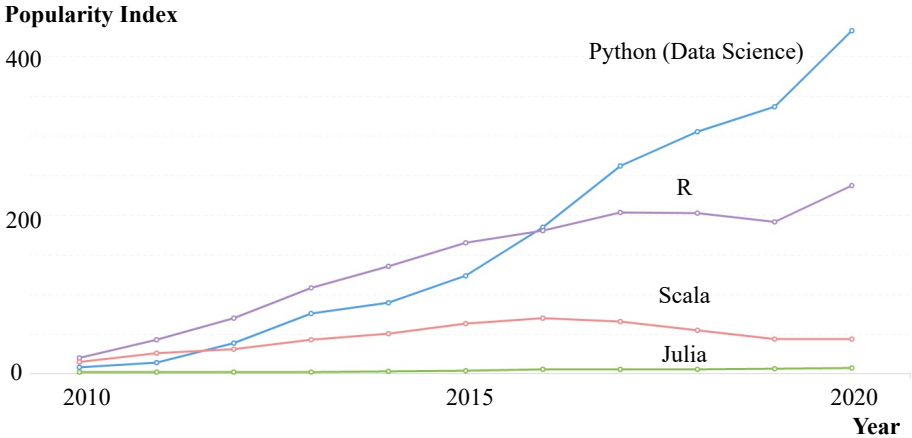
Topics on R

Konan Hara

University of Arizona

September 8, 2021

Why Use R?—Popularity in Data Science



- Popularity index is calculated based on # questions asked daily, # daily distinct users, and # view counts of questions in Stack Overflow. [Ref]
- Proprietary languages like Stata, Matlab, and SAS do not appear here.

Why Use R?—R vs. Python

- General idea:
  - R's functionality was developed with statisticians in mind
  - Python is often praised for being a general-purpose language with an easy-to-understand syntax
- Factors that may affect your decision:
  1. Which language do your colleagues use?
  2. What problems do you want to solve and what tasks do you need to accomplish?
  3. What are the net costs of learning a language?
  4. What are the commonly used tool(s) in your field?

Why Use R?—R vs. Python

- R advantages:
  - R is easier to learn if you have no coding experience.
  - Widely considered the best tool for making beautiful graphs and visualizations.
  - Has many functionalities for data/statistical analysis
  - Statistical models can be written with only a few lines.
- Python advantages:
  - General-purpose programming languages are useful beyond just data analysis.
  - Python's focus on readability and simplicity means its learning curve is relatively linear and smooth.
  - Great for mathematical computation and learning how algorithms work.

Why Use R?—R vs. Python

- R disadvantages:
  - Finding the right packages to use in R may be time consuming.
  - R can be considered slow if code is written poorly.
  - Not as popular as Python for deep learning and NLP.
- Python disadvantages:
  - Python doesn't have as many libraries for data science as R.
  - Visualizations are more convoluted in Python than in R, and results are not as eye-pleasing or informative.

Good Programming Habits

A good code is:

- Easy to maintain
- Easy to extend
- Easy to understand...even after a six month break!
- Straight-forward and direct...no side-effects or surprises!
- Reads like English (or some other human language)

Good Programming Habits

- Naming of functions, variables, and filenames: e.g.,
  - Begin or end function names with a verb.
  - Separate each word by CamelCase, '_', or '.'.
  - Examples, CalcValueFunc; calc_value_func.
- Comments:
  - Write why you did something rather than what you did.
  - One variable definition per line.
- Respect the local coding convention when working on code.
  - E.g., Google's R Style Guide
- Advanced habits:
  - Code publication
  - README file
  - Modification history
  - Reproducible research

Received: 4 January 2021 | Revised: 3 August 2021 | Accepted: 5 August 2021

DOI: 10.1002/sim.9169

FEATURED ARTICLE

Statistics in Medicine WILEY

Best practices in statistical computing

Ricardo Sanchez[1] | **Beth Ann Griffin**[2] | **Joseph Pane**[3] | **Daniel F. McCaffrey**[4]

[1]UnitedHealthcare, Minnetonka, Minnesota, USA

[2] RAND Corporation, Arlington, Virginia, USA

[3]RAND Corporation, Pittsburgh, Pennsylvania, USA

[4]Educational Testing Service, Princeton, New Jersey, USA

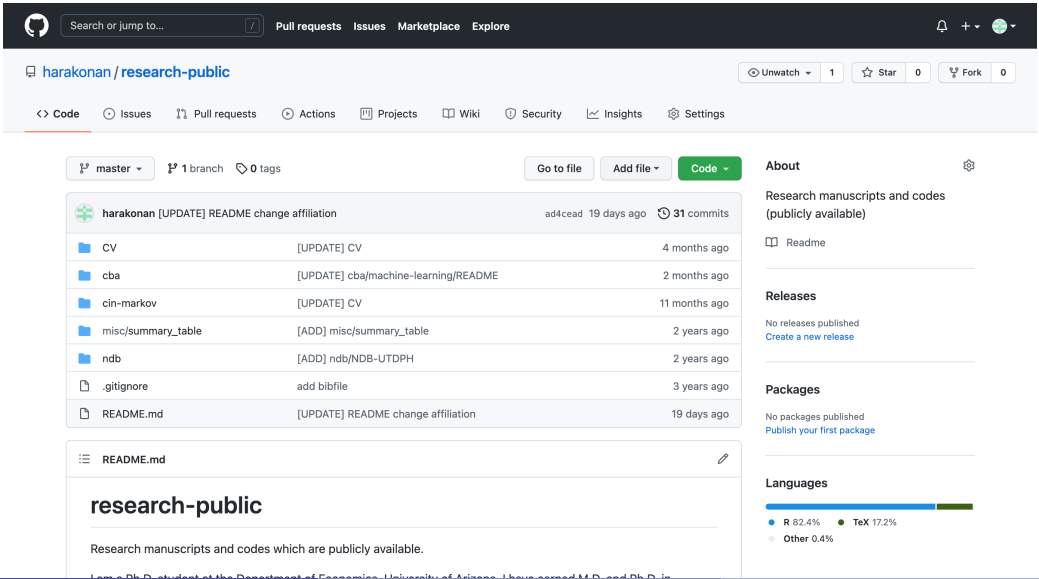
Correspondence
Beth Ann Griffin, RAND Corporation, Arlington, VA 22202, USA.
Email: bethg@rand.org

Funding information
National Institutes of Health, Grant/Award Number: R01DA045049

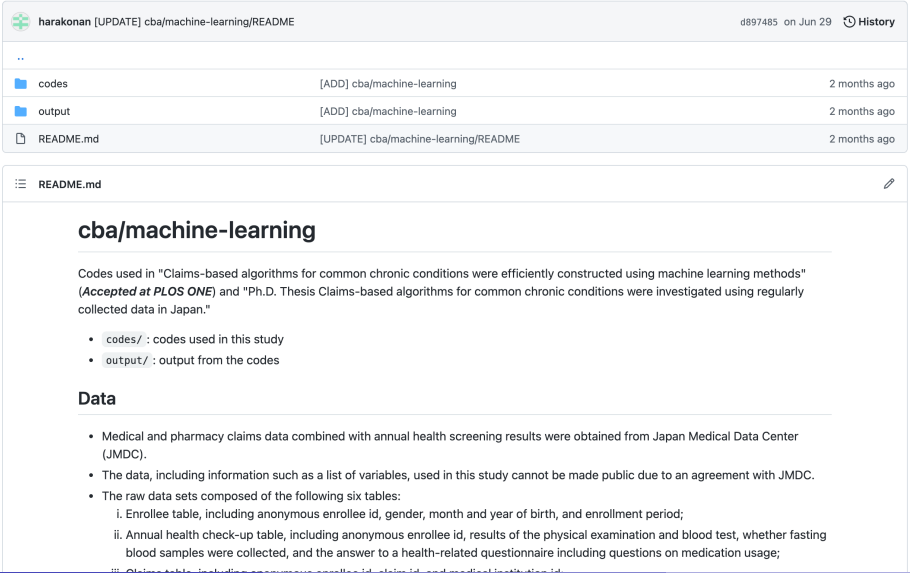
The world is becoming increasingly complex, both in terms of the rich sources of data we have access to and the statistical and computational methods we can use on data. These factors create an ever-increasing risk for errors in code and the sensitivity of findings to data preparation and the execution of complex statistical and computing methods. The consequences of coding and data mistakes can be substantial. In this paper, we describe the key steps for implementing a code quality assurance (QA) process that researchers can follow to improve their coding practices throughout a project to assure the quality of the final data, code, analyses, and results. These steps include: (i) adherence to principles for code writing and style that follow best practices; (ii) clear written documentation that describes code, workflow, and key analytic decisions; (iii) careful version control; (iv) good data management; and (v) regular testing and review. Following these steps will greatly improve the ability of a study to assure results are accurate and reproducible. The responsibility for code QA falls not only on individual researchers but institutions, journals, and funding agencies as well.

KEYWORDS

GitHub Can Help You: Code Publication



GitHub Can Help You: README File



harakonan [UPDATE] cba/machine-learning/README d897485 on Jun 29 History

..

codes	[ADD] cba/machine-learning	2 months ago
output	[ADD] cba/machine-learning	2 months ago
README.md	[UPDATE] cba/machine-learning/README	2 months ago

README.md

cba/machine-learning

Codes used in "Claims-based algorithms for common chronic conditions were efficiently constructed using machine learning methods" (***Accepted at PLOS ONE***) and "Ph.D. Thesis Claims-based algorithms for common chronic conditions were investigated using regularly collected data in Japan."

- `codes/` : codes used in this study
- `output/` : output from the codes

Data

- Medical and pharmacy claims data combined with annual health screening results were obtained from Japan Medical Data Center (JMDC).
- The data, including information such as a list of variables, used in this study cannot be made public due to an agreement with JMDC.
- The raw data sets composed of the following six tables:
  i. Enrollee table, including anonymous enrollee id, gender, month and year of birth, and enrollment period;
  ii. Annual health check-up table, including anonymous enrollee id, results of the physical examination and blood test, whether fasting blood samples were collected, and the answer to a health-related questionnaire including questions on medication usage;

GitHub Can Help You: README File

README.md

Codes

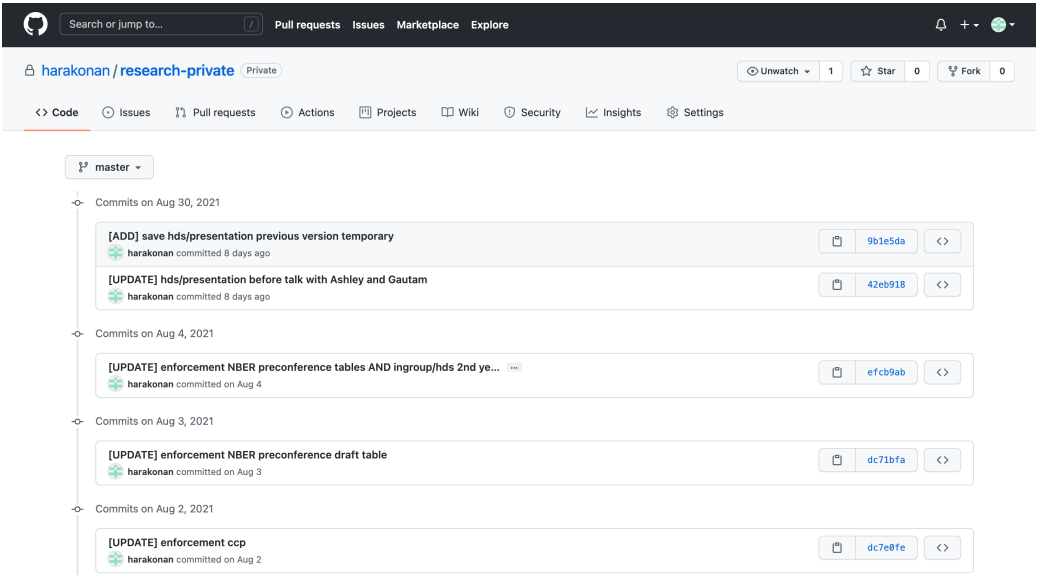
- Batch files
  - `batch.R`
    - Batch file for data cleaning and analysis
  - `batch_figtab.R`
    - Batch file for generating figures and tables
  - Variables
    - `set_env` : The codes were tested in the local environment and then executed on the production server. The simulated dataset in the test environment had a slightly different data structure than the production server dataset, so `set_env` switches between the test and production environments.
    - `sample_ratio` : A variable that allows for random sampling of a portion of the whole sample to test the code.
    - `target_disease` : A variable that specifies the target disease --- ht, hypertension; dm, diabetes; dl, dyslipidemia.
    - `full_data` : Due to the high computational burden of the machine learning analysis, the analysis with the whole data is performed with caution. `full_data` alerts us to execute the codes with the whole data after we have confirmed that the codes execute successfully with a portion of the whole data.
- Data cleaning
  - `cba_data_cleaning.R`
    - Column selection + data cleaning + sample selection
    - `cba_data_cleaning_test.R` is the codes for the test environment.
    - Because of the large data size of the raw data, column selection + data cleaning are crucial.
  - `cba_data_man_stat.R`
    - Create a dataset with variables for machine learning CBAs and gold standards.
    - Use the full collection of ICD-10 codes and ATC codes.
    - The unit of ICD-10 codes and ATC codes are an alphabet followed by two digits.
  - `cba_data_man_conv.R`
    - Create a dataset with variables for conventional CBAs and gold standards.

GitHub Can Help You: README File

README.md

```
# cba/machine-learning
Codes used in "Claims-based algorithms for common chronic conditions were efficiently constructed using machine learning methods" (***Accepted at PLOS ONE***) and "Ph.D. Thesis Claims-based algorithms for common chronic conditions were investigated using regularly collected data in Japan."

- `codes/`: codes used in this study
- `output/`: output from the codes

## Data

- Medical and pharmacy claims data combined with annual health screening results were obtained from Japan Medical Data Center (JMDC).
- The data, including information such as a list of variables, used in this study cannot be made public due to an agreement with JMDC.
- The raw data sets composed of the following six tables:
    1. Enrollee table, including anonymous enrollee id, gender, month and year of birth, and enrollment period;
    1. Annual health check-up table, including anonymous enrollee id, results of the physical examination and blood test, whether fasting blood samples were collected, and the answer to a health-related questionnaire including questions on medication usage;
    1. Claims table, including anonymous enrollee id, claim id, and medical institution id;
    1. Medical institution table, including medical institution id and specialty;
```

GitHub Can Help You: Modification History



GitHub Can Help You: Modification History

```
@@ -34,36 +34,42 @@ source(paste0(pathtotools,"create_table_ccp_tobit.R"))
34  34    analysis_cid_quarterly <- fread(paste0(pathtointdata,"analysis_cid_quarterly.csv"))
35  35
36  36    # use high capacity facilities
37      - cid_n <- length(unique(analysis_cid_quarterly[capacity > 100]$cid))
        37  + analysis_cid_quarterly <- analysis_cid_quarterly[capacity > 50]
        38  + cid_n <- length(unique(analysis_cid_quarterly$cid))
38  39    cid_n
39      - analysis_cid_quarterly <- analysis_cid_quarterly[capacity > 100]
        40  +
40  41
41  42    # rescale some variables
42  43    analysis_cid_quarterly[, capacity := capacity/100]
43  44    analysis_cid_quarterly[, total := total/1000]
44  45
45      - # factor variables
46      - analysis_cid_quarterly[, eparegion_f := as.factor(eparegion_f)]
47      - analysis_cid_quarterly[, year_f := as.factor(year_f)]
48      - analysis_cid_quarterly[, qtr_f := as.factor(qtr_f)]
        46  + # standardize race variables for interpretation
        47  + analysis_cid_quarterly[, black := (black - mean(black))/sd(black)]
        48  + analysis_cid_quarterly[, hispanic := (hispanic - mean(hispanic))/sd(hispanic)]
        49  +
```

Reproducible Research Using R: R Markdown

```
\section*{read tables}

<<>>=

# cems
cems_annual_unit_char <- as.data.table(read.dta13(paste0(pathtorawdata
        , "CEMS_fromLouisPreonas/cems_annual_unit_char.dta")))

# icis
icis_air_facilities <- fread(paste0(pathtorawdata
        , "ICIS-AIR_downloads/ICIS-AIR_FACILITIES.csv"))

# cems-icis crosswalk
icis_cems_cw <- fread(paste0(pathtointdata
        ,"icis_cems_cw.csv"))

@

\section*{objective 1: merge icis and cems-icis crosswalk}

<<>>=
# preliminaries for icis_air_facilities
# set common name for program ID "PGM_SYS_ID"
setnames(icis_air_facilities,"PGM_SYS_ID","PGM_SYS_ID_AIR")
# create 2-digit NAICS_CODES
icis_air_facilities[, NAICS2d := substr(NAICS_CODES, 1, 2)]
```

Reproducible Research Using R: R Markdown

Data check for merging ICIS and CEMS

Konan Hara*

March 24, 2021

read tables

```
# cems
cems_annual_unit_char <- as.data.table(read.dta13(paste0(pathtorawdata
        , "CEMS_fromLouisPreonas/cems_annual_unit_char.dta")))

# icis
icis_air_facilities <- fread(paste0(pathtorawdata
        , "ICIS-AIR_downloads/ICIS-AIR_FACILITIES.csv"))

# cems-icis crosswalk
icis_cems_cw <- fread(paste0(pathtointdata
        ,"icis_cems_cw.csv"))
```

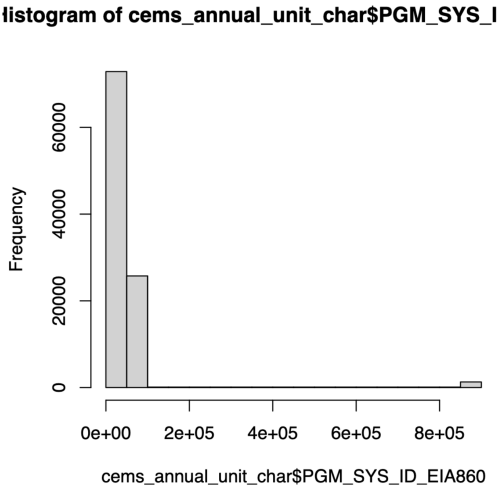
objective 1: merge icis and cems-icis crosswalk

```
# preliminaries for icis air facilities
```

Reproducible Research Using R: R Markdown

```
# so, they are m:m match
icis_cems_cw[, dup := NULL]

# check icis/crosswalk merge proportion
cw_icis_flag <- unique.data.frame(icis_cems_cw[,.(PGM_SYS_ID_AIR)])
icisfac_cw_check <- merge(icis_air_facilities, cw_icis_flag, by = "PGM_SYS_ID_AIR")
# proportion merged
icisfac_cw_check[,.N]/icis_air_facilities[,.N]

## [1] 0.01233558

# #s
icisfac_cw_check[,.N]

## [1] 3166

icis_air_facilities[,.N]

## [1] 256656

# dist. of 2-digit NAICS
# before merge
count_prop(icis_air_facilities,"NAICS2d")

##     NAICS2d     N prop
##  1:      21 59123 23.0
##  2:      32 35291 13.7
##  3:      33 30487 11.8
```

Reproducible Research Using R: R Markdown




```
# proportion
cems_annual_unit_char[, sum(PGM_SYS_ID_EIA860 > 800000)/.N]
```

Programming Habits: Demonstration

1. Change codes in a R markdown file
2. Execute them to check whether they work
3. Compile the R markdown file
4. Track changes using GitHub

Example R Codes

Topics_on_R_example_codes.pdf demonstrates some R codes examples:

1. Getting Help
2. R Objects
3. Loops
4. Regressions
5. Plots
6. User-defined Functions and Optimization

Import a Dataset

- Csv file
  - data.frame way: read.csv()
  - data.table way: fread()
  - tibble way: read_csv()
- Other file formats (using haven package)
  - SAS: read_sas()
  - SPSS: read_sav()
  - Stata: read_dta()

Lots of Others Things With R

- Potential topics
  - Web scraping
  - Natural language processing
  - Image recognition
- Learning plantforms:
  - Coursera
  - Datacamp