15.2: OLS Diagnostic Techniques

In this section, we examine the residuals from a multiple regression model for potential problems. Note that we use a subsample of the first 500 observations, drawn from the larger tbur.data" dataset, to permit easier evaluation of the plots of residuals. We begin with an evaluation of the assumption of the linearity of the relationship between the XXs and YY, and then evaluate assumptions regarding the error term.

Our multiple regression model predicts survey respondents' levels of risk perceived of climate change (YY) using political ideology, age, household income, and educational achievement as independent variables (XXs). The results of the regression model as follows:

```
ols1 <-  lm(glbcc_risk ~ age + education + income + ideol, data = ds.small)
summary(ols1)
```

```
##
## Call:
## lm(formula = glbcc_risk ~ age + education + income + ideol, data = ds.small)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1617 -1.7131 -0.0584  1.7216  6.8981
##
## Coefficients:
##                   Estimate    Std. Error t value            Pr(>|t|)
## (Intercept) 12.0848259959  0.7246993630  16.676 <0.0000000000000002 ***
## age         -0.0055585796  0.0084072695  -0.661               0.509
## education   -0.0186146680  0.0697901408  -0.267               0.790
## income       0.0000001923  0.0000022269   0.086               0.931
## ideol       -1.2235648372  0.0663035792 -18.454 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.353 on 445 degrees of freedom
## Multiple R-squared:  0.4365, Adjusted R-squared:  0.4315
## F-statistic: 86.19 on 4 and 445 DF,  p-value: < 0.00000000000000022
```

On the basis of the RR output, the model appears to be quite reasonable, with a statistically significant estimated partial regression coefficient for political ideology. But let's take a closer look.

15.2.1 Non-Linearity

One of the most critical assumptions of OLS is that the relationships between variables are linear in their functional form. We start with a stylized example (a fancy way of saying we made it up!) of what a linear and nonlinear pattern of residuals would look like. Figure 15.2.2 shows an illustration of how the residuals would look with a clearly linear relationship, and Figure 15.2.3 illustrates how the the residuals would look with a clearly non-linear relationship.

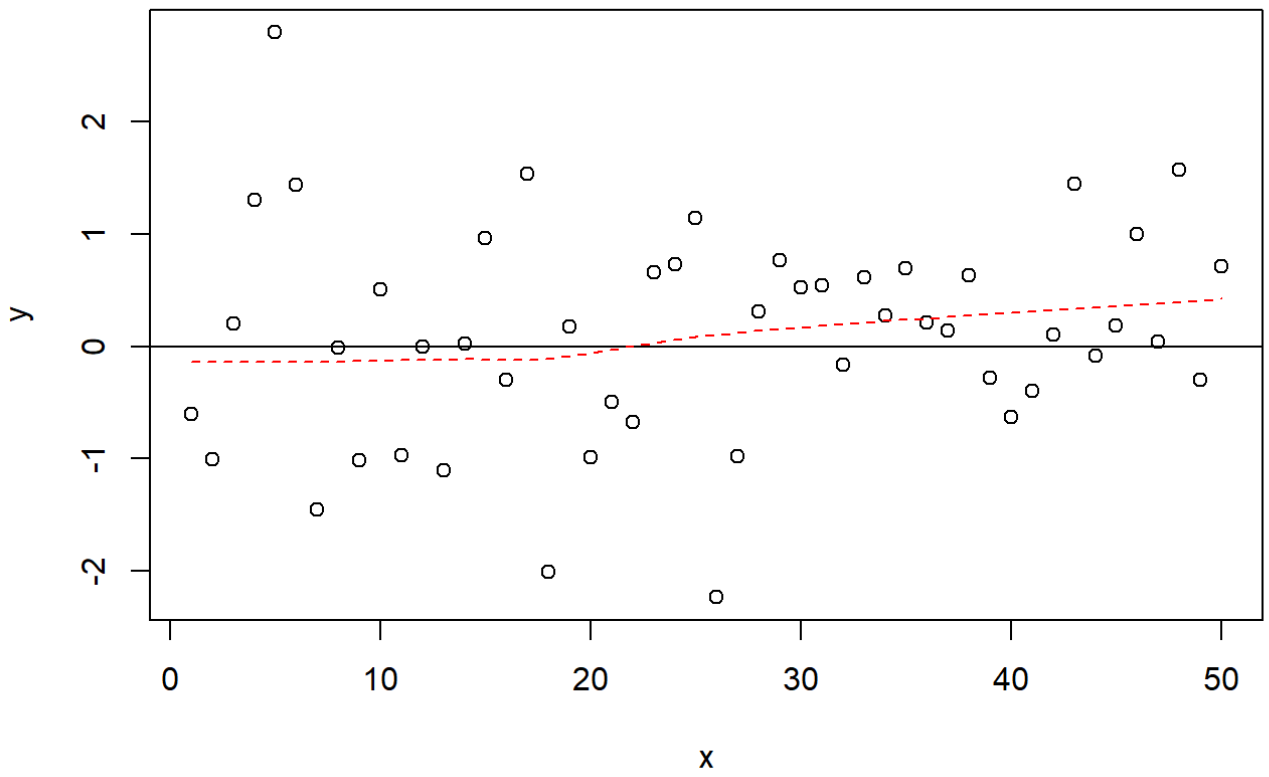

Figure 15.2.2 Linear

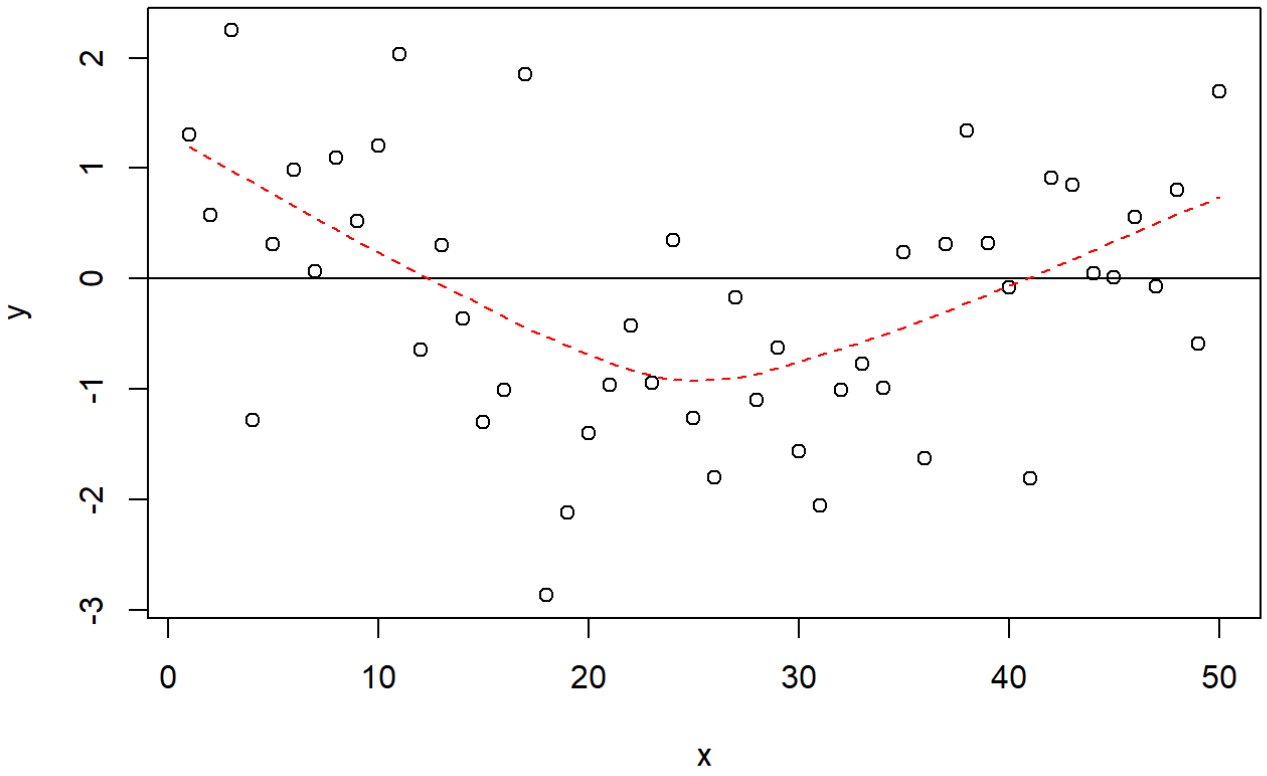

Figure 15.2.3: Non-Linear

Now let's look at the residuals from our example model. We can check the linear nature of the relationship between the DV and the IVs in several ways. First we can plot the residuals by the values of the IVs. We also can add a lowess line to demonstrate the relationship between each of the IVs and the residuals, and add a line at 00 for comparison.

```
ds.small$fit.r <- ols1$residuals
ds.small$fit.p <- ols1$fitted.values
```

```
library(reshape2)
ds.small %>%
  melt(measure.vars = c("age", "education", "income", "ideol", "fit.p")) %>%
  ggplot(aes(value, fit.r, group = variable)) +
  geom_point(shape = 1) +
  geom_smooth(method = loess) +
  geom_hline(yintercept = 0) +
  facet_wrap(~ variable, scales = "free")
```

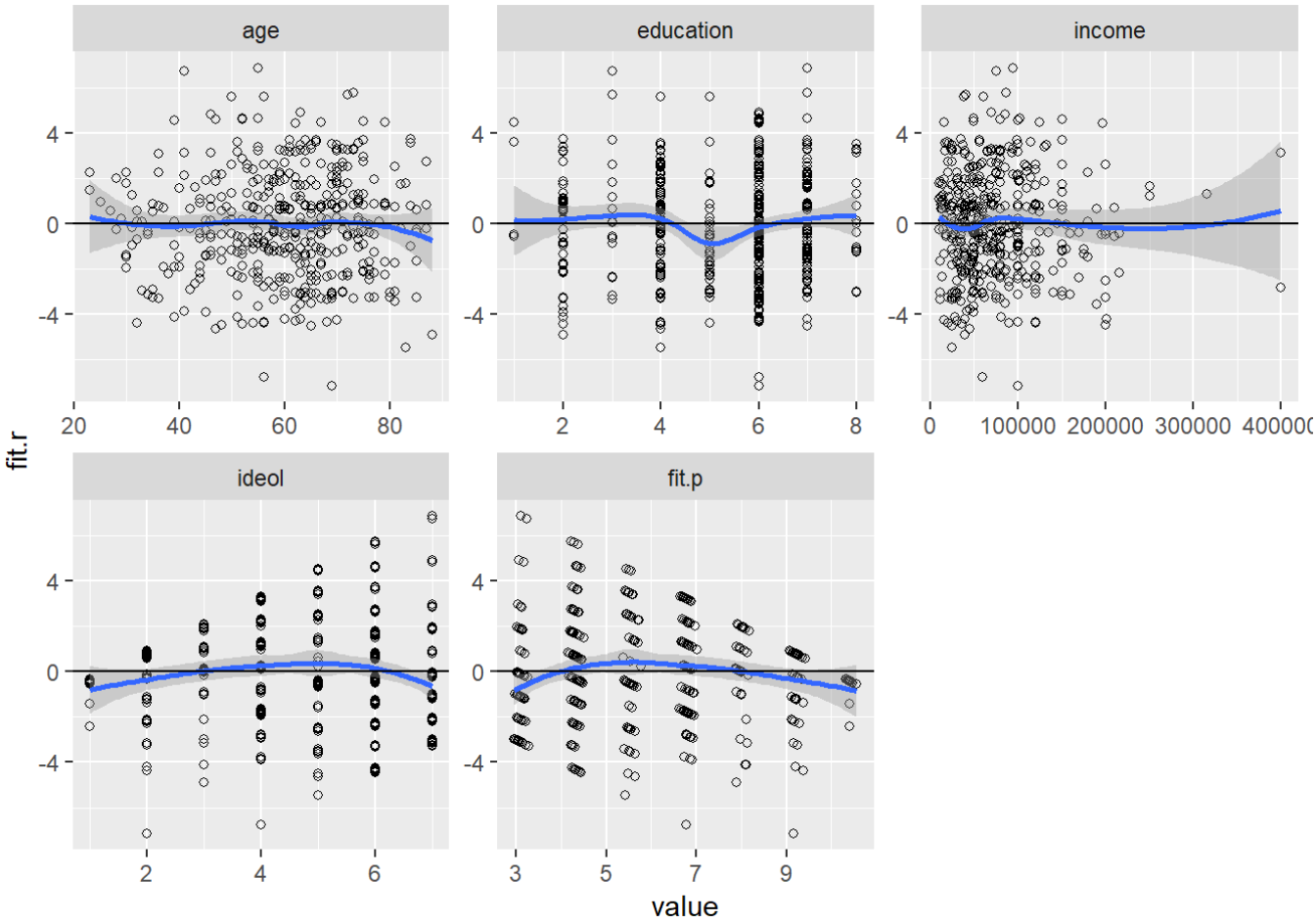

Figure 15.2.4: Checking for Non-Linearity

As we can see in Figure 15.2.4, the plots of residuals by both income and ideology seem to indicate non-linear relationships. We can check this "ocular impression" by squaring each term and using the `anova` function to compare model fit.

```
ds.small$age2 <- ds.small$age^2
ds.small$edu2 <- ds.small$education^2
ds.small$inc2 <- ds.small$income^2
ds.small$ideology2<-ds.small$ideol^2
ols2 <- lm(glbcc_risk ~ age+age2+education+edu2+income+inc2+ideol+ideology2, data=ds
summary(ols2)
```

```
##
## Call:
## lm(formula = glbcc_risk ~ age + age2 + education + edu2 + income +
##     inc2 + ideol + ideology2, data = ds.small)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1563 -1.5894  0.0389  1.4898  7.3417
##
## Coefficients:
##                      Estimate        Std. Error t value    Pr(>|t|)
## (Intercept)  9.66069872535646  1.93057305147186   5.004 0.000000812 ***
## age          0.02973349791714  0.05734762412523   0.518    0.604385
## age2        -0.00028910659305  0.00050097599702  -0.577    0.564175
## education   -0.48137978481400  0.35887879735475  -1.341    0.180499
## edu2         0.05131569933892  0.03722361864679   1.379    0.168723
## income       0.00000285263412  0.00000534134363   0.534    0.593564
## inc2        -0.00000000001131  0.00000000001839  -0.615    0.538966
## ideol       -0.05726196851107  0.35319018414228  -0.162    0.871279
## ideology2   -0.13270718319750  0.03964680646295  -3.347    0.000886 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.33 on 441 degrees of freedom
## Multiple R-squared:  0.4528, Adjusted R-squared:  0.4429
## F-statistic: 45.61 on 8 and 441 DF,  p-value: < 0.00000000000000022
```

The model output indicates that ideology may have a non-linear relationships with risk perceptions of climate change. For ideology, only the squared term is significant, indicating that levels of perceived risk of climate change decline at an increasing rate for those on the most conservative end of the scale. Again, this is consistent with the visual inspection of the relationship between ideology and the residuals in Figure 15.2.4 The question remains whether the introduction of these non-linear (polynomial) terms improves overall model fit. We can check that with an analysis of variance across the simple model (without polynomial terms) and the models with the squared terms.

```
anova(ols1,ols2)
```

```
## Analysis of Variance Table
##
## Model 1: glbcc_risk ~ age + education + income + ideol
## Model 2: glbcc_risk ~ age + age2 + education + edu2 + income + inc2 +
##     ideol + ideology2
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    445 2464.2
## 2    441 2393.2  4    71.059 3.2736 0.01161 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As we can see, the Anova test indicates that including the squared terms improves model fit, therefore the relationships include nonlinear components.

A final way to check for non-linearity is Ramsey's Regression Error Specification Test (RESET). This tests the functional form of the model. Similar to our test using squared terms, the RESET tests calculates an FF statistic that compares the linear model with a

model(s) that raises the IVs to various powers. Specifically, it tests whether there are statistically significant differences in the R2R2 of each of the models. Similar to a nested FF test, it is calculated by:

F=R21−R20q1−R21n−k1(15.1)(15.1)F=R12−R02q1−R12n−k1

where R20R02 is the R2R2 of the linear model, R21R12 is the R2R2 of the polynomial model(s), qq is the number of new regressors, and k1k1 is the number of IVs in the polynomial model(s). The null hypothesis is that the functional relationship between the XX's and YY is linear, therefore the coefficients of the second and third powers to the IVs are zero. If there is a low pp-value (i.e., if we can reject the null hypothesis), non-linear relationships are suspected. This test can be run using the `resettest` function from the `lmtest` package. Here we are setting the IVs to the second and third powers and we are examining the regressor variables.[24]

```
library(lmtest)
resettest(ols1,power=2:3,type="regressor")
```

```
##
##  RESET test
##
## data:  ols1
## RESET = 2.2752, df1 = 8, df2 = 437, p-value = 0.02157
```

Again, the test provides evidence that we have a non-linear relationship.

What should we do when we identify a nonlinear relationship between our YY and XXs? The first step is to look closely at the bi-variate plots, to try to discern the correct functional form for each XX regressor. If the relationship looks curvilinear, try a polynomial regression in which you include both XX and X2X2 for the relevant IVs. It may also be the case that a skewed DV or IV is causing the problem. This is not unusual when, for example, the income variable plays an important role in the model, and the distribution of income is skewed upward. In such a case, you can try transforming the skewed variable, using an appropriate log form.

It is possible that variable transformations won't suffice, however. In that case, you may have no other option by to try non-linear forms of regression. These non-OLS kinds of models typically use maximal likelihood functions (see the next chapter) to fit the model to the data. But that takes us considerably beyond the focus of this book.

15.2.2 Non-Constant Variance, or Heteroscedasticity

Recall that OLS requires constant variance because the even spread of residuals is assumed for both FF and tt tests. To examine constant variance, we can produce (read as "make up") a baseline plot to demonstrate what constant variance in the residuals should" look like.

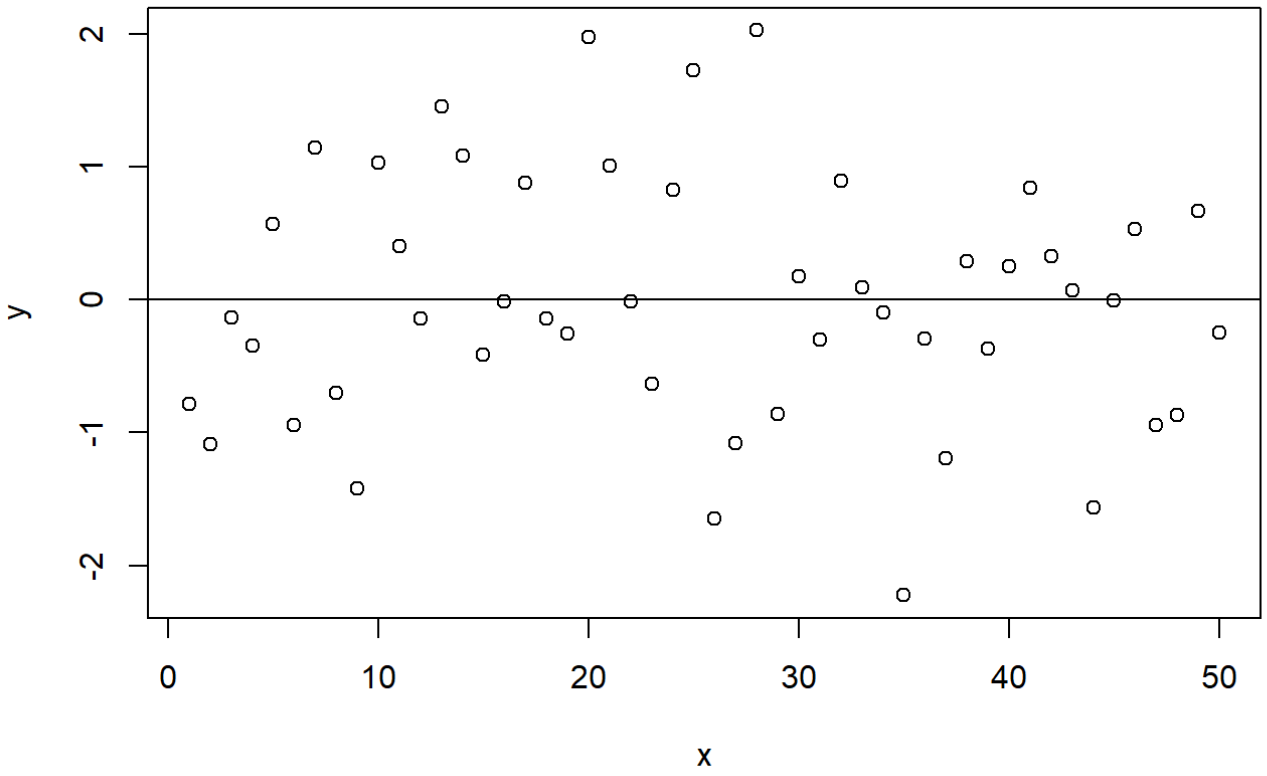

Figure 15.2.5: Constant Variance

As we can see in Figure 15.2.5, the residuals are spread evenly and in a seemingly random fashion, much like the sneeze plot" discussed in Chapter 10. This is the ideal pattern, indicating that the residuals do not vary systematically over the range of the predicted value for XX. The residuals are homoscedastistic, and thus provide the appropriate basis for the FF and tt tests needed for evaluating your hypotheses.

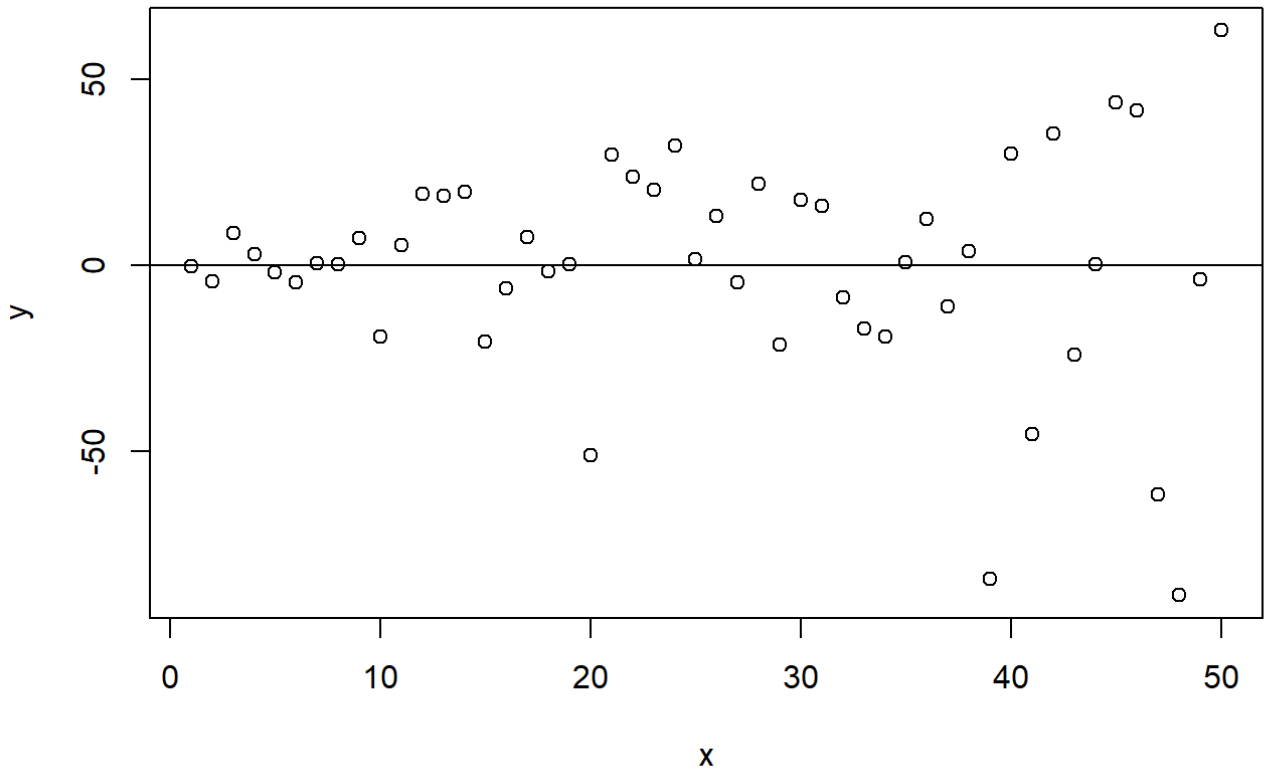

Figure 15.2.6: Heteroscedasticity

The first step in determining whether we have constant variance is to plot the the residuals by the fitted values for YY, as follows:[25]

```
ds.small$fit.r <- ols1$residuals
ds.small$fit.p <- ols1$fitted.values
ggplot(ds.small, aes(fit.p, fit.r)) +
  geom_jitter(shape = 1) +
  geom_hline(yintercept = 0, color = "red") +
  ylab("Residuals") +
  xlab("Fitted")
```

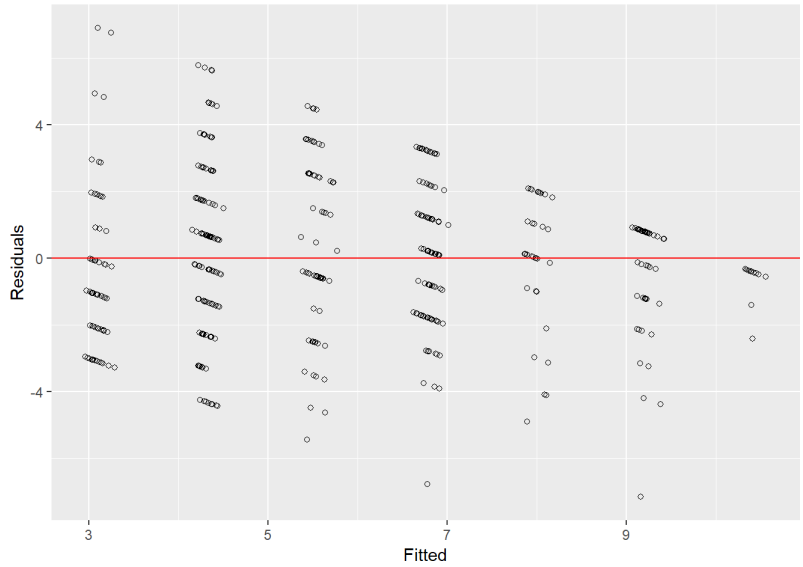

Figure 15.2.7: Multiple Regression Residuals and Fitted Values

Based on the pattern evident in Figure 15.2.7, the residuals appear to show heteroscedasticity. We can test for non-constant error using the Breusch-Pagan (aka Cook-Weisberg) test. This tests the null hypothesis that the error variance is constant, therefore a small p value would indicate that we have heteroscedasticity. In R we can use the ncvTest function from the car package.

```
library(car)
ncvTest(ols1)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 12.70938    Df = 1     p = 0.0003638269
```

The non-constant variance test provides confirmation that the residuals from our model are heteroscedastistic.

What are the implications? Our tt-tests for the estimated partial regression coefficients assumed constant variance. With the evidence of heteroscedasticity, we conclude that these tests are unreliable (the precision of our estimates will be greater in some ranges of XX than others).

They are several steps that can be considered when confronted by heteroscedasticity in the residuals. First, we can consider whether we need to re-specify the model, possibly because we have some omitted variables. If model re-specification does not correct the problem, we can use non-OLS regression techniques that include robust estimated standard errors. Robust standard errors are appropriate when error variance is unknown. Robust standard errors do not change the estimate of BB, but adjust the estimated standard error of each coefficient, SE(B)SE(B), thus giving more accurate pp values. In this example, we draw on White's (1980)[26] method to calculate robust standard errors.

White uses a **heteroscedasticity consistent covariance matrix** (hccm) to calculate standard errors when the error term has non-constant variance. Under the OLS assumption of constant error variance, the covariance matrix of bb is:

$V(b)=(X'X)-1X'V(y)X(X'X)-1V(b)=(X'X)-1X'V(y)X(X'X)-1$

where $V(y)=\sigma 2eInV(y)=\sigma e2In$,

therefore,

$V(b)=\sigma 2e(X'X)-1V(b)=\sigma e2(X'X)-1$.

If the error terms have distinct variances, a consistent estimator constrains $\Sigma\Sigma$ to a diagonal matrix of the squared residuals,

$\Sigma=diag(\sigma 21,\ldots,\sigma 2n)\Sigma=diag(\sigma 12,\ldots,\sigma n2)$

where $\sigma 2i\sigma i2$ is estimated by e2iei2. Therefore the hccm estimator is expressed as:

Vhccm(b)=(X′X)−1X′diag(e2i,…,e2n)X(X′X)−1Vhccm(b)=(X′X)−1X′diag(ei2,…,en2)X(X′X)−1

We can use the `hccm` function from the `car` package to calculate the robust standard errors for our regression model, predicting perceived environmental risk (YY) with political ideology, age, education and income as the XX variables.

```
library(car)
hccm(ols1) %>% diag() %>% sqrt()
```

```
##    (Intercept)            age      education         income          ideol
## 0.668778725013 0.008030365625 0.069824489564 0.000002320899 0.060039031426
```

Using the `hccm` function we can create a function in `R` that will calculate the robust standard errors and the subsequent tt-values and pp-values.

```
library(car)
robust.se <- function(model) {
  s <- summary(model)
  wse <- sqrt(diag(hccm(ols1)))
  t <- model$coefficients/wse
  p <- 2*pnorm(-abs(t))
  results <- cbind(model$coefficients, wse, t, p)
  dimnames(results) <- dimnames(s$coefficients)
  results
}
```

We can then compare our results with the original simple regression model results.

```
summary(ols1)
```

```
##
## Call:
## lm(formula = glbcc_risk ~ age + education + income + ideol, data = ds.small)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1617 -1.7131 -0.0584  1.7216  6.8981
##
## Coefficients:
##                  Estimate    Std. Error t value            Pr(>|t|)
## (Intercept) 12.0848259959  0.7246993630  16.676 <0.0000000000000002 ***
## age         -0.0055585796  0.0084072695  -0.661               0.509
## education   -0.0186146680  0.0697901408  -0.267               0.790
## income       0.0000001923  0.0000022269   0.086               0.931
## ideol       -1.2235648372  0.0663035792 -18.454 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.353 on 445 degrees of freedom
## Multiple R-squared:  0.4365, Adjusted R-squared:  0.4315
## F-statistic: 86.19 on 4 and 445 DF,  p-value: < 0.00000000000000022
```

```
robust.se(ols1)
```

```
##                     Estimate     Std. Error      t value
## (Intercept) 12.0848259958670 0.668778725013  18.06999168
## age         -0.0055585796372 0.008030365625  -0.69219509
## education   -0.0186146679570 0.069824489564  -0.26659225
## income       0.0000001922905 0.000002320899   0.08285175
## ideol       -1.2235648372311 0.060039031426 -20.37948994
##
## (Intercept) 0.000000000000000000000000000000000000000000000000000000000000000000000000000000000(
## age         0.488814823267768150394374515599338337779045104980468750000000000000000(
## education   0.789783121379820318708198101376183331012725830078125000000000000000000(
## income      0.933969416381485006972695828153518959879875183105468750000000000000000(
## ideol       0.000000000000000000000000000000000000000000000000000000000000000000000000000000000(
```

As we see the estimated BB's remain the same, but the estimated standard errors, tt-values and pp-values are adjusted to reflect the robust estimation. Despite these adjustments, the results of the hypothesis test remain unchanged.

It is important to note that, while robust estimators can help atone for heteroscedasticity in your models, their use *should not* be seen as an alternative to careful model construction. The first step should always be to evaluate your model specification and functional form (e.g., the use of polynomials, inclusion of relevant variables), as well as possible measurement error, before resorting to robust estimation.

15.2.3 Independence of EE

As noted above, we cannot test for the assumption that the error term EE is independent of the XX's. However we can test to see whether the error terms, EiEi, are correlated with each other. One of the assumptions of OLS is that E(∈i)≠E(∈j)E(∈i)≠E(∈j) for i≠ji≠j. When there is a relationship between the residuals, this is referred to as serial correlation or **autocorrelation**. Autocorrelation is most likely to occur with time-series data, however it can occur with cross-sectional data as well. To test for autocorrelation we use the Durbin-Watson, dd, test statistic. The dd statistic is expressed as:

d=∑ni=2(Ei−Ei−1)2∑ni=1E2i(15.2)(15.2)d=∑i=2n(Ei−Ei−1)2∑i=1nEi2

The dd statistics ranges from 00 to 44; 0≤d≤40≤d≤4. A 00 indicates perfect positive correction, 44 indicates perfect negative correlation, and a 22 indicates no autocorrelation. Therefore, we look for values of dd that are close to 22.

We can use the `dwtest` function in the `lmtest` package to test the null hypothesis that autocorrelation is 00, meaning that we don't have autocorrelation.

```
library(lmtest)
dwtest(ols1)
```

```
##
##  Durbin-Watson test
##
## data:  ols1
## DW = 1.9008, p-value = 0.1441
## alternative hypothesis: true autocorrelation is greater than 0
```

Generally, a Durbin-Watson result between 1.5 and 2.5 indicates, that any autocorrelation in the data will not have a discernible effect on your estimates. The test for our example model indicates that we do not have an autocorrelation problem with this model. If we did find autocorrelation, we would need to respecify our model to account for (or estimate) the relationships among the error terms. In time series analysis, where observations are taken sequentially over time, we would typically include a "lag" term (in which the value of YY in period tt is predicted by the value of YY in period t−1t−1). This is a typical AR1AR1 model, which

would be discussed in a time-series analysis course. The entangled residuals can, of course, be much more complex, and require more specialized models (e.g., ARIMA or vector-autoregression models). These approaches are beyond the scope of this text.

15.2.4 Normality of the Residuals

This is a critical assumption for OLS because (along with homoscedasticity) it is required for hypothesis tests and confidence interval estimation. It is particularly sensitive with small samples. Note that non-normality will increase sample-to-sample variation in model estimates.

To examine normality of the residuals we first plot the residuals and then run what is known as the Shapiro-Wilk normality test. Here we run the test on our example model, and plot the residuals.

```
p1 <- ggplot(ds.small, aes(fit.r)) +
  geom_histogram(bins = 10, color = "black", fill = "white")
```

```
p2 <- ggplot(ds.small, aes(fit.r)) +
  geom_density() +
  stat_function(fun = dnorm, args = list(mean = mean(ds.small$fit.r),
                                         sd = sd(ds.small$fit.r)),
                color = "dodgerblue", size = 2, alpha = .5)
```

```
p3 <- ggplot(ds.small, aes("", fit.r)) +
  geom_boxplot()
```

```
p4 <- ggplot(ds.small, aes(sample = fit.r)) +
  stat_qq(shape = 1) +
  stat_qq_line(size = 1.5, alpha = .5)
```

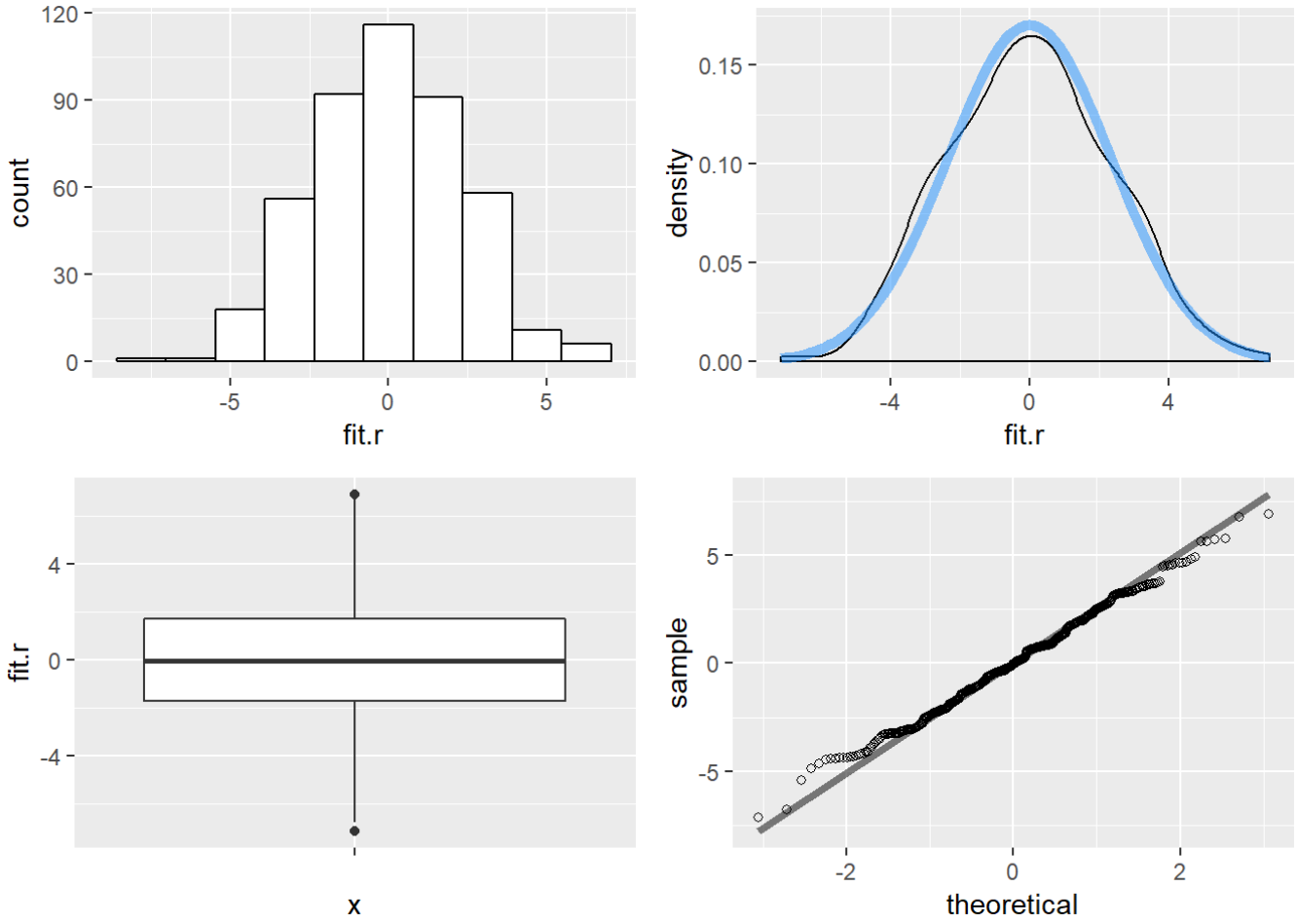

Figure 15.2.8: Multiple Regression Residuals

It appears from the graphs, on the basis of an ocular test“, that the residuals are potentially normally distributed. Therefore, to perform a statistical test for non-normality, we use the Shapiro-Wilk, WW, test statistic. WW is expressed as:

W=(∑ni=1aix(i))2∑ni=1(xi−¯x)2(15.3)(15.3)W=(∑i=1naix(i))2∑i=1n(xi−x¯)2

where x(i)x(i) are the ordered sample values and aiai are constants generated from the means, variances, and covariances of the order statistics from a normal distribution. The Shapiro-Wilk tests the null hypothesis that the residuals are normally distributed. To perform this test in `R` , use the `shapiro.test` function.

```
shapiro.test(ols1$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  ols1$residuals
## W = 0.99566, p-value = 0.2485
```

Since we have a relatively large pp value we fail to reject the null hypothesis of normally distributed errors. Our residuals are, accoridng to our visual examination and this test, normally distributed.

To adjust for non-normal errors we can use robust estimators, as discussed earlier with respect to heteroscedasticity. Robust estimators correct for non-normality, but produce estimated standard errors of the partial regression coefficients that tend to be larger, and hence produce less model precision. Other possible steps, where warranted, include transformation of variables that may have non-linear relationships with YY. Typically this involves taking log transformations of the suspect variables.

15.2.5 Outliers, Leverage, and Influence

Apart from the distributional behavior of residuals, it is also important to examine the residuals for unusual" observations. Unusual observations in the data may be cases of mis-coding (e.g., −99−99), mis-measurement, or perhaps special cases that require different kinds of treatment in the model. All of these may appear as unusual cases that are observed in your diagnostic analysis. The unusual cases that we should be most concerned about are regression outliers, that are potentially influential and that are suspect because of their differences from other cases.

Why should we worry about outliers? Recall that OLS minimizes the sum of the squared residuals for a model. Unusual cases – which by definition will have large outliers – have the potential to substantially influence our estimates of BB because their already large residuals are squared. A large outlier can thus result in OLS estimates that change the model intercept and slope.

There are several steps that can help identify outliers and their effects on your model. The first – and most obvious – is to examine the range of values in your YY and XX variables. Do they fall within the appropriate ranges?

This step – too often omitted even by experienced analysts – can help you avoid often agonizing mis-steps that result from inclusion of miscoded data or missing values (e.g., -99“) that need to be recoded before running your model. If you fail to identify these problems, they will show up in your residual analysis as outliers. But it is much easier to catch the problem *before* you run your model.

But sometimes we find outliers for reasons other than mis-codes, and identification requires careful examination of your residuals. First we discuss how to find outliers – unusual values of YY – and leverage – unusual values of XX – since they are closely related.

15.2.6 Outliers

A regression outlier is an observation that has an unusual value on the dependent variable YY, conditioned on the values of the independent variables, XX. Note that an outlier can have a large residual value, but not necessarily affect the estimated slope or intercept. Below we examine a few ways to identify potential outliers, and their effects on our estimated slope coefficients.

Using the regression example, we first plot the residuals to look for any possible outliers. In this plot we are plotting the raw residuals for each of the 500500 observations. This is shown in Figure 15.2.9.

```
ggplot(ds.small, aes(row.names(ds.small), fit.r)) +
  geom_point(shape = 1) +
  geom_hline(yintercept = 0, color = "red")
```

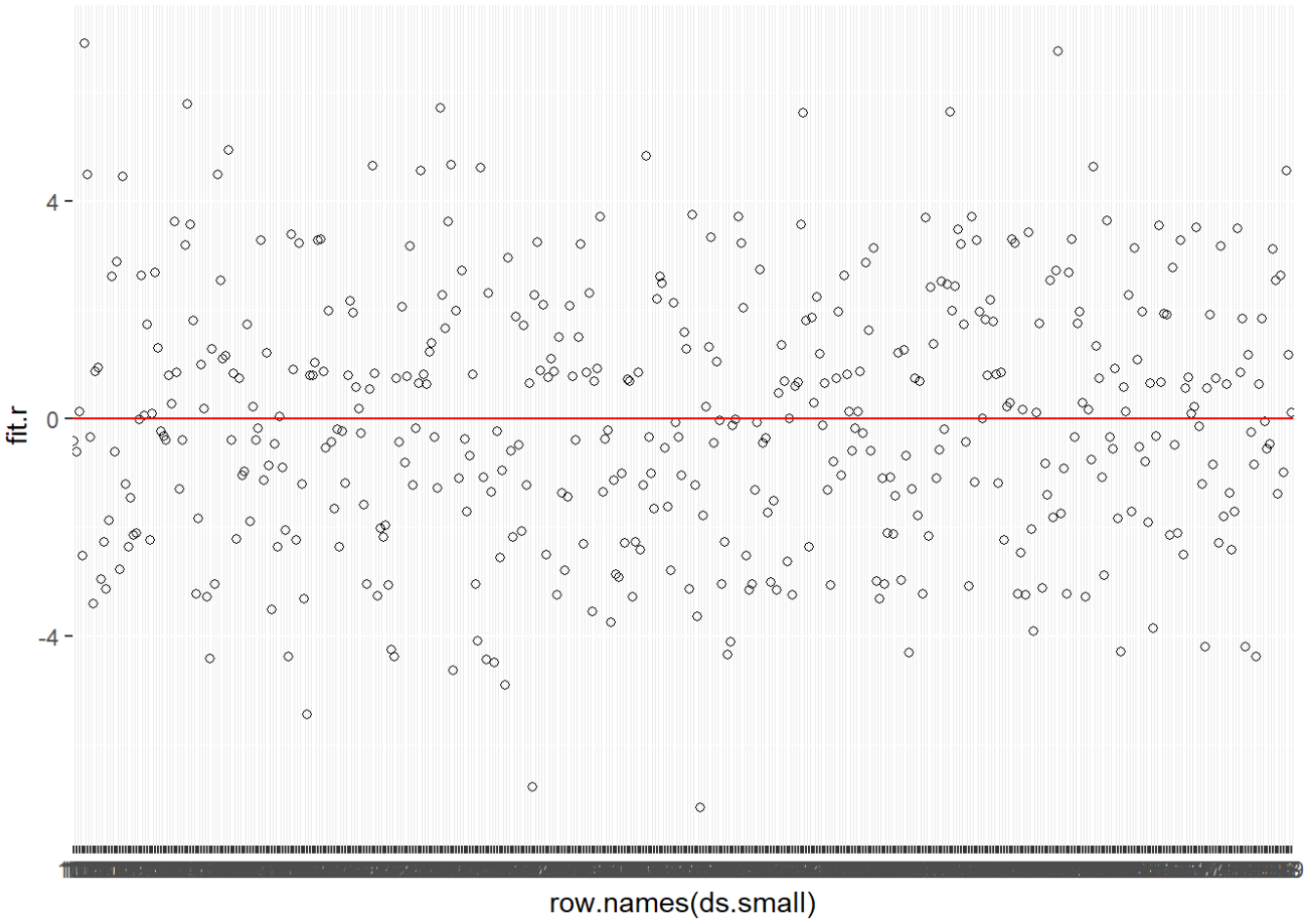

Figure 15.2.9: Index Plot of Residuals: Multiple Regression

Next, we can sort the residuals and find the case with the largest absolute value and examine that case.

```
#  Sort the residuals
output.1 <- sort(ols1$residuals)  # smallest first
output.2 <- sort(ols1$residuals, decreasing = TRUE) # largest first

#  The head function return the top results, the argument 1 returns 1 variable only
head(output.1, 1) # smallest residual absolute value
```

```
##       333
## -7.161695
```

```
head(output.2, 1) # largest residual absolute value
```

```
##      104
## 6.898077
```

Then, we can examine the XX and YY values of those cases on key variables. Here we examine the values across all independent variables in the model.

```
ds.small[c(298,94),c("age","education","income","ideol","glbcc_risk")] # [c(row numbe
```

```
##     age education income ideol glbcc_risk
## 333  69         6 100000     2          2
## 104  55         7  94000     7         10
```

By examining the case of 298, we can see that this is outlier because the observed values of YY are far from what would be expected, given the values of XX. A wealthy older liberal would most likely rate climate change as riskier than a 2. In case 94, a strong conservaitive rates climate change risk at the lowest possible value. This observation, while not consistent with the estimated relationship between ideology and environmental concern, is certainly not implausible. But the unusual appearance of a case with a strong conservative leaning, and high risk of cliamte change results in a large residual.

What we really want to know is: does any particular case substantially change the regression results? If a case substantively change the results than it is said to have influence. Individual cases can be outliers, but still be influential. Note that DFBETAS are **case statistics**, therefore a DFBETA value will be calculated for each variable for each case.

DFBETAS

DFBETAS measure the influence of case ii on the jj estimated coefficients. Specifically, it asks by how many standard errors does BjBj change when case ii is removed DFBETAS are expressed as:

DFBETASij=Bj(−i)−BjSE(Bj)(15.4)(15.4)DFBETASij=Bj(−i)−BjSE(Bj)

Note that if DFBETAS >0>0, then case ii pulls BjBj *up*, and if DFBETAS <0<0, then case ii pulls BjBj *down*. In general, if |DFBETASij|>2√n|DFBETASij|>2n then these cases warrant further examination. Note that this approach gets the top 5% of influential cases, given the sample size. For both simple (bi-variate) and multiple regression models the DFBETA cut-offs can be calculated in `R` .

```
df <- 2/sqrt(500)
df
```

```
## [1] 0.08944272
```

In this case, if |DFBETAS|>0.0894427|DFBETAS|>0.0894427 then they can be examined for possible influence. Note, however, than in large datasets this may prove to be difficult, so you should examine the largest DFBETAS first. In our example, we will look only at the largest 5 DFBETAS.

To calculate the DFBETAS we use the `dfbetas` function. Then we examine the DFBETA values for the first five rows of our data.

```
df.ols1 <- dfbetas(ols1)
df.ols1[1:5,]
```

```
##   (Intercept)          age   education      income        ideol
## 1 -0.004396485  0.005554545  0.01043817 -0.01548697 -0.005616679
## 2  0.046302381 -0.007569305 -0.02671961 -0.01401653 -0.042323468
## 3 -0.002896270  0.018301623 -0.01946054  0.02534233 -0.023111519
## 5 -0.072106074  0.060263914  0.02966501  0.01243482  0.015464937
## 7 -0.057608817 -0.005345142 -0.04948456  0.06456577  0.134103149
```

We can then plot the DFBETAS for each of the IVs in our regression models, and create lines for ±0.089±0.089. Figure 15.2.10 shows the DFBETAS for each variable in the multiple regression model.

```
melt(df.ols1, varnames = c("index", "variable")) %>%
  ggplot(aes(index, value)) +
  geom_point() +
  geom_hline(yintercept = df) +
  geom_hline(yintercept = -df) +
  facet_wrap(~ variable, scales = "free")
```

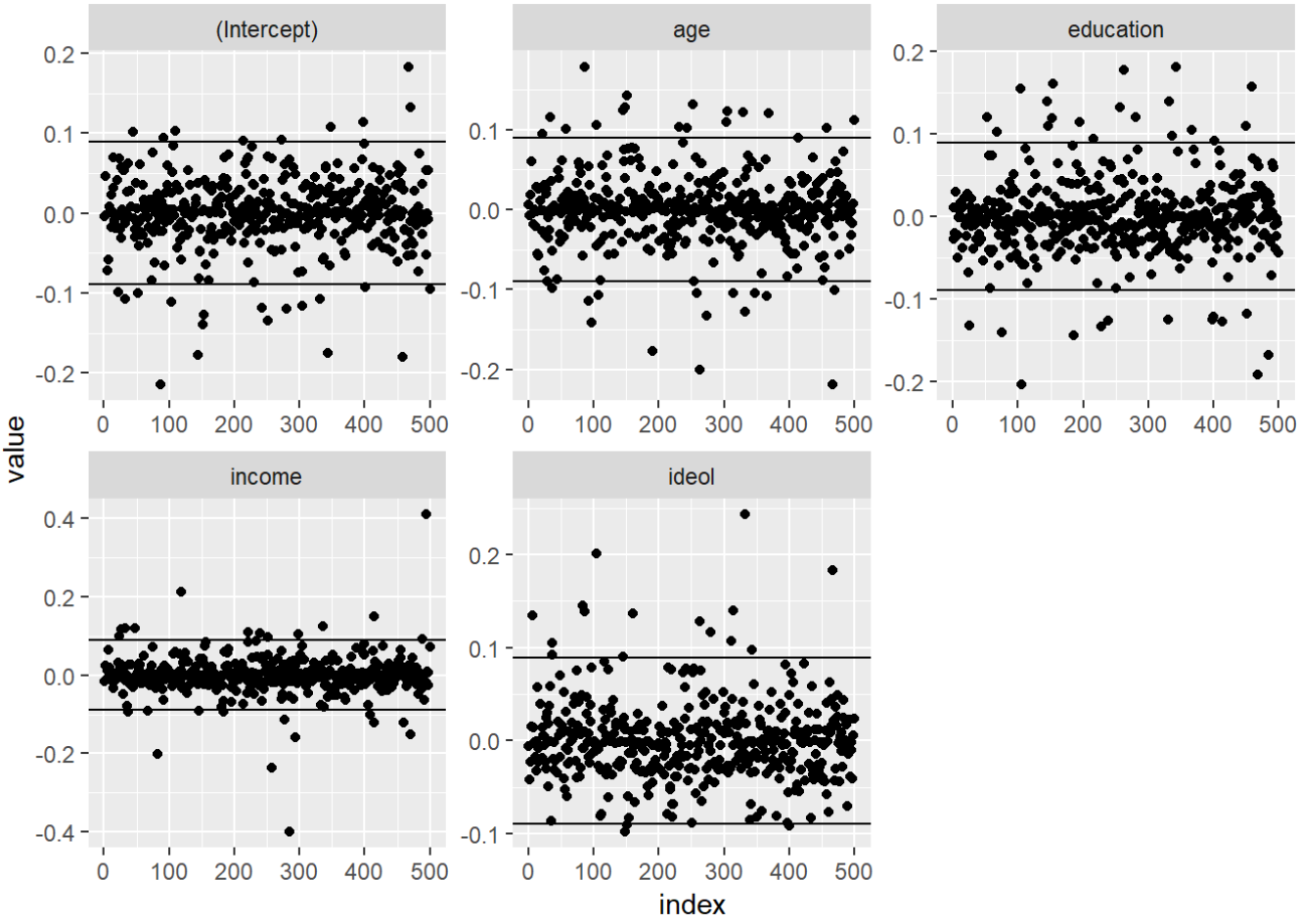

Figure 15.2.10: Index Plot of DFBETAS: Multiple Regression

As can be seen, several cases seem to exceed the 0.0890.089 cut-off. Next we find the case with the highest absolute DFBETA value, and examine the XX and YY values for that case.

```
#  Return Absolute Value dfbeta
names(df.ols1) <- row.names(ds.small)
df.ols1[abs(df.ols1) == max(abs(df.ols1))]
```

```
##      <NA>
## 0.4112137
```

```
# a observation name may not be returned - let's figure out the observation

#  convert df.osl1 from matrix to dataframe
class(df.ols1)
```

```
## [1] "matrix"
```

```
df2.ols1 <- as.data.frame(df.ols1)

#  add an id variable
df2.ols1$id <- 1:450 #  generate a new observation number

#  head function returns one value, based on ,1
#  syntax - head(data_set[with(data_set, order(+/-variable)), ], 1)

#  Ideology
head(df2.ols1[with(df2.ols1, order(-ideol)), ], 1) # order declining
```

```
##      (Intercept)        age   education      income     ideol  id
## 333 -0.001083869 -0.1276632 -0.04252348 -0.07591519 0.2438799 298
```

```
head(df2.ols1[with(df2.ols1, order(+ideol)), ], 1) # order increasing
```

```
##     (Intercept)       age   education     income       ideol  id
## 148  -0.0477082 0.1279219 -0.03641922 0.04291471 -0.09833372 131
```

```
#  Income
head(df2.ols1[with(df2.ols1, order(-income)), ], 1) # order declining
```

```
##     (Intercept)         age    education    income       ideol  id
## 494 -0.05137992 -0.01514244 -0.009938873 0.4112137 -0.03873292 445
```

```
head(df2.ols1[with(df2.ols1, order(+income)), ], 1) # order increasing
```

```
##     (Intercept)         age  education     income      ideol  id
## 284  0.06766781 -0.06611698 0.08166577 -0.4001515 0.04501527 254
```

```
#  Age
head(df2.ols1[with(df2.ols1, order(-age)), ], 1) # order declining
```

```
##    (Intercept)       age  education      income     ideol id
## 87  -0.2146905 0.1786665 0.04131316 -0.01755352 0.1390403 78
```

```
head(df2.ols1[with(df2.ols1, order(+age)), ], 1) # order increasing
```

```
##     (Intercept)        age  education     income     ideol  id
## 467    0.183455 -0.2193257 -0.1906404 0.02477437 0.1832784 420
```

```
#  Education - we find the amount - ID 308 for edu
head(df2.ols1[with(df2.ols1, order(-education)), ], 1) # order declining
```

```
##     (Intercept)        age education      income      ideol  id
## 343  -0.1751724 0.06071469 0.1813973 -0.05557382 0.09717012 308
```

```
head(df2.ols1[with(df2.ols1, order(+education)), ], 1) # order increasing
```

```
##     (Intercept)       age  education      income        ideol id
## 105  0.05091437 0.1062966 -0.2033285 -0.02741242 -0.005880984 95
```

```
#  View the output
df.ols1[abs(df.ols1) == max(abs(df.ols1))]
```

```
##      <NA>
## 0.4112137
```

```
df.ols1[c(308),] # dfbeta number is observation 131 - education
```

```
## (Intercept)         age   education      income       ideol
## -0.17517243  0.06071469  0.18139726 -0.05557382  0.09717012
```

```
ds.small[c(308), c("age", "education", "income", "ideol", "glbcc_risk")]
```

```
##     age education income ideol glbcc_risk
## 343  51         2  81000     3          4
```

Note that this “severe outlier” is indeed an interesting case – a 51 year old with a high school diploma, relatively high income, who is slightly liberal and perceivs low risk for climate change. But this outlier is not implausible, and therefore we can be reassured that – even in this most extreme case – we do not have problematic outliers.

So, having explored the residuals from our model, we found a number of outliers, some with significant influence on our model results. In inspection of the most extreme outlier gave us no cause to worry that the observations were inappropriately distorting our model results. But what should you do if you find puzzling, implausible observations that may influence your model?

First, as always, evaluate your theory. Is it possible that the case represented a class of observations that behave systematically differently than the other cases? This is of particular concern if you have a cluster of cases, all determined to be outliers, that have similar properties. You may need to modify your theory to account for this subgroup. One such example can be found in the study of American politics, wherein the Southern states routinely appeared to behave differently than others. Most careful efforts to model state (and individual) political behavior account for the unique aspects of southern politics, in ways ranging from the addition of dummy variables to interaction terms in regression models.

How would you determine whether the model (and theory) should be revised? Look closely at the deviant cases – what can you learn from them? Try experiments by running the models with controls – dummies and interaction terms. What effects do you observe? If your results suggest theoretical revisions, you will need to collect new data to test your new hypotheses. Remember: In empirical studies, you need to keep your discoveries distinct from your hypothesis tests.

As a last resort, if you have troubling outliers for which you cannot account in theory, you might decide omit those observations from your model and re-run your analyses. We do not recommend this course of action, because it can appear to be a case of jiggering the data" to get the results you want.

15.2.7 Multicollinearity

Multicollinearity is the correlation of the IVs in the model. Note that if any XiXi is a linear combination of other XX's in the model, BiBi cannot be estimated. As discussed previously, the partial regression coefficient strips both the XX's and YY of the overlapping covariation by regressing one XX variable on all other XX variables:

EXi|Xj=Xi−^Xi^Xi=A+BXjEXi|Xj=Xi−X^iX^i=A+BXj

If an X is perfectly predicted by the other XX's, then:

where R2kRk2 is the R2R2 obtained from regressing all XkXk on all other XX's.

We rarely find perfect multicollinearity in practice, but high multicollinearity results in loss of statistical resolution. Such as:

- Large standard errors
- Low tt-stats, high pp-values
- This erodes the resolution of our hypothesis tests
- Enormous sensitivity to small changes in:
- Data
- Model specification

You should always check the correlations between the IVs during the model building process. This is a way to quickly identify possible multicollinearity issues.

```
ds %>%
  dplyr::select(age, education, income, ideol) %>%
  na.omit() %>%
  data.frame() %>%
  cor()
```

```
##                   age   education      income       ideol
## age        1.00000000 -0.06370223 -0.11853753  0.08535126
## education -0.06370223  1.00000000  0.30129917 -0.13770584
## income    -0.11853753  0.30129917  1.00000000  0.04147114
## ideol      0.08535126 -0.13770584  0.04147114  1.00000000
```

There do not appear to be any variables that are so highly correlated that it would result in problems with multicolinearity.

We will discuss two more formal ways to check for multicollinearity. First, is the **Variance Inflation Factor** (VIF), and the second is **tolerance**. The VIF is the degree to which the variance of other coefficients is increased due to the inclusion of the specified variable. It is expressed as:

VIF=11−R2k(15.5)(15.5)VIF=11−Rk2

Note that as R2kRk2 increases the variance of XkXk increases. A general rule of thumb is that VIF>5VIF>5 is problematic.

Another, and related, way to measure multicollinearity is tolerance. The tolerance of any XX, XkXk, is the proportion of its variance not shared with the other XX's.

tolerance=1−R2k(15.6)(15.6)tolerance=1−Rk2

Note that this is mathematically equivalent to 1VIF1VIF. The rule of thumb for acceptable tolerance is partly a function of nn-size:

- If n<50n<50, tolerance should exceed 0.70.7
- If n<300n<300, tolerance should exceed 0.50.5
- If n<600n<600, tolerance should exceed 0.30.3
- If n<1000n<1000, tolerance should exceed 0.10.1

Both VIF and tolerance can be calculated in `R` .

```
library(car)
vif(ols1)
```

```
##       age education    income     ideol
##  1.024094  1.098383  1.101733  1.009105
```

```
1/vif(ols1)
```

```
##       age education    income     ideol
## 0.9764731 0.9104295 0.9076611 0.9909775
```

Note that, for our example model, we are well within acceptable limits on both VIF and tolerance.

If multicollinearity is suspected, what can you do? One option is to drop one of the highly co-linear variables. However, this may result in model mis-specification. As with other modeling considerations, you must use theory as a guide. A second option would be to add new data, thereby lessening the threat posed by multicolinearity. A third option would be to obtain data from specialized samples that maximize independent variation in the collinear variables (e.g., elite samples may disentangle the effects of income, education, and other SES-related variables).

Yet another strategy involves reconsidering why your data are so highly correlated. It may be that your measures are in fact different “indicators” of the same underlying theoretical concept. This can happen, for example, when you measure sets of attitudes that are all influenced by a more general attitude or belief system. In such a case, data scaling is a promising option. This can be accomplished by building an additive scale, or using various scaling options in RR. Another approach would be to use techniques such as factor analysis to tease out the underlying (or latent“) variables represented by your indicator variables. Indeed, the combination of factor analysis and regression modeling is an important and widely used approach, referred to as structural equation modeling (SEM). But that is a topic for another book and another course.