ITERA

Modul 8 Praktikum
Statistika Sains Data

Regresi polynomial dan splines

Program Studi Sains Data
Fakultas Sains
Institut Teknologi Sumatera

2024

A. Tujuan Praktikum

1. Mahasiswa dapat menerapkan Non-linear model pada regresi polinomial
2. Mahasiswa dapat menerapkan model regresi splines
3. Mahasiswa dapat memahami pengambilan derajad polinom untuk model regresi non-linear

B. Teori Dasar

Regresi polinomial memperluas model linier dengan menambahkan prediktor, yang diperoleh dengan menaikkan pangkat masing-masing prediktor. Contoh regresi kubik : menggunakan tiga variabel, X,$X2$, dan $X3$ sebagai prediktor. Pendekatan ini menggunakan cara sederhana untuk menyajikan kecocokan nonlinier dengan data.

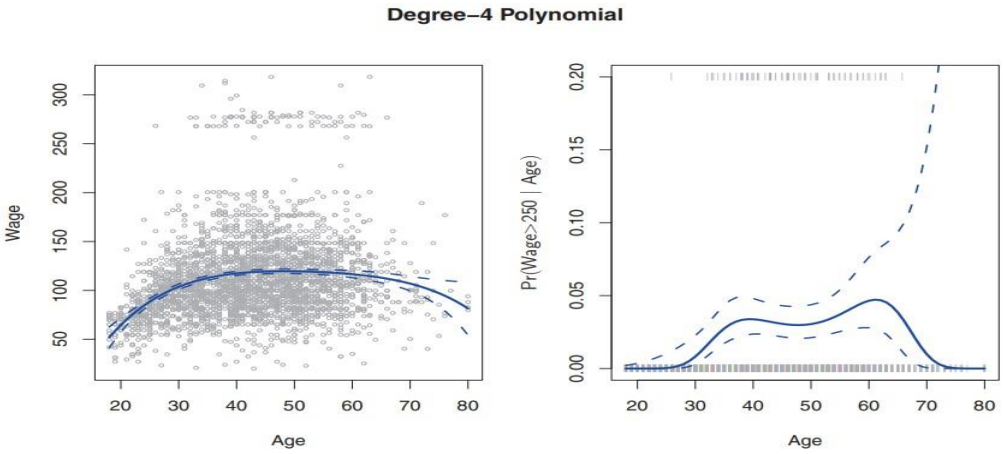
- Step functions memotong rentang variabel menjadi K wilayah yang berbeda untuk menghasilkan variabel kualitatif. Step functions memiliki efek menyesuaikan piecewise constant function.
- Regresi splines lebih fleksibel daripada polinomial dan step functions. Membagi rentang X menjadi K region yang berbeda. Dalam setiap region, fungsi polinomial sesuai dengan data. Namun, polinomial ini dibatasi sehingga bergabung dengan smooth di batas region atau simpul. Asalkan interval dibagi menjadi region yang cukup, ini dapat menghasilkan fitting yang sangat fleksibel.
- Smoothing spline serupa dengan regresi spline, tetapi muncul dalam situasi yang sedikit berbeda. Smoothing splines dihasilkan dari meminimalkan jumlah sisa kriteria kuadrat yang dikenakan penalti kehalusan.
- Local regression mirip dengan splines, tetapi berbeda dalam hal yang penting. Region dibiarkan overlap dan smooth.
- Generalized additive models memungkinkan untuk menangani banyak prediktor.

Polynomial Regression

Model linier standar $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$

Polynomial function $$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d + \epsilon_i, \qquad (7.1)$$

Untuk pangkat d yang cukup besar, regresi polinomial memungkinkan menghasilkan kurva yang sangat tidak linier. Koefisien pada (7.1) dapat dengan mudah diperkirakan menggunakan regresi linier kuadrat terkecil karena ini hanyalah model linier standar dengan prediktor x_i, x^2, x^3, …, x^d. Secara umum, tidak biasa menggunakan d lebih besar dari 3 atau 4 karena untuk nilai d yang besar, kurva polinomial dapat menjadi terlalu fleksibel dan dapat mengambil beberapa bentuk yang sangat aneh. Ini terutama benar di dekat batas variabel X.

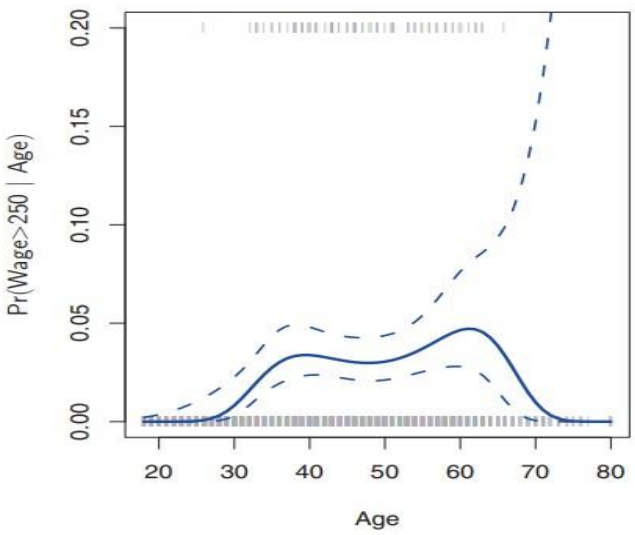


GAMBAR 7.1. Data Wage, which contains income and demographic information for males who reside in the central Atlantic region of the United States.
Kiri: Kurva biru solid adalah polinomial derajat-4 dari **wage** (dalam ribuan dolar) sebagai fungsi **age**, dicocokkan dengan kuadrat terkecil. Kurva putus-putus menunjukkan interval kepercayaan sekitar 95%.
Kanan: Model wage>250menggunakan regresi logistik, sekali lagi dengan polinomial derajat-4. Probabilitas posterior *fit* dari upah melebihi $250.000 ditunjukkan dengan warna biru, bersama dengan estimasi interval kepercayaan 95%.
Misalkan kita telah menghitung kecocokan pada nilai **age** tertentu, x0 :

$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \hat{\beta}_3 x_0^3 + \hat{\beta}_4 x_0^4. \qquad (7.2)$$

Least squares mengembalikan estimasi varians untuk masing-masing koefisien $\hat{\beta}_j$, serta kovarian antara pasangan estimasi koefisien. Estimasi standar error dari $\hat{f}(x_0)$ adalah akar kuadrat dari varians ini. Dua populasi yang berbeda: *high earners group* (berpenghasilan lebih dari $250.000 per tahun) dan *low earners group. Wage* sebagai variabel biner dengan membaginya menjadi dua kelompok ini. Regresi logistik kemudian dapat digunakan untuk memprediksi respons biner ini, dengan menggunakan fungsi polinomial *wage* sebagai prediktor. Modelnya

$$\Pr(y_i > 250|x_i) = \frac{\exp(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_d x_i^d)}{1 + \exp(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_d x_i^d)}. \qquad (7.3)$$



- Tanda abu-abu di bagian atas dan bawah panel menunjukkan *ages of the high earners* dan *the low earners*
- Kurva biru solid menunjukkan probabilitas yang pas untuk high earner, sebagai fungsi age.
- Perkiraan interval kepercayaan 95% juga ditampilkan (interval kepercayaan cukup lebar, terutama di sisi kanan). Ukuran sampel untuk kumpulan data ini sangat besar (n = 3.000), hanya ada 79 *high*, yang menghasilkan variasi yang tinggi dalam estimasi koefisien dan interval kepercayaan yang lebar.

Regression Splines

- Merupakan gabungan dari dua pendekatan : Regresi polynomial dan fungsi piecewise, pada titik potong
  (knot)s) $c_1, c_2, \ldots, c_k$
- Fungsi polynomial dari masing-masing sekatan diupayakan tergabung secara mulus (smooth)
- Fungsi basis pada regresi spline berordo M adalah dengan K buah knot adalah K+M-1 buah yang masing- masing adalah

$b_j(x_i) = x_i^j$, untuk $j = 1,2,\ldots,M-1$

$b_{M-1+k} = (x_i - c_k)_+^{M-1}$, untuk $k = 1,\ldots,K$

truncated power function

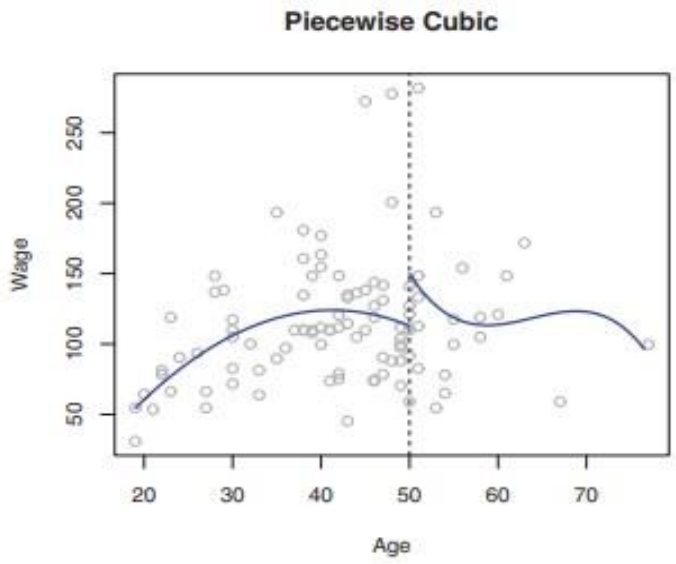
$$(x - c_j)_+^{M-1} = \begin{cases} (x - c_j)^{M-1} & , x > c_j \\ 0 & , \text{otherwise} \end{cases}$$

Regression Splines : Piecewise Polynomials

- Daerah peubah penjelas dibagi berdasarkan nilai titik potong (knot)
- Di masing-masing bagian dilakukan pemodelan regresi polynomial
- Misal, satu knot pada titik x=50, dan regresi kubik di masing-masing bagian

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon_i & \text{if } x_i < c; \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon_i & \text{if } x_i \geq c. \end{cases}$$

- Bentuk modelnya tidak kontinu pada titik knotnya à ini yang akan ditangani dengan regresi spline



- Piecewise polynomial regression melibatkan pemasangan polinomial berderajat rendah yang terpisah pada wilayah X yang berbeda.

  Misalnya, polinomial kubik bekerja dengan ftting model regresi kubik dari bentuk

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i, \qquad (7.8)$$

  di mana koefisien β0, β1, β2, dan β3 berbeda di berbagai bagian rentang X. Titik di mana koefisien berubah disebut knot.

  Contoh, sebuah piecewise cubic kubik tanpa knot hanyalah polinomial kubik standar, seperti pada (7.1) dengan d = 3. *Piecewise cubic polynomial* dengan single knot di titik c berbentuk

$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \epsilon_i & \text{if } x_i < c; \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \epsilon_i & \text{if } x_i \geq c. \end{cases}$$

  memasukkan dua fungsi polinomial yang berbeda ke dalam data, satu pada subhimpunan pengamatan dengan xi < c, dan satu pada subhimpunan pengamatan dengan xi ≥ c.

  Fungsi polinomial pertama memiliki koefisien β01, β11, β21, β31, dan yang kedua memiliki koefisien β02, β12, β22, β32. Masing-masing fungsi polinomial ini dapat difitting menggunakan *least squares* yang diterapkan pada fungsi sederhana dari prediktor aslinya.

C. Latihan Praktikum

Model Non-Linear

Pada praktikum modul 8 ini kita akan menerapkan dan memahami langkah dalam melakukan fitting dari model non linear yaitu regresi polinomial dan regresi splines. Sebagai percobaan kita menggunakan data 'Wage' dari package ISLR.

```
library(ISLR)
library(splines)
library(boot)
```

Pertama panggil dataset dari package

```
attach(Wage)
```

Pada Gambar di bawah ini menjelaskan observasi pada dataset yang mana education lebih mengecil pada variabel HS, dan ini mengartikan bahwa semakin naik pendapatan (Salary) maka education akan semakin tinggi.

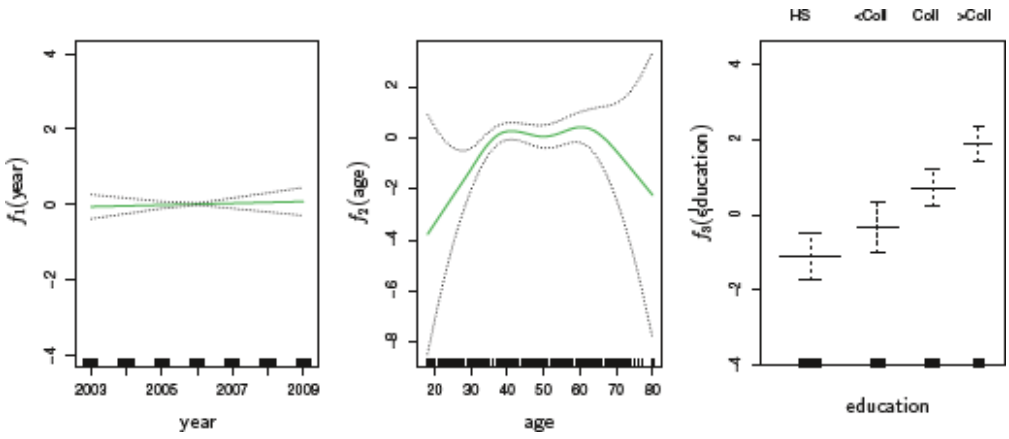


Figure 1: Gambar 1

Regresi Polynomial dan Step Functions

```
fit=lm(wage~poly(age,4), data=Wage)
coef(summary(fit))
```

```
##                 Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)    111.70361  0.7287409 153.283015 0.000000e+00
## poly(age, 4)1  447.06785 39.9147851  11.200558 1.484604e-28
## poly(age, 4)2 -478.31581 39.9147851 -11.983424 2.355831e-32
## poly(age, 4)3  125.52169 39.9147851   3.144742 1.678622e-03
## poly(age, 4)4  -77.91118 39.9147851  -1.951938 5.103865e-02
```

sintaks di atas menjelaskan bahwa model ini tetap menggunakan `lm()` function untuk memprediksi `wage`dengan menggunakan pangkat empat polinomial yaitu `age:poly:(age,4)`. Kemudian perintah dari poly() memperbolehkan kita untuk menghindari penulisan formula yang panjang. fungsi akan mengembalikan sebuah matriks yang mana kolom merupakan basis dari orthogonal polynomials. hal ini mengartikan bahwa setiap kolom adalah kombinasi linear dari variabel `age`, `age^2`, `age^3` dan `age^4`.

Bagaimanapun, kita dapat menggunakan fungsi `poly()` untuk menghitung langsung `age`, `age^2`, `age^3` dan `age^4`. kita juga dapat melakukan ini dengan memasukkan argumen `raw=TRUE` pada fungsi `poly()`. yang mana arguemn tadi dapat dilihat bahwa tidak mempengaruhi terhadap model yang signifikan.

```
fit2=lm(wage~poly(age,4,raw=T), data=Wage)
coef(summary(fit2))
```

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

Terdapat beberapa cara yang sama untuk melakukan fitting model, yang mana menunjukkan flexibilitas dari formula. contohnya:

```
fit2a=lm(wage~age+I(age^2)+I(age^3)+I(age^4),data=Wage)
coef(fit2a)
```

```
##   (Intercept)           age      I(age^2)      I(age^3)      I(age^4)
## -1.841542e+02  2.124552e+01 -5.638593e-01  6.810688e-03 -3.203830e-05
```

ini menunjukkan bahwa membuat polinomial dapat menggunakan fungsi wrapper I. adapun cara lainnya dapat menggunakan fungsi `cbind` untuk membuat matriks dari koleksi vektor. fungsi dapat digunakan sebagai wrapper.

```
fit2b=lm(wage~cbind(age,age^2,age^3,age^4), data=Wage)
coef(fit2a)
```

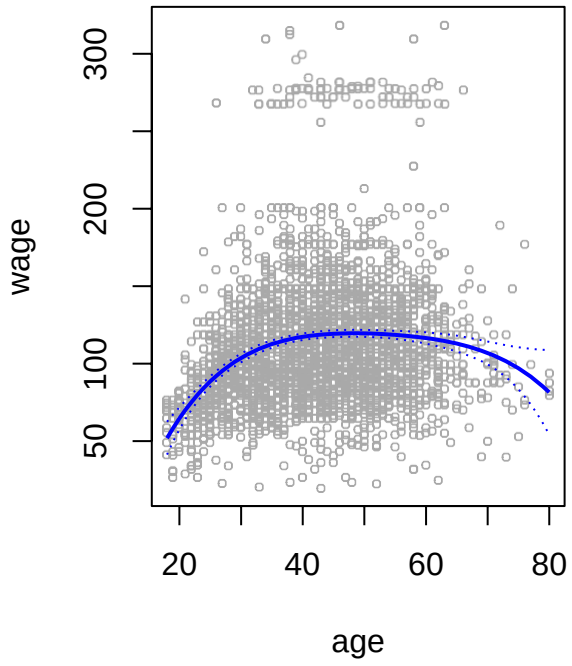
```
##   (Intercept)           age      I(age^2)      I(age^3)      I(age^4)
## -1.841542e+02  2.124552e+01 -5.638593e-01  6.810688e-03 -3.203830e-05
```

Selanjutnya kita membuat sebuah grid dari variabel age. untuk memprediksi model kita dapat menggunakan fungsi `predict()` dan juga dapat melakukan pengukuran eror terhadap model ini.

```
agelims=range(age)
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)
```

kemudian kita buat plot data dan menambahkan fit dari pangkat empat polinomial.

```r
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("Degre-4 Polynomial", outer=T)
lines(age.grid,preds$fit,lwd=2,col="blue")
matlines(age.grid, se.bands,lwd=1,col="blue",lty = 3)
```




```r
preds2=predict(fit2,newdata=list(age=age.grid),se=TRUE)
max(abs(preds$fit-preds2$fit))
```

```
## [1] 7.81597e-11
```

untuk menjalankan regresi polinomial, kita seharusnya memutuskan pangkat berapa yang akan kita gunakan. salah satu cara untuk menjawab pernyataan ini adalah melakukan hipotesis test. Pada model ini kita melakukan fitting model dengan jangkuan dari linear ke pangkat 5 polinomial, dan mencari model yang paling sederhana yang mana cukup menjelaskan hubungan dari `wage` dan `age`. Maka dari itu kita akan menggunakan *Analysis of Variance* (ANOVA, dengan menggunakan F-test) yang memerintahkan kita untuk melakukan tes null hipotesis bahwa model M_1 cukup untuk menjelaskan data terhadap hipotesis alternatif yang memerlukan kompleks model M_2. Untuk menggunakan fungsi anova(), M_1 dan M_2 harus merupakan model bertaut: prediktor di M1 harus merupakan subset dari prediktor di M2. Dalam hal ini, kita mencocokkan lima model berbeda dan secara berurutan membandingkan model yang lebih sederhana dengan model yang lebih kompleks.

```r
fit.1=lm(wage~age, data=Wage)
fit.2=lm(wage~poly(age,2),data=Wage)
fit.3=lm(wage~poly(age,3),data=Wage)
fit.4=lm(wage~poly(age,4),data=Wage)
fit.5=lm(wage~poly(age,5),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)
## Model 3: wage ~ poly(age, 3)
## Model 4: wage ~ poly(age, 4)
## Model 5: wage ~ poly(age, 5)
##   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
```

Membandingkan p-value dengan linear model 1 dengan kuadtratic model 2 yang nilanya menghampiri nol ($< 10^{-15}$) menunjukkan bahwa kecocokan linier tidak cukup. Demikian pula p-value yang membandingkan kuadrat model 2 dengan kubik model 3 yang sangat rendah (0,0017), sehingga kecocokan kuadrat juga tidak mencukupi. p-value yang membandingkan polinomial kubik dan derajat-4, Model 3 dan Model 4, kira-kira 5% sedangkan polinomial derajat-5 Model 5 tampaknya tidak diperlukan karena nilai-pnya adalah 0,37. Oleh karena itu, baik polinomial kubik atau kuartik tampaknya memberikan kecocokan yang masuk akal dengan data, tetapi model orde rendah atau tinggi tidak dibenarkan.

Dalam hal ini, alih-alih menggunakan fungsi anova(), kita dapat memperoleh p-value ini secara lebih ringkas dengan mengeksploitasi fakta bahwa poli() membuat polinomial ortogonal.

```
coef(summary(fit.5))
```

```
##                Estimate Std. Error     t value     Pr(>|t|)
## (Intercept)    111.70361  0.7287647 153.2780243 0.000000e+00
## poly(age, 5)1  447.06785 39.9160847  11.2001930 1.491111e-28
## poly(age, 5)2 -478.31581 39.9160847 -11.9830341 2.367734e-32
## poly(age, 5)3  125.52169 39.9160847   3.1446392 1.679213e-03
## poly(age, 5)4  -77.91118 39.9160847  -1.9518743 5.104623e-02
## poly(age, 5)5  -35.81289 39.9160847  -0.8972045 3.696820e-01
```

Perhatikan bahwa p-value adalah sama, dan faktanya kuadrat t-statistic sama dengan F-statistic dari fungsi anova(); Misalnya:

```
(-11.983)^2
```

```
## [1] 143.5923
```

Namun, metode ANOVA berfungsi apakah kita menggunakan polinomial ortogonal atau tidak; dan juga berfungsi ketika kita memiliki istilah lain dalam model juga. Misalnya, kita dapat menggunakan anova() untuk membandingkan ketiga model ini:

```
fit.1=lm(wage~education+age, data=Wage)
fit.2=lm(wage~education+poly(age,2), data=Wage)
fit.3=lm(wage~education+poly(age,3), 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)
## Model 3: wage ~ education + poly(age, 3)
##   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
```

Sebagai alternatif untuk menggunakan uji hipotesis dan ANOVA, kita dapat memilih derajat polinomial menggunakan validasi silang, seperti yang dibahas di Modul 5. Selanjutnya kita mempertimbangkan tugas memprediksi apakah seseorang berpenghasilan lebih dari $250.000 per tahun.kecuali bahwa pertama kita membuat vektor respons yang sesuai, dan kemudian menerapkan fungsi glm() menggunakan `family="binomial"` agar sesuai dengan model regresi logistik polinomial.

```
fit=glm(I(wage>250)~poly(age,4),data=Wage, family=binomial)
```

Perhatikan bahwa kita kembali menggunakan pembungkus I() untuk membuat variabel respons biner ini dengan cepat. Ekspresi Wage>250 dievaluasi menjadi variabel logis yang berisi BENAR dan SALAH, yang glm() memaksa ke biner dengan menyetel BENAR ke 1 dan FALSE ke 0. Sekali lagi, kita membuat prediksi menggunakan fungsi predict().

```
preds=predict(fit,newdata=list(age=age.grid),se=T)
```

Namun, menghitung interval kepercayaan sedikit terlibat daripada dalam kasus regresi linier. Jenis prediksi default untuk model glm() adalah `type="link"`, yang kita gunakan di sini. hal ini berarti kita mendapatkan prediksi untuk logit: yaitu sebagai berikut:

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

dan prediksi yang diberikan berbentuk $X\beta$. Kesalahan standar yang diberikan juga dalam bentuk ini. Untuk mendapatkan interval kepercayaan untuk $Pr(Y = 1|X)$, kita menggunakan transformasi sebagai berikut:

$$Pr(Y = 1|X) = \frac{\exp(X\beta)}{1 + \exp(X\beta)}$$

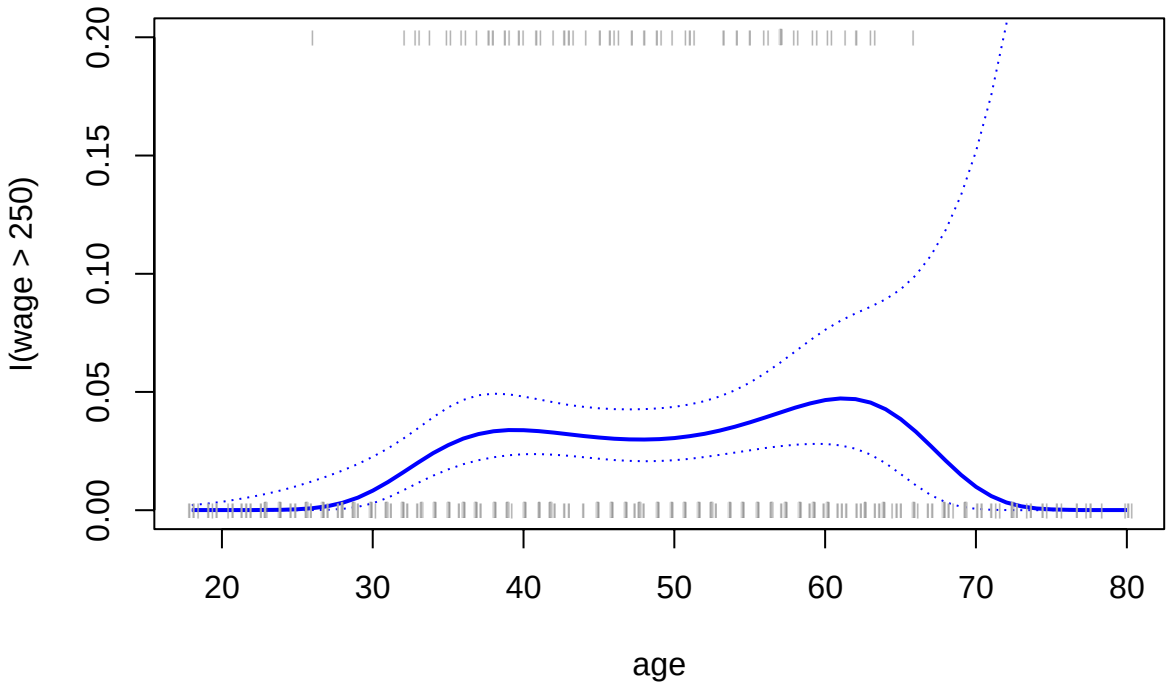
```
pfit=exp(preds$fit)/(1+exp(preds$fit))
se.bands.logit=cbind(preds$fit+2*preds$se.fit, preds$fit-2*preds$se.fit)
se.bands=exp(se.bands.logit)/(1+exp(se.bands.logit))
```

Kita juga dapat langsung melakukan komputasi probabiliti dengan memilih pilihan `type=response` di dalam fungsi `predict()`

```
preds=predict(fit,newdata=list(age=age.grid),type="response",   se=T)
```

Namun, interval kepercayaan yang sesuai tidak masuk akal karena kita akan berakhir dengan probabilitas negatif! Akhirnya, plot sebelah kanan dari Gambar di bawah ini dibuat sebagai berikut:

```
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)
```



Kita telah menggambar nilai usia yang sesuai dengan pengamatan dengan nilai upah di atas 250 sebagai tanda abu-abu di bagian atas plot, dan nilai upah di bawah 250 ditampilkan sebagai tanda abu-abu di bagian bawah plot. Kita menggunakan fungsi jitter() untuk sedikit mengubah nilai usia sehingga pengamatan dengan nilai usia yang sama tidak saling menutupi. Ini sering disebut *rug plot*.

kemudian untuk melakukan fitting step function, maka kita dapat menggunakan fungsi `cut()`

```
table(cut(age, 4))
```

```
##
## (17.9,33.5]   (33.5,49]   (49,64.5] (64.5,80.1]
##         750         1399          779           72
```

```
fit=lm(wage~cut(age,4),data=Wage)
coef(summary(fit))
```

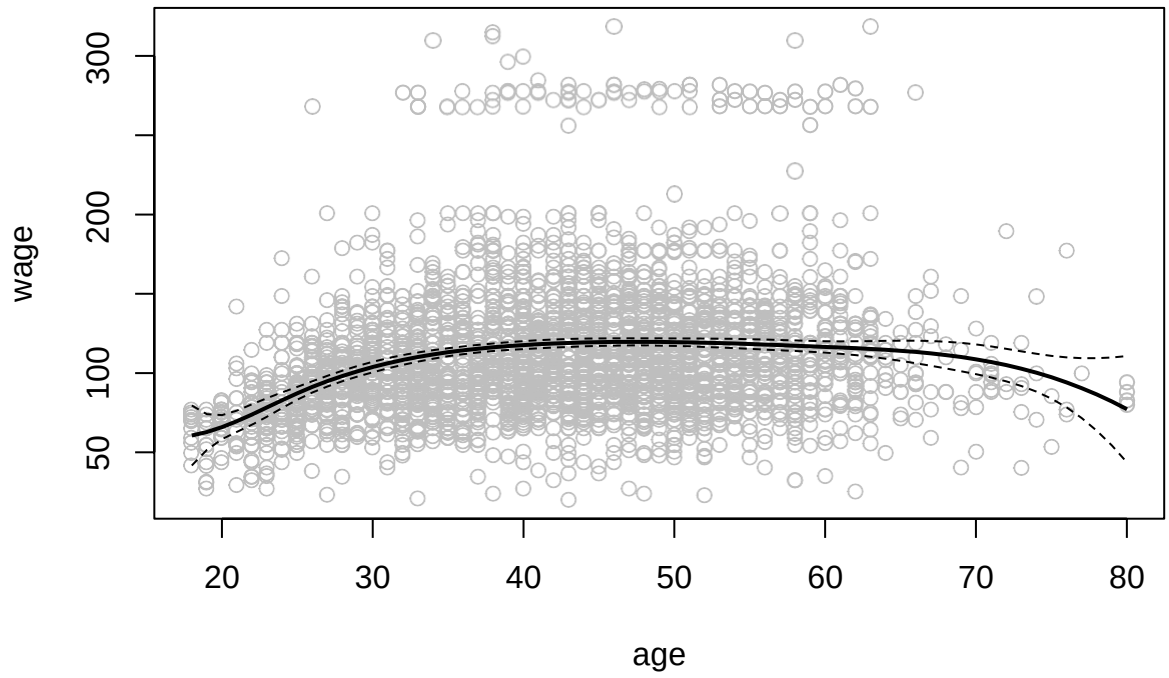
```
##                         Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)            94.158392   1.476069 63.789970 0.000000e+00
## cut(age, 4)(33.5,49]   24.053491   1.829431 13.148074 1.982315e-38
## cut(age, 4)(49,64.5]   23.664559   2.067958 11.443444 1.040750e-29
## cut(age, 4)(64.5,80.1]  7.640592   4.987424  1.531972 1.256350e-01
```

Di sini cut() secara otomatis memilih cutpoint pada usia 33,5; 49; dan 64,5 tahun. Kita juga bisa menentukan cutpoint kita sendiri secara langsung menggunakan opsi break. Fungsi cut() mengembalikan variabel kategori yang diurutkan; fungsi lm() kemudian membuat satu set variabel dummy untuk digunakan dalam regresi. Kategori usia <33,5 tidak dimasukkan, sehingga koefisien intersep sebesar $94.160 dapat diartikan sebagai gaji rata-rata untuk mereka yang berusia di bawah 33,5 tahun, dan koefisien lainnya dapat diartikan sebagai gaji tambahan rata-rata untuk kelompok usia lainnya. Kita dapat menghasilkan prediksi dan plot seperti yang dilakukan dalam kasus kecocokan polinomial

Splines

kita melihat bahwa splines regresi dapat disesuaikan dengan membuat matriks fungsi basis yang sesuai. Fungsi `bs()` menghasilkan seluruh matriks fungsi dasar untuk splines dengan kumpulan simpul yang ditentukan. Secara default, splines kubik juga diproduksi. Dengan melakukan fitting pada dataset wage dengan age menggunakan spline regresi sederhana adalah sebagai berikut (gunakan package splines):

```
fit=lm(wage~bs(age, knots=c(25, 40, 60)), data=Wage)
pred=predict(fit, newdata=list(age=age.grid), se=T)
plot(age, wage, col="gray")
lines(age.grid, pred$fit, lwd=2)
lines(age.grid, pred$fit+2*pred$se, lty="dashed")
lines(age.grid, pred$fit-2*pred$se, lty="dashed")
```



Di sini kita telah menentukan simpul pada usia 25, 40, dan 60. dan menghasilkan spline dengan enam fungsi dasar. (Ingat bahwa spline kubik dengan tiga simpul memiliki tujuh derajat kebebasan; derajat kebebasan ini digunakan oleh sebuah intersep, ditambah fungsi enam basis.) Kita juga dapat menggunakan opsi df untuk menghasilkan spline dengan simpul pada kuantil seragam dari data.

```
dim(bs(age, knots=c(25, 40, 60)))
```

```
## [1] 3000    6
```

```
dim(bs(age, df=6))
```

```
## [1] 3000    6
```

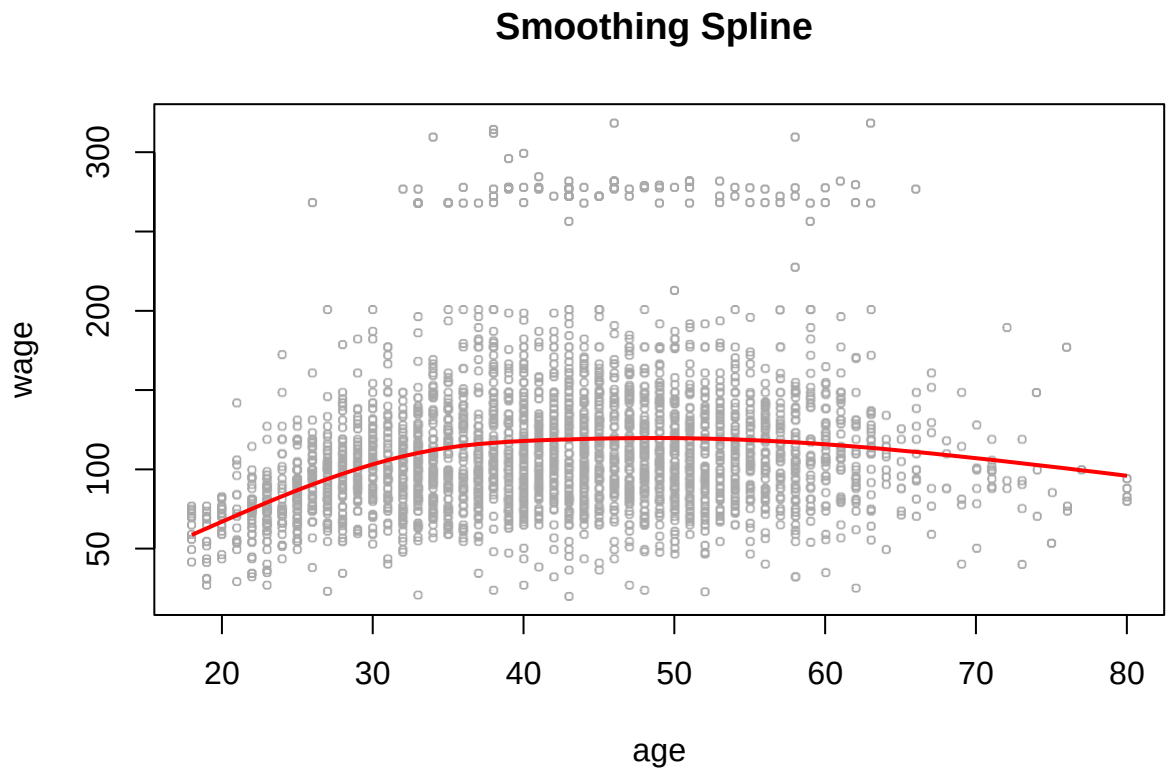
```
attr(bs(age,df=6),"knots")
```

```
##    25%    50%    75%
```

```
## 33.75 42.00 51.00
```

Dalam hal ini R memilih simpul pada usia 33,8; 42,0; dan 51,0; yang sesuai dengan persentil usia ke-25, ke-50, dan ke-75. Fungsi `bs()` juga memiliki argumen derajat, jadi kita dapat menyesuaikan spline dengan derajat apa pun, daripada derajat default 3 (yang menghasilkan spline kubik). Agar sesuai dengan spline alami, kita menggunakan fungsi ns(). Di sini kita memasang spline alami dengan empat derajat kebebasan.

```
fit2=lm(wage~ns(age, df=4), data=Wage)
pred2=predict(fit2, newdata=list(age=age.grid), se=T)
plot(age ,wage ,xlim=agelims ,cex =.5, col = "darkgrey")
lines(age.grid, pred2$fit, col="red", lwd=2)
title ("Smoothing Spline ")
```



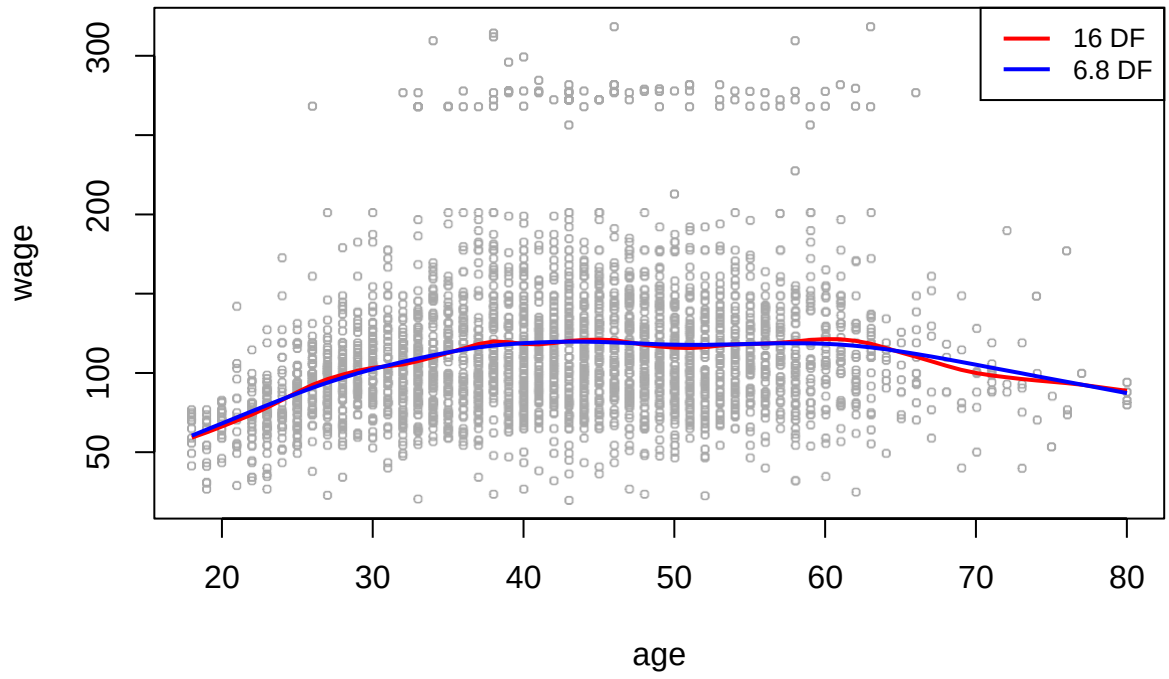

```
fit=smooth.spline(age ,wage ,df =16)
fit2=smooth.spline(age ,wage ,cv=TRUE)
```

```
## Warning in smooth.spline(age, wage, cv = TRUE): cross-validation with##
non-unique "x" values seems doubtful
```

```
fit2$df
```

```
## [1] 6.794596
```

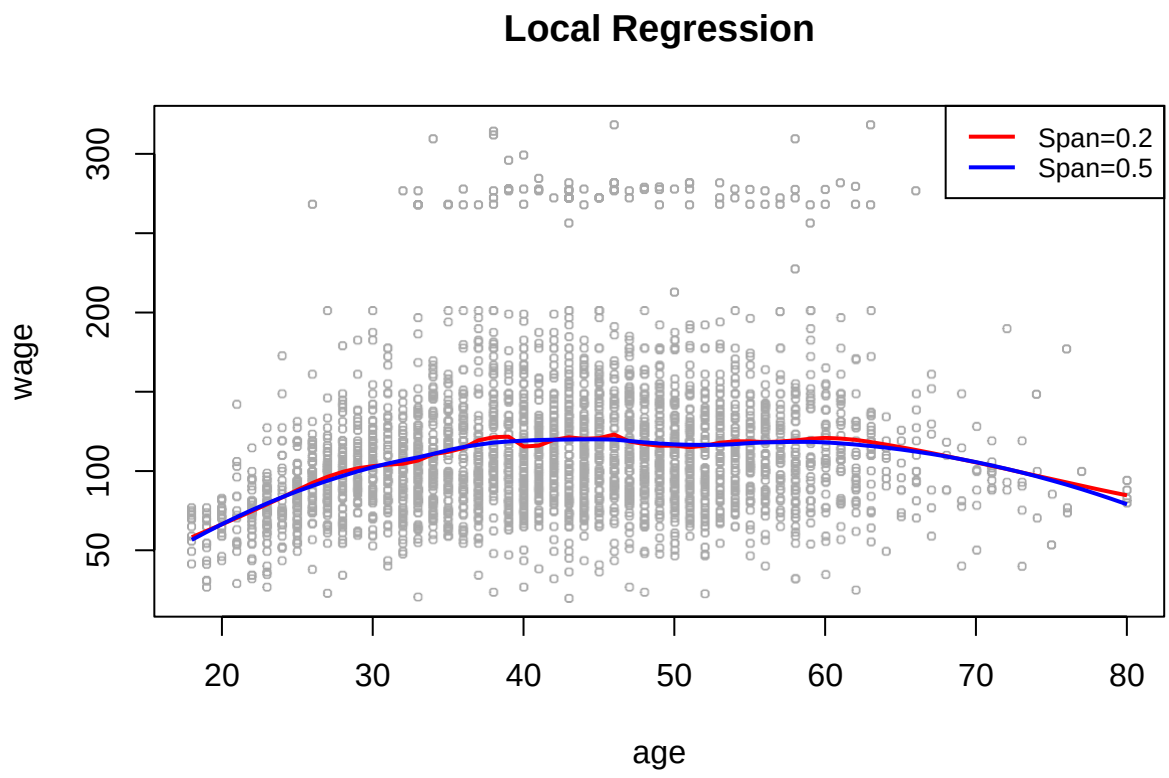
```
plot(age ,wage ,xlim=agelims ,cex =.5, col = "darkgrey")
lines(fit ,col ="red", lwd =2)
lines(fit2 ,col = "blue", lwd =2)
legend ("topright", legend =c("16 DF" ,"6.8 DF"),
        col=c("red","blue"), lty =1, lwd =2, cex =.8)
```



Perhatikan bahwa dalam panggilan pertama ke `smooth.spline()`, kita menetapkan df=16. Fungsi tersebut kemudian menentukan nilai lambda mana yang menghasilkan 16 derajat kebebasan. Dalam panggilan kedua ke `smooth.spline()`, kita memilih tingkat kehalusan dengan cross validation; dan menghasilkan nilai lambda yang menghasilkan 6,8 derajat kebebasan.

Untuk melakukan regresi lokal, kita menggunakan fungsi `loess()`

```
plot(age ,wage ,xlim=agelims ,cex =.5, col ="darkgrey")
title (" Local Regression ")
fit=loess(wage~age, span =.2, data=Wage)
fit2=loess(wage~age ,span =.5, data=Wage)
lines(age.grid ,predict (fit ,data.frame(age=age.grid)),
      col ="red",lwd =2)
lines(age.grid ,predict (fit2,data.frame(age=age.grid)),
      col ="blue",lwd =2)
legend("topright",legend=c("Span=0.2" ,"Span=0.5"),
       col=c("red","blue"),lty =1, lwd =2, cex =.8)
```



Di sini kita telah melakukan regresi linier lokal menggunakan rentang 0,2 dan 0,5: yaitu, setiap lingkungan terdiri dari 20% atau 50% dari pengamatan. Semakin besar rentangnya, semakin halus pasnya. Pustaka `locfit` juga dapat digunakan untuk menyesuaikan model regresi lokal di R.

Cross Validation untuk regresi polinomial

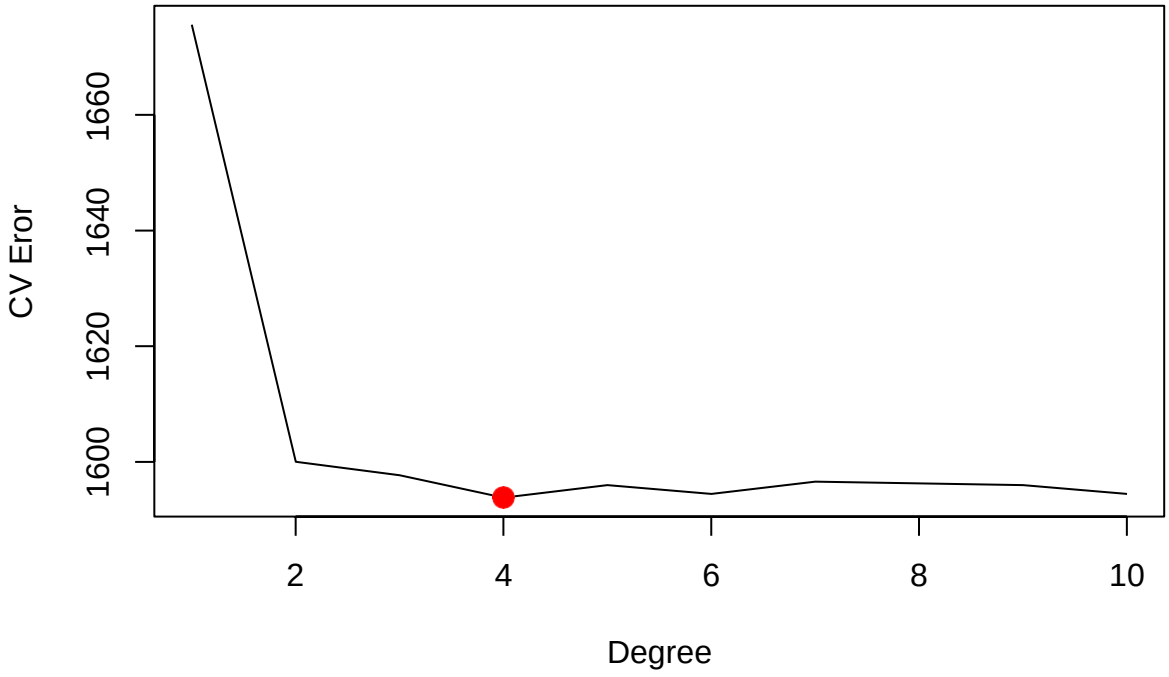
Berikut langkah untuk melakukan fitting regresi polinomial untuk memprediksi variabel wage menggunakan variabel age. Dengan menggunakan cross validation untuk memilih derajat optimal untuk polinomial.

```
set.seed(702)
deltas = rep(NA, 10)
for (i in 1:10) {
  glm.fit = glm(wage~poly(age, i), data=Wage)
  deltas[i] = cv.glm(Wage, glm.fit, K=10)$delta[1]
}
```

```
deltas
```

```
##  [1] 1675.587 1600.009 1597.688 1593.817 1595.988 1594.466 1596.577 1596.290
##  [9] 1595.993 1594.470
```

```
plot(1:10, deltas, xlab = "Degree", ylab = "CV Eror", type = "l")
d.min <- which.min(deltas)
points(which.min(deltas), deltas[which.min(deltas)], col = "red", cex = 2, pch = 20)
```



derajat,d=4 merupakan derajat optimal untuk polinomial yang dipilih oleh cross validation.

Kita menggunakan fungsi anova() untuk melakukan test null hypothesis model M_1 cukup untuk menjelaskan data terhadap hipotesis altenatif bahwa model yang lebih kompleks M_2 diperlukan.

Kita gunakan 10 derajat polinomial untuk ages

```
lm1 <- lm(wage ~ age, data = Wage)
lm2 <- lm(wage ~ poly(age, 2), data = Wage)
lm3 <- lm(wage ~ poly(age, 3), data = Wage)
lm4 <- lm(wage ~ poly(age, 4), data = Wage)
lm5 <- lm(wage ~ poly(age, 5), data = Wage)
```

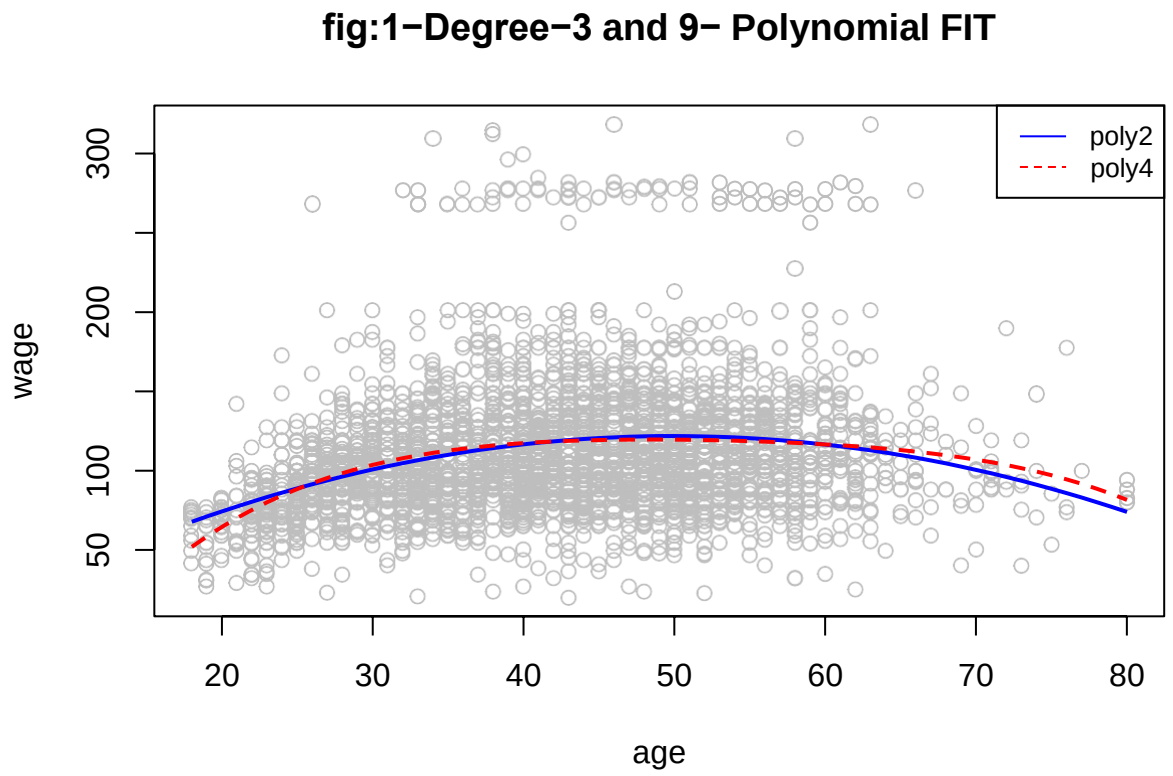
```
anova(lm1, lm2, lm3, lm4, lm5)
```

```
## Analysis of Variance Table
##
## Model 1: wage ~ age
## Model 2: wage ~ poly(age, 2)
## Model 3: wage ~ poly(age, 3)
## Model 4: wage ~ poly(age, 4)
## Model 5: wage ~ poly(age, 5)
##   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
```

Anova, P-value, menunjukkan bahwa regresi derajat polinomial tingkat 2, 3 dan 4 memberikan kesesuaian yang wajar dengan data. Diantaranya Polinomial urutan ke-2 memberikan kesesuaian terbaik untuk data. Dimana sebagai metode validasi silang memberikan kecocokan terbaik pada derajat polinominal 4.

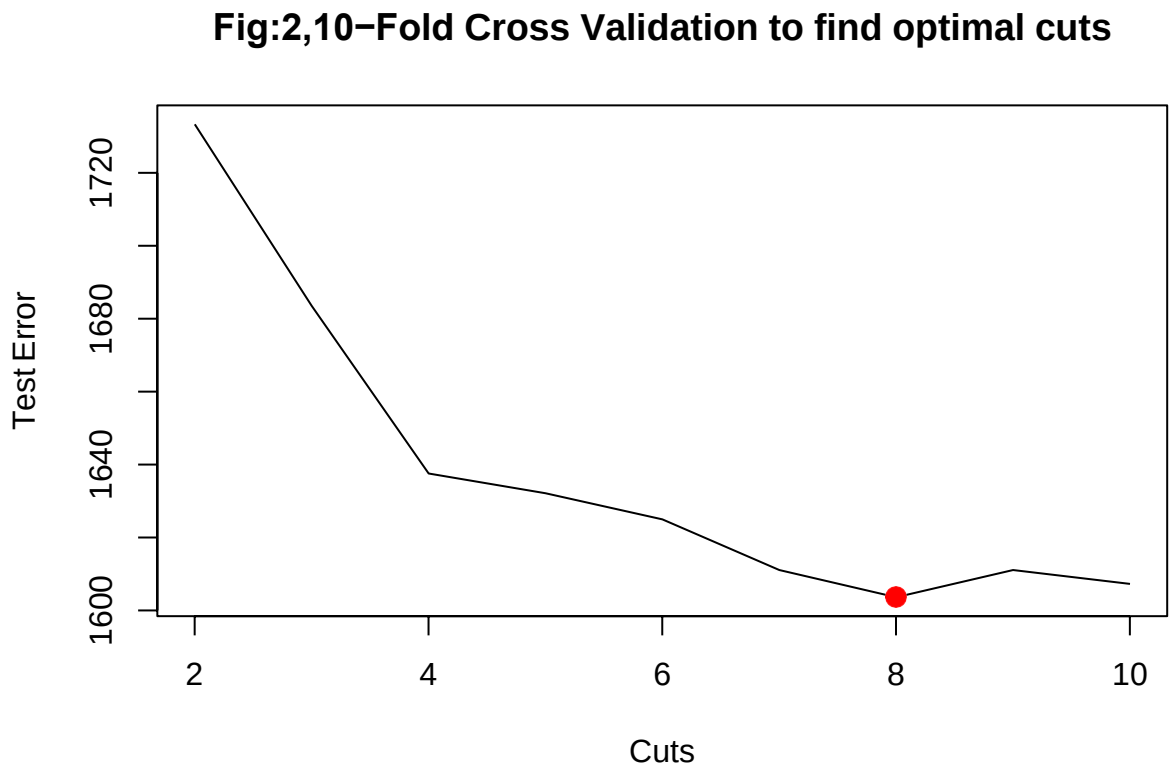
```
plot(wage ~ age, data = Wage, col = "gray")
agelims <- range(Wage$age)
age.grid <- seq(from = agelims[1], to = agelims[2])
poly2 <- lm(wage ~ poly(age, 2), data = Wage)
preds <- predict(poly2, newdata = list(age = age.grid))
lines(age.grid, preds, col = "blue", lwd = 2, lty = 1)
poly4 <- lm(wage ~ poly(age, 4), data = Wage)
preds4 <- predict(poly4, newdata = list(age = age.grid))
lines(age.grid, preds4, col = "red", lwd = 2, lty=2)
legend("topright", legend=c("poly2", "poly4"),
              col=c("blue", "red"), lty=1:2, cex=0.8)
title("fig:1-Degree-3 and 9- Polynomial FIT",outer=FALSE)
```



Fitting step function untuk memprediksi wage menggunakan age, dan lakukan cross validation untuk memilih jumlah pemotongan yang optimal. Buat plot sesuai yang diperoleh

```
set.seed(702)
cvs <- rep(NA, 10)
for (i in 2:10) {
    Wage$age.cut <- cut(Wage$age, i)
    fit <- glm(wage ~ age.cut, data = Wage)
    cvs[i] <- cv.glm(Wage, fit, K = 10)$delta[1]
```

```
}
plot(2:10, cvs[-1], xlab = "Cuts", ylab = "Test Error", type = "l")
d.min <- which.min(cvs)
points(which.min(cvs), cvs[which.min(cvs)], col = "red", cex = 2, pch = 20)
title("Fig:2,10-Fold Cross Validation to find optimal cuts",outer=FALSE)
```



Cross Validation menunjukkan bahwa kesalahan uji minimum untuk pemotongan optimal adalah k=8 pemotongan. Kemudian training data dengan potongan, k=8

```
lm.fit = glm(wage~cut(age, 8), data=Wage)
agelims = range(Wage$age)
age.grid = seq(from=agelims[1], to=agelims[2])
lm.pred = predict(lm.fit, data.frame(age=age.grid))
plot(wage~age, data=Wage, col="gray")
lines(age.grid, lm.pred, col="darkgreen", lwd=2)
title("Fig:3, Scatter plot of data with fitted line at cuts=8",outer=FALSE)
```

Fig:3, Scatter plot of data with fitted line at cuts=8