MATH 4070 R Session 3: Multiple Linear Regression II

Whitney

Contents

Species diversity on the Galapagos Islands

Load the data

```
library(faraway)
data(gala)
galaNew <- gala[, -2] # removing "Endemics"
```

General Linear F-Test

```
## First example
# Reduce Model
M1 <- lm(Species ~ Elevation, data = galaNew)
summary(M1)
```

```
##
## Call:
## lm(formula = Species ~ Elevation, data = galaNew)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -218.319  -30.721  -14.690    4.634  259.180
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.33511   19.20529   0.590     0.56
## Elevation    0.20079    0.03465   5.795 3.18e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 78.66 on 28 degrees of freedom
## Multiple R-squared:  0.5454, Adjusted R-squared:  0.5291
## F-statistic: 33.59 on 1 and 28 DF,  p-value: 3.177e-06
```

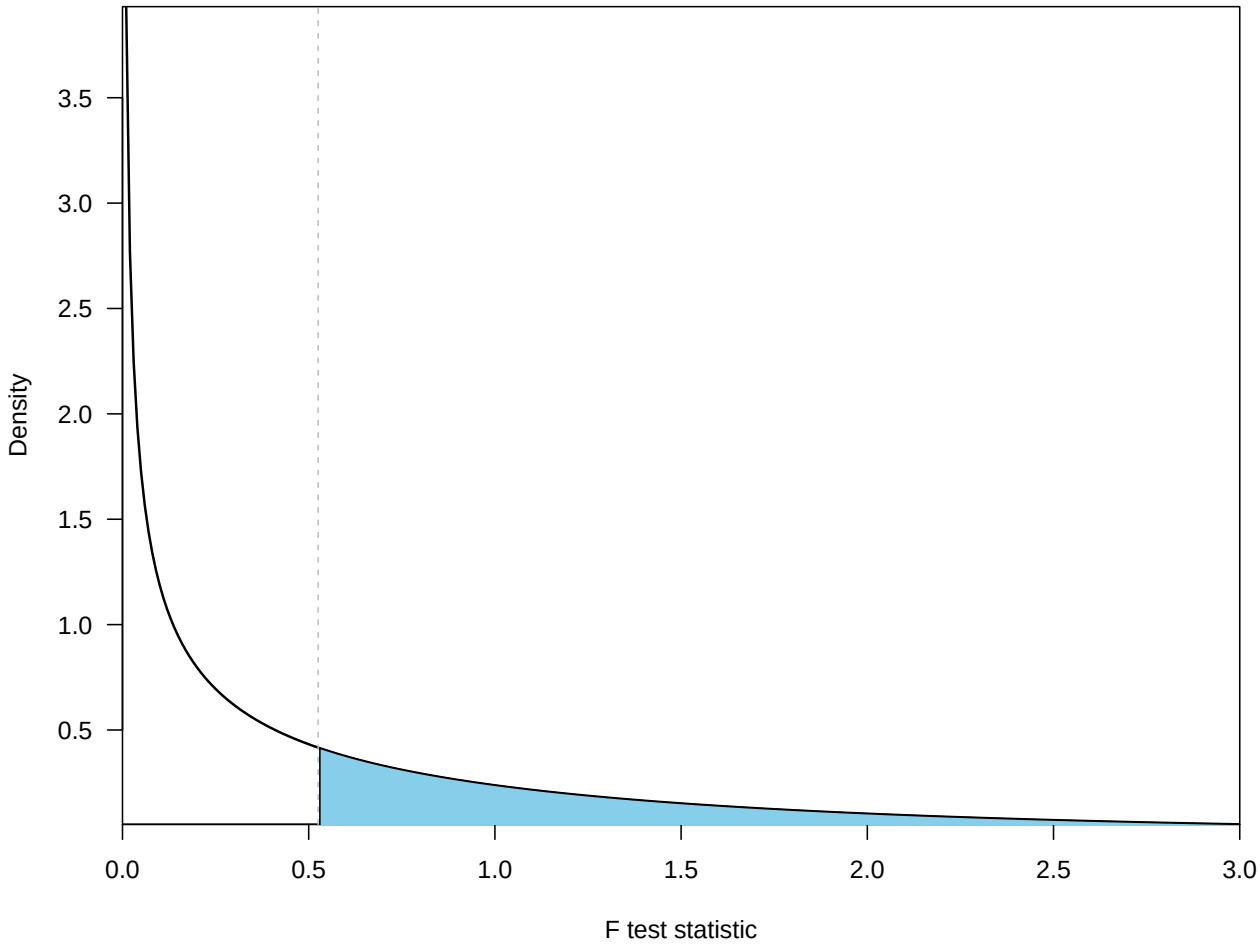
```
# "Full" Model
M2 <- lm(Species ~ Elevation + Area, data = galaNew)
summary(M2)
```

```
##
## Call:
## lm(formula = Species ~ Elevation + Area, data = galaNew)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -192.619  -33.534  -19.199    7.541  261.514
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.10519   20.94211   0.817  0.42120
## Elevation    0.17174    0.05317   3.230  0.00325 **
## Area         0.01880    0.02594   0.725  0.47478
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79.34 on 27 degrees of freedom
## Multiple R-squared:  0.554,  Adjusted R-squared:  0.521
## F-statistic: 16.77 on 2 and 27 DF,  p-value: 1.843e-05
```

```r
## General Linear F-Test
anova(M1, M2)
```

```
## Analysis of Variance Table
##
## Model 1: Species ~ Elevation
## Model 2: Species ~ Elevation + Area
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     28 173254
## 2     27 169947  1      3307 0.5254 0.4748
```

```r
# p-value
par(las = 1, mar = c(4.1, 4.1, 1.1, 1.1))
xg <- seq(0, 3, 0.01); yg <- df(xg, 1, 27)
plot(xg, yg, type = "l", xaxs = "i", yaxs = "i", lwd = 1.6,
     xlab = "F test statistic", ylab = "Density")
abline(v = 0.5254, lty = 2, col = "gray")
polygon(c(xg[xg > 0.5254], rev(xg[xg > 0.5254])),
        c(yg[xg > 0.5254], rep(0, length(yg[xg > 0.5254]))),
        col = "skyblue")
```



```r
# Another example
Full <- lm(Species ~ ., data = galaNew)
Reduce <- lm(Species ~ Elevation + Adjacent, data = galaNew)
## General Linear F-Test
anova(Reduce, Full)
```

```
## Analysis of Variance Table
##
## Model 1: Species ~ Elevation + Adjacent
## Model 2: Species ~ Area + Elevation + Nearest + Scruz + Adjacent
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     27 100003
## 2     24  89231  3     10772 0.9657  0.425
```

Prediction

First, fit a linear regression model:

```r
data(fat)
lmod <- lm(brozek ~ age + weight + height + neck + chest + abdom + hip + thigh
           + knee + ankle + biceps + forearm + wrist, data = fat)
```

Extract the design matrix X then calculate the median for each predictor:

```r
## Design matrix
X <- model.matrix(lmod)
(x0 <- apply(X, 2, median))
```

```
## (Intercept)         age      weight      height        neck       chest
##        1.00       43.00      176.50       70.00       38.00       99.65
##       abdom         hip       thigh        knee       ankle      biceps
##       90.95       99.30       59.00       38.50       22.80       32.05
##     forearm       wrist
##       28.70       18.30
```

Compute the prediction and use the *predict* command to obtain prediction uncertainty for a future observation and the mean response:

```r
(y0 <- sum(x0 * coef(lmod)))
```

```
## [1] 17.49322
```

```r
predict(lmod, new = data.frame(t(x0)))
```

```
##        1
## 17.49322
```

```r
predict(lmod, new = data.frame(t(x0)), interval = "prediction")
```

```
##        fit     lwr      upr
## 1 17.49322 9.61783 25.36861
```

```r
predict(lmod, new = data.frame(t(x0)), interval = "confidence")
```

```
##        fit      lwr      upr
## 1 17.49322 16.94426 18.04219
```

Multicollinearity

Here, we conduct a Monte Carlo simulation to demonstrate the effects of multicollinearity. Let the true linear model be:

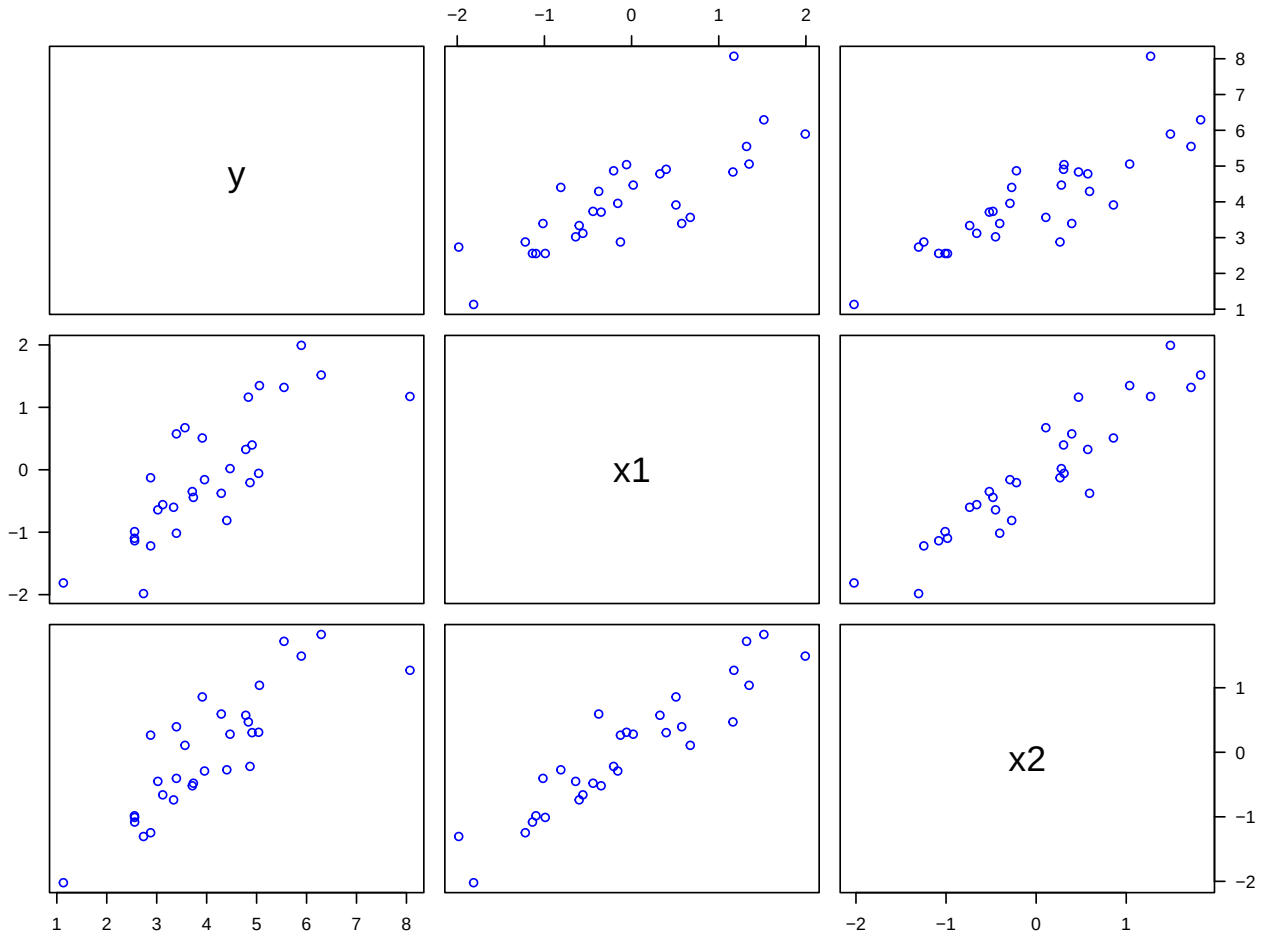
$$y = 4 + 0.8x_1 + 0.6x_2 + \epsilon,$$

where $\epsilon \overset{i.i.d}{\sim} \mathrm{N}(0, 1)$, and x_1 and x_2 are highly linearly correlated with $\rho = 0.9$. The Monte Carlo experiment is repeated 500 times.

```r
set.seed(123)
N = 500
library(MASS)
x <- replicate(N, mvrnorm(n = 30, c(0, 0), matrix(c(1, 0.9, 0.9, 1), 2)))
y <- array(dim = c(30, N))
for (i in 1:N){
  y[, i] = 4 + 0.8 * x[, 1, i] + 0.6 * x[, 2, i] + rnorm(30)
}
```

Let's take a look at the first simulated data:

```r
# Grab the first simulated data
sim1 <- data.frame(y = y[, 1], x1 = x[, 1, 1], x2 = x[, 2, 1])
# Make the scatterplot matrix
pairs(sim1, las = 1, col = "blue")
```




```
# Compute the correlation matrix
cor(sim1)
```

```
##            y        x1        x2
## y  1.0000000 0.7987777 0.8481084
## x1 0.7987777 1.0000000 0.9281514
## x2 0.8481084 0.9281514 1.0000000
```

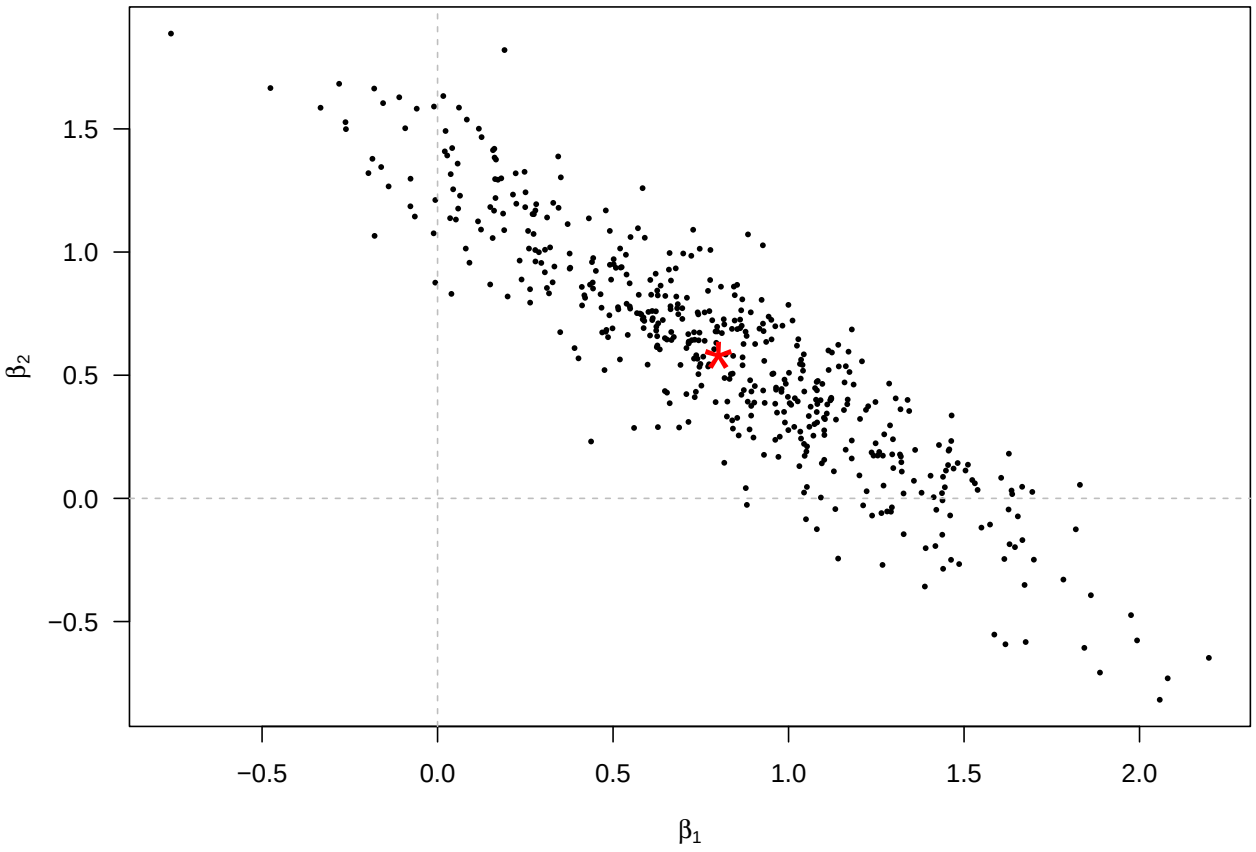
```
vif(sim1[, 2:3])
```

```
##       x1       x2
## 7.218394 7.218394
```

Examine the fitted regression coefficients under collinearity:

```
# Save the fitted regression coefficients
beta <- array(dim = c(3, N))
for (i in 1:N){
  beta[, i] <- lm(y[, i] ~ x[, 1, i] + x[, 2, i])$coefficients
}

plot(beta[2,], beta[3,], pch = 16, cex = 0.5,
```

```
     xlab = expression(beta[1]),
     ylab = expression(beta[2]), las = 1)
points(0.8, 0.6, pch = "*", cex = 3, col = "red")
abline(h = 0, lty = 2, col = "gray")
abline(v = 0, lty = 2, col = "gray")
```



Examine the regression fits under collinearity:

```
R.sq_M1 <- numeric(N)
for (i in 1:N){
  R.sq_M1[i] <- summary(lm(y[, i] ~ x[, 1, i] + x[, 2, i]))$r.squared
}

summary(R.sq_M1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3099  0.6049  0.6776  0.6630  0.7343  0.9016
```

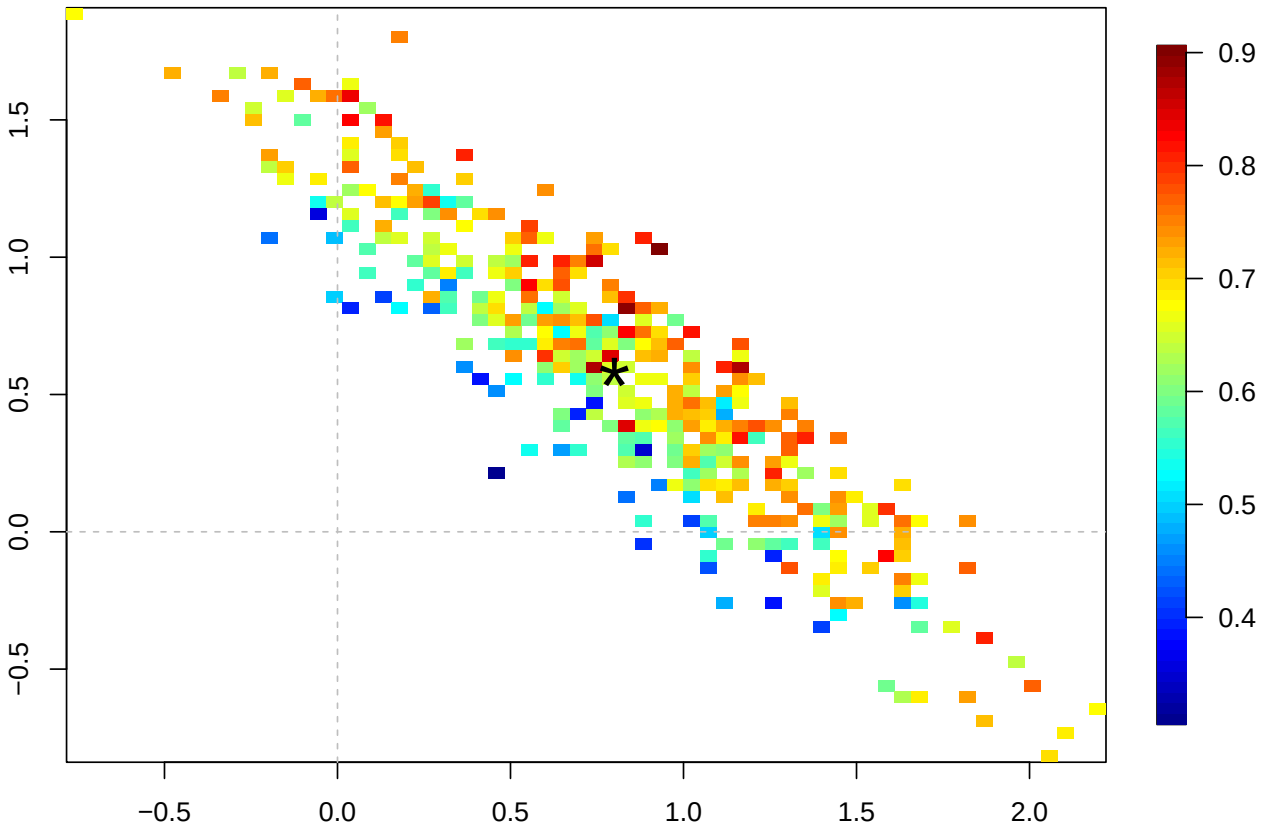
```
library(fields)
```

```
## Loading required package: spam
```

```
## Spam version 2.10-0 (2023-10-23) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
```

```
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridisLite

##
## Try help(fields) to get started.
```

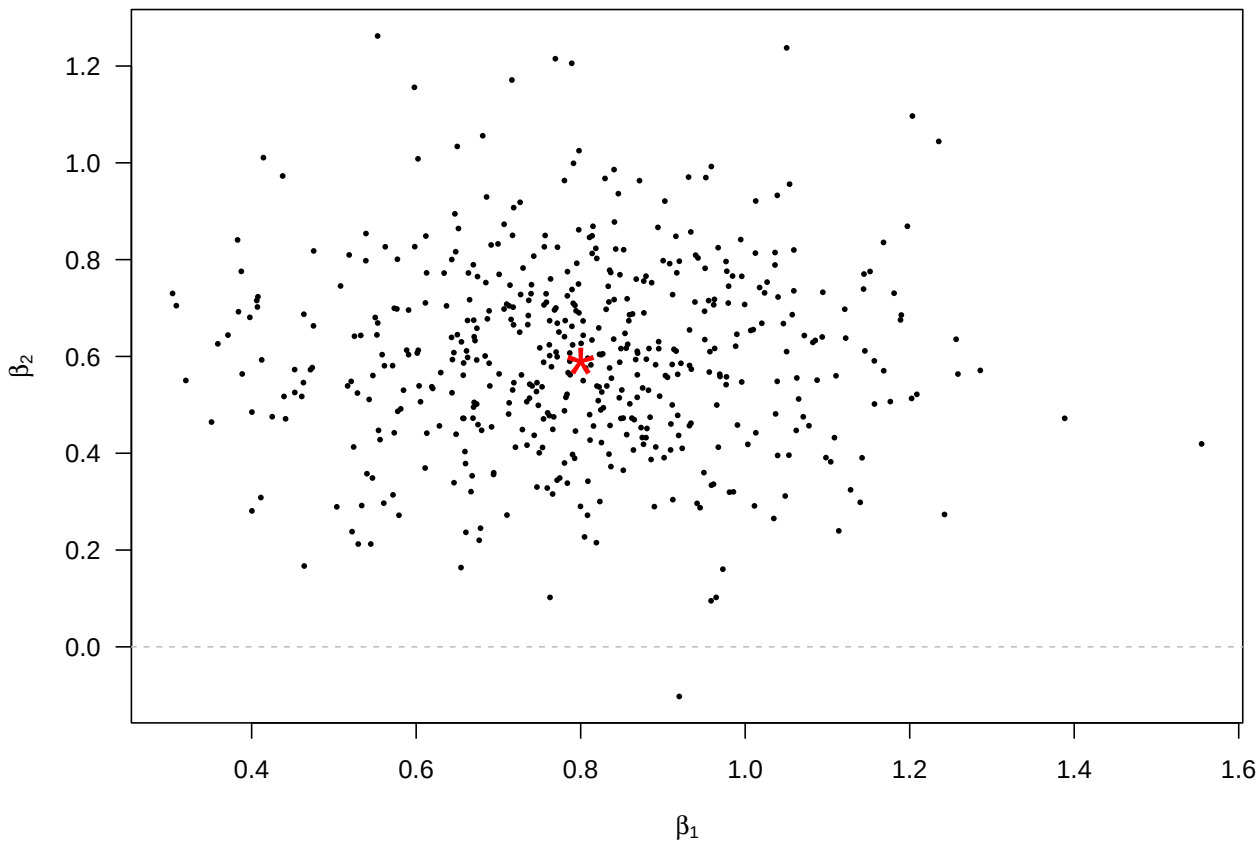
```
quilt.plot(beta[2,], beta[3, ], R.sq_M1)
points(0.8, 0.6, pch = "*", cex = 3)
abline(h = 0, lty = 2, col = "gray")
abline(v = 0, lty = 2, col = "gray")
```



Let's conduct another experiment where the predictors are independent of each other to contrast with the previous experiment and examine the effects due to multicollinearity.

```
x1 <- replicate(N, mvrnorm(n = 30, c(0, 0), matrix(c(1, 0, 0, 1), 2)))
y1 <- array(dim = c(30, N))
for (i in 1:N){
```

```
  y1[, i] = 4 + 0.8 * x1[, 1, i] + 0.6 * x1[, 2, i] + rnorm(30)
}
beta1 <- array(dim = c(3, N))
for (i in 1:N){
  beta1[, i] <- lm(y1[, i] ~ x1[, 1, i] + x1[, 2, i])$coefficients
}

plot(beta1[2,], beta1[3,], pch = 16, cex = 0.5,
     xlab = expression(beta[1]),
     ylab = expression(beta[2]), las = 1)
points(0.8, 0.6, pch = "*", cex = 3, col = "red")
abline(h = 0, lty = 2, col = "gray")
abline(v = 0, lty = 2, col = "gray")
```




```
R.sq_M2 <- numeric(N)
for (i in 1:N){
  R.sq_M2[i] <- summary(lm(y1[, i] ~ x1[, 1, i] + x1[, 2, i]))$r.squared
}
summary(R.sq_M2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1179  0.4375  0.5325  0.5181  0.6062  0.8419
```

```
# Compute the VIF
vif(x1[, 1:2, 1])
```

```
## [1] 1.042404 1.042404
```

Model Selection

All Subset Selection

```
library(leaps)
models <- regsubsets(Species ~ ., data = galaNew)
summary(models)
```

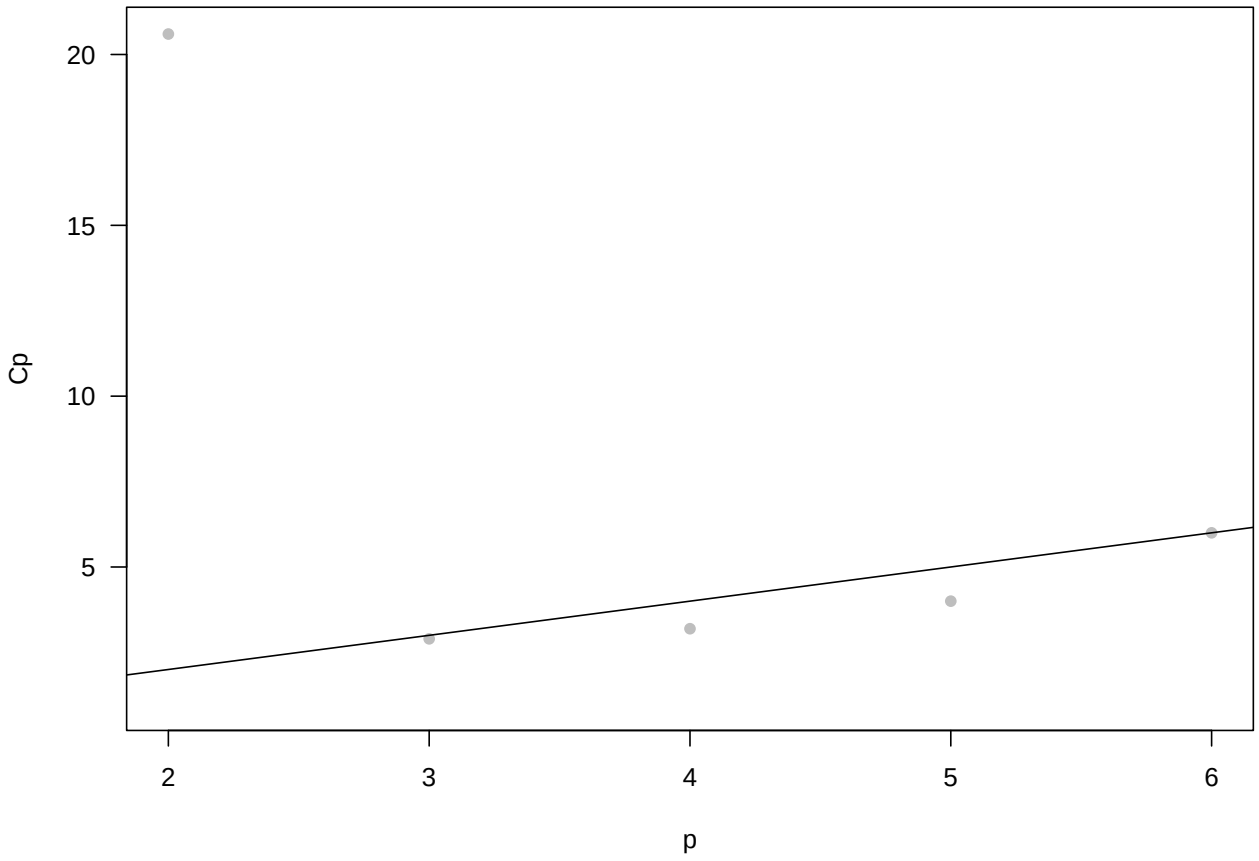
```
## Subset selection object
## Call: regsubsets.formula(Species ~ ., data = galaNew)
## 5 Variables  (and intercept)
##           Forced in Forced out
## Area          FALSE      FALSE
## Elevation     FALSE      FALSE
## Nearest       FALSE      FALSE
## Scruz         FALSE      FALSE
## Adjacent      FALSE      FALSE
## 1 subsets of each size up to 5
## Selection Algorithm: exhaustive
##          Area Elevation Nearest Scruz Adjacent
## 1  ( 1 ) " "  "*"       " "     " "   " "
## 2  ( 1 ) " "  "*"       " "     " "   "*"
## 3  ( 1 ) " "  "*"       " "     "*"   "*"
## 4  ( 1 ) "*"  "*"       " "     "*"   "*"
## 5  ( 1 ) "*"  "*"       "*"     "*"   "*"
```

Reporting model selection criteria

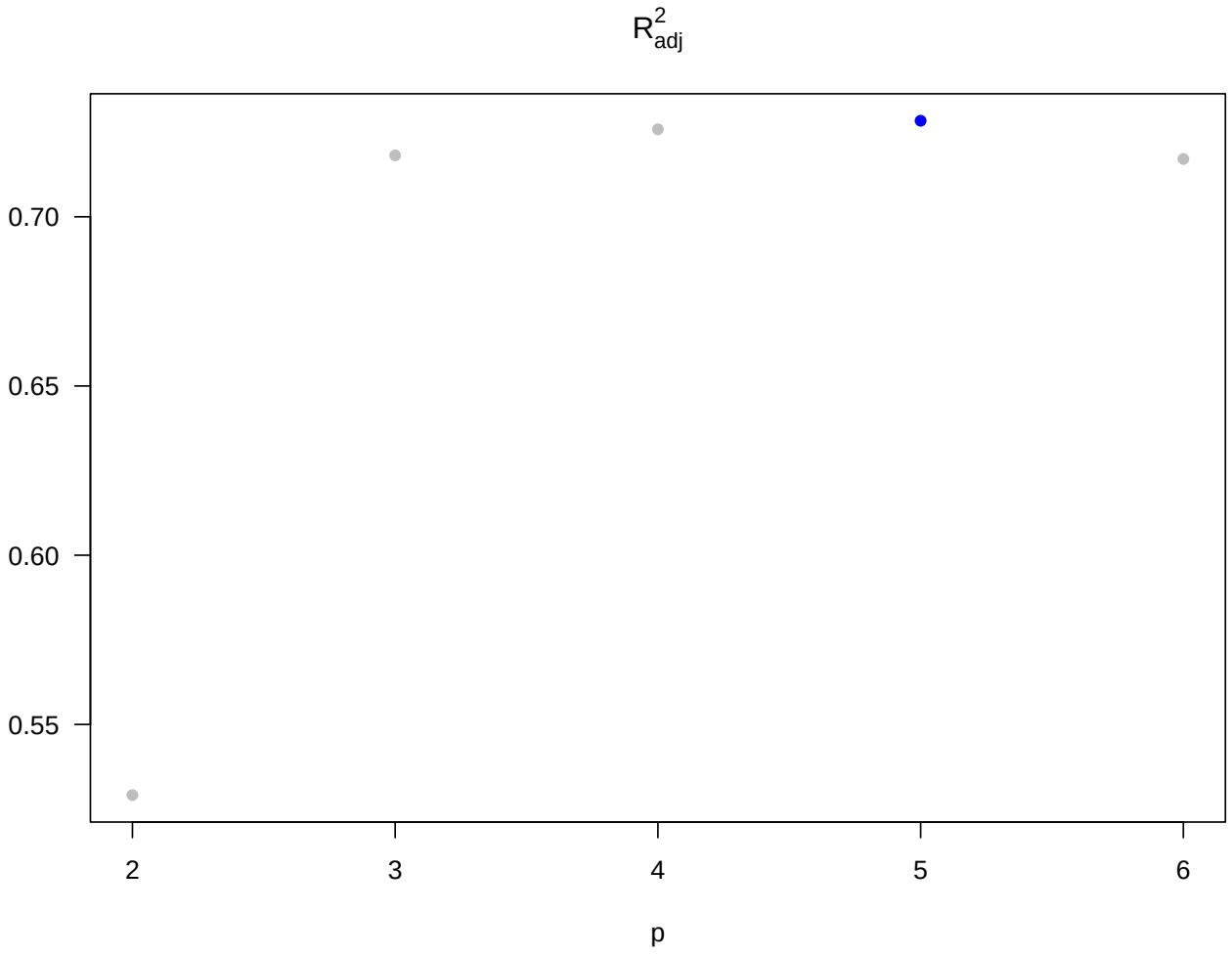
```
res.sum <- summary(models)
criteria <- data.frame(Adj.R2 = res.sum$adjr2,
  Cp = res.sum$cp, BIC = res.sum$bic)

criteria
```

```
##      Adj.R2        Cp       BIC
## 1 0.5291255 20.599003 -16.84525
## 2 0.7181425  2.897184 -29.93078
## 3 0.7258462  3.193068 -28.49317
## 4 0.7283816  4.000075 -26.54733
## 5 0.7170651  6.000000 -23.14622
```

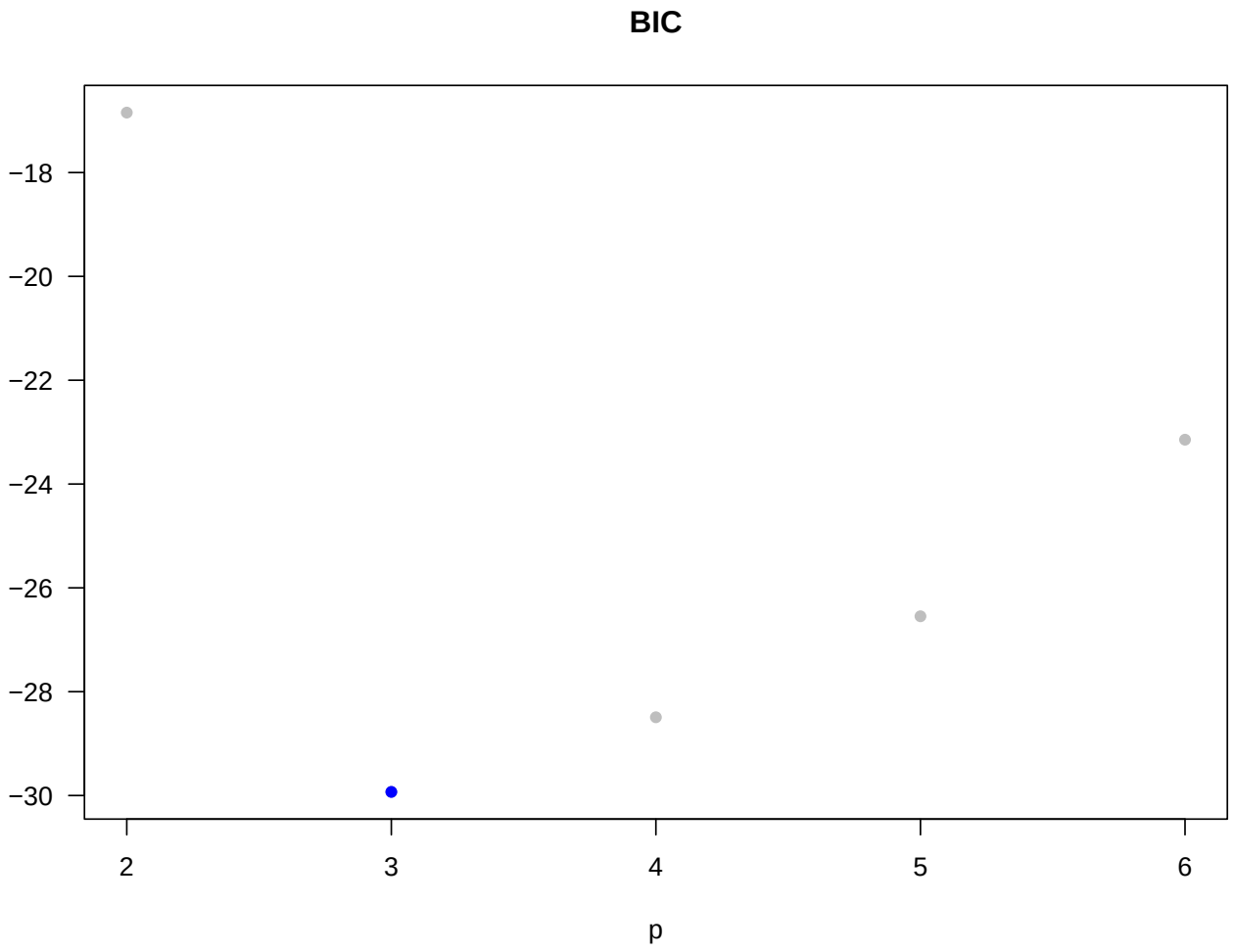
```
plot(2:6, criteria$Cp, las = 1, xlab = "p", ylab = "Cp",
     pch = 16, col = "gray", ylim = c(1, max(criteria$Cp)))
abline(0, 1)
```




```
plot(2:6, criteria$Adj.R2, las = 1, xlab = "p", ylab = "", pch = 16, col = "gray",
     main = expression(R['adj']^2))
points(5, criteria$Adj.R2[4], col = "blue", pch = 16)
```




```
plot(2:6, criteria$BIC, las = 1, xlab = "p", ylab = "", pch = 16, col = "gray", main = "BIC")
points(3, criteria$BIC[2], col = "blue", pch = 16)
```



Backward Selection

Starts with all the predictors and then removes predictors one by one using some criterion

```
full <- lm(Species ~ ., data = galaNew)
step(full, direction = "backward")
```

```
## Start:  AIC=251.93
## Species ~ Area + Elevation + Nearest + Scruz + Adjacent
##
##             Df Sum of Sq    RSS    AIC
## - Nearest    1         0  89232 249.93
## - Area       1      4238  93469 251.33
## - Scruz      1      4636  93867 251.45
## <none>                    89231 251.93
## - Adjacent   1     66406 155638 266.62
## - Elevation  1    131767 220998 277.14
##
## Step:  AIC=249.93
## Species ~ Area + Elevation + Scruz + Adjacent
##
##             Df Sum of Sq    RSS    AIC
```

```
## - Area       1      4436  93667 249.39
## <none>                    89232 249.93
## - Scruz      1      7544  96776 250.37
## - Adjacent   1     72312 161544 265.74
## - Elevation  1    139445 228677 276.17
##
## Step:  AIC=249.39
## Species ~ Elevation + Scruz + Adjacent
##
##             Df Sum of Sq    RSS    AIC
## - Scruz      1      6336 100003 249.35
## <none>                    93667 249.39
## - Adjacent   1     69860 163527 264.11
## - Elevation  1    275784 369451 288.56
##
## Step:  AIC=249.35
## Species ~ Elevation + Adjacent
##
##             Df Sum of Sq    RSS    AIC
## <none>                   100003 249.35
## - Adjacent   1     73251 173254 263.84
## - Elevation  1    280817 380820 287.47

##
## Call:
## lm(formula = Species ~ Elevation + Adjacent, data = galaNew)
##
## Coefficients:
## (Intercept)    Elevation     Adjacent
##     1.43287      0.27657     -0.06889
```

Stepwise Selection

A combination of backward elimination and forward selection can involve adding or deleting predictors at each stage

```
step(full, direction = "both")
```

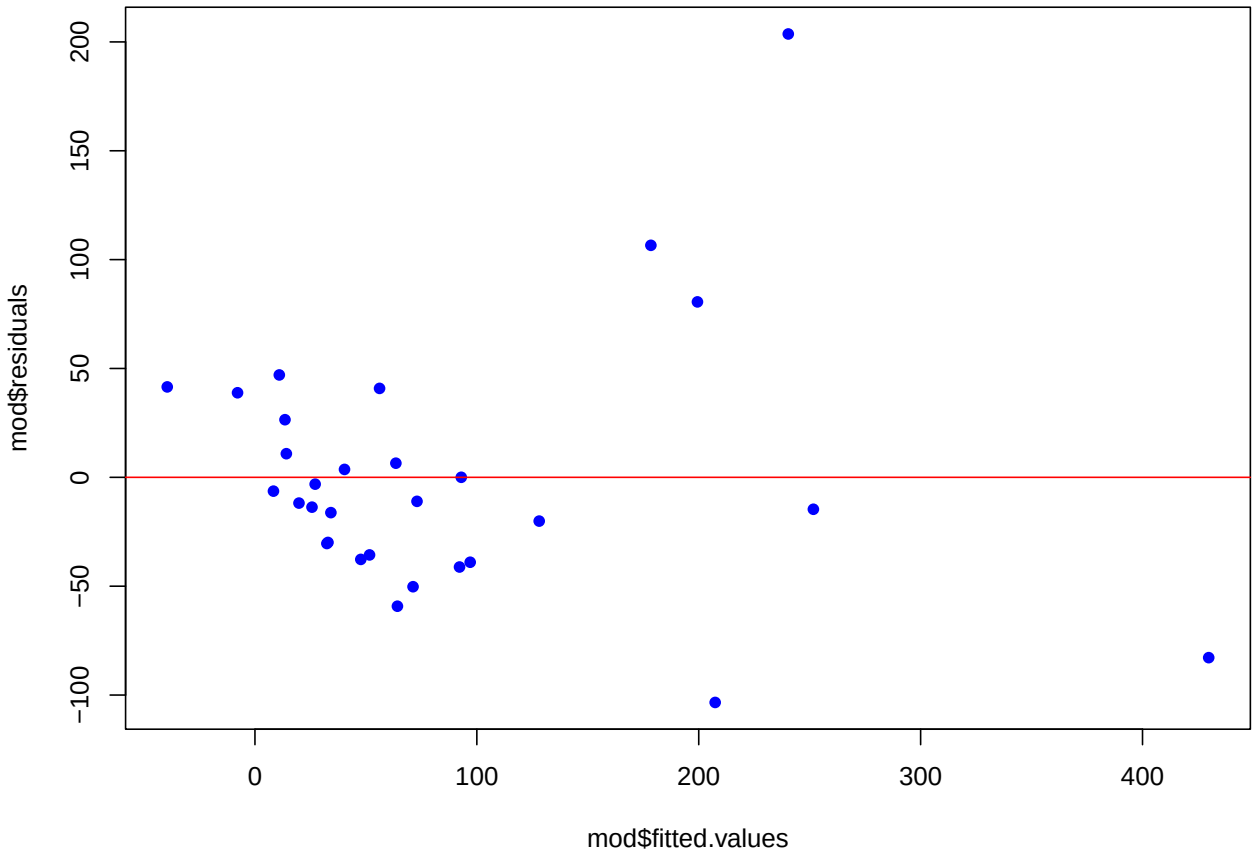
```
## Start:  AIC=251.93
## Species ~ Area + Elevation + Nearest + Scruz + Adjacent
##
##             Df Sum of Sq    RSS    AIC
## - Nearest    1         0  89232 249.93
## - Area       1      4238  93469 251.33
## - Scruz      1      4636  93867 251.45
## <none>                    89231 251.93
## - Adjacent   1     66406 155638 266.62
## - Elevation  1    131767 220998 277.14
##
## Step:  AIC=249.93
## Species ~ Area + Elevation + Scruz + Adjacent
##
##             Df Sum of Sq    RSS    AIC
```

```
## - Area       1      4436  93667 249.39
## <none>                    89232 249.93
## - Scruz      1      7544  96776 250.37
## + Nearest    1         0  89231 251.93
## - Adjacent   1     72312 161544 265.74
## - Elevation  1    139445 228677 276.17
##
## Step:  AIC=249.39
## Species ~ Elevation + Scruz + Adjacent
##
##             Df Sum of Sq    RSS    AIC
## - Scruz      1      6336 100003 249.35
## <none>                    93667 249.39
## + Area       1      4436  89232 249.93
## + Nearest    1       198  93469 251.33
## - Adjacent   1     69860 163527 264.11
## - Elevation  1    275784 369451 288.56
##
## Step:  AIC=249.35
## Species ~ Elevation + Adjacent
##
##             Df Sum of Sq    RSS    AIC
## <none>                   100003 249.35
## + Scruz      1      6336  93667 249.39
## + Area       1      3227  96776 250.37
## + Nearest    1      1550  98453 250.88
## - Adjacent   1     73251 173254 263.84
## - Elevation  1    280817 380820 287.47

##
## Call:
## lm(formula = Species ~ Elevation + Adjacent, data = galaNew)
##
## Coefficients:
## (Intercept)    Elevation     Adjacent
##     1.43287      0.27657     -0.06889
```

Model Diagnostics

Residual Plot

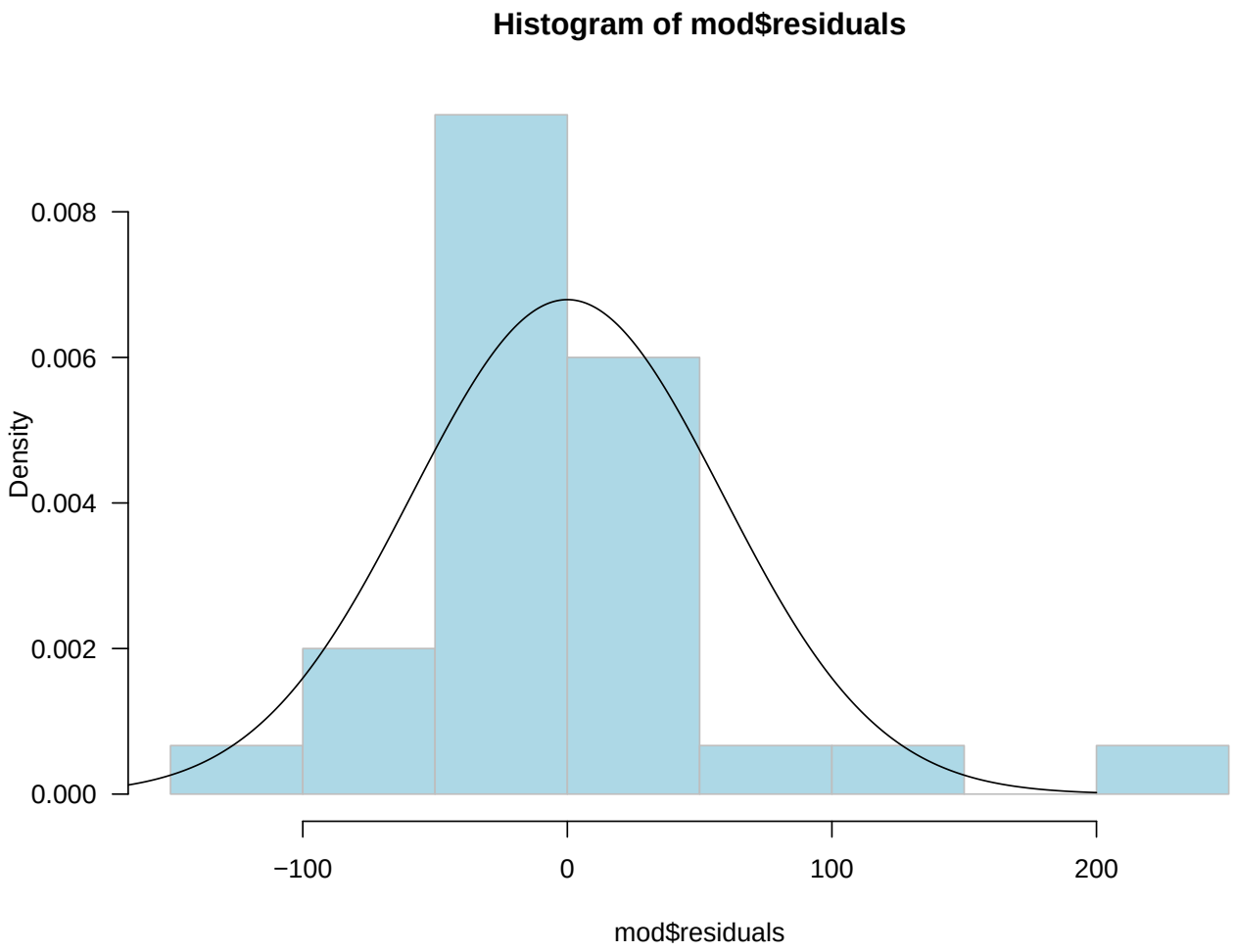
```r
mod <- lm(Species ~ Elevation + Adjacent, data = galaNew)
plot(mod$fitted.values, mod$residuals, pch = 16, col = "blue")
abline(h = 0, col = "red")
```



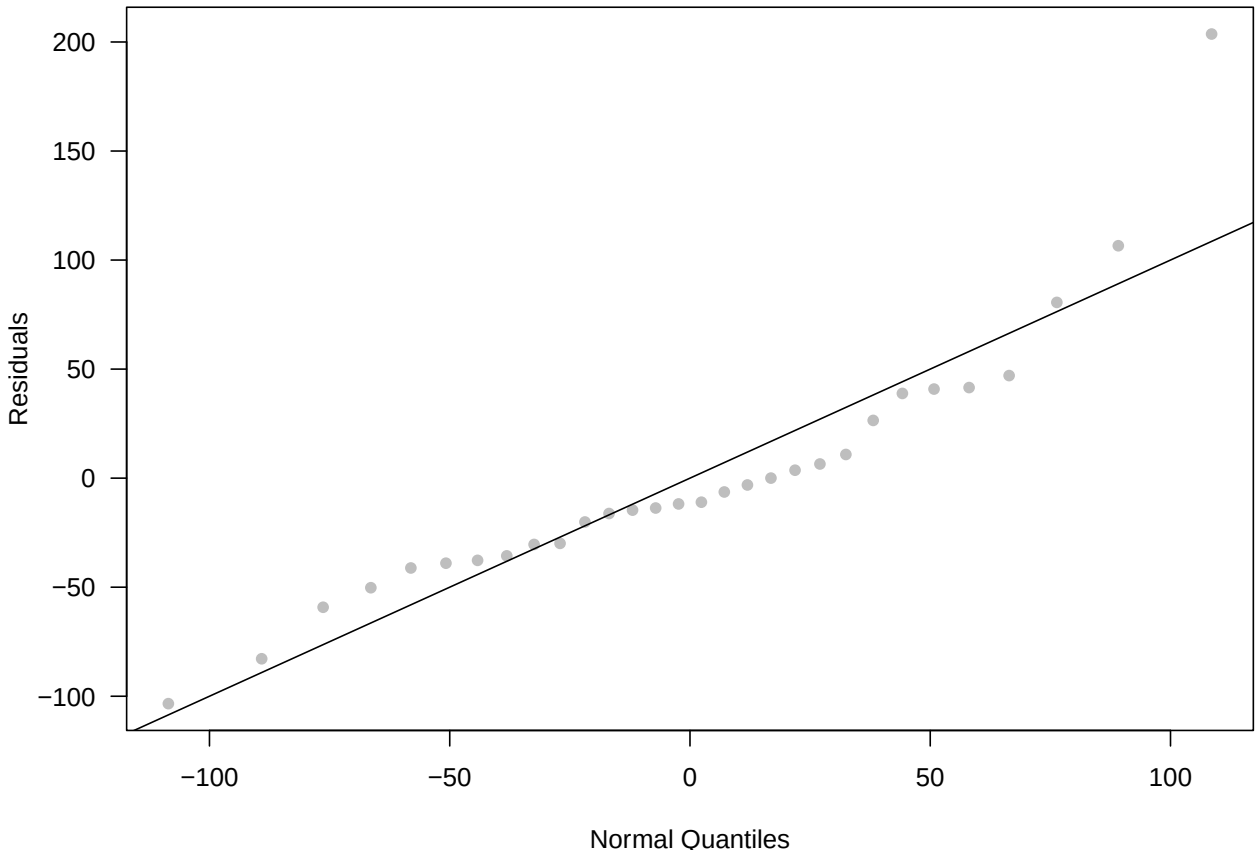
Residual Histogram/QQplot

These are used for assessing normality of residuals

```
par(las = 1)
hist(mod$residuals, 5, prob = T, col = "lightblue", border = "gray")
xg <- seq(-200, 200, 1)
sd <- sd(mod$residuals)
yg <- dnorm(xg, 0, sd)
lines(xg, yg)
```




```
plot(qnorm(1:30 / 31, 0, sd), sort(mod$residuals), pch = 16,
     col = "gray", xlab = "Normal Quantiles", ylab = "Residuals")
abline(0, 1)
```



Leverage

Detecting *extreme* predictor values

```
step_gala <- step(full, trace = F)
X <- model.matrix(step_gala)
H <- X %*% solve((t(X) %*% X)) %*% t(X)
diag(H)
```

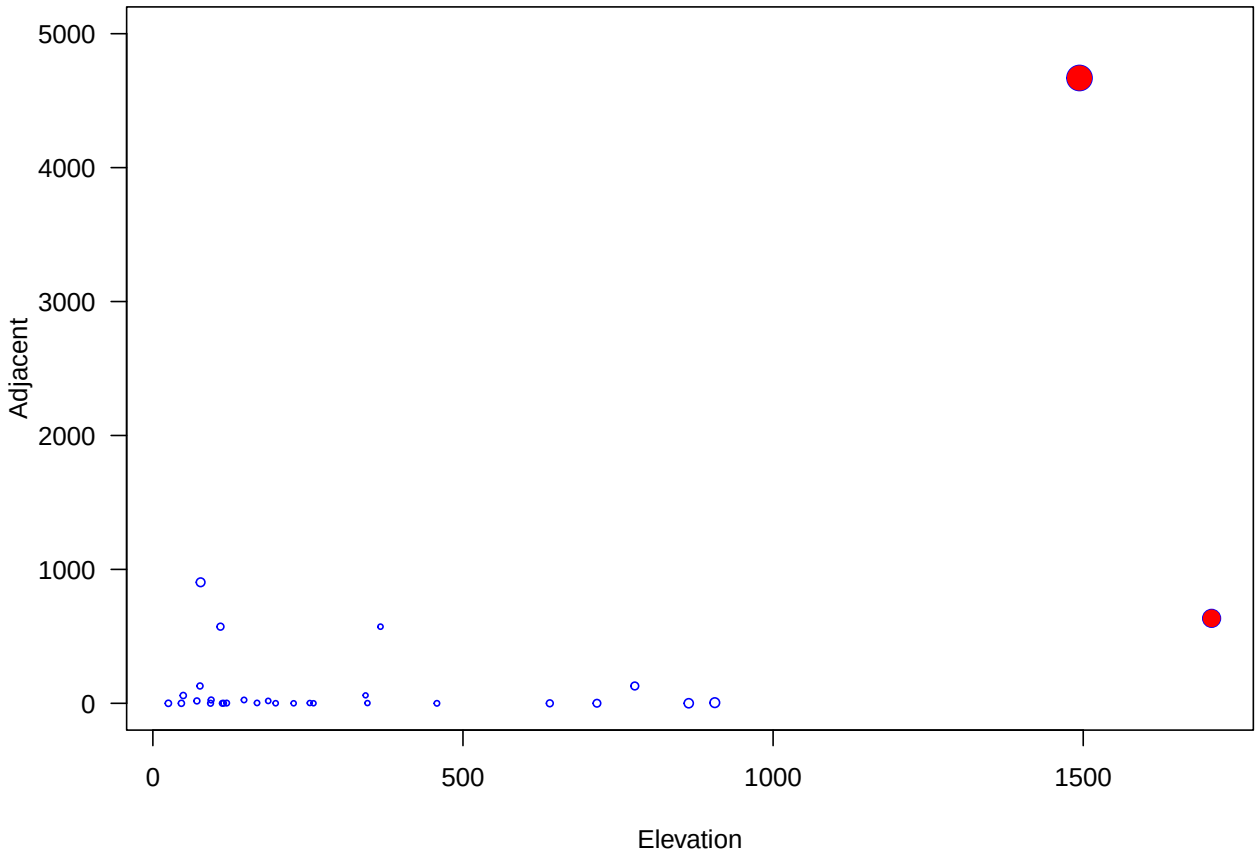
```
##       Baltra    Bartolome     Caldwell     Champion      Coamano Daphne.Major
##   0.03700564   0.06937466   0.04587610   0.05401592   0.10982345   0.04537841
## Daphne.Minor       Darwin         Eden      Enderby     Espanola   Fernandina
##   0.04812088   0.04119028   0.05090200   0.04607792   0.03929182   0.93009727
##     Gardner1     Gardner2     Genovesa      Isabela     Marchena       Onslow
##   0.05449980   0.03791638   0.05220755   0.45944837   0.03541621   0.05703802
##        Pinta       Pinzon   Las.Plazas       Rabida SanCristobal  SanSalvador
##   0.08768347   0.04330066   0.04817863   0.03965441   0.08363093   0.13605950
##    SantaCruz      SantaFe   SantaMaria      Seymour      Tortuga         Wolf
##   0.12315276   0.03692090   0.06800977   0.04281440   0.03988084   0.03703304
```

```
lev <- hat(X)
hatvalues(step_gala)
```

```
##       Baltra    Bartolome     Caldwell     Champion      Coamano Daphne.Major
```

```
##   0.03700564   0.06937466   0.04587610   0.05401592   0.10982345   0.04537841
## Daphne.Minor       Darwin         Eden      Enderby     Espanola   Fernandina
##   0.04812088   0.04119028   0.05090200   0.04607792   0.03929182   0.93009727
##     Gardner1     Gardner2     Genovesa      Isabela     Marchena       Onslow
##   0.05449980   0.03791638   0.05220755   0.45944837   0.03541621   0.05703802
##        Pinta       Pinzon   Las.Plazas       Rabida SanCristobal  SanSalvador
##   0.08768347   0.04330066   0.04817863   0.03965441   0.08363093   0.13605950
##    SantaCruz      SantaFe   SantaMaria      Seymour      Tortuga         Wolf
##   0.12315276   0.03692090   0.06800977   0.04281440   0.03988084   0.03703304
```

```
high_lev <- which(lev >= 2 * 3 / 30)
attach(gala)
par(las = 1)
plot(Elevation, Adjacent, cex = sqrt(5 * lev), col = "blue", ylim = c(0, 5000))
points(Elevation[high_lev], Adjacent[high_lev], col = "red", pch = 16,
       cex = sqrt(5 *lev[high_lev]))
```



Standardized Residuals

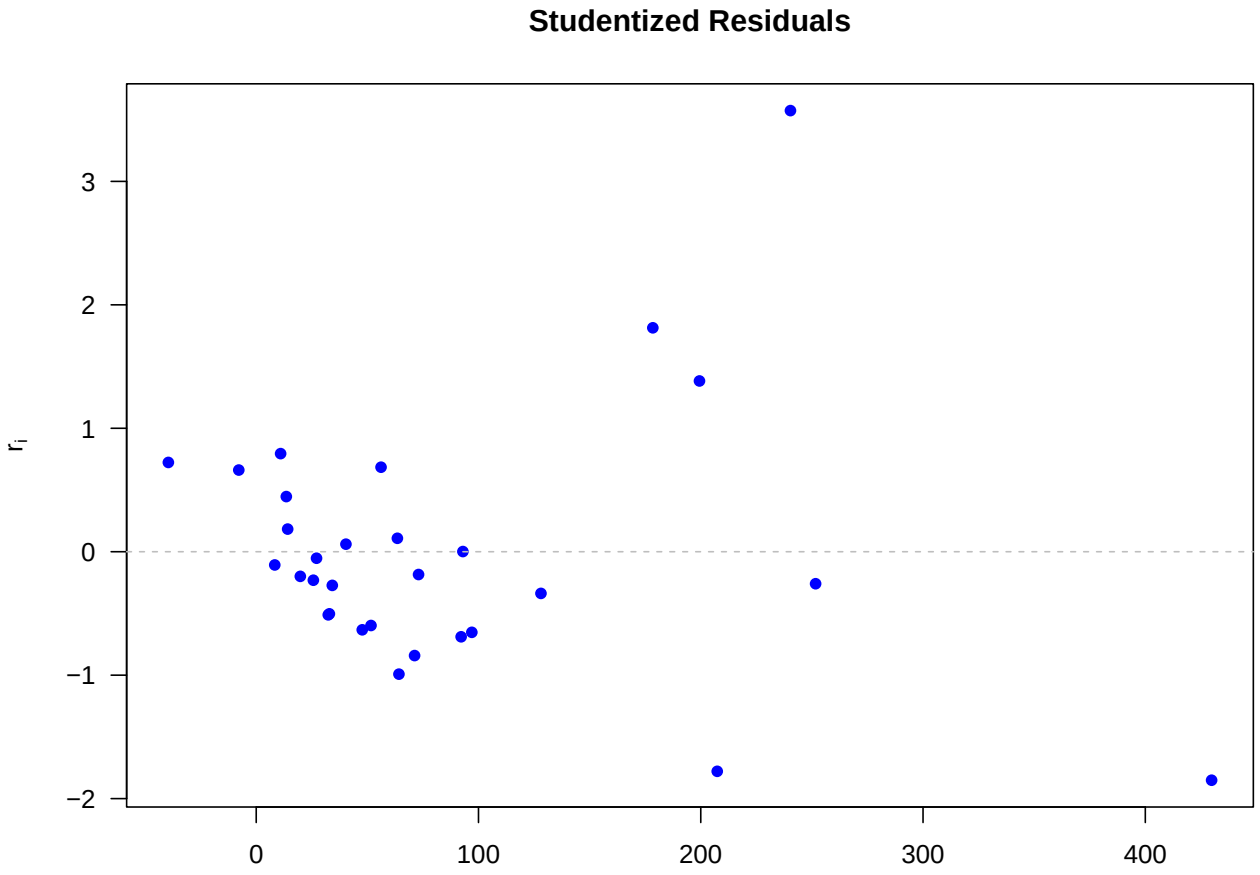
```
gs <- summary(step_gala)
gs$sig
```

```
## [1] 60.85898
```

```r
studRes <- gs$res / (gs$sig * sqrt(1 - lev))

rstandard(step_gala)
```

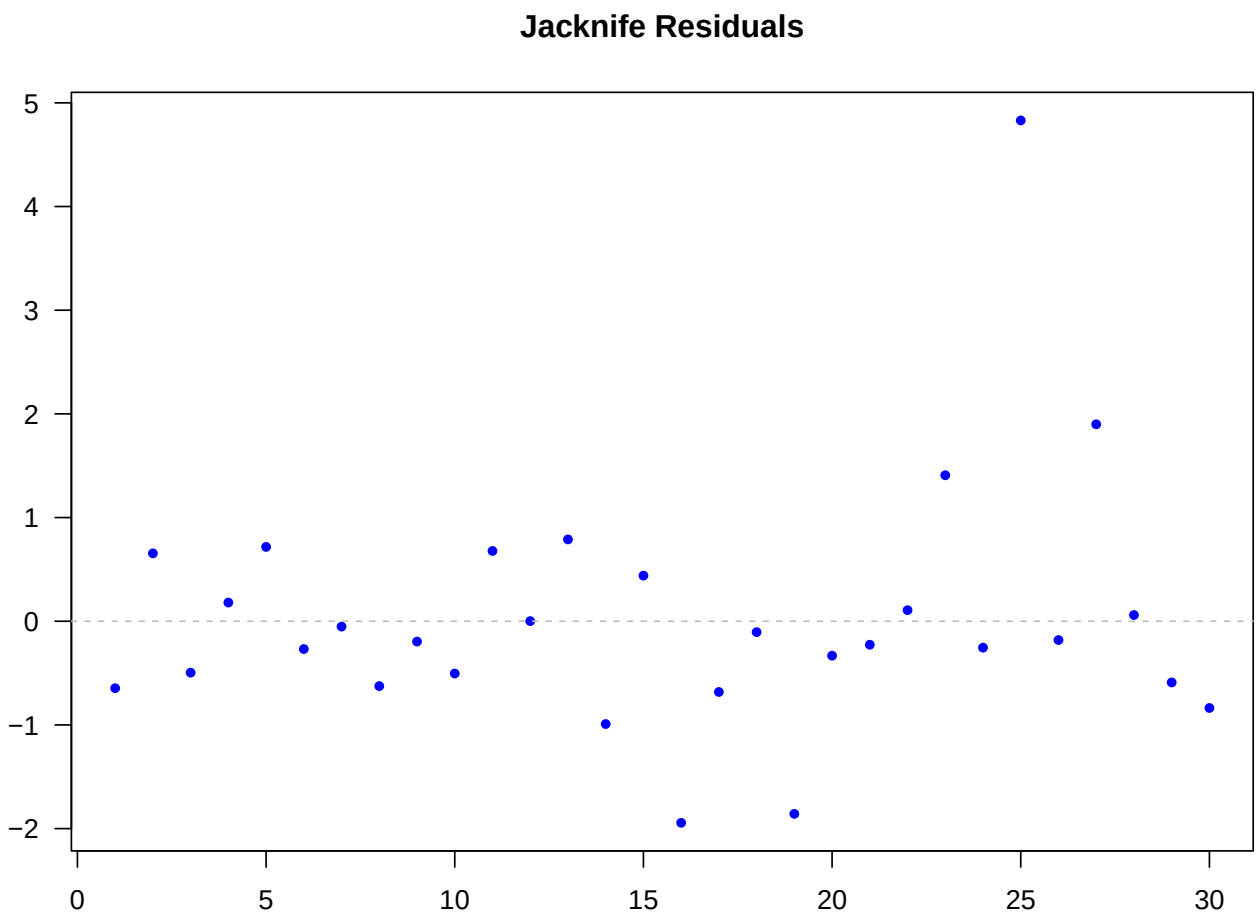
```
##       Baltra    Bartolome     Caldwell     Champion      Coamano Daphne.Major
## -0.653001500  0.661666192 -0.503105720  0.183425063  0.723293423 -0.272740922
## Daphne.Minor       Darwin         Eden      Enderby     Espanola   Fernandina
## -0.052719435 -0.632631364 -0.199574302 -0.511464841  0.684743212  0.001402059
##     Gardner1     Gardner2     Genovesa      Isabela     Marchena       Onslow
##  0.794716944 -0.991713650  0.446723234 -1.851112453 -0.689173432 -0.107282919
##        Pinta       Pinzon   Las.Plazas       Rabida SanCristobal  SanSalvador
## -1.778894534 -0.337647762 -0.230770414  0.108849636  1.383203903 -0.259281587
##    SantaCruz      SantaFe   SantaMaria      Seymour      Tortuga         Wolf
##  3.573496675 -0.184650534  1.813868781  0.061132164 -0.597622667 -0.841308195
```

```r
par(las = 1)
plot(step_gala$fitted.values, studRes, pch = 16, col = "blue",
     ylab = expression(r[i]), main = "Studentized Residuals", xlab = "")
abline(h = 0, lty = 2, col = "gray")
```



Studentized (Jackknife) Residuals

```r
jack <- rstudent(step_gala)

par(las = 1)
plot(jack, pch = 16, cex = 0.8, col = "blue", main =" Jacknife Residuals ",
     xlab = "", ylab = "")
abline(h = 0, lty = 2, col = "gray")
```

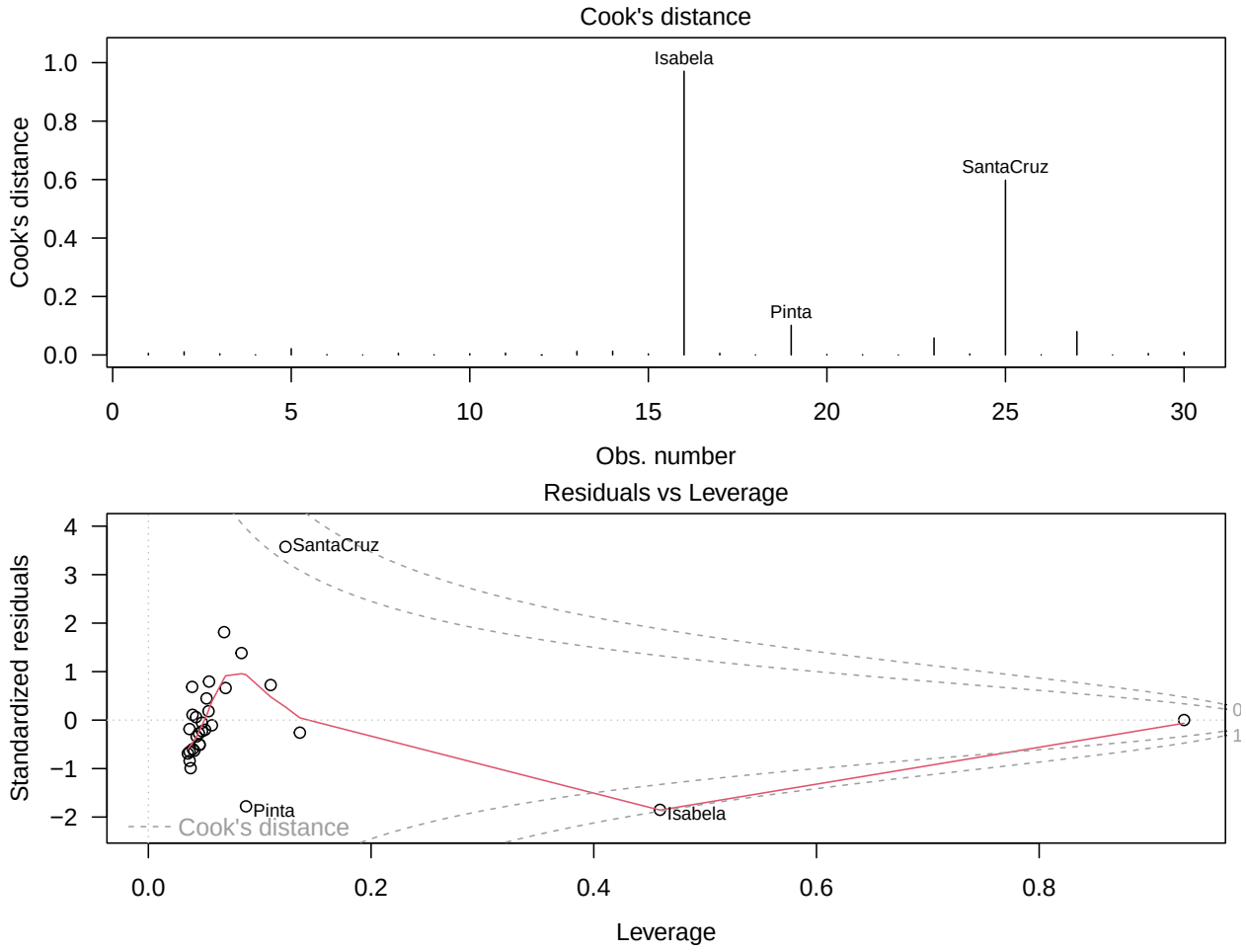


Identifying Influential Observations: Cook's Distance

```r
cooks.distance(step_gala)
```

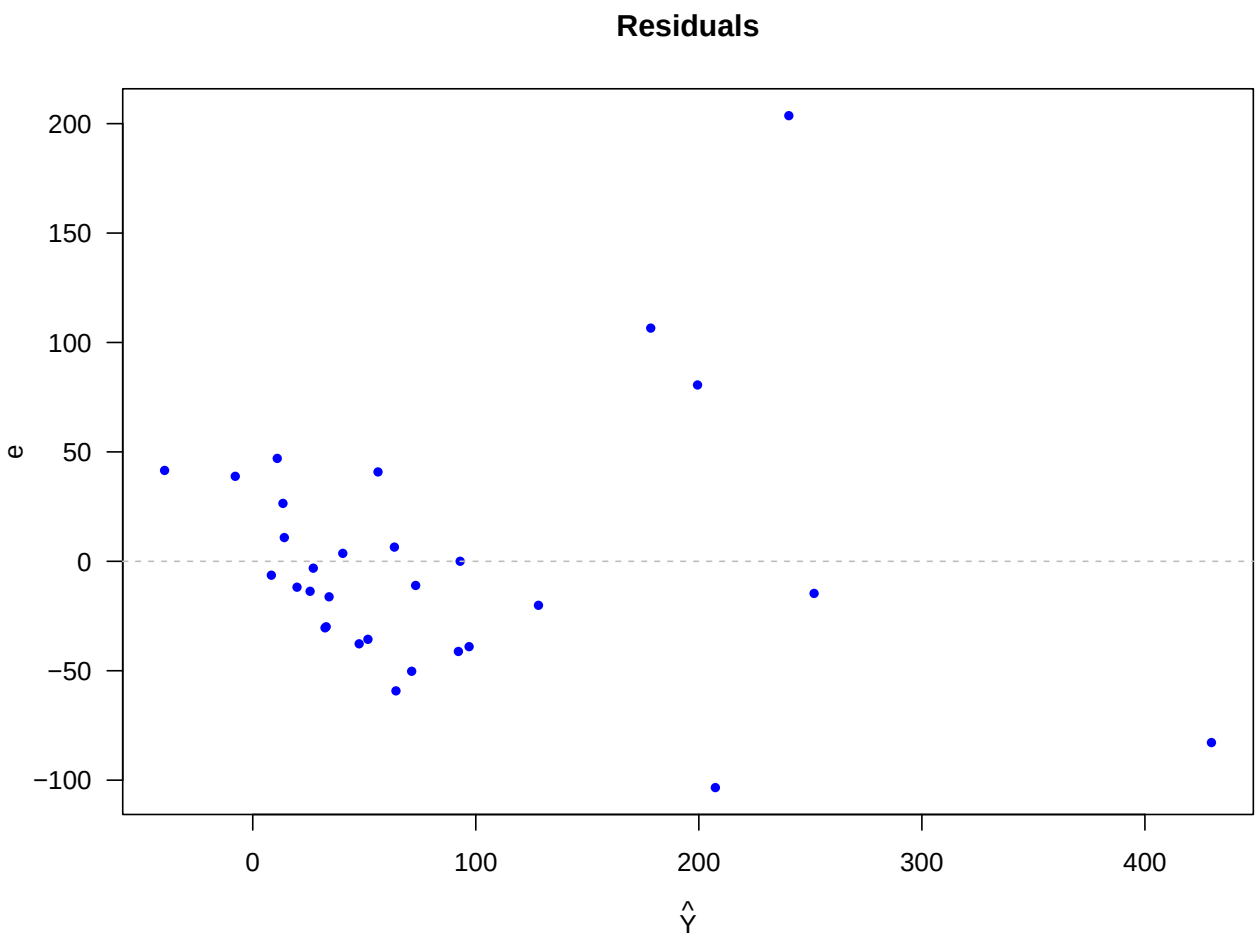
```
##       Baltra    Bartolome     Caldwell     Champion      Coamano Daphne.Major
## 5.461995e-03 1.087884e-02 4.056757e-03 6.403746e-04 2.151427e-02 1.178684e-03
## Daphne.Minor       Darwin         Eden      Enderby     Espanola   Fernandina
## 4.683516e-05 5.731160e-03 7.120521e-04 4.212018e-03 6.392119e-03 8.718575e-06
##     Gardner1     Gardner2     Genovesa      Isabela     Marchena       Onslow
## 1.213492e-02 1.292009e-02 3.664172e-03 9.708315e-01 5.812968e-03 2.320653e-04
##        Pinta       Pinzon   Las.Plazas       Rabida SanCristobal  SanSalvador
## 1.013798e-01 1.719988e-03 8.985413e-04 1.630785e-04 5.820331e-02 3.529126e-03
##    SantaCruz      SantaFe   SantaMaria      Seymour      Tortuga         Wolf
## 5.978410e-01 4.357026e-04 8.002956e-02 5.572012e-05 4.945065e-03 9.073336e-03
```

```
par(mfrow = c(2, 1), mar = c(3.8, 3.8, 1.2, 0.5), mgp = c(2.5, 1, 0), las = 1)
plot(step_gala, which = 4:5)
```

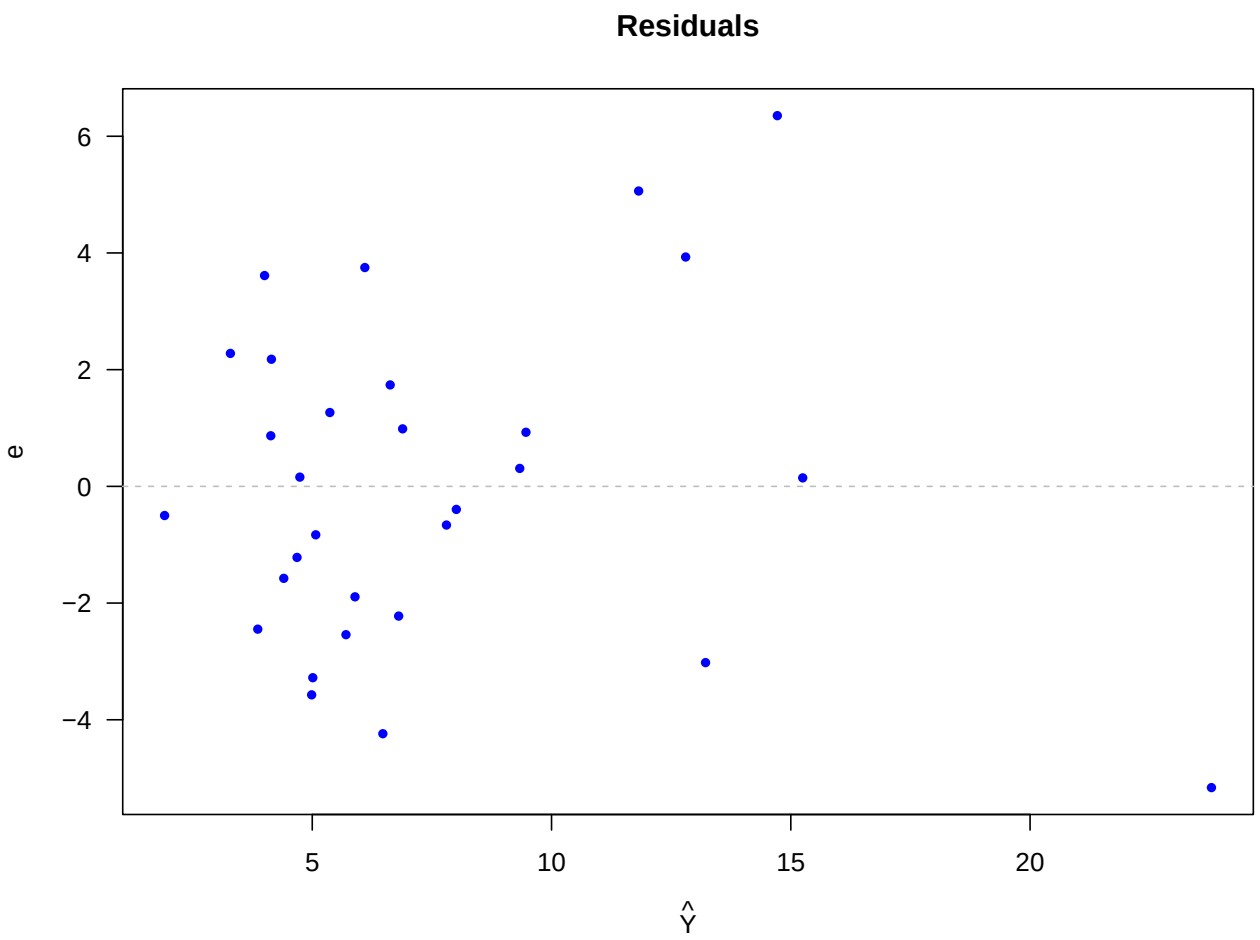


Response transformation

```
par(las = 1)
plot(step_gala$fitted.values, step_gala$residuals,
     pch = 16, cex = 0.8, col = "blue", main =" Residuals ",
     xlab = expression(hat(Y)), ylab = expression(e))
abline(h = 0, lty = 2, col = "gray")
```

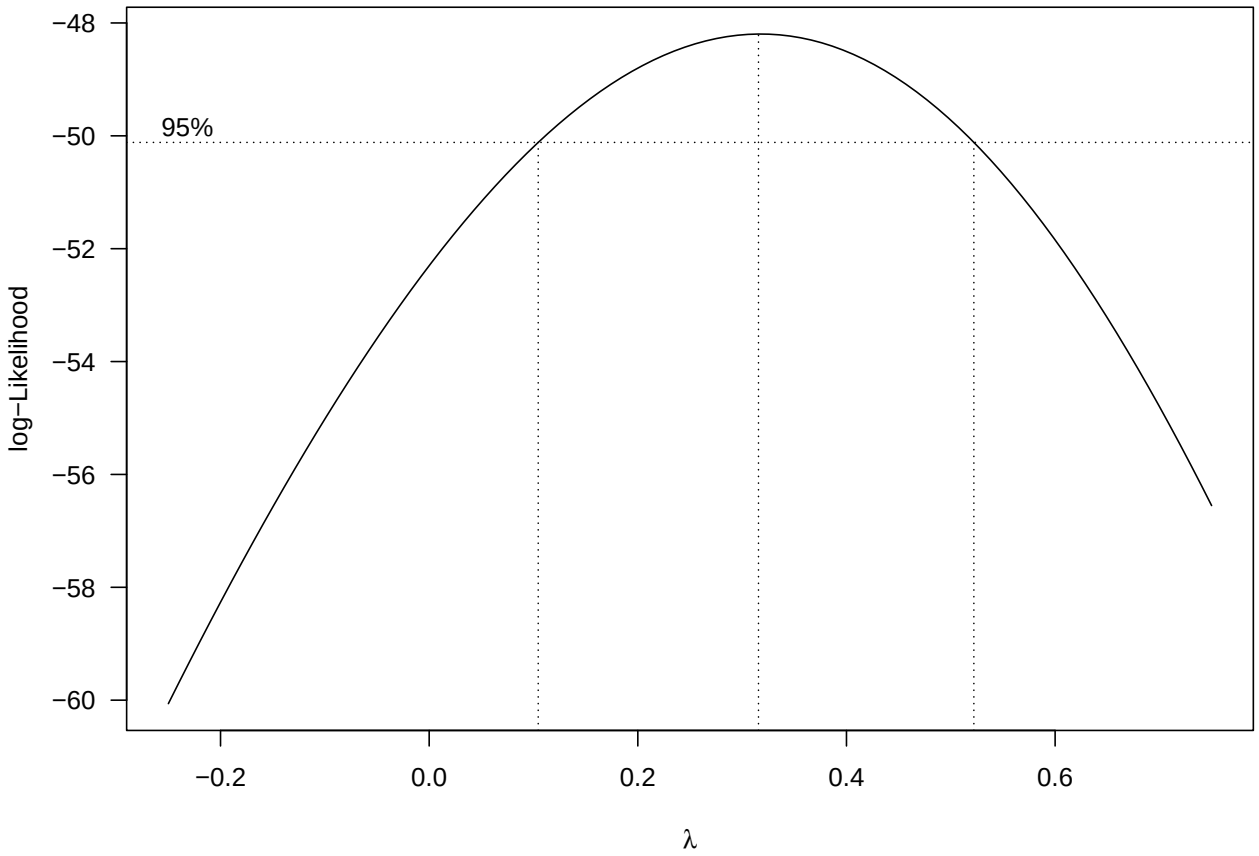



```r
sqrt_fit <- lm(sqrt(Species) ~ Elevation + Adjacent)

plot(sqrt_fit$fitted.values, sqrt_fit$residuals,
     pch = 16, cex = 0.8, col = "blue", main =" Residuals ",
     xlab = expression(hat(Y)), ylab = expression(e))
abline(h = 0, lty = 2, col = "gray")
```



Box-Cox Transformation

```
library(MASS)
par(las = 1)
boxcox <- boxcox(step_gala, plotit = T, lambda = seq(-0.25, 0.75, by = 0.05))
```


−48
−50
−52
−54
−56
−58
−60
95%
log−Likelihood
−0.2
0.0
0.2
0.4
0.6
λ