Business Statistics - Assignment 4

The Motorcycle Accelleration Data (available in csv format) contains a series of measurements of head acceleration in a simulated motorcycle accident, used to test crash helmets. The file has the following columns:

- times: The time in milliseconds since impact.
- accel: The recorded head acceleration (in g).
- strata: A numeric column indicating to which of the three strata (numbered 1, 2 and 3) the observations belong.
- v: An estimate of the residual variance for the observation. v is constant within the strata but a different estimate is used for each of the three strata.

The source of the data is Silverman, B.W. (1985) Some aspects of the spline smoothing approach to non-parametric curve fitting, *Journal of the Royal Statistical Society*, B, 47, 1-52.

The objective is to model the relationship between `accel)`, your output variable Y, and `times`, your X variable: $Y = f(X) + \epsilon$, where the function $f(\cdot)$ is approximated by

1. a polynomial of degree p: $f(X) = \beta_0 + \beta_1 X + \cdots + \beta_p X^p$,
2. a natural cubic spline,
3. a smoothing cubic spline.

You should address the following points:

a. Select the degree of the global polynomial p and comment on the goodness of fit.

b. Select the number of knots of the natural cubic spline (using an information criterion, such as AIC), when the knots are automatically located by the R function. Try using up to 10 knots, $K = 1, 2, \ldots, 10$. A `for` cycle could be useful for this task.

c. What knots you would have considered from graphical inspection of the scatterplot of Y versus X_1 (notice that the function `ns( )` allows you to input the x-values at which the user defined knots are located).

d. For the smoothing cubic spline discuss the estimated value for the smoothing parameter and compare the fit with that arising from your preferred natural cubic spline fit.

Please upload your report at https://www.dropbox.com/request/NQ1xhCglVGpHlVt0NbxM before 10:00 PM on 24/10/2019.

The following R functions will be needed:

```
lm()                # fit a linear regression model
ns()                # natural spline basis - library(gam)
poly()              # global polynomial basis
smooth.spline()     # fits a smoothing cubic spline
extractAIC()        # Computes the AIC or BIC for a fitted model
```