Graphics

April 26, 2016

Contents

1 Other Graphs

1.1 Stem-and-Leaf Plots

Stem produces a stem-and-leaf plot of the values in x. The parameter scale can be used to expand the scale of the plot.

```
x<-c(25, 45, 50, 54, 55, 61, 64, 68, 72, 75, 75,
      78, 79, 81, 83, 84, 84, 84, 85, 86, 86, 86,
      87, 89, 89, 89, 90, 91, 91, 92, 100)
stem(x)

##
##   The decimal point is 1 digit(s) to the right of the |
##
##    2 | 5
##    3 |
##    4 | 5
##    5 | 045
##    6 | 148
##    7 | 25589
##    8 | 1344456667999
##    9 | 0112
##   10 | 0

stem(x,scale=2)

##
##   The decimal point is 1 digit(s) to the right of the |
##
##    2 | 5
##    3 |
##    3 |
##    4 |
##    4 | 5
##    5 | 04
##    5 | 5
##    6 | 14
##    6 | 8
##    7 | 2
##    7 | 5589
##    8 | 13444
##    8 | 56667999
##    9 | 0112
##    9 |
##   10 | 0

stem(x,scale=.5)

##
##   The decimal point is 1 digit(s) to the right of the |
##
##    2 | 5
##    4 | 5045
##    6 | 14825589
##    8 | 13444566679990112
##   10 | 0
```

1.2 Bar plots

Bar plots display the distribution (frequencies) of a categorical variable through vertical or horizontal bars.

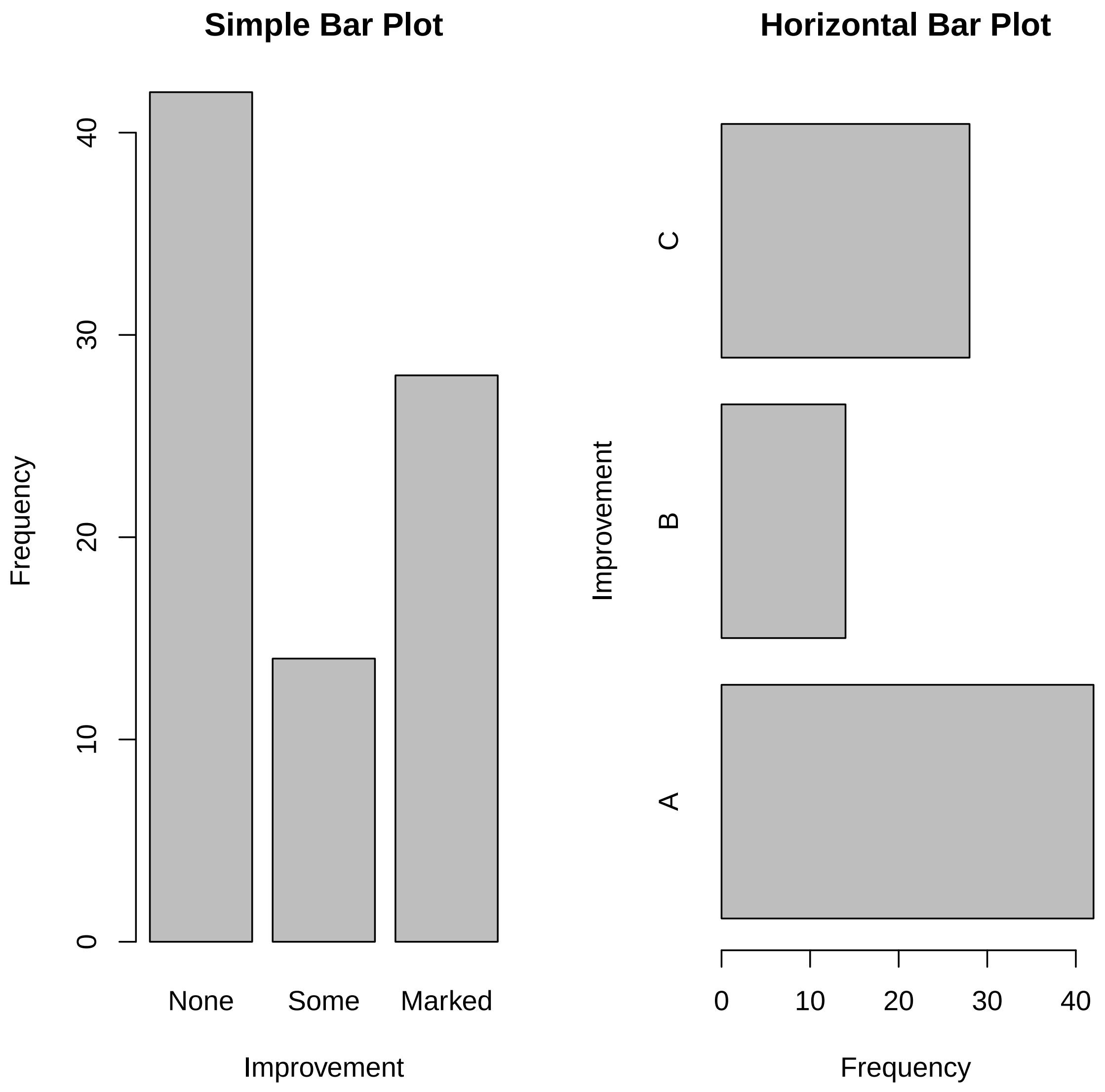
`barplot(height)`

In the following examples, we'll plot the outcome of a study investigating a new treatment for rheumatoid arthritis. The data are contained in the Arthritis data frame distributed with the vcd package . Because the vcd package isn't included in the default R installation, be sure to download and install it before first use (install. packages("vcd")) .

```r
library(vcd)
counts <- table(Arthritis$Improved)
counts

##
##   None   Some Marked
##     42     14     28

#opar<-par(no.readonly = TRUE)
par(mfrow = c(1, 2))
barplot(counts, main = "Simple Bar Plot", xlab = "Improvement", ylab = "Frequency")
barplot(counts, main = "Horizontal Bar Plot", xlab = "Frequency",
        ylab = "Improvement", horiz = TRUE,
        names.arg=c("A","B","C"))
```



```r
#par(opar)
```

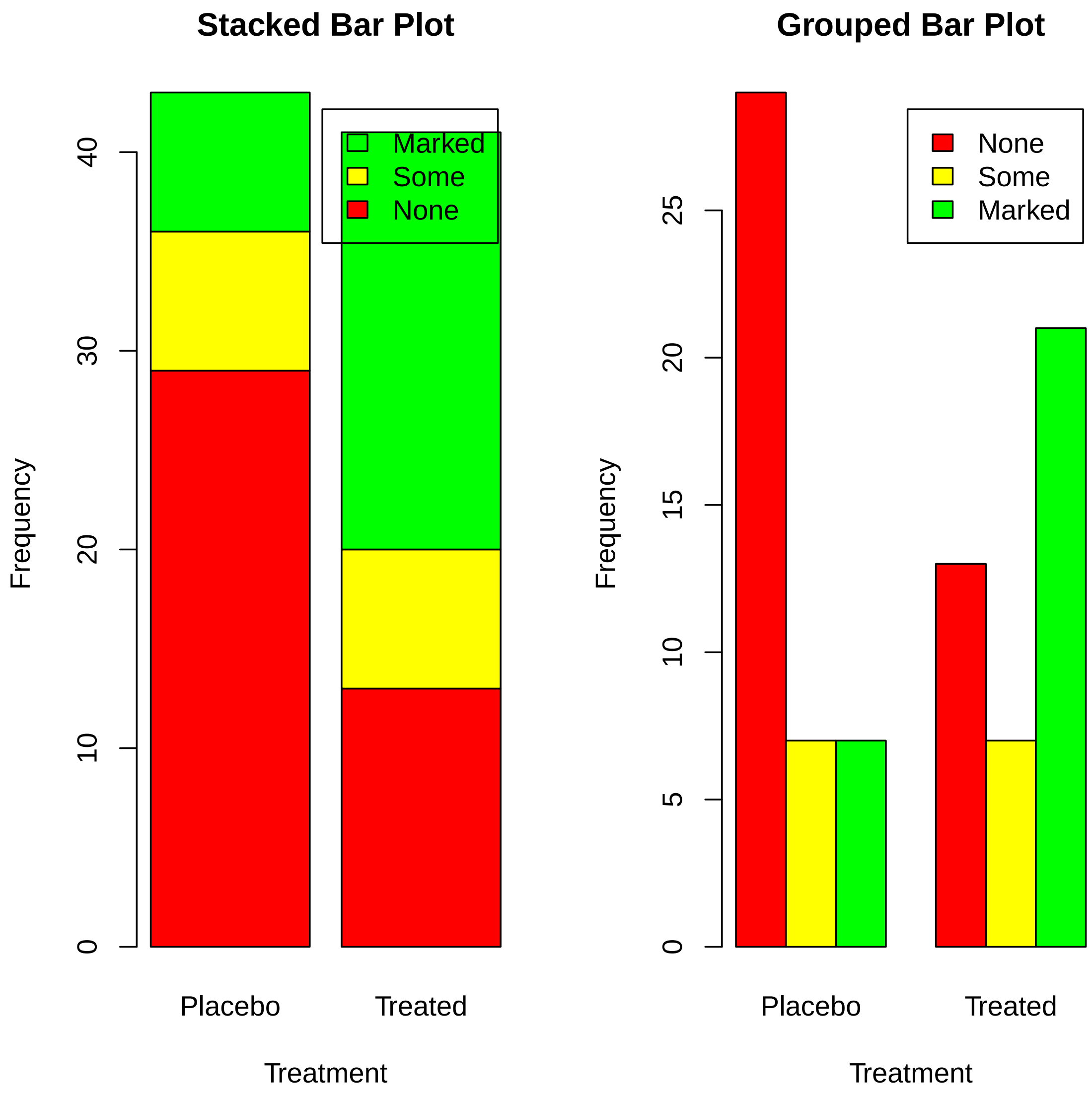
Stacked and grouped bar plots

If height is a matrix rather than a vector, the resulting graph will be a stacked or grouped bar plot. If beside=FALSE (the default), then each column of the matrix produces a bar in the plot, with the values in the column giving the heights of stacked "sub-bars." If beside=TRUE, each column of the matrix represents a group, and the values in each column are juxtaposed rather than stacked.

```r
library(vcd)
counts <- table(Arthritis$Improved, Arthritis$Treatment)
counts
```

```
##
##          Placebo Treated
##   None        29      13
##   Some         7       7
##   Marked       7      21
```

```
#opar<-par(no.readonly = TRUE)
par(mfrow = c(1, 2))
barplot(counts, main="Stacked Bar Plot", xlab="Treatment",
   ylab="Frequency", col=c("red", "yellow","green"), legend=rownames(counts))
barplot(counts, main="Grouped Bar Plot", xlab="Treatment",
   ylab="Frequency", col=c("red", "yellow", "green"),
   legend=rownames(counts), beside=TRUE)
```



```r
#par(opar)
```

Mean bar plots

Bar plots needn't be based on counts or frequencies. You can create bar plots that represent means, medians, standard deviations, and so forth by using the aggregate function and passing the results to the barplot() function .

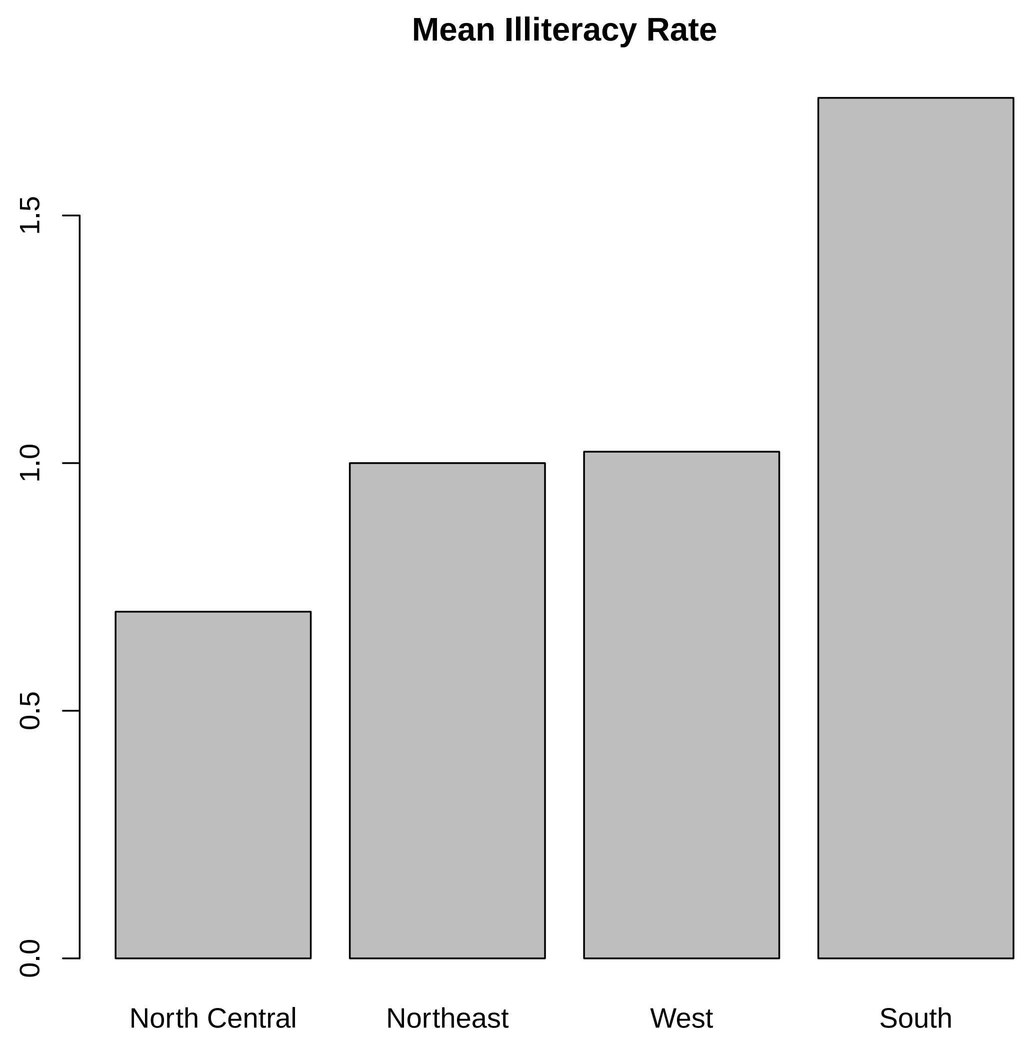
```r
states <- data.frame(state.region, state.x77)
means <- aggregate(states$Illiteracy, by = list(state.region),
    FUN = mean)
means
```

```
##         Group.1        x
## 1     Northeast 1.000000
## 2         South 1.737500
## 3 North Central 0.700000
## 4          West 1.023077
```

```r
means <- means[order(means$x), ]
means
```

```
##         Group.1        x
## 3 North Central 0.700000
## 1     Northeast 1.000000
## 4          West 1.023077
## 2         South 1.737500
```

```r
barplot(means$x, names.arg = means$Group.1)
title("Mean Illiteracy Rate")
```

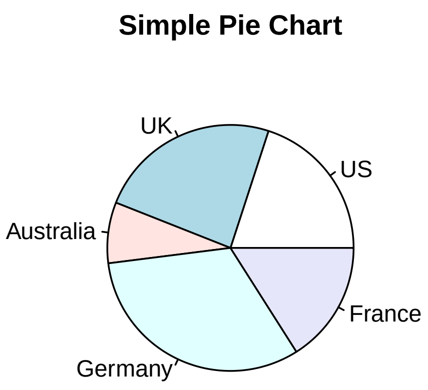


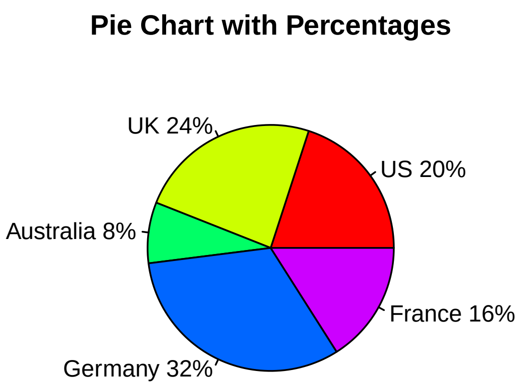
1.3 Pie charts

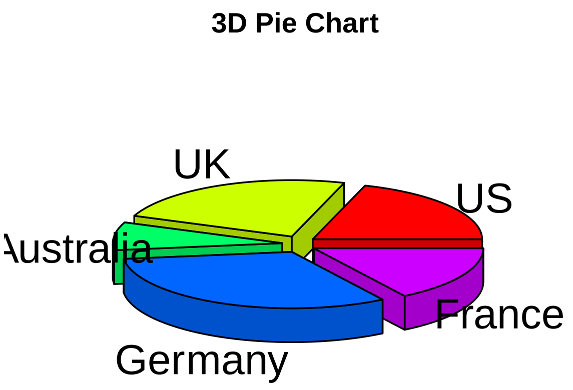
Whereas pie charts are ubiquitous in the business world, they're denigrated by most statisticians, including the authors of the R documentation. They recommend bar or dot plots over pie charts because people are able to judge length more accurately than volume. Perhaps for this reason, the pie chart options in R are quite limited when compared with other statistical software. install.packages("plotrix")

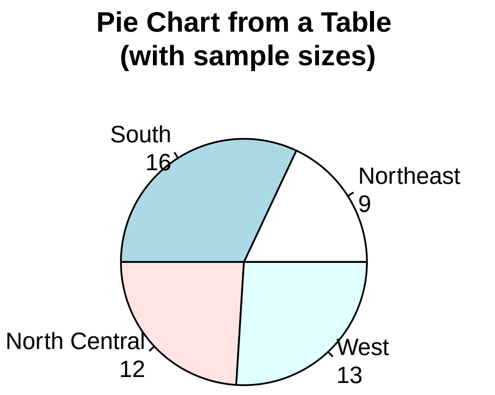
```
par(mfrow = c(2, 2))
#1
slices <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
pie(slices, labels = lbls, main = "Simple Pie Chart")
#2
pct <- round(slices/sum(slices) * 100)
```

```
lbls2 <- paste(lbls, " ", pct, "%", sep = "")
pie(slices, labels = lbls2, col = rainbow(length(lbls)),
    main = "Pie Chart with Percentages")
#3
library(plotrix)
pie3D(slices, labels = lbls, explode = 0.1, main = "3D Pie Chart ")
#4
mytable <- table(state.region)
lbls <- paste(names(mytable), "\n", mytable, sep = "")
pie(mytable, labels = lbls, main = "Pie Chart from a Table\n (with sample sizes)")
```





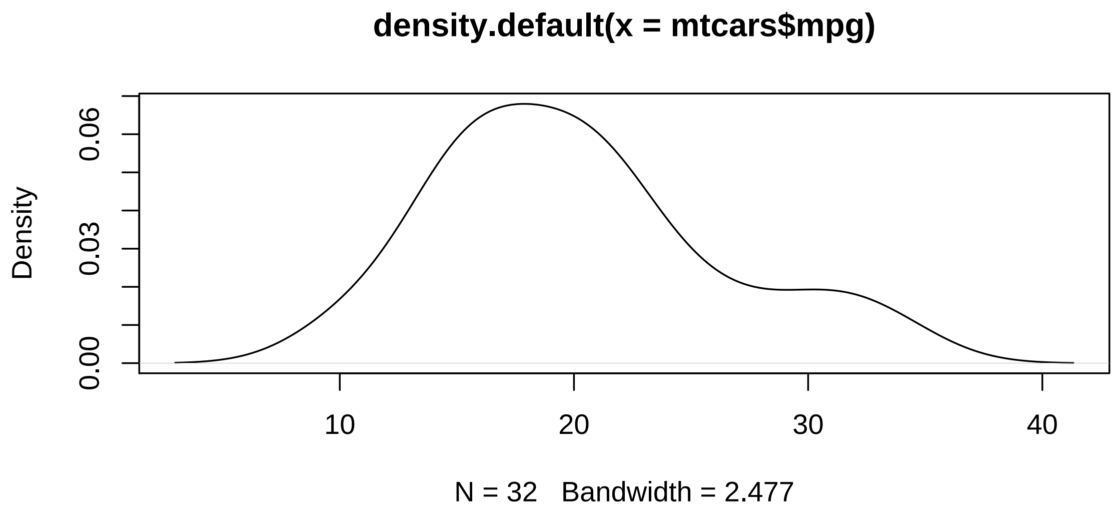


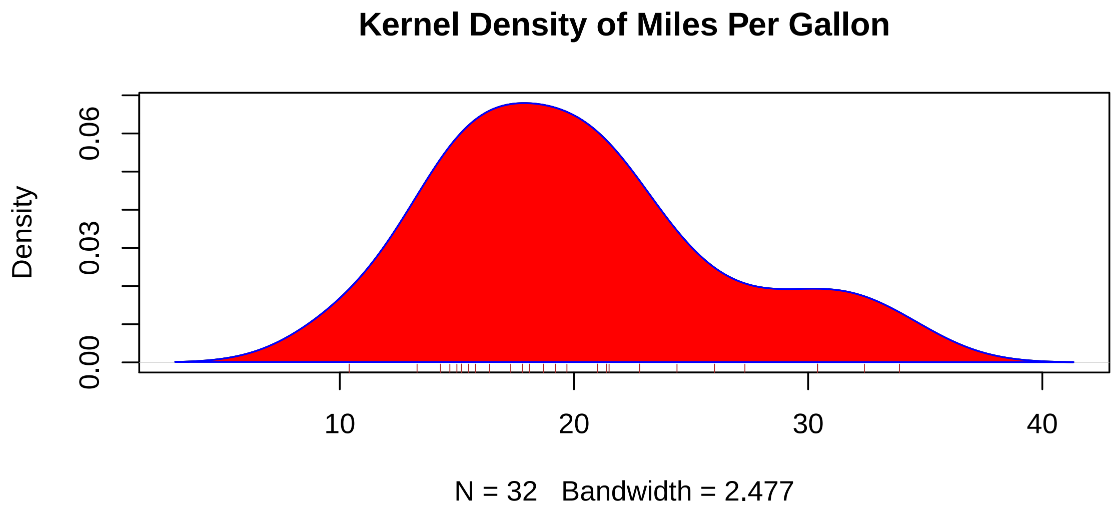


1.4 Kernel density plots

The format for a density plot (that's not being superimposed on another graph) is plot(density(x)).

```r
par(mfrow = c(2, 1))
d <- density(mtcars$mpg)
plot(d)
d <- density(mtcars$mpg)
plot(d, main = "Kernel Density of Miles Per Gallon")
polygon(d, col = "red", border = "blue")
rug(mtcars$mpg, col = "brown")
```





Kernel density plots can be used to compare groups. This is a highly underutilized approach, probably due to a general lack of easily accessible software. Fortunately, the sm package fills this gap nicely. The sm.density.compare() function in the sm package allows you to superimpose the kernel density plots of two or more groups. The format is sm.density.compare(x, factor). install.packages("sm")

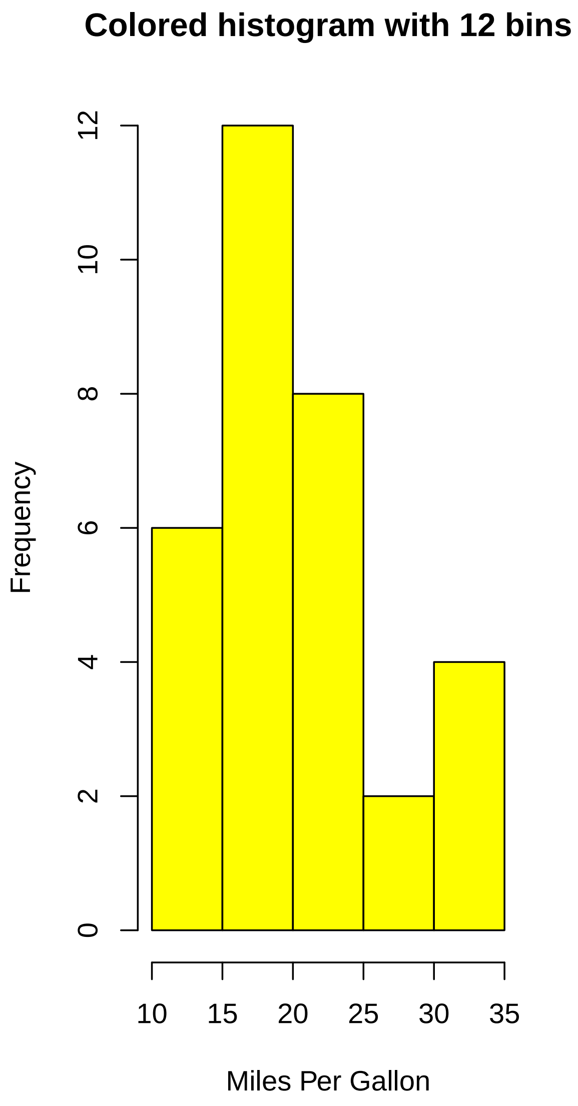
```r
par(lwd = 2)
library(sm)
attach(mtcars)
cyl.f <- factor(cyl, levels = c(4, 6, 8), labels = c("4 cylinder", "6 cylinder", "8 cylinder"))
sm.density.compare(mpg, cyl, xlab = "Miles Per Gallon")
title(main = "MPG Distribution by Car Cylinders")
colfill <- c(2:(1 + length(levels(cyl.f))))
cat("Use mouse to place legend...", "\n\n")
legend(locator(1), levels(cyl.f), fill = colfill)
detach(mtcars)
par(lwd = 1)
```

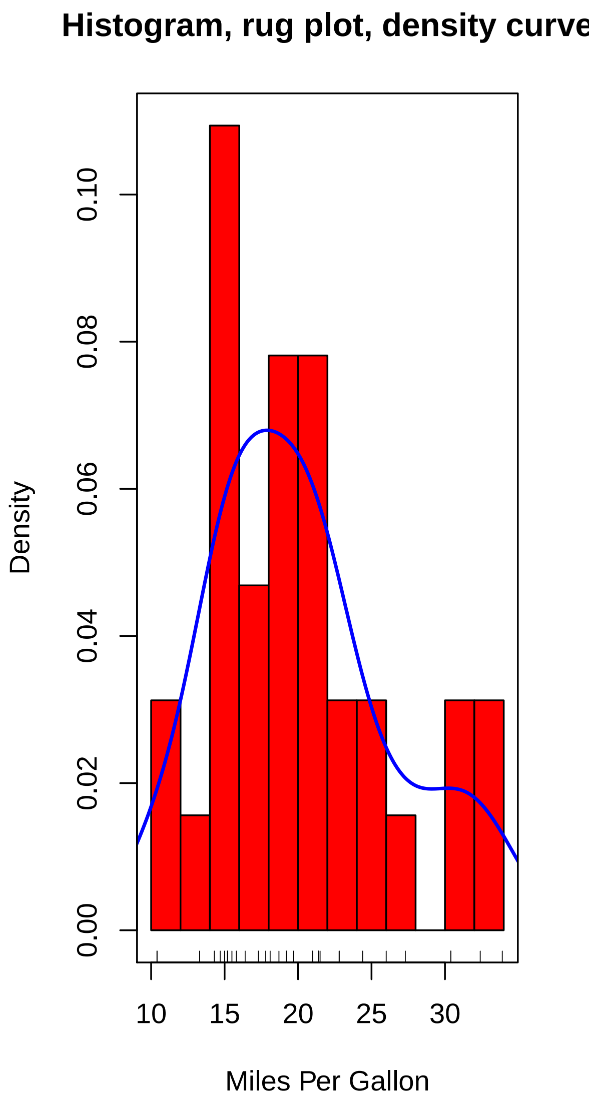
In the mtcars data frame, the variable cyl is a numeric variable coded 4, 6, or 8. cyl is transformed into a factor, named cyl.f, in order to provide value labels for the plot. The sm.density.compare() function creates the plot and a title() statement adds a main title. Finally, you add a legend to improve interpretability.First, a vector of colors is created. Here colfill is c(2, 3, 4). Then a legend is added to the plot via the legend() function . The locator(1) option indicates that you'll place the legend interactively by clicking on the graph where you want the legend to appear. The second option provides a character vector of the labels. The third option assigns a color from the vector colfill to each level of cyl.f.

1.5 Histograms

Histograms display the distribution of a continuous variable by dividing up the range of scores into a specified number of bins on the x-axis and displaying the frequency of scores in each bin on the y-axis.

```r
par(mfrow = c(1, 2))
hist(mtcars$mpg, breaks = 7, col = "yellow", xlab = "Miles Per Gallon",
    main = "Colored histogram with 12 bins")
hist(mtcars$mpg, freq = FALSE, breaks = 12, col = "red",        xlab = "Miles Per Gallon",
    main = "Histogram, rug plot, density curve")
rug(mtcars$mpg)
lines(density(mtcars$mpg), col = "blue", lwd = 2)
box()
```



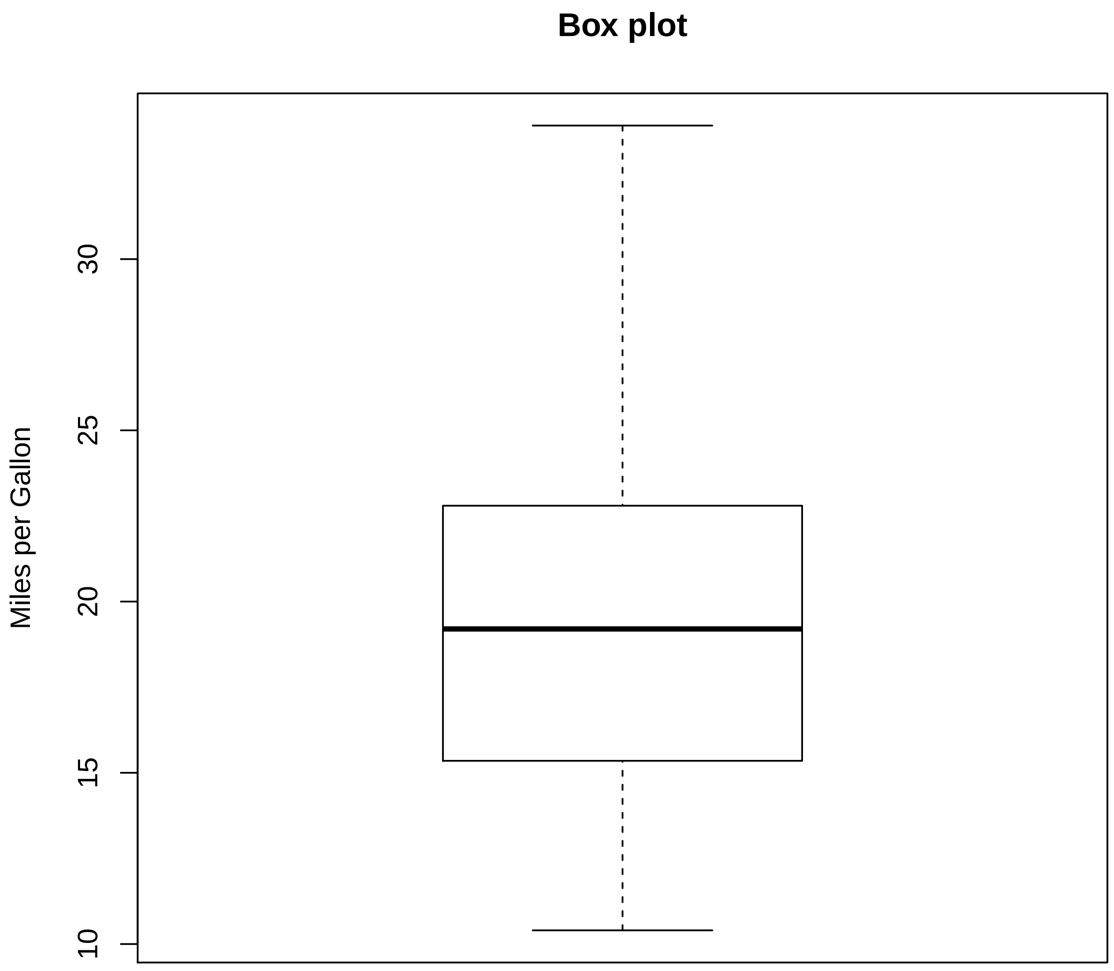


The first histogram demonstrates the default plot when no options are specified. In this case, five bins are created, and the default axis labels and titles are printed. For the second histogram, you've specified 12 bins, a red fill for the bars, and more attractive and informative labels and title. The third histogram maintains the colors, bins, labels, and titles as the previous plot, but adds a density curve and rug plot overlay.

1.6 Box plots

A "box-and-whiskers" plot describes the distribution of a continuous variable by plotting its five-number summary: the minimum, lower quartile (25th percentile), median (50th percentile), upper quartile (75th percentile), and maximum. It can also display observations that may be outliers (values outside the range of ± 1.5*IQR, where IQR is the interquartile range defined as the upper quartile minus the lower quartile).

```
boxplot(mtcars$mpg, main="Box plot", ylab="Miles per Gallon")
```

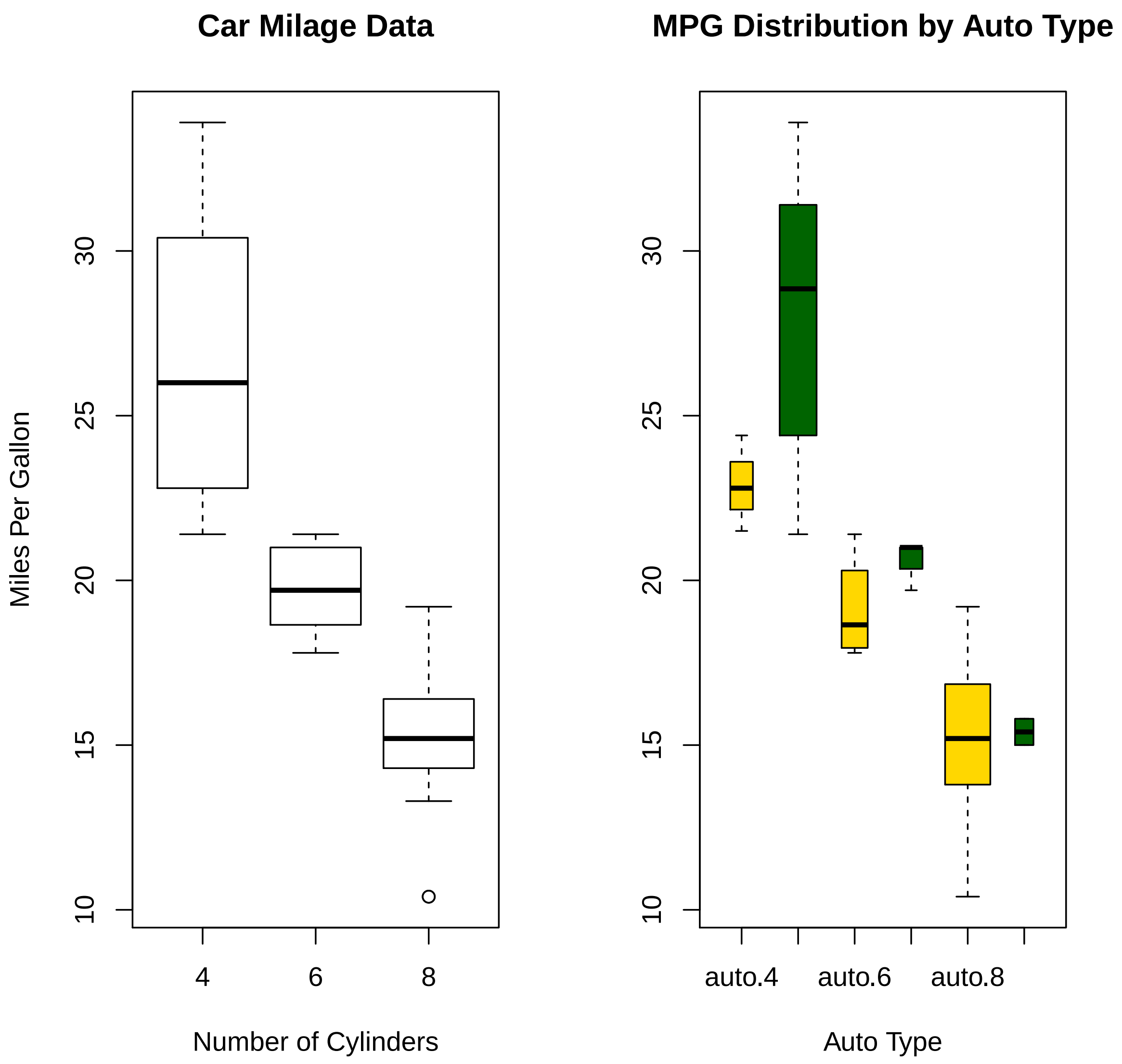


```
boxplot.stats(mtcars$mpg)

## $stats
## [1] 10.40 15.35 19.20 22.80 33.90
##
## $n
## [1] 32
##
## $conf
## [1] 17.11916 21.28084
##
## $out
## numeric(0)
```

Box plots can be created for individual variables or for variables by group. Adding the option varwidth=TRUE will make the box plot widths proportional to the square root of their sample sizes. Add horizontal=TRUE to reverse the axis orientation. You also can produce box plots for more than one grouping factor.

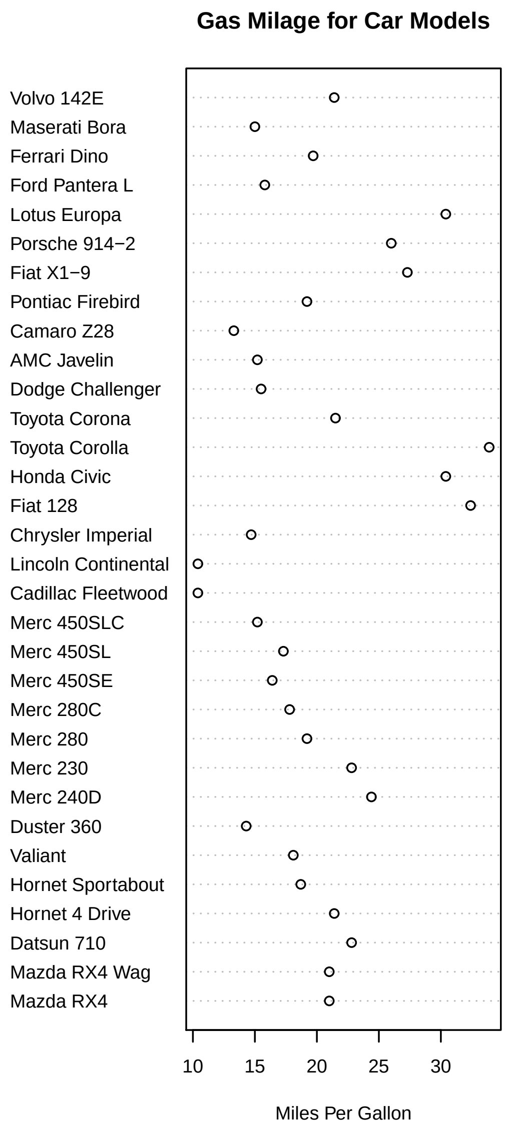
```r
par(mfrow = c(1, 2))
boxplot(mpg ~ cyl, data = mtcars, main = "Car Milage Data",
    xlab = "Number of Cylinders", ylab = "Miles Per Gallon")
mtcars$cyl.f <- factor(mtcars$cyl, levels = c(4, 6, 8),
    labels = c("4", "6", "8"))
mtcars$am.f <- factor(mtcars$am, levels = c(0, 1),
    labels = c("auto", "standard"))
boxplot(mpg ~ am.f * cyl.f, data = mtcars, varwidth = TRUE, col = c("gold", "darkgreen"),
    main = "MPG Distribution by Auto Type", xlab = "Auto Type")
```

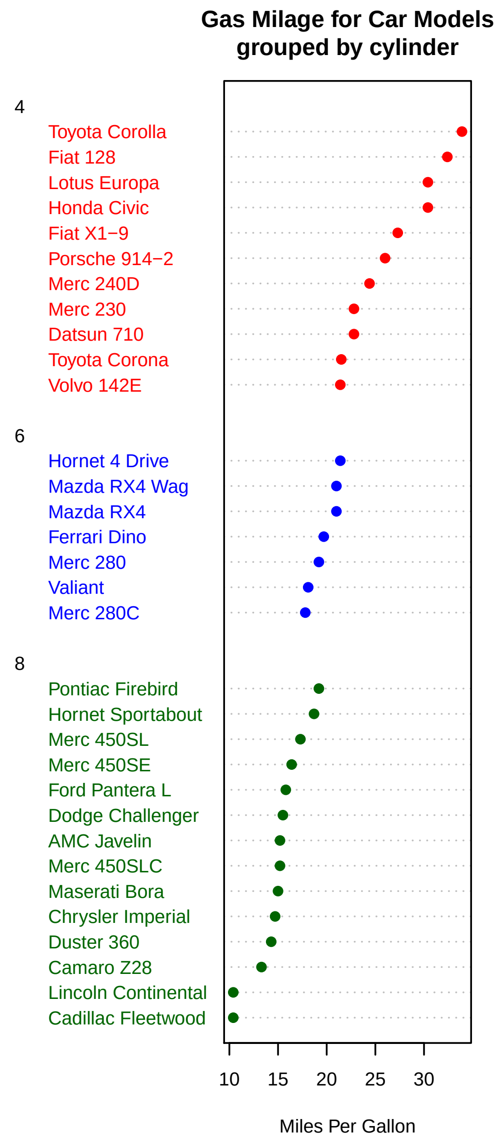


1.7 Dot plots

Dot plots provide a method of plotting a large number of labeled values on a simple horizontal scale. You can add a groups option to designate a factor specifying how the elements of x are grouped. If so, the option gcolor controls the color of the groups label and cex controls the size of the labels.

```r
#windows(width = 7, height = 4)
par(mfrow = c(1, 2))
dotchart(mtcars$mpg, labels = row.names(mtcars), cex = 0.7,
    main = "Gas Milage for Car Models", xlab = "Miles Per Gallon")
x <- mtcars[order(mtcars$mpg),]
x$cyl <- factor(x$cyl)
x$color[x$cyl == 4] <- "red"
x$color[x$cyl == 6] <- "blue"
x$color[x$cyl == 8] <- "darkgreen"
dotchart(x$mpg, labels = row.names(x), cex = 0.7, pch = 19,
    groups = x$cyl, gcolor = "black", color = x$color,
    main = "Gas Milage for Car Models\ngrouped by cylinder",
    xlab = "Miles Per Gallon")
```



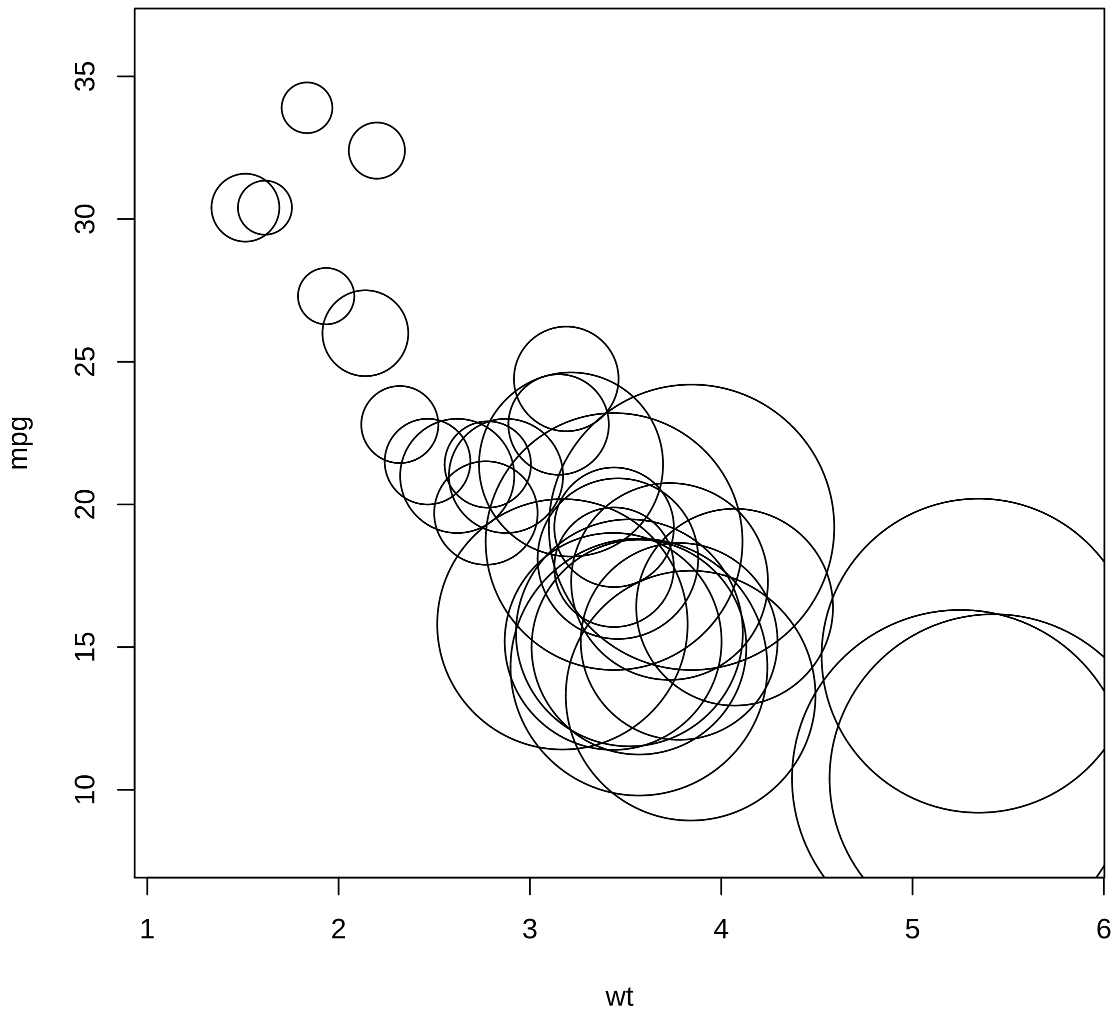


The data frame mtcars is sorted by mpg (lowest to highest) and saved as data frame x. The numeric vector cyl is transformed into a factor. A character vector (color) is added to data frame x and contains the values "red", "blue", or "darkgreen" depending on the value of cyl. In addition, the labels for the data points are taken from the row names of the data frame (car makes). Data points are grouped by number of cylinders. The numbers 4, 6, and 8 are printed in black. The color of the points and labels are derived from the color vector, and points are represented by filled circles. Additionally, the Hmisc package offers a dot plot function (aptly named dotchart2) with a number of additional features.

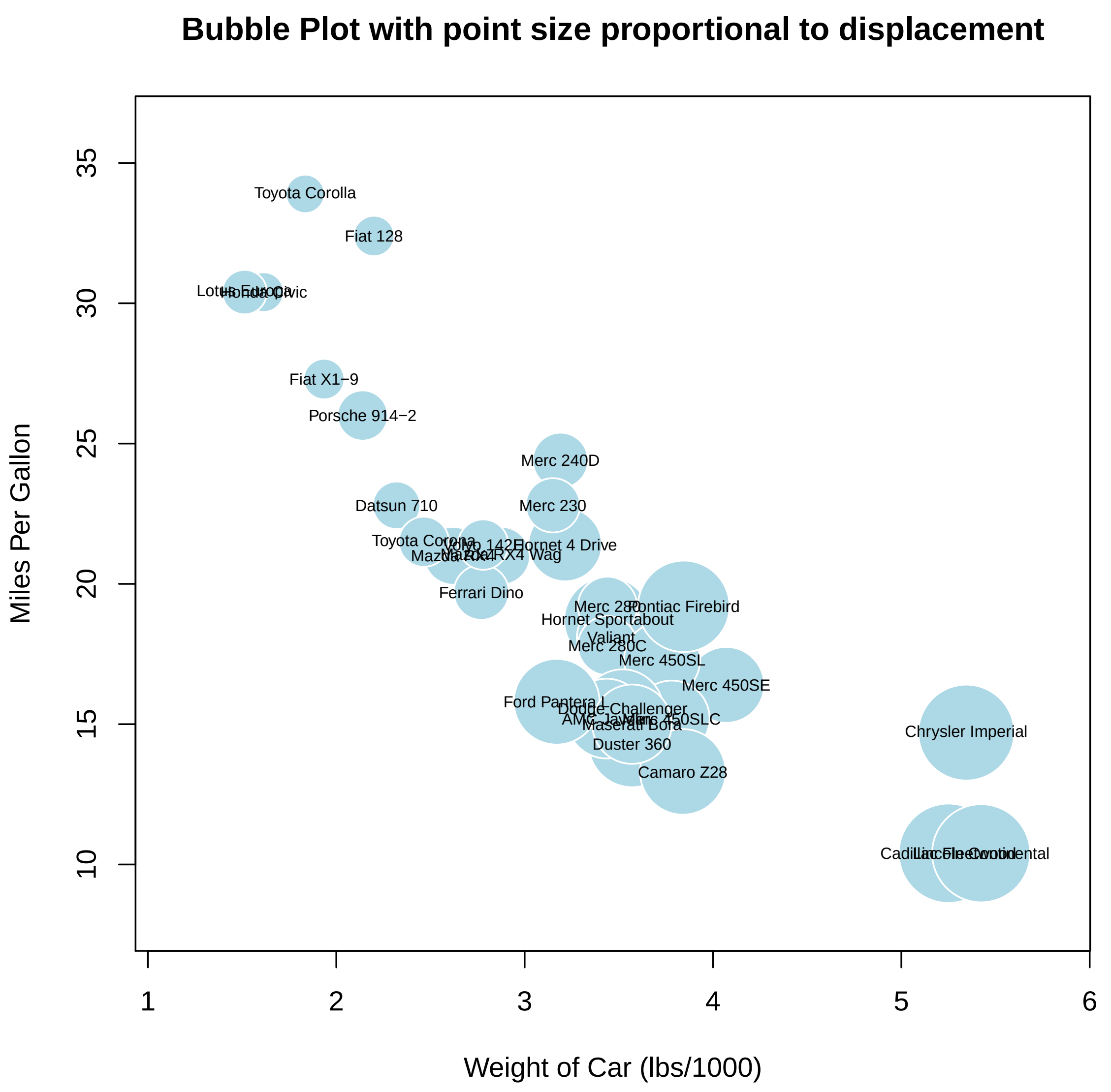
1.8 Bubble plots

You can create a bubble plot using the symbols() function. This function can be used to draw circles, squares, stars, thermometers, and box plots at a specified set of (x, y) coordinates.

```
attach(mtcars)
symbols(wt, mpg, circle=disp)
```



```
r <- sqrt(disp/pi)
symbols(wt, mpg, circle=r, inches=0.30,
fg="white", bg="lightblue",
main="Bubble Plot with point size proportional to displacement",
ylab="Miles Per Gallon", xlab="Weight of Car (lbs/1000)")
text(wt, mpg, rownames(mtcars), cex=0.6)
```



```
detach(mtcars)
```

The option inches is a scaling factor that can be used to control the size of the circles (the default is to make the largest circle 1 inch). The text() function is optional. Here it is used to add the names of the cars to the plot. From the figure, you can see that increased gas mileage is associated with both decreased car weight and engine displacement.

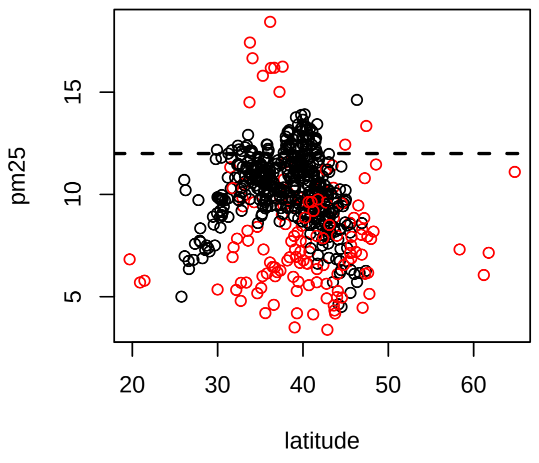
1.9 Contour*

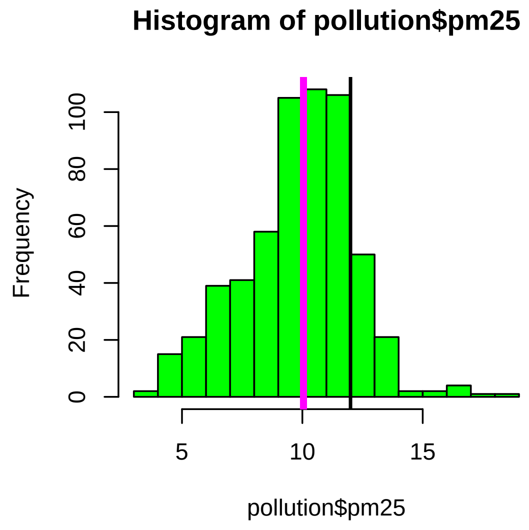
Create a contour plot, or add contour lines to an existing plot.

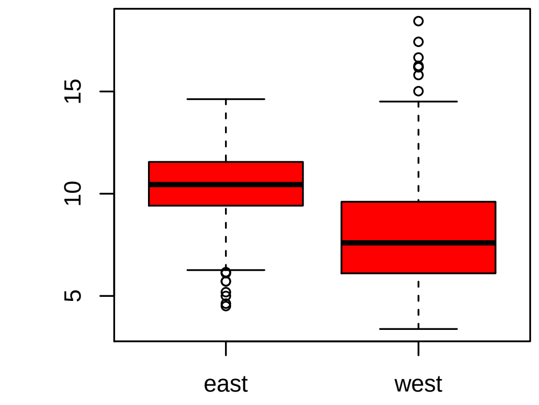
1.10 Exercise (Air Pollution in the United States)

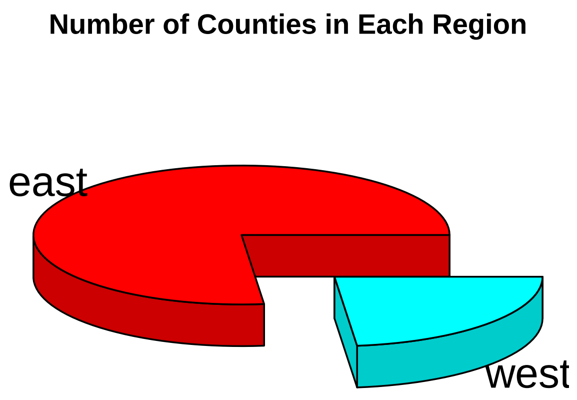
The U.S. Environmental Protection Agency (EPA) sets national ambient air quality standards for outdoor air pollution (U.S. National Ambient Air Quality Standards).

For fine particle pollution (PM2.5), the "annual mean, averaged over 3 years" cannot exceed $12\mu g/m^3$. Data on daily PM2.5 are available from the U.S. EPA web site (EPA Air Quality System). The question is "Are there any counties in the U.S. that exceed that national standard for fine particle pollution?" Plot the following Graphs.


pm25
latitude


Histogram of pollution$pm25
Frequency
pollution$pm25


east
west


Number of Counties in Each Region
east
west