PYTHON DATA ANALYTICS

FOR MERE MORTALS

OBJECTIVES

- Learn the difference between data "*science*" and "*analysis*", and understand the characteristics (and toolsets) of their practitioners.
- Learn how to use Python (with a few 3rd party modules) to modernize your data transformation, analysis, and visualization tasks.
- Learn how to integrate these new tools and techniques into your everyday applications by live-coding some analytical mico services

AGENDA

1. Intro, definitions, and level setting [10 min]
2. Stack overview and environment setup [10 min]
3. Whirlwind tour of :
    a. data exploration and ETL [30 min]
    b. data visualization [10 min]
    c. machine learning [10 min]
4. Live coding capstone [40 min]
5. Q&A

WHO THIS IS FOR

PRIMARY AUDIENCE

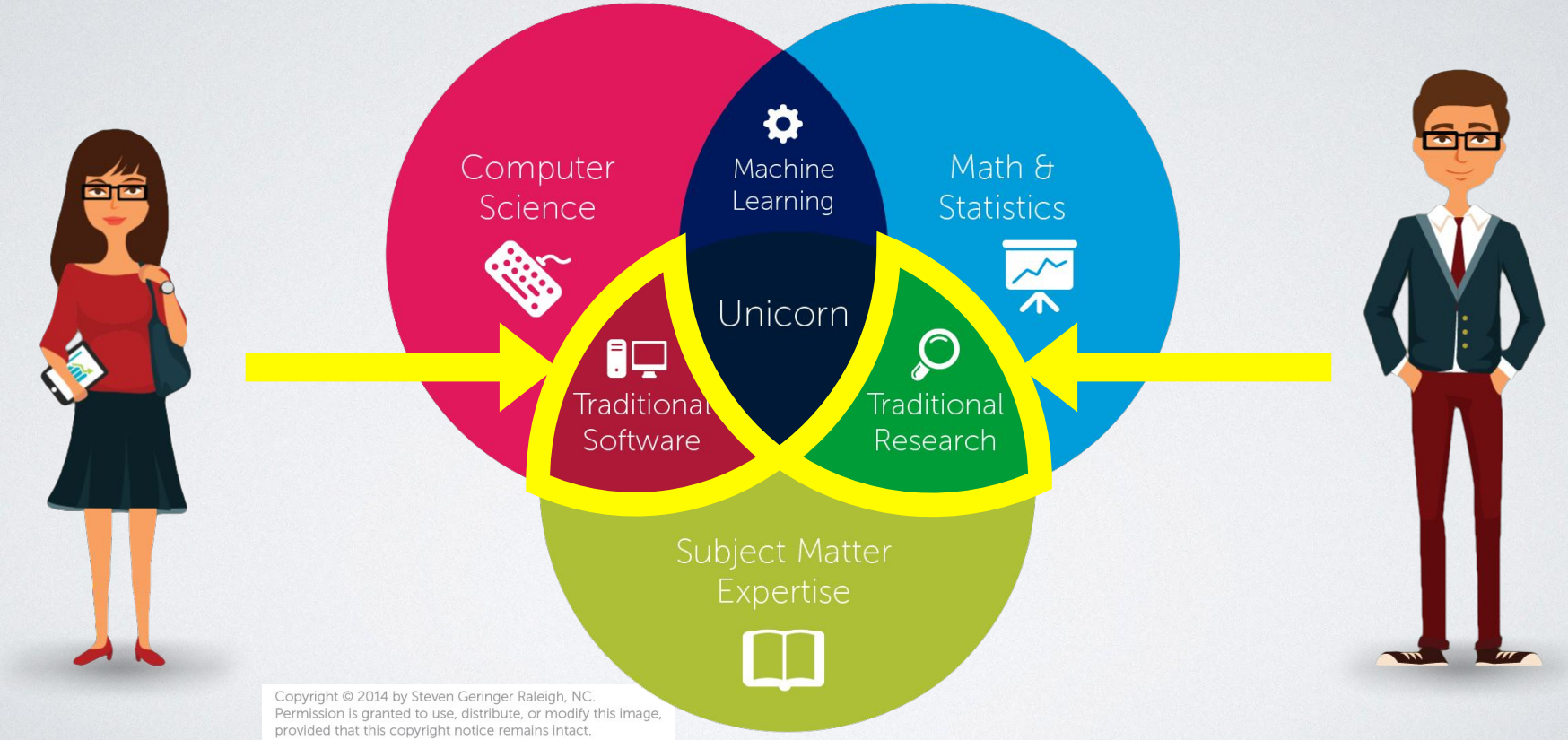
- Application developers
- Data/Business analysts *

ANYONE INTERESTED IN

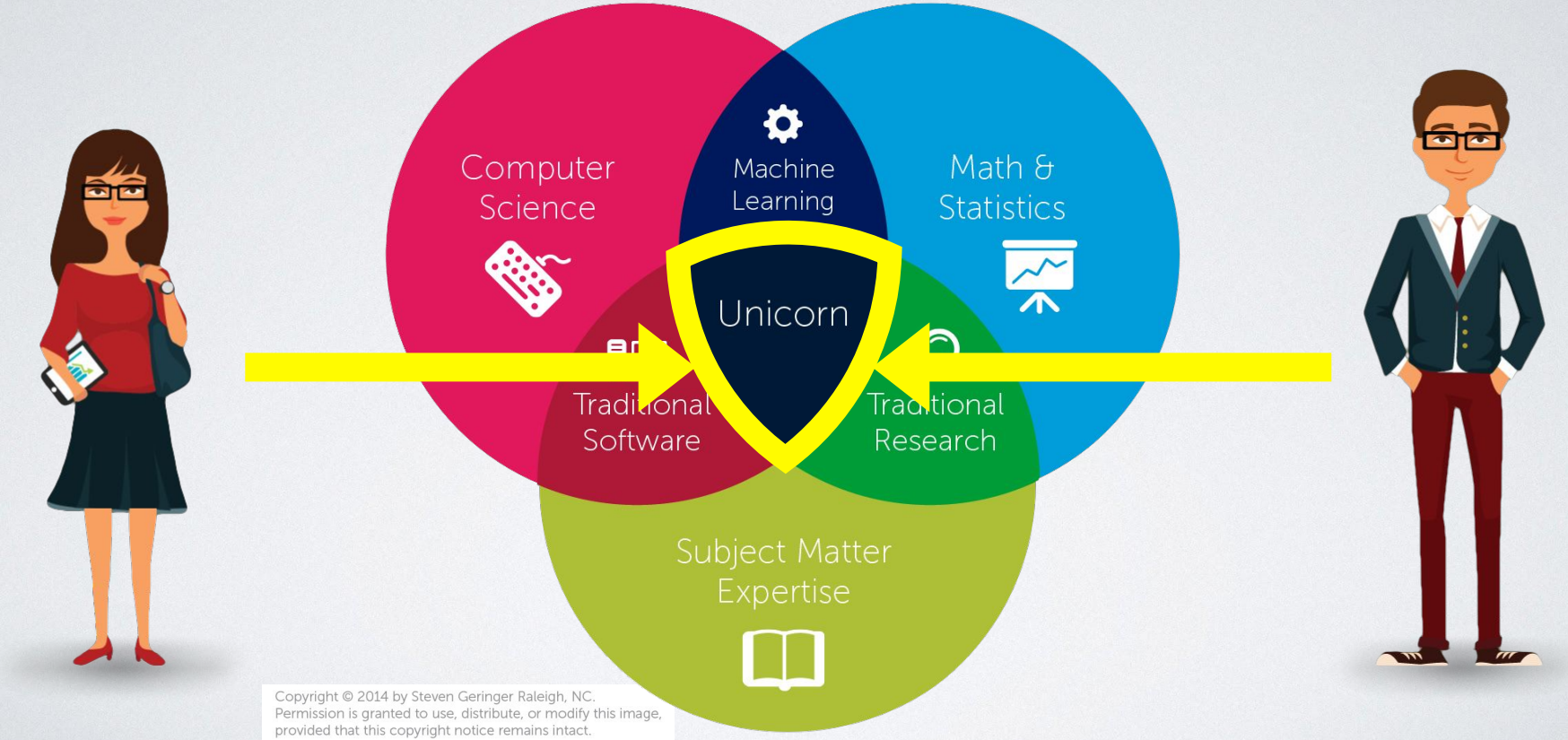
- Simple data analysis techniques/tools
- Learning basic predictive modeling
- Data visualization and reporting

* some coding experience required

FROM HERE…



… TO HERE



ENVIRONMENT SETUP

WHY PYTHON?

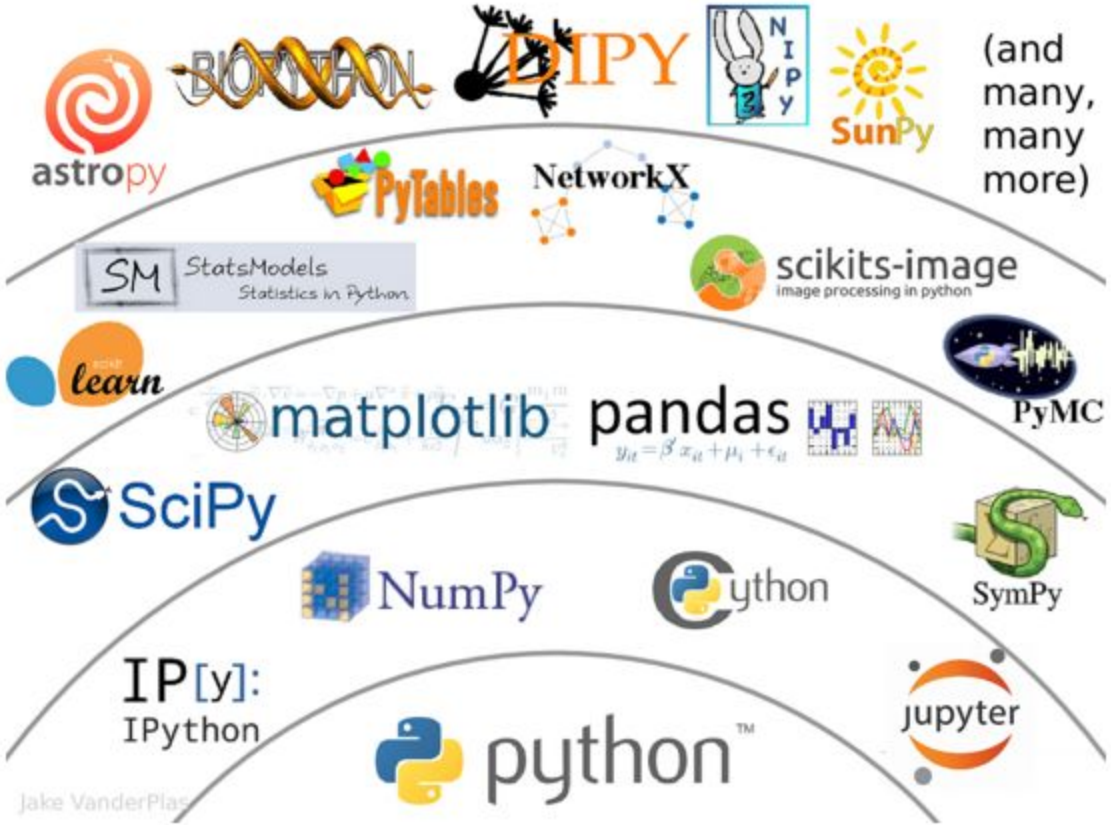
- This is one arena that isn't dominated by Java; this is the domain of R and Python... more-so R
- Then why Python and not R? In short - you can't build real apps in R… it's not general purpose enough and the learning curve is steep
- Powerful and easy to learn - 8 of the top 10 ComSci departments in the nation use Python in introductory programming courses *

INSTALL PYTHON

For our demo today I will use the Anaconda distribution, which includes the following on top of the reference distro you get @ python.org :

- Huge set of precompiled, tested, and certified modules for Windows, Mac, and Linux - production safe and no need to compile yourself
- Built-in environment virtualization and module management utilities (conda, pip)
- Wide variety of data science and analytics packages

The PyData Stack



REQUIRED PACKAGES

- **NumPy** - the fundamental package for scientific computing with Python
- **Pandas** - library providing high-performance, easy-to-use data structures and data analysis tools for Python
- **Scikit-learn** - probably the most useful library for machine learning in Python.
- **ipython/jupyter** - a web application that allows you to interactively create and share documents that contain live code, equations, visualizations and explanatory text
- **StatsModels** - allows for the estimation of many different statistical models, statistical tests, and statistical data exploration.
- **Seaborn** - a powerful statistical visualization library based on matplotlib
- **Bokeh** - interactive visualization library for Python that targets modern web browsers for presentation
- **FuzzyWuzzy** - an advanced string matching module for Python; crazy useful
- **Flask** - one of the best darn micro-frameworks for web development in python

INSTALL PACKAGES - TLDR;

```
> wget https://raw.githubusercontent.com/jpwhite3/python-analytics-demo/master/pydata.yml

> conda update conda

> conda env create -f pydata.yml

[pydata] > activate pydata

[pydata] > ipython notebook
```

ETL DEMO

SCENARIO

We need to perform some basic ETL against a set of sales data with the following requirements:

1. The data comes to us in several parts that we need to combine prior to analysis, and the format of each part is a little different.

2. Next, we need to add a calculated column to the end of the combined data set. While we are at it, clean up any data quality issues.

3. We then need to join the results to a different set of data that contains a common key - similar to a table join in SQL.

4. Finally, we must join to yet another source of data based on rough string matching and quantify our match strength - a vlookup on steroids.

OUTPUT

Once we have completed all the tasks on the prior slide we will need to produce various forms of output :

1. A simple spreadsheet with all rows and columns from our result set

2. Another spreadsheet pre-pivoted by a set of columns, including simple aggregates

3. A third spreadsheet that has a different cross-section of the pivot above on each tab

READY.
SET.
GO!

VISUALIZATION

From Simple, to Scientific, to Sexy

Charting Libraries

- Basic charting and plotting with MatPlotLib
- Interactive charting for the web with Bokeh
- Statistical charting with Seaborn

Machine Learning

What the what?!?

Machine Learning?!?

It's not as scary as it seems. Broadly, there are 3 types of machine learning algorithms… today we will only focus on the easiest one.

Supervised Learning:

- These types of algorithms consist of a dependent variable which is to be predicted from a given set of predictors (independent variables). Using these variables, we generate a mathematical function that maps inputs to desired outputs. This "training process" continues until the model achieves a desired level of accuracy.

- Examples of supervised learning include Linear & Logistic Regression, Decision Tree, Random Forest, KNN etc.

Linear Regression?!?

Linear regression is used to estimate real values (cost of houses, number of calls, total sales etc.) based on continuous variable(s). Here, we establish relationship between independent and dependent variables by fitting a “best line”.

This “best fit line” is known as the regression line and represented by a linear equation Y= a *X + b.

Logistic Regression?!?

Poorly named, this is a classification not a regression algorithm. It is used to estimate discrete values (0/1, yes/no, true/false) based on given set of independent variable(s).

Simply put, it predicts the probability of occurrence of an event by fitting data to a logit function. Since, it predicts the probability, its output values lies between 0 and 1.

Interpretation

Determine how likely (or unlikely) it is for random fluctuations to produce a test statistic as large as the one you actually got from your data (the "p value").

- If the p value is less than 0.05, then you conclude that the effect you observed was statistically significant.

Measure how close the data is to the fitted regression line. It is also known as the coefficient of determination, or R-squared. R-squared is always between 0 and 100%

- 0% indicates that the model explains none of the variability
- 100% indicates that the model explains all the variability

INTEGRATION

Building something useful

MATERIALS

All materials available on GitHub

https://github.com/jpwhite3/python-analytics-demo

REFERENCES

- http://www.forbes.com/sites/piyankajain/2013/02/25/data-science-or-analytics/
- http://www.edureka.co/blog/core-data-scientist-skills/
- http://pbpython.com
- http://www.marketingdistillery.com/
- http://pandas.pydata.org
- http://bokeh.pydata.org/en/latest
- https://www.continuum.io/why-anaconda
- https://s3.amazonaws.com/quandl-static-content/Documents/Quandl+-+Pandas,+SciPy,+NumPy+Cheat+Sheet.pdf
- http://web.stanford.edu/~mwaskom/software/seaborn/tutorial/regression.html

MORE REFERENCES