Institute of Primate Research

STANDARD OPERATING PROCEDURE (SOP) DOCUMENT

Handling large datasets and trend detection

SOP No.	Issue Number	Issue Date	Revision Status	Revision Date
SOP/KIPRE/RPD/DSAS/3.1.76	Version 01	October 2025	-	-

Approvals

	Name	**Signature**	**Date**
Developed by:	_Patrick Waweru Mwaura_	_____	**_6th October; 2025_**
	_____	_____	_____
	_____	_____	_____
Reviewed by:	_____	_____	_____
Approved by:	_____	_____	_____

Table of Contents

1. PURPOSE

To provide a standardized framework for the management, analysis, and interpretation of large-scale biomedical, ecological, genomic, and public health datasets within DS&AS. This SOP ensures efficient data ingestion, preprocessing, trend detection, and visualization, while maintaining reproducibility, scalability, and compliance with institutional SOPs (1, 6, 7, 8, and 9) and regulatory requirements (Kenya Data Protection Act, 2019). The procedures outlined facilitate the timely identification of temporal, spatial, and epidemiological patterns, supporting evidence-based decision-making and reporting.

2. SCOPE

Covers all DS&AS-supported projects that generate, process, or analyze large-scale datasets—including biomedical, ecological, genomic, and public health data—for trend detection, pattern recognition, and predictive insights.

3. PERSONS RESPONSIBLE:

- **Data Engineer:** Oversees large dataset ingestion, storage, and access pipelines.
- **Data Scientist / Biostatistician:** Performs statistical analyses and detects temporal, spatial, and epidemiological trends.
- **Head of DS&AS:** Reviews methodology, ensures compliance with institutional standards, and monitors operational efficiency.

4. FREQUENCY

- Applied continuously for projects involving high-volume, real-time, or longitudinal datasets.
- Reviewed at least annually to optimize workflows, storage, and trend-detection methodologies.

5. MATERIALS

- **Big Data Platforms:** Hadoop, Apache Spark, SQL and NoSQL databases, PostgreSQL — for storage, querying, and distributed processing of high-volume datasets.
- **Trend Detection & Analytics Tools:** Time-series analysis (ARIMA, ETS), generalized additive models (GAMs), Cox proportional hazards models, anomaly

detection algorithms, geospatial and spatiotemporal modelling packages, and statistical computing libraries (R, Python, Julia).

- **Visualization & Reporting Tools:** R Shiny, Tableau, PowerBI, SvelteKit dashboards, QGIS/ArcGIS for spatial analytics, and interactive reporting frameworks for exploratory data analysis and trend presentation.
- **Computational Resources:** High-performance computing (HPC) clusters or cloud-based platforms (AWS, Azure, GCP) to support large-scale data ingestion, processing, and model execution.

6. PROCEDURE

- **Data Ingestion:** Import datasets into scalable storage solutions (SQL/NoSQL databases, distributed file systems such as Hadoop/Spark). For geospatial data, ensure proper coordinate reference systems and metadata capture.
- **Preprocessing:** Apply automated cleaning, deduplication, normalization, and standardization. For spatial datasets, validate geometries and handle missing spatial attributes.
- **Trend Analysis:**
- **Statistical Methods:** Apply time-series models (ARIMA, ETS), generalized additive models (GAMs), Cox models, and other regression frameworks.
- **Probabilistic & Machine Learning Models:** Use Hidden Markov Models (HMMs) for state-based pattern detection, Gaussian Process (GP) mixtures for modelling complex non-linear trends, and clustering or anomaly detection for high-dimensional datasets.
- **Geospatial & Spatiotemporal Modelling:** Detect spatial and spatiotemporal trends using QGIS, ArcGIS, or geospatial libraries in R/Python (sf, raster, geopandas).
- **Visualization:** Generate interactive dashboards for monitoring trends, anomalies, and spatial patterns using R Shiny, Tableau, PowerBI, or SvelteKit. Include maps, time-series plots, and summary metrics for real-time exploration.

- **Archiving:** Store processed datasets, scripts, model outputs, and intermediate results in the central repository with clear versioning and metadata documentation.
- **Review & Updating:** Conduct annual audits of pipeline scalability, model performance, and dashboard functionality. Update workflows with new data, methods, or models as needed.

7. REFERENCES

1. Kenya Data Protection Act (2019) – Legal framework for data handling and privacy.
2. FAIR Principles – Wilkinson et al., 2016, for data findability, accessibility, interoperability, and reuse.
3. KIPRE Institutional Data Governance Guidelines – Policies for data access, storage, and sharing (linked to SOPs 1, 6, 7, 8, 9).
4. Big Data and Trend Detection Best Practices – Sandve et al., 2013, *PLoS Comput Biol.*
5. Apache Hadoop and Apache Spark Documentation – Guidelines for scalable data processing.
6. SQL and NoSQL Database Best Practices – PostgreSQL, MongoDB official documentation.
7. Time-Series and Machine Learning Trend Detection References:
    i. Box et al., 2015. *Time Series Analysis: Forecasting and Control.*
    ii. Hastie, Tibshirani, & Friedman, 2009. *The Elements of Statistical Learning.*
    iii. Chandola et al., 2009. *Anomaly Detection: A Survey.*

8. APPENDIX

Appendix A – Data Preprocessing Checklist

- Remove duplicates and missing values.
- Standardize variable names and formats.
- Apply anonymization/pseudonymization for sensitive data.
- Log preprocessing steps for reproducibility.

Appendix B – Trend Detection Methods Reference Table

Method	Type	Application	Software/Package
ARIMA	Time-series	Epidemiological trends, seasonal patterns	R `forecast`, Python `statsmodels`
GAMs	Regression	Non-linear trend detection	R `mgcv`
Cox Proportional Hazards	Survival	Disease progression trends	R `survival`
Clustering	ML	Anomaly detection, pattern recognition	Python `scikit-learn`
Anomaly Detection	ML	Rare events detection	R `anomalize`, Python `pyod`

etc

Appendix C – Dashboard & Visualization Templates

- R Shiny apps for real-time monitoring.
- Tableau/PowerBI dashboards for spatial and temporal trends.
- Standardized color schemes, labels, and metadata annotations for reproducibility.

Appendix D – Version Control & Archiving Guidelines

- Use Git repositories for all code and scripts.
- Maintain timestamped versioned datasets in central repository.
- Archive processed datasets with metadata for traceability.