Statistical Modelling 2020-2021: EXAM project

Question 1

1.A

By exploratory analysis, x6, x7 and x8 maybe the choices of non-linear components. By adding other covariates as linear components, we will add some of them by using AIC as the method to select a final semiparametric model. By adding 1, 2, 3 and 4 of other variables as linear components, we will get many models. I also change the degree of basis function to compare AIC and find the final model with the smallest AIC (1519.177).

1.B

The graphs of the components of the final model that are modeled in a nonlinear way are shown in Fig 1.

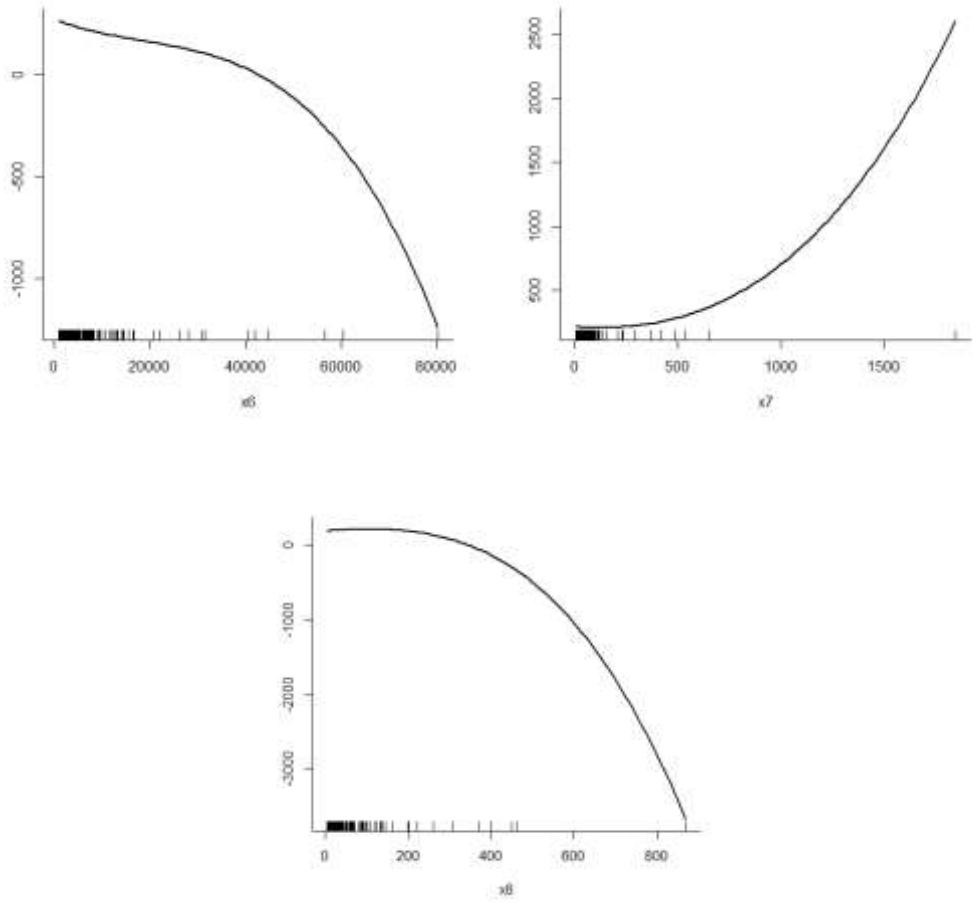

Fig 1. Graph of nonlinear components

1.C

The final model for this dataset is:

$$y_i = \beta_0 + \beta_1 x3_i + \beta_2 x4_i + \beta_3 x5_i + \beta_4 x9_i + f(x6_i) + f(x7_i) + f(x8_i) + \epsilon_i$$

$$Where\ \beta_0 = -73.69, \beta_1 = -2.299, \beta_2 = 0.3331, \beta_3 = 8.5080, \beta_4 = 1.9520$$

The appropriate distribution that has been used is gaussian distribution and the functions I have used are all truncated polynomial basis function with degree equals 3. The method for automatic smoothing parameter selection is maximum likelihood and the results are shown in Tab 1.

Tab 1. Summary for non-linear components

	df	spar	knots
$f(x6)$	3	3099000	36
$f(x7)$	3	43530	20
$f(x8)$	3	19080	19

1.D Representative R code for Question 1

```
fulldata = read.csv("HousePrices.txt", sep = " ", header = TRUE)
digitsum = function(x) sum(floor(x/10^(0:(nchar(x)-1)))%%10)

set.seed(studentnumber)
mysum = digitsum(studentnumber)

if((mysum %% 2) == 0) { # number is even
   rownumbers = sample(1:327,150,replace=F)
} else { # number is odd
   rownumbers = sample(309:585,150,replace=F)
}
mydata = fulldata[rownumbers,]
names(mydata) <- c('x1','x2','Municipality','y','x3','x4','x5','x6','x7','x8','x9')
attach(mydata)
###Q1. A
plot(mydata)
library(SemiPar)
#fit1
fit1 <-
spm(y~f(x6,basis="trunc.poly",degree=3)+f(x7,basis="trunc.poly",degree=3)+f(x8,basis="trunc.poly
",degree=3),family="gaussian",spar.method = 'ML')
AIC1 <- -2*fit1$fit$logLik+2*fit1$aux$df.fit
#adding 1 variable only give representative code
fit2 <-
spm(y~x3+f(x6,basis="trunc.poly",degree=3)+f(x7,basis="trunc.poly",degree=3)+f(x8,basis="trunc.
poly",degree=3),family="gaussian",spar.method = 'ML')
AIC2 <- -2*fit2$fit$logLik+2*fit2$aux$df.fit
#adding 2 variables
fit3 <-
spm(y~x4+x5+f(x6,basis="trunc.poly",degree=3)+f(x7,basis="trunc.poly",degree=3)+f(x8,basis="tr
unc.poly",degree=3),family="gaussian",spar.method = 'ML')
AIC3 <- -2*fit3$fit$logLik+2*fit3$aux$df.fit
```

```
#adding 3 variables
fit4 <-
spm(y~x3+x4+x9+f(x6,basis="trunc.poly",degree=3)+f(x7,basis="trunc.poly",degree=3)+f(x8,basis
="trunc.poly",degree=3),family="gaussian",spar.method = 'ML')
AIC4 <- -2*fit4$fit$logLik+2*fit4$aux$df.fit
#adding 4 variables
fit5 <-
spm(y~x3+x4+x5+x9+f(x6,basis="trunc.poly",degree=3)+f(x7,basis="trunc.poly",degree=3)+f(x8,b
asis="trunc.poly",degree=3),family="gaussian",spar.method = 'ML')
AIC5 <- -2*fit5$fit$logLik+2*fit5$aux$df.fit
#change the degree but not see any improvement
fit6 <-
spm(y~x3+x4+x5+x9+f(x6,basis="trunc.poly",degree=2)+f(x7,basis="trunc.poly",degree=2)+f(x8,b
asis="trunc.poly",degree=2),family="gaussian",spar.method = 'ML')
AIC6 <- -2*fit6$fit$logLik+2*fit6$aux$df.fit
c(AIC1,AIC2,AIC3,AIC4,AIC5,AIC6)
###Q1.B
plot(fit5,se=FALSE)
summary(fit5)
```

Question 2

2.A

A parametric additive model for these data is:

$$E(y_i) = \beta_0 + \beta_1 x6_i + +\beta_2 x6_i^2 + \beta_3 x9_i + \beta_4 x9_i^2$$

The null hypothesis of this model is:

$$H_0: E(y_i) = \beta_0 + \beta_1 x6_i + +\beta_2 x6_i^2 + \beta_3 x9_i + \beta_4 x9_i^2 \quad for\ each\ x$$

A nonparametric alternative hypothesis is,

$$H_\alpha: E(y_i) \neq \beta_0 + \beta_1 x6_i + +\beta_2 x6_i^2 + \beta_3 x9_i + \beta_4 x9_i^2 \quad for\ some\ x$$

For the construction of the order selection test statistic, I will use cosine functions $\psi_j(x) = \cos(\pi j x)$. The value of the test statistic is 13.789 and the corresponding $p - value$ is 0.0002. Because of the computation ability, an upper bound of $r = 16$ is used for the test. The test gives the evidence of lack of fit of the model, and we can reject H_0.

2.B Representative R code for Question 2

```
#Q2
par(mfrow=c(1,2))
plot(x6,y)
plot(x9,y)
fit_parAdd <- glm(y~x6+x9+I(x6^2)+I(x9^2))
summary(fit_parAdd)
par(mfrow=c(2,2))
```

```
plot(fit_parAdd)
# p-value formula
pvalue.Tos = function(Tos)
{ mlimit=100
1-exp(-sum((1-pchisq((1:mlimit)*Tos, 1:mlimit))/(1:mlimit))) }
q2x6 <- (x6-min(x6))/(max(x6)-min(x6))
q2x62 <- (x6^2-min(x6^2))/(max(x6^2)-min(x6^2))
q2x9 <- (x9-min(x9))/(max(x9)-min(x9))
q2x92 <- (x9^2-min(x9^2))/(max(x9^2)-min(x9^2))
m = 16
x = cbind(q2x6,q2x62,q2x9,q2x92)
Xcos = matrix(nrow=nrow(x),ncol=m)
f1 = function(a){if(a%%4==0){return(4)} else{return(a%%4)}    }
for (j in 1:m){ Xcos[,j] = cos(pi*j*x[,f1(j)]) }
X = cbind(x,Xcos)
LogLik = rep(NA,m)
for (j in 1: (m+4)){
   LogLik[j]= logLik(glm(y~ X[,1:j],control=list(maxit=5000)) )
}
T.OS = max(2*(LogLik[2:m]-LogLik[1])/(1:(m-1)))
pvalue.Tos(T.OS)
```

Question 3

3.A

As shown in Fig 2, by plotting such lines in different panels (one for each province), it is clearly that there are different intercepts for different provinces, as well as different slopes. We can build a certain mixed effect structure using $x2$ Province as the grouping variable. There is an effect of Province when regressing y on $x6$. Suggested form the Trellis plot, we can construct a mixed effect model with a specific random intercept and a specific random slope.

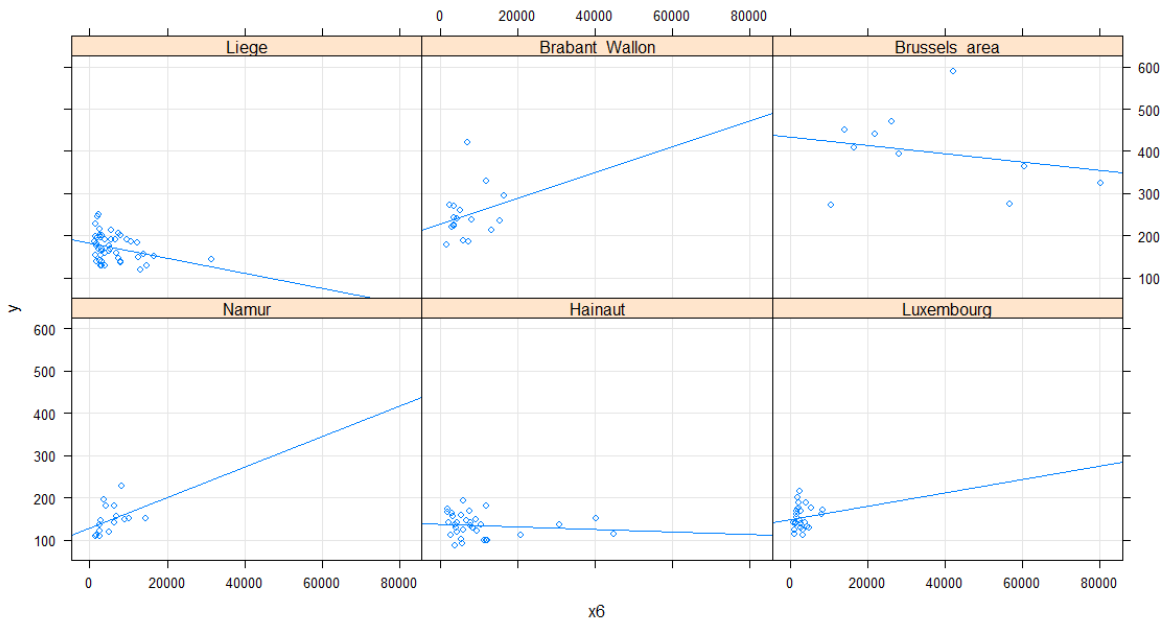

Fig 2. Y versus x6, per Province. Linear regression fits are added to each panel.

3.B

As suggested in 3.A, we can build a generalized linear mixed effect model. Based on the answer of question1, some covariates will be included in the model in a parametric way. The generalized linear mixed effect model of this dataset is,

$$y_{ij} = \beta_0 + \beta_1 x3_j + \beta_2 x4_j + \beta_3 x5_j + \beta_4 x6_j + \beta_5 x9_j + U_{i0} + U_{i1} x6_j + \varepsilon_{ij} \qquad \varepsilon_{ij} \sim (0, \sigma_\varepsilon^2)$$

The summary of the coefficients of fixed effects is shown in Tab 2.

Tab 2. Coefficients of fixed effects

Intercept	x3	x4	x5	x6	x9
25.04708	-0.22655	-0.01505	6.32735	-0.00192	2.47209

The summary of the coefficients of random effects is shown in Tab 3.

Tab 3. Coefficients of random effects

	Brabant_Wallon	Brussels_area	Hainaut	Liege	Luxembourg	Namur
Intercept	5.26150	182.09665	-53.17075	-33.39399	-49.77195	-51.02147
x6	0.00007	0.00065	-0.00014	-0.00023	-0.00018	-0.00017

The approximate covariance matrix of the fixed effects estimates is shown in Tab 4.

Tab 4. Covariance matrix of the fixed effects estimates.

	Intercept	x3	x4	x5	x6	x9
Intercept	1614.57824	0.17207	0.01084	-15.78120	-0.00185	1.92405
x3	0.17207	0.12456	0.00016	0.01847	-0.00010	-0.04378
x4	0.01084	0.00016	0.00003	-0.00008	0	-0.00013

x5	-15.78120	0.01847	-0.00008	0.58863	0.00018	-0.10721
x6	-0.00185	-0.00010	0	0.00018	0.00000	-0.00020
x9	1.92405	-0.04378	-0.00013	-0.10721	-0.00020	0.20130

To see whether the results support the suggestion from 3.A, we will construct 95% confidence intervals for the standard deviation of random effects as shown in Tab 5. The interval of $sd(intercept)$ is far from 0 so we should leave it in the model. The interval of $sd(x6)$ is close to 0 so we can say it is not important and we can leave it out of the model.

Tab 5. Approximate 95% confidence intervals

	lower	est.	upper
sd((Intercept))	46.15	83.637	151.57
sd(x6)	0.0000495	0.000384	0.00297
cor((Intercept),x6)	-0.969	0.7602	0.9994

3.C

```
#Q3.A
library(nlme)
library(MASS)
library(lattice)
par(mfrow=c(1,1))
xyplot(y~x6|x2, mydata,type=c('g','p','r'),index=function(x,y)coef(lm(y~x))[1])
#Q3.B
#penalized Quasi-likelihood
q3fit1 <- glmmPQL(y~+x3+x4+x5+x6+x9,random=~x6|x2,family = gaussian)
summary(q3fit1)
t <- summary(q3fit1)
round(t$coefficients$fixed,5)
round(t$coefficients$random$x2,5)
round(t$varFix,5)
intervals(q3fit1,which = 'var-cov')
```

Question 4

4.A

As shown in Tab 6, I construct a table containing the vector of estimated coefficients of the regression model and the alpha of elastic net estimator is 0.5.

Tab 6. Estimated coefficients of the regression model using four methods

	(1) maximum likelihood estimation	(2) Ridge regression	(3) Lasso estimation	(4) elastic net estimator alpha = 0.5
$Intercept$	226.6	98.13216	190.93995	88.50649
$x1Waals\_Gewest$	-201.3	-95.34917	-196.88264	-97.77816
$x2Brussels\_area$	NA	95.19833	0	98.20064
$x2Hainaut$	-59.08	-46.86060	-46.89745	-45.14993
$x2Liege$	-40.37	-26.78916	-25.69688	-23.80665
$x2Luxembourg$	-54.49	-39.28124	-39.80722	-37.54878
$x2Namur$	-54.45	-38.77113	-38.26769	-35.75357
$x3$	-0.2279	-0.26202	-0.11787	-0.12017
$x4$	-0.0309	-0.00999	-0.00796	-0.00766
$x5$	6.49	6.97494	7.29746	7.36842
$x6$	-0.00268	-0.00086	-0.00075	-0.00055
$x7$	0.05844	-0.01350	.	.
$x8$	0.2264	0.08927	.	.
$x9$	1.898	1.36116	1.48699	1.35295

4.B

To build this table, I find that the range of the number of industrial firms ($x8$) is $[3, 870]$ and the range of number of hotels and restaurants ($x7$) is $[6, 1853]$. To do the prediction of a municipality with a low number of industrial firms, I choose Cerfontaine (first row) with 19 industrial firms; to do the prediction of a municipality with a large number of hotels and restaurants, I choose Tubeke (third row) with 86 hotels and restaurants. The results are shown in Tab 7.

Tab 7. Predictions of Cerfontaine (Low number of industrial firms) and Tubeke (Large number of hotels and restaurants)

	(1) maximum likelihood estimation	(2) Ridge regression	(3) Lasso estimation	(4) elastic net estimator alpha = 0.5	PriceHouse
Cerfontaine	130.5106	133.7617	132.7582	133.4849	122.000
Tubeke	209.0606	199.0165	198.4413	196.8348	213.500

4.C

#Q4.A

```
library(glmnet)
library(ggplot2)
Y=mydata [,4]
X <- as.matrix(data.frame(model.matrix(~x1+x2+x3+x4+x5+x6+x7+x8+x9)))
X <- X[,-1]
mydata2 <- data.frame(cbind(X,Y))
#MLE
glmModel <- glm(Y~.,data=mydata2)
summary(glmModel)
#Ridge regression    alpha = 0
set.seed(819511)
ridge.cv_fit = cv.glmnet(X,Y, type.measure ="mse", alpha = 0, nfolds = 5)
s0 = ridge.cv_fit$lambda.min
round(coef(ridge.cv_fit, s = s0),5)
#Lasso regression alpha = 1
set.seed(819511)
lasso.cv_fit = cv.glmnet(X,Y, type.measure ="mse", alpha = 1, nfolds = 5)
s1 = lasso.cv_fit$lambda.min
round(coef(lasso.cv_fit, s = s1),5)
#elastic net alpha = 0.5
set.seed(819511)
elastic.cv_fit = cv.glmnet(X,Y, type.measure ="mse", alpha = 0.5, nfolds = 5)
s2 = elastic.cv_fit$lambda.min
round(coef(elastic.cv_fit, s = s2),5)
##Q4.B
summary(mydata2$x8)
summary(mydata2$x7)
#prediction of low industrial firms:Cerfontaine x8=19
#prediction of large industrial firms:Tubeke x7=86
newX<- X[c(1,3),]
predict(glmModel,mydata2[c(1,3),])
predict(ridge.cv_fit,newX,s=s0)
predict(lasso.cv_fit,newX,s=s1)
predict(elastic.cv_fit,newX,s=s2)
```