

Broadening Participation in Visualization 2020

BUILDING DATA VISUALIZATION CAPACITY

Summer Webinar Series

July 21 - 24, 2020

UTSA The University of Texas
at San Antonio™

Byrd Data Visualization Laboratory

PURDUE
UNIVERSITY

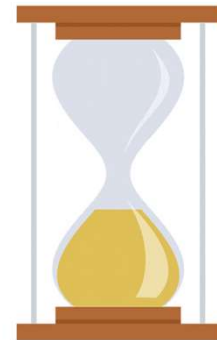
Website: <https://tinyurl.com/yctfhc6l>

Broadening Participation in Visualization 2020

©BPBiz

Administrivia & Logistics

- Take a moment to provide feedback for yesterday's session: You've got data now what?
- Feedback link located at the bottom of Day 3 Web page, can also be accessed from here <https://tinyurl.com/y3hlhs4m>



About Me

- Vetria Byrd, PhD
- Assistant Professor
- Purdue University
 - Department of Computer Graphics Technology
 - Polytechnic Institute
- Byrd Data Visualization Lab, Director
- Founder and Organizer of BPViz: Broadening Participation in Visualization Workshop

2020 | [2019](#) | [2018](#) | 2016 | 2014



Vetria Byrd
Assistant Professor
Byrd Data Visualization
Lab, Director
Purdue University

Webinar Organizers



Vetria Byrd
Assistant Professor
Byrd Data Visualization
Lab, Director
Purdue University



Ritu Arora
Assistant Vice President of
Research Computing
University of Texas at San
Antonio



Brent League
Director of Research
Computing Support
University of Texas at San
Antonio

Webinar Goals and Objectives

Goal

Introduce participants to data visualization

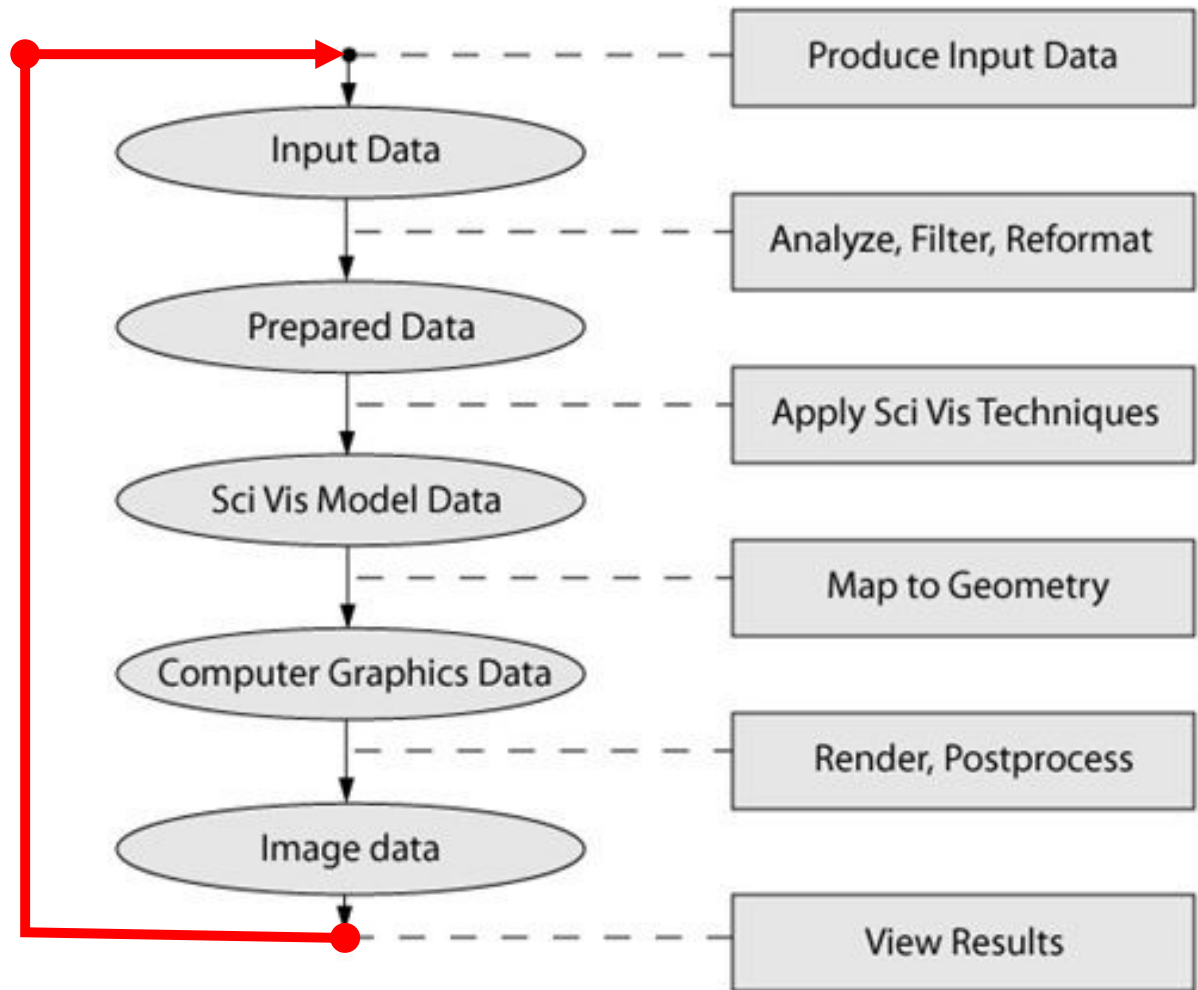
Objectives

1. Examine what data looks like, define data visualization.
2. Illustrate how data visualization can improve understanding of the story within the data.
3. Introduce the data visualization process.
4. Explore different data visualization paths.

1 minute Recap

Day 3: Introduction to Scientific Visualization

Scientific Visualization Pipeline



VISUALIZING DATA USING PARAVIEW

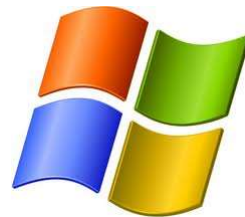
- Three Basic Steps:
 - First your data must be **read** into ParaView
 - Next, you may apply any number of **filters** that process the data to generate, extract, or derive features from the data
 - Finally, a viewable image is **rendered** from the data

ParaView

- Multi-platform parallel data analysis and visualization application
- Mature, feature-rich interface
- Good for general purpose, rapid visualization



Mac



Windows



Linux

We are here

Schedule	Topic	Skill Level
Day 0	Download and Install	Beginner
Day 1	Introduction to data visualization	Beginner
Day 2	You've got data now what?	Beginner/Intermediate
Day 3	Introduction to scientific Visualization	Intermediate/Advanced
Day 4	Visualization hackathon	Advanced

PULLING IT ALL TOGETHER

PANDAS, PYTHONS AND BEARS, OH MY!

Day: Friday, July 24, 2020

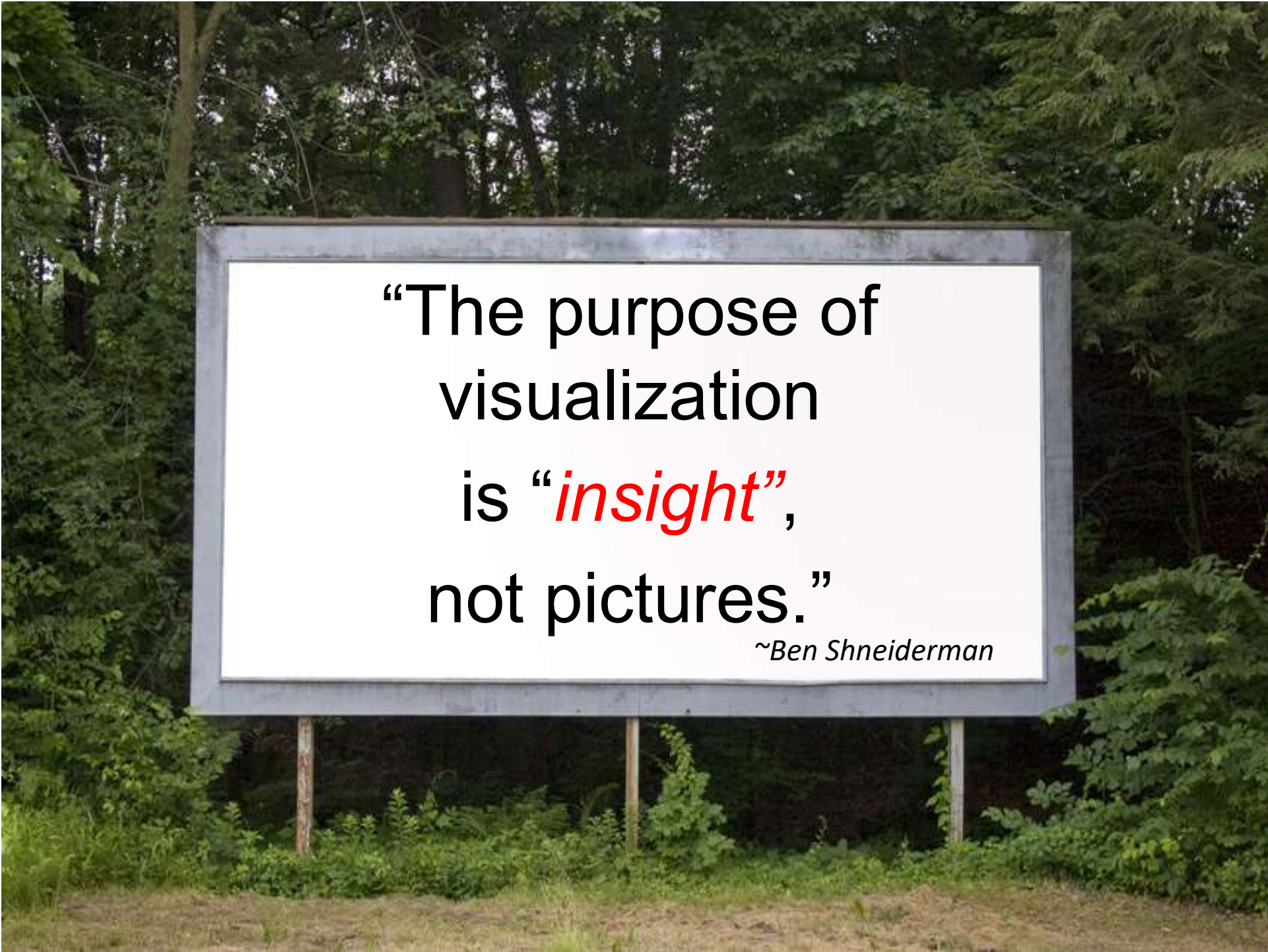
Time: 1PM - 3PM (CDT)/2PM - 4PM (EDT)

Skill level: Advanced

In this webinar we bring all of the knowledge gained this week together. A brief overview of advanced tools and additional resources will be provided. The main event, however, is the Visualization Hackathon!

Why We Visualize Data



A rectangular sign with a white background and a grey border is mounted on three wooden posts. The sign is set against a backdrop of dense green trees and foliage. The text on the sign is centered and reads: "The purpose of visualization is *insight*, not pictures."

“The purpose of
visualization
is “*insight*”,
not pictures.”

~Ben Shneiderman

Insight Enables



Storytelling: The Next Step for Visualization

Robert Kosara and Jock Mackinlay, Tableau Software, Seattle

Kosara, R., & Mackinlay, J. (2013). Storytelling: The next step for visualization. *Computer*, 46(5), 44-50.

“Insight” Leads to . .

Explanation Visualizing Spatial Relationships



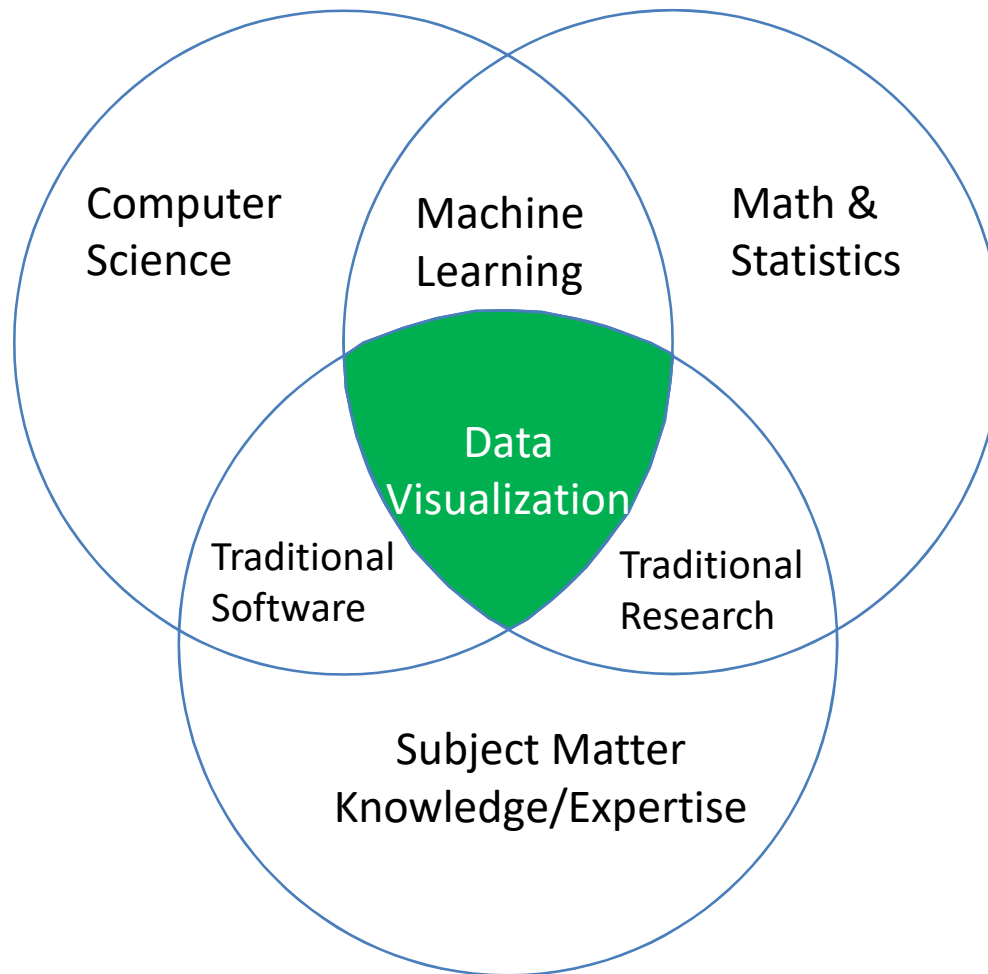
<http://datafl.ws/197>



<http://datafl.ws/198>

Muehlenhaus, I. (2012). **Chapter 8, Visualizing Spatial Relationships**, Visualize This: The Flowing Data Guide to Design, Visualization, and Statistics, pp 271-326.

Data Science



Adapted from Johnson, J. (2017). Data science & computing across the curriculum. *Journal of Computing Sciences in Colleges.*, 32, 187-188.

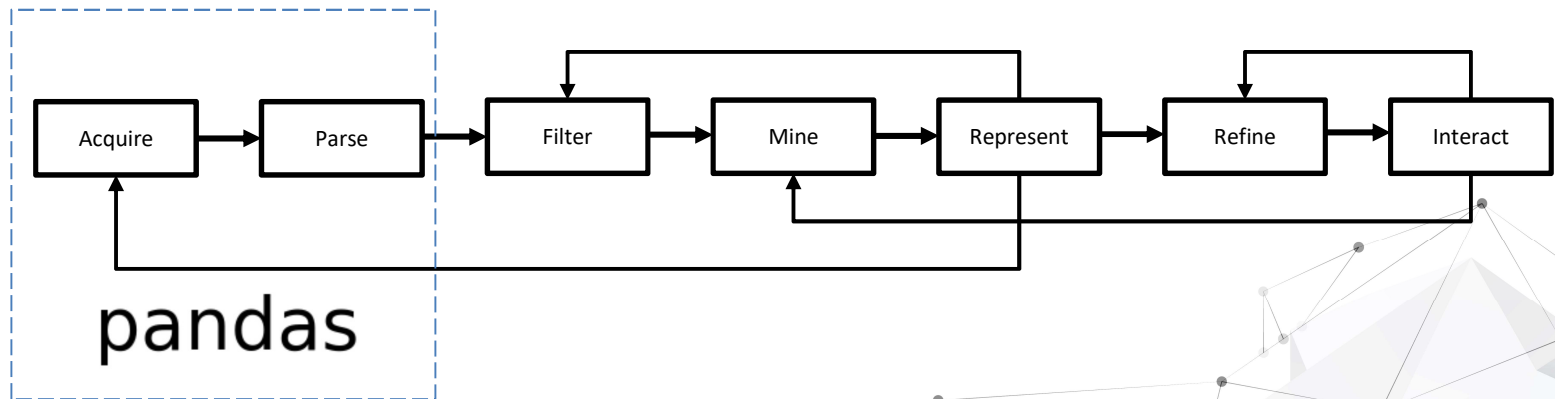


Python for Data Science

Data Visualization Skills & Tools

Pandas

Python Data Analysis Library (Some Basics)



Import convention:
`import pandas as pd`

Essential Python Library: Pandas

Pandas (<http://pandas.pydata.org>)

- Provides high-level data structures and functions designed to make working with structured or tabular data fast, easy, and expressive.
- Key objects
 - The *DataFrame*: a tabular, column-oriented data structure with both row and column labels, and
 - The *Series*, a one-dimensional labeled array object.
- Provides sophisticated indexing functionality to make it easy to reshape, slice and dice, perform aggregations and select subsets of data
- Handles:
 - Data structures with labeled axes supporting automatic or explicit data alignment
 - Integrated time series functionality
 - Same data structures handle both time series data and non-time series data

Getting Started with Pandas

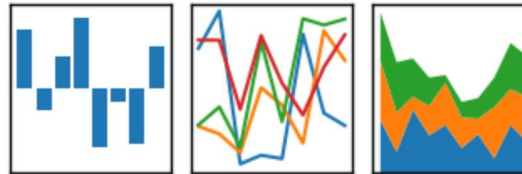
https://pandas.pydata.org



software

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



[home](#) // [about](#) // [get pandas](#) // [documentation](#) // [community](#) // [talks](#) // [donate](#)

Python Data Analysis Library

pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the [Python](#) programming language.

pandas is a [NumFOCUS](#) sponsored project. This will help ensure the success of development of *pandas* as a world-class open-source project, and makes it possible to [donate](#) to the project.

A Fiscally Sponsored Project of

NUMFOCUS
OPEN CODE = BETTER SCIENCE

v0.25.0 Final (July 18, 2019)

This is a major release from 0.24.2 and includes a number of API changes, new features, enhancements, and performance improvements along with a large number of bug fixes.

VERSIONS

Release

0.25.0 - July 2019

[download](#) // [docs](#) // [pdf](#)

Development

0.26.0 - September 2019

[github](#) // [docs](#)

Previous Releases

0.24.2 - [download](#) // [docs](#) // [pdf](#)

0.24.1 - [download](#) // [docs](#) // [pdf](#)

0.24.0 - [download](#) // [docs](#) // [pdf](#)

0.23.4 - [download](#) // [docs](#) // [pdf](#)

0.23.3 - [download](#) // [docs](#) // [pdf](#)

0.23.2 - [download](#) // [docs](#) // [pdf](#)

0.23.1 - [download](#) // [docs](#) // [pdf](#)

0.23.0 - [download](#) // [docs](#) // [pdf](#)

0.22.0 - [download](#) // [docs](#) // [pdf](#)

0.21.1 - [download](#) // [docs](#) // [pdf](#)

0.21.0 - [download](#) // [docs](#) // [pdf](#)

Pandas: Basics

Contains data structures and data manipulation tools designed to make data cleaning and analysis fast and easy in Python.

Often used in tandem with numerical computing tools like [NumPy](#) and data visualization libraries like [Matplotlib](#)

Pandas: Some Highlights

Used to load data into python from many different file formats

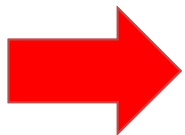
- Time series operations
- Data Frames represents collection of time series
- Can select all data points at a particular time.
- Easy to resample time series data.
- Can specify aggregate data and
- Moving window function

Getting Started with pandas

```
In [ ]: import pandas as pd
```

```
In [ ]: from pandas import Series, DataFrame
```

```
In [ ]: import numpy as np
np.random.seed(12345)
import matplotlib.pyplot as plt
plt.rc('figure', figsize=(10, 6))
PREVIOUS_MAX_ROWS = pd.options.display.max_rows
pd.options.display.max_rows = 20
np.set_printoptions(precision=4, suppress=True)
```



To run a command in windows interface:
press Shift + Enter Key

Pandas Data Frames

Documentation

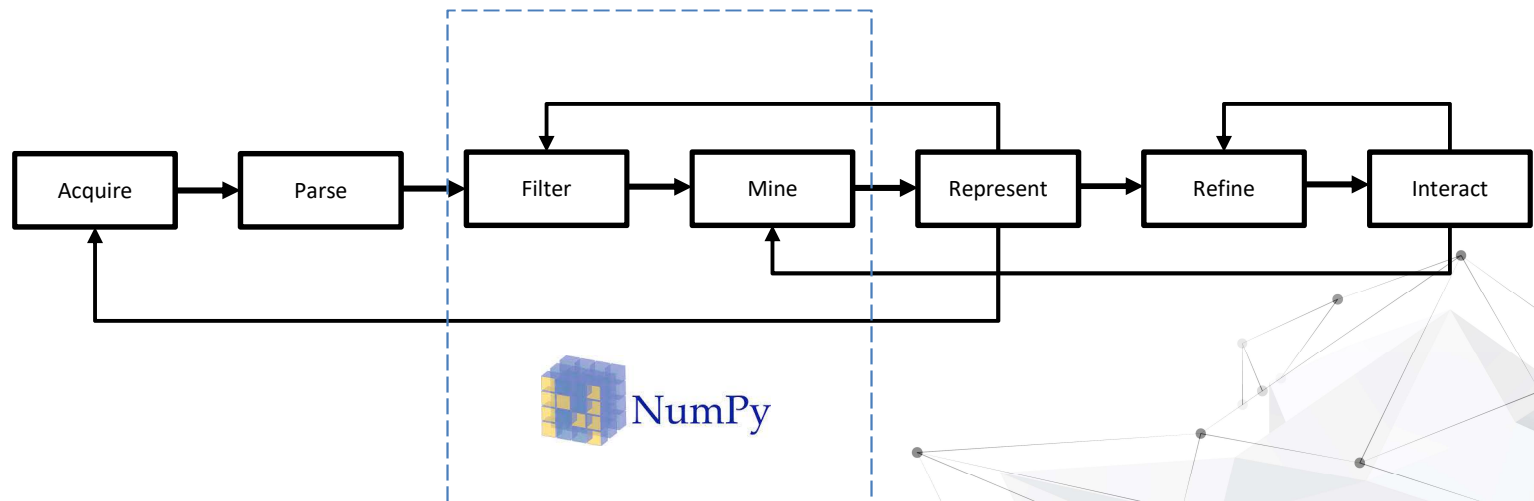
<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html>

Graphical Explanation

<https://www.geeksforgeeks.org/python-pandas-dataframe/>

NumPy

Python Data Analysis Library (Some Basics)



Import convention:
Import numpy as np

Getting Started with NumPy

https://numpy.org

Search



Scipy.org

NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

NumPy is licensed under the [BSD license](#), enabling reuse with few restrictions.

Getting Started

To install NumPy, we strongly recommend using a *scientific Python distribution*. See [Installing the SciPy Stack](#) for details.

Many high quality online tutorials, courses, and books are available to get started with NumPy. For a quick introduction to NumPy we provide the [NumPy Tutorial](#). We also recommend the [SciPy Lecture Notes](#) for a broader introduction to the scientific Python ecosystem.

For more information on the SciPy Stack (for which NumPy provides the fundamental array data structure), see [scipy.org](#).

Documentation

The most up-to-date NumPy documentation can be found at [Latest \(development\) version](#). It includes a user guide, full reference documentation, a developer guide, meta information, and "NumPy Enhancement Proposals" (which include the NumPy Roadmap and detailed plans for major new features).

About NumPy

Community

License

Code of Conduct

Old array packages

Essential Python Library: NumPy

NumPy (<http://numpy.org>)

- Aka: Numerical Python
- Provides the data structures, algorithms, and library glue needed for most scientific applications involving numerical data in Python.
- Contains
 - A fast and efficient multidimensional array object ndarray
 - Functions for performing element-wise computations with array or mathematical operations between arrays
 - Tools for reading and writing array-based datasets to disk
 - Linear algebra operations, Fourier transform, and random number generation
 - A mature C API to enable Python extensions and native C or C++ code to access NumPy data structures and computational facilities.
- Primary uses in data analysis is as a container for data to be passed between algorithms and libraries.

Getting Started with NumPy

```
In [2]: ▶ import numpy as np
```

```
In [3]: ▶ data = {i : np.random.randn() for i in range(7)}
```

```
In [4]: ▶ data
```

```
Out[4]: {0: -0.3135476447310109,  
1: 1.8227248210238562,  
2: -0.5632782805883568,  
3: 1.6251351482161371,  
4: 0.631008411585496,  
5: 2.2225544468005927,  
6: -1.7667730376552777}
```

The NumPy ndarray

:A Multidimensional Array object

- A fast, flexible container for large datasets in Python
- Arrays enable you to perform mathematical operations on whole blocks of data using similar syntax to the equivalent operations between scalar elements.
- Creating arrays in NumPy

<https://towardsdatascience.com/getting-started-with-numpy-59b22df56729>

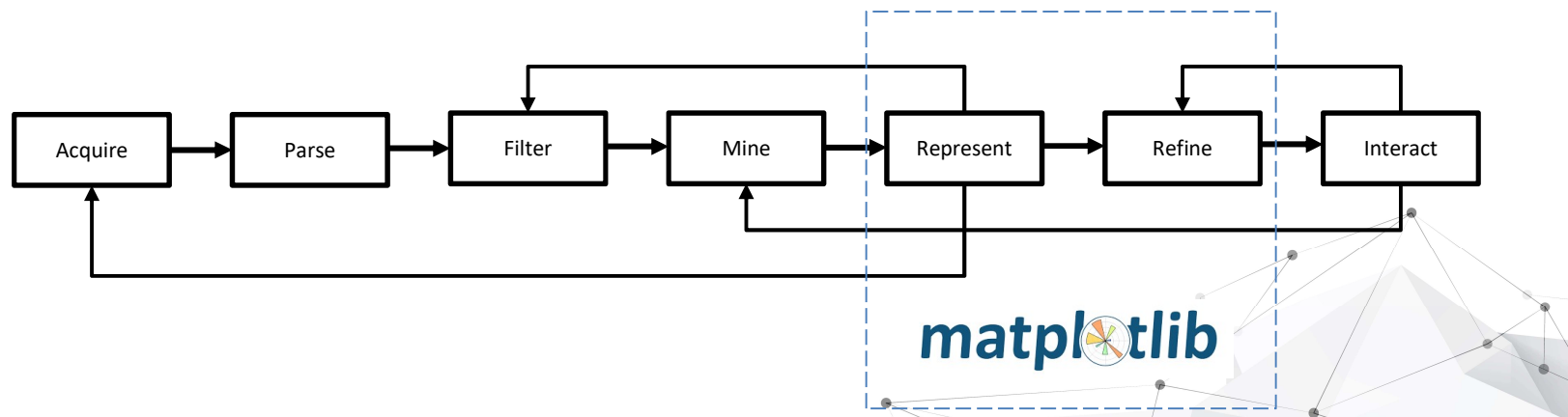
NumPy Resources

<http://cs231n.github.io/python-numpy-tutorial/>

<http://cs231n.github.io/python-numpy-tutorial/#numpy>

Matplotlib

Python Visualization Library (Some Basics)



Import convention:
Import matplotlib.pyplot as plt

Four types of Visualizations

GEORGES GRINSTEIN (KEYNOTE PRESENTATION, VINCI 2016)

- **Exploratory**
 - Have no hypotheses about the data
 - Explore data interactively as undirected searches
- **Confirmatory**
 - Have specific hypotheses about the data
 - Goal-oriented examination of the hypotheses
- **Presentation**
 - Facts to be presented are fixed a priori
 - Select appropriate presentation techniques
- **Interactive**
 - Interactions with a pre-defined animation

Getting Started with Matplotlib

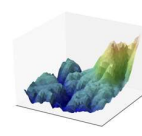
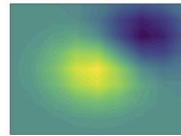
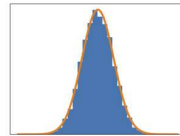
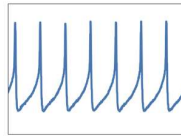


Fork me on GitHub

[home](#) | [examples](#) | [tutorials](#) | [API](#) | [contents](#) »

[modules](#) | [index](#)

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.



Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the [sample plots](#) and [thumbnail gallery](#).

For simple plotting the `pyplot` module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Installation

Visit the [Matplotlib installation instructions](#).

Documentation

This is the documentation for Matplotlib version 3.1.1.

Quick search

Go

Matplotlib 3.0 is Python 3 only.
For Python 2 support, Matplotlib 2.2.x will be continued as a LTS release and updated with bugfixes until January 1, 2020.

Support Matplotlib

Essential Python Library: Matplotlib

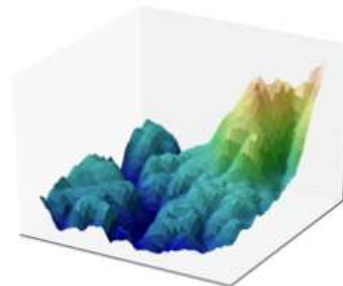
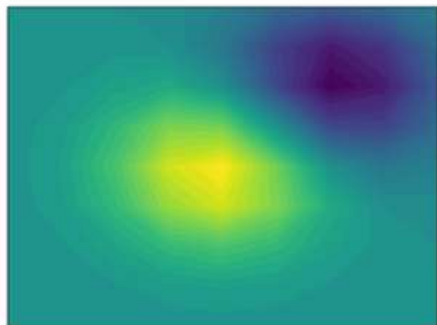
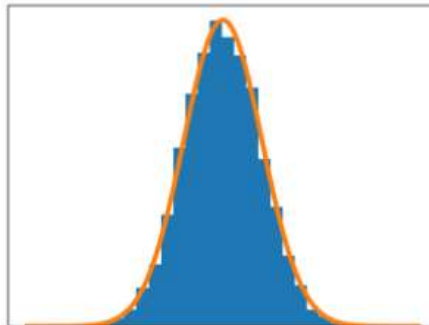
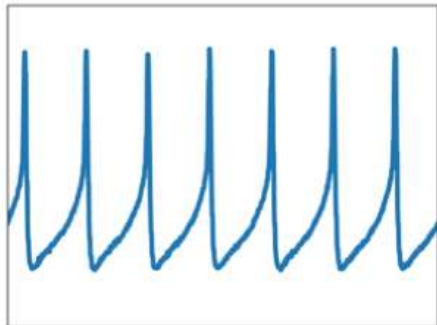
Matplotlib (<http://matplotlib.org>)

- Most popular Python Library for producing plots and other two-dimensional data visualizations.
- Was designed for creating plots suitable for publication.
- The most widely used visualization library available to Python programmers.

Sample plots in Matplotlib

https://matplotlib.org/tutorials/introductory/sample_plots.html#

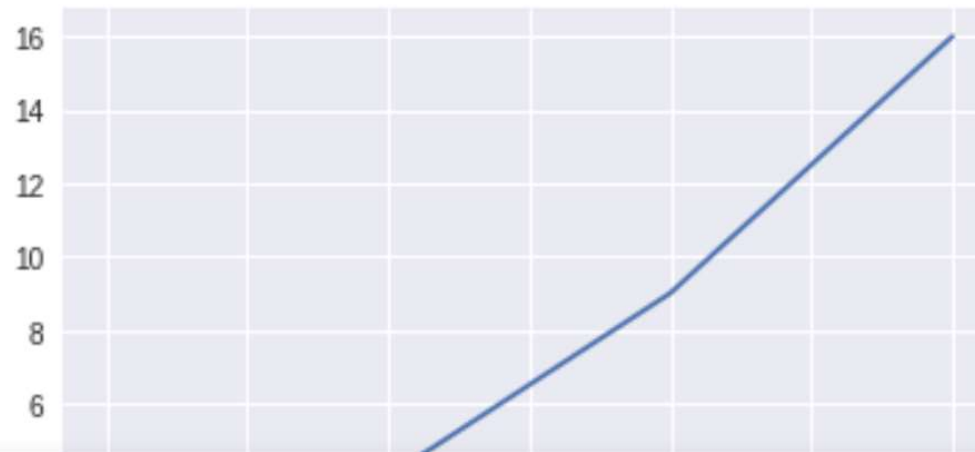
<http://cs231n.github.io/python-numpy-tutorial/#matplotlib>



Simple Example

```
import matplotlib.pyplot as plt
import numpy as np

plt.plot([1,2,3,4],[1,4,9,16])
plt.show()
```



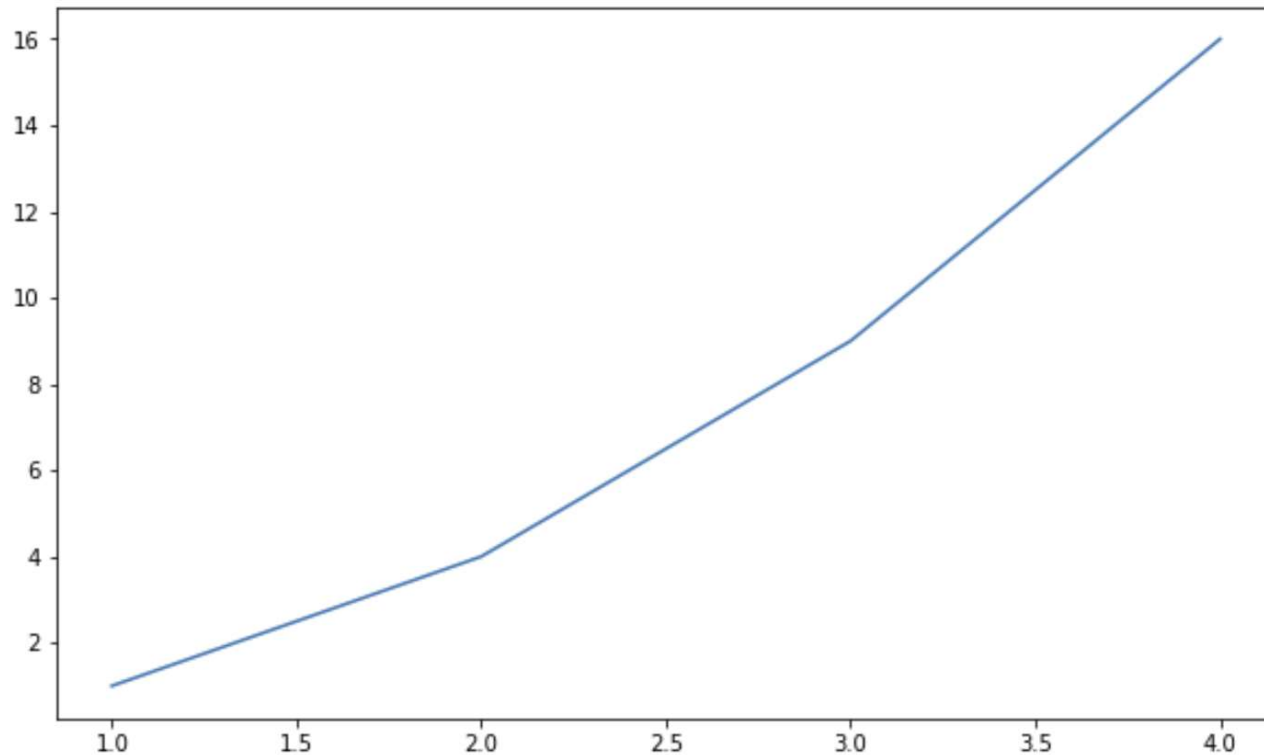
From Jupyter Notebook

```
In [35]: ▶ import matplotlib.pyplot as plt
```

```
In [36]: ▶ import numpy as np
```

```
In [37]: ▶ plt.plot([1,2,3,4],[1,4,9,16])
```

```
Out[37]: [<matplotlib.lines.Line2D at 0xcc9a278>]
```



Matplotlib Resources

<https://towardsdatascience.com/matplotlib-tutorial-learn-basics-of-pythons-powerful-plotting-library-b5d1b8f67596>

<https://realpython.com/python-matplotlib-guide/>

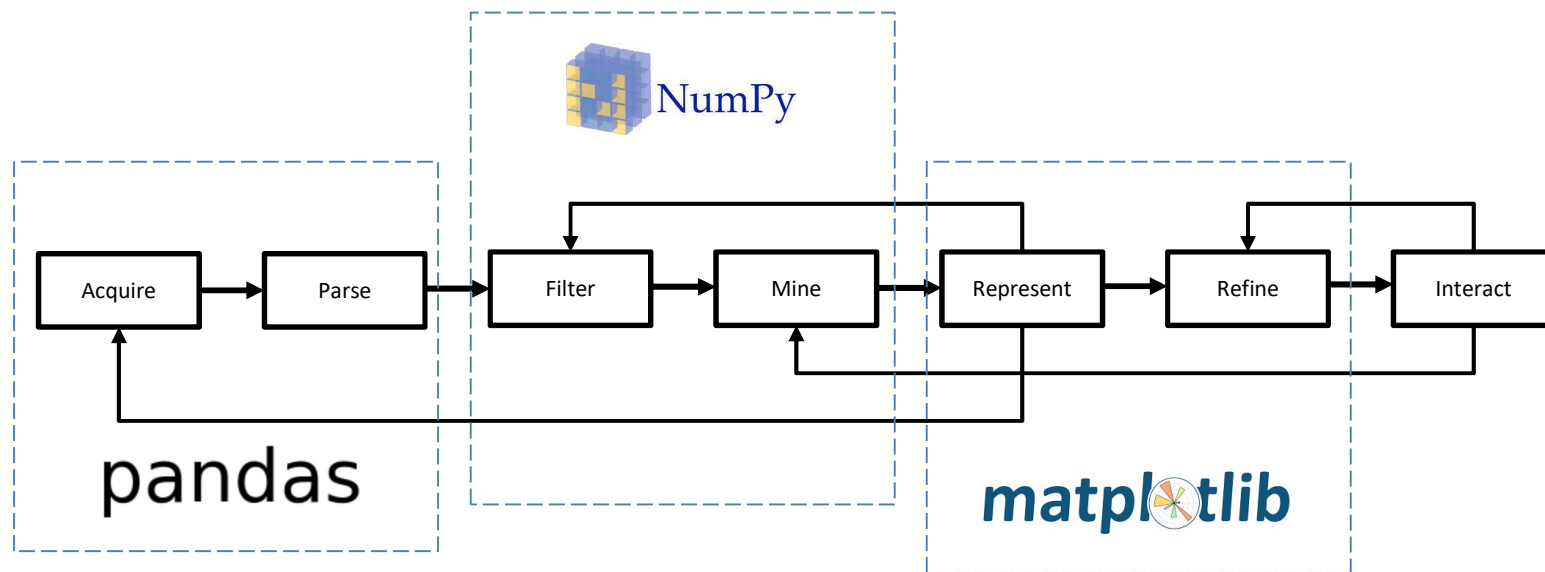
When should you think about visualizing your data?

As early
and
often!



<http://howtolaunch.com/howtolaunch/reach-your-audience-early-and-often/>

Recap: Data Visualization Tools for Insight

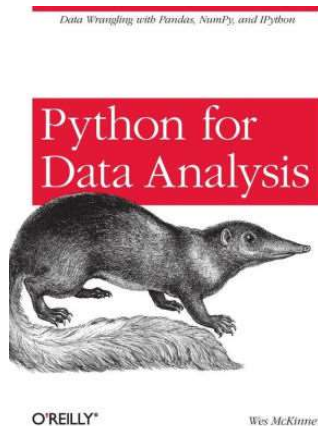


Adopted from Fry, B. (2008). Chapter 1, Visualizing data (Safari Books Online). Sebastopol, Calif.: O'Reilly Media.

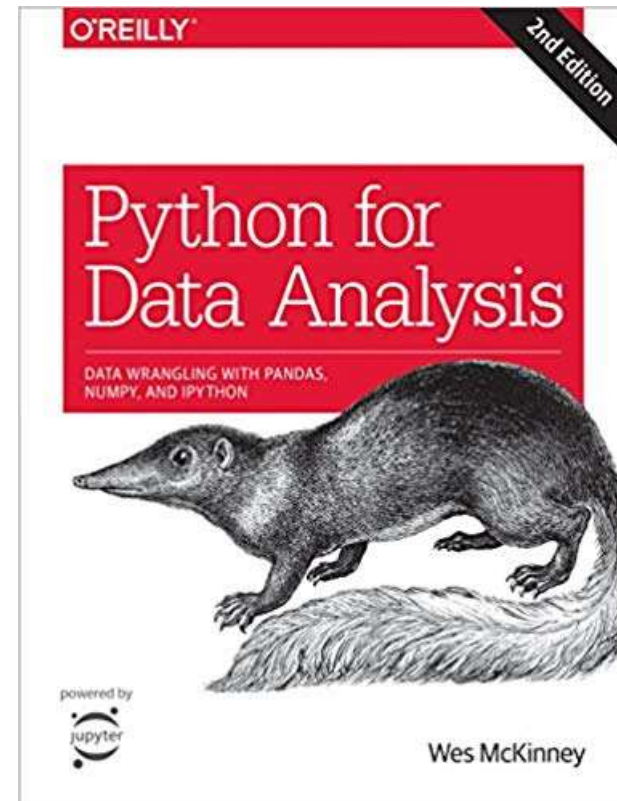
Main Source for Python Libraries

McKinney, W. (2017). Python for data analysis : Data wrangling with Pandas, NumPy, and IPython (Second ed.). O'Reilly Media.

GitHub: <https://github.com/wesm/pydata-book>
Sample data and code from book available



1st Edition



2nd Edition

PURDUE
POLYTECHNIC

Web pages referenced

Pandas Links

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html>
<https://www.geeksforgeeks.org/python-pandas-dataframe/>
<http://pandas.pydata.org>)

NumPy Links

<http://numpy.org>
<https://towardsdatascience.com/getting-started-with-numpy-59b22df56729>
<http://cs231n.github.io/python-numpy-tutorial/>
<http://cs231n.github.io/python-numpy-tutorial/#numpy>

Matplotlib Links

<http://matplotlib.org>
<https://towardsdatascience.com/matplotlib-tutorial-learn-basics-of-pythons-powerful-plotting-library-b5d1b8f67596>
<https://realpython.com/python-matplotlib-guide/>
https://matplotlib.org/tutorials/introductory/sample_plots.html#
<http://cs231n.github.io/python-numpy-tutorial/#matplotlib>

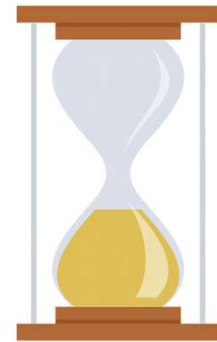
Other

Degrees in Data Visualization: <https://polytechnic.purdue.edu/degrees/data-visualization>
Human Genome Project: <https://pradipjntu.files.wordpress.com/2011/05/molecularmachine.jpg>
<http://howtolaunch.com/howtolaunch/reach-your-audience-early-and-often/>

Feedback survey

This weekend you will receive an email asking for feedback for the **webinar series**. Please take a moment to complete it.

Thank you!



Thank You

To the participants for your time,
University of Texas at San Antonio,
Dr. Ritu Arora and Brent League for hosting the
Webinar Series!



Ritu Arora

Assistant Vice President of
Research Computing
University of Texas at San
Antonio



Brent League

Director of Research
Computing Support
University of Texas at San
Antonio



Vetria L. Byrd

Assistant Professor

Computer Graphics Technology

vlbyrd@purdue.edu

<https://polytechnic.purdue.edu/profile/vbyrd>

<http://web.ics.purdue.edu/~vbyrd/>



@VByrdPhD, @BPViz, @ByrdVisLab

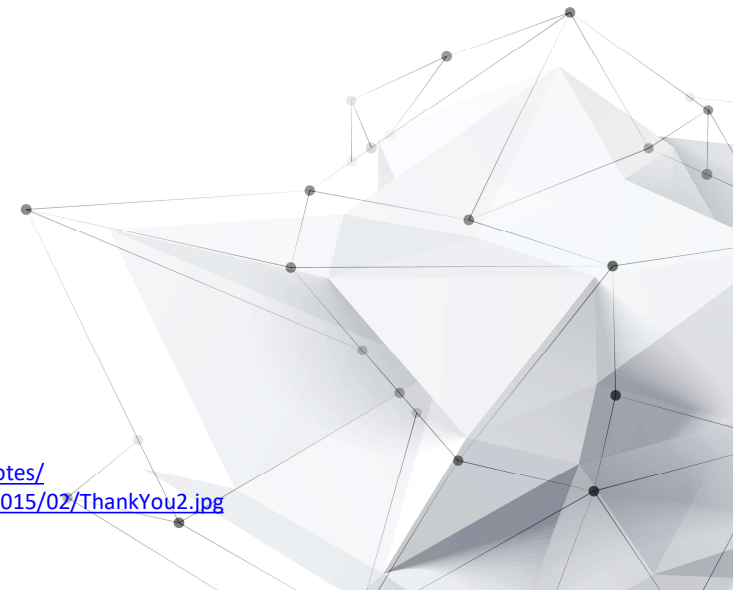
Purdue Polytechnic Institute

PURDUE
POLYTECHNIC

Thank You Image Source:

<http://careerconfidential.com/category/thank-you-notes/>

<http://careerconfidential.com/wp-content/uploads/2015/02/ThankYou2.jpg>



The pièce de résistance!

