The Sum of Nothing

Doing math with missing data in Python via *pandas*

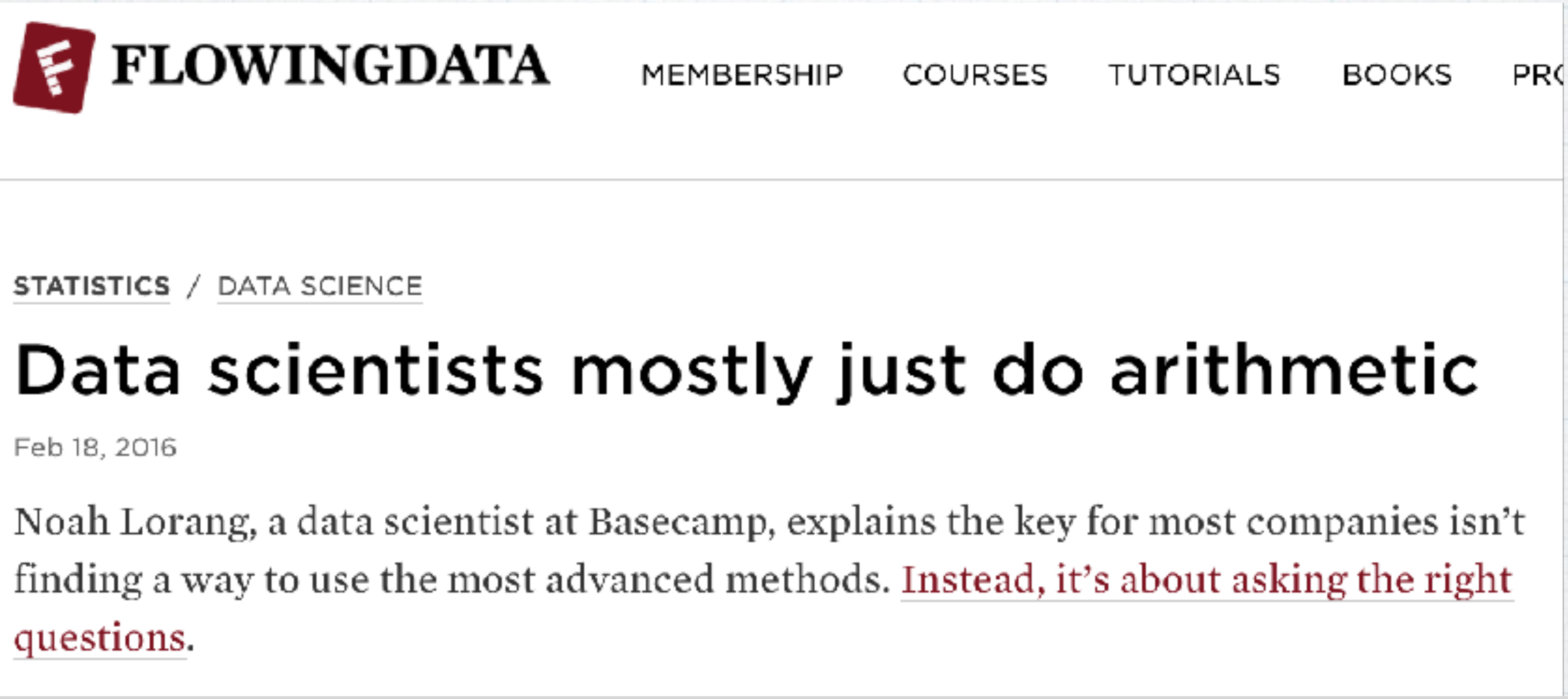
Christine Zhang (@christinezhang)

PyGotham 2018

@christinezhang

LIGHT FOR ALL

THE BALTIMORE SUN



STATISTICS / DATA SCIENCE

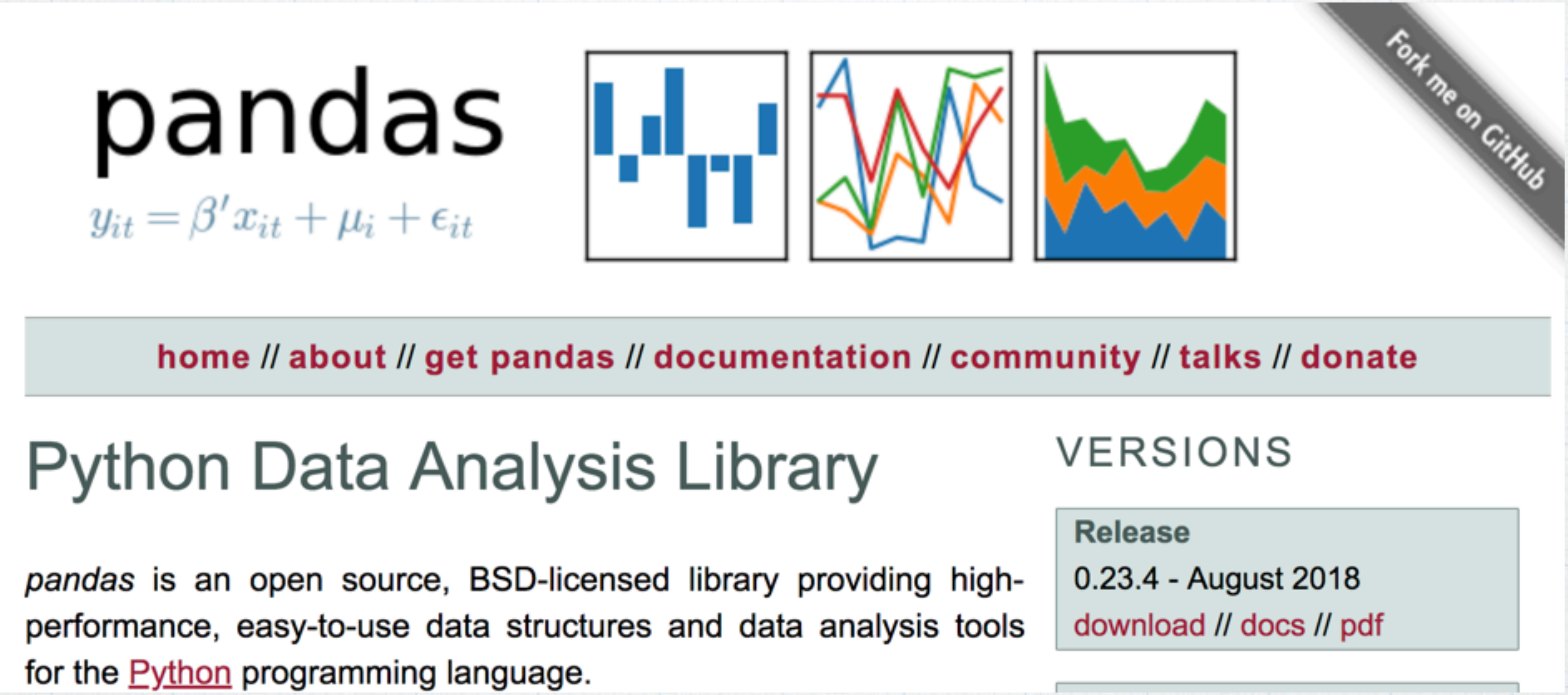
Data scientists mostly just do arithmetic

Feb 18, 2016

Noah Lorang, a data scientist at Basecamp, explains the key for most companies isn't finding a way to use the most advanced methods. Instead, it's about asking the right questions.

Sometimes I'm called a "data scientist."
Mostly, I just do arithmetic, and I'm ok with that.

— Noah Lorang



pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

home // about // get pandas // documentation // community // talks // donate

Python Data Analysis Library

pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

VERSIONS

Release

0.23.4 - August 2018

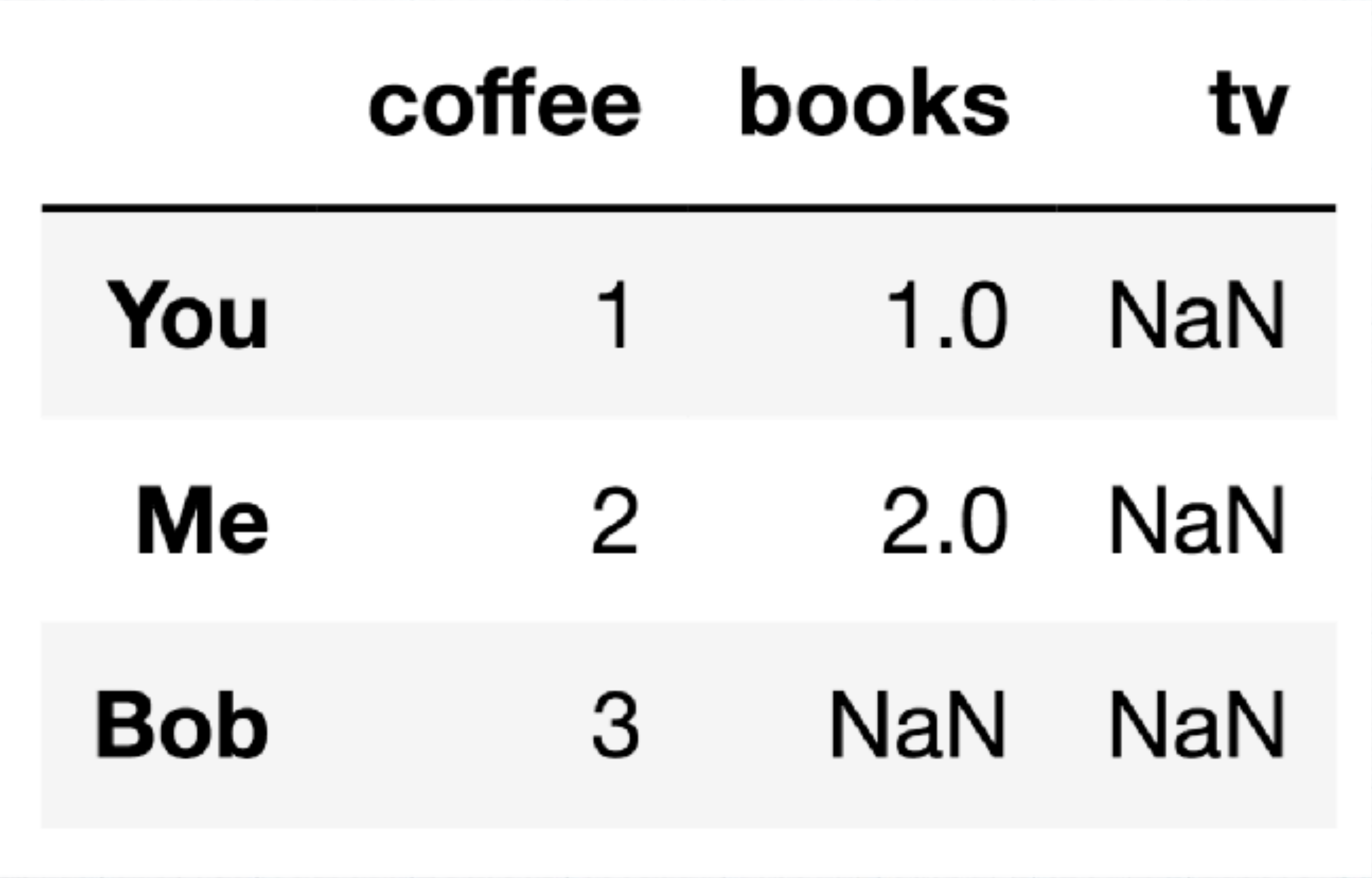
download // docs // pdf

How many **cups of coffee** have you had today?

How many **books** have you read this month?

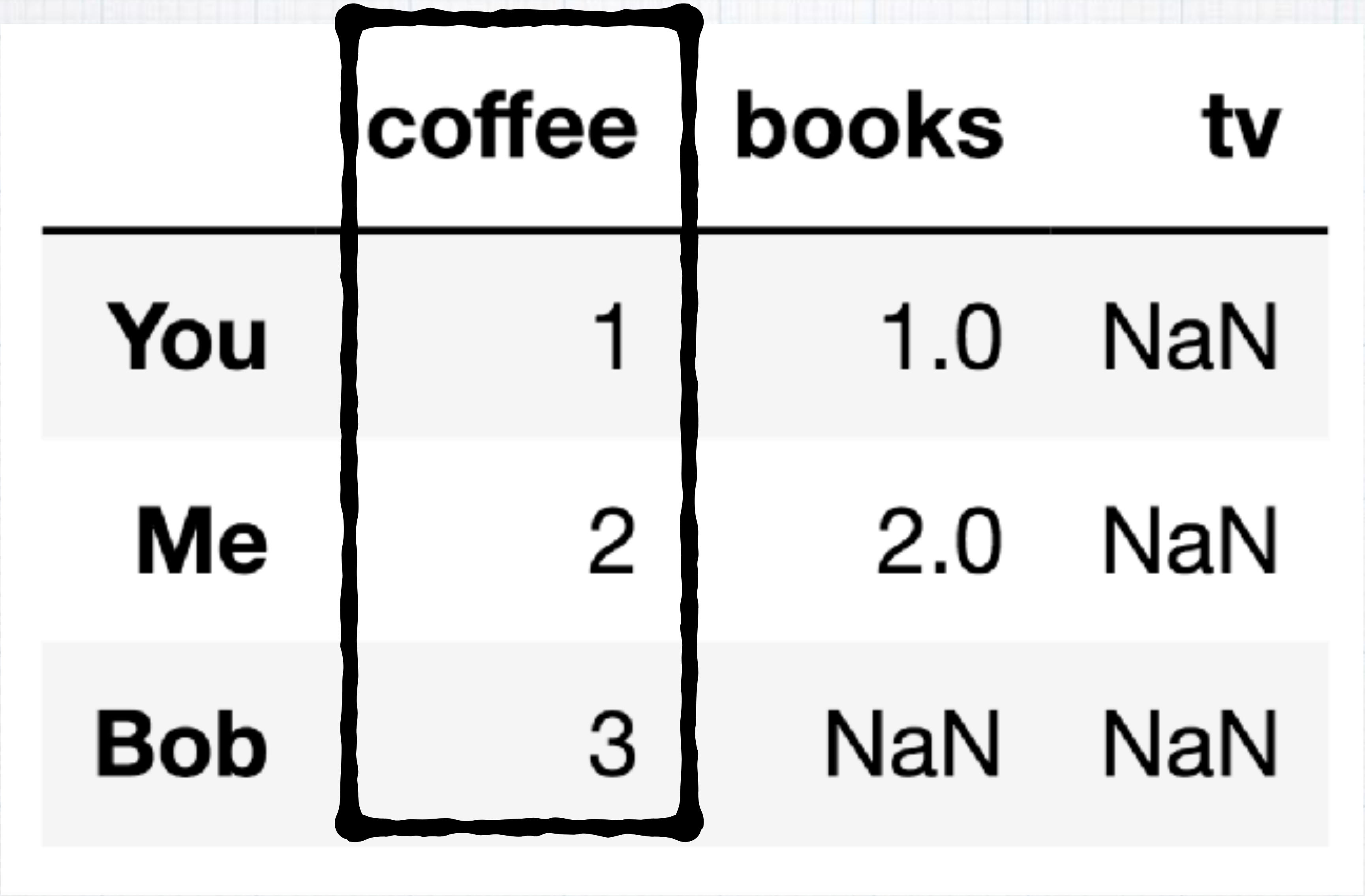
How many **hours of TV** have you watched this week?

Pandas dataframe: `d`



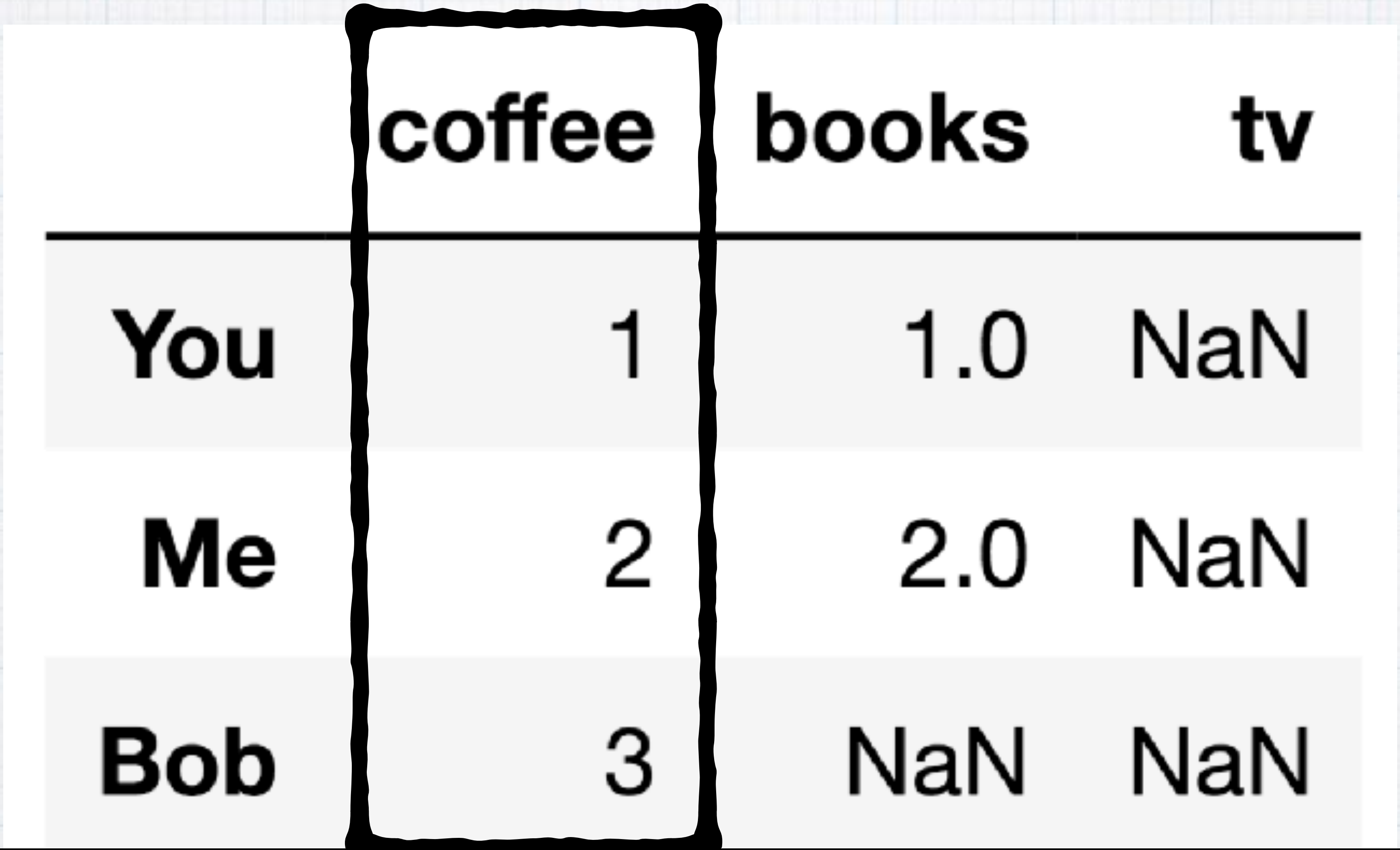
	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN

Pandas dataframe: `d`

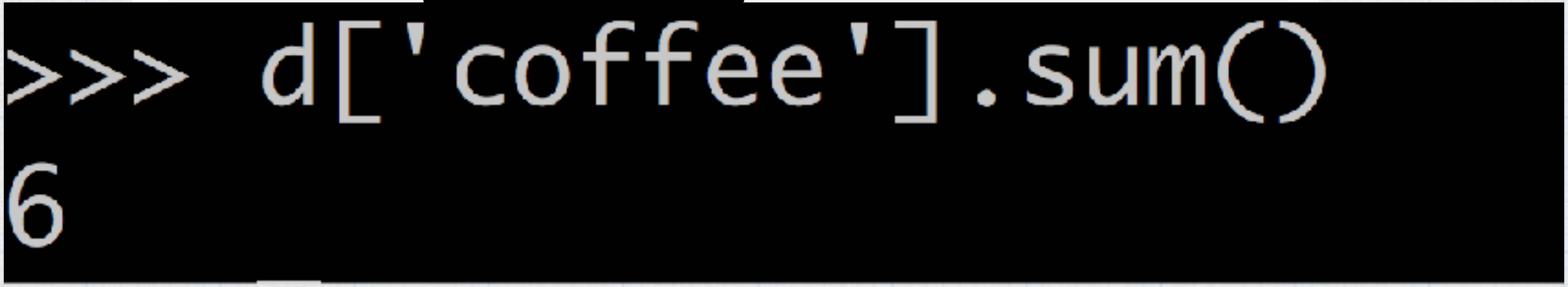


	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN

Pandas dataframe: `d`

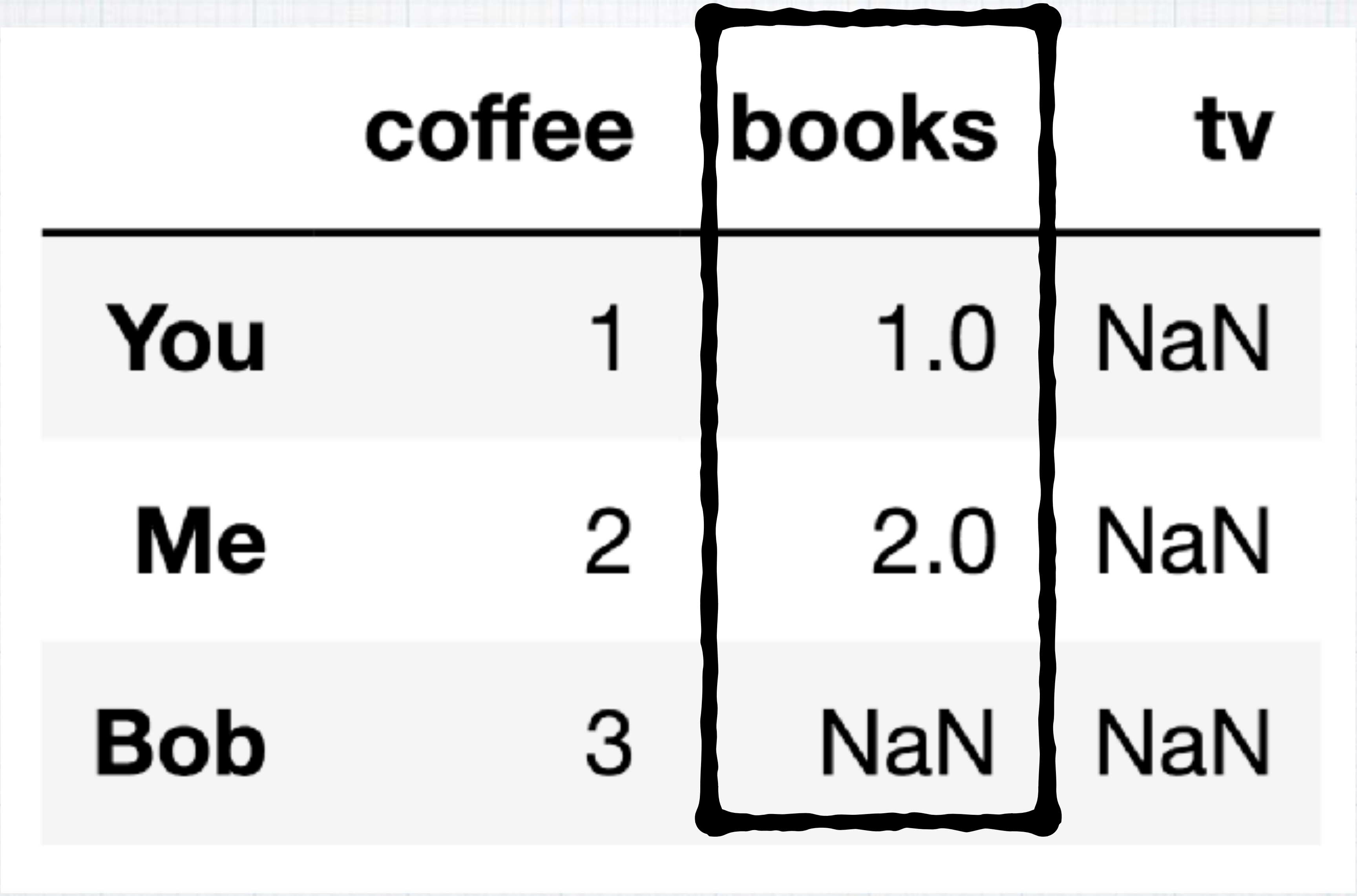


	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN



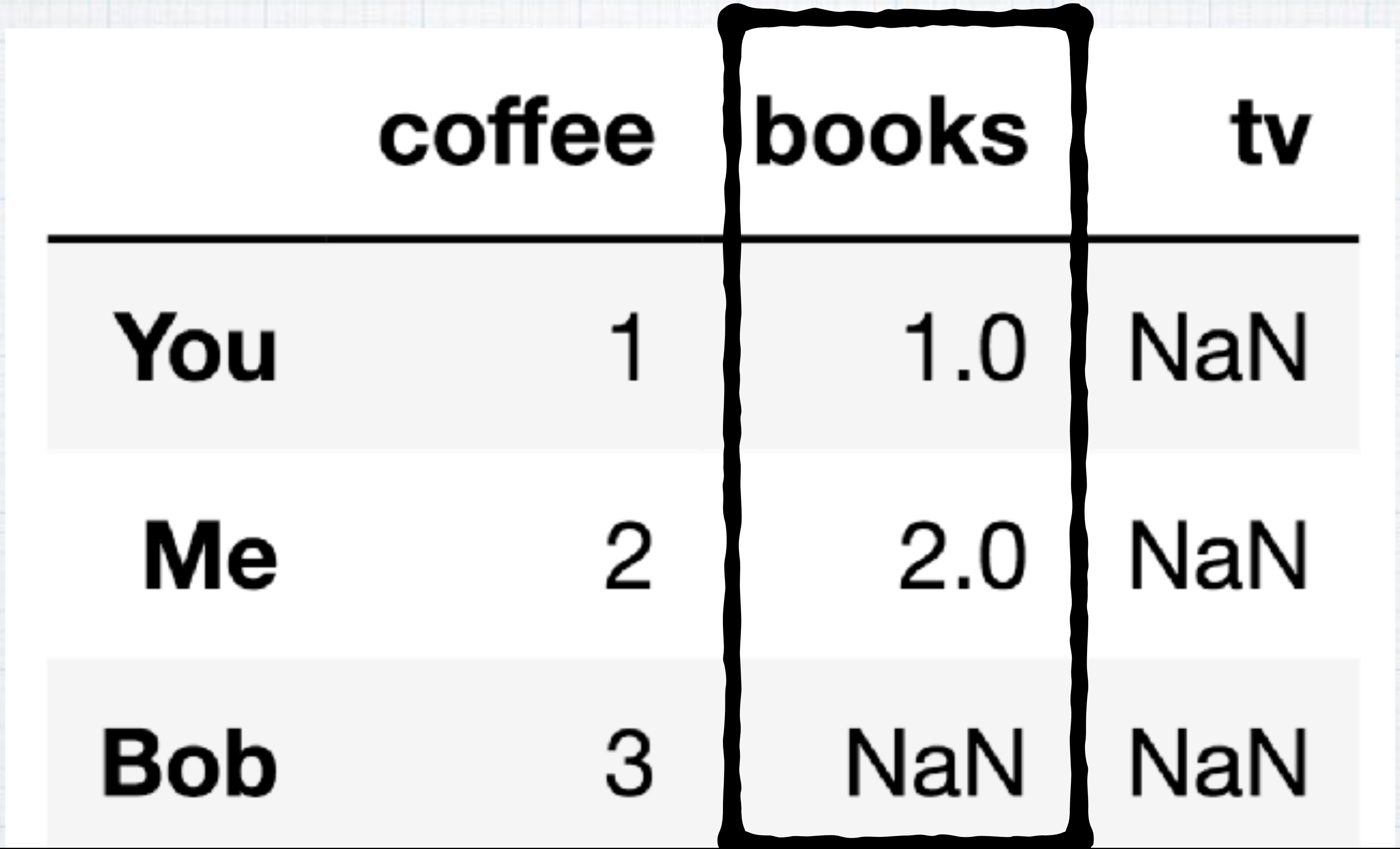
```
>>> d['coffee'].sum()
6
```

Pandas dataframe: `d`

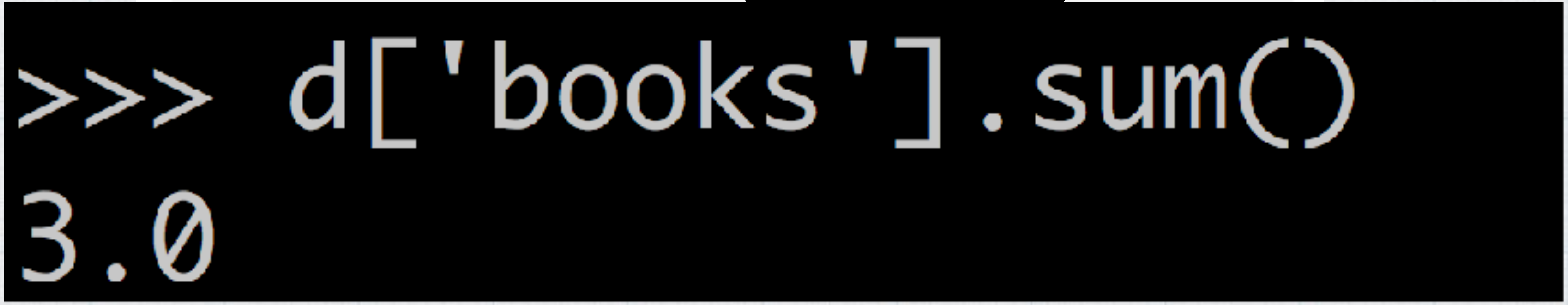


	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN

Pandas dataframe: `d`

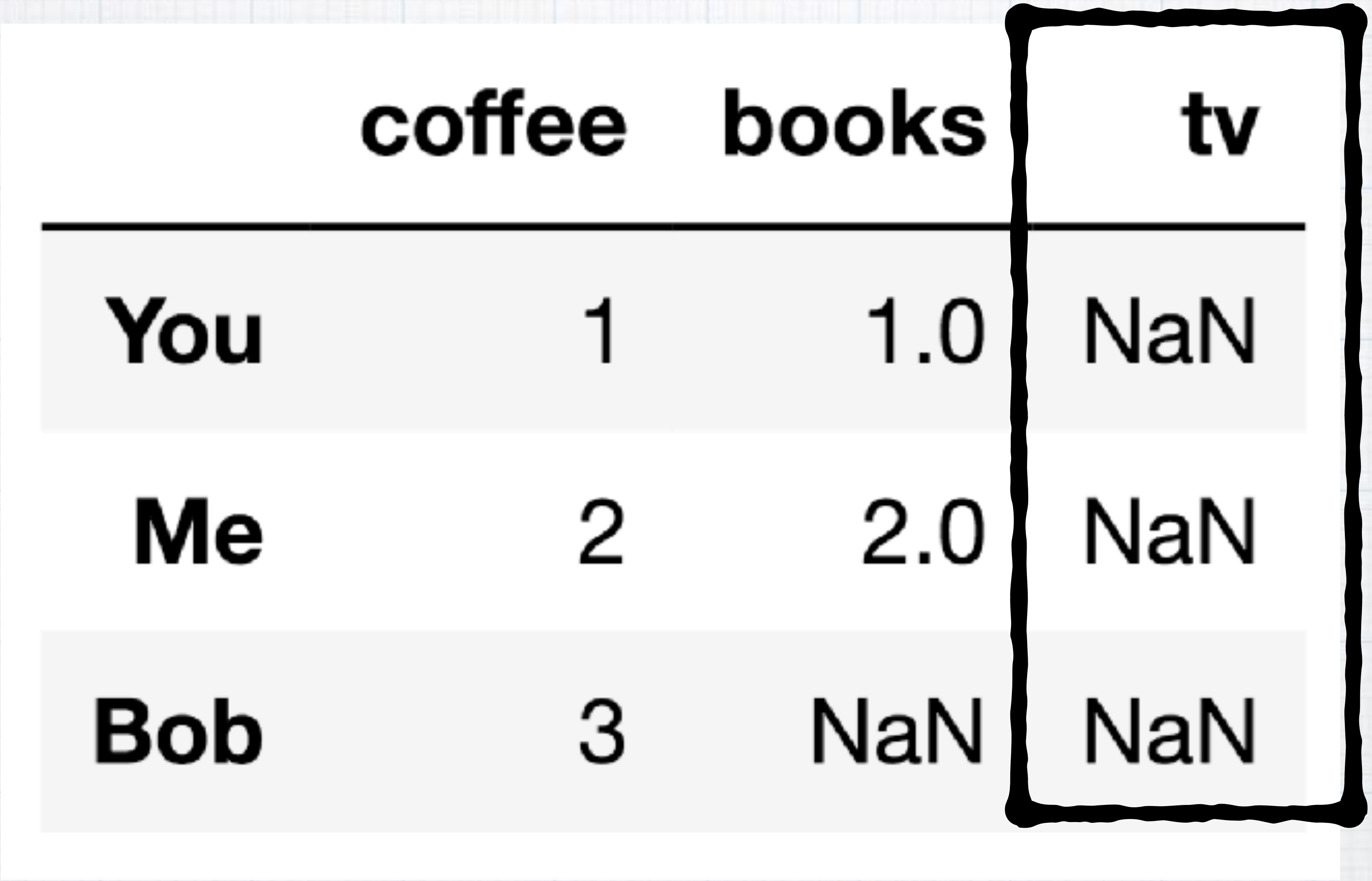


	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN



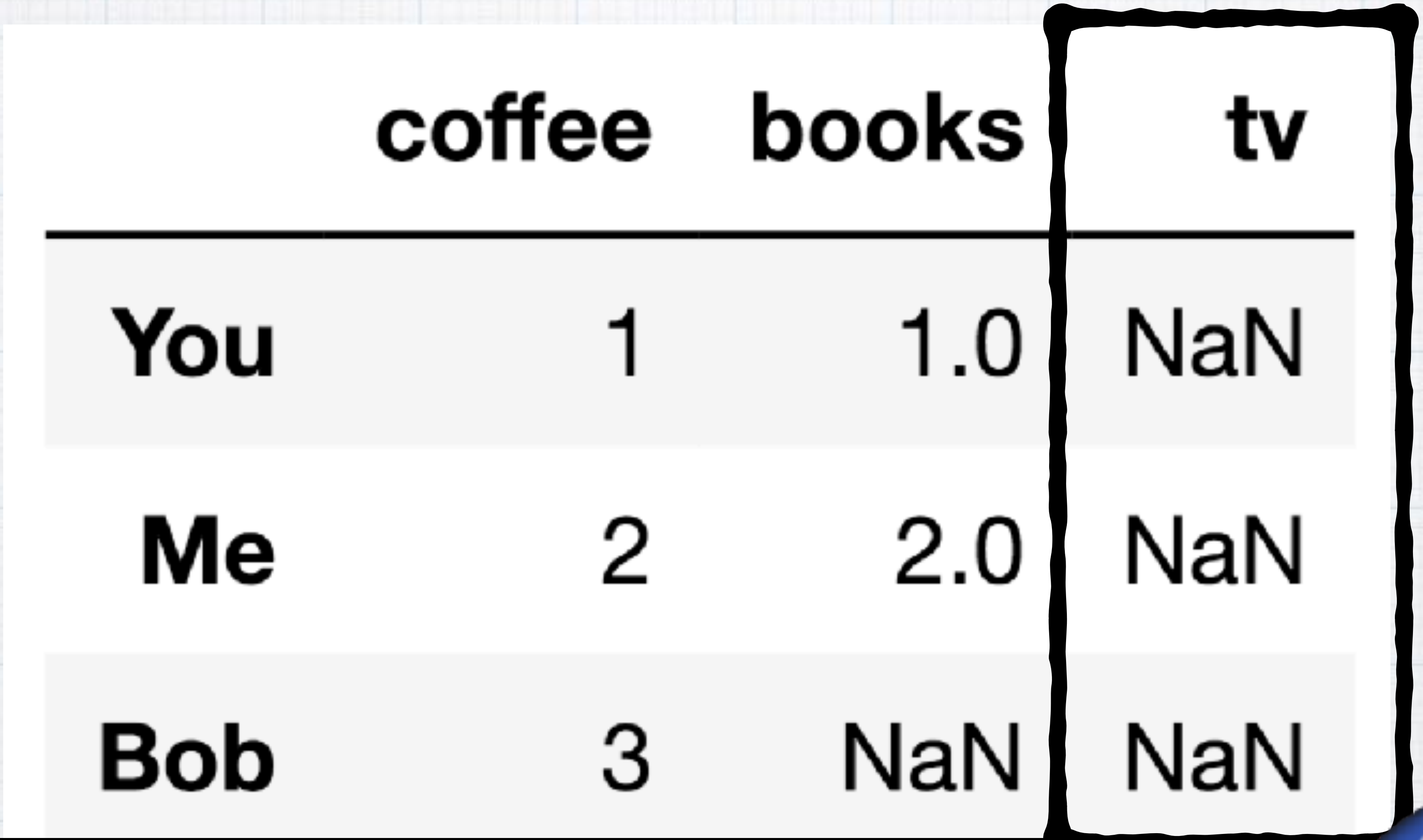
```
>>> d['books'].sum()
3.0
```

Pandas dataframe: `d`

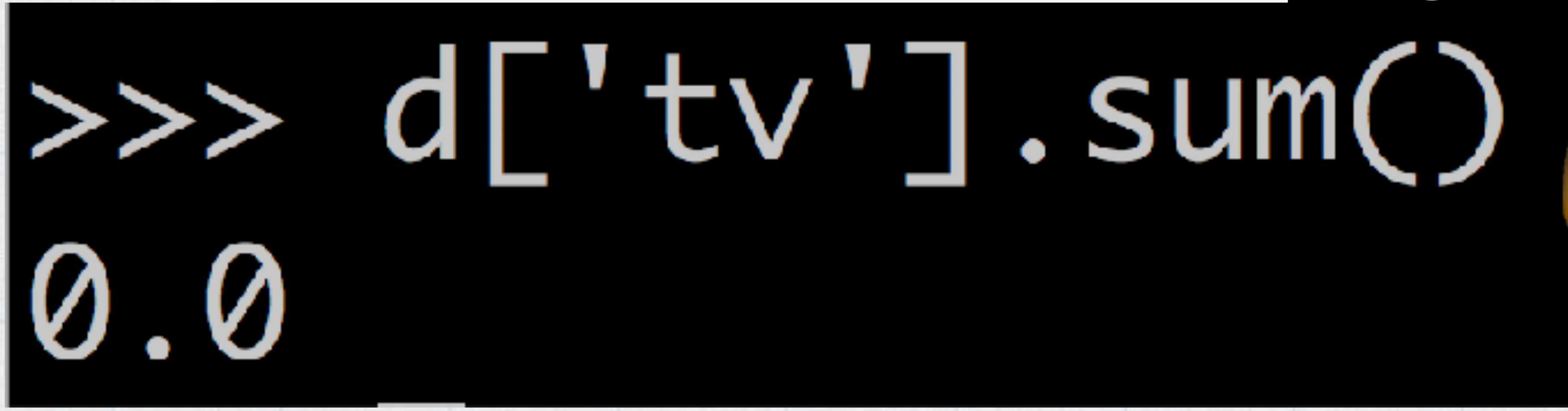


	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN

Pandas dataframe: `d`

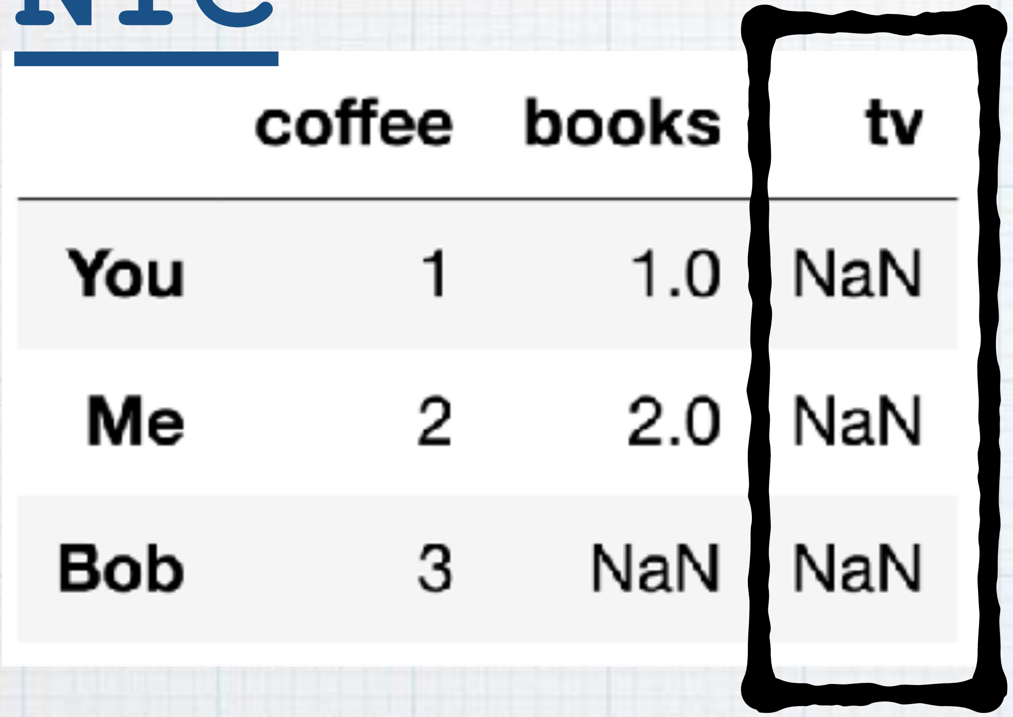


	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN



```
>>> d['tv'].sum()
0.0
```

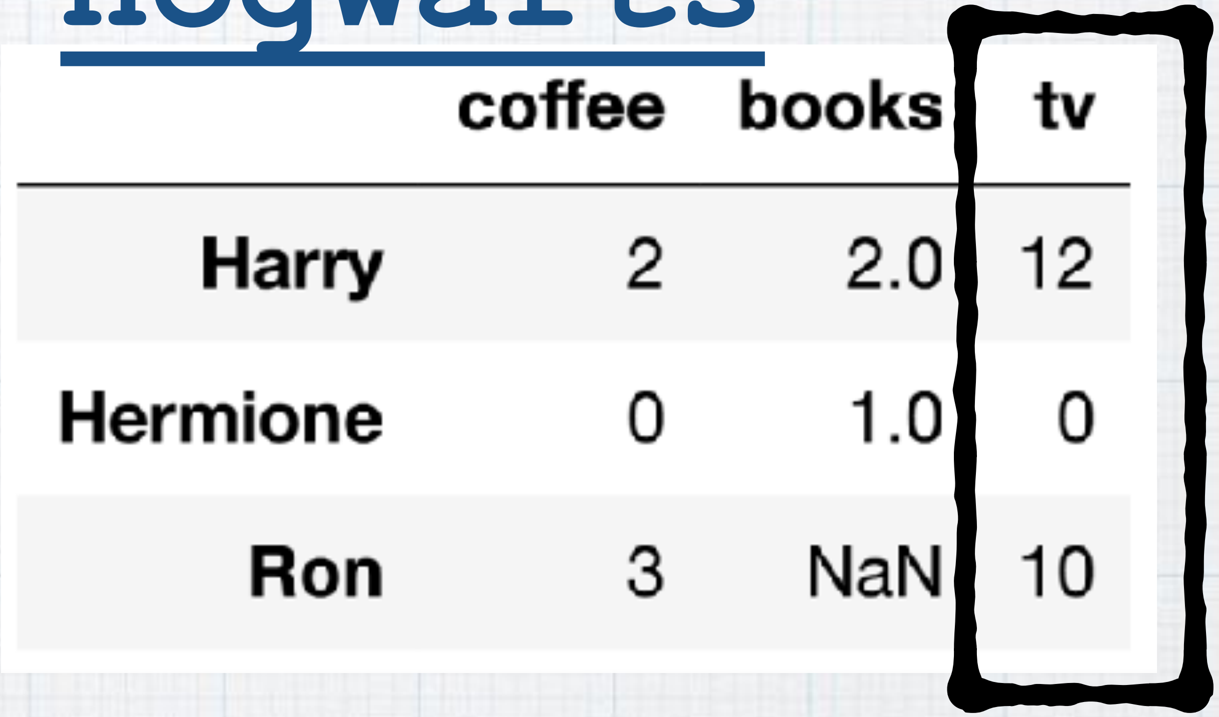
NYC



	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN

0

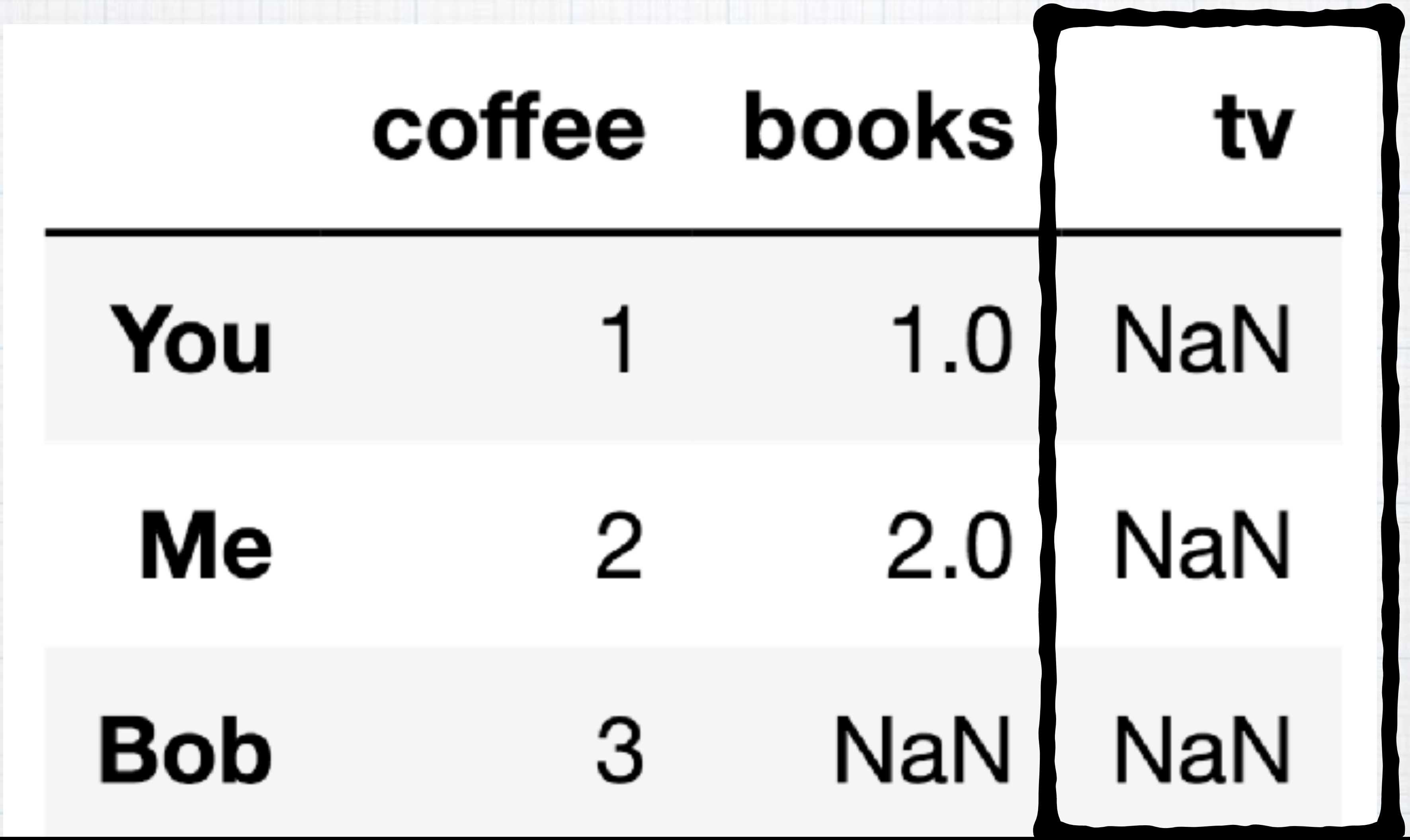
Hogwarts



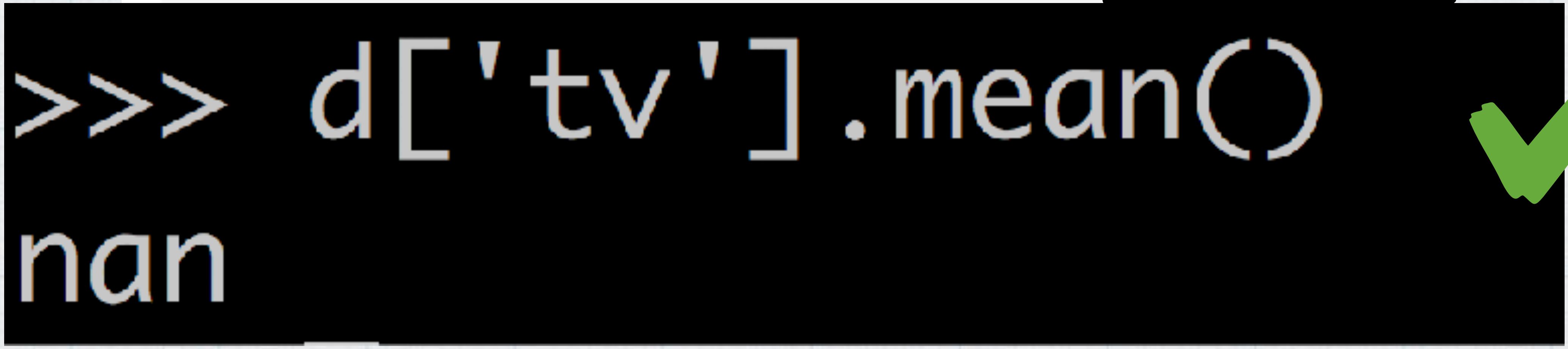
	coffee	books	tv
Harry	2	2.0	12
Hermione	0	1.0	0
Ron	3	NaN	10

22

Pandas dataframe: `d`



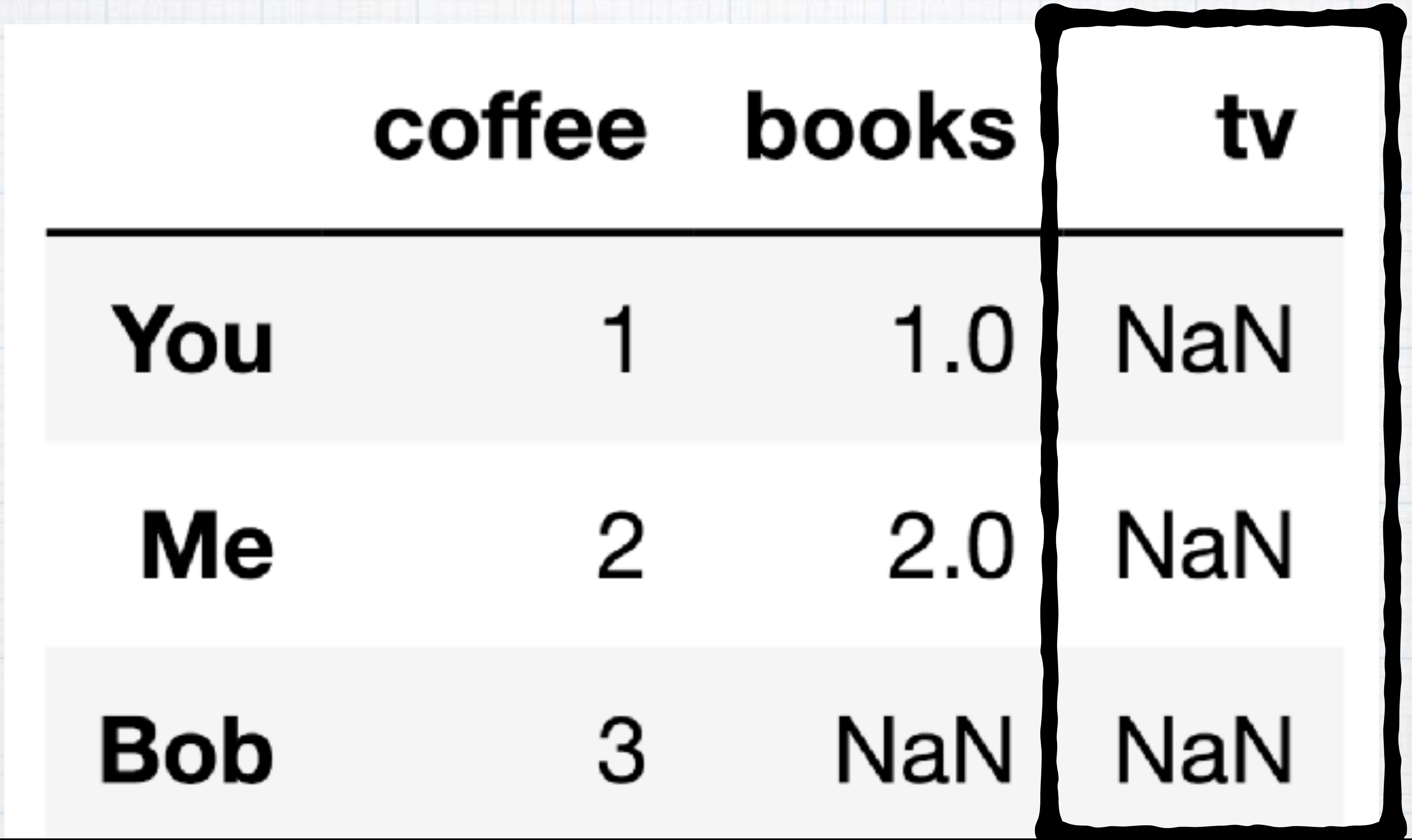
	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN



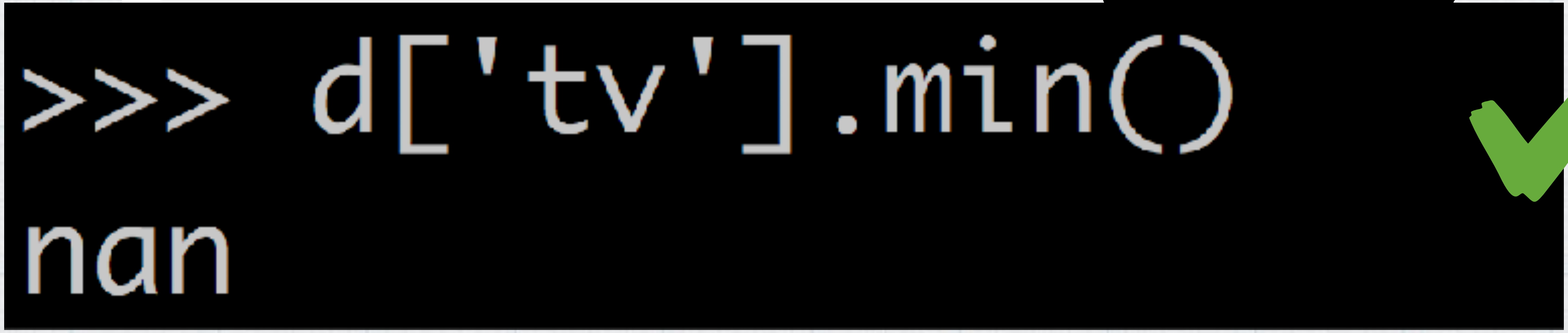
```
>>> d['tv'].mean()
nan
```

✓

Pandas dataframe: `d`



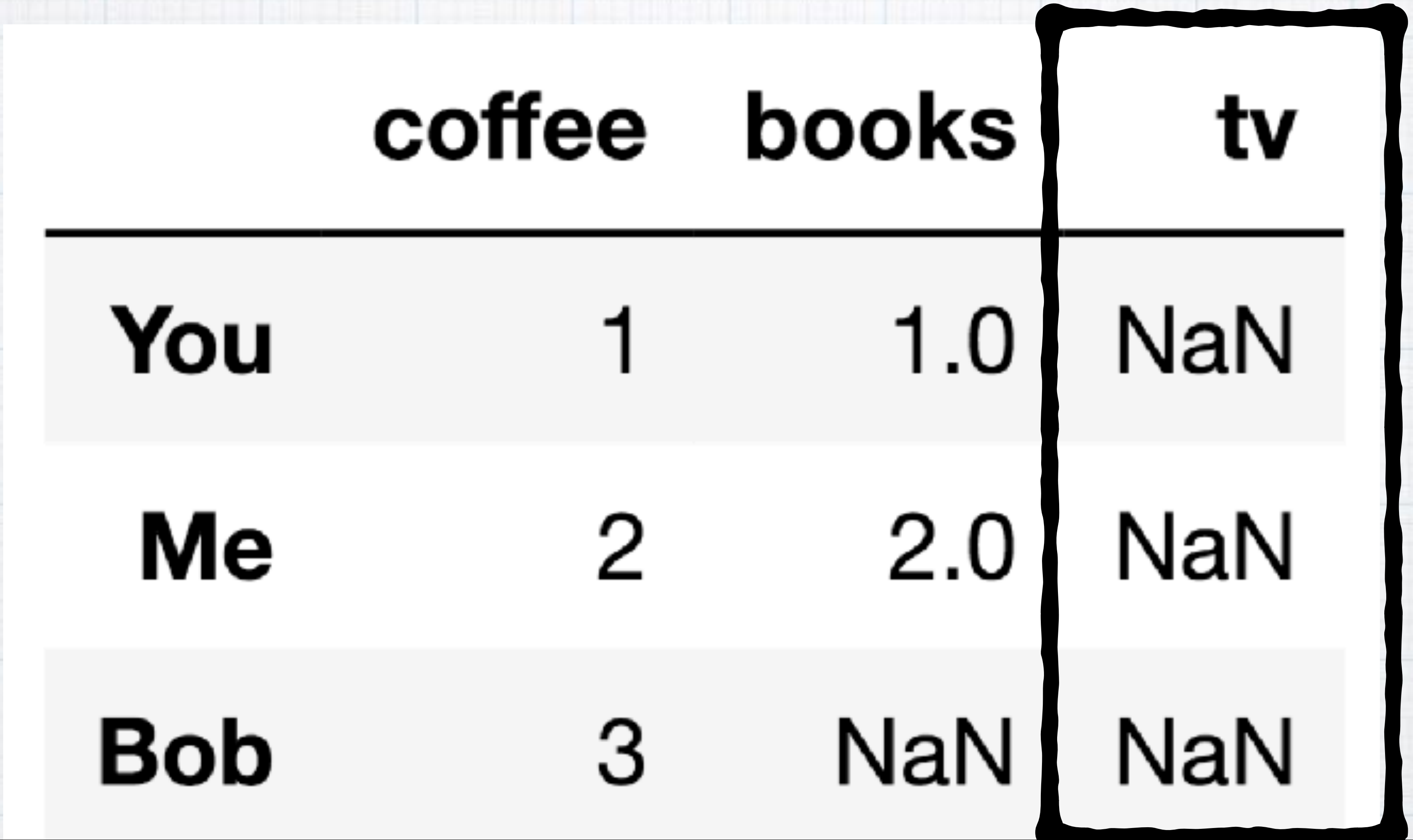
	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN



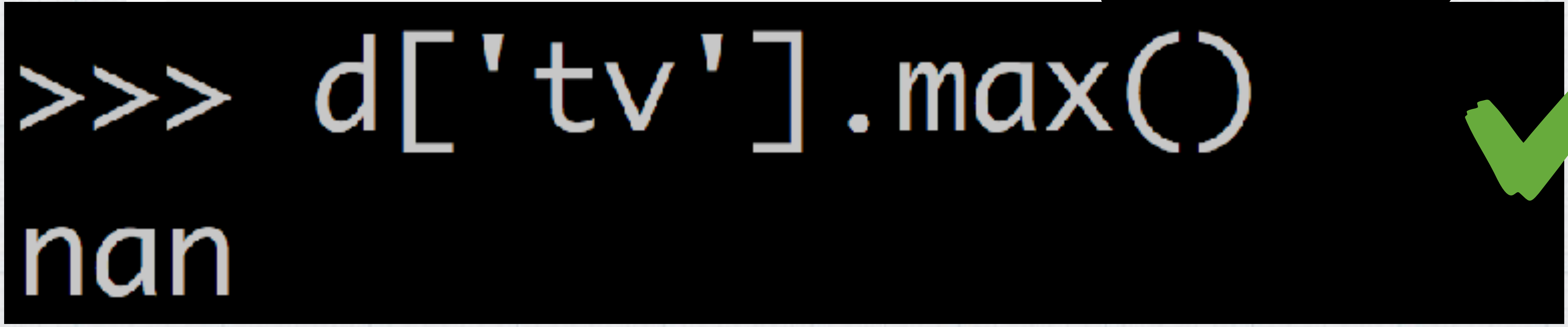
```
>>> d['tv'].min()
nan
```

✓

Pandas dataframe: `d`



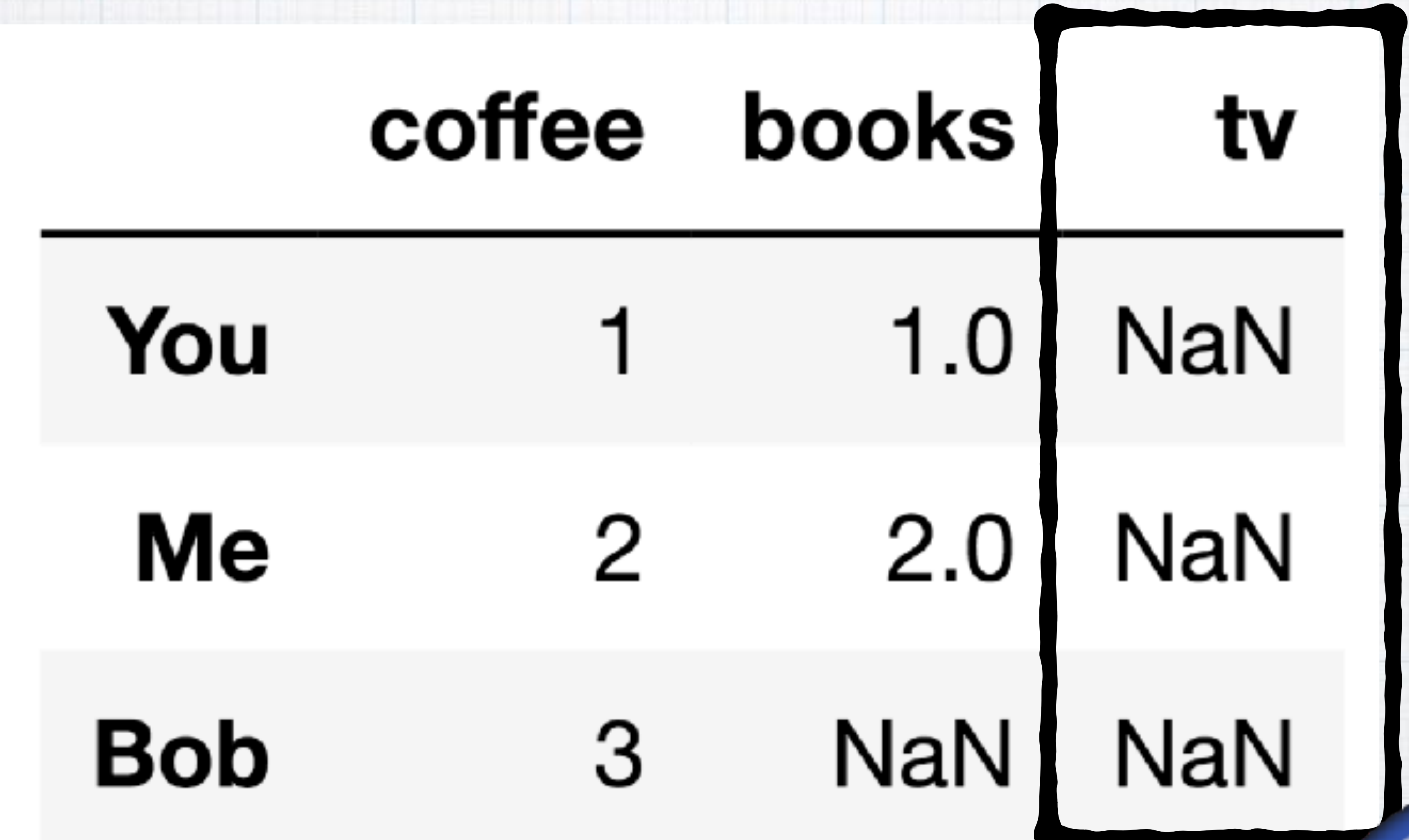
	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN



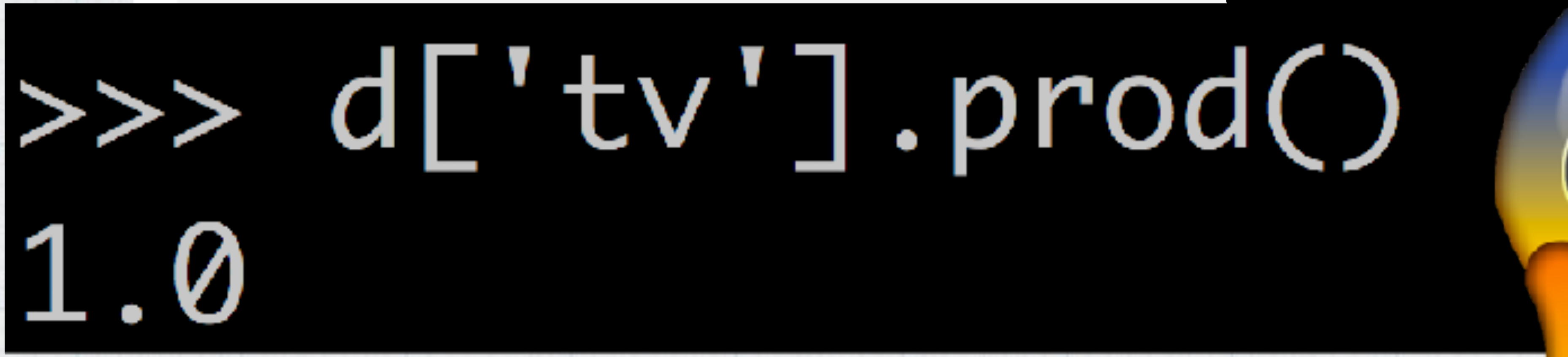
```
>>> d['tv'].max()
nan
```

✓

Pandas dataframe: `d`



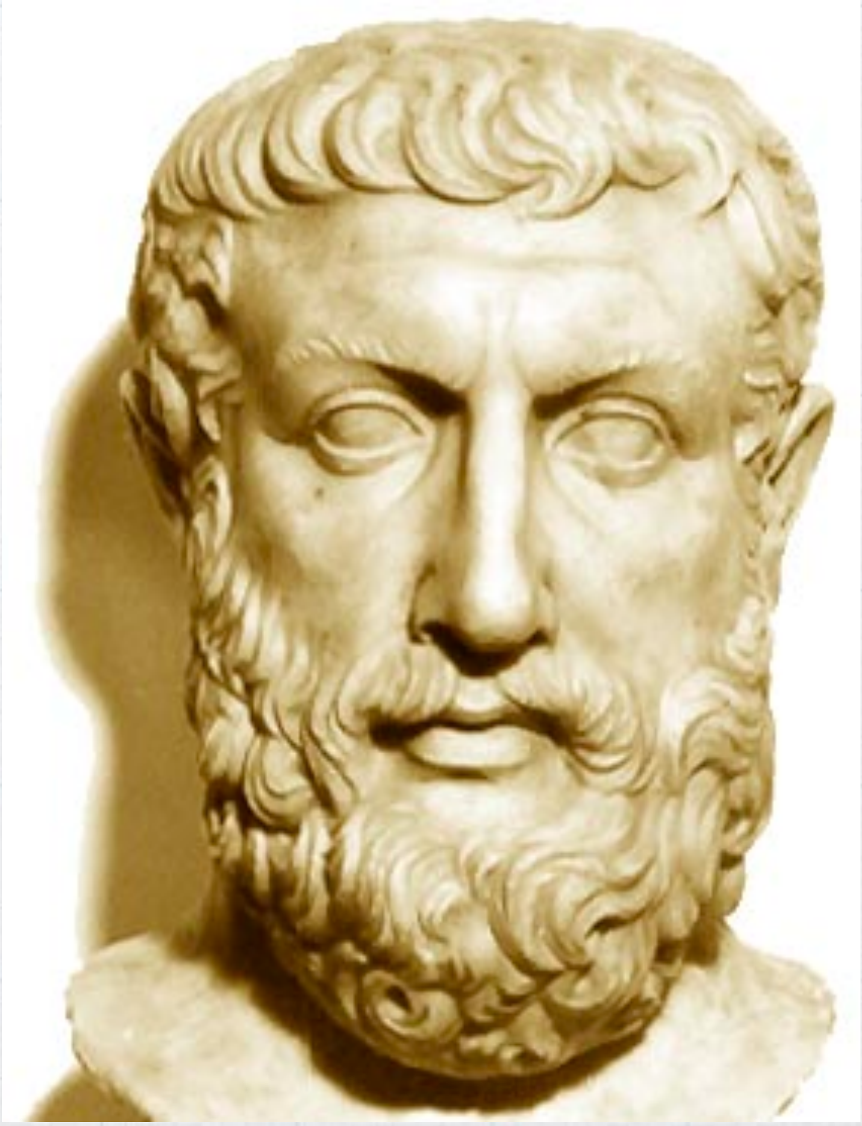
	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN

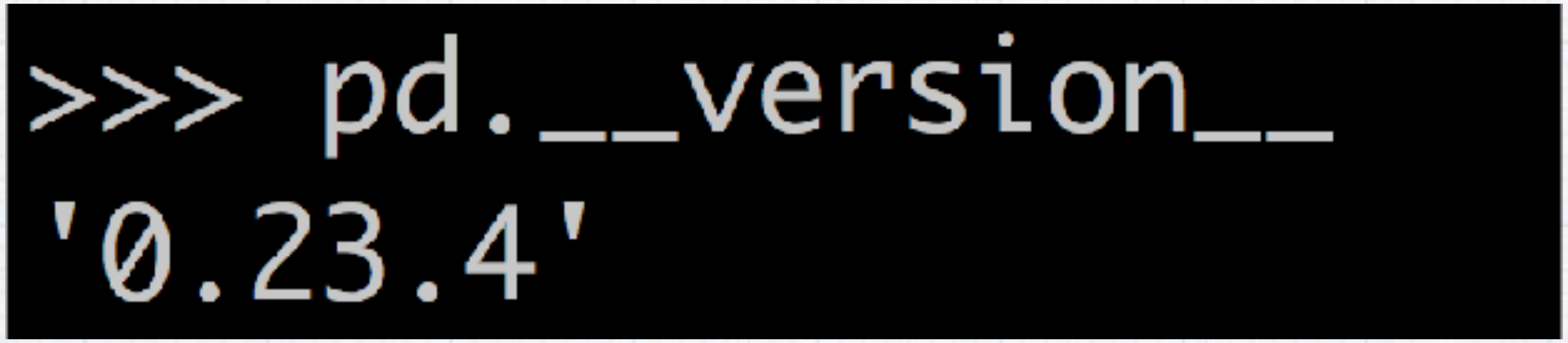


```
>>> d['tv'].prod()
1.0
```

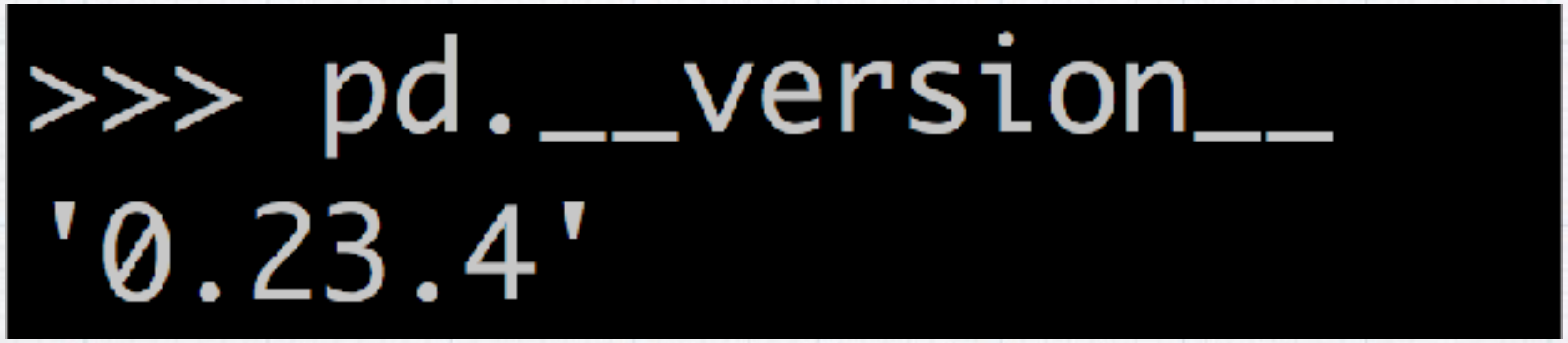
Something cannot come from nothing

—Parmenides (b. 515 BC)



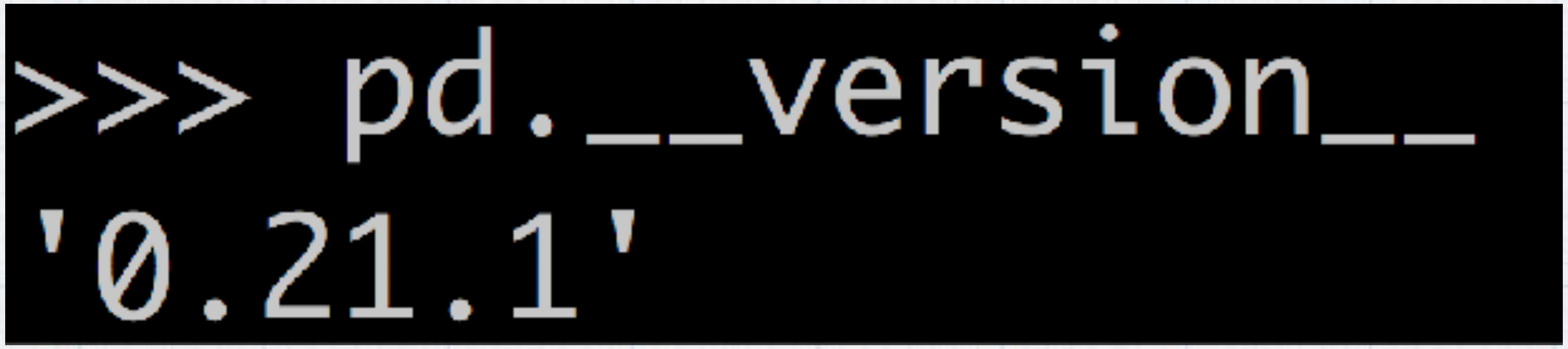


```
>>> pd.__version__
'0.23.4'
```

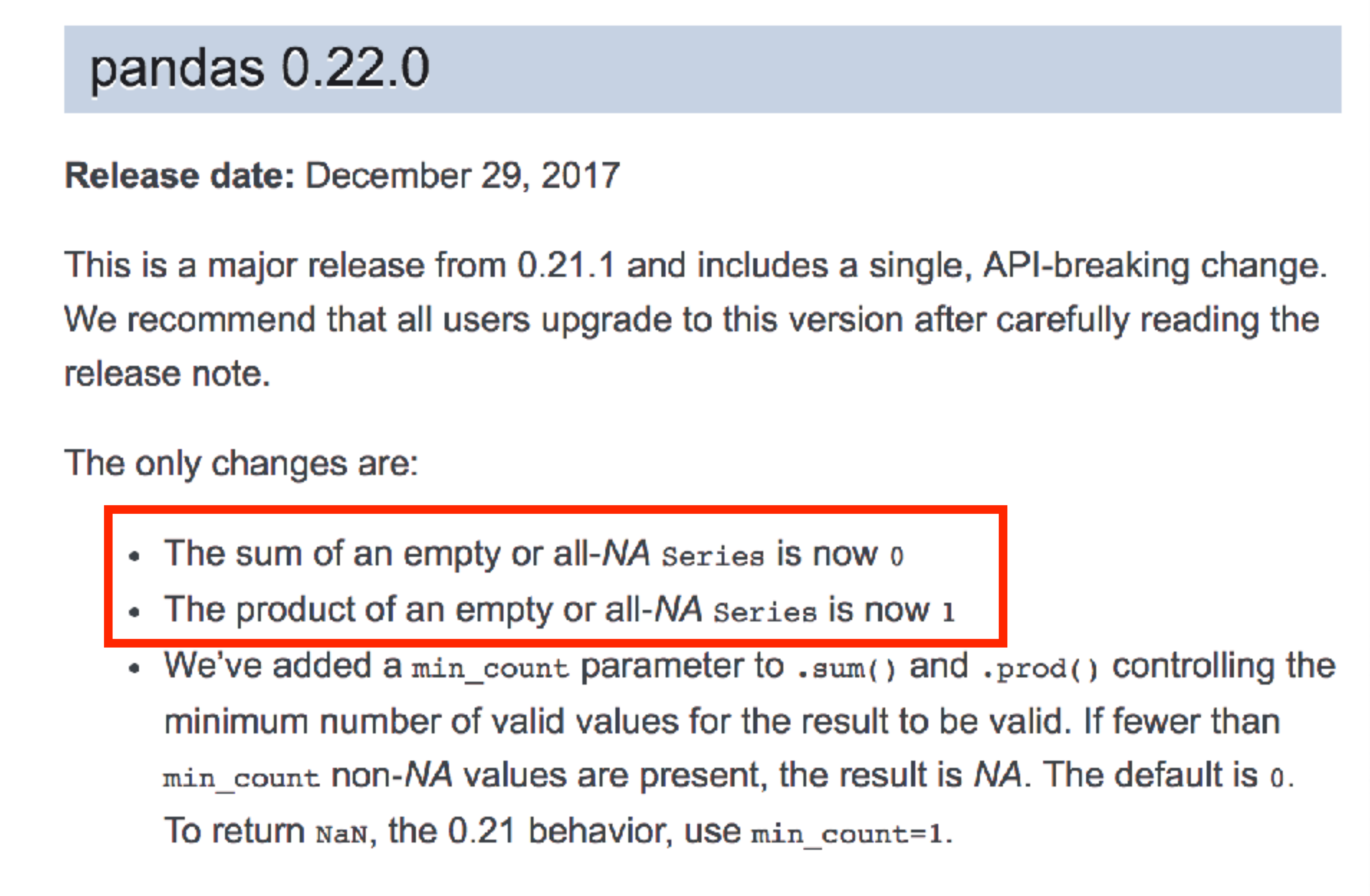


```
>>> pd.__version__
'0.23.4'
```

vs.



```
>>> pd.__version__
'0.21.1'
```



pandas 0.22.0

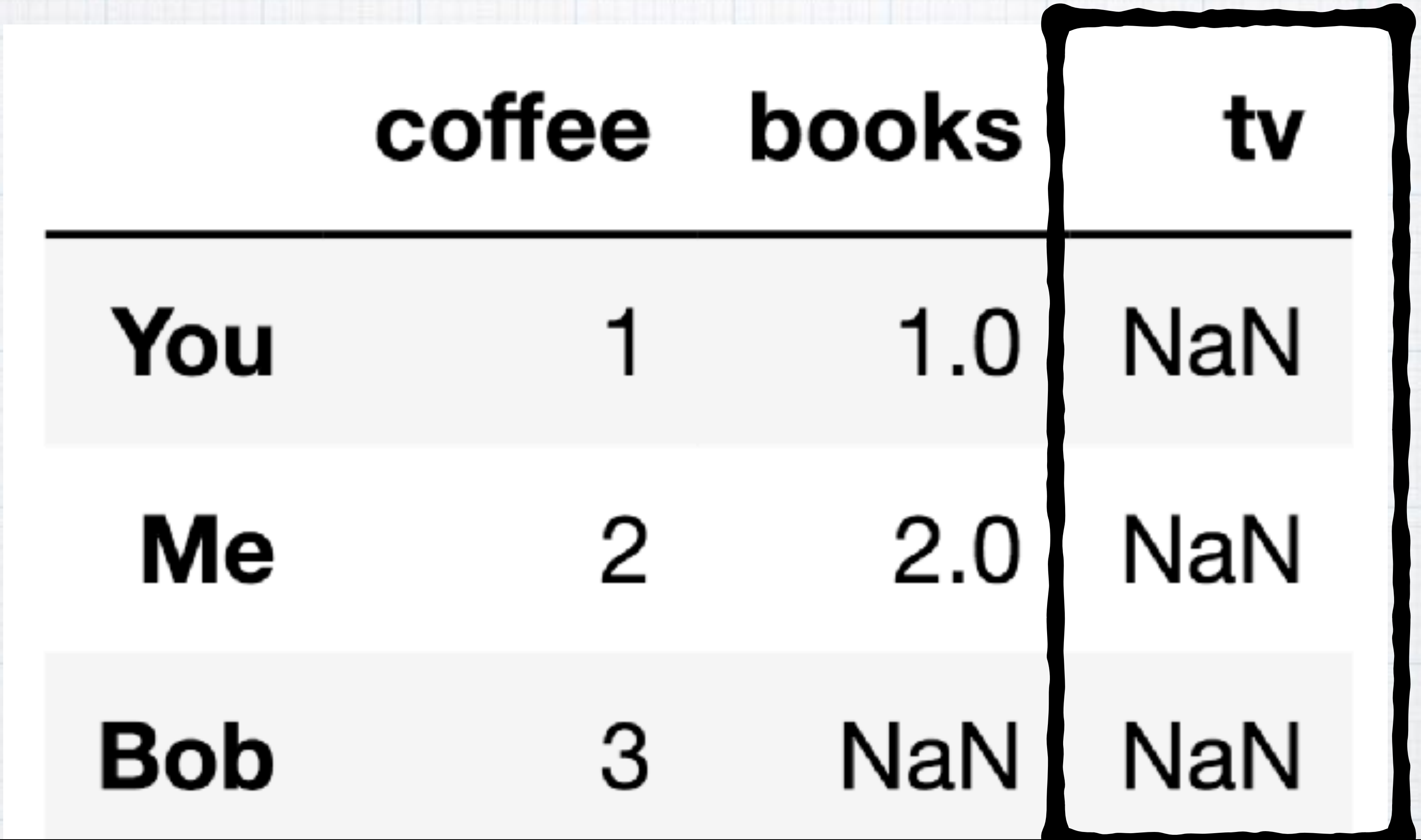
Release date: December 29, 2017

This is a major release from 0.21.1 and includes a single, API-breaking change. We recommend that all users upgrade to this version after carefully reading the release note.

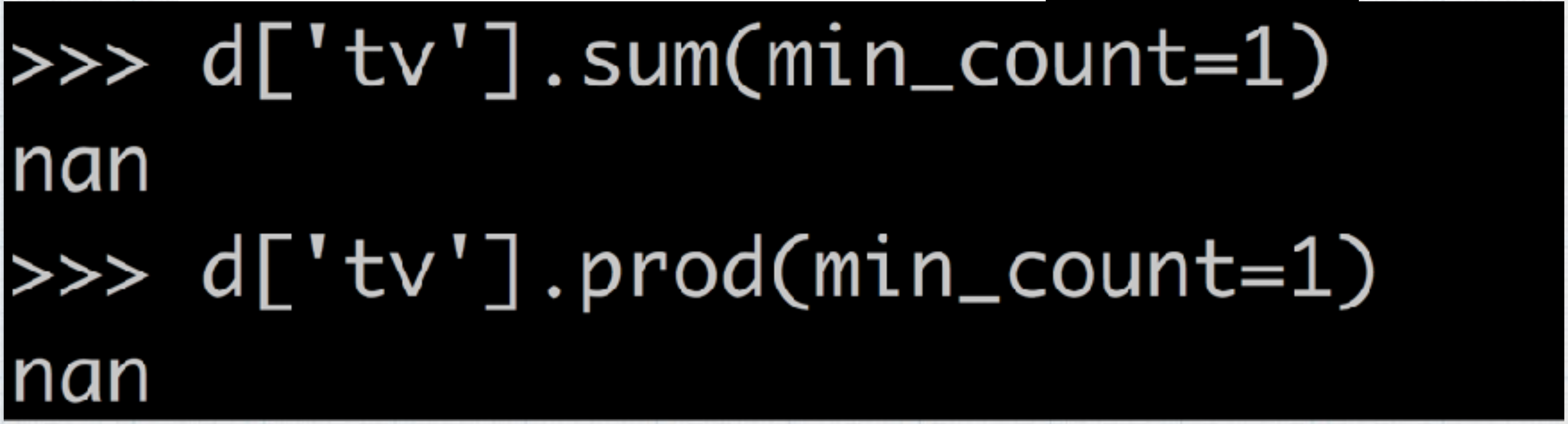
The only changes are:

- The sum of an empty or all-*NA* `Series` is now `0`
- The product of an empty or all-*NA* `Series` is now `1`
- We've added a `min_count` parameter to `.sum()` and `.prod()` controlling the minimum number of valid values for the result to be valid. If fewer than `min_count` non-*NA* values are present, the result is *NA*. The default is `0`. To return `NaN`, the 0.21 behavior, use `min_count=1`.

Pandas dataframe: `d`



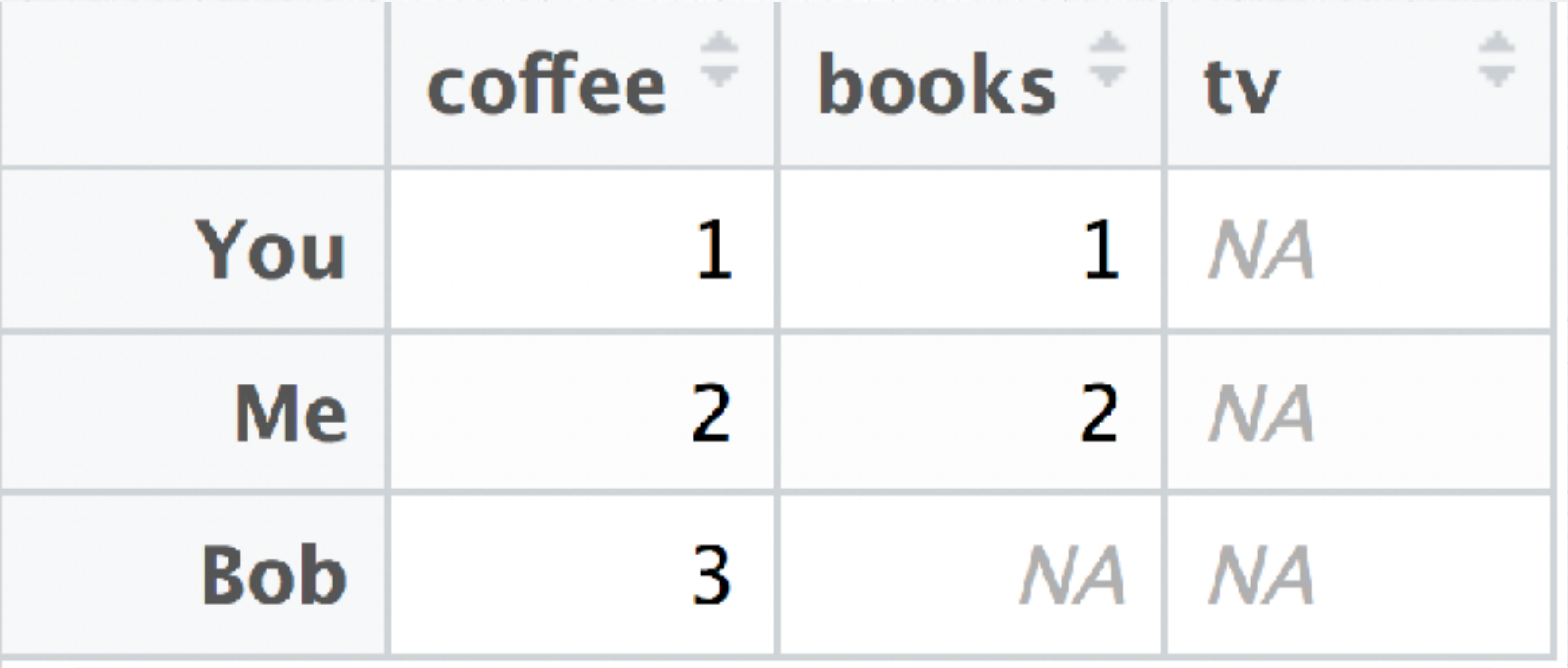
	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN



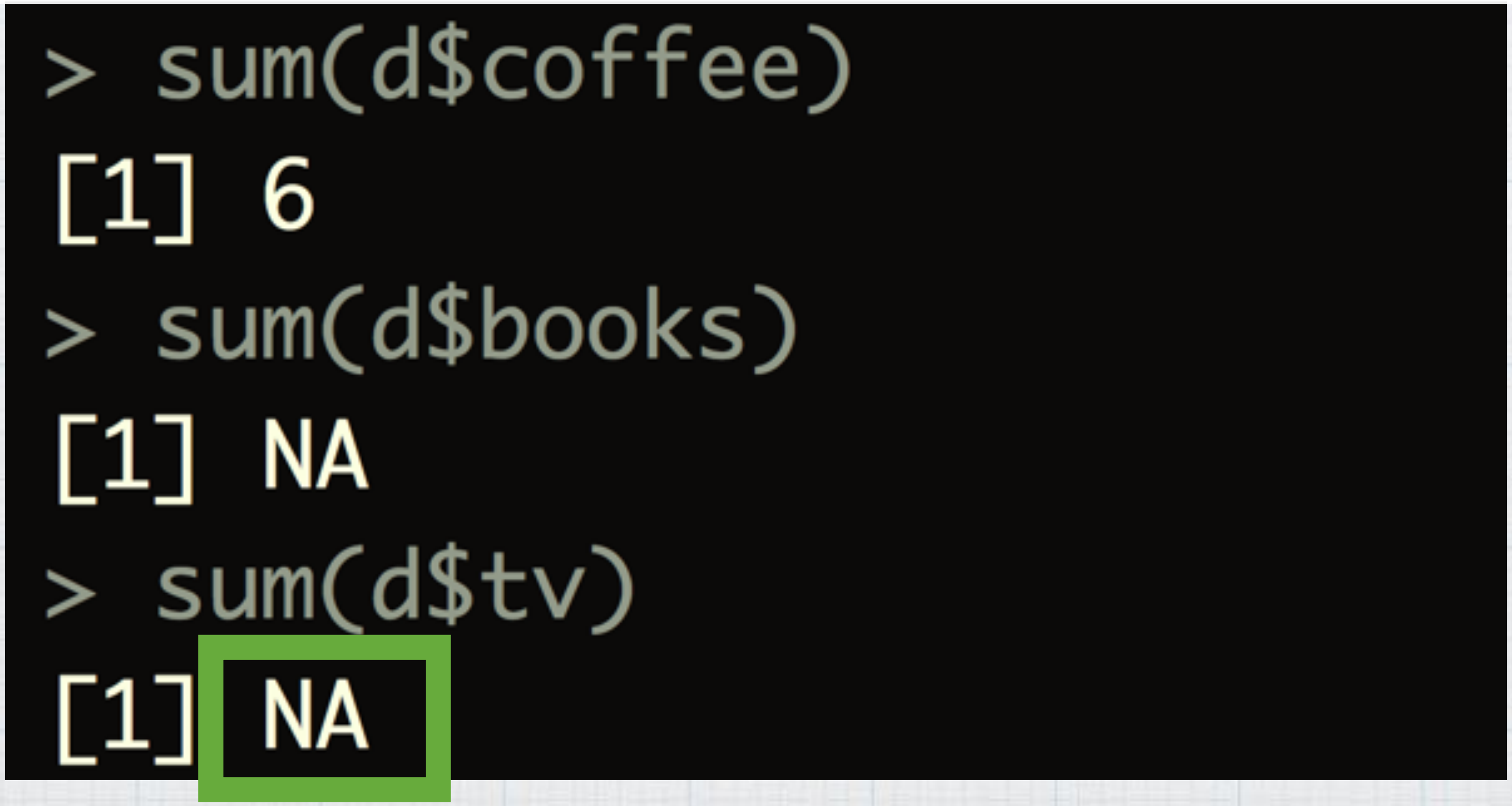
```
>>> d['tv'].sum(min_count=1)
nan
>>> d['tv'].prod(min_count=1)
nan
```

BUT WHY?

R dataframe: `d`

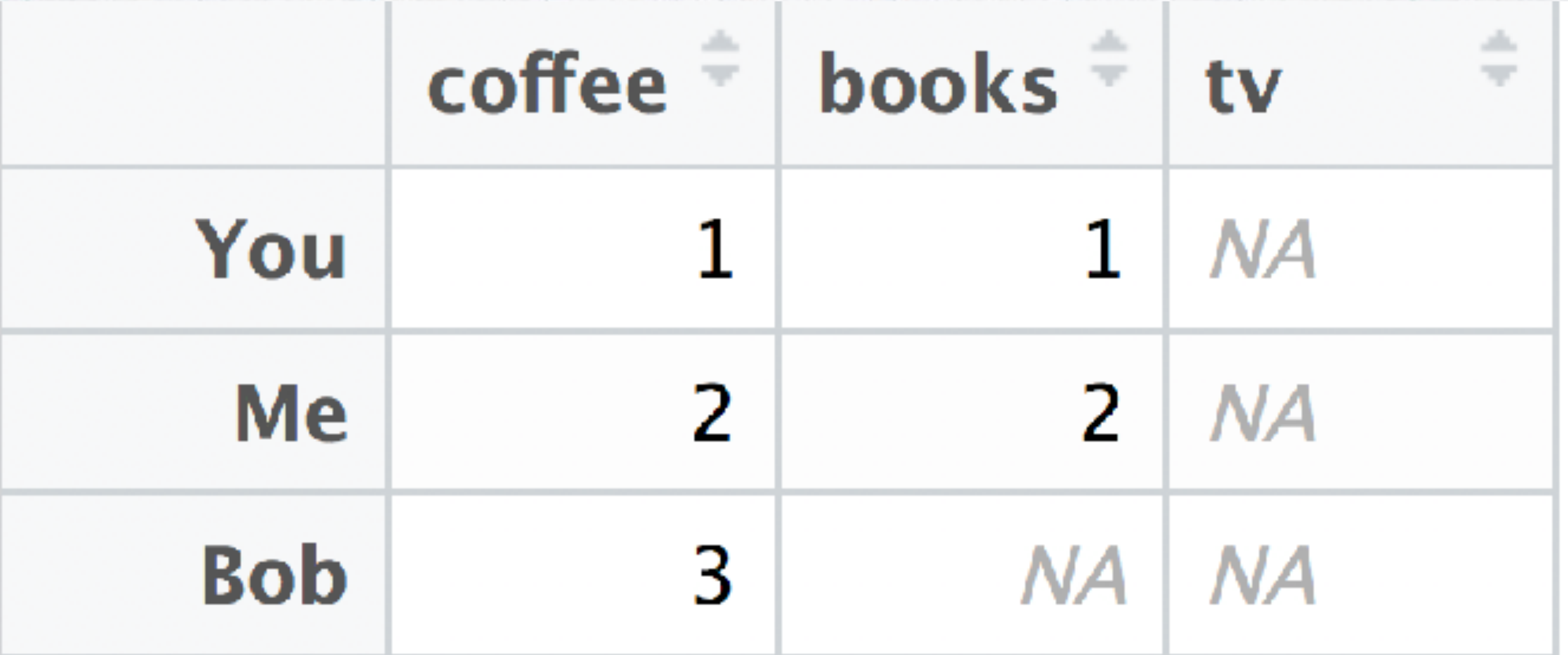


	coffee	books	tv
You	1	1	*NA*
Me	2	2	*NA*
Bob	3	*NA*	*NA*

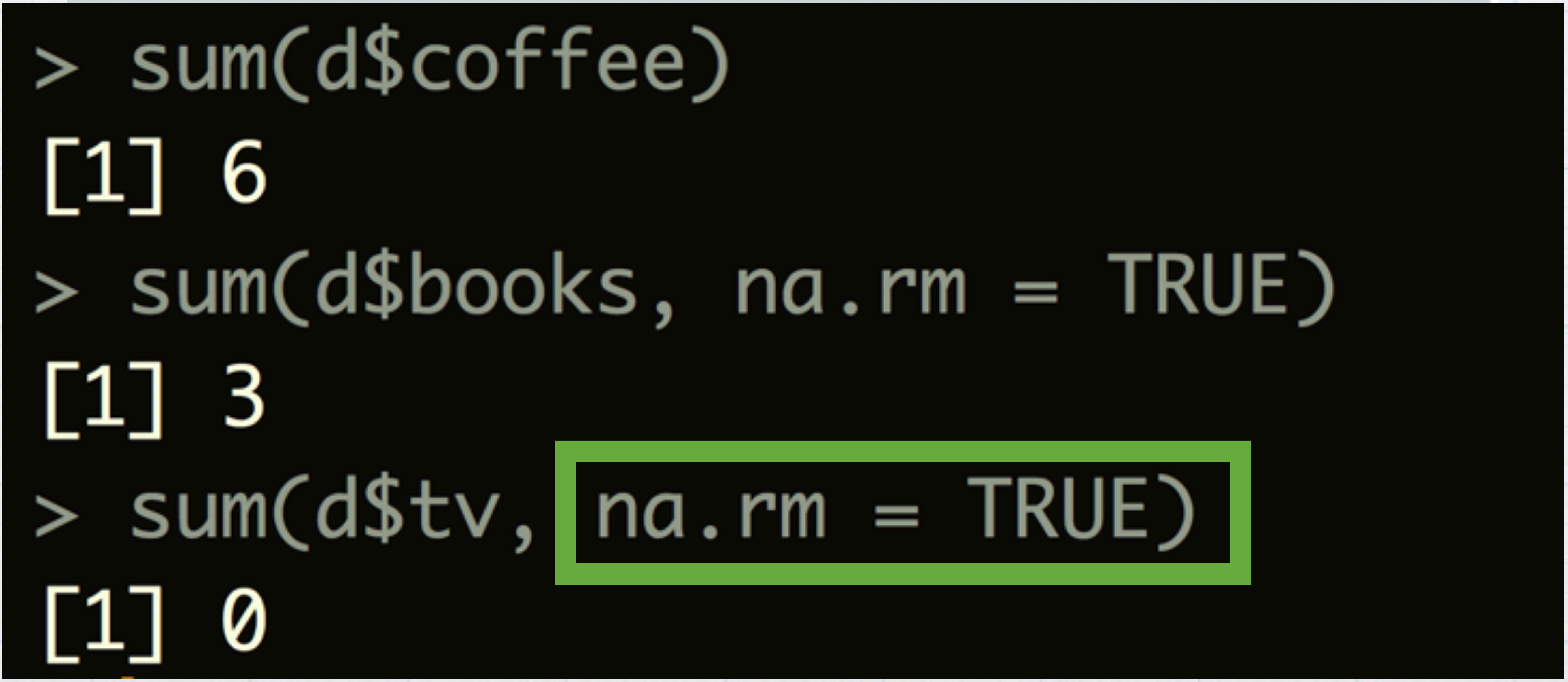


```
> sum(d$coffee)
[1] 6
> sum(d$books)
[1] NA
> sum(d$tv)
[1] NA
```

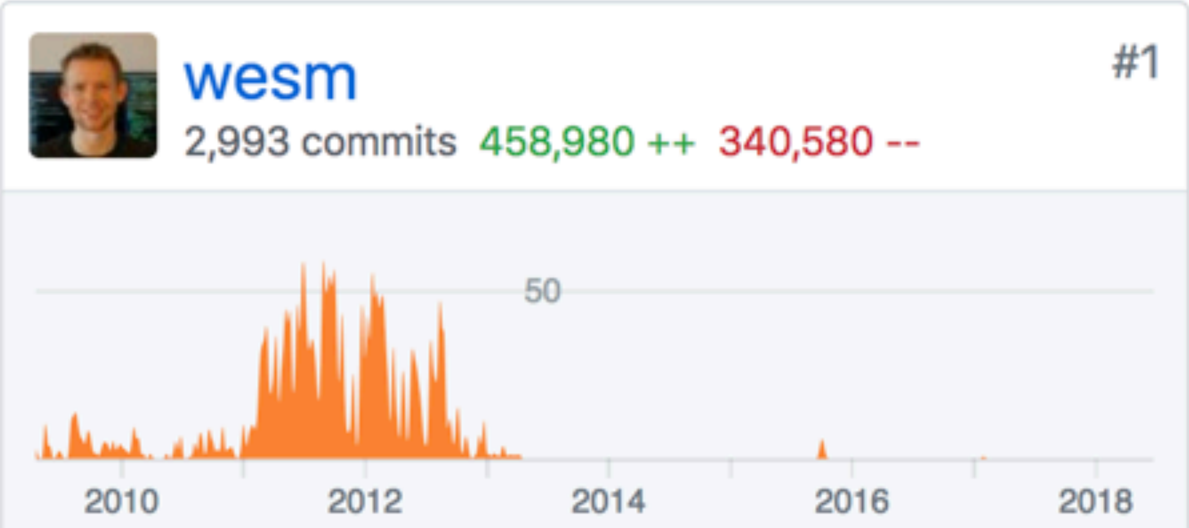
R dataframe: `d`



	coffee	books	tv
You	1	1	NA
Me	2	2	NA
Bob	3	NA	NA

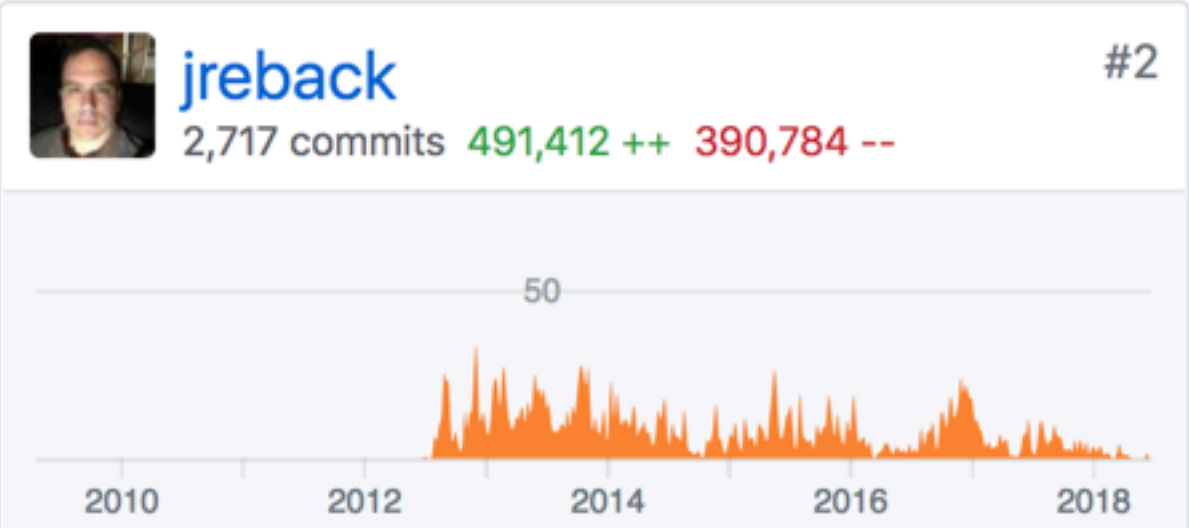


```
> sum(d$coffee)
[1] 6
> sum(d$books, na.rm = TRUE)
[1] 3
> sum(d$tv, na.rm = TRUE)
[1] 0
```



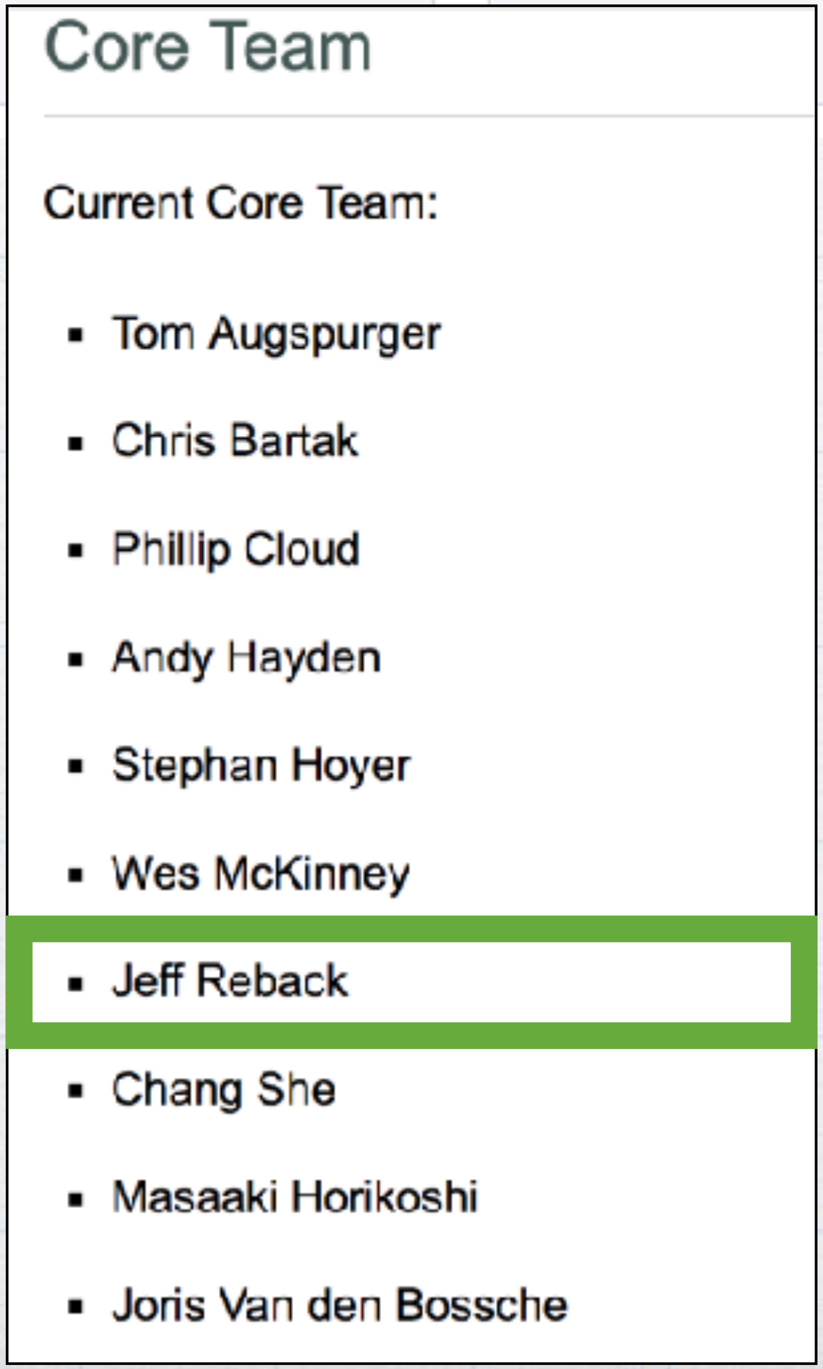
wesm #1

2,993 commits 458,980 ++ 340,580 --



jreback #2

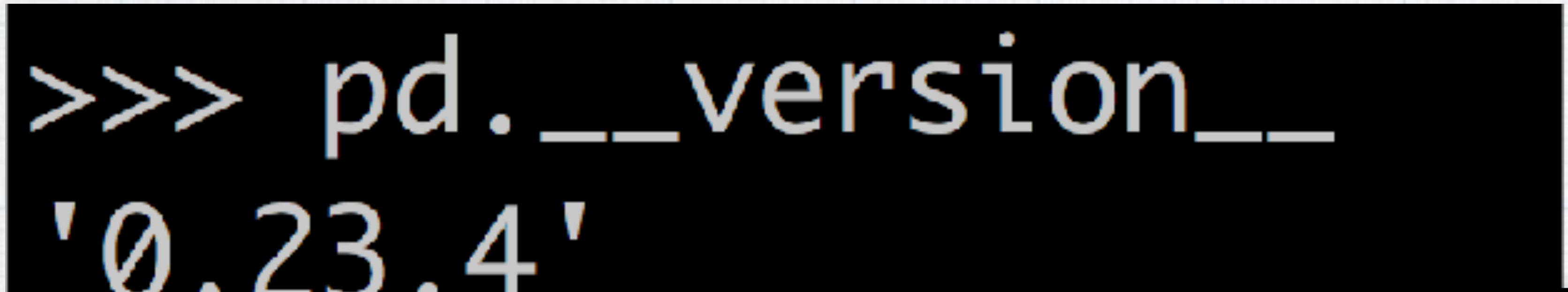
2,717 commits 491,412 ++ 390,784 --



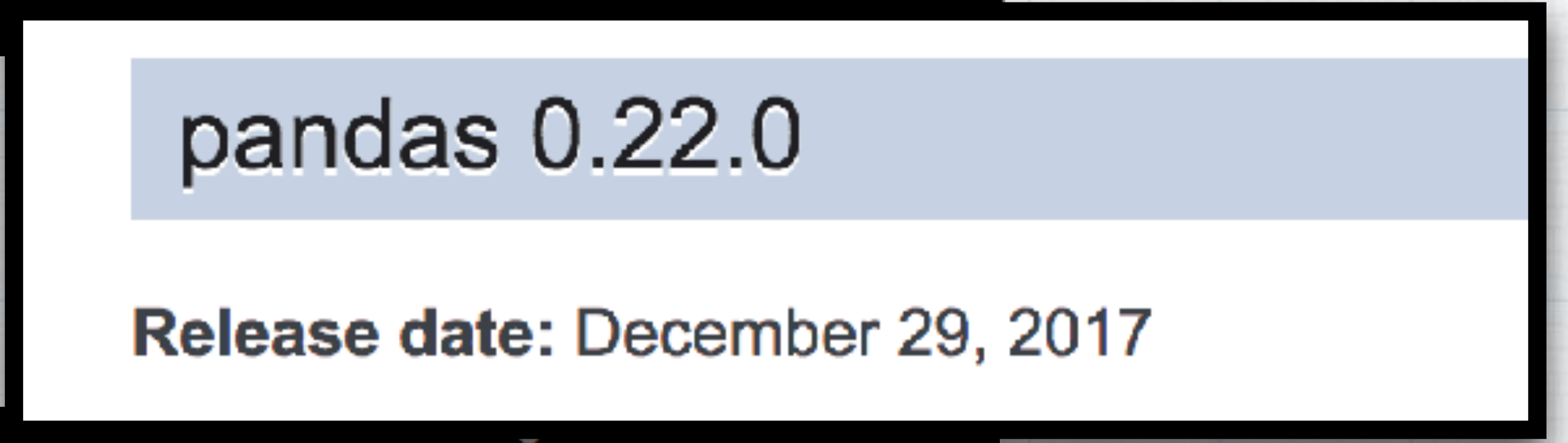
Core Team

Current Core Team:

- Tom Augspurger
- Chris Bartak
- Phillip Cloud
- Andy Hayden
- Stephan Hoyer
- Wes McKinney
- Jeff Reback
- Chang She
- Masaaki Horikoshi
- Joris Van den Bossche

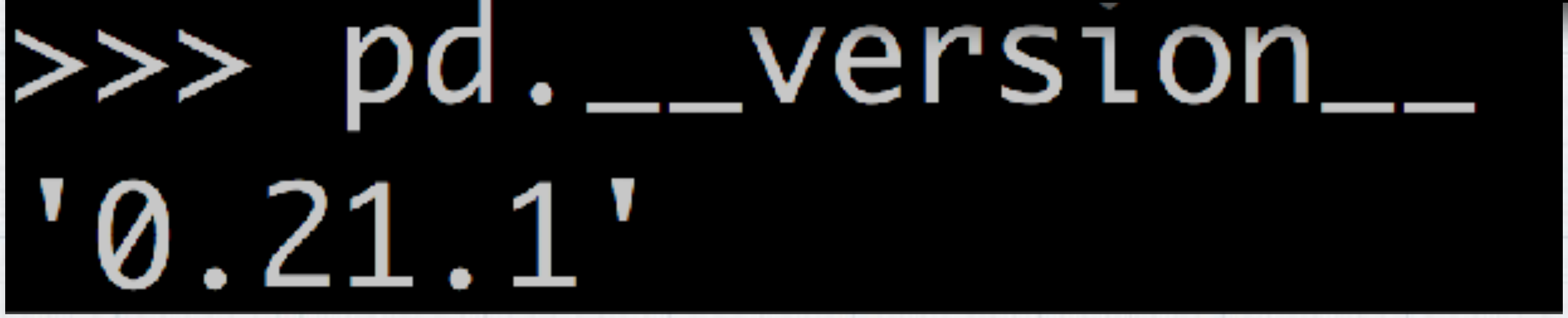


```
>>> pd.__version__
'0.23.4'
```

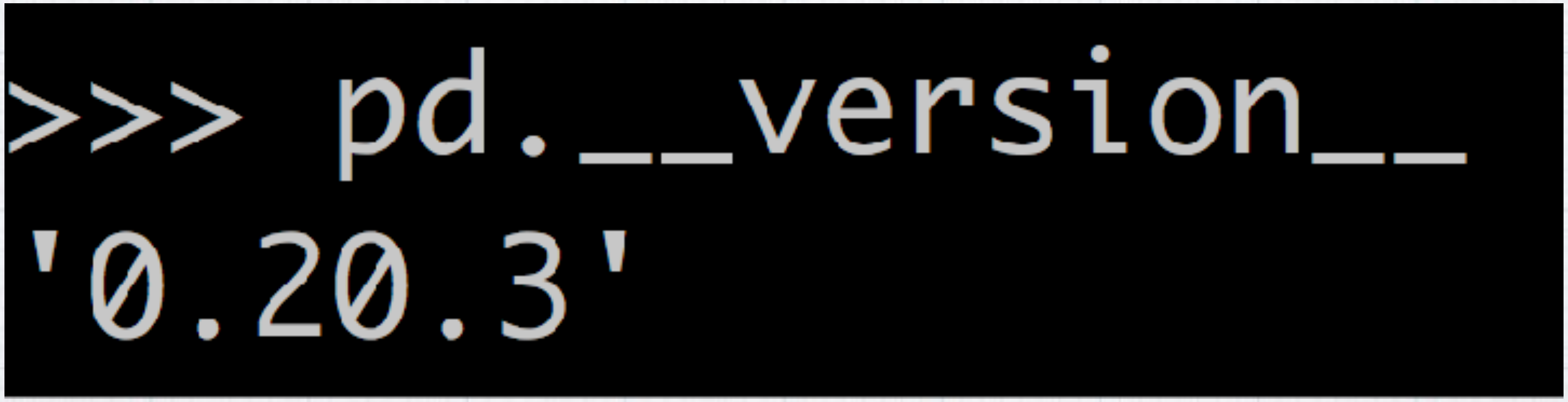


pandas 0.22.0

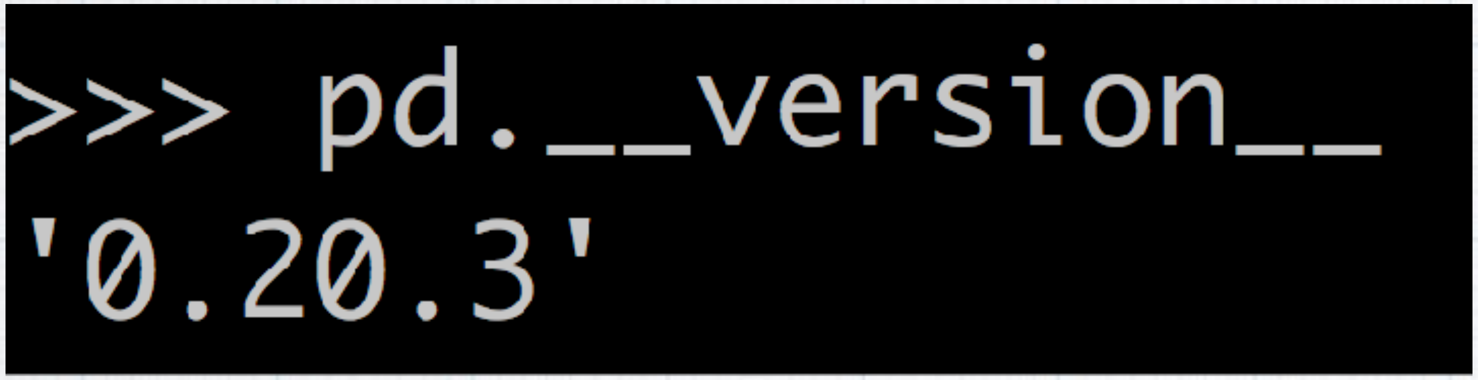
Release date: December 29, 2017



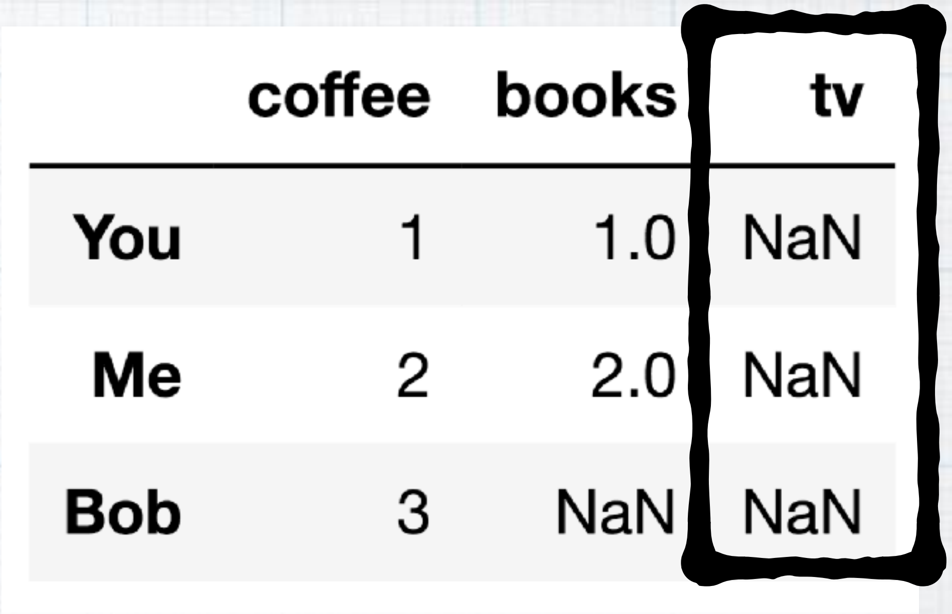
```
>>> pd.__version__
'0.21.1'
```



```
>>> pd.__version__
'0.20.3'
```

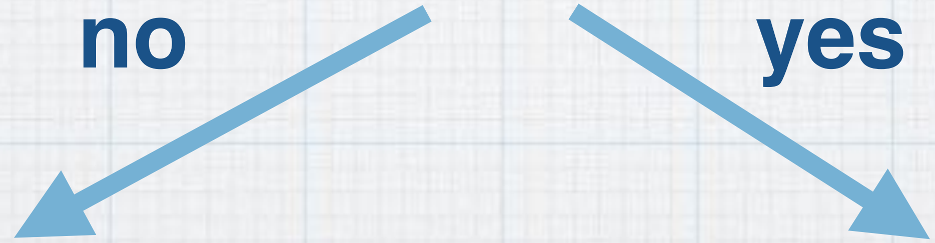


```
>>> pd.__version__
'0.20.3'
```



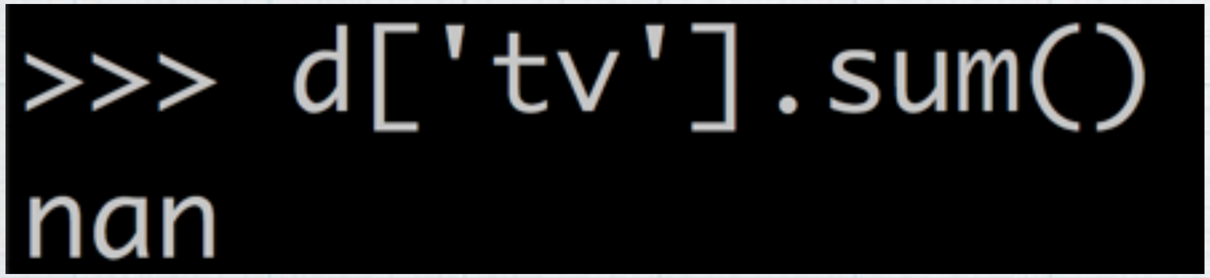
	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN

is `bottleneck`* installed?

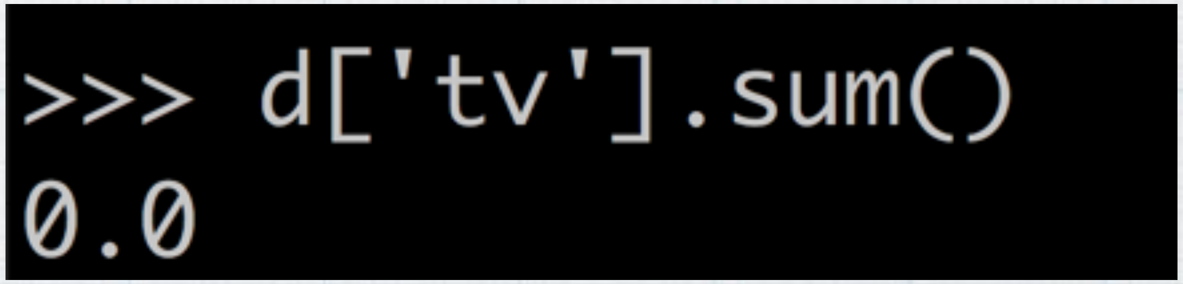


no

yes

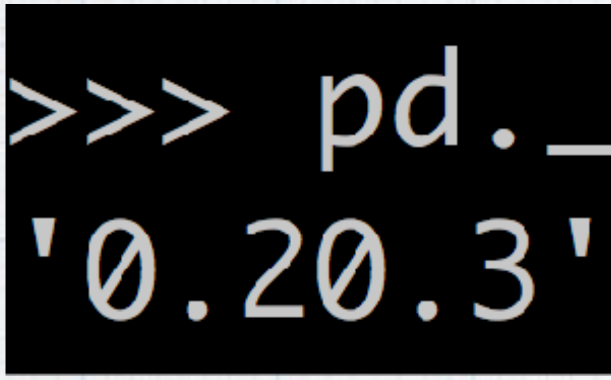


```
>>> d['tv'].sum()
nan
```

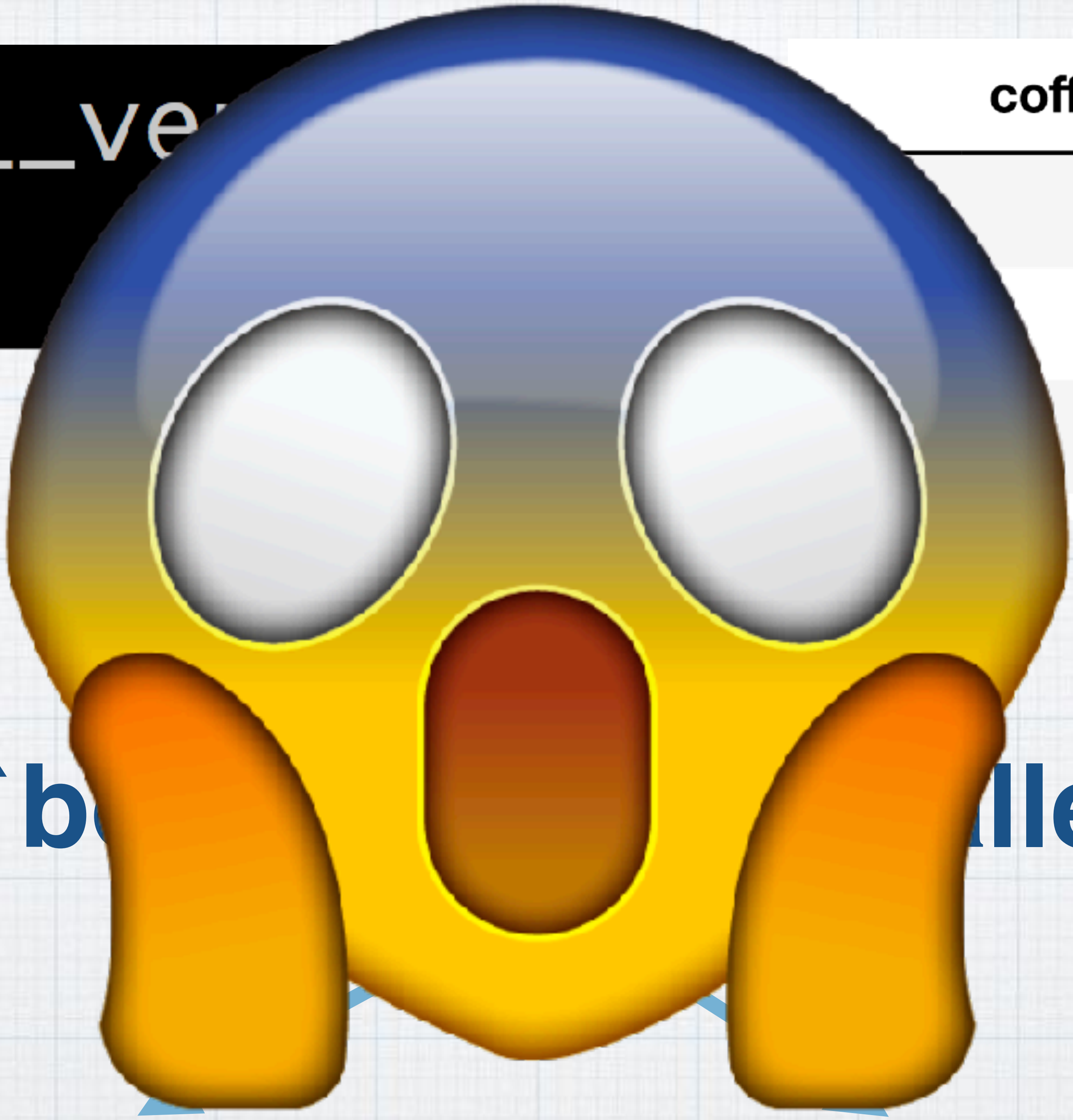


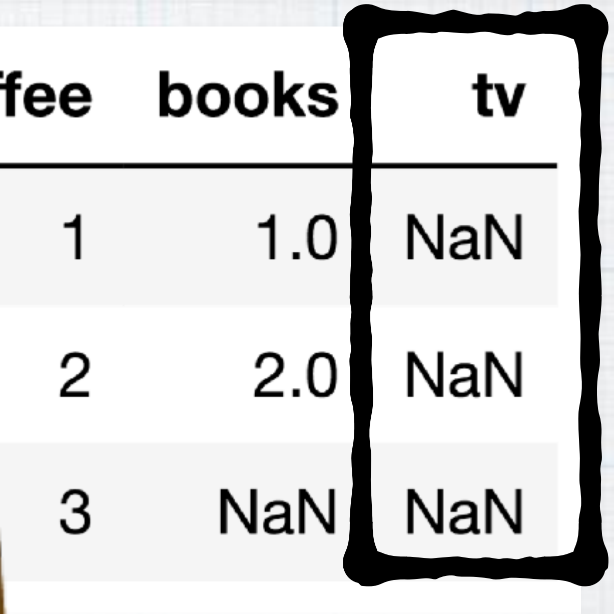
```
>>> d['tv'].sum()
0.0
```

*an optional dependency that `pandas` can call upon to perform operations



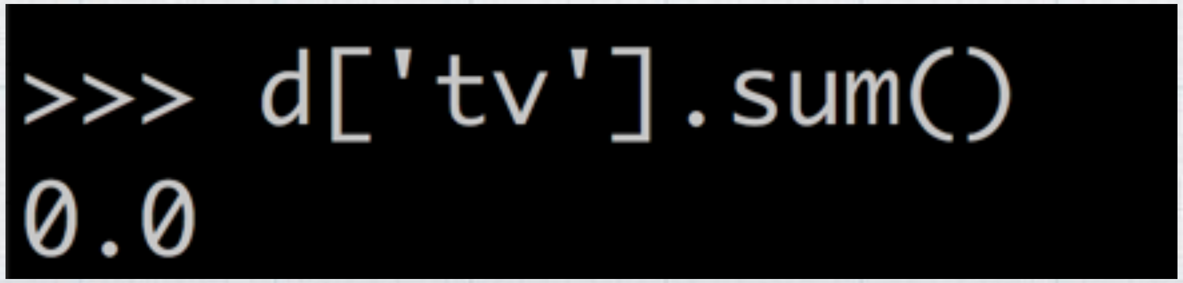
```
>>> pd.__ve
'0.20.3'
```



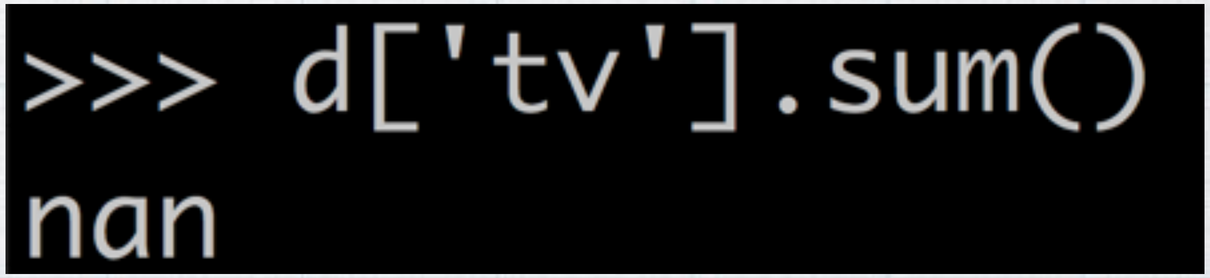


coffee	books	tv
1	1.0	NaN
2	2.0	NaN
3	NaN	NaN

is `b lled?

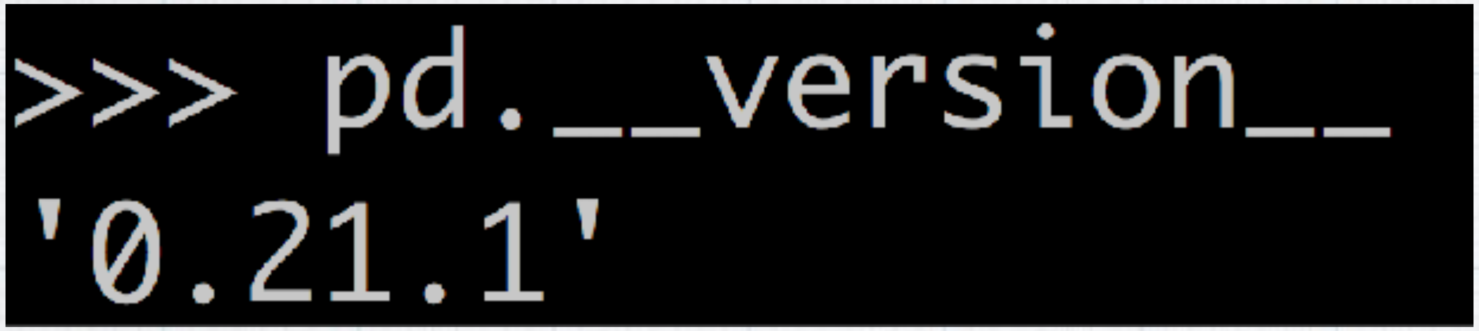


```
>>> d['tv'].sum()
0.0
```

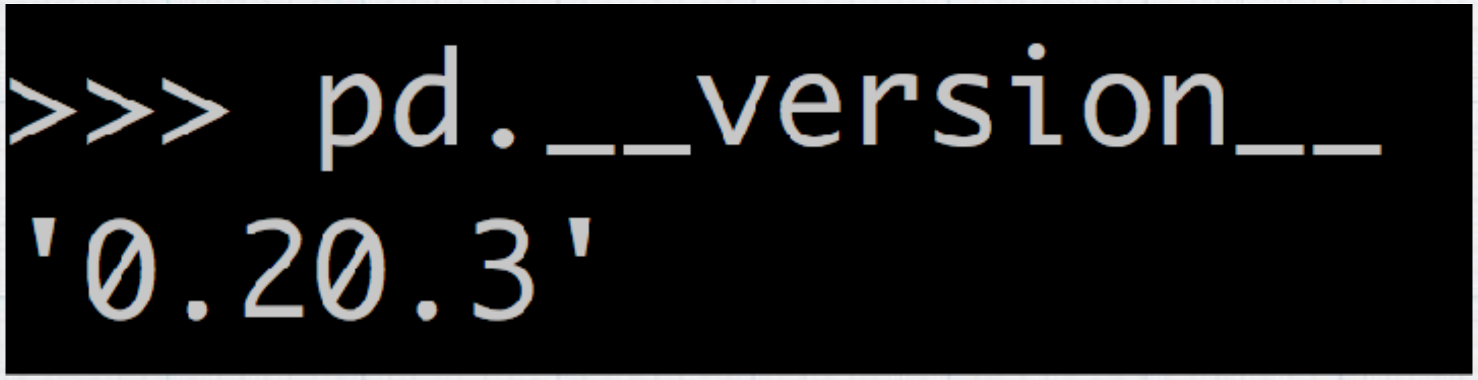


```
>>> d['tv'].sum()
nan
```

*an optional dependency that `pandas` can call upon to perform operations

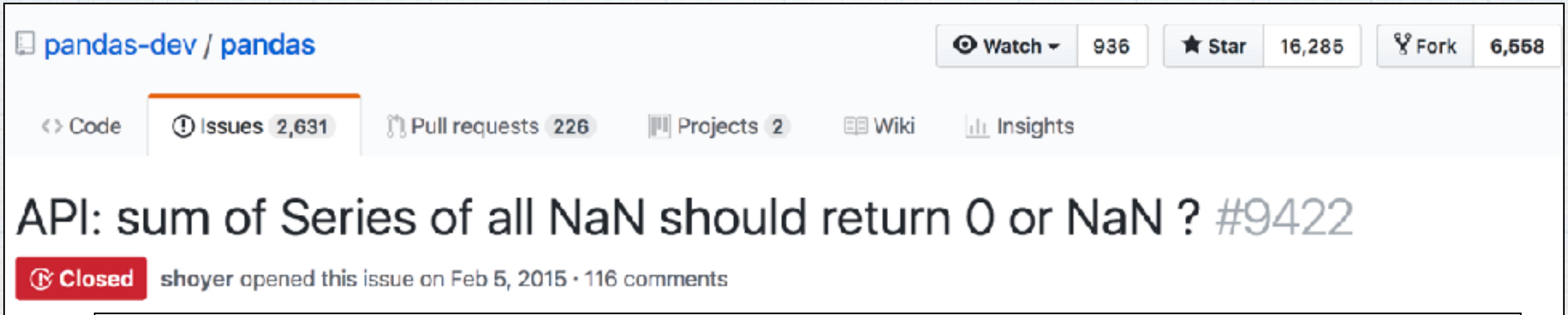


```
>>> pd.__version__
'0.21.1'
```



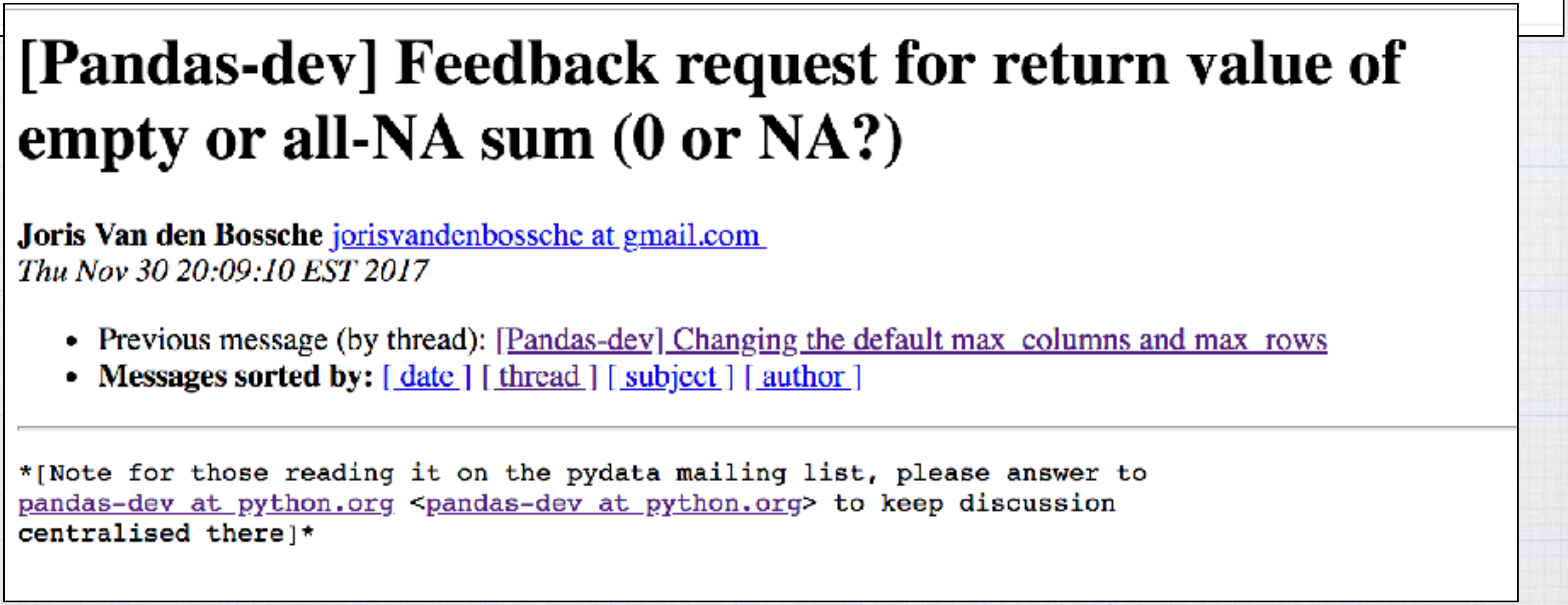
```
>>> pd.__version__
'0.20.3'
```

“I made the sum of an empty list to be NaN …
and the world screamed.”



API: sum of Series of all NaN should return 0 or NaN ? #9422

Closed shoyer opened this issue on Feb 5, 2015 · 116 comments



[Pandas-dev] Feedback request for return value of empty or all-NA sum (0 or NA?)

Joris Van den Bossche jorisvandenbossche at gmail.com
Thu Nov 30 20:09:10 EST 2017

- Previous message (by thread): [Pandas-dev] Changing the default max_columns and max_rows
- **Messages sorted by:** [date] [thread] [subject] [author]

```
*[Note for those reading it on the pydata mailing list, please answer to
pandas-dev at python.org <pandas-dev at python.org> to keep discussion
centralised there]*
```

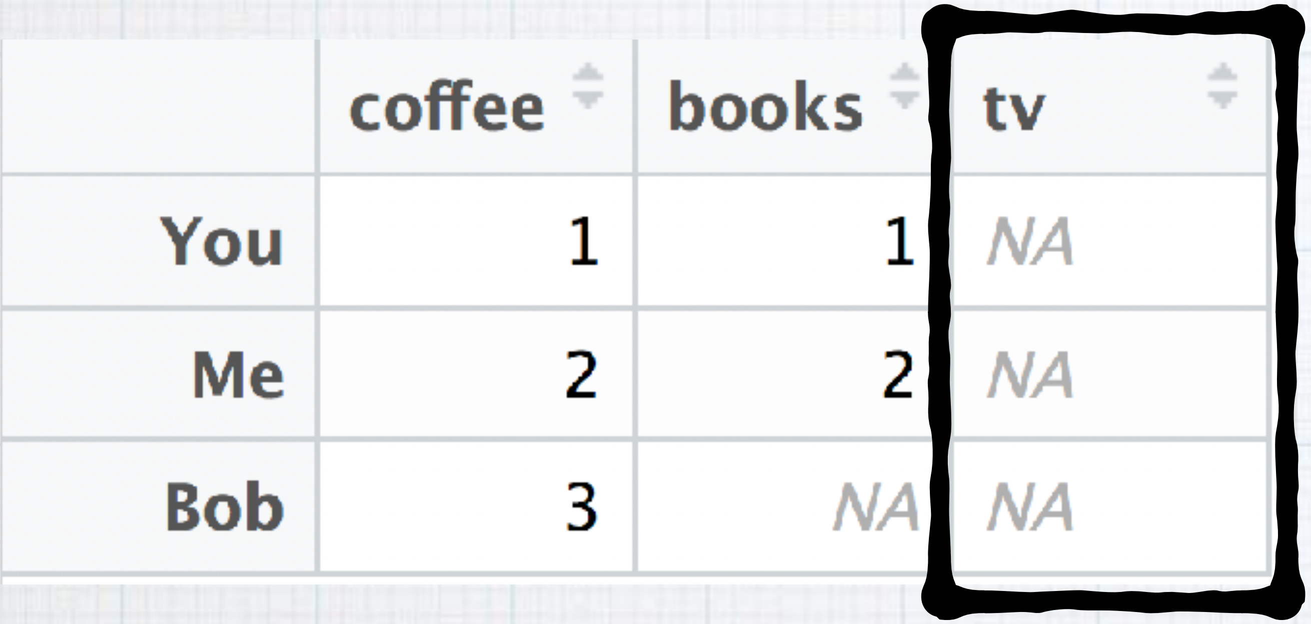
https://github.com/pandas-dev/pandas/issues/9422

https://mail.python.org/pipermail/pandas-dev/2017-November/000657.html

IDGI …

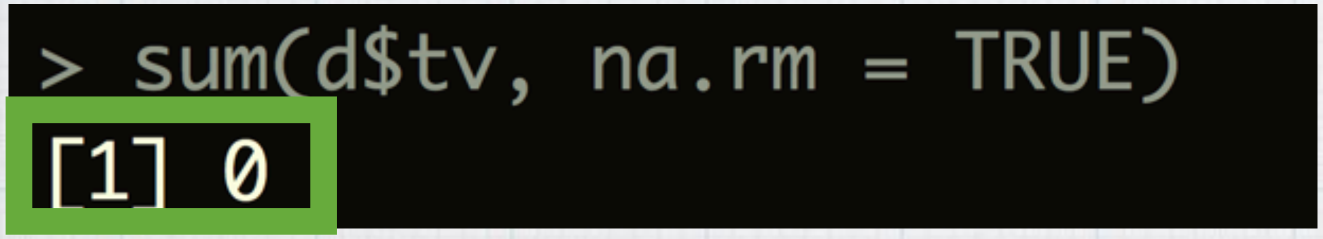
MATH

`d`

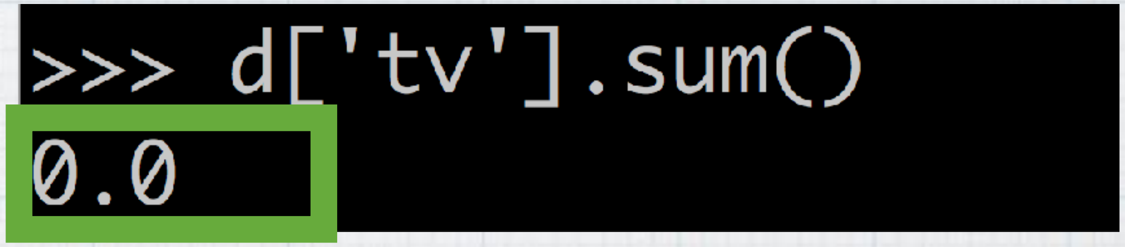


	coffee	books	tv
You	1	1	*NA*
Me	2	2	*NA*
Bob	3	*NA*	*NA*

`na.rm = TRUE` **(or equivalent) happens in Python pandas <u>by default</u>**

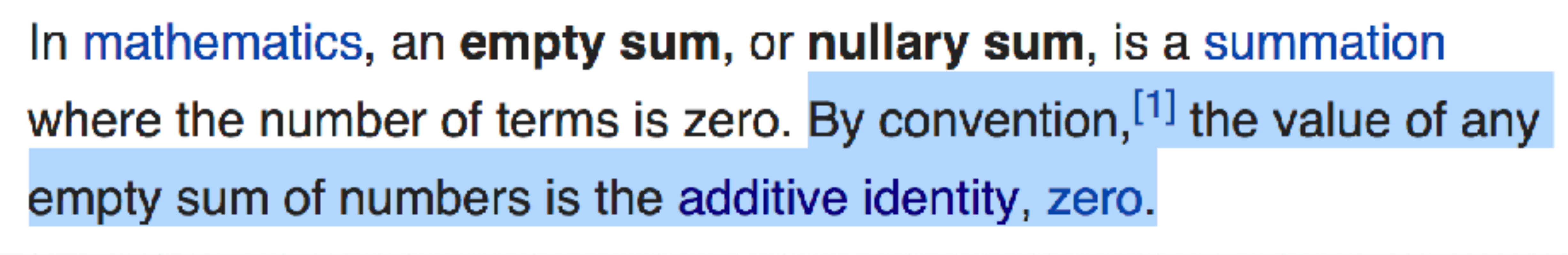


```
> sum(d$tv, na.rm = TRUE)
[1] 0
```



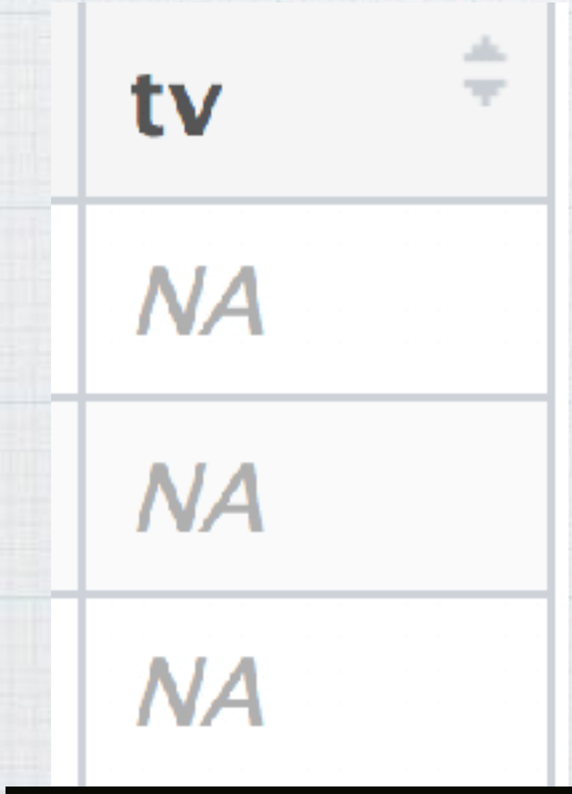
```
>>> d['tv'].sum()
0.0
```

Empty sum

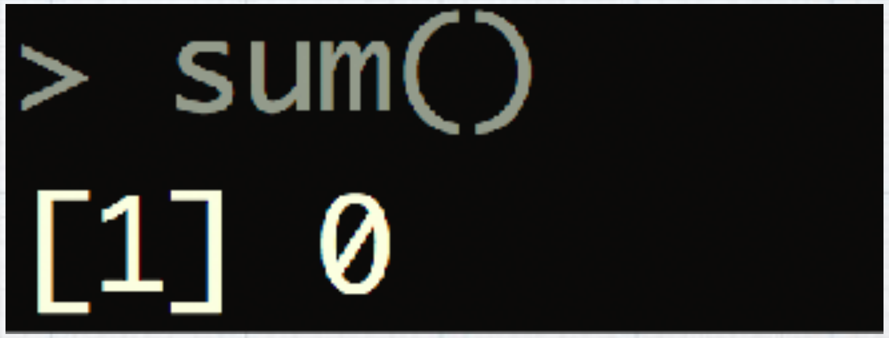


In mathematics, an **empty sum**, or **nullary sum**, is a summation where the number of terms is zero. By convention,[1] the value of any empty sum of numbers is the additive identity, zero.

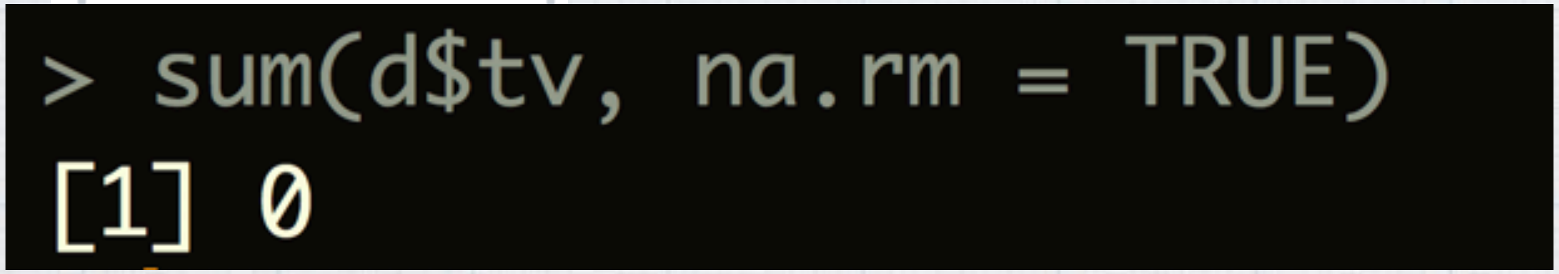
`d`



tv
NA
NA
NA

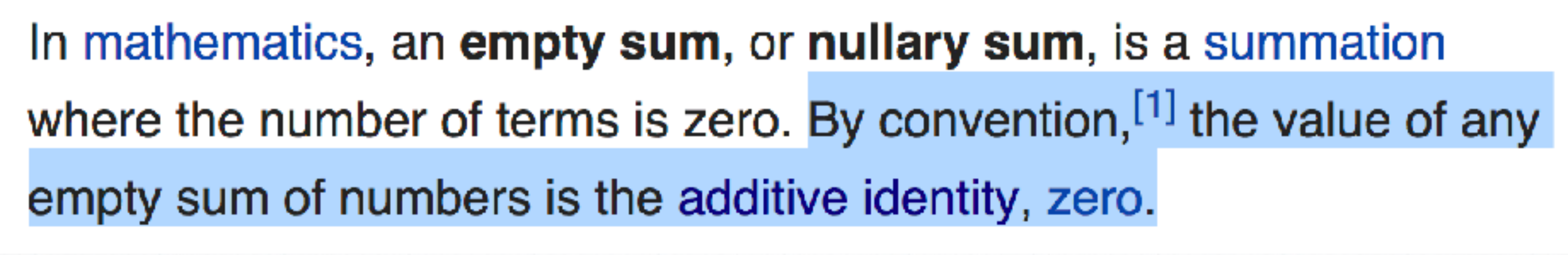


```
> sum()
[1] 0
```



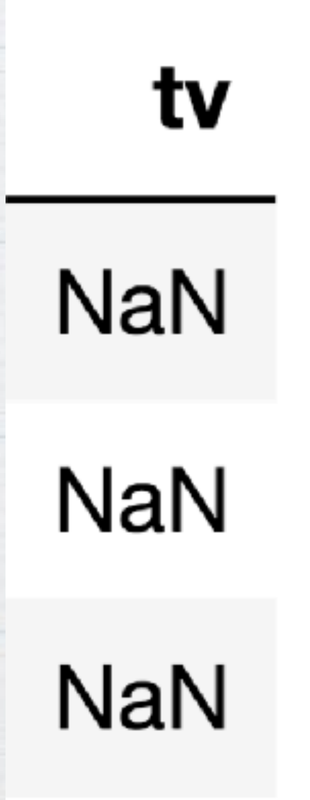
```
> sum(d$tv, na.rm = TRUE)
[1] 0
```

Empty sum

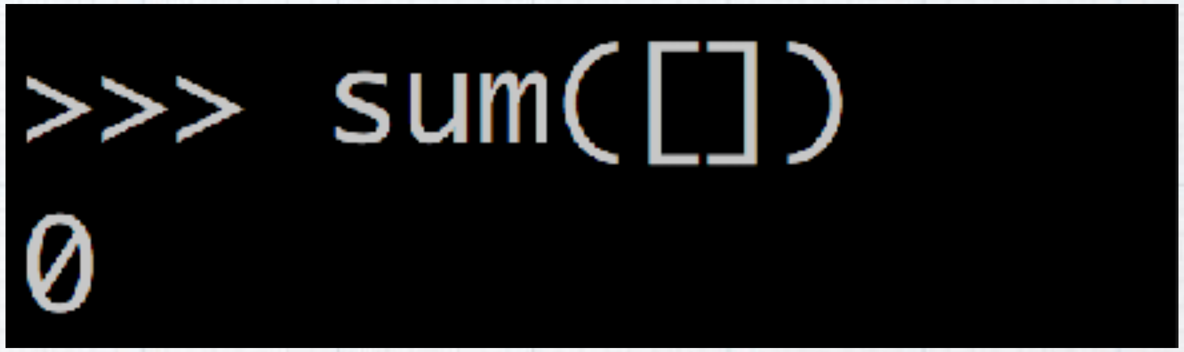


In mathematics, an **empty sum**, or **nullary sum**, is a summation where the number of terms is zero. By convention,[1] the value of any empty sum of numbers is the additive identity, zero.

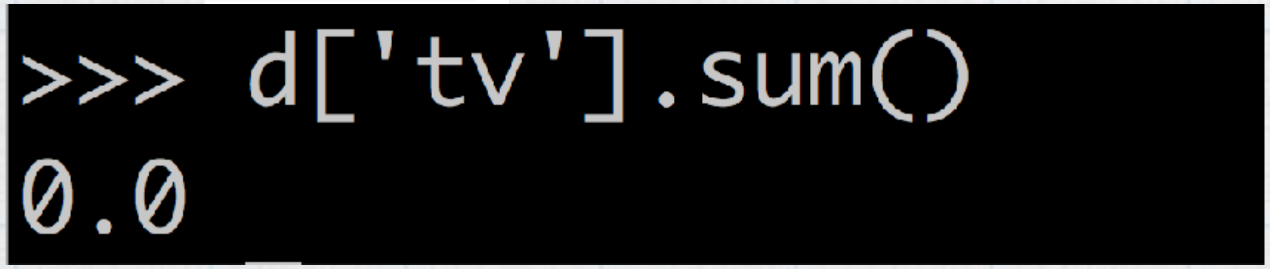
`d`



tv
NaN
NaN
NaN

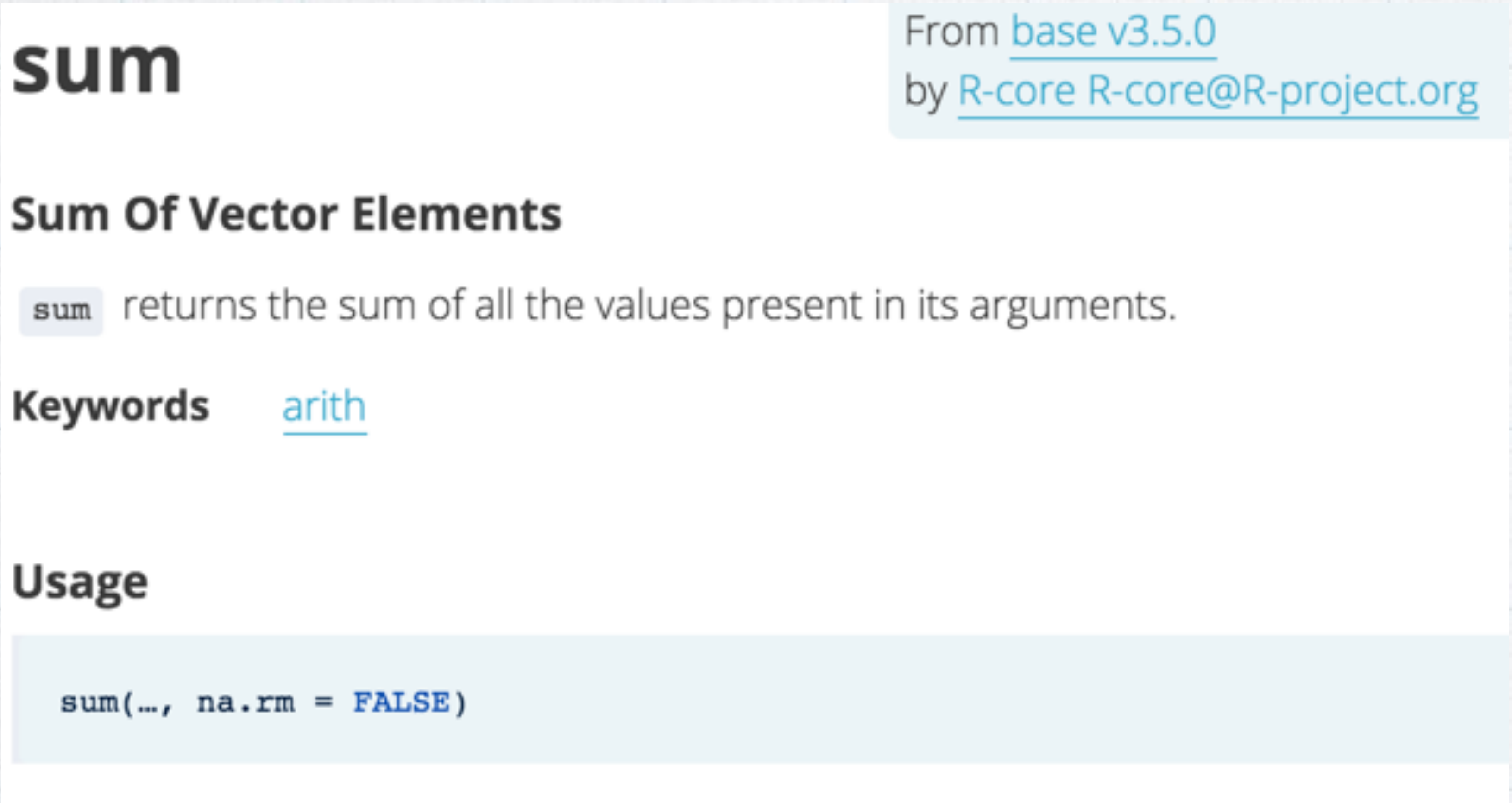


```
>>> sum([])
0
```



```
>>> d['tv'].sum()
0.0
```

Pandas



sum

From base v3.5.0
by R-core R-core@R-project.org

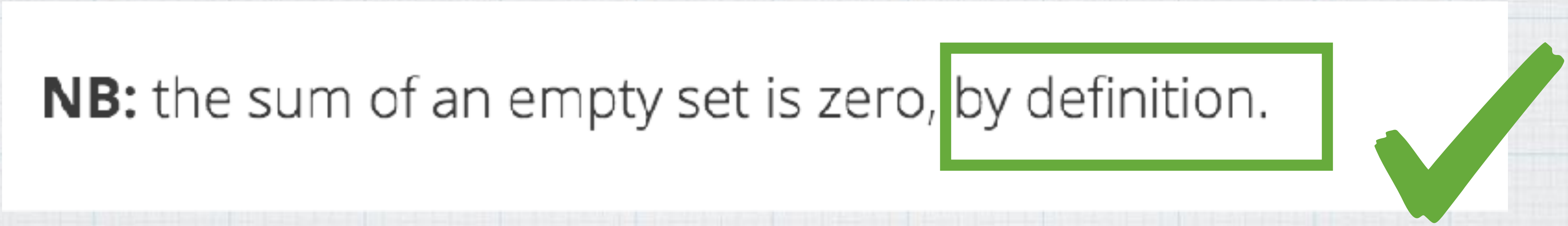
Sum Of Vector Elements

`sum` returns the sum of all the values present in its arguments.

Keywords arith

Usage

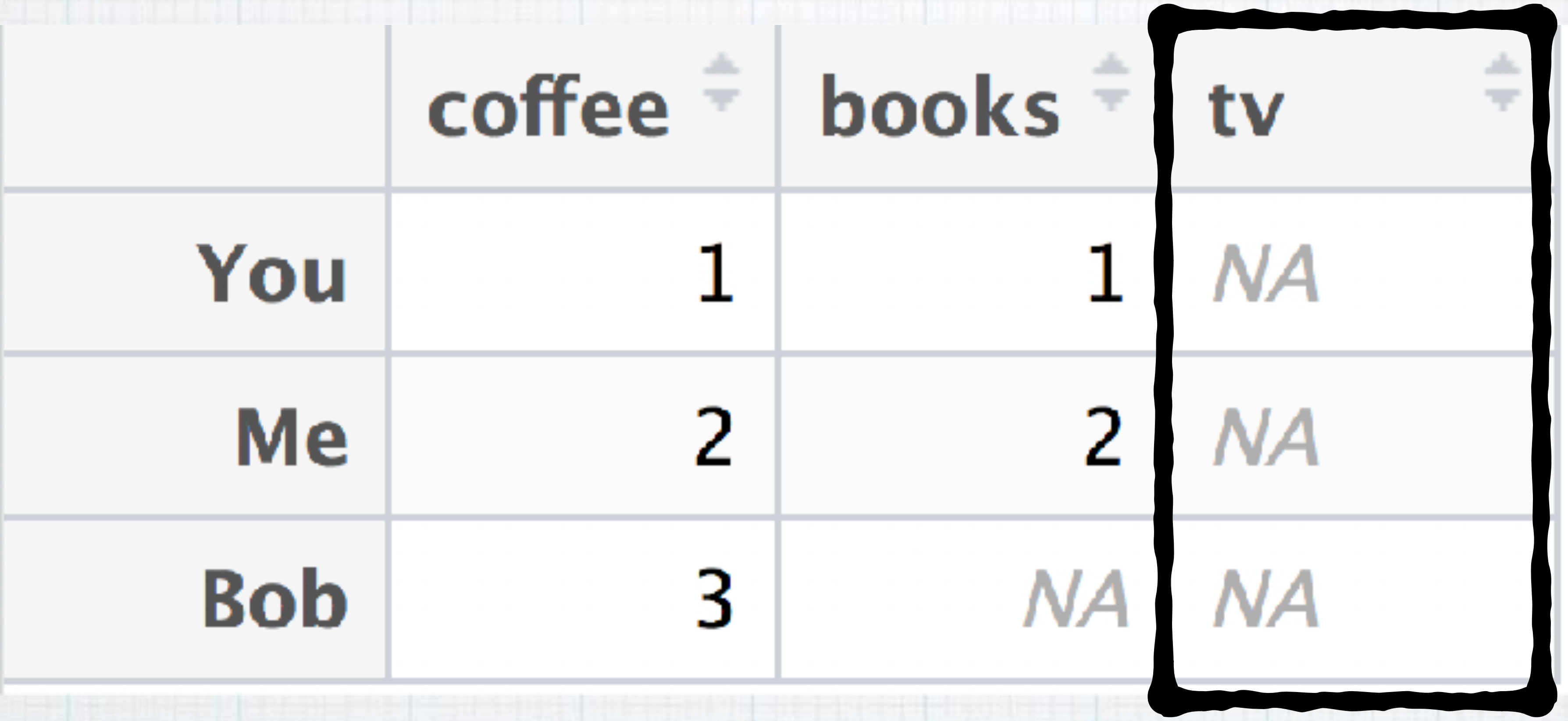
```
sum(…, na.rm = FALSE)
```



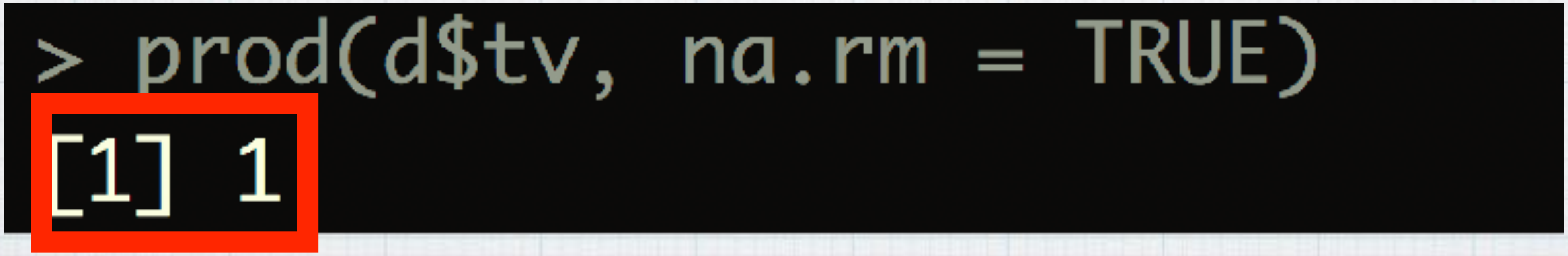
NB: the sum of an empty set is zero, by definition. ✓

https://www.rdocumentation.org/packages/base/versions/3.5.0/topics/sum

df

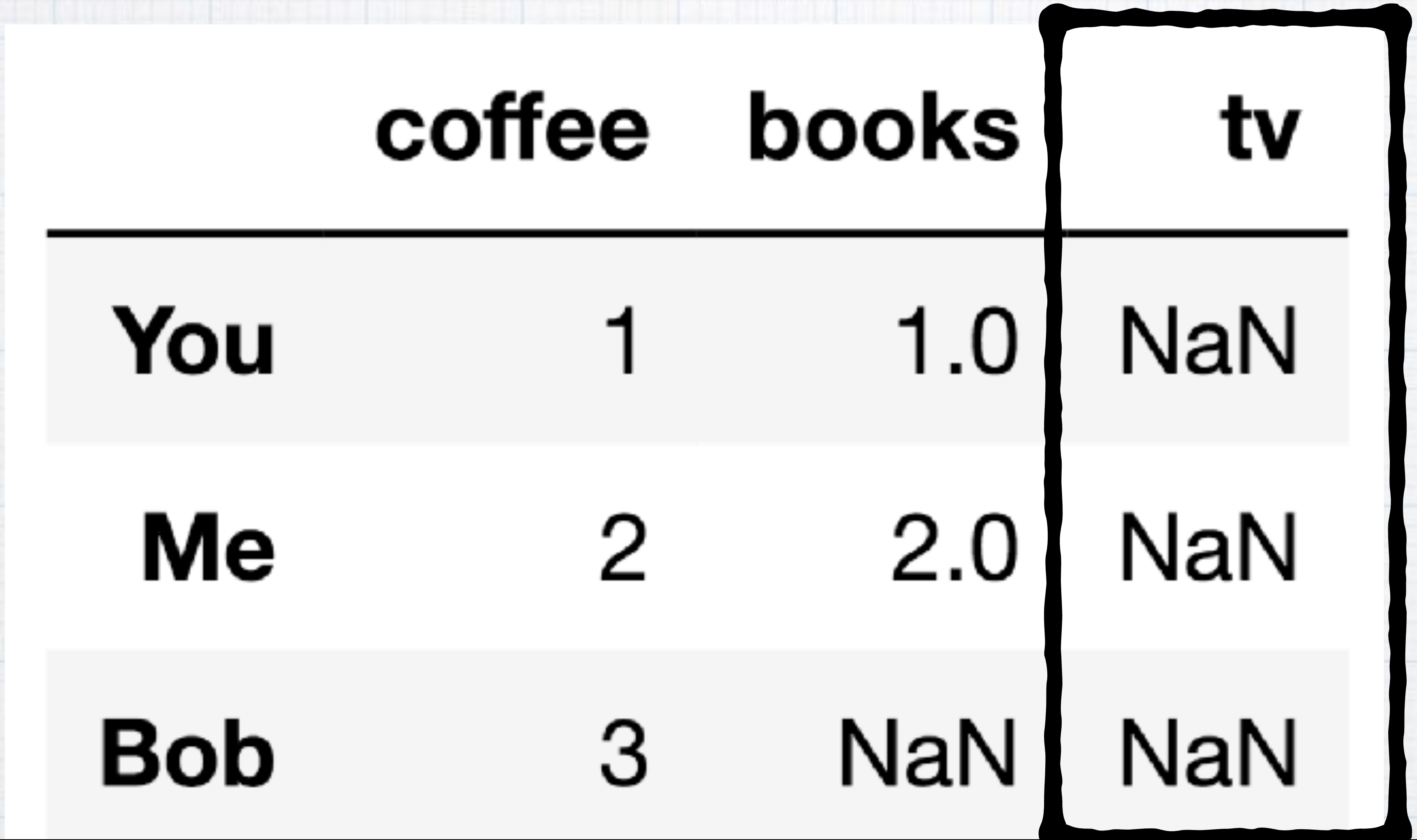


	coffee	books	tv
You	1	1	NA
Me	2	2	NA
Bob	3	NA	NA

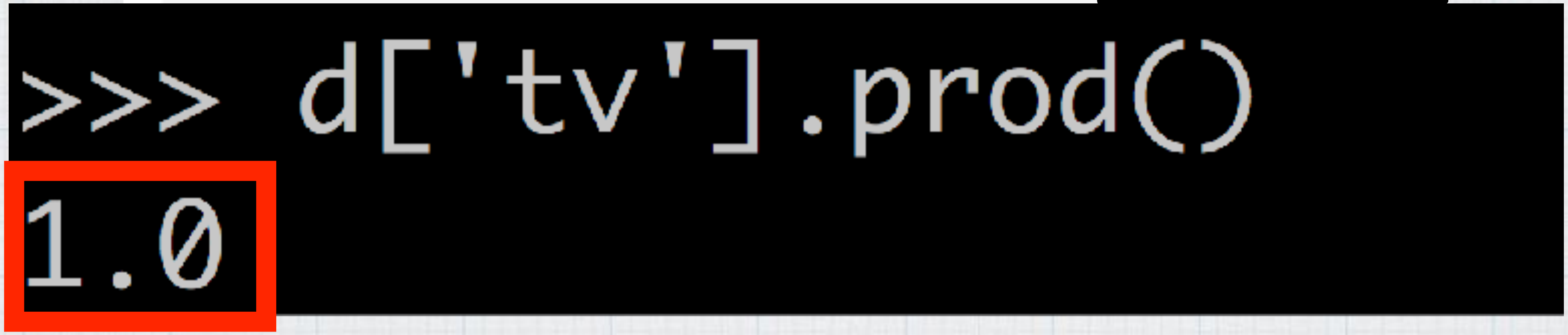


```
> prod(d$tv, na.rm = TRUE)
[1] 1
```

df

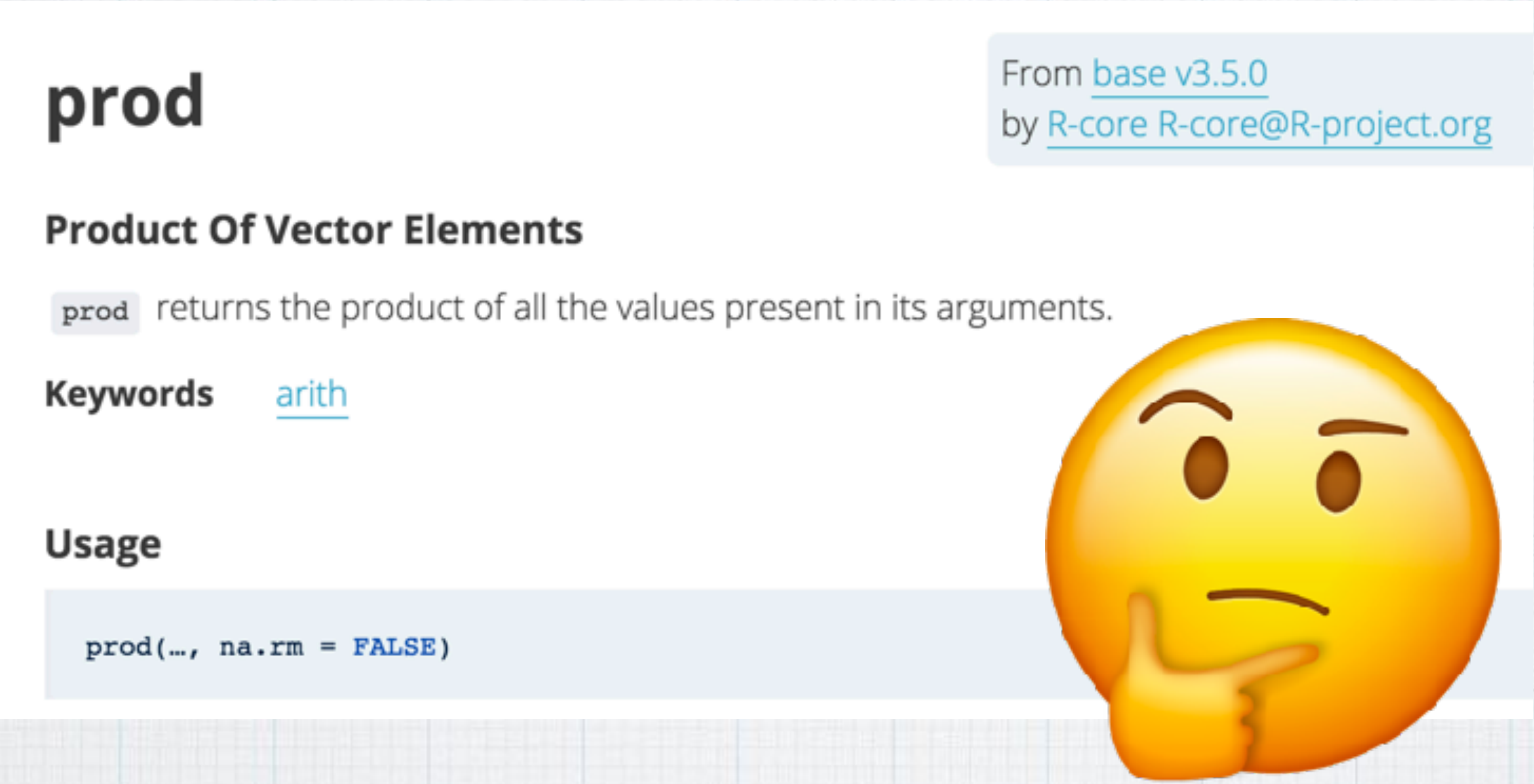


	coffee	books	tv
You	1	1.0	NaN
Me	2	2.0	NaN
Bob	3	NaN	NaN



```
>>> d['tv'].prod()
1.0
```

Pandas



prod

From base v3.5.0
by R-core R-core@R-project.org

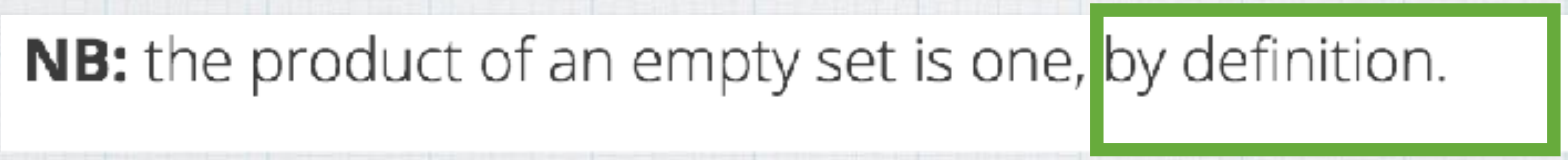
Product Of Vector Elements

`prod` returns the product of all the values present in its arguments.

Keywords arith

Usage

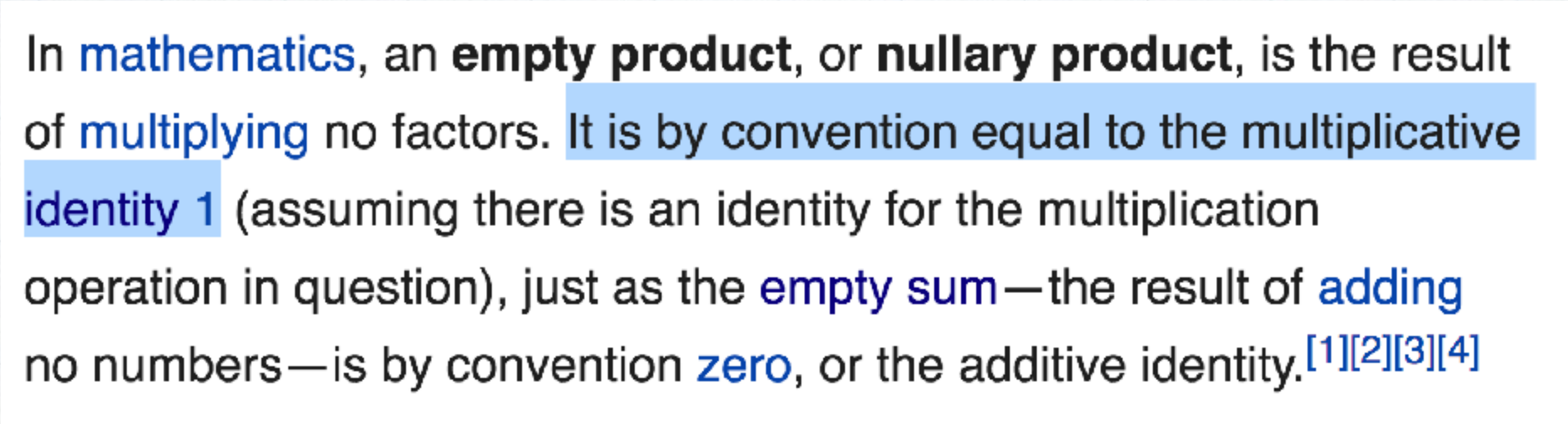
```
prod(…, na.rm = FALSE)
```



NB: the product of an empty set is one, by definition.

https://www.rdocumentation.org/packages/base/versions/3.5.0/topics/prod

Empty product



In mathematics, an **empty product**, or **nullary product**, is the result of multiplying no factors. It is by convention equal to the multiplicative identity 1 (assuming there is an identity for the multiplication operation in question), just as the empty sum—the result of adding no numbers—is by convention zero, or the additive identity.[1][2][3][4]

https://en.wikipedia.org/wiki/Empty_product e

$$e = e^1$$

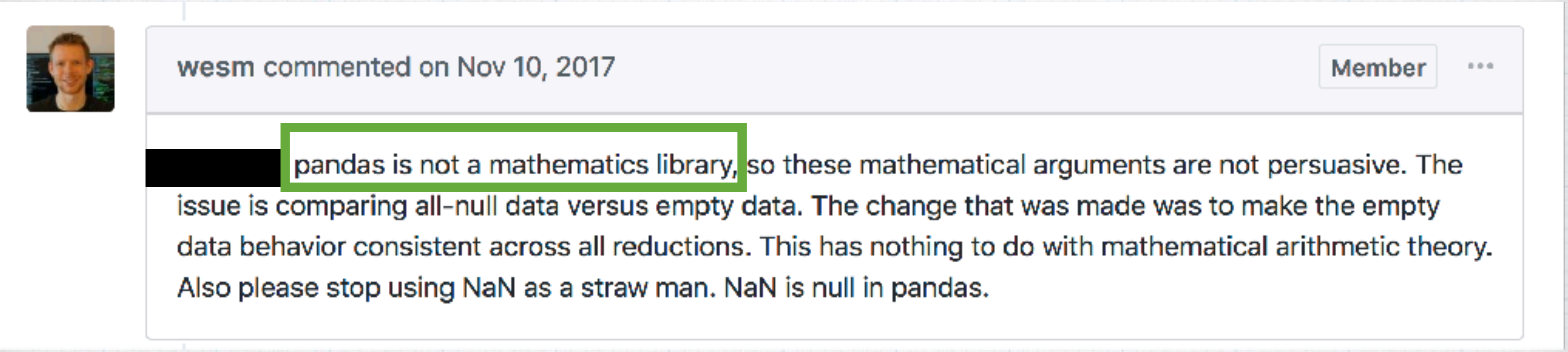
$$
\begin{aligned}
e &= e^{1} \\
&= e^{(1 + 0)}
\end{aligned}
$$

$$
\begin{aligned}
e &= e^{1} \\
&= e^{(1+0)} \\
&= e^{1} \times e^{0}
\end{aligned}
$$

the “empty product”

$$
\begin{aligned}
e &= e^{1} \\
&= e^{(1+0)} \\
&= e^{1} \times e^{0}
\end{aligned}
$$

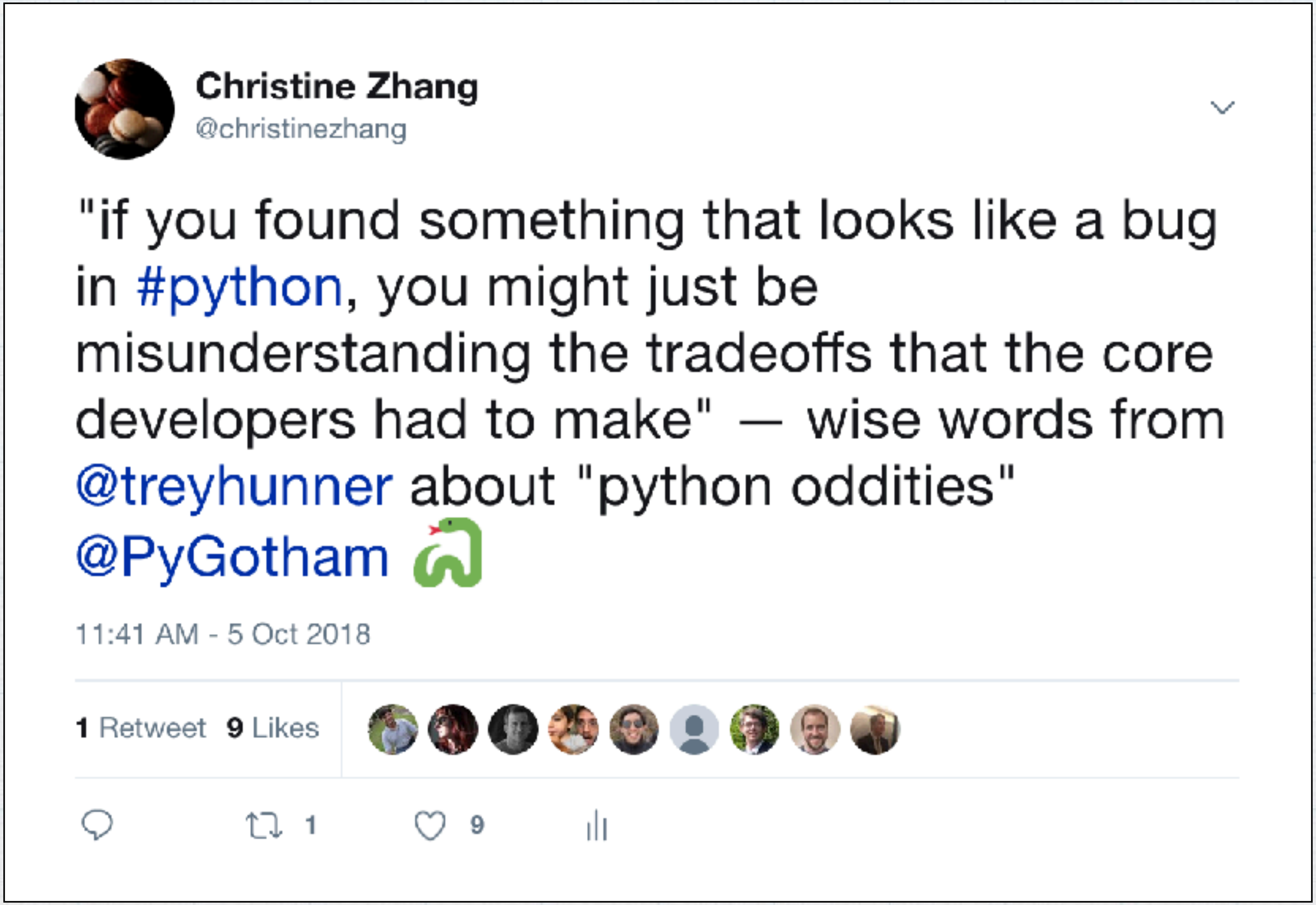
the “empty product” = 1



wesm commented on Nov 10, 2017

[redacted] pandas is not a mathematics library, so these mathematical arguments are not persuasive. The issue is comparing all-null data versus empty data. The change that was made was to make the empty data behavior consistent across all reductions. This has nothing to do with mathematical arithmetic theory. Also please stop using NaN as a straw man. NaN is null in pandas.

https://github.com/pandas-dev/pandas/issues/9422#issuecomment-343553473



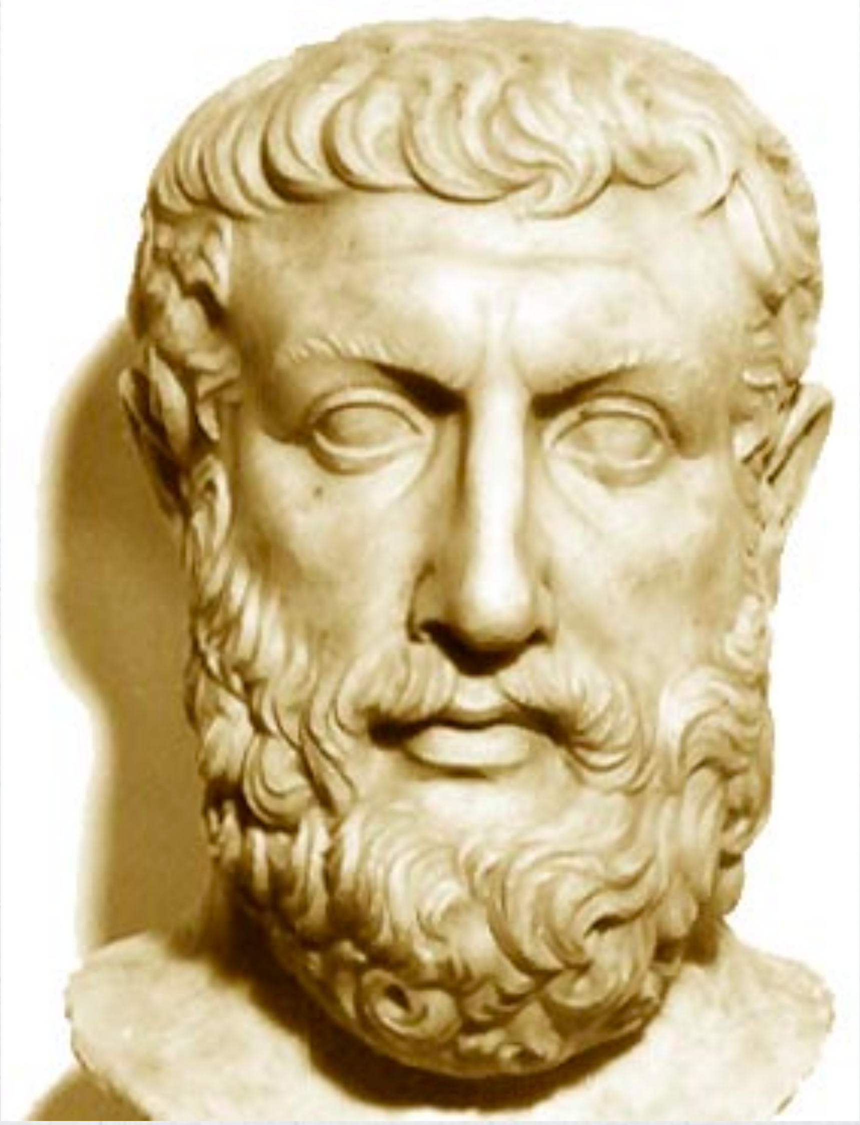
Christine Zhang
@christinezhang

"if you found something that looks like a bug in #python, you might just be misunderstanding the tradeoffs that the core developers had to make" — wise words from @treyhunner about "python oddities" @PyGotham 🐍

11:41 AM - 5 Oct 2018

1 Retweet **9** Likes

Something cannot come from nothing

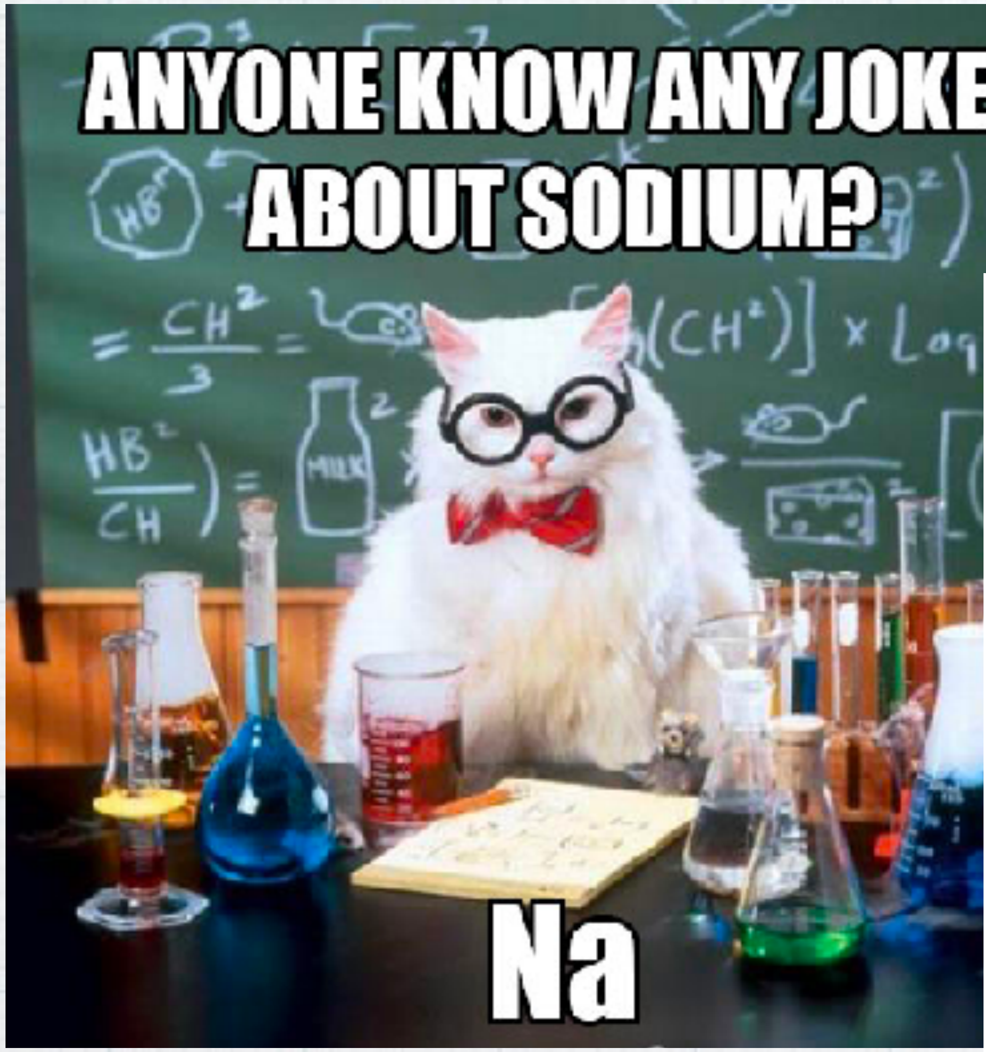


—Parmenides (b. 515 BC)

At first there was nothing … then nothing turned itself inside-out and became something

—Sun Ra (b. 1934) (and Yo La Tengo)

Thank you



@christinezhang

talia jane
@itsa_talia

trains so busted time isn't even numbers anymore

6:07 AM - 10 Jun 2018

ychristinezhang
at gmail dot com