Week 09

Time series

INFO 3402: Information Exposition

Brian Keegan, Ph.D.
Assistant Professor, Information Science
brian.keegan@colorado.edu

University of Colorado Boulder

Course Overview

Module	Week	Dates	Type	Skill
Shaping	1	Jan 11, Jan 13	Computation	Loading
	2	Jan 18, Jan 20	Computation	Aggregating
	3	Jan 25, Jan 27	Computation	Joining
	4	Feb 1, Feb 3	Computation	Tidying
Distribution	5	Feb 8, Feb 10	Computation	Histograms
	6	Feb 15, Feb 17	Communication	Audience
Comparison	7	Feb 22, Feb 24	Computation	Cat plots
	8	Mar 1, Mar 3	Communication	Persuasion
Trend	9	Mar 8, Mar 10	Computation	Time series
	10	Mar 15, Mar 17	Communication	Uncertainty
	11	Mar 22, Mar 24	Spring Break	
Relationship	12	Mar 29, Mar 31	Computation	Scatter plots
	13	Apr 5, Apr 7	Communication	Fallacies
Spatial	14	Apr 12, Apr 14	Computation	Choropleths
	15	Apr 19, Apr 21	Communication	Conventions
Projects	16	Apr 26, Apr 28	Projects	

Readings

- Questions for Friday's Weekly Quiz 09 will be drawn from these readings
  - pandas User Guide – Time Series / date functionality
    - Deep skim for overlap with content in lecture and notebook
  - Harrison & Petrou. (2020). Chapter 12: Time Series Analysis.
    - Reinforcing concepts and methods form lecture and notebook
  - Keegan, Gergle, & Contractor. (2012). "Hot off the Wiki."
    - The article that gave me my "break" as an academic and the first exploring Wikipedia's coverage of breaking news

Weekly Assignment 09

- **Skills**: Retrieving, analyzing, and visualizing time series data
- **Data**: Wikipedia revision histories and pageviews

Module Assignments 03 and 04

Module Assignment feedback

- Write for a general audience, not to me and Samantha
  - Open with a question, provocation, case …anything to engage the reader
  - Do not assume the reader knows anything about data analysis and visualization, programming, *etc*.
  - “Like you said in lecture, I did X”
- There is a place for tutorial-style writing on Medium, but that’s not what we’re focused on
  - “I used X_df and the Y method to turn this into Z which looks like this”
  - We don’t want a step-by-step narration of your analysis; we want narratives with polished results
- Quality matters
  - Use high-quality saved figures (docs) rather than screenshots whenever possible
  - No glaring spelling or grammatical problems; professional tone and style

Module Assignment 03

- Use the World Development Indicators, Democracy Index, or a similar country-level longitudinal dataset to **persuade** your readers about
  - You're welcome to bring in other data if it helps with your story
  - Emphasis on **persuasion**! Call to action, need to act, consequences of failing to act, necessity to act
  - Use WA08 to structure the flow of your argument

- 700 – 1000 words with at least one visualization (extra credit for visualizations with annotation!)

- Module Assignment 03 will be due on **Wednesday, March 9 by 11am**
  - Submit URL of your Medium post to Canvas or save and submit as an HTML file
  - Tag your post on Medium with "INFO3402S22A3" and whatever other tags you'd like

Submit MAs to Medium Publication

- Submit HTML/PDF to Canvas for grading ***AND*** submit to class publication

- To submit your post to class publication (docs)
  - Log in and write your assignment at https://medium.com/new-story
  - When you're done, click the triple dots in the toolbar
  - Select "Add to publication"
  - Select the "Information Expositions S2022" publication
  - Click "Save"

Module Assignment 04

- Use Wikipedia data to tell a story about the production and demand for information about some topic or event
  - More on incorporating **uncertainty** next week

- 700 – 1000 words with at least one visualization (extra credit for visualizations with annotation!)

- Module Assignment 04 will be due on **Wednesday, March 30 by 11am**
  - Wednesday after Spring Break
  - Submit URL of your Medium post to Canvas or save and submit as an HTML file
  - Tag your post on Medium with “INFO3402S22A4” and whatever other tags you’d like

Time series

Time is special

- Time data is all kinds of data: financial, social, meteorological, *etc*.
- Time as continuous but for data analysis it's more helpful to think of as categorical
  - Cumulative activity in a month, number of events per date, *etc*.

- Time is also a pain-in-the-ass because it has such an idiosyncratic structure
  - 60 seconds/minute, 60 minutes/hour, 24 hours/day, 28-31 days/month, leap years
  - Dates fall on different weekdays, AM/PM, timezones, leap seconds
  - Different date formats: "3-7-22" means different things in America versus rest of the world
    - March 7 2022? July 3 2022? July 22 2003?
  - ISO 8601 (YYYY-MM-DD) is the "metric system" of datetime, but not universally adopted

Representing time computationally

- Computers want to follow rules, time violates simple rules, computers represent time unusually

- The "UNIX Epoch" (or "Posix time") starts at midnight on January 1, 1970 (1970-01-01)
  - Most computational representations of time are the number of seconds or days since then

- Y2K bug: Years represented as two-digits, computers don't know to interpret "00" as 1900 or 2000
- Y22K bug: 32-bit systems using YYMMDDHHMM can't represent values greater than 2147483647
- Y38K bug: 2^{31} - 1 seconds since 1970-01-01 is 2038-01-19 03:13:07 and the 32-bit computers in everything (satellites, nuclear reactors, cars, *etc.*) will roll over to 1901-12-13 afterwards
- Other bugs in 2079, 2106, and 2262

Working with time

- There are so many edge cases and standards, you should never try to handle it yourself
  - The only thing more frustrating than dealing with time standards is character encodings

- Every language will have some kind of “datetime” support
  - Python: https://docs.python.org/3/library/datetime.html
    - numpy: https://numpy.org/doc/stable/reference/arrays.datetime.html
    - matplotlib: https://matplotlib.org/stable/api/dates_api.html
    - pandas: https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html
  - R: https://www.r-bloggers.com/2020/04/a-comprehensive-introduction-to-handling-date-time-in-r/
  - SQL: https://www.w3schools.com/sql/sql_dates.asp
  - These will all prefer/default to ISO 8601 style of YYYY-MM-DD (as they should!)

Time objects

- datetime or Timestamp for a single point in time (YYYY-MM-DD HH:MM:SS)

- Time formatting codes from the C standard
  - %m = zero-padded month MM vs %M = zero-padded minute
  - %B = Full month name *vs*. %b = Abbreviated month name
  - %Y = four-digit year YYYY *vs*. %y = two-digit year YY

- Python will not convert string or float representations of time unless you tell it
  - Datetime's `strptime` converts a string to a datetime; `strftime` converts a datetime to a string
  - pandas's `to_datetime` will auto-magically convert most strings to a Timestamp
  - From Timestamp back to datetime with `.to_pydatetime()` method
  - Epoch time to Timestamp with "unit" parameter in `to_datetime`

Notebook time!

- Download
  - “Week 09 – Lecture.ipynb” → lecture notebook
  - “sp500.csv” → dataset to play with
  - “wikifunctions.py” → scripts for demo and Weekly Assignment 09

Next class

Next Class

- Review concepts and exercises from last class
  - Complete “Thursday Questions” form! https://forms.gle/1Ji4ZkwXVzvcH8wA6 (ungraded/optional)
- Critique a data narrative or visualization
  - Share a link & a few sentences about a data story or visualization (ungraded/optional)
- Time to work on Weekly Assignment
  - Weekly Assignment due on Sunday by midnight by submitting HTML notebook to Canvas
- Weekly quiz at the end of class (12:00–12:30)