Solution to Series 9

1. **a)** Read in the data:

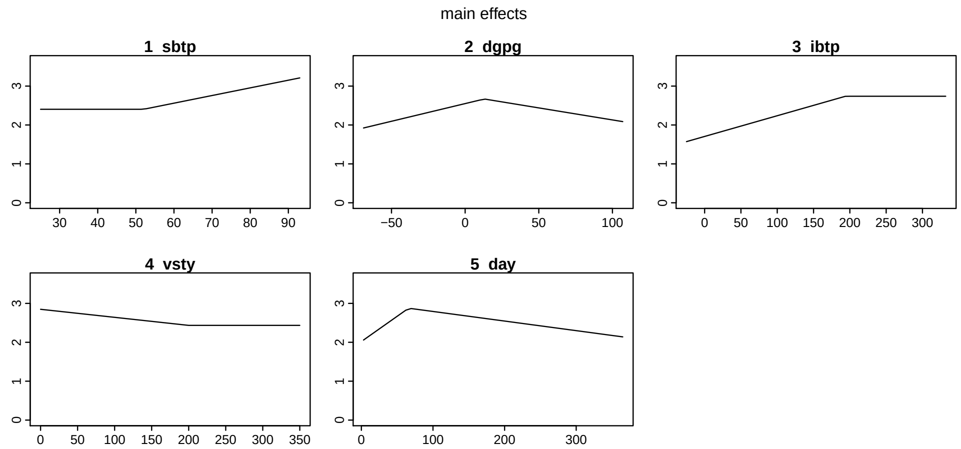
```
> ## load and transform the data:
> data(ozone, package = "gss")
> d.ozone <- subset(transform(ozone, logupo3 = log(upo3)), select = -upo3)
> d.ozone.e <- d.ozone[-which.max(d.ozone[,"wdsp"]),]
> d.ozone.es <- d.ozone.e
> d.ozone.es[,-10] <- scale(d.ozone.e[,-10])
```

b) We fit the MARS model with maximal interaction degree 2.

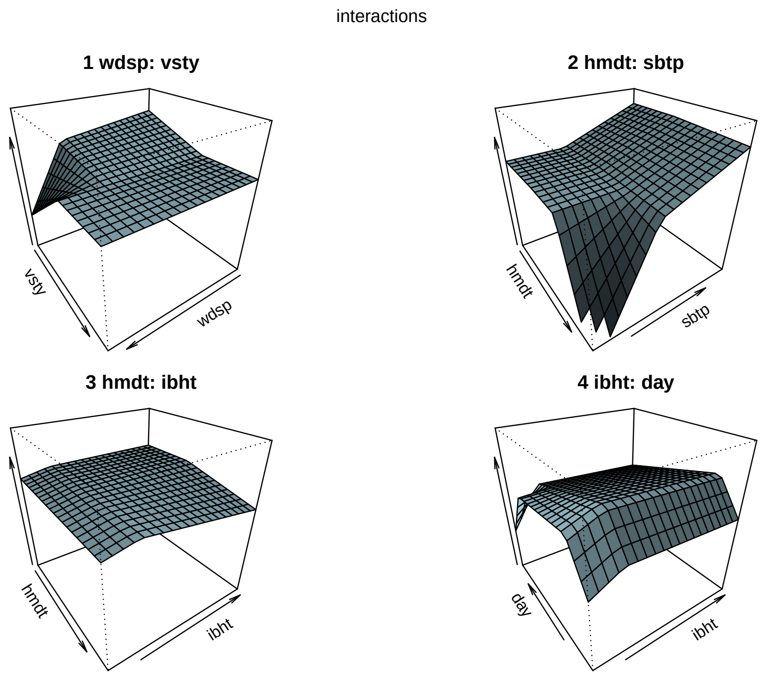
```
> library(earth) ## for MARS
> mars.fit2 <- earth(formula = logupo3 ~ .,data = d.ozone.e, degree = 2 )
> summary(mars.fit2,digits=3)
Call: earth(formula=logupo3~., data=d.ozone.e, degree=2)

                          coefficients
(Intercept)                     2.7570
h(sbtp-52)                      0.0197
h(13-dgpg)                     -0.0091
h(dgpg-13)                     -0.0062
h(194-ibtp)                    -0.0053
h(200-vsty)                     0.0014
h(66-day)                      -0.0130
h(day-66)                      -0.0025
h(8-wdsp) * h(200-vsty)         0.0002
h(wdsp-8) * h(200-vsty)        -0.0029
h(hmdt-71) * h(52-sbtp)        -0.0051
h(hmdt-51) * h(ibht-1049)       0.0000
h(1049-ibht) * h(day-286)       0.0000
h(1049-ibht) * h(286-day)       0.0000

Selected 14 of 21 terms, and 8 of 9 predictors
Termination condition: Reached nk 21
Importance: sbtp, ibtp, day, dgpg, vsty, ibht, hmdt, ...
Number of terms at each degree of interaction: 1 7 6
GCV 0.107    RSS 28.5    GRSq 0.809    RSq 0.845
> ## Plot the main effects
> plotmo(mars.fit2, degree2=FALSE,caption="main effects")
 grid:    vdht wdsp hmdt sbtp ibht dgpg ibtp vsty day
          5760    5   64   62 2109   24  168  120 178
```

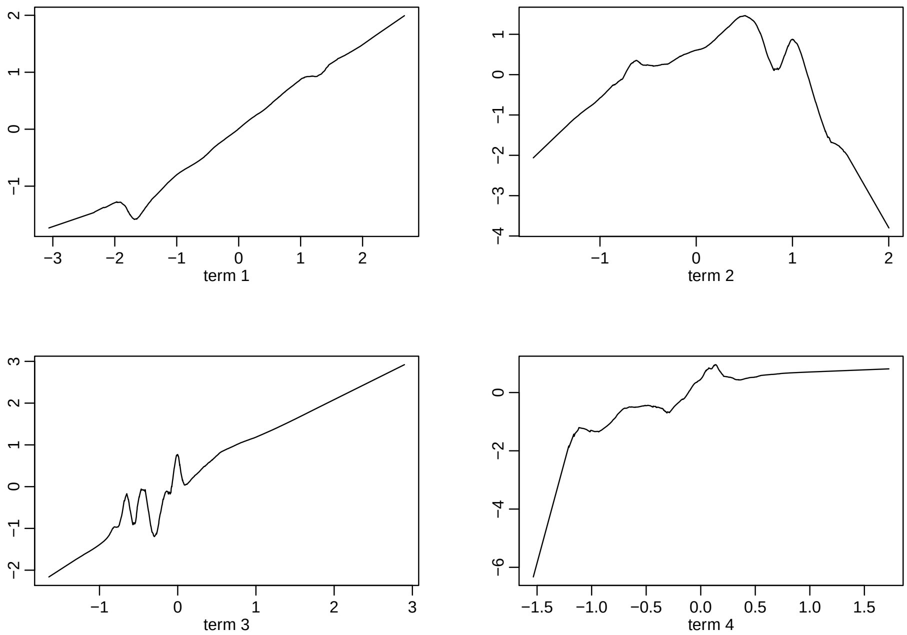



```
> ## Plot the interaction
> plotmo(mars.fit2, degree1=FALSE,caption="interactions")
```



c) Basic PPR fit and plot:

```
> ppr.fit <- ppr(logupo3~., nterms=4, data=d.ozone.es)
> sfsmisc::mult.fig(4)
> plot(ppr.fit, type='l')
```



The fitted lines of the ridge functions are too wiggly, therefore the model is not optimal. A better model would have smoother fitted lines of the ridge functions.

d) The following R -code calculates the CV-error for `nterms`$\in \{3, 4, 5\}$ and all three smoothers `sm.method` $\in$ {`"supsmu"`, `"spline"`, `"gcvspline"`}. `df` $\in \{5, 6, 7\}$ are used for `spline`. `max.terms` is always set equal to `nterms` + 3 .

```
> ## load the cv function:
> source("http://stat.ethz.ch/education/semesters/ss2014/CompStat/Exercises/cv-fun.R")
> ## number of terms
> N<- 3:5
> ## number of degrees of freedom
> DF<- 5:7
> ## Initialize cv scores
> cv.sm <- numeric(length = length(N))
> cv.gcv <- numeric(length = length(N))
> cv.ss <- matrix(0, nrow = length(N), ncol = length(DF),
                 dimnames=list(paste('df',DF), paste('nterms',N)))
> for(i in seq_along(N)){
   n <- N[i]
   cv.sm[i] <- cv(fitfn=ppr, nterms = n, max.terms = n + 3)
   cv.gcv[i] <- cv(ppr, sm.method = "gcvspline", nterms = n, max.terms = n + 3)
   for(j in seq_along(DF)){
     df <- DF[j]
     cv.ss[i, j] <- cv(ppr, sm.method = "spline", df = df, nterms = n, max.terms = n + 3)
   }
 }
```

e) Fit MARS models and print their cv:

```
> mars.1 <- cv(earth, degree = 1)
> mars.2 <- cv(earth, degree = 2)
> mars.3 <- cv(earth, degree = 3)
```

Compare with PPR cv:

```
> ##print the cv scores of the PPR fits
> cv.sm
[1] 53.05948 54.07697 53.92102
> cv.gcv
[1] 53.17953 54.75474 58.35856
```

```
> cv.ss
     nterms 3 nterms 4 nterms 5
df 5 40.23206 38.70816 43.42077
df 6 38.76893 35.00961 43.22224
df 7 38.38503 36.25633 44.19653
> ##print the cv scores of the MARS fits
> c(cv.mars.1=mars.1,cv.mars.2=mars.2  ,cv.mars.3=mars.3)
cv.mars.1 cv.mars.2 cv.mars.3
 40.06201  43.95574  44.21432
```

The best model in this case uses `sm.method = "spline"`, `nterms` = 4 and `df` = 6. Please note that there are other options to "tune" `supsmu` and `gcvspline` which should also be considered to do a fair comparison (see `bass, span, gcvpen`). MARS with interaction degree of 1,2 and 3 results in cv-errors of 41.14, 43.76 and 43.95, respectively. These values are worse than the best solution found with Projection Pursuit Regression (35.01).

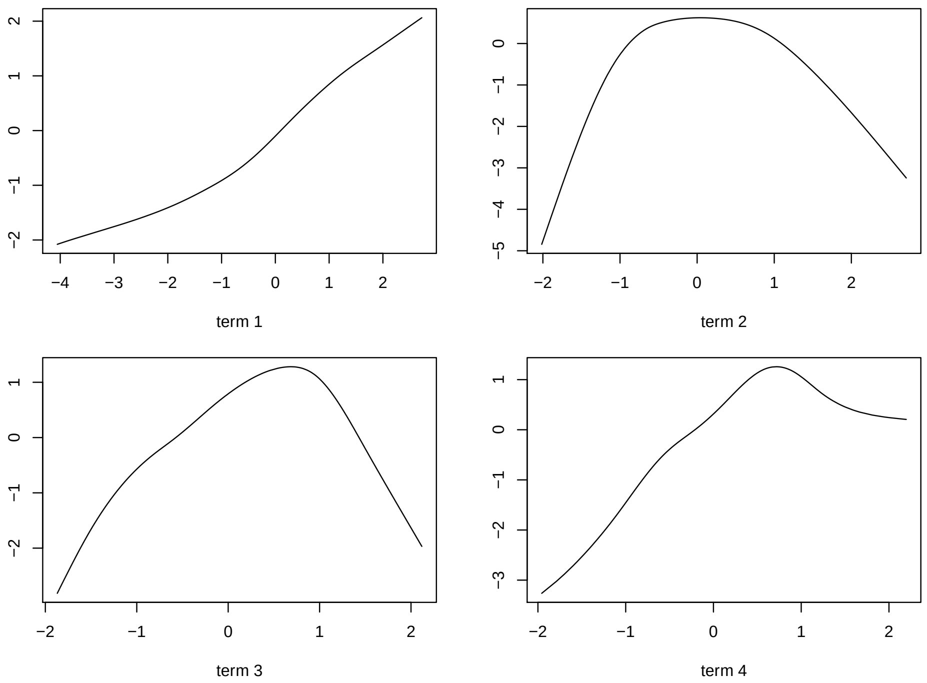
Now let's compute the CV-error of a Generalized additive model (GAM). Here we need to write a wrapper for the training algorithm, otherwise we run into trouble with the special formulas of `gam()`.

```
> library(mgcv)
> wrapGam <- function(formula, data,...){
   gamForm <- wrapFormula(formula, data = data)
   gam(gamForm, data=data, ...)
 }
> gam.cv <- cv(wrapGam, formula=as.formula('logupo3~.'), trace=FALSE)
```

The CV-error of GAM is 37.98, which is bigger than the best PPR, but compete quite well with all the fitted models.

f) The 4 ridge functions look much smoother than those of task c). In the former we used the super smoother (default value), which adapts the smoothness internally. The optimal solution found by cv, however, uses a spline with $df = 6$. The df of the spline fit are smaller than the "effective df" of the super smoother fit. Therefore, the spline fit looks smoother.

```
> ## Which index corresponds to min(cv.ss)?
> which(cv.ss==min(cv.ss),arr.ind=TRUE)
     row col
df 6   2   2
> fit <- ppr(formula=logupo3~.,data=d.ozone.es, sm.method = "spline",
             df = 6, nterms = 4, max.terms = 7)
> par(mfrow = c(2,2))
> plot(fit, type='l')
```



g) The scaling factors (β_k in the manuscript) of the ridge terms are

```
> round(fit$beta,digits=3)
```

```
term 1 term 2 term 3 term 4
 0.534  0.263  0.312  0.305
```

and the α-matrix is given by

```
> round(fit$alpha,2)
```

```
     term 1 term 2 term 3 term 4
vdht   0.48  -0.09  -0.06  -0.02
wdsp  -0.25  -0.11   0.09   0.12
hmdt  -0.03  -0.54   0.16   0.22
sbtp   0.48   0.43  -0.04  -0.37
ibht  -0.06  -0.17  -0.05   0.01
dgpg   0.47  -0.45  -0.11  -0.66
ibtp   0.07  -0.10  -0.10   0.32
vsty  -0.44  -0.10   0.05   0.26
day   -0.23  -0.49  -0.97   0.44
```

The coefficient of the variable `ibht` has relatively small absolute values in all of the terms: It seems to be not so important. `term 3` is dominated mainly by the value of `day`.