Line of Balance

Calculations and plotting of line of balance curve as taught by Dr. Ibrahim Odeh on.

Course page

Installation

1. Download and install anaconda It comes at about 400MB so you might want to use a wifi connection.
2. After installation, go to start menu (windows), locate anaconda folder. Open anaconda prompt.
3. Install with ::

   pip install https://github.com/Parousiaic/line_of_balance/archive/master.zip

Optional Step

Creating a virtual environment

Usage

Input data format

1. Make a copy of the included "input.txt" file
2. Edit the entries on the right hand side of the equal sign ("=") for the given parameters. An "input.jpg" file has been included to show the format in which the program expects inputs. Please preserve the order and format of the inputs.

Using the functions

```
import line_of_balance
```

To see an illustration of the concept of line of balance call the illustrate function

```
line_of_balance.illustrate()
```

In case you have a list of points to use for the illustration, call it like this

```
line_of_balance.illustrate_lob(list_of_points)
```

To plot a curve using default dataset provided with this package

```
line_of_balance.default_lob()
```

To plot a curve using your own dataset. Be sure that your data is in the format show in the example dataset. You can choose the input file via a file dialog which is fired up if tkinter is available on your system. Otherwise it will ask you to manually supply the path to your input file.

```
line_of_balance.plot_lob_curve()
```

This function returns a line_of_balance.line_of_balance.LineOfBalance object which you can manipulate. The plot is not show, rather you can see a pdf and a png output in the same directory from where you run the program. But if you have tkinter you can specify a directory to save the output.

The line_of_balance.line_of_balance.LineOfBalance can be initialized as shown below

```
class LineOfBalance:
    """LINE OF BALANCE
    Parameters
    ----------
    activity names : list
    man_hours_per_gang : list
    men_per_gang : list
    buffer_time : int
    productivity_rate : int
    number_of_units_to_produce : int
    hours_per_day : int
    days_per_week : int
    ymin : int"""

    def __init__(self,
        activity_names,
        man_hours_per_unit,
        men_per_gang,
        buffer_time,
        productivity_rate,
        number_of_units_to_produce,
        hours_per_day,
        days_per_week,
        ymin=0):

        self.activity_names = activity_names
        self.man_hours_per_unit = man_hours_per_unit
        self.men_per_gang = men_per_gang
        self.buffer_time = buffer_time
        self.productivity_rate = productivity_rate
        self.number_of_units_to_produce = number_of_units_to_produce
        self.hours_per_day = hours_per_day
        self.days_per_week = days_per_week
        self.number_of_activities = len(self.activity_names)
        self.ymax = self.number_of_units_to_produce
        self.ymin = ymin
```

Note

This is just an experimental project. The results turned out very well for two data sets I have tried it on so far. Example output is included. In the event that it gives results which differ from that which you obtained by working on your own, the safest bet is that your result is the correct one, unless you have very good reasons to doubt your work.

pdf generated with Markdown pdf by yzane