Circles graph

Programátorská dokumentace

Michal Drbohlav | drbohmi1@fit.cvut.cz

Obsah

- Zadání
- Popis programu
  - Kontrola parametrů
  - Kontrola vstupních souborů
  - Řazení a spojování
  - Příprava dat
  - Generování grafu
  - Generování animace
- Funkce
  - soft_error
  - error
  - verbose
  - percentage_done
  - is_number
  - pattern_time_format
  - check_time_format
  - check_max
  - check_min
  - check_max_time
  - check_min_time
  - check_time
  - check_speed
  - check_fps
  - check_legend
  - check_name
  - check_gnuplot
  - check_effect
  - check_pathname
  - check_file
  - check_data_line
  - select_drawable_data
  - count_frames
  - get_max_date
  - get_min_date
  - set_speed_fps_if_needed
  - set_y_range
  - set_x_tics
  - load_config
  - load_data_file
  - generate_video
  - process_data

Zadání

Zadáním je vytvořit script, který ze zadaných vstupních dat (uložených v souboru/souborech nebo na webu) vytvoří animaci (jako sadu snímků) grafu. Výstupní soubory budou uloženy v adresáři s názvem podle přepínače '**-n**' (jinak název scriptu). Pokud již adresář existuje, název adresáře bude končit na sufix '**_i**', kde *i = max(i,0)+1*. Script bude možné parametrizovat pomocí přepínačů a pomocí konfiguračního souboru (i současně - přednost mají přepínače).

Popis programu

Tato kapitola obsahuje popis nejdůležitějších částí programu. Některé věci jsou blíže popsány přímo ve zdrojovém kódu.

Kontrola parametrů

Nejprve se pomocí '**argparse**' načtou všechny přepínače do zvolených proměnných. Díky tomu lze poté určit pořadí kontroly jednotlivých parametrů (např kontrola '**-t**' před '**-x**'). Při použití neznámého přepínače bude výstupem ukázka použití.
Následně se provede načtení konfiguračního souboru, pokud by zadán. Nejprve se vyberou pouze direktivy s hodnotou (zahogí se prázdné řádky, bílé znaky i komentáře), poté se pro jednotlivé direktivy kontrolují shodné přepínače, zdali nebyly zadány. Pokud ano, tak se direktiva z konfiguračního souboru přeskočí, protože přepínače mají přednost. Při nalezení neznámé direktivy bude zobrazeno varování na základě nastavení hodnoty 'verbose'.

Kontrola vstupních souborů

V případě, že cesta ke vstupnímu souboru začíná na 'http', tak se vstupní soubor načte z internetu pomocí metody '**urllib.request.urlopen**'. S takto načteným souborem se poté pracuje stejně jako s každým lokálním souborem.
Všechny vstupní soubory se testují, a pokud některý neprojde testem, tak se pokračuje dalším souborem.
Poté se prochází postupně v cyklu obsah všech validních vstupních souborů. V tomto cyklu se každý řádek rozdělí na 2 hodnoty, time a value. Kontroluje se formát časové značky, pořadí datumů v rámci jednoho souboru nebo zdali je hodnota číslo.
Zdali byla zadaná hodnota pro 'x-min' či 'x-max', tak se podle data vyberou pouze řádky, které se nachází v daném rozmezí.

Fce na kontrolu dat vypadá takto:

```
pattern = pattern_time_format(time_format)
if not re.compile("^" + pattern + "$").match(time):
    return 1
elif not is_number(value):
    return 2
return 0
```

Dále se kontroluje, zdali zadaný datum neobsahuje nesprávné hodnoty, jako např. 13. měsíc.

Nakonec se časová značka převede na vteřiny a uloží se pro další použití společně s hodnotou, která k ní náleží.
Na konci každé iterace se uloží čas do proměnné '**prev**', která se v další iteraci kontroluje s novým časem, zdali jsou data ve správném pořadí.
Všechny validní řádky se ukládají do pole '**suitable_data**', kde každý prvek odpovídá jednomu vstupnímu souboru.

Řazení a spojování

Zde se již pracuje s polem 'suitable_data', vytvořeným v předchozím kroku.
Nejprve se vstupní data seřadí podle prvního datumu každého prvku v poli, a to pomocí bubble sortu:

```
for index_file in range(len(suitable_data)-1, 0, -1):
    for i in range(index_file):
        time = suitable_data[i].split()[0]
        timeNext = suitable_data[i+1].split()[0]
        if time > timeNext:
            tmp = suitable_data[i]
            suitable_data[i] = suitable_data[i+1]
            suitable_data[i+1] = tmp
```

Následné se zkontroluje, zdali se rozsahy datumů v jednotlivých prvcích pole překrývají:

```
overlaping = False
    for index, i_data in enumerate(suitable_data):
        if index == 0:
            continue
        prev_end = suitable_data[index-1].split()[-2:-1][0]
        start = suitable_data[index].split()[0]
        if start <= prev_end:
            overlaping = True
            break
```

Pokud je proměnná '**overlaping**' nastavena na '**False**', tak se jednotlivé prvky pole '**suitable_data**' spojí do jednoho:

```
joinedData = ""
for index, i_data in enumerate(suitable_data):
    joinedData += "\n" + i_data if index > 0 else i_data
```

Příprava dat

V této části script spočte všechny potřebné hodnoty. '**Distance**' obsahuje čas, po kterém se má zobrazit další bod (tzv. velikost rozsahu jednoho bodu na časové ose). Každý bod tedy může zahrnovat více řádků vstupu.
Následný cyklus prochází všechna data a vybírá pouze ta, která se budou tisknout na graf. Vzhledem k nastavené metodě buď vypočte hodnotu jako průměr všech vyhovujících vstupů pro jeden bod, nebo vezme tu v absolutní hodnotě nejvyšší.

Dále se kontroluje, pokud bylo nastaveno ‘y-max’ nebo ‘y-min’, zdali je hodnota v tomto rozsahu.
Výsledná data jsou uložena v poli ‘**res_output**’ (jeden záznam pro jeden vstupní soubor, pokud nebyly spojeny).
Zároveň se nalezne nejnižší a nejvyšší hodnota pro osu Y, a pokud bylo nastaveny ‘**ymin**’ na ‘**min**’ nebo ‘**y-max**’ na ‘**max**’, tak se upraví jejich hodnota, aby nad a pod grafem vznikla mezera a body nebyly úplně na hraně grafu.
Nakonec se ještě spočítá velikost posunu bodu v jednom kroku:

```
jump = (ymax - ymin) / settings["steps"]
```

a také počet obrázků nutný pro vygenerování celé animace.

Nyní nezbývá nic jiného, než nastavit obecné parametry pro gnuplot, konkrétně:

```
set term png truecolor
set key off
set xrange ["{xmin}":"{xmax}"] noreverse nowriteback
set yrange [{yrange}] noreverse nowriteback
unset autoscale
set xtics rotate by -45 scale 1 font ",10" ({xtics})
```

ještě připojíme uživatelem zvolené hodnoty a nastavíme ‘**title**’, pokud byl zvolen přepínačem. Také musíme vybrat barvy pro jednotlivé body, pokud jich bylo zadáno méně, nežli je celkový počet vstupních souborů (pokud nebyly spojeny do jednoho).

Generování grafu

V této části se se generují obrázky ‘gnuplotem’. Proměnná ‘**general_gnuplot**’ obsahuje nastavení, které je společné pro všechny obrázky. Proměnná ‘**partial_data**’ obsahuje upravené hodnoty na počáteční a při každém průchodu se upravují. Takto upravené hodnoty se ukládají do proměnné ‘**effect_data**’, které obsahuje již přesné hodnoty, jenž pošleme ‘gnuplotu’ pro každý obrázek.
Výpočet pozice jednoho bodu:

```
"'value' is a target value"
tmp = -1 if value < 0 else 1

"'partial_value' is value for the current frame"
val = partial_value - tmp * jump

if math.fabs(val) <= math.fabs(value) or (val > 0 and value < 0) or
(val < 0 and value > 0):
    val = value

partial_out[index][index_line] = "{} {}".format(partial_time, val)
effect_data[index] += "{} {}\n".format(partial_time, val)
```

Nyní již zavoláme pokaždé gnuplot s zadaným nastavením a hodnotami:

```
gnuplot = subprocess.Popen(["gnuplot", "-persist"],
stdin=subprocess.PIPE).stdin
gnuplot.write(gnuplot_settings.encode())
gnuplot.flush()
```

Toto se provede pro každý snímek animace.
Jednotlivé snímky se ukládají do dočasného adresáře vytvořeného pomocí ‘**tempfile.TemporaryDirectory**’.

Generování animace

Zde se vytvoří cílový adresář (metodou popsanou výše) a pomocí ‘ffmpeg’ se vygeneruje výsledná animace ve formátu ‘**mp4**’:

```
cmd = ''.join(('ffmpeg -i "{}/%0{}d.png"'.format(tmp_dir, digits),
               ' -r {}'.format(settings["fps"]),
               ' {}/{}'.format(settings["name"], video_name)))
proc = subprocess.Popen(shlex.split(cmd), stdout = subprocess.PIPE,
stderr = subprocess.STDOUT)
```

Funkce

Tato kapitola obsahuje seznam funkcí použitých v programu a jejich účel.

soft_error

Vstupní parametry jsou:

`message, req_lvl = 1, verbose_lvl = 1, ignore_error = True`

Tiskne chybové hlášky na ‘stderr’ a pokud nejsou ignorováný méně závažné chyby, tak ukončí vykonávání scriptu.

error

Vstupní parametry jsou:

`message`

Tiskne chybovou hlášku na ‘stderr’ a vždy ukončí vykonávání scriptu.

verbose

Vstupní parametry jsou:

`message, verbose_lvl, req_lvl`

Tiskne hlášku na ‘stderr’ vzhledem k nastavení levelu verbose (přepínač ‘-v’).

percentage_done

Vstupní parametry jsou:

`done, total`

Tiskne, kolik procent je zpracováno.

is_number

Vstupní parametry jsou:

`val`

Vrací ‘True’ nebo ‘False’ vzhledem k tomu, jestli je ‘val’ číslo.

pattern_time_format

Vstupní parametry jsou:

`val`

Vrací vzor pro časový formát.

check_time_format

Vstupní parametry jsou:

`val`

Kontroluje ‘val’, zdali se jedná o správný formát času.

check_max
Vstupní parametry jsou:
`settings, constants`
Kontroluje maximální hodnotu, zdali se jedná o číslo a vrací ji.

check_min
Vstupní parametry jsou:
`settings, constants`
Kontroluje minimální hodnotu, zdali se jedná o číslo a vrací ji.

check_max_time
Vstupní parametry jsou:
`settings, constants`
Kontroluje maximální hodnotu času a vrací ji ve vteřinách.

check_min_time
Vstupní parametry jsou:
`settings, constants`
Kontroluje minimální hodnotu času a vrací ji ve vteřinách.

check_time
Vstupní parametry jsou:
`settings, constants`
Kontroluje nastavený přepínač ‘time’, jestli je validní a porovnává s ‘FPS’ a ‘speed’. Vrací upravený celý slovník ‘settings’.

check_speed
Vstupní parametry jsou:
`settings, constants`
Kontroluje nastavený přepínač ‘speed’, jestli obsahuje validní hodnotu. Vrací pouze hodnotu ‘speed’.

check_fps
Vstupní parametry jsou:
`settings, constants`
Kontroluje nastavený přepínač ‘FPS’, jestli obsahuje validní hodnotu. Vrací pouze hodnotu ‘FPS’.

check_legend
Vstupní parametry jsou:
`val`
Kontroluje, zdali nastavený přepínač ‘legend’ není prázdný string. Vrací jeho hodnotu.

check_name
Vstupní parametry jsou:
`val, constants`
Kontroluje, zdali nastavený přepínač ‘name’ není prázdný string. Vrací jeho hodnotu.

check_gnuplot
Vstupní parametry jsou:
`settings`
Kontroluje uživatelem specifikované parametry pro gnuplot. Validní jsou pouze takové, které začínají na ‘set’, nebo ‘unset’. Vrací je v jednom stringu oddělené novými řádky.

check_effect
Vstupní parametry jsou:
`settings, constants`
Kontrolu nastavené parametry efektu. Vstupní strig rozdělí na jednotlivé parametry podle ‘:’. Poté oddělí název od hodnoty pomocí ‘=’. Nakonec zkontroluje, zdali se jedná o podporovaný parametr a validní hodnotu. Vrací celý slovník ‘settings’ s uloženými novými hodnotami.

check_pathname
Vstupní parametry jsou:
`val`
Kontroluje cestu k souboru, zdali existuje. Zároveň kontroluje, jestli soubor existuje a je čitelný. Vrací absolutní cestu k souboru, nebo vyhazuje výjimku.

check_file
Vstupní parametry jsou:
`val`
Rozhoduje, zdali se jedná o lokální soubor, nebo zdali se má stáhnout z internetu.

check_data_line
Vstupní parametry jsou:
`time, value, time_format`
Kontroluje jeden řádek vstupního souboru. Vrací 1 pro chybu v datu, 2 pro chybu v hodnotě a 0 pro bezproblémový průběh.

select_drawable_data
Vstupní parametry jsou:
`data, distance, settings`
Vybírá (podle zvolené metody i počítá) data, která budou přesně reprezentovat jednotlivé body v grafu. Zároveň najde v absolutní hodnotě největší hodnoty pro min a max na ose Y. Vrací pole s daty, ymax a ymin.

count_frames
Vstupní parametry jsou:
`data, ymax, ymin, jump, delay`
Počítá, kolik framů bude potřebovat každý bod, aby se dostal na svou pozici a vrací nejvyšší hodnotu.

get_max_date
Vstupní parametry jsou:
`data, prevMax`
Vrací maximální datum.

get_min_date
Vstupní parametry jsou:
`data, prevMin`
Vrací minimální datum.

set_speed_fps_if_needed
Vstupní parametry jsou:
`settings, frames`
Nastavuje ‘speed’ a ‘FPS’, pokud nejsou nastavena. Vrací celý slovník ‘settings’.

set_y_range
Vstupní parametry jsou:
`min_val, max_val, ymin, ymax`
Nastavuje rozsah pro osu Y. Ten také vrací.

set_x_tics
Vstupní parametry jsou:
`xmin, xmax`
Nastavuje popisky pro osu X, které zároveň vrací.

load_config
Vstupní parametry jsou:
`settings`
Načítá konfigurační soubor. Jediné co kontroluje je, zdali náhodou direktiva nebyla zadána již přepínačem, jinak jí je všechno jedno (pokud není nalezena neznámá direktiva) a vrací upravený slovník ‘settings’.

load_data_file
Vstupní parametry jsou:
`i_file`
Načítá data ze souborů, dekóduje je a pokud opravdu nějaké načte, tak je vrací.

generate_video

Vstupní parametry jsou:

`settings, digits, tmp_dir`

Vytváří cílový adresář pro animaci, kterou poté generuje pomocí ‘ffmpeg’.

process_data

Vstupní parametry jsou:

`data, settings, constants`

Dopočítává všechny potřebné hodnoty pro generování jednotlivých snímků. Ty poté generuje v dočasném adresáři a volá fci ‘generate_video’.