

NOME:

TURNO:

PROFESSOR: CLEBER PINHEIRO

DISCIPLINA: ANÁLISE ESTATÍSTICA DE DADOS

SEMESTRE:

DATA:

EXERCÍCIOS DE APRENDIZAGEM

1ª Lista

QUESTÃO 1

Crie e execute um programa em *python* que, a partir do arquivo “dolar.txt” e da seleção das opções desejada pelo usuário, determine: 1) o valor médio do dólar no mês (considerando o mês de agosto de 2022); 2) a média móvel semanal (7 dias) da cotação do dólar; 3) o valor do desvio padrão amostral quanto a cotação mensal. As características de opções estão na tabela a seguir.

Valor da Opção desejada pelo usuário	Descrição da saída via terminal/Resultado
1	Valor médio mensal em agosto
2	Média móvel semanal
3	Desvio padrão amostral da cotação mensal

Cotação do dólar – conteúdo do arquivo “dólar.txt” (fonte: <http://www.ipeadata.gov.br>):

01/08/2022	5,16
02/08/2022	5,2323
03/08/2022	5,284
04/08/2022	5,2403
05/08/2022	5,2159
08/08/2022	5,1241
09/08/2022	5,1218
10/08/2022	5,0491
11/08/2022	5,1121
12/08/2022	5,1017
15/08/2022	5,0919
16/08/2022	5,1334
17/08/2022	5,1779
18/08/2022	5,1767
19/08/2022	5,1955
22/08/2022	5,1703
23/08/2022	5,1024
24/08/2022	5,1044
25/08/2022	5,1167
26/08/2022	5,0897
29/08/2022	5,0423
30/08/2022	5,0611
31/08/2022	5,1784

Exemplo: se o usuário digitar para opção o valor igual a 1, uma saída no terminal deverá mostrar o a média mensal do dólar em agosto.

Observações:

- 1) Use estrutura condicional;
- 2) **Contextualização:**

As médias móveis suavizam os dados de preços para formar um indicador de tendência sequencial. Elas não preveem a direção dos preços, mas, antes, definem a sua direção atual com um atraso.

As médias móveis atrasam porque elas são baseadas em preços (ou cotações) passados. Apesar disso, as médias móveis ajudam a suavizar o preço da ação e filtram o ruído. Também, formam os fundamentos e estruturas para muitos outros indicadores como, por exemplo, valores de ativos financeiros.

O tipo de média móvel mais utilizado é a média móvel simples, onde esta é formada através do cálculo do preço (ou cotação) médio de um título ou indicador ao longo de um determinado número de períodos, conforme a definição dada pela fórmula abaixo:

$$\bar{x}_{t+1} = \frac{x_t + x_{t-1} + x_{t-2} + \dots + x_{t-k+1}}{k}$$

onde n é número escolhido de períodos consecutivos (diários/semanais/mensais etc), x_1, x_2, \dots, x_n sendo “ k ” o tamanho do conjunto de dados reais observados no passado (dados históricos) para o cálculo da média móvel, e \bar{x}_{t+1} é o valor futuro/ média móvel no instante “ $t+1$ ” (futuro) dos “ k ” valores imediatamente anteriores (passado).

Obs:

- 1) use as bibliotecas matemáticas *numpy* e *pandas*;
- 2) use a indexação com listas e arrays.

QUESTÃO 2

Em relação ao código da questão anterior, incorpore opções para a determinação da moda e da mediana.

QUESTÃO 3

Precipitação em Agosto de 2.022	Em meteorologia, precipitação descreve qualquer tipo de fenômeno relacionado à queda de água do céu. Isso inclui neve, chuva e chuva de granizo.
01/08/2022 0,6 mm	Para previsões climáticas na agricultura, por exemplo, é necessário o mapeamento do indicador de precipitação pluviométrica. Neste sentido, considere a tabela ao lado que servirá de base para a previsão de chuvas para o mês de agosto em Brasília-DF.
02/08/2022 1,6 mm	
03/08/2022 0,7 mm	A partir de tais dados, crie e execute um código em <i>python</i> que, a partir do arquivo “precipitacao.xlsx” (este último contém os dados da tabela de precipitação ao lado), determine: 1) o valor médio da precipitação pluviométrica no mês de agosto de 2.022; 2) o desvio padrão amostral da precipitação; 3) se houve ou não grande variação pluviométrica. Para tanto, deverá ser retornado via terminal as mensagens:
04/08/2022 1,4 mm	
05/08/2022 1,2 mm	<ul style="list-style-type: none"> • “houve grande variação de precipitação pluviométrica”;
06/08/2022 0,7 mm	
07/08/2022 0,6 mm	ou
08/08/2022 0,5 mm	
09/08/2022 3,3 mm	<ul style="list-style-type: none"> • “NÃO houve grande variação de precipitação pluviométrica”.
10/08/2022 1,0 mm	
11/08/2022 1,1 mm	Para que haja grande variação , o desvio padrão amostral deverá ser superior a 0,9.
12/08/2022 0,5 mm	
13/08/2022 0,1 mm	Observação:
14/08/2022 0,3 mm	
15/08/2022 0,4 mm	Fonte dos dados: https://pt.climate-data.org
16/08/2022 0,3 mm	
17/08/2022 0,7 mm	
18/08/2022 1,0 mm	
19/08/2022 1,9 mm	
20/08/2022 2,8 mm	
21/08/2022 1,2 mm	
22/08/2022 2,1 mm	
23/08/2022 1,3 mm	
24/08/2022 2,6 mm	
25/08/2022 1,4 mm	
26/08/2022 0,1 mm	

27/08/2022	0,8 mm
28/08/2022	1,4 mm
29/08/2022	2,3 mm
30/08/2022	0,4 mm
31/08/2022	0,8 mm

Obs:

Use as bibliotecas matemáticas *numpy* e *pandas*.

QUESTÃO 4

Considerando a questão 3, crie um código em ambiente *python* que construa um gráfico do “índice pluviométrico” versus “data”. O gráfico deve ser mostrado via terminal de execução.

Obs:

- 1) Use as bibliotecas *numpy*, *pandas*, *datetime* e a subbiblioteca *matplotlib* (*matplotlib.pyplot* e *matplotlib.dates*).
 - Neste caso, use a função *to_datetime* para mostrar as datas no formato tradicional;
- 2) Use a função *values.tolist()* para transformar dados de um *dataframe* em lista;
- 3) Use também as instruções abaixo que darão a formatação do gráfico e o espaçamento entre os pontos do eixo “x”:
 - *plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%d'))* # neste caso, pode-se mudar a formatação da data
 - *plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=5))* # neste caso, pode-se mudar o espaçamento entre as datas
 - *plt.gcf().autofmt_xdate()* # esta instrução farão a atualização do eixo x

QUESTÃO 5

Para atender e documentar uma sistemática relativa a pesquisas de mercado e opinião sobre um determinado assunto (ou sobre uma variável de estudo), usa-se métodos de dimensionamento de amostra. Isso é feito mediante aplicação de um questionário com questões objetivas, com possibilidades de respostas mediante uma escala previamente estabelecida.

Neste sentido, deve-se utilizar uma indicação da variabilidade existente na pesquisa, ou seja, deve ser apresentado um indicador numérico. Um exemplo de indicador a ser utilizado é a resposta obtida de cada pergunta/questão a ser avaliada pelo respondente. Tais questões podem apresentar um “grau” de importância diferente no contexto do questionário.

A medida geral de variabilidade é estimada através de uma média ponderada dos desvios padrões associados às respostas, onde os pesos estatísticos retratam a importância de cada pergunta a ser respondida no questionário. A fórmula para obter a variabilidade é dada por:

$$\sigma_q = \frac{\sum_{i=1}^k \sigma_i \times w_i}{\sum_{i=1}^k w_i} = \frac{\sigma_1 \times w_1 + \sigma_2 \times w_2 + \sigma_3 \times w_3 + \dots}{w_1 + w_2 + w_3 + \dots}$$

onde:

σ_q é o desvio padrão populacional para a variável em análise no questionário;

k é o número de questões;

σ_i é o desvio padrão populacional das respostas obtidas da pergunta/questão i ;

w_i é o peso estatístico quanto à importância da pergunta/questão i no questionário.

Dessa forma, o tamanho da amostra n é dado pela seguinte fórmula:

$$n = \left(\frac{z \times \sigma_q}{E} \right)^2$$

Nesse caso, n é o tamanho mínimo da amostra a ser consultada, E é o fator erro (este é fixado antes de se iniciar a pesquisa) e z a variável padronizada associada à margem de confiança do estudo (por exemplo, para uma margem de confiança de 90%, o valor de z é igual a 1,65; para uma margem de confiança de 95%, o valor de z é igual a 1,96).

Portanto, o dimensionamento do tamanho da amostra depende de quatro fatores principais: (i) o nível de significância/margem de confiança desejado(a); (ii) a variabilidade das respostas e grau de importância de cada pergunta; (iii) erro máximo de estimação/estimativa admitida na pesquisa.

O tamanho da amostra é determinado em duas etapas:

- 1) Aplica-se um questionário na forma de **PRÉ-TESTE** (questionário preliminar a ser aplicado a um conjunto de pessoas de maneira subjetiva, a critério do examinador, com o intuito de verificar problemas de compreensão das questões e fornecer subsídios para a sua modificação);
- 2) Com base nas respostas obtidas na pesquisa da etapa 1 acima, calcula-se o desvio padrão geral para a variável em análise no questionário σ_q e o tamanho da amostra n . Esse último conjunto “corrige” o tamanho escolhido pelo observador na etapa 1 no sentido probabilístico.

Como exemplo, considere um questionário com cinco questões, tendo um padrão de respostas na escala **LIKERT**, numeradas de 1 a 5 (veja a ilustração abaixo). As respostas obtidas de dez respondentes através de um pré-teste são dadas na tabela a seguir:

1	2	3	4	5
Discordo totalmente	Discordo parcialmente	Não concordo, nem discordo	Concordo parcialmente	Concordo totalmente

Respondentes	Questões/perguntas				
	1ª	2ª	3ª	4ª	5ª
1º	3	3	1	3	3
2º	5	4	2	4	5
3º	4	3	1	3	5
4º	4	3	2	3	5
5º	5	3	2	3	4
6º	4	2	2	4	5
7º	3	3	1	3	4
8º	3	4	2	3	5
9º	5	4	3	3	5
10º	5	4	2	2	4
Desvio Padrão Populacional		???	???	???	???

Considere ainda que as questões de um a cinco possuem pesos estatísticos iguais entre si (peso igual a 1). Com um nível de confiança de 95% e um fator de erro igual a 0,2, Crie e execute um programa em *python* que:

1) **simule** aleatoriamente as diversas respostas possíveis entre 1 e 5 para os respondentes. Neste caso, use a função “*random.random*” (este último pertencente às bibliotecas “*random*”). Use também arranjos na forma de matrizes para o armazenamento das respostas iniciais;

2) a partir do item **anterior**, determine o tamanho mínimo da amostra para a realização da pesquisa.

Obs: use as bibliotecas *numpy* e *random*.

QUESTÃO 6

A sigla “**ISO**” refere-se à *International Organization for Standardization*, organização não governamental que congrega normas de padronização em vários países (normatização).

Neste contexto, a expressão **ISO 9001** designa um grupo de normas técnicas que estabelecem um modelo de gestão da qualidade. A adoção das normas **ISO** é vantajosa para as organizações, uma vez que lhes confere maior organização, produtividade, credibilidade e qualidade.

Com o intuito de verificar a obtenção da certificação **ISO 9001:2015**, foi feita uma pesquisa engloba 35 fabricantes. Foi obtido um levantamento dos dados quanto à certificação, conforme informações/**legendas** abaixo:

C = Fabricante já certificado;

B = Em busca de certificação (já em processo de implementação/início de certificação – aguardando aprovação);

R = Apenas interesse futuro na certificação (não há pretensão e nem compromisso para obtenção no **atual** momento);

N = Sem interesse na certificação;

I = Indeciso quanto à obtenção da certificação.

Os dados foram organizados de acordo com a tabela de dados abaixo:

R	C	N	N	I	B	C
C	B	N	R	C	I	C
B	B	C	R	B	I	B
I	R	I	I	R	C	R
N	N	C	C	B	B	R

Tabela 1

Com base nas informações do texto acima, crie e execute um código em *python* que, a partir de um “*dataframe*” contendo as informações amostrais de legendas acima, determine:

- 1) uma nova tabela (“*dataframe*”) contendo o quantitativo de cada legenda. As informações devem ser carregadas a partir de uma lista de dados contendo a **tabela 1** acima;
- 2) uma outra tabela (“*dataframe*”) contendo os percentuais de cada legenda descrita na lista de 35 fabricantes;
- 3) a **porcentagem** de uma legenda e **quantidade** de fabricantes relacionada a mesma, mediante especificação/solicitação do usuário via entrada no código da legenda (deve-se usar a notação permitida do módulo *pandas*). Veja a formatação da resposta da execução via **terminal** abaixo:

Saída:

Digite a legenda desejada para consulta: ???

A legenda especificada possui as seguintes características:

Legenda solicitada: ???

Quantidade de fabricantes: ???

Percentual desta legenda na amostra: ???%

Obs:

- 1) Use a biblioteca matemática *pandas*;
- 2) Use a indexação com listas e arrays.

QUESTÃO 7

Execute um programa em ambiente *python* para criar um gráfico de *superfície (3D)* para a cotação do dólar (venda) referente ao período “*Agosto/2.022*”, conforme a tabela abaixo, onde inclui-se a variação percentual do valor no fim do dia (sendo esta obtida comparando o valor de fechamento da cotação de venda entre dois dias consecutivos): neste sentido, o código **deverá calcular automaticamente, na forma percentual, as variações diárias a partir do segundo dia**. A tabela a seguir deve ser armazenada via “*dataframe*”. As variáveis serão rotuladas da seguinte forma:

- 1) Eixo “*x*”: rótulo “*Data*”;
- 2) Eixo “*y*”: rótulo “*%Dia*”;
- 3) Eixo “*z*”: rótulo “*Cotação (Venda – R\$)*”.

Data	Cotação (Venda - R\$)	Variação (%) - Dia
01/08/2022	5,1606	-0,54
02/08/2022	5,2329	1,4
03/08/2022	5,2846	???
04/08/2022	5,2409	???
05/08/2022	5,2165	???
08/08/2022	5,1247	???
09/08/2022	5,1224	???
10/08/2022	5,0497	???
11/08/2022	5,1127	???
12/08/2022	5,1023	???
15/08/2022	5,0925	???
16/08/2022	5,134	???
17/08/2022	5,1784	???
18/08/2022	5,1773	???
19/08/2022	5,1961	???
22/08/2022	5,1709	???
23/08/2022	5,103	???
24/08/2022	5,105	???
25/08/2022	5,1173	???
26/08/2022	5,0903	???
29/08/2022	5,0428	???
30/08/2022	5,0617	???
31/08/2022	5,179	???

(fonte: Banco Central do Brasil - <https://www.bcb.gov.br>).

QUESTÃO 8

Considere a tabela abaixo. Crie um código em ambiente **python** que determine o valor do coeficiente de correlação de Pearson as variáveis x e y . A fórmula do coeficiente é:

$$\rho = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \times \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

x	y
1	1
2	1
3	2
4	2
5	4

Obs:

O código deve possuir apenas os seguintes comandos (funções): **sum**, **len**, **for** e **print**.

QUESTÃO 9

Considerando os dados do problema anterior, elabore um código em ambiente **python** que determine:

- 1)** a reta de regressão $y=ax+b$ entre as variáveis envolvidas;
- 2)** o valor estimado para a variável y quando $x=10$;

As fórmulas que permitem o cálculo dos coeficientes **a** e **b** são:

$$y = ax + b$$

$$a = \frac{n \sum_i x_i \times y_i - \left(\sum_i x_i \right) \times \left(\sum_i y_i \right)}{n \sum_i x_i^2 - \left(\sum_i x_i \right)^2}$$

$$b = \frac{\sum_i y_i - a \times \left(\sum_i x_i \right)}{n}$$

Obs:

O código deve possuir apenas os seguintes comandos (funções): **sum**, **len**, **for**, **float**, **input** e **print**.

QUESTÃO 10

Considerando os dados do problema 08, elabore um código em ambiente *python* que calcule o coeficiente de determinação referente à reta de regressão $y=ax+b$ entre as variáveis envolvidas.

A fórmula para o seu cálculo é:

$$R^2 = \frac{\text{variação explicada}}{\text{variação total}}$$

onde

$$\text{variação explicada} = \sum_i (y_i'' - \bar{y})^2$$

$$\text{variação total} = \sum_i (y_i - \bar{y})^2$$

Obs:

- 1) Use a biblioteca *numpy*;
- 2) O código deve possuir apenas os seguintes comandos (funções): *array*, *sum*, *len*, *for* e *print*.

QUESTÃO 11

A análise quanto ao grau de dependência entre diferentes tipos de variáveis no contexto do agronegócio é fundamental para a formulação de estratégias agrícolas, alocação de recursos e tomadas de decisão, pois a obtenção do melhor retorno financeiro, permitindo o desenvolvimento e o planejamento do setor agrícola, visando maximizar a produtividade e a rentabilidade.

Nesse sentido, considere os dados sobre o valor da produção (em R\$ 1.000) de uma cultura e a área destinada a colheita (em hectares) para 5 estados brasileiros, são apresentados na tabela abaixo.

Área (milhares de hectares)	Receita gerada pela produção (em milhares de R\$)
8277	80,211
7672	39,009
8567	86,956
9714	114,606
43510	411,926

Tabela: Produção (em R\$ 1.000) de banana (cacho) e área destinada à colheita (em hectares). Fonte: IBGE, Diretoria de Pesquisas, Coordenação de Agropecuária, Produção Agrícola Municipal 2013.

Determine o valor do coeficiente de determinação para as variáveis área (variável independente) e receita gerada (variável dependente). Determine também o tipo de correlação entre as variáveis supracitadas, de acordo com a classificação abaixo:

Coeficiente de correlação:

- Igual a 1 (positivo ou negativo): indica uma correlação perfeita;
- Maior ou igual a 0,9 e menor que 1 (positivo ou negativo): indica uma correlação muito forte;
- Maior ou igual a 0,7 e menor que 0,9 (positivo ou negativo): indica uma correlação forte;
- Maior ou igual a 0,5 e menor que 0,7 (positivo ou negativo): indica uma correlação moderada;

- Maior ou igual a 0,3 e menor que 0,5 (positivo ou negativo): indica uma correlação fraca;
- Maior ou igual a 0 e menor que 0,3 (positivo ou negativo): indica uma correlação desprezível.

Obs:

- 1) Use a biblioteca *numpy*;
- 2) O código deve possuir apenas os seguintes comandos (funções): *array*, *sum*, *len*, *for* e *print*.

Resposta: $R^2=0,9816$, aproximadamente ; $\rho = 0,990765$, aproximadamente.

QUESTÃO 12

Repita o problema anterior, considerando cálculos manuais de acordo com as definições dos coeficientes da reta de regressão. Neste mesmo problema, determine o valor do coeficiente de determinação R^2 . Use a tabelas abaixo para auxiliar nesses cálculos.

Tabela para auxílio no cálculo da reta de regressão

x	y	$x \times y$	x^2

Tabela para auxílio no cálculo do coeficiente de determinação

x	y	$(y_i' - \bar{y})^2$	$(y_i - \bar{y})^2$

Resposta: $y' = 0,0095x - 1,9029$; $R^2 = 0,9816$, aproximadamente: correlação **muito forte** !

QUESTÃO 13

No campo da auditoria, existem técnicas de mineração de dados que servem de auxílio no processo de detecção de fraudes financeira (por exemplo, em auditoria de obras públicas, fraude em registro contábil de certos parâmetros financeiros em empresas públicas e privadas). Uma metodologia muito utilizada é a **Lei de Newcomb-Benford**, que confronta a probabilidade de ocorrência esperada de dígitos com a probabilidade observada, ou seja, detecta discrepância em relação a um padrão universal de ocorrência dos dígitos. Tal probabilidade esperada é descrita pela referida lei através da função

Primeiro Dígito	Probabilidade esperada $P(d)$
1	0,301029996
2	0,176091259
3	0,124938737
4	0,096910013
5	0,079181246

$P(d) = \log_{10} (1 + 1/d)$, onde d é o valor do primeiro dígito e $P(d)$ é a probabilidade de ocorrência deste dígito num dado número.

Frank Benford demonstrou que tal lei se aplica **sob certas condições** a uma ampla variedade de conjuntos de dados, incluindo valores de contas comerciais, precificação de ações, mercado imobiliário, taxas de mortalidade, registros de imposto de renda, valores de faturas de compra e venda etc.

6	0,06694679
7	0,057991947
8	0,051152522
9	0,045757491

Tabela 1 – Probabilidade dada pela **Lei de Newcomb-Benford**.

A tabela 1 ilustra os valores da **Lei de Newcomb-Benford**.

Como ilustração para verificação, tem-se ao lado um conjunto de registros do primeiro dígito relativo a 500 valores de lucros de uma empresa, a partir de observações de dados históricos, obtidas de uma pesquisa. A tabela 2 mostra as frequências observadas.

Primeiro Dígito	Frequência observada
1	148
2	75
3	55
4	50
5	45
6	40
7	34
8	28
9	25

Tabela 2 – Frequência observada do primeiro dígito.

Com base nessas informações, elabore um código em ambiente **python** que:

- calcule o valor médio para o primeiro dígito;
- mostre um gráfico de barras mostrando a frequência observada *versus* primeiro dígito para a variável lucro.

Obs:

- Use as bibliotecas **matplotlib** e **numpy**;
- O código deve possuir exclusivamente os seguintes comandos (funções): **array**, **sum**, **len**, **for**, **print**, **bar**, **xlabel**, **ylabel**, **xticks** e **show()**.

QUESTÃO 14

O cálculo do retorno diário é uma ferramenta valiosa na análise de ativos financeiros. Neste estudo de caso, ao calcular os retornos diários de um ativo durante uma semana, o investidor pode avaliar o desempenho do ativo, identificar períodos de alta volatilidade e tomar decisões informadas sobre sua carteira de investimentos. Tal grandeza fornece uma visão mais detalhada e dinâmica do comportamento do ativo ao longo do tempo, auxiliando o investidor a gerenciar riscos e buscar oportunidades de rentabilidade no mercado financeiro.

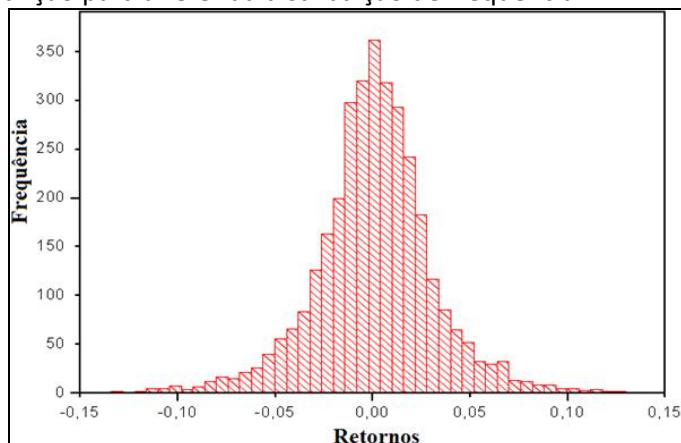
A medida do retorno diário permite entender o comportamento de um ativo financeiro. Ao analisa-lo ao longo do tempo, o investidor pode observar a variação do preço do ativo, bem como sua volatilidade. Valores positivos de retorno indicam ganhos, enquanto valores negativos indicam perdas.

O retorno diário de um ativo financeiro é usualmente definido como a diferença logarítmica entre valores diários consecutivos dos preços desse ativo, sendo definido por

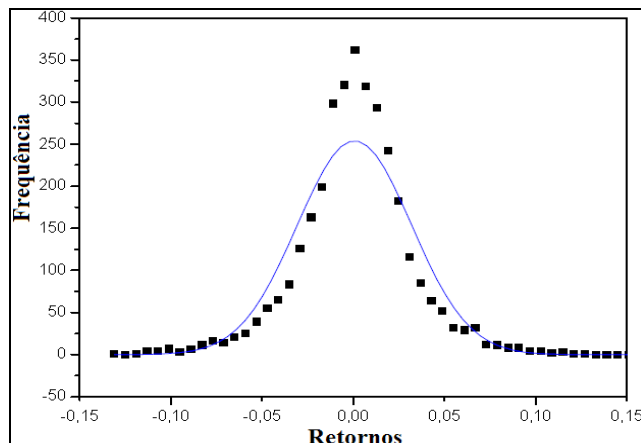
$$r_t = \ln(y_t) - \ln(y_{t-1}) = \ln\left(\frac{y_t}{y_{t-1}}\right)$$

onde y_t é o valor de fechamento da bolsa no instante t (que por sua vez é medido em dias de pregão), y_{t-1} é o valor de fechamento da bolsa no dia anterior. No caso de dia anterior sem pregão, o retorno é calculado utilizando-se os preços de fechamento do dia de pregão em questão e do penúltimo dia útil anterior.

Abaixo é apresentada uma distribuição de frequência dos retornos diários num certo período de uma determinada bolsa de valores. A figura (a) apresenta um histograma de frequência, enquanto que na figura (b), os dados empíricos aparecem como pontos/quadrados. Nesta figura, a linha sólida/curva representa um ajuste de função para a referida distribuição de frequência.



(a) Histograma dos retornos diários



(b) Gráfico de dispersão com ajuste de linha

Figura 1

Considere o seguinte conjunto de dados representando o comportamento de um ativo financeiro no mercado de ações:

Frequência	Preço do ativo (R\$)	Retorno diário
11	50	-0,9
11	20,5328	-089
12	8,8642	???
13	4,023	???
14	1,9194	???
16	0,9627	???
17	0,5076	???
18	0,2814	???
19	0,164	???
20	0,1005	???
21	0,0647	???
21	0,0438	???
22	0,0312	???
23	0,0233	???
24	0,0183	???
24	0,0151	???
25	0,0131	???
25	0,012	???
25	0,0115	???
25	0,0116	???
25	0,0123	???
25	0,0137	???

Com base nessas informações, elabore um código em ambiente *python* que:

- a) calcule todos os valores dos retornos diários da tabela acima referente ao ativo financeiro mencionado;
- b) mostre um gráfico de dispersão do retorno diários *versus* frequência observada. Neste caso, o tipo de gráfico construído se assemelha à curva normal? Explique.

Obs:

- 1) Use as bibliotecas *matplotlib* e *numpy*;
- 2) O código deve possuir necessariamente (mas, não exclusivamente) os seguintes comandos (funções): *array*, *size*, *append*, *log*, *range*, *for*, *xlabel*, *ylabel*, *title*, *show()* e *print*.

Gabarito Sugerido:

Questão 01

```
import pandas as pd
import numpy as np

# Carrega os dados do arquivo dolar.xlsx
def carregar_dados(arquivo):
    planilha = pd.read_excel(arquivo)
    tabela=pd.DataFrame(planilha)
    display(tabela)
    return planilha

# Calcula o valor médio mensal em agosto
def calcular_media_mensal(dados):
    mes_agosto = dados['Cotacao']
    media_mensal = mes_agosto.mean()
    return media_mensal

# Calcula a média móvel semanal
def calcular_media_movel_semanal(dados):
    periodo = 7 # período da média móvel
    media_movel = []
    for i in range(len(dados) - periodo + 1):
        media = sum(dados['Cotacao'][i:i+periodo]) / periodo
        media_movel.append(media)
    return media_movel

# Calcula o desvio padrão amostral da cotação mensal
def calcular_desvio_padrao(dados):
    desvio_padrao = dados['Cotacao'].std(ddof=1)
    return desvio_padrao

#=====Código Principal=====#
arquivo = "dolar.xlsx"
dados = carregar_dados(arquivo)

opcao = int(input("Selecione uma opção:\n1) Valor médio mensal em agosto\n2) Média móvel semanal\n3) Desvio Padrão Amostral da cotação mensal\n"))
```

```
if opcao == 1:
    media_mensal = calcular_media_mensal(dados)
    print("O valor médio mensal em agosto é:", round(media_mensal,2))
elif opcao == 2:
    media_movel = calcular_media_movel_semanal(dados)
    print("A média móvel semanal da cotação do dólar é dada pela lista abaixo:")
    display(media_movel)
elif opcao == 3:
    desvio_padrao = calcular_desvio_padrao(dados)
    print("O desvio padrão amostral da cotação mensal é:", desvio_padrao)
else:
    print("Opção inválida.")
```

Questão 02

```
import pandas as pd
import numpy as np

# Carrega os dados do arquivo dolar.xlsx
def carregar_dados(arquivo):
    planilha = pd.read_excel(arquivo)
    tabela=pd.DataFrame(planilha)
    display(tabela)
    return planilha

# Calcula o valor médio mensal em agosto
def calcular_media_mensal(dados):
    mes_agosto = dados['Cotacao']
    media_mensal = mes_agosto.mean()
    return media_mensal

# Calcula a média móvel semanal
def calcular_media_movel_semanal(dados):
    periodo = 7 # período da média móvel
    media_movel = []
    for i in range(len(dados) - periodo + 1):
        media = sum(dados['Cotacao'][i:i+periodo]) / periodo
        media_movel.append(media)
    return media_movel

# Calcula o desvio padrão amostral da cotação mensal
def calcular_desvio_padrao(dados):
    desvio_padrao = dados['Cotacao'].std(ddof=1)
    return desvio_padrao

# Calcula a moda
def calcular_moda(dados):
    moda = dados['Cotacao'].mode()
    return moda

# Calcula a mediana
def calcular_mediana(dados):
    mediana = dados['Cotacao'].median()
    return mediana
```

```
arquivo = "dolar.xlsx"
dados = carregar_dados(arquivo)

opcao = int(input("Selecione uma opção:\n1) Valor médio mensal em agosto\n2) Média móvel semanal\n3) Desvio Padrão Amostral da cotação mensal\n4) Moda\n5) Mediana\n"))

if opcao == 1:
    media_mensal = calcular_media_mensal(dados)
    print("O valor médio mensal em agosto é:", round(media_mensal,2))
elif opcao == 2:
    media_movel = calcular_media_movel_semanal(dados)
    print("A média móvel semanal da cotação do dólar é dada pela lista abaixo:")
    display(media_movel)
elif opcao == 3:
    desvio_padrao = calcular_desvio_padrao(dados)
    print("O desvio padrão amostral da cotação mensal é:",
round(desvio_padrao,2))
elif opcao == 4:
    moda = calcular_moda(dados)
    print("A moda da cotação do dólar é:")
    print(moda)
elif opcao == 5:
    mediana = calcular_mediana(dados)
    print("A mediana da cotação do dólar é:", round(mediana,2))
else:
    print("Opção inválida.")
```

Questão 03

```
import pandas as pd
import numpy as np

# Carrega os dados do arquivo indice.xlsx
def carregar_dados(arquivo):
    planilha = pd.read_excel(arquivo)
    tabela=pd.DataFrame(planilha)
    display(tabela)
    return planilha

# Calcula o valor médio da precipitação em agosto
def calcular_media_mensal(dados):
    mes_agosto = dados['Indice']
    media_mensal = mes_agosto.mean()
    return media_mensal

# Calcula o desvio padrão amostral da precipitação
def calcular_desvio_padrao(dados):
    desvio_padrao = dados['Indice'].std(ddof=1)
    return desvio_padrao

#=====Código Principal=====#
arquivo = "indice.xlsx"
dados = carregar_dados(arquivo)

media_mensal = calcular_media_mensal(dados)
print("O valor médio mensal em agosto é:", round(media_mensal,2))
```

```
desvio_padrao = calcular_desvio_padrao(dados)
print("O valor do desvio padrão amostral é igual a", desvio_padrao)
if desvio_padrao > 0.9:
    print("Houve grande variação de precipitação pluviométrica!")
else:
    print("NÃO houve grande variação de precipitação pluviométrica!")
```

Questão 04

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import datetime
import matplotlib.dates as mdates

planilha=pd.read_excel('indice.xlsx')
x1=planilha['Data'].values.tolist()
y=planilha['Indice'].values.tolist()
x2=pd.Series(pd.to_datetime(x1))

plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%d'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=5))
plt.plot(x2,y)
plt.gcf().autofmt_xdate()
plt.show()
```

Questão 05

```
import math
import numpy as np
import random

numrespond=300 # PODE afetar o tamanho MÍNIMO da amostra;
numquest=5 # NÃO afeta o tamanho MÍNIMO da amostra.
E=0.2
escalamax=5 # AFETA o tamanho MÍNIMO da amostra
matriz = np.zeros((numrespond, numquest))
mediaq =np.empty(numquest)
somaq =np.empty(numrespond)
somaquad =np.empty(numrespond)
var =np.empty(numquest)
desvio =np.empty(numquest)
arraysomaq =np.empty(numrespond)
somarespuestas = np.zeros(numrespond)
somadasvar = 0
```



```
varianciasomaresp = 0
tamanhon = 0

for j in range(numquest):
    for i in range(numrespond):
        numaleatorio=round(random.random()*escalamax,0)
        correcao=0
        if numaleatorio==0:
            correcao=1
        if numaleatorio==escalamax:
            correcao=0
        matriz[i][j]=numaleatorio+correcao

for j in range(numquest):
    somaq=0
    somaquad=0

    for linha in range(numrespond):
        somaq=somaq+matriz[linha,j]

    mediaq[j]=somaq/numrespond

    for linha in range(numrespond):
        somaquad=somaquad+(matriz[linha,j]-mediaq[j])**2

    var[j]=somaquad/(numrespond-1)
    desvio[j]=math.sqrt(var[j])
    somadasvar+=var[j]

for linha in range(numrespond):
    for j in range(numquest):
        somarespostas[linha]=somarespostas[linha]+matriz[linha,j]

varianciasomaresp=(np.std(somarespostas, ddof=0))**2
mediadesvio=np.mean(desvio)
tamanhon=(1.96*mediadesvio/E)**2

print(matriz)
print(round(tamanhon,0))
```

Questão 06

```
import pandas as pd

# Dados fornecidos
legendasfabricantes = ['R', 'C', 'N', 'N', 'I', 'B', 'C', 'C', 'B', 'N', 'R', 'C',
                        'I', 'C', 'B', 'B', 'C', 'R', 'B', 'I', 'B', 'I', 'R', 'I', 'I', 'R', 'C', 'R', 'N',
                        'N', 'C', 'C', 'B', 'B', 'R']

# Criar um DataFrame com os dados
df = pd.DataFrame(legendasfabricantes, columns=['Legenda'])
```

```
# Contar a quantidade de cada legenda
quantitativo_por_legenda = df['Legenda'].value_counts().sort_index()

# Calcular os percentuais de cada legenda
percentuais_por_legenda = quantitativo_por_legenda / len(df) * 100

# Solicitar a entrada da legenda desejada
legenda_desejada = input("Digite a legenda desejada para consulta: ")

# Verificar se a legenda solicitada está presente no DataFrame
if legenda_desejada in quantitativo_por_legenda.index:
    quantidade_legenda = quantitativo_por_legenda[legenda_desejada]
    percentual_legenda = percentuais_por_legenda[legenda_desejada]

    # Imprimir os resultados
    print(f"A legenda especificada possui as seguintes características:")
    print(f"Legenda solicitada: {legenda_desejada}")
    print(f"Quantidade de fabricantes com essa legenda: {quantidade_legenda}")
    print(f"Percentual em relação ao total: {percentual_legenda:.2f}%")
else:
    print("Legenda não encontrada no DataFrame.")

# Lista de referência
legendas = ['C', 'B', 'R', 'N', 'I']

# lista para armazenar a contagem de cada legenda
contagem_legendas = {}

# Calcular a contagem de cada legenda
for legenda in legendasfabricantes:
    if legenda in legendas:
        if legenda in contagem_legendas:
            contagem_legendas[legenda] += 1
        else:
            contagem_legendas[legenda] = 1

# Calcular os percentuais
total_elementos = len(legendasfabricantes)
percentuais = {legenda: (contagem / total_elementos * 100) for legenda, contagem in
                contagem_legendas.items()}

# Criar DataFrame com os percentuais
df_percentuais = pd.DataFrame(percentuais.items(), columns=['Legenda', 'Percentual'])

# Exibir o DataFrame criado
display(df_percentuais)
```

Questão 07

```
import pandas as pd
import matplotlib.pyplot as plt

# Dados de exemplo para cotação do dólar (venda) e variação percentual
data = pd.bdate_range(start='2022-08-01', end='2022-08-31', )
```

```
cotacao_venda = [5.1606, 5.2329, 5.2846, 5.2409, 5.2165, 5.1247, 5.1224, 5.0497,  
5.1127, 5.1023, 5.0925,  
5.134, 5.1784, 5.1773, 5.1961, 5.1709, 5.103, 5.105, 5.1173, 5.0903,  
5.0428, 5.0617, 5.179]  
data_dia = list(map(int, [d.strftime('%d') for d in data]))  
  
variacao_percentual = [0.0] + [(cotacao_venda[i] - cotacao_venda[i-1]) /  
cotacao_venda[i-1] * 100 for i in range(1, len(cotacao_venda))]  
  
# Criar DataFrame  
df = pd.DataFrame({'Data': data, 'Variação(%)': variacao_percentual, 'Cotação (Venda  
- R$)': cotacao_venda})  
#display(df)  
# Criar figura e eixo 3D  
fig = plt.figure()  
ax = fig.add_subplot(111, projection='3d')  
  
# Plotar gráfico de superfície  
surface = ax.plot_trisurf(data_dia, cotacao_venda, variacao_percentual, cmap="Blues")  
  
# Adicionar rótulos dos eixos  
ax.set_xlabel('Data')  
ax.set_ylabel('Variação(%)')  
ax.set_zlabel('Cotação (Venda - R$)')  
  
# Adicionar título  
plt.title('Variação da Cotação do Dólar (Agosto/2022)')  
  
# Mostrar o gráfico  
plt.show()
```

Questão 08

```
# Variáveis x e y  
x = [1, 2, 3, 4, 5]  
y = [1, 1, 2, 2, 4]  
  
# Médias de x e y  
mean_x = sum(x) / len(x)  
mean_y = sum(y) / len(y)  
  
# Calculando as somas dos produtos e somas dos quadrados  
sum_xy = 0  
sum_x_squared = 0  
sum_y_squared = 0  
  
for i in range(len(x)):  
    sum_xy += (x[i] - mean_x) * (y[i] - mean_y)  
    sum_x_squared += (x[i] - mean_x) ** 2  
    sum_y_squared += (y[i] - mean_y) ** 2  
  
# Calculando o coeficiente de correlação de Pearson  
correlation = sum_xy / (pow(sum_x_squared, 0.5) * pow(sum_y_squared, 0.5))  
  
print(f"Coeficiente de correlação de Pearson é:", correlation)
```

Questão 09

```
# Variáveis x e y
x = [1, 2, 3, 4, 5]
y = [1, 1, 2, 2, 4]

# Calculando as somas do produto xy e dos quadrados de x, bem como o tamanho do conjunto
sum_xy = 0
sum_x_squared = 0

for i in range(len(x)):
    sum_xy += x[i] * y[i]
    sum_x_squared += x[i] ** 2
n=len(x)

# Calculando os coeficientes a e b
a=(n*sum_xy-sum(x)*sum(y))/(n*sum_x_squared-(sum(x))**2)
b=(sum(y)-a*sum(x))/n

# Determinação da reta de regressão
print("Valor do coeficiente 'a':", a)
print("Valor do coeficiente 'b':", b)
print(f"A reta de regressão é: y={a}x+({b})")

# Determinação do valor estimado para y quando o usuário entra com um valor para x
x=float(input("Digite um valor para a variável x: "))
yestimado=a*x+b
print(f"O valor estimado para y quando x é igual a {x} vale {yestimado}")
```

Questão 10

```
import numpy as np

# Variáveis x e y
x =np.array([1, 2, 3, 4, 5])
y =np.array([1, 1, 2, 2, 4])

# Calculando as somas do produto xy e dos quadrados de x, bem como o tamanho do conjunto
sum_xy = 0
sum_x_squared = 0
yestimado=[]

for i in range(len(x)):
    sum_xy += x[i] * y[i]
    sum_x_squared += x[i] ** 2
n=len(x)

# Calculando os coeficientes a e b
a=(n*sum_xy-sum(x)*sum(y))/(n*sum_x_squared-(sum(x))**2)
```

```
b=(sum(y)-a*sum(x))/n

# Determinação da reta de regressão
print("Valor do coeficiente 'a':", a)
print("Valor do coeficiente 'b':", b)
print(f"A reta de regressão é: y={a}x+({b})")

# Determinação do valor médio para os dados de y's
ymedio=sum(y)/len(y)

# Determinação do valor estimado para y

yestimado=a*x+b

# Cálculo do coeficiente de determinação R2
variacaoexplicada=sum((yestimado-ymedio)**2)
variacaototal=sum((y-ymedio)**2)
R2=variacaoexplicada/variacaototal
print(f"O valor do coeficiente de determinacao vale {R2}")
```

Questão 11

```
import numpy as np

# Variáveis x e y
x =np.array([8277, 7672, 8567, 9714, 43510])
y =np.array([80.211, 39.009, 86.956, 114.606, 411.926])

# Calculando as somas do produto xy e dos quadrados de x, bem como o tamanho do conjunto
sum_xy = 0
sum_x_squared = 0
yestimado=[]

for i in range(len(x)):
    sum_xy += x[i] * y[i]
    sum_x_squared += x[i] ** 2
n=len(x)

# Calculando os coeficientes a e b
a=(n*sum_xy-sum(x)*sum(y))/(n*sum_x_squared-(sum(x)**2))
b=(sum(y)-a*sum(x))/n

# Determinação do valor médio para os dados de y's
ymedio=sum(y)/len(y)

# Determinação do valor estimado para y

yestimado=a*x+b

# Cálculo do coeficiente de determinação R2
variacaoexplicada=sum((yestimado-ymedio)**2)
variacaototal=sum((y-ymedio)**2)
R2=variacaoexplicada/variacaototal
print(f"O valor do coeficiente de determinacao vale {R2}")
```

Questão 13

```
import numpy as np
import matplotlib.pyplot as plt

digitos=np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])
frequencia=np.array([148, 75, 55, 50, 45, 40, 34, 28, 25])

tamanho_array=len(digitos)
n=sum(frequencia)

soma_prod = 0

for i in range(tamanho_array):
    soma_prod += digitos[i] *frequencia[i]

media=soma_prod/n
print("O valor do dígito médio é aproximadamente igual a:", round(media))

plt.bar(digitos, frequencia, color='green')
plt.xlabel('Dígito')
plt.ylabel('Frequência')
plt.xticks(digitos)
plt.show()
```

Questão 14

```
import numpy as np
import matplotlib.pyplot as plt

# Inicialização de Variáveis: Preço, Frequência e Retorno Diário
preco =np.array([50, 20.5328, 8.8642, 4.023, 1.9194, 0.9627, 0.5076, 0.2814, 0.164,
0.1005, 0.0647, 0.0438, 0.0312, 0.0233, 0.0183, 0.0151, 0.0131, 0.012, 0.0115,
0.0116, 0.0123, 0.0137])
frequencia =np.array([11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 21, 22, 23, 24, 24, 25,
25, 25, 25, 25])
retorno=[]

tamanho=np.size(preco)

for t in range(1,tamanho):
    retorno.append(np.log(preco[t]/preco[t-1]))

plt.bar(retorno, frequencia, width=0.1)
plt.xlabel('Retorno Diário')
plt.ylabel('Frequência')
plt.title("Distribuição de Frequência - Retorno Diário de um Ativo Financeiro")

plt.show()
```