# 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.

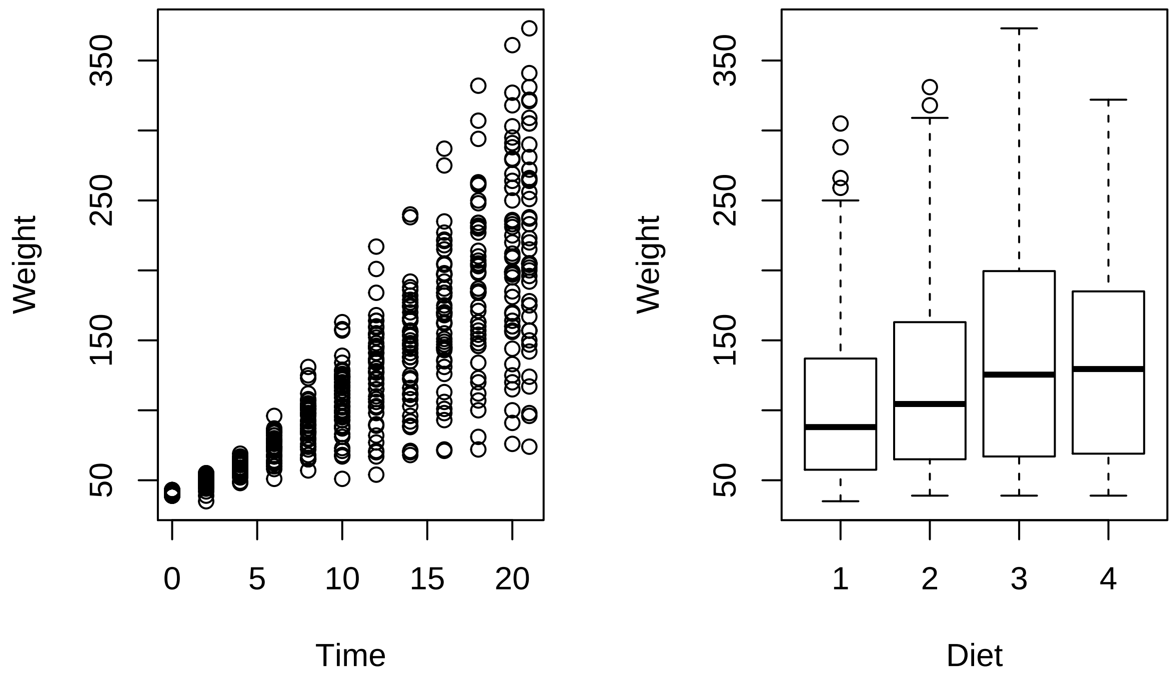
```
data(ChickWeight)
str(ChickWeight)
```

```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':   578 obs. of  4 variables
##  $ weight: num  42 51 59 64 76 93 106 125 149 171 ...
##  $ Time  : num  0 2 4 6 8 10 12 14 16 18 ...
##  $ Chick : Ord.factor w/ 50 levels "18"<"16"<"15"<..: 15 15 15 15 15 15 15 15 15 15 ...
##  $ Diet  : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "formula")=Class 'formula'  language weight ~ Time | Chick
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "outer")=Class 'formula'  language ~Diet
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "labels")=List of 2
##   ..$ x: chr "Time"
##   ..$ y: chr "Body weight"
##  - attr(*, "units")=List of 2
##   ..$ x: chr "(days)"
##   ..$ y: chr "(gm)"
```

```
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)
```

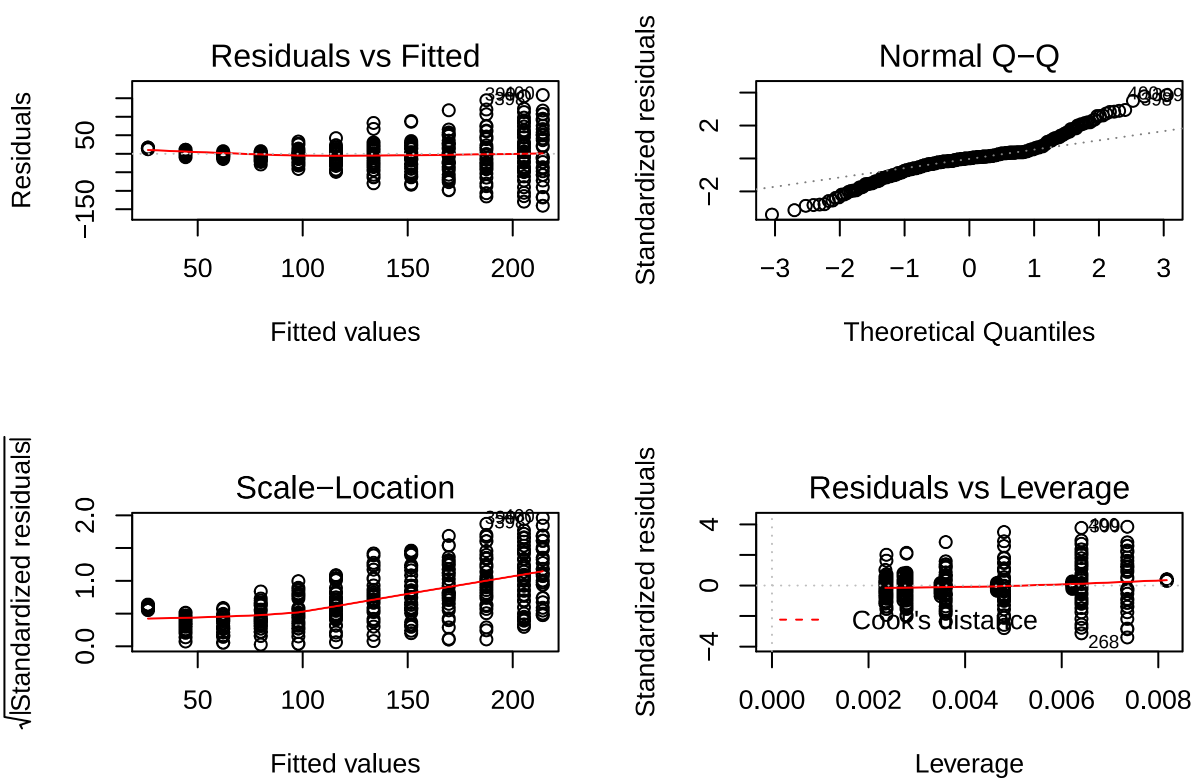
```
##
## Call:
## lm(formula = weight ~ Time, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -140.314  -16.648    0.778   14.682  158.686
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   26.318      3.743   7.031 8.06e-12 ***
## Time           8.952      0.291  30.760  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41.4 on 431 degrees of freedom
## Multiple R-squared:  0.687,  Adjusted R-squared:  0.6863
## F-statistic: 946.2 on 1 and 431 DF,  p-value: < 2.2e-16
```

### Plotting the residuals

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

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




### 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.

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

```
##
## Call:
## lm(formula = weight ~ Time + Diet, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -137.857  -20.492   -1.685   16.955  137.365
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.4109     4.1372   2.033 0.042670 *
## Time          8.9086     0.2682  33.218  < 2e-16 ***
## Diet2        16.3645     4.9235   3.324 0.000965 ***
## Diet3        40.1424     4.8907   8.208 2.67e-15 ***
## Diet4        32.1873     5.2503   6.131 1.99e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.13 on 428 degrees of freedom
## Multiple R-squared:  0.7363, Adjusted R-squared:  0.7338
## F-statistic: 298.8 on 4 and 428 DF,  p-value: < 2.2e-16
```

The adjusted R-squared for lm2 is 0.7338, which is an improvement of lm1's 0.6863.

### 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.

```r
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 738546
## 2    428 622323  3    116222 26.644 8.107e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

### Linear models are not always straight lines

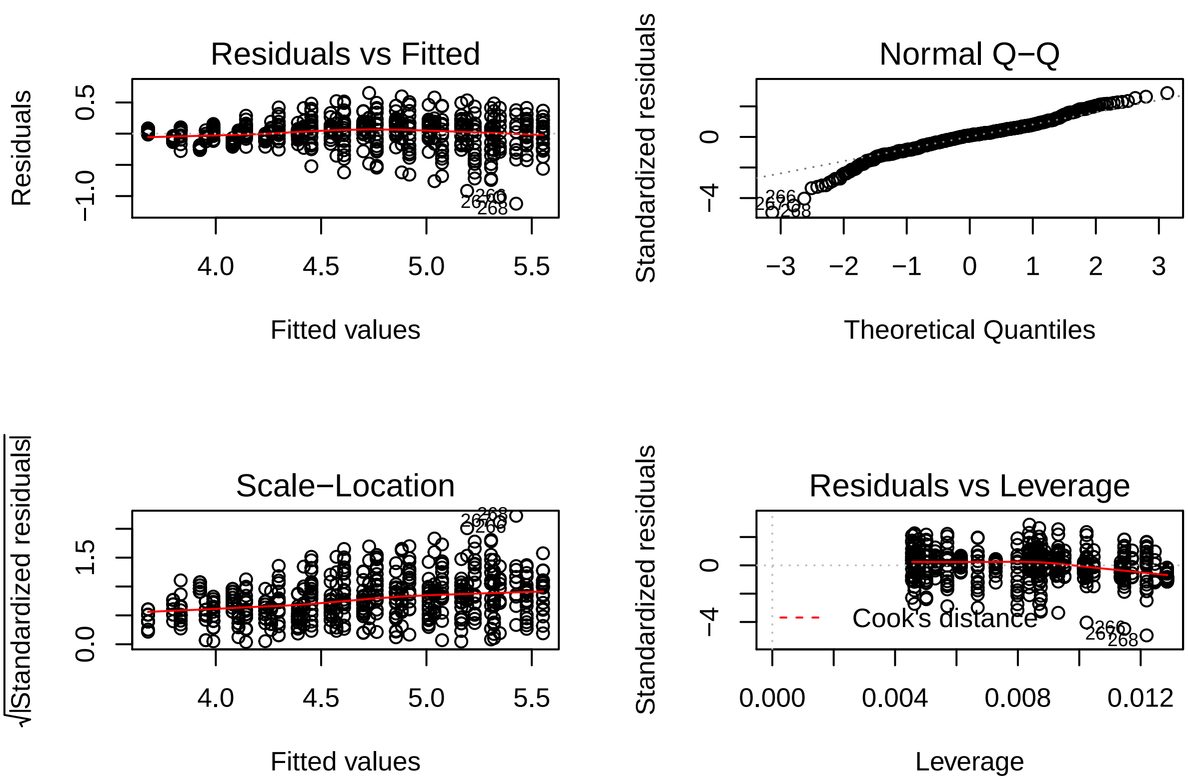
Next we try predicting the log of weight to illustrate that linear models are not always straight lines. This damped down some of the variation in the residuals. The lm3 model had a higher R-squared of 0.8474. We cannot do anova() comparing lm3 because it has a different target, the log(weight) instead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

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

```
##
```

```
## Call:
## lm(formula = log(weight) ~ Time + Diet, data = ChickWeight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.12169 -0.11939  0.02129  0.12597  0.65207
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.678196   0.021301 172.680  < 2e-16 ***
## Time        0.077548   0.001406  55.161  < 2e-16 ***
## Diet2       0.119065   0.025897   4.598 5.26e-06 ***
## Diet3       0.246586   0.025897   9.522  < 2e-16 ***
## Diet4       0.246354   0.026034   9.463  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2281 on 573 degrees of freedom
## Multiple R-squared:  0.8484, Adjusted R-squared:  0.8474
## F-statistic:   802 on 4 and 573 DF,  p-value: < 2.2e-16
```

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



### Predict on test data with model 1

rmse = 30.44 cor = .87

```r
pred <- predict(lm1, newdata=test)
cor(pred, test$weight)
```

```
## [1] 0.8720341
```

```r
mse <- mean((pred-test$weight)^2)
rmse <- sqrt(mse)
```

### Predict on test data with model 2

cor is slightly better at .887 rmse is slightly better at 29.00

```r
pred <- predict(lm2, newdata=test)
cor(pred, test$weight)
```

```
## [1] 0.8872011
```

```r
mse2 <- mean((pred-test$weight)^2)
rmse2 <- sqrt(mse2)
```

### Predict on test data with model 3

correlation is better at .9289 rmse is significanlty better, at 0.2

```r
pred <- predict(lm3, newdata=test)
cor(pred, log(test$weight))
```

```
## [1] 0.9289614
```

```r
mse3 <- mean((pred-log(test$weight))^2)
rmse3 <- sqrt(mse3)
```

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