

**Nomes:** João Rodrigues, Marcos de Campos, Vinícius Ferreira

**Observação:** a lista foi realizada usando um notebook em Python. Ao final do documento há o código com as definições das funções utilizadas para resolver os problemas da lista. O notebook com todos os códigos e cálculos foi enviado juntamente com esse PDF.

## Exercício C2

### Questão (i)

Ao regredirmos *lwage* sobre a lista de variáveis exógenas *educ*, *exper*, *tenure*, *married*, *South*, *urban*, *black* e *IQ*, obtém-se a seguinte regressão:

```
## Fazendo a Regressão Original
x = df[['educ', 'exper', 'tenure', 'married', 'south', 'urban', 'black', 'IQ']]
y = df['lwage']
Regressão_Múltipla(x,y)
```

O erro padrão da regressão é 0.36315 e a SQR é 122.12027

OLS Regression Results						
Dep. Variable:	lwage	R-squared:	0.263			
Model:	OLS	Adj. R-squared:	0.256			
Method:	Least Squares	F-statistic:	41.27			
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	1.52e-56			
Time:	08:07:43	Log-Likelihood:	-375.09			
No. Observations:	935	AIC:	768.2			
Df Residuals:	926	BIC:	811.7			
Df Model:	8					
Covariance Type:	nonrobust					
coef	std err	t	P> t	[0.025	0.975]	
const	5.1764	0.128	40.441	0.000	4.925	5.428
educ	0.0544	0.007	7.853	0.000	0.041	0.068
exper	0.0141	0.003	4.469	0.000	0.008	0.020
tenure	0.0114	0.002	4.671	0.000	0.007	0.016
married	0.1998	0.039	5.148	0.000	0.124	0.276
south	-0.0802	0.026	-3.054	0.002	-0.132	-0.029
urban	0.1819	0.027	6.791	0.000	0.129	0.235
black	-0.1431	0.039	-3.624	0.000	-0.221	-0.066
IQ	0.0036	0.001	3.589	0.000	0.002	0.006

Substituindo *IQ* por *KWW* como proxy de aptidão individual, o coeficiente de educação passa de 0,0544 para 0,0576, ou seja, há um aumento no retorno da educação sobre o salário de aproximadamente 0,32%. Além disso, o coeficiente *t* de educação com *KWW* como proxy é maior do que o de *IQ*, o que sugere uma estimativa mais precisa.

```

## Substituindo IQ por KWW nas variáveis exógenas
x = df[['educ','exper','tenure','married','south','urban','black','KWW']]
y = df['lwage']
Regressão_Múltipla(x,y)

```

#### OLS Regression Results

Dep. Variable:	lwage	R-squared:	0.259			
Model:	OLS	Adj. R-squared:	0.252			
Method:	Least Squares	F-statistic:	40.39			
Date:	Wed, 17 Feb 2021	Prob (F-statistic):	1.93e-55			
Time:	22:39:03	Log-Likelihood:	-377.71			
No. Observations:	935	AIC:	773.4			
Df Residuals:	926	BIC:	817.0			
Df Model:	8					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	5.3588	0.114	47.172	0.000	5.136	5.582
educ	0.0576	0.007	8.428	0.000	0.044	0.071
exper	0.0122	0.003	3.773	0.000	0.006	0.019
tenure	0.0111	0.002	4.507	0.000	0.006	0.016
married	0.1895	0.039	4.848	0.000	0.113	0.266
south	-0.0916	0.026	-3.502	0.000	-0.143	-0.040
urban	0.1755	0.027	6.494	0.000	0.122	0.229
black	-0.1643	0.039	-4.263	0.000	-0.240	-0.089
KWW	0.0050	0.002	2.764	0.006	0.001	0.009

#### Questão (ii)

Estimando a regressão com *IQ* e *KWW* como proxies para aptidão individual, obtém-se os resultados abaixo. O retorno marginal da educação sobre o salário diminui para menos de aproximadamente 5%.

```

##Usando IQ E KWW como proxys para aptidão
x = df[['educ','exper','tenure','married','south','urban','black','IQ','KWW']]
Regressão_Múltipla(x,y)

```

#### OLS Regression Results

Dep. Variable:	lwage	R-squared:	0.266			
Model:	OLS	Adj. R-squared:	0.259			
Method:	Least Squares	F-statistic:	37.28			
Date:	Wed, 17 Feb 2021	Prob (F-statistic):	1.25e-56			
Time:	22:40:34	Log-Likelihood:	-372.94			
No. Observations:	935	AIC:	765.9			
Df Residuals:	925	BIC:	814.3			
Df Model:	9					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	5.1756	0.128	40.506	0.000	4.925	5.426
educ	0.0498	0.007	6.863	0.000	0.036	0.064
exper	0.0128	0.003	3.947	0.000	0.006	0.019
tenure	0.0109	0.002	4.467	0.000	0.006	0.016
married	0.1921	0.039	4.938	0.000	0.116	0.269
south	-0.0820	0.026	-3.128	0.002	-0.133	-0.031
urban	0.1758	0.027	6.534	0.000	0.123	0.229
black	-0.1304	0.040	-3.268	0.001	-0.209	-0.052
IQ	0.0031	0.001	3.079	0.002	0.001	0.005
KWW	0.0038	0.002	2.066	0.039	0.000	0.007

### Questão (iii)

Fazendo o teste de significância conjunta de  $IQ$  e  $KWW$ , vê-se que as variáveis são conjuntamente estatisticamente significantes até mesmo ao nível de 1% de significância, com um p-valor baixíssimo de 0,0002. Assim, ambas as variáveis são boas de serem usadas como proxies para aptidão intelectual e devem ser incluídas no modelo.

```
## Testando a significância conjunta de IQ e KWW
x = df[['educ', 'exper', 'tenure', 'married', 'south', 'urban', 'black', 'IQ', 'KWW']]
y = df['lwage']
Teste_F(x,y, df[['IQ', 'KWW']])
```

O valor de F é 8.595 e seu p-valor é 0.0002002. Portanto, rejeita-se  $H_0$  à significância de 5.0%.

## Exercício C3

### Questão (i)

O modelo de regressão simples de  $lscrap$  sobre  $grant$  pode violar a hipótese RLM.4, uma vez que características da empresa que podem influenciar na obtenção ou não de um subsídio – como seu tamanho, aproximado pela coluna de  $sales$  – foram relegados ao termo de erro e, portanto, é provável que  $E(u|x) \neq 0$ .

### Questão (ii)

Os resultados da estimação usando apenas os dados de 1988 podem ser vistos abaixo. Como não se pode supor causalidade, não se pode dizer que uma concessão de subsídio aumente a taxa de refugo da empresa. Contudo, o coeficiente de  $grant$  indica uma correlação: a concessão de subsídio aumenta em aproximadamente 5,66% a taxa de refugo da empresa – o que vai contra a intuição -, apesar de seu coeficiente não ser estatisticamente significante.

```
## Fazendo a regressão simples de lscrap sobre grant no ano de 1988
# Baixando a base de dados e pegando apenas o ano de 1988
coletar_dados('JTRAIN')
df_88 = df.loc[df['d88'] == 1]

# Dropando os valores nulos
df_88.dropna(subset=['lscrap', 'grant'], axis = 0, inplace = True)

# Fazendo a regressão
x = df_88[['grant']]
y = df_88['lscrap']
Regressão_Múltipla(x,y)
```

#### OLS Regression Results

Dep. Variable:	lscrap	R-squared:	0.000			
Model:	OLS	Adj. R-squared:	-0.019			
Method:	Least Squares	F-statistic:	0.01948			
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	0.890			
Time:	08:26:28	Log-Likelihood:	-94.660			
No. Observations:	54	AIC:	193.3			
Df Residuals:	52	BIC:	197.3			
Df Model:	1					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	0.4085	0.241	1.698	0.095	-0.074	0.891
grant	0.0566	0.406	0.140	0.890	-0.757	0.870

### Questão (iii)

Usando o *lscrap\_87* como variável exógena ao lado de *grant*, o coeficiente de *grant* se torna negativo (o que era esperado, uma vez que se espera que o subsídio governamental diminua a *turnover* da empresa), além de se tornar estatisticamente significante ao nível de 10% contra uma alternativa bilateral. Isso indica que há características estruturais da empresa que permanecem ao longo do tempo e afeta, de forma significativa a sua taxa de refugo, como pode ser visto pela estatística *t* de *lscrap\_87*.

Dividindo o p-valor da hipótese bilateral por 2, o p-valor da alternativa unilateral é 0,045, ou seja, rejeita-se a hipótese nula de que o coeficiente de *grant* é maior que 0 em favor de  $H_1: \beta_{\text{grant}} < 0$  ao nível de 5%.

```
## Usando a variável lscrap de 87 como regressora
df_88.rename(columns={'lscrap_1':'lscrap_87'}, inplace=True)

# Fazendo a regressão
x = df_88[['grant', 'lscrap_87']]
y = df_88['lscrap']
Regressão_Múltipla(x,y)
```

OLS Regression Results						
Dep. Variable:	lscrap	R-squared:	0.873			
Model:	OLS	Adj. R-squared:	0.868			
Method:	Least Squares	F-statistic:	174.9			
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	1.47e-23			
Time:	08:44:16	Log-Likelihood:	-39.000			
No. Observations:	54	AIC:	84.00			
Df Residuals:	51	BIC:	89.97			
Df Model:	2					
Covariance Type:	nonrobust					
coef	std err	t	P> t	[0.025	0.975]	
const	0.0212	0.089	0.238	0.813	-0.158	0.200
grant	-0.2540	0.147	-1.727	0.090	-0.549	0.041
lscrap_87	0.8312	0.044	18.701	0.000	0.742	0.920
Omnibus:	13.769	Durbin-Watson:	1.541			
Prob(Omnibus):	0.001	Jarque-Bera (JB):	35.075			
Skew:	-0.526	Prob(JB):	2.42e-08			
Kurtosis:	6.805	Cond. No.	3.95			

### Questão (iv)

O coeficiente de *lscrap\_87* é 0,8312 e seu erro padrão é 0,044. Ao nível de significância de 5%, seu intervalo de confiança não atinge 1, ou seja, rejeita-se  $H_0: \beta_{\text{lscrap}_87} = 1$  em favor de  $H_1: \beta_{\text{lscrap}_87} \neq 1$ .

## Questão (v)

Fazendo a regressão usando erros robustos em relação a heteroscedasticidade:

```
# Fazendo a regressão com erros robustos com relação a heteroscedasticidade
x = df_88[['grant', 'lscrap_87']]
y = df_88['lscrap']
Regressão_Múltipla(x,y, robusta='S')
```

OLS Regression Results						
Dep. Variable:	lscrap	R-squared:	0.873			
Model:	OLS	Adj. R-squared:	0.868			
Method:	Least Squares	F-statistic:	77.79			
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	3.22e-16			
Time:	08:49:30	Log-Likelihood:	-39.000			
No. Observations:	54	AIC:	84.00			
Df Residuals:	51	BIC:	89.97			
Df Model:	2					
Covariance Type:	HC1					
	coef	std err	t	P> t	[0.025	0.975]
const	0.0212	0.100	0.213	0.832	-0.179	0.222
grant	-0.2540	0.146	-1.735	0.089	-0.548	0.040
lscrap_87	0.8312	0.074	11.302	0.000	0.684	0.979
Omnibus:	13.769	Durbin-Watson:	1.541			
Prob(Omnibus):	0.001	Jarque-Bera (JB):	35.075			
Skew:	-0.526	Prob(JB):	2.42e-08			
Kurtosis:	6.805	Cond. No.	3.95			

O p-valor dividido por 2 do coeficiente de *grant* é 0,0485, ou seja, o coeficiente é ainda mais significantemente diferente de 0 com os erros padrões robustos, indicando uma estimativa mais precisa do coeficiente.

O intervalo de confiança de *lscrap\_87* se aproxima mais de 1, mas o coeficiente ainda é estatisticamente diferente de um ao nível de significância de 5%.

## Exercício C4

### Questão (i)

O coeficiente de *DC* é muito grande e estatisticamente muito significante, o que indica que o estado de *DC* possuirá uma mortalidade infantil muito maior que os outros estados (todos os outros fatores mantidos constantes), como mostra a estimativa abaixo:

```
## Pegando apenas os dados de 1990
df_90 = df.loc[df['year'] == 1990]

# Fazendo a regressão com a dummy do estado de Columbia
x = df_90[['lpcinc', 'lphysic', 'lpopul', 'DC']]
y = df_90['infmort']
Regressão_Múltipla(x,y)
```

OLS Regression Results						
Dep. Variable:	infmort	R-squared:	0.691			
Model:	OLS	Adj. R-squared:	0.664			
Method:	Least Squares	F-statistic:	25.71			
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	3.15e-11			
Time:	09:01:04	Log-Likelihood:	-80.968			
No. Observations:	51	AIC:	171.9			
Df Residuals:	46	BIC:	181.6			
Df Model:	4					
Covariance Type:	nonrobust					
coef	std err	t	P> t	[0.025	0.975]	
const	23.9548	12.419	1.929	0.060	-1.044	48.954
lpcinc	-0.5669	1.641	-0.345	0.731	-3.871	2.737
lphysic	-2.7418	1.191	-2.303	0.026	-5.139	-0.345
lpopul	0.6292	0.191	3.293	0.002	0.245	1.014
DC	16.0350	1.769	9.064	0.000	12.474	19.596

### Questão (ii)

As estimativas para o ano de 1990 sem o estado de DC na regressão estão abaixo:

```
# Pegando apenas os dados fora de DC
df_nDC = df_90.loc[df_90['DC'] == 0]

# Fazendo a regressão com a dummy do estado de Columbia
x = df_nDC[['lpcinc', 'lphysic', 'lpopul']]
y = df_nDC['infmort']
Regressão_Múltipla(x,y)
```

OLS Regression Results						
Dep. Variable:	infmort	R-squared:	0.273			
Model:	OLS	Adj. R-squared:	0.226			
Method:	Least Squares	F-statistic:	5.763			
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	0.00197			
Time:	09:07:04	Log-Likelihood:	-79.876			
No. Observations:	50	AIC:	167.8			
Df Residuals:	46	BIC:	175.4			
Df Model:	3					
Covariance Type:	nonrobust					
coef	std err	t	P> t	[0.025	0.975]	
const	23.9548	12.419	1.929	0.060	-1.044	48.954
lpcinc	-0.5669	1.641	-0.345	0.731	-3.871	2.737
lphysic	-2.7418	1.191	-2.303	0.026	-5.139	-0.345
lpopul	0.6292	0.191	3.293	0.002	0.245	1.014

Comparando com a regressão da questão (i), todas as estimativas e erros padrões são idênticos, ou seja, a inclusão de uma dummy para apenas uma observação equivale a retirar essa observação da regressão.

## Exercício C8

### Questão (i)

A média de *stotal* é 0,047 e seu erro padrão é 0,853, como mostra a imagem abaixo. Um fato interessante é que a mediana da variável é 0 e, portanto, menor que sua média:

```
## Coletando as estatísticas descritivas de 'stotal'  
df['stotal'].describe()
```

```
count    6763.00000  
mean     0.047483  
std      0.853544  
min     -3.324797  
25%    -0.327343  
50%     0.000000  
75%     0.610791  
max      2.235366
```

### Questão (ii)

A regressão de *jc* sobre *stotal* faz com que o coeficiente da variável exógena seja 0,0112 e seu erro padrão seja 0,011, resultando em um valor tanto estatisticamente quanto economicamente não significante (próximo de 0).

A regressão de *univ* sobre *stotal* faz com que o coeficiente da variável exógena seja 1,16 e seu erro padrão seja 0,029, resultando em um valor estatisticamente e economicamente bastante significante.

Assim, *univ* é bastante correlacionada com a proxy de aptidão individual, o que, a partir da Solução Plugada do Problema de Variáveis Omitidas, pode tornar os estimadores viesados, mas o viés pode ser menor do que aquele que ocorreria caso se omitisse a variável proxy. Isso ocorre em razão de pessoas com maior aptidão – aproximada por *stotal* – terem maior chance de irem a uma faculdade completa de 4 anos.

```
## Fazendo as regressões simples  
y1 = df['jc']  
x = df[['stotal']]  
Regressao_Multipla(x,y1)
```

```
OLS Regression Results  
=====
```

Dep. Variable:	jc	R-squared:	0.000			
Model:	OLS	Adj. R-squared:	0.000			
Method:	Least Squares	F-statistic:	1.032			
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	0.310			
Time:	09:17:21	Log-Likelihood:	-7846.3			
No. Observations:	6763	AIC:	1.570e+04			
Df Residuals:	6761	BIC:	1.571e+04			
Df Model:	1					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	0.3384	0.009	35.983	0.000	0.320	0.357
stotal	0.0112	0.011	1.016	0.310	-0.010	0.033

```
## Fazendo as regressões simples
y2 = df['univ']
x = df[['stotal']]
Regressao_Multipla(x,y2)
```

#### OLS Regression Results

Dep. Variable:	univ	R-squared:	0.189
Model:	OLS	Adj. R-squared:	0.189
Method:	Least Squares	F-statistic:	1575.
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	8.95e-310
Time:	09:18:16	Log-Likelihood:	-14512.
No. Observations:	6763	AIC:	2.903e+04
Df Residuals:	6761	BIC:	2.904e+04
Df Model:	1		
Covariance Type:	nonrobust		
coef	std err	t	P> t
const	1.8707	0.025	74.248
stotal	1.1697	0.029	39.683

### Questão (iii)

Adicionando *stotal* à equação 4.17 do livro, obtém-se as estimativas abaixo:

```
## Fazendo a regressão de 4.17 adicionando 'stotal'
x = df[['jc','univ','exper','stotal']]
y = df['lwage']
Regressao_Multipla(x,y)
```

#### OLS Regression Results

Dep. Variable:	lwage	R-squared:	0.228
Model:	OLS	Adj. R-squared:	0.228
Method:	Least Squares	F-statistic:	500.2
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	0.00
Time:	09:26:50	Log-Likelihood:	-3862.5
No. Observations:	6763	AIC:	7735.
Df Residuals:	6758	BIC:	7769.
Df Model:	4		
Covariance Type:	nonrobust		
coef	std err	t	P> t
const	1.4953	0.021	70.473
jc	0.0631	0.007	9.243
univ	0.0686	0.003	26.759
exper	0.0049	0.000	31.036
stotal	0.0494	0.007	7.251

O teste *t* bilateral de  $H_0: \beta_{jc} = \beta_{univ}$  contra  $H_1: \beta_{jc} < \beta_{univ}$  resulta em um p-valor bilateral de 0,42, de modo que o p-valor unicaudal do teste é  $0,42/2 = 0,21$ . Assim, não se rejeita a 5% de significância a hipótese de que os retornos são iguais, como é possível ver no código abaixo, que fornece o p-valor bilateral:

```
## Fazendo o teste t para ver se os coeficientes de jc e univ são iguais:
Teste_t_Dois_Coeficientes_Iguais(x,y, df[['jc','univ']])
```

A estatística do teste é  $[-0.80552407]$ , o que resulta em um p-valor de 0.420545663943354

#### Questão (iv)

Adicionando *stotal2* à equação da questão (iii), obtém-se as estimativas abaixo:

```
## Criando a coluna de stotal2
df['stotal2'] = df['stotal']**2

## Fazendo a regressão
x = df[['jc', 'univ', 'exper', 'stotal', 'stotal2']]
y = df['lwage']
Regressao_Multipla(x,y)
```

#### OLS Regression Results

Dep. Variable:	lwage	R-squared:	0.228
Model:	OLS	Adj. R-squared:	0.228
Method:	Least Squares	F-statistic:	400.2
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	0.00
Time:	09:39:04	Log-Likelihood:	-3862.4
No. Observations:	6763	AIC:	7737.
Df Residuals:	6757	BIC:	7778.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	1.4940	0.021	69.635	0.000	1.452	1.536
jc	0.0632	0.007	9.251	0.000	0.050	0.077
univ	0.0685	0.003	26.508	0.000	0.063	0.074
exper	0.0049	0.000	31.036	0.000	0.005	0.005
stotal	0.0502	0.007	7.086	0.000	0.036	0.064
stotal2	0.0019	0.005	0.404	0.686	-0.007	0.011

A transformação quadrática de *stotal* produz uma estimativa nada estatisticamente significante, de modo que ela pode ser omitida e prefere-se o modelo mais simples da questão (iii), visando diminuir a variância dos estimadores dos coeficientes significantes.

#### Questão (v)

Adicionando os termos de interação *stotal\*jc* e *stotal\*univ* à equação e fazendo um teste F, vê-se que os novos termos são estatisticamente não conjuntamente significantes ao nível de 5%, uma vez que o p-valor do teste bilateral é 0,141.

```
## Criando a coluna de stotal_jc e stotal_univ
df['stotal_jc'] = df['stotal']*df['jc']
df['stotal_univ'] = df['stotal']*df['univ']

## Fazendo o teste F
x = df[['jc', 'univ', 'exper', 'stotal', 'stotal_jc', 'stotal_univ']]
y = df['lwage']
Teste_F(x,y,df[['stotal_jc', 'stotal_univ']])
```

O valor de F é 1.959 e seu p-valor é 0.1410201. Portanto, não se rejeita  $H_0$  à significância de 5%.

#### Questão (vi)

O modelo “mais correto” para esse caso é o da questão (iii), uma vez que é o mais simples e não perde em qualidade com relação aos modelos mais complexos usados na questão (iv) e (v) (cujas adições são estatisticamente não significantes ao nível de 5%). A adição de variáveis não significantes não torna os estimadores viesados, mas aumenta a variância das estimativas, especialmente caso haja um alto grau de multicolinearidade entre as variáveis exógenas, como é o caso entre *stotal2* e *univ*, por exemplo.

## Exercício C10

### Questão (i)

Em JTRAIN2 há cerca de 445 homens na amostra, dentre os quais 41,57% receberam treinamento. Em JTRAIN3, há 2675, dentre os quais apenas 6,92% receberam treinamento (em termos totais, a quantidade de pessoas que receberam treinamento foi igual em ambas as amostras). Essa diferença percentual pode ser explicada pelo fato de que as pessoas em JTRAIN2 foram parte de um experimento direcionado, enquanto a amostra de JTRAIN3 diz respeito a uma subpopulação da população ocupada em 1978.

```
▶ M4
## Calculando a proporção dos homens que recebeu treinamento profissional em train2
observacoes = len(train2['train'])
treinamento = len(train2['train'].loc[train2['train'] == 1])

print(f"Há {observacoes} homens na amostra, dentre os quais {treinamento} receberam
      algum treinamento ({round(treinamento/observacoes,4)*100}% do total.)")
Há 445 homens na amostra, dentre os quais 185 receberam algum treinamento (41.57% do total).
```

```
▶ M4
## Calculando a proporção dos homens que recebeu treinamento profissional em train3
observacoes = len(train3['train'])
treinamento = len(train3['train'].loc[train3['train'] == 1])

print(f"Há {observacoes} homens na amostra, dentre os quais {treinamento} receberam
      algum treinamento ({round(treinamento/observacoes,4)*100}% do total.)")
Há 2675 homens na amostra, dentre os quais 185 receberam algum treinamento (6.92% do total).
```

### Questão (ii)

Segundo a estimativa do modelo simples de regressão de *re78* (ganhos reais em 1978) sobre *train*, há um efeito positivo (e marginalmente estatisticamente significante ao nível de 5%) do treinamento de cerca de \$ 1.794,3 na renda anual, como mostram os resultados abaixo:

```
## Fazendo a regressão de re78 sobre train em TRAIN2
x = train2[['train']]
y = train2['re78']
Regressao_Multipla(x,y)

OLS Regression Results
=====
Dep. Variable:          re78    R-squared:       0.018
Model:                 OLS     Adj. R-squared:   0.016
Method:                Least Squares  F-statistic:     8.039
Date: Thu, 18 Feb 2021  Prob (F-statistic): 0.00479
Time: 13:41:48          Log-Likelihood: -1468.8
No. Observations:      445    AIC:             2942.
Df Residuals:          443    BIC:             2950.
Df Model:                  1
Covariance Type:    nonrobust
=====

            coef    std err          t      P>|t|      [0.025      0.975]
const      4.5548    0.408     11.162      0.000      3.753      5.357
train      1.7943    0.633      2.835      0.005      0.551      3.038
=====
```

### Questão (iii)

Controlando para outras variáveis exógenas, o coeficiente de *train* não muda muito, sendo ligeiramente menor que o estimado pela regressão simples. Contudo, o coeficiente estimado não é mais estatisticamente significante ao nível de 5%, mas é ao nível de 10% de significância.

Essa falta de mudança pode ser explicada em razão dos dados serem experimentais: caso o grupo do treinamento e o grupo controle sejam divididos aleatoriamente, o recebimento ou não de treinamento não é correlacionado com as características individuais das observações da amostra.

### Questão (iv)

Usando a base de dados JTRAIN3, as estimativas da regressão simples de *re78* sobre *train* dão resultados completamente opostos (mas muito significantes) àquelas obtidas a partir dos dados de JTRAIN2. Agora, o treinamento teria sido responsável por diminuir a renda anual de 1978 em cerca de \$ 15.204,8:

```
## Fazendo a regressão de re78 sobre train em TRAIN3
x = train3[['train']]
y = train3['re78']
Regressao_Multipla(x,y)
```

OLS Regression Results						
Dep. Variable:	re78	R-squared:	0.061			
Model:	OLS	Adj. R-squared:	0.061			
Method:	Least Squares	F-statistic:	173.4			
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	2.03e-38			
Time:	13:53:50	Log-Likelihood:	-11066.			
No. Observations:	2675	AIC:	2.214e+04			
Df Residuals:	2673	BIC:	2.215e+04			
Df Model:	1					
Covariance Type:	nonrobust					
coef	std err	t	P> t	[0.025	0.975]	
const	21.5539	0.304	70.985	0.000	20.959	22.149
train	-15.2048	1.155	-13.169	0.000	-17.469	-12.941

Controlando para outros fatores, o sinal do estimador de *train* se torna positivo, mas ele é estatisticamente não significante, com um p-valor bastante alto de 0,8:

```
## Fazendo uma regressão controlando para outras variáveis
x = train3[['train','re74','re75','educ','age','black','hisp']]
y = train3['re78']
Regressao_Multipla(x,y)
```

OLS Regression Results						
Dep. Variable:	re78	R-squared:	0.586			
Model:	OLS	Adj. R-squared:	0.584			
Method:	Least Squares	F-statistic:	538.4			
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	0.00			
Time:	13:53:51	Log-Likelihood:	-9971.5			
No. Observations:	2675	AIC:	1.996e+04			
Df Residuals:	2667	BIC:	2.001e+04			
Df Model:	7					
Covariance Type:	nonrobust					
coef	std err	t	P> t	[0.025	0.975]	
const	1.6476	1.301	1.266	0.205	-0.903	4.198
train	0.2132	0.853	0.250	0.803	-1.460	1.887
re74	0.2810	0.028	10.071	0.000	0.226	0.336
re75	0.5693	0.028	20.648	0.000	0.515	0.623
educ	0.5201	0.075	6.914	0.000	0.373	0.668
age	-0.0751	0.020	-3.667	0.000	-0.115	-0.035
black	-0.6477	0.492	-1.317	0.188	-1.612	0.317
hisp	2.2026	1.093	2.016	0.044	0.060	4.345

A mudança em relação ao que se observou usando os dados de JTRAIN2 é justamente o caráter não-experimental das observações de JTRAIN3, o que pode ser visto pelas estimativas da segunda regressão.

*Train* estaria, portanto, correlacionada com outras variáveis em JTRAIN3, como mostram as estimativas de uma regressão auxiliar de *train* sobre as demais variáveis exógenas do modelo 2 da questão (iv): segundo elas, pessoas que receberam treinamento já possuíam menores rendas nos anos anteriores (o que pode ter motivado a escolha pelo treinamento), são de minorias étnicas e possuíam um menor nível de educação e idade.

```
## Fazendo uma regressão de train sobre as demais variáveis exógenas do modelo 2
x = train3[['re74', 're75', 'educ', 'age', 'black', 'hisp']]
y = train3['train']
Regressao_Multipla(x,y)
```

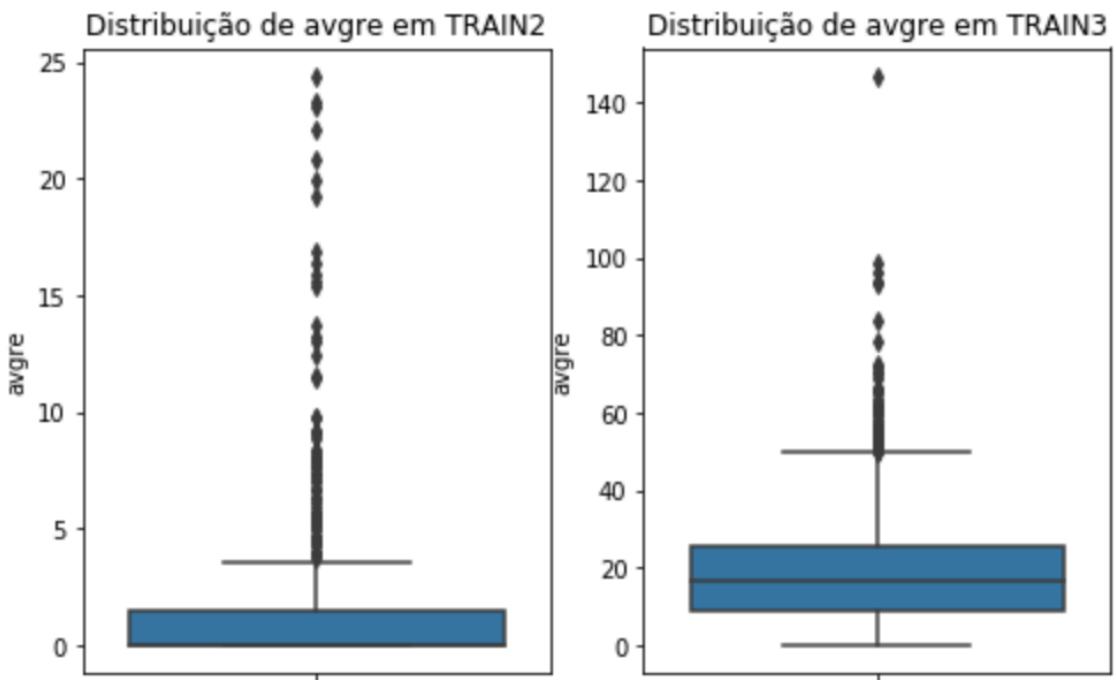
OLS Regression Results						
Dep. Variable:	train	R-squared:	0.190			
Model:	OLS	Adj. R-squared:	0.189			
Method:	Least Squares	F-statistic:	104.5			
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	1.53e-118			
Time:	13:59:16	Log-Likelihood:	155.53			
No. Observations:	2675	AIC:	-297.1			
Df Residuals:	2668	BIC:	-255.8			
Df Model:	6					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	0.2193	0.029	7.508	0.000	0.162	0.277
re74	-0.0020	0.001	-3.243	0.001	-0.003	-0.001
re75	-0.0021	0.001	-3.354	0.001	-0.003	-0.001
educ	-0.0004	0.002	-0.230	0.818	-0.004	0.003
age	-0.0034	0.000	-7.323	0.000	-0.004	-0.002
black	0.1431	0.011	13.235	0.000	0.122	0.164
hisp	0.0825	0.025	3.335	0.001	0.034	0.131

## Questão (v)

Fazendo  $avgre = (re74+re75)/2$ , obtém-se:

- O mínimo de  $avgre$  em JTRAIN2 e JTRAIN3 são equivalentes entre si e iguais a 0;
- O máximo em JTRAIN2 é \$ 24.376,45, enquanto em JTRAIN3 é \$146.901,00
- A média em JTRAIN2 é \$ 1.739,70, enquanto em JTRAIN3 é \$18.040,44
- O desvio padrão em JTRAIN2 é \$ 3.900,09, enquanto em JTRAIN3 é \$ 13.293,44

Assim, vê-se que a amostra em JTRAIN3 possui rendas muito maiores e espalhadas do que aquela de JTRAIN2, corroborando o que foi discutido no item (iv) - JTRAIN2 possui observações de indivíduos com rendas historicamente menores. Assim, as bases de dados não representam as mesmas populações, como fica ainda mais evidente em seus boxplots:



### Questão (vi)

Selecionando apenas os homens de JTRAIN2 com *avgre* menor que 10 e fazendo a regressão solicitada na questão, vê-se que a estimativa de *train* é 1,583 com uma estatística *t* de 2,503 e um p-valor de 0,013.

Fazendo o mesmo para os dados de JTRAIN3, o coeficiente de *train* é 1,8445 com uma estatística *t* de 2,065 e um p-valor de 0,039.

Assim, para a subpopulação de homens de baixa renda, as amostras de JTRAIN2 (dados experimentais) e JTRAIN3 (dados reais) são muito parecidas, uma vez que é justamente essa população que mais procura treinamento (de fato, 179 das 185 pessoas que realizaram o treinamento tem *avgre* menor que 10), o que vai de acordo com as hipóteses da questão (iv).

### Questão (vii)

Usando apenas os homens que estavam desempregados em 1974 e 1975 ( $unemp_{74} = unemp_{75} = 1$ ), a estimativa de *train* usando os dados de JTRAIN2 é similar à original - 1,84 agora vs. 1,79 com o conjunto de dados completo - e é ainda mais significante. Já usando a base JTRAIN3, a estimativa de *train* salta para 3,8 – antes, era -15,2 - com uma estatística *t* de 4,3.

Os sumários das regressões podem ser vistos abaixo (JTRAIN2 e JTRAIN3, respectivamente):

```
# Regressao simples de re78 sobre train em TRAIN2_74_75
x = train2_74_75[['train']]
y = train2_74_75['re78']
Regressao_Multipla(x,y)
```

#### OLS Regression Results

Dep. Variable:	re78	R-squared:	0.025
Model:	OLS	Adj. R-squared:	0.022
Method:	Least Squares	F-statistic:	7.144
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	0.00797
Time:	14:38:30	Log-Likelihood:	-879.83
No. Observations:	280	AIC:	1764.
Df Residuals:	278	BIC:	1771.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	4.1124	0.430	9.563	0.000	3.266	4.959
train	1.8421	0.689	2.673	0.008	0.485	3.199

```
# Regressao simples de re78 sobre train em TRAIN3_74_75
x = train3_74_75[['train']]
y = train3_74_75['re78']
Regressao_Multipla(x,y)
```

#### OLS Regression Results

Dep. Variable:	re78	R-squared:	0.064
Model:	OLS	Adj. R-squared:	0.061
Method:	Least Squares	F-statistic:	18.52
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	2.36e-05
Time:	14:38:41	Log-Likelihood:	-916.01
No. Observations:	271	AIC:	1836.
Df Residuals:	269	BIC:	1843.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	2.1512	0.560	3.838	0.000	1.048	3.255
train	3.8033	0.884	4.303	0.000	2.063	5.543

### Questão (viii)

É importante termos populações representativas porque, como mostrado no item (v), JTRAIN3 possui observações de homens com renda muito alta e que, por isso, não escolheram participar do treinamento, o que faz com que a estimativa de treinamento fique negativa – uma vez que apenas homens de menor renda escolheram participar.

Quando controlado pela renda média de anos anteriores ou até mesmo pelo histórico de emprego, as duas bases de dados entregam estimativas muito parecidas e que indicam o efeito benéfico do treinamento para aqueles que participaram.

## Exercício C11

### Questão (i)

Fazendo a regressão de *mrd rte* sobre *exec* e *unem*, vê-se que a quantidade de execuções nos últimos 3 anos é tanto economicamente como estatisticamente não significante, ou seja, o modelo mostra que não há evidência de um efeito dissuasor da pena de morte sobre a taxa de homicídios por 100.000 pessoas.

```
## Pegando apenas os dados de 1993
df = df.loc[df['year'] == 93]

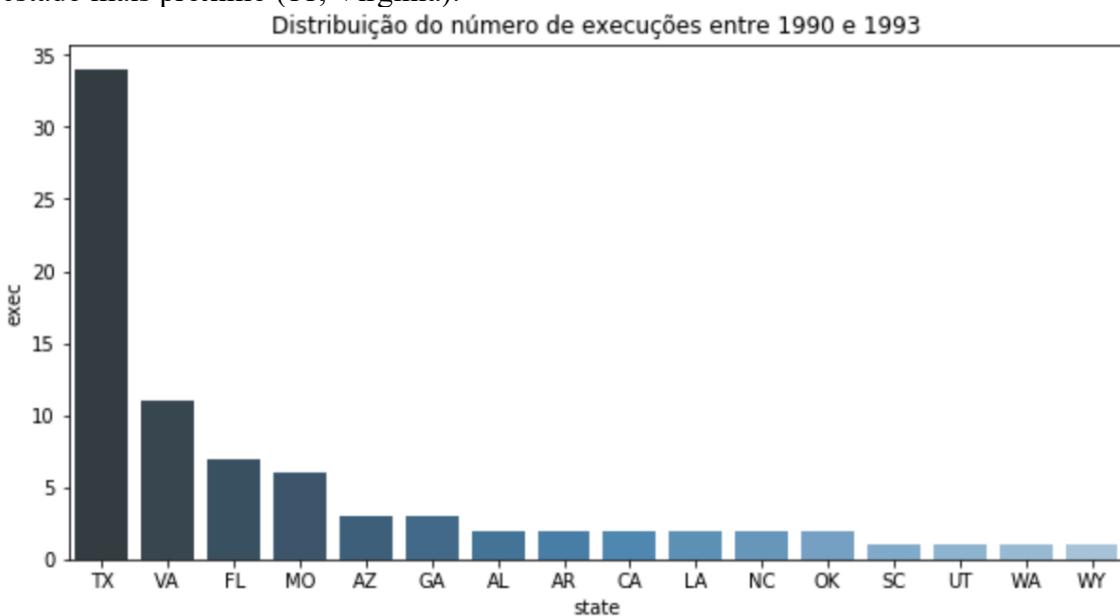
# Fazendo a regressão
x = df[['exec', 'unem']]
y = df['mrd rte']
Regressao_Multipla(x,y)
```

```
OLS Regression Results
=====
Dep. Variable:          mrd rte    R-squared:           0.115
Model:                 OLS     Adj. R-squared:      0.078
Method:                Least Squares   F-statistic:         3.126
Date:      Thu, 18 Feb 2021   Prob (F-statistic): 0.0529
Time:      15:03:48            Log-Likelihood:   -189.70
No. Observations:      51        AIC:                  385.4
Df Residuals:          48        BIC:                  391.2
Df Model:               2
Covariance Type:       nonrobust
=====
              coef    std err      t      P>|t|      [0.025      0.975]
-----
```

	coef	std err	t	P> t	[0.025	0.975]
const	-6.5686	6.315	-1.040	0.303	-19.266	6.128
exec	0.0849	0.288	0.295	0.769	-0.494	0.663
unem	2.4158	0.981	2.463	0.017	0.443	4.388

### Questão (ii)

De 1990 a 1993, o Texas registrou 34 execuções, mais que o triplo do segundo estado mais próximo (11, Virginia):



Adicionando uma *dummy* para o Texas (equivalente a retirá-lo da amostra, como visto no exercício C2), obtém-se os resultados abaixo, indicando que o coeficiente da *dummy* não é estatisticamente significante. Assim, Texas não é um *outlier* importante para a nossa análise e, portanto, deve ser mantido no modelo.

```
## Adicionando uma dummy para texas
df_93['tx'] = df_93['state'].apply(lambda i: 1 if i == 'TX' else 0)

## Fazendo a regressão
x = df_93[['exec', 'unem', 'tx']]
y = df_93['mrd rte']
Regressao_Multipla(x,y)
```

Dep. Variable:	mrd rte	R-squared:	0.117			
Model:	OLS	Adj. R-squared:	0.061			
Method:	Least Squares	F-statistic:	2.079			
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	0.116			
Time:	15:29:08	Log-Likelihood:	-189.65			
No. Observations:	51	AIC:	387.3			
Df Residuals:	47	BIC:	395.0			
Df Model:	3					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	-6.8265	6.426	-1.062	0.293	-19.753	6.100
exec	0.2949	0.717	0.411	0.683	-1.148	1.738
unem	2.4304	0.991	2.452	0.018	0.436	4.425
tx	-8.3123	25.971	-0.320	0.750	-60.558	43.934

### Questão (iii)

Adicionando a taxa de homicídios por 100.000 habitantes de três anos atrás nas variáveis exógenas,  $\beta_{exec}$  se torna negativo e significante ao nível de 5%, registrando um p-valor de 0,023. *Mrtde\_90* também é estatisticamente muito significativo, com uma estatística *t* de 65,319, o que indica que há características estruturais e temporalmente rígidas de cada estado que afetam a taxa de homicídios.

```
## Modificando o dataframe para conseguir a taxa de homicídios defasada
df_90_93 = df.loc[(df['year'] == 90) | (df['year'] == 93)]
df_90_93['mrd rte_90'] = df_90_93['mrd rte'].shift(1)
df_93 = df_90_93.loc[df_90_93['year'] == 93]
```

```
## Fazendo a regressão
x = df_93[['exec', 'unem', 'mrd rte_90']]
y = df_93['mrd rte']
Regressao_Multipla(x,y)
```

Dep. Variable:	mrd rte	R-squared:	0.990			
Model:	OLS	Adj. R-squared:	0.990			
Method:	Least Squares	F-statistic:	1609.			
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	2.34e-47			
Time:	15:39:28	Log-Likelihood:	-74.461			
No. Observations:	51	AIC:	156.9			
Df Residuals:	47	BIC:	164.6			
Df Model:	3					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	0.8826	0.676	1.306	0.198	-0.477	2.242
exec	-0.0713	0.030	-2.342	0.023	-0.133	-0.010
unem	-0.0903	0.110	-0.819	0.417	-0.312	0.132
mrd rte_90	1.0098	0.015	65.319	0.000	0.979	1.041

#### Questão (iv)

Adicionando uma *dummy* para Texas no modelo da questão (iii), o coeficiente de *exec* se torna estatisticamente não significante - uma vez que boa parte das execuções acontecem no Texas. O coeficiente da *dummy* é estatisticamente não significante, o que faz com que seja melhor deixá-lo na análise, ou seja, a observação do estado do Texas continua não sendo um *outlier* significante.

```
## Adicionando uma dummy para texas
```

```
df_93['tx'] = df_93['state'].apply(lambda i: 1 if i == 'TX' else 0)
```

```
## Fazendo a regressão com a dummy de texas
```

```
x = df_93[['exec', 'unem', 'mrdrt_90', 'tx']]
```

```
y = df_93['mrdrt']
```

```
Regressao_Multipla(x,y)
```

#### OLS Regression Results

Dep. Variable:	mrdrt	R-squared:	0.990			
Model:	OLS	Adj. R-squared:	0.990			
Method:	Least Squares	F-statistic:	1185.			
Date:	Thu, 18 Feb 2021	Prob (F-statistic):	9.55e-46			
Time:	15:46:28	Log-Likelihood:	-74.383			
No. Observations:	51	AIC:	158.8			
Df Residuals:	46	BIC:	168.4			
Df Model:	4					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	0.8491	0.688	1.234	0.223	-0.536	2.234
exec	-0.0454	0.076	-0.599	0.552	-0.198	0.107
unem	-0.0879	0.112	-0.788	0.435	-0.313	0.137
mrdrt_90	1.0095	0.016	64.648	0.000	0.978	1.041
tx	-1.0243	2.741	-0.374	0.710	-6.542	4.494

#### Anexo: definições das funções utilizadas

```
def Teste_t_Dois_Coeficientes_Iguais(x, y, Coeficientes_Testados_para_serem_iguais, Nivel_de_Significância = 0.05):
    """
    Função que executa um teste t para verificar se dois coeficientes são iguais.

    x: lista ou array com os valores das variáveis independentes;
    y: lista ou array com os valores da variável dependente;
    Coeficientes_Testados_para_serem_iguais: array com os valores dos coeficientes que querem ser testados;
    Nivel_de_Significância: nível de significância do teste. Caso branco, o nível de significância padrão é de 5%.
    """

    ##Fazendo a regressão do modelo irrestrito
    Regressao_Multipla(x, y)
    clear_output()

    #Fazendo o objeto de lista que será usado no teste
    Teste =[0]
    Num_de_Variaveis = 1

    for i in list(x):
        if i not in list(Coeficientes_Testados_para_serem_iguais):
            Teste.append(0)
        elif (Num_de_Variaveis % 2 == 0):
            Teste.append(-1)
        else:
            Teste.append(1)
        Num_de_Variaveis += 1

    Teste_t = Resultado.t_test(Teste)
    print(f"A estatística do teste é {Teste_t.tvalue}, o que resulta em um p-valor de {Teste_t.pvalue}")
```

```

def Regressao_Multipla(x, y, constante = "S", robusta = "N"):
    """
        Função que calcula uma regressão múltipla, sendo, por default, computada com um intercepto e com erros padrões não robustos.
        x: lista ou array com os valores das variáveis independentes;
        y: lista ou array com os valores da variável dependente;
        constante: "S" para regressão com intercepto e qualquer outro valor para sem intercepto. Caso em branco, a regressão é computada com intercepto;
        robusta: "N" para regressão com erros-padrão tradicionais e qualquer outro valor para erros-padrões robustos. Caso em branco, a regressão é computada com erros-padrão comuns.
    """

    global Resultado, Lista_ychapeu, Resíduos, SQR, EPR

    #adicionando uma constante ao modelo de Ordinary Least Squares (OLS)
    if constante == "S":
        X = sm.add_constant(x)
    else:
        X = x
    #Criando o Modelo levando em conta a opção de ser uma regressão robusta p/ heteroscedasticidade ou não
    Modelo = sm.OLS(y,X)

    if robusta == "N":
        Resultado = Modelo.fit()
    else:
        Resultado = Modelo.fit(cov_type = 'HC1', use_t = True)

    Lista_ychapeu = Resultado.predict()
    Resíduos = y - Lista_ychapeu

    #Calculando o Erro Padrão da Regressão (EPR)
    SQR =sum([i**2 for i in Resíduos])
    Número_de_Observações = len(y)
    GL = Número_de_Observações - len(Resultado.params)
    VariânciaReg = SQR/GL
    EPR = math.sqrt(VariânciaReg)

    ##Printando o Resultado
    print(f"O erro padrão da regressão é {round(EPR,5)} e a SQR é {round(SQR,5)}\n")
    print(Resultado.summary())

    print("\nPara ver os valores previstos ou os resíduos, basta chamar 'Lista_ychapeu' e 'Resíduos'.")
    print("Os resultados do modelo podem ser obtidos através de métodos usando a variável 'Resultado'.")
    print("Valores de condição maiores que 20 indicam problemas de multicolinearidade.
    Para ver como achar esse número, entre em https://www.statsmodels.org/dev/examples/notebooks/generated/ols.html")

def Teste_F(x, y, Restrições, Nivel_de_Significância = 0.05):
    """
        Função que calcula um teste F e dá o resultado teste de hipótese para o caso de todas as restrições serem conjuntamente estatisticamente não-significantes.
        x: lista ou array com os valores das variáveis independentes;
        y: lista ou array com os valores da variável dependente;
        Restrições: lista ou array com os valores a serem tirados do modelo restrito;
        Nivel_de_Significância: nível de significância do teste. Caso branco, o nível de significância padrão é de 5%.
    """

    ##Definindo as variáveis de cada modelo
    #para testar igualdade dos coeficientes, F2, p_valueF2 = results.Ftest(['ACT', 'skipped'], equal=True)
    ModeloIrrestrito = list(x)
    ModeloRestrito = []
    Restrições = list(Restrições)

    Número_de_Observações = len(y)
    GL_ir = Número_de_Observações - (len(ModeloIrrestrito) + 1)
    GL_r = len(Restrições)

    for i in ModeloIrrestrito:
        if i not in Restrições:
            ModeloRestrito.append(i)

    ##Fazendo as regressões de cada modelo
    Regressao_Multipla(x, y)
    SQR_ir = SQR
    VariânciaReg_ir = EPR**2

    Regressao_Multipla(df[ModeloRestrito], y)
    SQR_r = SQR

    #Limpando a tela
    clear_output()

    ##Calculando F
    F = (SQR_r - SQR_ir)/(len(Restrições)*VariânciaReg_ir)

    ##Calculando o P-valor de F
    P_valor = stats.f.sf(F,GL_r,GL_ir)

    if Nivel_de_Significância > P_valor:
        print(f"O valor de F é {round(F,3)} e seu p-valor é {round(P_valor,7)}. Portanto, rejeita-se H0 à significância de {Nivel_de_Significância*100}%.")
    else:
        print(f"O valor de F é {round(F,3)} e seu p-valor é {round(P_valor,7)}. Portanto, não se rejeita H0 à significância de {Nivel_de_Significância*100}%.")

```