IN-CORE:

Interdependent Networked Community Resilience Modeling Environment

Jong S. Lee, Ph.D.

Principal Research Scientist, NCSA, UIUC (jonglee1@Illinois.edu)

@ PAM2020 (1/16/2020)

IN-CORE

ILLINOIS

NCSA | National Center for Supercomputing Applications

Resilience

Center of Excellence for Risk-Based Community Resilience Planning

The National Institute of Standards and Technology (NIST) funded the multi-university five-year Center of Excellence for Risk-Based Community Resilience Planning (CoE, http://resilience.colostate.edu/), headquartered at Colorado State University, to develop the measurement science to support community resilience assessment and IN-CORE platform

Colorado State University
ILLINOIS
ILLINOIS NCSA | National Center for Supercomputing Applications
Oregon State UNIVERSITY OSU
UNIVERSITY of WASHINGTON
STATE

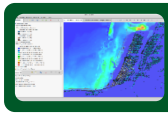
TEXAS A&M UNIVERSITY
RICE
USA
CU Boulder
CAL POLY POMONA
The UNIVERSITY of OKLAHOMA
KU SCHOOL OF ENGINEERING The University of Kansas

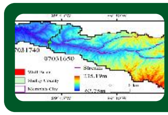
What is IN-CORE?

- Measurement science is implemented on a platform called **I**nterdependent **N**etworked **Co**mmunity **R**esilience Modeling **E**nvironment (IN-CORE)
- It incorporates a risk-based approach to decision-making that enables quantitative comparisons of alternative resilience strategies.
- On the platform, users can run scientific analyses that model the impact of natural hazards and resiliency against the impact on communities.

What is IN-CORE?

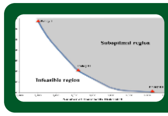
Individual Hazards

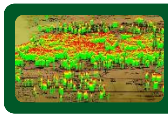

Multiple and Competing Hazards

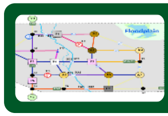

Long-term Resilience Assessment

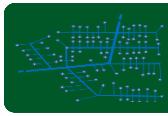
Social Systems

Economic Systems


Optimization Strategies


Buildings


Transportation networks


Water & Wastewater networks

Energy networks

Communication networks

Centerville
(EQ, Tornado)

Seaside, Oregon
(EQ, Tsunami)

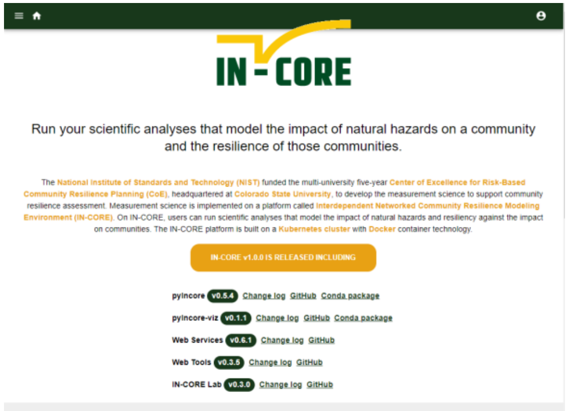
Memphis, TN & MMSA
(EQ, Flood)

Joplin, Missouri
(Tornado)

Galveston, TX
(Hurricane: Surge, Waves, Wind)

IN-CORE v1.0.0

- Released on Dec 20, 2019
- Source code at GitHub
  - https://github.com/IN-CORE
  - Mozilla Public License v2.0 (MPL-2.0)
- Conda packages
  - https://anaconda.org/IN-CORE
- IN-CORE landing page
  - https://incore.ncsa.illinois.edu/



Run your scientific analyses that model the impact of natural hazards on a community and the resilience of those communities.

The National Institute of Standards and Technology (NIST) funded the multi-university five-year Center of Excellence for Risk-Based Community Resilience Planning (CoE), headquartered at Colorado State University, to develop the measurement science to support community resilience assessment. Measurement science is implemented on a platform called Interdependent Networked Community Resilience Modeling Environment (IN-CORE). On IN-CORE, users can run scientific analyses that model the impact of natural hazards and resiliency against the impact on communities. The IN-CORE platform is built on a Kubernetes cluster with Docker container technology.

IN-CORE v1.0.0 IS RELEASED INCLUDING

pyincore v0.5.4 Change log GitHub Conda package

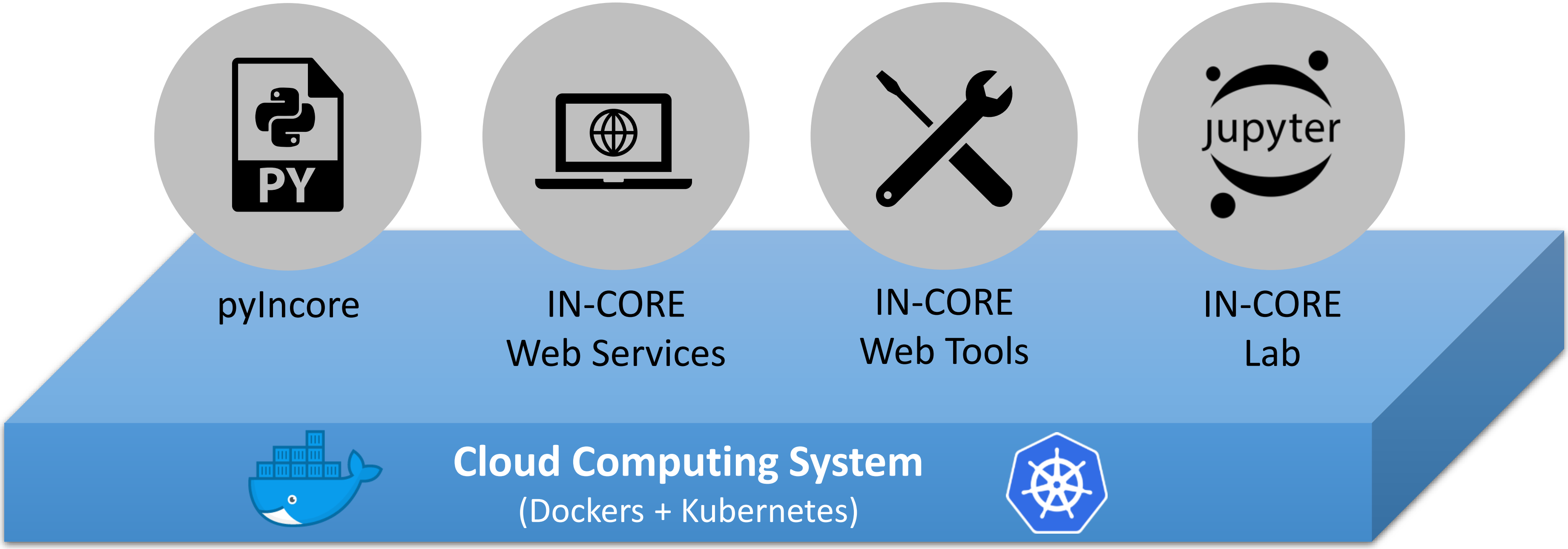
pyincore-viz v0.1.1 Change log GitHub Conda package

Web Services v0.6.1 Change log GitHub

Web Tools v0.3.5 Change log GitHub

IN-CORE Lab v0.3.0 Change log GitHub

Architecture



pyIncore

IN-CORE
Web Services

IN-CORE
Web Tools

IN-CORE
Lab

Cloud Computing System

(Dockers + Kubernetes)

Containers on Kubernetes

- Container (Docker): A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it
- Kubernetes is a container management system
- The technology brings us
  - Automatic scaling corresponding to demands
  - Portability – deployable to different cloud
  - Streamline deployment from development and testing

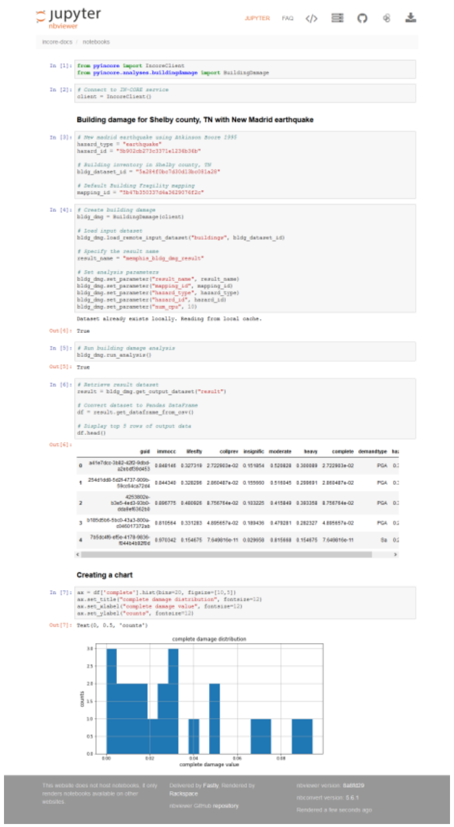
pyIncore

- Python library (modules) for IN-CORE
- Three components
  - Interact with IN-CORE web services
  - Base classes for analysis and datasets
  - Analyses
- pyIncore-viz
  - Visualization methods and utilities
- How to install

```
conda install -c in-core pyincore
conda install -c in-core pyincore-viz
```

- Documentation is available
  - Jupyter notebooks with example analysis
  - Technical reference documents
- Example:
  - https://nbviewer.jupyter.org/github/IN-CORE/incore-docs/blob/master/notebooks/building_dmg.ipynb

PY
pyIncore



```
In [1]: from pyincore import IncoreClient
        from pyincore.analyses.buildingdamage import BuildingDamage

In [2]: # Connect to IN-CORE service
        client = IncoreClient()
```

Building damage for Shelby county, TN with New Madrid earthquake

```
In [3]: # New madrid earthquake using Atkinson Boore 1995
        hazard_type = "earthquake"
        hazard_id = "5b902cb273c3371e1236b36b"

        # Building inventory in Shelby county, TN
        bldg_dataset_id = "5a284f0bc7d30d13bc081a28"

        # Default Building Fragility mapping
        mapping_id = "5b47b350337d4a3629076f2c"

In [4]: # Create building damage
        bldg_dmg = BuildingDamage(client)

        # Load input dataset
        bldg_dmg.load_remote_input_dataset("buildings", bldg_dataset_id)

        # Specify the result name
        result_name = "memphis_bldg_dmg_result"

        # Set analysis parameters
        bldg_dmg.set_parameter("result_name", result_name)
        bldg_dmg.set_parameter("mapping_id", mapping_id)
        bldg_dmg.set_parameter("hazard_type", hazard_type)
        bldg_dmg.set_parameter("hazard_id", hazard_id)
        bldg_dmg.set_parameter("num_cpu", 10)

        Dataset already exists locally. Reading from local cache.

Out[4]: True

In [5]: # Run building damage analysis
        bldg_dmg.run_analysis()

Out[5]: True

In [6]: # Retrieve result dataset
        result = bldg_dmg.get_output_dataset("result")

        # Convert dataset to Pandas DataFrame
        df = result.get_dataframe_from_csv()

        # Display top 5 rows of output data
        df.head()

Out[6]:
```

	guid	immocc	lifesfty	collprev	insignific	moderate	heavy	complete	demandtype	ha
0	a41e7dcc-3b82-42f2-9dbd-a2ebdf38d453	0.848146	0.327319	2.722903e-02	0.151854	0.520828	0.300089	2.722903e-02	PGA	0.:
1	254d1dd8-5d2f-4737-909b-59cc64ca72d4	0.844340	0.328296	2.860487e-02	0.155660	0.516045	0.299691	2.860487e-02	PGA	0.:
2	4253802e-b3e5-4ed3-93b0-dda8ef6362b8	0.896775	0.480926	8.756764e-02	0.103225	0.415849	0.393358	8.756764e-02	PGA	0.:
3	b185d5b6-5bc0-43a3-800a-c0460173 72ab	0.810564	0.331283	4.895657e-02	0.189436	0.479281	0.282327	4.895657e-02	PGA	0.2
4	7b5dc4f6-ef5e-4178-9836-f044b4b92f6d	0.970342	0.154675	7.649816e-11	0.029658	0.815668	0.154675	7.649816e-11	Sa	0.2

Creating a chart

```
In [7]: ax = df['complete'].hist(bins=20, figsize=[10,5])
        ax.set_title("complete damage distribution", fontsize=12)
        ax.set_xlabel("complete damage value", fontsize=12)
        ax.set_ylabel("counts", fontsize=12)

Out[7]: Text(0, 0.5, 'counts')
```

This website does not host notebooks, it only renders notebooks available on other websites.

Delivered by Fastly, Rendered by Rackspace

nbviewer GitHub repository

nbviewer version: 8a6fd29

nbconvert version: 5.6.1

Rendered a few seconds ago

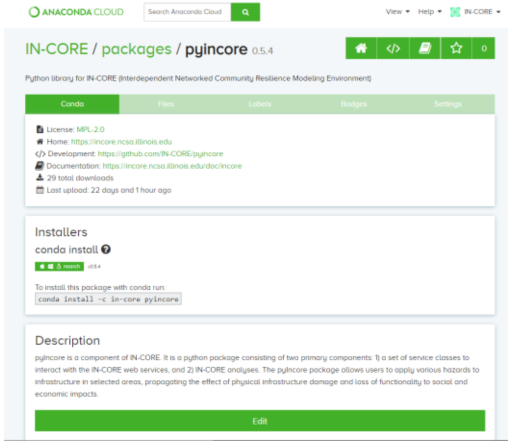
Currently Available Analyses

- Bridge damage
- Building damage
- Cumulative building damage
- Electric power facility damage
- Nonstructural building damage
- Tornado Electric Power Network (EPN) damage
- Pipeline damage
- Pipeline damage with repair rate
- Water facility damage
- Mean damage
- Monte Carlo failure probability
- Building functionality analysis
- Building Portfolio recovery
- Transportation recovery
- Housing unit allocation
- Population dislocation
- Joplin Computable General Equilibrium (CGE)

Two testbeds are available as Jupyter Notebook
More analyses will be added in near future

pyIncore Resources

- pyIncore
  - GitHub: https://github.com/IN-CORE/pyincore
  - Anaconda: https://anaconda.org/IN-CORE/pyincore
  - General documentation: https://incore.ncsa.illinois.edu/doc/incore/pyincore.html
  - Technical reference documentation: https://incore.ncsa.illinois.edu/doc/pyincore/
- pyIncore-viz
  - More capability will come in future release
  - GitHub: https://github.com/IN-CORE/pyincore-viz
  - Anaconda: https://anaconda.org/IN-CORE/pyincore-viz



IN-CORE / packages / pyincore 0.5.4

Python library for IN-CORE (Interdependent Networked Community Resilience Modeling Environment)

License: MPL-2.0
Home: https://incore.ncsa.illinois.edu
Development: https://github.com/IN-CORE/pyincore
Documentation: https://incore.ncsa.illinois.edu/doc/incore
29 total downloads
Last upload: 22 days and 1 hour ago

Installers

conda install

To install this package with conda run:

```
conda install -c in-core pyincore
```

Description

pyIncore is a component of IN-CORE. It is a python package consisting of two primary components: 1) a set of service classes to interact with the IN-CORE web services, and 2) IN-CORE analyses. The pyIncore package allows users to apply various hazards to infrastructure in selected areas, propagating the effect of physical infrastructure damage and loss of functionality to social and economic impacts.

IN-CORE Web Services

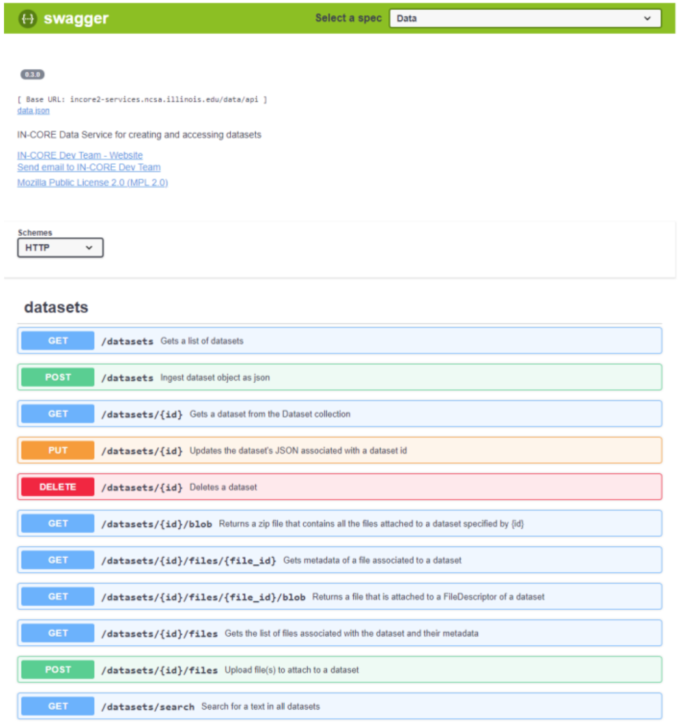
- RESTful Web Service Technology
- Database: MongoDB
- Authentication service
- Data service
  - Storing/managing datasets
- Hazard service
  - Storing hazard definitions
  - Getting hazard value by location
  - Earthquake
  - Tsunami
  - Tornado
  - Hurricane wind field
- DFR3 service
  - Storing/managing fragility curve sets, damage functions, repair, recovery, restoration
  - Matching inventory to fragility curve set
- Geospatial Viz service
  - Generating geospatial map/layer images
- Semantic service
  - Storing/managing definition of datasets
  - Coming to next release
- Space service
  - Creating content spaces
  - Access control

IN-CORE
Web Services

IN-CORE Web Services

- How to use IN-CORE Web Services
  - Need to have a user account managed by NCSA identity management system
    - For authentication
    - For authorization (access control)
  - Various ways
    - RESTful web service clients
    - Web browser
    - pyIncore
    - IN-CORE Web Tools (browsing only)
- Technical reference documentation is available

IN-CORE
Web Services



[Base URL: incore2-services.ncsa.illinois.edu/data/api]
data.json

IN-CORE Data Service for creating and accessing datasets

IN-CORE Dev Team - Website
Send email to IN-CORE Dev Team
Mozilla Public License 2.0 (MPL 2.0)

Schemes
HTTP

datasets

GET `/datasets` Gets a list of datasets

POST `/datasets` Ingest dataset object as json

GET `/datasets/{id}` Gets a dataset from the Dataset collection

PUT `/datasets/{id}` Updates the dataset's JSON associated with a dataset id

DELETE `/datasets/{id}` Deletes a dataset

GET `/datasets/{id}/blob` Returns a zip file that contains all the files attached to a dataset specified by {id}

GET `/datasets/{id}/files/{file_id}` Gets metadata of a file associated to a dataset

GET `/datasets/{id}/files/{file_id}/blob` Returns a file that is attached to a FileDescriptor of a dataset

GET `/datasets/{id}/files` Gets the list of files associated with the dataset and their metadata

POST `/datasets/{id}/files` Upload file(s) to attach to a dataset

GET `/datasets/search` Search for a text in all datasets

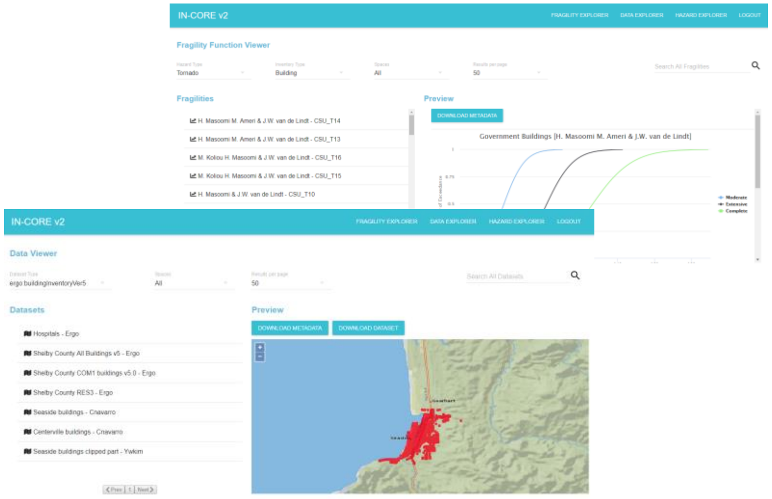
IN-CORE Web Services Resources

- GitHub:
  - https://github.com/IN-CORE/incore-services
- Technical reference documentation:
  - https://incore.ncsa.illinois.edu/doc/api/

IN-CORE Web Tools

IN-CORE
Web Tools

- Lightweight web applications for IN-CORE Web Services
- Allows users to browse, search, and preview data from the service
- Data browser
  - Client to data service
- Fragility browser
  - Client to DFR3 service
  - Currently it shows fragilities
- Hazard browser
  - Client to hazard service
- Login with your account credential to access tools at
  - https://incore.ncsa.Illinois.edu



Fragility Function Viewer

Hazard Type: Tornado
Inventory Type: Building
Spaces: All
Results per page: 50

Fragilities

- H. Masoomi M. Ameri & J.W. van de Lindt - CSU_T14
- H. Masoomi M. Ameri & J.W. van de Lindt - CSU_T13
- M. Koliou H. Masoomi & J.W. van de Lindt - CSU_T16
- M. Koliou H. Masoomi & J.W. van de Lindt - CSU_T15
- H. Masoomi & J.W. van de Lindt - CSU_T10

Preview

Government Buildings [H. Masoomi M. Ameri & J.W. van de Lindt]

Data Viewer

Dataset Type: ergo buildingInventoryVer5
Spaces: All
Results per page: 50

Datasets

- Hospitals - Ergo
- Shelby County All Buildings v5 - Ergo
- Shelby County COM1 buildings v5.0 - Ergo
- Shelby County RES3 - Ergo
- Seaside buildings - Cnavarro
- Centerville buildings - Cnavarro
- Seaside buildings clipped part - Ywkim

Preview

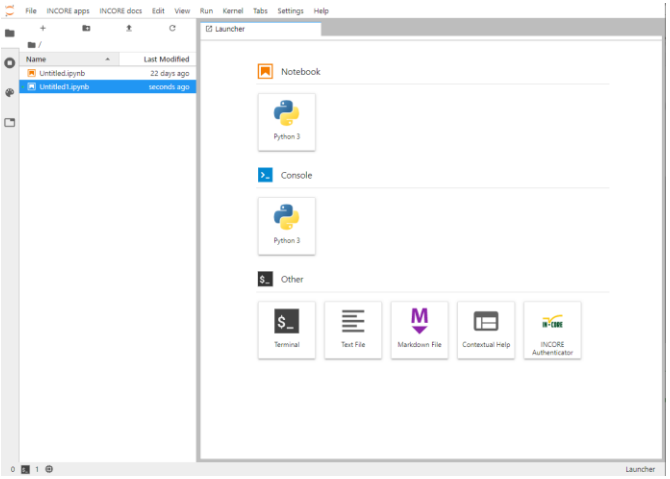
IN-CORE Web Tools Resources

- Access at
  - https://incore.ncsa.Illinois.edu
- GitHub:
  - https://github.com/IN-CORE/incore-ui
- General documentation:
  - https://incore.ncsa.illinois.edu/doc/incore/webtools.html

IN-CORE Lab

- Customized JupyterLab
- Integrated environments for developing algorithms
  - Menu items to access documentations, IN-CORE Web Tools
  - Authentication (single-sign-on)
- Two ways to use IN-CORE Lab
  - Locally (a docker image will be available)
  - Online (JupyterHub at NCSA)
- Online version:
  - pyIncore is installed with all dependent libraries
  - Includes popular python libraries such as Pandas, GeoPandas, Matplotlib, etc.
  - Account and allocation policy are under development for public access

IN-CORE Lab



Name	Last Modified
Untitled.ipynb	22 days ago
Untitled1.ipynb	seconds ago

Demo

- Note: how to sign up an account
  - https://incore.ncsa.illinois.edu/doc/incore/account.html
- https://incore.ncsa.Illinois.edu
  - Manuals
  - Example Jupyter notebooks
  - Two testbeds
  - Web tools
  - IN-CORE Lab

Acknowledgement

- This work was conducted as part of the NIST Center of Excellence for Risk-Based Community Resilience Planning under Cooperative Agreement 70NANB15H044 between the National Institute of Standards and Technology (NIST) and Colorado State University. The content expressed in this report are the views of the authors and do not necessarily represent the opinions or views of NIST or the U.S Department of Commerce.

Thank You

- Center for Risk-based Community Resilience Planning
  - http://resilience.colostate.edu/
- IN-CORE landing page
  - https://incore.ncsa.illinois.edu/
- Source code at GitHub
  - https://github.com/IN-CORE
- Conda packages
  - https://anaconda.org/IN-CORE
- Support:
  - Email: incore-dev@lists.Illinois.edu
  - Documentation: tutorials, tips, and FAQ
  - Webinars
  - Slack channel will be available soon