class wrap up

adam okulicz-kozaryn

adam.okulicz.kozaryn@gmail.com

this version: Thursday 24th April, 2025 08:55

outline

ps5

- do not overcomplicate!
  - better to have simple clean code that does the job
  - than messy complex fancy code that is wrong
- if chunks of code take long time to run, say $> 1min$
  - optimize it, take random sample, etc
- again, it always must start from the very raw data!
- easy to make mistake:
  - think about it AND cross check
  - ideally present to others, eg descriptive stats
  - correctness is important!

ps5

- explain what you are doing!
- interpret things!
  - eg when you run descriptive stats, *and* find sth interesting, put a comment and say what you have found in few words
  - (don't comment output of every command)

ps5

- google things!
- before writing the code check if someone already wrote it
- and build on others work! ie copy and adapt and improve
- eg googling "python notebook basketball analysis" yields:
  `https://www.google.com/search?client=firefox-b-1-d&q=python+notebook+basketball+analysis`
  - first couple hits look good

get into flow with programming!

https://en.wikipedia.org/wiki/Flow_(psychology)

this is super important! remember this!!

- publishing (and maybe conferences) is *the only way* to get in touch with academics/experts exactly in your area
- there's just a handful of them,
  - almost never at you university, sometimes at a conference
  - usually at a journal where you submit; (if you pick the right one, almost always at a journal)
- this is *the only way* to take your work to next level!!
  - it does take time; start now; otherwise you may never make it
  - start simple, even just some des stats...but keep on submitting papers
- or even just put online: arxiv, ssrn, etc

likewise for non-academia: for-profit and non-profits

- there are also non-academic experts, practicioners
  - people who actually do things outside of the ivory tower
  - often the applied/real knowledge is better than theoretical/academic knowledge
- may try to get in touch with people who do similar work/analysis
- again, first step is just to google what youare doing with keywords 'visualization' 'python' etc, and look at code and images; like lit rev in academia

in general: make it public, show to stakeholders

- the worst thing you can do is to keep it in a drawer
- when you share it (locally/globally)
  - get ideas and directions
  - become part of decision making
  - find mistakes and misconceptions

    eg i came to nj from tx and knew nothing about nj
  - and i'm presenting to like 100 new jerseyans
  - saying Cape May highest alcohol consumption
  - someone gets up and says no, its few older folks live there
  - but youngsters from elsewhere coming and drinking
  - so liquor store per capita is high but not because locals drink

protect your organization

- just remember (rightfully so) each organization is scared to get hit on the head with their own data
  - so they're scared to share data and make it public
  - so make sure you'll deidentify it! and maybe fake it too! say on github your org is in chickasaw county mississippi!
  - and do not share any org specific info
  - in addition to deidentyfying like dropping geo locations, may take subsample (say male only or 35+ only)

GIGO: dont trust anybody! esp ur org

- say if you have data from census, many people use and probably found most mistakes and fixed it
- but your organization's data–probably nobody is looking at these data or very few people
- so almost for sure there are many mistakes and problems
- eg just mistake-mistake age of 20 miscoded as 200 or zip 08102 coded as 8102
- or problems: data not representative, missing data, etc etc
- in addition to vis do:
  - info()
  - value_counts(dropna=False)

future research

- you've probably realized that i am into
  - Python and data
- and always happy to talk more
- keep in touch!

what next?

- make use of your data management skills
- use it or lose it
- apply for postdocs
- collaborate with faculty
  - faculty need your data management skills
  - they know much less about data mgmt than you!
  - make $!

make $

- industry data jobs usually require SAS, SQL, Python, Java
- a ton of data science jobs:
  - `http://www.icrunchdata.com/`
  - `http://www.cybercoders.com/`

see again

- theory.pdf
- intro slides