SDM4 in R: Multiple Regression Wisdom (Chapter 29)

Nicholas Horton (nhorton@amherst.edu), Patrick Frenett, and Sarah McDonald

June 13, 2018

Introduction and background

This document is intended to help describe how to undertake analyses introduced as examples in the Fourth Edition of *Stats: Data and Models* (2014) by De Veaux, Velleman, and Bock. More information about the book can be found at http://wps.aw.com/aw_deveaux_stats_series. This file as well as the associated R Markdown reproducible analysis source file used to create it can be found at http://nhorton.people.amherst.edu/sdm4.

This work leverages initiatives undertaken by Project MOSAIC (http://www.mosaic-web.org), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignettes (http://cran.r-project.org/web/packages/mosaic). A paper describing the mosaic approach was published in the *R Journal*: https://journal.r-project.org/archive/2017/RJ-2017-024.

Chapter 29: Multiple Regression Wisdom

```
library(mosaic)
library(readr)
Coasters <- read_csv("http://nhorton.people.amherst.edu/sdm4/data/Roller_coasters_2014.csv")
glimpse(Coasters)
```

```
## Observations: 198
## Variables: 10
## $ Name       <chr> "Top Thrill Dragster", "Superman The Escap", "Mille...
## $ Park       <chr> "Cedar Point", "Six Flags Magic Mountain", "Cedar P...
## $ Track      <chr> "Steel", "Steel", "Steel", "Steel", "Steel", "Steel...
## $ Speed      <chr> "120", "100", "93", "85", "85", "82", "82", "80", "...
## $ Height     <dbl> 420.0, 415.0, 310.0, 235.0, 245.0, 160.0, 205.0, 20...
## $ Drop       <dbl> 400.0, 328.1, 300.0, 255.0, 255.0, 228.0, 130.0, 22...
## $ Length     <dbl> 2800, 1235, 6595, 4500, 5312, 3200, 2202, 5843, 156...
## $ Duration   <int> NA, NA, 165, 180, 210, NA, 62, 163, NA, 240, NA, NA...
## $ Inversions <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, ...
## $ Opened     <int> 2003, 1997, 2000, 2000, 2001, 2001, 2002, 1909, 200...
```

```
tally(~ Speed, data = Coasters)
```

```
## Speed
##  *100   *56   100   120 149.1  16.2  17.4  21.8  22.4  23.6  25.7  26.7
##     1     1     1     1     1     1     1     1     2     1     3     1
##    28  28.6  29.1  32.9  34.2    35    36    37  37.3  38.5   4.5    40
##     5     1     3     1     1     1     1     1     2     1     1     2
##  40.4    41    42  43.5  44.7    45    46  46.6    47  47.9  49.7    50
##     1     1     1     6     2     3     1     3     6     2    11     6
##  50.1    52  52.8    53  53.7  54.7    55  55.9    56    59    60  60.3
```

```
##     1     3     6     2     1     1    11     3     2     1     1     2
##    62  62.1    63  64.8    65  65.2  65.3  65.6    66  66.3    67  67.1
##     4     8     4     1    10     1     7     4     2     1     4     2
##    68    70  71.5    72    73    74  74.6    75    76    77  78.4    80
##     4     3     1     1     4     1     1     4     1     1     1     5
##  80.8    82    83  83.3    85    90    93    95
##     2     2     1     1     2     1     1     1
```

```
filter(Coasters, Speed == "*100" | Speed == "*56")
```

```
## # A tibble: 2 x 10
##   Name   Park   Track Speed Height  Drop Length Duration Inversions Opened
##   <chr>  <chr>  <chr> <chr>  <dbl> <dbl>  <dbl>    <int>       <int>  <int>
## 1 Tower~ Dream~ Steel *100    377.  328.   1235       28           0   1997
## 2 Wild ~ Param~ Wood  *56     415    78    3150      150           0   1981
```

Note that two of the observations for speed have special characters in them. We first need to address these data irregularities:

```
library(readr)
Coasters <- mutate(Coasters, Speed = parse_number(Speed))
```

To match the output we need to remove these two coasters.

```
dim(Coasters)
```

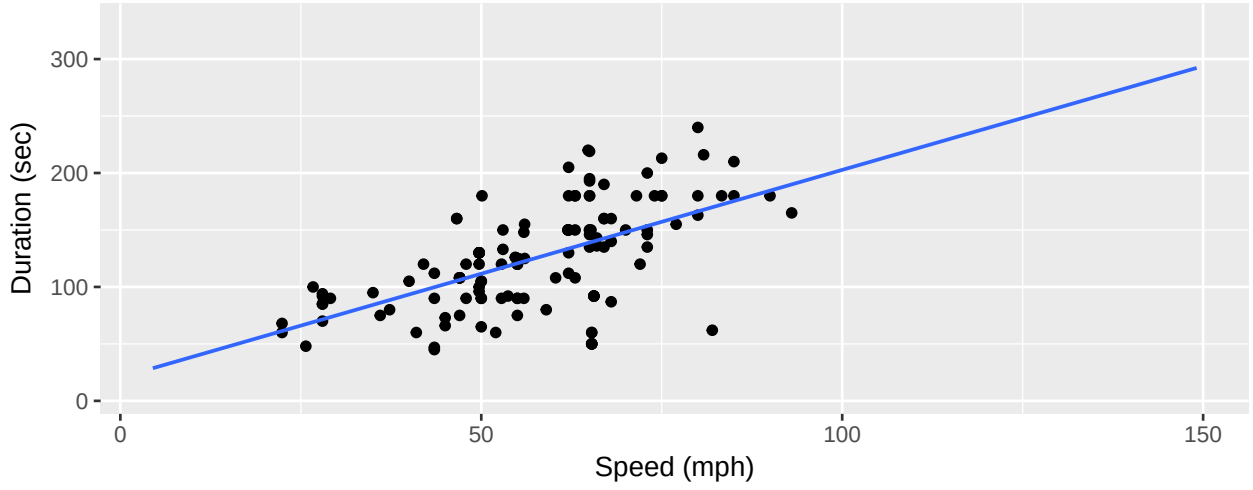
```
## [1] 198  10
```

```
Coasters <- filter(Coasters,
  !(Name %in% c("Tower of Terror", "Wild Beast")))
dim(Coasters)
```

```
## [1] 196  10
```

Figure 29.1 (page 860) displays the scatterplot of duration as a function of speed.

```
gf_point(Duration ~ Speed, xlab = "Speed (mph)",
  ylab = "Duration (sec)", data = Coasters) %>%
  gf_lm()
```

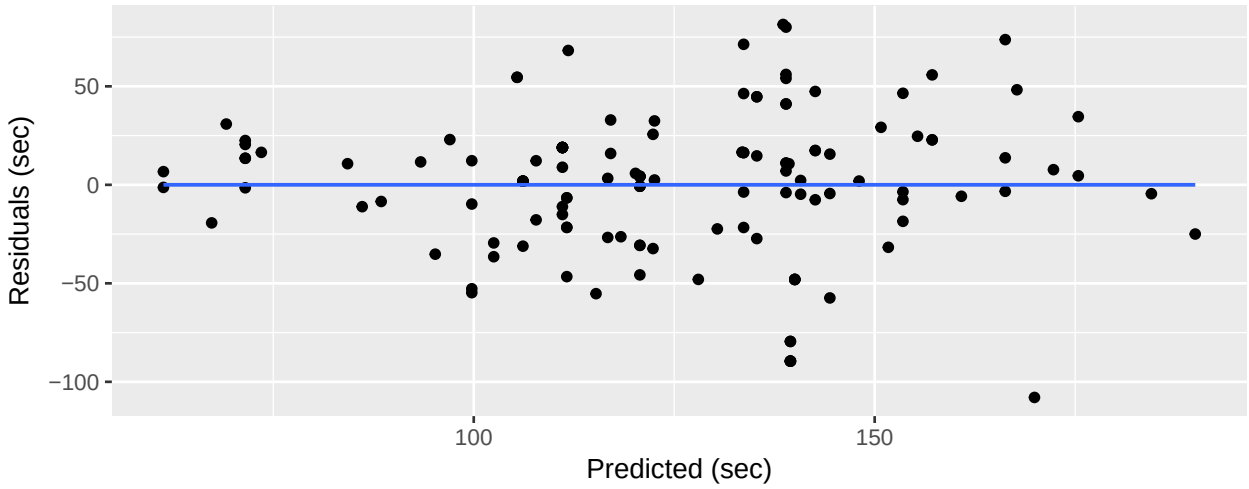


We can also replicate the regression model (and regression diagnostics) on the bottom of page 860.

```
mod1 <- lm(Duration ~ Speed, data = Coasters)
msummary(mod1)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.474     12.925    1.58     0.12
## Speed          1.823      0.219    8.34  8.6e-14 ***
##
## Residual standard error: 36.4 on 132 degrees of freedom
##   (62 observations deleted due to missingness)
## Multiple R-squared:  0.345,  Adjusted R-squared:  0.34
## F-statistic: 69.6 on 1 and 132 DF,  p-value: 8.59e-14
```

```
gf_point(resid(mod1) ~ fitted(mod1), xlab = "Predicted (sec)",
  ylab = "Residuals (sec)") %>%
  gf_lm()
```



Section 29.1: Indicators

The model displayed on the bottom of page 861 is restricted to roller coasters with no inversions.

```
noinversion <- filter(Coasters, Inversions == 0)  # 0 is No (or FALSE)
mod2 <- lm(Duration ~ Speed, data = noinversion)
msummary(mod2)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   33.945     13.313    2.55    0.014 *
## Speed          1.795      0.214    8.38  2.8e-11 ***
##
## Residual standard error: 31.4 on 53 degrees of freedom
##   (30 observations deleted due to missingness)
## Multiple R-squared:  0.57,   Adjusted R-squared:  0.562
## F-statistic: 70.2 on 1 and 53 DF,  p-value: 2.79e-11
```

We can do the same for coasters with inversions:

```
mod3 <- lm(Duration ~ Speed, data = filter(Coasters, Inversions == 1))
msummary(mod3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   22.773     27.712    0.82   0.4137
## Speed          1.635      0.486    3.36   0.0012 **
##
## Residual standard error: 37.6 on 77 degrees of freedom
##   (32 observations deleted due to missingness)
## Multiple R-squared:  0.128,  Adjusted R-squared:  0.117
## F-statistic: 11.3 on 1 and 77 DF,  p-value: 0.00121
```

The model at the top of page 862 adds the Inversion indicator to the model.

```
mod4 <- lm(Duration ~ Speed + Inversions, data = Coasters)
msummary(mod4)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   35.997     13.343    2.70   0.0079 **
## Speed          1.760      0.212    8.31    1e-13 ***
## Inversions   -20.273      6.187   -3.28   0.0013 **
##
## Residual standard error: 35.1 on 131 degrees of freedom
##   (62 observations deleted due to missingness)
## Multiple R-squared:  0.395,  Adjusted R-squared:  0.385
## F-statistic: 42.7 on 2 and 131 DF,  p-value: 5.24e-15
```

```
mod4fun <- makeFun(mod4)
```

We can generate predicted values from this model.

```
mod4fun(Speed = 55, Inversions = 0)  # Hayabusa
```

```
##     1
## 132.8
```

```
mod4fun(Speed = 60.3, Inversions = 1)  # Hangman
```

```
##     1
## 121.9
```

On page 864 the Burger King example is introduced with a subset of items from the menu.

```
BK <- read.csv("http://nhorton.people.amherst.edu/sdm4/data/BK_items.csv",
               stringsAsFactors = FALSE)
BKmod <- lm(Calories ~ Carbs.g. + Meat. + Carbs.g.*Meat., data = BK)
msummary(BKmod)
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       137.40      58.72    2.34   0.0267 *
## Carbs.g.            3.93       1.11    3.53   0.0014 **
## Meat.Y            -26.16      98.48   -0.27   0.7925
## Carbs.g.:Meat.Y     7.88       2.18    3.61   0.0012 **
##
## Residual standard error: 106 on 28 degrees of freedom
## Multiple R-squared:  0.781,  Adjusted R-squared:  0.757
## F-statistic: 33.2 on 3 and 28 DF,  p-value: 2.32e-09
```

The same predicted values can be calculated using this model.

```
BKfun <- makeFun(BKmod)
BKfun(Carbs.g = 53, Meat = "Y")
```

```
##     1
## 737.1
```

```
BKfun(Carbs.g = 43, Meat = "N")
```

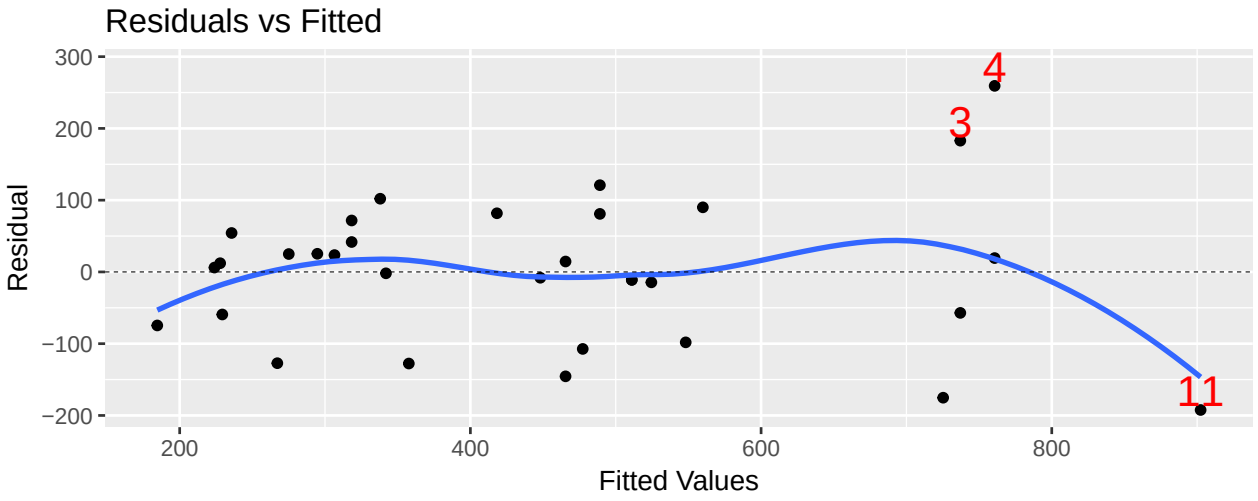
```
##     1
## 306.5
```

Section 29.2: Diagnosing Multiple Regression Models

Below are 6 plots made using the `mplot` function. These all work to better inform decisions when interpreting extreme values / the appropriateness of the model. Whilst the normal quantile and residual plots test the assumptions of the multiple regression model, the plots involving Cook's Distance and leverage help us identify possible outliers.

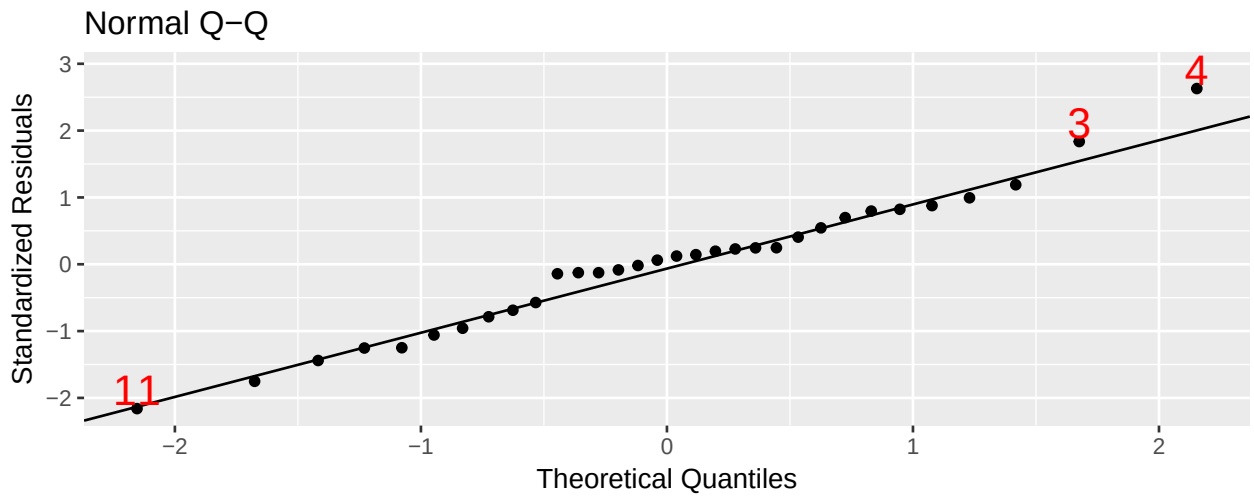
```
BK <- mutate(BK, resid = resid(BKmod), fitted = fitted(BKmod),
             student = rstudent(BKmod), CooksD = cooks.distance(BKmod))
mplot(BKmod, which = 1)
```

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



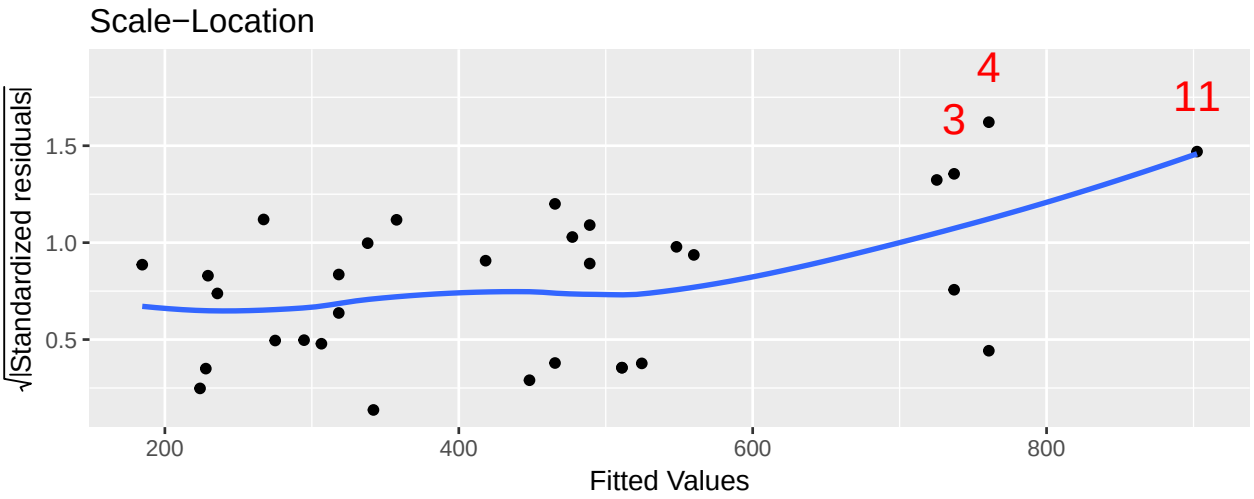
```
mplot(BKmod, which = 2)
```

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



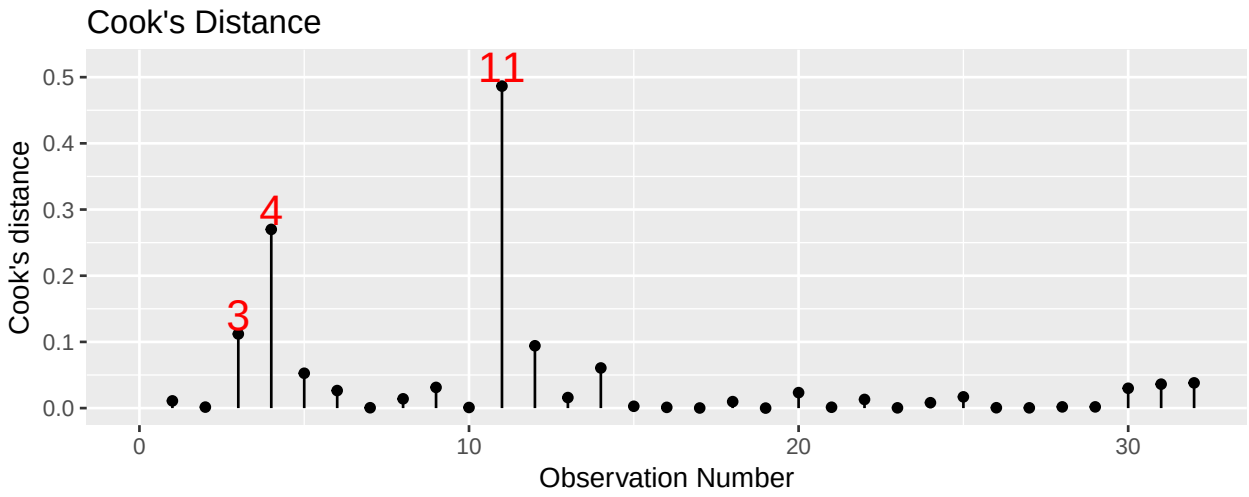

```
mplot(BKmod, which = 3)
```

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




```
mplot(BKmod, which = 4)
```

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




```
favstats(~ CooksD, data = BK)
```

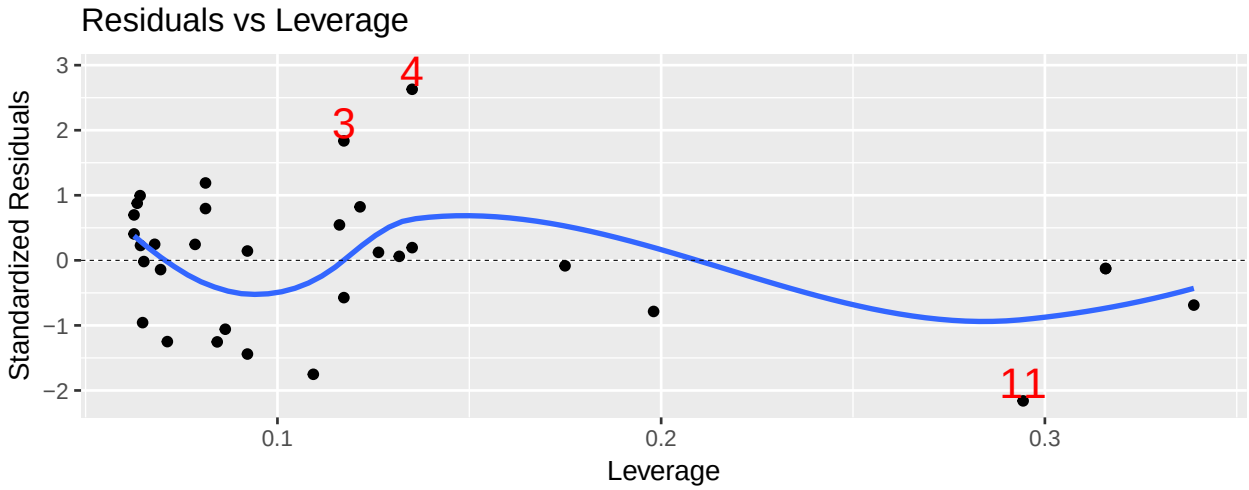
```
##        min       Q1  median     Q3    max   mean      sd  n missing
##  6.114e-06 0.001238 0.01197 0.0325 0.4865 0.0426 0.09613 32       0
```

```
BK[11,]   # or filter(BK, Carbs.g. > 0.486)
```

```
##           Item Calories Protein Total.Fat Carbs.g.   Na.S Meat.  resid
## 11 BK Big Fish      710      24        38       67 4.5627     Y -192.4
##    fitted student CooksD
## 11  902.4  -2.323 0.4865
```

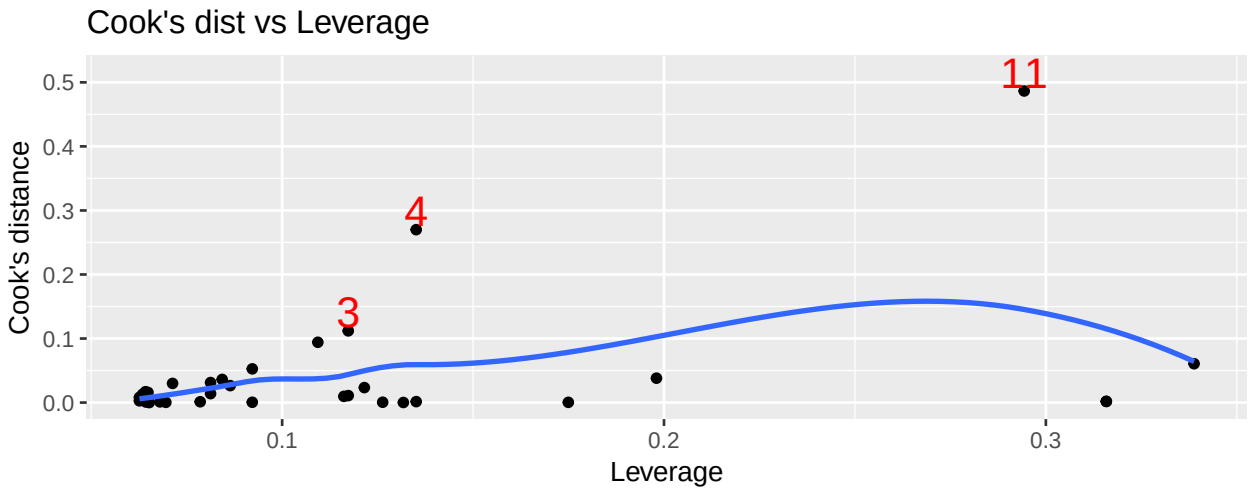
```
mplot(BKmod, which = 5)
```

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




```
mplot(BKmod, which = 6)
```

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



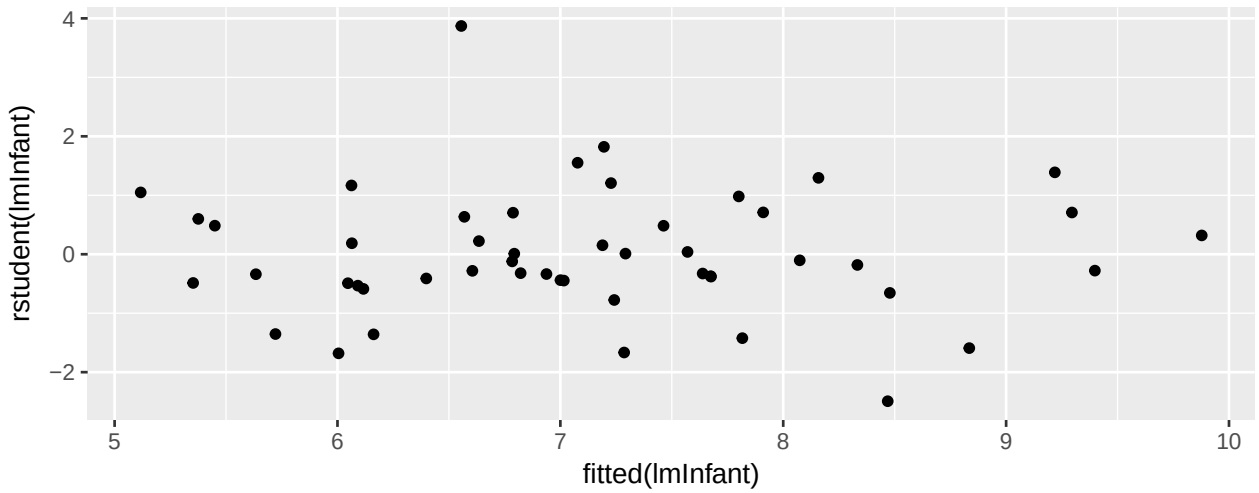
Section 29.3: Building Multiple Regression Models

Using the Infant Mortality data used in the step by step example on page 874, we can create a linear model of all the variables exept `State`. To do this we can use `.` to specify adding all the variables to the model and then `-State` to remove `State` from the model.

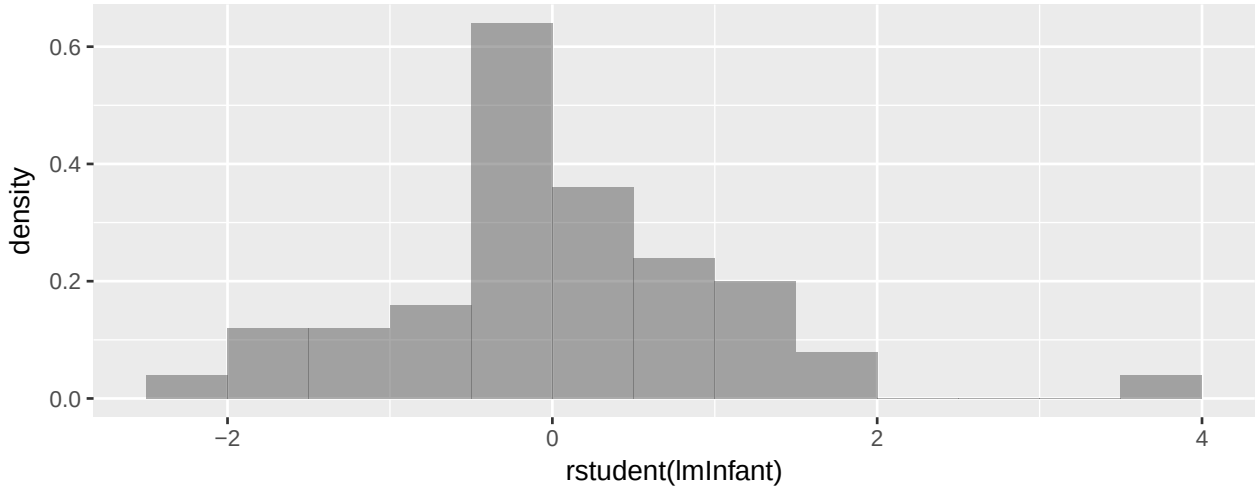
```
InfantMortality <- read_csv("http://nhorton.people.amherst.edu/sdm4/data/Infant_Mortality.csv")

lmInfant <- lm(Infantmort ~ . -State, data = InfantMortality)
msummary(lmInfant)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.63168    0.91240    1.79    0.081 .
## CDR          0.03123    0.01385    2.25    0.029 *
## HSdrop      -0.09971    0.06105   -1.63    0.110
## lowBW        0.66103    0.11891    5.56  1.5e-06 ***
## TeenBirths   0.01357    0.02380    0.57    0.571
## TeenDeaths   0.00556    0.01128    0.49    0.624
##
## Residual standard error: 0.752 on 44 degrees of freedom
## Multiple R-squared:  0.713,  Adjusted R-squared:  0.68
## F-statistic: 21.8 on 5 and 44 DF,  p-value: 6.31e-11
```

```
gf_point(rstudent(lmInfant) ~ fitted(lmInfant))
```




```
gf_dhistogram(~ rstudent(lmInfant), binwidth = 0.5, center = 0.25)
```



The `summary` output shows that only two of the variables are significant at the 5% level. This could be due to multicollinearity. A correlation matrix will help identify variables that are strongly correlated to one and other.

```
cor(InfantMortality[,-1])
```

```
##             Infantmort    CDR HSdrop  lowBW TeenBirths TeenDeaths
## Infantmort      1.0000 0.6522 0.2536 0.7583     0.5417     0.5214
## CDR             0.6522 1.0000 0.4195 0.4662     0.5917     0.8116
## HSdrop          0.2536 0.4195 1.0000 0.3761     0.7593     0.2982
## lowBW           0.7583 0.4662 0.3761 1.0000     0.6215     0.3160
## TeenBirths      0.5417 0.5917 0.7593 0.6215     1.0000     0.4456
## TeenDeaths      0.5214 0.8116 0.2982 0.3160     0.4456     1.0000
```

From this we see that `HSdrop`, `TeenBirths` and `TeenDeaths` are correlated. They are also the 3 least significant variables in the full model so it might not be unreasonable to remove them from the model.

```
lmInfant2 <- lm(Infantmort ~ . -State -TeenDeaths -TeenBirths -HSdrop, data = InfantMortality)
msummary(lmInfant2)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.43612    0.71027    2.02  0.04890 *
## CDR          0.03378    0.00814    4.15  0.00014 ***
## lowBW        0.64630    0.10239    6.31    9e-08 ***
##
## Residual standard error: 0.757 on 47 degrees of freedom
## Multiple R-squared:  0.689,  Adjusted R-squared:  0.676
## F-statistic: 52.1 on 2 and 47 DF,  p-value: 1.2e-12
```

Notice that the R^2 of this final model is 0.689: this is different that in the book. This illustrates how there are a number of approaches when choosing the 'best' model and that different models may be better or worse given the constraints/specifications of the problem.