Chapter 6
Linear Models and Regression Diagnostics

The linear regression model estimated with ordinary least squares (OLS) is a workhorse model in Political Science. Even when a scholar uses a more advanced method that may make more accurate assumptions about his or her data—such as probit regression, a count model, or even a uniquely crafted Bayesian model—the researcher often draws from the basic form of a model that is linear in the parameters. By a similar token, many of the R commands for these more advanced techniques use functional syntax that resembles the code for estimating a linear regression. Therefore, an understanding of how to use R to estimate, interpret, and diagnose the properties of a linear model lends itself to sophisticated use of models with a similar structure.

This chapter proceeds by describing the `lm` (**l**inear **m**odel) command in R, which estimates a linear regression model with OLS, and the command's various options. Then, the chapter describes how to conduct regression diagnostics of a linear model. These diagnostics serve to evaluate whether critical assumptions of OLS estimation hold up, or if our results may be subject to bias or inefficiency.

Throughout the chapter, the working example is an analysis of the number of hours high school biology teachers spend teaching evolution. The model replicates work by Berkman and Plutzer (2010, Table 7.2), who argue that this policy outcome is affected by state-level factors (such as curriculum standards) and teacher attributes (such as training). The data are from the National Survey of High School Biology Teachers and consist of 854 observations of high school biology teachers who were surveyed in the spring of 2007. The outcome of interest is the number of hours a teacher devotes to human and general evolution in his or her high school biology class (**hrs_allev**), and the twelve input variables are as follows:

Electronic supplementary material: The online version of this chapter (doi: 10.1007/978-3-319-23446-5_6) contains supplementary material, which is available to authorized users.

J.E. Monogan III, *Political Analysis Using R*, Use R!,
DOI 10.1007/978-3-319-23446-5_6

phase1: An index of the rigor of ninth & tenth grade evolution standards in 2007 for the state the teacher works in. This variable is coded on a standardized scale with mean 0 and standard deviation 1.

senior_c: An ordinal variable for the seniority of the teacher. Coded −3 for 1–2 years experience, −2 for 3–5 years, −1 for 6–10 years, 0 for 11–20 years, and 1 for 21+ years.

ph_senior: An interaction between standards and seniority.

notest_p: An indicator variable coded 1 if the teacher reports that the state does not have an assessment test for high school biology, 0 if the state does have such a test.

ph_notest_p: An interaction between standards and no state test.

female: An indicator variable coded 1 if the teacher is female, 0 if male. Missing values are coded 9.

biocred3: An ordinal variable for how many biology credit hours the teacher has (both graduate and undergraduate). Coded 0 for 24 hours or less, 1 for 25–40 hours, and 2 for 40+ hours.

degr3: The number of science degrees the teacher holds, from 0 to 2.

evol_course: An indicator variable coded 1 if the instructor took a specific college-level course on evolution, 0 otherwise.

certified: An indicator coded 1 if the teacher has normal state certification, 0 otherwise.

idsci_trans: A composite measure, ranging from 0 to 1, of the degree to which the teacher thinks of him or herself as a scientist.

confident: Self-rated expertise on evolutionary theory. Coded −1 for "less" than many other teachers, 0 for "typical" of most teachers, 1 for "very good" compared to most high school biology teachers, and 2 for "exceptional" and on par with college-level instructors.

6.1 Estimation with Ordinary Least Squares

To start, we need to load the survey data, which we will name `evolution`. In this example, we load a Stata-formatted data set. This is easily possible through the `foreign` library, which provides us with the `read.dta` command:[1]

```
rm(list=ls())
library(foreign)
evolution<-read.dta("BPchap7.dta",convert.factors=FALSE)
```

[1] Berkman and Plutzer's data file, named `BPchap7.dta`, is available from the Dataverse linked on page vii or the chapter content linked on page 79. Remember that you may need to use the `setwd` command to point to where you have saved the data.

As a rule, we want to start by viewing the descriptive statistics from our data set. At minimum, use the `summary` command, and perhaps some of the other commands described in Chaps. 3 and 4:

```
summary(evolution)
```

In addition to the descriptive statistics `summary` gives us, it will also list the number of missing observations we have on a given variable (under `NA's`), if any are missing. The default condition for most modeling commands in R is to delete any case that is missing an observation on any variable in a model. Hence, the researcher needs to be aware not only of variation in relevant variables, but also how many cases lack an observation.[2] Additionally, researchers should be careful to notice anything in the descriptive statistics that deviates from a variable's values that are listed in the codebook. For example, in this case the variable **female** has a maximum value of 9. If we know from our codebook that 0 and 1 are the only valid observed values of this variable, then we know that anything else is either a miscode or (in this case) a missing value.

Before proceeding, we need to reclassify the missing observations of **female**:

```
evolution$female[evolution$female==9]<-NA
summary(evolution)
evolution<-subset(evolution,!is.na(female))
```

This command recodes only the values of **female** coded as a 9 as missing. As the subsequent call to `summary` shows, the 13 values coded as a 9 are now listed as missing, so they will automatically be omitted in our subsequent analysis. To make sure any computations we make focus only on the observations over which we fit the model, we subset our data to exclude the missing observations. As an alternative to using `subset` here, if we had missing values on multiple variables, we instead may have wanted to type: `evolution<-na.omit(evolution)`.

Having cleaned our data, we now turn to the model of hours spent teaching evolution described at the start of the chapter. We estimate our linear model using OLS:

```
mod.hours<-lm(hrs_allev~phase1*senior_c+phase1*notest_p+
     female+biocred3+degr3+evol_course+certified+idsci_trans+
     confident,data=evolution)
summary(mod.hours)
```

The standard syntax for specifying the formula for a model is to list the outcome variable to the left of the tilde (~), and the input variables on the right-hand side separated by plus signs. Notice that we did include two special terms: `phase1*senior_c` and `phase1*notest_p`. Considering the first, `phase1*senior_c`, this *interactive notation* adds three terms to our model: **phase1**, **senior_c**, and the product of the two. Such interactive models allow for

[2] A theoretically attractive alternative to *listwise deletion* as a means of handling missing data is *multiple imputation*. See Little and Rubin (1987), Rubin (1987), and King et al. (2001) for more details.

conditional effects of a variable.[3] The `data` option of `lm` allows us to call variables from the same dataset without having to refer to the dataset's name with each variable. Other prominent options for the `lm` command include `subset`, which allows the user to analyze only a portion of a dataset, and `weights`, which allows the user to estimate a linear model with weighted least squares (WLS). Observe that we had to name our model upon estimation, calling it `mod.hours` by choice, and to obtain the results of our estimation, we need to call our model with the `summary` command. The output of `summary(mod.hours)` looks like this:

```
Call:
lm(formula=hrs_allev~phase1*senior_c+phase1*notest_p+
    female+biocred3+degr3+evol_course+certified+idsci
      _trans+
    confident,data=evolution)

Residuals:
    Min      1Q  Median      3Q     Max
-20.378  -6.148  -1.314   4.744  32.148

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      10.2313     1.1905   8.594  < 2e-16 ***
phase1            0.6285     0.3331   1.886   0.0596 .
senior_c         -0.5813     0.3130  -1.857   0.0636 .
notest_p          0.4852     0.7222   0.672   0.5019
female           -1.3546     0.6016  -2.252   0.0246 *
biocred3          0.5559     0.5072   1.096   0.2734
degr3            -0.4003     0.3922  -1.021   0.3077
evol_course       2.5108     0.6300   3.985 7.33e-05 ***
certified        -0.4446     0.7212  -0.617   0.5377
idsci_trans       1.8549     1.1255   1.648   0.0997 .
confident         2.6262     0.4501   5.835 7.71e-09 ***
phase1:senior_c  -0.5112     0.2717  -1.881   0.0603 .
phase1:notest_p  -0.5362     0.6233  -0.860   0.3899
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 8.397 on 828 degrees of freedom
Multiple R-squared:  0.1226,Adjusted R-squared:  0.1099
F-statistic: 9.641 on 12 and 828 DF,  p-value: < 2.2e-16
```

[3] See Brambor et al. (2006) for further details on interaction terms. Also, note that an equivalent specification of this model could be achieved by replacing `phase1*senior_c` and `phase1*notest_p` with the terms `phase1+senior_c+ph_senior+notest_p+ph_notest_p`. We are simply introducing each of the terms separately in this way.

Table 6.1 Linear model of hours of class time spent teaching evolution by high school biology teachers (OLS estimates)

Predictor	Estimate	Std. Error	t value	Pr(> $\|t\|$)
Intercept	10.2313	1.1905	8.59	0.0000
Standards index 2007	0.6285	0.3331	1.89	0.0596
Seniority (centered)	−0.5813	0.3130	−1.86	0.0636
Standards × seniority	−0.5112	0.2717	−1.88	0.0603
Believes there is no test	0.4852	0.7222	0.67	0.5019
Standards × believes no test	−0.5362	0.6233	−0.86	0.3899
Teacher is female	−1.3546	0.6016	−2.25	0.0246
Credits earned in biology (0–2)	0.5559	0.5072	1.10	0.2734
Science degrees (0–2)	−0.4003	0.3922	−1.02	0.3077
Completed evolution class	2.5108	0.6300	3.99	0.0001
Has normal certification	−0.4446	0.7212	−0.62	0.5377
Identifies as scientist	1.8549	1.1255	1.65	0.0997
Self-rated expertise (−1 to +2)	2.6262	0.4501	5.84	0.0000

Notes: $N = 841$. $R^2 = 0.1226$. $F_{12,828} = 9.641$ ($p < 0.001$). Data from Berkman and Plutzer (2010)

The top of the printout repeats the user-specified model command and then provides some descriptive statistics for the residuals. The table that follows presents the results of primary interest: The first column lists every predictor in the model, including an intercept. The second column presents the OLS estimate of the partial regression coefficient. The third column presents the t-ratio for a null hypothesis that the partial regression coefficient is zero, and the fourth column presents a two-tailed p-value for the t-ratio. Finally, the table prints dots and stars based on the thresholds that the two-tailed p-value crosses.[4] Below the table, several fit statistics are reported: The standard error of regression (or residual standard error), the R^2 and adjusted R^2 values, and the F-test for whether the model as a whole explains a significant portion of variance. The results of this model also are presented more formally in Table 6.1.[5]

[4]Users are reminded that for one-tailed tests, in which the user wishes to test that the partial coefficient specifically is either greater than or less than zero, the p-value will differ. If the sign of the coefficient matches the alternative hypothesis, then the corresponding p-value is half of what is reported. (Naturally, if the sign of the coefficient is opposite the sign of the alternative hypothesis, the data do not fit with the researcher's hypothesis.) Additionally, researchers may want to test a hypothesis in which the null hypothesis is something other than zero: In this case, the user can construct the correct t-ratio using the reported estimate and standard error.

[5]Researchers who write their documents with LaTeX can easily transfer the results of a linear model from R to a table using the `xtable` library. (HTML is also supported by `xtable`.) On first use, install with: `install.packages("xtable")`. Once installed, simply entering `library(xtable); xtable(mod.hours)` would produce LaTeX-ready code for a table that is similar to Table 6.1. As another option for outputting results, see the `rtf` package about how to output results into Rich Text Format.

Many researchers, rather than reporting the *t*-ratios and *p*-values presented in the default output of `lm` will instead report *confidence intervals* of their estimates. One must be careful in the interpretation of confidence intervals, so readers unfamiliar with these are urged to consult a statistics or econometrics textbook for more information (such as Gujarati and Porter 2009, pp. 108–109). To construct such a **conf**idence **int**erval in R, the user must choose a confidence level and use the `confint` command:

```
confint(mod.hours,level=0.90)
```

The `level` option is where the user specifies the confidence level. `0.90` corresponds to 90 % confidence, while `level=0.99`, for instance, would produce a 99 % confidence interval. The results of our 90 % confidence interval are reported as follows:

```
                          5 %         95 %
(Intercept)       8.27092375 12.19176909
phase1            0.07987796  1.17702352
senior_c         -1.09665413 -0.06587642
notest_p         -0.70400967  1.67437410
female           -2.34534464 -0.36388231
biocred3         -0.27927088  1.39099719
degr3            -1.04614354  0.24552777
evol_course       1.47336072  3.54819493
certified        -1.63229086  0.74299337
idsci_trans       0.00154974  3.70834835
confident         1.88506881  3.36729476
phase1:senior_c -0.95856134 -0.06377716
phase1:notest_p -1.56260919  0.49020149
```

Among other features, one useful attribute of these is that a reader can examine a 90 % (for instance) confidence interval and reject any null hypothesis that proposes a value outside of the interval's range for a two-tailed test. For example, the interval for the variable **confident** does not include zero, so we can conclude with 90 % confidence that the partial coefficient for this variable is different from zero.[6]

6.2 Regression Diagnostics

We are only content to use OLS to estimate a linear model if it is the Best Linear Unbiased Estimator (BLUE). In other words, we want to obtain estimates that on average yield the true population parameter (unbiased), and among unbiased

[6]In fact, we also could conclude that the coefficient is *greater* than zero at the 95 % confidence level. For more on how confidence intervals can be useful for one-tailed tests as well, see Gujarati and Porter (2009, p. 115).

estimators we want the estimator that minimizes the error variance of our estimates (best or efficient). Under the Gauss–Markov theorem, OLS is BLUE and valid for inferences if four assumptions hold:

1. Fixed or exogenous input values. In other words the predictors (X) must be independent of the error term. $\text{Cov}(X_{2i}, u_i) = \text{Cov}(X_{3i}, u_i) = \cdots = \text{Cov}(X_{ki}, u_i) = 0$.
2. Correct functional form. In other words, the conditional mean of the disturbance must be zero.
   $E(u_i|X_{2i}, X_{3i}, \ldots, X_{ki}) = 0$.
3. Homoscedasticity or constant variance of the disturbances (u_i). $\text{Var}(u_i) = \sigma^2$.
4. There is no autocorrelation between disturbances. $\text{Cov}(u_i, u_j) = 0$ for $i \neq j$.

While we never observe the values of disturbances, as these are population terms, we can predict residuals ($\hat{u}$) after estimating a linear model. Hence, we typically will use residuals in order to assess whether we are willing to make the Gauss–Markov assumptions. In the following subsections, we conduct regression diagnostics to assess the various assumptions and describe how we might conduct remedial measures in R to correct for apparent violations of the Gauss–Markov assumptions. The one exception is that we do not test the assumption of *no autocorrelation* because we cannot reference our example data by time or space. See Chap. 9 for examples of autocorrelation tests and corrections. Additionally, we describe how to diagnose whether the errors have a normal distribution, which is essential for statistical inference. Finally, we consider the presence of two notable data features—multicollinearity and outlier observations—that are not part of the Gauss–Markov assumptions but nevertheless are worth checking for.

6.2.1 Functional Form

It is critical to have the correct functional form in a linear model; otherwise, its results will be *biased*. Therefore, upon estimating a linear model we need to assess whether we have specified the model correctly, or whether we need to include nonlinear aspects of our predictors (such as logarithms, square roots, squares, cubes, or splines). As a rule, an essential diagnostic for any linear model is to do a scatterplot of the residuals ($\hat{u}$). These plots ought to be done against both the fitted values ($\hat{Y}$) and against the predictors (X). To construct a plot of residuals against fitted values, we would simply reference attributes of the model we estimated in a call to the `plot` command:

```
plot(y=mod.hours$residuals,x=mod.hours$fitted.values,
     xlab="Fitted Values",ylab="Residuals")
```

Notice that `mod.hours$residuals` allowed us to reference the model's residuals ($\hat{u}$), and `mod.hours$fitted.values` allowed us to call the predicted values ($\hat{Y}$). We can reference many features with the dollar sign (`$`). Type `names(mod.hours)` to see everything that is saved. Turning to our output plot, it is presented in Fig. 6.1. As analysts, we should check this plot for a few features: Does the local average of the residuals tend to stay around zero? If the

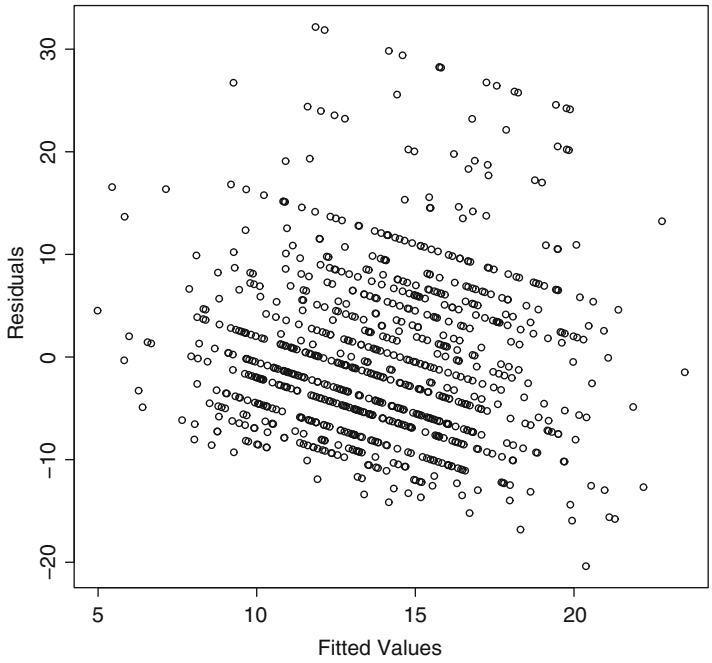

Fig. 6.1 Scatterplot of residuals against fitted values from model of hours of teaching evolution

residuals show a clear pattern of rising or falling over any range, then the functional form of some variable may be wrong. Does the spread of the residuals differ at any portion in the graph? If so, there may be a heteroscedasticity issue. One apparent feature of Fig. 6.1 is that the residuals appear to hit a diagonal "floor" near the bottom of the cloud. This emerges because a teacher cannot spend fewer than zero hours teaching evolution. Hence, this natural floor reflects a limit in the dependent variable. A functional form limitation such as this is often best addressed within the Generalized Linear Model framework, which will be considered in the next chapter.

Another useful tool is to draw figures of the residuals against one or more predictors. Figure 6.2 shows two plots of the residuals from our model against the composite scale of the degree to which the teacher self-identifies as a scientist. Figure 6.2a shows the basic plot using the raw data, which a researcher should always look at. In this case, the predictor of interest takes on 82 unique values, but many observations take on the same values, particularly at the upper end of the scale. In cases like this, many points on the plot will be superimposed on each other. By *jittering* the values of **idsci_trans**, or adding a small randomly drawn number, it becomes easier to see where a preponderance of the data are. Figure 6.2b shows a revised plot that jitters the predictor. The risk of the jittered figure is that moving the data can distort a true pattern between the predictor and residuals. However, in a case of an ordinal (or perhaps semi-ordinal) input variable, the two subfigures

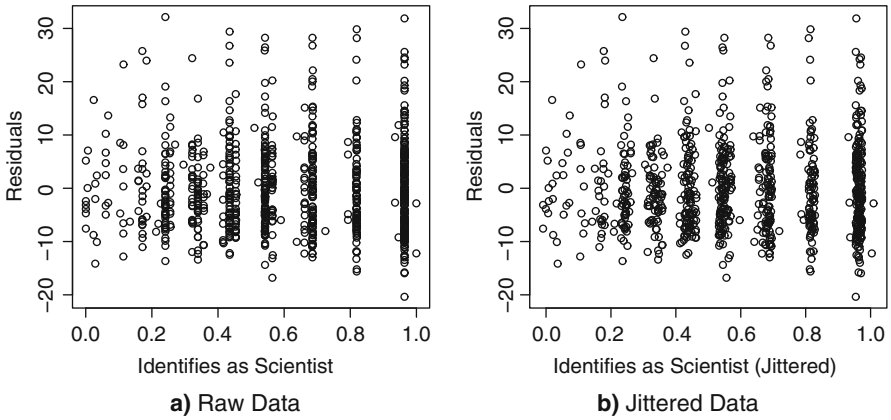

Fig. 6.2 Scatterplot of residuals against the degree to which a teacher identifies as a scientist. (**a**) Raw data. (**b**) Jittered data

can complement each other to offer the fullest possible picture. The two scatterplots from Fig. 6.2 are produced as follows:

```
plot(y=mod.hours$residuals,x=evolution$idsci_trans,
     xlab="Identifies as Scientist",ylab="Residuals")
plot(y=mod.hours$residuals,x=jitter(evolution$idsci_trans,
     amount=.01),xlab="Identifies as Scientist (Jittered)",
     ylab="Residuals")
```

Much like the residual-to-fitted value plot of Fig. 6.1, we examine the residual-to-predictor plots of Fig. 6.2 for changes in the local mean as well as differences in the spread of the residuals, each contingent on the predictor value. On functional form, there is little to suggest that the running mean is changing markedly across values. Hence, as with the residual-to-fitted plot, we see little need to respecify our model with a nonlinear version of this predictor. However, the spread of the residuals looks a bit concerning, so we will revisit this issue in the next section.

In addition to graphical methods, one common test statistic for diagnosing a misspecified functional form is Ramsey's RESET test (regression specification error test). This test proceeds by reestimating the original model, but this time including the fitted values from the original model in some nonlinear form (such as a quadratic or cubic formula). Using an F-ratio to assess whether the new model explains significantly more variance than the old model serves as a test of whether a different form of one or more predictors should be included in the model. We can conduct this test for a potential cubic functional form as follows:

```
evolution$fit<-mod.hours$fitted.values
reset.mod<-lm(hrs_allev~phase1*senior_c+phase1*notest_p+
     female+biocred3+degr3+evol_course+certified+idsci_trans+
     confident+I(fit^2)+I(fit^3), data=evolution)
anova(mod.hours, reset.mod)
```

The first line of code saves the fitted values from the original model as a variable in the data frame. The second line adds squared and cubed forms of the fitted values into the regression model. By embedding these terms within the `I` function (again meaning, "as **i**s"), we can algebraically transform the input variable on the fly as we estimate the model. Third, the `anova` command (for **an**alysis **o**f **va**riance) presents the results of an F-test that compares the original model to the model including a quadratic and cubic form of the fitted values. In this case, we get a result of $F_{2826} = 2.5626$, with a p-value of $p = 0.07772$. This indicates that the model with the cubic polynomial of fitted values does fit significantly better at the 90 % level, implying another functional form would be better.

To determine which predictor could be the culprit of the misspecified functional form, we can conduct Durbin–Watson tests on the residuals, sorting on the predictor that may be problematic. (*Note* that traditionally Durbin–Watson tests sort on *time* to test for temporal autocorrelation. This idea is revisited in Chap. 9.) A discernible result indicates that residuals take similar values at similar values of the input—a sign that the predictor needs to be respecified. The `lmtest` library (users will need to install with `install.packages` the first time) provides commands for several diagnostic tests, including the Durbin–Watson test. Sorting the residuals on the rigor of evolution standards (**phase1**), we run the test:

```
install.packages("lmtest")
library(lmtest)
dwtest(mod.hours, order.by=evolution$phase1)
```

This yields a statistic of $d = 1.8519$ with an approximate p-value of $p = 0.01368$, indicating that the residuals are similar based on the value of the covariate. Therefore, we might proceed to respecify our functional form by adding polynomial terms for **phase1**:

```
mod.cubic<-lm(hrs_allev~phase1*senior_c+phase1*notest_p+
     female+biocred3+degr3+evol_course+certified+idsci_trans+
     confident+I(phase1^2)*senior_c+I(phase1^3)*senior_c+
     I(phase1^2)*notest_p+I(phase1^3)*notest_p,data=evolution)
```

As with the RESET test itself, our new model (`mod.cubic`) illustrates how we can use additional features of the `lm` command. Again, by using the `I` function, we can perform algebra on any input variable within the model command. As before, the caret (`^`) raises a variable to a power, allowing our polynomial function. Again, for interaction terms, simply multiplying two variables with an asterisk (`*`) ensures that the main effects and product terms of all variables in the interaction are included. Hence, we allow seniority and whether there is no assessment test each to interact with the full polynomial form of evolution standards.

6.2.2 Heteroscedasticity

When the error variance in the residuals is not uniform across all observations, a model has heteroscedastic error variance, the estimates are inefficient, and the standard errors are biased to be too small. The first tool we use to assess whether the error variance is homoscedastic (or constant for all observations) versus heteroscedastic is a simple scatterplot of the residuals. Figure 6.1 offered us the plot of our residuals against the fitted values, and Fig. 6.2 offers an example plot of the residuals against a predictor. Besides studying the running mean to evaluate functional form, we also assess the spread of residuals. If the dispersion of the residuals is a constant band around zero, then we may use this as a visual confirmation of homoscedasticity. However, in the two panels of Fig. 6.2, we can see that the preponderance of residuals is more narrowly concentrated close to zero for teachers who are less inclined to self-identify as a scientist, while the residuals are more spread-out among those who are more inclined to identify as a scientist. (The extreme residuals are about the same for all values of X, making this somewhat tougher to spot, but the spread of concentrated data points in the middle expands at higher values.) All of this suggests that self-identification as a scientist corresponds with heteroscedasticity for this model.

Besides visual methods, we also have the option of using a test statistic in a Breusch–Pagan test. Using the `lmtest` library (which we loaded earlier), the syntax is as follows:

```
bptest(mod.hours, studentize=FALSE)
```

The default of `bptest` is to use Koenker's studentized version of this test. Hence, the `studentize=FALSE` option gives the user the choice of using the original version of the Breusch–Pagan test. The null hypothesis in this chi-squared test is homoscedasticity. In this case, our test statistic is $\chi^2_{12\text{df}} = 51.7389$ ($p < 0.0001$). Hence, we reject the null hypothesis and conclude that the residuals are not homoscedastic.

Without homoscedasticity, our results are not efficient, so how might we correct for this? Perhaps the most common solution to this issue is to use Huber–White robust standard errors, or sandwich standard errors (Huber 1967; White 1980). The downside of this method is that it ignores the inefficiency of the OLS estimates and continues to report these as the parameter estimates. The upside, however, is that although OLS estimates are inefficient under heteroscedasticity, they are unbiased. Since the standard errors are biased, correcting them fixes the biggest problem heteroscedasticity presents us. Computing Huber–White standard errors can be accomplished using the `sandwich` (needing a first-time install) and `lmtest` libraries:

```
install.packages("sandwich")
library(sandwich)
coeftest(mod.hours,vcov=vcovHC)
```

The `lmtest` library makes the `coeftest` command available, and the `sandwich` library makes the variance-covariance matrix `vcovHC` available within this. (Both libraries require installation on first use.) The `coeftest` command will now present the results of `mod.hours` again, with the same OLS estimates as before, the new Huber–White standard errors, and values of t and p that correspond to the new standard errors.

Finally, we also have the option to reestimate our model using WLS. To do this, the analyst must construct a model of the squared residuals as a way of forecasting the heteroscedastic error variance for each observation. While there are a few ways to do this effectively, here is the code for one plan. First, we save the squared residuals and fit an auxiliary model of the logarithm of these squared residuals:

```
evolution$resid2<-mod.hours$residuals^2
weight.reg<-lm(log(resid2)~phase1*senior_c+phase1*notest_p+
     female+biocred3+degr3+evol_course+certified+idsci_trans+
     confident, data=evolution)
```

A key caveat of WLS is that all weights must be nonnegative. To guarantee this, the code here models the logarithm of the squared residuals; therefore, the exponential of the fitted values from this auxiliary regression serve as positive predictions of the squared residuals. (Other solutions to this issue exist as well.) The auxiliary regression simply includes all of the predictors from the original regression in their linear form, but the user is not tied to this assumption. In fact, WLS offers the BLUE under heteroscedasticity, but only if the researcher properly models the error variance. Hence, proper specification of the auxiliary regression is essential. In WLS, we essentially want to highly weight values with a low error variance and give little weight to those with a high error variance. Hence, for our final WLS regression, the `weights` command takes the reciprocal of the predicted values (exponentiated to be on the original scale of the squared residuals):

```
wls.mod<-lm(hrs_allev~phase1*senior_c+phase1*notest_p+
     female+biocred3+degr3+evol_course+certified+idsci_trans+
     confident,data=evolution,
     weights=I(1/exp(weight.reg$fitted.values)))
summary(wls.mod)
```

This presents us with a set of estimates that accounts for heteroscedasticity in the residuals.

6.2.3 Normality

While not part of the Gauss–Markov theorem, an important assumption that we make with linear regression models is that the disturbances are normally distributed. If this assumption is not true, then OLS actually is still BLUE. The normality assumption is, however, essential for our usual inferential statistics to be accurate. Hence, we test this assumption by examining the empirical distribution of the

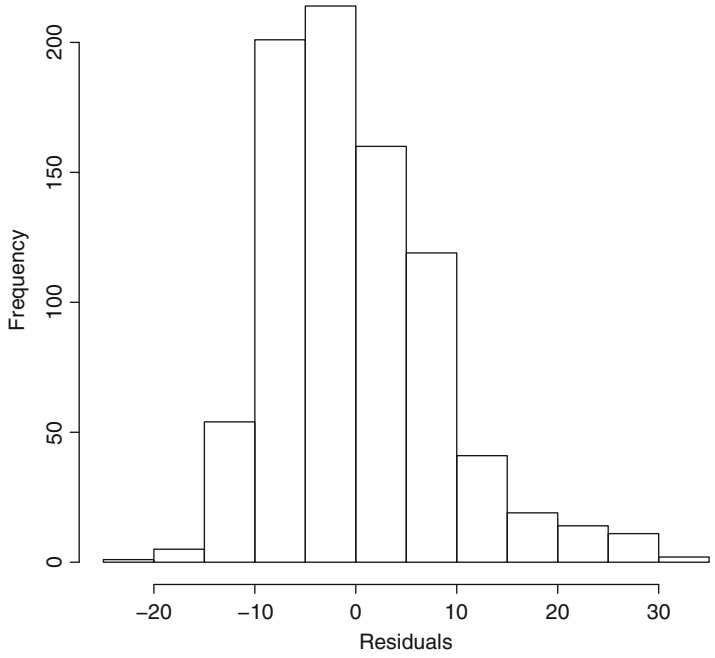

Fig. 6.3 Histogram of residuals from model of hours teaching evolution

predicted residuals. An easy first place to start is to examine a histogram of the residuals.

```
hist(mod.hours$residuals,xlab="Residuals",main="")
```

This histogram is reported in Fig. 6.3. Generally, we would like a symmetric bell curve that is neither excessively flat nor peaked. If both *skew* (referring to whether the distribution is symmetric or if the tails are even) and *kurtosis* (referring to the distribution's peakedness) are similar to a normal distribution, we may use this figure in favor of our assumption. In this case, the residuals appear to be right-skewed, suggesting that normality is not a safe assumption in this case.

A slightly more complex figure (albeit potentially more informative) is called a quantile–quantile plot. In this figure, the quantiles of the empirical values of the residuals are plotted against the quantiles of a theoretical normal distribution. The less these quantities correspond, the less reasonable it is to assume the residuals are distributed normally. Such a figure is constructed in R as follows:

```
qqnorm(mod.hours$residuals)
qqline(mod.hours$residuals,col="red")
```

The first line of code (`qqnorm`) actually creates the quantile–quantile plot. The second line (`qqline`) adds a guide line to the existing plot. The complete graph is located in Fig. 6.4. As can be seen, at lower and higher quantiles, the sample values

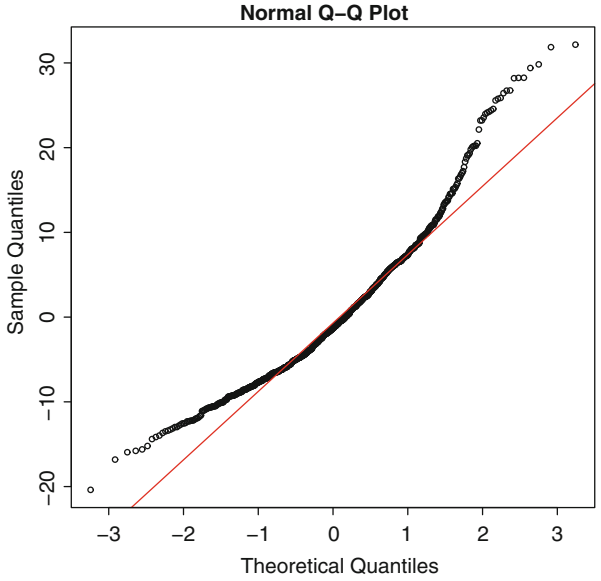

Fig. 6.4 Normal quantile–quantile plot for residuals from model of hours of teaching evolution

deviate substantially from the theoretical values. Again, the assumption of normality is questioned by this figure.

Besides these substantively focused assessments of the empirical distribution, researchers also can use test statistics. The most commonly used test statistic in this case is the Jarque–Bera test, which is based on the skew and kurtosis of the residuals' empirical distribution. This test uses the null hypothesis that the residuals are normally distributed and the alternative hypothesis that they are not.[7] The `tseries` library can calculate this statistic, which we install on first use:

```
install.packages("tseries")
library(tseries)
jarque.bera.test(mod.hours$residuals)
```

In our case, $\chi^2 = 191.5709$, so we reject the null hypothesis and conclude that the residuals are not normally distributed. Like diagnostics for heteroscedasticity, we would prefer a null result since we prefer not to reject the assumption.

All three diagnostics indicate a violation of the normality assumption, so how might we respond to this violation? In many cases, the best answer probably lies in the next chapter on Generalized Linear Models (GLMs). Under this framework, we can assume a wider range of distributions for the outcome variable, and we also can transform the outcome variable through a link function. Another somewhat

[7] In other words, if we fail to reject the null hypothesis for a Jarque–Bera test, then we conclude that there is not significant evidence of non-normality. Note that this is different from concluding that we do have normality. However, this is the strongest conclusion we can draw with this test statistic.

similar option would be to transform the dependent variable somehow. In the case of our running example on hours spent on evolution, our outcome variable cannot be negative, so we might add 1 to each teacher's response and take the logarithm of our dependent variable. Bear in mind, though, that this has a bigger impact on the model's *functional form* (see Gujarati and Porter 2009, pp. 162–164), and we have to assume that the disturbances of the model with a logged dependent variable are normally distributed for inferential purposes.

6.2.4 Multicollinearity

Although it is not a statistical assumption of the linear model, we now turn to diagnosing the presence of multicollinearity among predictors. Multicollinearity means that a predictor is a function of one or more other predictors. If a predictor is an exact function of other predictors, then there is perfect multicollinearity in the set of regressors. Under perfect multicollinearity, the model cannot be estimated as is and must be respecified. For example, if a researcher included both "year of birth" and "age" of a survey respondent in a cross-sectional analysis, one variable would be a perfect function of the other and therefore the model would not be estimable.

A common situation is for a predictor to have high, but not perfect, multicollinearity. The issue that emerges is that standard errors for regression coefficients will start to become large. Importantly, though, OLS is still BLUE in the case of high but imperfect multicollinearity. In other words, the large standard errors are accurate and still reflective of the most efficient estimator that is possible. Nevertheless, it is often a good idea to get a sense of whether multicollinearity is present in a regression model.

The general approach for assessing multicollinearity relies on auxiliary regressions of predictors. Among the summary measures of these results is the variance inflation factor (VIF). For each predictor, the VIF gives us some idea of the degree to which common variance among predictors increases the standard error of the predictor's coefficient. VIFs can take any non-negative value, and smaller values are more desirable. A common rule of thumb is that whenever a VIF exceeds 10, it can be concluded that multicollinearity is shaping the results.[8]

In R, VIFs can be computed for all coefficients using the `car` library, installed in Chap. 2:

```
library(car)
vif(mod.hours)
```

[8] A VIF of 10 means that 90 % of the variance in a predictor can be explained by the other predictors, which in most contexts can be regarded as a large degree of common variance. Unlike other diagnostic tests, though, this rule of thumb should not be regarded as a test statistic. Ultimately the researcher must draw a substantive conclusion from the results.

Table 6.2 Variance inflation factors for predictors of hours spent teaching evolution

Predictor	VIF
Standards index 2007	1.53
Seniority (centered)	1.12
Standards × seniority	1.10
Believes there is no test	1.12
Standards × believes no test	1.63
Teacher is female	1.08
Credits earned in biology (0–2)	1.15
Science degrees (0–2)	1.11
Completed evolution class	1.17
Has normal certification	1.03
Identifies as scientist	1.12
Self-rated expertise (−1 to +2)	1.20

The VIFs calculated in this way are presented in Table 6.2. As can be seen in the table, all of the VIFs are small, implying that multicollinearity is not a major issue in this model. In situations where multicollinearity does emerge, though, sometimes the best advice is to do nothing. For a discussion of how to decide whether doing nothing is the best approach or another solution would work better, see Gujarati and Porter (2009, pp. 342–346).

6.2.5 Outliers, Leverage, and Influential Data Points

As a final diagnostic, it is a good idea to determine whether any observations are exerting excessive influence on the results of a linear model. If one or two observations drive an entire result that otherwise would not emerge, then a model including these observations may be misleading. We consider three types of problematic data points: outliers (for which the residual is exceedingly large), leverage points (which take a value of a predictor that is disproportionately distant from other values), and influence points (outliers with a lot of leverage). The most problematic of these are influence points because they have the greatest ability to distort partial regression coefficients.

A simple diagnostic for these features of observations again is to simply examine scatterplots of residuals, such as those reported in Figs. 6.1 and 6.2. If an observation stands out on the predictor's scale then it has leverage. If it stands out on the residual scale then it is an outlier. If it stands out on both dimensions, then it is an influence point. Neither of the figures for this model show any warning signs in this respect. Another option for assessing these attributes for observations is to calculate the quantities of Studentized residuals to detect outliers, hat values to detect leverage points, and Cook's distances to detect influential data points. The `car` library again offers a simple way to view these quantities for all observations.

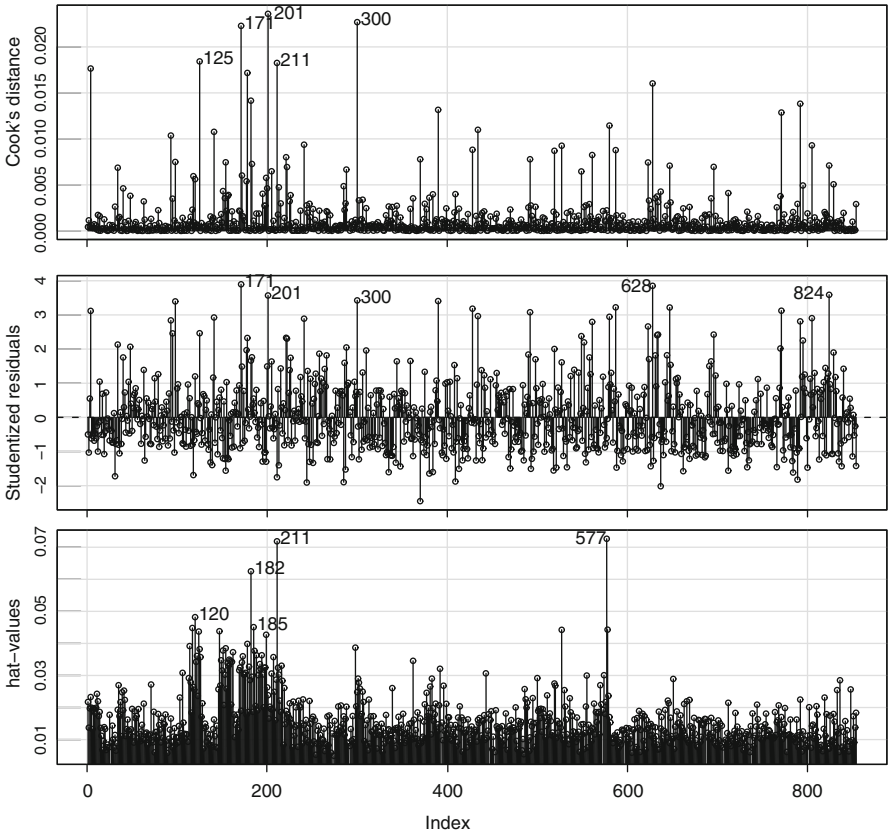

Fig. 6.5 Cook's distances, Studentized residuals, and hat values from model of hours teaching evolution

```
influenceIndexPlot(mod.hours,
     vars=c("Cook","Studentized","hat"),id.n=5)
```

The values of these three quantities are reported in Fig. 6.5, which shows Cook's distances, Studentized residuals, and hat values, respectively. In any of these plots, an extreme value relative to the others indicates that an observation may be particularly problematic. In this figure, none of the observations stand out particularly, and none of the values of Cook's distance are remotely close to 1 (which is a common rule-of-thumb threshold for this quantity). Hence, none of the observations appear to be particularly problematic for this model. In an instance where some observations do appear to exert influence on the results, the researcher must decide whether it is reasonable to keep the observations in the analysis or if any of them ought to be removed. Removing data from a linear model can easily be accomplished with the `subset` option of `lm`.

We now have considered how to fit linear models in R and how to conduct several diagnostics to determine whether OLS presents us with the BLUE. While this is a common model in Political Science, researchers frequently need to model limited dependent variables in the study of politics. To address dependent variables of this nature, we turn in the next chapter to GLMs. These models build on the linear model framework but allow outcome variables that are bounded or categorical in nature.

6.3 Practice Problems

This set of practice problems will draw from Owsiak's (2013) work on democratization, in which he shows that states that settle all of their international borders tend to become more democratic. Please load the foreign library and then download a subset of Owsiak's data, saved in the Stata-formatted file `owsiakJOP2013.dta`. This file can be downloaded from the Dataverse linked on page vii or the chapter content linked on page 79. These are *panel data* that include observations for 200 countries from 1918 to 2007, with a total of 10,434 country-years forming the data. The countries in these data change over time (just as they changed in your history book) making this what we call an unbalanced panel. Hence, our subsequent model includes lagged values of several variables, or values from the previous year. See Chap. 8 for more about nested data, and Chap. 9 for more about temporal data. For this exercise, our standard OLS tools will work well.

1. Start by using the `na.omit` command, described on page 81, to eliminate missing observations from these data. Then compute the descriptive statistics for the variables in this data set.
2. To replicate Model 2 from Owsiak (2013), estimate a linear regression with OLS using the following specification (with variable names in parentheses): The dependent variable is Polity score (**polity2**), and the predictors are an indicator for having all borders settled (**allsettle**), lagged GDP (**laggdpam**), lagged change in GDP (**laggdpchg**), lagged trade openness (**lagtradeopen**), lagged military personnel (**lagmilper**), lagged urban population (**lagupop**), lagged pervious non-democratic movement (**lagsumdown**), and lagged Polity score (**lagpolity**).
3. Plot the residuals against the fitted values.
4. Is there heteroscedasticity in the residuals? Based on scatterplots and a Breusch–Pagan test, what do you conclude?
    a. Estimate Huber–White standard errors for this model with the `sandwich` library and `coeftest` command.
    b. For bonus credit, you can reproduce Owsiak's (2013) results exactly by computing *clustered standard errors*, clustering on country (variable name: **ccode**). You can do this in three steps: First, install the `multiwayvcov` library. Second, define an error variance-covariance matrix using the `cluster.vcov` command. Third, use that error variance-covariance matrix as an argument in the `coeftest` command from the `lmtest` library.

5. Determine whether multicollinearity is a concern by calculating the VIFs for the predictors in this model.
6. Are the residuals of this model normally distributed? Use any of the discussed methods to draw a conclusion.
7. For bonus credit, you can evaluate whether there is autocorrelation in the residuals, as will be discussed further in Chap. 9. To to this, first install the `plm` library. Second, refit your model using the `plm` command. (Be sure to specify `model="pooling"` as an option in the command to estimate with OLS.) Third, use the `pbgtest` to conduct a panel Breusch–Godfrey test to evaluate whether there is serial correlation in the residuals. What conclusion do you draw?