

# SAS<sup>®</sup> Viya<sup>™</sup> on Oracle Cloud

October 2017

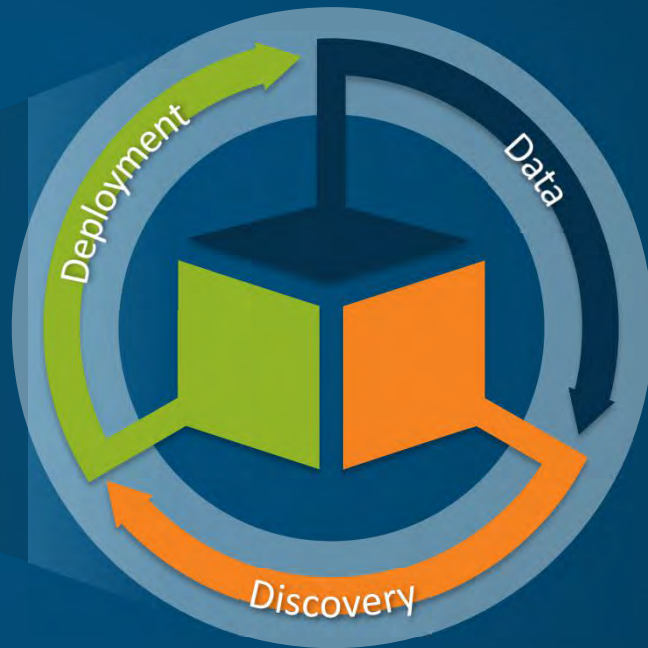


# 83,000 SAS Customer Sites

Over 50% rely on Oracle Database, Oracle Exadata or Oracle Big Data Appliance



# The SAS Platform



# Inside the SAS Platform

The SAS Platform – *for data*  
Producing data that cultivates analytics assets



The SAS Platform – *for discovery*  
Building analytics that solve any question



The SAS Platform – *for deployment*  
Realize the value from analytics assets in action



## Source-based Engines

### In-Stream



### In-Hadoop



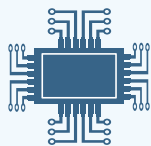
### In-Database



Parallel & Serial, Pub / Sub,  
Web Services, MQs

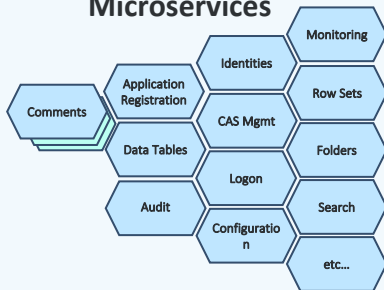
# SAS® Viya™

## In-Memory Runtime Engine



## Cloud Analytic Services (CAS)

## Microservices



## Solutions



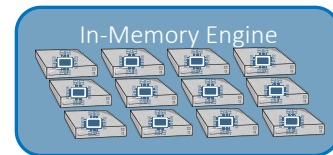
## APIs



## Platform



# Cloud Analytic Services (CAS) In-Memory Engine



The CAS in-memory engine is a fast, scalable, and resilient run-time environment for data management and analytics for SAS® Viya™



## Fast

- Multi-threaded
- Distributed In-Memory
- Inter-node Communication



## Scalable

- Single Machine to Distributed MPP
- Memory-mapping to process data larger than physical memory

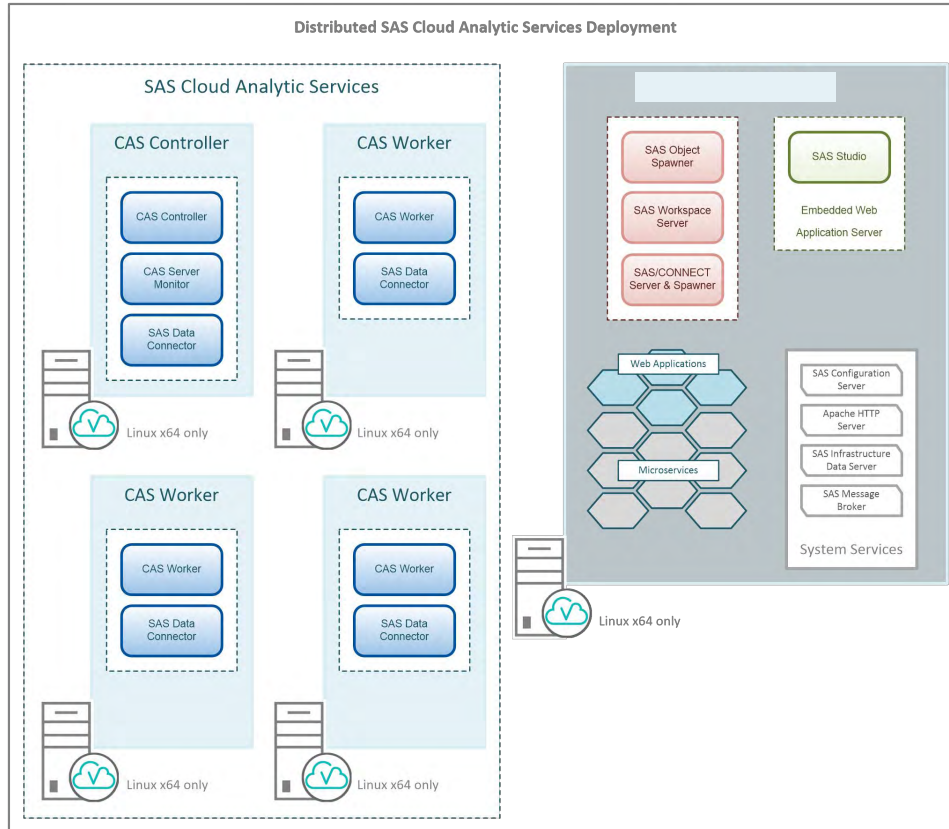


## Resilient

- Fault-tolerant to Node and Network Failures
- Worker and Controller\* Failover (*\*New feature in Nov 2017*)
- Session Independence

# SAS Viya Multi-Machine Deployment

## Distributed Cloud Analytics Services



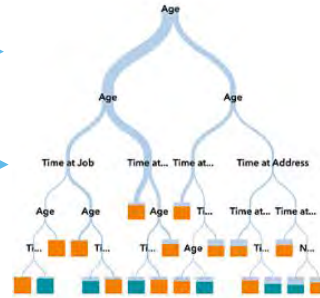
# Consistent Results across Usage Approaches

Visual Interfaces



Open APIs

Programming Interfaces

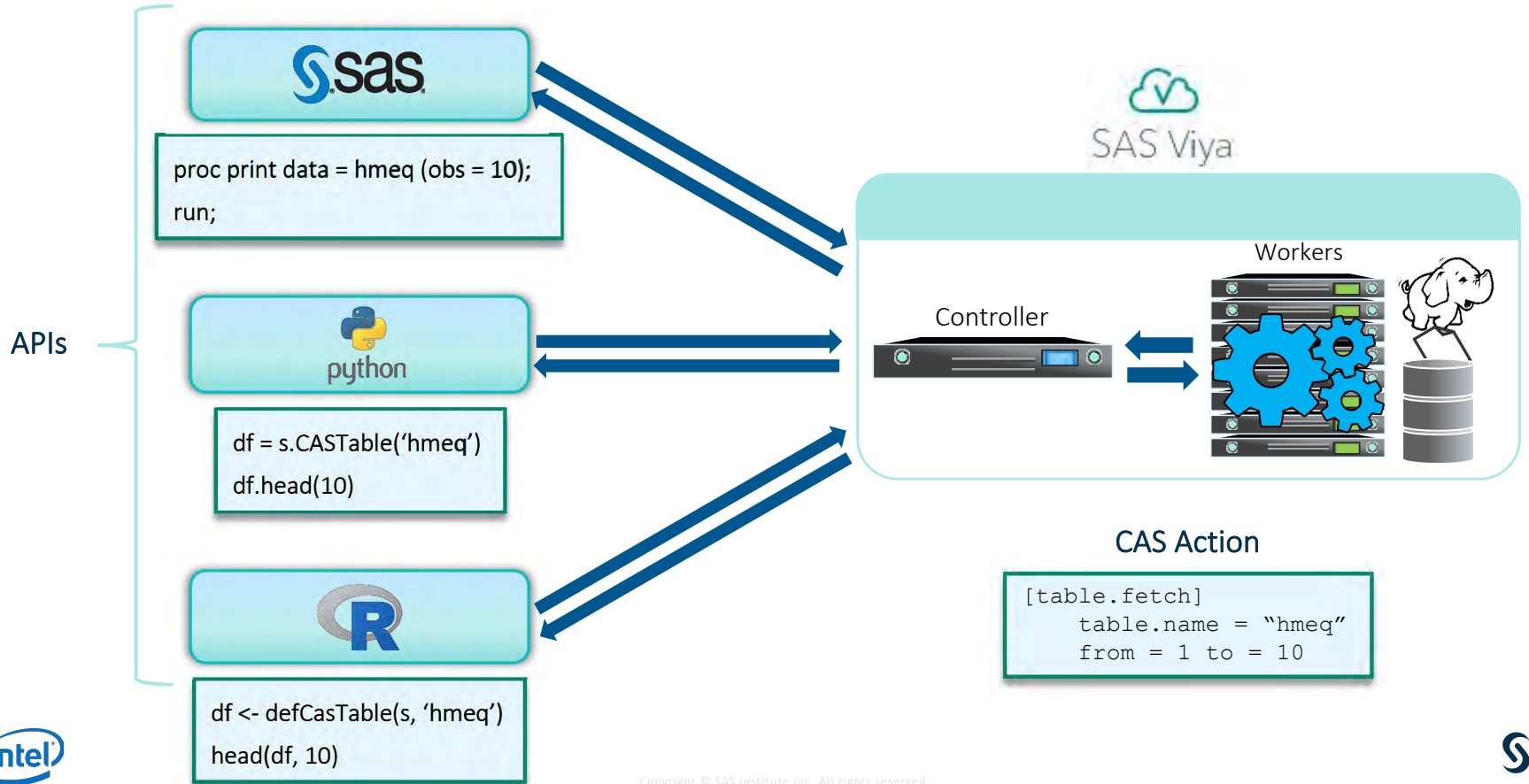


API Interfaces





# API's provide consist use of CAS Actions



# Open API Examples



This screenshot shows the GitHub interface for the repository `sas-viya-programming`. The file `Data Mining Workflow Python.ipynb` is selected, showing its metadata (3400 lines, 3399 sloc, 232 KB) and a preview of its content. The notebook content includes a title, a sub-heading, a note about the `swat` package, and a code cell for setting up the Jupyter environment.

```
In [1]: # Import necessary packages and modules
import swat
import pandas as pd
import collections
from matplotlib import pyplot as plt
swat_options.cas.print_messages = False
%matplotlib inline

# Define directory and data file name
indata_dir = '/viyafiles'
indata = 'hmeq'
```



This screenshot shows the GitHub interface for the repository `sas-viya-programming`. The file `DataMiningWorkflowR.pdf` is selected, showing its metadata (210 KB) and a preview of its content. The notebook content includes a title, a sub-heading, and a code cell for loading necessary packages in R.

```
# Load necessary packages
library('swat')

## SWAT 1.0.0
library('ggplot2')
library('reshape2')
options(cas.print.messages = FALSE)
```



# Analytics on SAS<sup>®</sup> Viya<sup>™</sup>

Visualization



Statistics &  
Machine Learning



Forecasting



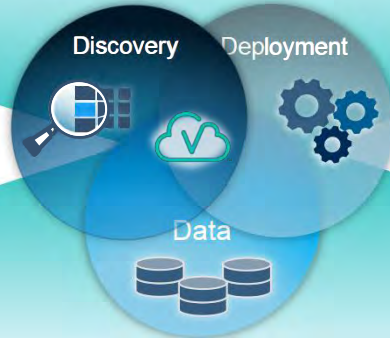
Text Mining



Econometrics



Optimization



Interactive



Programmatic



Automated



Modern

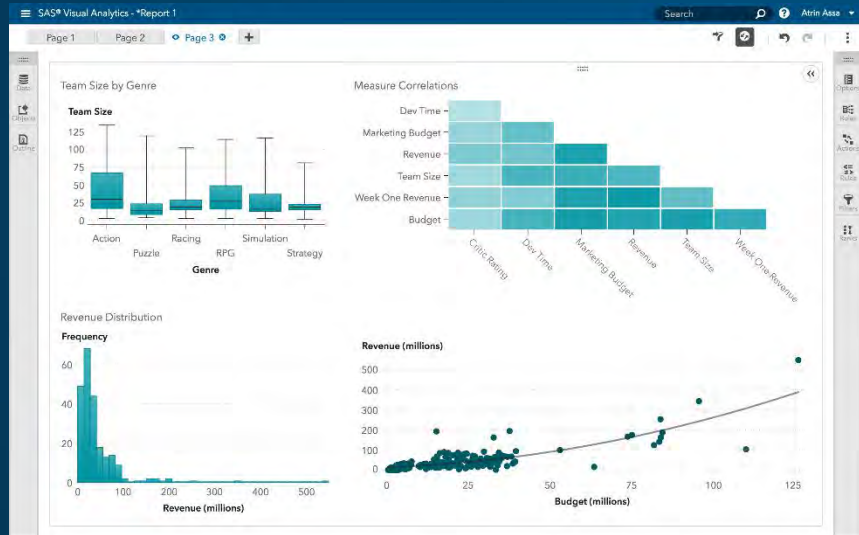


Approachable





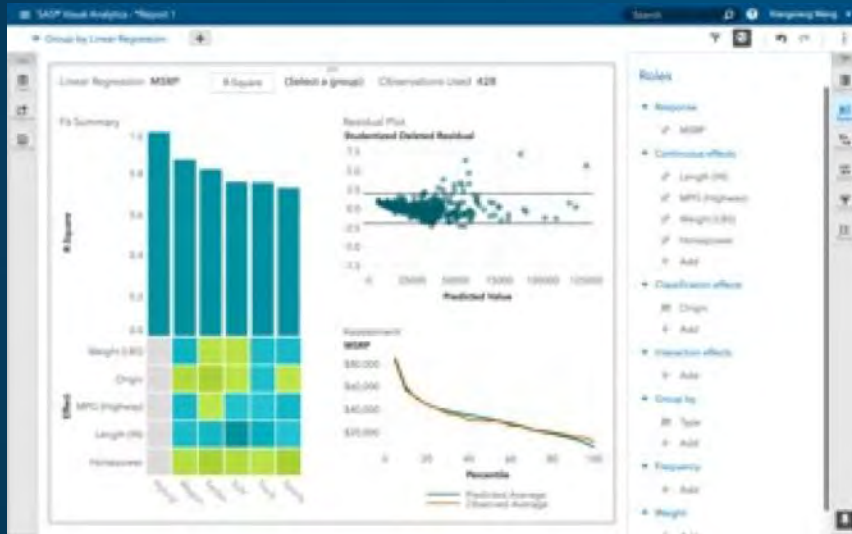
# SAS® Visual Analytics - Exploration



- Discover relationships, trends, outliers
- Analytics driven visualizations
- Descriptive statistics, correlation
- Forecasting and scenario analysis
- Decision trees
- Text analytics



# SAS® Visual Statistics - Predictive Analytics



- Linear Regression
- Logistic Regression
- GLM Regression
- Decision Tree
- Group-By Processing



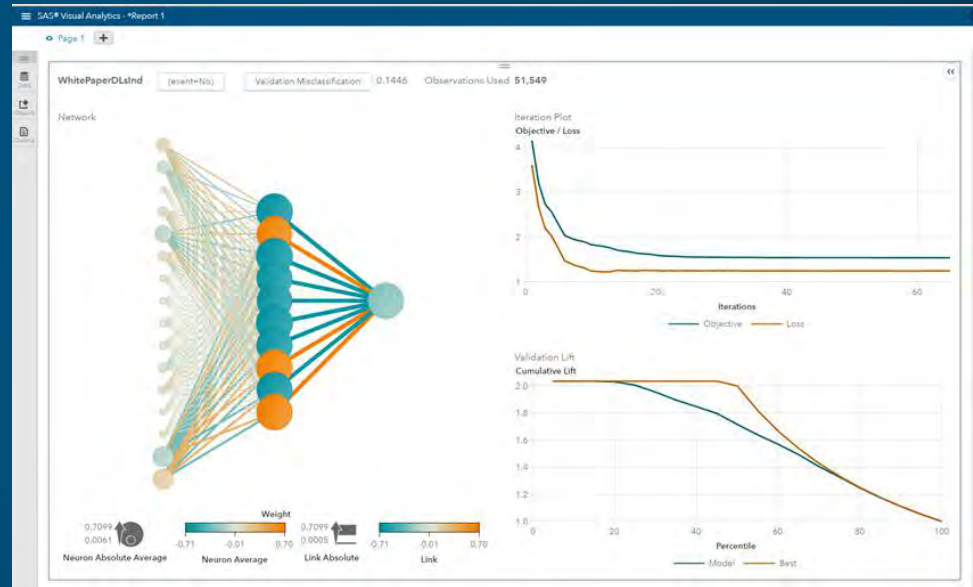
# SAS® Visual Data Mining and Machine Learning

- Machine Learning Techniques

- Factorization Machine
- Forest
- Gradient Boosting
- Neural Network
- Support Vector Machine

- Common Features

- Training-validation
- Auto-tuning
- Model Assessment
- Score Code or Analytic Store
- Model comparison
- Ability to export model statistics into Excel



# Why SAS?

DIVERSITY.

SCALE.

TRUST.





Oracle OpenWorld - Booth 2814

[sas.com/viya](https://sas.com/viya)

