funFEM: an R package for clustering functional data

Julien Jacques

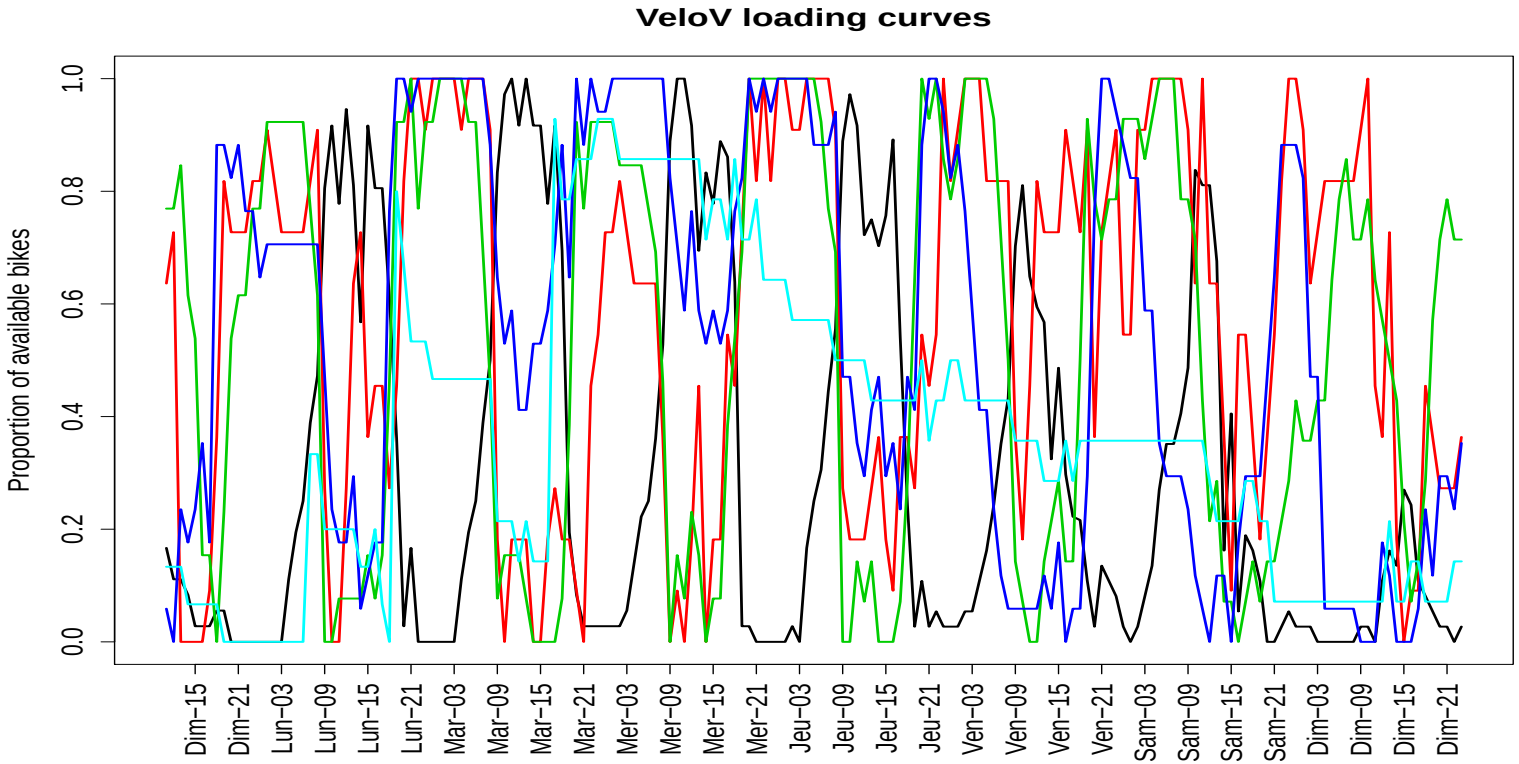
Laboratoire ERIC, University of Lyon

This is a joint work with
Charles Bouveyron (U. Paris Descartes) & Etienne Côme (IFSTTAR)

The data

The data [1]

- proportion of available bikes at each of the 345 Velo'V stations in Lyon
- observation period: Sunday 9th March - Sunday 16th March, 2014
- about one observation per hour



[1] available in the funFEM package

Outline

Transformation of the observed curves

Let us first assume that the observed curves $\{x_1, ..., x_n\}$ are independent realizations of a L_2-continuous stochastic process $X = \{X(t)\}_{t\in[0,T]}$.

Let us also assume that the stochastic process X admits the following basis expansion:

$$X(t) = \sum_{j=1}^{p} \gamma_j(X)\psi_j(t), \tag{1}$$

where:

- $\{\psi_1, \ldots, \psi_p\}$ is a basis of functions,
- $\Gamma = (\gamma_1(X), ..., \gamma_p(X))$ is a random vector in $\mathbb{R}^p$.

The DFM model

Let $F[0,T]$ be a latent subspace of $L_2[0,T]$ assumed to be:

- the most discriminative subspace for the K groups,
- spanned by a basis of d basis functions $\{\varphi_j\}_{j=1,\ldots,d}$ with $d < K < p$.

The basis $\{\varphi_j\}_{j=1,\ldots,d}$ is obtained from $\{\psi_j\}_{j=1,\ldots,p}$ through a linear transformation

$$\varphi_j = \sum_{\ell=1}^{p} u_{j\ell}\psi_\ell,$$

such that the $p \times d$ matrix $U = (u_{j\ell})$ is orthogonal. Let $\Lambda = \{\lambda_1, ..., \lambda_n\}$ be

the basis expansion coefficients of the stochastic process $X(t)$ in the basis $\{\varphi_j\}_{j=1,\ldots,d}$.

The DFM model

The previous modeling implies that Γ and Λ are linked by:

$$\Gamma = U\Lambda + \varepsilon, \tag{2}$$

where $\varepsilon \in \mathbb{R}^p$ is an independent and random noise term.

Distribution assumptions, for $k = 1, ..., K$.:

$$\Lambda_{|Z=k} \sim \mathcal{N}(\mu_k, \Sigma_k),$$

$$\varepsilon \sim \mathcal{N}(\mathbf{0}, \Xi),$$

The marginal distribution of Γ is then:

$$f(\gamma) = \sum_{k=1}^{K} \pi_k \phi(y; m_k, S_k),$$

where $m_k = U\mu_k$ and $S_k = U\Sigma_k U^T + \Xi$.

The DFM model

We finally assume that the noise covariance matrix Ξ is such that $\Delta_k = W^T S_k W$ has the following form:

$$\Delta_k = \begin{pmatrix} \Sigma_k & & \mathbf{0} \\ & \beta_k & & 0 \\ & & \ddots & \\ \mathbf{0} & & & \ddots \\ & 0 & & & \beta_k \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}a\\a\end{matrix}}\right\} d \leq K-1 \\ \left.\vphantom{\begin{matrix}a\\a\end{matrix}}\right\} (p-d) \end{matrix}$$

where $W = [U, V]$.

This model is referred to by $\mathrm{DFM}_{[\Sigma_k \beta_k]}$ and 11 submodels can be obtained by constraining parameters within or between groups.

Estimation *via* an EM-like algorithm

The funFEM algorithm alternates over:

- **a E step** which computes the posterior probabilities $t_{ik} = E[z_{ik} = 1|y_i]$,

Estimation *via* an EM-like algorithm

The funFEM algorithm alternates over:

- **a E step** which computes the posterior probabilities $t_{ik} = E[z_{ik} = 1|y_i]$,
- **a F step** which determines the orientation matrix U according to the $t_{ik}^{(q)}$ by solving:

$$\max_U \frac{Var\left[E\left[\omega(X)|Z\right]\right]}{Var\left[\omega(X)\right]}, \text{ wrt } \int u_j(t)u_l(t)dt = 0, \quad \forall j \neq l$$

where $\omega(X) = \int_0^T X(t)u(t)dt$ is the projection of X on the function u.

Estimation *via* an EM-like algorithm

The funFEM algorithm alternates over:

- **a E step** which computes the posterior probabilities $t_{ik} = E[z_{ik} = 1|y_i]$,
- **a F step** which determines the orientation matrix U according to the $t_{ik}^{(q)}$ by solving:

$$\max_U \frac{Var\left[E\left[\omega(X)|Z\right]\right]}{Var\left[\omega(X)\right]}, \text{ wrt } \int u_j(t)u_l(t)dt = 0, \quad \forall j \neq l$$

where $\omega(X) = \int_0^T X(t)u(t)dt$ is the projection of X on the function u.

Proposition: U is solution of the generalized eigenproblem

$$\mathbf{\Gamma' T T' \Gamma} W \nu = \eta \mathbf{\Gamma' \Gamma} W \nu,$$

where $\mathbf{\Gamma} = (\gamma_{ij})_{i,j}$, $\mathbf{T} = \left(\frac{t_{ik}^{(q-1)}}{\sqrt{n_k^{(q-1)}}} \right)_{i,k}$ and $\mathbf{W} = \int_{[0,T]} \Psi(s)\Psi'(s)ds$.

Estimation *via* an EM-like algorithm

The funFEM algorithm alternates over:

- **a E step** which computes the posterior probabilities $t_{ik} = E[z_{ik} = 1|y_i]$,
- **a F step** which determines the orientation matrix U according to the $t_{ik}^{(q)}$ by solving:

$$\max_U \frac{Var\left[E\left[\omega(X)|Z\right]\right]}{Var\left[\omega(X)\right]}, \text{ wrt } \int u_j(t)u_l(t)dt = 0, \quad \forall j \neq l$$

where $\omega(X) = \int_0^T X(t)u(t)dt$ is the projection of X on the function u.

Proposition: U is solution of the generalized eigenproblem

$$\mathbf{\Gamma' T T' \Gamma} W \nu = \eta \mathbf{\Gamma' \Gamma} W \nu,$$

where $\mathbf{\Gamma} = (\gamma_{ij})_{i,j}$, $\mathbf{T} = \left(\frac{t_{ik}^{(q-1)}}{\sqrt{n_k^{(q-1)}}} \right)_{i,k}$ and $\mathbf{W} = \int_{[0,T]} \Psi(s)\Psi'(s)ds$.

- **a M step** which updates the mixture parameters.

Outline

The main function: `funFEM`

Main arguments

- `fd`: a functional data object produced by the `fda` package
- `K`: integer vector of numbers of clusters
- `model`: the specific DFM submodels ("all" is possible)
- `crit`: the criterion to be used for model selection ('bic', 'aic' or 'icl')
- `lambda`: the ℓ_0 penalty (between 0 and 1) for the sparse version

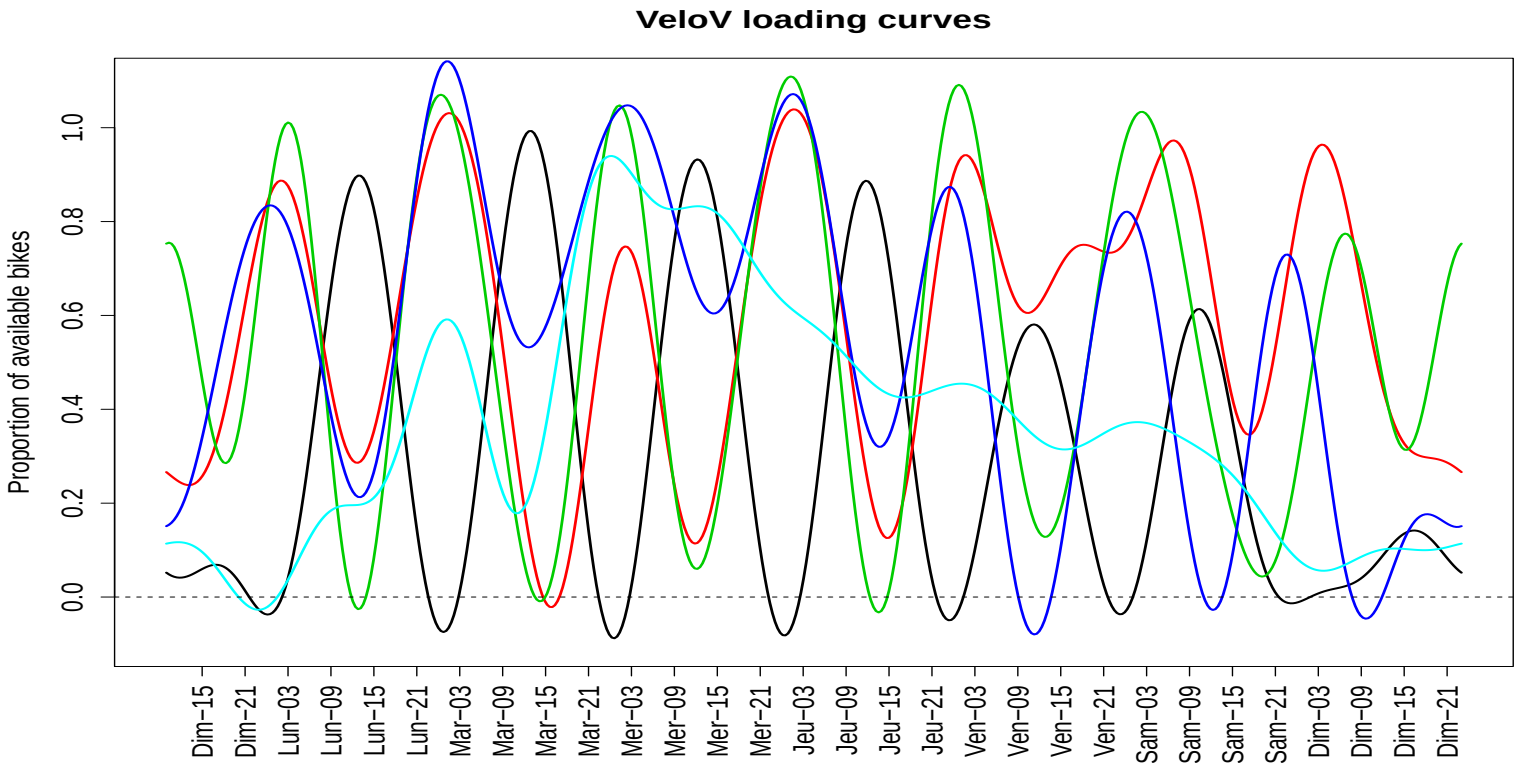
Main outputs

- `cls`: the group membership of each individual
- `P`: the posterior probabilities of each individual for each group
- `U`: the orientation of the functional subspace according to the basis functions
- `aic, bic, icl, loglik...`

Outline

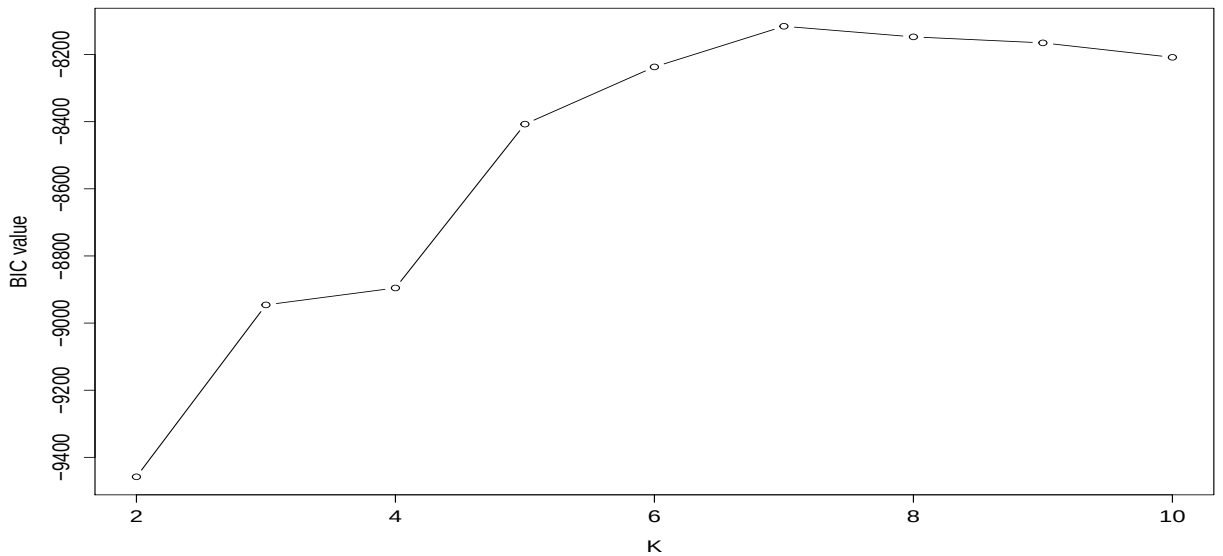
Clustering with funFEM

```
# Load the velov data and smoothing
R> library(funFEM)
R> data(velov)
R> basis <- create.fourier.basis(c(0, 181), nbasis=25)
R> fdobj <- smooth.basis(1:181,t(velov$data),basis)$fd
```



Clustering with funFEM

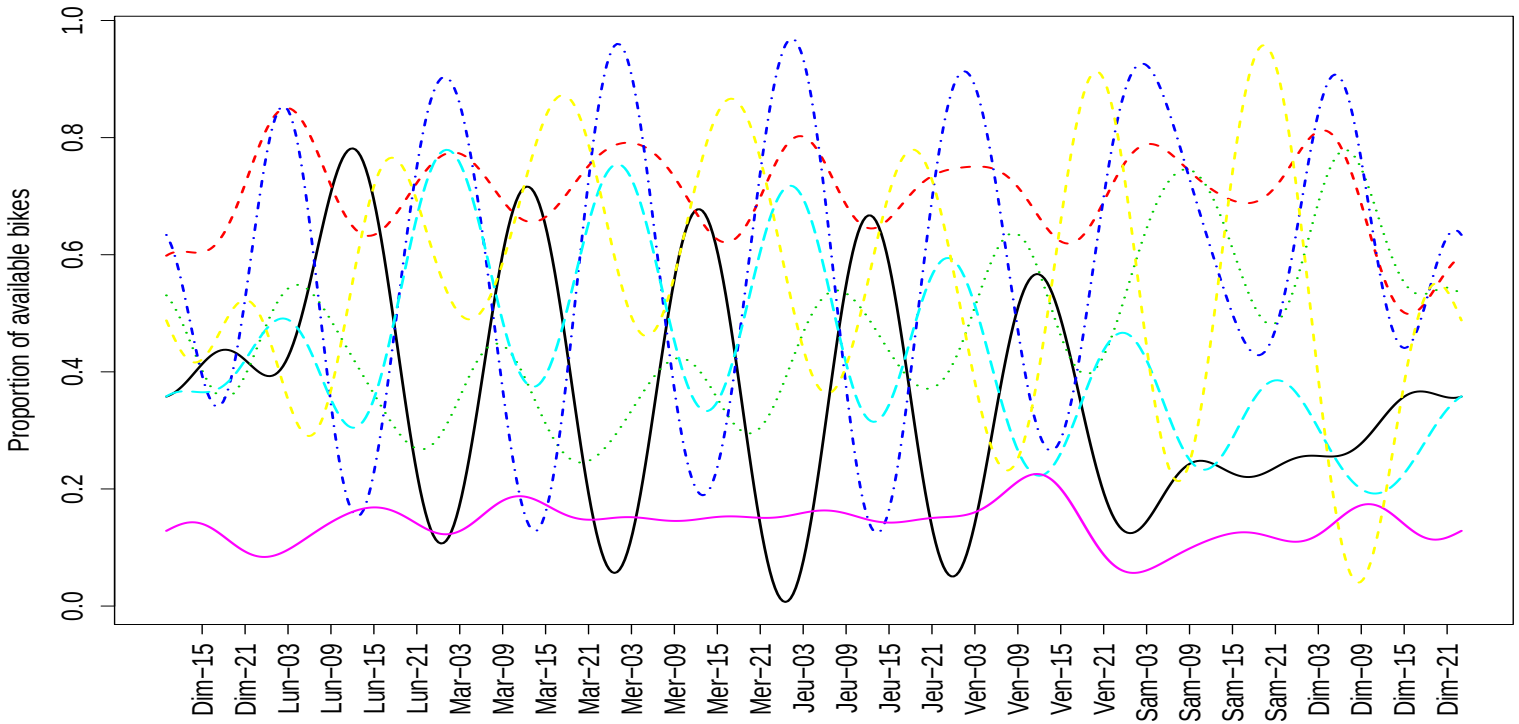
```
# Clustrering with FunFEM
R> res = funFEM(fdobj,K=2:10,model="AkjBk",init="kmeans",lambda=0)
R> plot(1:10,res$plot$bic,type='b',xlab='K',main='BIC')
```



BIC tells us to select $K = 7$

Clustering with funFEM

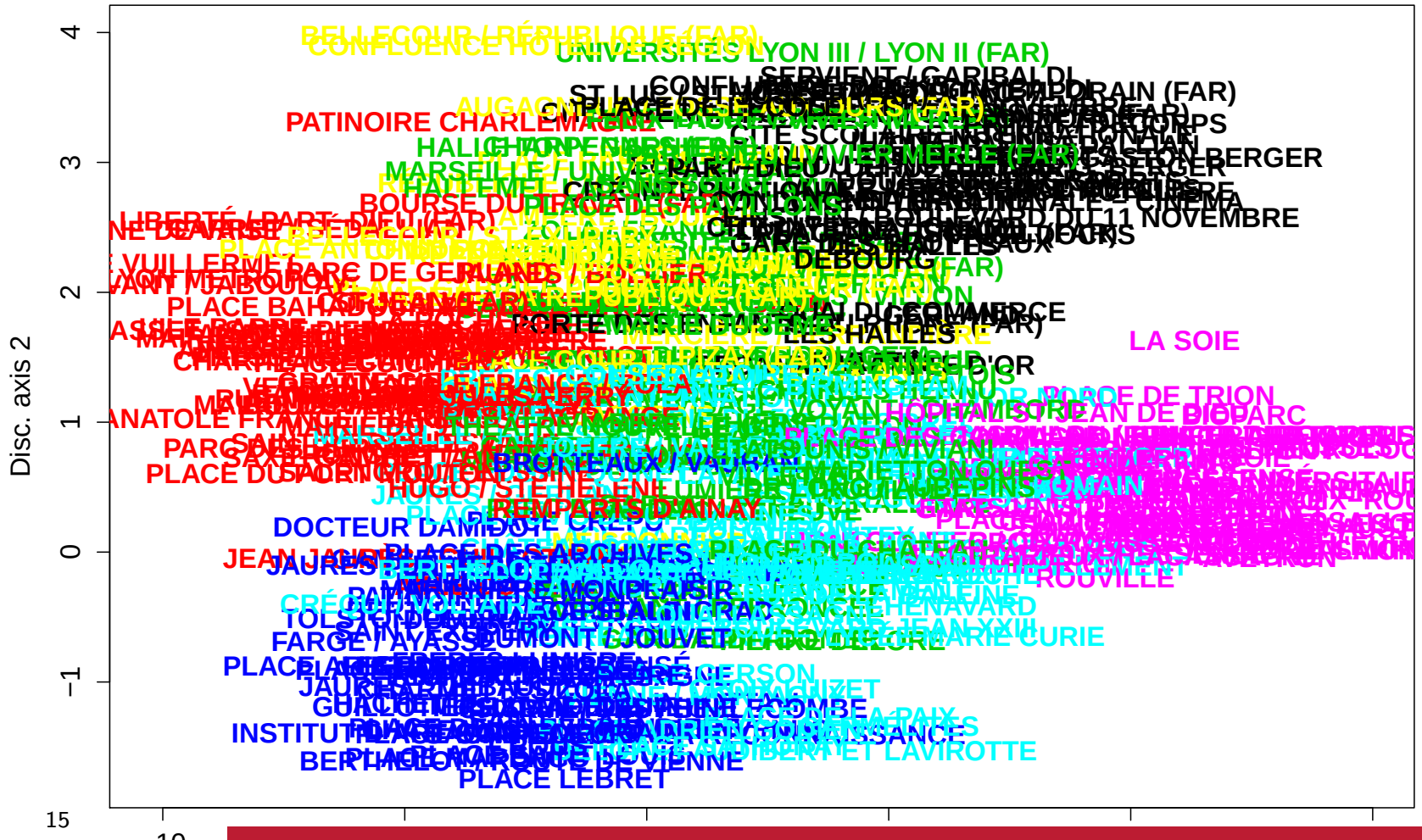
```
# Visualization of group means
R> fdmeans = fdobj; fdmeans$coefs = t(res$prms$my)
R> plot(fdmeans,col=1:res$K,xaxt='n',lwd=2)
R> axis(1,at=seq(5,181,6),labels=mesdates[seq(5,181,6)],las=1)
```



some interpretations: empty / full, morning / afternoon ...

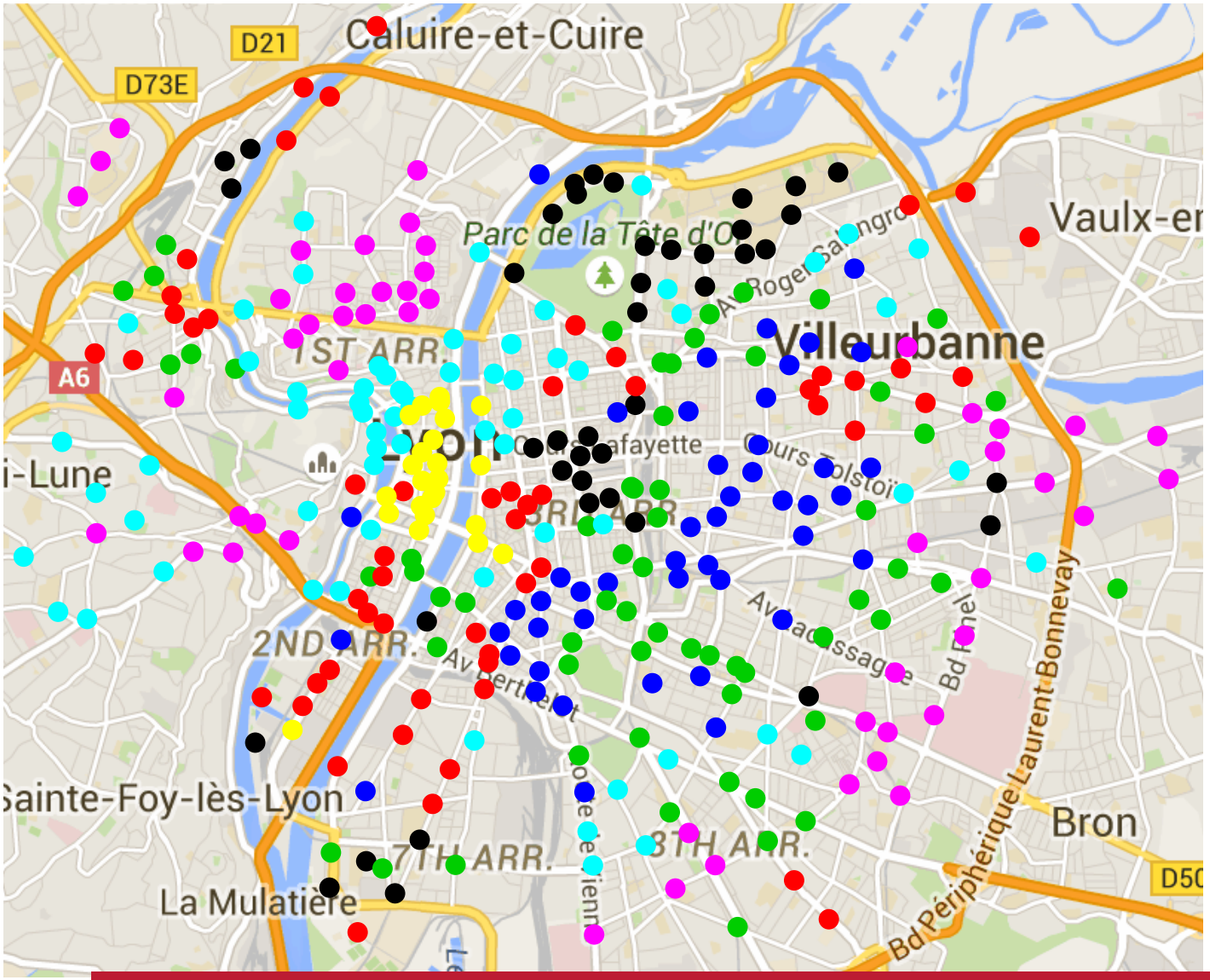
Clustering with funFEM

```
# Visualization in the discriminative subspace (projected scores)
R> plot(t(fdobj$coefs) %*% res$U,type='n',xlab='Disc.  axis 1',ylab='Disc.  axis 2')
R> text(t(fdobj$coefs) %*% res$U,labels=names,col=res$cls,cex=1,font=2)
```



Clustering with funFEM

Finally, we can visualize the clusters repartition in Lyon using `ggmap`



Sources

- `funFEM` available on CRAN:
  `http://cran.r-project.org/web/packages/funFEM`
- preprint available on HAL:
  `http://hal.archives-ouvertes.fr/hal-01024186`