STA 141A
Fundamentals of Statistical Data Science

Fall 2016

Instructor: Debashis Paul

Lecture 3

Basic plotting functions in R

- plot( ) # draw a scatterplot
- points( ) # adding points to an existing plot
- pairs( ) # draw a scatter plot matrix (for 2 or more variables)
- lines( ) # joining points (based on criteria) with straight line segments
- matplot( ) # plotting columns of a matrix against a variable
- abline( ) # draw a single straight line with specified intercept and slope

A simple graph

- We create graph of a function of x and add some stuff to it.

```
x = seq(0,1,0.05)
y = sin(2*pi*x)*exp(-4*x^2)
plot(x,y)  # plot of y vs. x, points indicated by "o"
plot(x,y,pch=2) # replaces the "o" by triangles (pch stands for point character)
plot(x,y,pch=2,type="b") # joins the points by line segments
plot(x,y,pch=2,type="b",col=3)  # uses green color (color = 3) for the graph
plot(x,y,pch=2,type="b",col=3,cex=0.5) # make the symbols smaller (by factor cex)
plot(x,y,type="l") # only plot the lines joining the points, not the points themselves
```

Overlay a graph and customize

```
z = 0.4+0.4*cos(2*pi*x)
points(x,z,type="l",col=2,lty=2)  # add the graph of z vs. x as a broken red line
abline(a=0.5,b=0.3) # draw a line in current graph with intercept 0.5 and slope 0.3
title("Graph of y and z vs. x")  # adds a title to the graph
text(0.4,0.4,"Graph of y vs. x",col=1)  # adds a text centered at (0.4,0.4)
legend(0.5,0.7,legend=c("y", "z"),col=c(1,2),lty=c(1,2)) # adds a legend centered at (0.5,0.7)
par(mfrow=c(1,2))  # creates a matrix of plots
plot(x,y,type="l",main="left panel", xlab="x", ylab="y")  # on left panel, plot y vs x
plot(x,z,type="l", main="right panel",xlim=c(0,1),ylim=c(0,0.85))  # on right panel, plot z vs x
```

Some functions for customizing plots

- par( ) # used to set graphical parameters
- axis( ) # customizes axes of a plot
- legend( ) # add a legend to a plot
- text( ), mtext( ) # add text to a plot
- title( ) # add a title to a figure
- box( ) # draw a box around a current plot
- rectangle( ), polygon( ), # draw a rectangle, or a polygon

Basic graphical statistical summaries

- hist( ) # draw histogram (of numeric data)
- boxplot( ) # draw Boxplot (visual description of five-point summary)
- dotchart( ) # draw Dot chart (of numeric data)
- barplot( ) # draw Bar plot
- pie( ) # draw Pie chart (of categorical data)
- qqplot( ), qqline( ), qqnorm( ) # draw Q-Q (quantile vs quantile) plot

Histogram and density plot : Old faithful

- Illustrate the use of hist( ) and density( ) functions through the eruption recordings of Old Faithful geyser. The data set is named **faithful** in R. It is a data frame (more on it soon).

dim(faithful) # returns c(272,2)

head(faithful,10) # shows first 10 rows; column names **eruptions** and **waiting**

faithful**$**eruptions # first column, same as faithful[,1]

attach(faithful) # useful if we simply want to work with individual columns

hist(eruptions,freq=F) # plots relative frequency histogram of eruptions

points(density(eruptions),type="l") # overlays a kernel density estimator of eruptions

rug(eruptions) # shows the actual observations as tick-marks along the bottom ("rug plot")

More on Matrices

```
M = matrix(nrow=2,ncol=3)   # matrix with 2 rows and 3 columns with all entries NA
M[1,] = c(3,-4,6)  # assign the values of the first row of matrix M
M[2,c(1,3)] = c(-7,5)  # assign values to (2,1) and (2,3) elements of matrix M
dim(M) # dimension of M ; returns c(2,3)
length(M) # returns 6; reminder: a matrix is actually a vector (stack the columns)
which(is.na(M)) # returns 4, the fourth element in as.vector(M), which is still NA
sum(M^2, na.rm=T) # computes sum of squares of elements of M, excluding NAs
M[2,2] = 0 # assign the value zero to (2,2) element of M
M[ ,-2]  # 2 x 2 matrix consisting of 1st and 3rd  columns of matrix M
```

Matrix operations

M %*% c(4,-3,1) # multiply matrix M to vector c(4,-3,1)

M + c(1,-2) # add the column vector c(1,-2) to each column of M

M2 = cbind(M,c(4,5)) # create matrix M2 by appending column c(4,5) to M

M3 = rbind(M2,c(5,6,-4)) # create matrix M3 by appending row c(5,6,-4) to M2

onealt = rep(c(1,-1),4) # creates the vector c(1,-1,1,-1)

M4 = cbind(onealt,c(-2,4,1,7)) # 4 x 2 matrix 1st column c(1,-1,1,-1) and 2nd column c(-2,4,1,7)

- For easier access of a data set, expressed as a matrix, it is useful to name its rows and/or columns

rownames(M) = c("first", "second") # name the rows of M

colnames(M) = c("a", "b","c") # name the columns of M

identical(M[,"a"],M[,1]) # returns TRUE

Data frame : First look

- Data frames are like matrices, having rows and columns. However they differ from matrices in that, different columns may have different mode. You can create data frames by combining vectors

```
name = c("Bob", "Jane")
GPA = c(3.2,3.7)
age = c(19,21)
students = data.frame(name,age,GPA,stringsAsFactors=FALSE)  # creates the data frame "students"
str(students) # displays the structure of the data frame "students"
students[[1]] # returns the vector c("Bob", "Jane")
students$GPA # returns c(3.2,3.7)
students[,2] # returns c(19,21), shows that a data frame is treated like a matrix
```

Basic operations on data frames

- You can perform standard matrix operations on a data frame

```
students[1,2:3] # extracts the age and GPA of student number 1 (“Bob”)
students[students$age>20, ]  # returns the subdata frame corresponding to “Jane”
subset(students,age>20) # does the same job as the command above
students = rbind(students,c(“Ashley”,20,3.1)) # adds a row to the data frame
students = rbind(students,list(“Jack”,21,3.4)) # adds another row
students = cbind(students,list(resident=c(T,T,T,F))) # adds a new column resident
```

- You can import tabular data into R as a data frame (Discussion Section on 09/30)