**Relatório Sistema Multi Agentes**

Modelo de Disseminação de Opiniões em Redes Sociais

Nicolly Lira de Albuquerque

Victor Ferreira

**Visão geral**

O modelo é implementado em Python usando as bibliotecas agentpy e networkx. O objetivo é analisar como diferentes tipos de agentes e suas interações influenciam a formação e evolução das opiniões ao longo do tempo. A visualização dos resultados é realizada com a biblioteca plotly.

**Descrição**

**Tipos de Agentes**

Os agentes são categorizados em diferentes tipos, cada um com uma regra específica para atualizar sua opinião:

1. **Common (Comum)**: Atualiza sua opinião pela média das opiniões recebidas.
2. **Influencer (Influenciador)**: Adota a opinião mais comum entre as recebidas.
3. **Extremist (Extremista)**: Não muda de opinião.
4. **Moderate (Moderado)**: Atualiza sua opinião pela média das opiniões recebidas, ajustada por uma distribuição normal.
5. **Conformist (Conformista)**: Adota a mediana das opiniões recebidas.
6. **Rebel (Rebelde)**: Adota o valor negativo da média das opiniões recebidas.
7. **OpinionLeader (Líder de Opinião)**: Escolhe aleatoriamente uma das opiniões recebidas.
8. **Sporadic (Esporádico)**: Adota uma opinião recebida com uma probabilidade de 80%.

**Envio de Mensagens**

O modelo utiliza o protocolo FIPA-ACLna classe FipaAclMessage para comunicação entre agentes. Cada mensagem possui os seguintes atributos:

* **performative**: Tipo da mensagem (ex: "request", "inform").
* **sender**: ID do agente que envia a mensagem.
* **receiver**: ID do agente que recebe a mensagem.
* **content**: Conteúdo da mensagem.
* **conversation\_id**: Identificador opcional da conversação.

**Rede de Interação**

A rede de interação entre os agentes é definida pelo parâmetro network\_type, que pode ser:

* **random**: Rede aleatória.
* **scale\_free**: Rede livre de escala.
* **small\_world**: Rede de mundo pequeno.

**Parâmetros do Modelo**

Os principais parâmetros do modelo são:

* **initial\_opinion**: Opinião inicial dos agentes.
* **influence\_factor**: Fator de influência.
* **num\_agents**: Número de agentes.
* **frações de diferentes tipos de agentes**: Especifica a proporção de cada tipo de agente na população.

**Execução do Modelo**

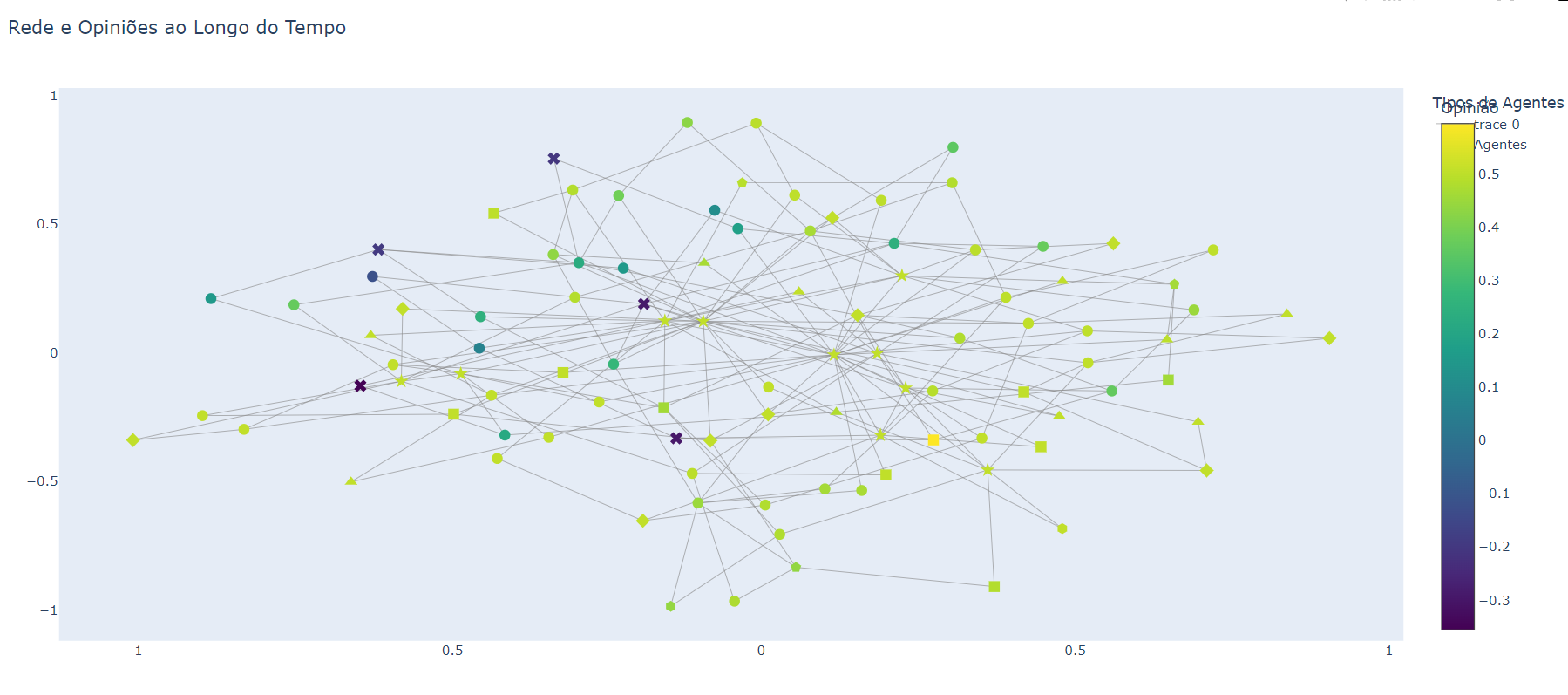
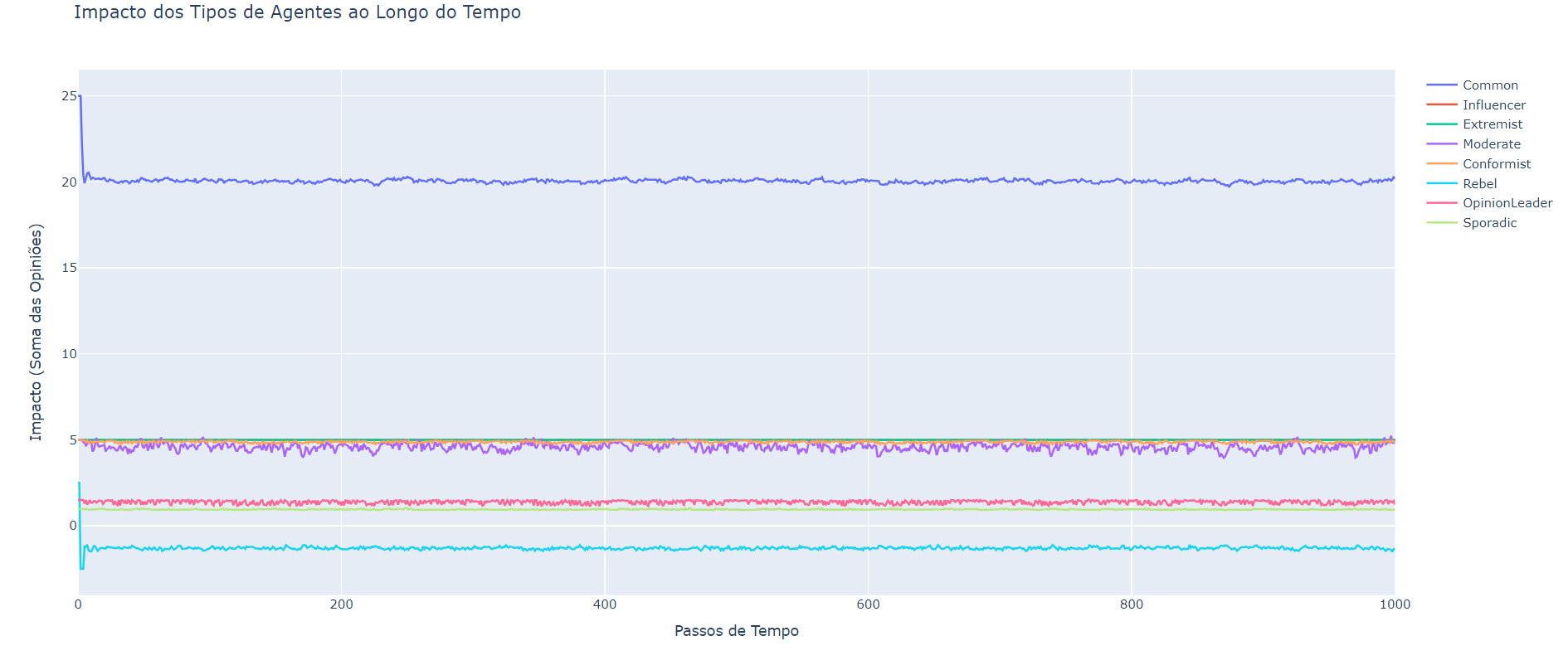
O modelo é executado por meio da classe OpinionModel, que cria a rede, inicializa os agentes e executa a simulação por um número definido de passos de tempo.

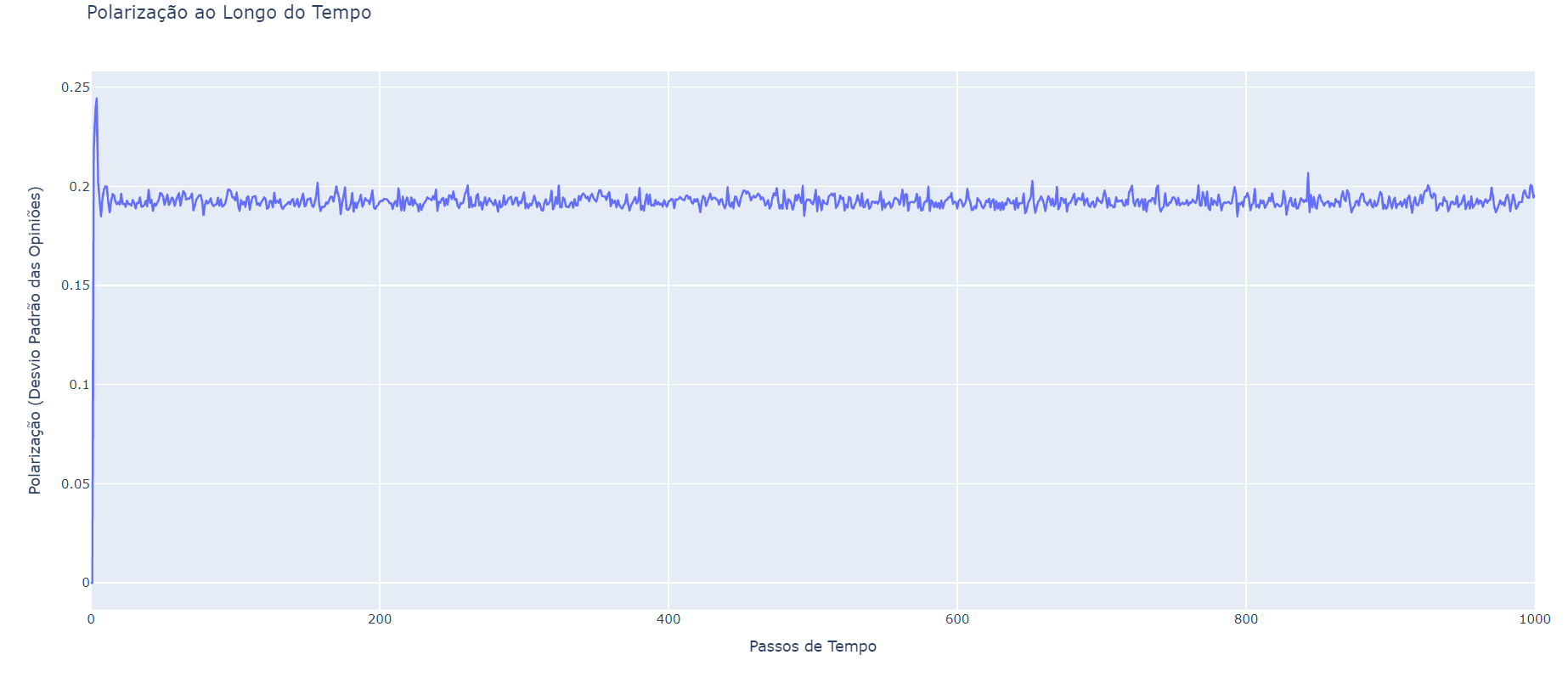
1. **setup()**: Cria a rede e inicializa os agentes.
2. **step()**: Realiza a interação entre os agentes e atualiza suas opiniões.
3. **update()**: Registra os dados da simulação.
4. **end()**: Relata as opiniões finais dos agentes.

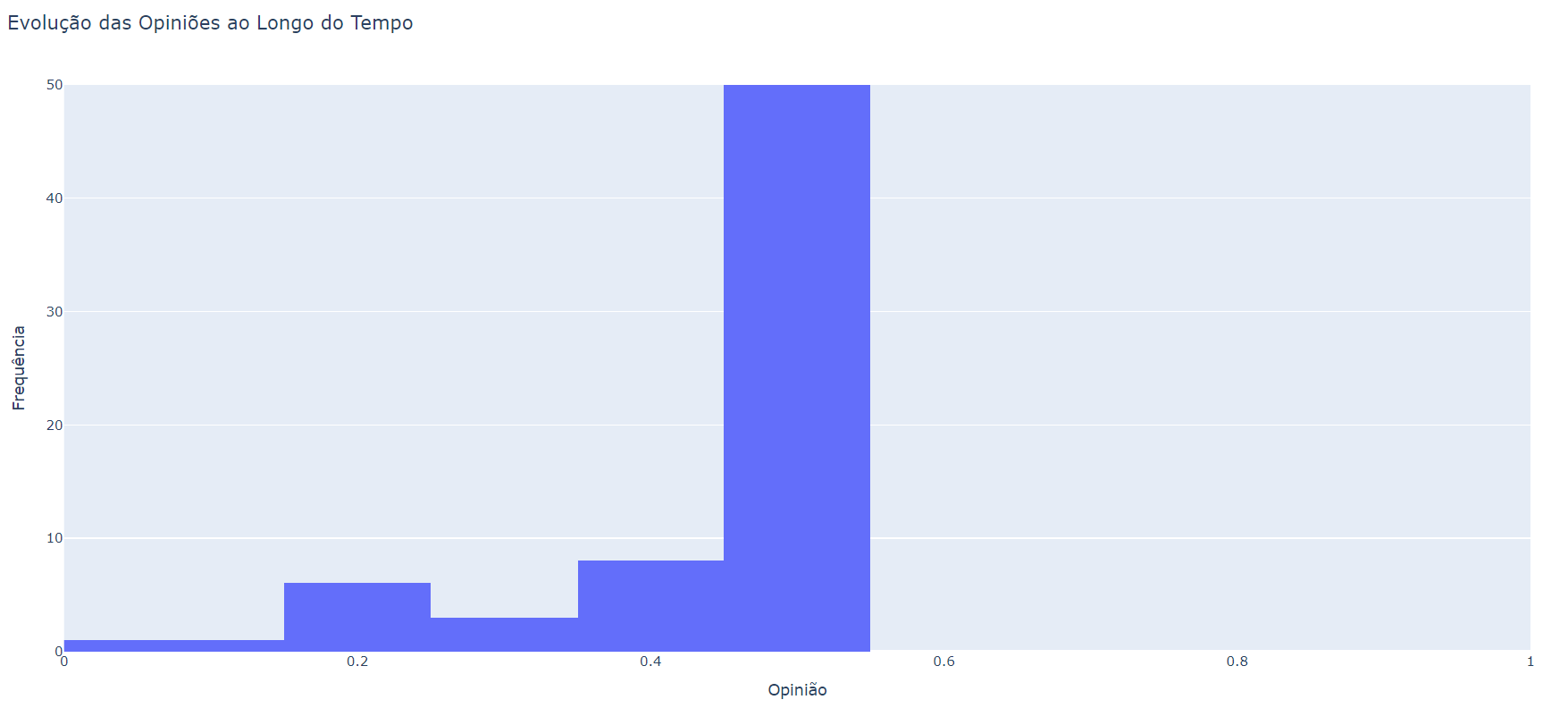
**Resultados**

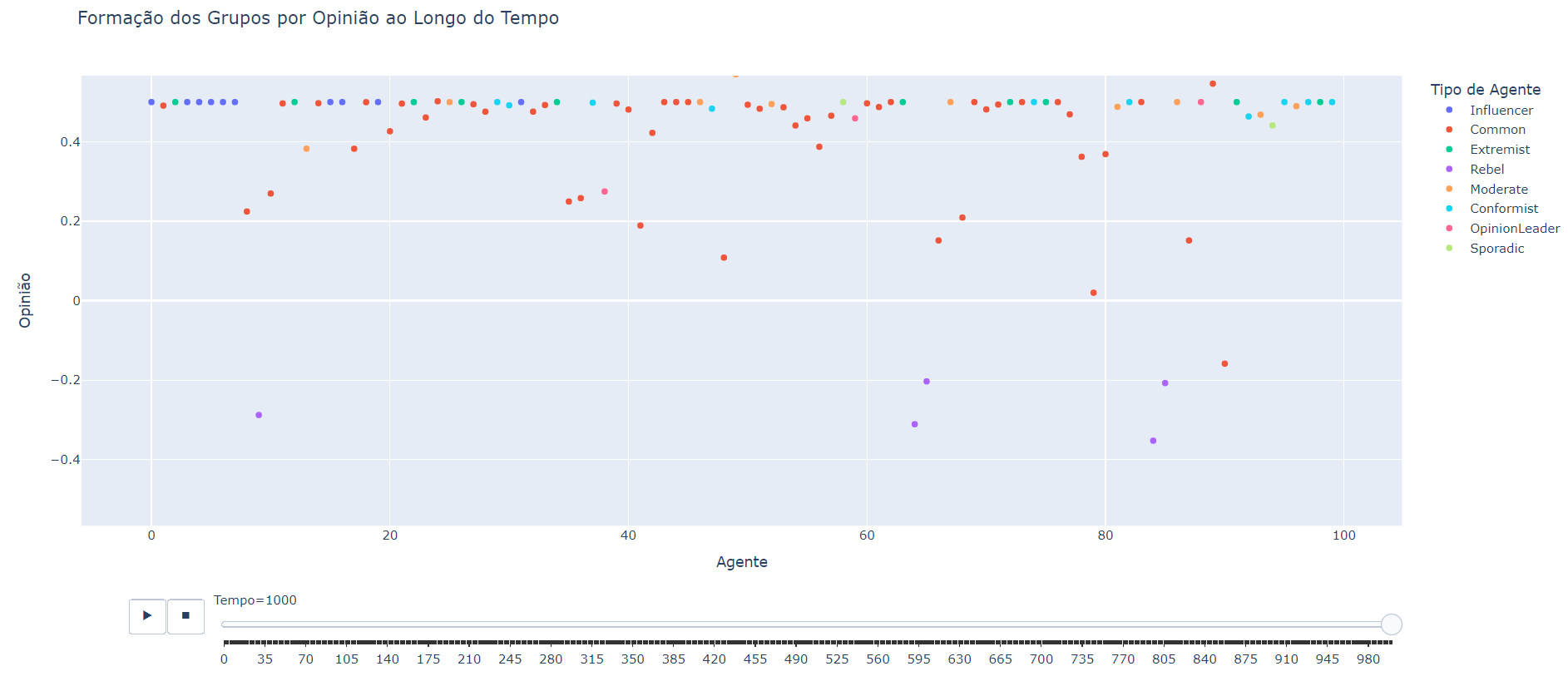
Os resultados da simulação indicam que, independentemente da topologia da rede, os agentes apresentaram uma forte resistência à mudança de opinião. Os gráficos abaixo estão ordenados da seguinte forma para cada topologia de rede:

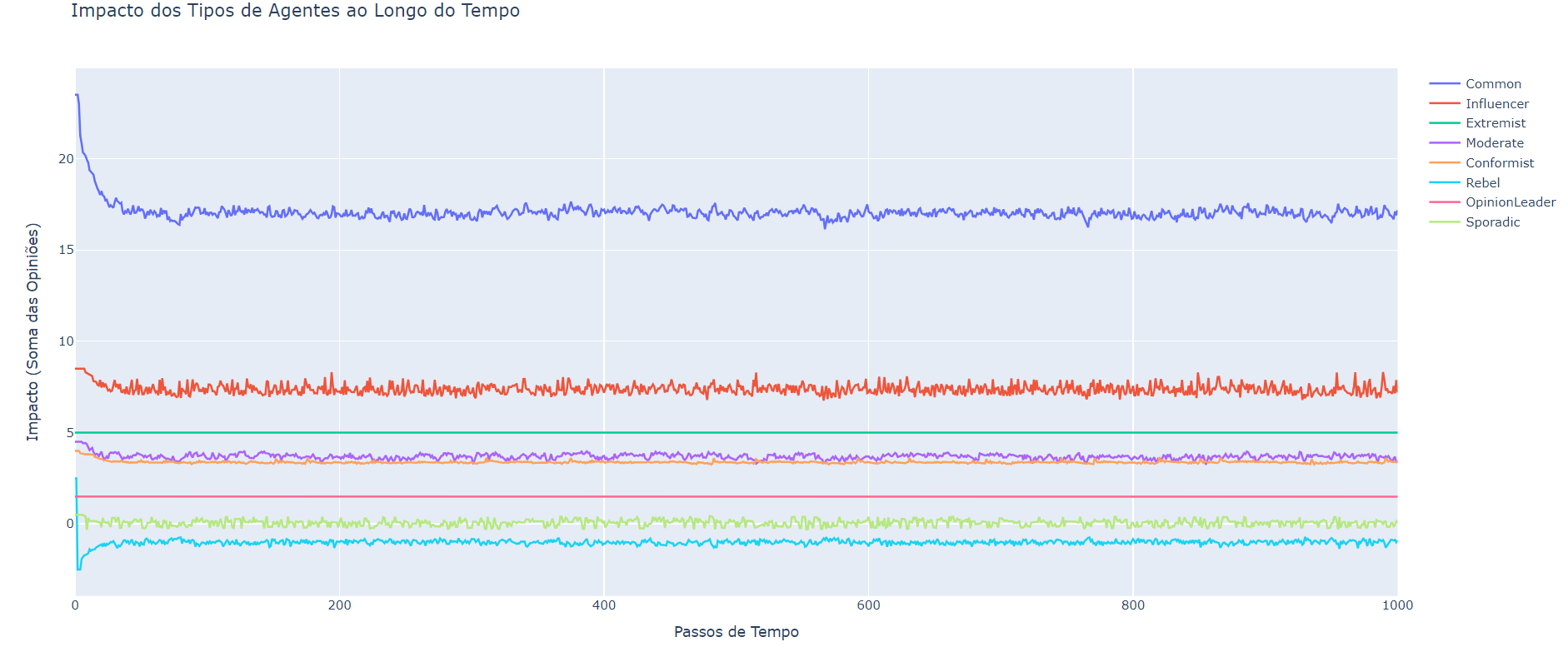
1. **Impacto dos tipos de agente ao longo do tempo:** Impacto dos agentes na mudança de opinião. Eixo x são os steps e no eixo y é a soma das opiniões de agentes de um determinado tipo.
2. **Rede e opiniões ao longo do tempo:** Apresenta a topologia de rede. Os agentes são dispostos na rede e alteram a cor de acordo com a mudança de opinião ao longo do tempo.
3. **Polarização ao longo do tempo:** Analisa a polaridade pelo desvio padrão. Eixo Y é o desvio padrão e eixo x os steps.
4. **Formação dos grupos por opinião ao longo do tempo:** Visualiza a posição dos vários tipos de agente ao longo do tempo. Eixo x são os steps e eixo y a opinião.
5. **Evolução das opiniões ao longo do tempo:** Apresenta como as opiniões se comportam ao longo do tempo. Eixo x - opinião e eixo y - quantidade de agentes.

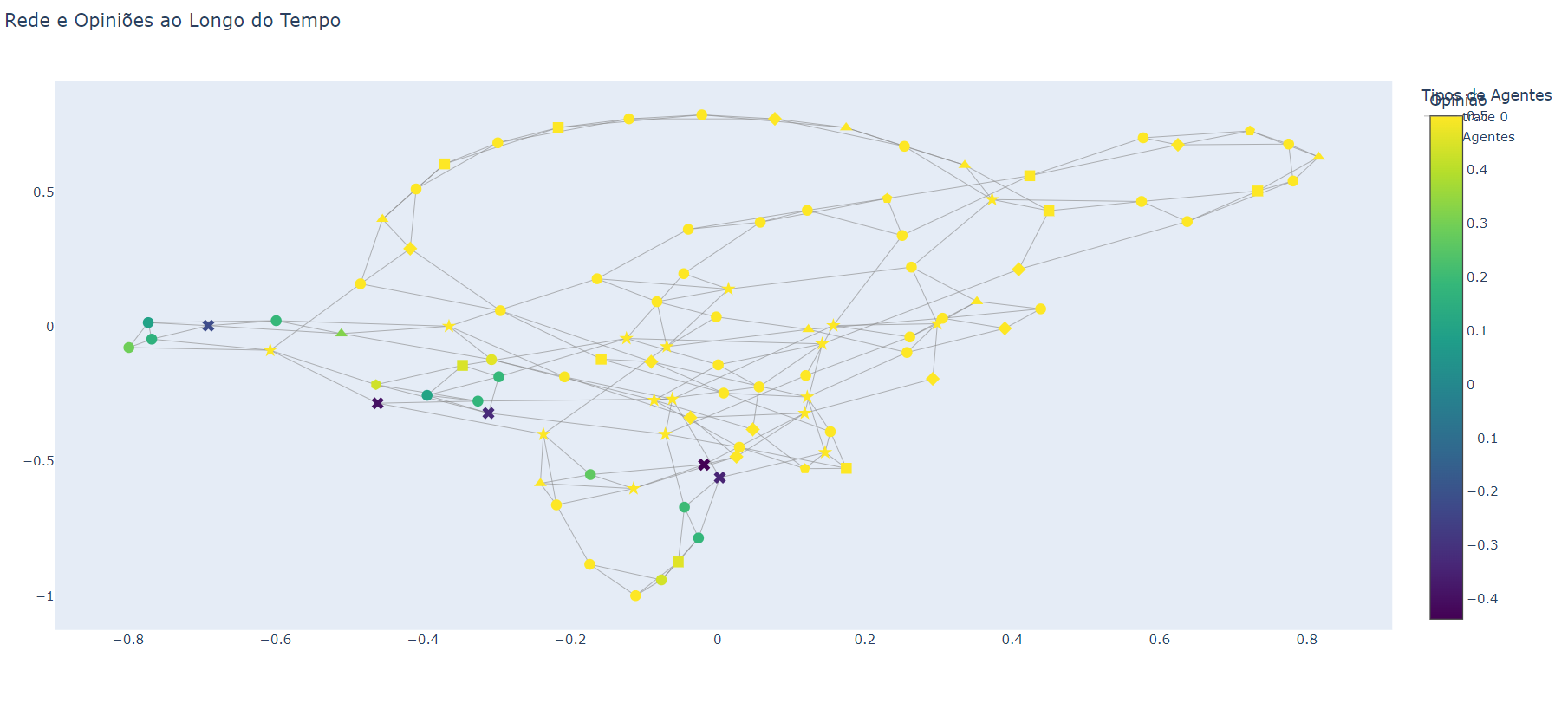
**Scale Free**

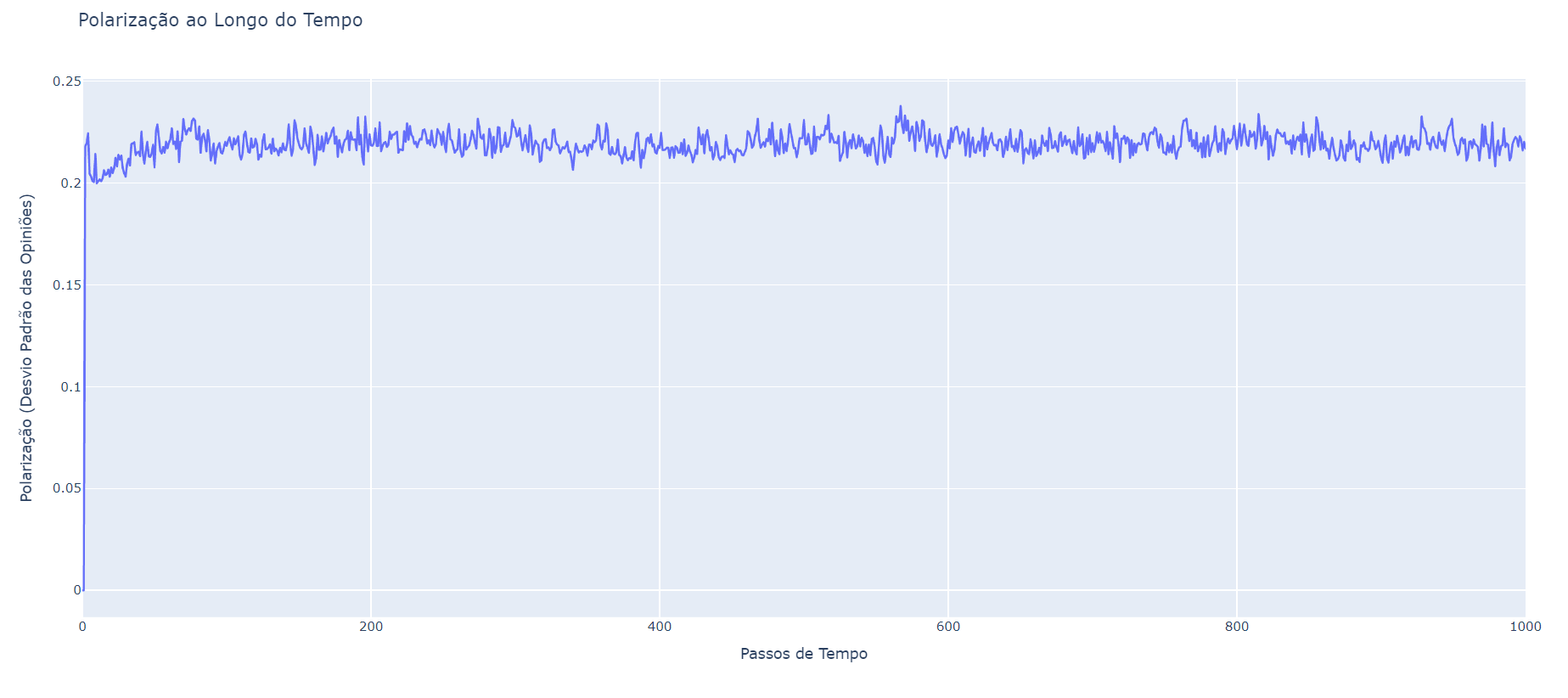
****

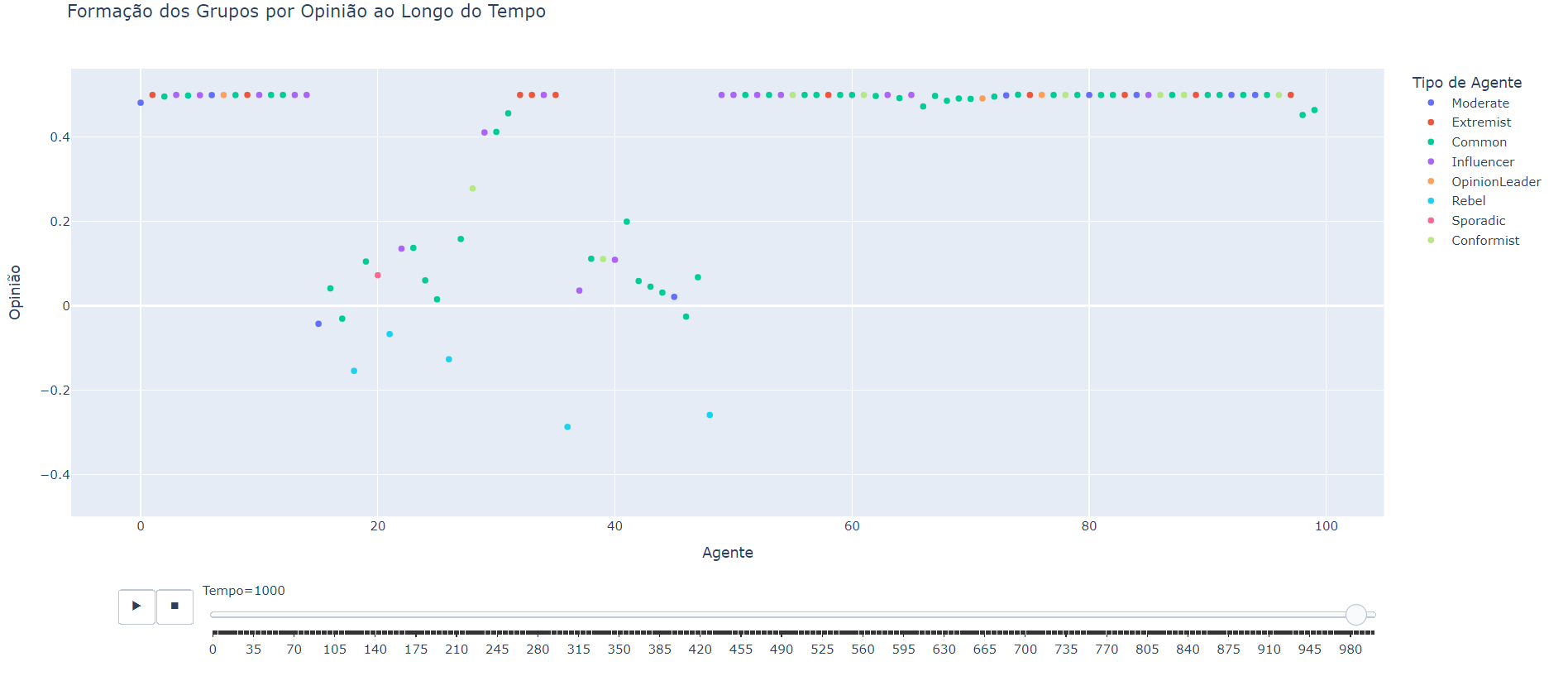
****

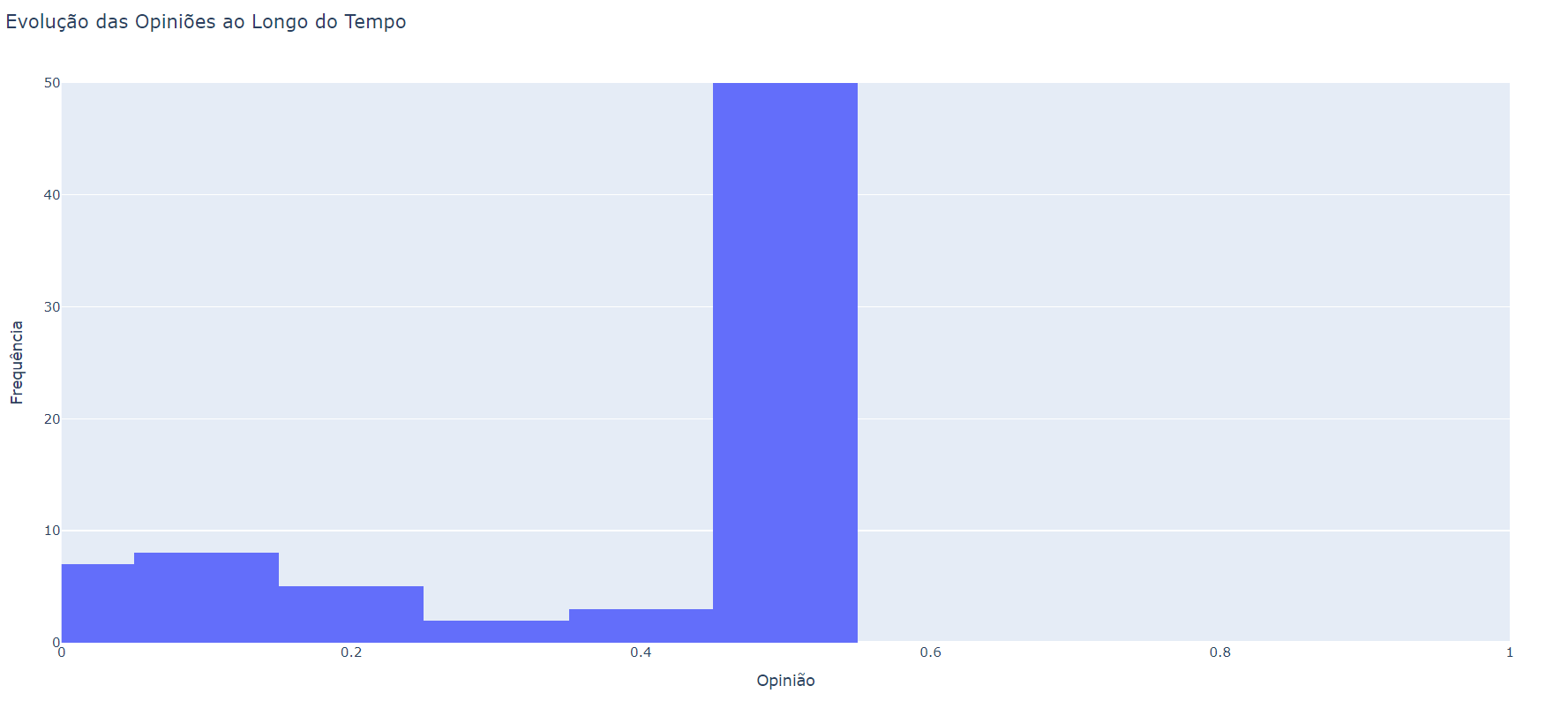
****

**Small World**

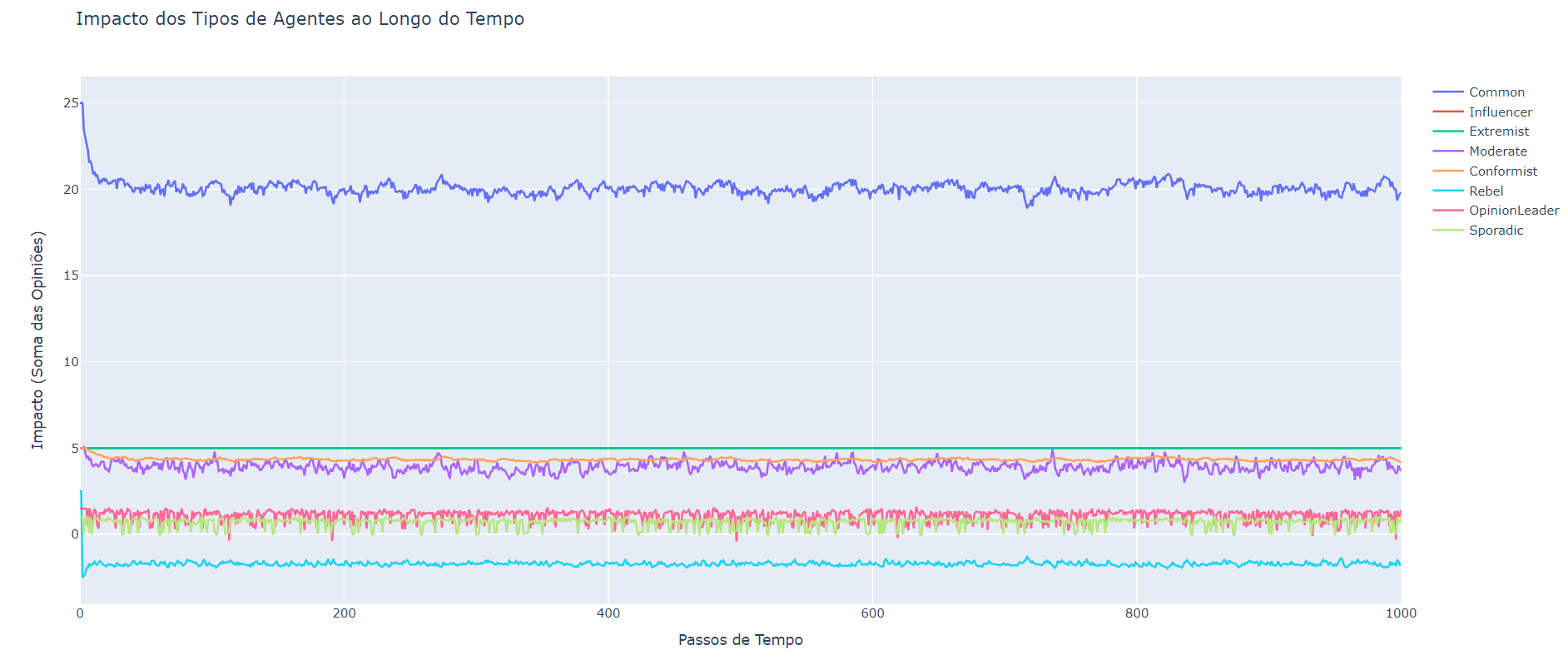
****

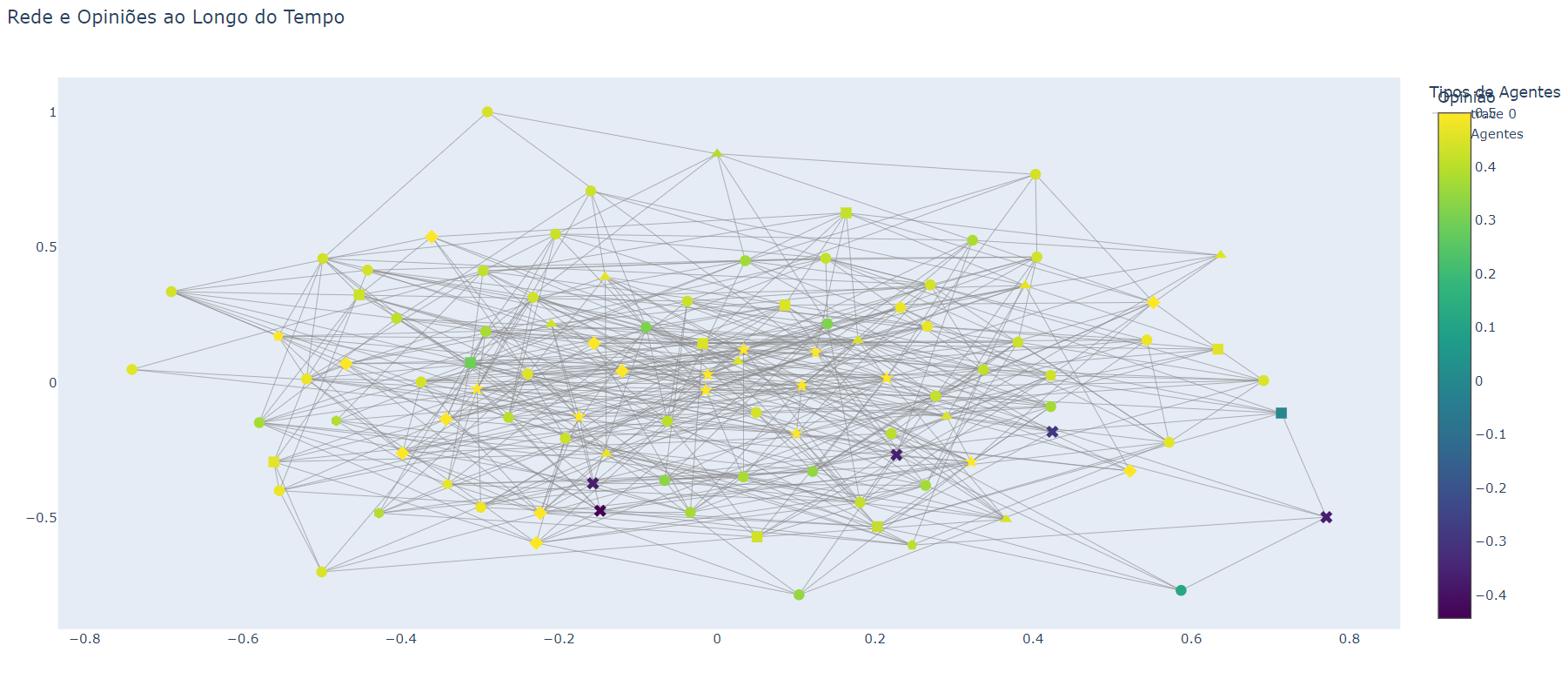
****

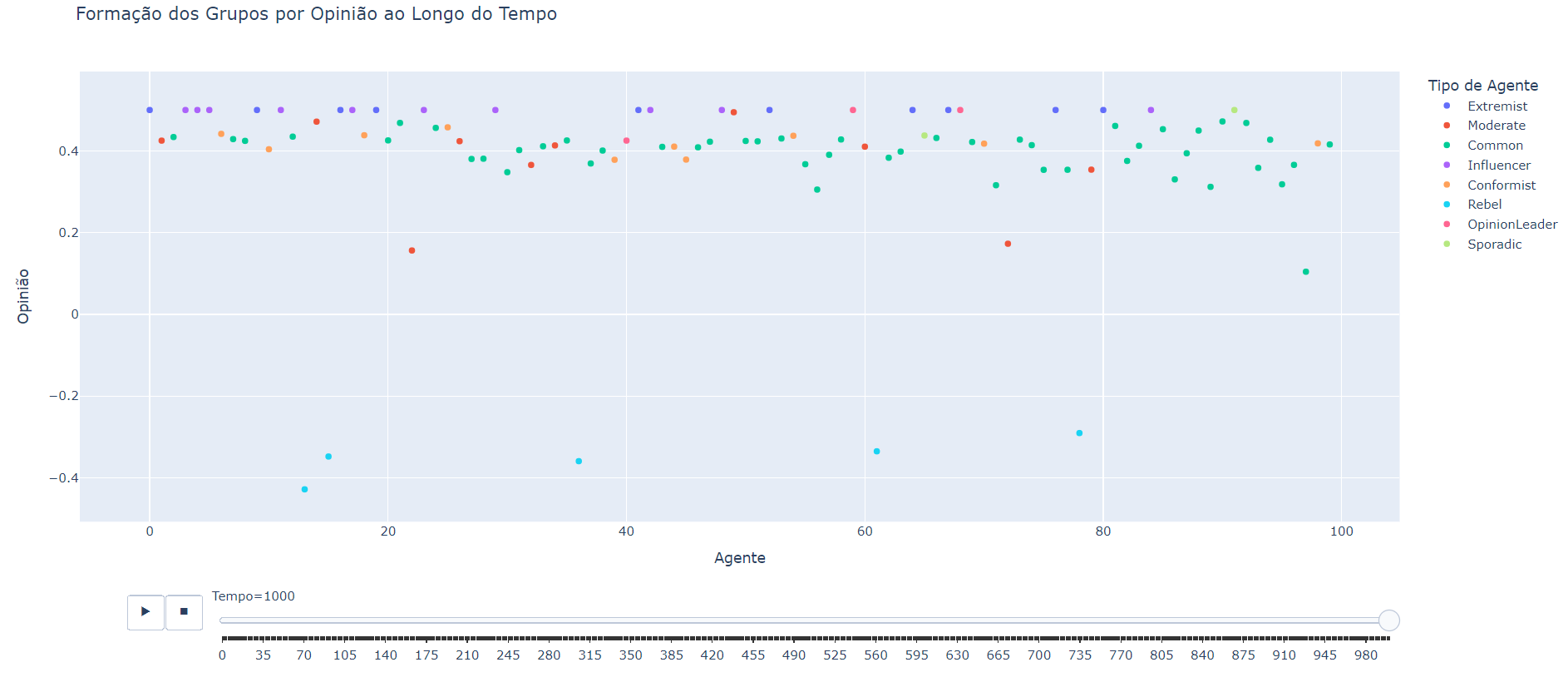
****

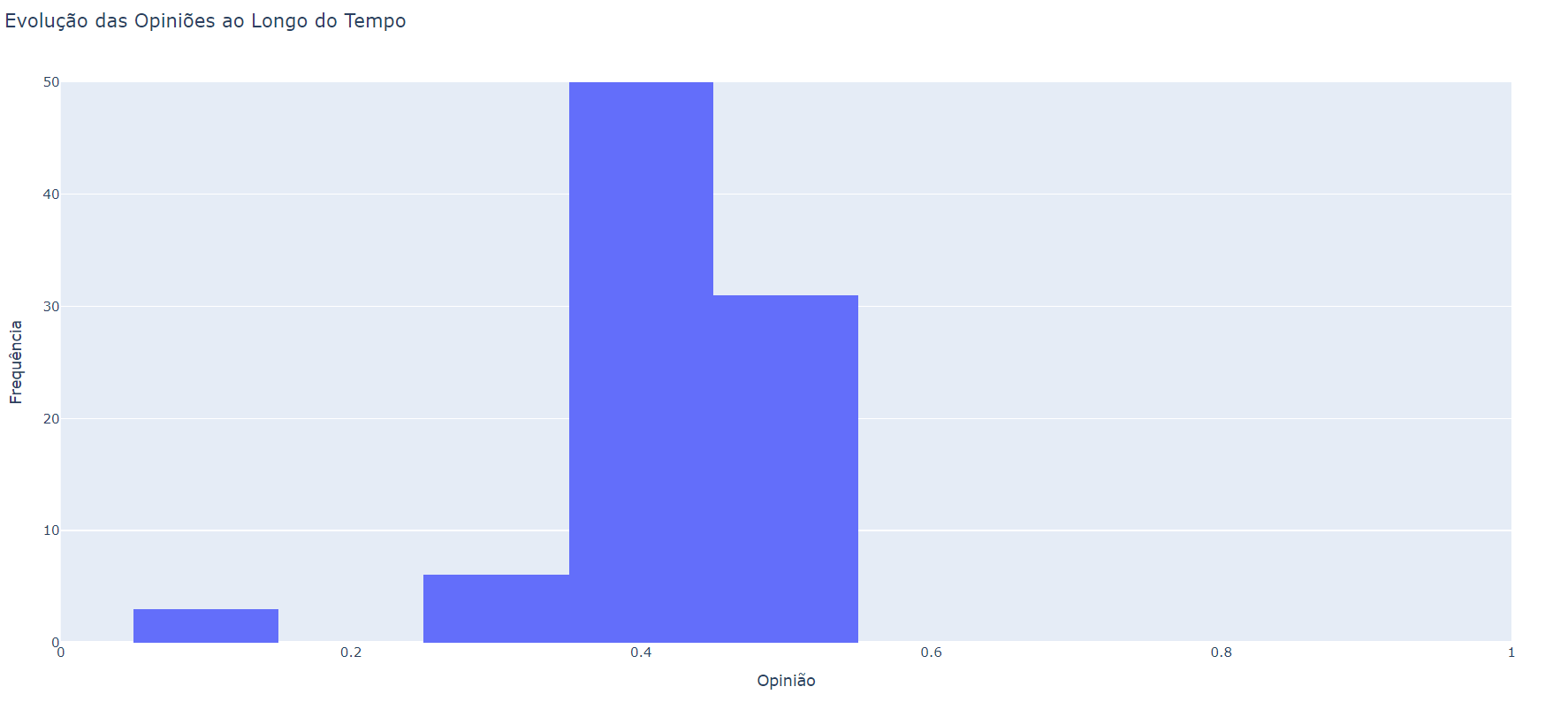
****

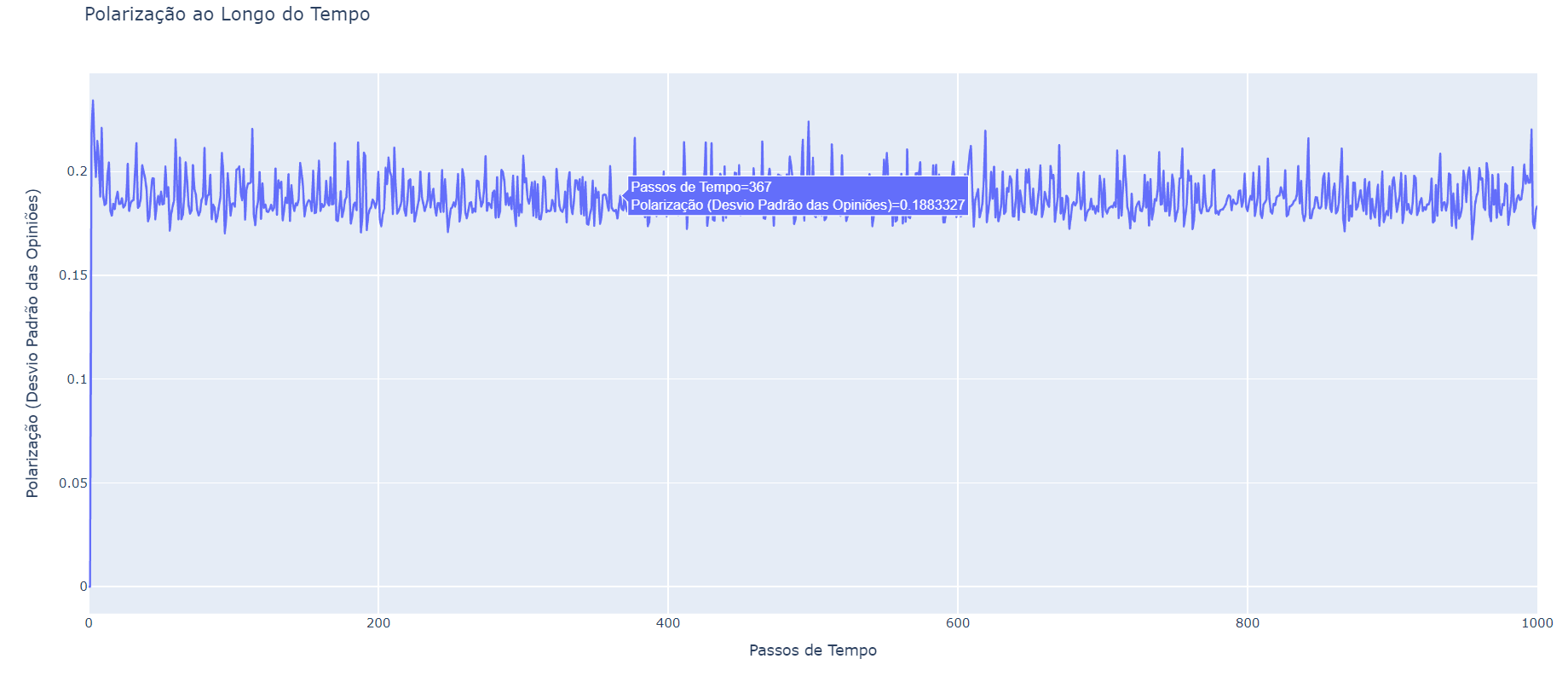
**Randon**

****

****

****

****

****