SPLINES

Oluwafunmibi, Omotayo Fasanya

October 11, 2024

Outline

- Introduction
- Motivation Dataset
- From Linear to Non-Linear Methods
- Polynomial Models
    - Polynomial Fitting Issues
- Piecewise Polynomials
    - Basis Functions
    - Choosing Basis Functions
- Splines
    - Quadratic Spline
    - Cubic Spline
    - Natural Spline

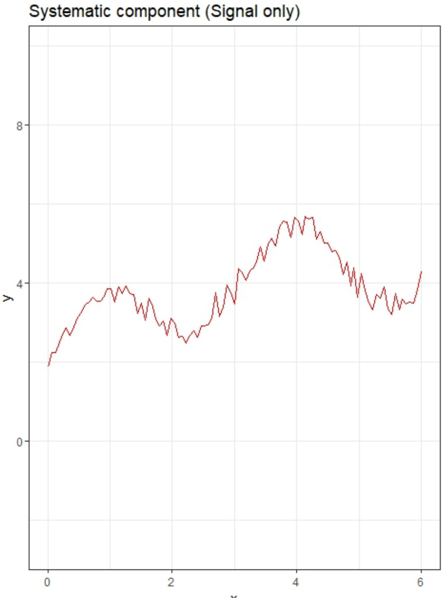
Introduction

- Most statistical modeling and machine learning tasks focus on the equation:

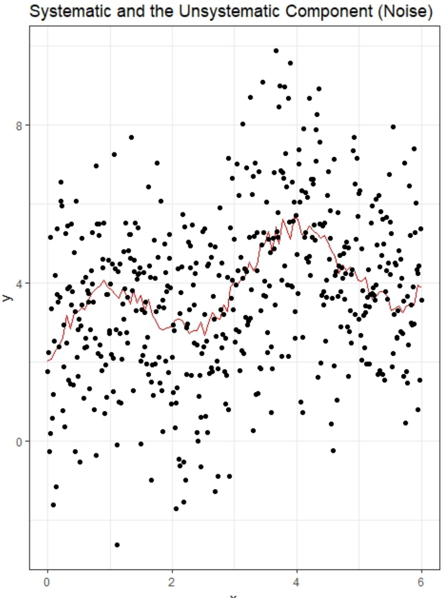
$$Y = f(X) + e$$

- Where:
  - Y is the outcome variable of interest.
  - X are predictor variables.
  - $f(X)$ is the systematic component of the model (the signal).
  - e is the unsystematic error component (the noise).
- Our main task is to separate the signal from the noise, as $f(X)$ is unknown and must be estimated.
- Before fitting, let's visualize these concepts!

Introduction



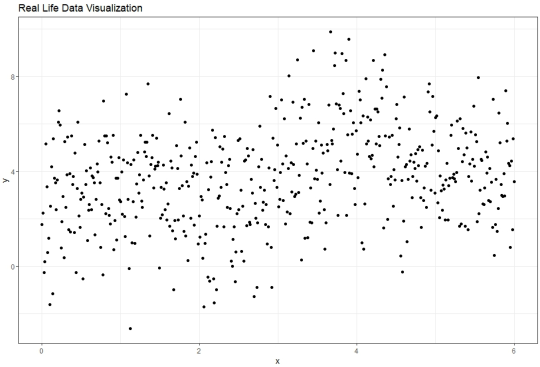
Systematic component (Signal only)



Systematic and the Unsystematic Component (Noise)

Introduction

- In real life, we only have the plot below, and it's up to us to estimate and fit the red curve (systematic component) to separate the signal from the noise.



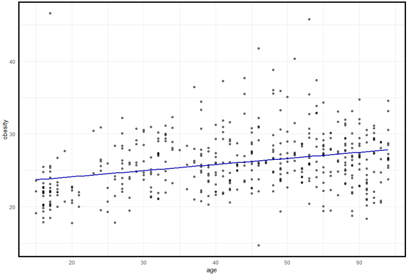
Motivation Dataset

- The data used is a retrospective sample of males in a heart-disease high-risk region of the Western Cape, South Africa.
- These data are taken from a larger dataset, described in Rousseauw et al, 1983, South African Medical Journal.
- There are 462 observations with 10 variables. The following two variables was used for explaining Splines
  - Age:age at onset
  - Obesity: a numeric vector

From Linear to Non-Linear Methods

- In Linear Methods, we always look at the relationships between two variables as a straight line, i.e., if you increase the predictor by 1 unit, you expect to see an X unit increase.

- However, this is mostly not the case in real-world data. Not all real-world data have a linear relationship.

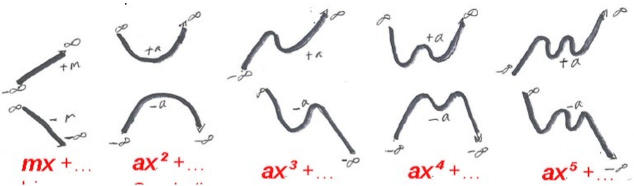


Polynomial Models

- A common alternative to linear models is the polynomial model.
- A polynomial model of degree d can be expressed as:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \cdots + \beta_d X^d$$

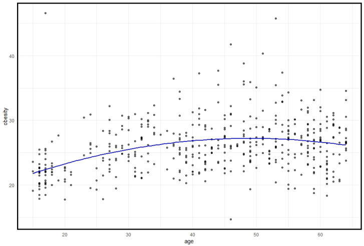
- Here:
  - X is a feature (assuming one feature in this example).
  - $\beta_0, \beta_1, \ldots, \beta_d$ represent the coefficients associated with the powers of X, indicating changes in the outcome.



Polynomial Models

- Even though the model uses polynomial terms which makes it non-linear, it remains linear in the parameters

- By expanding the feature space, we can fit a line in the transformed space, resulting in a curve when mapped back to the original data.

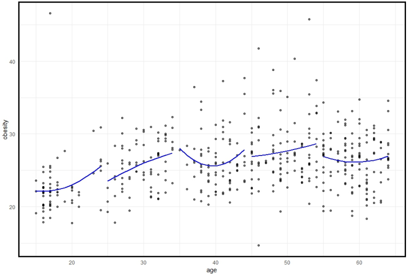


Polynomial Fitting Issues

- One issue with polynomials is their global behavior.
- A polynomial assumes a certain relationship across the entire dataset, which can lead to:
    - High wiggle at the beginning and end of the data, giving bad interpolation or extrapolation results.
    - Overfitting, especially with higher-degree polynomials and
    - For higher-degree polynomials, the model becomes more sensitive to the removal of existing points or the addition of new data
- We want something smoother and more reliable for curve fitting. This leads us to **Piecewise Polynomial and splines**.

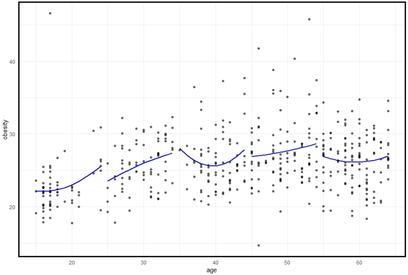
Piecewise Polynomials

- Instead of using one polynomial across the entire range, we can fit lower-degree polynomials over sub-regions.
- **Knots**: Points where different polynomials meet and share common values to ensure smooth transitions.



- If we have a knots located at points a (X = 25,35,45,55) and a piecewise polynomial is fit, the most basic structure we could have is shown above

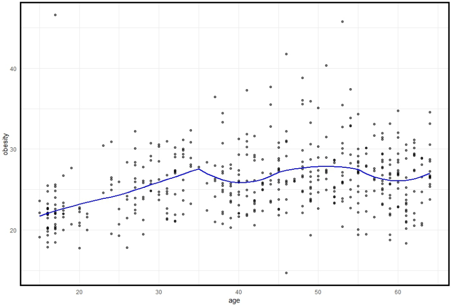
Piecewise Polynomials Equation



$$
\begin{aligned}
Y &= \beta_{01} + \beta_{11}X + \beta_{21}X^2 && \text{for} \quad X < 25 \\
Y &= \beta_{02} + \beta_{12}X + \beta_{22}X^2 && \text{for} \quad 25 \leq X < 35 \\
Y &= \beta_{03} + \beta_{13}X + \beta_{23}X^2 && \text{for} \quad 35 \leq X < 45 \\
Y &= \beta_{04} + \beta_{14}X + \beta_{24}X^2 && \text{for} \quad 45 \leq X < 55 \\
Y &= \beta_{05} + \beta_{15}X + \beta_{25}X^2 && \text{for} \quad X \geq 55
\end{aligned}
$$

We can impose an additional condition of continuity, ensuring that the segments are smoothly connected without any gaps or breaks.

Piecewise Polynomials



$$
\begin{aligned}
Y = &\beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 (X-25)_+ + \beta_4 (X-25)_+^2 \\
&+ \beta_5 (X-35)_+ + \beta_6 (X-35)_+^2 + \beta_7 (X-45)_+ \\
&+ \beta_8 (X-45)_+^2 + \beta_9 (X-55)_+ + \beta_{10} (X-55)_+^2
\end{aligned}
$$

Where:

$$
(X-k)_+ = \begin{cases} 0 & \text{if } X < k \\ (X-k) & \text{if } X \geq k \end{cases}
$$

Basis Functions

The concept of basis expansion involves replacing the original features in X with the transformed versions of X.

The general equation for basis expansion is:

$$f(X) = \sum_{i=1}^{p} \beta_i B_i(X_i)$$

$$\begin{aligned} y =& \beta_0 \cdot 1 + \beta_1 b_1(x) + \beta_2 b_2(x) + \beta_3 b_3(x) + \cdots + \beta_{10} b_{10}(x) + \epsilon \\ =& \beta_0 \cdot 1 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x - 25)_+ + \cdots + \beta_{10} (x - 55)_+^2 + \epsilon \end{aligned}$$

Here, $B_i(X_i)$ represents the basis functions and β_i are the coefficients.

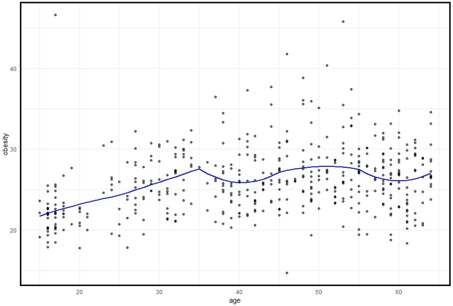
A linear combination of the basis functions creates the model's fit that we observe.

Choosing Basis Functions

We have the flexibility to choose arbitrary basis functions. These choices can be made based on domain knowledge, insights gained from exploratory data analysis, or even through trial and error.

- We can use **polynomials**, which leads to polynomial regression.
- We can use other functions such as powers, logarithms, or square roots.
- Another option is the indicator functions $I(a_i \leq x_k < b_j)$ to divide the original predictor into non-overlapping subsets.
    - Just like we previously did. Within each subset, we fit a polynomial locally. This approach leads to fitting piecewise polynomials.

Piecewise Polynomials



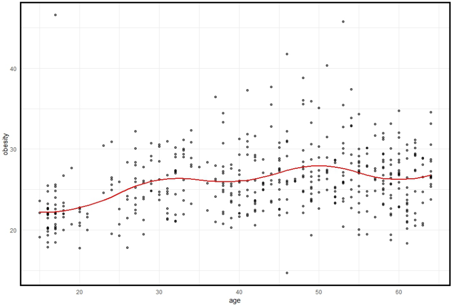
$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 (X - 25)_+ + \cdots + \beta_9 (X - 55)_+ + \beta_{10} (X - 55)_+^2$$

- We notice that there is a sharp change at the Knots which makes our plot rough.
- Also the function is *not differentiable* at the knots, i.e., discontinuities are observed at these knots.
- So how do we fit a smooth curve?

SPLINE

- Just like we force the piecewise polynomial to be connected, we also want to force this to be continuous.
- To achieve this, we drop all the lower-order polynomials in our model.

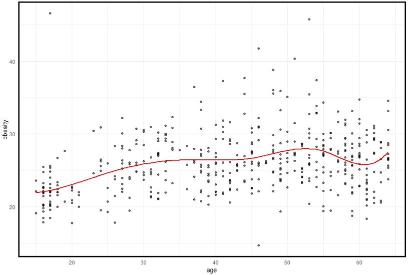
$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3(X-25)_+^2 + \beta_6(X-35)_+^2 + \beta_8(X-45)_+^2 + \beta_{10}(X-55)_+^2$$



- The plot above is called a **Quadratic Spline**

Cubic Splines

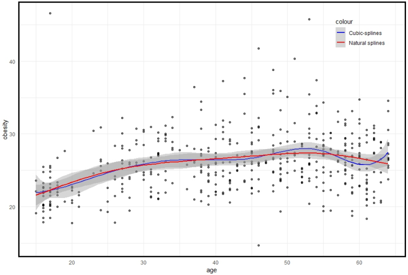
- **Cubic splines** are piecewise polynomials of degree 3, ensuring smoothness at the knots.
- They maintain continuous first and second derivatives, making them useful for modeling non-linear relationships with smooth transitions.



Note: It is possible to use higher-degree polynomial basis functions if needed and Splines can be extended to include more knots for finer control over local behavior, while maintaining global smoothness.

Natural Splines

- **Splines** can behave unpredictably (high variance) at the boundaries of the data (the tail) expecially for small sample size
- Natural splines address this by constraining behavior at each boundary region
    - The spline function is constrained to be nearly linear when X is less than the smallest knot.
    - The spline function is also constrained to be nearly linear when X exceeds the largest knot.



Next Presentation

Next, We'll Cover:

Smoothing Splines

Generalized Additive Models (GAM)

Stay tuned for more on advanced methods in modeling!

R Code for Splines

Access the R code for Splines:

https://github.com/OluwaFunmibiOmotayo/Splines/blob/main/SplinesCode.R

References

- Armando Teixeira-Pinto, and Jaroslaw Harezlak. Machine Learning for Biostatistics Module 5: Beyond Linearity, University of Sydney.
- Martin, Kumar, Lao. Bayesian Modeling and Computation in Python: Splines https://bayesiancomputationbook.com/markdown/chp_05.html