STAT 435 Quiz 4

Pat McCornack

2022-11-29

Q1 Polynomial Regression

Comparing models using the anova() function shows that no value is gained beyond the third degree polynomial. The MSE for the third degree polynomial is 1.053 * 10^29. Plotting the predictions confirms that the third degree polynomial does provide a good fit. The points are the real data and the line is the predicted values.

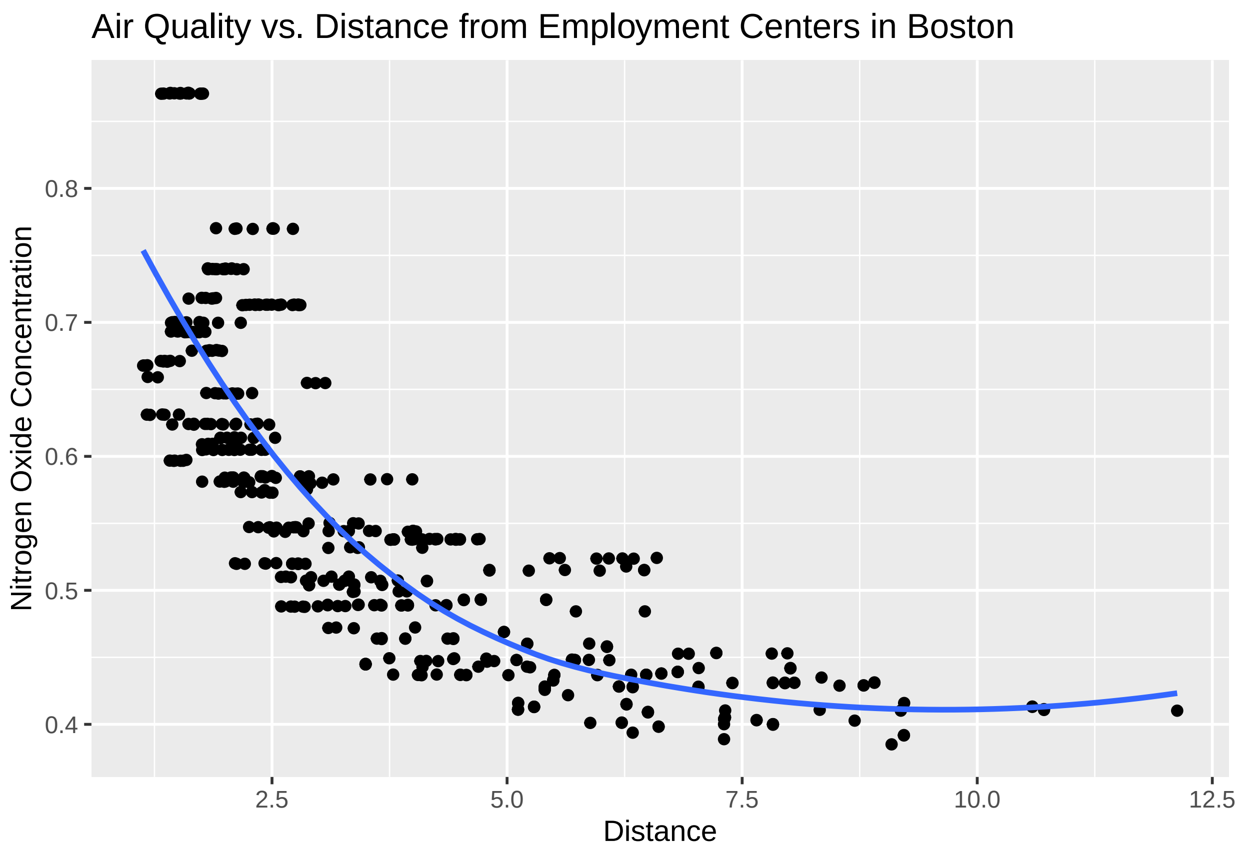
```
library(ISLR)
library(ggplot2)
library(MASS)
data(Boston)



poly.model.1 <- lm(nox ~ poly(dis, 1), data = Boston)
poly.model.2 <- lm(nox ~ poly(dis, 2), data = Boston)
poly.model.3 <- lm(nox ~ poly(dis, 3), data = Boston)
poly.model.4 <- lm(nox ~ poly(dis, 4), data = Boston)
poly.model.5 <- lm(nox ~ poly(dis, 5), data = Boston)

anova(poly.model.1, poly.model.2, poly.model.3, poly.model.4, poly.model.5)
```

```
## Analysis of Variance Table
##
## Model 1: nox ~ poly(dis, 1)
## Model 2: nox ~ poly(dis, 2)
## Model 3: nox ~ poly(dis, 3)
## Model 4: nox ~ poly(dis, 4)
## Model 5: nox ~ poly(dis, 5)
##   Res.Df    RSS Df Sum of Sq        F    Pr(>F)
## 1    504 2.7686
## 2    503 2.0353  1   0.73330 191.4334 < 2.2e-16 ***
## 3    502 1.9341  1   0.10116  26.4073 3.972e-07 ***
## 4    501 1.9330  1   0.00113   0.2938   0.58804
## 5    500 1.9153  1   0.01769   4.6185   0.03211 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Compute MSE
nox.pred <- predict(poly.model.3, newdata = Boston)
n <- dim(Boston)[1]
MSE <- (sum(Boston$nox - nox.pred))^2 / n
MSE
```

```
## [1] 1.05308e-29
```

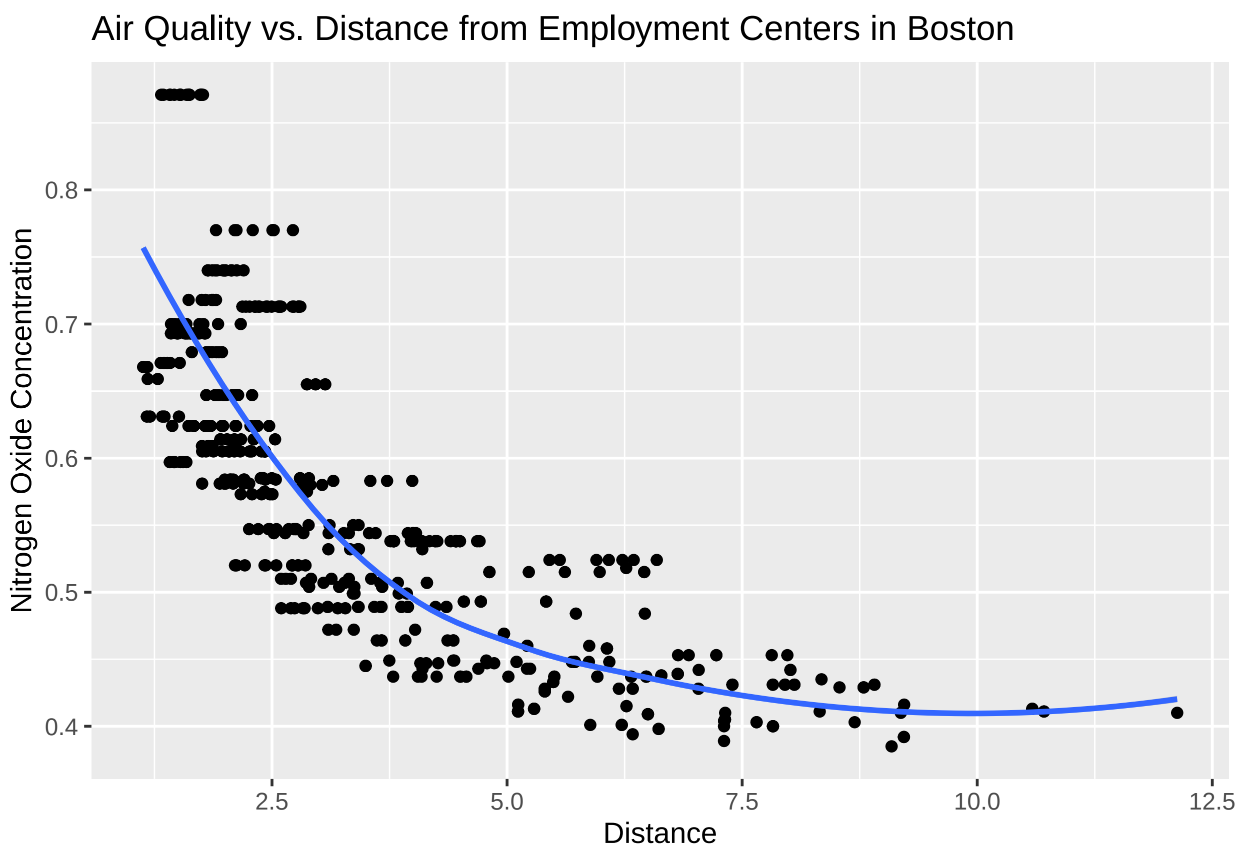
```
# Plot the prediction against the spaces
ggplot(data = Boston) +
  geom_jitter(aes(x=dis, y=nox)) +
  geom_smooth(aes(x=dis, y=nox.pred), se=F) +
  labs(title = "Air Quality vs. Distance from Employment Centers in Boston", x = "Distance", y = "Nitrog
```



Q2 Cubic Spline

The mean squared error using a cubic natural spline model is 1.05 * 10^29.

```
library(splines)
range.dis <- range(Boston$dis)
d <- (range.dis[2] - range.dis[1]) / 4

spline.model <- lm(nox ~ ns(dis, knots=c(3.33, 5.53, 7.73, 9.93)), data = Boston)

spline.pred <- predict(spline.model, newdata = Boston)

# Plot the prediction against the data
ggplot(data = Boston) +
  geom_point(aes(x=dis, y=nox)) +
  geom_smooth(aes(x=dis, y=spline.pred)) +
  labs(title = "Air Quality vs. Distance from Employment Centers in Boston", x = "Distance", y = "Nitrog
```




```
MSE.spline <- (sum(Boston$nox - spline.pred))^2 / n
MSE
```

```
## [1] 1.05308e-29
```

Q3 Cross validating the number of knots

The best fitting natural spline model that was obtained using cross-validation used 2 knots and had an MSE of .0216.

```
attach(Boston)
knots <- c(2,4,6,8,10)
k <- 5  # Number of folds
ncv <- ceiling(nrow(Boston)/k)  # Number observations per fold
cv.ind = rep(1:k, ncv)  # Used to assign folds to observations
cv.ind.rand = sample(cv.ind, nrow(Boston), replace = F)  # Randomize cv.ind

# Vectors used to track error
cv.error <- c()  # Avg MSE between folds for a given knot
MSE.cv <- c()  # MSE for a given fold


for(i in knots){  # For every knot in the list
  for(j in 1:k){  # For each fold
    dis.train <- Boston[cv.ind.rand != j, 'dis']  # Training predictor
    nox.train <- Boston[cv.ind.rand != j, 'nox']  # Training response
```

```
    model <- lm(nox.train ~ ns(dis.train, df = i), data=Boston)

    dis.test <- data.frame(Boston[cv.ind.rand == j, 'dis'])  # Test predictor
    colnames(dis.test) <- 'dis'
    nox.test <- Boston[cv.ind.rand == j, 'nox']  # Test response

    cv.nox.pred <- predict(model, newdata=data.frame())  # Predict response

    MSE.cv[j] <- mean((nox.test - cv.nox.pred)^2)  # Calculate MSE for that fold
  }
  cv.error[i] <- mean(MSE.cv)
}

# Return the lowest MSE:
(min <- which.min(cv.error))
```

```
## [1] 2
```

```
cv.error[min]
```

```
## [1] 0.02183062
```