HKJournalist Module

A Custom Automatic Report Generator for Python program

Xinyi Li

December 24, 2019

Basic Idea

Template

- Write a `.md` report with `{var_name}` placeholders instead of real values in some critical place

Python Runtime

- Maintain a global `dict` variable `config` to fetch mappings in time

```
config = {'var_name':value}
```

- Read `.md` template and fill in real value bound with `var_name`

```
report_template_text = open('template.md','r').read()
Path('raw_report.md').write_text(report_template_text.format(**config))
```

- Call `pandoc` to convert `md` report to `pdf` slides

```
pandoc -t beamer raw_report.md -o report.pdf
```

Wrap Python runtime tasks into a module

```
1 from hkjournalist import Journalist
2
3 config = dict(...)
4
5 reporter = Journalist(template_file='template.md')
6 reporter.hear(config)
7 reporter.report(output_file='output.pdf',beamer=True)
```

Display support for special types:

- `pandas.DataFrame`: $\rightarrow$ 3-line table (if numeric, round to `{.2f}`)
- `matplotlib.axes.SubplotBase`:
  1. save into a `pdf` standalone file in temp directory
  2. use `![]({var_name})` in template for referring
  3. runtime change as `![](../tmp/xxx.pdf)`
  4. **Note:** before assigning it to `dict`, use

```
1 plt.tight_layout()
```

  for better performance.
- `functions`: print its definition
- `list(str)`: concatenate into a sentance, following its length (e.g. show features list)

Install

Pre-requirements

1. Pandoc: https://pandoc.org/installing.html
2. pdfLaTeX: integrated in TeXLive and MacTeX for MacOS users.

Make sure your environment variables set properly.

- Install from PyPI:

```
pip install hkjournalist
```

For `Jupyter Notebook` users with Chrome

- cannot open `.pdf` on file browser. Download Firefox.

Example

template.md

```
% Hello World
% Xinyi Li
% 2019-12-19

---

### sine plot

![]({sin_plot})

### sine table

{sin_table}

### sine function

```{.python}
{sin_func}
```
```

- **Note**: use {{}} to escape {}

Example I

demo.py (leave out headers)

```
config = {}
def sin_2x_and_cos_2x(x):
    y = np.sin(x) * np.sin(x) + np.cos(x) *  np.cos(x)
    return y

x = np.arange(0, 4 * np.pi, 0.1)
y1 = np.sin(x)
y2 = np.cos(x)

df = pd.DataFrame({'x': x, 'sin(x)': y1, 'cos(x)': y2})
df['sin^2^(x)+cos^2^(x)'] = sin_2x_and_cos_2x(df['x']).values
df = df.set_index('x')

# plot sine curve as sin_plot
ax = df.plot()
```

Example II

demo.py (leave out headers)

```
16 plt.tight_layout()
17 config['sin_plot'] = ax
18
19 # random select 5 point (x,y) as sin_table
20 config['sin_table'] = df.sample(5)
21 # sin_2x_and_cons_2x as sin_func
22 config['sin_func'] = sin_2x_and_cos_2x
23
24 # HK journalist runs faster than everyone! hear variables and make a
       big report
25 reporter = Journalist(template_file='template.md')
26 reporter.hear(config)
27 reporter.report(output_file='big_news.pdf', beamer=True,
       overwrite=True)
```

Example

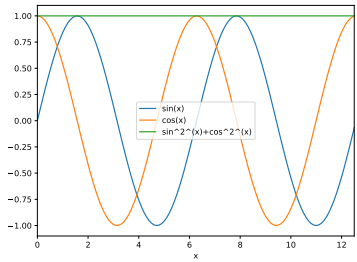
big_news.pdf

Hello World

Xinyi Li

2019-12-19

Sine Plot



Sine Table

x	sin(x)	cos(x)	$\sin^2(x)+\cos^2(x)$
6.6	0.31	0.95	1
5.6	-0.63	0.78	1
7.3	0.85	0.53	1
5.9	-0.37	0.93	1
4.7	-1	-0.01	1

Sine Function

```
1 def sin_2x_and_cos_2x(x):
2     y = np.sin(x) * np.sin(x) + np.cos(x) *
          np.cos(x)
3     return y
```

Features

Snapshot

```
1 Journalist.report(overwrite=False)
```

- it is why use `.pdf` instead of `.html` or raw `.md`
- add a timestamp at the end of output filename, such as `1_prophet_report_2019-12-18_22:06:18.pdf`

Generate Template

```
1 Journalist.generate_template()
```

- after `hear` method
- generate template with **each** variable on subsection/slide page according its type: `var_name` as title, `value` as page content
- slight modification for usage