# Why I Use Python
## (and Why You Should Too)

Paul Zivich

Institute of Global Health and Infectious Diseases
Causal Inference Research Laboratory
University of North Carolina at Chapel Hill

August 25, 2022

# Acknowledgements

Supported by NIH T32-AI007001.[1]

Python: 

 pzivich@unc.edu  @PausalZ  pzivich

Slides and code at https://github.com/pzivich/Presentations

[1] Footnotes are reserved asides for possible later discussion or questions

# Outline

My background

Value add of  [2]

Illustrative applications

Installation and Conclusions

[2] I am going to pick on R, please save angry emails till after the presentation

# About me

An epidemiologist working in methods and infectious diseases

Using  since 2016

- Largely self-taught

Active contributor

- `zEpid`, `delicatessen`[3], `MossSpider`[4]
- `lifelines`

 is my primary software

- Also use R, SAS

---

[3] Zivich PN, et al. (2022) Delicatessen: M-Estimation in Python. *arXiv:2203.11300*

[4] Zivich PN, et al. (2022) Targeted maximum likelihood estimation of causal effects with interference: A simulation study. *Statistics in Medicine*

# A Software Philosophy

To be a good epidemiologist / biostatistician / data scientist / someone who works with data, familiarity with multiple software languages is important

- No software is complete for all tasks
- *Lingua franca* of fields will change
- Harder to be replaced

Why  should be added to your repertoire

# What is  ?

High-level programming language

- Interpreted
- Object-oriented
- Free, open-source
- Supported for all major platforms
- Scales to available hardware

Some advantages from my perspective

- Language features
- Cross-software interactions
- Popularity

# Advantage: language-specific features

Class objects

Namespaces and modules

Readability

Accuracy

# Class objects

Object that hold

- Functions, other objects
- Each function can have unique parameters and docs
- Store hidden parameters for testing

```
class Frame:
    def __init__(self, title):
        doc.append(NoEscape(r'\begin{frame}{'+str(
                                        title)+'}'))

    def append(self, text):
        doc.append(text)
```

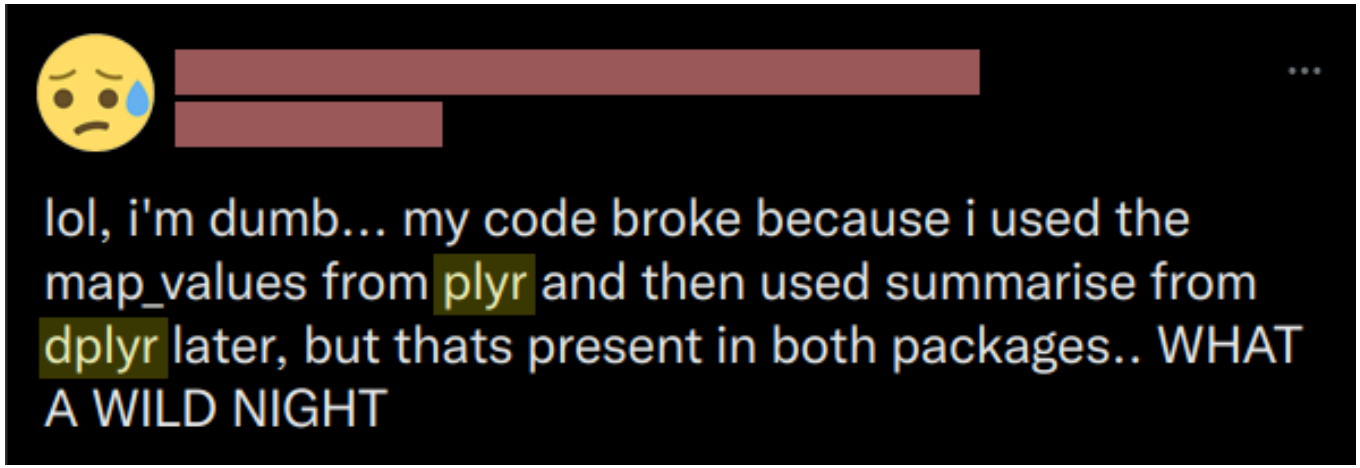
# Class objects

```r
library(tmle)

result <- tmle(Y = d$y,
               A = d$a,
               W = select(d, x, z),
               gform = "a ~ x + z + x:z"
               Qform = "y ~ a + x + z + a:x",
               family = "binomial",
               gbound = c(0.01, 0.99))
```

# Class objects

```
# Targeted Maximum Likelihood Estimation
from zepid.causal.doublyrobust import TMLE

tmle = TMLE(d, exposure="a", outcome="y")
tmle.exposure_model("x + z + x:z", bound=0.01)
tmle.outcome_model("a + x + z + a:x")
tmle.fit()
tmle.summary(decimal=2)
```

# Namespace of modules




R's namespace conflicts in my work

- Network analysis in R: `sna`, `igraph`
- Non-overlapping functionalities
- Overlapping functionalities have conflicts

# Namespace of modules

Not a problem in 

```
import math
import numpy

math.sqrt(25)
numpy.sqrt(25)
```

# Readability

```
cost = 0.
grocery_items = c("apple", "celery", "bread")
sale = FALSE

for (i in grocery_items){
    if (i == "apple"){
        cost = cost + 1.50
    }
        if (i == "celery"){
      cost = cost + 3.50}
    if (i == "bread"){
if (sale){
    cost = cost + (2.50*0.9)
                }
        else{
            cost = cost + 2.50}}
}
```

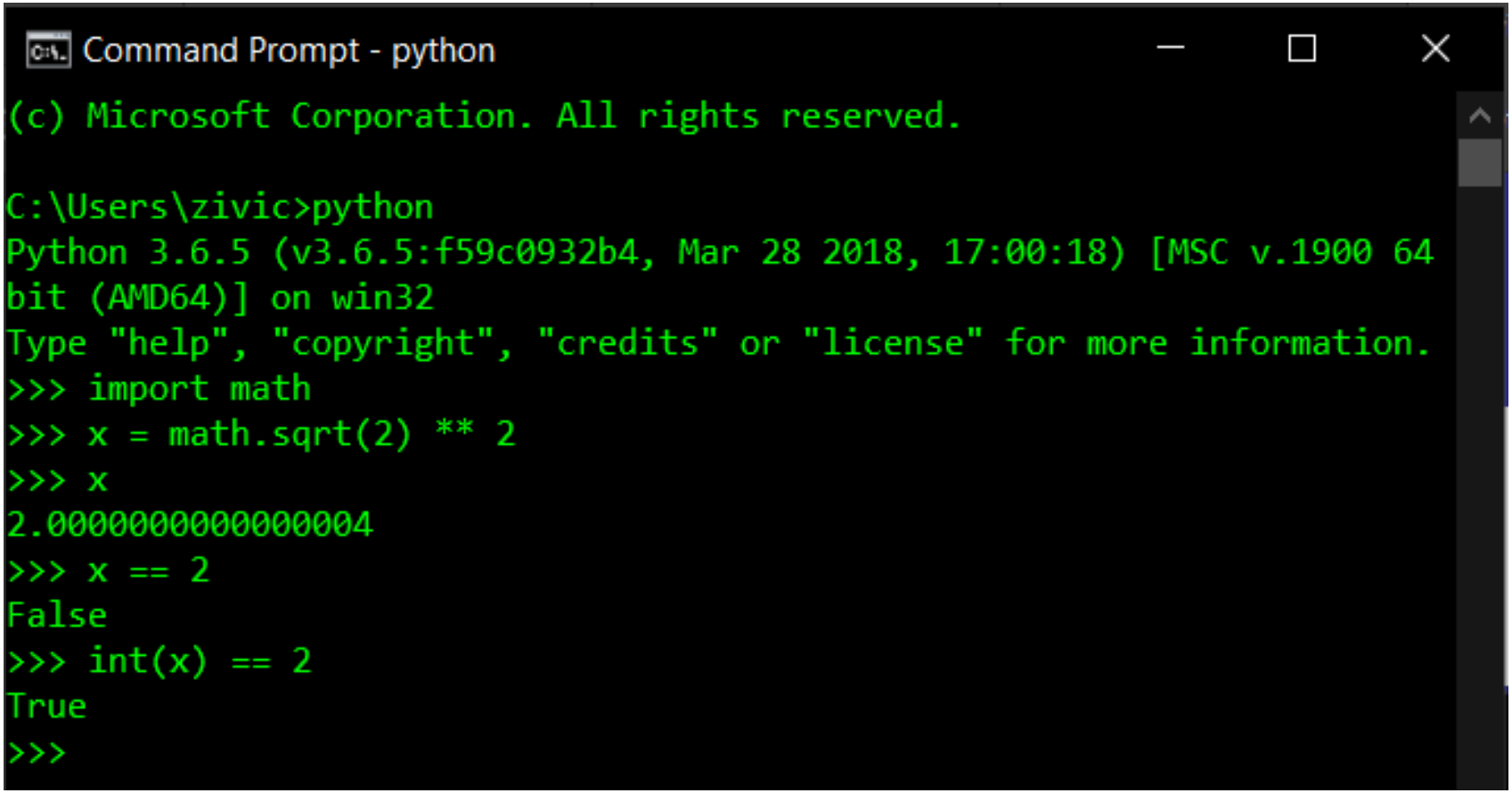
# Readability

```
cost = 0.
grocery_items = ["apple", "celery", "bread"]
sale = False

for item in grocery_items:
    if item == "apple":
        cost = cost + 1.50
    if item == "celery":
        cost = cost + 3.50
    if item == "bread":
        if sale:
            cost = cost + (2.50*0.9)
        else:
            cost = cost + 2.50
```

# Accuracy

```
R version 4.1.0 (2021-05-18) -- "Camp Pontanezen"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x = sqrt(2)**2
> x
[1] 2
> x == 2
[1] FALSE
>
```

# Accuracy[5]




[5] Julia also presents this correctly

# Advantage: cross-software interactions

Python is a good glue language

- Easily interacts with other software
  - `C`, `C++`
- Interact with other software:
  - `R`: `RPy2`
  - `Stan`: `PyStan`
  - `Julia`: `PyJulia`
  - `SAS`: `SASPy`

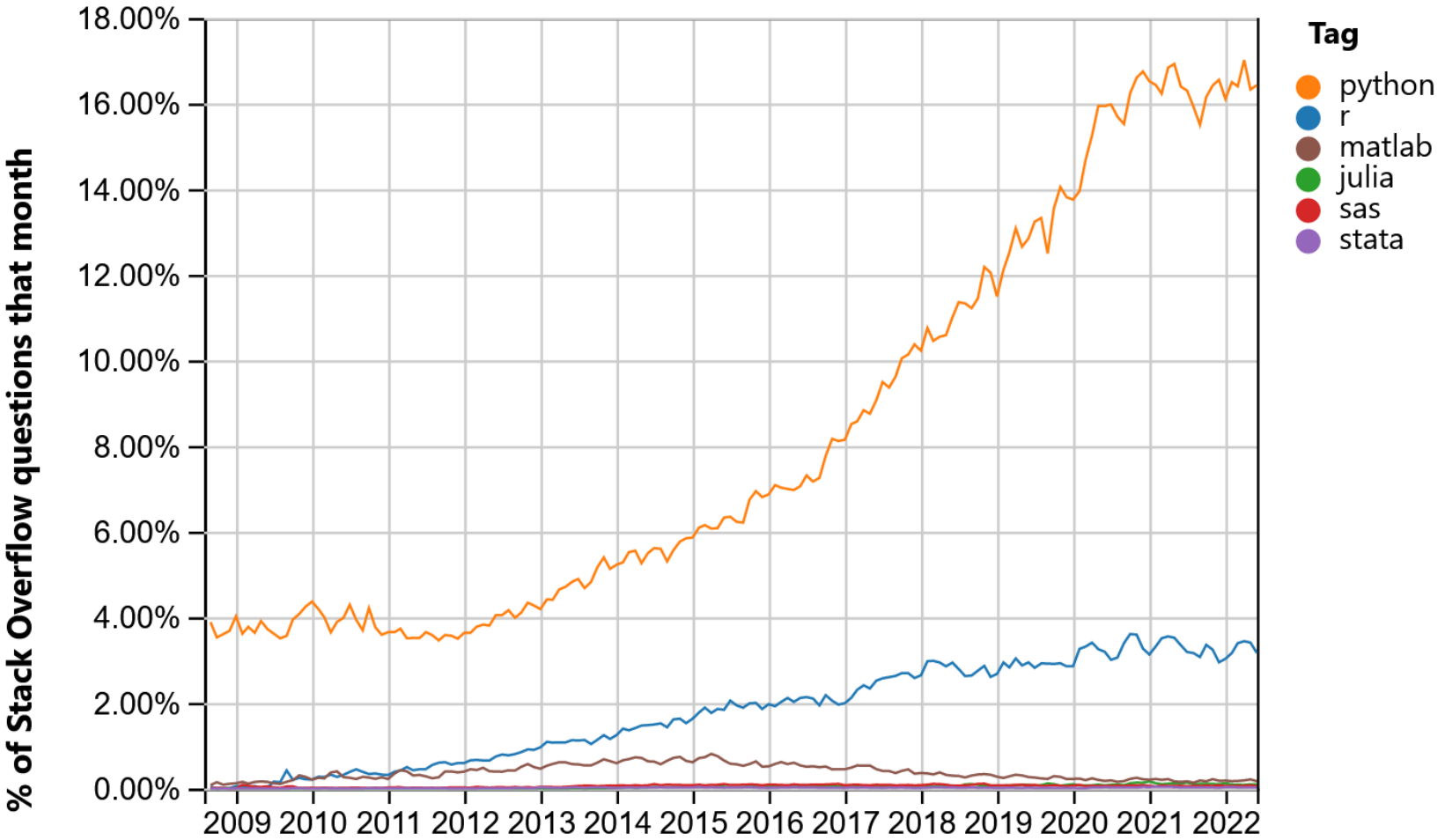
# Advantage: cross-software interactions

All slides made with  and LaTeX

- Using `pylatex`

```
######################
# TeX setup
doc = Document(tex_file_name, documentclass="beamer")

doc.preamble.append(Command('usetheme', 'Copenhagen'))
doc.preamble.append(Command('usecolortheme', 'whale'))

doc.packages.append(Package('amsmath'))
doc.packages.append(Package('xcolor'))
doc.packages.append(Package('graphicx'))
doc.packages.append(Package('fontawesome5'))
doc.packages.append(Package('pythonhighlight'))
```

# Advantage: popularity



# Advantage: popularity

Combination of:

- Programmers
- Scientists
- Statisticians

Wide support for use-cases

# Example: Black Holes

Computations and image processing done using [6]

- The telescope array generates >350 terabytes of data daily

[6] Akiyama K, et al. (2022) First Sagittarius A* Event Horizon Telescope Results. I. The Shadow of the Supermassive Black Hole in the Center of the Milky Way. *The Astrophysical Journal Letters* 930.2 (2022): L12

# Illustrative Applications

Examples

- Basic statistical applications
- Plasmode data simulation with GANs
- Scientific abstract text generator

# Basics: Regression

```
# Logistic Regression
import statsmodels.api as sm
import statsmodels.formula.api as smf

fm = smf.glm("y ~ x + z",
             d,
             family=sm.families.Binomial()).fit()
print(fm.summary())
```

# Basics: Inverse Probability Weighting

```
# Inverse Probability Weighting
fm = smf.glm("a ~ x_1 + x_2 + x_3",
             d,
             family=sm.families.Binomial()).fit()
pi_a = fm.predict()

ipw = 1 / (d['a'] * pi_a + (1-d['a'])*(1-pi_a))

f = sm.families.family.Binomial(sm.families.links.
                                identity())
msm = smf.gee("y ~ a", d.index, d,
              weights=ipw,
              family=f).fit()
print(msm.summary())
```

7

[7] Can also be done using zEpid

# Basics: Survival Analysis

```
# Cox Proportional Hazards
from lifelines import CoxPHFitter

cph = CoxPHFitter()
cph.fit(d[['time', 'delta', 'a', 'z']],
        duration_col='time',
        event_col='delta',
        strata='x')
cph.print_summary()
```

# Plasmode Simulations with GAN

Generative adversarial neural network (GAN) to generate data[8]

Generate new data from existing data

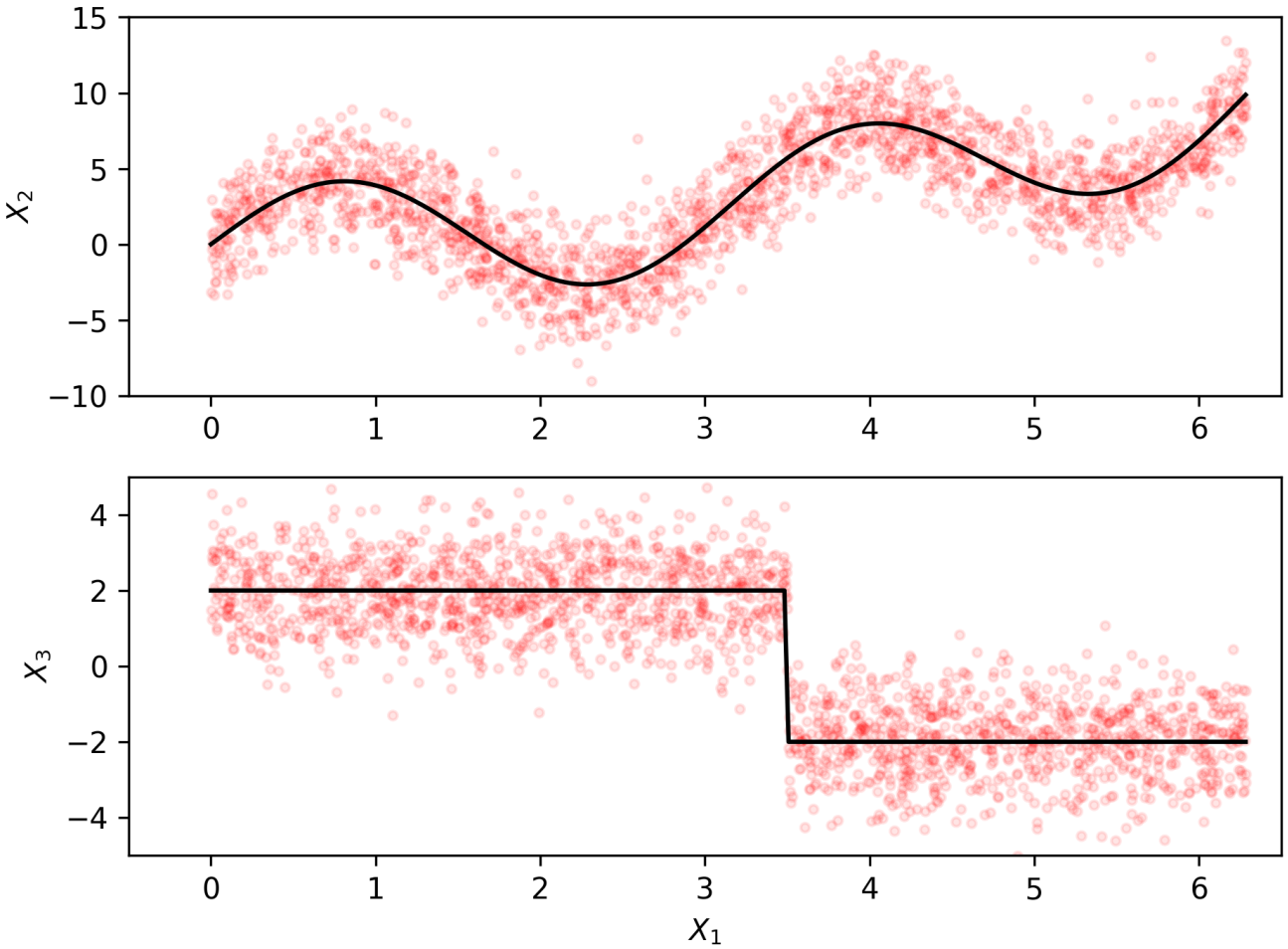
- Avoid arbitrary data generating decisions
- Reflect performance in your particular application
- Share data without re-identification

Less than 150 lines

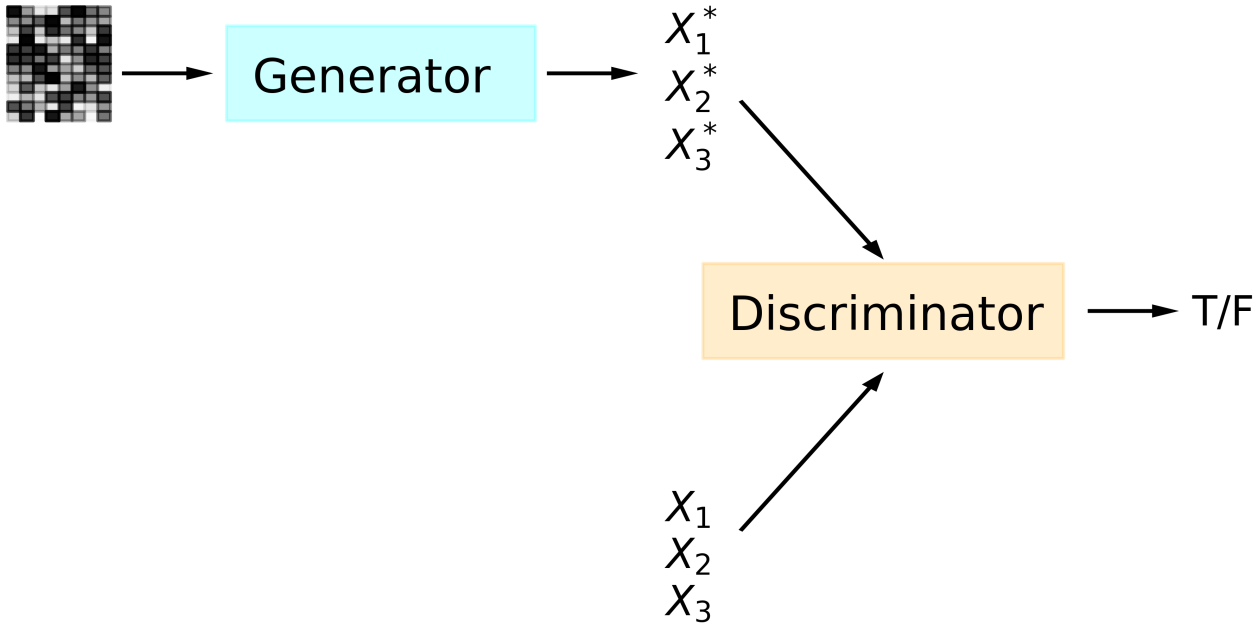
- Compatible with arbitrary input data

[8] Athey S et al. (2021). Using Wasserstein generative adversarial networks for the design of Monte Carlo simulations. *Journal of Econometrics*

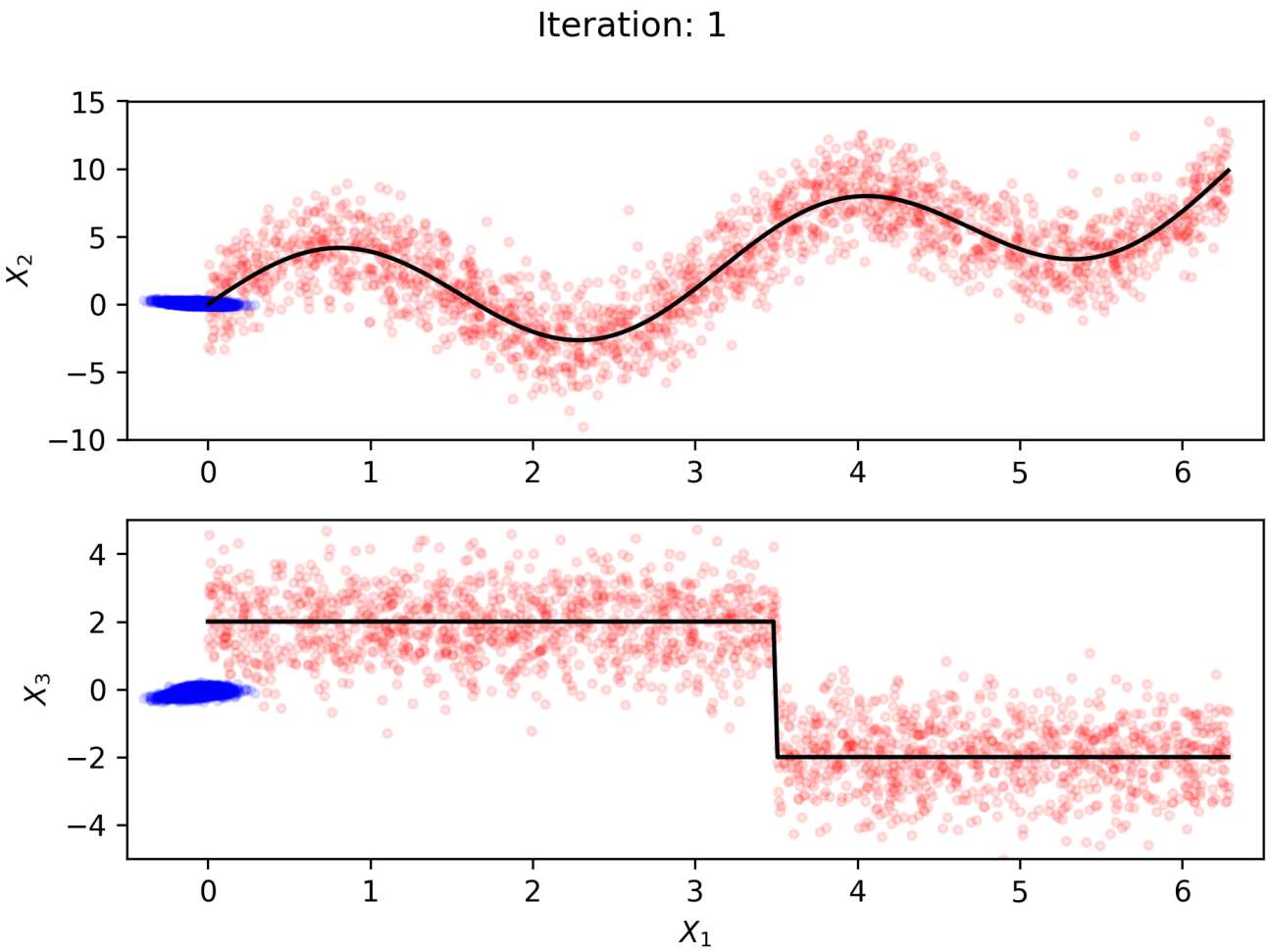
# Plasmode Simulations with GAN



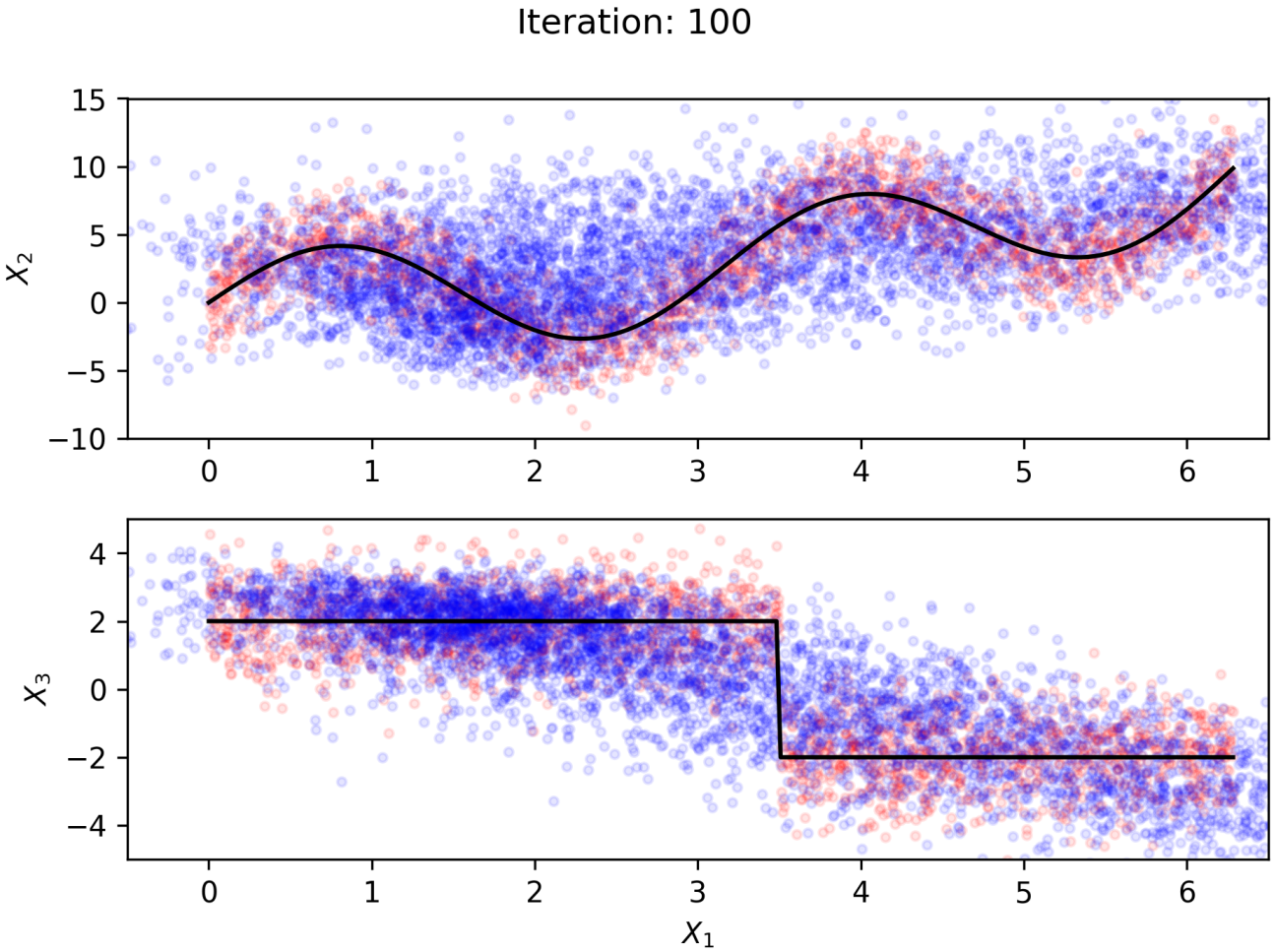
# Plasmode Simulations with GAN



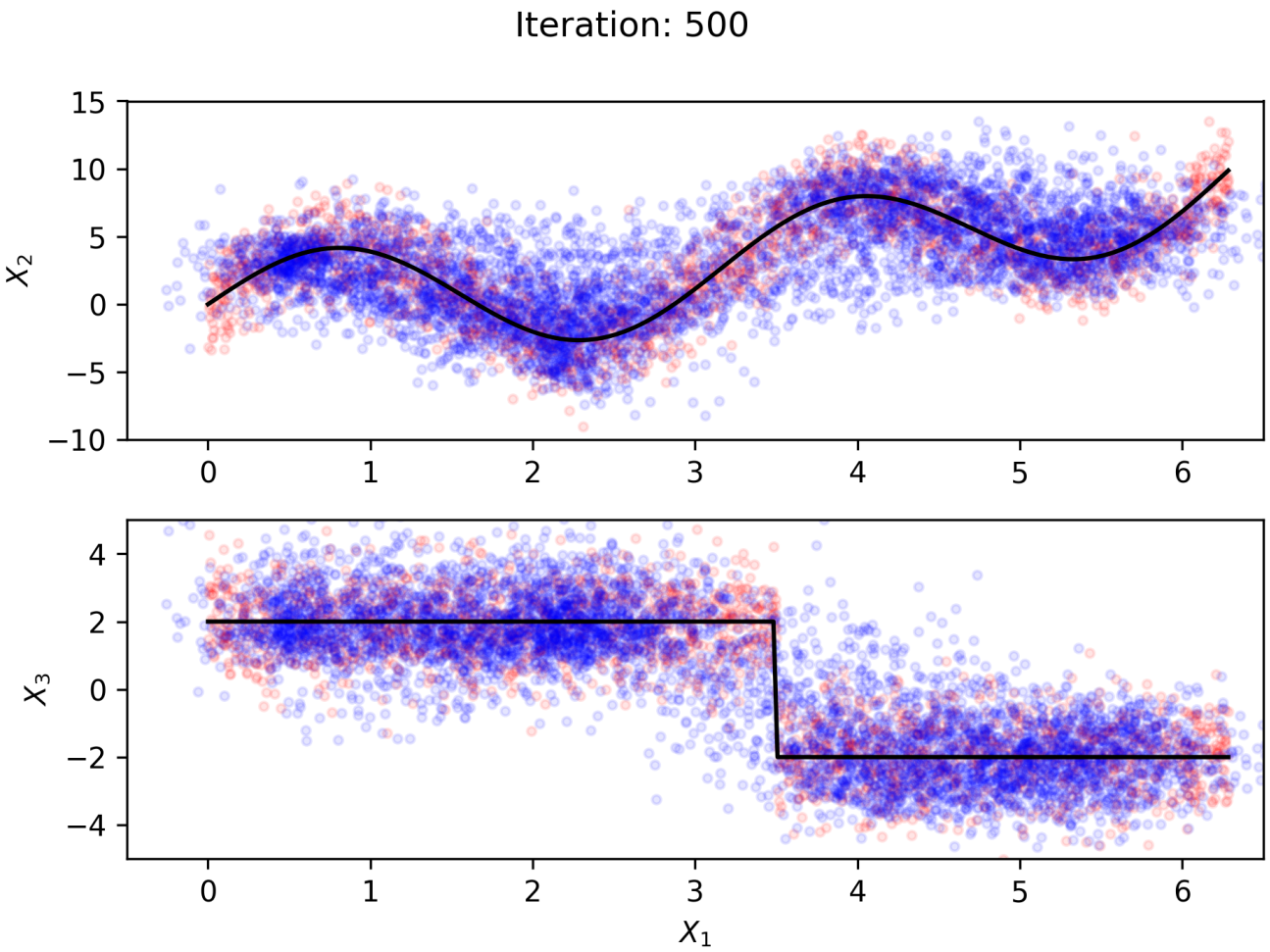
# Plasmode Simulations with GAN



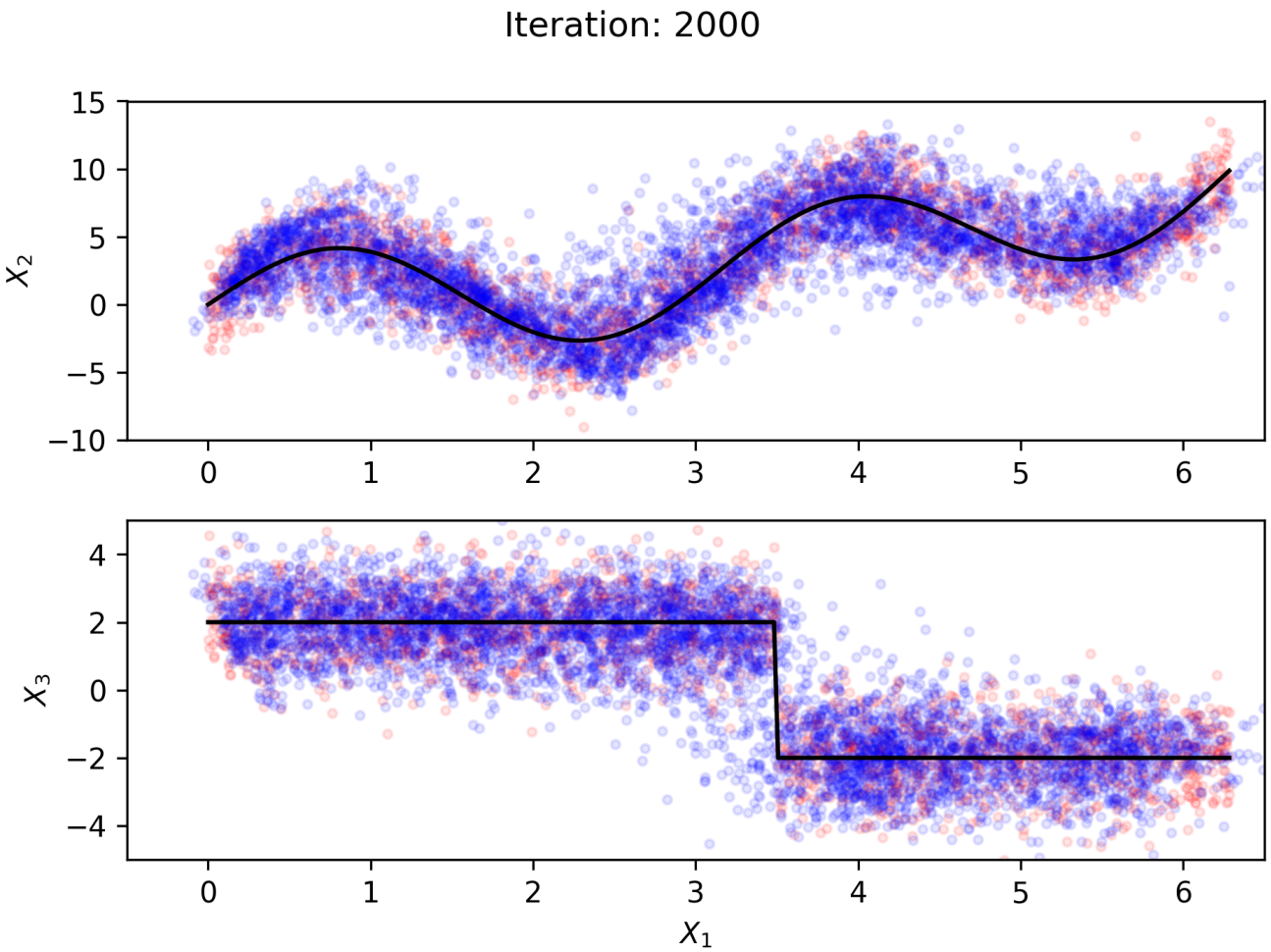
# Plasmode Simulations with GAN



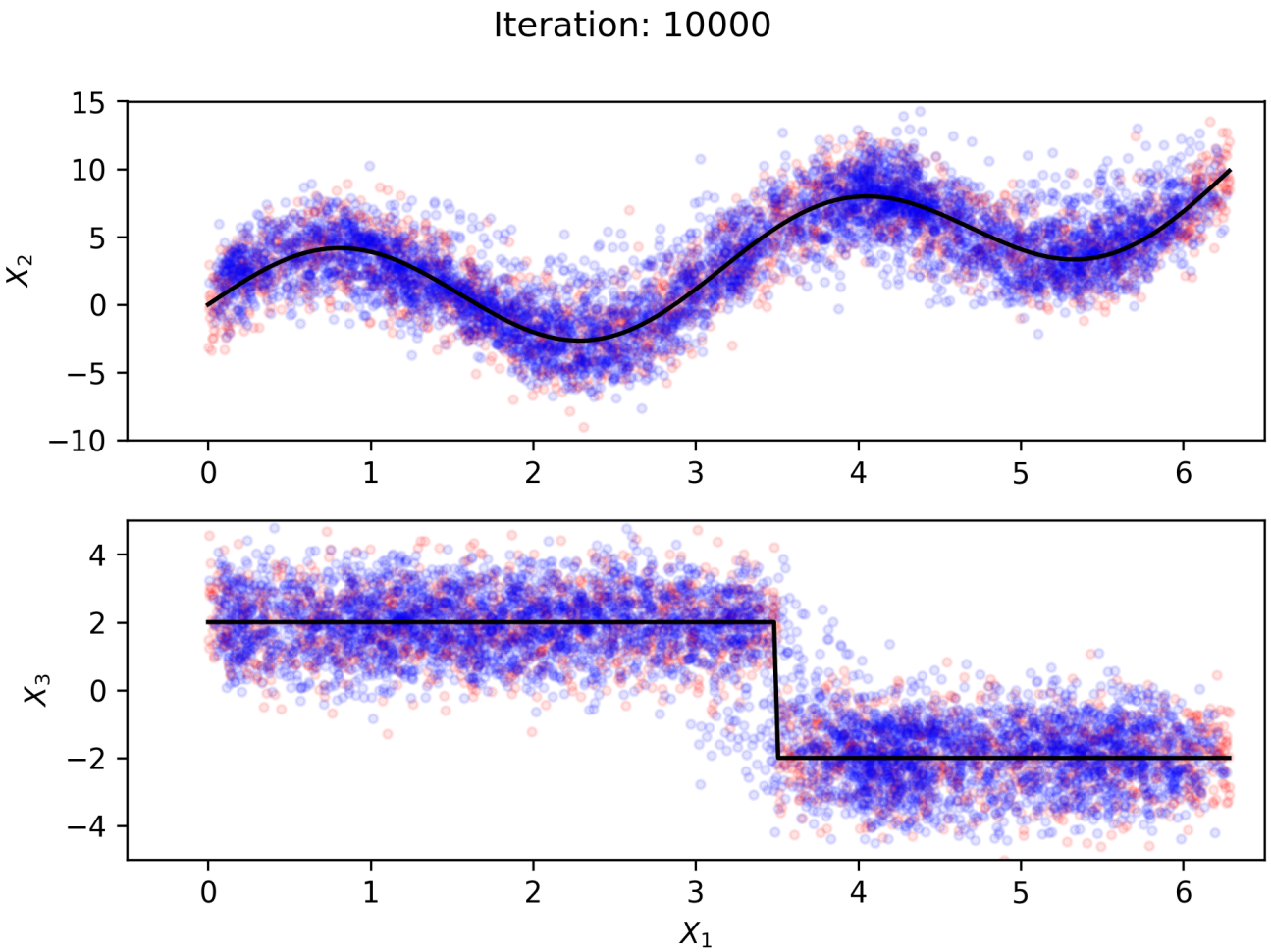
# Plasmode Simulations with GAN



# Plasmode Simulations with GAN



# Plasmode Simulations with GAN



# Text Generation with RNN

Recurrent neural network (RNN) to generate abstracts[9,10]

Generate abstracts focusing on causal inference

- Whole process in 
- Train using published abstracts
  - Query PubMed

Less than 300 lines

- Written generally, so could be re-trained for other topics

[9] Code available at https://github.com/pzivich/RNN-Abstract-Generator

[10] Sutskever I, Martens J, & Hinton GE (2011). Generating text with recurrent neural networks. In *ICML*

# RNN overview

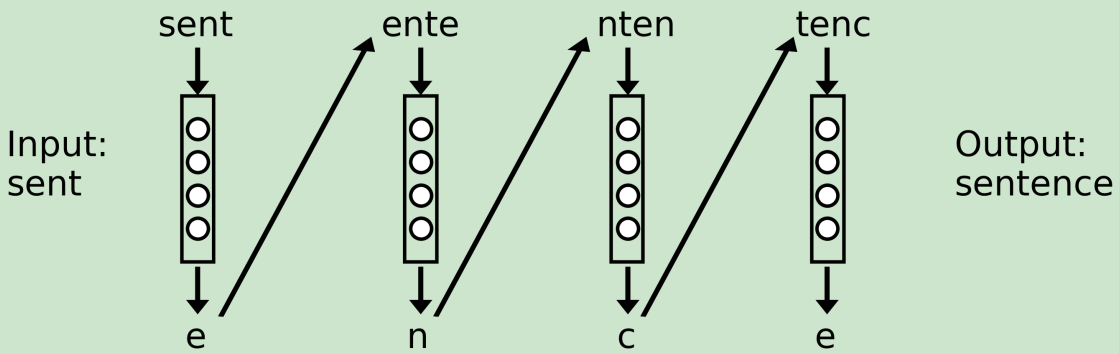
1: Query PubMed

- 1a: Conduct search & extract PubMed IDs
- 1b: Select random sample
- 1c: Pull meta-data from PubMed

2: Text processing

- 2a: Extract abstracts
- 2b: Format text to training data

3: RNN



`Input:` Randomized control trials have been criticized as

`Input:` Randomized control trials have been criticized as

`Output:` Randomized control trials have been criticized as a mediation analysis of social schools and other time series (asthma multiple compositions, and confounding. maps that each of the hazard ratio of their assumptions that may examine the effects of causal inference network (rct) and the causal relationship between disease construction and intervention (i.e., all motivating population status are presented to assess robustness framework for interested in the method to infer the substance use and investigating the associations between structured patients

`Input:` Inverse probability of treatment weights were

`Input:` Inverse probability of treatment weights were

`Output:` Inverse probability of treatment weights were associated with all interventions or as a causal association between long-term sources and covariate and the consumption of the behavioral research results. in this article, we describe the method of the results of responses and work in a variety of high-current for genetic variants are problematic in statisticians

`Input:` results were statistically significant (p=0.

`Input:` results were statistically significant (p=0.

`Output:` Results were statistically significant (p=0.011)

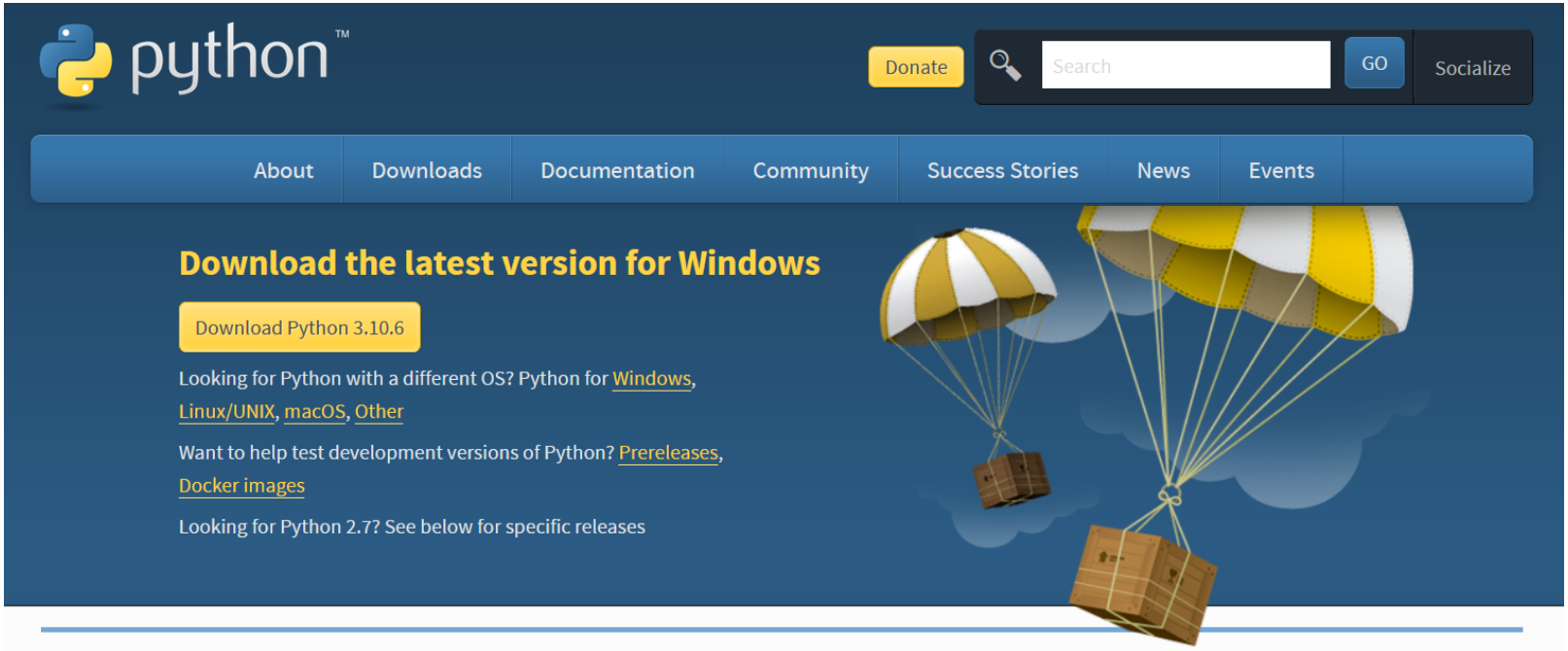
`Output:` a causal inference approach to interpret, and the a propensity score (ps)

`Output:` controlling for pregnancy manifererisequally associated

`Output:` Was protective but not statistically significant (p=0.02)
`Output:` We found no evidence of a causal effect (p=0.012)

# Getting Started with

https://www.python.org/downloads/




But often will want multiple versions of  available

# A Better Way...

Use pyenv [11]

```
Microsoft Windows [Version 10.0.19044.1889]
(c) Microsoft Corporation. All rights reserved.

C:\Users\zivic>pyenv versions
  3.10.5
  3.11.0b4
* 3.6.5 (set by C:\Users\zivic\.pyenv\pyenv-win\version)
  3.7.8
  3.9.4

C:\Users\zivic>pyenv global 3.9.4

C:\Users\zivic>python
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr  6 2021, 13:40:21) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> quit()

C:\Users\zivic>pyenv global 3.6.5

C:\Users\zivic>python
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> quit()
```

[11] A good introduction is available at https://realpython.com/intro-to-pyenv/

# Getting Started with 

Integrated Development Environment (IDE)

- PyCharm
- Jupyter Notebook
- Atom
- RStudio

# Essential Packages

Basics

- NumPy, SciPy, pandas

Statistics

- statsmodels, lifelines

Visualization

- matplotlib, seaborn

Machine learning

- sci-kit learn, torch

# Learning

A number of online resources

Some I've made:

- https://github.com/pzivich/Python-for-Epidemiologists
- https://github.com/pzivich/publications-code
- Smith MJ et al. (2022). Introduction to computational causal inference using reproducible Stata, R, and Python code: A tutorial. *Statistics in Medicine*, 41(2), 407-432.

What worked for me:

- Replicate a completed project in
- Then start a project in

# Conclusions

Be familiar with more than one software

Strongly consider  as the next

- Language features
- Cross-software capabilities
- Popularity

Uptake in epidemiology / biostatistics is low

- More dominated by comp sci / data science
- Lots of opportunity for contributions

# Questions?