Popixplore

Exploring & visualizing data

Marc Lavielle
Inria Saclay, team Popix

May 27, 2013

1 Introduction

popixplore is a set of Matlab functions for visualizing longitudinal data.

What we call *data* is a set of variables: identification of the individuals, time, regression variables, individual covariates, observations, doses (or more generally source terms),....

This data can either be stored in a data file, or may have been simulated. In order to easily manipulate these data, it is important to use a suitable and flexible data structure. When using Matlab, we propose to store the different variables of the data in a cell array. Each cell is a structure which contains information about a given variable. Fields of such structure are

name : the name of the variable

label : the type of variable. Label can be

id: identification (id_i) of the individual, by definition, *id* varies with the individual and define the level of variability *iid*.

covariate: variable (c_i) which varies with the individual. The level of variability is *iid*.

time: time is the reference regression variable for longitudinal data,

regressor: variable (x_{ij}) which varies with the individual and the time. Here, x_{ij} is the value of the regression variable x at time t_{ij} for individual i. If there is no variable *time* in the data, then the first regression variable is the reference. Levels of variability are *iid* and *iobs*.

observation: variable (y_{ij}) which varies with the individual and the time. Levels of variability are *iid* and *iobs*. An observation has different types: *continuous*, *categorical*, *count* and *event*.

source: inputs (u_{im}) (doses for pharmacometric data). Levels of variability are *iid* and *isource*.

header : names of the variables of the matrix **values**.

units : units of the variables of the matrix **values**.

values : numerical values, including the indexes (according to the level of variability), time, regression variables,...

Functions of *popixplore* are:

readdatapx reads a data file and creates a data structure,

exploredatapx performs a graphical exploration of a data structure,

splitdatapx reorganizes a data structure per labels,

summarypx computes basic statistics for a set of covariates.

We will illustrate these functions with this the `joint_demo.m` example.

2 The functions of popixplore

2.1 readdatapx

`data=readdatapx(datafile,info)` read the data file `datafile` and create a cell array data using additional information contained in the structure `info`.

Example `joint_data.csv` contains four different types of observations. These observations are stored in the same column `y` and the column `ytype` identifies the type of observation: PK measurements (concentration, `ytype=1`), PD measurements (prothrombin complex activity PCA, `ytype=2`), repeated events (hemorrhaging, `ytype=3`), categorical data (state of the patient, `ytype=4`).

Each patient receive repeated doses. Column `amt` contains the amounts of drug administered to each patients.

Column `time` contains the times of administration of the drug, the times of measurements (concentration, PCA and state) and the time to events (hemorrhaging).

There are three continuous covariates (weight, height and age) and three categorical covariates (gender, country and treatment group).

#id	time	amt	y	ytype	weight	height	age	gender	country	group
1	0	14.95	.	.	59.8	159.7	44.8	M	Arg	A
1	0	.	115	2	59.8	159.7	44.8	M	Arg	A
1	0	.	0	3	59.8	159.7	44.8	M	Arg	A
1	0	.	2	4	59.8	159.7	44.8	M	Arg	A
1	0.5	.	1.42	1	59.8	159.7	44.8	M	Arg	A
1	4	.	1.4	1	59.8	159.7	44.8	M	Arg	A
1	8	.	1.88	1	59.8	159.7	44.8	M	Arg	A
1	12	.	2.56	1	59.8	159.7	44.8	M	Arg	A
1	16	.	1.42	1	59.8	159.7	44.8	M	Arg	A
1	20	.	1.33	1	59.8	159.7	44.8	M	Arg	A
1	24	14.95	.	.	59.8	159.7	44.8	M	Arg	A
1	24	.	2.53	1	59.8	159.7	44.8	M	Arg	A
1	24	.	47.1	2	59.8	159.7	44.8	M	Arg	A
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
80	240	.	63.1	2	69.2	162.3	57.1	M	Bra	D
80	240	.	1	4	69.2	162.3	57.1	M	Bra	D
80	264	.	71.4	2	69.2	162.3	57.1	M	Bra	D
80	264	.	1	4	69.2	162.3	57.1	M	Bra	D
80	288	.	75.3	2	69.2	162.3	57.1	M	Bra	D
80	288	.	1	4	69.2	162.3	57.1	M	Bra	D
80	312	.	72.8	2	69.2	162.3	57.1	M	Bra	D
80	312	.	1	4	69.2	162.3	57.1	M	Bra	D
80	336	.	69.1	2	69.2	162.3	57.1	M	Bra	D
80	336	.	1	4	69.2	162.3	57.1	M	Bra	D

datafile is a Matlab structure with 2 fields: name and format.

info is a Matlab structure. Field header is mandatory: it allows to identify the content of the data file. The keywords are those used by MONOLIX.

```
datafile.name='joint_data.csv';
datafile.format='csv';  % can be "csv", "space", "tab" or ";"

info.header = {'ID','TIME','AMT','Y','YTYPE','COV','COV','COV','CAT','CAT','CAT'};
info.observation.name={'Concentration','PCA','Hemorrhaging','State'};
info.observation.type={'continuous','continuous','event','categorical'};
info.observation.unit={'mg/l','%','',''};
info.observation.category{4}={'low','high'};
info.covariate.unit={'kg','cm','year','','',''};
info.time.unit='h';
info.dose.unit={'mg/kg'};
```

datafile requires these two input arguments. The output of the function is the data structure data.

```
>> data=readdatapx(datafile,info)
>> whos('data')
  Name      Size             Bytes  Class    Attributes

  data      1x12            192234  cell
```

The first cell contains information on the identification of the individuals

```
>> disp(data{1})
     name: '#id'
    label: 'id'
    value: {80x1 cell}
```

The second cell contains information on the dose regimen (amount and time of administration). Note that any additional information can easily be added as additional fields (route of administration for instance).

```
>> disp(data{2})
      name: 'doseRegimen'
     label: 'source'
     level: {'iid'  'isource'}
    header: {'iid'  'isource'  'time'  'amt'}
      unit: {'i'  'm'  'h'  'mg/kg'}
    values: [560x4 double]

>> disp(data{2}.values(1:5,:))
    1.0000    1.0000         0   14.9500
    1.0000    2.0000   24.0000   14.9500
    1.0000    3.0000   48.0000   14.9500
    1.0000    4.0000   72.0000   14.9500
    1.0000    5.0000   96.0000   14.9500
```

Cells 3 to 8 contain information on the six covariates.

```
>> disp(data{3})
```

```
      name: 'weight'
     label: 'covariate'
      type: 'continuous'
     level: {'iid'}
    header: {'iid'  'weight'}
      unit: {'i'  'kg'}
    values: [80x2 double]

>> disp(data{3}.values(1:5,:))
    1.0000   59.8000
    2.0000   99.4000
    3.0000   50.9000
    4.0000   42.7000
    5.0000   52.7000

>> disp(data{6})
        name: 'gender'
       label: 'covariate'
        type: 'categorical'
    category: {'F'  'M'}
       level: {'iid'}
      header: {'iid'  'gender'}
        unit: {'i'  ''}
      values: [80x2 double]

>> disp(data{6}.values(1:5,:))
     1     2
     2     2
     3     1
     4     1
     5     1
```

Cells 9 to 12 contain information about the observations.

```
>> disp(data{9})
      name: 'Concentration'
     label: 'observation'
     level: {'iid'  'iobs'}
      type: 'continuous'
    header: {'iid'  'iobs'  'time'  'Concentration'}
      unit: {'i'  'j'  'h'  'mg/l'}
    values: [1440x4 double]

>> disp(data{9}.values(1:5,:))
    1.0000    1.0000    0.5000    1.4200
    1.0000    2.0000    4.0000    1.4000
    1.0000    3.0000    8.0000    1.8800
    1.0000    4.0000   12.0000    2.5600
    1.0000    5.0000   16.0000    1.4200
```

2.2 splitdatapx

`datas=splidatapx(data)` transform the cell array `data` into a structure `datas`. Each field of `datas` is a cell array which contains variables with the same label.

Example:

```
>> data=readdatapx(datafile,info);
>> datas=splitdatapx(data)

datas =

      covariate: {1x6 cell}
             id: {[1x1 struct]}
    observation: {1x4 cell}
         source: {[1x1 struct]}
```

```
>> datas.covariate{1}

ans =

      name: 'weight'
     label: 'covariate'
      type: 'continuous'
     level: {'iid'}
    header: {'iid'  'weight'}
      unit: {'i'  'kg'}
    values: [80x2 double]
```

2.3 summarypx

`summarypx(c)` computes basic statistics for a set of covariates. Here c is a data structure which contains only individual covariates.

Example: We first use `splitdatapx` for grouping the six covariates in a data structure `datas.covariate`. We then use `summarypx` for computing basic statistics for these covariates. We compute the mean, the quartiles, the standard deviation, the minimum and maximum values and the correlation matrix of the continuous covariates. We also compute the marginal and conditional distributions of the categorical covariates.

```
>> data=readdatapx(datafile,info);
>> datas=splitdatapx(data);
>> rescov=summarypx(datas.covariate);
>> disp(rescov)
              N: 80
     continuous: {'weight'  'height'  'age'}
           mean: [65.3675 167.3250 44.4700]
       quartile: [3x3 double]
            std: [15.0683 16.3699 4.8091]
            min: [42.7000 129.6000 35.6000]
            max: [99.4000 204.7000 57.1000]
           corr: [3x3 double]
    categorical: {'gender'  'country'  'group'}
```

```
           freq: {[39 41]  [12 37 31]  [20 20 20 20]}
          cdist: {3x3 cell}

>> disp(rescov.quartile)
   52.2000  154.7000   41.4500
   60.0500  163.1000   44.3500
   76.1000  178.9500   46.9000

>> disp(rescov.corr)
    1.0000    0.9274    0.0269
    0.9274    1.0000   -0.0607
    0.0269   -0.0607    1.0000
```

Field `cdist` is a cell array which contains all the conditional distributions of the categorical covariates. `rescov.cdist{k,\ell}` is the conditional distribution of the ℓ-th covariate, given the k-th covariate For instance, `rescov.cdist{1,2}` is the conditional distribution of `country` (Argentina, Brazil, Colombia) given `gender` (female and male) while `rescov.cdist{1,2}` is the conditional distribution of `gender` given `country`.

```
>> disp(rescov.cdist)
    [2x2 double]    [2x3 double]    [2x4 double]
    [3x2 double]    [3x3 double]    [3x4 double]
    [4x2 double]    [4x3 double]    [4x4 double]

>> disp(rescov.cdist{1,2})
    0.1282    0.3077    0.5641
    0.1707    0.6098    0.2195

>> disp(rescov.cdist{2,1})
    0.4167    0.5833
    0.3243    0.6757
    0.7097    0.2903
```

2.4 exploredatapx

`exploredatapx(data)`: graphical exploration of the data structure `data`.

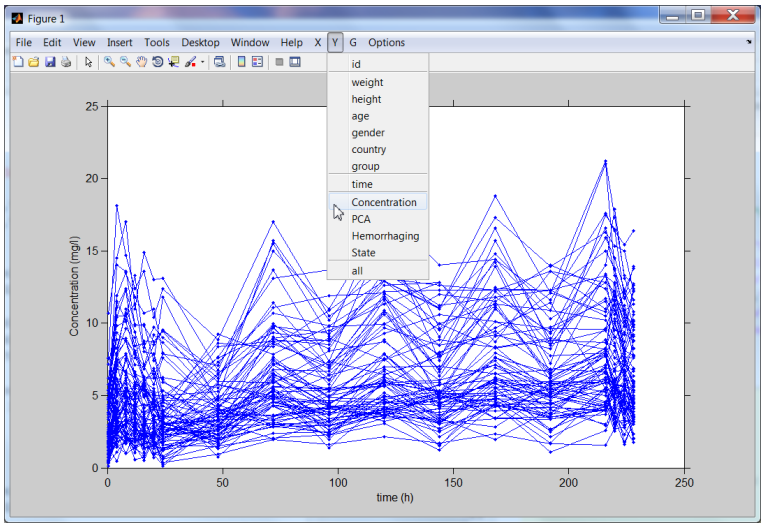
Example

```
>> data=readdatapx(datafile,info);
>> exploredatapx(data)
```

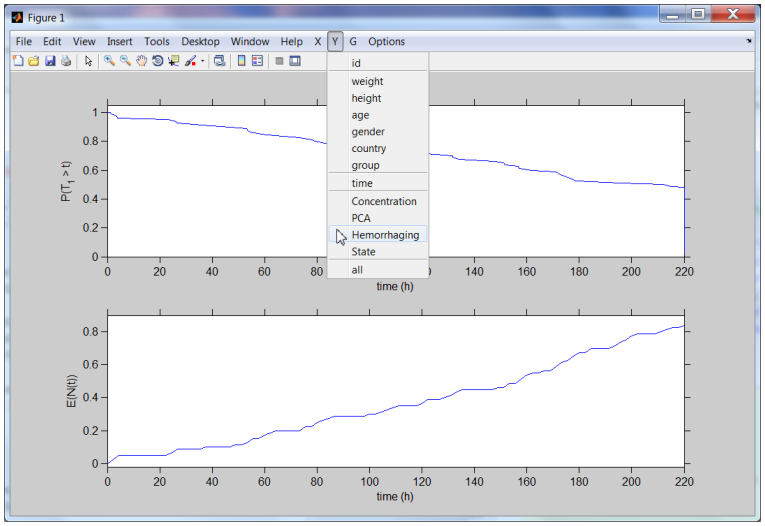
`X` in the menu bar allows to select the variable displayed on the x-axis (by default, the time or the first regression variable are used).

`Y` in the menu bar allows to select the variable displayed on the y-axis (by default, the first observation is displayed).

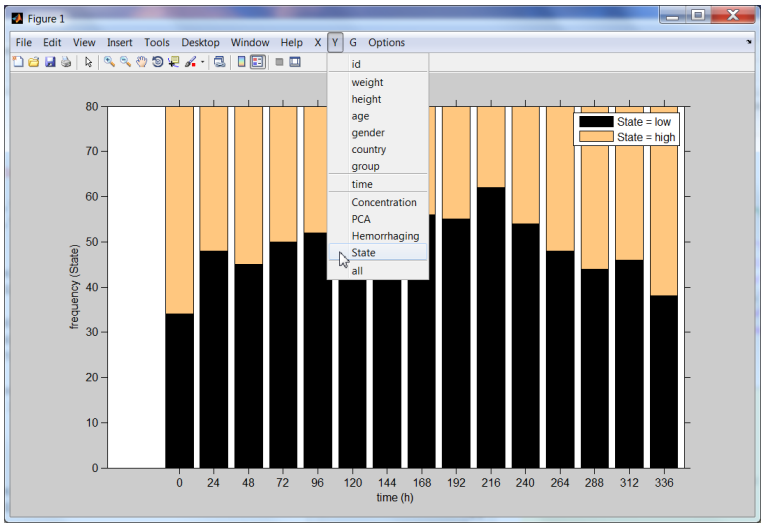
"Spaghetti plots" are used for continuous data.



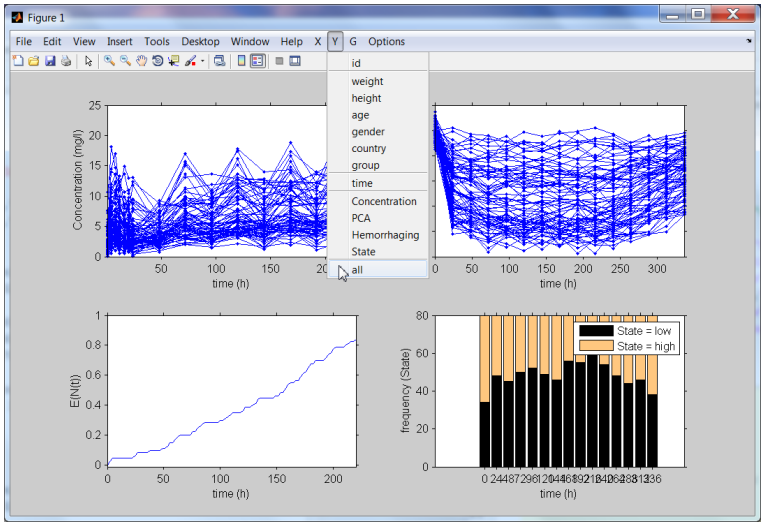
Kaplan-Meier plot is displayed for time-to-event data. in the case of repeated events, the mean cumulate number of events per individual is also displayed



Bar graph is used for categorical data



It is possible to display the different types of observation in different subplots



`G` in the menu bar allows to stratify the data according to a categorical covariate and display the data in different subplots