Non linear modeling

In this notebook we'll use the Wage data from the ISLR library to explore the realm of non linear models.

```
library (ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.6.3
```

```
attach (Wage)
```

```
head(Wage)
```

```
##        year age            maritl     race       education             region
## 231655 2006  18 1. Never Married 1. White    1. < HS Grad 2. Middle Atlantic
## 86582  2004  24 1. Never Married 1. White 4. College Grad 2. Middle Atlantic
## 161300 2003  45       2. Married 1. White 3. Some College 2. Middle Atlantic
## 155159 2003  43       2. Married 3. Asian 4. College Grad 2. Middle Atlantic
## 11443  2005  50      4. Divorced 1. White      2. HS Grad 2. Middle Atlantic
## 376662 2008  54       2. Married 1. White 4. College Grad 2. Middle Atlantic
##              jobclass         health health_ins  logwage      wage
## 231655  1. Industrial      1. <=Good      2. No 4.318063  75.04315
## 86582  2. Information 2. >=Very Good      2. No 4.255273  70.47602
## 161300  1. Industrial      1. <=Good     1. Yes 4.875061 130.98218
## 155159 2. Information 2. >=Very Good     1. Yes 5.041393 154.68529
## 11443  2. Information      1. <=Good     1. Yes 4.318063  75.04315
## 376662 2. Information 2. >=Very Good     1. Yes 4.845098 127.11574
```

Polynomial models

Is the wage a 4 order polynomial of the age of the person, considering Gaussian noise in it?

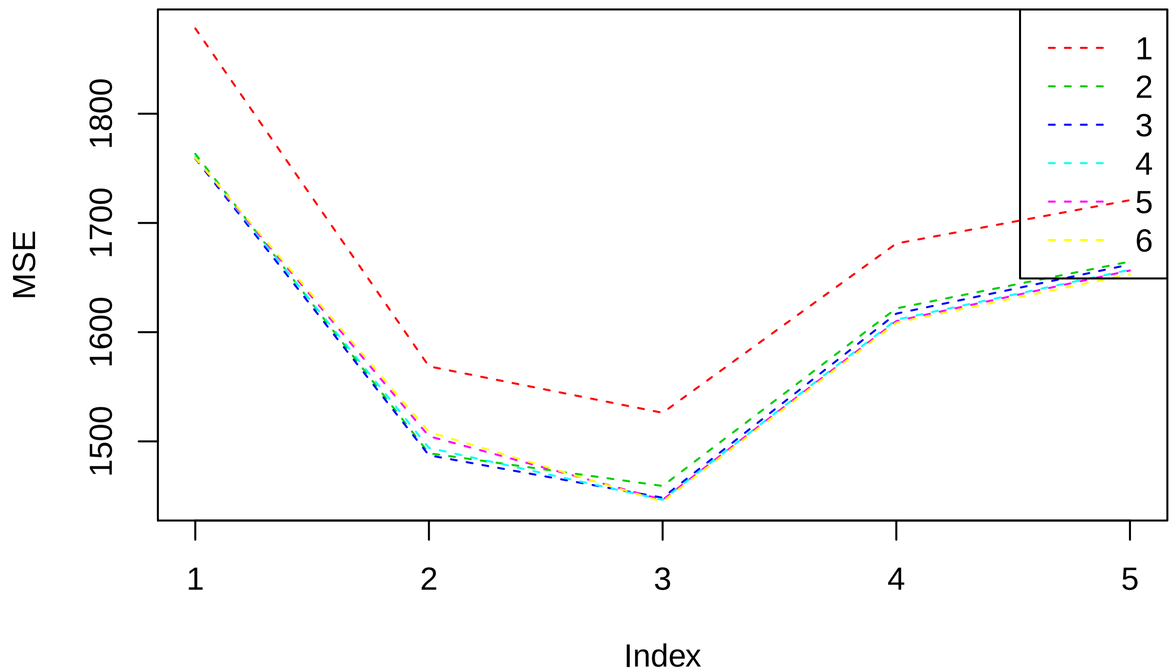
```
fit=lm(wage~poly(age ,4) ,data=Wage)
summary(fit)
```

```
##
## Call:
## lm(formula = wage ~ poly(age, 4), data = Wage)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -98.707 -24.626  -4.993  15.217 203.693
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    111.7036     0.7287 153.283  < 2e-16 ***
## poly(age, 4)1  447.0679    39.9148  11.201  < 2e-16 ***
## poly(age, 4)2 -478.3158    39.9148 -11.983  < 2e-16 ***
## poly(age, 4)3  125.5217    39.9148   3.145  0.00168 **
## poly(age, 4)4  -77.9112    39.9148  -1.952  0.05104 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 39.91 on 2995 degrees of freedom
## Multiple R-squared:  0.08626,    Adjusted R-squared:  0.08504
## F-statistic: 70.69 on 4 and 2995 DF,  p-value: < 2.2e-16
```

The answer to the above question seems to be partially positive, because the four order polynomial doesn't seem statistically significant to predict the response. Let's use cross-validation to evaluate the different models.

```r
ncv <- 5
n <- dim(Wage)[1]
#shuffling
indices <- sample(1:n, size = n, replace=F)
#splitting
folds <- cut(indices, breaks = ncv, labels = F)
#poly order
od <- c(1,2,3,4,5,6)
res <- matrix(nrow=ncv, ncol=6)
for(order in od){
  for(i in 1:ncv){
    test <- indices[folds==i]
    fit<-lm(wage~poly(age ,order) ,data=Wage, subset=-test)
    preds<-predict(fit, newdata=Wage[test,])
    error<-sum((preds-Wage$wage[test])**2)/length(test)
    res[i,order]<-error
  }
}
plot(res[,1], type="l", lty="dashed", col=2, ylim =c(min(res),max(res)), ylab="MSE")
lines(res[,2], lty="dashed", col=3)
lines(res[,3], lty="dashed", col=4)
lines(res[,4], lty="dashed", col=5)
lines(res[,5], lty="dashed", col=6)
lines(res[,6], lty="dashed", col=7)
legend("topright",legend=c("1","2","3","4","5","6"), col=c(2,3,4,5,6,7), lty="dashed")
```




```
colMeans(res)
```

```
## [1] 1675.014 1599.598 1594.726 1593.914 1595.433 1594.930
```

```
which.min(colMeans(res))
```

```
## [1] 4
```

The fourth order degree seems to be the best one according to our cross validation! Let's see what the glm automatic cross validation would say.

```
library(boot)
res.glm <- numeric(6)
for(order in od){
  fit.glm <- glm(wage~poly(age ,order) ,data=Wage)
  res.glm[order] <- cv.glm(Wage, fit.glm, K=ncv)$delta[2]
}
res.glm
```

```
## [1] 1676.723 1598.682 1594.481 1594.187 1596.937 1595.256
```

```
which.min(res.glm)
```

```
## [1] 4
```

The glm cross-validation and our cross validation seem to agree. What about the ANOVA test?

```
fit1 <- lm(wage~poly(age ,1) ,data=Wage)
fit2 <- lm(wage~poly(age ,2) ,data=Wage)
fit3 <- lm(wage~poly(age ,3) ,data=Wage)
```

```r
fit4 <- lm(wage~poly(age ,4) ,data=Wage)
fit5 <- lm(wage~poly(age ,5) ,data=Wage)
fit6 <- lm(wage~poly(age ,6) ,data=Wage)
anova(fit1,fit2,fit3,fit4,fit5,fit6)
```

```
## Analysis of Variance Table
##
## Model 1: wage ~ poly(age, 1)
## Model 2: wage ~ poly(age, 2)
## Model 3: wage ~ poly(age, 3)
## Model 4: wage ~ poly(age, 4)
## Model 5: wage ~ poly(age, 5)
## Model 6: wage ~ poly(age, 6)
##   Res.Df     RSS Df Sum of Sq        F    Pr(>F)
## 1   2998 5022216
## 2   2997 4793430  1    228786 143.6636 < 2.2e-16 ***
## 3   2996 4777674  1     15756   9.8936  0.001675 **
## 4   2995 4771604  1      6070   3.8117  0.050989 .
## 5   2994 4770322  1      1283   0.8054  0.369565
## 6   2993 4766389  1      3932   2.4692  0.116201
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

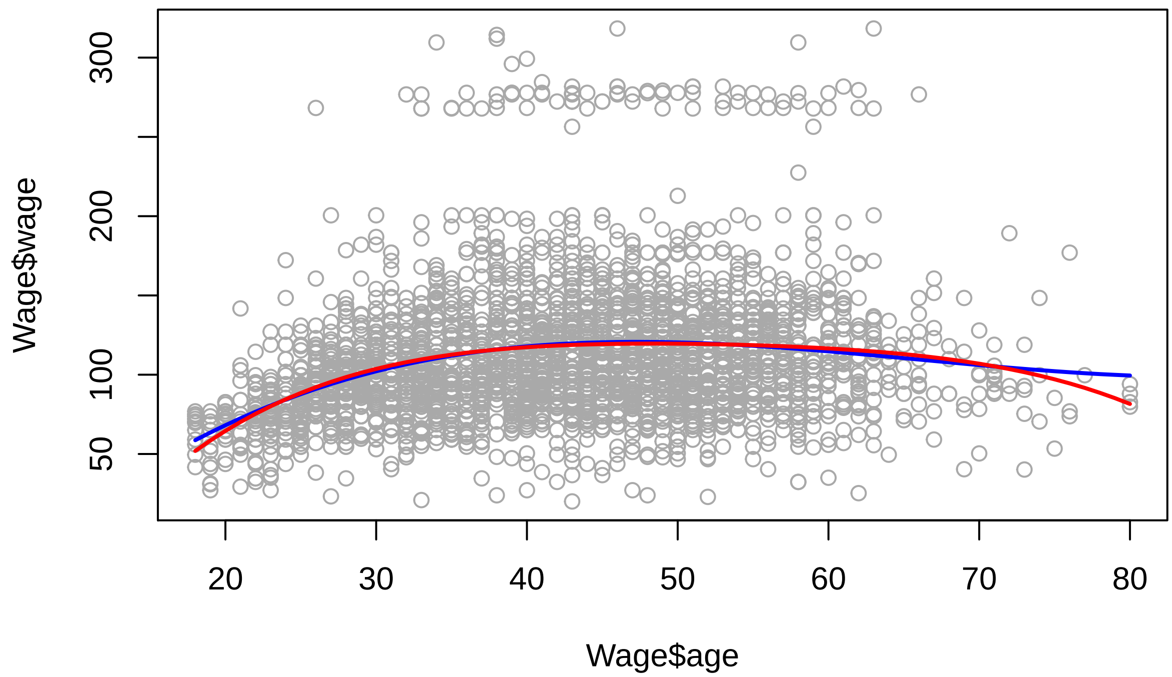
Note that the p-values obtained with the ANOVA are the same we obtain from the T-test in the biggest model.

```r
summary(fit6)
```

```
##
## Call:
## lm(formula = wage ~ poly(age, 6), data = Wage)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -98.521 -24.536  -4.848  15.471 202.108
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    111.7036     0.7286 153.316  < 2e-16 ***
## poly(age, 6)1  447.0679    39.9063  11.203  < 2e-16 ***
## poly(age, 6)2 -478.3158    39.9063 -11.986  < 2e-16 ***
## poly(age, 6)3  125.5217    39.9063   3.145  0.00167 **
## poly(age, 6)4  -77.9112    39.9063  -1.952  0.05099 .
## poly(age, 6)5  -35.8129    39.9063  -0.897  0.36956
## poly(age, 6)6   62.7077    39.9063   1.571  0.11620
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.91 on 2993 degrees of freedom
## Multiple R-squared:  0.08726,    Adjusted R-squared:  0.08543
## F-statistic: 47.69 on 6 and 2993 DF,  p-value: < 2.2e-16
```

This happens because the poly function automatically builds orthogonal coordinates, hence the p-value associated with one predictor cannot be influenced by the presence/absence of other predictors. The Anova hence, like the T-test, doesn't see the fourth order term as statistically significant. Let's have a look at the third and fourth order fits.

```
plot(Wage$age,Wage$wage, col="darkgray")
agelims <- range(Wage$age)
age.grid <- seq(from=agelims[1],to=agelims[2])
preds3 <- predict(fit3, newdata = data.frame(age=age.grid))
preds4 <- predict(fit4, newdata = data.frame(age=age.grid))
lines(age.grid, preds3, col="blue",lwd=2)
lines(age.grid, preds4, col="red",lwd=2)
```



Step functions