

# **MPLslides 0.1**

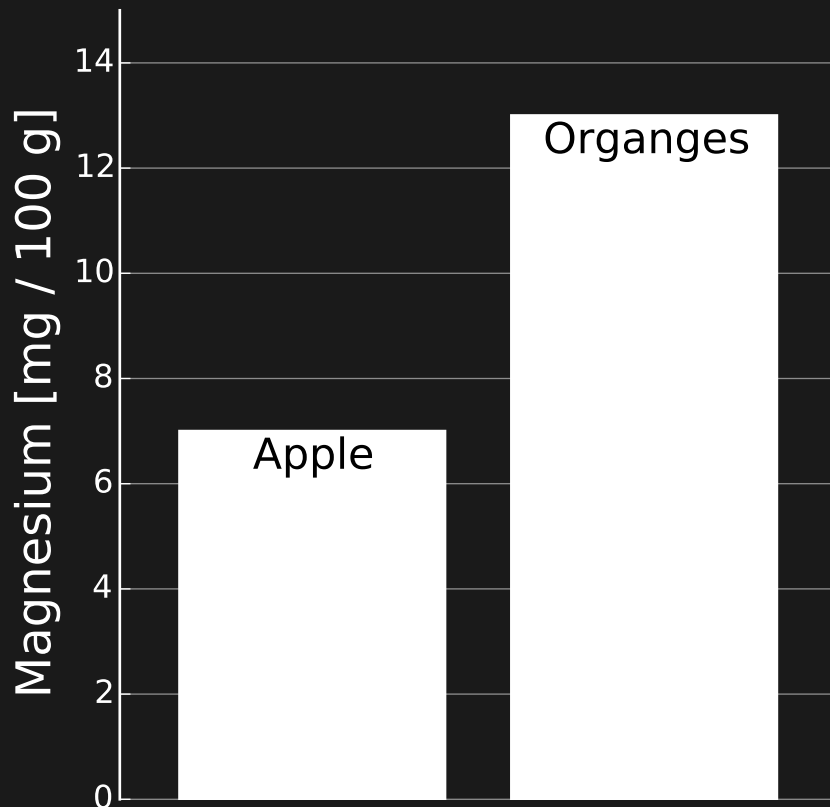
Making presentations in matplotlib

Press left and right to navigate slides



# Slides with Matplotlib?

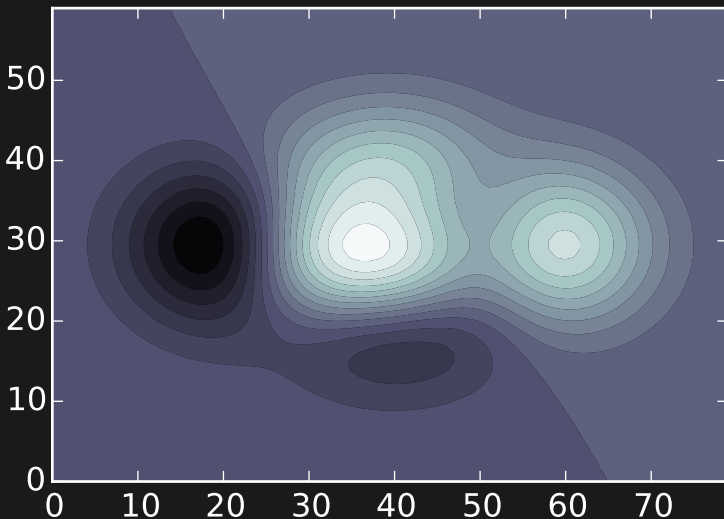
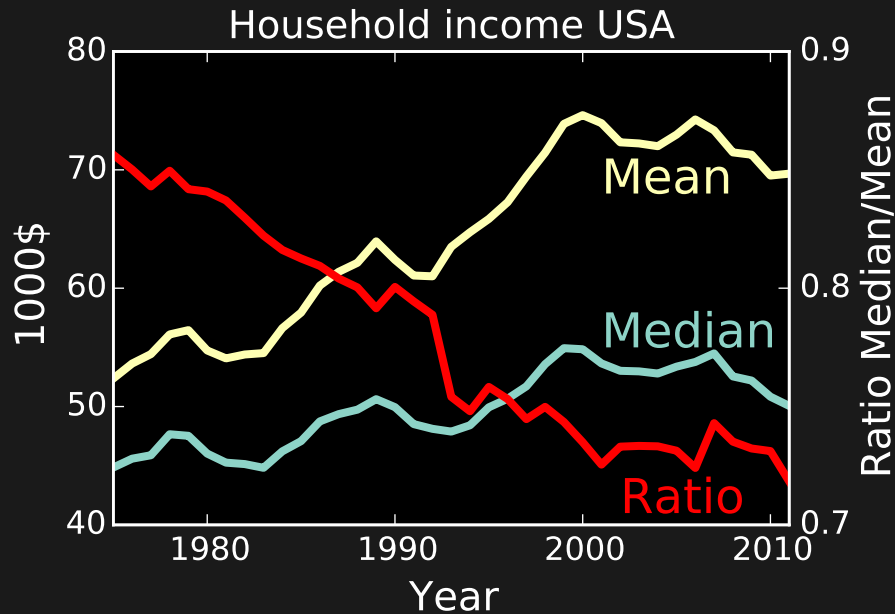
Why would you do that?



- For fun!
- Problematic alternatives.
- All my figures are done with mpl.

# Features

Almost none!



- Pictures!
- Axes!
- Enumerated text!
- ..which was harder than its sounds.
- $e^{\pm i\theta} = \cos\theta \pm i\sin\theta$
- PDF export!

# Code of this slide

Sorry, no code formatting!

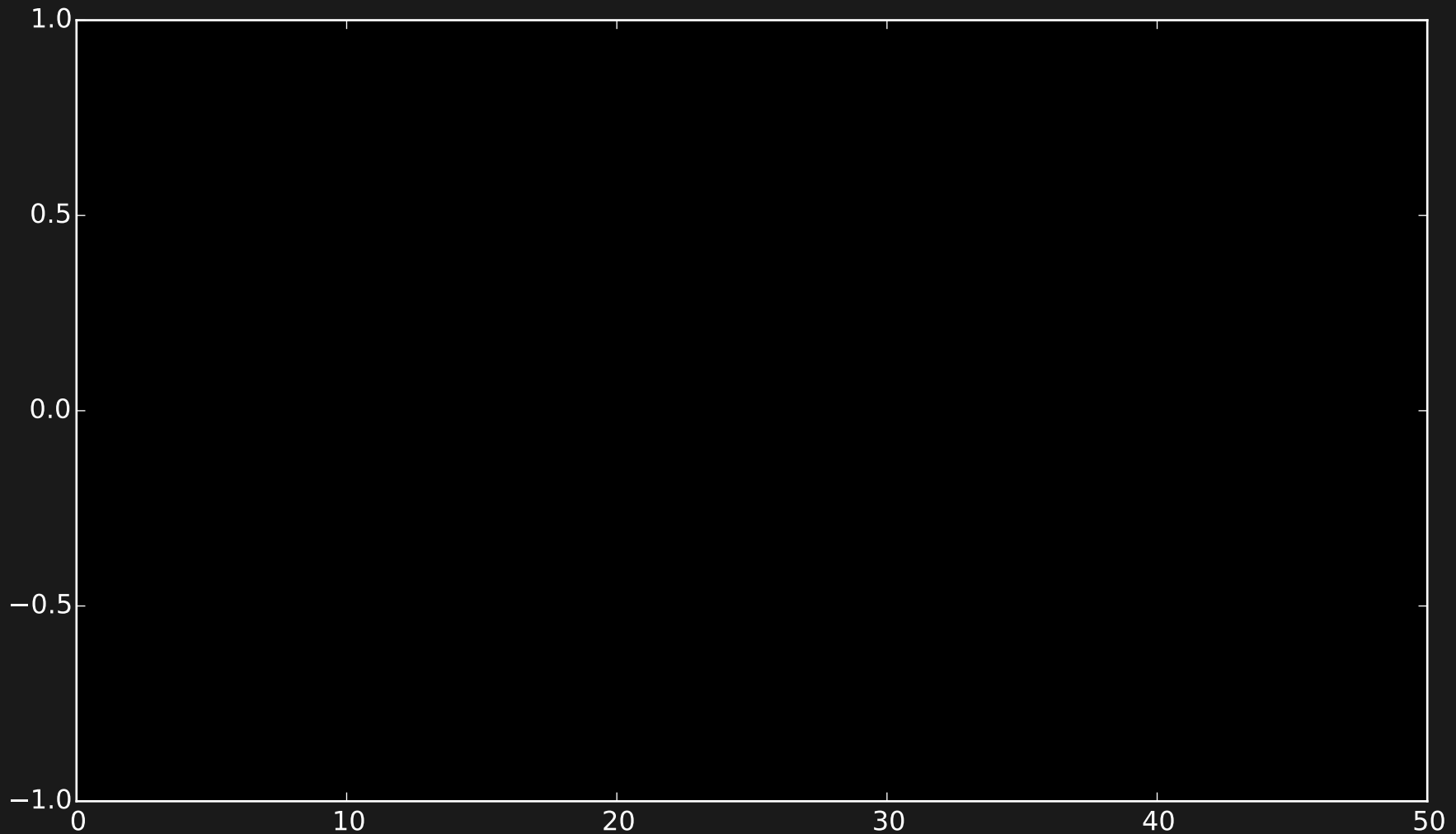
```
from content import add_text
from presentation import Presentation
from slides import NormalSlide

pres = Presentation()
presentation.background.append(add_image('mpl_slide.png', ha='right',
                                         pos=(0.98, 0.02), zoom=0.4))

...
title = 'Code of this slide'
subtitle = 'Sorry, no code formatting!'
slide = NormalSlide(title, subtitle)
txt = ...the shown txt...
fontprops = dict(va='top', fontname='monospace', fontsize=15)
code_txt = add_text(txt, (0.1, 0.7), **fontprops)
slide.add_content(code_txt)
presentation.add_slide(s)
```

# Animated Plots

Only in interactive Backends



# What sucks?

And why?

- Matplotlib's typesetting is very basic.
- Manual newlines.
- No `<b>`, `<i>` or other inline formatting.
- Layout options are still very basic.
- But should be easily extensible.

# What can be done?

Probably not by me

- One could directly use Latex for text processing.
- Thats cheating!  
Also why not use Beamer than...
- Make a fancier text class, with support for simple formatting.
- AUTOMATIC LINE BREAKS!
- Appearing and disapperaing elements should also be quite doable.
- More content fuctions, smarter Layout.

# Personal conclusion

For the amount of code, it is quite awesome!