

# Mô hình hồi qui đa thức

**Tuan V. Nguyen**

Garvan Institute of Medical Research

University of New South Wales (UNSW Sydney), Australia

University of Technology, Sydney (UTS), Australia

Ton Duc Thang University, Vietnam



# Mô hình hồi qui đa thức

- Mô hình hồi qui đa thức (polynomial regression)
- Hoán chuyển dữ liệu

# Mô hình đa thức

- Mô hình tuyến tính = mô hình đa thức bậc 1

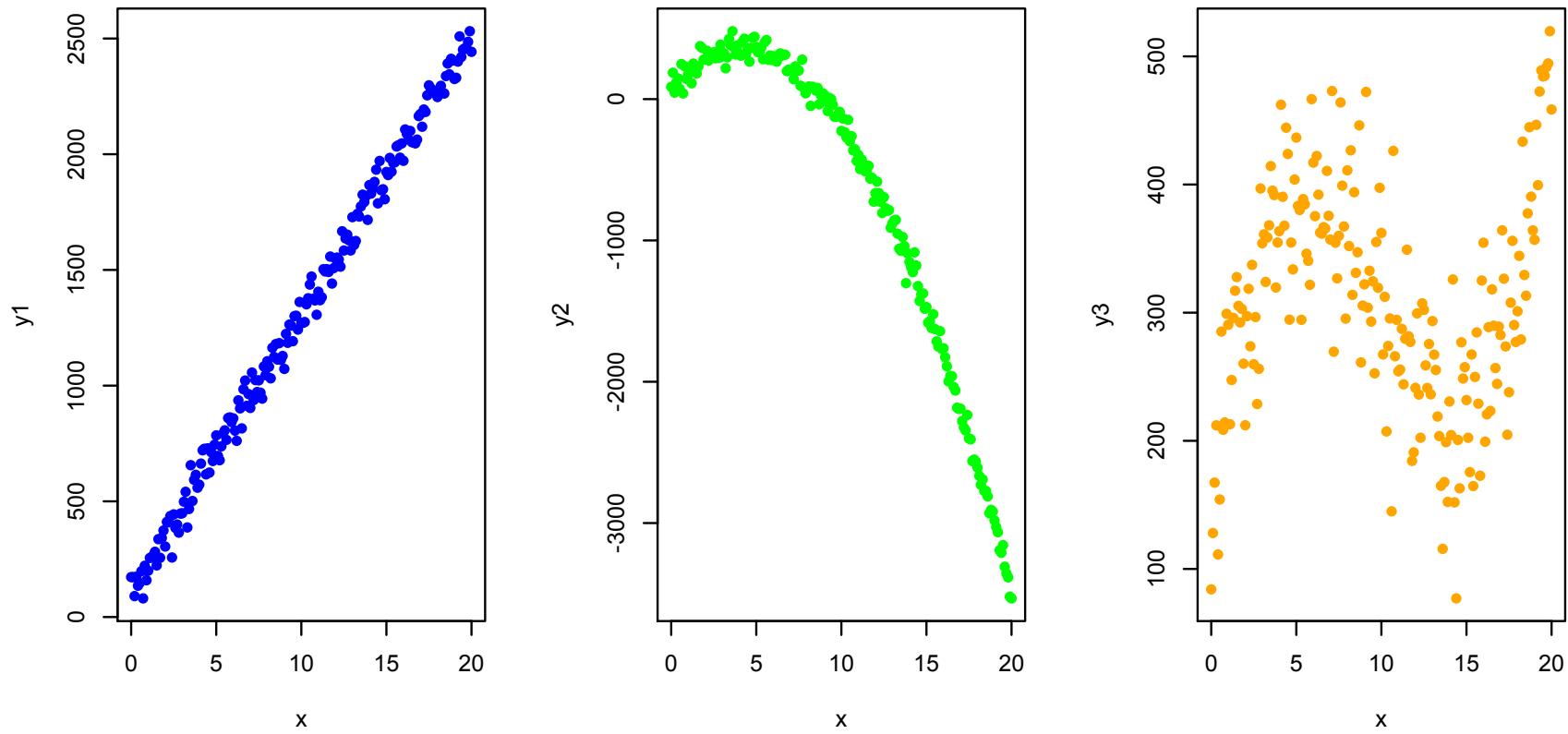
$$Y = \beta_0 + \beta_1 X + \varepsilon$$

- Mô hình đa thức bậc 2

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon$$

- Mô hình đa thức bậc 3

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \varepsilon$$



```
x = seq(from=0, to=20, by=0.1)
y1 = 100 + 120*x + rnorm(length(x), mean=10, sd=50)
y2 = 100 + 120*x -15*x^2 + rnorm(length(x), mean=10, sd=50)
y3 = 100 + 120*x - 15*x^2 + 0.5*x^3 + rnorm(length(x), mean=10, sd=50)
par(mfrow=c(1,3))
plot(y1 ~ x, pch=16, col="blue")
plot(y2 ~ x, pch=16, col="green")
plot(y3 ~ x, pch=16, col="orange")
```

# Mô phỏng mô hình đa thức bậc 3

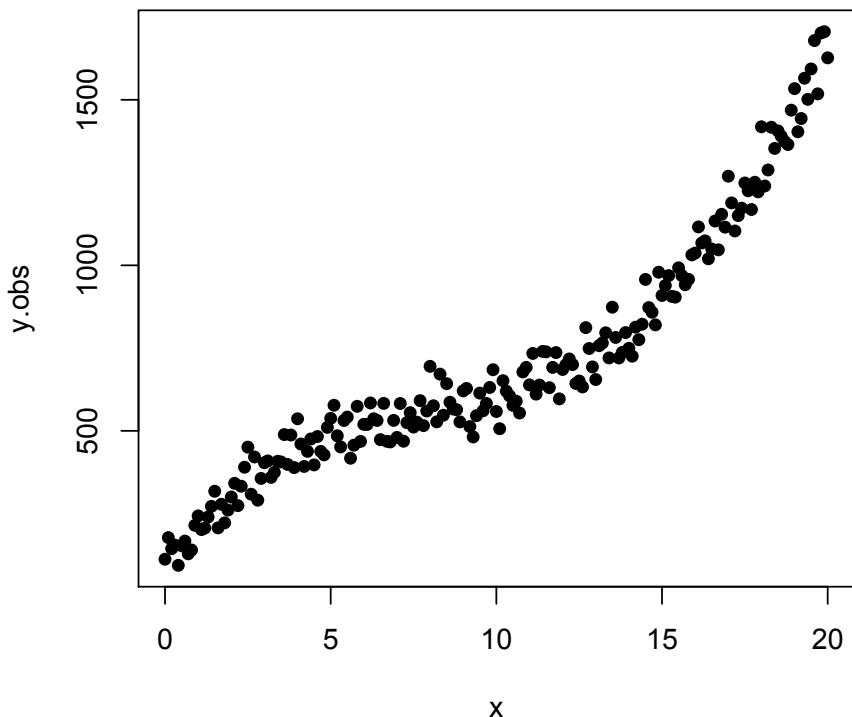
- Mô phỏng mô hình 'thật':

$$X = 0.0, 0.1, 0.2, \dots, 20$$

$$Y = 100 + 120X - 12X^2 + 0.5X^3 + \text{error}$$

- Dùng R

```
set.seed(20)
x = seq(from=0, to=20, by=0.1)
err = rnorm(length(x), mean=10, sd=50)
y.obs = 100 + 120*x - 12*x^2 + 0.5*x^3 + err
plot(y.obs ~ x, pch=16)
```



# Chúng ta dùng mô hình đa thức bậc 1 (tuyến tính)

```
m = lm(y.obs ~ x)  
summary(m)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	111.746	17.899	6.243	2.54e-09 ***
x	<b>59.717</b>	<b>1.548</b>	<b>38.572</b>	< 2e-16 ***
---				

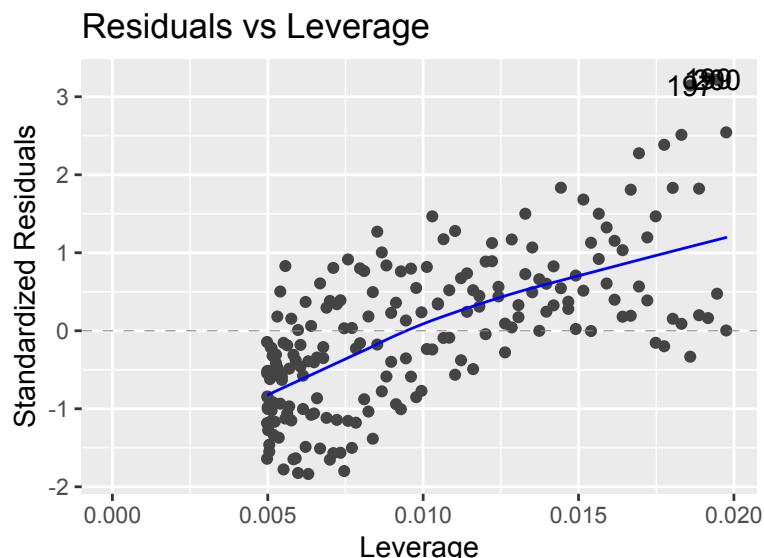
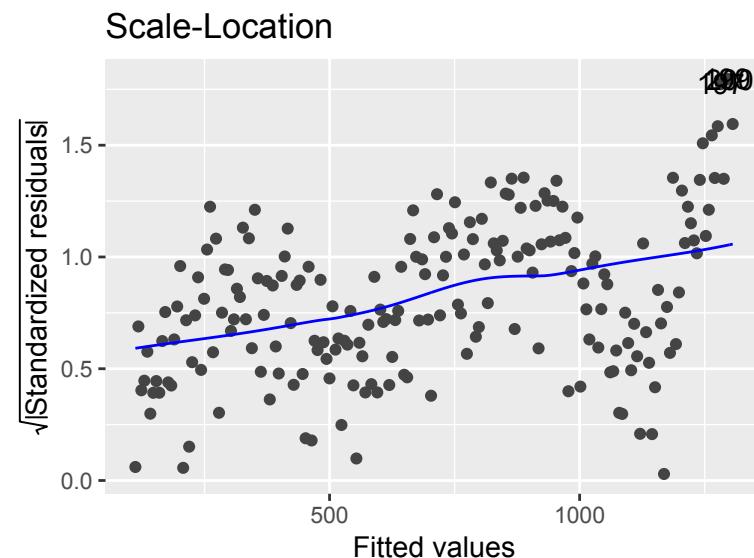
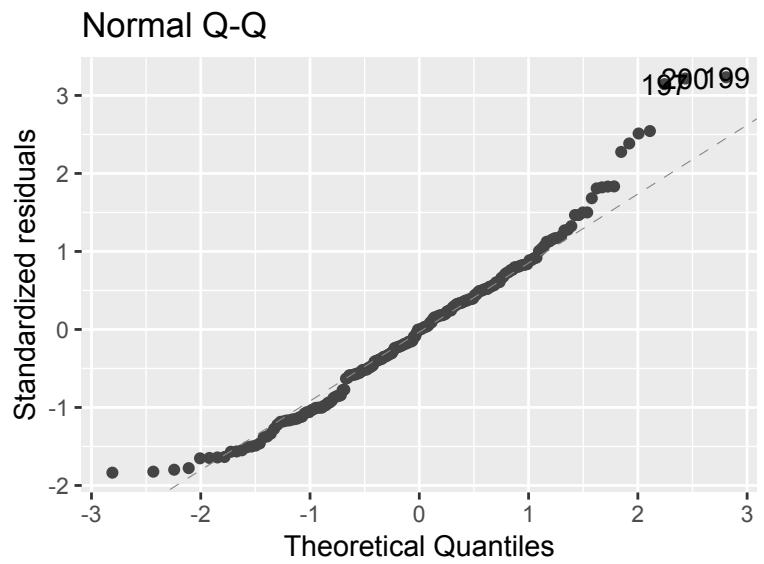
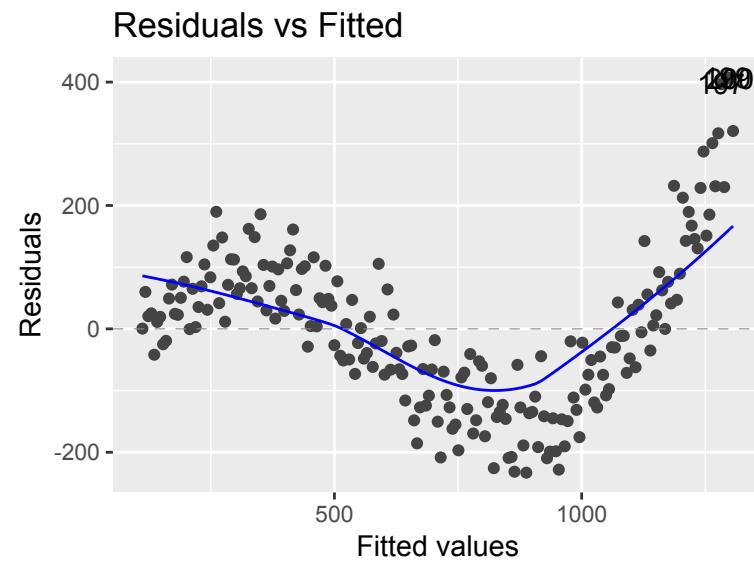
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 127.4 on 199 degrees of freedom  
Multiple R-squared: 0.882, Adjusted R-squared: 0.8814  
F-statistic: 1488 on 1 and 199 DF, p-value: < 2.2e-16

# Kiểm tra mô hình hồi qui tuyến tính

```
library(ggfortify)
```

```
autoplot(m)
```



# Chúng ta dùng mô hình đa thức bậc 3

```
m = lm(y.obs ~ x + I(x^2) + I(x^3))
```

```
summary(m)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	106.47183	14.59743	7.294	7.18e-12	***
x	121.32984	6.33666	19.147	< 2e-16	***
I(x^2)	-12.05405	0.73728	-16.349	< 2e-16	***
I(x^3)	0.49769	0.02423	20.541	< 2e-16	***
---					

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 52.71 on 197 degrees of freedom

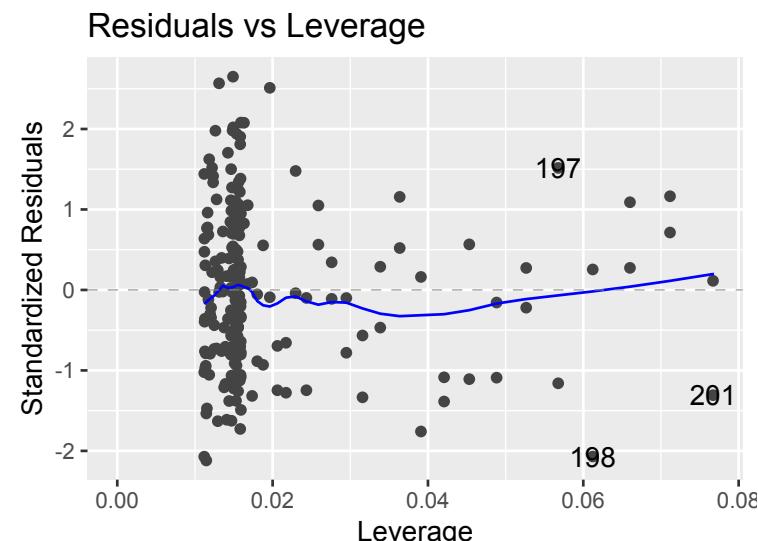
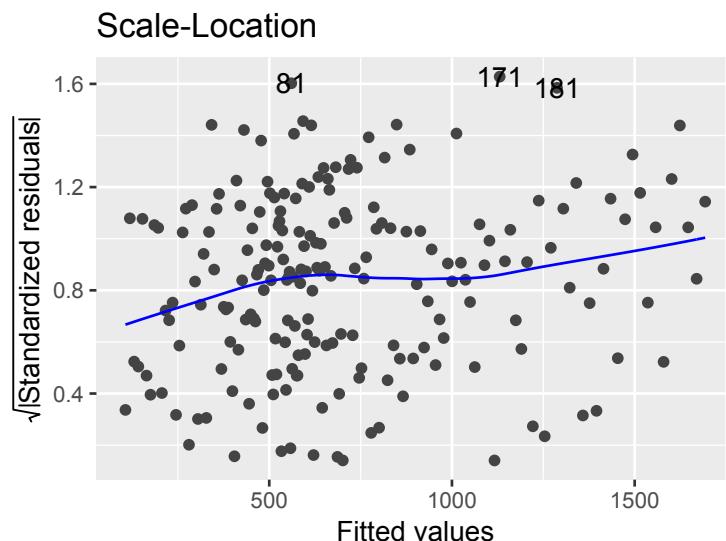
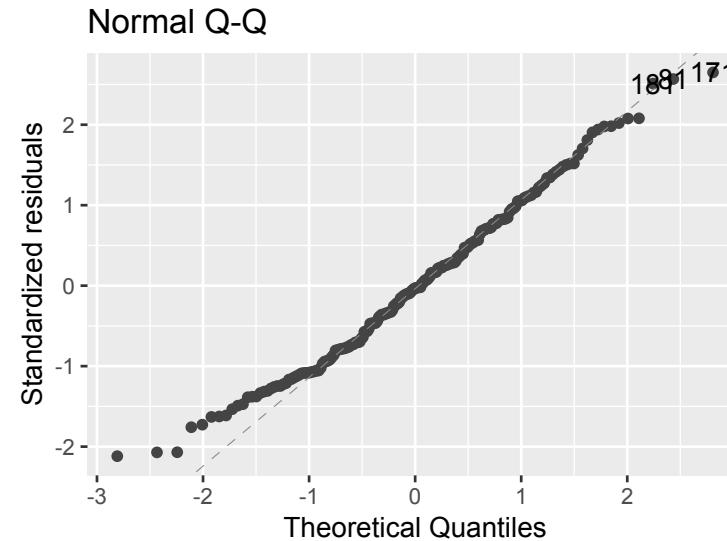
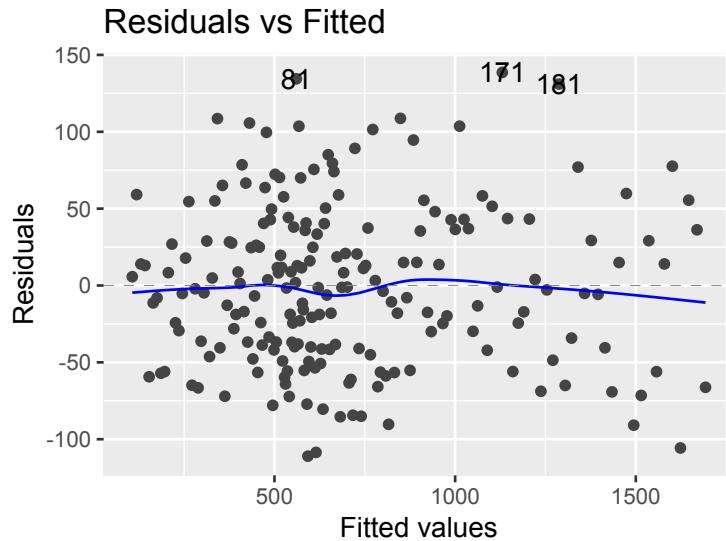
Multiple R-squared: 0.98, Adjusted R-squared: 0.9797

F-statistic: 3217 on 3 and 197 DF, p-value: < 2.2e-16

# Kiểm tra mô hình hồi qui bậc 3

```
library(ggfortify)
```

```
autoplot(m)
```



# Hoán chuyển Y: Phương pháp Box-Cox

Box-Cox đề nghị hoán chuyển Y sao cho mối liên quan giữa X và Y là tuyến tính

$$g(y) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0 \\ \log(y), & \text{if } \lambda = 0 \end{cases}$$

Phương pháp Box-Cox được triển khai trong R qua package "MASS"

# Dữ liệu về giá nhà Boston

## # Phân bố giá nhà MEDV

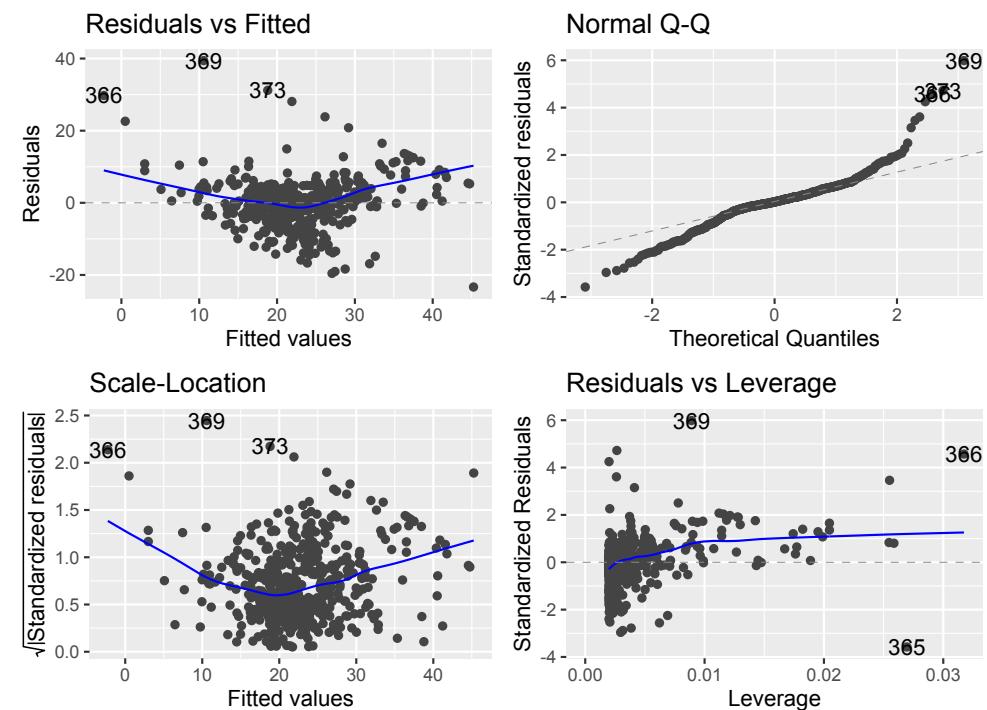
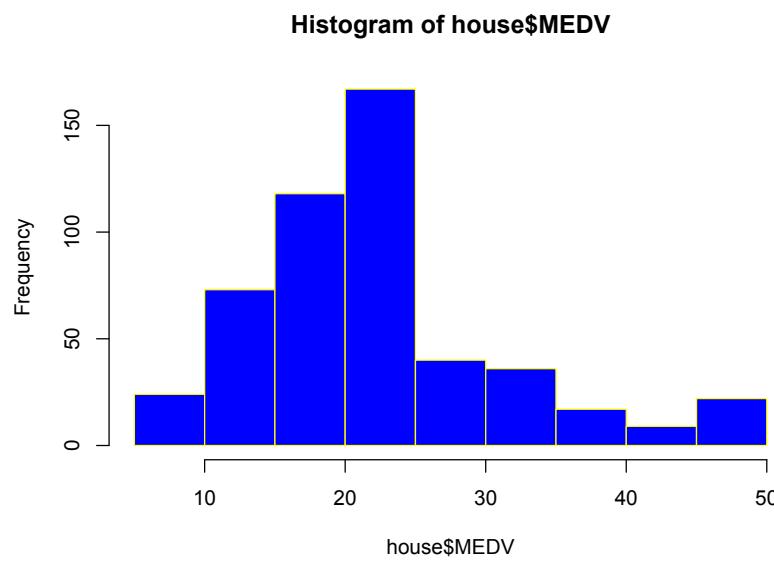
```
hist(house$MEDV, col="blue", border="yellow")
```

## # Mô hình hồi qui tuyến tính

```
m = lm(MEDV ~ rooms, data=house)
```

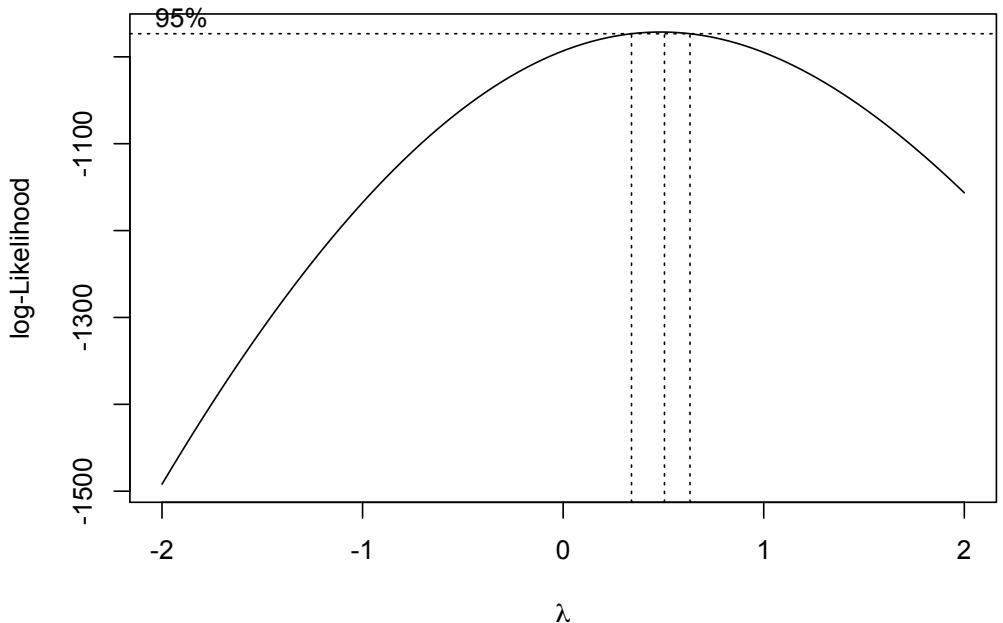
```
library(ggfortify)
```

```
autoplot(m)
```



# Box-Cox transformation of MEDV

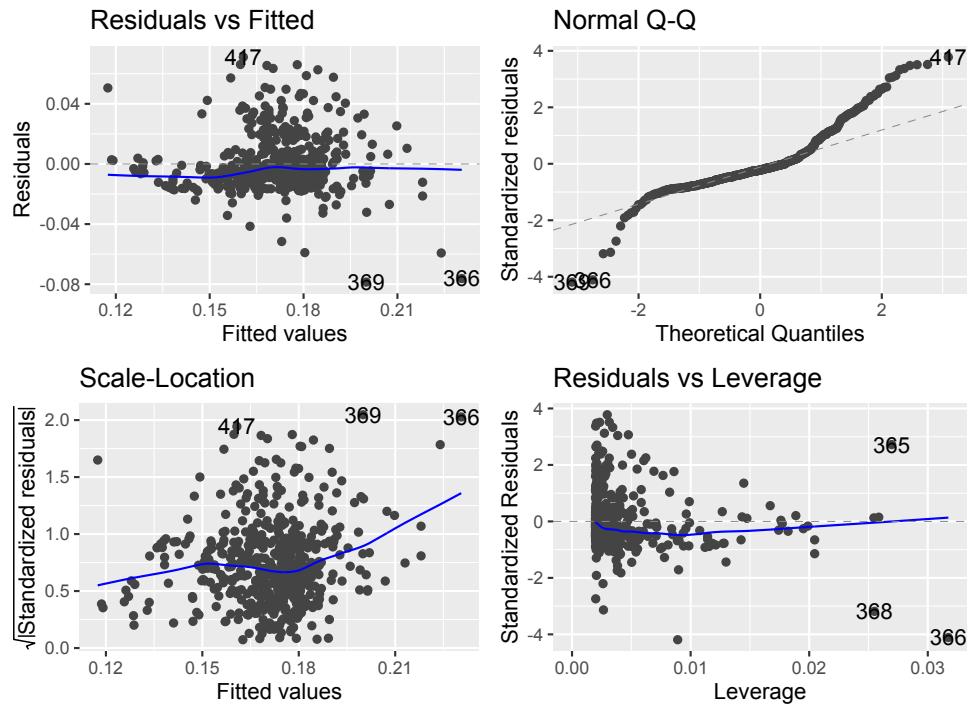
```
library(MASS)
t = boxcox(MEDV ~ rooms, data=house)
lambda = t$x[which.max(t$y)]
lambda
0.5050505
```



# Box-Cox transformation of MEDV

```
# create a new Y variable
```

```
house$newY = (house$MEDV^0.5 -  
1)/house$MEDV  
  
# Fit new model with new Y  
  
m = lm(newY ~ rooms, data=house)  
  
summary(m)  
autoplot(m)
```



# Tóm lược

- Mô hình hồi qui tuyến tính có thể mở rộng để ‘fit’ các mối liên quan phi tuyến tính
- Phương pháp Box-Cox có thể có ích trong việc hoán chuyển dữ liệu