IS4 in R: Regression Wisdom (Chapter 8)

Patrick Frenett, Vickei Ip, and Nicholas Horton (nhorton@amherst.edu)

June 19, 2018

Introduction and background

This document is intended to help describe how to undertake analyses introduced as examples in the Fourth Edition of *Intro Stats* (2013) by De Veaux, Velleman, and Bock. More information about the book can be found at http://wps.aw.com/aw_deveaux_stats_series. This file as well as the associated R Markdown reproducible analysis source file used to create it can be found at https://nhorton.people.amherst.edu/is4.

This work leverages initiatives undertaken by Project MOSAIC (http://www.mosaic-web.org), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignettes (http://cran.r-project.org/web/packages/mosaic). A paper describing the mosaic approach was published in the *R Journal*: https://journal.r-project.org/archive/2017/RJ-2017-024.

Note that some of the figures in this document may differ slightly from those in the IS4 book due to small differences in datasets. However in all cases the analysis and techniques in R are accurate.

Chapter 8: Regression Wisdom

Section 8.1: Examining residuals

Figure 8.1 (page 215) displays the scatterplot of heart rate vs duration for the Penguins dataset (along with a superimposed regression line and a smoother).

```r
library(mosaic)
library(readr)
options(digits = 3)
Penguins <- read_csv("https://nhorton.people.amherst.edu/sdm4/data/Penguins.csv")
gf_point(DiveHeartRate ~ Duration, data = Penguins) %>%
  gf_labs(x = "Duration (mins)", y = "Dive Heart Rate (bpm)") %>%
  gf_lm() %>%
  gf_smooth(col = "red", se = FALSE)
```

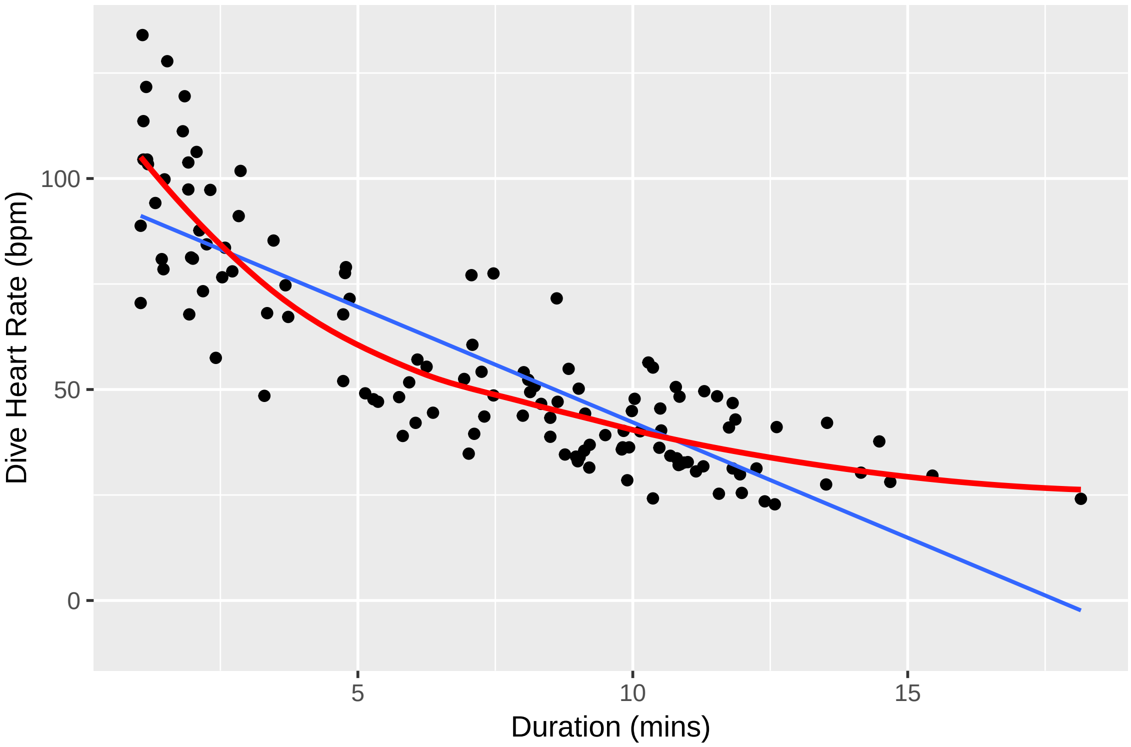


Figure 8.2 (page 215) displays the residuals from a linear regression model as a function of duration.

```
Penguinmod <- lm(DiveHeartRate ~ Duration, data = Penguins)
gf_point(resid(Penguinmod) ~ Duration, data = Penguins) %>%
  gf_smooth(col = "darkslategray", se = FALSE)
```

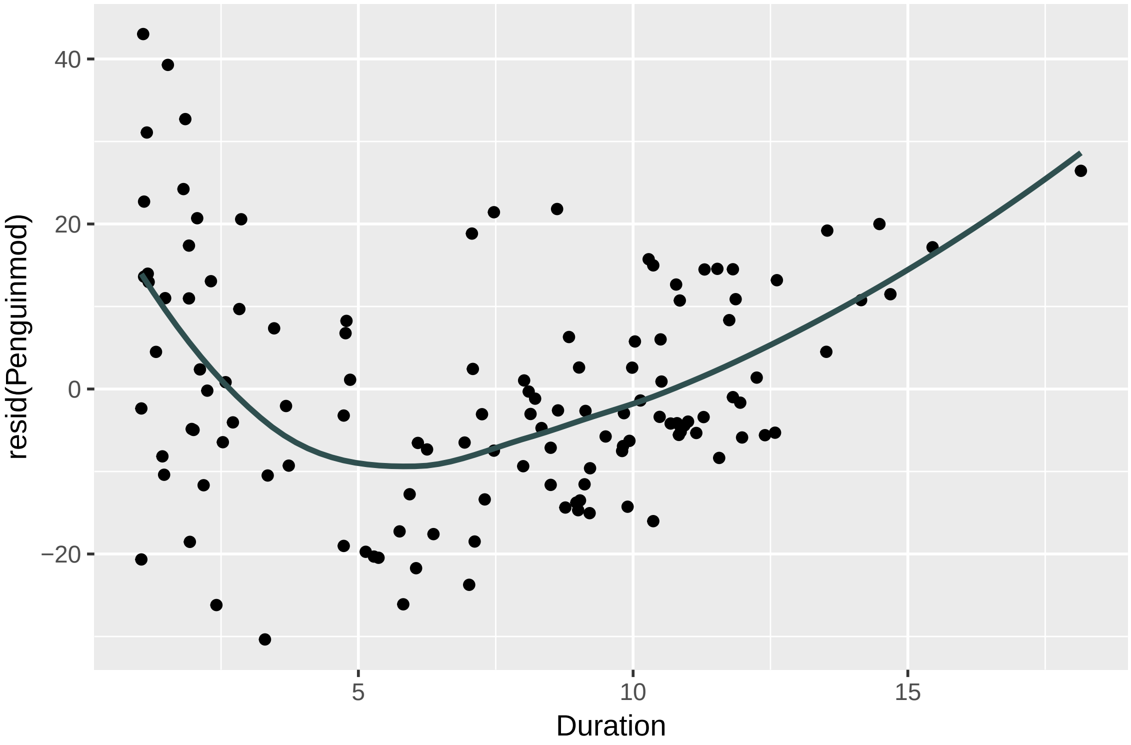


Figure 8.3 (page 216) displays the histogram of residuals for the cereal data from Chapter 7.

```
Cereals <- read_csv("https://nhorton.people.amherst.edu/sdm4/data/Cereals.csv")
Cerealmod <- lm(calories ~ sugars, data = Cereals)
gf_histogram(..density.. ~ resid(Cerealmod), binwidth = 7.5, center = 3.75,
             fill = "darkseagreen3",  col = TRUE) %>%
  gf_labs(y = "Density")
```

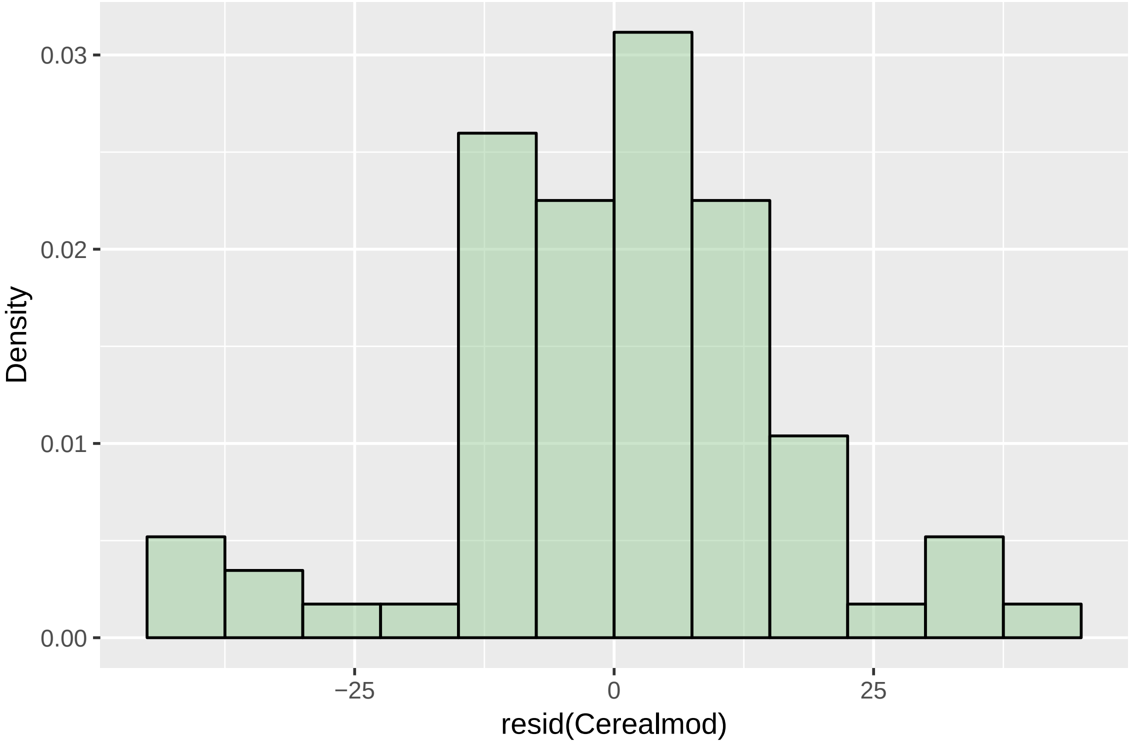
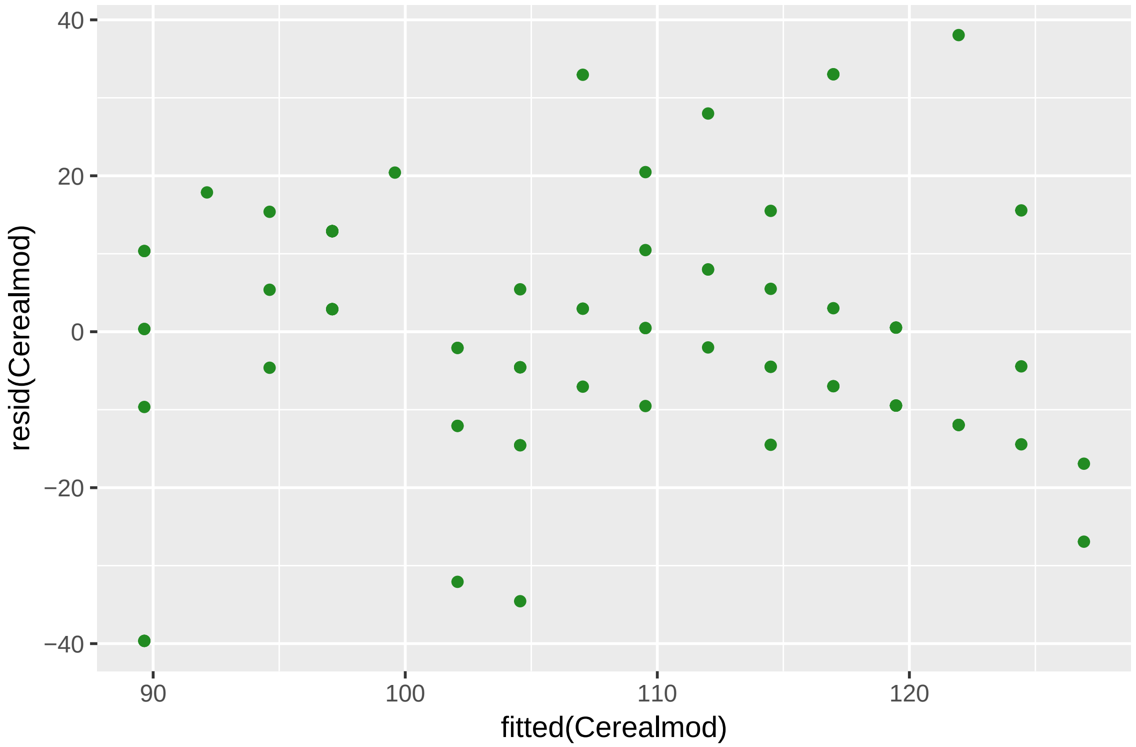


Figure 8.4 (page 216) displays a scatterplot of the residuals vs predicted values. Without jittering, the display has an odd pattern.

```
gf_point(resid(Cerealmod) ~ fitted(Cerealmod), col = "forestgreen")
```



By adding some random noise we can more easily observe values that are shared by more than one cereal.

```
gf_jitter(resid(Cerealmod) ~ fitted(Cerealmod), alpha = 0.5, width = 0.2,
          height = 0, col="darkslateblue")
```

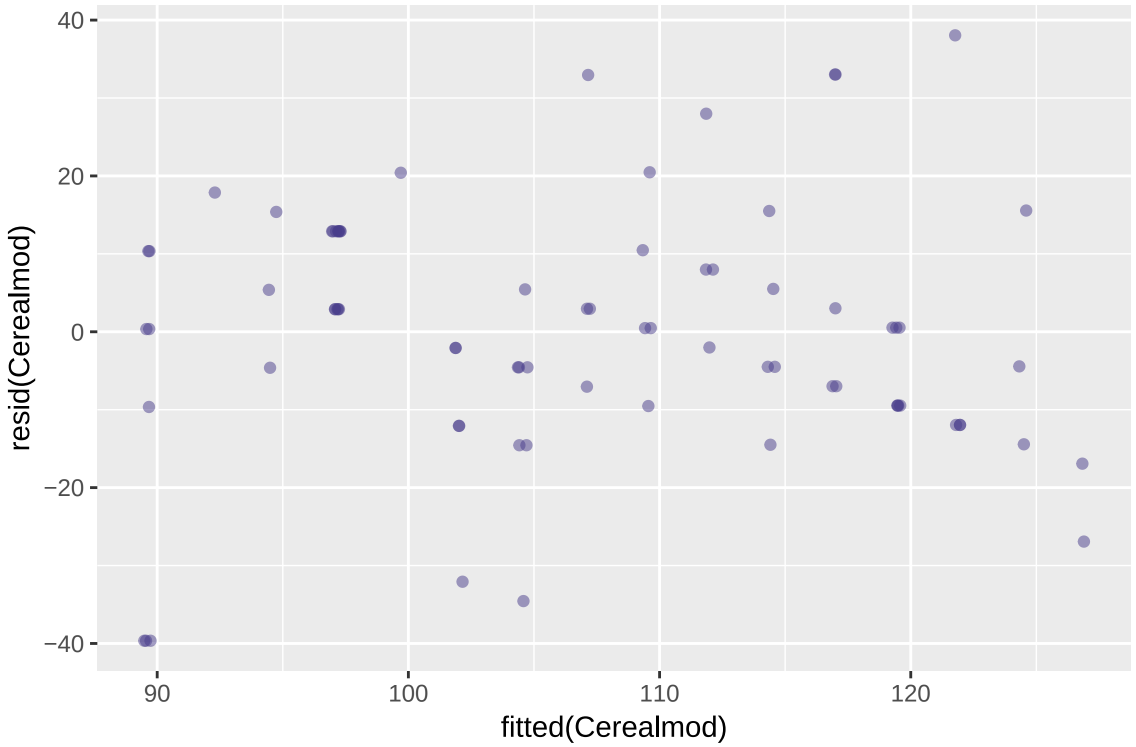


Figure 8.5 (page 217) displays the scatterplot stratified by what shelf it is displayed on at the store.

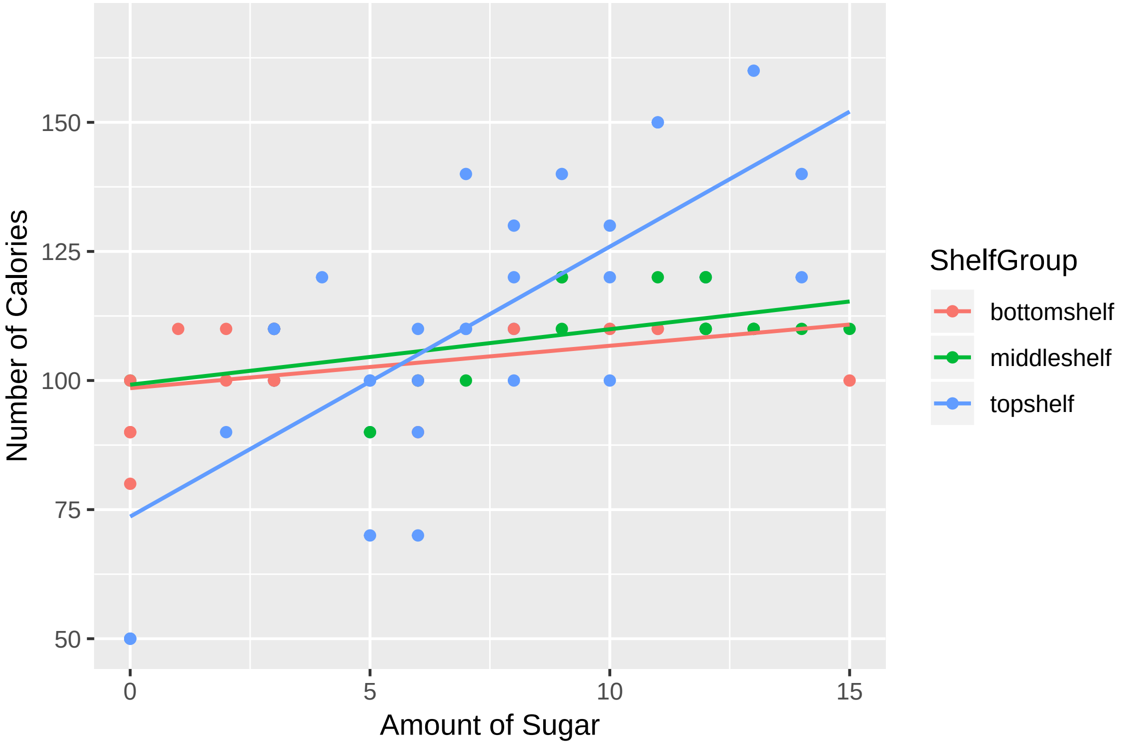
```
tally(~ shelf, data = Cereals)
```

```
## shelf
##  1  2  3
## 20 21 36
```

```
Cereals <- mutate(Cereals, ShelfGroup = derivedFactor(
  bottomshelf = shelf == 1,
  middleshelf = shelf == 2,
  topshelf = shelf == 3
))
tally(~ ShelfGroup, data = Cereals)
```

```
## ShelfGroup
## bottomshelf middleshelf    topshelf
##          20          21          36
```

```
gf_point(calories ~ sugars, group = ~ ShelfGroup, col = ~ ShelfGroup,
         data = Cereals, xlab = "Amount of Sugar", ylab = "Number of Calories") %>%
  gf_lm() %>%
  gf_labs(x = "Amount of Sugar", y = "Number of Calories")
```



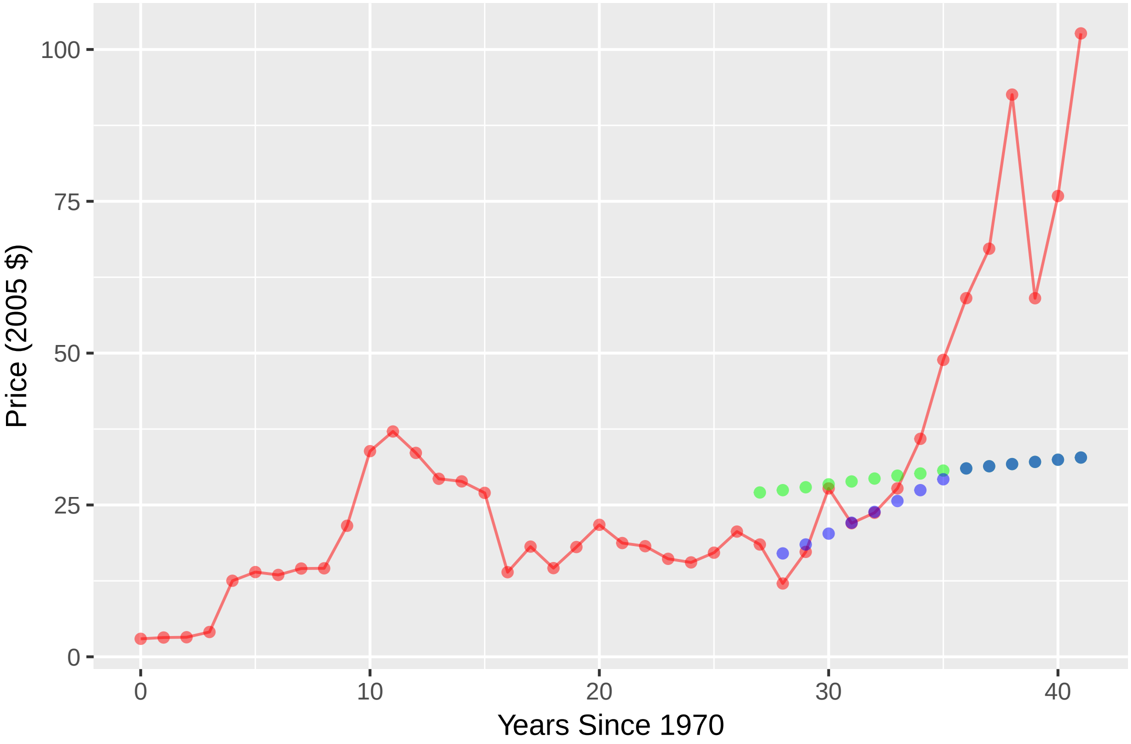
Section 8.2: Extrapolation and reaching beyond the data

The `plot` function from base R will give a line graph if it's type is specified as "b" for both points and lines. `?help` will give the full list of types of graphs `plot` can generate. Below is figure 8.8 on page 219.

```
Oil <- read_csv("http://www3.amherst.edu/~nhorton/sdm4/data/Historical_Oil_Prices_2014.csv")
```

```
## Parsed with column specification:
## cols(
##   Year = col_integer(),
##   `Years since 1970` = col_integer(),
##   `Price(EIA) in 2014$` = col_double(),
##   `Nominal Price(EIA)` = col_double(),
##   `AE098 Forecast in 2014$` = col_double(),
##   `AE099 Forecast in 2014$` = col_double(),
##   CPI = col_double()
## )
```

```
Oil <- Oil[1:42, ]
gf_point(Oil$`Nominal Price(EIA)` ~ Oil$`Years since 1970`,
         alpha = 0.5, col = "red") %>%
  gf_line(alpha = 0.5, col = "red") %>%
  gf_labs(x = "Years Since 1970", y = "Price (2005 $)") %>%
  gf_point(Oil$`AE098 Forecast in 2014$` ~ Oil$`Years since 1970`,
           alpha = 0.5, col = "green") %>%
  gf_point(Oil$`AE099 Forecast in 2014$` ~ Oil$`Years since 1970`,
           alpha = 0.5, col = "blue")
```



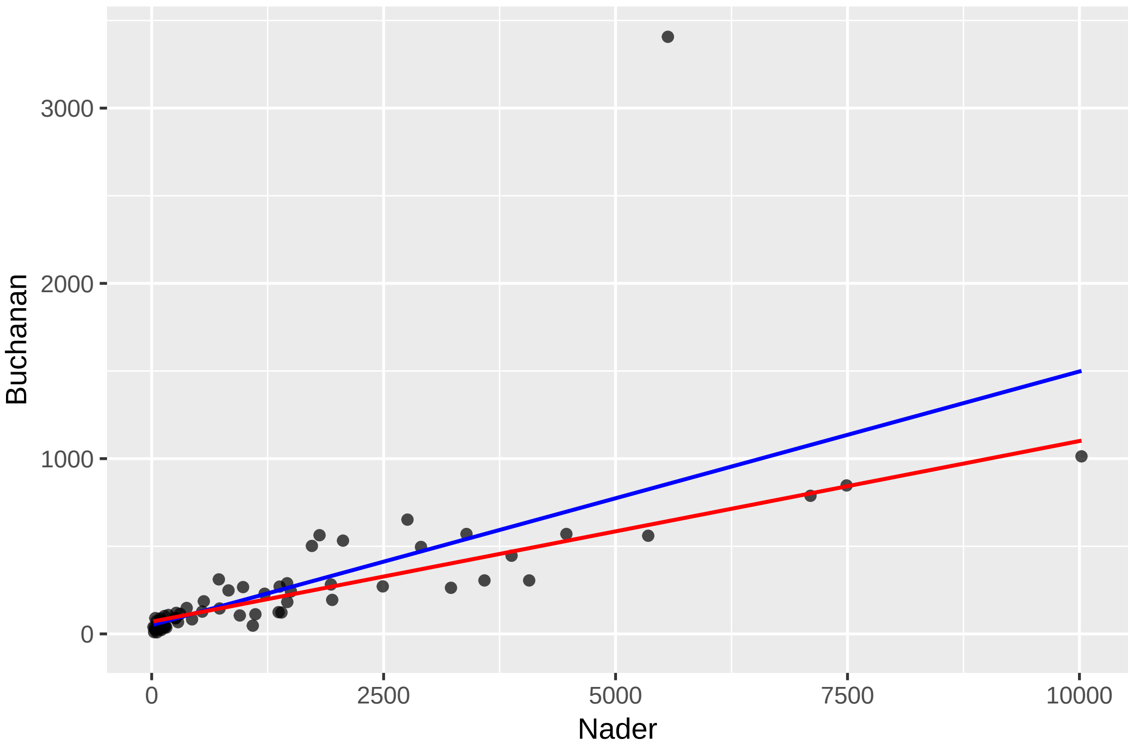
Section 8.3: Outliers, leverage, and influence

To get the two regression lines shown by figure 8.10 on page 222 the `filter` function is used to create a dataset without the Palm Beach measurement. The `gf_point` function and `gf_lm` functions together will add the regression lines.

```
Election2000 <- read_csv("http://www3.amherst.edu/~nhorton/sdm4/data/Election_2000.csv")

Election2000_2 <- filter(Election2000, Buchanan < 2000)

gf_point(Buchanan ~ Nader, data = Election2000, alpha = 0.7) %>%
    gf_lm(Buchanan ~ Nader, data = Election2000, col = "blue") %>%
  gf_lm(Buchanan ~ Nader, data = Election2000_2, col = "red")
```


Buchanan
Nader