

Projeto Estatística computacional

Desenvolvido por: Prof. Hugo J. Bravo

Instruções

Este repositório fornece exemplos utilizando as bibliotecas de estatística do Python, onde são fornecidas questões para o usuário desenvolver um programa em Python utilizando ou não o Jupiter Notebook. Cada exercício contém um arquivo em Python com a sua respectiva solução e um arquivo geral utilizando o Jupiter Notebook com a solução de todos os exercícios. O arquivo `Conceitos_Estatistica.pdf` contém informações básicas sobre estatística, diferentes distribuições e definição do valor p .

Questões

- 1) Faça a leitura do arquivo Excel `IdadeAltura.xlsx` utilizando Pandas que faça leitura da planilha chamada `IdadeAltura` e calcule:
 - a) Média do arranjo
 - b) Mediana do arranjo
 - c) Modo do arranjo
 - d) Desvio padrão do arranjo
 - e) Assimetria do arranjo
- 2) Crie uma distribuição normal com 1000 pontos que tenha media 0.5 e desvio padrão de 0.1. Gere um histograma destes dados com 20 bins e sobreponha a ele um gaussiana.
- 3) Crie uma distribuição Binomial com 1000 pontos, com $n=20$ tentativas, probabilidade com sucesso $p=0.8$ e com $n=20$ tentativas. Faça o histogram desta distribuição. Sobreponha um gráfico de densidade aos dados.
- 4) Crie uma distribuição de Poisson com 10000 pontos, com uma taxa $\mu=4$. Gere um histograma dos dados e sobreponha um gráfico de densidade aos dados.
- 5) Crie uma distribuição de Bernoulli com 1000 pontos e probabilidade de sucesso $p=0.6$. Gere um histograma destes dados e sobreponha um gráfico de densidade.
- 6) Faça a leitura do arquivo Excel `AjusteCurva.xlsx` utilizando Pandas fazendo a leitura da planilha `Curvas`. Este arquivo contém a primeira coluna valores da

variável x e as demais colunas valores das funções a, b e c que dependem da variável x.

- a) Grafique as três funções no mesmo gráfico
 - b) Grafique separadamente a função a(x) e realize ajuste de curvas com graus 1, 2 e 3.
 - c) Grafique separadamente a função b(x) e realize ajuste de curvas com graus 1, 2 e 3.
 - d) Grafique separadamente a função c(x) e realize ajuste de curvas com graus 1, 2 e 3.
 - e) Qual o grau do polinômio de cada curva?
- 7) Gere uma variável x que contenha 1000 valores aleatórios obtidos de uma distribuição normal. Calcule a variável y dada por;

$$y = 1.051 * x + \sigma$$

onde σ são valores aleatórios também gerados a partir de uma distribuição normal. Grafique a mesma em função de x. Utilize a função stats.linregress da biblioteca SciPi para realizar um ajuste linear aos dados. A partir deste ajuste determine:

- a) Coeficiente linear
- b) Coeficiente angular
- c) Valor de R²
- d) Valor p (p-value)
- e) Erro (std)
- f) Sobreponha o ajuste ao gráfico e verifique se o ajuste está condizente com os dados

- 8) Dada a equação

$$S = S_0 + v_0 t + \frac{a}{2} t^2$$

com $S_0=0.5$, $V_0=2.0$ e $a=1.5$. Crie um vetor t com 500 elementos no intervalo $0 < t < 5$ espaçado linearmente. Crie uma função chamada cinemática que aceite as variáveis t, S_0 , v_0 e a como entrada e retorne o valor de S da equação acima. Perturbe os valores de S utilizando valores aleatórios de uma distribuição normal. Utilize a rotina curve_fit da biblioteca scipy.optimize para ajustar os dados que tenha como entrada a função cinemática e os novos valores de S. Faça o gráfico do deslocamento em função do tempo e sobreponha a curva ajustada ao gráfico exibindo os valores de S_0 , v_0 e a.

- 9) Dadas as funções:

$$x = 5 * \sigma_1$$
$$y = 1.051 * x + 2.05 * \sigma_2$$

onde σ_1 e σ_2 são números aleatórios obtidos de uma distribuição normal com 1000 elementos.

- a) Grafique y em função de x
- b) Utilizando a rotina train_test_split separe a amostra em valores de treino e teste

- c) Carrega a rotina `linear_model.LinearRegression()` da biblioteca `SkLearn`.
 - d) Treine o modelo
 - e) Faça um ajuste linear desta função utilizando a rotina
 - f) Exiba os coeficientes angular e linear
 - g) Exiba o valor do erro quadrático médio e da variância
 - h) Sobreponha o ajuste ao gráfico
- 10) Faça uma regressão logística criando um vetor x com 1000 números aleatórios obtidos de uma distribuição normal e um arranjo y que contenha apenas os valores positivos de x . Carregue a função `linear_model.LogisticRegression` com $C=1e5$ e solver `lbfgs`. Ajuste o modelo, gere uma amostra de teste com 400 valores entre -4 e 4 e faça a previsão do modelo. Sobreponha a curva aos dados.
- 11) Gere duas distribuições de Poisson com $\text{loc}=18, \mu=35, \text{size}=15$ para a primeira, $\text{loc}=18, \mu=25, \text{size}=100$ para a segunda e concatene as duas. Gere outra distribuição de Poisson com $\text{loc}=18, \mu=32, \text{size}=150$ para a primeira e $\text{loc}=18, \mu=28, \text{size}=100$ para a segunda concatenando ambas. Determine o valor p (p -value) destas duas distribuições e analise os mesmos.