

Manual Técnico e Acadêmico de Uso dos Scripts de Análise Estatística

1. Introdução

Os scripts fornecidos foram desenvolvidos para investigar o comportamento de sistemas determinísticos baseados em *hashes criptográficos* (HMAC-SHA256). Eles simulam a geração de multiplicadores e analisam estatisticamente padrões como sequências consecutivas abaixo de um valor-alvo, variações percentuais acumuladas e indicadores financeiros, como médias móveis e o índice de força relativa (RSI).

Esses estudos têm aplicações em campos como:

- **Teoria dos Jogos:** Avaliação de sistemas baseados em seeds criptográficas.
- **Estatística aplicada:** Identificação de distribuições e padrões probabilísticos.
- **Sistemas determinísticos:** Exploração da previsibilidade de valores a partir de hashes públicos.

Os scripts podem ser utilizados como base para experimentos acadêmicos e atividades práticas, permitindo a exploração de métodos computacionais avançados em estatística.

2. Estrutura Geral dos Scripts

2.1. Funções principais

Todos os scripts utilizam três funções fundamentais:

1. **createCrashMulti(hash)**

Converte o valor de um *hash* hexadecimal em um multiplicador. A lógica é baseada em:

$$\text{multiplicador} = \left(\frac{4294967296}{\text{hash_decimal} + 1} \right) \times 0.99$$

Onde:

- 429496729642949672964294967296 representa $2^{32} \times 2^{32}$, um limite comum para números inteiros.
- O fator 0.990.990.99 reduz o multiplicador como uma forma de ajustar o sistema.

2. **get_next_seed(seed)**

Calcula o próximo valor da seed utilizando o algoritmo **SHA-256**, garantindo que cada valor seja determinístico, mas impossível de prever sem conhecer a seed original.

3. `compute_hmac(seed, message)`

Gera um HMAC-SHA256 usando uma seed e uma mensagem fixa, representando a base de muitos sistemas criptográficos.

2.2. Arquivos e suas características

- **`scj.py`**

Realiza a análise básica, gerando históricos de multiplicadores e identificando a maior sequência de valores consecutivos abaixo de um alvo.

Aplicação acadêmica:

- Cálculo de distribuições estatísticas básicas.
- Análise sequencial.

- **`scjs.py`**

Extensão do `scj.py`, com execução paralela de múltiplos jogos utilizando

`ThreadPoolExecutor`.

Aplicação acadêmica:

- Simulações independentes paralelizadas.
- Análise de convergência de distribuições em sistemas paralelos.

- **`scjv2.py`**

Adiciona relatórios detalhados que incluem distribuições de sequências de perdas, número total de ocorrências, e momento das maiores sequências.

Aplicação acadêmica:

- Estatística descritiva aplicada.
- Geração de relatórios acadêmicos.

- **`scjv4.py`**

Introduz ferramentas gráficas interativas, como gráficos de médias móveis e RSI, usando **`Bokeh`.**

Aplicação acadêmica:

- Visualização de dados estatísticos.
- Interpretação de indicadores financeiros no contexto estatístico.

- **`scjv8.py`**

Inclui uma interface gráfica para monitoramento em tempo real com **`Tkinter`**, permitindo análise interativa.

Aplicação acadêmica:

- Análise dinâmica de eventos.
 - Experimentação com visualização em tempo real.
-

3. Fundamentos Matemáticos e Estatísticos

3.1. Multiplicadores e Distribuições

Os multiplicadores são derivados de um processo pseudoaleatório, mas determinístico, baseado em seeds e HMACs. Isso garante:

- **Imprevisibilidade teórica:** A saída de um hash criptográfico é caoticamente distribuída.
- **Determinismo:** Dada a mesma seed e mensagem, os valores gerados serão idênticos.

Distribuição esperada dos multiplicadores

A fórmula utilizada gera uma distribuição que tende a se concentrar em valores próximos de 1. Estatisticamente, isso pode ser analisado para identificar desvios do comportamento esperado.

3.2. Sequências consecutivas

A análise de sequências consecutivas abaixo de um alvo pode ser conectada à **teoria de ruínas de jogadores** na estatística:

- **Probabilidade condicional:** Qual é a chance de um multiplicador $M < 2$, dado $M_{anterior} < 2$?
- **Máximos locais:** Identificação do maior número consecutivo de eventos desfavoráveis.

3.3. Indicadores Avançados

Os scripts avançados (v4 e v8) incorporam:

1. **Média móvel e desvio padrão:** Indicadores clássicos usados em estatística e análise técnica.

$$\text{média_móvel} = \frac{\sum_{i=n-j}^n x_i}{j}$$

2. **RSI (Índice de Força Relativa):** Mede a força das variações em relação a ganhos e perdas acumulados: $RSI = 100 - \frac{100}{1 + RS}$, $RS = \frac{\text{média_ganhos}}{\text{média_perdas}}$
 $RSI = 100 - \frac{100}{1 + RS}$, $RS = \frac{\text{média_perdas}}{\text{média_ganhos}}$

$$RSI = 100 - \frac{100}{1 + RS}, \quad RS = \frac{\text{média_ganhos}}{\text{média_perdas}}$$

4. Customização

4.1. Parâmetros ajustáveis

Todos os scripts permitem a modificação de variáveis-chave:

- **seed**: Base da geração determinística.
Recomendação: Utilize seeds fornecidas pelas plataformas ou derive suas próprias para experimentos controlados.
- **hist length**: Define o tamanho do histórico de multiplicadores.
Impacto: Valores maiores aumentam o tempo de execução e a precisão estatística.
- **target**: Multiplicador-alvo.
Exemplo de estudo: Identificar o comportamento para diferentes alvos (1.5x, 2x, 5x1.5x, 2x, 5x1.5x, 2x, 5x).
- **Parâmetros gráficos (em v4 e v8)**:
Permitem ajustar a sensibilidade dos indicadores.

4.2. Ajustes de desempenho

Scripts com execução paralela (ex.: `scjs.py`) permitem modificar:

- **num_jogos**: Número de jogos rodados simultaneamente.
 - Valor maior exige mais processamento.
-

5. Métodos de Uso

5.1. Execução básica

1. **Instale as dependências necessárias:**

```
bash
CopiarEditar
pip install numpy pandas tqdm bokeh
```

2. **Rode o script desejado:**

```
bash
CopiarEditar
python scjv4.py
```

5.2. Relatórios gerados

Scripts como `scjv2.py` geram relatórios em `.txt` contendo:

- Sequências de perdas abaixo do alvo.
 - Estatísticas descritivas.
-

6. Estudos e Expansões Acadêmicas

Esses scripts podem ser expandidos para estudos em:

1. **Simulação estocástica:** Estendendo as seeds com algoritmos diferentes.
2. **Análise de séries temporais:** Aplicando métodos como ARIMA ou SARIMA nos dados gerados.
3. **Criptografia aplicada:** Estudo das propriedades matemáticas do HMAC-SHA256.