Sampling: Reproducibility

```
$ echo "Data Science Institute"
```

Sampling in Python

Sampling in Python

- Some reasons you might need to sample in Python:
  - Simulating data
  - Bootstrap sampling
  - Hypothesis testing
- Common sampling functions in Python:
  - `numpy.random.choice()` – sample items from an array
  - `numpy.random.uniform()` – sample from a uniform distribution (input low and high)
  - `numpy.random.normal()` – sample from a normal distribution (input mean and standard deviation)

Sampling in Python

- These functions are all forms of **probability sampling**, therefore there is always some randomness involved.
- The “randomness” in Python and R results from **pseudorandom number generators**.
- Pseudorandom number generators output numbers based on a **set algorithm and an initial seed**.
- The same initial seed will produce the same outputted number(s) or item(s) every time

Seeds

(Partial review from Module 3)

- `numpy.random.seed()` in python allow you to set the initial seed for Python functions that use pseudorandom number
- A sampling function called after a specific seed is set will produce the same output each time
- Setting a seed makes sampling procedures reproducible

Seeds: Scope (Python)

`numpy.random.seed()` sets the initial state of the random number generator within NumPy's environment.

- The state of the random number generator **changes deterministically** after executing functions that depend on the seed.
- ⚠️ Setting the seed at the beginning of a script will produce consistent results when the script is run in its entirety, but not necessarily for individual calls to the same function in. ⚠️

When to set seed:

- Writing reproducible sampling scripts or simulation-based studies
- Creating reproducible examples
- Code testing and debugging

When NOT to set seeds:

- Inside a function or loop that you want to produce different results every time

Data Documentation

Data Documentation

- Open data is one of the hallmarks of open science and reproducibility
- Having access to the data behind a study is useless if you know nothing about how that data was collected or manipulated prior to analysis
- **Data documentation** refers to the process of recording all of the steps taken to obtain and process your data
- Similar to commenting code, documenting data communicates important features of your data set that may impact its analysis and allows other researchers to more easily reproduce your results

What should be documented?

- All aspects of survey design should be documented. This includes (but is not limited to):
- Populations
  - Target, frame
- Sampling methodology
  - Type of sampling used, eligibility criteria, sample size, any supplemental data used
- Mode

...

What should be documented? (continued...)

- Timeline
  - Release and closing date, frequency (for repeat surveys or longitudinal studies)
- Response rate
- Cleaning procedure
  - Any editing to address missing values, repeated values, outliers
- Weights
  - Weighting procedure and supplemental data used
- Accuracy
  - Known sources of error, bias, limitations

How should data be documented?

- In addition to raw data files, consider:
  - Recording and summarising all steps taken during the survey design and sampling process
  - Including all original questionnaire or survey materials
  - Including code that was used for processing
    - Be sure code is reproducible by using comments, seeds, R projects, etc.
- Website like **GitHub** and **The Open Science Framework (OSF)** can help store relevant files in self-contained repositories for later reference.

Next

Inquity