SDM4 in R: Understanding and Comparing Distributions (Chapter 4)

Nicholas Horton (nhorton@amherst.edu) and Sarah McDonald

June 4th, 2018

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 4: Understanding and comparing distributions

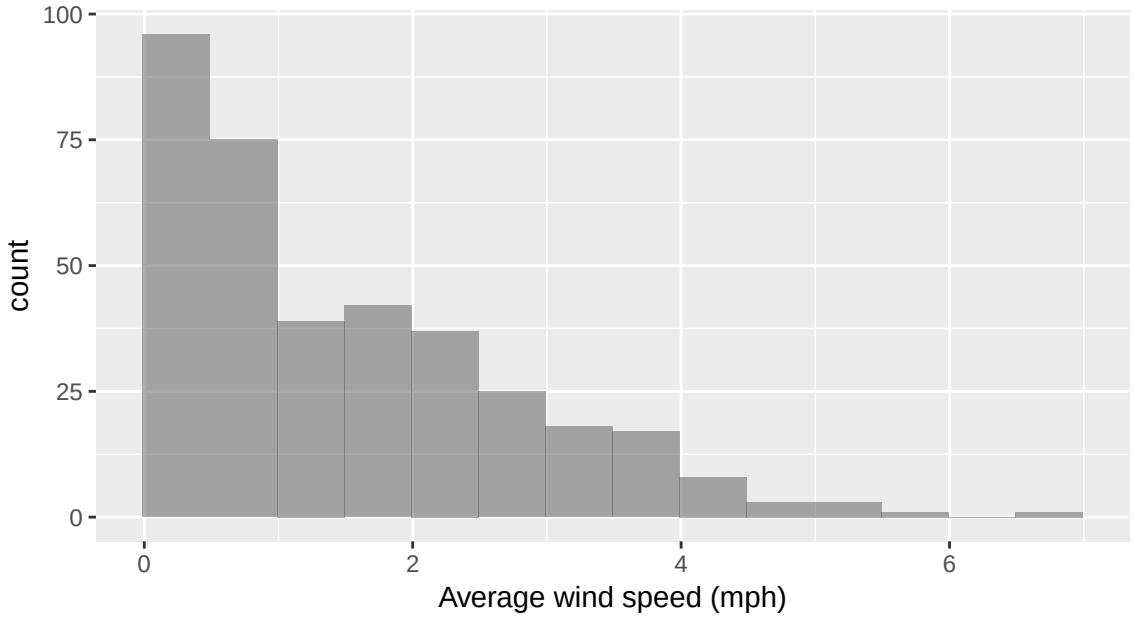
Section 4.1: Comparing groups with histograms

See Figure 4.1 on page 85

```
library(mosaic)
library(readr)
options(digits = 3)
Hopkins <-
read_delim("http://nhorton.people.amherst.edu/sdm4/data/Hopkins_Forest_2011.txt", delim = "\t")
names(Hopkins)
```

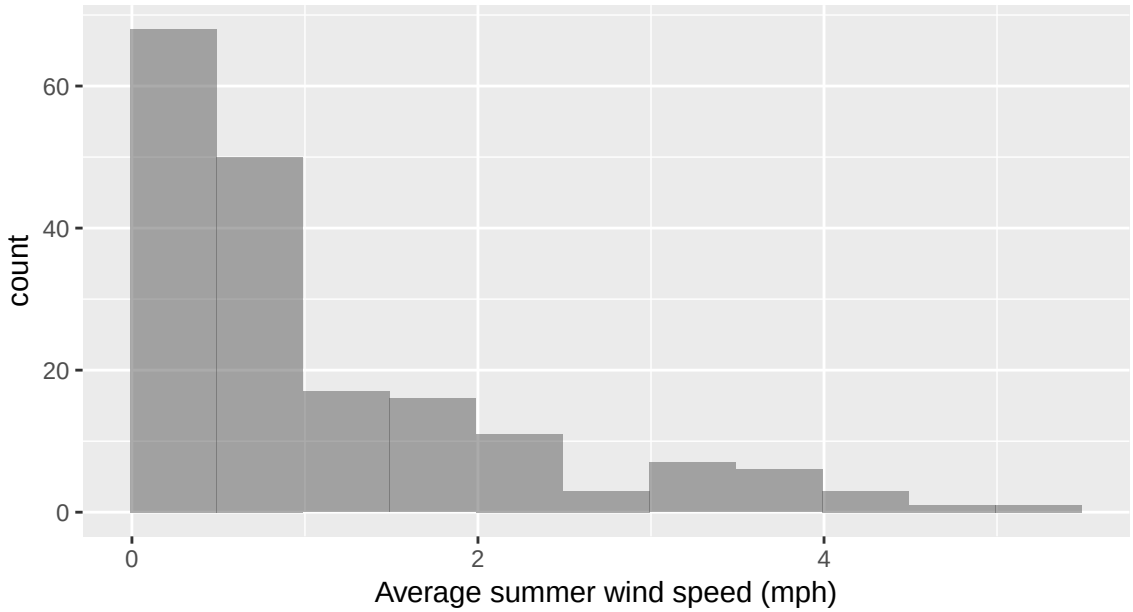
```
##  [1] "Season"       "AvgWindSpeed" "Month"        "Day"
##  [5] "DayofYear"    "AvgTempC"     "AvgTempF"     "MaxWindSpeed"
##  [9] "AvgBarom"     "Precip"
```

```
gf_histogram(~ AvgWindSpeed, binwidth = 0.5, center = 0.24,
          xlab = "Average wind speed (mph)", data = Hopkins)
```



Here we reproduce Figure 4.2 on page 85

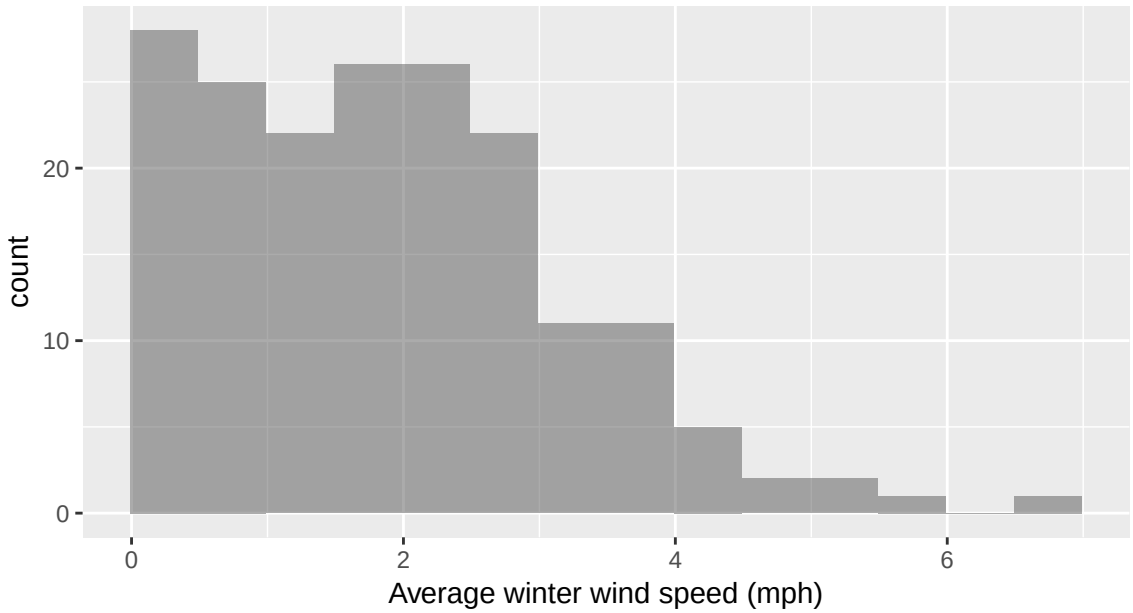
```
Hopkins <- mutate(Hopkins,
  Summer = Month >= 4 & Month <= 9,
  Winter = !Summer
)
gf_histogram(~ AvgWindSpeed, binwidth = 0.5, center = 0.24,
          xlab = "Average summer wind speed (mph)", data = filter(Hopkins, Summer == TRUE))
```




```
favstats(~ AvgWindSpeed, data = filter(Hopkins, Summer == TRUE))
```

```
##  min   Q1 median   Q3  max mean  sd   n missing
##    0 0.35   0.71 1.62 5.47 1.11 1.1 183       0
```

```
gf_histogram(~ AvgWindSpeed, binwidth = 0.5, center = 0.24,
          xlab = "Average winter wind speed (mph)", data = filter(Hopkins, Winter == TRUE))
```



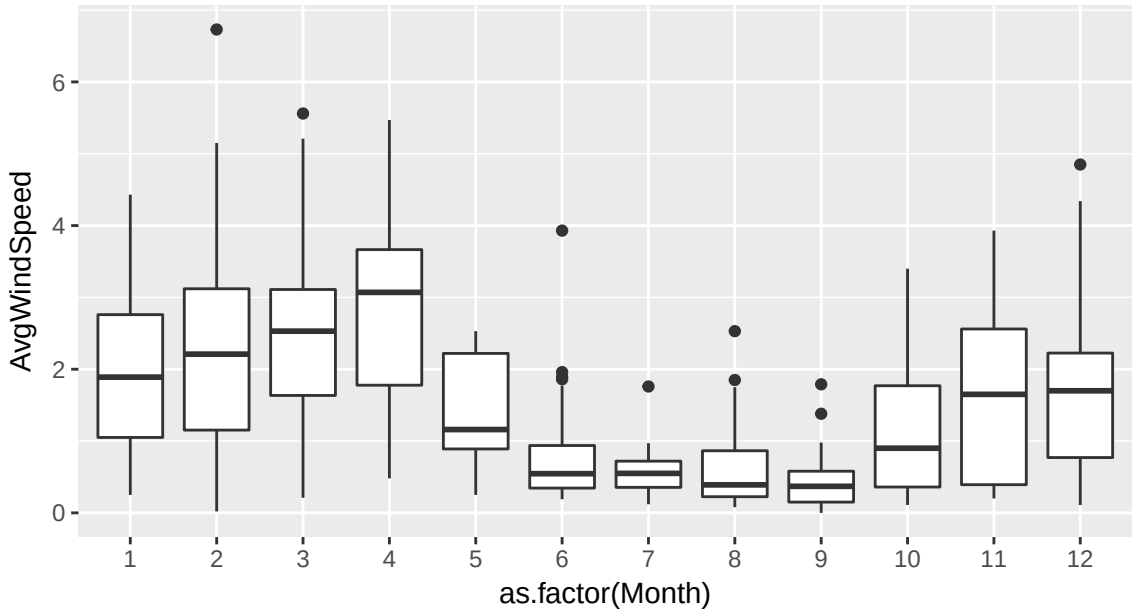

```
favstats(~ AvgWindSpeed, data = filter(Hopkins, Winter == TRUE))
```

```
##   min   Q1 median   Q3  max mean   sd   n missing
##  0.02 0.84   1.72 2.66 6.73  1.9 1.29 182       0
```

Section 4.2: Comparing groups with boxplots

Here we reproduce Figure 4.3 on page 87

```
gf_boxplot(AvgWindSpeed ~ as.factor(Month), data = Hopkins)
```



Section 4.3: Outliers

```
filter(Hopkins, Month == 2, AvgWindSpeed > 6)   # in February
```

```
## # A tibble: 1 x 12
##   Season AvgWindSpeed Month   Day DayofYear AvgTempC AvgTempF MaxWindSpeed
##   <chr>         <dbl> <int> <int>     <int>    <dbl>    <dbl>        <dbl>
## 1 Winter         6.73     2    19        50    -5.09     22.8         39.5
## # ... with 4 more variables: AvgBarom <dbl>, Precip <dbl>, Summer <lgl>,
## #   Winter <lgl>
```

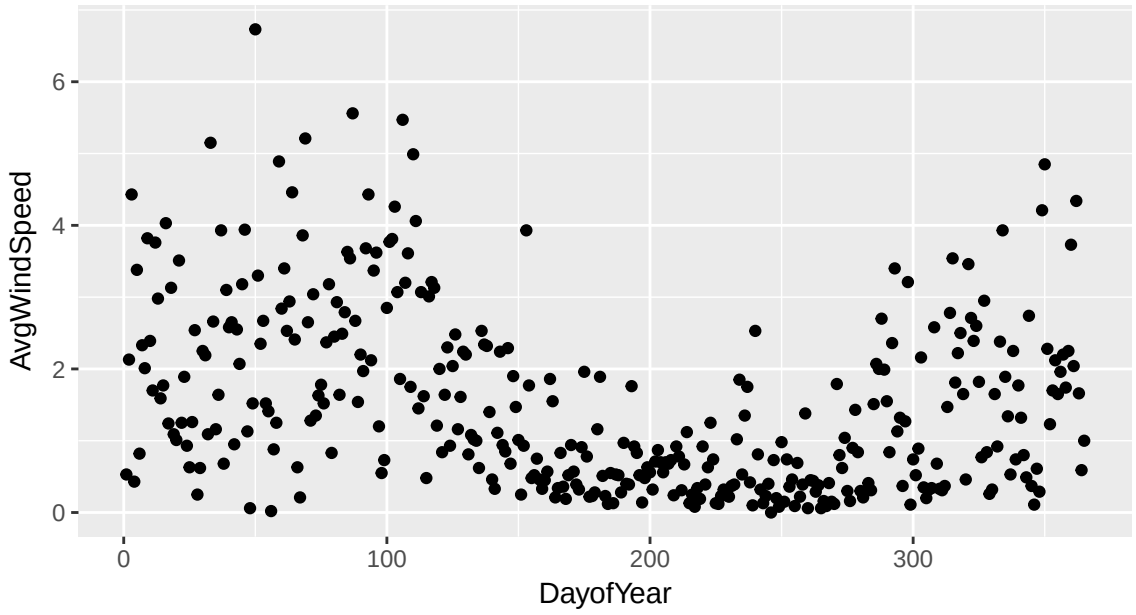
```
filter(Hopkins, Month == 6, AvgWindSpeed > 3.9) # in June
```

```
## # A tibble: 1 x 12
##   Season AvgWindSpeed Month   Day DayofYear AvgTempC AvgTempF MaxWindSpeed
##   <chr>         <dbl> <int> <int>     <int>    <dbl>    <dbl>        <dbl>
## 1 Summer         3.93     6     2       153     14.7     58.5         38.8
## # ... with 4 more variables: AvgBarom <dbl>, Precip <dbl>, Summer <lgl>,
## #   Winter <lgl>
```

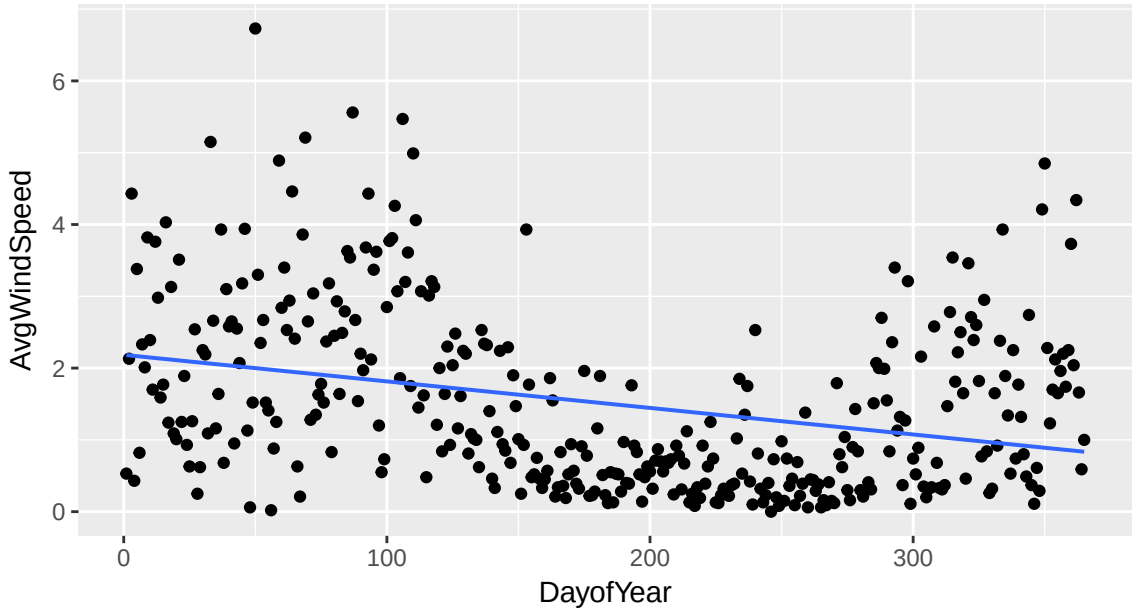
Section 4.4: Timeplots: Order, please!

See Figures 4.4 through 4.6 starting on page 92

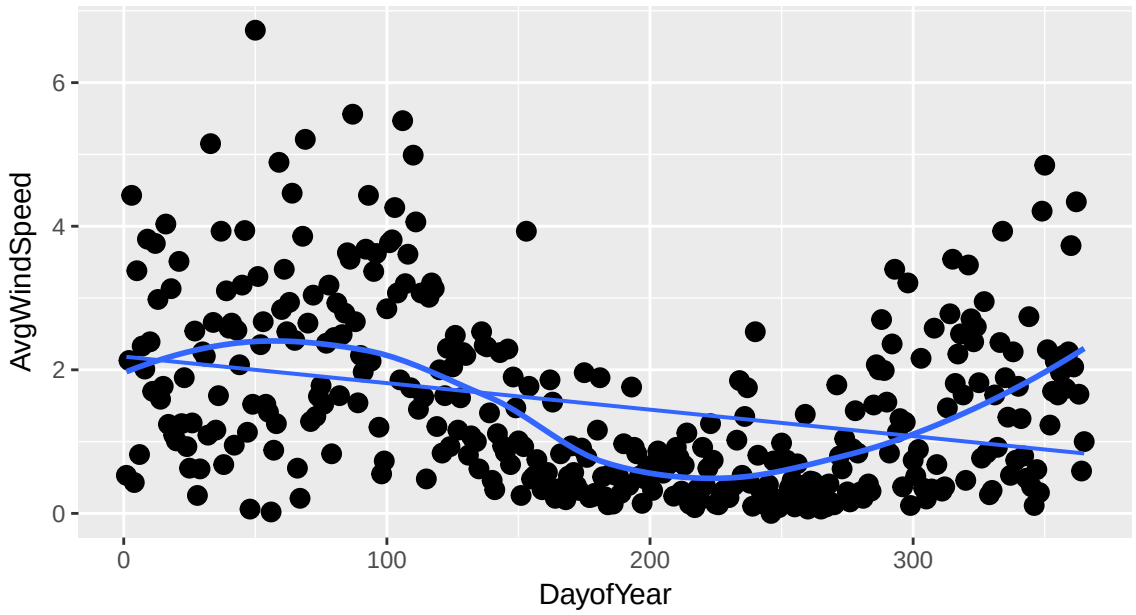
```
gf_point(AvgWindSpeed ~ DayofYear, data = Hopkins)
```




```
gf_point(AvgWindSpeed ~ DayofYear, data = Hopkins) %>%
  gf_lm()
```




```
gf_point(AvgWindSpeed ~ DayofYear, lwd = 3, data = Hopkins) %>%
  gf_lm() %>%
  gf_smooth(se = FALSE)
```



Section 4.5: Re-expressing data: A first look

See Figure 4.7 on page 94

```
CEO <- read_delim("http://nhorton.people.amherst.edu/sdm4/data/CEO_Salary_2012.txt", delim = "\t")
favstats(~ One_Year_Pay, data = CEO)
```

```
##  min   Q1 median   Q3 max mean   sd   n missing
##    0 3.88   6.97 13.4 131 10.5 11.5 500       0
```

```
gf_histogram(~ One_Year_Pay, binwidth = 2.5, center = 1.24, data = CEO)
```

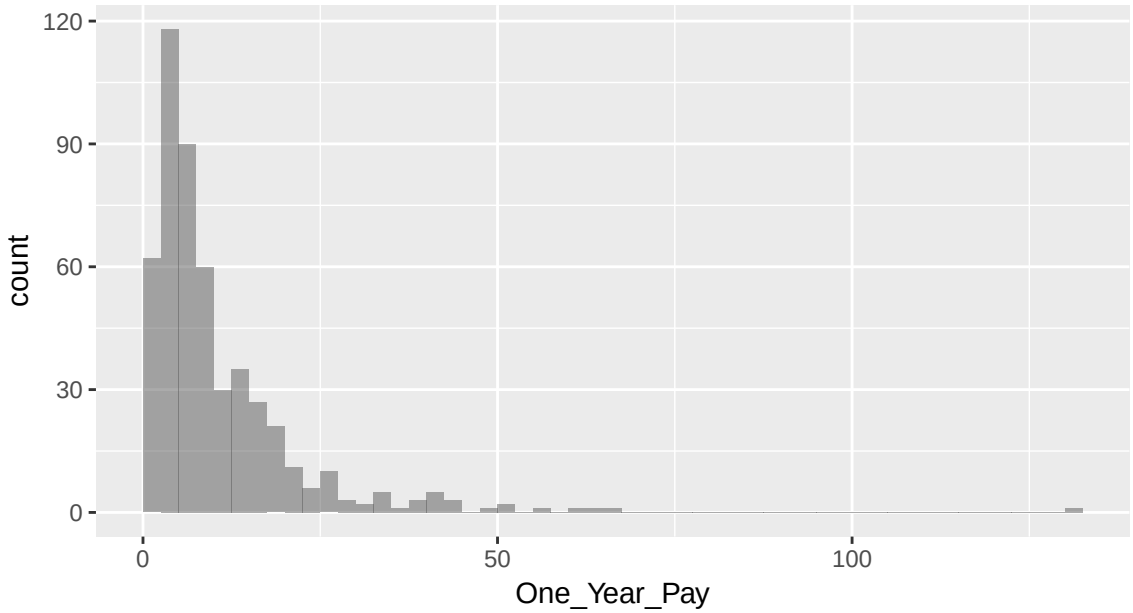


Figure 4.8 on page 95

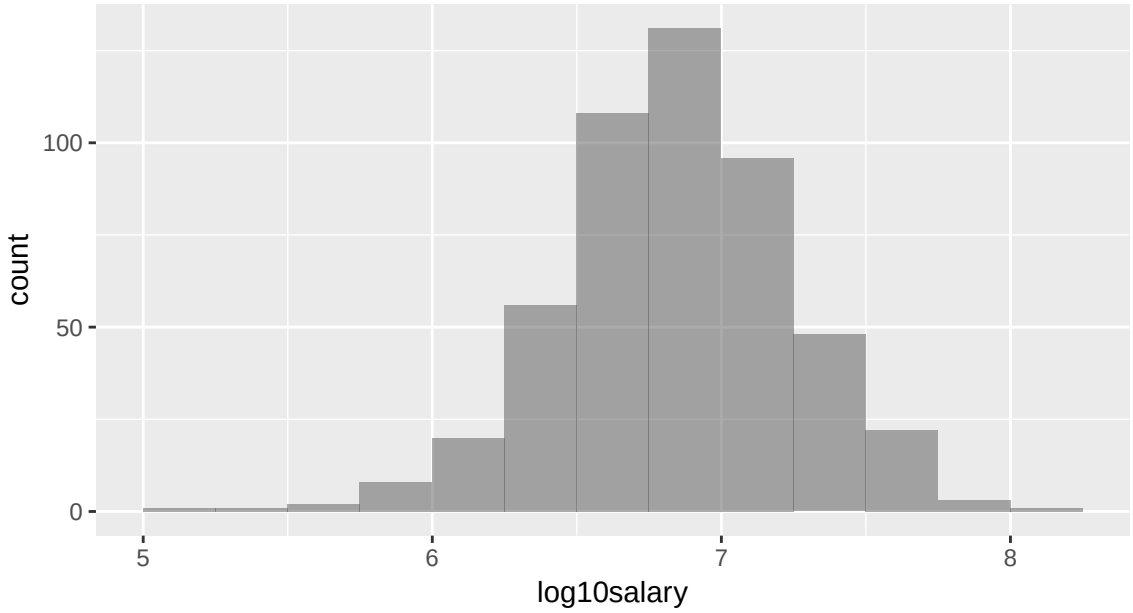
```
nrow(CEO)   # let's get rid of the CEO's with 0 salaries...
```

```
## [1] 500
```

```
CEO <- filter(CEO, One_Year_Pay > 0)
nrow(CEO)
```

```
## [1] 497
```

```
CEO <- mutate(CEO, log10salary = log10(One_Year_Pay * 1000000))
gf_histogram(~ log10salary, binwidth = .25, center = .124, data = CEO)
```



On the log 10 scale, we can roughly interpret the values as the number of digits in the CEO salary.