Models with R and Python

What is

What is

What is

Language Platform

Community

Ecosystem

What is

What is R

Language Platform

- A statistics programming language
- A data visualization tool
- Open source

Community

- 2.5+M users
- Taught in most universities
- New and recent grad's use it
- Thriving user groups worldwide

Ecosystem

- 12,000+ free packages in CRAN
- Scalable to big data
- Rich application & platform integration

A brief history of R

- 1993: Research Project in Auckland, NZ
- 1995: Released as Open-Source Software
- 1997: R Core Group Formed
- 2000: R 1.0.0 Released
- 2003: R Foundation Formed in Austria
- 2004: R 2.0.0 Released, First International User Conference in Vienna
- 2007: Revolution Analytics founded
- 2009: New York Times article on R
- 2013: Revolution R Open released
- 2015: R Consortium founded
- 2018 R 3.5 Released


Photo credit: Robert Gentleman

```
hello <- function(name) {
    # Prints name
    paste("Hello,", name)
}

hello("LATAM AI+ Tour!")
```

What is

Language Platform

- A general purpose scripting language
- With statistics and visualization packages
- Open source
- Has become a very popular language for deep learning

Community

- Millions of users
- Taught in most universities
- New and recent grad's use it
- Thriving user groups worldwide

Ecosystem

- 150,000+ free packages in PyPI
- Scalable to big data
- Rich application & platform integration

Python is…

… was created as a Christmas ‘hobby’ in 1989 by Guido Van Rossum.

… that is ‘duck-typed’

… with a lightweight syntax

… that emphasizes readability, simplicity, and elegance

… meant to be ‘beautiful’

```python
def hello(name):
    # Prints name
    print("Hello, " + name)

if __name__ == "__main__":
    hello("LATAM AI+ Tour!")
```

Guido

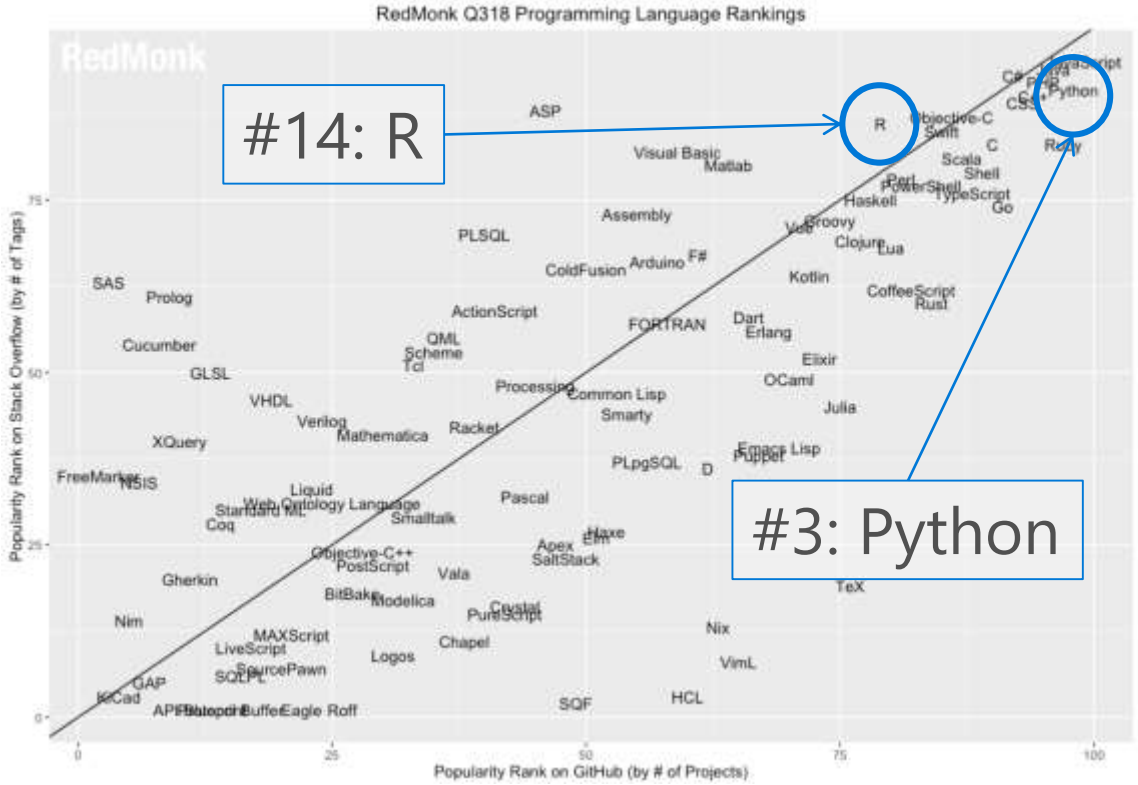
R and Python are the most popular analytic languages

IEEE Spectrum Top Programming Languages

Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C++		99.7
3. Java		97.5
4. C		96.7
5. C#		89.4
6. PHP		84.9
7. R		82.9
8. JavaScript		82.6
9. Go		76.4
10. Assembly		74.1

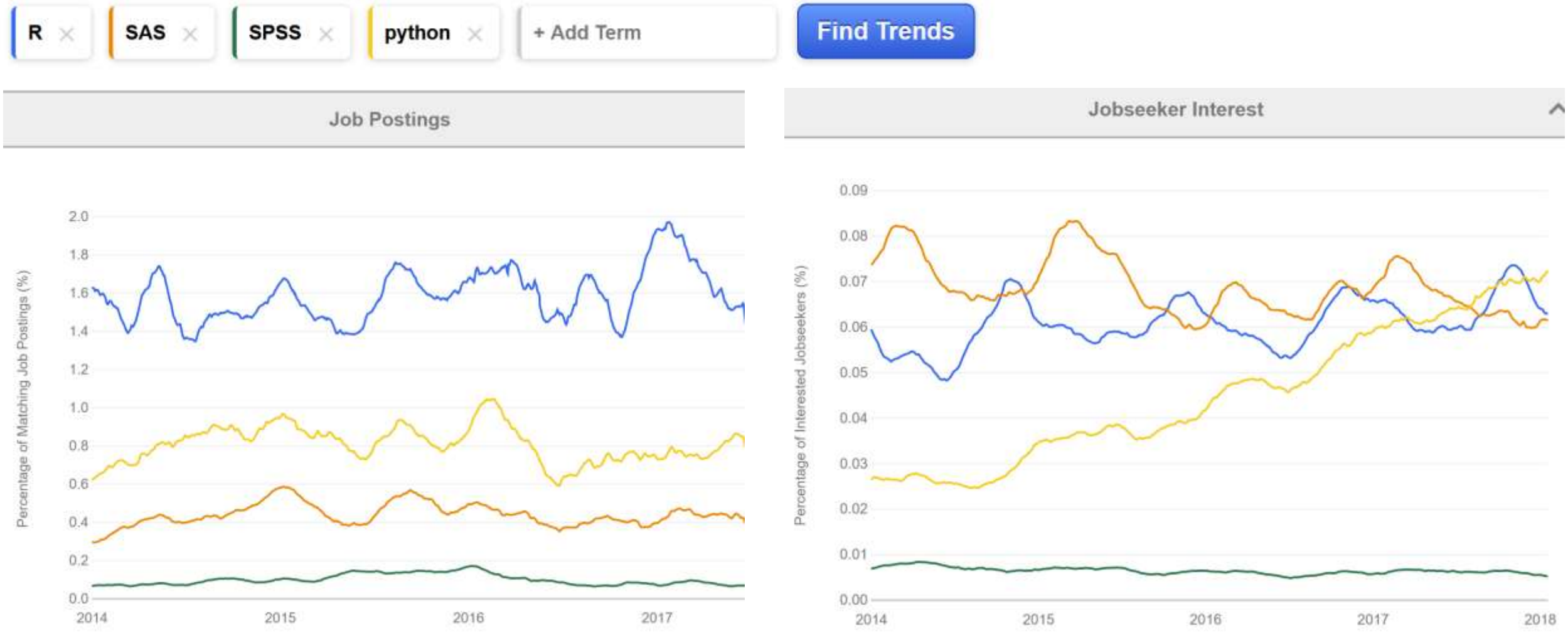
[IEEE Spectrum, July 2018]

Redmonk Language Rankings



[Redmonk, June 2018]

But what does the market say?



Source: https://www.indeed.com/jobtrends/q-R-q-SAS-q-SPSS-q-python.html

R Extensibility – publically available libraries

CRAN Task Views

CRAN Task Views are guides to the packages and functions useful for certain disciplines and methodologies. Many long-term R users I know have no idea they exist. As an effort to make them more widely known I thought I'd jazz up the index page. Images are free to use, and got from SXC stock photo site. Visual puns are mine. Task View links go to the cran.r-project.org site and not a mirror.

Bayesian Inference

Applied researchers interested in Bayesian statistics are increasingly attracted to R because of the ease of which one can code algorithms to sample...[more]

Natural Language Processing

This CRAN task view contains a list of packages useful for natural language processing...[more]

Analysis of Spatial Data

Base R includes many functions that can be used for reading, vizualising, and analysing spatial data. The focus in this view is on "geographical" spatial...[more]

Chemometrics and Computational Physics

Chemometrics and computational physics are concerned with the analysis of data arising in chemistry and physics experiments, as well as the simulation of...[more]

Analysis of Pharmacokinetic Data

The primary goal of pharmacokinetic (PK) data analysis is to determine the relationship between the dosing regimen and the body's exposure to the drug as...[more]

Clinical Trial Design, Monitoring, and Analysis

This task view gathers information on specific R packages for design, monitoring and analysis of data from clinical trials. It focuses on including...[more]

Official Statistics & Survey Methodology

This CRAN task view contains a list of packages that includes methods typically used in official statistics and survey methodology. Many packages provide...[more]

Survival Analysis

Survival analysis, also called event history analysis in social science, or reliability analysis in engineering, deals with time until occurrence of an...[more]

Cluster Analysis & Finite Mixture Models

This CRAN Task View contains a list of packages that can be used for finding groups in data and modelling unobserved cross-sectional heterogeneity. Many...[more]

Phylogenetics, Especially Comparative Methods

The history of life unfolds within a phylogenetic context. Comparative phylogenetic methods are statistical approaches for analyzing historical...[more]

Time Series Analysis

Base R ships with a lot of functionality useful for time series, in particular in the stats package. This is complemented by many packages on CRAN, which are...[more]

Probability Distributions

For most of the classical distributions, base R provides probability distribution functions (p), density functions (d), quantile functions (q), and...[more]

Multivariate Statistics

Base R contains most of the functionality for classical multivariate analysis, somewhere. There are a large number of packages on CRAN which extend this...[more]

Robust Statistical Methods

Robust (or "resistant") methods for statistics modelling have been available in S from the start, in R in package stats (e.g., median(), mean(*, trim = .),...[more]

Computational Econometrics

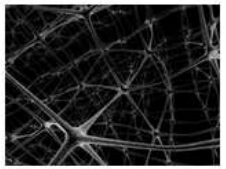
Base R ships with a lot of functionality useful for computational econometrics, in particular in the stats package. This functionality is complemented by many...[more]

Optimization and Mathematical Programming

This CRAN task view contains a list of packages which offer facilities for solving optimization problems. Although every regression model in statistics...[more]

Analysis of Ecological and Environmental Data

This Task View contains information about using R to analyse ecological and environmental data...[more]



Machine Learning & Statistical Learning

Several add-on packages implement ideas and methods developed at the borderline between computer science and statistics - this field of research is usually...[more]

Statistics for the Social Sciences

Social scientists use a wide range of statistical methods. To make the burden carried by this task view lighter, I have suppressed detail in some areas that...[more]

Design of Experiments (DoE) & Analysis of Experimental Data

This task view collects information on R packages for experimental design and analysis of data from experiments. Please feel free to suggest enhancements,...[more]

Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization

R is rich with facilities for creating and developing interesting graphics. Base R contains functionality for many plot types including coplots, mosaic...[more]

gRaphical Models in R

Wikipedia defines a graphical model as a graph that represents independencies among random variables by a graph in which each node is a random variable, and...[more]

Empirical Finance

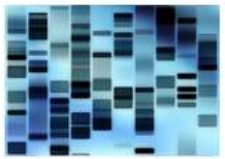
This CRAN Task View contains a list of packages useful for empirical work in Finance, grouped by topic....[more]

High-Performance and Parallel Computing with R

This CRAN task view contains a list of packages, grouped by topic, that are useful for high-performance computing (HPC) with R. In this context, we are...[more]

Reproducible Research

The goal of reproducible research is to tie specific instructions to data analysis and experimental data so that scholarship can be recreated, better...[more]



Statistical Genetics

Great advances have been made in the field of genetic analysis over the last years. The availability of millions of single nucleotide polymorphisms (SNPs)...[more]

Medical Image Analysis

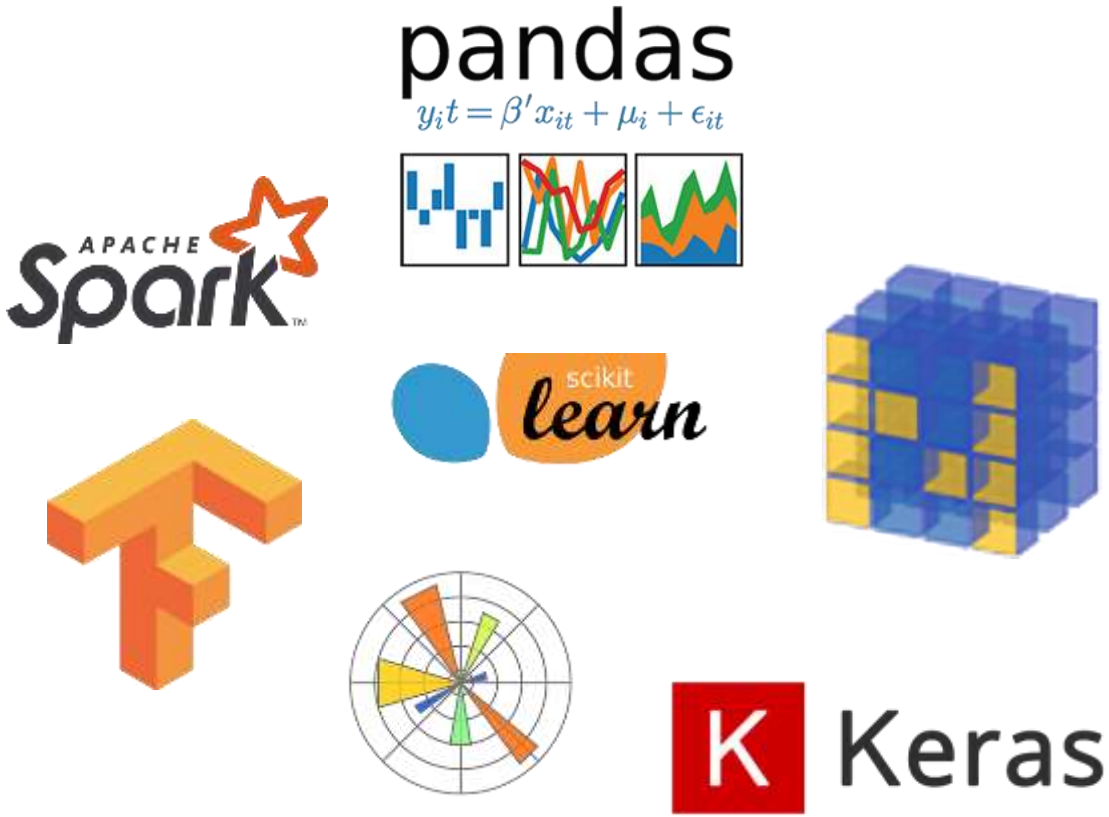
This task view is for input, output, and analysis of medical imaging files....[more]

Psychometric Models and Methods

Psychometrics is concerned with the design and analysis of research and the measurement of human characteristics. Psychometricians have also worked...[more]

http://www.maths.lancs.ac.uk/~rowlings/R/TaskViews/

Python Extensibility – publically available libraries



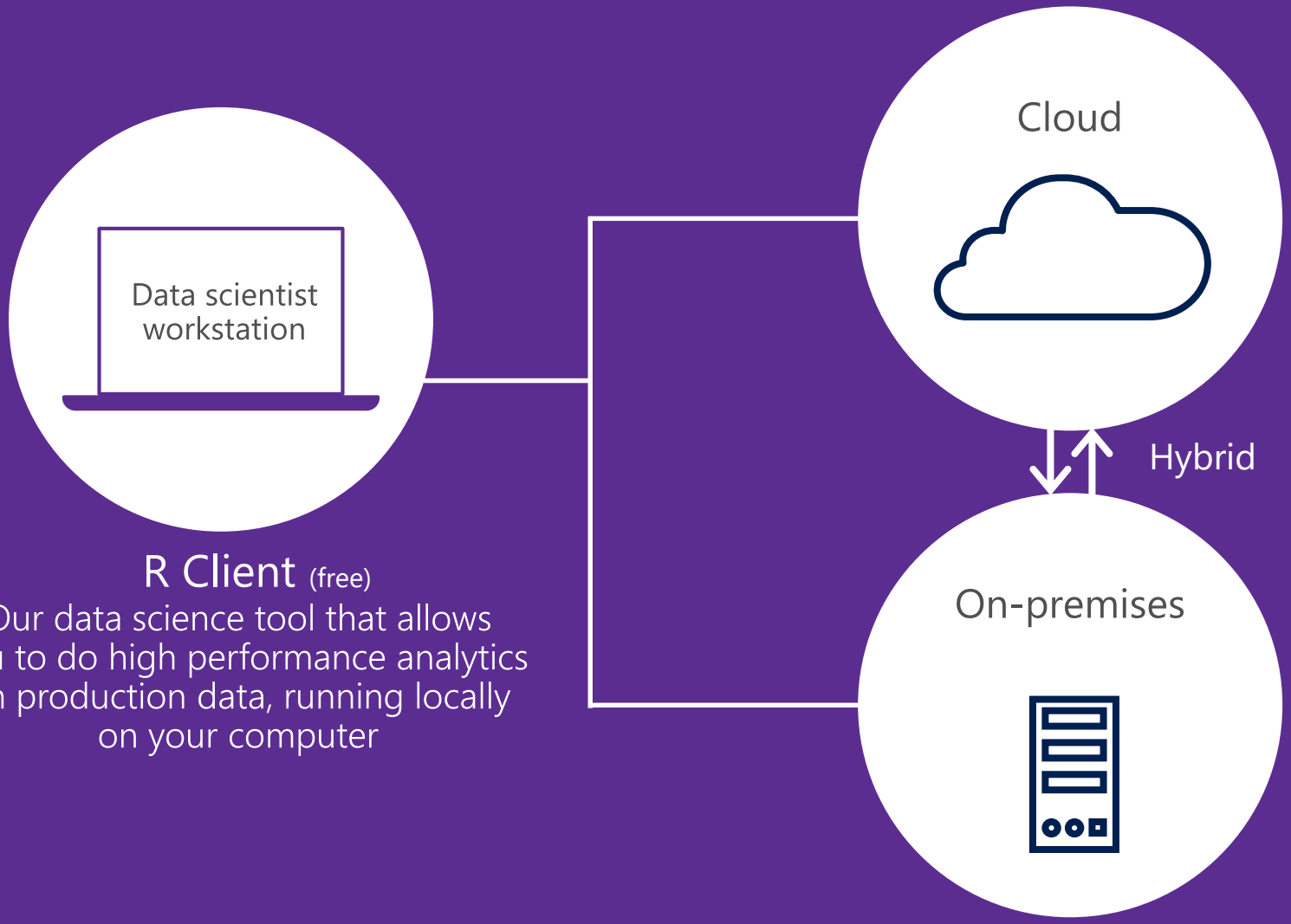
150,000+ packages on https://pypi.org/

So what language should I choose?

HINT: It depends, but likely both

Microsoft Data and AI Tools that Support Python and/or R

Microsoft ML Server



R Client (free)
Our data science tool that allows you to do high performance analytics on production data, running locally on your computer

Big Data
R Server for HDInsight

Linux and Windows Servers
Virtual Machines

Big Data
R Server for Hadoop/Spark

In-Database
SQL Server 2016 (R Services)

SQL Server 2017 (Windows, R/Python)

Linux and Windows Servers
ML Server for Linux and Windows

Azure ML Services

Automated machine learning and hyper-parameter tuning

Identify the best algorithms faster with automated machine learning, and find the best model efficiently with intelligent hyper-parameter tuning.

Version control and reproducibility

Increase your rate of experimentation by tracking and logging your experiments for reproducibility and easy modification.

Support for open source libraries and IDEs

Use machine learning libraries such as Tensorflow, PyTorch, and scikit-learn. Azure Machine Learning service integrates with your favorite Python IDE, including Visual Studio Code, Visual Studio, Azure Databricks notebooks, or Jupyter notebooks.

Model management

Proactively manage and monitor your models using the image and model registry, and upgrade them through integrated CI/CD.

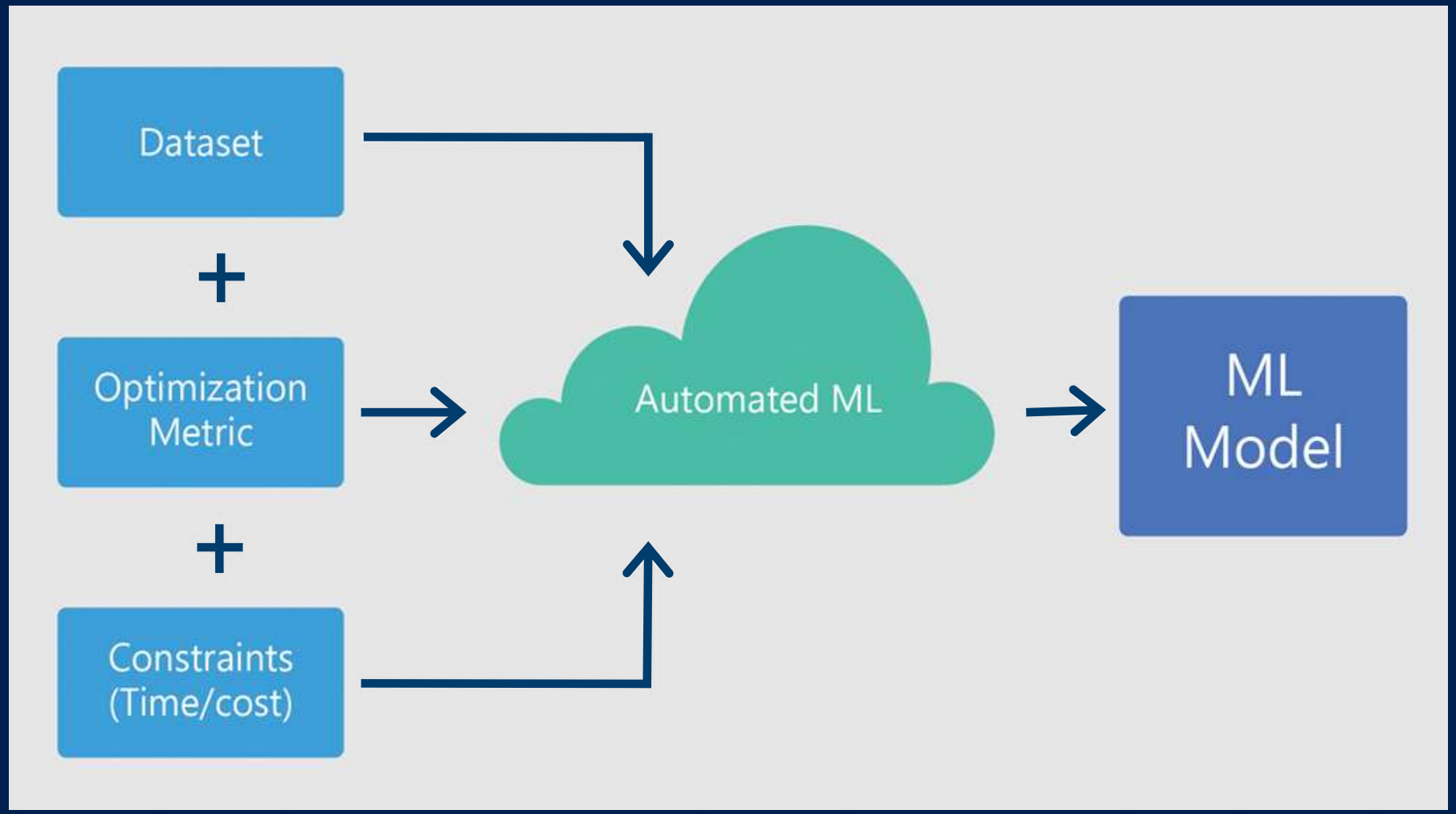
Hybrid deployment

Deploy models where you need them most with managed deployments to the cloud and the edge.

Distributed deep learning

Build better models faster with massive, managed GPU clusters. Train models quickly with distributed deep learning, and deploy them on FPGAs.

Azure ML Services – Automated ML



Azure Databricks

Microsoft Azure | PORTAL | erikzwi@microsoft.com

taxi-demo-file-loading (Python)

Attached: training | File | View: Code | Permissions | Stop Execution | Clear | Schedule | Comments | Revision history

Azure Databricks | Home | Workspace | Recent | Data | Clusters | Jobs | Search

NYC Taxi Commission Dataset

A demo of Databricks functionality using the publically available dataset from the NYC Taxi Commission

```
# Mount blob ADLS in DBFS applicationId = '974d7063-4897-453f-967d-c022c6c4fa52' ...
```

```
# Create Schema Object  from pyspark.sql.types import * greenSchema = StructType ...
```

```
greenDF: pyspark.sql.dataframe.DataFrame = [vendor_id: integer, pickup_datetime: timestamp ... 18 more fields]
```

```
filteredGreenDF = (greenDF.filter("pickup_longitude<0")
                   .filter("pickup_longitude>-78")
                   .filter("pickup_latitude>0")
                   .sample(False, 0.3, 11234)
                  )

latitudes = filteredGreenDF.rdd.map(lambda x: x['pickup_latitude']).collect()
longitudes = filteredGreenDF.rdd.map(lambda x: x['pickup_longitude']).collect()
```

(2) Spark Jobs Cancel

Job 40 View (1 stages)

Stage 41: 22/22 (0 running)

Job 41 View (1 stages)

```
filteredGreenDF: pyspark.sql.dataframe.DataFrame = [vendor_id: integer, pickup_datetime: timestamp ... 18 more fields]
```

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()
ax.scatter(longitude, latitude)
```

HDInsight / Spark

jupyter Untitled (autosaved)

File Edit View Insert Cell Kernel Widgets Help PySpark

Code CellToolbar

```
In [1]: from pyspark.sql.types import *
```

```
Creating SparkContext as 'sc'
```

ID	YARN Application ID	Kind	State	Spark UI	Dri log
0	application_1483718293579_0006	pyspark	idle	Link	Lin

```
Creating HiveContext as 'sqlContext'
SparkContext and HiveContext created. Executing user code
```

```
In [ ]:
```

File Edit Code View Plots Session Build Debug Tools Help

testhdi_spark.r

Source on Save Run Source

```
#copy local file to HDFS
rxHadoopMakeDir("/share")
rxHadoopCopyFromLocal("/usr/lib64/MRS-*/library/RevoScaleR/Samp

myNameNode <- "default"
myPort <- 0

# Location of the data
bigDataDirRoot <- "/share"

# define HDFS file system
hdfsFS <- RxHdfsFileSystem(hostName=myNameNode, port=myPort)
```

69:62 (Top Level) R Script

Console

```
Coefficients: (1 not defined because of singularities)
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         -3.19458    0.20413 -15.650 2.22e-16 ***
CRSDepTime           0.97862    0.01126  86.948 2.22e-16 ***
DayOfWeek=Monday     2.08100    0.18602  11.187 2.22e-16 ***
DayOfWeek=Tuesday    1.34015    0.19881   6.741 1.58e-11 ***
DayOfWeek=Wednesday  0.15155    0.19679   0.770    0.441
DayOfWeek=Thursday  -1.32301    0.19518  -6.778 1.22e-11 ***
DayOfWeek=Friday     4.80042    0.19452  24.679 2.22e-16 ***
DayOfWeek=Saturday   2.18965    0.19229  11.387 2.22e-16 ***
DayOfWeek=Sunday     Dropped    Dropped Dropped  Dropped
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 40.39 on 582620 degrees of freedom
Multiple R-squared: 0.01465
Adjusted R-squared: 0.01464
```

Environment History

Import Dataset List

Global Environment

model	List of 26
model1	List of 26
model2	List of 26
myHadoopMRCluster	Formal class RxHadoopMR
myNameNode	"default"
myPort	0
mySparkCluster	Formal class RxSpark
rxgLastPendingJob	Formal class RxDistributedHadoopMRJob

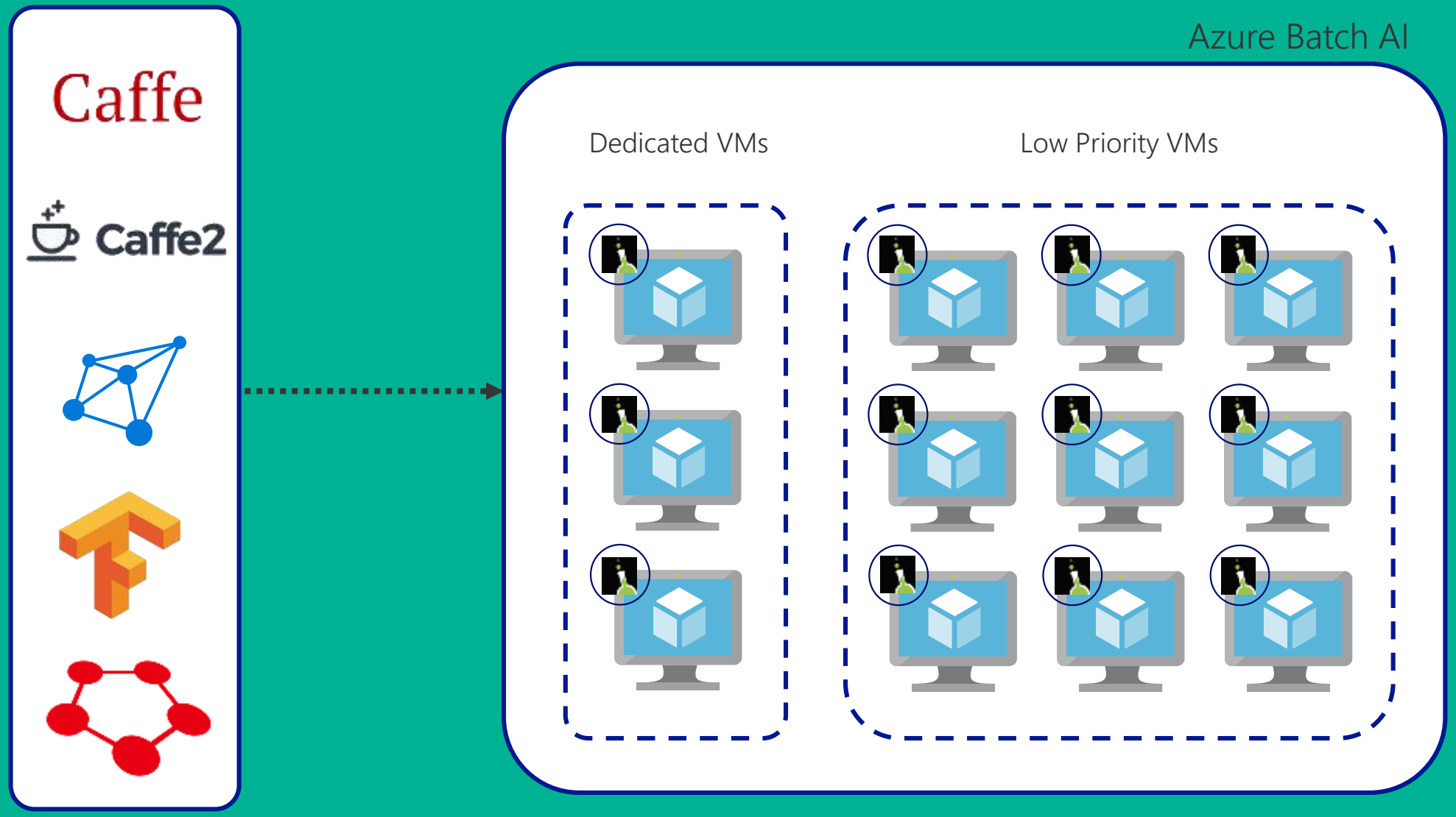
Files Plots Packages Help Viewer

New Folder Upload Delete Rename More

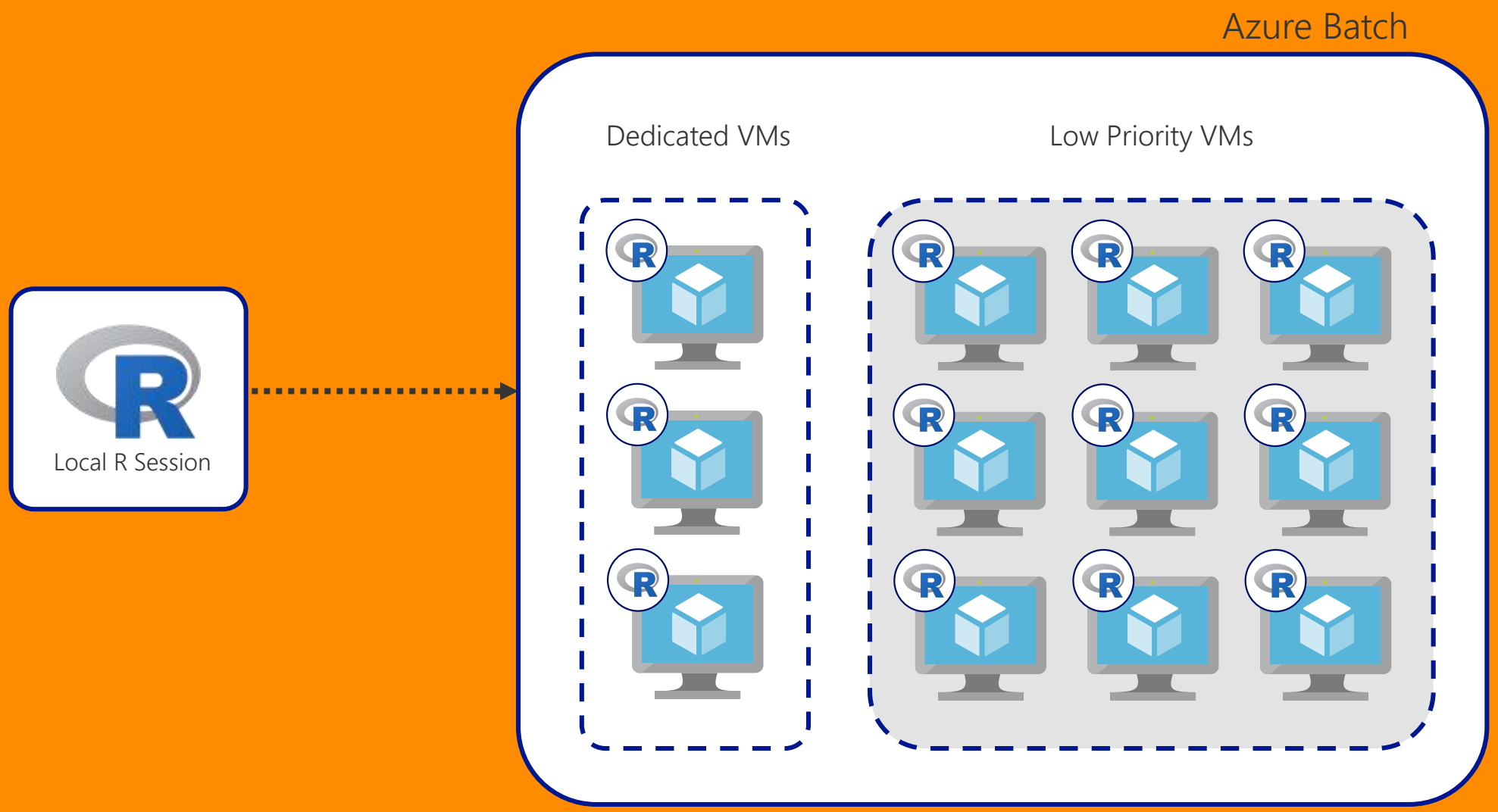
Home

Name	Size	Modified
R		
testhdi_spark.r	1.9 KB	Mar 24, 2016, 8:48 PM

Azure Batch AI



doAzureParallel()



And a bunch more…

Azure Notebooks	R	Python
Power BI	R	
mmlspark		Python
AzureML		Python
Brainwave		Python
DSVM	R	Python

So what language should I choose?

Decision Criteria

1. Polyglot company?
2. What are you wanting to do?

R Strengths	Python Strengths
• Widely taught in statistical programs • Beautiful visualizations • Best IDE options • Fervent community • Research often lands here first	• Strong deployment options • De facto language of deep learning • Broadest Spark support • Largest developer community

The good news is that with either choice, math is still math

Let's see some code,
but first how do we get these setup?

Install - R

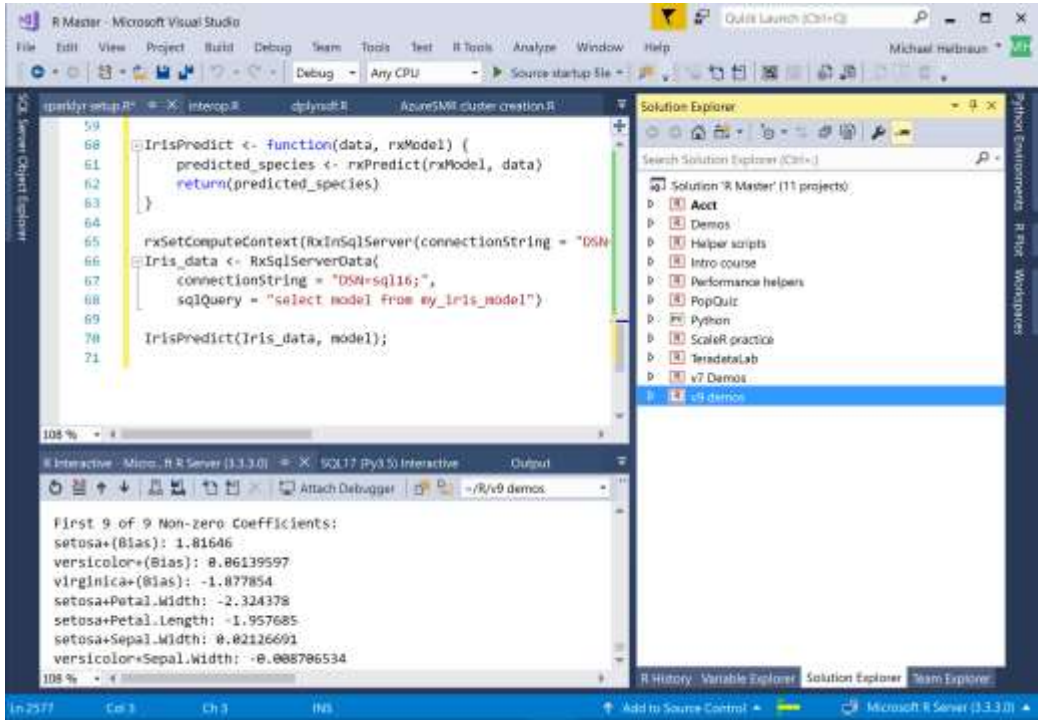
The first decision is the R version

- CRAN R
- Microsoft R Open
- Microsoft Machine Learning Server

Install – R IDEs

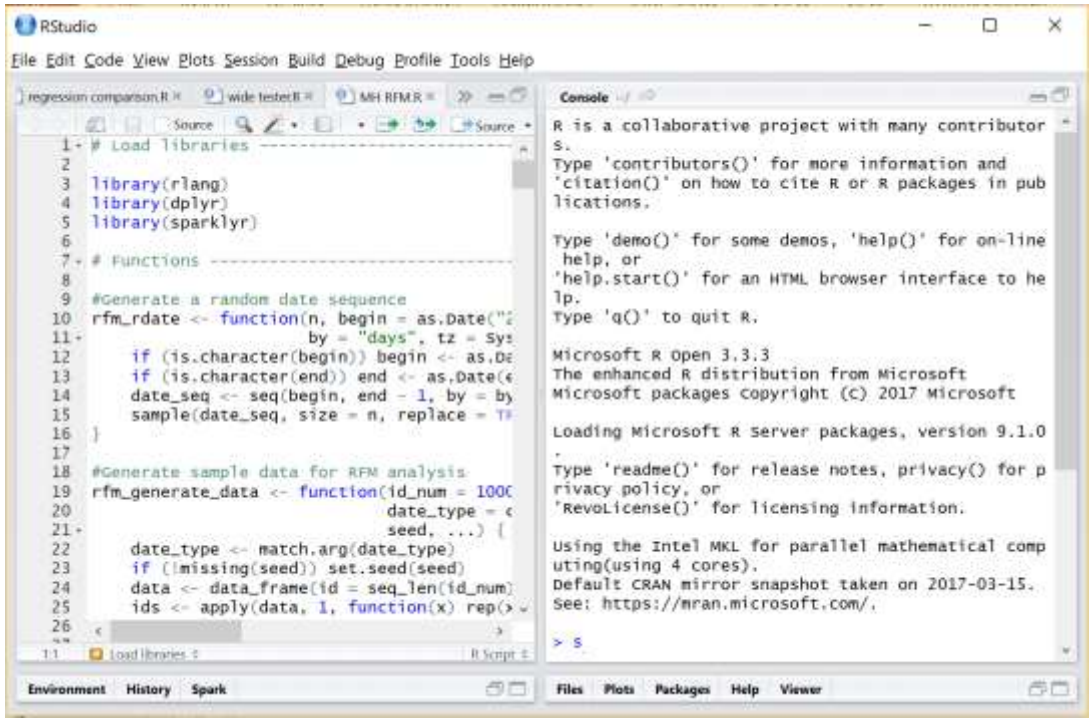
RTVS

- Supports multiple languages
- 1st party



RStudio

- Gold standard
- Mature and lightweight
- Thin client for Linux



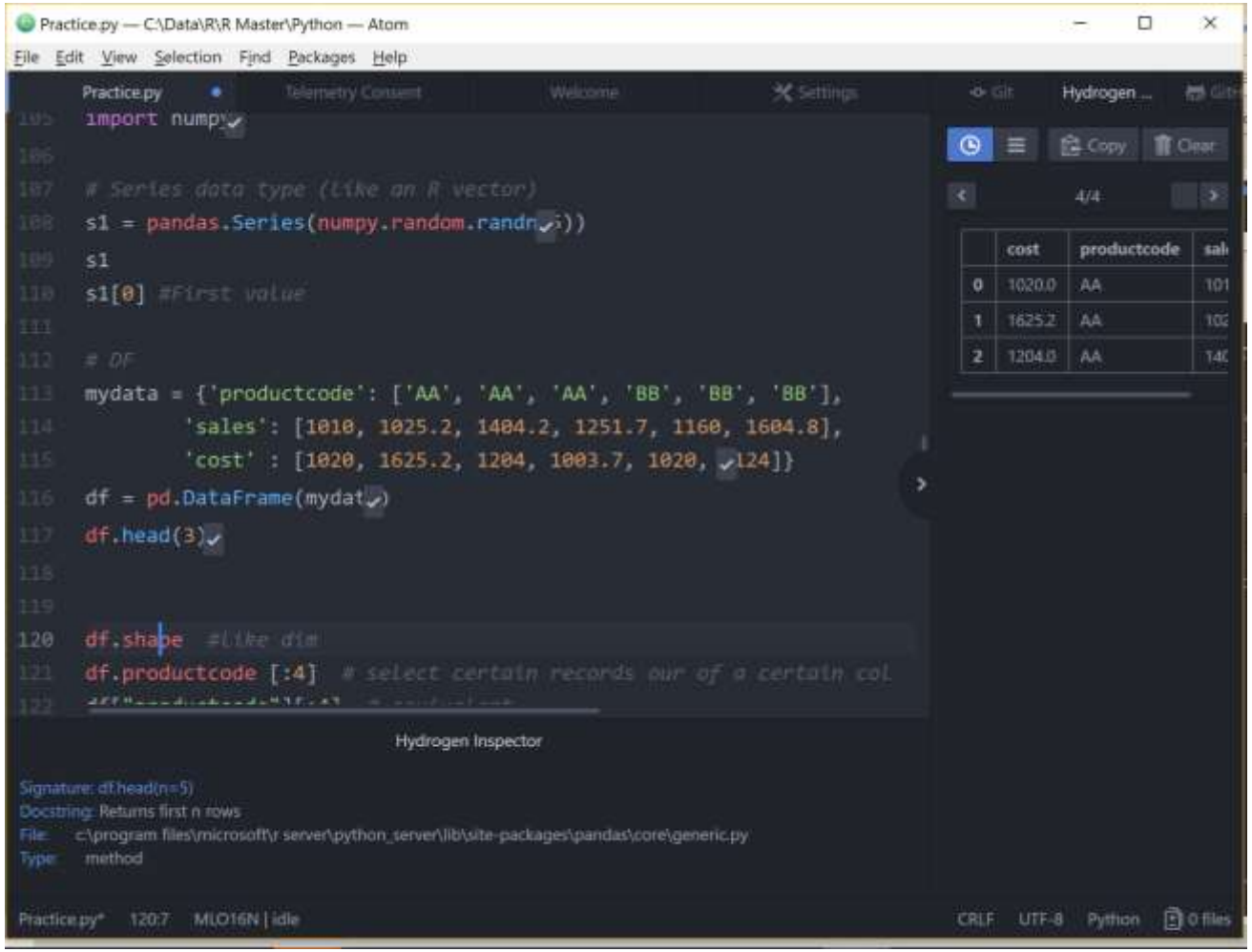
Install - Python

- The first decision is Python 2 or 3
- Install from:
  - python.org
  - Anaconda

Install – Python Dev Environments

Many Python Dev's prefer a notebook environment, but there isn't the same standardization as with R. Options include:

- VS, VS Code, PyCharm, Spyder, Rodeo, Hydrogen, Jupyter



DEMO

R Learning Resources

- Books
  - Norman Matloff, The Art of R programming
  - Andrie DeVries, R for Dummies
  - Jared Lander, R for Everyone
- MLS Documentation
- Blogs
  - http://blog.revolutionanalytics.com
  - http://trevorstephens.com/kaggle-titanic-tutorial/getting-started-with-r/
- RUGs and Meetups
  - http://blog.revolutionanalytics.com/local-r-groups.html
  - http://meetup.com
- Online training:
  - MLS Video Series – This is a 4 part series that was just put together to provide an intro to MRS specifically.
  - EdX course - Analyzing Big Data with Microsoft R Server; free 4 week course with 4hrs effort per week.
  - Data Camp course – Free course from Data Camp on MRS.

Python Learning Resources

- Books
  - Wes McKinney, Python for Data Analysis
- MLS Documentation
- Blogs
  - https://www.anaconda.com/blog/developer/new-advances-conda-0/
- RUGs and Meetups
  - https://wiki.python.org/moin/LocalUserGroups
  - http://meetup.com
- Online training:
  - EdX: Introduction to Python: Absolute Beginner, Introduction to Python: Fundamentals, Introduction to Python for Data Science, Programming with Python for Data Science
  - PluralSight – free trial and paid course
  - DataCamp – free and paid courses