**Video Script: Section 4 Video 2 user-defined models**

|  |  |  |  |
| --- | --- | --- | --- |
| 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’ll carry on investigating different ways of uncovering trends in our data: polynomial (non-loess) regression and splines. |
| 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. |  | The data used here has to be sufficiently complicated such that no obvious model comes to mind. However, it is still helpful for the plot to show trends in the data |
| 3 | Guidance (How to do it and how it works): |  | We’ll use two ways of summarizing the data in order to uncover possible trends in the data. |
| 4 |  |  | Open activity\_04\_02.R  We’ll use the chickweight dataset again. |
| 5 |  | Highlight and run:  library(ggplot2)  # polynomial fit  ggplot(ChickWeight, aes(x=Time, y=weight, colour=Diet)) +  geom\_smooth(alpha=.2, size=1, method="lm", formula = y~poly(x,2)) +  ggtitle("Using a polynomial")  04_02_polynomial.png | Run the first ggplot command. |
| 6 |  |  | By using the formula y ~ poly(x,2), we are fitting a polynomial model of degree 2.  The higher the degree of the polynomial, the wigglier the line can be. |
| 7 |  |  | Another way to smooth the data and uncover possible trends is to use splines.  Splines connect datapoints piecewise as smoothly as possible. They can accommodate very complex shapes. |
| 8 |  | Highlight and run:  # spline fit  library(splines)  ggplot(ChickWeight, aes(x=Time, y=weight, colour=Diet)) +  geom\_smooth(alpha=.2, size=1, method="lm", formula = y~ns(x,df=3)) +  ggtitle("Using splines")  04_02_splines.png | Run the next ggplot command to see the effect of splines. |
| 9 |  |  | The formula is y ~ ns(x,df=3). The function ns comes from the package ‘splines’ and will produce the splines. We are not limited by the range of functions offered by ggplot2. The parameter df (degrees of freedom) controls how wiggly the curve is. |
| 10 | Note: the code has some comments for each of the model. Again, this is not to be explained in the video but for the user to study in their own time. | Highlight and run:  # more examples of methods  library(splines)  ggplot(mtcars, aes(x = qsec, y = wt)) +  geom\_point() +  geom\_smooth(se=FALSE, size = 1.2, colour="black", method = loess) +  geom\_smooth(se=FALSE, size = 1.2, colour="red", method = "lm") +  geom\_smooth(se=FALSE, size = 1.2, colour="green", method = "lm", formula = y ~ ns(x,3)) +  geom\_smooth(se=FALSE, size = 1.2, colour="blue", method = "lm", formula = y ~ poly(x,2)) +  geom\_smooth(se=FALSE, size = 1.2, colour="yellow", method = "lm", formula = y ~ log(x))  04_01_many.png | More methods are available, run the last ggplot command for a few extra examples.  The datapoints in the graph are the black dots.  We use se=FALSE to hide the gray standard error ribbon and plot the trend only. |
| 11 |  |  | Have a closer look at them in your own time and experiment with your own functions. |
| 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 |  | We saw alternatives for graphically summarizing the data, using any model of your choice.  In the next video, we’ll learn how to plot any user-defined function. |