Data Analysis 2021 Spring





**Lecture 10:**

**Moving Beyond Linearity**

May 12 & May 17, 2021

**Taesoo Kwon**

[tskwon@seoultech.ac.kr](mailto:tskwon@seoultech.ac.kr)

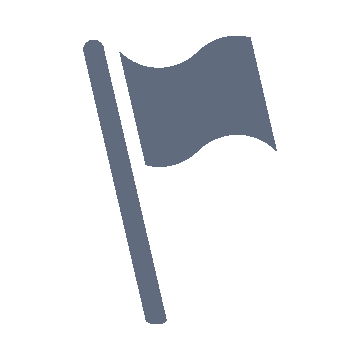
©copyright CIEL 2015 https://ciel.seoultech.ac.kr/

# Course Schedule (Tentative)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week** | **Topics** | **Note** | **Date (W)** | **Date (M)** |
| 1 | Orientation, Statistical Learning (Ch2) | Online | 03/03 | 03/08 |
| 2 | Statistical Learning (Ch2), Python Programming | Online | 03/10 | 03/15 |
| 3 | Probability & Statistics | Online | 03/17 | 03/22 |
| 4 | Probability & Statistics | Online | 03/24 | 03/29 |
| 5 | Linear Regression (Ch3) | Online | 03/31 | 04/05 |
| 6 | Linear Regression (Ch3) | Online | 04/07 | 04/12 |
| 7 | Classification (Ch4) | Online | 04/14 | 04/19 |
| 8 | **Midterm exam** | **Class hours (W1-W7)** | **04/21** | **04/26** |
| 9 | Resampling Methods (Ch5) | Online | 04/28 | 05/03 |
| 10 | Linear Model Selection and Regularization (Ch6) | Online | 05/05 | 05/10 |
| **11** | Moving Beyond Linearity (Ch7) | Online | 05/12 | 05/17 |
| 12 | Tree-Based Methods (Ch8) | Online | 05/19 | 05/24 |
| 13 | Support Vector Machines (Ch9) | Online | 05/26 | 05/31 |
| 14 | Unsupervised Learning (Ch10) | Online | 06/02 | 06/07 |
| 15 | **Final exam** | **7pm or Class hours (W9-W14)** | **06/??** | **06/??** |

©copyright CIEL 2015 **2**





* Moving beyond linearity

**OUTLINES**

* + Polynomial regression, step functions, basis function
  + Regression splines, smoothing splines
  + Local regression
  + Generalized additive models (GAM)
* Python lab
* Summary & Next class

©copyright CIEL 2015



**Moving Beyond Linearity**



**: Ch 7**

## Moving beyond linearity

* Python lab
* Summary & Next class

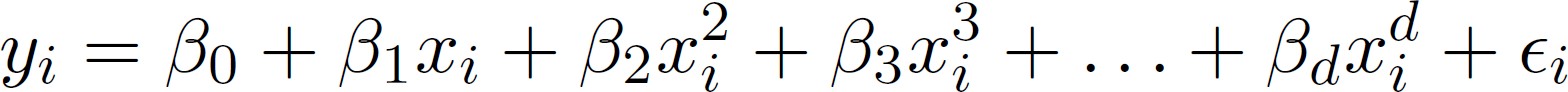
©copyright CIEL 2015 https://ciel.seoultech.ac.kr/

# Moving Beyond Linearity

* The truth is never linear!
* Or almost never!
* But often the linearity assumption is good enough
* When it’s not ...
* The followings offer a lot of flexibility, without losing the ease and interpretability of linear models:
  + polynomials,
  + step functions,
  + splines,
  + local regression, and
  + generalized additive models (GAM)

©copyright CIEL 2015 **5**

# Polynomial Regression



Degree 4 polynomial

Estimated 95% confidence interval

~ 2 × standard error

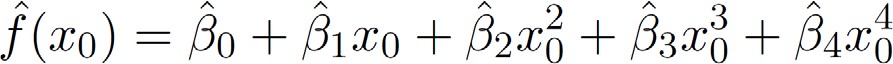




©copyright CIEL 2015 **6**

# Polynomial Regression [cont.]

* Create new variables 𝑋𝑋1 = 𝑋𝑋, 𝑋𝑋2 = 𝑋𝑋2 etc and then treat as multiple linear regression
* Not really interested in coefficients; more interested in fitted function values at any value 𝑥𝑥0:



* Since 𝑓𝑓̂(𝑥𝑥0) is a linear function of

̂

𝑙𝑙

𝛽𝛽

, can get a simple expression for pointwise-variances

Var 𝑓𝑓̂(𝑥𝑥0)

at any value 𝑥𝑥0

* + In the figure we have computed the fit and pointwise standard errors on a grid of values for 𝑥𝑥0
  + We show

𝑓𝑓̂(𝑥𝑥0) ± 2 × SE

𝑓𝑓̂(𝑥𝑥0)

~ 95% confidence interval



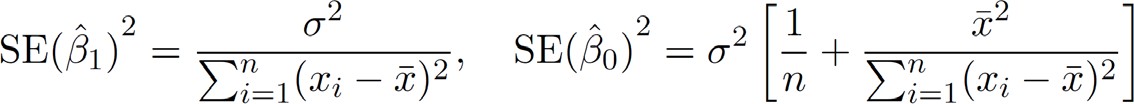
* We either fix the degree 𝑑𝑑 at some reasonably low value, else use cross-validation to choose 𝑑𝑑

©copyright CIEL 2015 **7**

# [Review] Assessing Accuracy of Coefficient Estimates

* The standard error of an estimator reflects how it varies under repeated sampling





Ross

Var 𝐵𝐵

=

𝜎𝜎

2

𝑆𝑆𝑥𝑥𝑥𝑥

,

Var 𝐴𝐴

= 𝜎𝜎 + 𝜎𝜎

2

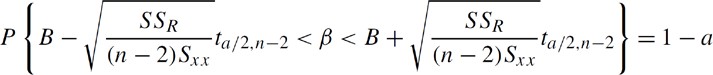
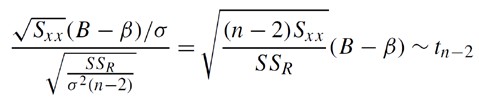
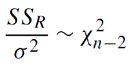
2

𝑛𝑛

𝑆𝑆𝑥𝑥𝑥𝑥

𝑥𝑥̅2

* Confidence intervals using standard errors
  + A 95% confidence interval is defined as a range of values such that with 95% probability, the range will contain the true unknown value of the parameter.
  + It has the form



Unknown

~Normal

~Square root

of 𝜒𝜒2 /(𝑛𝑛 − 2)

𝑛𝑛−2

𝑃𝑃 𝑡𝑡30 > 2.04 = 0.025

**SE**(𝜷𝜷�𝟏𝟏)

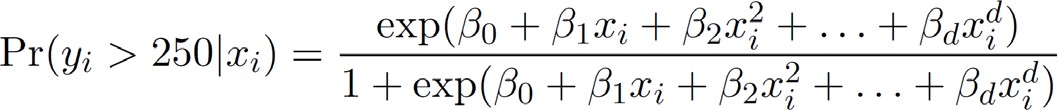


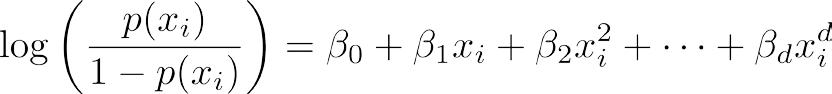
Ross

©copyright CIEL 2015 **8**

# Polynomial Regression [cont.]

* Logistic regression follows naturally
  + For example, in figure we model



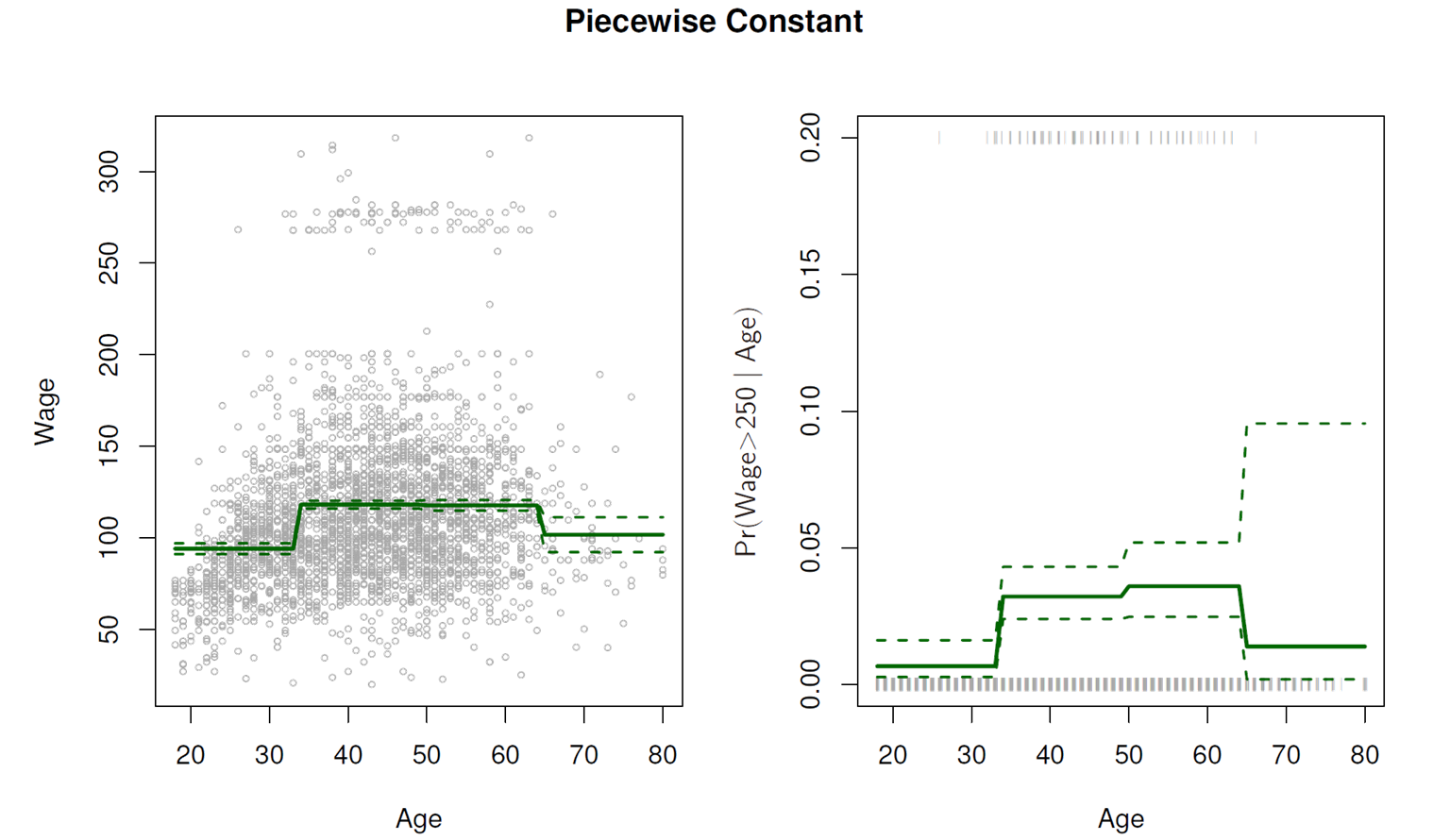
* To get confidence intervals, compute upper and lower bounds on the logit scale, and then invert to get on probability scale
  + Logit
* Can do separately on several variables ( just stack the variables into one matrix), and separate out the pieces afterwards (see GAMs later)
* Can fit using y ~ poly(x, degree = 3) in formula.

©copyright CIEL 2015 **9**

# Step Functions

* Another way of creating transformations of a variable
  + Cut the variable into distinct regions  step functions (piecewise constant regression)

©copyright CIEL 2015





**10**

# Step Functions [cont.]

* Easy to work with
  + Creates a series of dummy variables representing each group
* Useful way of creating interactions that are easy to interpret
  + For example, interaction effect of Year and Age:

𝐼𝐼(Year < 2500) � Age, 𝐼𝐼(Year ≥ 2500) � Age

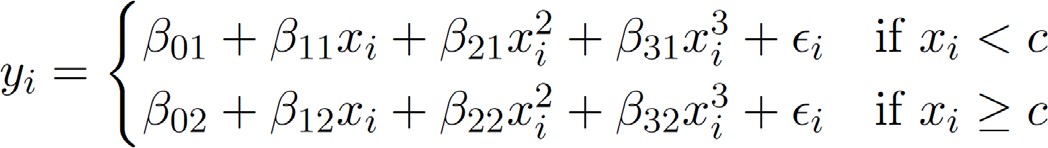
would allow for different linear functions in each age category

* Choice of cutpoints or knots can be problematic
* For creating nonlinearities, smoother alternatives such as splines are available.

©copyright CIEL 2015 **11**

# Piecewise Polynomials

* Instead of a single polynomial in 𝑋𝑋 over its whole domain, we can rather use different polynomials in regions defined by knots
  + E.g. (see figure)

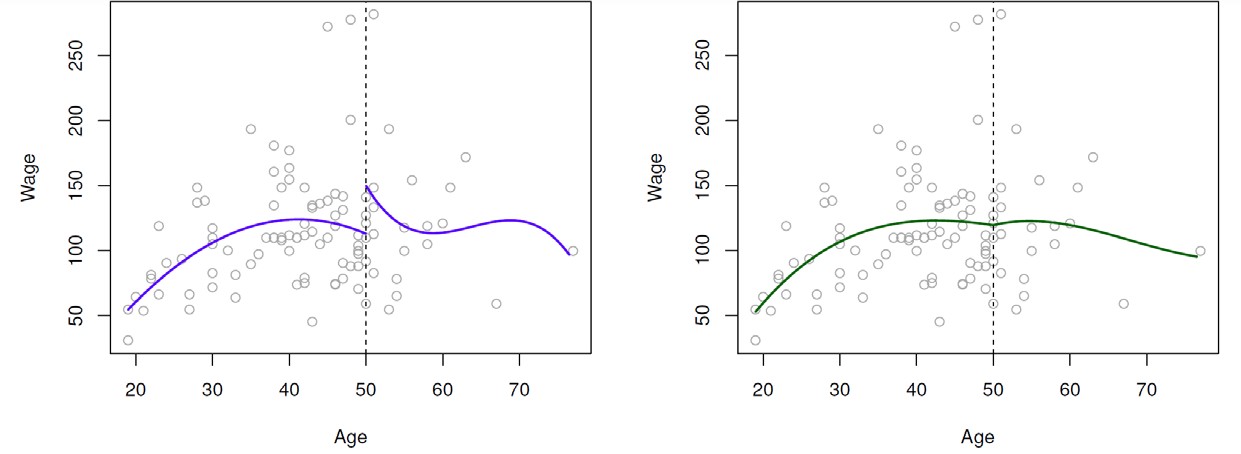


* Better to add constraints to the polynomials, e.g., continuity
* Splines have the "maximum" amount of continuity

©copyright CIEL 2015 **12**

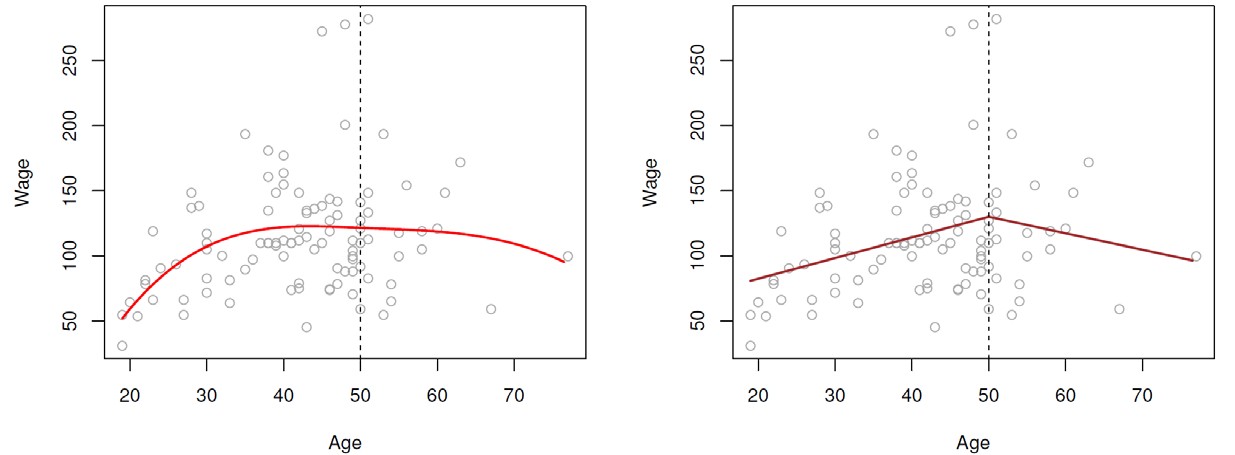
# Piecewise Polynomials [cont.]

**Piecewise cubic Continuous**



**piecewise cubic**

**Linear spline**



**Cubic spline**

©copyright CIEL 2015

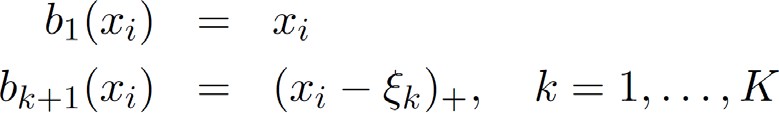
**13**

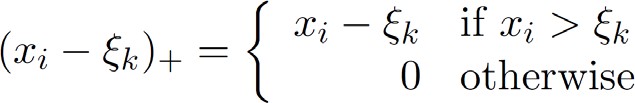
# Linear Splines

A linear spline with knots at 𝜉𝜉𝑘𝑘, 𝑘𝑘 = 1, ⋯ , 𝐾𝐾 is a piecewise linear polynomial continuous at each knot

* We can represent this model as

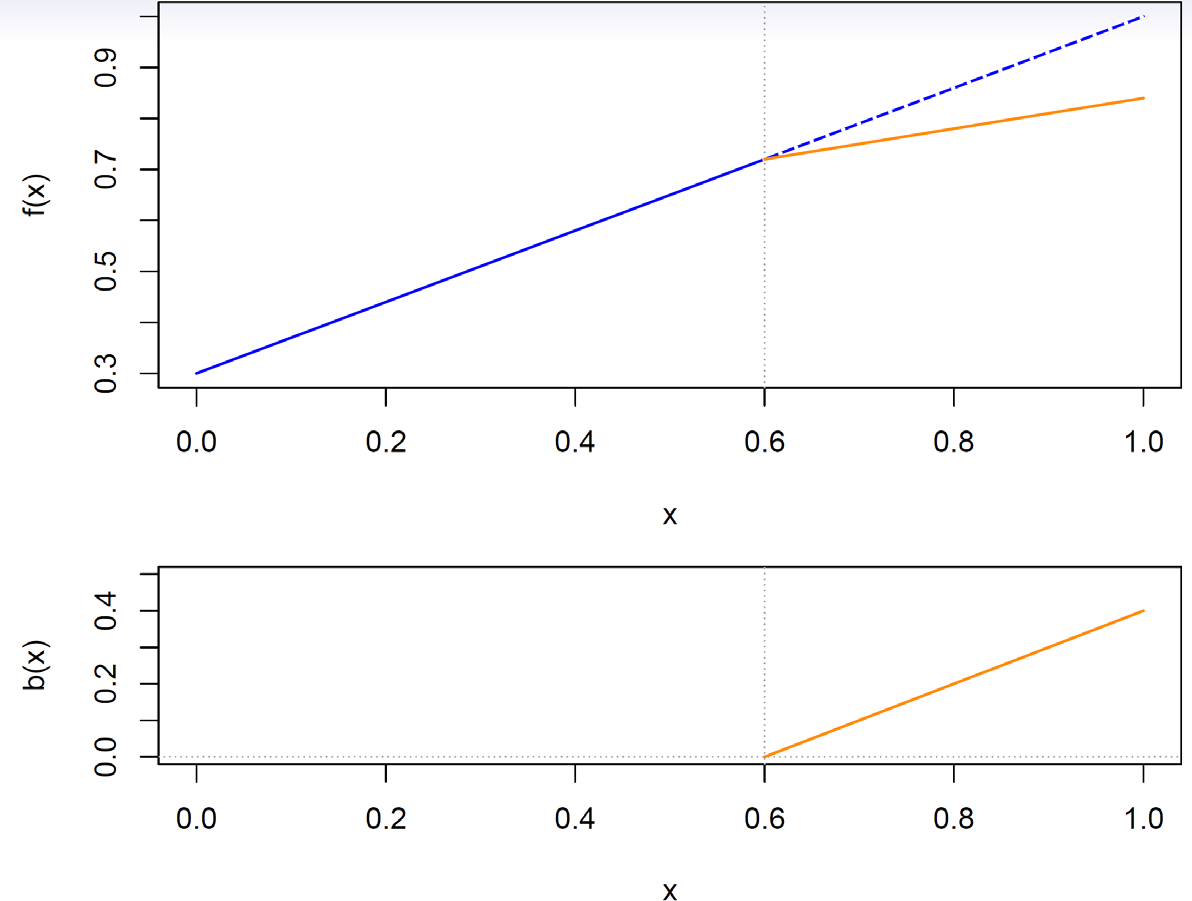


* + 𝑏𝑏𝑘𝑘 are basis functions
  + Here the ( )+ means positive part; i.e.,



©copyright CIEL 2015 **14**

# Linear Splines [cont.]

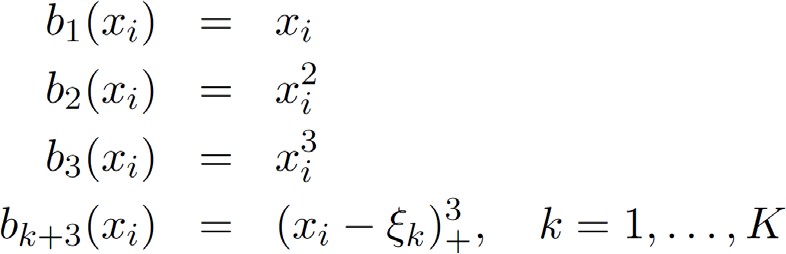


©copyright CIEL 2015 **15**

# Cubic Splines

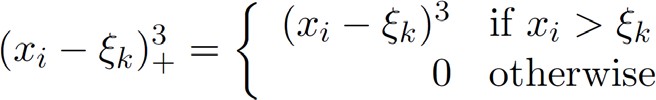
A cubic spline with knots at 𝜉𝜉𝑘𝑘, 𝑘𝑘 = 1, ⋯ , 𝐾𝐾 is a piecewise cubic polynomial with continuous derivatives up to order 2 at each knot

* Again we can represent this model with truncated power basis function





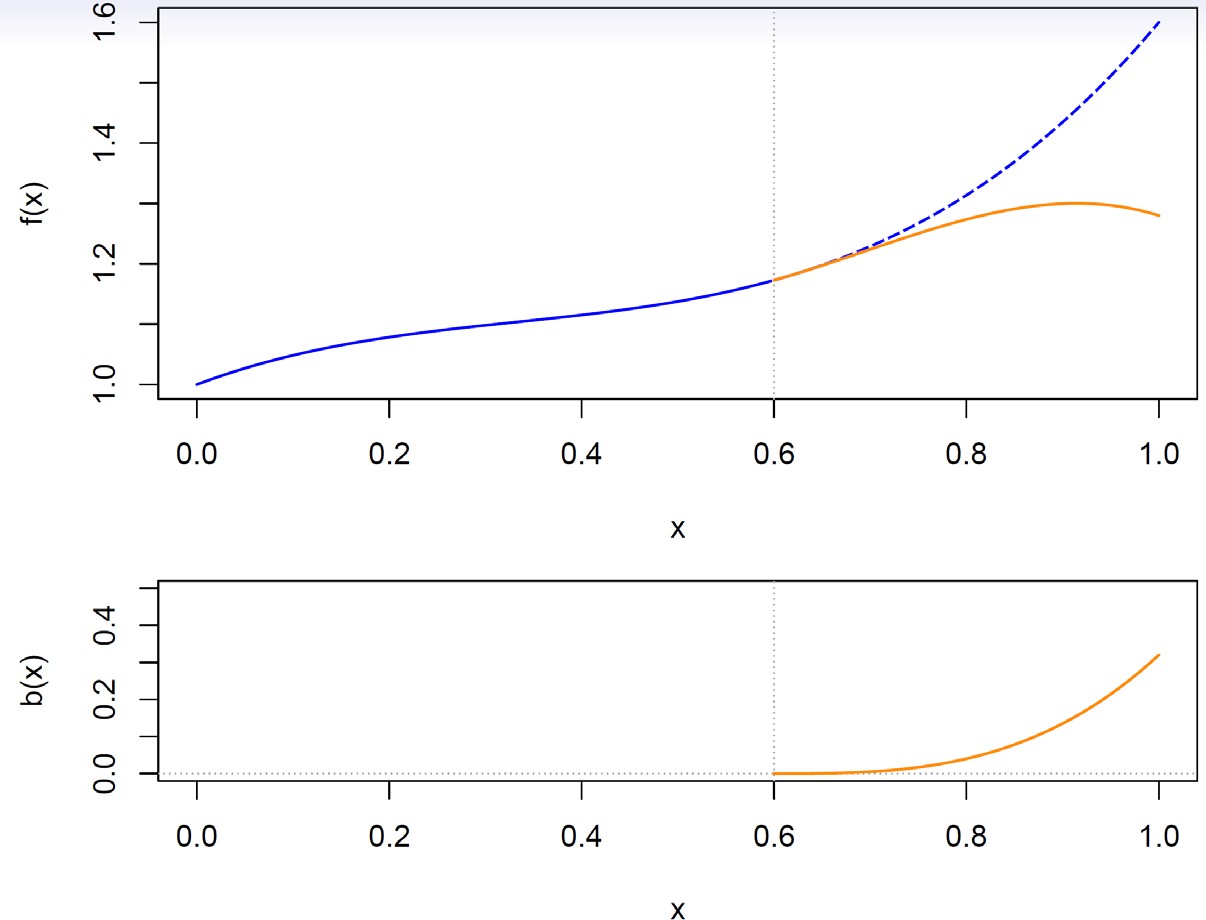
* + 𝑏𝑏𝑘𝑘 are basis functions

where 

©copyright CIEL 2015 **16**

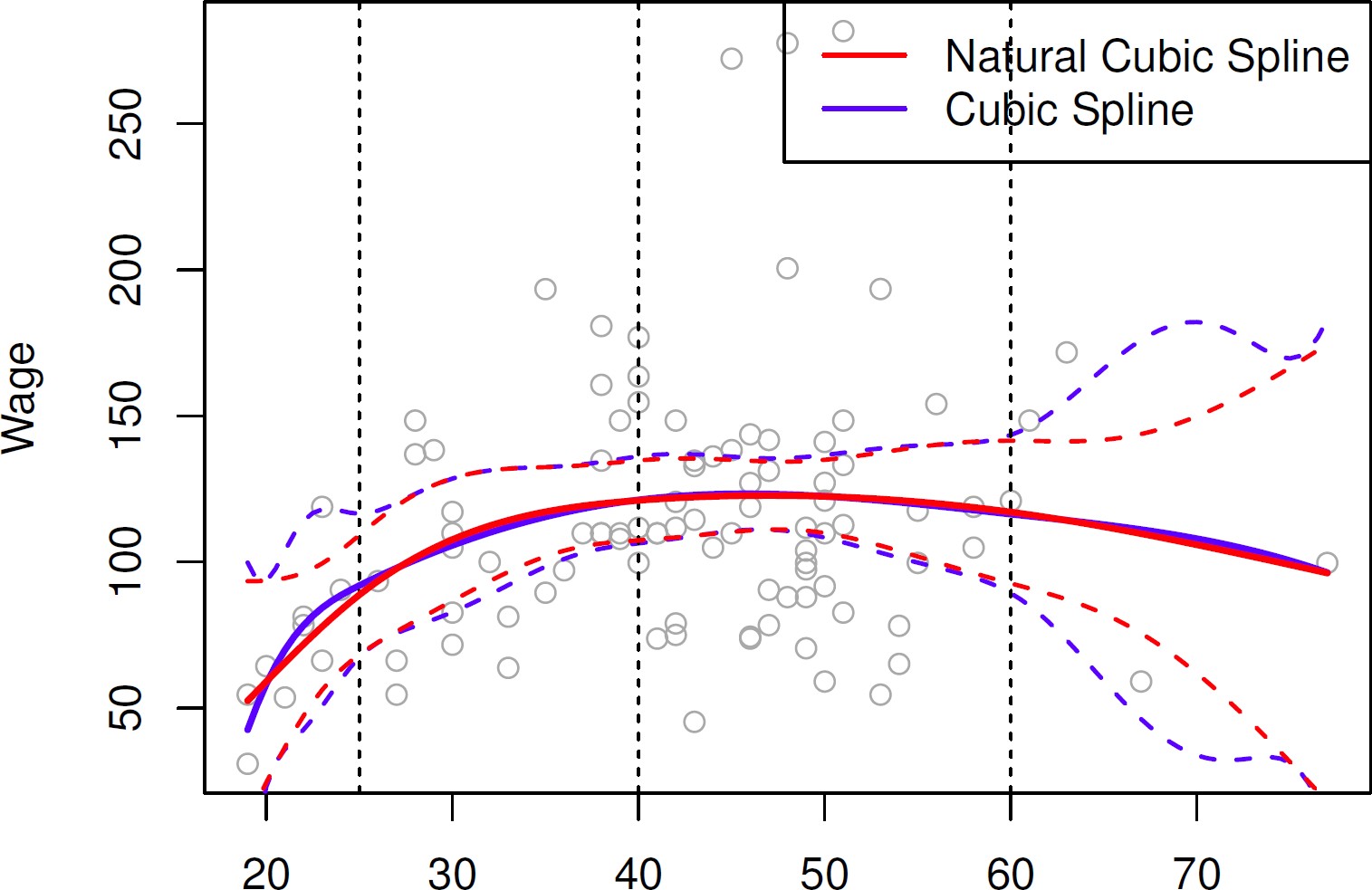
# Cubic Splines [cont.]

©copyright CIEL 2015 **17**



# Natural Cubic Splines

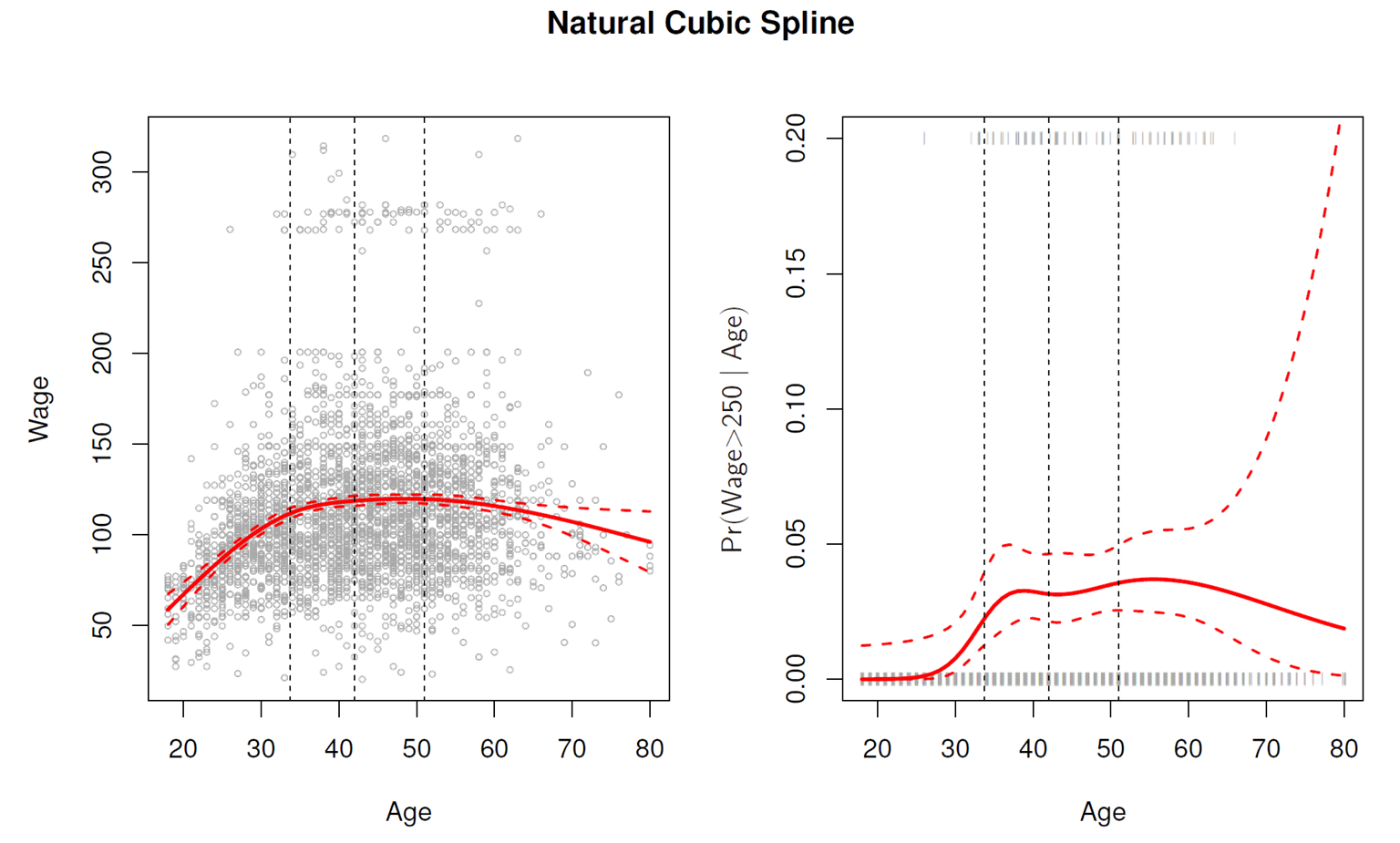
* A natural cubic spline extrapolates linearly beyond the boundary knots
* This adds 4 = 2 × 2 extra constraints, and allows us to put more internal knots for the same degrees of freedom as a regular cubic spline



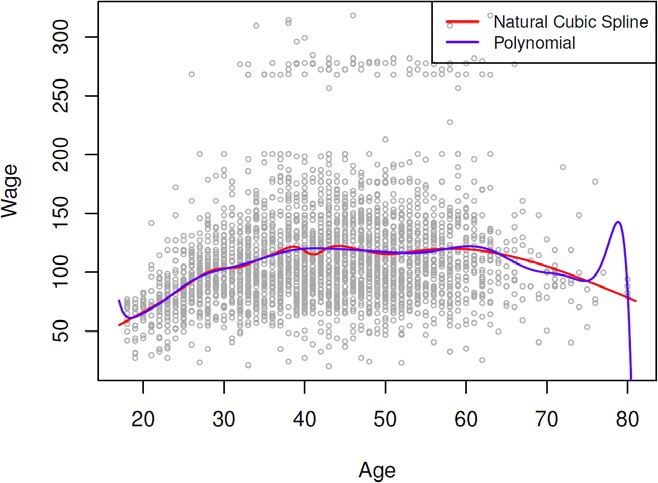
©copyright CIEL 2015 **18**

# Natural Cubic Splines [cont.]

©copyright CIEL 2015 **19**



# Knot Placement

* One strategy is to decide 𝐾𝐾, the number of knots, and then place them at appropriate quantiles of the observed 𝑋𝑋.
* A cubic spline with 𝐾𝐾 knots has 𝐾𝐾 + 4 parameters or degrees of freedom
* A natural spline with 𝐾𝐾 knots has 𝐾𝐾 degrees of freedom

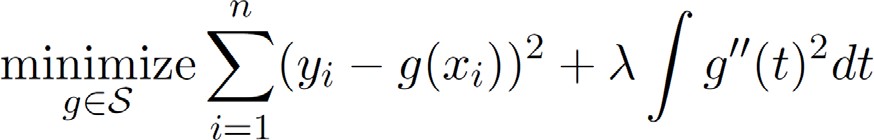
Comparison of a degree-14 polynomial and a natural cubic spline,

each with 15df

©copyright CIEL 2015 **20**

# Smoothing Splines

* This section is a little bit mathematical
* Consider this criterion for fitting a smooth function 𝑔𝑔(𝑥𝑥) to some data:



* The first term is RSS, and tries to make 𝑔𝑔(𝑥𝑥) match the data at each 𝑥𝑥𝑖𝑖
* The second term is a roughness penalty and controls how wiggly 𝑔𝑔(𝑥𝑥) is.
* It is modulated by the tuning parameter λ ≥ 0
  + The smaller λ, the more wiggly the function, eventually interpolating 𝑦𝑦𝑖𝑖 when λ = 0.
  + As λ → ∞, the function g(x) becomes linear

©copyright CIEL 2015 **21**

# Smoothing Splines [cont.]

* The solution is a natural cubic spline, with a knot at every unique value of 𝑥𝑥𝑖𝑖
* The roughness penalty still controls the roughness via λ
* Some details
  + Smoothing splines avoid the knot-selection issue, leaving a single λ to be chosen.
  + The algorithmic details are too complex to describe here

o In Python or R, the algorithm has been implemented

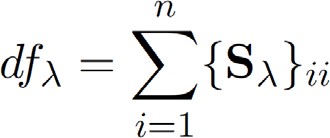
* + The vector of 𝑛𝑛 fitted values can be written as

λ)

* + The effective degrees of freedom are given by

**g**�𝜆𝜆 = **S**𝜆𝜆**y**, where **S**𝜆𝜆 is a 𝑛𝑛 × 𝑛𝑛 matrix (determined by 𝑥𝑥𝑖𝑖 and

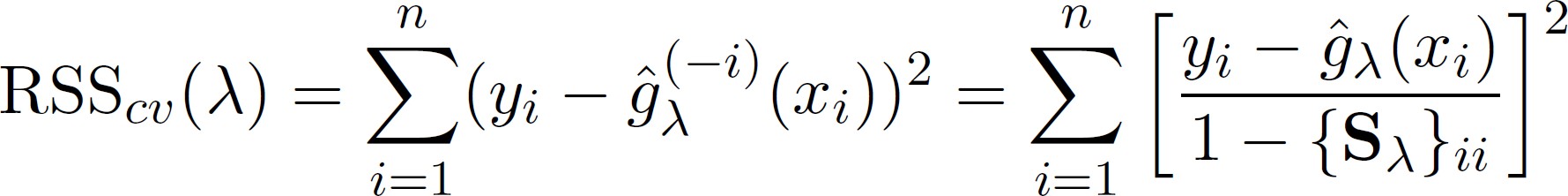




©copyright CIEL 2015 **22**

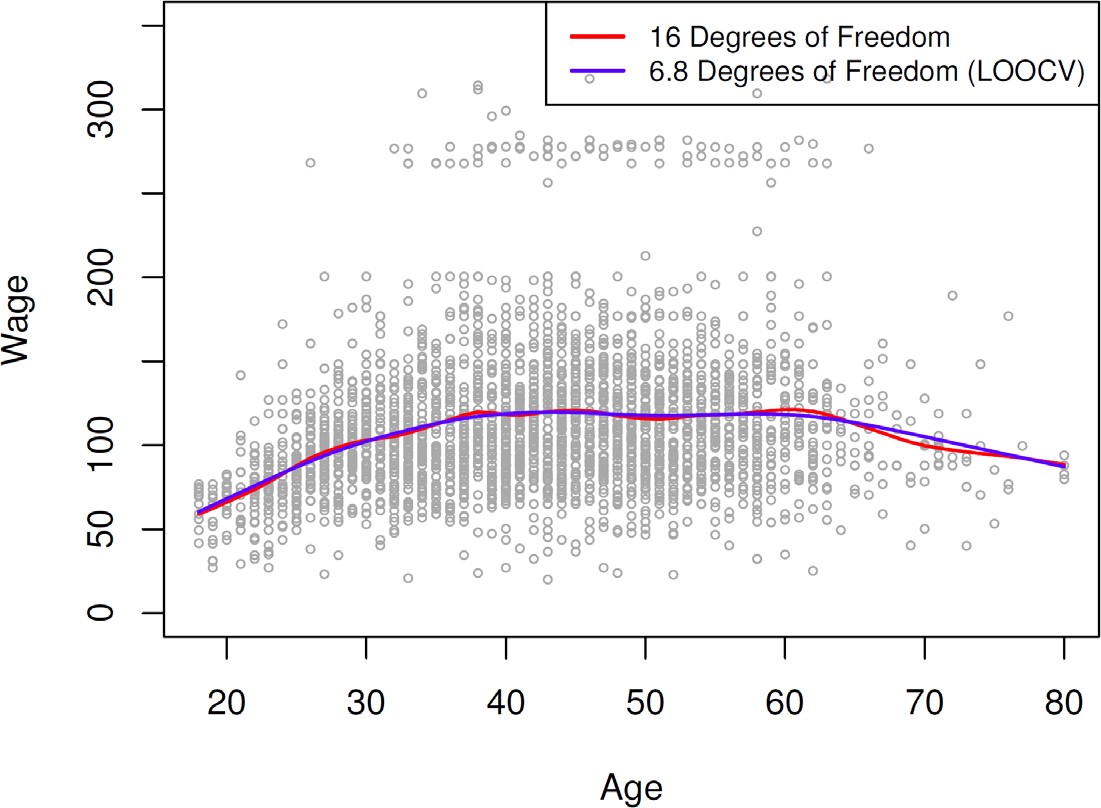
# Smoothing Splines [cont.]: Choosing 𝝀𝝀

* We can specify 𝑑𝑑𝑓𝑓 rather than 𝜆𝜆
* The leave-one-out (LOO) cross-validated error is given by



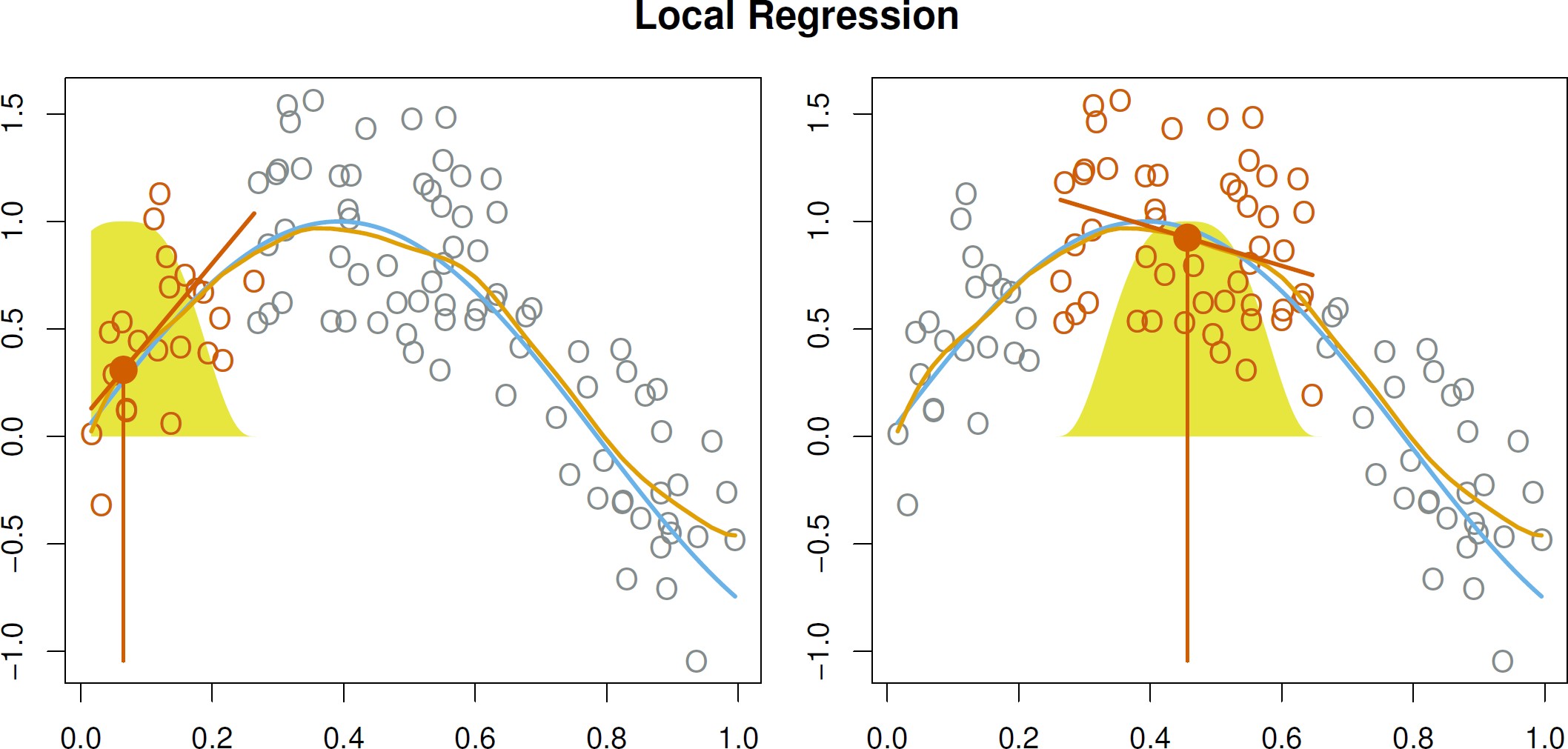
©copyright CIEL 2015 **23**

# Smoothing Splines [cont.]



©copyright CIEL 2015 **24**

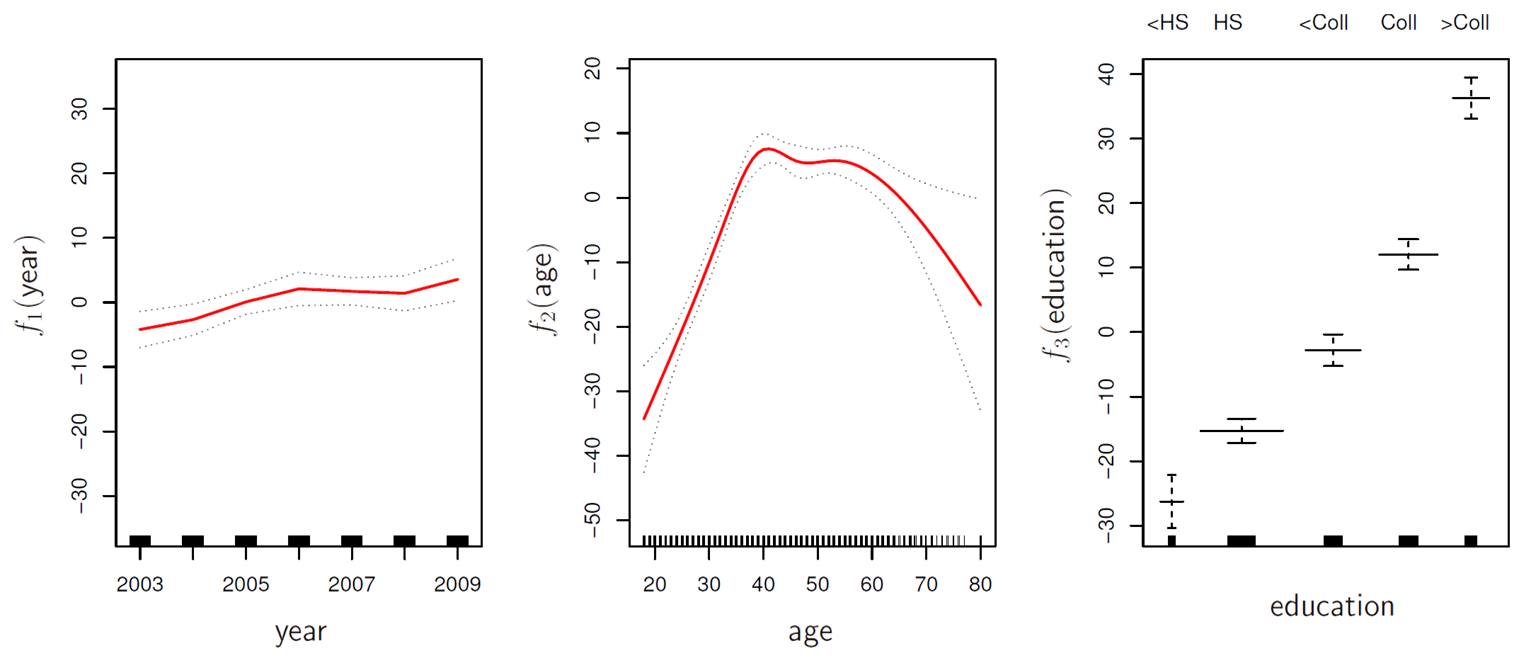
# Local Regression

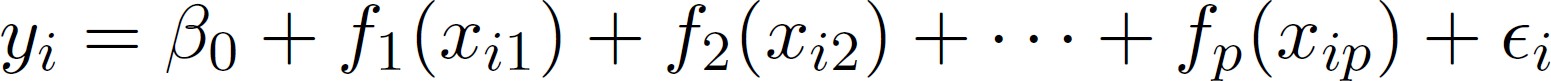
* With a sliding weight function, we fit separate linear fits over the range of 𝑋𝑋 by weighted least squares

©copyright CIEL 2015 **25**

# Generalized Additive Models [GAM]

* Allows for flexible nonlinearities in several variables, but retains the additive structure of linear models





©copyright CIEL 2015 **26**

# GAM [cont.]

* Can fit a GAM simply using, e.g. natural splines:



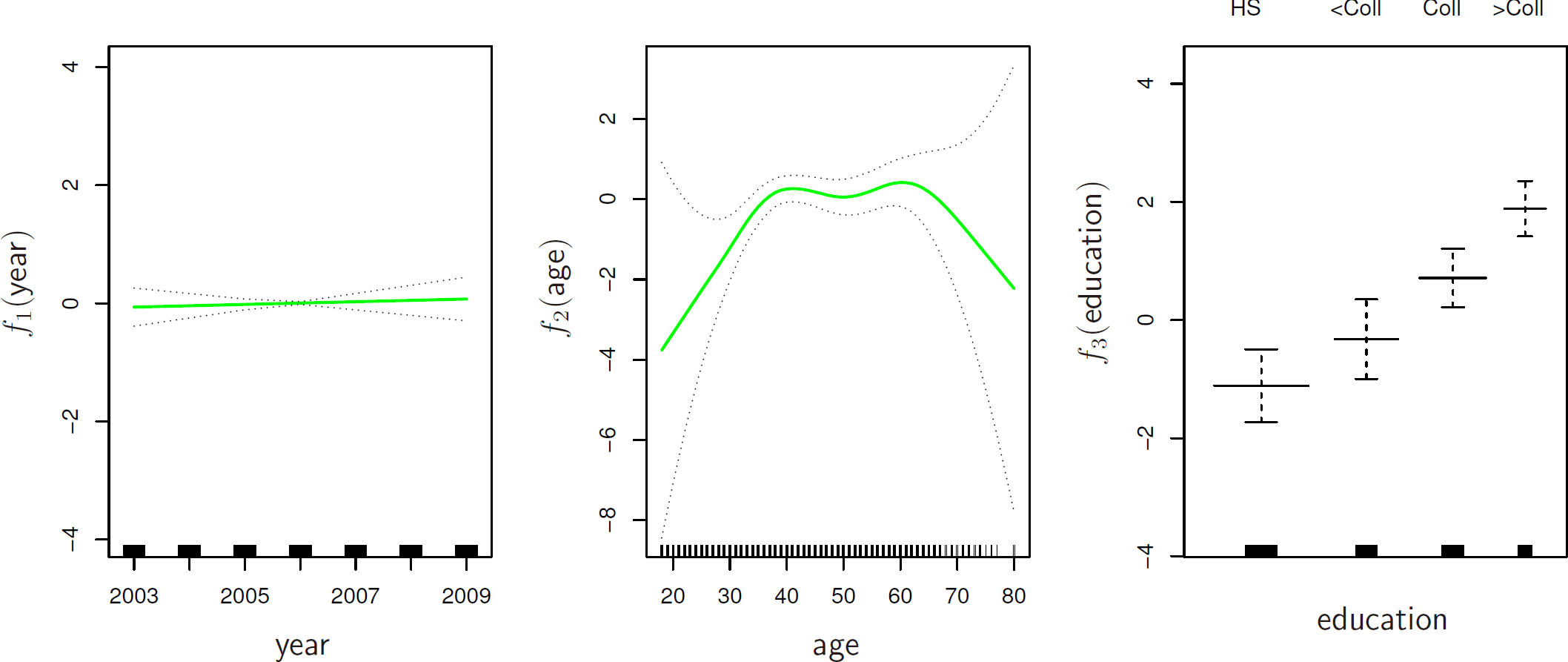
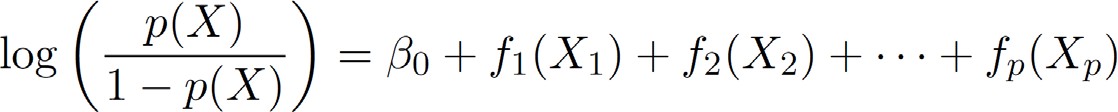
* Can mix terms: some linear, some nonlinear
* Can use smoothing splines or local regression as well

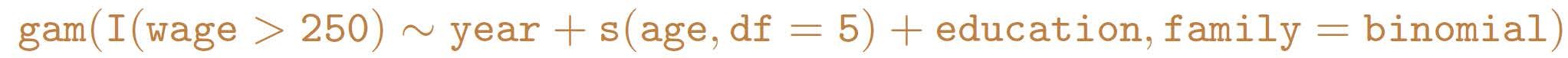


* GAMs are additive, although low-order interactions can be included in a natural way using, e.g. bivariate smoothers or interactions

©copyright CIEL 2015 **27**

# GAMs for Classification





©copyright CIEL 2015

**28**



**Python Lab**

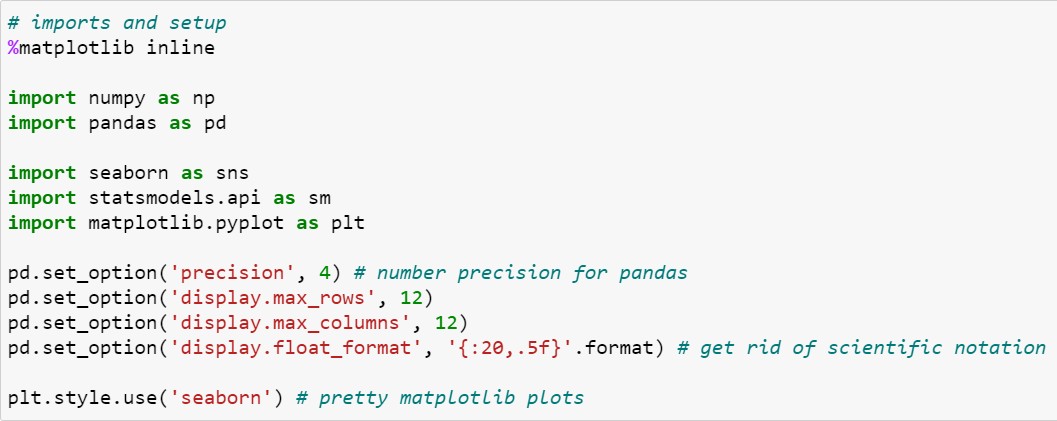
* Moving beyond linearity

## Python lab

* Summary & Next class

©copyright CIEL 2015 https://ciel.seoultech.ac.kr/

# 7.8 Lab: Non-linear Modeling

* + Using Python Libraries
    - Import the libraries that are often used for data analysis

©copyright CIEL 2015 **30**

# 7.8.1 Polynomial Regression and Step Functions

* + Load data: Wage data set
    - For reanalyzing Wage data considered in this class

panda.apply(pd.Categorical)



: pandas cast columns to category

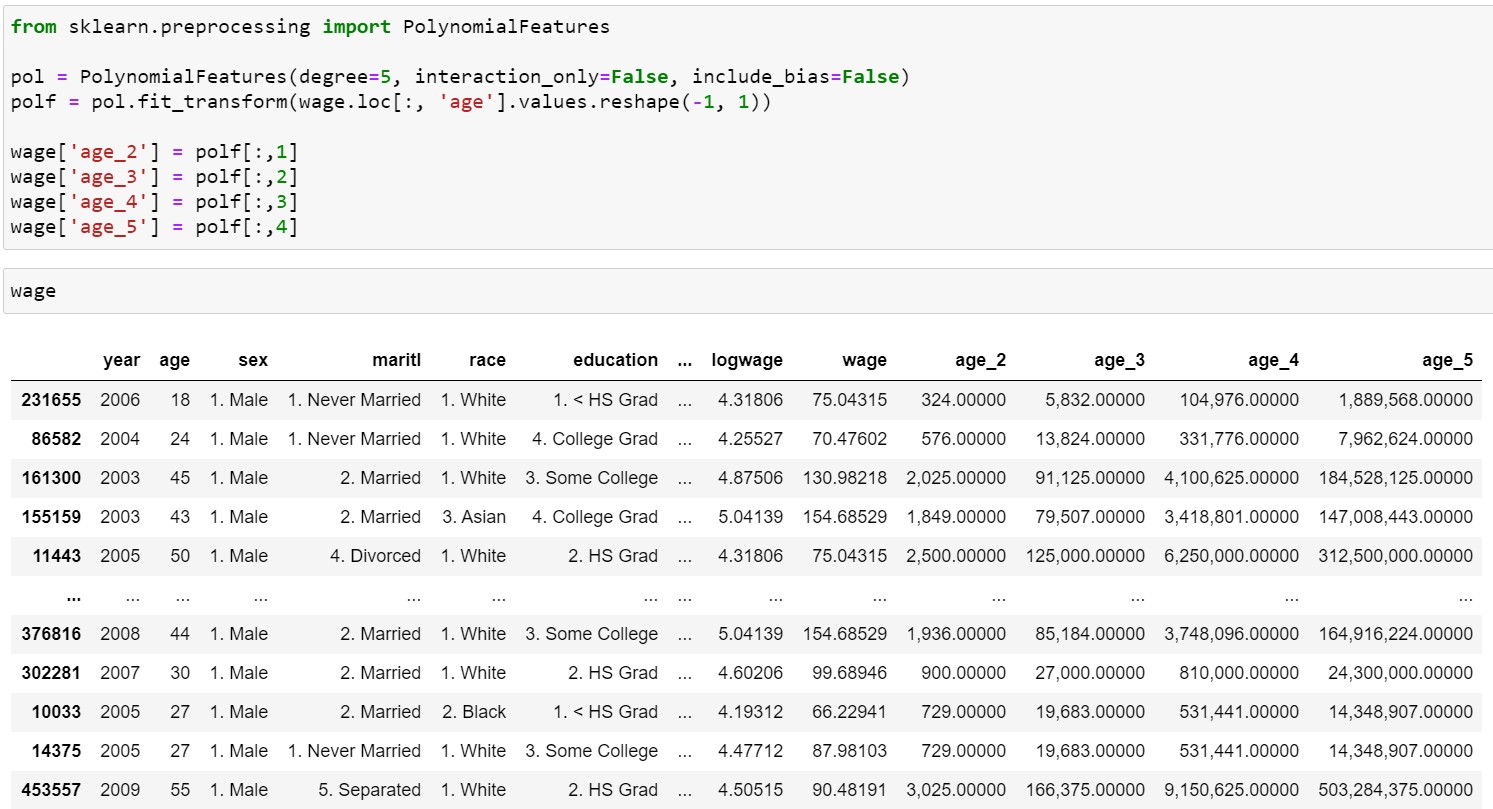
panda.apply

: apply a function along an axis of the DataFrame.

©copyright CIEL 2015 **31**

# 7.8.1 Polynomial Regression and Step Functions

* + Preprocessing for polynomial regression Generate polynomial and interaction features



interaction\_only: If true, only interaction features are produced

include\_bias: If true (default), then include a bias column

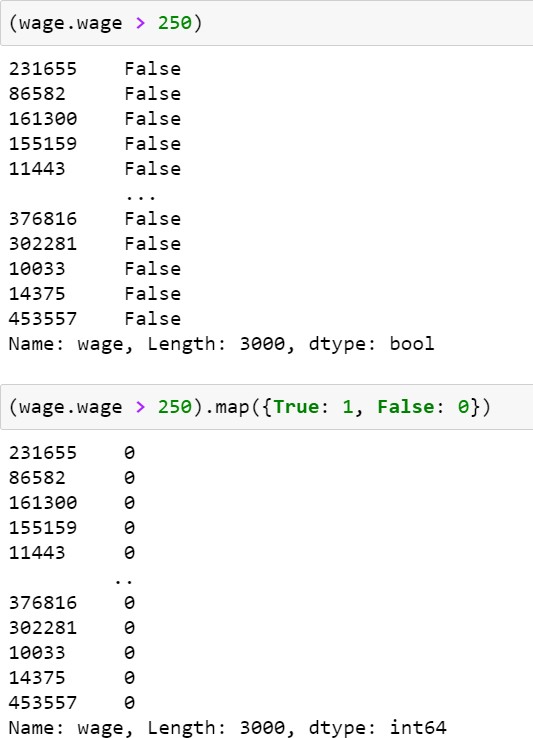
Fit to data, then transform it.

©copyright CIEL 2015 **32**

# Polynomial Regression and Step Functions

* + - * Preprocessing for polynomial regression

apply a function on all the elements of specified iterable and return map object



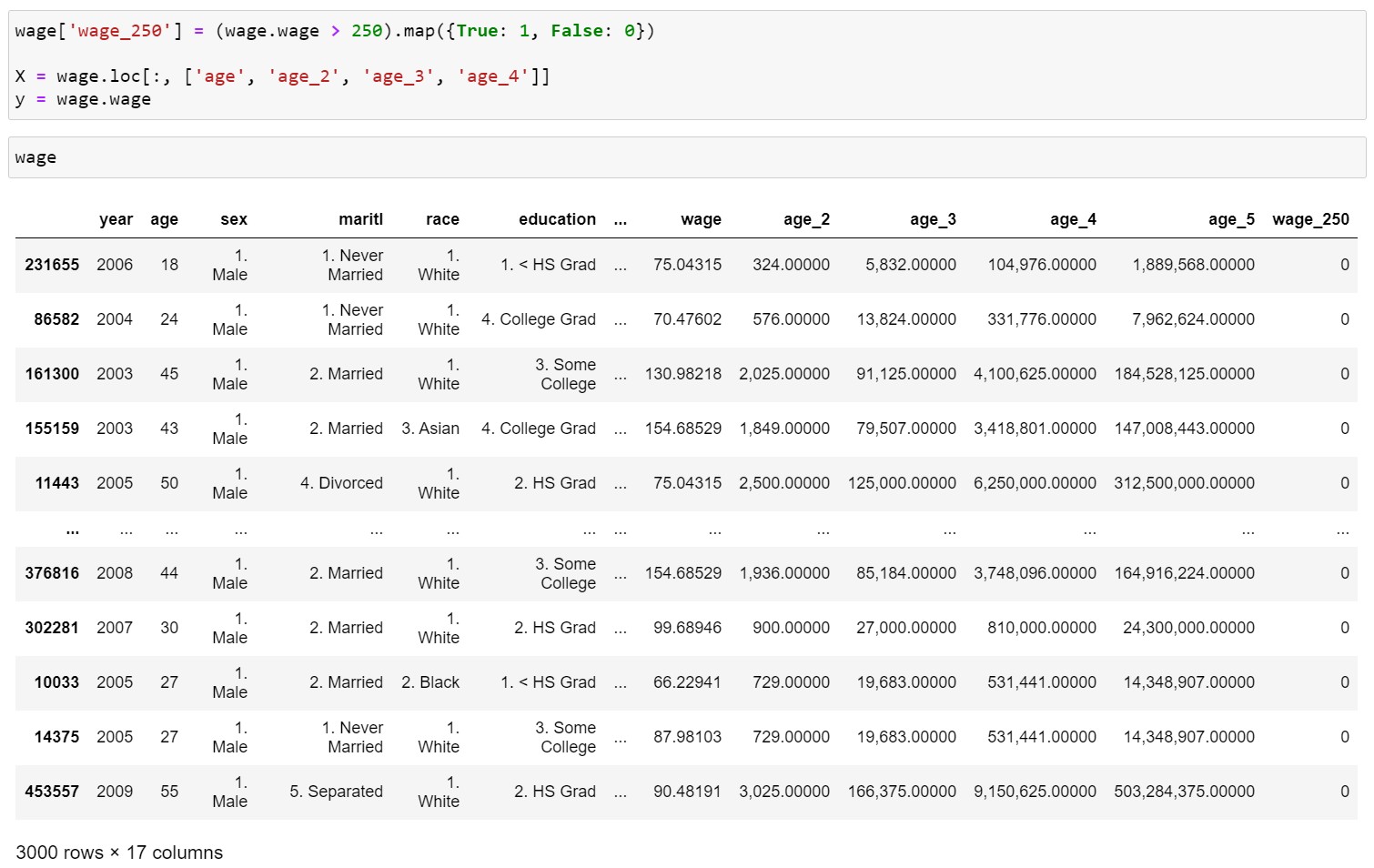
True1, False0

©copyright CIEL 2015 **33**

# 7.8.1 Polynomial Regression and Step Functions

* + Preprocessing for polynomial regression

True1, False0



©copyright CIEL 2015 **34**

# 7.8.1 Polynomial Regression and Step Functions

* + Polynomial regression using linear regression & using seaborn.regplot

Linear regression using age\_2, …, age\_5



transparency

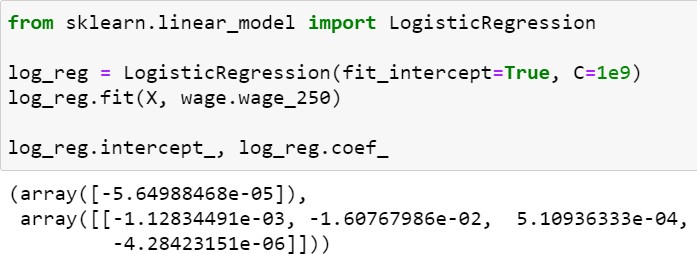
Plot regression using seaborn library

seaborn.regplot: plot data and a linear regression model fit.

©copyright CIEL 2015 **35**

# 7.8.1 Polynomial Regression and Step Functions

* + Logistic regression
    - Using scikit-learn



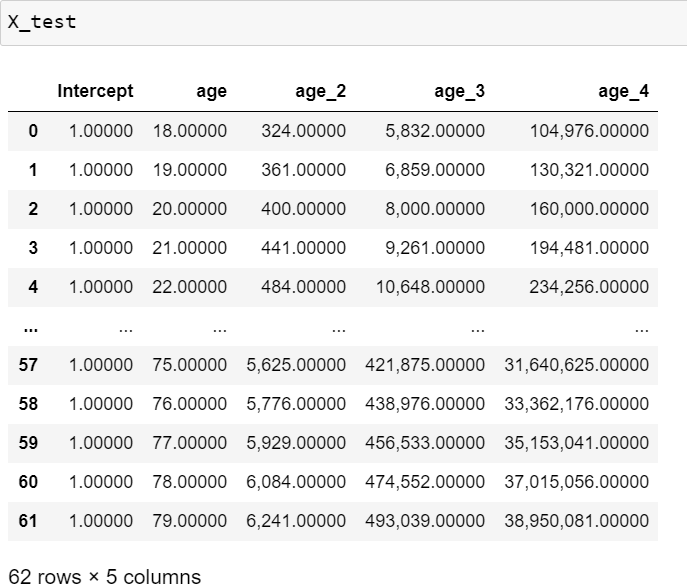
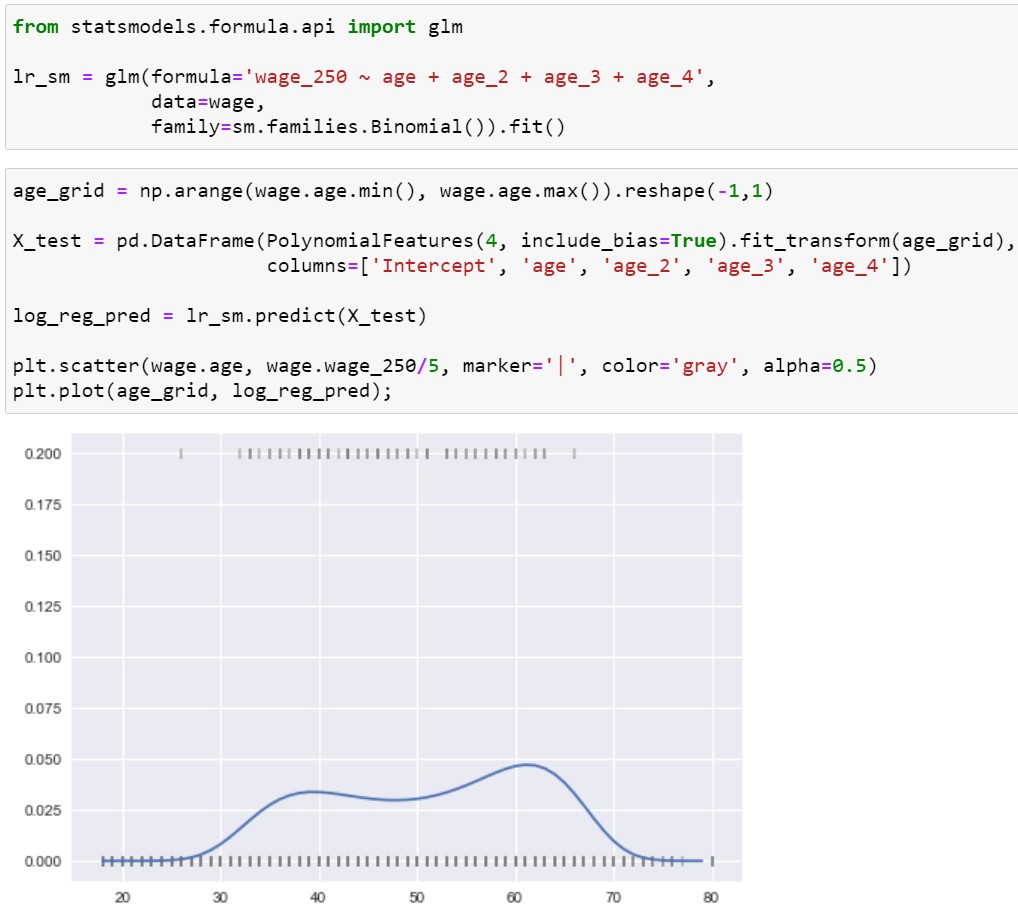
Inverse of regularization strength; must be a positive float. Smaller values specify stronger regularization



©copyright CIEL 2015 **36**

# 7.8.1 Polynomial Regression and Step Functions

* + More R compatible results
    - Using statsmodels.formula.api.glm



R compatible polynomial regression

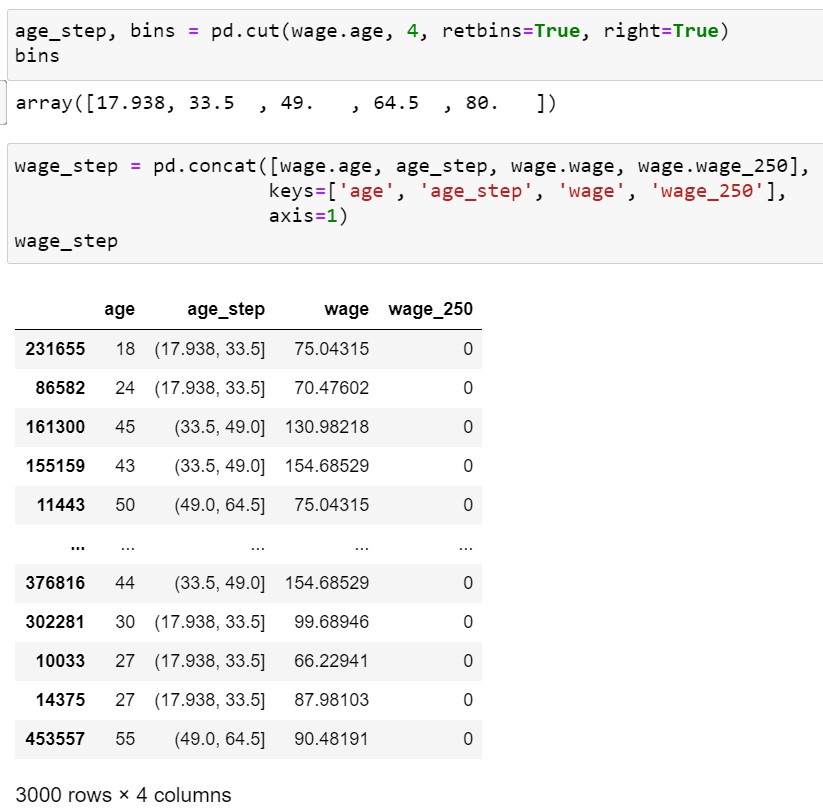
Implementing logistic regression using glm

©copyright CIEL 2015 **37**

# 7.8.1 Polynomial Regression and Step Functions

* + Step functions: preprocessing

Bin values into discrete intervals



bins: int, sequence of scalars, or IntervalIndex retbins: Whether to return the bins or not right: Indicates whether bins includes the

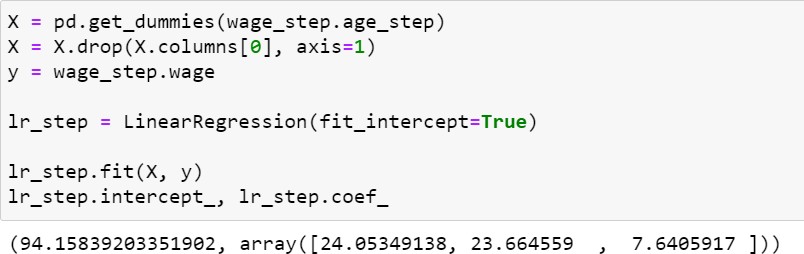
rightmost edge or not

Concatenate pandas objects along a particular axis with optional set logic along the other axes

©copyright CIEL 2015 **38**

# 7.8.1 Polynomial Regression and Step Functions

* + Step functions: regression



Convert categorical variable into dummy/indicator variables

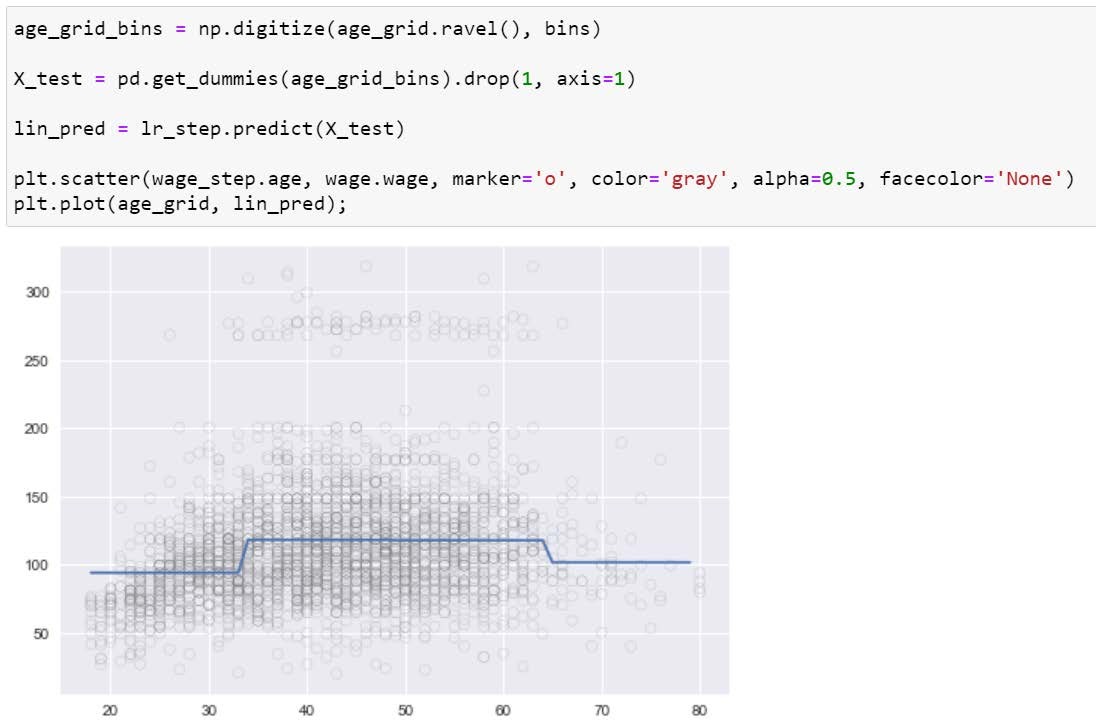


Drop specified labels from rows or columns

©copyright CIEL 2015 **39**

# 7.8.1 Polynomial Regression and Step Functions

* + Step functions: predicting & plotting Return the indices of the bins to which



each value in input array belongs

Return a contiguous flattened array, e.g.,

>> x = np.array([[1, 2, 3], [4, 5, 6]])

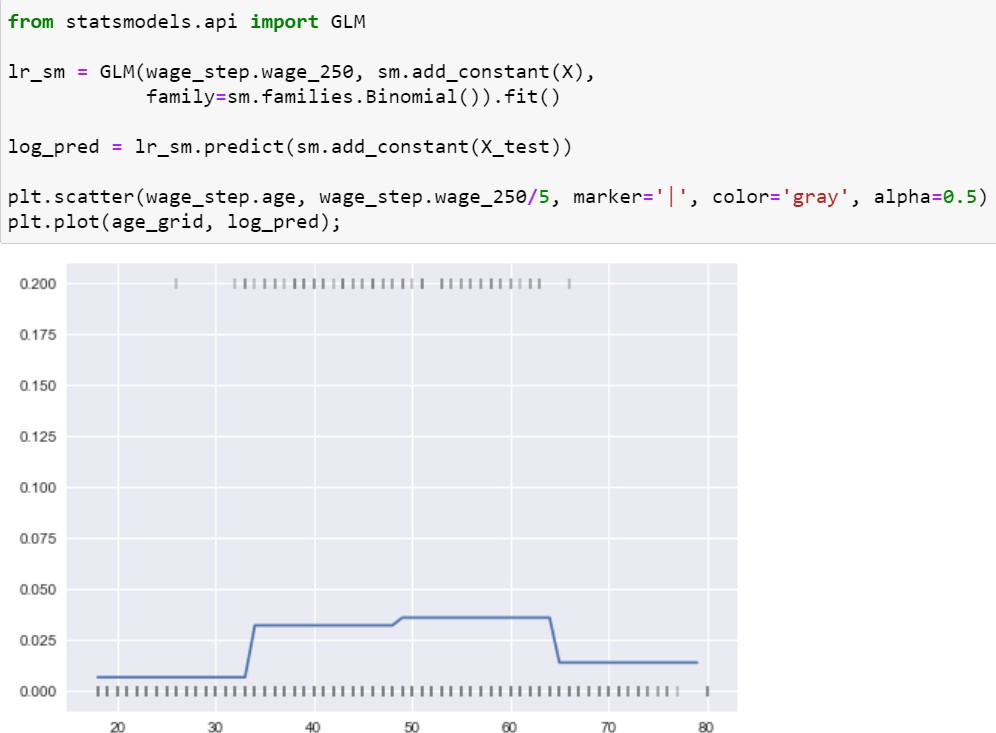
>> np.ravel(x)

array([1, 2, 3, 4, 5, 6])

©copyright CIEL 2015 **40**

# 7.8.1 Polynomial Regression and Step Functions

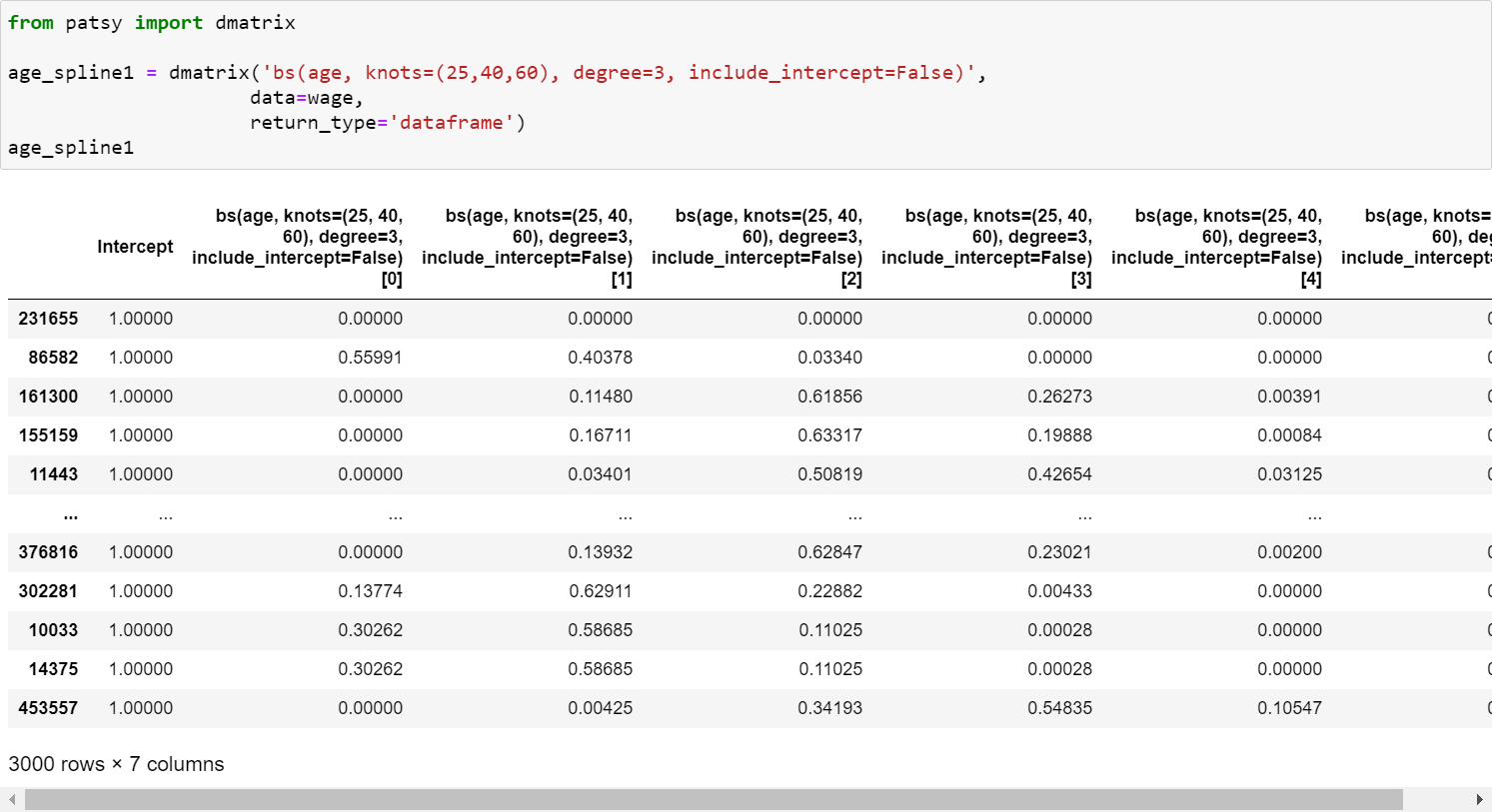
* + Step functions: for wage > 250



Logistic regression using GLM

©copyright CIEL 2015 **41**

# 7.8.2 Splines



* + Cubic splines with specified knots: preprocessing

Python library for describing statistical models (especially linear models) with a small string-based “formula syntax,” which is inspired by (but not exactly the same as) the formula syntax used by R



Construct a single design matrix given a formula\_like and data

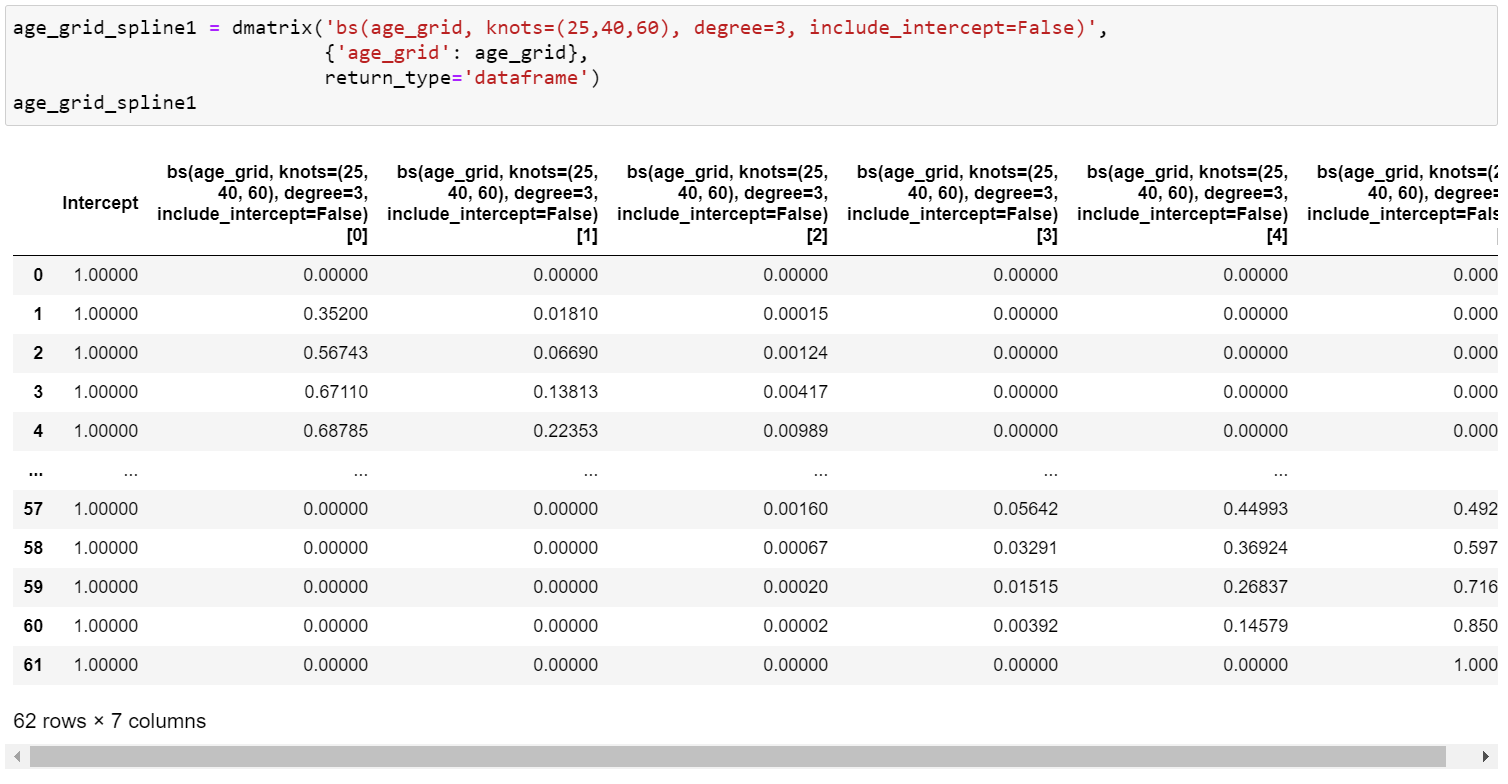
Generates a spline basis for x, allowing non-linear fits

©copyright CIEL 2015 **42**

# 7.8.2 Splines

* + Cubic splines with specified knots: preprocessing

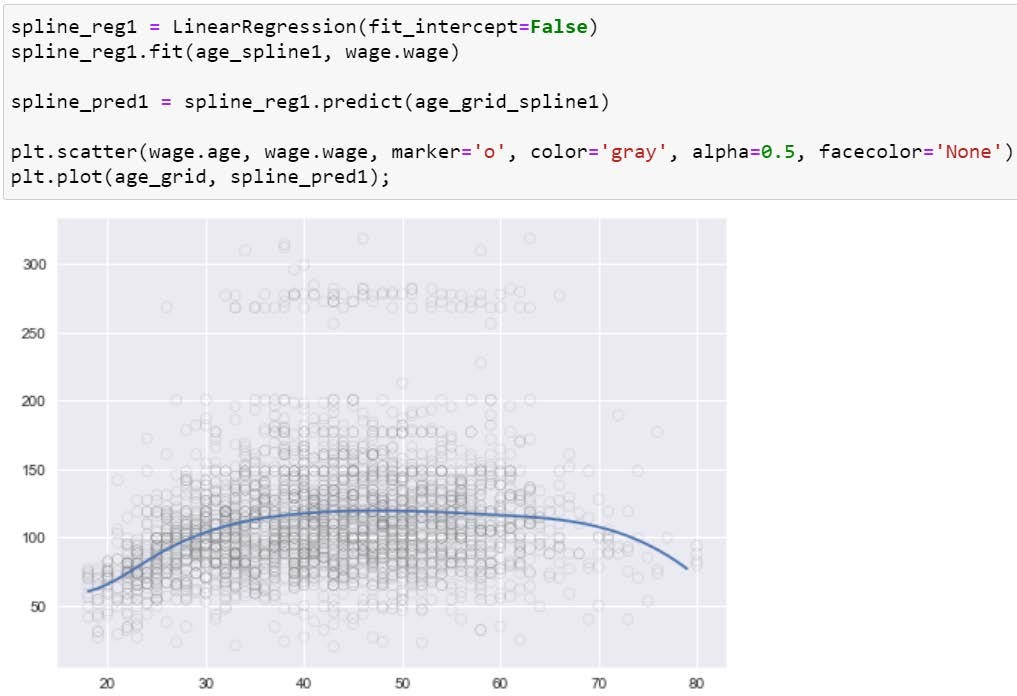
For predicting



©copyright CIEL 2015 **43**

# 7.8.2 Splines

* + Cubic splines with specified knots: applying linear regression



Applying linear regression

©copyright CIEL 2015 **44**

# 7.8.2 Splines

* + Cubic splines with specified degree of freedom



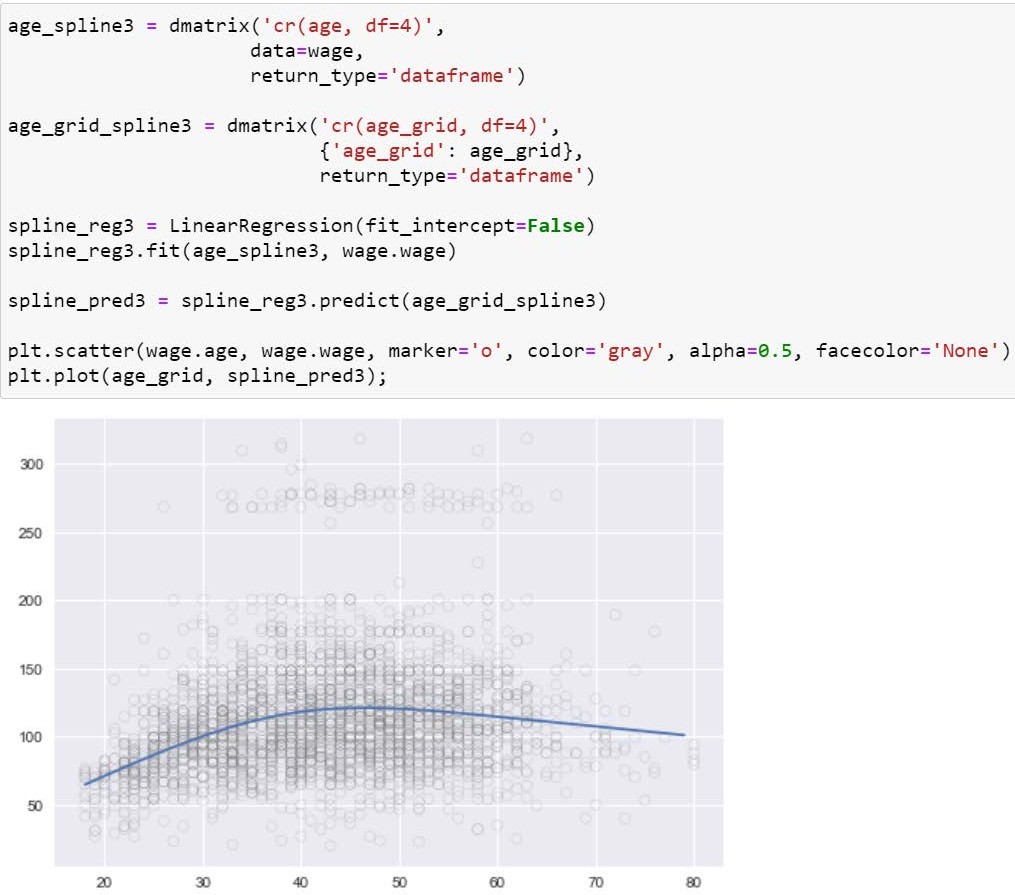
df: 6

©copyright CIEL 2015 **45**

# 7.8.2 Splines

* + Natural splines

Generates a natural cubic spline basis for x



df: The number of degrees of freedom to use for this spline. The return value will have this many columns. You must specify at least one of df and knots.

knots: The interior knots to use for the spline. If unspecified, then equally spaced quantiles of the input data are used. You must specify at least one of df and knots.

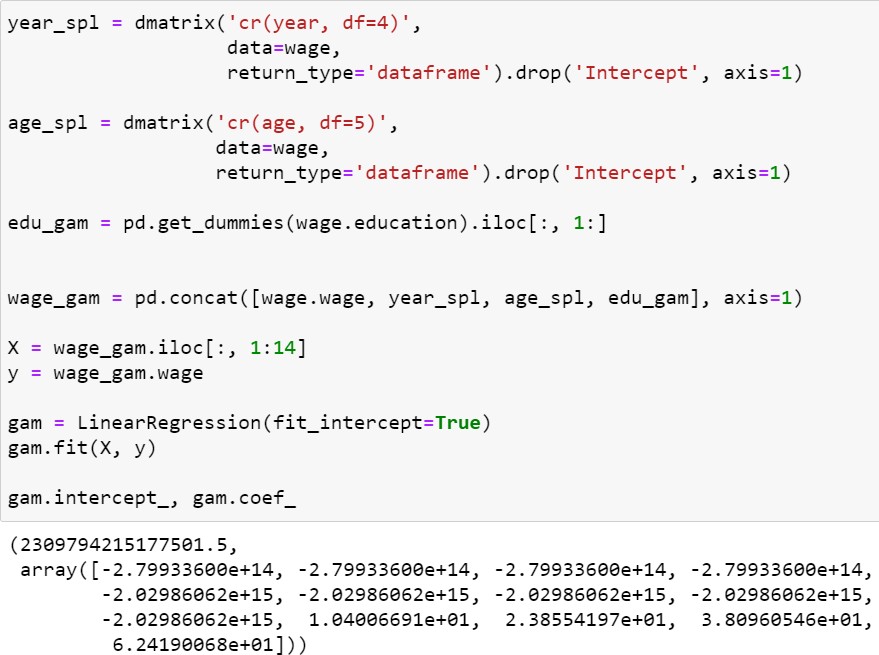


©copyright CIEL 2015 **46**

# 7.8.3 GAMs

* + GAMs

Natural cubic splines for each of two predictors



©copyright CIEL 2015 **47**



**Summary & Next Class**

* + - Moving beyond linearity
    - Python lab

## Summary & Next class

©copyright CIEL 2015 https://ciel.seoultech.ac.kr/

# Summary

* + Moving beyond linearity without losing the ease and interpretability of linear models
* Moving beyond linearity
* Polynomial regression
* Piecewise constant regression (step functions)
* Linear regression using basis function
* Regression splines, smoothing splines
* Local regression
* Generalized additive models (GAM)
  + Python lab
* Implementing polynomial regression, piecewise constant regression
* Implementing regression splines, smoothing splines
* Commonly, using linear regression functions

©copyright CIEL 2015 **49**

# Assignments

* + eClass > Assignments
* Upload files (do not compress them)
  + Python practices in today’s lecture
* Upload a single ipynb file
* Referring to the lecture slides marked with [P]
* File name: “StudentID” + “\_AssignmentNo w/ 2 digits” + “\_1.ipynb”, e.g., **20211234\_02\_1.ipynb**
  + Textbook exercise problems for today’s lecture
* Conceptual
  + Solving the given problems, then upload your own solution (only docx/hwp formats, not pdf/jpg/bmp etc.)
  + Only include your answers (not need to describe problems)
  + File name: “StudentID” + “\_AssignmentNo w/ 2 digits” + “\_2.ipynb”, e.g., **20211234\_02\_2.docx**
* Applied
  + Implement your Python code for the given problems, then upload another single ipynb file
  + File name: “StudentID” + “\_AssignmentNo w/ 2 digits” + “\_1.ipynb”, e.g., **20211234\_02\_3.ipynb**
  + If not complying with the above policies, some penalty on assignment scores may be given.

©copyright CIEL 2015 **50**

# Course Schedule (Tentative)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week** | **Topics** | **Note** | **Date (W)** | **Date (M)** |
| 1 | Orientation, Statistical Learning (Ch2) | Online | 03/03 | 03/08 |
| 2 | Statistical Learning (Ch2), Python Programming | Online | 03/10 | 03/15 |
| 3 | Probability & Statistics | Online | 03/17 | 03/22 |
| 4 | Probability & Statistics | Online | 03/24 | 03/29 |
| 5 | Linear Regression (Ch3) | Online | 03/31 | 04/05 |
| 6 | Linear Regression (Ch3) | Online | 04/07 | 04/12 |
| 7 | Classification (Ch4) | Online | 04/14 | 04/19 |
| 8 | **Midterm exam** | **Class hours (W1-W7)** | **04/21** | **04/26** |
| 9 | Resampling Methods (Ch5) | Online | 04/28 | 05/03 |
| 10 | Linear Model Selection and Regularization (Ch6) | Online | 05/05 | 05/10 |
| 11 | Moving Beyond Linearity (Ch7) | Online | 05/12 | 05/17 |
| **12** | Tree-Based Methods (Ch8) | Online | 05/19 | 05/24 |
| 13 | Support Vector Machines (Ch9) | Online | 05/26 | 05/31 |
| 14 | Unsupervised Learning (Ch10) | Online | 06/02 | 06/07 |
| 15 | **Final exam** | **7pm or Class hours (W9-W14)** | **06/??** | **06/??** |

©copyright CIEL 2015 **51**