

IS5 in R: Scatterplots, Association, and Correlation (Chapter 6)

Nicholas Horton (nhorton@amherst.edu)

December 13, 2020

Introduction and background

This document is intended to help describe how to undertake analyses introduced as examples in the Fifth Edition of *Intro Stats* (2018) by De Veaux, Velleman, and Bock. 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/is5>.

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 (<https://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>.

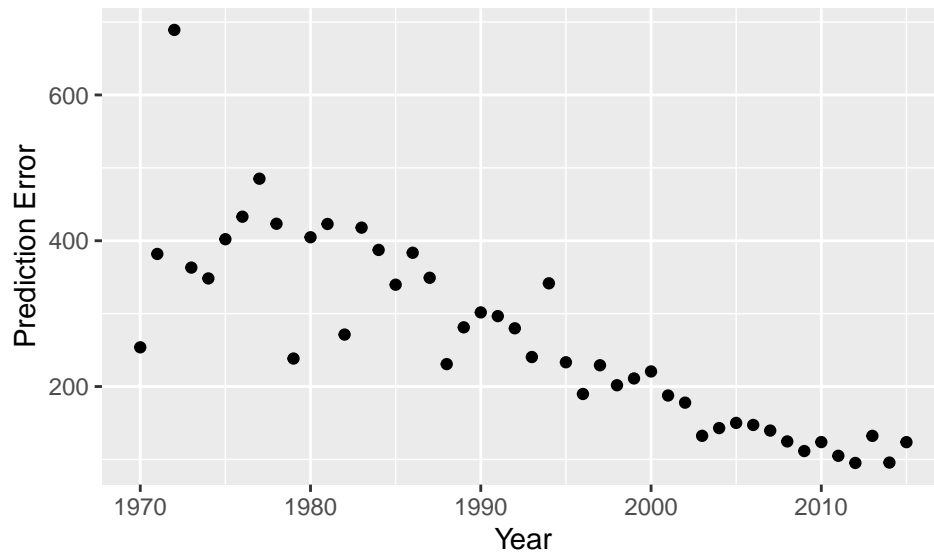
Chapter 6: Scatterplots, Association, and Correlation

```
library(mosaic)
library(readr)
library(janitor)
Hurricanes <- read_csv("http://nhorton.people.amherst.edu/is5/data/Tracking_hurricanes_2015.csv")

##
## -- Column specification -----
## cols(
##   Year = col_double(),
##   Error_24h = col_double(),
##   Error_48h = col_double(),
##   Error_72h = col_double()
## )
```

By default, `read_csv()` prints the variable names. These messages can be suppressed using the `message=FALSE` code chunk option to save space and improve readability.

```
# Figure 6.1, page 164
gf_point(Error_72h ~ Year, data = Hurricanes, ylab = "Prediction Error")
```



Section 6.1: Scatterplots

See dots on pages 164-165.

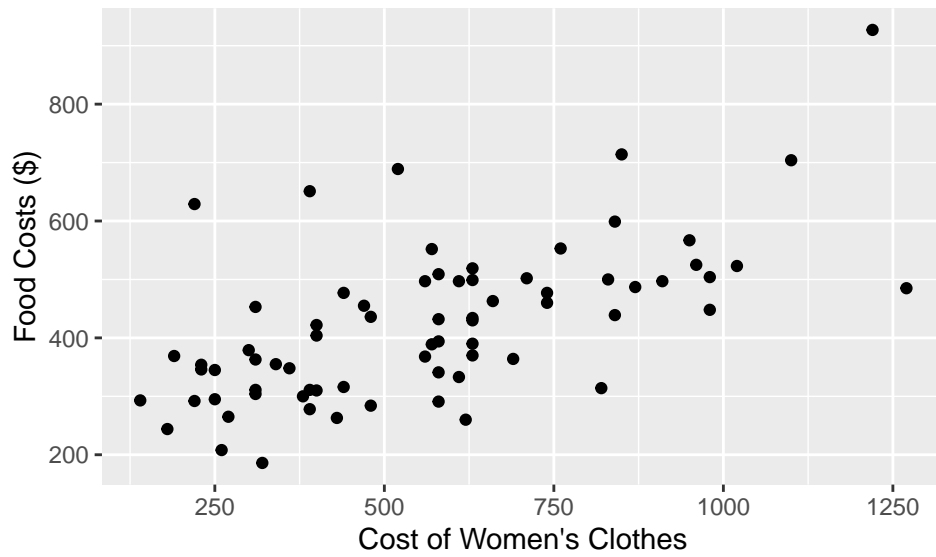
```
Prices <- read_csv("http://nhorton.people.amherst.edu/is5/data/Prices_and_Earnings.csv") %>%
  janitor::clean_names()
names(Prices)
```

Example 6.1: Comparing Prices Worldwide

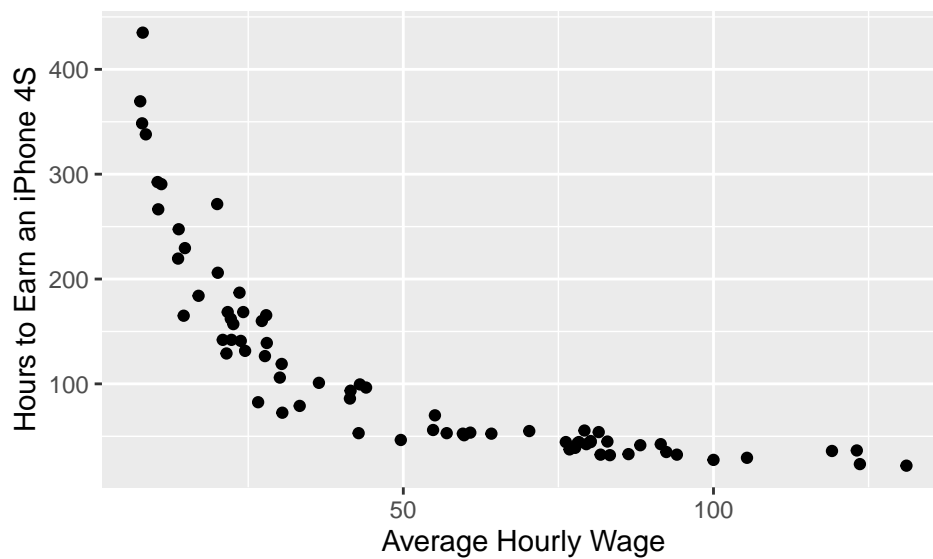
```
## [1] "city"                "food_costs"
## [3] "womens_clothing"     "mens_clothing"
## [5] "i_phone_4s_hr"       "clothing_index"
## [7] "hours_worked"        "wage_gross"
## [9] "wage_net"            "vacation_days"
## [11] "col_excl_rent"       "col_incl_rent"
## [13] "pur_power_gross"     "pur_power_net"
## [15] "pur_power_annual"    "big_mac_min"
## [17] "bread_kg_in_min"     "rice_kg_in_min"
## [19] "goods_and_services" "good_and_services_index"
## [21] "food_index"
```

Here we use the `clean_names()` function from the `janitor` package to sanitize the names of the columns (which would otherwise contain special characters or whitespace).

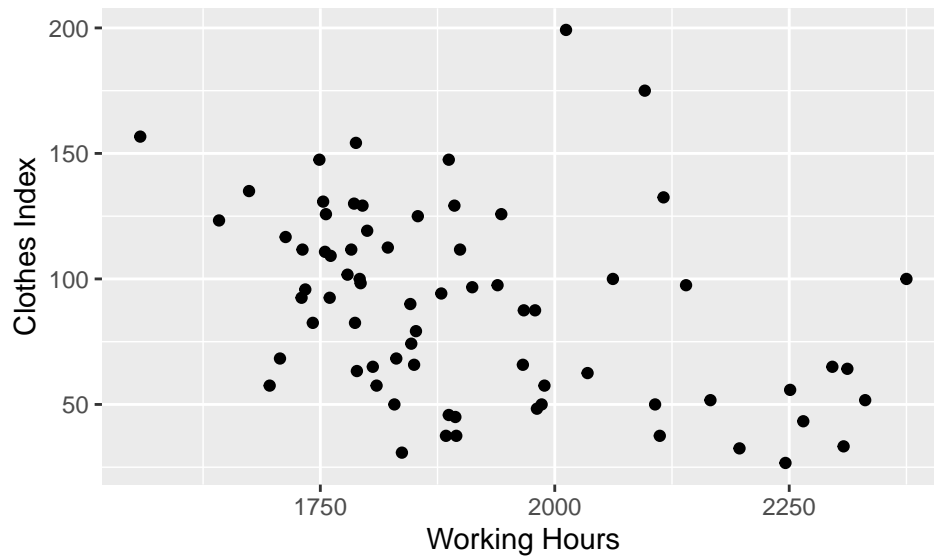
```
gf_point(food_costs ~ womens_clothing, data = Prices) %>%
  gf_labs(x = "Cost of Women's Clothes", y = "Food Costs ($)")
```



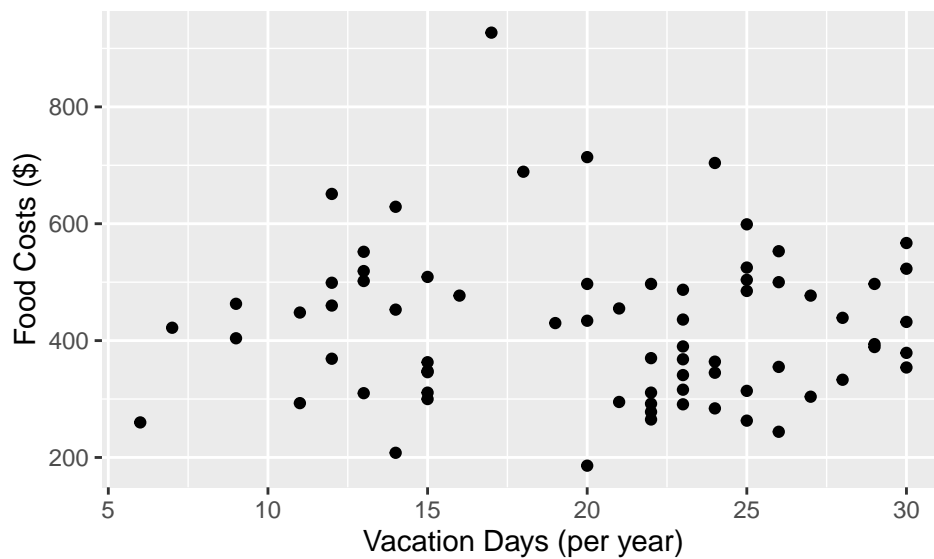
```
gf_point(i_phone_4s_hr ~ wage_gross, data = Prices) %>%
  gf_labs(x = "Average Hourly Wage", y = "Hours to Earn an iPhone 4S")
```



```
gf_point(clothing_index ~ hours_worked, data = Prices) %>%
  gf_labs(x = "Working Hours", y = "Clothes Index")
```

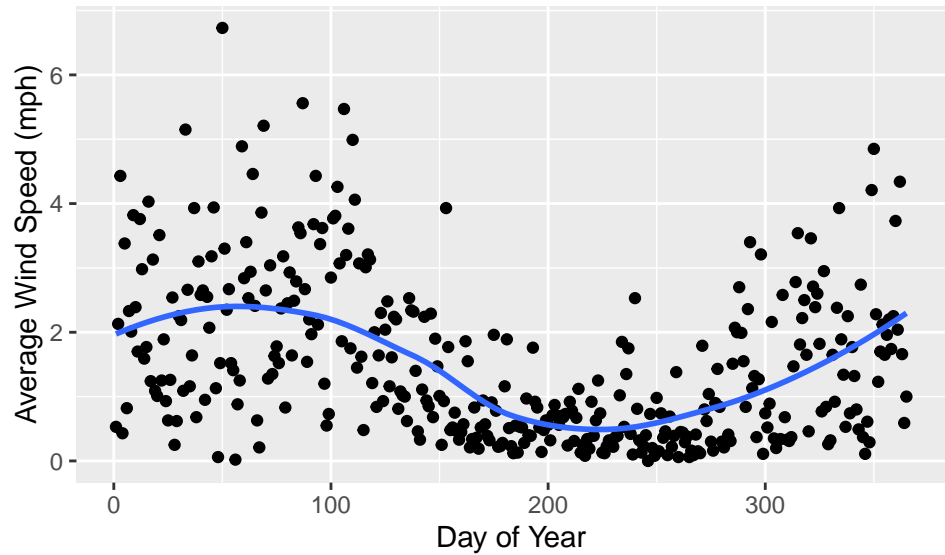


```
gf_point(food_costs ~ vacation_days, data = Prices) %>%
  gf_labs(x = "Vacation Days (per year)", y = "Food Costs ($)")
```



Roles for Variables

```
HopkinsForest <- read_csv("http://nhorton.people.amherst.edu/is5/data/Hopkins_Forest.csv") %>%
  janitor::clean_names()
# Figure 6.2, page 168
gf_point(avg_wind_mph ~ day_of_year, data = HopkinsForest) %>%
  gf_smooth(se = FALSE) %>%
  gf_labs(x = "Day of Year", y = "Average Wind Speed (mph)")
```

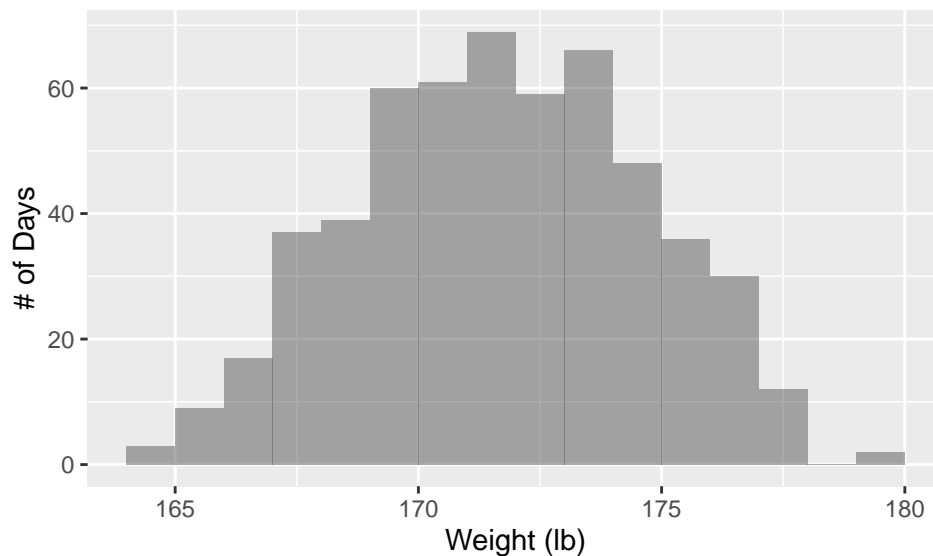


Smoothing Scatterplots

```
Fitness <- read_csv("http://nhorton.people.amherst.edu/is5/data/Fitness_data.csv") %>%
  janitor::clean_names()
gf_histogram(~weight, data = Fitness, binwidth = 1, center = .5) %>%
  gf_labs(x = "Weight (lb)", y = "# of Days")
```

Example 6.2: Smoothing Timeplots

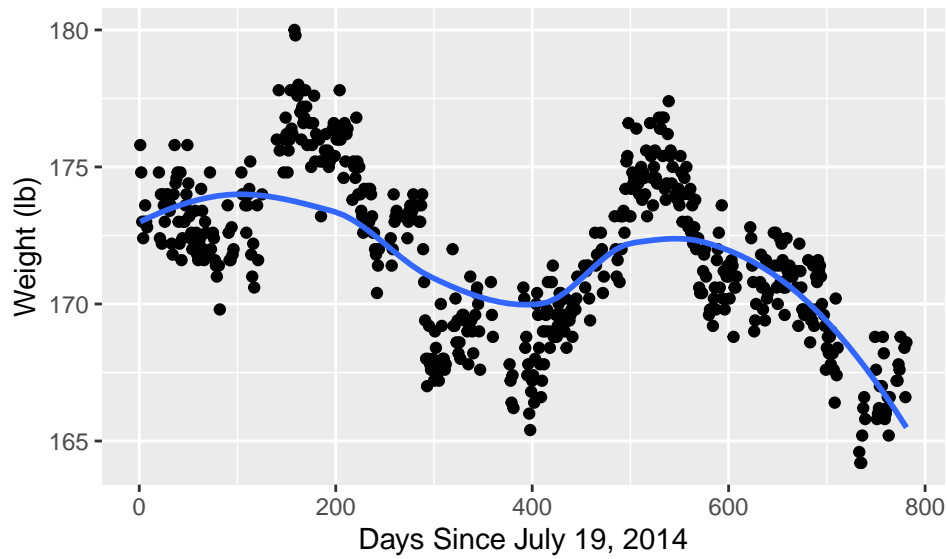
Warning: Removed 70 rows containing non-finite values (stat_bin).



```
gf_point(weight ~ days_since_july_19_2014, data = Fitness) %>%
  gf_smooth(se = FALSE) %>%
  gf_labs(x = "Days Since July 19, 2014", y = "Weight (lb)")
```

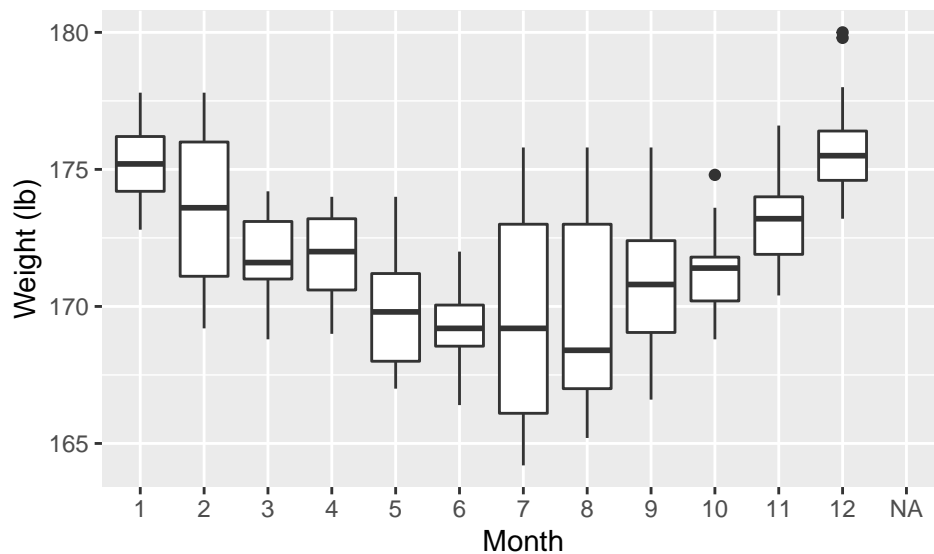
Warning: Removed 70 rows containing non-finite values (stat_smooth).

Warning: Removed 70 rows containing missing values (geom_point).



```
gf_boxplot(weight ~ as.factor(month), data = Fitness) %>%
  gf_labs(x = "Month", y = "Weight (lb)")
```

Warning: Removed 70 rows containing non-finite values (stat_boxplot).



Warnings can be suppressed with the `warnings=FALSE` chunk option.

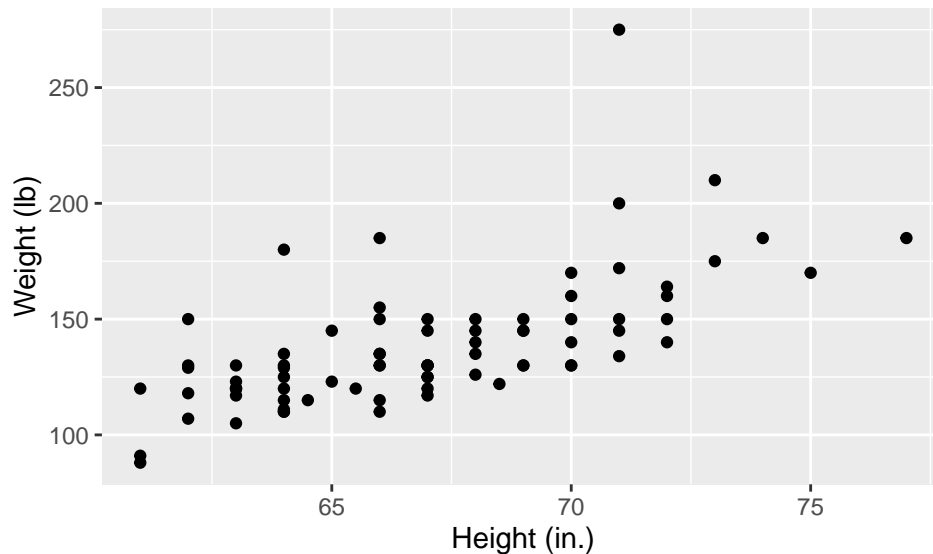
Section 6.2: Correlation

```
HeightsWeights <- read_csv("http://nhorton.people.amherst.edu/is5/data/Heights_and_weights.csv")

##
## -- Column specification -----
## cols(
##   Weight = col_double(),
##   Height = col_double()
## )
```

Figure 6.3, page 170

```
gf_point(Weight ~ Height, data = HeightsWeights) %>%
  gf_labs(x = "Height (in.)", y = "Weight (lb)")
```



```
cor(Weight ~ Height, data = HeightsWeights)
```

```
## [1] 0.6440311
```

See displays on pages 170 - 171.

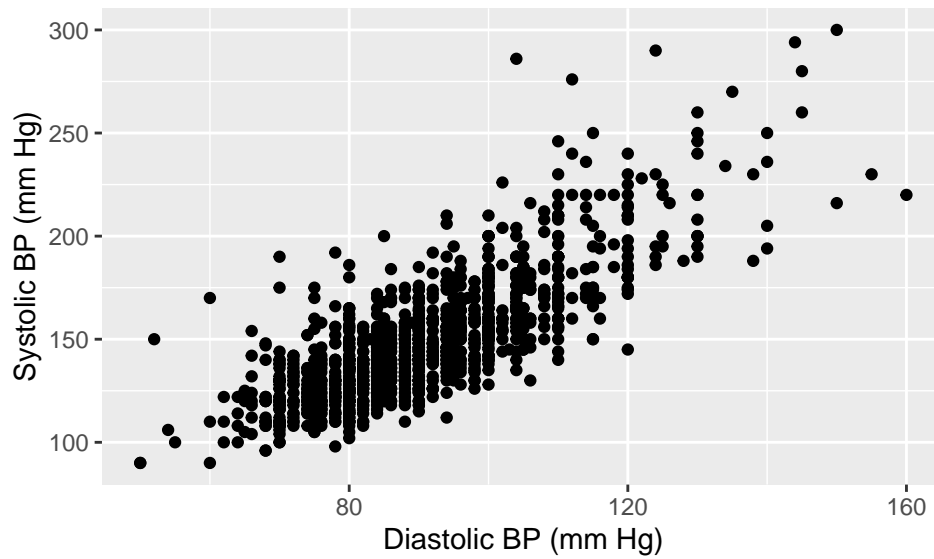
```
Framingham <- read_csv("http://nhorton.people.amherst.edu/is5/data/Framingham.csv")
```

Step-by-Step Example: Looking at Association

```
##
## -- Column specification -----
## cols(
##   Cholesterol = col_double(),
##   Age = col_double(),
##   Sex = col_character(),
##   SBP = col_double(),
##   DBP = col_double(),
##   CIG = col_double()
## )

## Warning: 1 parsing failure.
##   row col expected actual                                     file
## 1090 CIG a double      . 'http://nhorton.people.amherst.edu/is5/data/Framingham.csv'

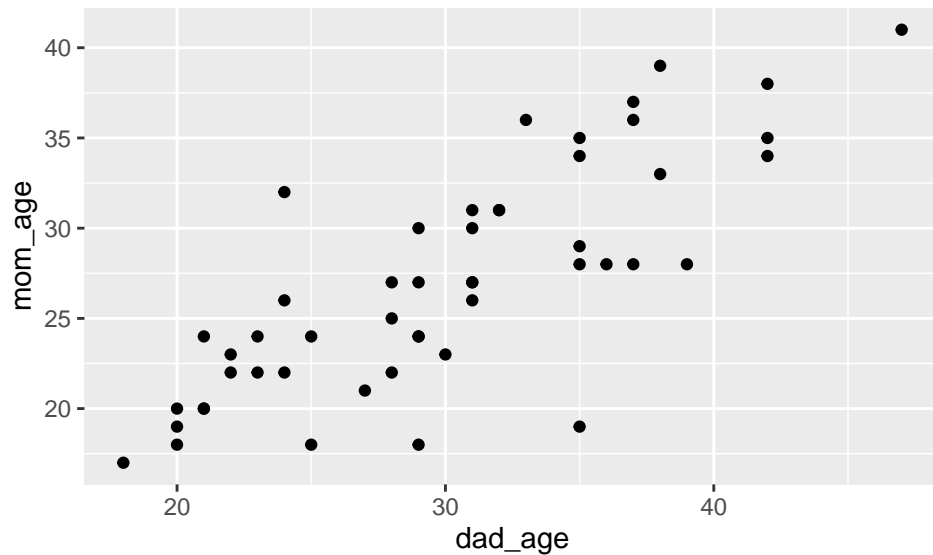
gf_point(SBP ~ DBP, data = Framingham) %>%
  gf_labs(x = "Diastolic BP (mm Hg)", y = "Systolic BP (mm Hg)")
```



```
cor(SBP ~ DBP, data = Framingham)
```

```
## [1] 0.7924792
```

```
LiveBirths <- read_csv("http://nhorton.people.amherst.edu/is5/data/Babysamp_98.csv") %>%
  janitor::clean_names()
LiveBirths <- LiveBirths %>%
  filter(dad_age != "NA")
set.seed(14513) # To ensure we get the same values when we run it multiple times
numsim <- 10000 # Number of samples
gf_point(mom_age ~ dad_age, data = sample(LiveBirths, size = 50))
```



Random Matters: Correlations Vary

```
# Graph will look different for different samples
cor(mom_age ~ dad_age, data = LiveBirths)
```

```
## [1] 0.7516507
```



```

# What does do() do?
cor(mom_age ~ dad_age, data = sample(LiveBirths, size = 50)) # Correlation of one random sample

## [1] 0.7596176

cor(mom_age ~ dad_age, data = sample(LiveBirths, size = 50)) # Correlation of another random sample

## [1] 0.8087199

do(2) * cor(mom_age ~ dad_age, data = sample(LiveBirths, size = 50)) # Finds the correlation twice

##           cor
## 1 0.7447461
## 2 0.7583890

# For the visualization, we need 10,000 correlations
LiveCorr <- do(numsim) * cor(mom_age ~ dad_age, data = sample(LiveBirths, size = 50))

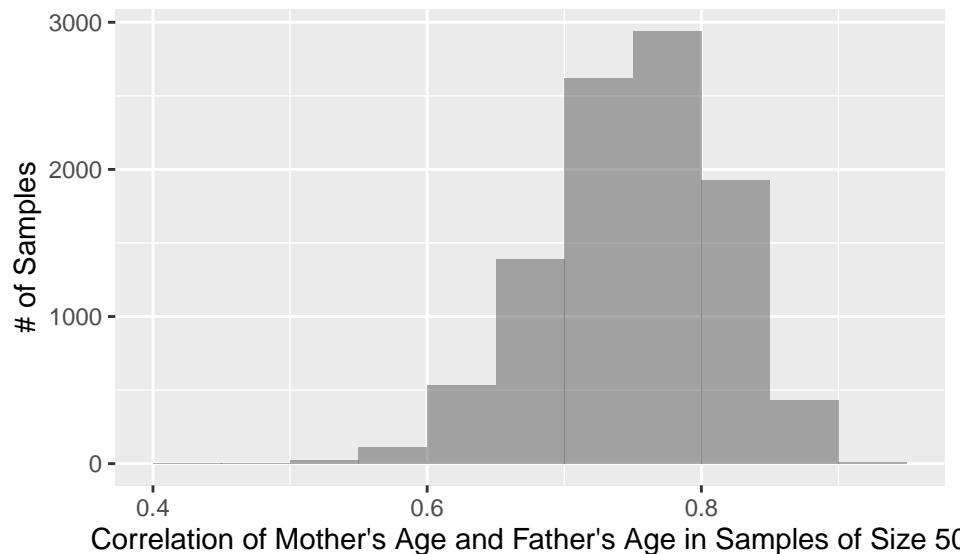
```

The `do()` function runs, 10,000 times, the correlation and sampling functions on a random sample of 50. (We can use the chunk option `cache=TRUE` to enable caching to save results for next time.)

```

# Figure 6.8, page 176
gf_histogram(~cor, data = LiveCorr, binwidth = .05, center = .025) %>%
  gf_labs(
    x = "Correlation of Mother's Age and Father's Age in Samples of Size 50",
    y = "# of Samples"
  )

```



Section 6.3: Warning: Correlation \neq Causation

```

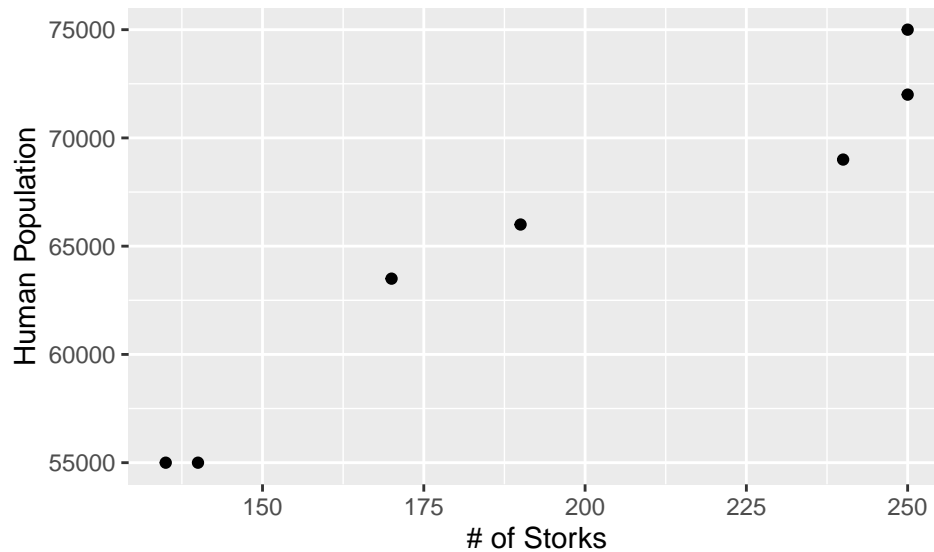
Storks <- read_csv("http://nhorton.people.amherst.edu/is5/data/Storks.csv")

##
## -- Column specification -----
## cols(
##   Storks = col_double(),
##   Population = col_double()
## )

```

Figure 6.9

```
gf_point(Population ~ Storks, data = Storks) %>%
  gf_labs(x = "# of Storks", y = "Human Population")
```



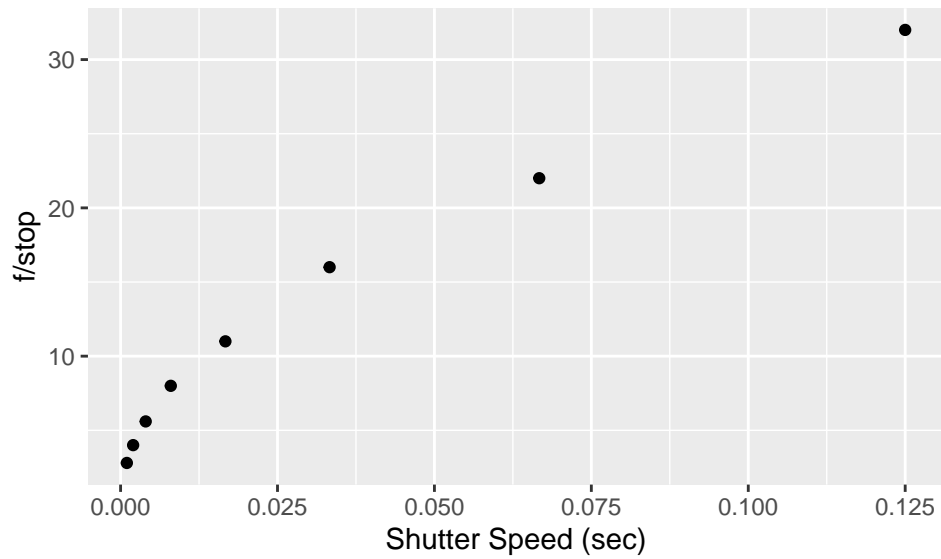
```
Companies <- read_csv("http://nhorton.people.amherst.edu/is5/data/Companies.csv") %>%
  janitor::clean_names()
# Table 6.1, page 178
Companies %>%
  select(assets, sales, market_value, profits, cash_flow, employees) %>%
  cor()
```

Correlation Tables

##	assets	sales	market_value	profits	cash_flow	employees
## assets	1.0000000	0.7464649	0.6822122	0.6016986	0.6409018	0.5943581
## sales	0.7464649	1.0000000	0.8788920	0.8137758	0.8549172	0.9240429
## market_value	0.6822122	0.8788920	1.0000000	0.9681987	0.9702851	0.8182161
## profits	0.6016986	0.8137758	0.9681987	1.0000000	0.9887795	0.7621057
## cash_flow	0.6409018	0.8549172	0.9702851	0.9887795	1.0000000	0.7866148
## employees	0.5943581	0.9240429	0.8182161	0.7621057	0.7866148	1.0000000

Section 6.4: Straightening Scatterplots

```
FStops <- read_csv("http://nhorton.people.amherst.edu/is5/data/F-stops.csv") %>%
  janitor::clean_names()
# Figure 6.10, page 179
gf_point(f_stop ~ shutter_speed, data = FStops) %>%
  gf_labs(x = "Shutter Speed (sec)", y = "f/stop")
```



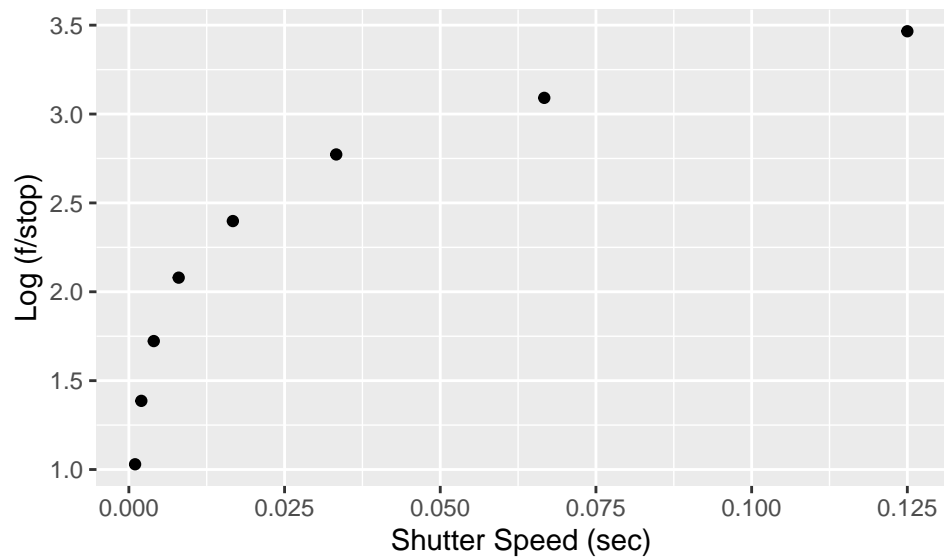
```
cor(f_stop ~ shutter_speed, data = FStops)
```

```
## [1] 0.9786716
```

The Ladder of Powers

Figure 6.11, page 181

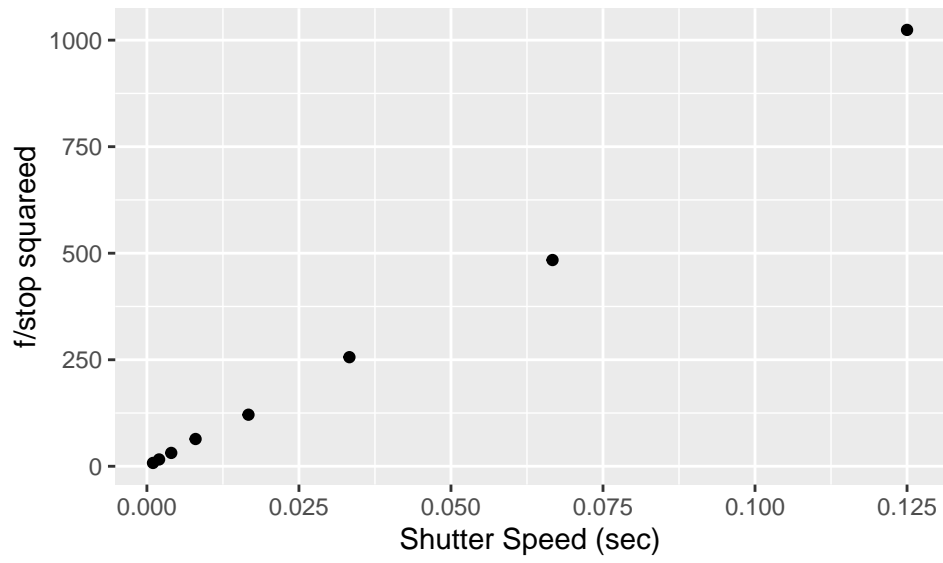
```
gf_point(log(f_stop) ~ shutter_speed, data = FStops) %>%
  gf_labs(x = "Shutter Speed (sec)", y = "Log (f/stop)")
```



f/Stops Again

Figure 6.12

```
gf_point((f_stop)^2 ~ shutter_speed, data = FStops) %>%
  gf_labs(x = "Shutter Speed (sec)", y = "f/stop squared")
```



See displays in “What Can Go Wrong?” on pages 181-183.