Distribuições de Probabilidade e os Retornos do Ibovespa: Aplicação em Python

Marcelo Otavio Milani

August 5, 2022

1 Bibliotecas necessárias

```
#Biblioteca para analises e manipulacoes de dados
import pandas as pd
#Biblioteca para extracao de dados do Mercado Financeiro do Yahoo Finance
import yfinance as yf
#Biblioteca para trabalhar com datas
from datetime import date
from datetime import timedelta
#Biblioteca para utilizacao de Distribuicoes de Probabilidade (derivada da Scipy)
from fitter import Fitter, get_common_distributions, get_distributions
#Biblioteca para criar graficos
import matplotlib.pyplot as plt
```

2 Breve fundamentação teórica

Referências utilizadas para a breve fundamentação teórica: [Referência 1], [Referência 2], [Referência 3], [Referência 4], [Referência 5].

2.1 Variáveis aleatórias e Processos estocásticos

Processo estocástico pode ser definido como uma variável que se comportam de maneira aleatória (parcial ou total). Essa sequência de aleatoriedade está associada ao tempo e/ou a ocorrência de determinados eventos.

Um processo estocástico pode ser categorizado como um processo em tempo discreto ou em tempo contínuo. Tempo discreto é composto por variáveis discretas onde os valores de tais variáveis se alteram em um determinado instante de tempo. Já o tempo contínuo é composto de variáveis contínuas onde os valores de tais variáveis se alteram a qualquer instante de tempo.

Alguns exemplos de processos estocásticos:

- Variação da temperatura de uma determinada cidade ao longo do tempo.
- Preço de uma ação listada na B3 (Bolsa de valores do Brasil) ao longo do tempo.
- Quantidade de pessoas que vão ao supermercado por dia.

2.2 Distribuições de Probabilidade

Sabendo da existência de processos estocásticos, as distribuições de probabilidade por sua vez descrevem o comportamento aleatório desses processos. Sendo assim, a distribuição de probabilidade que melhor se ajusta a uma variável aleatória "X" é, portanto, uma descrição das probabilidades relacionadas com os possíveis valores de "X".

Como existem variáveis do tipo discreta e do tipo contínua, também existem distribuições de probabilidade para quando as variáveis são discretas e distribuições de probabilidade para quando as variáveis são contínuas. Seguem alguns exemplos:

Distribuições de probabilidade para **variáveis do tipo discreta:**

- Distribuição de Bernoulli.
- Distribuição Binomial.
- Distribuição de Poisson.
- Distribuição Geométrica.

Distribuições de probabilidade para **variáveis do tipo contínua:**

- Distribuição Uniforme.
- Distribuição Exponencial.
- Distribuição Normal.
- Distribuição Lognormal.
- Distribuição Gama.
- Distribuição Beta.
- Distribuição Weibull.
- Distribuição Triangular.

3 Aplicação prática: Distribuições de Probabilidade e os Retornos do Ibovespa

3.1 Objetivo

- Encontrar a distribuição de probabilidade que **melhor se ajusta ao comportamento dos retornos diários do Ibovespa** em diferentes períodos de tempo.

3.2 Considerações e Premissas

- O estudo é feito usando a linguagem de programação **Python**.
- É usado o **retorno diário** do Ibovespa, representado pela seguinte equação:

$$Retorno_{diario} = (\frac{Cotacao_d}{Cotacao_{d-1}} - 1) * 100 \tag{1}$$

- As janelas temporais consideradas são os últimos: **6 meses, 12 meses, 2 anos, 5 anos, 10 anos e 20 anos**.
- O retorno diário do Ibovespa é uma **variável contínua**.
- As distribuições de probabilidade consideradas são: **Uniforme, Exponencial, Normal, Lognormal, Gama, Beta e Triangular**.
- É usado o método **SSE - Sum of squared errors** na determinação de qual distribuição de probabilidade melhor se ajusta ao comportamento do retorno diário do Ibovespa, em diferentes janelas temporais.

3.3 Coletar dados do Ibovespa e calcular o Retorno diário

Passo 1: Definir as datas para os períodos desejados.

```
#Data ultimos 6 meses (180 dias)
data_ult_6meses = date.today()-timedelta(180)
#Data ultimos 12 meses (365 dias)
data_ult_12meses = date.today()-timedelta(365)
#Data ultimos 2 anos (730 dias)
data_ult_2anos = date.today()-timedelta(730)
#Data ultimos 5 anos (1825 dias)
data_ult_5anos = date.today()-timedelta(1825)
#Data ultimos 10 anos (3650 dias)
data_ult_10anos = date.today()-timedelta(3650)
#Data ultimos 20 anos (7300 dias)
data_ult_20anos = date.today()-timedelta(7300)
```

Passo 2: Função para coletar dados do Ibovespa e calcular o Retorno diário do período desejado.

```
def f_retorno_diario(ativo, data_periodo):
    #Coletar a Cotacao do Ibovespa
    df_IBOV = yf.download(ativo, start=data_periodo, end=date.today(), progress=False)
    #Calcular o retorno diario do Ibovespa com base nos fechamentos diarios
    df_IBOV["Retorno_Diario"] = df_IBOV["Close"].pct_change(periods=1)
    #Excluir valores NaN
    df_IBOV.dropna(subset=["Retorno_Diario"], inplace=True)
    #Armazenar retornos diarios do periodo em um vetor
    vetor_retorno = df_IBOV["Retorno_Diario"].values
    #Retornar vetor com os retornos diarios
    return vetor_retorno
```

Passo 3: Calcular o vetor com os retornos diários do Ibovespa nos períodos desejados.

```
# "^BVSP" eh a sigla do  ndice  Ibovespa no Yahoo Finance
Retorno_6meses = f_retorno_diario("^BVSP", data_ult_6meses)
Retorno_12meses = f_retorno_diario("^BVSP", data_ult_12meses)
Retorno_2anos = f_retorno_diario("^BVSP", data_ult_2anos)
Retorno_5anos = f_retorno_diario("^BVSP", data_ult_5anos)
Retorno_10anos = f_retorno_diario("^BVSP", data_ult_10anos)
Retorno_20anos = f_retorno_diario("^BVSP", data_ult_20anos)
```

3.4 Estatística Descritiva do Retorno diário do Ibovespa para cada período analisado

```
#Exibir estatisticas
df_Estatisticas
```

	Tamanho_Amostra	Média	Desvio_padrão	Mínimo	1°_Quartil	Mediana	3°_Quartil	Máximo
Retorno_Últimos_6_meses	122.0	-0.000556	0.011764	-0.029006	-0.006842	-0.000039	0.007451	0.024253
Retorno_Últimos_12_meses	249.0	-0.000563	0.012669	-0.037805	-0.008129	-0.000410	0.007842	0.036626
Retorno_Últimos_2_anos	495.0	0.000134	0.012938	-0.051201	-0.007668	0.000428	0.008776	0.036626
Retorno_Últimos_5_anos	1234.0	0.000486	0.016811	-0.147797	-0.007462	0.000604	0.009401	0.139082
Retorno_Últimos_10_anos	2471.0	0.000358	0.015755	-0.147797	-0.008141	0.000333	0.009037	0.139082
Retorno_Últimos_20_anos	4946.0	0.000625	0.017376	-0.147797	-0.008500	0.000851	0.010199	0.146560

3.5 Lista das Distribuições de Probabilidade utilizadas

Passo 1: Criar a lista com as distribuições. Usar o comando **"get_distributions()" da biblioteca "fitter"** para encontrar o nome correto das distribuições.

```
Lista_Distribuicoes = ["uniform","expon","norm","lognorm","gamma","beta","triang"]
```

3.6 Distribuição de Probabilidade do Retorno Diário do Ibovespa

Passo 1: Calcular e Comparar os valores teóricos das distribuições de probabilidade com os valores reais. **OBS:** a função ”Fitter” ordena as distribuições da que mais se ajusta para a que menos se ajusta, com base no ”sumsquare_error”.

```
distribuicao = Fitter(data=Retorno , distributions=Lista_Distribuicoes)
distribuicao.fit()
distribuicao.summary(Nbest=7)
```

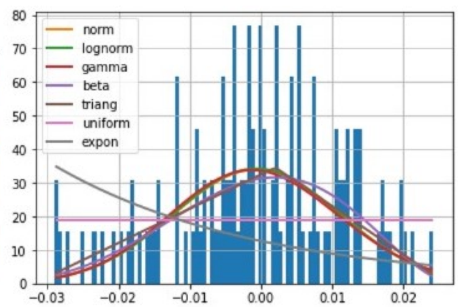
Passo 2: Encontrar a distribuição de probabilidade que melhor se ajusta aos valores reais.

```
distribuicao.get_best(method='sumsquare_error')
```

3.6.1 Calcular e Comparar os valores teóricos das distribuições de probabilidade com os valores reais

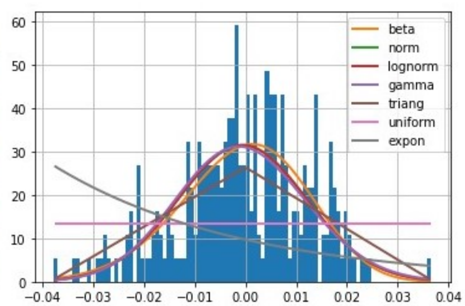
Resultados para o período dos **últimos 6 meses:**

	sumsquare_error	aic	bic	kl_div
norm	35679.495246	-526.970599	702.361910	inf
lognorm	35704.196901	-524.071798	707.250365	inf
gamma	35913.875966	-524.634521	707.964736	inf
beta	35970.528977	-534.124931	712.961056	inf
triang	35976.935158	-546.501499	708.178761	inf
uniform	45752.581155	-582.518634	732.699900	inf
expon	60169.114921	-520.720214	766.117039	inf



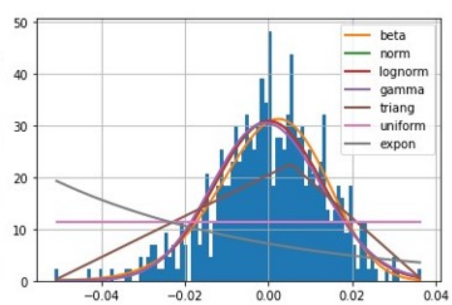
Resultados para o período dos **últimos 12 meses:**

	sumsquare_error	aic	bic	kl_div
beta	7815.287400	-380.336956	880.219458	inf
norm	7975.835803	-397.581536	874.247894	inf
lognorm	7988.424347	-395.645570	880.158043	inf
gamma	8139.929141	-398.569431	884.836239	inf
triang	9483.677908	-462.838402	922.881131	inf
uniform	20000.737546	-515.577011	1103.166717	inf
expon	27573.642556	-454.206397	1183.118423	inf



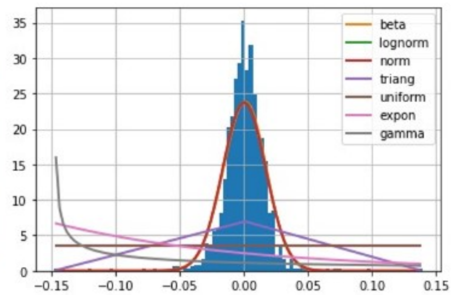
Resultados para o período dos **últimos 2 anos:**

	sumsquare_error	aic	bic	kl_div
beta	1929.852932	-286.974478	698.335682	inf
norm	2038.686427	-264.225000	713.083204	inf
lognorm	2049.363960	-258.393392	721.873537	inf
gamma	2192.781904	-247.506474	755.356096	inf
triang	4958.867209	-424.882078	1159.279222	inf
uniform	14511.744832	-482.477240	1684.596250	inf
expon	21336.543448	-418.790403	1875.400409	inf



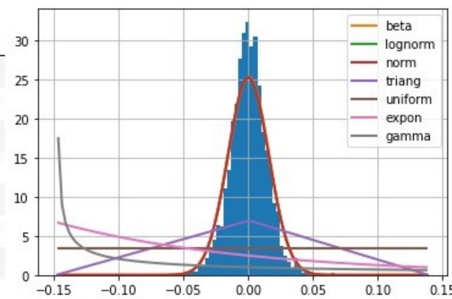
Resultados para o período dos **últimos 5 anos:**

	sumsquare_error	aic	bic	kl_div
beta	480.973173	1890.913951	-1134.208546	inf
lognorm	491.760816	1829.051003	-1113.955262	inf
norm	500.535230	1807.422377	-1099.249330	inf
triang	4759.961774	-185.093754	1687.227779	inf
uniform	6398.268300	-245.738993	2045.111832	inf
expon	7022.737365	-184.259344	2160.028951	inf
gamma	7473.597080	-56.220457	2243.930569	inf



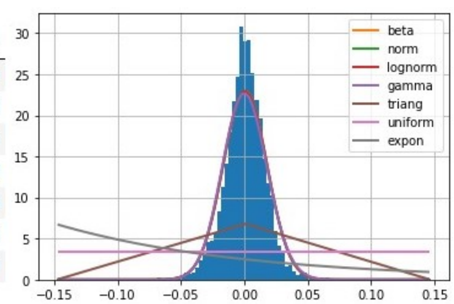
Resultados para o período dos **últimos 10 anos:**

	sumsquare_error	aic	bic	kl_div
beta	251.085498	2168.906496	-5618.901271	inf
lognorm	254.030999	2146.715427	-5597.894891	inf
norm	256.748824	2130.319863	-5579.410972	inf
triang	4557.304148	-184.430046	1535.956809	inf
uniform	6200.154062	-245.738993	2288.824205	inf
expon	6825.857498	-184.264928	2526.395617	inf
gamma	7390.567514	-34.153455	2730.619268	inf



Resultados para o período dos **últimos 20 anos:**

	sumsquare_error	aic	bic	kl_div
beta	253.445869	1776.759678	-14661.451637	inf
norm	255.341388	1769.479929	-14641.610861	inf
lognorm	256.699521	1772.977311	-14606.867007	inf
gamma	282.789493	1787.170464	-14128.111223	inf
triang	3882.122522	-178.962305	-1172.387969	inf
uniform	5416.488036	-240.592569	466.448134	inf
expon	5997.506117	-179.214950	970.426215	inf



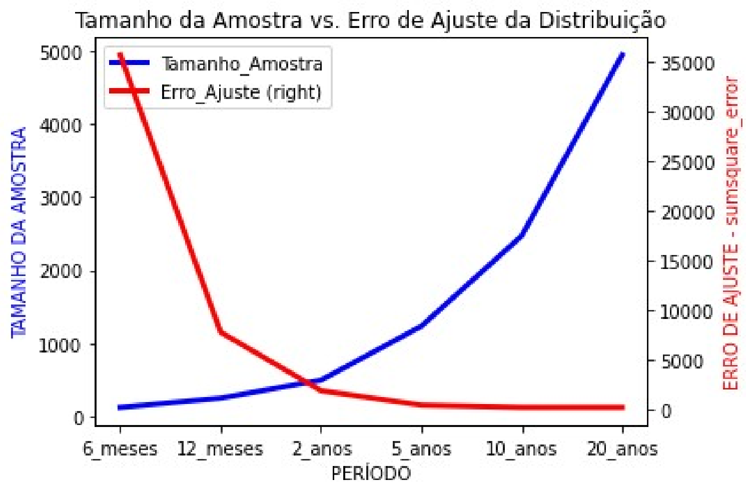
3.6.2 Encontrar a distribuição de probabilidade que melhor se ajusta aos valores reais

Resultados da **melhor distribuição** para cada período analisado:

Períodos	Distribuição de melhor ajuste e seus parâmetros
Últimos 6 meses	(norm: (loc: -0.000385, scale: 0.011818))
Últimos 12 meses	(beta: (a: 198.347, b: 27.639, loc: -0.5127, scale: 0.5836))
Últimos 2 anos	(beta: (a: 3672692.722, b: 33.5037, loc: -8200.8164, scale: 8200.8914))
Últimos 5 anos	(beta: (a: 117231.007, b: 450.6372, loc: -92.9045, scale: 93.2622))
Últimos 10 anos	(beta: (a: 8484.880, b: 789.8121, loc: -4.9615, scale: 5.4237))
Últimos 20 anos	(beta: (a: 44458.980, b: 10724.9507, loc: -8.3087, scale: 10.3138))

4 Considerações finais

4.1 Relação entre o tamanho da amostra e o ajuste das distribuições aos dados reais



Para essa comparação considerou-se a distribuição que melhor se ajustou em cada período analisado.

É possível observar a existência de uma relação inversamente proporcional entre o tamanho da amostra e o erro de ajuste da distribuição aos valores reais.

A distribuição de probabilidade que melhor se ajustou ao comportamento dos retornos diários do Ibovespa nos últimos 6 meses obteve um alto índice de erro entre os valores teóricos da distribuição e os valores reais. Por outro lado, nos períodos de 5 anos ou mais, o erro de ajuste dos valores teóricos frente aos valores reais obteve um valor bem menor.

O fato do erro de ajuste diminuir com o aumento do tamanho da amostra, evidencia o seguinte ponto: quanto menos amostras se tem, maior a incerteza em relação ao comportamento das variáveis ao longo do tempo e/ou ocorrência de eventos.

4.2 Resultados das distribuições encontradas para os períodos analisados

De forma geral, podemos observar que das distribuições de probabilidade testadas, a Exponencial e a Uniforme são as que possuem os piores ajustes em relação aos valores reais nos diferentes períodos analisados.

Por outro lado, as distribuições Beta, Normal e Lognormal são as que possuem os melhores ajustes em relação aos valores reais nos diferentes períodos analisados. Dos 6 períodos analisados, a distribuição Beta obteve o melhor ajuste em 5.

Por que encontrar a distribuição de probabilidade que melhor se ajusta ao comportamento de variáveis de comportamento aleatório/estocástico? Respondendo com um viés no campo das finanças, essas distribuições de probabilidade podem servir como inputs para modelos (matemáticos, de simulação, entre outros) que possuem o objetivo de suportar um processo de tomada de decisão mais inteligente.