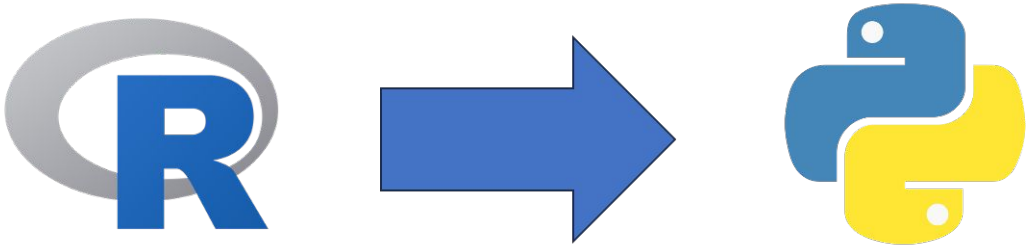


Translating R Packages into Python:

A Pilot Project of *epiDisplay*

Anna Kroening, Cat Kim, Jiayi Ding,

Marthin Mandig, Suphanat Wongsanuphat

Background and Objectives

epiDisplay: R package for epidemiological data exploration and visualization

- Problem: epiDisplay does not exist in **Python**, which has **limited** epidemiological data analysis packages

Objectives

1. **Translate the epiDisplay R package into Python:** including understanding differences in syntax, data structures, and library ecosystems between R and Python.
2. **Develop a Python package:** translates core functionalities of the R package *EpiDisplay*

Publicly available documentation:

Package ‘epiDisplay’

July 22, 2025

Version 3.5.0.2

Date 2022-05-09

Title Epidemiological Data Display Package

Author Virasakdi Chongsuvivatwong <cvirasak@medicine.psu.ac.th>

Maintainer Virasakdi Chongsuvivatwong <cvirasak@medicine.psu.ac.th>

Depends R (>= 2.6.2), foreign, survival, MASS, nnet

Description Package for data exploration and result presentation. Full 'epicalc' package with data management functions is available at '<https://medipe.psu.ac.th/epicalc/>'.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2022-05-18 14:20:02 UTC

Contents

Primary Working Dataset: ***Outbreak***

AN UNUSUAL OUTBREAK OF FOOD POISONING

Lakkana Thaikruea[1], Junya Pataraarechachai[2], Pathom Savanpunyalert[1] and Ubonrat Naluponjiragul[3]

[1]Field Epidemiology Training Program, Division of Epidemiology, Ministry of Public Health, Bangkok; [2]Faculty of Tropical Medicine, Mahidol University Bangkok; [3]Suphun Buri Provincial Health Office, Ministry of Public Health, Bangkok, Thailand

Abstract. On August 25 1990, over 400 people who attended a Thailand handicappeds' sport day at a provincial physical education college developed gastrointestinal symptoms after having dinner. An epidemiological team want to determine cause(s) and recommend how to prevent and control a food poisoning outbreak. The investigation included interviewing all 1,210 persons who attended the sport's day. In addition, an environmental survey, laboratory analysis of food samples, and rectal, ear, throat and nasal swabs from foodhandlers were also performed. A case was defined as a person who ate any items of dinner food and experienced vomiting, nausea, abdominal pain, and diarrhea. There were 485 cases out of 1,094 persons, an attack rate of 43%. Interviews were completed for 470 out of 485 cases. The three most common symptoms were nausea (93%), vomiting (88%), and abdominal pain (81.5%). The mean incubation period was 3.20 hours. Three out of four items of food had a significant association with illness. Among these 3 items, eclairs had the highest crude relative risk, 7.0 (95% CI = 4.8, 10.2). For statistical analysis, logistic regression by unconditional method was used, and found that only eclairs which were prepared during the night before the dinner and kept at room temperature for at least 12 hours before serving, remained statistically significant in the model (RR = 11.96; 95% CI = 9-22). Laboratory examination of foods and foodhandlers indicated heavy growth of *Staphylococcus aureus* producing toxins A and C and *Bacillus cereus* in eclairs. Culture of nasal swabs from healthy foodhandlers identified *B. cereus* and *S. aureus* of different phage types from those in eclairs. The incubation period, symptoms, and the laboratory results suggested that enterotoxin produced by *S. aureus* or *B. cereus* was the most likely responsible agent for this outbreak caused by improper food handling practices.

Thaikruea L, Pataraarechachai J, Savanpunyalert P, Naluponjiragul U. An unusual outbreak of food poisoning. Southeast Asian J Trop Med Public Health. 1995;26(1):78-85.

User Story:

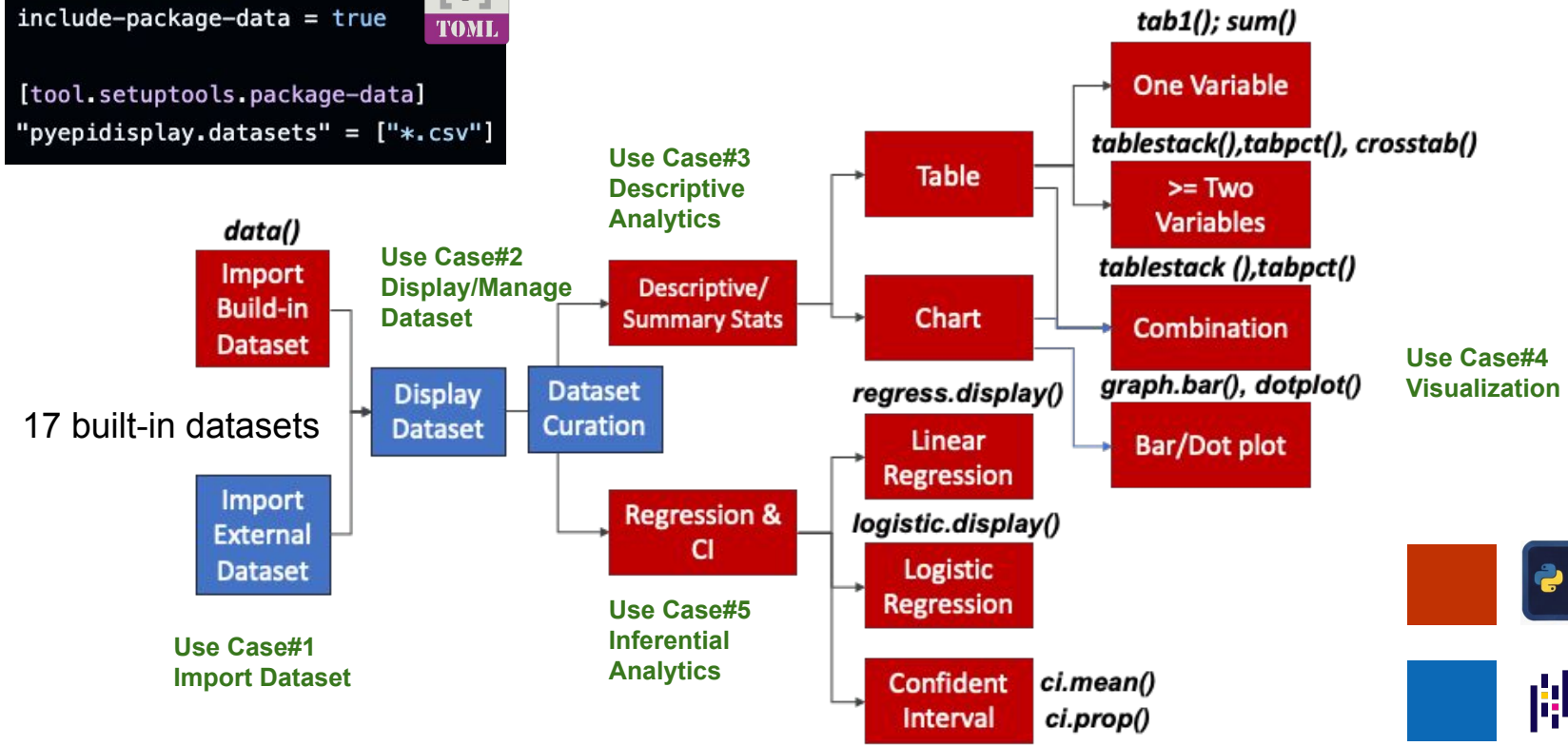
- Henry = epidemiology M.S. student with no prior background in R but has lots of experience in Python.
  - Priorities:
    - Become more familiar with R for epidemiology applications
    - Wants a python version of epiDisplay that is intuitive, easy to use, as well as easy to understand the documentation of as he still wants to gain exposure in R.
- James = software developer and wants to use functions for R but only works in Python.
  - Priorities:
    - Doesn't care too much about the epi data, just wants to add some of the functions in converting R to python in his packages.
    - Doesn't have much time to learn it.

User Case:

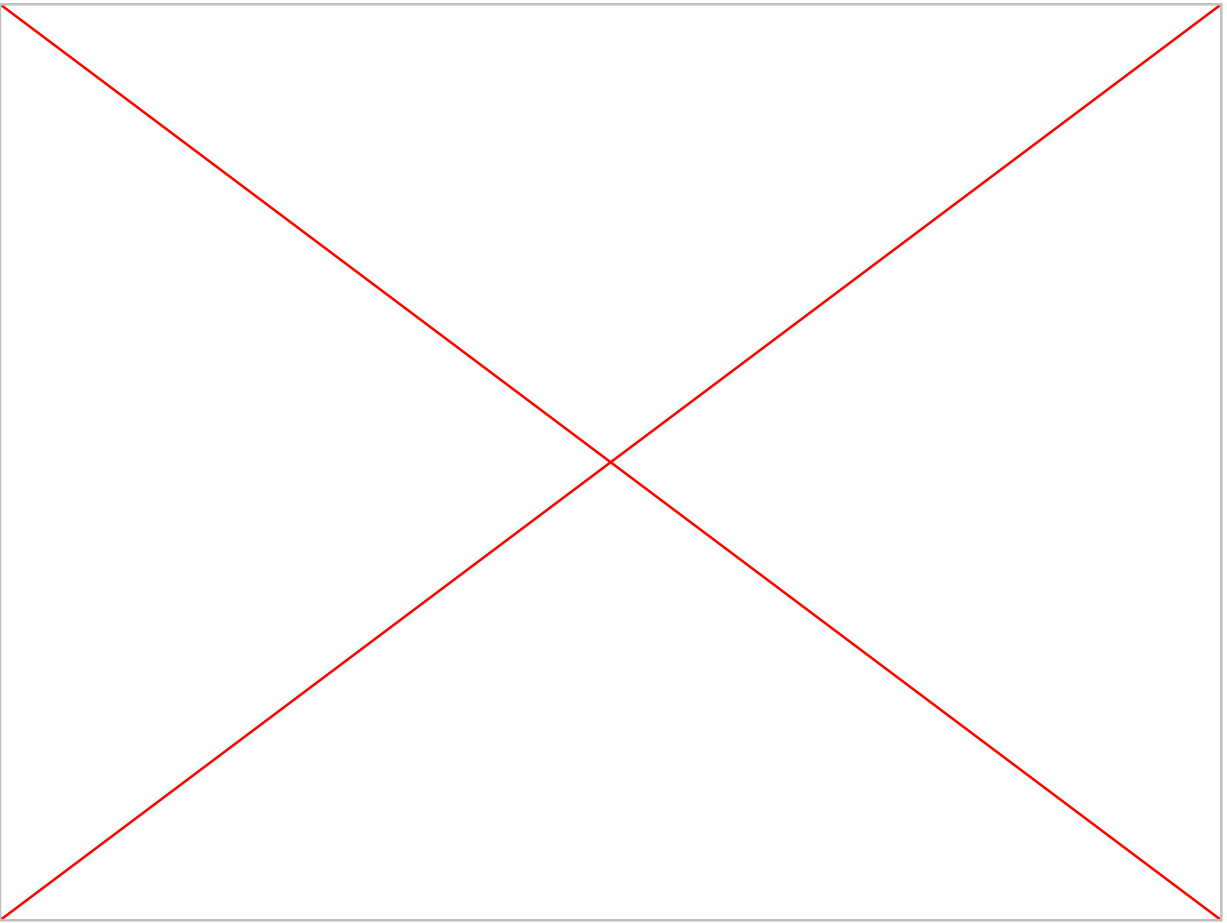
1. Import external dataset
2. Import build-in dataset
3. Display dataset and Pivot table with multiple variables
4. Descriptive analytics
   a. Frequency, Count, Sum
   b. Central of Tendency: Mean, Median
   c. Measure of Variation: SD, IQR, …
   d. Check for outliers
5. Data Visualization
   e. Exploratory Data Visualization
   f. Publication/ Report Data Visualization
6. Inferential Analytics
   g. Logistic regression with OR and 95% confidence interval
   h. Linear regression

Design: Use Cases and Components

```
[tool.setuptools]
include-package-data = true

[tool.setuptools.package-data]
"pyepidisplay.datasets" = ["*.csv"]
```



Demo & Repo Structure



```
.
├── docs
│   ├── component_specification.md
│   └── functional_specification.md
├── environment.yml
├── examples
│   ├── example_ci_mean.py
│   ├── example_ci_prop.py
│   ├── ……
│   ├── ……
│   └── example_tabpct.py
├── LICENSE
├── pyproject.toml
├── README.md
├── src
│   └── pyepidisplay
│       ├── __init__.py
│       ├── ci_mean.py
│       ├── ci_prop.py
│       ├── …….
│       ├── …….
│       └── tabpct.py
├── tests
│   ├── test_ci_mean.py
│   ├── test_ci_prop.py
│   ├── ……
│   ├── ……
│   └── test_tabpct.py
└── tree_structure.txt

6 directories, 43 files
```

Testing: Pattern Test

pyEpi Display

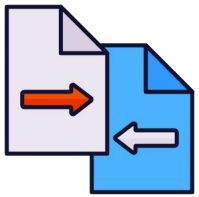
pyEpiDisplay

epiDisplay

```
============ Python RESULT ============

                            0              1              9          Test stat.  P value
Total                      91            998              5
sex                                                                Kruskal-Wallis test  < 0.001
  Median (IQR)  1.0 (0.0, 1.0)  1.0 (0.0, 1.0)  0.0 (0.0, 0.0)
nausea                                                             Kruskal-Wallis test  < 0.001
  Median (IQR)  0.0 (0.0, 0.0)  0.0 (0.0, 1.0)  0.0 (0.0, 0.0)

============ R RESULT ============

             0       1       9       Test stat.          P value
Total        91      998     5

sex                                  Kruskal-Wallis test < 0.001
  median(IQR) 1 (0,1) 1 (0,1) 0 (0,0)

nausea                               Kruskal-Wallis test < 0.001
  median(IQR) 0 (0,0) 0 (0,1) 0 (0,0)
```

Same but not exactly matched

MATCH: False



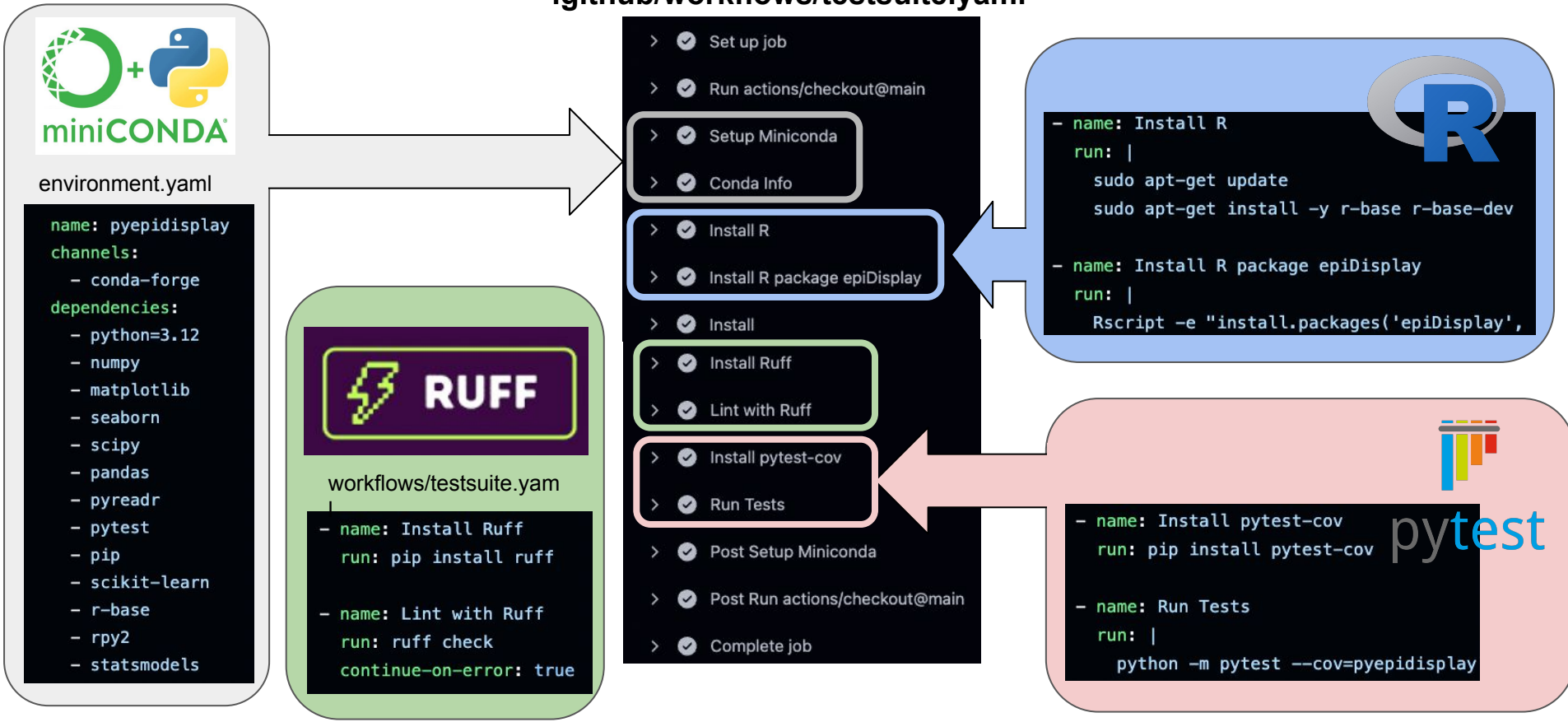
Normalize Func

```python
def normalize(s):
    s = str(s).lower()              # all lowercase
    s = s.replace(" ", "")          # remove spaces
    s = s.replace("\n", "")         # remove newlines
    s = s.replace(".0", "")         # remove trailing .0
    return s
match = normalize(py_output) == normalize(r_output)
```

MATCH: True

Iterative cross-check with all possible parameters

Continuous Integration: Ruff and Pytest-cov



Lessons Learned and Future Work

Lessons learned

- Separate branches help avoid merge conflicts.
- Commit small, frequent changes to ensure each step is accounted for.
- Keep functions short - long functions became hard to debug quickly.
- Set up continuous integration earlier.
- Use a minimal set of well-maintained dependencies whenever possible.
- Pattern testing is a powerful approach for translating code from one language (R) to another (Python).

Future work

- Implement remaining epiDisplay functions
- Add broader test coverage for edge cases: small samples, all-missing vectors, rare categories.
- Add more automatic checks so fewer bugs slip through.
- Make it easier to extend the package later without breaking things
- Ongoing maintenance to maintain compatibility with the latest versions of external dependencies.
- Add more components that do not exist in the original epiDisplay, focusing on distribution by Time (e.g., epidemic curve), Place (e.g., epidemic map), and Person.