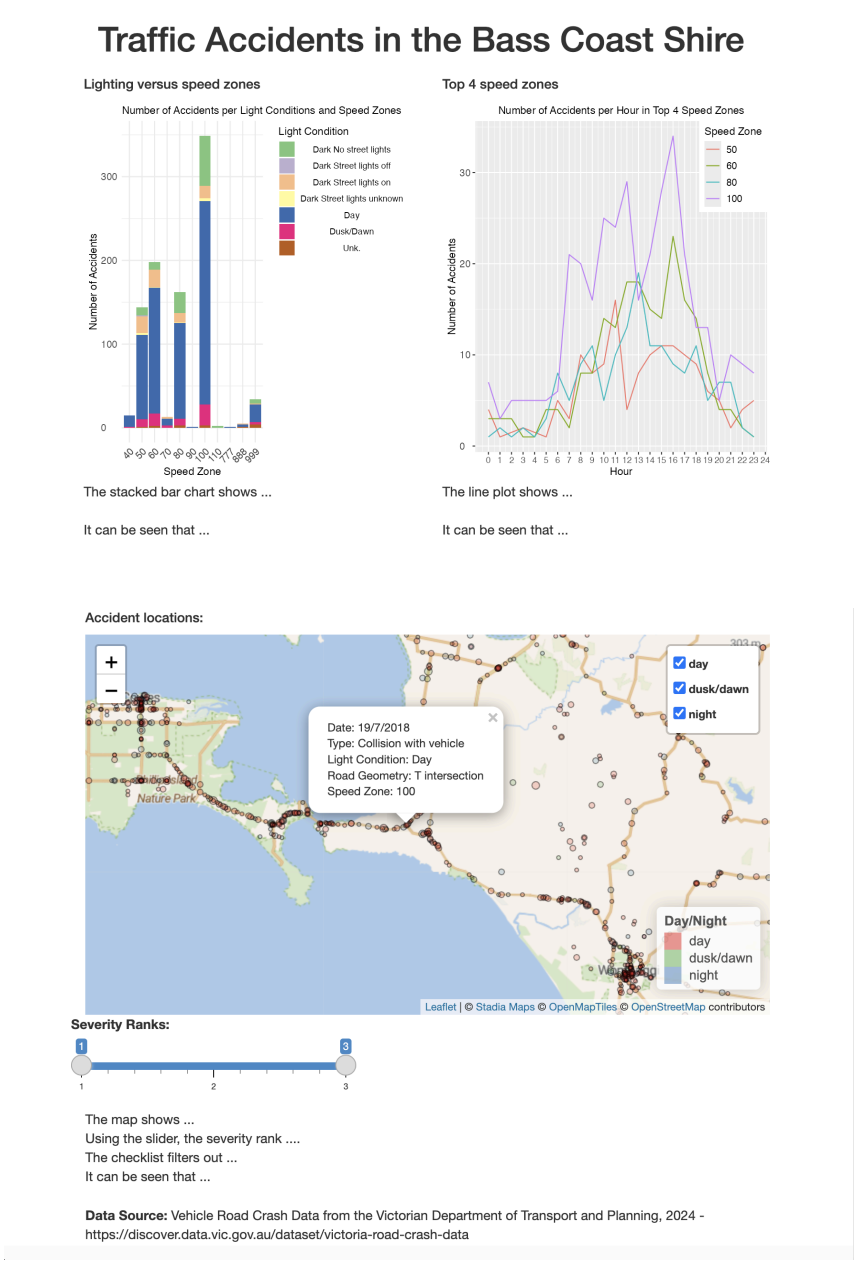
Detailed Class Feedback on PE2

Like we did for PE1, we would like to give you some general feedback about what you should have done in Programming Exercise 2 (PE2). Once again, there are a number of plausible responses to the tasks, but this might help you better understand what you might have missed and better understand the feedback you received. As always, talk to your marker if you need further guidance.

This assignment was about creating visual elements using R and R Shiny. You needed to make two static elements (VIS1 and VIS2), a proportional map and various interactive components. You also had various text boxes you needed to fill out and position. Each had requirements, so marks were given based on whether you met those requirements well, partially or not at all.

Overall, your work should have ended up looking something like this.



Data Cleaning & Wrangling:
You were told you didn't need to clean the data. Therefore you should have left the data unchanged and none of the data should have been omitted. This included the speed zone categories and the years.

However, you did need to do some wrangling of the data. For instance, you needed to identify the hours, create the daynight column, and identify the top 4 speed zone categories. These only needed to be done once, not every time you remade the visualisation or processed the user input in the server code. Consequently, they were best separated from the server and the UI code, or at least not included in any server processing code. None of this should have had values hardcoded (it was okay to hardcode 4 for the top 4 as that was part of the specification for the design). Even the minimum and maximum severity rank values (which were needed if you had an interactive slider) could be calculated once outside of the server and the UI code.

The data for VIS 2 could have also been divided up outside of the server code. It is a static visualisation. You only need to divide up the data once for the top 4 speed zone categories.

VIS 1:
For VIS1, you were asked to visualise the number of accidents per speed zone category and lighting condition. You weren't restricted to how you visualise this but a stacked bar chart was suggested as an option. The crucial thing was that VIS2 was based around the top 4 speed zone categories that were shown in VIS1. If you put the light condition descriptions on the x-axis for VIS1 and the speed zones in the stacks, you couldn't easily see which were the top four. It wasn't a suitable stacked bar chart. You needed to have the speed zone on the x-axis and the light conditions on the stack so you could see the top 4 on VIS1. Other types of visualisation may have been also suitable, but were rarely used. Some students divided the top 4 speed zones from the rest using a numerical filter (e.g., > 100 accidents), but while that might have looked good, it was relying on a hardcoded value (i.e., 100), instead of calculating the top 4.

VIS 2:
VIS 2 was about the number of accidents per hour in the top 4 speed zone categories - 050, 060, 080, 100. These had to be calculated, not hardcoded. The choice of visualisation was left to you, but the best were multiple line graphs, either on the same graph or as separate graphs per speed zone category. A stacked bar chart or cumulative area graph was not ideal because it was hard to compare the rise and fall of the values. It could only really show the peaks within each category, not clearly when one category had more accidents than the other.

Contextual information on VIS1 and VIS2:
VIS1 and VIS2 also had to provide additional contextual information, like a title about what they were showing, and sensible labels and values for any axes. This had to include a legend. If you didn't label each hour of the day, or use the descriptions of the light conditions then your visual elements would be hard to understand.

Colour usage on VIS1 and VIS2:
Categorical data is most effective to show using colour hue but some of you inappropriately chose colour gradients or used colours (like yellow) that were too faded that they were almost impossible to see. Ideally, if you used colour for VIS1 and VIS2 to represent the same thing (i.e., speed zone categories), there should be consistency for which colour is used for which value[1].

Creating the map:
The map needed to have a circle for every location in which an accident occurred. Each circle needed to be coloured according to the daynight value. Ideally, the daynight values should have been already determined, rather than having to process them everytime the map was updated. The assignment specified that those values had to be 'day', 'dusk/dawn', and 'night', so 'other' or 'dark' or 'unknown' were not acceptable.

You were told to set the radii of the circles according to the reverse of the severity rank values. This means that accidents with a severity rank of 1 should have the largest circles and those with rank 3 should be the smallest. The simplest formula was simply the inverse relationship (i.e., 1/SEVERITY_RANK), but there were many other suitable formulas. You could also have scaled the ranges of sizes up or down to suit your map.

The issue for both these map elements was how easily they could be seen, especially due to the frequently overlapping circles. Some students' maps had two circle sizes that were distinct and obviously different from each other but the third size was too similar. Likewise, some of the colour choices were either too similar or the border of the circles wasn't easy to see against the map tiles or other circles. Some of you tried lowering the opacity of the circles, but this then resulted in there being multiple gradients of colour on the map, making it even harder to work out where each overlapping circle was located. It wasn't an easy task, but one of the clearest designs was to use black edges for the circles and then fill in the middle with the three colours, with some opacity.

The base map tile styling was a small issue for some students. To understand the data on the maps, you needed to understand where each accident occurred. A number of students identified that certain types of accidents occurred in the towns or on major roads. For this interpretation to be possible you needed to have a meaningful collection of map tiles, styled in a way that was readable and useful for your exploration. Unfortunately, some students chose base map styles with faint roads and labels, making it hard to see and compare this contextual information.

Interacting with the map:
The map also had various interactive elements. You had the choice of which two of the three to implement and if you did all three, your mark for these was determined by the best two. These required you to get your server and ui code to communicate and be aware of how to respond to changes.

If you had tooltips, when you moved the mouse over a plot, it should have shown a popup tooltip providing certain information about the accident at that location. When you move off the plot or select a new circle, it should hide the old tooltip and display a new one. Of course,

[1] These choices also relate to Grice's Maxim of Manner (which will be covered in the coming weeks) by keeping things simple and not complicated.

if there were overlapping data plotted, then it might be hard to select some of the data. This is where the interactive zoom facility of the map can be used. This is part of the map by default, so you didn't need to do anything to include it, but it is still an interactive element. While you could have required the user to click on the mouse when above a circle to get the tooltip, this is not what the assignment requested.The description asked for the tooltip to appear when focus is given to the data point, i.e., clicking is not required.

You could also provide a slider to help filter which accidents were visible on the map. The slider set the range of severity rank values allowed for the visible accidents. Your code had to change the accidents plotted on the map to only be those that correspond to the ranges on the slider. As you moved the slider, the accidents on the map should have changed. By default, the map should start with the maximum range possible on the slider.

The third interactive element was the checklist. This allowed the user to restrict the accidents on the map to those that occurred in the daynight values that were selected. By default, all values were selected, but it should have been possible to unselect all the values, meaning that there are no accidents plotted on the map.

A regular problem with both the slider and the checklist was that it was overriding the user's zoom setting. If a user zoomed in on a particular part of the map, say the bridge across to Phillip Island, then interacted with the slider or checklist, some maps would reset to the default zoom and location. This is not very user friendly and is a poor design. The default map settings should have only been used for the initial map creation.

Textual descriptions
You also had to write some text about the data and the visual elements. This text had to identify the visual elements by type (like calling it a stacked bar chart), explain what it was showing and how to use it (for the map), and interpret what it showed. This is all part of the narrative of the entire visualisation you created. It helps the viewer understand what they are seeing and get the message you want about the data and the topic. Similarly, it was useless to use the term 'VIS1' if you never labelled anything as VIS1. You can't expect the audience to know.

Citing the data source
You were asked to provide information about the data source. At a bare minimum, this should have been a link to the government source for the data. What you needed to provide was contextual information so the user is clear about what data is represented on the visualisation, like the name of the dataset and the publication/access data. The name of the datafile is not enough.

Layout of the visualisation
Of course, all of these components had to be readable and in the correct positions, given the layout that was in the assignment description. The slider/s and checklist could be placed on the map or just outside of the map, but it shouldn't have been anywhere else. There was not a strict size of any of the components but we expected them to be in positions next to each other in the way stipulated in the layout. You also needed to be careful when labelling visual elements. Not only do you need to include titles and label the axes, if the text wasn't dark enough and large enough to be easily read, then it is meaningless. This is relatively easy to fix in R using layers in ggplot().

In closing

These were a lot of little things that you needed to consider for what might look like a simple visualisation, but they are all required in order to make it an effective visualisation. This task will have expanded your understanding of how to use the visualisation skills you developed in the activities in the applied sessions and workshops and exercise them in a more real world situation. We hope to see them also be used in your DVP, whether using R and R Shiny or D3 Javascript.

Thanks for all your submissions and insights,

Michael, and Sarah, on behalf of the entire FIT5147 teaching team