Detecting and Dealing With Outliers

David Gerard

2018-12-07

Learning Objective

- Sections 11.3 and 11.4
- Detect outliers in multiple linear regression
- Know how to treat outliers in multiple linear regression.

Case Study: Blood Brain Barrier

- A new treatment was proposed to disrupt the blood-brain barrier (to let drugs enter the brain).
- Rats were induced to have brain tumors.
- Rats were randomized to receive either the treatment or a control.
- Response: ratio of drug concentration in brain to drug concentration in liver.
- Designed explanatory variables: sacrifice time, treatment/control.
- They measured other variables that might influence the response: days post inoculation, tumor weight, initial weight, weight loss.

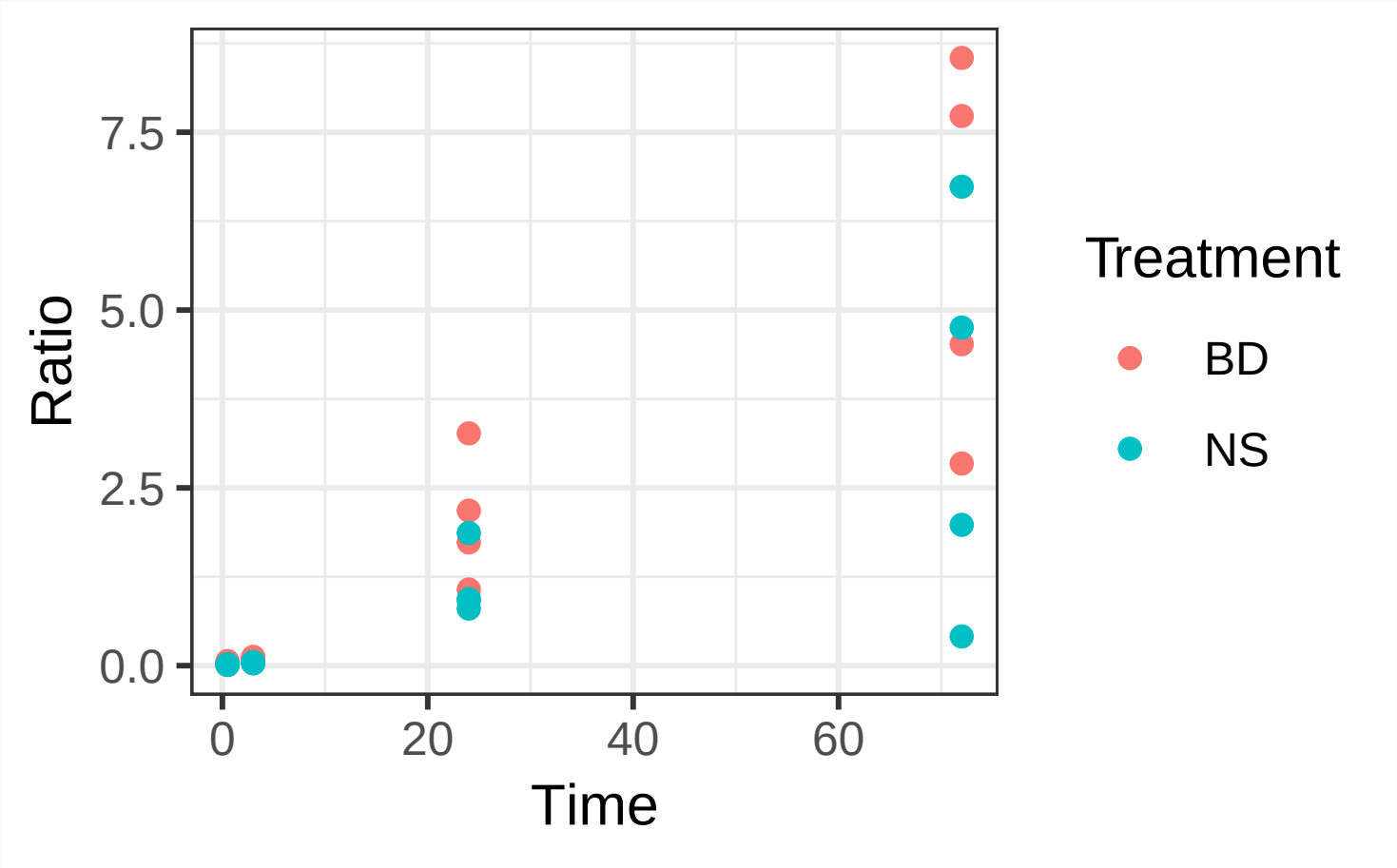
Case Study: Blood Brain Barrier

```
library(Sleuth3)
data(case1102)
case1102$Ratio <- case1102$Brain / case1102$Liver
head(case1102)
```

```
##    Brain   Liver Time Treatment Days    Sex Weight Loss
## 1  41081 1456164  0.5        BD   10 Female    239  5.9
## 2  44286 1602171  0.5        BD   10 Female    225  4.0
## 3 102926 1601936  0.5        BD   10 Female    224 -4.9
## 4  25927 1776411  0.5        BD   10 Female    184  9.8
## 5  42643 1351184  0.5        BD   10 Female    250  6.0
## 6  31342 1790863  0.5        NS   10 Female    196  7.7
```

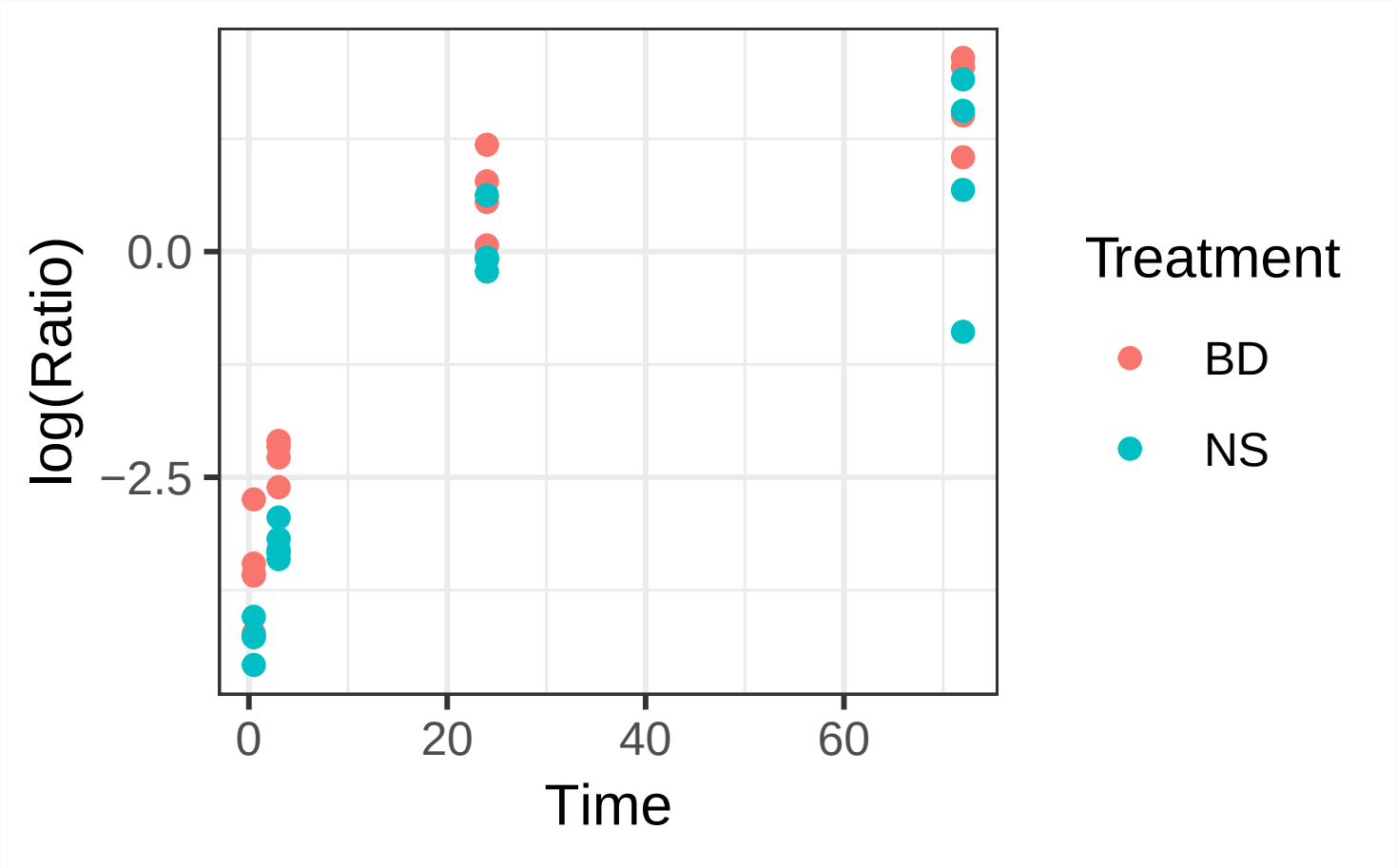
Step 1: Make a lot of scatterplots

```
qplot(Time, Ratio, color = Treatment,
      data = case1102)
```



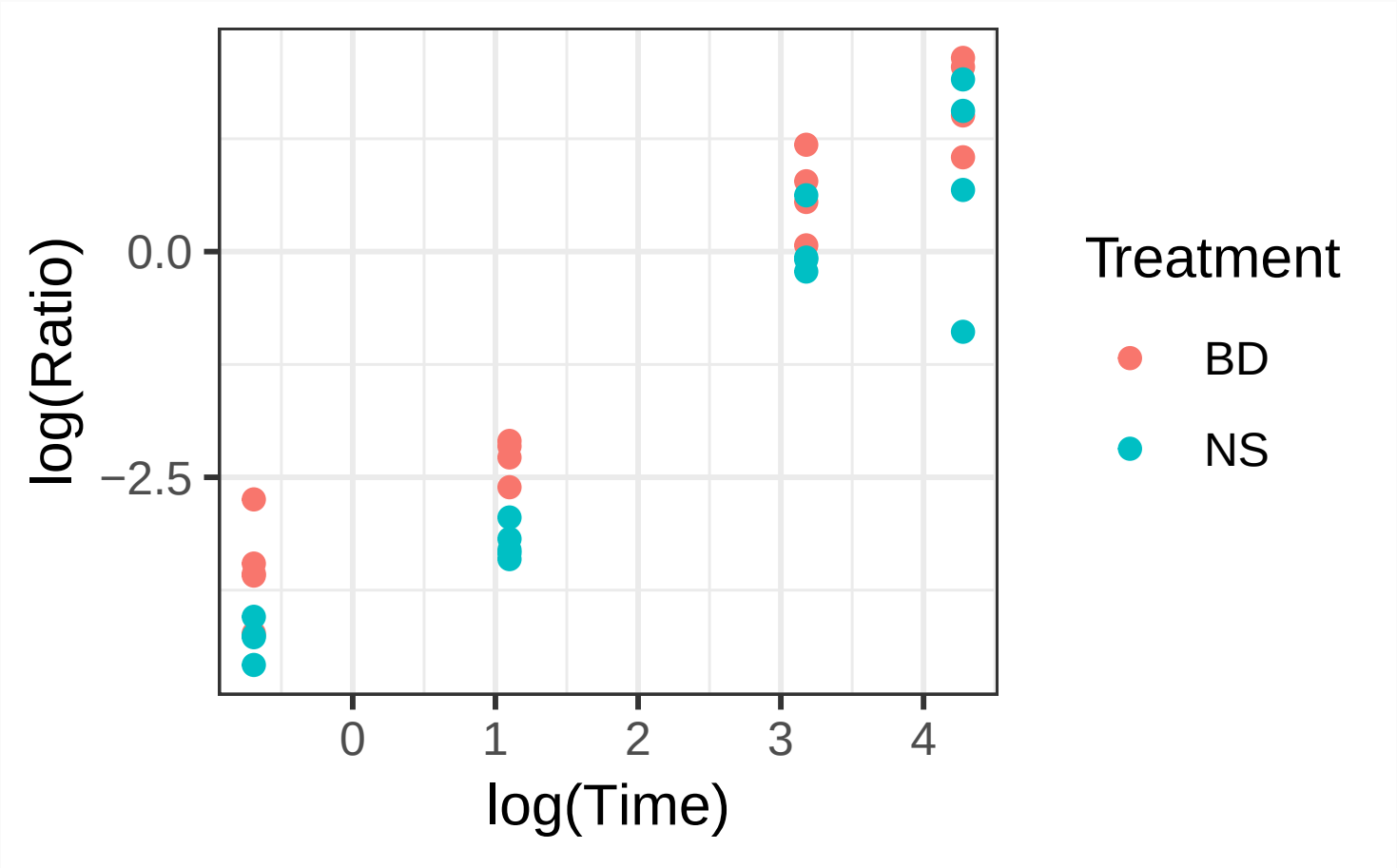
Step 1: Make a lot of scatterplots

```
qplot(Time, log(Ratio), color = Treatment,
      data = case1102)
```



Step 1: Make a lot of scatterplots

```
qplot(log(Time), log(Ratio), color = Treatment,
      data = case1102)
```

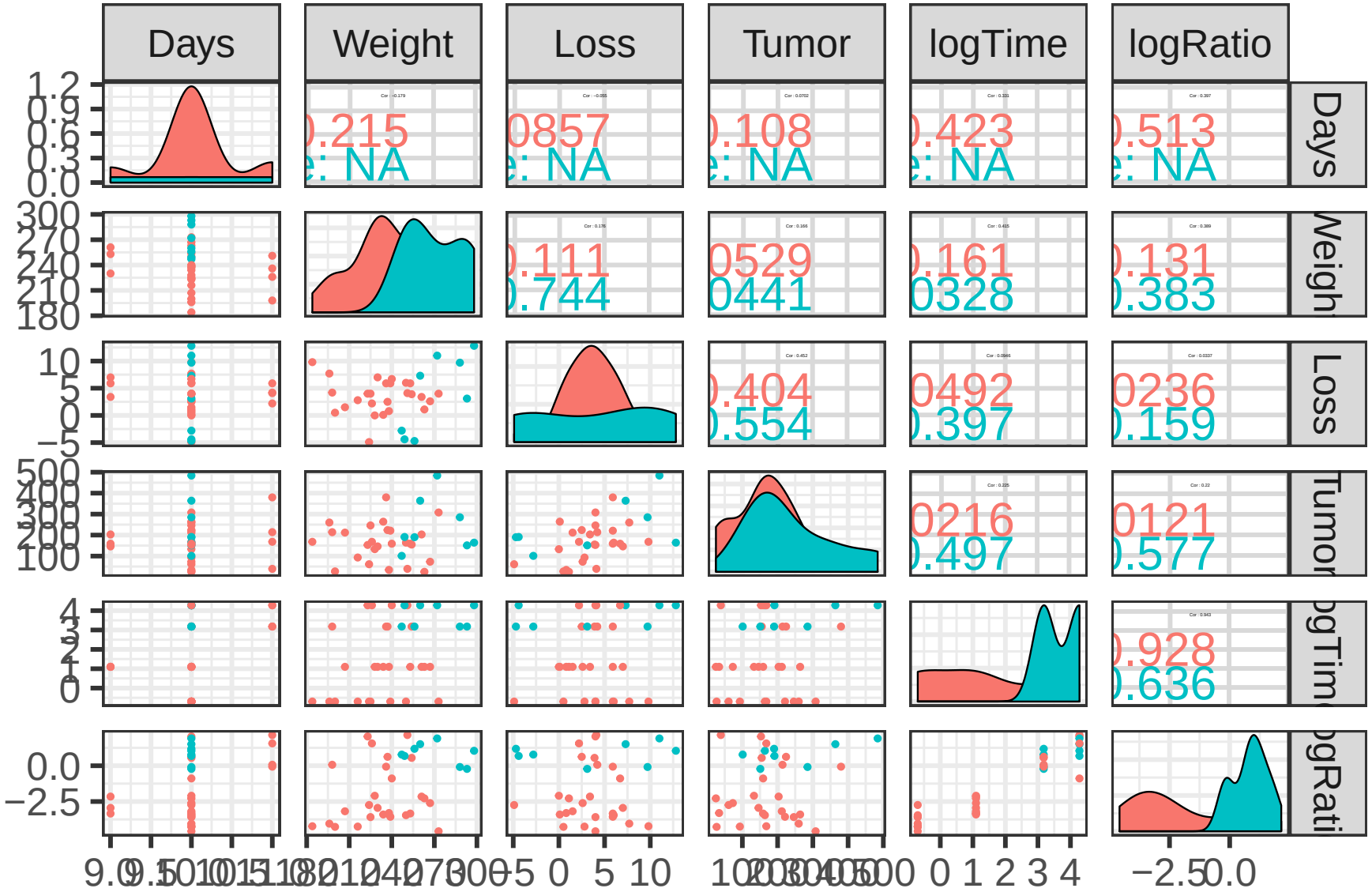


Step 1: Make a lot of scatterplots

```
case1102$logTime  <- log(case1102$Time)
case1102$logRatio <- log(case1102$Ratio)
```

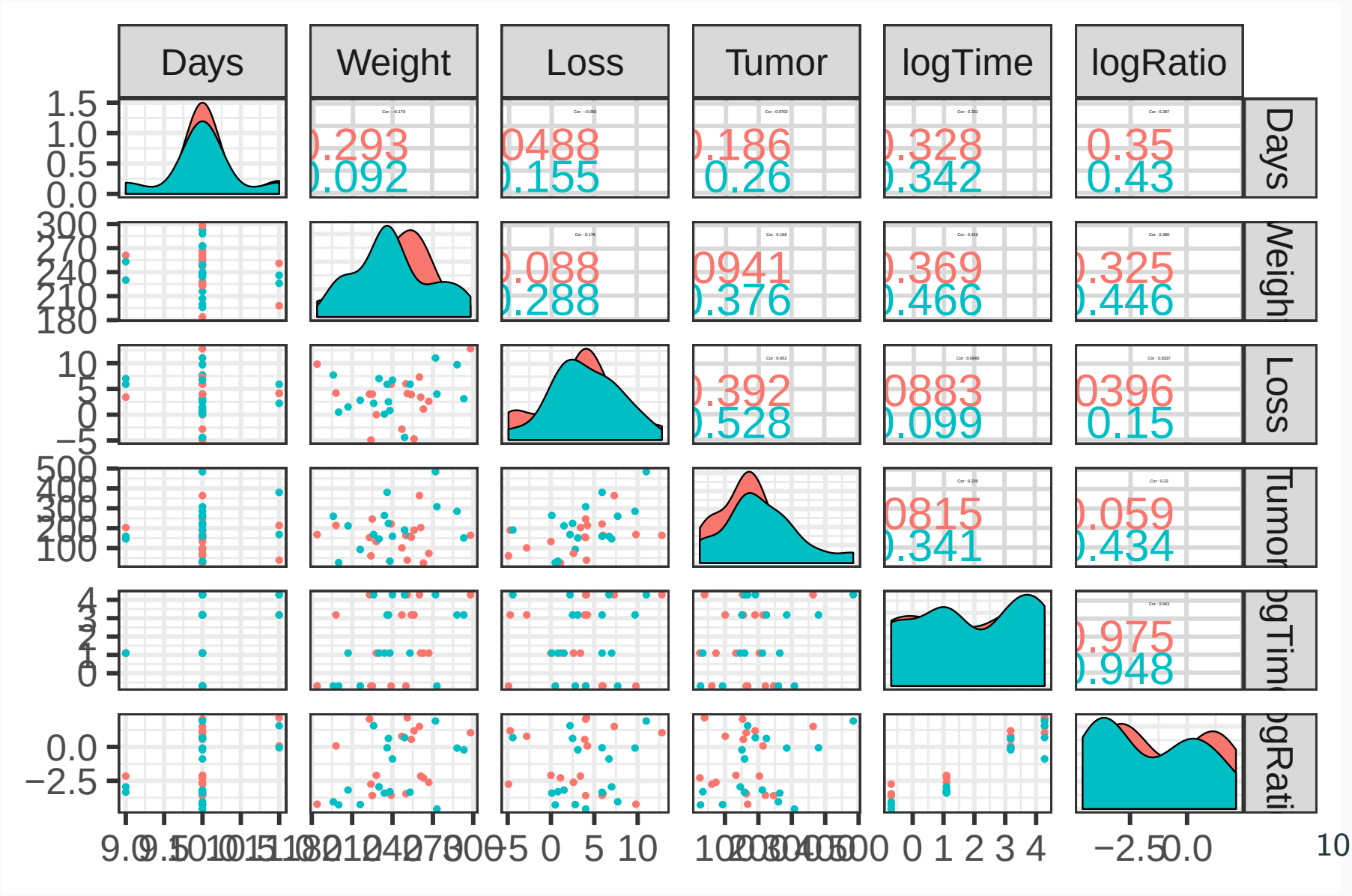
Step 1: Make a lot of scatterplots (color coded by Sex)

```
library(GGally)
ggpairs(case1102, columns = c(5, 7, 8, 9, 11, 12), aes(size = I(0.2), color = Sex))
```



Step 1: Make a lot of scatterplots (color coded by Treatment)

```
ggpairs(case1102, columns = c(5, 7, 8, 9, 11, 12), aes(size = I(0.2), color = Treatment))
```



Conclusions

- `logRatio` is associated with `logTime`, `Sex`, `Treatment`, `Weight`, and `Days`.
- No other transformations are needed.

Step 2: Fit a rich model and check residuals.

- As an initial fit for a rich model, we'll include all explanatory variables and use `Time` as a factor (categorical variable).

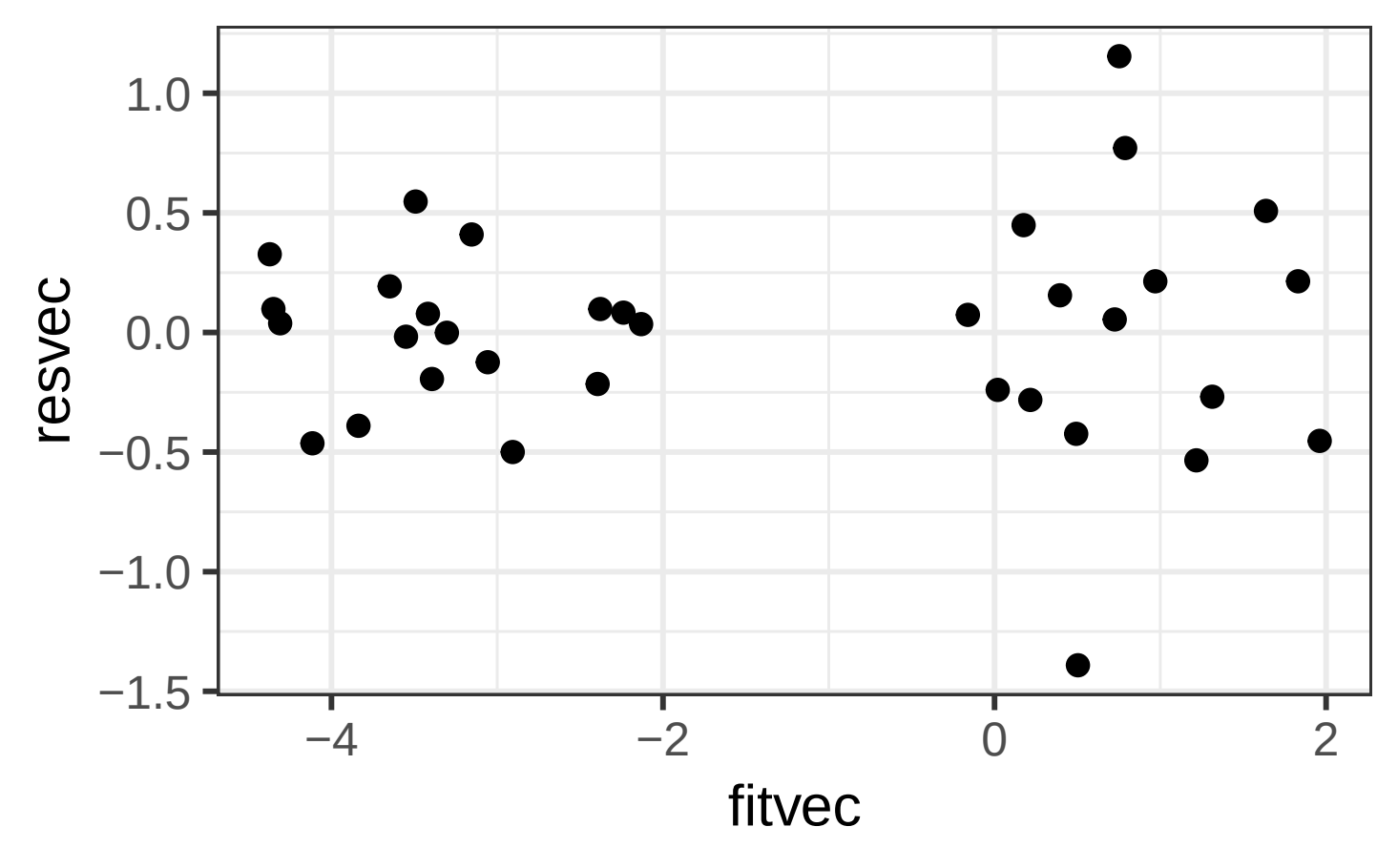
$$\mu(logRatio|logTime, Treatment, Days, Sex, Weight, Loss, Tumor)$$
$$= logTime + Treatment + logtime \times Treatment + Days + Sex$$
$$+ Weight + Loss + Tumor$$

```
lmrich <- lm(logRatio ~ as.factor(logTime) * Treatment +
                 Sex + Weight + Loss + Tumor,
               data = case1102)

resvec <- resid(lmrich)
fitvec <- fitted(lmrich)
```

Step 2: Fit a rich model and check residuals.

```
qplot(fitvec, resvec)
```



Step 2: Conclusions

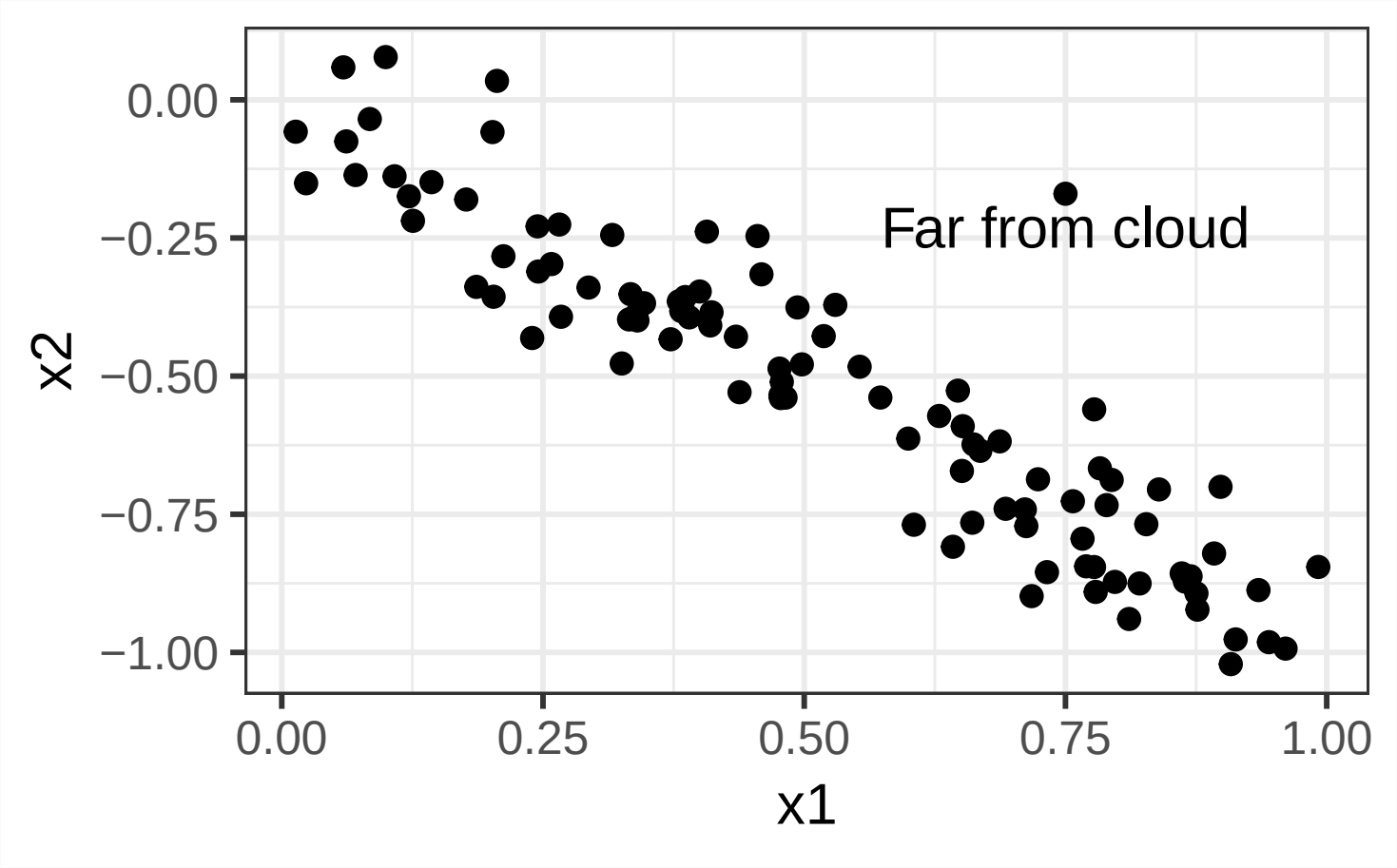
- Notice two points that are possibly outlying observations.
- Move on to formal evaluations of influence.

Step 3: Case-Influence Statistics

- Case-Influence Statistics: Numerical Measures of how influential a single observation is to the linear regression.

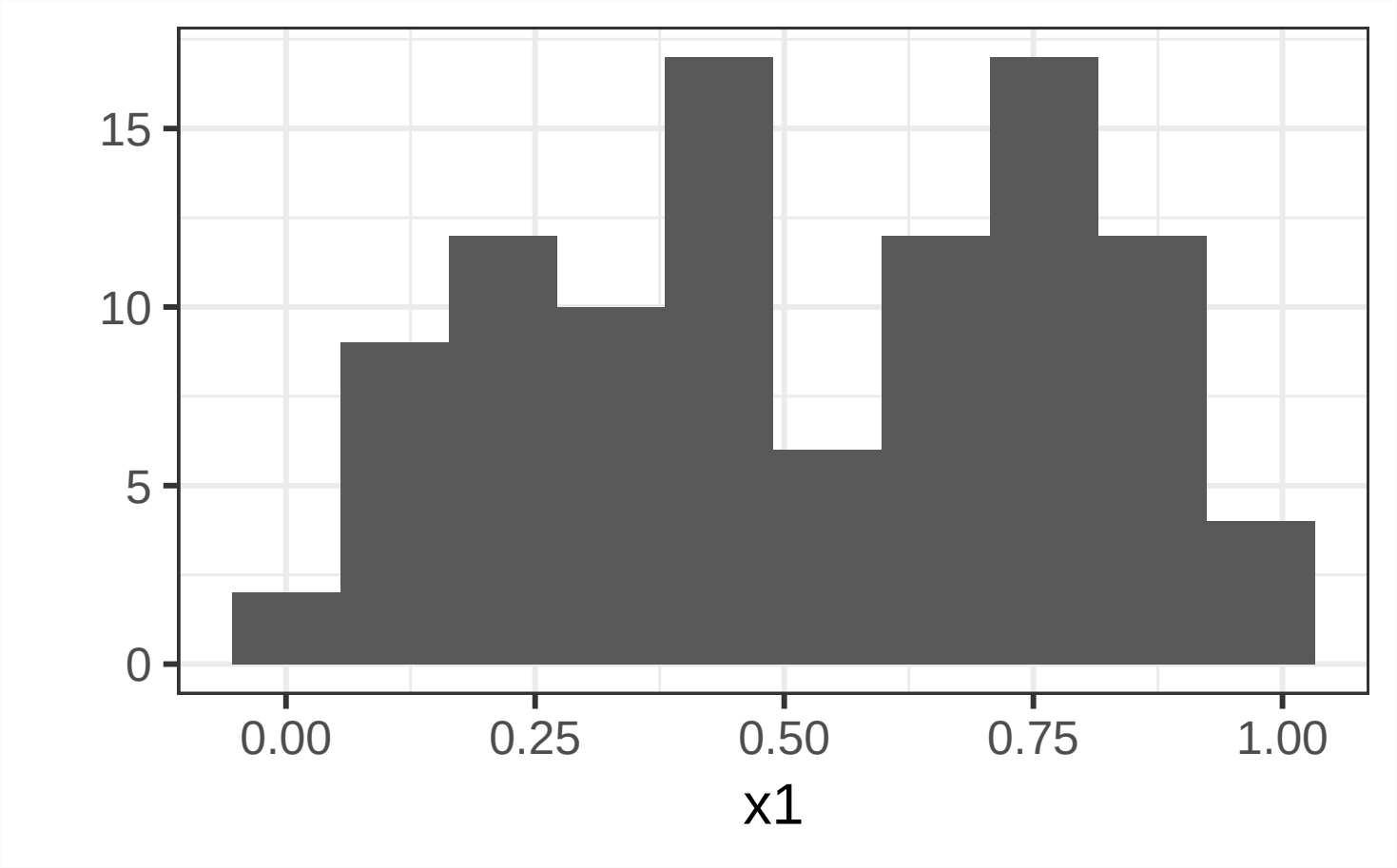
Step 3: Leverage

- Leverage: how far away an observational units explanatory variables are from the rest of the group.



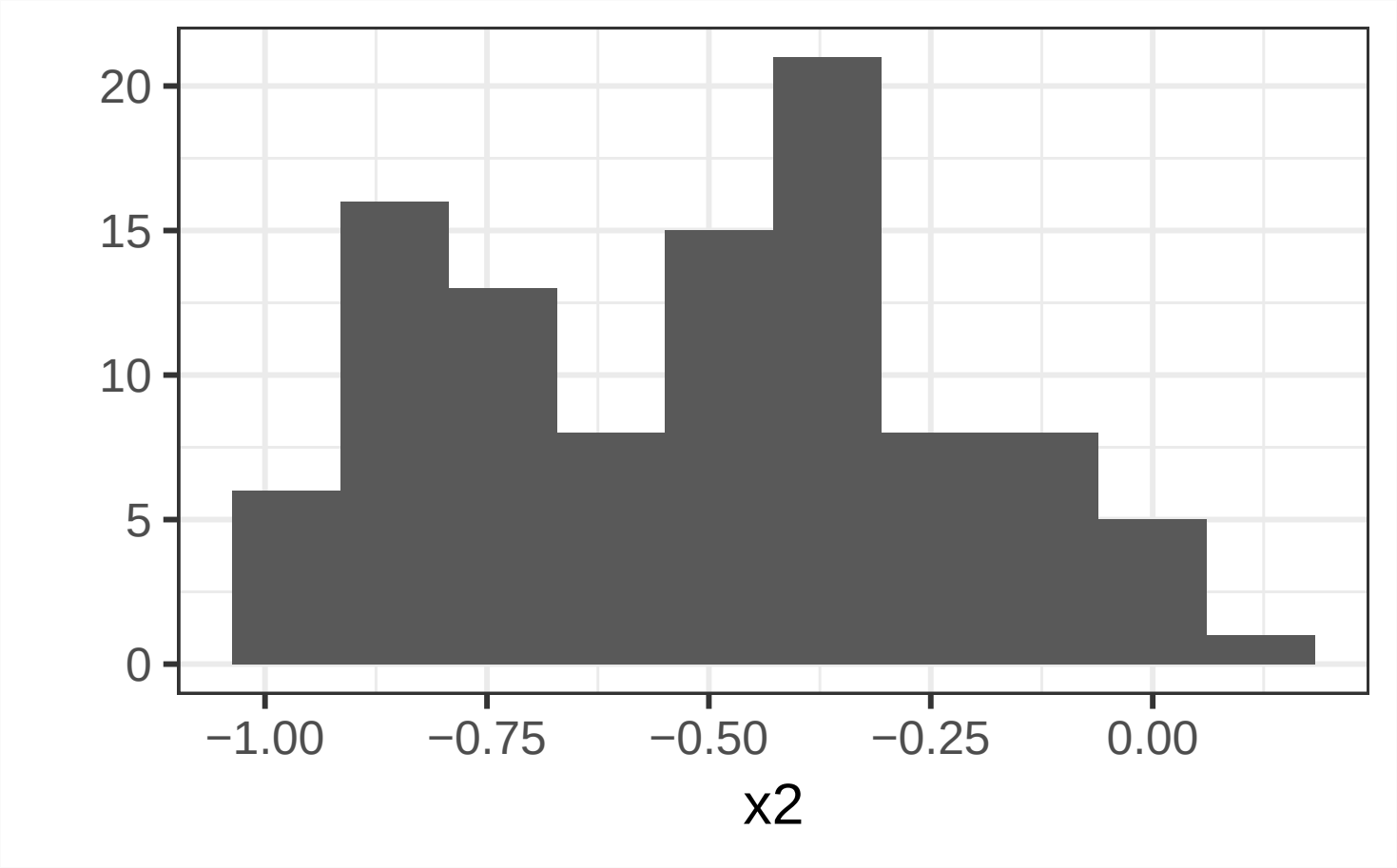
Step 3: Leverage

- Histograms of both variables don't show that this is an extreme point



Step 3: Leverage

- Histograms of both variables don't show that this is an extreme point

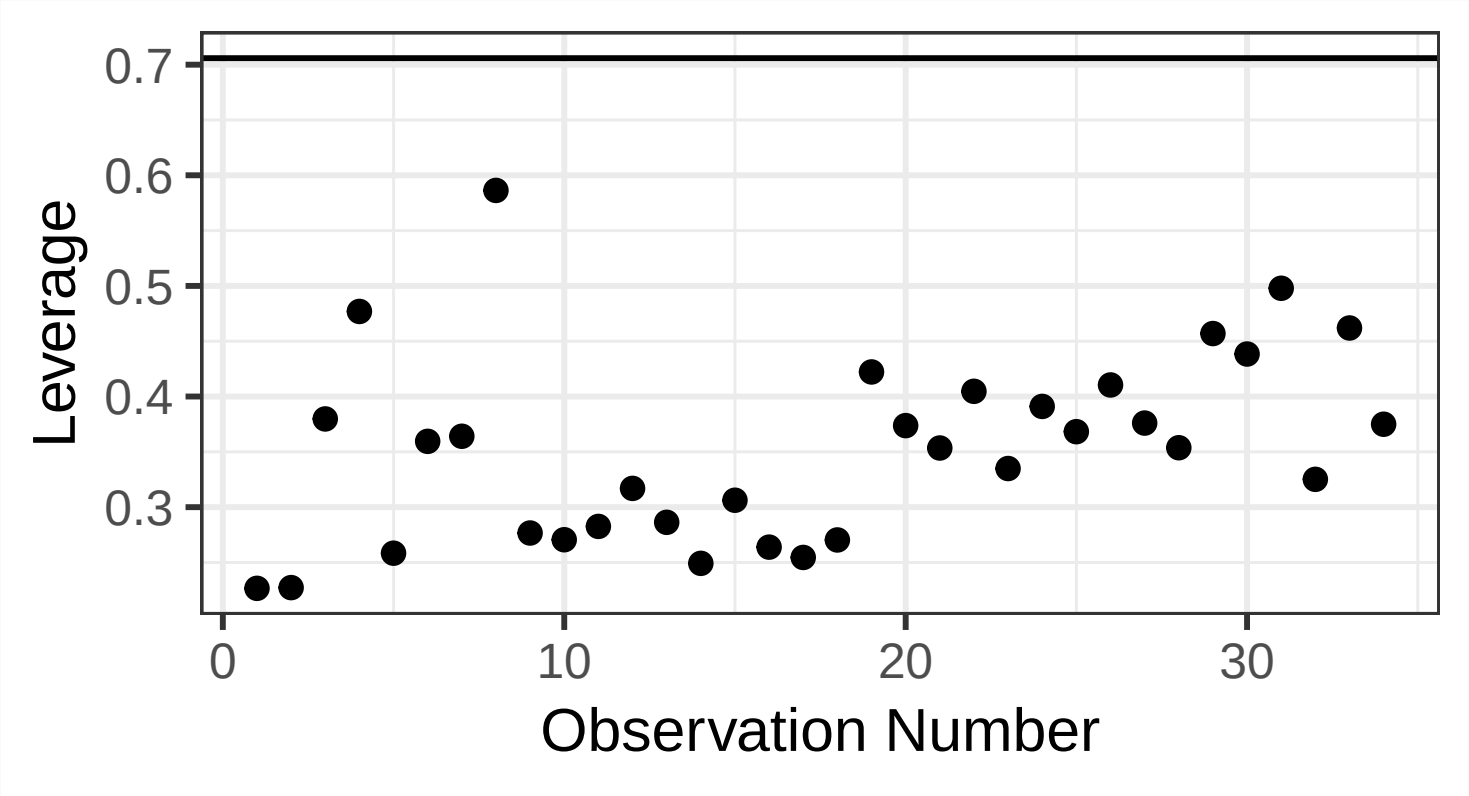


Step 3: Leverage

- It is even more difficult when you have more than two explanatory variables.
- Leverage measures how far away from the cloud of points an observation is.
- Always greater than 0.
- Typical leverage values are around p/n, where $p =$ number of parameters and $n =$ number of observations.
- Rule of thumb: Points with leverage values over $2p/n$ have high leverage.

Step 3: Leverage in R

```
n <- nrow(case1102)
p <- n - df.residual(lmrich)
lev_vec <- hatvalues(lmrich) ## gets leverage
qplot(x = seq_along(lev_vec), lev_vec, geom = "point") +
  geom_hline(yintercept = 2 * p / n) +
  xlab("Observation Number") + ylab("Leverage")
```

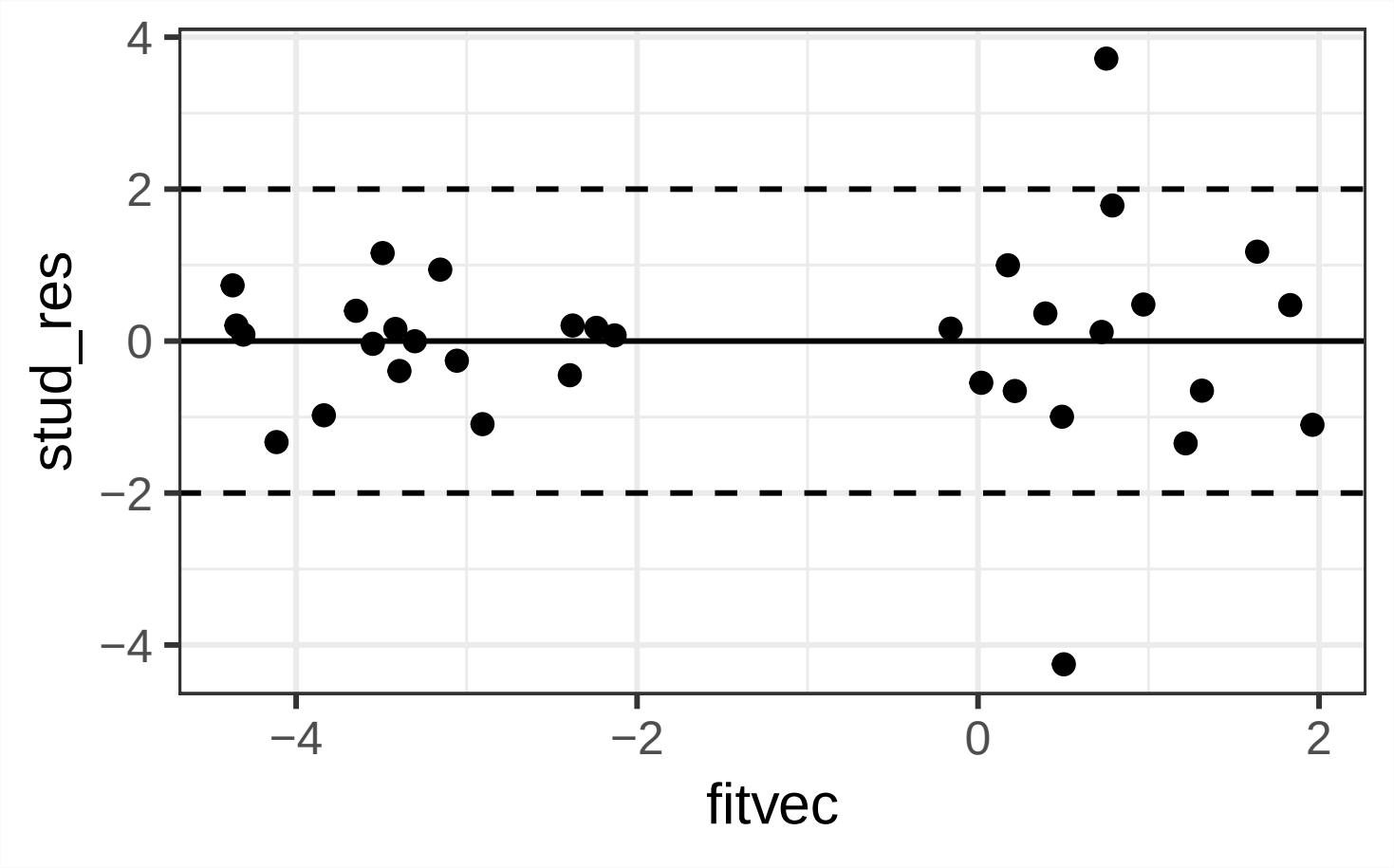


Step 3: Studentized Residuals

- Residuals are expected to have different variances depending on how far away from the cloud they are.
- Studentized residuals calculate the number of standard deviations away from 0 a residual is.
- Expect about 95% of residuals to fall inside of 2 standard deviations.
- But expect 5% to fall outside of 2 standard deviations.

Step 3: Studentized Residuals in R

```
stud_res <- rstudent(lmrich)
qplot(fitvec, stud_res) + geom_hline(yintercept = 0) +
  geom_hline(yintercept = 2, lty = 2) +
  geom_hline(yintercept = -2, lty = 2)
```

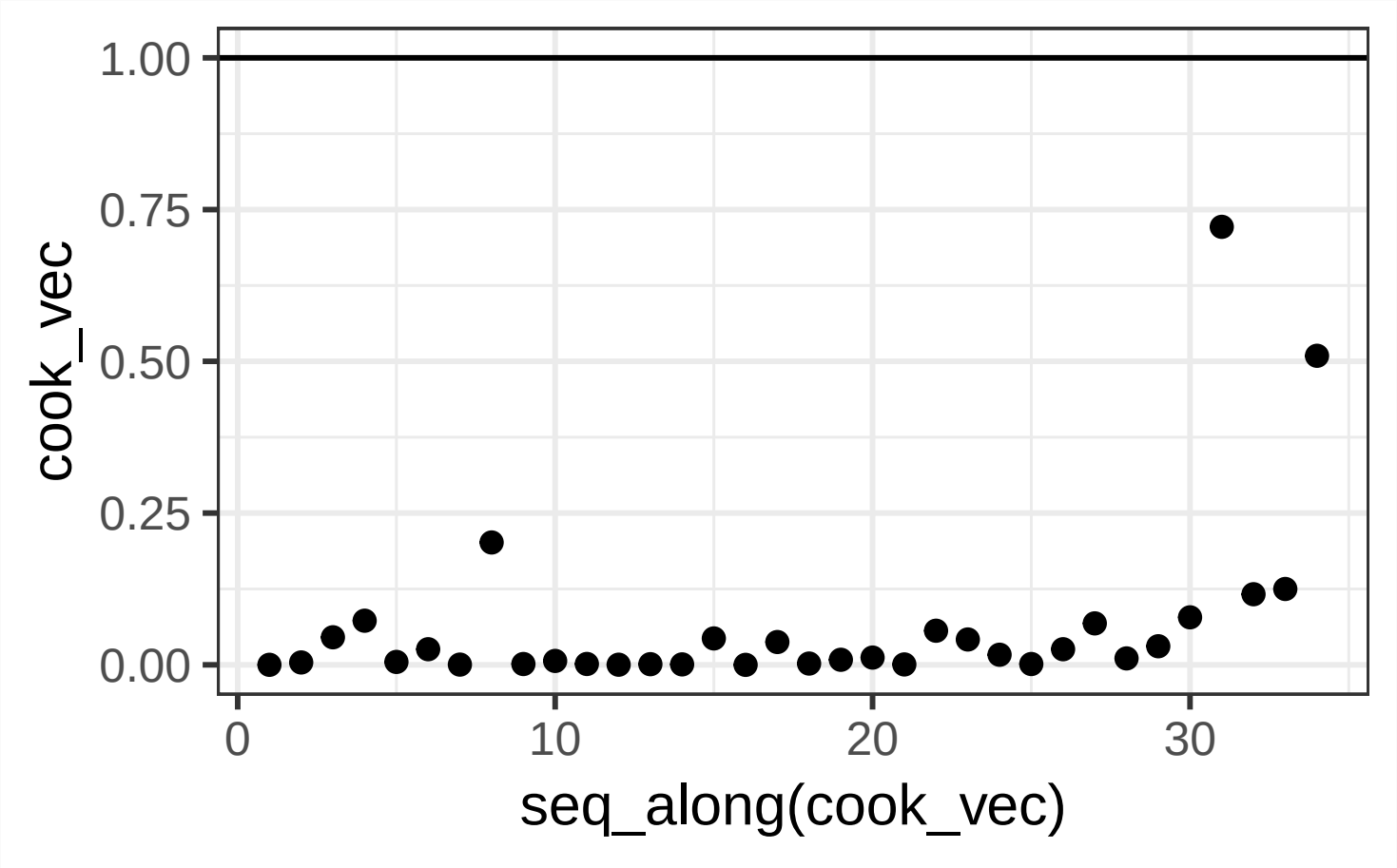


Step 3: Cook's Distance

- Cooks distance: Measures overall influence of an observational unit.
- Idea: Refit regression wihout observatioanl unit, see how much the fits change. Average over all observational units.
- Cooks distance is always greater than 0.
- Rule of thumb: If cooks distance is greater than 1, then this indicates a large influence.

Step3: Cook's Distance in R

```
cook_vec <- cooks.distance(lmrich)
qplot(x = seq_along(cook_vec), cook_vec, geom = "point") +
  geom_hline(yintercept = 1)
```



Step 3: Conclusions

- The influential plots seems to indicate no major influential points.

Step 4: What if have outliers?

- We can check the values of the extreme observations.
- If they have weird X values, then we can omit them from the data and state that the scope of inference is only valid for a subset of X values.
- E.g. if all mice have weight less than 300 g, but we have one that is 350, then we can remove that mouse and state that our results are only for mice less than 300 g.

Step 4: What if have outliers?

- To be safe, we can fit our finalized model (not the rich) both with and without the outliers.
  - If results don't change, keep the outliers.
  - If results change, report both results, or try a more robust method.