

# Web Science:

# Intro to InfoVis with R and Python

## (Part 1 - InfoVis Principles)

CS 432/532  
Old Dominion University

Many slides courtesy Tamara Munzner, [VAD minicourse](#), June 2014

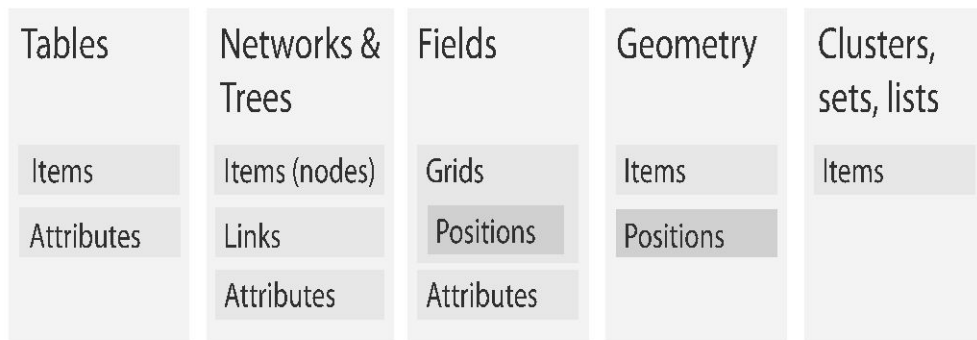
Based on Tamara Munzner, [Visualization Analysis and Design](#), AK Peters / CRC Press, Oct 2014



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](#)

# Dataset and data types

## ➔ Data and Dataset Types



## ➔ Data Types

➔ Items   ➔ Attributes   ➔ Links   ➔ Positions   ➔ Grids

## ➔ Dataset Availability

➔ Static



➔ Dynamic



# Attribute types

## ➔ Attribute Types

➔ Categorical



➔ Ordered

➔ *Ordinal*



➔ *Quantitative*



## Encode

## ② Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



## ② Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



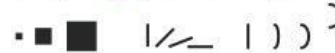
→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



→ Motion

*Direction, Rate, Frequency, ...*

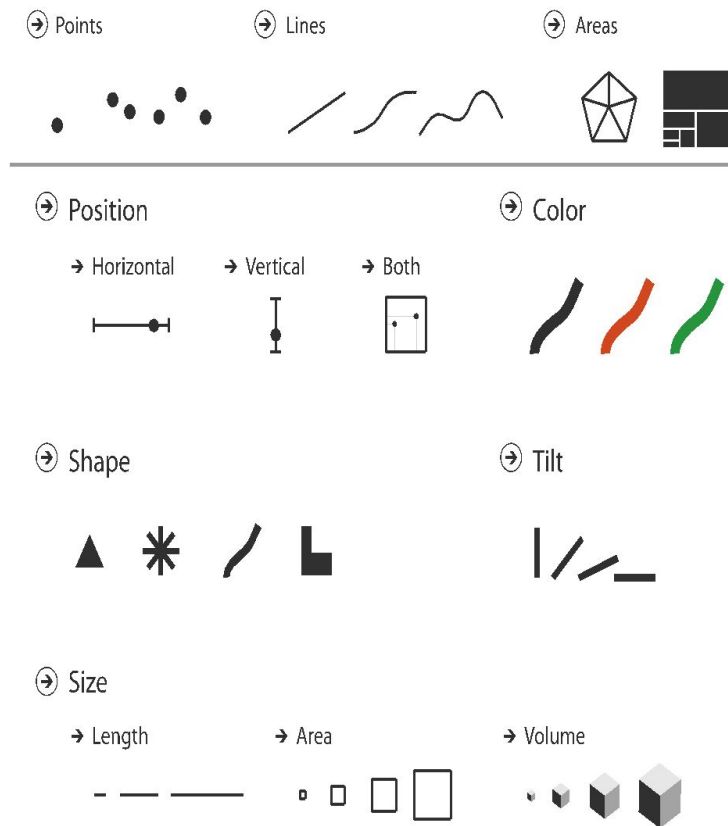
What?

Why?

How?

# Definitions: Marks and channels

- marks
  - geometric primitives
- channels
  - control appearance of marks
  - can redundantly code with multiple channels
- interactions
  - point marks only convey position; no area constraints
    - can be size and shape coded
  - line marks convey position and length
    - can only be size coded in 1D (width)
  - area marks fully constrained
    - cannot be size or shape coded



# Color: Luminance, saturation, hue

- 3 channels

- identity channel for categorical

- hue

- magnitude channels for ordered

- luminance

- saturation

Hue



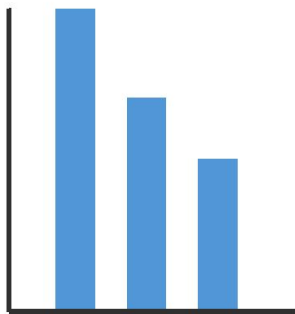
Luminance



Saturation

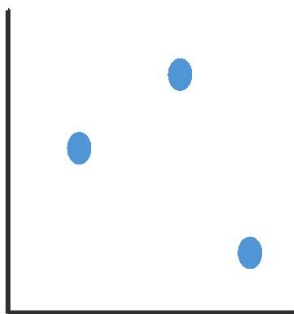


# Visual encoding



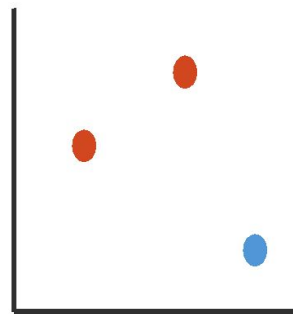
1:  
vertical position

mark: line



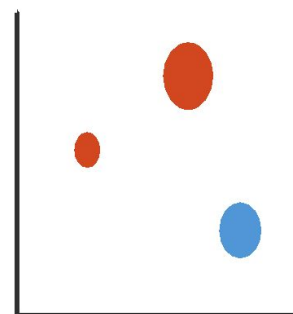
2:  
vertical position  
horizontal position

mark: point



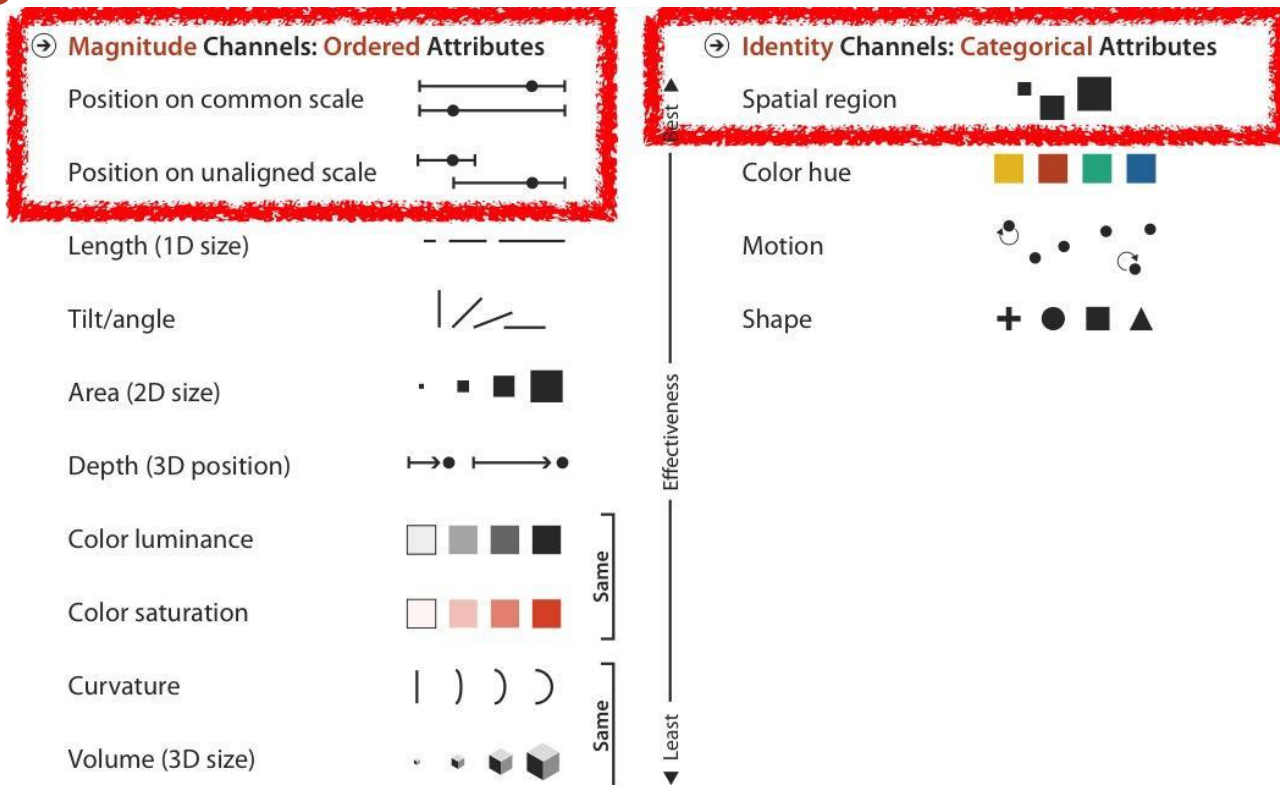
3:  
vertical position  
horizontal position  
color hue

mark: point



4:  
vertical position  
horizontal position  
color hue  
size (area)  
mark: point

# Channels: Expressiveness types and effectiveness rankings



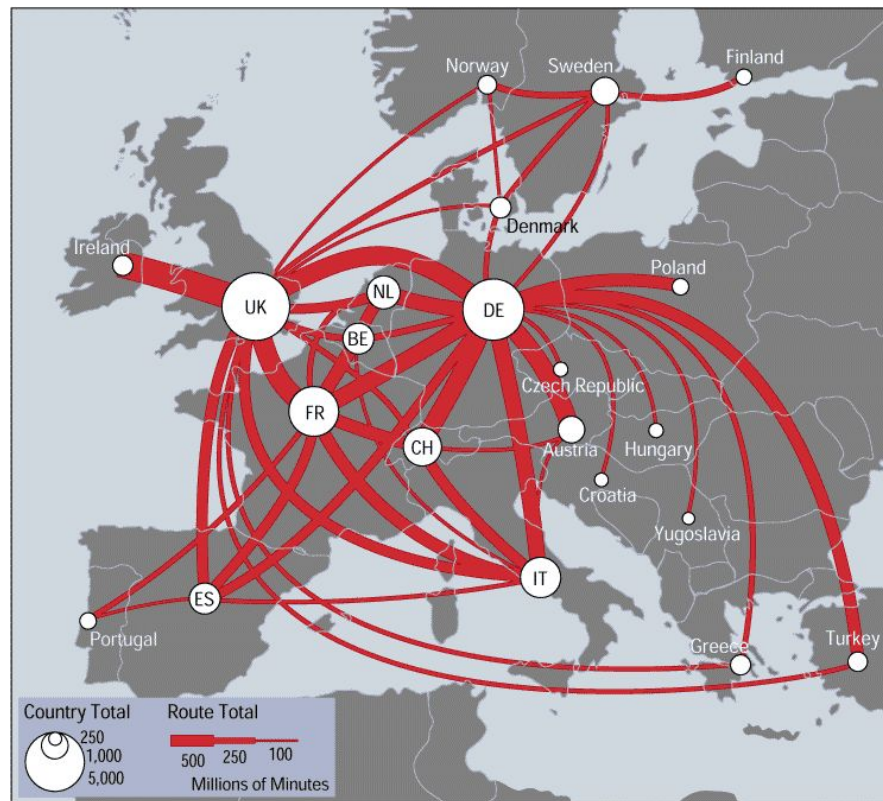


## Effectiveness and expressiveness principles

- effectiveness principle
  - encode most important attributes with highest ranked channels
- expressiveness principle
  - match channel and data characteristics

# Discriminability: How many usable steps?

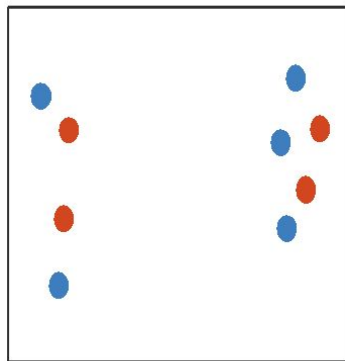
- linewidth: only a few



source: [Telecommunications Traffic Flow Map](#)

# Separability vs. Integrality

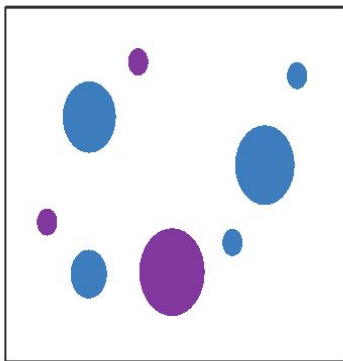
Position  
+ Hue (Color)



Fully separable

2 groups each

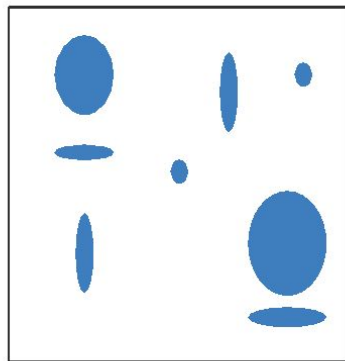
Size  
+ Hue (Color)



Some interference

2 groups each

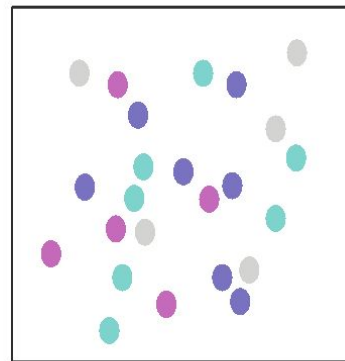
Width  
+ Height



Some/significant  
interference

3 groups total:  
integral area

Red  
+ Green

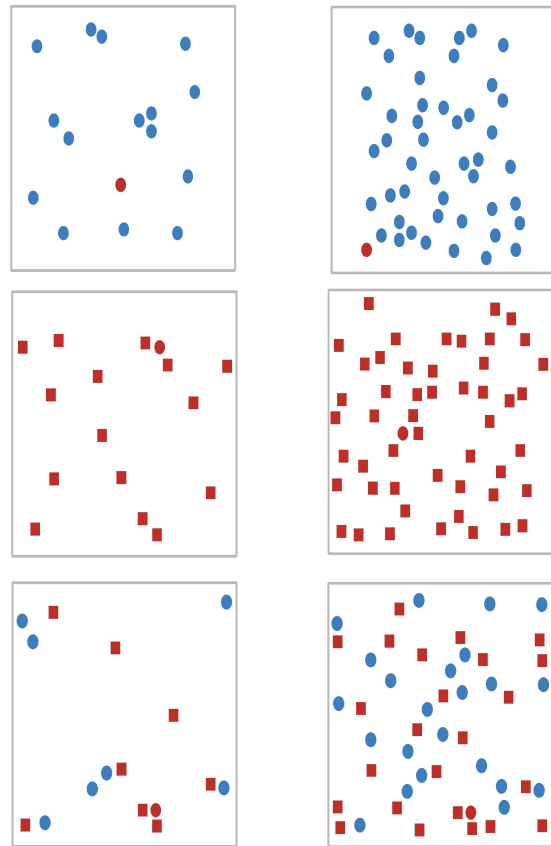


Major interference

4 groups total:  
integral hue

# Popout

- find the red dot
  - how long does it take?



➔ **Arrange**

→ Express      → Separate



→ Express      → Separate



→ Order      → Align



→ Order      → Align



→ Use



➔ Map

from categorical and ordered attributes

→ Color

→ Hue      → Saturation      → Luminance



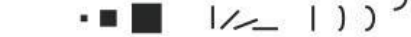
→ Hue      → Saturation      → Luminance



→ Hue      → Saturation      → Luminance



- Size, Angle, Curvature, ...

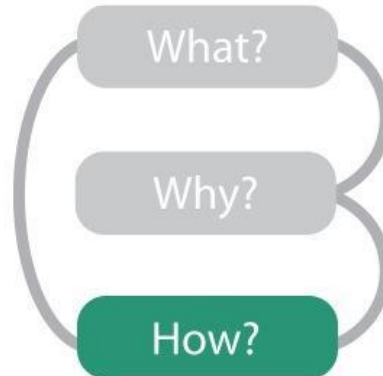


→ Shape



→ Motion

*Direction, Rate, Frequency, ...*



# Arrange tables

② Express Values



③ Separate, Order, Align Regions

→ Separate



→ Order



→ Align



# Web Science:

# Intro to InfoVis with R and Python

## (Part 2 - Visualization Idioms)

CS 432/532  
Old Dominion University

Many slides courtesy Tamara Munzner, [VAD minicourse](#), June 2014

Based on Tamara Munzner, [Visualization Analysis and Design](#), AK Peters / CRC Press, Oct 2014



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](#)

# Idiom: scatterplot

- **express** values

- quantitative attributes

- no keys, only values

- data

- 2 quant attribs

- mark: points

- channels

- horiz + vert position

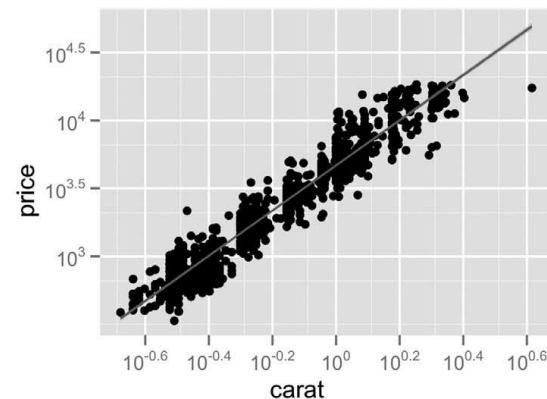
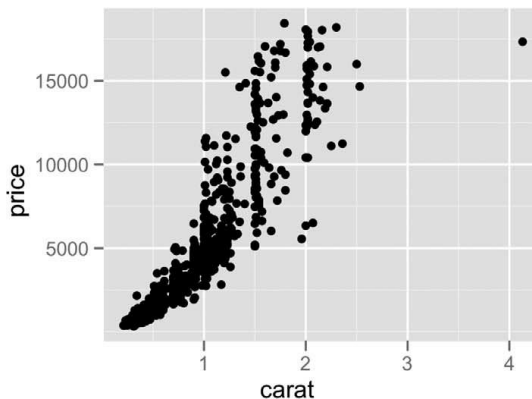
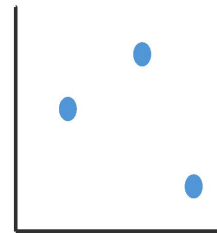
- tasks

- find trends, outliers, distribution, correlation, clusters

- scalability

- hundreds of items

➞ Express Values

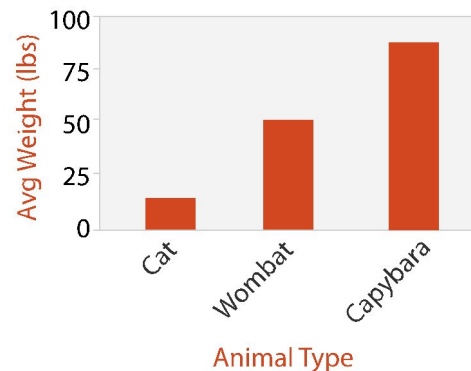
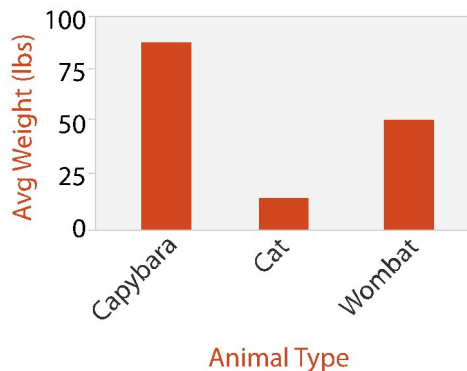


source: [A layered grammar of graphics. Wickham. Journ. Computational and Graphical Statistics 19:1 (2010), 3–28.]



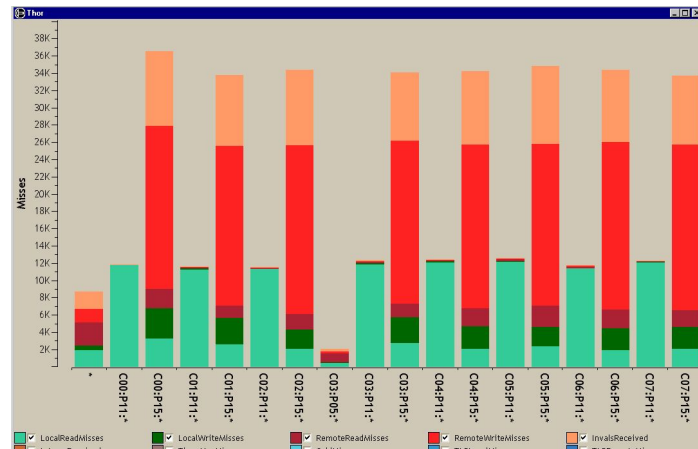
# Idiom: bar chart

- one key, one value
  - data
    - 1 categ attrib, 1 quant attrib
  - mark: lines
  - channels: length to express quant value
    - spatial regions: one per mark
      - **separated** horizontally, **aligned** vertically
      - **ordered** by quant attrib
        - » by label (alphabetical), by length attrib (data-driven)
  - task
    - compare, lookup values
  - scalability
    - dozens to hundreds of levels for key attrib



# Idiom: stacked bar chart

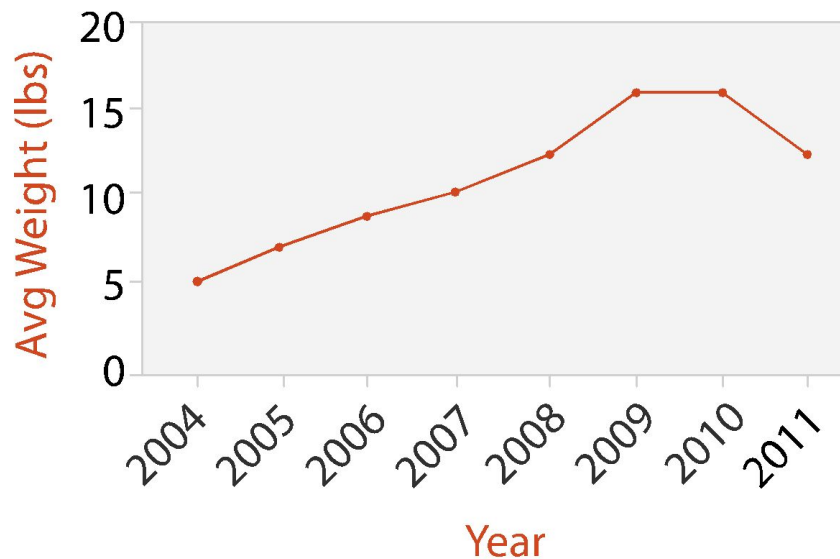
- one more key
  - data
    - 2 categ attrib, 1 quant attrib
  - mark: vertical stack of line marks
    - **glyph**: composite object, internal structure from multiple marks
  - channels
    - length and color hue
    - spatial regions: one per glyph
      - aligned: full glyph, lowest bar component
      - unaligned: other bar components
  - task
    - part-to-whole relationship
  - scalability
    - several to one dozen levels for stacked attrib



source: [Using Visualization to Understand the Behavior of Computer Systems. Bosch. Ph.D. thesis, Stanford Computer Science, 2001.]

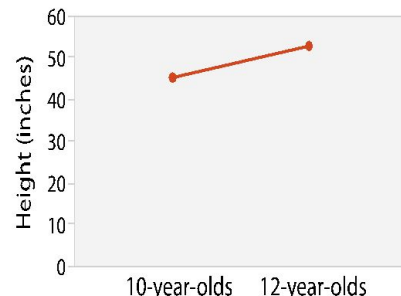
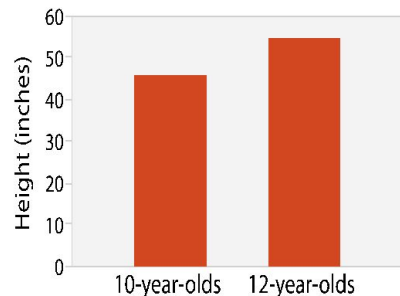
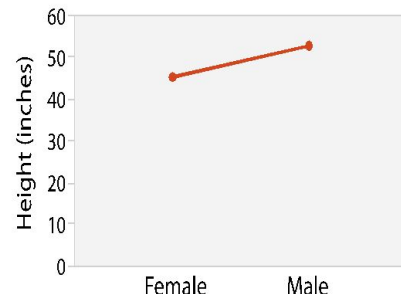
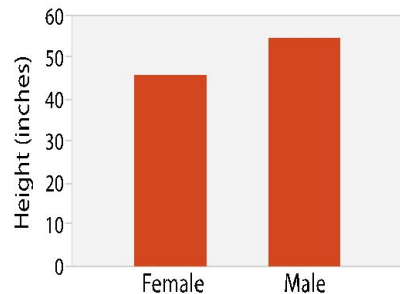
# Idiom: line chart

- one key, one value
  - data
    - 1 quant attrib, 1 ordered attrib
  - mark: points
    - line connection marks between them
  - channels
    - aligned vertical position
    - separated and ordered by key attrib into horizontal regions
  - task
    - find trend
      - connection marks emphasize ordering of items along key axis by explicitly showing relationship between one item and the next



# Choosing bar vs line charts

- depends on type of key attrib
  - bar charts if categorical
  - line charts if ordered
- do not use line charts for categorical key attribs
  - violates expressiveness principle
    - implication of trend so strong that it overrides semantics!
      - “The more male a person is, the taller he/she is”



source: *[Bars and Lines: A Study of Graphic Communication. Zacks and Tversky. Memory and Cognition 27:6 (1999), 1073–1079.]*

# Idiom: scatterplot matrix

- scatterplot matrix (SPLOM)

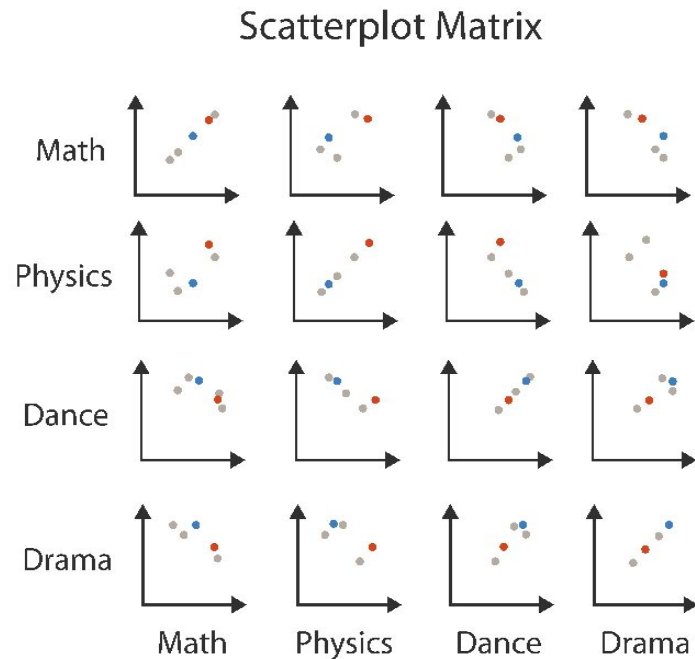
- rectilinear axes, point mark
- all possible pairs of axes
- scalability

- one dozen attribs
- dozens to hundreds of items

- task: correlation

- positive correlation: diagonal low-to-high
- negative correlation: diagonal high-to-low
- uncorrelated

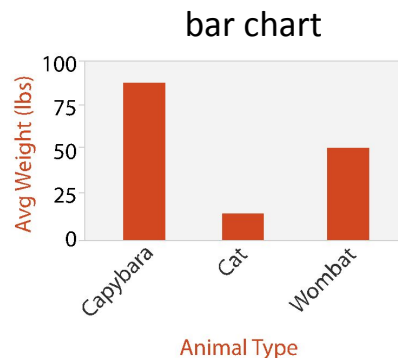
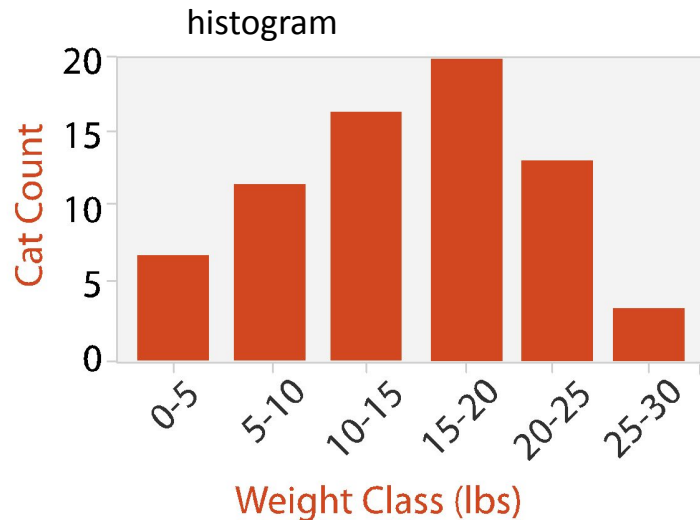
Table			
Math	Physics	Dance	Drama
85	95	70	65
90	80	60	50
65	50	90	90
50	40	95	80
40	60	80	90



source: [\[Visualization Course Figures\]](#). McGuffin, 2014]

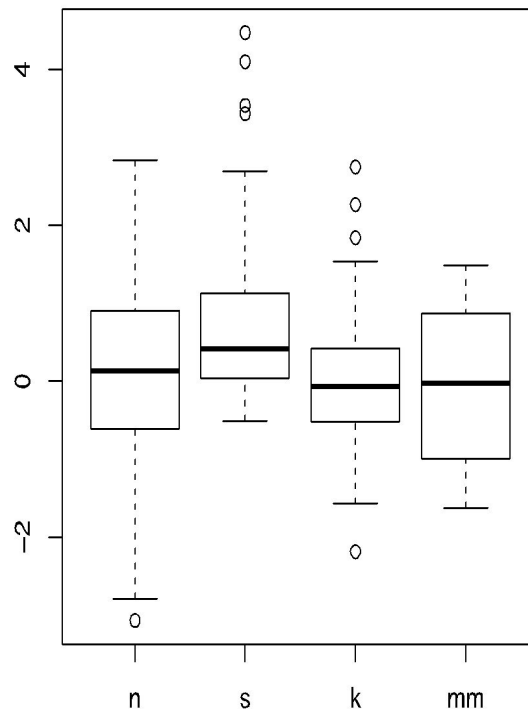
## Idiom: histogram

- static item aggregation
- task: find distribution
- data: table
- derived data
  - new table: keys are bins, values are counts
- bin size crucial
  - pattern can change dramatically depending on discretization (bin size)



# Idiom: **boxplot**

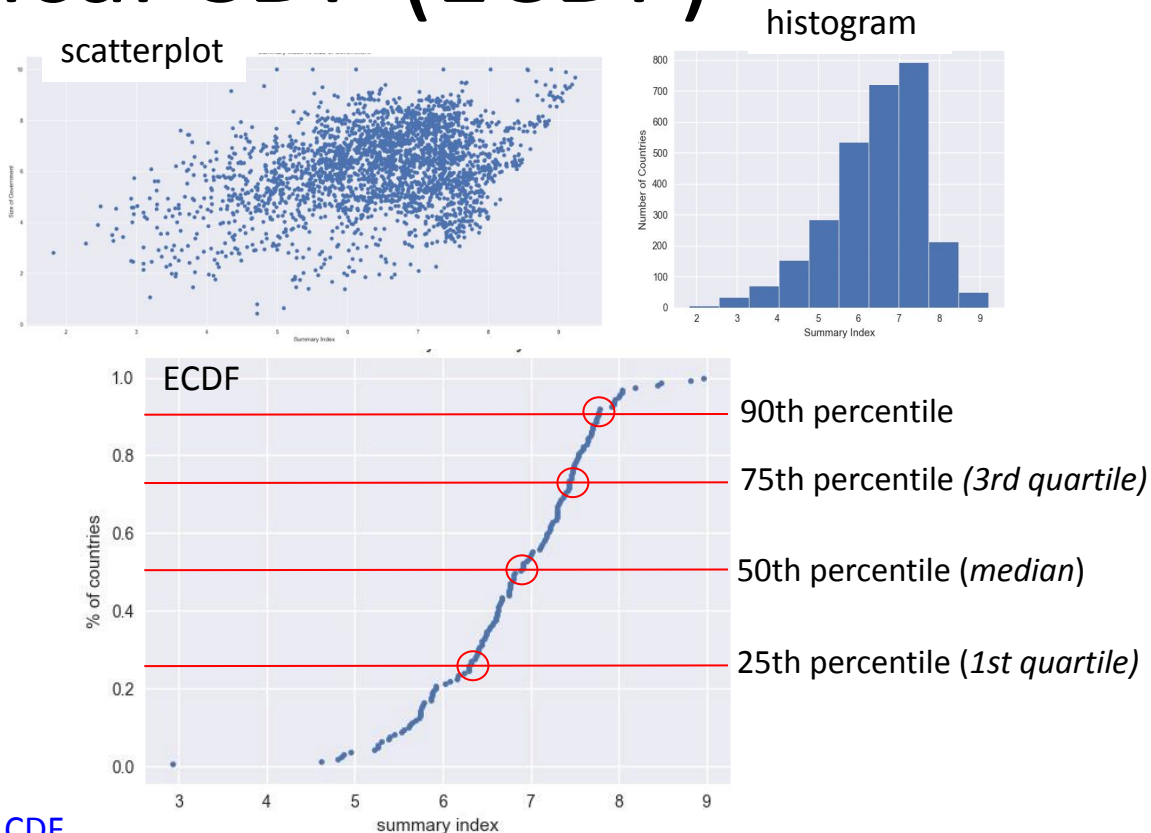
- static item aggregation
- task: find distribution
- data: table
- derived data
  - 5 quant attribs
    - median: central line
    - lower and upper quartile: boxes
    - lower upper fences: whiskers
      - values beyond which items are outliers
  - outliers beyond fence cutoffs explicitly shown



source: [[40 years of boxplots](#). Wickham and Stryjewski. 2012]

# Empirical CDF (ECDF)

- Expresses percentage of values  $\leq x$
- Never decreases
- Easy to spot median, quantiles



source: [What, Why, and How to Read Empirical CDF](#)



# Interested in learning more?

Watch Tamara Munzner's [D3 Unconference Keynote](#), Nov 2015 (55 min)

Still interested?

Take CS 625 - Data Visualization

*covers most of Munzner's textbook*

Still interested?

Take CS 725/825 - Information Visualization and Visual Analytics

*more advanced visualizations, research-based, assumes knowledge of all material from CS 625*

# Web Science:

# Intro to InfoVis with R and Python

(Part 3 - Charts with R)

CS 432/532

Old Dominion University



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

# What is R?

- Software for statistical computing and graphics
- Free (GNU General Public Licence) implementation of the S programming language
- Based on a command line interface (various GUIs are available)
- Extensible through loadable libraries
- Available for multiple platforms including Linux, MacOS, Windows
- Well-documented with its own documentation format, similar to UNIX man pages

available at [The R Project for Statistical Computing](https://www.R-project.org/)  
popular GUI: [RStudio](https://rstudio.com/)

ref: ["An Incomplete R Tutorial"](#), Martin Klein

# R Resources

- [The R Project](#) - main R website
  - [R Documentation](#)
- [RStudio](#) - most popular GUI for R
  - [Connect RStudio to Git and GitHub](#)
- [RStudio and R for Beginners](#)
- [R for Data Science](#) - tutorials from O'Reilly book
  - [R TidyVerse packages](#) - helper packages for Data Science (used in "R for Data Science")
- ["An Incomplete R Tutorial"](#), by Martin Klein

# R and Google Colab

Using R as a calculator:

```
1 / 200 * 30
(59 + 73 + 2) / 3
sin(pi/2)
```

```
0.15
44.6666666666667
1
```

**Important:** Variable assignment is done with `<-`, not with `=`

```
x <- 3*4
x
```

```
12
```

**Note:** There's no menu option to create an R notebook, so you have to create a notebook with certain options in the URL (see link below)

[Create new Google Colab R notebook](https://colab.research.google.com/notebook#create=true&language=r)

(<https://colab.research.google.com/notebook#create=true&language=r>)

# ggplot2

- [How to make any plot in ggplot2? | ggplot2 Tutorial](#)
  - basic getting started guide
- [The Complete ggplot2 Tutorial - Part1 | Introduction To ggplot2 \(Full R code\)](#)
  - simple syntax explanation
  - scatterplot
- [Top 50 ggplot2 Visualizations - The Master List \(With Full R Code\)](#)
  - code for 50 different types of charts
- [Data Visualization Cheat Sheet](#)
- [A ggplot2 Tutorial for Beautiful Plotting in R](#) and [Beautiful plotting in R: A ggplot2 cheatsheet](#)
  - examples on how to change a ton of options for customizing the look

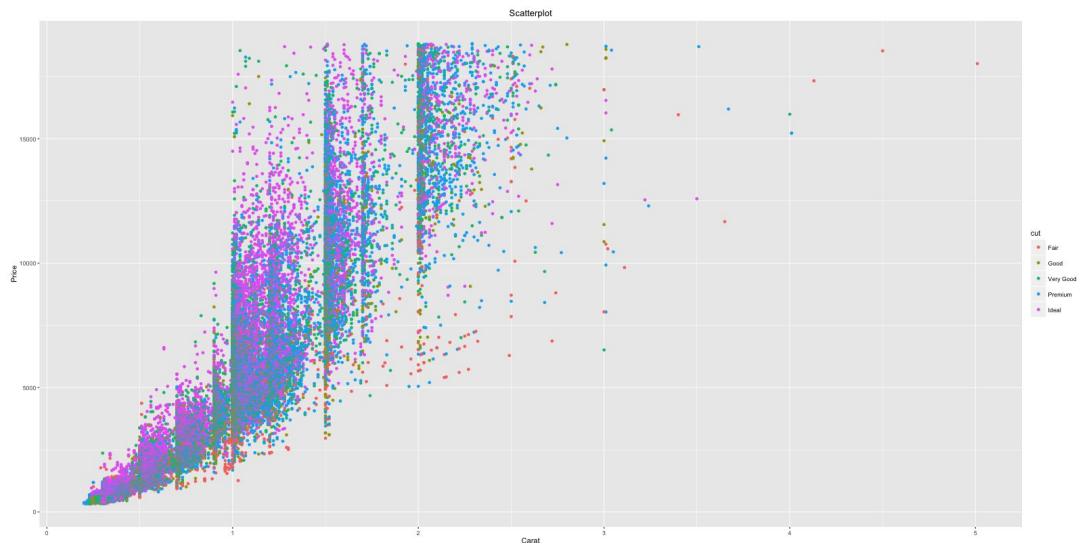
# ggplot2 Intro

```
library(ggplot2)
ggplot(diamonds) # if only the dataset is known.
ggplot(diamonds, aes(x=carat)) # if only X-axis is known. The Y-axis can be specified in r
espective geoms.
ggplot(diamonds, aes(x=carat, y=price)) # if both X and Y axes are fixed for all layers.
ggplot(diamonds, aes(x=carat, color=cut)) # Each category of the 'cut' variable will now h
ave a distinct color, once a geom is added.
```

ref: [How to make any plot in ggplot2?](#)

# ggplot2 Intro

```
library(ggplot2)
gg <- ggplot(diamonds, aes(x=carat, y=price, color=cut)) + geom_point() + labs(title="Scatterplot", x="Carat", y="Price") # add axis labels and plot title.
print(gg)
```



ref: [How to make any plot in ggplot2?](#)



# **Web Science:**

# **Intro to InfoVis with R and Python**

## **(Part 4 - Charts with Python)**

CS 432/532

Old Dominion University



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

# Ways to Plot Data in Python

- [Matplotlib](#)
  - most popular Python plotting library, similar to Matplot
- [Seaborn](#)
  - high-level interface to Matplotlib
- [Plotly](#)
  - interactive, open-source, and browser-based graphing library, built on top of D3
- [Bokeh](#)
  - interactive visualization library for modern web browsers, outputs plots as HTML files
- [Altair](#)
  - declarative statistical visualization library for Python, based on Vega and Vega-Lite
- [Pygal](#)
  - focus on visual appearance, produces SVG plots
- [Pandas](#)
  - popular data science library for Python, visualization is wrapper around Matplotlib

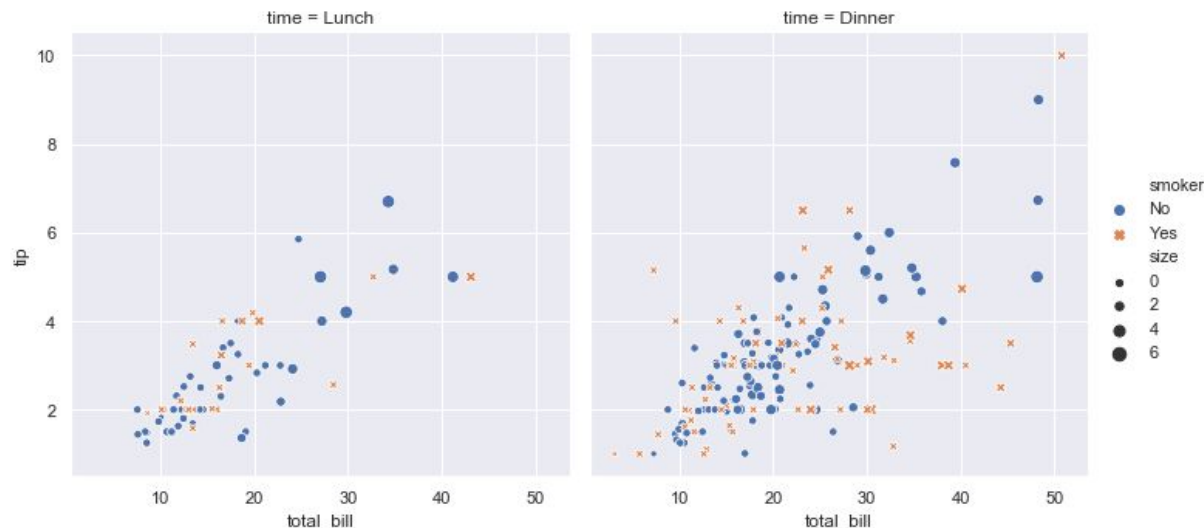
ref: [The 7 most popular ways to plot data in Python](#)

# Seaborn

```
import seaborn as sns
sns.set()
tips = sns.load_dataset("tips")
sns.relplot(x="total_bill", y="tip", col="time",
            hue="smoker", style="smoker", size="size",
            data=tips);
```

**Note:** A couple updates to the Google Colab notebook have been made since the lecture video was recorded:

- `distplot()` is deprecated, so that has been replaced with `histplot()`
- Seaborn has added the `ecdfplot()` function, so I've added an example using that.



# Objectives

- Distinguish between categorical and ordered attributes.
- Explain how marks and channels are related.
- Distinguish between the identity channel type and the magnitude channel type and indicate which channels belong to each type.
- Distinguish between the principles of expressiveness and effectiveness in visual encoding.
- List the channels for ordered attributes in order from most effective to least effective.
- Explain how the concepts of express, separate, order, and align all relate to arranging tabular data.
- Differentiate between line charts and bar charts and explain when each is appropriate.
- Explain how the boxplot idiom can characterize a distribution.
- Use R to create a bar chart, line chart, and scatterplot.
- Use Python charting libraries to create a bar chart, line chart, and scatterplot.