STA 141A
Fundamentals of Statistical Data Science

Fall 2016

Instructor: Debashis Paul

Lecture 8

Scatter plots using **car** package

- Function scatterplot( ) allows us to plot bivariate scatter plot, conditional on a third (typically categorical) variable.

```
library(car)

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)
```

- The function also adds a least squares fit and lowess smoother to each stratum of the data formed by different values of *cyl*. Optional argument *span* can be used to set the bandwidth of the lowess smoother. Different colors and symbols are used to distinguish the strata.

Scatter plot matrix

- We can instead use the function scatterplotMatrix( ) in package **car** to plot all the pairwise scatter plots for a set of variables, grouped by a categorical variable (optional), overlaid with the least squares regression line fit and a lowess smoother fit (where the variables in the x and y axes are treated as explanatory and dependent variables, respectively).

```
scatterplotMatrix( ~ mpg + disp + drat + wt | cyl , data = mtcars,
        spread=FALSE,  span = 0.8,  diagonal = "histogram”,
              main = "Scatter plot matrix grouped by # cylinders")
```

- The option *diagonal* can be used to plot density (default) or boxplot instead of histogram along the diagonal.
- We can choose to omit the lowess smoother from the plots by setting *smooth=FALSE*.

3D scatter plots

- We can use package **scatterplot3d** to plot static 3-dimensional scatter plots. We can also add supporting vertical lines to the points plotted for better visibility. In addition, we can fit a regression plane (with the variable on the z axis being the dependent variable and the ones on the x and y axes being explanatory variables) and display it.

```
library(scatterplot3d)
cars.s3d = scatterplot3d(wt, disp, mpg,
              scale.y = 1, angle = 40, highlight.3d = TRUE, type="h",
   main = "3D scatter plot of wt (x), disp (y) and mpg (z) with lines and regression plane")
cars.fit = lm(mpg ~ wt + disp)
cars.s3d$plane3d(cars.fit)
```

Spinning 3D plots using package **rgl**

- We can use the plot3d( ) function in package **rgl** to create a 3D scatter plot that can be rotated.

```
library(rgl)

plot3d(wt, disp, mpg, col="red", size=5)
```

- We can spin the plot around a pivot point by dragging it with a mouse pointer. Some particular projection may obscure specific features in the data, such as presence of clusters, due to occlusion.
- We can instead use a 3D perspective plot of a surface and spin it the same way using persp3d( ) .

```
z = 2 * volcano        # Exaggerate the relief of a volcano surface (data volcano is part of rgl package)

x =10 * (1:nrow(z))   # 10 meter spacing (S to N)

y = 10 * (1:ncol(z))   # 10 meter spacing (E to W)

persp3d(x, y, z, col = "green3", aspect = "iso") # perspective plot of surface z on the grid of x and y
```

Bubble plot using symbols( ) function

- Bubble plots are useful for plotting more than 2 quantitative variables on a 2D scatter plot, where the size and shape of the bubble carry information about the additional variables (beyond those represented by the x and y coordinates of the plots).
- Typically, when plotting 3 quantitative variables (with the third variable being nonnegative), we can set the area of the bubbles (disks) to be proportional to the third variable.

```
attach(mtcars)

rad = sqrt(disp/pi)

symbols(wt, mpg, circle=rad, inches=0.3, fg = "white", bg = "lightblue”,
    main = "Bubble plot with bubble size proportional to displacement”,
        ylab = "Miles per Gallon”, xlab = "Weight of car (in 1000 lbs)")
```

Correlogram using package **corrgarm**

- We can compute the covariance and correlation matrices for a data frame, and use corrgram( ) function in package **corrgram** to visually represent the latter.

```
cov(mtcars) # covariance matrix of variables in mtcars

cor(mtcars) # correlation matrix of variables in mtcars

library(corrgram)

corrgram(mtcars, order = TRUE, lower.panel = panel.shade, upper.panel = panel.pie,
         text.panel = panel.txt, main = "Correlogram of mtcars data")
```

- For lower as well as upper panels (i.e., below and above diagonal) blue and red colors represent positive and negative correlation; darker shade indicates stronger correlation
- The pies above the diagonal contain the same information as the squares below the diagonal, where the strength of the correlation is displayed by the size of the pie slice

Correlogram : an alternative view

- The correlogram can be constructed in such a way that it contains additional information, such as, for each pair of variables, a scatter plot smoother, or the elliptical contour of a bivariate normal distribution fitted to the data.

```
corrgram(mtcars, order = FALSE,
        lower.panel=panel.ellipse, upper.panel = panel.pts,
        text.panel = panel.txt,
        diag.panel = panel.minmax,
        main = "Correlogram of mtcars data”)
```

High density scatter plot

- We can plot a smoothed density surface (with some outliers indicated by points) using function smoothScatter( )

```
n = 10000
c1 = matrix(rnorm(n, mean=0, sd = 0.5), ncol=2)
c2 = matrix(rnorm(n, mean=3, sd = 2), ncol=2)
datamat = as.data.frame(rbind(c1,c2))
# 10000 observations from mixture of two bivariate normals with equal mixture proportions
names(datamat) = c("x","y")
with(datamat,
    smoothScatter(x,y, bandwidth = 0.25, main="Scatterplot Colored by Smoothed Densities"))
```

High density scatter plot : package **hexbin**

- We can instead plot a bivariate histogram with hexagonal binning where the color or intensity of each bin indicates bin count (or frequency). For this we use function hexbin( ) in package **hexbin**.

```
library(hexbin)
bins = hexbin(datamat$x, datamat$y, xbins=50)
# argument xbins specifies the number of subvisions along x axis, which in turn
# determines the number of hexagonal bins
plot(bins, main = "Hexagonal binning with 10000 observations")
```