Multicolineariade

Guilherme Valle Moura e Denise Manfredini

20/05/2019

Hipóteses de MQO em Regressão Múltipla

No modelo de regressão múltipla, estendemos as três hipóteses de mínimos quadrados do modelo de regressão simples (ver Capítulo 4) e adicionamos uma quarta suposição. Estas hipóteses são apresentadas no Conceito Chave abaixo. Não entraremos nos detalhes das hipóteses 1-3, já que as vimos anteriormente e elas são facilmete generalizáveis para o caso de múltiplos regressores. Vamos nos concentrar na quarta suposição: hipótese que exclui a correlação perfeita entre os regressores.

Conceito Chave

As hipóteses de mínimos quadrados no modelo de regressão múltipla

O modelo de regressão múltipla é dado por:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_1 X_{2i} + \cdots + \beta_k X_{ki} + u_i \ \ i = 1, \ldots, n.$$

As hipóteses de MQO no modelo de regressão múltipla são uma extensão das feitas para o modelo de regressão simples:
1. Regressores $(X_{1i}, X_{2i}, \ldots, X_{ki}, Y_i)$ $i = 1, \ldots, n$, são amostrados de forma aleatória da mesma distribuição (i.e. são i.i.d.).
2. u_i é um termo de erro com média condicional zero dado os regressores, ou seja,

$$E(u_i|X_{1i}, X_{2i}, \ldots, X_{ki}) = 0.$$

3. Valores muito grandes e discrepantes são improváveis, formalmente $X_{1i}, \ldots, X_{ki}$ e Y_i possuem quarto momentos finitos.
4. Não há multicolinearidade perfeita.

Multicolineariadade

Multicolinearidade significa que dois ou mais regressores em um modelo de regressão múltipla são *fortemente* correlacionados. Se a correlação entre dois ou mais regressores é perfeita, isto é, um regressor pode ser escrito como uma combinação linear do (s) outro (s), temos *multicolinearidade perfeita.* Embora a multicolinearidade forte em geral seja ruim, pois faz com que a variância do estimador MQO seja grande (discutiremos isso com mais detalhes posteriormente), a presença de multicolinearidade perfeita torna impossível a solução para o estimador MQO, ou seja, o modelo não pode nem ser estimado.

Atenção: O fenômeno de multicolinearidade ocorre apenas durante regressões múltiplas.

A próxima seção apresenta alguns exemplos de multicolinearidade perfeita e demonstra como `lm ()` lida com eles.

Pontuação do Teste

Você tem dois conjuntos de variáveis explicativas e tem que escolher um desses conjuntos para analisar qual será a nota dos alunos. O primeiro conjunto tem as variáveis:

```
1.STR = Razão alunos-professor

2.english = alunos aprendendo inglês

3.FracEL = english/100, fração de alunos aprendendo inglês
```

O segundo conjunto também consiste de três variáveis:

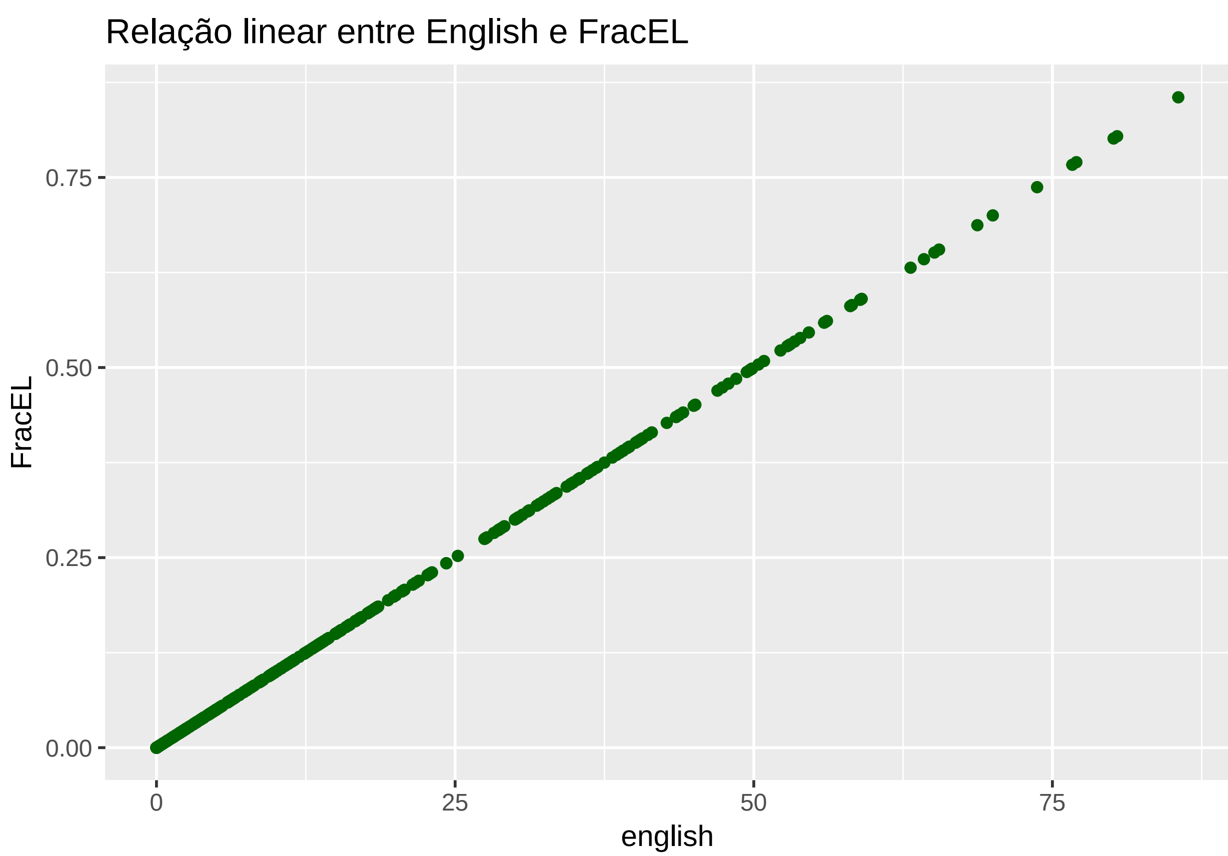
```
1.STR = Razão alunos-professor

2.english = alunos aprendendo inglês

3.income = Média da renda do distrito (USD 1000)
```

Qual dos dois conjuntos você acha que fornece mais informações sobre a nota dos alunos no teste?

O segundo conjunto fornece mais informações que o primeiro, pois as três variáveis são diferentes entre si e fornecem informações diferentes (estamos fazendo apenas uma análise intuitiva nesse momento). Além disso, nenhuma das variáveis no segundo conjunto é uma combinação **linear** de outra variável no sistema.

```r
# define a fração de alunos de inglês
CASchools$FracEL <- CASchools$english / 100
```

```r
ggplot(CASchools, aes(x = english, y = FracEL)) +
  geom_point(color = "darkgreen") +
  ggtitle("Relação linear entre English e FracEL")
```



Exemplos de multicolinearidade perfeita

Como o R reage se tentarmos estimar um modelo com regressores perfeitamente correlacionados?

lm produzirá um aviso na primeira linha do resultados da estimação, na seção dos coeficientes, dizendo 1 não definido devido a singularidades e ignora o(s) regressor(es) que é (são) combinação linear do(s) outro(s). Considere o exemplo a seguir, onde adicionamos à base de dados CASchools outra variável FracEL, a fração de alunos aprendendo inglês, cujas observações são valores das observações para english apenas em outra escala e usamos essa nova variável como um regressor juntamente com STR e english em um modelo de regressão múltipla. Neste exemplo, english e FracEL são perfeitamente colineares. O código R é o seguinte.

```r
# estima o modelo
mult.mod <- lm(score ~ STR + english + FracEL, data = CASchools)
# resume os resultados do modelo
summary(mult.mod)
```

```
##
## Call:
## lm(formula = score ~ STR + english + FracEL, data = CASchools)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.845 -10.240  -0.308   9.815  43.461
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 686.03224    7.41131  92.566  < 2e-16 ***
## STR          -1.10130    0.38028  -2.896  0.00398 **
## english      -0.64978    0.03934 -16.516  < 2e-16 ***
## FracEL             NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.46 on 417 degrees of freedom
## Multiple R-squared:  0.4264, Adjusted R-squared:  0.4237
## F-statistic:   155 on 2 and 417 DF,  p-value: < 2.2e-16
```

A linha FracEL na seção de coeficientes da saída tem valores NA, já que FracEL foi excluído do modelo.

Se fôssemos calcular as estimativas de MQO manualmente, nos depararíamos com o mesmo problema. As contas simplesmente não funcionam! Por que é isso? Veja o seguinte exemplo:

Suponha que você queira estimar um modelo de regressão linear simples com uma constante e um único regressor X. Como mencionado acima, para que a multicolinearidade perfeita esteja presente, X tem que ser uma combinação linear dos outros regressores. Como o único outro regressor é uma constante (pense no lado direito da equação do modelo como $\beta_0 \times 1 + \beta_1 X_i + u_i$ para que β_1 seja sempre multiplicado por 1 para cada observação), X tem que ser constante também. Por $\hat{\beta}_1$ temos

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2} = \frac{\widehat{Cov}(X,Y)}{\widehat{Var}(X)}. \tag{6.7}$$

A variância do regressor X está no denominador. Como a variância de uma constante é zero, não podemos calcular essa fração e $\hat{\beta}_1$ é indefinido.

Multicolinearidade Perfeita com Variável Binária

Vamos considerar outro exemplo em que nossa seleção de regressores induz a multicolinearidade perfeita. Primeiro, suponha que pretendemos analisar o efeito do tamanho da classe na pontuação do teste usando uma variável fictícia que identifica classes que não são pequenas (NS). Nós definimos que uma escola tem o atributo NS quando a média da relação aluno-professor é de pelo menos 12,

$$NS = \begin{cases} 0, & \text{se STR <12} \\ 1 & \text{caso contrário.} \end{cases}$$

Adicionamos a coluna correspondente ao objeto CASchools e estimamos um modelo de regressão múltipla com covariáveis computer e english.

```
# se STR menor 12, NS = 0, mais NS = 1
CASchools$NS <- ifelse(CASchools$STR < 12, 0, 1)
# estima o modelo
mult.mod <- lm(score ~ computer + english + NS, data = CASchools)
# Resultados
summary(mult.mod)
```

```
##
## Call:
## lm(formula = score ~ computer + english + NS, data = CASchools)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.492  -9.976  -0.778   8.761  43.798
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 663.704837   0.984259 674.319  < 2e-16 ***
## computer      0.005374   0.001670   3.218  0.00139 **
## english      -0.708947   0.040303 -17.591  < 2e-16 ***
## NS                  NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.43 on 417 degrees of freedom
## Multiple R-squared:  0.4291, Adjusted R-squared:  0.4263
## F-statistic: 156.7 on 2 and 417 DF,  p-value: < 2.2e-16
```

Novamente, a saída de summary(mult.mod) nos diz que a inclusão de NS na regressão tornaria a estimativa inviável. O que aconteceu aqui? Este é um exemplo em que cometemos um erro lógico ao definir o regressor NS: examinar NS, a medida redefinida para o tamanho da classe, revela que não há uma única escola com $STR < 12$, portanto, NS é igual a 1 para todas as observações. Podemos verificar isso imprimindo o conteúdo de CASchools$NS' ou usando a função table.

```
table(CASchools$NS)
```

```
##
##   1
## 420
```

CASchools$NS é um vetor com 420 valores iguais a 1 e nosso conjunto de dados inclui 420 observações. Isto obviamente viola a suposição 4 do Conceito Chave acima, uma vez que as observações para a constante já são sempre iguais a 1,

$$intercept = \lambda \cdot NS$$

$$\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \lambda \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$
$$\Leftrightarrow \lambda = 1.$$

Como os regressores podem ser escritos como uma combinação linear um do outro, nós nos deparamos com uma multicolinearidade perfeita e o R exclui NS do modelo. Logo, é importante pensar cuidadosamente sobre como os regressores em seus modelos se relacionam!

Multicolinearidade pode surgir por diversos fatores. Alguns desses fatores são: (i) inclusão ou uso incorreto de variáveis binárias (como NS); (ii) uso de variáveis derivadas de outras variáveis do sistema (como FracEL) e; (iii) uso de variáveis de natureza similar ou que fornecem informações similares.

Multicolinearidade Imperfeita

Ao contrário da multicolinearidade perfeita, a multicolinearidade imperfeita é - até certo ponto - menos problemática. Na verdade, a multicolinearidade imperfeita é a razão pela qual estamos interessados em estimar modelos de regressão múltipla: o estimador MQO nos permite *isolar* influências de regressores *correlacionados* na variável dependente. Se não fosse por essas dependências, não haveria uma razão para recorrer a uma abordagem de regressão múltipla e poderíamos simplesmente trabalhar com um modelo de regressão simples. No entanto, isso raramente acontece na prática. Já sabemos que ignorar as dependências entre os regressores que influenciam a variável de resultado gera viés de variáveis omitidas.

Então, quando e por que a multicolinearidade imperfeita é um problema? Suponha que você tenha o modelo de regressão

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + u_i \tag{6.9}$$

e você está interessado em estimar β_1, o efeito em Y_i de uma mudança de uma unidade em X_{1i}, enquanto mantém X_{2i} constante. Mas você não está certo de que o modelo verdadeiro realmente inclui X_2. Você segue algum raciocínio econômico e adiciona X_2 como uma covariável ao modelo para tratar um possível viés de variável omitida. Você está confiante de que $E(u_i|X_{1i}, X_{2i}) = 0$ e que não há razão para suspeitar de uma violação das premissas 2 e 3 feitas no Conceito Chave acima. Se X_1 e X_2 são altamente correlacionados, o método de MQO tenta estimar com precisão β_1. Isso significa que, embora $\hat{\beta}_1$ seja um estimador consistente e não viesado para β_1, ele tem uma grande variância devido à inclusão de X_2 no modelo. Se os erros forem homoscedásticos, esse problema poderá ser melhor compreendido a partir da fórmula da variação de $\hat{\beta}_1$ no modelo (6.9):

$$\sigma^2_{\hat{\beta}_1} = \frac{1}{n} \left(\frac{1}{1 - \rho^2_{X_1,X_2}} \right) \frac{\sigma^2_u}{\sigma^2_{X_1}}. \tag{6.10}$$

Primeiro, se $\rho_{X_1,X_2} = 0$, ou seja, se não houver correlação entre os dois regressores, incluir X_2 no modelo não terá influência na variância de $\hat{\beta}_1$. Em segundo lugar, se X_1 e X_2 estiverem correlacionados, $\sigma^2_{\hat{\beta}_1}$ é inversamente proporcional a $1 - \rho^2_{X_1,X_2}$, então quanto mais forte a correlação entre X_1 e X_2, menor é $1 - \rho^2_{X_1,X_2}$ e, portanto, maior é a variância de $\hat{\beta}_1$. Em terceiro lugar, aumentar o tamanho da amostra ajuda a reduzir a variação de $\hat{\beta}_1$. Naturalmente, isso não se limita ao caso de dois regressores: em regressões múltiplas, a multicolinearidade imperfeita inflaciona a variância de um ou mais estimadores de coeficientes. Quando o tamanho da amostra é pequeno, muitas vezes temos que tomar a decisão de aceitar a consequência de se adicionar um grande número de covariáveis (maior variância), ou de usar um modelo com apenas alguns regressores (possível viés de variável omitida). Isso é chamado de *trade-off de viés e variância*.

Esse *trade-off* geralmente ocorre.

Na medida que um modelo inclui mais regressores, os erros serão menores e as predições melhores, mas será mais difícil nterpretar os coeficientes. Por isso, se você está interessado em explicar a relaçao entre os regressores e o regredido, geralmente queremos um modelo que se ajuste bem, mas com um baixo número de regressores com pouca correlação.

Em resumo, consequências indesejáveis de multicolinearidade imperfeita geralmente não são o resultado de um erro lógico feito pelo pesquisador (como é frequentemente o caso da multicolinearidade perfeita), mas sim um problema que está ligado aos dados utilizados, o modelo a ser estimado e a questão de pesquisa em mãos.

Exercício de simulação: multicolinearidade imperfeita

Vamos realizar um exercício de simulação para ilustrar os problemas esboçados acima.

1. Usamos (6.9) como o processo de geração de dados e escolhemos $\beta_0 = 5$, $\beta_1 = 2.5$ e $\beta_2 = 3$ e u_i é um termo de erro distribuído como $\mathcal{N}(0, 5)$. Em uma primeira etapa, nós amostramos os dados do regressor a partir de uma distribuição normal bivariada:

$$X_i = (X_{1i}, X_{2i}) \overset{iid}{\sim} \mathcal{N}\left[\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}10 & 2.5\\2.5 & 10\end{pmatrix}\right]$$

   . É fácil ver que a correlação entre X_1 e X_2 na população é bastante baixa:

$$\rho_{X_1,X_2} = \frac{Cov(X_1, X_2)}{\sqrt{Var(X_1)}\sqrt{Var(X_2)}} = \frac{2.5}{10} = 0.25$$

2. Em seguida, estimamos o modelo (6.9) e salvamos as estimativas de β_1 e β_2. Isso é repetido 10000 vezes com um loop `for`, então acabamos com um grande número de estimativas que nos permitem descrever as distribuições de $\hat{\beta}_1$ e $\hat{\beta}_2$.
3. Repetimos os passos 1 e 2, mas aumentamos a covariância entre X_1 e X_2 de 2.5 para 8.5, de modo que a correlação entre os regressores é alta:

$$\rho_{X_1,X_2} = \frac{Cov(X_1, X_2)}{\sqrt{Var(X_1)}\sqrt{Var(X_2)}} = \frac{8.5}{10} = 0.85$$

4. Para avaliar o efeito sobre a precisão dos estimadores de aumentar a colinearidade entre X_1 e X_2, estimamos os desvios de $\hat{\beta}_1$ e $\hat{\beta}_2$ e compare.

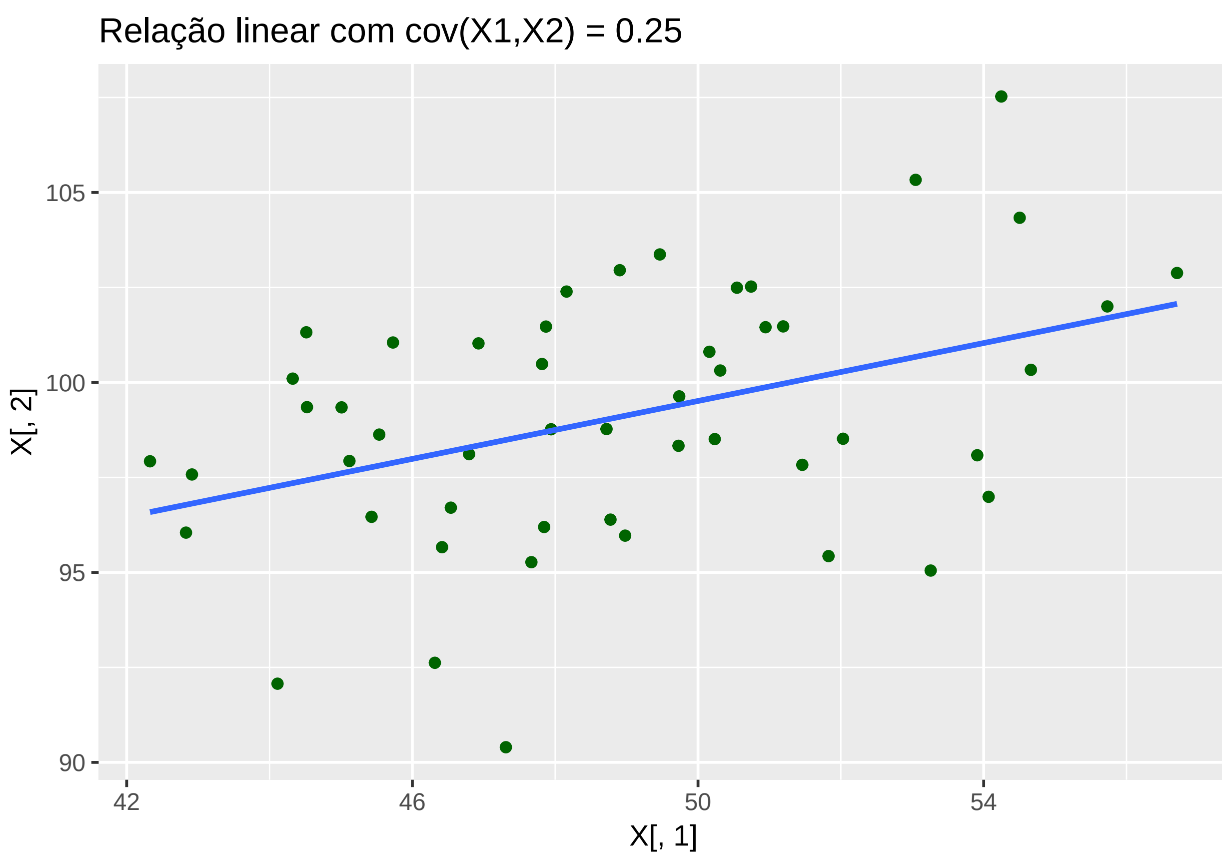
Colinearidade de 0.25

```
# Carrega pacotes
library(MASS)
library(mvtnorm)
# fixa número de observações
n <- 50
# inicializa vetor de coeficientes
coefs1 <- cbind("hat_beta_1" = numeric(10000), "hat_beta_2" = numeric(10000))
```

```r
coefs2 <- coefs1
# fixa semente
set.seed(1)
# loop para as diversas estimações
for (i in 1:10000) {

  # para o caso de cov(X_1,X_2) = 0.25
  X <- rmvnorm(n, c(50, 100), sigma = cbind(c(10, 2.5), c(2.5, 10)))
  u <- rnorm(n, sd = 5)
  Y <- 5 + 2.5 * X[, 1] + 3 * X[, 2] + u
  coefs1[i, ] <- lm(Y ~ X[, 1] + X[, 2])$coefficients[-1]
}
```

Gráfico da Relação Linear entre X_1 e X_2 – $cov(X_1, X_2) = 0.25$

```r
ggplot(data.frame(X), aes(x = X[,1], y = X[,2])) +
  geom_point(color = "darkgreen") +
  ggtitle("Relação linear com cov(X1,X2) = 0.25") +
  geom_smooth(method='lm', se = FALSE)
```

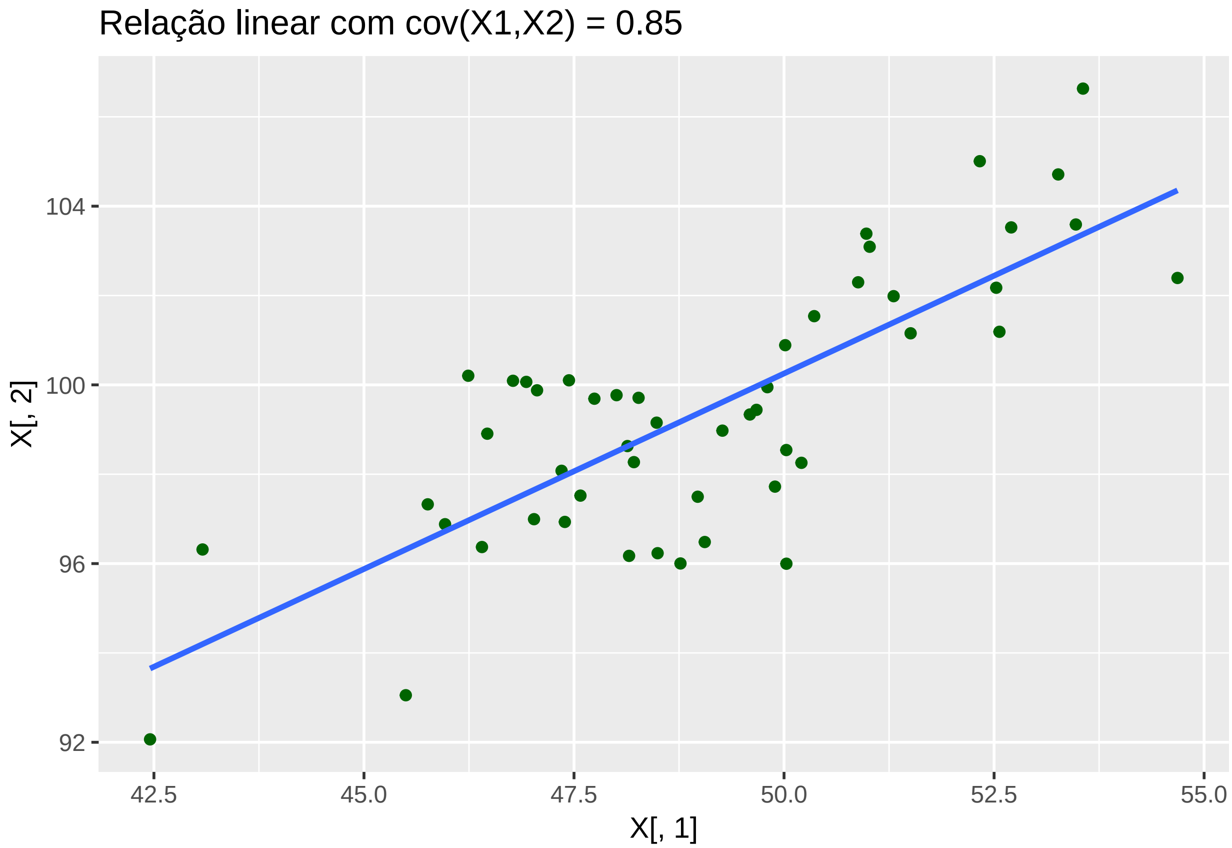


Colinearidade de 0.85

```
  # para o caso de cov(X_1,X_2) = 0.85
# loop para as diversas estimações
for (i in 1:10000) {
  X <- rmvnorm(n, c(50, 100), sigma = cbind(c(10, 8.5), c(8.5, 10)))
  Y <- 5 + 2.5 * X[, 1] + 3 * X[, 2] + u
  coefs2[i, ] <- lm(Y ~ X[, 1] + X[, 2])$coefficients[-1]

}
```

Gráfico da Relação Linear entre X_1 e X_2 – $cov(X_1, X_2) = 0.85$

```
ggplot(data.frame(X), aes(x = X[,1], y = X[,2])) +
  geom_point(color = "darkgreen") +
  ggtitle("Relação linear com cov(X1,X2) = 0.85") +
  geom_smooth(method='lm', se = FALSE)
```




```
# estimativa das variâncias
diag(var(coefs1))
```

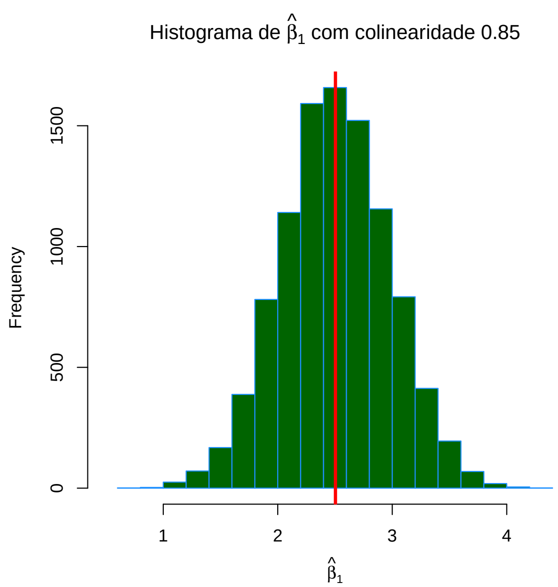
```
## hat_beta_1 hat_beta_2
## 0.05878630 0.05845242
```

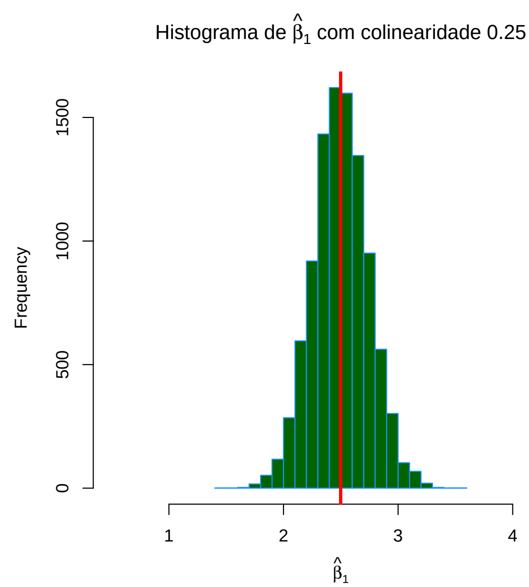
```r
diag(var(coefs2))
```

```
## hat_beta_1 hat_beta_2
##  0.2208951  0.2195864
```

Estamos interessados nas variâncias que são os elementos da diagonal. Vemos que, devido à alta colinearidade, as variâncias de $\hat{\beta}_1$ e $\hat{\beta}_2$ mais do que triplicaram, o que significa que é mais difícil estimar com precisão os coeficientes verdadeiros e testar hipóteses.

```r
par(mfrow = c(1, 2))
hist(coefs2[, 1],
     col = "darkgreen",
     border = "dodgerblue",
     main = expression("Histograma de " *hat(beta)[1]* " com colinearidade 0.85"),
     xlab = expression(hat(beta)[1]),
     breaks = 20,
     xlim=c(0.5, 4.5))
abline(v = mean(coefs2[, 1]), col = "red", lwd = 3)
hist(coefs1[, 1],
     col = "darkgreen",
     border = "dodgerblue",
     main = expression("Histograma de " *hat(beta)[1]* " com colinearidade 0.25"),
     xlab = expression(hat(beta)[1]),
     breaks = 20,
     xlim=c(0.5, 4.5))
abline(v = mean(coefs1[, 1]), col = "red", lwd = 3)
```





Médias de $\hat{\beta}_1$ com colineridade 0.25 e 0.85

```r
mean(coefs1[, 1])
```

```
## [1] 2.498793
```

```r
mean(coefs2[, 1])
```

```
## [1] 2.503915
```

Devios-Padrão de $\hat{\beta}_1$ com colineridade 0.25 e 0.85

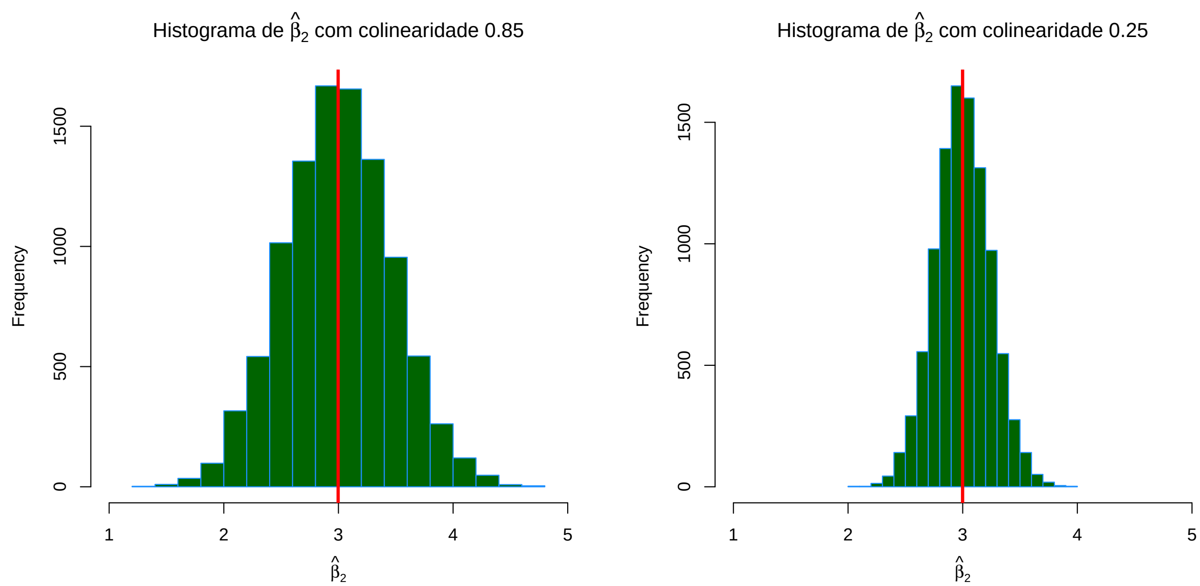
```r
sd(coefs1[, 1])
```

```
## [1] 0.2424589
```

```r
sd(coefs2[, 1])
```

```
## [1] 0.4699948
```

```r
par(mfrow = c(1, 2))
hist(coefs2[, 2],
     col = "darkgreen",
     border = "dodgerblue",
     main = expression("Histograma de " *hat(beta)[2]* " com colinearidade 0.85"),
     xlab = expression(hat(beta)[2]),
     breaks = 20,
     xlim=c(1, 5))
abline(v = mean(coefs2[, 2]), col = "red", lwd = 3)
hist(coefs1[, 2],
     col = "darkgreen",
     border = "dodgerblue",
     main = expression("Histograma de " *hat(beta)[2]* " com colinearidade 0.25"),
     xlab = expression(hat(beta)[2]),
     breaks = 20,
     xlim=c(1, 5))
abline(v = mean(coefs1[, 2]), col = "red", lwd = 3)
```



Médias de $\hat{\beta}_2$ com colineridade 0.25 e 0.85

```r
mean(coefs1[, 2])
```

```
## [1] 2.998939
```

```r
mean(coefs2[, 2])
```

```
## [1] 2.997355
```

Desvios-Padrão de $\hat{\beta}_2$ com colineridade 0.25 e 0.85

```r
sd(coefs1[, 2])
```

```
## [1] 0.2417693
```

```r
sd(coefs2[, 2])
```

```
## [1] 0.4686005
```

Na média, as estimativas estão corretas, mas a variação é novamente muito maior com alta colinearidade.

Outro Exercício de Multicolinearidade Imperfeita

Exemplo baseado em https://daviddalpiaz.github.io/appliedstats/collinearity.html

O conjunto de dados `seatpos` apresenta vários atributos dos motoristas, como altura, peso e idade. A nossa variável de interesse nessa base é hipcenter, que mede a "distância horizontal do ponto médio dos quadris a partir de um local fixo no carro em mm". Essencialmente, mede a posição do banco para um determinado motorista. Esta é uma informação potencialmente útil para os fabricantes de automóveis, considerando conforto e segurança ao projetar veículos.

Vamos tentar ajustar um modelo que prediz `hipcenter`. Dois regressores são imediatamente interessantes para o modelo: altura do pé com calçado em cm,`HtShoes` e altura descalço pé em cm, `Ht`.

```r
options(width = 100)
library(faraway)
```

```
## Warning: package 'faraway' was built under R version 3.4.4
```

```r
round(cor(seatpos), 3)
```

```
##              Age Weight HtShoes     Ht Seated    Arm  Thigh    Leg hipcenter
## Age        1.000  0.081  -0.079 -0.090 -0.170  0.360  0.091 -0.042     0.205
## Weight     0.081  1.000   0.828  0.829  0.776  0.698  0.573  0.784    -0.640
## HtShoes   -0.079  0.828   1.000  0.998  0.930  0.752  0.725  0.908    -0.797
## Ht        -0.090  0.829   0.998  1.000  0.928  0.752  0.735  0.910    -0.799
## Seated    -0.170  0.776   0.930  0.928  1.000  0.625  0.607  0.812    -0.731
## Arm        0.360  0.698   0.752  0.752  0.625  1.000  0.671  0.754    -0.585
## Thigh      0.091  0.573   0.725  0.735  0.607  0.671  1.000  0.650    -0.591
## Leg       -0.042  0.784   0.908  0.910  0.812  0.754  0.650  1.000    -0.787
## hipcenter  0.205 -0.640  -0.797 -0.799 -0.731 -0.585 -0.591 -0.787     1.000
```

Lembre-se de que a correlação mede a força e a direção da relação **linear** entre as variáveis. A correlação entre `Ht` e `HtShoes` é extremamente alta, 0.998.

Como a multicolinearidade entre essas variáveis não é perfeita, podemos estimar um modelo de MQO com as duas.

Que efeitos essa alta colinearidade gera?

```r
hip_model = lm(hipcenter ~ ., data = seatpos)
summary(hip_model)
```

```
##
## Call:
## lm(formula = hipcenter ~ ., data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -73.827 -22.833  -3.678  25.017  62.337
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 436.43213  166.57162   2.620   0.0138 *
## Age           0.77572    0.57033   1.360   0.1843
## Weight        0.02631    0.33097   0.080   0.9372
## HtShoes      -2.69241    9.75304  -0.276   0.7845
## Ht            0.60134   10.12987   0.059   0.9531
## Seated        0.53375    3.76189   0.142   0.8882
## Arm          -1.32807    3.90020  -0.341   0.7359
## Thigh        -1.14312    2.66002  -0.430   0.6706
## Leg          -6.43905    4.71386  -1.366   0.1824
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.72 on 29 degrees of freedom
## Multiple R-squared:  0.6866, Adjusted R-squared:  0.6001
## F-statistic:  7.94 on 8 and 29 DF,  p-value: 1.306e-05
```

Uma das primeiras coisas que devemos notar é que o teste F para a regressão nos diz que a regressão é significativa, no entanto, cada preditor individual não é. Outro resultado interessante são os sinais opostos dos coeficientes para `Ht` e`HtShoes`. Isso deve parecer bastante contra-intuitivo. Aumentar `Ht` aumenta`hipcenter`, mas aumentar `HtShoes` diminui`hipcenter`?

Isso acontece como resultado de os preditores estarem altamente correlacionados. Por exemplo, a variável `HtShoe` explica uma grande quantidade da variação em`Ht`. Quando ambos estão no modelo, seus efeitos na resposta são pequenos individualmente, mas juntos eles ainda explicam uma grande parte da variação do `hipcenter`.

Vamos agora olhar para um modelo menor:

```r
hip_model_small = lm(hipcenter ~ Age + Arm + Ht, data = seatpos)
summary(hip_model_small)
```

```
##
## Call:
## lm(formula = hipcenter ~ Age + Arm + Ht, data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -82.347 -24.745  -0.094  23.555  58.314
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 493.2491   101.0724   4.880 2.46e-05 ***
## Age           0.7988     0.5111   1.563  0.12735
## Arm          -2.9385     3.5210  -0.835  0.40979
## Ht           -3.4991     0.9954  -3.515  0.00127 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.12 on 34 degrees of freedom
## Multiple R-squared:  0.6631, Adjusted R-squared:  0.6333
## F-statistic:  22.3 on 3 and 34 DF,  p-value: 3.649e-08
```

```r
vif(hip_model_small)
```

```
##      Age      Arm       Ht
## 1.749943 3.996766 3.508693
```