Predicting Wages: Polynomial Regression & Step Functions

By Vishwa Pardeshi

In this notebook, we will predict the wages of males who reside in the central Atlantic region of the United States.

Learning Outcome: By following the notebook you will be able to

1. Implement Polynomial Regression & Step Funtion
2. Identify optimal degree using cross-validation
3. Perform hypothesis testing using ANOVA

Setup

```
library(ISLR)
library(boot)
```

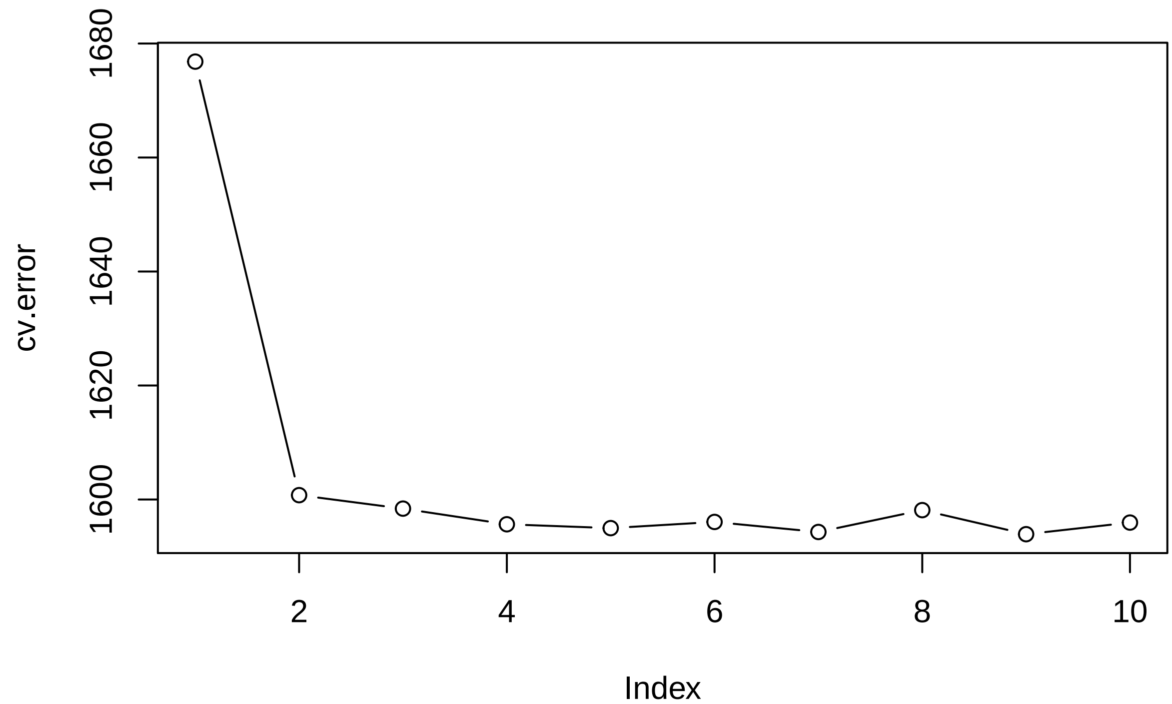
Find optimal degree for polynomial regression.

```
set.seed(1)

cv.error <- rep(0,10)
for (i in 1:10) {
  glm.fit <- glm(wage~poly(age,i), data=Wage)
  cv.error[i] <- cv.glm(Wage, glm.fit, K=10)$delta[1]  # [1]:std, [2]:bias-corrected
}
cv.error
```

```
##  [1] 1676.826 1600.763 1598.399 1595.651 1594.977 1596.061 1594.298
##  [8] 1598.134 1593.913 1595.950
```

Plot the cross validation error



The optimal degree for polynomial regression model is 9 as it lowest cross validation error.

Hypothesis Testing using ANOVA

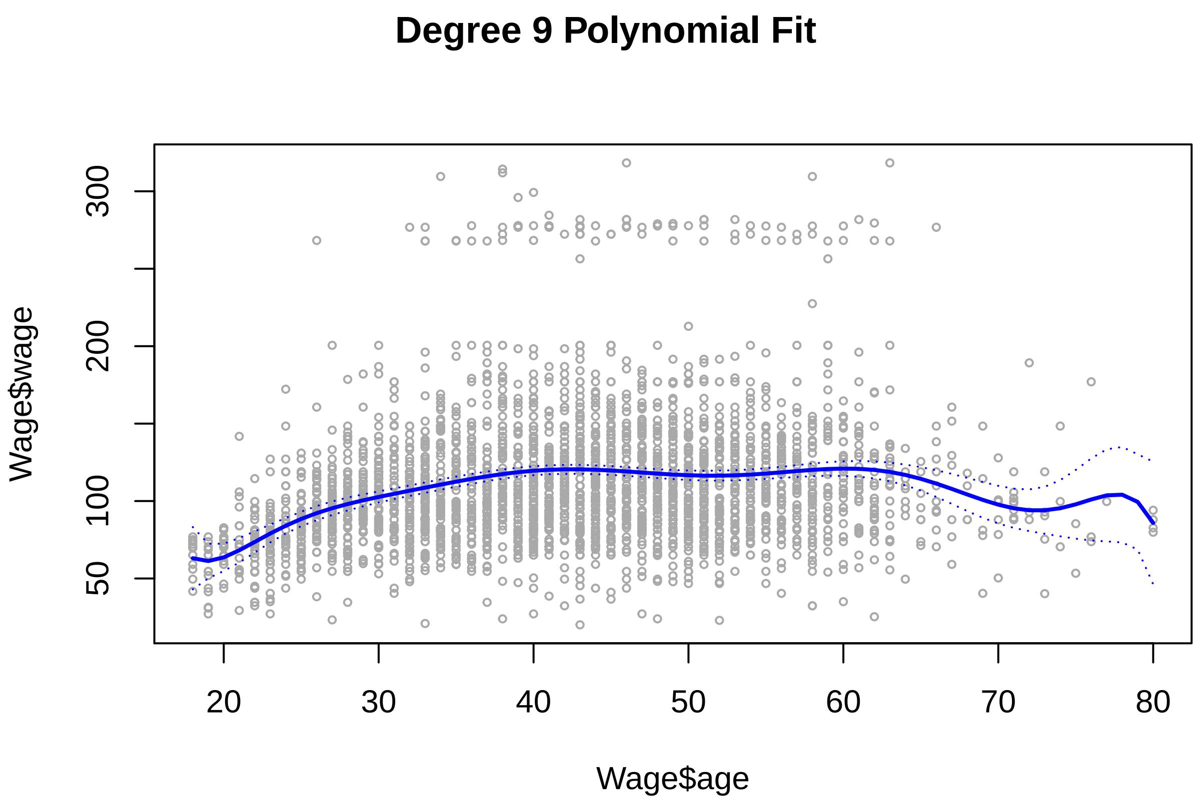
```
fit.01 <- lm(wage~age, data=Wage)
fit.02 <- lm(wage~poly(age,2), data=Wage)
fit.03 <- lm(wage~poly(age,3), data=Wage)
fit.04 <- lm(wage~poly(age,4), data=Wage)
fit.05 <- lm(wage~poly(age,5), data=Wage)
fit.06 <- lm(wage~poly(age,6), data=Wage)
fit.07 <- lm(wage~poly(age,7), data=Wage)
fit.08 <- lm(wage~poly(age,8), data=Wage)
fit.09 <- lm(wage~poly(age,9), data=Wage)
fit.10 <- lm(wage~poly(age,10), data=Wage)
anova(fit.01,fit.02,fit.03,fit.04,fit.05,fit.06,fit.07,fit.08,fit.09,fit.10)
```

```
## 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)
## Model  6: wage ~ poly(age, 6)
## Model  7: wage ~ poly(age, 7)
## Model  8: wage ~ poly(age, 8)
## Model  9: wage ~ poly(age, 9)
## Model 10: wage ~ poly(age, 10)
##    Res.Df     RSS Df Sum of Sq        F    Pr(>F)    
## 1    2998 5022216                                    
## 2    2997 4793430  1    228786 143.7638 < 2.2e-16 ***
## 3    2996 4777674  1     15756   9.9005  0.001669 ** 
## 4    2995 4771604  1      6070   3.8143  0.050909 .  
## 5    2994 4770322  1      1283   0.8059  0.369398    
## 6    2993 4766389  1      3932   2.4709  0.116074    
## 7    2992 4763834  1      2555   1.6057  0.205199    
## 8    2991 4763707  1       127   0.0796  0.777865    
## 9    2990 4756703  1      7004   4.4014  0.035994 *  
## 10   2989 4756701  1         3   0.0017  0.967529    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOVA hypothesis shows statistically significant result for degree 3 and 9. Since we noticed that the cv error for degree 9 was the lowest, we pick degree 9 as the optimal degree for our polynomial regression model.

```r
agelims <- range(Wage$age)
age.grid <- seq(agelims[1], agelims[2])
preds <- predict(fit.09, newdata=list(age=age.grid), se=TRUE)
se.bands <- preds$fit + cbind(2*preds$se.fit, -2*preds$se.fit)
par(mfrow=c(1,1), mar=c(4.5,4.5,1,1), oma=c(0,0,4,0))
plot(Wage$age, Wage$wage, xlim=agelims, cex=0.5, col="darkgrey")
title("Degree 9 Polynomial Fit", outer=TRUE)
lines(age.grid, preds$fit, lwd=2, col="blue")
matlines(age.grid, se.bands, lwd=1, col="blue", lty=3)
```



Find optimal cut for step function using CV

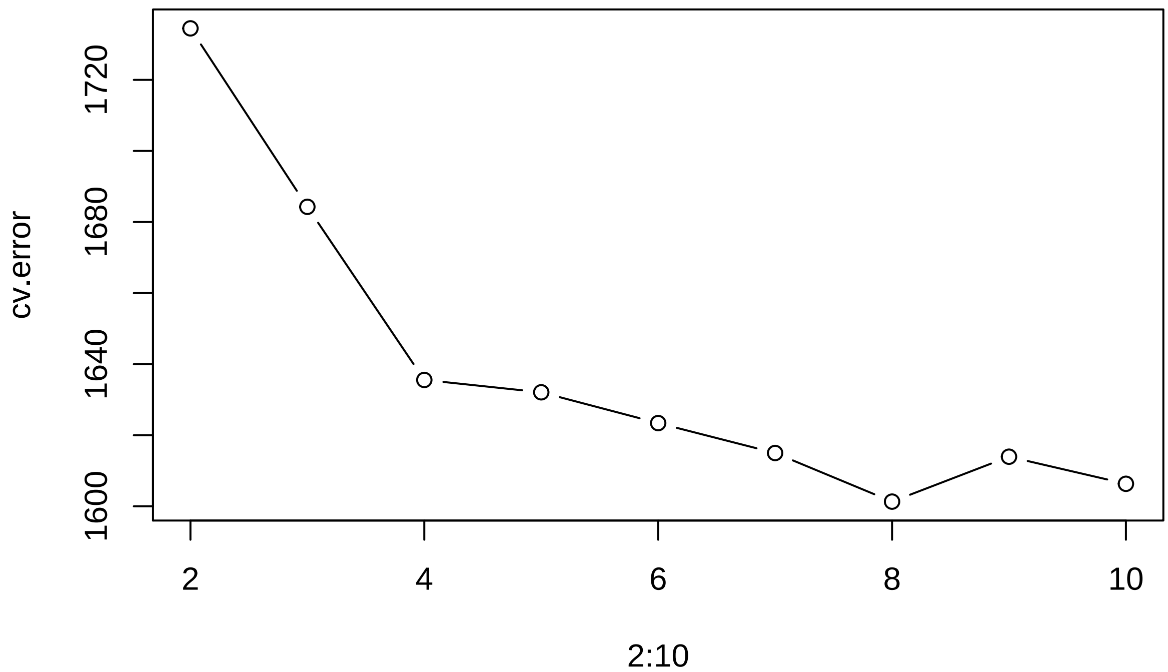
```
set.seed(1)

# cross-validation
cv.error <- rep(0,9)
for (i in 2:10) {
  Wage$age.cut <- cut(Wage$age,i)
  glm.fit <- glm(wage~age.cut, data=Wage)
  cv.error[i-1] <- cv.glm(Wage, glm.fit, K=10)$delta[1]  # [1]:std, [2]:bias-corrected
}
cv.error
```

```
## [1] 1734.489 1684.271 1635.552 1632.080 1623.415 1614.996 1601.318 1613.954
## [9] 1606.331
```

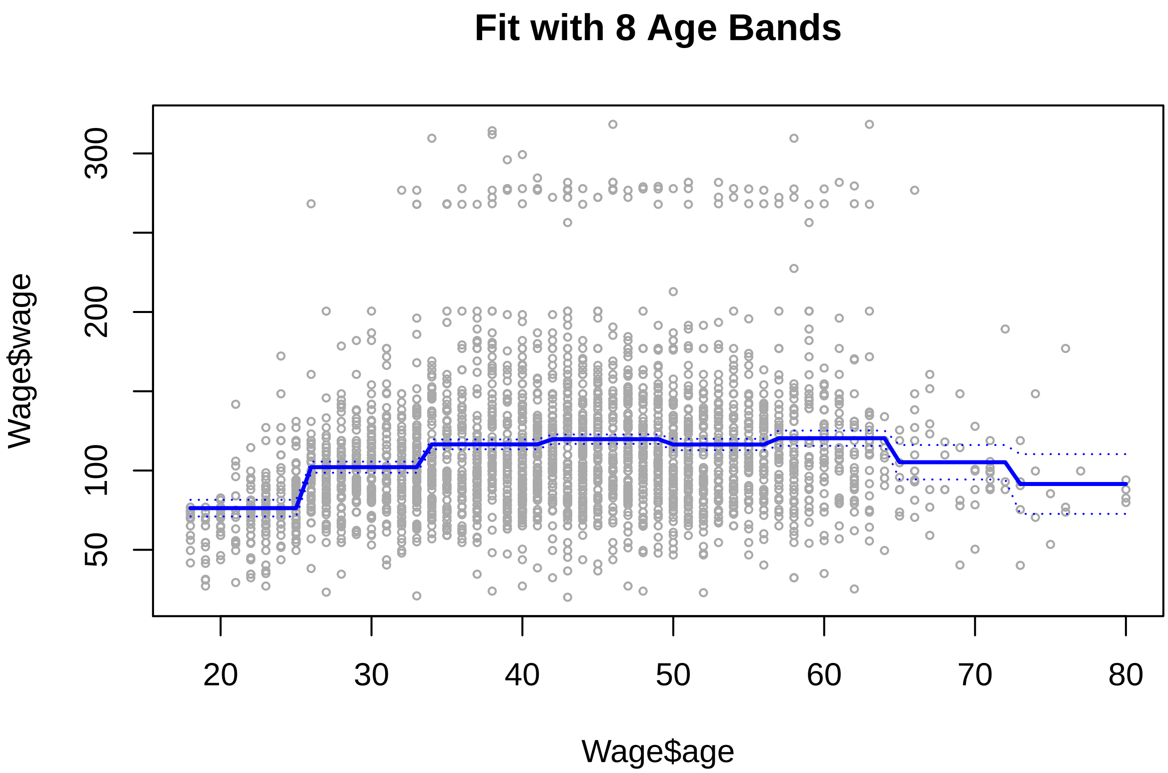
Plot for cv error

```
plot(2:10, cv.error, type="b")
```



Implement step functions

```
cut.fit <- glm(wage~cut(age,8), data=Wage)
preds <- predict(cut.fit, newdata=list(age=age.grid), se=TRUE)
se.bands <- preds$fit + cbind(2*preds$se.fit, -2*preds$se.fit)
plot(Wage$age, Wage$wage, xlim=agelims, cex=0.5, col="darkgrey")
title("Fit with 8 Age Bands")
lines(age.grid, preds$fit, lwd=2, col="blue")
matlines(age.grid, se.bands, lwd=1, col="blue", lty=3)
```



Fit with 8 Age Bands
Wage$wage
50
100
200
300
20
30
40
50
60
70
80
Wage$age