Notes(2): Missing data, d3, principle & design

1. Missing Data:

   What is Missing Data?
   - Missing data are simply unobserved values in a data set, but they can be of different types and may be missing for different reasons.
   - The first type of missing data mechanism is **"missing completely at random" (MCAR)**. Statistically speaking, this means that the probability of the observation being missing is not dependent on any observed or unobserved covariates. Example:
   - The second type of missing data mechanism is **"missing at random" (MAR)**. In this case, the probability of the observation being missing is dependent only on observed values; that is, it is conditionally missing.
   - The final type of missing data mechanism is **"missing not at random" (MNAR)**. In this instance, the probability of an observation being missing is dependent on the missing responses or an unobserved covariate.

   Why do we care about missing data?
   - Why not just take the complete cases at hand to analyze the data? Such an approach, which is the default in most statistical software, can have a drastic impact on the statistical inference drawn from the data.
   - There are several other reasons why missing data should not simply be ignored. Using only complete cases reduces sample size and, therefore, reduces power.

   Before you do anything:

1. Step 1: observe the missingness.
   - Completely random: Just throw out the cases
   - Random: create a new variable to model it
   - Not Random: find the cause
2. Step 2: Decide on best analysis strategy to yeild the least biased estimates
   - Deletion
   - Single Imputation
   - Model Based

   Types:
   - Listwise Deletion:

Listwise deletion means that any individual in a data set is deleted from an analysis if they're missing data on *any* variable in the analysis.
When Listwise Deletion Works:

i. The Data are Missing Completely at Random
ii. You have sufficient power anyway, even though you lost part of your data set

- Pairwise Deletion:
  pairwise deletion only removes the specific missing values from the analysis (not the entire case).
  In other words, all available data is included
  Pairwise deletion is useful when sample size is small or missing values are large because there are not many values to begin with, so why omit even more with listwise deletion.

- Imputation - mean/mode substitution
  - complete case analysis
  - reduces the variability
  - weakens the covariance and correlation estimation

- Dummy Variable creation: - indicator for missing value
  - complete case analysis
  - gives biased results
- Regression Imputation: - prediction
  - info from observed data
  - overestimates model fit
  - weakens variation
- Maximum Likelihood Imputation: - produce highest log-likelihood
  - consider observations/ independent variables
  - take into account the info you know so not so biased
  - missing variability so harder to have conclusive result
  - might need adjustments to reduce the bias

2. Principle and Design:
   - Is there a "wall" between domains in graphical displays?
     You need to rely on some tools to help you design the viz in a more advanced and sophisticated way. Tools make it easier for all of use to be designers. The wall is more ambiguous today because of these new technologies.

- We are all designers and not just programmers. Memes can also be programmable.
- We spend most of our time being users but spend less time being designers. So it is useful when we design stuff because then we can think as users and design in a way that is much more clear and understandable to any layman.
- We change priority in applying colors. In kisses case we use categorical numbers in sequential colors.

- Use perception rules to guide the design process
  we know as users how to interpret it
  keep in mind
  by perception rules: the brain does not tell you but you just recognize them
  it includes : we have to understand data type (7 types) and recognize viz tasks and lastly other design choices made by the user

- Data Ink ratio: Tufte. More standard way.
  Minimalistic bar chart. Focus on the data and not the rest of the stuff
  Left chart uses extra ink for background. If you can minimize it, we can focus more on the data.
  But people like colors. People like eye appeasing engagement with the user.

- Underground London tunnel. Different colors show different trains.
  London tube line. The thyme river is useful because it gives reference to west and east bank
  Some landmarks are references for users to understand what is where.
  A) just users different colors for different trains. But headed in same dirrection.
  B) Different trains but on same line. So use shades of green .
  C) Sequential colors use a lot of space. Rather use the space to display more info.
  C combined all trains in same direction in one line.
  Vary differently based on visual elements

- Need to consider the domain in designing.
  Look in to literature
  Know the limits you can display
  Need key elements to improve the already available.
  Practice makes you read better.

- Sometimes we need cheat sheets to understand the trends, the qualities, the scheme we need to understand to read the viz

- Directions make it easier for viewer to know where to look at.
- Wow factors are created by all but need to use it properly to have a wow effect.
- Point of interests are highlighted. The technique is simple. No extra stuff. Generalization improve clarity. BEcause only imp details are emphasised. Balance the two: Manipulate the perception and communicate intent of viz.

3. D3:

Environment setup
Include D3 Library from CDN:
Eg:
```
<head>
    <script src="https://d3js.org/d3.v4.min.js"></script>
</head>
```

Select DOM Elements using D3

a. Types:
   i. d3.select(css-selector) -> Returns the first matching element in the HTML document based on specified css-selector
   ii. d3.selectAll(css-selector) -> Returns all the matching elements in the HTML document based on specified css-selector
b. Select by:
   i. Name
   ii. Id
   iii. CSS Class Name

DOM Manipulation using D3

Method	Description
text("content")	Gets or sets the text of the selected element
append("element name")	Adds an element inside the selected element but just before the end of the selected element.
insert("element name")	Inserts a new element in the selected element
remove()	Removes the specified element from the DOM

html("content")	Gets or sets the inner HTML of selected element
attr("name", "value")	Gets or sets an attribute on the selected element.
property("name", "value")	Gets or sets an attribute on the selected element.
style("name", "value")	Gets or sets the style of the selected element
classed("css class", bool)	Gets, adds or removes a css class from the selection

...