Extensiones del modelo lineal: regresores polinómicos

Contents

1 Datos **1**

2 Regresión polinómica **3**
- 2.1 Introducción 3
- 2.2 Modelo 4
- 2.3 Selección del grado máximo del polinomio 7

3 Polinomios ortogonales **9**
- 3.1 Definición del modelo 9
- 3.2 Propiedades 10

1 Datos

Datos: Wage

Wage and other data for a group of 3000 male workers in the Mid-Atlantic region.

```
d = read.csv("datos/Wage.csv")
str(d)
```

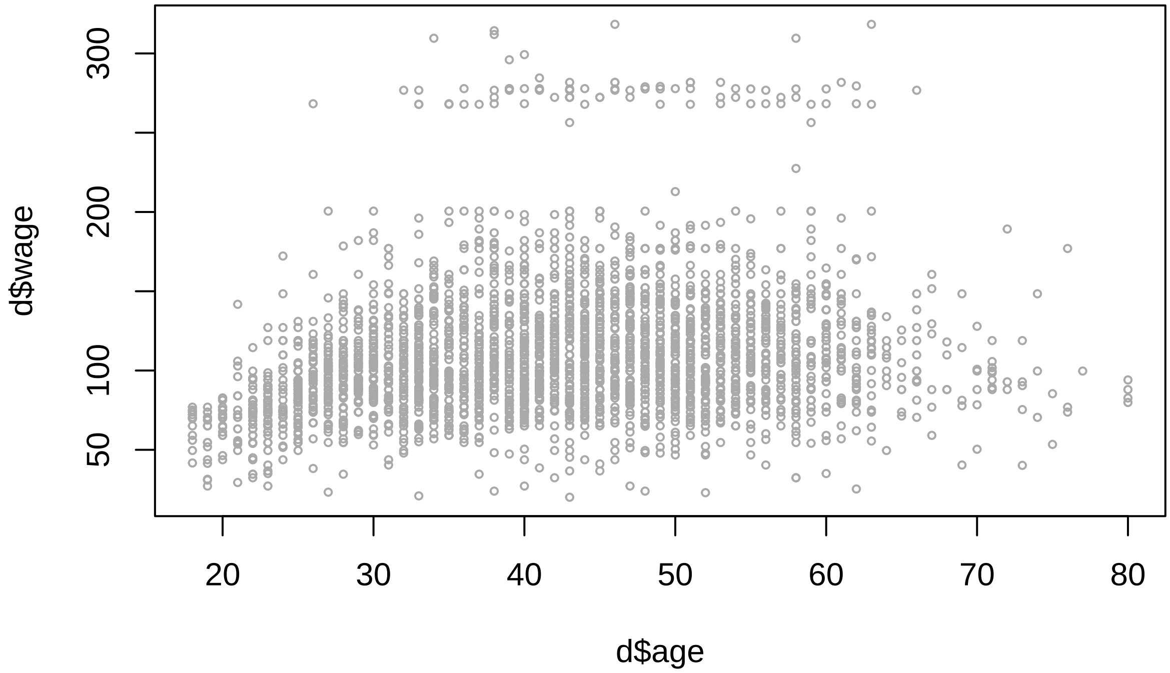
```
## 'data.frame':    3000 obs. of  12 variables:
##  $ X         : int  231655 86582 161300 155159 11443 376662 450601 377954 228963 81404 ...
##  $ year      : int  2006 2004 2003 2003 2005 2008 2009 2008 2006 2004 ...
##  $ age       : int  18 24 45 43 50 54 44 30 41 52 ...
##  $ maritl    : chr  "1. Never Married" "1. Never Married" "2. Married" "2. Married" ...
##  $ race      : chr  "1. White" "1. White" "1. White" "3. Asian" ...
##  $ education : chr  "1. < HS Grad" "4. College Grad" "3. Some College" "4. College Grad" ...
##  $ region    : chr  "2. Middle Atlantic" "2. Middle Atlantic" "2. Middle Atlantic" "2. Middle Atlant
##  $ jobclass  : chr  "1. Industrial" "2. Information" "1. Industrial" "2. Information" ...
##  $ health    : chr  "1. <=Good" "2. >=Very Good" "1. <=Good" "2. >=Very Good" ...
##  $ health_ins: chr  "2. No" "2. No" "1. Yes" "1. Yes" ...
##  $ logwage   : num  4.32 4.26 4.88 5.04 4.32 ...
##  $ wage      : num  75 70.5 131 154.7 75 ...
```

A data frame with 3000 observations on the following 11 variables:

- year: Year that wage information was recorded
- age: Age of worker
- maritl: A factor with levels 1. Never Married 2. Married 3. Widowed 4. Divorced and 5. Separated indicating marital status
- race: A factor with levels 1. White 2. Black 3. Asian and 4. Other indicating race
- education: A factor with levels 1. < HS Grad 2. HS Grad 3. Some College 4. College Grad and 5. Advanced Degree indicating education level
- region: Region of the country (mid-atlantic only)
- jobclass: A factor with levels 1. Industrial and 2. Information indicating type of job
- health: A factor with levels 1. <=Good and 2. >=Very Good indicating health level of worker

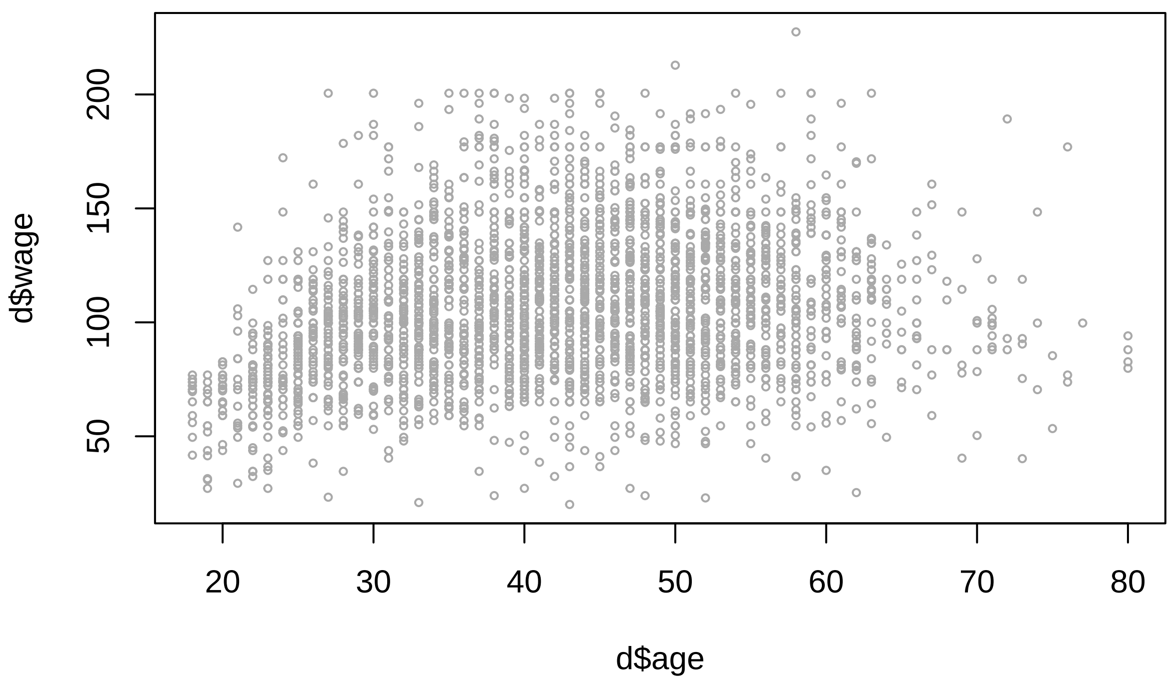
- health_ins: A factor with levels 1. Yes and 2. No indicating whether worker has health insurance
- logwage: Log of workers wage
- wage: Workers raw wage

```
plot(d$age,d$wage, cex = 0.5, col = "darkgrey")
```



Parece que hay dos grupos diferenciados: los que ganan más de 250.000$ y los que ganan menos. Vamos a trabajar con los que ganan menos

```
d = d[d$wage<250,]
plot(d$age,d$wage, cex = 0.5, col = "darkgrey")
```



2 Regresión polinómica

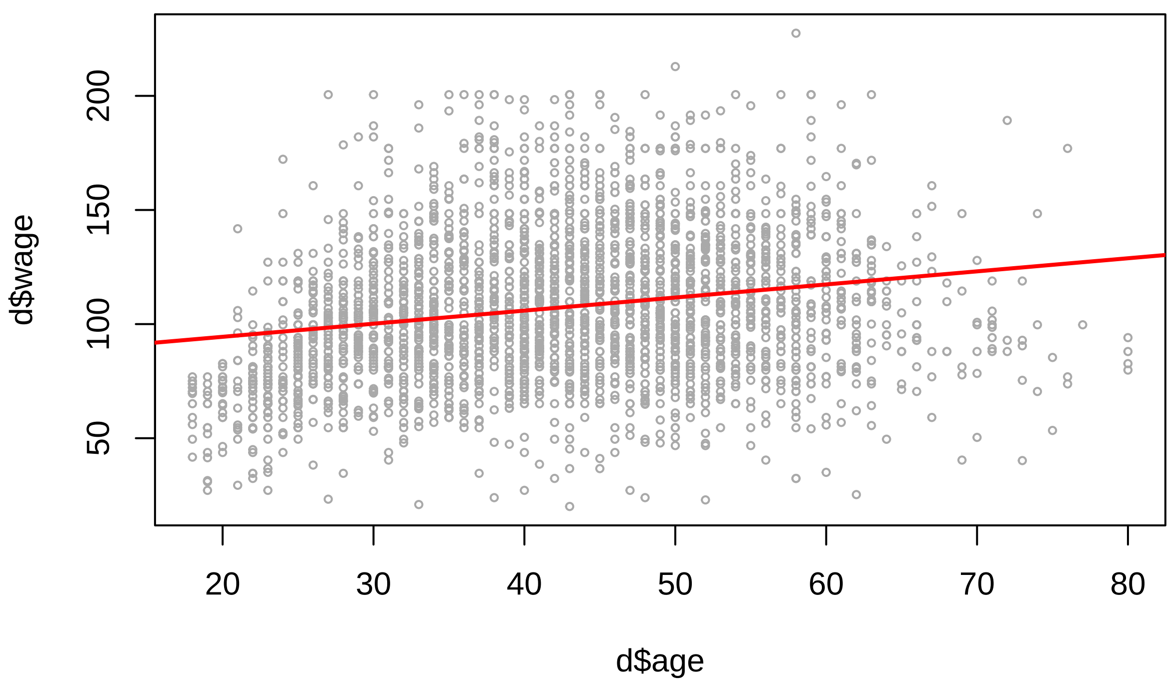
2.1 Introducción

Utilizar una recta no es satisfactorio

```
m0 = lm(wage ~ age, data = d)
summary(m0)
```

```
##
## Call:
## lm(formula = wage ~ age, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -93.23 -21.68  -2.70  18.72 111.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 82.95877    2.18977   37.88   <2e-16 ***
## age          0.57355    0.04993   11.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.25 on 2919 degrees of freedom
## Multiple R-squared:  0.04324,    Adjusted R-squared:  0.04292
## F-statistic: 131.9 on 1 and 2919 DF,  p-value: < 2.2e-16
```

```
plot(d$age,d$wage, cex = 0.5, col = "darkgrey")
abline(m0, col = "red", lwd = 2)
```



No se ajusta bien a los datos por lo que el R^2 es pequeño.

2.2 Modelo

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_k x_i^k + u_i$$

- el modelo se estima con mínimos cuadrados, utilizando como regresores: $x_i, x_i^2, x_i^3, \cdots, x_i^k$.
- todas las cuestiones de inferencia estudiadas en el tema de regresión lineal son válidas aquí también.

Hay varias maneras de implementarlos en R:

- Con la función $I()$:

```
m1 = lm(wage ~ age + I(age^2) + I(age^3) + I(age^4), data = d)
summary(m1)
```

```
##
## Call:
## lm(formula = wage ~ age + I(age^2) + I(age^3) + I(age^4), data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -93.565 -20.689  -2.015  17.584 116.228
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.620e+02  4.563e+01  -3.550 0.000391 ***
## age          1.948e+01  4.477e+00   4.350 1.41e-05 ***
## I(age^2)    -5.150e-01  1.569e-01  -3.283 0.001039 **
## I(age^3)     6.113e-03  2.334e-03   2.619 0.008869 **
## I(age^4)    -2.800e-05  1.250e-05  -2.240 0.025186 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.16 on 2916 degrees of freedom
## Multiple R-squared:  0.1094, Adjusted R-squared:  0.1082
## F-statistic: 89.55 on 4 and 2916 DF,  p-value: < 2.2e-16
```

Para hacer predicciones con este modelo, por ejemplo, para *age = 29*:

```
age = 29
xp = data.frame(age, I(age^2), I(age^3), I(age^4))
predict(m1, newdata = xp, interval = "confidence")
```

```
##        fit    lwr      upr
## 1 99.03991 96.883 101.1968
```

Si vemos el contenido de xp

```
print(xp)
```

```
##   age age.2 age.3  age.4
## 1  29   841 24389 707281
```

Esto sugiere otra manera de hacer la predicción:

```
xp1 = data.frame(age = age, age.2 = age^2, age.3 = age^3, age.4 = age^4)
predict(m1, newdata = xp1, interval = "confidence")
```

```
##        fit    lwr      upr
## 1 99.03991 96.883 101.1968
```

- Definiendo un cambio de variables:

```
z1 = d$age
z2 = d$age^2
z3 = d$age^3
z4 = d$age^4
m2 = lm(wage ~ z1 + z2 + z3 + z4, data = d)
summary(m2)
```

```
##
## Call:
## lm(formula = wage ~ z1 + z2 + z3 + z4, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -93.565 -20.689  -2.015  17.584 116.228
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.620e+02  4.563e+01  -3.550 0.000391 ***
## z1           1.948e+01  4.477e+00   4.350 1.41e-05 ***
## z2          -5.150e-01  1.569e-01  -3.283 0.001039 **
## z3           6.113e-03  2.334e-03   2.619 0.008869 **
## z4          -2.800e-05  1.250e-05  -2.240 0.025186 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.16 on 2916 degrees of freedom
## Multiple R-squared:  0.1094, Adjusted R-squared:  0.1082
## F-statistic: 89.55 on 4 and 2916 DF,  p-value: < 2.2e-16
```

Para predecir:

```
xp2 = data.frame(z1 = age, z2 = age^2, z3 = age^3, z4 = age^4)
predict(m1, newdata = xp2, interval = "confidence")
```

```
##        fit    lwr      upr
## 1 99.03991 96.883 101.1968
```

- Con la función *poly()*:

```
m3 = lm(wage ~ poly(age, degree = 4, raw = T), data = d)
summary(m3)
```

```
##
## Call:
## lm(formula = wage ~ poly(age, degree = 4, raw = T), data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -93.565 -20.689  -2.015  17.584 116.228
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -1.620e+02  4.563e+01  -3.550 0.000391 ***
## poly(age, degree = 4, raw = T)1  1.948e+01  4.477e+00   4.350 1.41e-05 ***
## poly(age, degree = 4, raw = T)2 -5.150e-01  1.569e-01  -3.283 0.001039 **
## poly(age, degree = 4, raw = T)3  6.113e-03  2.334e-03   2.619 0.008869 **
```

```
## poly(age, degree = 4, raw = T)4 -2.800e-05  1.250e-05  -2.240 0.025186 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.16 on 2916 degrees of freedom
## Multiple R-squared:  0.1094, Adjusted R-squared:  0.1082
## F-statistic: 89.55 on 4 and 2916 DF,  p-value: < 2.2e-16
```

La opción raw = T es necesaria, porque de lo contrario utiliza polinomios ortogonales. Para predecir:

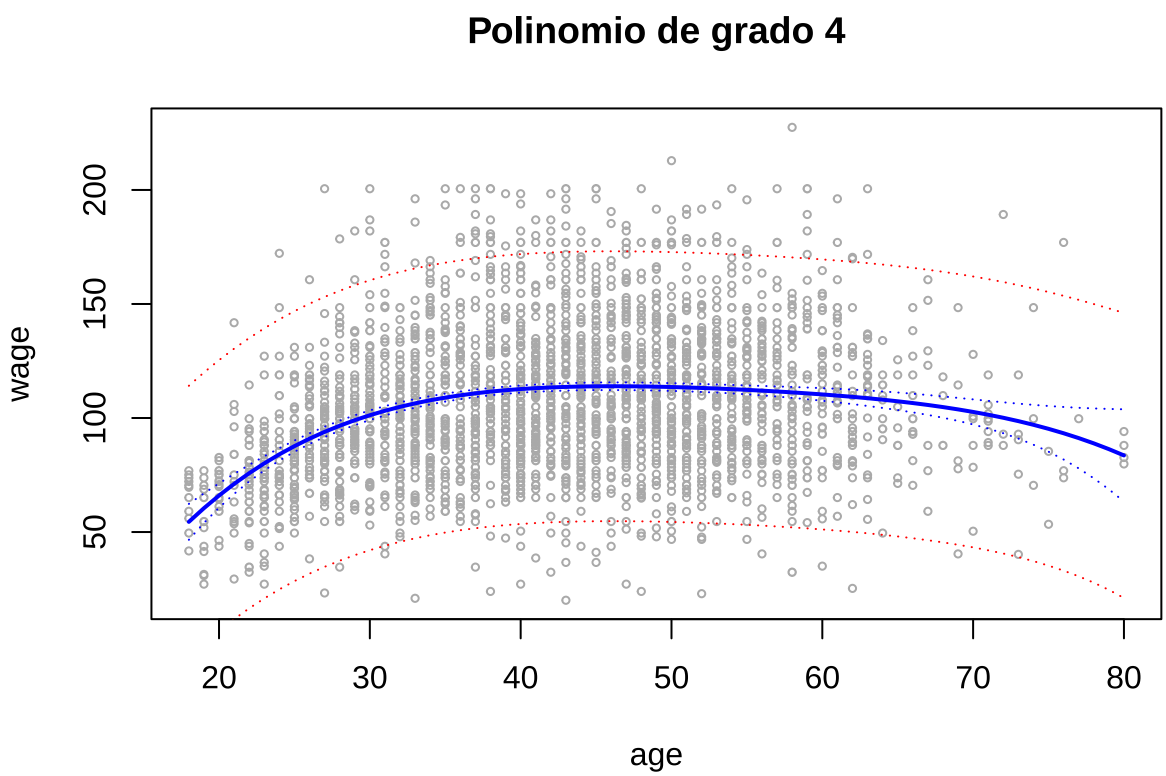
```
xp3 = data.frame(age = age)
predict(m1, newdata = xp3, interval = "confidence")
```

```
##        fit    lwr      upr
## 1 99.03991 96.883 101.1968
```

Es decir, la función *poly()* internamente crea las variables necesarias a partir de *age*. Vamos a dibujar la curva y los intervalos de confianza y predicción:

```
age_grid = seq(from = min(d$age), to = max(d$age), by = 1)
yp = predict(m1, newdata = data.frame(age = age_grid), se = TRUE)
yp = predict(m1, newdata = data.frame(age = age_grid), interval = "confidence", level = 0.95)
yp1 = predict(m1, newdata = data.frame(age = age_grid), interval = "prediction", level = 0.95)
```

```
plot(wage ~ age, data = d, xlim = range(age), cex = 0.5, col = "darkgrey")
title("Polinomio de grado 4")
lines(age_grid, yp[,1], lwd = 2, col = "blue")
# intervalos de confianza para el nivel medio
lines(age_grid, yp[,2], col = "blue", lty = 3)
lines(age_grid, yp[,3], col = "blue", lty = 3)
# intervalos de prediccion
lines(age_grid, yp1[,2], col = "red", lty = 3)
lines(age_grid, yp1[,3], col = "red", lty = 3)
```

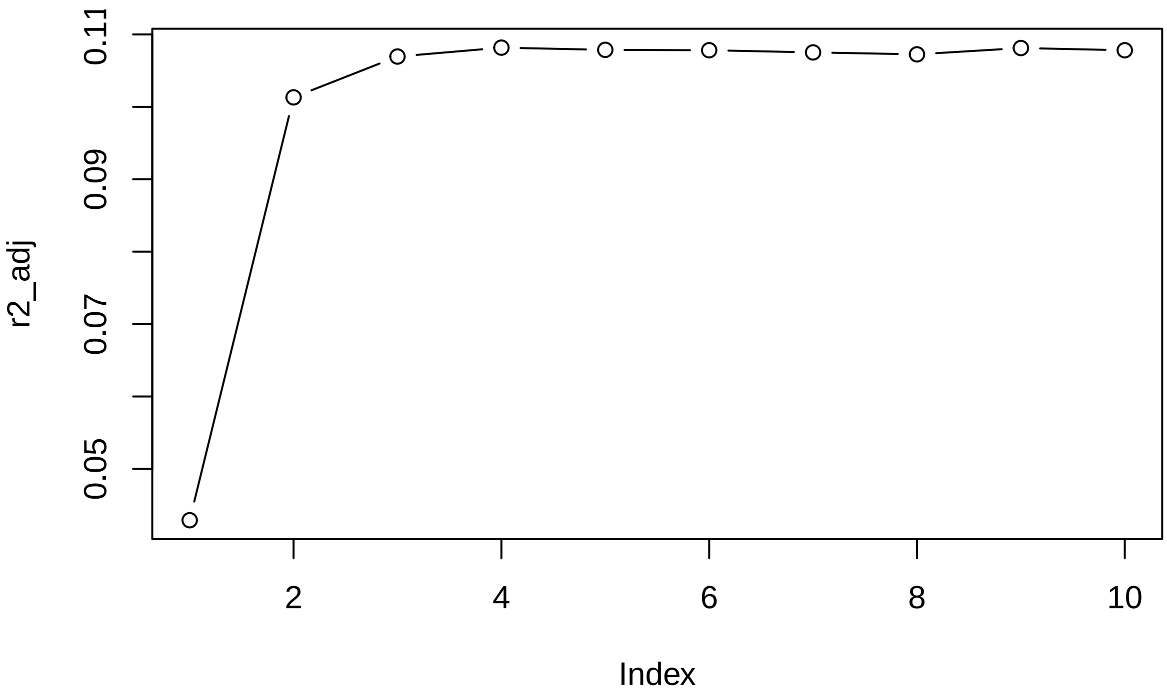


2.3 Selección del grado máximo del polinomio

Vamos a ir aumentando el grado del polinomio:

```
# numero maximo de escalones
max_grad = 10

r2_adj = rep(0, max_grad)
for (i in 1:max_grad){
  mi = lm(wage ~ poly(age, degree = i, raw = T), data = d)
  mi_summary = summary(mi)
  r2_adj[i] = mi_summary$adj.r.squared
}
plot(r2_adj, type = "b")
```



Como vemos, no aumentamos el R2 para ordenes mayores de 4. Podemos afinar más utilizando el contraste de la F:

$$F_0 = \frac{(SSR(m) - SSR(k))/(k-m)}{SSR(k)/(n-k-1)} \sim F_{k-m,n-k-1}$$

Vamos a comparar los modelos de grado 3, 4 y 5:

```
mk3 = lm(wage ~ poly(age, degree = 3, raw = T), data = d)
mk4 = lm(wage ~ poly(age, degree = 4, raw = T), data = d)
mk5 = lm(wage ~ poly(age, degree = 5, raw = T), data = d)
```

```
anova(mk3,mk4)
```

```
## Analysis of Variance Table
##
## Model 1: wage ~ poly(age, degree = 3, raw = T)
## Model 2: wage ~ poly(age, degree = 4, raw = T)
##   Res.Df     RSS Df Sum of Sq      F  Pr(>F)
## 1   2917 2657641
## 2   2916 2653077  1    4563.9 5.0162 0.02519 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mk4,mk5)
```

```
## Analysis of Variance Table
##
## Model 1: wage ~ poly(age, degree = 4, raw = T)
## Model 2: wage ~ poly(age, degree = 5, raw = T)
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1   2916 2653077
## 2   2915 2653073  1    3.8638 0.0042 0.9481
```

```r
anova(mk3,mk5)
```

```
## Analysis of Variance Table
##
## Model 1: wage ~ poly(age, degree = 3, raw = T)
## Model 2: wage ~ poly(age, degree = 5, raw = T)
##   Res.Df     RSS Df Sum of Sq      F  Pr(>F)
## 1   2917 2657641
## 2   2915 2653073  2    4567.8 2.5094 0.08149 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Como vemos, orden 3 y orden 5 son equivalentes, luego nos quedamos con el de orden 3 porque siempre preferimos modelos sencillos a modelos complejos.

3 Polinomios ortogonales

3.1 Definición del modelo

Uno de los principales problemas que tiene utilizar el modelo anterior es que para polinomios de grado elevado, la matriz $X^T X$ es casi singular, y podemos tener problemas en la estimación del modelo. Por ejemplo:

```r
mk6 = lm(wage ~ poly(age, degree = 6, raw = T), data = d)
summary(mk6)
```

```
##
## Call:
## lm(formula = wage ~ poly(age, degree = 6, raw = T), data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -93.347 -20.526  -1.956  17.549 115.680
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      1.219e+02  3.275e+02   0.372    0.710
## poly(age, degree = 6, raw = T)1 -2.428e+01  4.944e+01  -0.491    0.623
## poly(age, degree = 6, raw = T)2  2.168e+00  2.986e+00   0.726    0.468
## poly(age, degree = 6, raw = T)3 -7.784e-02  9.255e-02  -0.841    0.400
## poly(age, degree = 6, raw = T)4  1.391e-03  1.556e-03   0.894    0.372
## poly(age, degree = 6, raw = T)5 -1.231e-05  1.349e-05  -0.913    0.362
## poly(age, degree = 6, raw = T)6  4.299e-08  4.723e-08   0.910    0.363
##
## Residual standard error: 30.17 on 2914 degrees of freedom
## Multiple R-squared:  0.1097, Adjusted R-squared:  0.1078
## F-statistic: 59.81 on 6 and 2914 DF,  p-value: < 2.2e-16
```

Como vemos, en este modelo salen todos los parámetros no significativos, incluso β_1 y β_2.

Una opción es utilizar el modelo:

$$y_i = \beta_0 + \beta_1 P_1(x_i) + \beta_2 P_2(x_i) + \cdots + \beta_k P_k(x_i) + u_i$$

donde $P_k(x_i)$ es el polinomio de orden k que verifica:

$$\sum_{i=1}^{n} P_r(x_i)P_s(x_i) \neq 0, \text{ cuando } r = s;$$

$$\sum_{i=1}^{n} P_r(x_i)P_s(x_i) = 0, \text{ cuando } r \neq s;$$

es decir, son polinomios ortogonales. El modelo sigue siendo $y = X\beta + u$, con

$$X = \begin{bmatrix} 1 & P_1(x_1) & \cdots & P_k(x_1) \\ 1 & P_1(x_2) & \cdots & P_k(x_2) \\ \vdots & \vdots & & \vdots \\ 1 & P_1(x_n) & \cdots & P_k(x_n) \end{bmatrix}$$

Por tanto:

$$X^T X = \begin{bmatrix} n & 0 & \cdots & 0 \\ 0 & \sum_{i=1}^n P_1^2(x_i) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \sum_{i=1}^n P_k^2(x_i) \end{bmatrix}, \quad X^T y = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n P_1(x_i) y_i \\ \vdots \\ \sum_{i=1}^n P_k(x_i) y_i \end{bmatrix}$$

Esta matriz es invertible (al ser diagonal) y:

$$\hat{\beta}_j = \frac{\sum_{i=1}^n P_j(x_i) y_i}{\sum_{i=1}^n P_j^2(x_i)}$$

Una consecuencia importante es que como $Var[\hat{\beta}] = \sigma^2 (X^T X)^{-1}$, se tiene que:

$$Var[\hat{\beta}_j] = \frac{\sigma^2}{\sum_{i=1}^n P_j^2(x_i)}$$

3.2 Propiedades

- Los parámetros del modelo ortogonal no coinciden con los del modelo polinómico no ortogonal:

```r
mk4a = lm(wage ~ poly(age, degree = 4), data = d)
summary(mk4a)
```

```
##
## Call:
## lm(formula = wage ~ poly(age, degree = 4), data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -93.565 -20.689  -2.015  17.584 116.228
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             107.2187     0.5581  192.11  < 2e-16 ***
## poly(age, degree = 4)1  358.9196    30.1635   11.90  < 2e-16 ***
## poly(age, degree = 4)2 -418.0999    30.1635  -13.86  < 2e-16 ***
## poly(age, degree = 4)3  133.0075    30.1635    4.41 1.07e-05 ***
```

```
## poly(age, degree = 4)4  -67.5570    30.1635   -2.24   0.0252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.16 on 2916 degrees of freedom
## Multiple R-squared:  0.1094, Adjusted R-squared:  0.1082
## F-statistic: 89.55 on 4 and 2916 DF,  p-value: < 2.2e-16
```

```r
summary(mk4)
```

```
##
## Call:
## lm(formula = wage ~ poly(age, degree = 4, raw = T), data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -93.565 -20.689  -2.015  17.584 116.228
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -1.620e+02  4.563e+01  -3.550 0.000391 ***
## poly(age, degree = 4, raw = T)1  1.948e+01  4.477e+00   4.350 1.41e-05 ***
## poly(age, degree = 4, raw = T)2 -5.150e-01  1.569e-01  -3.283 0.001039 **
## poly(age, degree = 4, raw = T)3  6.113e-03  2.334e-03   2.619 0.008869 **
## poly(age, degree = 4, raw = T)4 -2.800e-05  1.250e-05  -2.240 0.025186 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.16 on 2916 degrees of freedom
## Multiple R-squared:  0.1094, Adjusted R-squared:  0.1082
## F-statistic: 89.55 on 4 and 2916 DF,  p-value: < 2.2e-16
```

- pero coincide la varianza residual, el R2,...
- En el fondo, se ha hecho un cambio de base, pero el modelo final es el mismo. El cambio de base se puede expresar como:

$$y = X\hat{\beta} + e = XCC^{-1}\hat{\beta} + e = X^*\hat{\beta}^* + e$$

donde C es la matriz de cambio de base:

$$X^* = XC, \ \hat{\beta}^* = C^{-1}\hat{\beta}.$$

Como vemos, cambian los estimadores pero no los residuos.

- La predicción tiene que ser la misma:
  - Con polinomios "nomales": $\hat{y}_p = x_p \beta$
  - Con polinomios ortogonales: $\hat{y}_p^* = x_p^* \beta^*$

Teniendo en cuenta la matriz de cambio de base:

$$\hat{y}_p^* = x_p^* \beta^* = (x_p^* C) \cdot (C^{-1}\beta^*) = x_p \beta = \hat{y}_p$$

```r
predict(mk4, newdata = data.frame(age = 22), interval = "confidence")
```

```
##        fit      lwr      upr
## 1 75.79416 72.15342 79.43489
```

```
predict(mk4a, newdata = data.frame(age = 22), interval = "confidence")
```

```
##        fit      lwr      upr
## 1 75.79416 72.15342 79.43489
```

- Los regresores se van añadiendo sin modificar las estimaciones obtenidas para los parámetros ya obtenidos:

```
summary(lm(wage ~ poly(age, degree = 3), data = d))
```

```
##
## Call:
## lm(formula = wage ~ poly(age, degree = 3), data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -94.439 -20.772  -2.009  17.727 117.373
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             107.2187     0.5585 191.980  < 2e-16 ***
## poly(age, degree = 3)1  358.9196    30.1842  11.891  < 2e-16 ***
## poly(age, degree = 3)2 -418.0999    30.1842 -13.852  < 2e-16 ***
## poly(age, degree = 3)3  133.0075    30.1842   4.407 1.09e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.18 on 2917 degrees of freedom
## Multiple R-squared:  0.1079, Adjusted R-squared:  0.1069
## F-statistic: 117.6 on 3 and 2917 DF,  p-value: < 2.2e-16
```

```
summary(lm(wage ~ poly(age, degree = 4), data = d))
```

```
##
## Call:
## lm(formula = wage ~ poly(age, degree = 4), data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -93.565 -20.689  -2.015  17.584 116.228
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             107.2187     0.5581  192.11  < 2e-16 ***
## poly(age, degree = 4)1  358.9196    30.1635   11.90  < 2e-16 ***
## poly(age, degree = 4)2 -418.0999    30.1635  -13.86  < 2e-16 ***
## poly(age, degree = 4)3  133.0075    30.1635    4.41 1.07e-05 ***
## poly(age, degree = 4)4  -67.5570    30.1635   -2.24   0.0252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.16 on 2916 degrees of freedom
## Multiple R-squared:  0.1094, Adjusted R-squared:  0.1082
## F-statistic: 89.55 on 4 and 2916 DF,  p-value: < 2.2e-16
```