# perch

*Chris Parrish*

*January 8, 2016*

perch

reference:
- Cannon, et al., Stat2, chapter 04, examples 4.4, 4.6

Import the data.

```
data <- read.csv("Perch.csv", header=TRUE)
head(data)
```

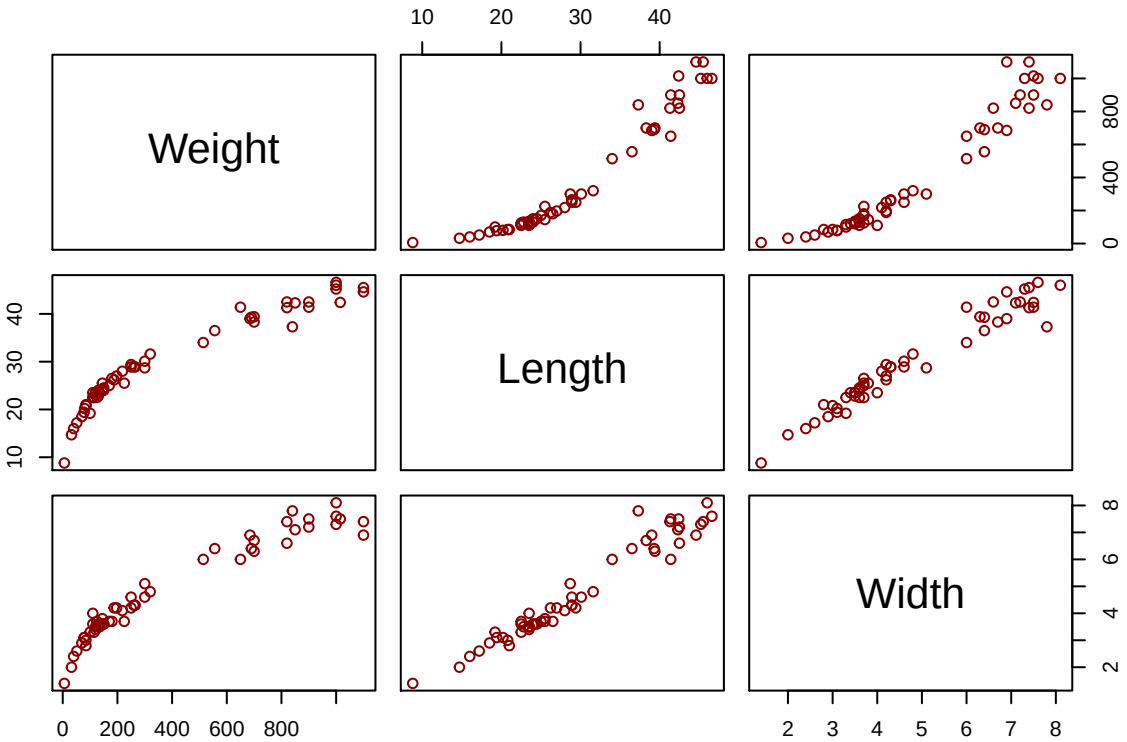
```
##   Obs Weight Length Width
## 1 104    5.9    8.8   1.4
## 2 105   32.0   14.7   2.0
## 3 106   40.0   16.0   2.4
## 4 107   51.5   17.2   2.6
## 5 108   70.0   18.5   2.9
## 6 109  100.0   19.2   3.3
```

```
dim(data)
```

```
## [1] 56  4
```

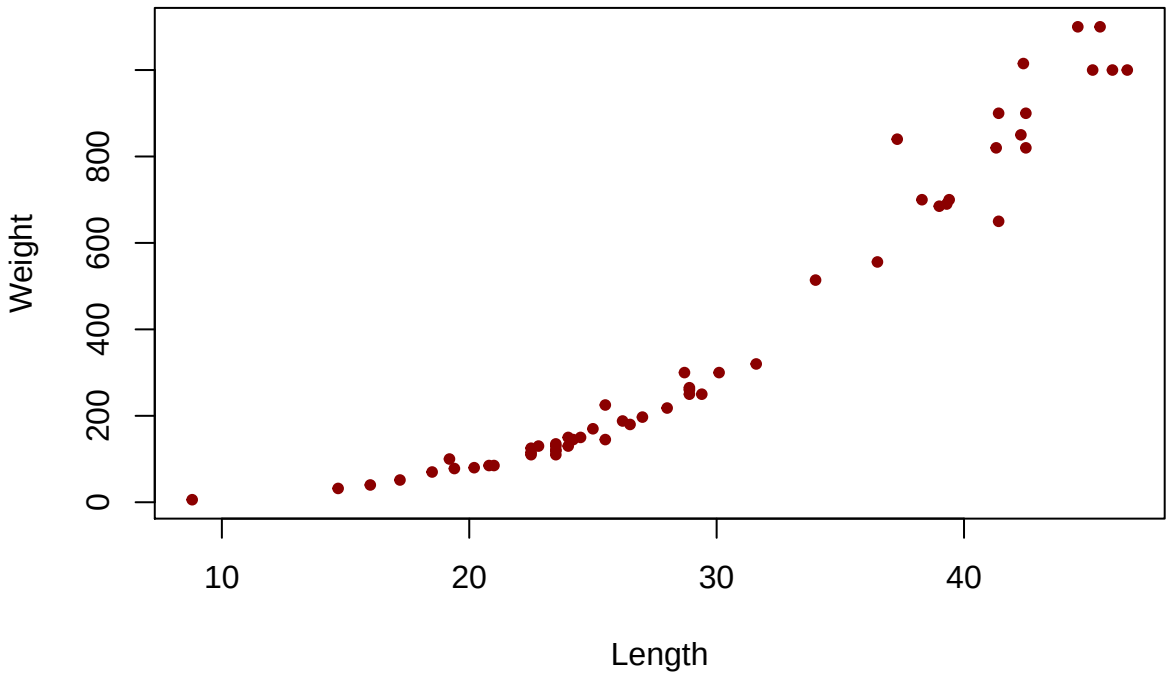
Scatterplot matrix.

```
pairs(~ Weight + Length + Width, data=data, col="darkred")
```



Separate linear models for length and width

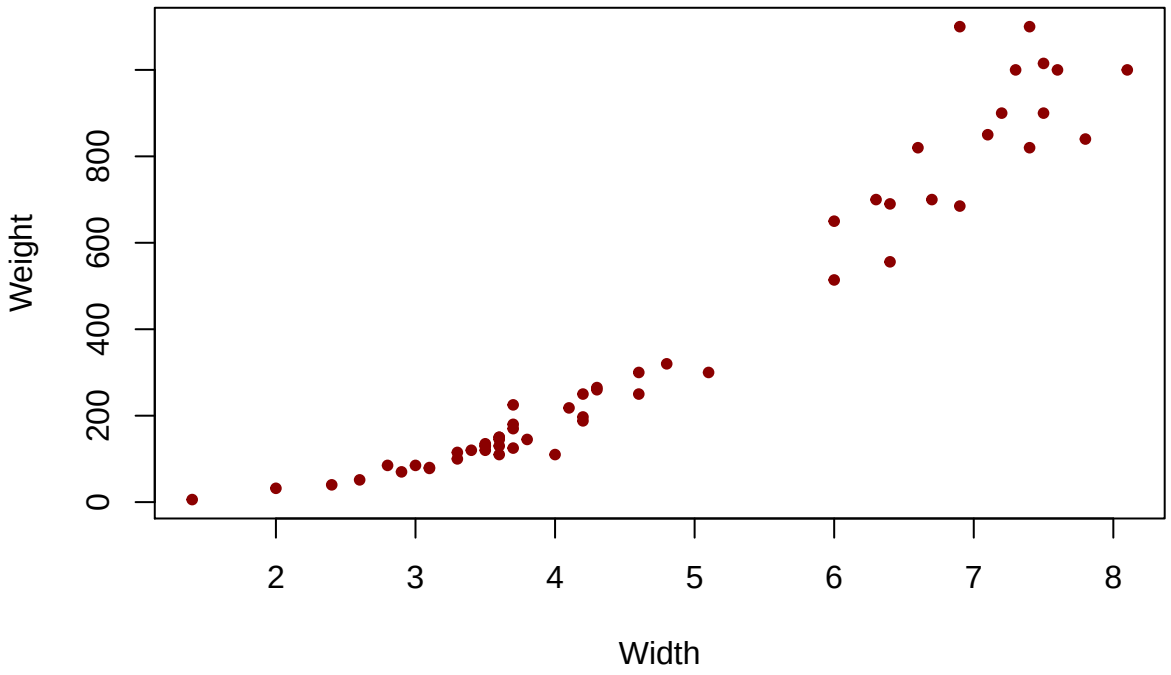
```
# length
plot(Weight ~ Length, data=data,
     pch=20, col="darkred")
```




```
perch.length.lm <- lm(Weight ~ Length, data=data)
coef(perch.length.lm)
```

```
## (Intercept)      Length
##  -652.78714    35.00089
```

```
# width
plot(Weight ~ Width, data=data,
     pch=20, col="darkred")
```




```
perch.width.lm <- lm(Weight ~ Width, data=data)
coef(perch.width.lm)
```

```
## (Intercept)       Width
##   -509.2893    188.1146
```

Multiple regression with interaction.

```
perch.lm1 <- lm(Weight ~ Length * Width, data=data)
options(show.signif.stars=FALSE)
summary(perch.lm1)
```

```
##
## Call:
## lm(formula = Weight ~ Length * Width, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -140.106  -12.226    1.230    8.489  181.408
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  113.9349    58.7844   1.938    0.058
## Length        -3.4827     3.1521  -1.105    0.274
## Width        -94.6309    22.2954  -4.244 9.06e-05
## Length:Width   5.2412     0.4131  12.687  < 2e-16
##
## Residual standard error: 44.24 on 52 degrees of freedom
## Multiple R-squared:  0.9847, Adjusted R-squared:  0.9838
## F-statistic:  1115 on 3 and 52 DF,  p-value: < 2.2e-16
```

Multiple regression without interaction.

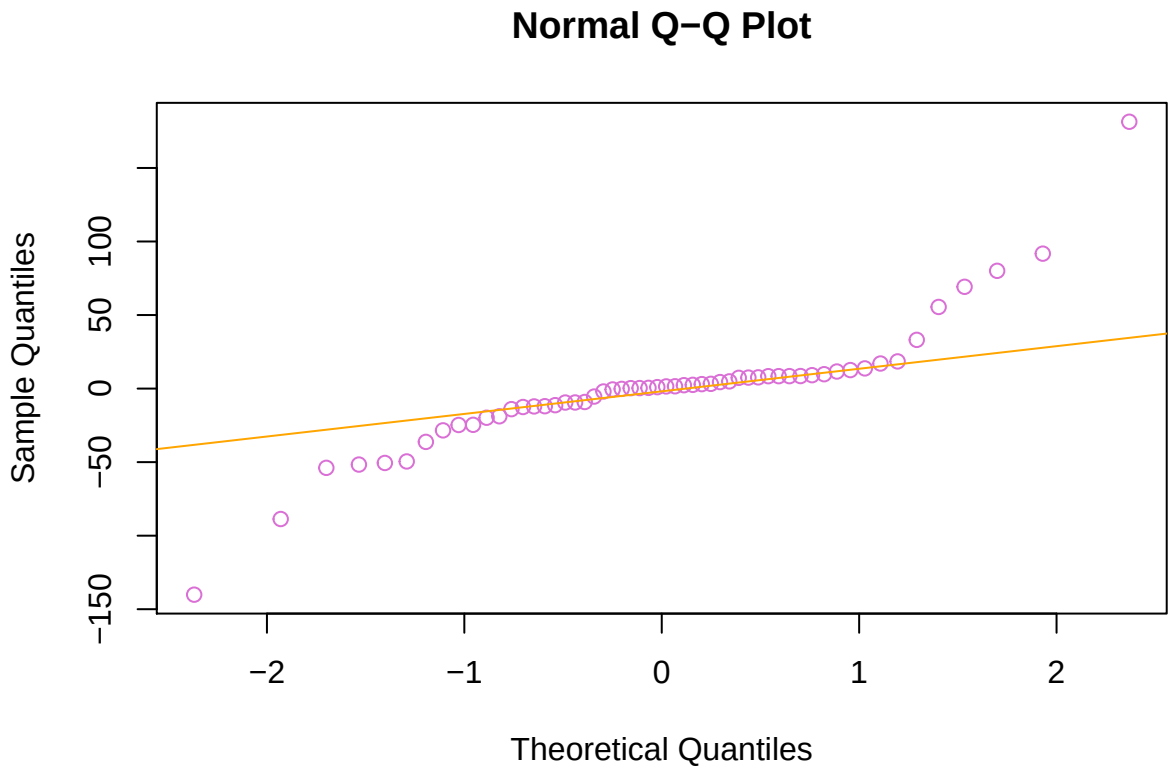
```
perch.lm2 <- lm(Weight ~ Length + Width, data=data)
summary(perch.lm2)
```

```
##
## Call:
## lm(formula = Weight ~ Length + Width, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -113.86  -59.02  -23.29   30.93  299.85
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -578.758     43.667 -13.254  < 2e-16
## Length        14.307      5.659   2.528 0.014475
## Width        113.500     30.265   3.750 0.000439
##
## Residual standard error: 88.68 on 53 degrees of freedom
## Multiple R-squared:  0.9373, Adjusted R-squared:  0.9349
## F-statistic: 396.1 on 2 and 53 DF,  p-value: < 2.2e-16
```

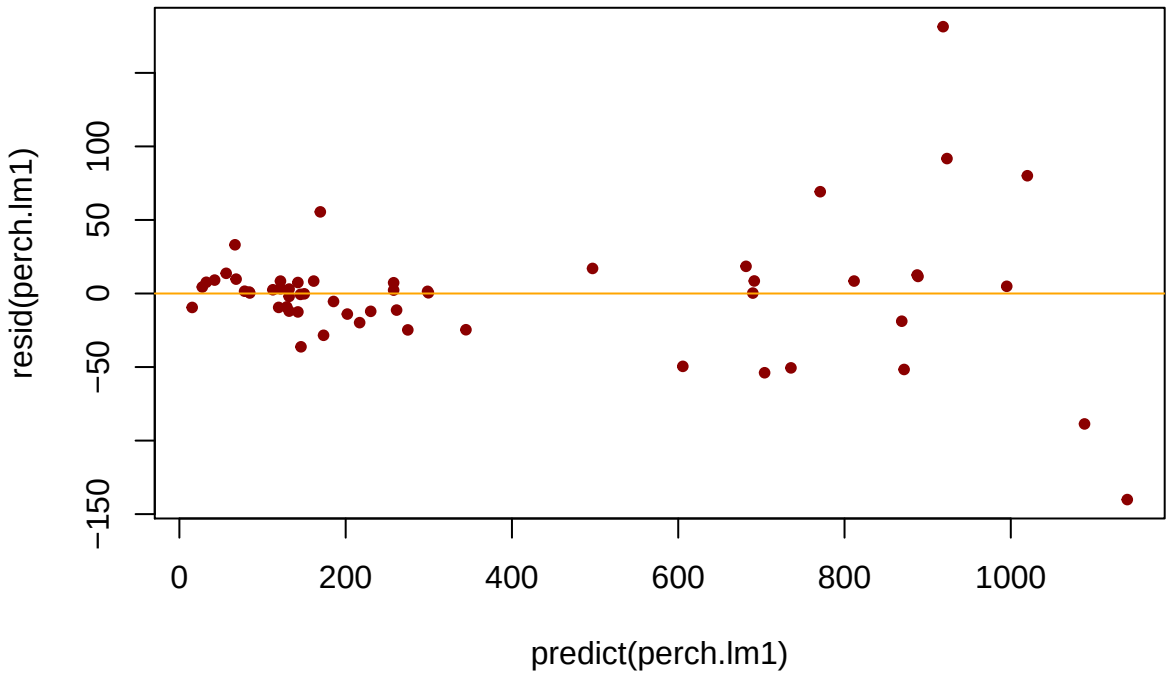
Residuals.

```
# with interaction
hist(resid(perch.lm1),
     col="wheat")
```

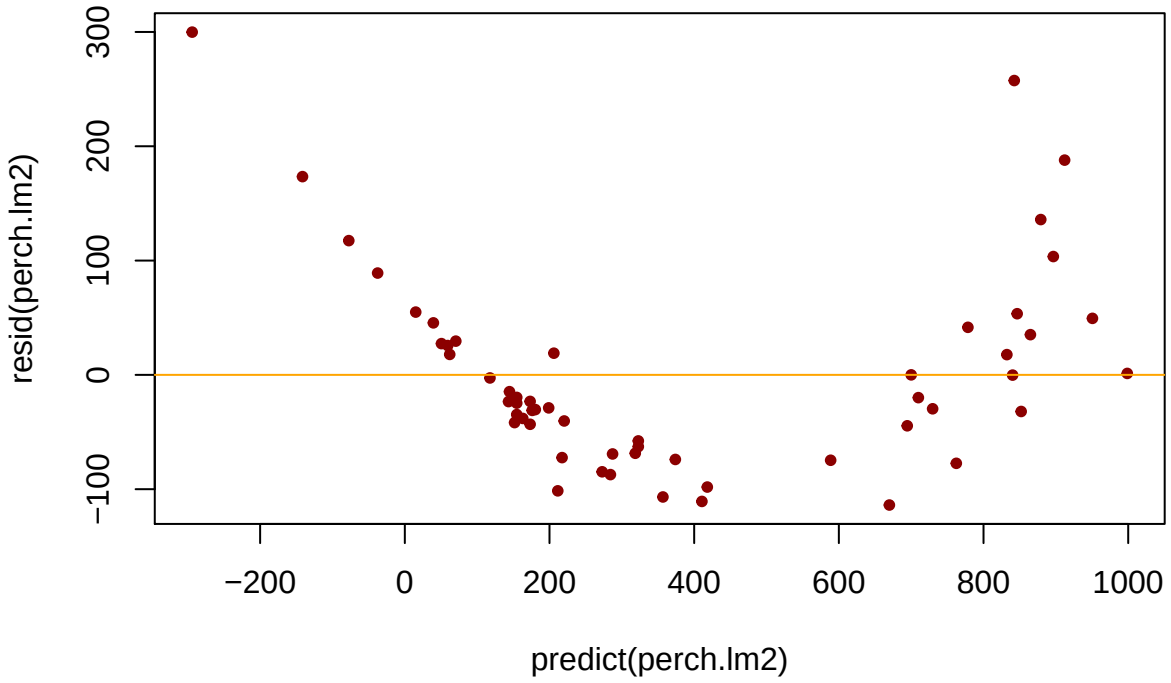
```r
qqnorm(resid(perch.lm1),
       col="orchid")
qqline(resid(perch.lm1),
       col="orange")
```




```r
plot(predict(perch.lm1), resid(perch.lm1),
     pch=20, col="darkred")
abline(h=0, col="orange")
```




```
# without interaction
plot(predict(perch.lm2), resid(perch.lm2),
     pch=20, col="darkred")
abline(h=0, col="orange")
```




Complete second-order model.

```
perch.lm3 <- lm(Weight ~ Length + Width + I(Length^2) + I(Width^2) + Length:Width, data=data)
summary(perch.lm3)
```

```
##
## Call:
## lm(formula = Weight ~ Length + Width + I(Length^2) + I(Width^2) +
##     Length:Width, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -117.175  -11.904    2.822   11.556  157.596
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  156.3486    61.4152   2.546   0.0140
## Length       -25.0007    14.2729  -1.752   0.0860
## Width         20.9772    82.5877   0.254   0.8005
## I(Length^2)    1.5719     0.7244   2.170   0.0348
## I(Width^2)    34.4058    18.7455   1.835   0.0724
## Length:Width  -9.7763     7.1455  -1.368   0.1774
##
## Residual standard error: 43.13 on 50 degrees of freedom
## Multiple R-squared:  0.986,  Adjusted R-squared:  0.9846
## F-statistic: 704.6 on 5 and 50 DF,  p-value: < 2.2e-16
```

```
anova(perch.lm3)
```

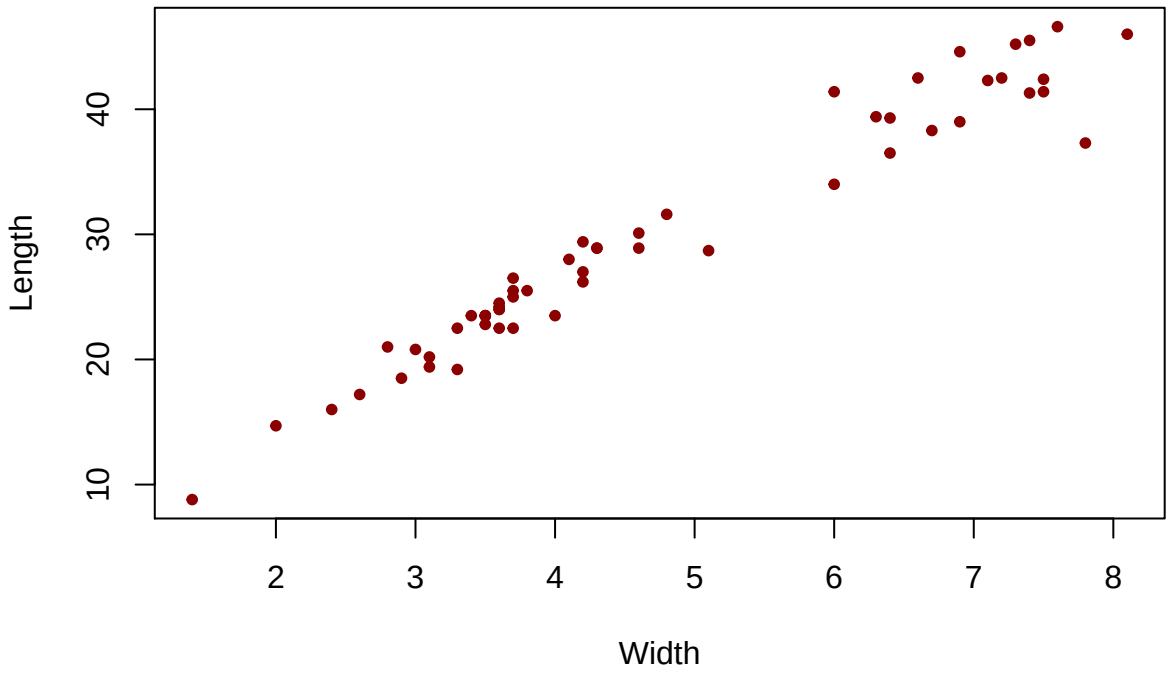
```
## Analysis of Variance Table
##
## Response: Weight
##              Df  Sum Sq Mean Sq   F value    Pr(>F)
## Length        1 6118739 6118739 3289.6413 < 2.2e-16
## Width         1  110593  110593   59.4585 4.667e-10
## I(Length^2)   1  314899  314899  169.3002 < 2.2e-16
## I(Width^2)    1    5381    5381    2.8932   0.09517
## Length:Width  1    3482    3482    1.8719   0.17737
## Residuals    50   93000    1860
```

Correlated predictors.

```
with(data,
     cor(cbind(Weight, Length, Width)))
```

```
##           Weight    Length     Width
## Weight 1.0000000 0.9595061 0.9642244
## Length 0.9595061 1.0000000 0.9751074
## Width  0.9642244 0.9751074 1.0000000
```

```
plot(Length ~ Width, data=data,
     pch=20, col="darkred")
```




```
perch.lm4 <- lm(Weight ~ Length + Width, data=data)
summary(perch.lm4)
```

```
##
## Call:
## lm(formula = Weight ~ Length + Width, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -113.86  -59.02  -23.29   30.93  299.85
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -578.758     43.667 -13.254  < 2e-16
## Length        14.307      5.659   2.528 0.014475
## Width        113.500     30.265   3.750 0.000439
##
## Residual standard error: 88.68 on 53 degrees of freedom
## Multiple R-squared:  0.9373, Adjusted R-squared:  0.9349
## F-statistic: 396.1 on 2 and 53 DF,  p-value: < 2.2e-16
```

```
perch.lm5 <- lm(Weight ~ Length + Width + Length:Width, data=data)
summary(perch.lm5)
```

```
##
## Call:
## lm(formula = Weight ~ Length + Width + Length:Width, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -140.106  -12.226    1.230    8.489  181.408
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  113.9349    58.7844   1.938    0.058
## Length        -3.4827     3.1521  -1.105    0.274
## Width        -94.6309    22.2954  -4.244 9.06e-05
## Length:Width   5.2412     0.4131  12.687  < 2e-16
##
## Residual standard error: 44.24 on 52 degrees of freedom
## Multiple R-squared:  0.9847, Adjusted R-squared:  0.9838
## F-statistic:  1115 on 3 and 52 DF,  p-value: < 2.2e-16
```

Collinearity.

```
cor.matrix <- with(data,
                   cor(cbind(Length, Width, Length^2, Width^2, Length * Width)))
var.names <- c("Length", "Width", "Length^2", "Width^2", "Length*Width")
rownames(cor.matrix) <- colnames(cor.matrix) <- var.names
cor.matrix
```

```
##                 Length     Width  Length^2   Width^2 Length*Width
## Length       1.0000000 0.9751074 0.9888600 0.9523922    0.9785746
## Width        0.9751074 1.0000000 0.9680791 0.9896026    0.9878073
## Length^2     0.9888600 0.9680791 1.0000000 0.9644068    0.9905755
## Width^2      0.9523922 0.9896026 0.9644068 1.0000000    0.9914553
## Length*Width 0.9785746 0.9878073 0.9905755 0.9914553    1.0000000
```

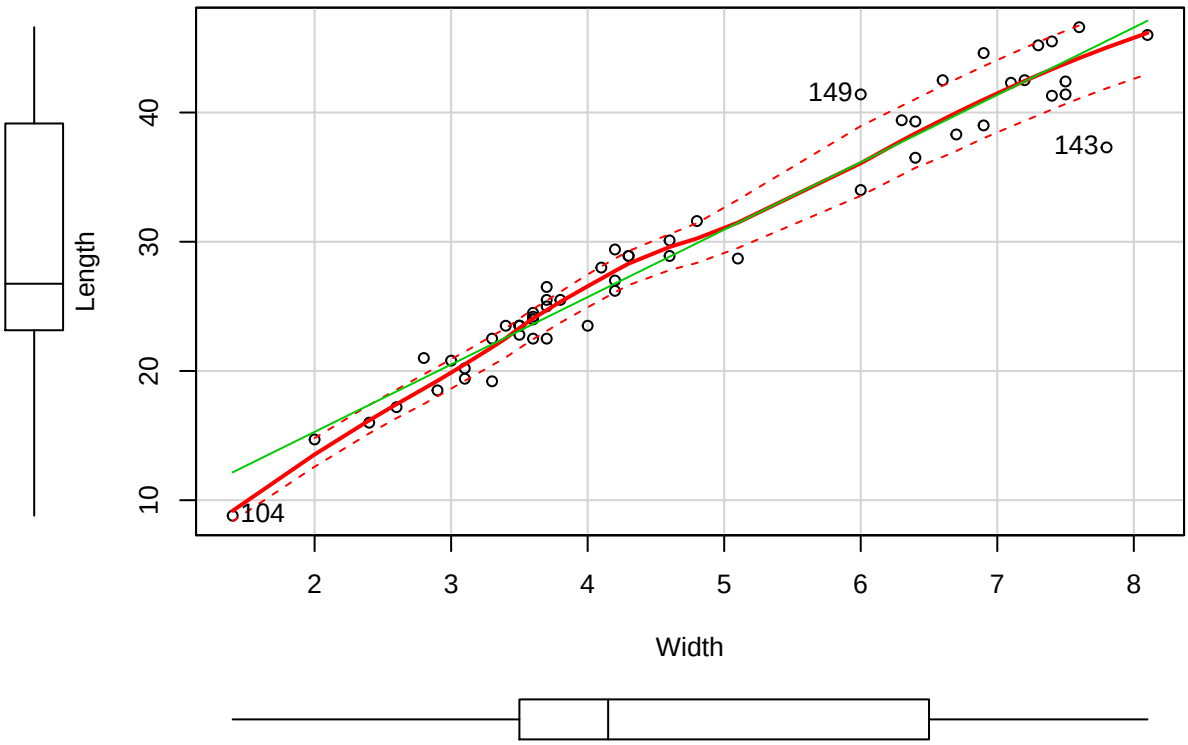
Nested F-tests.

```
#  perch.lm5 <- lm(Weight ~ Length + Width + Length:Width, data=data)
#  perch.lm3 <- lm(Weight ~ Length + Width + I(Length^2) + I(Width^2) + Length:Width, data=data)
anova(perch.lm5, perch.lm3)
```

```
## Analysis of Variance Table
##
## Model 1: Weight ~ Length + Width + Length:Width
## Model 2: Weight ~ Length + Width + I(Length^2) + I(Width^2) + Length:Width
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     52 101765
## 2     50  93000  2    8764.6 2.3561 0.1052
```

Identify unusual points.

```
library(car)
scatterplot(Length ~ Width, data=data,
            id.n=3, labels=data$Obs)
```




```
## 104 143 149
##   1  40  46
```

```
perch.diag <- ls.diag(perch.lm1)        # Weight ~ Length * Width, data=data
summary(perch.diag)
```

```
##              Length Class  Mode
## std.dev       1     -none- numeric
## hat          56     -none- numeric
## std.res      56     -none- numeric
## stud.res     56     -none- numeric
## cooks        56     -none- numeric
## dfits        56     -none- numeric
## correlation  16     -none- numeric
## std.err       4     -none- numeric
## cov.scaled   16     -none- numeric
## cov.unscaled 16     -none- numeric
```

Leverage.

```
n <- nrow(data)
k <- 3              # k = number of regressors
typical.leverage <- (k + 1) / n
hi.threshold <- 3 * typical.leverage
data[perch.diag$hat > hi.threshold, ]
```

```
##    Obs Weight Length Width
## 1  104    5.9    8.8   1.4
## 40 143  840.0   37.3   7.8
```

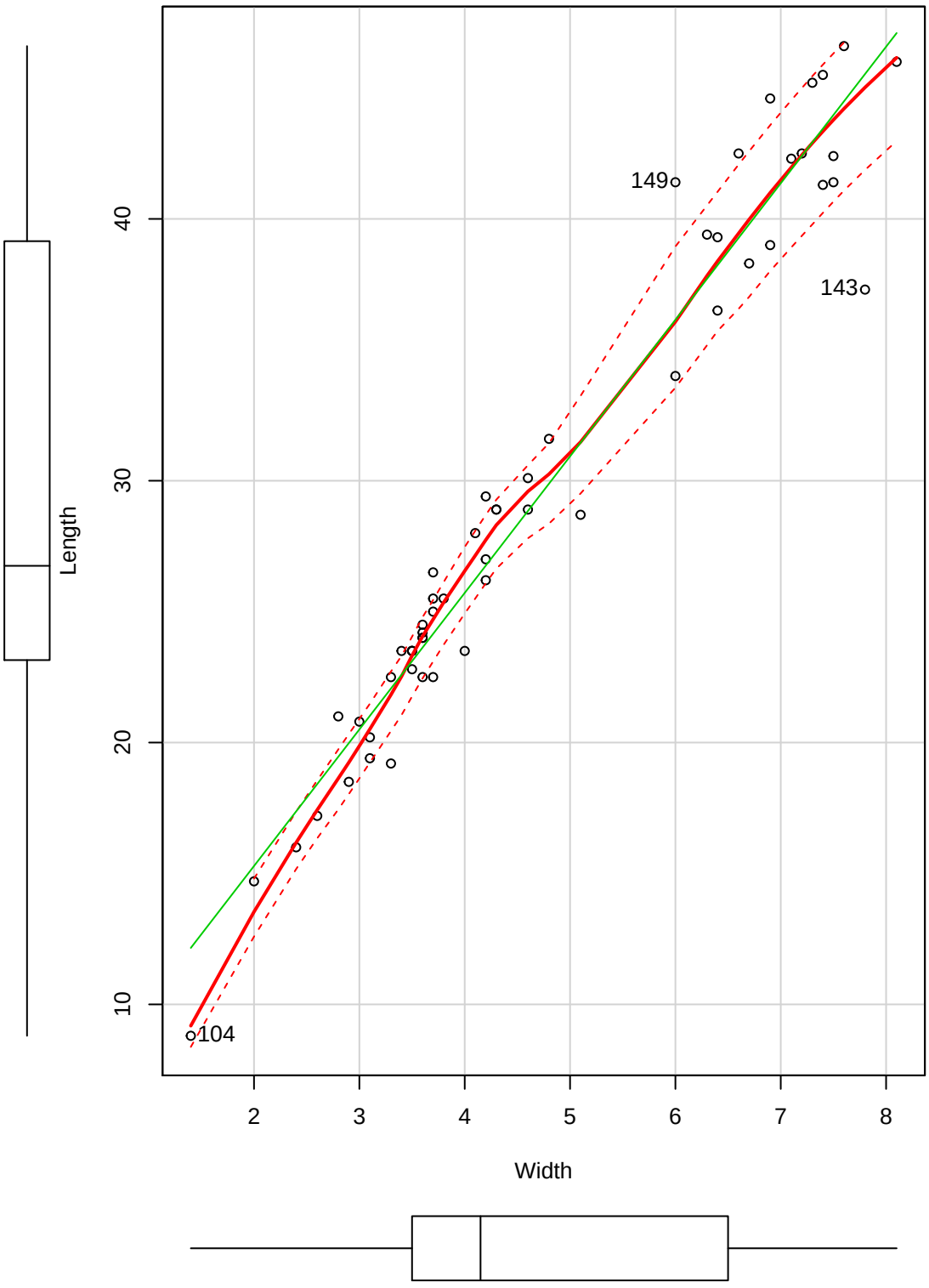
Standardized residuals.

```
hi.threshold <- 3 * typical.leverage
std.res.threshold <- 2
idx <- (1:n)[perch.diag$hat > hi.threshold |
             abs(perch.diag$std.res) > std.res.threshold]
results <- cbind(data[idx, ],
                 perch.lm1$fit[idx],
                 perch.diag$hat[idx],
                 perch.lm1$resid[idx],
                 perch.diag$std.res[idx])
names(results) <- c("Obs", "Weight", "Length", "Width", "Fit", "Hat", "Resid", "Std.Res")
results
```

```
##    Obs Weight Length Width       Fit        Hat       Resid   Std.Res
## 1  104    5.9    8.8   1.4   15.3760 0.43136204   -9.475996 -0.284060
## 40 143  840.0   37.3   7.8  770.7952 0.36285955   69.204835  1.959846
## 50 153 1015.0   42.4   7.5  923.2506 0.07771097   91.749402  2.159598
## 52 155 1100.0   44.6   6.9  918.5921 0.12055293  181.407883  4.372748
## 55 158 1000.0   46.0   8.1 1140.1059 0.15090161 -140.105946 -3.437009
## 56 159 1000.0   46.6   7.6 1088.6834 0.14340985  -88.683357 -2.166002
```

Identify unusual points.

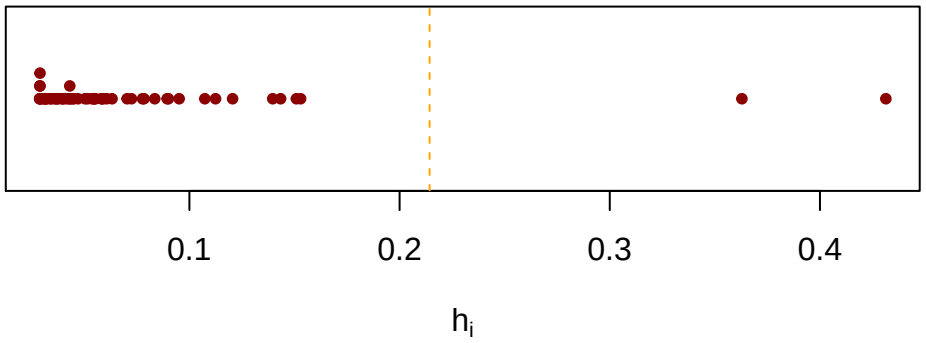
```
library(car)
scatterplot(Length ~ Width, data=data,
            id.n=3, labels=data$Obs)
```




```
## 104 143 149
##   1  40  46
```

Dot plot of $h_i$ values.

```
stripchart(perch.diag$hat,
           pch=20, col="darkred",
           method="stack", xlab=expression(h[i]))
abline(v=hi.threshold, col="orange", lty="dashed")
```




Cook's distance.

```
hi.threshold <- 3 * typical.leverage
std.res.threshold <- 2
idx <- (1:n)[perch.diag$hat > hi.threshold |
             abs(perch.diag$std.res) > std.res.threshold]
results <- cbind(data[idx, ],
                 perch.diag$std.res[idx],
                 perch.diag$hat[idx],
                 perch.diag$cooks[idx])
names(results) <- c("Obs", "Weight", "Length", "Width", "Std.Res", "Hat", "Cook's D")
results
```

```
##    Obs Weight Length Width   Std.Res        Hat   Cook's D
## 1  104    5.9    8.8   1.4 -0.284060 0.43136204 0.01530264
## 40 143  840.0   37.3   7.8  1.959846 0.36285955 0.54687418
## 50 153 1015.0   42.4   7.5  2.159598 0.07771097 0.09824288
## 52 155 1100.0   44.6   6.9  4.372748 0.12055293 0.65526509
## 55 158 1000.0   46.0   8.1 -3.437009 0.15090161 0.52485254
## 56 159 1000.0   46.6   7.6 -2.166002 0.14340985 0.19636471
```

```
plot(perch.diag$hat, perch.diag$std.res,
     pch=20, col="darkred", ylim=c(-4.5,4.5),
     xlab="Leverage", ylab="Standardized residuals")
abline(h=c(-3, -2, 2, 3), col="orange", lty="dashed")
abline(v=c(-3, -2, 2, 3) * typical.leverage, col="orange", lty="dashed")
cook <- function(pm, D, h){
  k <- 3   # number of predictors
  val <- pm * sqrt(D * (k + 1) * (1 - h) / h)
  return(val)
}
curve(cook(1, 1, x), from=0.05, to=0.45, col="orange", lty="dashed", add=TRUE)
curve(cook(1, 0.5, x), from=0.05, to=0.45, col="orange", lty="dashed", add=TRUE)
curve(cook(-1, 0.5, x), from=0.05, to=0.45, col="orange", lty="dashed", add=TRUE)
curve(cook(-1, 1, x), from=0.05, to=0.45, col="orange", lty="dashed", add=TRUE)
text(x=perch.diag$hat[40], perch.diag$std.res[40], labels=c("40"), pos=2)
```

```
text(x=perch.diag$hat[52], perch.diag$std.res[52], labels=c("52"), pos=2)
text(x=perch.diag$hat[55], perch.diag$std.res[55], labels=c("55"), pos=2)
```