

Regressão Linear Multivariada

Mini-curso de Introdução a ML e AI

Mário Olímpio de Menezes

Maio/2020

Regressão Linear Multivariada

- Quando temos mais do que uma variável preditora (explicativa), a regressão linear simples se transforma em **regressão linear multivariada**.
 - Uma regressão quadrática tem duas preditoras (X e X^2).
 - A regressão cúbica tem três preditoras (X, X^2 , e X^3).
 - Uma regressão polinomial é um caso especial de uma regressão múltipla

- A habilidade de uma variável independente **adicional** melhorar o modelo de regressão está relacionada não somente à sua correlação com a variável dependente, mas também às correlações da variável independente adicional com as outras variáveis independentes já presentes no modelo.
 - **Colinearidade** é a associação, medida como correlação, entre duas variáveis independentes.
 - **Multicolinearidade** se refere à correlação entre três ou mais variáveis independentes, (evidenciada quando uma é *regredida* em relação às outras).

O impacto da multicolinearidade é reduzir qualquer poder preditivo de uma variável independente única pela extensão a qual ela está associada com outra variável independente.

- Conforme a colinearidade aumenta, a variância única explicada por cada variável independente diminui e o percentual de predição compartilhada aumenta.
 - Como esta predição compartilhada somente conta uma vez, a predição total *aumenta* muito mais lentamente quando variáveis altamente correlacionadas são adicionadas ao modelo.
- Para maximizar a predição de um dado número de variáveis independentes, devemos procurar aquelas que tenham baixa multicolinearidade com outras variáveis independentes mas que **também** tenham alta correlações com a variável dependente.

Criando Variáveis Adicionais

- O relacionamento básico representado na regressão múltipla é a associação *linear* entre a variável dependente (métrica) e as variáveis independentes (também métricas).
- Um problema frequentemente encontrado é a incorporação de dados não-métricos, tais como gênero, ocupação, etc., na equação de regressão.
 - A regressão múltipla é limitada a dados métricos (numéricos).
- Outro problema é a incapacidade de se representar diretamente relacionamentos não lineares.
- Quando temos estas situações, novas variáveis devem ser criadas por **transformações**:
 - Esta é a maneira de incorporarmos variáveis não-métricas ou para representar quaisquer outros efeitos além de relacionamentos não lineares.
 - Outro uso de *transformações* é para acertar violações de alguma das premissas (hipóteses) estatísticas.
- Duas razões básicas para transformarmos variáveis:
 - Melhorar ou modificar o relacionamento entre as variáveis dependente e independentes.
 - Habilitar o uso de variáveis não métricas na equação de regressão.

Exemplo 1

Montgomery, Peck e Vining - Pacoate MPV

A variável resposta é o calor liberado por grama de cimento, e as variáveis explicativas quanto tem de cada componente; são elas:

- Y = heat evolved in calories per gram of cement
- X_1 = tricalcium aluminate
- X_2 = tricalcium silicate
- X_3 = tetracalcium alumino ferrite
- X_4 = dicalcium silicate

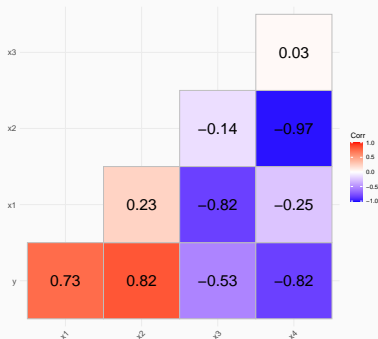
Obtendo os dados

```
> library(MPV)

> cimento <- MPV::cement
> names(cimento)
[1] "y" "x1" "x2" "x3" "x4"

> str(cimento)
'data.frame': 13 obs. of 5 variables:
 $ y : num 78.5 74.3 104.3 87.6 95.9 ...
 $ x1: num 7 1 11 11 7 11 3 1 2 21 ...
 $ x2: num 26 29 56 31 52 55 71 31 54 47 ...
 $ x3: num 6 15 8 8 6 9 17 22 18 4 ...
 $ x4: num 60 52 20 47 33 22 6 44 22 26 ...

> library(ggcorrplot)
> ggcorrplot(cor(cimento), type = "lower", lab = TRUE,
  colors = c("blue", "white", "red"))
```



- Olhando a matriz de correlação identificamos dois pares de variáveis com correlações significativas entre si: (x_1, x_3) correlação -0.82 e (x_2, x_4) correlação de -0.97
- Estas variáveis, quando as adicionarmos todas ao modelo, vão *bagunçar* o algoritmo e os resultados serão comprometidos.

Construindo o Modelo

```
> modcim <- lm(y ~ x1 + x2 + x3 + x4, data = cimento)
> summary(modcim)
```

Call:

```
lm(formula = y ~ x1 + x2 + x3 + x4, data = cimento)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.1750	-1.6709	0.2508	1.3783	3.9254

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	62.4054	70.0710	0.891	0.3991
x1	1.5511	0.7448	2.083	0.0708
x2	0.5102	0.7238	0.705	0.5009
x3	0.1019	0.7547	0.135	0.8959
x4	-0.1441	0.7091	-0.203	0.8441

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.446 on 8 degrees of freedom

Multiple R-squared: 0.9824, Adjusted R-squared: 0.9736

F-statistic: 111.5 on 4 and 8 DF, p-value: 4.756e-07

- Olhando o modelo, vemos que **todos** os parâmetros estão sem significância estatística, *como tínhamos previsto* a partir dos dados da matriz de correlação: **multicolinearidade**
- Apesar disto, o modelo tem um R^2 ajustado *maravilhoso*, de 0.9736
- Mas, precisamos acertar o modelo, removendo as variáveis que não têm significância estatística.
- Começamos pela última variável, x_4

Atualizando o Modelo

```
> modcim <- update(modcim, . ~ . - x4)
> summary(modcim)
```

Call:

```
lm(formula = y ~ x1 + x2 + x3, data = cimento)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.2543	-1.4726	0.1755	1.5409	3.9711

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	48.19363	3.91330	12.315	6.17e-07 ***
x1	1.69589	0.20458	8.290	1.66e-05 ***
x2	0.65691	0.04423	14.851	1.23e-07 ***
x3	0.25002	0.18471	1.354	0.209

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.312 on 9 degrees of freedom

Multiple R-squared: 0.9823, Adjusted R-squared: 0.9764

F-statistic: 166.3 on 3 and 9 DF, p-value: 3.367e-08

- Usando a função update vamos *atualizar* o nosso modelo, removendo a variável x4
- Fazendo sumário do modelo atualizado, já identificamos uma melhora significativa nos parâmetros.
- Apenas a variável x3 continua sem significância estatística, e portanto, vamos removê-la do modelo.

Atualizando o Modelo

```
> modcim <- update(modcim, . ~ . - x3)
```

```
> summary(modcim)
```

Call:

```
lm(formula = y ~ x1 + x2, data = cimento)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.893	-1.574	-1.302	1.363	4.048

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	52.57735	2.28617	23.00	5.46e-10 ***
x1	1.46831	0.12130	12.11	2.69e-07 ***
x2	0.66225	0.04585	14.44	5.03e-08 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.406 on 10 degrees of freedom

Multiple R-squared: 0.9787, Adjusted R-squared: 0.9744

F-statistic: 229.5 on 2 and 10 DF, p-value: 4.407e-09

- Vemos agora que todos os parâmetros têm significância estatística, e o R^2 ajustado do modelo está *muito bom* , 0.9744.
- E também vemos que o p-value do modelo é muito bom (4.407×10^{-9}), ou seja, praticamente zero.
- Isso significa que este modelo é estatisticamente diferente do modelo nulo, ou seja, somente a aleatoriedade (sem nenhum parâmetro) **não** consegue explicar a variabilidade de y – nossa variável resposta.
- O teste com a ANOVA mostra exatamente isso. Veja o Pr(>F) com valor **zero** (arredondamento do $4.407 \cdot 10^{-9}$)

```
> modelonulo <- lm(y ~ 1, data = cimento)
```

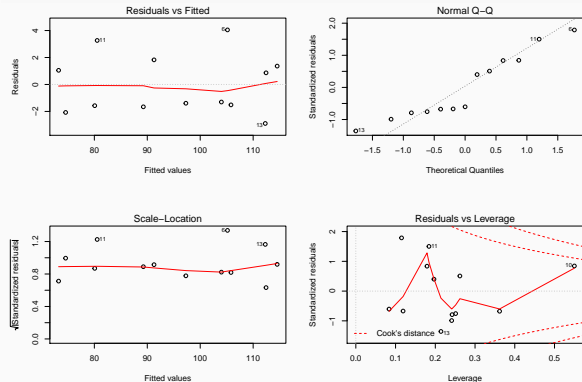
```
> anova(modelonulo, modcim)
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
12	2715.76308	NA	NA	NA	NA
10	57.90448	2	2657.859	229.5037	0

Diagnóstico do Modelo

```
> par(mfrow = c(2, 2))
```

```
> plot(modcim)
```



- Analisaremos os dois gráficos superiores: *Residuals vs Fitted* e *Normal Q-Q*.
- O Gráfico dos *Residuals vs Fitted values* é onde analisamos se a variabilidade dos resíduos tem dependência com os valores ajustados, se aumenta ou diminui, se demonstra algum padrão, etc. No gráfico ao lado, não conseguimos identificar este tipo de comportamento, a amplitude de variação dos resíduos é basicamente a mesma ao longo de todos os valores ajustados.
- O Gráfico *Normal Q-Q* analisamos se os resíduos têm distribuição normal. Quando isso acontece, os pontos do gráfico (topo, à direita), devem permanecer sobre a linha tracejada, sem grandes desvios, principalmente nas extremidades, o que parece acontecer.
- O Gráfico *Scale-Location* tem basicamente a mesma informação do *Residuals vs Fitted*, mas com valores absolutos dos resíduos padronizados; serve também para identificarmos dependência em relação aos valores ajustados.
- O Gráfico *Residuals vs Leverage* aponta observações que podem precisar de atenção por serem *outliers*, pontos de alta alavancagem, etc., que distorcem o ajuste, prejudicando sua qualidade. Não vamos analisar este último gráfico neste material.

Diagnóstico do Modelo

```
> library(gvlma)
> display.gvlmatests(gvlma(modcim))
```

```
ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
Level of Significance = 0.05
```

```
Call:
gvlma(x = modcim)
```

	Value	p-value	Decision
Global Stat	1.5244265	0.8223	Assumptions acceptable.
Skewness	0.5484212	0.4590	Assumptions acceptable.
Kurtosis	0.5559113	0.4559	Assumptions acceptable.
Link Function	0.0004057	0.9839	Assumptions acceptable.
Heteroscedasticity	0.4196883	0.5171	Assumptions acceptable.

- Apesar de os gráficos diagnósticos permitirem uma rápida inspeção visual se o modelo atende às premissas do Método dos Mínimos Quadrados (MMQ), esta análise tem um aspecto subjetivo que depende da habilidade do observador.
- Para tornar o diagnóstico mais objetivo, o pacote gvlma oferece uma função que faz uma avaliação global do modelo com relação ao atendimento às premissas do MMQ.
- Olhando os resultados ao lado vemos que nosso modelo atendeu a todas as premissas. Os testes estatísticos tem a hipótese nula de que o modelo atende as premissas. Pelos p-values maiores do que o nível de significância (0.05), todos os critérios indicam a aceitação da hipótese nula.

Exemplo 2 - Dataset hipotético

Este segundo exemplo utiliza um *dataset* hipotético com duas variáveis explicativas (x_1 e x_2).

Como já fiz no exemplo anterior as análises das etapas de construção e análise do modelo, comentarei apenas rapidamente a interpretação dos parâmetros do modelo – a última parte.

```
> library(readr)
> novodf <- read_csv("datasets/dadoshipot.csv")

> cor(novodf)
```

	x1	y	x2
x1	1.000000000	0.9737924	-0.002647846
y	0.973792432	1.0000000	-0.180223309
x2	-0.002647846	-0.1802233	1.000000000

Modelo Exemplo II

```
> summary(modhipot <- lm(y ~ x1 + x2, data = novodf))
```

Call:

```
lm(formula = y ~ x1 + x2, data = novodf)
```

Residuals:

Min	1Q	Median	3Q	Max
-44.584	-24.565	-3.266	22.330	63.201

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1665.9399	220.2398	-7.564	1.11e-05 ***
x1	10.7812	0.4743	22.730	1.35e-10 ***
x2	-15.6050	3.7616	-4.149	0.00162 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 34.93 on 11 degrees of freedom

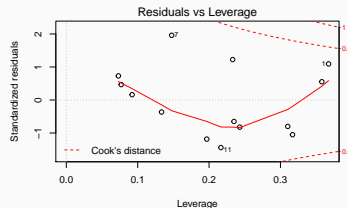
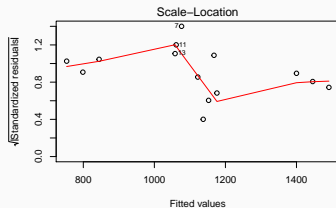
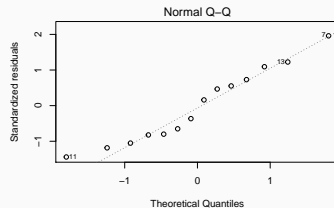
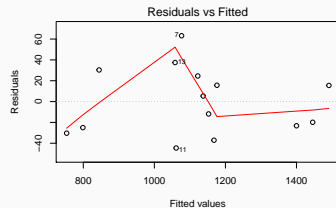
Multiple R-squared: 0.9798, Adjusted R-squared: 0.9762

F-statistic: 267.2 on 2 and 11 DF, p-value: 4.742e-10

Diagnóstico do Modelo

```
> par(mfrow = c(2, 2))
```

```
> plot(modhipot)
```



Diagnóstico do Modelo

```
> display.gvlmatests(gvlma(modhipot))
```

```
ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS  
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:  
Level of Significance = 0.05
```

```
Call:
```

```
gvlma(x = modhipot)
```

	Value	p-value	Decision
Global Stat	1.9038	0.7535	Assumptions acceptable.
Skewness	0.3045	0.5811	Assumptions acceptable.
Kurtosis	0.4901	0.4839	Assumptions acceptable.
Link Function	0.9673	0.3254	Assumptions acceptable.
Heteroscedasticity	0.1419	0.7064	Assumptions acceptable.

Interpretando os resultados do modelo

```
> summary(modhipot)
```

Call:

```
lm(formula = y ~ x1 + x2, data = novodf)
```

Residuals:

Min	1Q	Median	3Q	Max
-44.584	-24.565	-3.266	22.330	63.201

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1665.9399	220.2398	-7.564	1.11e-05 ***
x1	10.7812	0.4743	22.730	1.35e-10 ***
x2	-15.6050	3.7616	-4.149	0.00162 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 34.93 on 11 degrees of freedom

Multiple R-squared: 0.9798, Adjusted R-squared: 0.9762

F-statistic: 267.2 on 2 and 11 DF, p-value: 4.742e-10

```
> summary(novodf)
```

x1	y	x2
Min. :289.3	Min. : 722.8	Min. :39.00
1st Qu.:319.3	1st Qu.:1036.5	1st Qu.:40.22
Median :319.3	Median :1140.3	Median :41.92
Mean :319.7	Mean :1120.4	Mean :42.29
3rd Qu.:323.1	3rd Qu.:1180.5	3rd Qu.:44.85
Max. :349.3	Max. :1506.8	Max. :46.02

Interpretando os resultados do modelo

- O nosso modelo final tem os coeficientes -1665.9398661, 10.7811579, -15.6050128 e podemos então escrevê-lo na forma

$$y = -1665.9399 + 10.7812 \times x1 - 15.6050 \times x2$$

- Vemos que $x1$ tem um impacto de 10.7812 no valor de y para cada variação de uma unidade, mantendo-se o valor de $x2$ constante. Por outro lado, $x2$ tem um impacto negativo de -15.6050 no valor de y para cada unidade de variação, mantendo-se o valor de $x1$ constante.

Próximo Bloco

Regressão Logística