PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO
GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

INTELIGÊNCIA ARTIFICIAL

CLARIMAR JOSE COELHO

ATIVIDADE EXTRA DISCIPLINAR II

Escrever um programa em Python para implementar a regressão linear múltipla usando os dados de sêmen. Devem ser calculados: R^2, R^2 ajustado, SQT, SQR e SQE.

MARCOS RODOLFO CRUVINEL GOULART QUERINO

GOIÂNIA
2020

1. Introdução

A regressão linear é um método de análise estatística que nos permite estimar o valor de uma determinada variável resposta (variável dependente) como função de outras variáveis preditoras (variáveis independentes).

O método de estimação dos mínimos quadrados ordinários é uma técnica de otimização que busca o melhor ajuste do modelo através da minimização dos quadrados do erro da regressão. A regressão linear possui os seguintes pressupostos:

1. A relação entre a variável resposta e os preditores é linear.
2. A variável resposta possui distribuição normal. Os testes estatísticos desse método são todos baseados no pressuposto da normalidade. Caso tentemos analisar uma variável que não possui distribuição normal com esse método, os resultados não serão bons.
3. O termo do erro é não correlacionado com a variável resposta.
4. O termo do erro possui distribuição normal com média 0.
5. O termo do erro é homoscedástico, ou seja, possui variância constante em toda a sua extensão.
6. Os parâmetros populacionais são constantes, ou seja, valores fixos desconhecidos os quais serão estimados pela equação de regressão.

2. Desenvolvimento

Primeiramente, os dados de teste repassados pelo professor foram transcritos em um arquivo 'semen.csv' (em anexo). E em seguida, foram lidos pelo código (em anexo).

3. Código em *Python*

```
import scipy as sp
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.formula.api as sm

# Carregar dados do arquivo .csv para variavel 'base'
base= pd.read_csv('semen.csv') # Usamos a biblioteca 'pandas' para ler esse arquivo
```

```
base.shape # 'shape' da forma a esse arquivo, uma vez que seus dados estão numa variavel
           #(array)

# Regressao linear multipla
# 4 variaveis independentes: x1, x2, x3 e x4
x1= base.iloc[:,0].values # passa a coluna 0 para x1
x2= base.iloc[:,1].values
x3= base.iloc[:,2].values
x4= base.iloc[:,3].values
y= base.iloc[:,4].values # passa a coluna y para variavel y
# n= uma variável categórica binária
# p= erro comumente distribuído para servir de banco de dados para regressão
# size= tamanho do 'array' de dados de cada variavel
categoria= sp.random.binomial(n=1, p=.5, size=13)
erro= sp.random.normal(size=13)

# organizando os dados em cada variavel e printando para verificar
org= {"y":y, "x1":x1, "x2":x2, "x3":x3, "x4":x4, "categoria":categoria}
dados= pd.DataFrame(data= org)
print("\nDataFrame dos dados:\n", dados)

# Estatisticas descritivas e distribuição da variavel resposta
print("\nEstatísticas descritivas de y:")
print(dados['y'].describe())

# estimar o modelo e mostrar os resultados
modelo= sm.ols(formula='y~x1+x2+x3+x4+categoria', data=dados).fit()
print(modelo.summary())

# calculo de resíduos (epsilon = y - ŷ)
y_hat= modelo.predict()
residuo= y - y_hat
```

```
# calcular SQE= somatório de (y - ŷ)²
soma_y= sum(y)
soma_y_hat= sum(y_hat)
SQE= (soma_y - soma_y_hat)**2
print("\nSoma dos quadrados do erro: ", SQE)

# calcular SQT= somatorio de (y - média amostral de ŷ)²
# fazer a média amostral de ŷ
soma_y_hat= soma_y_hat/13

SQT= (soma_y - soma_y_hat)**2
print("\nSoma dos quadrados totais: ", SQT)

# calcular SQR= SQT - SQE
print("\nSoma dos quadrados da regressão: ", SQT-SQE)

# gráfico de residuos
print("\nGráfico de residuos:\n")
plt.scatter(y= residuo, x= y_hat, color= 'blue', s= 50, alpha=.6)
plt.hlines(y= 0, xmin= -10, xmax= 15, color= 'red')
plt.ylabel('$\epsilon = y - \hat{y}$ - Resíduos')
plt.xlabel('$\hat{y}$ ou $E(y)$ - Predito')
plt.show()

# obter os coeficientes da regressão
# reta de regressão entre 'y' e 'x1'
coeficientes = pd.DataFrame(modelo.params)
coeficientes.columns = ['Coeficientes de regressão']
print(coeficientes)

# gráfico de regressão
plt.scatter(x1,y)
```

```
plt.title('Reta de regressão')
plt.ylabel('$y$ - Variável Dependente')
plt.xlabel('$x1$ - Preditor')
plt.plot(x1, y, color= 'red')
```

4. Resultados obtidos

Figura 1 – Dataframe dos dados: y, x1, x2, x3 e x4

```
DataFrame dos dados:
        y   x1     x2     x3    x4  categoria
0   25.5  1.0   1.74   5.30  10.8          0
1   31.2  1.0   6.32   5.42   9.4          0
2   25.9  1.0   6.22   8.41   7.2          0
3   38.4  1.0  10.52   4.63   8.5          0
4   18.4  1.0   1.19  11.60   9.4          1
5   26.7  1.0   1.22   5.85   9.9          1
6   26.4  1.0   4.10   6.62   8.0          0
7   25.9  1.0   6.32   8.72   9.1          1
8   32.0  1.0   4.08   4.42   8.7          0
9   25.2  1.0   4.15   7.60   9.2          0
10  39.7  1.0  10.15   4.83   9.4          1
11  35.7  1.0   1.72   3.12   7.6          1
12  26.5  1.0   1.70   5.30   8.2          0

Estatísticas descritivas de y:
count    13.000000
mean     29.038462
std       6.042425
min      18.400000
25%      25.900000
50%      26.500000
75%      32.000000
max      39.700000
Name: y, dtype: float64
```

Fonte: Trabalho AED II corrigido sem sklearn em Spyder (Python 3.6)

Figura 2 – Resultados da regressão com foco nos resultados de R^2 e R^2 ajustado

OLS Regression Results

Dep. Variable:	y	R-squared:	0.973
Model:	OLS	Adj. R-squared:	0.960
Method:	Least Squares	F-statistic:	72.10
Date:	Thu, 15 Oct 2020	Prob (F-statistic):	2.60e-06
Time:	14:25:53	Log-Likelihood:	-17.830
No. Observations:	13	AIC:	45.66
Df Residuals:	8	BIC:	48.49
Df Model:	4		
Covariance Type:	nonrobust		

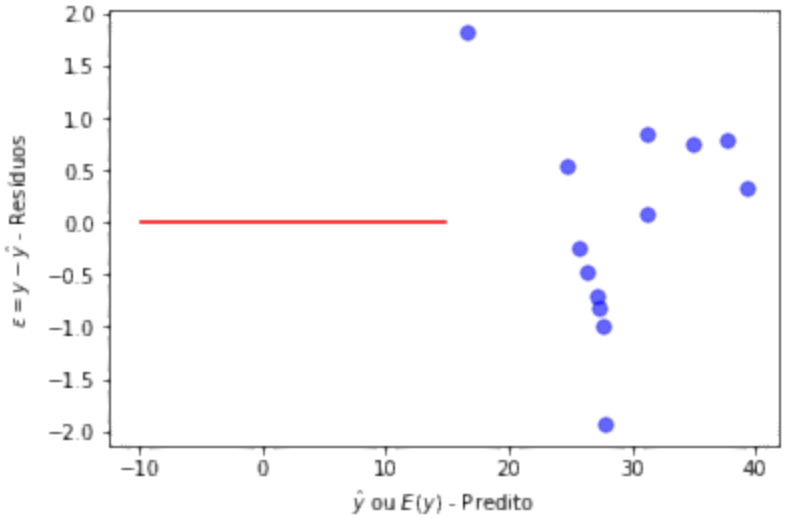
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	20.3142	1.735	11.709	0.000	16.314	24.315
x1	20.3142	1.735	11.709	0.000	16.314	24.315
x2	1.0492	0.112	9.349	0.000	0.790	1.308
x3	-1.9781	0.159	-12.430	0.000	-2.345	-1.611
x4	-0.5759	0.366	-1.573	0.154	-1.420	0.268
categoria	3.0565	0.717	4.262	0.003	1.403	4.710

Omnibus:	0.119	Durbin-Watson:	1.892
Prob(Omnibus):	0.942	Jarque-Bera (JB):	0.118
Skew:	-0.119	Prob(JB):	0.943
Kurtosis:	2.598	Cond. No.	4.43e+16

Fonte: Trabalho AED II corrigido sem sklearn em Spyder (Python 3.6)

Figura 3 – SQE, SQT, SQR e gráfico de resíduos

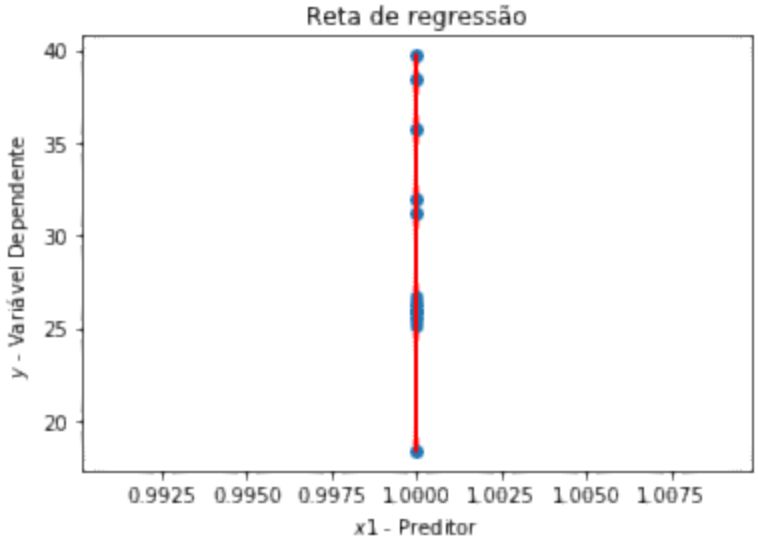
```
Soma dos quadrados do erro:  3.2311742677852644e-25

Soma dos quadrados totais:  121425.4437869822

Soma dos quadrados da regressão:  121425.4437869822

Gráfico de residuos:

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda
kurtosistest only valid for n>=20 ... continuing anyway, n=13
  "anyway, n=%i" % int(n))
```



Fonte: Trabalho AED II corrigido sem sklearn em Spyder (Python 3.6)

Figura 4 – Reta de regressão

Coeficientes de regressão	
Intercept	20.314216
x1	20.314216
x2	1.049163
x3	-1.978133
x4	-0.575850
categoria	3.056515



Fonte: Trabalho AED II corrigido sem sklearn em Spyder (Python 3.6)

5. Bibliografia

Curso de formação de cientista de dados com Python. Disponível em https://www.udemy.com. Acesso em 30 de setembro de 2020.

Entenda o que é o formato CSV e saiba como importar e exportar esses arquivos. Disponível em https://rockcontent.com/br/blog/csv/. Acesso em 01 de outubro de 2020.

Implementando Regressão Linear Simples em Python. Disponível em https://medium.com/data-hackers/implementando-regress%C3%A3o-linear-simples-em-python-91df53b920a8. Acesso em 01 de outubro de 2020.

Modelo de Regressão Linear — Mínimos Quadrados Ordinários. Disponível em https://medium.com/@caderno/regress%C3%A3o-linear-parte-1-330faf2f0010. Acesso em 01 de outubro de 2020.

NASCIMENTO, João Paulo Cândido. **OS DESAFIOS EM LIDAR COM DADOS PROBLEMÁTICOS**: UM ESTUDO EM CIÊNCIA DE DADOS SOBRE A DENGUE EM BRASÍLIA / DF. 2019. Trabalho de conclusão de curso. Faculdade de Engenharia elétrica – FEELT. Uberlândia, Minas Gerais. 2019. Disponível em: https://repositorio.ufu.br/bitstream/123456789/26678/2/DesafiosLidarDados.pdf. Acesso em 30 de setembro de 2020.

Regressão Linear Múltipla em Python. Disponível em http://artedosdados.blogspot.com/2013/09/regressao-linear-multipla-em-python.html. Acesso em 30 de setembro de 2020.

Salvar arquivo CSV no Python. Disponível em: https://www.youtube.com/watch?v=Ry8nhTMH7uo. Acesso em 01 de outubro de 2020.