

```
In [5]: import yfinance as yf
import pandas as pd
import numpy as np
```

```
In [6]: # Coletando dados do IBOV do Yahoo Finance
ibov_yf = yf.download('^BVSP', start='2020-01-01', end='2024-05-01')

# Criando a coluna de resultado percentual diário
ibov_yf['resultado'] = ibov_yf['Close'].pct_change() * 100
ibov_yf.dropna(inplace=True) # Remover linhas com valores NaN após a operação de pct_change

# Adicionando colunas auxiliares para a análise usando .loc corretamente
ibov_yf.loc[:, 'Ano'] = ibov_yf.index.year
ibov_yf.loc[:, 'Mes_numerico'] = ibov_yf.index.month
ibov_yf.loc[:, 'dia_semana'] = ibov_yf.index.strftime('%A')
ibov_yf.loc[:, 'result_binario'] = np.where(ibov_yf['resultado'] > 0, 'positivo', 'negativo')

ibov_yf
```

[*****100%*****] 1 of 1 completed

Out[6]:

	Open	High	Low	Close	Adj Close	Volume	resultado	Ano	Mes_numerico	dia_semana	result_binario
Date											
2020-01-03	118564.0	118792.0	117341.0	117707.0	117707.0	6834500	-0.730352	2020	1	Friday	negativo
2020-01-06	117707.0	117707.0	116269.0	116878.0	116878.0	6570000	-0.704291	2020	1	Monday	negativo
2020-01-07	116872.0	117076.0	115965.0	116662.0	116662.0	4854100	-0.184808	2020	1	Tuesday	negativo
2020-01-08	116667.0	117335.0	115693.0	116247.0	116247.0	5910500	-0.355729	2020	1	Wednesday	negativo
2020-01-09	116248.0	116820.0	115411.0	115947.0	115947.0	5953500	-0.258071	2020	1	Thursday	negativo
...
2024-04-24	125149.0	125473.0	124556.0	124741.0	124741.0	10526300	-0.325215	2024	4	Wednesday	negativo
2024-04-25	124718.0	124732.0	123703.0	124646.0	124646.0	10093100	-0.076158	2024	4	Thursday	negativo
2024-04-26	124651.0	126826.0	124651.0	126526.0	126526.0	8900000	1.508271	2024	4	Friday	positivo
2024-04-29	126527.0	127352.0	126467.0	127352.0	127352.0	8385100	0.652830	2024	4	Monday	positivo
2024-04-30	127352.0	127352.0	125856.0	125924.0	125924.0	10889600	-1.121302	2024	4	Tuesday	negativo

1075 rows × 11 columns

Calcular o Coeficiente de Variação

```
In [7]: # Calculando o coeficiente de variação
coef_var_resultados = ibov_yf.groupby('result_binario')['resultado'].agg(lambda x: abs((x.std() / x.mean()) * 100))

# Renomeando a coluna
coef_var_resultados.name = 'Coeficiente de Variação'

coef_var_resultados = coef_var_resultados.to_frame()

# Exibindo o resultado
coef_var_resultados
```

Out[7]:

	Coeficiente de Variação
result_binario	
negativo	125.078992
positivo	101.266436

Análise de Retornos Diários Agrupados por Categoria

```
In [8]: retornos_diarios = ibov_yf.groupby('result_binario')['resultado'].mean()

retornos_diarios.name = "Retornos Diários Médios por Categoria"

retornos_diarios = retornos_diarios.to_frame()

retornos_diarios
```

Out[8]:

	Retornos Diários Médios por Categoria
result_binario	
negativo	-1.135700
positivo	1.132426

Cálculo de Estatísticas Descritivas para Diferentes Grupos

```
In [9]: estatisticas_semanais = pd.pivot_table(ibov_yf, index='dia_semana', values='resultado', aggfunc=['mean', 'median', 'std', 'max', 'min'])

print("Estatísticas Descritivas dos Resultados Diários por Dia da Semana:")

estatisticas_semanais
```

Out[9]:

	mean	median	std	max	min
	resultado	resultado	resultado	resultado	resultado
dia_semana					
Friday	-0.058565	-0.013400	1.739112	13.908215	-5.508944
Monday	-0.064873	-0.021522	2.001467	6.521614	-13.921479
Thursday	-0.053200	0.031930	1.662396	3.674156	-14.779679
Tuesday	0.227150	-0.031846	1.507710	9.688532	-3.281539
Wednesday	0.048201	0.097281	1.634142	7.496164	-10.348848

Visualização de Volume Negociado

```
In [10]: volume_mensal = ibov_yf.groupby(['Ano', 'Mes_numerico'])['Volume'].sum()

print("Volume Negociado por Ano e Mês:")

volume_mensal = volume_mensal.to_frame()

volume_mensal.tail()
```

Out[10]:

		Volume
Ano	Mes_numerico	
2023	12	191666500
2024	1	210769000
	2	203808700
	3	198088400
	4	231476400

Identificação de Tendências Sazonais

```
In [11]: tendencias_sazonais = ibov_yf.groupby('Mes_numerico')['resultado'].mean()

tendencias_sazonais.name = "Tendências Sazonais Mensais"

tendencias_sazonais = tendencias_sazonais.to_frame()

tendencias_sazonais
```

Out[11]:

	Tendências Sazonais Mensais
Mes_numerico	
1	-0.009182
2	-0.210727
3	-0.183938
4	0.031324
5	0.256936
6	0.070233
7	0.143206
8	-0.052650
9	-0.117118
10	-0.053488
11	0.288369
12	0.191301

Comparação de Performance Entre Grupos

```
In [12]: performance_anual = ibov_yf.groupby('Ano')['resultado'].mean()

performance_anual.name = "Performance Anual Média"

performance_anual = performance_anual.to_frame()

performance_anual
```

Out[12]:

Performance Anual Média	
Ano	
2020	0.042880
2021	-0.043692
2022	0.028193
2023	0.085945
2024	-0.073571

Resultados Agrupados por Ano e Mês

```
In [13]: resultados_ano_mes = ibov_yf.groupby(['Ano', 'Mes_numerico'])['resultado'].mean()

resultados_ano_mes.name = "Resultados Médios por Ano e Mês"

resultados_ano_mes = resultados_ano_mes.to_frame()

resultados_ano_mes.tail()
```

Out[13]:

Resultados Médios por Ano e Mês		
Ano	Mes_numerico	
2023	12	0.279603
2024	1	-0.220515
	2	0.055937
	3	-0.033441
	4	-0.074957