`tidyfun`: Tidy Functional Data

A new framework for representing and working with function-valued data in `R`

Fabian Scheipl[1] Arthur Jeff Goldsmith[2]

[1]: Dept. of Statistics, LMU Munich

[2]: Columbia University Mailman School of Public Health

JSM 2018

tidyfun

The goal of tidyfun is to provide a tidyverse-compliant, accessible and well-documented way to deal with functional data in R, specifically for data wrangling and exploratory analysis.
tidyfun provides:

- new R data types for representing functional data: tfd & tfb
- arithmetic operators, descriptive statistics and graphics functions for such data
- tidyverse-verbs for handling functional data **inside** data frames.

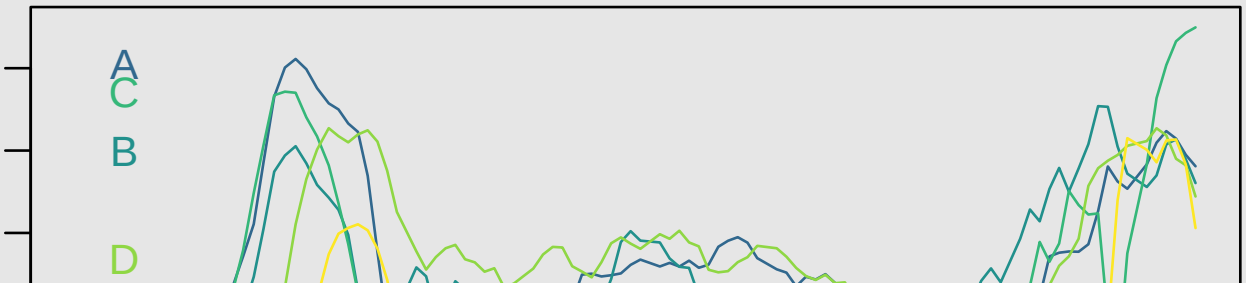
`tf`-Class: Definition

`tf`-class

`tf` is a new data type for (vectors of) functional data:

- abstract superclass for functional data
  - as (argument, value)-tuples: subclass `tfd`, also irregular or sparse
  - or in basis representation: subclass `tfb`
- basically, a `list` of numeric vectors
  (... since `lists` work well as columns of data frames ...)
- with additional attributes that help define *function-like* behavior:
  - how to **evaluate** the given 'functions' for new arguments
  - their **domain**
  - the **resolution** of the argument values

Example Data



tf-Class: Methods

Subset & subassign

```
ex[1:2]
```

```
## tfd[2] on (0,1) based on 93 evaluations each
## interpolation by approx_spline
## A: (0.000,0.49);(0.011,0.52);(0.022,0.54); ...
## B: (0.000,0.47);(0.011,0.49);(0.022,0.50); ...
```

```
ex[1:2] = ex[2:1]
ex
```

```
## tfd[5] on (0,1) based on 93 evaluations each
## interpolation by approx_spline
## B: (0.000,0.47);(0.011,0.49);(0.022,0.50); ...
## A: (0.000,0.49);(0.011,0.52);(0.022,0.54); ...
## C: (0.000,0.50);(0.011,0.51);(0.022,0.54); ...
## D: (0.000,0.40);(0.011,0.42);(0.022,0.44); ...
## E: (0.000,0.40);(0.011,0.41);(0.022,0.40); ...
```

Evaluate

```
ex[1:2, seq(0, 1, l = 3)]
```

```
##           0       0.5         1
## B 0.4721627 0.4984125 0.5802742
## A 0.4909345 0.5307563 0.5904773
## attr(,"arg")
```

Wrangling `tfs` inside data frames: `dplyr`

All `dplyr` verbs work well with `tf`-columns:

```
# group-wise functional means:
dti %>% group_by(case, sex) %>% summarize(mean_rcst = mean(rcst, na.rm = TRUE)) %>%
```

```
## # A tibble: 4 x 3
##   case    sex                          mean_rcst
##   <fct>   <fct>                            <tfd>
## 1 control male   (0.0000,0.514);(0.0185,0.50...
## 2 control female (0.0000,0.517);(0.0185,0.53...
## 3 MS      male   (0.0000,0.533);(0.0185,0.52...
## 4 MS      female (0.0000,0.524);(0.0185,0.51...
```

```
# which subjects go below cca = .26:
dti %>% filter(anywhere(cca, value < .26))
```

```
## # A tibble: 3 x 5
##      id sex   case                              cca                          rcst
##   <dbl> <fct> <fct>                            <tfd>                        <tfd>
## 1  2017 male  MS    (0.000,0.38);(0.011,0.38);~ (0.0741,0.519);(0.0926,0.~
## 2  2017 male  MS    (0.000,0.34);(0.011,0.35);~ (0.0000,0.616);(0.0185,0.~
## 3  2083 male  MS    (0.000,0.39);(0.011,0.43);~ (0.0000,0.511);(0.0185,0.~
```

```
# center & scale:
dti %>% mutate(cca = tfb(cca), cca_z = (cca - mean(cca))/sd(cca)) %>% glimpse
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   90.20   97.10   97.80   97.47   98.40   99.40
## Observations: 382
```