

Simulando pesquisas eleitorais com Python

```
In [212]: import math
import random
import matplotlib.pyplot as plt
%matplotlib inline
```

Primeiro, definimos o tamanho do eleitorado e o número de votos para os quais vamos simular e definimos a função 'pesquisa'

```
In [313]: eleitorado = 150000000
votacao = 60000000
def pesquisa(eleitorado, votacao, amostra):
    resultado = 0
    for i in range(amostra):
        chave = random.randint(0,eleitorado)
        if chave < votacao:
            resultado += 1
        else:
            resultado += 0
    erro = (((resultado/amostra) * eleitorado) - votacao)/eleitorado*100
    return (resultado/amostra) * eleitorado
```

Agora vamos simular uma pesquisa:

```
In [316]: amostra = 100
resultado_pesquisa = pesquisa(eleitorado, votacao, amostra)
erro = ((resultado_pesquisa - votacao)/eleitorado)*100

print("resultado: {}, erro:{}".format(resultado_pesquisa, erro))

resultado: 51000000.0, erro:-6.0
```

Depois, definimos o número de simulações de pesquisa que queremos executar e definimos a função 'Simulacao'

Essa função retorna o resultado médio e o erro médio depois de realizadas 'simulacoes' pesquisas.

```
In [249]: simulacoes = 2000
amostra = 2000
def simulacao(simulacoes, amostra):
    soma_resultado = 0
    soma_erro = 0
    for i in range(simulacoes):
        resultado = pesquisa(eleitorado, votacao, amostra)
        soma_resultado += resultado
        soma_erro += resultado / eleitorado
    erro_medio = soma_erro / simulacoes
    resultado_medio = soma_resultado/ simulacoes
    return resultado_medio, erro_medio
```

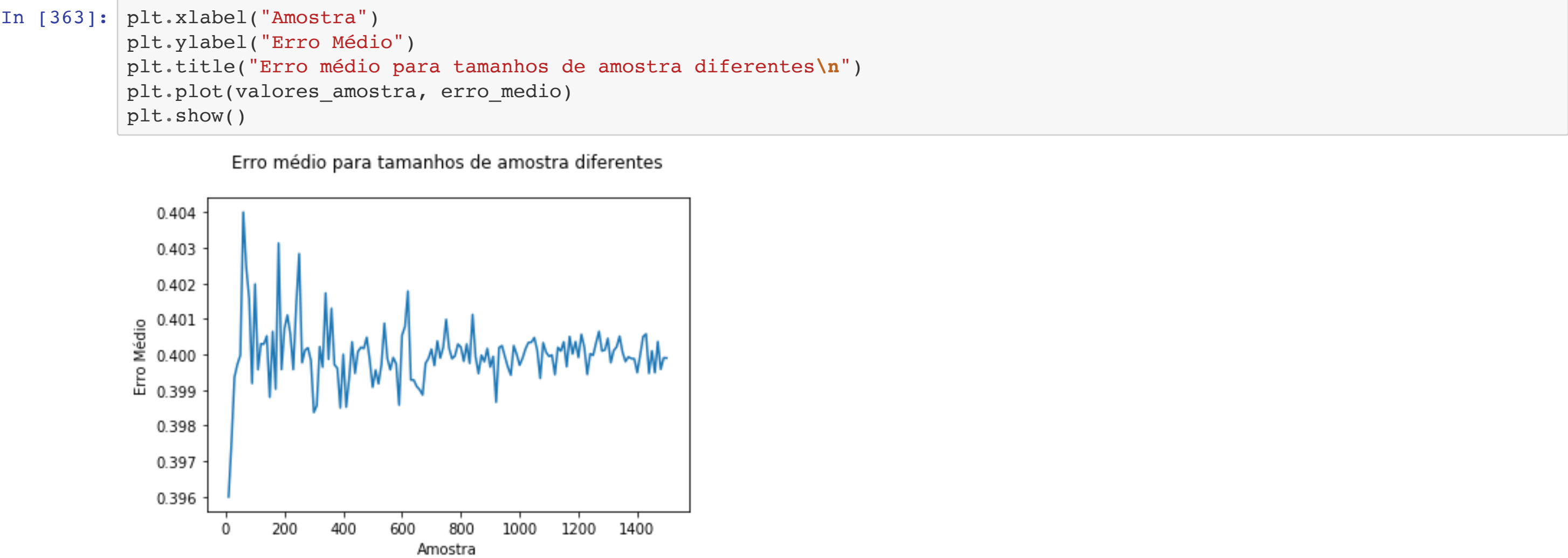
Agora vamos visualizar a diferença do erro médio para tamanhos diferentes de amostra

O código abaixo faz 150 simulações em diferentes tamanhos de amostra. A amostra varia de 10 em 10, partindo de 10 a 1500. No total, são feitas 150.000 'pesquisas'.

```
In [362]: valores_amostra = []
resultado_medio = []
erro_medio = []
for i in range(1,151):
    resultado_medio.append(0)
    erro_medio.append(0)
for i in range(1,151):
    valores_amostra.append(i*10)
for i in range(0,150):
    resultado_medio[i], erro_medio[i] = simulacao(1000, valores_amostra[i])
print("Finalizado")

Finalizado
```

Vizualizando a variação do erro quando se varia as amostras.



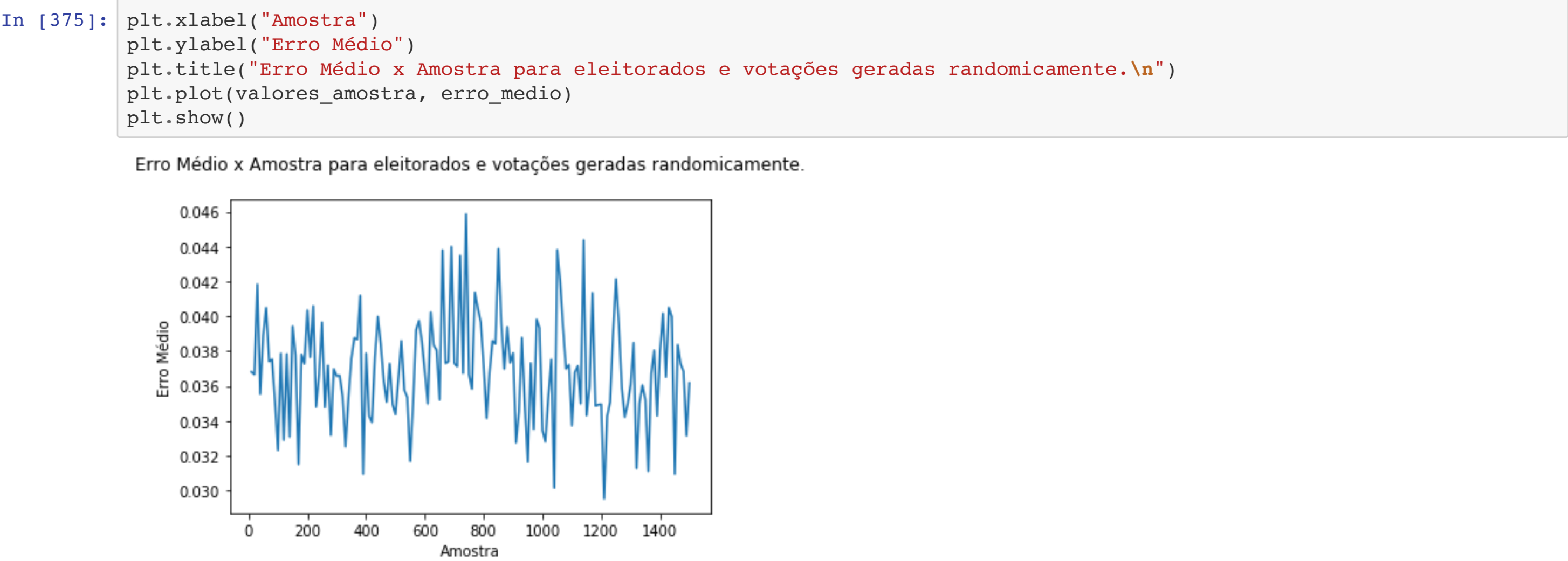
Agora, algo também interessante. Podemos variar aleatoriamente a votação e o eleitorado para cada uma das 150.000 pesquisas. Nessa função estamos apenas interessados no erro.

```
In [372]: def simulacao_aleatoria(simulacoes, amostra):
    soma_resultado = 0
    soma_erro = 0
    for i in range(simulacoes):
        eleitorado = random.randint(0,100000000000) #De 0 a 10 Bilhões
        votação = random.randint(0,eleitorado)
        resultado = pesquisa(eleitorado, votacao, amostra)
        soma_erro += resultado / eleitorado
    erro_medio = soma_erro / simulacoes
    return erro_medio

In [373]: valores_amostra = []
erro_medio = []
for i in range(1,151):
    erro_medio.append(0)
for i in range(1,151):
    valores_amostra.append(i*10)
for i in range(0,150):
    erro_medio[i] = simulacao_aleatoria(1000, valores_amostra[i])
print("Finalizado")

Finalizado
```

Agora, vamos vizualizar Erro Médio x Amostra para eleitorados e votações geradas aleatoriamente.



Podemos concluir, então, que mesmo com eleitorados e votações completamente aleatórias o erro não excede 4.6 pontos percentuais para amostras menores que 1500.