Semiparametric regression - Homework 8

Klaudia Weigel

1 Exercise 1

We consider the `WarsawApts` dataset containing information about house prices in Warsaw.

```r
library(HRW); library(mgcv); library(tidyverse)
data(WarsawApts)
```

1.1 (a)

We will first fit a semiparametric model associating “construction.date” and “areaPerMzloty”. We will use cubic spline basis functions and “REML” as a smoothing parameter selection criterion.

```r
fitGAMcr <- gam(areaPerMzloty~s(construction.date, bs = 'cr', k = 30),
                data = WarsawApts, method = "REML")
```

1.2 (b)

In this point we will introduce “district” to the model as an additive factor. Our model will have the following form

$$\begin{aligned}\texttt{areaPerMzloty}_i &= \beta_0 + \beta_1 I(\texttt{district}_i == \texttt{Srodmiescie}) + \beta_2 I(\texttt{district}_i == \texttt{Wola}) \\ &+ \beta_3 I(\texttt{district}_i == \texttt{Zoliborz}) + f(\texttt{construction.date}_i) + \epsilon_i.\end{aligned}$$

```r
fitGAMcrDist <- gam(areaPerMzloty~s(construction.date, bs = 'cr', k = 30) +
                      factor(district),
                    data = WarsawApts, method = "REML")
summary(fitGAMcrDist)
```

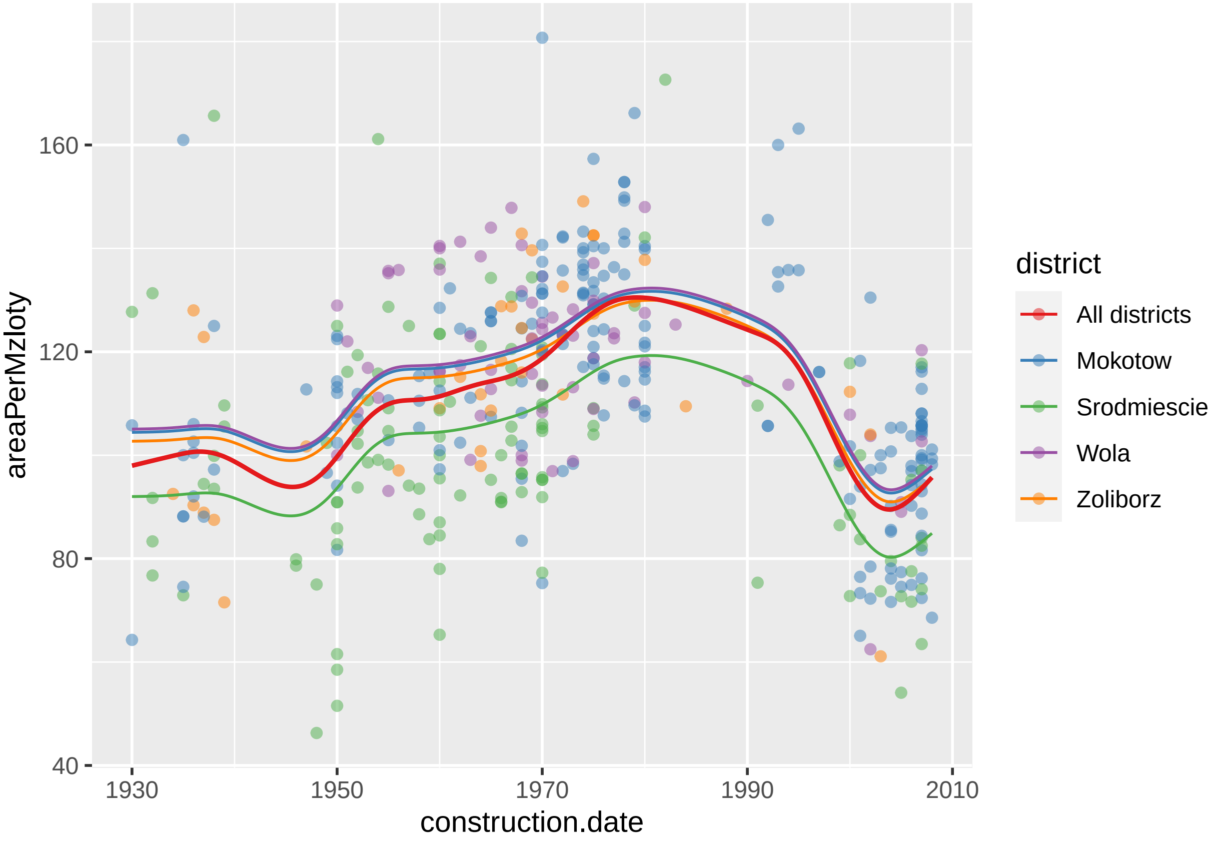
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## areaPerMzloty ~ s(construction.date, bs = "cr", k = 30) + factor(district)
##
## Parametric coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  113.866      1.286  88.558  < 2e-16 ***
## factor(district)Srodmiescie  -12.419      2.158  -5.755 1.74e-08 ***
## factor(district)Wola           0.626      2.555   0.245    0.807
## factor(district)Zoliborz      -1.717      3.231  -0.531    0.595
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                        edf Ref.df     F p-value
## s(construction.date) 9.34  11.47 18.17  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.403   Deviance explained = 42.1%
## -REML = 1743.2  Scale est. = 292.32    n = 409
```

We can observe that there is no significant difference between Mokotow, Wola and Zoliborz.

1.3 (c)

We will now plot the fitted lines. First let us see fits from a) and b) on one plot.

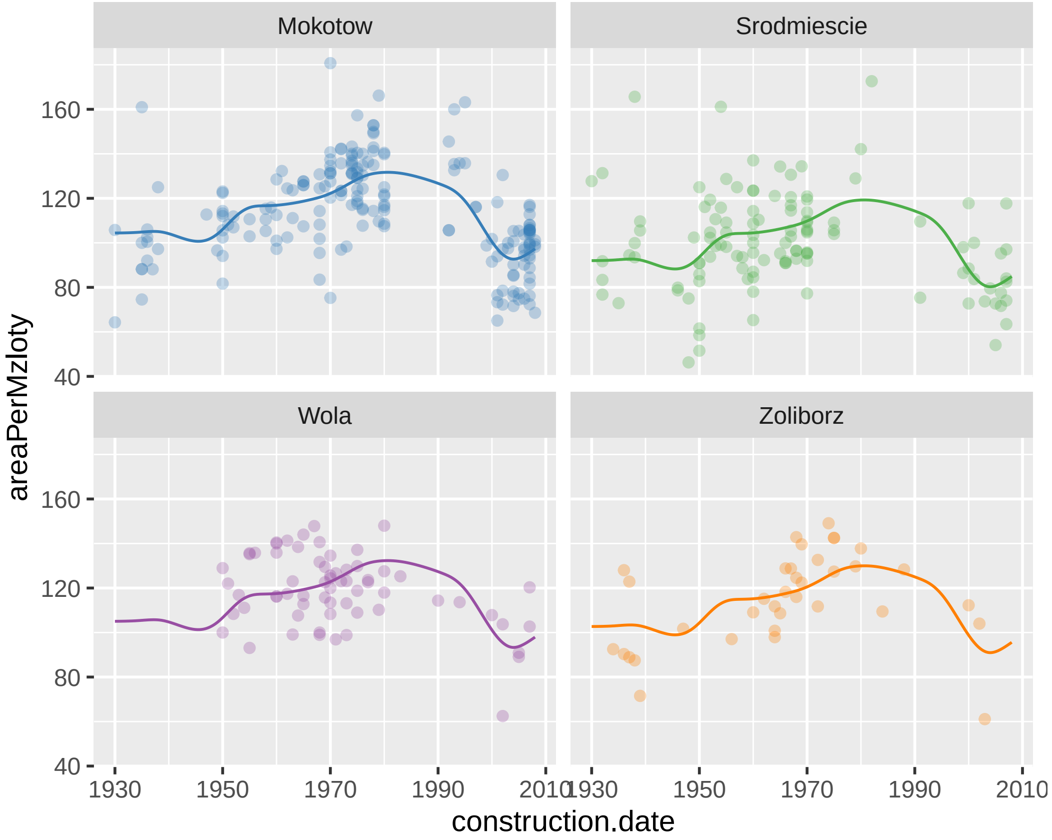
```
xg <- seq(min(WarsawApts$construction.date), max(WarsawApts$construction.date),
          length = 1001)
fHatgGAMcr <- predict(fitGAMcr, newdata = data.frame(construction.date = xg))
newdat <- expand.grid(construction.date = xg,
                      district = levels(as.factor(WarsawApts$district)))
fHatgGAMcrDist <- predict(fitGAMcrDist, newdata = newdat)

data_plot <- data.frame(newdat, fHatgGAMcrDist)
colnames(data_plot) <- c('xg', 'district', 'fit')
df1 <- data.frame(xg, rep("All districts", length(xg)), fHatgGAMcr)
colnames(df1) <- c('xg', 'district', 'fit')

rbind(data_plot, df1) %>%
  mutate(highlight = district == "All districts") %>%
  ggplot() +
  geom_point(data = WarsawApts, aes(x=construction.date,
                                    y=areaPerMzloty,
                                    color = district), alpha = 0.5) +
  geom_line(aes(x = xg, y = fit, color = district, size = highlight)) +
  scale_color_brewer(palette="Set1") +
  guides(size = FALSE) +
  scale_size_manual(values = c("TRUE" = 0.8, "FALSE" = 0.5))
```



We see that the fitted lines are almost the same for districts Wola, Mokotow and Zoliborz. The area is significantly lower for Srodmiescie district, when compared Mokotow. The fitted line estimated from the model in point (a) is closer to the lines for Wola, Mokotow and Zoliborz.

Let us also see the fitted lines from the model from point (b) with respect to subset of the data corresponding to a particular district:

```
cbind(newdat, fHatgGAMcrDist) %>%
  ggplot() +
  geom_point(data = WarsawApts, aes(x=construction.date,
                                    y=areaPerMzloty,
                                    color=district),
             alpha = 0.3) +
  geom_line(aes(x = construction.date, y = fHatgGAMcrDist, color = district)) +
  scale_color_manual(values = my_col) +
  guides(color="none") +
  facet_wrap(~district)
```



1.4 (d)

Overall there is little difference in `areaPerMzloty` as a function of `construction.date` for districts Mokotow, Wola, Zoliborz. The only district that is significantly different is Srodmiescie, where the mean value of area to price ratio is smaller by about 12 units when compared to Mokotow.

2 Exercise 2

In this exercise we will use data `retirePlan`. Our goal is to choose the "best additive model" to predict the "log(contrib)", i.e. the natural log of the contributions to the retirement plan.

```
retirePlan <- read.table("retirePlan.txt", header = T)
head(retirePlan)
```

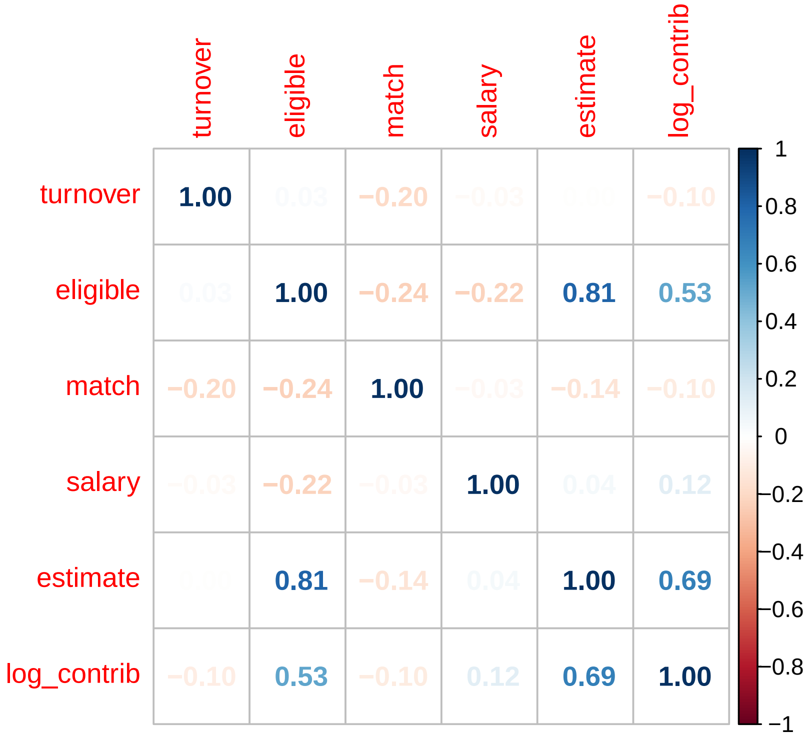
```
##   contrib group turnover eligible vest failsafe match  salary estimate susan
## 1   36675     1       14       69    1        0    25 26296.3    75432     0
## 2   63733     0       10       33    1        0    50 14133.7    50000     0
## 3   25560     1        8       21    1        0    50 24000.0    45000     0
## 4  177970     0       10       67    1        0    25 36833.3   235000     0
## 5   86873      1        10        47     1         0     50 41140.1    146965      0
## 6   39051      0        10        12     1         0      0 66463.8     95000      0
```

Variables “susan”, “group”, “vest”, “failsafe” are categorical taking values 0 or 1, so we will treat them as factors.

```
retirePlan_cat <- retirePlan
cols <- c("failsafe", "susan", "group", "vest")
retirePlan_cat[,cols] <- lapply(retirePlan_cat[,cols], factor)
```

We might want to check correlations between numerical variables.

```
df <- retirePlan %>%
  mutate(log_contrib = log(contrib))
corrplot::corrplot(cor(df[,!names(df) %in% c(cols, "contrib")]), method = "number")
```



We see that there is a strong correlation between “log(contrib)” and “estimate”. “Estimate” is strongly correlated with “eligible”, so it would be reasonable to keep one of these variables in the model.

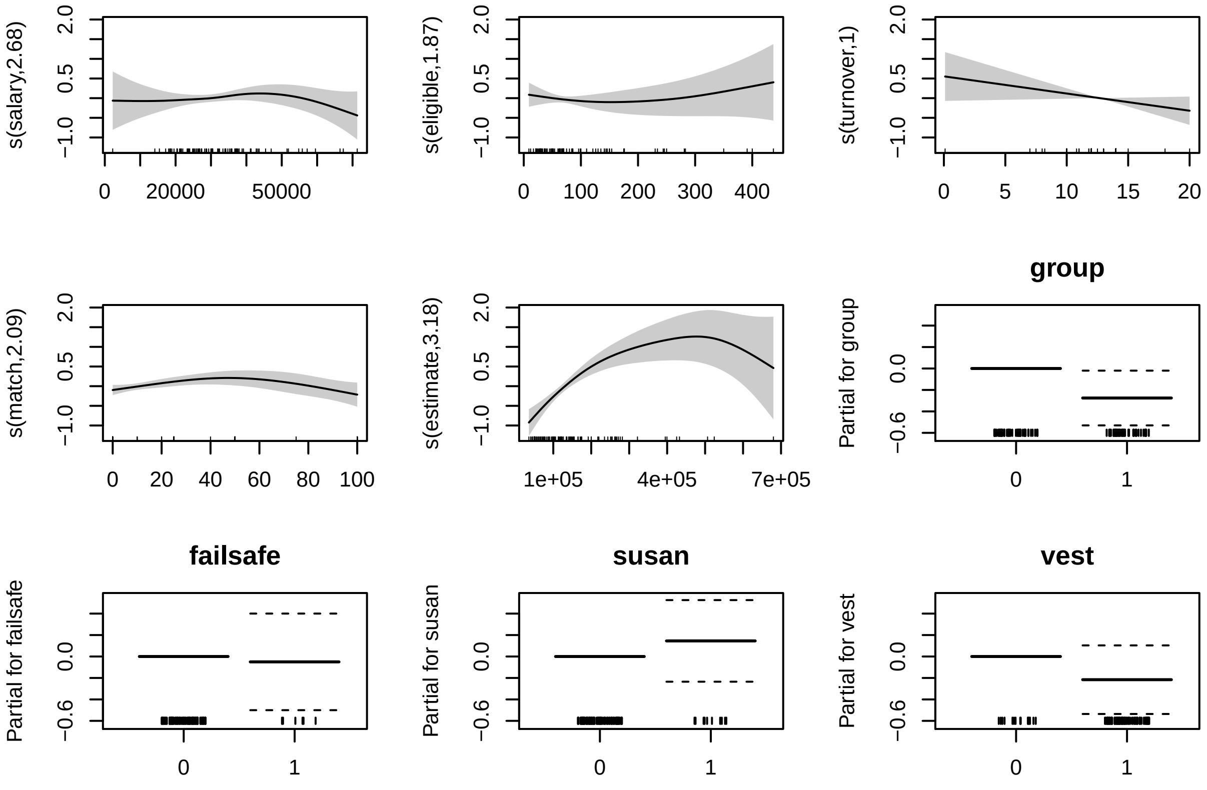
2.1 (a)

First we will try a univariate approach where we where each variable is assessed via its contribution and only the variables with significant contributions are used in the final model with an appropriate functional form.

```
m_a1 <- gam(log(contrib) ~ group +  s(salary) + s(eligible) +
              failsafe + susan + s(turnover) + s(eligible) +
              vest + s(match, k=5) + s(estimate), data = retirePlan_cat)
summary(m_a1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(contrib) ~ group + s(salary) + s(eligible) + failsafe + susan +
##     s(turnover) + s(eligible) + vest + s(match, k = 5) + s(estimate)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.34578    0.16269  69.737   <2e-16 ***
## group1      -0.27528    0.12754  -2.158    0.034 *
## failsafe1   -0.04991    0.22515  -0.222    0.825
## susan1       0.14550    0.19024   0.765    0.447
## vest1       -0.21609    0.15985  -1.352    0.180
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df     F  p-value
## s(salary)   2.684  3.405 0.902   0.3968
## s(eligible) 1.870  2.331 0.906   0.4506
## s(turnover) 1.000  1.000 3.154   0.0797 .
## s(match)    2.087  2.448 2.767   0.0440 *
## s(estimate) 3.179  3.757 9.420 7.42e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.63   Deviance explained =   69%
## GCV = 0.30754  Scale est. = 0.25465   n = 92
```

```
par(mar = c(3,4,3,2))
plot(m_a1, pages = 1, scheme = 1, seWithMean = FALSE, all.terms = TRUE, shade = TRUE)
```



We see that the only significant variables are “group”, “estimate”, “turnover” and “match”. We also observe that “turnover” is a linear effect (edf = 1).

```
m_a <- gam(log(contrib) ~ group + s(estimate) + turnover + s(match, k=5), data = retirePlan_cat)
summary(m_a)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(contrib) ~ group + s(estimate) + turnover + s(match, k = 5)
```

```
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.76747    0.30584  38.476   <2e-16 ***
## group1      -0.29063    0.11132  -2.611   0.0107 *
## turnover    -0.04590    0.02374  -1.934   0.0565 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df      F p-value
## s(estimate) 2.818  3.502 36.046  <2e-16 ***
## s(match)    2.027  2.382  1.807   0.124
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.612   Deviance explained = 64.1%
## GCV = 0.29193  Scale est. = 0.26704   n = 92
```

In this model "match" turns out to be insignificant, we might want remove, getting:

```
m_a <- gam(log(contrib) ~ group + s(estimate) + turnover, data = retirePlan_cat)
summary(m_a)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(contrib) ~ group + s(estimate) + turnover
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.74075    0.30808  38.110   <2e-16 ***
## group1      -0.29596    0.11268  -2.627   0.0102 *
## turnover    -0.04360    0.02401  -1.816   0.0729 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df    F p-value
## s(estimate) 2.674  3.334 37.3  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.591   Deviance explained = 61.2%
## GCV = 0.29949  Scale est. = 0.28102   n = 92
```

2.2 (b)

We will try and select the best model using `Gam.select` function from the `gam` library.

```
library(gam)
fitInitial <- gam::gam(log(contrib) ~ group + turnover +
                         eligible + vest + failsafe +
                         match + salary + estimate + susan, data = retirePlan_cat)
stepFit <- step.Gam(fitInitial,
                    scope = list("salary" = ~ 1 + salary + s(salary,2),
                                 "failsafe" = ~ 1 + failsafe,
                                 "susan" = ~ 1 + susan,
                                 "group" = ~ 1 + group,
                                 "eligible" = ~ 1 + eligible + s(eligible,2),
                                 "estimate" = ~ 1 + estimate + s(estimate,2),
                                 "vest" = ~ 1 + vest,
                                 "match" = ~ 1 + match + s(match,2)),
                    trace = FALSE)
print(names(stepFit$"model")[-1])
```

```
## [1] "turnover"       "s(salary, 2)"   "group"          "s(estimate, 2)"
## [5] "s(match, 2)"
```

We can now fit a model using the `mgcv` package with the chosen predictor variables.

```
detach(package:gam)
m_b <- gam(log(contrib) ~ turnover + s(salary) + group + s(estimate) + s(match, k=5),
           data = retirePlan_cat)
summary(m_b)
```

```
## 
## Family: gaussian 
## Link function: identity 
## 
## Formula:
## log(contrib) ~ turnover + s(salary) + group + s(estimate) + s(match, 
##     k = 5)
## 
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 11.73508    0.30937  37.932   <2e-16 ***
## turnover    -0.04372    0.02407  -1.817   0.0730 .  
## group1      -0.28089    0.11784  -2.384   0.0195 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Approximate significance of smooth terms:
##               edf Ref.df      F p-value    
## s(salary)   2.767  3.508  1.108  0.3017    
## s(estimate) 2.652  3.292 35.842  <2e-16 ***
## s(match)    2.036  2.391  2.080  0.0922 .  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## R-sq.(adj) =  0.625   Deviance explained = 66.4%
## GCV = 0.29085  Scale est. = 0.2578    n = 92
```

Even though the “salary” variable was chosen it is not significant.

2.3 (c)

We will now try to select a model via `gam.selection`. Because `select = TRUE` only penalizes smooth parameters we will define the factor variables as random effects (bs = “re”).

```
m_c <- gam(log(contrib) ~ s(group, bs = "re") +  s(salary) + s(eligible) +
              s(failsafe, bs = "re") + s(susan, bs = "re") + s(turnover) +
              s(vest, bs = "re")  + s(match, k=5) + s(estimate),
           data = retirePlan_cat, select = TRUE)

summary(m_c)
```

```
## 
## Family: gaussian 
## Link function: identity 
## 
## Formula:
## log(contrib) ~ s(group, bs = "re") + s(salary) + s(eligible) + 
##     s(failsafe, bs = "re") + s(susan, bs = "re") + s(turnover) + 
##     s(vest, bs = "re") + s(match, k = 5) + s(estimate)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.0934     0.1864   59.52   <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(group)    8.181e-01      1 4.913  0.0137 *
## s(salary)   2.996e+00      9 0.584  0.1981
## s(eligible) 1.199e+00      9 0.159  0.2849
## s(failsafe) 6.274e-11      2 0.000  0.9924
## s(susan)    1.139e-10      2 0.000  0.3500
## s(turnover) 6.582e-01      9 0.234  0.0741 .
## s(vest)     7.065e-01      1 1.420  0.1484
## s(match)    1.788e+00      4 2.169  0.0138 *
## s(estimate) 2.781e+00      9 5.812  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.636   Deviance explained =   68%
## GCV = 0.28752  Scale est. = 0.25018   n = 92
```

```
AIC(m_a, m_b, m_c)
```

```
##             df      AIC
## m_a  6.674399 151.7982
## m_b 11.455465 148.1830
## m_c 12.947284 146.7097
```

The lowest AIC score has been obtained for the model in part c).

3 Exercise 3

We again consider the `WarsawApts` dataset.

3.1 (a)

First we will fit a model with "construction.date" as a predictor and one curve for Srodmiescie and another curve for the other three districts, i.e. a binary-by-curve interaction model.

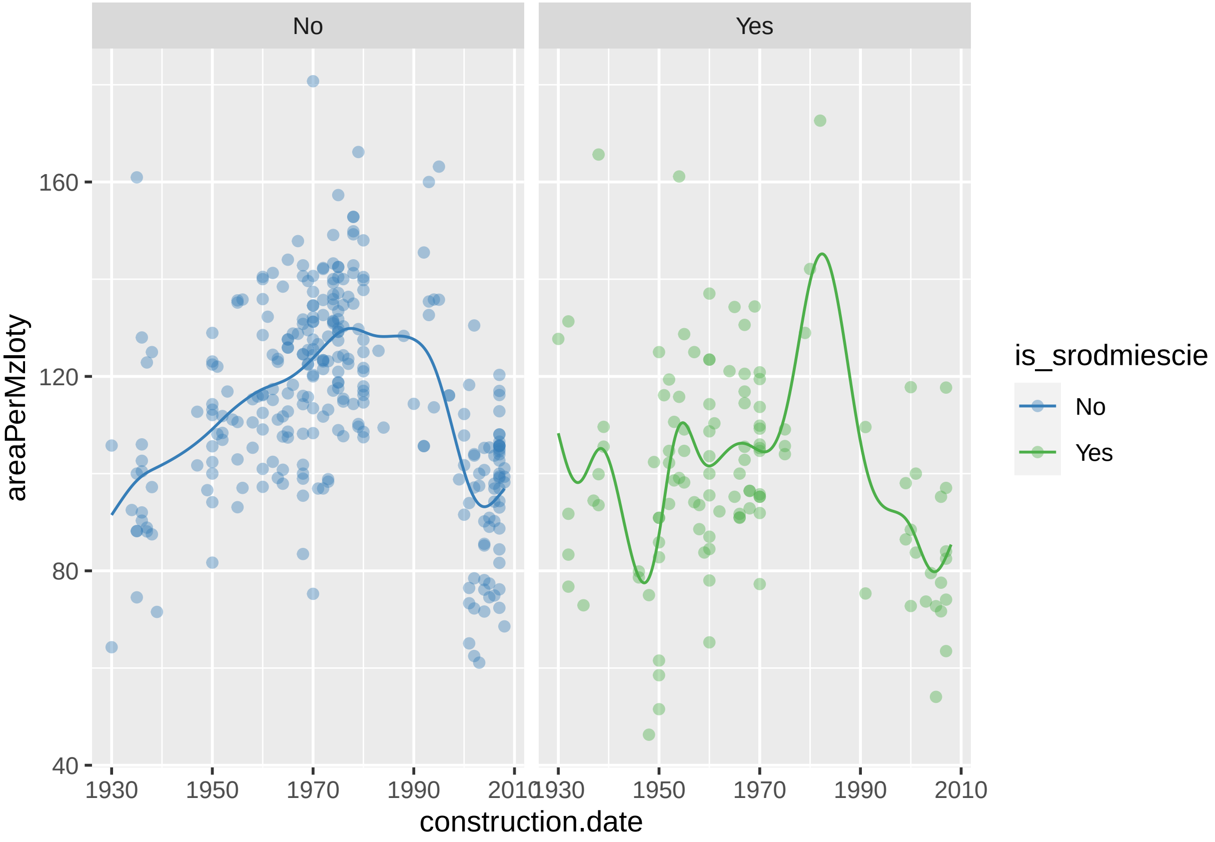
```
WarsawAptsSrod <- WarsawApts %>%
  mutate(is_srodmiescie = case_when(district == "Srodmiescie" ~"Yes",
                                    TRUE ~ "No"))
WarsawAptsSrod$is_srodmiescie <- factor(WarsawAptsSrod$is_srodmiescie)

fitGAMWarsaw_a <- gam(areaPerMzloty ~ is_srodmiescie +
                  + s(construction.date,by = is_srodmiescie, k = 25),
                data = WarsawAptsSrod,method = "REML")
summary(fitGAMWarsaw_a)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## areaPerMzloty ~ is_srodmiescie + +s(construction.date, by = is_srodmiescie,
##     k = 25)
##
## Parametric coefficients:
```

```
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           113.799      0.980 116.123  < 2e-16 ***
## is_srodmiescieYes     -12.260      2.107  -5.818 1.25e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                          edf Ref.df      F p-value
## s(construction.date):is_srodmiescieNo   8.269  10.18 19.486  <2e-16 ***
## s(construction.date):is_srodmiescieYes 11.235  13.46  5.128  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.449   Deviance explained = 47.7%
## -REML = 1747.1  Scale est. = 269.65    n = 409
```

We see that all the terms are significant. Let us also see the fitted lines.

```
newdat_3a <- expand.grid(construction.date = xg, is_srodmiescie = levels(WarsawAptsSrod$is_srodmiescie))
fHatGAMWarsaw_a <- predict(fitGAMWarsaw_a, newdata = newdat_3a)
data_plot <- data.frame(newdat_3a, fHatGAMWarsaw_a)
colnames(data_plot) <- c('xg', 'is_srodmiescie', 'fit')

data_plot %>%
  ggplot() +
  geom_point(data = WarsawAptsSrod, aes(x=construction.date,
                                        y=areaPerMzloty,
                                        color = is_srodmiescie),
             alpha = 0.4) +
  geom_line(aes(x = xg, y = fit, color = is_srodmiescie)) +
  scale_color_manual(values = my_col) +
  facet_wrap(~is_srodmiescie)
```



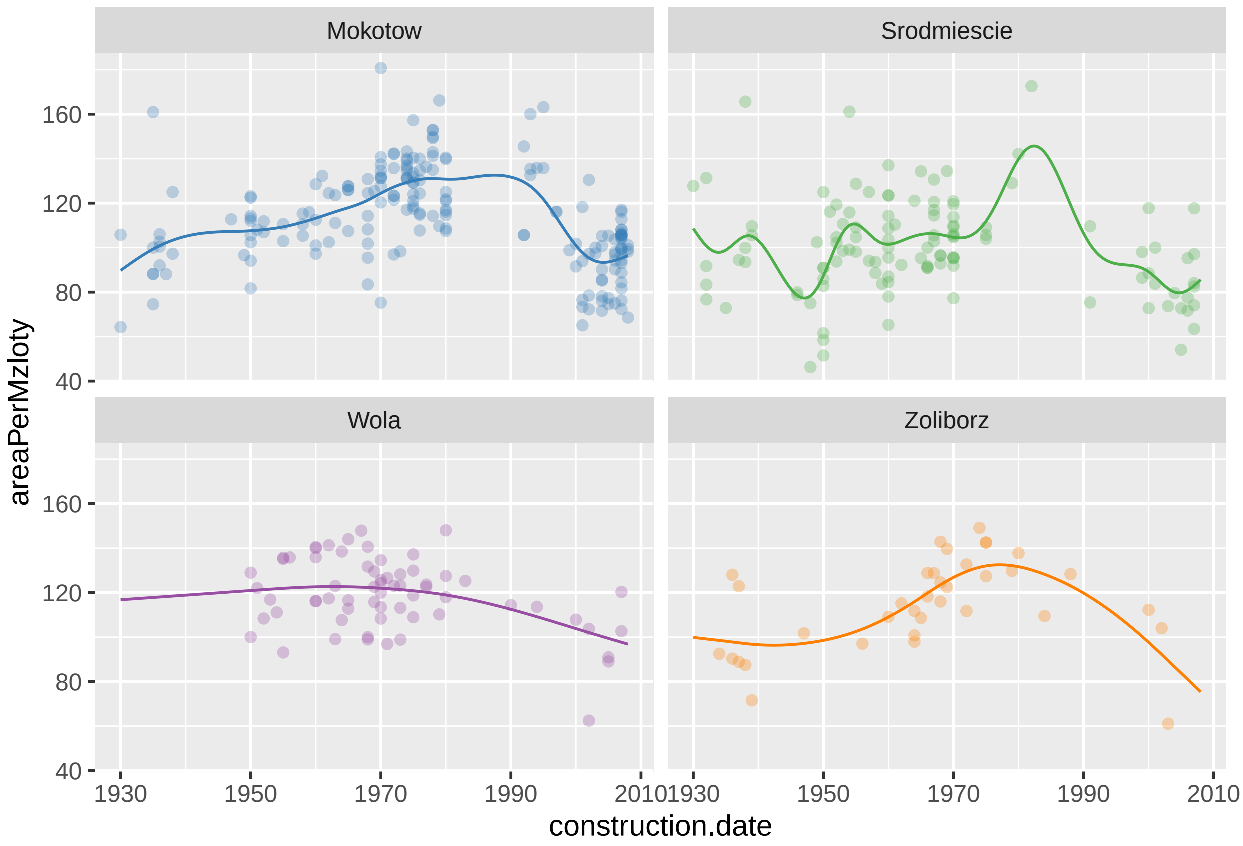
3.2 (b)

Let us now fit a model with separate lines for each district

```
fitGAMWarsaw_b <- gam(areaPerMzloty ~ as.factor(district) +
                        + s(construction.date,by = as.factor(district), k = 25),
                      data = WarsawApts,method = "REML")
summary(fitGAMWarsaw_b)
```

```
## 
## Family: gaussian 
## Link function: identity 
## 
## Formula:
## areaPerMzloty ~ as.factor(district) + +s(construction.date, by = as.factor(district), 
##     k = 25)
## 
## Parametric coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                       113.491      1.275  89.036  < 2e-16 ***
## as.factor(district)Srodmiescie    -11.968      2.250  -5.320 1.78e-07 ***
## as.factor(district)Wola             2.527      2.811   0.899    0.369    
## as.factor(district)Zoliborz        -3.869      3.710  -1.043    0.298    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Approximate significance of smooth terms:
##                                                      edf Ref.df      F
## s(construction.date):as.factor(district)Mokotow      7.677  9.366 17.473
## s(construction.date):as.factor(district)Srodmiescie 11.358 13.592  5.233
## s(construction.date):as.factor(district)Wola         2.072  2.542  4.613
## s(construction.date):as.factor(district)Zoliborz     3.616  4.442  6.362
##                                                    p-value    
## s(construction.date):as.factor(district)Mokotow      < 2e-16 ***
## s(construction.date):as.factor(district)Srodmiescie  < 2e-16 ***
## s(construction.date):as.factor(district)Wola         0.00498 ** 
## s(construction.date):as.factor(district)Zoliborz    3.29e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## R-sq.(adj) =  0.458   Deviance explained = 49.5%
## -REML = 1737.5  Scale est. = 265.47    n = 409
```

```
fHatGAMWarsaw_b <- predict(fitGAMWarsaw_b, newdata = newdat)
data_plot <- data.frame(newdat, fHatGAMWarsaw_b)
colnames(data_plot) <- c('xg', 'district', 'fit')

data_plot %>%
  ggplot() +
  geom_point(data = WarsawApts, aes(x=construction.date,
                                    y=areaPerMzloty,
                                    color = district),
             alpha = 0.3) +
  geom_line(aes(x = xg, y = fit, color = district)) +
  scale_color_manual(values = my_col) +
  facet_wrap(~district) +
  guides(color="none")
```



3.3 (c)

We will test models with th F-test.

```
anova(fitGAMWarsaw_a, fitGAMWarsaw_b, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: areaPerMzloty ~ is_srodmiescie + +s(construction.date, by = is_srodmiescie,
##     k = 25)
## Model 2: areaPerMzloty ~ as.factor(district) + +s(construction.date, by = as.factor(district),
##     k = 25)
##   Resid. Df Resid. Dev     Df Deviance      F Pr(>F)
## 1    379.24     104490
## 2    370.12     100952 9.1171   3537.4 1.4616 0.1594
```

```
AIC(fitGAMWarsaw_a, fitGAMWarsaw_b)
```

```
##                      df      AIC
## fitGAMWarsaw_a 26.63377 3481.100
## fitGAMWarsaw_b 34.66008 3483.066
```

According to the F-test and the AIC score we do not have sufficient evidence to prefer the model from part b).