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] 25 68 70  9 47  7
```

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] 225 318 111 268 375 ...
##   ..$ assess  : num [1:70] 294 295 202 254 354 ...
##   ..$ bdrms   : int [1:70] 3 4 4 3 5 3 3 3 4 2 ...
##   ..$ lotsize : num [1:70] 6003 92681 4315 5167 6637 ...
##   ..$ sqrft   : int [1:70] 1768 1696 1535 1980 2293 1574 1932 2106 2750 2205 ...
##   ..$ colonial: int [1:70] 0 1 1 1 1 0 0 1 1 0 ...
##   ..$ lprice  : num [1:70] 5.42 5.76 4.71 5.59 5.93 ...
##   ..$ lassess : num [1:70] 5.68 5.69 5.31 5.54 5.87 ...
##   ..$ llotsize: num [1:70] 8.7 11.44 8.37 8.55 8.8 ...
##   ..$ lsqrft  : num [1:70] 7.48 7.44 7.34 7.59 7.74 ...
##   ..- attr(*, "time.stamp")= chr "25 Jun 2011 23:03"
##  $ testdata :'data.frame':   18 obs. of  10 variables:
##   ..$ price   : num [1:18] 209 280 478 373 350 ...
##   ..$ assess  : num [1:18] 289 336 478 319 355 ...
##   ..$ bdrms   : int [1:18] 4 4 7 4 4 3 4 3 4 3 ...
##   ..$ lotsize : num [1:18] 6400 8460 8400 6095 9773 ...
##   ..$ sqrft   : int [1:18] 1854 1915 3529 2514 2051 2048 1732 1715 2438 1502 ...
##   ..$ colonial: int [1:18] 1 1 1 1 1 1 0 0 1 0 ...
##   ..$ lprice  : num [1:18] 5.34 5.63 6.17 5.92 5.86 ...
##   ..$ lassess : num [1:18] 5.67 5.82 6.17 5.77 5.87 ...
##   ..$ llotsize: num [1:18] 8.76 9.04 9.04 8.72 9.19 ...
##   ..$ lsqrft  : num [1:18] 7.53 7.56 8.17 7.83 7.63 ...
```

```
##   ..- 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] 370 330 332 268 471 ...
##   ..$ assess  : num [1:70] 352 360 368 267 482 ...
##   ..$ bdrms   : int [1:70] 3 3 3 3 5 4 4 3 4 3 ...
##   ..$ lotsize : num [1:70] 9903 8178 9000 5642 15834 ...
##   ..$ sqrft   : int [1:70] 2076 2186 2067 1376 2617 1732 2321 1715 1760 1536 ...
##   ..$ colonial: int [1:70] 1 1 1 1 1 0 1 0 1 1 ...
##   ..$ lprice  : num [1:70] 5.91 5.8 5.81 5.59 6.16 ...
##   ..$ lassess : num [1:70] 5.86 5.88 5.91 5.59 6.18 ...
##   ..$ llotsize: num [1:70] 9.2 9.01 9.1 8.64 9.67 ...
##   ..$ lsqrft  : num [1:70] 7.64 7.69 7.63 7.23 7.87 ...
##   ..- attr(*, "time.stamp")= chr "25 Jun 2011 23:03"
##  $ testdata :'data.frame':   18 obs. of  10 variables:
##   ..$ price   : num [1:18] 285 306 206 466 244 ...
##   ..$ assess  : num [1:18] 314 389 236 414 307 ...
##   ..$ bdrms   : int [1:18] 4 2 3 5 4 3 3 4 3 4 ...
##   ..$ lotsize : num [1:18] 6000 20700 6000 8566 7893 ...
##   ..$ sqrft   : int [1:18] 2336 2205 1767 2754 2090 1890 2768 1928 1932 2589 ...
##   ..$ colonial: int [1:18] 1 0 0 1 1 0 0 1 0 1 ...
##   ..$ lprice  : num [1:18] 5.65 5.72 5.33 6.14 5.5 ...
##   ..$ lassess : num [1:18] 5.75 5.96 5.46 6.03 5.73 ...
##   ..$ llotsize: num [1:18] 8.7 9.94 8.7 9.06 8.97 ...
##   ..$ lsqrft  : num [1:18] 7.76 7.7 7.48 7.92 7.64 ...
##   ..- attr(*, "time.stamp")= chr "25 Jun 2011 23:03"
```

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

```
summary(my.lm2)
```

```
##
## Call:
## lm(formula = my.formula, data = my.datasets$traindata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -129.973  -36.697   -6.127   30.926  224.476
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.70835   28.06883   0.168    0.867
## sqrft        0.14540    0.01368  10.630 4.24e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66.81 on 68 degrees of freedom
```

```
## Multiple R-squared:  0.6243, Adjusted R-squared:  0.6188
## F-statistic:   113 on 1 and 68 DF,  p-value: 4.242e-16
```

```
predict.test <- predict(my.lm2, newdata = my.datasets$testdata)
```

```
head(predict.test)
```

```
##       11       75        9        6       83       10
## 344.3545 325.3075 261.6239 405.1302 308.5869 279.5077
```

5 Task III

Write a function that accepts a formula, a dataset and split as an input. It creates the training and test datasets, and runs the regression using the training dataset. Finally, it calculates the RSS for the training dataset and the test dataset.

```
my.RSSTrainTest("price ~ sqrft", my.data=hprice1, split=0.8)
```

```
## $rssTrain
## [1] 267114.3
##
## $rssTest
## [1] 84537.46
```

```
my.RSSTrainTest("price ~ sqrft", my.data=hprice1, split=0.8)
```

```
## $rssTrain
## [1] 306135.4
##
## $rssTest
## [1] 43493.8
```