**Video Script: Section 4 Video 5 big data with bigvis: smoothing and peeling**

|  |  |  |  |
| --- | --- | --- | --- |
| 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, we are going to look at two other important features of BigVis: smoothing plots and peeling 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 very large dataset is bound to contain extreme values which can lead to a distorted and misleading plot. |
| 3 | Guidance (How to do it and how it works): |  | We are going to see how to automatically remove outliers and how to smooth the original plot. |
| 4 |  | Highlight and run.  library(ggplot2)  library(bigvis)  # bin and condense the whole dataset  myMoviesSummarised<- with(movies, condense(bin(length), bin(rating))) | Open activity\_04\_05.R and run the first 3 commands.  We’ll be using the movies dataset again. We bin and condense the data as before but note how we’ve not manually removed extreme values this time. |
| 5 |  | Highlight and run:  ggplot( data = myMoviesSummarised ) +  geom\_tile(aes( x = length, y = rating, fill = .count)) +  ggtitle("Whole data set summarised")  A description... | Let’s plot the summarized data by running this code. |
| 6 |  |  | Because of a few movies with extreme lengths (more than 300 minutes), the plot is squashed on the left handside and is difficult to read. |
| 7 |  | Highlight and run:  myMoviesSummarised<- peel(myMoviesSummarised)  ggplot( data = myMoviesSummarised ) +  geom\_tile(aes( x = length, y = rating, fill = .count)) +  ggtitle("Whole data set summarised and peeled")  A description... | Datapoints with very rare values are usually considered outliers.  The function peel() removes regions with very low density (I.e very isolated points, that are therefore likely to be outliers),  By default, peel will keep 99% of the data. You can control this cutoff point with the parameter keep.  The advantage of using peel over a manual selection is that the cut-off point is chosen in the context of the whole dataset.  Run the second portion of code.  The plot is now more informative. |
| 8 |  | Highlight and run:  myMoviesSummarised<- smooth(myMoviesSummarised, h = c(10,1))  ggplot( data = myMoviesSummarised ) +  geom\_tile(aes( x = length, y = rating, fill = .count)) +  ggtitle("Whole data set summarised, peeled and smoothed")  A description... | Next we apply smooth() to decrease the ruggedness of the original plot.  In the same manner we used geom.\_smooth in a previous video, we want ignore noise to find trends.  The parameter h controls the size of the kernel for each dimension, i.e. the amount of smoothing applied.  Recall that .count is a column of myMoviesSummarised, created during the condensing process. |
| 9 |  |  | We now have a nicer looking plot which is easier to read because unimportant local differences have been ironed out. |
| 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 PPT | We’ve used two functions from the package BigVis to automatically remove outliers and smooth the original plot.  In the next section, we’ll see how to customize our plots. |