—— WORKSHOP ——

Applied Classical and Modern Multivariate Statistical Analysis

Module 2-1: Multivariate Data Visualization

Weixing Song, Juan Du

Department of Statistics
Kansas State University

August 14, 2018

Outline

1. Why Graphics?
2. Display for Univariate Data
3. Display for Bivariate Data
4. Displays for Three-Dimensional Data

Outline

1 Why Graphics?

2 Display for Univariate Data

3 Display for Bivariate Data

4 Displays for Three-Dimensional Data

You can observe a lot by just watching.

— *Yogi Berra*

R Graphics: Basics

`plot` function:

```
> x=seq(0,12,by=0.1)
> plot(x, dchisq(x,4),type="l")
> lines(x,dchisq(x,5))
> lines(x,dchisq(x,6))
> lines(x,dchisq(x,7))
> pdf(file="chisquare.pdf")
> plot(x, dchisq(x,4),type="l",ylab="Chi-squared density")
> lines(x,dchisq(x,5))
> lines(x,dchisq(x,6))
> lines(x,dchisq(x,7))
> text(c(3,4.2,5.2,6.2),c(0.18,0.153,0.137,0.123),
      labels=c("4 df", "5 df","6 df", "7 df"))
> dev.off()
```

Outline

1 Why Graphics?

2 Display for Univariate Data

3 Display for Bivariate Data

4 Displays for Three-Dimensional Data

Example: We consider the `CD4` data in the `boot` library. This data set presents the baseline counts of CD4 cells of patients at the time of enrollment in the clinical trial and again after 1 year of treatment.

```
> install.packages("boot")
> require(boot)
> cd4
> head(cd4)
> hist(cd4$baseline)
> stem(cd4$baseline)
> plot(cd4$baseline,cd4$oneyear,xlab="Baseline",ylab="At 1 year")
> library(aplpack)
> stem.leaf.backback(cd4$baseline,cd4$oneyear)
```

Outline

1 Why Graphics?

2 Display for Univariate Data

3 Display for Bivariate Data

4 Displays for Three-Dimensional Data

Example: We consider the **housing** data from

http://www.rossmanchance.com/iscam2/data/housing.txt

This data set has five variables: `sqft`, `price`, `City`, `bedrooms`, `baths`.

```
# Load the data set
> house=read.table("http://www.rossmanchance.com/iscam2/data/housing.txt",
header=T,sep = "\t")
> attach(house)
> class(house)

# Scatter Plot
> plot(sqft,price)
> plot(house)

# Scatter plot with rug fringes
> require(graphics)
> plot(sqft,price,axes=F,cex=1.5,pch=16,col="red")
> rug(sqft,side=1)
> rug(price,side=2)
```

```
# Bivariate boxplot
> install.packages("MVA")
> require(MVA)
> bvbox(cbind(sqft,price),xlab="sqft",ylab="price",pch=19,cex=1.25,col="red")

# Convex hull
> ch=chull(cbind(sqft,price)) # Returns the indices of the data points
                               # that are most extreme in each direction.
> ch=c(ch,ch[1])
> plot(sqft,price,cex=1.5,pch=16,col="red")
> lines(sqft[ch],price[ch],type="l",col=3,lwd=2)

# A fancy convex hull
> library(aplpack)
> nlev=5;
> colors=heat.colors(9)[3:(nlev+2)]
> plothulls(c(sqft,price),n.hull=nlev, col.hull = colors,
     xlab="sqft",ylab="price",lty.hull=1:nlev,
     density=NA,col=0,main="")
> points(c(sqft,price),pch=16,cex=1,col="green")
```

Outline

1 Why Graphics?

2 Display for Univariate Data

3 Display for Bivariate Data

4 Displays for Three-Dimensional Data

Example: We consider the `JanTemp` data, which is attached in the distributed files. This data set contains six variables: `State, City, Temperature (Temp), Longitude (Long), Latitude (Lat), Altitude (Alt)`.

Bubble plot:

```
> myfile="C:/Users/weixi/Dropbox/Teaching/Stat730/slides/Module 2/janTemp.txt"
> JanTemp=read.table(myfile,header=T)
> require(MASS)
> attach(JanTemp)
> plot(Long,Lat,pch=16,cex=0.7,xlab="Longitude, east",ylab="Latitude",col="red")
> with(JanTemp,symbols(Long,Lat,circles=Alt,inches=0.3,add=T,lwd=3,fg="green"))
> landmarks=c("AK","HI","MA","PR")
> lmi=match(landmarks,State)
> text(Long[lmi],Lat[lmi],labels=landmarks,pos=c(1,1,4,3),col="blue")
```

Scatter plot matrix

```
> pairs( Long+Lat+Alt,pch=16,col="red")
> pairs(cbind(Long,Lat,Alt),pch=16,col="red")
> pairs(cbind(Long,Lat,Alt),lwd=3,pch=16,cex=1.25,col="red",gap=0,xaxt="n",
   yaxt="n",panel=panel.smooth,col.smooth="blue")
> library(lattice) #Scatter plot using lattice package
> splom(cbind(Long,Lat,Alt))
```

Surface, Contour Plots

```
> x=seq(-5,5,length=100);
> y=seq(-5,5,length=100);
> z=matrix(kronecker(x^2,y^2,"+"),nrow=100);
> persp(x,y,z,phi=45,theta=45,xlab="x",ylab="y",main="surface plot")
    # Surface plot using base function persp
> contour(x,y,z,nlevels=10) # contour plot
> library(lattice); # Surface plot using wireframe from lattice package
> g=expand.grid(x=seq(-5,5,length=50),y=seq(-5,5,length=50),gr=1);
> g$z=((g$x^2 + g$y^2));
> wireframe(g$z~g$x*g$y, shade=T,aspect = c(1,1),light.source=c(10,0,10));
```

Bag plot: The **`bagplot.pairs`** routine combines several features we have covered: the bivariate boxplot, the convex hull, and the pairs scatterplot. The bagplot contains a scatter plot of bivariate data. The outer polygon of each bagplot is a convex hull, excluding extreme outliers. The inner polygon (the "bag") of the bagplot contains the central 50% of the data. Data points between the two convex hulls are connected with lines to the central, bivariate median.

```
> library(aplpack)
> bagplot.pairs(cbind(Long,Lat,Alt),gap=0,col.baghull="green")
```

Conditioning plot: `coplot` gives a series of pair scatterplots of other variables, each representing a view of a slice of a third variable.

```
> library(graphics)
> coplot(Long Lat|Alt,rows=1,pch=16,cex=1.75,col="red",
      bar.bg=c(num="blue",fac=gray(0.95)),number=10)
```

Star Plot: The star plot summarizes each data item as a circle with different sized wedges. Every star is an individual (row) and every wedge on the star represents a different variable (column) in the data. The radius of each wedge in the star indicates the size of the scaled values for that individual, relative to all other measurements for that variable. These variables are listed in different colors and directions. The key for this is given in the lower right corner of the figure. The radius of each wedge is scaled and can be used to compare the relative magnitude of each variable values across all individuals.

```
> library(graphics) # for the stars program
> palette(rainbow(7)) # set colors
> alts=JanTemp[order(-JanTemp$Alt), ] # order by Alt, decreasing
> stars(alts[,-c(1,2)], len=1, cex=0.5, key.loc=c(17, 1),
     labels=row.names(alts), draw.segments=TRUE)
```

Chernoff face plot

```
> library(aplpack)
> faces(JanTemp[,-c(1,2)],face.type=1, scale=TRUE, labels=JanTemp$State,
      plot.faces=TRUE, nrow.plot =8, ncol.plot =8)
```

Three-dimensional scatter plot

```
> library(lattice)
> cloud(Temp Long+Lat,xlab="Longitude",ylab="Latitude",zlab="Temperature")
```