**Section 6.6: Beyond Linearity in Python**

**Duration:** 2.5 hours

**Concepts:**

* Polynomial Regression
* Step Functions
* Regression Splines
* Smoothing Splines
* Local Regression
* Generalised Additive Models

**Textbook section:** An Introduction to Statistical Learning on Python, Chapter 7

|  |  |
| --- | --- |
| **Materials and Resources** | **Learning Goals** |
| * Computers for students with Jupyter Notebook * Slides * Exercises in Jupyter Notebook file | * Fit the different models to data * Use anovas to pick the best model complexity |

|  |  |  |
| --- | --- | --- |
| **Duration** | **Lesson Section** | **Learning Objectives** |
| 10 mins | Go through the polynomial regression section of the slides. | * Polynomial regression |
| 20 mins | Go through the polynomial regression section in the Jupyter Notebook file as a class. | * Use `sm.OLS()` to fit a polynomial regression * Plot fit with confidence intervals * Use `anova\_lm()` to compare polynomial regression of increasing degrees |
| 8 mins | Go through the step function section. | * Step function |
| 15 mins | Go through the step function section in the Jupyter Notebook file as a class. | * Use `pd.qcut()` and `pd.get\_dummies()` to fit a step function * Make predictions |
| 15 mins | Go through the regression splines section. | * Regression splines * Degrees of freedom * Location and number of knots * Constraints of regression splines |
| 15 mins | Go through the regression splines section in the Jupyter Notebook file as a class. | * Use `bs()` to fit a piecewise polynomial regression * Use `NaturalSpline()` and `ns()` to fit a natural spline * Plot results with confidence intervals |
| 15 mins | Go through the local regression section. | * Local regression |
| 15 mins | Go through the local regression section in the Jupyter Notebook file as a class. | * Use `lowess()` to perform local regression * Plot the results with confidence intervals |
| 10 mins | Go through the generalised additive models section. | * GAMs * Pros and cons of GAMs |
| 20 mins | Go through the generalised additive models section in the Jupyter Notebook file as a class. | * Fit a GAM to data * Use `plot\_gam()` to summarise it * Use an anova to choose best GAM |