Regression Diagnostics with R

Anne Boomsma

Department of Statistics & Measurement Theory
University of Groningen

April 26, 2012

`Regrdiag_R.tex`

Regression Diagnostics with R

Anne Boomsma

Department of Statistics & Measurement Theory, University of Groningen

1. Introduction

In our opinion, the best start for regression applications in R is either Faraway's (2005) book *Linear models with R*, or Fox's (2002) *R and S-Plus companion to applied regression.* In this document, we present an overview of regression diagnostics using material from chapter four of Faraday's book mainly. When running through the examples, the power of the R environment will become unmistakably clear, especially in the versatility of its graphical options.

First, install the packages `faraway` (Faraway), `car` (Fox), and `lmtest` (R) from a Comprehensive R Archive Network (`CRAN`) mirror by choosing Packages → Install packages at the upper tool bar of `RGui` (R's Graphical user interface). Next, load the `faraway` package, and from that package data frame `savings`.

```
> library(faraway)                        # loading package 'faraway'
> data(savings)                           # documentation on data set 'sexab'
```

The command `attach(savings)` is not strictly necessary in the sequel, nor recommended here: for some commands, country labels would vanish in the output.

```
> ? savings                               # documentation of "Savings rates"
```

This data frame contains the savings rates in `n = 50` countries (source: Belsley, Kuh & Welsch, 1980). The data are averaged over the period 1960–1970. The data frame `(50 x 5)` contains the following objects or variables:

`sr`	savings rate – personal saving divided by disposable income
`pop15`	percent population under age of `15`
`pop75`	percent population over age of `75`
`dpi`	per-capita disposable income in dollars
`ddpi`	percent growth rate of `dpi`

```
> savings                                # list complete data frame 'savings'
```

The linear regression model `M1` for response variable savings rate `sr` is specified and estimated as follows:

```
> M1 <- lm(sr ~ pop15 + pop75 + dpi + ddpi, data=savings)
> (M1_sum <- summary(M1))                 # summary of estimated model

  Call:
  lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = savings)

  Residuals:
     Min      1Q Median      3Q     Max
  -8.242 -2.686 -0.249  2.428  9.751

  Coefficients:
               Estimate Std. Error t value Pr(>|t|)
  (Intercept) 28.566087   7.354516    3.88  0.00033 ***
  pop15       -0.461193   0.144642   -3.19  0.00260 **
  pop75       -1.691498   1.083599   -1.56  0.12553
  dpi         -0.000337   0.000931   -0.36  0.71917
  ddpi         0.409695   0.196197    2.09  0.04247 *
  ---
  Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

  Residual standard error: 3.8 on 45 degrees of freedom
  Multiple R-squared: 0.338,      Adjusted R-squared: 0.28
  F-statistic: 5.76 on 4 and 45 DF,  p-value: 0.00079
```

▷ Check whether the details of this summary are well understood.

```
> options(show.signif.stars=F)            # suppress stars of significance
> options(digits=4)                       # set numbers of significant digits
```

The fitted values $\hat{Y}_i$ and the residuals e_i can be obtained as follows:

```
> fitted(M1)                              # predicted Y: $\hat{Y}_i$
> residuals(M1)                           # residuals $e_i$
> which.max(abs(residuals(M1)))           # largest absolute residual $|e_i|$?

    Zambia
        46
```

▷ Diagnostic purpose of residuals: locating large errors in prediction.

2. Checking model assumptions

We need to inspect the validity of the main assumptions of the linear regression model. This refers, first of all, to the (conditional) distribution of the model's errors terms ϵ_i: homogeneous variance, normality, and independence. Analysis of observed residuals e_i may help to evaluate the plausibility of these assumptions. Checking for unusual and influential observations is another part of regression diagnostics. In addition, the validity of the structural model itself, i.e., its linearity $E(Y) = \mathbf{X}\boldsymbol{\beta}$ and the selection of explanatory variables, should be examined.

2.1 Constant variance

- Residual plot: $\hat{Y}_i$ against e_i

```
> par(las=1)                             # horizontal style of axis labels
> plot(fitted(M1), residuals(M1), xlab="Fitted", ylab="Residuals")
> abline(h=0, col="red")                 # draws a horizontal red line at y = 0
```

There are a number of specific plot diagnostics for an `lm()` object, which allow for standard plotting jobs — all available in the built-in `stats` package.

```
> ? plot.lm
> plot(M1, which=1)                     # Ŷi against ei
> plot(M1, ask=TRUE)                    # all six standard lm() plots available
```

- Absolute residual plot: $\hat{Y}_i$ against $|e_i|$

```
> plot(fitted(M1), abs(residuals(M1)), xlab="Fitted", lab="|Residuals|")
```

This plot is designed to check for constant variance only.

- Absolute residual plot: $\hat{Y}_i$ against `sqrt(standardized` $|e_i|$`)`

```
> plot(M1, which=3)          # R's standardized residuals scale-location plot
```

- Quick and dirty test

Faraway (2005) mentions the following F-test as a quick way to check non-constant variance by a regression of $|e_i|$ on $\hat{Y}_i$, where $|e_i|$ is the response and $\hat{Y}_i$ the explanatory variable.

```
> summary(lm(abs(residuals(M1)) ~ fitted(M1)))
```

```
Call:
lm(formula = abs(residuals(M1)) ~ fitted(M1))

Residuals:
    Min      1Q  Median      3Q     Max
-2.8395 -1.6078 -0.3493  0.6625  6.7036

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.8398     1.1865   4.079 0.000170
fitted(M1)    -0.2035     0.1185  -1.717 0.092501

Residual standard error: 2.163 on 48 degrees of freedom
Multiple R-Squared: 0.05784,    Adjusted R-squared: 0.03821
F-statistic: 2.947 on 1 and 48 DF,  p-value: 0.0925
```

It turns out that the absolute residuals $|e_i|$ are not predicted very well by $\hat{Y}_i$, which is roughly satisfying. Hence, we conclude that there does not seem to be a problem with the constant variance assumption.

- Interpretation of residual plots

For a proper evaluation of residual plots it may be helpful to generate some artificial plots for situations where true relationships are known.

```
> par(mfrow=c(3,3))            # setting a plot device, here a (3 x 3) matrix
                               # with three plots in each of the three rows;
                               # first, an empty plot window pops up

> for(i in 1:9) plot(1:50,rnorm(50))                        # constant variance
> for(i in 1:9) plot(1:50,(1:50)*rnorm(50))                 # strong heterogeneity
> for(i in 1:9) plot(1:50,sqrt((1:50))*rnorm(50))           # mild heterogeneity
> for(i in 1:9) plot(1:50,cos((1:50)*pi/25)+rnorm(50)) # non-linearity
```

2.2 Normality

- Q-Q plots

Observed ordered residuals e_i (the sample quantiles at the y-axis) are plotted against expected normal quantiles $\Phi^{(-1)}[i/(n+1)]$ at the x-axis, where $\Phi(x)$ is the standard normal distribution function, i.e., $\Phi(x) = Pr(X < x)$. Recall that $e_i \sim \mathcal{N}(0, \sigma^2)$.

For the `savings` data we try the following:

```
> par(mfrow=c(1,1))                             # reset plotting device
> qqnorm(residuals(M1), ylab="Residuals")       # Q-Q plot
> qqline(residuals(M1))                         # line through Q1 and Q3
```

- Interpretation of Q-Q plots

To get an idea of the variation to be expected in a Q-Q plot, inspect the plots generated for a number of probability distributions. In the examples below, we use the standard normal, the lognormal, Student's t with one degree of freedom, and the uniform $\mathcal{U}(0,1)$ distribution, respectively. Nine independent pseudo-random samples of size 50 are generated from each distribution. For each sample, a Q-Q plot with a quartile-line is produced.

```
> par(mfrow=c(3,3))
> for(i in 1:9) x = rnorm(50); qqnorm(x); qqline(x)
      # i.e., standard normal distribution (symmetric)


> for(i in 1:9) x = rlnorm(50); qqnorm(x); qqline(x)
      # lognormal distribution (long right tail, skew to right)


> for(i in 1:9) x = rt(50,1); qqnorm(x); qqline(x)
      # Student t-distribution with one df (heavy tails, platykurtic)


> for(i in 1:9) x = runif(50); qqnorm(x); qqline(x)
      # uniform (0,1) distribution (short tails, leptokurtic)
```

If the errors ϵ_i are not normal, the least squares estimates may not be optimal. They will still be best linear unbiased estimates, but other robust estimators may be more effective. Tests and confidence intervals may not be exact. Long-tailed distributions in particular, cause large inaccuracies. Mild non-normality may be safely ignored, according to Faraway (2005, p. 59), but we may need more specificity here.

- Histograms and box plots

Histograms and box plots graphs are also suitable for checking normality, along with descriptive statistics like skewness and kurtosis, for example.

```
> par(mfrow=c(1,1))
> hist(residuals(M1))
> boxplot(residuals(M1))
```

- Shapiro-Wilks normality test

```
> shapiro.test(residuals(M1))

     Shapiro-Wilk normality test

   data:  residuals(M1)
   W = 0.987, p-value = 0.8524
```

The null hypothesis is that the residuals have a normal distribution. The p-value of the test statistic is large in this example. It thus follows that the null hypothesis is not rejected. Faraway (2005) only recommends this test in conjunction with a Q-Q plot. For large samples the test may be too sensitive, and for small samples its power may be too small – the usual dilemma.

2.3 Independent errors

The data set `airquality` from the `datasets` package serves as a more appropriate illustration here than the `savings` data. The data are daily air quality measurements in New York, from May to September 1973 (source: Chambers, Cleveland, Kleiner & Tukey, 1983). We have a data frame with `n = 153` observations on `6` numerical variables.

`Ozone`	Ozone (ppb – in parts per billion particles)
`Solar.R`	Solar R (Solar radiation in Langleys)
`Wind`	Wind (mph)
`Temp`	Temperature (degrees F)
`Month`	Month (1–12)
`Day`	Day of month (1–31)

```
> airquality                                    # notice missing values (NAs)
> attach(airquality)
> names(airquality)
```

- Scatter plots

Take a look at scatter plots first. The function `pairs()` produces a matrix of scatter plots for all pairs of variables in a data frame.

```
> pairs(airquality, panel=panel.smooth)        # matrix of scatter plots
```

Inspection of correlations for linear relationships (listwise deletion of missing cases), given these scatter plots, can be illustrative too.

```
> round(cor(airquality, use="complete.obs"), digits=2)
```

Next a linear regression model `M2` for `Ozone` is fitted to the data, where `Month` and `Day` are not used as linear predictors.

```
> M2 <- lm(Ozone ~ Solar.R + Wind + Temp, data=airquality,
+  na.action=na.exclude)
> summary(M2)                        # summary of the estimated linear model

   Call:
   lm(formula = Ozone ~ Solar.R + Wind + Temp, data = airquality,
       na.action = na.exclude)

   Residuals:
       Min      1Q  Median      3Q     Max
   -40.485 -14.219  -3.551  10.097  95.619

   Coefficients:
                Estimate Std. Error t value Pr(>|t|)
   (Intercept) -64.34208   23.05472  -2.791  0.00623
   Solar.R       0.05982    0.02319   2.580  0.01124
   Wind         -3.33359    0.65441  -5.094 1.52e-06
   Temp          1.65209    0.25353   6.516 2.42e-09

   Residual standard error: 21.18 on 107 degrees of freedom
   Multiple R-Squared: 0.6059,     Adjusted R-squared: 0.5948
   F-statistic: 54.83 on 3 and 107 DF,  p-value: < 2.2e-16

> table(complete.cases(airquality))  # number of complete cases
```

We notice that the data frame has missing values. There are `111` complete cases only. The default with respect to missing values for regression analysis in R is to omit any case that contains a missing value. The option `na.action=na.exclude` does not use cases with missing values in the computation but keeps track of which cases are missing in the residual, fitted values and other quantities.

Residual diagnostics show some non-constant variance and non-linearity—see the previous `pairs()` plots. Therefore, a logarithmic transformation of the response variable `Ozone` is made, resulting in model `M2_log`.

- Transformation of the response variable

```
> M2_log <- lm(log(Ozone) ~ Solar.R + Wind + Temp, airquality,
+  na.action=na.exclude)
> summary(M2_log)                        # summary of the estimated linear model

   Call:
   lm(formula = log(Ozone) ~ Solar.R + Wind + Temp, data = airquality,
       na.action = na.exclude)

   Residuals:
         Min        1Q    Median        3Q       Max
   -2.061929 -0.299696 -0.002312  0.307559  1.235783

    Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
   (Intercept) -0.2621323  0.5535669  -0.474 0.636798
   Solar.R      0.0025152  0.0005567   4.518 1.62e-05
   Wind        -0.0615625  0.0157130  -3.918 0.000158
   Temp         0.0491711  0.0060875   8.077 1.07e-12

   Residual standard error: 0.5086 on 107 degrees of freedom
   Multiple R-Squared: 0.6644,     Adjusted R-squared: 0.655
   F-statistic: 70.62 on 3 and 107 DF,  p-value: < 2.2e-16
```

▷ Notice the improvement of fit of model `M2_log` over that of model `M2`, where `Ozone` was untransformed.

We now check for correlated error terms. Recall that there is a time component in the `airquality` data.

- Index plot of residuals e_i, i.e., a plot of e_i against time

```
> par(las=1, mfrow=c(1,1))
> plot(residuals(M2_log), ylab="Residuals")
> abline(h=0)
```

If there was serial correlation, we would see either long runs of residuals above or below the line for positive correlation, or greater than normal fluctuations for negative correlation. Unless the effects are strong, they may be difficult to detect. Therefore, it is often better to plot successive residuals.

- Plot of successive residuals e_i `[1,152]` against e_{i+1} `[2,154]`

```
> plot(residuals(M2_log)[-153], residuals(M2_log)[-1],
+  xlab=expression(hat(epsilon)[i]), ylab=expression(hat(epsilon)[i+1]))
```

No obvious problem with correlated errors is shown. There is an outlier though, which we may try to identify. Is there really only one outlier?

```
> identify(residuals(M2_log)[-153], residuals(M2_log)[-1], n=4)
```

- Regression of e_{i+1} `[response]` on e_i `[explanatory variable]`

```
> summary(lm(residuals(M2_log)[-1] ~ -1 + residuals(M2_log)[-153]))
```

```
   Call:
   lm(formula = residuals(M2_log)[-1] ~ -1 + residuals(M2_log)[-153])

   Residuals:
        Min        1Q    Median        3Q       Max
   -2.07274 -0.28953  0.02583  0.32256  1.32594

   Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
   residuals(M2_log)[-153]    0.1104     0.1053   1.048    0.297

   Residual standard error: 0.5078 on 91 degrees of freedom
   Multiple R-Squared: 0.01193,    Adjusted R-squared: 0.001073

   F-statistic: 1.099 on 1 and 91 DF,  p-value: 0.2973
```

This regression model of successive residuals omits the intercept term, `-1`, because the mean of the residuals is zero, by definition.

Clearly, there is no substantive correlation (take the square root of `R-Squared`, which gives `0.10922`), also to be shown as follows:

```
> cor(residuals(M2_log)[-1], residuals(M2_log)[-153], use="complete.obs")
```

```
[1] 0.1092547
```

- Durbin-Watson test

The function for this test statistic is implemented in the `lmtest` package.

```
> library(lmtest)
  Loading required package:  zoo         # a message that can be ignored
                                         # zoo means Z's Ordered Observations
```

```
> dwtest(Ozone ~ Solar.R + Wind + Temp, data=na.omit(airquality))

    Durbin-Watson test

  data:  Ozone ~ Solar.R + Wind + Temp
  DW = 1.9355, p-value = 0.3347
  alternative hypothesis: true autocorrelation is greater than 0
```

The p value indicates that there is no evidence of correlated errors, but the results should be viewed with skepticism because of the omission of the missing values, according to Faraway (2005). Interestingly, Faraway does not show the test results for `log(Ozone)`, which are slightly worse (`DW = 1.8068, p-value = 0.1334`).

In general, if the errors appear to be correlated, we can use generalized least squares estimation, implemented by the function `gls()`.

- Runs test

A runs test is an alternative to the Durbin-Watson test. The function `runs.test()` in the package `tseries` computes the runs test statistic for randomness of a dichotomous (binary) data series `x`. Its application is not appropriate here, because of missing values `NAs`).

3. Detecting unusual observations

The search for unusual, weird data points and influential observations is as important as checking model assumptions, if not a more crucial task indeed. For illustrations, we return to the `savings` data set in the `faraway` package, and to model `M1` as defined on page 2.

3.1 Leverage points

First, notice that the function `influence()` returns values from four vectors or matrices.

- `hat`: a vector containing the diagonal of the `hat` matrix (see Boomsma, 2010) — the diagonal elements are the so-called leverage points h_i.

- `coefficients`: unless `do.coef` is `FALSE`, a matrix whose ith row contains the resulting change in the estimated coefficients when the ith case is dropped from the regression.

- `sigma`: a vector whose ith element contains the estimate of the residual standard deviation obtained when the ith case is dropped from the regression.

- `wt.res`: a vector of `weighted` (or for class `glm` rather `deviance`) residuals.

For more details, use the following commands:

```
> help(influence)                        # details of the four vectors/matrices
> M1_inf <- influence(M1)                # the listed influence information
> M1_inf$hat                             # leverages h_i of savings data
> summary(M1_inf$hat)

      Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
   0.03730 0.06427 0.07502 0.10000 0.09702 0.53150
```

▷ The purpose of leverages h_i is to detect outliers in explanatory variables X_j.

Outliers, according to Stevens (1992), are values of $h_i > 2p/n$; here $2p/n = 10/50 = 0.20$.

```
> which(M1_inf$hat > 0.20)

  Ireland      Japan      United States     Libya
       21         23                 44        49

> sum(M1_inf$hat)                        # sum equals number of predictors
[1] 5
```

As an efficient alternative, the function `hatvalues()` could be used, as recommended in the R documentation of `influence()`.

```
> hatvalues(M1); sum(hatvalues(M1))
```

- Half-normal plots for leverages

Plot the data against the positive normal quantiles. We are looking for outliers. The function `halfnorm()` is implemented in the `faraway` package.

```
> par(mfrow=c(1,1))
> countries <- rownames(savings)         # stores names of countries
```

```
> halfnorm(lm.influence(M1)$hat, labs=countries, ylab="Leverages")
```

In this half-normal plot, the labels of countries having the two largest leverages are shown by default, see `help(halfnorm)`.

R has a function for `lm()` objects, plotting leverage points against standardized residuals (as defined by R), and ranges of Cooks's distances.

```
> plot(M1, which=5)           # leverage against R's standardized residuals
```

3.2 Outliers

- Standardized residuals

```
> M1_sum <- summary(M1)       # linear model 'M1' for savings rate 'sr'
> M1_sum                      # summary of estimated model, as shown before
```

```
   Call:
   lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = savings)

   Residuals:
       Min      1Q  Median      3Q     Max
   -8.2422 -2.6857 -0.2488  2.4280  9.7509


   Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
   (Intercept) 28.5660865  7.3545161   3.884 0.000334
   pop15       -0.4611931  0.1446422  -3.189 0.002603
   pop75       -1.6914977  1.0835989  -1.561 0.125530
   dpi         -0.0003369  0.0009311  -0.362 0.719173
   ddpi         0.4096949  0.1961971   2.088 0.042471

   Residual standard error: 3.803 on 45 degrees of freedom
   Multiple R-Squared: 0.3385,     Adjusted R-squared: 0.2797
   F-statistic: 5.756 on 4 and 45 DF,  p-value: 0.0007904
```

```
> M1_sum$sig                  # sqrt(MSE)
```

```
  [1] 3.802669
```

The statistic `sig` gives an estimate of residual standard error, as it is called; in fact, `sqrt(MSE)` is an unbiased estimator of the standard deviation of ϵ_i.

```
> zresid <- residuals(M1)/(M1_sum$sig)              # standardized residuals
> zresid                               # standardized residuals and country names
> qqnorm(zresid, ylab="Standardized Residuals")      # Q-Q plot
> abline(0,1)                                        # line 'y = x'
```

Under normality, we expect the points to follow the diagonal line `y = x`, approximately. Compare this Q-Q plot with that for the unstandardized residuals, shown earlier.

▷ Diagnostic purpose of standardized residuals: locating large errors in prediction.

For an overview of the names of the arguments that can be selected from the summary table of the function `lm`, use the following commands:

```
> ? summary.lm                                 # summarizing linear model fits
> M1_sum$r.squared                             # R-squared
> M1_sum$adj.r.squared                         # adjusted R-squared
```

- Studentized residuals

```
> stud <- residuals(M1)/(M1_sum$sig*sqrt(1 - M1_inf$hat))
> stud                                  # Studentized residuals and country names
> qqnorm(stud, ylab="Studentized Residuals")    # Q-Q plot
> abline(0,1)                                   # line 'y = x'
```

Here too, as the Studentized residuals are standardized, we expect the points to follow the diagonal line `y = x`, approximately, if normality holds.

▷ Diagnostic purpose of Studentized residuals: detection of outliers in response variable Y.

Notice that the Studentized residuals `stud`, as defined above, equal the standardized residuals as computed by function `rstandard()` in R.

```
> rstandard(M1)                                 # standardized residuals in R
```

R has a different convention than usual (as in SPSS, for example) in defining standardized [function `rstandard()`] and Studentized residuals [function `rstudent()`], respectively. When computing these residuals for the i-th data point, the R function `rstandard()` uses an unbiased estimator of the standard deviation of the observed residuals (not the standard deviation of the error terms ϵ_i, but that of e_i). This now accounts for the fact that `rstandard` is equivalent with the usual formula for Studentized residuals (see Boomsma, 2010). The R function `rstudent()`, on the other hand, calculates residuals from a regression where all points are used except observation i. The general idea of the functions `rstandard()` and `rstudent()` is to "renormalize the residuals to have unit variance, using an overall and leave-one-out measure of error variance respectively"; see `help(influence.measures)`.

With this knowledge, for plotting objectives we might as well use a fancier plotting function:

```
> plot(M1, which=2)                     # R's standardized residuals Q-Q plot
```

It should also be noticed that the package `stats` incorporates the general function `influence.measures(model)`, which covers a set of subfunctions for regression (leave-one-out deletion) diagnostics. Some of these functions will be addressed now. Again, for an overview see the R documentation:

```
> ? influence.measures                  # regression deletion diagnostics
```

- Jackknifed Studentized residuals

▷ Diagnostic purpose of Studentized deleted residuals: detection of influential observations, as well as for validation purposes.

```
> jack <- rstudent(M1)                  # leave-one-out Studentized residuals
> jack[which.max(abs(jack))]

   Zambia
 2.853558
```

This value of 2.85, the largest Studentized deleted residual, is pretty large for a standard normal scale. But is it an outlier, we should ask. We could test whether this observation is an outlier, using a Student's t-statistic with $n-p-1$ degrees of freedom, where p is the number of predictors in an intercept model. If we would use a Bonferroni correction to have a minimal overall α level of 0.05, and a significance level α/n for each individual test, the critical Bonferroni value is computed as follows. Notice that for the `savings` data $n = 50$ and $p = 5$, hence $df = 44$.

```
> qt(.05/(50*2), 44)            # quantile for two-sided alpha = 0.05
                                # in a Student's t-distribution with df = 44
[1] -3.525801
```

Since 2.85 is less that 3.52, we conclude that Zambia is not an outlier.

The `car` package contains a Bonferroni outlier test which just calculates the very thing:

```
> library(car)
> outlier.test(M1)                # equivalent result from the "car" package


   max|rstudent| = 2.8536, degrees of freedom = 44,
   unadjusted p = 0.0065667, Bonferroni p = 0.32833

   Observation: Zambia
```

3.3 Influential Observations

- Cook's distance

Cook's distance measure is a combination of a residual effect and leverage, as shown by Equation 19 in Boomsma (2010). This combination leads to influence.

▷ Diagnostic purpose of Cook's distance measure: the detection of influential observations; detection of the joint influence of outliers, both in the response variable Y and the explanatory variables X_j.

A half-normal plot can be used to identify influential observations.

```
> (cook <- cooks.distance(M1))
> countries <- rownames(savings)
> halfnorm(cook, 3, labs=countries, ylab="Cook's distance")
> which.max(cook)

     Libya
      49
```

There are efficient alternative options in the `car` package:

```
> plot(cookd(M1))
> identify(1:50, cookd(M1), countries)
```

But there is also a diagnostic `lm` plotting function from R itself, providing direct identifying information:

```
> plot(M1, which=4)               # Cook's distance measure
```

We can also plot leverage points against Cook's distance.

```
> plot(M1, which=6)                    # leverage against Cook's distance
```

If we exclude `Lybia`, we can examine how the fit of the linear regression model changes.

```
> M1_L <- lm(sr ~ pop15 + pop75 + dpi + ddpi, data=savings,
+  subset=(cook < max(cook))
> summary(M1_L)                        # linear model estimates without Libya

   Call:
   lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = savings,
       subset = (cook < max(cook)))

   Residuals:
       Min      1Q  Median      3Q     Max
   -8.0699 -2.5408 -0.1584  2.0934  9.3732

   Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
   (Intercept) 24.5240460  8.2240263   2.982  0.00465
   pop15       -0.3914401  0.1579095  -2.479  0.01708
   pop75       -1.2808669  1.1451821  -1.118  0.26943
   dpi         -0.0003189  0.0009293  -0.343  0.73312
   ddpi         0.6102790  0.2687784   2.271  0.02812

   Residual standard error: 3.795 on 44 degrees of freedom
   Multiple R-Squared: 0.3554,     Adjusted R-squared: 0.2968
   F-statistic: 6.065 on 4 and 44 DF,  p-value: 0.0005617

> M1_inf <- influence(M1); M1_inf$coef
```

Recall that in the coefficients matrix `M1_inf$coef`, the ith row contains the change in the estimated coefficients which results when the ith case is dropped from the regression.

```
> M1_inf$coef[,2]
```

The second column of `M1_inf$coef` is related to the regression coefficient of `pop15`, the first explanatory variable (after the intercept term).

```
> plot(M1_inf$coef[,2], ylab="Change in pop15 coefficient")
> abline(h=0)
> identify(1:50, M1_inf$coef[,2], countries)        # identify plotted points
                                                    # use 'Esc' to leave plot
```

Here, we have plotted the change in the second parameter estimate when a single case is left out. The `identify()` function was used to identify plotted points. The country with the largest change could also be identified with the following command:

```
> which.max(abs(M1_inf$coef[,2]))

     Japan
        23
```

The previous plot should be repeated for the other coefficients. In the last plot, `Japan` is an influential observation. We might therefore examine the effect of removing this country from the sample data.

```
> M1_J <- lm(sr ~ pop15 + pop75 + dpi + ddpi, data=savings,
+  subset=(countries != "Japan"))
> summary(M1_J)                       # linear model estimates without Japan

   Call:
   lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = savings,
       subset = (countries != "Japan"))

   Residuals:
       Min      1Q  Median      3Q     Max
   -7.9969 -2.5918 -0.1150  2.0318 10.1571

   Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
   (Intercept) 23.9401714  7.7839968   3.076  0.00361
   pop15       -0.3679015  0.1536296  -2.395  0.02096
   pop75       -0.9736743  1.1554502  -0.843  0.40397
   dpi         -0.0004706  0.0009191  -0.512  0.61116
   ddpi         0.3347486  0.1984457   1.687  0.09871

   Residual standard error: 3.738 on 44 degrees of freedom
   Multiple R-Squared: 0.277,      Adjusted R-squared: 0.2113
   F-statistic: 4.214 on 4 and 44 DF,  p-value: 0.005649
```

▷ Compare the results of this model with those of the full model.

4. Checking the structure of the model

In this section we check whether the systematic part of the model, $E(\mathbf{Y}) = \mathbf{X}\boldsymbol{\beta}$, is correct. Questions under investigation here are, for example: Does the linearity assumption hold? What is the effect of X_j on Y?

- Added variable plot or partial regression plot

We could regress the Xs on Y without explanatory variable X_j, and get residuals $\hat{\delta}$ which represent Y with the other X-effect (`j' != j`) taken out. Similarly, if we regress X_j on all X except X_j, we get residuals $\hat{\gamma}$, which represent X_j with the other X-effects taken out. The added variable plot shows $\hat{\delta}$ against $\hat{\gamma}$. Look for non-linearity, outliers, and influential observations in the plot.

The estimated slope of a line fitted to this plot is b_j. The partial regression plot shows the marginal relationship between the response and an explanatory variable, after the effect of the other explanatory variables has been removed (partialled out). We focus here on the relationship between one predictor, `pop15`, and the response `sr`.

```
> delta <- residuals(lm(sr ~ pop75 + dpi + ddpi, data=savings))
> gamma <- residuals(lm(pop15 ~ pop75 + dpi + ddpi, data=savings))
> plot(gamma, delta, xlab="pop15 residuals", ylab="savings residuals")
```

```
> M1d <- lm(delta ~ gamma)           # linearity between residuals?
> coef(M1d)

   (Intercept)         gamma
  5.425926e-17 -4.611931e-01

> coef(M1)                            # coefficients of the full linear model

    (Intercept)         pop15         pop75           dpi          ddpi
  28.5660865407 -0.4611931471 -1.6914976767 -0.0003369019  0.4096949279

> abline(coef(M1d)["(Intercept)"], coef(M1d)["gamma"], col="red")
> abline(0,coef(M1)["pop15"], col="blue")
```

The added variable plot function `av.plots()` in the `car` package does a similar job. The reader might have inferred by now that `car` is the acronym of *Companion to Applied Regression*, the (2002) book of John Fox.

```
> av.plots(lm(sr ~ pop15 + pop75 + dpi + ddpi, data=savings))
```

```
> av.plots(M1)                       # for short
? av.plots
```

The help documentation of `av.plots()` shows options for variable selection and point identification.

▷ Notice that the slope in the residual plot and the slope for `pop15` in the full regression model are the same.

- Partial residual plot

This competitor of the added variable plot, plots $e_i + b_j X_{ij}$ against X_j. Again, the estimated slope will be b_j. Partial residual plots are better for the detection of linearity, added variable plots are better for the detection of outliers and influential data points.

```
> plot(savings$pop15, residuals(M1)+coef(M1)["pop15"]*savings$pop15,
+  xlab="pop15", ylab="Savings Adjusted")
> abline(0,coef(M1)["pop15"])
```

More directly, the partial residual plot function `prplot()` from the `faraway` package can be used, which provides the same result.

Notice that the source file `wilcox14.R` contains a (different) function with the label `prplot()`, which might cause interaction problems (error messages) at some point—check with command `fix(prplot)`.

```
> source("wilcox14.R")              # load source file 'wilcox14.R'
> prplot(M1, 1)                     # partial residual plot, where the second
                                    # argument indexes the explanatory variable
```

The function `cr.plots` `[component + residual (partial residual) plots]` in the `car` package could also be used.

```
> cr.plots(lm(sr ~ pop15 + pop75 + dpi + ddpi, data=savings),
+  variable="pop15")
> cr.plots(M1, variable="pop15")  # for short
```

It appears from these plots that there are different relationships in two groups: a group with a low percentage of the population under 15 years (`pop15`), and a group with a high percentage of `pop15`. A division could be made at `pop15 = 35`. We could, therefore, perform two separate analyses, one for each group. First we identify the groups, as follows:

```
> subset(savings, pop15 < 35)     # rich countries, it seems
> subset(savings, pop15 > 35)     # poor countries
```

```
> M1_low  <- lm(sr ~ pop15+pop75+dpi+ddpi, data=savings, subset=(pop15 < 35))
> M1_high <- lm(sr ~ pop15+pop75+dpi+ddpi, data=savings, subset=(pop15 > 35))
```

```
> summary(M1_low)

   Call:
   lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = savings,
     subset = (pop15 < 35))

   Residuals:
       Min      1Q  Median      3Q     Max
   -5.5890 -1.5015  0.1165  1.8857  5.1466

   Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
   (Intercept) 23.9617950  8.0837502   2.964  0.00716
   pop15       -0.3858976  0.1953686  -1.975  0.06092
   pop75       -1.3277421  0.9260627  -1.434  0.16570
   dpi         -0.0004588  0.0007237  -0.634  0.53264
   ddpi         0.8843944  0.2953405   2.994  0.00668

   Residual standard error: 2.772 on 22 degrees of freedom
   Multiple R-Squared: 0.5073,     Adjusted R-squared: 0.4177
   F-statistic: 5.663 on 4 and 22 DF,  p-value: 0.002734

> summary(M1_high)

   Call:
   lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = savings,
       subset = (pop15 > 35))

   Residuals:
        Min       1Q   Median       3Q      Max
   -5.55105 -3.51012  0.04428  2.67638  8.49830

   Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
   (Intercept) -2.4339689 21.1550278  -0.115    0.910
   pop15        0.2738537  0.4391910   0.624    0.541
   pop75       -3.5484769  3.0332806  -1.170    0.257
   dpi          0.0004208  0.0050001   0.084    0.934
   ddpi         0.3954742  0.2901012   1.363    0.190

   Residual standard error: 4.454 on 18 degrees of freedom
   Multiple R-Squared: 0.1558,     Adjusted R-squared: -0.03185
   F-statistic: 0.8302 on 4 and 18 DF,  p-value: 0.5233
```

▷ Try to interpret the results of these analyses, and draw appropriate conclusions. Notice, for example, the different estimates of the residual standard errors in the two groups, and the different R-squared values.

5. More diagnostics

The general suite of functions `influence.measures(model)` also contains the functions `dffits(model)`, `dfbeta(model)`, `dfbeta(model)` and `dfbetas(model)`, as described by Boomsma (2010).

In Section 4 we have not discussed regression diagnostics with respect to the problem of multicollinearity. It practice, this potential problem should not be left unattended, of course. Many of the regression diagnostics described above can also be used for generalized linear model fitting. The `stats` package contains the workhorse function `gls()`, by which we can work with non-normal error distributions — like the families of binomial, Poisson and gamma distributions — and link functions as well.

References

Belsley, D.A., Kuh, E., & Welsch, R.E. (1980). *Regression diagnostics.* New York: Wiley.

Boomsma, A. (2010). *An overview of regression diagnostics.* Unpublished manuscript, Department of Statistics & Measurement Theory, University of Groningen.

Chambers, J.M., Cleveland, W.S., Kleiner, B., & Tukey, P.A. (1983). *Graphical methods for data analysis.* Belmont, CA: Wadsworth.

Faraway, J.J. (2002). *Practical regression and anova using R.* Unpublished manuscript. Retrieved May 20, 2009, from `http://cran.r-project.org/doc/contrib/Faraway-PRA.pdf`. [Data sets and scripts are also directly available from `http://www.maths.bath.ac.uk/~jjf23/book/`.]

Faraway, J.J. (2005). *Linear models with R.* Boca Raton, FL: Chapman & Hall/CRC.

Fox, J. (2002). *An R and S-Plus companion to applied regression.* Thousand Oaks, CA: Sage.

Stevens, J. (1992). *Applied multivariate statistics for the social sciences* (2nd ed.). Hillsdale, NJ: Erlbaum.

Analysis of covariance with R