RSM8512 Assignment - Non-Linearity

Yanbing Chen 1009958752 2023/11/24

Question 2 [15 marks]

Suppose that a curve $\hat{g}$ is computed to smoothly fit a set of n points using the following formula:

$$\hat{g} = \text{argmin}(\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int [g^m(x)]^2 \, dx)$$

where g^m represents the mth derivative of g (and $g^0 = g$). Provide example sketches of $\hat{g}$ in each of the following scenarios.

(a) $\lambda = \infty, m = 0$

Answer: The minimization is purely based on the fit to the data points, ignoring the smoothness. The regularization term λ is not relevant here b/c $m = 0$ (we are looking at $g(x)$ itself, not its derivatives). In this case, $\hat{g}(x)$ would closely fit the data points but won't necessarily be proportional to x^2 since the regularization doesn't enforce any smoothness. Summarily, $g(x) = k$ because RSS term is ignored and $g(x) = k$ would minimize the area uner the cruve of $g^{(0)}$.

(b) $\lambda = \infty, m = 1$

Answer: $g(x)$ would be quadratic to minimize the area under the curve of its first derivative. This scenario minimizes the first derivative, which would lead to a function where the first derivative is constant. However, b/c λ is infinite, the solution would be to have the first derivative as close to zero as possible. Since $\hat{g}(x)$ must be proportional to x^2, the only solution that satisfies this with a constant (and effectively zero) first derivative is $\hat{g}(x) = 0$.

(c) $\lambda = \infty, m = 2$

Answer: $g(x)$ would be cubic to minimize the area under the curve of its second derivative. The minimization of the second derivative (the curvature) is enforced. Since $\hat{g}(x)$ is proportional to x^2, its second derivative is a constant. However, with an infinite λ, any non-zero curvature would lead to an infinite penalty, so the only solution is a curve wiht zero curvature, which is again $\hat{g}(x) = 0$, a flat line.

(d) $\lambda = \infty, m = 3$

Answer: $g(x)$ would be quartic to minimize the area under the curve of its third derivative. The condition minimizes the third derivative. Since the third derivative of $\hat{g}(x)$ proportional to x^2 is zero, this condition is naturally satisfied. So in this case, $\hat{g}(x)$ could be any multiple of x^2.

(e) $\lambda = \infty, m = 3$

Answer: As in (d), the penalty term no longer matters. This is the formula for linear regression, to choose g based on minimizing RSS.

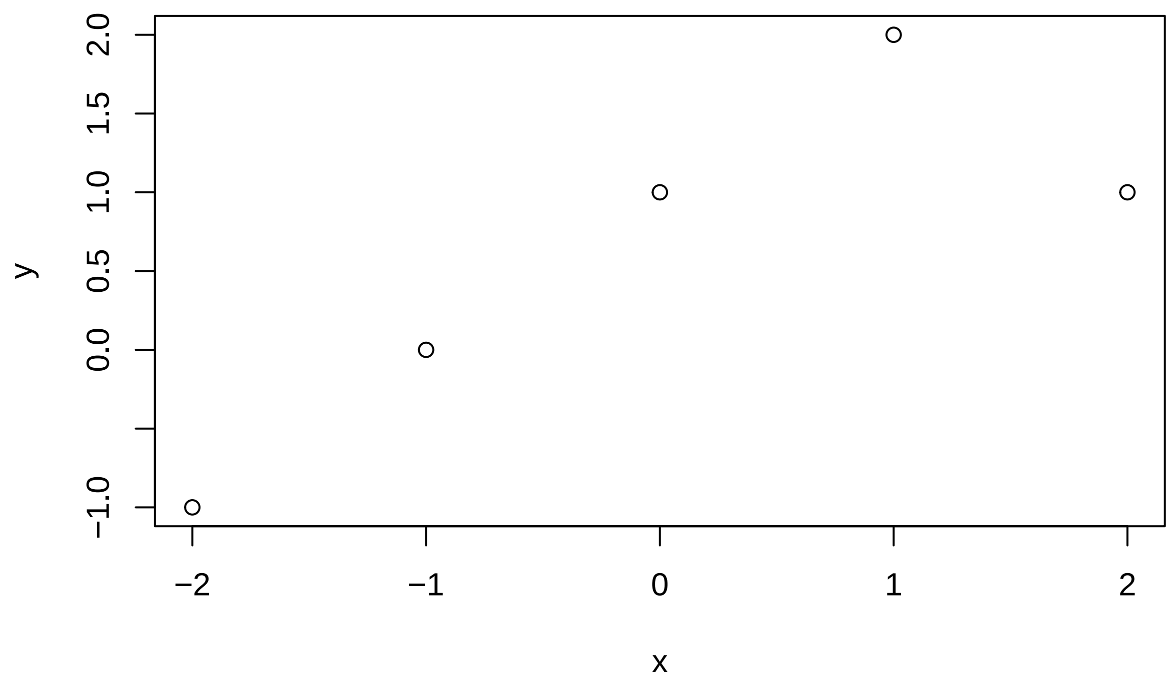
Quesiton 3 [7 marks]

Suppose we fit a curve with basis functions $b_1(X) = X, b_2(X) = (X-1)^2 I(X \geq 1)$. (Note taht $I(X \geq 1)$ equals 1 for $X \geq 1$ and 0 otherwise.) We fit the linear regression model

$$Y = \beta_0 + \beta_1 b_1(X) + \beta_2 b_2(X) + \epsilon$$

and obtain coefficient estimates $\hat{\beta}_0 = 1$, $\hat{\beta}_1 = 1$, $\hat{\beta}_2 = -2$. Sketch the estimated curve between $X = -2$ and $X = 2$. Note the intercepts, slopes, and other relevant information.

```
x = -2:2
beta_0 = 1
beta_1 = 1
beta_2 = -2
y = beta_0 + beta_1*x + beta_2*(x-1)^2*I(x>1)
plot(x, y)
```



Question 9 [18 marks]

This question uses the variables dis (the weighted mean of distances to five Boston employment centers) and nox (nitrogen oxides concentration in parts per 10 million) from the $Boston$ data. We will treat dis as the predictor and nox as the response.

(a) Use the poly() function to fit a cubic polynomial regression to predict nox using dis. Report the regression output, and plot the resulting data and polynomial fits.

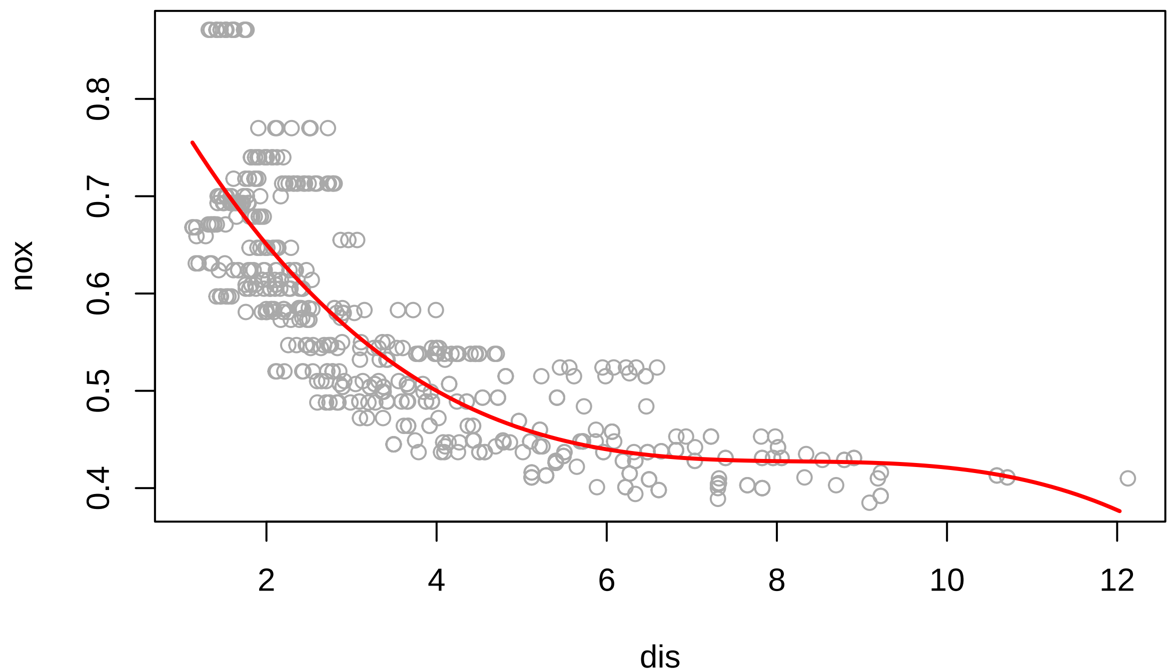
```
#head(Boston)
fit = lm(nox~poly(dis,3), data = Boston)
summary(fit)
```

```
##
## 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
```

From the summary above, we can see that all polynomial term have p-value less than 0.05, which are significant enough for predicting nox.

```r
dislims = range(dis)
dis.grid= seq(from = dislims[1], to = dislims[2], by = 0.1)
preds = predict(fit, list(dis = dis.grid))
plot(nox ~ dis, data = Boston, col = 'darkgrey')
lines(dis.grid, preds, col = "red", lwd = 2)
```



The curve in the plot is smooth which proves it fit the data well.

(b) Plot the polynomial fits for a range of different polynomial degrees (say, from 1 to 10), and report the associated residual sum of squares.

```
rss = rep(NA,10)

for (i in 1:10){
  fit = lm(nox~poly(dis,i), data = Boston)
  rss[i] = round(sum(fit$residuals^2),3)
}
print(rss)
```

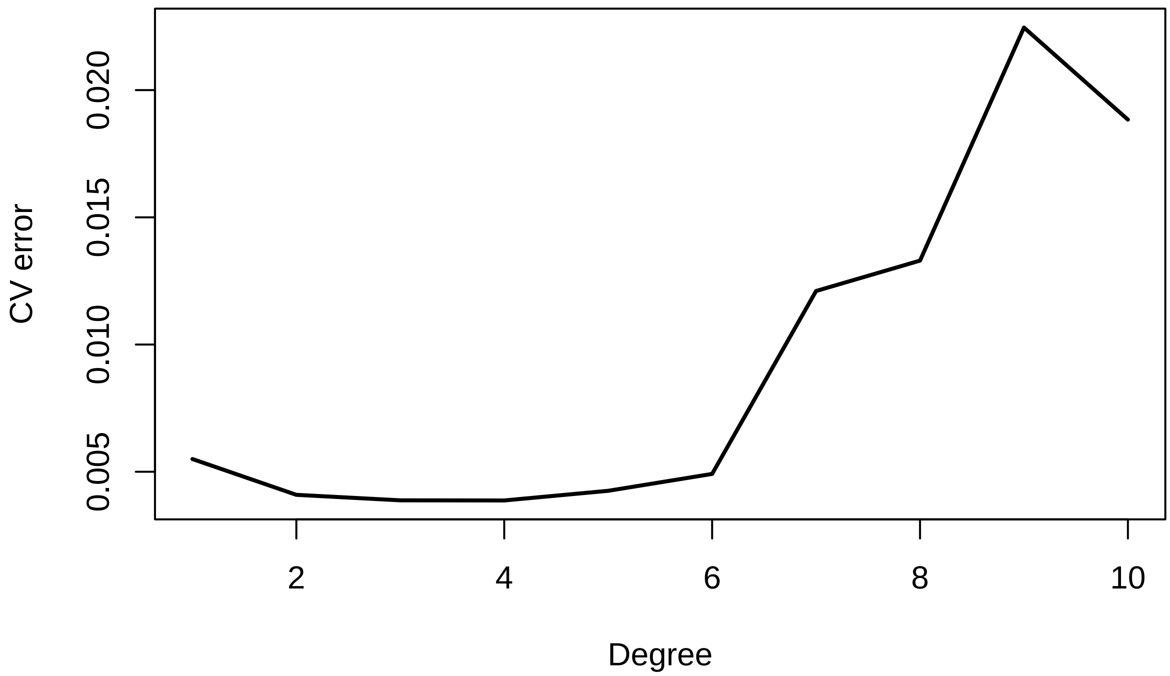
```
##  [1] 2.769 2.035 1.934 1.933 1.915 1.878 1.849 1.836 1.833 1.832
```

RSS decreases with the increase of the degree of polynomial.

(c) Perform cross-validation or another approach to select the optimal degree for the polynomial, and explain your results.

```
library(boot)
cv = rep(NA, 10)

for (i in 1:10){
  fit = glm(nox~poly(dis, i), data = Boston)
  cv[i] = cv.glm(Boston, fit, K=10)$delta[2]
}
plot(1:10, cv, xlab = "Degree", ylab = "CV error", type = "l", pch = 20, lwd = 2)
```



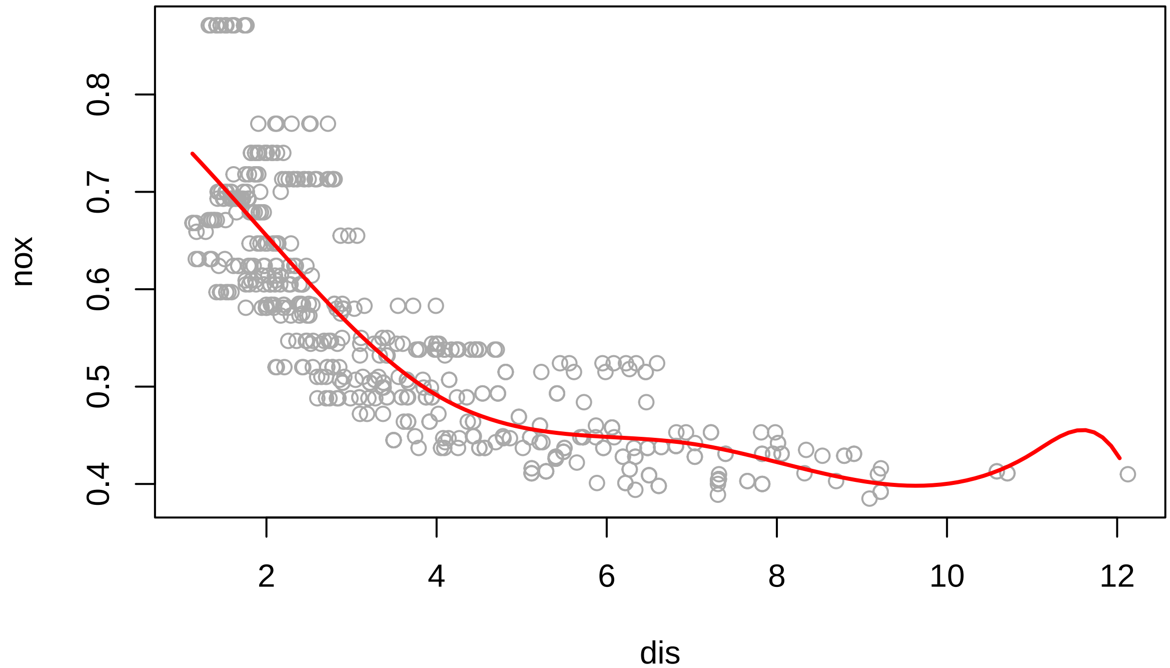
From the plot we can see the CV error decreases at degrees 1 to 5. When it reaches degree of 5, it turns up to an upward trend, which means the cv error increases with the degree increases.

(d) Use the $bs()$ function to fit a regression spline to predict nox using dis. Report the output for the fit using four degrees of freedom. How did you choose the knots? Plot the resulting fit.

```
library(splines)
fit = lm(nox~bs(dis, df = 4, knots = c(4,7,11)), data = Boston)
summary(fit)
```

```
##
## Call:
## lm(formula = nox ~ bs(dis, df = 4, knots = c(4, 7, 11)), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.124567 -0.040355 -0.008702  0.024740  0.192920
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            0.73926    0.01331  55.537  < 2e-16 ***
## bs(dis, df = 4, knots = c(4, 7, 11))1 -0.08861    0.02504  -3.539  0.00044 ***
## bs(dis, df = 4, knots = c(4, 7, 11))2 -0.31341    0.01680 -18.658  < 2e-16 ***
## bs(dis, df = 4, knots = c(4, 7, 11))3 -0.26618    0.03147  -8.459 3.00e-16 ***
## bs(dis, df = 4, knots = c(4, 7, 11))4 -0.39802    0.04647  -8.565  < 2e-16 ***
## bs(dis, df = 4, knots = c(4, 7, 11))5 -0.25681    0.09001  -2.853  0.00451 **
## bs(dis, df = 4, knots = c(4, 7, 11))6 -0.32926    0.06327  -5.204 2.85e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06185 on 499 degrees of freedom
## Multiple R-squared:  0.7185, Adjusted R-squared:  0.7151
## F-statistic: 212.3 on 6 and 499 DF,  p-value: < 2.2e-16
```

```r
dislims = range(dis)
dis.grid= seq(from = dislims[1], to = dislims[2], by = 0.1)
preds = predict(fit, list(dis = dis.grid))
plot(nox ~ dis, data = Boston, col = 'darkgrey')
lines(dis.grid, preds, col = "red", lwd = 2)
```



The summary shows that all terms in spline fit are significant. Plot shows that the spline fits data well except at the extreme values of dis, (especially dis>10).

(e) Now fit a regression spline for a range of degrees of freedom, and plot the resulting fits and report the resulting RSS. Describe the results obtained.

```
rss_new = rep(NA,16)

for (i in 3:16){
  fit = lm(nox~poly(dis,df = i), data = Boston)
  rss_new[i] = round(sum(fit$residuals^2),3)
}
rss_new[-c(1,2)]
```

```
##  [1] 1.934 1.933 1.915 1.878 1.849 1.836 1.833 1.832 1.832 1.830 1.813 1.811
## [13] 1.787 1.785
```

Train RSS monotonically decreases till df=14 and then slightly increases for df=15 and df=16.

(f) Perform cross-validation or another approach in order to select the best degrees of freedom fro a regression spline on this data.

```
cv_new = rep(NA, 16)

for (i in 3:16){
  fit= glm(nox~bs(dis, i), data = Boston)
  cv_new[i] = cv.glm(Boston, fit,K=10)$delta[2]
}
```

```
## Warning in bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.137, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.137, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.1296, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.1296, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`50%` = 3.1174), Boundary.knots =
## c(1.137, : some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`50%` = 3.1174), Boundary.knots =
## c(1.137, : some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`50%` = 3.1827), Boundary.knots =
## c(1.1296, : some 'x' values beyond boundary knots may cause ill-conditioned
## bases

## Warning in bs(dis, degree = 3L, knots = c(`50%` = 3.1827), Boundary.knots =
## c(1.1296, : some 'x' values beyond boundary knots may cause ill-conditioned
## bases

## Warning in bs(dis, degree = 3L, knots = c(`33.33333%` = 2.40296666666667, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
## Warning in bs(dis, degree = 3L, knots = c(`33.33333%` = 2.40296666666667, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`33.33333%` = 2.3817, `66.66667%`
## = 4.23913333333333: some 'x' values beyond boundary knots may cause ill-
## conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`33.33333%` = 2.3817, `66.66667%`
## = 4.23913333333333: some 'x' values beyond boundary knots may cause ill-
## conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`25%` = 2.11175, `50%` = 3.3175, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`25%` = 2.11175, `50%` = 3.3175, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`25%` = 2.10035, `50%` = 3.2157, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`25%` = 2.10035, `50%` = 3.2157, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`20%` = 1.9769, `40%` = 2.6439, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`20%` = 1.9769, `40%` = 2.6439, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`20%` = 1.97024, `40%` = 2.70112, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`20%` = 1.97024, `40%` = 2.70112, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`16.66667%` = 1.82231666666667, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
## Warning in bs(dis, degree = 3L, knots = c(`16.66667%` = 1.82231666666667, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`16.66667%` = 1.84728333333333, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`16.66667%` = 1.84728333333333, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`14.28571%` = 1.80237142857143, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`14.28571%` = 1.80237142857143, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`12.5%` = 1.7353625, `25%` = 2.0608, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`12.5%` = 1.7353625, `25%` = 2.0608, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`12.5%` = 1.754625, `25%` = 2.0941, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`12.5%` = 1.754625, `25%` = 2.0941, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`11.11111%` = 1.66282222222222, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`11.11111%` = 1.66282222222222, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`10%` = 1.62952, `20%` = 1.96376, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`10%` = 1.62952, `20%` = 1.96376, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
## Warning in bs(dis, degree = 3L, knots = c(`9.090909%` = 1.63724545454545, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`9.090909%` = 1.63724545454545, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`9.090909%` = 1.61450909090909, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`9.090909%` = 1.61450909090909, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`8.333333%` = 1.57935, `16.66667%`
## = 1.82023333333333, : some 'x' values beyond boundary knots may cause ill-
## conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`8.333333%` = 1.57935, `16.66667%`
## = 1.82023333333333, : some 'x' values beyond boundary knots may cause ill-
## conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`8.333333%` = 1.5893, `16.66667%`
## = 1.85853333333333, : some 'x' values beyond boundary knots may cause ill-
## conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`8.333333%` = 1.5893, `16.66667%`
## = 1.85853333333333, : some 'x' values beyond boundary knots may cause ill-
## conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`7.692308%` = 1.57254615384615, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`7.692308%` = 1.57254615384615, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`7.692308%` = 1.57254615384615, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`7.692308%` = 1.57254615384615, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
## Warning in bs(dis, degree = 3L, knots = c(`7.142857%` = 1.56255714285714, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`7.142857%` = 1.56255714285714, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`7.142857%` = 1.52855, `14.28571%` =
## 1.7984, : some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`7.142857%` = 1.52855, `14.28571%` =
## 1.7984, : some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```r
plot(3:16, cv_new[-c(1, 2)], lwd = 2, type = "l", xlab = "df", ylab = "CV error")
```