Answers exercises Splines and GAMs - Meeting 9

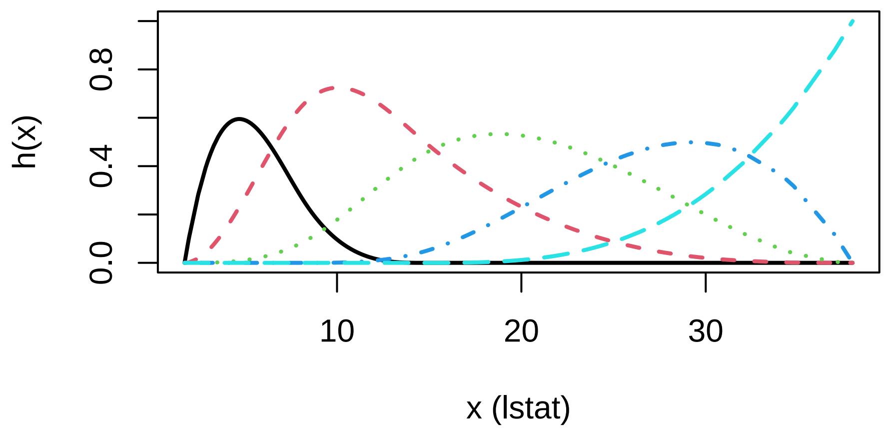
Exercise 1: Fit natural and cubic splines

Load packages and data:

```
library("MASS")
data(Boston)
library("splines")
bs_x <- bs(Boston$lstat, df = 5)
```

a)

```
matplot(Boston$lstat[order(Boston$lstat)], bs_x[order(Boston$lstat), ],
        type = "l", lwd = 2, xlab = "x (lstat)", ylab = "h(x)")
```




```
head(bs_x) ## shows the first 6 rows of the design matrix
```

```
##              1          2            3            4 5
## [1,] 0.5910036 0.26790606 0.0110909274 0.000000e+00 0
## [2,] 0.1599432 0.70905115 0.1309062042 9.948541e-05 0
## [3,] 0.5711464 0.14927001 0.0039309841 0.000000e+00 0
## [4,] 0.4093920 0.04610658 0.0005723661 0.000000e+00 0
## [5,] 0.5766033 0.31509124 0.0150738879 0.000000e+00 0
## [6,] 0.5826565 0.29880037 0.0136161871 0.000000e+00 0
```

```r
attr(bs_x, "knots")
```

```
## [1]  8.316667 14.696667
```

```r
attr(bs_x, "degree")
```

```
## [1] 3
```

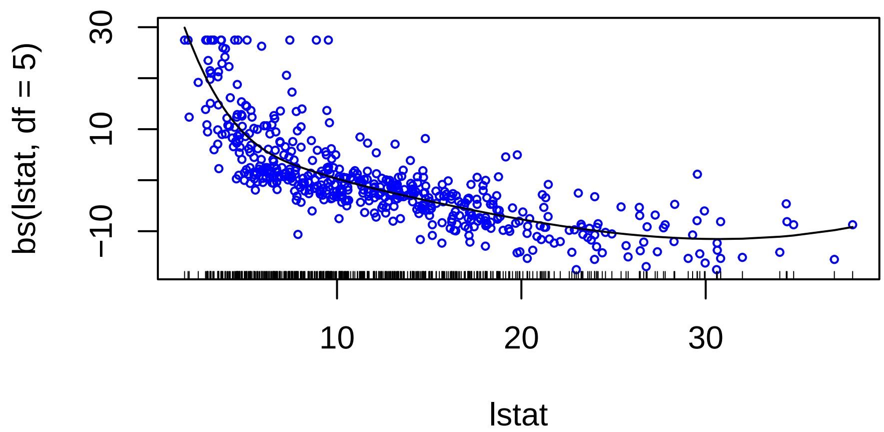
The knots were placed uniformly across the distribution of the predictor. With two knots, they are placed at 1/3 and 2/3 of the observed values. The knots are difficult to distinguish from the plotted basis functions though; B-spline bases are set up as a linear combination of the truncated power basis (as described in the ISLR book and slides) to provide better numerical stability in estimation.

b)

```r
library("gam")
mod_df5 <- gam(medv ~ bs(lstat, df = 5), data = Boston)
summary(mod_df5)
```

```
##
## Call: gam(formula = medv ~ bs(lstat, df = 5), data = Boston)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -15.1774  -3.1790  -0.7981   2.0964  26.6755
##
## (Dispersion Parameter for gaussian family taken to be 27.109)
##
##     Null Deviance: 42716.3 on 505 degrees of freedom
## Residual Deviance: 13554.52 on 500 degrees of freedom
## AIC: 3113.663
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                    Df Sum Sq Mean Sq F value    Pr(>F)
## bs(lstat, df = 5)   5  29162  5832.4  215.14 < 2.2e-16 ***
## Residuals         500  13554    27.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

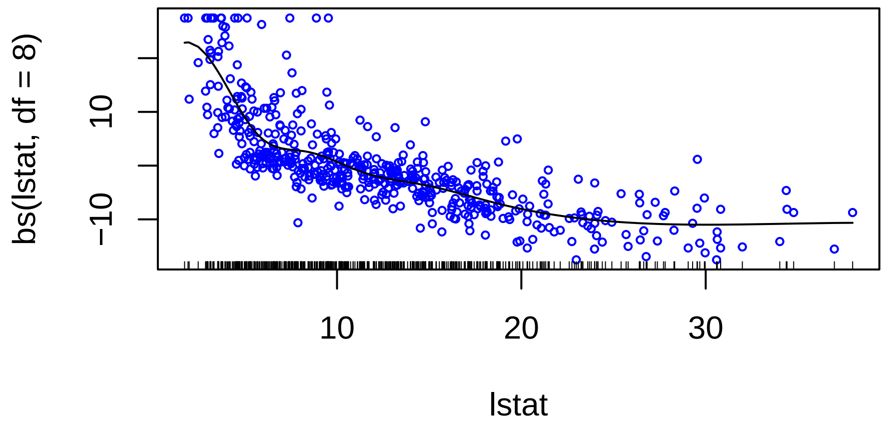
```r
plot(mod_df5, residuals = TRUE, col = "blue", cex = .5)
```




```
mod_df8 <- gam(medv ~ bs(lstat, df = 8), data = Boston)
summary(mod_df8)
```

```
##
## Call: gam(formula = medv ~ bs(lstat, df = 8), data = Boston)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -14.9627  -3.1253  -0.6612   2.0831  26.0972
##
## (Dispersion Parameter for gaussian family taken to be 26.7118)
##
##     Null Deviance: 42716.3 on 505 degrees of freedom
## Residual Deviance: 13275.77 on 497 degrees of freedom
## AIC: 3109.148
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                    Df Sum Sq Mean Sq F value    Pr(>F)
## bs(lstat, df = 8)   8  29441  3680.1  137.77 < 2.2e-16 ***
## Residuals         497  13276    26.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

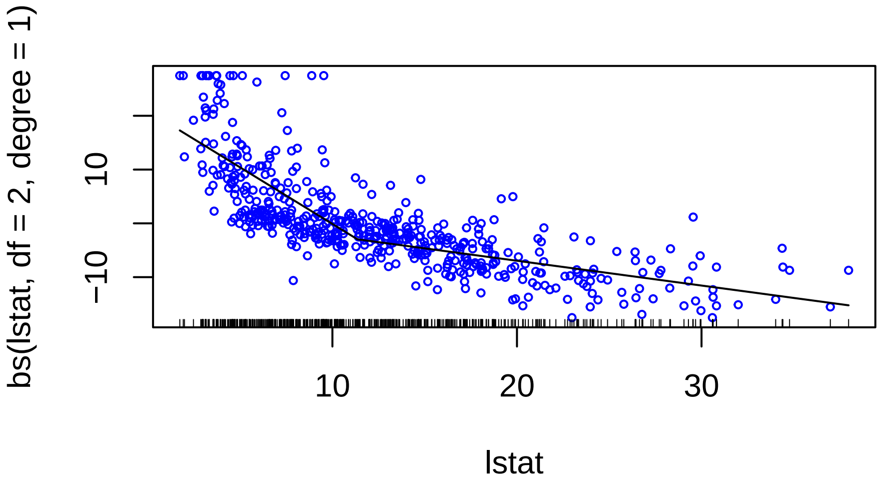
```
plot(mod_df8, residuals = TRUE, col = "blue", cex = .5)
```




```r
mod_df2 <- gam(medv ~ bs(lstat, df = 2, degree = 1), data = Boston)
summary(mod_df2)
```

```
##
## Call: gam(formula = medv ~ bs(lstat, df = 2, degree = 1), data = Boston)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -14.980  -3.677  -0.710   2.495  26.590
##
## (Dispersion Parameter for gaussian family taken to be 30.2618)
##
##     Null Deviance: 42716.3 on 505 degrees of freedom
## Residual Deviance: 15221.69 on 503 degrees of freedom
## AIC: 3166.36
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                                 Df Sum Sq Mean Sq F value    Pr(>F)
## bs(lstat, df = 2, degree = 1)   2  27495 13747.3  454.28 < 2.2e-16 ***
## Residuals                     503  15222    30.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(mod_df2, residuals = TRUE, col = "blue", cex = .5)
```




```
BIC(mod_df2)
```

```
## [1] 3183.266
```

```
BIC(mod_df5)
```

```
## [1] 3143.249
```

```
gam(medv ~ lstat + rm + ptratio + crim + dis, data = Boston)
```

```
## Call:
## gam(formula = medv ~ lstat + rm + ptratio + crim + dis, data = Boston)
##
## Degrees of Freedom: 505 total; 500 Residual
## Residual Deviance: 12995.37
```

c)

The 5 df cubic spline fits best according to BIC, the plots suggest similar: 8 df yields a slightly too wiggly function, while the 2 df linear spline seems too inflexible. Note that models with different (number and/or location of) knots are not nested, so although we are still in the parametric realm and we might want to test significance of the difference in fit, we should not.

d)

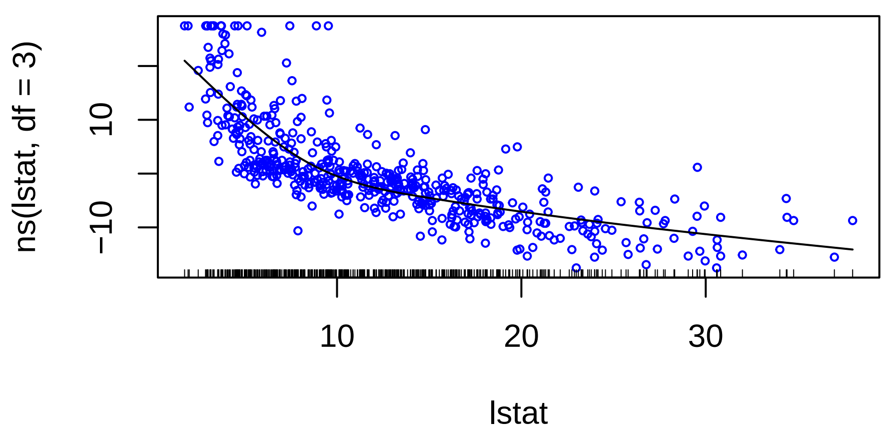
```
ns3 <- ns(Boston$lstat, df = 3)
attr(ns3, "knots")
```

```
## [1]  8.316667 14.696667
```

```r
mod_ns3 <- gam(medv ~ ns(lstat, df = 3), data = Boston)
summary(mod_ns3)
```

```
##
## Call: gam(formula = medv ~ ns(lstat, df = 3), data = Boston)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7595  -3.3628  -0.6468   2.3062  27.2857
##
## (Dispersion Parameter for gaussian family taken to be 28.4261)
##
##     Null Deviance: 42716.3 on 505 degrees of freedom
## Residual Deviance: 14269.9 on 502 degrees of freedom
## AIC: 3135.688
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                    Df Sum Sq Mean Sq F value    Pr(>F)
## ns(lstat, df = 3)   3  28446  9482.1  333.57 < 2.2e-16 ***
## Residuals         502  14270    28.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(mod_ns3, residuals = TRUE, col = "blue", cex = .5)
```



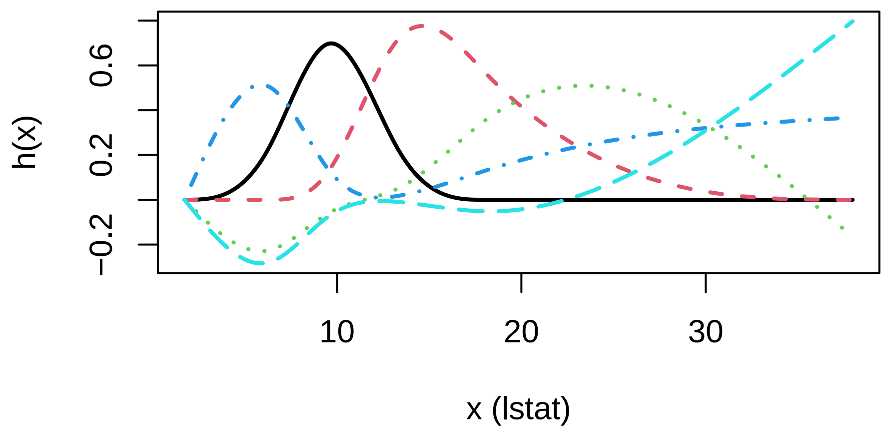
The number and location of the knots in the 3 df natural spline are identical to those of the 5 df cubic spline. By using a natural spline, we thus free up 2 df.

e)

```r
ns5_x <- ns(Boston$lstat, df = 5)
attr(ns5_x, "knots")
```

```
## [1]  6.29  9.53 13.33 18.06
```

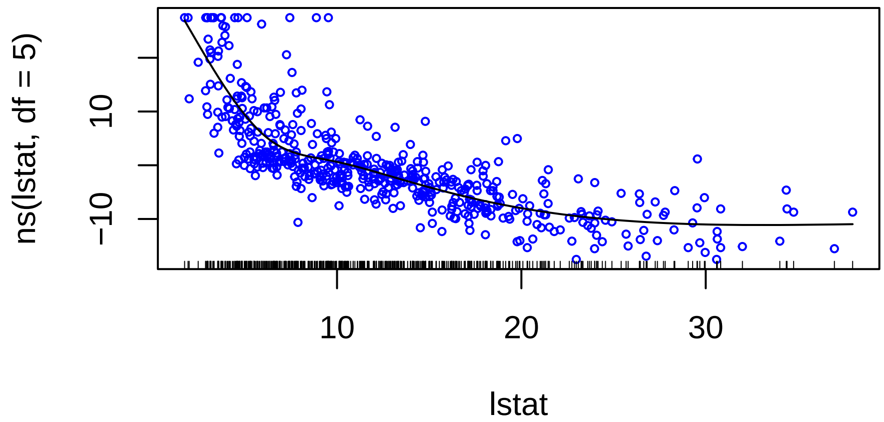
```r
matplot(Boston$lstat[order(Boston$lstat)], ns5_x[order(Boston$lstat), ],
        type = "l", lwd = 2, xlab = "x (lstat)", ylab = "h(x)")
```




```r
mod_ns5 <- gam(medv ~ ns(lstat, df = 5), data = Boston)
summary(mod_ns5)
```

```
##
## Call: gam(formula = medv ~ ns(lstat, df = 5), data = Boston)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -13.9811  -3.0266  -0.7252   2.1416  26.5111
##
## (Dispersion Parameter for gaussian family taken to be 26.9021)
##
##     Null Deviance: 42716.3 on 505 degrees of freedom
## Residual Deviance: 13451.03 on 500 degrees of freedom
## AIC: 3109.785
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                    Df Sum Sq Mean Sq F value    Pr(>F)
## ns(lstat, df = 5)   5  29265  5853.1  217.57 < 2.2e-16 ***
## Residuals         500  13451    26.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(mod_ns5, residuals = TRUE, col = "blue", cex = .5)
```




```r
attr(bs_x, "knots")
```

```
## [1]  8.316667 14.696667
```

```r
attr(bs_x, "degree")
```

```
## [1] 3
```

```r
BIC(mod_ns3)
```

```
## [1] 3156.82
```

```r
BIC(mod_ns5)
```

```
## [1] 3139.37
```

The residual deviance is lower for the natural spline than for the cubic spline with same df. According to the BIC, the natural spline with 5 df fits best; it is also better than the earlier cubic spline with 5 df. The plots suggest similar. The natural spline indeed seems an improvement over the cubic spline.

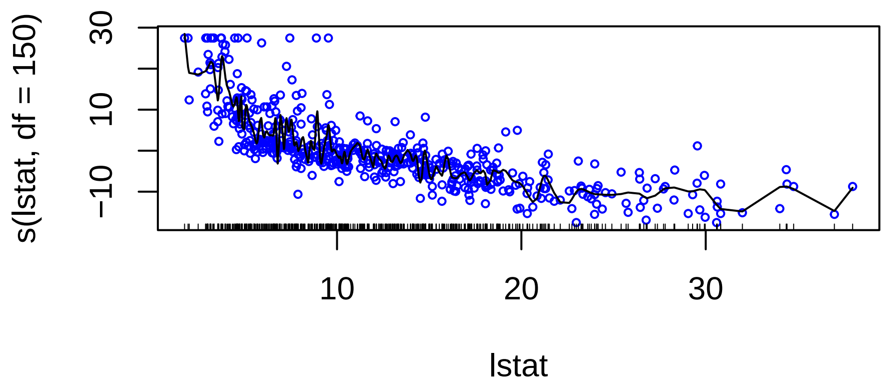
Exercise 2: Fit a smoothing spline

```r
mod_sc <- gam(medv ~ s(lstat, df = 150), data = Boston) # complex fit
summary(mod_sc)
```

High df (low value of λ)

```
##
## Call: gam(formula = medv ~ s(lstat, df = 150), data = Boston)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -14.0712  -2.7340  -0.4702   2.0649  21.9654
##
## (Dispersion Parameter for gaussian family taken to be 27.6915)
##
##     Null Deviance: 42716.3 on 505 degrees of freedom
## Residual Deviance: 10772.01 on 389 degrees of freedom
## AIC: 3219.4
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                     Df Sum Sq Mean Sq F value    Pr(>F)
## s(lstat, df = 150)   1  23244 23243.9  839.39 < 2.2e-16 ***
## Residuals          389  10772    27.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                    Npar Df Npar F     Pr(F)
## (Intercept)
## s(lstat, df = 150)     115 2.7321 2.103e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(mod_sc, residuals = TRUE, cex = .5, col = "blue")
```



The results present both a parametric and non-parametric effect of `lstat`:

- The parametric effect represents the linear slope, which uses up only 1 df. The non-parametric effects represent the non-linear effects.
- The non-linear part of the smoothing spline for `lstat` took up 115 degrees of freedom. This is less than the requested degrees of freedom we specified, because by default the knots are placed at a subset of the observations, for computational considerations. In addition, >115 knots would rarely be needed to approximate a curve. As the plot above already indicates, this would yield a very wiggly curve. In fact, 10 df would in most cases more than suffice to flexibly approximate shapes in most data problems.

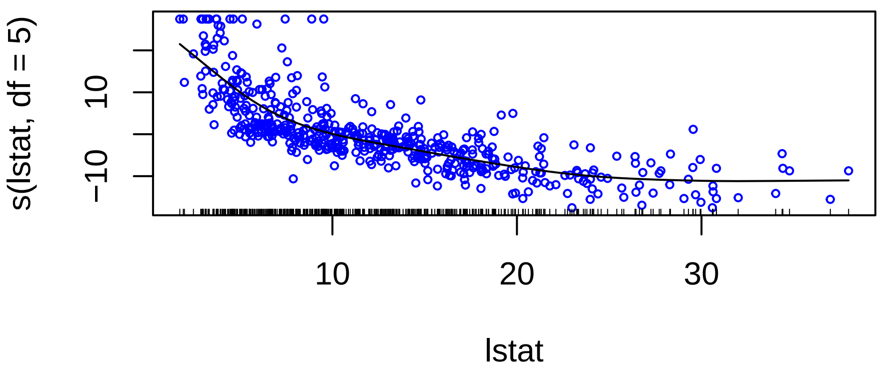
```r
mod_ss <- gam(medv ~ s(lstat, df = 5), data = Boston) # more simple fit
summary(mod_ss)
```

Low df (high value of λ)

```
##
## Call: gam(formula = medv ~ s(lstat, df = 5), data = Boston)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -13.6332  -3.2159  -0.6577   2.2051  26.8386
##
## (Dispersion Parameter for gaussian family taken to be 27.6492)
##
##     Null Deviance: 42716.3 on 505 degrees of freedom
## Residual Deviance: 13824.6 on 499.9999 degrees of freedom
## AIC: 3123.646
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                   Df Sum Sq Mean Sq F value    Pr(>F)
## s(lstat, df = 5)   1  23244 23243.9  840.67 < 2.2e-16 ***
```

```
## Residuals        500  13825    27.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                  Npar Df Npar F     Pr(F)
## (Intercept)
## s(lstat, df = 5)       4 51.065 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(mod_ss, residuals = TRUE, col = "blue", cex = .5)
```



With 5 df, the flexibility is much lower, and we obtain a much smoother fit.

```r
BIC(mod_ss)
```

```
## [1] 3136.326
```

```r
BIC(mod_sc)
```

```
## [1] 3232.079
```

```r
BIC(mod_ns5)
```

```
## [1] 3139.37
```

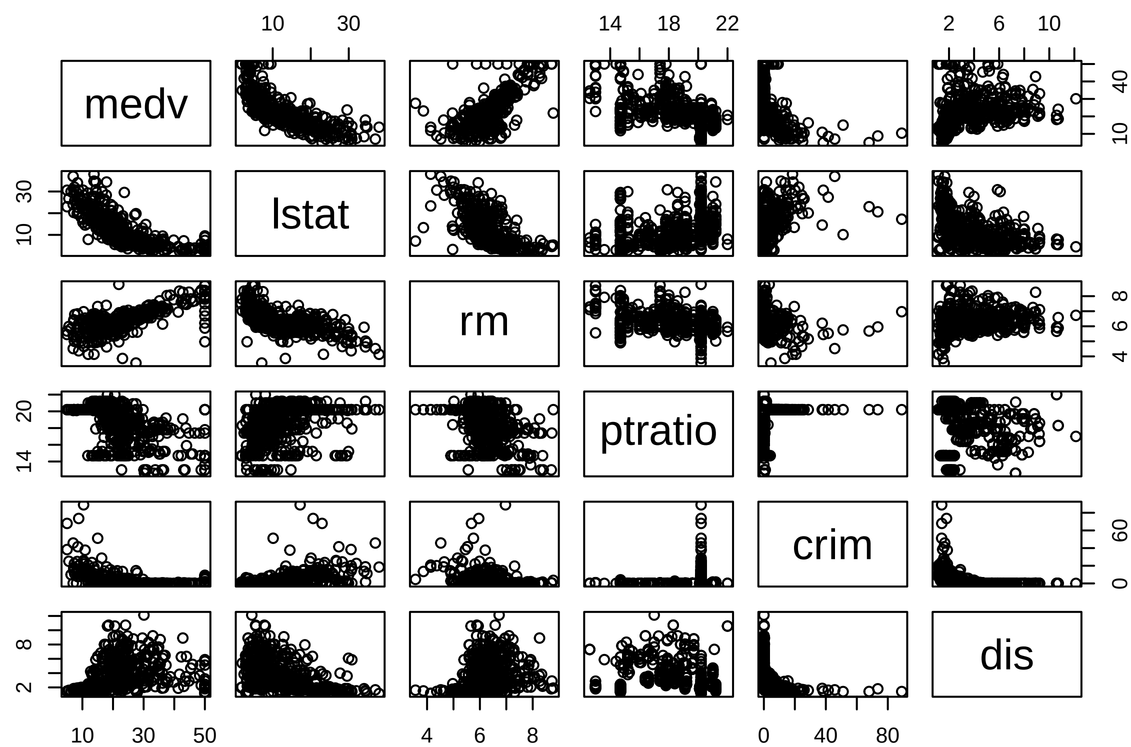
According to the BIC, the more heavily penalized smoothing spline (i.e., with df = 5) has better fit than the smoothing spline with higher df / lower penalty. This is in accordance with what we can conclude from visual inspection of the fitted smoothing splines. The smoothing spline with 5 df also outperforms the natural spline with 5 df from the previous exercise. Thus, the non-parametric smoothing spline approach appears to improve on the parametric spline approaches.

Exercise 3: Fit a GAM (multiple predictor variables)

```
detach("package:gam", unload = TRUE)
```

Before we fit the model, we first inspect variable distributions:

```
vars <- c("medv", "lstat", "rm", "ptratio", "crim", "dis")
plot(Boston[ , vars])
```



Most variable pairs show some dependence. Most variables have somewhat skewed distributions. Crime rate (`crim`) is most heavily skewed.

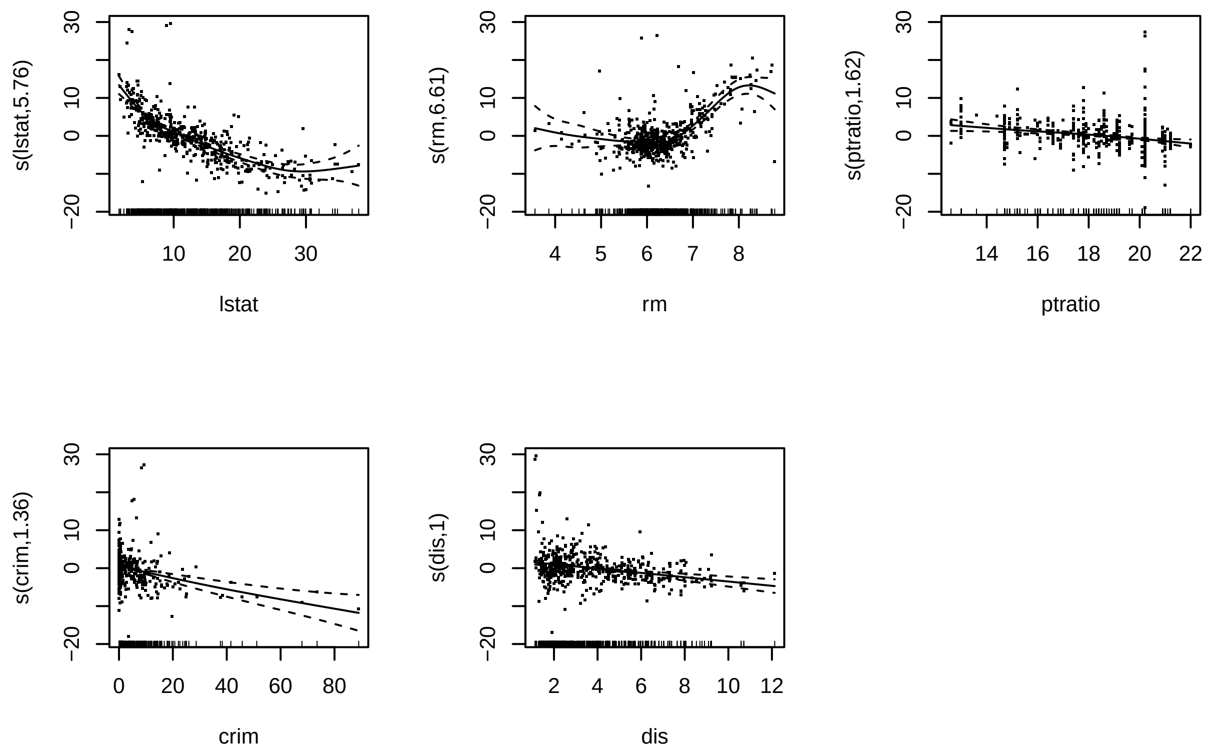
We fit a model with smoothing splines for the five predictors:

```
library("mgcv")
boston_GAM <- gam(medv ~ s(lstat) + s(rm) + s(ptratio) + s(crim) + s(dis),
                  data = Boston, method = "REML")
```

We used restricted maximum likelihood estimation to fit the model (`method = "REML"`). Note that we did not specify the degrees of freedom, like with the `s` function of package **gam**. Package **mgcv** performs automatic smoothness selection: It estimates the optimal value of the smoothing parameter directly, using (restricted) maximum likelihood.

This is basically amounts to estimating a mixed-effects model: The linear effects are treated as unpenalized, fixed-effects terms. The non-linear effects are treated as random effects, which are shrunken towards zero using a quadratic penalty.

```r
par(mfrow = c(2, 3))
plot(boston_GAM, residuals = TRUE)
```



The plots show the conditional effects of the predictors. The effects of `lstat` and `rm` are obviously non-linear. Also, the plots show 95% confidence intervals for the pointwise effects.

```r
summary(boston_GAM)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## medv ~ s(lstat) + s(rm) + s(ptratio) + s(crim) + s(dis)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.5328     0.1763   127.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df     F  p-value
## s(lstat)   5.758  6.945 46.39  < 2e-16 ***
## s(rm)      6.614  7.762 25.21  < 2e-16 ***
```

```
## s(ptratio) 1.620  2.019 12.98 3.26e-06 ***
## s(crim)    1.364  1.637 24.16 2.77e-07 ***
## s(dis)     1.002  1.004 27.99 6.46e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.814   Deviance explained =   82%
## -REML = 1431.1  Scale est. = 15.724    n = 506
```

Package `mgcv` does not distinguish between the parametric and non-parametric part of the smoothing spline.

a) The `lstat` (percent of lower status population) variable seems to have the strongest effect: It has the highest F-value and its effect takes up most of the range on the y-axis of the plots. The F-values are computed from both the variation in $\hat{f}_j$ (the fitted spline curve), as well as its precision, so are not direct measures of the magnitude of a variable's effects. The plots indicate the next most important variable is `rm`, the number of rooms.

b) The `edf` values indicate that `dis`, `crim` and `ptratio` have close to linear effects, while `lstat` and `rm` have clearly non-linear effects. The `edf` values (empirical degrees of freedom) quantify the amount of non-linearity. With a value of 1, the spline takes up only 1 df, reflecting only a linear slope. With df > 1, non-linear effects were captured by the smoothing spline.

c) The p-values test the null hypothesis of no effect. The results indicate that all predictors significantly contribute to predicting median values of owner-occupied homes.

Exercise 4: Prove continuity of a cubic spline at the knot

a) If $X \leq \xi$ then we have $f_1(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3$.

   Thus, $a_1 = \beta_0$, $b_1 = \beta_1$, $c_1 = \beta_2$ and $d_1 = \beta_3$.

b) If $X > \xi$ then we have $f_2(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 (X - \xi)^3$.

   Further, $(X - \xi)^3_+ = X^3 - 3\xi X^2 + 3\xi^2 X - \xi^3$.

   Thus, $f_2(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 (X^3 - 3\xi X^2 + 3\xi^2 X - \xi^3)$.

   Thus, $a_2 = \beta_0 - \beta_4 \xi^3$, $b_2 = \beta_1 + 3\beta_4 \xi^2$, $c_2 = \beta_2 - 3\beta_4 \xi$ and $d_2 = \beta_3 + \beta_4$.

c) If $X = \xi$ then $(X - \xi)^3_+ = 0$. Thus, $f_2(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 = f_1(X)$ for $X = \xi$, and we've shown that $f(X)$ is continuous at $X = \xi$.

d) $f_1'(X) = \beta_1 + 2\beta_2 X + 3\beta_3 X^2$, thus $f_1'(\xi) = \beta_1 + 2\beta_2 \xi + 3\beta_3 \xi^2$.

   $f_2'(X) = b_2 + 2c_2 X + 3d_2 X^2 = \beta_1 + 3\beta_4 \xi^2 + 2(\beta_2 - 3\beta_4 \xi) X + 3(\beta_3 + \beta_4) x^2$.

   $f_2'(\xi) = \beta_1 + 3\beta_4 \xi^2 + 2\beta_2 \xi - 6\beta_4 \xi^2 + 3\beta_3 \xi^2 + 3\beta_4 \xi^2$.

   $f_2'(\xi) = \beta_1 + 2\beta_2 \xi + 3\beta_3 \xi^2$.

   Thus $f_1'(\xi) = f_2'(\xi)$. $f'(X)$ is continuous at $X = \xi$.

e) $f_1''(X) = 2\beta_2 + 6\beta_3 X$, thus $f_1'(\xi) = 2\beta_2 + 3\beta_3\xi$.

$f_2''(X) = 2(\beta_2 - 3\beta_4\xi) + 6(\beta_3 + \beta_4)x$.

$f_2''(\xi) = 2\beta_2 - 6\beta_4\xi + 6\beta_3\xi + 6\beta_4\xi = 2\beta_2 + 6\beta_3\xi$.

Thus $f_1''(\xi) = f_2''(\xi)$. $f''(X)$ is continuous at $X = \xi$.