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?

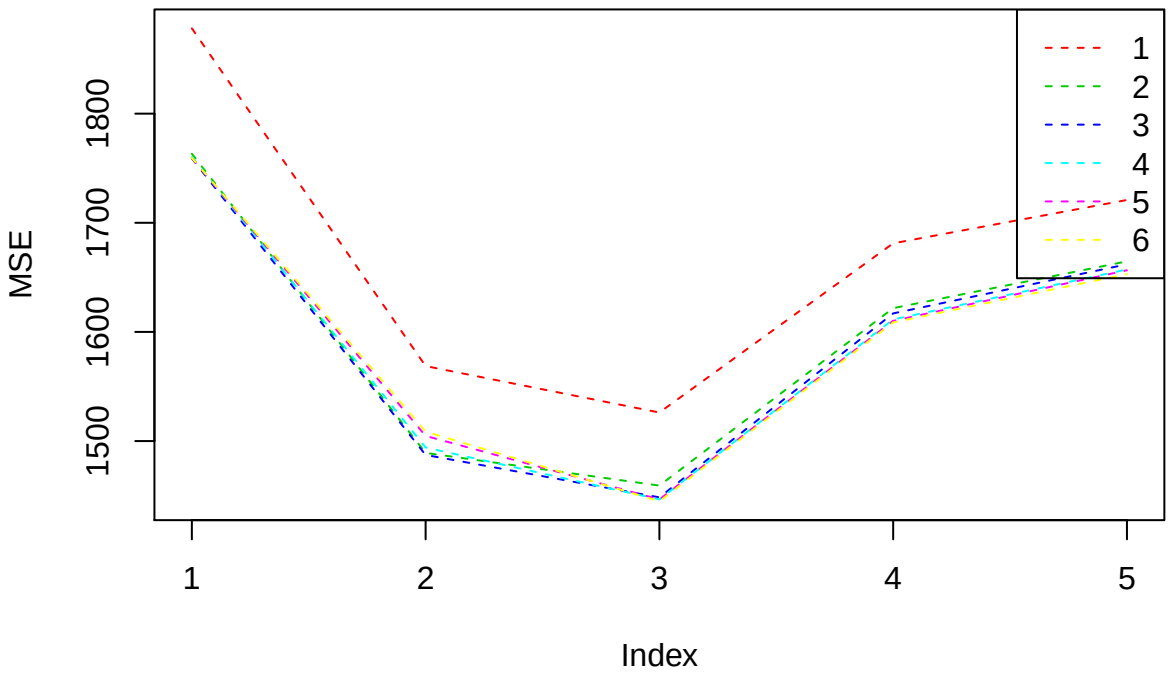
```
fit=lm(wage~poly(age ,4) ,data=Wage)
summary(fit)
```

```
##
## 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.

```
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] 1676.788 1600.578 1599.512 1594.569 1593.563 1592.002
```

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

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

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.

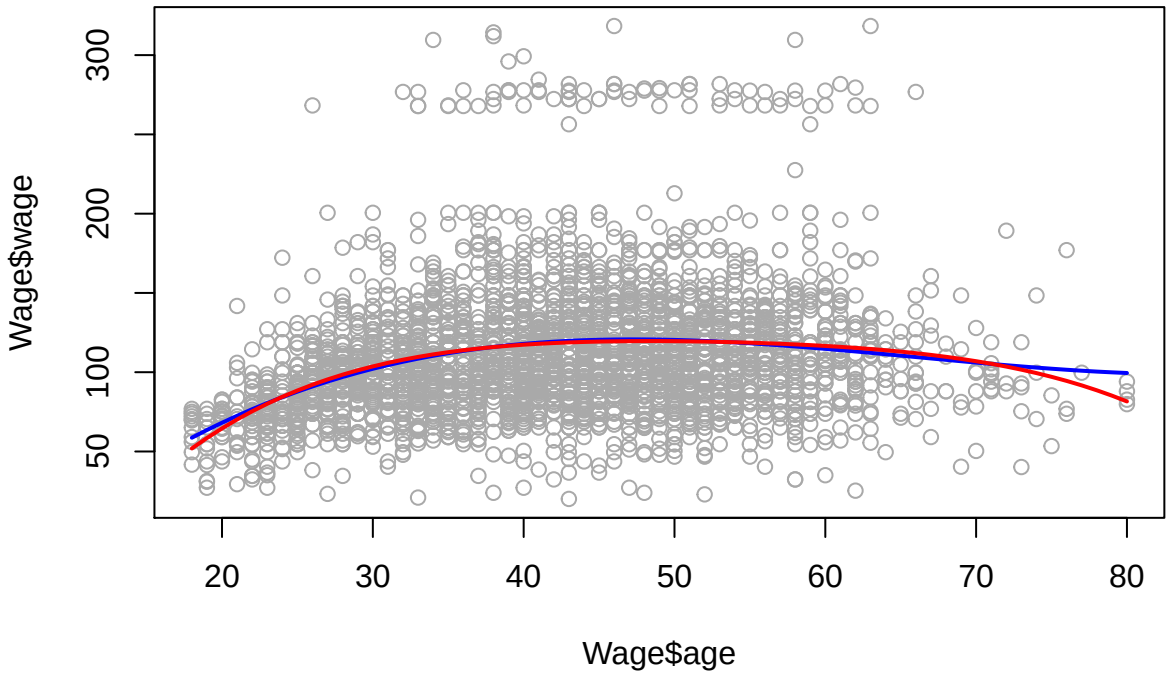
```r
summary(fit6)
```

```
##
## 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

We now want to fit a step function to predict wage using age, and perform cross-validation to choose the optimal number of cuts.

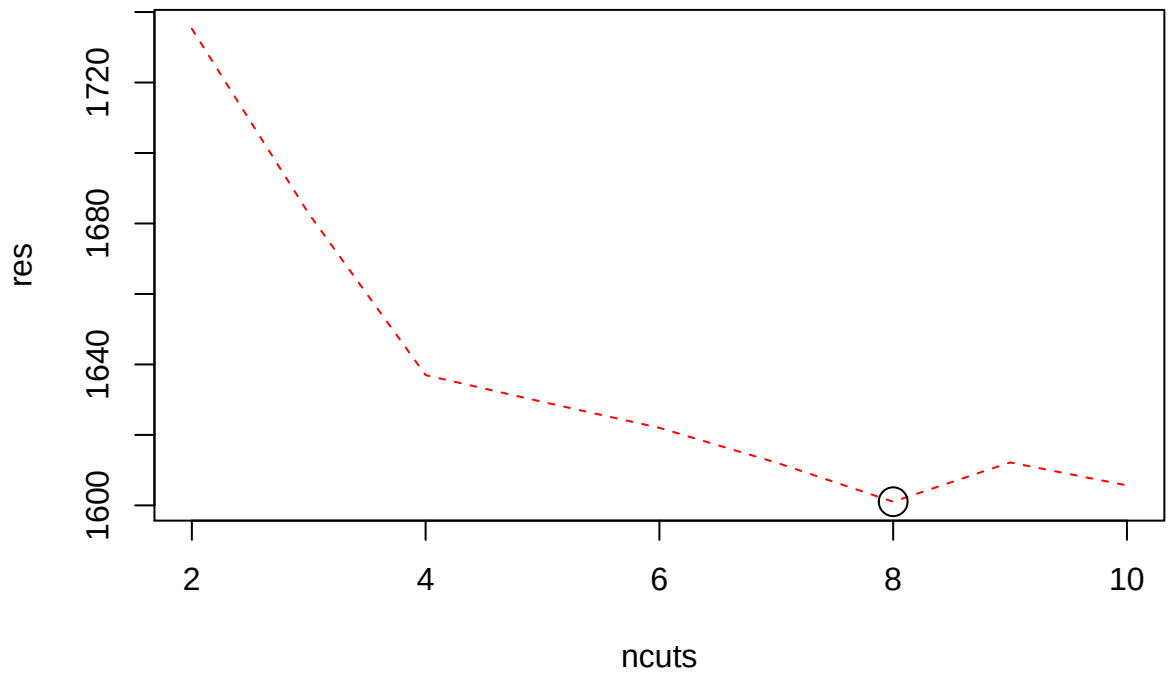
```
# The number of cuts we want to experiment with
ncuts <- c(2,3,4,5,6,7,8,9,10)
res <- numeric(length(ncuts))
for(j in 1:length(ncuts)){
  nc <- ncuts[j]
  # saving the new factor variable in the dataframe
  Wage$age.cut <- cut(age,nc)
  # fit step function to the train data
  fit.step <- glm(wage~age.cut, data=Wage)
  # evaluate the fit on the test data
  cv.res <- cv.glm(Wage, fit.step, K=ncv)$delta[2]
  res[j] <- cv.res
```

```
}

res
```

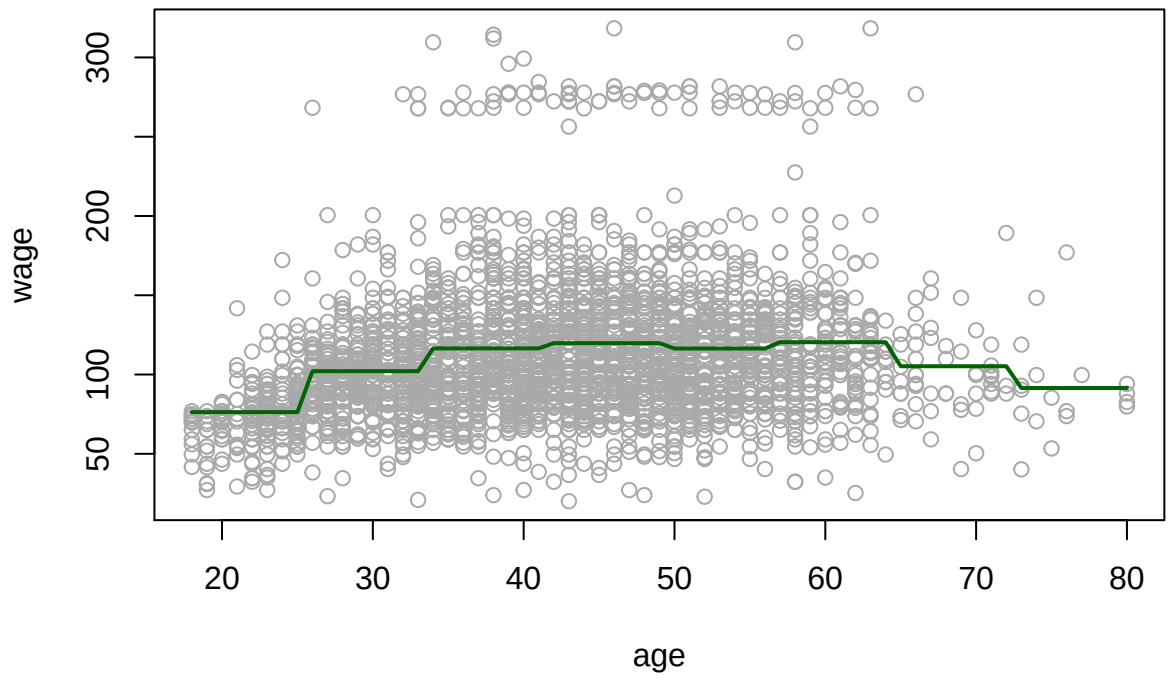
```
## [1] 1735.238 1682.796 1636.964 1629.375 1622.011 1612.144 1601.045 1612.188
## [9] 1605.684
```

```
plot(ncuts, res, type="l", lty="dashed", col="red")
points(ncuts[which.min((res))], min(res), cex=2)
```



Hence the minimum is obtained with 8 cuts. Let's have a look at the fitted function.

```
fit.8.steps <- glm(wage~cut(age,8), data=Wage)
preds.steps <- predict(fit.8.steps, newdata=data.frame(age=age.grid))
plot(age, wage, col="darkgray")
lines(age.grid, preds.steps, lwd=2, col="darkgreen")
```



Let's now explore other the non-linear relationships between wage and other variables.

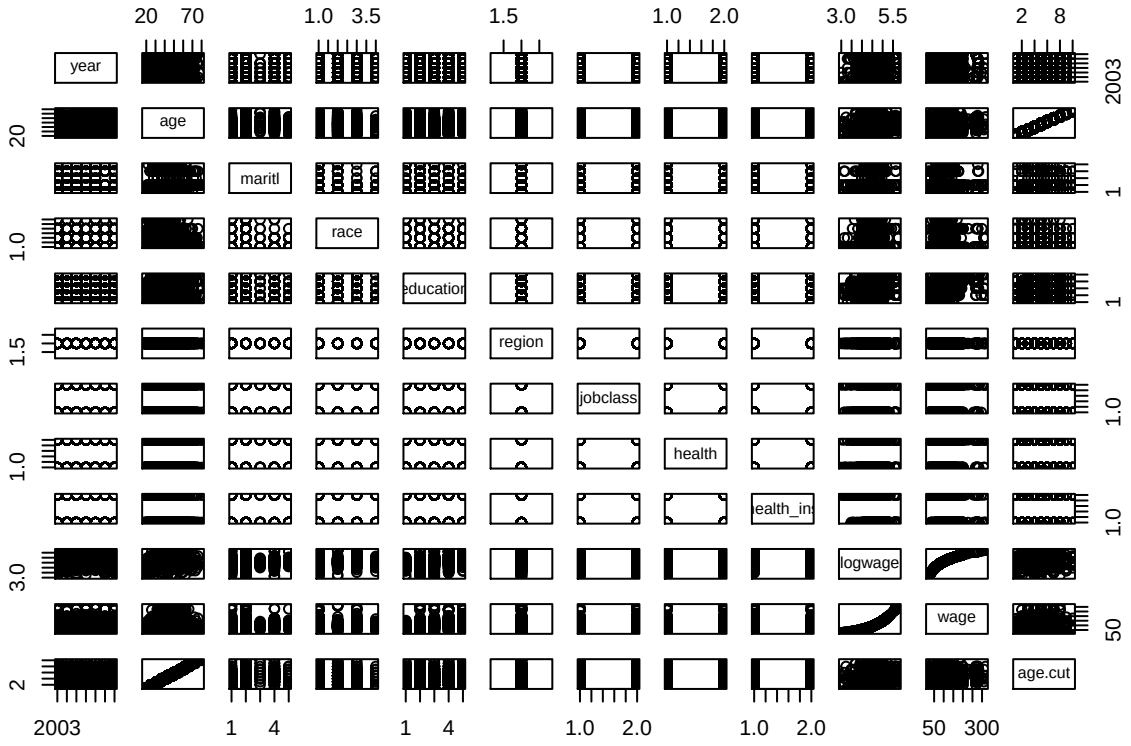
```
p <- dim(Wage)[2]
p
```

```
## [1] 12
```

```
head(Wage)
```

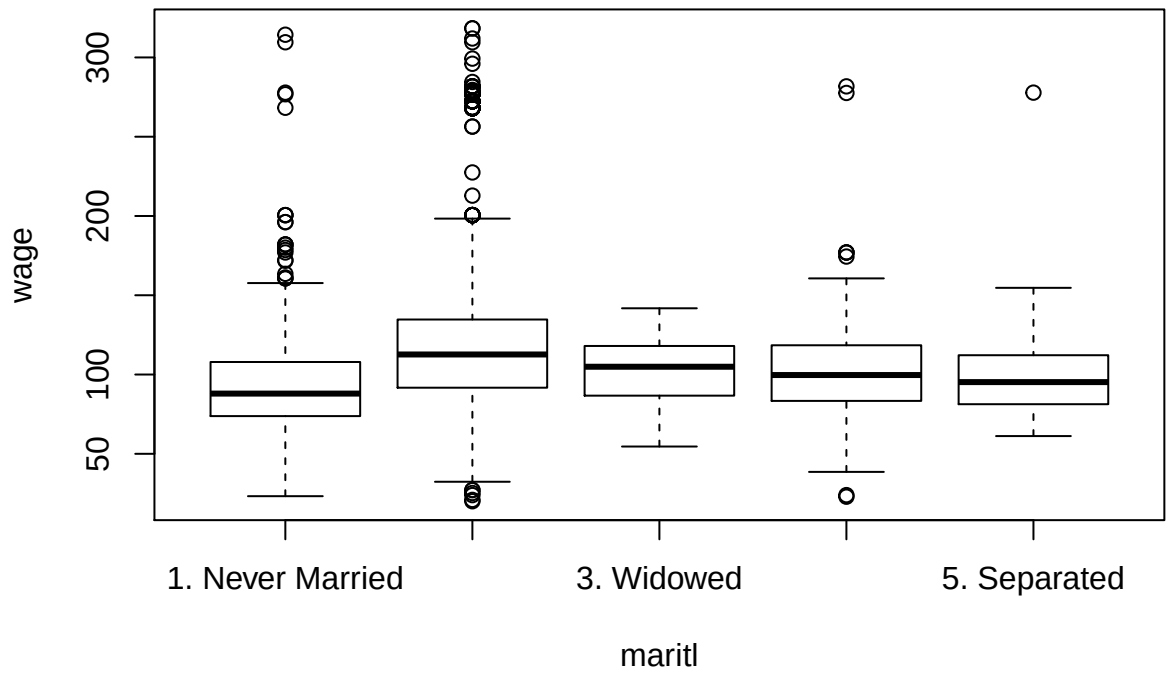
```
##        year age           maritl     race       education             region
## 231655 2006  18 1. Never Married 1. White    1. < HS Grad 2. Middle Atlantic
## 86582  2004  24 1. Never Married 1. White 4. College Grad 2. Middle Atlantic
## 161300 2003  45       2. Married 1. White 3. Some College 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     age.cut
## 231655  1. Industrial      1. <=Good      2. No 4.318063  75.04315 (17.9,24.2]
## 86582  2. Information 2. >=Very Good      2. No 4.255273  70.47602 (17.9,24.2]
## 161300  1. Industrial      1. <=Good     1. Yes 4.875061 130.98218   (42.8,49]
## 155159 2. Information 2. >=Very Good     1. Yes 5.041393 154.68529   (42.8,49]
## 11443  2. Information      1. <=Good     1. Yes 4.318063  75.04315   (49,55.2]
## 376662 2. Information 2. >=Very Good     1. Yes 4.845098 127.11574   (49,55.2]
```

```
# 11 predictors
pairs(Wage)
```



We'll now build a generalized additive model to predict the Wage based on the age, the marital status and the eductation. We'll first do some exploratory analysis to evaluate the relationships between wage and marital status and education.

```
plot(maritl,wage, xlab="maritl", ylab="wage")
```



Since the main difference seems to be between never married, married and the the other 3 categories together let's try models where we use this categorical variable with some of the levels.

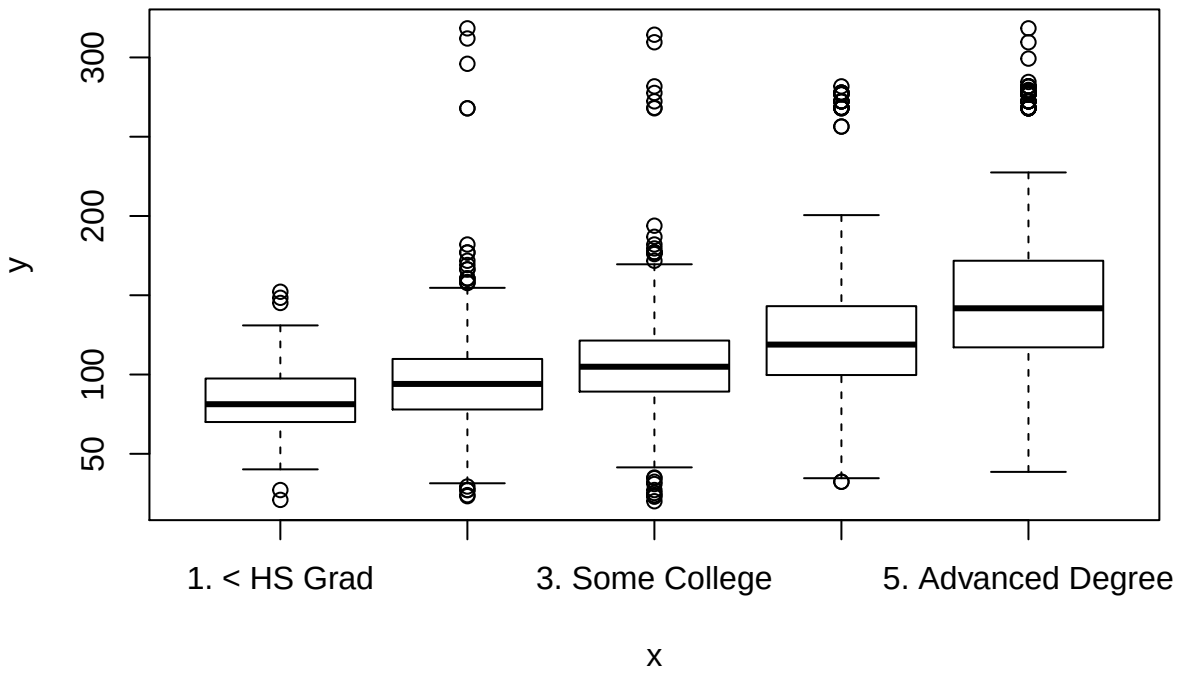
```
lvl1 <- maritl == levels(as.factor(maritl))[1]
lvl2 <- maritl == levels(as.factor(maritl))[2]
fit1 <- lm(wage~lvl2)
fit2 <- lm(wage~lvl2+lvl1)
fit3 <- lm(wage~maritl)
anova(fit1,fit2,fit3)
```

```
## Analysis of Variance Table
##
## Model 1: wage ~ lvl2
## Model 2: wage ~ lvl2 + lvl1
## Model 3: wage ~ maritl
##   Res.Df     RSS Df Sum of Sq       F    Pr(>F)
## 1   2998 4877943
## 2   2997 4859287  1   18655.6 11.4991 0.0007053 ***
## 3   2995 4858941  2     345.8  0.1066 0.8989166
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The anova test confirms our intuitions: there's no statistically significant difference between Widowed Divorced and Separated, however there is a difference between these three together and Married or Never Married.

Let's now look at education:

```
plot(education, wage)
```



If we do the same we've done before for marital status here to education we'll probably obtain opposite results: all the categories are statistically significantly different. But let's put it into numbers.

```
edu1 <- education == levels(as.factor(education))[1]
edu3 <- education == levels(as.factor(education))[3]
edu5 <- education == levels(as.factor(education))[5]
fit1 <- lm(wage~edu3)
fit2 <- lm(wage~edu3+edu1)
fit3 <- lm(wage~edu3+edu1+edu5)
fit4 <- lm(wage~education)
anova(fit1,fit2,fit3,fit4)
```

```
## Analysis of Variance Table
##
## Model 1: wage ~ edu3
## Model 2: wage ~ edu3 + edu1
## Model 3: wage ~ edu3 + edu1 + edu5
## Model 4: wage ~ education
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1   2998 5209152
## 2   2997 4960140  1    249012 186.65 < 2.2e-16 ***
## 3   2996 4325281  1    634860 475.86 < 2.2e-16 ***
## 4   2995 3995721  1    329559 247.02 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Our intution was right. Let's now turn to the variable age again, but this time let's use splines.

```
library(gam)
```

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

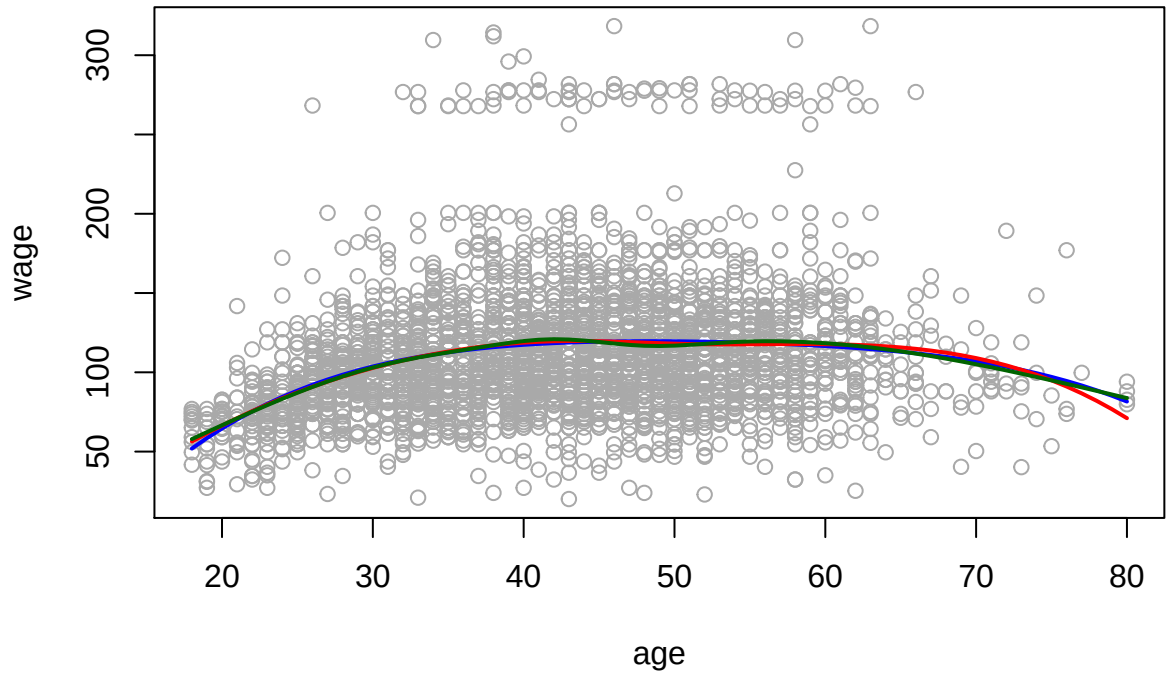
```
## Loading required package: splines
```

```
## Loading required package: foreach
```

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

```
## Loaded gam 1.16.1
```

```
poly.fit <- glm(wage~poly(age,4), data=Wage)
spline.fit <- glm(wage~bs(age, df = 6))
nat.spline.fit <- glm(wage~ns(age, df = 6))
plot(age,wage, col="darkgray")
preds.poly <- predict(poly.fit, newdata=data.frame(age=age.grid))
preds.spline <- predict(spline.fit, newdata=data.frame(age=age.grid))
preds.nat.spline <- predict(nat.spline.fit, newdata=data.frame(age=age.grid))
lines(age.grid, preds.poly, lwd=2, col="blue")
lines(age.grid, preds.spline, lwd=2, col="red")
lines(age.grid, preds.nat.spline, lwd=2, col="darkgreen")
```



Note that the natural spline and the polynomial fit are almost identical. Due to its robustness at the boundary we choose the natural spline, and proceed to fit a generalized additive model.

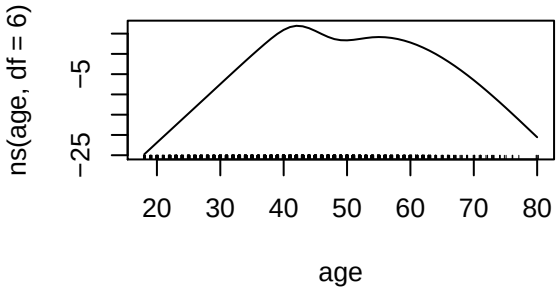
```
gam.fit <- gam(wage~ns(age, df = 6)+education+lvl2+lvl1)
```

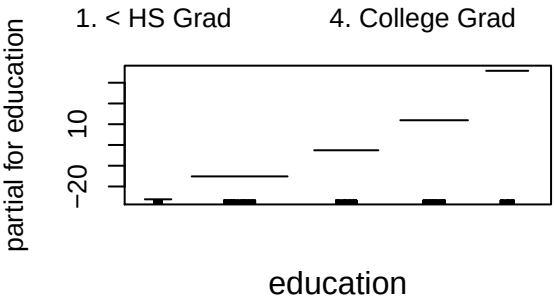
```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
## ignored
```

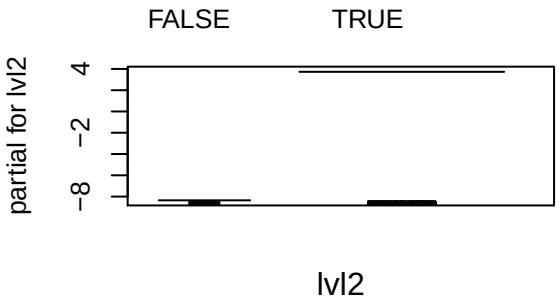
```r
summary(gam.fit)
```

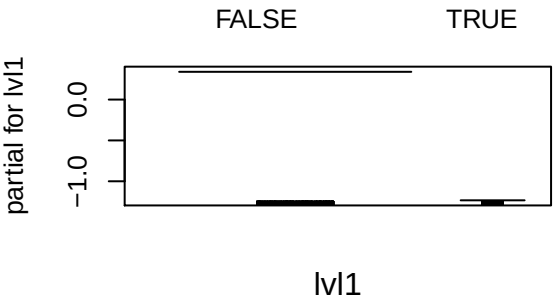
```
##
## Call: gam(formula = wage ~ ns(age, df = 6) + education + lvl2 + lvl1)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -116.075  -19.287   -2.713   14.217  212.816
##
## (Dispersion Parameter for gaussian family taken to be 1212.358)
##
##     Null Deviance: 5222086 on 2999 degrees of freedom
## Residual Deviance: 3621314 on 2987 degrees of freedom
## AIC: 29829.57
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                   Df  Sum Sq Mean Sq  F value Pr(>F)
## ns(age, df = 6)    6  458327   76388  63.0076 <2e-16 ***
## education          4 1052120  263030 216.9572 <2e-16 ***
## lvl2               1   89922   89922  74.1708 <2e-16 ***
## lvl1               1     403     403   0.3324 0.5643
## Residuals       2987 3621314    1212
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
par(mfrow=c(2,2))
plot.Gam(gam.fit)
```


ns(age, df = 6)
−25
−5
20
30
40
50
60
70
80
age


1. < HS Grad
4. College Grad
partial for education
−20
10
education


FALSE
TRUE
partial for lvl2
−8
−2
4
lvl2


FALSE
TRUE
partial for lvl1
−1.0
0.0
lvl1