Reporting Differently, Thank Markdown - Demo Report

Ottmar Gobrecht

2016-11-13

This is a demo report to show the current possible chart types with Markdown Reporter[1]*. We use the population development of New York, Rio and Tokio as example data.*

The line chart and the area charts have captions. The prefix ”Figure x:” is automatically placed by the LaTeX engine when you define a figure. Since you use Markdown you do not define the figures by yourself – this is done automatically by Pandoc[2], when you place an image link in your text – and the charts are in fact images. LaTeX is using so called Floats[3] to place figures in the text stream. The thing is, that depending on the size of a figure, you have no guarantee, that a figure is placed where you put your links or code blocks for the charts – LaTeX is using the best place in respect of the figure size and the typographic layout.

You can define a caption in your code block like so: `{.sql .chart .line caption="Line Chart"}`

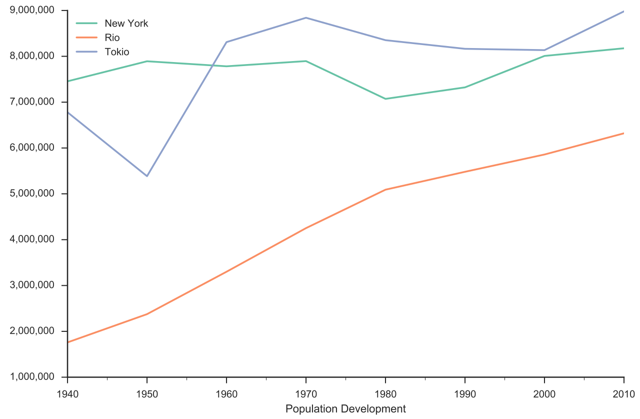


Figure 1: Line Chart

The area chart is per default stacked in Python matplotlib. To be consistent with the chart key words, Markdown Reporter uses always the base keyword for an unstacked chart (area, bar, barh) and the postfix _stacked for an stacked chart (area_stacked, bar_stacked, barh_stacked).

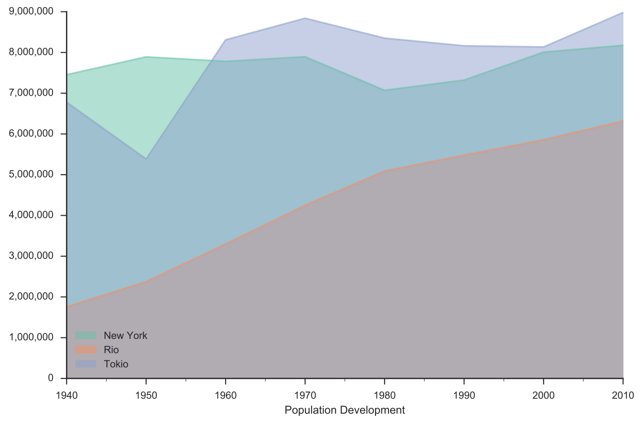


Figure 2: Area Chart

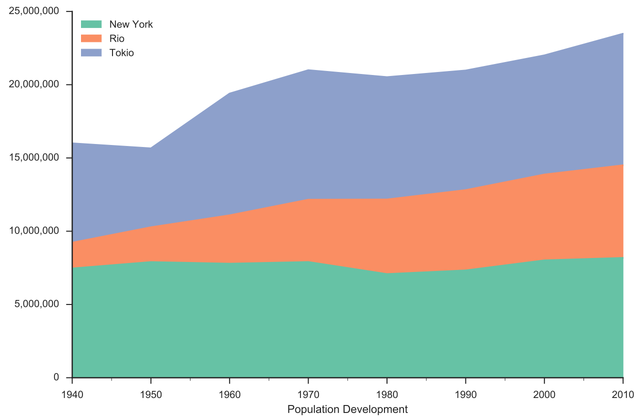


Figure 3: Area Chart Stacked

In LaTeX a new page can be forced with the command `\newpage`. Pandoc is accepting this command and passes it to output formats who understand this. We use it now :-)

[1] https://github.com/ogobrecht/markdown-reporter

[2] http://pandoc.org/MANUAL.html#extension-implicit_figures

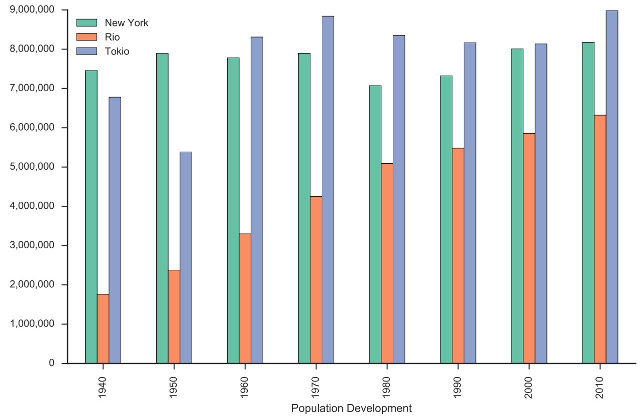
[3] https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions

Markdown Reporter has by default these options set for numeric value formatting:

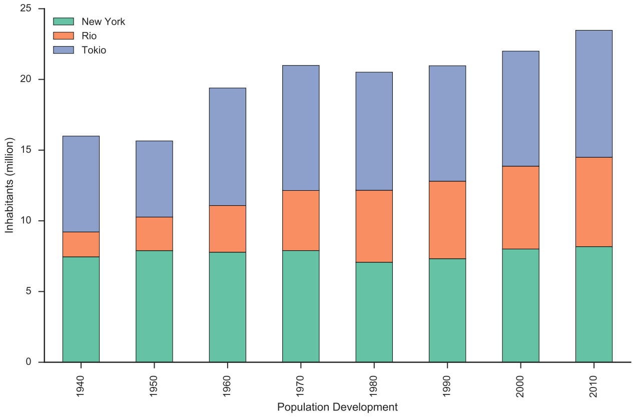
- `numlang=en`
- `numformat=%.0f`
- `numgrouping=true`

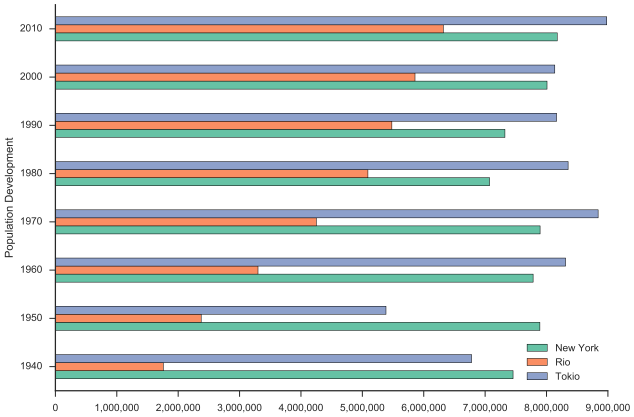
You can of course overwrite these options. The `numformat` is a Python format mask[4].

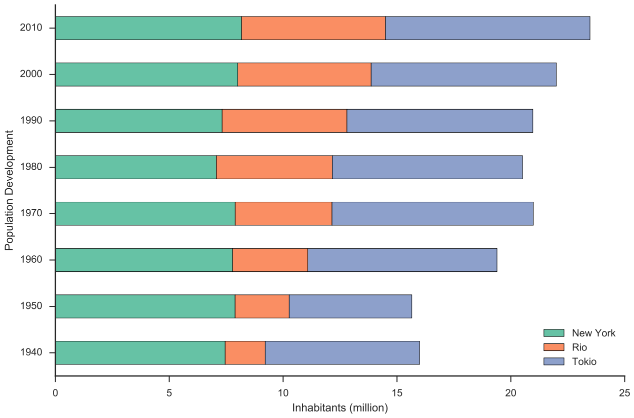
If you need to format your date values then you can use Pythons date and time format mask[5] like `dateformat="%Y-%m-%d"` for a ISO date. The default in Markdown Reporter is `dateformat="%Y"`.



We deliver an extra y label and values in millions for the stacked bar chart: `ylabel="Inhabitants (million)"`.







Another inline usage of `\newpage` here :-)

[4] https://pyformat.info/#number

[5] http://strftime.org/

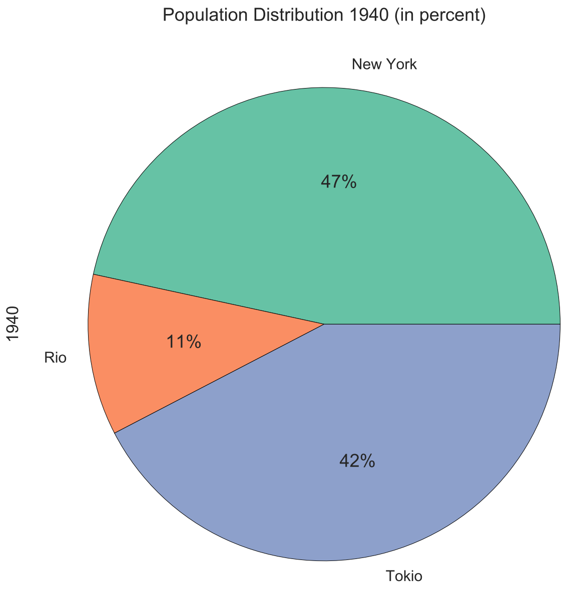
The special thing on an pie chart is, that you have to deliver the data in a different format – you transpose it from a vertical into a horizontal representation. Pandas has also some possibilities[6] to reshape our data – but since we focus here on easy, generic chart generation this is no option for us. On a pie chart you can set the option `y` to define which column in your data should be used for the pie chart - we use here `y=0` (the default, if you omit this option) for the first (1940) and `y=1` (2010) for the second pie chart.

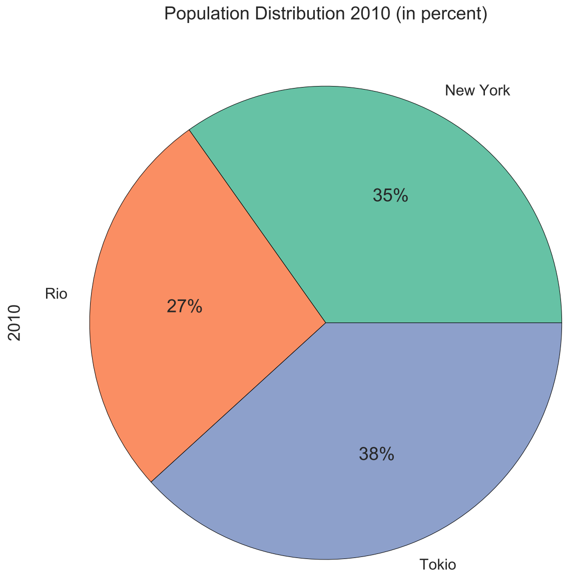
Pie chart data:

```
Population Development,1940,2010
New York,7454995,8175133
Rio,1759277,6320446
Tokio,6778804,8980768
```

All other charts data:

```
Population Development,New York,Rio,Tokio
1940,7454995,1759277,6778804
1950,7891957,2375280,5385071
1960,7781984,3300431,8310027
1970,7895563,4251918,8840942
1980,7071639,5090723,8351893
1990,7322564,5480768,8163573
2000,8008278,5857904,8134688
2010,8175133,6320446,8980768
```





On the pie chart we use also the title attribute, since this chart type has no axes to place our index header: `title="Population Distribution (in percent)"`. Additionally we set the format mask for the percent values, which are calculated by matplotlib: `autopct=%.0f%%`. This is a Python format mask[7] and Markdown Reporter uses per default `autopct=%.1f%%`.

[6] http://pandas.pydata.org/pandas-docs/stable/reshaping.html

[7] https://pyformat.info/#number