Interpreting Residuals

Michael Noonan

Biol 520C: Statistical modelling for biological data

Table of contents

Model Residuals

Model Residuals

So far we've been fitting models of the general form:
$y_i = \beta_0 + \beta_1 x_i + \ldots + \varepsilon_i$

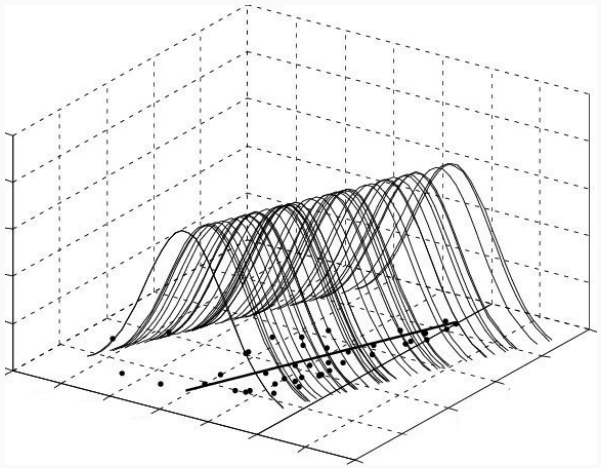
A model isn't always a perfect representation of what's going on in the real world, and there will be deviations between what actually happened (i.e., the observed values), and what the model predicted would happen (i.e., the predicted values).

The difference between the predicted and observed value is called the residual:

$$\text{Residual} = \text{Observed} - \text{Predicted}$$

Residuals as diagnostic tools

By definition, if these models are behaving properly they should result in some amount of residual spread around values predicted by a model's deterministic component.



Source: Zuur et al. 2009

Because residuals are supposed to have very specific behaviour they provide a useful tool for evaluating how well a model fits the data and that the assumptions of the model are being met.

Today we'll cover how to interpret a model's residuals to help you both understand and improve a regression model.

Types of Residuals

The first thing to know about calculating residuals is that there are three forms of residuals:

- Ordinary residuals.
- Standardised residuals.
- Studentised residuals.

Ordinary Residuals

Ordinary residuals are the most commonly used residuals.

They are defined as the difference between the expected and observed values.

For a simple linear regression model of the form $y_i = \beta_0 + \beta_1 x_i$

$\text{Observed}_i = y_i$

$\text{Expected}_i = \beta_0 + \beta_1 x_i$

$\text{Residual}_i = \text{Observed}_i - \text{Expected}_i = y_i - \beta_0 + \beta_1 x_i$

Standardised Residuals

Standardised residuals are ordinary residuals divided by their standard deviation and are useful for identifying outliers.

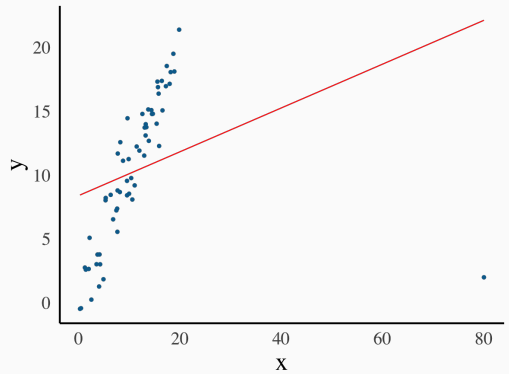
$$\text{Standardised Residual}_i = \frac{\text{Residual}_i}{\text{Standard Deviation of Residuals}}$$

Standardised residuals will have mean $= 0$ and standard deviation $= 1$.

Rule of thumb: If your data are normally distributed, 95% of your data should be ± 2 SDs from the mean. If you have something greater than that, then you're probably looking at an outlier.

Studentised Residuals

If an outlier influences a regression model to such an extent that the estimated fit is "pulled" towards the outlier, standardised residuals may not flag it as an outlier



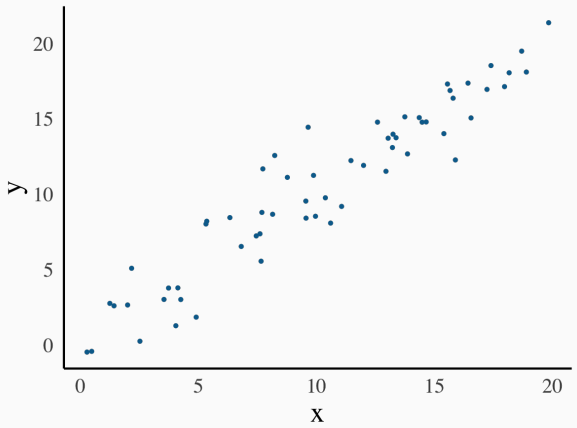
The basic idea behind studentised residuals is to drop observations one at a time and refit the regression model on the remaining n–1 observations. Then, we compare the observed y_i values to their expected values based on the models with the i^{th} observation removed. This produces deleted residuals. Standardising these residuals produces studentised residuals.

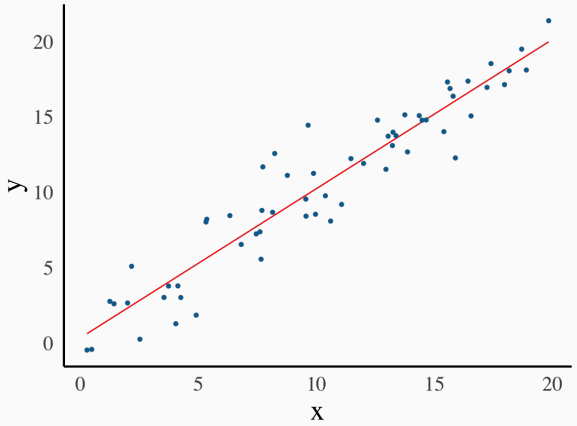
If data point i is 'influential' it pulls the regression line towards itself, and the observation would be close to the predicted response. But, if you removed the outlier, then the regression line would bounce back to the bulk of the data, resulting in a large studentised residual.

Calculating Ordinary Residuals

We'll start with a best case scenario by fiting a model to some data simulated from a simple linear process with Gaussian distributed error.

```
linear <- function(x){
  B_0 <- 0
  B_1 <- 1
  sig <- 2
  eps <- rnorm(n = length(x), sd = sig)
  y = B_0 + B_1*x + eps
  y}

x <- runif(60, min = 0, max = 20)
y <- linear(x)

MODEL <- lm(y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-3.7911 -1.1243 -0.1473  0.9906  4.5536

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept)  0.35535    0.49162   0.723    0.473
x            0.98904    0.04255  23.244   <2e-16 ***

Residual standard error: 1.763 on 58 degrees of freedom
Multiple R-squared:  0.9031,  Adjusted R-squared:  0.9014
F-statistic: 540.3 on 1 and 58 DF,  p-value: < 2.2e-16
```

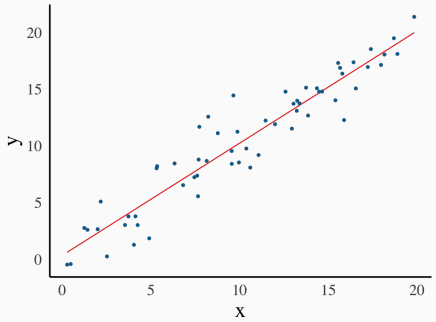




Calculating Ordinary Residuals Cont.

Ordinary residuals = Observed – Predicted.

For our simulated dataset and fitted model we can do this in R:




```
Observed <- y

Predicted <- MODEL$coefficients[1] + MODEL$coefficients[2]*x

Residuals <- Observed - Predicted

head(Residuals)

[1]  2.4202350 -0.4793203  0.5456157 -0.2637988 -3.0652396 -0.9883308
```

Alternatively, you can use the `residuals()` function, ultimately, the result is the same

```
Residuals2 <- residuals(MODEL)

head(Residuals2)

         1          2          3          4          5          6
 2.4202350 -0.4793203  0.5456157 -0.2637988 -3.0652396 -0.9883308
```

Residuals and R^2

The residuals measure how accurately a fitted model predicts the observed data.

They are also used to calculate the coefficient of determination (i.e., R^2).

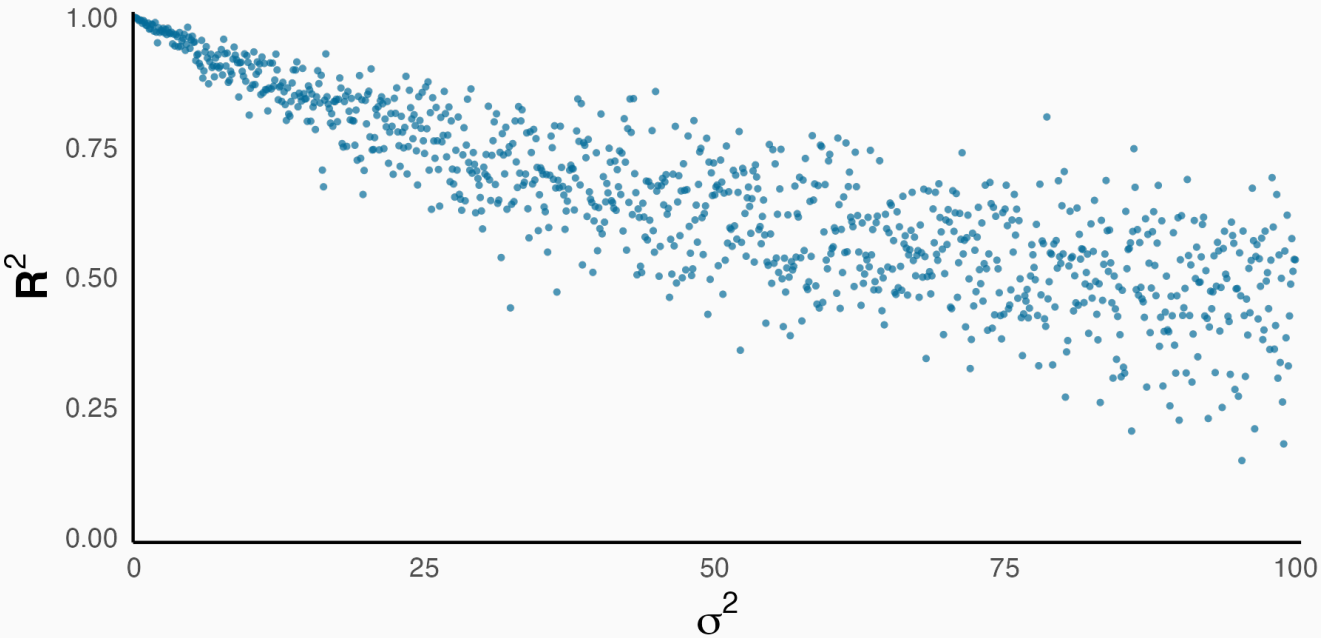
R^2 is the proportion of the variance in the response variable that is predictable from the predictor(s).

```
SS_res <- sum(Residuals^2)

SS_tot <- sum((Observed - mean(Observed))^2)

Rsquare <- 1 - (SS_res/SS_tot)

Rsquare
[1] 0.9030564
```

A note on R^2

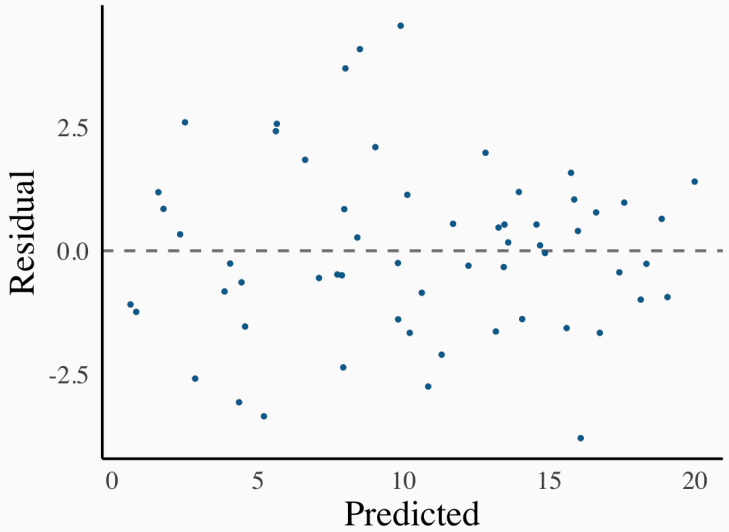
Gaussian regression models are of the form $y_i = \beta_0 + x_i\beta_1 + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ ...and R^2 is the proportion of the variance in the response variable that is predictable.

By definition the R^2 can't explain σ^2, so it's an easy metric to break (even if all the other terms are estimated perfectly).



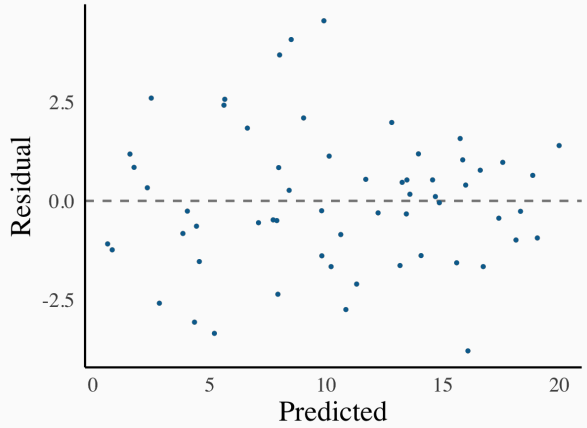
Inspecting Residuals

The most useful way to examine the residuals is by plotting the predicted values of on the x-axis, and the residuals on the y-axis.



The distance from the line at 0 is how bad the prediction was for that value. Positive values for the residual mean the prediction was low, negative values mean the prediction was high, 0 means the model was exactly correct.

Inspecting Residuals Cont.



Notice how in this scenario the residuals:

- Are evenly distributed around 0
- Have low single digit values (i.e., on the order of 2, not 200)
- Lack any clear patterns

This is what you hope to see if a model is performing well. The predictions aren't far from the observations, and there are no remaining patterns that aren't being explained by the model. If the residuals aren't evenly distributed vertically, or they have an outlier, or they have clear patterns, then the model has room for improvement.

Diagnosing Residuals

Diagnosing Residuals

Diagnosing residuals is part science, part art.

The more residual plots you see, the better you'll get at seeing patterns and diagnosing issues.

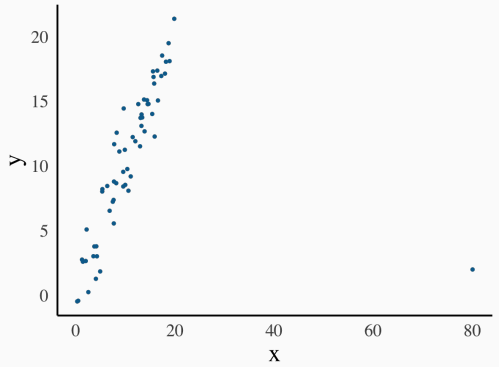
Let's take a look at what happens to the residuals when there are known issues in the data.

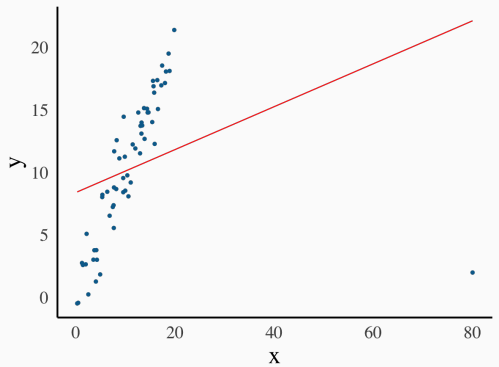
Outliers

Problem: What if the normal range of your data was ~0 to 20, but one of your datapoints had an x value of 80?

Let's use the exact same data as before, but add an outlier

```
x <- c(x, 80)
y <- c(y, 2)

MODEL <- lm(y ~ x)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-20.110  -3.024   1.158   3.905   9.580

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept)  8.39362    1.03457   8.113 3.52e-11 ***
x            0.17146    0.06731   2.547   0.0135 *
---

Residual standard error: 5.428 on 59 degrees of freedom
Multiple R-squared:  0.09907, Adjusted R-squared:  0.0838
F-statistic: 6.488 on 1 and 59 DF,  p-value: 0.01349
```

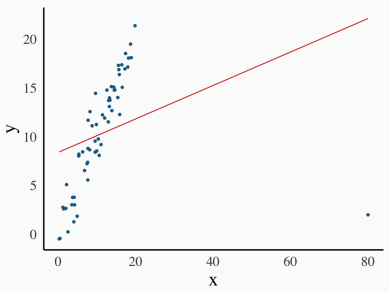




Remember, our intercept was 0 and our slope was 1. What about the R^2

Outliers Cont.

Implications: Because the outlier is so far from the bulk of the data, it has a disproportionate effect on the model and pulls the fit towards itself



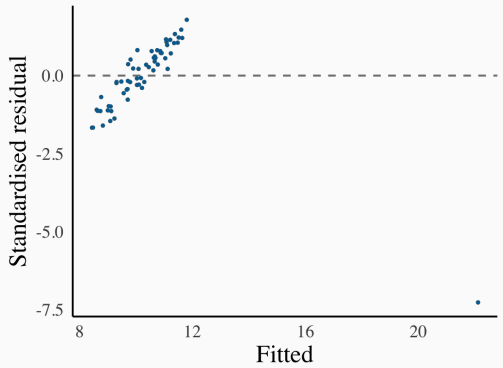
How to solve the issue:

- It's possible that this is a measurement or data entry error. If this is the case, remove the outlier as it is providing misinformation.
- It's possible that what appear to be just a couple outliers are in fact the result of a non-linear relationship between x and y. Consider adding a variable or changing the model.
- If the data is not an entry/measurement error, you should assess the impact of the outlier. E.g., note the coefficients of your current model, then filter out that datapoint from the regression. If the model doesn't change much, there's not much to worry about. If there's a big change, examine the models and decide which one feels better to you given your knowledge of the system. It's okay to discard outliers in a defensible way.

Correction: Filtering outliers

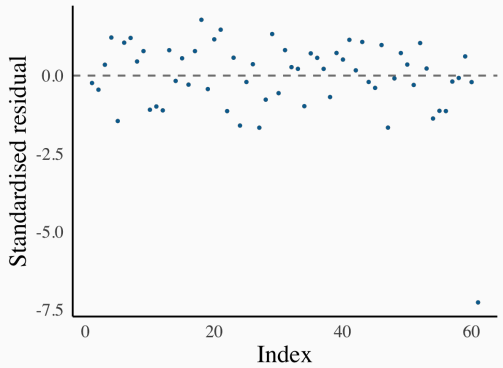
$$\text{Standardised Residual}_i = \frac{\text{Residual}_i}{\text{Standard Deviation of Residuals}}$$

```
Observed <- y

Predicted <- MODEL$coefficients[1] + MODEL$coefficients[2]*x

Residuals <- Observed - Predicted

Residuals <- Residuals/sd(Residuals)

head(Residuals)

[1] -0.2371666 -0.4519790  0.3482399  1.2166389 -1.4501977
      1.0522036
```



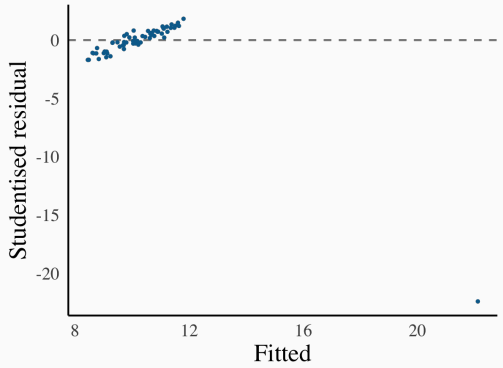
Alternatively, you can use `rstandard()`

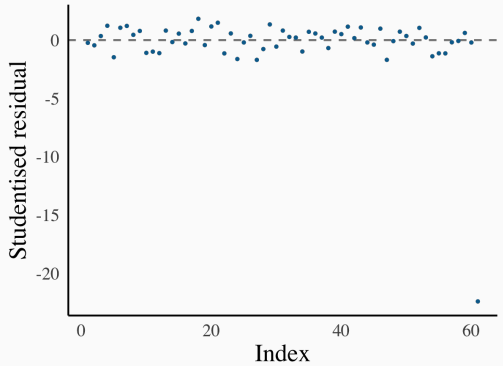
```
Residuals2 <- rstandard(MODEL)

head(Residuals2)

         1          2          3          4          5          6

-0.2378209 -0.4524664  0.3481916  1.2208638 -1.4561621  1.0556393
```



Correction: Filtering outliers cont.


```
R_Student <- vector()
for(i in 1:length(x)){
  x_sub <- x[-i]
  y_sub <- y[-i]

  SUB_MODEL <- lm(y_sub ~ x_sub)

  Predicted <- coef(SUB_MODEL)[1] + coef(SUB_MODEL)[2]*x[i]

  Observed <- y[i]

  RESIDUAL <- Observed - Predicted

  R_Student[i] <- RESIDUAL/sd(residuals(SUB_MODEL))
}

head(R_Student)

[1] -0.2406048 -0.4575714  0.3514427  1.2513493 -1.5017222
      1.0782553

R_Student2 <- rstudent(MODEL)

head(R_Student2)

         1          2          3          4          5          6

-0.2359099 -0.4493959  0.3455835  1.2260592 -1.4704338  1.0566817
```





Correction: Filtering outliers cont.[2]

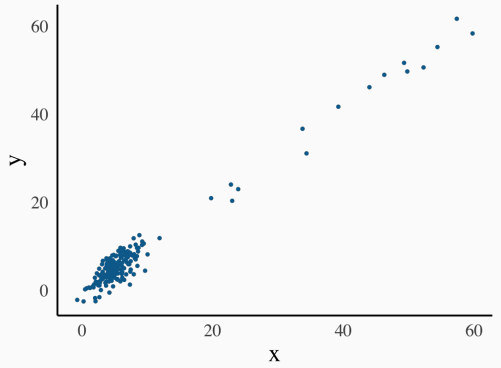
Both standardised and studentised residuals provide strong evidence to support dropping the outlier.

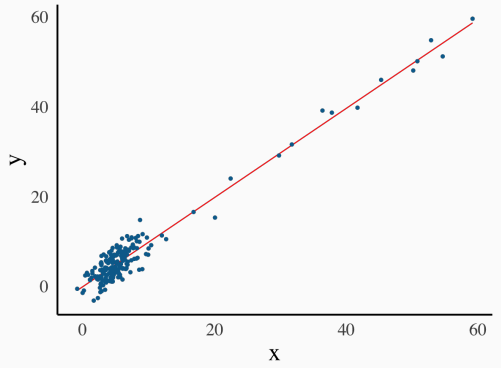
Based on this information, you would then drop the outlier and move forward with your analysis.

Unbalanced x-axis

Problem: Imagine that, during your data collection, your x values were mostly centered around 5, but every now and then you got a very high value.

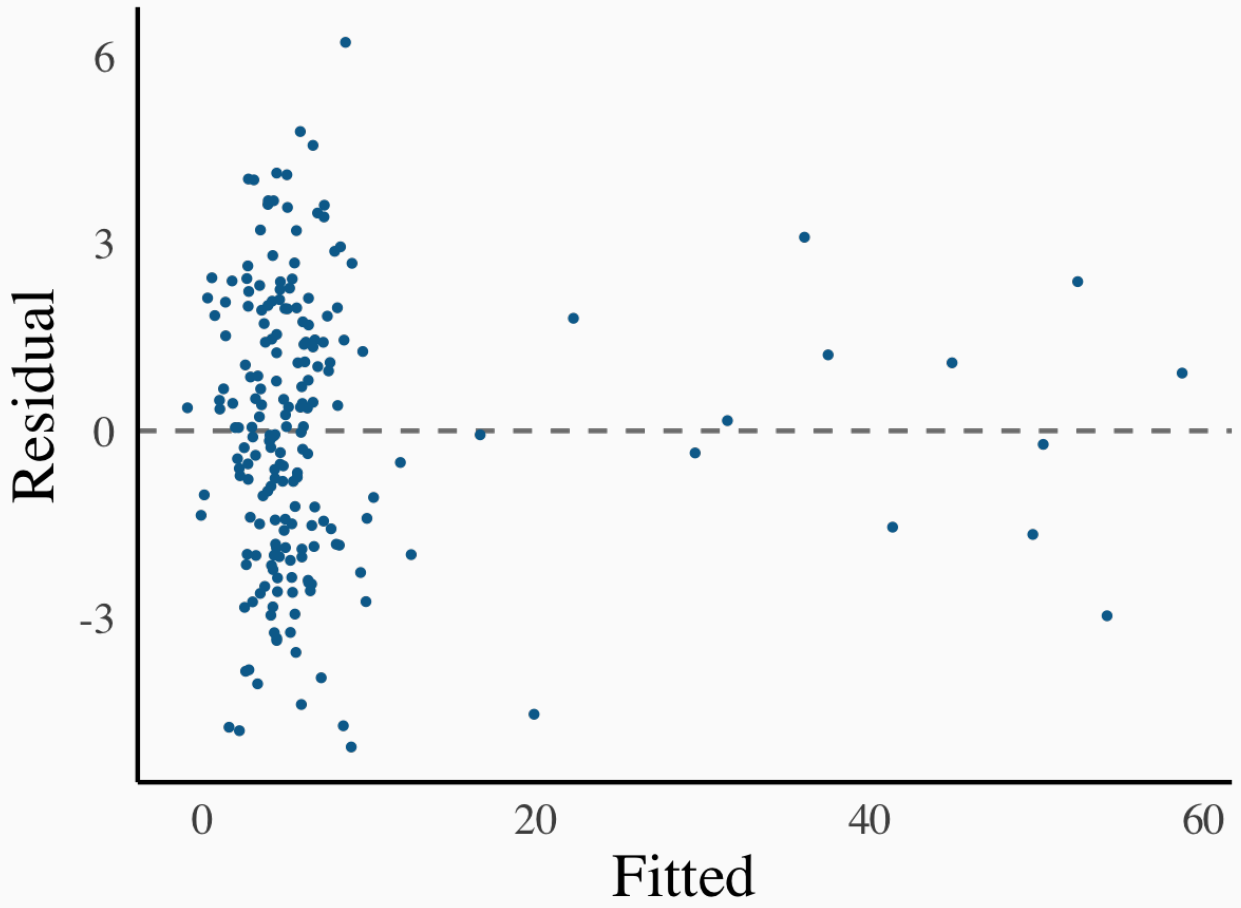
```
x <- c(rnorm(180, 5, 2), runif(20, 0, 60))
y <- linear(x)

MODEL <- lm(y ~ x)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-5.0696 -1.6995  0.0146  1.7043  6.2317

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept) -0.06137    0.19602  -0.313    0.755
x            0.99320    0.01617  61.422   <2e-16 ***
---

Residual standard error: 2.21 on 198 degrees of freedom
Multiple R-squared:  0.9501,  Adjusted R-squared:  0.9499
F-statistic:  3773 on 1 and 198 DF,  p-value: < 2.2e-16
```





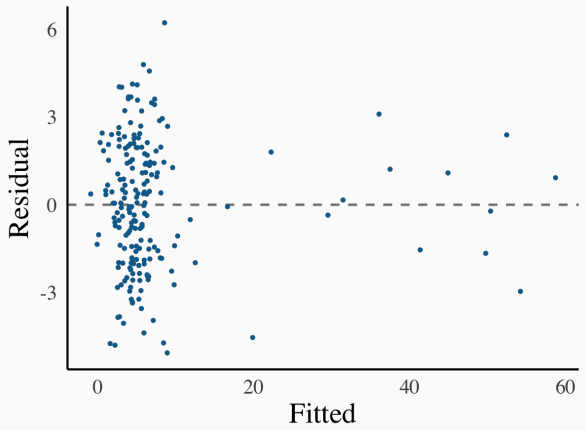
Unbalanced x-axis residuals

And the residuals on this fit would look like this



Unbalanced x-axis cont.

Implications: Sometimes there's actually nothing wrong with your model. Other times, however, an unbalanced x-axis can result in similar problems caused by outliers as we just saw (especially for non-linear relationships). Most of the time you'll find that the model was directionally correct but with inaccurate parameter estimates.



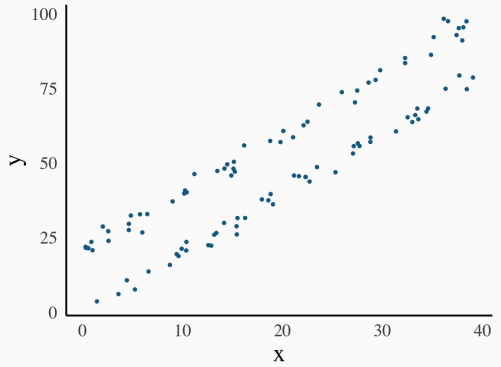
How to solve the issue:

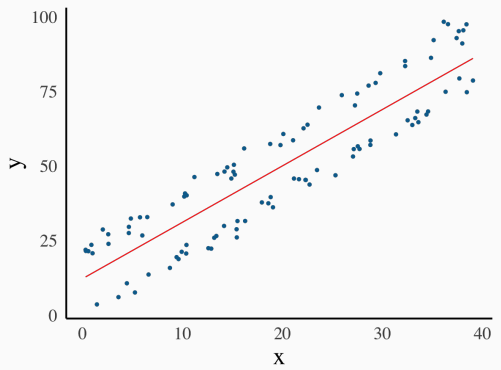
- If you're lucky, no correction is needed.
- The solution to this is almost always to transform your data, typically an explanatory variable.
- If you can, collect more data.
- It's also possible that the model is missing a variable.

Missing Variables

Problem: One of the most common reason why a model struggles to fit a particular dataset is that not all the necessary variables have been included. This particular issue results in a wide range of residual structures, and has a lot of possible solutions.

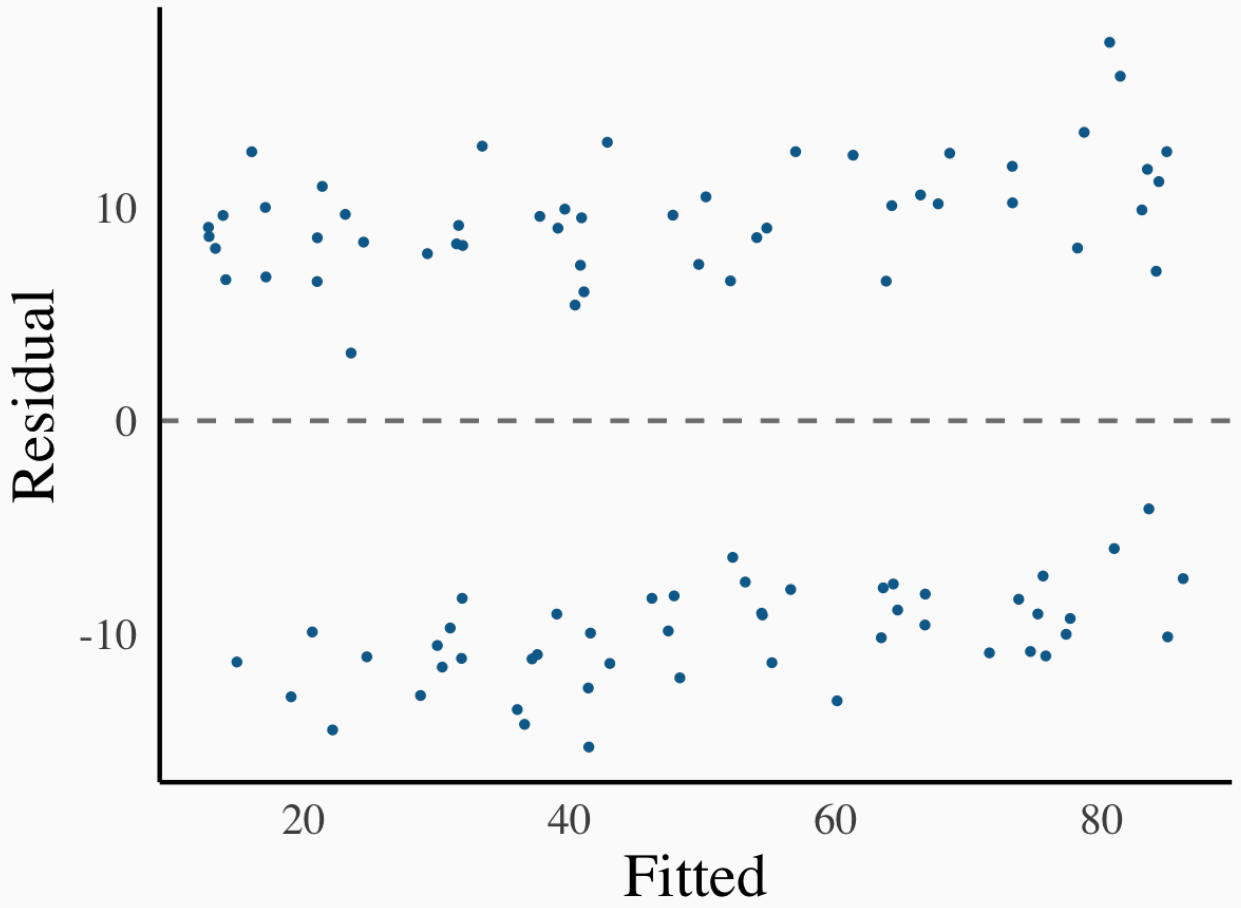
```
linear_2param <- function(x, x_2) {
  B_0 <- 0
  B_1 <- 2
  B_2 <- 20
  sig <- 2
  eps <- rnorm(n = length(x), sd = sig)
  y <- B_0 + B_1*x + B_2*x_2 + eps
  y}

x <- runif(100, 0, 40)
x_2 <- rbinom(100,1,.5)
y <- linear_2param(x, x_2)

MODEL <- lm(y ~ x)

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept)  7.69031    2.11970   3.628 0.000456 ***
x            2.10136    0.09428  22.288  < 2e-16 ***
---
Residual standard error: 10.41 on 98 degrees of freedom
Multiple R-squared:  0.8352,  Adjusted R-squared:  0.8335
F-statistic: 496.7 on 1 and 98 DF,  p-value: < 2.2e-16
```





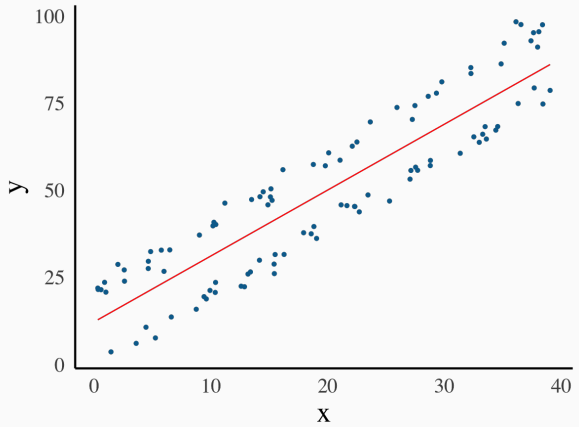
Missing variables residuals

And the residuals on this fit would look like this



Missing Variables cont.

Implications: Notice how the slope is still accurate, but the estimated intercept is both off, and significant. The model isn't completely worthless, but it's definitely not as good as if you had all the variables you needed.



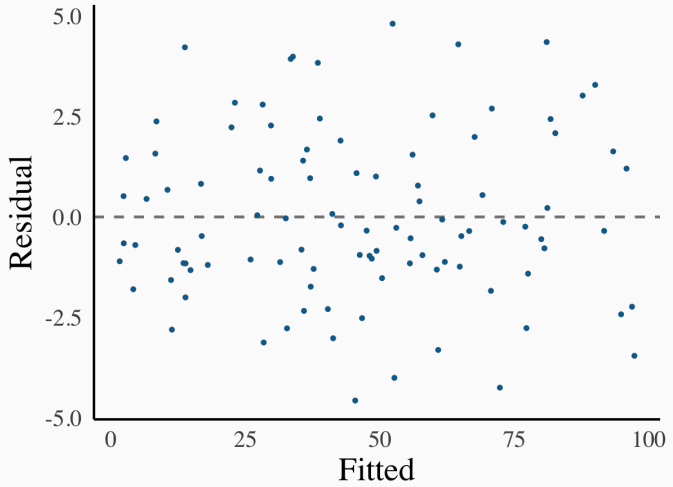
How to solve the issue:

- Depending on the magnitude of the issue, you probably need to deal with the missing variable problem.

Missing Variables Solution

Attempting a fix: Let's add a second parameter to our model.

```
MODEL2 <- lm(y ~ x + x_2)

Call:
lm(formula = y ~ x + x_2)

Residuals:
    Min      1Q  Median      3Q     Max
-5.2565 -1.4237  0.1298  1.3260  4.3949

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept) -0.45906    0.45905    -1.0     0.32
x            2.00473    0.01754   114.3   <2e-16 ***
x_2         20.38987    0.41025    49.7   <2e-16 ***
---

Residual standard error: 2.029 on 97 degrees of freedom
Multiple R-squared:  0.9934,  Adjusted R-squared:  0.9933
F-statistic:  7349 on 2 and 97 DF,  p-value: < 2.2e-16
```

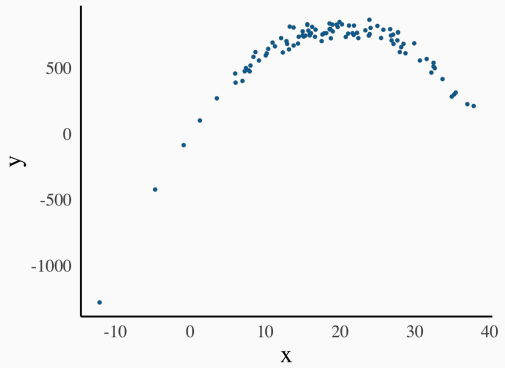


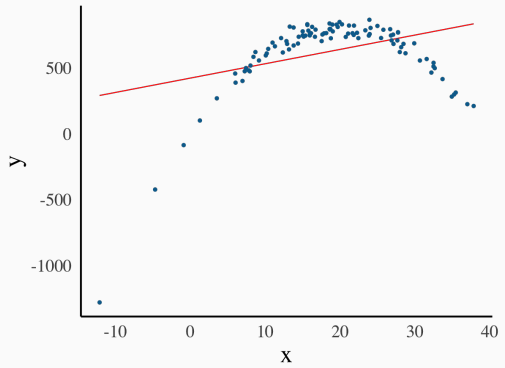
The parameter estimates match the model we simulated from, and fitted model makes far more accurate predictions because it's able to take into account the additional information from x_2

Non-linearity

Problem: Imagine a situation where y tends to be small at small values of x, large and intermediate values of x, but then small again at the largest values of x. This scenario represents a non-linear relationship between x and y, which ends up being very common in practice.

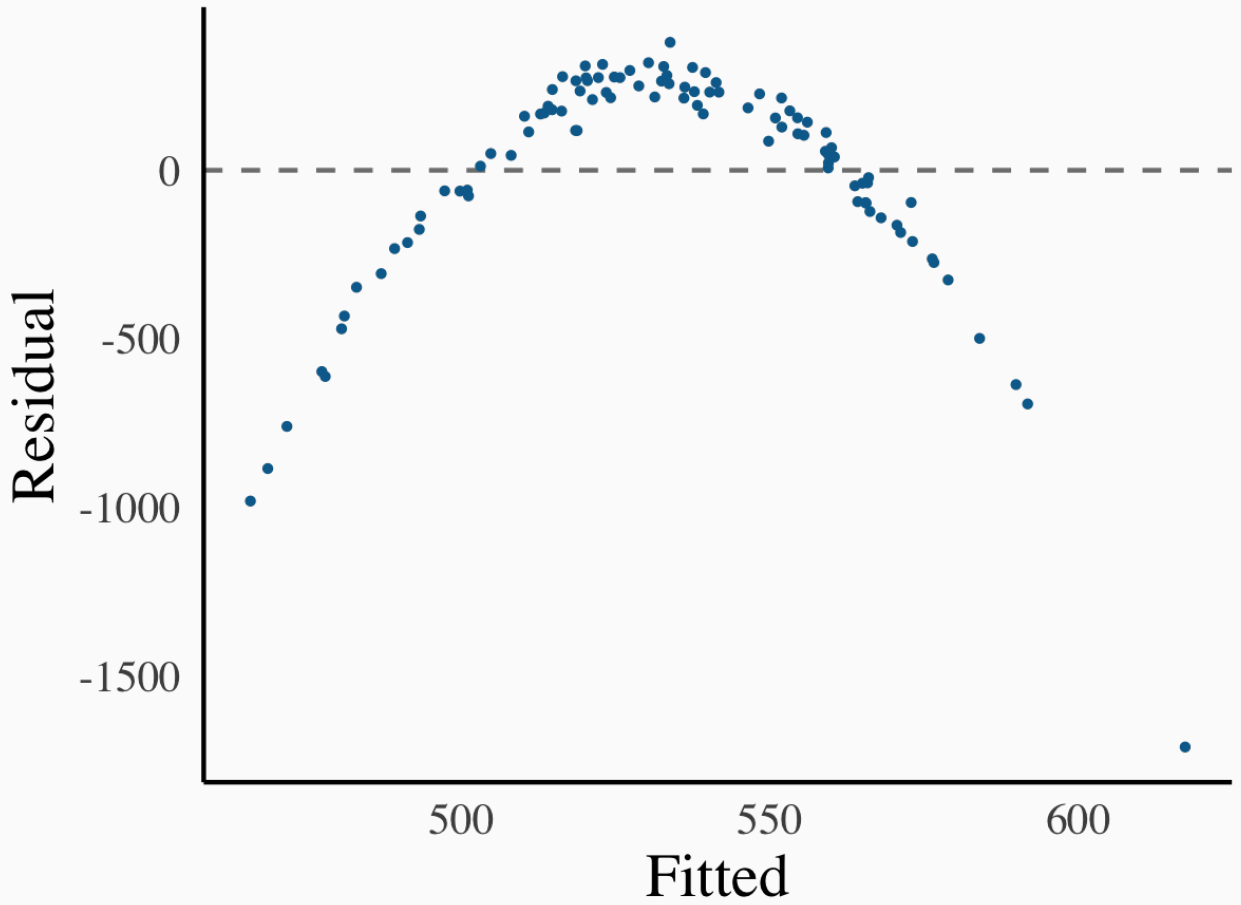
```
quad <- function(x) {
  B_0 <- 0
  B_1 <- 80
  B_2 <- -2
  sig <- 40
  eps <- rnorm(n = length(x), sd = sig)
  y <- B_0 + B_1*x + B_2*x^2 + eps
  y}

x <- rnorm(100, mean = 20, 10)
y <- quad(x)

MODEL <- lm(y ~ x)

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept)  478.594     67.807   7.058 2.44e-10 ***
x              5.247      3.175   1.653    0.102
---
Residual standard error: 330.8 on 98 degrees of freedom
Multiple R-squared:  0.02712, Adjusted R-squared:
      0.01719
F-statistic: 2.732 on 1 and 98 DF,  p-value: 0.1016
```





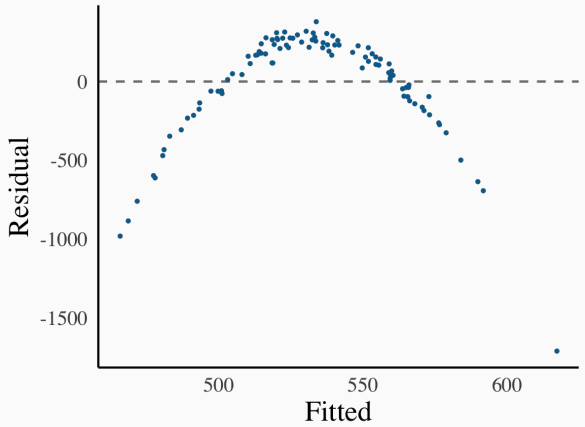
Non-linear residuals

And the residuals on this fit would look like this



Non-linear relationships cont.

Implications: If your model is off, as in the example above, your predictions will be effectively worthless. In situations like this the model is doing very little to explain any relationship between x and y. You can see this by the fact that the residuals (i.e., what's left after the model has made a predicition) look exactly like the data.



How to solve the issue:

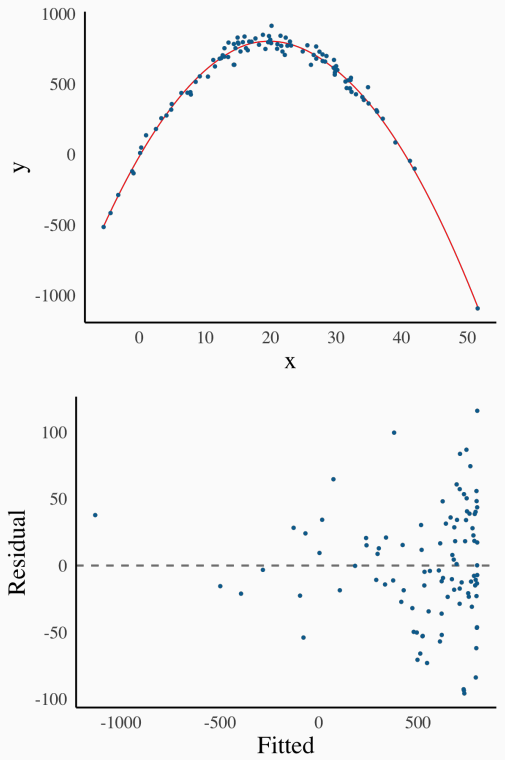
- Sometimes patterns like this can be overcome by transforming a variable.
- If the pattern is actually as clear as this example, you probably need to add a non-linear term.
- Or, as always, it's possible that the issue is a missing variable.

Non-linear Solution

Attempting a fix: You might notice that the shape here is typically associated with a quadratic formula: $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$

So if we add an x^2 term, our model might have a better chance of fitting the data.

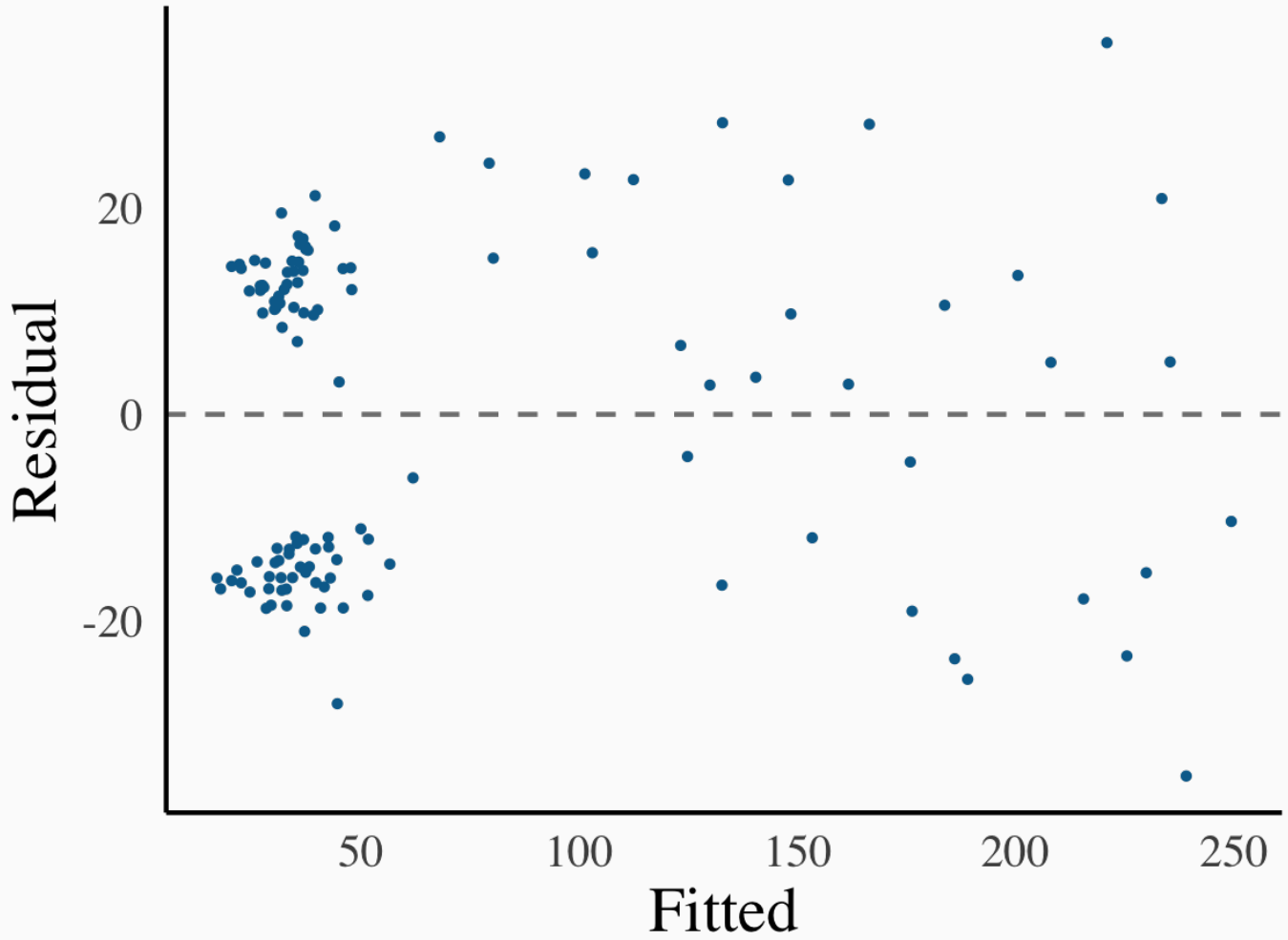
```
MODEL2 <- lm(y ~ x + I(x^2))

Residuals:
   Min     1Q Median     3Q    Max
-88.31 -29.03  -6.62  33.34  95.89

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept)  2.82349   11.09741   0.254      0.8
x           80.22120    1.10150  72.829   <2e-16 ***
I(x^2)      -2.02119    0.02742 -73.706   <2e-16 ***
---
Residual standard error: 44.04 on 97 degrees of freedom
Multiple R-squared:  0.9829,  Adjusted R-squared:  0.9826
F-statistic:  2793 on 2 and 97 DF,  p-value: < 2.2e-16
```



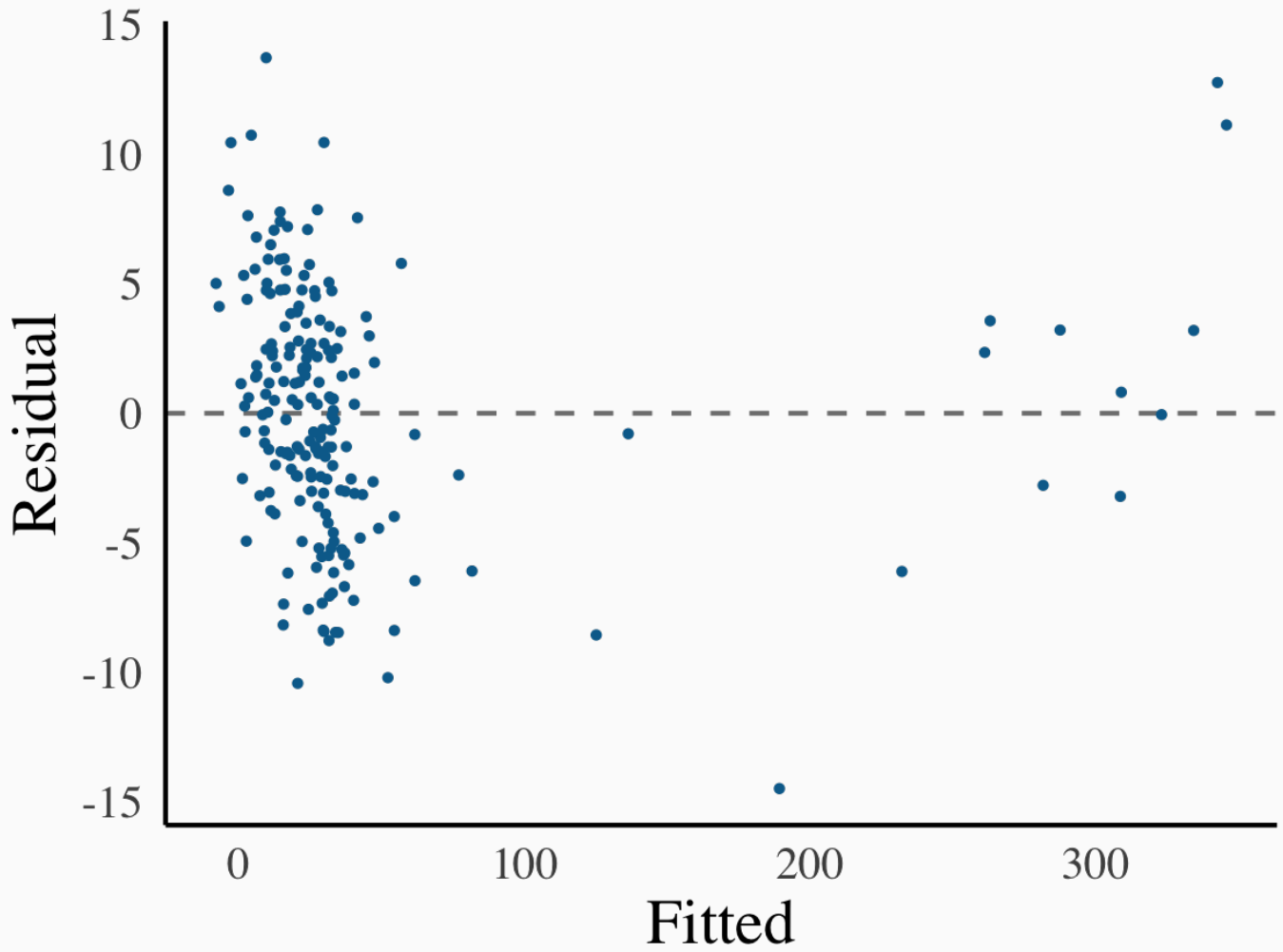
The residuals are slightly unbalanced, but otherwise decent looking.

Diagnosing Residuals in Practice

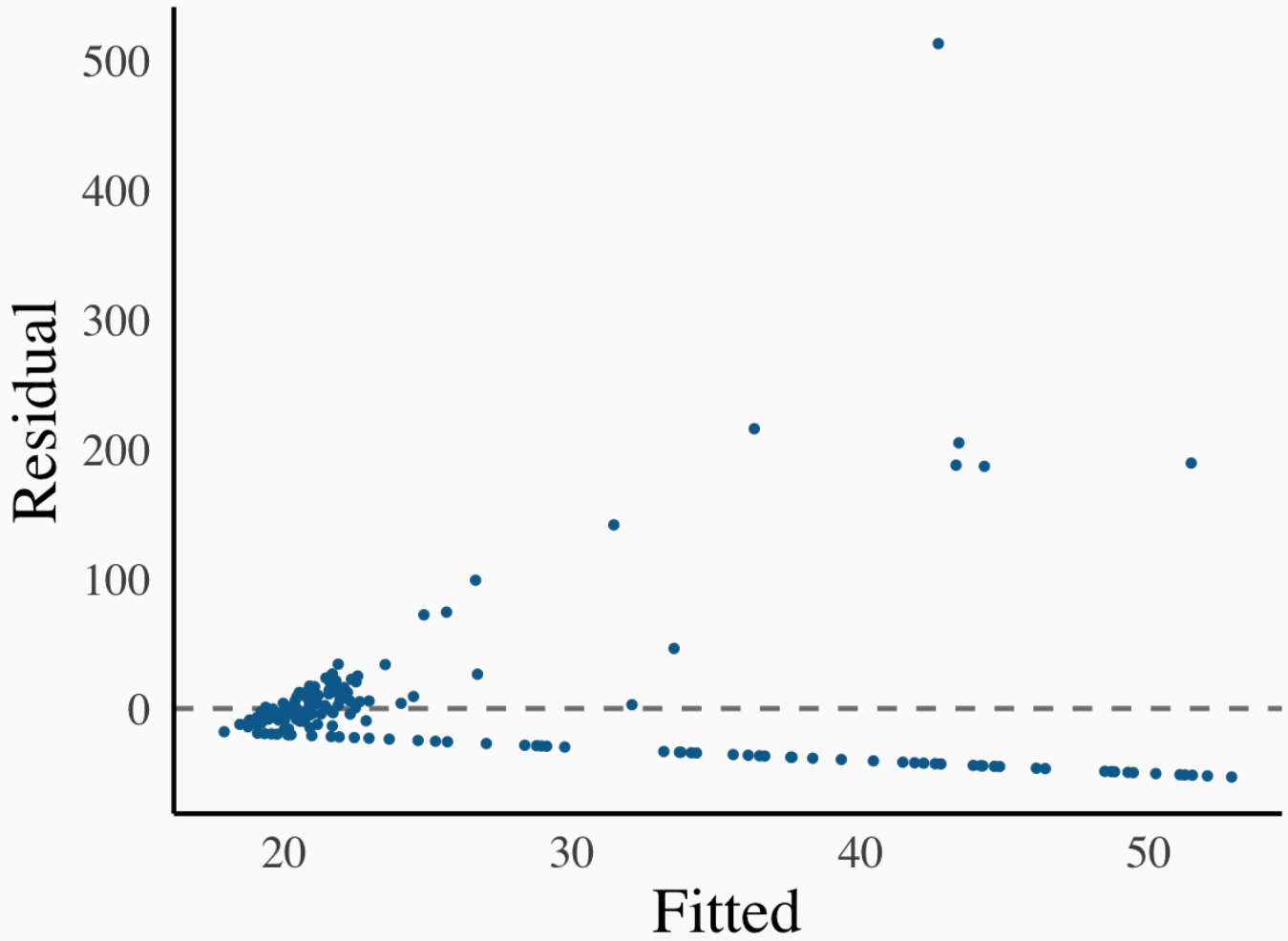
Diagnosing Residuals Ex. 1



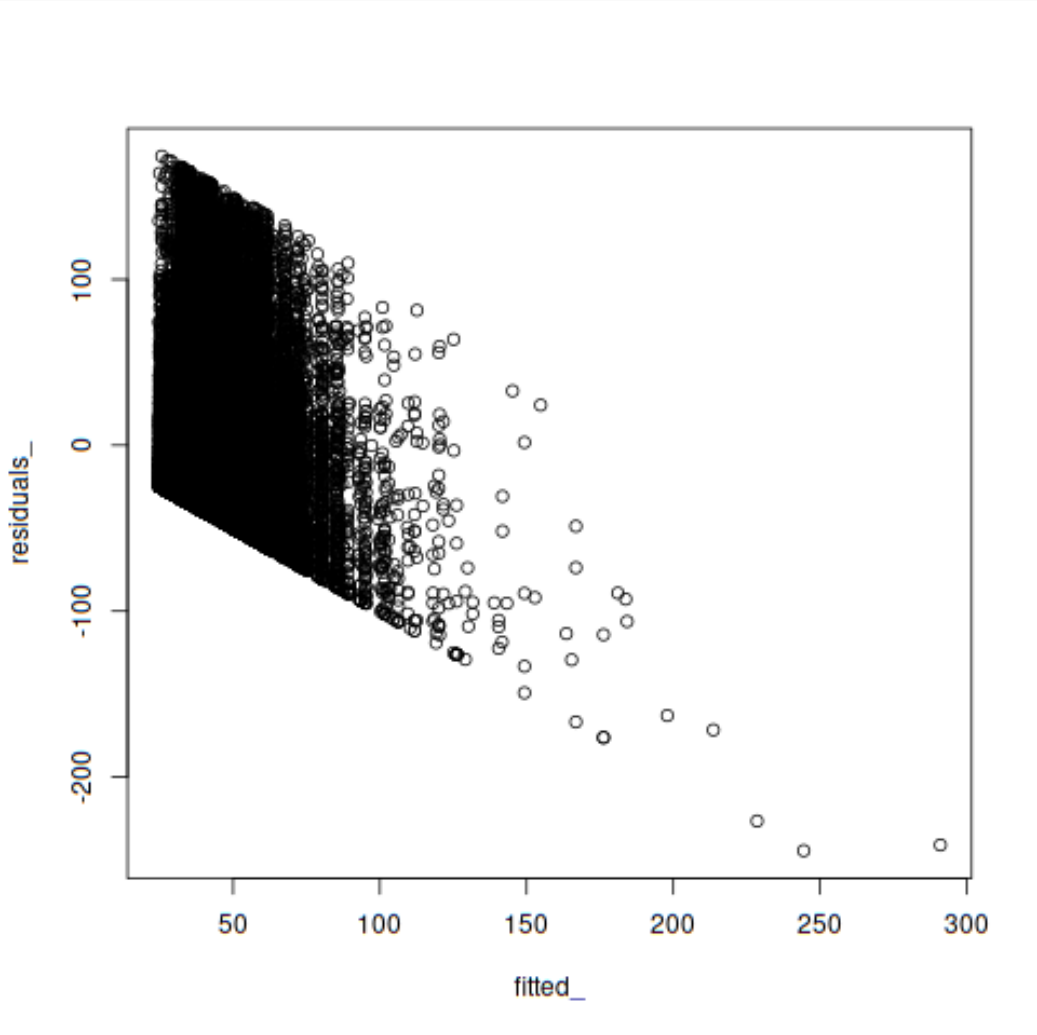
Diagnosing Residuals Ex. 2



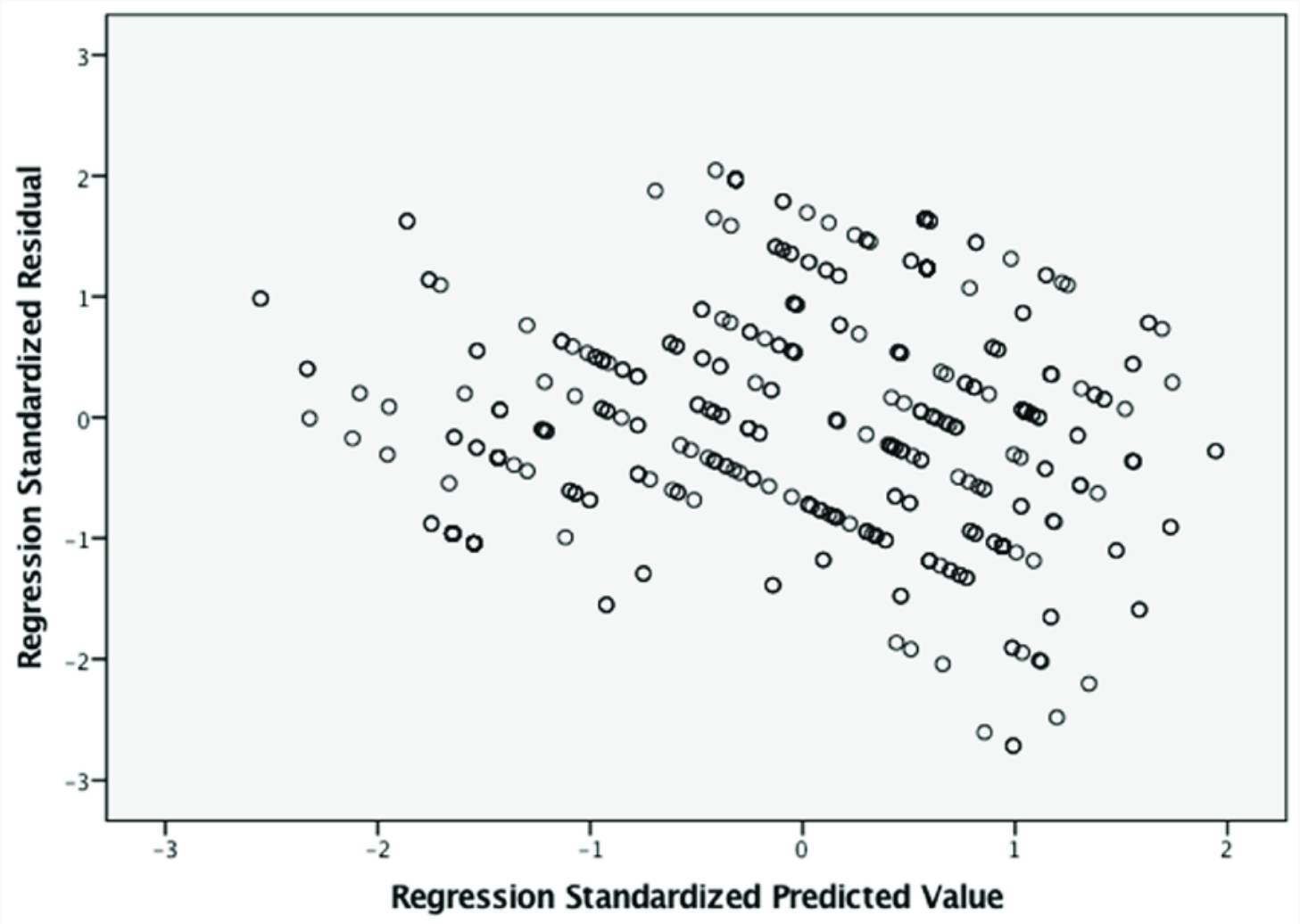
Diagnosing Residuals Ex. 3



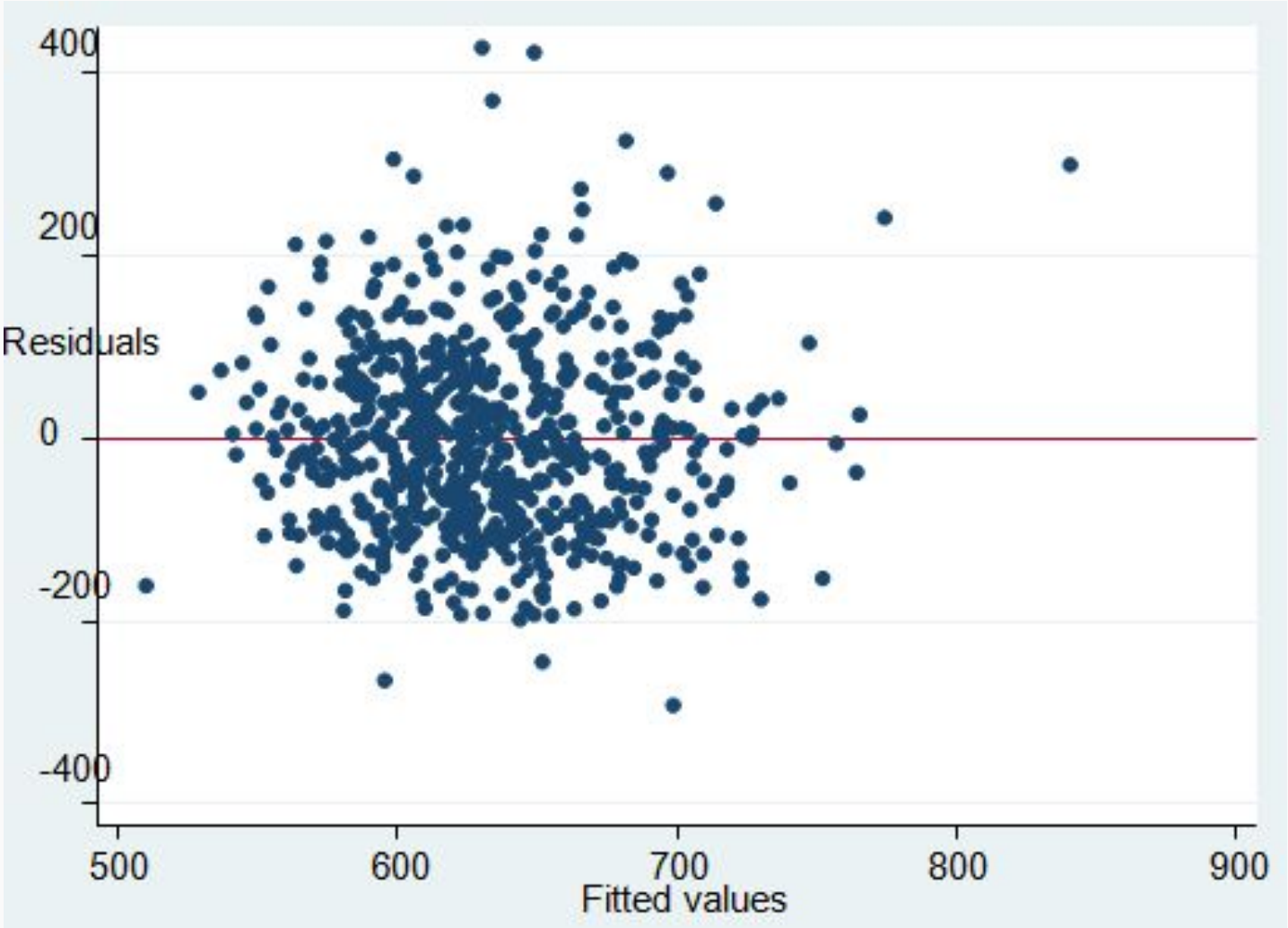
Diagnosing Residuals Ex. 4



Diagnosing Residuals Ex. 5



Diagnosing Residuals Ex. 6



Residuals summary

Residuals are what's left in your data after your model has done its work.

By definition, the residuals of a standard linear regression model should be normally distributed.

Deviations from this expectation provide you with clues on how you might be able to improve the fit of your model.