OVH
Innovation is Freedom

Code d'Armor

Rediscover the known Universe with NASA datasets

Horacio Gonzalez

@LostInBrittany

Horacio Gonzalez

Code d'Armor

@LostInBrittany

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

Breizh C@mp
Mix de technologies

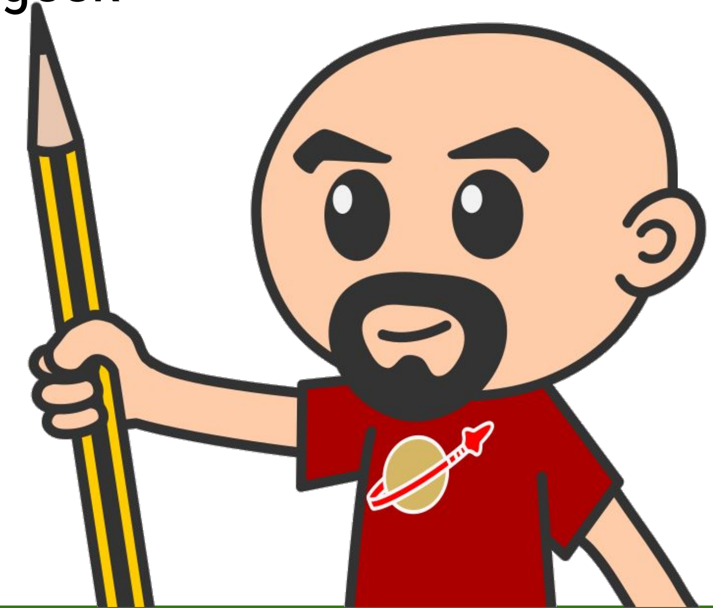
la cantine BREST

Finist Devs

Warp 10 Meetup

Google Developers Experts 2015
Web Technologies GDE

OVH
Team DevRel



HelloExoWorld

Looking for exoplanets in NASA datasets

Code d'Armor

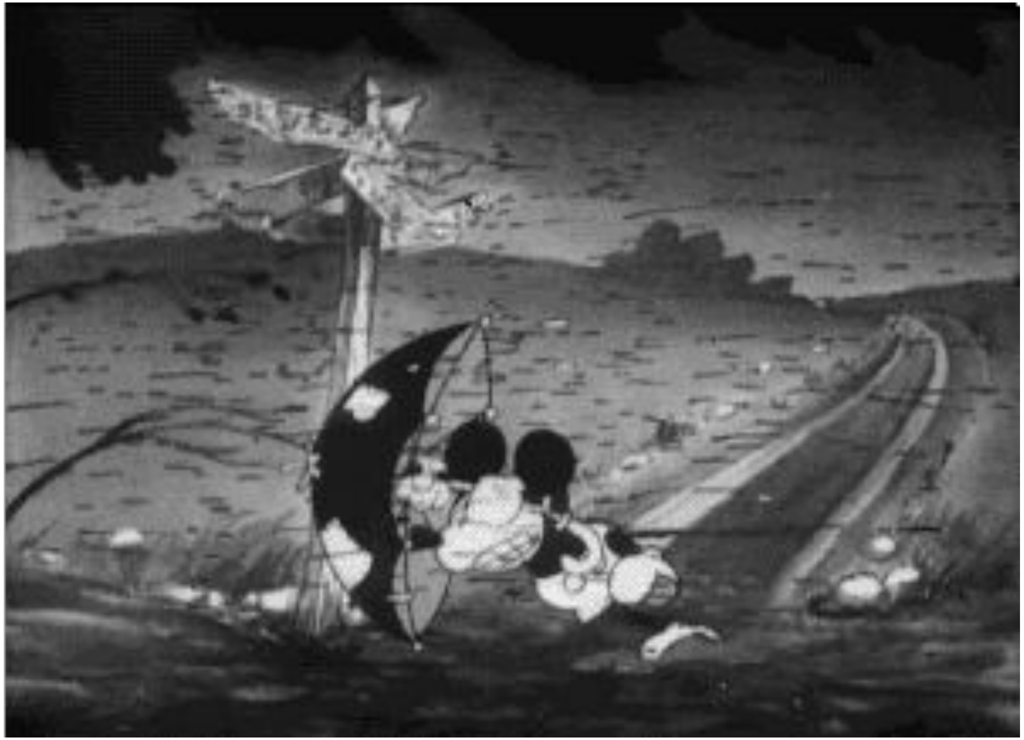
HelloExoWorld

Once upon a time...

An amateur astronomer

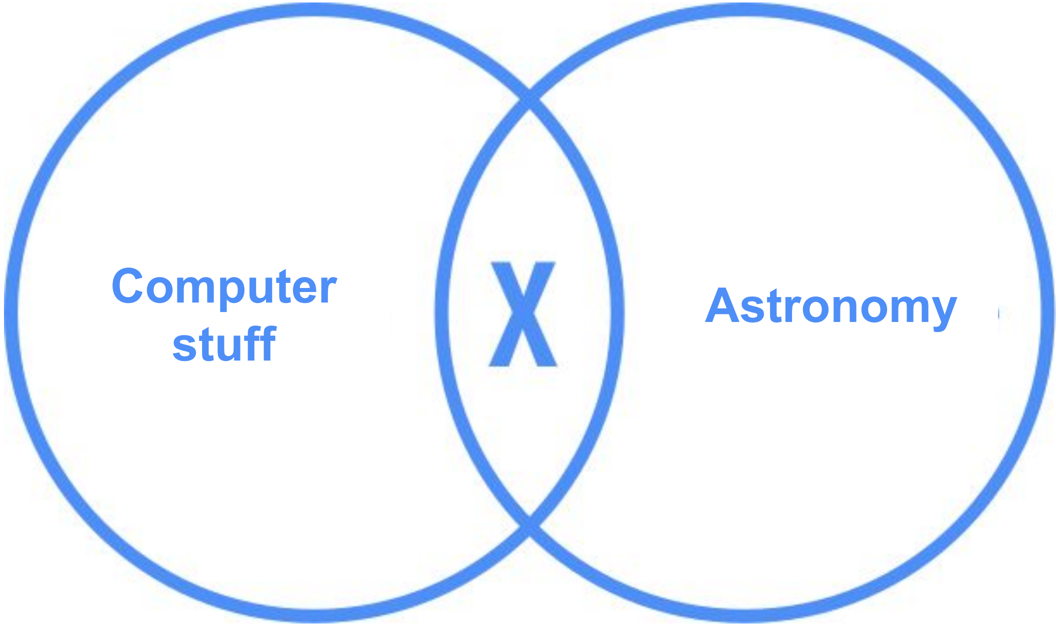
Pierre Zemb, DevOps OVH

What not to do if you love astronomy



Live in Brest

Looking for solutions



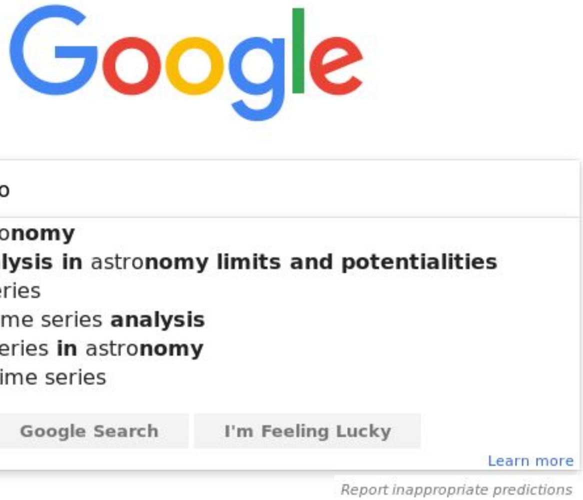
Computer stuff

X

Astronomy

Mixing passions

Google is your friend...

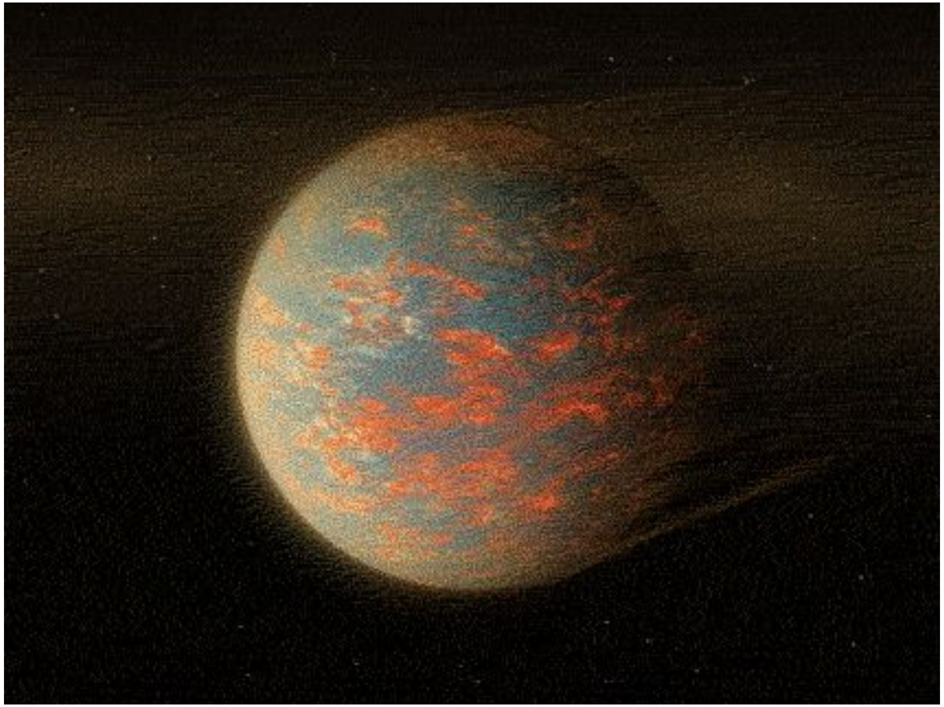


time series astro

time series astro**nomy**
time series **analysis in** astro**nomy limits and potentialities**
astro**ml.**time series
astro**nomical** time series **analysis**
random time series **in** astro**nomy**
astro**physical** time series

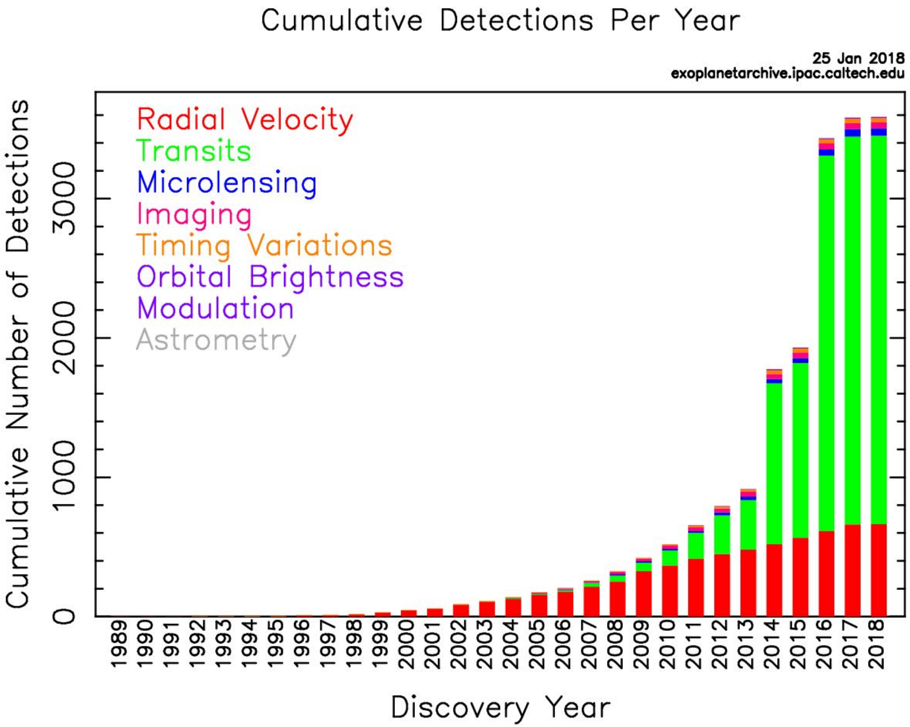
Let's find a project

Exoplanets?



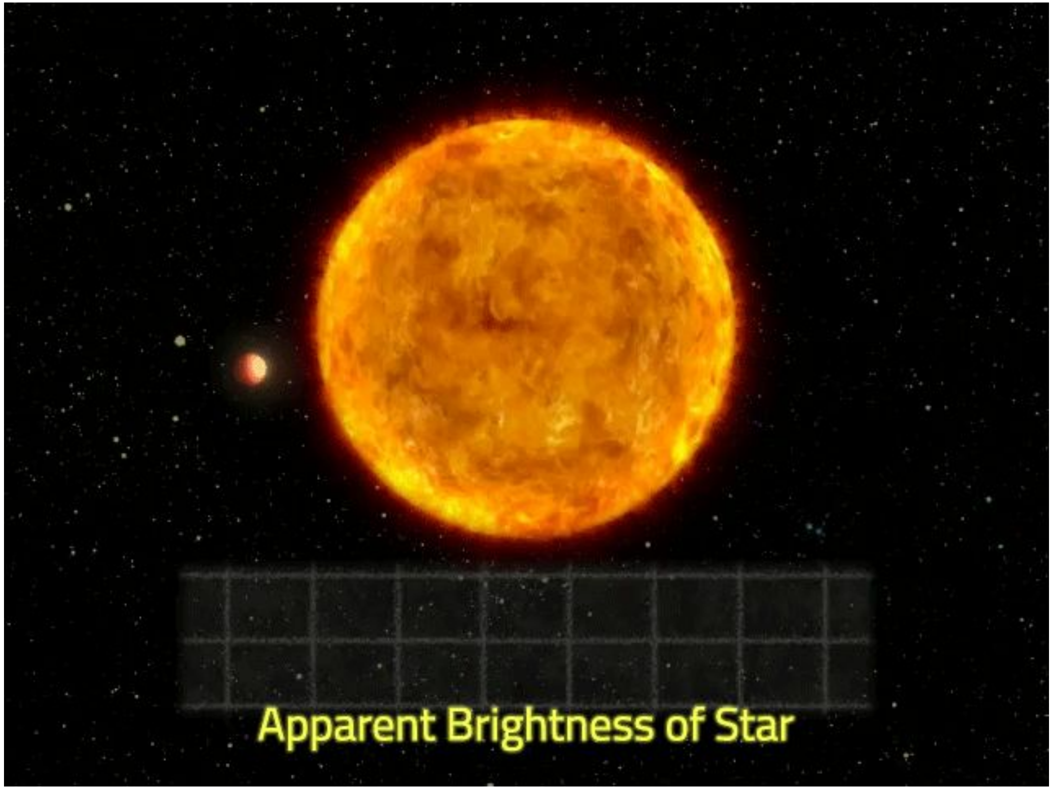
Planets orbiting stars far away

How do we find them?



The transit method seems the best

The transit method



How do we look for transits?

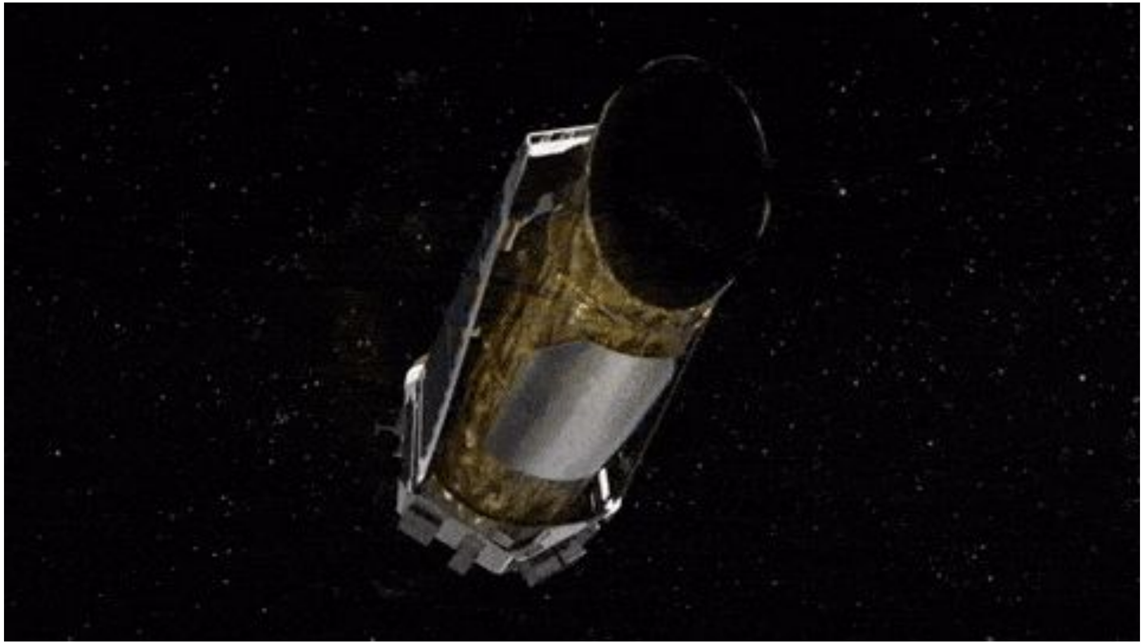
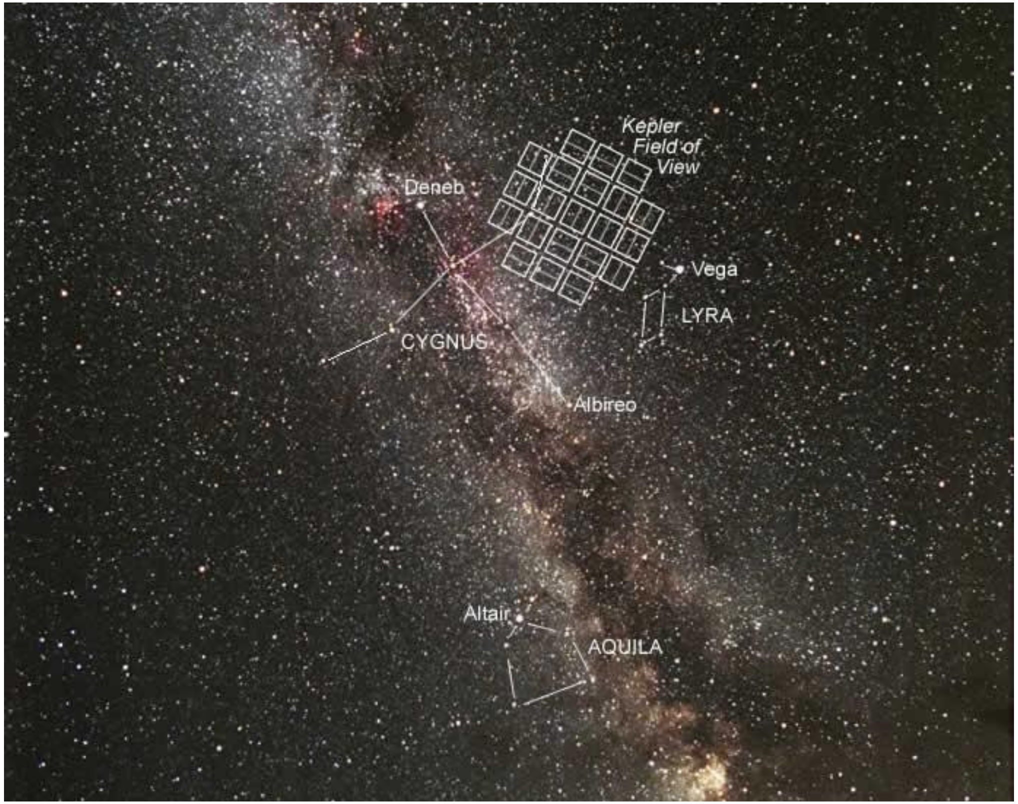


Image credits : NASA

Kepler

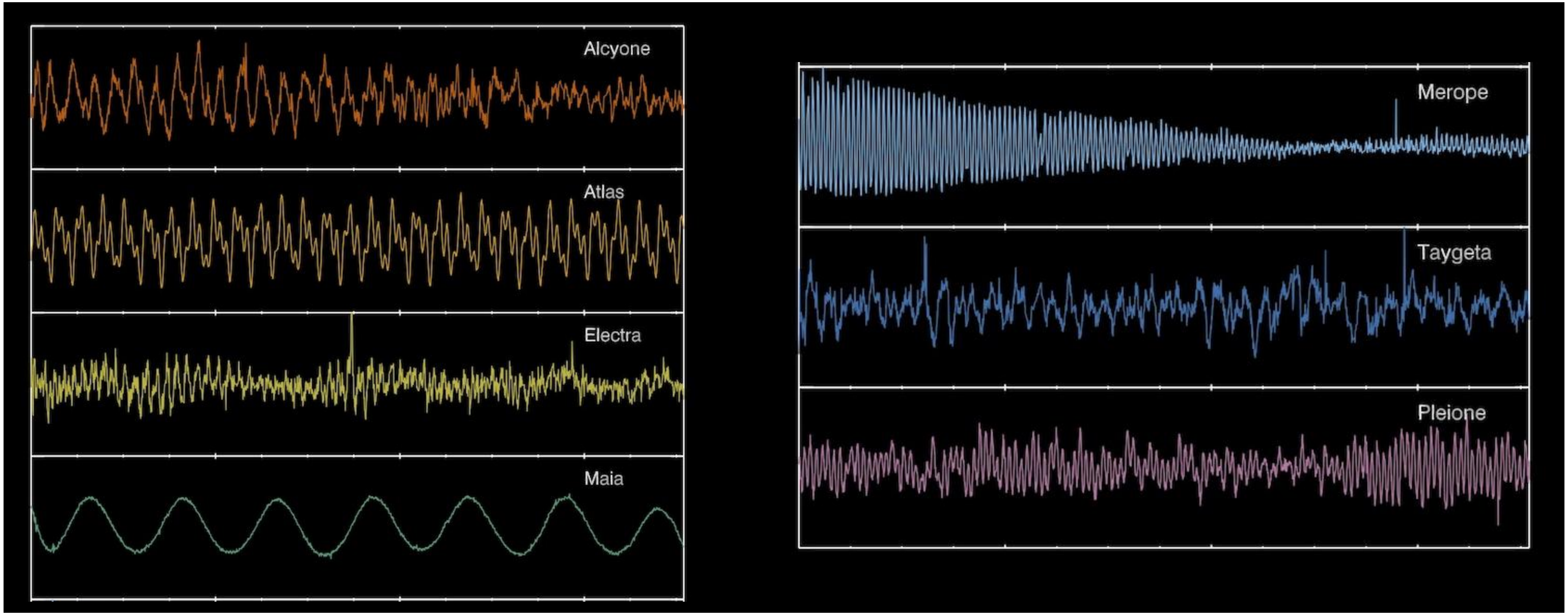
Watching the sky



And what kind of data we get?

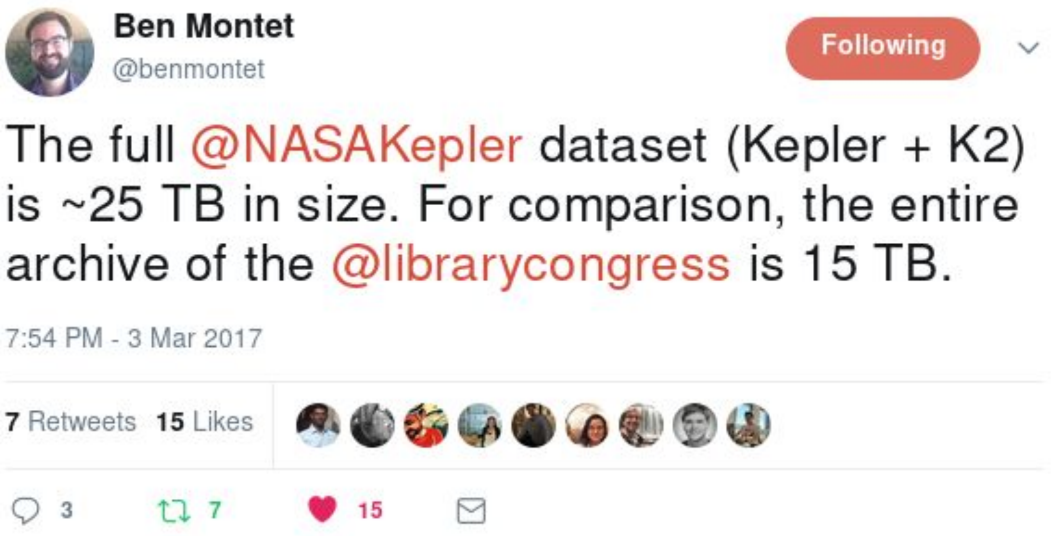
Pleia … mmon

Well, that's the problem



Seven stars, seven different profiles

Kinda big data



Ben Montet
@benmontet

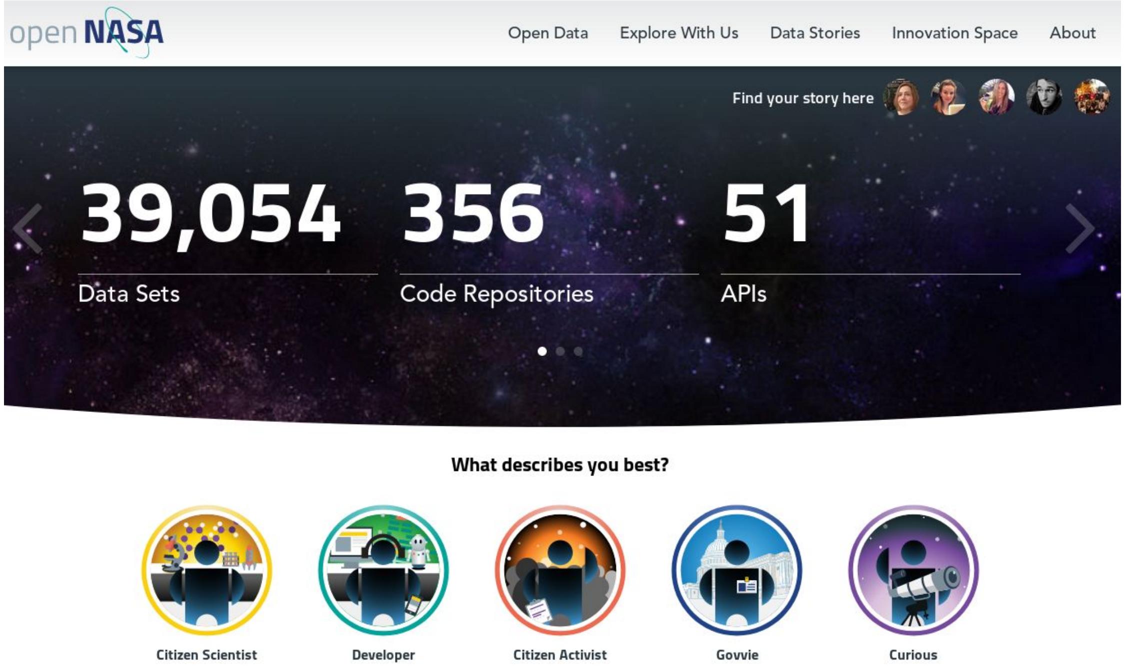
The full @NASAKepler dataset (Kepler + K2) is ~25 TB in size. For comparison, the entire archive of the @librarycongress is 15 TB.

7:54 PM - 3 Mar 2017

7 Retweets 15 Likes

Over 40 million light curves

Big AND open data



open NASA

39,054 Data Sets

356 Code Repositories

51 APIs

What describes you best?

Lots of datasets in #opendata

And we can help with that!

Let's use our tools to analyse the data

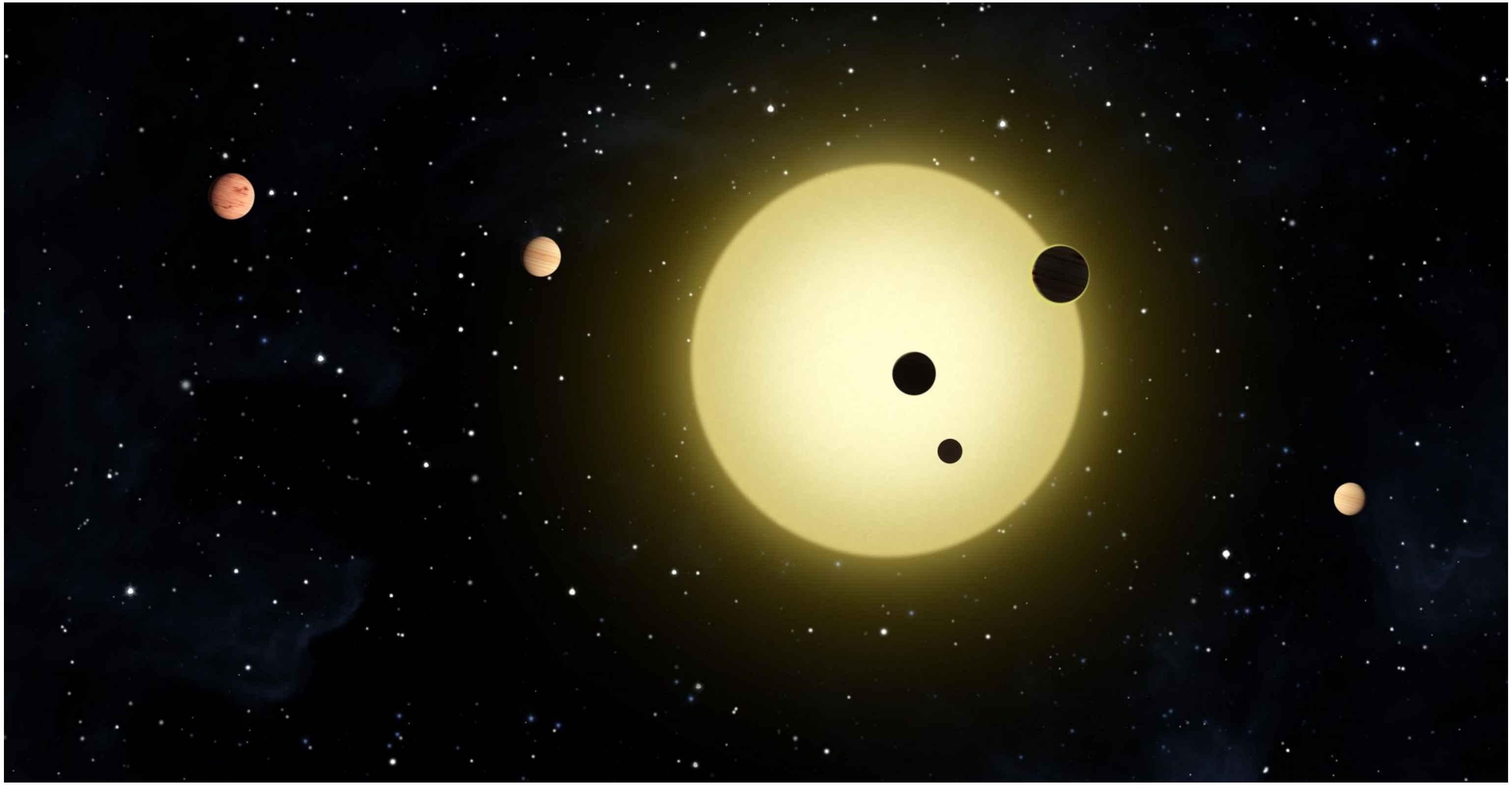
A match made in heaven

Warp 10, OVH Metrics and HelloExoWorld

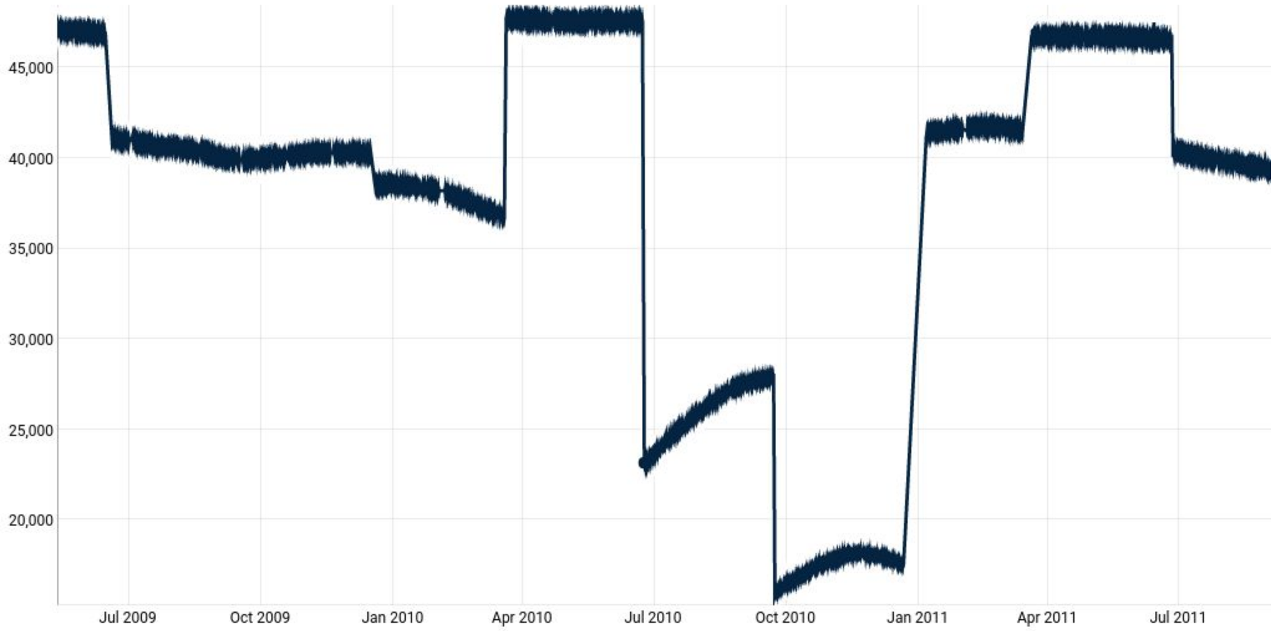
What we have done

- 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

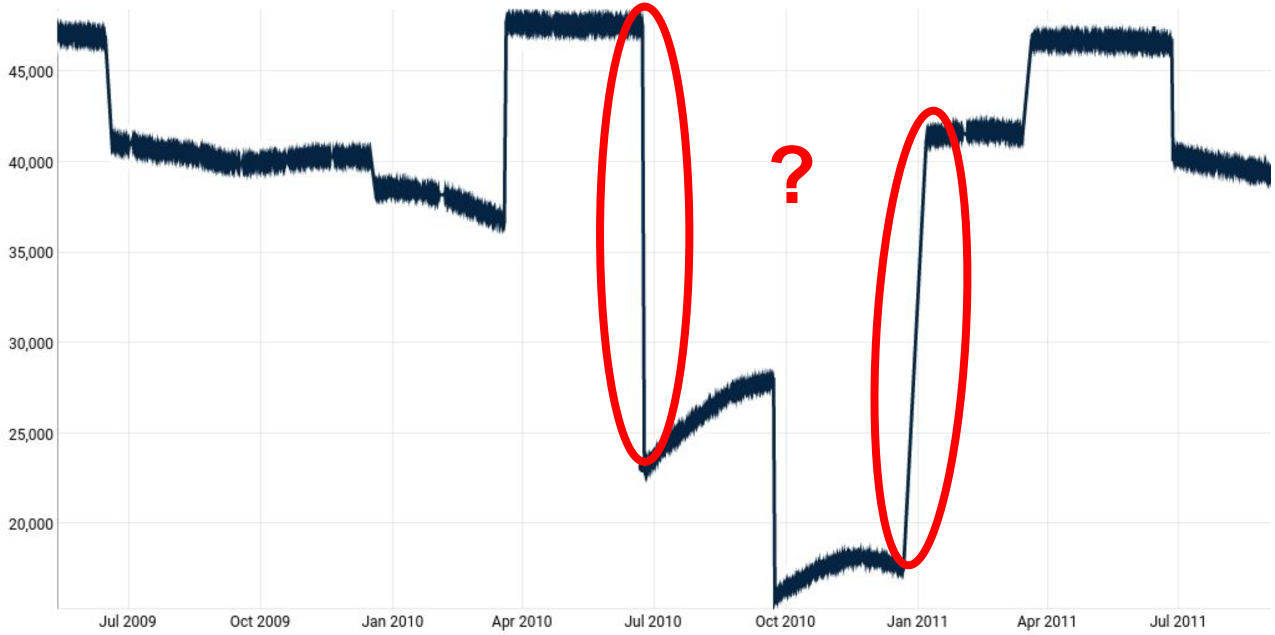


Looking at the raw signal...



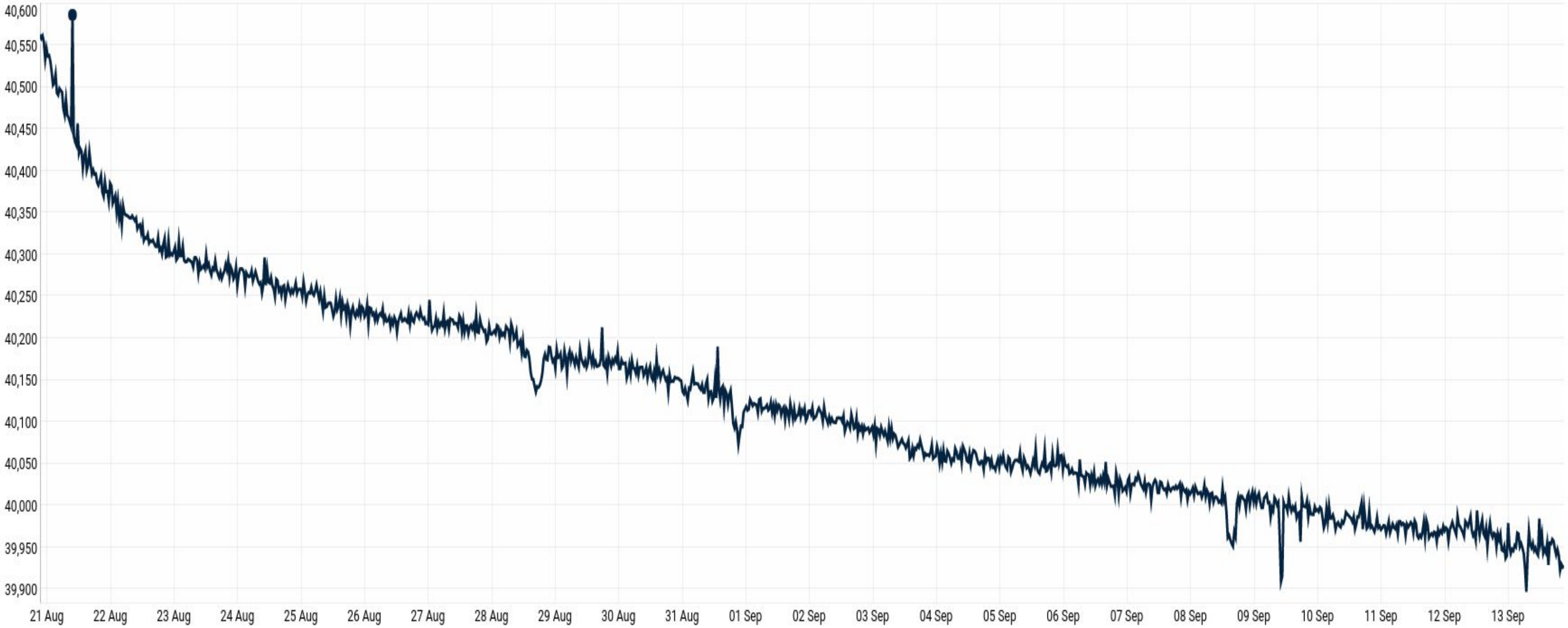
SAP_FLUX:
The flux in units of electrons per second contained in the optimal aperture pixels collected by the spacecraft.

Looking at the raw signal...



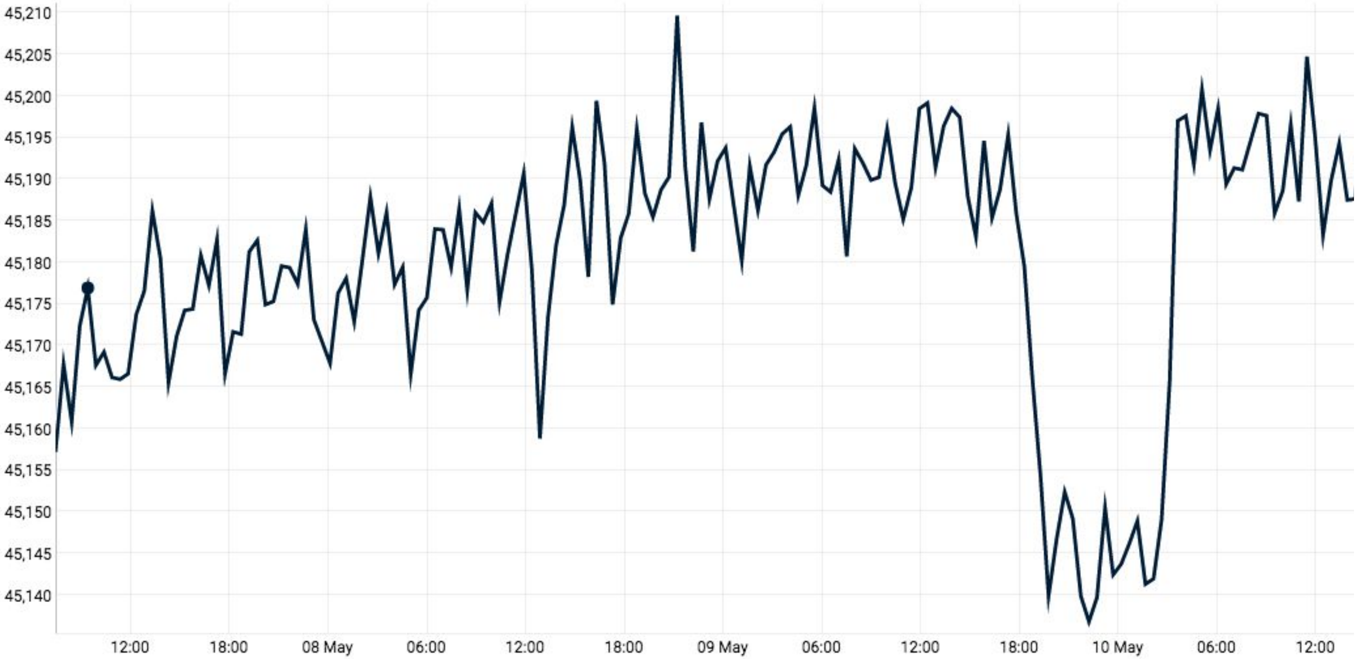
SAP_FLUX:
The flux in units of electrons per second contained in the optimal aperture pixels collected by the spacecraft.

Looking at one record



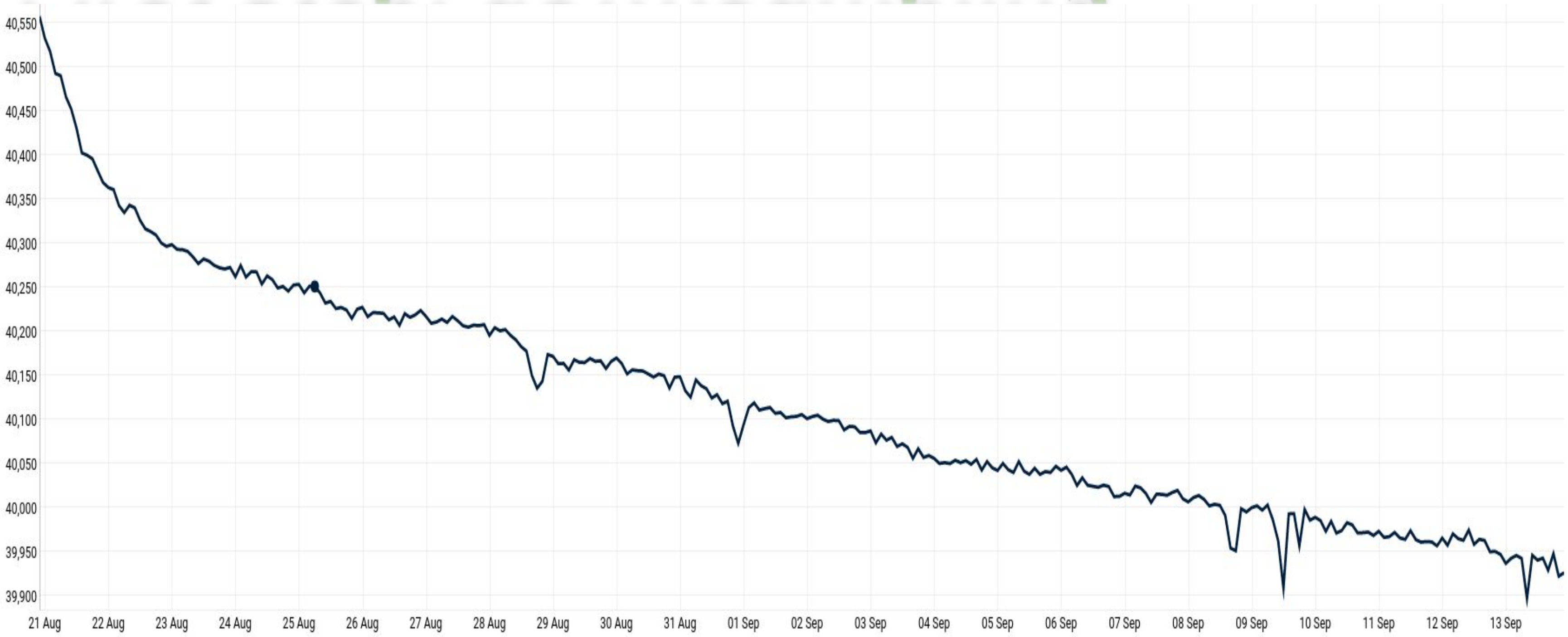
Perturbations in dirty signals

Transits are tiny

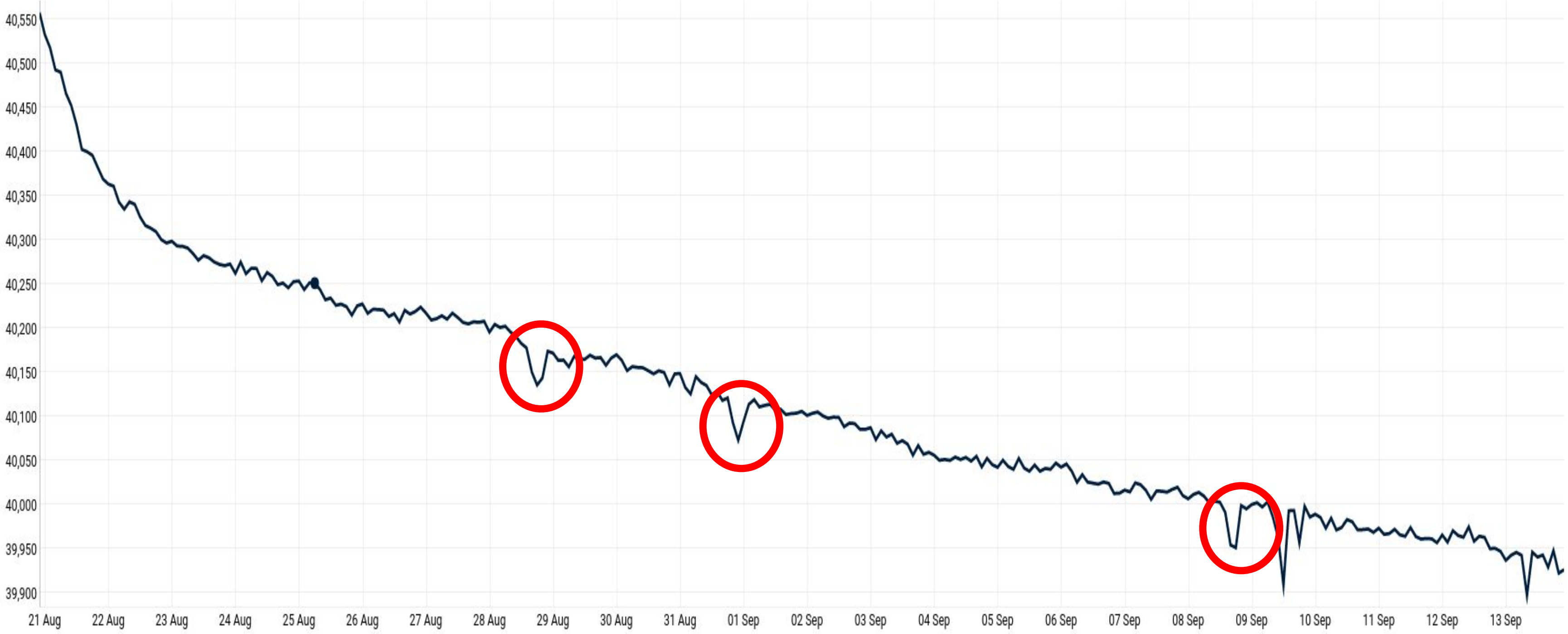


~40 electrons per second

First step: downsampling

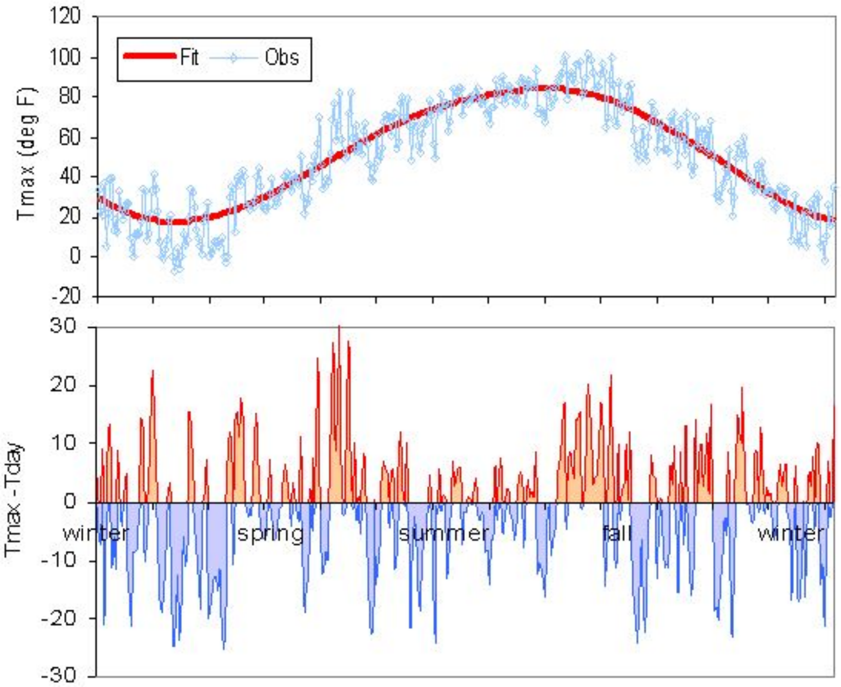


First step: downsampling



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

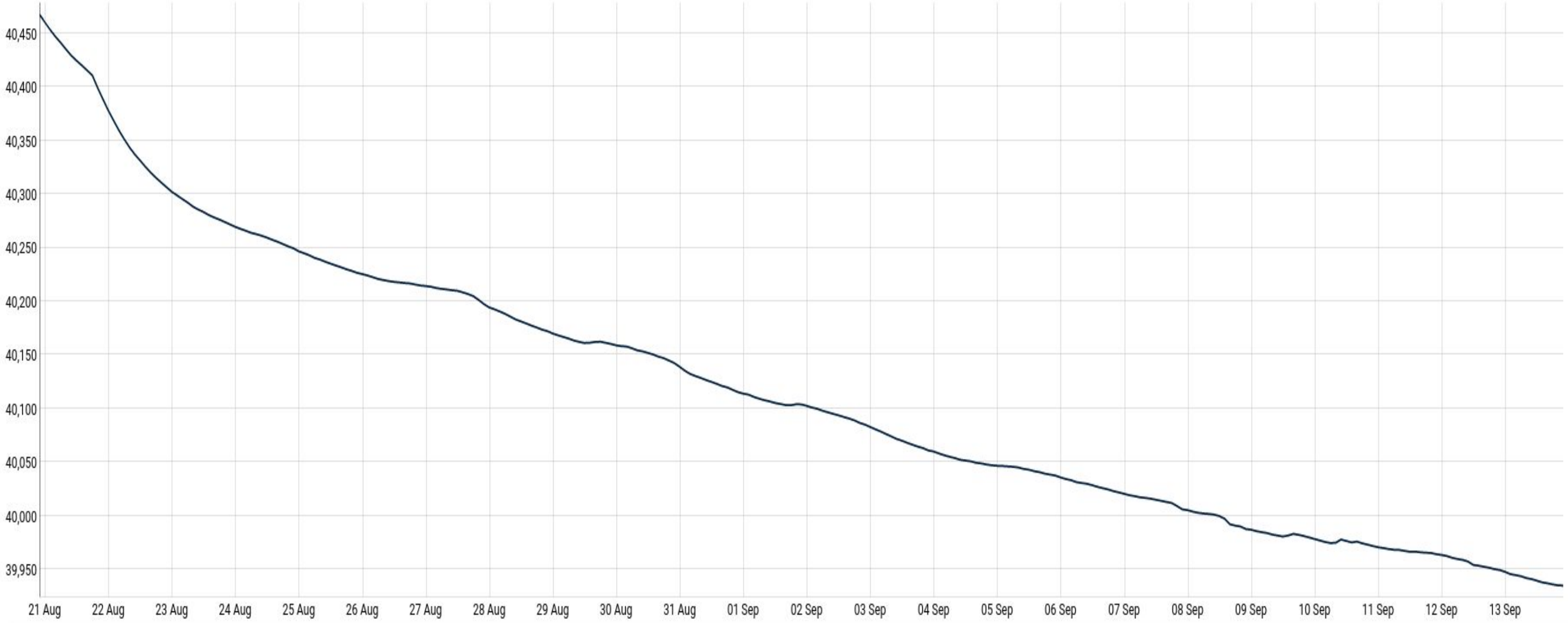
If you ♥ signal processing



High pass filter

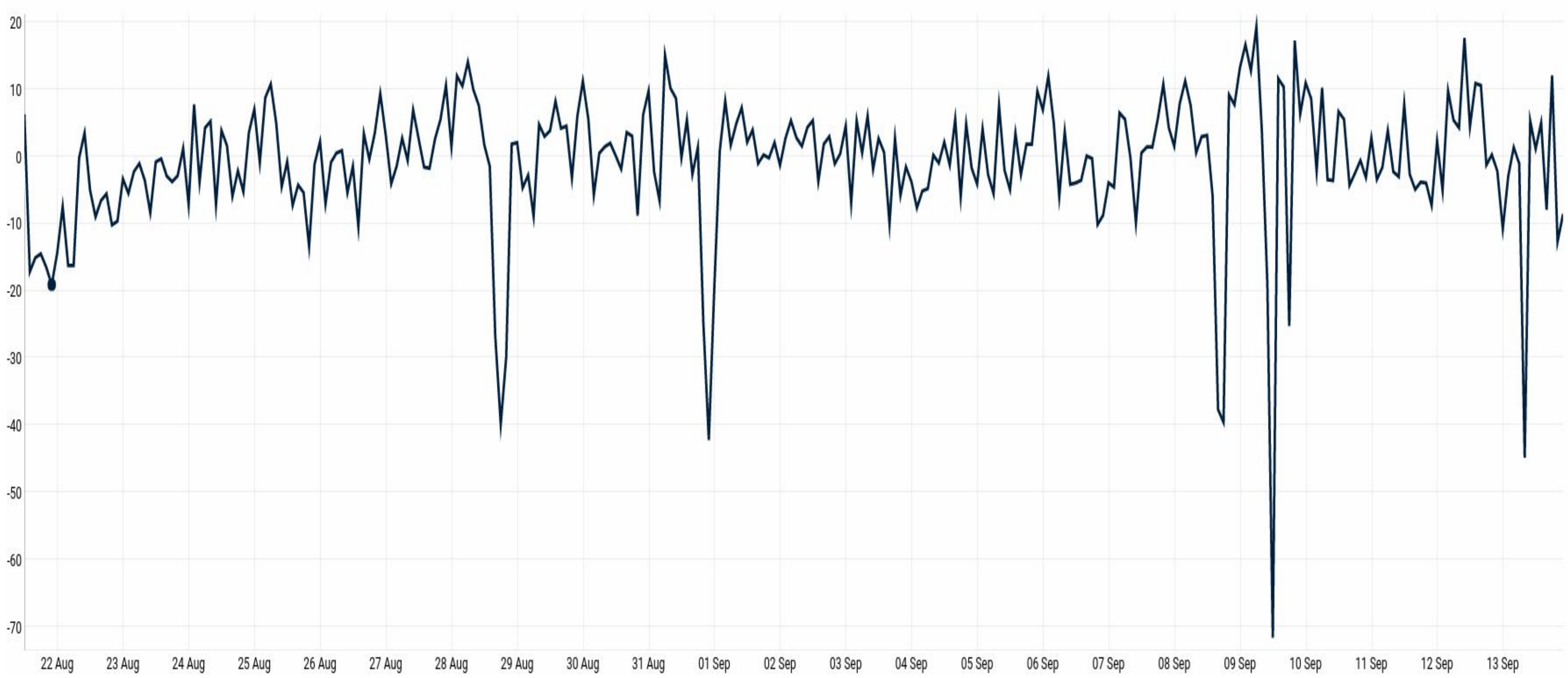
Poor person's high pass filter

Code d'Armor



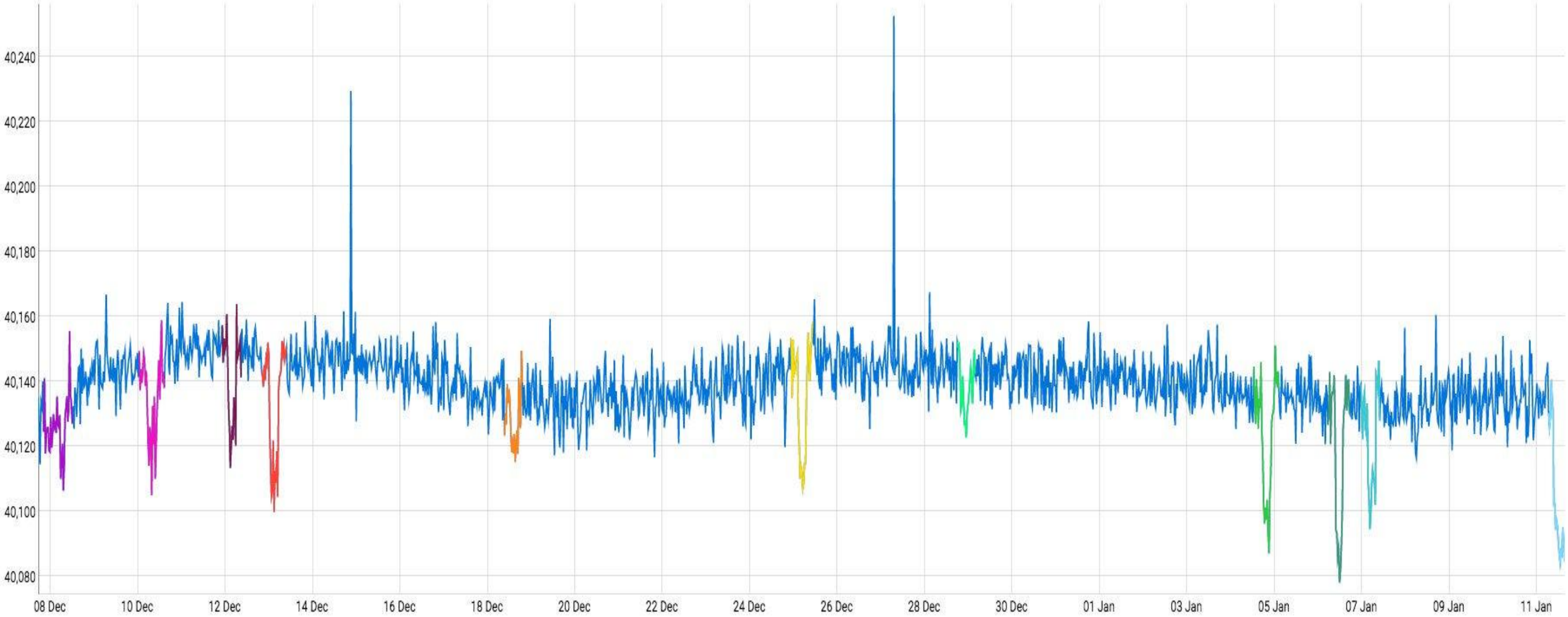
Using the trend

Signal - Trend



Now you can see them well

After some tuning

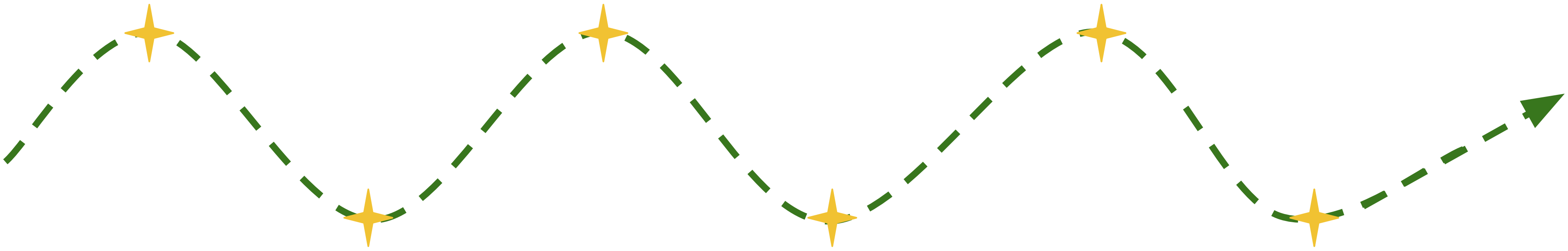


We have our transit candidates

What's next?

Where do we go from here?

Only the beginning



New import method

Better detection

Deep learning

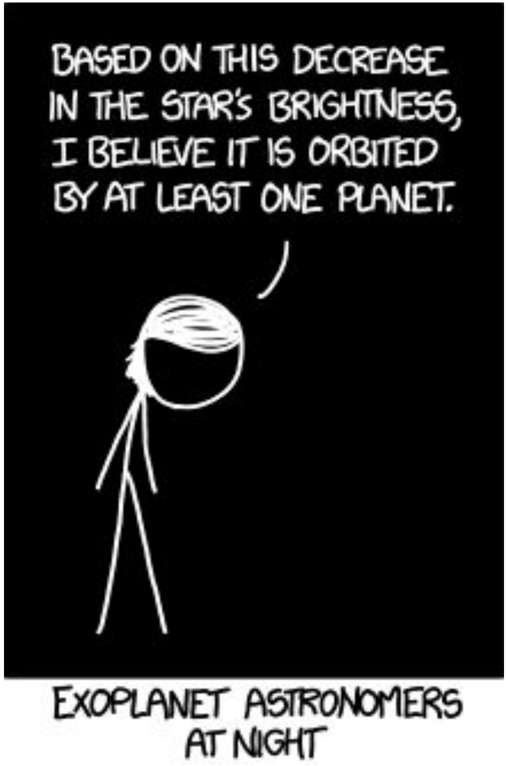
Explorer

satellite/star location

Yours?

A growing team

And you!


BASED ON THIS DECREASE IN THE STAR'S BRIGHTNESS, I BELIEVE IT IS ORBITED BY AT LEAST ONE PLANET.
EXOPLANET ASTRONOMERS AT NIGHT

https://xkcd.com/1371/

Join us!

https://helloexo.world

Thank you!

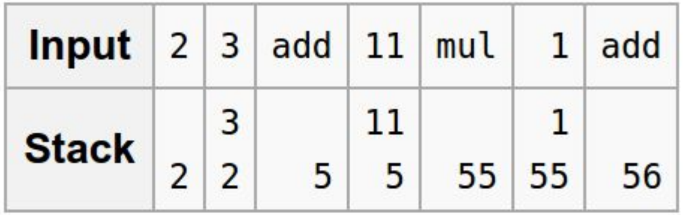
рахмат
danke
謝謝
ngiyabonga
شكراً جزيلاً
spas
Баярлалаа
mersi
kia ora
barka
welalin
tack
teşekkür ederim
mahalo
спасибо
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
tapadh leat
kiitos
dankie
nanni
nandri
dhanyavad
thank you
gracias
хвала
asante
manana
bedankt
bayarlalaa
gracje
hvala
maururu
köszönöm
akun
dankon
aciu
obrigada
murakoze
tenki
enkosi
djiere dieuf
tau
mochchakkeram
дякую
mamnun
chokrane
dziękuję
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
go raibh maith agat
sobodi
dēkuji
sukriya
kop khun krap
arigatô
takk
dakujem
trugarez
obrigado
mèsi
didi madloba
sagolun
najis tuke
ありがとう
tanemirt
grazie
shukriya
merce
мерси
kam sah hamnida
terima kasih
rahmat
rahmet
ευχαριστώ
diolch
dhanyavadagalu
তোমাকে ধন্যবাদ
감사합니다
xièxie
merci

Want to know more?

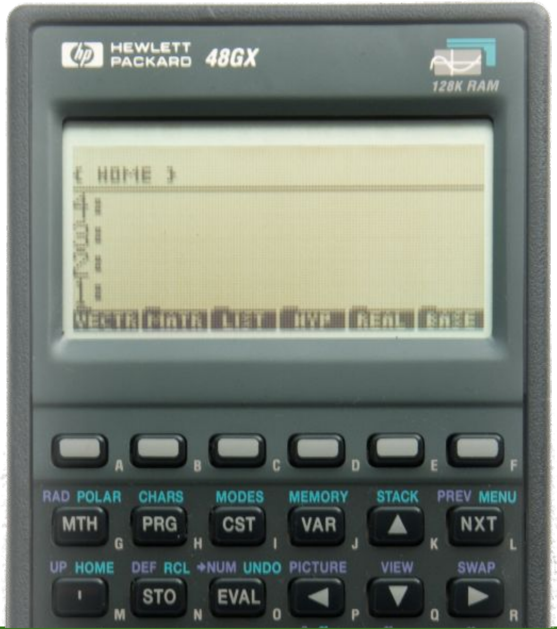
Analysing with WarpScript

WarpScript

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
```

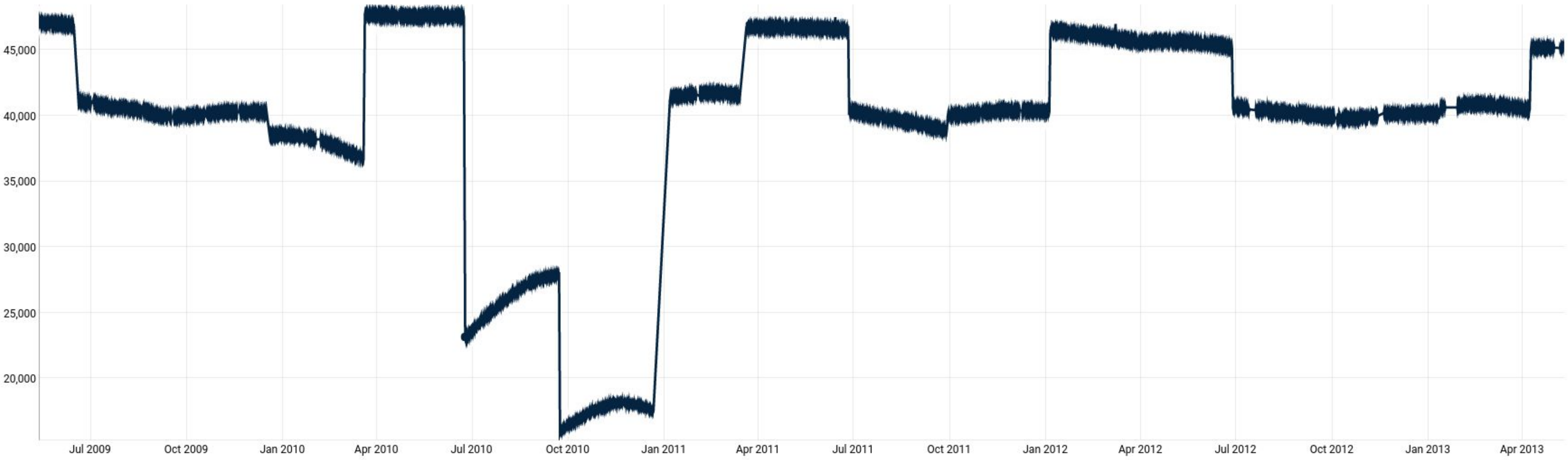
What are the available series?

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

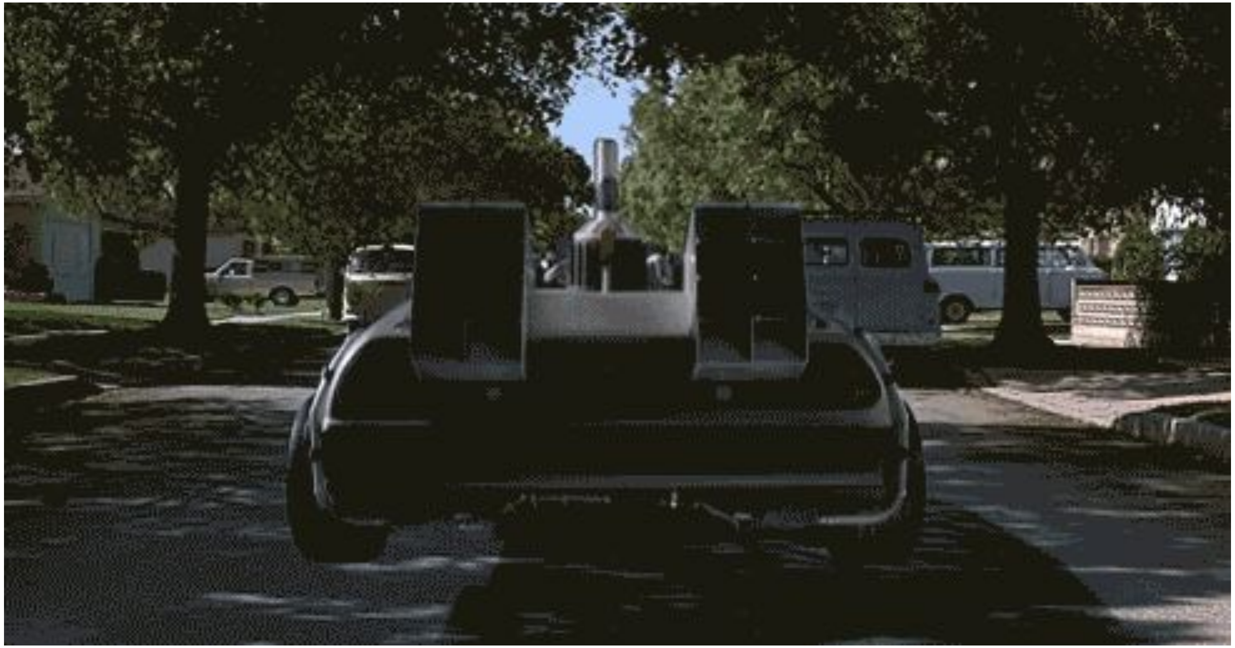
Get raw data

```
[
        $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
```

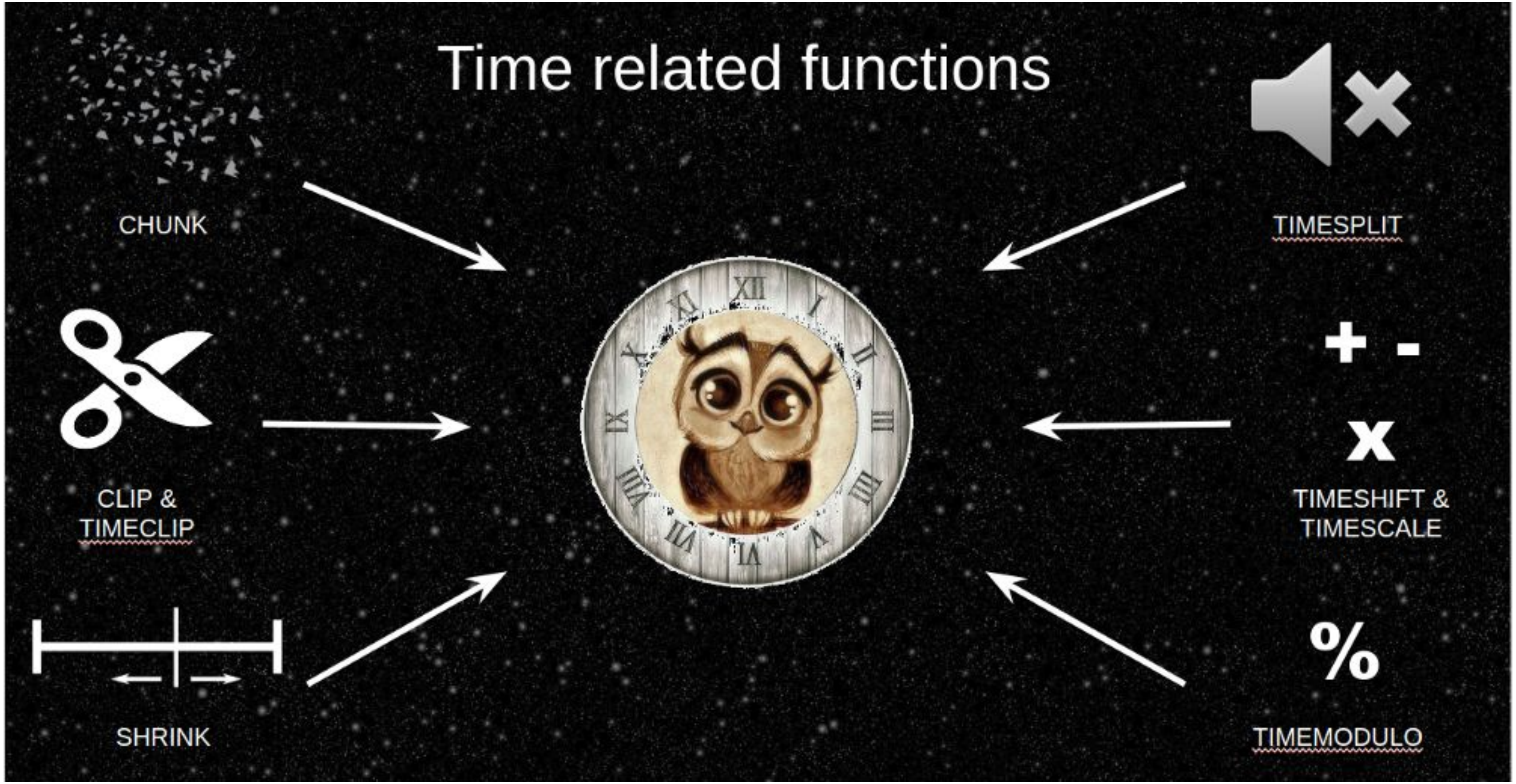
Kepler-11: Raw data



Time manipulation



Time related functions



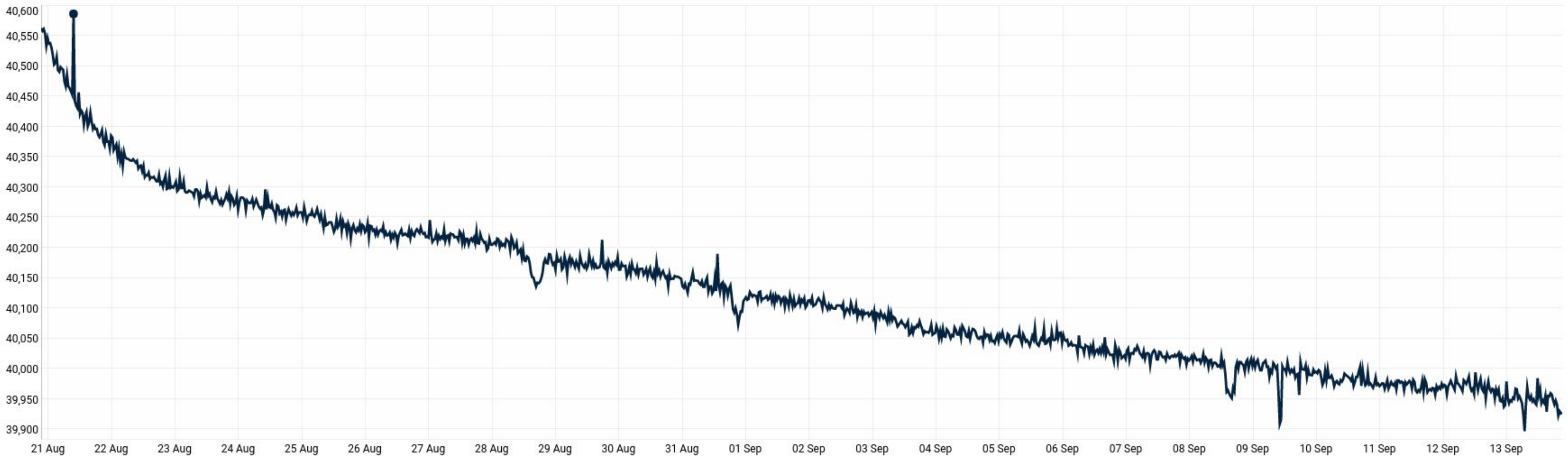
How to split a Time series

```
$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
```

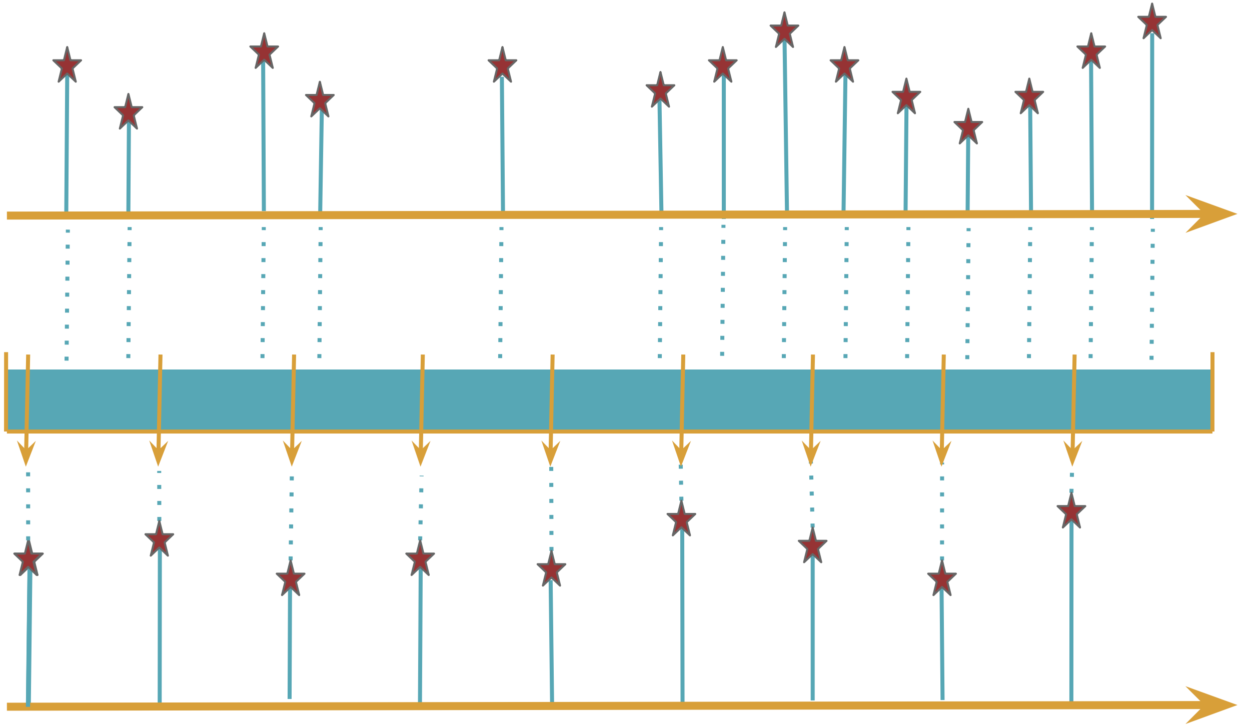
Filtering

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

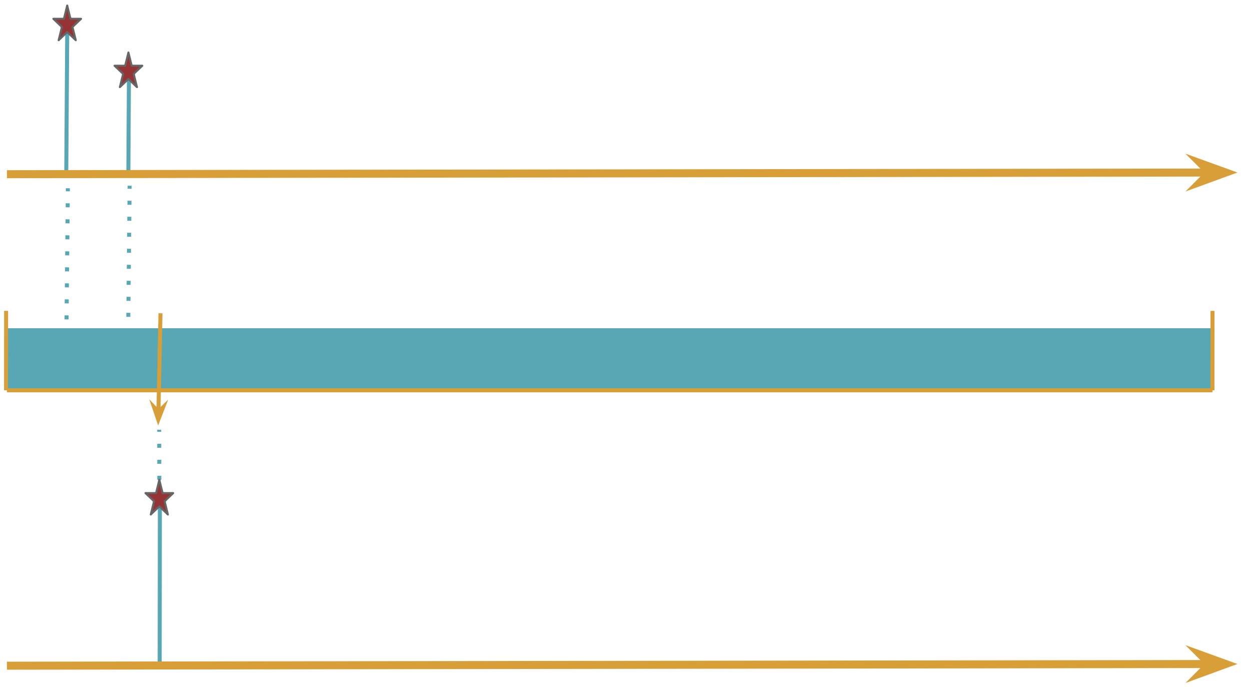
Reference record: 5



Downsampling

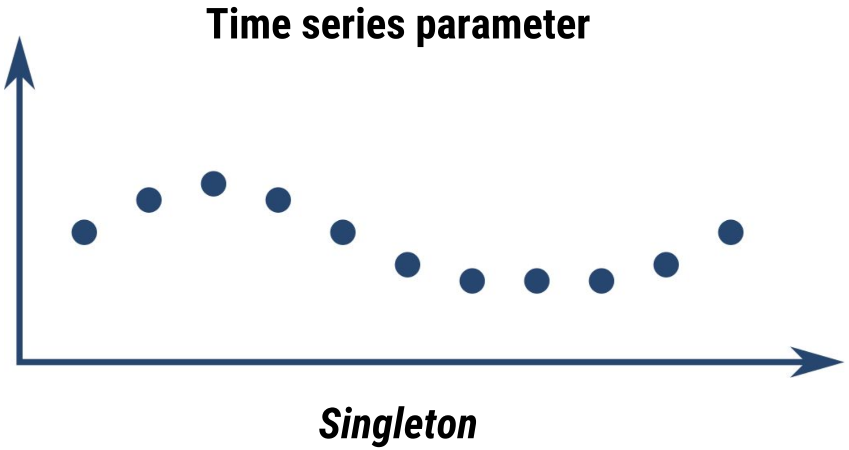


Bucketize



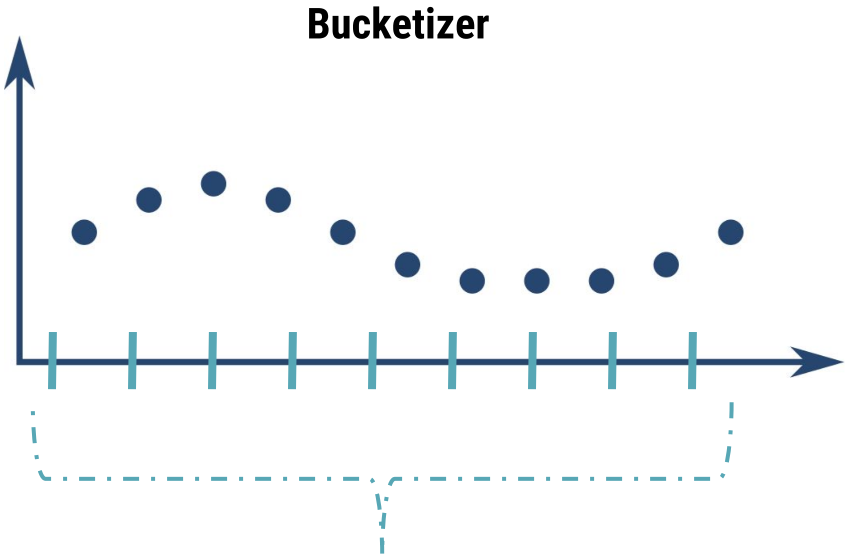
Syntax

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



Syntax

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

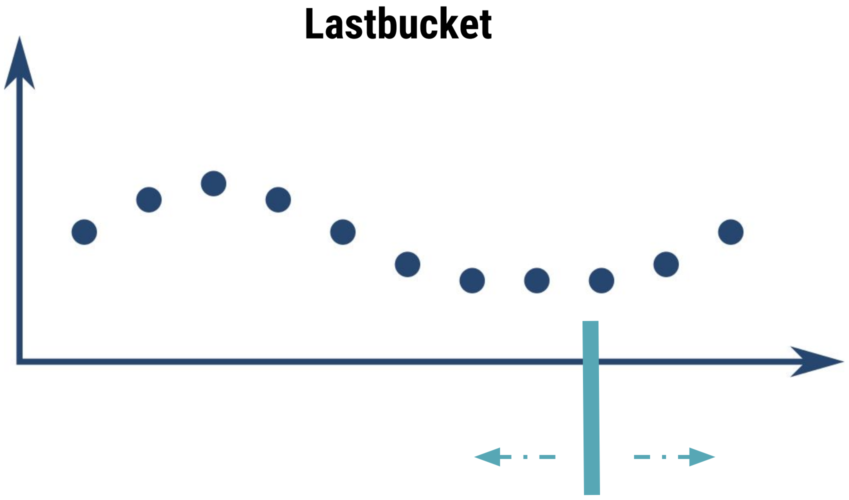


Type of operator to apply on each bucket

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

Syntax

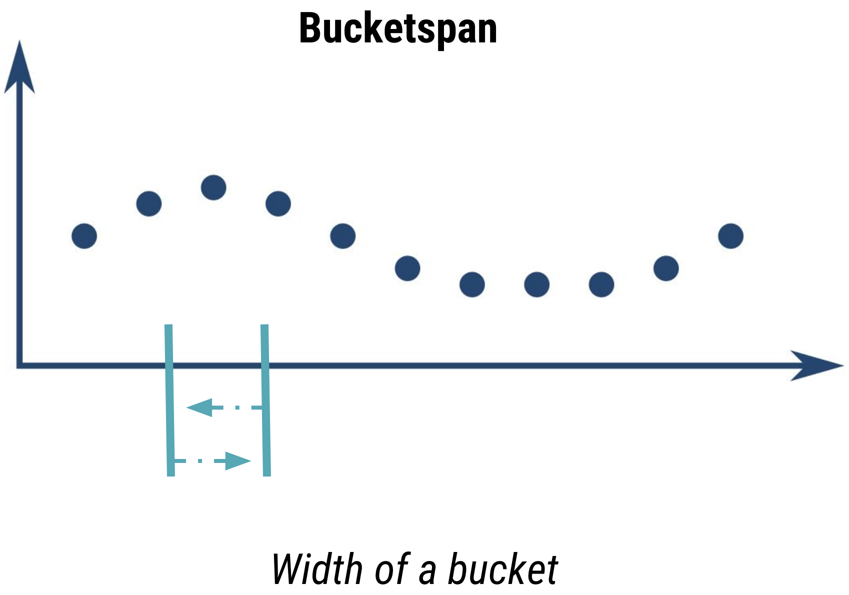
```
[
    $gts
    bucketizer.min
    0
    2 h
    0
]
BUCKETIZE
```



End timestamp of the more recent bucket

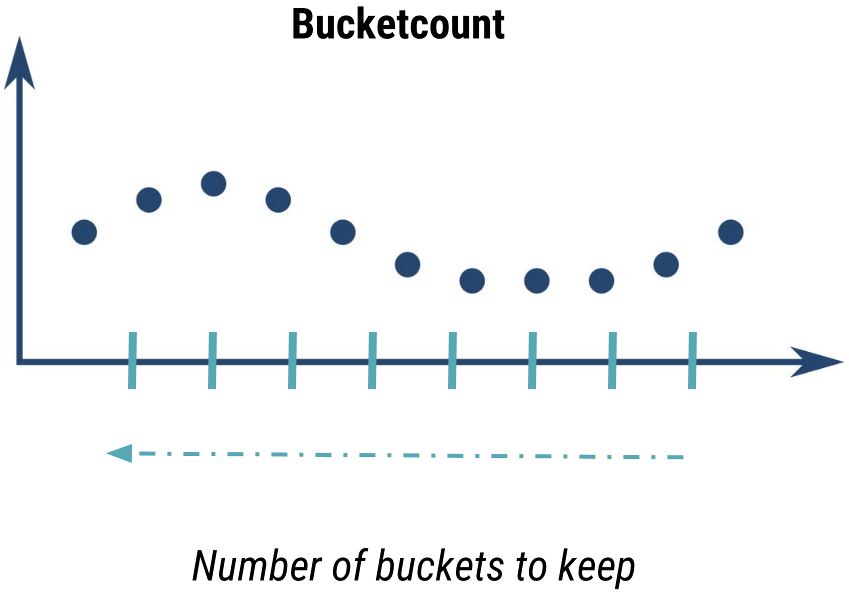
Syntax

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

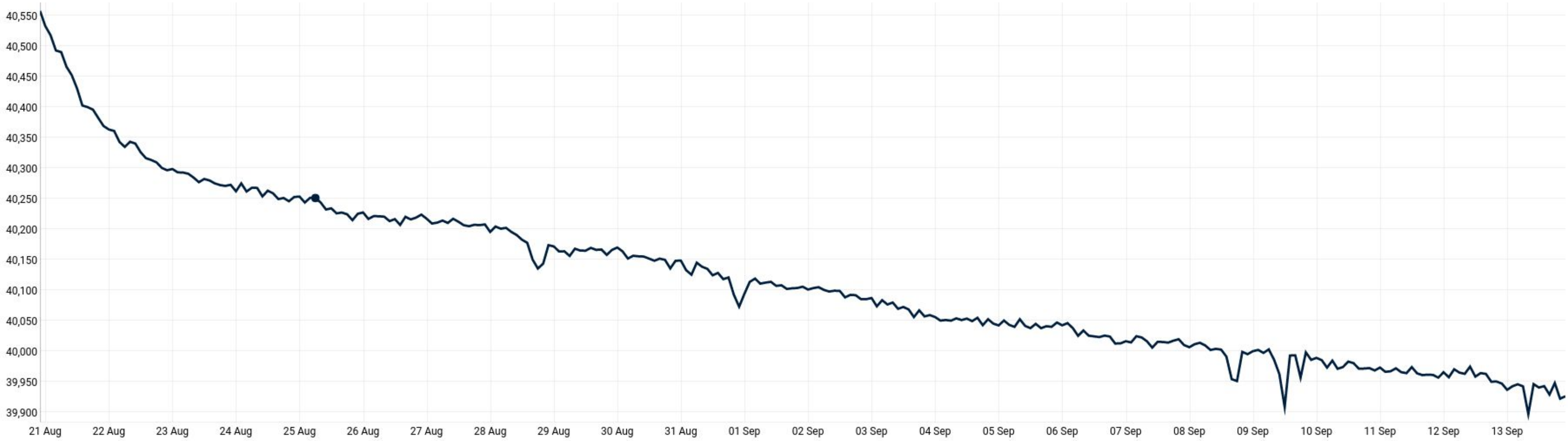


Syntax

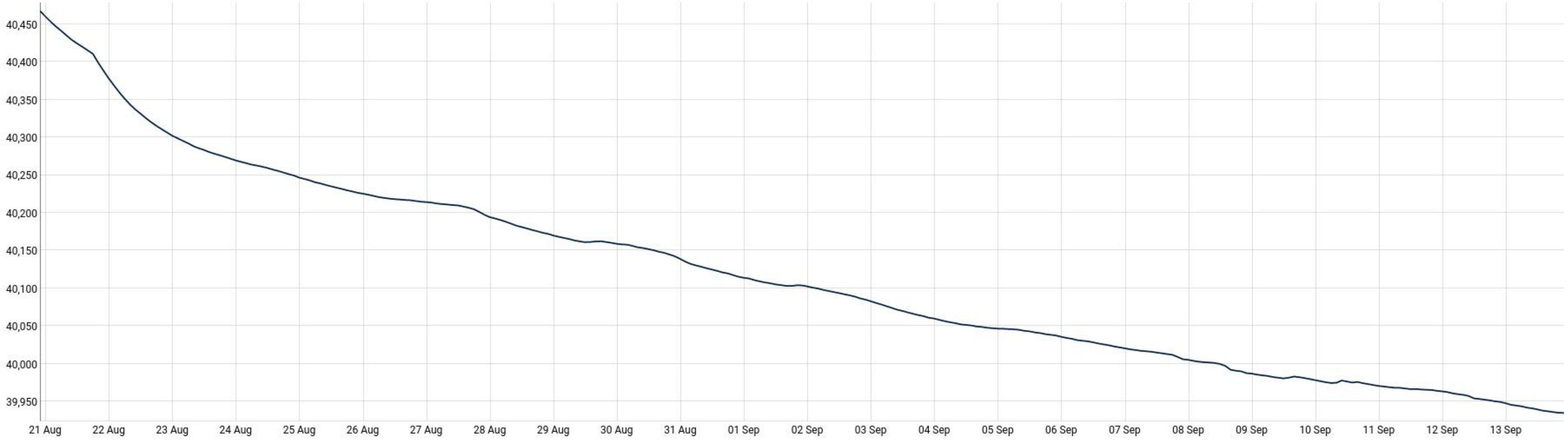
```
[
        $gts
        bucketizer.min
        0
        2 h
        0
]
BUCKETIZE
```



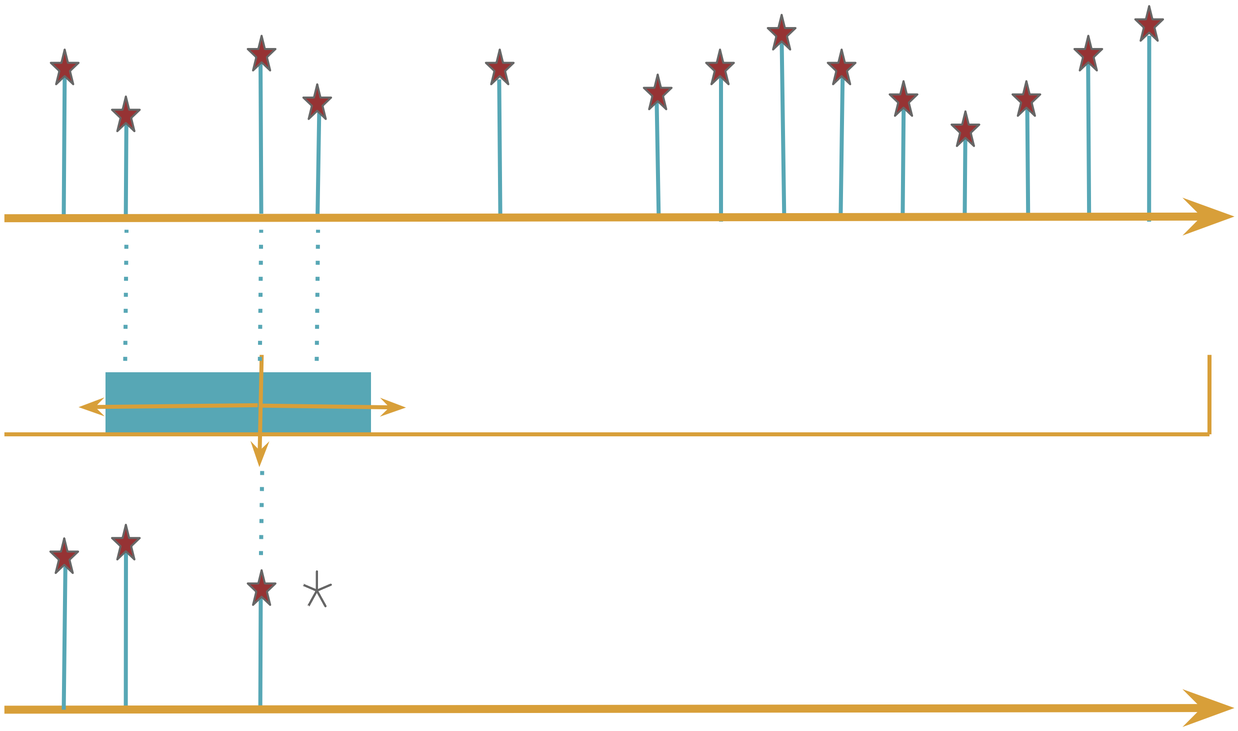
Actual


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



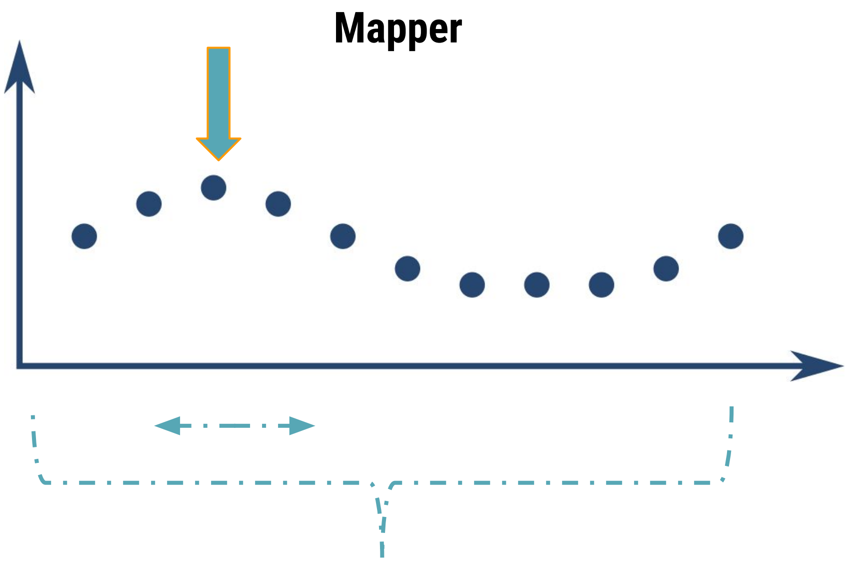
Syntax

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

Time series parameter

Syntax

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

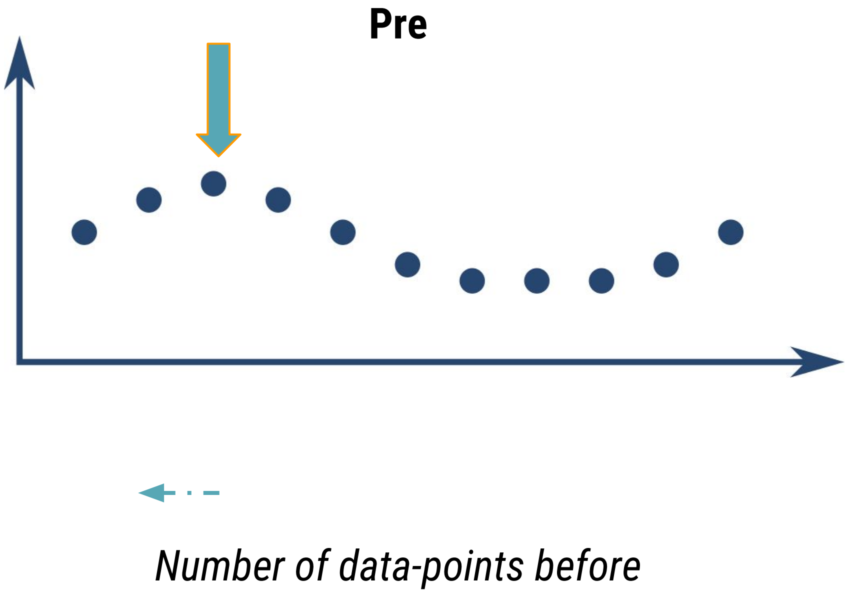


Type of operator to apply on each window

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

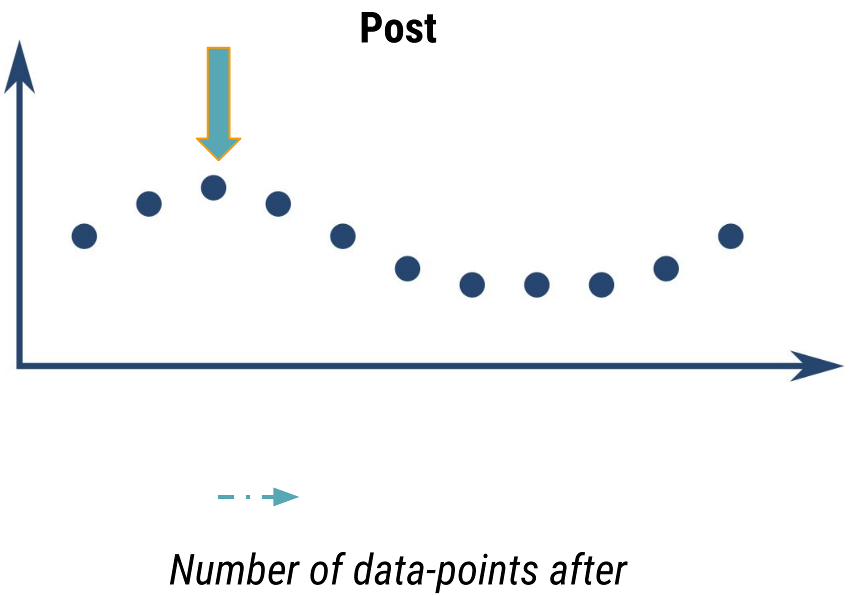
Syntax

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



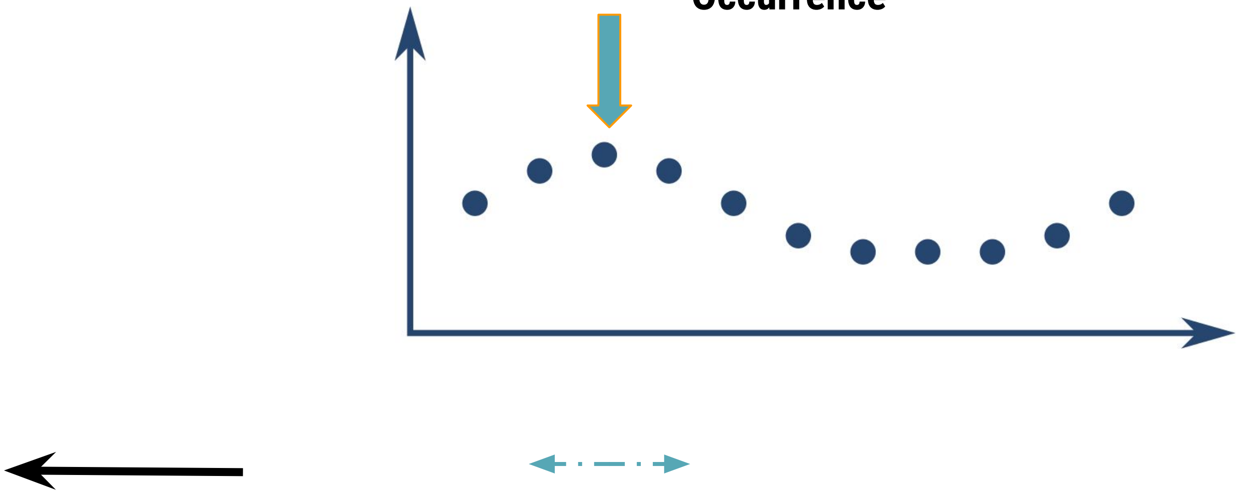
Syntax

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



Syntax

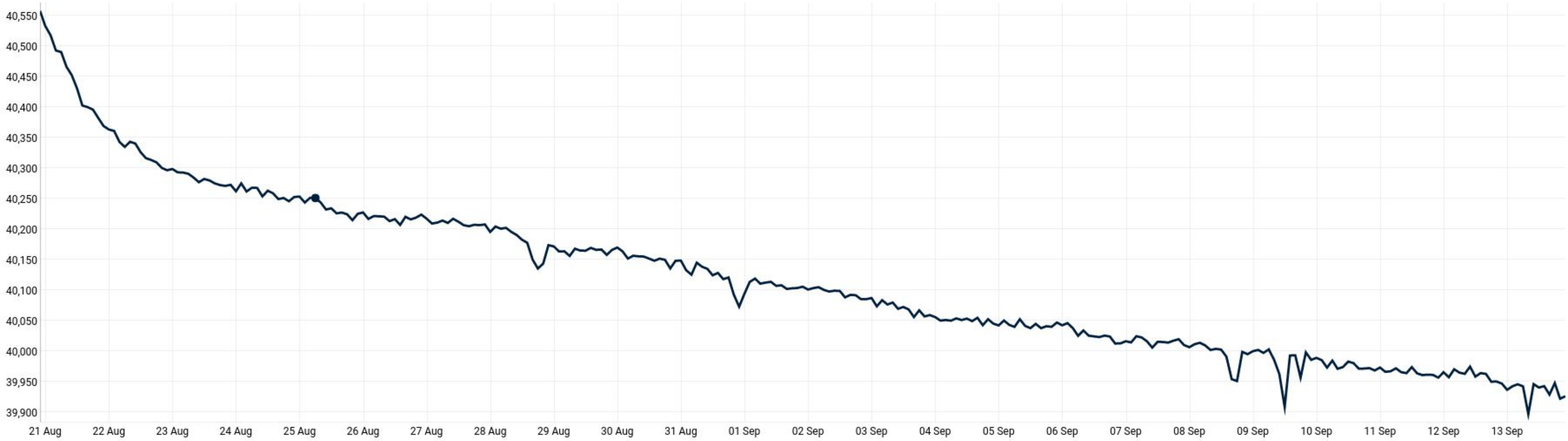
```
[
        $gts
        mapper.mean
        2
        2
        0
]
MAP
```



Occurrence

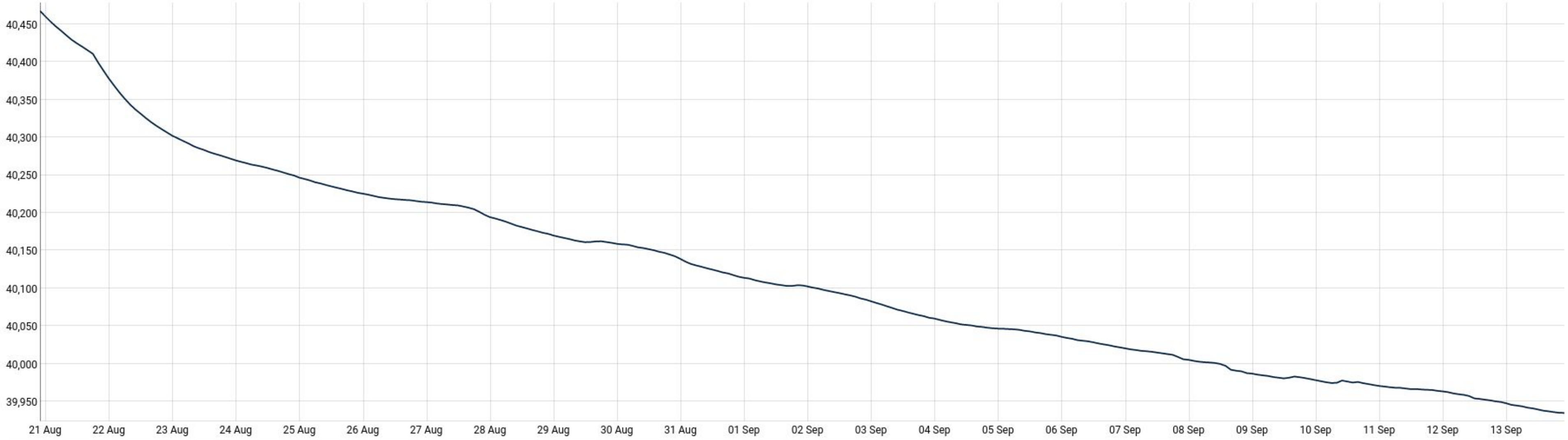
Maximal number of calculation for a data-point

Actual



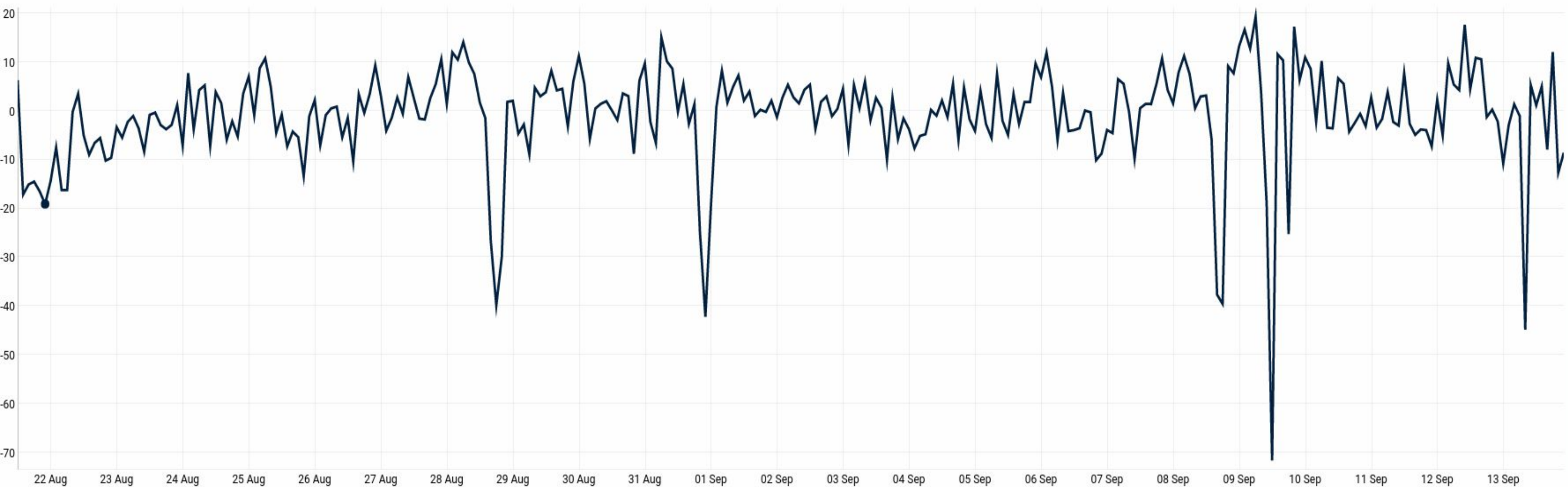
40,550
40,500
40,450
40,400
40,350
40,300
40,250
40,200
40,150
40,100
40,050
40,000
39,950
39,900
21 Aug
22 Aug
23 Aug
24 Aug
25 Aug
26 Aug
27 Aug
28 Aug
29 Aug
30 Aug
31 Aug
01 Sep
02 Sep
03 Sep
04 Sep
05 Sep
06 Sep
07 Sep
08 Sep
09 Sep
10 Sep
11 Sep
12 Sep
13 Sep

Trend

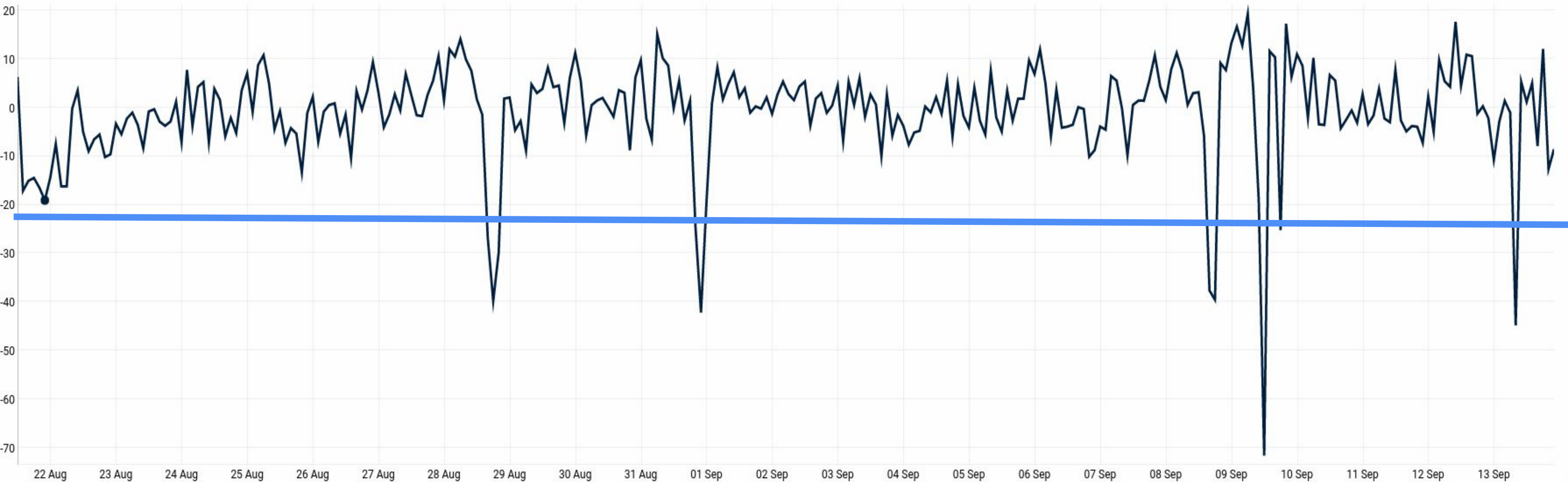


40,450
40,400
40,350
40,300
40,250
40,200
40,150
40,100
40,050
40,000
39,950
21 Aug
22 Aug
23 Aug
24 Aug
25 Aug
26 Aug
27 Aug
28 Aug
29 Aug
30 Aug
31 Aug
01 Sep
02 Sep
03 Sep
04 Sep
05 Sep
06 Sep
07 Sep
08 Sep
09 Sep
10 Sep
11 Sep
12 Sep
13 Sep

Actual - trend



Actual - trend



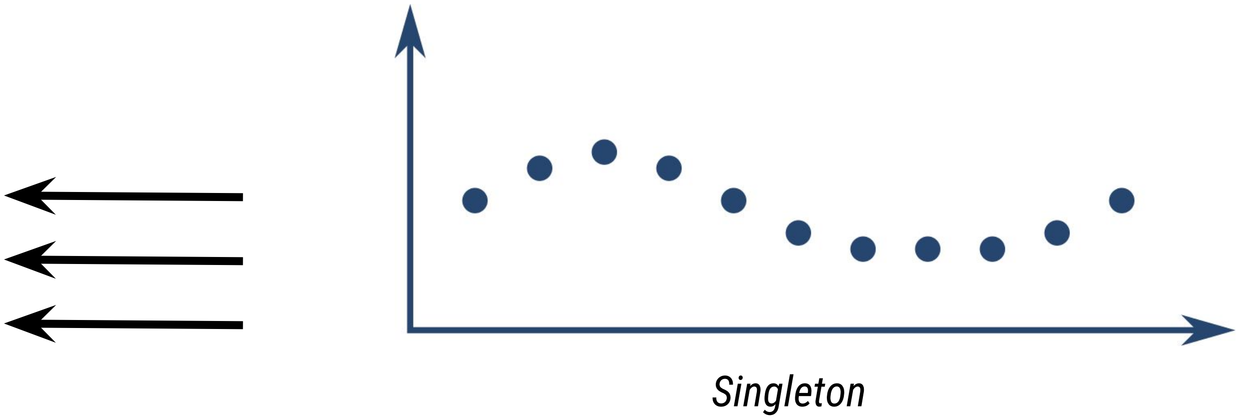
Time to level-up!

Time series operation

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

Syntax

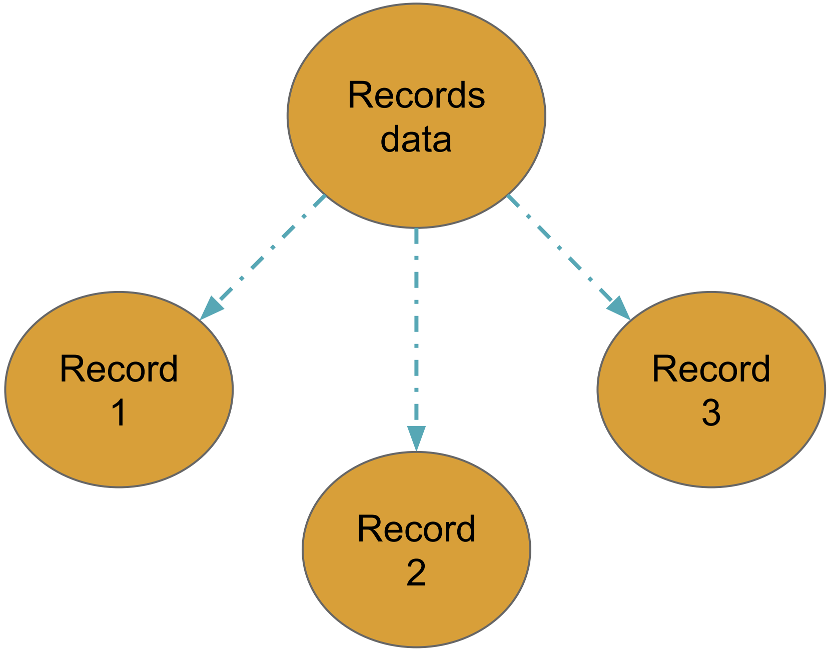
```
[
    $gts0
    …
    $gtsN
    [ ‘record’ ]
    op.add
]
APPLY
```



Syntax

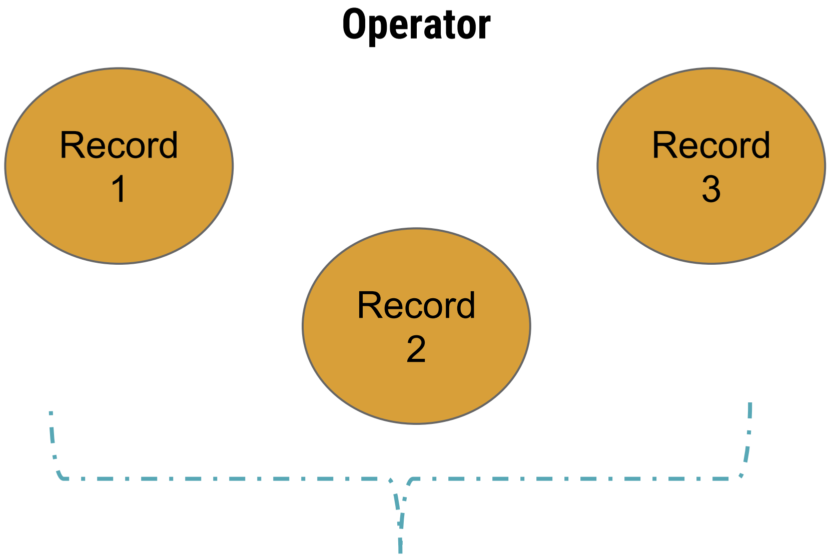
```
[
        $gts0
        …
        $gtsN
        [ 'record' ]
        op.add
]
APPLY
```

Equivalence class



Syntax

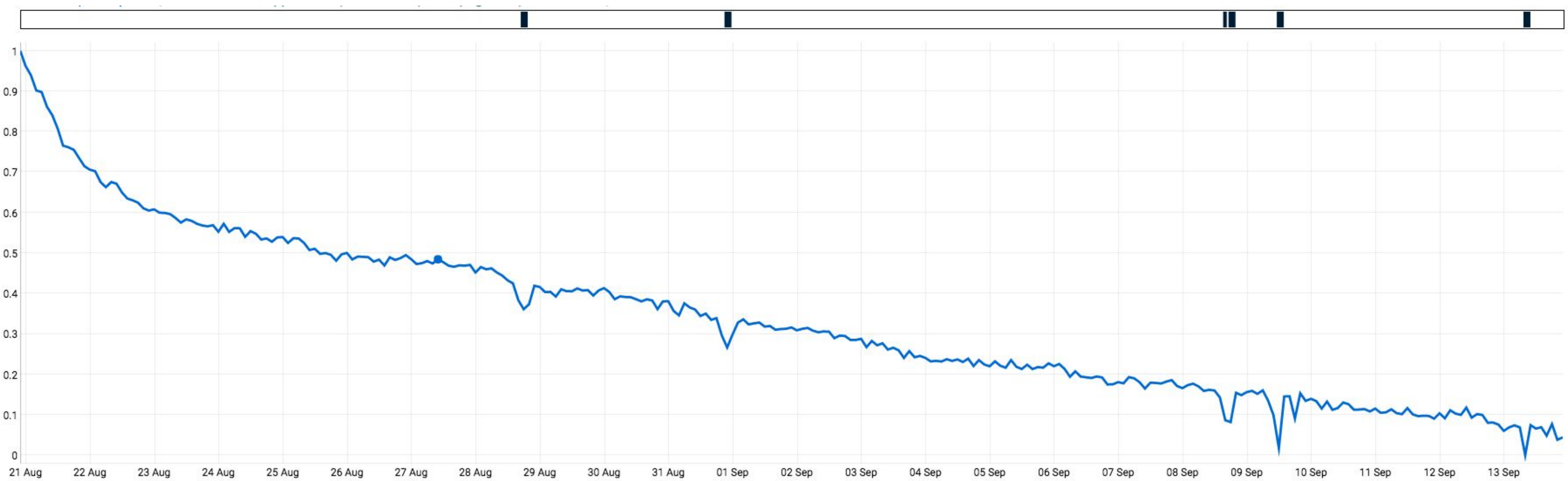
```
[
        $gts0
        ...
        $gtsN
        [ 'record' ]
        op.add
]
APPLY
```



Type of operator to apply on each class

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

Final result


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