

# Introdução ao Software R e à Análise Econométrica

Alexandre Xavier Ywata Carvalho  
Geraldo Sandoval Góes

# Seleção de Variáveis em Modelos de Regressão

# Seleção de Variáveis

- Considere agora o modelo geral de regressão:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_k x_{ki} + \epsilon_i$$

- Imagine que o nosso objetivo é de fazer previsões sobre a variável  $y_i$  com base em conjunto possível de variáveis preditoras
- Em geral, a inclusão adicional de variáveis preditoras na regressão, como já mencionamos aumenta o coeficiente de determinação ( $R^2$ )
- A inclusão adicional de variáveis preditoras também reduz (mesmo que marginalmente) o erro de previsão (*dentro da amostra*) mesmo que as variáveis preditoras não façam sentido
- Portanto, quanto mais incluimos variáveis explicativas na regressão, o  $R^2$  aumenta e a soma do quadrado dos erros diminui

$$SQE = \sum_{i=1}^n [y_i - \hat{y}_i]^2$$

# Seleção de Variáveis

- O problema é que a soma dos quadrados dos erros SQE corresponde aos erros da regressão dentro da amostra (*in-sample error*)
- Nós gostaríamos de ter um modelo de regressão que possa ter boas previsões para dados fora da amostra
  - Exemplo: queremos um modelo para avaliar a probabilidade de sucesso de novos cursos, com base em uma base de dados histórica de cursos anteriores, que fracassaram ou foram bem sucedidos
- Portanto, nós gostaríamos de ter um modelo que apresentasse baixo erro de previsão fora da amostra (*out-of-sample error*)
- Essa ideia de termos um bom modelo para previsão fora da amostra está intrinsicamente ligada aos procedimentos de validação cruzada (*cross-validation*) de um determinado modelo de regressão
  - A ideia da validação cruzada é dividir a amostra disponível em duas subamostras; por exemplo, uma delas com 80% das observações, e a outra com 20%.
  - Essa divisão tem que ser cuidadosa, para manter um certo balanço das informações em cada uma delas.

# Seleção de Variáveis

- Validação cruzada:
  - Dividimos a amostra em duas partes – podemos fazer uma divisão aleatória entre as observações que vão entrar em cada subamostra; a primeira amostra com  $n_1$  observações e a segunda com  $n_2$  observações
  - A primeira subamostra é usada para estimar os coeficientes da regressão ( $\beta_0, \beta_1, \beta_2, \dots, \beta_k$ )
  - Usamos os coeficientes estimados na primeira amostra para prever a variável reposta na segunda amostra
  - Calculamos agora o erro médio quadrático de previsão (*mean square prediction error*) com base apenas na segunda amostra

$$MSPE = \frac{1}{n_2} \sum_{i=1}^{n_2} [y_i - \hat{y}_i]^2$$

- Podemos então procurar o modelo de regressão, com as variáveis preditoras, que nos forneça o menor MSPE
  - O MSPE nos dá uma ideia do erro fora da amostra
- Um dos pacotes em R para previsão: “caret” (<http://topepo.github.io/caret/index.html>)

# Façam suas Apostas!

- Vamos agora testar o erro de previsão via *cross-validation* para três modelos
- Vamos dividir a amostra em 20% e 80% aleatoriamente, baleando-se por macrorregião; evitamos assim que uma região fique subrepresentada em uma das amostras
  - Amostra de treinamento: “dadosTrain”
  - Amostra de test: “dadosTest”
- Usaremos o pacote “caret” em R
- Usaremos 80% da amostra para estimação e 20% para testar os erros de previsão de cada modelo
- Três modelos serão avaliados – qual a sua aposta?
  - Vamos utilizar métodos estatísticos para encontrar o “melhor” modelo

# Façam suas Apostas!

- Vamos agora testar o erro de previsão via *cross-validation* para três modelos
- Vamos dividir a amostra em 20% e 80% aleatoriamente, baleando-se por macrorregião; evitamos assim que uma região fique subrepresentada em uma das amostras
  - Amostra de treinamento: “dadosTrain”
  - Amostra de test: “dadosTest”

```
set.seed(2104)
trainIndex <- createDataPartition(dados3$Regiao,
                                   p = .8, list = FALSE, times = 1) #-- balanceando entre regiões
```

```
head(trainIndex)
```

```
dadosTrain <- dados3[ trainIndex,] #--- amostra de treinamento
dadosTest  <- dados3[-trainIndex,] #--- amostra usada para testar a previsão
```

```
table(dadosTrain$Regiao)
table(dadosTest$Regiao)
```

# Façam suas Apostas!

- Modelo 1:

```
mod1 <- lm(mort_infantil ~ renda_per_capita
+ I(renda_per_capita^2)
+ I(renda_per_capita^3)
+ indice_gini
+ salario_medio_mensal
+ perc_crianças_extrem_pobres
+ perc_crianças_pobres
+ perc_pessoas_dom_agua_estogo_inadequados
+ perc_pessoas_dom_paredes_inadequadas
+ perc_pop_dom_com_coleta_lixo, data = dadosTrain)
summary(mod1)
```

# Façam suas Apostas!

- Modelo 2:

```
mod2 <- lm(mort_infantil ~ renda_per_capita
+ indice_gini
+ salario_medio_mensal
+ perc_crianças_extrem_pobres
+ perc_crianças_pobres
+ perc_pessoas_dom_agua_estogo_inadequados
+ perc_pessoas_dom_paredes_inadequadas
+ perc_pop_dom_com_coleta_lixo
+ perc_pop_rural
+ as.factor(Regiao)
+ as.factor(Regiao)*renda_per_capita, data = dadosTrain)
summary(mod2)
```

# Façam suas Apostas!

- Modelo 3:

```
mod3 <- lm(mort_infantil ~ renda_per_capita
+ indice_gini
+ salario_medio_mensal
+ perc_crianças_extrem_pobres
+ perc_crianças_pobres
+ perc_pessoas_dom_agua_estogo_inadequados
+ perc_pessoas_dom_paredes_inadequadas
+ perc_pop_dom_com_coleta_lixo
+ perc_pop_rural, data = dadosTrain)
summary(mod3)
```

# Façam suas Apostas!

- Comparando os três modelos:

```
mod1.pred <- predict(mod1, newdata = dadosTest, se.fit = T)
mod2.pred <- predict(mod2, newdata = dadosTest, se.fit = T)
mod3.pred <- predict(mod3, newdata = dadosTest, se.fit = T)
```

```
mod1.pred.error <- mod1.pred$fit - dadosTest$mort_infantil
mod2.pred.error <- mod2.pred$fit - dadosTest$mort_infantil
mod3.pred.error <- mod3.pred$fit - dadosTest$mort_infantil
```

```
mod1.mspe <- mean(mod1.pred.error^2)
mod2.mspe <- mean(mod2.pred.error^2)
mod3.mspe <- mean(mod3.pred.error^2)
```

```
mod1.mspe
mod2.mspe
mod3.mspe
```

# Seleção de Variáveis

- *K-fold cross-validation*:

Atualmente, uma regra de ouro para a seleção de modelos de previsão baseia-se na metodologia chamada *K-fold cross-validation*

1. Nesse caso, dividimos (em geral, aleatoriamente) a amostra total de dados em  $K$  subamostras; Em geral, usa-se  $K = 10$ ; podemos usar também  $K = 5$  ou  $K = 20$
2. Depois de dividir a amostra em  $K = 10$  partes, separamos a primeira dessas partes, e estimamos os coeficientes da regressão com base nas outras nove partes conjuntamente
3. Calculamos agora o erro médio quadrático de previsão (*mean square prediction error*) com base apenas na primeira das 10 partes

$$MSPE_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} [y_i - \hat{y}_i]^2$$

4. Repetimos os passos 2 e 3 mais nove vezes, cada uma das vezes deixando um 1/10 da amostra de fora estimações, e depois calculando o erro médio de previsão justamente na subamostra deixada de fora
5. Combinamos os erros médios quadráticos das 10 partes para chegarmos a uma medida agregada do erro de previsão fora da amostra

# Seleção de Variáveis

- Em geral, a inclusão indiscriminada de novas variáveis preditoras, apesar de reduzir o erro dentro da amostra, acaba aumentando o erro fora da amostra
- Por outro lado, a não inclusão de variáveis preditoras importantes pode também causar também uma perda de poder de previsão fora da amostra
- Portanto, todos os métodos de seleção automática de variáveis na literatura consideram essa relação de compromisso entre o aumento do poder explicatório da regressão versus a parcimônia na especificação
- Chamamos de ***trade-off viés-variância***
  - Quando incluimos variáveis, ***reduzimos o viés*** do modelo (é possível capturar mais especificidades da relação entre preditores e variável resposta)
  - No entanto, quando incluimos variáveis, temos que estimar mais parâmetros e a imprecisão (***variância***) de cada um deles aumenta
- Uma série de técnicas e indicadores foram criados para encontrarmos modelos para atender a essa relação de compromisso, sem necessariamente termos que recorrer à validação cruzada
  - Vamos estudar algumas dessas técnicas nos próximos slides

# Método de Máxima Verossimilhança

- Para um conjunto de parâmetros  $\beta_0, \beta_1, \beta_2, \dots, \beta_k$  para um modelo de regressão
- A função de verossimilhança, assumindo que os resíduos da regressão são normais, independentes e identicamente distribuídos, é escrita como

$$L(\beta_0, \beta_1, \dots, \beta_k, \sigma^2) = \prod_{i=1}^n \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} e^{\left[-\frac{1}{2\sigma^2}(y_i - \beta_0 - \beta_1 x_{1i} - \dots - \beta_k x_{ki})^2\right]}$$

- O termo  $\sigma^2$  corresponde à variância dos resíduos da regressão, que também precisa ser estimada
- O método de máxima verossimilhança é comumente empregado para estimar os parâmetros do modelo de regressão linear
- Pode-se mostrar que a fórmula para a estimativa dos coeficientes via máxima verossimilhança é idêntica à fórmula para estimativa via método de mínimos quadrados ordinários
- A diferença entre os dois métodos está na estimativa da variância  $\sigma^2$

# Método de Máxima Verossimilhança

- Pelo método de mínimos quadrados ordinário (MQO), a estimativa de  $\sigma^2$  é dada por

$$s^2 = \frac{SQE}{n - k - 1} = \frac{\sum_{i=1}^n [y_i - \hat{y}_i]^2}{n - k - 1}$$

- Pelo método de máxima verossimilhança, a estimativa de  $\sigma^2$  tem expressão

$$\hat{\sigma}^2 = \frac{SQE}{n} = \frac{\sum_{i=1}^n [y_i - \hat{y}_i]^2}{n}$$

- A estimativa via MQO para  $\sigma^2$  é não viesada; porém, o viés na estimativa via máxima verossimilhança desaparece quando o número de observações na amostra aumenta
- Por motivos numéricos e analíticos, trabalhamos com o log da função de verossimilhança, ao invés da função original

$$\log L(\beta_0, \beta_1, \dots, \beta_k, \sigma^2) = -\frac{n}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{1i} - \dots - \beta_k x_{ki})^2$$

# Método de Máxima Verossimilhança

- Por motivos numéricos e analíticos, trabalhamos com o log da função de verossimilhança, ao invés da função original

$$\log L(\beta_0, \beta_1, \dots, \beta_k, \sigma^2) = -\frac{n}{2} \log 2\pi\hat{\sigma}^2 - \frac{n}{2} = -\frac{n}{2} (1 + \log 2\pi\hat{\sigma}^2)$$

$$\text{onde } \hat{\sigma}^2 = \frac{SQE}{n} = \frac{\sum_{i=1}^n [y_i - \hat{y}_i]^2}{n}$$

- Portanto, quanto menor a SQE, maior a função de log verossimilhança
- O parâmetro  $\hat{\sigma}^2$  é calculado usando-se os erros dentro da amostra; portanto, quando incluímos mais variáveis (mesmo desnecessárias) no modelo, o erro diminui e a função de log verossimilhança aumenta
- Diversos indicadores surgiram com base em penalizações da função de log verossimilhança para a inclusão de mais variáveis na regressão
- Critérios comuns: AIC, BIC

# Seleção de Variáveis

- Critério de Informação de Akaike - AIC

$$AIC = -2 \log L(\beta_0, \beta_1, \dots, \beta_k, \sigma^2) + 2 \times p$$

O número  $p$  corresponde ao número de parâmetros livres na regressão. No caso da regressão linear, temos: um intercepto,  $k$  variáveis preditoras, a variância dos resíduos

$$p = 1 + k + 1 = 2 + k$$

No caso de regressão linear, o AIC é equivalente o critério  $C_p$  de Mallow

- Critério de Informação Bayesiano - BIC

$$BIC = -2 \log L(\beta_0, \beta_1, \dots, \beta_k, \sigma^2) + \log n \times p$$

- Os termos  $[2 \times p]$  e  $[\log n \times p]$ , no AIC e BIC, correspondem a pênaltis para a inclusão adicional de variáveis
- Portanto, a inclusão de variáveis vai aumentar  $\log L(\beta_0, \beta_1, \dots, \beta_k, \sigma^2)$ , mas aumenta também os pênaltis  $[2 \times p]$  e  $[\log n \times p]$

# Seleção de Variáveis

- Portanto, quando da escolha de um melhor modelo, selecionar aquele que resulte em menor BIC ou menor AIC
- O pênalti no BIC é mais pesado do que no AIC; consequência: o BIC em geral indica a escolha de modelos mais parcimoniosos
- Outros critérios existem, considerando-se outros pênaltis para o número de parâmetros livres  $p$ , entre eles:
  - Critério de informação de Hannan-Quinn
  - AIC corrigido
- No R:

AIC(mod1)

AIC(mod2)

AIC(mod3)

BIC(mod1)

BIC(mod2)

BIC(mod3)

# Seleção Automática de Variáveis

- Imagine agora queremos encontrar automaticamente um conjunto de variáveis que resulte em um melhor modelo para fins de previsão
- Diversas possibilidades existem na literatura, entre elas:
  - Seleção *best subset* \*
  - Seleção *stepwise*
  - Seleção *backwards*
  - Seleção *forward*
  - Regressão *ridge*
  - Lasso
- Todas elas buscam satisfazer a relação de compromisso entre erro dentro da amostra e parcimônia do modelo
- O R possui ferramentas para utilização dos métodos acima

# Seleção Automática de Variáveis

- Seleção *best subset*
  - Varre todas as combinações possíveis de variáveis preditoras para encontrar o conjunto com melhor  $R^2$  ajustado ou melhor critério  $C_p$  de Mallows, por exemplo
  - Computacionalmente, pode ser bastante demandante, e pode se tornar inviável quando temos muitas variáveis candidatas
  - Se tivermos  $M$  variáveis candidatas, há  $2^M$  conjuntos possíveis; por exemplo,  $M = 100$ , há  $1,268 \times 10^{30}$  regressões possíveis
- No R:
  - Pacote “leaps”
  - Para cada número de variáveis, encontra o modelo com menos SQE
  - Podemos analisar o valor do  $C_p$  (AIC), do BIC, do  $R^2$  ajustado e do  $R^2$  para cada número de variáveis
  - Para cada número de variáveis na regressão, o pacote “leaps” encontra o melhor conjunto de variáveis

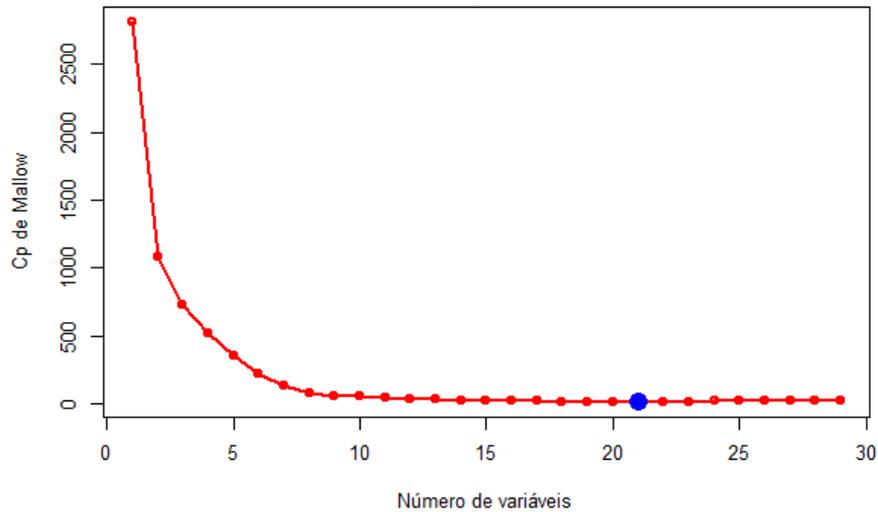
# Seleção Automática de Variáveis

- Modelo completo:

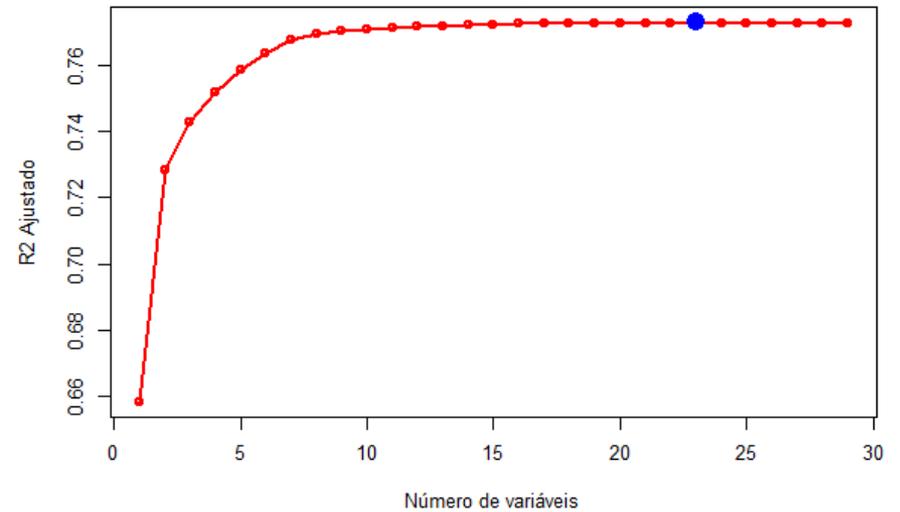
```
bestsub <- regsubsets(mort_infantil ~ renda_per_capita
+ l(renda_per_capita^2)
+ l(renda_per_capita^3)
+ l(renda_per_capita^4)
+ l(renda_per_capita^5)
+ indice_gini
+ l(indice_gini^2)
+ l(indice_gini^3)
+ l(indice_gini^4)
+ l(indice_gini^5)
+ salario_medio_mensal
+ l(salario_medio_mensal^2)
+ l(salario_medio_mensal^3)
+ l(salario_medio_mensal^4)
+ l(salario_medio_mensal^5)
+ perc_crianças_extrem_pobres
+ perc_crianças_pobres
+ perc_pessoas_dom_agua_estogo_inadequados
+ perc_pessoas_dom_paredes_inadequadas
+ perc_pop_dom_com_coleta_lixo
+ perc_pop_rural
+ as.factor(Regiao)
+ as.factor(Regiao)*renda_per_capita, data = dados3,
nvmax = 50)
```

# Seleção Automática de Variáveis

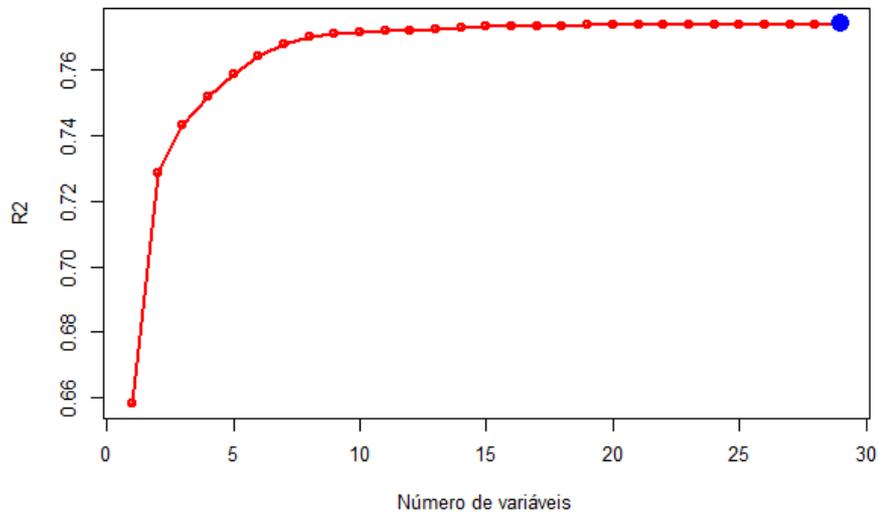
**Critério Cp de Mallow**



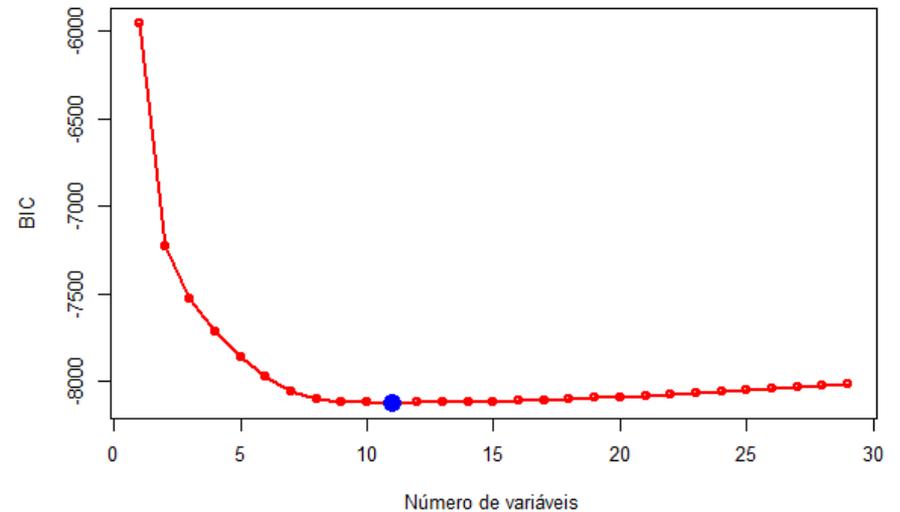
**Critério R2 Ajustado**



**Critério R2**



**Critério BIC**



# Seleção Automática de Variáveis

- Para um número grande (maior do que 50 ou 100) de potenciais preditores, a opção de *best subset* pode ser inviável computacionalmente
- Alternativas computacionalmente viáveis incluem:
  - Seleção *stepwise*
  - Seleção *backwards*
  - Seleção *forward*
- Seleção *forward*:
  1. Comece com uma regressão com apenas o intercepto
  2. Para as demais variáveis candidatas, escolha aquela cuja inclusão implica em maior aumento de  $R^2$
  3. Se essa nova adição foi estatisticamente significativa, mantenha a variável; caso contrário, retire a variável, volte ao modelo anterior, e pare o algoritmo
  4. Repita os passos 2 e 3 até que a adição de qualquer nova variável não seja estatisticamente significativa (a um nível de significância pré-especificado)

# Seleção Automática de Variáveis

- Seleção *backwards*:
  1. Comece com uma regressão com todas as variáveis candidatas
  2. Se houver alguma variável cujo coeficiente é estatisticamente não significativo, elimine a variável que tenha menor nível de significância no modelo (maior p-valor); caso contrário, esse é o modelo final
  3. Repita o passo 2 até atingir um modelo no qual todas as variáveis são estatisticamente significantes (a um nível de significância pré-definido)
- Seleção *Stepwise*:
  1. Trata-se de uma combinação das seleções do tipo *forward* e *backwards*
  2. Os passos *forward* e *backwards* são intercalados, de forma a adicionarmos variáveis que sejam significativas e retirarmos variáveis que não sejam estatisticamente significativas
  3. O algoritmo para quando não for mais possível adicionar variáveis novas que sejam estatisticamente significantes, ou retirar variáveis incluídas que forem estatisticamente não significantes
- Os passos acima dão uma ideia geral dos algoritmos; diferentes softwares possuem versões que são variações ao redor dessa ideia geral

```
#-----  
#--- Backwards, forward e stepwise selection  
#-----
```

```
mod.full <- lm(mort_infantil ~ renda_per_capita  
  + l(renda_per_capita^2)  
  + l(renda_per_capita^3)  
  + indice_gini  
  .....  
  + as.factor(Regiao)  
  + as.factor(Regiao)*renda_per_capita, data = dados3)  
summary(mod.full)
```

```
step1 <- step(mod.full, direction = "backward")  
summary(step1)
```

```
step2 <- step(mod.full, direction = "forward")  
summary(step2)
```

```
step3 <- step(mod.full, direction = "both")  
summary(step3)  
formula(step3)
```

```
mod.step3 <- lm(formula = formula(step3), data = dados3)  
summary(mod.step3)
```

- **Exercício 6 - para entregar em 2 semanas:**

- Utilize como base o código em R 'Análise\_de\_Regressao\_Linear\_Exercicios\_Praticos\_2'. Considere o modelo completo abaixo. Usando os diversos métodos aprendidos em sala de aula, encontre um modelo, subconjunto do modelo abaixo, que apresente o menor AIC. O grupo de alunos que obtiver o modelo com AIC de menor valor terá a nota deste exercício multiplicada por dois. No resultado entregue, você deverá incluir o código em R para obter o melhor modelo, e deverá incluir também a fórmula em R para essa “melhor” regressão

```
mod.full <- lm(mort_infantil ~ renda_per_capita
  + I(renda_per_capita^2)
  + I(renda_per_capita^3)
  + I(renda_per_capita^4)
  + I(renda_per_capita^5)
  + indice_gini
  + I(indice_gini^2)
  + I(indice_gini^3)
  + I(indice_gini^4)
  + I(indice_gini^5)
  + salario_medio_mensal
  + I(salario_medio_mensal^2)
  + I(salario_medio_mensal^3)
  + I(salario_medio_mensal^4)
  + I(salario_medio_mensal^5)
  + perc_crianças_extrem_pobres
  + perc_crianças_pobres
  + perc_pessoas_dom_agua_estogo_inadequados
  + perc_pessoas_dom_paredes_inadequadas
  + perc_pop_dom_com_coleta_lixo
  + perc_pop_rural
  + as.factor(Regiao)
  + as.factor(Regiao)*renda_per_capita, data = dados3)
```

Obrigado!