STA 141A
Fundamentals of Statistical Data Science

Fall 2016

Instructor: Debashis Paul

Lecture 7

Revised agenda

- Quiz [less than 10 minutes]
- Applying functions to data frames
- Using various R packages for visualizing multivariate data

1. scatter plot matrix
2. lowess smoothing
3. 3D scatter plot
4. bubble plot
5. correlogram

Quiz #1

For each of the following questions, only write down the R code (not output) as your answer. No need to copy the question.

1. Write a code (without using for or while loop) that, given a vector x with numerical values, creates a vector y with entries equal to “nonnegative” if the corresponding entry of x is nonnegative, and “negative” otherwise.
2. Write an R function that, given x, returns the value of a cubic polynomial in x, where the coefficients of the polynomial are a0, a1, a2, a3 (given).

Applying functions to data frames

```
dfx = data.frame(
         group = c(rep('A', 8), rep('B', 15), rep('C', 6)),
        gender = sample(c("M", "F"), size = 29, replace = TRUE),
        age = round(runif(n = 29, min = 18, max = 54))  )
attach(dfx)
# Using tapply( )
tapply(age, list(group,gender),mean)  # compute mean categorized by levels of  group and gender
# Using aggregate( )
aggregate(age, list(group,gender),sd) # compute mean aggregated by levels of  group and gender
# returns data frame with columns Group.1, Group.2  (group and gender) and sd(age) within strata
```

Use of **plyr** package

- Function ddply( ) splits data according to values of variables and applies functions to resulting strata

```
library(plyr)
ddply(dfx, ~gender)  # sorts records according to values of gender
ddply(dfx, ~gender, nrow) # counts number of rows for different values of gender
# Compute statistics within strata by making use of “summarize” function within ddply( )
ddply(dfx, .(group, gender), summarize, mean = mean(age), sd = round(sd(age), 2))
# Use of the “.” function allows group and gender to be used without quoting
ddply(dfx, c("group", "gender"), summarize, mean = mean(age), sd = round(sd(age), 2)) # same as above
# Can compute summary for different variables within strata defined by a (or a group of) variable(s)
ddply(dfx,.(group), summarize, count_female=sum(gender=="F"), mean_age=mean(age))
```

Statistically Informed Data Visualization

- The main goal is to convey information about the relationship among variables for a multivariate data using different visual representation of the data.
- **Useful basic constructs:** (i) scatter plots subdivided or conditioned on categorical variables; (ii) local smoothing (easy-to-understand visual description of approximate relationship among pairs of variables); (iii) pairwise scatter plot; (iv) 3D scatter plot for better visual representation of three variables; (v) bubble plots
- Want to display multiple numerical summaries in an easily interpretable way. Examples: Subdivided boxplots, Correlograms.
- Want to have mechanisms that allow us to display large amount of data without being unduly affected by congestion and occlusion.

An exploratory analysis : mtcars data

```
data(mtcars)  # part of available data sets in R
```

- The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

(Reference: Henderson and Velleman (1981). Building multiple regression models interactively. *Biometrics*, **37**, 391-411.)

```
attach(mtcars)
plot(wt,mpg, main = "Scatter plot of MPG vs. Weight”,
    xlab= "Weight of car (in 1000 lbs)", ylab = "Miles per Gallon", pch = 10)
abline(lm(mpg ~ wt), col="red", lwd=2, lty = 1) # plots least squares regression line of "mpg" on "wt”
lines(lowess(wt,mpg), col="blue", lwd=2, lty=2) # plots  lowess regrssion (basic scatterplot smoother) line
```

Enhanced plotting using **car** package

```
library(car)
# Draw scatter plot of mpg vs. wt, add least squares line and lowess smoothers,  grouped by values of cyl
scatterplot(mpg ~ wt | cyl,    data = mtcars,     lwd=2,
            main = "Scatter plot of MPG vs Weight by # of Cylinders”,
            xlab = "Weight of car (in 1000 lbs)", ylab = "Miles per Gallon",
             legend.plot = TRUE, legend.coords = "topleft”,  # adds legend and controls its position
             boxplot = "xy",        # adds boxplots of wt and mpg on margins of x and y axes
             span = 0.9,              # controls degree of smoothing for lowess smoother
             id.method = "identify",        # allows us to identify points by their labels
             labels=row.names(mtcars))      # row names are to be used as labels
```