

Building your First Python Analytics Solution

GETTING STARTED WITH PYTHON FOR ANALYTICS



Janani Ravi

CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Python for data analytics

Installing Python on Windows and MacOS

Running Python from a terminal window

Installing Python packages using pip

Installing packages within a virtual environment

Development environments

Code editors and execution environments

Prerequisites and Course Outline

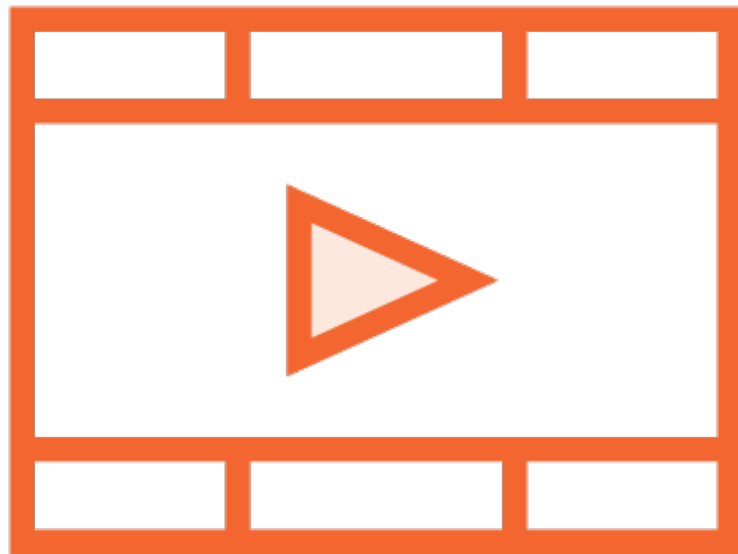
Prerequisites



Basic Python programming

**Ability to install tools and packages on
your computer**

Prerequisites



Python Fundamentals

Course Outline



Getting started with Python for analytics

Working with Python using Anaconda

Working with Python using other IDEs

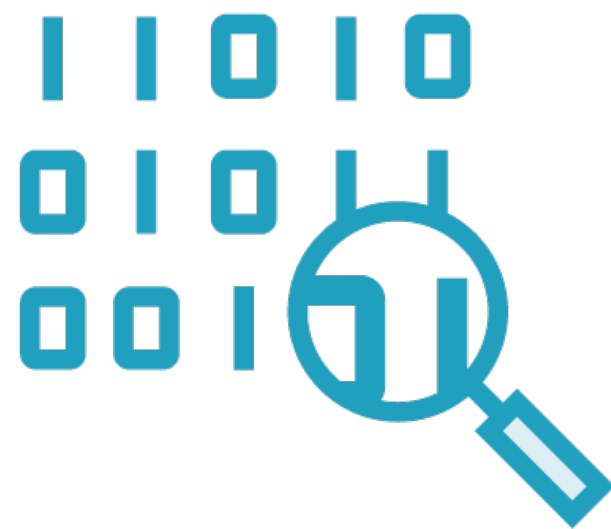
Working with Python on the cloud

Python for Data Analytics

“When the facts change, I change my mind. What do you do, Sir?”

John Maynard Keynes

Thoughtful, Fact-based Point of View



Fact-based

Built with
painstakingly
collected data



Thoughtful

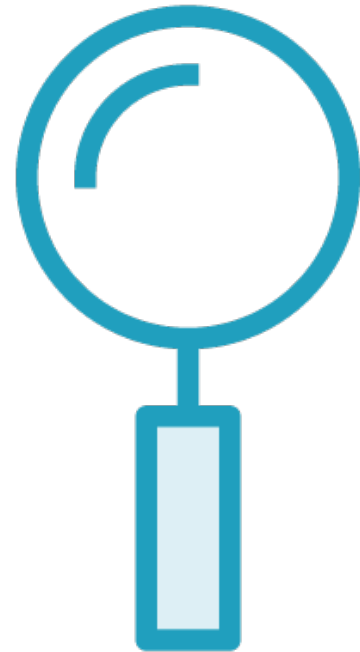
Balanced, weighing
pros and cons



Point of View

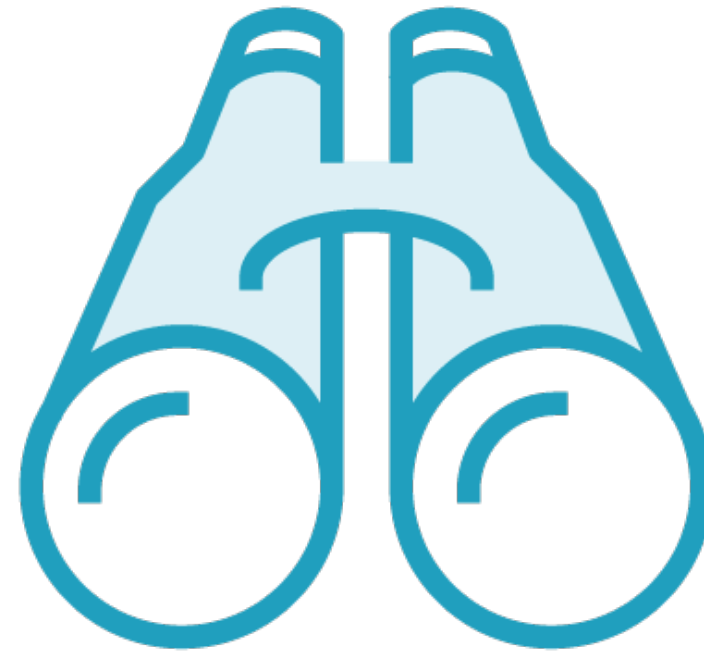
Prediction,
recommendation,
call to action

Two Sets of Statistical Tools



Descriptive Statistics

Identify important elements in a dataset



Inferential Statistics

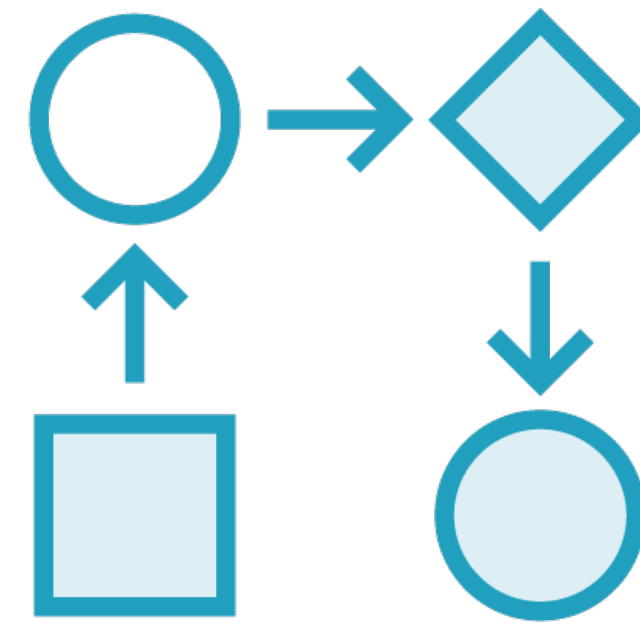
Explain those elements via relationships with other elements

Two Hats of a Data Professional



Find the Dots

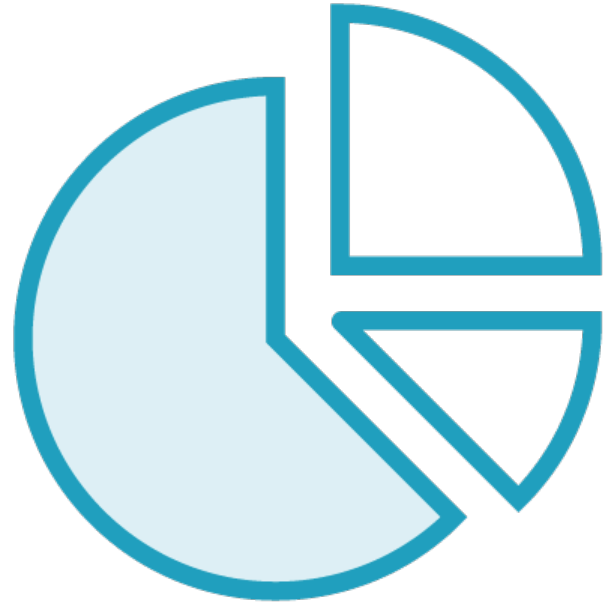
Identify important elements in a dataset



Connect the Dots

Explain those elements via relationships with other elements

Finding the Dots

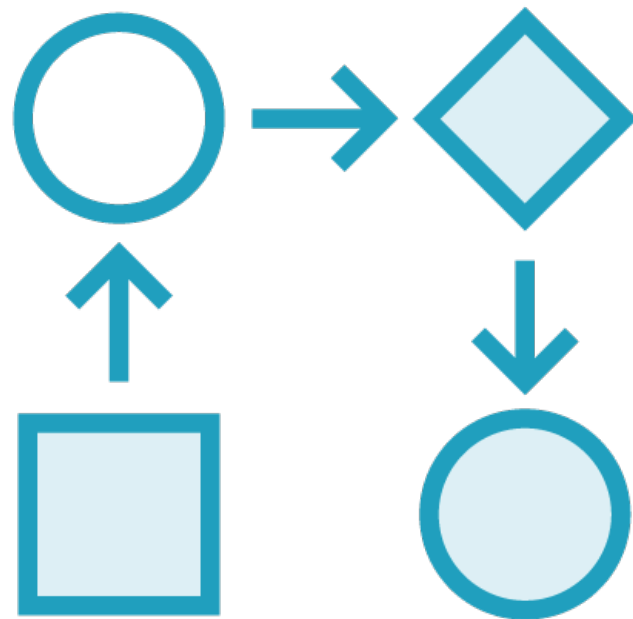


Data is more and more plentiful

However careful handling is needed

- Missing values
- Outliers
 - Genuine outliers
 - Erroneously measured points

Connecting the Dots



Spreadsheets

Programming languages

- In-memory processing
- Distributed processing

SQL

- Relational databases
- Data warehouses

Python has truly democratized
data analysis more than any
technology since Microsoft Excel

Essential Analytical Building Blocks

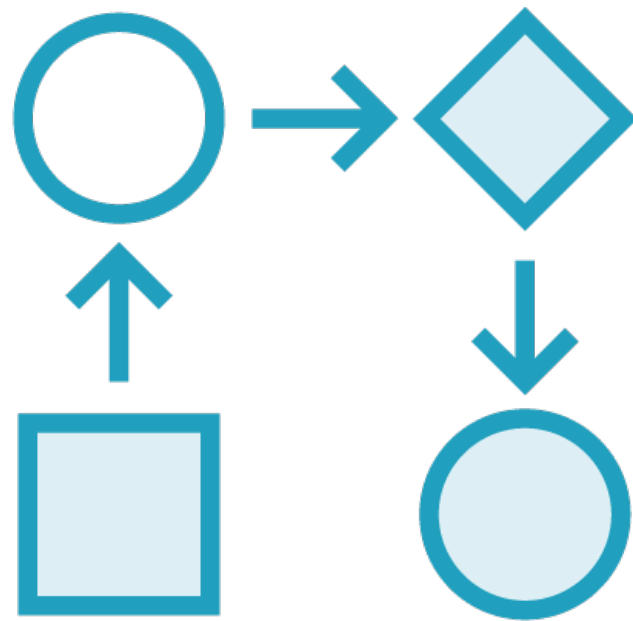
Conditional Execution

Interconnected Calculations

**Repeated Execution
(Iteration)**

**Re-use of Logic
(Composition)**

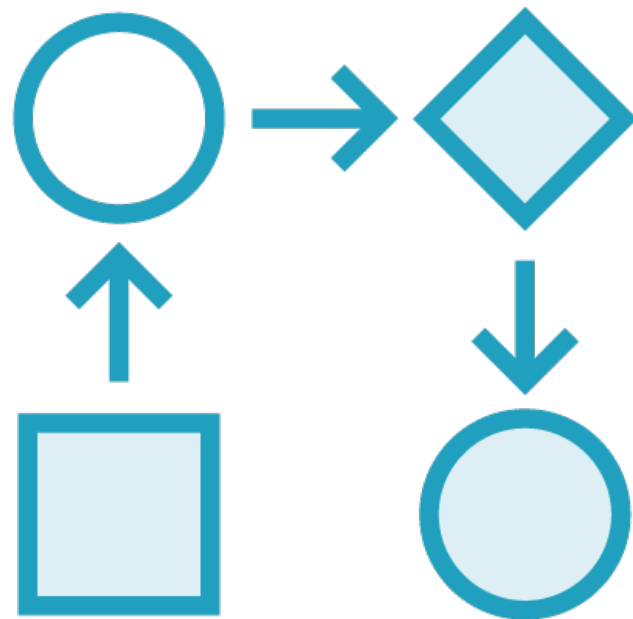
Choices of Technology



Spreadsheets are best for

- Complex inter-connected calculations
- Rapid prototyping

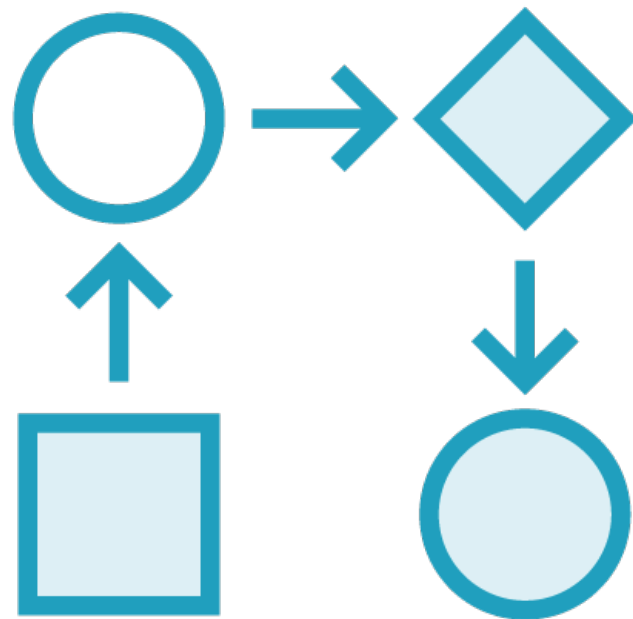
Choices of Technology



SQL is best for

- Iterating over independent rows
- Simple syntax

Spreadsheets for Analytics

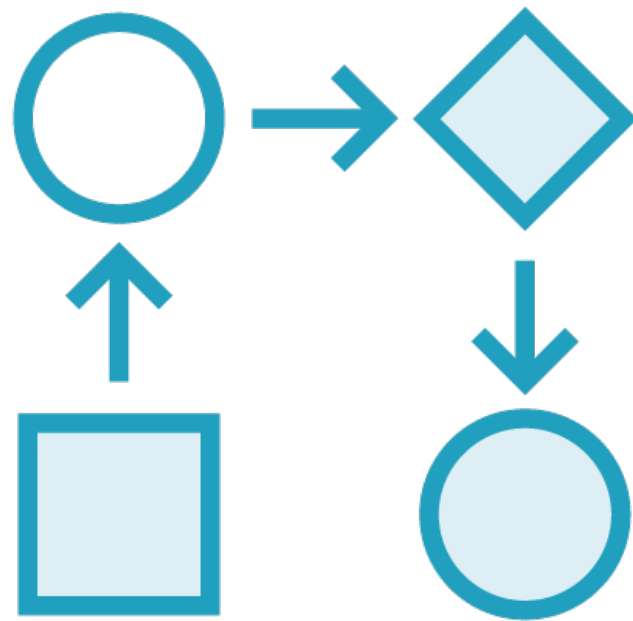


Conditionals: if() function within cells

Iteration: Copy-paste, or worse, macros

Composition: Not possible

SQL Databases for Analytics

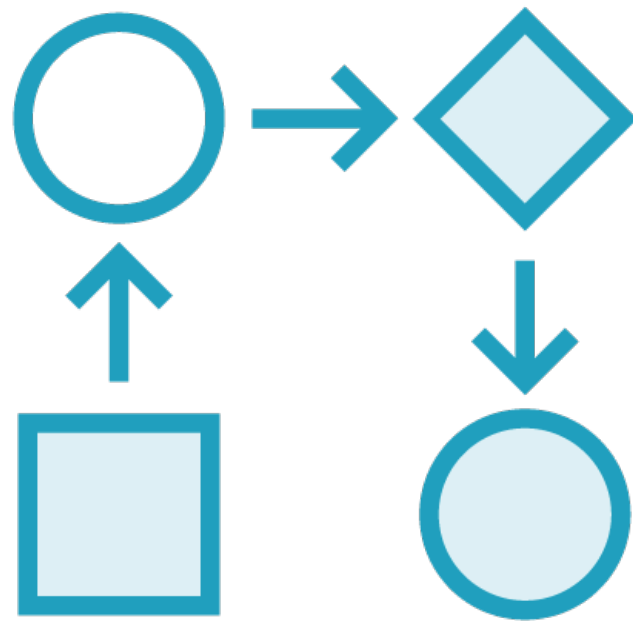


Conditionals: if() function within queries

Iteration: Queries, cursors

Composition: Views, stored procedures

Python for Analytics



Programming languages offer full support for analytical operations

Conditionals: If-else

Iteration: For and while loops

Composition: Functions

Python Development Environments

Python for Data Analytics

On the one hand

Python combines Excel's ease-of-prototyping with SQL's simple syntax

But on the other

Python has yet to prove itself as robust as Java for big projects

Jupyter Notebook

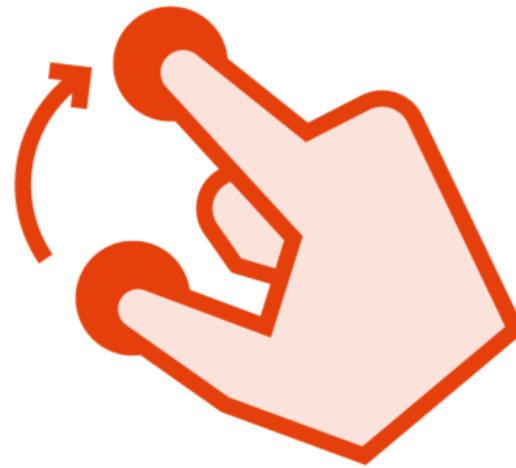
Open-source web application that allows interactive development in Python and several other languages.

Python on Jupyter



Accessible

Web-based and free -
no cost or installation
hassle



Interactive

Read-Evaluate-Print-
Loop for instant
feedback



Powerful

Integrations with
cloud and distributed
technologies

Jupyter is fine for prototyping, but for enterprise-scale development, IDEs still matter

Integrated Development Environment

Application that makes software development easy - usually by combining a code editor; build, execute and debug tools; and source control.

IDEs and Python



**Jupyter is an execution environment,
not an IDE**

**For large projects that span
notebooks, seriously consider an IDE**

Features of Most IDEs



Code editor

- Auto-completion
- Intuitive code/text highlighting

Execution environment

- Run
- Build
- Debug

Source control integration

Full IDEs



PyCharm

IDLE

Spyder

Thonny

Eclipse with PyDev

Visual Studio

Code Editors



SublimeText

Emacs

Vim

Visual Studio Code

Online IDEs



Online playgrounds e.g. repl.it

Interactive execution environment

Debugging and listing

Files and third-party packages

Hosting and deployment support

Packages and Libraries

Libraries in Python



Python has an incredibly rich set of third-party libraries

Made available by vast community of developers

Hosted in a comprehensive repository called PyPI

Perhaps single biggest driver of popularity of Python

Libraries in Python



PyPI is the *Python Package Index*

Libraries in Python are called packages

These packages encapsulate code in files called modules

Package

A unit of directories and files that can be easily imported for use in a Python program. Can contain namespaces, modules (.py files), and nested packages.

Creating Packages in Python



Anyone can package up their code for use by other developers

Package code in specific structure

Publish it for inclusion on PyPI

Using Packages in Python



**Packages can be easily installed
for use in Python programs**

Two common ways to do this

- Conda
- Pip

Packages in Python

Pip

Install from PyPI

Install Python packages only

Installs “wheels” (source distributions)

To install other tools (e.g. interpreter) need a package manager or installer

Can not create isolated environments, need to use virtualenv in addition

Conda

Install from Anaconda repository

Install packages as well as other tools

Installs binaries (even for python packages)

Can also install any binaries including other language libraries, interpreters

Can create isolated environments to manage different version of Python

Demo

**Windows: Installing Python and using
Pip to install packages**

Demo

MacOS: Using brew to install Python 3

Demo

MacOS: Using Pip to install packages

Demo

Installing and working with virtual environments

Demo

**Editing Python scripts using nano and
vim**

Demo

**Editing Python scripts using
SublimeText**

Demo

Using online editors to write Python code

Summary

Python for data analytics

Installing Python on Windows and MacOS

Running Python from a terminal window

Installing Python packages using pip

Installing packages within a virtual environment

Development environments

Code editors and execution environments