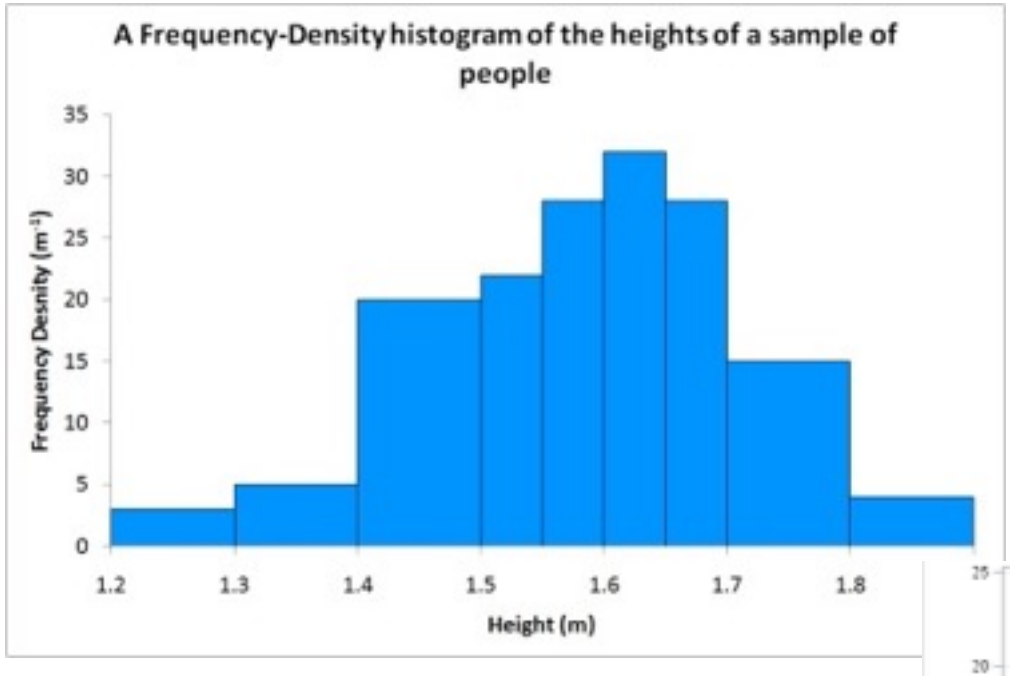
Part 1: Why visualise data?

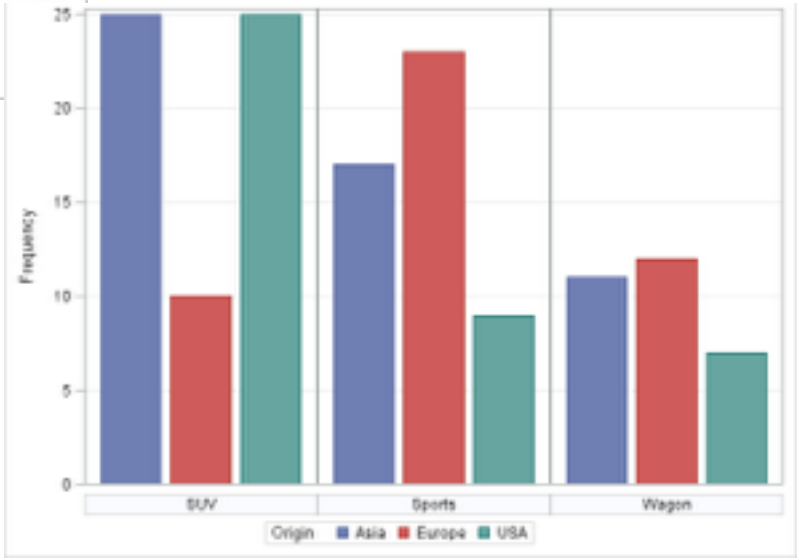
Why?

- Another way of summarising data
- Many at times , simple numeric summaries are not enough, for example while finding patterns in the data
- Images captures imagination of audience better than dry numbers in many cases

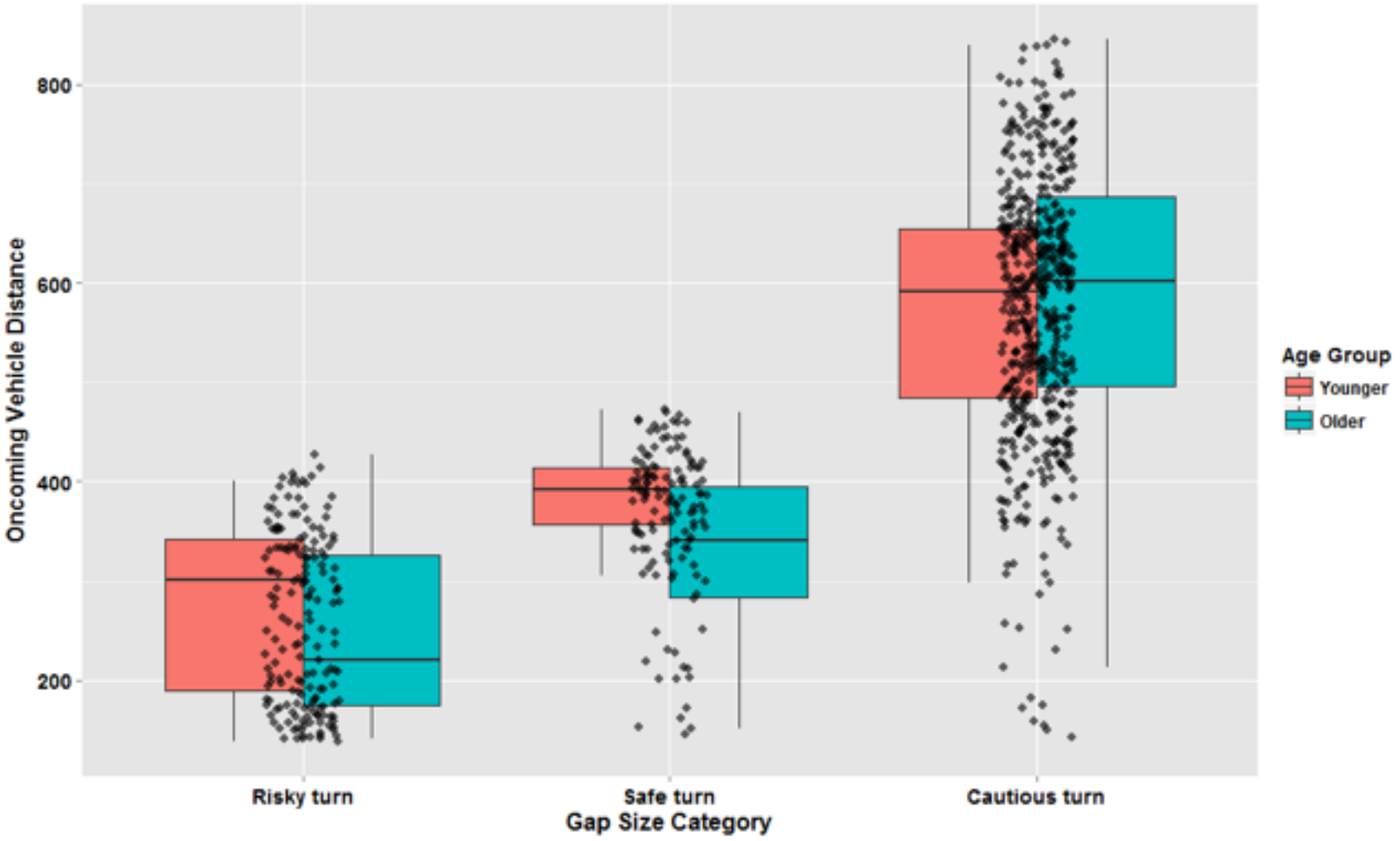
Popular Visualisations

Data Distribution : Histogram, Bar Plots

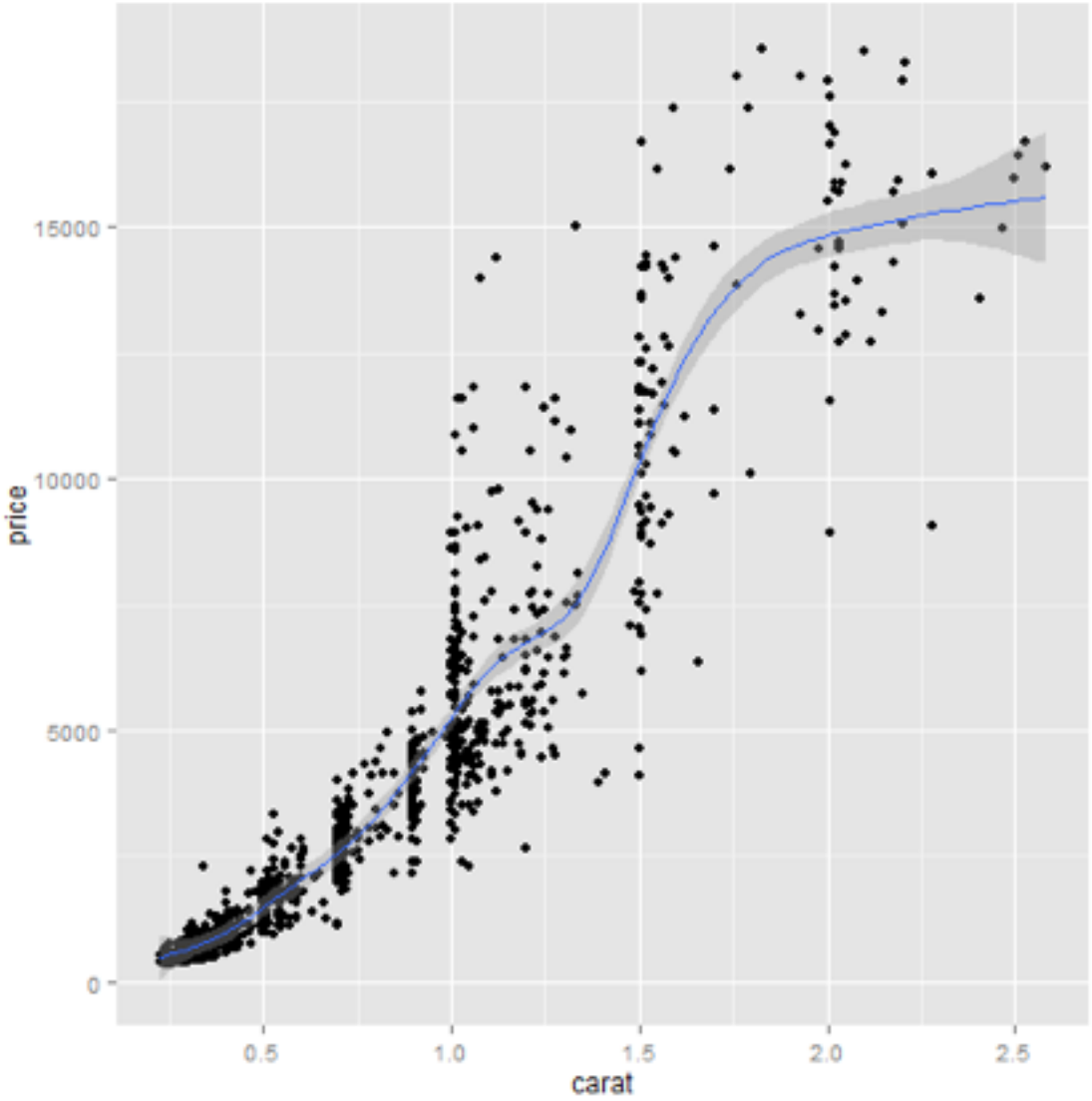




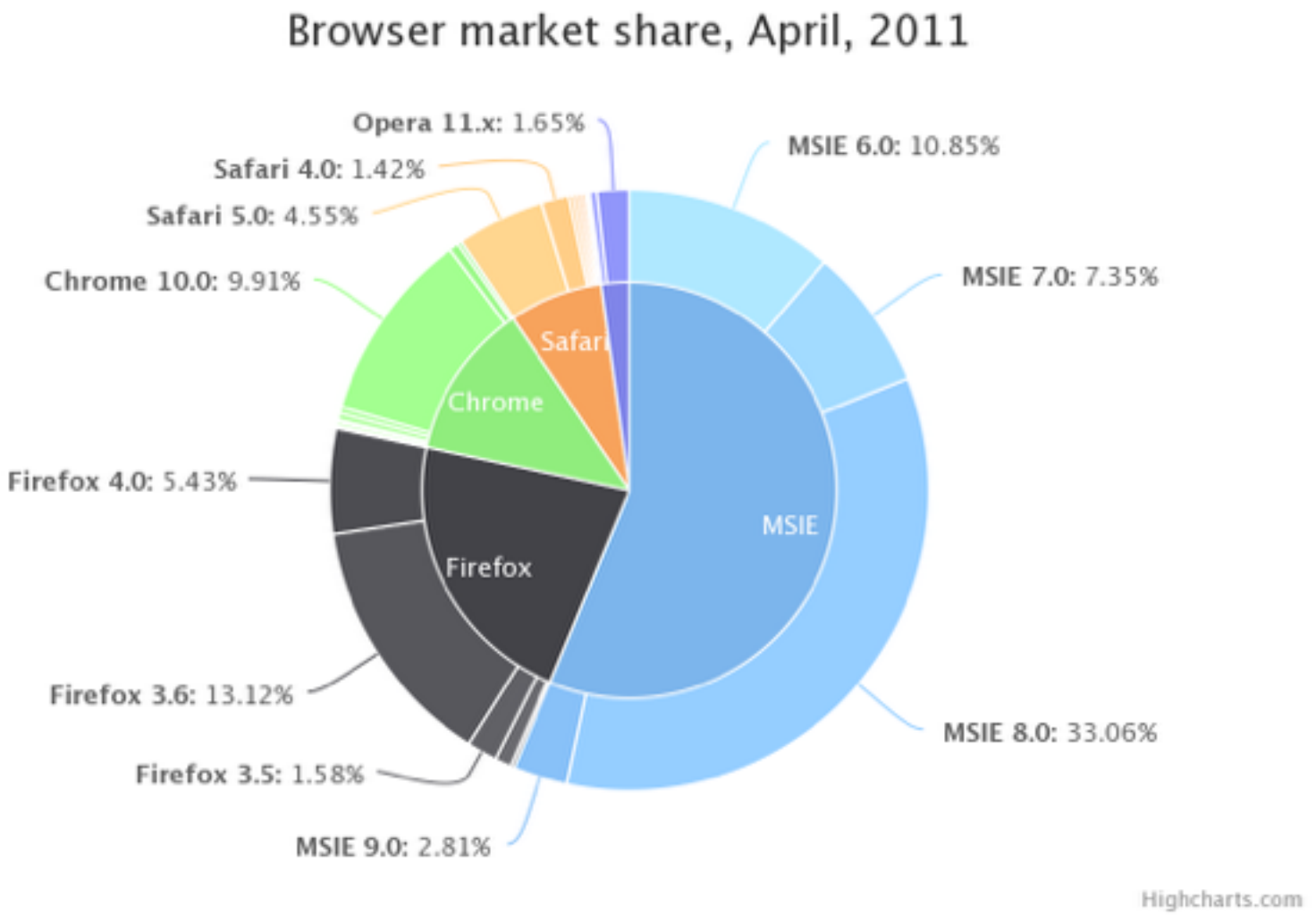
Data Distribution with outliers : Box Plots, jitter, violin etc



Variable Relationship : Scatter Plot , Smoothing Curve



Pie Charts : Bar plots in polar world



Part 2: Ways of ggplot2

Philosophy : Grammar of Graphics

- Idea is to streamline visualisation and do away with approaching each visualisation separately with some popular name.

- Visualisation process is made up of two steps
  - Aesthetics mapping for a geometry
  - Overlaying geometric layer on top of mappings to generate visualisation

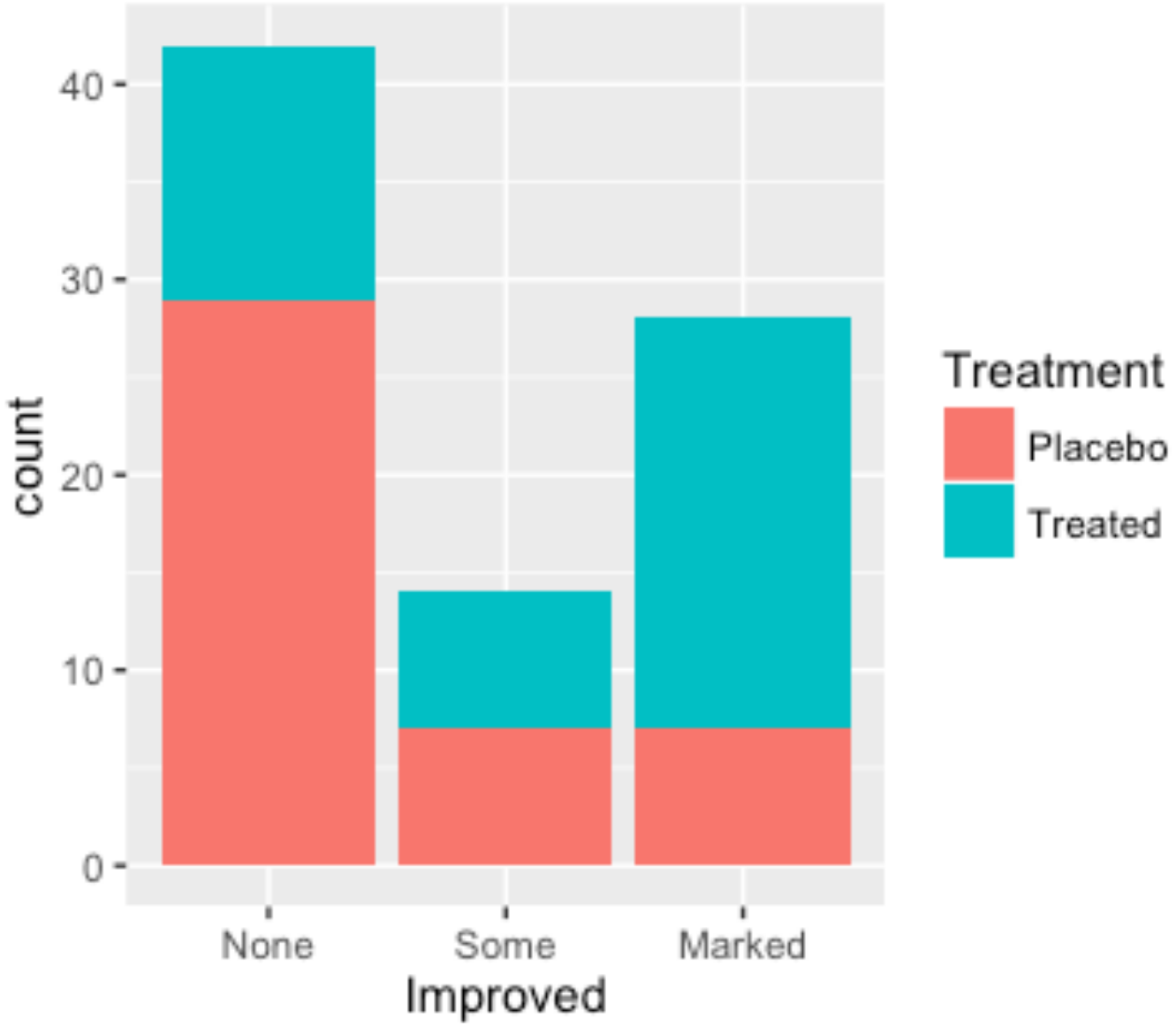
Aesthetics Mapping

- Each geometry can have [need to have] certain properties which we can use to display our data.
  - points : size , colour , shape , position (x and y axis)
  - bars : fill , colour , position (x axis) [y axis is reserved for frequency]

- Choice of these geometries depend on the context of the what we are trying to do
  - Visualising categorical variable : bar plots
  - Relation ship between two numeric vars : scatter plot , smoothing curves

Aesthetic Mappings : Example Point

Aesthetic Mappings : Example Bar



Non data mappings

- Each geometry has properties which may or may not be mapped to data
  - size , colour of points can be fixed to a certain value rather than mapped to data [treated as aesthetics]

- Visualisation in general can have properties which need not be necessarily mapped to data
  - axis labels , legends , titles , coordinate types

Building a visualisation & aesthetic inheritance

- Building a visualisation starts with constructing a data layer , with basic aesthetics mapping [aesthetic can be left blank too in data layer]

- Whatever geometry layers you add to this data layer , inherits aesthetics mapping from it

- In case of adding multiple geometries , all of them will inherit same mappings from data layer

- Each geometry layer can have its local aesthetic mapping which are not shared with other geometric layers

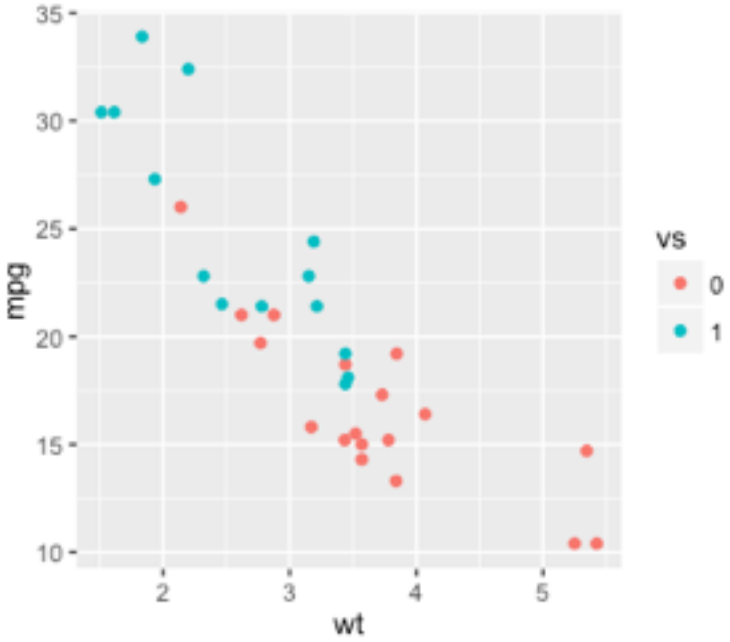
Benefit of this methodology

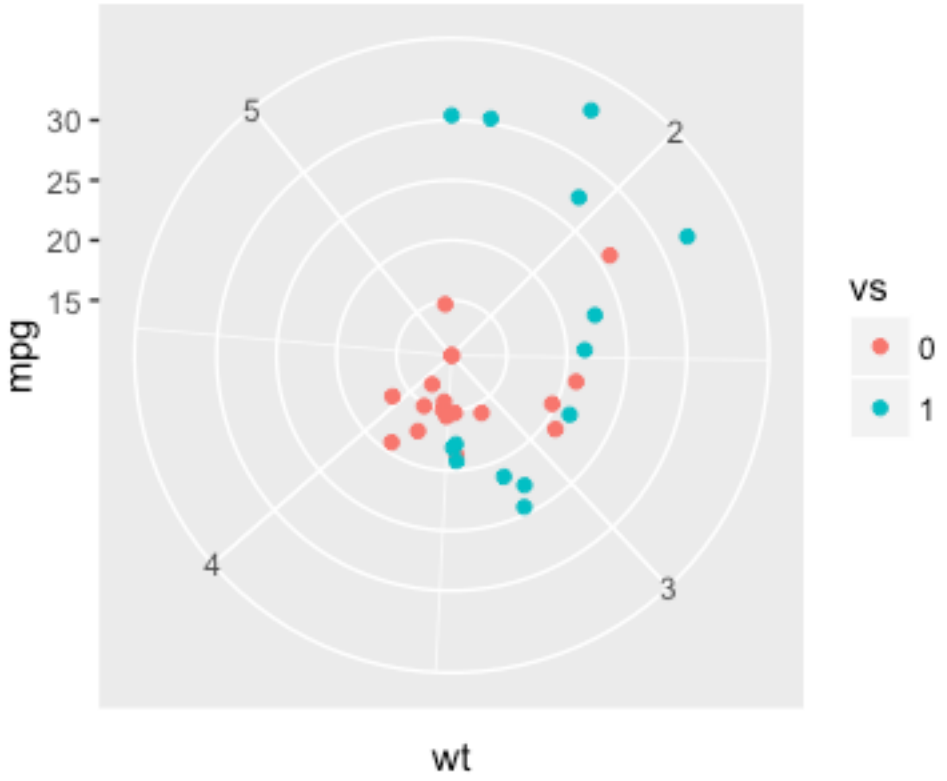
- That you do have a methodology !!
- You can have multiple geometries (visualisations) in single visualisation
- You can make visualisation , which need not have a standard name yet useful in context of your problem.
- No need to rely on specific visualisation type , but focus what makes them : geometries !!

Few useful aspects

- coord_* : Default coordinate system is euclidian [ranges of vars adjusted to fit in somewhat aesthetically appealing dimensions]
  - coord_polar : transforms your plot to a polar/ circular plot . x-y axis get mapped to radius and angle
- theme : function can be used to modify overall properties of visualisation : grid , labels etc
- scale_color_brewer : can be used to override default colour palate
- There is many more additional functions to deal with other aspects of visualisations

Example : Coordinate Transformation





Example : Coordinate Transformation

Part 3: Implementation in R

Implementation in R