Learning goals:

- Get some tips for feasible and interesting project proposals.
- See some examples of interesting research questions.
- Pause to talk about Pandas

Projects

COGS 108 Spring 2020
Will McCarthy
Discussion 3

wmccarthy@ucsd.edu
OH: Fri 10a-11a on Zoom

Individual vs. Group

- **You should have already chosen and filled out a form (either way!)**
- **Individual: your job throughout the quarter will be to learn the concepts well enough to deploy them quickly and effectively**
- **Group: your job throughout the quarter will be to come up with interesting idea, collaborate, and produce something more in-depth than is possible in just a couple of days**

Guide for a Good Project Proposal

- Find 3 interesting datasets.
  - I suggest looking at Data is Plural.
- Come up with 3 research questions for each dataset.
- Pick one.
- Why does this work? Quantity > quality for brainstorming.

How do I pick a question?

- Ask a question that would be interesting to a friend.
- Many good questions relate two quantities that are not obviously related.
  - Boring: What’s the most common name in COGS 108?
  - Boring: Can you predict a person’s sex from their name?
  - Fun: Can you predict a person’s age from their name?
  - Fun: Can you predict a person’s sex from the last letter of their name?

Baby names demo: https://github.com/COGS108/Section-Sp20/blob/master/Will/disc03/disc03.ipynb

[We will also recap Pandas here]

(The demo is based off of https://www.textbook.ds100.org/ch/01/lifecycle_intro.html)

Example research questions from Data is Plural newsletter:

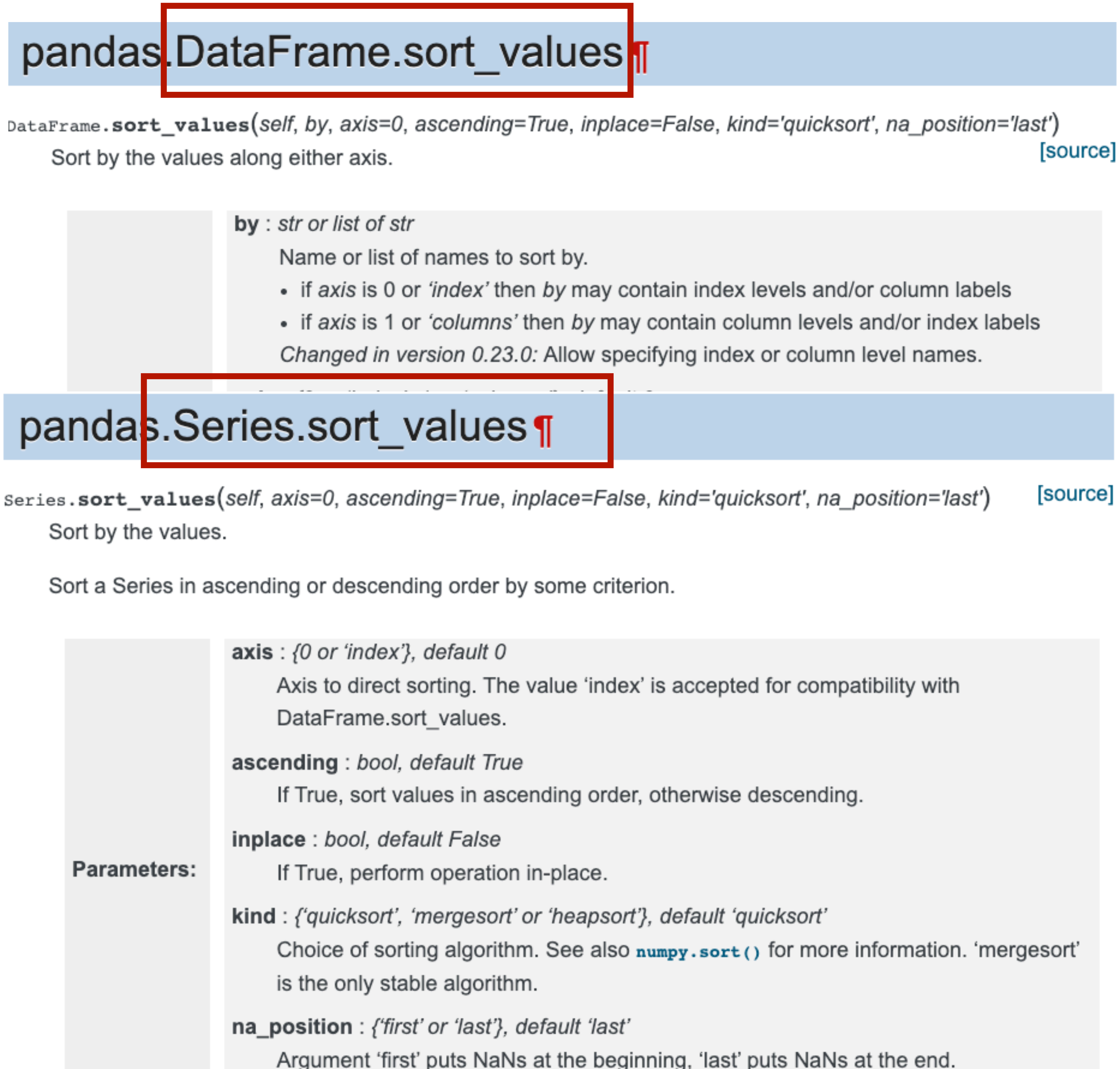
- Does China primarily loan to countries with low GDP? Or countries that are military / economic allies?
- Are there more radio stations per capita for mountainous areas?
- Do cities with more disconnected streets have worse health conditions?
- Are cannabis testing labs consistent with each other?
- Does the number of backyard ice skating rinks change with global temperature patterns?

Rest of time:
Work on project proposals/ A2.

I will *virtually* walk around and give feedback.

Preview of Next week

- Difference between pandas DataFrames and Series.
- How to use Google to solve problems on A2.
- How to read the pandas documentation.
- A2 problem walkthroughs.



pandas.DataFrame.sort_values¶

`DataFrame.sort_values`(*self*, *by*, *axis=0*, *ascending=True*, *inplace=False*, *kind='quicksort'*, *na_position='last'*) [source]

Sort by the values along either axis.

by : *str or list of str*

Name or list of names to sort by.

- if *axis* is 0 or *'index'* then *by* may contain index levels and/or column labels
- if *axis* is 1 or *'columns'* then *by* may contain column levels and/or index labels

Changed in version 0.23.0: Allow specifying index or column level names.

pandas.Series.sort_values¶

`Series.sort_values`(*self*, *axis=0*, *ascending=True*, *inplace=False*, *kind='quicksort'*, *na_position='last'*) [source]

Sort by the values.

Sort a Series in ascending or descending order by some criterion.

Parameters:

axis : *{0 or 'index'}, default 0*

Axis to direct sorting. The value 'index' is accepted for compatibility with DataFrame.sort_values.

ascending : *bool, default True*

If True, sort values in ascending order, otherwise descending.

inplace : *bool, default False*

If True, perform operation in-place.

kind : *{'quicksort', 'mergesort' or 'heapsort'}, default 'quicksort'*

Choice of sorting algorithm. See also `numpy.sort()` for more information. 'mergesort' is the only stable algorithm.

na_position : *{'first' or 'last'}, default 'last'*

Argument 'first' puts NaNs at the beginning, 'last' puts NaNs at the end.