# Revision Notes – Data Visualisation

## Why Visualise Data?

In a world with constant flows of data, we can very quickly find ourselves swamped with Big Data with real no meaning to it. Visualising this data gives us a better idea of what we are looking at without visualisation. Information Overload.

Takes a long time to comprehend long lists of numbers at an instant

Easier to make instant assumptions with visualisation

## Challenges

How to make use of all this data?

How do we make sense of the data?

How do we use this data in decision-making?

We want to transform the data into information (understanding, insight) thus making it useful.

## What is Information Visualisation?

Definition:

* “…finding the artificial memory that best supports our natural means of perception.”

[Bertin 1967]

* “The use of computer-generated, interactive, visual representations of data to amplify cognition.”

[Card, Mackinlay, & Shneiderman 1999]

### What we aim for:

“To have the best of both sides”

* Machine:
  + Statistical Analysis
  + Data Mining
  + Data Management
  + Compression & Filtering
  + Semantics-based approaches
  + Graphics and Rendering
  + Information Visualisation
  + Human-centred Computing
  + Information Design
  + Decision Making Theory
  + Human Cognition
  + Perception
  + Visual Intelligence
* Human

## Common information Visualisation:

* Weather
* Political data
* Spotifiy wrapped
* Pie chart
* Bar chart
* Calendar view
* Wikipedia Page
  + Visualising Wikipedia Edit Evolution
    - <https://www.theguardian.com/world/interactive/2013/feb/12/state-of-the-union-reading-level>
* Transitions between various visualization types
* lines
* horizons
* areas
* stacked areas
* streamgraph
* overlapping areas
* grouped bars
* stacked bars
* bars
* donut

Why Create Visualization?

### “A Picture is worth a thousand words”

* Answer questions (or discover them)
* Make decisions
* See data in context
* Expand memory
* Support graphical calculation
* Find patterns
* Present argument or tell a story
* Inspire

## What are the Key Values of Information Visualisation?

### **Record** information

– Blueprints, photographs, seismographs,

Examples:

* Egyptian Hieroglyphs
* Blueprints
  + E.J. Marey’s sphygmograph (1854)
  + an instrument which produces a line recording the strength and rate of a person’s pulse.

### **Communicate** information to others

– Share and persuade

– Collaborate and revise

Examples:

* **Share and collaborate**
* **3D Diagram/scan of a hand to demonstrate knowledge.**
  + Doctors bone scan
  + DNA Helix
* **Persuade**
* **Nightingale’s Graph**
  + Diagram of the causes of mortality in the army in the East
* **Clarify/Revise**
  + **London Underground map**

### Analyse data to support **Reason**

- Find patterns / Discover errors in data

– Expand memory

– Develop and assess hypotheses

Examples:

* Find Patterns
* **The Most Powerful Brain**
* New York City Weather
* Looking at more data points
* Develop and Assess Hypothesis:
* London Cholera Map
* The closer to the Broad Street Water pump, the greater the number of deaths.
* The information helped convince the public a true sewage system was needed.
* Surprises in Data
* “The greatest value of a picture is when it forces us to notice what we never expected to see.”
* John Tukey, 1977
* “Contained within the data of any investigation is information that can yield conclusions to questions not even originally asked. That is, there can be surprises in the data...”
* W. Cleveland -The Elements of Graphing Data

## Different Stages of Visualisation

* Data
  + Raw Information
    - Data Transformation
    - Create a structural model (schema), mapping raw data into data tables.
  + Data Set
* Visual Form
  + Visual Form
    - Visual Mappings
    - Create a visual spatial model, transforming data tables into visual structures.
  + Views
* Task
  + User Task
    - View Transformation
    - Create view of the visual Structures by specifying graphical parameters such as position, scaling, and clipping.
    - Interaction

**Raw Information -> Dataset -> Visual Form -> Views -> User Task**

**Acquire** → **Parse** → **Filter** → **Mine** → **Represent** → **Refine** → **Interact**

* **Acquire**
  + Obtain the data, whether from a file on a disk or a source over a network.
* **Parse**
  + Provide some structure for the data’s meaning, and order it into categories.
* **Filter**
  + Remove all but the data of interest.
* **Mine**
  + Apply methods from statistics or data mining as a way to discern patterns or place the data in mathematical context.
* **Represent**
  + Choose a visual model, such as a bar, graph, list, or tree.
* **Refine**
  + Improve the basic representation to make it clearer and more visually engaging.
* **Interact**
  + Add methods for manipulating the data or controlling what features are visible.

**How to process data?**

* Data models
* Processing algorithms

**How to encode the data using images (the visual channel)?**

* Visual encoding (mapping)

## Data

**Data Models**

* Data models are formal descriptions
* Characterise data through three components:
  + **Objects** (Item of Interest)
    - Students, courses, semesters
  + **Attributes** (Properties of data)
    - Name, age, id, data, score
  + **Relations** (how two or more objects relate)
    - Student takes course, course during semester, etc.
* **Taxonomy of Data Types**
  + 1D (sets and sequences)
  + 2D (Maps)
  + 3D (Shapes)
  + nD (relational)
  + Temporal
  + Trees (hierarchies)
  + Networks (graphs)

## Nominal, Ordinal & Quantitative

**N - Nominal (label or categories)**

* Operations: =, ≠
* eg: Maths, Art (Course)

**O - Ordered**

* Operations: =, ≠, <, >
* eg: A, A-, B+, B (Grades)

**Q - Interval (location of zero arbitrary)**

* Operations: =, ≠, <, >, -
* Can measure distance or spans
* eg: (3.24, -1.2) (GPS)

**Q - Ratio (zero fixed)**

* Operations: =, ≠, <, >, -, %
* Can Measure ratios or proportions
* eg: 20, 19, 22, 21 (Age)

## Examples:

Variables: Cases

Name: Tom – Nominal

Age: 20 – Quantitative (Ratio)

Grade: A+ – Ordered

Course: Maths – Nominal

Entry Year: 2002 – Quantitative (Ratio)

### Dimensions and Measures

**Dimensions** (independent variables)

* + Discrete variables describing data:
    - N, O
  + Categories, dates, binned quantities

**Measures** (dependent variables)

* + Data values that can be aggregated:
    - Q
  + Numbers to be analysed
  + Aggregate as sum, count, avg

## Data Processing

### Data cleaning and filtering

* for quality control
* Remove (Outlier, missing data)
* Modify (conversion of format, etc)

### Data Adjustment

* Depends on your task and questions to ask.

### Relational algebra:

* Aggregation, mean, sort, projection
* Reformatting and integration
* **Data Cleaning and Filtering**
  + **Missing Data**
    - No measurements, redacted
  + **Erroneous Values**
    - Misspelling, outliers
  + **Types Conversion**
    - Zip code to lat-lon
  + **Entity Resolution**
    - different values for the same thing
  + **Data Integration**
    - effort/errors when combining data

Anticipate problems with your data.

Many research problems around these issues!

Exercise Skepticism

Check data quality and your assumptions.

Start with univariate summaries, then start to consider relationships among variables

Avoid premature fixation

**Image**

**Different Stages of Visualisation**

* Visual Language is a Sign System
  + Images perceived as a set of signs
  + Sender encodes information in signs
  + Recover decodes information from signs
* “Resemblance, order and proportion are three sign fields in graphics”
  + Jacques Bertin
* Information in Hue and Value
  + Value is perceived as ordered
    - Encode ordinal variables (O)
    - Encode continuous variables (Q)
  + Hue is normal perceived as unordered
    - Encode nominal variables (N) using colour
* Rectangle area: Market Cap (Q)
* Rectangle position: Market Sector (N)
* Color Hue: Loss vs Gain (N, O)
* Color Value: Magnitude of loss or gain (Q)

## Design Criteria

What design criteria should we follow?

### Choosing Visual Encodings

Assume k visual encodings and n data attributes. We would like to pick the “best” encoding among a combinatorial set of possibilities of size (n+1)k

Principle of Consistency

– The properties of the image (visual variables) should match the properties of the data.

Principle of Importance Ordering

– Encode the most important information in the most effective way.

## Expressiveness:

Tell the truth

A set of facts is expressible in a visual language if the sentences (i.e. the visualizations) in the language (1) express all the facts in the set of data, and (2) only the facts in the data.

Unable to express all facts in a layout (fails first criterion)

Expresses information not inherent in the dataset (fails second criterion)

A length is interpreted as a quantitative value.

## Effectiveness

Use proper encoding

A visualization is more effective than another visualization if the information conveyed by one visualization is more readily perceived than the information in the other visualization.

Accuracy Ranking for Quantitative Information

Accuracy Ranking for Nominal/Ordinal Information?

Quantitative- We can use, but not so accurate.

Nominal - problematic if there are too many categories; • Can be expected to encode ordinal information

Conjectured Effectiveness of Encodings by Data Type

Nominal/ Ordinal variables: detect differences • Quantitative variables: estimate magnitudes

Mackinlay’s Design Algorithm • APT - “A Presentation Tool”, 1986 • User formally specifies data model and type – Input: ordered list of data variables to show • APT searches over design space – Test expressiveness of each visual encoding Generate encodings that pass test – Rank by perceptual effectiveness criteria • Output the “most effective” visualization

Automatically generate chart for car data • Input variables: – Price – Mileage – Repair – Weight

Limitations of APT? • Does not cover many visualization techniques – Networks, hierarchies, maps, diagrams – Also: 3D structure, animation, illustration, ... • Does not consider interaction • Does not consider semantics / conventions • Assumes single visualization as output

Summary of Design Criteria • Choose expressive and effective encodings – Rule-based tests of expressiveness – Perceptual effectiveness rankings • Prioritizes encodings that are most easily/accurately interpreted • Principle of Importance Ordering: Encode more important information more effectively (Mackinlay) • Question: how do we establish effectiveness criteria? – Subject of the visual perception lecture...

## Graphs

For uni, bi and tri-variate data

How Many Variables? • Data sets of dimensions 1, 2, 3 are common • Number of variables per class – 1 - Univariate data – 2 - Bivariate data – 3 - Trivariate data – >3 - Hypervariate data

Data Dimensions – 1 - Univariate data – 2 - Bivariate data – 3 - Trivariate data – >3 - Hypervariate data • Data Types – Nominal, Ordinal, Quantitative • Visualization Representations – Points, Lines, Bars, Boxes

Components of Graphs • Framework – Measurement types, scale • Content (Specifier) – Marks, lines, points • Labels – Title, axes, ticks

Points, Lines, Bars, Boxes • Points – Useful in scatterplots for 2-values – Can replace bars when scale doesn’t start at 0 • Lines – Connect values in a series – Show changes, trends, patterns – Not for a set of nominal or ordinal values • Bars – Emphasizes individual values – Good for comparing individual values • Boxes – Shows a distribution of values

Univariate Data • In univariate representations, we often think of the data case as being shown along one dimension, and the value in another. • Statistical view – Independent variable on xaxis (data case) – Track dependent variable along y-axis (value)

Bivariate Data • Scatter plot is commonly used • Each mark is now a data case • Objective: – Two variables, want to see relationship – Is there a linear, curved or random pattern?

Trivariate Data 3D scatter plot 2D + mark property Represent each variable in its own explicit way

Dot Plots • When to use: – When analyzing values that are spaced at irregular intervals – continuous, quantitative, univariate data

Scatter Plot • When to use: – To compare how two quantitative variables change – continuous, quantitative, bivariate data – relationships for two variables

Line Graphs • When to use: – When quantitative values change during a continuous period of time (for more than one group) – Time series data (Non-cyclical data over time)

Bump Chart • When to use: – Similar to line graph – Y-axis: rank rather than (continuous) values

Area Graph • When to use: – Commonly one compares with an area chart two or more quantities. – The area between axis and line are commonly emphasized with colors and textures.

Radar Graphs • When to use: – When you want to represent data across the cyclical nature of time – A two-dimensional chart of three or more quantitative variables represented on axes starting from the same point

Bar Graphs • When to use: – When you want to support the comparison of individual values between different groups – Can run vertically or horizontally

Stacked Bar Chart • When to use: – When you want to present the total in a clear way while comparing part-towhole relationship between different groups – Harder to compare the size of each categories

Histogram • When to use: – the most commonly used graph to show frequency distributions – Continuous, quantitative, univariate data

Pareto chart • When to use: – When analyzing data about the frequency of problems or causes in a process. – containing both bars and a line graph

Box Plots • When to use: – You want to show allow for comparison of data from different categories – graphically depicting groups of numerical data through their quartiles

Heat Maps • When to use: – When you want to display a large quantity of cyclical data (too much for radar) – Color choices: grayscales, rainbow, etc.

Crosstab Plot • When to use: – Comparing different groups while presenting values (count) – Similar to heatmap

Trellis Display • When to use: – Typically varies on one variable – Distribute different values of that variable across views

Hybrid: Map based Heatmap • When to use: – When you want to display a large quantity of cyclical data over different geo-locations

**Design Principles**

**Expressiveness**

* Visualizations must represent all and only the data facts.

**Effectiveness**

* Visualization choice should enhance the viewer's ability to perceive information easily (Mackinlay, 1986).

**Data Dimensions and Types**

**Data Dimensions:**

* Univariate, Bivariate, Trivariate, and Hypervariate (3+ variables).

**Data Types:**

* Nominal, Ordinal, Quantitative.

**Visualisation Representations:**

* Points, Lines, Bars, Boxes.

**Design Challenges for Multivariate Data**

* **Example Data**: Dog characteristics - Variety (N), Group (N), Size (O), Smartness (N), Popularity (Q), Ranking (Q).
* **Projection to 2D**:
  + Use visual mapping to display multidimensional data in 2D.
* **Presentation Methods**:
  + **Textual** (Tables): Good for precise lookups but lacks spatial relationship insights.
  + **Symbolic** (Graphs): Effective for conveying relationships and trends.

**Visualisation Techniques**

* **Iconic Representations (Glyphs)**:
  + Glyphs symbolize data cases using visual properties to represent different data variables.
* **Examples of Techniques**:
  + Chernoff Faces: Uses human facial features to encode data dimensions.
  + Parallel Coordinates: Each variable is aligned horizontally with vertical lines representing variable values.

**Specific Multivariate Visualisation Tools**

* **Chernoff Faces**: Employs facial expressions to create an emotional association with data.
* **Table Lens**:
  + Converts tables into visual bar representations, enhancing readability and analysis.
* **Mosaic Plot**:
  + Effective for categorical data, visualises relationships among multiple categorical variables.

**Strategies for Effective Visualisation**

* **Avoid Over-Encoding**: Keep visualisations clear and focused on relevant information.
* **Utilise Small Multiples**:
  + Break down data into manageable parts for comparison.
* **Coordinate Multiple Views**:
  + Use interaction tools to generate appropriate visualisations based on context.

**Visualisation Tools Overview**

* **Chart Typology**:
  + Template-based, easy but limited.
* **Component Architecture**:
  + Supports combinatorial designs but requires software engineering.
* **Declarative Languages (e.g., ggplot2)**:
  + Descriptive programming that separates specification from execution, simplifying iteration and performance.

Visualisation Tools

ExpresssivenessVVVVV

Chart Typologies

* Excel, many eyes, google charts

Visual Analysis Grammars

* VizQL,, ggplot2

Visualisation Grammars

* Protovis, D3,js

Component Architectives

* Prefuse, flare, improvise, VTK

Graphics APIs

* Processing, OpenGL, Java2D

Ease of Use^^^^^^

**Interactions in Visualization**

**Overview**

* **Interaction**: Key concept in visualization, allows users to explore, adjust, and better understand data
* **Why Interaction is Important**:
  + Large datasets are hard to interpret without interaction
  + Interaction helps users engage with data incrementally or in focused ways
  + Address limitations like screen space, cognitive load, and time constraints

**Defining "Interactive"**

* **Response Times**:
  + **0.1 sec**: Animation, sliders (visual continuity)
  + **1 sec**: System response time (conversational flow)
  + **10 sec**: Cognitive response time

**Types of Interactions (Taxonomy of Interactions)**

1. **Select**
   * Mark items of interest for further actions
   * Examples: Selecting a location on Google Maps, cell selection in tables
   * **Generalized Selection**: Select items based on attributes rather than specific items
2. **Explore**
   * Examine different subsets of data
   * Example: Panning in Google Earth
3. **Reconfigure**
   * Change spatial arrangement to gain new perspectives
   * Examples: Sorting columns, rearranging scatter plot attributes
4. **Encode**
   * Change visual appearance (color, size, font, etc.)
   * Allows for different data representations within limited screen space
5. **Abstract/Elaborate**
   * Adjust level of detail (overview vs. details)
   * Examples: Drill-down in Treemap, details-on-demand
6. **Filter**
   * Display data conditionally based on specific criteria
   * Examples: Buttons for binary options, sliders for quantitative adjustments
   * **Dynamic Query**: Filtering as a SQL query, promoting exploration
7. **Connect**
   * Highlight relationships and associations
   * Example: Brushing to highlight cases across multiple views

**Challenges in Interaction**

* **Characterization**: Difficult to pin down due to varying user intents and techniques
* **User-centered vs. System-centered**:
  + User intent (purpose of interaction) impacts design choices

**User Intent-Based Categorization of Interactions**

* **7 Categories of User Intent**:
  1. Select
  2. Explore
  3. Reconfigure
  4. Encode
  5. Abstract/Elaborate
  6. Filter
  7. Connect

**Summary**

* Interaction enables a dialogue between user and visualization
* Facilitates data exploration and comprehension
* Crucial for multi-view systems, especially with large or complex datasets

## Fundamentals of R

Data Manipulations:

R is for Raw information -> Dataset part of the seven stages of visualisation

Libraries for Data Manipulations

• Packages

– plyr

– data.table

– reshape2

– doBY

– sqldf

– and many more

### dplyr: A Grammar of Data Manipulation

• Very intuitive, once you understand the basics

• Very fast

– Created with execution times in mind

• Easy for those migrating from the SQL world

• When written well, your code reads like a “recipe”

• “Code the way you think”

### Pipe Operator

• Library(maggritr)

– A R package launched on Jan 2014

– A “magic” operator called the PIPE was introduced

– %>%

– i.e. “AND THEN”, “PIPE TO”

dplyr takes the %>% operator and uses it to great effect for manipulating data frames

Works only with data frames

5 basic “verbs” work for 90% of data

Filter() – select a subset of ROWS by conditions

Arrange() – Reorder ROWS in a data frame

Select() – Select the Columns of interest

Mutate() – Create New columns based on existing columns (mutations!)

Summarise() – Aggregate values for each group, reduces to single value

## 5 Basic Verbs

Filter Rows – filter()

Usage: filter(data, condition)

- Returns a subset of rows

- Multiple Conditions can be supplied

- They are combined with an AND

Select Column Types – select()

Usage: select(data, columns)

Arrange Rows (SORT) – arrange()

Usage: arrange(data, column\_to\_sort\_by)

* Returns a reordered set of rows
* Mutliple inputs are arranged from left-to-right

Mutate (into something new) – mutate()

Usage: mutate(data, new\_col = func(oldcolumns)

Creates new columns, that are functions of existing variables.

Summarise by groups – summarise()

Usage: Group\_by(data, column\_to\_group) %>% summarize(function\_of\_variable)

* Group\_by creates groups of data
* Summarise aggregates the data for each group

## Data Processing with R

R Advanced: check your data

Data Cleaning and Filtering • Exercise Skepticism • Check data quality and your assumptions. • Start with univariate summaries, then start to consider relationships among variables. • Avoid premature fixation!

ggplot2 • ‘gg’ is for ‘grammar of graphics’ (term by Lee Wilkinson) • A set of terms that defines the basic components of a plot • Used to produce figures using coherent, consistent syntax • Easy to get started, plenty of power for complex figures

Building a Plot in ggplot2 data to visualize (a data frame) map variables to aesthetic attributes geometric objects – what you see (points, bars, etc) scales map values from data to aesthetic space

Data • Must be a data frame, pulled into the ggplot() object • Example: the iris dataset – A multivariate dataset introduced by Fisher (1936)

Aesthetics (aes) • How your data are represented visually – i.e. mapping – Which data on the x – Which data on the y – But also: color, size, shape, transparency

Geometry (geom) • The geometric objects in the plot • Points, lines, polygons, etc. • Shortcut functions – geom\_point() – geom\_bar() – geom\_line()

Building a Plot in ggplot2 data to visualize (a data frame) map variables to aesthetic attributes geometric objects – what you see (points, bars, etc) scales map values from data to aesthetic space

An Example: Visualizing iris Data • ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) + geom\_point()

Changing the Aesthetics of a geom: increase the size of points • ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) + geom\_point(size = 3)

Changing the aesthetics of a geom: Add some color • ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) + geom\_point(size = 3)

Changing the aesthetics of a geom: Differentiate points by shape • ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) + geom\_point(aes(shape = Species), size = 3)

Uni-/Bivariate visualizations for checking your data and exercising skepticism when doing data manipulation

Histograms and Bar plots

X axis is Height of bar represents Common Name

Continuous Count Histogram

Discrete Count Bar Graph

Continuous Value Bar graph

Discrete Value Bar graph