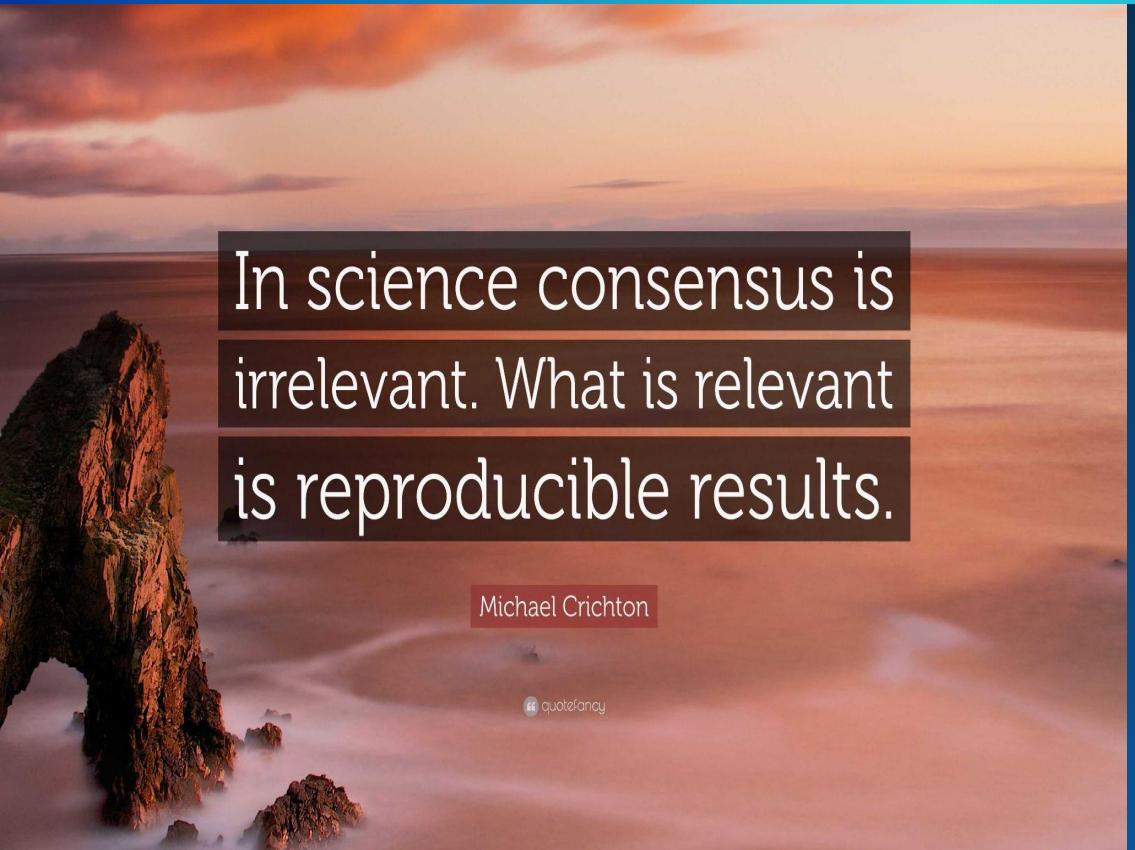
phuse·share
connect·share·advance

Single Day Event

Reproducible Realities: Fortifying Statistical Programming for the Era of Open Science

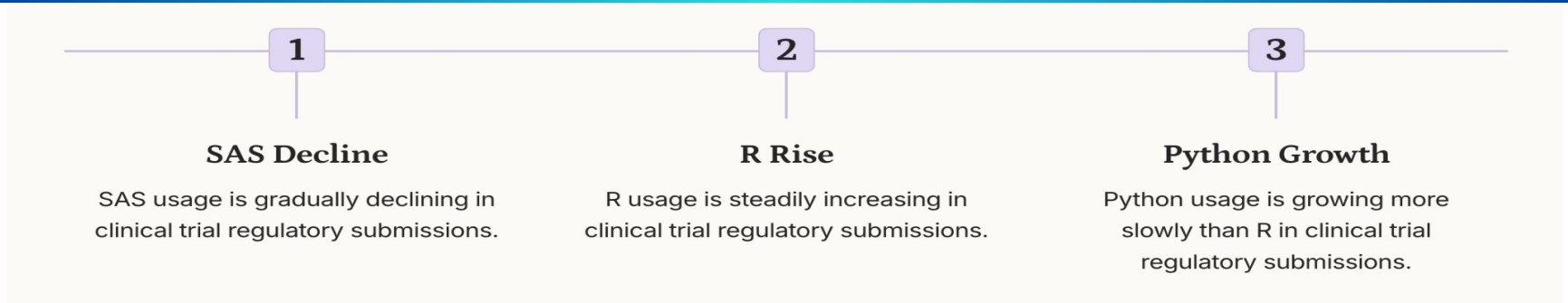
Priyanka Sawant,
Sycamore Informatics

Introduction



- Open Source Trends in Clinical research
- Challenges in Reproducibility
- Future-driven SCE Suite Framework
- Use cases
  - Generating TLF code with LLMs
  - LlaMa model within SCE suite
- Summary

Open Source Trends



Statistics on Reproducibility

Nature survey

- 70% of researchers reported unable to reproduce another scientist's findings
- 60% reported - unable to reproduce their own findings.

Study on RCTs

- 83.33% of data cannot be reproduced at the systematic review level
- 14.14% of data cannot be reproduced at the trial level
- 40.74% of data cannot be reproduced at the meta-analysis level

Ref: https://nap.nationalacademies.org/catalog/25303/reproducibility-and-replicability-in-science, https://www.sciencedirect.com/science/article/abs/pii/S1087079222001216
https://www.nature.com/articles/d42473-019-00004-y#:~:text=The%20reproducibility%20problem&text=A%202016%20Nature%20survey3,descriptions%20of%20interventions%20and%20methods.

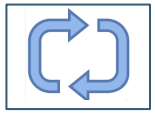
Challenges in Reproducibility

Version specific challenges in regulatory submissions

Frequent changes in R versions significantly impacts reproducibility, through deprecated functions and new version features

Issues in exploratory analysis

Resource assumptions play a critical role, with implications for local vs production environment scalability. Challenges in team collaborative tasks



Package version changes or validation status

Packages undergoing version changes or under validation, may impact existing program dependencies. Changes in OS versions also impact

Environment specific settings

Local user settings, memory usage have implications on the program behaviour Default CRAN mirror setup may impact if package versions change

Future-driven SCE suite framework

Future-driven SCE suite

Containerized compute environments as data science workbench

Flexible with various languages & packages

Ready integrations for multiple data sources

Flexible user workflows & Operational monitoring

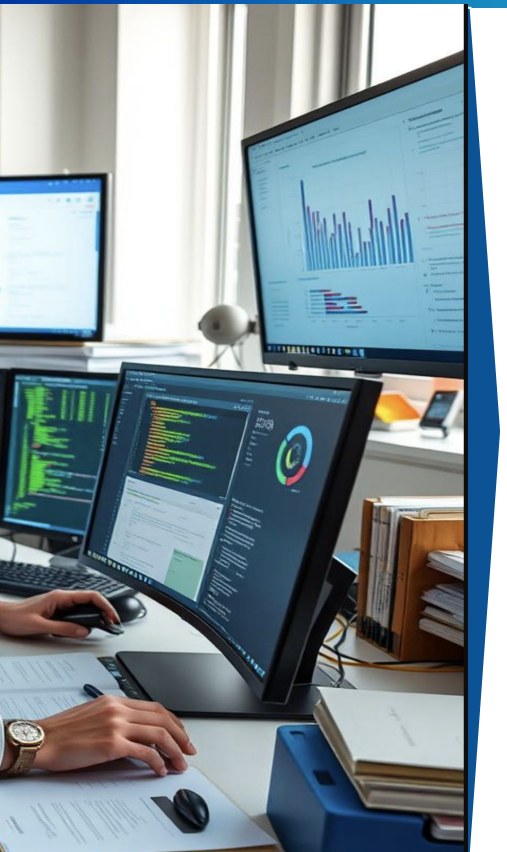
Publish & access role based Interactive Apps

Data security & traceability of analysis & interactive apps

Audit across all user actions and program executions

Scalability with High Performance resources

Containerized Compute Environment



Configured Compute with fixed versions

Containerized environments available as data science workbench with specific versions of R and Python, along with required packages, are defined and archived for future use.

Versatile Use Cases

These containerized environments are utilized by development IDEs, job execution, and published applications, ensuring consistency and reproducibility.

Accessibility

Development environments such as Rstudio, VSCode, Jupiter Notebook can be accessed from the web application directly. Program executions are run within the context of configured compute specs. Archived containers are readily available for future access, enabling seamless continuation of projects and applications.

Integrations and ecosystems

Compute

SAS
R
python
julia
Ruby
SAS Viya
BASH

Tools and IDEs

RStudio
jupyter
PyCharm
SAS Studio/EG
VS Code
UltraEdit

Data Stores & Sources

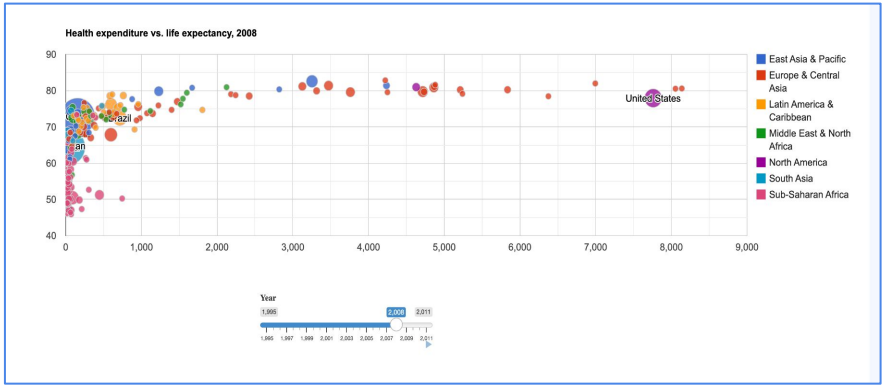
amazon REDSHIFT
.sas7bdat
CSV
PostgreSQL
amazon S3
medidata Rave
Veeva CTMS
Clinical Trial Management System
eClinical
Powered by elluminate
MDR

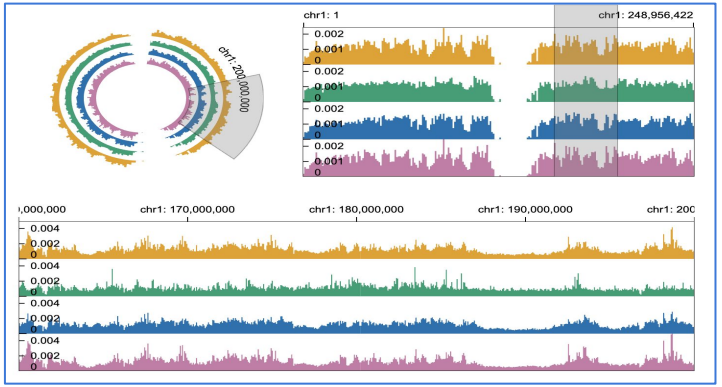
Packages & Libraries

Technology

CIS Benchmarks
docker
Amazon EKS
kubernetes
amazon web services
AUTO SCALING
Red Hat Enterprise Linux
GitHub
OpenZFS

Role based Interactive Apps





Developing interactive apps

Utilization of R Shiny and StreamLit for creating engaging and functional interactive applications, which can be published and launched within SCE

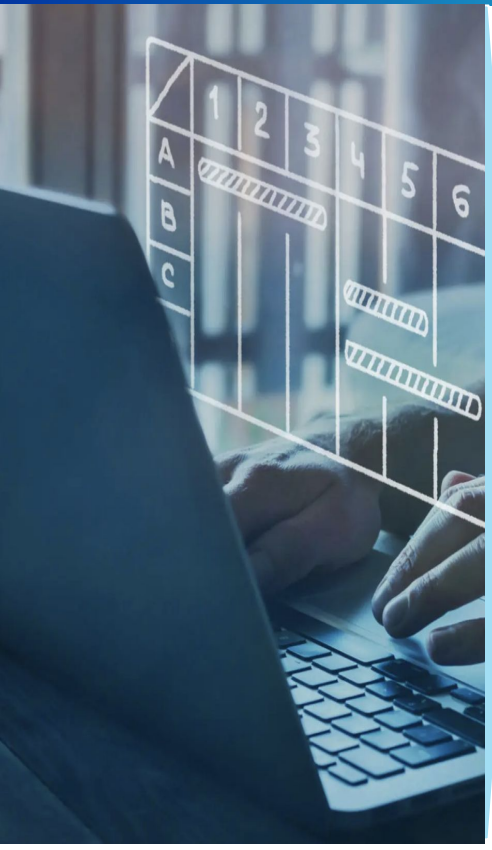
Visualizations

Present study data insights and operational progress through intuitive, customizable dashboards and charts that aid decision-making.

Secure Access

Control access to blinded outputs, ensuring clinicians, safety teams, and other non-SCE roles can securely view and interact with relevant data.

Flexible user workflows with Operational monitoring



Configurable Workflows

Granular access controls enable secure, role-based workflows for all users.

Real-Time Deliverable Tracking

Monitor project progress with instant visibility into deliverables and deadlines.

Faster Validations

Automated validations and issue resolution streamline the analysis process.

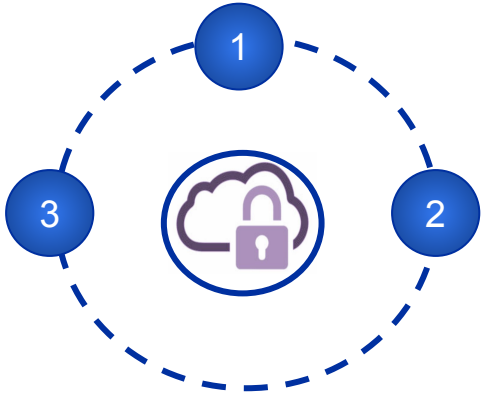
Interactive Dashboards

Gain at-a-glance insights into global operations with customizable dashboards.

Data Security, Traceability & Audit

Audit

Comprehensive audit logs track all user and program execution related actions, enabling full transparency and accountability for all data processing activities.



Data Security

Robust encryption, user authentication, and access controls across multi layered file system ensure sensitive clinical trial data is protected from unauthorized access or modification.

Traceability

Compute configurations traceable within IDE & web based executions allow in-depth review of data lineage, processing steps, and analytical decisions to meet regulatory requirements.

Scalability & Performance

High-Performance Computing (HPC)

Powerful supercomputers and computing clusters optimized for complex, compute-intensive workloads.

Servers and Clusters

Flexible on-premise infrastructure with scalable processing power and storage capacity.

Cloud Computing

Elastic, pay-as-you-go cloud resources for flexible, on-demand scaling of computing power.

High-Capacity Storage

Enterprise-grade storage solutions to handle large datasets and backups.

Database Systems

Robust, scalable database management systems for efficient data storage and retrieval.

Modeling and Analysis Tools

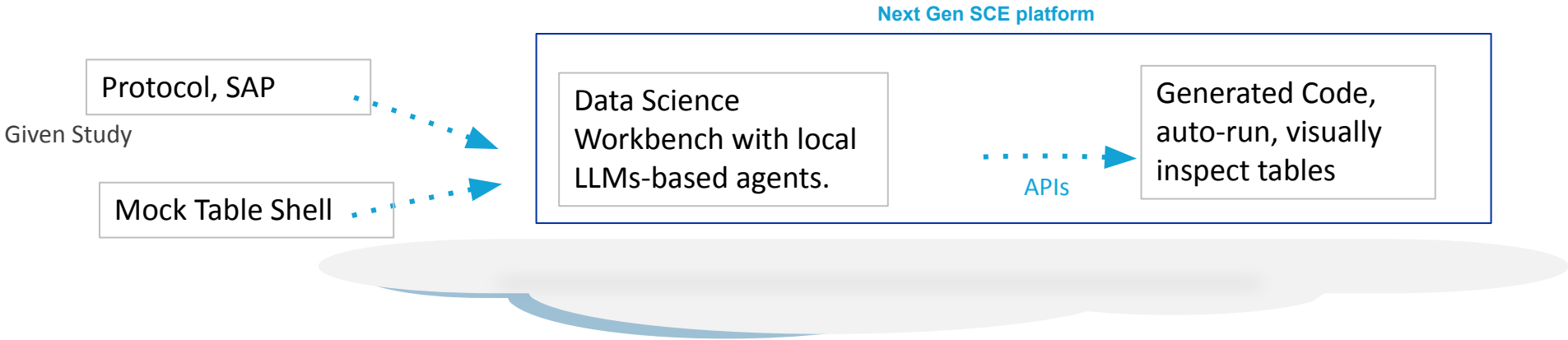
Specialized software for data modeling, simulation, and advanced analytics.

Use Cases

Generating Code for TLFs with LLMs

- Using SAP and a series of mock shells, generate SAS program that will create the table with real data.
- This can be used with dummy data initially or real data as data starts flowing.
- System learns the content of the mock shell using LLMs, then it further prompts code generation LLM to create SAS code to create the table

- Business Value
  - Generating programs faster
  - Reuse existing macros, coding conventions
  - Make users focus on analysis and not on automatable steps



LLM AI-based Code Generation and Execution

Shell | Code

SAP

PharmaCo_Oncology_SAP_V3.docx

1 Accounting of subjects with re-categorized reasons for treatment discontinuation and
2 Number of subjects in each treatment group by analysis population
3 Treatment History – ITT Population
4 Relapse History – ITT Population
5 Demography - ITT population
6 Exposure to study treatment - ITT population
7 Exposure to study treatment up to treatment discontinuation - ITT population
9 Baseline EDSS scores - ITT population
10 Baseline MRI evaluation - ITT population
11 Medical History at baseline – ITT population
12 Time on study – ITT population
13 Summary of annualized relapse rate (INEC confirmed relapses) - ITT population
14 Summary of annualized relapse rate (INEC confirmed relapses) up to Last PatientLas
19 Number of new or newly-enlarging T2 lesions at Week 96 - ITT population
20 Summary of time to 3-months sustained disability progression measured by increase
30 Summary of proportion of subjects relapsed (INEC confirmed relapses) -
31 Summary of proportion of subjects with a ≥7.5 point worsening from baseline at We

Shell | Code

Language

- (•) SAS
- ( ) Python
- ( ) R

Prompt

Write the table to subject_demog.html

Generate Code

1. Baseline Characteristics

```sas
libname adam './examples/data/adam/';

proc sql;
    create table demog as
    select
        'Subjects' as Characteristic,
        catx(' ', count(case when sex='M' then USUBJID er
        catx(' ', count(case when sex='F' then USUBJID er
        catx(' ', count(USUBJID), '(', put(count(USUBJID)
    from adam.adsl
    where saffl = 'Y';
quit;

ods html file='subject_demog.html' style=htmlblue;
title '1. Baseline Characteristics';
proc print data=demog noobs label;
    var Characteristic Male Female Total;
run;
ods html close;
```

Run Code

```
1                                          Altair SLC

NOTE: Copyright 2002-2024 World Programming, an Altair Co
NOTE: Altair SLC 2024 (05.24.01.00.001044)
      License: Altair Managed Licensing: Workstation, 10
NOTE: This session is executing on the MACOS platform and

1         libname adam './examples/data/adam/';
NOTE: Library adam assigned as follows:
      Engine:        SAS7BDAT
      Physical Name: /Users/jeffwinkler/repos/AI-Interns/

2
3         proc sql;
4             create table demog as
5             select
6                 'Subjects' as Characteristic,
7                 catx(' ', count(case when sex='M' then
7       ! sex='M' then USUBJID end) / count(USUBJID) * 10
8                 catx(' ', count(case when sex='F' then
8       ! sex='F' then USUBJID end) / count(USUBJID) * 10
```

Subjects by Gender

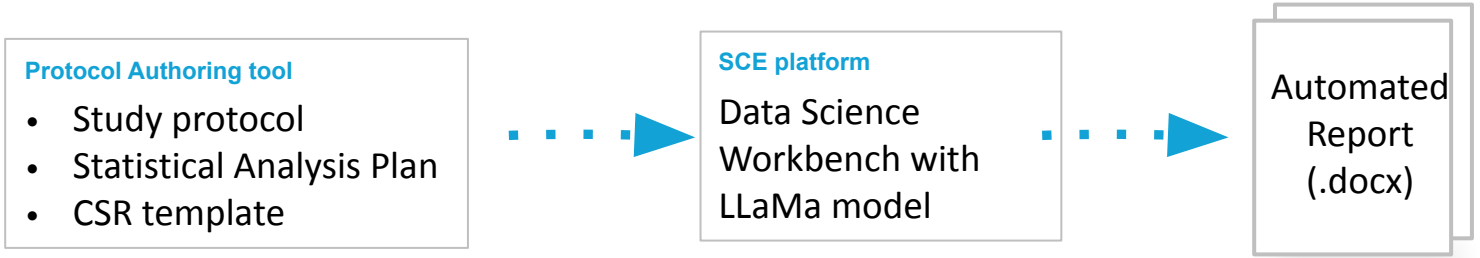
Category	Male	Female	Total
Subjects	111(43.7%)	143(56.3%)	254(100.0%)

LLaMa model with SCE suite

Facilitated the deployment of LLaMA model within a secure and scalable SCE suite framework.

- seamless integration of doc templates from Structured Protocol Authoring tool
- input data as structured documents such as protocol, SAP and the final TLFs
- generated into a study report document
- streamlined and efficient process for medical writing teams.

- Business Value
  - Reduction in doc authoring timelines
  - Lesser time spent on manual reviews
  - Reduction in report delivery timelines
  - Improved report content & format quality
  - Cost reduction based on medical writing efforts
  - Better regulatory compliance leading to faster drug approvals



Summary

- We covered the current industry challenges for implementing open source technologies, the proposed strategy for Future driven SCE suite and use cases where this approach has been successfully implemented
- With this framework, we are ensuring all analyses are reproducible from any machine and at anytime in the future
- Reproducibility is not just about the outputs, it has to be ensured with data security, traceability and scalability
- Investing in futuristic SCEs platforms that cater to such multiple technologies, development tools and controlled compute environments is the need of the hour

Thank You

https://www.sycamoreinformatics.com/dsw

The Global Healthcare Data Science Community

Contact Channels
UK +44 1843 609600
office@phuse.global
phuse.global

Social Media
@phusetwitta
/phusebook
/phusetube
/company/phuse