Multiple Linear Regression

Karen Mazidi

ChickWeight is a built-in R data set with 578 rows and 4 columns of data resulting from an experiment on the effect of different types of feed on chick weight. Each observation (row) in the data set represents the weight in grams of a given chick on a given day, recorded in column Time.

Data exploration

Let's explore the data with R functions and plots.

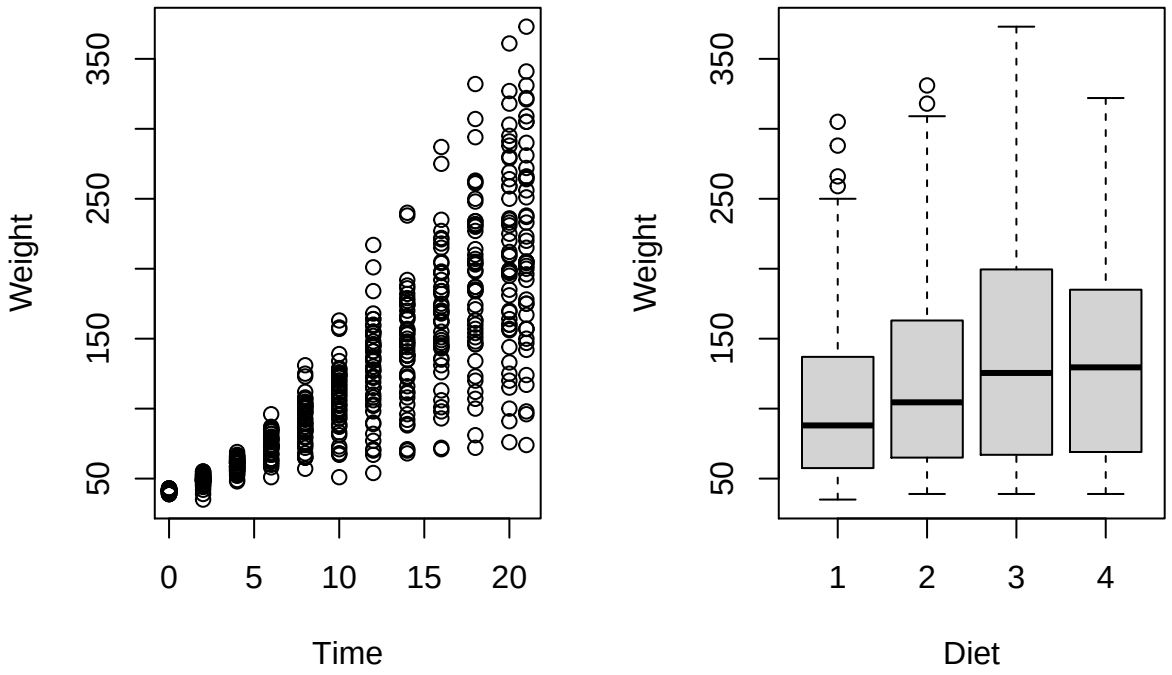
```
data(ChickWeight)
dim(ChickWeight)
```

```
## [1] 578   4
```

```
head(ChickWeight)
```

```
##   weight Time Chick Diet
## 1     42    0     1    1
## 2     51    2     1    1
## 3     59    4     1    1
## 4     64    6     1    1
## 5     76    8     1    1
## 6     93   10     1    1
```

```
par(mfrow=c(1,2))
plot(ChickWeight$Time, ChickWeight$weight,
     xlab="Time", ylab="Weight")
plot(ChickWeight$Diet, ChickWeight$weight,
     xlab="Diet", ylab="Weight")
```



Divide the data into train and test sets

We randomly sample the rows to get a vector i with row indices. This is used to divide into train and test sets.

```
set.seed(1234)
i <- sample(1:nrow(ChickWeight), nrow(ChickWeight)*0.75, replace=FALSE)
train <- ChickWeight[i,]
test <- ChickWeight[-i,]
```

Simple linear regression

In simple linear regression we have a single predictor variable for our target variable. Here we wish to see the impact of Time on weight.

```
lm1 <- lm(weight~Time, data=train)
summary(lm1)
```

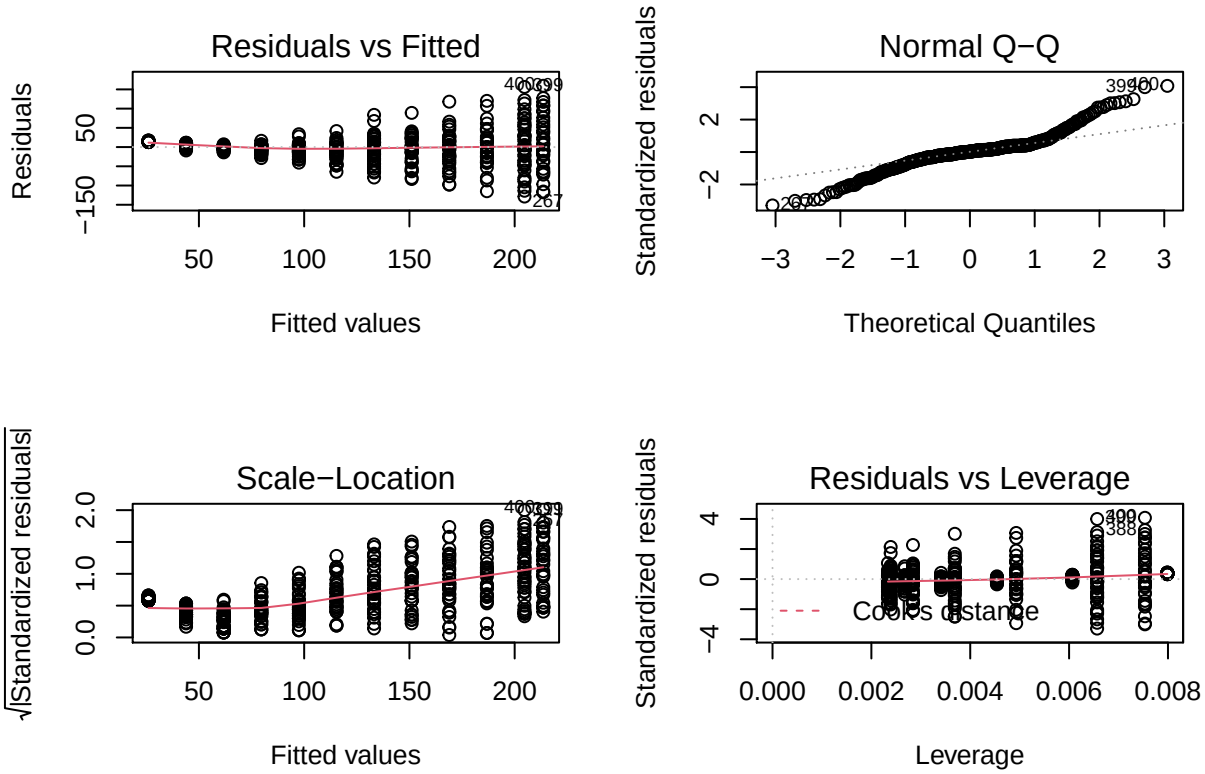
```
##
## Call:
## lm(formula = weight ~ Time, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -128.669  -13.765    1.098   14.961  159.400
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  26.0395     3.5066   7.426 6.05e-13 ***
## Time          8.9315     0.2758  32.380  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.24 on 431 degrees of freedom
```

```
## Multiple R-squared:  0.7087, Adjusted R-squared:  0.708
## F-statistic:  1048 on 1 and 431 DF,  p-value: < 2.2e-16
```

Plotting the residuals

The 4 residual plots are placed in a 2x2 grid.

```
par(mfrow=c(2,2))
plot(lm1)
```



Evaluate on the test set

```
pred1 <- predict(lm1, newdata=test)
cor1 <- cor(pred1, test$weight)
mse1 <- mean((pred1-test$weight)^2)
rmse1 <- sqrt(mse1)

print(paste('correlation:', cor1))
```

```
## [1] "correlation: 0.821351767209117"
```

```
print(paste('mse:', mse1))
```

```
## [1] "mse: 1442.35619965639"
```

```
print(paste('rmse:', rmse1))
```

```
## [1] "rmse: 37.978364889189"
```

Multiple Linear Regression

If we have more than one predictor in linear regression we call it multiple linear regression. Here we want to see the effect of both Time and Diet on chick weight.

```
lm2 <- lm(weight~Time+Diet, data=train)
summary(lm2)
```

```
##
## Call:
## lm(formula = weight ~ Time + Diet, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -127.331  -17.051   -2.445   14.530  141.590
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.7267     3.9279   2.476 0.013660 *
## Time          8.8718     0.2569  34.531  < 2e-16 ***
## Diet2        16.1681     4.7374   3.413 0.000704 ***
## Diet3        35.3745     4.9122   7.201 2.70e-12 ***
## Diet4        30.6332     4.7539   6.444 3.14e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.53 on 428 degrees of freedom
## Multiple R-squared:  0.7493, Adjusted R-squared:  0.7469
## F-statistic: 319.8 on 4 and 428 DF,  p-value: < 2.2e-16
```

The adjusted R-squared for lm2 is an improvement over lm1.

The anova() function

The analysis of variance function here is used to compare the two models. We see that lm2 lowered the errors, RSS, and had a low p-value. These are indications that lm2 is a better model than lm1.

```
anova(lm1, lm2)
```

```
## Analysis of Variance Table
##
## Model 1: weight ~ Time
## Model 2: weight ~ Time + Diet
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    431 663507
## 2    428 571051  3     92456 23.098 7.081e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Evaluate on the test set

```
pred2 <- predict(lm2, newdata=test)
cor2 <- cor(pred2, test$weight)
mse2 <- mean((pred2-test$weight)^2)
rmse2 <- sqrt(mse2)

print(paste('correlation:', cor2))
```

```
## [1] "correlation: 0.855794849833585"
```

```r
print(paste('mse:', mse2))
```

```
## [1] "mse: 1185.00650070072"
```

```r
print(paste('rmse:', rmse2))
```

```
## [1] "rmse: 34.4239233775106"
```

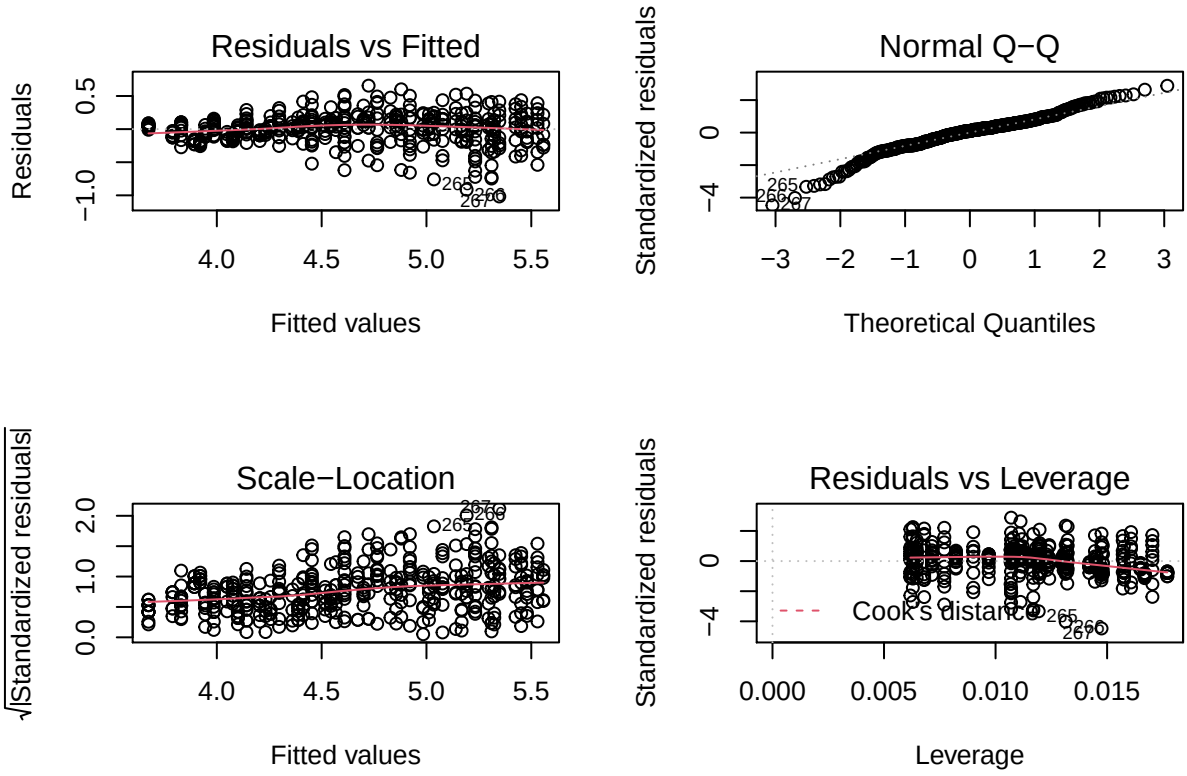
Linear models are not always straight lines

Next we try predicting the log of weight to illustrate that linear models are not always straight lines. This damped down some of the variation in the residuals. The lm3 model had a higher R-squared of 0.8474. We cannot do anova() comparing lm3 because it has a different target, the log(weight) instead of weight.

```r
lm3 <- lm(log(weight)~Time+Diet, data=train)
summary(lm3)
```

```
##
## Call:
## lm(formula = log(weight) ~ Time + Diet, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.01662 -0.12515  0.02366  0.12528  0.65600
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.673863   0.024623 149.203  < 2e-16 ***
## Time        0.077932   0.001611  48.387  < 2e-16 ***
## Diet2       0.114844   0.029698   3.867 0.000127 ***
## Diet3       0.219280   0.030794   7.121 4.56e-12 ***
## Diet4       0.246804   0.029801   8.282 1.57e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.229 on 428 degrees of freedom
## Multiple R-squared:  0.8518, Adjusted R-squared:  0.8504
## F-statistic: 614.8 on 4 and 428 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(lm3)
```



Evaluate on the test set

```
pred3 <- predict(lm3, newdata=test)
pred3 <- exp(pred3)
cor3 <- cor(pred3, test$weight)
mse3 <- mean((pred3-test$weight)^2)
rmse3 <- sqrt(mse3)

print(paste('correlation:', cor3))
```

```
## [1] "correlation: 0.849406031433899"
```

```
print(paste('mse:', mse3))
```

```
## [1] "mse: 1258.92115603879"
```

```
print(paste('rmse:', rmse3))
```

```
## [1] "rmse: 35.4812789515653"
```

Note that we can't do an anova comparison with model 3 because it has a target of log(weight) and lm1 and lm2 have weight as a target.