Faraway

Suparna

2025-09-12

Multiple linear regression

Now let's look at an example concerning the number of species found on the various Galápagos Islands. There are 30 cases (Islands) and seven variables in the dataset. We start by reading the data into R and examining it:

```
library(faraway)
data(gala)
head(gala[,-2])
```

```
##             Species  Area Elevation Nearest Scruz Adjacent
## Baltra           58 25.09       346     0.6   0.6     1.84
## Bartolome        31  1.24       109     0.6  26.3   572.33
## Caldwell          3  0.21       114     2.8  58.7     0.78
## Champion         25  0.10        46     1.9  47.4     0.18
## Coamano           2  0.05        77     1.9   1.9   903.82
## Daphne.Major     18  0.34       119     8.0   8.0     1.84
```

The variables are Species — the number of species found on the island, Area —the area of the island (km2), Elevation—the highest elevation of the island (m), Nearest — the distance from the nearest island (km), Scruz — the distance from Santa Cruz Island (km), Adjacent—the area of the adjacent island (km2). We have omitted the second column (which has the number of endemic species).

```
lmod <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
data=gala)
S=summary(lmod)
S
```

```
##
## Call:
## lm(formula = Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
##     data = gala)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -111.679  -34.898   -7.862   33.460  182.584
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.068221  19.154198   0.369 0.715351
## Area        -0.023938   0.022422  -1.068 0.296318
## Elevation    0.319465   0.053663   5.953 3.82e-06 ***
## Nearest      0.009144   1.054136   0.009 0.993151
## Scruz       -0.240524   0.215402  -1.117 0.275208
## Adjacent    -0.074805   0.017700  -4.226 0.000297 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.98 on 24 degrees of freedom
## Multiple R-squared:  0.7658, Adjusted R-squared:  0.7171
## F-statistic:  15.7 on 5 and 24 DF,  p-value: 6.838e-07
```

```
sumary(lmod)
```

```
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  7.068221  19.154198  0.3690 0.7153508
## Area        -0.023938   0.022422 -1.0676 0.2963180
## Elevation    0.319465   0.053663  5.9532 3.823e-06
## Nearest      0.009144   1.054136  0.0087 0.9931506
## Scruz       -0.240524   0.215402 -1.1166 0.2752082
## Adjacent    -0.074805   0.017700 -4.2262 0.0002971
##
## n = 30, p = 6, Residual SE = 60.97519, R-Squared = 0.77
```

```
anova(lmod)
```

```
## Analysis of Variance Table
##
## Response: Species
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Area       1 145470  145470 39.1262 1.826e-06 ***
## Elevation  1  65664   65664 17.6613 0.0003155 ***
## Nearest    1     29      29  0.0079 0.9300674
## Scruz      1  14280   14280  3.8408 0.0617324 .
## Adjacent   1  66406   66406 17.8609 0.0002971 ***
## Residuals 24  89231    3718
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can estimate σ using the formula or extract it from the summary object:

```
sqrt(deviance(lmod)/df.residual(lmod))
```

```
## [1] 60.97519
```

```
S$sigma
```

```
## [1] 60.97519
```

Estimate $\hat{\beta}$ without using lm()

First extract the X-matrix

```
x <- model.matrix( ~ Area + Elevation + Nearest + Scruz + Adjacent,
gala)
```

and the response is

```
y <- gala$Species
```

Now let's construct $(X^TX)^{-1}$. t() does transpose and %*% does matrix multiplication. solve(x) computes x^{-1}

```
xtx <- solve(t(x) %*% x)
```

Now we can get $\hat{\beta}$ directly, using $(X^T X)^{-1} y$.

```
xtx %*% t(x) %*% y
```

```
##                     [,1]
## (Intercept)  7.068220709
## Area        -0.023938338
## Elevation    0.319464761
## Nearest      0.009143961
## Scruz       -0.240524230
## Adjacent    -0.074804832
```

This is a very bad way to compute $\hat{\beta}$. It is inefficient and can be very inaccurate when the predictors are strongly correlated. Instead we can use

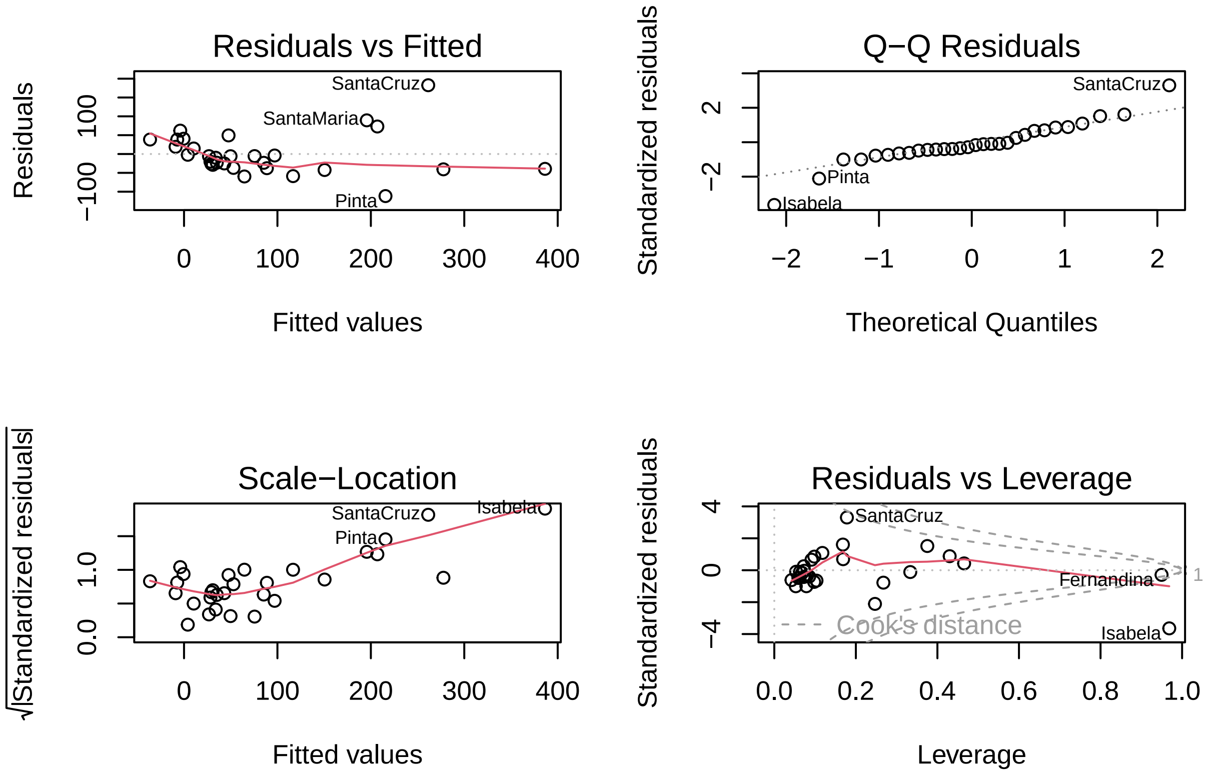
```
solve(crossprod(x,x),crossprod(x,y))
```

```
##                     [,1]
## (Intercept)  7.068220709
## Area        -0.023938338
## Elevation    0.319464761
## Nearest      0.009143961
## Scruz       -0.240524230
## Adjacent    -0.074804832
```

Next diagnostic plots of residuals for the multiple linear regression

```
par(mfrow=c(2,2))
plot(lmod)
```

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```



Pairwise correlation between predictors.

```
cor(gala[sapply(gala, is.numeric)])
```

```
##                Species     Endemics       Area   Elevation      Nearest
## Species     1.00000000  0.970876516  0.6178431  0.73848666 -0.014094067
## Endemics    0.97087652  1.000000000  0.6169791  0.79290437  0.005994286
## Area        0.61784307  0.616979087  1.0000000  0.75373492 -0.111103196
## Elevation   0.73848666  0.792904369  0.7537349  1.00000000 -0.011076984
## Nearest    -0.01409407  0.005994286 -0.1111032 -0.01107698  1.000000000
## Scruz      -0.17114244 -0.154264319 -0.1007849 -0.01543829  0.615410357
## Adjacent    0.02616635  0.082658026  0.1800376  0.53645782 -0.116247885
##                  Scruz    Adjacent
## Species    -0.17114244  0.02616635
## Endemics   -0.15426432  0.08265803
## Area       -0.10078493  0.18003759
## Elevation  -0.01543829  0.53645782
## Nearest     0.61541036 -0.11624788
## Scruz       1.00000000  0.05166066
## Adjacent    0.05166066  1.00000000
```

There are few large pairwise correlations between predictors. Now we check the eigendecomposition of $X^T X$ (not including the intercept in X):

```
x <- model.matrix(lmod)[,-1]
e <- eigen(t(x) %*% x)
e$val
```

```
## [1] 36598003.629 17873972.614  2243824.284   167925.744     3293.719
```

```
sqrt(e$val[1]/e$val)
```

```
## [1]   1.000000   1.430929   4.038632  14.762845 105.410881
```