IS5 in R: Displaying and Describing Data (Chapter 2)

Margaret Chien and Nicholas Horton (nhorton@amherst.edu)

July 13, 2018

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. 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/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 (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 2: Displaying and Describing Data

Section 2.1: Summarizing and Displaying a Categorical Variable

```r
library(mosaic)
library(readr)
library(janitor) #for variable names
options(digits = 3)
Titanic <- read_csv("http://nhorton.people.amherst.edu/is5/data/Titanic.csv")
```

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.

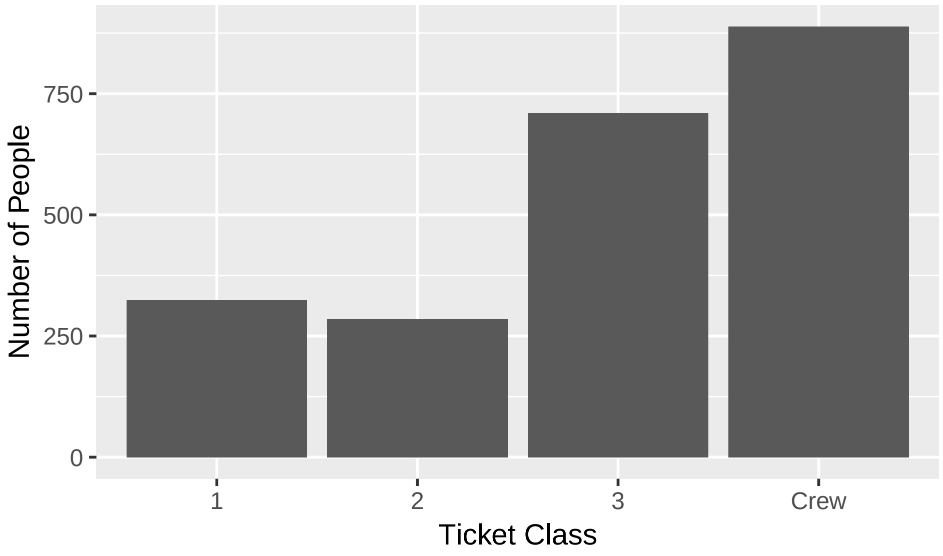
```r
#Table 2.2, page 19
tally(~ Class, data = Titanic)
```

```
## Class
##    1    2    3 Crew
##  324  285  710  889
```

```r
#Table 2.3
tally(~ Class, format = "percent", data = Titanic)
```

```
## Class
##    1    2    3 Crew
## 14.7 12.9 32.2 40.3
```

```r
#Figure 2.2, page 19
gf_bar(~ Class, data = Titanic) %>%
  gf_labs(x = "Ticket Class", y = "Number of People")
```



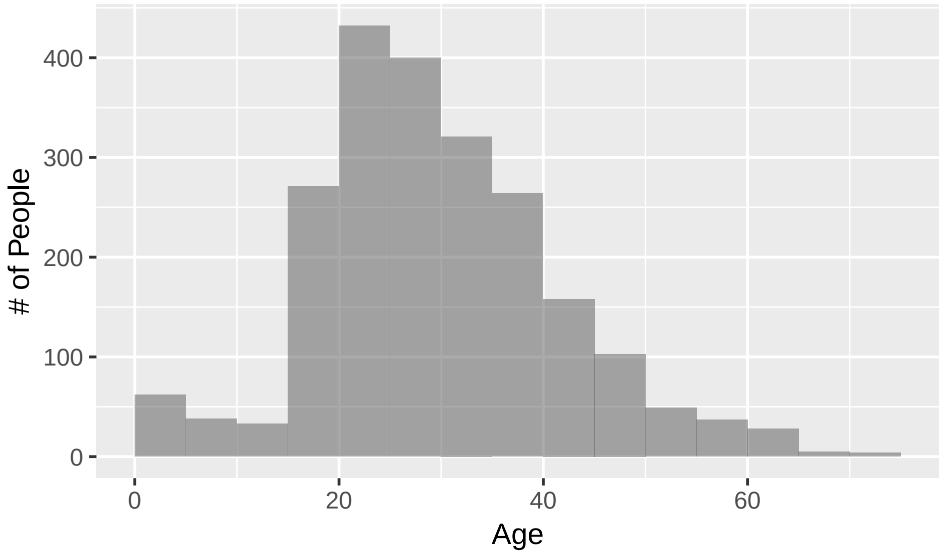
`~ x` is the general modeling language for one variable in `mosaic`.
We use `gf_bar()` to make a bar graph using the `ggformula` system, which is automatically downloaded with the `mosaic` package.

Section 2.2: Displaying a Quantitative Variable

Ages of Those Aboard the Titanic

```
# Figure 2.7, page 24
gf_histogram(~ Age, data = Titanic, binwidth = 5, ylab = "# of People", center = 5/2)
```

```
## Warning: Removed 3 rows containing non-finite values (stat_bin).
```



The function generates a warning because three of the ages are missing; this output can be suppressed by adding `warning=FALSE` as an option in this code chunk.

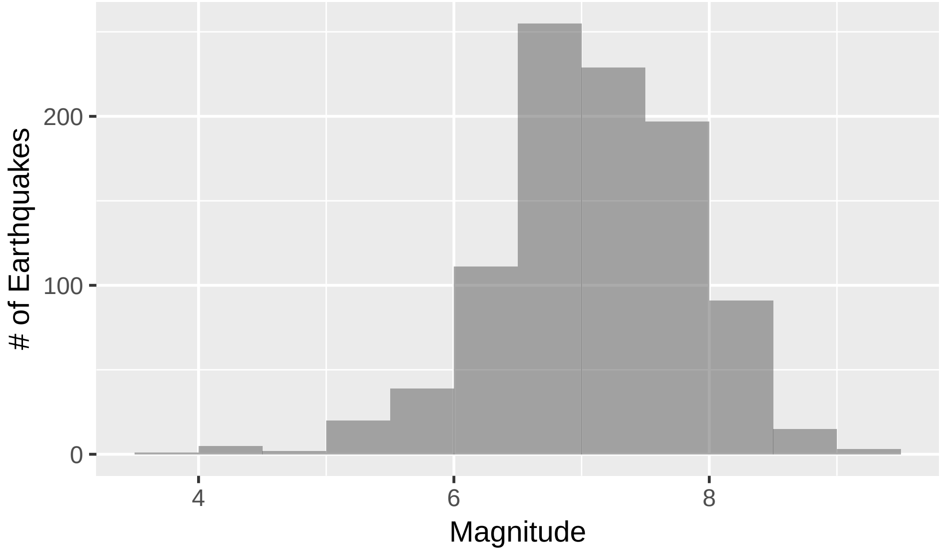
Earthquakes and Tsunamis

```
#Example 2.3, page 25
Earthquakes <- read_csv("http://nhorton.people.amherst.edu/is5/data/Tsunamis_2016.csv")
```

```
## Parsed with column specification:
## cols(
##   Year = col_integer(),
##   Focal_Depth = col_integer(),
##   Primary_Magnitude = col_double(),
##   Country = col_character(),
##   Latitude = col_double(),
##   Longitude = col_double(),
##   Deaths = col_integer(),
##   Missing = col_integer(),
##   Injuriez = col_integer(),
##   `Damage($M)` = col_double()
## )
```

```
gf_histogram(~ Primary_Magnitude, data = Earthquakes, binwidth = 0.5,
             ylab = "# of Earthquakes", xlab = "Magnitude", center = 0.25)
```

```
## Warning: Removed 119 rows containing non-finite values (stat_bin).
```



```
# XX MC Not sure why the bins by the peak are off from textbook
```

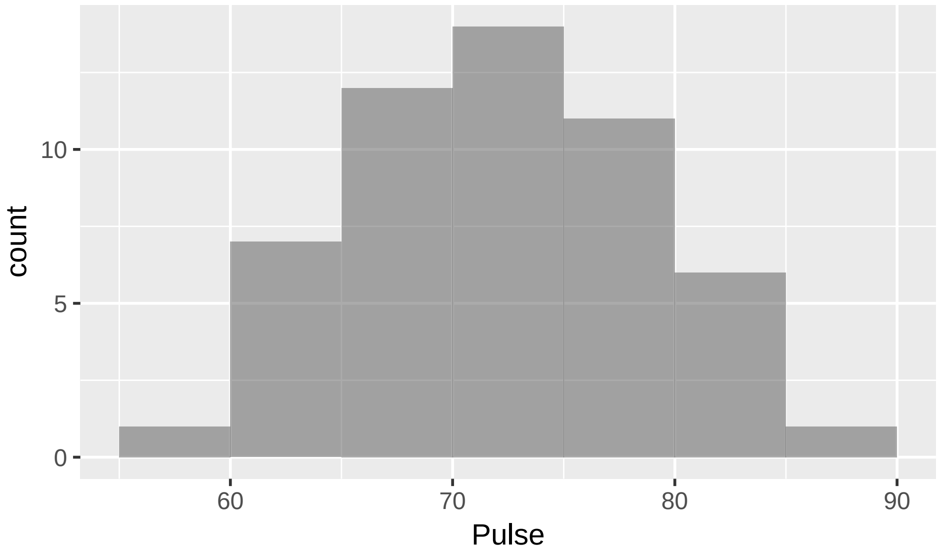
Stem-and-Leaf Displays

See page 26.

```
# Figure 2.8, page 26
Pulse_rates <- read_csv("http://nhorton.people.amherst.edu/is5/data/Pulse_rates.csv")
```

```
## Parsed with column specification:
## cols(
##   Pulse = col_integer()
## )
```

```
gf_histogram(~ Pulse, data = Pulse_rates, binwidth = 5, center = 5/2)
```



```
with(Pulse_rates, stem(Pulse))
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   5 | 7
##   6 | 13444
##   6 | 556668888899
##   7 | 0012223333444
##   7 | 5557777888889
##   8 | 0112233
##   8 | 6
```

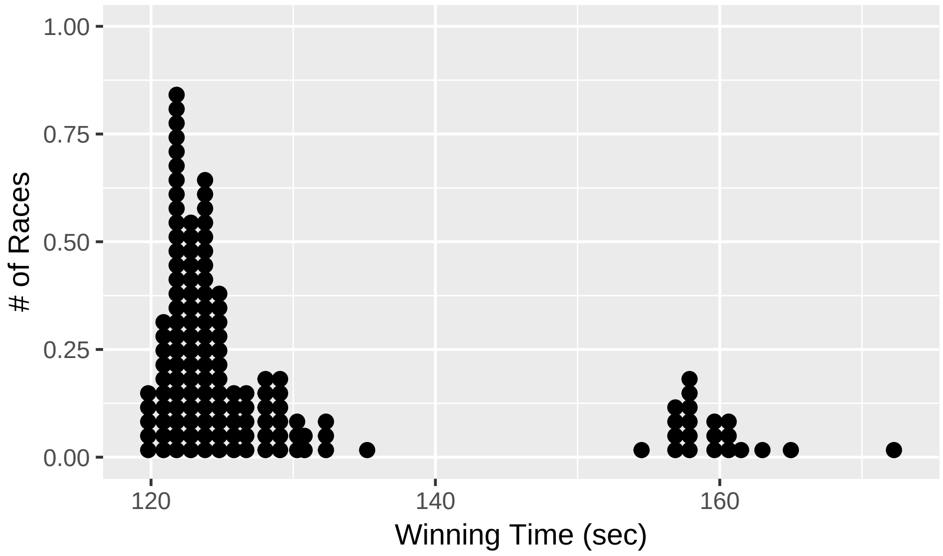
```
## MC The distribution looks very different. Seems like the textbook displayed the distributions of the
```

Dotplot

```
# Figure 2.9, page 27
Derby <- read_csv("http://nhorton.people.amherst.edu/is5/data/Kentucky_Derby_2016.csv")
```

```
## Parsed with column specification:
## cols(
##   Year = col_integer(),
##   Year_no = col_integer(),
##   Date = col_character(),
##   Winner = col_character(),
##   Mins = col_integer(),
##   Secs = col_double(),
##   Time_Sec = col_double(),
##   Distance = col_double(),
##   Speed_mph = col_double()
## )
```

```
gf_dotplot(~ Time_Sec, data = Derby, binwidth = 1) %>%
  gf_labs(x = "Winning Time (sec)", y = "# of Races")
```

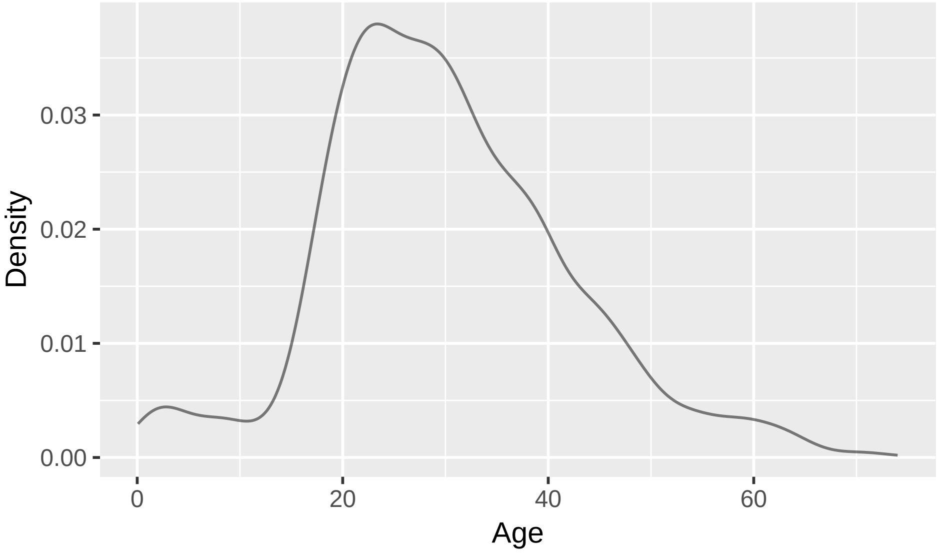


```
# XX MC and NH the y axis not displaying counts seems to be a bug in ggformula that cannot be fixed
```

Density Plots

```
# Figure 2.10, page 27
gf_dens(~ Age, data = Titanic, ylab = "Density")
```

```
## Warning: Removed 3 rows containing non-finite values (stat_density).
```



```
# XX BL be consistent with labs in ggformula
```

Section 2.3: Shape

See displays on pages 28-29.

Consumer Price Index

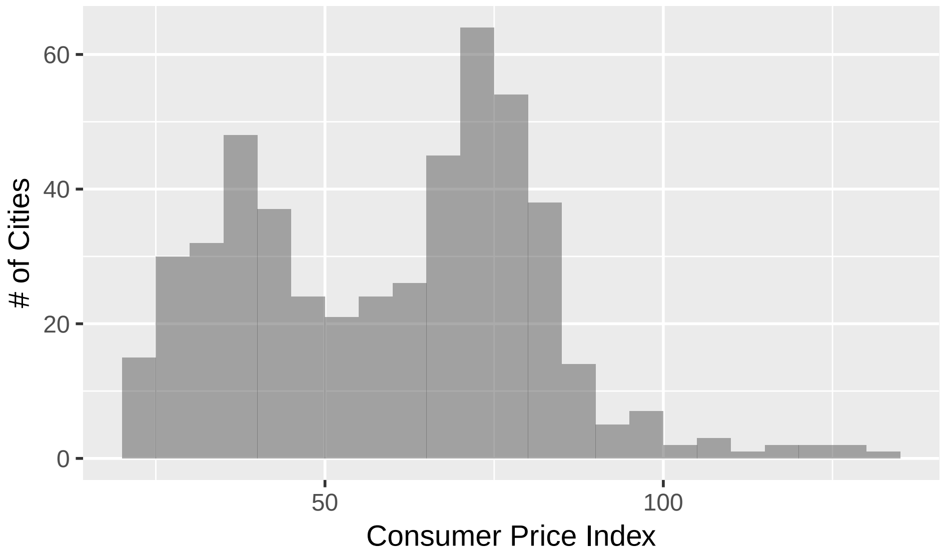
```
CPI <- read_csv("http://nhorton.people.amherst.edu/is5/data/CPI_Worldwide.csv") %>%
  clean_names()
```

```
## Parsed with column specification:
## cols(
##   City = col_character(),
##   Consumer.Price.Index = col_double(),
##   Rent.Index = col_double(),
##   Consumer.Price.Plus.Rent.Index = col_double(),
##   Groceries.Index = col_double(),
##   Restaurant.Price.Index = col_double(),
##   Local.Purchasing.Power.Index = col_double()
## )
```

```
names(CPI)
```

```
## [1] "city"                          "consumer_price_index"
## [3] "rent_index"                    "consumer_price_plus_rent_index"
## [5] "groceries_index"               "restaurant_price_index"
## [7] "local_purchasing_power_index"
```

```
#Example 2.5, page 30
gf_histogram(~ consumer_price_index, data = CPI, ylab = "# of Cities",
             xlab = "Consumer Price Index", binwidth = 5, center = 5/2)
```



<<<<<<< HEAD We can use the `clean_names()` function from the `janitor` package to format the names of the columns when necessary. You can use the `names()` function to check the reformatted names.
======= We can use `clean_names()` from the `janitor` package to format the names of the columns when necessary. You can use the `names()` function to check the reformatted names.
Piping (`%>%`) takes the output of the line of code and uses it in the next.
>>>>>>> 1a2a863fddfad80156770641f1b41383c6eefe9c

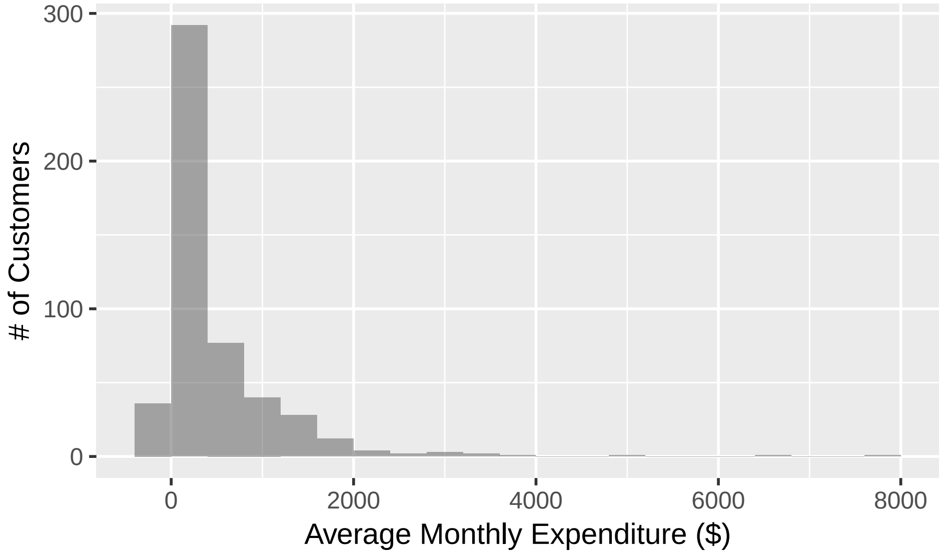
Credit Card Expenditures

```
CreditCardEx <- read_csv("http://nhorton.people.amherst.edu/is5/data/Credit_card_charges.csv") %>%
  clean_names()
```

```
## Parsed with column specification:
```

```
## cols(
##   `Charges($)` = col_double()
## )
```

```
# Figure 2.6, page 30
gf_histogram(~ charges, data = CreditCardEx, ylab = "# of Customers",
             xlab = "Average Monthly Expenditure ($)", binwidth = 400, center = 200)
```

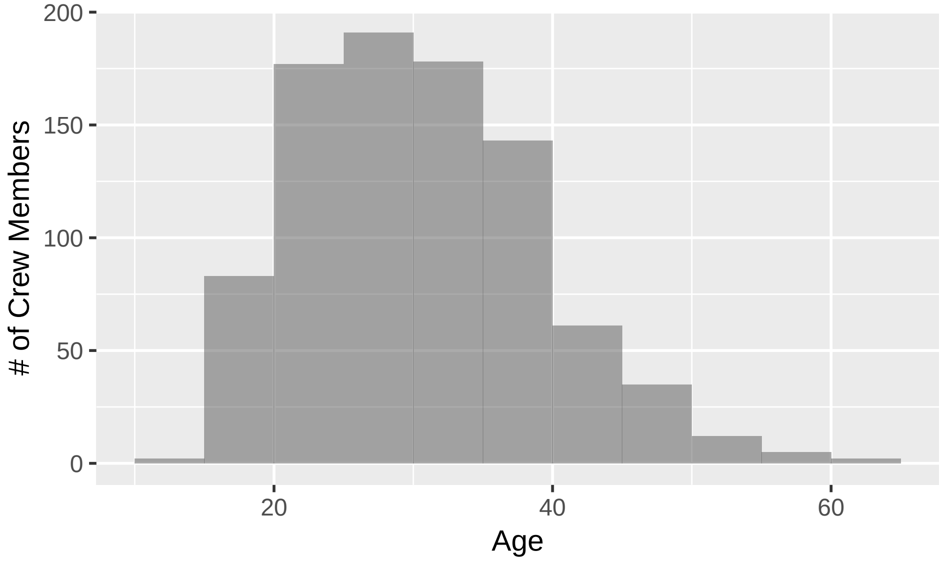


Section 2.4: Center

Finding Median and Mean

XX MC Do you want to include Figure 2.15 on page 32 for the coloring effects? That or include Figure 2.17 on page 33?

```
TitanicCrew <- filter(Titanic, Class == "Crew")
# Figure 2.16, page 33
gf_histogram(~ Age, data = TitanicCrew, ylab = "# of Crew Members", binwidth = 5, center = 5/2)
```



```
favstats(~ Age, data = TitanicCrew)
```

```
##  min Q1 median Q3 max mean   sd   n missing
##   14 24     30 37  62 31.1 8.55 889       0
```

Section 2.5: Spread

The Range

```
range(~ Age, data = TitanicCrew)
```

```
## [1] 14 62
```

```
diff(range(~ Age, data = TitanicCrew))
```

```
## [1] 48
```

The `range()` function returns the maximum and minimum values, so we can use the `diff()` function to find the difference between the two values.

The Interquartile Range

```
favstats(~ Age, data = TitanicCrew)
```

```
##  min Q1 median Q3 max mean   sd   n missing
##   14 24     30 37  62 31.1 8.55 889       0
```

```
IQR(~ Age, data = TitanicCrew)
```

```
## [1] 13
```

Using the `IQR()` function allows us to avoid having to manually find the IQR by subtracting Q1 from Q3 from the `favstats()` output.

Standard Deviation

```
sd(~ Age, data = TitanicCrew)
```

```
## [1] 8.55
```

```
var(~ Age, data = TitanicCrew)
```

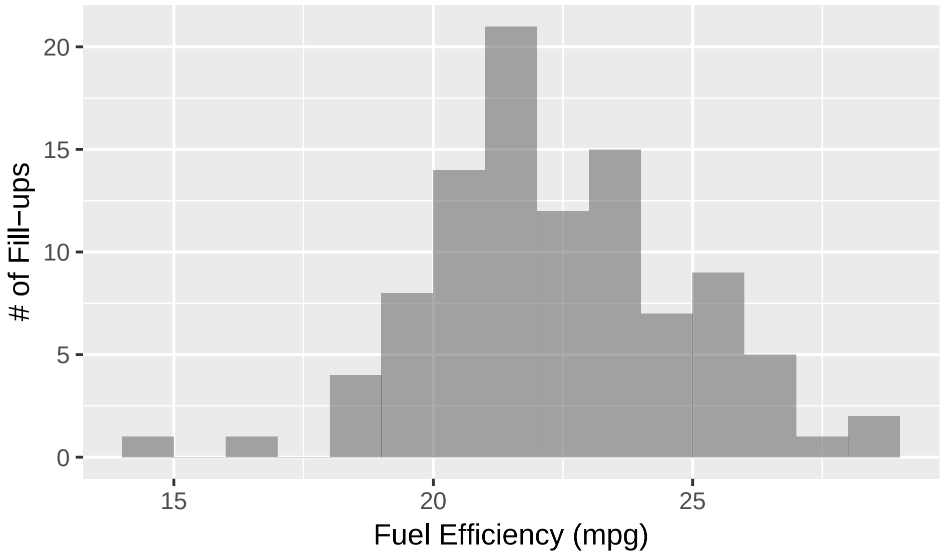
```
## [1] 73.1
```

Summarizing a Distribution

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

```
## Parsed with column specification:
## cols(
##   mpg = col_double()
## )
```

```
# Step-by-Step Example, page 39
gf_histogram(~ mpg, data = Nissan, binwidth = 1, xlab = "Fuel Efficiency (mpg)",
             ylab = "# of Fill-ups", center = 5/2)
```



```
favstats(~ mpg, data = Nissan)
```

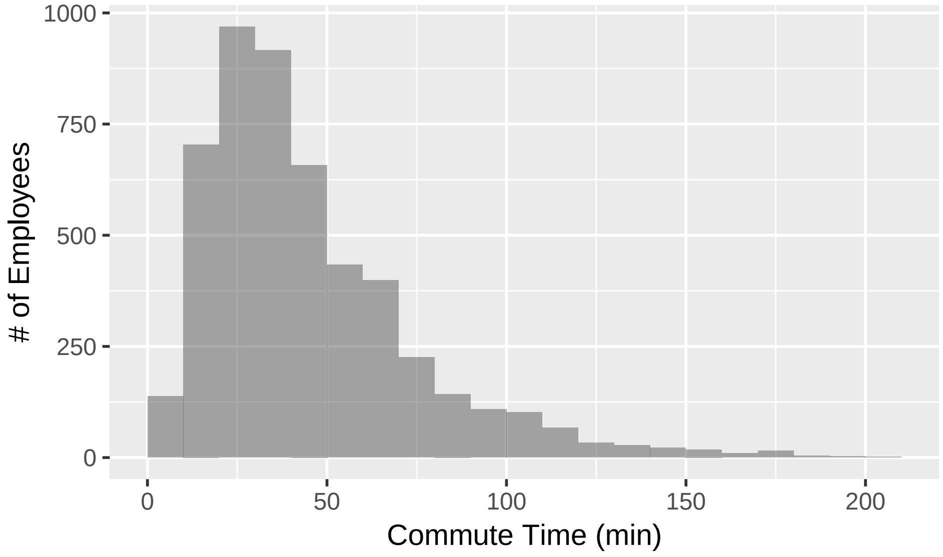
```
##   min   Q1 median Q3  max mean   sd   n missing
##  14.7 20.8   22.1 24 28.2 22.4 2.45 100       0
```

Random Matters

```
Commute <- read_csv("http://nhorton.people.amherst.edu/is5/data/Population_Commute_Times.csv") %>%
  clean_names()
```

```
## Parsed with column specification:
## cols(
##   Commute.Time = col_integer()
## )
```

```
# Figure 2.19, page 40
gf_histogram(~ commute_time, data = Commute, binwidth = 10, xlab = "Commute Time (min)",
             ylab = "# of Employees", center = 5)
```


# of Employees
Commute Time (min)