Value at Risk (VaR) in Socio-Economic and Environmental Risk Evaluation

Mark A. Asch

ML-PREP 2025

Abstract

Value at Risk (VaR) is a statistical technique that originated in financial risk management but has been increasingly applied to socio-economic and environmental contexts. We describe some examples of its application and then present an interactive dashboard for Landslide Susceptibility VaR.

1. VaR for the Environment

Climate Change Risk Assessment

- **Coastal Infrastructure Protection**: VaR models help quantify potential economic losses from sea-level rise and storm surge events, enabling city planners to justify protective infrastructure investments.
- **Agricultural Planning**: VaR calculations estimate crop yield risks under various climate scenarios, allowing farmers and governments to develop appropriate adaptation strategies.

Natural Disaster Management

- **Flood Risk Management**: Environmental agencies use VaR to express the maximum expected economic damage from flooding within a specific probability (e.g., "there is a 95% confidence that flood damages will not exceed $50 million in the next year").
- **Drought Risk Assessment**: Water resource managers apply VaR to estimate water shortage risks and their economic implications for communities and industries.

Public Health Risk Evaluation

- **Pandemic Preparedness**: Health authorities use VaR to quantify the potential economic impact of disease outbreaks, helping to justify investment in preventive measures.
- **Heat Wave Response Planning**: Cities employ VaR models to estimate mortality and healthcare burden risks during extreme heat events, informing resource allocation.

Environmental Policy Decision-Making

- **Cost-Benefit Analysis**: Policymakers use VaR to evaluate the economic risks of environmental degradation against the costs of regulation.
- **Biodiversity Conservation**: Conservation organizations apply VaR to quantify potential economic losses from ecosystem service degradation.

Social Impact Investing

- **Impact Investment Risk Assessment**: Investors use VaR to balance financial returns against social and environmental risk factors.
- **Sustainable Development Projects**: Development agencies employ VaR to evaluate the risk profile of infrastructure projects in vulnerable communities.

VaR's strength in these contexts comes from its ability to express complex risks in simple, probabilistic terms (e.g., "there is a 99% chance that losses will not exceed X amount over Y time period"), making it valuable for communicating risks to stakeholders and decision-makers.

2. Landslide Susceptibility VaR Dashboard

We propose an interactive dashboard for landslide susceptibility mapping based on Value at Risk (VaR) principles that enables decision-makers to analyze risk-cost relationships. The dashboard allows decision-makers to:

1. **Adjust risk confidence levels** (50-99%) to see how different risk tolerances affect potential losses.
2. **Select different regions** with varying characteristics (population density, slope, vegetation, rainfall, etc.).
3. **Set time horizons** (1-50 years) to evaluate short and long-term risks.
4. **Modify mitigation budgets** to see cost-benefit relationships and ROI.

Key Features

- **Value at Risk (VaR) Calculation**: Shows the maximum economic loss expected at your chosen confidence level.
- **Comprehensive Impact Assessment**: Includes:
  - Economic losses (infrastructure damage, land value).
  - Human impact (casualties, displaced persons).
  - Income loss estimates.
  - Land area affected.
- **Visualization Components**:
  - VaR curve showing how risk changes with confidence level.
  - Cost breakdown pie chart.
  - Mitigation ROI analysis.
  - Region-specific risk factors.

Practical Applications

This dashboard supports decision-making by helping to:

1. **Prioritize interventions** across different regions.
2. **Optimize resource allocation** by finding the most efficient mitigation budget.
3. **Justify investments** by demonstrating clear ROI for risk reduction measures.
4. **Communicate risk** to stakeholders in clear, financial terms.

The tool bridges the gap between technical risk assessment and practical decision-making by expressing complex landslide risks in terms of their socio-economic impacts. The interactive visualization shows the relationship between economic impact, casualties, and mitigation costs - with all calculations based on Value at Risk methodology.

Overview of the Python version based on Dash

A Python class-based implementation that uses:

1. `dash` - A Python framework for building web applications (similar to how React can be used in a Java/TypeScript version).
2. `plotly` - For interactive data visualization.
3. `pandas` & `numpy` - For data manipulation.
4. `matplotlib` & `seaborn` - For optional static reporting capabilities.

Key Components of the Python Implementation

1. **Risk Calculation Function**: The `calculate_var()` function in Python implements the Value at Risk (VaR) methodology, with formulas for calculating:
   - Risk scores based on population, slope, vegetation, rainfall, and historical factors
   - Impact calculations for casualties, infrastructure loss, land loss, and displacement
   - Mitigation effectiveness and ROI
2. **Interactive Controls**: The dashboard provides four controls:
   - Confidence level slider (50-99%).
   - Region selection dropdown.
   - Time horizon slider (1-50 years).
   - Mitigation budget slider (0-10M).
3. Data Generation Functions

   - Create VaR curve data showing how risk changes with confidence level.

   - Generate data showing the effects of different mitigation budgets.
4. **Charts and Visualizations**:
   - Value at Risk curve showing how VaR changes with confidence level.
   - Impact breakdown pie chart showing the distribution of economic impacts.
   - Mitigation analysis chart with dual y-axes for impact and ROI.
5. **Key Metrics**: four key metrics panels are displayed:
   - Value at Risk (VaR) with confidence level and return period.
   - Estimated casualties with mitigation effects.
   - Land impact in hectares and monetary value.
   - Mitigation ROI with percentage risk reduction.

Additional Features

The Python version offers both:

- Interactive web visualizations using Plotly (within the Dash app).

- An optional static report generator using Matplotlib for creating exportable reports.

The additional standalone function `generate_landslide_risk_report()` produces static reports using Matplotlib. This gives users flexibility to generate reports programmatically without running the full web dashboard.

How to Use This Implementation

- First, install the required packages:

  ```
  conda install dash pandas numpy plotly matplotlib seaborn jupyter
  ```

- **Interactive Dashboard**: Run the main class to launch a web-based dashboard

  ```
  dashboard = LandslideRiskDashboard()
  dashboard.run_server()
  ```

- **Static Reports**: Generate exportable reports without running the web server

  ```
  fig = generate_landslide_risk_report('Region C', confidence_level=90)
  plt.savefig('landslide_risk_report.png')
  ```

Streamlit Version

This version uses Streamlit, which is one of the most popular frameworks for creating interactive data applications in Python.

In this version we have:

1. **Interactive Controls**: interactive sliders and selectors in Streamlit's sidebar.
   - Confidence level slider (50-99%) .
   - Region selection dropdown.
   - Time horizon slider (1-50 years).
   - Mitigation budget slider (0-10M).
2. **Visualization**: use Plotly (Python) for creating interactive charts:
   - Value at Risk curve showing how VaR changes with confidence level.
   - Impact breakdown pie chart showing the distribution of economic impacts.
   - Mitigation analysis chart with dual y-axes for impact and ROI.
3. **Layout Adaptation**: Organized the dashboard into a logical flow with sections:
   - Key metrics at the top.
   - Analysis charts in the middle.
   - Region details toward the bottom.

To run this Python dashboard:

1. First, install the required packages:

   ```
   conda install streamlit pandas numpy plotly
   ```

2. Save the code to a file named `landslide_dashboard.py`

3. Run the dashboard with:

   `streamlit run landslide_dashboard.py`

This will open the dashboard in your default web browser, with all functionality of the notebook-based version.