Fundamental Concepts in Data Insight:

Programming & Automation

Fundamentals for a General Audience

Programming & Automation

- What is Automation?
  - How do organizations get an ROI on automation?
  - What types of Insight can Automated?
  - Why replace human-driven applications with programs?
  - WB. What tools do analysts & scientists use?
  - Review: What tools do analysts and scientists use?
- Review: What is Python?
  - Demo: How do you write a program?
  - Demo: How do you use libraries in a program?
  - Review: What are the advantages of python?
  - Review: What is Programming?
  - Review: How do you program Insight?
- Reflection
  - Individual Project

What is Automation?

- automation is the process of replacing or assiting a human
    - to increase the productivity of a task

How do organizations get an ROI on automation?

- many businesses process are very hard to automate
  - requiring high-quality decision-making
  - which cannot be automated
- automation generates an ROI when
  - the target process is low-producivity
  - with human operators
  - because of the rate/scale of decision-making or action

What types of Insight can Automated?

Why replace human-driven applications with programs?

- moving away from graphical user interfaces ("gui")
  - is a major challenge for analysts
- modern automation requires them to be programmers
  - however they can be given limited code

WB. What tools do analysts & scientists use?

WB. What tools do analysts & scientists use?

Review: What tools do analysts and scientists use?

- excel: spreadsheets
    - spreadsheets hide formula
    - expose values
    - provide graphical exploration tools
    - allow analysts to think visually
        - and about data only
- mssql: sql
    - expose formula which can be shared easily
    - produce tables
    - no graphical tools
    - often results imported into excel for exploration
- jupyter: notebooks
    - expose formula/algorithms which can be shared easily
    - expose values
    - provide graphical exploration tools
    - allow analysts to think visually
    - allow analysts to think algorithmically
- pycharm: development environments
    - raw text for specifying algorithms
    - text editor also runs algorithms
    - has features for software engineers
    - used to build automation systems

Demo: How do you write a program?

Demo: How do you use libraries in a program?

Review: What is Python?

python is one of the most popular human-readable languages for writing algorithms

Review: What are the advantages of python?

With wide adoption there are many available programmers.

Python is regarded as very human-readable and therefore easier to pick up than altneratives.

Python has been adapted with non-python state-of-the-art data tools which enable it to have best-in-class performance on data problems

Many key insight organizations (google, facebook, et al.) build and release insight-focused libraries in python

Review: What is Programming?

- digital computers are electrical systems
  - which can affect electrical devices in programmable ways
- we can alter the state of an lcd screen
  - with a cpu and graphics card
  - and we can change the behaviour of both devices with a program

Review: How do you program Insight?

an algorithm is a sequence of steps with a well-defined input and output

algorithms are company-agnostic, they work if the input is correct

libraries are pre-written bundles of algorithms which solve analytical problems

the primary job of a programmer-analyst is to assmble pre-existing algorithms together in order to solve the business problem

Individual Project

- walk throughs: consult the accompanying notebooks
    - run each cell and interpret the output
    - follow the exercise instructions
    - extra: modify a cell, add in code...

Reflection

- Automation as a Strategy
- Programming as Automated Insight
    - Excel vs. Notebooks
    - Programs vs. Notebooks
- Individual Project: Forensic Python Case Study
    - Run the walkthrough code and interpret the output

Appendix

An example of a python algorithm,

```python
def algorithm(input_data):
    output = 0.2 * input_data + 1
    return output

x_age = 31
y_profit = algorithm(x_age)

print(y_profit)
```

```
7.2
```

An example of using the numpy library,

```python
import numpy as np # load numpy

x_age = np.random.normal(30, 5, 10_000) # simulate 10k ages

y_profit = algorithm(x_age) # compute 10k predictions

print(y_profit.mean()) # show mean prediction

# for each of the first five xs and ys
for x, y in zip(x_age[:5], y_profit[:5]):
```