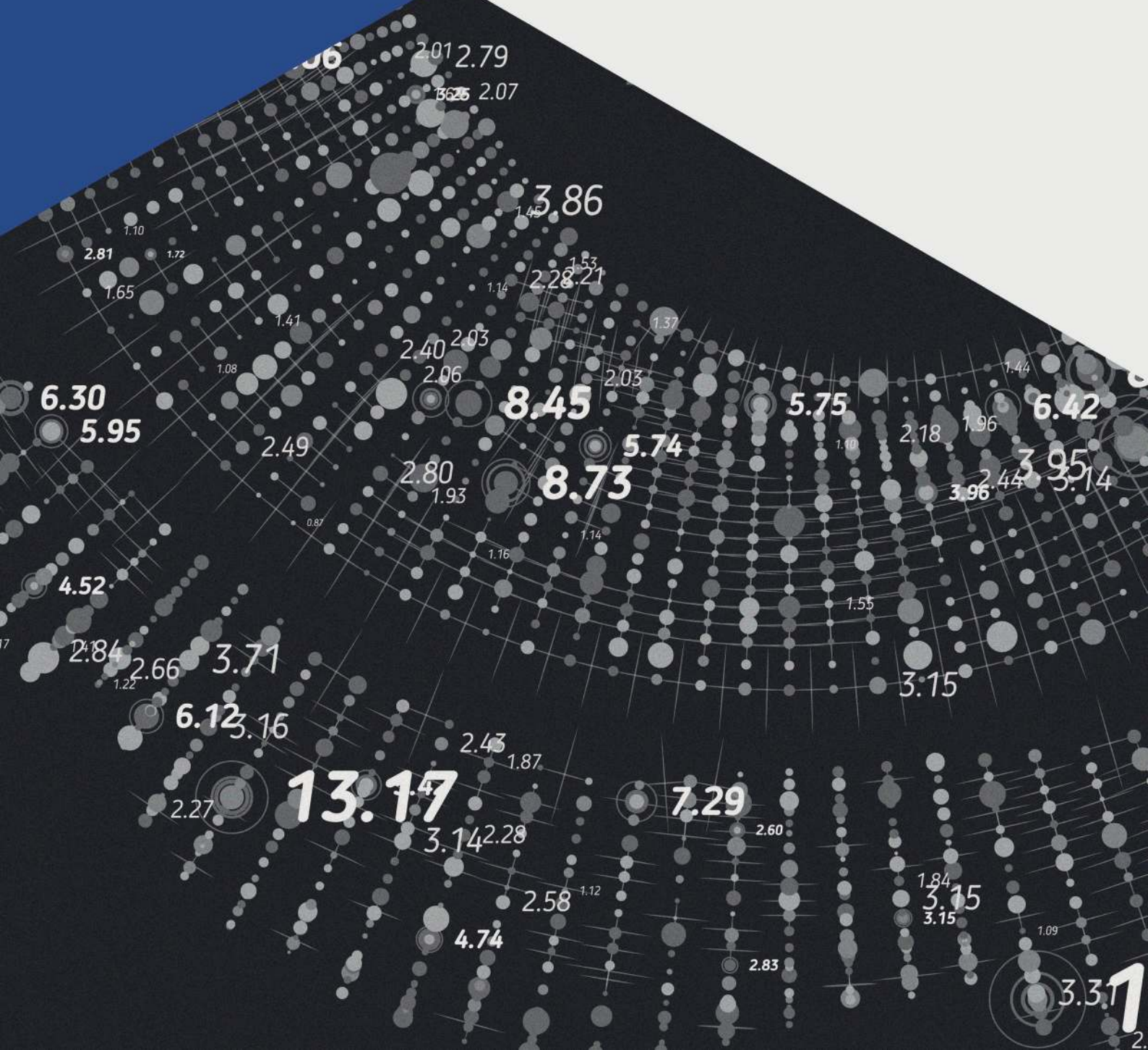
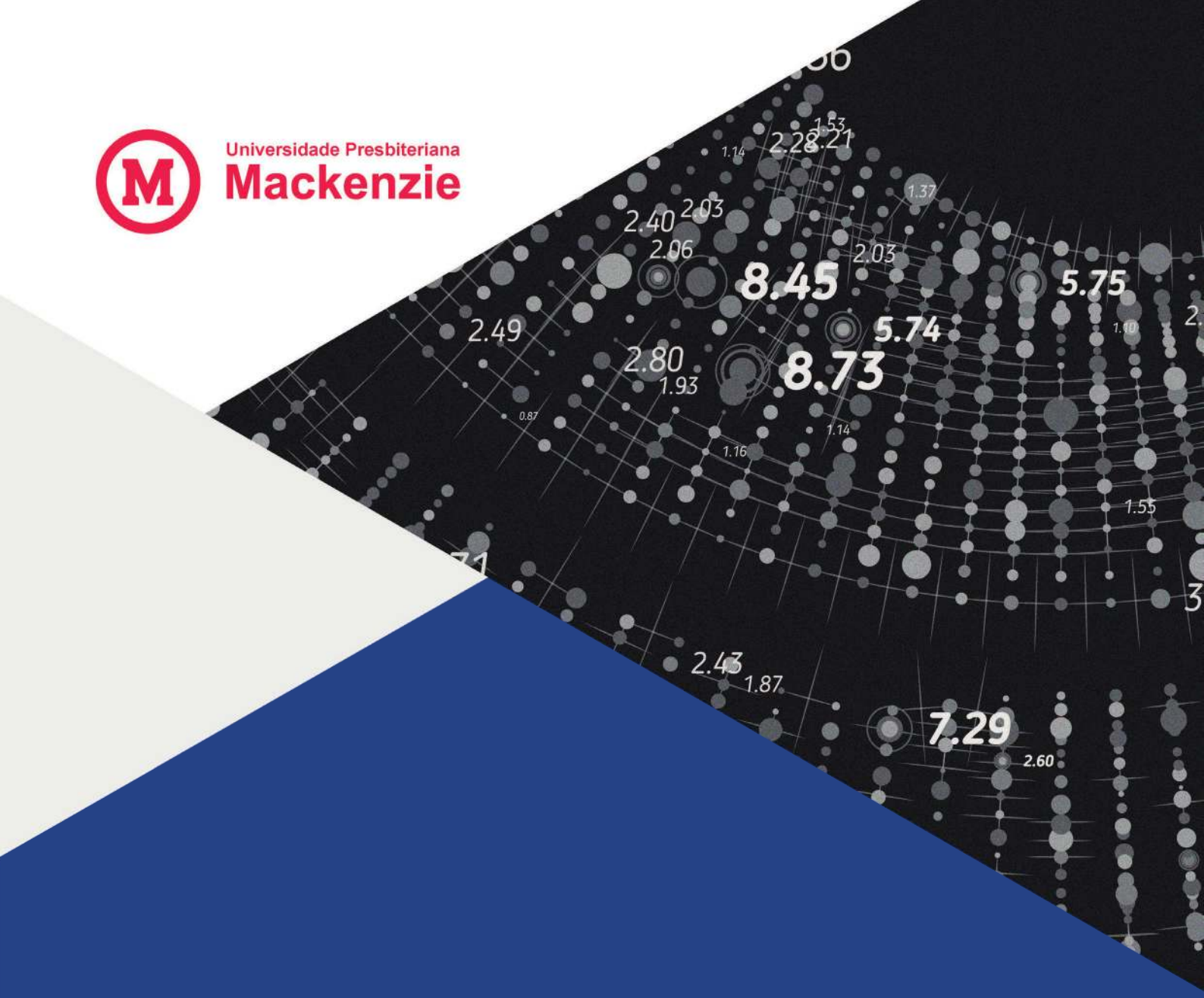
Universidade Presbiteriana
Mackenzie

CIÊNCIAS DE DADOS (BIG DATA ANALYTICS)

BIG DATA Analytics: Análise Estatística



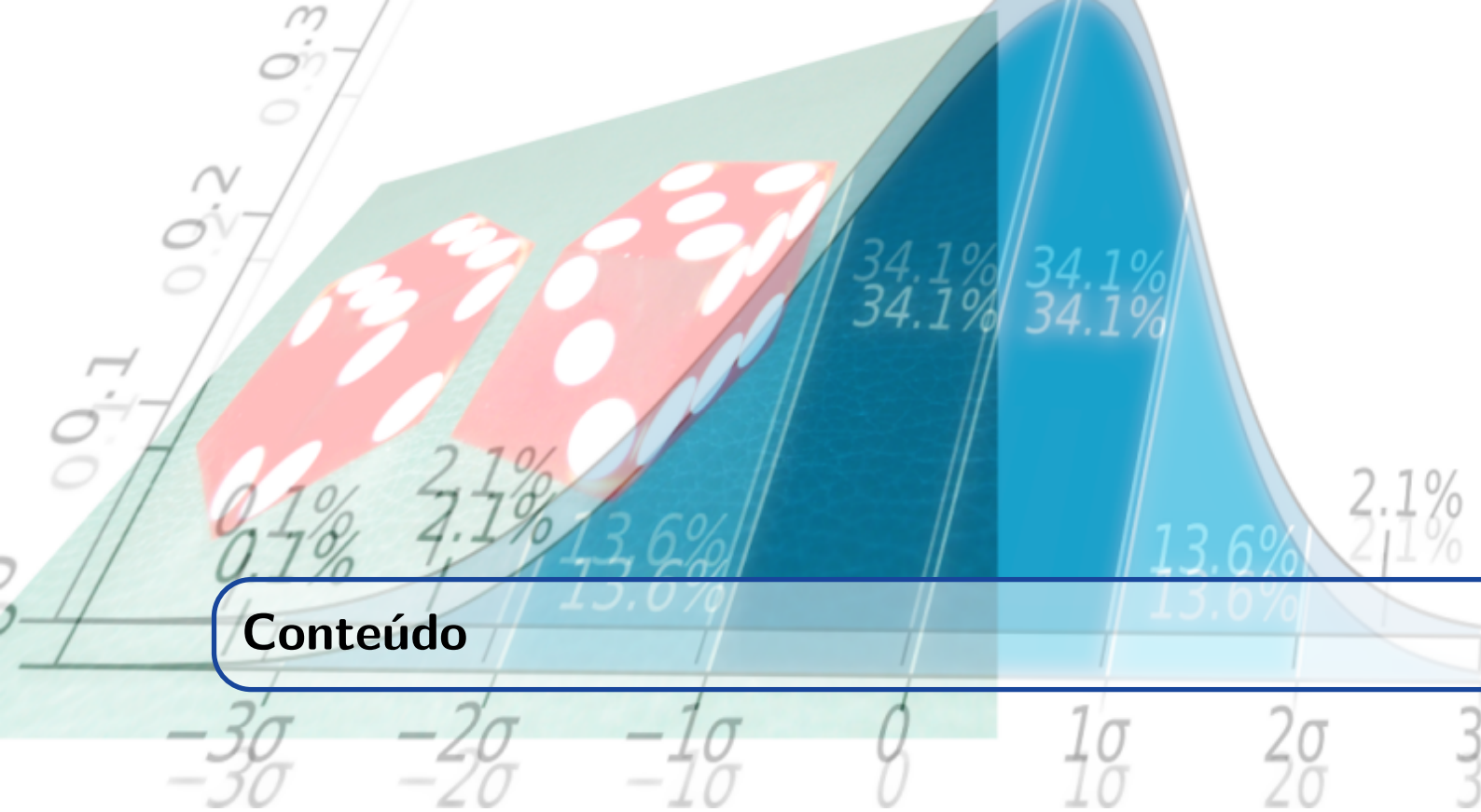
Universidade Presbiteriana
Mackenzie



Material Complementar - Trilha 05

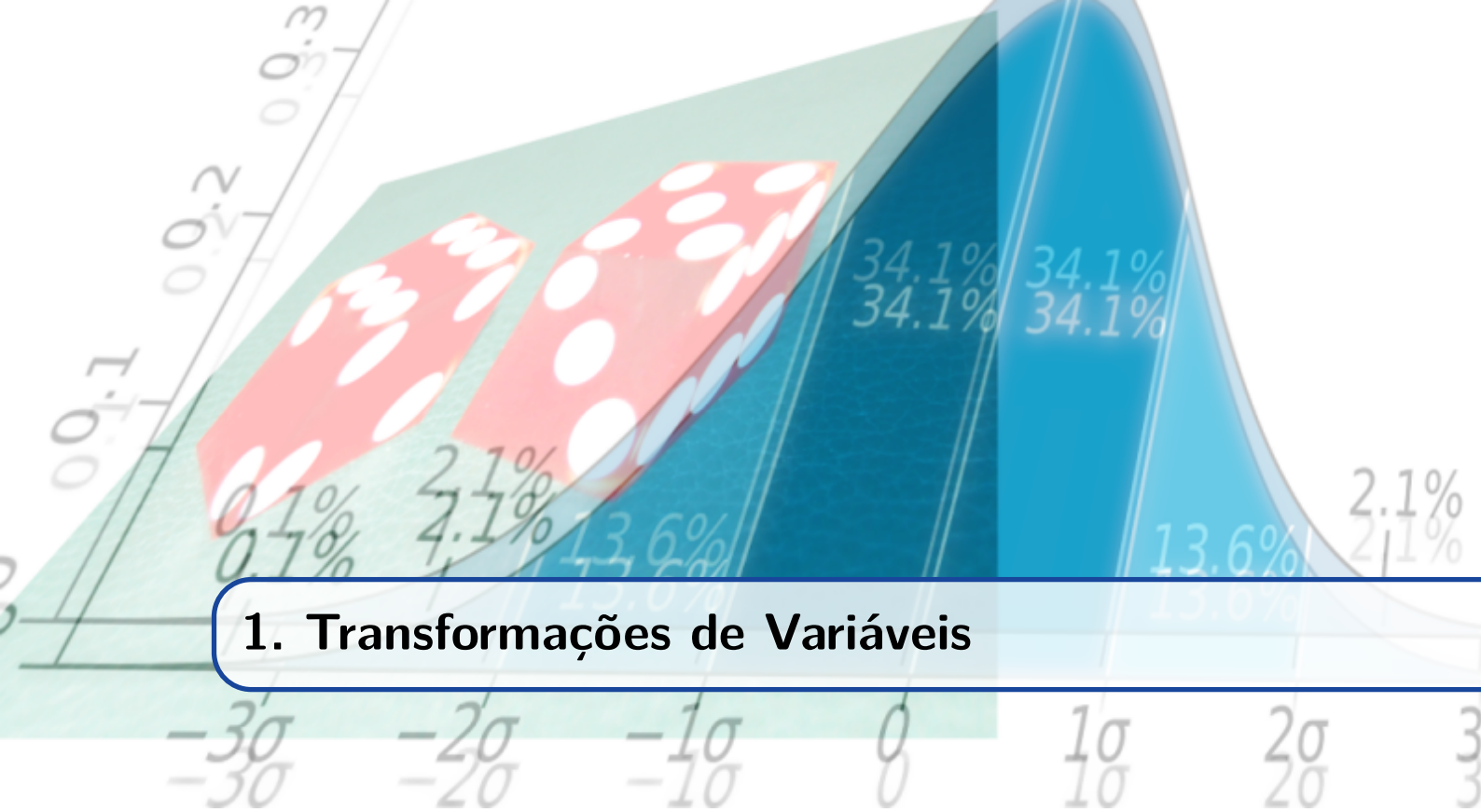
Transformação de Variáveis e Seleção de Modelos

Mário Olímpio de Menezes



Conteúdo

1 Transformação de Variáveis



1. Transformações de Variáveis

Regressão Linear Multivariada

A técnica de Regressão Linear Multivariada, que é o foco desta Trilha, situa-se dentro do grande tópico das técnicas de Análise Multivariada.

A **Análise Multivariada** se refere a todas as técnicas que analisam simultaneamente múltiplas medidas de indivíduos ou objetos sob investigação. Assim, quaisquer análises simultâneas de mais do que duas variáveis pode ser considerada uma *análise multivariada* (HAIR JR et al. (2014)).

Dados multivariados surgem quando se medem várias variáveis para cada observação na amostra. A maioria dos conjuntos de dados coletados por pesquisadores em todas as áreas da ciência são multivariados (JOHNSON; WICHERN (1992)).

Codificação Fictícia (dummy coding)

Um problema frequentemente encontrado na regressão múltipla é a incorporação de dados não-métricos, tais como gênero, ocupação, etc., na equação de regressão, ou seja, variáveis categóricas. Isso porque a regressão múltipla é limitada a dados métricos (numéricos).

Quando temos variáveis destes tipos (nominal ou ordinal), elas devem ser transformadas em variáveis numéricas utilizando um esquema de codificação; dentre os esquemas possíveis de transformação, há a codificação de zeros e uns, chamada de *dummy coding*. Assumindo que x_i seja um fator com k níveis, a submatriz de $\mathbf{X}$ correspondente a x_i é uma matriz $n \times k$ de zeros

e uns, onde o j-ésimo elemento na i-ésima linha é um quando x_{i1} estiver no j-ésimo nível.

Suponha um conjunto de dados que contém uma variável categórica (*nominal*) indicando o continente de origem de uma observação (indivíduo, etc), conforme mostrado no Quadro 1:

Quadro 1: Conjunto de dados exemplo – uma variável categórica

Observ.	Continente
01	Africa
02	Europa
03	America
04	Asia
05	Oceania

Fonte: Elaborado pelo próprio autor

Utilizando a abordagem de *dummy coding*, esta variável categórica *Continente* seria transformada conforme mostra o Quadro 2:

Quadro 2: Codificação *dummy* – uma variável categórica

Observ.	Cont.Africa	Cont.Europa	Cont.America	Cont.Asia	Cont.Oceania
01	1	0	0	0	0
02	0	1	0	0	0
03	0	0	1	0	0
04	0	0	0	1	0
05	0	0	0	0	1

Fonte: Elaborado pelo próprio autor

Outro esquema de codificação de variáveis categóricas muito utilizado é denominado *indicator coding*, no qual cada categoria (nível) da variável não métrica também é representado ou por zero ou por um, mas utiliza-se uma **categoria de referência**, isto é, um *nível* da variável categórica será omitido, e será representado por zero.

Utilizando os dados do exemplo anterior e fazendo **Africa** como o nível de referência, a codificação do tipo *indicator coding* é mostrada no Quadro 3:

Quadro 3: Codificação *indicator coding* – uma variável categórica

Observ.	Cont.Africa	Cont.Europa	Cont.America	Cont.Asia	Cont.Oceania
01	0	0	0	0	0
02	0	1	0	0	0
03	0	0	1	0	0
04	0	0	0	1	0
05	0	0	0	0	1

Fonte: Elaborado pelo próprio autor

Ou seja, podemos excluir a coluna `Cont.Africa` da codificação, e o resultado é mostrado no Quadro 4:

Quadro 4: Codificação *indicator coding*, removendo a coluna de referência (`Cont.Africa`– uma variável categórica

Observ.	Cont.Europa	Cont.America	Cont.Asia	Cont.Oceania
01	0	0	0	0
02	1	0	0	0
03	0	1	0	0
04	0	0	1	0
05	0	0	0	1

Fonte: Elaborado pelo próprio autor

Pequeno exemplo de codificação de variáveis *dummy* no R

O **R** tem algumas facilidades para lidarmos com variáveis categóricas em regressão, isto é, para a criação de variáveis *dummy* através da codificação mencionada anteriormente.

No trecho de código mostrado abaixo, um `data.frame` fictício é criado com as seguintes variáveis:

- `observ` – identificação da observação
- `contin` – continente de origem.
- `idademax` – idade máxima observada.
- `numvmed` – número total de visitas ao médico.

Suponha que queiramos modelar a idade máxima observada como uma função do continente de origem e do número de visitas ao médico, ou seja: `lm(idademax ~ numvmed + contin)`

A variável `contin` é categórica, com 5 níveis. Inicialmente, não vamos fazer nenhuma codificação explícita, apenas deixaremos o **R** fazê-lo.

Não estamos interessados em analisar o modelo sob o ponto de vista de significância estatística e nem de poder de explicação, apenas a questão da codificação da variável categórica.

```
> set.seed(123456)
> observ <- sample(1:99,size = 20)
> contin <- sample(rep(c("Africa","America","Europa","Asia","Oceania"),100),20)
> idademax <- rbinom(n=20,size=100,prob=0.8)
> numvmed <- rbinom(n=20,size=20,prob=0.4)
> df <- data.frame(observ, contin, idademax, numvmed)
> df
   observ  contin idademax numvmed
1      60 Oceania       84       9
2      42 America       83       6
3      71  Europa       88       2
4      54  Europa       80      10
5      74  Europa       84      12
6       3    Asia       84       8
7      67  Africa       79       7
8      23    Asia       90       6
9      93 America       82       8
10     38  Europa       84       8
11      2    Asia       88       6
12     80 America       83       7
13     85  Europa       79       7
14     65    Asia       74       6
15     56 Oceania       82       6
16     24 Oceania       76       8
17     46  Africa       84       4
18     14    Asia       76      10
```

```
19     79  Europa       83      10
20     35  Africa       84       5
```

```
> str(df,strict.width = "wrap")
'data.frame':   20 obs. of  4 variables:
$ observ : int 60 42 71 54 74 3 67 23 93 38 ...
$ contin : Factor w/ 5 levels "Africa","America",..: 5 2 4 4 4 3 1 3 2 4
   ...
$ idademax: int 84 83 88 80 84 84 79 90 82 84 ...
$ numvmed : int 9 6 2 10 12 8 7 6 8 8 ...
```

```
> mod <- lm(idademax ~ contin + numvmed, data = df)
> summary(mod)

Call:
lm(formula = idademax ~ contin + numvmed, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-9.2342 -1.8378  0.5232  1.6152  6.7658

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   86.04093    3.56379  24.143 8.27e-13 ***
continAmerica  1.49196    3.64619   0.409    0.689
continAsia     1.36433    3.30427   0.413    0.686
continEuropa   2.63633    3.36170   0.784    0.446
continOceania -0.04459    3.72711  -0.012    0.991
numvmed       -0.69517    0.47300  -1.470    0.164
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.36 on 14 degrees of freedom
Multiple R-squared:  0.1645,    Adjusted R-squared:  -0.1338
F-statistic: 0.5514 on 5 and 14 DF,  p-value: 0.7351
```

Como observamos no código, o **R** atribuiu uma codificação para a variável categórica `contin`, definindo o continente "Africa" como referência (nível 0). Foram criadas as seguintes variáveis *dummy*:

- `continAmerica` – quando a observação for da America.

- `continAsia` – quando a observação for da Asia.
- `continEuropa` – quando a observação for da Europa.
- `continOceania` – quando a observação for da Oceania.

A variável `continAfrica` não foi criada; ela é subentendida considerando-se os outros níveis como 0.

Para cada variável *dummy* criada, temos uma *curva* de regressão, já que estas variáveis não podem existir simultaneamente, ou seja, quando uma for 1, as demais ficam como 0.

Assim, quando utilizamos a função `lm` do **R**, se a codificação das variáveis categóricas não for realizada explicitamente, como mostrado anteriormente, o **R** se encarregará de fazê-la, utilizando normalmente o último formato mostrado, isto é, com uma categoria como referência. Esta forma automática de se realizar a codificação geralmente está atrelada aos *níveis* das variáveis do tipo `factor` que se tem no conjunto de dados.

Outro problema (ou restrição), é a inabilidade de se representar diretamente relacionamentos não lineares das variáveis preditoras (independentes).

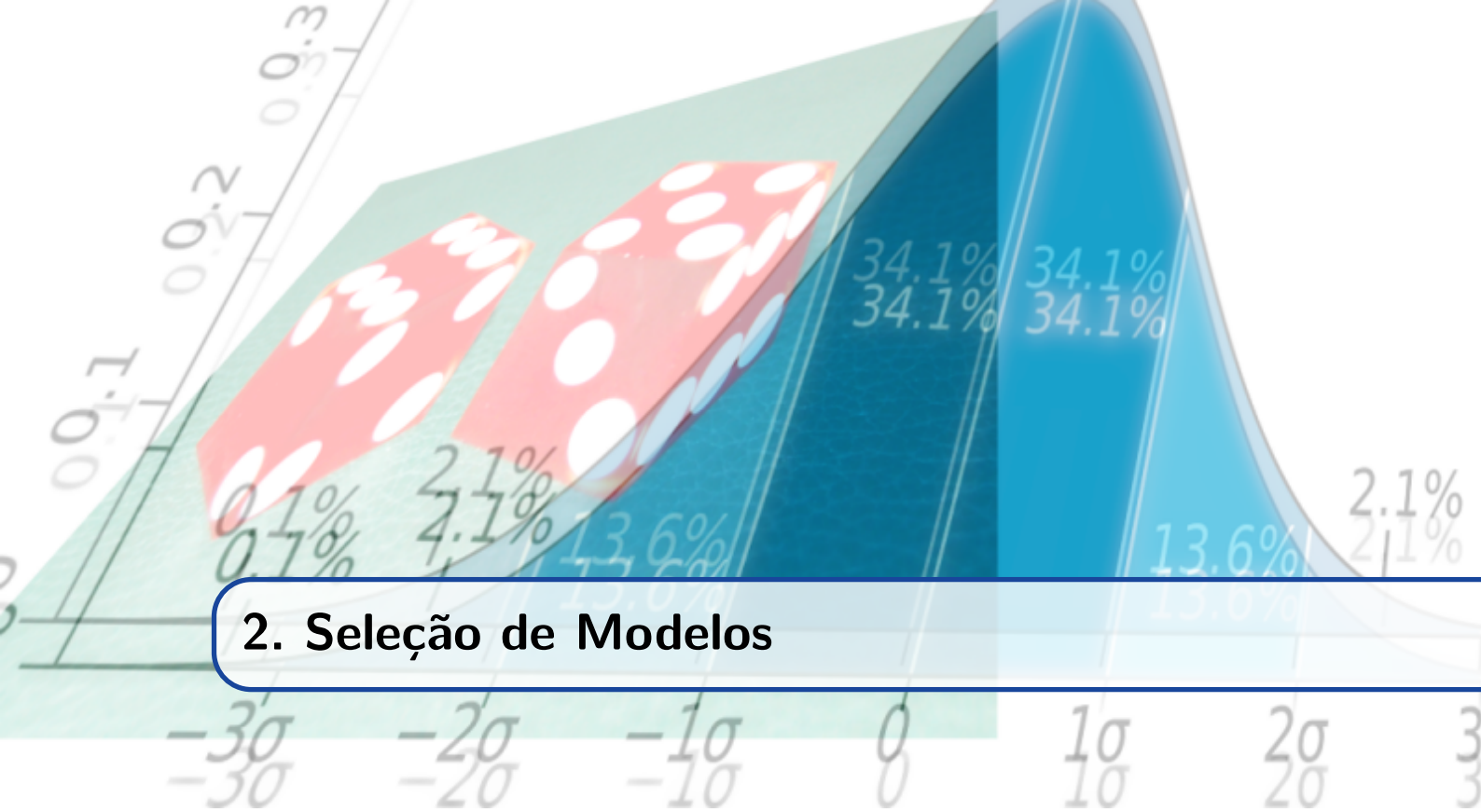
Uma alternativa para estas situações (relacionamentos não lineares) é a criação de novas variáveis através de transformações algébricas que eliminam os termos não lineares.

Outro uso para transformações de variáveis é para acertar violações de alguma das premissas (hipóteses) estatísticas.

Assim, temos duas razões básicas para transformarmos variáveis:

- Melhorar ou modificar o relacionamento entre as variáveis dependente e independentes (não linearidade ou violação de premissas estatísticas do método de mínimos quadrados).
- Habilitar o uso de variáveis não métricas na equação de regressão (*dummy coding*).

II Seleção de Modelos



2. Seleção de Modelos

Há muitas maneiras de se avaliar se um certo modelo é melhor do que outro. Nossa definição de *melhor* embarca o conceito de parcimônia no número de preditoras e também o que apresenta os melhores resíduos (i.e., o menor valor de SSE).

Critérios de avaliação de modelos

Algumas avaliações objetivas de modelos de regressão foram criadas, dentre elas, vamos estudar três possíveis avaliações numéricas baseadas nesta definição do **melhor**, elaboradas a seguir.

Definição 1. Akaike Information Criterion (AIC)

Um dos critérios mais comumente utilizados para a comparação de modelos é o AIC. A ideia do AIC é selecionar o modelo que minimiza o valor negativo da verosimilhança (*likelihood*) penalizado pelo número de parâmetros. O AIC também é conhecido como o **log-likelihood penalizado** (CRAWLEY (2013)). Para um modelo do qual podemos obter o *log-likelihood*, o AIC é dado pela Equação (1):

$$AIC(M) = -2 \times \log(-\text{likelihood}) + 2(p_M + 1) \tag{1}$$

onde p_M é o número de parâmetros no modelo, e o $\mathbf{1}$ é adicionado para uma variância estimada.

O termo $\log(-\text{likelihood})$, ou seja, o *log-likelihood* é dado pela Equação (2):

$$\log - \text{likelihood} = -\frac{n}{2}\log(2\pi) - n\log(\sigma) - \sum(y_i - \mu)^2/2\sigma^2 \quad (2)$$

Onde n é o número de observações, μ é o valor ajustado pelo modelo. O termo $\sum(y_i - \mu)^2$ é também denominado de SSE, como já vimos anteriormente. A variância σ^2 pode ser calculada como $\text{SSE}/(n-2)$ e $\sigma = \sqrt{(\sigma^2)}$.

O **AIC é o avaliador mais comumente utilizado**; ele é utilizado no comando `step()` do R.

Definição 2. Bayesian Information Criterion (BIC)

O BIC – Bayesian Information Criterion (CLAESKENS; HJORT (2008)), é definido como:

$$BIC(M) = -2 \times \log(-\text{likelihood}) + \log(n)(p_M + 1) \quad (3)$$

onde p_M é o número de preditoras no modelo.

O **BIC tem uma penalidade** $\log(n)$ **para o número de preditoras** em contraste com o valor 2 no AIC. Isto implica que $AIC(M) \geq BIC(M)$ para $\exp(x) \approx 7.3 \leq n$.

Do ponto de vista prático, BIC tem uma penalidade mais dura para um maior número de variáveis independentes do que o AIC e os procedimentos de seleção de modelos que utilizam BIC tem uma tendência de escolher modelos com menos variáveis independentes do que aqueles que usam o AIC.

Definição 3. Estatística C_p de Mallow

A estatítsica C_p de Mallow tem um comportamento similar ao do AIC para a seleção de variáveis na regressão, e é definida como (MALLOWS (1973)):

$$C_p(M) = \text{SSE}_M/\text{MSE}_{full} + 2(p_M + 1) - n \quad (4)$$

onde MSE_{full} é o MSE do modelo completo (i.e., o modelo com todas as X's), e p_M é o número de preditoras no modelo.

Conclusões

Para cada critério, um menor valor representa um modelo *melhor*. Com cada critério, podemos ordenar qual modelo é melhor que o outro. Se o número de preditoras é fixo, cada critério é simplesmente uma função de SSE

Exemplo de Seleção de Modelos

Muitas vezes temos um conjunto grande variáveis preditoras (explicativas) e nemm todas são significativas para um modelo estatístico dos dados. Vamos utilizar uma base de dados do pacote `faraway` – `prostate` que tem nove variáveis relacionadas a um estudo com 97 homens com câncer de próstata. A variável resposta é o Logaritmo do antígeno específico da próstata, denotada genericamente por Y, e as demais são candidatas a preditoras. Uma descrição mais detalhada deste conjunto de dados pode ser obtida no *help* do **R**: (`?prostate`).

```
> library(faraway)
> data("prostate")
```

```
> prostata <- prostate[,c('lpsa','lcavol','lweight',
...                         'age','lbph','svi','lcp','gleason','pgg45')]
> names(prostata) <- c("Y","X1","X2","X3","X4","X5","X6","X7","X8")
> head(prostata)
         Y         X1     X2 X3        X4 X5       X6 X7 X8
1 -0.43078 -0.5798185 2.7695 50 -1.386294  0 -1.38629  6  0
2 -0.16252 -0.9942523 3.3196 58 -1.386294  0 -1.38629  6  0
3 -0.16252 -0.5108256 2.6912 74 -1.386294  0 -1.38629  7 20
4 -0.16252 -1.2039728 3.2828 58 -1.386294  0 -1.38629  6  0
5  0.37156  0.7514161 3.4324 62 -1.386294  0 -1.38629  6  0
6  0.76547 -1.0498221 3.2288 50 -1.386294  0 -1.38629  6  0
> prostata[,6] = factor(prostata[,6])
```

A variável `svi`, que corresponde à coluna 6 no conjunto de dados `protata` é do tipo fator, mas está armazenada como um número; por isso, é necessário fazer uma transformação explícita, como mostrado acima.

Stepwise via AIC

Uma maneira de escolher entre as 2^8 regressões possíveis é utilizar os procedimentos *Backward* (Inclusão passo atrás), *Forward* (Inclusão passo a frente) ou uma combinação de ambos - o *Stepwise* (Seleção passo-a-passo). Comumente, estes procedimentos automáticos avaliam em cada passo os `p-valores` das preditoras em comparação a um α-critico. No entanto, no R, não há nenhuma função que considere o critério do `p-valor`.

A função `stepwise` considera em cada passo os critérios AIC (Akaike Information Criterion) ou BIC (Bayes Information Criterion). A seguir estão os comandos das funções na direção backward, forward ou em ambas, considerando o critério AIC. Note que é necessário definir os modelos nulo e cheio, pois são argumentos dessas funções.

Rodando a função no modo *backward*, isto é, indo do modelo completo para o modelo nulo.

```
> nulo <- lm(Y ~ 1, data=prostata)
> completo <- lm(Y ~ ., data=prostata)
> step(completo,data=prostata,direction="backward",trace=FALSE)

Call:
lm(formula = Y ~ X1 + X2 + X3 + X4 + X5, data = prostata)

Coefficients:
(Intercept)           X1           X2           X3           X4
    0.95100      0.56561      0.42369     -0.01489      0.11184
         X51
    0.72095
```

Para fins de comparação (e equivalência) de resultados, rodamos a função `step` com direção *forward* e também *both*.

```
> step(nulo,scope=list(lower=nulo, upper=completo),
...      data=prostata, direction="forward",trace=FALSE)

Call:
lm(formula = Y ~ X1 + X2 + X5 + X4 + X3, data = prostata)

Coefficients:
(Intercept)           X1           X2          X51           X4
    0.95100      0.56561      0.42369      0.72095      0.11184
         X3
```

```
   -0.01489
> step(completo,data=prostata,direction="both",trace=FALSE)

Call:
lm(formula = Y ~ X1 + X2 + X3 + X4 + X5, data = prostata)

Coefficients:
(Intercept)           X1           X2           X3           X4
    0.95100      0.56561      0.42369     -0.01489      0.11184
         X51
    0.72095
```

Se colocarmos o argumento `trace=TRUE`, os passos intermediários dos procedimentos serão exibidos, o que geraria uma saída imensa.

Podemos notar que os três procedimentos indicaram o modelo com as explicativas X_1, X_2, X_3, X_4 e X_5 como o melhor, segundo o critério AIC.

Vamos checar!

```
> summary(lm(Y ~ X1 + X2 + X3 + X4 + X5, data=prostata))

Call:
lm(formula = Y ~ X1 + X2 + X3 + X4 + X5, data = prostata)

Residuals:
     Min       1Q   Median       3Q      Max
-1.83505 -0.39396  0.00414  0.46336  1.57888

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.95100    0.83175   1.143 0.255882
X1           0.56561    0.07459   7.583 2.77e-11 ***
X2           0.42369    0.16687   2.539 0.012814 *
X3          -0.01489    0.01075  -1.385 0.169528
X4           0.11184    0.05805   1.927 0.057160 .
X51          0.72095    0.20902   3.449 0.000854 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7073 on 91 degrees of freedom
```

```
Multiple R-squared:  0.6441,    Adjusted R-squared:  0.6245
F-statistic: 32.94 on 5 and 91 DF,  p-value: < 2.2e-16
```

No entanto, observamos que a variável X3 não é significativa e deve ser retirada do modelo. Fazemos isso, e ajustamos novamente e verificamos o sumário. O que acontece?

```
> summary(lm(Y ~ X1 + X2 + X4 + X5, data=prostata))

Call:
lm(formula = Y ~ X1 + X2 + X4 + X5, data = prostata)

Residuals:
     Min       1Q   Median       3Q      Max
-1.82653 -0.42270  0.04362  0.47041  1.48530

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.14554    0.59747   0.244  0.80809
X1           0.54960    0.07406   7.422 5.64e-11 ***
X2           0.39088    0.16600   2.355  0.02067 *
X4           0.09009    0.05617   1.604  0.11213
X51          0.71174    0.20996   3.390  0.00103 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7108 on 92 degrees of freedom
Multiple R-squared:  0.6366,    Adjusted R-squared:  0.6208
F-statistic: 40.29 on 4 and 92 DF,  p-value: < 2.2e-16
```

```
> summary(lm(Y ~ X1 + X2 + X5, data=prostata))

Call:
lm(formula = Y ~ X1 + X2 + X5, data = prostata)

Residuals:
     Min       1Q   Median       3Q      Max
-1.72964 -0.45764  0.02812  0.46403  1.57013

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.26809    0.54350  -0.493  0.62298
```

```
X1           0.55164    0.07467   7.388  6.3e-11 ***
X2           0.50854    0.15017   3.386  0.00104 **
X51          0.66616    0.20978   3.176  0.00203 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7168 on 93 degrees of freedom
Multiple R-squared:  0.6264,    Adjusted R-squared:  0.6144
F-statistic: 51.99 on 3 and 93 DF,  p-value: < 2.2e-16
```

Vemos que os métodos automáticos nem sempre são os melhores na avaliação dos modelos. A visão de um especialista, tanto no domínio do problema, quanto em análise estatística sempre é essencial para a escolha do melhor modelo.

Melhores Subconjuntos (*Best subsets*)

Ao invés de listar todas as possíveis regressões com as 8 variáveis candidatas a preditoras, a função `regsubsets` lista os *k* melhores modelos, segundo o critério de menor soma de quadrados residual, para subgrupos de preditoras de todos os tamanhos (desde uma até oito variáveis explicativas).

Além de mostrar as melhores regressões, a função também retorna os valores da soma de quadrados residual `RSS` (denominado nesta função de s^2), da Estatística C_p de Mallows e de $R^2 - ajustado$ para cada regressão.

Essas medidas podem ser utilizadas como critérios de seleção destes modelos, pois deseja-se modelos com o menor número p de parâmetros, menor s^2, $R^2 - ajustado$ alto e Estatística C_p de Mallows com valor próximo de p.

A chamada da função `regsubsets` é:

```
subsets <- regsubsets(y ~ x1 + x2 ... + xk, dataframe)
```

onde:

- `subsets` é um objeto no qual os resultados estarão armazenados; a função não produz resultados, então temos que atribuir a saída a um objeto.

- `y ~ x1 + x2 ... + xk` é uma fórmula especificando o modelo a ser analisado.
- `dataframe` contém os dados (variáveis) do modelo.

As opções mais úteis da função são:

- `nbest = n` – especifica quantos modelos de cada tamanho devem ser mantidos no objeto resultante; o default é `nbest = 1`.
- `nvmax = n` – especifica o tamanho máximo do modelo (número de variáveis a serem incluídas); default é `nvmax = 8`
- `force.in = n [, n...]` – especifica um ou mais variáveis a ser incluída em todos os modelos, forçadamente. As variáveis são especificadas pela ordem na fórmula.

A função `regsubsets` não produz nenhum resultado direto. Mas podemos obter alguns dos resultados através das funções `summary` e `plot`. Destes, a função `plot` é bem interessante, apesar de o gráfico mostrado ser um tanto quanto *estranho* à primeira vista. No gráfico, cada linha representa um modelo; os retângulos preenchidos nas colunas indicam as variáveis incluídas no modelo. Os números na margem esquerda são os valores da estatística escolhida; o default é **BIC**, mas também é possível especificar o $R^2_{ajustado}$. O eixo não é quantitativo, mas é ordenado. Os tons dos preenchimentos simplesmente representa a ordenação dos valores da estatística especificada.

Para exemplificar, encontraremos a seguir as duas melhores regressões (k = 2) para cada tamanho. Veja que k é o argumento *nbest* desta função.

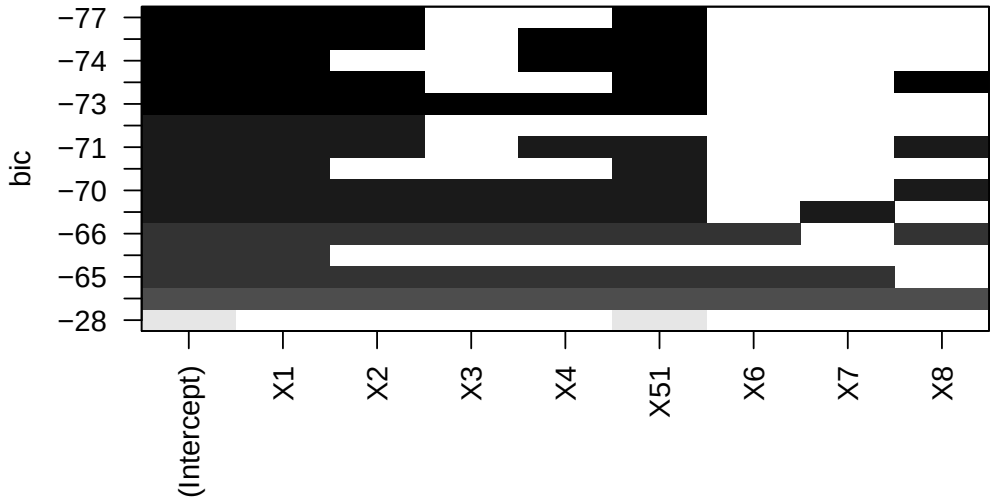
```
> require(leaps)
> rsbs <- regsubsets(Y ~., nbest=2,data=prostata)
```

A Figura 1 apresenta o gráfico sumário da função `regsubsets`.

```
> plot(rsbs)
> mtext("Fonte: Elaborado pelo autor", xpd = NA, cex = 0.7,
...       side = 1, line = 4.5, adj = -.2)
```

Analisando o gráfico mostrado na Figura 1, na primeira linha (parte de baixo), temos um modelo com Intercept e X5 com um BIC = -28. Já um modelo com Intercept, X1, X2, X3, X4 e X5 tem um BIC de -70. Neste gráfico, o melhor modelo está no topo, com BIC = -77, e inclui o Intercept e as variáveis X1, X2 e X5, que coincide com o resultado obtido com a função `step` após a eliminação das variáveis não significantes (X3 e X4).

Figura 1: Visualização do BIC para diversos modelos pela função 'regsubsets'



Fonte: Elaborado pelo autor

Podemos também visualizar o resultado através da função `summary`.

```
> (rs <- summary(rsbs))
Subset selection object
Call: regsubsets.formula(Y ~ ., nbest = 2, data = prostata)
8 Variables  (and intercept)
    Forced in Forced out
X1      FALSE      FALSE
X2      FALSE      FALSE
X3      FALSE      FALSE
X4      FALSE      FALSE
X51     FALSE      FALSE
X6      FALSE      FALSE
X7      FALSE      FALSE
X8      FALSE      FALSE
2 subsets of each size up to 8
Selection Algorithm: exhaustive
         X1  X2  X3  X4  X51 X6  X7  X8
1  ( 1 ) "*" " " " " " " " " " " " "
1  ( 2 ) " " " " " " " " " "*" " " " " "
2  ( 1 ) "*" "*" " " " " " " " " " " " "
2  ( 2 ) "*" " " " " " " " "*" " " " " "
3  ( 1 ) "*" "*" " " " " " "*" " " " " "
3  ( 2 ) "*" " " " " "*" "*" " " " " " "
4  ( 1 ) "*" "*" " " "*" "*" " " " " " "
4  ( 2 ) "*" "*" " " " " "*" " " " " "*"
```

```
5  ( 1 ) "*" "*" "*" "*" "*" " " " " " " " "
5  ( 2 ) "*" "*" " " "*" "*" " " " " " " "*"
6  ( 1 ) "*" "*" "*" "*" "*" " " " " " " "*"
6  ( 2 ) "*" "*" "*" "*" "*" " " "*" " " "
7  ( 1 ) "*" "*" "*" "*" "*" "*" " " " "*"
7  ( 2 ) "*" "*" "*" "*" "*" "*" "*" " " "
8  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*"
```

Abaixo, os valores específicos dos índices Cp e do BIC para cada valor de p (número de parâmetros).

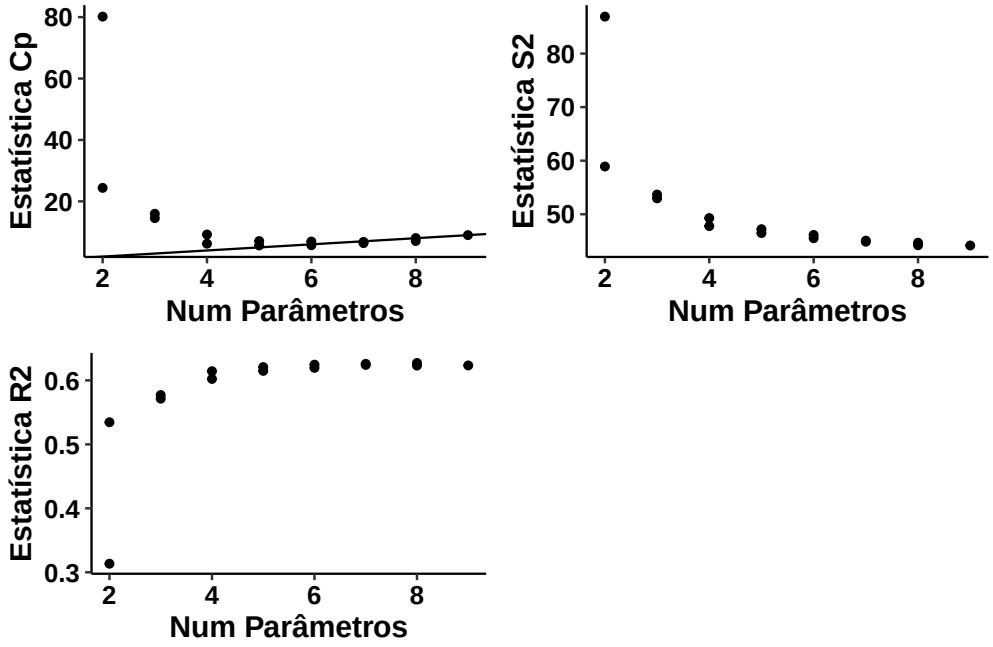
```
> # Mallow's Cp para cada p (número de parâmetros)
> summary(rsbs)$cp
 [1] 24.394559 80.172023 14.541475 15.958255  6.216935  9.208478  5.626422
 [8]  7.074224  5.715016  6.922392  6.401965  6.806372  7.082184  8.047624
[15]  9.000000
> # BIC para cada p (número de parâmetros)
> summary(rsbs)$bic
 [1] -66.05415 -28.34590 -71.80379 -70.51033 -77.21486 -74.21419 -75.31573
 [8] -73.81130 -72.76362 -71.48111 -69.60319 -69.16541 -66.47110 -65.41370
[15] -61.98694
```

Mais detalhes sobre o `summary(rsbs)` pode ser obtido na página de manual, digitando-se `?summary.regsubsets`

Uma outra maneira de se escolher o melhor modelo é analisando os gráficos mostrados a seguir:

```
> library(ggplot2)
> library(ggpubr)
> n_parametros = as.numeric(rownames(rs$which))+1
> Cp = rs$cp
> R2_ajustado = rs$adjr2
> s2 = rs$rss
> df.n <- data.frame(np = n_parametros, Cp = Cp)
> g1 <- ggplot(data = df.n, aes(x = np, y = Cp)) +
...   geom_point() + geom_abline(slope = 1, intercept = 0) +
...   theme_pubr() + labs_pubr() +
...   ylab("Estatística Cp") + xlab("Num Parâmetros")
> df.s2 <- data.frame(np = n_parametros, s2 = s2)
> g2 <- ggplot(df.n, aes(x = n_parametros, y = s2)) +
...   geom_point() + theme_pubr() + labs_pubr()+
```

Figura 2: Comportamento de Cp, R^2 e S^2 em função do número de parâmetros do modelo



Fonte: Elaborado pelo autor

```
...    ylab("Estatística S2") + xlab("Num Parâmetros")
> df.r2 <- data.frame(np = n_parametros, r2 = R2_ajustado)
> g3 <- ggplot(df.r2, aes(x = n_parametros, y = r2)) +
...    geom_point() + theme_pubr() + labs_pubr() +
...    ylab("Estatística R2") + xlab("Num Parâmetros") +
...    theme(plot.caption = element_text(hjust = 0, size = 9)) +
...    labs(caption = "Fonte: Elaborado pelo autor" )
> ggarrange(g1, g2, g3, ncol = 2, nrow = 2)
```

Analisando os gráficos acima, temos:

- Do gráfico do topo à esquerda, notamos que as Estatísticas C_p de Mallows já são próximas de p quando $p = 5$ ou $p = 6$.
- Dos outros dois gráficos, observa-se que o incremento em $R^2 - ajustado$ e o decremento em s^2 (RSS) são bem pequenos ao passar de $p = 5$ para $p = 6$.

Assim, tendo em vista um modelo mais parcimonioso, poderíamos escolher o valor de $p = 5$.

Podemos visualizar estes resultados na listagem a seguir, onde temos os seguintes valores:

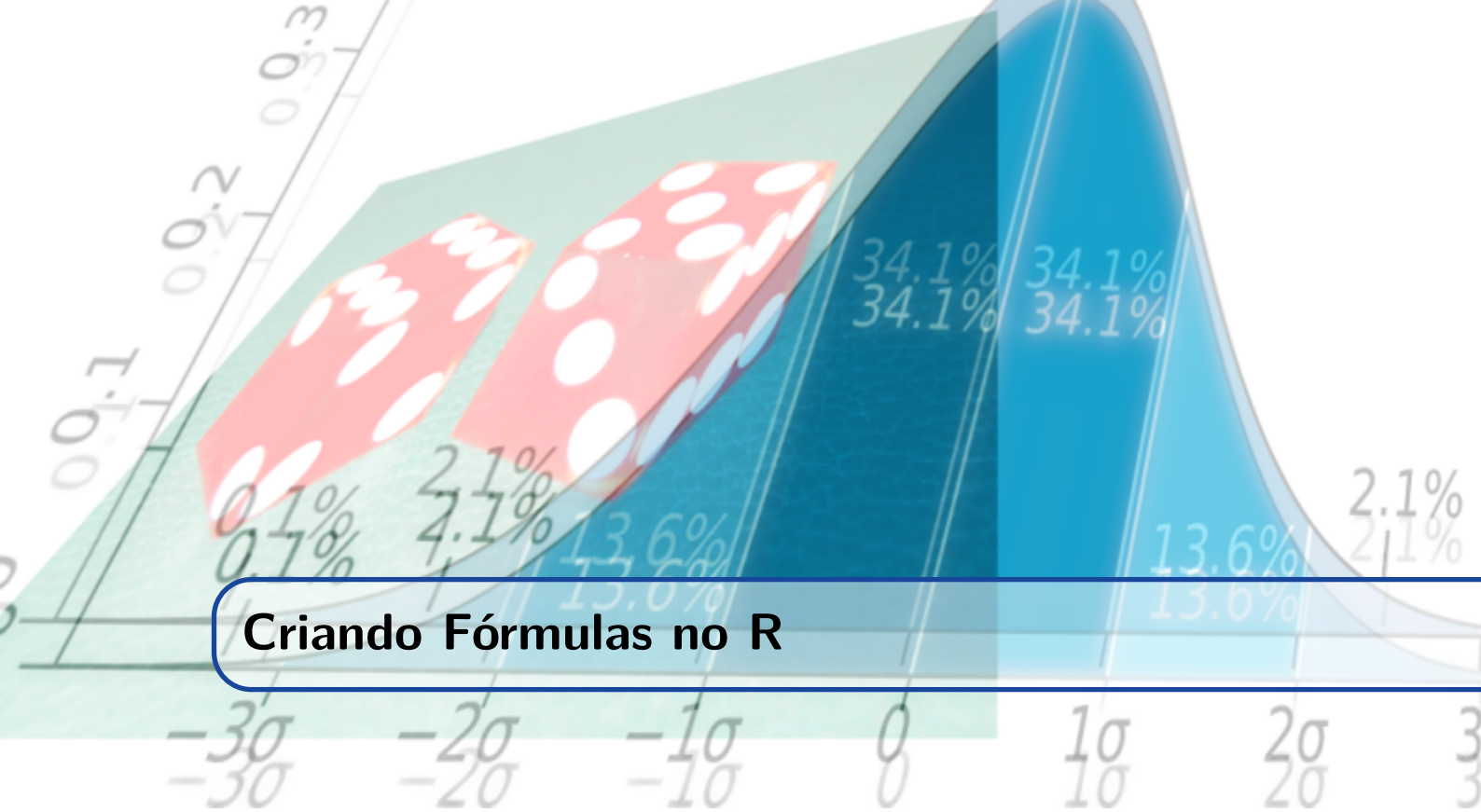
- Número de variáveis explicativas no modelo
- Número de parâmetros, contando o Intercept
- Estatística Cp
- R^2_{ajustado}
- Soma dos Quadrados dos Resíduos (RSS) – s2

```
> n_variaveis = n_parametros-1
> cbind(n_variaveis,n_parametros,Cp,R2_ajustado,s2)
      n_variaveis n_parametros        Cp R2_ajustado       s2
 [1,]           1            2 24.394559   0.5345838 58.91476
 [2,]           1            2 80.172023   0.3134515 86.90682
 [3,]           2            3 14.541475   0.5771246 52.96626
 [4,]           2            3 15.958255   0.5714480 53.67727
 [5,]           3            4  6.216935   0.6143899 47.78486
 [6,]           3            4  9.208478   0.6022748 49.28617
 [7,]           4            5  5.626422   0.6208036 46.48480
 [8,]           4            5  7.074224   0.6148766 47.21139
 [9,]           5            6  5.715016   0.6245476 45.52556
[10,]           5            6  6.922392   0.6195505 46.13149
[11,]           6            7  6.401965   0.6258707 44.86660
[12,]           6            7  6.806372   0.6241784 45.06956
[13,]           7            8  7.082184   0.6272521 44.20427
[14,]           7            8  8.047624   0.6231666 44.68878
[15,]           8            9  9.000000   0.6233681 44.16302
```

Dentre os dois modelos com 5 parâmetros, o primeiro, constituído pelas variáveis X1, X2, X4 e X5 seria o mais adequado segundo esta análise, mas é preciso ver saída completa do `summary(rsbs)`.

III

Apêndices



Criando Fórmulas no R

A estrutura de um modelo no R é especificada como uma fórmula do tipo:

variável resposta ~ variável(is) explicativa(s)

onde o símbolo ~ significa **é modelado como uma função de**

Assim, uma regressão linear simples de y em x seria escrita como:

$y \sim x$

E uma ANOVA com um fator, por exemplo, *sex* – um fator de dois níveis, seria escrito como:

$y \sim$ sex

O Quadro 5 apresenta exemplos de fórmulas.

Quadro 5: Exemplos de Fórmulas

Modelo	Fórmula do Modelo	Comentários
Nulo	$y \sim 1$	1 é o deslocamento em modelos de regressão, mas aqui é a média geral de y
Regressão	$y \sim x$	x é uma variável explicativa contínua
Regressão pela origem	$y \sim x - 1$	Não ajusta o deslocamento
ANOVA para um fator	$y \sim$ sex	sex é uma variável categórica de dois fatores
ANOVA para um fator	$y \sim$ sex $- 1$	como o acima, mas não ajusta o deslocamento (dá duas médias ao invés de uma média e uma diferença)
ANOVA para dois fatores	$y \sim$ sex $+$ genotipo	genotipo é uma variável categórica de 4 níveis
ANOVA Fatorial	$y \sim N * P * K$	N, P e K são fatores de 2 níveis a serem ajustados com suas interações
ANOVA para três fatores	$y \sim N * P * K - N : P : K$	Como o anterior, mas não ajusta as interações entre os 3 fatores
Análise de Covariância	$y \sim x +$ sex	Uma única inclinação para y contra x, com dois deslocamentos, um para cada sexo
Análise de Covariância	$y \sim x *$ sex	Duas inclinações e dois deslocamentos
ANOVA Aninhada	$y \sim a/b/c$	Fator c aninhado dentro do fator b dentro do fator a
Regressão Múltipla	$y \sim x + z$	Duas variáveis explicativas contínuas, ajuste de superfície plana
Regressão Múltipla	$y \sim x * z$	Ajusta um termo de interação também ($x + z + x : z$)
Regressão Múltipla	y $\sim x +$ `I(x^2)` $+$ z $+$ `I(z^2)`	Ajusta um termo quadrático para ambos x e z
Regressão Múltipla	$y \sim poly(x, 2) + z$	Ajusta um polinômio quadrático para x e linear em z
Regressão Múltipla	y $\sim$ `(x + z + w)^2`	Ajusta três variáveis mais suas interações até dois fatores
Modelo não paramétrico	$y \sim s(x) + s(z)$	y é uma função de x e z suavizados em um modelo generalizado aditivo
Variáveis resposta e explicativas transformadas	$log(y) \sim$ I(1/x) $+$ sqrt(z)	Todas as três variáveis são transformadas no modelo

Fonte: CRAWLEY (2013). Traduzido e adaptado pelo autor.

O lado direito das fórmulas de modelo mostram:

- o número de variáveis explicativas e suas identidades – seus atributos (p.explo, contínuas ou categóricas) são usualmente definidos antes de se ajustar o modelo.
  - Ou seja, você deve saber isso previamente.
- As interações entre as variáveis explicativas (se existirem);
- Termos não lineares nas variáveis explicativas.

Do lado direito do ~ (til), também podem ser especificados termos de erros ou deslocamentos (*offsets*) em alguns casos especiais. As variáveis explicativas também podem ser transformadas, ter potências ou polinômios. É muito importante notar que os símbolos são utilizados de modo diferente em uma fórmula de modelo em relação a expressões aritméticas. Em particular:

- + indica a inclusão de uma variável explicativa no modelo (não adição)
- − indica a remoção de uma variável explicativa do modelo (não subtração); importante quando usamos `update`
- $*$ indica a inclusão de variáveis explicativas e interações (não multiplicação)
- / indica o aninhamento de variáveis explicativas no modelo (não divisão)
- | indica o condicionamento (não ou lógico), de modo que $y \sim x|z$ é lido como y *como uma função de* x *dado* z

Alguns termos também podem ser escritos na forma expandida, como mostrado no Quadro 6.

Quadro 6: Exemplos de fórmulas em forma expandida

Termo	Significado
`A*B*C`	é o mesmo que A+B+C+A:B+A:C+B:C+A:B:C
A/B/C	é o mesmo que A+B%in%A+C%in%B%in%A
`(A+B+C)^3`	é o mesmo que A*B*C
`(A+B+C)^2`	é o mesmo que A*B*C − A:B:C

Fonte: CRAWLEY (2013). Traduzido e adaptado pelo autor.

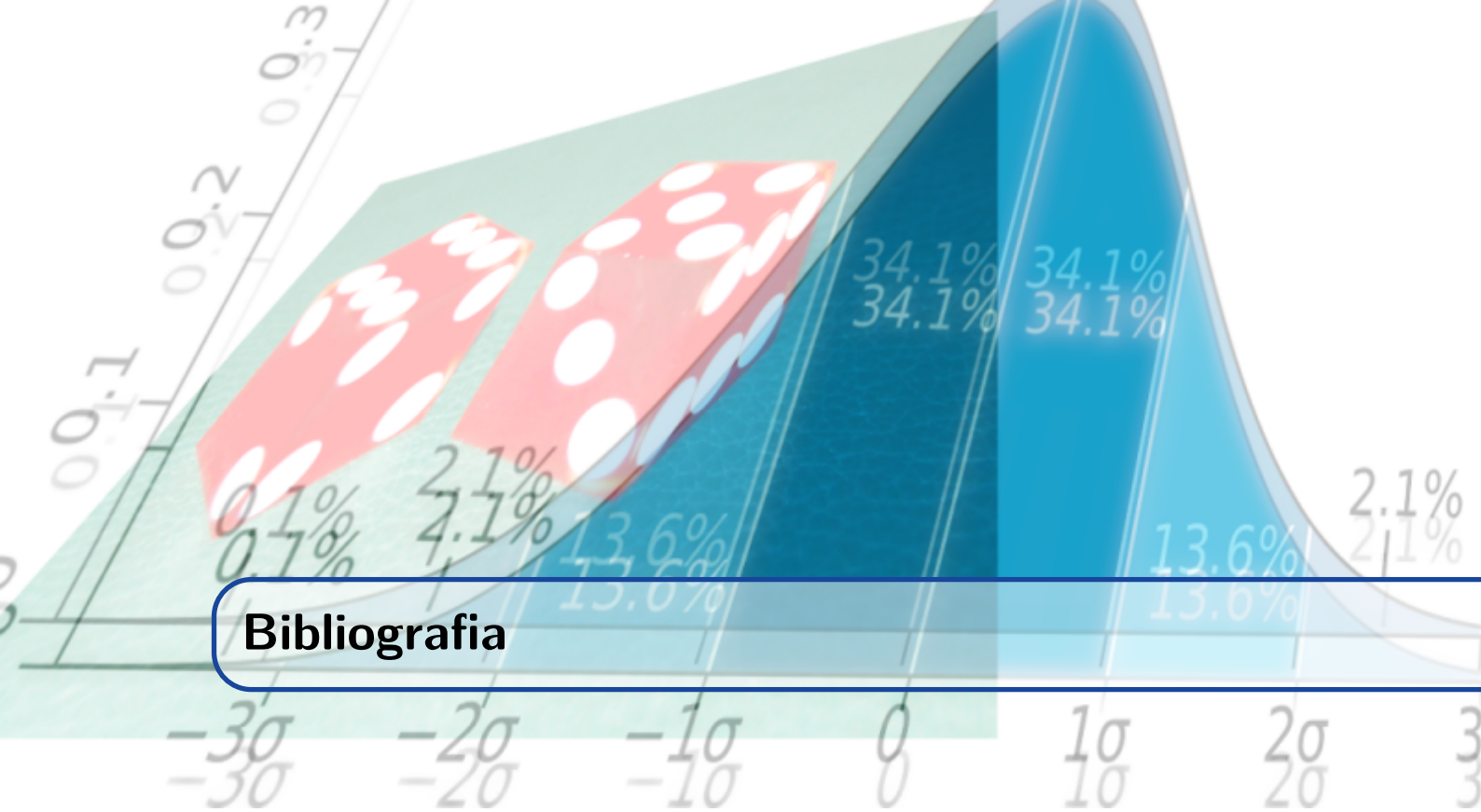
Funções úteis no ajuste de modelos lineares

O Quadro 7 apresenta algumas funções úteis quando trabalhamos com ajuste de modelos lineares.

Quadro 7: Funções úteis no ajuste de modelos lineares

Função	Ação
`summary()`	Mostra resultados detalhados para o modelo ajustado
`coefficients()`	Lista os parâmetros do modelo (deslocamento e inclinações) para o modelo ajustado
`confint()`	Provês os intervalos de confiança para os parâmetros do modelo (95% por padrão)
`fitted()`	Lista os valores preditos em um modelo ajustado
`residuals()`	Lista os valores dos resíduos em um modelo ajustado
`anova()`	Gera uma tabela de ANOVA para o modelo ajustado, ou uma tabela de ANOVA comparando dois ou mais modelos ajustados
`vcov()`	Lista a matriz de covariância para os parâmetros do modelo
`AIC()`	Imprime (mostra) o *Akaike's Information Criterion*
`plot()`	Gera os gráficos de diagnóstico para a avaliação do ajuste de um modelo
`predict()`	Usa o modelo ajustado para prever valores de resposta para um novo conjunto de dados

Fonte: KABACOFF (2015). Traduzido e adaptado pelo autor.



Bibliografia

CLAESKENS, G.; HJORT, N. L. **Model Selection and Model Averaging**. Cambridge, UK: Cambridge University Press, 2008.

CRAWLEY, M. J. **The R Book**. 2nd. ed. West Sussex, UK: Wiley & Sons, Ltd, 2013.

HAIR JR, J. F. et al. **Multivariate Data Analysis**. 7th. ed. Harlow, Essex, UK: Pearson Education Ltd, 2014.

JOHNSON, R. A.; WICHERN, D. W. **Applied Multivariate Statistical Analysis**. 3rd. ed. Englewood Cliffs, NJ, USA: Prentice Hall, 1992.

KABACOFF, R. I. **R in Action - Data Analysis and graphics with R**. 2nd. ed. Shelter Island, NY - USA: Manning Publications Co., 2015.

MALLOWS, C. L. Some Comments on C_p. **Technometrics**, v. 15, n. 4, p. 661–675, 1973.