Semiparametric regression

Homework Assignment #2

Makowski Michał

31 march 2017

Contents

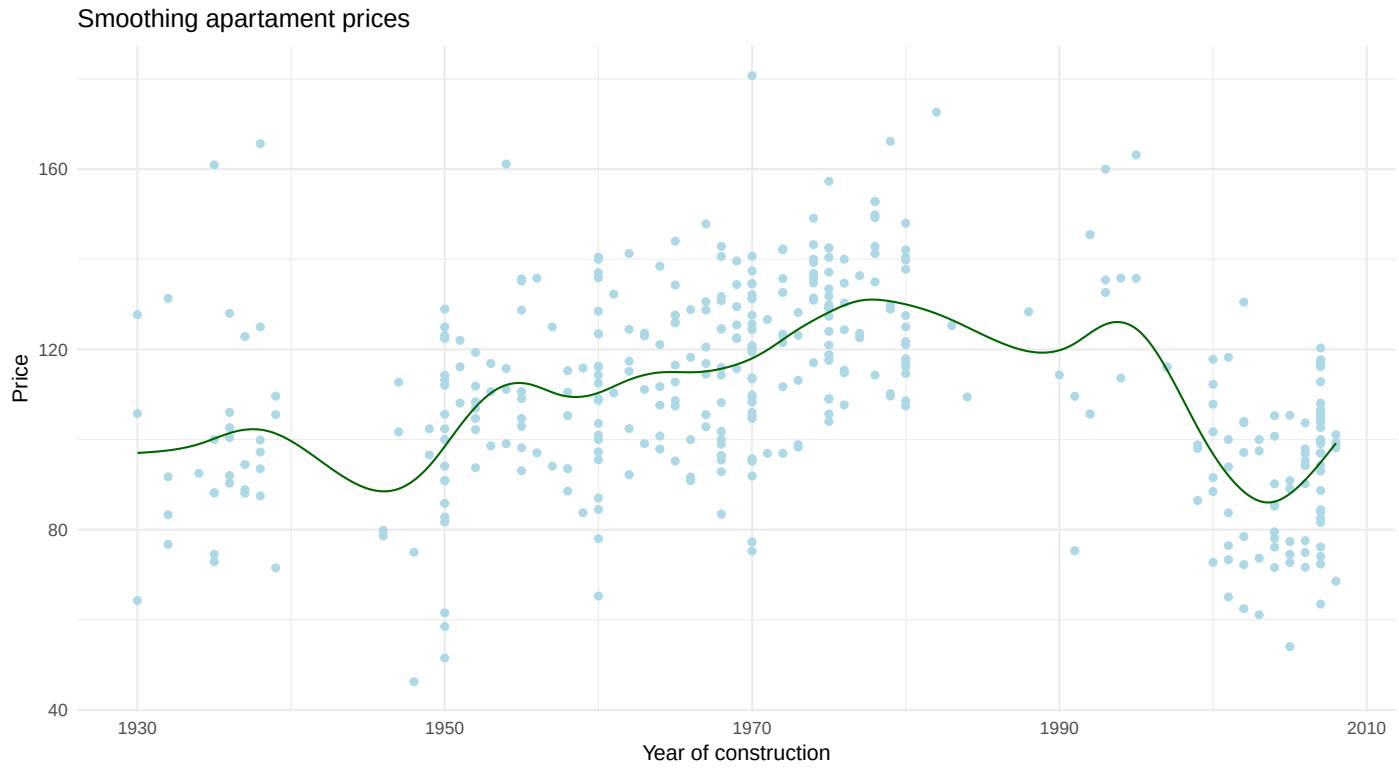
Those pages cover second homework for Semiparametric Regression, a course conducted by proffesor Jarosław Harężlak at University of Wrocław. On the following pages two exercises will be presented. First compare diffrent aproaches for obtaining fit line - by changing bases function, number of knots or type of estimation. Second is focused on regression model testing, linear model and nonlinear model will be teastes.

Exercise I

In exercise I we want to fit smooth line to the data about prices of apartments in Warsaw. Packages **HRW** and **mgcv** will be used. Package **mgcv** is a powerfull tool for fitting generalized additive models to data

At first, we fit and plot penalized spline to the data by issuing following code. It uses 30 cubic regression spline basis functions and GCV-based smoothing parameter selection.

```
x = WarsawApts$construction.date
y = WarsawApts$areaPerMzloty


fitGAMcr = gam(y~s(x, bs="cr", k=30))

xg = seq(min(x), max(x), length=1001)
fHatgGAMcr = predict(fitGAMcr, newdata=data.frame(x=xg))

points = cbind.data.frame(x, y)
fit1 = cbind.data.frame(xg, fHatgGAMcr)

ggplot() +
    geom_point(data=points, aes(x=x, y=y), colour="lightblue") +
    geom_line(data=fit1, aes(x=xg, y=fHatgGAMcr), colour="darkgreen") +
    labs(title="Smoothing apartament prices", x="Year of construction", y="Price") +
    theme_minimal()
```

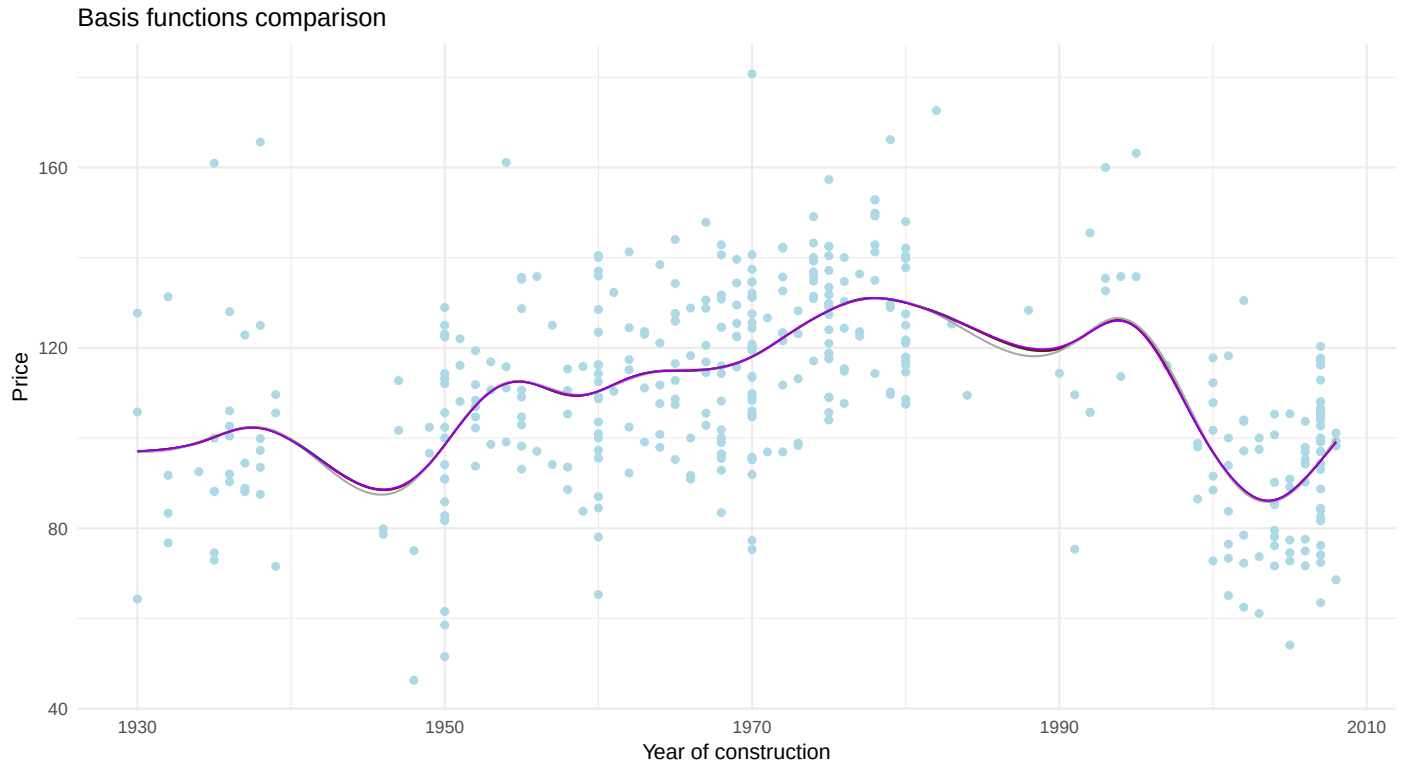


We can seem that fitted line represents trend in apartemnts prices, although there are time periods where we observe lack of data (for axample years 1980-1995) and line seems to be overfitted.

a)

In this point we compare diffrent types of penalized splines: Gaussians process basis functions, P-splines and thin plate regression. For more information on basis function and smoothers please visit THIS LINK. We will compare all 4 methods on one following plot:

```r
fitGAMgp = gam(y~s(x, bs="gp", k=30)) # Gaussian process
fHatgGAMgp = predict(fitGAMgp, newdata=data.frame(x=xg))

fitGAMps = gam(y~s(x, bs="ps", k=30)) # P-splines
fHatgGAMps = predict(fitGAMps, newdata=data.frame(x=xg))

fitGAMtp = gam(y~s(x, bs="tp", k=30)) # Thin plate
fHatgGAMtp = predict(fitGAMtp, newdata=data.frame(x=xg))

fit2 = cbind.data.frame(xg, fHatgGAMcr, fHatgGAMgp, fHatgGAMps, fHatgGAMtp)

ggplot() +
    geom_point(data=points, aes(x=x, y=y), colour="lightblue") +
    geom_line(data=fit2, aes(x=xg, y=fHatgGAMcr), colour="darkgreen") +
    geom_line(data=fit2, aes(x=xg, y=fHatgGAMgp), colour="darkred") +
    geom_line(data=fit2, aes(x=xg, y=fHatgGAMps), colour="darkgrey") +
    geom_line(data=fit2, aes(x=xg, y=fHatgGAMtp), colour="darkviolet") +
    labs(title="Basis functions comparison", x="Year of construction", y="Price") +
    theme_minimal()
```



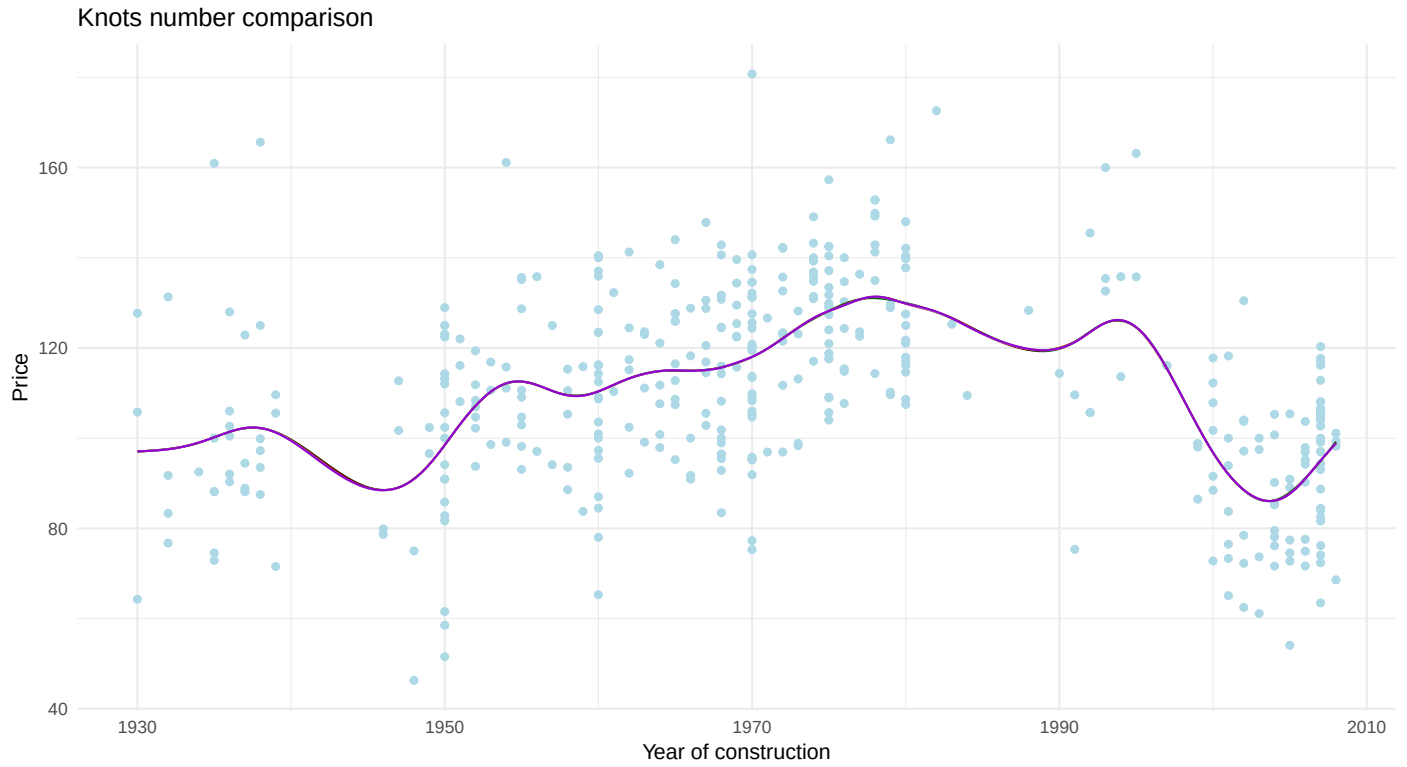
As we can see (or rather as we cannot see), in this particular example the diffrences beetwen diffrent basis function are marginal. Only in time periods, where number of observations is low there are visible little diffrences between them. We can conclude that choice of basis functions doeas not have large effect on line quality (on this particular data).

b)

In second point we compare diffrent number of knots (i.e. basis functions) used to fit. 30, 40, 50 and 60 basis functions will be caompared on one, following plot:

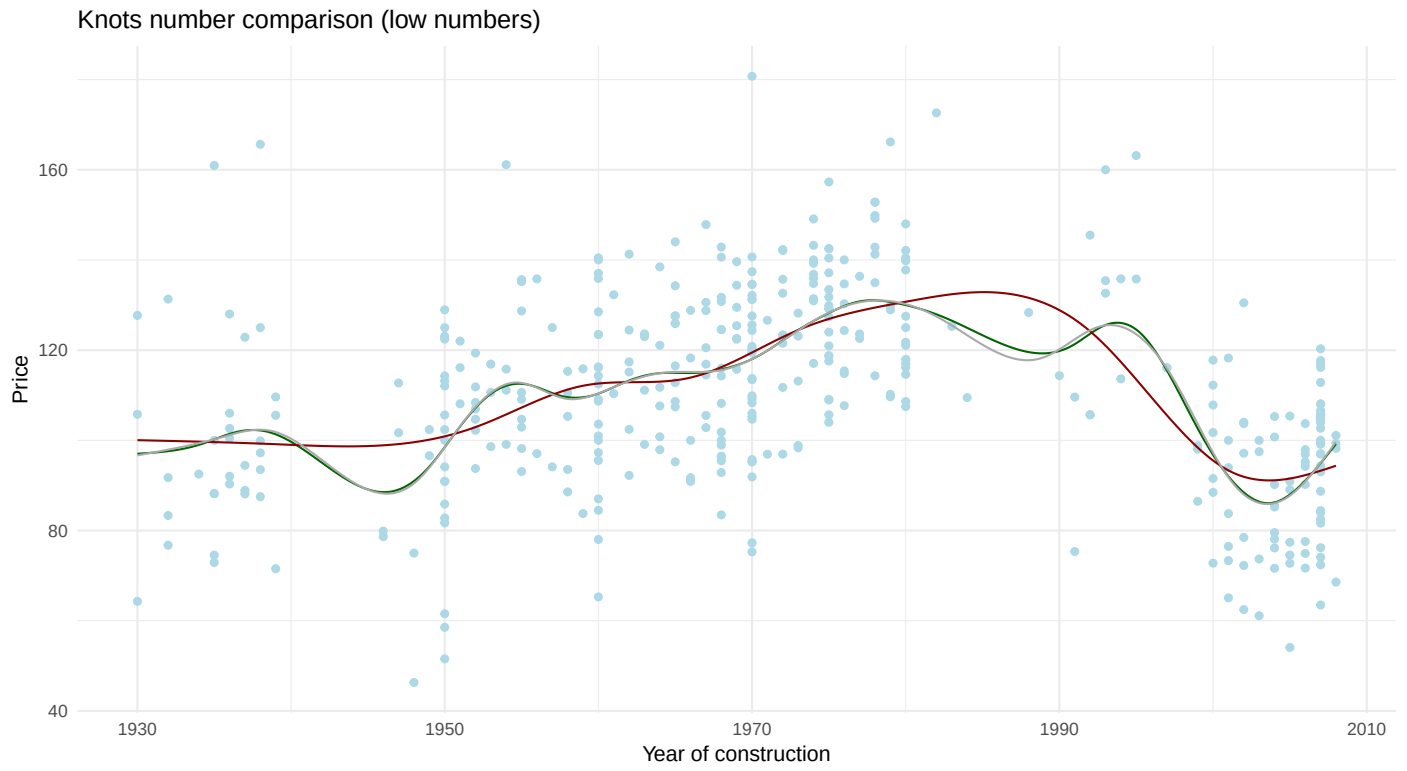
```r
fitGAMcr40 = gam(y~s(x, bs="cr", k=40))
fHatgGAMcr40 = predict(fitGAMcr40, newdata=data.frame(x=xg))

fitGAMcr50 = gam(y~s(x, bs="cr", k=50))
fHatgGAMcr50 = predict(fitGAMcr50, newdata=data.frame(x=xg))
```

```
fitGAMcr60 = gam(y~s(x, bs="cr", k=60))
fHatgGAMcr60 = predict(fitGAMcr60, newdata=data.frame(x=xg))

fit3 = cbind.data.frame(xg, fHatgGAMcr, fHatgGAMcr40, fHatgGAMcr50, fHatgGAMcr60)

ggplot() +
    geom_point(data=points, aes(x=x, y=y), colour="lightblue") +
    geom_line(data=fit3, aes(x=xg, y=fHatgGAMcr), colour="darkgreen") +
    geom_line(data=fit3, aes(x=xg, y=fHatgGAMcr40), colour="darkred") +
    geom_line(data=fit3, aes(x=xg, y=fHatgGAMcr50), colour="darkgrey") +
    geom_line(data=fit3, aes(x=xg, y=fHatgGAMcr60), colour="darkviolet") +
    labs(title="Knots number comparison", x="Year of construction", y="Price") +
    theme_minimal()
```



As we can see, there is on diffrence visible beetwen all four fits. It means, that for this data, there exist *minimal* number of knots which provide *full* model. Let's plot comparison beetwen 10, 20 and 30 knots:

```
fitGAMcr10 = gam(y~s(x, bs="cr", k=10))
fHatgGAMcr10 = predict(fitGAMcr10, newdata=data.frame(x=xg))

fitGAMcr20 = gam(y~s(x, bs="cr", k=20))
fHatgGAMcr20 = predict(fitGAMcr20, newdata=data.frame(x=xg))

fit4 = cbind.data.frame(xg, fHatgGAMcr, fHatgGAMcr10, fHatgGAMcr20)

ggplot() +
    geom_point(data=points, aes(x=x, y=y), colour="lightblue") +
    geom_line(data=fit4, aes(x=xg, y=fHatgGAMcr), colour="darkgreen") +
    geom_line(data=fit4, aes(x=xg, y=fHatgGAMcr10), colour="darkred") +
    geom_line(data=fit4, aes(x=xg, y=fHatgGAMcr20), colour="darkgrey") +
    labs(title="Knots number comparison (low numbers)", x="Year of construction", y="Price") +
    theme_minimal()
```



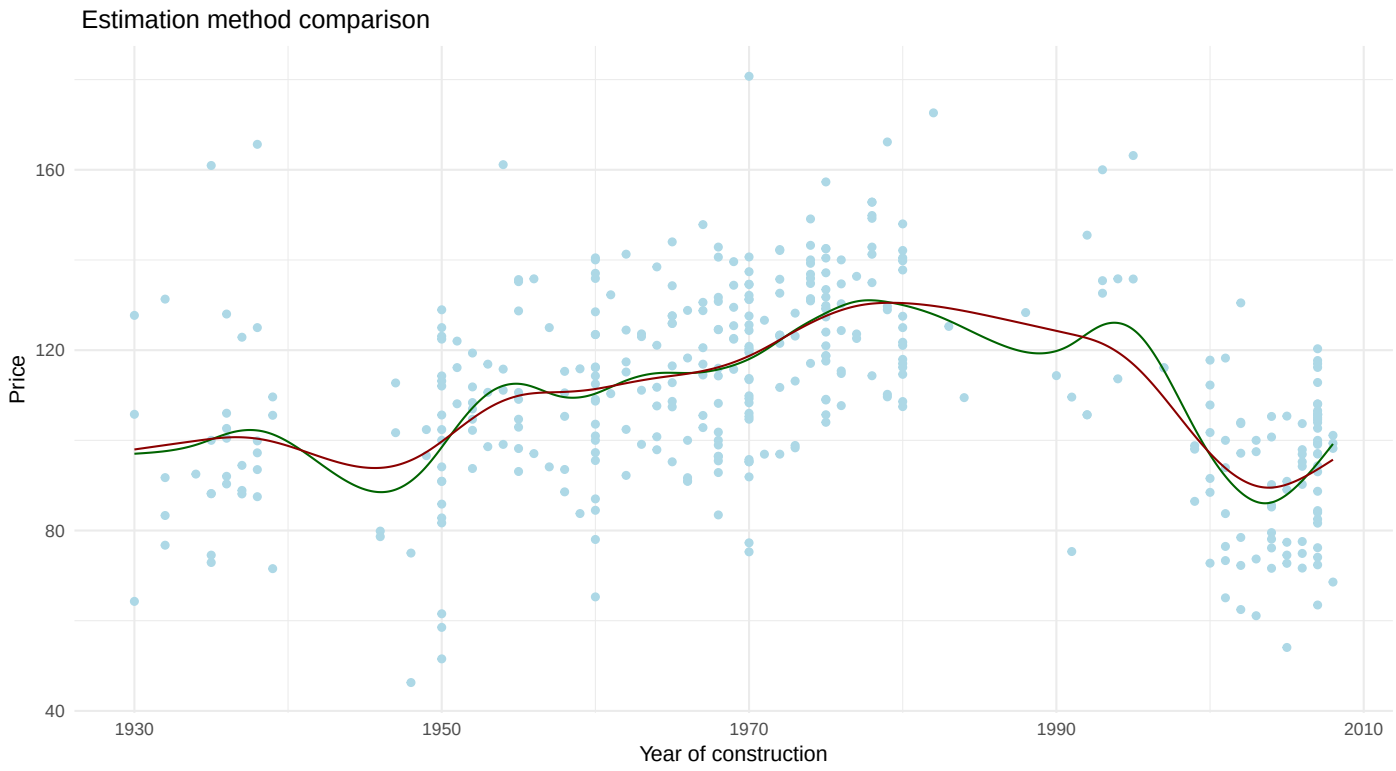
It seems that even 20-knot-model is quite similar to the 30-and-more-knots-models. It represents all trend in the data, every extremum present in higher models also also visible in 20-knots-model. 10-knots-model is much diffrent, it is less fitted to the data, but, for me, it is not overfitted, it represent general trend in the data, but it is not so sensitive for the outliners. Common way to choose number of knots is: $N_{\text{knots}} = \min\{\frac{\text{number of unique } x_i}{4}, 35\}$

We do not see diffrences beetwen 30, 40, 50 and 60 knots due to the fact, tha relation beetwen number of knots and fit is logarithmic, that means we have to choose hundreds of knots to see the diffrence, but on provided data that number of knots does not make any sense.

c)

In last but one point of this exercise, the **REML** and **GCV** smoothing paramether estimation methods will be compared. Diffrent methods are also possible, please visit THIS LINK for more information.

```
fitGAMcrREML = gam(y~s(x, bs="cr", k=30), method="REML")
fHatgGAMcrREML = predict(fitGAMcrREML, newdata=data.frame(x=xg))

fit5 = cbind.data.frame(xg, fHatgGAMcr, fHatgGAMcrREML)

ggplot() +
    geom_point(data=points, aes(x=x, y=y), colour="lightblue") +
    geom_line(data=fit5, aes(x=xg, y=fHatgGAMcr), colour="darkgreen") +
    geom_line(data=fit5, aes(x=xg, y=fHatgGAMcrREML), colour="darkred") +
    labs(title=" Estimation method comparison", x="Year of construction", y="Price") +
    theme_minimal()
```



It is visible that, in contarary to earlier comparisons, choice of estimation methdos has strong effect on fit. Those two methods are based on diffrent approaches, **REML** is based on likelihood, **GCV** on model selection. If we compare methods based on model selection, we should receive almost the same fit for every of them (they are equvalent). **REML** looks slighty like **GCV** with smaller number of knots, it is not strongly sensitive for outliners.

d)

Every point was followed by a short analysis of every model paramether, let's conclude everything in table.

Paramather	Impact of fit
Type of basis	Weak
Number of basis functions	Strong (under certain level)/Weak (over it)
Method of estimation	Strong (beetwen approaches)

Exercise II

The objective of second exercise is to test hyphotesis about nonparametric regresion model.

Let's consider

$$y_i = f(x_i) + \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2), \qquad 1 \leqslant i \leqslant n$$

and the hypothesis testing problem:

$$\mathrm{H}_0 : f \text{ is a linear function} \qquad \text{VS} \qquad \mathrm{H}_1 : f \text{ is a smooth nonlinear function}$$

The package **RLRsim** will be used in this exercise.

b)

At the beggining the set of 200 observations will be generated.

```
set.seed(1)
x = seq(0, 1, length=200)
y = x + rnorm(200)
```

It is set of points evenly spread on the line $y = x$ on $[0, 1]$ with the standard white gaussian noise added.

c)

3 tests will be conducted on this data:

1) F-test isung P-spline
2) F-test using OLS
3) RLRT (Restricted Likelihood Ratio Test)

Let's conduct those tests:

```
fitLine = gam(y~x)
fitDfltPenSpl = gam(y~s(x))
anova(fitLine, fitDfltPenSpl, test="F")$"Pr(>F)"[2] # P-spline
```

```
[1] 7.514685e-09
```

```
fitOLSspl = gam(y~s(x, k=5, sp=0))
anova(fitLine, fitOLSspl, test="F")$"Pr(>F)"[2] # OLS
```

```
[1] 0.6872285
```

```
fitGAMM = gamm(y~s(x), method="REML")
exactRLRT(fitGAMM$lme)$p.value # RLRT
```

```
[1] 1
```

It is visible, that tests behave totally diffrent, for **P-spline** p-value is really small, so we reject null hypothesis, for two other tests p-value is much bigger, so null hypothesis should not be rejected.

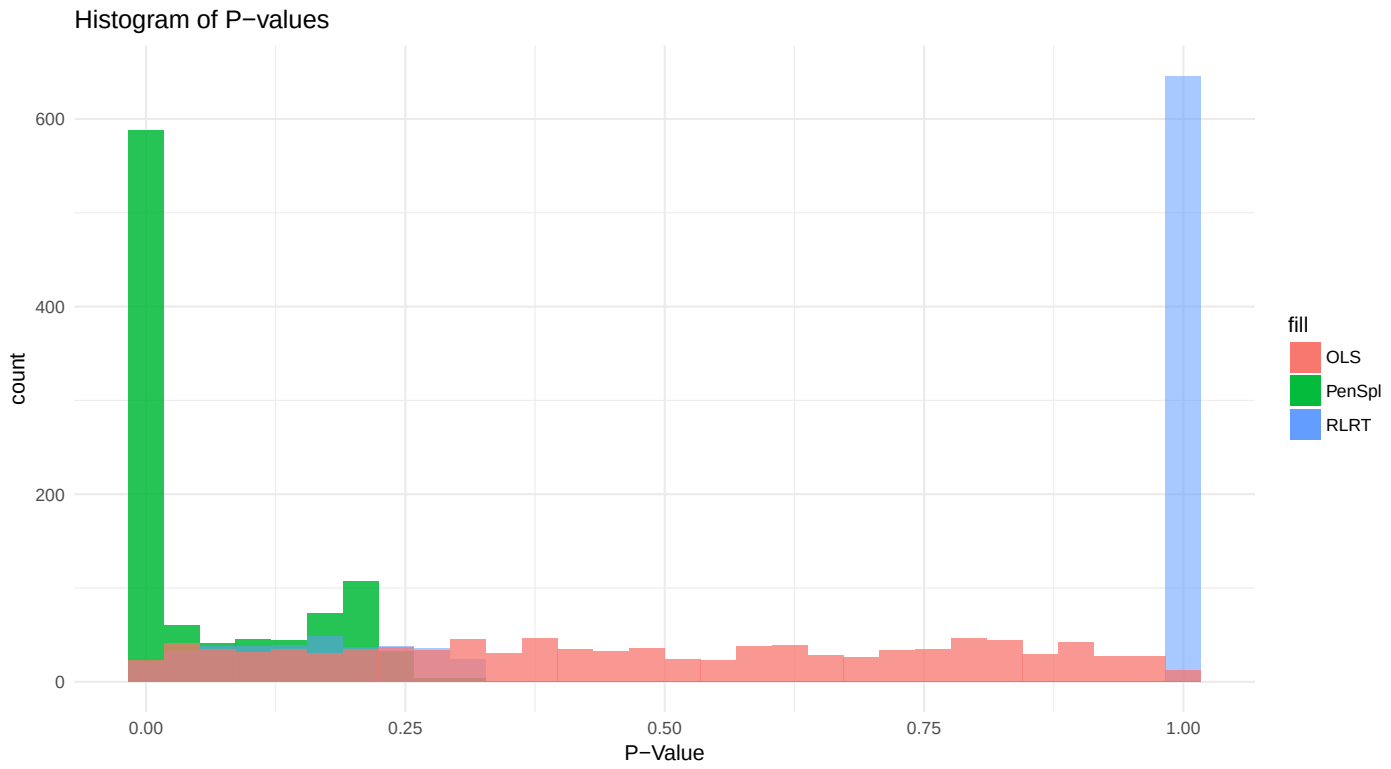
Unfortunetelly, I have not found more meaningful interpretation for those results (yet).

d)

Now 1000 simulations as above will be conducted and histograms of p-values will be plotted.

```
set.seed(1)

replications = 1000
pValues = matrix(nrow=replications, ncol=3)

for(i in 1:replications)
{
    y = x+rnorm(200)
    fitLine = gam(y~x)
    fitDfltPenSpl = gam(y~s(x))
    fitOLSspl = gam(y~s(x, k=5, sp=0))
    fitGAMM = gamm(y~s(x), method="REML")

    pValues[i,] = c(anova(fitLine, fitDfltPenSpl, test="F")$"Pr(>F)"[2],
                    anova(fitLine, fitOLSspl, test="F")$"Pr(>F)"[2],
                    exactRLRT(fitGAMM$lme)$p.value)
}

pValuesDF = data.frame(pValues)
names(pValuesDF) = c("PenSpl", "OLS", "RLRT")
```

```
ggplot(pValuesDF) +
    geom_histogram(aes(PenSpl, fill = "PenSpl"), alpha=.85) +
    geom_histogram(aes(RLRT, fill = "RLRT"), alpha=.55) +
    geom_histogram(aes(OLS, fill = "OLS"), alpha=.75) +
    labs(title="Histogram of P-values", x="P-Value") +
    theme_minimal()
```



Histogram of p-values of those test are completely diffrent, P-spline based test has p-values cumulated around zero, RLRT based test has big number of p-values equal to 1, but also significant number of p-values are around zero and below clasical sigificance level 0.05, OLS based taset is in beetwen them, having p-values laying eqivalently on $[0, 1]$.

e)

In last point the ratio beetwen number of rejected hypotheses to overall number of hypotheses will be calculated. In other words, the ratio of type I errors will be calculated.

```
ratio = colSums(pValues<.05)/nrow(pValues)
names(ratio) =c("PenSpl", "OLS", "RLRT")
ratio
```

```
PenSpl    OLS   RLRT
 0.647  0.061  0.055
```

It is visible that ratio of type I errors is close to significance level 0.05 only for OLS and RLRT based tests. For P-spline test the ratio of rejected null hypotheses is very large, although all assumpions for linear model are met.