Tutorial 9: Regression Diagnostics I

Johannes Karreth

RPOS 517, Day 9

This tutorial shows you:

- how to use and interpret dummy variables in linear regression
- how to diagnose outliers in a multiple regression model
- how to assess the influence of outliers on regression results

Note on copying & pasting code from the PDF version of this tutorial: Please note that you may run into trouble if you copy & paste code from the PDF version of this tutorial into your R script. When the PDF is created, some characters (for instance, quotation marks or indentations) are converted into non-text characters that R won't recognize. To use code from this tutorial, please type it yourself into your R script or you may copy & paste code from the *source file* for this tutorial which is posted on my website.

Note on R functions discussed in this tutorial: I don't discuss many functions in detail here and therefore I encourage you to look up the help files for these functions or search the web for them before you use them. This will help you understand the functions better. Each of these functions is well-documented either in its help file (which you can access in R by typing `?ifelse`, for instance) or on the web. The *Companion to Applied Regression* (see our syllabus) also provides many detailed explanations.

As always, please note that this tutorial only accompanies the other materials for Day 9 and that you are expected to have worked through the reading for that day before tackling this tutorial.

Dummy variables in regression

Some of you are working with so-called dummy variables in your replication assignments, so we will briefly explore how these variables are used in multiple regression. Dummy variables are also explained well in chapter 7 of *AR* (assigned on Day 11), but it doesn't hurt to explore them earlier. Dummy variables are binary indicators that are set to 1 for all observations matching a particular classification and 0 to all other observations. For instance, a dummy variable in a survey for married respondents will be coded the following way:

$$married = \begin{cases} 1, & \text{if respondent is married} \\ 0, & \text{otherwise} \end{cases}$$

Survey data example: Attitudes toward Hillary Clinton

We'll start with an example dataset that I've taken from the accompanying materials to Kellstedt and Whitten's *Fundamentals of Political Science Research*. This dataset is a modified extract from the 1996 edition of the (American) National Election studies. This dataset has 1714 observations and 8 variables:

Variable	Description
demrep	Party identification (1 = strong Democrat, 7 = strong Republican)
clinton.therm	Feeling thermometer toward Hillary Clinton
dem.therm	Feeling thermometer toward the Democrats
female	Female (1 = yes, 0 = no)

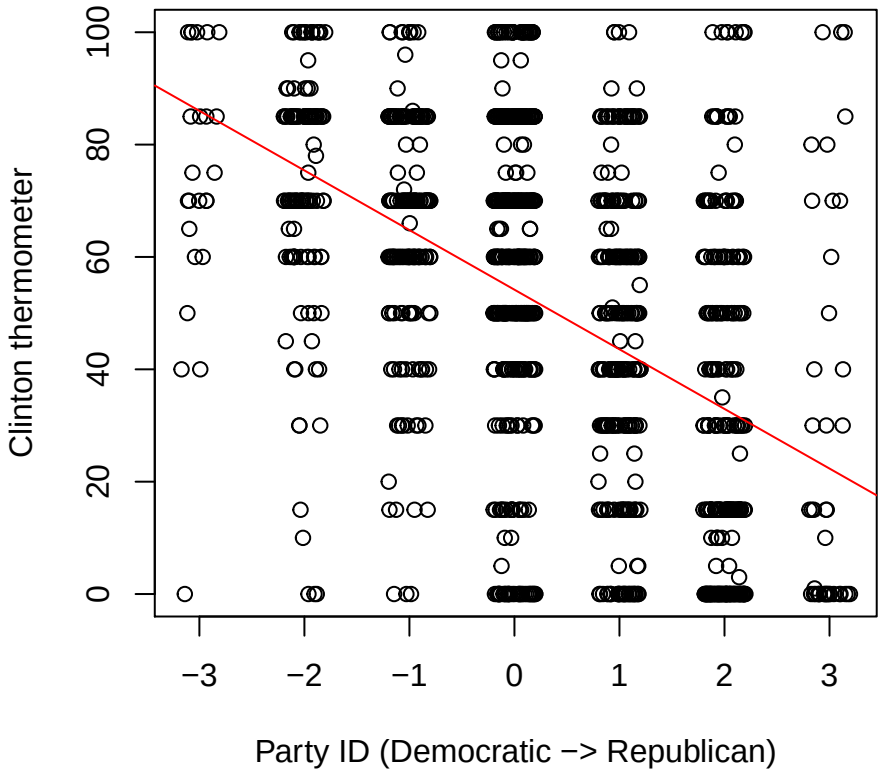
Variable	Description
age	Age in years
educ	Education (1 = 8 grades or less, 7 = advanced degree)
income	Income (1 = less than $2999, 24 = $105,000 or more)
region	Northeast, North Central, South, or West

```
nes.dat <- read.csv("http://www.jkarreth.net/files/nes1996subset.csv")
summary(nes.dat)
```

```
##      demrep      clinton.therm      dem.therm          female
##  Min.   :1.000   Min.   :  0.00   Min.   :  0.00   Min.   :0.0000
##  1st Qu.:3.000   1st Qu.: 30.00   1st Qu.: 40.00   1st Qu.:0.0000
##  Median :4.000   Median : 60.00   Median : 60.00   Median :1.0000
##  Mean   :4.327   Mean   : 52.81   Mean   : 58.86   Mean   :0.5519
##  3rd Qu.:5.000   3rd Qu.: 70.00   3rd Qu.: 70.00   3rd Qu.:1.0000
##  Max.   :7.000   Max.   :100.00   Max.   :100.00   Max.   :1.0000
##  NA's   :385     NA's   :29       NA's   :27
##       age             educ           income               region
##  Min.   :18.00   Min.   :1.000   Min.   : 1.00   North Central:458
##  1st Qu.:34.00   1st Qu.:3.000   1st Qu.:11.00   Northeast    :260
##  Median :44.00   Median :4.000   Median :16.00   South        :642
##  Mean   :47.54   Mean   :4.105   Mean   :15.03   West         :354
##  3rd Qu.:61.00   3rd Qu.:6.000   3rd Qu.:20.00
##  Max.   :93.00   Max.   :7.000   Max.   :24.00
##  NA's   :2       NA's   :3       NA's   :150
```

Say you are interested in explaining why some respondents exhibit a more positive attitude toward Hillary Clinton than others. You could use bivariate regression to test the (somewhat obvious) argument that Republican respondents might be less likely to approve of Clinton than more Democratic respondents. First, you may want to means-center the party ID variable for ease of interpretation:

```
nes.dat$demrep.ctr <- nes.dat$demrep - median(nes.dat$demrep, na.rm = TRUE)
m1 <- lm(clinton.therm ~ demrep.ctr, data = nes.dat)
plot(x = jitter(nes.dat$demrep.ctr), y = nes.dat$clinton.therm,
     xlab = "Party ID (Democratic -> Republican)",
     ylab = "Clinton thermometer")
abline(m1, col = "red")
```



You might have a theory that female respondents are more likely to hold a positive attitude toward Hillary Clinton than male respondents. Based on what you've learned so far, you can account for this by "controlling" for gender.

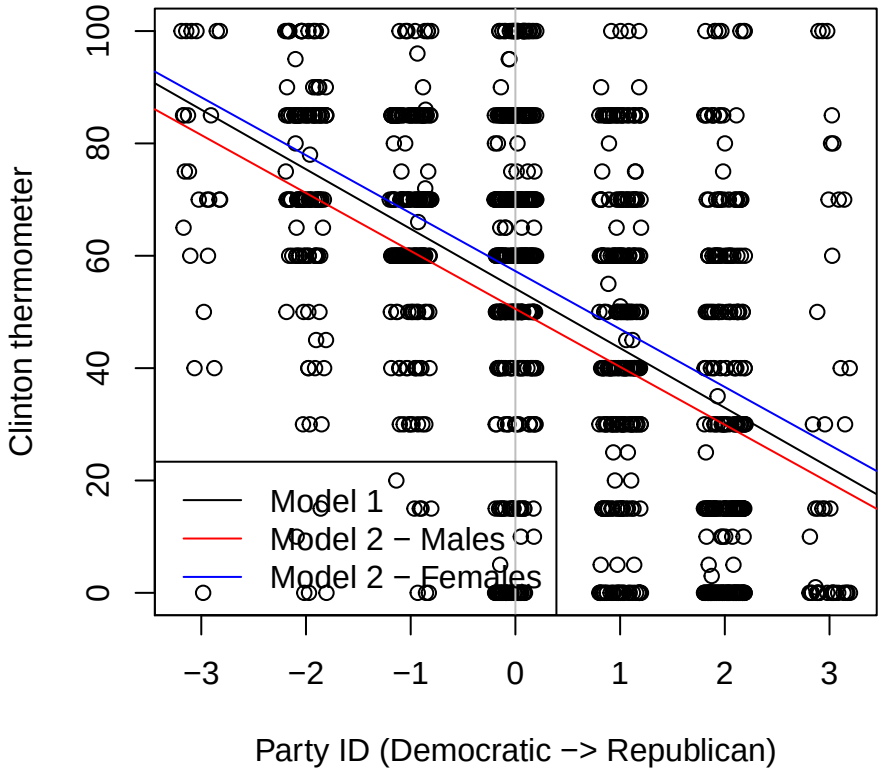
```
m2 <- lm(clinton.therm ~ demrep.ctr + female, data = nes.dat)
library(texreg)
screenreg(list(m1, m2))
```

```
##
## ====================================
##              Model 1      Model 2
## ------------------------------------
## (Intercept)    54.17 ***    50.55 ***
##                (0.75)       (1.09)
## demrep.ctr    -10.61 ***   -10.31 ***
##                (0.53)       (0.53)
## female                       6.72 ***
##                             (1.47)
## ------------------------------------
## R^2             0.24         0.25
## Adj. R^2        0.23         0.25
## Num. obs.    1316         1316
## ====================================
## *** p < 0.001, ** p < 0.01, * p < 0.05
```

2. How do you interpret these results?

A good way to examine the role of dummy variables in regression is to visualize them in a scatterplot. Because a dummy variable can only take on two values, a dummy variable only shifts the intercept ($\hat{y}$ when $x = 0$).

```
with(nes.dat, plot(x = jitter(demrep.ctr), y = clinton.therm, type = "p",
                   xlab = "Party ID (Democratic -> Republican)",
                   ylab = "Clinton thermometer"))
abline(v = 0, col = "gray")
abline(a = coef(m1)[1], b = coef(m1)[2])
abline(a = coef(m2)[1], b = coef(m2)[2], col = "red")
abline(a = coef(m2)[1] + coef(m2)[3], b = coef(m2)[2], col = "blue")
legend("bottomleft", legend = c("Model 1", "Model 2 - Males", "Model 2 - Females"),
       col = c("black", "red", "blue"), lty = 1)
```



Dummy variables are always binary, but they can also be created based on categorical variables with more than two categories. For instance, you might consider the geographic region of respondents. You can use the `region` variable to this end. But this is a categorical variable with four values:

```
str(nes.dat$region)
```

```
##  Factor w/ 4 levels "North Central",..: 2 2 2 2 2 2 2 2 2 2 ...
```

```
table(nes.dat$region)
```

```
##
## North Central     Northeast         South          West
##           458           260           642           354
```

You can manually create four dummy variables, one for each region:

```r
nes.dat$region.northeast <- ifelse(nes.dat$region == "Northeast", 1, 0)
nes.dat$region.northcentral <- ifelse(nes.dat$region == "North Central", 1, 0)
nes.dat$region.south <- ifelse(nes.dat$region == "South", 1, 0)
nes.dat$region.west <- ifelse(nes.dat$region == "West", 1, 0)
```

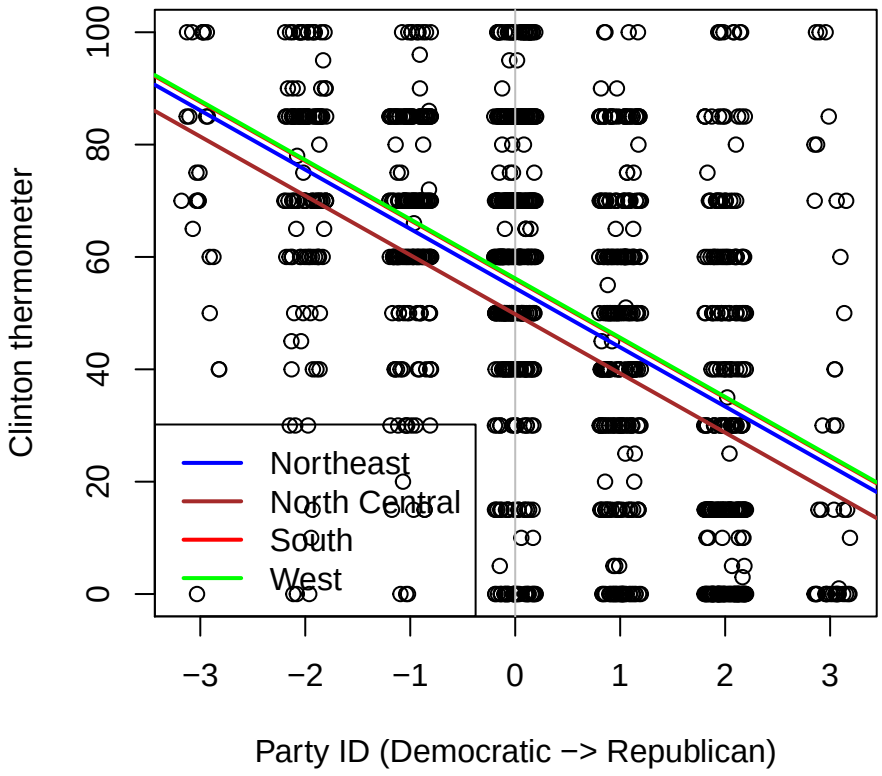
```r
m3 <- lm(clinton.therm ~
           demrep.ctr + region.northeast + region.northcentral + region.south + region.west,
         data = nes.dat)
summary(m3)
```

```
##
## Call:
## lm(formula = clinton.therm ~ demrep.ctr + region.northeast +
##     region.northcentral + region.south + region.west, data = nes.dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -81.432 -18.375   1.269  18.284  77.165
##
## Coefficients: (1 not defined because of singularities)
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          56.1761     1.5249  36.840  < 2e-16 ***
## demrep.ctr          -10.5403     0.5275 -19.980  < 2e-16 ***
## region.northeast     -1.7201     2.3833  -0.722  0.47059
## region.northcentral  -6.3650     2.0810  -3.059  0.00227 **
## region.south         -0.1769     1.9653  -0.090  0.92830
## region.west              NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.45 on 1311 degrees of freedom
##   (398 observations deleted due to missingness)
## Multiple R-squared:  0.2428, Adjusted R-squared:  0.2405
## F-statistic: 105.1 on 4 and 1311 DF,  p-value: < 2.2e-16
```

3. Why does your regression output not contain estimates for the Western United States?

You can again plot this output:

```r
with(nes.dat, plot(x = jitter(demrep.ctr), y = clinton.therm, type = "p",
                   xlab = "Party ID (Democratic -> Republican)",
                   ylab = "Clinton thermometer"))
abline(v = 0, col = "gray")
abline(a = coef(m3)[1] + coef(m3)[3], b = coef(m3)[2], col = "blue", lwd = 2)
abline(a = coef(m3)[1] + coef(m3)[4], b = coef(m3)[2], col = "brown", lwd = 2)
abline(a = coef(m3)[1] + coef(m3)[5], b = coef(m3)[2], col = "red", lwd = 2)
abline(a = coef(m3)[1], b = coef(m3)[2], col = "green", lwd = 2)
legend("bottomleft", legend = c("Northeast", "North Central", "South", "West"),
       col = c("blue", "brown", "red", "green"), lty = 1, lwd = 2)
```



Note the overlap of the lines for South and West.

Alternatively, rather than creating individual dummy variables by hand, you can use the `factor()` function within the regression equation to let R do the work:

```
m3b <- lm(clinton.therm ~
           demrep.ctr + factor(region),
         data = nes.dat)
summary(m3b)
```

```
##
## Call:
## lm(formula = clinton.therm ~ demrep.ctr + factor(region), data = nes.dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -81.432 -18.375   1.269  18.284  77.165
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                49.8111     1.4308  34.814  < 2e-16 ***
## demrep.ctr                -10.5403     0.5275 -19.980  < 2e-16 ***
## factor(region)Northeast     4.6449     2.3178   2.004  0.04528 *
## factor(region)South         6.1881     1.8796   3.292  0.00102 **
## factor(region)West          6.3650     2.0810   3.059  0.00227 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.45 on 1311 degrees of freedom
##   (398 observations deleted due to missingness)
```

```
## Multiple R-squared:  0.2428, Adjusted R-squared:  0.2405
## F-statistic: 105.1 on 4 and 1311 DF,  p-value: < 2.2e-16
```

4. Which coefficient estimates changed compared to `m3` and why?

Lastly, if you have a factor with many levels and would like to create dummy variables for your dataset, but avoid doing this as many times as the variable has categories, you can use the code below to achieve this task. This code does the same thing you did above using the `ifelse()` function by hand:

```
for(level in unique(nes.dat$region)){
    nes.dat[paste("region", level, sep = "_")] <- ifelse(nes.dat$region == level, 1, 0)
}
summary(nes.dat)
```

```
##      demrep      clinton.therm      dem.therm          female
##  Min.   :1.000   Min.   :  0.00   Min.   :  0.00   Min.   :0.0000
##  1st Qu.:3.000   1st Qu.: 30.00   1st Qu.: 40.00   1st Qu.:0.0000
##  Median :4.000   Median : 60.00   Median : 60.00   Median :1.0000
##  Mean   :4.327   Mean   : 52.81   Mean   : 58.86   Mean   :0.5519
##  3rd Qu.:5.000   3rd Qu.: 70.00   3rd Qu.: 70.00   3rd Qu.:1.0000
##  Max.   :7.000   Max.   :100.00   Max.   :100.00   Max.   :1.0000
##  NA's   :385     NA's   :29       NA's   :27
##       age             educ           income                region
##  Min.   :18.00   Min.   :1.000   Min.   : 1.00   North Central:458
##  1st Qu.:34.00   1st Qu.:3.000   1st Qu.:11.00   Northeast    :260
##  Median :44.00   Median :4.000   Median :16.00   South        :642
##  Mean   :47.54   Mean   :4.105   Mean   :15.03   West         :354
##  3rd Qu.:61.00   3rd Qu.:6.000   3rd Qu.:20.00
##  Max.   :93.00   Max.   :7.000   Max.   :24.00
##  NA's   :2       NA's   :3       NA's   :150
##    demrep.ctr      region.northeast region.northcentral  region.south
##  Min.   :-3.0000   Min.   :0.0000   Min.   :0.0000      Min.   :0.0000
##  1st Qu.:-1.0000   1st Qu.:0.0000   1st Qu.:0.0000      1st Qu.:0.0000
##  Median : 0.0000   Median :0.0000   Median :0.0000      Median :0.0000
##  Mean   : 0.3273   Mean   :0.1517   Mean   :0.2672      Mean   :0.3746
##  3rd Qu.: 1.0000   3rd Qu.:0.0000   3rd Qu.:1.0000      3rd Qu.:1.0000
##  Max.   : 3.0000   Max.   :1.0000   Max.   :1.0000      Max.   :1.0000
##  NA's   :385
##   region.west     region_Northeast region_North Central  region_South
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000       Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000       1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000       Median :0.0000
##  Mean   :0.2065   Mean   :0.1517   Mean   :0.2672       Mean   :0.3746
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:1.0000       3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000       Max.   :1.0000
##
##   region_West
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2065
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
```

A simulation example

The above example uses real survey data, and you can adjudicate yourself how much of a difference the inclusion of the dummy variables as control variables makes to your estimates. However, we can also simulate data where the results from a model without and with the dummy variable differ dramatically. In this example, I simulate a continuous variable x_1, a binary (dummy) variable x_2, and a variable y that is first created from a DGP so that $y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$. Next, I add 1 to the values of x_1 for all observations where $x_2 = 1$.

```
set.seed(123)
n.obs <- 100
x1 <- runif(n = n.obs, min = -1, max = 1)
x2 <- rbinom(n = n.obs, size = 1, prob = 0.5)
e <- rnorm(n.obs, mean = 0, sd = 2)
a <- 0.2
b1 <- 2.5
b2 <- -7.5
y <- a + b1 * x1 + b2 * x2 + e
sim.dat <- data.frame(y, x1, x2)
sim.dat[sim.dat$x2 == 1, ]$x1 <- sim.dat[sim.dat$x2 == 1, ]$x1 + 1
sim.dat$color <- ifelse(sim.dat$x2 == 1, "blue", "red")
```

Now, I fit two models: Model 1 predicts y with x_1, and Model 2 adds the (binary) x_2 as a control variable.

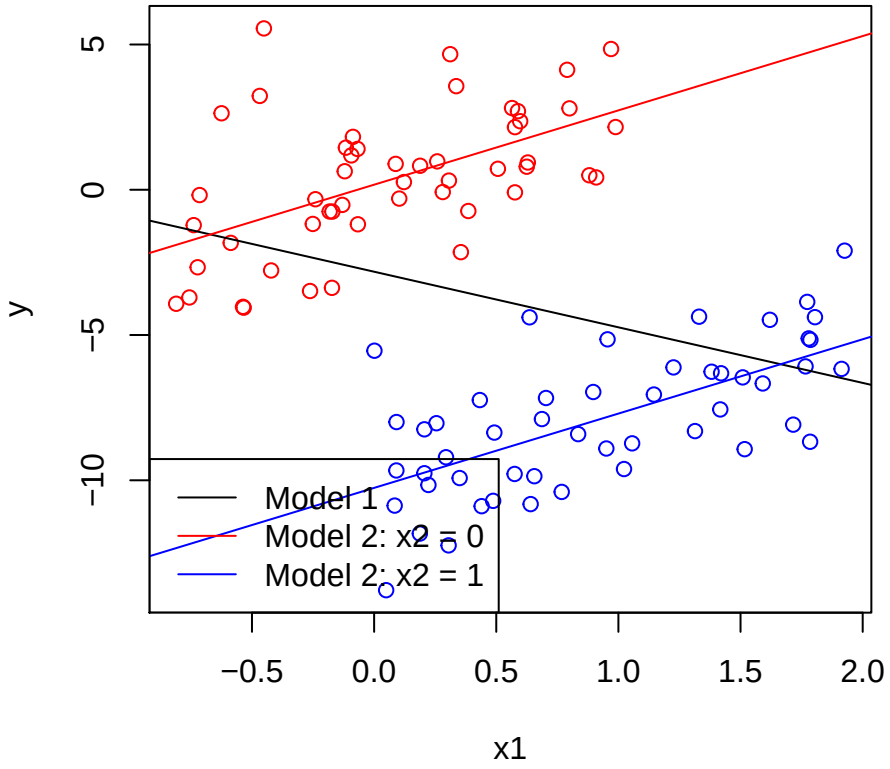
```
m1 <- lm(y ~ x1, data = sim.dat)
m2 <- lm(y ~ x1 + x2, data = sim.dat)
library(texreg)
screenreg(list(m1, m2))
```

```
##
## ===================================
##              Model 1     Model 2
## -----------------------------------
## (Intercept)   -2.82 ***    0.17
##               (0.56)      (0.28)
## x1            -1.91 **     2.56 ***
##               (0.65)      (0.35)
## x2                       -10.44 ***
##                           (0.49)
## -----------------------------------
## R^2            0.08        0.84
## Adj. R^2       0.07        0.84
## Num. obs.    100         100
## ===================================
## *** p < 0.001, ** p < 0.01, * p < 0.05
```

5. How do your conclusions about the relationship between x_1 and y from the results above differ between Model 1 and 2?

If you visualize the two models and their predictions, the difference becomes pretty clear:

```
with(sim.dat, plot(x = x1, y = y, col = color))
abline(a = coef(m1)[1], b = coef(m1)[2])
abline(a = coef(m2)[1], b = coef(m2)[2], col = "red")
abline(a = coef(m2)[1] + coef(m2)[3], b = coef(m2)[2], col = "blue")
legend("bottomleft", legend = c("Model 1", "Model 2: x2 = 0", "Model 2: x2 = 1"),
       col = c("black", "red", "blue"), lty = 1)
```



We will work with dummy variables again on Day 11, and you will encounter them in your midterm exam (based on the treatment in this tutorial.)

Dealing with unusual and influential data points

The main part of Day 9's class is dedicated to dealing with unusual and influential data points. You already encountered the role of unusual and influential data points in the in-class exercise 2.1 on Day 2. The remainder of this tutorial follows closely chapter 11 in *AR*; please refer to that reading for explanations of the concepts used below.

Notation and definitions

As you read in the introduction to chapter 11 in *AR*, problems with unusual and influential data can often be dealt with long before you fit a statistical model - through examining data, building careful theoretical models, and transforming data where necessary. If they do occur, though, unusual data points are

> "problematic in linear models fit by least squares because they can unduly influence the results of the anlaysis and because their presence may be a signal that the model fails to capture important characteristics of the data." (*AR*, p. 241)

Unsual data points are also referred to as outliers. Outliers are"conditionally unusual [data points] given the value of the explanatory variable." (*AR*, p. 241). This means that you should consider residuals when you think about outliers. Residuals are defined as

$$\varepsilon_i = y_i - \hat{y}$$

Because residuals do not have equal variance, they cannot be directly compared when your goal is to assess how unusual individual observations are. For this reason, we often use studentized residuals to express the "unusualness" of data points on a comparable scale (see more in *AR*, p. 246-247):

$$\varepsilon_{Ti} = \frac{\varepsilon_i}{\hat{\sigma}_{-i}\sqrt{1 - h_i}}$$

Unusual data points alone, however, are not necessarily a reason for concern when you are interested in using OLS to make inferences. Concern is most warranted from a combination of "unusualness" *and* leverage on your parameter estimates. In *AR*'s words:

$$\text{Influence on coefficients} = \text{Leverage} \times \text{Discrepancy}$$

This symbolic equation guides the remainder of this tutorial.

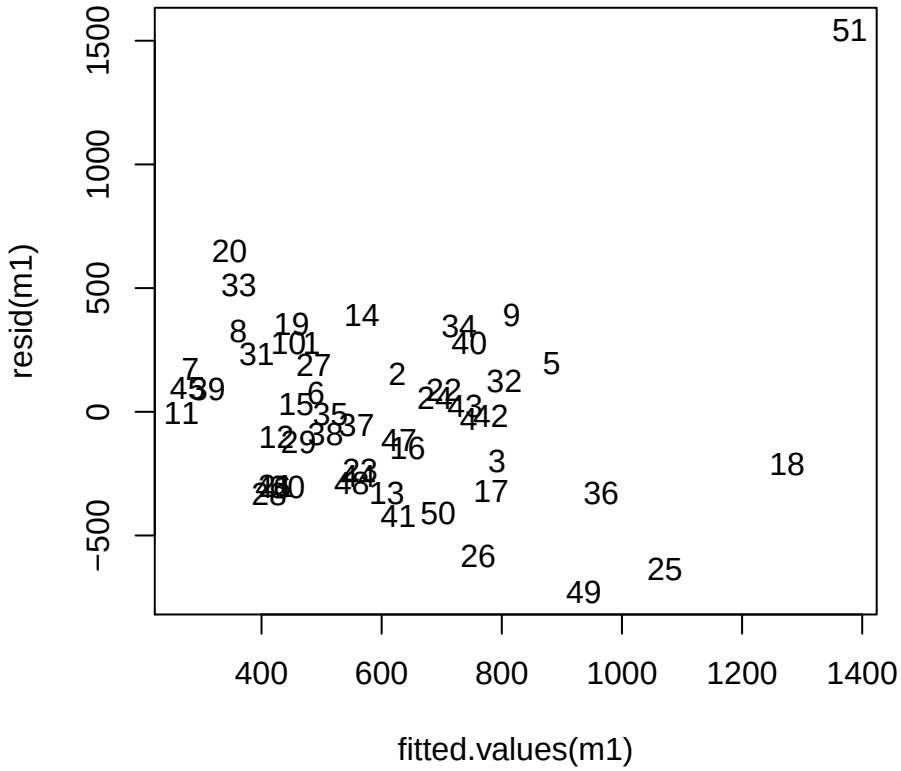
Example data: crime rates in the United States

As example data, we use the crime dataset that appears in *Statistical Methods for Social Sciences, Third Edition* by Alan Agresti and Barbara Finlay (Prentice Hall, 1997), and that is provided by UCLA's Institute for Digital Research and Education. The dataset has 51 rows (each row is one state) and 9 variables. The variables are:

Variable	Description
sid	State ID
state	State name
crime	Violent crimes per 100,000 people
murder	Murders per 1,000,000 people
pctmetro	Percent of the population living in metropolitan areas
pctwhite	Percent of the population that is white
pcths	Percent of population with a high school education or above
poverty	Percent of population living under poverty line
single	Percent of population that are single parents

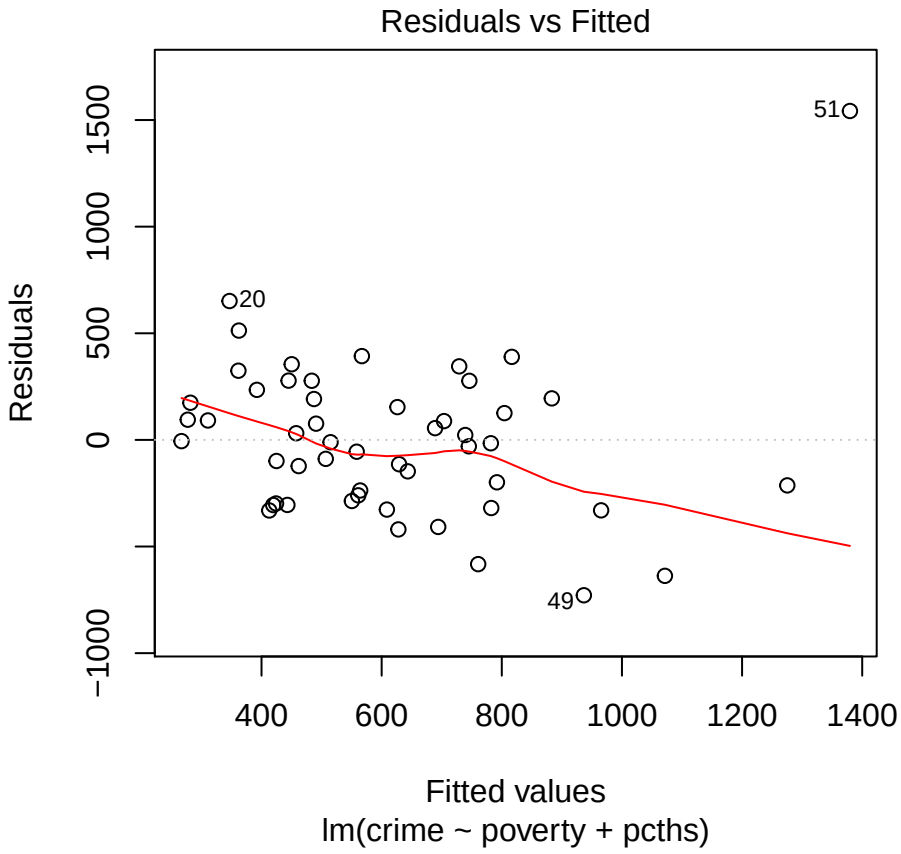
First, we estimate a regression model, predicting the crime rate with the poverty rate and the percent of population with a high school education (or above). After fitting the model, we can create a scatterplot of fitted values and residuals to check any conditionally unusual observations.

```
library(foreign)
crime.dat <- read.dta("http://www.ats.ucla.edu/stat/data/crime.dta")
m1 <- lm(crime ~ poverty + pcths, data = crime.dat)
plot(x = fitted.values(m1), y = resid(m1), type = "n")
text(x = fitted.values(m1), y = resid(m1), labels = names(fitted.values(m1)))
```



R also offers a built-in residual plot that labels the most unusual points (with the row number of that observation). Otherwise, this plot is identical to the one you just created "by hand". See `?plot.lm` for more information.

```
plot(m1, which = c(1))
```



6. What do you conclude from this residual plot about any of the regression assumptions you've encountered in Days 6 and 8?

It looks like observations 20, 49, and 51 are somewhat unusual. Let's look at these cases in the data by calling these rows individually - remember, you can access individual rows of a dataframe by specifying their numbers before the comma in the hard brackets, and you can access individual commas by doing the same after the comma.

```
crime.dat[c(20, 49, 51), ]
```

```
##    sid state crime murder pctmetro pctwhite pcths poverty single
## 20  20    md   998   12.7     92.8     68.9  78.4     9.7   12.0
## 49  49    wv   208    6.9     41.8     96.3  66.0    22.2    9.4
## 51  51    dc  2922   78.5    100.0     31.8  73.1    26.4   22.1
```

Outliers

We can calculate studentized residuals (see above) and add them as a new variable to our dataset. Then, I use the `arrange()` function from the "dplyr" package to conveniently sort the dataset by the absolute value of the studentized residuals. Studentized residuals can be positive or negative, so sorting them in ascending descending order won't do what I want here. The `arrange()` function can be applied to any data frame. Its first argument is the dataframe to which you want to apply it. The second argument takes the variable by which you want to sort the data. Here, I use two nested commands. First, I use `abs(m1.studentized.resid)` rather than only the variable name `m1.studentized.resid` because I'd like to sort by the absolute values of

the variable. Next, I nest this term in `desc()`. This command stands for "sort in `desc`ending order." Lastly, I use again row indexing to show me only the first 10 rows of the sorted dataframe.

```
crime.dat$m1.studentized.resid <- rstudent(m1)
library(dplyr)
arrange(crime.dat, desc(abs(m1.studentized.resid)))[1:10, ]
```

```
##    sid state crime murder pctmetro pctwhite pcths poverty single
## 1   51    dc  2922   78.5    100.0     31.8  73.1    26.4   22.1
## 2   49    wv   208    6.9     41.8     96.3  66.0    22.2    9.4
## 3   25    ms   434   13.5     30.7     63.3  64.3    24.7   14.7
## 4   20    md   998   12.7     92.8     68.9  78.4     9.7   12.0
## 5   26    mt   178    3.0     24.0     92.6  81.0    14.9   10.8
## 6   33    nv   875   10.4     84.8     86.7  78.8     9.8   12.4
## 7   41    sd   208    3.4     32.6     90.2  77.1    14.2    9.4
## 8   50    wy   286    3.4     29.7     95.9  83.0    13.3   10.8
## 9   14    il   960   11.4     84.0     81.0  76.2    13.6   11.5
## 10   9    fl  1206    8.9     93.0     83.5  74.4    17.8   10.6
##    m1.studentized.resid
## 1              6.339983
## 2             -2.091204
## 3             -1.853594
## 4              1.801713
## 5             -1.615579
## 6              1.397613
## 7             -1.124149
## 8             -1.121716
## 9              1.050410
## 10             1.048094
```

7. What would you need to do to show you the first 5 rows of the dataframe sorted by the crime rate?

I can also use the `outlierTest()` function from the "car" package to obtain "statistically significant" outliers. This test is based on the Bonferroni p-value as it is explained in section 11.3.1 of *AR*. You can find out more about this function via `?outlierTest`. Don't forget to load the "car" package before using this function.

```
library(car)
outlierTest(m1, cutoff = 0.1)
```

```
##    rstudent unadjusted p-value Bonferonni p
## 51 6.339983         8.2151e-08   4.1897e-06
```

With tests like these, please keep in mind the warning in section 11.5 of *AR*: the numerical cutoffs used for these tests are not a be-all end-all criterion for how you should deal with outliers. Rather, a careful combination of numerical and visual tests as well as inspection of the cases should guide your practice.

Leverage

To diagnose leverage on OLS coefficients, *AR* recommends to investigate hat values (section 11.2). You can easily access them using the `hatvalues()` function (no R package needed) and again add them to your dataset and sort the dataset to see which observations have the highest hat values.

```
crime.dat$m1.hatvalues <- hatvalues(m1)
arrange(crime.dat, desc(m1.hatvalues))[1:10, ]
```

```
##    sid state crime murder pctmetro pctwhite pcths poverty single
## 1   51    dc  2922   78.5    100.0     31.8  73.1    26.4   22.1
## 2   18    la  1062   20.3     75.0     66.7  68.3    26.4   14.9
## 3   25    ms   434   13.5     30.7     63.3  64.3    24.7   14.7
## 4   17    ky   463    6.6     48.5     91.8  64.6    20.4   10.6
## 5   39    ri   402    3.9     93.6     92.6  72.0    11.2   10.8
## 6   49    wv   208    6.9     41.8     96.3  66.0    22.2    9.4
## 7    1    ak   761    9.0     41.8     75.2  86.6     9.1   14.3
## 8    2    al   780   11.6     67.4     73.5  66.9    17.4   11.5
## 9    3    ar   593   10.2     44.7     82.9  66.3    20.0   10.7
## 10  45    va   372    8.3     77.5     77.1  75.2     9.7   10.3
##    m1.studentized.resid m1.hatvalues
## 1             6.3399832   0.24915431
## 2            -0.6160271   0.17360933
## 3            -1.8535942   0.13194366
## 4            -0.8919457   0.10793022
## 5             0.2513072   0.09867824
## 6            -2.0912037   0.09265962
## 7             0.7655789   0.09138438
## 8             0.4221204   0.08900606
## 9            -0.5448110   0.08279538
## 10            0.2582032   0.07752945
```

Influence

In section 11.4 of *AR*, you encounter Cook's D(istance) statistic as a measure of influence. Again, you can use the appropriately named function in R to obtain these values and sort the dataset by them.

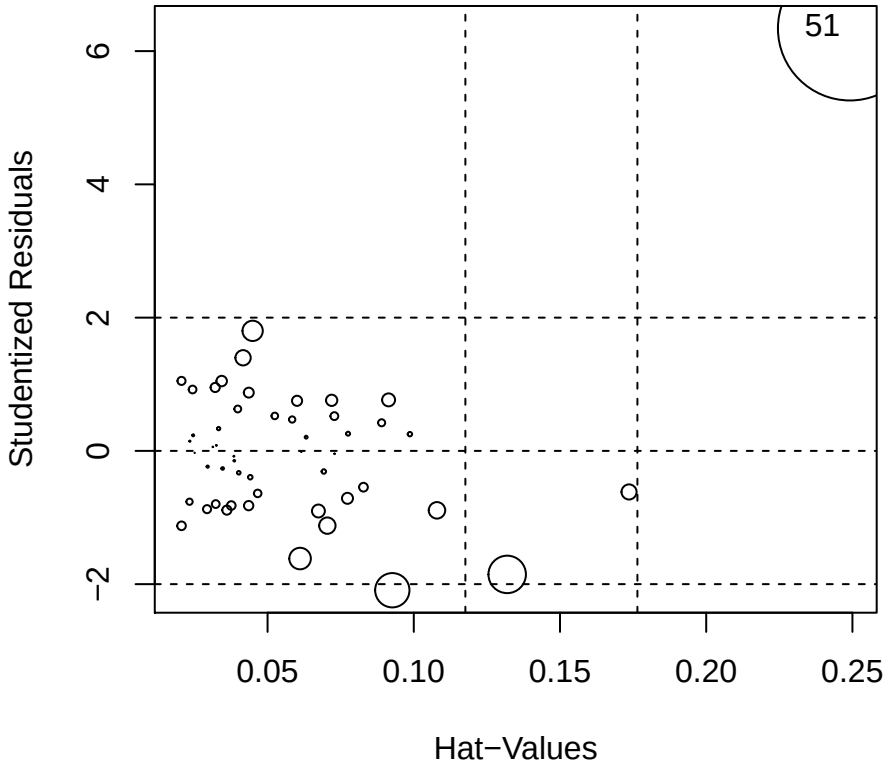
```
crime.dat$m1.cooksD <- cooks.distance(m1)
arrange(crime.dat, desc(m1.cooksD))[1:10, ]
```

```
##    sid state crime murder pctmetro pctwhite pcths poverty single
## 1   51    dc  2922   78.5    100.0     31.8  73.1    26.4   22.1
## 2   25    ms   434   13.5     30.7     63.3  64.3    24.7   14.7
## 3   49    wv   208    6.9     41.8     96.3  66.0    22.2    9.4
## 4   26    mt   178    3.0     24.0     92.6  81.0    14.9   10.8
## 5   20    md   998   12.7     92.8     68.9  78.4     9.7   12.0
## 6   17    ky   463    6.6     48.5     91.8  64.6    20.4   10.6
## 7   50    wy   286    3.4     29.7     95.9  83.0    13.3   10.8
## 8   33    nv   875   10.4     84.8     86.7  78.8     9.8   12.4
## 9   18    la  1062   20.3     75.0     66.7  68.3    26.4   14.9
## 10   1    ak   761    9.0     41.8     75.2  86.6     9.1   14.3
##    m1.studentized.resid m1.hatvalues  m1.cooksD
## 1             6.3399832   0.24915431 2.44748707
## 2            -1.8535942   0.13194366 0.16567269
## 3            -2.0912037   0.09265962 0.13909032
## 4            -1.6155787   0.06112029 0.05480014
## 5             1.8017133   0.04489217 0.04858550
## 6            -0.8919457   0.10793022 0.03222207
```

```
## 7            -1.1217157   0.07044526 0.03161483
## 8             1.3976125   0.04162281 0.02772722
## 9            -0.6160271   0.17360933 0.02692259
## 10            0.7655789   0.09138438 0.01982036
```

The "car" package also contains the `influencePlot()` function, which you can use to plot leverage, discrepancy, *and* influence all in one plot. The y-axis (studentized residuals) indicates how unusual a data point is, the x-axis (hat-values) shows its leverage on the coefficients, and the size of the bubbles in the plot indicates the Cook's D value of each data point. See Figure 11.5 in *AR* and `?influencePlot` for reference.

```
influencePlot(m1)
```

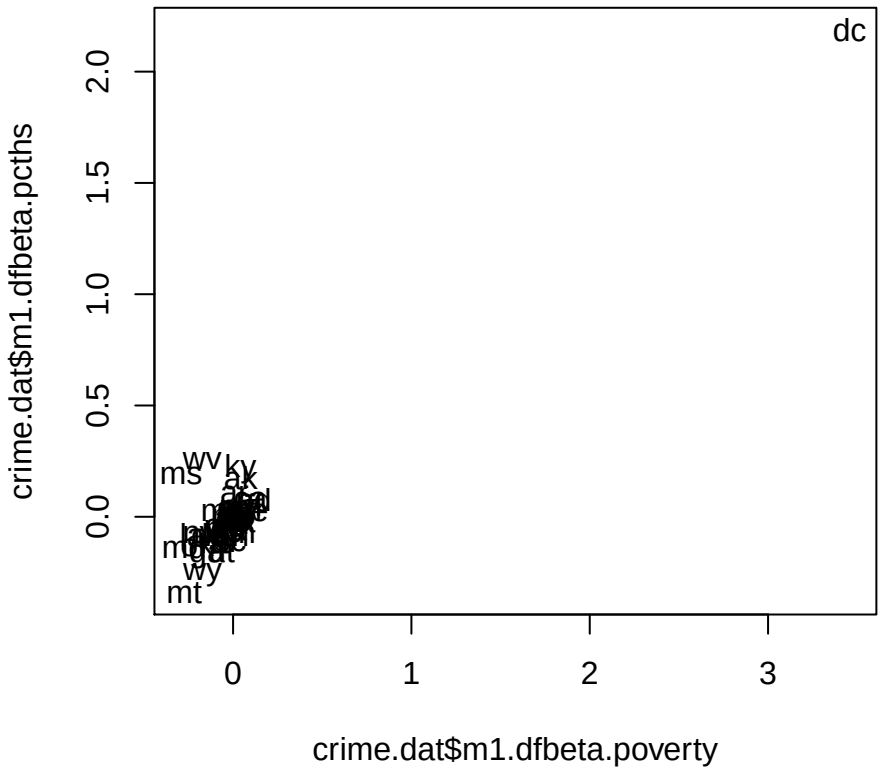



```
##     StudRes       Hat    CookD
## 51 6.339983 0.2491543 1.564445
```

8. What do you conclude from this influence plot?

An alternative but related measure of influence are dfbetas (see *AR* section 11.4). You can add dfbetas to your data frame and then create a scatterplot of them, using state IDs as labels. Then, you will easily see which observation(s) exercise high influence on both coefficients.

```
crime.dat$m1.dfbeta.poverty <- dfbetas(m1)[, c("poverty")]
crime.dat$m1.dfbeta.pcths <- dfbetas(m1)[, c("pcths")]
plot(x = crime.dat$m1.dfbeta.poverty,
     y = crime.dat$m1.dfbeta.pcths,
     type = "n")
text(x = crime.dat$m1.dfbeta.poverty,
     y = crime.dat$m1.dfbeta.pcths,
     labels = crime.dat$state)
```



9. What is the meaning of the scale of the x- and y-axes in the plot above?

Dropping observations

In section 11.6 of *AR*, you learn about the joint influence of data points on regression coefficients. You can see what is meant by this by comparing regression results from two models. Above, it seemed as if the District of Columbia exerts an undue influence on your regression results. So let's estimate our model from above, but drop observation 51 (DC) from the data.

```
m2 <- lm(crime ~ poverty + pcths, data = crime.dat[-c(51), ])
library(texreg)
screenreg(list(m1, m2))
```

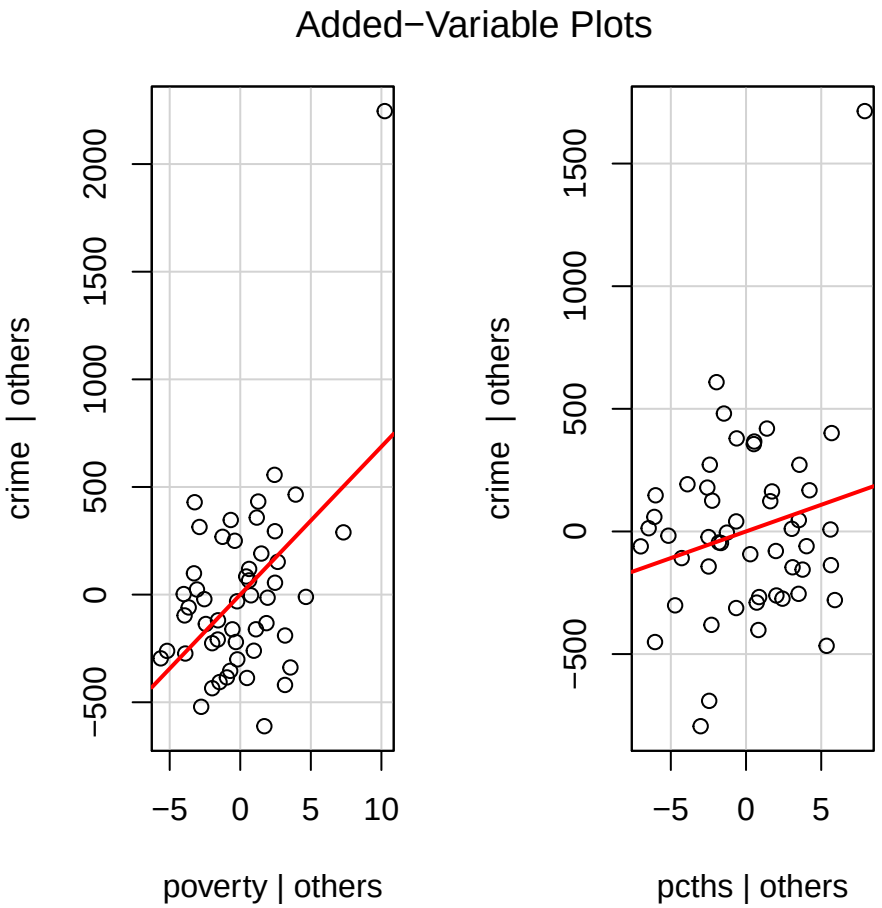
```
##
## ====================================
##              Model 1       Model 2
## ------------------------------------
## (Intercept)  -2023.27        345.85
##              (1288.79)     (1026.64)
## poverty         68.74 ***     23.93
##                (17.47)       (14.76)
## pcths           21.72         -1.50
##                (14.32)       (11.24)
## ------------------------------------
## R^2              0.29          0.14
## Adj. R^2         0.26          0.10
## Num. obs.       51            50
```

```
## ===================================
## *** p < 0.001, ** p < 0.01, * p < 0.05
```

10. Does dropping DC change your conclusions from this regression model?

You can also use added-variable plots to see whether observations might exert leverage on the respective coefficients. This might be useful if you don't want to drop observations based on previous diagnostics and explore your data before continuing. The "car" package provides the `avPlot()` function, and these plots are explained in more detail in section 11.6.1 in *AR*.

```
avPlots(m1)
```



At the end of this tutorial, please consider the advice in section 11.7 of *AR*:

> "Outlying and influential data should not be ignored, but they also should not simply be deleted without investigation."

Outlying and influential data are often a good starting point for further analysis that may reveal new and important insights.

11. Now try on your own. Download a dataset of social indicators provided by the United Nations. The dataset is available on John Fox's webpage at http://socserv.socsci.mcmaster.ca/jfox/Books/Applied-Regression-3E/datasets/index.html under the name "UnitedNations.txt". Read it into R using the `read.table()` function and the URL of the dataset. Then, estimate a regression model predicting infant mortality, with a set of at least 2 predictors you think are theoretically related to infant mortality. Check whether outliers influence your inferences. If yes, how do you propose to deal with these outliers?