*) Data-~~set~~ attributes

↳ make
↳ fuel type
↳ dimensions
↳ number of cylinders
↳ millage per gallon
↳ price

a) Data Abstraction

i) domain specific language to explain the data-set

*) The given data-set belongs to tabular data
*) Attributes (columns) and Items (rows)
*) It is a flat table.
*) Each column is attribute.
*) Each cell holds value for item-attribute pair.
*) There may be unique key which can be implicit.

ii) data-type of the attributes

*) Binary data-type attributes ⇒ doors
*) Quantitative attributes ⇒ length, width, height, weight, mpg, price
*) Qualitative attributes ⇒ make, fuel, doors, bodystyle, drive number of cylinders

*) Continuous attributes ⇒ length, width, height, weight, mpg, price
*) Discrete attributes ⇒ make, fuel, doors, body style, drive, number of cylinders

*) Ordinal attributes ⇒ doors
*) Nominal attributes ⇒ make, fuel, doors, body style, drive

b) Task

question-1 : How many drive-modes are there with respect to make [Bar chart] and 'audi' make ?

importing the data-set

*) df = read.csv ('cars.csv')

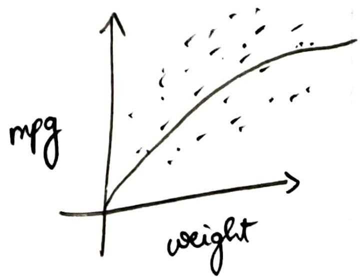
head (df)

*) barplot (df ['audi']
main = 'Driving Modes' in Audi'
xlab = 'Driving Modes available',
ylab = 'Make company',
names.arg = c ('rwd', 'fwd', '4wd')
horiz = FALSE)

question-2 : Comparing weight and mpg of the car.

Is the car's weight will be affecting the car's mpg ?

*) ggplot (data = df ,
mapping = aes (x = 'weight' ; y = 'mpg') +
geom - point ( ) +
geom - smooth (method = ' loess')
~~geom~~ geom -
)



question - 3

Is all the Make fuel is gas?

```
for i in df$ fuel
   if (df $ fuel == 'gas')

        cnt += 1

if (cnt == len (df $ fuel)) {
        cat ("All are gas")  }

else {
     cat ("Not all are gas")

   }
```

b) Vector color coding scheme

*) Divergence :

i) Computed using the definition with their paritial derivative.

ii) gives a good impression of where the flow enters and exits

*) Curl

i) very useful while practicing to find vortices = region of high vorticity.

ii) Highly important in flow simulation (i.e [aero, hydro] dynamics)

*) Vector glyphs

i) Sample points are chosen by avoiding uniform grids.

i) It is done by random sampling.

a) Contouring algorithm

for (each cell c in D) {

S = AE

for (each edge e = (p_i , p_j) of c) {

if ($v_i < v < v_j$) {

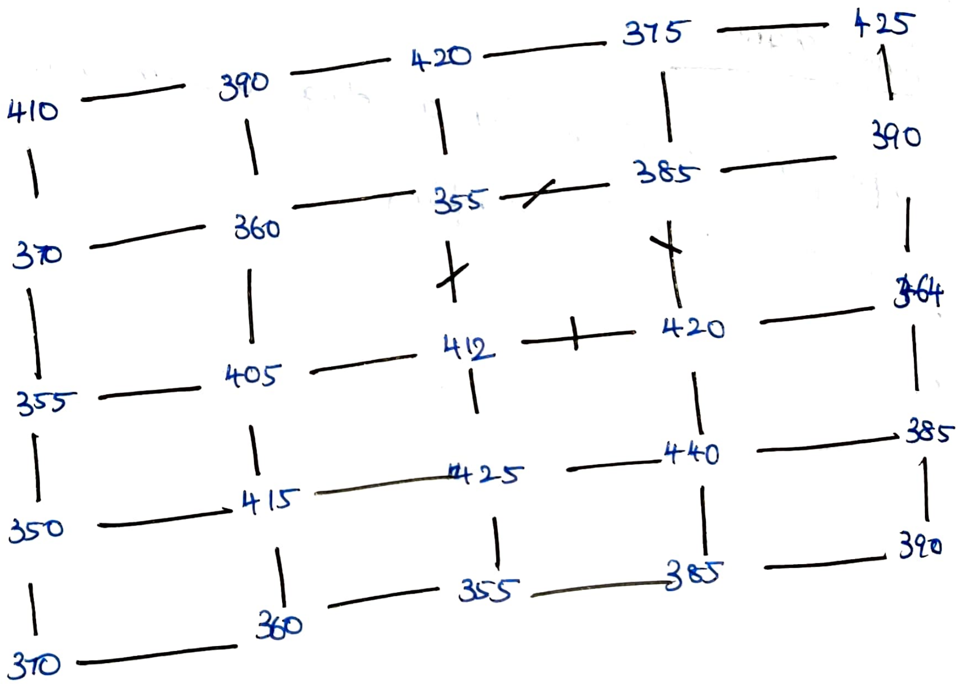
$$q = p_i (v_j - v) + p_j (v - v_i)$$

S = S E q

}

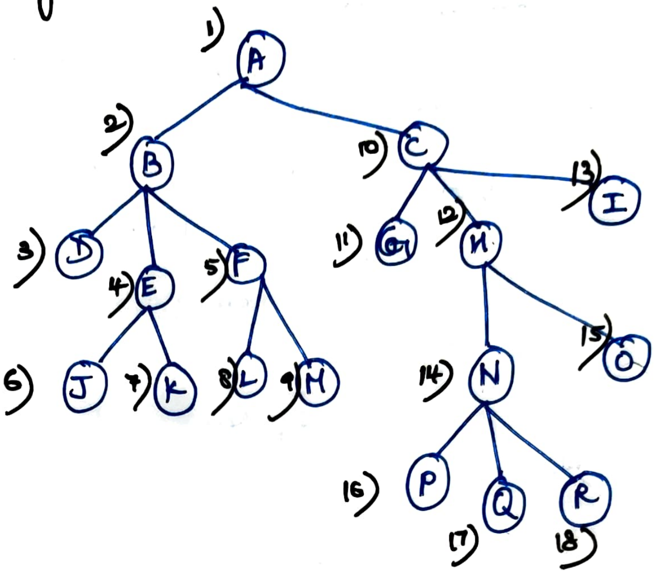
} connect points in S with lines to build contour

}

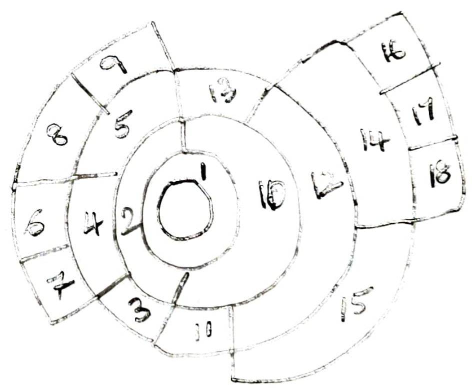


Question-3

a) Radial filling space



Radial Layout



*) Some times refers to sun-burst displays.

*) These will have the roots of the hierarchy at the center of the display and use nested rings to convey the layer of the hierarchy.

*) Each ring is divided into based on the number of nodes at that level.

start = start angle for a node [starting it is 0]

end = end angle for a Node (starting at 360)

origin = position of center of sun-burst eg: [0,0]

level = current level of the hierarchy (starting at 0)

width = thickness of each radial band - based on max-depth and display size.

sunburst (Node, start, end, level) # actual function call

End. Main Program

sunburst (node n, angle start, angle end, level l) {

if n is a terminal node (No children meaning)

draw-radial-selection (origin, start, end, l * width, (l+1) * width)

recursive call

return

for each child of n (child-i) {

sum up number of terminal nodes

computing the percentage of terminal nodes in n from each subtree (perc-i)

for each sub-tree {

comput start/end angle based on the size of sub-tree, order and angle range)

sunburst (child-i, start-i, end-i, l+i)

}

end: sunburst.

histog

b) *) vore : histogram : hist (df$ vore, col= 'blue', breaks = 50)

Stacked bar-chart

*) ggplot (df,
aes = (x= ~~bodywt~~ order, fill = vore) +
geom_bar (position = 'stack') +
labs ('Stacked Bar Chart')

density plot

*) ggplot (df,
aes = (x= bodwt, fill = vore) +
geom_density (alpha= 0.3) + ~~la~~
labs ('Density plot')