

# Modelos de Regressão e Previsão

## MRLM: verificando as hipóteses II

Prof. Carlos Trucíos  
carlos.trucios@facc.ufrj.br  
ctruciosm.github.io

Faculdade de Administração e Ciências Contábeis  
Universidade Federal do Rio de Janeiro

## Aula 11

# Má-especificação funcional

# Má-especificação funcional

Suponha que o modelo populacional é da forma

$$\log(wage) = \beta_0 + \beta_1 educ + \beta_2 exper + \beta_3 exper^2 + u$$

Mas na modelagem utilizamos

$$\log(wage) = \beta_0 + \beta_1 educ + \beta_2 exper + u$$

Ou seja, omitimos  $exper^2$

# Má-especificação funcional

Suponha que o modelo populacional é da forma

$$\log(wage) = \beta_0 + \beta_1 educ + \beta_2 exper + \beta_3 exper^2 + \beta_4 female + \beta_5 female * educ + u$$

Mas na modelagem utilizamos

$$\log(wage) = \beta_0 + \beta_1 educ + \beta_2 exper + \beta_3 exper^2 + \beta_4 female + u$$

Ou seja, omitimos *female \* educ*

# Má-especificação funcional

Suponha que o modelo populacional é da forma

$$\log(wage) = \beta_0 + \beta_1 educ + \beta_2 exper + \beta_3 exper^2 + \beta_4 female + \beta_5 female * educ + u$$

Mas na modelagem utilizarmos

$$wage = \beta_0 + \beta_1 educ + \beta_2 exper + \beta_3 exper^2 + \beta_4 female + \beta_5 female * educ + u$$

Ou seja, modelamos *wage* no lugar de  $\log(wage)$

# Má-especificação funcional

- Em todos os casos estamos enfrentando um problema de má-especificando a forma funcional

# Má-especificação funcional

- Em todos os casos estamos enfrentando um problema de má-especificando a forma funcional
- Os  $\hat{\beta}'$ s tendem a ser viesados

# Má-especificação funcional

- Em todos os casos estamos enfrentando um problema de má-especificando a forma funcional
- Os  $\hat{\beta}'$ s tendem a ser viesados
- Estudaremos algumas formas de detectar e corrigir este problema



# Má-especificação funcional

- Em todos os casos estamos enfrentando um problema de má-especificando a forma funcional
- Os  $\hat{\beta}'$ s tendem a ser viesados
- Estudaremos algumas formas de detectar e corrigir este problema

# Má-especificação funcional

- Em todos os casos estamos enfrentando um problema de má-especificando a forma funcional
- Os  $\hat{\beta}'$ s tendem a ser viesados
- Estudaremos algumas formas de detectar e corrigir este problema

## Com detectar?

- 1 Incluir termos quadráticos (das variáveis que foram encontradas como estatisticamente significativas) na modelagem
- 2 Fazer um *teste F* onde  $H_0 : \beta_{x_1^2} = 0, \dots, \beta_{x_p^2} = 0$
- 3 Se rejeitarmos  $H_0$  concluímos que pelo menos uma relação quadrática existe

# Má-especificação funcional

```
library(wooldridge)
modelo = lm(narr86~pcnv + avgsen + ptime86 + qemp86
            + inc86 + black + hispan, data = crime1)
round(summary(modelo)$coef,5)
```

##	Estimate	Std. Error	t value	Pr(> t )
## (Intercept)	0.56927	0.03605	15.79226	0.00000
## pcnv	-0.13283	0.04035	-3.29176	0.00101
## avgsen	0.00309	0.00469	0.65835	0.51037
## ptime86	-0.03901	0.00869	-4.48626	0.00001
## qemp86	-0.05097	0.01444	-3.53065	0.00042
## inc86	-0.00148	0.00034	-4.35252	0.00001
## black	0.32663	0.04542	7.19121	0.00000
## hispan	0.19478	0.03971	4.90512	0.00000

# Má-especificação funcional

Variáveis estatisticamente significativas:

- `pcnv`
- `ptime86`
- `qmp86` (qualitativa)
- `inc86`
- `black` (qualitativa)
- `hispan` (qualitativa)

Então vamos incluir no modelo:

- $pcnv^2$
- $ptime86^2$
- $inc86^2$

*Por quê não incluímos as outras variáveis significativas?*

# Má-especificação funcional

##	Estimate	Std. Error	t value	Pr(> t )
## (Intercept)	0.50499	0.03684	13.70820	0.00000
## pcnv	0.55632	0.15423	3.60716	0.00032
## avgsen	-0.00270	0.00466	-0.58053	0.56160
## ptime86	0.28874	0.04425	6.52492	0.00000
## qemp86	-0.01457	0.01736	-0.83935	0.40135
## inc86	-0.00340	0.00080	-4.23553	0.00002
## black	0.29239	0.04484	6.52149	0.00000
## hispan	0.16436	0.03945	4.16628	0.00003
## I(pcnv^2)	-0.73376	0.15611	-4.70020	0.00000
## I(ptime86^2)	-0.02956	0.00386	-7.65156	0.00000
## I(inc86^2)	0.00001	0.00000	2.80504	0.00507

# Má-especificação funcional

$H_0 : \beta_{pcnv^2} = 0, \beta_{ptime86^2} = 0, \beta_{inc86^2} = 0$  vs  $H_1 : H_0$  não é verdade

```
modeloi = lm(narr86~pcnv + avgsen + ptime86 + qemp86
+ inc86 + black + hispan + I(pcnv^2) + I(ptime86^2) + I(inc86^2)
modelor = lm(narr86~pcnv + avgsen + ptime86 + qemp86
+ inc86 + black + hispan, data = crime1)
anova(modelor,modeloi)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Model 1: narr86 ~ pcnv + avgsen + ptime86 + qemp86 + inc86 + black + hispan
```

```
## Model 2: narr86 ~ pcnv + avgsen + ptime86 + qemp86 + inc86 + black + hispan
```

```
##      I(pcnv^2) + I(ptime86^2) + I(inc86^2)
```

```
##   Res.Df    RSS Df Sum of Sq    F    Pr(>F)
```

```
## 1     2717 1866.1
```

```
## 2     2714 1803.5  3     62.602 31.403 < 2.2e-16 ***
```

# Má-especificação funcional

- Os termos quadráticos são (individualmente e em conjunto) estatisticamente significativos
- Parece que o modelo inicial deixou de fora algumas não linearidades no modelo, que foram capturadas quando incluímos os quadrados
- Incluímos apenas quadrados das variáveis mas outros tipos de não linearidades não foram considerados
- Existem testes que nos ajudam nesse sentido

# Má-especificação funcional: Teste RESET

Se o modelo

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + u$$

satisfazer HRML4 ( $E(u|X) = 0$ ) nenhuma função linear de das  $x$ 's deve ser significativa quando incluídas na regressão.



# Má-especificação funcional: Teste RESET

Se o modelo

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + u$$

satisfazer HRML4 ( $E(u|X) = 0$ ) nenhuma função linear de das  $x$ 's deve ser significativa quando incluídas na regressão.

Então se ajustarmos o modelo (note que  $\hat{y} = X'\hat{\beta}$ )

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + \delta_1 \hat{y}^2 + \delta_2 \hat{y}^3 + u \quad (1)$$

$\delta_1$  e  $\delta_2$  deveriam ser 0

# Má-especificação funcional: Teste RESET

Se o modelo

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + u$$

satisfazer HRML4 ( $E(u|X) = 0$ ) nenhuma função linear de das  $x$ 's deve ser significativa quando incluídas na regressão.

Então se ajustarmos o modelo (note que  $\hat{y} = X'\hat{\beta}$ )

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + \delta_1 \hat{y}^2 + \delta_2 \hat{y}^3 + u \quad (1)$$

$\delta_1$  e  $\delta_2$  deveriam ser 0

O Teste RESET, utiliza a estatística F para testar

$$H_0 : \delta_1 = 0, \delta_2 = 0 \quad \text{vs} \quad H_1 : H_0 \text{ não é verdade}$$

# Má-especificação funcional: Teste RESET

```
modelo = lm(price~lotsize+sqrft+bdrms, data = hprice1)
yhat = fitted(modelo)
modelofull = lm(price~lotsize+sqrft+bdrms +
                I(yhat^2)+I(yhat^3), data = hprice1)
anova(modelo,modelofull)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Model 1: price ~ lotsize + sqrft + bdrms
```

```
## Model 2: price ~ lotsize + sqrft + bdrms + I(yhat^2) + I(yhat^3)
```

```
##   Res.Df    RSS Df Sum of Sq    F  Pr(>F)
```

```
## 1      84 300724
```

```
## 2      82 269984  2    30740 4.6682 0.01202 *
```

```
## ---
```

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

# Má-especificação funcional: Teste RESET

- No slide anterior rejeitamos  $H_0$  com um nível de significância de 0.05
- Precisamos testar outras formas funcionais
- Na prática é difícil descobrir a forma funcional exata
- Uma das primeiras coisas que devemos fazer é testar o  $\log(\cdot)$
- $\log(\cdot)$  e funções quadráticas costumam resolver o problema

# Má-especificação funcional: Teste RESET

```
modelo = lm(log(price)~log(lotsize)+log(sqrft)+bdrms, data = hpr
yhat = fitted(modelo)
modelofull = lm(log(price)~log(lotsize)+log(sqrft)+bdrms +
                I(yhat^2)+I(yhat^3), data = hprice1)
anova(modelo,modelofull)
```

```
## Analysis of Variance Table
```

```
##
## Model 1: log(price) ~ log(lotsize) + log(sqrft) + bdrms
## Model 2: log(price) ~ log(lotsize) + log(sqrft) + bdrms + I(y
##      I(yhat^3)
##   Res.Df    RSS Df Sum of Sq    F  Pr(>F)
## 1      84 2.8626
## 2      82 2.6940  2    0.16854 2.565 0.08308 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Má-especificação funcional: Gráficos

- O que acontece se identificarmos problemas de má-especificação e  $\log(\cdot)$  não resolve?

# Má-especificação funcional: Gráficos

- O que acontece se identificarmos problemas de má-especificação e  $\log(\cdot)$  não resolve?
- Análise de resíduos podem nos auxiliar nesta tarefa

# Má-especificação funcional: Gráficos

- O que acontece se identificarmos problemas de má-especificação e  $\log(\cdot)$  não resolve?
- Análise de resíduos podem nos auxiliar nesta tarefa
- Quando temos poucas variáveis independentes é super útil



# Má-especificação funcional: Gráficos

- O que acontece se identificarmos problemas de má-especificação e  $\log(\cdot)$  não resolve?
- Análise de resíduos podem nos auxiliar nesta tarefa
- Quando temos poucas variáveis independentes é super útil
- Quando temos muitas variáveis independentes essa tarefa pode ser bastante *cansativa*

# Má-especificação funcional: Gráficos

*dadossim* corresponde a um conjunto de dados simulados da forma

$$y = 0.8 + 0.7x - 0.3x^2 + u$$

```
modelo = lm(y~x, data = dadossim)
round(summary(modelo)$coef,6)
```

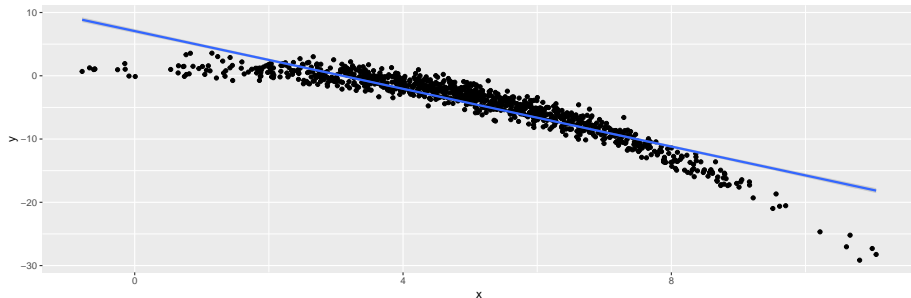
##		Estimate	Std. Error	t value	Pr(> t )
##	(Intercept)	7.065441	0.168237	41.99693	0
##	x	-2.281661	0.031069	-73.43750	0

```
summary(modelo)$r.squared
```

```
## [1] 0.8438445
```

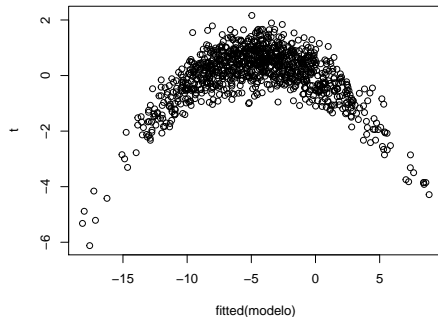
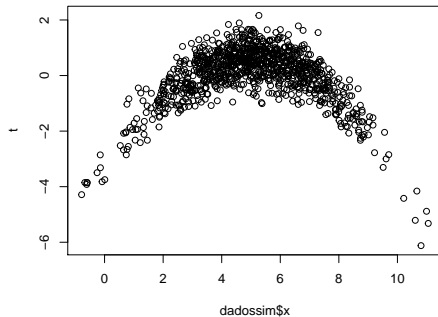
# Má-especificação funcional: Gráficos

```
library(ggplot2)
ggplot(dadoscim, aes(x,y)) + geom_point() +
  geom_smooth(method = "lm")
```



# Má-especificação funcional: Gráficos

```
t = rstudent(modelo) # studentized deleted residuals  
par(mfrow=c(1,2))  
plot(dadosim$x,t)  
plot(fitted(modelo),t)
```



# Má-especificação funcional: Gráficos

$$y = 0.8 + 0.7x - 0.3x^2 + u$$

```
modelo2 = lm(y~x+I(x^2), data = dadossim)
round(summary(modelo2)$coef,6)
```

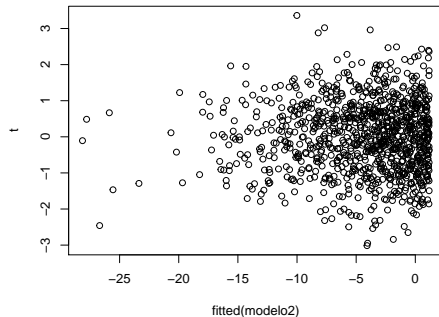
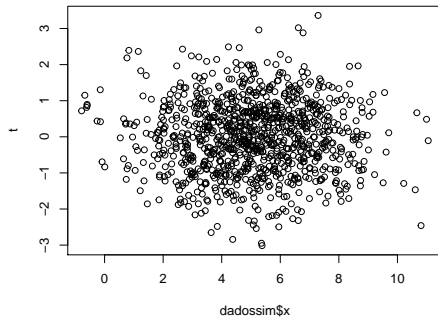
##		Estimate	Std. Error	t value	Pr(> t )
##	(Intercept)	0.722095	0.154197	4.682942	3e-06
##	x	0.734947	0.062017	11.850820	0e+00
##	I(x^2)	-0.302733	0.006000	-50.458325	0e+00

```
summary(modelo2)$r.squared
```

```
## [1] 0.9560584
```

# Má-especificação funcional: Gráficos

```
t = rstudent(modelo2) # studentized deleted residuals  
par(mfrow=c(1,2))  
plot(dadosim$x,t)  
plot(fitted(modelo2),t)
```



# Má-especificação funcional: Gráficos

$$y = 0.8 + 0.7x_1 - 0.3x_2^2 + 0.8x_2 + u$$

```
modelo = lm(y~x1, data = dadossim)
round(summary(modelo)$coef,6)
```

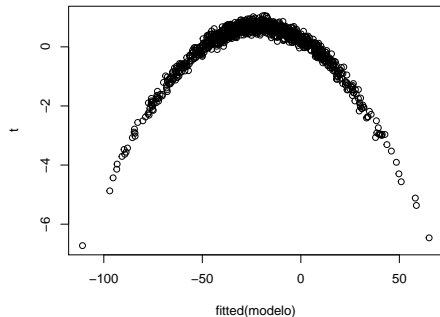
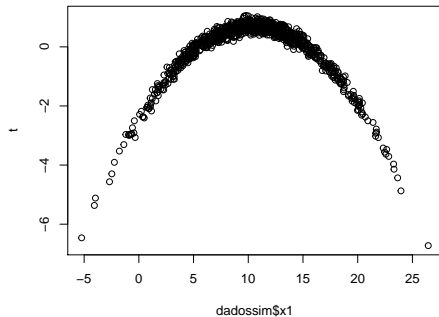
##		Estimate	Std. Error	t value	Pr(> t )
##	(Intercept)	36.063129	0.768887	46.90301	0
##	x1	-5.553633	0.066991	-82.90130	0

```
summary(modelo)$r.squared
```

```
## [1] 0.8731994
```

# Má-especificação funcional: Gráficos

```
t = rstudent(modelo) # studentized residuals  
par(mfrow = c(1,2))  
plot(dadosim$x1,t)  
plot(fitted(modelo),t)
```





# Má-especificação funcional: Gráficos

$$y = 0.8 + 0.7x_1 - 0.3x_2^2 + 0.8x_2 + u$$

```
modelo2 = lm(y~x1+I(x1^2), data = dadosim)
round(summary(modelo2)$coef,6)
```

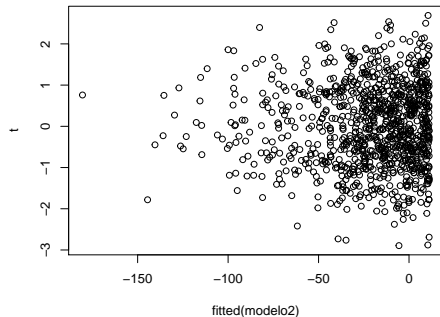
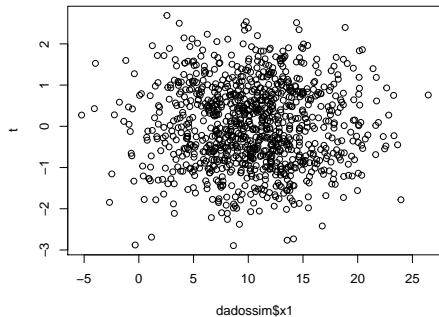
	Estimate	Std. Error	t value	Pr(> t )
## (Intercept)	10.683321	0.164435	64.96994	0
## x1	0.717248	0.031105	23.05865	0
## I(x1^2)	-0.300426	0.001414	-212.53722	0

```
summary(modelo2)$r.squared
```

```
## [1] 0.9972618
```

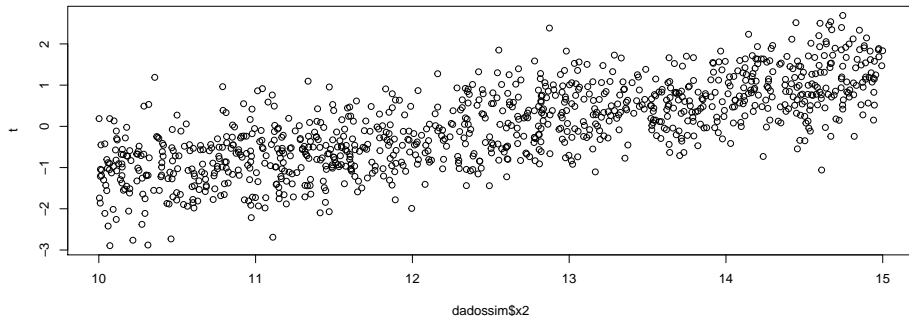
# Má-especificação funcional: Gráficos

```
t = rstudent(modelo2) # studentized residuals  
par(mfrow = c(1,2))  
plot(dadosim$x1,t)  
plot(fitted(modelo2),t)
```



# Má-especificação funcional: Gráficos para variáveis omitidas

```
t = rstudent(modelo2) # studentized residuals  
plot(dadosim$x2,t)
```



# Má-especificação funcional: Gráficos

$$y = 0.8 + 0.7x_1 - 0.3x_2^2 + 0.8x_2 + u$$

```
modelo3 = lm(y~x1+I(x1^2) + x2, data = dadossim)
round(summary(modelo3)$coef,6)
```

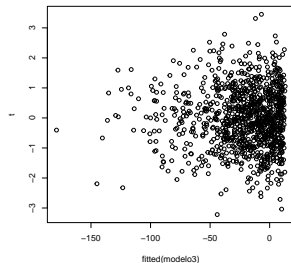
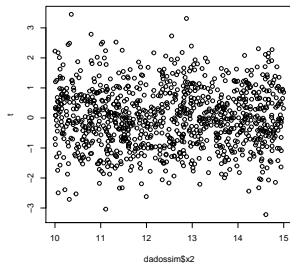
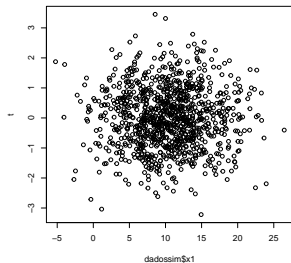
##		Estimate	Std. Error	t value	Pr(> t )
##	(Intercept)	0.946213	0.294581	3.212068	0.00136
##	x1	0.708472	0.020650	34.308370	0.00000
##	I(x1^2)	-0.300207	0.000938	-319.928238	0.00000
##	x2	0.784264	0.022038	35.587512	0.00000

```
summary(modelo3)$r.squared
```

```
## [1] 0.9987946
```

# Má-especificação funcional: Gráficos

```
t = rstudent(modelo3)
par(mfrow = c(1,3))
plot(dadosim$x1,t)
plot(dadosim$x2,t)
plot(fitted(modelo3),t)
```



# Má-especificação funcional: Gráficos

$$y = 0.8 + 0.7x_1 - 0.3x_1^2 + u$$

```
modelo = lm(y~x1+I(x1^2), data = dadossim)
round(summary(modelo)$coef,6)
```

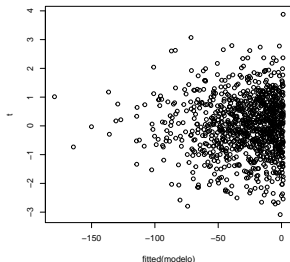
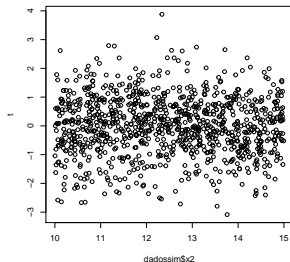
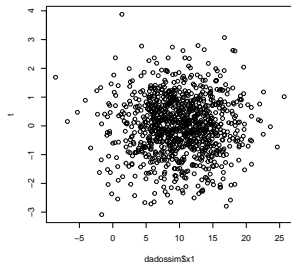
##		Estimate	Std. Error	t value	Pr(> t )
##	(Intercept)	0.914795	0.098091	9.325933	0
##	x1	0.691754	0.019082	36.252037	0
##	I(x1^2)	-0.300085	0.000924	-324.675923	0

```
summary(modelo)$r.squared
```

```
## [1] 0.9985805
```

# Má-especificação funcional: Gráficos

```
t = rstudent(modelo)
par(mfrow = c(1,3))
plot(dadosim$x1,t)
plot(dadosim$x2,t)
plot(fitted(modelo),t)
```



# Resumo do processo de modelagem

- ① EDA: Detectar outliers, definir formas funcionais, uma primeira olhada aos dados
- ② Ajustar o modelo de regressão
- ③ Verificar outliers e forma funcional ( $\hat{y}$  vs resíduos |  $X_j$  vs resíduos)
- ④ Verificar variáveis omitidas (variáveis omitidas vs resíduos)
- ⑤ Verificar homocedasticidade: se tivermos evidência de heterocedasticidade, calcular a variância dos  $\beta$ 's de forma robusta
- ⑥ Verificar não correlação dos erros (gráfico de sequência e ACF)
- ⑦ Verificar Normalidad\* (gráfico de probabilidade normal)
- ⑧ Interpretar os parâmetros e fazer inferência estatística (Usar as versões robustas dos testes t e F)



# Leituras recomendadas

## Leituras recomendadas

- Wooldridge, Jeffrey M. *Introdução à Econometria: Uma abordagem moderna*. (2016). Cengage Learning. – **Cap 9**