Principles of Analytics Graphics

Thursday, August 4, 2016 3:38 PM

- Show comparisons
  - Evidence for hypothesis is always relative to another
  - At times merely a control comparison
- Show causality, mechanism, explanation, systematic structure
  - Think about what causes explain hypothesis
  - Make plots to narrow onto those causes and how they relate to hypothesis
- Show multi variate data
  - Real world is inherently multi variate
  - Simpson's Paradox: Relationship between 2 variables may be confounding the relationship of one of the two and other unrelated variable
- Integration of evidence
  - Integrate all data to make data clear and trends clear
  - Don't let tool drive analysis
- Describe and document evidence with labels, scales, and sources
- Content is king

Exploratory Graphs

Thursday, August 4, 2016 3:49 PM

- To understand data properties
- To find patterns in data
- To suggest modeling strategies
- To "debug" analyses
- To communicate results

Characteristics:

- Made quickly
- Large number of them
- Personal understanding
- Axes/legends are generally cleaned up
- Color/size primarily used for information

One Dimensional Data:

Simple summaries of data:

- Five number summary
  - Given field (column) of data
  - Get minimum, 1st quartile, median, mean, 3rd quartile, max
  - summary()
- Box plot
  - boxplot(data, col ="color")
- Histogram
  - hist(data, col="color", breaks="number of bars")
  - rug(data) -> gives outliers under hist plot
- Overlaying features
  - abline(h="height") -> location of line
- Barplot (categorical data)
  - barplot(table, graph info)

Two Dimensional Data:

- Multiple/overplayed 1D plots
  - Multiple Boxplots
- Scatter and smooth scatter plots
  - with(table, plot data)
  - Can add color to add a new variable
  - To add more variables, can also mix scatter plots

More than 2 Dimensions

- Overlayed/multiple 2-D plots
- Use color, size, shape to add dimensions
- Spinning plots
- Actual 3D plots

Plotting Systems in R

Thursday, August 4, 2016 4:17 PM

- "Artist's palette model" (Base Model)- graphics package
  - Build up blank canvas
  - Start with plot function
  - Use annotation functions to add/modify
  - (
    - Text,
    - Lines,
    - Points,
    - Axis)
  - Convenient, mirrors how we think of building plots and analyzing data
  - Can't go back once plot has started
  - Difficult to translate to others
  - Must control each part of it
- Lattice System- lattice package
  - Single function constructs entire plot at once
  - Must specify in call to function
  - Best for "conditioning" types of plots- how y changes with x across levels of z
  - Cannot add to plot after created, must know everything at first call
- Ggplot2
  - Splits difference of base and lattice
  - Auto spacing, text, etc.
  - Has function to add to a plot and work with
  - More intuitive version of lattice system

Base Plotting System

Thursday, August 4, 2016 4:47 PM

- Where is plot located?
- How will plot be used?

Process:

- What graphics system to use? (Base, lattice, ggplot2)
- Base:
    - 2D Graphics
    - 2 Phases
        - Initializing
        - Annotating
    - Call plot (x, y) or hist(x)
    - Key base parameters:
        - Pch: plotting symbol (default is circle)
        - Lty: Line type (dashed, dotted, etc)
        - Lwd: line width
        - Col: color
        - Xlab: x-axis label
        - Ylab: y-axis label
    - Global base graph parameters:
        - Las: orientation of axis labels on plot
        - Bg: background color
        - Mar: margin size
        - For more look at documentation
- Base Plotting Functions:
    - plot: make scatterplot
    - lines: add lines to a plot
    - points: add points to a plot
    - text: add text labels to a plot
    - title: add annotations to x,y axis labels, title, subtitle, outer margin
    - mtext: add arbitrary text to margins of plot
    - axis: adding axis ticks and labels

Graphics Devices in R

Thursday, August 4, 2016 9:05 PM

- Graphics Device is something that can make a plot appear (ie sent to:)
  - Screen
  - PDF file
  - PNG or JPEG
  - SVG
- Most common place is screen via windows() (or other)
- How does a plot get created?
  - Method 1
    - Plotting function, like plot, xyplot, or qplot
    - Plot appears on screen device
    - Annotate plot if necessary
    - Enjoy!
  - Method 2
    - Explicitly launch graphics device
    - Call plotting function to make plot
    - Annotate if necessary
    - Explicitly close graphics device with dev.off()
    - Like opening sockets and connections to dbs in Java connectors
- Vector formats
  - pdf: very portable
  - svg: xml based scalable vector graphics, popular web based plots
  - win.metafile: only on Windows
  - postscript: older format
- Bitmap formats:
  - png
  - jpeg
  - tiff
  - bmp
- Multiple open graphic devices
  - Plotting can only occur on one at a time
  - dev.cur() = current device
  - dev.set(int) to choose device
  - dev.copy(format, filename) = copy plots to other graphics devices
  - dev.copy2pdf() = copy plot to pdf

Lattice Plotting

Sunday, August 7, 2016 10:50 AM

- lattice package
- grid = different graphing system, lattice built on it
- Don't separate plotting + annotation
- xyplot- main one
    - xyplot (y ~ x| f * g, data)
    - f,g = conditioning variables
    - data = data frames
- Lattice graphics return an object of class trellis
    - base plots just print to screen
    - Lattice must be printed
    - Usually auto-printed for you, but can be worse
- Panel functions
    - lattice- default panel functions, can supply your own
    - panel receives x/y coordinates and other arguments (how to format the panel)
        - Literally 2 panels of graphs

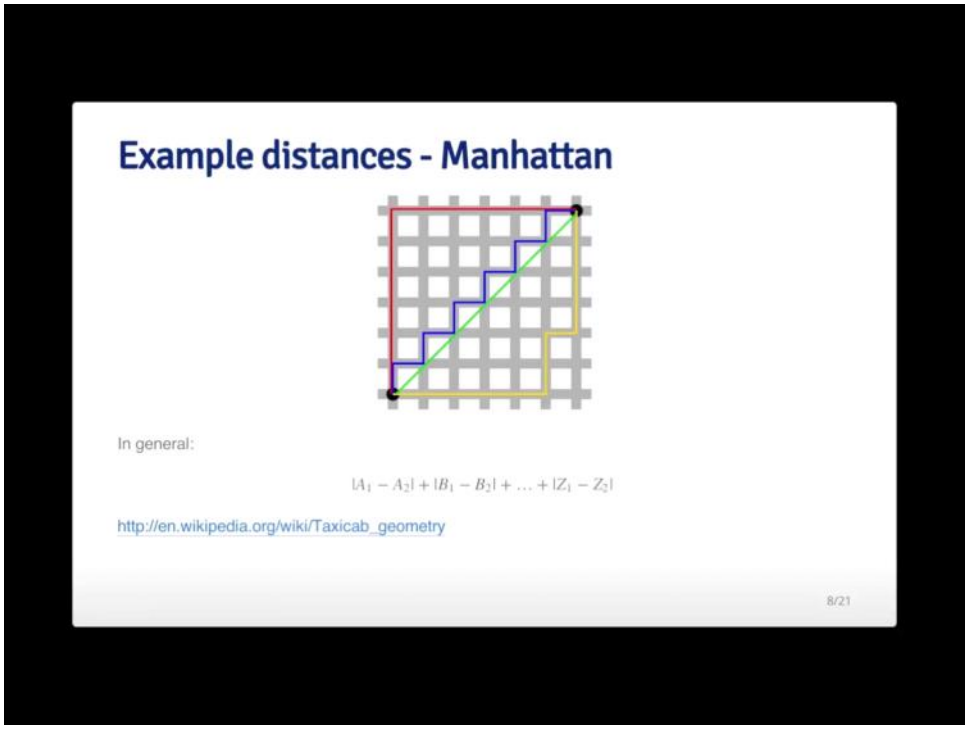
ggplot2

Sunday, August 7, 2016 11:07 AM

- ggplot2
  - Grammar of Graphics- graphics = abstract concepts, like verbs, nouns, and adjectives
  - qplot()- plot() in base
    - Looks for data in data frame
    - plots made of
      - aesthetics(size, shape, and color)
      - geoms(line, points, etc.)
    - Factors must be properly labelled
    - qplot(x coord, y coord, data = dataframe)
    - Adding a geom
      - qplot(x coord, y coord, data = dataframe, geom = c(geoms))
      - histograms: qplot(x coord, data = dataframe, fill = y coord)- fill not necessary, but can allow for separation via factors/specifying y coord
    - Facets (like panels)
      - qplot(facets = .~separable factor var)- column panels
      - qplot(facets = separable factor var.~)- row panels
    - qplot = quick plot (maybe?), but essentially doesn't use all of ggplot2. Can use ggplot() function to harness full potential
- Components of ggplot2
  - data frame
  - aesthetic mappings- color, size, etc
  - geoms- points, lines, shapes
  - facets- conditional plots (like panels)
  - stats- statistical transformations
  - scales- scales and legends
  - coordinate system
- Plot in layer
  - Plot data
  - Overlay summary- smooth/regression line
  - Annotate- legends, categorizing, etc.
- Call summary on a ggplot object will get all metadata
- Can make a ggplot with defaults of aesthetics but no actual data; will not let you plot but will have defaults
- Adding layers
  - Smooth- geom_smooth()
    - Will make a line smoother and fit a workable trend more easily
    - Has potential attributes
  - Facets- facet_grid() function (same syntax as in qplot)
  - Annotations
    - xlab(), ylab(), labs(), ggtitle()
    - Global stuff changed via theme()
  - Axis Limits (gets rid of outliers)
    - ylim(limits for outlier)

Hierarchical Clustering

Sunday, August 7, 2016 11:34 AM

- Clustering organizes things that are close into groups
  - How do we define close and apart?
  - How do we group?
  - How do we visualize grouping?
  - How do we interpret grouping?
- "Agglomerative" approach
  - Find closest 2 things
  - Put together
  - Find next closest, put together
- Requires
  - Defined distance
  - Merging approach
- Produces
  - Tree of closeness of things to each other
- Distances or Similarity
  - Continuous
    - Euclidean or correlation measure
  - Binary (Manhattan)
  - Potentially another?
- Euclidean- straightforward
- Manhattan
  - 

  - Green = Euclidean
  - Manhattan = blue, yellow, or red
- **Function: hclust()**
  - Makes a cluster object
  - Can be plotted as a dendogram (kinda a tree diagram)

    - myplclust() use for coloring and prettier dendograms
- How to merge points together?
    - Average linkage- new coordinate location = average of all coordinates in cluster
    - Complete linkage- take distance from maximally farthest point in each cluster
- **heatmap()**
    - Rows of table and columns of table to visualize clusters across high dimensional data
- Need to cut somewhere along clustering algorithm to make sure doesn't become one huge cluster; unique decision

K-Means Clustering

Sunday, August 7, 2016 11:53 AM

- Partitioning approach
  - Fix a number of clusters
  - Get "centroids" of clusters
  - Assign things to each centroid
  - Recalculate centroid till even
- Requires
  - Defined distance metric
  - Number of clusters
  - Initial guess to centers
- Produces
  - Final estimate of cluster centroids
  - Assignment of each point to clusters
- **kmeans()**
  - kmeans(data frame, centers = #)
  - Produces kmeansObj
- Heatmaps
  - Differentiate clusters in a table

Dimension Reduction

Sunday, August 7, 2016 12:44 PM

- Matrix data
- Cluster data
- Adding a pattern
    - Some numerical pattern in data
- Tens of thousands of variables, not all independent of each other, so varied correlation across data
- Need to find fewer variables to explain original data
- **First: Statistical Analysis**
- **Second: Data Compression**
- SVD
    - Separate into 3 matrices: $X = UDV^T$
    - U is orthogonal as is V
- PCA
    - Simply scale down matrix
- Can't run dimensional reduction techniques if missing values
    - Many solutions
    - One of which is **imputing,** use nearby rows to make and estimate