Análise de dados com Python

Parceria:

Créditos

SUMÁRIO

Mãos à obra!

Introdução ao Python - Variáveis

1. Abra um terminal. Digite python, para entrar no modo idle.

2. Atribua os seguintes valores à variáveis, um a um:

a. 347

b. 2.71

c. “347”

d. 2+3j

3. Quais os tipos das variáveis acima?

4. Faça uma atribuição múltipla das variáveis do exercício 2.

5. Declare a seguinte variável e verifique o que acontece:

a. teste = true

6. Transforme as variáveis do exercício 2 conforme segue:

a. para float

b. para inteiro

c. para float

d. para string

7. Crie uma variável complexa com os valores de 2.a e 2.b.

Introdução ao Python - Operadores

1. Abra um terminal. Digite python, para entrar no modo idle.

2. Defina as seguintes variáveis x = 3, y = 4.0, z = 12 e t = ‘banana’

3. Calcule:

a. x + y

b. z - y

c. y ** x

d. x * z

e. z / x

f. y / x

g. y % x

4. Teste as seguintes relações

a. x > y

b. x *y == z

c. t == y

d. x < y and x >z

Introdução ao Python - Operadores com strings

1. Abra um terminal. Digite python, para entrar no modo idle.

2. Crie uma variável com seu nome completo.

3. Escreva a variável em lowercase.

4. Escreva em uppercase.

5. Verifique se o nome começa com a letra P.

6. Verifique se o nome termina com a letra J.

7. Fatie a string para apenas o 1o nome.

8. Fatie a string de 2 em 2.

9. Considerando os espaços, qual o tamanho do seu nome?

Introdução ao Python - Listas

1. Abra um terminal. Digite python, para entrar no modo idle.

2. Crie uma lista com nomes de 4 times de futebol.

3. Acesse o time que está na 3a posição.

4. Crie uma nova lista com duas listas de 3 times de futebol, cada uma de uma divisão
diferente.

5. Crie uma lista com 3 diferentes moedas. Acrescente mais 2 outras moedas à essa
mesma lista.

6. Crie uma string com a lista do exercício anterior.

7. Agora utilize a string do exercício 6 para recriar uma lista.

Introdução ao Python - Dicionários

1. Abra um terminal. Digite python, para entrar no modo idle.

2. Crie um dicionário chamado cardapio em que as chaves são os dias da semana e os respectivos valores sejam os pratos do dia.

3. Crie um dicionário chamado hemograma que contenha as seguintes chave:valor:

- *hemacias: 4.71*
- *hemoglobina: 14.1*
- *hematocrito: 41.2*
- *linfocitos: 38*
- *monocitos: 7*
- *resultado: saudável*

4. Corrija o dicionário acima com monocitos:12.

Introdução ao Python - Leitura e Escrita

1. Abra um bloco de notas e crie um código chamado imc.py que:

a. Pergunte o nome da pessoa e atribua na variável nome.

b. Pergunte a altura (em metros) e atribua na variável alt. Não esqueça de que a variável deve ser do tipo float.

c. Pergunte o peso (em quilos) e atribua na variável kg. Não esqueça de que a variável deve ser do tipo float.

d. Calcule o IMC através da fórmula IMC = peso/(alt*alt)

e. Escreva o resultado do cálculo do IMC como “Olá <Fulano>, seu IMC é <xx>”, em que <Fulano> seja o nome da pessoa e <xx> seja o valor do IMC.

Introdução ao Python - Condicionais

1. Atualize o programa imc.py feito anteriormente para que o resultado exibido seja

“Olá <Fulano>, seu IMC é <xx>, logo você está <situação>.” em que a <situação> segue as condições abaixo:

Resultado	Situação
Abaixo de 17	Muito abaixo do *peso*
Entre 17 e 18,49	Abaixo do *peso*
Entre 18,50 e 24,99	*Peso* normal
Entre 25 e 29,99	Acima do *peso*
Entre 30 e 34,99	*Obesidade* I
Entre 35 e 39,99	*Obesidade* II (severa)
Acima de 40	*Obesidade* III (mórbida)

Introdução ao Python - Loops: for

1. CAbra um bloco de notas e crie um programa chamado tabuada.py que faça:

a. Declare uma lista multiplos = [1,2,3,4,5,6,7,8,9,10]

b. Peça ao usuário um número inteiro de 1 a 10 e atribua na variável number.

c. Faça um laço for que imprima os valores da tabuada de number.

Introdução ao Python - Loops: while

1. Abra um bloco de notas e crie um programa chamado fatorial.py que:

a. Peça um número inteiro entre 2 e 15 ao usuário e atribua na variável valor.

b. Crie uma variável fat e atribua um valor inicial igual a zero.

c. Crie um contador cont e atribua um valor inicial igual a zero.

d. Usando um loop while, enquanto cont for menor que valor, atualize fat como fat = fat x valor

e. Quando o loop terminar, imprima “O fatorial de <valor> é <fat>”, em que <valor> é o número dadblo usuário e <bfat> seja o resultado do loop.

Numpy - Fatiamento de arrays

1. Abra o Jupyter lab

2. Crie um array data com a seguinte lista: [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]

3. Selecione apenas os dados diferentes de 4,6,8,10. Aloque em um array chamado data_2.

4. Crie um array dim2 que tenha 2 dimensões e a lista [1,2,3,4,5] na primeira linha e a lista [6,7,8,9,10] na 2a linha.

5. Crie um array chamado dim2_2 que seja uma fatia de dim2 e que contenha apenas os valores 3,4,8,9.

Numpy - Funções úteis

1. Abra o Jupyter lab

2. Crie um array data com a seguinte lista: [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]

3. Verifique a dimensão, a quantidade de elementos e a forma desse array.

4. Crie um novo array chamado data2, com 4x4 elementos, aplicando a função reshape no array data.

5. Verifique a dimensão, a quantidade de elementos e a forma desse array.

6. Crie um novo array chamado data_t, que seja a transposição do array data2.

7. Use a função np.flatten() e crie o array data_flat.

8. Compare o array data com o array data_flat.

Numpy - Operações com arrays

1. Abra o Jupyter lab

2. Crie 5 arrays a, b, c, d e e com as seguintes listas:

a. [7]

b. [[4,6,8],[3,5,7]]

c. [3,3,3]

d. [[3,2,1],[1,2,3]]

e. [5,10]

3. Faça as seguintes operações:

a. a * b

b. d - b

c. c + b

d. c + e

4. O que aconteceu no item d) do último exercício? Explique.

Numpy - Funções e métodos matemáticos

1. Abra o Jupyter lab

2. Crie 2 arrays a e b com as seguintes listas:

a. [[4,6,8],[3,5,7]]

b. [[11,13,17],[23,29,31]]

3. Use a função cbrt() e calcule a raiz cúbica dos arrays a e b.

4. Calcule a soma cumulativa de a.

5. Calcule a média de a e b.

6. Aplique a função np.negative() no array b e soma com b. O que aconteceu?

Numpy - Concatenação

6. Abra o Jupyter lab

7. Crie 2 arrays arrays com as seguintes listas:

a. [[1,2,3],[4,5,6]]

b. [[2,4,6,8],[10,12,14,16]]

c. [[11,13,17],[23,29,31]]

d. [[13,14,5],[19,21,23]]

8. Concatene a e b. O que aconteceu?

9. Concatene b e c, tanto usando o eixo 0 (linha) quanto o eixo 1 (coluna).

10. Concatene a e d, tanto usando o eixo 0 (linha) quanto o eixo 1 (coluna).

Pandas - Manipulando Dataframes

1. Abra o arquivo world_happiness_report_2015.csv e o aloque em um dataframe.

2. Verifique o cabeçalho e o final do dataframe.

3. Quais as colunas desse dataframe?

4. Quais os tipos de dados temos no dataframe?

5. Há valores faltantes ou nulos? Em quais colunas?

6. Renomeie as variáveis como segue:

happiness rank	=>	rank_felicidade
happiness score	=>	score_felicidade
standard error	=>	stand_error
economy (GDP per Capita	=>	PIB
health (Life Expectancy)	=>	expect_vida
trust (Government Corruption)	=>	corrupcao

7. Quais os valores médios de expect_vida? E o valor mediano? E o máximo da
variável PIB?

8. Crie uma series que contenha a altura de 5 colegas e deixe seus nomes como
índice.

Pandas - loc e iloc

1. Selecione apenas os dados de country, region, family e freedom (usando loc)

2. Selecione apenas os dados de country, region, family e freedom (usando iloc)

3. Selecione apenas as primeiras 15 linhas de country e PIB (usando loc)

4. Selecione apenas as primeiras 15 linhas de country e PIB (usando iloc)

5. Selecione apenas os dados cujo score_felicidade seja maior que 5.

6. Selecione apenas os dados que sejam da Southern Asia.

Pandas - Operações com Dataframes

1. Qual a média do score_felicidade?

2. Qual a soma do PIB?

3. Qual a soma do freedom e corrupcao?

4. Há dados duplicados? Quantos? Verifique quais são eles.

5. Verifique a quantidade de dados faltantes.

6. Crie um novo dataframe onde os valores faltantes de score_ felicidade sejam substituídos por -9999.

7. Quantas e quais são regions existentes nos dados?

8. Verifique a frequência dos dados segundo suas regiões. Qual a região com maior quantidade de dados? E a região com a menor quantidade?

Pandas - merge, concat

1. Crie um dataframe para cada um dos arquivos nba_2015_a.csv, nba_2015_b.csv, nba_2015_c.csv, e bust_nba_2015.csv. Chame esses dataframes de df_a, df_b, df_c, bust, respectivamente.

2. Visualize o head() de cada um dos dataframes.

3. Concatene os arquivos df_a, df_b e df_c usando a função concat() usando os índices. Salve um dataframe chamado df_total.

4. Faça a concatenação do dataframe df_total com o dataframe bust utilizando a função merge() e a variável ID.

5. Busque a documentação das funções concat() e merge() e veja que outros parâmetros podem ser utilizados.

Pandas - groupby

1. Abra o arquivo preferencias.csv como um dataframe chamado pref.

2. Visualize os 5 primeiras linhas do arquivo.

3. Agrupe os dados pela variável Gender (gênero).

4. Verifique a contagem de itens por cada gênero.

5. Agrupe os dados pelas variáveis Gender e Favorite Color (cor favorita).

6. Quantos itens de gênero F também possuem cor favorita Cool?

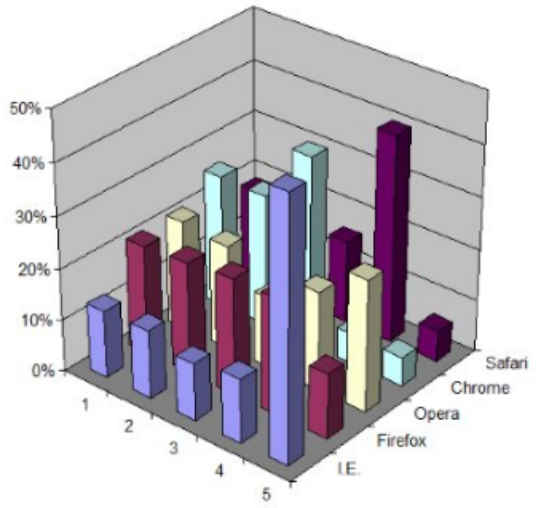
7. Agrupe os dados pelas variáveis Gender e Favorite Color e Favorite Beverage (bebida favorita).

8. Verifique a quantidade de itens de gênero M que têm cor preferida Warm e que preferem Beer.

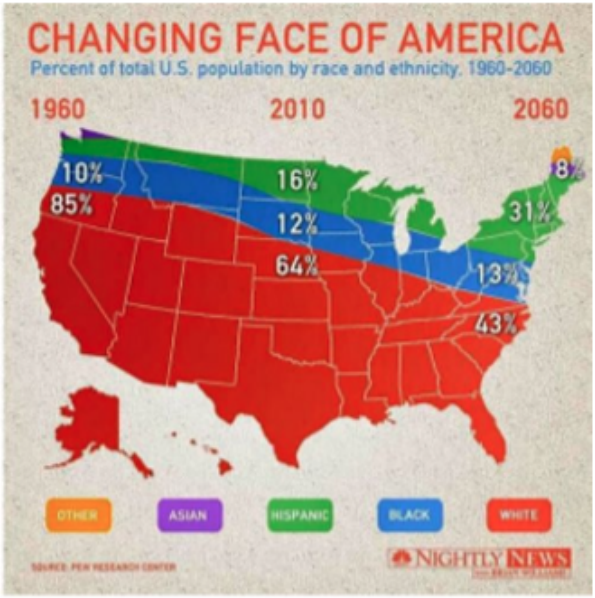
Visualização de Dados - Information Design

1. Como exibir as informações dos gráficos abaixo de maneira mais eficiente? Faça, usando apenas papel e caneta, um esboço de como os dados poderiam ser melhor apresentados.

a)



b)



Visualização de Dados - Matplotlib

1. Crie dois arrays x e y com as listas [-3,-2.5,-2,-1,0,1,2,2.5,3] e [-27,-15.62,-8,-1,0,1,8,15.62,27].

2. Use a função plot() e faça um gráfico de x e y.

3. Use a função xlabel() para adicionar o nome “Valor de X” ao eixo x. Deixe com fonte de tamanho 15.

4. Use a função ylabel() para adicionar o nome “Valor de Y” ao eixo y. Deixe com fonte de tamanho 15.

5. Use a função legend() para adicionar o nome “Função Cúbica” ao gráfico. Deixe com fonte de tamanho 20.

6. Rode novamente seu gráfico, mas mude agora a linha para a cor magenta.

Visualização de Dados - Seaborn

1. Importe a biblioteca Seaborn

2. Use a função load_dataset() e coloque o dataset exercises num dataframe chamado exercicios.

3. Use a função set_style() e mude para ‘dark’.

4. Faça o histograma da variável pulse usando a função distplot() e o argumento kde=False.

5. Agora faça a distribuição de densidade da mesma variável, mas utilizando o argumento hist=False.

6. Use a função swarplot() e análise a relação entre as variáveis kind (x) e pulse (y).

Visualização de Dados - Interpretando gráficos

1. Utilizando os datasets do Seaborn, crie o dataframe voo com o datasets flights.

2. Faça um gráfico de barras com as variáveis year e passengers.

3. Que informação podemos extrair desse gráfico?

4. Use a função boxplot() e verifique a variável passengers.

5. O que podemos depreender desse gráfico?

6. Qual a diferença entre os resultados obtidos com o gráfico de barras e o boxplot?

7. Faça um gráfico de barras com as variáveis month e passengers.

8. Qual conclusão podemos extrair desse gráfico?

Data Mining - Limpeza dos Dados

1. Crie um dataframe com o arquivo carros.csv

2. Faça uma primeira visualização dos dados.

3. Faça o describe das variáveis numéricas.

4. Faça o describe das variáveis categóricas.

5. Verifique se há dados nulos ou faltantes. Caso haja, substitua pela média das variáveis.

6. Verifique se há dados duplicados. Caso haja, elimine esses dados duplicados do dataframe.

Data Mining - Entendendo as variáveis

1. Use o dataframe anterior.

2. Faça histogramas das variáveis numéricas.

3. Escreva pequenas conclusões sobre os histogramas.

4. Faça countplots das variáveis categóricas.

5. Escreva pequenas conclusões sobre os gráficos.

Data Mining - Alguma estatísticas

1. Qual a média da variável losses? Qual a mediana da variável highway mpg?

2. Qual a média da variável price? Qual a mediana da variável price? Discuta os valores encontrados.

3. Qual a moda da variável fuel type? Qual a moda da variável make?

4. Calcule a moda das variáveis horsepower e price.

5. Escolha 3 variáveis numéricas e faça seus boxplots. Que informações é possível obter?

Data Mining - Correlações

1. Use a função corr() e faça a correlação entre as variáveis do dataframe utilizado anteriormente.

2. Quais as variáveis com maior correlação?

3. Quais as variáveis com maior anticorrelação?

4. Escolha 2 correlações e 2 anticorrelações e tentem argumentar o porquê dessa relação.

Data Mining - Estratificação

1. Vamos analisar os dados sob a perspectiva da quantidade de portas.

2. Faça gráficos de countplot simples para as variáveis riskiness e body.

3. Agora refaça os gráficos anteriores, mas estratificando pela variável doors. (Dica: coloque a variável estratificadora no parâmetro hue.)

4. O que podemos concluir a partir desses gráficos?

5. Faça o countplot da variável aspiration estratificada pela variável doors.

6. O que podemos concluir?

7. Comparando os resultados, qual deles foi mais fácil de obter?

Data Mining - Hipóteses

1. Crie mais 2 outras hipóteses que poderiam ser feitas com base nos dados sobre carros.
2. Explique em que situações essas hipóteses poderiam ser utilizadas.
3. Cite 2 hipóteses que não poderiam ser testadas com os dados sobre carros.
4. Que outros dados seriam necessários para testá-las?

Machine Learning - Entendimento de Negócio

1. Pense em um problema a ser resolvido e defina-o de maneira clara e objetiva.
2. Qual a pergunta a ser respondida?
3. Qual sua hipótese?
4. Quais os dados necessários?
5. Como medir seus resultados? Como saber se meu modelo é bom?
6. Quem usaria os resultados obtidos?

Machine Learning - Ciclo de Processos

1. Pesquise o significado de overfitting e reescreva utilizando suas palavras.

2. Pesquise quais as melhores práticas para separação de amostra de treino e teste.

3. É razoável dividir os dados em amostra de treino e teste com os mesmos tamanhos? Por quê?

Machine Learning - Treino e Teste

1. Utilizando os datasets embutidos no Scikit-Learn crie o conjunto de dados diabetes, utilizando o argumento return_X_y=True.

2. Utilizando o dataset diabetes, crie as variáveis preditoras (X) e a variável resposta(y).

3. Procure a documentação da função train_test_split e verifique qual o significado do argumento random_state e explique qual sua importância.

4. Faça uma separação de amostra de treino e teste onde a amostra de treino contenha 65% dos dados e com random_state=42.

Machine Learning - Regressão

1. Utilize o dataset diabetes criado anteriormente. Pesquise sobre o que se trata.

2. Separe-o em X,y.

3. Faça a separação em amostra de treino e teste, deixando 30% para teste. Use random_state=42.

4. Importe o LinearRegression, crie o modelo model.

5. Treino o modelo aplicando a função fit() aos dados de treino.

6. Faça a predição, aplicando a função predict aos dados X_train, chamando o resultado de preditos.

7. Meça o valor de r2_score. O quão bom está seu modelo?

8. Faça um gráfico de espalhamento com os valores de y_train e predito

Machine Learning - Classificação

1. Utilize os datasets embutidos no Scikit-Learn crie o conjunto de dados cancer, utilizando o argumento return_X_y=True. Pesquise sobre o dataset.

2. Separe-o em X,y.

3. Faça a separação em amostra de treino e teste, deixando 30% para teste. Use random_state=42.

4. Importe o LogisticRegression, crie o modelo de classificação clf.

5. Treino o modelo aplicando a função fit() aos dados de treino.

6. Faça a predição, aplicando a função predict aos dados X_train, , chamando o resultado de predito.

7. Utilize a função de métrica classification_report. O quão bom está seu modelo?