Linear Regression

Prof. Dr. Salmai Qari

Contents


```
require(wooldridge)
require(dplyr)
```

1 Loading and inspecting the data

```
data(hprice1)
```

1.1 A glimpse

```
glimpse(hprice1)
```

```
## Rows: 88
## Columns: 10
## $ price    <dbl> 300.000, 370.000, 191.000, 195.000, 373.000, 466.275, 332.500~
## $ assess   <dbl> 349.1, 351.5, 217.7, 231.8, 319.1, 414.5, 367.8, 300.2, 236.1~
## $ bdrms    <int> 4, 3, 3, 3, 4, 5, 3, 3, 3, 3, 4, 5, 3, 3, 3, 4, 4, 3, 3, 4, 3~
## $ lotsize  <dbl> 6126, 9903, 5200, 4600, 6095, 8566, 9000, 6210, 6000, 2892, 6~
## $ sqrft    <int> 2438, 2076, 1374, 1448, 2514, 2754, 2067, 1731, 1767, 1890, 2~
## $ colonial <int> 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1~
## $ lprice   <dbl> 5.703783, 5.913503, 5.252274, 5.273000, 5.921578, 6.144775, 5~
## $ lassess  <dbl> 5.855359, 5.862210, 5.383118, 5.445875, 5.765504, 6.027073, 5~
## $ llotsize <dbl> 8.720297, 9.200593, 8.556414, 8.433811, 8.715224, 9.055556, 9~
## $ lsqrft   <dbl> 7.798934, 7.638198, 7.225482, 7.277938, 7.829630, 7.920810, 7~
```

1.2 Use the help function to inspect details of the dataset

```
?hprice1
```

```
## starting httpd help server ... done
```

2 A first regression

```
my.lm1 <- lm(price ~ sqrft, data=hprice1)
```

```
summary(my.lm1)
```

```
##
## Call:
## lm(formula = price ~ sqrft, data = hprice1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -117.112  -36.348   -6.503   31.701  235.253
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.20415   24.74261   0.453    0.652
## sqrft        0.14021    0.01182  11.866   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63.62 on 86 degrees of freedom
## Multiple R-squared:  0.6208, Adjusted R-squared:  0.6164
## F-statistic: 140.8 on 1 and 86 DF,  p-value: < 2.2e-16
```

2.1 Note:

We can define the formula separately

```
my.formula <- as.formula("price ~ sqrft")
my.lm2 <- lm(my.formula, data=hprice1)
```

2.1.1 Further hint

```
as.character(my.formula)
```

```
## [1] "~"     "price" "sqrft"
```

2.2 Using predict for predictions

```
my.price.hat <- predict(my.lm1)
```

```
head(my.price.hat)
```

```
##        1        2        3        4        5        6
## 353.0385 302.2821 203.8540 214.2296 363.6945 397.3452
```

3 Random data subsets

A simple way of obtaining a random sample is the following:

- First, reorder rows randomly
- Second, take (for example) the first 80 percent of the rows

```
my.random.index <- sample(nrow(hprice1), replace = FALSE)
head(my.random.index)
```

```
## [1] 85 88 65 37 11 27
```

4 Tasks:

- Write a function my.RSS with two inputs: a string describing the formula and the dataset. The function carries out the regression and returns the RSS (residual sum of squares)
- Write a function that has two input arguments: a data and a number for the split (e.g. 0.8 for 80 percent). The function returns randomly selected training and test datasets

4.1 Task 1: Testing

```
my.RSS(my.str = "price ~ sqrft", my.data=hprice1)
```

```
## [1] 348053.4
```

4.2 Task 2: Testing 1

```
my.datasets <- my.data.select(my.data = hprice1, split=0.8)
```

```
glimpse(my.datasets)
```

```
## List of 2
##  $ traindata:'data.frame':   70 obs. of  10 variables:
##   ..$ price   : num [1:70] 230 225 230 330 230 295 360 315 246 251 ...
##   ..$ assess  : num [1:70] 276 291 218 360 212 ...
##   ..$ bdrms   : int [1:70] 3 3 4 3 3 3 4 4 4 3 ...
##   ..$ lotsize : num [1:70] 4054 7566 4806 8178 5305 ...
##   ..$ sqrft   : int [1:70] 1736 1567 1573 2186 1171 1837 2750 1638 1928 1630 ...
##   ..$ colonial: int [1:70] 1 0 1 1 0 1 1 1 1 1 ...
##   ..$ lprice  : num [1:70] 5.44 5.42 5.44 5.8 5.44 ...
##   ..$ lassess : num [1:70] 5.62 5.67 5.38 5.88 5.36 ...
##   ..$ llotsize: num [1:70] 8.31 8.93 8.48 9.01 8.58 ...
##   ..$ lsqrft  : num [1:70] 7.46 7.36 7.36 7.69 7.07 ...
##   ..- attr(*, "time.stamp")= chr "25 Jun 2011 23:03"
##  $ testdata :'data.frame':   18 obs. of  10 variables:
##   ..$ price   : num [1:18] 310 300 332 191 190 ...
##   ..$ assess  : num [1:18] 326 416 368 218 212 ...
##   ..$ bdrms   : int [1:18] 4 5 3 3 4 2 3 4 3 3 ...
##   ..$ lotsize : num [1:18] 8602 7047 9000 5200 3500 ...
##   ..$ sqrft   : int [1:18] 1835 2634 2067 1374 1702 2205 1536 1674 1587 1715 ...
##   ..$ colonial: int [1:18] 1 1 1 0 0 0 1 1 1 0 ...
##   ..$ lprice  : num [1:18] 5.74 5.7 5.81 5.25 5.25 ...
##   ..$ lassess : num [1:18] 5.79 6.03 5.91 5.38 5.36 ...
##   ..$ llotsize: num [1:18] 9.06 8.86 9.1 8.56 8.16 ...
##   ..$ lsqrft  : num [1:18] 7.51 7.88 7.63 7.23 7.44 ...
```

```
##   ..- attr(*, "time.stamp")= chr "25 Jun 2011 23:03"
```

4.3 Task 2: Testing 2

```
my.datasets <- my.data.select(my.data = hprice1, split=0.8)
```

```
glimpse(my.datasets)
```

```
## List of 2
##  $ traindata:'data.frame':   70 obs. of  10 variables:
##   ..$ price   : num [1:70] 350 270 268 268 725 ...
##   ..$ assess  : num [1:70] 355 319 267 254 709 ...
##   ..$ bdrms   : int [1:70] 4 3 3 3 5 3 3 4 3 4 ...
##   ..$ lotsize : num [1:70] 9773 7800 5642 5167 31000 ...
##   ..$ sqrft   : int [1:70] 2051 2124 1376 1980 3662 1768 1662 1850 1374 1696 ...
##   ..$ colonial: int [1:70] 1 1 1 1 0 0 1 1 0 1 ...
##   ..$ lprice  : num [1:70] 5.86 5.6 5.59 5.59 6.59 ...
##   ..$ lassess : num [1:70] 5.87 5.76 5.59 5.54 6.56 ...
##   ..$ llotsize: num [1:70] 9.19 8.96 8.64 8.55 10.34 ...
##   ..$ lsqrft  : num [1:70] 7.63 7.66 7.23 7.59 8.21 ...
##   ..- attr(*, "time.stamp")= chr "25 Jun 2011 23:03"
##  $ testdata :'data.frame':   18 obs. of  10 variables:
##   ..$ price   : num [1:18] 242 209 206 225 230 ...
##   ..$ assess  : num [1:18] 252 231 236 250 212 ...
##   ..$ bdrms   : int [1:18] 4 4 3 3 3 3 5 4 2 5 ...
##   ..$ lotsize : num [1:18] 4950 5600 6000 18838 5305 ...
##   ..$ sqrft   : int [1:18] 1774 1674 1767 1294 1171 1536 2293 1732 2205 2617 ...
##   ..$ colonial: int [1:18] 1 1 0 0 0 1 1 0 0 1 ...
##   ..$ lprice  : num [1:18] 5.49 5.34 5.33 5.42 5.44 ...
##   ..$ lassess : num [1:18] 5.53 5.44 5.46 5.52 5.36 ...
##   ..$ llotsize: num [1:18] 8.51 8.63 8.7 9.84 8.58 ...
##   ..$ lsqrft  : num [1:18] 7.48 7.42 7.48 7.17 7.07 ...
##   ..- attr(*, "time.stamp")= chr "25 Jun 2011 23:03"
```