DSA 8020 R Session 4: Multiple Linear Regression III

Whitney Huang, Clemson University

Contents

We will use the same data and code as before:

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

Model Selection

All Subset Selection

Fit all possible subsets of predictors and summarize model selection criteria:

```
library(leaps)

# Fit all subset regression models using all predictors
models <- regsubsets(Species ~ ., data = galaNew)

# View model selection results (e.g., Cp, BIC, adjusted R^2)
(res.sum <- 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

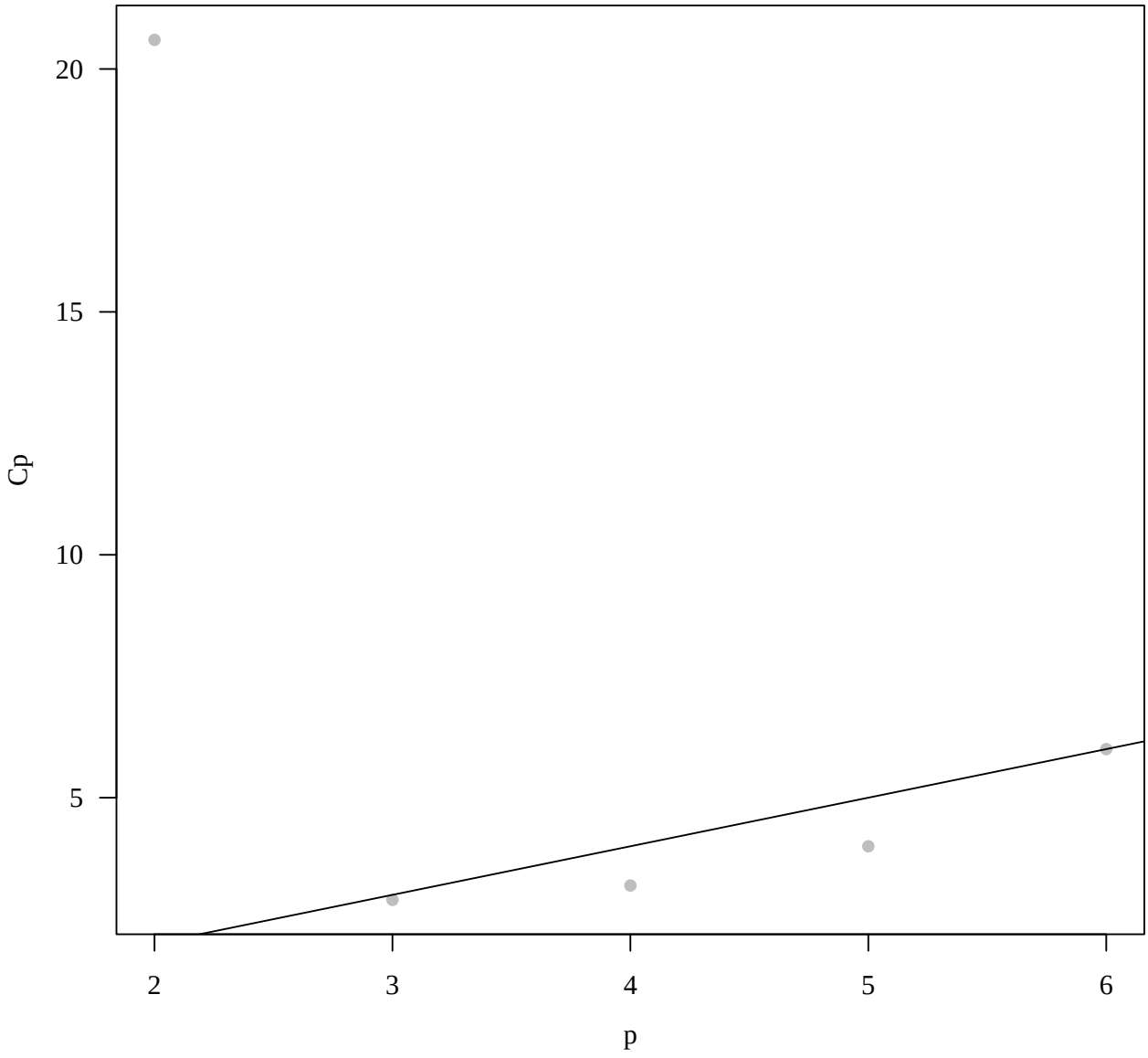
Extract and compare key criteria across subset models:

```r
# Create a table of model selection criteria
criteria <- data.frame(
  Adj.R2 = res.sum$adjr2,
  Cp     = res.sum$cp,
  BIC    = res.sum$bic)

criteria
```

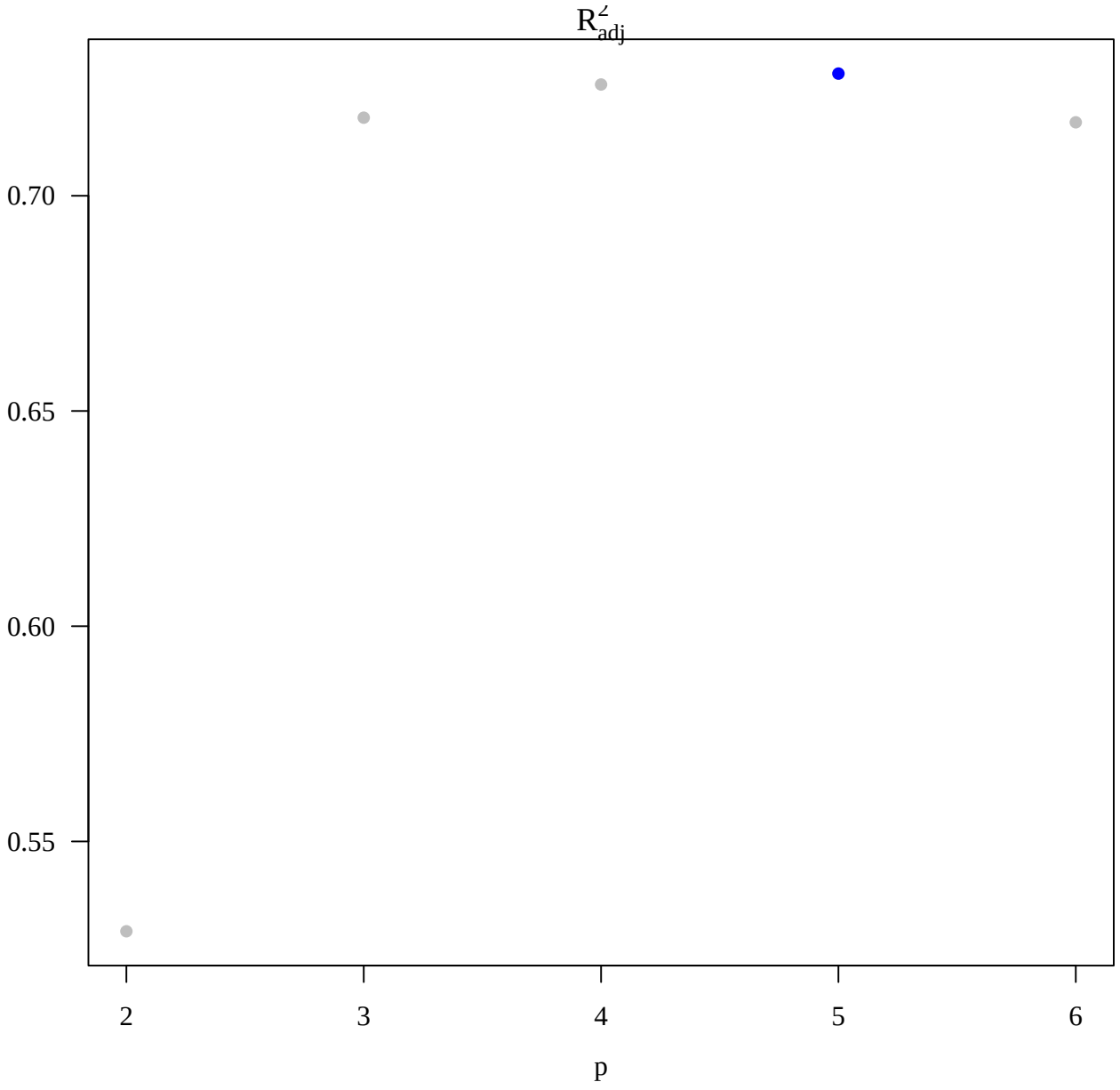
```
##      Adj.R2        Cp       BIC
## 1 0.5291255 20.599003 -16.84525
## 2 0.7181425  2.897184 -29.93078
## 3 0.7258462  3.193068 -28.49317
## 4 0.7283816  4.000075 -26.54733
## 5 0.7170651  6.000000 -23.14622
```

Visualize criteria vs. number of predictors (p):

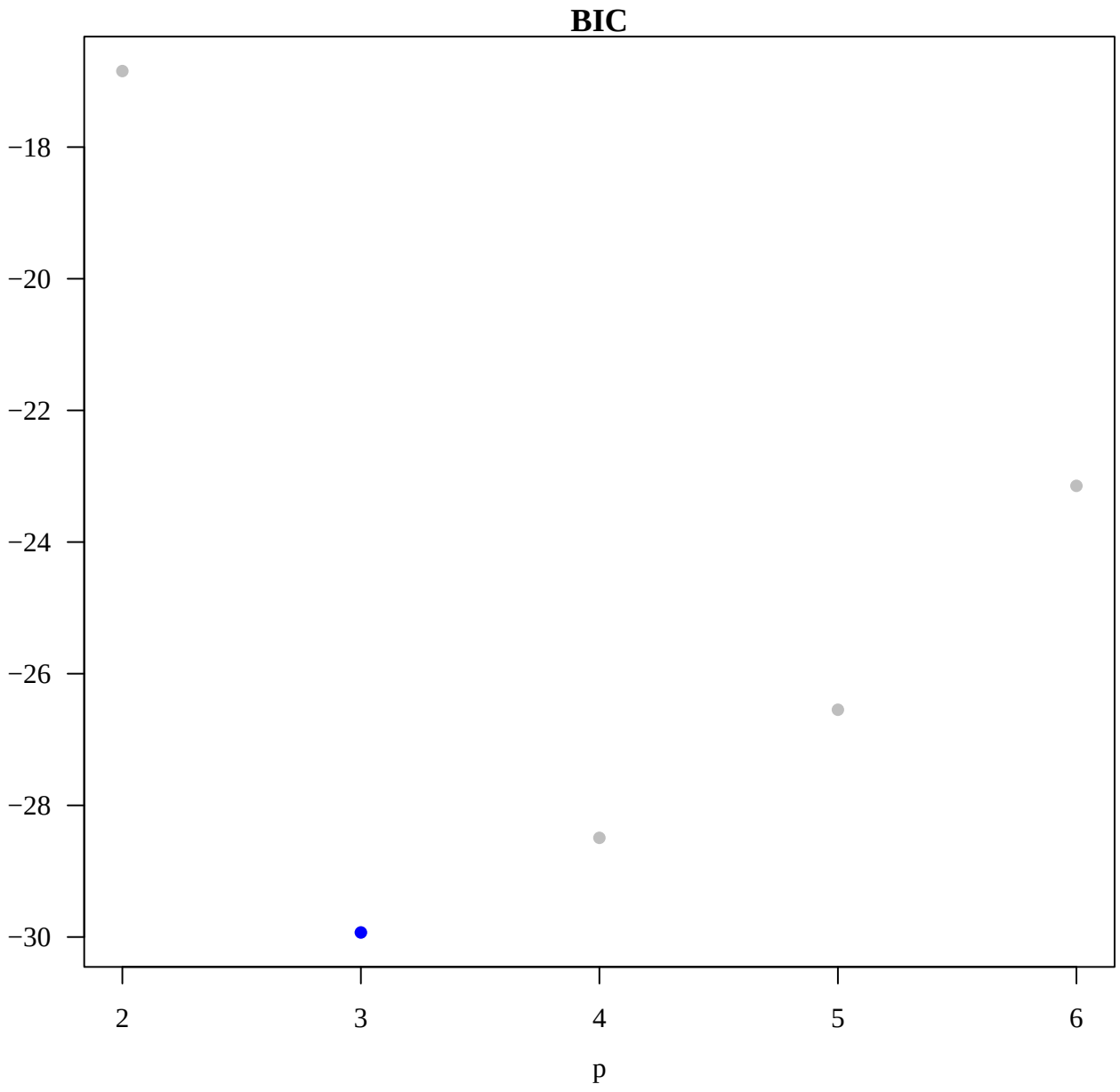
```r
par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2.5, 1, 0), family = "serif")

p <- 2:6  # number of predictors

# Cp plot (smaller is better; compare to diagonal line)
plot(p, criteria$Cp, pch = 16, col = "gray",
     xlab = "p", ylab = "Cp", las = 1)
abline(0, 1)
```




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




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



Backward Selection

Start with the full model and remove predictors step-by-step:

```
full <- lm(Species ~ ., data = galaNew)

# Backward elimination
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

Allow both adding and removing predictors:

```r
# Stepwise selection (both directions)
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
```

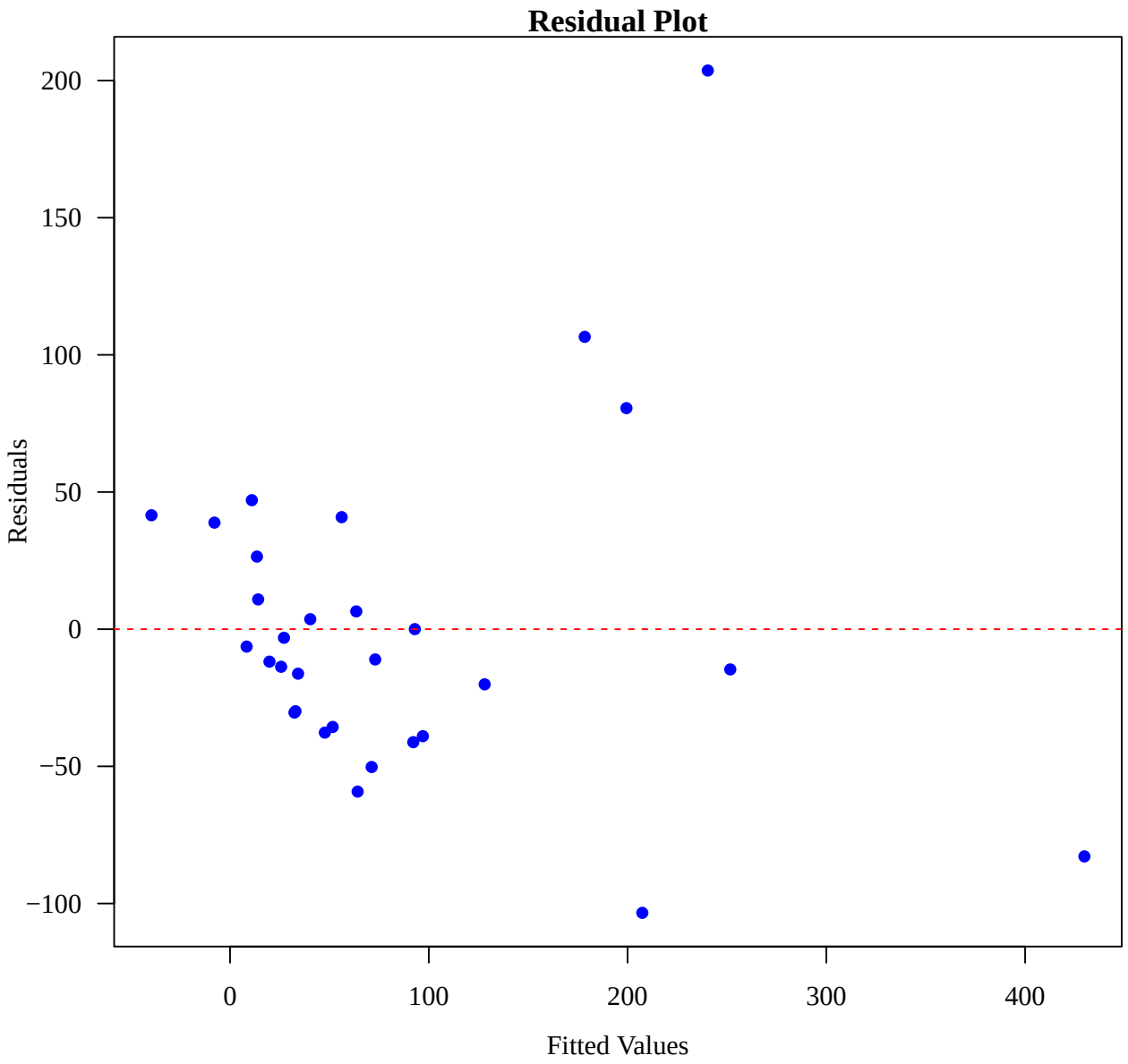
Model Diagnostics

After selecting a model, we check whether the regression assumptions are reasonable.

Residual Plot

Use the residual plot to check linearity and constant variance.

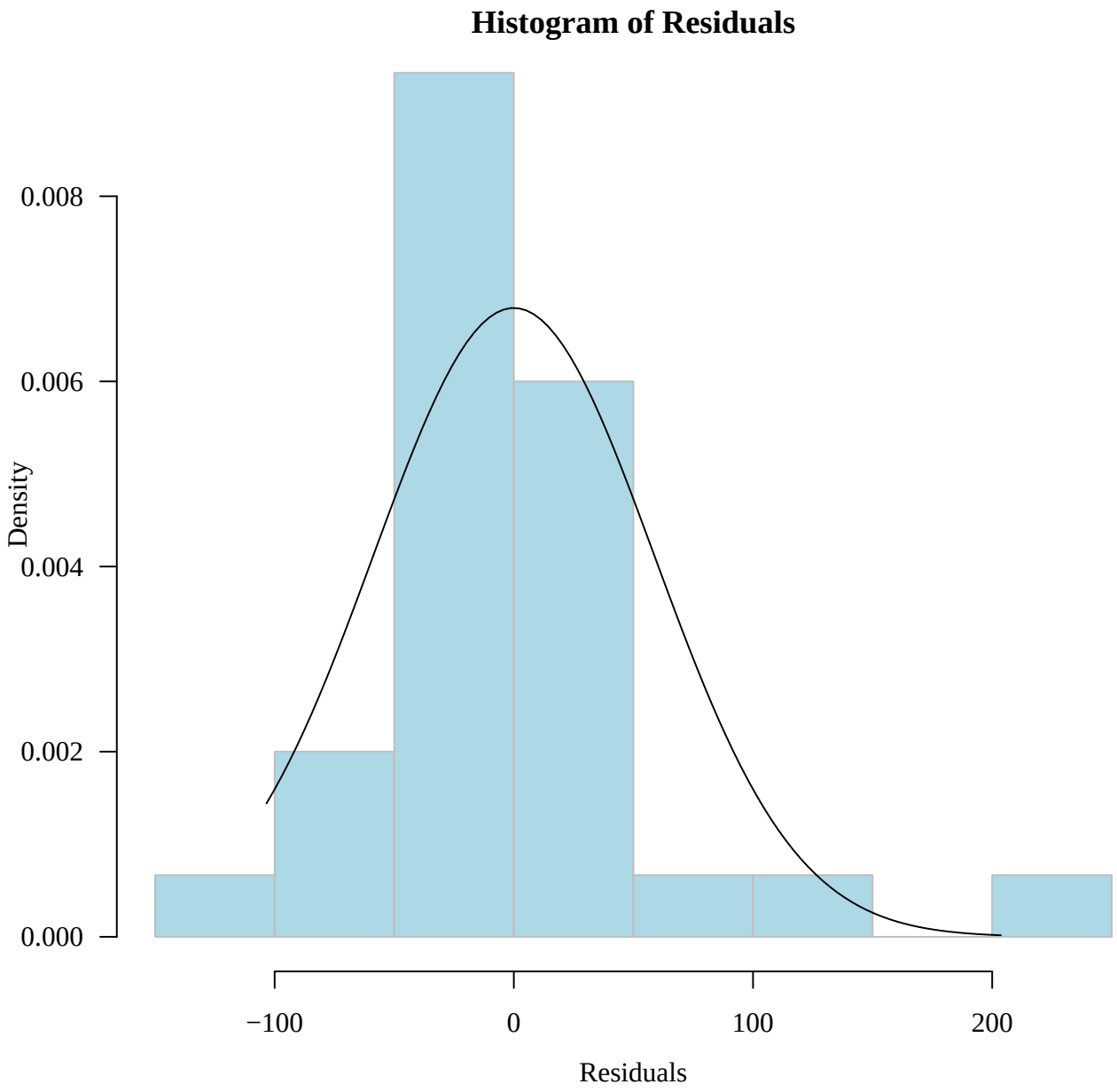
```
mod <- lm(Species ~ Elevation + Adjacent, data = galaNew)

par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2.5, 1, 0), family = "serif")
plot(mod$fitted.values, mod$residuals,
     pch = 16, col = "blue",
     xlab = "Fitted Values",
     ylab = "Residuals",
     main = "Residual Plot")
abline(h = 0, col = "red", lty = 2)
```



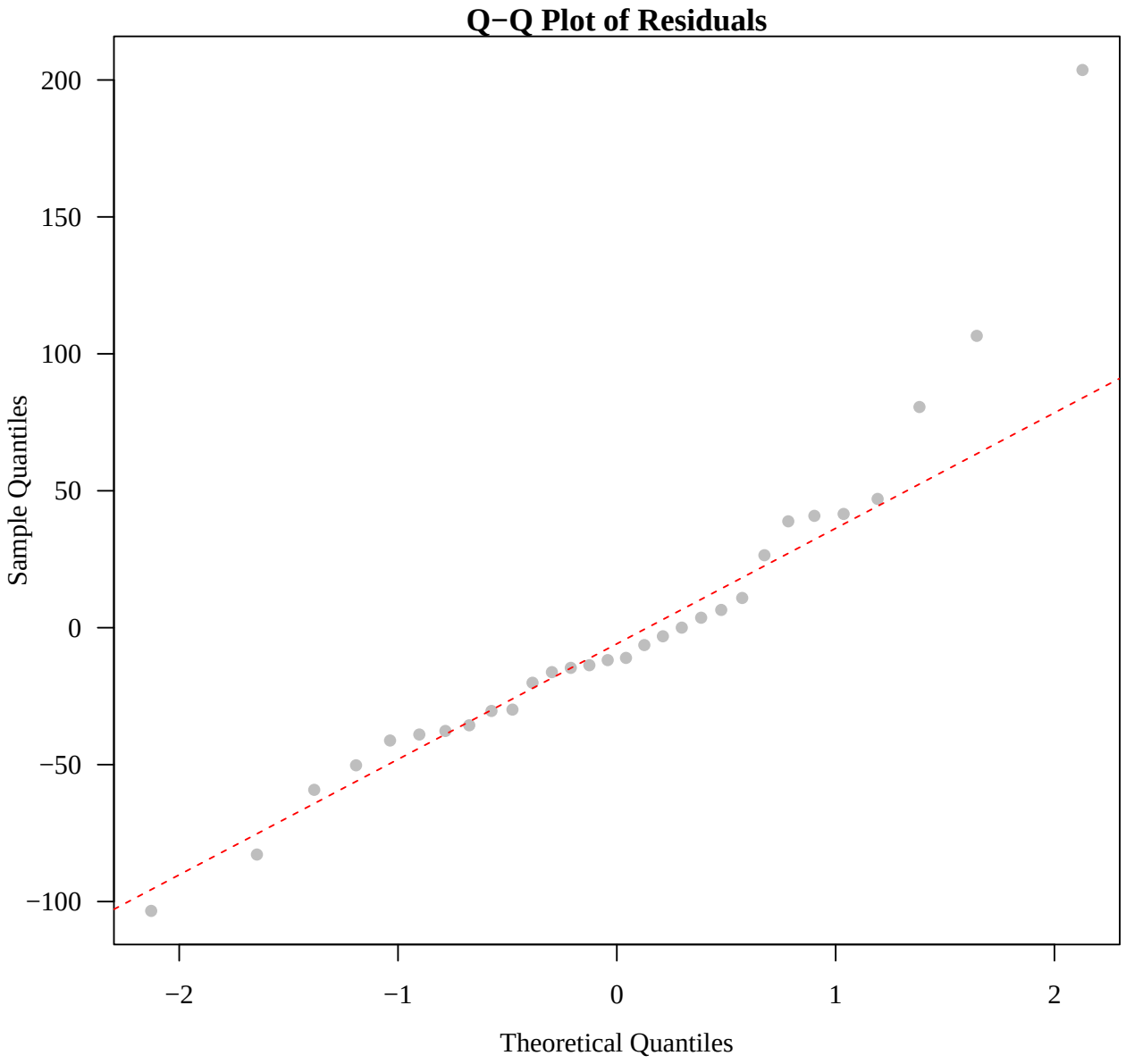
Residual Histogram/QQplot

Use the histogram and Q-Q plot to assess whether residuals are approximately normal.

```
par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2.5, 1, 0), family = "serif")
# Histogram of residuals with normal curve
hist(mod$residuals, breaks = 5, probability = TRUE,
     col = "lightblue", border = "gray",
     xlab = "Residuals",
     main = "Histogram of Residuals")

xg <- seq(min(mod$residuals), max(mod$residuals), length.out = 100)
yg <- dnorm(xg, mean = 0, sd = sd(mod$residuals))
lines(xg, yg)
```



```r
# Q-Q plot
qqnorm(mod$residuals, pch = 16, col = "gray",
       main = "Q-Q Plot of Residuals")
qqline(mod$residuals, col = "red", lty = 2)
```

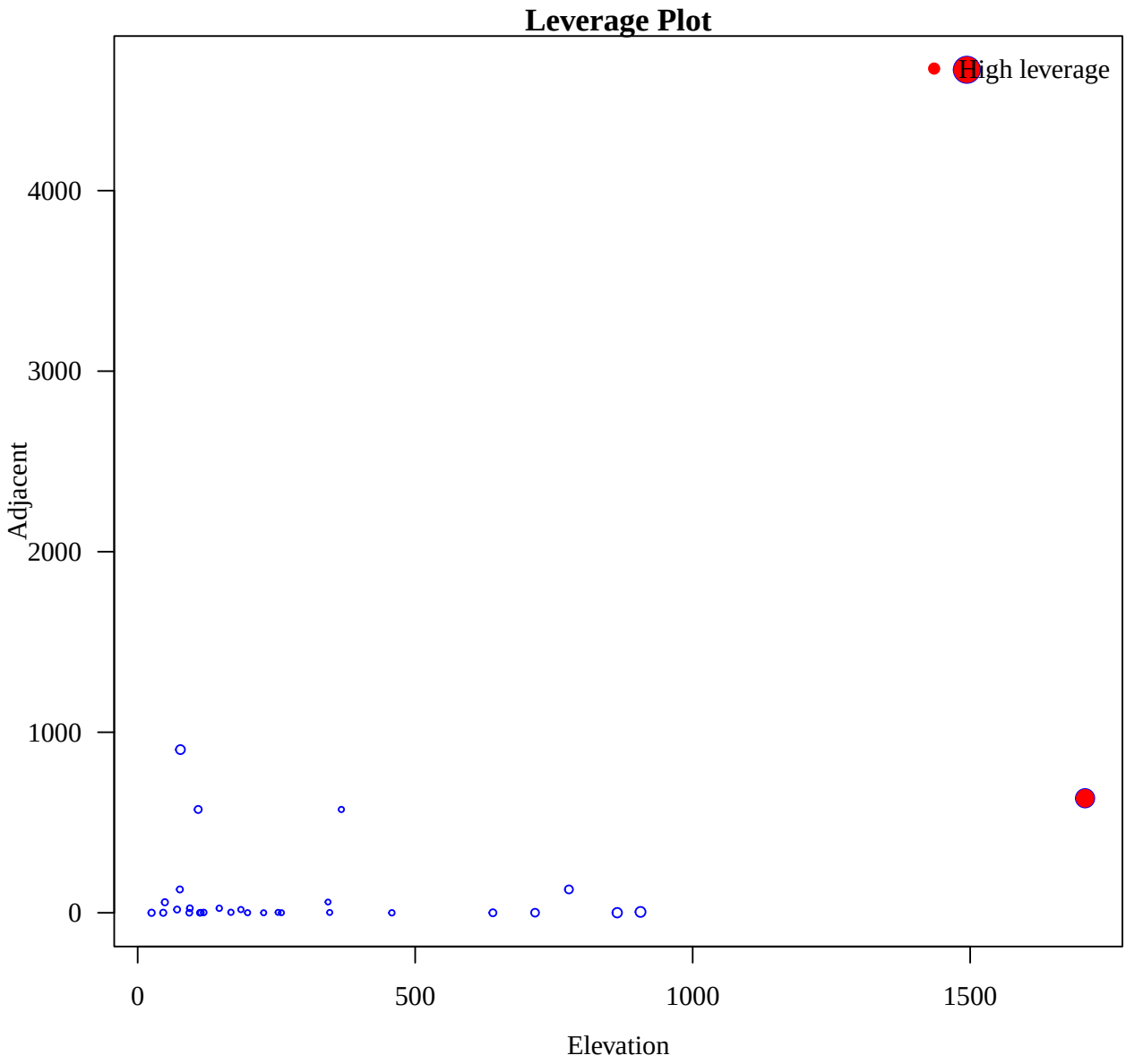


Leverage

Leverage identifies observations with unusual predictor values.

```r
par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2.5, 1, 0), family = "serif")
# Use the selected model
step_gala <- step(full, trace = FALSE)

# Compute leverage values
lev <- hatvalues(step_gala)
```

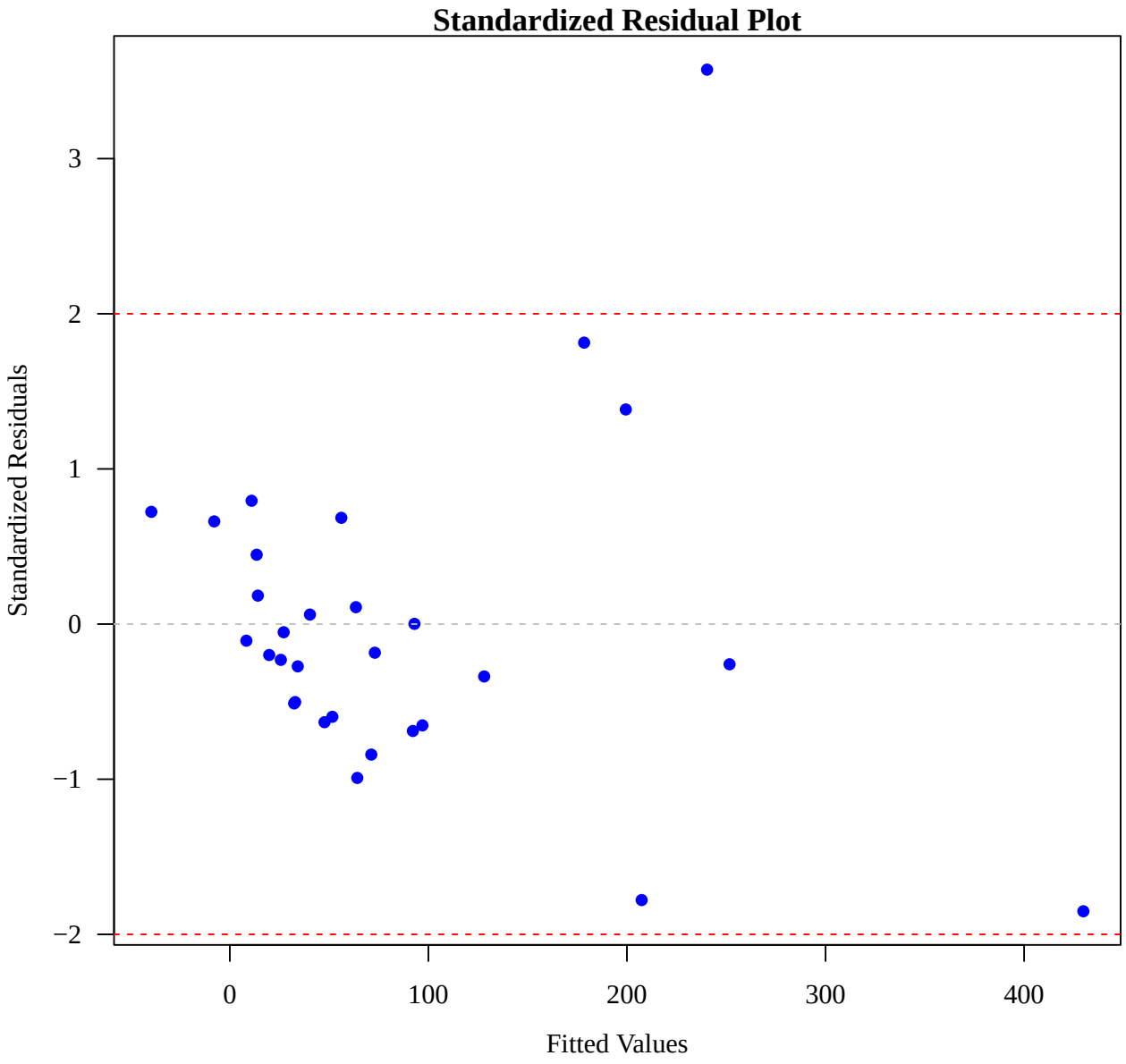
```r
# Rule of thumb for high leverage
p <- length(coef(step_gala))   # number of parameters, including intercept
n <- nrow(galaNew)

high_lev <- which(lev > 2 * p / n)

# Visualize leverage using circle size
par(las = 1)
plot(galaNew$Elevation, galaNew$Adjacent,
     cex = sqrt(5 * lev),
     col = "blue",
     xlab = "Elevation",
     ylab = "Adjacent",
     main = "Leverage Plot")

points(galaNew$Elevation[high_lev], galaNew$Adjacent[high_lev],
       col = "red", pch = 16,
       cex = sqrt(5 * lev[high_lev]))

legend("topright",
       legend = "High leverage",
       col = "red", pch = 16, bty = "n")
```



Standardized Residuals

Standardized residuals adjust residuals for unequal variance caused by leverage.

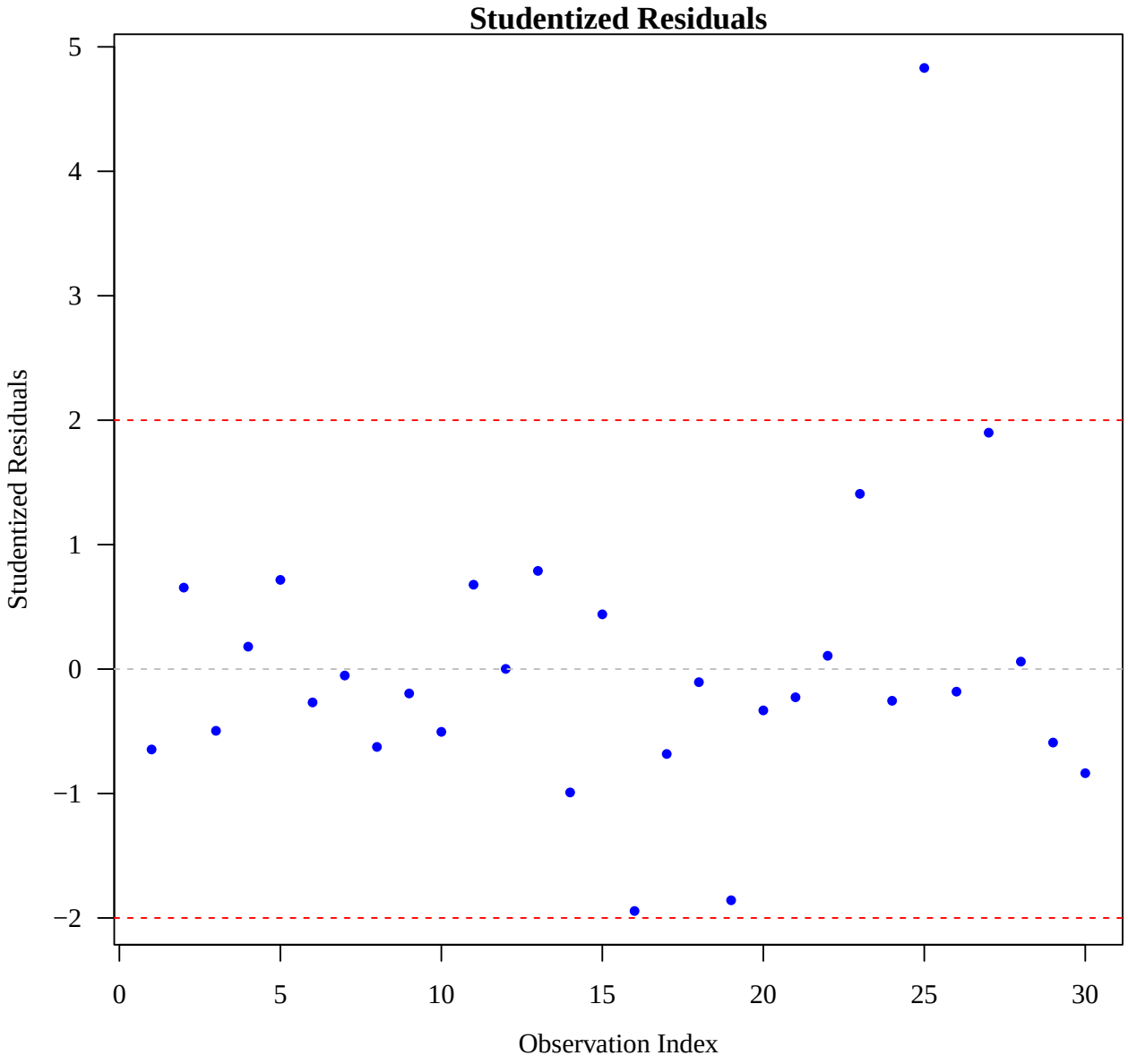
```
# Standardized residuals
std_res <- rstandard(step_gala)

par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2.5, 1, 0), family = "serif")
plot(step_gala$fitted.values, std_res,
     pch = 16, col = "blue",
     xlab = "Fitted Values",
     ylab = "Standardized Residuals",
     main = "Standardized Residual Plot")
abline(h = 0, col = "gray", lty = 2)
abline(h = c(-2, 2), col = "red", lty = 2)
```



Studentized (Jackknife) Residuals

Studentized residuals are useful for detecting possible outliers.

```
# Studentized residuals
jack_res <- rstudent(step_gala)

par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2.5, 1, 0), family = "serif")
plot(jack_res,
     pch = 16, cex = 0.8, col = "blue",
     xlab = "Observation Index",
     ylab = "Studentized Residuals",
     main = "Studentized Residuals")
abline(h = 0, col = "gray", lty = 2)
abline(h = c(-2, 2), col = "red", lty = 2)
```



Identifying Influential Observations: DFFITS

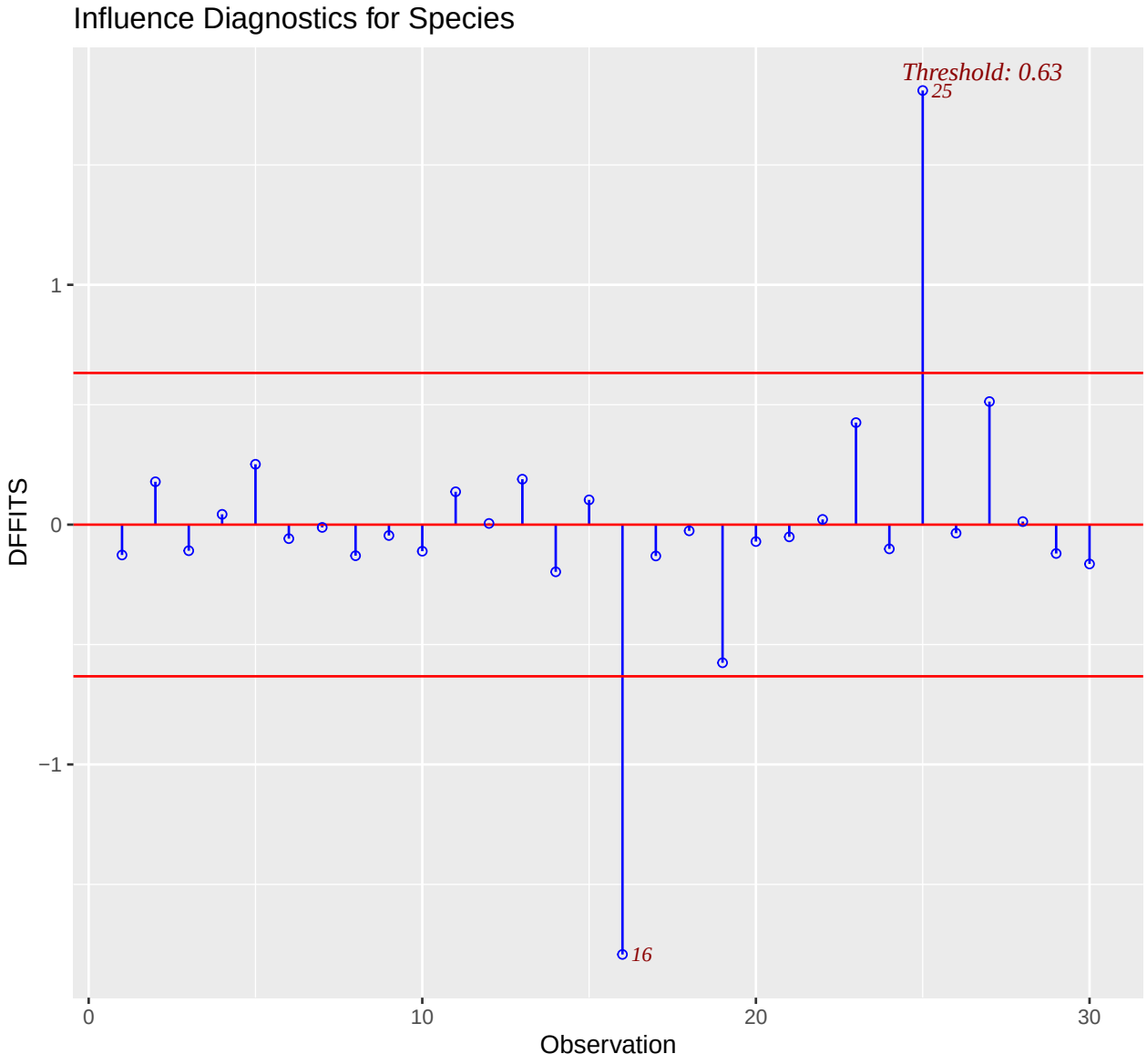
DFFITS measures how much the fitted value changes when one observation is removed.

```
library(olsrr)

# Numerical DFFITS values
dffits(step_gala)
```

```
##       Baltra    Bartolome     Caldwell     Champion      Coamano Daphne.Major
## -0.126618703  0.178733773 -0.108767759  0.043038112  0.251754666 -0.058433675
## Daphne.Minor       Darwin         Eden      Enderby     Espanola   Fernandina
## -0.011632519 -0.129637172 -0.045388086 -0.110847189  0.137085618  0.005018665
##     Gardner1     Gardner2     Genovesa      Isabela     Marchena       Onslow
##  0.189462681 -0.196813788  0.103267647 -1.792290026 -0.130742944 -0.025897813
```

```
##        Pinta       Pinzon   Las.Plazas       Rabida SanCristobal  SanSalvador
## -0.575984137 -0.070639403 -0.050999176  0.021709963  0.425401441 -0.101097482
##    SantaCruz      SantaFe   SantaMaria      Seymour      Tortuga         Wolf
##  1.810238758 -0.035500535  0.513106873  0.012688243 -0.120321428 -0.164065528
```

```
# DFFITS plot
par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2.5, 1, 0), family = "serif")
ols_plot_dffits(step_gala)
```



Identifying Influential Observations: Cook's Distance

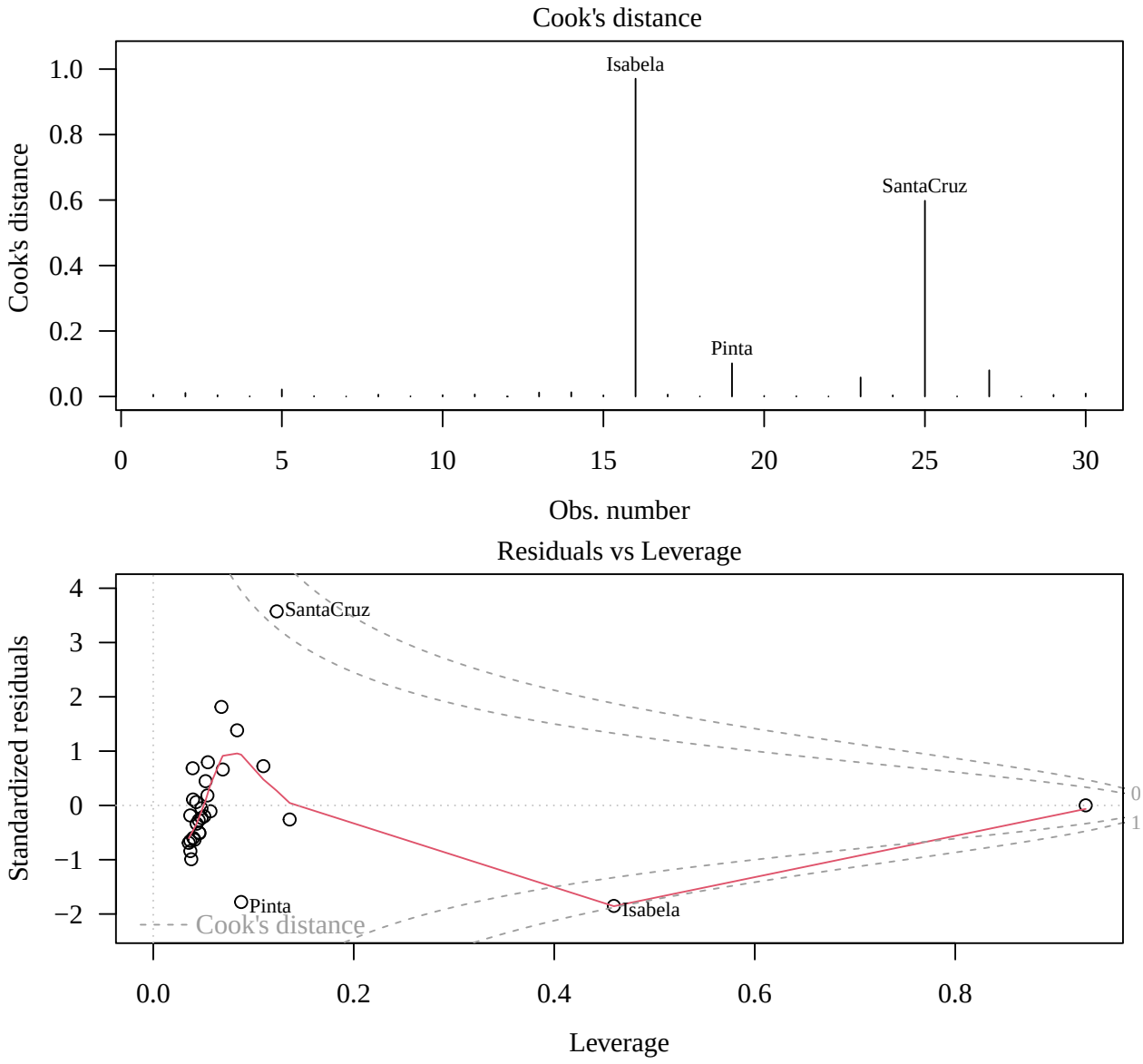
Cook's distance measures the overall influence of each observation on the fitted model.

```
par(las = 1, mar = c(3.5, 3.5, 1, 1), mgp = c(2.5, 1, 0), family = "serif")
```

```r
# Numerical Cook's distance values
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
```

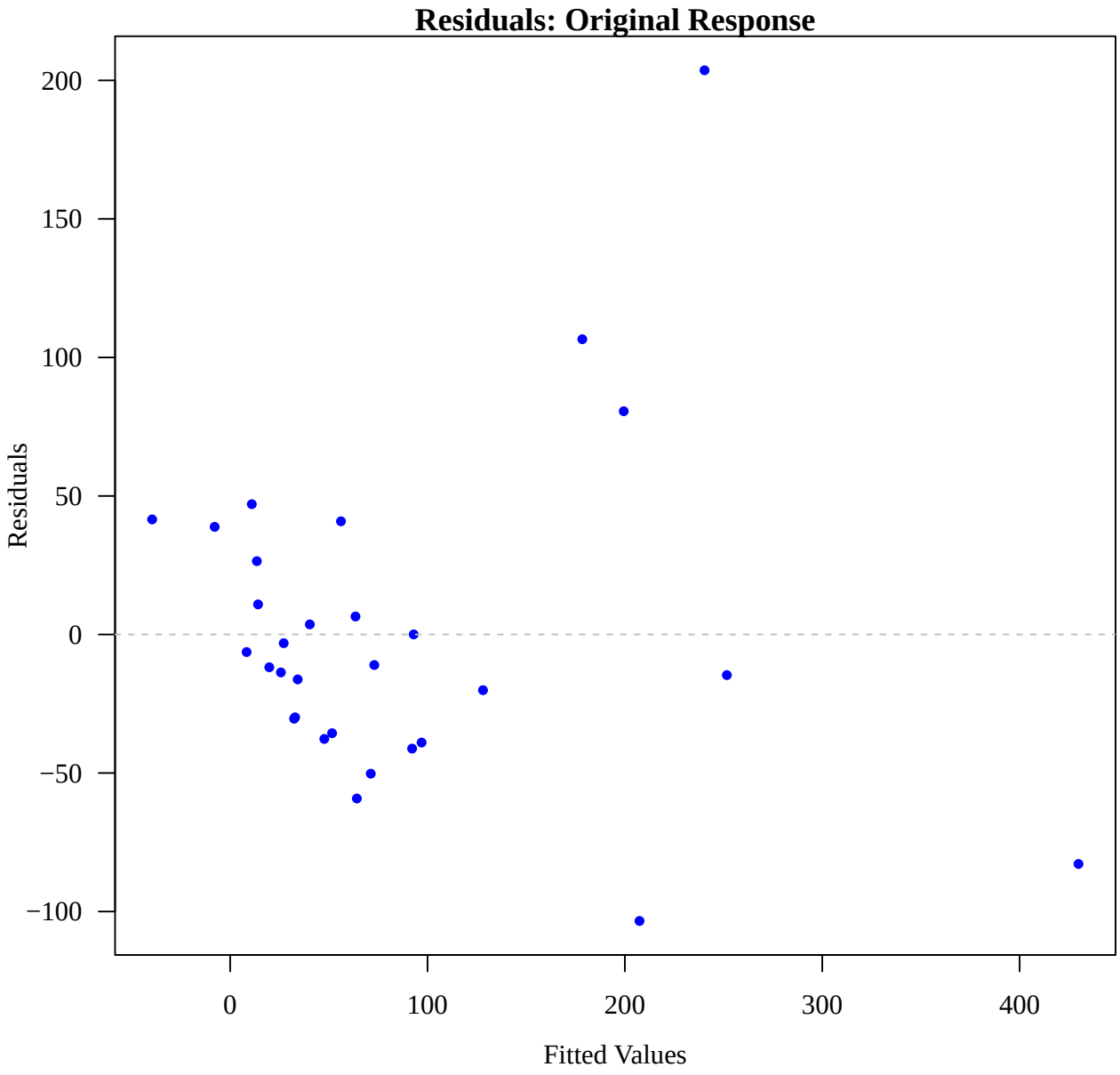
```r
# Diagnostic plots for Cook's distance and leverage
par(mfrow = c(2, 1),
    mar = c(3.8, 3.8, 1.2, 0.5),
    mgp = c(2.5, 1, 0))

plot(step_gala, which = 4:5)
```



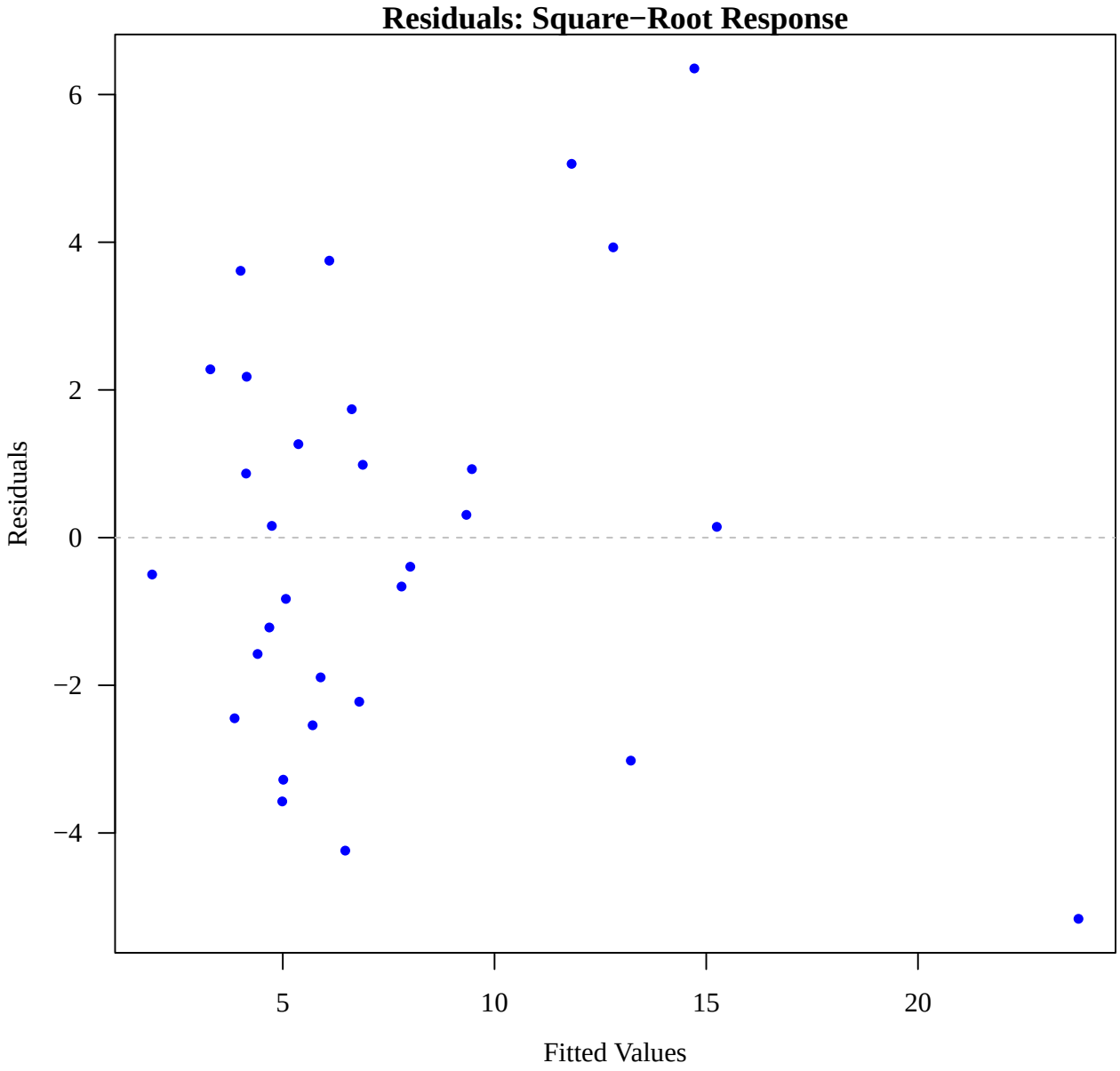
Response Transformation

A response transformation can help stabilize variance and improve residual patterns.

```
par(las = 1, mar = c(3.5, 3.5, 1, 1), mgp = c(2.5, 1, 0), family = "serif")

# Residual plot before transformation
plot(step_gala$fitted.values, step_gala$residuals,
     pch = 16, cex = 0.8, col = "blue",
     xlab = "Fitted Values",
     ylab = "Residuals",
     main = "Residuals: Original Response")
abline(h = 0, lty = 2, col = "gray")
```



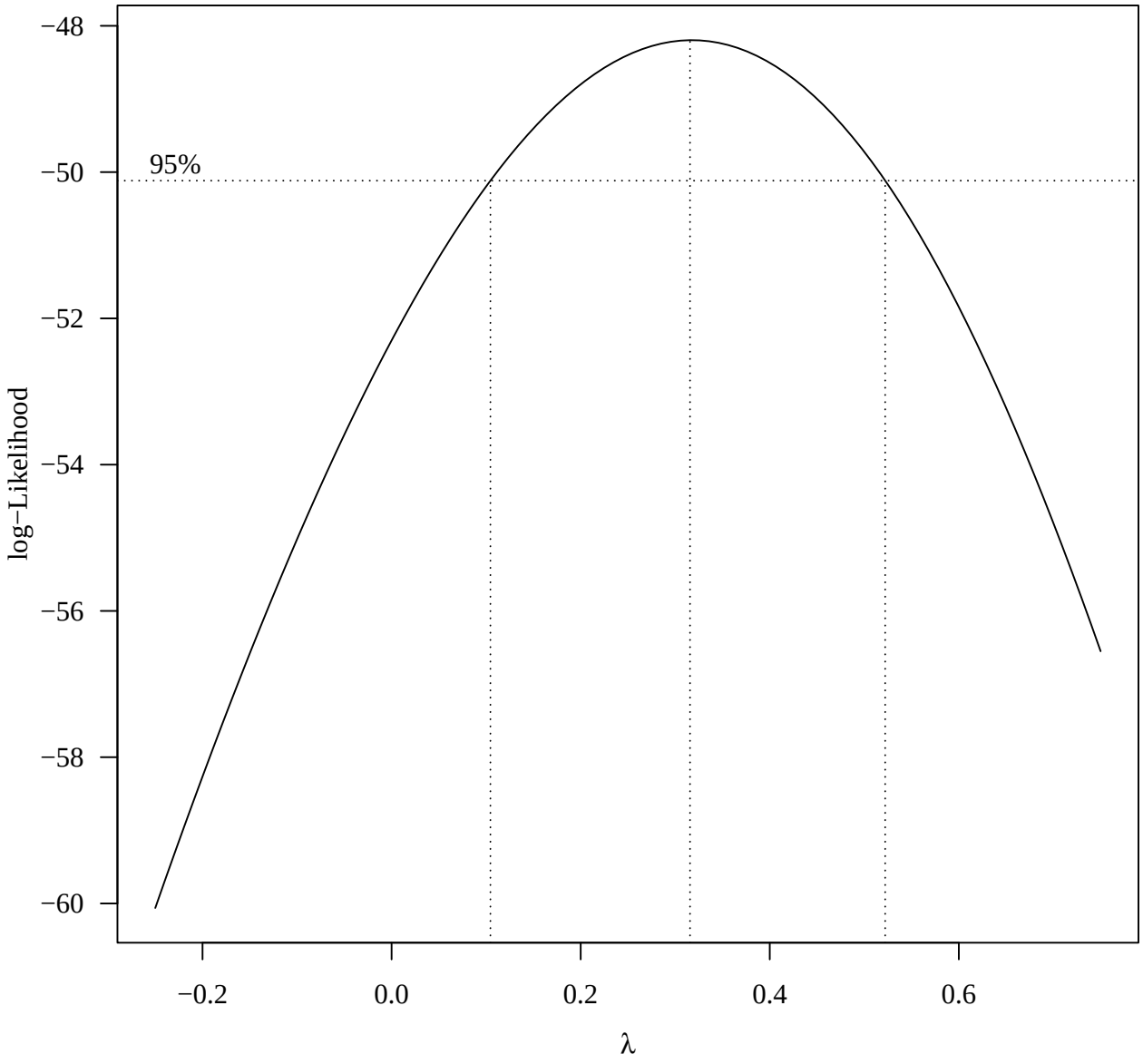

```r
# Fit model with square-root transformed response
sqrt_fit <- lm(sqrt(Species) ~ Elevation + Adjacent, data = galaNew)

# Residual plot after transformation
plot(sqrt_fit$fitted.values, sqrt_fit$residuals,
     pch = 16, cex = 0.8, col = "blue",
     xlab = "Fitted Values",
     ylab = "Residuals",
     main = "Residuals: Square-Root Response")
abline(h = 0, lty = 2, col = "gray")
```



Box-Cox Transformation

Use the Box-Cox plot to identify a suitable power transformation for the response.

```
library(MASS)

# Search over possible power transformations
par(las = 1, mar = c(3.5, 3.5, 1, 1), mgp = c(2.5, 1, 0), family = "serif")
bc <- boxcox(step_gala,
             plotit = TRUE,
             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