**Video Script: Section 1 Video 4 – Understanding some subtleties with aesthetics**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Description | Action on screen | Narration |
| 1 | Introduction  (Outcome and why it is desirable)  This should give the viewer an idea of the outcome of the task at the beginning of the videos and set the stage and expectations of the viewer. | Opening slides. | **In this video**, we are going to see how aesthetics work in more details.  In particular, we’re going to see that 1) aesthetics can be defined in the first call to ggplot and 2) aesthetics can be set to a constant value, rather than mapped from the data. |
| 2 | Context(Problem/Solution)  Present the viewer with a real-world solution and how the situation would pose as a challenge. It always helps to draw the viewer's attention using a use-case. Metadata template can be used here. |  | A typical error when using learning ggplot2 is to mix the concepts of mapping and setting aesthetics. We clarify these two concepts with a simple example in this video. |
| 3 | Guidance (How to do it and how it works): | Highlight and run:  # example 01  ggplot(small, aes(x = carat, y = price, colour = cut)) + geom\_point() + ggtitle('example 03')  A colourful plot appears. | Open the activity01\_04.R and run the first 4 lines to prepare some data.  Then run example 01. |
| 4 | Sharing aesthetics | Highlight aes(x = carat, y = price, colour = cut) | The aesthetics x, y and colour are defined in the call to ggplot(), so you don't need to specify them in geom\_point() since they're already available. |
| 5 |  | Highlight and run:  # example 02  ggplot(small, aes(x = carat, y = price)) + geom\_point(aes(colour = cut)) + ggtitle('example 04')  The same plot appears. | Now run example 02. In example 02, only the aesthetics x and y are defined in ggplot(). But note that the aesthetic colour is defined in geom\_point(). But the plots generated by Examples 01 and 02 are identical.  In all cases, the resulting graphs are identical. This is useful for sharing aesthetics among geom.  You can think of it as defining global variables in the call to ggplot, which can then be accessed by all geoms. |
| 6 |  | Highlight and run:  # example 03  ggplot(small, aes(x = carat, y = price)) + geom\_point(colour = 'blue') + ggtitle('example 03') | Rather than having a map from the data to graphical properties, we can also set graphical properties to certain value. |
| 7 |  | Highlight colour = 'blue' | To do this, we simply set them outside the call to aes().  Run example 03. The colour is now defined outside the aesthetics, to be a constant value. |
| 8 |  |  | What would have happened if you had defined the colour within aesthetics? |
| 9 |  | Highlight and run:  # example 04  ggplot(small, aes(x = carat, y = price)) + geom\_point(aes(colour = 'blue')) + ggtitle('example 04')  01_04_example04.png | Run example 04.  As you can see, this is not what we wanted: the points are colored in red, rather than the expected blue. |
| 10 |  |  | What happened is that because the colour was defined to a constant value within aes(), ggplot understood this as a map from data to the aesthetics. But in this case, the data consisted of only one value (the string 'blue') and therefore was translated to the first colour of the default colour scheme: red.  If you see this kind of unexpected plot, you now know where the error comes from. |
| 11 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 | Conclusion:The video concludes by showing the viewer that the goal has been achieved, and reminding them why they should be happy about that. A PowerPoint summary slide with the key points emphasized would make it easier for the viewer to remember what was covered in the video | Last slide of the PPT | We've seen that when you define aesthetics in the call to ggplot, they are shared by all geoms in the plot. You can also define aesthetics for individual geoms.  When aesthetics are defined inside the aes function, ggplot assumes that they refer to mapped variables. If they are defined directly in ggplot or the geom, then they are assumed to be constant.  In the next video, we will see how to use ggplot in scripts, rather than in the command line. |