Scalar on Function-Regression: PASAT scores of MS patients with `pfr`

Model Specification

$E(y_{ij}) = g(\beta_0 + b_i + f(T) + \int x(s)\beta(s)ds)$ with basis for $\beta(s)$ given by spline basis (`lf`) or FPC basis (`fpc`), or $E(y_{ij}) = g(\beta_0 + b_i + f(T) + \int f(x(s), s)ds)$ (`af`)

```
set.seed(12121)
train_ind <- unlist(c(which(dti$Nscans == 2), #88 obs
  tapply(which(dti$Nscans == 3), dti$ID[dti$Nscans == 3], sample, size = 2),
  tapply(which(dti$Nscans > 3), dti$ID[dti$Nscans > 3], sample, size = 3)))
#243 (of 340) in training set, sampled stratified by ID so re() won't shit the bed
test <- dti[-train_ind, ]
```

Cross-validate for 2 to 40 FPCs for `fpc`-term:

```
cv_fpc_be <- cbind(FPCs = seq(2, 40, by=2), Brier = sapply(seq(2, 40, by=2), function(npc) {
  m <- pfr(pasat_rel ~ re(subject) + s(visit.time) +
    fpc(cca, ncomp=npc, k=40), family=betar, data =dti,
    method = "GCV.Cp",
    weights = ifelse(row(dti)[, 1] %in% train_ind, 1, 0))
  mean((fitted(m)[-train_ind] - test$pasat_rel)^2)
}))
print(cv_fpc_be)
```

```
##       FPCs       Brier
##  [1,]    2 0.009980595
##  [2,]    4 0.010016440
##  [3,]    6 0.010019032
##  [4,]    8 0.010014593
##  [5,]   10 0.010012013
##  [6,]   12 0.009990938
##  [7,]   14 0.009983679
##  [8,]   16 0.009969728
##  [9,]   18 0.009979690
## [10,]   20 0.009966733
## [11,]   22 0.009963019
## [12,]   24 0.009960177
## [13,]   26 0.009959967
## [14,]   28 0.009956529
## [15,]   30 0.009957831
## [16,]   32 0.009957294
## [17,]   34 0.009956263
## [18,]   36 0.009958456
## [19,]   38 0.009958569
## [20,]   40 0.010137270
```

So use 34 FPCs, arguably could reduce this to just 2 since differences in predicitive accuracy are small (but coefficient function actually changes quite a bit, much more similar to spline based coefficient function for 2 FPCs).

Fit models with linear spline- and FPC-based effects and non-linear effect of CCA-FA:

```
formulas <- list(
  lf = pasat_rel ~ re(subject) + s(visit.time, k=10) +
    # use cubic B-splines with first order difference penalty over t
    lf(cca, k=10, bs="ps", m=c(2,1)),
  fpc = pasat_rel ~ re(subject) + s(visit.time) +
    fpc(cca, ncomp = 34, k=40),
  af = pasat_rel ~ re(subject) + s(visit.time) +
    # use second order differences for `af` in x-direction to penalize deviations
    # from linearity.
    af(cca, k=c(5, 5), bs="ps", m=list(c(2, 1), c(2, 2)))) #t, x
# use 0-weights to leave out test data:
models1_be <- lapply(formulas, function(f) {
  pfr(f, data = dti, family=betar,  method = "REML",
    weights = ifelse(row(dti)[, 1] %in% train_ind, 1, 0))
})
preds1_be <- sapply(models1_be, function(m) fitted(m)[-train_ind])
fits1_be <- sapply(models1_be, function(m) fitted(m)[train_ind])
cm1_be <- lapply(models1_be, function(m) list(refundDevel:::coef.pfr(m, 1),
  refundDevel:::coef.pfr(m, 2), refundDevel:::coef.pfr(m, 3)))
#Brier scores on test set:
(brier1_be <- colMeans((preds1_be - test$pasat_rel)^2))
```

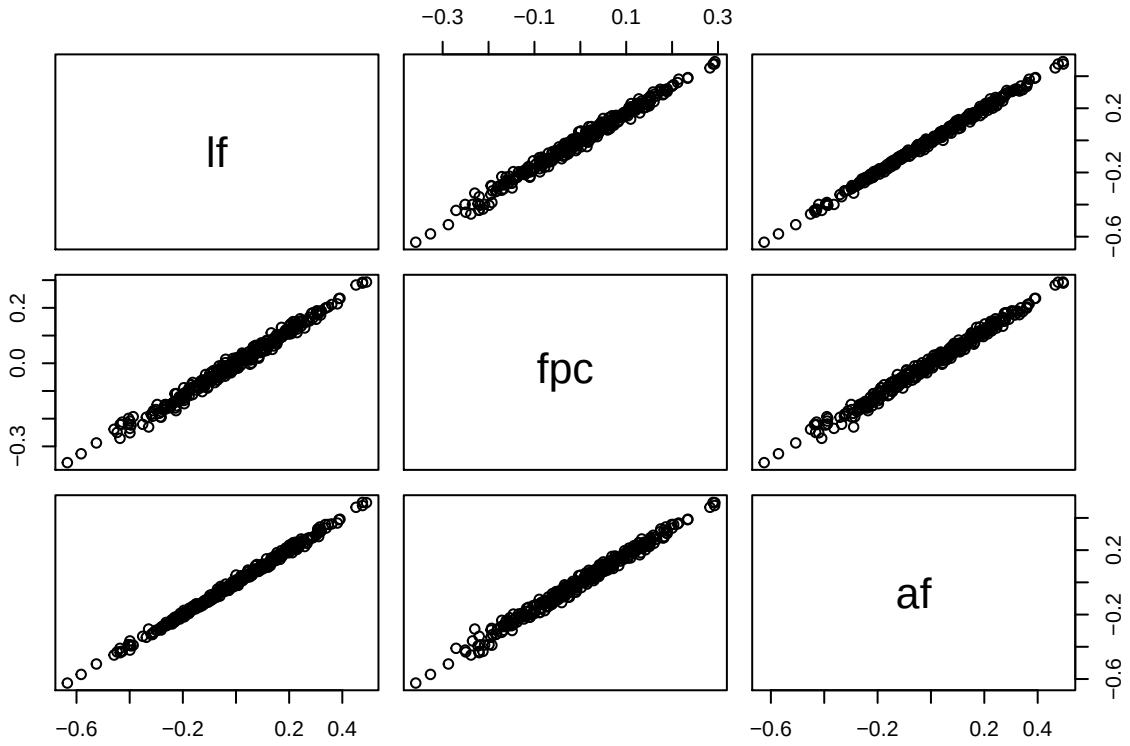
```
##          lf         fpc          af
## 0.009912787 0.009956263 0.009933235
```

```
#deviance on test set:
(logscore1_be <- sapply(models1_be, function(m) {
  mu <- fitted(m)[-train_ind]
  phi <- exp(m$family$getTheta())
  -mean(dbeta(test$pasat_rel_be, shape1 = mu * phi, shape2 = phi - mu * phi,
    log=TRUE))
}))
```

```
##        lf       fpc        af
## -1.108382 -1.105948 -1.107722
```

```
terms1_be <- lapply(models1_be, function(m) predict(m, type="terms", se=TRUE))

#check similarity of contributions of CCA effect to additive predictor:
pairs(sapply(terms1_be, function(x) x$fit[,3]))
```




```
# (different scale for `fpc`!)

saveRDS(list(m=models1_be, pr = preds1_be, fit = fits1_be, cm = cm1_be,
  brier = brier1_be, logscore = logscore1_be, terms=terms1_be), "models/m1_be.rds")
```

Plot coefficents:

```
## Warning: Removed 95 rows containing non-finite values (stat_contour).
```

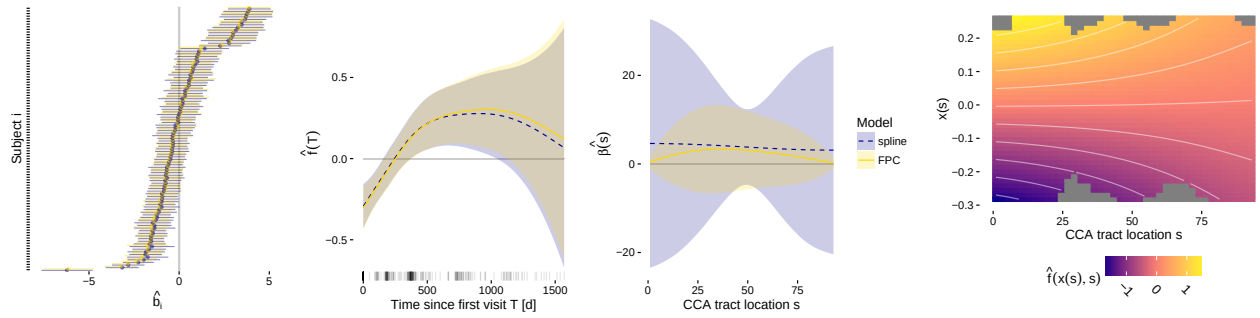


Figure 1: Coefficient estimates

Computational Details:

```
sessionInfo()
```

```
## R version 3.3.0 (2016-05-03)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 14.04.4 LTS
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=de_DE.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=de_DE.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=de_DE.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] gridExtra_2.2.1    ggplot2_2.1.0      zoo_1.7-13
## [4] mgcv_1.8-12        nlme_3.1-128       refundDevel_0.1-16
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.5      knitr_1.13       magrittr_1.5     magic_1.5-6
##  [5] splines_3.3.0    MASS_7.3-44      munsell_0.4.3    colorspace_1.2-6
##  [9] lattice_0.20-33  grpreg_3.0-0     fda_2.4.4        minqa_1.2.4
## [13] stringr_1.0.0    plyr_1.8.3       tools_3.3.0      parallel_3.3.0
## [17] grid_3.3.0       gtable_0.2.0     htmltools_0.3.5  gamm4_0.2-3
## [21] yaml_2.1.13      lme4_1.1-12      digest_0.6.9     Matrix_1.2-6
## [25] nloptr_1.0.4     formatR_1.4      codetools_0.2-14 RLRsim_3.1-2
## [29] evaluate_0.9     rmarkdown_0.9.6  labeling_0.3     stringi_1.1.1
## [33] pbs_1.1          scales_0.4.0     boot_1.3-17
```