REPORT

수강과목	:	빅데이터통계분석
담당교수	:	선호근
학 과	:	통계학과
학 번	:	201611531
이 름	:	정호재
제출일자	:	2020.09.17.

Homework Assignment 01

The Due Date : By Thursday, September, 17th in class
Your solution should include R codes and the answer of each question.
You need to upload your R codes on http://plato.pusan.ac.kr for full credits.
You may collaborate on this problem but you must write up your own solution.

변수설명은 다음과 같고 변수들 중 target값은 crim이다.
crim : per capita crime rate by town.
zn : proportion of residential land zoned for lots over 25,000 sq.ft.
indus : proportion of non-retail business acres per town.
chas: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).
nox : nitrogen oxides concentration (parts per 10 million).
rm : average number of rooms per dwelling.
age : proportion of owner-occupied units built prior to 1940.
dis : weighted mean of distances to five Boston employment centres.
rad : index of accessibility to radial highways.
tax : full-value property-tax rate per $10,000.
ptratio : pupil-teacher ratio by town.
black : 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town.
lstat : lower status of the population (percent).
medv : median value of owner-occupied homes in $1000s.

또한 55번 관측치와 chas, dis, rad 세 변수를 제외한 변수들만 feature로 두고 분석을 진행한다.

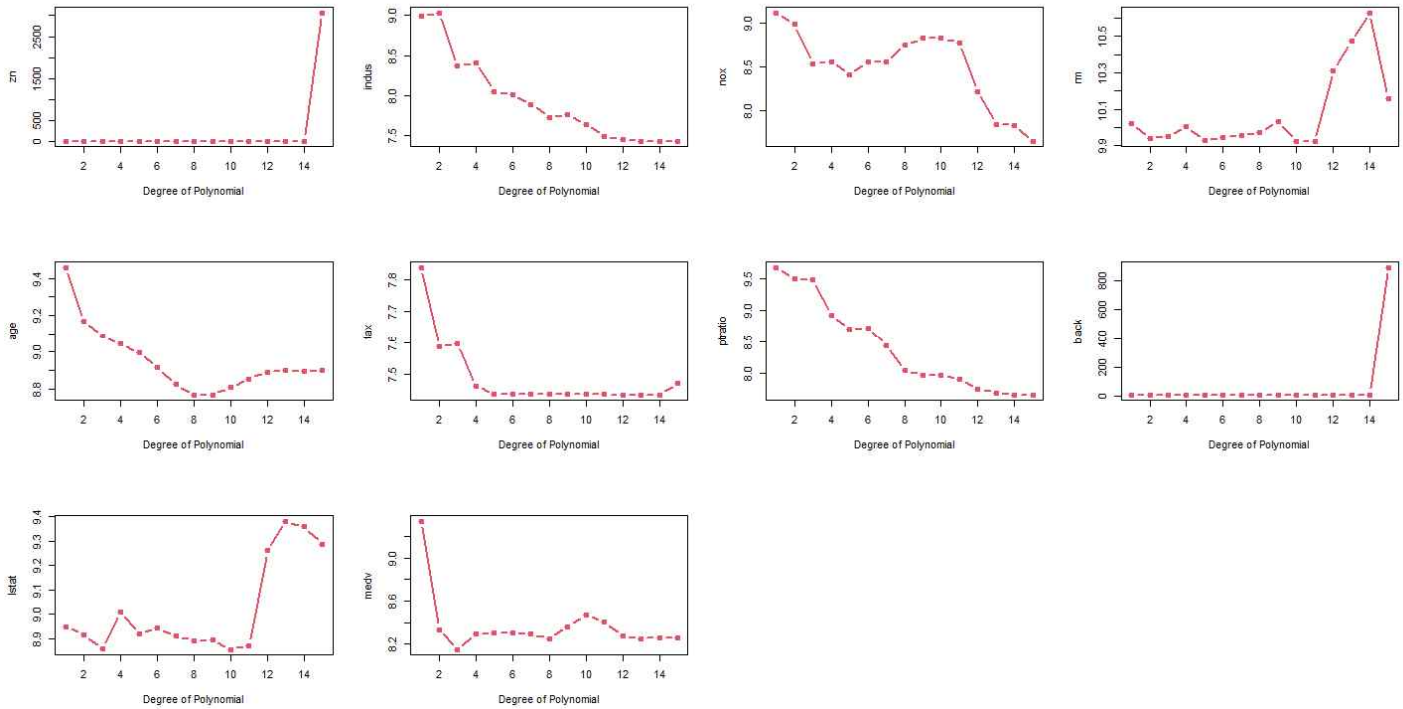
1. For each predictor, apply a polynomial regression model with the kth degree, where $k = 1, 2, \ldots, 15$. You should use the `R` function `poly(..., k)`. Since you have a total of 10 predictors, you can have 10 polynomial regression models where each model has 15 different degrees of the polynomial. For each model, find the optimal value of k that can minimize the prediction error (PE) of the test set. The PE is defined as

$$\mathrm{PE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{f}(x_i)\right)^2},$$

where $\hat{f}(\cdot)$ is a fitted polynomial regression using the training set and $m = 55$. For each model (predictor), provide the optimal value of k and the numerical value of the corresponding PE.

다음 표는 각 변수에 대한 degree별 PE값을 나타낸 데이터이다.
poly 모델에 대한 각 변수별 degree별 PE값을 비교 해봤을 때 degree가 커짐에 따라 극명하게 PE값이 줄고 있으나 몇몇 그렇지 않은 값도 존재한다. 조금 더 쉽게 보기 위하여 그래프를 그려 확인해 적당한 degree를 가진 최종모델을 찾고자 한다.

	zn	indus	nox	rm	age	tax	ptratio	black	lstat	medv
1	10.043964	8.985327	9.123556	10.021944	9.458352	7.839268	9.679275	8.745484	8.949729	9.345050
2	9.918015	9.032868	8.992473	9.942636	9.168938	7.586897	9.505326	8.726587	8.916178	8.334238
3	9.897628	8.365914	8.535951	9.951910	9.086060	7.597863	9.487394	8.760252	8.858170	8.147778
4	9.891760	8.412605	8.561998	10.006793	9.044814	7.463775	8.926924	8.733982	9.009155	8.295878
5	9.890364	8.051734	8.414579	9.928414	8.994212	7.437374	8.703354	8.811375	8.919893	8.299175
6	9.889951	8.010690	8.554915	9.946743	8.914949	7.436708	8.730197	8.879368	8.944036	8.303398
7	9.889651	7.888519	8.556116	9.954324	8.823420	7.437382	8.446871	8.945005	8.910268	8.292054
8	9.889532	7.731113	8.756390	9.971790	8.768530	7.437713	8.056924	9.032873	8.892185	8.247531
9	9.889458	7.757107	8.828382	10.033071	8.769231	7.437019	7.974110	9.420726	8.893296	8.361435
10	9.889443	7.640473	8.834832	9.925192	8.808293	7.437372	7.978791	9.421128	8.855429	8.468070
11	9.889435	7.493358	8.778024	9.925218	8.854541	7.436876	7.915575	9.441039	8.871934	8.405831
12	9.889439	7.449662	8.218143	10.313651	8.889617	7.436039	7.761962	9.403179	9.260601	8.276843
13	9.889447	7.433759	7.840010	10.472887	8.900967	7.435990	7.697606	9.382977	9.381958	8.250126
14	9.890180	7.431920	7.834333	10.630234	8.895956	7.436088	7.668816	9.373075	9.358028	8.257137
15	3065.571960	7.431768	7.649432	10.160331	8.900669	7.471207	7.671441	893.969269	9.287547	8.258293



가장 간단한 모델이면서 최적의 PE값을 선택하고자한다.

zn 변수는 degree가 1부터 14까지 큰 변화가 없으므로 가장 간단한 모델인 degree가 1인 모델을 선택한다.
indus 변수는 degree가 11까지 점차 감소하고 이후로는 큰 변화가 없어 보여 degree가 11인 모델을 선택한다.
nox 변수는 degree가 13에서 크게 감소하였고 이후는 모델이 너무 복잡하여 degree가 13인 모델을 택한다.
rm 변수는 1부터 11까지의 PE 값이 비슷하나 조금 더 낮은 PE값을 위해 1이 아닌 degree 2인 모델을 택한다.
age 변수는 최솟값인 degree 8인 모델까지 점차 감소하고 그 이후는 약간 증가한다. degree 8인 모델을 택한다.
tax는 degree 4 이후로 PE 감소폭이 줄어들어 간단한 모델인 degree 4 모델을 선택한다.
ptratio 변수는 degree 14까지 점점 감소한다 이때 8에서의 감소폭이 가장 컸고 간단한 모델을 위해 degree 8인 모델을 선택한다.
black 변수는 degree 1과 최솟값인 2인 모델의 PE값에서 큰 차이가 없으므로 degree 1인 모델을 선택한다.
lstat 변수는 degree 3일 때의 PE값이 최솟값인 degree가 10일 때의 PE값과 큰 차이가 없으므로 간단한 모델인 degree 3인 모델을 선택한다.
medv 변수는 degree 3에서 간단한 모델이면서 최솟값을 가지므로 이 때의 모델을 선택한다.

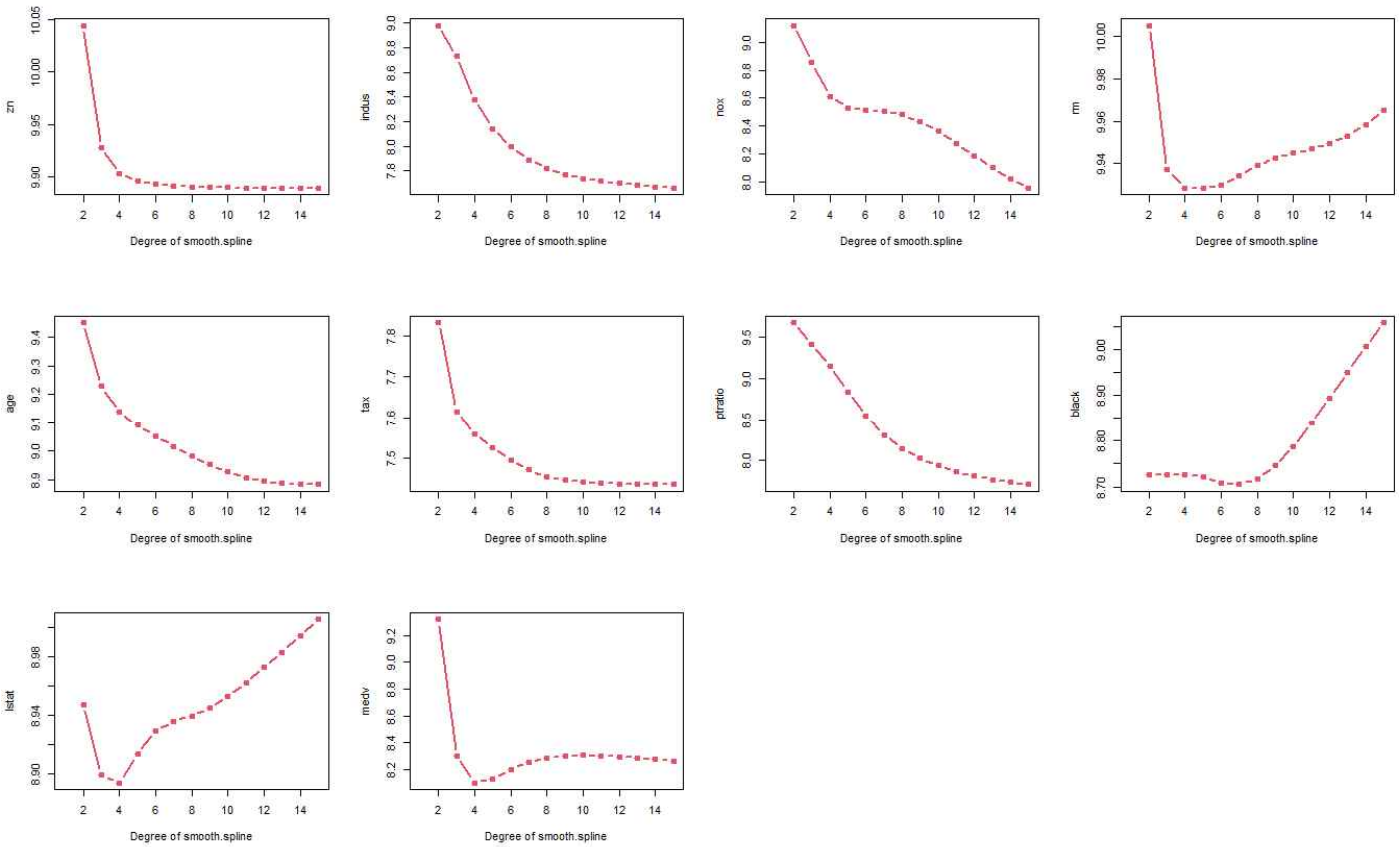
선택한 차수별 PE와 최솟값 일 때의 차수를 비교해보면 다음과 같다.

	df	model_poly_min		poly_df_pe
zn	11	9.889435	1	10.043964
indus	15	7.431768	11	7.493358
nox	15	7.649432	13	7.840010
rm	10	9.925192	2	9.942636
age	8	8.768530	8	8.768530
tax	13	7.435990	4	7.463775
ptratio	14	7.668816	8	8.056924
black	2	8.726587	1	8.745484
lstat	10	8.855429	3	8.858170
medv	3	8.147778	3	8.147778

2. For each predictor, apply smoothing spline with the degree of freedoms k from 2 to 15, i.e., $k = 2, 3, \ldots, 15$. You can use the `R` function `smooth.spline(..., df= )`, which has been discussed in class. Find the optimal value of k that can minimize the prediction error (PE) of the test set. For each model (predictor), provide the optimal value of k and the numerical value of the corresponding PE.

다음 표는 각 변수에 대한 degree별 PE값을 나타낸 데이터이다. smooth.spline 모델은 degree가 1일 때는 성립하지 않는다. 이 모델들에 대한 각 변수별 degree별 PE값을 비교해 봤을 때 degree가 커짐에 따라 극명하게 PE값이 줄고 있으나 몇몇 그렇지 않은 값도 존재한다. 조금 더 쉽게 보기 위하여 그래프를 그려 확인해 적당한 degree를 가진 최종모델을 찾고자 한다.

	zn	indus	nox	rm	age	tax	ptratio	black	lstat	medv
1	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
2	10.043922	8.981238	9.121631	10.005262	9.450754	7.832294	9.679160	8.726939	8.947109	9.317741
3	9.928101	8.729187	8.857618	9.937074	9.228163	7.614913	9.415599	8.726941	8.899067	8.300601
4	9.903471	8.379307	8.613213	9.928371	9.137265	7.559168	9.153627	8.726942	8.893568	8.103097
5	9.896256	8.144290	8.529093	9.928350	9.090248	7.526054	8.838948	8.721217	8.913226	8.133811
6	9.893220	7.995117	8.511544	9.930034	9.052993	7.495026	8.540805	8.708857	8.929197	8.203357
7	9.891692	7.889556	8.503062	9.934280	9.017322	7.470708	8.309800	8.705181	8.935627	8.256055
8	9.890831	7.816802	8.480177	9.939100	8.983098	7.454947	8.144737	8.717082	8.939339	8.287239
9	9.890311	7.768389	8.431574	9.942684	8.952335	7.446155	8.024502	8.745765	8.944815	8.302555
10	9.889986	7.736278	8.359961	9.944998	8.927022	7.441480	7.934056	8.787880	8.952685	8.307238
11	9.889782	7.713956	8.274009	9.946961	8.908003	7.438937	7.863655	8.838647	8.962177	8.304698
12	9.889652	7.697123	8.184797	9.949486	8.895086	7.437475	7.809007	8.894034	8.972431	8.297471
13	9.889570	7.683181	8.099080	9.953185	8.887562	7.436585	7.766728	8.950713	8.983021	8.287559
14	9.889516	7.670512	8.020917	9.958410	8.884226	7.436011	7.733827	9.005867	8.993994	8.276635
15	9.889480	7.658422	7.952312	9.965345	8.883978	7.435618	7.708716	9.058144	9.005384	8.265808



zn 변수는 degree가 4부터 15까지 큰 변화가 없으므로 가장 간단한 모델인 degree가 4인 모델을 선택한다.
indus 변수는 degree가 8까지 점차 감소한다. 이후로는 큰 변화가 없어 보여 degree가 8인 모델을 선택한다.
nox 변수는 degree가 15까지 계속 감소하며 PE 값의 차이가 크므로 가장 복잡한 모델이지만 낮은 PE값을 위하여 degree가 15인 모델을 택한다.
rm 변수는 degree 4에서 최솟값을 가지고 3의 모델과는 격차가 있어 degree 4인 모델을 택한다.
age 변수는 최솟값인 degree 12인 모델까지 점차 감소하고 그 이후는 큰 변화가 없다. degree 12인 모델을 택한다.
tax는 degree 8 이후로 PE 감소폭이 줄어들어 간단한 모델인 degree 8 모델을 선택한다.
ptratio 변수는 degree 12까지 점점 감소한다. 이후에는 큰 변화가 없으므로 degree 12인 모델을 선택한다.
black 변수는 degree 7에서 최솟값을 가지지만 2인 모델의 PE값에서 큰 차이가 없으므로 degree 2인 모델을 선택한다.
lstat 변수는 degree 3일 때의 PE값이 최솟값인 degree가 4일 때의 PE값과 큰 차이가 없으므로 간단한 모델인 degree 3인 모델을 선택한다.
medv 변수는 degree 4에서 간단한 모델이면서 최솟값을 가지므로 이 때의 모델을 선택한다.

선택한 차수별 PE와 최솟값 일 때의 차수를 비교해보면 다음과 같다.

	df	model_spline_min		spline_df_pe
zn	15	9.889480	4	9.903471
indus	15	7.658422	8	7.816802
nox	15	7.952312	15	7.952312
rm	5	9.928350	4	9.928371
age	15	8.883978	12	8.895086
tax	15	7.435618	8	7.454947
ptratio	15	7.708716	12	7.809007
black	7	8.705181	2	8.726939
lstat	4	8.893568	3	8.899067
medv	4	8.103097	4	8.103097

3. Next, we apply a step function to each of 10 predictors, where we break the range of `x` into bins and fit a different constant in each bin. The step function basically converts a continuous variable into an ordered categorical variable. For example, we construct $k+1$ new variables for the j-th predictor X_j such that

$$C_0(X_j) = I(X_j < c_1), C_1(X_j) = I(c_1 \leq X_j < c_2), C_2(X_j) = I(c_2 \leq X_j < c_3), \ldots, C_k(X_j) = I(X_j \geq c_k),$$

where $c_1, c_2, \ldots, c_k$ are k cutpoints and $I(\cdot)$ is an indicator function. Note that

$$C_0(X_j) + C_1(X_j) + \cdots + C_k(X_j) = 1,$$

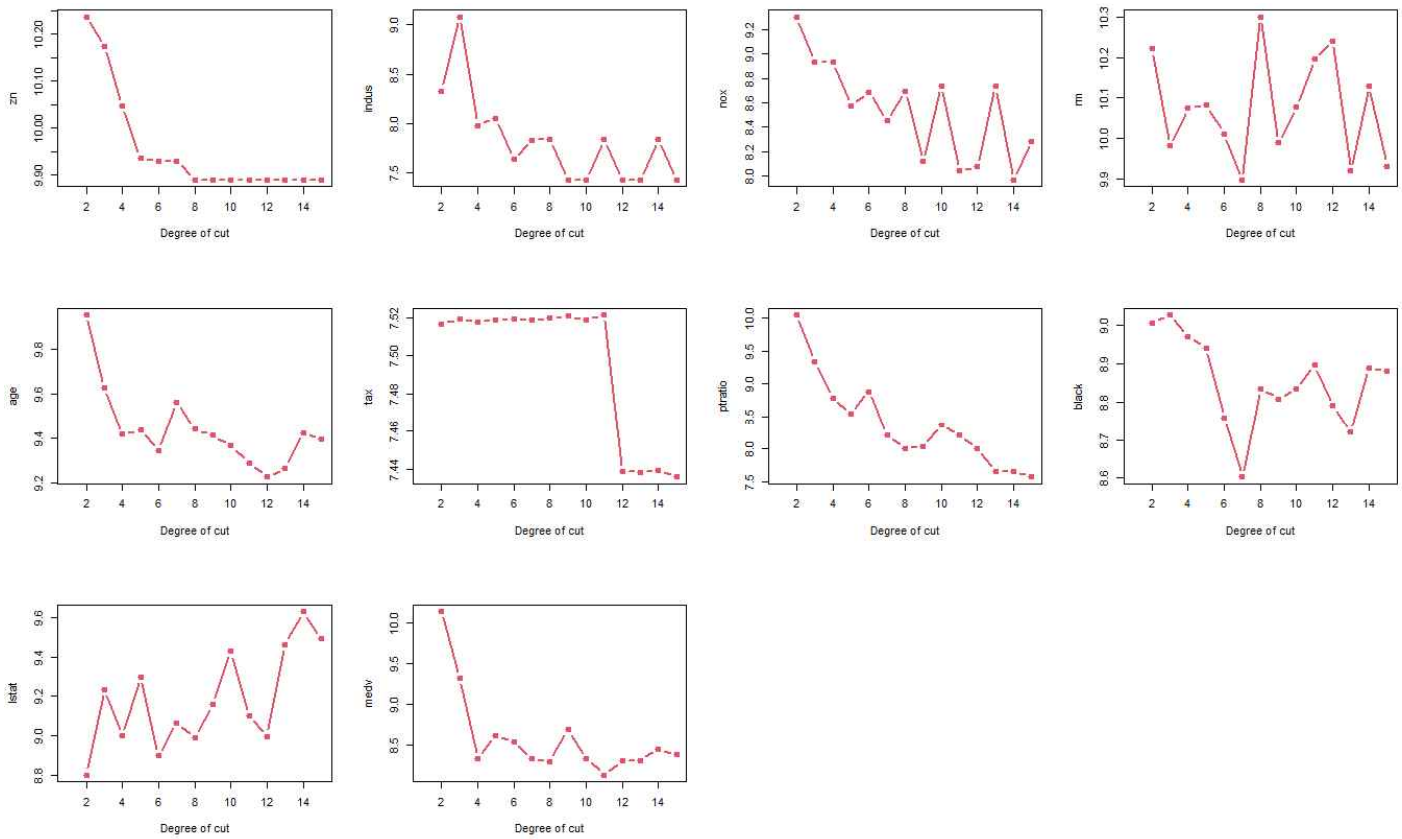
since X_j must be in exactly one of the $k+1$ intervals. The `R` function `cut(x[,j], k)` can generate `k` intervals of the jth predictor of `x`. This function will give you the numerical values of k cutpoints. Consider $k = 2, 3, \ldots, 15$ for each predictor. We then consider $C_1(X_j), C_2(X_j), \ldots, C_k(X_j)$ as predictors of a linear model, so

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \ldots + \beta_k C_k(x_i) + \epsilon_i,$$

where β_0 is the mean value of y_i for $X_j < c_1$. Therefore, you can use the `R` function `lm(y ~ cut(x[,j], k), subset=tran)` to fit the linear model with k categorical variables based on the training set. For each model (predictor), provide the optimal value of k and the numerical value of the corresponding PE.

다음 표는 각 변수에 대한 bin별 PE값을 나타낸 데이터이다. cut 모델은 문제 2번의 모델과 마찬가지로 bin가 1일 때는 성립하지 않는다. 이 모델들에 대한 각 변수별 bin별 PE값을 비교해 봤을 때 bin가 커짐에 따라 극명하게 PE값이 줄고 있으나 몇몇 그렇지 않은 값도 존재한다. 조금 더 쉽게 보기 위하여 그래프를 그려 확인해 적당한 bin를 가진 최종모델을 찾고자 한다.

	zn	indus	nox	rm	age	tax	ptratio	black	lstat	medv
1	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
2	10.237384	8.331454	9.306384	10.221088	9.956260	7.516401	10.050875	9.007296	8.801308	10.129553
3	10.174603	9.078592	8.936580	9.982053	9.629724	7.519271	9.331603	9.028349	9.236301	9.318818
4	10.047959	7.983341	8.932184	10.075966	9.416435	7.517645	8.788289	8.972592	9.001919	8.343028
5	9.935773	8.055087	8.579161	10.082605	9.437435	7.518578	8.548001	8.941918	9.298600	8.626412
6	9.930518	7.638364	8.689168	10.011389	9.346097	7.519404	8.885266	8.759931	8.897643	8.542247
7	9.930458	7.836360	8.454971	9.897713	9.564386	7.518492	8.206095	8.603050	9.066194	8.336830
8	9.889459	7.836933	8.697636	10.300382	9.439827	7.519624	8.010652	8.836380	8.991128	8.308199
9	9.889457	7.431258	8.119743	9.989824	9.411699	7.520723	8.042665	8.809259	9.159405	8.692608
10	9.889555	7.431066	8.735842	10.077811	9.364920	7.518694	8.373055	8.834463	9.428193	8.340305
11	9.889513	7.836986	8.042839	10.196569	9.289321	7.521424	8.206914	8.896623	9.102116	8.142274
12	9.889432	7.431655	8.075348	10.240534	9.225558	7.438804	8.010823	8.790334	8.997233	8.312744
13	9.889432	7.431376	8.735875	9.920226	9.264945	7.438271	7.654929	8.720475	9.464374	8.318669
14	9.889438	7.836565	7.967711	10.130295	9.425014	7.439210	7.655894	8.889505	9.630187	8.453464
15	9.889510	7.431552	8.288170	9.932063	9.396321	7.435804	7.571922	8.882569	9.493663	8.386798



zn 변수는 bin가 4부터 최솟값과 PE가 비슷하다. 이중 가장 간단한 모델인 bin가 4인 모델을 선택한다.
indus 변수는 bin가 6에서 최솟값인 bin 10일 때 PE와 비슷하므로 bin가 6인 모델을 선택한다.
nox 변수는 bin가 9에서 최솟값인 bin 14일 때 PE와 비슷하므로 bin가 9인 모델을 선택한다.
rm 변수는 bin 7에서 최솟값을 가지고 이보다 간단한 모델 중 최소인 bin 3일 때와는 PE값이 차이가 있으므로 bin 7인 모델을 택한다.
age 변수는 최솟값인 bin 6인 모델까지 점차 감소하고 그 이후는 증가했다가 다시 PE값이 감소한다. 또한 최솟값인 bin 12인 모델과 큰 차이가 없어보이므로 bin 6인 모델을 택한다.
tax는 bin 12 이후로 PE 감소폭이 줄어들어 간단한 모델인 bin 12 모델을 선택한다.
ptratio 변수는 bin 13까지 점점 감소하고 15일 때 최솟값을 가진다. 이때 모델이 너무 복잡하므로 이전 값 중 PE값이 최소인 bin 8인 모델을 선택한다.
black 변수는 bin 7에서 최솟값을 가지고 다른값과 차이가 크므로 bin 7인 모델을 선택한다.
lstat 변수는 가장 간단하고 PE값이 최소인 bin 2인 모델을 선택한다.
medv 변수는 bin 4에서 간단한 모델이면서 최솟값과 큰 차이가 없으므로 이 때의 모델을 선택한다.

선택한 차수별 PE와 최솟값 일 때의 차수를 비교해보면 다음과 같다.

	df	model_cut_min	cut_df_pe
zn	12	9.889432	4 10.047959
indus	10	7.431066	6 7.638364
nox	14	7.967711	9 8.119743
rm	7	9.897713	7 9.897713
age	12	9.225558	6 9.346097
tax	15	7.435804	12 7.438804
ptratio	15	7.571922	8 8.010652
black	7	8.603050	7 8.603050
lstat	2	8.801308	2 8.801308
medv	11	8.142274	4 8.343028

4. You have a total of 30 models so far (10 models from each question) and the corresponding 30 PEs. You can pick up the best model among 30 models in terms of the smallest PE. However, this result is based on validation set approach (450 training sets and 55 test sets). Next, repeat Q1, Q2 and Q3 to obtain the optimal k and PE from 30 models using the 10-fold cross-validation, instead of the validation set approach. You must use the following R code to generate 10 folds.

```
> RNGkind(sample.kind = "Rounding")
> set.seed(123)
> u <- sample(rep(seq(10), length=length(y)))
```

For 30 models, provide the PE along with the optimal value of k using the following table,

	Q1		Q2		Q3	
	k	PE	k	PE	k	PE
Model 1						
Model 2						
⋮						
⋮						
Model 9						
Model 10						

Note that Model j means that you use only the jth predictor of X in your model.

순서대로 K가 10일 때 K-Fold CV를 적용한 poly, smooth.spline, cut모델의 변수별 degree 및 bin 변화에 따른 PE값을 폴드를 적용 후 평균을 내어 나타내었다. 이 때 cut모델의 경우, K-Fold 과정에서 train set에서 값이 없어 학습 후 적합이 되지 않은 모델은 제외하여 계산해주었다.

	zn	indus	nox	rm	age	tax	ptratio	black	lstat	medv
1	7.764632	7.153790	7.078976	7.722747	7.395469	6.219571	7.556741	7.320903	6.891416	7.336025
2	7.703729	7.103453	6.989146	7.654422	7.223482	6.051448	7.462332	7.336146	6.852404	6.390502
3	7.693159	6.688895	6.458106	7.718315	7.168041	6.062926	7.423095	7.368133	6.917562	6.085055
4	7.691253	6.700197	6.473243	7.680063	7.135158	5.880345	6.983541	7.403448	6.952188	6.046121
5	7.690804	6.447092	6.370346	7.463862	7.111652	5.846292	6.870746	7.421693	7.093144	6.095233
6	7.690750	6.434675	6.298403	7.508424	7.099662	5.842962	6.888541	7.450100	7.283909	6.163593
7	7.690498	6.286435	6.313295	7.584068	7.102130	5.841459	6.682934	7.431888	7.303774	6.208309
8	7.690560	6.144567	6.292272	8.016352	7.124145	5.838863	6.378007	7.440734	7.405494	6.188905
9	7.690568	6.172639	6.269080	8.025323	7.129639	5.837875	6.318493	7.442635	7.803821	6.149528
10	7.690782	6.019761	6.295286	9.175604	7.115478	5.837683	6.326817	7.463184	8.475927	6.087013
11	7.690151	5.876860	6.228858	14.501742	7.117579	5.835313	6.380096	7.486427	9.341165	6.022565
12	7.695825	5.844785	5.990084	22.961144	7.110590	5.834067	6.334564	7.449606	9.753883	5.978361
13	7.789408	5.829054	5.827890	22.970386	7.057284	5.834129	6.259286	7.477424	10.553573	5.985494
14	8.299344	5.824231	5.837456	29.155698	7.034982	5.836658	6.518037	7.471222	12.320228	6.013383
15	16872.998462	5.824020	5.891091	147.889989	7.096718	8.253498	7.283650	7563.652991	7.456129	6.042527

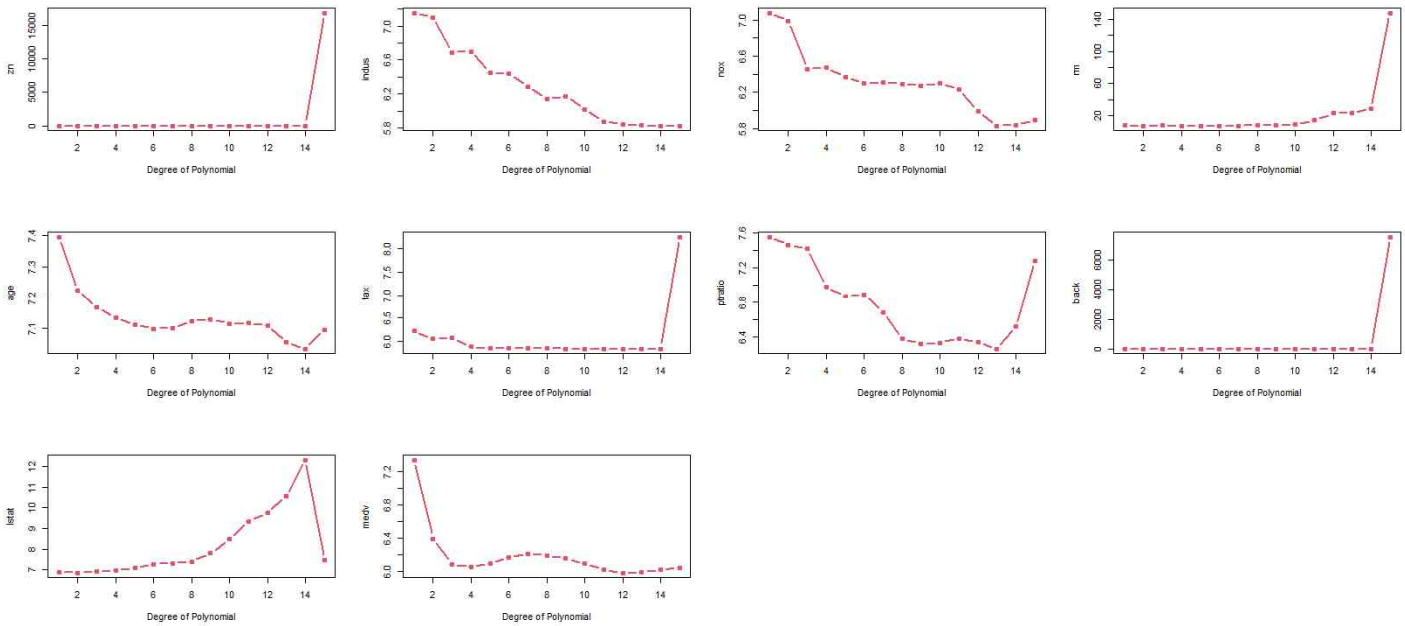
	zn	indus	nox	rm	age	tax	ptratio	black	lstat	medv
1	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
2	7.764601	7.147196	7.076460	7.705647	7.388113	6.212770	7.556637	7.369841	6.889334	7.307809
3	7.695434	6.802310	6.714509	7.625953	7.211725	6.030257	7.331751	7.369838	6.860346	6.289248
4	7.690369	6.502569	6.403238	7.612825	7.170457	5.999063	7.078924	7.369839	6.873054	6.066036
5	7.689916	6.344580	6.297534	7.607940	7.148609	5.944121	6.784193	7.378987	6.898110	6.014473
6	7.689998	6.214915	6.231306	7.598488	7.132189	5.890841	6.547600	7.395359	6.923473	6.018421
7	7.690031	6.114684	6.160845	7.585333	7.119324	5.857382	6.395103	7.408159	6.952870	6.036357
8	7.690032	6.046686	6.089024	7.568700	7.109574	5.842076	6.299816	7.418731	6.987471	6.053596
9	7.690033	6.000858	6.016563	7.549881	7.102153	5.836488	6.235067	7.427026	7.024796	6.066093
10	7.690041	5.969209	5.946435	7.531735	7.096239	5.834427	6.187964	7.433063	7.063250	6.072962
11	7.690052	5.947287	5.882936	7.517800	7.091022	5.833439	6.152371	7.437597	7.101750	6.074289
12	7.690064	5.932032	5.830955	7.509856	7.086259	5.832789	6.125539	7.441722	7.140237	6.070756
13	7.690075	5.921317	5.791282	7.508067	7.081878	5.832255	6.105488	7.446243	7.178359	6.063713
14	7.690083	5.913545	5.762582	7.511016	7.078176	5.831761	6.090704	7.451502	7.215690	6.054756
15	7.690090	5.907664	5.742905	7.516954	7.075118	5.831279	6.079898	7.457491	7.251854	6.045608

	zn	indus	nox	rm	age	tax	ptratio	black	lstat	medv
1	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
2	7.861653	6.692967	6.693072	7.915900	7.657400	5.978787	7.779285	7.398538	7.162443	7.808928
3	7.834300	6.995816	7.058580	7.609506	7.520923	5.974004	7.310032	7.397490	7.106968	7.389418
4	7.780062	6.361288	6.372720	7.575934	7.409377	5.976108	6.967565	7.401520	7.067929	6.837823
5	7.706520	6.469114	6.466558	7.637177	7.446744	5.970951	6.811644	7.431959	7.071903	6.623689
6	7.701049	5.942396	6.254104	7.663644	7.372924	5.973554	6.980792	7.443800	7.068094	6.272005
7	7.701030	6.288678	6.196586	7.695470	7.347068	5.972983	6.497942	7.435742	7.061181	6.016568
8	7.690167	6.287588	6.387826	7.566566	7.293547	5.972705	6.338820	7.446311	7.127113	6.167245
9	7.690168	5.824566	5.879749	7.555991	7.301604	5.970861	6.737357	7.378345	7.107138	6.481921
10	7.690224	5.829377	6.389800	7.616653	7.304132	5.970546	6.669915	7.514263	7.151694	6.410025
11	8.050183	6.287273	5.908874	7.664045	7.269484	5.970605	5.846437	7.078923	7.183777	6.225435
12	7.690153	5.824025	5.914147	7.489669	7.330699	5.839735	5.698144	7.610756	7.289617	6.237090
13	8.050132	5.824647	6.438583	7.565534	7.329031	5.838302	5.440334	7.081452	7.252879	6.243196
14	7.690165	6.287827	5.823535	7.608529	7.323043	5.838663	5.859192	7.592860	7.263517	6.154064
15	8.050178	5.823662	6.173758	7.637454	7.313054	5.836220	5.698901	7.135907	7.347872	6.115166

각각의 모델에서 PE의 최솟값에 대한 degree값을 나타내주었다.

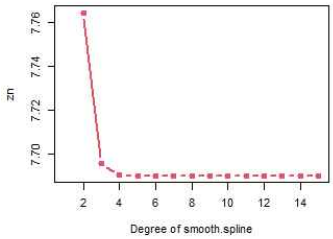
```
> cbind(model_KCV1_, model_KCV2_, model_KCV3_)
        df model_KCV1 df model_KCV2 df model_KCV3
zn      11   7.690151  5   7.689916 12   7.690153
indus   15   5.824020 15   5.907664 15   5.823662
nox     13   5.827890 15   5.742905 14   5.823535
rm       5   7.463862 13   7.508067 12   7.489669
age     14   7.034982 15   7.075118 11   7.269484
tax     12   5.834067 15   5.831279 15   5.836220
ptratio 13   6.259286 15   6.079898 13   5.440334
black    1   7.320903  3   7.369838 11   7.078923
lstat    2   6.852404  3   6.860346  7   7.061181
medv    12   5.978361  5   6.014473  7   6.016568
```

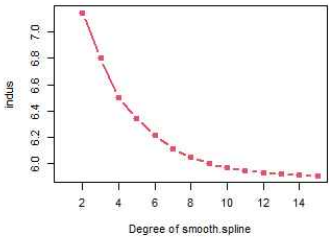
최솟값인 PE를 계산했으므로 모델의 차수가 높아질 위험이 있다. 따라서 모델의 그래프를 확인하여 값을 비교해보고자한다.

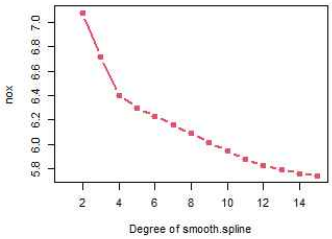


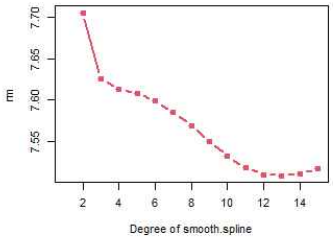
첫 번째 모델인 poly모델을 적용한 K-Fold 결과를 살펴본다.

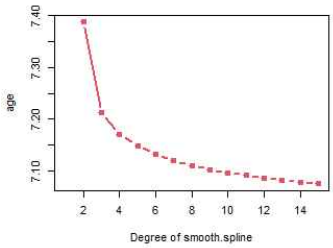
zn 변수는 degree가 1부터 14까지 큰 변화가 없으므로 가장 간단한 모델인 degree가 1인 모델을 선택한다.
indus 변수는 degree가 11까지 점차 감소하고 이후로는 큰 변화가 없어 보여 degree가 11인 모델을 선택한다.
nox 변수는 degree가 13에서 크게 감소하였고 이후는 모델이 너무 복잡하여 degree가 13인 모델을 택한다.
rm 변수는 degree 5에서 최솟값을 가지고 이전 값들과 차이가 없으므로 degree 5인 모델을 택한다.
age 변수는 degree가 6인 변수까지 감소하다가 조금 증가폭을 보이다 degree 13부터 큰 감소폭을 보였다. 하지만 최솟값인 degree 14일 때와 6일때의 PE값의 차이가 적으므로 degree 6인 모델을 택한다.
tax는 degree 4 이후로 PE 감소폭이 줄어들어 간단한 모델인 degree 4 모델을 선택한다.
ptratio 변수는 degree 8까지 점점 감소한다. 8일 때의 PE값이 최솟값과 큰 차이가 없으므로 degree 8 모델을 선택한다.
black 변수는 degree 1에서 간단한 모델이면서 최솟값을 가지므로 이 때의 모델을 선택한다.
lstat 변수는 degree 1일 때의 PE값이 최솟값인 degree가 2일 때의 PE값과 큰 차이가 없으므로 간단한 모델인 degree 1인 모델을 선택한다.
medv 변수는 degree 3에서 간단한 모델이면서 최솟값을 가지므로 이 때의 모델을 선택한다.

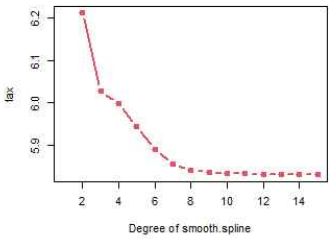


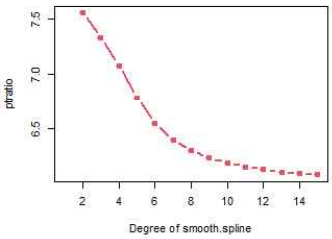


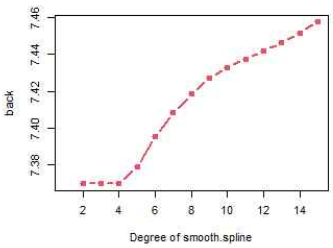


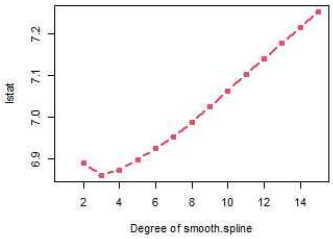


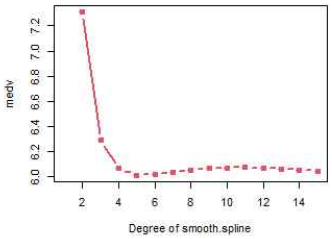






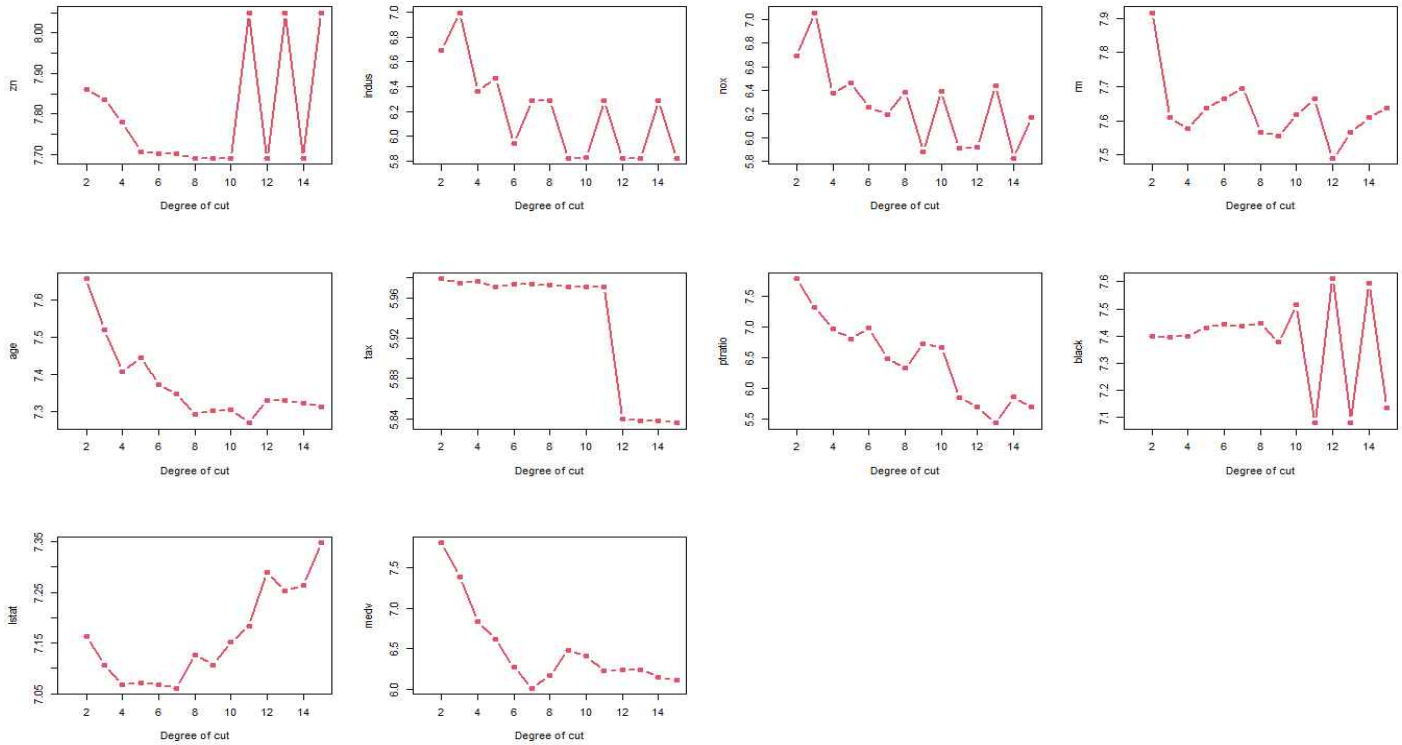






두 번째 모델인 smooth.spline모델을 적용한 K-Fold 결과를 살펴본다.

zn 변수는 degree가 4부터 15까지 큰 변화가 없으므로 가장 간단한 모델인 degree가 4인 모델을 선택한다.
indus 변수는 degree가 10까지 점차 감소한다. 이후로는 큰 변화가 없어 보여 degree가 8인 모델을 선택한다.
nox 변수는 degree가 15까지 계속 감소한다. degree 12인 모델에서 부터는 최솟값과 큰 차이가 없으므로 degree가 12인 모델을 택한다.
rm 변수는 degree 13에서 최솟값을 가지고 12의 모델과는 큰 차이가 없어 degree 12인 모델을 택한다.
age 변수는 최솟값인 degree 10인 모델까지 점차 감소하고 그 이후는 큰 변화가 없다. degree 10인 모델을 택한다.
tax는 degree 8 이후로 PE 감소폭이 줄어들어 간단한 모델인 degree 8 모델을 선택한다.
ptratio 변수는 degree 12까지 점점 감소한다. 이후에는 큰 변화가 없으므로 degree 12인 모델을 선택한다.
black 변수는 degree 3에서 최솟값을 가지지만 2인 모델의 PE값에서 큰 차이가 없으므로 degree 2인 모델을 선택한다.
lstat 변수는 degree 3일 때의 PE값이 최솟값을 가지며 간단한 모델이므로 이 때의 모델을 선택한다.
medv 변수는 degree 5에서 최솟값을 가지고 이는 앞의 degree가 4일때의 모델과 PE값의 큰 차이가 나지 않으므로 가지므로 이 때의 모델을 선택한다.

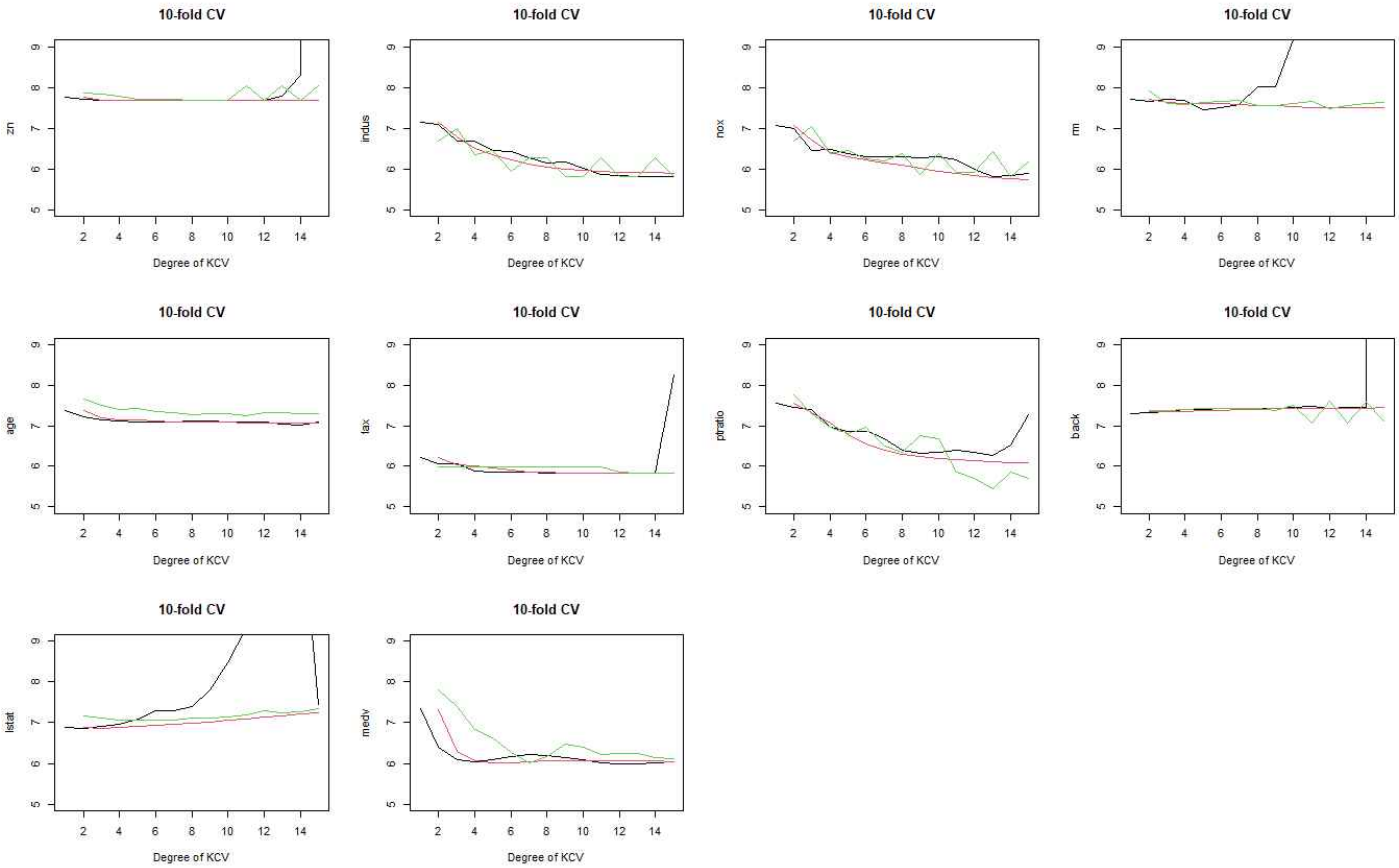


마지막 모델인 cut을 적용한 K-Fold 결과를 살펴본다.

zn 변수는 bin가 12에서 최솟값을 가지나 bin가 5에서의 PE와 큰 차이가 없으므로 bin가 5인 모델을 선택한다.
indus 변수는 bin가 6에서 최솟값인 bin 15일 때 PE와 비슷하므로 bin가 6인 모델을 선택한다.
nox 변수는 bin가 9에서 최솟값인 bin 14일 때 PE와 비슷하므로 bin가 9인 모델을 선택한다.
rm 변수는 bin 12에서 최솟값을 가지고 이보다 bin가 낮은 모델 중 최소인 bin 4일 때와는 PE값이 차이가 적으므로 bin 4인 모델을 택한다.
age 변수는 최솟값인 bin 8인 모델까지 점차 감소하고 그 이후는 증가했다가 다시 PE값이 감소한다. 또한 최솟값인 bin 11인 모델과 큰 차이가 없어보이므로 bin 8인 모델을 택한다.
tax는 bin 12 이후로 PE 감소폭이 줄어들어 간단한 모델인 bin 12 모델을 선택한다.
ptratio 변수는 bin 13까지 점점 감소하고 이 때 최솟값을 가진다. 다른 간단한 모델들과는 PE값에서 차이가 있기에 bin 13인 모델을 선택한다.
black 변수는 bin 11에서 최솟값을 가지고 다른값과 차이가 크므로 bin 11인 모델을 선택한다.
lstat 변수는 bin 7에서 최솟값을 가지고 bin 4에서의 PE값이 차이가 없으므로 bin 4인 모델을 선택한다.
medv 변수는 bin 7에서 간단한 모델이면서 최솟값이므로 이 때의 모델을 선택한다.

	KCV_poly		KCV_spline		KCV_cut	
zn	1	7.764632	4	7.690369	5	7.706520
indus	11	5.876860	8	6.046686	6	5.942396
nox	13	5.827890	12	5.830955	9	5.879749
rm	5	7.463862	12	7.509856	4	7.575934
age	6	7.099662	10	7.096239	8	7.293547
tax	4	5.880345	8	5.842076	12	5.839735
ptratio	8	6.378007	12	6.125539	13	5.440334
black	1	7.320903	2	7.369841	11	7.078923
lstat	1	6.891416	3	6.860346	4	7.067929
medv	3	6.085055	5	6.014473	7	6.016568

최종적으로 선택된 모델들의 degree와 이 때의 PE값은 다음과 같다.



검은색이 poly, 빨간색이 smooth.spline, 연두색이 cut모델이다.
전체 그래프의 경향을 보았을 때 poly의 모델의 경우 전반적으로 PE값이 커보이고, smooth.spline 모델의 경우 PE값이 낮아보인다.

5. Which model among 30 models is the best model when you use the 10-fold cross-validation? How about the validation set approach? Did they pick up the same model as the best model?

validation set approach의 선택 모델 결과

```
> val_fit
            poly_df_pe     spline_df_pe     cut_df_pe
zn         1  10.043964  4      9.903471  4 10.047959
indus     11   7.493358  8      7.816802  6  7.638364
nox       13   7.840010 15      7.952312  9  8.119743
rm         2   9.942636  4      9.928371  7  9.897713
age        8   8.768530 12      8.895086  6  9.346097
tax        4   7.463775  8      7.454947 12  7.438804
ptratio    8   8.056924 12      7.809007  8  8.010652
black      1   8.745484  2      8.726939  7  8.603050
lstat      3   8.858170  3      8.899067  2  8.801308
medv       3   8.147778  4      8.103097  4  8.343028
```

validation set approach에서 최종적으로 각 방법들의 PE와 모델의 degree 및 bin을 고려하였을 때
zn 변수는 k=1 일 때 poly모델로 PE는 10.043964
indus 변수는 k=6 일 때 cut모델로 PE는 7.638364
nox 변수는 k=13 일 때 poly모델로 PE는 7.840010
rm 변수는 k=2 일 때 poly모델로 PE는 9.942636
age 변수는 k=8 일 때 poly모델로 PE는 8.768530
tax 변수는 k=4 일 때 poly모델로 PE는 7.463775
ptratio 변수는 k=8 일 때 cut모델로 PE는 8.010652
black 변수는 k=1 일 때 poly모델로 PE는 8.745484
lstat 변수는 k=2 일 때 cut모델로 PE는 8.801308
medv 변수는 k=3 일 때 poly모델로 PE는 8.147778
다음과 같이 선택하였다.

10-fold cross-validation의 선택 모델 결과

```
> KCV_fit
            KCV_poly     KCV_spline      KCV_cut
zn         1 7.764632  4    7.690369  5 7.706520
indus     11 5.876860  8    6.046686  6 5.942396
nox       13 5.827890 12    5.830955  9 5.879749
rm         1 7.722747 12    7.509856  9 7.555991
age        6 7.099662 10    7.096239  8 7.293547
tax        4 5.880345  8    5.842076 12 5.839735
ptratio    8 6.378007 12    6.125539 13 5.440334
black      1 7.320903  2    7.369841 11 7.078923
lstat      1 6.891416  3    6.860346  4 7.067929
medv       3 6.085055  5    6.014473  7 6.016568
```

10-Fold cross-validation에서 최종적으로 각 방법들의 PE와 모델의 degree 및 bin을 고려하였을 때
zn 변수는 k=1 일 때 poly모델로 PE는 7.764632
indus 변수는 k=6 일 때 cut모델로 PE는 5.942396
nox 변수는 k=9 일 때 cut모델로 PE는 5.879749
rm 변수는 k=1 일 때 poly모델로 PE는 7.722747
age 변수는 k=6 일 때 poly모델로 PE는 7.099662
tax 변수는 k=4 일 때 poly모델로 PE는 5.880345
ptratio 변수는 k=8 일 때 poly모델로 PE는 6.378007
black 변수는 k=1 일 때 poly모델로 PE는 7.320903
lstat 변수는 k=1 일 때 poly모델로 PE는 6.891416
medv 변수는 k=3 일 때 poly모델로 PE는 6.085055
다음과 같이 선택하였다.

두 결과를 비교해보았을 때 모델선택에서 zn, tax, black, medv의 경우에 선택된 차수와 적합방법이 같았다.
나머지의 경우는 모두 달랐다. 먼저 각 30개씩 뽑힌 degree 및 bins와 PE값을 비교해보자

```
> summary(val_fit)
       V1          poly_df_pe          V3          spline_df_pe          V5          cut_df_pe
 Min.   : 1.00   Min.   : 7.464   Min.   : 2.0   Min.   :7.455   Min.   : 2.00   Min.   : 7.439
 1st Qu.: 2.25   1st Qu.: 7.894   1st Qu.: 4.0   1st Qu.:7.851   1st Qu.: 4.50   1st Qu.: 8.038
 Median : 3.50   Median : 8.447   Median : 6.0   Median :8.415   Median : 6.50   Median : 8.473
 Mean   : 5.40   Mean   : 8.536   Mean   : 7.2   Mean   :8.549   Mean   : 6.50   Mean   : 8.625
 3rd Qu.: 8.00   3rd Qu.: 8.836   3rd Qu.:11.0   3rd Qu.:8.898   3rd Qu.: 7.75   3rd Qu.: 9.210
 Max.   :13.00   Max.   :10.044   Max.   :15.0   Max.   :9.928   Max.   :12.00   Max.   :10.048
> summary(KCV_fit)
       V1          KCV_poly          V3          KCV_spline          V5          KCV_cut
 Min.   : 1.0   Min.   :5.828   Min.   : 2.00   Min.   :5.831   Min.   : 4.00   Min.   :5.440
 1st Qu.: 1.5   1st Qu.:5.932   1st Qu.: 4.25   1st Qu.:6.023   1st Qu.: 5.25   1st Qu.:5.895
 Median : 4.5   Median :6.635   Median : 8.00   Median :6.493   Median : 7.50   Median :6.542
 Mean   : 5.3   Mean   :6.659   Mean   : 7.60   Mean   :6.639   Mean   : 7.90   Mean   :6.584
 3rd Qu.: 7.5   3rd Qu.:7.266   3rd Qu.:11.50   3rd Qu.:7.301   3rd Qu.:10.50   3rd Qu.:7.240
 Max.   :13.0   Max.   :7.765   Max.   :12.00   Max.   :7.690   Max.   :13.00   Max.   :7.707
```

전체 PE값이 최소인 그리고 전체 degree값이 작은 모델이 더 좋다고 볼 수 있다.
평균값을 비교해보면 degree가 10-fold cross validation의 경우에 대체로 낮았고, PE값은 극명한 차이로 10-fold cross validation의 결과가 더 좋았다.

또한 위의 선택된 모델의 degree와 PE값을 살펴봤을 때 모든 경우에서 10-fold cross validation방법의 선택모델이 낮은 차수와 낮은 PE값을 가졌다. 따라서 validation set approach 보다는 10-fold cross validation으로 적합모델을 찾는 것이 이 데이터에선 더 적절해 보인다.