

A decorative border surrounds the central text, consisting of horizontal lines with various colored wavy and zigzag patterns interspersed.

LICENCIATURA EM ENGENHARIA INFORMÁTICA

Inteligência Artificial

Regressão Linear R e Python

Jorge Ribeiro

• jribeiro@estg.ipvc.pt



1. Regressão

2. Regressão Linear

- O que é?
- Regressão vs Correlação

3. Implementação em Python

4. Implementação em R

5. Bibliografia

Regressão

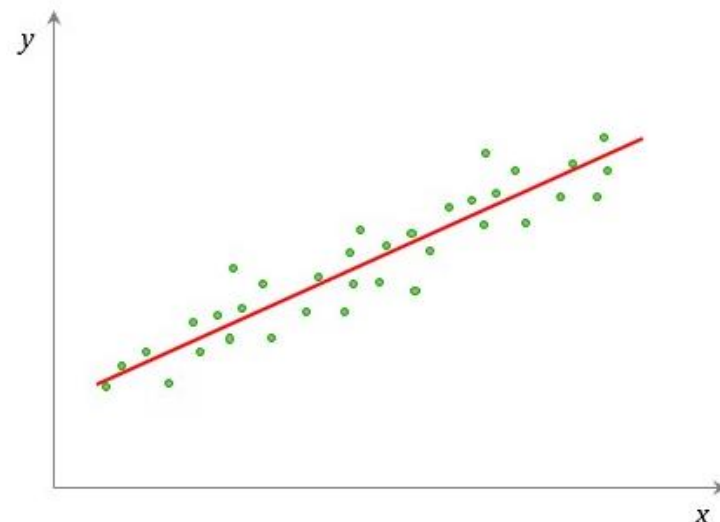
É o ato de prever, a partir de dados históricos, dados no futuro. Em Estatística, o termo a procurar é **Análise de Regressão**. Este, estuda a relação entre duas ou mais variáveis, em que uma depende de outra ou outras. Quando temos uma relação de um para um, temos uma Regressão Linear, e quando eu tenho uma relação de um para muitos, eu tenho uma Regressão Linear Múltipla.

Na Estatística, temos um conceito chamado **Correlação**. Neste podemos ter uma variável dependendo positivamente ou negativamente em relação a alguma outra. Exemplo: Podemos ter a distância percorrida, em metros, em função da velocidade de um objeto, em metros por segundo. Claro que aqui haverá uma correlação perfeita. Mas imagine-se um outro exemplo: encontrar uma correlação entre o preço de um imóvel e seu tamanho. Neste caso, podemos não ter uma correlação perfeita, mas podemos entender que há um padrão

nos dados, e que um valor depende de outro ou outros.

Isto é correlação. A **Regressão**, já é a **descoberta Dessa função, que expressa o padrão nos dados.**

Ou seja, quando se desenvolve um trabalho de Análise de Regressão, procura-se entender o padrão de um valor dependendo de outro ou outros, e assim encontrar função que expressa esse padrão.



Diferenças entre Regressão e Correlação

1. Uma medida estatística que determina a co-relação ou associação de duas quantidades é conhecida como correlação. A regressão descreve como uma variável independente está numericamente relacionada à variável dependente.
2. A correlação é usada para representar a relação linear entre duas variáveis. Pelo contrário, a regressão é usada para ajustar a melhor linha e estimar uma variável com base em outra variável.
3. Na correlação, não há diferença entre variáveis dependentes e independentes, ou seja, a correlação entre x e y é semelhante a y e x . Pelo contrário, a regressão de y em x é diferente de x em y .
4. A correlação indica a força da associação entre as variáveis. Diferentemente da regressão, ela reflete o impacto da mudança da unidade na variável independente na variável dependente.
5. A correlação visa encontrar um valor numérico que expresse a relação entre as variáveis. Diferentemente da regressão, cujo objetivo é prever os valores da variável aleatória com base nos valores da variável fixa.

Embora estes dois sejam estudados juntos. A correlação é usada quando o pesquisador deseja saber se as variáveis em estudo estão correlacionadas ou não. Na análise de regressão, é estabelecida uma relação funcional entre duas variáveis para fazer projeções futuras de eventos.

Regressão Linear

- Uma regressão só é chamada “linear” quando se considera que a relação da resposta às variáveis é tratada como uma função linear ($Ax+b$). Aqueles modelos de regressão que não podem ser traduzidos em uma função linear dos parâmetros são denominados não-lineares.
- Este tipo de regressão é uma das primeiras formas de análise regressiva a ser estudada, e é também usada extensamente em aplicações práticas. Isto só acontece porque modelos que dependem de uma forma linear de seus parâmetros desconhecidos, são mais facilmente ajustados que modelos não-lineares, além disto, as propriedades estatísticas dos estimadores resultantes são fáceis de determinar.

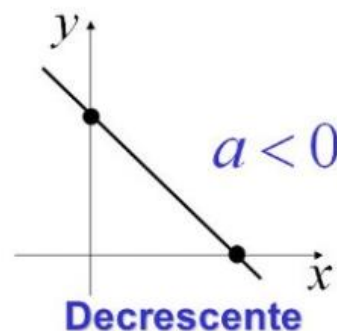
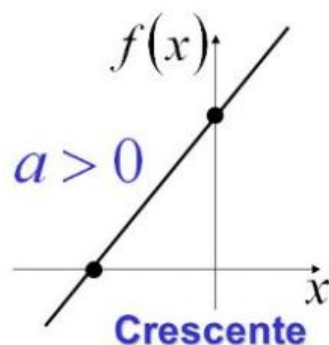
Como calcular os coeficientes numa função de primeiro grau?

Os algoritmos de Regressão Linear, em sua maioria, utilizam um método para calcular tais coeficientes: o nome dele é Método dos Mínimos Quadrados (MMQ), ou Métodos dos Quadrados Ordinários (MQO) ou ainda, em inglês, Ordinary Least Squares (OLS). Tal método, visa buscar o melhor valor que os coeficientes possam atingir, de maneira que a diferença entre o valor predito pela função e o real, sejam os menores.

Função de 1º Grau – (Reta)

$$f(x) = ax + b$$

$$y = ax + b$$



Conceitos importantes

Variável preditora:

É a variável independente, que tem o poder de influenciar na variável que nós queremos encontrar.. Na equação abaixo, ela será o valor de **a**.

Variável alvo ou dependente:

É a variável que queremos prever.. Na equação abaixo, ela terá como valor o resultado da função para cada valor de **y**.

$$f(x) = ax + b$$

$$y = ax + b$$

Avaliando o Modelo de Regressão Linear

Teste F de Significância global:

Afirma se ao menos uma variável do meu modelo está relacionada com a variável alvo. Para isso, o valor-p desta estatística precisa ser **menor que 0.05**

Teste de Significância individuais ou p-values dos coeficientes:

Diz o quanto das variáveis preditoras explicam a variável alvo. A métrica padrão é o p-value ser **menor que 0.05**.

Coeficiente R^2 :

Diz o quanto o meu modelo explica seus resultados. É um valor entre 0 e 1. Quanto mais próximo de 1, melhor.



Entendendo o cálculo de R^2

O R^2 , também chamado de Coeficiente de Determinação, diz o quanto meu modelo está prevendo corretamente. O cálculo dele, envolve três medidas:

- **Soma Total dos Quadrados (STQ):** mostra a variação de y em torno da própria média. É o somatório das diferenças entre o valor alvo real e sua média elevado ao quadrado.

$$\sum (y - \bar{y})^2$$

y é o valor real, \bar{y} é a média

- **Soma dos Quadrados dos Resíduos (SQU):** variação de Y que não é explicada pelo modelo elaborado. É o somatório das diferenças entre o valor predito e o valor real elevados ao quadrado.

$$\sum (y - \hat{y})^2$$

y é o valor real, \hat{y} é o valor predito

Importar as bibliotecas necessárias

As seguintes bibliotecas devem ser instaladas
Previamente através de comandos pip ou através
De anaconda com o comando conda

Exemplos:

Instalação da module scikit-learn

Instalação com conda:
conda install scikit-learn

Instalação com pip:
pip install -U scikit-learn scipy matplotlib

Para python3:
pip3 install -U scikit-learn scipy matplotlib

```
# importa as libs
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
import statsmodels.api as sm
import os
```

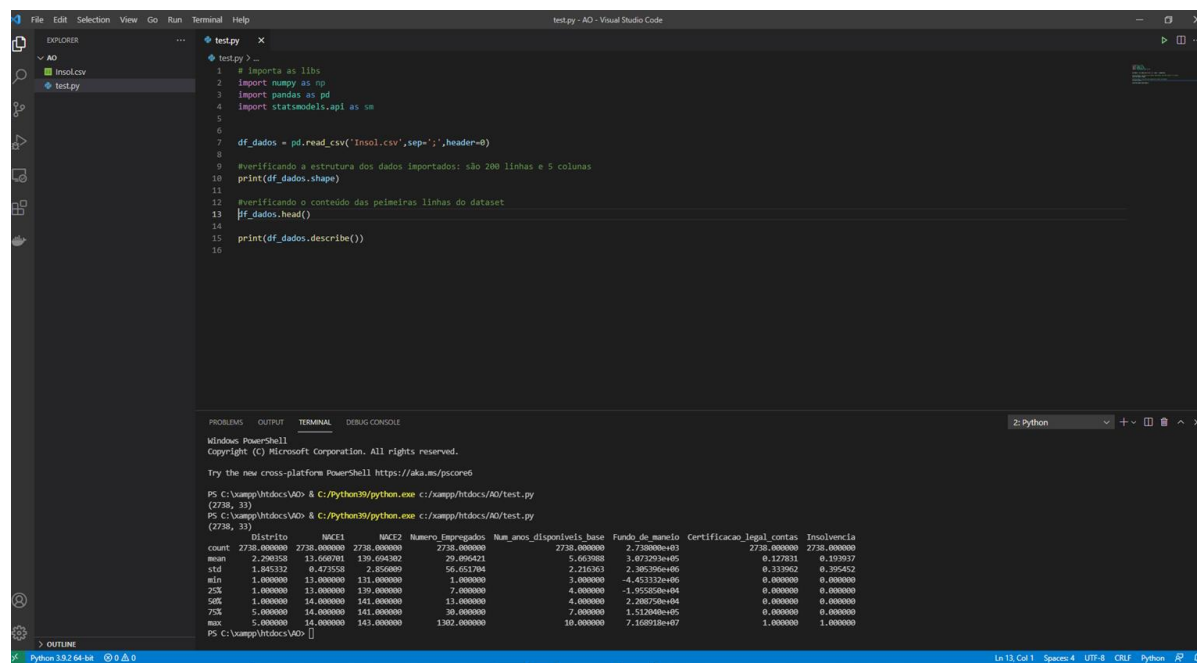
Exemplo prático usando python

Após importarmos as devidas bibliotecas e o nosso dataset utilizamos os seguintes comandos para realizar uma análise básica ao dataset para melhor entender o comportamento de cada variável, maiores e menores valores, médias , etc.

Neste caso não temos colunas vazias mas caso existam podemos utilizar um comando do tipo

```
df_dados.drop(['Unnamed: 0'],  
axis=1)
```

Para eliminar colunas não importantes sendo Unnamed o nome da coluna e axis o número da colunas e df_dados o nome dado ao dataset importado



```
testpy - x
1 # Importa as libs
2 import numpy as np
3 import pandas as pd
4 import statsmodels.api as sm
5
6
7 df_dados = pd.read_csv('Insol.csv', sep=';', header=0)
8
9 #verificando a estrutura dos dados importados: são 200 linhas e 5 colunas
10 print(df_dados.shape)
11
12 #verificando o conteúdo das primeiras linhas do dataset
13 df_dados.head()
14
15 print(df_dados.describe())
16
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell <https://aka.ms/powershell>

PS C:\xampp\htdocs\VAO> & C:\Python39\python.exe c:\xampp\htdocs\AO/test.py
(2738, 33)

PS C:\xampp\htdocs\VAO> & C:\Python39\python.exe c:\xampp\htdocs\AO/test.py
(2738, 33)

	Distrito	NACE1	NACE2	Numero Empregados	Num_anos_disponiveis_base	Fundo_de_maneio	Certificacao_legal_contas	Insolvencia
count	2738.000000	2738.000000	2738.000000	2738.000000	2738.000000	2.738000e+03	2738.000000	2738.000000
mean	2.200358	13.660701	139.694382	29.096421	5.663988	3.073293e+05	0.127811	0.193917
std	1.865112	6.471558	2.456000	56.651784	2.221061	2.309366e+05	0.333962	0.395452
min	1.000000	13.000000	131.000000	1.000000	3.000000	-4.45132e+06	0.000000	0.000000
25%	1.000000	13.000000	139.000000	7.000000	4.000000	-1.95250e+04	0.000000	0.000000
50%	1.000000	14.000000	141.000000	13.000000	4.000000	2.282750e+04	0.000000	0.000000
75%	5.000000	14.000000	141.000000	30.000000	7.000000	1.512040e+05	0.000000	0.000000
max	5.000000	14.000000	143.000000	1302.000000	10.000000	7.168918e+07	1.000000	1.000000

O que é o fundo de maneio

O fundo de maneio é o valor que a empresa necessita para desenvolver a sua atividade de forma normal e equilibrada, durante um período de tempo. De forma mais simplista, é uma espécie de almofada financeira que as empresas devem assegurar para gerar liquidez a curto prazo



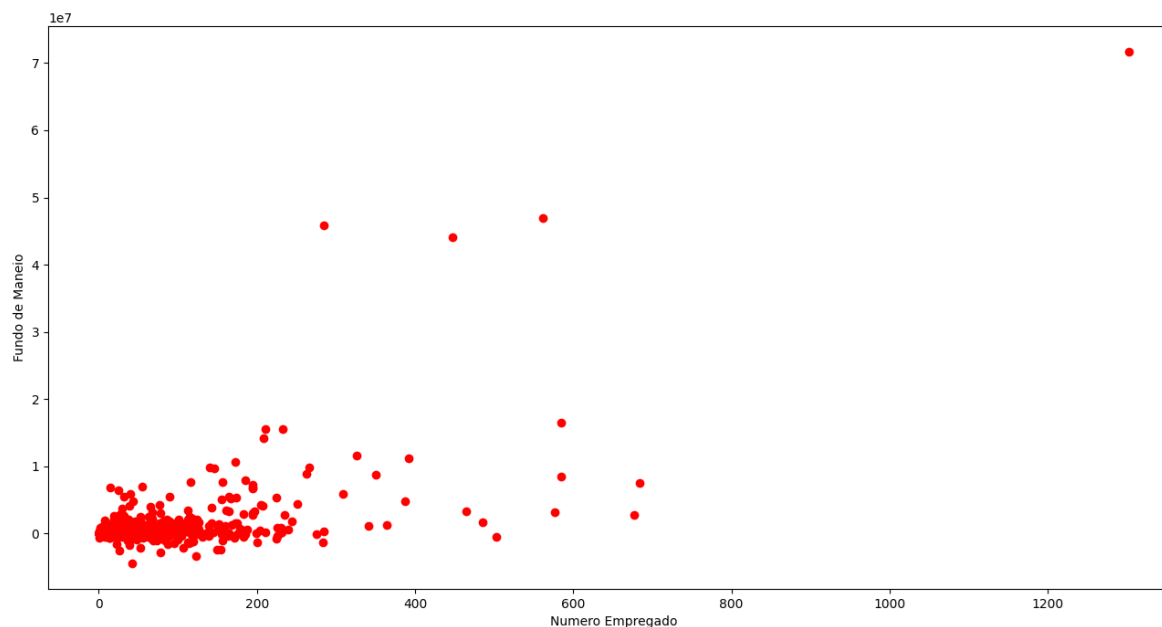
Continuação do exemplo prático

Como sabemos, a regressão linear simples tem a capacidade de usar uma variável de entrada para prever o valor de uma variável de saída, em nosso exemplo vamos utilizar o Fundo de maneio e o número de empregados. Primeiro vamos criar uma plotagem para entendermos o comportamento do fundo de maneio necessário conforme o número de Trabalhadores.

```
21
22
23 plt.figure(figsize = (16,8))
24 plt.scatter(
25     df_dados['Fundo_de_maneio'],
26     df_dados['Numero_Empregados'],
27     c='red')
28 plt.xlabel("Fundo de Maneio")
29 plt.ylabel("Numero Empregado")
30 plt.show()
```

Podemos observar que os Fundo de maio mais baixos estão atribuídos por norma a menos trabalhadores

Agora vamos criar o modelo para prever o quanto teremos de maneo conforme o número de empregados



Análise e Exploração dos Dados – Importar Insol.csv

```
X = df_dados['Numero_Empregados'].values.reshape(-1,1)
y = df_dados['Fundo_de_maneio'].values.reshape(-1,1)

reg = LinearRegression()
reg.fit(X, y)

print("O modelo é: Maneio = {:.5} + {:.5}X".format(reg.intercept_[0], reg.coef_[0][0]))
```

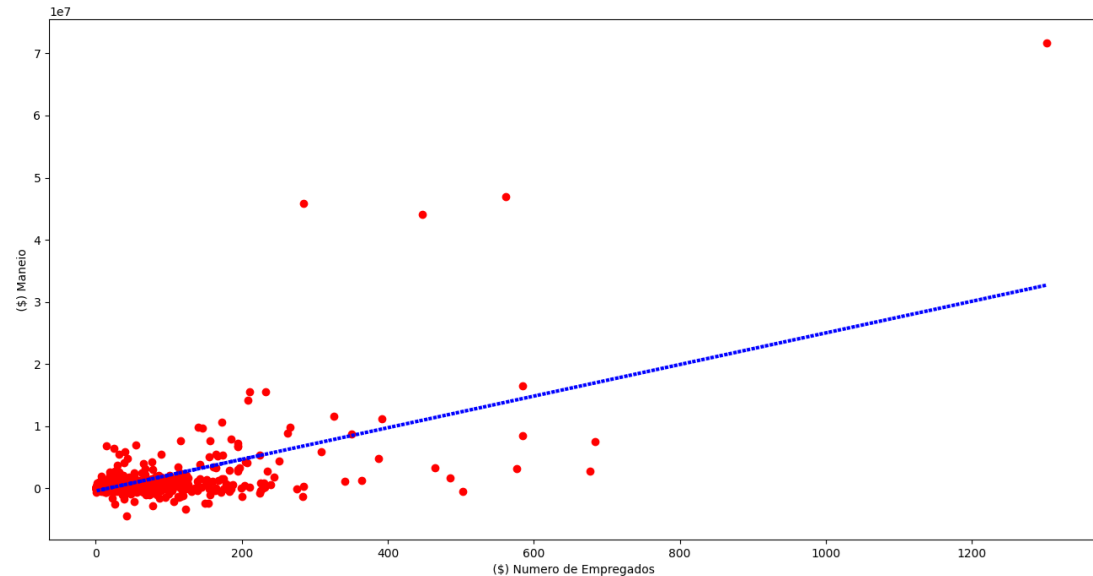
```
max 3.000000 14.000000 143.000000 1302.000000
O modelo é: Maneio = -4.3302e+05 + 2.5445e+04X
PS C:\xampp\htdocs\AO>
```

Aqui temos o nosso modelo, onde os dois primeiros valores são as constantes geradas pelo modelo, e o "X" representa o número de empregados, ou seja, essa é a conta matemática que nos ajuda a prever o maneio que vai estar atribuído a esse número.

Agora vamos plotar o nosso modelo em cima dos dados e analisar se ele é bom, mau, explica muito ou explica pouco o comportamento de nossos dados:

■ Exemplo Prático Python

```
45 f_previsaoes = reg.predict(X)
46
47
48 plt.figure(figsize = (16,8))
49 plt.scatter(
50     df_dados['Numero_Empregados'],
51     df_dados['Fundo_de_maneyo'],
52     c='red')
53
54
55 plt.plot(
56     df_dados['Numero_Empregados'],
57     f_previsaoes,
58     c='blue',
59     linewidth=3,
60     linestyle=':')
61 )
62
63 plt.xlabel(" ($) Numero de Empregados")
64 plt.ylabel(" ($) Maneio ")
65 plt.show()
66
```



Agora aplicamos a prática aprendida na teoria de avaliação dos modelos descrita anteriormente.

Criando um resumo que mostra várias características de nosso modelo. Avaliaremos sua qualidade através do R^2 e do "p-valor"

■ Exemplo Prático Python

```
67
68 X = df_dados['Numero_Empregados']
69 y = df_dados['Fundo_de maneio']
70 X2 = sm.add_constant(X)
71 est = sm.OLS(y, X2)
72 est2 = est.fit()
73 print(est2.summary())
74
```

O modelo é: Maneio = -4.3302e+05 + 2.5445e+04X

OLS Regression Results

Dep. Variable:	Fundo_de maneio	R-squared:	0.391
Model:	OLS	Adj. R-squared:	0.391
Method:	Least Squares	F-statistic:	1756.
Date:	Sat, 05 Jun 2021	Prob (F-statistic):	6.08e-297
Time:	17:21:08	Log-Likelihood:	-43319.
No. Observations:	2738	AIC:	8.664e+04
Df Residuals:	2736	BIC:	8.665e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	-4.33e+05	3.87e+04	-11.200	0.000	-5.09e+05	-3.57e+05
Numero_Empregados	2.544e+04	607.153	41.908	0.000	2.43e+04	2.66e+04

Omnibus: 4686.412 Durbin-Watson: 1.961
Prob(Omnibus): 0.000 Jarque-Bera (JB): 7114610.499
Skew: 11.370 Prob(JB): 0.00
Kurtosis: 251.689 Cond. No. 71.6

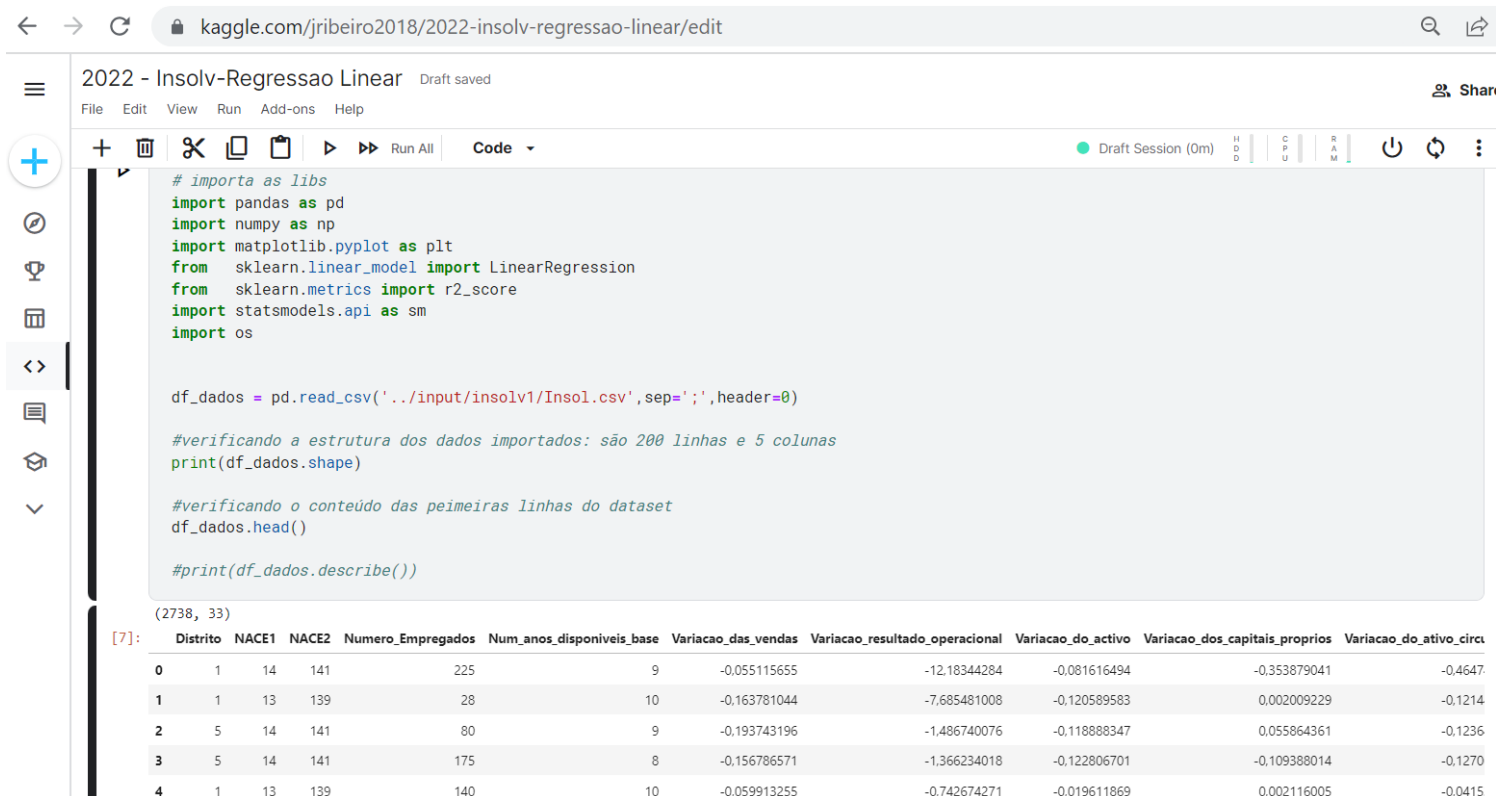
Observa-se que o R^2 está em 0.391, isso quer dizer que aproximadamente 40% do comportamento da variável "Fundo Maneio" é explicado pela variável "Número Empregados".

Quando analisamos o "p-valor" observamos que ele está bem baixo, algo próximo do zero foi encoberto pelo arredondamento, o que indica que podemos rejeitar a hipótese nula.

No entanto 40% não seria algo considerado bom em uma empresa com situações do mundo real sendo ainda preciso mais dados para alcançar uma precisão maior para esta caso específico estudado.

Isto pode ainda ser provado pela nossa F-statística muito alta que significa que existe um variância muito grande no nosso conjunto de dados

Exemplo prático usando Python Kaggle



The screenshot shows a Kaggle Notebook interface. The title is "2022 - Insolv-Regressao Linear" and it's a draft. The code cell contains the following Python code:

```
# importa as libs
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
import statsmodels.api as sm
import os

df_dados = pd.read_csv('../input/insolv1/Insol.csv', sep=';', header=0)

#verificando a estrutura dos dados importados: são 200 linhas e 5 colunas
print(df_dados.shape)

#verificando o conteúdo das primeiras linhas do dataset
df_dados.head()

#print(df_dados.describe())
```

The output shows the shape of the dataset: (2738, 33). Below the code, a preview of the data is shown as a table:

[7]:	Distrito	NACE1	NACE2	Numero_Empregados	Num_anos_disponiveis_base	Variacao_das_vendas	Variacao_resultado_operacional	Variacao_do_ativo	Variacao_dos_capitais_proprios	Variacao_do_ativo_circu
0	1	14	141	225	9	-0.055115655	-12.18344284	-0.081616494	-0.353879041	-0.4647
1	1	13	139	28	10	-0.163781044	-7.685481008	-0.120589583	0.002009229	-0.1214
2	5	14	141	80	9	-0.193743196	-1.486740076	-0.118888347	0.055864361	-0.1236
3	5	14	141	175	8	-0.156786571	-1.366234018	-0.122806701	-0.109388014	-0.1270
4	1	13	139	140	10	-0.059913255	-0.742674271	-0.019611869	0.002116005	-0.0415

Exemplo Prático usando Python com Notebook

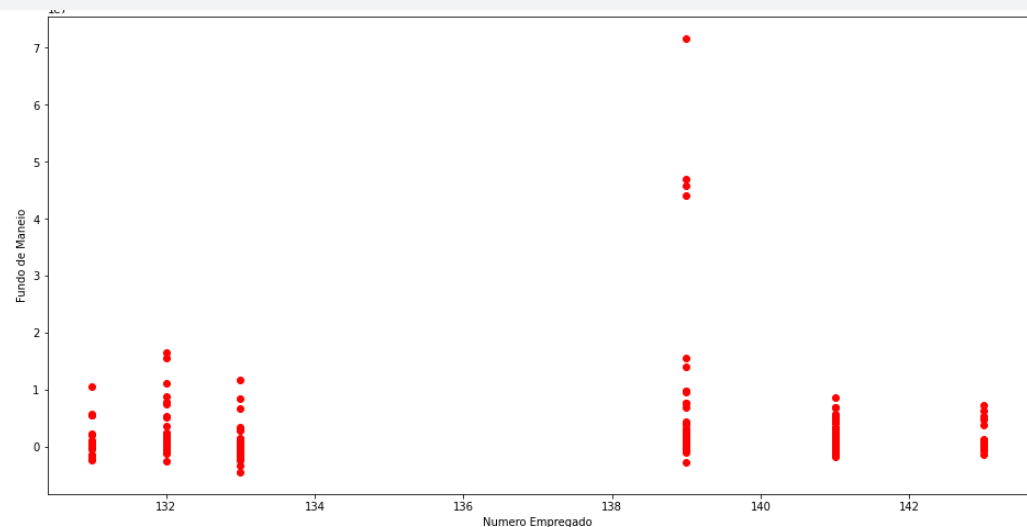
Kaggle

```
plt.figure(figsize = (16,8))
plt.scatter(

    df_dados['NACE2'],
    df_dados['Fundo_de_maneio'],
    c='red')
plt.xlabel("Numero Empregado")
plt.ylabel("Fundo de Maneio")
plt.show()

X = df_dados['Numero_Empregados'].values.reshape(-1,1)
y = df_dados['Fundo_de_maneio'].values.reshape(-1,1)

reg = LinearRegression()
reg.fit(X, y)
```



■ Exemplo Prático usando Python com Notebook

Kaggle



```
print("O modelo é: Maneio = {:.5} + {:.5}X".format(reg.intercept_[0], reg.coef_[0][0]))
f_previsoes = reg.predict(X)

plt.figure(figsize = (16,8))
plt.scatter(
    df_dados['Numero_Empregados'],
    df_dados['Fundo_de_maneyo'],
    c='red')
plt.plot(
    df_dados['Numero_Empregados'],
    f_previsoes,
    c='blue',
    linewidth=3,
    linestyle=':')

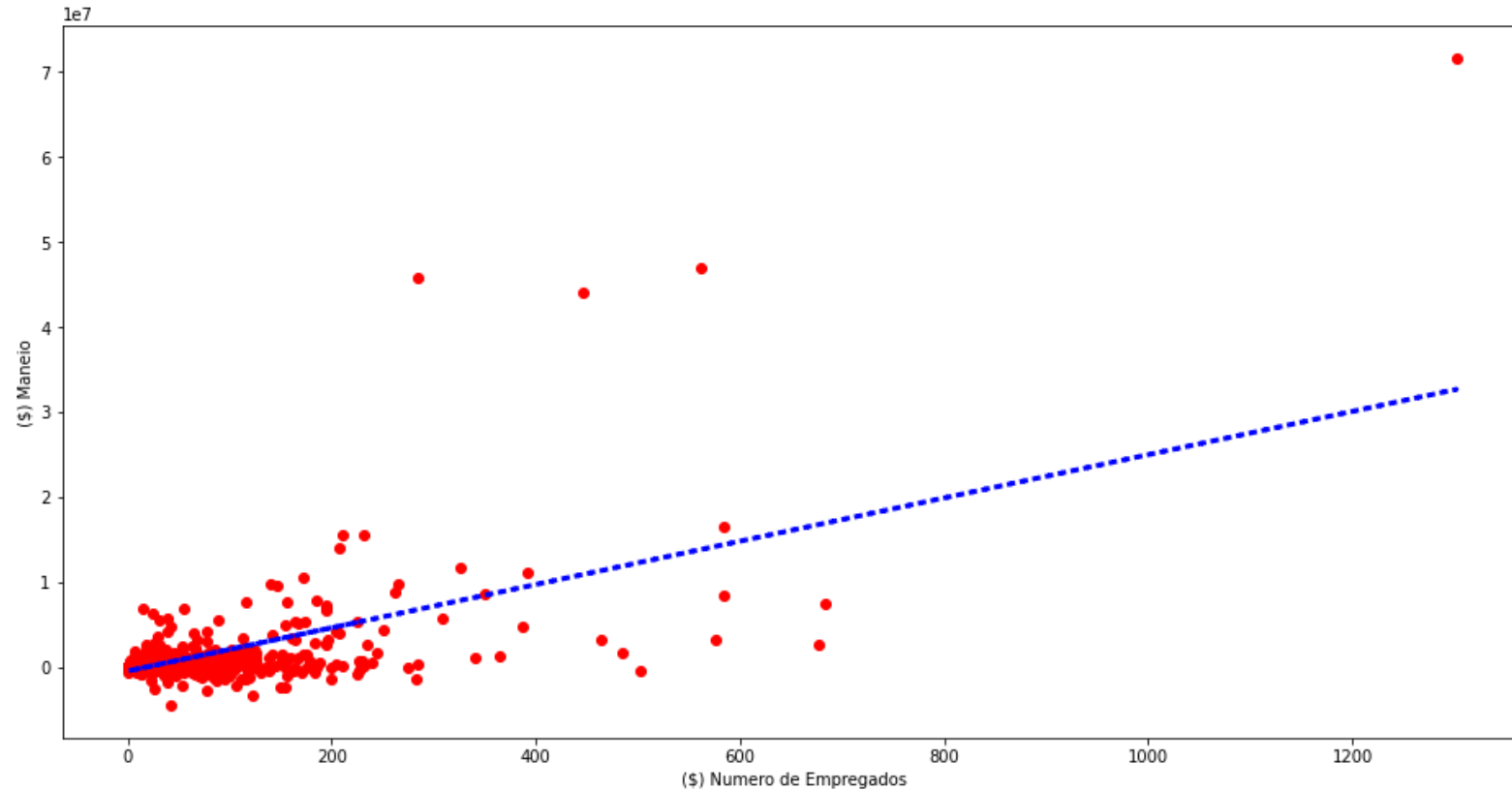
plt.xlabel(" ($) Numero de Empregados")
plt.ylabel(" ($) Maneio ")
plt.show()

X = df_dados['Numero_Empregados']
y = df_dados['Fundo_de_maneyo']
X2 = sm.add_constant(X)
est = sm.OLS(y, X2)
est2 = est.fit()
print(est2.summary())
```

Exemplo Prático usando Python com Notebook

Kaggle

O modelo é: Maneio = $-4.3302e+05 + 2.5445e+04X$



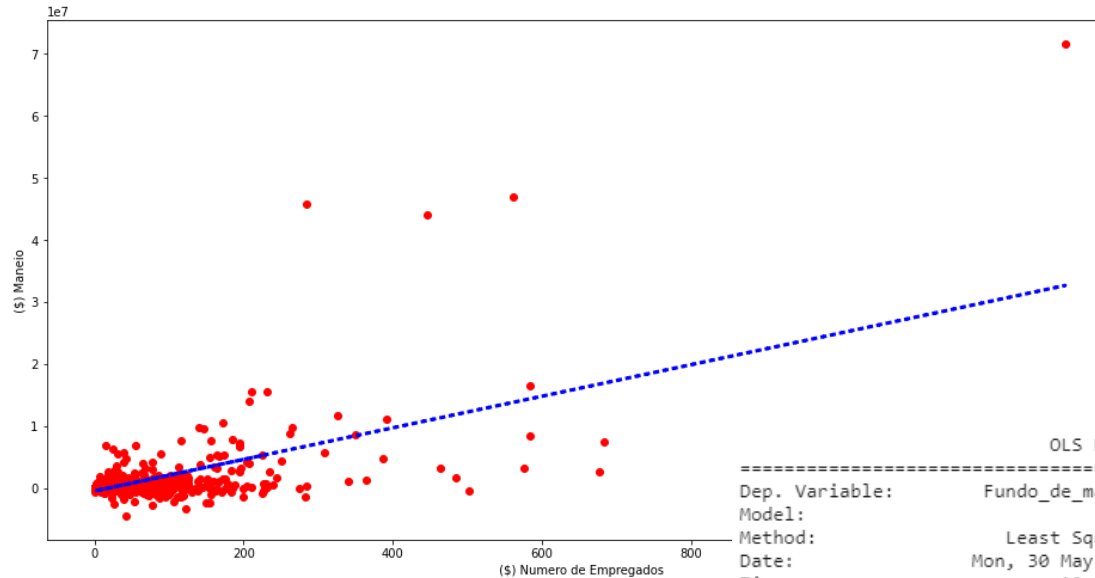
OLS Regression Results

```
=====
Dep. Variable:    Fundo_de_maneyo    R-squared:        0.391
Model:            OLS                Adj. R-squared:    0.391
=====
```

Exemplo Prático usando Python com Notebook

Kaggle

O modelo é: Maneio = $-4.3302e+05 + 2.5445e+04X$



```
=====
OLS Regression Results
=====
Dep. Variable:    Fundo_de_maneio    R-squared:        0.391
Model:            OLS                Adj. R-squared:    0.391
Method:            Least Squares      F-statistic:       1756.
Date:             Mon, 30 May 2022    Prob (F-statistic): 6.08e-297
Time:             19:16:59            Log-Likelihood:    -43319.
No. Observations: 2738                AIC:               8.664e+04
Df Residuals:      2736                BIC:               8.665e+04
Df Model:           1
Covariance Type:   nonrobust
=====
```

```
=====
OLS Regression Results
=====
Dep. Variable:    Fundo_de_maneio    R-squared:        0.391
Model:            OLS                Adj. R-squared:    0.391
Method:            Least Squares      F-statistic:       1756.
Date:             Mon, 30 May 2022    Prob (F-statistic): 6.08e-297
Time:             19:16:59            Log-Likelihood:    -43319.
No. Observations: 2738                AIC:               8.664e+04
Df Residuals:      2736                BIC:               8.665e+04
Df Model:           1
Covariance Type:   nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
const	-4.33e+05	3.87e+04	-11.200	0.000	-5.09e+05	-3.57e+05
Numero_Empregados	2.544e+04	607.153	41.908	0.000	2.43e+04	2.66e+04

```
=====
Omnibus:            4686.412    Durbin-Watson:       1.961
Prob(Omnibus):      0.000      Jarque-Bera (JB):     7114610.499
Skew:                11.370     Prob(JB):              0.00
Kurtosis:            251.689     Cond. No.              71.6
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Exemplo prático usando R

Primeiro devemos criar o nosso projeto e importar a biblioteca pacman coma ferramenta seguindo as seguntes instruções:

... e a partir daí, vamos começar a trabalhar.

STEP 1: Open RStudio and start a new project.

1. Figure 2.1 - New Project. STEP 2: Start a new script.
2. Figure 2.2 - New script. STEP 3: Execute the following command.
install.packages("pacman") STEP 4: Make sure **pacman package** is installed.
3. Figure 2.2 - Console. STEP 5: Load the **pacman package**. **library(pacman)**

05/04/2020


```
> if(!require(pacman)) install.packages("pacman")
Loading required package: pacman
> library(pacman)
>
> pacman::p_load(dplyr, ggplot2, car, rstatix, lme4, ggpubr)
Installing Package into 'C:/Users/Hugo/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
also installing the dependencies 'fansi', 'pkgconfig', 'purrr', 'cli', 'crayon', 'utf8', 'ellipsis', 'generics', 'glue', 'lifecycle', 'magrittr', 'R6', 'rlang', 'tibble', 'tidyselect', 'vctrs', 'pillar'
```

[illegible]

```
package 'ggrepel' successfully unpacked and MD5 sums checked
package 'ggsci' successfully unpacked and MD5 sums checked
package 'cowplot' successfully unpacked and MD5 sums checked
package 'ggsignif' successfully unpacked and MD5 sums checked
package 'gridExtra' successfully unpacked and MD5 sums checked
package 'polynom' successfully unpacked and MD5 sums checked
package 'ggpubr' successfully unpacked and MD5 sums checked
```

```
gccpubr installed
```

@2022. Jorge Ribeiro| **Unidade Curricular: Aprendizagem Organizacional**– Ano Letivo 2021/2022

Carregar a nossa base de dados

```
>
> # Importante: selecionar o diretório de trabalho (working directory)
> # Isso pode ser feito manualmente: Session > Set Working Directory > Choose Directory
>
> dados <- read.csv2('Insol.csv', stringsAsFactors = T) # Carregamento do arquivo csv
> View(dados) # Visualização dos dados em janela separada
> glimpse(dados) # Visualização de um resumo dos dados
Rows: 2,738
Columns: 33
$ I..Distrito
$ NACE1
$ NACE2
$ Numero_Empregados
$ Num_anos_disponiveis_base
$ Variacao_das_vendas
$ Variacao_resultado_operacional
$ Variacao_do_ativo
$ Variacao_dos_capitais_proprios
$ Variacao_do_ativo_circulante
$ Variacao_das_existencias
$ Variacao_do_imobilizado
$ Fundo_de_maneyio
$ REPV
$ Prazo_medio_de_recebimento
$ Liquidez_geral
$ Liquidez_reduzida
$ Solvabilidade
$ Autonomia_financeira
$ Endividamento
$ Estrutura_financeira
$ Passivo_de_curto_prazo_a_dividir_passivo_total
$ Custos_dos_encargos_financeiros_dividir_resultado_operacional
$ Rendibilidade_operacional_vendas
$ Rendibilidade_liquida_das_vendas
$ Rendibilidade_do_ativo
$ Rendibilidade_capitais_proprios
$ Passivo_curto_prazo_dividir_vendas
$ Peso_das_amortizacoes_dividir_vendas
$ Peso_encargos_financeiros_dividir_vendas
$ Produtividade_por_trabalhador
$ Certificacao_legal_contas
$ Insolvencia
>
>
```

```
<int> 1, 1, 5, 5, 1, 5, 1, 1, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 5, 1~
<int> 14, 13, 14, 14, 13, 14, 13, 13, 14, 14, 14, 13, 13, 13, ~
<int> 141, 139, 141, 141, 139, 141, 139, 139, 141, 141, 141, ~
<int> 225, 28, 80, 175, 140, 229, 65, 34, 86, 70, 31, 59, 29, ~
<int> 9, 10, 9, 8, 10, 10, 10, 10, 10, 10, 9, 7, 10, 10, 5, 1~
<dbl> -0.055115655, -0.163781044, -0.193743196, -0.156786571, ~
<dbl> -12.183442840, -7.685481008, -1.486740076, -1.366234018~
<dbl> -0.081616494, -0.120589583, -0.118888347, -0.122806701, ~
<dbl> -0.353879041, 0.002009229, 0.055864361, -0.109388014, 0~
<dbl> -0.464740547, -0.121441848, -0.123646343, -0.127069249, ~
<dbl> -0.424967741, -0.231066322, 0.113842558, 0.079843417, ~
<dbl> 0.466459003, -0.104213681, -0.074894492, -0.106139680, ~
<int> -753912, 1557403, 1675145, 1529773, 9857170, 58434, 399~
<dbl> 11.4680931, 10.5766529, 8.1529636, 4.9280958, 7.6508458~
<dbl> 53.17243, 132.82895, 31.24659, 46.36006, 127.44042, 116~
<dbl> 0.4873000, 2.8132551, 1.9732527, 3.1698169, 6.6907927, ~
<dbl> 0.3433478, 2.3932337, 0.8713576, 1.9044642, 5.9766310, ~
<dbl> 0.42055419, 1.96253227, 1.19854495, 2.99992624, 7.78028~
<dbl> 0.29604924, 0.66245093, 0.54515357, 0.74999539, 0.88610~
<dbl> 0.70395028, 0.33754907, 0.45484616, 0.25000461, 0.11389~
<dbl> 2.37781486, 0.50954576, 0.83434501, 0.33334153, 0.12852~
<dbl> 1.0000000, 1.0000000, 1.0000000, 1.0000000, 1.0000000, ~
<dbl> -9.6936455, -0.8076247, -17.2563028, -1.7314670, 4.7478~
<dbl> -0.123984621, -0.004935802, -0.015714433, -0.021468601, ~
<dbl> -0.139525438, 0.000885839, 0.007058774, -0.059088467, 0~
<dbl> -0.162145523, 0.001328347, 0.028843605, -0.092117357, 0~
<dbl> -0.353879041, 0.002009229, 0.055864872, -0.109388435, 0~
<dbl> 0.90460643, 0.10335909, 0.10013587, 0.42279228, 0.22354~
<dbl> 0.040788155, 0.025632202, 0.007319208, 0.016605048, 0.0~
<dbl> 0.012790299, 0.006111504, 0.000910649, 0.012399082, 0.0~
<dbl> 0.9053539, 1.3304156, 1.2058435, 0.9294785, 1.7746687, ~
<int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
<int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
```

■ Exemplo Prático R

Carregar a nossa base de dados

	1..Distrito	NACE1	NACE2	Numero_Emplegados	Num_anos_disponiveis_base	Variacao_das_vendas	Variacao_resultado_operacional	Variacao_do_activo	Variacao_dos_capitais_proprios	Variacao_do_activo_circulante	Variacao_das_existencias
1	1	14	141	225	9	-0.055115655	-1.218344e+01	-0.081616494	-3.538790e-01	-0.464740547	-0.424967741
2	1	13	139	28	10	-0.163781044	-7.685481e+00	-0.120589583	2.009229e-03	-0.121441848	-0.231066322
3	5	14	141	80	9	-0.193743196	-1.486740e+00	-0.118888347	5.586436e-02	-0.123646343	0.113842558
4	5	14	141	175	8	-0.156786571	-1.366234e+00	-0.122806701	-1.093880e-01	-0.127069249	0.079843417
5	1	13	139	140	10	-0.059913255	-7.426743e-01	-0.019611869	2.116005e-03	-0.041525475	-0.093254188
6	5	14	141	229	10	0.020864192	-6.146437e-01	0.132667083	4.946310e-01	0.188161698	0.088134919
7	1	13	139	65	10	-0.085960123	-6.006328e-01	-0.001992579	1.484003e-02	0.013301799	0.001616991
8	1	13	139	34	10	-0.078627227	-5.906208e-01	-0.079501363	2.465032e-02	-0.057614608	0.074862410
9	5	14	141	86	10	-0.162168944	-5.205390e-01	-0.004541500	6.885403e-02	-0.017202442	-0.235599228
10	1	14	141	70	9	-0.075617524	-5.085054e-01	-0.088133731	2.346505e-03	-0.100527603	-0.114144022
11	1	14	141	31	7	-0.167016827	-4.356107e-01	0.004582846	3.308423e-01	-0.048573110	-0.402278743
12	1	13	139	59	10	-0.089238713	-3.822100e-01	0.054113060	2.291801e-03	0.072320525	0.051509164
13	1	13	139	29	10	-0.228662239	-3.623400e-01	-0.003783828	2.740638e-02	0.015426724	-0.046642495
14	1	13	133	104	10	-0.047668975	-3.427305e-01	-0.184070076	-3.464455e-01	-0.202449035	-0.261168307
15	1	14	143	43	5	-0.058445918	-3.367530e-01	-0.012531387	1.228694e-02	0.003292265	0.431920427
16	1	13	139	17	10	0.009047675	-2.898037e-01	0.078333327	6.591196e-02	0.098692365	0.062070782
17	1	14	143	503	10	-0.230967121	-2.723515e-01	-0.124435270	-3.895456e-01	0.603331414	0.020347655
18	5	13	139	208	10	-0.096633893	-2.165425e-01	-0.033931236	-7.392067e-03	-0.018783820	-0.047467313
19	1	14	141	205	10	-0.005725962	-1.627703e-01	0.092729117	1.248653e-01	0.107842013	-0.295778968
20	3	14	141	87	4	-0.264036397	-1.627283e-01	-0.274520330	5.058638e-01	-0.344781990	-0.111486534
21	1	14	141	48	10	0.040989241	-1.209812e-01	-0.092125445	3.255864e-02	-0.092883398	0.267465831
22	1	13	131	32	10	0.012374325	-9.437071e-02	-0.108802719	3.120543e-03	-0.120993219	-0.211744977
23	1	13	132	262	6	-0.138576857	-5.692331e-02	-0.024735696	2.455965e-02	-0.017509278	0.201115067
24	5	13	139	211	10	-0.031928013	-3.823434e-02	-0.048246091	6.409146e-03	-0.040525966	-0.163566911
25	1	14	141	88	10	0.045551070	-3.812579e-02	-0.050767738	2.325409e-02	-0.197908017	-0.407532189
26	5	14	141	183	8	0.160994780	9.824328e-03	0.019865119	1.294728e-02	-0.050185694	-0.260672700
27	5	14	141	19	8	-0.030501888	6.725872e-02	0.196649992	2.039723e-01	0.229037899	0.269477448
28	5	13	139	117	10	0.016926203	1.035330e-01	-0.174829441	3.715177e-02	-0.185081771	-0.106882393
29	1	14	141	112	10	0.065675287	2.017970e-01	0.029394129	4.317458e-02	0.051797948	0.101352231
30	1	14	141	61	6	0.306670695	2.025954e-01	0.086232709	3.398194e-01	0.089979851	0.328536651
31	1	14	141	47	8	-0.037203564	3.106125e-01	-0.104398470	2.476318e-01	-0.121009339	-0.221631621
32	5	14	141	49	9	0.113843167	6.412580e-01	-0.132047150	-3.337995e-02	-0.169939242	-0.136587477
33	5	14	141	74	10	0.363529911	1.377419e+00	0.172867822	1.826096e-01	0.864912866	-0.191359258
34	1	14	141	126	10	0.257090777	2.508205e+00	0.135658612	3.818913e-01	0.163900779	-0.038922952
35	1	13	139	68	10	-0.026554177	2.907121e+00	0.072909640	8.767878e-03	0.061797109	0.025295391
36	1	14	141	66	10	0.179289088	2.917486e+00	-0.055734876	-1.054338e-01	-0.091664397	-0.216104110
37	1	14	141	100	7	1.695110918	3.576696e+00	0.783309475	-7.084709e-01	1.038122515	845.991260900
38	1	13	131	89	7	-0.068342519	3.595353e+00	-0.030320330	-2.128223e-03	0.126116039	-0.110322593
39	1	14	141	66	10	0.539870809	5.464827e+00	-0.273097852	-1.335281e-01	-0.396147975	-0.560433736
40	1	13	139	156	9	0.126328338	7.811167e+00	0.011285193	1.417887e-01	-0.127079122	0.077260791
41	5	13	131	66	6	0.295027102	1.479796e+01	0.057770488	6.640860e-01	0.514312588	0.184372086
42	5	14	141	309	10	-0.261065864	-1.220944e+00	-0.134499069	-1.169732e-01	-0.149417320	-0.199238670
43	1	13	132	103	10	-0.183090893	-9.342349e+00	-0.223110855	-3.028151e-01	-0.253747872	-0.018153311
44	1	13	132	22	6	0.182341815	-1.198144e+00	0.110595101	4.824962e-02	0.189956948	0.454624431
45	1	13	139	207	10	0.050375041	-2.180141e-01	-0.020474542	7.375795e-02	0.030501737	0.209743511

Criação do plot e visualização gráfica

```
# Passo 3: Verificação dos pressupostos para a regressão linear

## Relação linear entre a Numero_Empregados e Fundo_de_maneio:
plot(dados$Fundo_de_maneio, dados$Numero_Empregados)

## Construção do modelo:
mod <- lm(Numero_Empregados ~ Fundo_de_maneio, dados)

## Análise gráfica:
par(mfrow=c(2,2))

plot(mod)

par(mfrow=c(1,1))
```

Entender os plots de diagnóstico

1. Residuals vs Fitted

Permite ver se existem padrões não lineares, por exemplo se tivermos uma parábola então a relação não linear não é explicada pelo modelo e foi deixada de fora nos residuals

2. Normal Q-Q

Serve para verificar se os resíduos estão distribuídos normalmente, seguem a linha reta? Ou estão severamente afastados

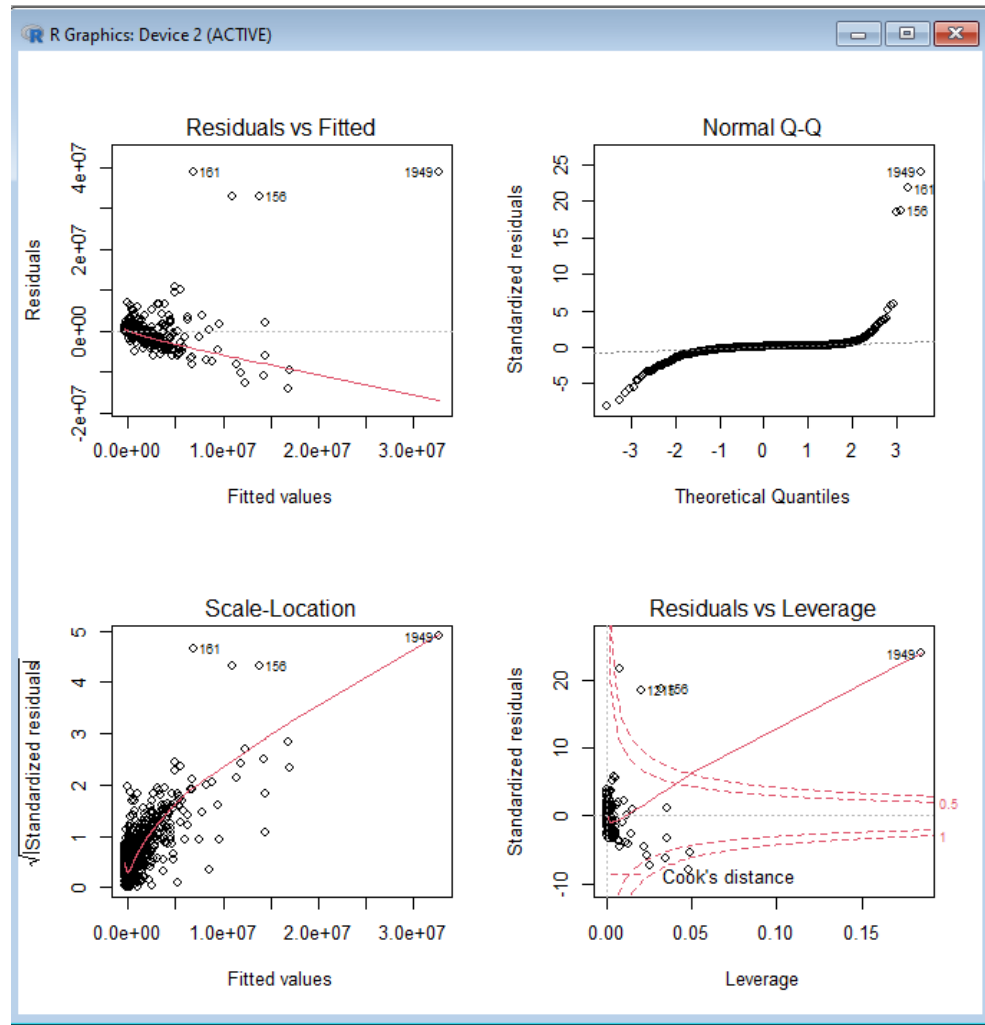
3. Scale Location

Serve para verificação da hipótese da homocedasticidade dos resíduos.

4. Residuals vs Leverage

Pode ser útil para detectar a presença de pontos influenciadores.

Plots em R



Analizando os gráficos anteriores podemos concluir que temos pouca data para os “fitted values” superiores comparada aos outros através do scale location, uma vez que no mesmo temos uma reta com inclines muito grandes em vez de horizontal. O que indica que não temos equal variance.

Podemos concluir também através do quarto gráfico que existem casos influenciados uma vez que existem casos afastados do conglomerado de pontos.

De seguida vamos realizar alguns testes estatísticos que devem correlar com os gráficos obtidos.



Testes estatísticos

```
R Console
> ## Normalidade dos resíduos:
> shapiro.test(mod$residuals)

      Shapiro-Wilk normality test

data:  mod$residuals
W = 0.3373, p-value < 2.2e-16

>
```

Valor de P muito baixo podemos rejeitar a hipótese nula e que a distribuição não é normal

```
W = 0.3373, p-value < 2.2e-16

>
>
> ## Outliers nos resíduos:
> summary(rstandard(mod))
      Min.    1st Qu.    Median     Mean    3rd Qu.     Max.
-8.015012 -0.098519  0.091694  0.000737  0.183277 24.000123

>
```

Valor mínimo de -8 e máximo de 24 como estão longe um do outro temos outliers

Testes estatísticos

```
> ## Independência dos resíduos (Durbin-Watson):  
> durbinWatsonTest(mod)  
lag Autocorrelation D-W Statistic p-value  
1 0.05277543 1.885957 0.016  
Alternative hypothesis: rho != 0  
>  
>
```

Apenas importante quando existem medidas repetidas segundo e quando existe normalidade
“Fávero, L. P., & Belfiore, P. (2017). Manual de análise de dados: estatística e modelagem multivariada com Excel®, SPSS® e Stata®.” Elsevier Brasil.”

```
>  
> ## Homocedasticidade (Breusch-Pagan):  
> bptest(mod)  
  
studentized Breusch-Pagan test  
  
data: mod  
BP = 208.18, df = 1, p-value < 2.2e-16  
>
```

Como o nosso p está abaixo de 0.05 podemos dizer que não há homocedasticidade

■ Exemplo Prático R

```
>
> # Passo 4: Análise do modelo
> summary(mod)

Call:
lm(formula = Fundo_de_maneyio ~ Numero_Empregados, data = dados)

Residuals:
    Min       1Q   Median       3Q      Max
-14071352  -177250   164968   329733  38993154

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -433021.3   38662.1  -11.20  <2e-16 ***
Numero_Empregados  25444.7     607.2   41.91  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

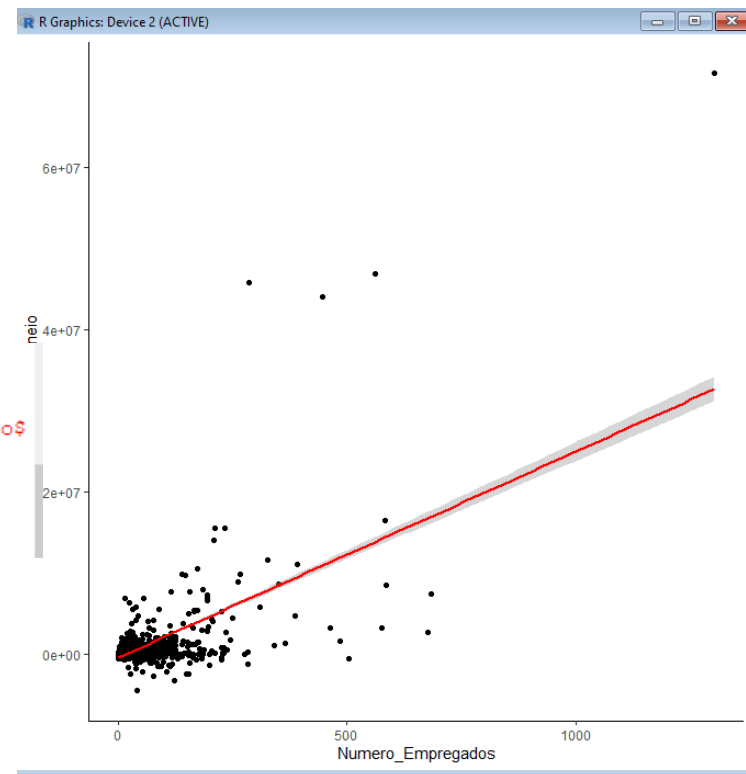
Residual standard error: 1799000 on 2736 degrees of freedom
Multiple R-squared:  0.391,    Adjusted R-squared:  0.3907
F-statistic: 1756 on 1 and 2736 DF,  p-value: < 2.2e-16
```

Como temos hipótese nula muito baixo podemos rejeitar a hipótese nula que indica que para aumentar uma unidade de fundos de maneio são necessários 25 444 empregados em média.

Gráfico da regressão linear

Tal como anteriormente em python
geramos agora o gráfico da regressão
linear

```
>  
>  
>  
> ggplot(data = dados, mapping = aes(x = Numero_Empregados, y = Fundo_de_manueio$  
+   geom_point() +  
+   geom_smooth(method = "lm", col = "red") +  
+   theme_classic()  
'geom_smooth()' using formula 'y ~ x'  
>
```



Bibliografia

- <http://math.furman.edu/~dcs/courses/math47/R/library/lmtest/html/dwtest.html>
- <https://data.library.virginia.edu/diagnostic-plots/>
- <https://community.rstudio.com/t/error-after-r-update-lib-c-program-files-r-r-3-5-0-library-is-not-writable/7947/2>
- <https://medium.com/data-hackers/implementando-regress%C3%A3o-linear-simples-em-python-91df53b920a8>

o teu • de partida



Instituto Politécnico
de Viana do Castelo

www.ipvc.pt