About the first part

Project.
The project I will be working on is based on the template data visualization.

Extension.

I will be building on the following extensions.

1. The waffle chart extension on the template.

2. An extension that will add a light spot at the area of the chart that the mouse is pointing at.

Coding techniques.

I will try to make my own design and code for the waffle extension based on the following points.

1. Description on the app in flow chart.

2. Specification of the values calculation.

3. Specifications of the functions that will be used.

4. Specification of the program flow and how the functions are called.

5. Specifications of the data structures used in the program.

6. Specification of how to produce the drawing.

7. Specification of the interactivity.

8. The data will be made up by a program.

Extension complexity and challenges

There is a number of challenges I am facing besides the programming language itself, I need a better grasp of the purpose and usage of the following :

1. Purpose and usage of the setup function, which variable and functions must be there and why.

2. The purpose and usage of the sketch, again what functions must reside there and why.

3. The purpose and usage of private and global variable, need a better grasp at encapsulation.

And, the calling of functions from within a function, from within another function and arguments passing is a bit confusing.

About the second part

App description

The visualization app will be augmented with two additional extensions. No changes will operate on the existing visualizations but rather additional features are added.

Extension 1 description

The extension consist of an implementation of the vis extension from the lectures in the template of data visualizations.

The structure from vis-extension will mostly transfer over, that it will include the following:

- The entire library.

- A sketch.js file.

- A box,js file.

- A waffle.js file.

- This particular implementation will include another file waffle.js file.

- Another data.js file will create the data in a csv format.

- The data.csv

The grid will consist of ten by ten fixed boxes because in my view it portrays well the idea that is being expressed, and at the same time it allows for a good correlation with the percent ratio. The audience will be allowed to read percent values from the data by counting boxes in the image directly.

It is my objective to describe with words or schematics how exactly a function works, or how a data structure is like an array with objects and so on before the coding task is tackled. The objective of having an idea as complete as possible will hopefully lead to less code re-writings and more productivity overall.
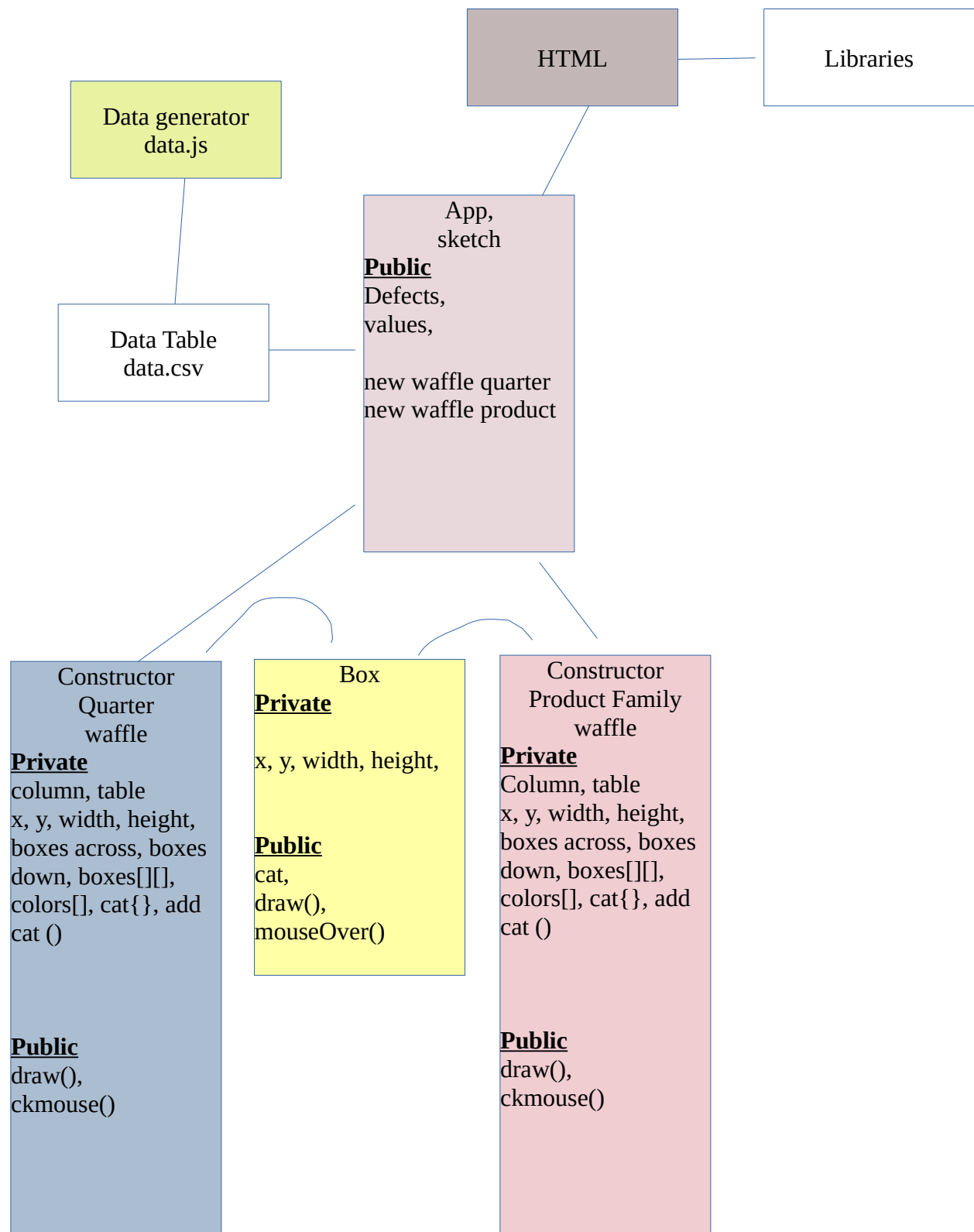
Extension 1 description

The extension adds a round shape spot-light to the area around the mouse, to highlight the area of the chart that is being explained in a meeting. As the mouse moves it shines a light on the focus area and the rest of the canvas is a bit dimmed but still visible.

This extension will be available for all the visualizations in the menu and will not interfere with the original design of those extensions, that is the mouseOver showing a legend of the box in the waffle chart will still be shown.

The extension will have a toggle that will appear in the visualization screen, for the user to be able turn on or off this feature, if she chooses to.
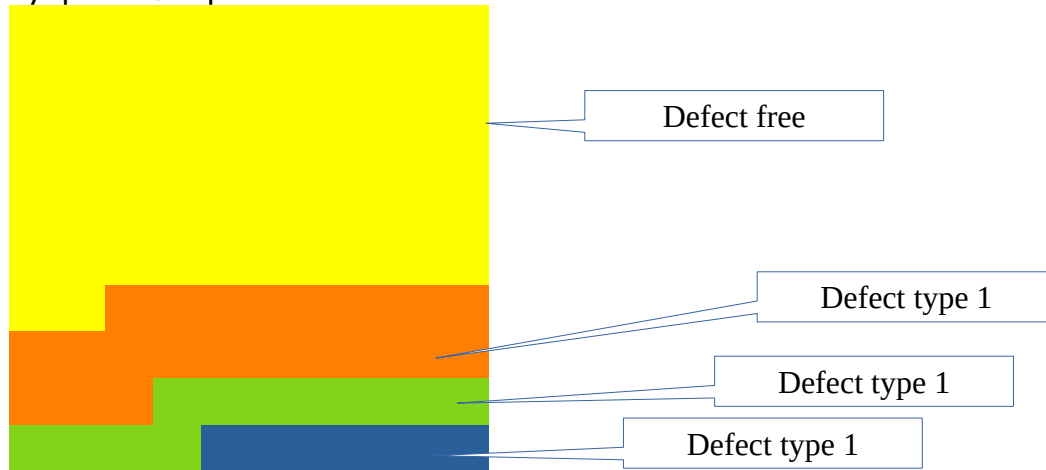
Second part

HTML

Libraries

Data generator
data.js

App,
sketch
Public
Defects,
values,

new waffle quarter
new waffle product

Data Table
data.csv

Constructor
Quarter
waffle
Private
column, table
x, y, width, height,
boxes across, boxes
down, boxes[][],
colors[], cat{}, add
cat ()

Public
draw(),
ckmouse()

Box
Private

x, y, width, height,

Public
cat,
draw(),
mouseOver()

Constructor
Product Family
waffle
Private
Column, table
x, y, width, height,
boxes across, boxes
down, boxes[][],
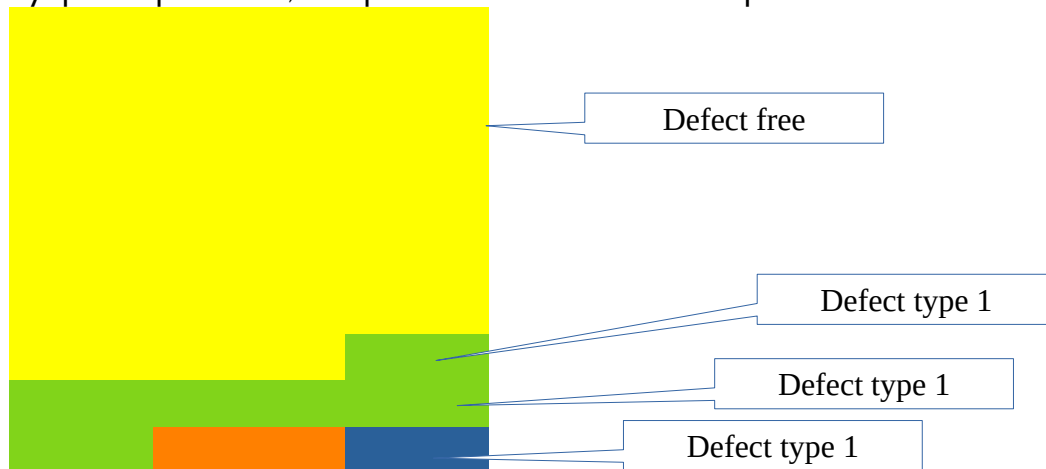colors[], cat{}, add
cat ()

Public
draw(),
ckmouse()

Second part
The interface will look like this, one waffle on top for the entire quarter and four waffles below with the breakdown for four family products.

By quarter , all products combined

Defect free

Defect type 1

Defect type 1

Defect type 1

By quarter product A, four product lines across for the quarter

Defect free

Defect type 1

Defect type 1

Defect type 1

Second part
The all products table data will look like this.

Batch	Product	flag
2021525	A	d1
2021525	B	df
2021525	C	d2
2021525	D	d3
2021526	A	df
…	…	…

For the all products waffle chart,
Total = sum of column Qty, all products all batches in the quarter.
D1 = sum (all defects type 1, all batches, all products).
D2 = sum (all defects type 2, all batches, all products).
D3 = sum (all defects type 3, all batches, all products).
Df = Total – sum (D1, D2, D3), the number of defect-free products.

The proportions are as follows, for the four categories.
pD1 = ceiling (D1 / Total), the ceiling function applied to obtain an integer.
pD2 = ceiling (D2 / Total), the ceiling function applied to obtain an integer.
pD3 = ceiling (D3 / Total), the ceiling function applied to obtain an integer.
pDf = 100 – sum (pD1, pD2, pD3)

The proportions above are also the number of squares per category in a 10 x 10 grid waffle.

For the product families may look as follows.

Batch	Product	Qty	Defect type 1	Defect type 2	Defect type 3
…	…	…	…	…	…
	A	Sum	Sum	Sum	Sum

The above formulas apply, but this time the
TotalA = sum (qty of product A in the quarter, all batches)

D1A = sum (all defects type 1, all batches, product A).
D2A = sum (all defects type 2, all batches, product A).
D3A = sum (all defects type 3, all batches, product A).
Df = TotalA – sum (D1A, D2B, D3C), the number of defect-free product A.

The proportions are as follows, for the four categories.
pD1A = ceiling (D1A / TotalA), the ceiling function applied to obtain an integer.
pD2A = ceiling (D2A / TotalA), the ceiling function applied to obtain an integer.
pD3A = ceiling (D3A / TotalA), the ceiling function applied to obtain an integer.
pDfA = 100 – sum (pD1A, pD2A, pD3A)

The corresponding formulas for the products B, C, and D.