

3.3) Sequences

Vitor Kamada

December 2019

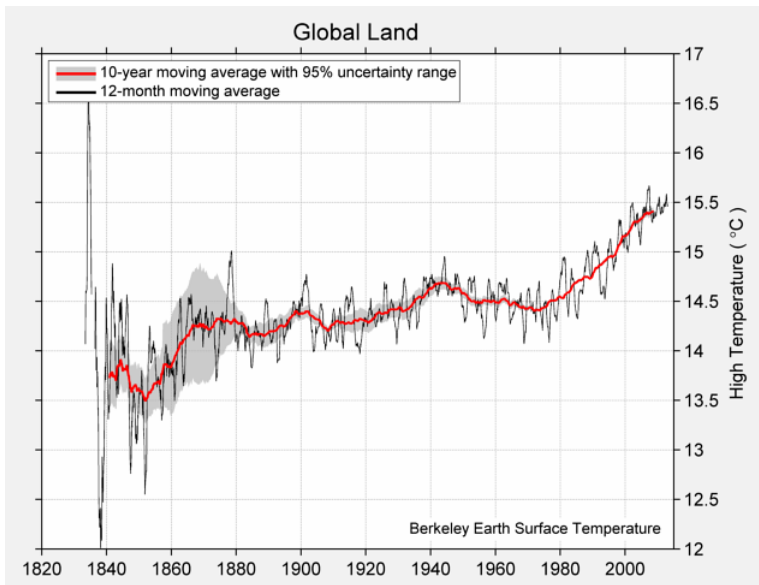
Tables, Graphics, and Figures from

Computational and Inferential Thinking: The Foundations of Data Science

Adhikari & DeNero (2019): Ch 5. Sequences

<https://www.inferentialthinking.com/>

Mean of Daily High Temperature



Convert to Fahrenheit

```
from datascience import *  
baseline_high = 14.48  
highs = make_array(baseline_high - 0.880, baseline_high - 0.093,  
                    baseline_high + 0.105, baseline_high + 0.684)  
highs
```

```
array([13.6   , 14.387, 14.585, 15.164])
```

```
(9/5) * highs + 32
```

```
array([56.48   , 57.8966, 58.253   , 59.2952])
```

highs

$$(9/5) * \begin{bmatrix} 13.6 \\ 14.387 \\ 14.585 \\ 15.164 \end{bmatrix} + 32 = \begin{bmatrix} (9/5) * 13.6 + 32 \\ (9/5) * 14.387 + 32 \\ (9/5) * 14.585 + 32 \\ (9/5) * 15.164 + 32 \end{bmatrix} = \begin{bmatrix} 56.48 \\ 57.8966 \\ 58.253 \\ 59.2952 \end{bmatrix}$$

Average Daily High Temperatures

```
highs.size
```

4

```
highs.sum()
```

57.7360000000000004

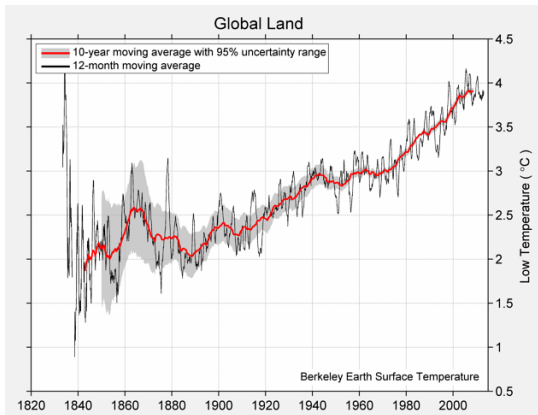
```
sum(highs)/len(highs)
```

14.4340000000000001

```
highs.mean()
```

14.4340000000000001

Mean of Daily Low Temperature



```
baseline_low = 3.00
```

```
lows = make_array(baseline_low - 0.872, baseline_low - 0.629,  
                  baseline_low - 0.126, baseline_low + 0.728)
```

```
lows
```

.item vs array

highs

13.6
14.387
14.585
15.164

lows

2.128
2.371
2.874
3.728

-

=

=

13.6 - 2.128
14.387 - 2.371
14.585 - 2.874
15.164 - 3.728

11.472
12.016
11.711
11.436

```
make_array(  
    highs.item(0) - lows.item(0),  
    highs.item(1) - lows.item(1),  
    highs.item(2) - lows.item(2),  
    highs.item(3) - lows.item(3)  
)
```

```
array([11.472, 12.016, 11.711, 11.436])
```

```
highs - lows
```

Leibniz's Formula for π

$$\pi = 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots \right)$$
$$4 \cdot \left(\left(1 + \frac{1}{5} + \frac{1}{9} + \dots + \frac{1}{9997} \right) - \left(\frac{1}{3} + \frac{1}{7} + \frac{1}{11} + \dots + \frac{1}{9999} \right) \right)$$

```
import numpy as np
positive_term_denominators = np.arange(1, 10000, 4)
positive_term_denominators

array([ 1, 5, 9, ..., 9989, 9993, 9997])

positive_terms = 1 / positive_term_denominators
negative_terms = 1 / (positive_term_denominators + 2)
4 * ( sum(positive_terms) - sum(negative_terms) )
```

3.1413926535917955

Wallis' Formula for π

$$\pi = 2 \cdot \left(\frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdots \right)$$

$$\pi \approx 2 \cdot \left(\frac{2}{1} \cdot \frac{4}{3} \cdot \frac{6}{5} \cdots \frac{1,000,000}{999999} \right) \cdot \left(\frac{2}{3} \cdot \frac{4}{5} \cdot \frac{6}{7} \cdots \frac{1,000,000}{1,000,001} \right)$$

```
even = np.arange(2, 1000001, 2)
one_below_even = even - 1
one_above_even = even + 1
2 * np.prod(even/one_below_even) * np.prod(even/one_above_even)
```

3.1415910827951143