

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

Nicholas Horton (nhorton@amherst.edu)

2025-01-08

## Table of contents

|  |    |
|--|----|
| Introduction and background . . . . .                          | 1  |
| Chapter 4: Understanding and Comparing Distributions . . . . . | 2  |
| Section 4.1: Displays for Comparing Groups . . . . .           | 3  |
| Section 4.3: Re-Expressing Data: A First Look . . . . .        | 11 |

## 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 Quarto 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>.

We begin by loading packages that will be required for our analyses.

```
library(mosaic)
library(tidyverse)
```

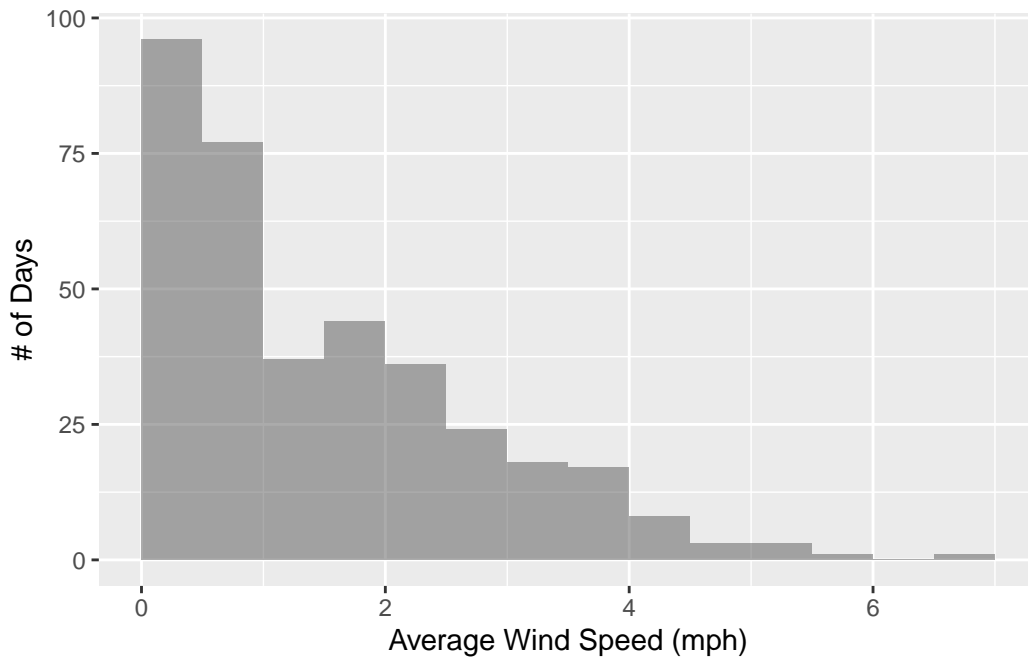
## Chapter 4: Understanding and Comparing Distributions

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

By default, `read_csv()` prints the variable names. We suppressed these using the `message = FALSE` code chunk option to save space and improve readability. 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). You can use the `names()` function to check the cleaned names.

```
# Figure 4.1, page 96
gf_histogram(~ avg_wind_mph,
  data = HopkinsForest,
  xlab = "Average Wind Speed (mph)",
  ylab = "# of Days",
  binwidth = 0.5,
  center = 0.25
)
```



```
df_stats(~ avg_wind_mph, data = HopkinsForest) # an alternative version of "favstats()"
```

|   | response     | min | Q1   | median | Q3   | max  | mean     | sd       | n   | missing |
|---|--------------|-----|------|--------|------|------|----------|----------|-----|---------|
| 1 | avg_wind_mph | 0   | 0.46 | 1.12   | 2.28 | 6.73 | 1.507808 | 1.260161 | 365 | 0       |

## Section 4.1: Displays for Comparing Groups

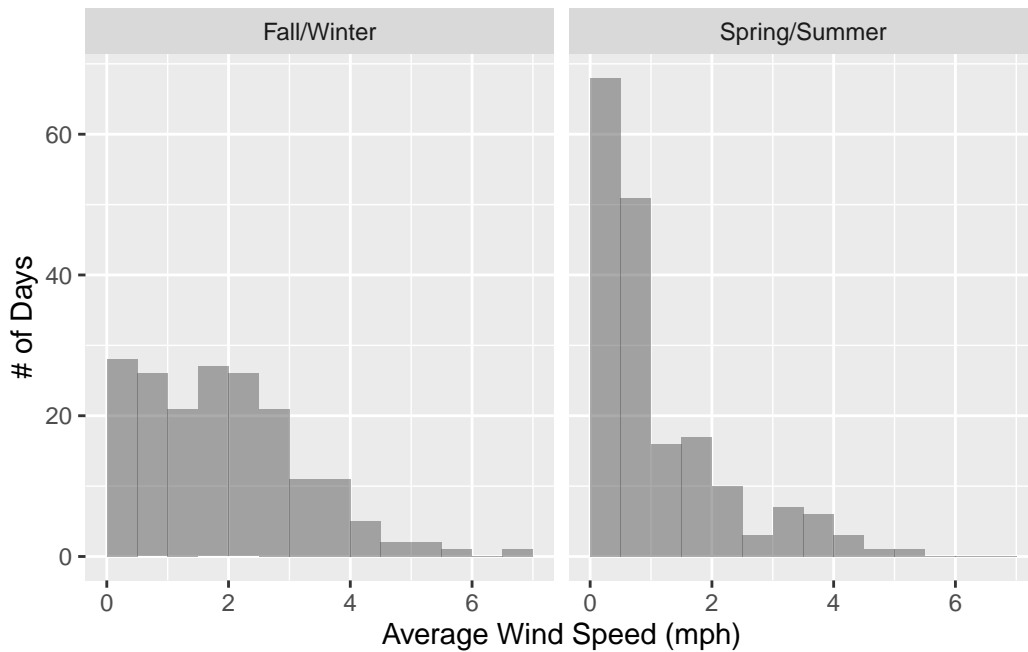
### Histograms

We began by creating a new month to categorize the dates.

```
HopkinsForest <- HopkinsForest |>
  mutate(catmonth = ifelse(
    month <= 9 & month >= 4,
    "Spring/Summer",
    "Fall/Winter")
  )
```

```
# Figure 4.2, page 96
gf_histogram(~ avg_wind_mph,
  data = HopkinsForest, binwidth = 0.5, center = 0.25,
  xlab = "Average Wind Speed (mph)", ylab = "# of Days"
```

```
) |>
  gf_facet_wrap(~ catmonth)
```



```
df_stats(avg_wind_mph ~ catmonth, data = HopkinsForest)
```

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

### Example 4.1: Comparing Groups with Stem-And-Leaf

We begin by reading in the data.

```
# Figure 4.1, page 97
NestEgg <- read_csv("http://nhorton.people.amherst.edu/is5/data/Nest_Egg_Index.csv") |>
  janitor::clean_names()
with(NestEgg, stem(nest_egg_index)) # or stem(NestEgg$nest_egg_index)
```

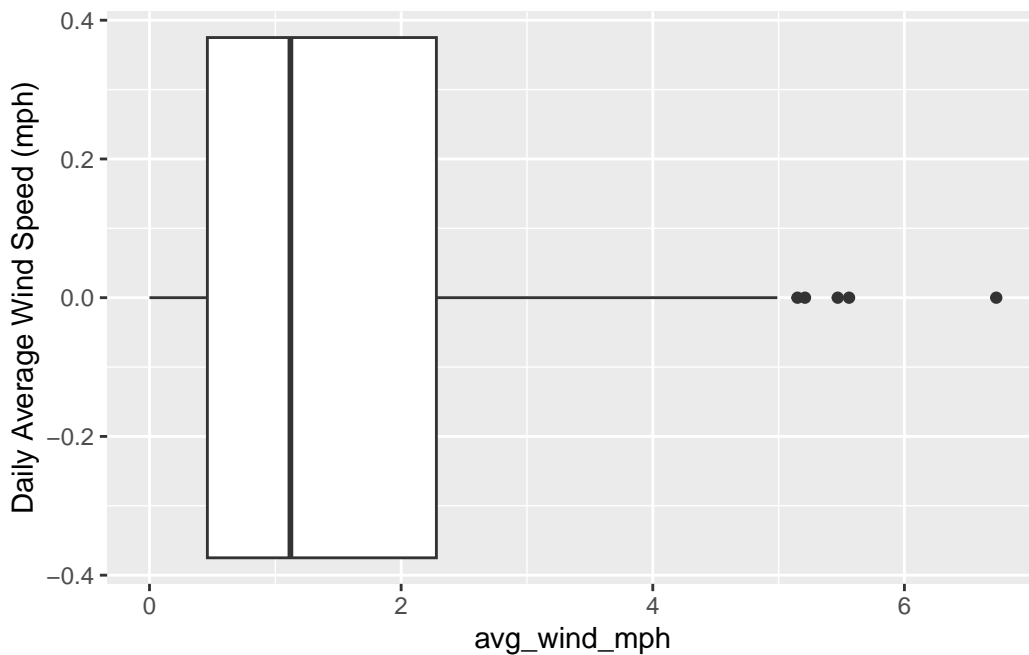
The decimal point is 1 digit(s) to the right of the |

```
8 | 57789
9 | 0123344
9 | 667777888899
10 | 0012233333344
10 | 5566779
11 | 122444
```

## Boxplots

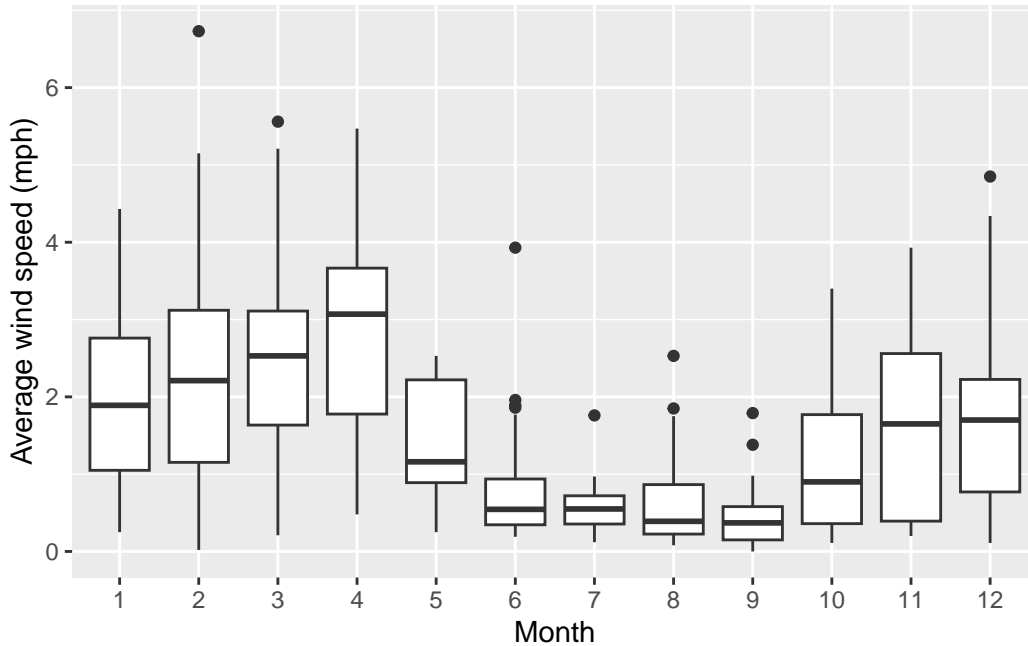
As noted in the book, boxplots are most useful to compare distributions. Below, we have replicated the single boxplot from page 98.

```
# Step 4 on page 98
gf_boxplot(~ avg_wind_mph, data = HopkinsForest) |> # or gf_boxplot(X ~ 1)
gf_labs(y = "Daily Average Wind Speed (mph)")
```



The use of single boxplots isn't recommended. Instead, one can make comparisons more easily by placing boxplots side by side with the following code:

```
# Figure 4.3, page 99
gf_boxplot(avg_wind_mph ~ as.factor(month), data = HopkinsForest) |>
  gf_labs(x = "Month", y = "Average wind speed (mph)")
```



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

We also use `gf_labs()` to clean up the code for the first line and improve readability.

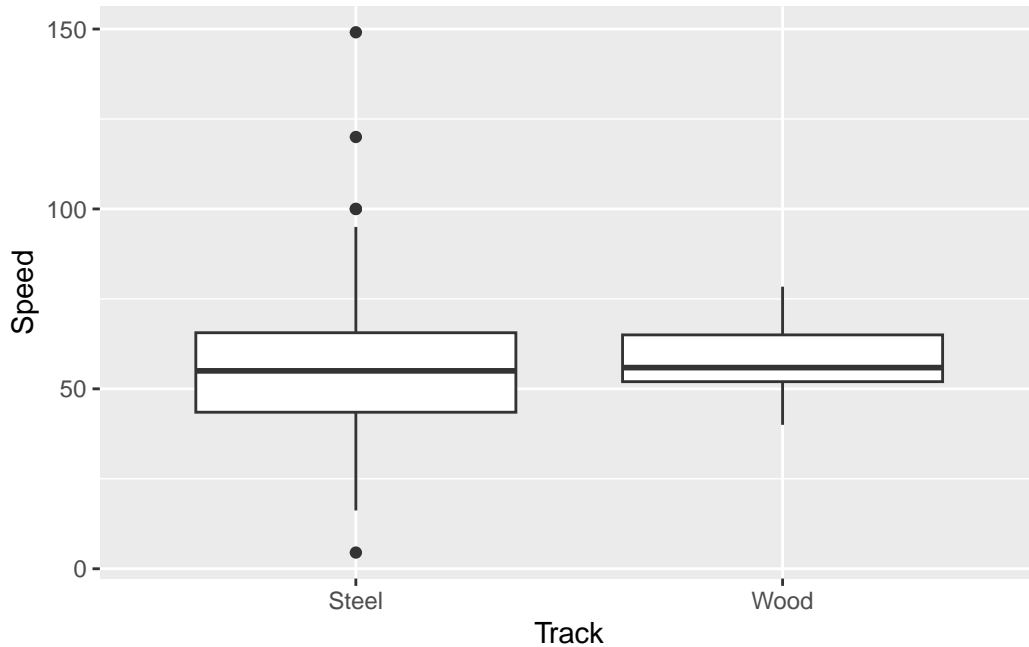
Here we use the mosaic modeling language to specify the variables. The `~` symbol is used to separate the response variable from the explanatory variable.

As a general form, `GOAL(Y ~ X)` carries out a specific goal for Y as a function of X.

### Example 4.2: Comparing Groups with Boxplots

We begin by reading in the data.

```
# Example 4.2, page 99
Coasters <- read_csv("http://nhorton.people.amherst.edu/is5/data/Coasters_2015.csv")
gf_boxplot(Speed ~ Track, data = Coasters)
```



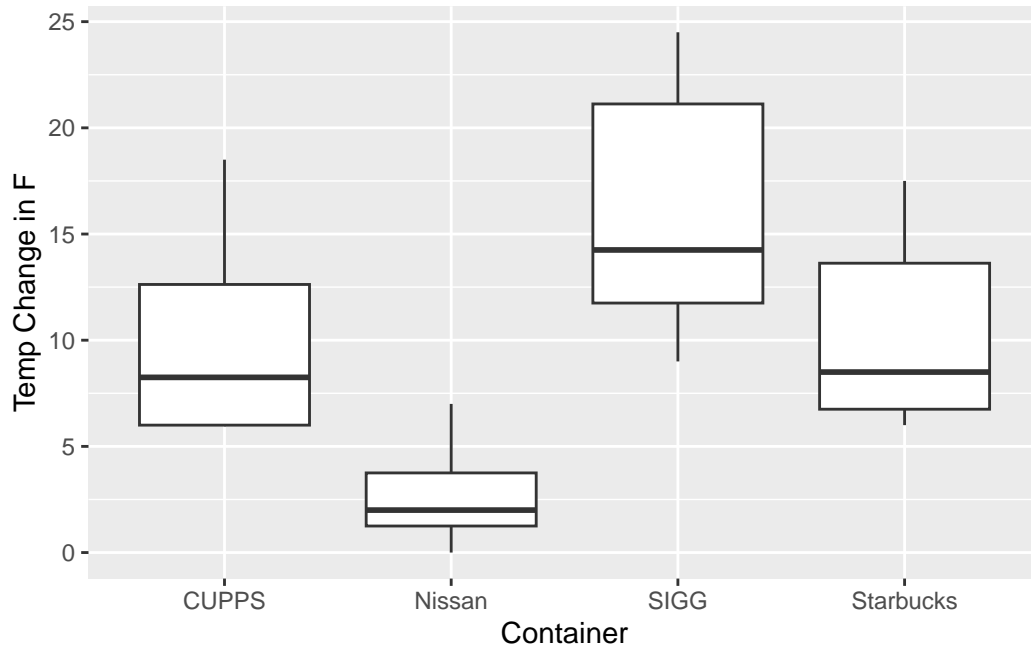
### Step-By-Step Example: Comparing Groups

We begin by reading in the data.

```
Cups <- read_csv("http://nhorton.people.amherst.edu/is5/data/Cups.csv")
df_stats(Difference ~ Container, data = Cups)
```

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

```
# Mechanics, page 101
gf_boxplot(Difference ~ Container, data = Cups, ylab = "Temp Change in F")
```



### Just Checking

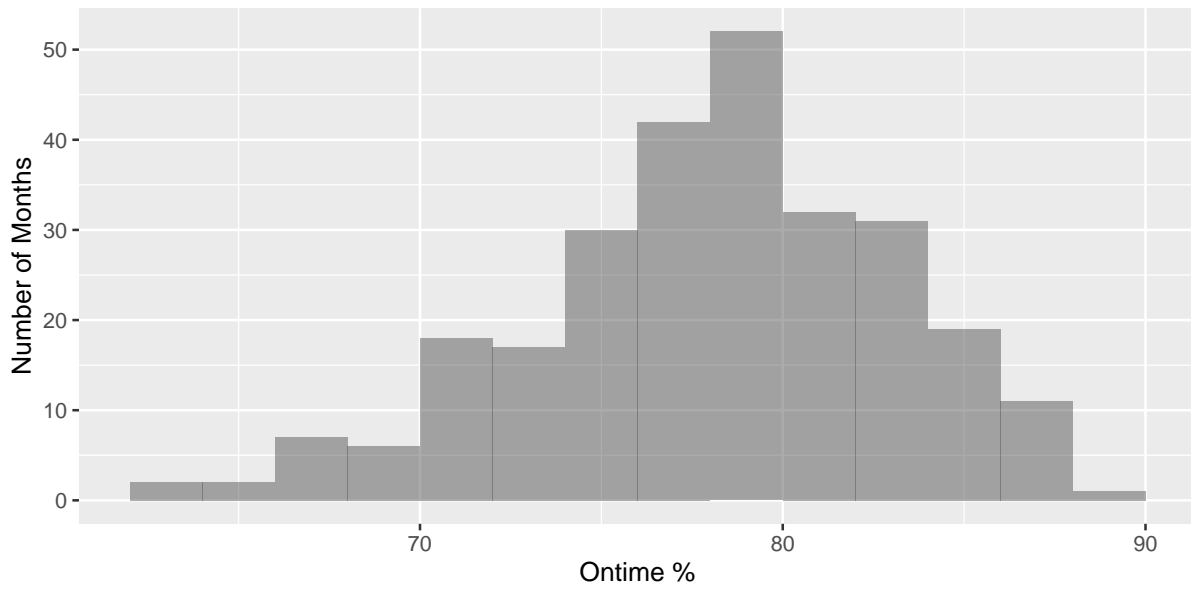
We begin by reading in the data.

```
Flights <-
  read_csv("http://nhorton.people.amherst.edu/is5/data/Flights_on_time_2016.csv") |>
  janitor::clean_names()
# Let's improve the ordering of the months (by default they are alphabetical!)
Flights <- Flights |>
  mutate(month = forcats::fct_relevel(
    month,
    "January", "February", "March", "April",
    "May", "June", "July", "August",
    "September", "October", "November", "December"
  )
)
```

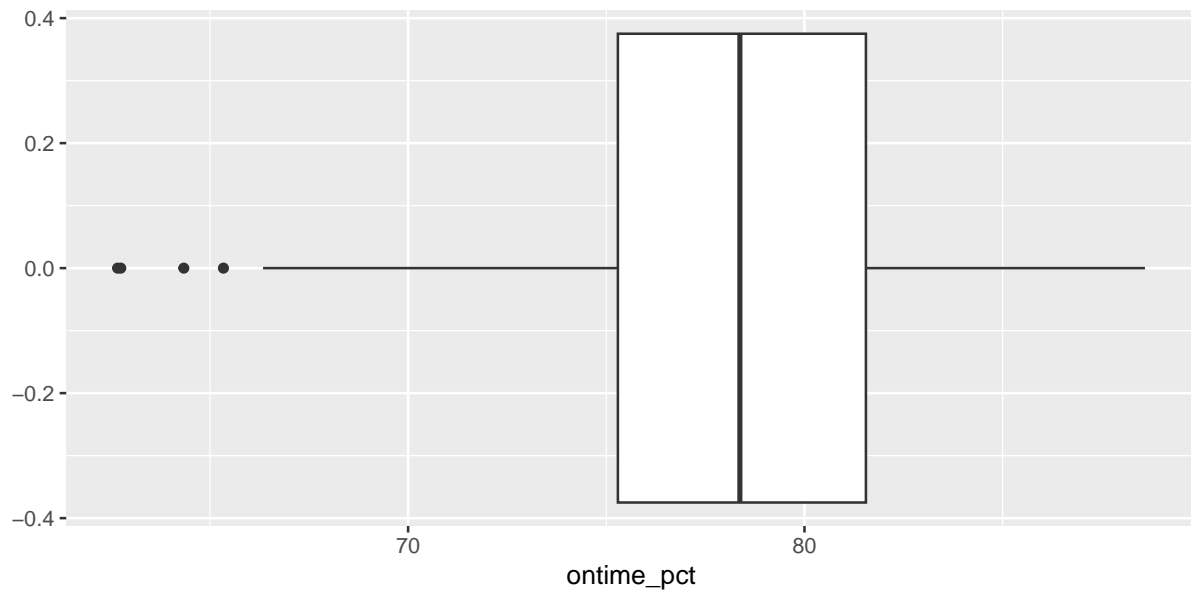
Here we use the `fct_relevel()` function from the `forcats` package to reorder the months in the dataset (the default is that the months are ordered alphabetically, which isn't very helpful). This function is a very useful idiom to remember when you want to reorder factor levels.

```
# Bureau of Transportation Statistics, page 101
gf_histogram(~ ontime_pct, data = Flights, binwidth = 2, center = 1) |>
  gf_labs(x = "Ontime %", y = "Number of Months")
```

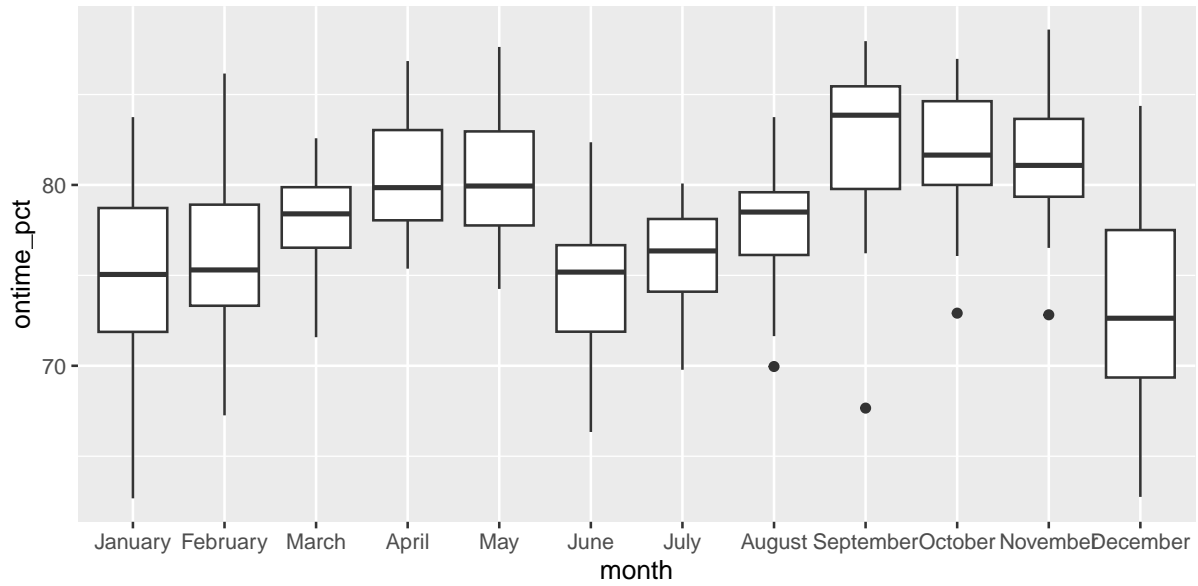




```
gf_boxplot(~ ontime_pct, data = Flights)
```



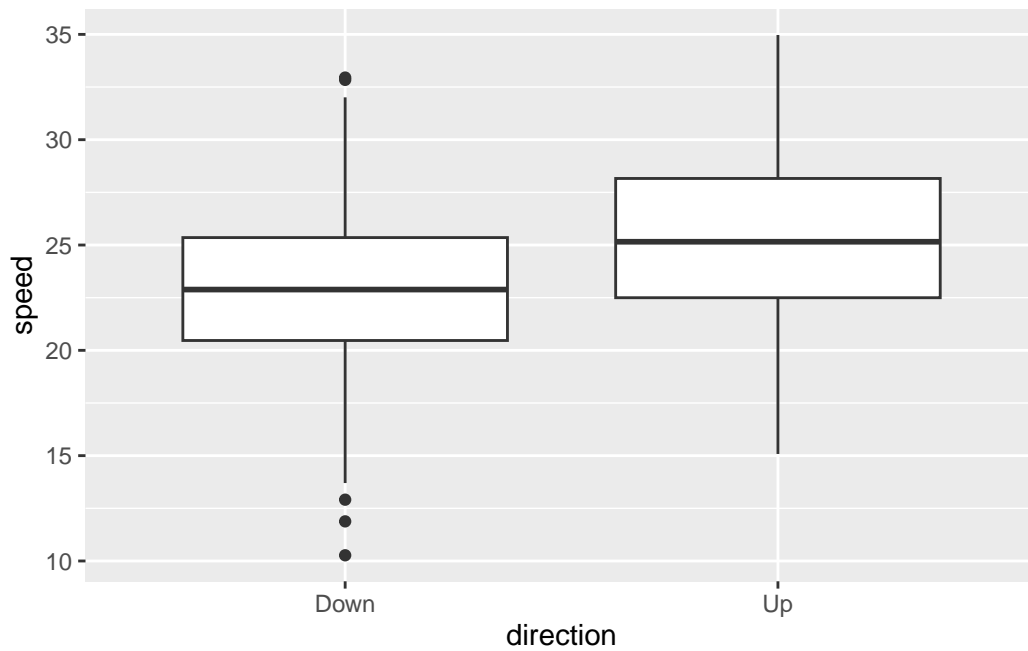
```
gf_boxplot(ontime_pct ~ month, data = Flights) # now they are in order!
```



## Random Matters

We begin by reading in the data.

```
# Figure 4.4, page 102
CarSpeeds <- read_csv("http://nhorton.people.amherst.edu/is5/data/Car_speeds.csv")
gf_boxplot(speed ~ direction, data = CarSpeeds)
```



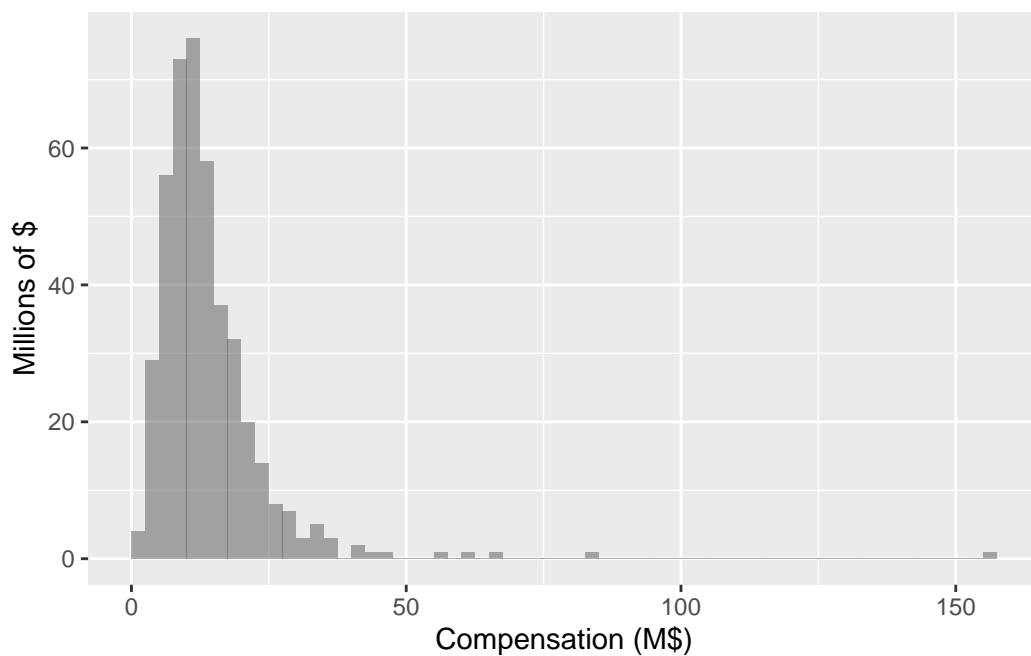
## Section 4.3: Re-Expressing Data: A First Look

### Re-Expressing to Improve Symmetry

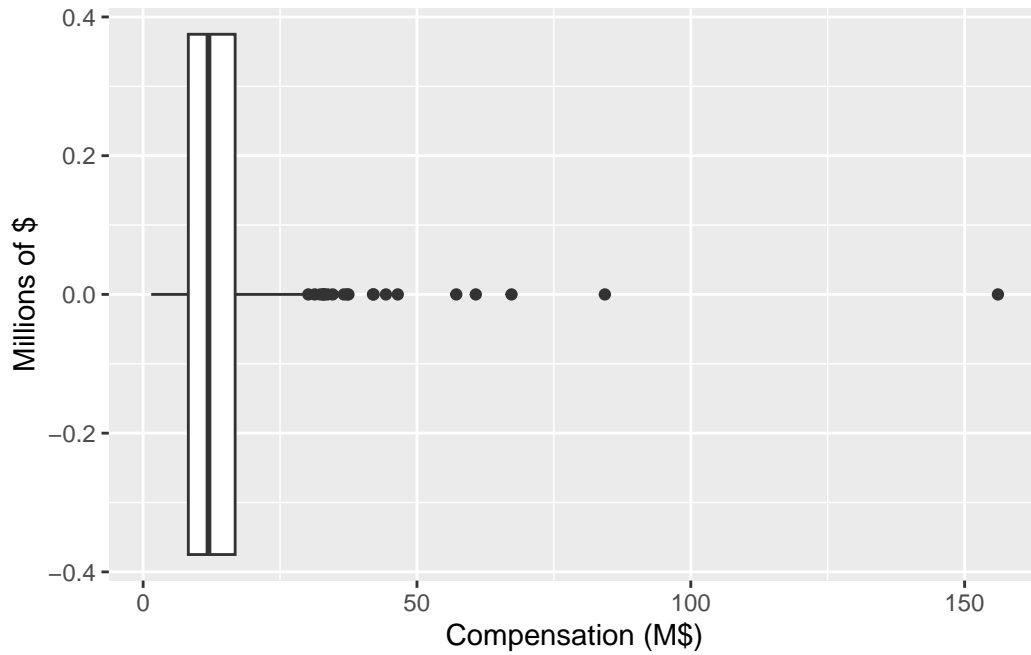
We begin by reading in the data.

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

```
# Figure 4.6, page 105  
gf_histogram(~ ceo_compensation_m, data = CEOComp, binwidth = 2.5, center = 2.5 / 2) |>  
  gf_labs(x = "Compensation (M$)", y = "Millions of $")
```

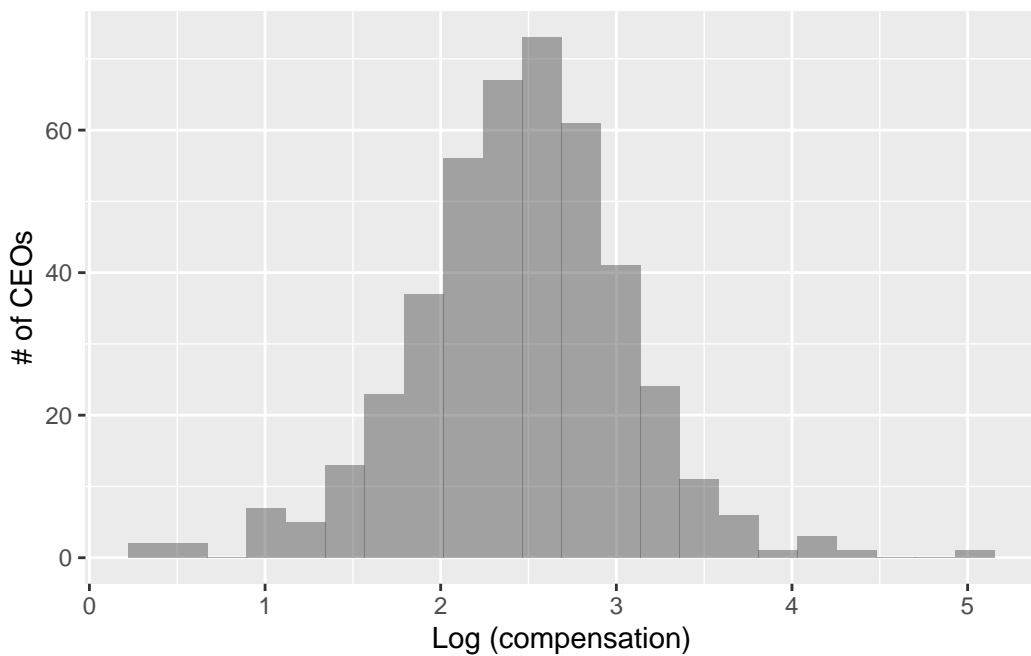


```
gf_boxplot(~ ceo_compensation_m, data = CEOComp) |>  
  gf_labs(x = "Compensation (M$)", y = "Millions of $")
```



# Figure 4.7, page 106

```
gf_histogram(~ log(ceo_compensation_m), data = CEOComp, binwidth = 0.224, center = 0.112) |>
  gf_labs(x = "Log (compensation)", y = "# of CEOs")
```



Here we needed to pick magic numbers for the binwidth (e.g., 0.224) and centering of the

histogram so that it matched the results from the book.

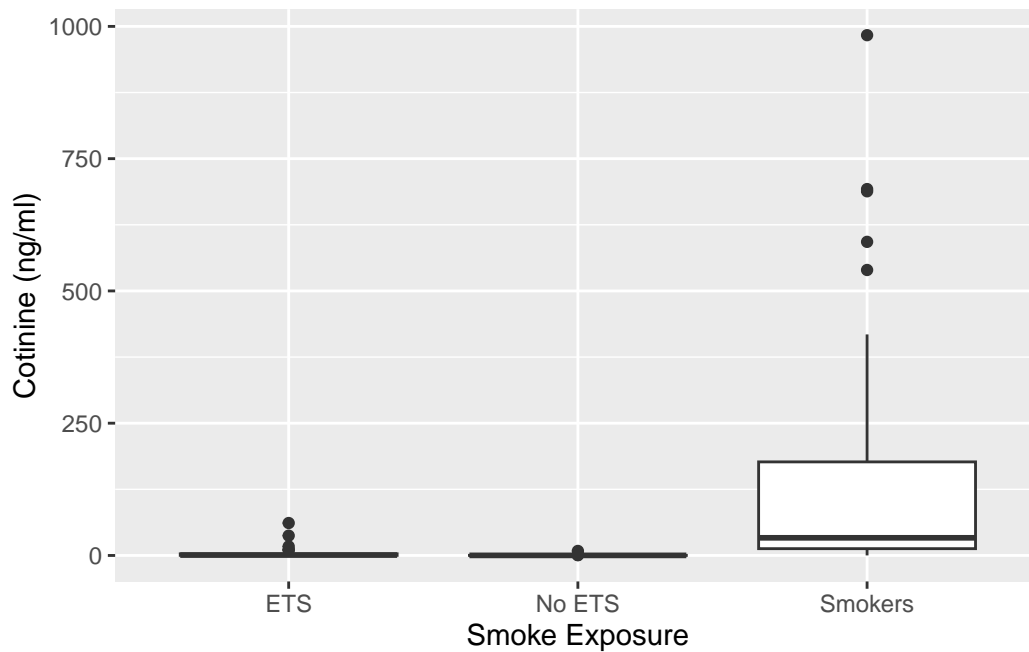
## Re-Expression to Equalize Spread Across Groups

We begin by reading in the data.

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

```
# Figure 4.8, page 107
```

```
gf_boxplot(cotinine ~ smoke_exposure, data = PassiveSmoke) |>  
gf_labs(x = "Smoke Exposure", y = "Cotinine (ng/ml)")
```



```
# Figure 4.9
```

```
gf_boxplot(log(cotinine) ~ smoke_exposure, data = PassiveSmoke) |>  
gf_labs(x = "Smoke Exposure", y = "Log(cotinine)")
```

