

BC COMS 1016: Intro to Comp Thinking & Data Science

Lecture 3

Tables, Array, Sequences

Reminders



- HW00 due tomorrow
 - Individual assignment
 - Ask questions in slack #homeworks
 - Answer your peers questions as well
 - Can count for participation grade
- Lab00 due tonight



- Partners for labs:
 - Upgrade Zoom to v5.x
 - Choose a breakout room on google sheet
 - Sheet is pinned on #labs



Autograders/Grading

Question 1.1. In the next cell, assign

1. the **absolute value** of $2^5 - 2^1$
2. $5 \times 13 \times 31 + 5$.

Try to use just one statement (one line of code) for both parts.

```
new_year = ...  
new_year
```

```
grader.check("q1_1")
```



Autograders/Grading – Error 1

```
NameError Traceback (most recent call last)
<ipython-input-1-1ad9a283f073> in <module>()
----> 1 grader.check("q1_1")

NameError: name 'grader' is not defined
```

```
# Initialize Otter
import otter
grader = otter.Notebook()
```



Autograders/Grading – Error 2

```
-----  
NameError: name 'new_year' is not defined
```

```
In [ ]: new_year = ...  
        new_year
```

```
In [2]: grader.check("q1_1")
```

Grading based on autograder



- Before we “publish” scores
 - Visible:
 - Status of tests (pass/fail)
 - Errors of failing test
 - Not visible
 - points associated with the tests
- Publish results after the assignment submission is closed
 - 2 days after deadline

Course/lab registration

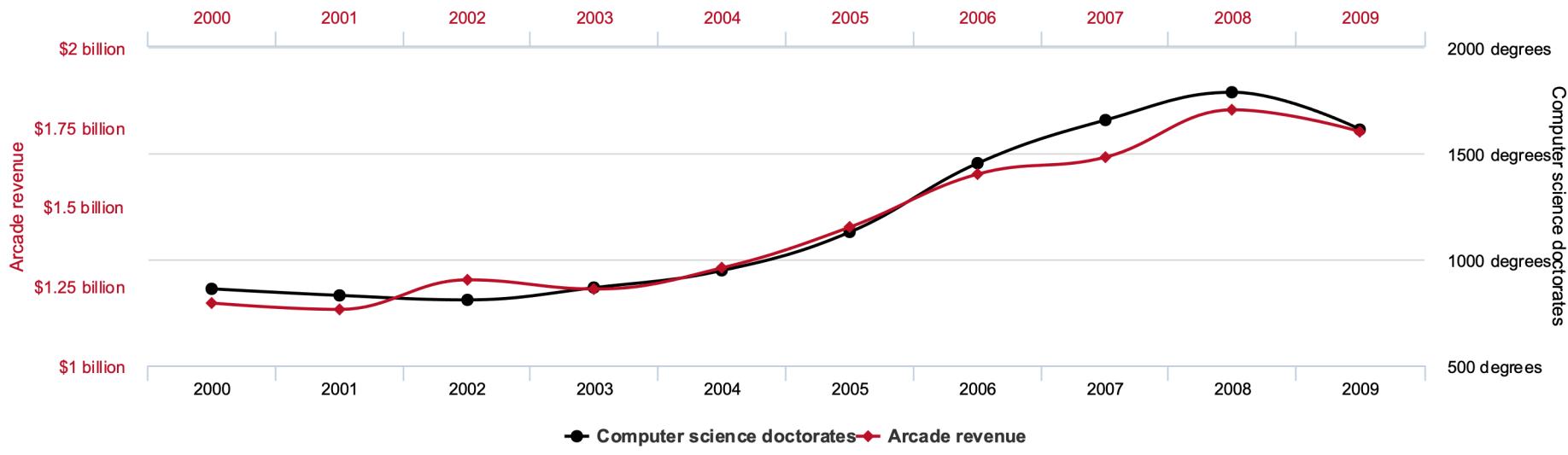


- If not officially registered for the course
 - BC – email class dean, cc' me
 - CU – fill out change of registration form

Cause & Effect



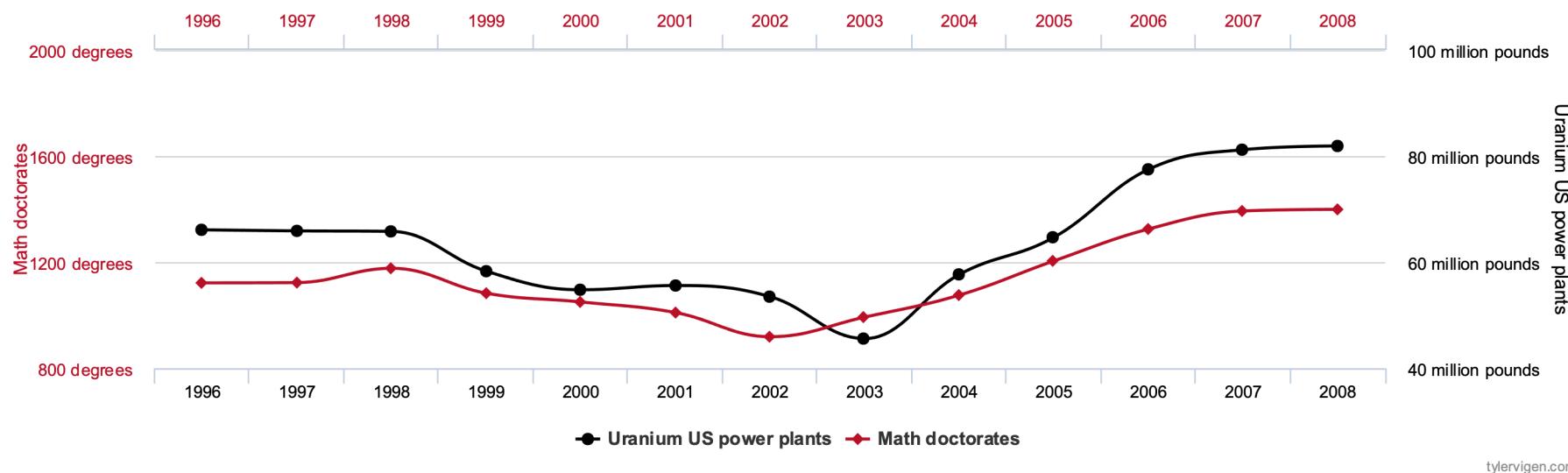
Total revenue generated by arcades correlates with Computer science doctorates awarded in the US



Cause & Effect



Math doctorates awarded correlates with Uranium stored at US nuclear power plants



<https://www.tylervigen.com/spurious-correlations>

A black and white photograph of the exterior of a classical building, likely Barnard College. The building features large, ornate Corinthian columns supporting a multi-tiered entablature. The word "BARNARD" is visible on the facade above one of the columns. The sky is clear and blue.

Tables



Table Structure

- A Table is a sequence of labeled columns
- Row: represents one individual
- Column: represents one attribute of the individuals

Name	Code	Area (m2)
California	CA	163696
Nevada	NV	110567



Creating a Table

- `Table.read_table(filename)` – reads a table from a spreadsheet
- `Table()` – an empty table



Table methods

- Creating and extending tables:
 - `Table().with_column` and `Table.read_table`
- Finding the size:
 - `num_rows` , `num_columns`
- Referring to columns: labels, relabeling and indices
 - `labels` and `relabelled`; column indices start at 0

Some Table Operations



- `t.select(label)` – constructs a new table with just the specified columns
- `t.drop(label)` – constructs a new table in which the specified columns are omitted
- `t.sort(label)` – constructs a new table with rows sorted by the specified column
- `t.where(label, condition)` – constructs a new table with just the rows that match the condition
- These operations create a new table



— Array —



An array contains a sequence of values

- All elements of an array should have the same type
- Arithmetic is applied to each element individually
- Adding arrays add elements (**if same length!**)
- A column of a table is in an array



A range is an array of consecutive numbers

- np.arange(end):
An array of increasing integers from 0 up to end
- np.arange(start, end):
An array of increasing integers from start up to end
- np.arange(start, end, step):
A range with step between consecutive values

The range always include start but excludes end

Array Functions & Methods



Name	Chapter	Description
<code>max(array)</code>	3.3	Returns the maximum value of an array
<code>min(array)</code>	3.3	Returns the minimum value of an array
<code>sum(array)</code>	3.3	Returns the sum of the values in an array
<code>abs(num), np.abs(array)</code>	3.3	Take the absolute value of number or each number in an array.
<code>round(num), np.round(array)</code>	3.3	Round number or array of numbers to the nearest integer.
<code>len(array)</code>	3.3	Returns the length (number of elements) of an array
<code>make_array(val1, val2, ...)</code>	5	Makes a numpy array with the values passed in
<code>np.average(array) np.mean(array)</code>	5.1	Returns the mean value of an array
<code>np.std(array)</code>	14.2	Returns the standard deviation of an array
<code>np.diff(array)</code>	5.1	Returns a new array of size <code>len(arr)-1</code> with elements equal to the difference between adjacent elements; <code>val_2 - val_1, val_3 - val_2</code> , etc.
<code>np.sqrt(array)</code>	5.1	Returns an array with the square root of each element
<code>np.arange(start, stop, step) np.arange(start, stop) np.arange(stop)</code>	5.2	An array of numbers starting with <code>start</code> , going up in increments of <code>step</code> , and going up to but excluding <code>stop</code> . When <code>start</code> and/or <code>step</code> are left out, default values are used in their place. Default step is 1; default start is 0.
<code>array.item(index)</code>	5.3	Returns the i-th item in an array (remember Python indices start at 0!)
<code>np.random.choice(array, n) np.random.choice(array)</code>	9	Picks one (by default) or some number 'n' of items from an array at random. By default, with replacement.
<code>np.count_nonzero(array)</code>	9	Returns the number of non-zero (or <code>True</code>) elements in an array.
<code>np.append(array, item)</code>	9.2	Returns a copy of the input array with <code>item</code> (must be the same type as the other entries in the array) appended to the end.
<code>percentile(percentile, array)</code>	13.1	Returns the corresponding percentile of an array.

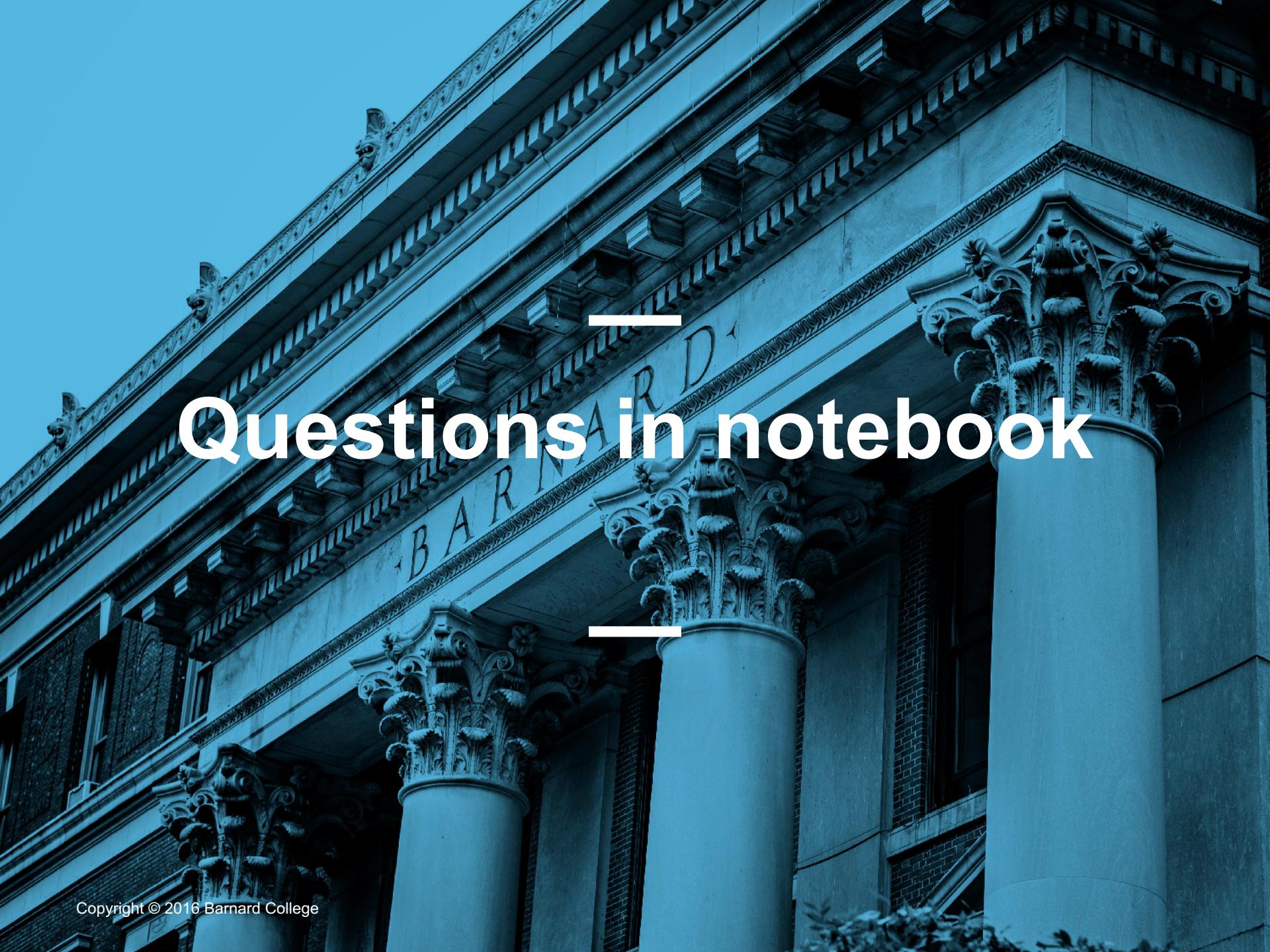


Tables & Arrays

Table methods



- Accessing data in a column
 - `Column` takes a label or index and returns an array
- Using array methods to work with data in columns
 - `item`, `sum`, `min`, `max`, and so on
- Creating new tables containing some of the original columns
 - `select`, `drop`



Questions —

in notebook



Questions:

The table **nba** has columns

PLAYER, POSITION, and SALARY

```
table = Table.read_table('https://www.inferentialthinking.com/data/nba_salaries.csv')
```

1. Create an array containing the names of all centers (C) who make more than \$15M/year

```
centers = table.where('POSITION', 'C')
```

```
centers.where('\'15-\'16 SALARY', are.above(15)).column('PLAYER')
```

Answer:

'Dwight Howard', 'Roy Hibbert', 'Marc Gasol', 'Enes Kanter', 'DeMarcus Cousins'