CWRU DSCI351-351M-453: Week11b-p-Simple-LinRegrAlgorithm-LPAwR

2008-351-351m-451-w11b-p-LinRegrAlgo-LPAwR

Profs: R. H. French, L. S. Bruckman, P. Leu, K. Davis, S. Cirlos

10 November, 2022

Contents

12.2.2.1 Understanding simple linear regression

12.2.2.1.1 Build and use our own simple linear regression algorithm

- Create multiple linear regression models in R
- Perform diagnostic tests of such models
- Score new data using a linear regression model
- Examine how well the model predicts the new data

Regression seeks to obtain the model coefficients

- that explain the variable's relationship the best
- but such a model only seldom reflects the relationship entirely

Indeed, measurement error,

- And also attributes that are not included in the analysis
  - affect also the data.

The model residuals

- express the deviation of the observed data points
  - to the model.

The residual's value

- is the vertical distance from a point
  - to the regression line.

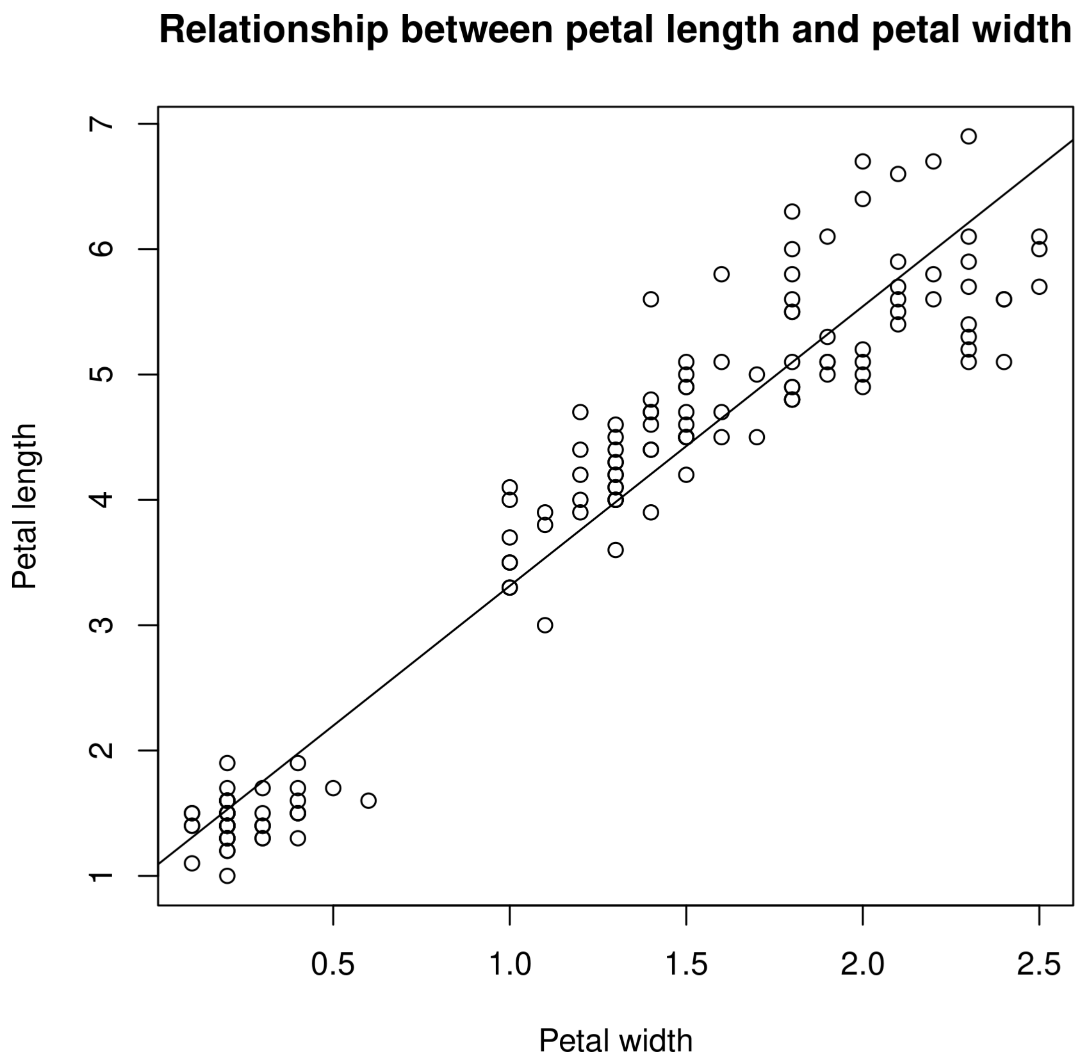
12.2.2.2 Let's examine this with an example of the Fisher's/Anderson's iris dataset.

We have already seen that the dataset contains data about iris flowers.

For the purpose of this example,

- we will consider the petal length as the response
  - sometimes the response is referred to as the "criterion"
- and the petal width as the predictor

```
plot(
  iris$Petal.Length ~ iris$Petal.Width,
  main = "Relationship between petal length and petal width",
  xlab = "Petal width",
  ylab = "Petal length"
)
iris.lm = lm(iris$Petal.Length ~ iris$Petal.Width)
abline(iris.lm)
```




```
SlopeCoef = cor(iris$Petal.Length, iris$Petal.Width) *
  (sd(iris$Petal.Length) / sd(iris$Petal.Width))
SlopeCoef
```

12.2.2.2.1 Computing the intercept and slope coefficient

```
## [1] 2.22994
```

```
coeffs = function(y, x) {
  ((length(y) * sum(y * x)) -
     (sum(y) * sum(x)))  /
    (length(y) * sum(x ^ 2) - sum(x) ^ 2)
}

coeffs(iris$Petal.Length, iris$Petal.Width)
```

```
## [1] 2.22994
```

```

iris.lm
```

12.2.2.2.2 Now make your linear regression function

```
##
## Call:
## lm(formula = iris$Petal.Length ~ iris$Petal.Width)
##
## Coefficients:
##      (Intercept)  iris$Petal.Width
##            1.084             2.230
```

```
regress =  function(y, x) {
  slope = coeffs(y, x)
  intercept = mean(y) - (slope * mean(x))
  model = c(intercept, slope)
  names(model) = c("intercept", "slope")
  model
}
```

```

model = regress(iris$Petal.Length, iris$Petal.Width)
model
```

12.2.2.2.3 Now perform regression on Petal Length and Petal Width

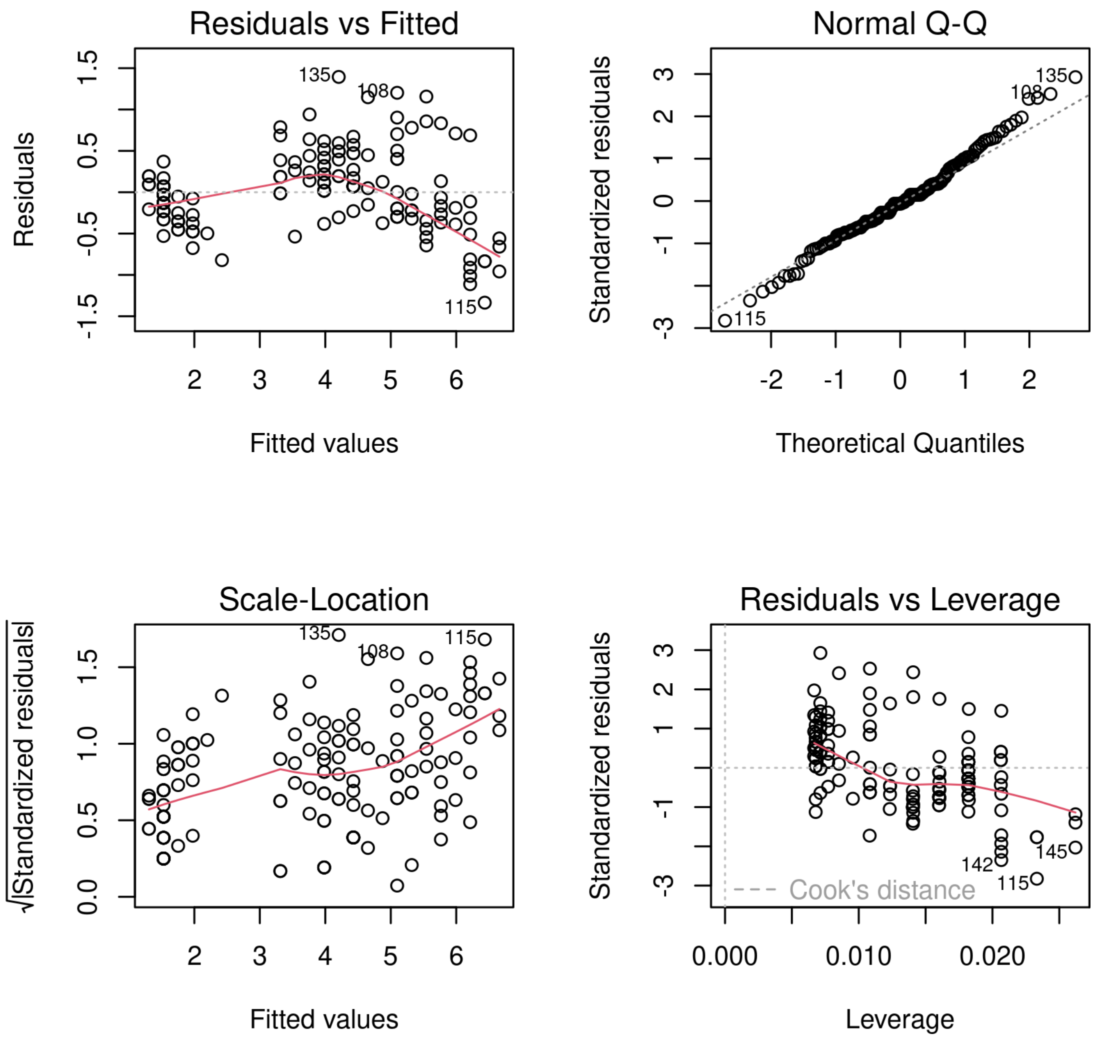
```
## intercept     slope
##  1.083558  2.229940
```

```

resids = function(y, x, model) {
  y - model[1] - (model[2] * x)
}

Residuals = resids(iris$Petal.Length, iris$Petal.Width, model)

head(round(Residuals, 2))
```

12.2.2.2.4 Obtaining the residuals

```
## [1] -0.13 -0.13 -0.23 -0.03 -0.13 -0.28
```

```r
par(mfrow = c(2, 2))
plot(iris.lm)
```



12.2.2.2.5 Computing the significance of the coefficients

This is also the uncertainty

- in your regression coefficients

```r
Significance = function(y, x, model) {
  SSE = sum(resids(y, x, model) ^ 2)
  DF = length(y) - 2
  S = sqrt(SSE / DF)
  SEslope = S / sqrt(sum((x - mean(x)) ^ 2))
  tslope = model[2] / SEslope
  sigslope = 2 * (1 - pt(abs(tslope), DF))
  SEintercept = S * sqrt((1 / length(y) + mean(x) ^ 2 / sum((x - mean(
    x
  )) ^ 2)))
  tintercept = model[1] / SEintercept
  sigintercept = 2 * (1 - pt(abs(tintercept), DF))
  RES = c(SEslope,
```

```
         tslope,
         sigslope,
         SEintercept,
         tintercept,
         sigintercept)
  names(RES) = c("SE slope",
                 "T slope",
                 "sig slope",
                 "SE intercept",
                 "t intercept",
                 "sig intercept")
  RES
}

round(Significance(iris$Petal.Length, iris$Petal.Width, model), 3)
```

```
##      SE slope       T slope     sig slope  SE intercept   t intercept
##         0.051        43.387         0.000         0.073        14.850
## sig intercept
##         0.000
```

```
summary(iris.lm)
```

```
##
## Call:
## lm(formula = iris$Petal.Length ~ iris$Petal.Width)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33542 -0.30347 -0.02955  0.25776  1.39453
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.08356    0.07297   14.85   <2e-16 ***
## iris$Petal.Width  2.22994    0.05140   43.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4782 on 148 degrees of freedom
## Multiple R-squared:  0.9271, Adjusted R-squared:  0.9266
## F-statistic:  1882 on 1 and 148 DF,  p-value: < 2.2e-16
```

12.2.2.3 Links Learning Predictive Analytics with R, Eric Mayor, Packtpub 2015