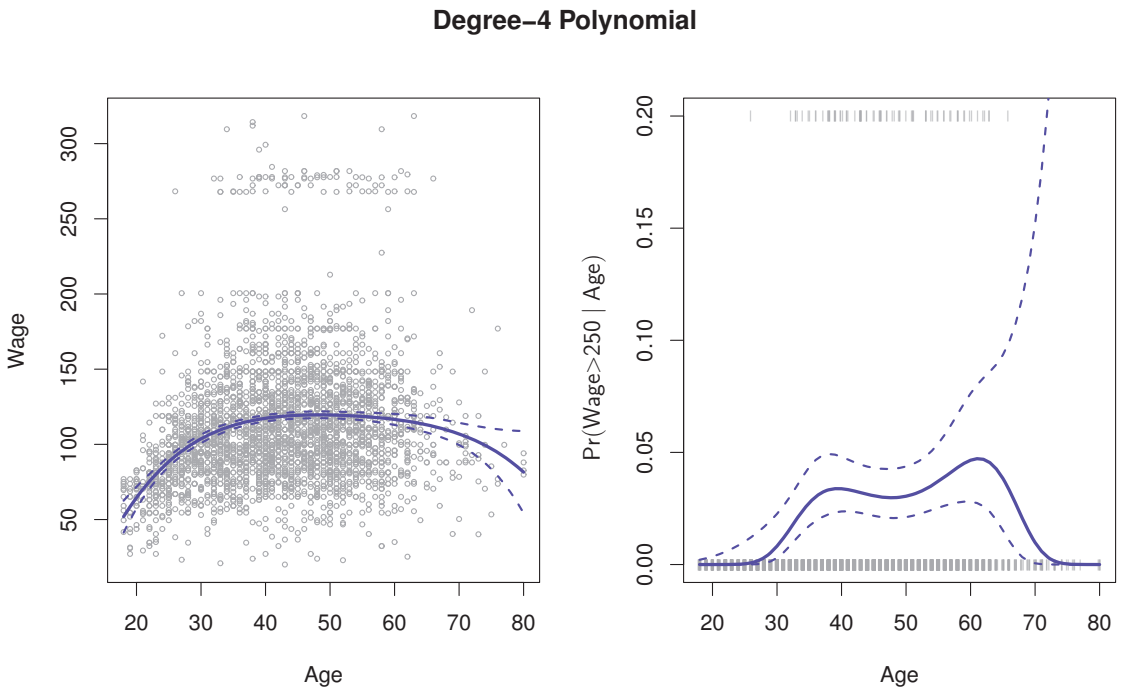


The Wage data which contains income and demographic information for males who reside in the central Atlantic region of the United States. Left: The solid blue curve is a degree-4 polynomial of wage (in thousands of dollars) as a function of age, fit by least squares. The dashed curves indicate an estimated 95% confidence interval. Right: We model the binary event wage > 250 using logistic regression, again with a degree-4 polynomial. The fitted posterior probability of wage exceeding \$250,000 is shown in blue, along with an estimated 95% confidence interval.

- It seems like the wages in figure are from two distinct populations: there appears to be a high earners group earning more than \$250,000 per annum, as well as a low earners group.
- We can treat wage as a binary variable by splitting it into these two groups. Logistic regression can then be used to predict this binary response, using polynomial functions of age as predictors.

- In other words, we fit the model

$$\Pr(y_i > 250|x_i) = \frac{\exp\left(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d\right)}{1 + \exp\left(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d\right)}$$

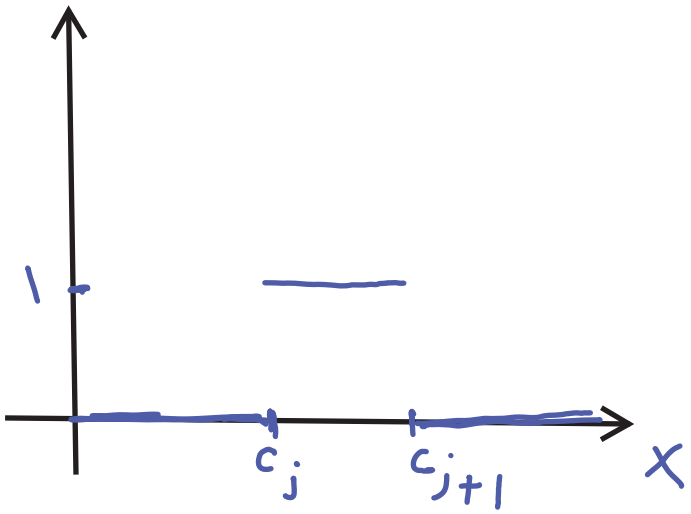
Step Functions

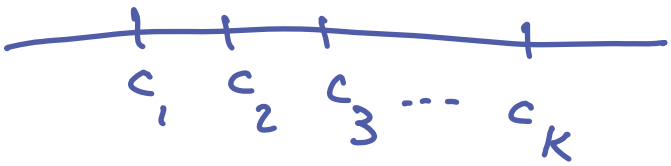
- Here we break the range of X into bins, and fit a different constant in each bin. This amounts to converting a continuous variable into an ordered categorical variable.
- In greater detail, we create cutpoints $c_1, c_2, \ldots, c_K$ in the range of X, and then construct $K + 1$ new variables

$$\begin{aligned}
C_0(X) &= I(X < c_1), \\
C_1(X) &= I(c_1 \leqslant X < c_2), \\
C_2(X) &= I(c_2 \leqslant X < c_3), \\
&\vdots \\
C_{K-1}(X) &= I(c_{K-1} \leqslant X < c_K), \\
C_K(X) &= I(c_K \leqslant X),
\end{aligned}$$

$I(\cdot)$ is an indicator function. For example

$$C_j(X) = I(c_j \leqslant X < c_{j+1}) = \begin{cases} 1 & c_j \leqslant X < c_{j+1} \\ 0 & \text{otherwise} \end{cases}$$





For any value of X,

$$C_0(X) + C_1(X) + \cdots + C_K(X) = 1,$$

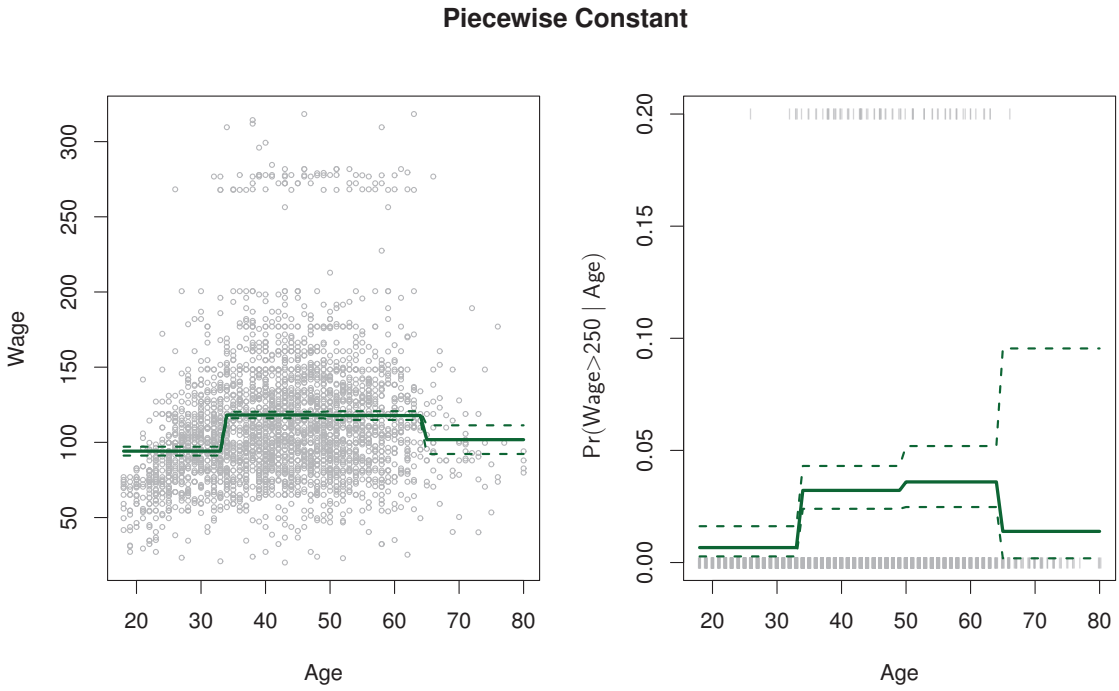
since X must be in exactly one of the $K+1$ intervals.

- We then use least squares to fit a linear model using $C_1(X), C_2(X), \ldots, C_K(X)$ as predictors

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \ldots + \beta_K C_K(x_i) + \epsilon_i.$$

For a given value of X, at most one of $C_1, C_2, \ldots, C_K$ can be non-zero.

- Note that when $X < c_1$, all of the predictors in the model are zero, so β_0 can be interpreted as the mean value of Y for $X < c_1$.
- By comparison, the model predicts a response of $\beta_0 + \beta_j$ for $c_j \leqslant X < c_{j+1}$, so β_j represents the average increase in the response for X in $c_j \leqslant X < c_{j+1}$ relative to $X < c_1$.



The Wage data. Left: The solid curve displays the fitted value from a least squares regression of wage (in thousands of dollars) using step functions of age. The dashed curves indicate an estimated 95% confidence interval. Right: We model the binary event wage > 250 using logistic regression, again using step functions of age. The fitted posterior probability of wage exceeding $250,000 is shown, along with an estimated 95% confidence interval.

we fit the logistic regression model

$$Pr(y_i > 250 | x_i) = \frac{e^{(\beta_0 + \beta_1 c_1(x) + \cdots + \beta_K c_K(x))}}{1 + e^{(\beta_0 + \beta_1 c_1(x) + \cdots + \beta_K c_K(x))}}$$

in order to predict the probability that an individual is a high earner on the basis of age.

- Unfortunately, unless there are natural breakpoints in the predictors, piecewise-constant functions can miss the action.

Performing Polynomial Regression and Step Functions in R

Here, we re-analyze the Wage data. We begin by loading the ISLR2 library, which contains the data.

```
> library(ISLR2)
> attach(Wage)
> names(Wage)
 [1] "year"       "age"        "maritl"     "race"       "education"
 [6] "region"     "jobclass"   "health"     "health_ins" "logwage"
[11] "wage"
> dim(Wage)
[1] 3000   11
```

Then, we fit the model using the following command:

```
> fit <- lm(wage ~ poly(age, 4, raw = T), data = Wage)
> coef(summary(fit))
                            Estimate   Std. Error   t value
(Intercept)            -1.841542e+02 6.004038e+01 -3.067172
poly(age, 4, raw = T)1  2.124552e+01 5.886748e+00  3.609042
poly(age, 4, raw = T)2 -5.638593e-01 2.061083e-01 -2.735743
poly(age, 4, raw = T)3  6.810688e-03 3.065931e-03  2.221409
poly(age, 4, raw = T)4 -3.203830e-05 1.641359e-05 -1.951938
                           Pr(>|t|)
(Intercept)            0.0021802539
poly(age, 4, raw = T)1 0.0003123618
poly(age, 4, raw = T)2 0.0062606446
poly(age, 4, raw = T)3 0.0263977518
poly(age, 4, raw = T)4 0.0510386498
```

We now create a grid of values for age at which we want predictions, and then call the generic predict() function, specifying that we want standard errors as well.

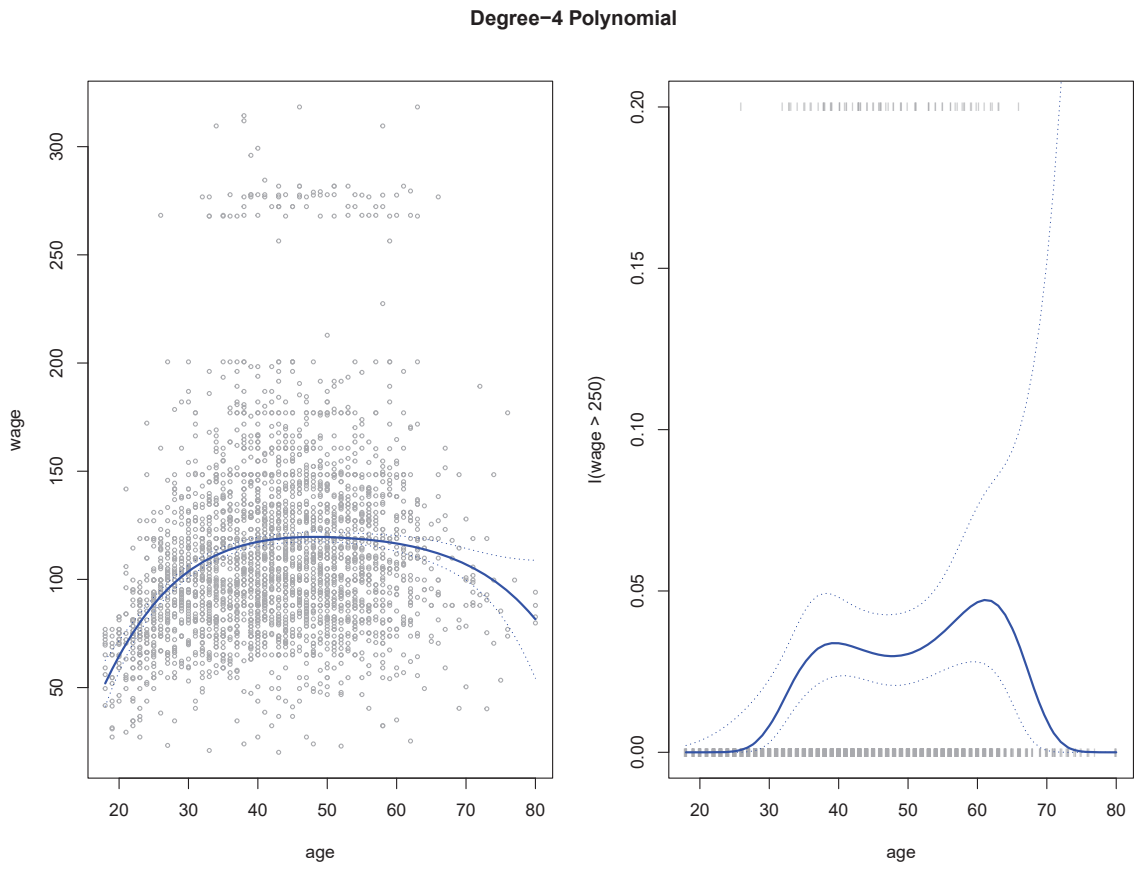
```
> agelims <- range(age)
> agelims
[1] 18 80
> age_grid <- seq(from = agelims[1], to = agelims[2])
```

```
> preds <- predict(fit, newdata = list(age = age_grid), se = TRUE)
> se_bands <- cbind(preds$fit + 2 * preds$se.fit,
    preds$fit - 2 * preds$se.fit)
```

Finally, we plot the data and add the fit from the degree-4 polynomial.

```
> par(mfrow = c(1, 2), mar = c(4.5, 4.5, 1, 1), oma = c(0, 0, 4, 0))
> plot(age, wage, xlim = agelims, cex = .5, col = "darkgrey")
> title("Degree-4 Polynomial", outer = T)
> lines(age_grid, preds$fit, lwd = 2, col = "blue")
> matlines(age_grid, se_bands, lwd = 1, col = "blue", lty = 3)
```

Here the mar and oma arguments to par() allow us to control the margins of the plot, and the title() function creates a figure title that spans both subplots.



In performing a polynomial regression we must decide on the degree of the polynomial to use. One way to do this is by using hypothesis tests. We now fit models ranging from linear to a degree-5 polynomial and seek to determine the simplest model which is sufficient to explain the relationship between wage and age.

We use the anova() function, which performs an analysis of variance (ANOVA, using an F-test) in order to test the null hypothesis that a model $\mathcal{M}_1$ is sufficient to explain the data against the alternative hypothesis that a more complex model $\mathcal{M}_2$ is required.

In order to use the anova() function, $\mathcal{M}_1$ and $\mathcal{M}_2$ must be nested models: the predictors in $\mathcal{M}_1$ must be a subset of the predictors in $\mathcal{M}_2$. In this case, we fit five different models and sequentially compare the simpler model to the more complex model.

```
> fit_1 <- lm(wage ~ age, data = Wage)
> fit_2 <- lm(wage ~ poly(age, 2, raw = T), data = Wage)
> fit_3 <- lm(wage ~ poly(age, 3, raw = T), data = Wage)
> fit_4 <- lm(wage ~ poly(age, 4, raw = T), data = Wage)
> fit_5 <- lm(wage ~ poly(age, 5, raw = T), data = Wage)
> anova(fit_1, fit_2, fit_3, fit_4, fit_5)
Analysis of Variance Table

Model 1: wage ~ age
Model 2: wage ~ poly(age, 2, raw = T)
Model 3: wage ~ poly(age, 3, raw = T)
Model 4: wage ~ poly(age, 4, raw = T)
Model 5: wage ~ poly(age, 5, raw = T)
  Res.Df     RSS Df Sum of Sq        F    Pr(>F)
1   2998 5022216
2   2997 4793430  1    228786 143.5931 < 2.2e-16 ***
3   2996 4777674  1     15756   9.8888  0.001679 **
4   2995 4771604  1      6070   3.8098  0.051046 .
5   2994 4770322  1      1283   0.8050  0.369682
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The p-value comparing the linear Model 1 to the quadratic Model 2 is essentially zero ($< 10^{-15}$), indicating that a linear fit is not sufficient. Similarly the p-value comparing the quadratic Model 2 to the cubic Model 3 is very low (0.0017), so the quadratic fit is also insufficient. The p-value comparing the cubic and degree-4 polynomials, Model 3 and Model 4, is approximately 5% while the degree-5 polynomial Model 5 seems unnecessary because its p-value is 0.37. Hence, either a cubic or a quartic polynomial appear to provide a reasonable fit to the data, but lower- or higher-order models are not justified.

The ANOVA method also works when we have other terms in the model as well. For example, we can use anova() to compare these three models:

```
> fit_1 <- lm(wage ~ education + age, data = Wage)
> fit_2 <- lm(wage ~ education + poly(age, 2, raw = T), data = Wage)
> fit_3 <- lm(wage ~ education + poly(age, 3, raw = T), data = Wage)
> anova(fit_1, fit_2, fit_3)
Analysis of Variance Table

Model 1: wage ~ education + age
Model 2: wage ~ education + poly(age, 2, raw = T)
Model 3: wage ~ education + poly(age, 3, raw = T)
  Res.Df     RSS Df Sum of Sq        F Pr(>F)
1   2994 3867992
2   2993 3725395  1    142597 114.6969 <2e-16 ***
3   2992 3719809  1      5587   4.4936 0.0341 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Next we consider the task of predicting whether an individual earns more than $250,000 per year. We proceed much as before, except that first we create the appropriate response vector, and then apply the glm() function using family = "binomial" in order to fit a polynomial logistic regression model. And once again, we make predictions using the predict() function.

```
> fit <- glm(I(wage > 250) ~ poly(age, 4), data = Wage, family = binomial)
> preds <- predict(fit, newdata = list(age = age_grid), se = T)
```

Here, we get predictions for the logit, or log-odds. That is, we have fit a model of the form

$$\log\left(\frac{Pr(Y=1|X)}{1-Pr(Y=1|X)}\right)=X\beta$$

and the predictions given are of the form of $X\hat{\beta}$. The standard errors given are also for $X\hat{\beta}$. In order to obtain confidence intervals for $Pr(Y=1|X)$, we use the transformation $Pr(Y=1|X)=\frac{e^{X\beta}}{1+e^{X\beta}}$.

```
> pfit <- exp(preds$fit) / (1 + exp(preds$fit))
> se_bands_logit <- cbind(preds$fit + 2 * preds$se.fit,
   preds$fit - 2 * preds$se.fit)
> se_bands <- exp(se_bands_logit) / (1 + exp(se_bands_logit))
```

Finally, we plot the results.

```
> plot(age, I(wage > 250), xlim = agelims, type = "n", ylim = c(0, .2))
> points(jitter(age), I(wage > 250) / 5, cex = .5, pch = "|", col = "darkgrey")
> lines(age_grid, pfit, lwd = 2, col = "blue")
> matlines(age_grid, se_bands, lwd = 1, col = "blue", lty = 3)
```

We have drawn the age values corresponding to the observations with wage values above 250 as gray marks on the top of the plot, and those with wage values below 250 are shown as gray marks on the bottom of the plot. We used the jitter() function to jitter the age values a bit so that observations with the same age value do not cover each other up. This is often called a rug plot.

In order to fit a step function, we use the cut() function.