Lab 3: Intro to Data Visualization
STAT 630, Fall 2021

Topics:

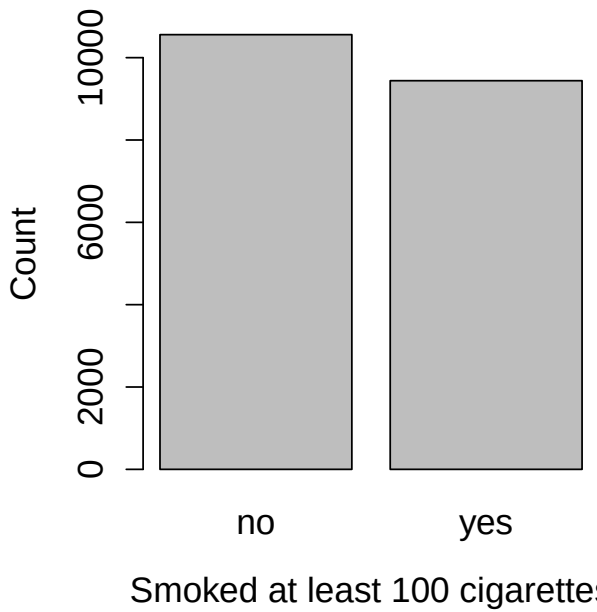
- Tables
- Contingency Tables
- Bar Plots
- Histograms
- Maps

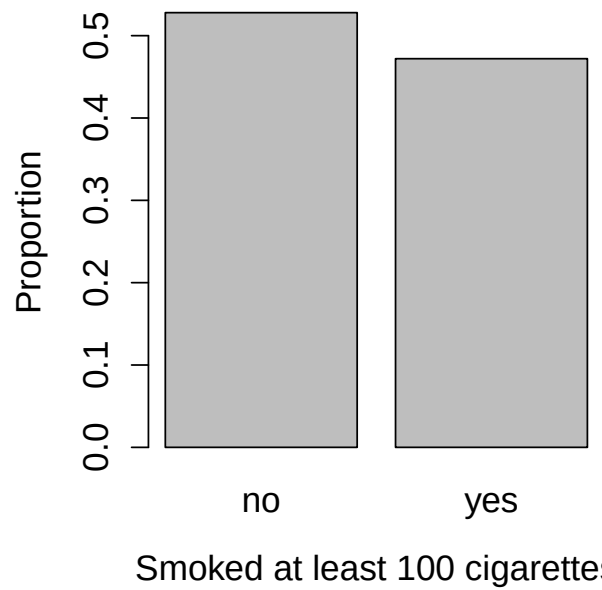
Tables

```
# frequency table
> table(cdc$smoke100)
    0     1
10559  9441

# relative frequency table
> table(cdc$smoke100) / 20000
      0       1
0.52795 0.47205
```

Bar Plots

```
> smoke_tb <- table(cdc$smoke100)
> barplot(smoke_tb, xlab="Smoked at least 100 cigarettes",
          names.arg = c("no", "yes"), ylab="Count")
> barplot(smoke_tb/20000, xlab="Smoked at least 100 cigarettes",
          names.arg = c("no", "yes"), ylab="Proportion")
```



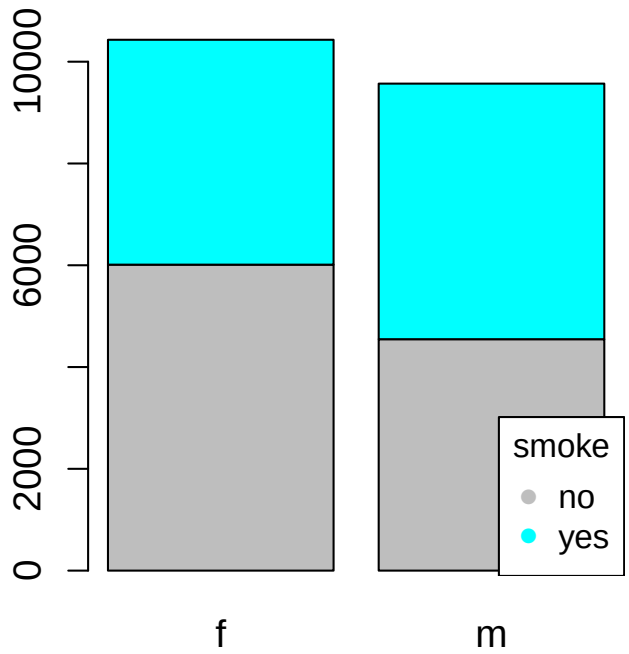


Contingency Tables

```
> table(cdc$smoke100, cdc$gender)

      f    m
0 6012 4547
1 4419 5022

> addmargins(table(cdc$smoke100, cdc$gender))

          f     m   Sum
0      6012  4547 10559
1      4419  5022  9441
Sum   10431  9569 20000
```

Stacked Bar Plot

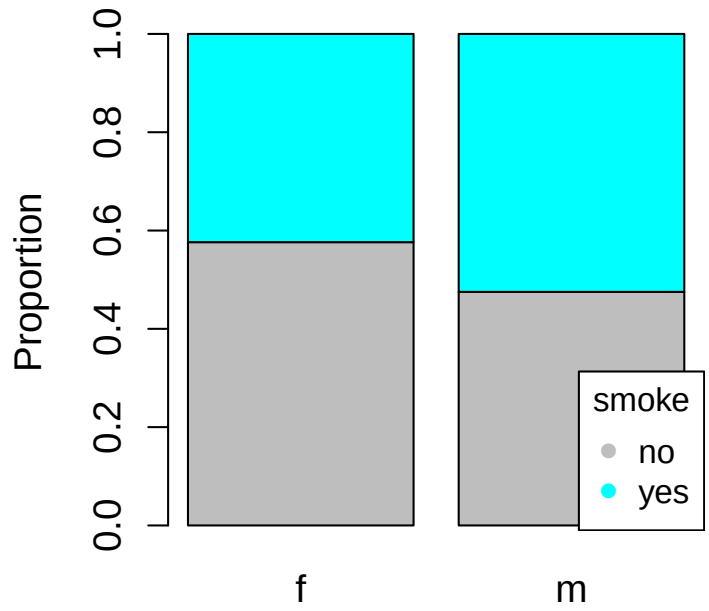
```
> barplot(table(cdc$smoke100, cdc$gender),
          col=c("grey", "cyan"))
> legend("bottomright", c("no", "yes"), col=c("grey", "cyan"),
         title="smoke", pch=16)
```



Row and Column Proportions

```
> prop.table(table(cdc$smoke100, cdc$gender))
          f       m
  0 0.30060 0.22735
  1 0.22095 0.25110

# divides counts by row totals
> prop.table(table(cdc$smoke100, cdc$gender), margin=1)
            f         m
  0 0.5693721 0.4306279
  1 0.4680648 0.5319352

# divides counts by column totals
> prop.table(table(cdc$smoke100, cdc$gender), margin=2)
            f         m
  0 0.5763589 0.4751803
  1 0.4236411 0.5248197
```

```
> proptb <- prop.table(table(cdc$smoke100, cdc$gender), margin=2)
> barplot(proptb, col=c("grey", "cyan"), ylab="Proportion")
> legend("bottomright", c("no", "yes"), col=c("grey", "cyan"),
        title = "smoke", pch=16)
```



Factors

- Categorical data in R is often represented as a data type called a **factor**.
- Specifically, factors are stored as integers that have labels associated with each unique integer value. The labels are the names of the different categories.
- Use the `factor()` function to create a factor, and the `levels` argument to specify the ordering of the categories.

```
> class(cdc$genhlth)
[1] "character"

> table(cdc$genhlth)
excellent      fair      good      poor very good
     4657      2019      5675       677      6972

# create factor and specify order for levels
> cdc$genhlth <- factor(cdc$genhlth,
    levels=c("poor", "fair", "good", "very good", "excellent"))

> class(cdc$genhlth)
[1] "factor"

> table(cdc$genhlth)
     poor      fair      good very good excellent
      677      2019      5675      6972      4657
```

```
> barplot(table(cdc$genhlth), xlab="General health", ylab="Count")
```

ggplot2

ggplot2 is a popular R package for data visualization. It was created by Hadley Wickham.

Reference: `https://ggplot2.tidyverse.org`

In this class, so far we have focus on the base R approach to creating graphics (the original plotting system in R). I think it's important to know both approaches, since each has its advantages – base R graphics tend to be more customizable, while ggplot2 graphics tend to look nicer without many adjustments. The ggplot2 approach also has some advantages when dealing with categorical data.

To install `ggplot2` run the following command in the console:

```
> install.packages("ggplot2")
```

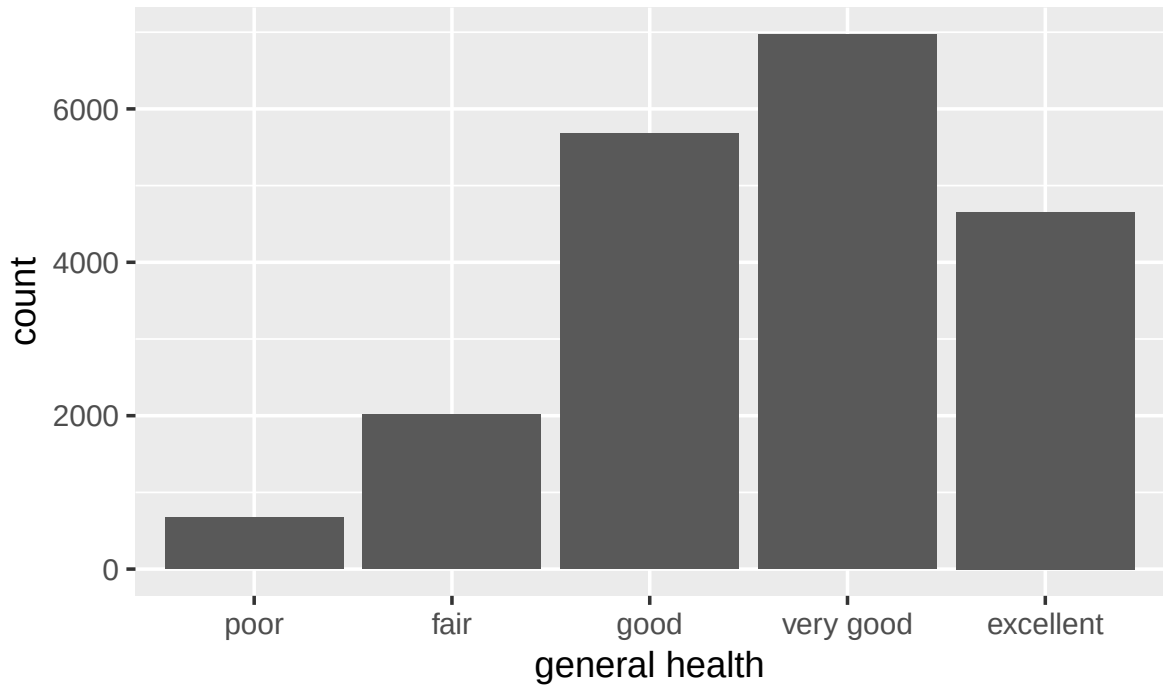
You only need to install the package once. (If you are using the R Studio Cloud, you don't need to do this since the package should already be installed.)

To load `ggplot2` into your current R session run the following command:

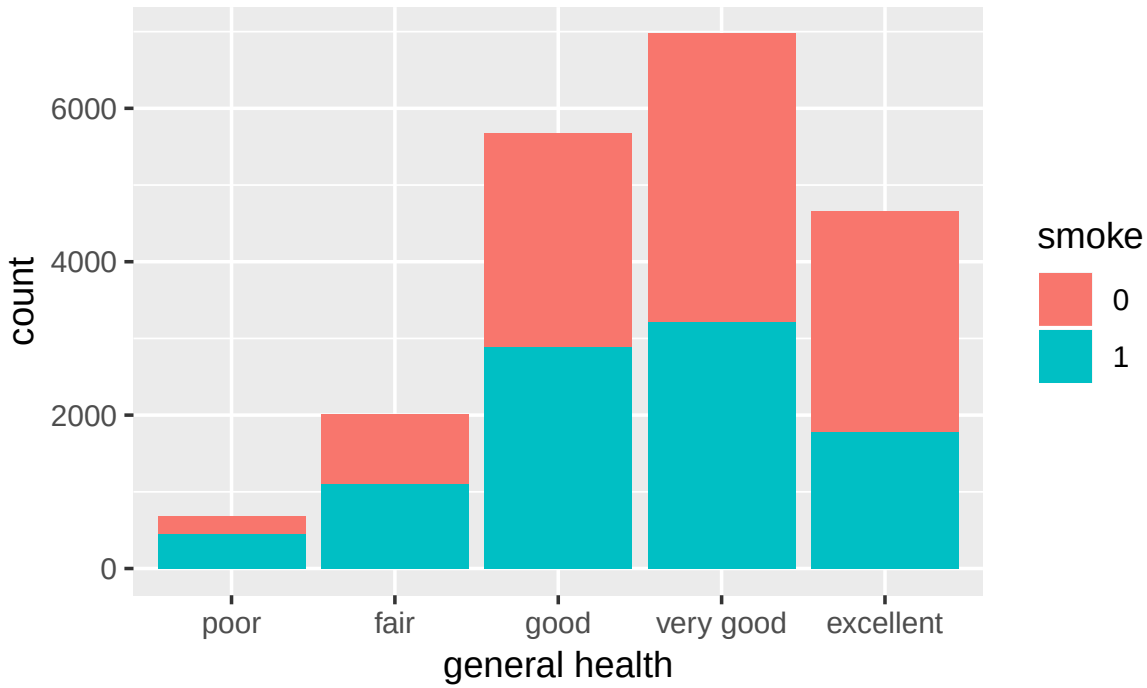
```
> library(ggplot2)
```

This command needs to be run during each R session when you use the package.

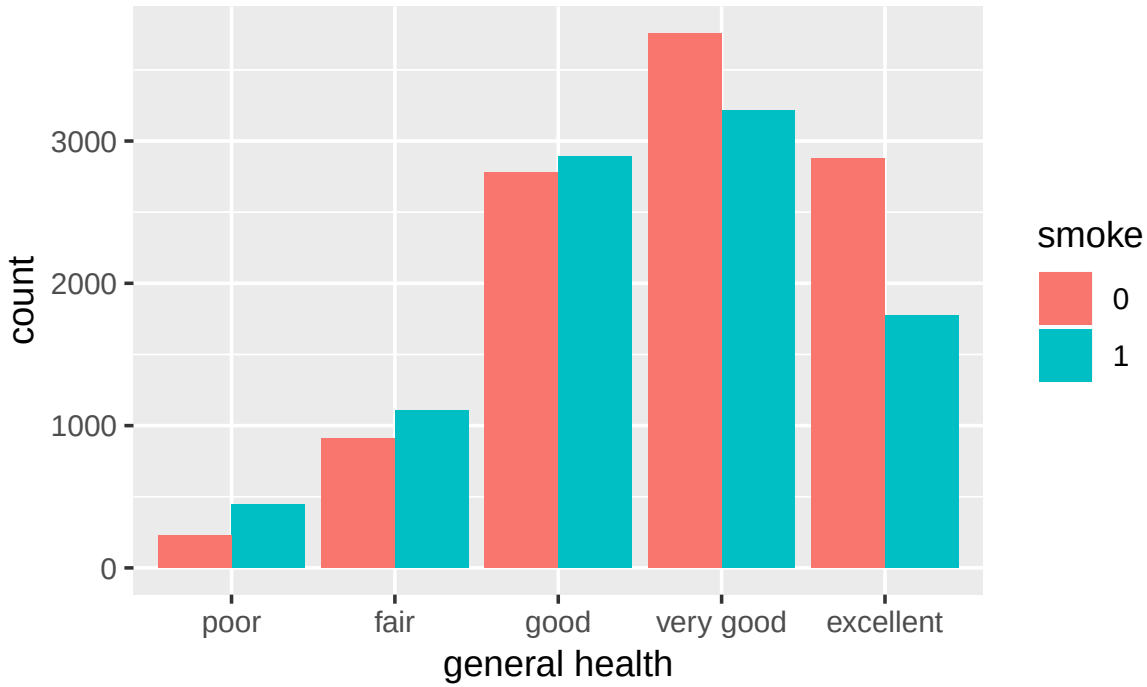
```
library(ggplot2)
ggplot(data = cdc) +
  geom_bar(aes(x=genhlth)) +
  labs(x="general health")
```



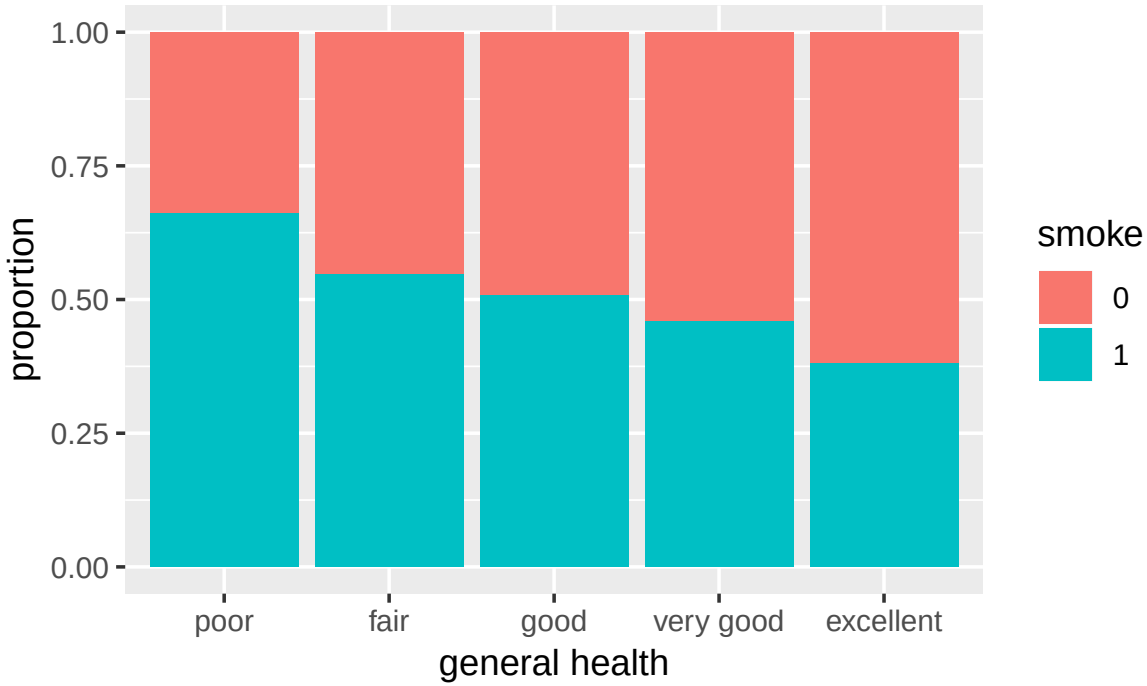
```
ggplot(data = cdc) +
  geom_bar(aes(x=genhlth, fill=factor(smoke100))) +
  labs(x="general health", fill="smoke")
```



```
ggplot(data = cdc) +
  geom_bar(aes(x=genhlth, fill=factor(smoke100)), position="dodge") +
  labs(x="general health", fill="smoke")
```



```
ggplot(data = cdc) +
  geom_bar(aes(x=genhlth, fill=factor(smoke100)), position="fill") +
  labs(x="general health", y="proportion", fill="smoke")
```



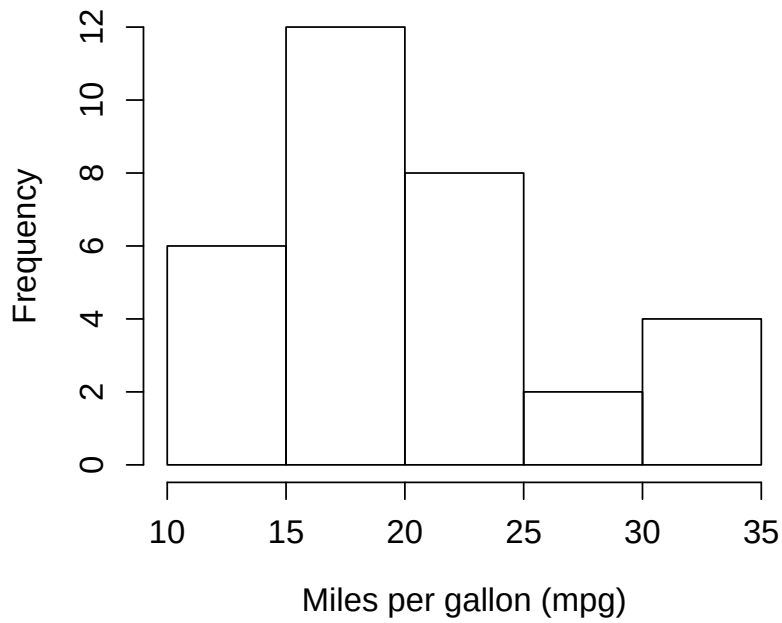
Histograms

- Histograms are a useful way to visualize the distribution of a numerical (continuous) variable.
- To construct a histogram, the range of the data is divided into bins of equal width. Then the number of observations falling in each bin are counted. The counts are plotted as rectangles over each bin.

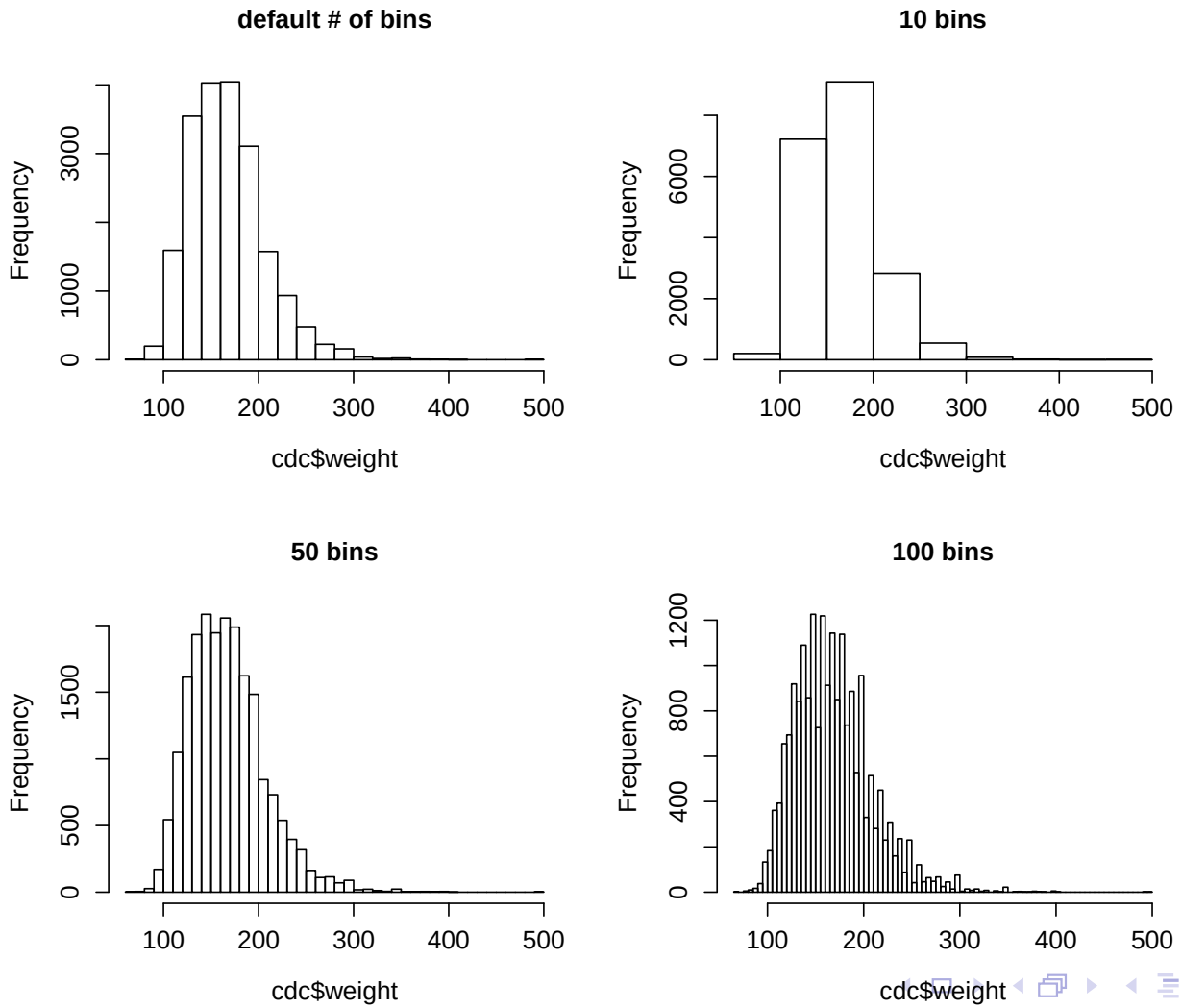
```
> sort(mtcars$mpg)
 [1] 10.4 10.4 13.3 14.3 14.7 15.0 15.2 15.2 15.5 15.8 16.4
[12] 17.3 17.8 18.1 18.7 19.2 19.2 19.7 21.0 21.0 21.4 21.4
[23] 21.5 22.8 22.8 24.4 26.0 27.3 30.4 30.4 32.4 33.9
```

Bin	$(10, 15]$	$(15, 20]$	$(20, 25]$	$(25, 30]$	$(30, 35]$
Count	6	12	8	2	4

```
> hist(mtcars$mpg, main='', xlab="Miles per gallon (mpg)")
```



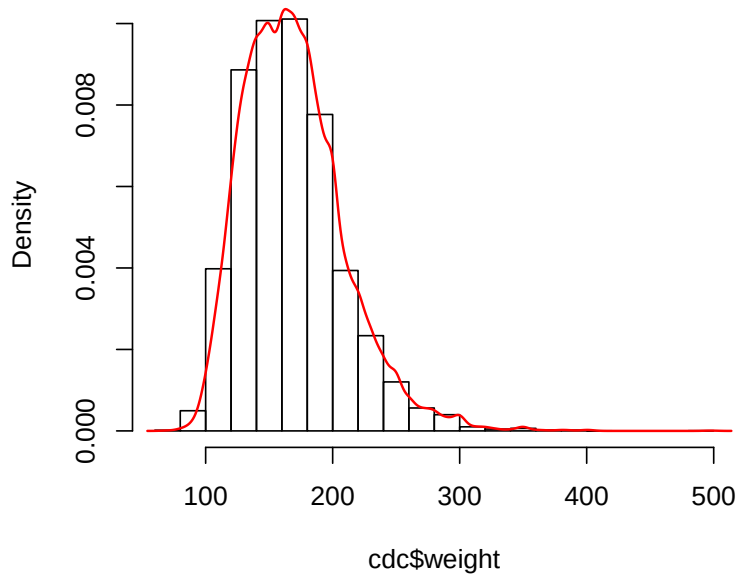
```
> par(mfrow=c(2,2)) # split plot into 4 panels
> hist(cdc$weight, main="default # of bins")
> hist(cdc$weight, breaks=10, main="10 bins")
> hist(cdc$weight, breaks=50, main="50 bins")
> hist(cdc$weight, breaks=100, main="100 bins")
> dev.off() # resets graphical parameters
```



Histogram Density Plot

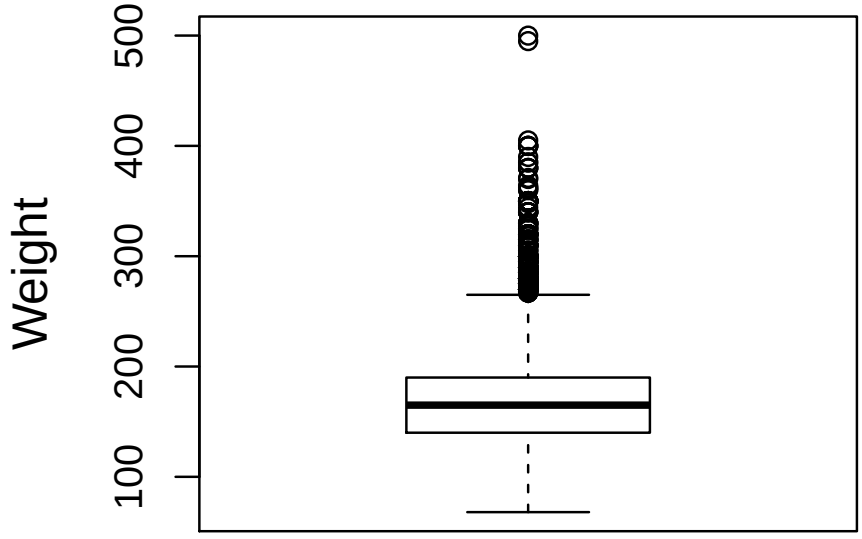
- To make a histogram density plot set `freq=FALSE`
- When density values are plotted the area under the histogram is 1 (i.e., integrates to 1). Note that the area under a histogram is computed by summing up the areas of each rectangle (bin widths $\times$ heights)
- Use `density()` to superimpose a smooth density curve.

```
> hist(cdc$weight, freq=FALSE, main='')
> lines(density(cdc$weight), col="red")
```



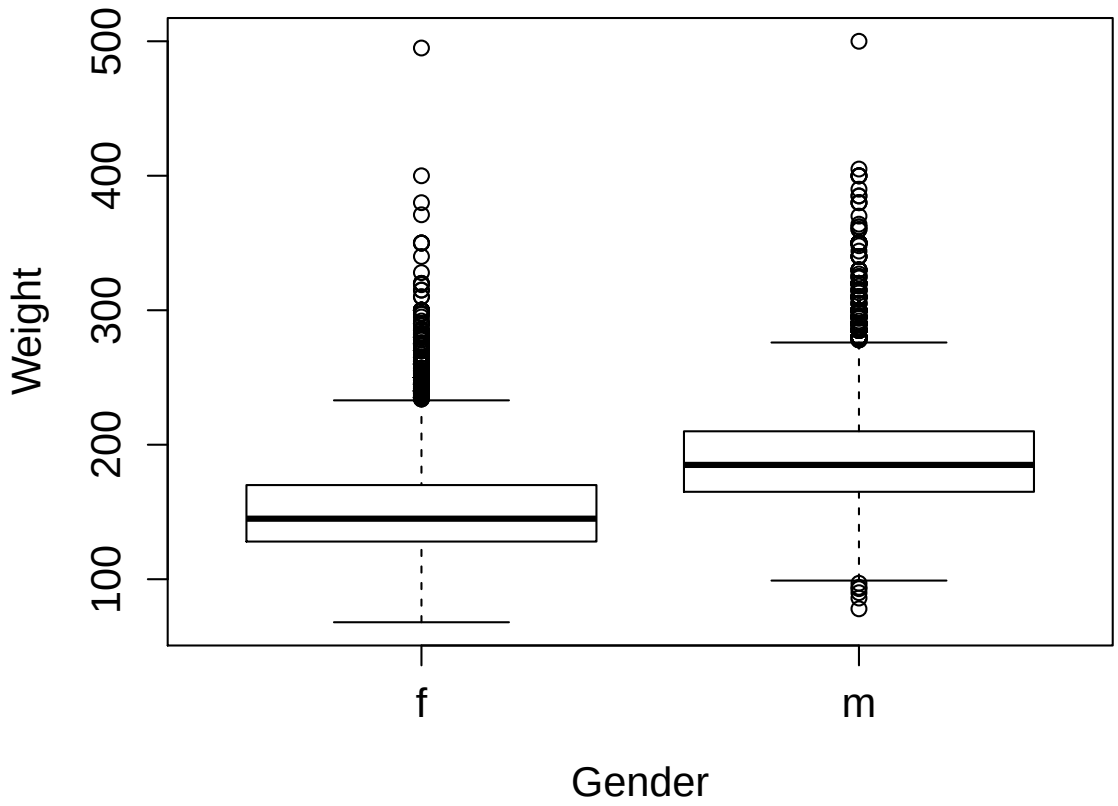
Box Plot

```
> boxplot(cdc$weight, ylab = "Weight")
```



Side-by-Side Box Plot

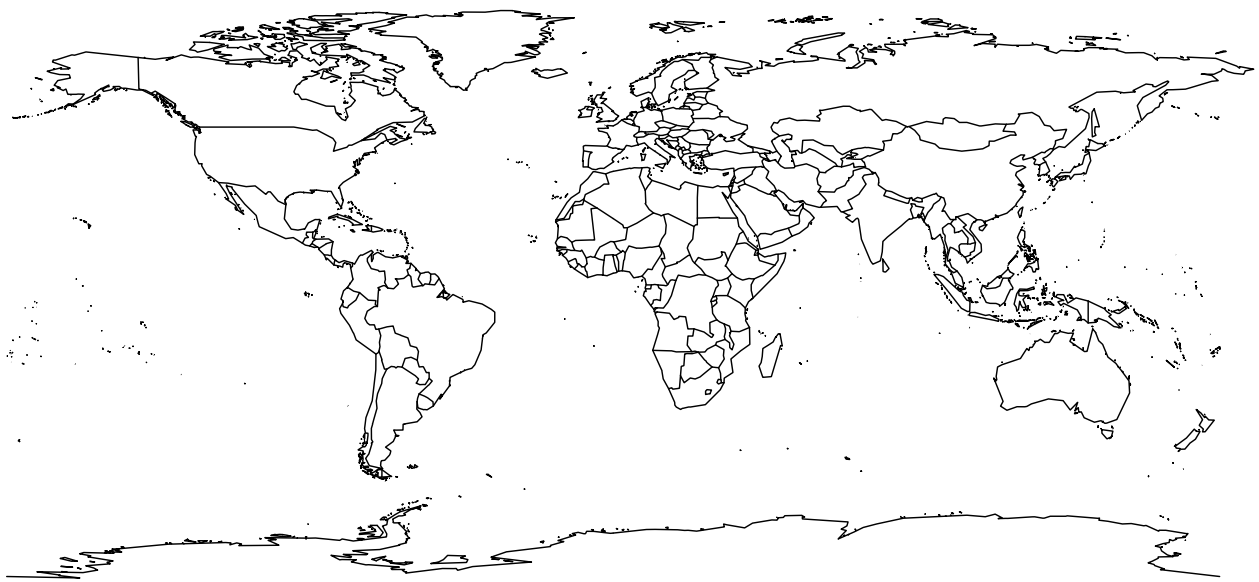
```
> boxplot(weight ~ gender, data=cdc, xlab="Gender", ylab="Weight")
```



Maps

Use `library()` to load the `maps` package into R.

```
> library(maps)
> map("world")
```



Maps

```
> map("state", "california")
```

Maps

```
> map("county", "ca")
```

EPA Stream Data Set

- The Environmental Protection Agency (EPA) sampled nearly 2000 stream sites across the conterminous US during the summer months of 2008/09.
- This was part of a larger environmental monitoring program called the National Rivers and Stream Assessment (NRSA).
- The condition of the stream sites were evaluated as `Good`, `Fair`, or `Poor` according to an aquatic health index.

EPA United States Environmental Protection Agency

Environmental Topics | Laws & Regulations | About EPA | Search EPA.gov

CONTACT US SHARE

National Aquatic Resource Surveys

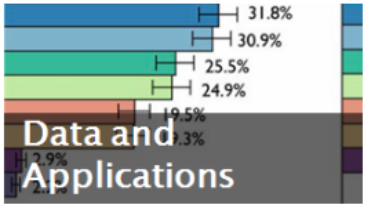
Reports, Sampling and Timeline

- NLA 2012 Report
- NWCA 2011 Report
- Photos of NARS Sampling
- Timeline of Field Seasons

The National Aquatic Resource Surveys (NARS) are collaborative programs between EPA, states, and tribes designed to assess the quality of the nation's coastal waters, lakes and reservoirs, rivers and streams, and wetlands using a statistical survey design. The NARS provide critical, groundbreaking, and nationally-consistent data on the nation's waters.

- Background
- Indicators

- National Coastal Condition Assessment (NCCA)

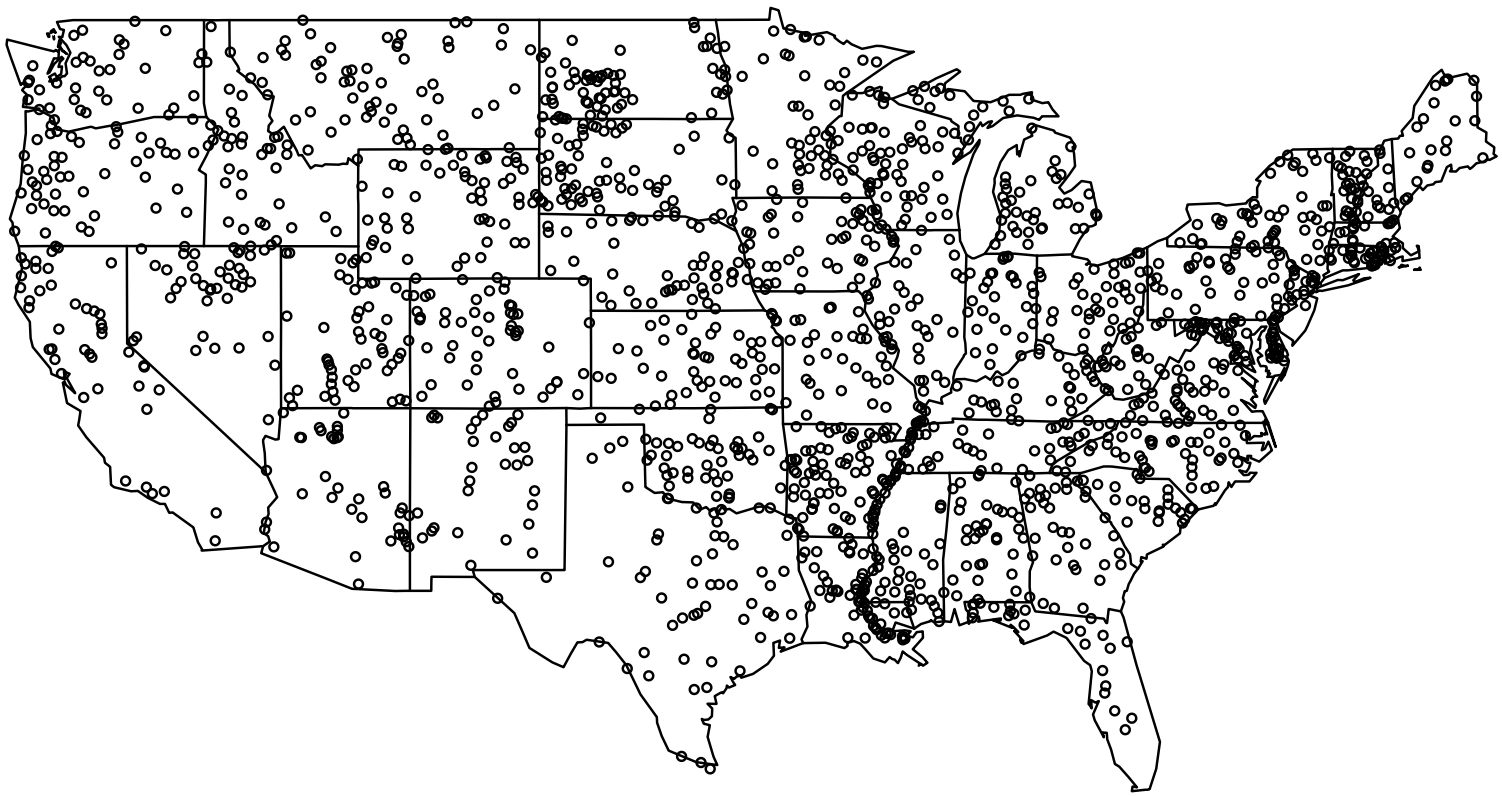


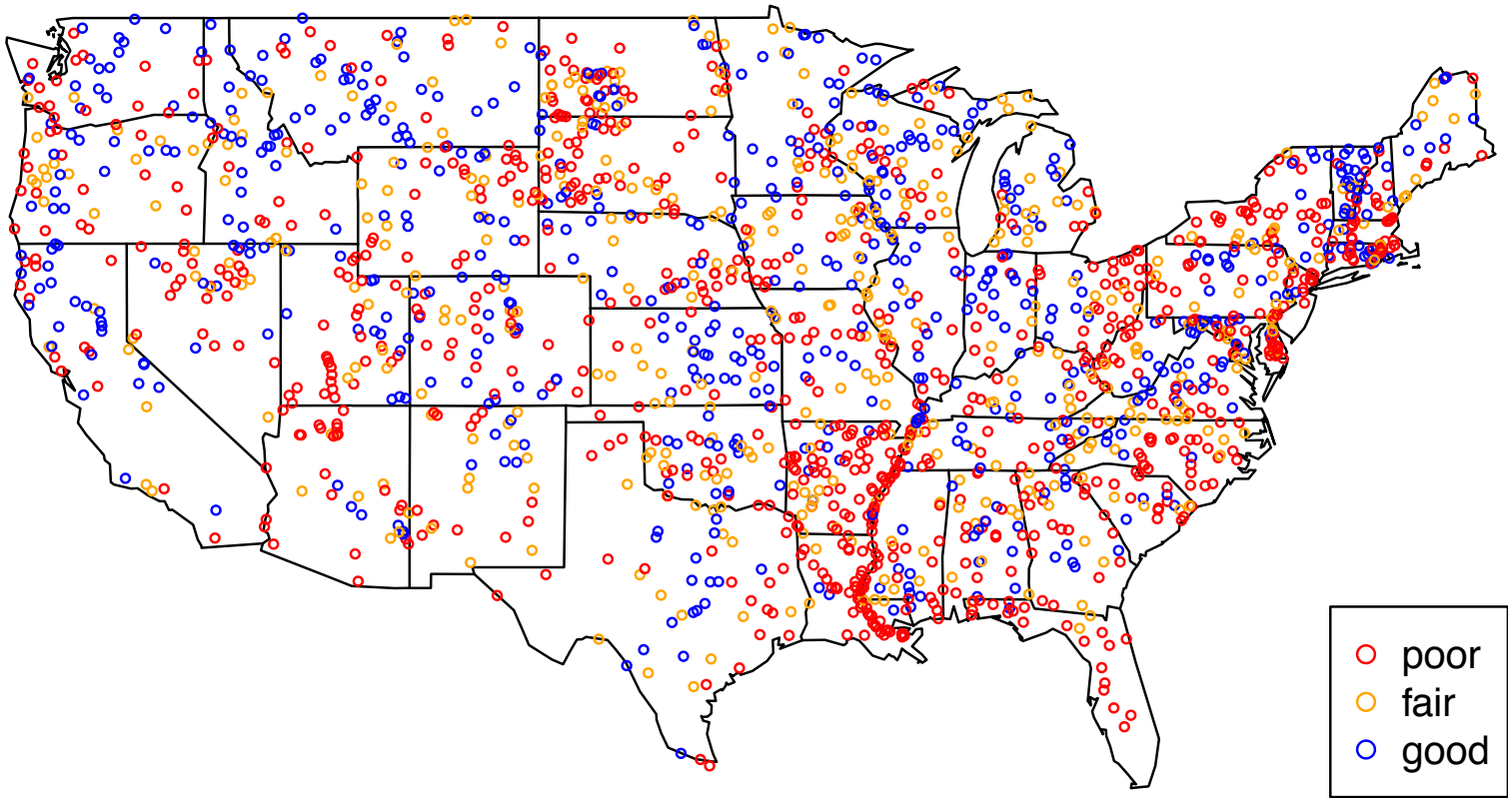
- NARS Data
- Journal Articles

EPA Stream Data Set

```
> nrsa <- readRDS(url("https://ericwfox.github.io/data/nrsa.rds"))

> head(nrsa, n=10)
         lon      lat cond
1  -86.88816 33.22342 Poor
2  -86.77562 33.42492 Poor
3  -87.08381 31.67664 Poor
4  -86.32363 33.87273 Good
5  -86.36186 32.99387 Poor
6  -87.73796 34.09180 Poor
7  -85.75963 33.77874 Fair
8  -87.14547 33.35812 Fair
9  -85.61117 34.71586 Poor
10 -87.04203 34.95092 Poor

> dim(nrsa)
[1] 1859    3
```

```
> map("state")
> points(nrsa$lon, nrsa$lat, cex=0.5)
```




poor
fair
good

Code used to create last map:

```
> nrsa_good <- subset(nrsa, cond == "Good")
> nrsa_fair <- subset(nrsa, cond == "Fair")
> nrsa_poor <- subset(nrsa, cond == "Poor")

> map("state")
> points(nrsa_good$lon, nrsa_good$lat, cex=0.5, col = "blue")
> points(nrsa_fair$lon, nrsa_fair$lat, cex=0.5, col = "orange")
> points(nrsa_poor$lon, nrsa_poor$lat, cex=0.5, col = "red")
> legend("bottomright", c("poor", "fair", "good"),
        col=c("red","orange","blue"), pch=1)
```