Sunday, December 20, 2015 7:34 PM

Principles of Analytics Graphics

1. Show comparisons for data. Box plot can help
2. Show causality, mechanism, explanation and systematic structure
3. Show multivariate data
4. Integrate different modes of evidence
5. Describe and document the evidence with source and labels with units.
6. Content is the king

Why graphs

- To understand, find patterns, modelling strategies, debug, and to communicate results.
- Exploratory graphs are made for personal understanding and needs to be quick.

Simple summaries of data

- Five number summary
    - Summary()
- Box plots
    - Boxplot()
    - Abline()
- Histograms
    - Hist()- breaks
    - Rug()
- Density plots
- Bar plots
    - Barplot()

Multi dimension summary

- Multiple plots
    - With(pollution, plot())
    - Color can differentiate

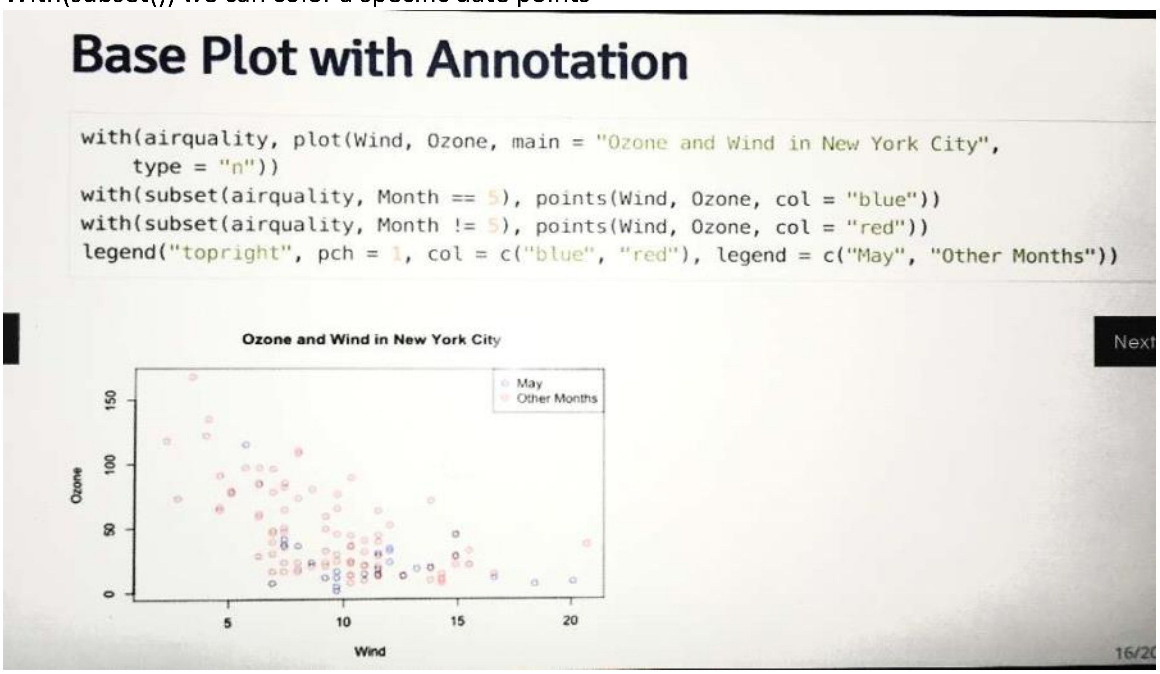
Plotting systems in R

1. The base Plotting system
    1. Plot one by one like artist
    2. Plot() and annotate
    3. It's not possible to go back once created
2. Lattice system
    1. Entire function specified by one function.
    2. Xyplot, bwplot
    3. Good for many plots
3. Ggplot2
    1. Mixes elements of base and Lattice

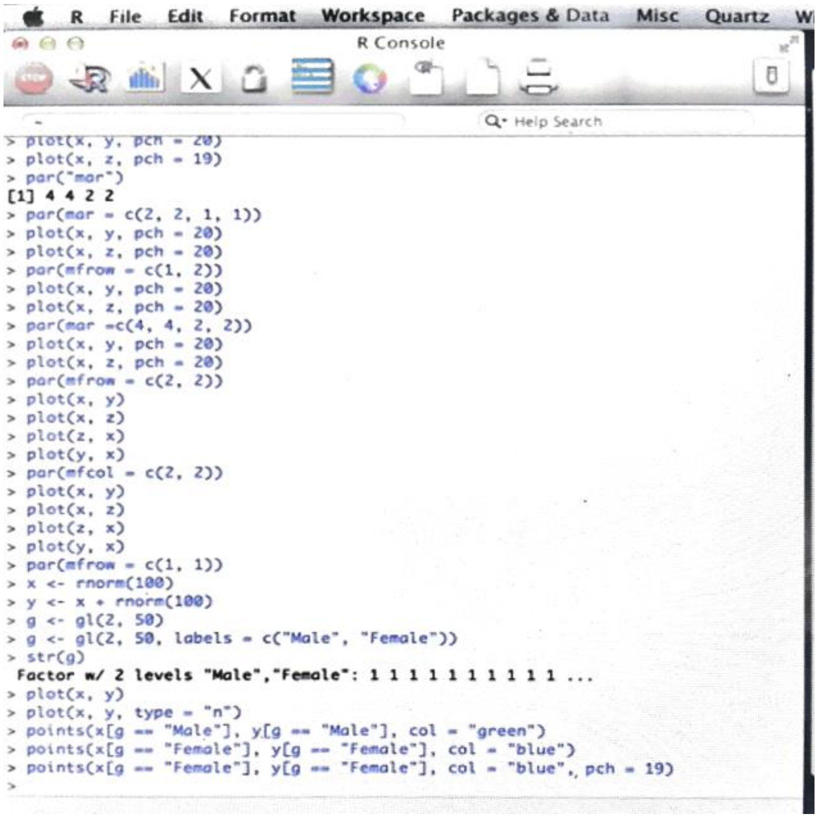
Base Plotting system

Graphics : contains plot, Hist,Boxplot
Grdevices : x11, pdf, postscript,page
Before plotting think the output destination and style
Two steps : initialize the plot then annotate
Plot(x,y) and hist(x)
Library(datasets)
Hist(airquality$0zone)
With(air quality, plot(wind,zone)
Boxplot()
Important parameters

Pch : plotting symbol 1 to 20
Lty : line type
Lwd : line width
Col : color in hex
Xlab : x axis label
Ylab : y axis label
Par() can specify global parameters that affect all plots in R session
Las : label orientation
Bg: the background color
Mar: margin size
Oma: the outer margin
Mfrow, mfcol : number of columns and rows
Base Plotting functions
Plot, line, points, text, title, mtext(margin text), axis
With(subset()) we can color a specific date points

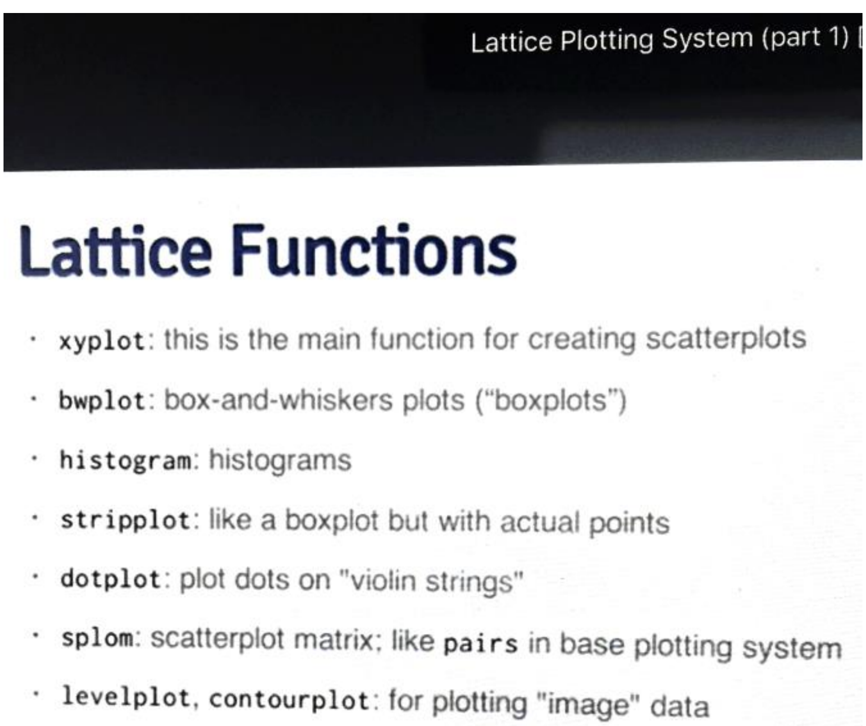


Regression line
Linear model
Model <- lm(ozone~wind, airquality)
Abline(model, lwd=2)
Multiple plots
Par(Mfrow =c(1,2)) can create multiple plots in single device
Example(points) is the demo in R.
Graphics



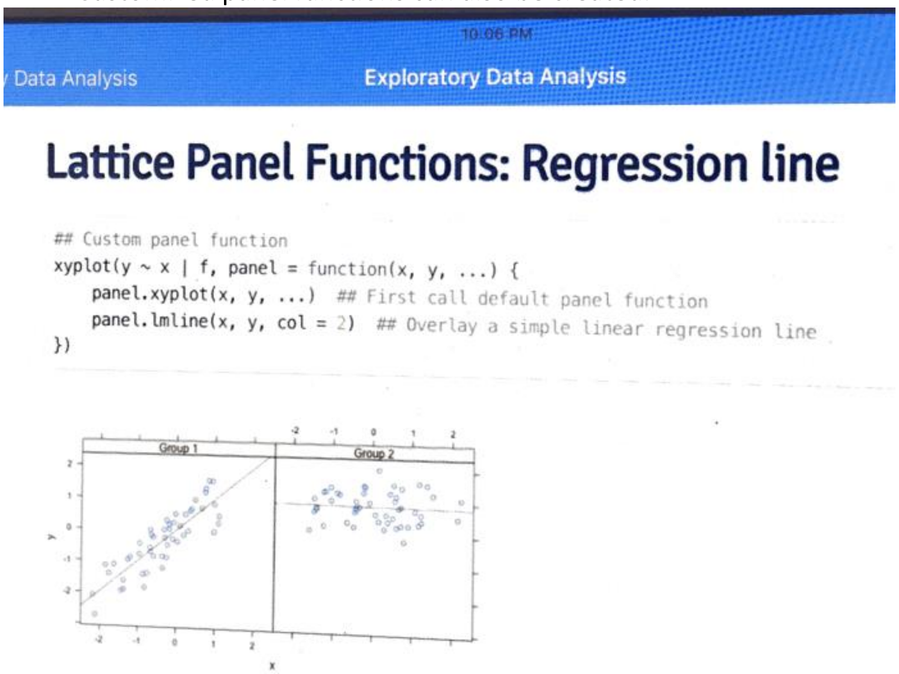
```
> plot(x, y, pch = 20)
> plot(x, z, pch = 19)
> par("mar")
[1] 4 4 2 2
> par(mar = c(2, 2, 1, 1))
> plot(x, y, pch = 20)
> plot(x, z, pch = 20)
> par(mfrow = c(1, 2))
> plot(x, y, pch = 20)
> plot(x, z, pch = 20)
> par(mar =c(4, 4, 2, 2))
> plot(x, y, pch = 20)
> plot(x, z, pch = 20)
> par(mfrow = c(2, 2))
> plot(x, y)
> plot(x, z)
> plot(z, x)
> plot(y, x)
> par(mfcol = c(2, 2))
> plot(x, y)
> plot(x, z)
> plot(z, x)
> plot(y, x)
> par(mfrow = c(1, 1))
> x <- rnorm(100)
> y <- x + rnorm(100)
> g <- gl(2, 50)
> g <- gl(2, 50, labels = c("Male", "Female"))
> str(g)
 Factor w/ 2 levels "Male","Female": 1 1 1 1 1 1 1 1 1 1 ...
> plot(x, y)
> plot(x, y, type = "n")
> points(x[g == "Male"], y[g == "Male"], col = "green")
> points(x[g == "Female"], y[g == "Female"], col = "blue")
> points(x[g == "Female"], y[g == "Female"], col = "blue", pch = 19)
>
```

Dev.off will close pdf
File formats
1. Vector formats : PDFs, svg, win.metafile, postscript
2. Bitmap formats: png,jpeg,bmp,tiff
Dev.set and Dev.cur are used to set and see devices
Dev.copy and Dev.copy2pdf can be used to copy plots
Lattice plotting



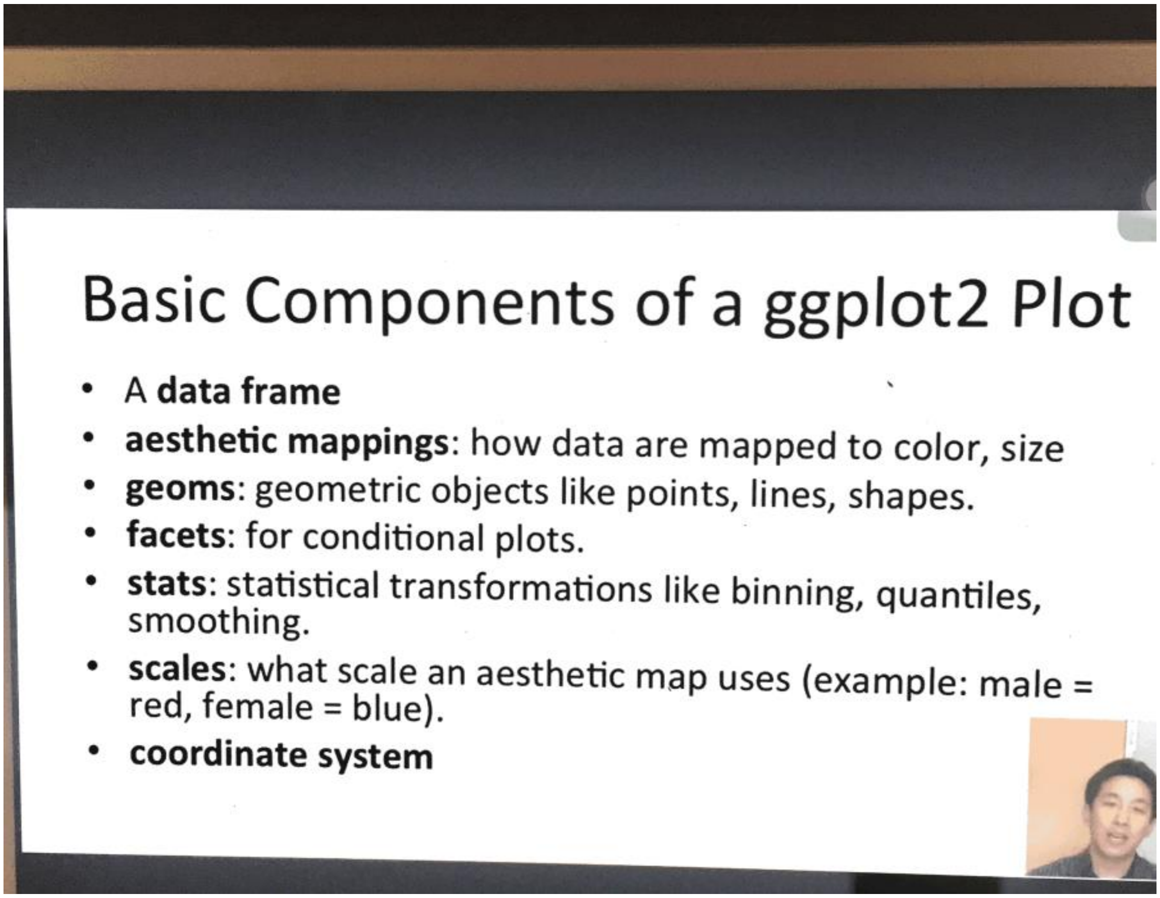
Lattice can have transform function for factor a variable.
Xyplot (y~x, data=frame)

- Base graphics plot data on a device while Lattice returns a class trellis. So it can be stored. Trellis object can be printed.
- Panel functions: It control what happens inside each panel of the plot.
- Customized panel functions can also be created.

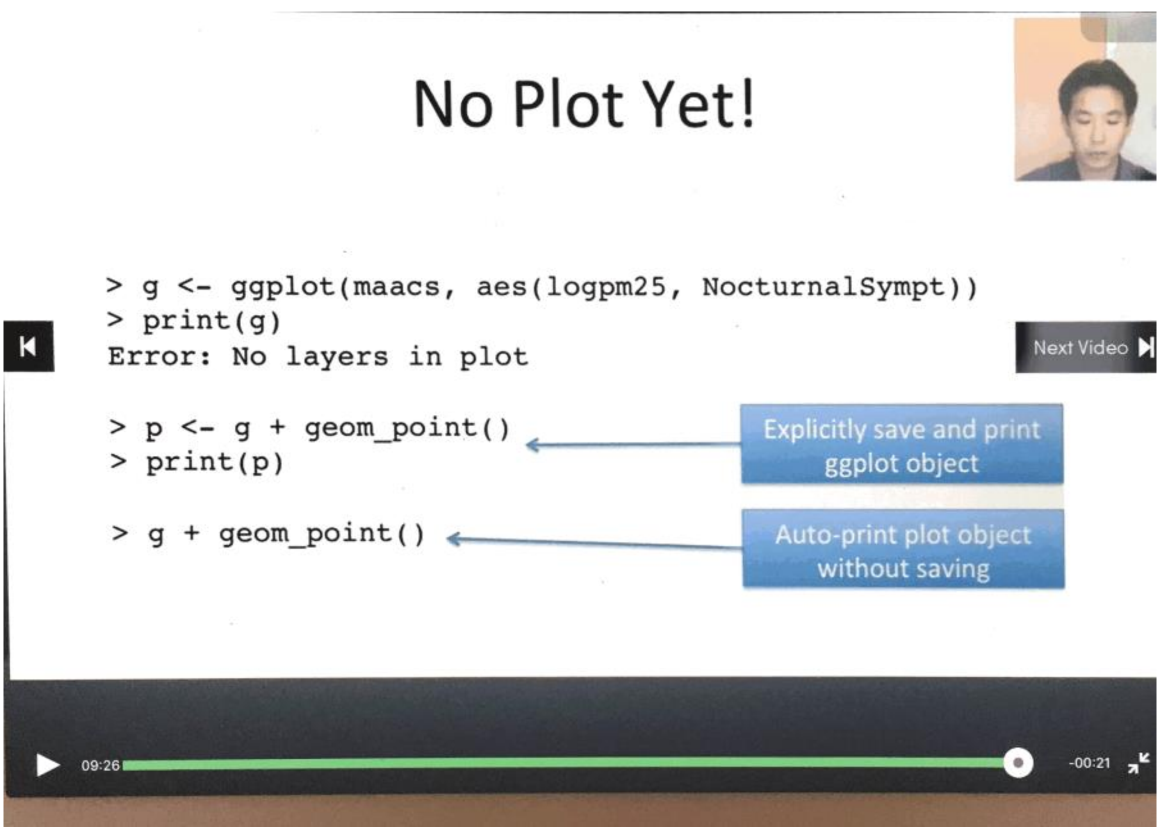


- Any of the base functions annotations cannot be used here.
- Aspects like margins and spacing are automatically handled

Ggplot2
Grammar of Graphics: mapping data to aesthetic attributes of geometric objects.
Qplot() : quick plot
Looks for a data frame
Plots are made of aesthetics(size, shape, color) and geoms (points, lines)
<u>Gplot(x,y,data=DF, color=drv,geom=c("point","smooth"),method='lm')</u>
Other geom is density
Histogram:
Fil for histogram.
Facets for grouping with respect to a specific data. Facets=.~drv; separate by column
Factors are important for indicating subset of the data. They should be labeled.
Ggplot() is the core function.



Basic Components of a ggplot2 Plot
• A data frame
• aesthetic mappings: how data are mapped to color, size
• geoms: geometric objects like points, lines, shapes.
• facets: for conditional plots.
• stats: statistical transformations like binning, quantiles, smoothing.
• scales: what scale an aesthetic map uses (example: male = red, female = blue).
• coordinate system



No Plot Yet!
> g <- ggplot(maacs, aes(logpm25, NocturnalSympt))
> print(g)
Error: No layers in plot
> p <- g + geom_point()
> print(p)
Explicitly save and print ggplot object
> g + geom_point()
Auto-print plot object without saving
Next Video
09:26
-00:21

Annotation

- Labels: xlab(), ylab(), labs(), ggtitle()
- Each of the "geom" functions has options to modify
- For things that only make sense globally, use theme()
  - Example: theme(legend.position = "none")
- Two standard appearance themes are included
  - theme_gray(): The default theme (gray background)
  - theme_bw(): More stark/plain

ggplot(z_Sales,aes(z_Sales$i_ratkWtokWh,z_Sales$g_Mpge))+geom_point(stat = "identity") + facet_grid(.~j_EV)+stat_smooth(method = "lm", col = "red")

Hierarchical Clustering

- Clustering organizes things that are close into groups
- Cluster Analysis is really good in science
- Distance or Simliarity -> pick which you want to use
- Euclidian Distance and Manhattan distance

- Dist function - dist(list1,list2) Gives the distance matrix
-
```
dataFrame <- data.frame(x = x, y = y)
distxy <- dist(dataFrame)
hClustering <- hclust(distxy)
plot(hClustering)
```
- Pretty Dentograms - Course notes
- Colored Dentograms - Google
- Heatmaps() is good too

K-Means clustering

- Fixed number of cluster, centroid, assign things to closest centroid and recalculate

Project

Sunday, December 27, 2015 6:00 PM

```
NEI_DATA <- readRDS("summarySCC_PM25.rds")
SCC_DATA <- readRDS("Source_Classification_Code.rds")
```

Q1

```
Total_Emission <- aggregate(Emissions ~ year,NEI_DATA, sum)
barplot(Total_Emission$Emissions,   names.arg=Total_Emission$year,   xlab="Year",   ylab="PM2.5
Emissions",   main="Total PM2.5 Emissions" )
```

Q2

```
NEI_Baltimore <- NEI_DATA[NEI_DATA$fips=="24510",]
Emission_Baltimore<-aggregate(Emissions ~ year, NEI_Baltimore,sum)
barplot(Emission_Baltimore$Emissions,names.arg=Emission_Baltimore$year, xlab="Year",ylab="PM2.5
Emissions",main="Total PM2.5 Emissions From Baltimore")
```

Q3

```
Plot_by_Source <- ggplot(NEI_Baltimore,aes(factor(year),Emissions,fill=type)) +
  geom_bar(stat="identity") +
  facet_grid(.~type)+
  labs(x="year", y="Total Emission", title="Emissions in Baltimore City by Source Type from 1999 to
2008")
print(Plot_by_Source)
```

Q4

```
Coal_Cumbustion <- grepl("Fuel Comb.*Coal", SCC_DATA$EI.Sector)
Coal_Cumbustion_Data <- SCC[Coal_Cumbustion,]
Coal_Cumbustion_emissions <- NEI_DATA[(NE_DATAI$SCC %in% Coal_Cumbustion_Data$SCC), ]
Emission_to_Plot <- aggregate(Emissions ~ year, data=Coal_Cumbustion_emissions, FUN=sum)
library(ggplot2)
ggp<-ggplot(Emission_to_Plot, aes(x=factor(year), y=Emissions))+geom_bar(stat="identity")+
labs(x="year", y="total PM2.5 emissions")+ggtitle(" Combined coal combustion-related emissions")
print(ggp)
```

Q5

```
Baltimore_MV <- NEI_Baltimore[NEI_Baltimore$type=="ON-ROAD",]
MV_to_Plot <- aggregate(Emissions ~ year, data=Baltimore_MV, FUN=sum)
print(ggplot(MV_to_Plot, aes(x=factor(year), y=Emissions)) +
     geom_bar(stat="identity") +
     labs(x="year", y="total PM2.5 emissions") +
     ggtitle("Emissions in Baltimore from motor vehicle sources"))
```

Q6

```
Baltimore_MV$city <-"Baltimore"
NEI_LosAngeles <- NEI_DATA[NEI_DATA$fips =="06037"&NEI_DATA$type=="ON-ROAD",]
NEI_LosAngeles$city <- "Los Angeles"
Combined_NEI_Baltimore_LosAngeles <- rbind(Baltimore_MV,NEI_LosAngeles)
```

```
print(ggplot(Combined_NEI_Baltimore_LosAngeles, aes(x=factor(year), y=Emissions, fill=city)) +
    geom_bar(stat="identity") +
    facet_grid(.~city) +
    labs(x="year", y="Total PM2.5 Emission") +
    labs(title="PM2.5 Motor Vehicle Emissions in Baltimore & LA from 1999-2008"))
```