Lab03 Linear Regression

Diego Castro

10/28/2021

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

setwd ("C:/Users/usuario/OneDrive - University of East Anglia/PhD/First Semestre/Econometrics/Laboratories/Lab3") list.files()

library (ggplot2)

library (stargazer)

library (data.table)

install.packages("Hmisc")

library(Hmisc)

```
#Lab Linear Regresion
library (ggplot2)
library (stargazer)
```

```
## Warning: package 'stargazer' was built under R version 4.1.1
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary
Statistics Tables.
```

```
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
library (data.table)
```

```
## Warning: package 'data.table' was built under R version 4.1.1
```

```
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.1.1
```

```
## Loading required package: lattice
```

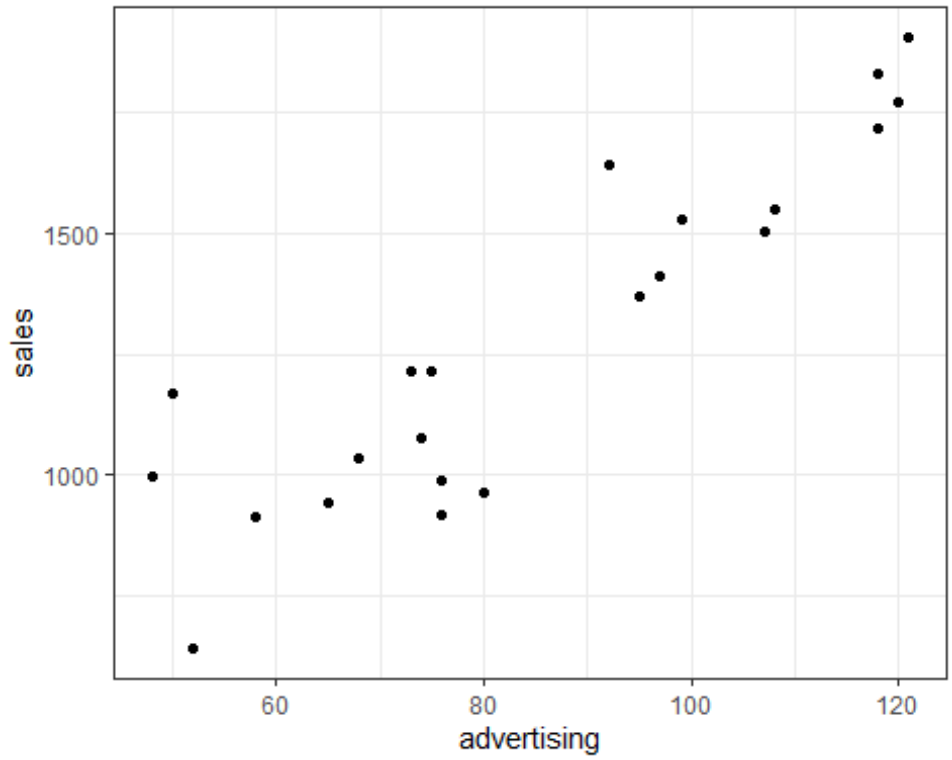
```
## Loading required package: survival
```

```
## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units

#This is a ".csv" file. In order to load a csv file into R we need to use the
function "read.csv()". When using this function we must name the data while
loading it. For now, I will name the data "sales".
sales <- read.csv("sales-data.csv")

#The next step is to convert the data into the data.table format. We do this
using the function data.table and, for convenience, we will add "dt" to the
file name so that we know it is in the data.table format

dt.sales <- data.table(sales)

rm(sales)
ncol(dt.sales)

## [1] 2

nrow(dt.sales)

## [1] 22

colnames(dt.sales)

## [1] "sales"       "advertising"

head(dt.sales)

##    sales advertising
## 1:   999          48
## 2:  1169          50
## 3:  1036          68
## 4:   643          52
## 5:   988          76
## 6:  1076          74

stargazer(dt.sales, type = "text")

##
## ===============================================================
## Statistic   N    Mean    St. Dev. Min Pctl(25) Pctl(75)  Max
## ---------------------------------------------------------------
## sales       22 1,286.636 353.621  643  990.8   1,543.8  1,905
## advertising 22  85.000    23.759  48    69.2     105     121
## ---------------------------------------------------------------
```

```
summary(dt.sales)
```

```
##      sales          advertising
##  Min.   : 643.0   Min.   : 48.00
##  1st Qu.: 990.8   1st Qu.: 69.25
##  Median :1215.0   Median : 78.00
##  Mean   :1286.6   Mean   : 85.00
##  3rd Qu.:1543.8   3rd Qu.:105.00
##  Max.   :1905.0   Max.   :121.00
```

```
qplot(data = dt.sales , x= advertising , y= sales , geom = "point") +
theme_bw()
```




```
##Correlation
```

```
dt.sales [, cor(sales, advertising)]
```

```
## [1] 0.9003409
```

```
## Is the correlation Statistically significant
dt.sales[, rcorr(sales, advertising, type = "pearson")]
```

```
##     x   y
## x 1.0 0.9
## y 0.9 1.0
##
## n= 22
##
```

```
##
## P
##   x  y
## x     0
## y  0
```

```
##Simple Regression Analysis
lm.sales <- lm(sales ~ advertising, data = dt.sales)
summary(lm.sales)
```

```
##
## Call:
## lm(formula = sales ~ advertising, data = dt.sales)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -254.63  -71.78  -17.34   82.97  351.38
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  147.590    127.618   1.157    0.261
## advertising   13.401      1.448   9.252 1.15e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 157.7 on 20 degrees of freedom
## Multiple R-squared:  0.8106, Adjusted R-squared:  0.8011
## F-statistic:  85.6 on 1 and 20 DF,  p-value: 1.15e-08
```

```
##Alternatevely
stargazer(lm.sales, type = "text")
```

```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                                sales
## -----------------------------------------------
## advertising                  13.401***
##                               (1.448)
##
## Constant                      147.590
##                              (127.618)
##
## -----------------------------------------------
## Observations                    22
## R2                             0.811
## Adjusted R2                    0.801
## Residual Std. Error      157.691 (df = 20)
## F Statistic           85.604*** (df = 1; 20)
```

```
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

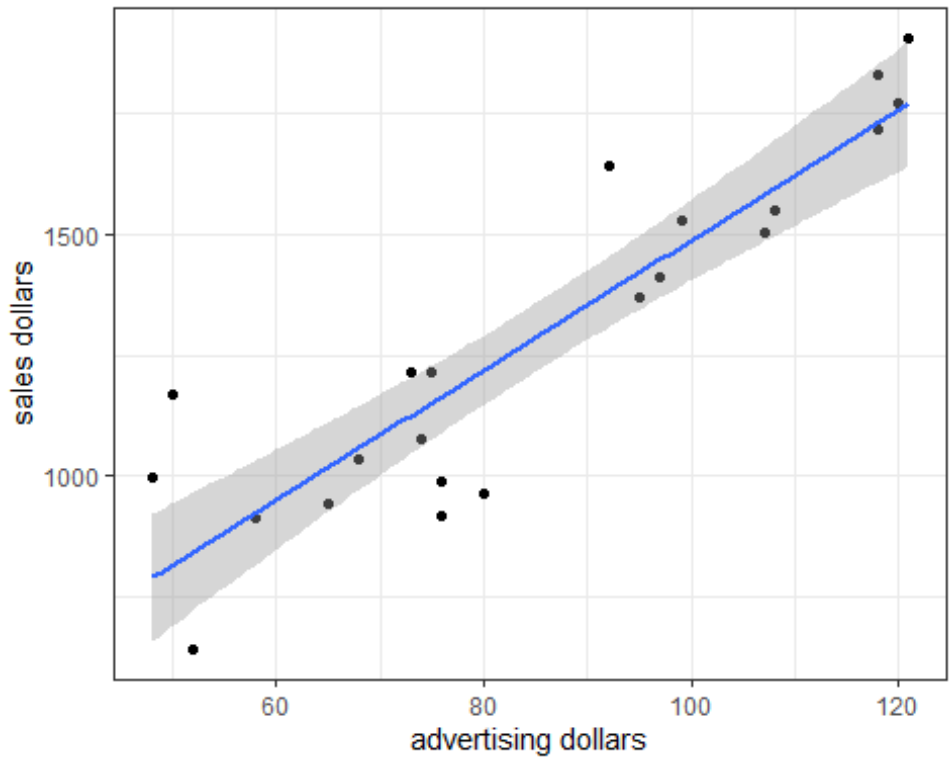
```
## Extracting the parameters of the estimated regression equation using the coefficients function
coeffs = coefficients(lm.sales)
coeffs
```

```
## (Intercept) advertising
##   147.59047    13.40054
```

```
##Interpretation
#B0 = 147.6 gives us the average sales level when the advertising investment is zero.
#B1 = 13.4 gives us the increase in sales that results from a 1 unit (dollar) increase in advertising investment.
#BR2 gives us the percentage of the variation in sales that is explained by the variation in the advertising investment.

#Plots
qplot(data = dt.sales, x=advertising, y=sales, geom = c("point", "smooth"), method=lm) + theme_bw() + labs(x = "advertising dollars", y = "sales dollars")
```

```
## Warning: Ignoring unknown parameters: method
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```r
#Predicted Values
#Obtain the predicted sales for an advertising investment of 100

advertising=100
sales=coeffs[1] + coeffs[2]*advertising
sales
```

```
## (Intercept)
##    1487.644
```

```r
#Alternatively, you can use the "predict" function to do this automatically.
We first wrap the parameters inside a new data table variable called newdata

my.budget = data.table(advertising=100)
predict(lm.sales, my.budget, interval = "predict")
```

```
##        fit      lwr      upr
## 1 1487.644 1148.274 1827.014
```