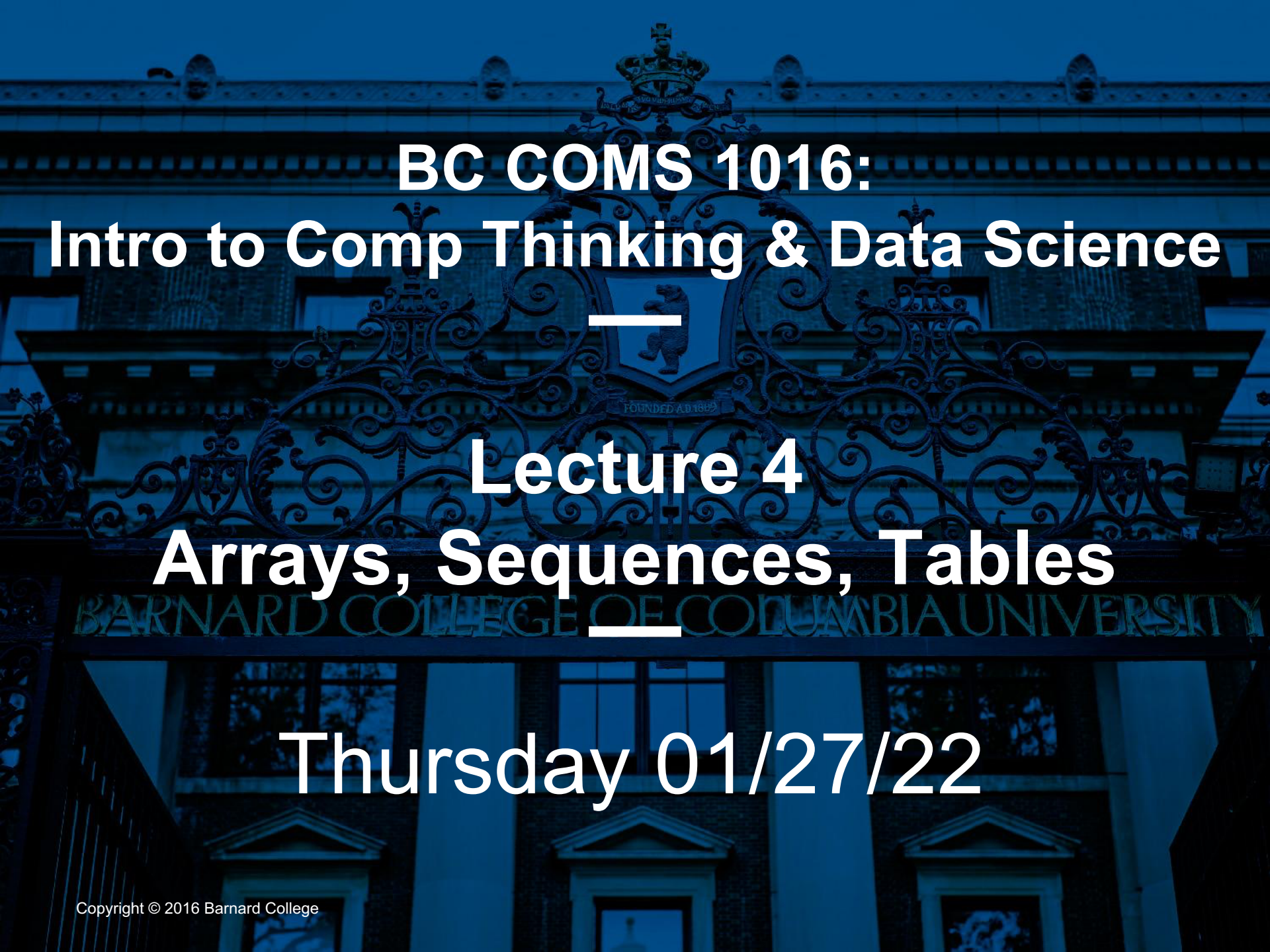


BC COMS 1016:
Intro to Comp Thinking & Data Science

Lecture 4
Arrays, Sequences, Tables

Thursday 01/27/22

Reminders

- HW00 due tonight
  - Individual assignment
  - Only 41 submissions (as of10am today)
  - You can use 2 late days
- Lab01 due Monday
- Lab00:
  - If you havent gotten it in yet, do it before the other assignments
- HW01
  - Will be released tonight
  - Due Thursday 02/03

Office Hours & Next few classes

- Today:
  - Adam 1pm-2pm after class
- Next week:
  - Tuesday still remote
  - Thursday TBD
- Tuesday Feb 8th:
  - TA review or watch last year's recording or no class
  - Your choice

Gradescope – Results vs Code View

STUDENT
Adam Poliak

AUTOGRADER SCORE
- / 16.0

QUESTION 2
Question 1 - / 2.0 pts

QUESTION 3
Question 4.3 - / 1.0 pts

QUESTION 4
Question 6.1 - / 2.0 pts

QUESTION 5
Question 6.2 - / 1.0 pts

QUESTION 6
Question 6.3 - / 1.0 pts

QUESTION 7
Question 6.4 - / 1.0 pts

QUESTION 8
Question 9.2 - / 0.0 pts

STUDENT
Adam Poliak

AUTOGRADER SCORE
- / 16.0

FAILED TESTS
q2_1 - Public (0.0/1.0)
q2_2 - Public (0.0/1.0)
q3_1 - Public (0.0/1.0)
q3_2 - Public (0.0/1.0)
q3_3 - Public (0.0/1.0)
q4_1 - Public (0.0/1.0)
q4_2 - Public (0.0/1.0)
q5_1 - Public (0.0/1.0)
q5_2 - Public (0.0/1.0)
q5_3 - Public (0.0/3.0)
q5_4 - Public (0.0/1.0)
q8_1 - Public (0.0/1.0)
q7_1 - Public (0.0/1.0)

PASSED TESTS
q9_1 - Public (0.0/0.0)

QUESTION 2
Question 1 - / 2.0 pts

QUESTION 3
Question 4.3 - / 1.0 pts

QUESTION 4
Question 6.1 - / 2.0 pts

QUESTION 5
Question 6.2 - / 1.0 pts

Course Outline

- Exploration — **Week 1 - 5**
  - Introduction to Python
  - Working with data
- Inference — **Week 6 - 10**
  - Probability
  - Statistics
- Prediction — **Week 11-14**
  - Machine Learning
  - Regression & Classification

Course Outline

- Exploration **Week 1 - 5**
  - Discover patterns
  - Articulate insights
- Inference **Week 6 - 10**
  - Make reliable conclusions about the world
  - Statistics is useful
- Prediction **Week 11-14**
  - Informed guesses about unseen data

Types – Every value has a type

We've seen 5 types so far:

- int: 2
- float: 2.2
- str: 'Red fish, blue fish'
- builtin_function_or_method: abs, max, min
- Table

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

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 relabeled; column indices start at 0

Tables – select and drop

- t.select(…) – constructs a new table with just the specified columns
- t.drop(...) – constructs a new table in which the specified columns are omitted

- These operations create a new table

Tables – select and drop

- .select(<Column Name>)
  - Returns a new table with the specified columns
- .select(<Int i>)
  - Returns a new table with the column at index I
- drop(<Column Name>)
  - Returns a new table without the specified columns
- .drop(<Int i>)
  - Returns a new table without the column at index i

Some Table Operations

- t.sort(label) – constructs a new table with rows sorted by the specified column
- t.where(label, condiction) – constructs a new table with just the rows that match the condition

- More are listed at http://coms1016.barnard.edu/python-reference.html

Array

Arrays

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
  - All values in a single column are the same type

Ranges

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.arrange(start, end, step):
  A range with step between consecutive values

The range always include start but excludes end

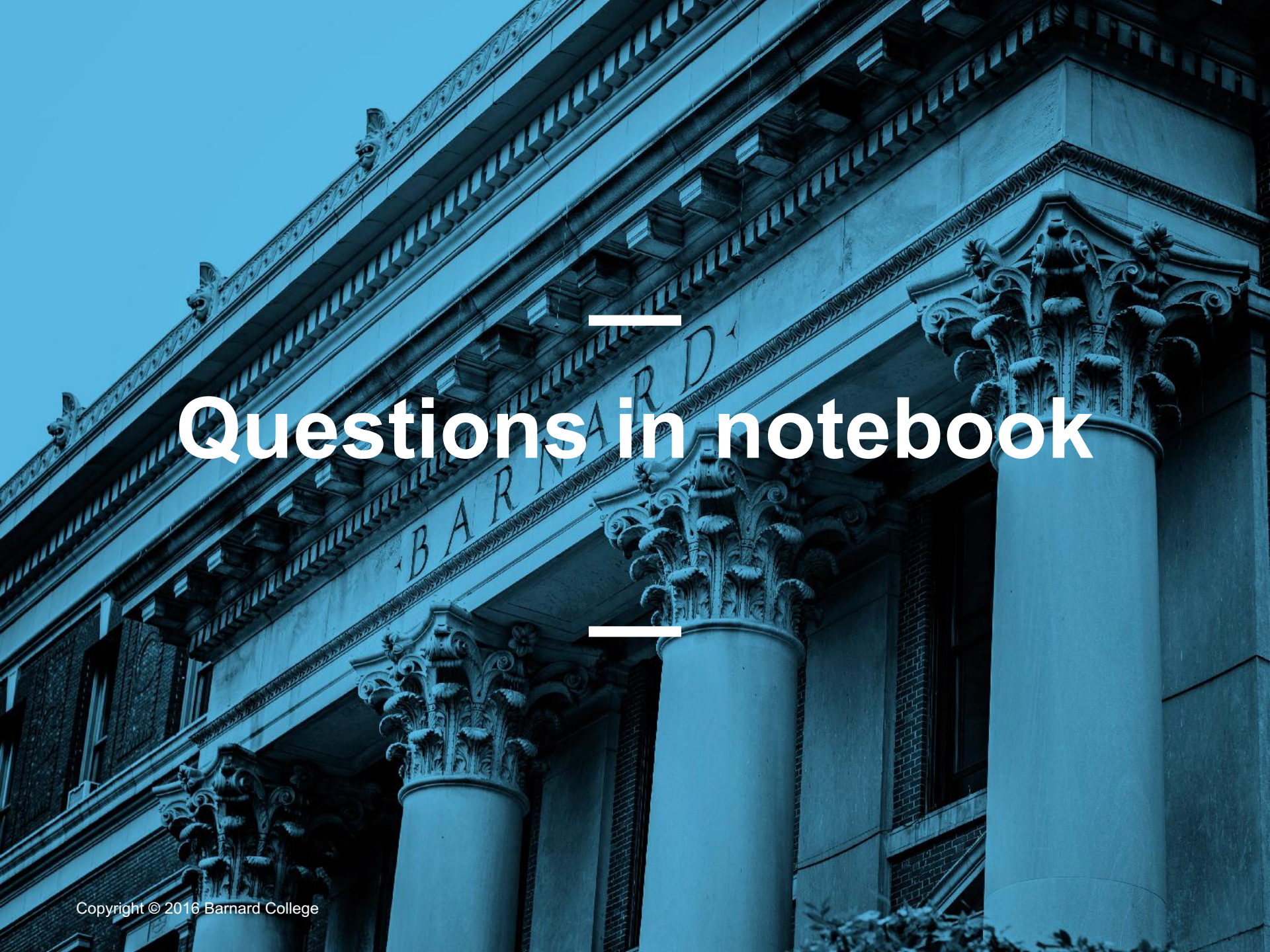
Array Functions & Methods

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

Attribute Types

Types of Attributes

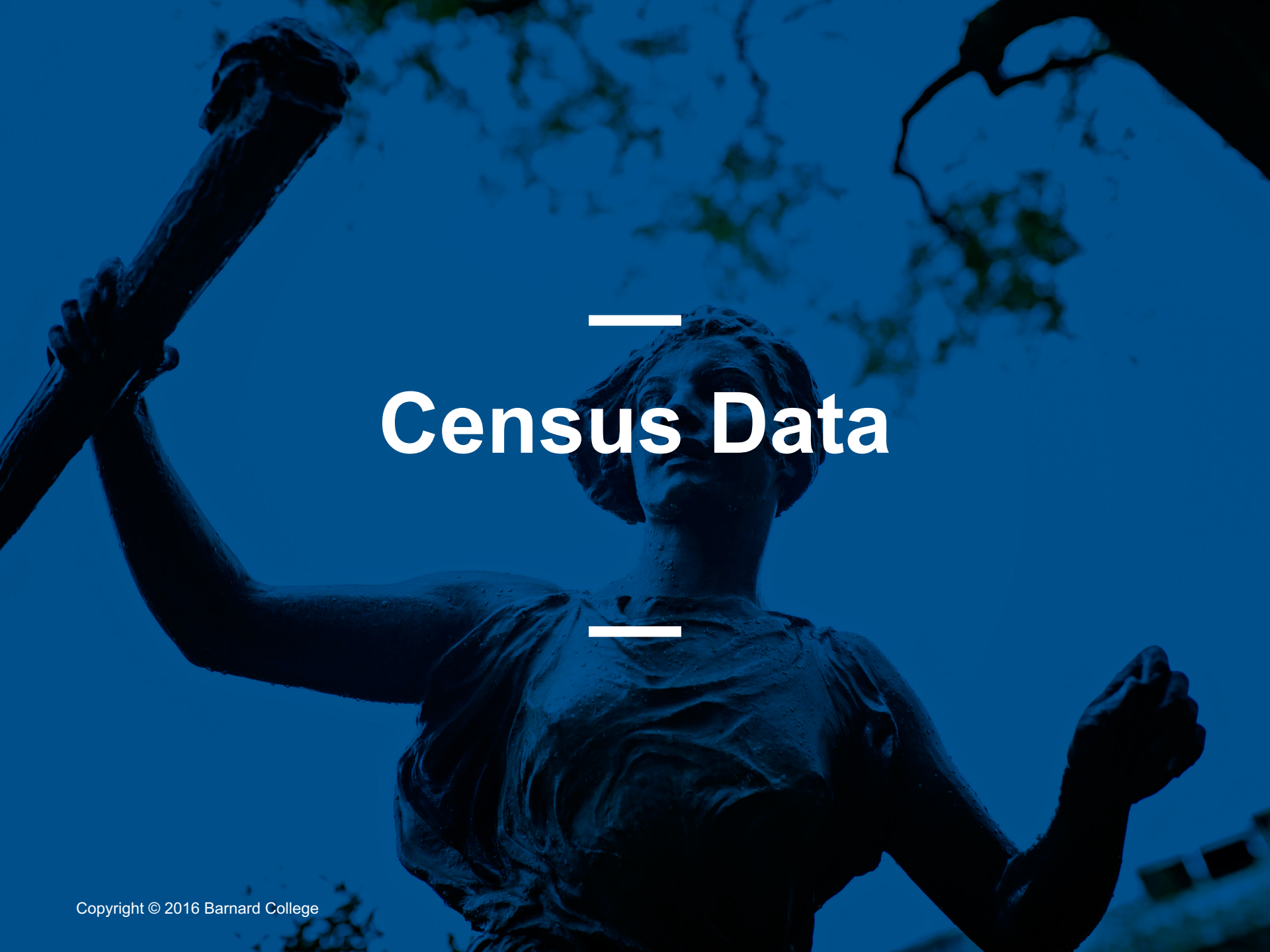
All values in a column of a table should be both the same type **and** be comparable to each other

- **Numerical** – values are from a numerical scale
  - Numerical measurements are ordered
  - Differences are meaningful
- **Categorical** – values from a fixed inventory
  - May or may not have an ordering
  - Categories are the same or different

Numerical Attributes

Values as numbers are not guaranteed to be numerical

- Census example: SEX code (0, 1, 2)
- Arithmetic on these “numbers” is meaningless
- The variable SEX is still categorical, even though numbers were used for the categories



Census Data

The Decennial Census

- Every ten years, Census Bureau counts how many people there are in the U.S.
- Census Bureau estimates how many people are in US during the other 9 years
- U.S. Constitution Article 1, Section 2:
  - “Representatives and direct Taxes shall be apportioned among the several States ... according to their respective Numbers ...”

- https://www2.census.gov/programs-surveys/popest/datasets/
- https://www2.census.gov/programs-surveys/popest/datasets/2010-2015/national/totals/
- demo