Predicting Nitrogen Oxide concentration in Boston: Splines

By Vishwa Pardeshi

In this notebook, we will predict the concentration of nitrogen oxides in Boston housing (nox) using Boston Housing Price dataset. We will use the weighted mean of distances to five Boston employment centers i.e. dis as the predictor variable.

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

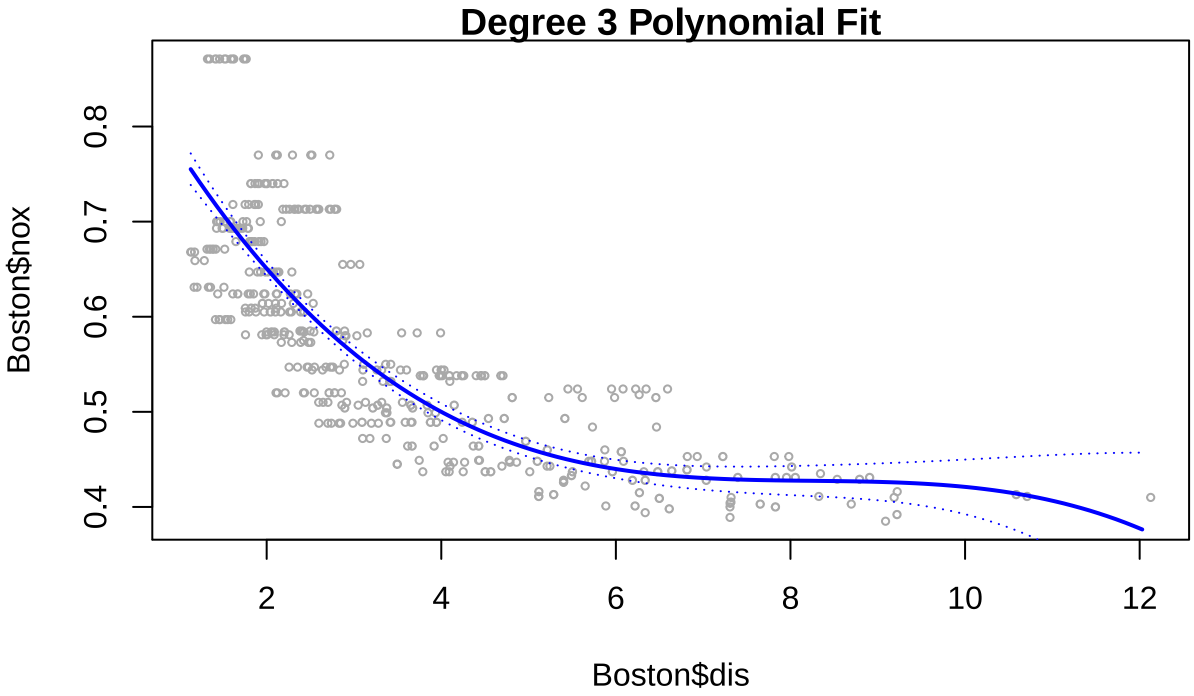
1. Implement Polynomial Regression & Spline
2. Identify optimal degree of freedom using cross-validation

Setup

```r
library(ISLR)
library(MASS)
library(splines)
```

Implement & plot cubic polynomial regression model

```r
fit.cubic <- lm(nox~poly(dis,3), data=Boston)
dislims <- range(Boston$dis)
dis.grid <- seq(dislims[1], dislims[2], 0.1)
preds <- predict(fit.cubic, newdata=list(dis=dis.grid), se=TRUE)
se.bands <- preds$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(Boston$dis, Boston$nox, xlim=dislims, cex=0.5, col="darkgrey")
title("Degree 3 Polynomial Fit")
lines(dis.grid, preds$fit, lwd=2, col="blue")
matlines(dis.grid, se.bands, lwd=1, col="blue", lty=3)
```



```
summary(fit.cubic)
```

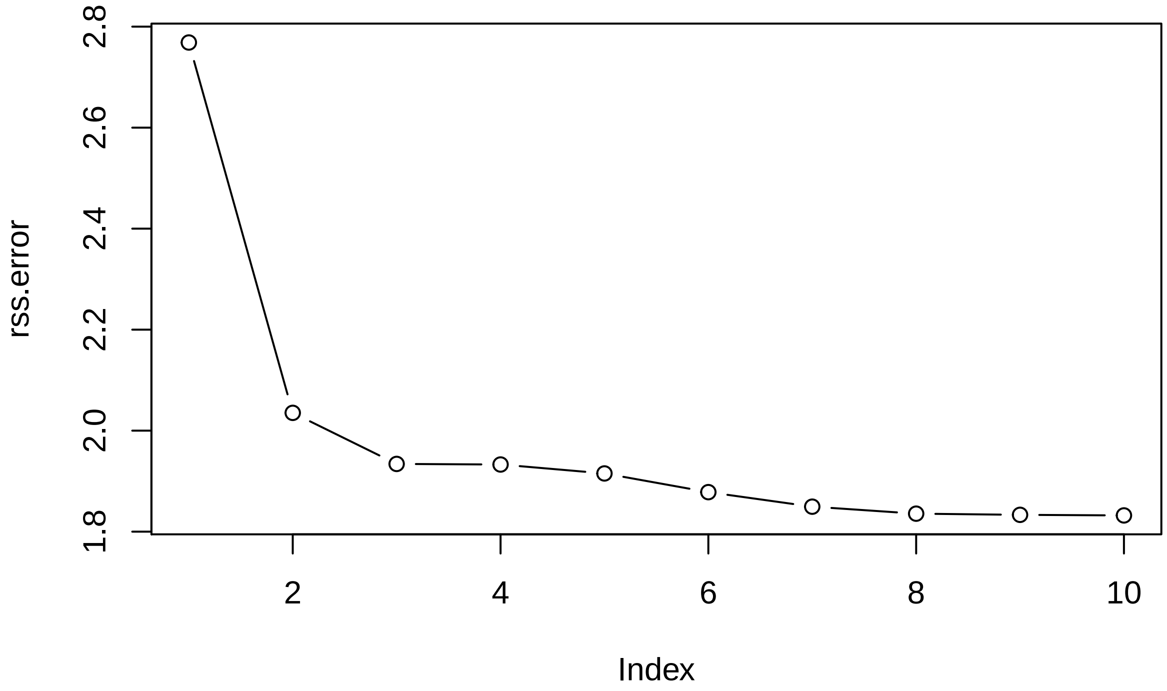
```
##
## Call:
## lm(formula = nox ~ poly(dis, 3), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.121130 -0.040619 -0.009738  0.023385  0.194904
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.554695   0.002759 201.021  < 2e-16 ***
## poly(dis, 3)1 -2.003096   0.062071 -32.271  < 2e-16 ***
## poly(dis, 3)2  0.856330   0.062071  13.796  < 2e-16 ***
## poly(dis, 3)3 -0.318049   0.062071  -5.124 4.27e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06207 on 502 degrees of freedom
## Multiple R-squared:  0.7148, Adjusted R-squared:  0.7131
## F-statistic: 419.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Plot the polynomial fits for a range of different polynomial degrees

```
rss.error <- rep(0,10)
for (i in 1:10) {
  lm.fit <- lm(nox~poly(dis,i), data=Boston)
  rss.error[i] <- sum(lm.fit$residuals^2)
}
rss.error
```

```
##  [1] 2.768563 2.035262 1.934107 1.932981 1.915290 1.878257 1.849484
##  [8] 1.835630 1.833331 1.832171
```

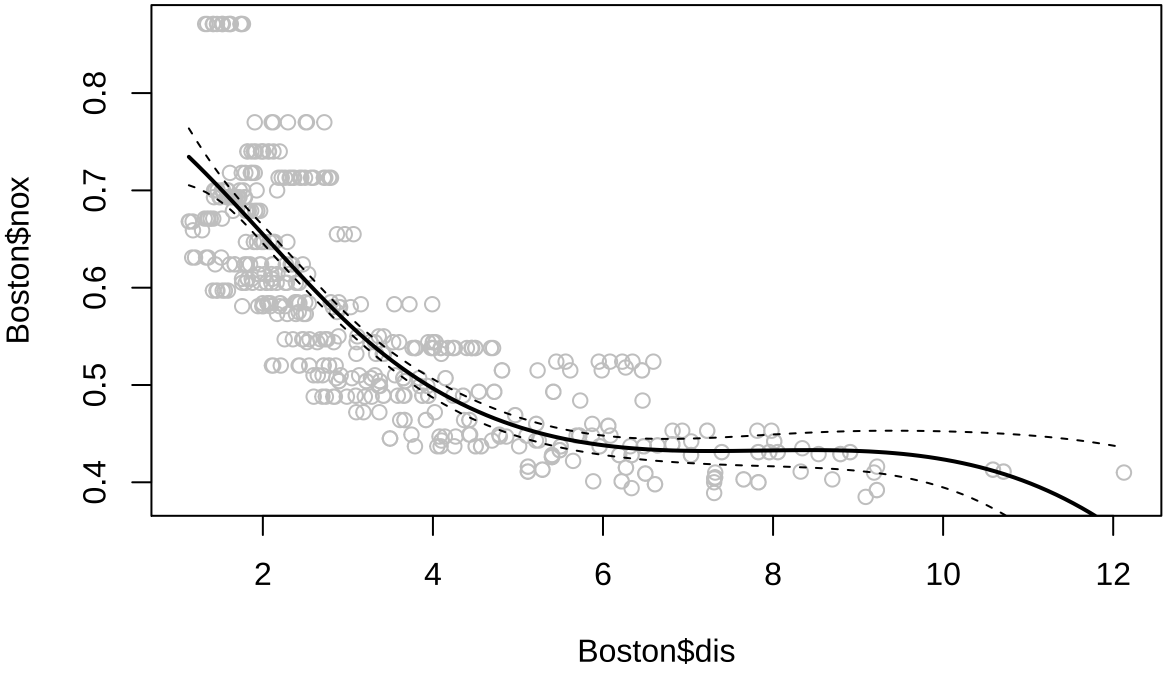
Plot the RSS error



The lowest RSS is for the higher order of polynomial regression model.

fit a regression spline to predict nox using dis.

```
fit.spline <- lm(nox~bs(dis, df=4), data=Boston)
pred <- predict(fit.spline, newdata=list(dis=dis.grid), se=T)
plot(Boston$dis, Boston$nox, col="gray")
lines(dis.grid, pred$fit, lwd=2)
lines(dis.grid, pred$fit+2*pred$se, lty="dashed")
lines(dis.grid, pred$fit-2*pred$se, lty="dashed")
```



```
#set df to select knots at uniform quantiles of `dis`
attr(bs(Boston$dis,df=4),"knots")  # only 1 knot at 50th percentile
```

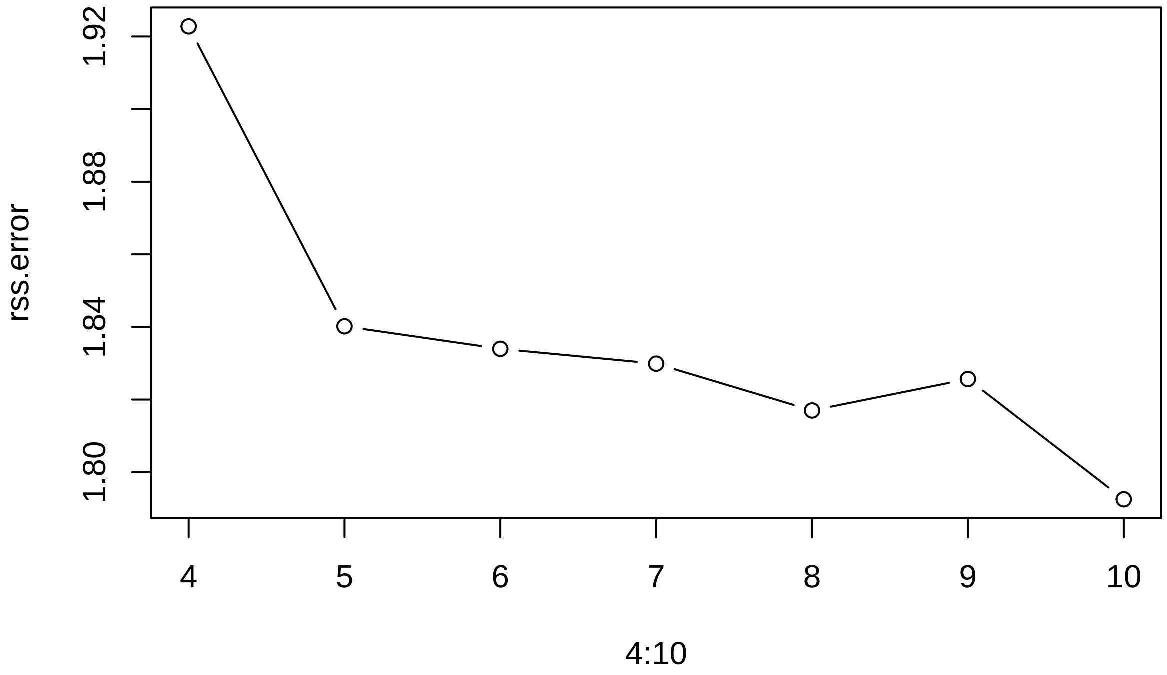
```
##     50%
## 3.20745
```

Fit a regression spline for a range of degrees of freedom

```
set.seed(1)
rss.error <- rep(0,7)
for (i in 4:10) {
  fit.sp <- lm(nox~bs(dis, df=i), data=Boston)
  rss.error[i-3] <- sum(fit.sp$residuals^2)
}
rss.error
```

```
## [1] 1.922775 1.840173 1.833966 1.829884 1.816995 1.825653 1.792535
```

```
plot(4:10, rss.error, type="b")
```



RSS decreases on train set as the model becomes more flexible.