THE UNIVERSITY OF SYDNEY

BIOL1008 Workshop 3

Data Sketching
Curves in Processing

What to do before Workshop 3:

1. Watch the video: **https://youtu.be/FliEk2riT2k**
2. Work through this sheet
3. Check the folder on Canvas for additional information you will need
4. Complete the pre-work quiz on Canvas

A little bit of visualisation theory

Let's have another look at one examples that we have seen a few times. Comparing Phil's activity to Emma's over the same week (maybe you could look at your own too). We can't see numerical values (that's not the goal of this particular visualisation) but the differences are quite clear. Including when Emma went to a music festival (can you guess which day that was?)

Emma:

Phil:

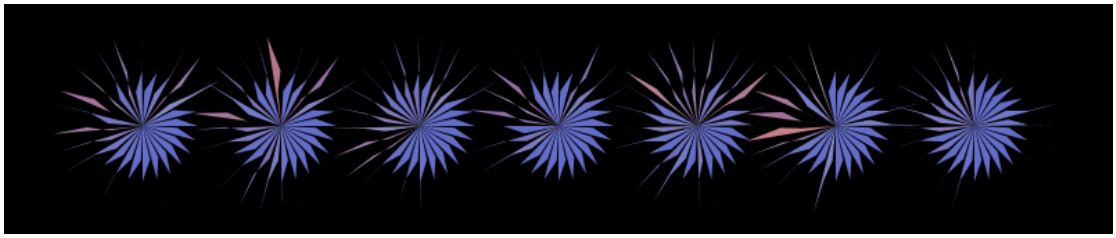


Figure 1: Physical Activity from the example code

But, why does visualisation work?

Figure 2 is from Colin Ware, a data visualisation researcher. It shows us some of the preattentive attributes of visual processing. The human brain has evolved to be excellent and paying attention to certain things, in fact, the brain makes comparisons between these attributes BEFORE you are consciously aware that you compare them. This has been very useful in helping humankind survive as long as we have: quickly comparing shape and movement helped early man avoid predators; comparing colours helps us avoid eating poisonous berries (as men were traditionally hunters, and women were traditionally gatherers, this *may* explain in evolutionary terms why it is very rare for women to be colour blind).

So, before we are consciously aware of it, the visual cortex is able to compare things like the position, orientation, colour, size, curvature and shape of things we see.

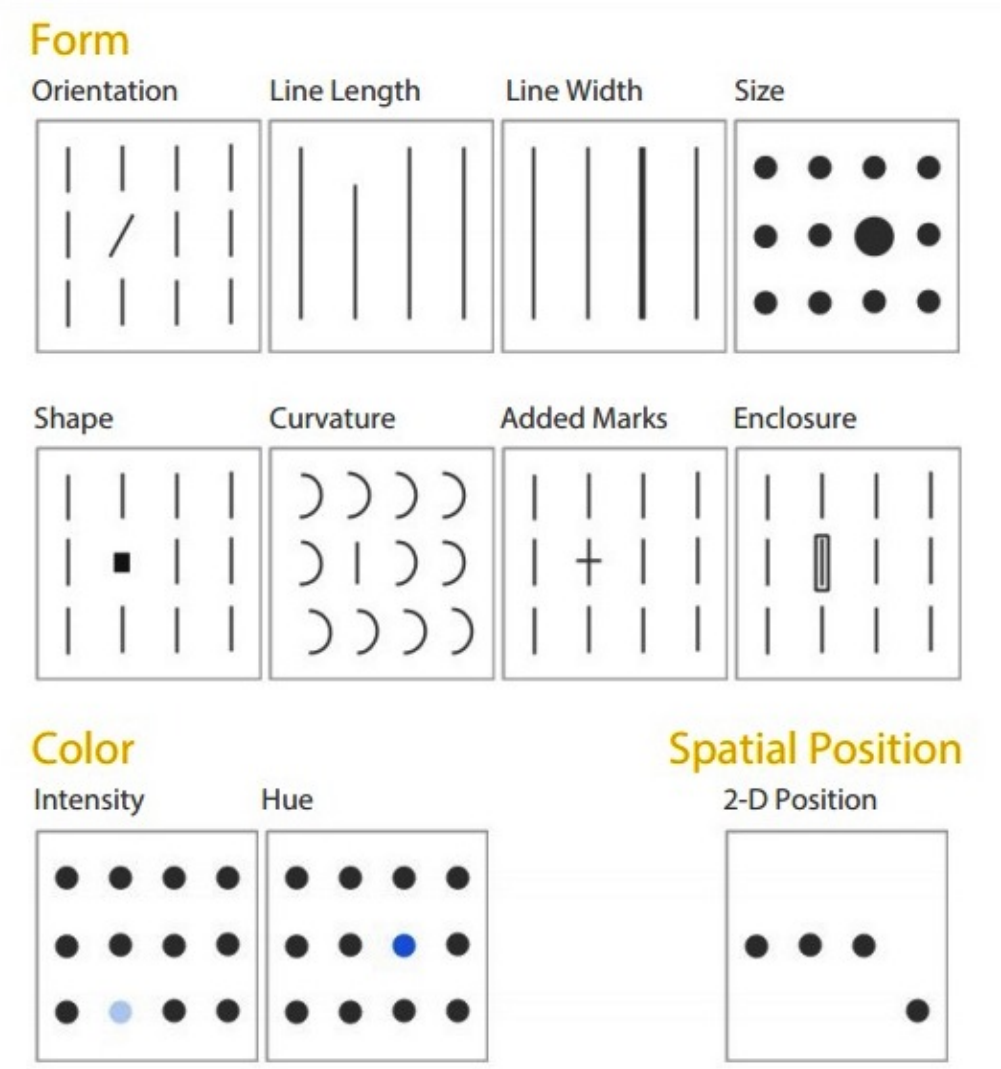


Figure 2: Colin Ware's Preattentive Visual Processing – see Ware, C. (2012). Information Visualization: Perception for Design (3rd ed.). San Diego: Morgan Kaufmann.

Mapping

We have used the `map()` function in Processing to scale a number from one value to another. Data visualisation is all about mapping. To visualise data is to map its value to a graphical attribute, such as colour, position, orientation or size. This allows us to leverage the visual processing functions of our brains. Your task in this assessment is to choose activity and/or nutrition data to map to visual attributes of an image.

Data Sketching

Step 1: get data
Step 2: ???
Step 3: visualisation

So how do we go from data to a visual representation? The data sketching method is one example used by visualisation designers Nadieh Bremer and Shirley Wu.
The example we would like you to look at is Shirley Wu's project from July.
http://www.datasketch.es/july/

In these project Nadieh and Shirley outline three steps:

1. How they got their data, why they chose it and what they wanted to do with it.
2. Sketch concepts by hand. This is MUCH faster than coming up with ideas using code.
3. Code the concept they designed. This process was highly iterative.

We will use a similar process, and outline it in the processing PDF:

1. Ask a question of the data.
    a. It should be short
    b. It can show data over the week, or compare your data to the class
2. Sketch some concepts by hand. The template designs in the workshop folder have some example shapes, and you can use them as a starting point.
3. Adapt the templates to show your own data.

curveVertex() and bezierVertex()

The templates use `curveVertex()` and `bezierVertex()` to draw shapes with smoother lines than we can make using `vertex()`. All of these functions need to be used in between `beginShape()` and `endShape()`. Here is how you use them:

vertex(x, y);

```
beginShape();
vertex(x1, y1);
vertex(x2, y2);
vertex(x3, y3); // etc...
endShape();
```

draws a straight line between each consecutive vertex location, noted by x and y coordinates. Can use as many vetex functions as you like between beginShape/endShape.

curveVertex(x, y);

```
beginShape();
curveVertex(hx1, hy1);
curveVertex(x2, y2);
curveVertex(x3, y3);
curveVertex(x4, y4);
curveVertex(hx5, hx6);
```

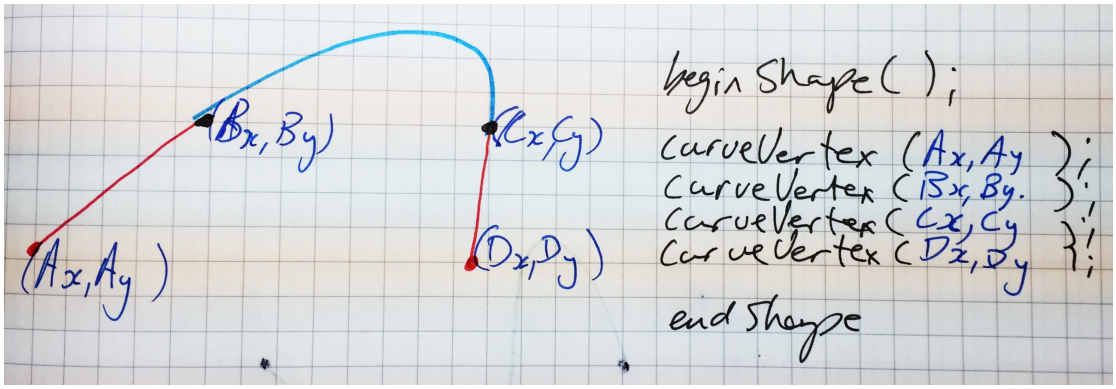


Figure 3 curve drawn by curveVertex

draws a curved line between each consecutive curveVertex location, noted by x and y coordinates. Can you as many curveVertex functions as you like between beginShape/endShape. Also, the first and last point will be hidden, as the curve shape is defined by these two points. You will need to use at least 4 curveVertex locations to draw a curve.

`bezierVertex(cpx1, cpy1, cpx2, cpy2, x3, y3);`

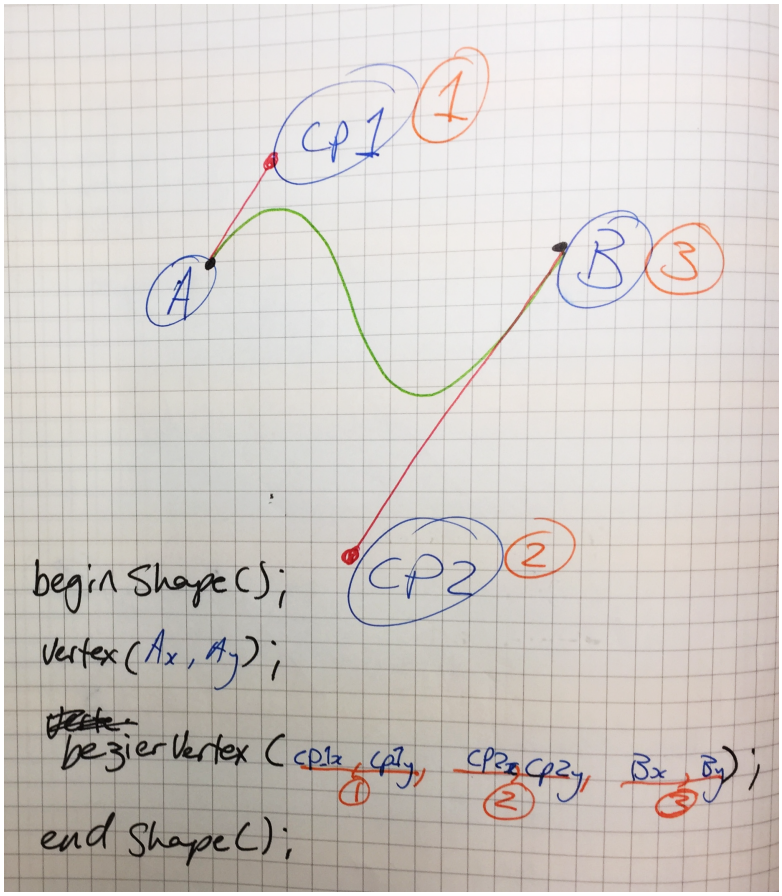


Figure 4 Curve drawn by bezierVertex

```
beginShape();
vertex(x, y);
bezierVertex(cpx1, cpy1, cpx2, cpy2, x3, y3);
endShape();
```

draws a Bézier curve between two points. The first bezierVertex after beginShape **must** come after a vertex function, as seen in the code above. The bezierVertex function takes 6 parameters: the x and y location of two control points (shown in red in Figure 4), which adjust the shape of the curve, followed by the x and y location where the curve ends.

You can choose whichever which kinds of shapes you wish to draw. The curves are more complex to use, but give you more freedom.

Working with Templates

The data folder has a processing file called `exampleShapes.pde` that includes 15 shapes that you can use as a starting point for your own templates.

Let's have a look at what you need to do.

1 Select the data you would like to compare

Think about the data you have collected and look on the datasets at how it is measured.

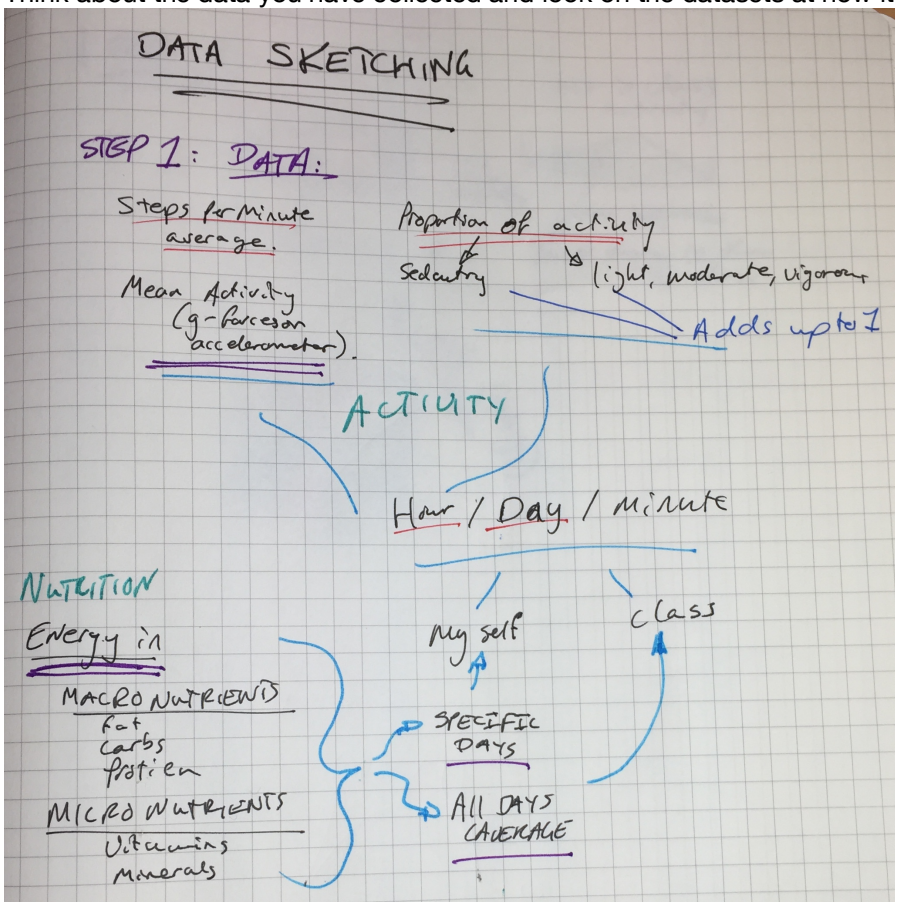


Figure 5 A sketch of data and a simple concept sketch

Activity

- Average steps per minute for each row (so average steps per minute per day, average steps per minute by hour or by minute for each file)
- Mean activity level (in g-forces)
- Proportion of each type of activity: sedentary, light, moderate or vigorous (normalised, which means it is between 0 and 1)
- Proportion of each row that you were sleeping or wearing the tracker

Group data is for every day of the week, broken down by gender. Hours and minutes aren't so useful, as it appears that they are not necessarily aligned to the time of day.
Remember: these are estimates, based on the software that processes your raw data.

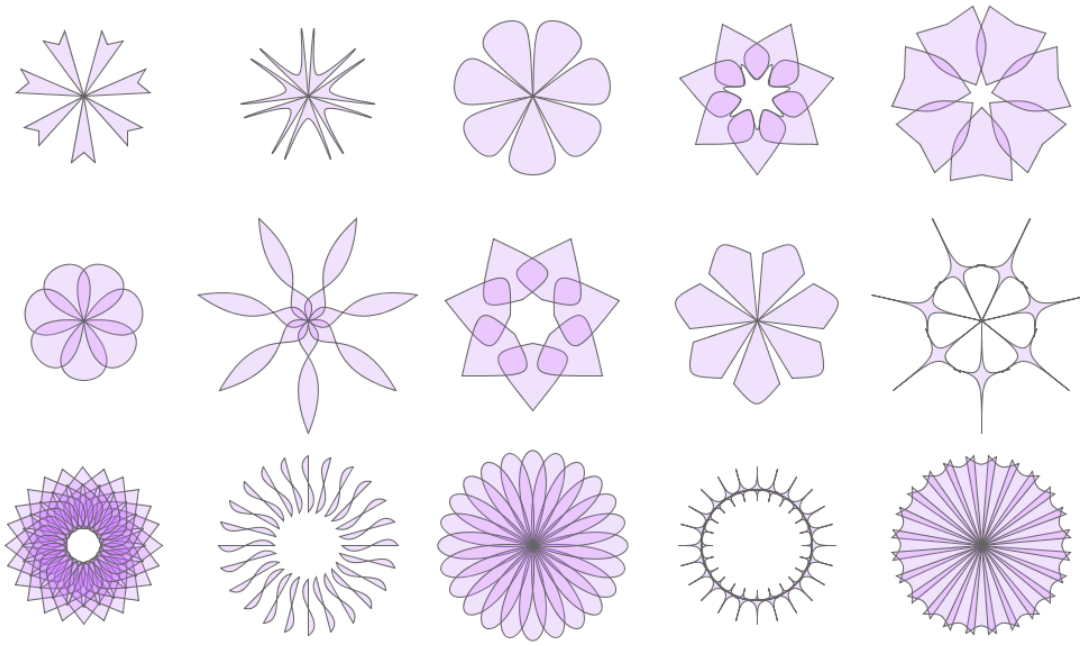
Nutrition

- Amount of energy intake (kCal or kJ)
- Macronutrients (e.g. fats, carbs, protein)
- Micronutrients (e.g. vitamins, minerals)

Group data is for every day of the week, broken down by gender, and your own data will only include days that you did your dietary recall.

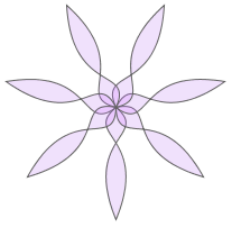
The example we will work through compares hourly levels of activity, as well as the hourly and daily number of steps.

2 Select a starting shape



In the processing code, the shapes are ordered by row, and then left to right. There is a comment in between each row. A single for loop creates each one of the designs above.

Let's start with the one that is second from the left on the middle row.



And come up with a concept.

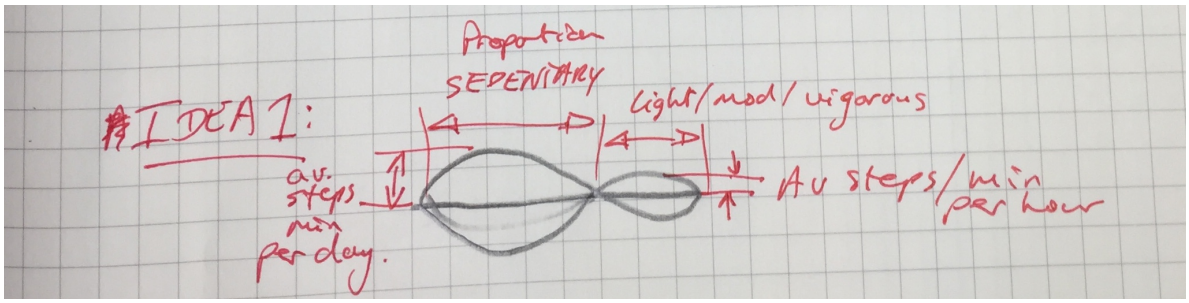


Figure 6 quick sketch of how we can show data

3 Incorporate your ideas into code

Copy and paste the appropriate for loop that creates this into a new processing sketch. There are a few things you will need to change:

Data
You will start by adding data to your sketch. Drag the appropriate file onto (the white text area of) the processing sketch, and load it:

```
Table myData;
myData = loadTable("myDataFile.csv", "header");
float numRows = myData.getRowCount();

size(1000, 300); // pick an appropriate size for your sketch
```

Note: you will have to change the name of your data file to replace myData.csv above.

Also, if you are using something like step count, you will need to calculate the maximum number of steps you had in any row. You have seen this a number of times, so I won't repeat is here again.

Position
We need to make sure this shape draws on the screen of our processing sketch. Most of the changes you will make will be inside the for loop. But you will also

In the for loop, you can change the values for centreX and centreY to display in the middle of the screen. In this example:

```
float centerX = 300.0;
float centerY = 300.0;
```

We are going to use hourly data to display these shapes for each day. So, each day's data needs to be at a new x-position. You can use this code in your own project, as all the data files have a column called day.

```
float numDays = myData.getRow(numRows-1).getInt("Day");
float dayNumber = myData.getRow(i).getInt("Day");
float centerX = map(dayNumber, 0, numDays, 100, width-100);
float centerY = height/2;
```

This will place each day's flower in its own location.

Rotation
In our example we are using the hourly data, so we want to have each 'petal' in our flower spread evenly around the centre. To do this we change

```
float angle = map(i, 0, numRows, 0, TWO_PI);
```

to

```
float angle = map(i, 0, 24, 0, TWO_PI);
```

We use 24 in this line of code because there are 24 hours in a day. You won't need to change this if you are using the daily data. If you want to use minute data, you could use 60 instead of 24. This gives us what we see in Figure 7

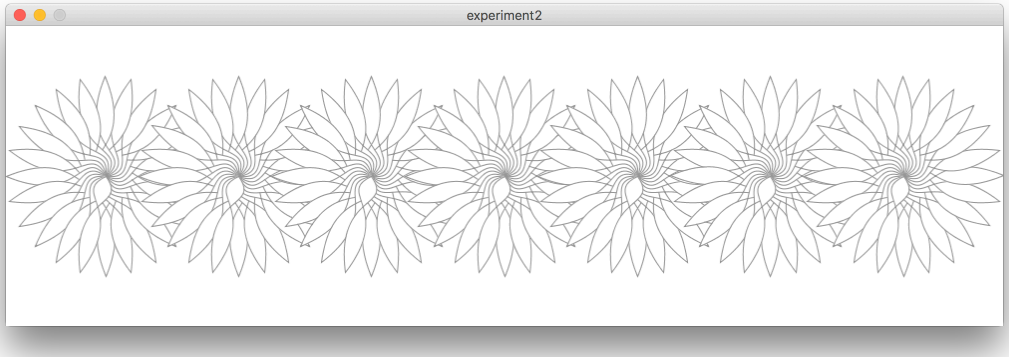


Figure 7 Starting the processing sketch — no data yet!

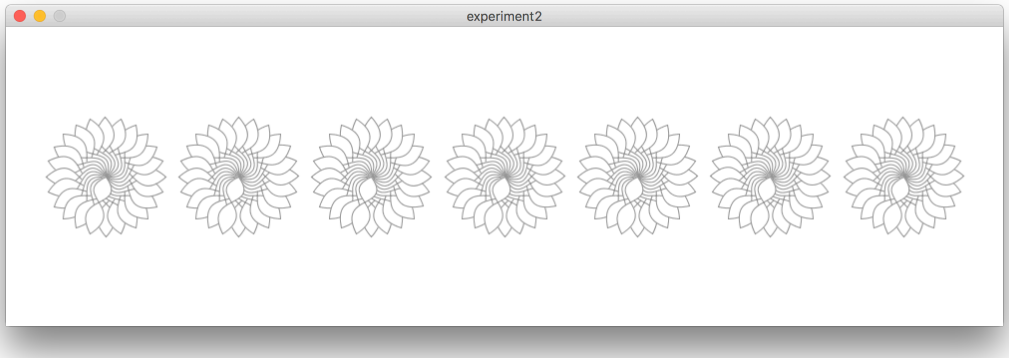
Adapting the shape to data

By altering a few variables we change the overall shape that is drawn.

```
float distanceOffset = 40;
float angleOffset1 = 50;
float angleOffset2 = 50;
float outerRadius = 100;
```

These variables are in the for loop we are using. They are all we will need to change with our data. All of the examples in this template have variables with similar names that you can change. To understand what each one does, change its value (one at a time) to see what effect it has on the final drawing. All of the values you need to change are numbers. You do not need to change any of the vertex, curveVertex or bezierVertex functions.

For example, if we change the `outerRadius` to be 60, we get this change:



Experiment with the values to see what effect they have.

Once you have done this, you can add your data.

Load the data for each interesting column using something like this:

```
float sedentary = myData.getRow(i).getFloat("Sedentary Activity (proportion)");
```

Then map it to an appropriate range:

```
float mappedSedentary = map(sedentary, 0, 1, 10, 40);
```

I chose 10, 40 for the last two numbers in the map function by trial and error.

Then you can replace one of the numerical values with mappedSedentary. You will probably need to do something similar with the other columns in the data set. I used:

```
float distanceOffset = mappedSedentary;
float angleOffset1 = mappedDaySteps;
float angleOffset2 = mappedSteps;
float outerRadius = 65;
```

I had also calculated the maxSteps value and mapped them to using

```
mappedDaySteps = map(steps, 0, maxSteps, 10, 40);
```

Finally, I mapped other activity levels to create colours.

```
float vigorous = hoursData.getRow(i).getFloat("Vigorous Activity (proportion)");
float mappedVigorous = map(vigorous, 0, 1, 180, 255);

float moderate = hoursData.getRow(i).getFloat("Vigorous Activity (proportion)");
float mappedModerate = map(moderate, 0, 1, 120, 255);

float light = hoursData.getRow(i).getFloat("Light Activity (proportion)");
float mappedLight = map(light, 0, 1, 0, 250);

float colorSedentary = map(sedentary, 0, 1, 0, 250);

float redCol = mappedVigorous + mappedLight;
float greenCol = mappedModerate + mappedLight;
float blueCol = colorSedentary;
float opacity = 50;

fill(redCol, greenCol, blueCol, opacity);
```

I spent a fair amount of time to get the values of these colours right, and create this image:

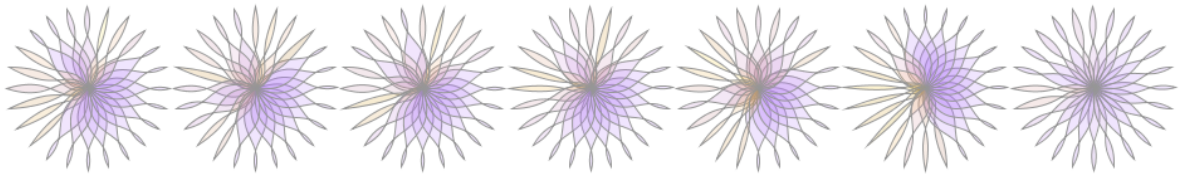


Figure 8 The final design

I added one line after the end of the for loop to create a .png file:

```
save("output.png");
```

If you are using the setup and draw functions, you can save your file by adding this code to the very end of your sketch (after the end of the } that closes the draw function) and clicking the mouse on the image processing produces:

```
void mouseClicked() {save("output.png");}
```

Either of these lines will create a file, output.png, in the same folder as your sketch file. You can change the name of the file, if you want to make multiple copies.

Final Assessment

Processing PDF

Marking Rubric
Include the **Image** you created in Processing, a brief (2-3 sentence) description of the **Research Question** you have chosen and a short Justification of the choices you have made to create the image. These should outline the columns of data that was used, why this data was chosen and how you mapped the data to visual elements, such as colour, position, orientation, scale, shape and size.

The PDF will ideally be a single page.

Final Plan and Video

Marking Rubric
You need to submit your revised video plan using the **blank video template document**, which includes the justification of the topic/audience/style (as you had in the one submitted for the peer review exercise.

You will also need to:

- Upload your video as an .mp4 to Canvas. This is done in Adobe Spark by selecting the download icon along the top of the page.
- Add a URL that links to your video. In Adobe Spark, this can be done by clicking the *share* button along the top, selecting a category (choose *education* if you are unsure) and selecting *Create Link*.

Due Dates

Both of these are due on Midnight, on the Friday after the week 9/10 living data workshop:

- Stream A due 23:59, Friday 18th May
- Stream B due 23:59, Friday 25th May

Research

In the next few weeks there will be some emails to invite you to participate in two research projects. Neither of these are compulsory, and they will not affect your marks in this course in any way. If you do choose to participate, you will be able to do so anonymously.

1 The Living Data Project: Creative code and video in biology education

This project includes two parts:

1. Online survey on Canvas, available to all students, which asks questions about the Living data class activities.
2. Focus groups, available to all students, but will be limited to a small number of students. We will aim for equal representation between male/female, with those who express interest first having priority.

You will receive an email through Canvas about participating in this study in a few weeks, that has additional information about our research project.

2 The Living Data Project: Open Dataset

Our goal for this project was to start an open dataset that will evolve and grow over future iterations of this course, which will be able to be used for scientific research. We will send you an email with more information about how you can participate in this project in the coming weeks.

Thanks for a great semester and good luck with your assessments

Figure 9 from one of my favourite comics
https://www.smbc-comics.com/comic/to-be-or-not-to-be