ActiveState®

Adding a Programming Language

ActiveState Webinar

Panelists

- **Francois Ouellet**, Director of Development Practice, *Manulife*
- **George Williams**, Director of Data Science and Chief Evangelist, *GSI Technology*

Housekeeping

- Webinar recording and slides will be available shortly
- Share questions with panelists using the Question panel
- Q&A session following presentations

Adding a Language

Capital One | Bank of America | American National | HSBC | Tesco Bank | Credit Suisse

Bombardier | GE Aviation | Northrop Grumman | NASA | Siemens | Cisco

Intel | ca technologies | IBM | Microsoft | Hewlett Packard Enterprise | verizon

Track-record: 97% of Fortune 1000, 20+ years open source

Polyglot: 5 languages - Python, Perl, Tcl, Go, Ruby

Runtime Focus: concept to development to production

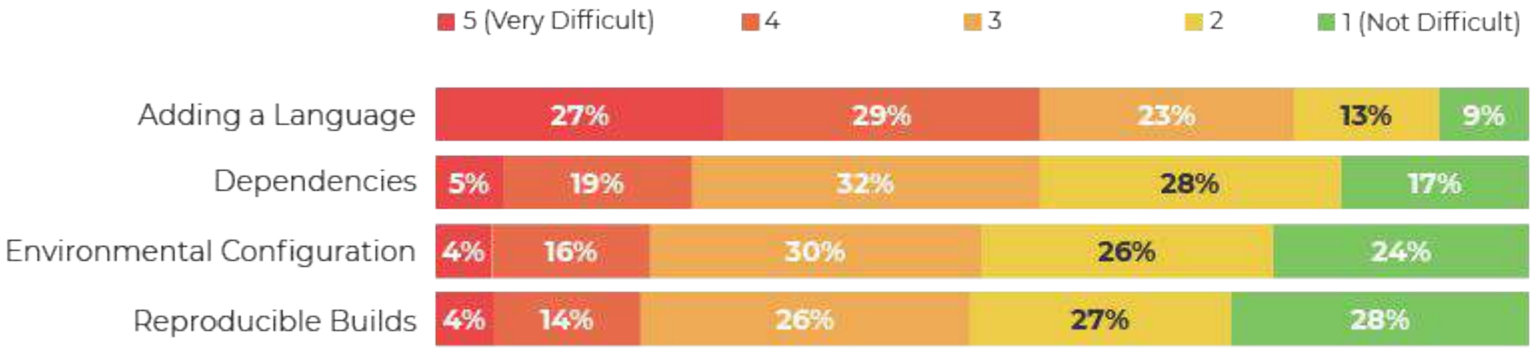
ActiveState®

Adding a Programming Language

Gains vs Pains

Adding a Language

Software Development Challenges



Source: ActiveState Developer Survey 2018, Open Source Runtime Pains

What’s so Difficult?

- **Education** - learn the new language & its tooling
- **Tooling** - extend or replace your toolchain
- **Workflow/Processes** - update your software development lifecycle

Education Resources

Learn at your own Pace:

- **Paid Classes**: lynda.com, Codecademy, Code School, Udemy, etc
- **Free Resources**: Code Camp, Edx, MIT Open Courseware, etc

Learn from Peers:

- Learn one; do one; teach one

Tooling

Gains:

- Polyglot IDEs
- Source code repositories like Git
- Binary repositories like Nexus
- Flexible code quality tools like SonarQube
- Popular automated testing tools like Selenium

Pains:

- Unit/ integration/ functional testing tools
- Language-specific build tools
- Polyglot IDEs vs dedicated IDEs

Workflow/ Processes

Considerations:

- **Builds** of Compiled vs Interpreted languages
  - e.g., Java + Maven vs Python + individual packages

- **Quality** of Statically- vs Dynamically-typed languages
  - e.g., C/C++ maturity vs JavaScript’s novelty (0 days since last new framework)

Language Distributions

Adopt a standard distribution:

- **Community** - free and ubiquitous (probably came with your OS)
  - Great way to get started learning the basics
- **Commercial** - vendor-supported; includes popular, third-party libraries
  - Best for exploring the language and its ecosystem
- **Do-It-Yourself** - don’t!
  - Too complex when you’re just starting out

Adding a Language

Introducing a New Programming Language

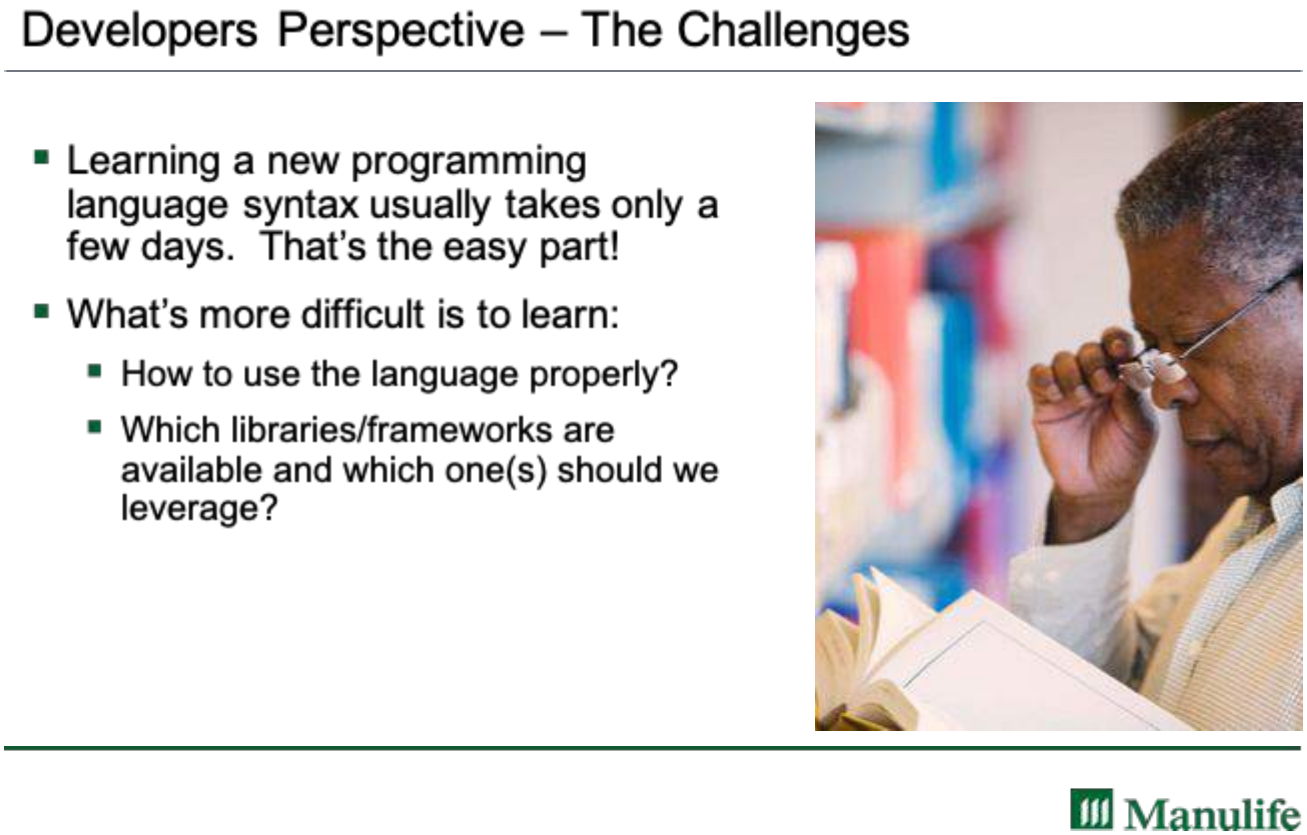
Challenges & Lessons Learned

Francois Ouellet

Director, Development
Practice, Canadian Division

Manulife

Manulife

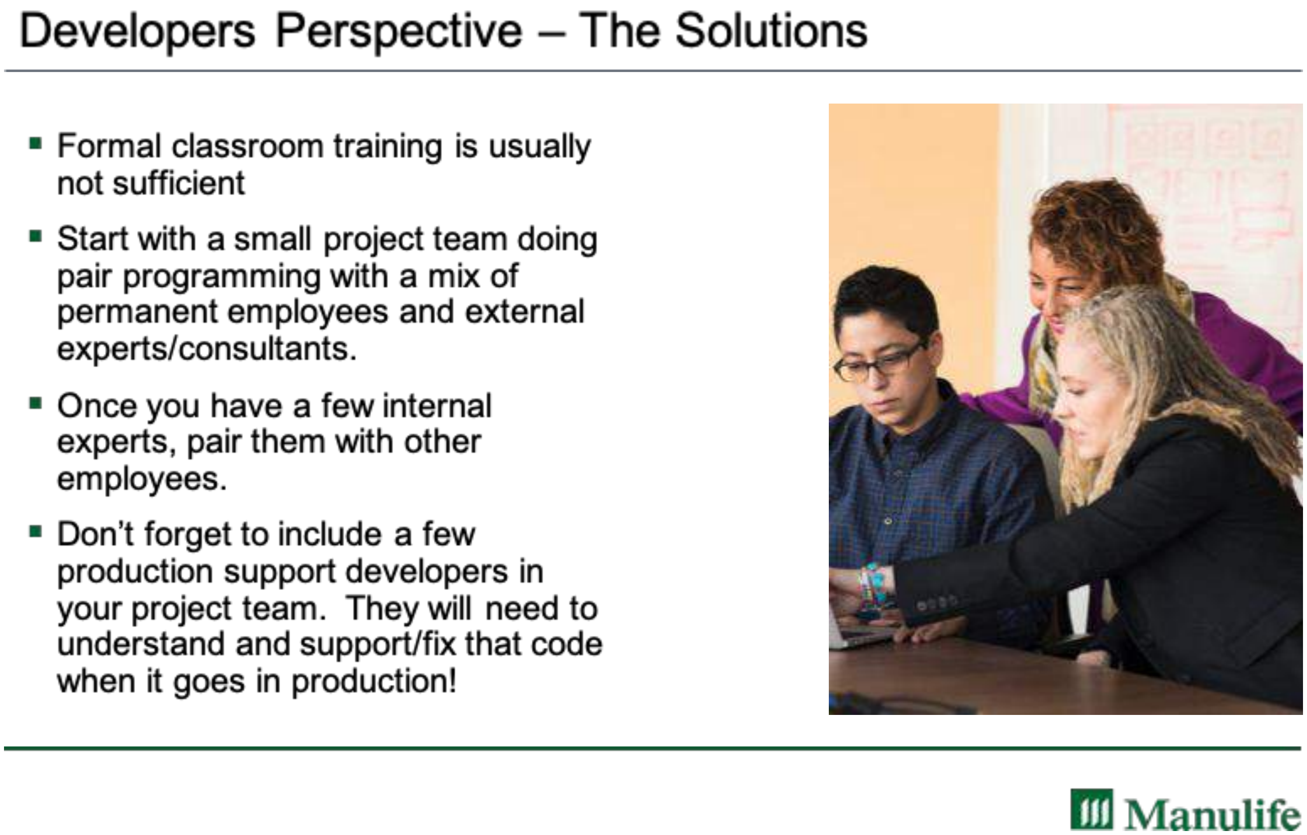


Developers Perspective – The Challenges

- Learning a new programming language syntax usually takes only a few days. That’s the easy part!
- What’s more difficult is to learn:
  - How to use the language properly?
  - Which libraries/frameworks are available and which one(s) should we leverage?

Manulife

Adding a Language



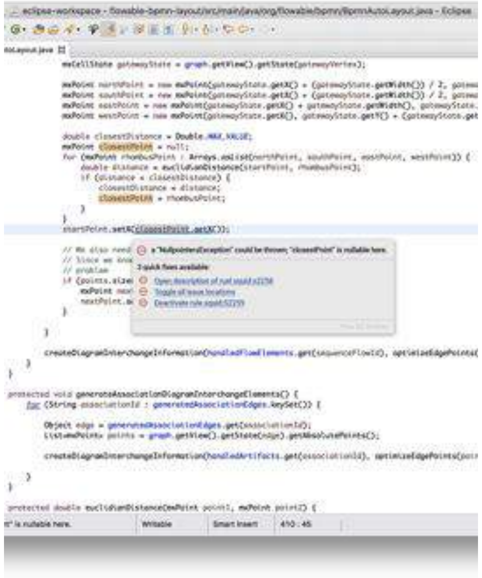
Developers Perspective – The Solutions

- Formal classroom training is usually not sufficient
- Start with a small project team doing pair programming with a mix of permanent employees and external experts/consultants.
- Once you have a few internal experts, pair them with other employees.
- Don't forget to include a few production support developers in your project team. They will need to understand and support/fix that code when it goes in production!

Manulife

Developers Perspective – The Solutions

- Make sure there's at least one good linter for the new programming language and use it:
  - Great tool to help avoiding some of the common bugs and pitfalls
  - It's a great time to enforce a coding standard and style
- It's even better if the linter is integrated in your developers IDE and perform on-the-fly code review
- You are new to the language but not to the business that you are building software for
  - Great opportunity to start building some shared libraries from day one



eclipse-workspace - flowable-bpmn-layout/src/main/java/org/flowable/bpmn/BpmnAutoLayout.java - Eclipse

```
mxCellState gatewayState = graph.getView().getState(gatewayVertex);

mxPoint northPoint = new mxPoint(gatewayState.getX() + (gatewayState.getWidth()) / 2, gatewa
mxPoint southPoint = new mxPoint(gatewayState.getX() + (gatewayState.getWidth()) / 2, gatewa
mxPoint eastPoint = new mxPoint(gatewayState.getX() + gatewayState.getWidth(), gatewayState.
mxPoint westPoint = new mxPoint(gatewayState.getX(), gatewayState.getY() + (gatewayState.get

double closestDistance = Double.MAX_VALUE;
mxPoint closestPoint = null;
for (mxPoint rhombusPoint : Arrays.asList(northPoint, southPoint, eastPoint, westPoint)) {
    double distance = euclidianDistance(startPoint, rhombusPoint);
    if (distance < closestDistance) {
        closestDistance = distance;
        closestPoint = rhombusPoint;
    }
}
startPoint.setX(closestPoint.getX());

// We also need
// Since we know
// problem
if (points.size
    mxPoint next
    nextPoint.se
}

}

createDiagramInterchangeInformation(handledFlowElements.get(sequenceFlowId), optimizeEdgePoints(
}
}

protected void generateAssociationDiagramInterchangeElements() {
    for (String associationId : generatedAssociationEdges.keySet()) {

        Object edge = generatedAssociationEdges.get(associationId);
        List<mxPoint> points = graph.getView().getState(edge).getAbsolutePoints();

        createDiagramInterchangeInformation(handledArtifacts.get(associationId), optimizeEdgePoints(poi

    }
}

protected double euclidianDistance(mxPoint point1, mxPoint point2) {
```

a "NullPointerException" could be thrown; "closestPoint" is nullable here.

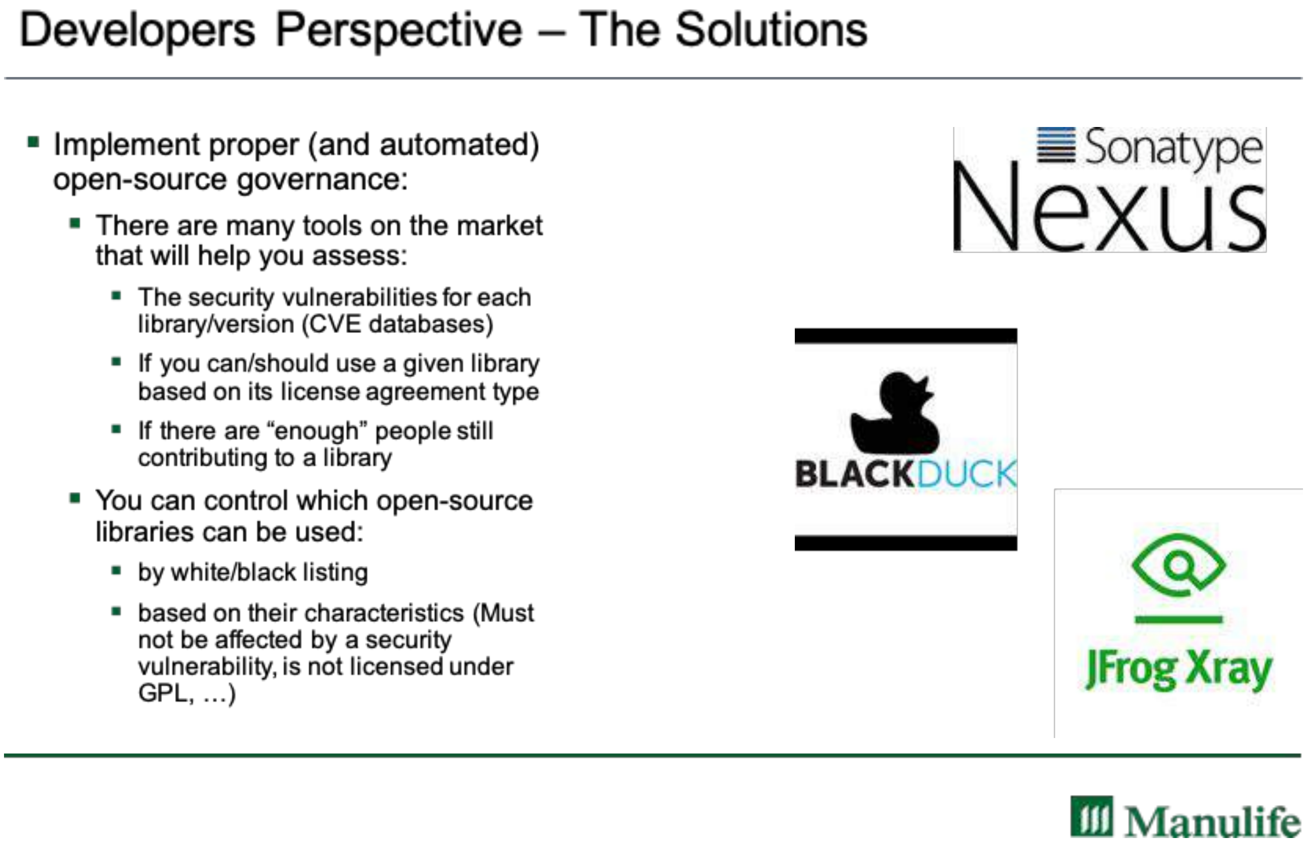
3 quick fixes available:

- Open description of rule squid:S2259
- Toggle all issue locations
- Deactivate rule squid:S2259

Developers Perspective – The Solutions

- Make sure there is a large and active community of people using that programming language in the industry:
  - Google is your developers' best friend when they are looking for information and answers
  - The more people use a language the more likely you are to find a lot of code examples or open-source libraries the will help accelerate the work of your project teams.

Adding a Language



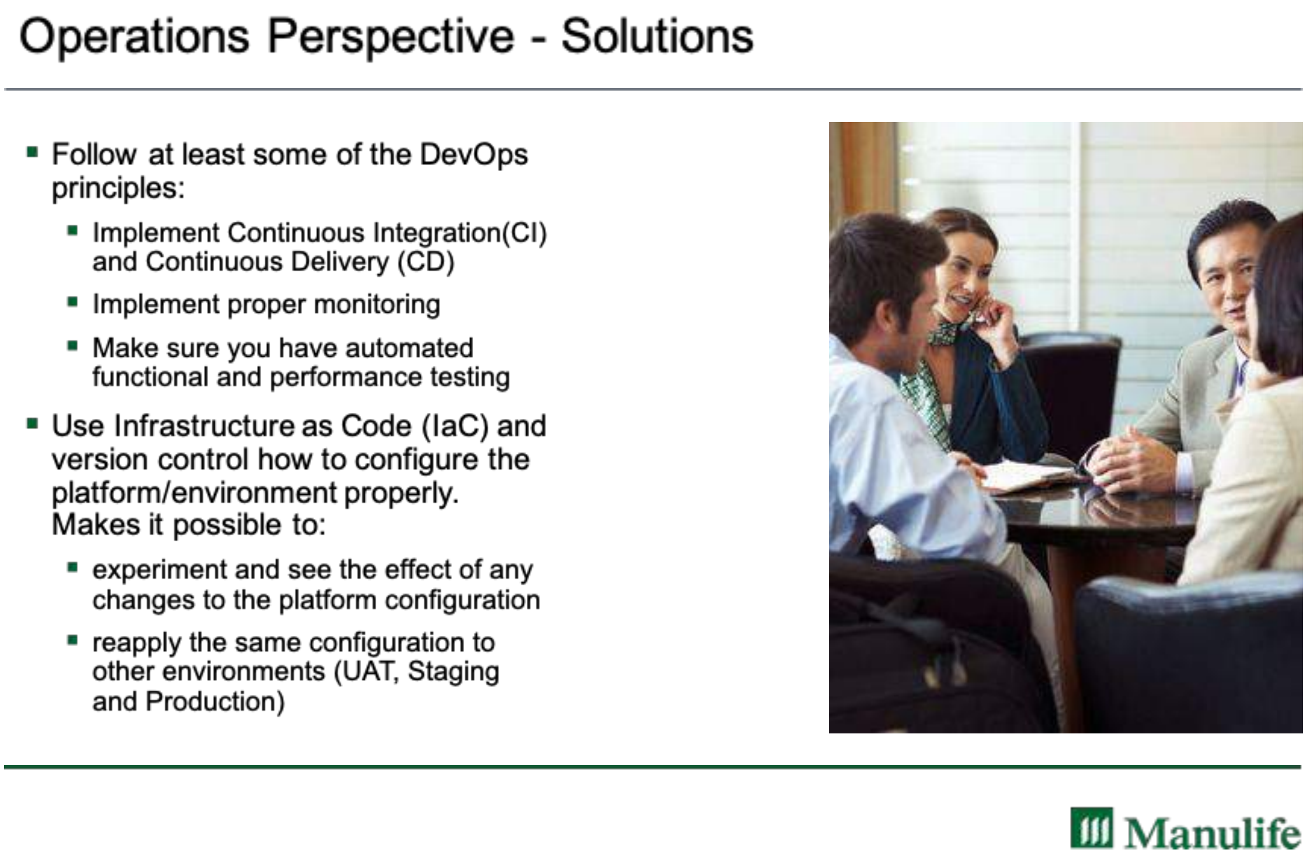
Developers Perspective – The Solutions

- Implement proper (and automated) open-source governance:
  - There are many tools on the market that will help you assess:
    - The security vulnerabilities for each library/version (CVE databases)
    - If you can/should use a given library based on its license agreement type
    - If there are “enough” people still contributing to a library
  - You can control which open-source libraries can be used:
    - by white/black listing
    - based on their characteristics (Must not be affected by a security vulnerability, is not licensed under GPL, …)

Operations Perspective - Challenges

- What do we need to introduce in our infrastructure to support that new programming language?
  - JVM
  - .Net Framework
  - V8 engine
  - …
- How do we configure that properly?
  - Memory
  - Disk
  - …
- How do we monitor an application written in that new programming language?

Adding a Language



Operations Perspective - Solutions

- Follow at least some of the DevOps principles:
  - Implement Continuous Integration(CI) and Continuous Delivery (CD)
  - Implement proper monitoring
  - Make sure you have automated functional and performance testing
- Use Infrastructure as Code (IaC) and version control how to configure the platform/environment properly. Makes it possible to:
  - experiment and see the effect of any changes to the platform configuration
  - reapply the same configuration to other environments (UAT, Staging and Production)

Adding a Language

Thank you

Adding A Language

George Williams

Who Am I?

Director, GSI Technology

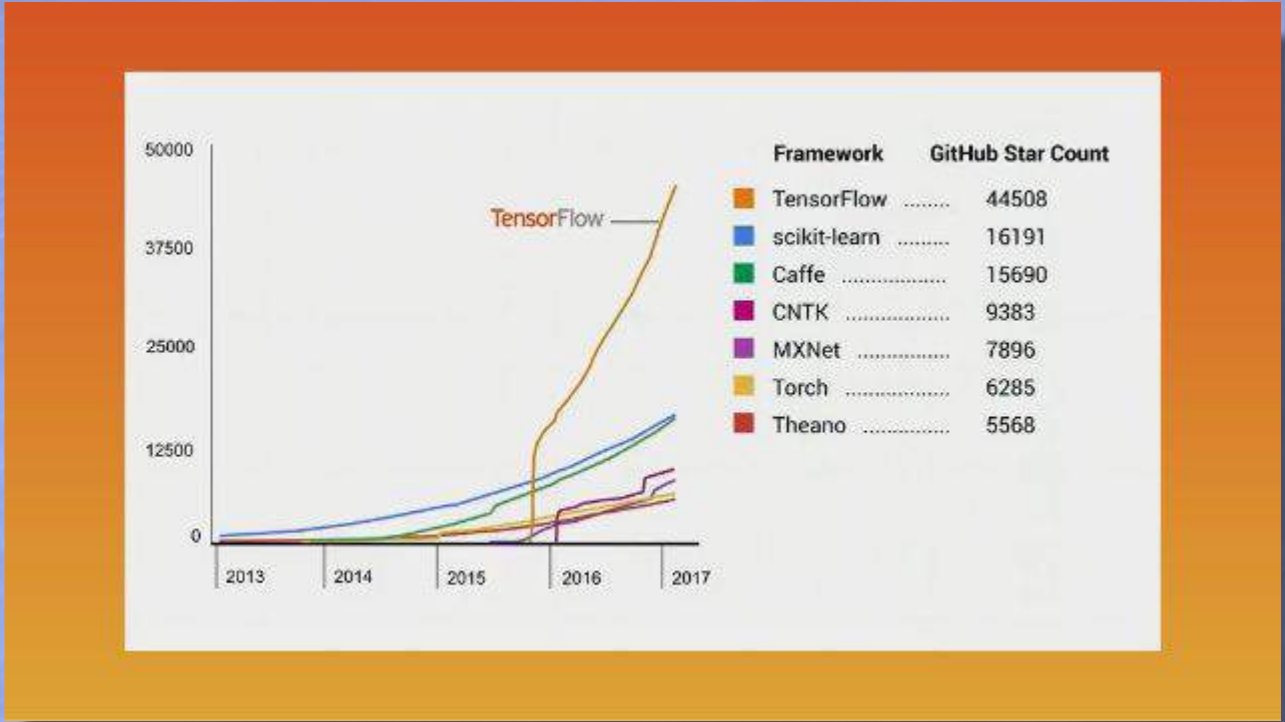
Previously, Chief Data Scientist

Senior Data Scientist

AI Research Scientist

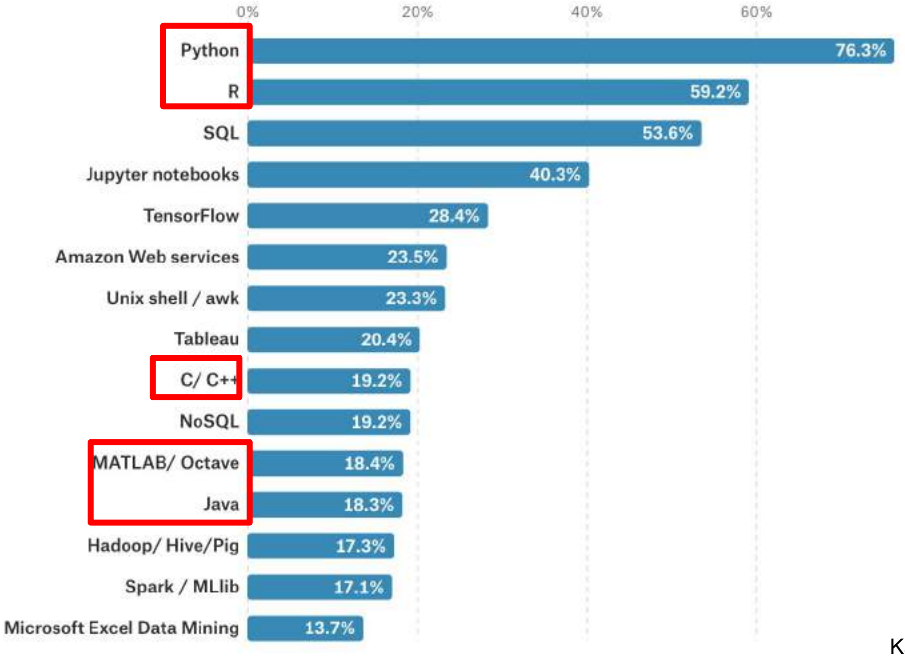
Software Engineer

“AI” Frameworks’ Explosion



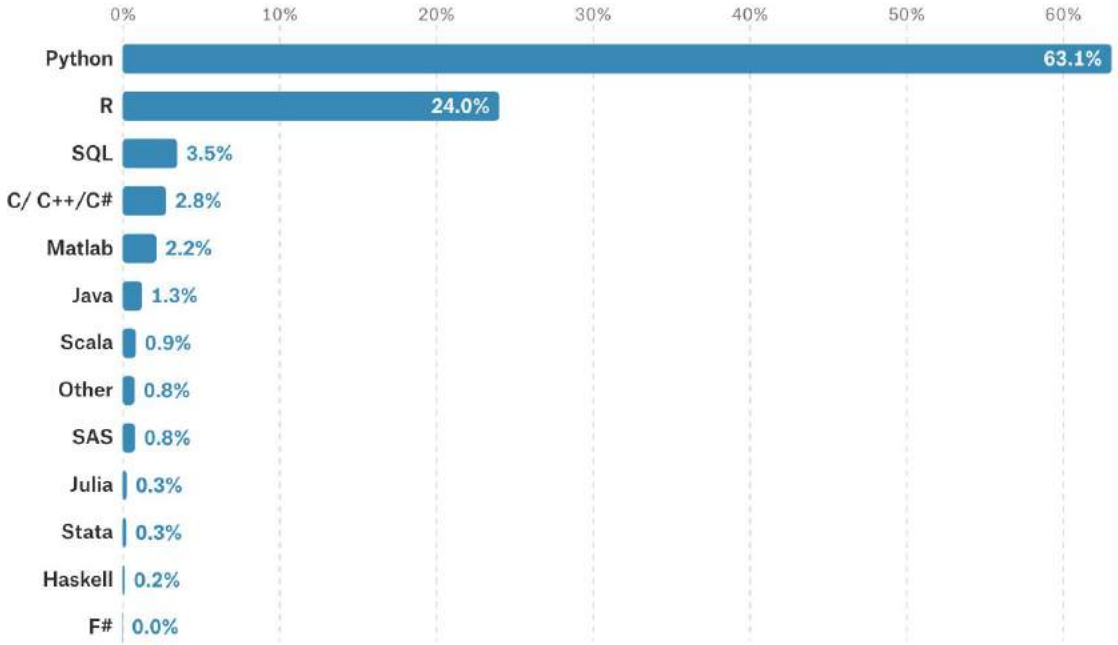
	Framework	GitHub Star Count
🟧	TensorFlow	44508
🟦	scikit-learn	16191
🟩	Caffe	15690
🟥	CNTK	9383
🟪	MXNet	7896
🟨	Torch	6285
🟥	Theano	5568

Data Science “Tools”

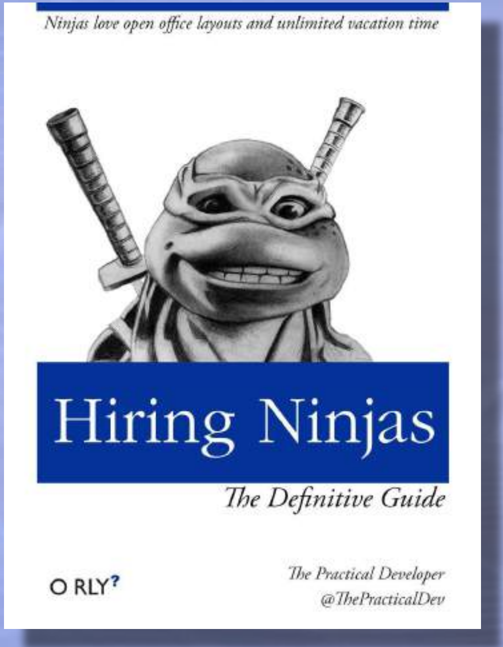


Kaggle, 2017

Recommended Languages



Hiring Data Science "Ninjas"



Ninjas love open office layouts and unlimited vacation time

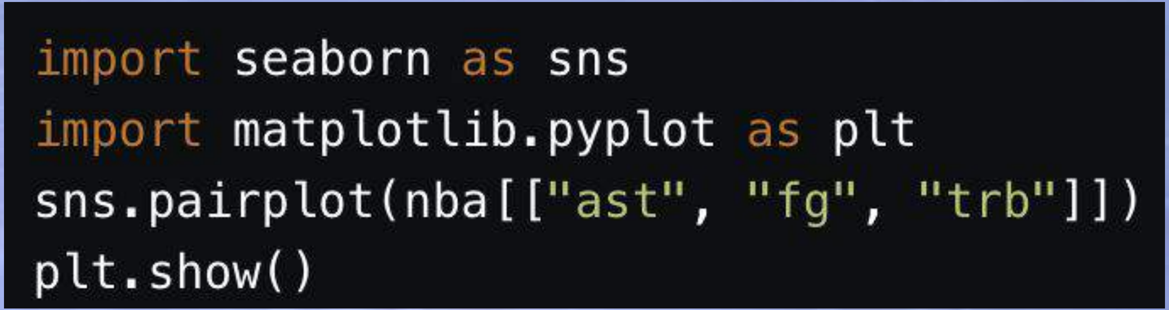
Hiring Ninjas

The Definitive Guide

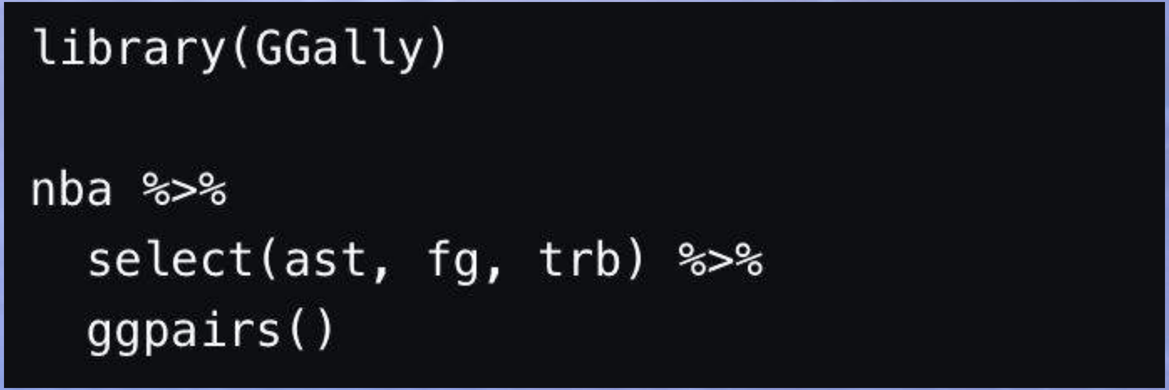
O RLY?

The Practical Developer
@ThePracticalDev

Statistical Analysis

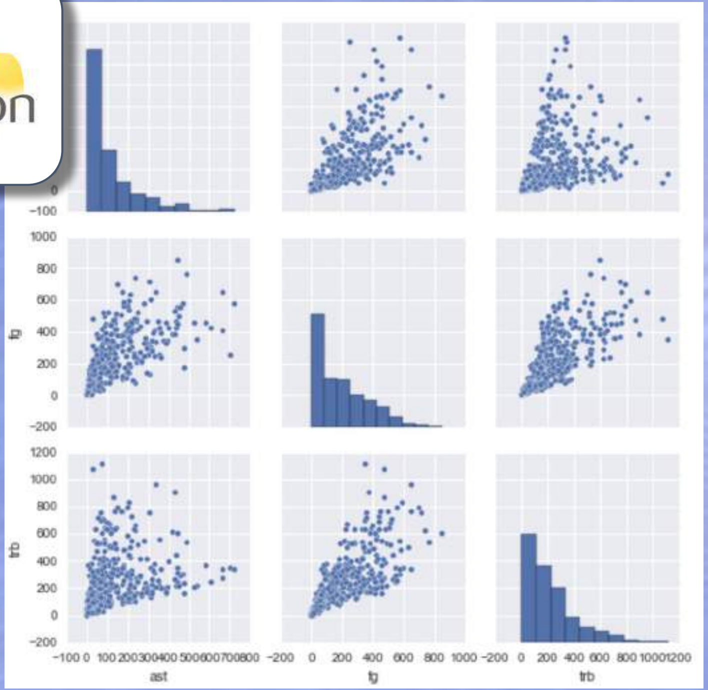


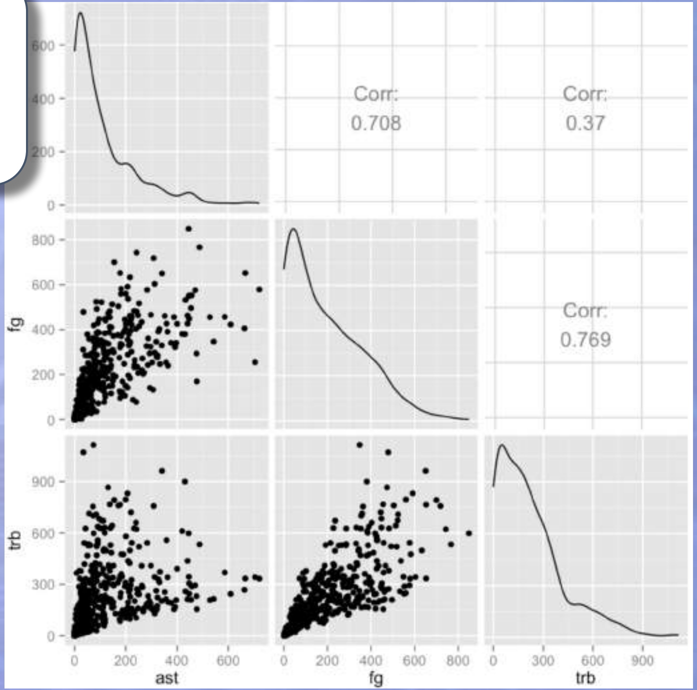
```python
import seaborn as sns
import matplotlib.pyplot as plt
sns.pairplot(nba[["ast", "fg", "trb"]])
plt.show()
```



```r
library(GGally)


nba %>%
  select(ast, fg, trb) %>%
  ggpairs()
```

Statistical Analysis





Packages

python

- pandas
- scikit-learn
- seaborn
- tensorflow
- pytorch
- matplotlib

R

- ggplot
- dplr
- shiny
- tidyr
- quantmod
- caret

Package Management

python

- pip/virtualenv
- pypi
- (ana)conda
- pyenv

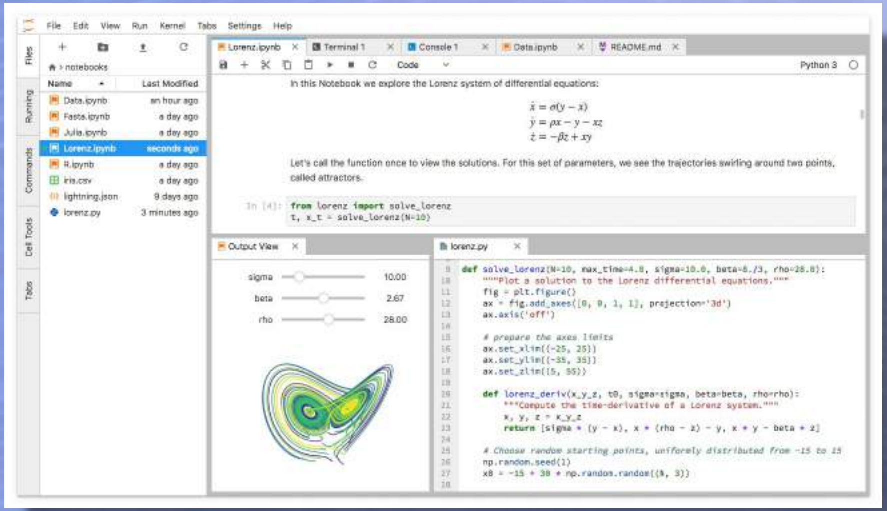
R

- builtin
- CRAN
- (ana)conda

Integrated Development Environment

Jupyter Lab

R Studio



notebooks

Name	Last Modified
Data.ipynb	an hour ago
Fasta.ipynb	a day ago
Julia.ipynb	a day ago
Lorenz.ipynb	seconds ago
R.ipynb	a day ago
iris.csv	a day ago
lightning.json	9 days ago
lorenz.py	3 minutes ago

In this Notebook we explore the Lorenz system of differential equations:

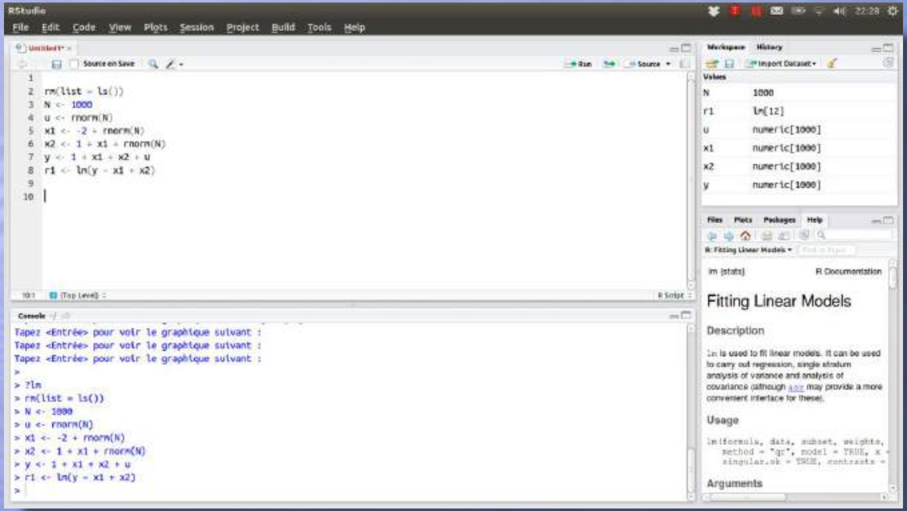
$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

Let's call the function once to view the solutions. For this set of parameters, we see the trajectories swirling around two points, called attractors.

```
In [4]: from lorenz import solve_lorenz
        t, x_t = solve_lorenz(N=10)
```

sigma	10.00
beta	2.67
rho	28.00

```
def solve_lorenz(N=10, max_time=4.0, sigma=10.0, beta=8./3, rho=28.0):
    """Plot a solution to the Lorenz differential equations."""
    fig = plt.figure()
    ax = fig.add_axes([0, 0, 1, 1], projection='3d')
    ax.axis('off')

    # prepare the axes limits
    ax.set_xlim((-25, 25))
    ax.set_ylim((-35, 35))
    ax.set_zlim((5, 55))

    def lorenz_deriv(x_y_z, t0, sigma=sigma, beta=beta, rho=rho):
        """Compute the time-derivative of a Lorenz system."""
        x, y, z = x_y_z
        return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]

    # Choose random starting points, uniformly distributed from -15 to 15
    np.random.seed(1)
    x0 = -15 + 30 * np.random.random((N, 3))
```



```

rm(list = ls())
N <- 1000
u <- rnorm(N)
x1 <- -2 + rnorm(N)
x2 <- 1 + x1 + rnorm(N)
y <- 1 + x1 + x2 + u
r1 <- lm(y ~ x1 + x2)

```

```
Tapez <Entrée> pour voir le graphique suivant :
Tapez <Entrée> pour voir le graphique suivant :
Tapez <Entrée> pour voir le graphique suivant :
>
> ?lm
> rm(list = ls())
> N <- 1000
> u <- rnorm(N)
> x1 <- -2 + rnorm(N)
> x2 <- 1 + x1 + rnorm(N)
> y <- 1 + x1 + x2 + u
> r1 <- lm(y ~ x1 + x2)
>
```

Values

N	1000
r1	lm[12]
u	numeric[1000]
x1	numeric[1000]
x2	numeric[1000]
y	numeric[1000]

lm {stats} R Documentation

Fitting Linear Models

Description

lm is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance (although aov may provide a more convenient interface for these).

Usage

```
lm(formula, data, subset, weights,
   method = "qr", model = TRUE, x =
   singular.ok = TRUE, contrasts =
```

Arguments

Analytics Back-End Integration

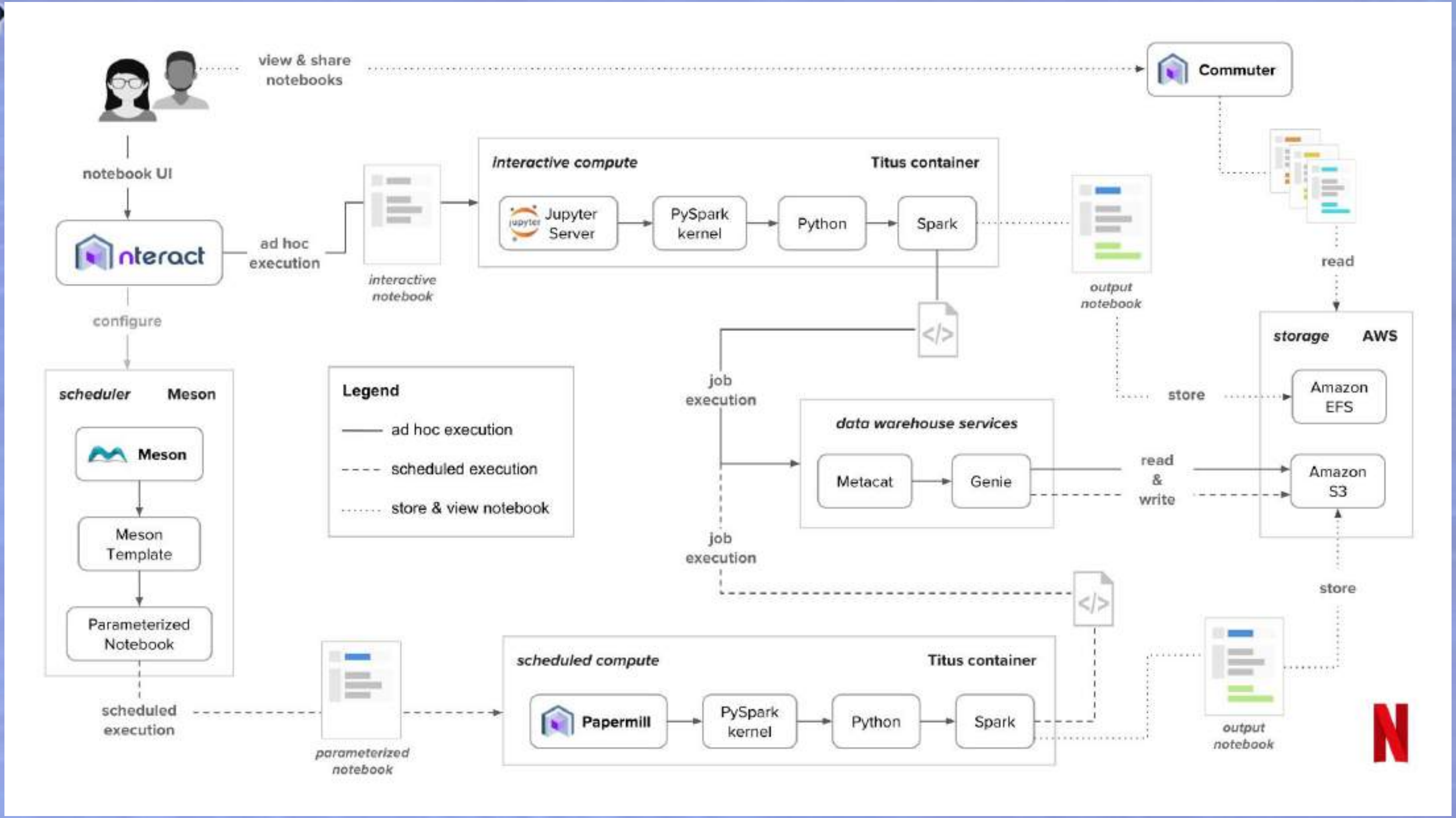
APACHE Spark™

NVIDIA®

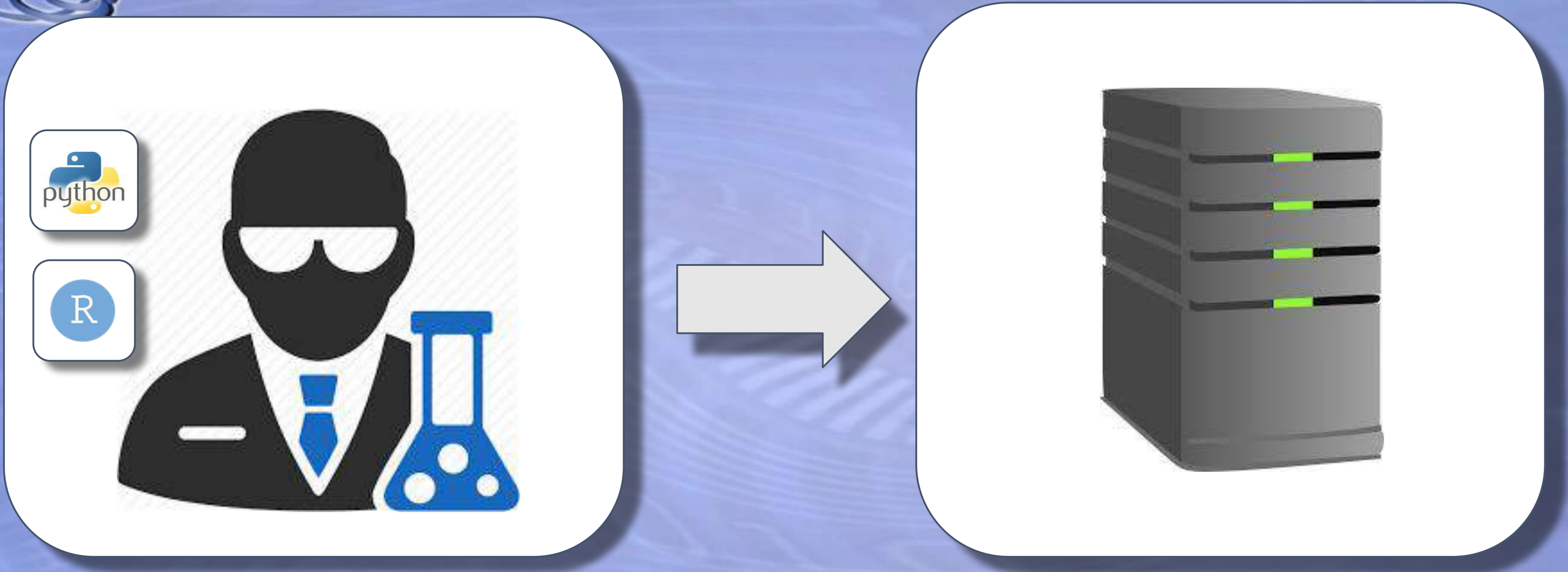
python

R

Netflix Notebook Infrastructure



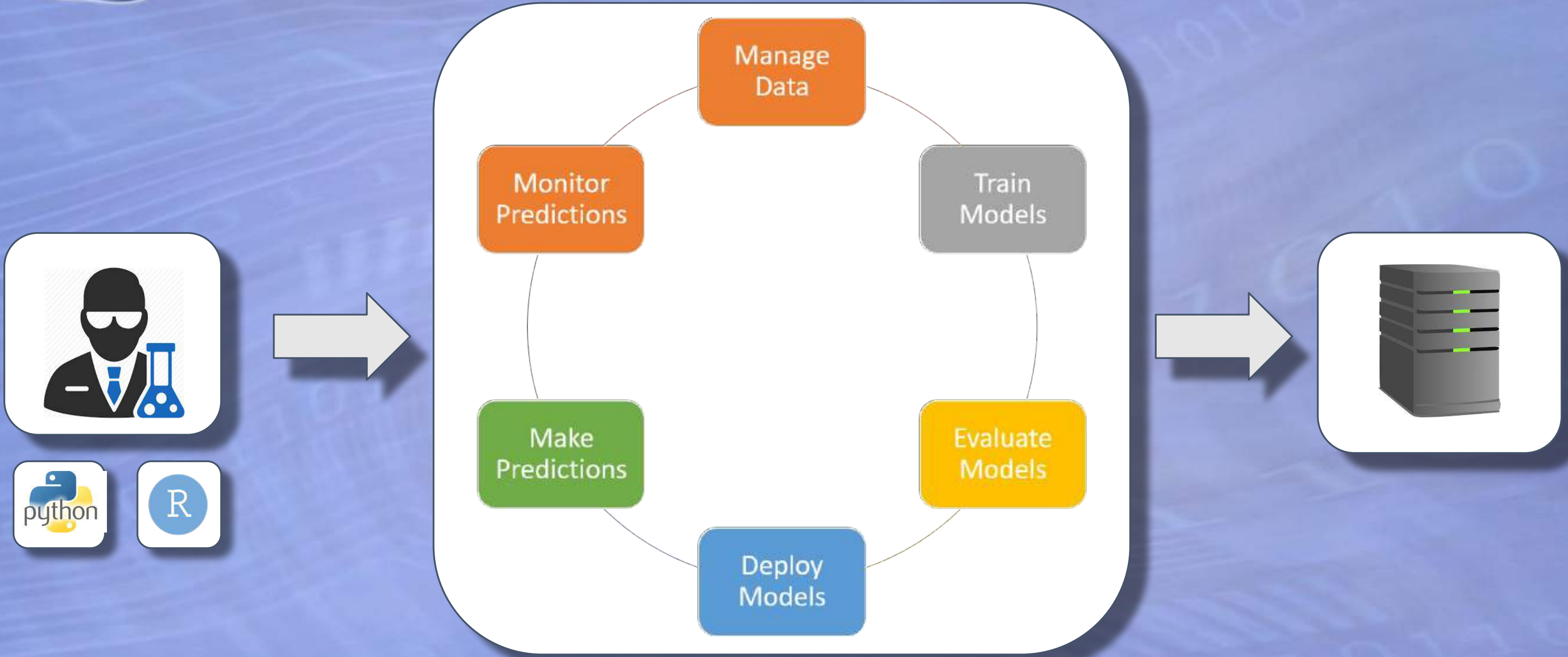
Productionalization



Experiments

Production

{Data Science, ML,AI} - As - A - Service



Manage Data

Train Models

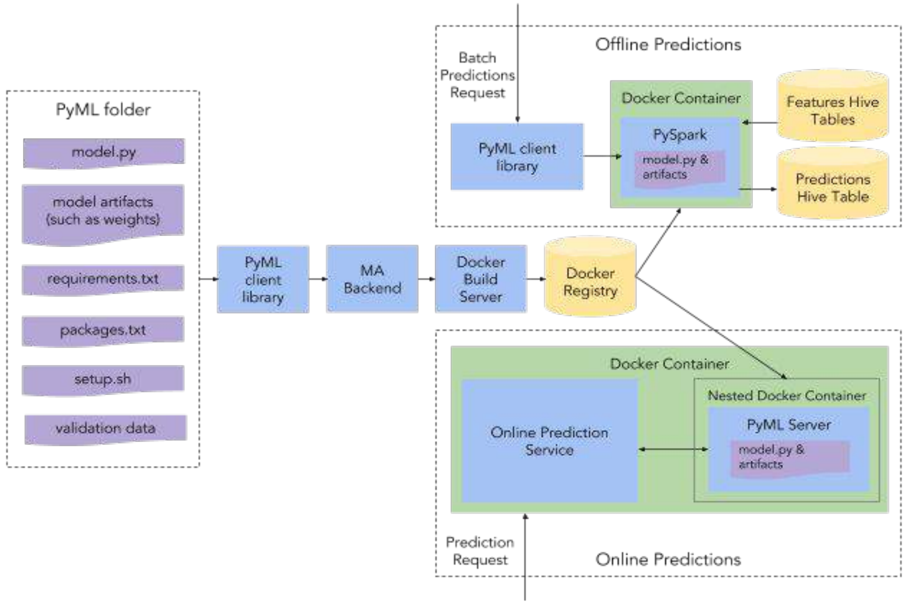
Evaluate Models

Deploy Models

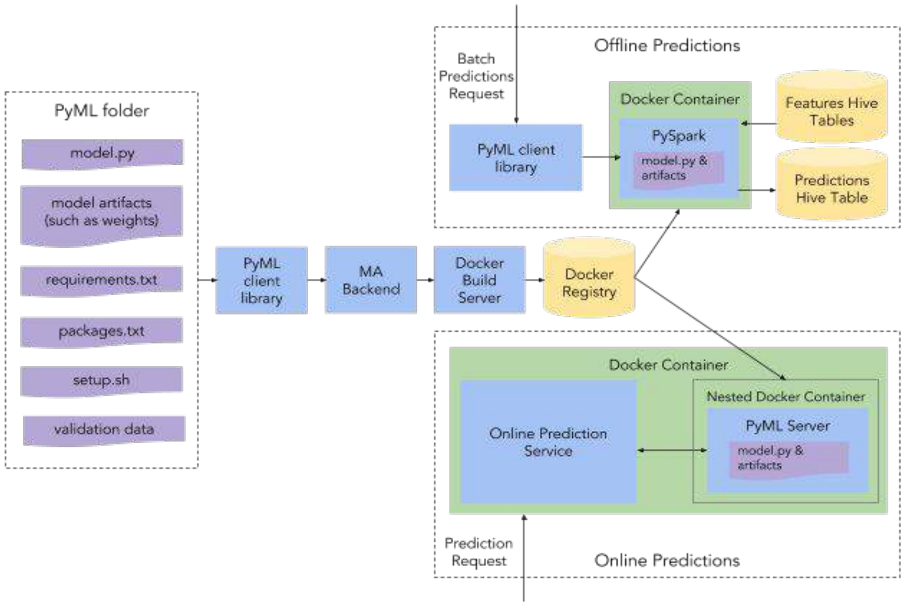
Make Predictions

Monitor Predictions

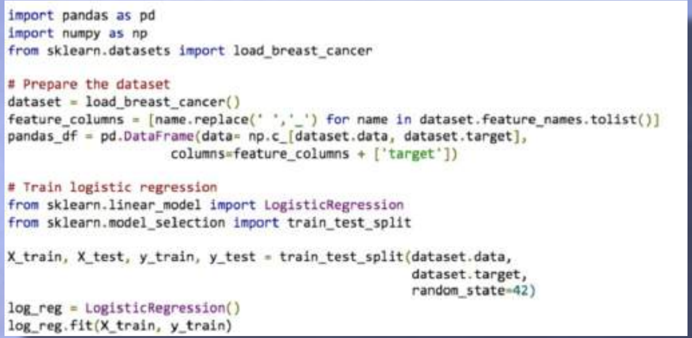
Uber’s PyML


PyML folder
model.py
model artifacts (such as weights)
requirements.txt
packages.txt
setup.sh
validation data
PyML client library
MA Backend
Docker Build Server
Docker Registry
Offline Predictions
Batch Predictions Request
PyML client library
Docker Container
PySpark
model.py & artifacts
Features Hive Tables
Predictions Hive Table
Docker Container
Online Prediction Service
Nested Docker Container
PyML Server
model.py & artifacts
Prediction Request
Online Predictions

Uber’s PyML

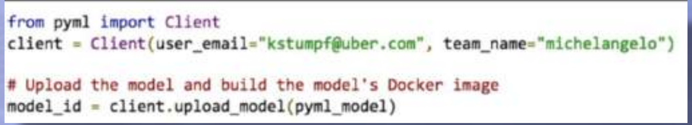


Train An ML Model:



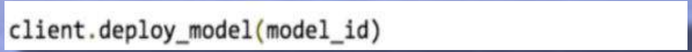
```
import pandas as pd
import numpy as np
from sklearn.datasets import load_breast_cancer

# Prepare the dataset
dataset = load_breast_cancer()
feature_columns = [name.replace(' ','_') for name in dataset.feature_names.tolist()]
pandas_df = pd.DataFrame(data= np.c_[dataset.data, dataset.target],
                     columns=feature_columns + ['target'])

# Train logistic regression
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(dataset.data,
                                                    dataset.target,
                                                    random_state=42)
log_reg = LogisticRegression()
log_reg.fit(X_train, y_train)
```

Dockerize:



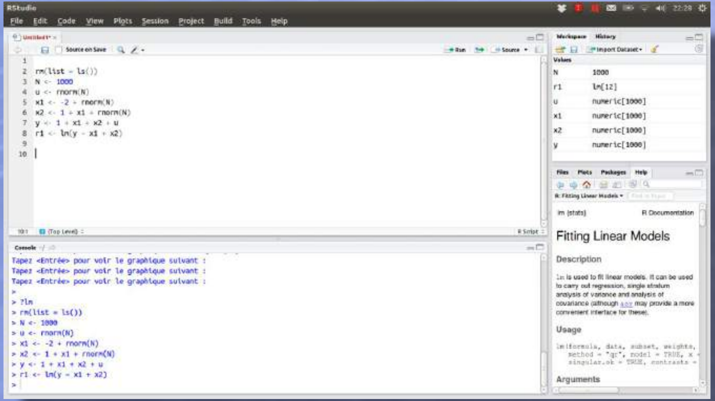
```
from pyml import Client
client = Client(user_email="kstumpf@uber.com", team_name="michelangelo")

# Upload the model and build the model's Docker image
model_id = client.upload_model(pyml_model)
```

Deploy:



```
client.deploy_model(model_id)
```

R Server



```
rm(list = ls())
N <- 1000
u <- rnorm(N)
x1 <- -2 + rnorm(N)
x2 <- 1 + x1 + rnorm(N)
y <- 1 + x1 + x2 + u
r1 <- lm(y ~ x1 + x2)
```

```
Tapez <Entrée> pour voir le graphique suivant :
Tapez <Entrée> pour voir le graphique suivant :
Tapez <Entrée> pour voir le graphique suivant :
>
> ?lm
> rm(list = ls())
> N <- 1000
> u <- rnorm(N)
> x1 <- -2 + rnorm(N)
> x2 <- 1 + x1 + rnorm(N)
> y <- 1 + x1 + x2 + u
> r1 <- lm(y ~ x1 + x2)
>
```

Values

```
N     1000
r1    lm[12]
u     numeric[1000]
x1    numeric[1000]
x2    numeric[1000]
y     numeric[1000]
```

lm {stats} R Documentation

Fitting Linear Models

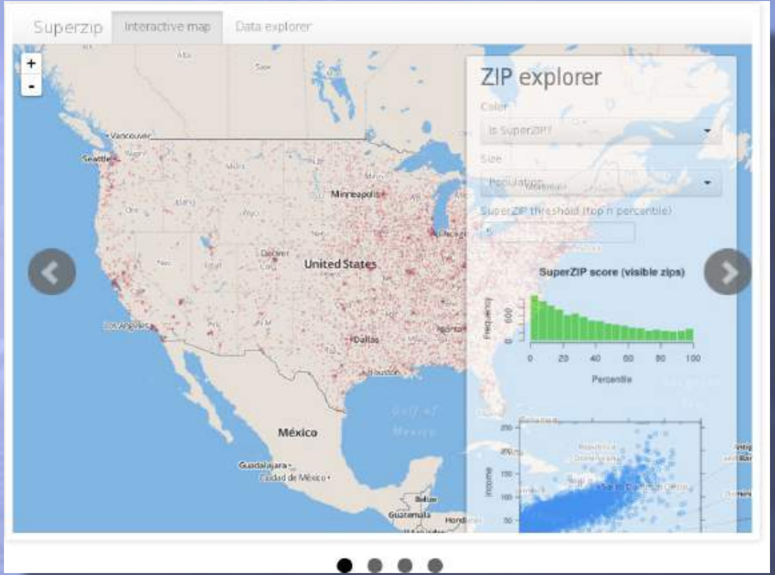
Description

lm is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance (although aov may provide a more convenient interface for these).

Usage

```
lm(formula, data, subset, weights,
   method = "qr", model = TRUE, x =
   singular.ok = TRUE, contrasts =
```

Arguments



Who’s Better ?

VS

Adding A Language

- ✓ It’s not just about the language.
- ✓ Consider the broader ecosystem.
- ✓ The IDE is just as important as the language
- ✓ Does it fit within a platform / pipeline ?

Q & A

ActiveState®

Thank you to our panelists

- **Francois Ouellet**, Director of Development Practice, *Manulife*
- **George Williams**, Director of Data Science and Chief Evangelist, *GSI Technology*

ActiveState®

What’s Next

- Watch a demo: https://www.youtube.com/watch?v=c5AIxN9ehrI
- Get a demo marketing@activestate.com
- Contact us for the language build you need: platform@activestate.com

ActiveState®

ActiveState®

Where to find us

Tel: **1.866.631.4581**

Website: **www.activestate.com**

Twitter: **@activestate**

Facebook: **/activestatesoftware**

ActiveState®