Wage Example – Step Function

Basic Implementation

We can also try to cut the age into groups and then run a regression.

We cut age into four groups as follows:

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

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

The R-func "cut" divides the interval of the predictor *"age"* into equal lengths between the min (17.9) and max value (80.1)

Wage Example – Step Function

Basic Implementation

Then, we fit a regression model with the 4 age groups.

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

Call:
lm(formula = wage ~ cut(age, 4), data = Wage)

Residuals:
    Min      1Q  Median      3Q     Max
-98.126 -24.803  -6.177  16.493 200.519

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)               94.158      1.476  63.790   <2e-16 ***
cut(age, 4)(33.5,49]      24.053      1.829  13.148   <2e-16 ***
cut(age, 4)(49,64.5]      23.665      2.068  11.443   <2e-16 ***
cut(age, 4)(64.5,80.1]     7.641      4.987   1.532    0.126
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 40.42 on 2996 degrees of freedom
Multiple R-squared:  0.0625,    Adjusted R-squared:  0.06156
F-statistic: 66.58 on 3 and 2996 DF,  p-value: < 2.2e-16
```

Intercept = 1st Interval

2nd Interval compared with 1st: wage increases by $24.05K on average

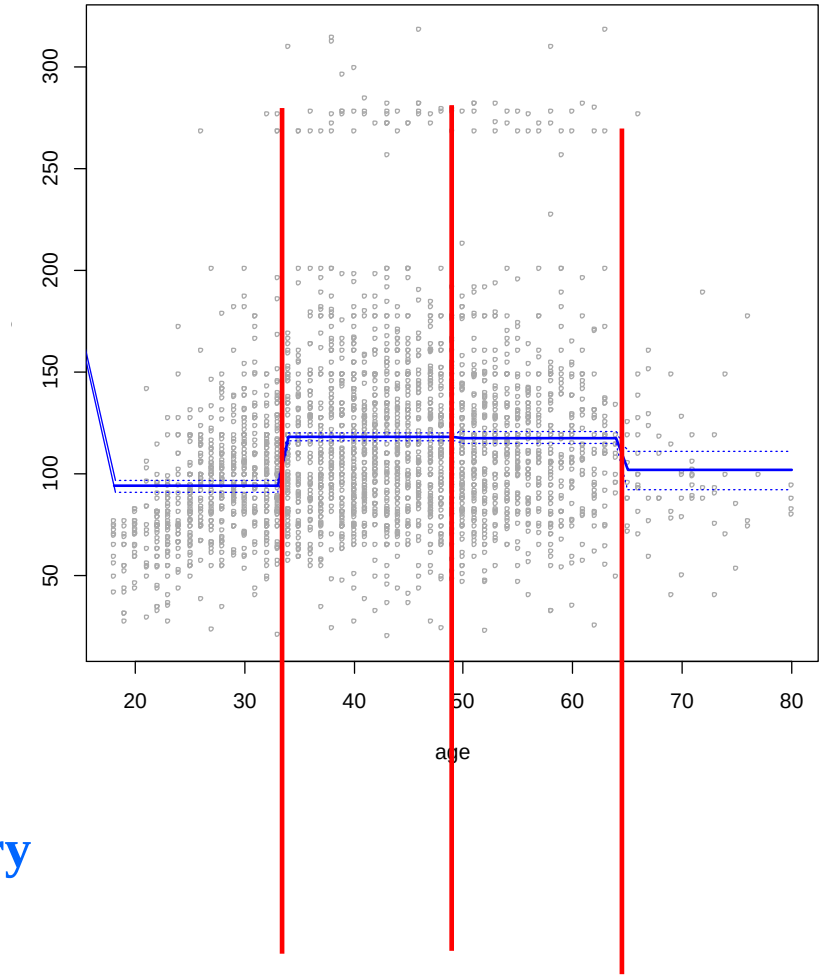
etc

Wage Example – Step Function

Basic Implementation

For visualization of regression model with the 4 age groups, we plot a graph.

```
#Predict the grid
preds<-predict(fit,newdata=list(age=age.grid),se=TRUE)
plot(age,wage,xlim=agelims,cex=.5,col="darkgray")
lines(age.grid,preds$fit,lwd=2,col="blue")
#construct confidence interval
se.bands <-cbind(preds$fit+2*preds$se.fit,preds$fit-2*preds$se.fit)
#plot confidence interval with the existing graph
matlines(age.grid,se.bands,lwd=1, col='blue',lty=3)
```



Note:

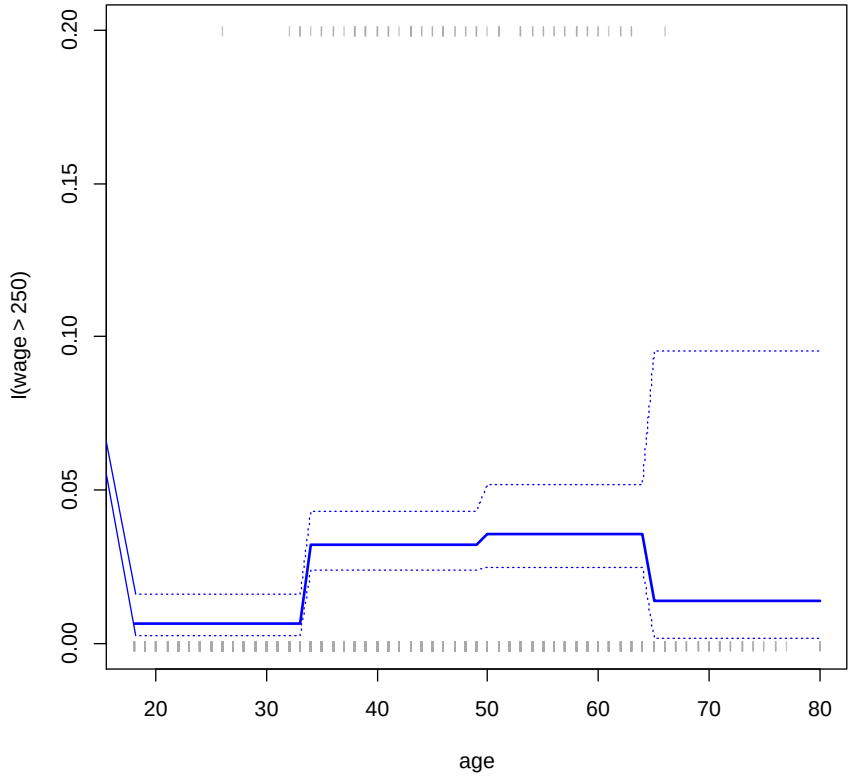
Probably suggests the 2nd cutoff point is NOT necessary

Wage Example – Step Function

Basic Implementation

We can do the same thing with logistic regression

```
#Cut age into four parts and then run a logistic regression
fit<-glm(I(wage>250)~cut(age,4),data=Wage,family=binomial)
#Predict the wage for each age
#Predict the probability for each of age
preds<-predict(fit,newdata=list(age=age.grid),se=TRUE)
pfit<-exp(preds$fit)/(exp(preds$fit)+1)
#Construct the confidence interval for each of age
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))
#Plot a similar plot
plot(age,I(wage>250),xlim=agelims,type='n',cex=.5,ylim=c(0,.2))
points(age, I((wage>250)/5), cex=.5, pch="|", col="darkgray")
lines(age.grid,pfit,lwd=2,col="blue")
matlines(age.grid,se.bands,lwd=1, col='blue',lty=3)
```



Wage Example – Step Function

Basic Implementation

Note:

Can also do ANOVA

If you want to specify your own cutoff points → no auto. procedure, need to write h's yourself

We will learn some more advanced techniques