Discussion 3

Rummerfield & Berman

11/17/2017

Case Diagnostics for Linear Models

In simple linear regression we saw how outliers and influential points could influence our regression model. We're going to expand on that idea with multiple linear regressions. There are four measures we will discuss about how to identify an outlier and/or influential points.

Leverage

Each point of data consists of (X_i, y_i). However, leverage is calculated using only the explanatory variables, the X_i. Thus if a point has high leverage it has an unusual explanatory variable (or a combination of unusual explanatory variables). In the simple linear case leverage is calculated by

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_j (x_j - \bar{x})^2}$$

In simple linear regression, $\sum_{i=1}^n h_i = 2$. With multiple linear regression, leverage is the diagonal elements of the hat matrix, hence the notation of h. You don't need to know how to calculate the hat matrix for Stat 110 or Stat 201, but you do need to know that $\sum_{i=1}^n h_i = p + 1$; p = number of explanatory variables. The average leverage value is $\bar{h} = \frac{\sum_{i=1}^n h_i}{n} = \frac{p+1}{n}$, thus a point of data should have a leverage value of about $\approx \frac{p+1}{n}$. When a point of data has a leverage value greater than $\frac{2(p+1)}{n}$ it is considered a point of high leverage and should the value be greater than $\frac{3(p+1)}{n}$ the point has very high leverage.

Residuals

There are two types of residuals we will discuss in this class: standardized residuals and studentized residuals. Both residuals are based off the difference between the observed value and the fitted value. This means that if a point of data is flagged for having a high standardized or studentized residual the response value is considered unusual. Standardized residuals are calculated as:

$$\frac{y_i - \hat{y}_i}{\sqrt{\hat{\sigma}^2(1 - h_i)}}$$

and the Studentized residuals are calculated as:

$$\frac{y_i - \hat{y}_i}{\sqrt{\hat{\sigma}^2_{(i)}(1 - h_i)}}$$

where $\hat{\sigma}^2_{(i)}$ is the estimate of population variance with data point i omitted from the regression.

Cook's distance

The final measure we will discuss is Cook's Distance. To calculate Cook's Distance:

$$D_i = \underbrace{\frac{(y_i - \hat{y}_i)^2}{\hat{\sigma}^2(1-h_i)}}_{\text{stand. residual}^2} \times \frac{h_i}{1-h_i} \times \frac{1}{p+1}$$

The Cook's distance is considered high if it is greater than 0.5 and extreme if it is greater than 1. If a point has been flagged by the Cook's distance, the point is considered highly influential and has a combination of unusual explanatory variables and response values (the combination of X_i's and y_i are unusual).

How to visually spot outliers and influential points of data

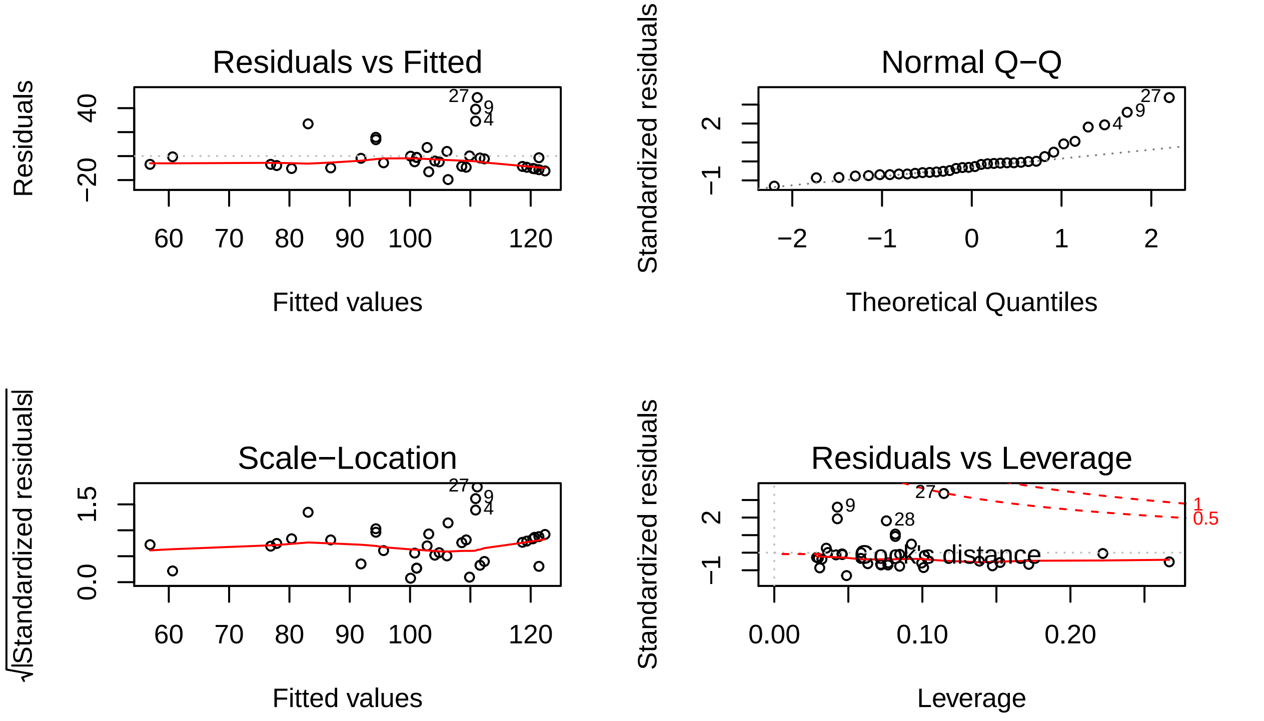
Below is an example of how to find influential points and possible outliers using the Cereal data set. Recall the Cereal data set consists of the following variables:

Cereal	Brand name of cereal
Calories	Number of calories per serving
Sugar	Number of grams of sugar (per serving)
Fiber	Number of grams of fiber (per serving)

We want to create a linear model that predicts the amount of calories in a serving of a cereal based on its sugar and fiber content.

```r
data("Cereal")
cereal.mod <- lm(Calories ~ Fiber + Sugar, data = Cereal)
```

Previously you saw how to plot the linear model object using the `plot()` function in R. Let's do that with `cereal.mod` and see what we get:

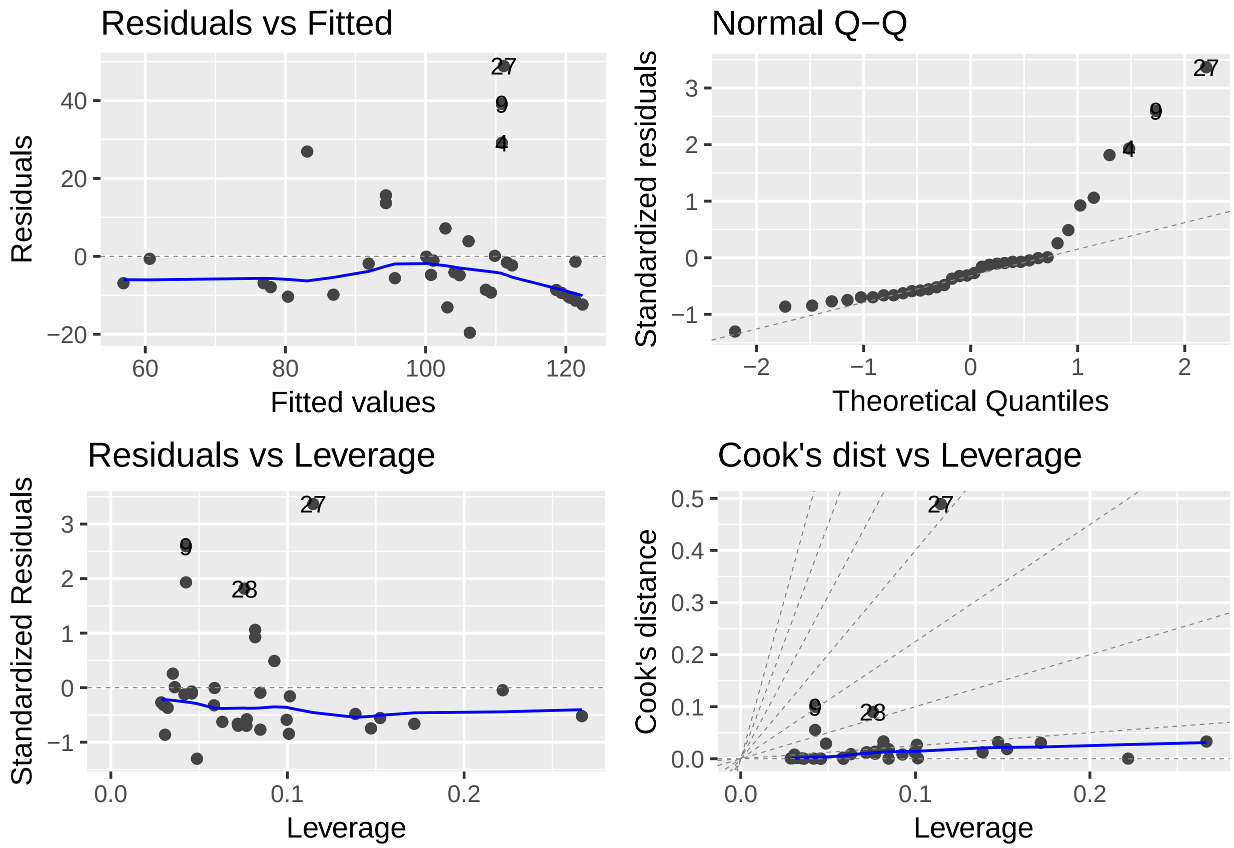


```r
plot(cereal.mod)
```

In the first plot, we get the residuals vs. fitted values, this plot can help us determine which residuals are high, but because it plots the raw residual it is not very useful. The second plot, the QQ plot, has been discussed in the past and is not relevant to this topic. The third plot, the scale-location plot is also not relevant. The fourth plot, the residuals vs. leverage, is quite useful. On the y-axis are the standardized residuals, so any point that is above 2 (or below -2) can be classified as having a high standardized residual value. The x-axis gives us the leverage value for each point; since there are 36 cereals in the data set and 3 coefficients to fit (2 explanatory variables + the intercept) then the average leverage is $3/36 \approx 0.083$ then any point beyond 0.167 (2×0.083) has a high leverage value. Lastly, the plot draws the contours for the Cook's Distance; any point that exists outside of the 0.5 contour line has a high Cook's distance.

Alternatively, you can use `ggfortify` package to create similar plots using `ggplot`.

```r
library(ggfortify)
autoplot(cereal.mod, which = c(1:2,5:6), label.size = 3)
```



Note that the third and fourth plot are not the same as above! The third plot is the standardized residuals vs. the leverage, which is the same as the fourth plot from the `plot` function in base R. The fourth plot is Cook's distance versus leverage, which is not included in `plot` function in base R. To read this plot, the y-axis models Cook's distance, so any point above 0.5 has a high Cook's distance value, while the x-axis is leverage, so any point beyond 0.167 has a high leverage value.

How to flag outliers and influential data points

From a linear model object, in our example, `cereal.mod`, we can get the leverage, standardized residuals, studentized residuals, and Cook's distance for each point of data. It is very helpful to add each of these to our initial data set. Let's look at the first six rows to see what happens when we add those columns to the cereal data set.

```
Cereal$Leverage <- hatvalues( cereal.mod )
Cereal$Stand.resid <- rstandard( cereal.mod )
Cereal$Stud.resid <- rstudent( cereal.mod )
Cereal$Cooks.dis <- cooks.distance( cereal.mod )
```

```
##   Calories Sugar Fiber Leverage Stand.resid Stud.resid Cooks.dis
## 1      100     6     3   0.0286      -0.270     -0.266  0.000715
## 2      100     3     1   0.0770      -0.579     -0.573  0.009329
## 3       50     0    14   0.2667      -0.522     -0.516  0.033032
## 4      140     9     2   0.0426       1.931      2.020  0.055311
## 5       70     5    10   0.1385      -0.482     -0.476  0.012429
## 6       90     5     6   0.0417      -0.124     -0.122  0.000222
```

The `which()` function

Producing the data frame with these new columns is good, but we need to somehow filter this list so it only displays high or extreme values. One way to catch these cases is to use the `which()` function in R. The `which()` function simply returns a vector of which rows meet these conditions.

```
which( abs( Cereal$Stand.resid ) >=2 )
```

```
## [1]  9 27
```

By subsetting the original dataset we can identify which cereals have high standardized residuals:

```
Cereal[ which( abs( Cereal$Stand.resid ) >=2 ) , ]
```

```
##    Calories Sugar Fiber Leverage Stand.resid Stud.resid Cooks.dis
## 9       150     9     2   0.0426        2.59       2.86    0.0998
## 27      160    13     3   0.1146        3.37       4.09    0.4892
```

If we repeat the same procedure for Cook's distance we get an odd result:

```
Cereal[ which( Cereal$Cooks.dis >= 0.5 ) , ]
```

```
## [1] Calories    Sugar       Fiber       Leverage    Stand.resid Stud.resid
## [7] Cooks.dis
## <0 rows> (or 0-length row.names)
```

The output here indicates that there are no data points in the cereal data set that have a Cook's distance that is greater than 0.5. The output `integer(0)` means there is no row that matches this criteria.

```
which( Cereal$Cooks.dis >= 0.5 )
```

```
## integer(0)
```

How to flag for leverage

Recall, to identify a point with a high leverage value we need to know the number of beta coefficients (typically, this equals the number of explanatory variables + 1 for the intercept) and the number of data points in our model. Acquiring both of these values via R can make it easier to flag cases with high leverage values.

```
num.Of.Coefs <- length( cereal.mod$coefficients )
num.Of.Cases <- length( cereal.mod$residuals )

which( Cereal$Leverage > 2 * num.Of.Coefs / num.Of.Cases )
```

```
## [1]  3 26 34
```

The data points which are flagged as having high leverage are:

```
Cereal[ which( Cereal$Leverage > 2 * num.Of.Coefs / num.Of.Cases ), ]
```

```
##    Calories Sugar Fiber Leverage Stand.resid Stud.resid Cooks.dis
## 3        50     0    14    0.267     -0.5219    -0.5161  0.033032
## 26      100     0     0    0.172     -0.6635    -0.6577  0.030453
## 34       60     0    13    0.222     -0.0466    -0.0459  0.000207
```

Flagging all the influential points

Many of the points with high leverage will likely be the points with high residual values. It would be nice to create a table of all the data points that are flagged for one reason or another. Below are all the data points we want in this table with only flagged cases:

```
which( abs( Cereal$Stand.resid ) >= 2 )
```

```
## [1]  9 27
```

```
which( abs( Cereal$Stud.resid ) >= 2 )
```

```
## [1]  4  9 27
```

```
which( Cereal$Leverage >= 2 * num.Of.Coefs / num.Of.Cases )
```

```
## [1]  3 26 34
```

```
which( Cereal$Cooks.dis >= 0.5 )
```

```
## integer(0)
```

By using the `union()` function we can create a nice table. The `union()` function simply joins the elements from two sets with the *or* function.

```
resids <- union( which(abs(Cereal$Stand.resid) >= 2), which(abs(Cereal$Stud.resid) >= 2) )
lev.Cooks <- union( which( Cereal$Leverage >= 2*num.Of.Coefs/num.Of.Cases),
                    which( Cereal$Cooks.dis >= 0.5 ))
sort( union(resids, lev.Cooks) )
```

```
## [1]  3  4  9 26 27 34
```

All the flagged cases in the cereal data set:

```
Cereal[ sort( union(resids, lev.Cooks) ) , ]
```

```
##    Calories Sugar Fiber Leverage Stand.resid Stud.resid Cooks.dis
## 3        50     0    14   0.2667     -0.5219    -0.5161  0.033032
## 4       140     9     2   0.0426      1.9314     2.0195  0.055311
## 9       150     9     2   0.0426      2.5944     2.8634  0.099795
## 26      100     0     0   0.1719     -0.6635    -0.6577  0.030453
## 27      160    13     3   0.1146      3.3681     4.0941  0.489201
## 34       60     0    13   0.2219     -0.0466    -0.0459  0.000207
```