**Video Script: Section 3 Video 3 Minimizing overplotting with jitter**

|  |  |  |  |
| --- | --- | --- | --- |
| 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. | Refer to ppt | In this video,you’ll get the solution for plotting many data points that could mask each other. |
| 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. |  | With a large dataset or with many data points having the same value, the resulting graph can be misleading because it doesn’t convey the true number of data points plotted: graphical elements will hide or mask others. This is called ‘overplotting’. |
| 3 | Guidance (How to do it and how it works): |  | We’re going to see how to slightly jitter points to minimise overplotting. |
| 4 |  | Highlight and run:  library(ggplot2)  nrow(mpg)  # over plotting  ggplot(mpg, aes( x = cty, y = hwy)) +  geom\_point( aes(colour = manufacturer), size = 3) +  ggtitle("No jitter: some points are covering others")  On the screen:  [1] 234  And the plot:  A description...A description... | Open activity\_03\_03.R in Rstudio.  Run the first 9 lines of the code.  Notice that with nrow(mpg), we’re expecting 234 data points to appear on the graph. |
| 5 |  |  | We plotted the consumption in city and on highway for each of the 234 cars in the dataset. |
| 6 |  |  | How many points do you see on the plot? Certainly fewer than 234.  This is because the reported consumptions have a limited number of values, so cars having the same value for both city and highway end up on top of each other. |
| 7 |  |  | To minimize this, we can use the parameter ‘position’ with the function ‘position\_jitter’ to randomly jitter the points . Let’s see how this is done. |
| 8 |  | Highight and run:  ggplot(mpg, aes( x = cty, y = hwy)) +  geom\_point( aes(colour = manufacturer), position=position\_jitter(width=0.1), size = 3) +  ggtitle("After jittering. The grpahs conveys the actual number of points better.")  A description...A description... | Run the second ggplot command. With position\_jitter(width=0.1),), we randomly move each points by a small amount horizontally only (‘width’)  The amount of jittering should be moderate, otherwise the position of the points will be so different from their original values that the graph will be misleading.  You can also use the aesthetics ‘alpha’ to make the points slightly transparent so that they cannot hide each other. Set it to a number between 0 and 1, with 0 being completely transparent and 1 being completely opaque. |
| 9 |  |  | We can now see most of the 234 points and have a better sense of how many cars are represented. |
| 10 |  |  |  |
| 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 | Back to the PPT | We saw how to use ‘position\_jitter’ to avoid overplotting by randomly moving the data, but without producing a misleading graph.  In the next video, we’ll see how to easily generate plots for subsets of the data. |