The Basics of Plotting in R
R has a built-in Datasets Package:

- iris
- mtcars
- precip
- faithful
- state.x77
- USAArrests
- presidents
- ToothGrowth
- USJudgeRatings

You can call built-in functions like `hist()` or `plot()` on a built-in data set to quickly produce a chart
Basic plotting in R

```
hist(nhtemp)
plot(nhtemp)
```
Basic plotting in R

attach(iris)
plot(Petal.Length ~ Petal.Width)
Basic plotting in R

attach(iris)
plot(Petal.Length ~ Petal.Width, col = Species)
Basic plotting in R

plot(Petal.Length ~ Petal.Width, col = Species, main = "Iris: Petal width vs. length")
Basic plotting in R

attach(faithful)

duration = eruptions
waiting = waiting

plot(duration, waiting, xlab = "Eruption duration in minutes", ylab = "Time waited in minutes")

plot(duration, waiting, xlab = "Eruption duration in minutes", ylab = "Time waited in minutes", col = "seagreen")
Histograms

Histogram of waiting

Histogram of eruptions

\texttt{hist(waiting)}

\texttt{hist(eruptions)}
Histograms

`hist(waiting, breaks = 5)`  
`hist(waiting, breaks = 20)`
Histagrams

```r
hist(waiting,
    main = "Old Faithful",
    cex.main = 1.75,
    xlab = "Waiting Time in Minutes",
    ylab = "Frequency",
    cex.lab = 1.25)
```
Histograms

```r
hist(waiting,
    main = "Old Faithful",
    cex.main = 1.75,
    xlab = "Waiting Time in Minutes",
    ylab = "Frequency",
    cex.lab = 1.25
    col = "blue",
    border = "orange")
```
Stem and leaf plot

> stem(waiting)

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

4 | 3
4 | 5556666677778888999999
5 | 000001111122222333334444444444
5 | 555555666677788889999999
6 | 00000022223334444
6 | 555667899
7 | 000011111233333333444444
7 | 555555555566666666667777777777788888888888888888899999999999
8 | 000000000111111111111122222222222233333333333333444444444444
8 | 55555566666667788888888899
9 | 00000012334
9 | 6
> stem(eruptions)

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

```
  16 | 070355555588
  18 | 00002223333335577777778888223357778888
  20 | 00002223378800035778
  22 | 002335578023578
  24 | 00228
  26 | 23
  28 | 080
  30 | 7
  32 | 2337
  34 | 250077
  36 | 0000823577
  38 | 233335582225577
  40 | 00000335778888002233555577778
  42 | 0335555778800233335555577778
  44 | 02223355577800000002333357778888
  46 | 000023335770000023578
  48 | 0000022335800333
  50 | 0370
```
Bar charts

```r
> races_younger
[1] "Black"  "Hispanic"  "Other"  "White"

> population_in_millions_younger
[1] 10.76 19.03  7.76 40.50

> barplot(population_in_millions_younger,
          names.arg = races_younger,
          main = "Younger than 18, 2014",
          ylab = "Population in Millions")
```
Bar charts

> races_all
[1] "Native" "Asian" "Black" "Hispanic" "Two" "White"

> population_in_millions_all
[1] 1.39 18.12 38.60 55.61 5.67 195.35

barplot(population_in_millions_all,
names.arg = races_all,
main = "All ages, 2014",
ylab = "Population in Millions")
Bar charts

> races_all_other
[1] "Black" "Hispanic" "White" "Other"

> population_in_millions_all_other
[1] 38.60 55.61 195.35 25.19

barplot(population_in_millions_all_other,
        names.arg = races_all_other,
        main = "All ages, 2014",
        ylab = "Population in Millions")
Stacked and grouped bar charts
Pie charts

define the labels for the pie chart:

```r
coutesy_labels <-
paste(races_coutesy, 
counters_population_in_millions_coutesy, 
sep = "\n")
```

create the pie chart:

```r
coutesy_population_in_millions_coutesy, 
main = "Younger than 18\nPopulation in Millions",
labels = coutesy_labels_coutesy)"
pie_labels_all_other <-
paste(races_all_other, 
population_in_millions_all_other, 
sep = "\n")

pie(population_in_millions_all_other, 
main = "All Ages\nPopulation in Millions",
labels = pie_labels_all_other)
Pie charts

**Younger than 18 Population in Millions**
- Hispanic: 19.03
- Black: 10.76
- White: 40.5
- Other: 7.76

**All Ages Population in Millions**
- Hispanic: 55.61
- Black: 38.6
- White: 195.35
- Other: 25.19
Dot plots

```r
states <- data.frame(state.x77)
sorted_states <- states[order(states$Income), ]
dotchart(sorted_states$Income,
        rownames(sorted_states),
        cex = .25,
        main = "Income per capita")
```
Box plots

boxplot(Petal.Length ~ Species,
       main = "Iris Petal Length by Species",
       xlab = "Species",
       ylab = "Petal Length")
Notched Box plots

attach(ToothGrowth)

boxplot(len ~ supp * dose,
       data = Toothgrowth,
       notch = TRUE,
       col = (c("gold", "darkgreen")),
       main = "Tooth Growth",
       xlab = "Supplement and Dose")

If the notches do not overlap, then the medians of the groups are different (because their confidence intervals do not overlap).
Simple Scatterplots

```R
plot(USJudgeRatings$RTEN ~ USJudgeRatings$FAMI,
xlab = "Familiarity with Law",
ylab = "Worthy of Retention",
main = "Law Familiarity vs. Worthy of Retention",
pch = "+")
```
Scatterplot Matrices

CONT Number of contacts of lawyer with judge.
INTG Judicial integrity.
DMNR Demeanor.
DILG Diligence.

pairs(~USJudgeRatings$CONT +
  USJudgeRatings$INTG +
  USJudgeRatings$DMNR +
  USJudgeRatings$DILG,
  main = "US Judge Ratings")
features <- c("Contacts", "Integrity", "Demeanor", "Diligence")

pairs(~USJudgeRatings$CONT +
     USJudgeRatings$INTG +
     USJudgeRatings$DMNR +
     USJudgeRatings$DILG,
     labels = features,
     main = "US Judge Ratings")
3D Scatterplots

```r
library(scatterplot3d)

scatterplot3d(mtcars$wt, mtcars$disp, mtcars$mpg,
             main = "3D Scatterplot")
```
Graphical parameters

- **Text and symbol size** (*cex*: `.axis`, `.lab`, `.main*)
- **Fonts** (*font*: `.axis`, `.lab`, `.main*):
  - 1=plain, 2=bold, 3=italic, 4=bold italic, 5=symbol
- **Colors** (*col*): [colors](http://www.statmethods.net/advgraphs/parameters.html), and more [colors](http://www.statmethods.net/advgraphs/parameters.html)
Graphical parameters (cont’d)

Plotting symbols (**pch**)

Line type (**lty**)