6

2022-06-12

Multivariable regression

When there is more than one independent variable (*regressors* are many). We assume again homoscedasticity (i.e. noise is normally distributed with mean 0 and variance σ and does not depend on the regressors). β coefficients are again retrieved by minimizing MSE. As for univariable regression, this is accomplished with the `lm()` function too.

```
#both Pepal.Length and Petal.Width are predictors (regressors) of Sepal.Length:
sl_pw <- lm(iris$Sepal.Length ~ iris$Petal.Width)
sl_pl <- lm(iris$Sepal.Length ~ iris$Petal.Length)

sum_pw <- summary(sl_pw)
sum_pw$coefficients
```

```
##                   Estimate Std. Error  t value      Pr(>|t|)
## (Intercept)      4.7776294 0.07293476 65.50552 3.340431e-111
## iris$Petal.Width 0.8885803 0.05137355 17.29645  2.325498e-37
```

```
sum_pl <- summary(sl_pl)
sum_pl$coefficients
```

```
##                    Estimate Std. Error  t value      Pr(>|t|)
## (Intercept)       4.3066034 0.07838896 54.93890 2.426713e-100
## iris$Petal.Length 0.4089223 0.01889134 21.64602  1.038667e-47
```

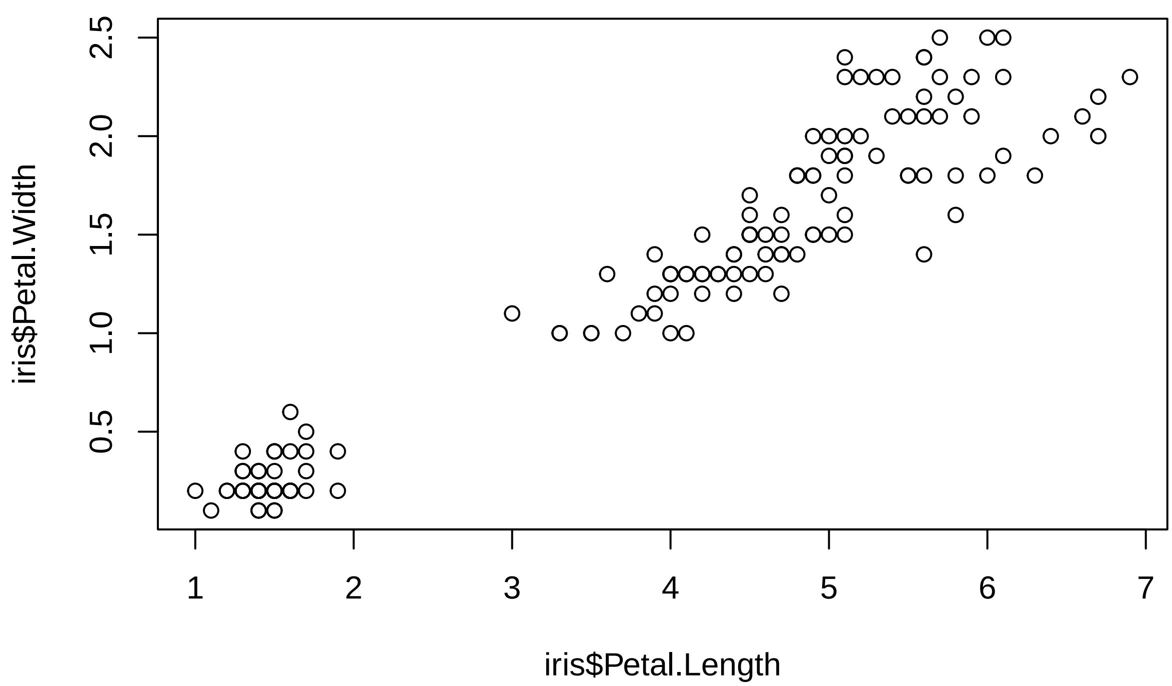
```
#thus it is reasonable to use them together in the regression:
sl_pwpl <- lm(iris$Sepal.Length ~ iris$Petal.Width + iris$Petal.Length)

sum_pwpl <- summary(sl_pwpl)
sum_pwpl$coefficients
```

```
##                     Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)        4.1905824 0.09704587 43.181459 2.092645e-85
## iris$Petal.Width  -0.3195506 0.16045262 -1.991557 4.827246e-02
## iris$Petal.Length  0.5417772 0.06928179  7.819907 9.414477e-13
```

Interpretation: for *fixed Petal.Length* an increase of 1 cm in Petal.Width leads to a decrease by 0.32 cm in Sepal.Length (and for *fixed Petal.Width*, an increase of 1 cm in Petal.Length leads to an increase of 0.54 cm in Sepal.Length). The two predictors/regressors are correlated.

Comparing the coefficient values in the three cases above we can conclude that in the multivariable regression, at fixed Petal.Length, Petal.Width doesn't give additional information on Sepal.Length:

```r
plot(x=iris$Petal.Length, y=iris$Petal.Width)
```



Nested models

These two regression models are nested:

```r
sl_pwpl$call
```

```
## lm(formula = iris$Sepal.Length ~ iris$Petal.Width + iris$Petal.Length)
```

```r
sl_pw$call
```

```
## lm(formula = iris$Sepal.Length ~ iris$Petal.Width)
```

These two models are NOT nested:

```r
sl_pl$call
```

```
## lm(formula = iris$Sepal.Length ~ iris$Petal.Length)
```

```r
sl_pw$call
```

```
## lm(formula = iris$Sepal.Length ~ iris$Petal.Width)
```

Definition: two regression models are called *nested* if the regressors (independent variables) of one model are a subset of the regressors of the other model: in the case above, in `sl_pw` Petal.Width (regressor) is subset of the regressors of `sl_pwpl` (that are Petal.Width indeed and Petal.Length). In this example:

- sl_pwpl = *full* model
- sl_pw = *reduced* model

The full model has lower MSE than the reduced model (because it contains more parameters). That's why we get different β values in the two cases.

The R^2

R^2 is an alternative to MSE to evaluate the quality of a regression model.

- The smaller the MSE, the higher R^2
- $0 < R^2 < 1$
- R^2 is the fraction of variation of Y that can be explained by its linear dependence on the X (that's why we want R^2 to be high)

```r
#We'll retrieve R square of our bi-variable model sl_pwpl:
num <- sum((iris$Sepal.Length - sl_pwpl$fitted.values)^2)
den <- sum((iris$Sepal.Length - mean(iris$Sepal.Length))^2)

r_square <- 1-num/den
r_square
```

```
## [1] 0.7662613
```

In the code above, `sum` is for sommatoria. As always, `iris$Sepal.Length` are the observed values of sepal length, while `sl_pwpl$fitted.values` are those predicted by the bi-variable model.

More rapidly than calculating it, we can do:

```r
sum_pwpl$r.squared
```

```
## [1] 0.7662613
```

ANOVA test for nested models

The full model will display the best MSE but also the best R^2 compared to the reduced model.

The presence of more parameters (the regressors) will improve the R^2, even the addition of random parameters. We need to make a discrimination between addition of a nonsense parameter and the addition of an actual regressor (i.e., is the improvement of R^2 caused by a regressor *significantly greater* than what it would be when adding un unrelated parameter?)

The ANOVA test provides this piece of information by answering to the question above. The test requires the `anova()` function:

```r
an_test <- anova(sl_pwpl, sl_pl) #we could've also used sl_pw instead
an_test
```

```
## Analysis of Variance Table
##
## Model 1: iris$Sepal.Length ~ iris$Petal.Width + iris$Petal.Length
## Model 2: iris$Sepal.Length ~ iris$Petal.Length
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    147 23.881
## 2    148 24.525 -1  -0.64434 3.9663 0.04827 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The P-value you got (0.04827) it's exactly the same associated to Petal.Width in:

```r
sum_pwpl$coefficients
```

```
##                      Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)         4.1905824 0.09704587 43.181459 2.092645e-85
## iris$Petal.Width   -0.3195506 0.16045262 -1.991557 4.827246e-02
## iris$Petal.Length   0.5417772 0.06928179  7.819907 9.414477e-13
```

Just written as 4.827246e-02.

Same for:

```r
anova(sl_pwpl, sl_pw)
```

```
## Analysis of Variance Table
##
## Model 1: iris$Sepal.Length ~ iris$Petal.Width + iris$Petal.Length
## Model 2: iris$Sepal.Length ~ iris$Petal.Width
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    147 23.881
## 2    148 33.815 -1   -9.9342 61.151 9.414e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Where the P-value is identical to that of Petal.Length reported by the `lm()` function.

Summarizing: the P-values reported by the `lm()` function when you fit a multivariable model are those obtained with an ANOVA test comparing the full model (`sl_pwpl`) to the reduced models (`sl_pl` or `sl_pw`), where each regressor has been removed in turn.

Back to univariable models

Let's build the following *reduced* model: $Y = \beta_0 + noise \rightarrow$ Y doesn't depend on X. We fit this model in this way:

```r
sl_int <- lm(iris$Sepal.Length ~  1)
summary(sl_int)$coefficients
```

```
##             Estimate Std. Error  t value      Pr(>|t|)
## (Intercept) 5.843333 0.06761132 86.42537 3.331256e-129
```

5.843 is just the mean of the sepal length values:

```r
mean(iris$Sepal.Length)
```

```
## [1] 5.843333
```

This particular reduced model is called *null* model. Let's run an ANOVA test to compare it with the univariable model using PL:

```r
a_int <- anova(sl_pl, sl_int)
a_int$`Pr(>F)`[2]
```

```
## [1] 1.038667e-47
```

Which is identical to the P-value in the univariable model:

```r
sum_pl$coefficients
```

```
##                   Estimate Std. Error  t value      Pr(>|t|)
## (Intercept)      4.3066034 0.07838896 54.93890 2.426713e-100
## iris$Petal.Length 0.4089223 0.01889134 21.64602  1.038667e-47
```

Let's now use ANOVA to compare the bi-variable model with the null one:

```r
a_biv_int <- anova(sl_pwpl, sl_int)
a_biv_int$`Pr(>F)`[2]
```

```
## [1] 3.996697e-47
```

This output tells that the bivariable model is actually better than the intercept only model in predicting sepal length.

Exercise