Universidade do Estado do Rio de Janeiro – UERJ

Campus Regional Instituto Politécnico do Estado do Rio de Janeiro - IPRJ

Curso de Graduação em Engenharia da Computação

TRABALHO 1 DE MODELOS LINEARES

Leonardo Simões

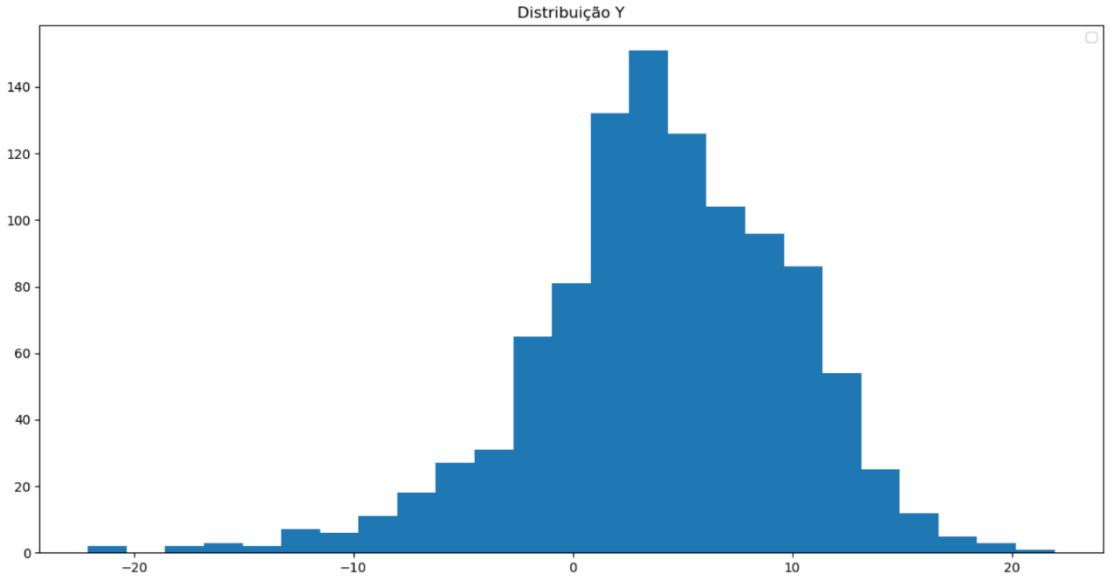
Leonardo T. Muzí de Carvalho

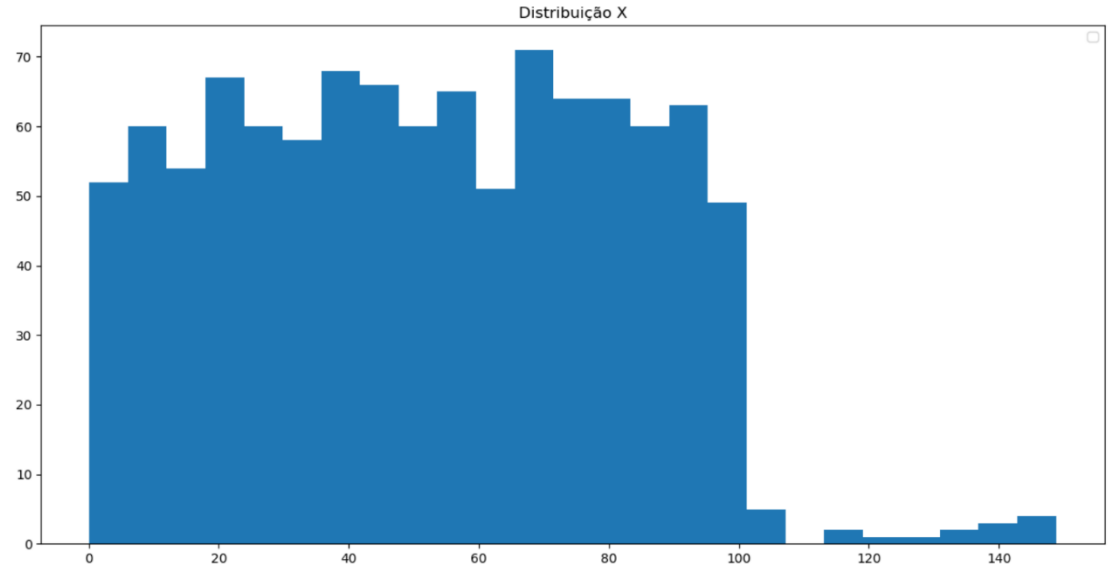
Professor:

Bernardo Sotto-Maior Peralva

Nova Friburgo, 08 de Junho de 2018.

Item A

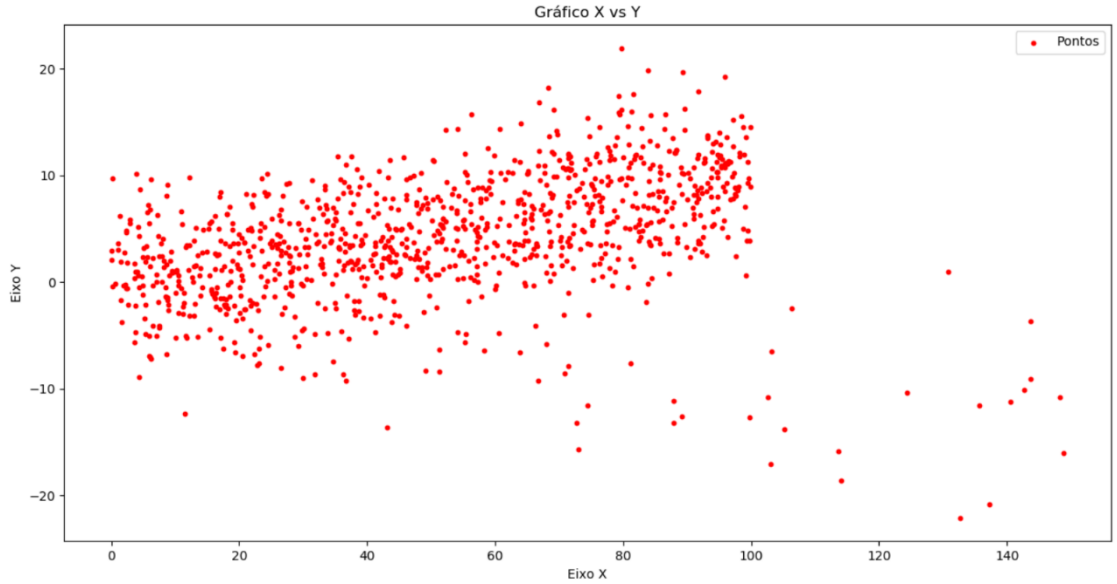




Item B

Sim, é possível identificar pontos influentes com Y < -10 e X > 100.

Item C



Visualmente é possível identificar que as variáveis X e Y estão correlacionadas positivamente.

Item D

O coeficiente de correlação calculado foi de aproximadamente 0.32, que indica uma correlação positivamente, relativamente fraca, quase moderada. O coeficiente de correlação possui este valor devido a existência de vários pontos influentes que se apresentam bem longe de onde se concentram a maior parte dos pontos.

Item E

A reta de quadrados mínimos calculada (aproximando os valores dos betas) é dada por:

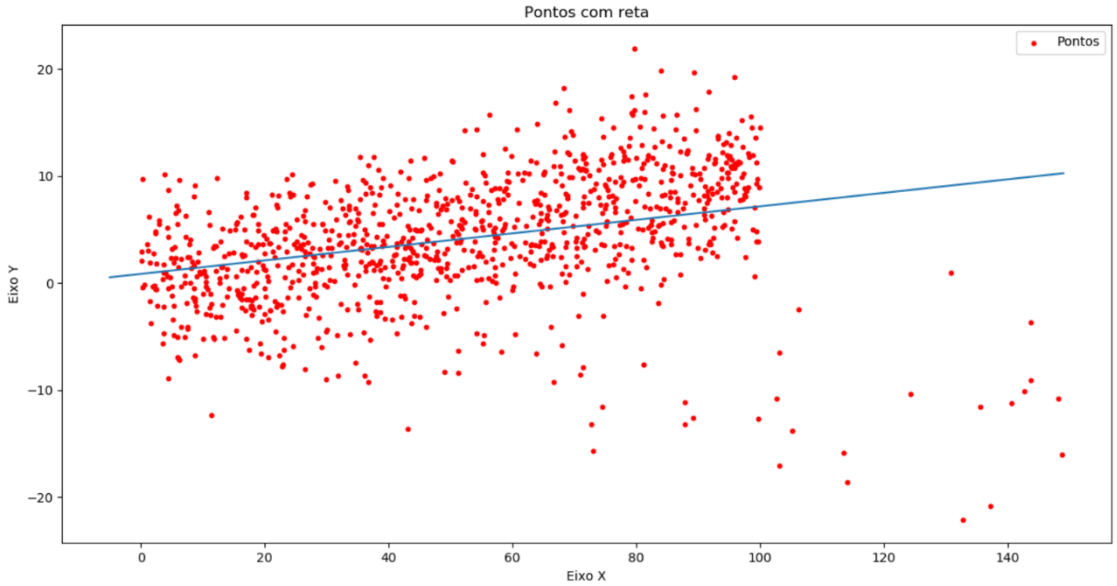
$$y_i = 0.860 + 0.063x_i$$

$$\widehat{\beta_0} = 0.860$$

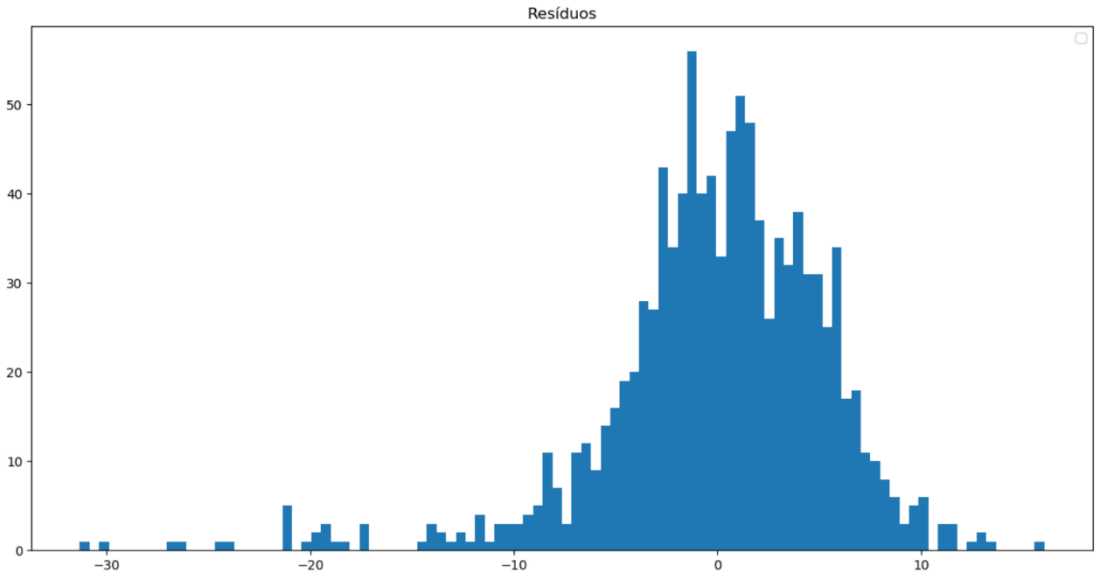
$$\widehat{\beta_1} = 0.0631$$

$$\sigma^2 = 30.622$$

Item F



Item G



Item H

Os valores de resíduos indicam que a maioria dos pontos está próximo a reta de regressão e que os pontos influentes mais longe da reta de regressão se encontram abaixo desta.

Item I

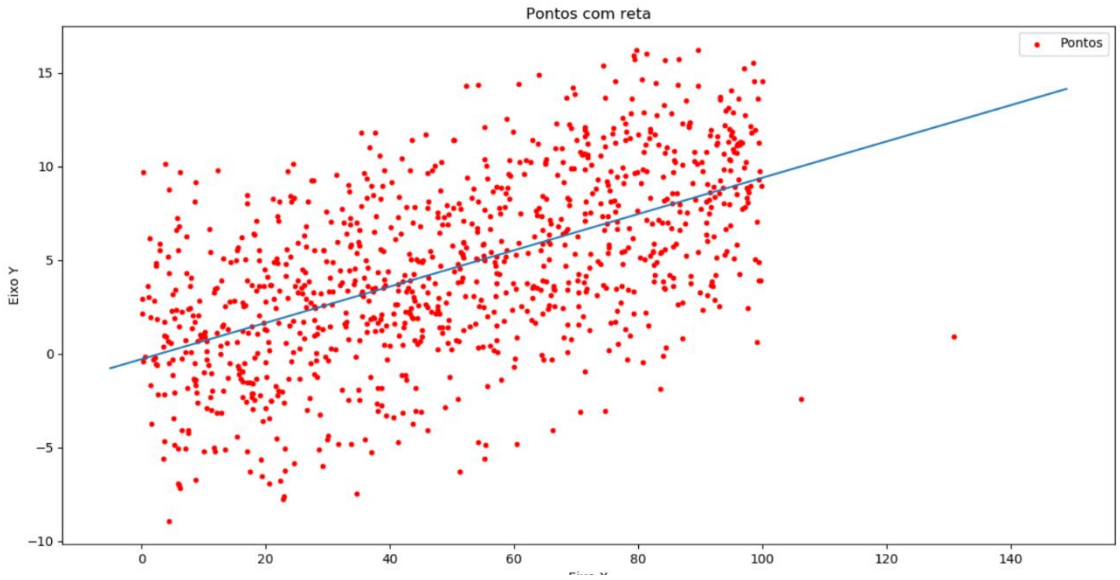
Tabela ANOVA:

Fonte (Fonte de Variação)	GL (Graus de liberdade)	SQ (Soma de quadrados)	QM (Quadrado Médio)	F_0
Regressão	1	3747.218	3747.218	122.367
Erro	1048	32092.606	30.623	
Total	1049	35839.825		

A hipótese nula é rejeitada.

Item J

1° Caso: Retirando os 50 pontos mais influentes



$$y_i = -0.290 + 0.096x_i$$

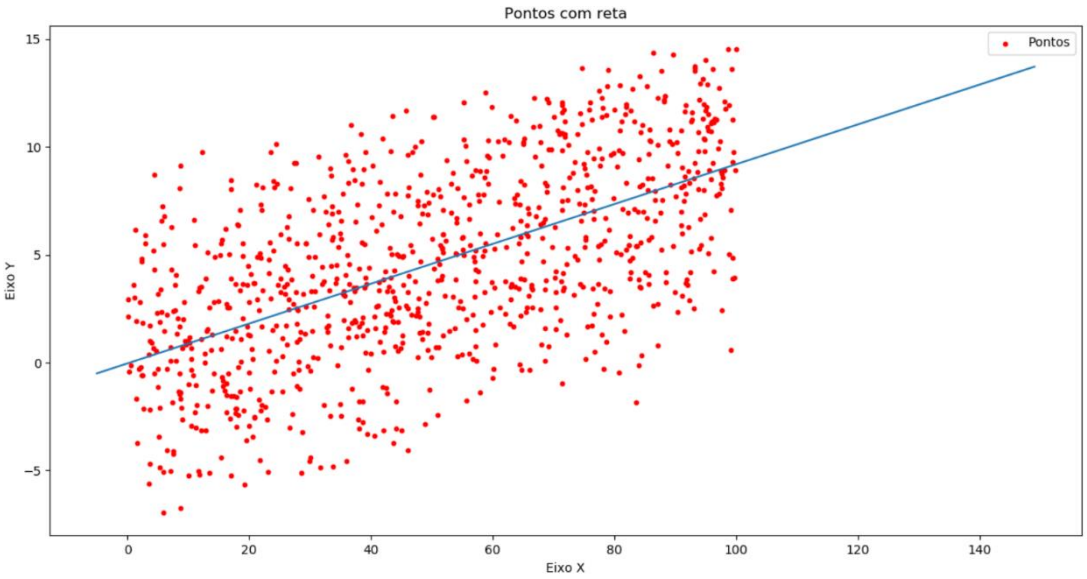
$$\widehat{\beta_0} = -0.290$$

$$\widehat{\beta_1} = 0.096$$

$$\sigma^2 = 15.961$$

$$\rho = 0.570$$

2° Caso: Retirando os 100 pontos mais influentes



$$y_i = -0.035 + 0.092x_i$$

$$\widehat{\beta_0} = -0.035$$

$$\widehat{\beta_1} = 0.092$$

$$\sigma^2 = 12.634$$

$$\rho = 0.595$$

Em ambos os casos, o coeficiente de correlação indica correlação moderada, representando um grande aumento na correlação em relação a do modelo original. Retirando os 50 pontos mais influentes representou uma grande melhora do modelo em relação ao original, visto tanto visualmente quanto pelo coeficiente de correlação. Retirando os 100 pontos mais influentes representou uma melhora muito pequena em relação ao modelo sem os 50 pontos mais influentes, de modo que o coeficiente de correlação aumentou relativamente pouco.

```
"""
Trabalho 1 de Modelos Lineares
Leonardo Simões e Leonardo Muzi
"""

import math
import matplotlib.pyplot as plt
from matplotlib import style

#Leitura de Pontos
def leArquivo(diretorio):
    amostras = []
    arquivo = open(diretorio,'r')
    for linha in arquivo:
        ponto = (float(linha.split()[0]), float(linha.split()[1]))
        amostras.append(ponto)
    arquivo.close()
    return amostras

def imprimePardePontos(pontos):
    #so para testes
    for p in pontos:
        print(p[0], " ", p[1])

#Calculos
def mediaX(pontos):
    soma = 0.0
    for p in pontos:
        soma += p[0]
    return soma/len(pontos)

def mediaY(pontos):
    soma = 0.0
    for p in pontos:
        soma += p[1]
    return soma/len(pontos)

def Sxx(pontos):
    soma = 0.0
    xm = mediaX(pontos)
    for p in pontos:
        soma += (p[0] - xm)**2
    return soma

def Syy(pontos):
    soma = 0.0
    ym = mediaY(pontos)
    for p in pontos:
        soma += (p[1] - ym)**2
    return soma

def Sxy(pontos):
    soma = 0.0
    xm = mediaX(pontos)
    ym = mediaY(pontos)
    for p in pontos:
        soma += (p[0] - xm)*(p[1] - ym)
    return soma

def coeficienteDeCorrelacao(pontos):
    return Sxy(pontos)/((Sxx(pontos)*Syy(pontos))**(1/2))

def retaQuadradosMinimos(pontos):
    B1 = Sxy(pontos)/Sxx(pontos)
    B0 = mediaY(pontos) - B1*mediaX(pontos)
    return (B0,B1)
```

```
def y(B, x):
    return B[0] + B[1]*x

def sigma2(pontos, B):
    return SQE(pontos,B)/(len(pontos)-2)

def residuo(ponto, B):
    return ponto[1] - y(B, ponto[0])

def residuos(pontos, B):
    R = []
    for p in pontos:
        #R.append(residuo(p, B)
        R.append(p[1] - y(B, p[0]))
    return R

#Tabela de ANOVA
def SQT(pontos):
    soma = 0.0
    ym = mediaY(pontos)
    for p in pontos:
        soma += (p[1]-ym)**2
    return soma

def SQE(pontos, B):
    soma = 0.0
    for p in pontos:
        soma += (p[1] - y(B, p[0])) ** 2
    return soma

def SQR(pontos, B):
    soma = 0.0
    ym = mediaY(pontos)
    for p in pontos:
        soma += (y(B, p[0]) - ym) ** 2
    return soma

def F0(pontos, B):
    return SQR(pontos, B)/(SQE(pontos, B)/(len(pontos)-2))

def testeHipoteseH0(pontos, B):
    #True aceita H0 e False rejeita H0
    if F0(pontos,B) > 3.84:
        #> distribuicao F 1,n-2,alpha
        print("Hipótese rejeitada!")
        return False
    else:
        print("Hipótese não rejeitada!")
        return True

def imprimeTabelaAnova(B, pontos):
    regressao = []
    erro = []
    total = []
    N = len(pontos)
    #calculos
    sqt = SQT(pontos)
    sqr = SQR(pontos, B)
    sqe = SQE(pontos, B)
    #fim calculos
    #primeira coluna
    regressao.append(1)
    erro.append(N-2)
    total.append(N-1)
    #segunda coluna
    regressao.append(sqr)
    erro.append(sqe)
    total.append(sqt)
```

```
    #terceira coluna
    regressao.append(sqr)
    erro.append(sqe/(N-2))
    #quarta coluna
    regressao.append(sqr*(N-2)/sqe)
    print(regressao)
    print(erro)
    print(total)


#Plote
def divideXY(pontos):
    X = []
    Y = []
    for p in pontos:
        X.append(p[0])
        Y.append(p[1])
    return (X,Y)


def plotarXY(pontos):
    (X, Y) =divideXY(pontos)
    plt.figure(1)
    plt.xlabel('Eixo X')
    plt.ylabel('Eixo Y')
    plt.title('Gráfico X vs Y')
    plt.scatter(X, Y, label= 'Pontos', color = 'r', marker = 'o', s = 10)
    plt.legend()
    plt.show()

def plotarResiduos(R):
    plt.hist(R, bins=100)
    plt.title('Resíduos')
    plt.legend()
    plt.show()

def plotarReta(B, pontos):
    (X, Y) =divideXY(pontos)
    xr = [i for i in range(-5,150)]
    yr = [y(B, x) for x in xr]
    plt.figure(1)
    plt.xlabel('Eixo X')
    plt.ylabel('Eixo Y')
    plt.title('Pontos com reta')
    plt.scatter(X, Y, label= 'Pontos', color = 'r', marker = 'o', s = 10)
    plt.plot(xr, yr)
    plt.legend()
    plt.show()

def distribuicaoVariaveis(pontos):
    (X, Y) = divideXY(pontos)
    plt.hist(X, bins=25)
    # histype= 'stepfilled'
    plt.title('Distribuição X')
    plt.legend()
    plt.show()
    plt.hist(Y, bins=25)
    plt.title('Distribuição Y')
    plt.legend()
    # histype= 'stepfilled'
    plt.show()

def removeNpontosInfluentes(pontos, n):
    B = retaQuadradosMinimos(pontos)
    pontos.sort(key=lambda p: math.fabs(residuo(p,B)),reverse=True)
    for i in range(n):
        pontos.pop(0)
```

```
def removeNpontosNaoInfluentes(pontos, n):
    B = retaQuadradosMinimos(pontos)
    pontos.sort(key=lambda p: math.fabs(residuo(p,B)),reverse=True)
    for i in range(n):
        pontos.pop(-1)

if __name__ == '__main__':
    diretorio = "D:\\data7.txt"
    amostras = leArquivo(diretorio)
    #item A
    distribuicaoVariaveis(amostras)

    #item C
    plotarXY(amostras)

    #item D
    print('R0 = ', coeficienteDeCorrelacao(amostras))

    #item E
    B = retaQuadradosMinimos(amostras)
    print("Beta = ",B)
    sig = sigma2(amostras, B)
    print("sigma² = ", sig)

    #item F
    plotarReta(B, amostras)

    #item G
    R = residuos(amostras, B)
    plotarResiduos(R)

    #item I
    imprimeTabelaAnova(B, amostras)
    print('F0 = ',  F0(amostras, B))
    print("F1 = ", 3.84)
    testeHipoteseH0(amostras,B)

    #item J
    N = 50
    print('\nRetirando ', N, ' pontos influentes')
    removeNpontosInfluentes(amostras, N)
    print("repetindo itens E e F sem pontos influentes...")
    print('R0 = ', coeficienteDeCorrelacao(amostras))

    #E
    B = retaQuadradosMinimos(amostras)
    print("Beta = ",B)
    sig = sigma2(amostras, B)
    print("sigma² = ", sig)

    #F
    plotarReta(B, amostras)
```