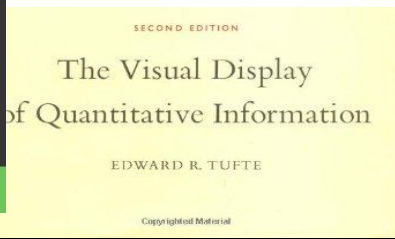
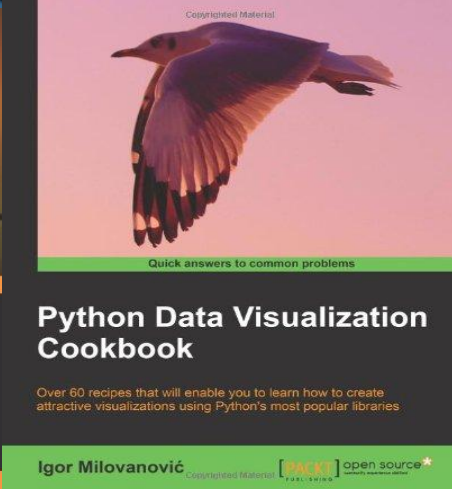
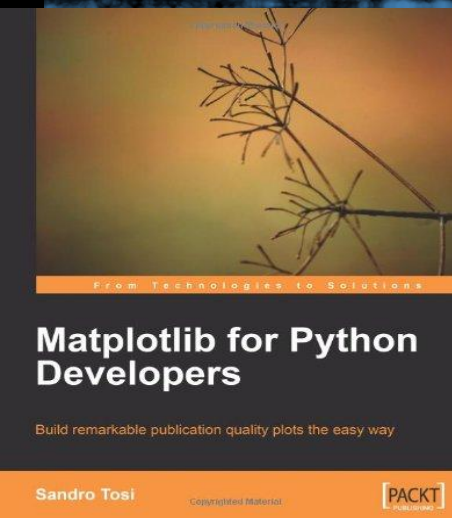
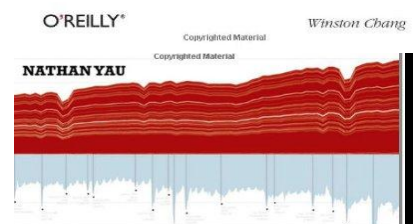
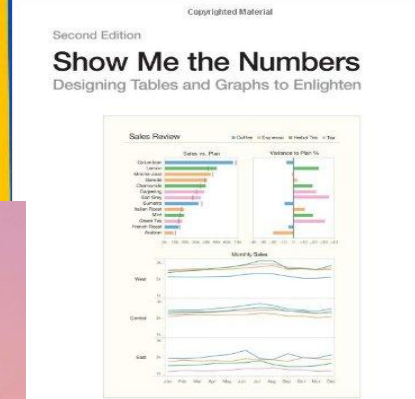
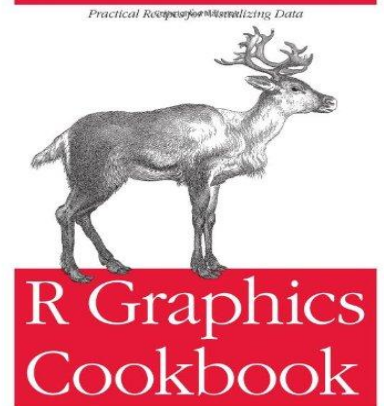
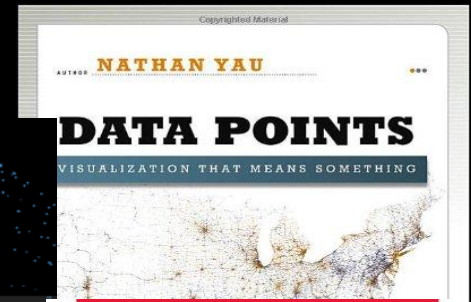
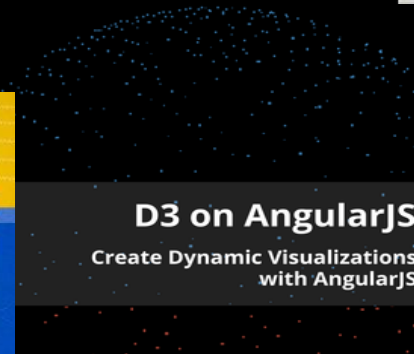
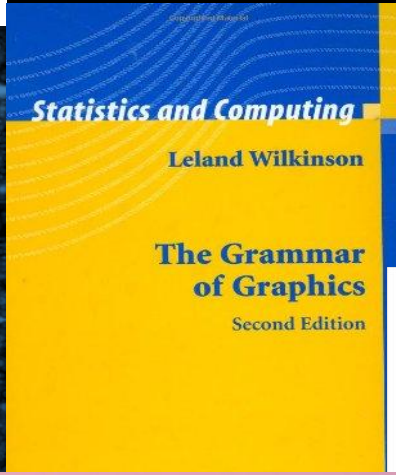
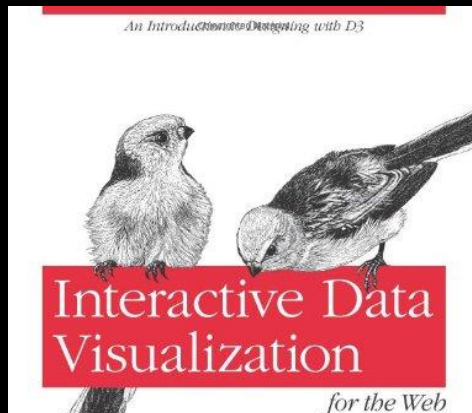


APIs de Visualização em Python

TDC 2014

Wilson Freitas
<http://aboutwilson.net>
@aboutwilson

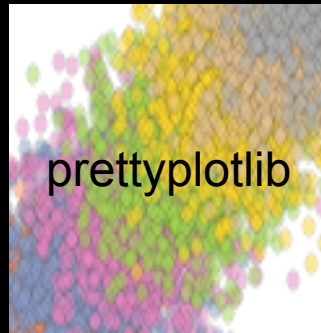
Visualização



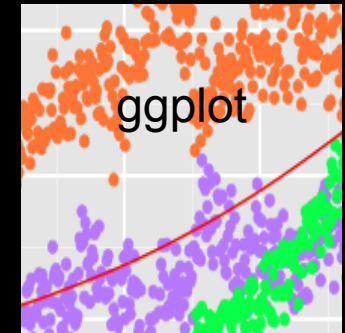
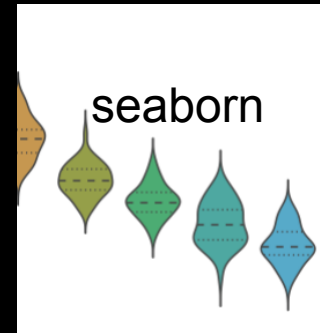
What matters to statistical users?

- Standard suite of linear algebra, matrix operations (NumPy, SciPy)
- Availability of statistical models and functions
 - More than there used to be, but nothing compared to R / CRAN
 - **rpy2** is coming along, but it doesn't seem to be an “end-user” project
- Data visualization and graphics tools (matplotlib, ...)
- Interactive research environment (IPython)

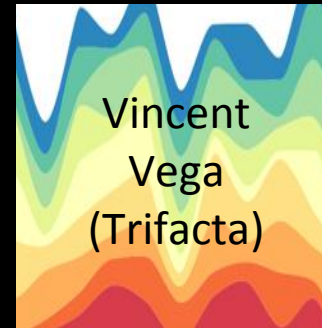
Visualização + Python



Pandas Support



Javascript Support



bokeh

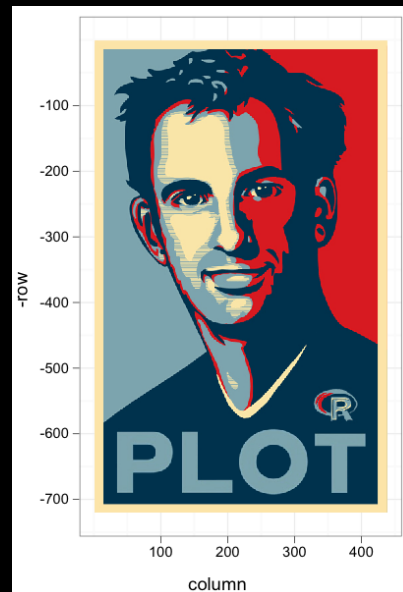
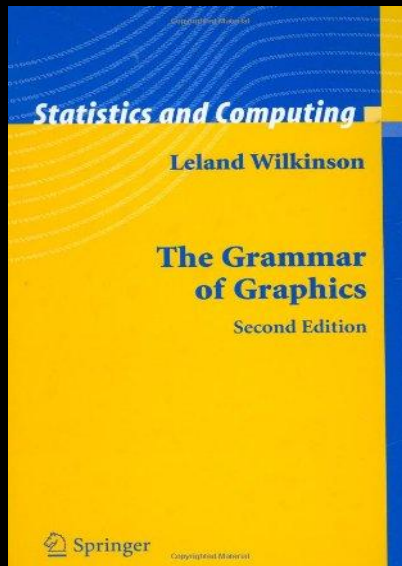
ggplot



Wes McKinney Mod → Tim · 2 years ago

I definitely agree with you there. ggplot2 is awesome (!). You *can* make attractive graphics with matplotlib but it definitely requires a lot of tweaking / customization. I'm hopeful that a kind soul will put some work into implementing the Grammar of Graphics for Python (ggpy anyone?). We shall see

1 ^ | v Reply Share ›



\hat{y} hat

seaborn

- Biblioteca de visualização baseada no matplotlib

"seaborn is a library for making attractive and informative statistical graphics in Python "

- Boa para análise exploratória de dados
- Fortemente integrada com pandas e numpy

mpld3

" Bringing Matplotlib to the Browser "

- Integra matplotlib com D3js
- Cria visualização de dados interativa no navegador



Vincent

" A Python to Vega translator "



" Vega is a visualization grammar, a declarative format for creating, saving and sharing visualization designs. "

- Estruturas de dados Python
- Visualização em JavaScript
- Integração com pandas





Bokeh

- Biblioteca de visualização interativa
- Integração com pandas
- Foco: explorar a visualização dos navegadores



CONTINUUM
ANALYTICS

datos: Top 100 Wines of the World 2014



WAWWJ

ASOCIACIÓN MUNDIAL DE
PERIODISTAS Y ESCRITORES
DE VINOS Y LICORES



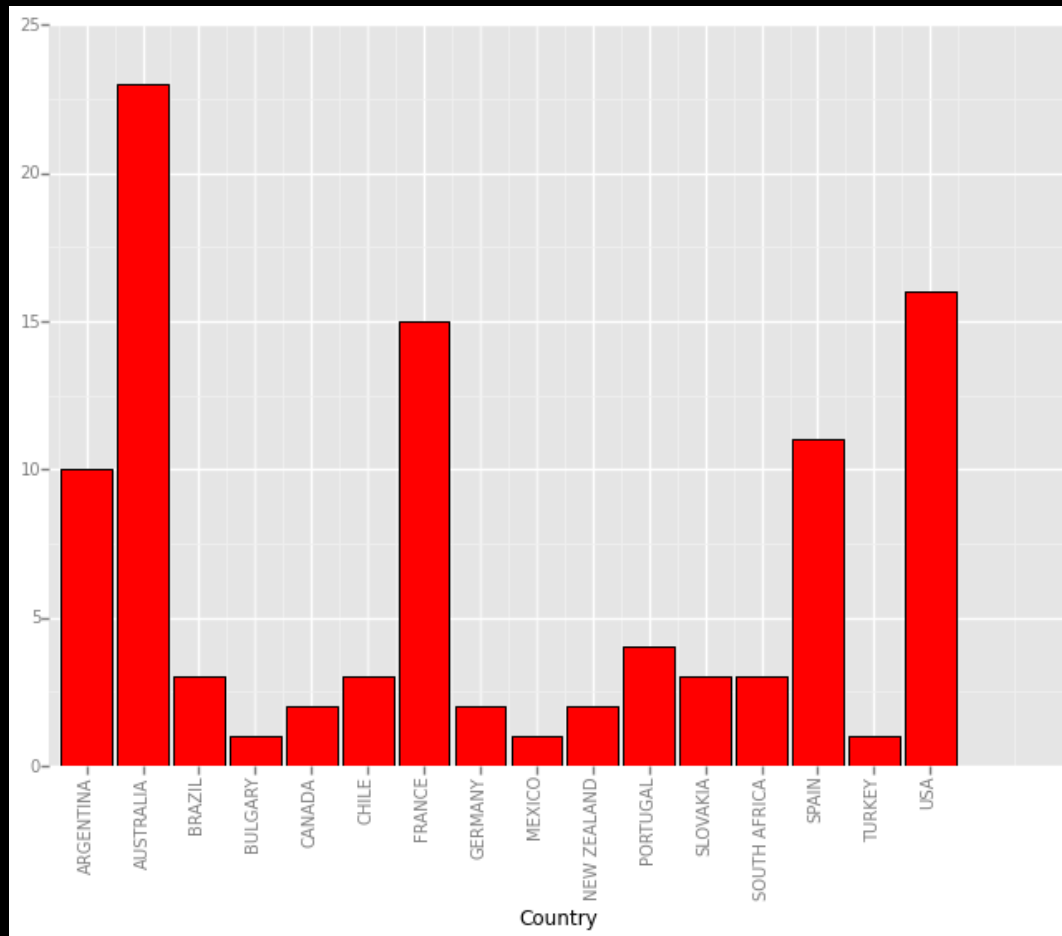
DataFrame pandas: wty

```
In [50]: wty.head(15)
```

```
Out[50]:
```

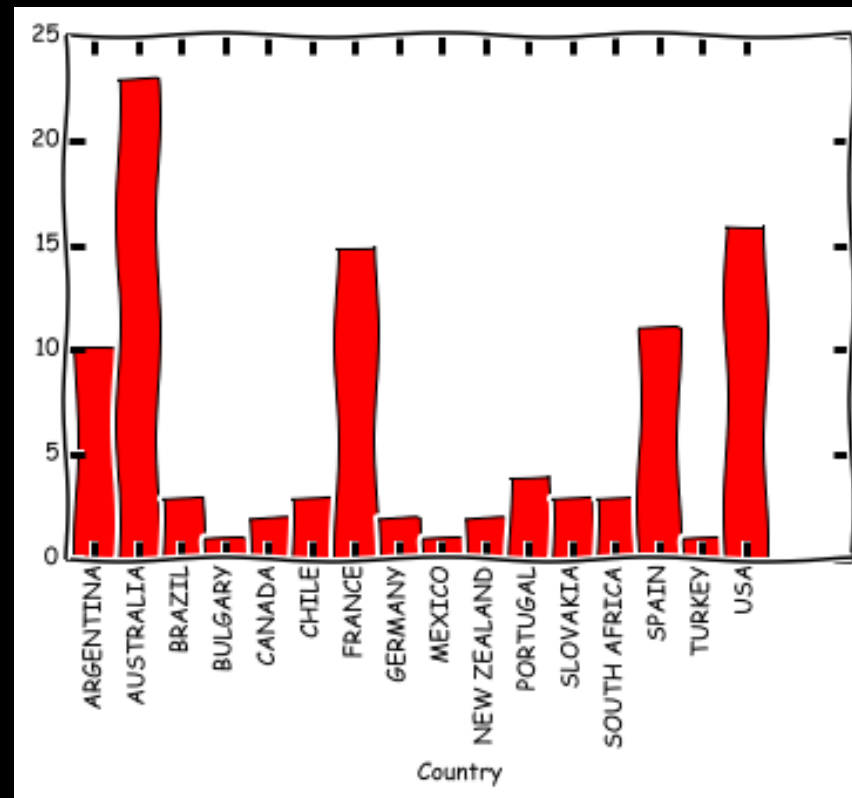
	Brand	Wine	Type	Country	Prizes	Points
ID						
1	VRANKEN POMMERY MONOPOLE HEIDSIECK	Charles Heidsieck Blanc des Millénaires 1995	CHAMPAGNE	FRANCE	8	329.50
2	BODEGA ZEMLIA DE LAS CASUARINAS S.A.	Zemlia Himno Malbec Bicentenario 2010	MALBEC	ARGENTINA	11	314.91
3	TAYLORS / WAKEFIELD WINES PTY LTD	Taylors St Andrews Shiraz Clare Valley 2010	SYRAH/SHIRAZ	AUSTRALIA	13	297.33
4	PELLER ESTATES WINERY	Peller Estates Niagara-On-The-Lake Riesling Ic...	ICEWINE/EISWEIN	CANADA	11	293.25
5	JACOBS CREEK PERNOD RICARD PACIFIC	Jacobs Creek Centenary Hill Barossa Shiraz 2009	SYRAH/SHIRAZ	AUSTRALIA	9	273.13
6	GATT WINES	Gatt Shiraz 2010	SYRAH/SHIRAZ	AUSTRALIA	12	271.25
7	VRANKEN POMMERY MONOPOLE HEIDSIECK	Champagne Piper Heidsieck Rare Millésimé 2002	CHAMPAGNE	FRANCE	8	267.50
8	GATT WINES	Gatt Shiraz 2008	SYRAH/SHIRAZ	AUSTRALIA	8	258.25
9	BODEGAS LUIS CAÑAS SA	Luis Cañas Reserva Selección De La Familia Doc...	TEMPRANILLO	SPAIN	7	256.38
10	VRANKEN POMMERY MONOPOLE HEIDSIECK	Champagne Piper Heidsieck Brut	CHAMPAGNE	FRANCE	11	255.50
11	GATT WINES	Gatt Cabernet Sauvignon 2009	CABERNET SAUVIGNON	AUSTRALIA	10	248.38
12	VRANKEN POMMERY MONOPOLE HEIDSIECK	Champagne Charles Heidsieck Brut Reserve	CHAMPAGNE	FRANCE	8	236.75
13	VRANKEN POMMERY MONOPOLE HEIDSIECK	Champagne Piper Heidsieck Brut Millesime 2006	CHAMPAGNE	FRANCE	9	235.00
14	VRANKEN POMMERY MONOPOLE HEIDSIECK	Charles Heidsieck Brut Rosé Millésimé 1999	CHAMPAGNE	FRANCE	5	230.75
15	GATT WINES	Gatt High Eden Shiraz 2009	SYRAH/SHIRAZ	AUSTRALIA	10	229.25

Quantos vinhos por País?



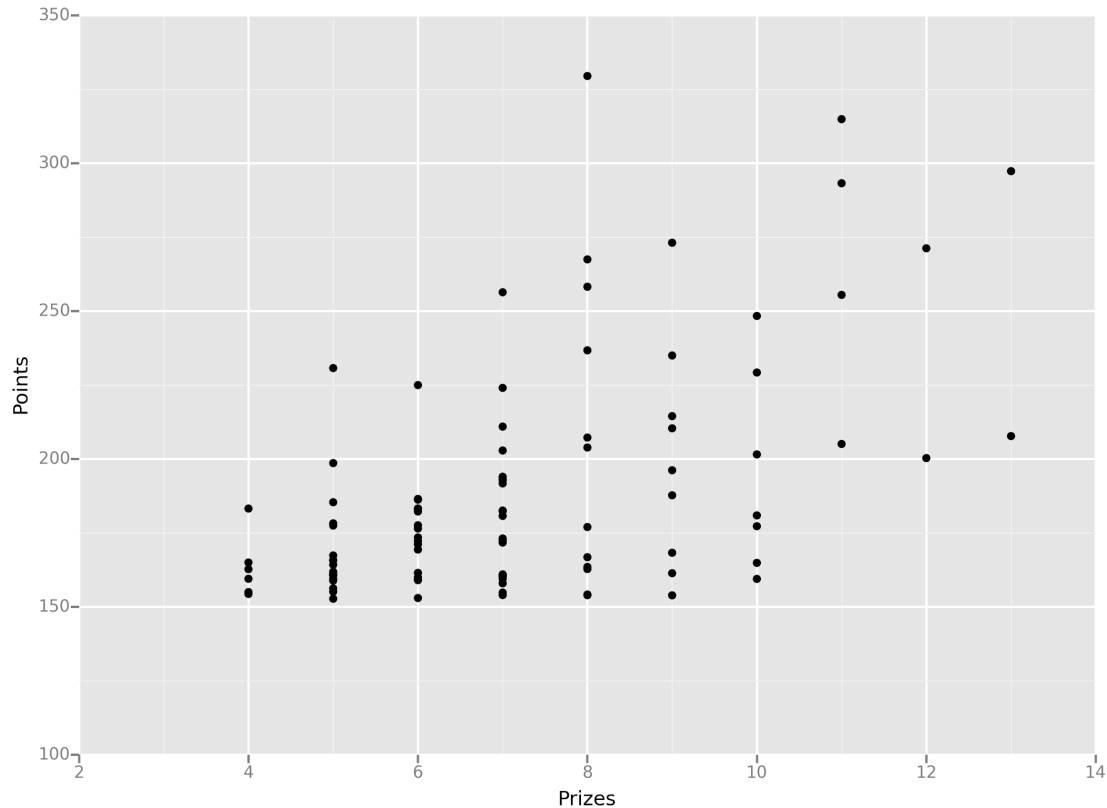
```
ggplot(wty, aes("Country")) + geom_bar(fill="red", colour="black") +  
  theme(axis_text_x=element_text(angle=90))
```

Quantos vinhos por País? (com estilo)



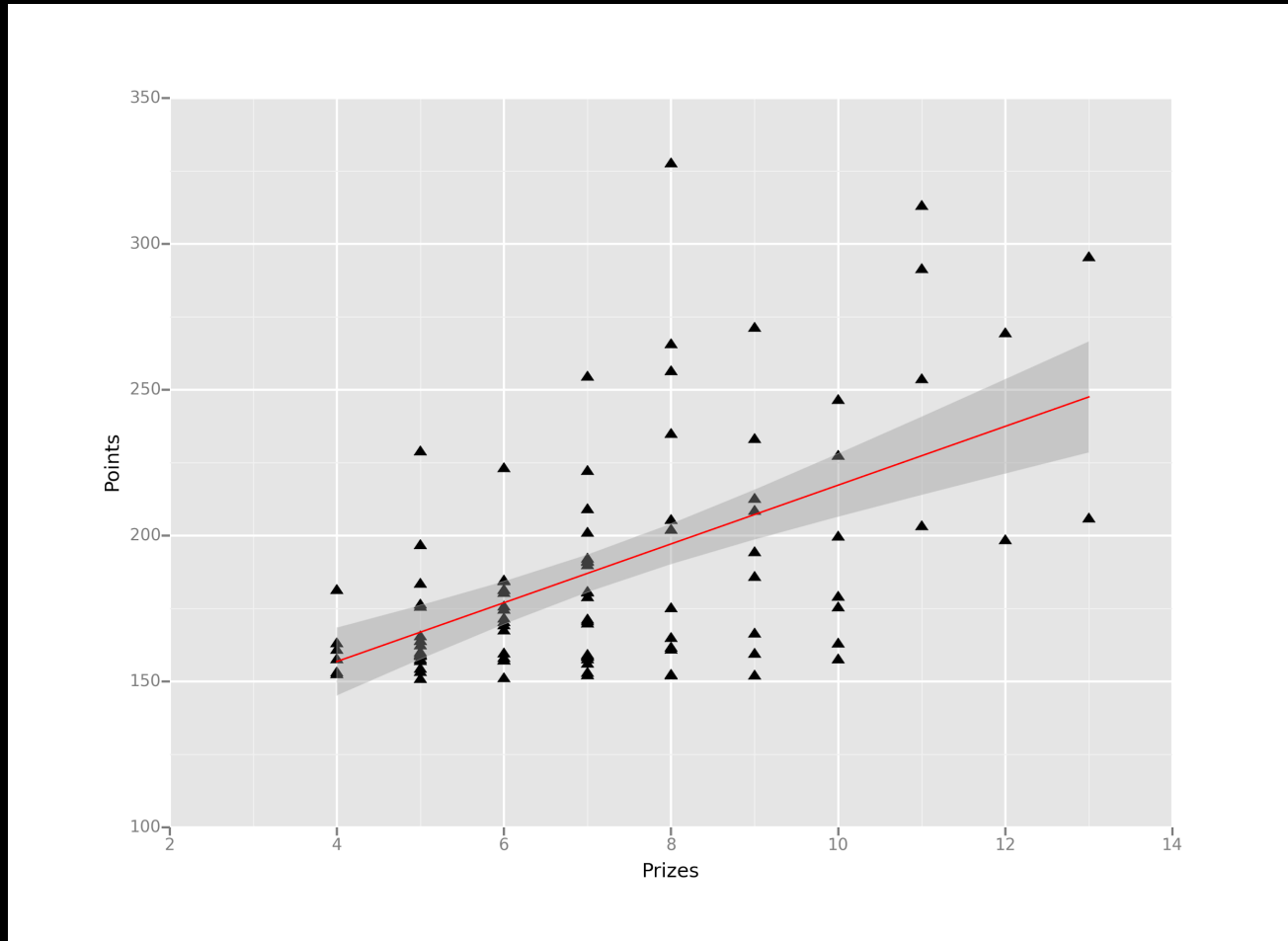
```
ggplot(wty, aes("Country")) + geom_bar(fill="red", colour="black") +  
  theme_xkcd() + theme(axis_text_x=element_text(angle=90))
```

Qual a relação entre Pontos e Prêmios?



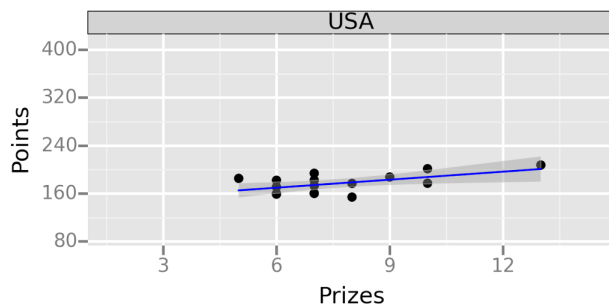
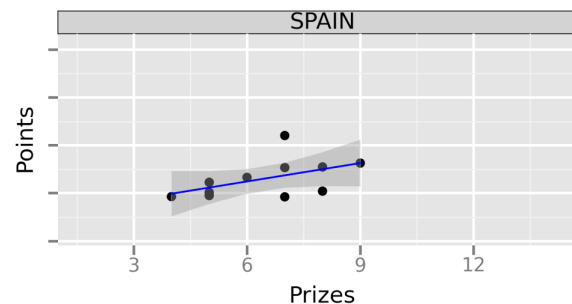
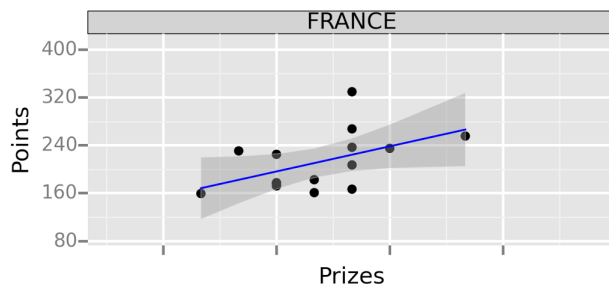
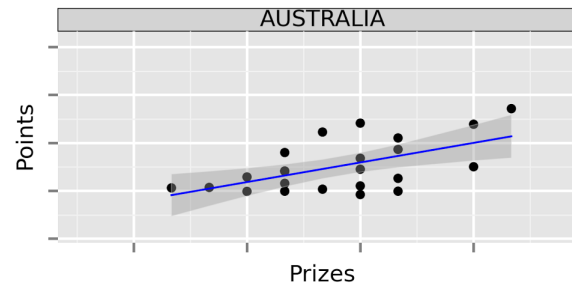
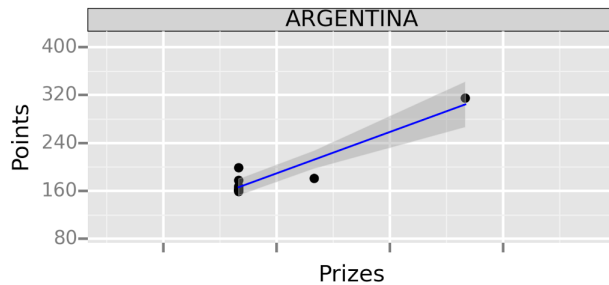
```
ggplot(wty, aes(x='Prizes', y='Points')) + geom_point()
```

Qual a relação entre Pontos e Prêmios?



```
ggplot(wty, aes(y="Points", x="Prizes")) + geom_point(size=60, shape=6) +  
  stat_smooth(method='lm', colour='red')
```

Qual a relação entre Pontos e Prêmios por País?



```
idx = (wtv.Country == 'AUSTRALIA') |  
      (wtv.Country == 'FRANCE') |  
      (wtv.Country == 'SPAIN') |  
      (wtv.Country == 'USA') |  
      (wtv.Country == 'ARGENTINA')
```

```
gplot(wtv[idx], aes(x="Prizes", y="Points")) +  
  geom_point() +  
  stat_smooth(method='lm', color='blue') +  
  facet_wrap("Country", scales="fixed")
```


Como se distribuem os prêmios por País?

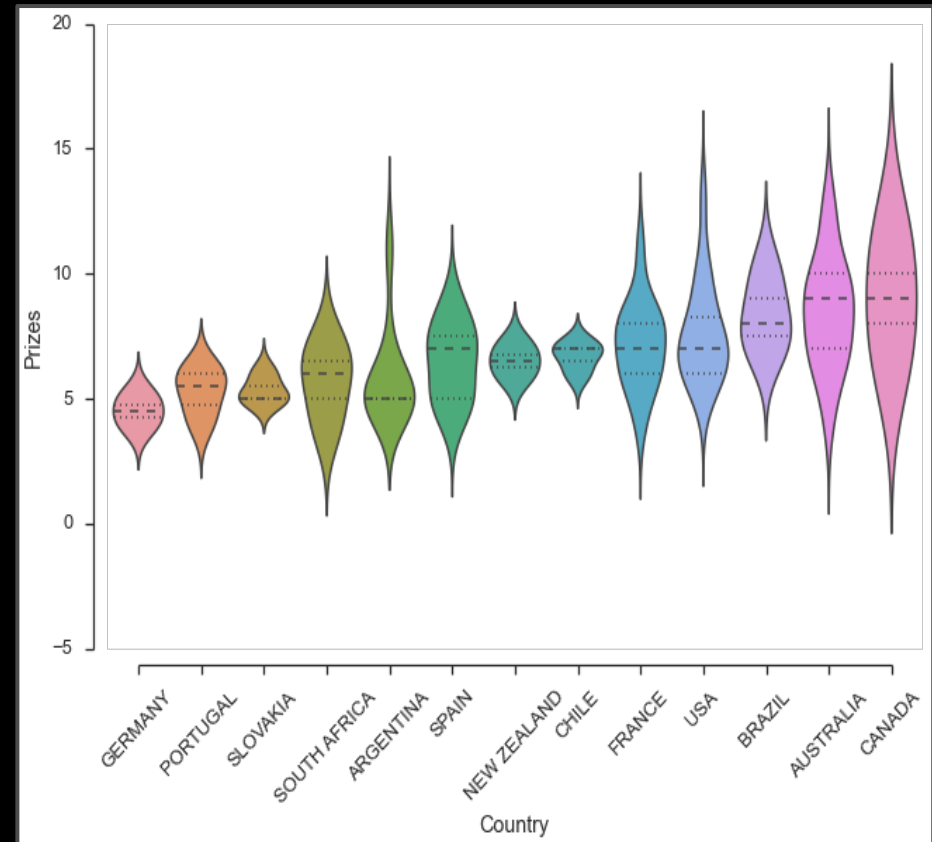
```
wtv_country_prizes = wtv['Prizes'].groupby(wtv['Country'])
idx = wtv_country_prizes.count() == 1
wtv_t = wtv.ix[[c not in idx[idx].index for c in wtv.Country], \
               ['Country', 'Prizes']]
wtv_t_country_mean = wtv_t.groupby(wtv_t.Country).mean()
country_names = wtv_t_country_mean.Prizes.sort(inplace=False).index

import seaborn as sb

sb.set(style="ticks")
f, ax = plt.subplots()
sb.offset_spines()

sb.violinplot(wtv_t['Prizes'], wtv_t['Country'], order=country_names)

locs, labels = plt.xticks()
plt.setp(labels, rotation=45)
sb.despine(trim=True)
```



Mas qual País possui o melhor vinho?

```
wty_country = wty.groupby('Country')
country_count = wty_country['Wine'].aggregate(len)
country_count.sort(inplace=True)
```

```
fig, ax = plt.subplots(figsize=(12,8))
```

```
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.get_xaxis().tick_bottom()
ax.get_yaxis().tick_left()
```

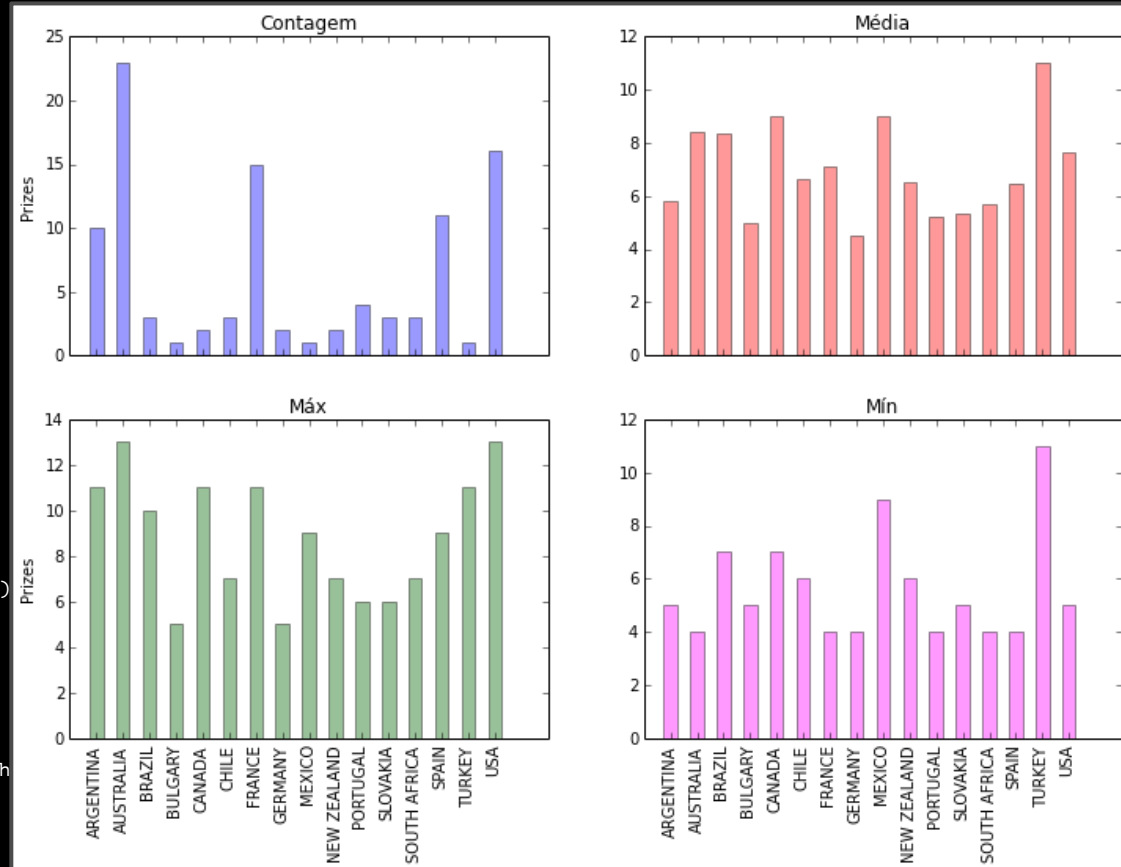
```
plt.subplot(221)
g = wty_country['Prizes'].count()
x = np.arange(len(g))
countries = list(g.index.values)
```

```
y = g.values
plt.bar(x+1, y, align='center', width=0.5, alpha=0.4)
plt.xticks(x+1, [' ']*10, rotation=90)
plt.title('Contagem')
plt.ylabel('Prizes')
```

```
plt.subplot(222)
g = wty_country['Prizes'].mean()
y = g.values
plt.bar(x+1, y, align='center', color='red', width=0.5, alpha=0.4)
plt.xticks(x+1, [' ']*10, rotation=90);
plt.title(u'Média')
```

```
plt.subplot(223)
g = wty_country['Prizes'].max()
y = g.values
plt.bar(x+1, y, align='center', color='darkgreen', width=0.5, alpha=0.4)
plt.xticks(x+1, countries, rotation=90);
plt.title(u'Máx')
plt.ylabel('Prizes')
```

```
plt.subplot(224)
g = wty_country['Prizes'].min()
y = g.values
plt.bar(x+1, y, align='center', color='magenta', width=0.5, alpha=0.4)
plt.xticks(x+1, countries, rotation=90);
plt.title(u'Mín')
```



mpld3 salva fig em html

```
>>> mpld3.save_html(fig, 'fig.html')
```

```
<div id="fig_el672345700473765832459779"></div>
```

```
<script>
```

```
function mpld3_load_lib(url, callback){  
  var s = document.createElement('script');  
  s.src = url;  
  s.async = true;  
  s.onreadystatechange = s.onload = callback;  
  s.onerror = function(){console.warn("failed to load library " + url);};  
  document.getElementsByTagName("head")[0].appendChild(s);  
}
```

```
if(typeof(mpld3) !== "undefined" && mpld3._mpld3IsLoaded){  
  // already loaded: just create the figure  
  !function(mpld3){  
    ...  
  }  
}
```

dados: Ações (Quandl)

```
import Quandl
```

```
tickers = [  
    "GOOG/BVMF_BOVA11.4",  
    "YAHOO/SA_PETR3.6",  
    "GOOG/BVMF_VALE5.4",  
    "GOOG/BVMF_BBDC4.4",  
    "GOOG/BVMF_BBAS3.4",  
    "GOOG/BVMF_ITUB4.4"]
```

```
series = Quandl.get(tickers,  
    trim_start="2013-01-01",  
    trim_end="2013-12-31")
```

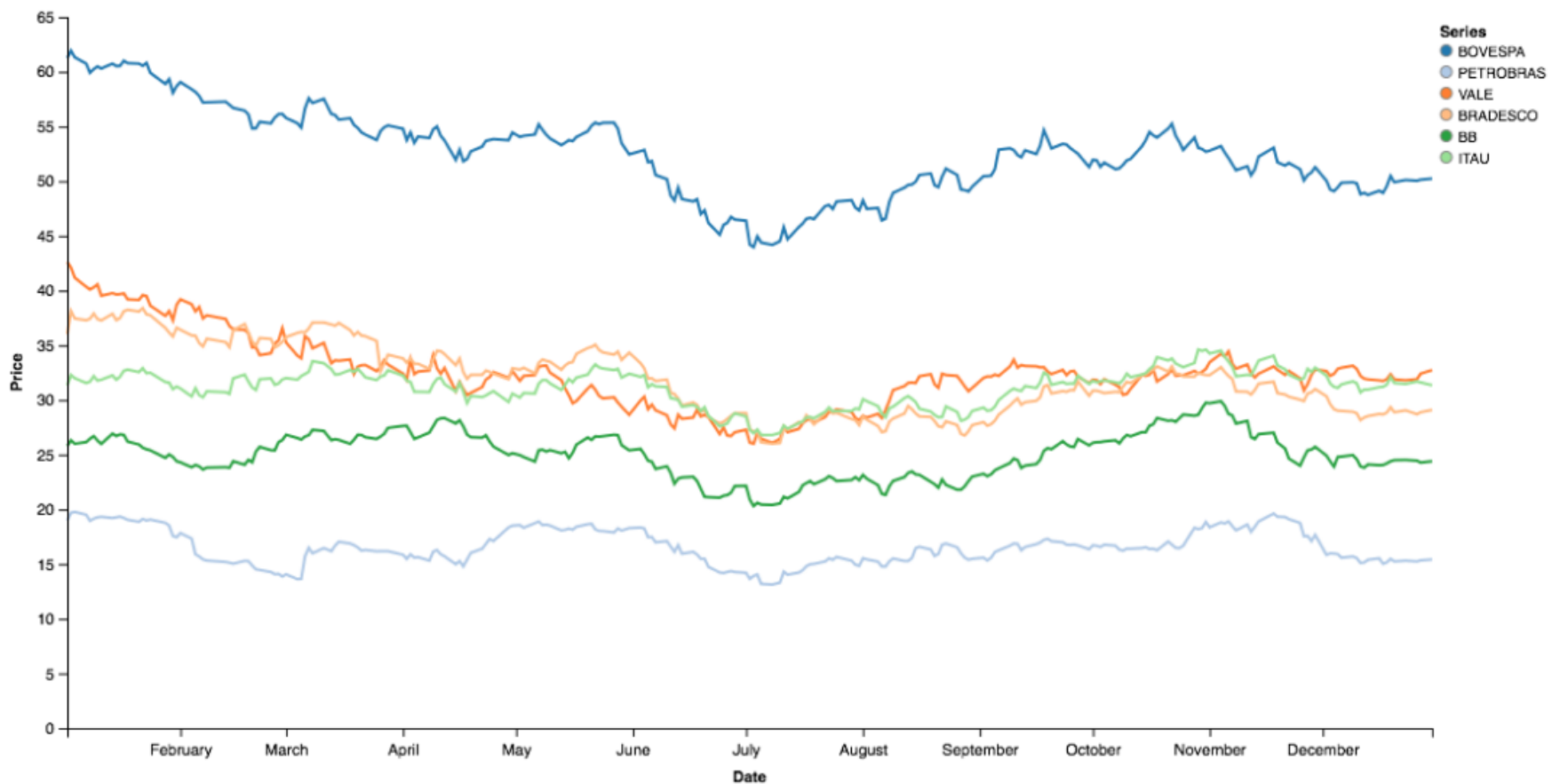
```
series.columns = [  
    "BOVESPA",  
    "PETROBRAS",  
    "VALE",  
    "BRADESCO",  
    "BB",  
    "ITAU"]
```

```
series.dropna(inplace=True)
```

```
>>> print series
```

	BOVESPA	PETROBRAS	VALE	BRADESCO	BB	ITAU
Date						
2013-01-02	61.27	19.00	42.60	36.02	25.80	31.3
2013-01-03	61.92	19.70	42.09	38.12	26.31	32.2
2013-01-04	61.32	19.77	41.17	37.45	26.00	31.9
2013-01-07	60.75	19.51	40.35	37.29	26.15	31.5
2013-01-08	59.91	18.96	40.10	37.42	26.45	31.6
2013-01-09	60.25	19.15	40.30	37.90	26.70	32.1
2013-01-10	60.49	19.27	40.57	37.40	26.30	31.7
2013-01-11	60.30	19.32	39.53	37.27	25.98	31.8
2013-01-14	60.75	19.21	39.77	37.88	26.93	32.2
2013-01-15	60.52	19.27	39.65	37.30	26.70	31.8
2013-01-16	60.54	19.34	39.68	37.49	26.86	32.1
2013-01-17	61.01	19.21	39.75	38.11	26.81	32.6
2013-01-18	60.79	19.05	39.20	38.25	26.20	32.7
2013-01-21	60.75	18.87	39.15	38.09	25.93	32.5
2013-01-22	60.55	19.12	39.59	38.40	25.64	32.9
...						

Visualização de Séries de Preço



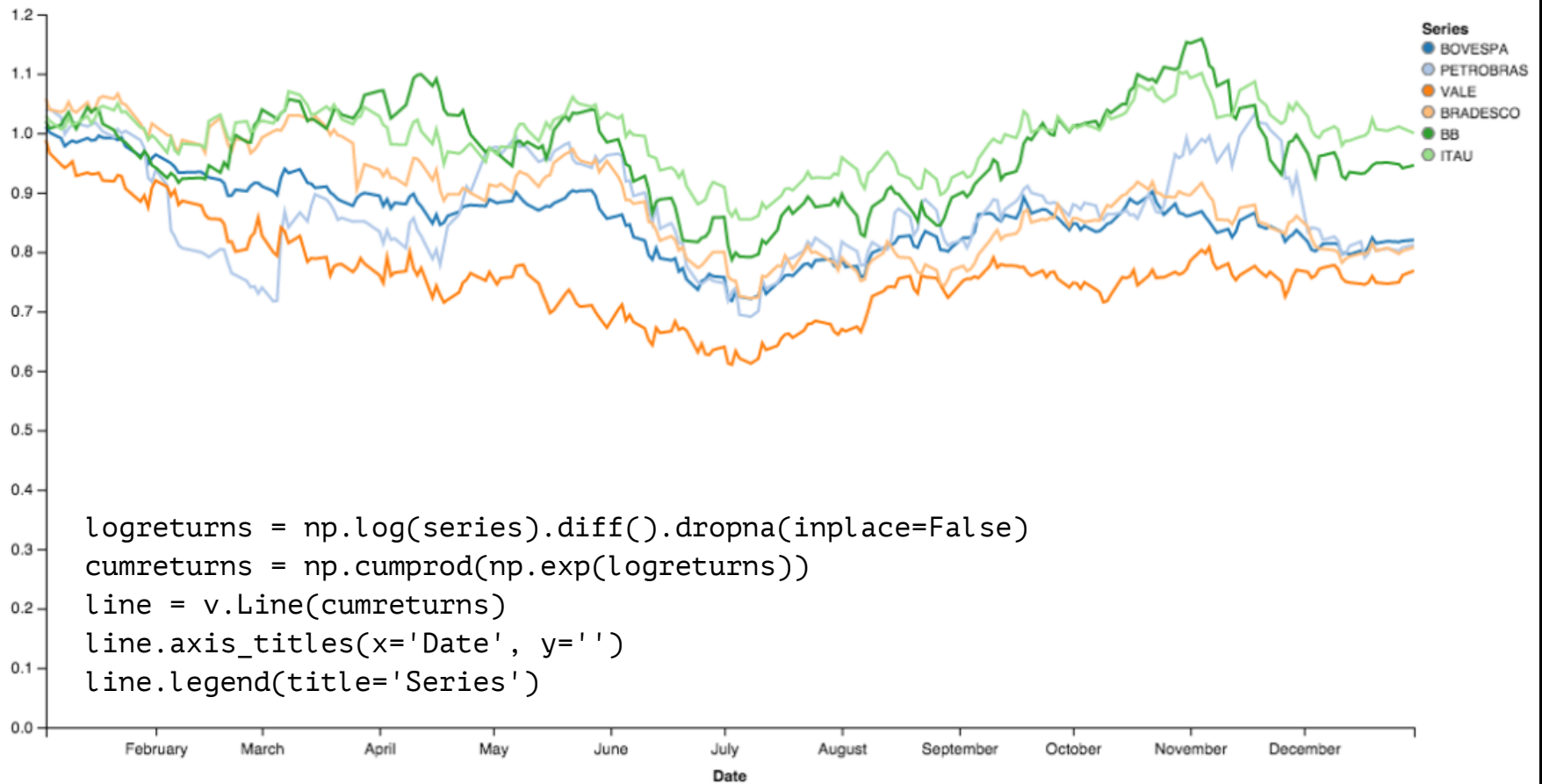
```
import vincent as v
line = v.Line(series)
line.axis_titles(x='Date', y='Price')
line.legend(title='Series')
```

Exporta a Gramática de Gráficos

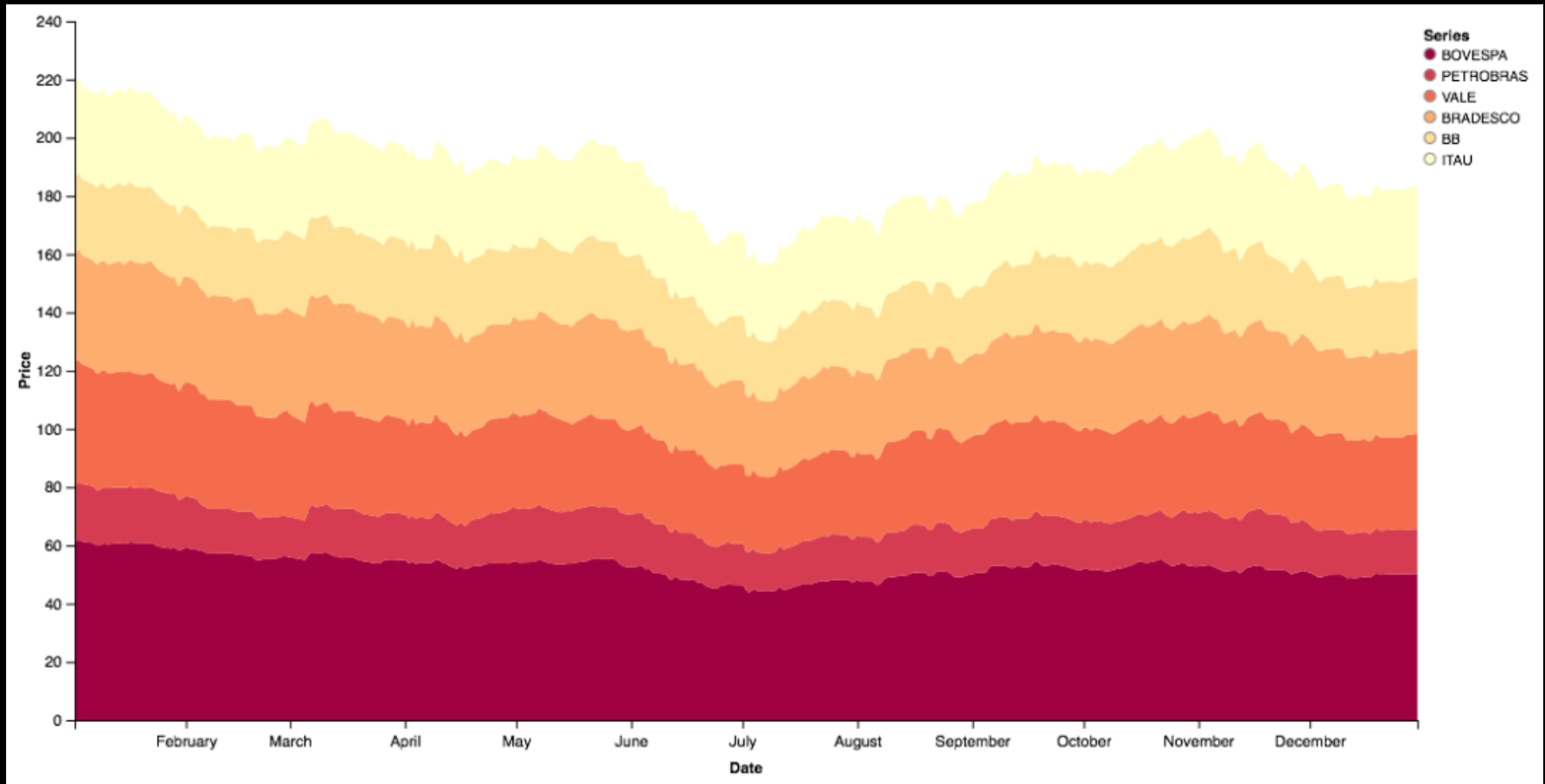
```
>>> print line.to_json()
```

```
{  
  "axes": [  
    { "scale": "x", "title": "Date", "type": "x" },  
    { "scale": "y", "title": "Price", "type": "y" }  
  ],  
  "data": [  
    {  
      "name": "table",  
      "values": [  
        { "col": "BOVESPA", "idx": 1357092000000, "val": 61.27 },  
        { "col": "PETROBRAS", "idx": 1357092000000, "val": 19.0 },  
        { "col": "VALE", "idx": 1357092000000, "val": 42.6 },  
        ...  
      ]  
    }  
  ]  
}
```

Variação acumulada



Séries de Preço Empilhadas



```
stacked = v.StackedArea(series)
stacked.axis_titles(x='Date', y='Price')
stacked.legend(title='Series')
stacked.colors(brew='Spectral')
```


Qual a correlação entre séries?

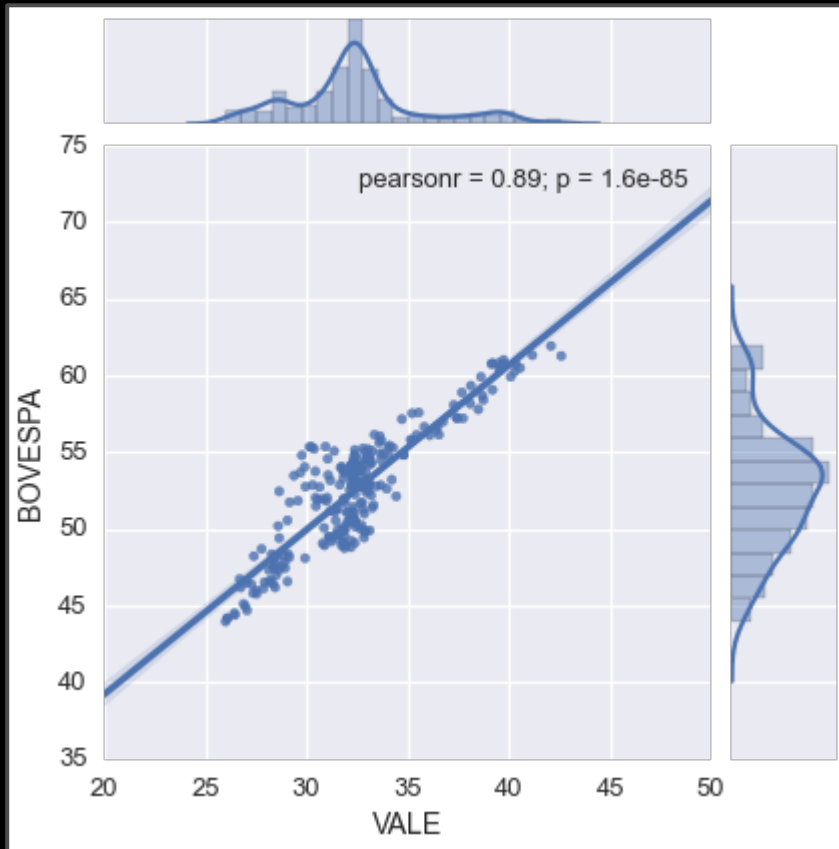
com `seaborn`

```
sb.heatmap(np.log(series).diff().dropna())
```

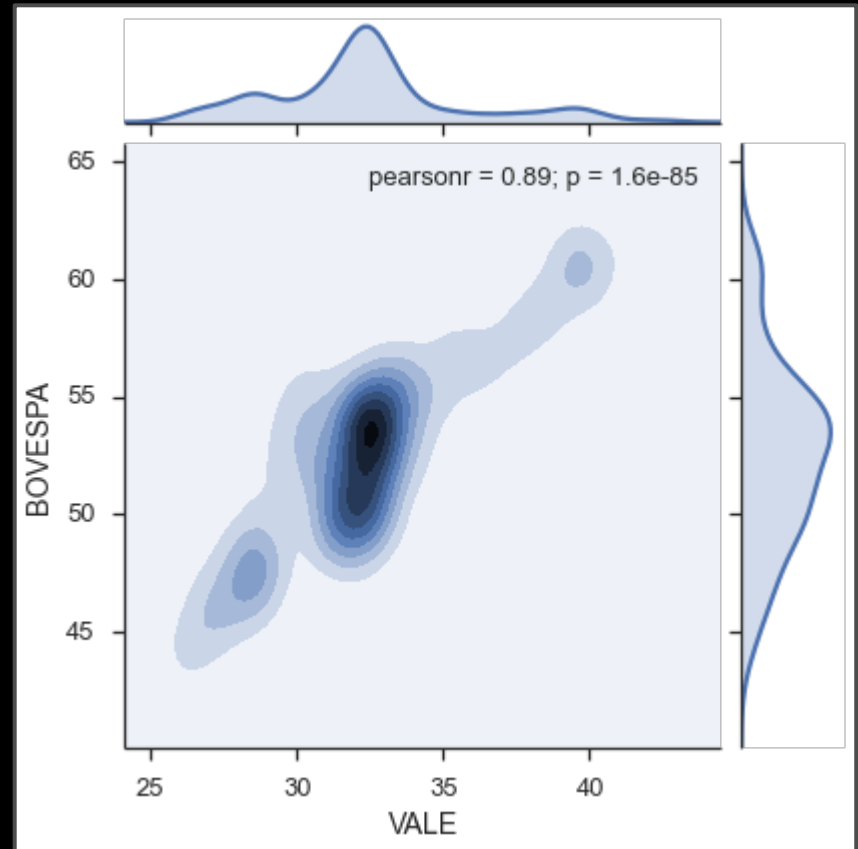


Mais no detalhe

```
sb.jointplot("VALE", "BOVESPA",  
            series, kind="reg");
```



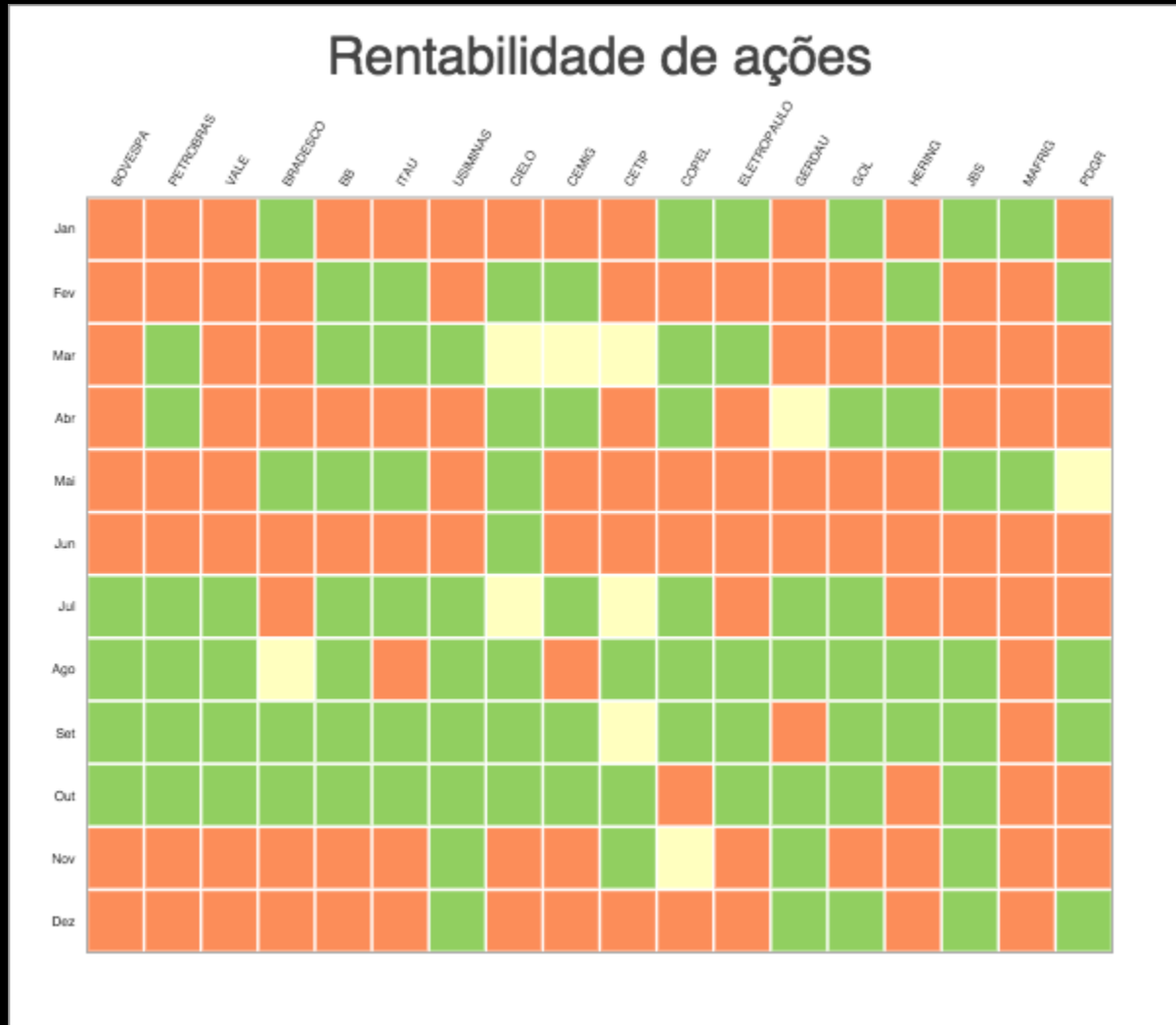
```
sb.jointplot("VALE", "BOVESPA",  
            series, kind="kde");
```



Rentabilidade de ações

- Qual a rentabilidade de cada ação por mês?
 - Como visualizar tudo no mesmo gráfico?
 - Como fazer isso higienicamente?
-
- Vamos considerar 18 ações no ano de 2013

Rentabilidade de ações



Rentabilidade de ações

```
from bokeh.plotting import *
output_notebook()
import numpy as np
import Quandl
import pandas as pd
from collections import OrderedDict
tickers = ["GOOG/BVMF_BOVA11.4", "GOOG/BVMF_PETR3.4", "GOOG/BVMF_VALE5.4", "GOOG/BVMF_BBDC4.4", "GOOG/BVMF_BBAS3.4", "GOOG/BVMF_ITUB4.4", "GOOG/BVMF_USIM5.4", "GOOG/BVMF_CIEL3.4", "GOOG/BVMF_CMIG4.4", "GOOG/BVMF_CTIP3.4", "GOOG/BVMF_CPLE6.4", "GOOG/BVMF_ELET6.4", "GOOG/BVMF_GGBR4.4", "GOOG/BVMF_GOLL4.4", "GOOG/BVMF_HGTX3.4", "GOOG/BVMF_JBSS3.4", "GOOG/BVMF_MRF63.4", "GOOG/BVMF_PDGR3.4"]
series = Quandl.get(tickers, trim_start="2013-01-01", trim_end="2013-12-31", authtoken="nJ1NhTYdEs2p3MsS4Cvd")
series.columns = ['BOVESPA', 'PETROBRAS', 'VALE', 'BRADESCO', 'BB', 'ITAU', 'USIMINAS', 'CIELO', 'CEMIG', 'CETIP', 'COPEL', 'ELETROPOLULO', 'GERDAU', 'GOL', 'HERING', 'JBS', 'MAFRIG', 'PDGR']
series.dropna(inplace=True)
logreturns = np.log(series).diff().dropna()
months = ["Jan", "Fev", "Mar", "Abr", "Mai", "Jun", "Jul", "Ago", "Set", "Out", "Nov", "Dez"]
series_mr = logreturns.resample('M', how=sum)*100
series_mr['Month'] = [months[d.month-1] for d in series_mr.index]
series_mr.set_index('Month', inplace=True)
months = list(series_mr.index)
stocks = list(series_mr.columns)
import brewer2mpl
bmap = brewer2mpl.get_map('RdYlGn', 'diverging', 3)
colors = bmap.hex_colors
levels = np.array([0, 1])
month = []
stock = []
color = []
rate = []
for y in stocks:
    for m in months:
        month.append(m)
        stock.append(y)
        monthly_rate = series_mr[y][m]
        rate.append(monthly_rate)
        color.append(colors[sum(levels < monthly_rate)])
source = ColumnDataSource(data=dict(month=month, year=stock, color=color, rate=rate, ))
figure()
rect('year', 'month', 0.95, 0.95, source=source, x_range=stocks, y_range=list(reversed(months)), color='color', line_color=None, tools="resize,hover,save", title="Rentabilidade de ações", plot_width=300, plot_height=400)
grid().grid_line_color = None
axis().axis_line_color = None
axis().major_tick_line_color = None
axis().major_label_text_font_size = "5pt"
axis().major_label_standoff = 0
xaxis().location = "top"
xaxis().major_label_orientation = np.pi/3
from bokeh.objects import HoverTool
hover = [t for t in curplot().tools if isinstance(t, HoverTool)][0]
hover.tooltips = OrderedDict([('date', '@month'), ('rate', '@rate'),])
show()
```

Esse gráfico já não é tão simples! ;-)

Referências

- [seaborn \(github\)](#)
 - [vincent \(github\)](#)
 - [Trifacta - vega](#)
 - [ggplot \(github\)](#)
 - [ggplot tutorial](#)
 - [ggplot2 no R](#)
 - [bokeh \(github\)](#)
 - [mpld3 \(github\)](#)
 - [prettyplotlib \(github\)](#)
 - [plot.ly](#)
 - [Quandl](#)
 - [brewer2mpl \(github\)](#)
-
- [AboutWilson.net: Refinando o estilo do matplotlib](#)
 - [AboutWilson.net: Refinando o estilo do matplotlib com seaborn](#)
 - [EuroPython 2014: Combining the powerful worlds of Python and R](#)
 - [EuroPython 2014: Scientific Visualization with GR](#)

<https://github.com/wilsonfreitas/tdc-2014>

Wilson Freitas

<http://aboutwilson.net>

@aboutwilson