Day 11 - Lab 7.1

Kate Ardinger

1/21/2015

```
library(ISLR)
attach(Wage)
```

Fitting the model - with linear combinations of each element poly(age, 4)

```
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
```

Fitting again - raw=T represents exactly age, age^2, age^3, etc.

```
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
```

Another way to fit the model - I() is a “wrapper” function

```
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
```

A more compact way to do the same model - cbind() is also a wrapper function

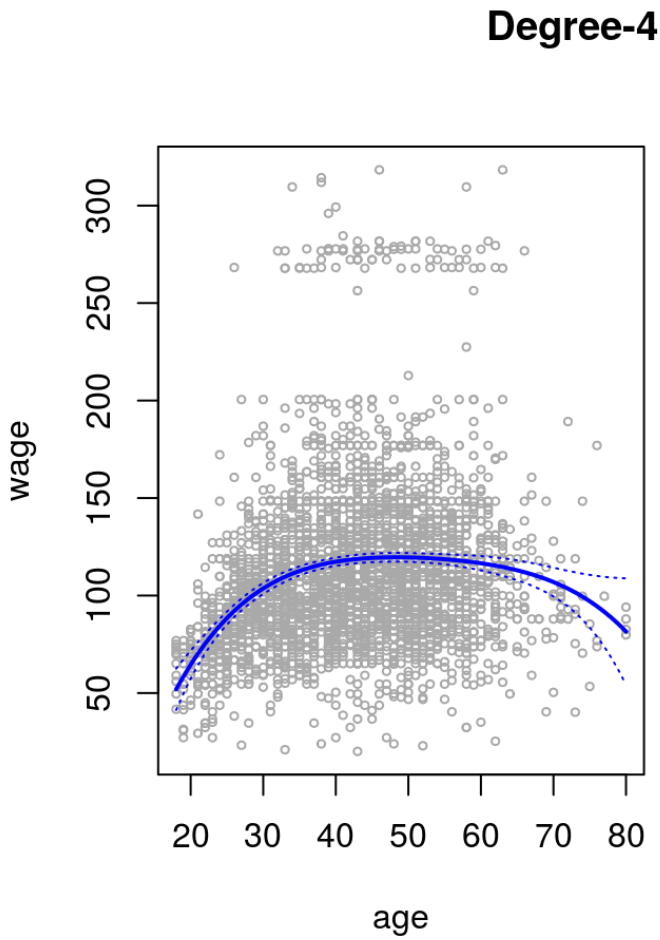
```
fit2b <- lm(wage~cbind(age, age^2, age^3, age^4), data=Wage)
```

Now we will generate values of age for which we will make predictions; also, specify that we want the standard errors

```
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)
```

Now plotting the data

```
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)
```



the fitted values are the same for both ways of using poly()

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

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

can also decide the degree of polynomial to use by using hypothesis tests

```
fit1 <- lm(wage~age, data=Wage)
fit2 <- lm(wage~poly(age, 2), data=Wage)
fit3 <- lm(wage~poly(age, 3), data=Wage)
fit4 <- lm(wage~poly(age, 4), data=Wage)
fit5 <- lm(wage~poly(age, 5), data=Wage)
anova(fit1, fit2, fit3, fit4, fit5)
```

```
## 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
```

Null hypothesis i: that model i is sufficient to explain the data against the alternative hypothesis that a more complex model (i+1) is required.

We can do the same thing using the poly()

```
coef(summary(fit5))
```

```
##                 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
```

The square of the t-statistics are the same as the F-statistics in the anova() function

Can also use anova() to compare these models

```
fit1 <- lm(wage~education+age, data=Wage)
fit2 <- lm(wage~education+poly(age,2), data=Wage)
fit3 <- lm(wage~education+poly(age,3), data=Wage)
anova(fit1,fit2,fit3)
```

```
## 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
```

Notice that we could also choose the correct polynomial degree using cross-validation

now we will predict if an individual earns more than $250,000 a year

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

we have to use a transformation to come up with the C.I. for logistic regressions

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

can compute probabilities using the predict() function

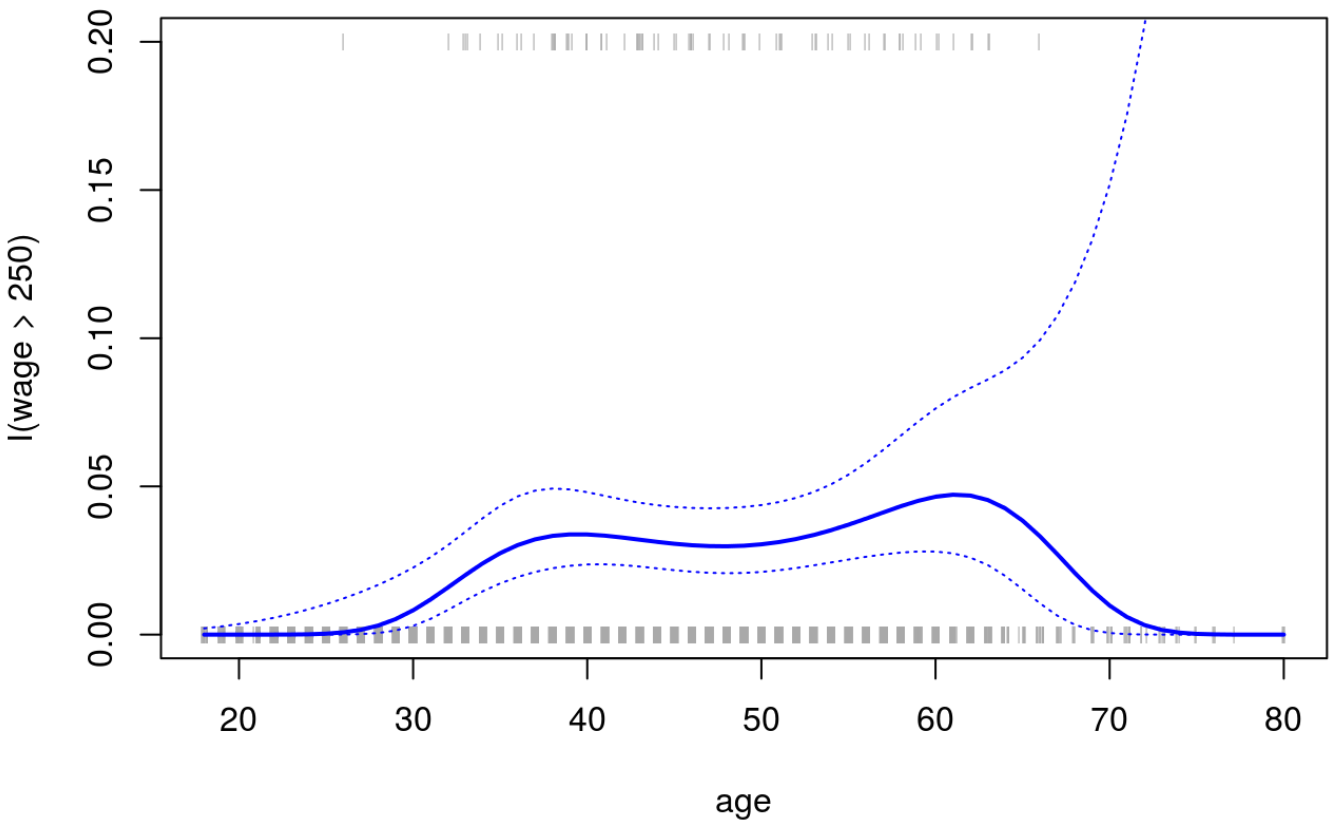
```
preds <- predict(fit, newdata=list(age=age.grid), type="response")
head(preds)
```

```
##            1            2            3            4            5
## 9.826427e-09 7.577844e-08 4.746699e-07 2.449201e-06 1.055188e-05
##            6
## 3.845805e-05
```

However, we cannot compute confidence intervals for these probabilities because they would be negative.

A Rug Plot

```
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)
```



Fitting a step function

```
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
```