```r
library(ISLR2)
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(splines)
```

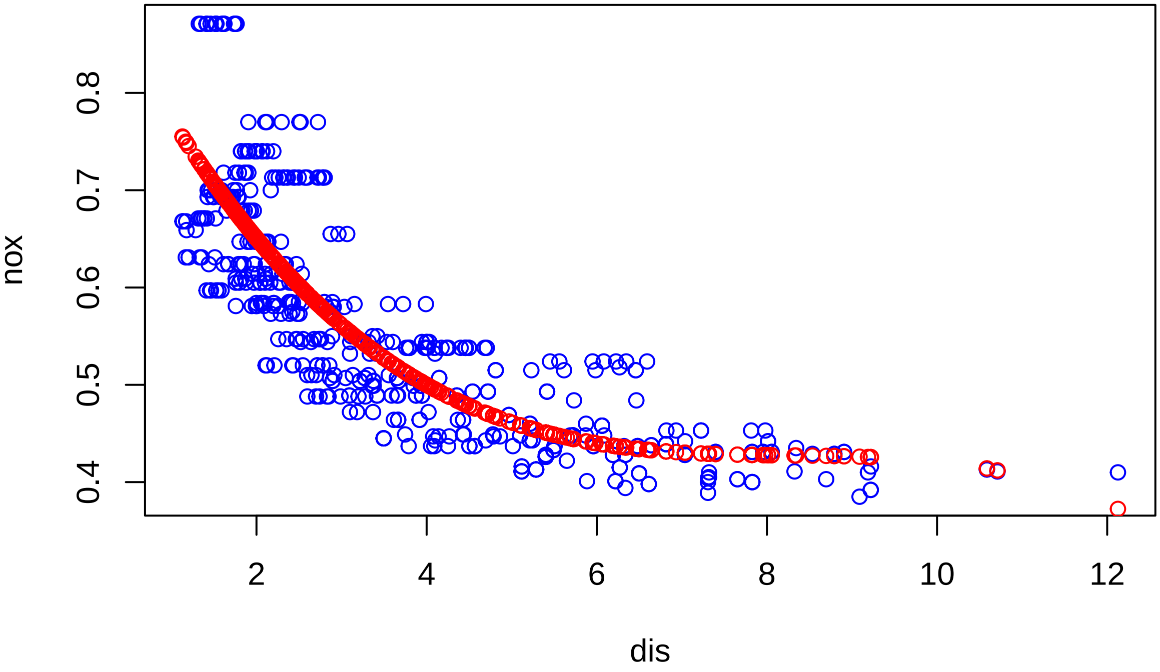
```r
data = Boston
dis = data$dis
nox = data$nox
set.seed(666)
```

#a

```r
model1 = lm(nox ~ poly(dis ,degree = 3))
model1
```

```
##
## Call:
## lm(formula = nox ~ poly(dis, degree = 3))
##
## Coefficients:
##            (Intercept)  poly(dis, degree = 3)1  poly(dis, degree = 3)2
##                 0.5547                 -2.0031                  0.8563
## poly(dis, degree = 3)3
##                -0.3180
```

```r
plot(dis,nox,col = "blue")
points(dis, fitted.values(model1), col = "red")
```

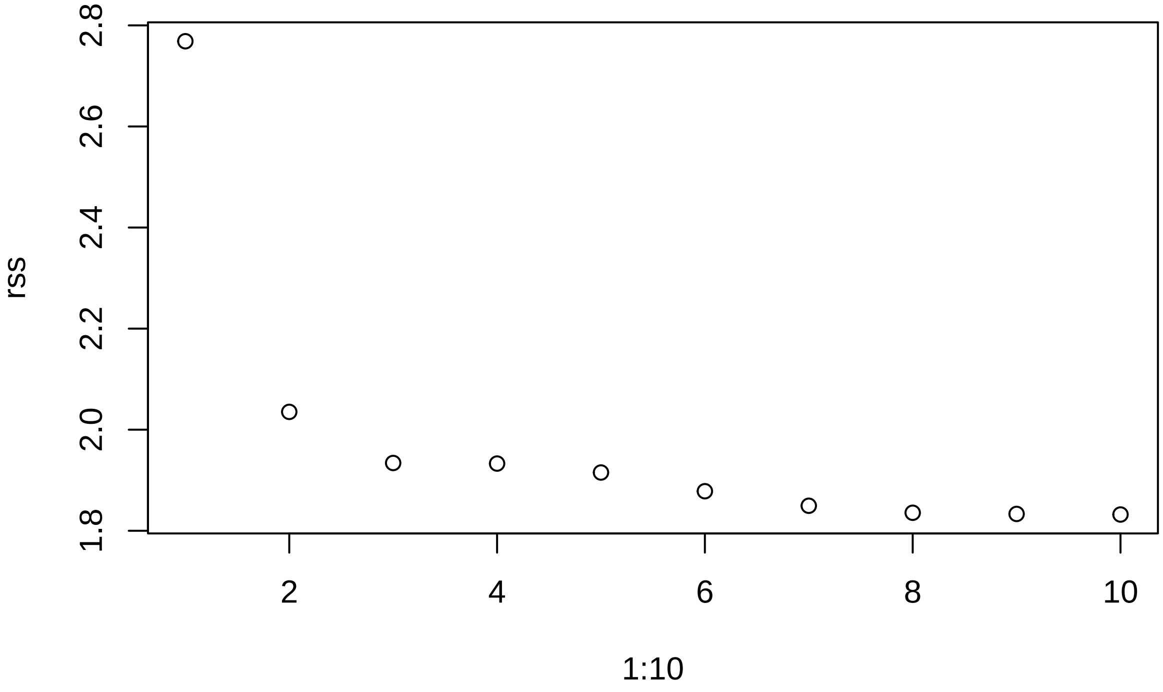


#b

```r
rss = c()
for (i in (1:10)){
  modelx = lm(nox ~ poly(dis, degree = i))
  rss[i] = sum(residuals(modelx)^2)
}
rss
```

```
##  [1] 2.768563 2.035262 1.934107 1.932981 1.915290 1.878257 1.849484 1.835630
##  [9] 1.833331 1.832171
```

```
plot(1:10,rss)
```

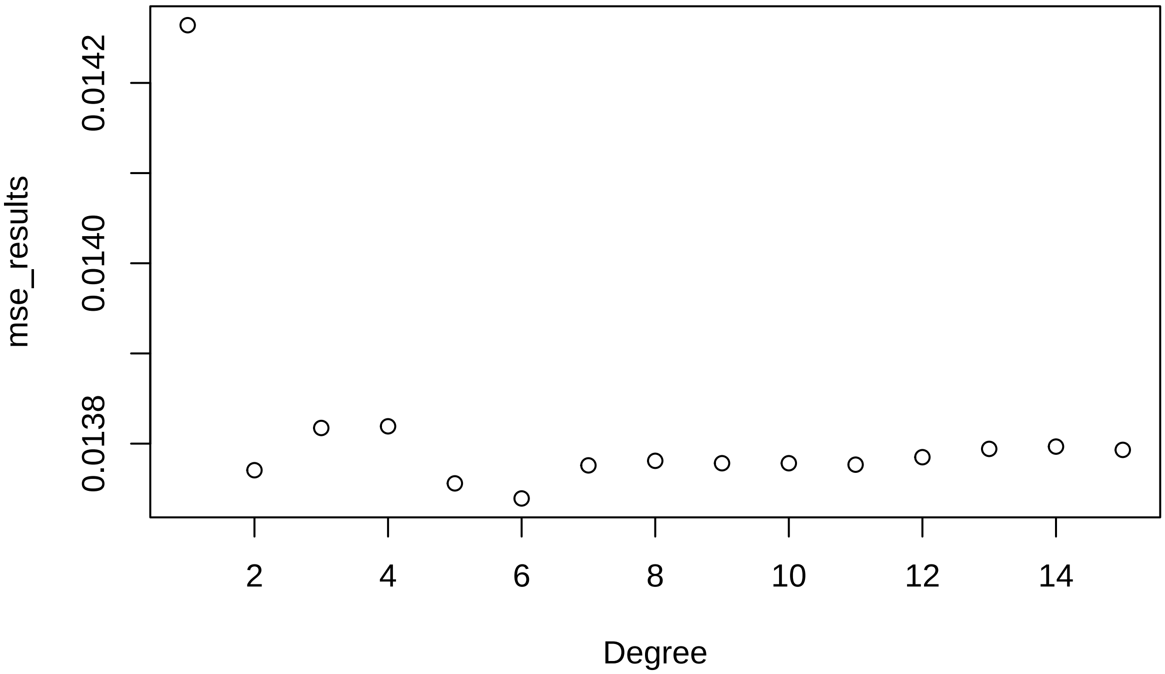


#c

```
folds = createFolds(nox, k = 5)
mse_results = c()
for (i in 1:15) {
  fold_mse = c()
  for (j in 1:5) {
    train_indices = unlist(folds[-j])
    test_indices = unlist(folds[j])
    modelx = lm(nox[train_indices] ~ poly(dis[train_indices], degree = i))
    predictions = predict(modelx, data.frame(dis[test_indices]))
    fold_mse[j] = mean((predictions - nox[test_indices])^2)
  }
  mse_results[i] = mean(fold_mse)
}
mse_results
```

```
##  [1] 0.01426399 0.01377050 0.01381735 0.01381920 0.01375593 0.01373921
##  [7] 0.01377586 0.01378093 0.01377826 0.01377827 0.01377667 0.01378497
## [13] 0.01379417 0.01379671 0.01379306
```

```
plot((1:15), mse_results,xlab='Degree')
```



#d

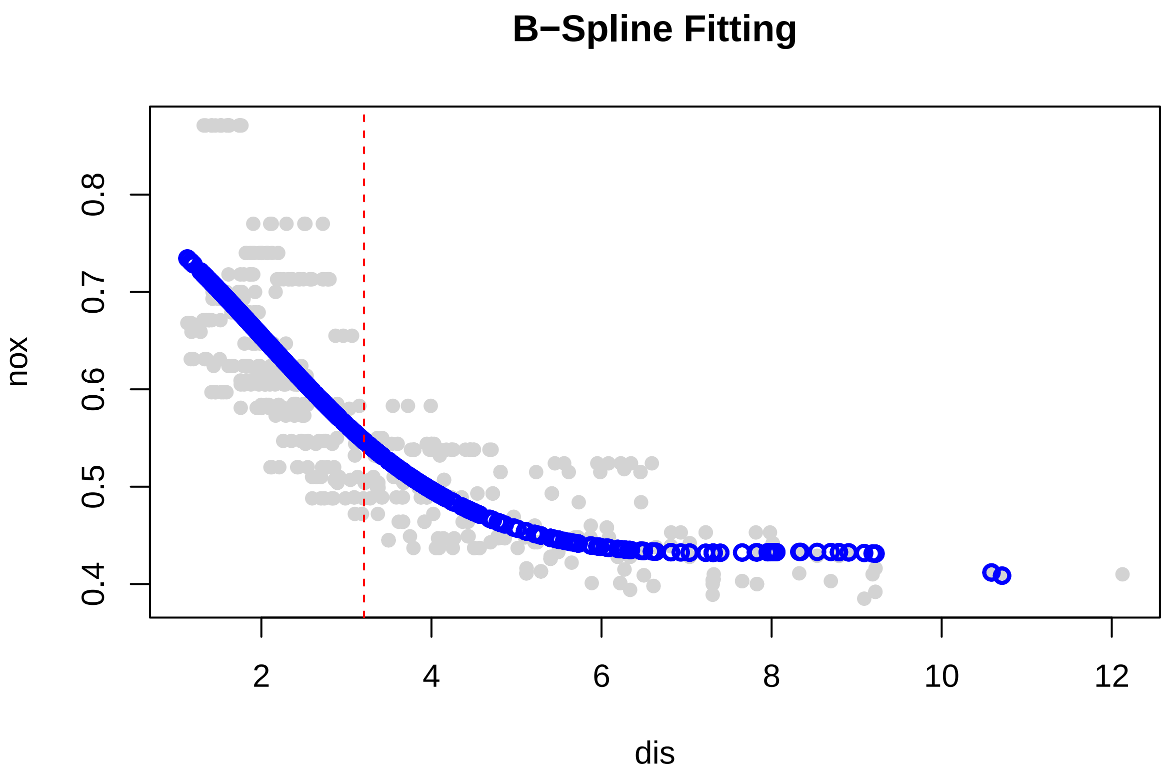
```
bs_s = bs(dis,df = 4)
model2 = lm(nox ~ bs_s)
model2
```

```
##
## Call:
## lm(formula = nox ~ bs_s)
##
## Coefficients:
## (Intercept)        bs_s1        bs_s2        bs_s3        bs_s4
##      0.7345      -0.0581      -0.4636      -0.1998      -0.3888
```

```
knots = attr(bs_s, "knots")
knots
```

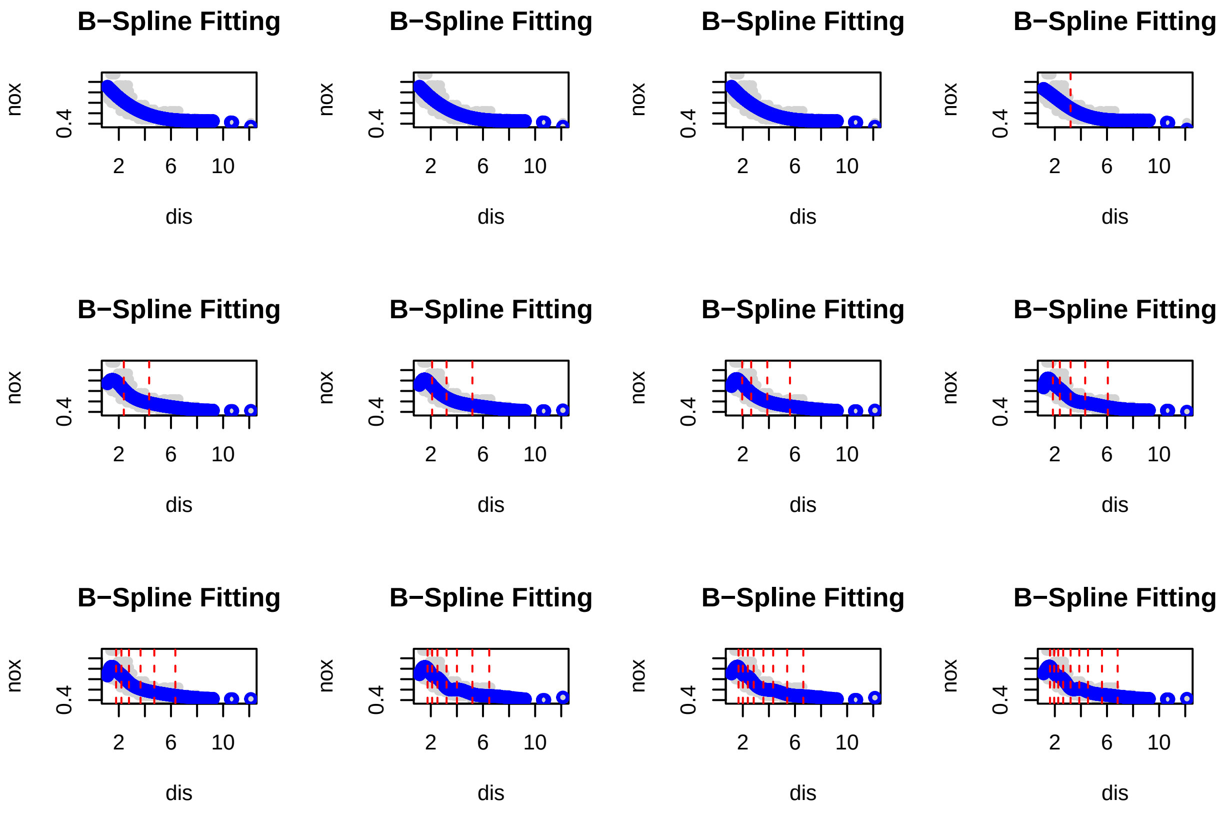
```
## [1] 3.20745
```

```
predictions = predict(model2)
plot(dis, nox, main = "B-Spline Fitting", pch = 16, col = "lightgray")
points(dis, predictions, col = "blue", lwd = 2)
abline(v = knots, col = 'red',lty = 2)
```



#e

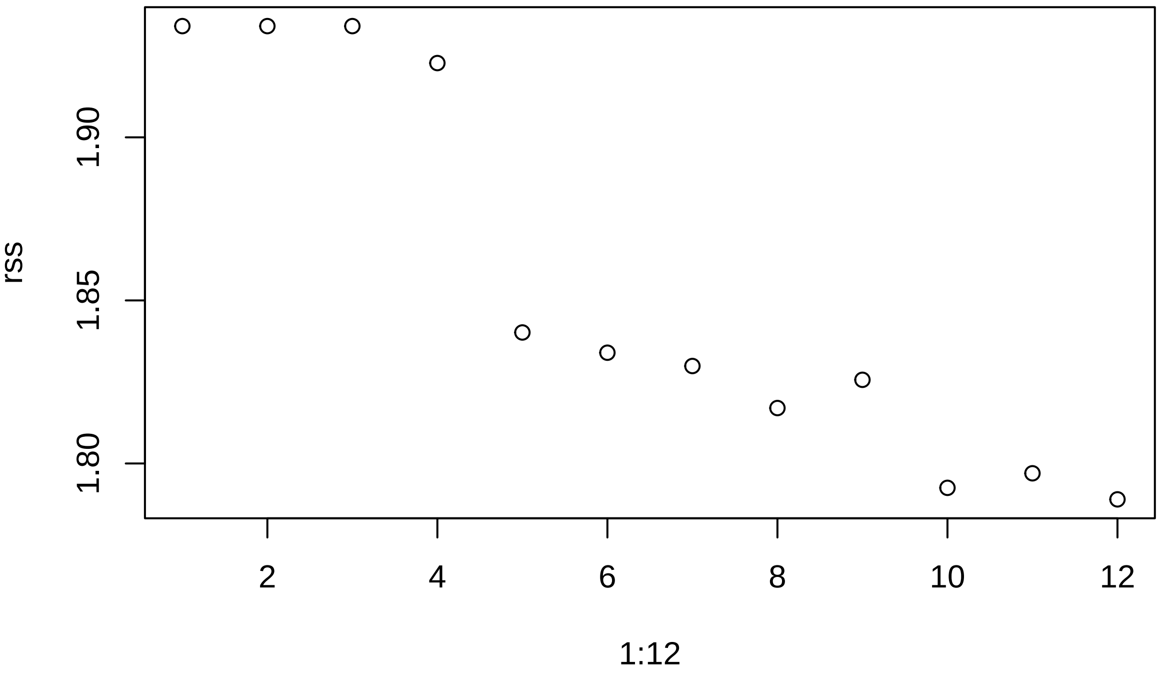
```
rss = c()
par(mfrow = c(3, 4))
for (i in (1:12)){
  bs_s = bs(dis,df = i)
  modelx = lm(nox ~ bs_s)
  rss[i] = sum(residuals(modelx)^2)
  predictions = predict(modelx)
  knots = attr(bs_s, "knots")
  plot(dis, nox, main = "B-Spline Fitting", pch = 16, col = "lightgray")
  points(dis, predictions, col = "blue", lwd = 2)
  abline(v = knots, col = 'red',lty = 2)
}
```



```
rss
```

```
##  [1] 1.934107 1.934107 1.934107 1.922775 1.840173 1.833966 1.829884 1.816995
##  [9] 1.825653 1.792535 1.796992 1.788999
```

```
par(mfrow = c(1, 1))
plot(1:12, rss)
```

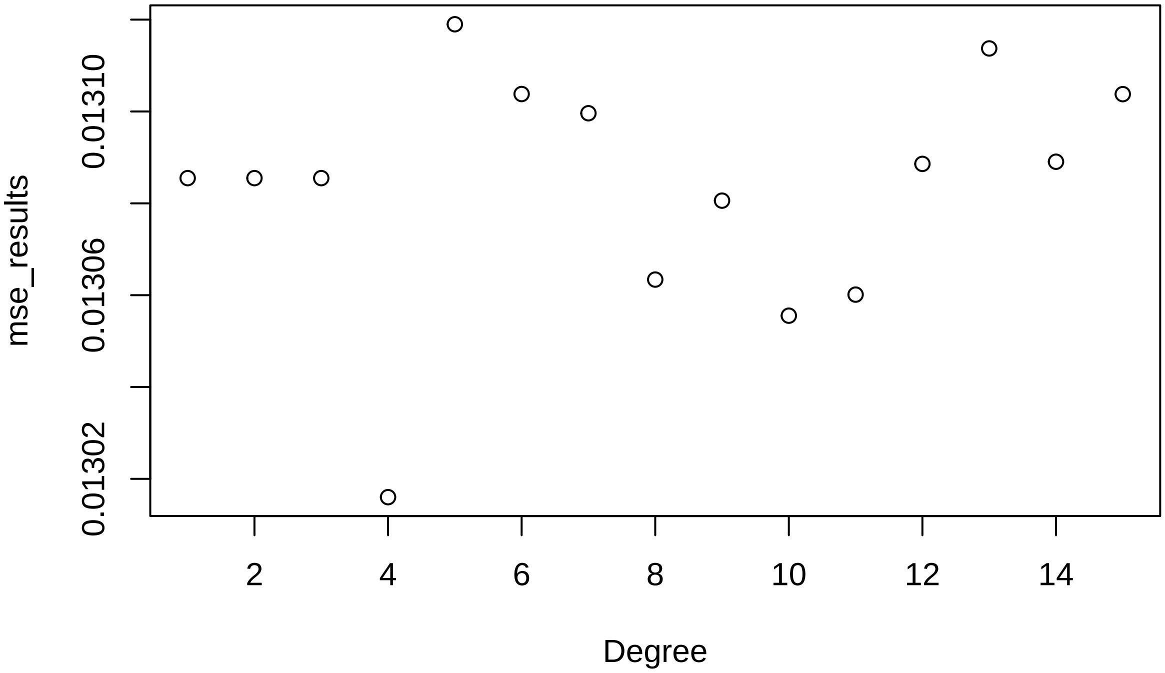


#f

```
folds = createFolds(nox, k = 5)
mse_results = c()
for (i in 1:15) {
  fold_mse = c()
  for (j in 1:5) {
    train_indices = unlist(folds[-j])
    test_indices = unlist(folds[j])
    bs_s = bs(dis[train_indices],df = i)
    modelx = lm(nox[train_indices] ~ bs_s)
    predictions = predict(modelx, data.frame(dis[test_indices]))
    fold_mse[j] = mean((predictions - nox[test_indices])^2)
  }
  mse_results[i] = mean(fold_mse)
}
mse_results
```

```
##  [1] 0.01308549 0.01308549 0.01308549 0.01301602 0.01311899 0.01310380
##  [7] 0.01309962 0.01306340 0.01308059 0.01305554 0.01306012 0.01308859
## [13] 0.01311373 0.01308906 0.01310378
```

```
plot((1:15), mse_results,xlab='Degree')
```


mse_results
Degree