SDM4 in R: Re-expressing Data: Get it Straight! (Chapter 9)

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

August 12, 2017

Introduction and background

This document is intended to help describe how to undertake analyses introduced as examples in the Fourth Edition of *Stats: Data and Models* (2014) 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 http://nhorton.people.amherst.edu/sdm4.

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.

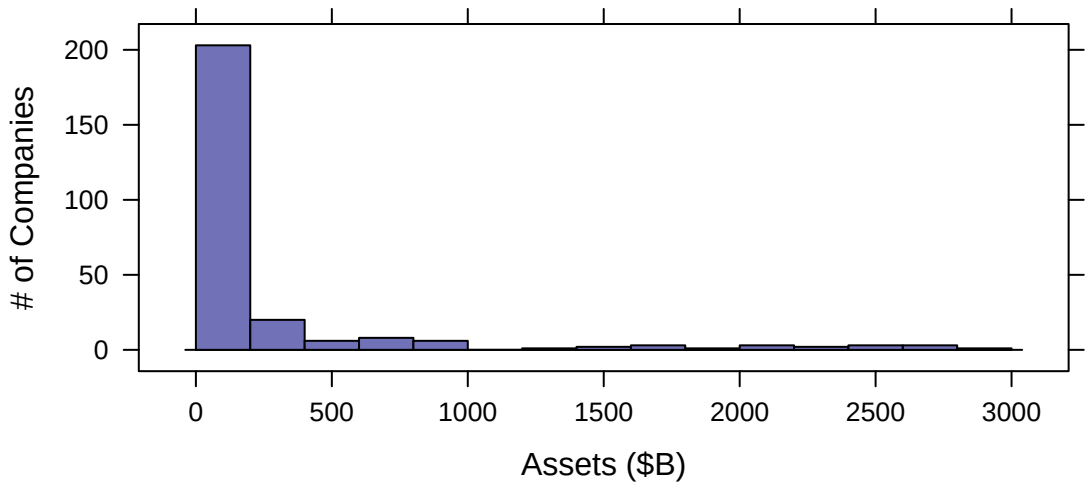
Chapter 9: Re-expressing Data: Get it Straight!

Section 9.1: Straightening Scatterplots - The Four Goals

The `histogram` function will generate the histograms shown by figure 9.4 on page 249.

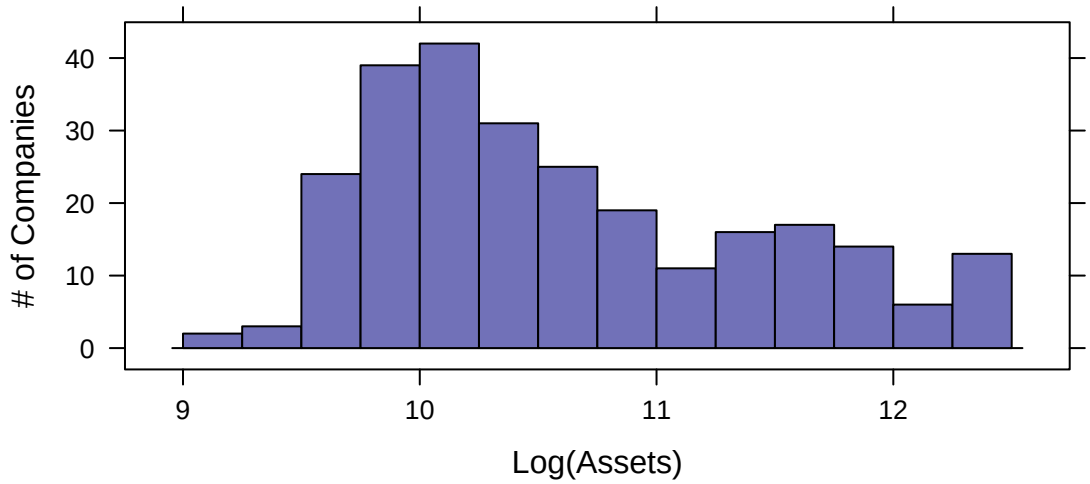
```
library(mosaic); library(readr)
options(digits=3)
Forbes <- read.csv("http://nhorton.people.amherst.edu/sdm4/data/Forbes_Global_2000.csv")

histogram(~ Assets..B., data = Forbes,
          center = 100, width = 200, type = "count",
          xlab = "Assets ($B)", ylab = "# of Companies")
```



As `Assets..B.` are the assets in billions, we have to add 9 (log(1,000,000,000)) to each value of log(Assets..B.) to get log(Assets)

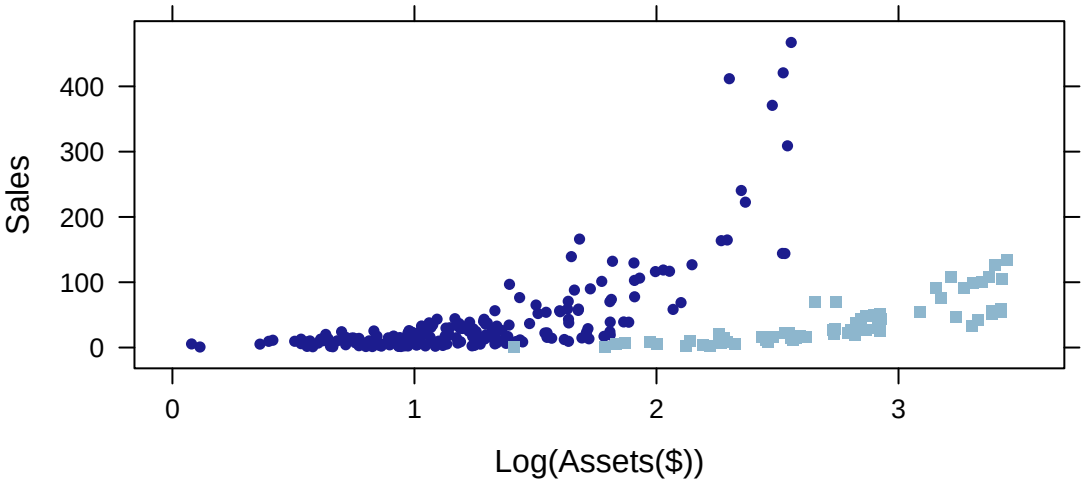
```
histogram(~ (log(Assets..B., 10) + 9), data = Forbes,
          center = 0.25/2, width = 0.25, type = "count",
          xlab = "Log(Assets)", ylab = "# of Companies")
```



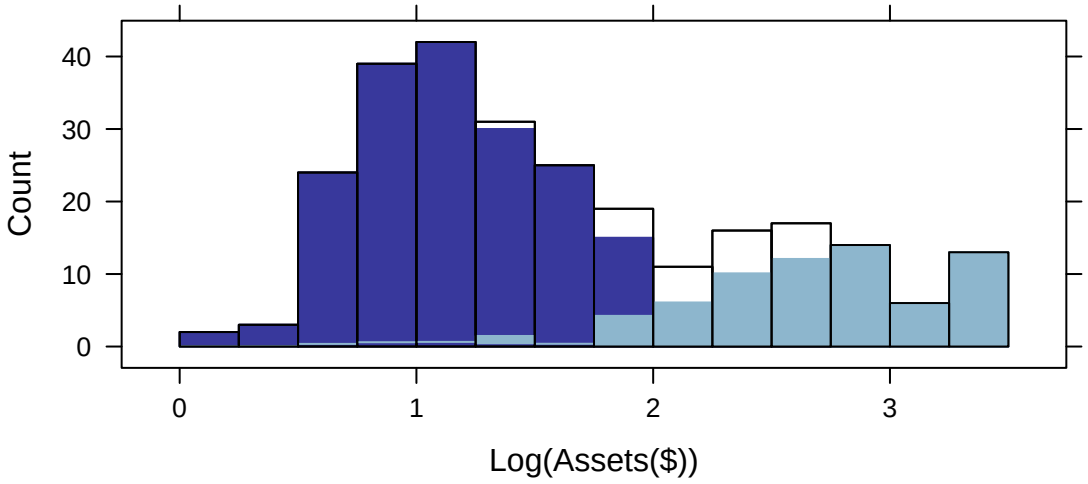
To group by whether the `Sector` is Finance or not, we use the `mutate` and `ifelse` functions. Then the scatterplot and histogram of figure 9.7 on page 251 can be generated by utilizing the `groups =` query.

```
Forbes <- mutate(Forbes, isFin = ifelse(Sector == "Finance", 1, 0))

xyplot(Sales ~ (log(Assets..B., 10)), data = Forbes,
       groups = isFin, auto.key = "true",
       xlab = "Log(Assets($))", ylab = "Sales")
```




```
histogram( ~ (log(Assets..B., 10)), data = Forbes,
       groups = isFin, type = "count", stripes = "horizontal",
       width = 0.75/3, center = 0.75/6,
       xlab = "Log(Assets($))")
```

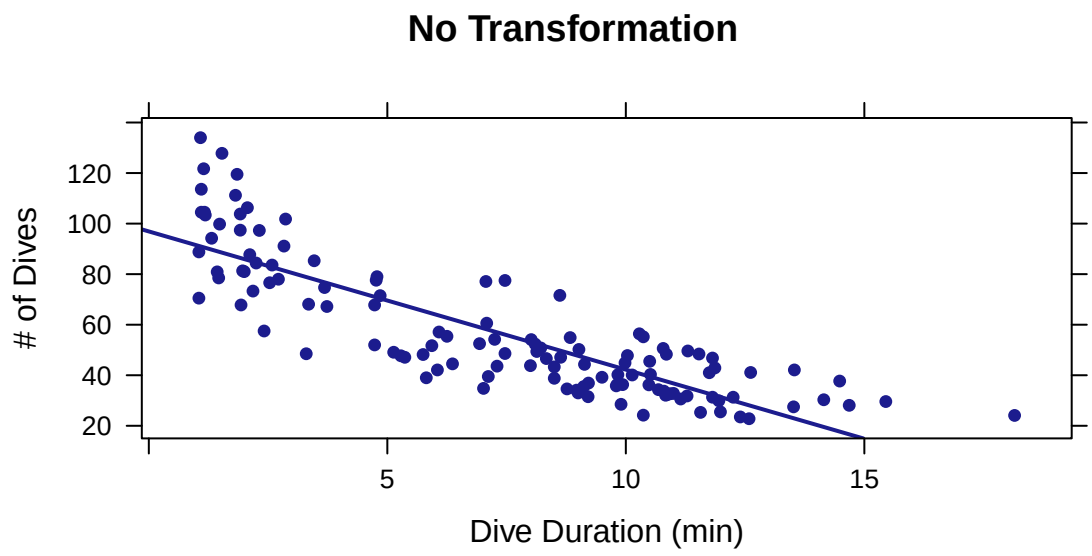


Section 9.2: Finding a Good Re-expression

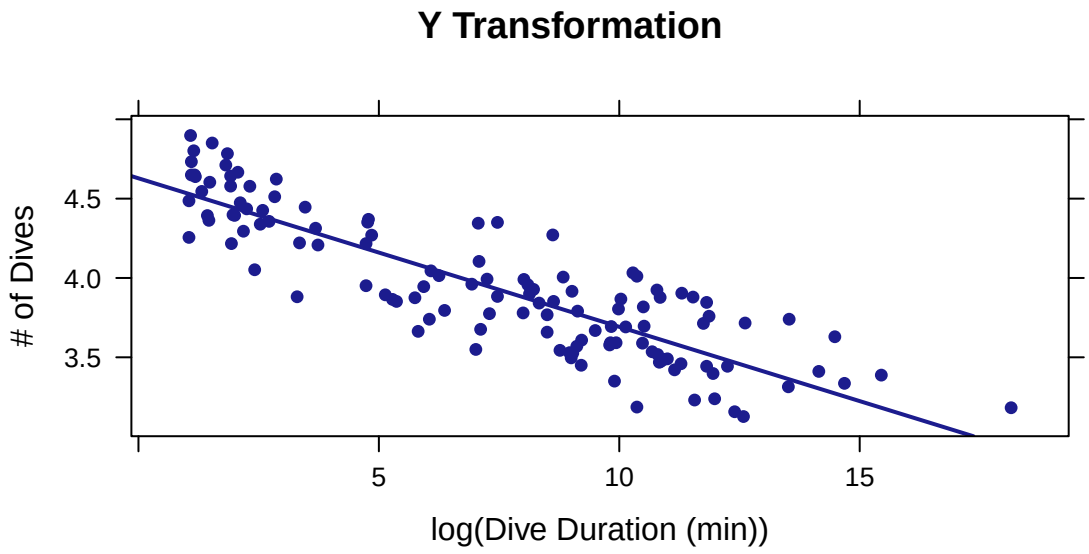
Looking at the penguins example mentioned on page 251 we can see how different log transformations affect the `xyplot` of the two variables:

```
Penguins <- read.csv("http://nhorton.people.amherst.edu/sdm4/data/Penguins.csv")

xyplot(DiveHeartRate ~ Duration, data = Penguins, type = c("p", "r"),
       main = "No Transformation", xlab = "Dive Duration (min)", ylab = "# of Dives")
```




```
xyplot(log(DiveHeartRate) ~ Duration, data = Penguins, type = c("p", "r"),
       main = "Y Transformation", xlab = "log(Dive Duration (min))", ylab = "# of Dives")
```




```
xyplot(log(DiveHeartRate) ~ log(Duration), data=Penguins, type = c("p", "r"),
       main = "X and Y Transformations", xlab = "log(Dive Duration (min))", ylab = "log(# of Dives)")
```