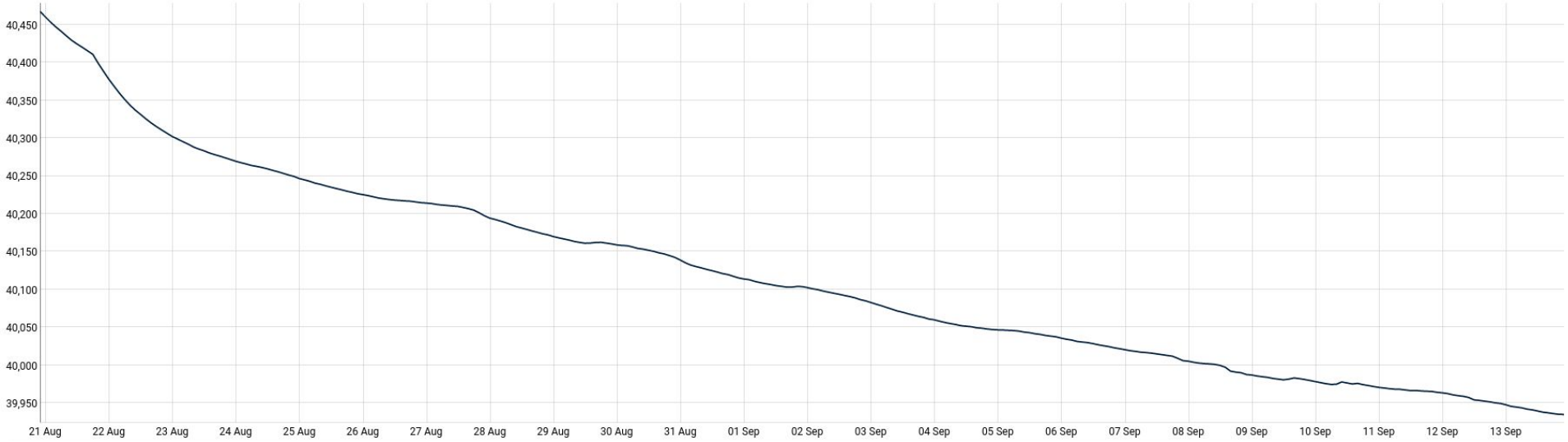


Maximal number of calculation for a data-point

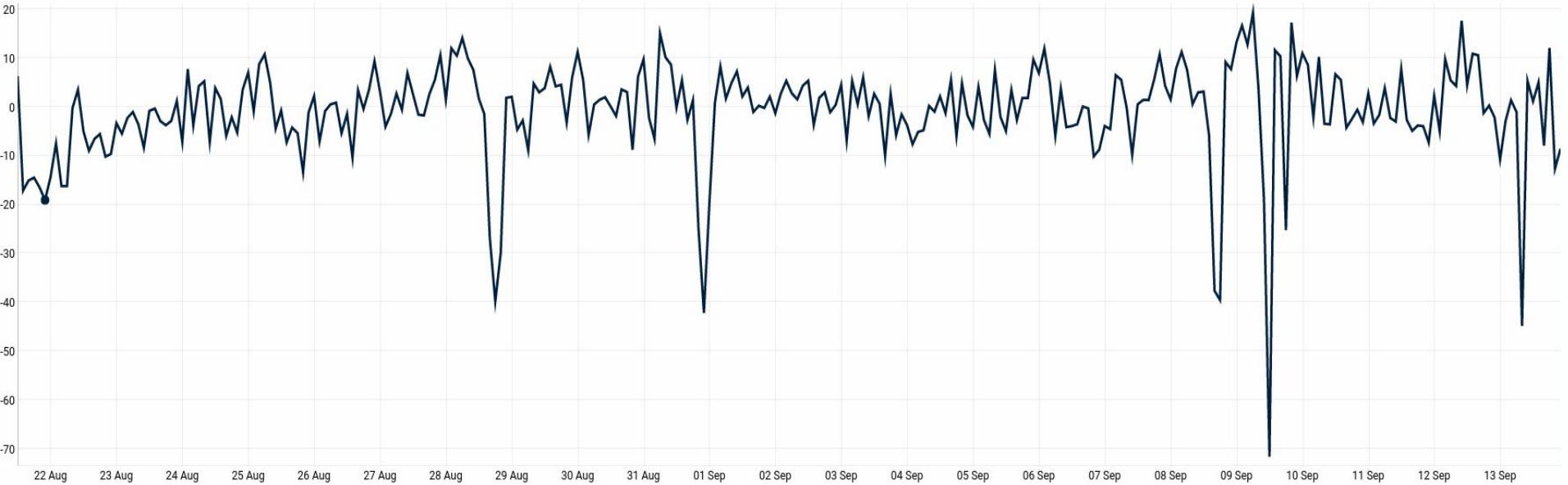
Actual


40,550
40,500
40,450
40,400
40,350
40,300
40,250
40,200
40,150
40,100
40,050
40,000
39,950
39,900
21 Aug
22 Aug
23 Aug
24 Aug
25 Aug
26 Aug
27 Aug
28 Aug
29 Aug
30 Aug
31 Aug
01 Sep
02 Sep
03 Sep
04 Sep
05 Sep
06 Sep
07 Sep
08 Sep
09 Sep
10 Sep
11 Sep
12 Sep
13 Sep

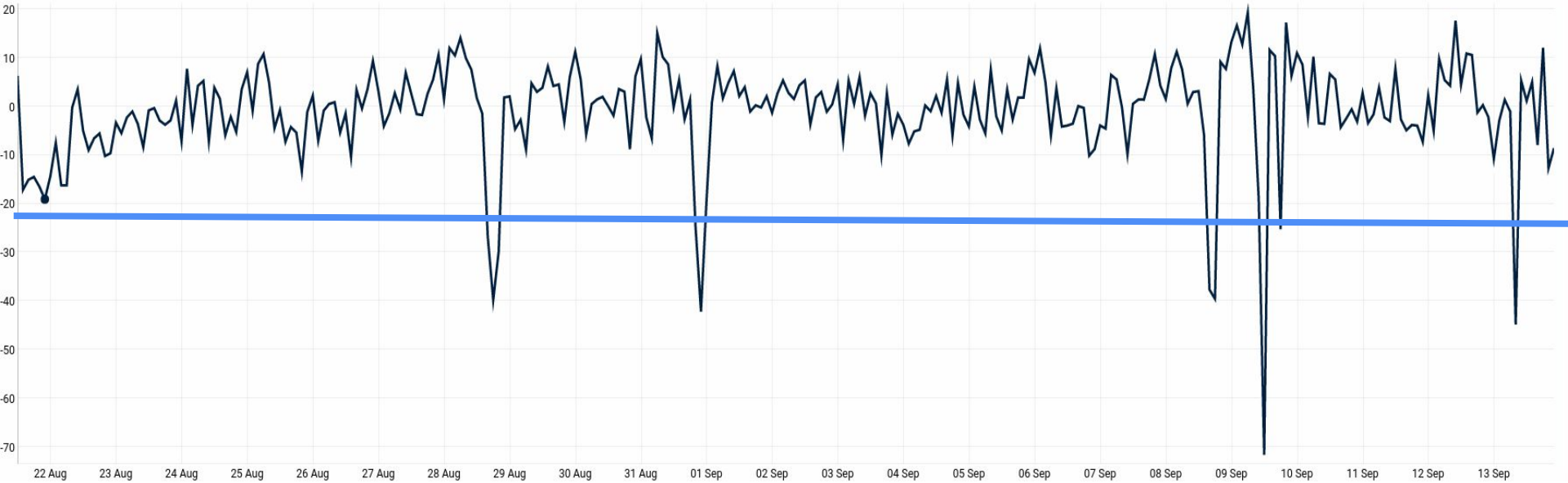
Trend


40,450
40,400
40,350
40,300
40,250
40,200
40,150
40,100
40,050
40,000
39,950
21 Aug
22 Aug
23 Aug
24 Aug
25 Aug
26 Aug
27 Aug
28 Aug
29 Aug
30 Aug
31 Aug
01 Sep
02 Sep
03 Sep
04 Sep
05 Sep
06 Sep
07 Sep
08 Sep
09 Sep
10 Sep
11 Sep
12 Sep
13 Sep

Actual - trend



Actual - trend



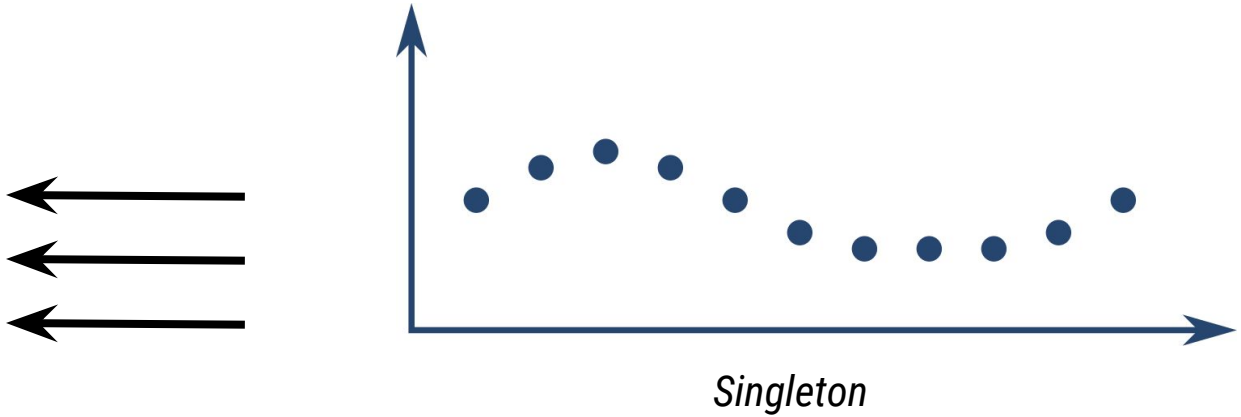
Time to level-up!

Time series operation

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


Syntax

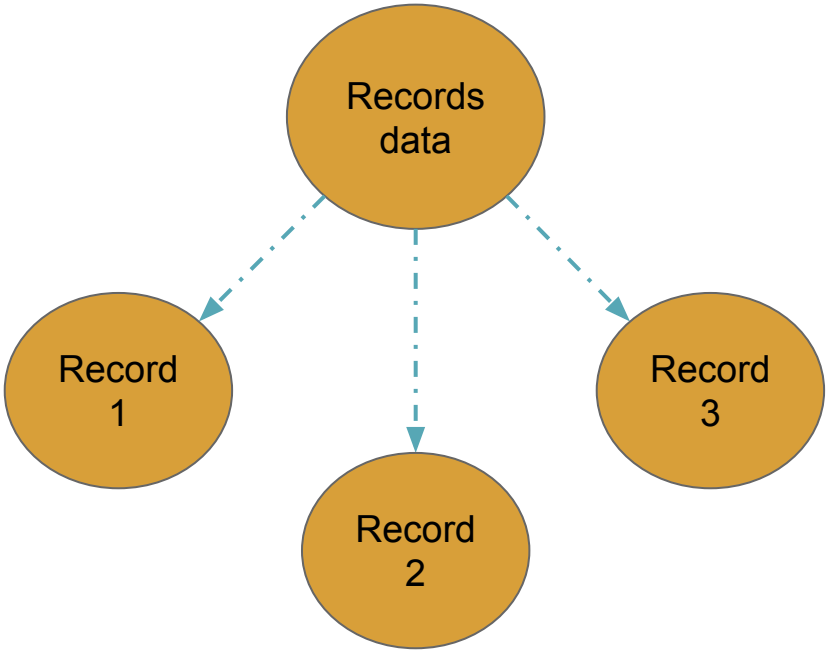
```
[
        $gts0
        …
        $gtsN
        [ ‘record’ ]
        op.add
]
APPLY
```



Syntax

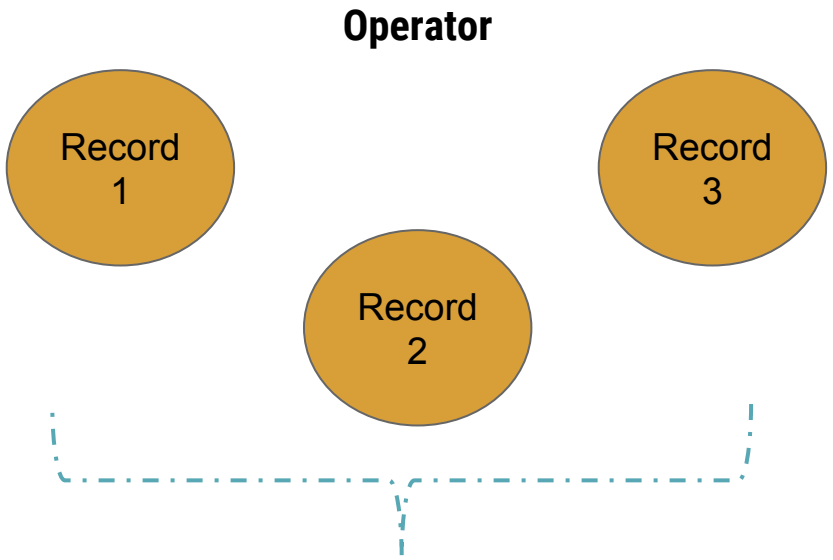
```
[
        $gts0
        …
        $gtsN
        [ ‘record’ ]
        op.add
]
APPLY
```

Equivalence class



Syntax

```
[
        $gts0
        …
        $gtsN
        [ ‘record’ ]
        op.add
]
APPLY
```



Type of operator to apply on each class

sub, gt, mask, and, mul ...

Final result