Linear Regression 1

Data Analytics and Visualization

Instructor: Debopriya Ghosh

Estimation

DATASET: This dataset captures the number of species of tortoise on the various Galapagos Islands. There are 30 cases (Islands) and 7 variables in the dataset.

The variables are:

- Species The number of species of tortoise found on the island
- Endemics The number of endemic species
- Area The area of the island (km^2)
- 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 (km^2)

```
library(faraway)
data(gala)
summary(gala)
```

```
##     Species          Endemics         Area            Elevation
##  Min.   :  2.00   Min.   : 0.00   Min.   :   0.010   Min.   :  25.00
##  1st Qu.: 13.00   1st Qu.: 7.25   1st Qu.:   0.258   1st Qu.:  97.75
##  Median : 42.00   Median :18.00   Median :   2.590   Median : 192.00
##  Mean   : 85.23   Mean   :26.10   Mean   : 261.709   Mean   : 368.03
##  3rd Qu.: 96.00   3rd Qu.:32.25   3rd Qu.:  59.237   3rd Qu.: 435.25
##  Max.   :444.00   Max.   :95.00   Max.   :4669.320   Max.   :1707.00
##     Nearest          Scruz           Adjacent
##  Min.   : 0.20   Min.   :  0.00   Min.   :   0.03
##  1st Qu.: 0.80   1st Qu.: 11.03   1st Qu.:   0.52
##  Median : 3.05   Median : 46.65   Median :   2.59
##  Mean   :10.06   Mean   : 56.98   Mean   : 261.10
##  3rd Qu.:10.03   3rd Qu.: 81.08   3rd Qu.:  59.24
##  Max.   :47.40   Max.   :290.20   Max.   :4669.32
```

Fitting a linear model in R is done using the lm() command. The syntax for specifying the predictors in the model is called Wilkinson-Rogers notation.

```
lm.fit = lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent, data = gala)
summary(lm.fit)
```

```
##
## 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
```

We can identify several useful quantities in this output.

Lets define the matrix X, and response variable y.

```r
X = cbind(1, gala[,-c(1,2)])
X = as.matrix(X)
y = gala$Species
```

Now let's compute X^TX.

```r
t(X)%*%X
```

```
##                  1          Area  Elevation   Nearest     Scruz    Adjacent
## 1            30.00       7851.26    11041.0    301.80   1709.30     7832.95
## Area       7851.26   23708665.46 10852798.5  39240.84 275516.84  5950313.65
## Elevation 11041.00   10852798.53  9218227.0 109139.20 616237.80  8553187.95
## Nearest     301.80      39240.84   109139.2   8945.30  34527.34    37196.67
## Scruz      1709.30     275516.84   616237.8  34527.34 231613.77   534409.98
## Adjacent   7832.95    5950313.65  8553187.9  37196.67 534409.98 23719568.46
```

Next, compute $(X^TX)^-1$. Inverses can be taken using the solve() command:

```r
xtxi = solve(t(X) %*% X)
xtxi
```

```
##                       1          Area     Elevation       Nearest
## 1          9.867829e-02  3.778242e-05 -1.561976e-04 -2.339027e-04
## Area       3.778242e-05  1.352247e-07 -2.593617e-07  1.294003e-06
## Elevation -1.561976e-04 -2.593617e-07  7.745339e-07 -3.549366e-06
## Nearest   -2.339027e-04  1.294003e-06 -3.549366e-06  2.988732e-04
## Scruz     -3.760293e-04 -4.913149e-08  3.080831e-07 -3.821077e-05
## Adjacent   2.309832e-05  4.620303e-08 -1.640241e-07  1.424729e-06
##                   Scruz      Adjacent
## 1         -3.760293e-04  2.309832e-05
## Area      -4.913149e-08  4.620303e-08
## Elevation  3.080831e-07 -1.640241e-07
## Nearest   -3.821077e-05  1.424729e-06
## Scruz      1.247941e-05 -1.958356e-07
## Adjacent  -1.958356e-07  8.426543e-08
```

A more direct way of computing $(X^TX)^-1$ is:

```
lm.fit = lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent, data = gala)
gs = summary(lm.fit)
gs$cov.unscaled
```

```
##               (Intercept)          Area     Elevation       Nearest
## (Intercept)  9.867829e-02  3.778242e-05 -1.561976e-04 -2.339027e-04
## Area         3.778242e-05  1.352247e-07 -2.593617e-07  1.294003e-06
## Elevation   -1.561976e-04 -2.593617e-07  7.745339e-07 -3.549366e-06
## Nearest     -2.339027e-04  1.294003e-06 -3.549366e-06  2.988732e-04
## Scruz       -3.760293e-04 -4.913149e-08  3.080831e-07 -3.821077e-05
## Adjacent     2.309832e-05  4.620303e-08 -1.640241e-07  1.424729e-06
##                     Scruz      Adjacent
## (Intercept) -3.760293e-04  2.309832e-05
## Area        -4.913149e-08  4.620303e-08
## Elevation    3.080831e-07 -1.640241e-07
## Nearest     -3.821077e-05  1.424729e-06
## Scruz        1.247941e-05 -1.958356e-07
## Adjacent    -1.958356e-07  8.426543e-08
```

The names() command is the way to see the components of an R object.

```
names(gs)
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```
names(lm.fit)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

The ???tted (or predicted) values and residuals are:

```
lm.fit$fitted.values
```

```
##       Baltra    Bartolome     Caldwell     Champion      Coamano
##  116.7259460   -7.2731544   29.3306594   10.3642660  -36.3839155
## Daphne.Major Daphne.Minor       Darwin         Eden      Enderby
##   43.0877052   33.9196678   -9.0189919   28.3142017   30.7859425
##     Espanola   Fernandina     Gardner1     Gardner2     Genovesa
##   47.6564865   96.9895982   -4.0332759   64.6337956   -0.4971756
##      Isabela     Marchena       Onslow        Pinta       Pinzon
##  386.4035578   88.6945404    4.0372328  215.6794862  150.4753750
##   Las.Plazas       Rabida SanCristobal  SanSalvador    SantaCruz
##   35.0758066   75.5531221  206.9518779  277.6763183  261.4164131
##      SantaFe   SantaMaria      Seymour      Tortuga         Wolf
##   85.3764857  195.6166286   49.8050946   52.9357316   26.7005735
```

```
lm.fit$residuals
```

```
##       Baltra    Bartolome     Caldwell     Champion      Coamano
##   -58.725946    38.273154   -26.330659    14.635734    38.383916
## Daphne.Major Daphne.Minor       Darwin         Eden      Enderby
##   -25.087705    -9.919668    19.018992   -20.314202   -28.785943
##     Espanola   Fernandina     Gardner1     Gardner2     Genovesa
##    49.343513    -3.989598    62.033276   -59.633796    40.497176
```

```
##      Isabela     Marchena       Onslow        Pinta       Pinzon
##   -39.403558   -37.694540    -2.037233  -111.679486   -42.475375
##   Las.Plazas       Rabida SanCristobal  SanSalvador    SantaCruz
##   -23.075807    -5.553122    73.048122   -40.676318   182.583587
##      SantaFe   SantaMaria      Seymour      Tortuga         Wolf
##   -23.376486    89.383371    -5.805095   -36.935732    -5.700573
```

We can get $\hat{\beta}$ directly:

```
solve(t(X) %*% X, t(X) %*% y)
```

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

We can estimate σ using:

```
sqrt(sum(lm.fit$residuals^2)/(30-6))
```

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

We also obtain the standard errors for the coef???cients.

```
sqrt(diag(xtxi))*60.975
```

```
##           1         Area   Elevation     Nearest       Scruz    Adjacent
## 19.15413865  0.02242228  0.05366264  1.05413269  0.21540158  0.01770013
```

Finally we may compute R^2.

```
1-sum(lm.fit$residuals^2)/sum((y-mean(y))^2)
```

```
## [1] 0.7658469
```

Compare these to the results above.