

IS5 in R: Understanding and Comparing Distributions (Chapter 4)

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

June 27, 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 4: Understanding and Comparing Distributions

```
library(mosaic)
library(readr)
library(janitor)
HopkinsForest <- read_csv("http://nhorton.people.amherst.edu/is5/data/Hopkins_Forest.csv") %>%
  clean_names()

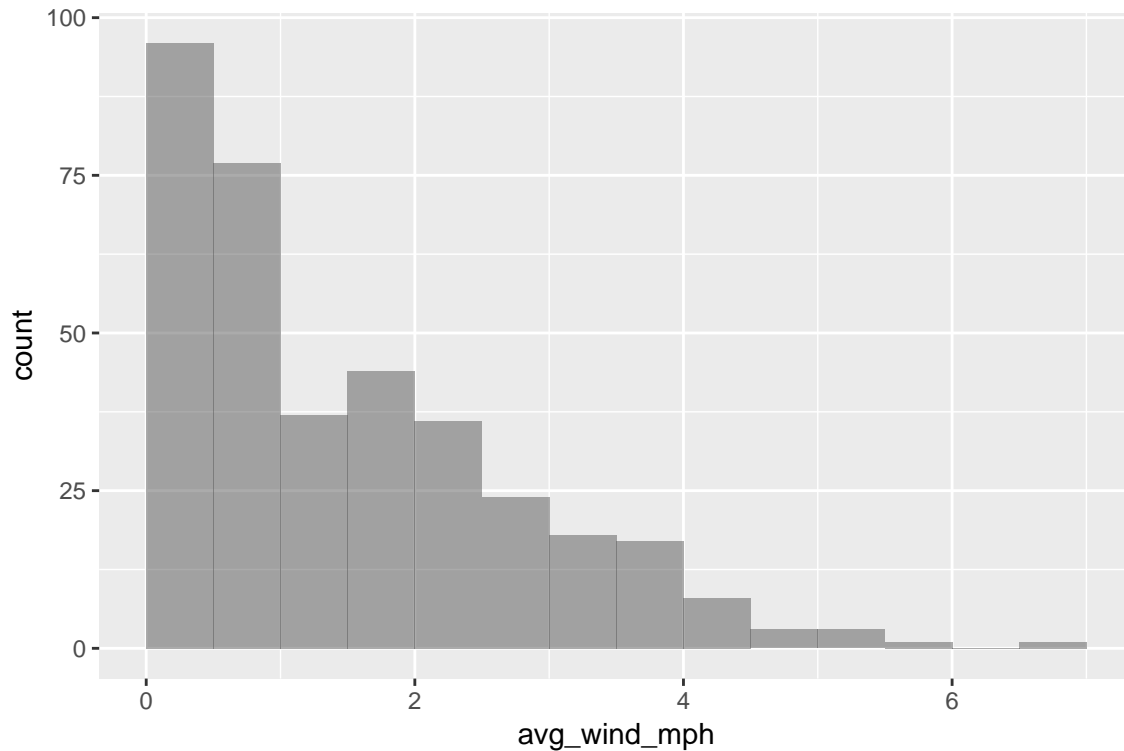
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   Date = col_character(),
##   Year = col_integer(),
##   Month = col_integer(),
##   Day = col_integer(),
##   `Day of Year` = col_integer(),
##   `Max Sol Rad (w/m^2)` = col_integer(),
##   `Min Sol Rad (w/m^2)` = col_integer(),
##   `Total Sol Rad (w/m^2)` = col_integer(),
##   `Min Wind (mph)` = col_integer(),
##   `Max Barom (mb)` = col_integer(),
##   `Min Barom (mb)` = col_integer()
## )

## See spec(...) for full column specifications.
names(HopkinsForest)

## [1] "date"          "year"          "month"
## [4] "day"           "day_of_year"   "avg_temp_c"
## [7] "max_temp_c"    "min_temp_c"    "avg_temp_f"
```

```
## [10] "max_temp_f"      "min_temp_f"      "avg_rel_hum_percent"
## [13] "max_rel_hum_percent" "min_rel_hum_percent" "avg_sol_rad_w_m_2"
## [16] "max_sol_rad_w_m_2" "min_sol_rad_w_m_2" "total_sol_rad_w_m_2"
## [19] "avg_wind_mph"     "max_wind_mph"     "min_wind_mph"
## [22] "avg_barom_mb"     "max_barom_mb"     "min_barom_mb"
## [25] "precip_in"        "deep_well_ft"     "shallow_well_ft"
## [28] "x80_cm_soil_c"    "x10_cm_soil_c"
```

```
gf_histogram(~ avg_wind_mph, data = HopkinsForest, binwidth = 0.5, center = 0.25)
```



```
favstats(~ avg_wind_mph, data = HopkinsForest)
```

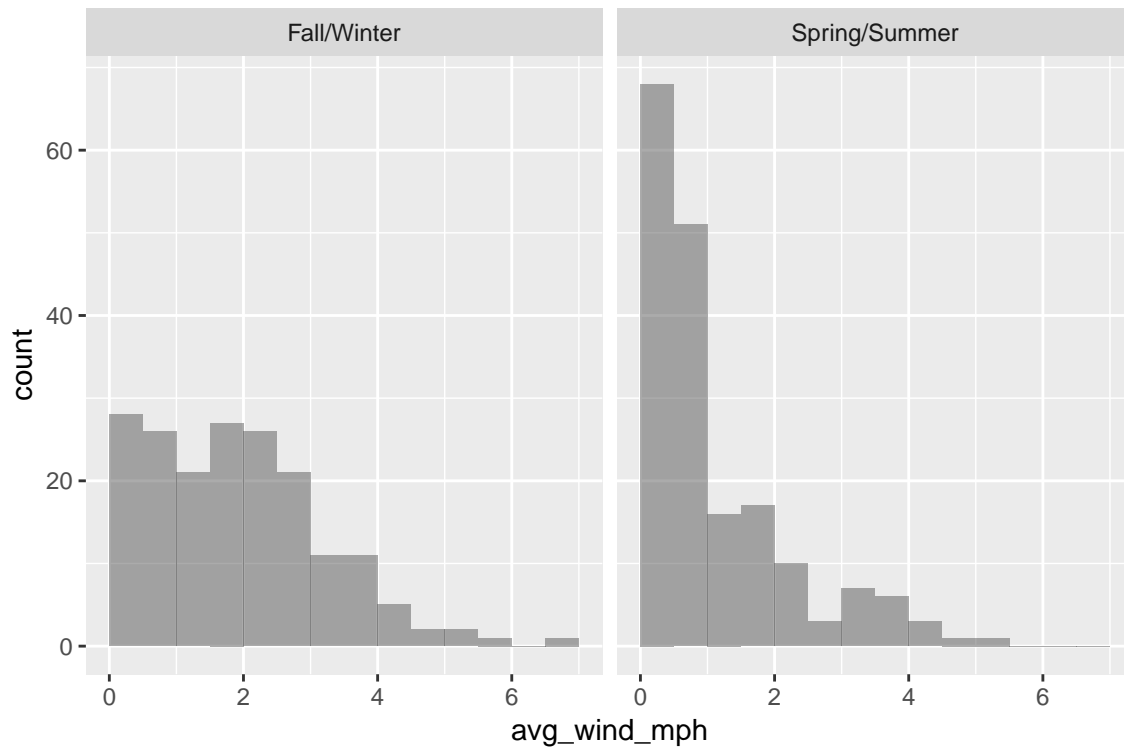
```
## min   Q1 median   Q3 max     mean      sd   n missing
##    0 0.46   1.12 2.28 6.73 1.507808 1.260161 365      0
```

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.

Section 4.1: Displays for Comparing Groups

Histograms

```
HopkinsForest <- HopkinsForest %>%
  mutate(catmonth = ifelse(month <= 9 & month >= 4, "Spring/Summer", "Fall/Winter"))
gf_histogram(~ avg_wind_mph, data = HopkinsForest, binwidth = 0.5, center = 0.25) %>%
  gf_facet_wrap(~ catmonth)
```



```
favstats(~ avg_wind_mph | catmonth, data = HopkinsForest)
```

```
##      catmonth  min   Q1 median   Q3  max    mean      sd  n missing
## 1  Fall/Winter 0.02 0.84  1.72 2.6575 6.73 1.904176 1.287233 182      0
## 2 Spring/Summer 0.00 0.35  0.71 1.6150 5.47 1.113607 1.102176 183      0
```

Example 4.1: Comparing Groups with Stem-And-Leaf

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

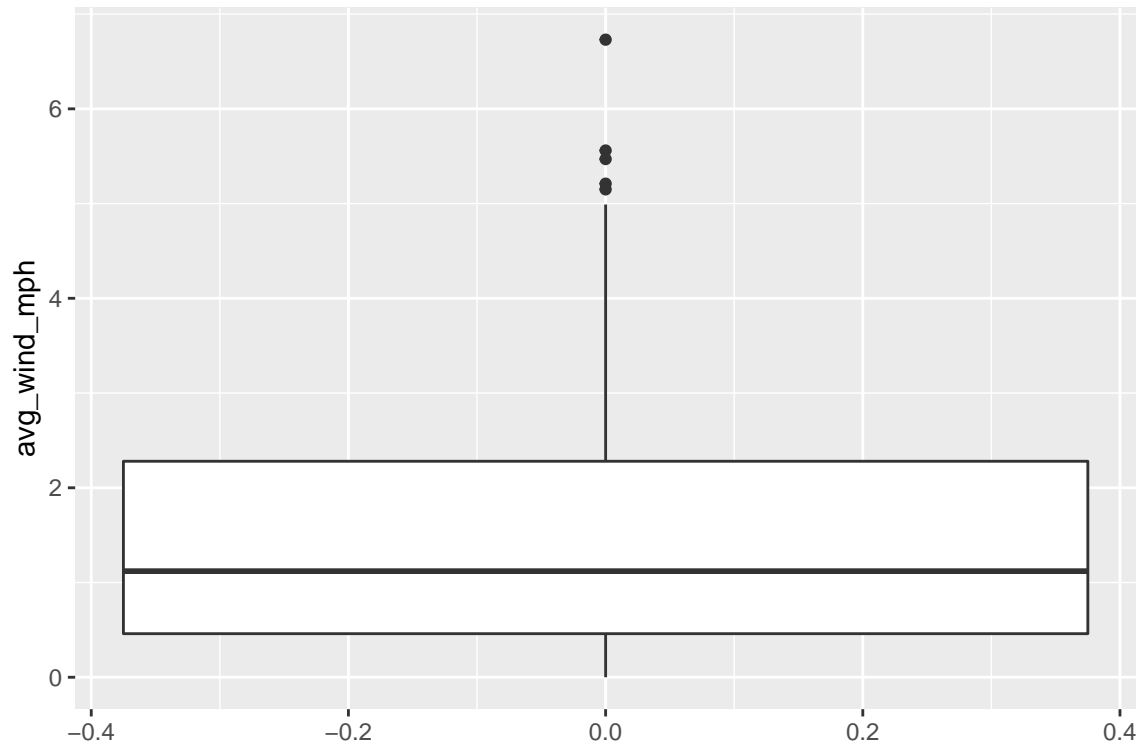
```
## Parsed with column specification:
## cols(
##   State = col_character(),
##   Nest.Egg.Index = col_double(),
##   Region = col_character()
## )
```

```
with(NestEgg, stem(nest_egg_index))
```

```
##
## The decimal point is 1 digit(s) to the right of the |
##
## 8 | 57789
## 9 | 0123344
## 9 | 667777888899
## 10 | 0012233333344
## 10 | 5566779
## 11 | 122444
```

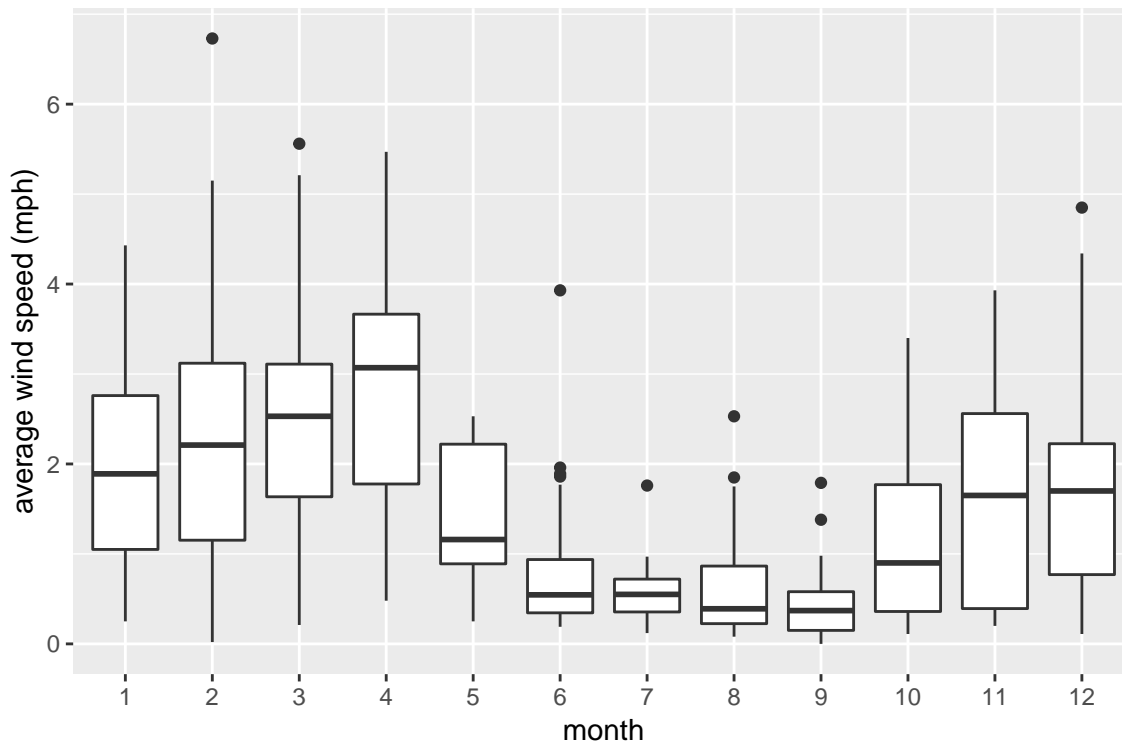
Boxplots

```
gf_boxplot(~ avg_wind_mph, data = HopkinsForest) # or gf_boxplot(X ~ 1)
```



As noted in the book, boxplots are most useful to compare distributions. Here we replicate the single boxplot from page 98 but don't recommend the use of single boxplots.

```
gf_boxplot(avg_wind_mph ~ as.factor(month), data = HopkinsForest) %>%  
  gf_labs(x = "month", y = "average wind speed (mph)")
```



Here we replicate the results from Figure 4.3 (page 99).

We use `as.factor()` to convert a variable into a factor.

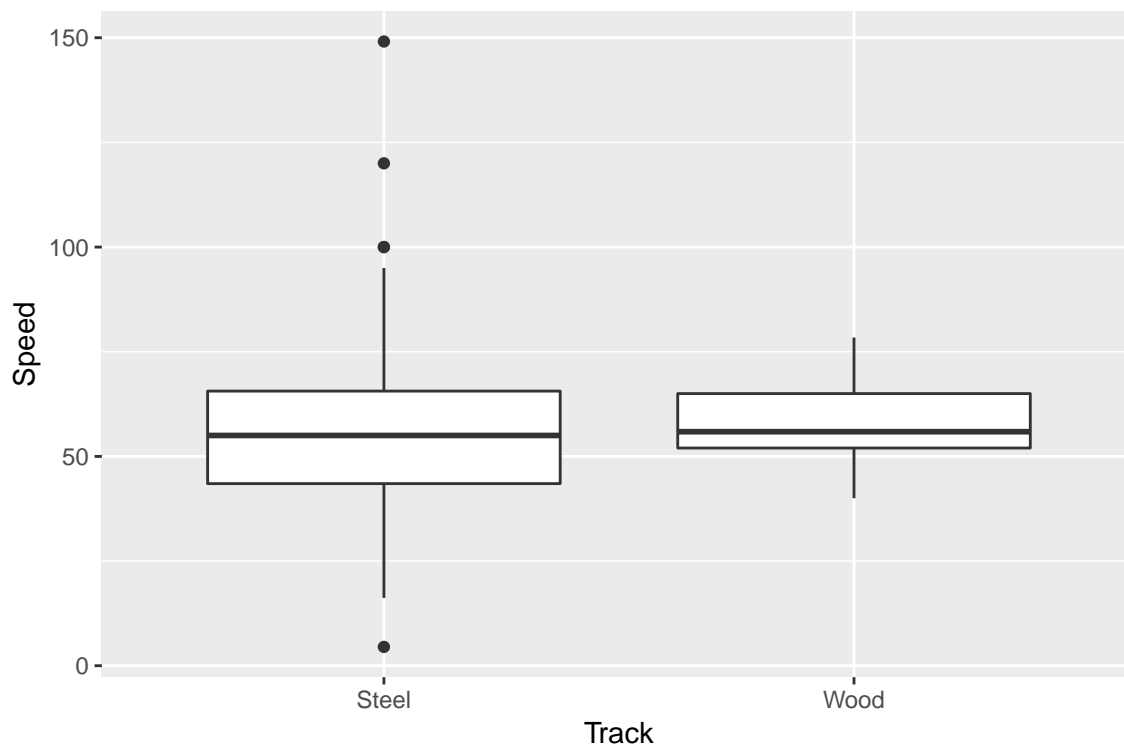
Example 4.2: Comparing Groups with Boxplots

```
Coasters <- read_csv("http://nhorton.people.amherst.edu/is5/data/Coasters_2015.csv")
```

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

```
## cols(  
##   Name = col_character(),  
##   Park = col_character(),  
##   Track = col_character(),  
##   Speed = col_double(),  
##   Height = col_double(),  
##   Drop = col_double(),  
##   Length = col_double(),  
##   Duration = col_integer(),  
##   Inversions = col_integer()  
## )
```

```
gf_boxplot(Speed ~ Track, data = Coasters)
```



Step-By-Step Example: Comparing Groups

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

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

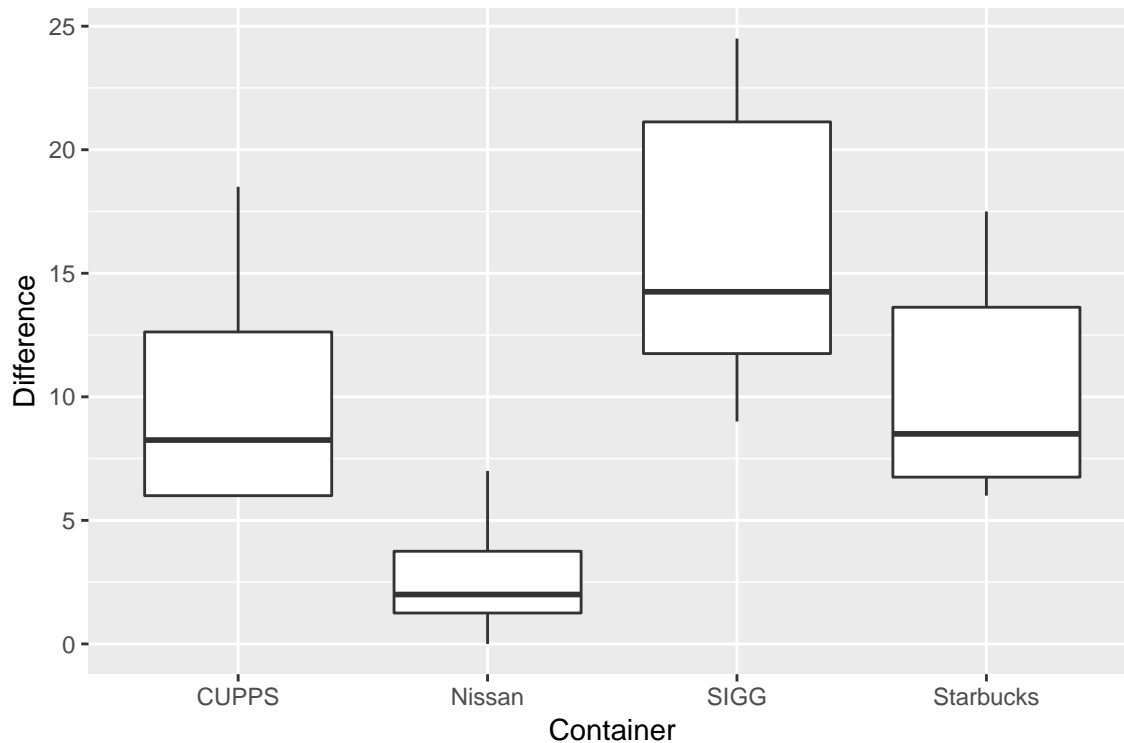
```
## cols(  
##   Difference = col_double(),  
##   Container = col_character()  
## )
```

```
## )
```

```
favstats(~ Difference | Container, data = Cups)
```

```
##   Container min    Q1 median    Q3 max    mean      sd n missing
## 1   CUPPS    6  6.00  8.25 12.625 18.5 10.1875 5.202592 8      0
## 2   Nissan    0  1.25  2.00  3.750  7.0  2.7500 2.507133 8      0
## 3    SIGG    9 11.75 14.25 21.125 24.5 16.0625 5.900590 8      0
## 4 Starbucks  6  6.75  8.50 13.625 17.5 10.2500 4.551295 8      0
```

```
gf_boxplot(Difference ~ Container, data = Cups)
```

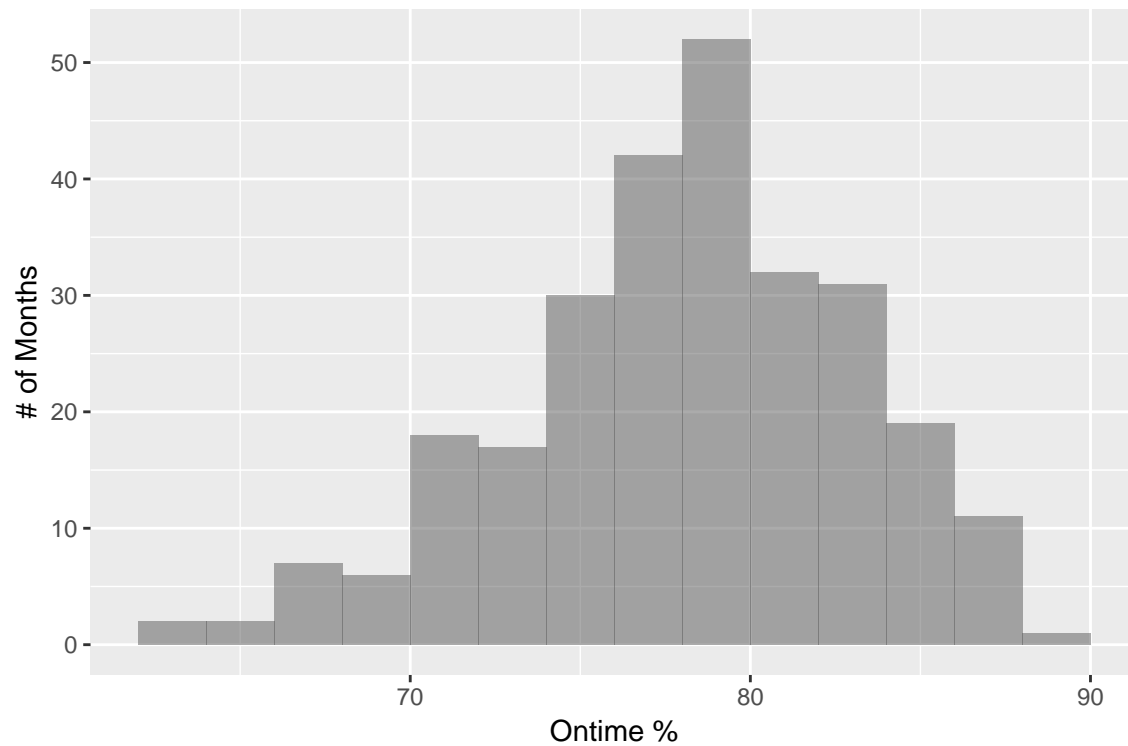


Just Checking

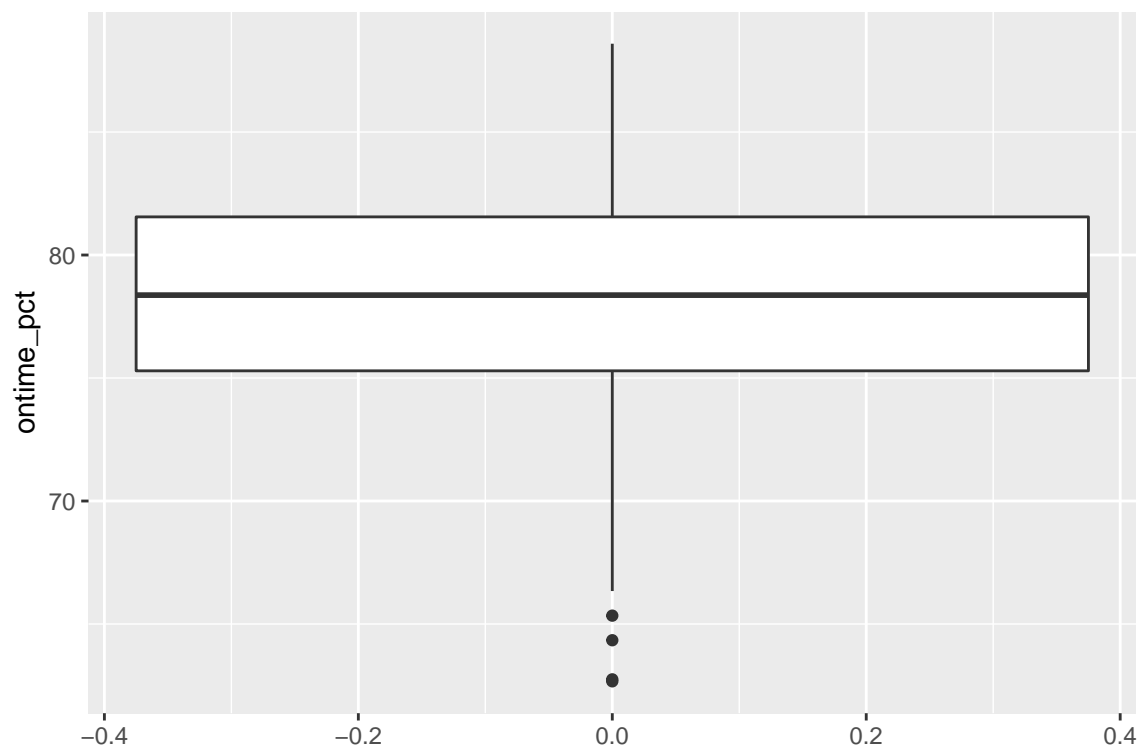
```
Flights <- read_csv("http://nhorton.people.amherst.edu/is5/data/Flights_on_time_2016.csv") %>%
  clean_names()
```

```
## Parsed with column specification:
## cols(
##   Year = col_double(),
##   Month = col_character(),
##   Onetime.Arrivals = col_integer(),
##   Ontime.pct = col_double(),
##   Arrival.Delays = col_integer(),
##   Delayed.pct = col_double(),
##   Flights.Cancelled = col_integer(),
##   Cancelled.Pct = col_double(),
##   Diverted = col_integer(),
##   Flight.Operations = col_integer()
## )
```

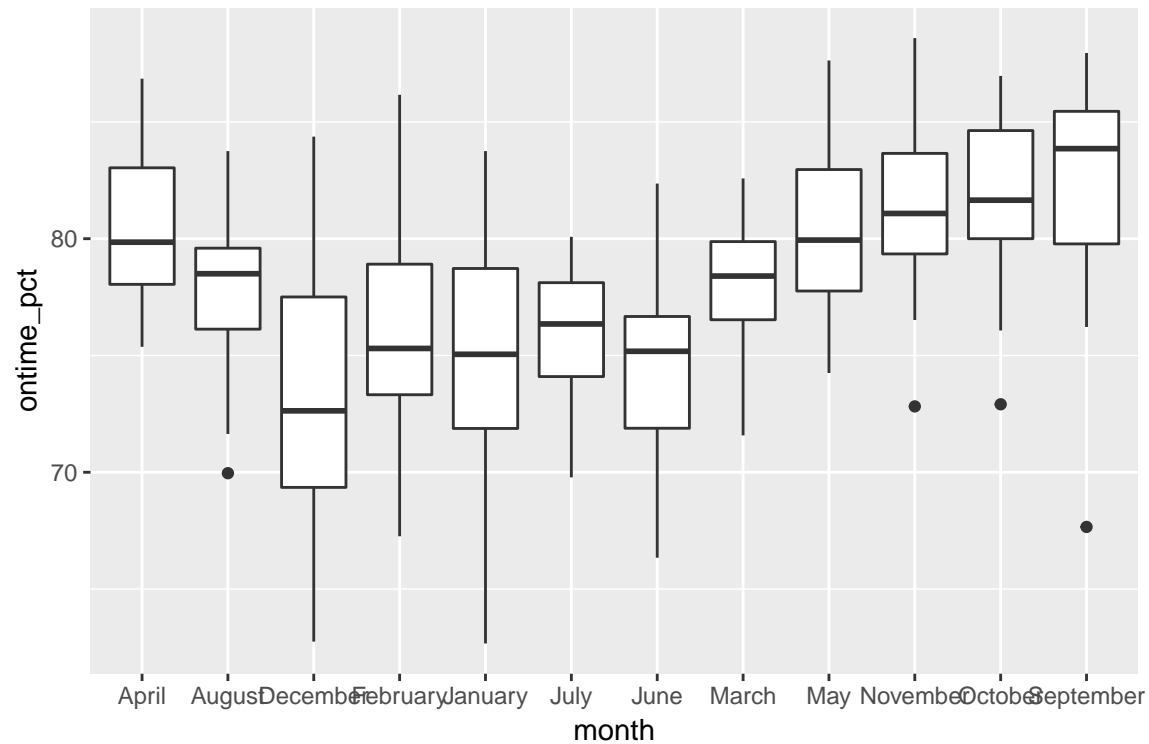
```
gf_histogram(~ ontime_pct, data = Flights, binwidth = 2, ylab = "# of Months", xlab = "Ontime %",
             center = 1)
```



```
gf_boxplot(~ ontime_pct, data = Flights)
```



```
gf_boxplot(ontime_pct ~ month, data = Flights)
```

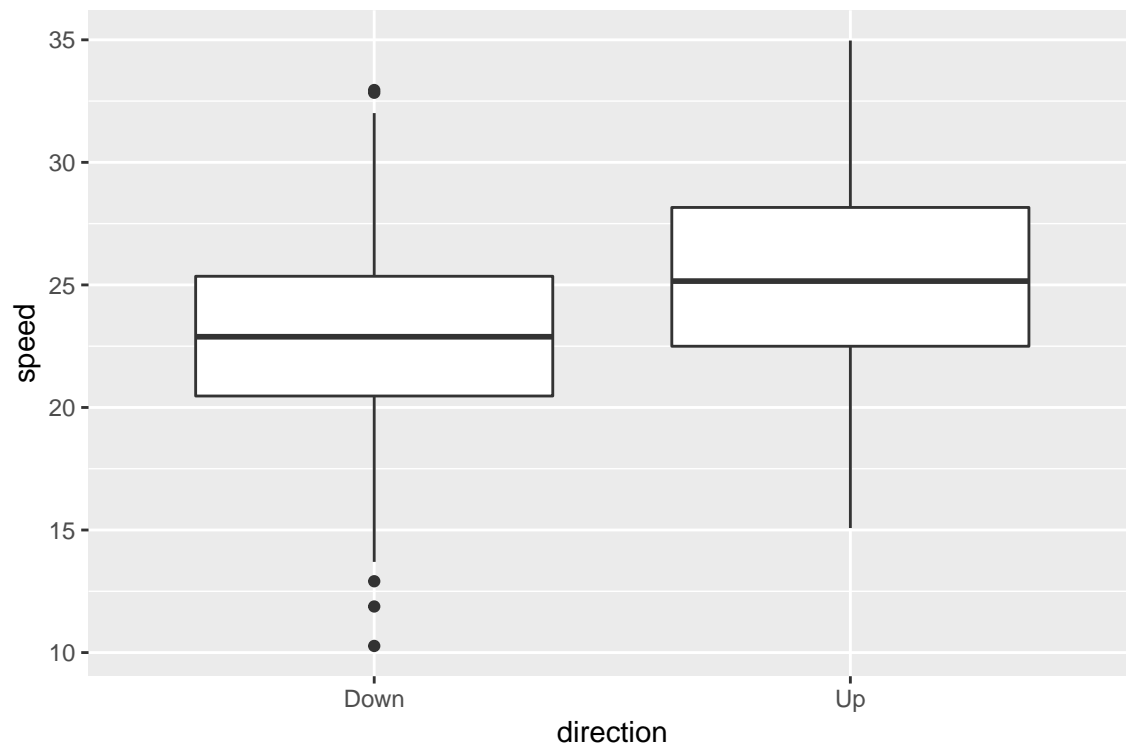


Random Matters

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

```
## Parsed with column specification:
## cols(
##   direction = col_character(),
##   speed = col_double()
## )
```

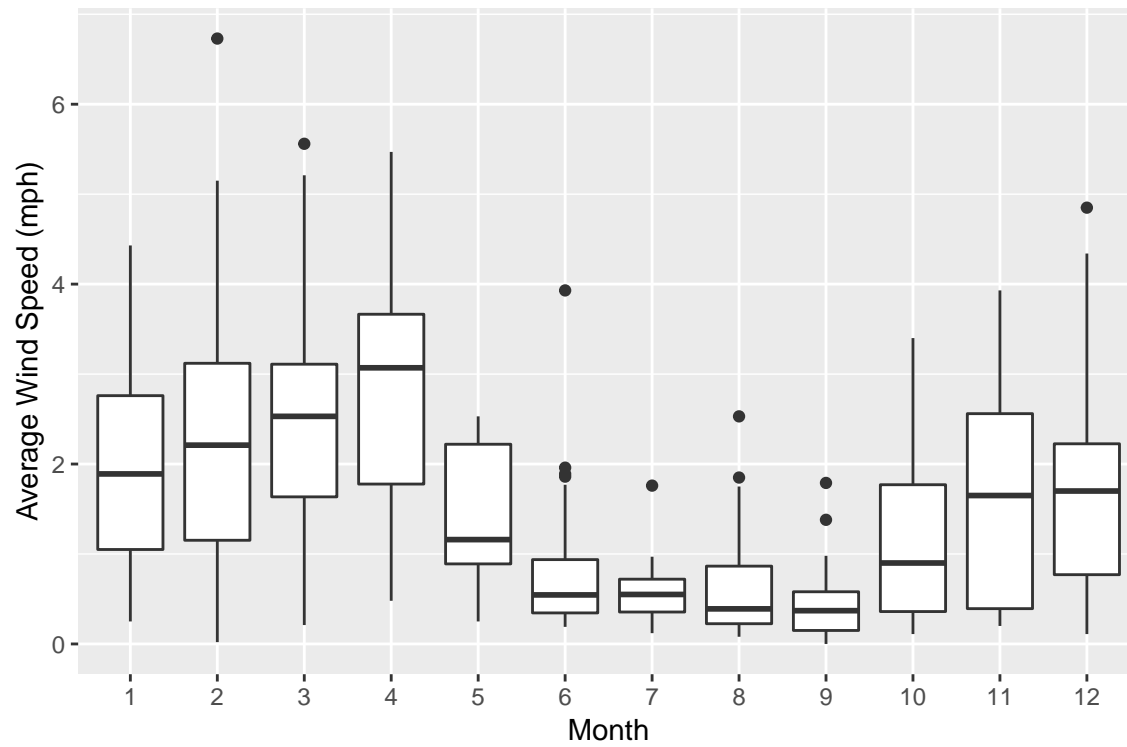
```
gf_boxplot(speed ~ direction, data = CarSpeeds)
```

See histogram on page 103.

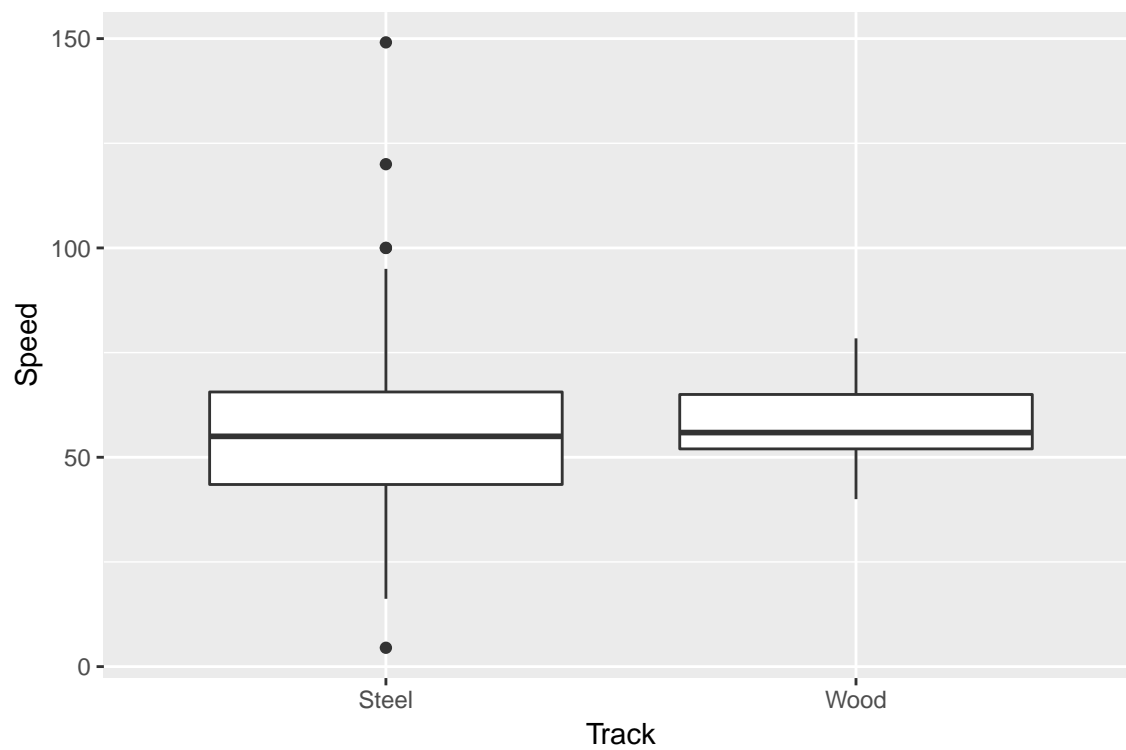
Section 4.2: Outliers

```
gf_boxplot(avg_wind_mph ~ as.factor(month), data = HopkinsForest) %>%  
  gf_labs(x = "Month", y = "Average Wind Speed (mph)")
```



Example 4.3: Checking Out the Outliers

```
gf_boxplot(Speed ~ Track, data = Coasters)
```



Section 4.3: Re-Expressing Data: A First Look

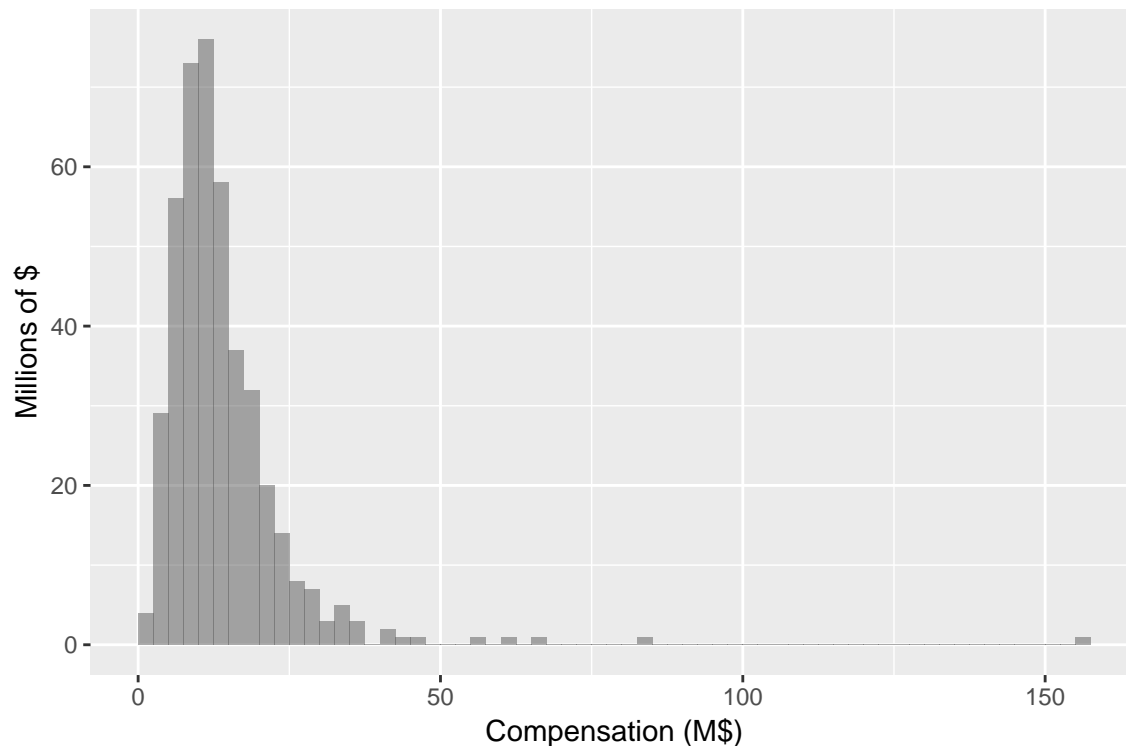
Re-Expressing to Improve Symmetry

```
CEOComp <- read_csv("http://nhorton.people.amherst.edu/is5/data/CEO_Compensation_2014.csv") %>%  
  clean_names()
```

```
## Parsed with column specification:  
## cols(  
##   Employer = col_character(),  
##   CEO = col_character(),  
##   CEO_Compensation = col_integer(),  
##   Median_Worker_Comp = col_integer(),  
##   Ratio = col_integer(),  
##   Company_Rating = col_double(),  
##   `CEO_Compensation_($M)` = col_double()  
## )
```

Figure 4.6, page 105

```
gf_histogram(~ ceo_compensation_m, xlab = "Compensation (M$)", ylab = "Millions of $",  
  data = CEOComp, binwidth = 2.5, center = 2.5/2)
```



```
gf_boxplot(~ ceo_compensation_m, xlab = "Compensation (M$)", ylab = "Millions of $",  
  data = CEOComp)
```

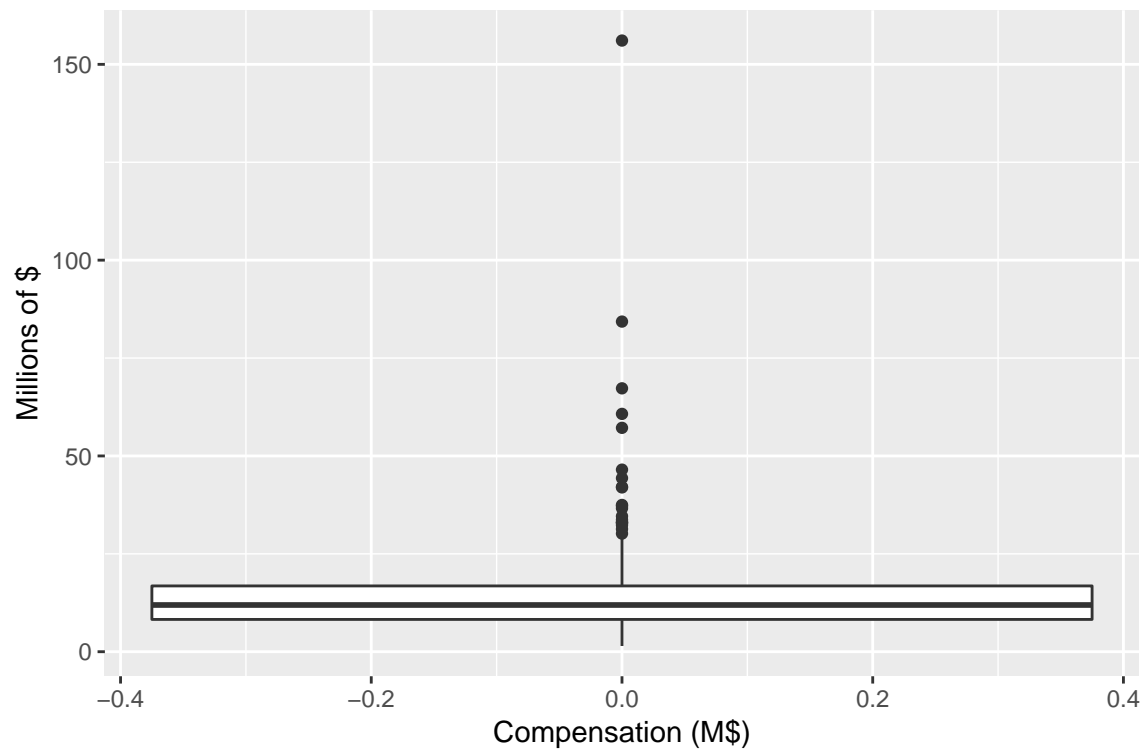
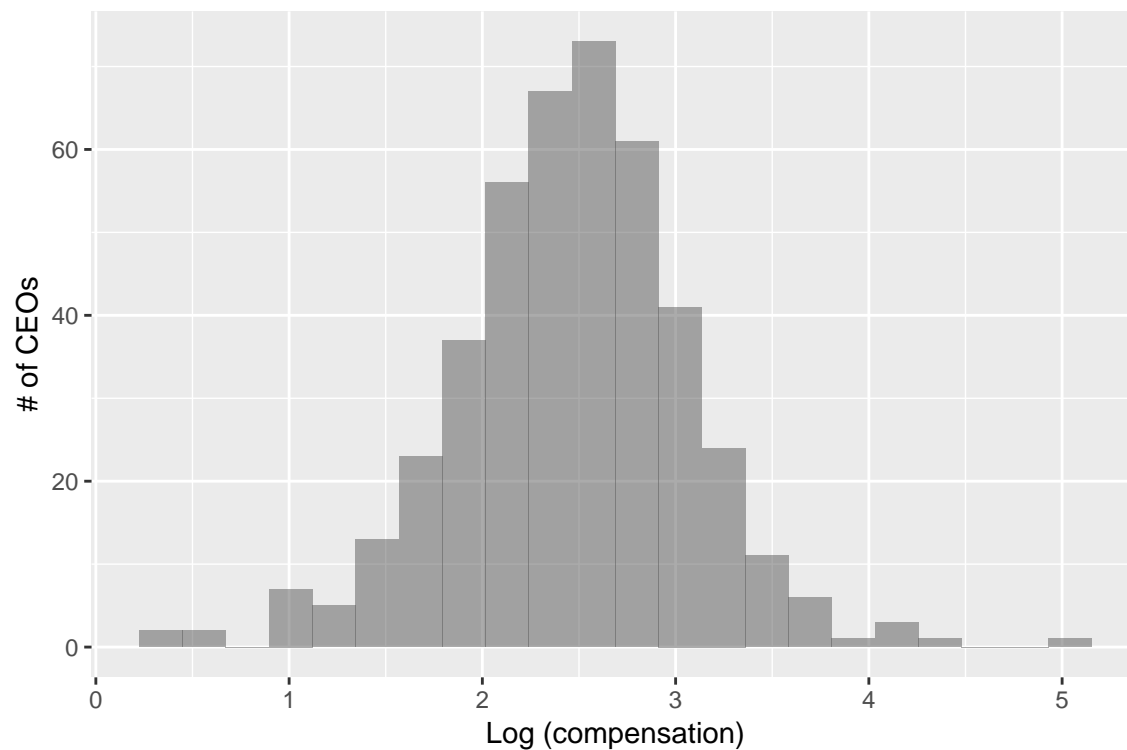


Figure 4.7, page 106
`gf_histogram(~ log(ceo_compensation_m), xlab = "Log (compensation)", ylab = "# of CEOs",
 data = CEOComp, binwidth = .224, center = .112)`



Re-Expression to Equalize Spread Across Groups

```
## Parsed with column specification:
## cols(
##   cotinine = col_double(),
##   smoke_exposure = col_character()
## )
```

Box plot showing Cotinine (ng/ml) levels across three groups: ETS, No ETS, and Smokers. The y-axis represents Cotinine concentration in ng/ml, ranging from 0 to 1000. The x-axis represents Smoke Exposure categories. The plot shows that cotinine levels are significantly higher in smokers compared to those exposed to environmental tobacco smoke (ETS) or no ETS.

Smoke Exposure	Median (ng/ml)	Q1 (ng/ml)	Q3 (ng/ml)	Min (ng/ml)	Max (ng/ml)	Outliers (ng/ml)
ETS	~0	~0	~0	~0	~0	~50, ~100
No ETS	~0	~0	~0	~0	~0	~10
Smokers	~50	~10	~180	~0	~400	~550, ~600, ~700, ~980

13

