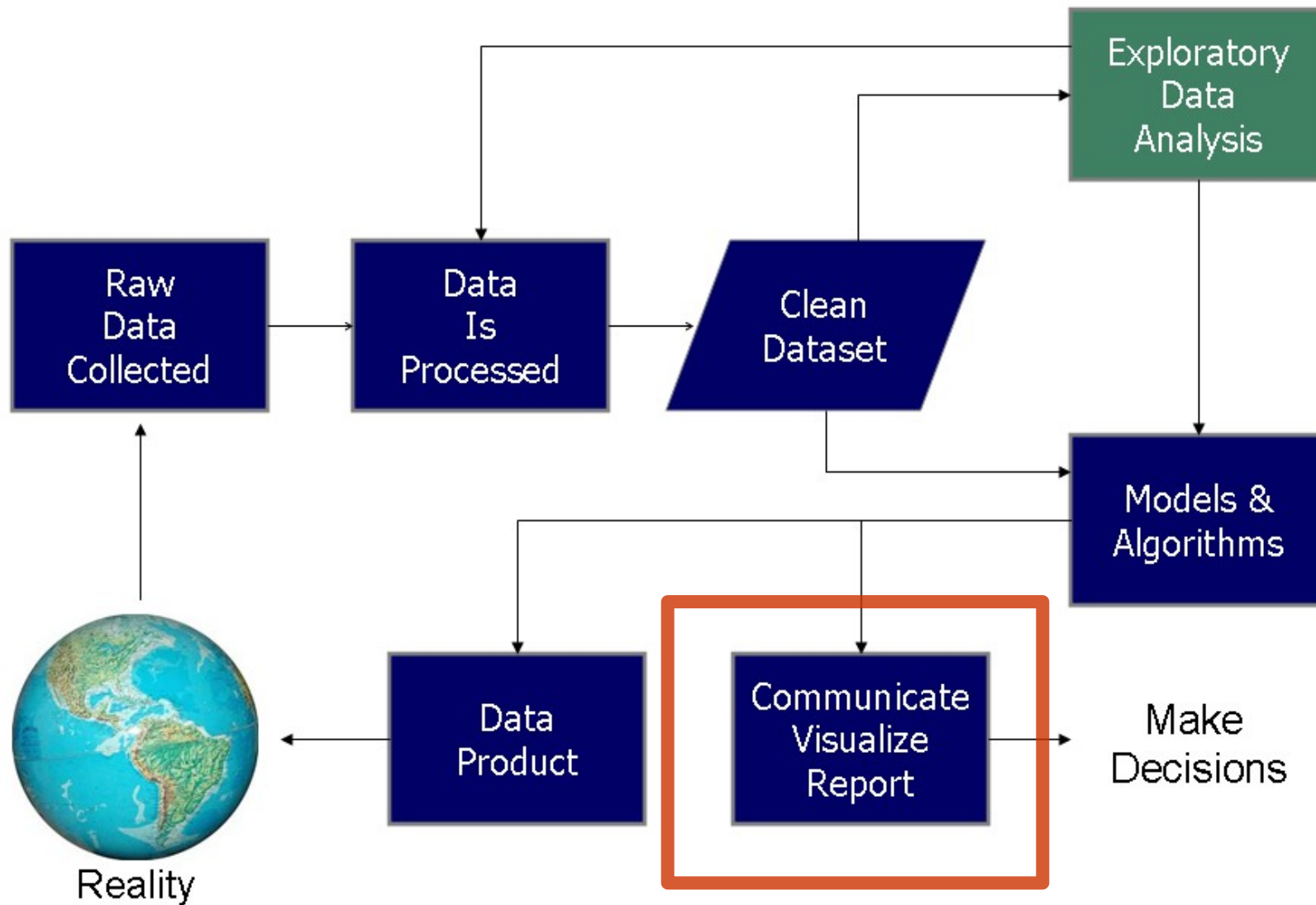




Visualização – Ferramentas

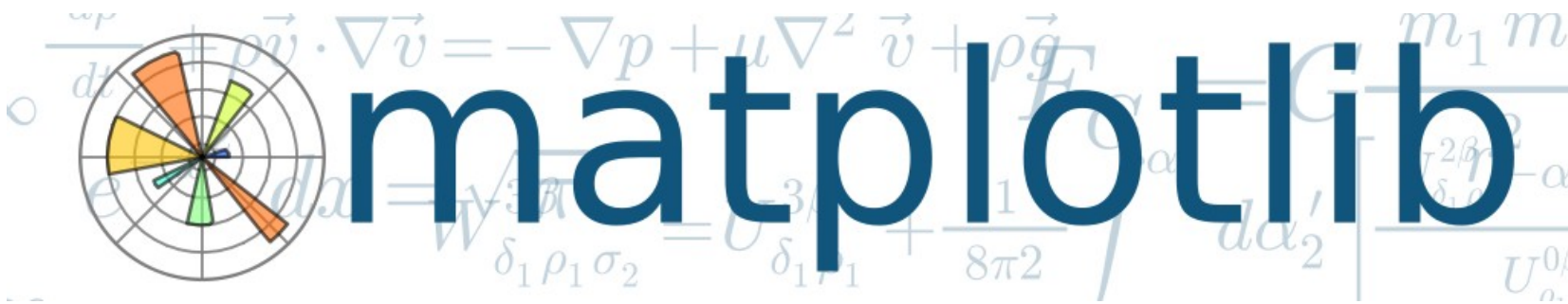
Luiz Celso Gomes-Jr

Data Science Process



Ferramentas

- Matplotlib – Visualizações básicas
- Seaborn – Visualizações avançadas
- Bokeh - Interatividade/Dashboarding



- Usado para plotar vários tipos de gráficos, de histogramas a grafos
- API de mais baixo nível, que provê flexibilidade porém apresenta maior complexidade
- Integrado ao Jupyter para simplificar plotagens

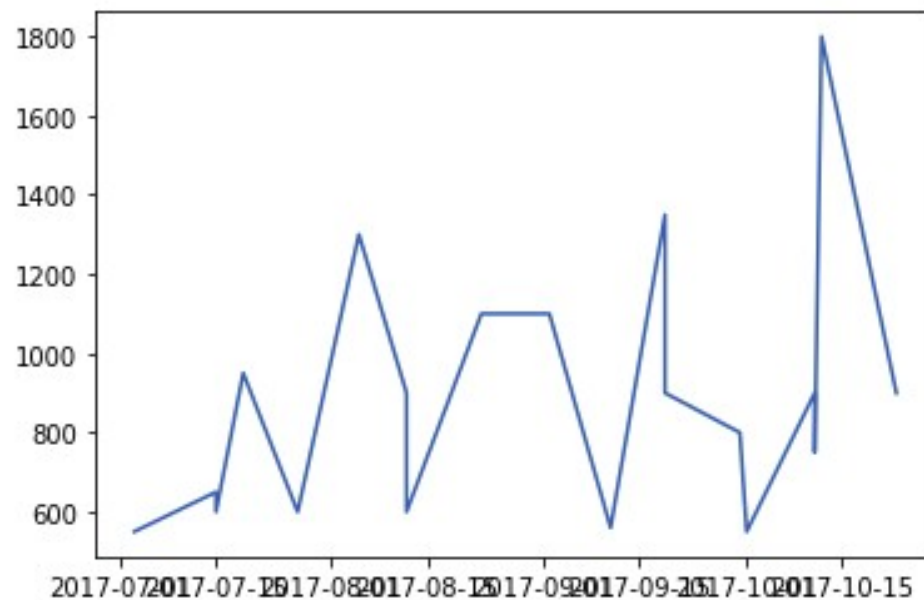
Matplotlib - Conceitos

- Figure: A figura completa que em geral contém um ou mais eixos.
- Axes: A parte da plotagem na figura. Nas plotagens 2D, uma figura contém um ou dois eixos.
- Axis: Objetos que representam as escalas de cada eixo.
- Artist: Qualquer desenho visto numa figura, incluindo textos, linhas, formas. Em geral associados a Axes.

Matplotlib

Para plotagem usando o matplotlib, em geral usa-se o pyplot, que provê uma interação similar ao MATLAB. Para plotar valores em duas coordenadas, basta especificar os valores dos eixos x e y como demonstrado abaixo:

```
# Importando o pyplot  
import matplotlib.pyplot as plt  
  
# Constrói a plotagem  
plt.plot(df['data'], df['aluguel'])  
  
# Exibe a figura  
plt.show()
```

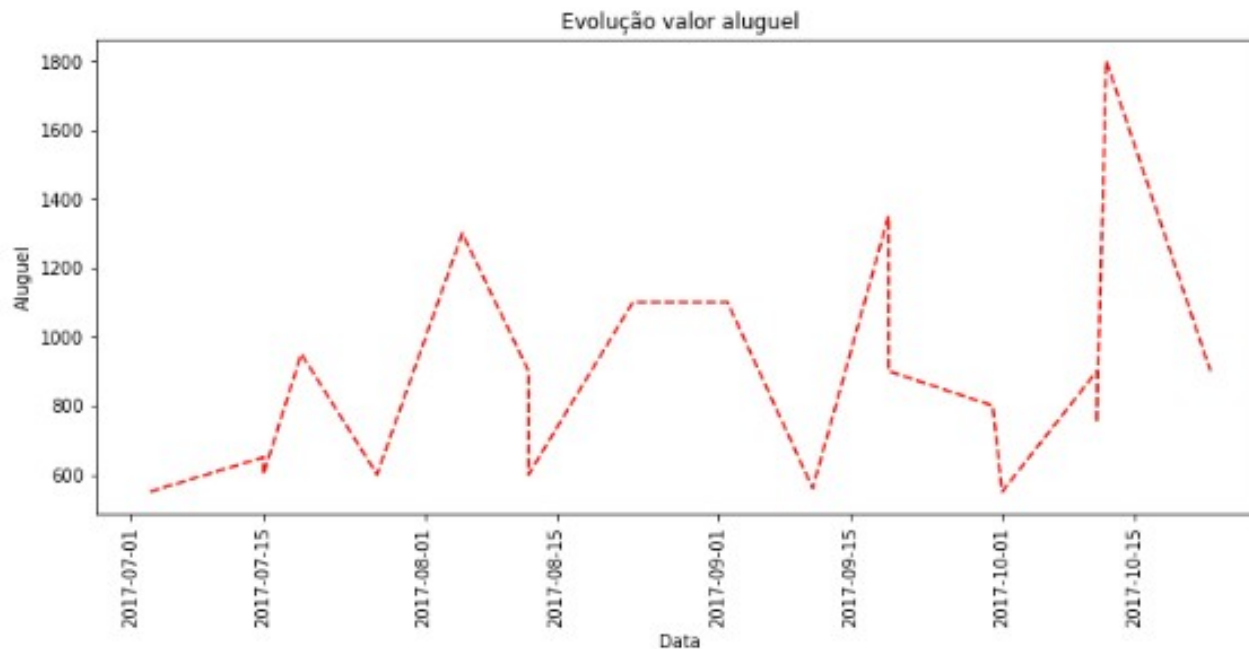


Matplotlib - Formatação

No exemplo abaixo ajustamos tamanho, estilo e rótulos em uma plotagem. O parâmetro `r--` especifica uma linha tracejada (`--`) da cor vermelha (`r`).

```
plt.figure(figsize=(12,5))
plt.plot(df['data'], df['aluguel'], "r--")
plt.title('Evolução valor aluguel')
plt.xlabel('Data')
plt.xticks(rotation='vertical')
plt.ylabel('Aluguel')
```

```
Text(0, 0.5, 'Aluguel')
```

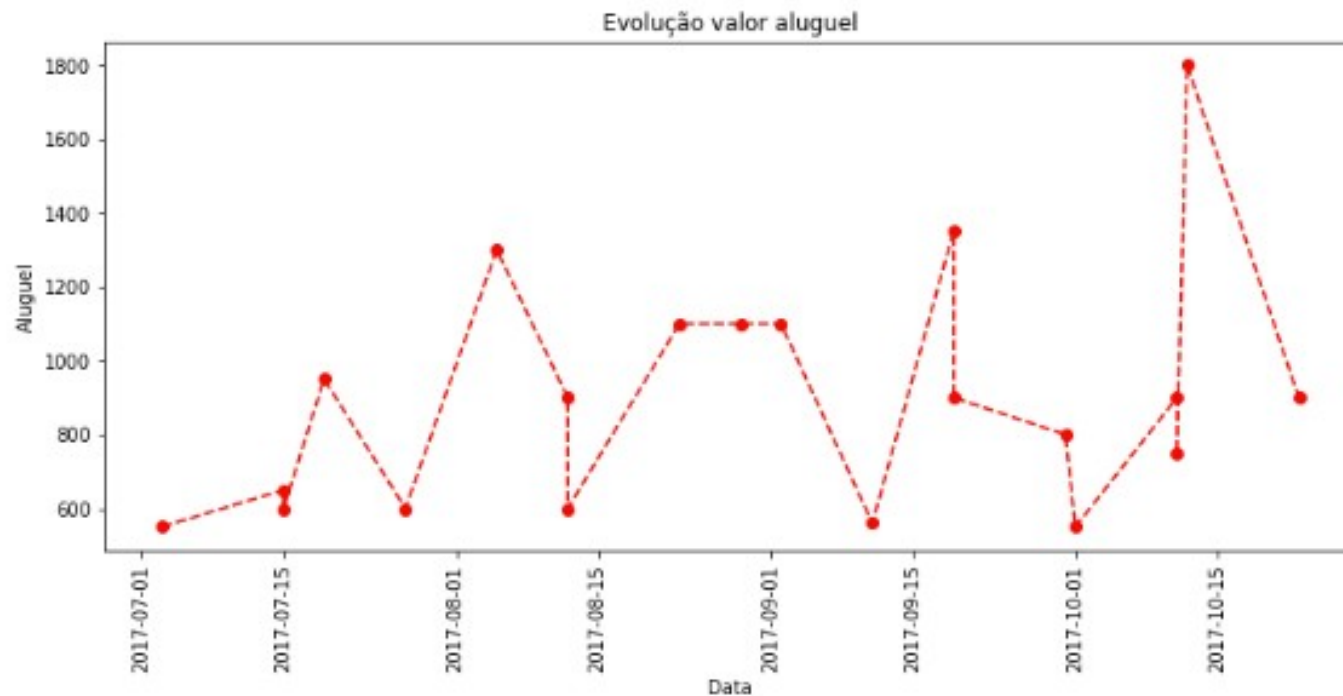


Matplotlib - Formatação

No exemplo abaixo ajustamos tamanho, estilo e rótulos em uma plotagem. O parâmetro `or--` especifica marcadores circulares (o), com linha tracejada (--) da cor vermelha (r).

















```
plt.figure(figsize=(12,5))  
plt.plot(df['data'], df['aluguel'], "or--")  
plt.title('Evolução valor aluguel')  
plt.xlabel('Data')  
plt.xticks(rotation='vertical')  
plt.ylabel('Aluguel')
```

```
Text(0, 0.5, 'Aluguel')
```



Matplotlib - Formatação

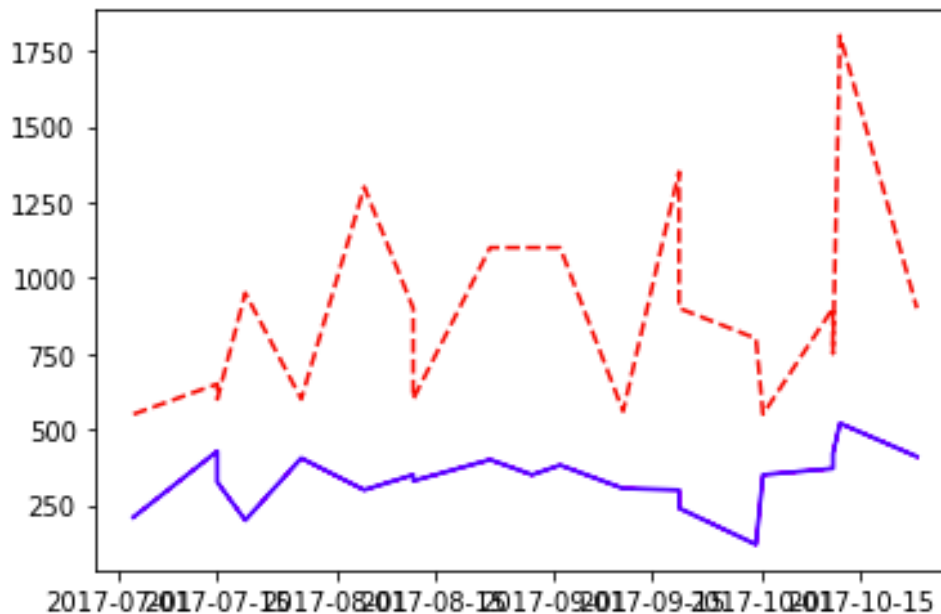
linestyle	description
'-' or 'solid'	solid line
'--' or 'dashed'	dashed line
'-.' or 'dashdot'	dash-dotted line
':' or 'dotted'	dotted line
'None'	draw nothing
' '	draw nothing
''	draw nothing

marker	symbol	description
"."		point
","		pixel
"o"		circle
"v"		triangle_down
"^"		triangle_up
"<"		triangle_left
">"		triangle_right
"1"		tri_down
"2"		tri_up
"3"		tri_left
"4"		tri_right
"8"		octagon
"s"		square
"p"		pentagon
"P"		plus (filled)
"*"		star
"h"		hexagon1
"H"		hexagon2
"+"		plus
"x"		x

Matplotlib - Linhas

Para incluir outras linhas na plotagem, basta adicionar mais parâmetros na chamada do método:

```
ax = plt.plot(df['data'], df['aluguel'], "r--",  
              df['data'], df['condominio'], "b-")
```



Matplotlib – Múltiplas plotagens

Um recurso útil é a criação de figuras com mais de uma plotagem (subplot). Abaixo definimos uma figura que tem duas linhas e uma coluna de plotagens.

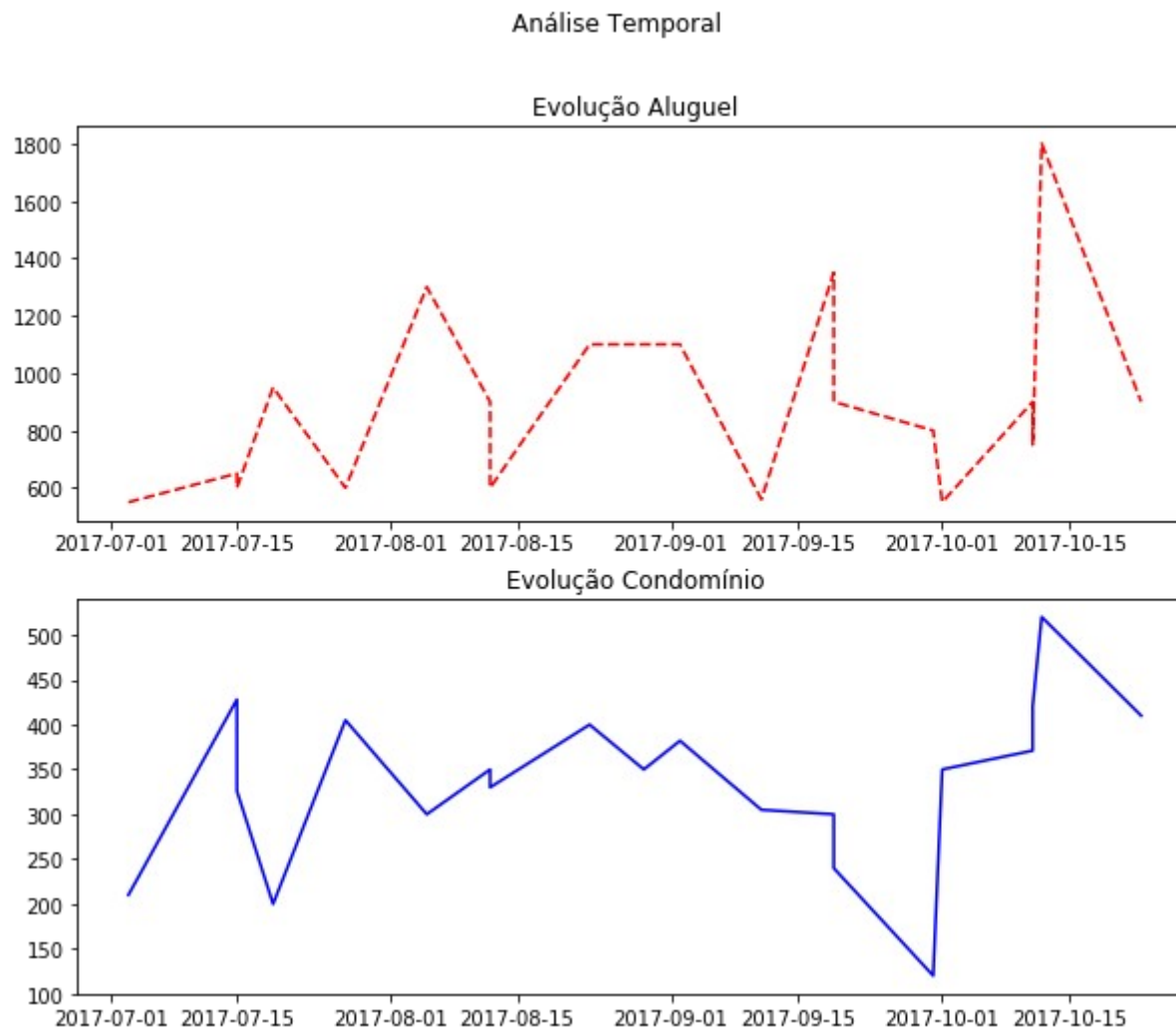
```
plt.figure(figsize=(10,8))
plt.subplot(2,1,1)
plt.plot(df['data'], df['aluguel'], "r--")
plt.title("Evolução Aluguel")

plt.subplot(2,1,2)
plt.plot(df['data'], df['condominio'], "b-")
plt.title("Evolução Condomínio")

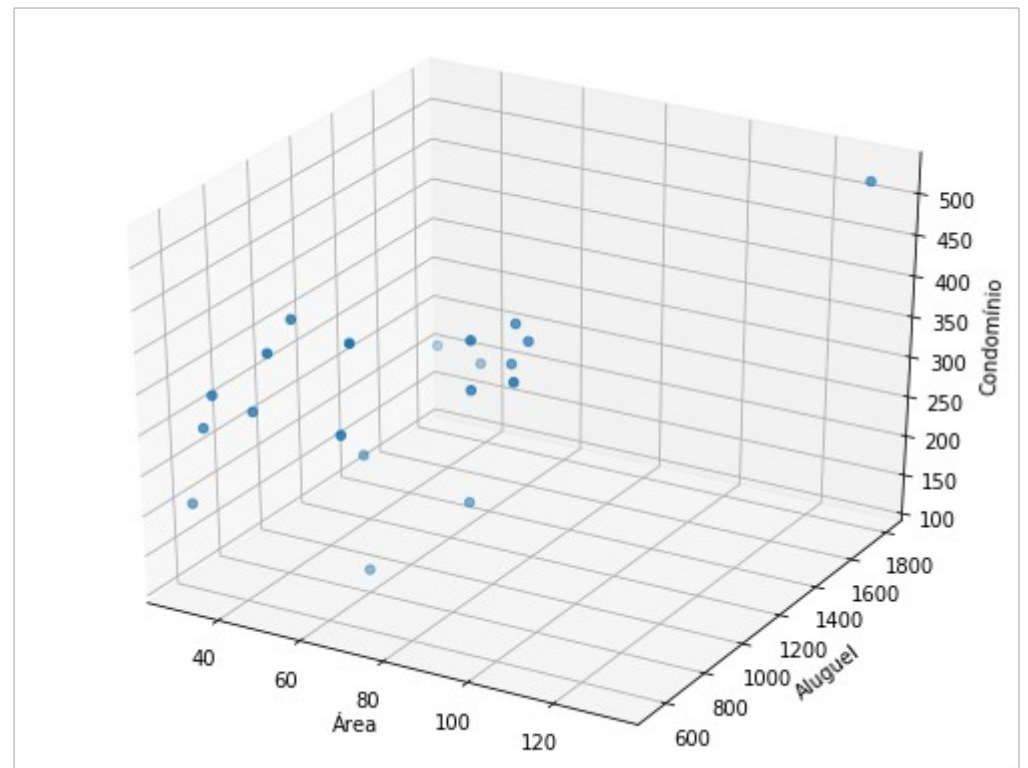
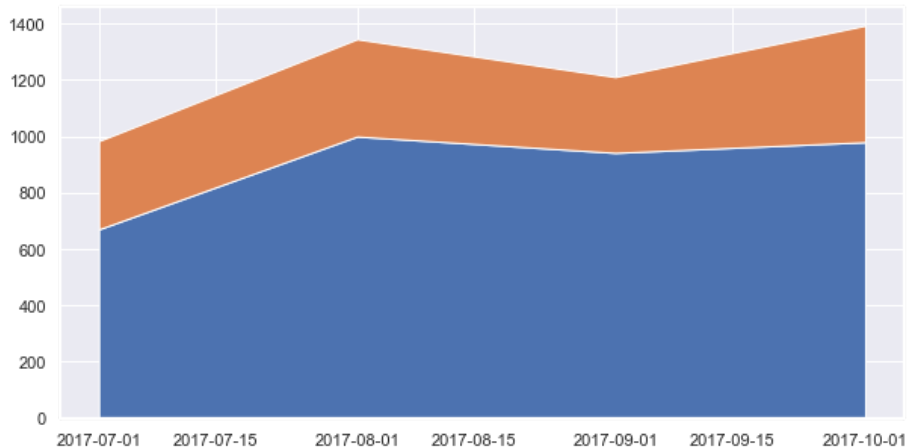
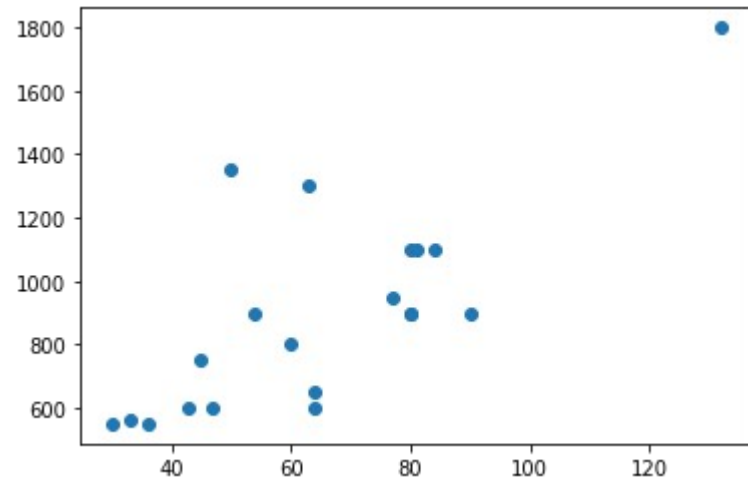
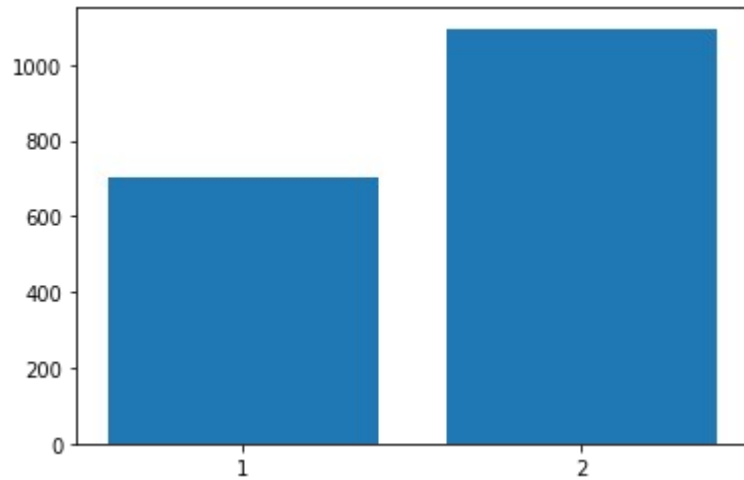
plt.suptitle("Análise Temporal")
```

Matplotlib – Múltiplas plotagens

Um recurso útil é a criação de figuras com mais de uma plotagem (subplot). Abaixo definimos uma figura que tem duas linhas e uma coluna de plotagens.



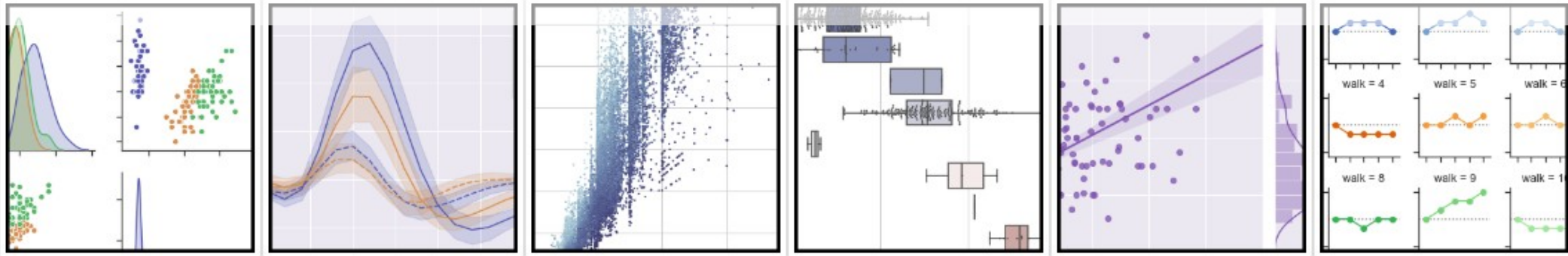
Matplotlib – Outras plotagens



Exercícios!

- Revise o conteúdo do notebook:
 - 06a-Visualização – Matplotlib.ipynb

Seaborn

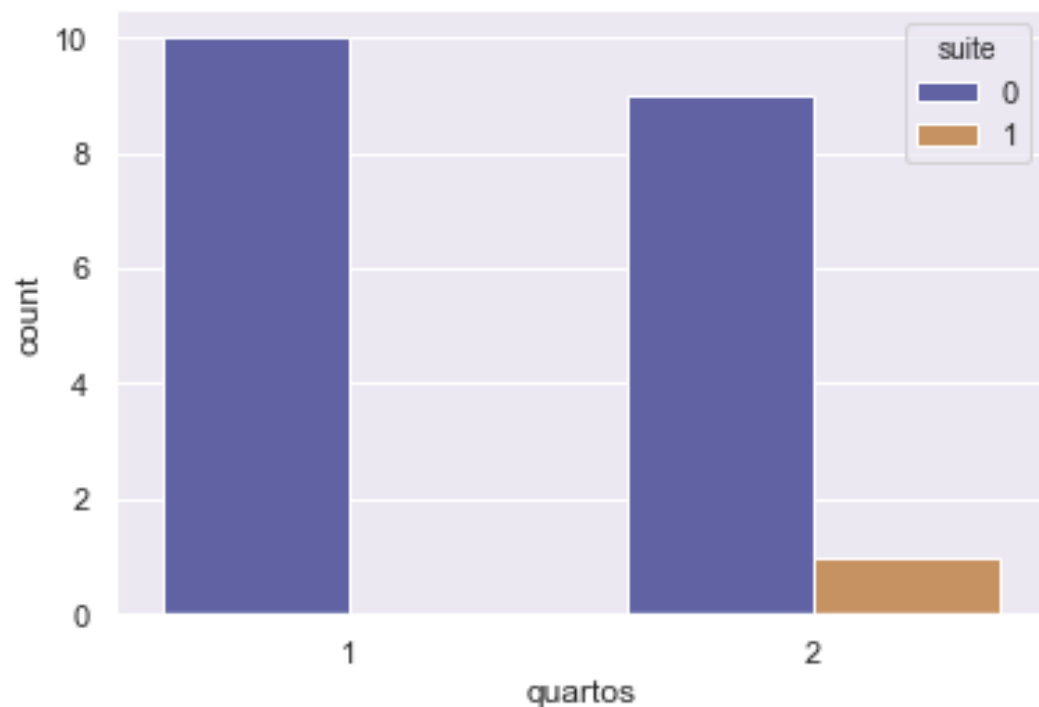


- Baseada no matplotlib
- Recursos para a criação de múltiplas plotagens baseadas em variáveis categóricas
- Recursos para avaliação de relacionamentos entre variáveis
- Controle de estilo simplificado
- Se integra à estrutura de DataFrames do Pandas

Plotagens por categoria

O Seaborn simplifica a criação de plotagens mostrando dados separados por categoria. No exemplo abaixo exibimos as contagens de apartamentos por quarto e usamos as cores (hue) para indicar os que têm e não têm suítes.

```
ax = sns.countplot(x="quartos", hue="suite", data=df)
```



Plotagens por categoria

Podemos incluir ainda mais categorias na visualização. Por exemplo, podemos incluir a informação de número de vagas e exibir a separação em colunas (col) diferentes da figura:

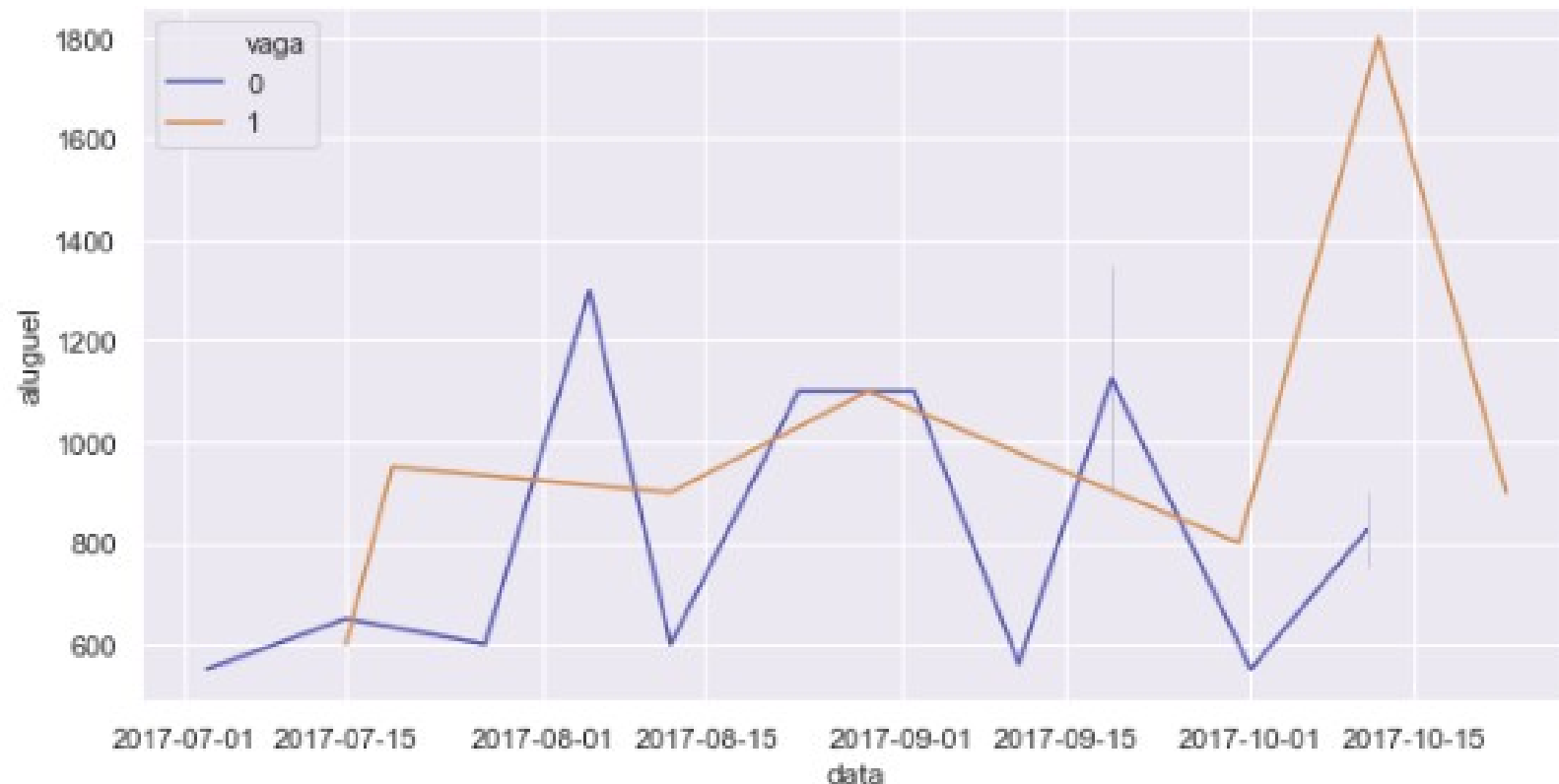
```
ax = sns.catplot(x="quartos", hue="suite",  
                 col = "vaga", kind="count", data=df)
```



Séries

Séries temporais são tipicamente exibidas como gráficos de linhas com as datas no eixo x.

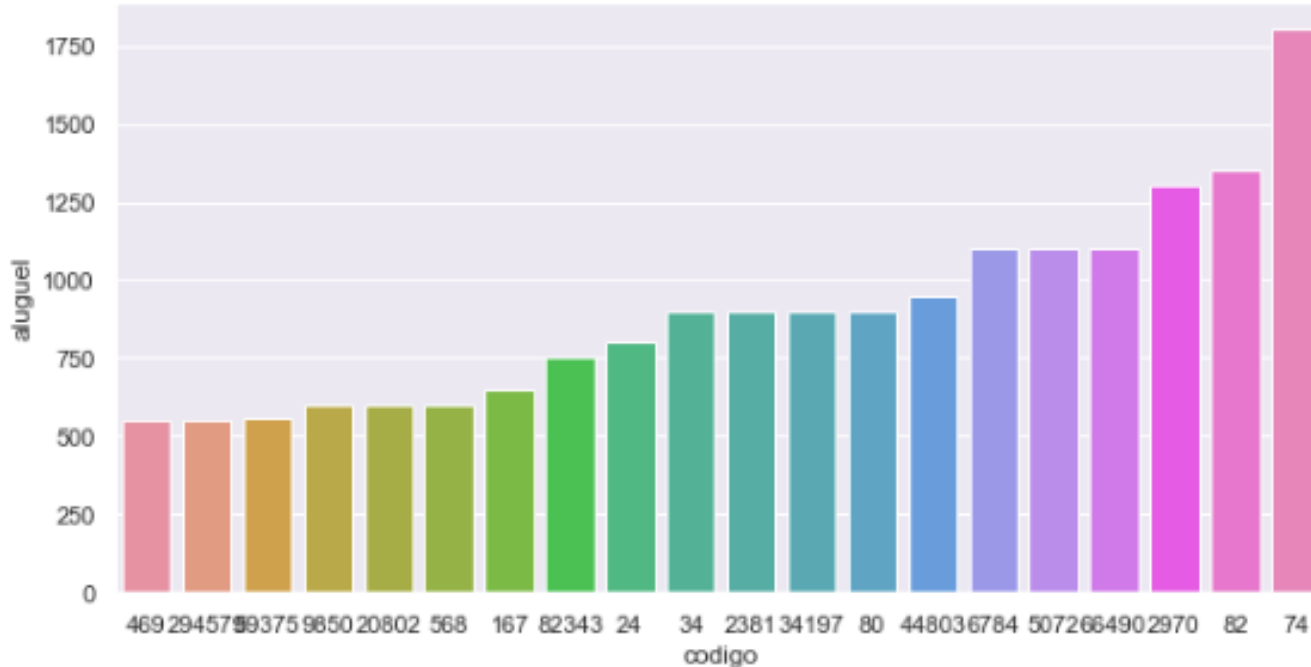
```
fig, ax = plt.subplots(figsize=(10, 5))  
ax = sns.lineplot(x="data", y='aluguel', hue="vaga", data=df)
```



Ranqueamento

Para mostrar o ranqueamento de valores, podemos usar um gráfico de barras. No Seaborn é preciso passar um parâmetro indicando a ordem dos valores exibidos:

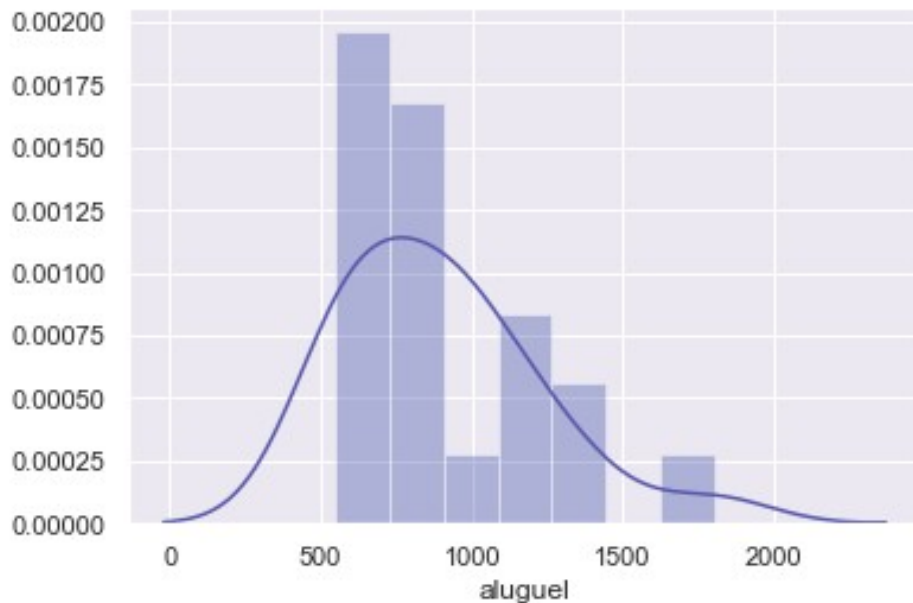
```
fig, ax = plt.subplots(figsize=(10, 5))  
  
ordem = df.sort_values('aluguel')['codigo']  
ax = sns.barplot(x="codigo", y="aluguel", order=ordem, data=df)
```



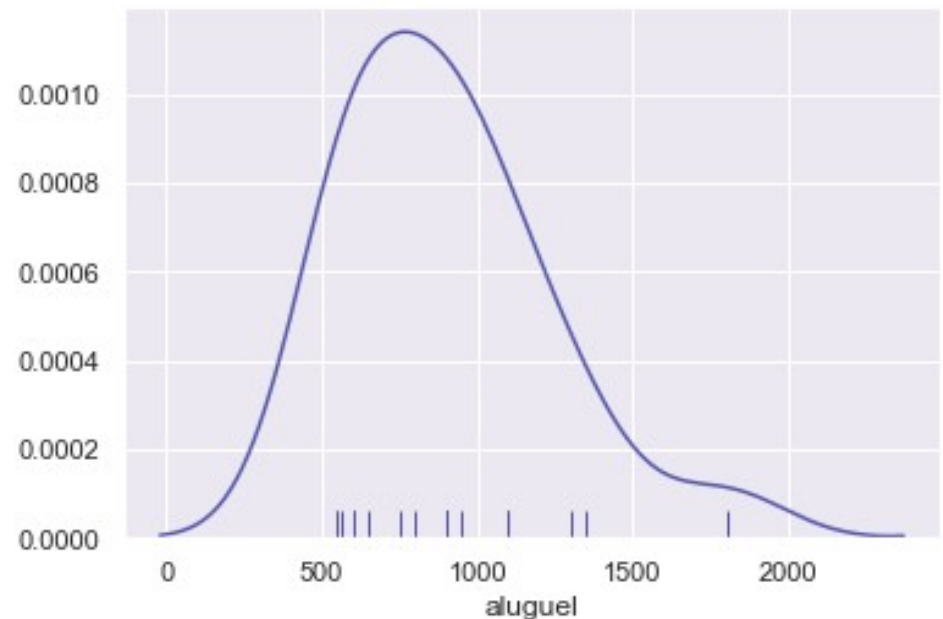
Distribuição

O Seaborn possui diversos recursos úteis para visualizar distribuição de variáveis. Por exemplo, plotagem de histograma com curva KDE:

```
sns.distplot(df['aluguel'], bins=7);
```



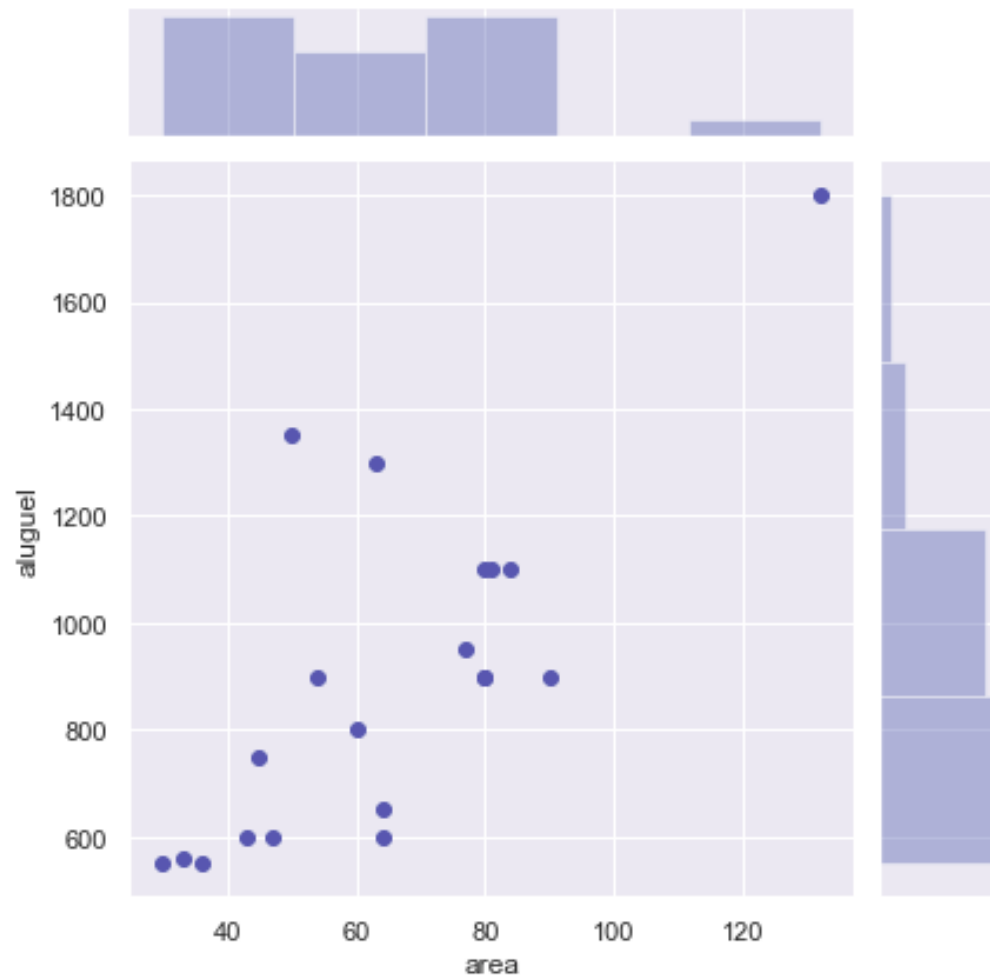
```
sns.distplot(df['aluguel'], hist=False, rug=True);
```



Distribuição

Scatterplot com histogramas das variáveis integrados:

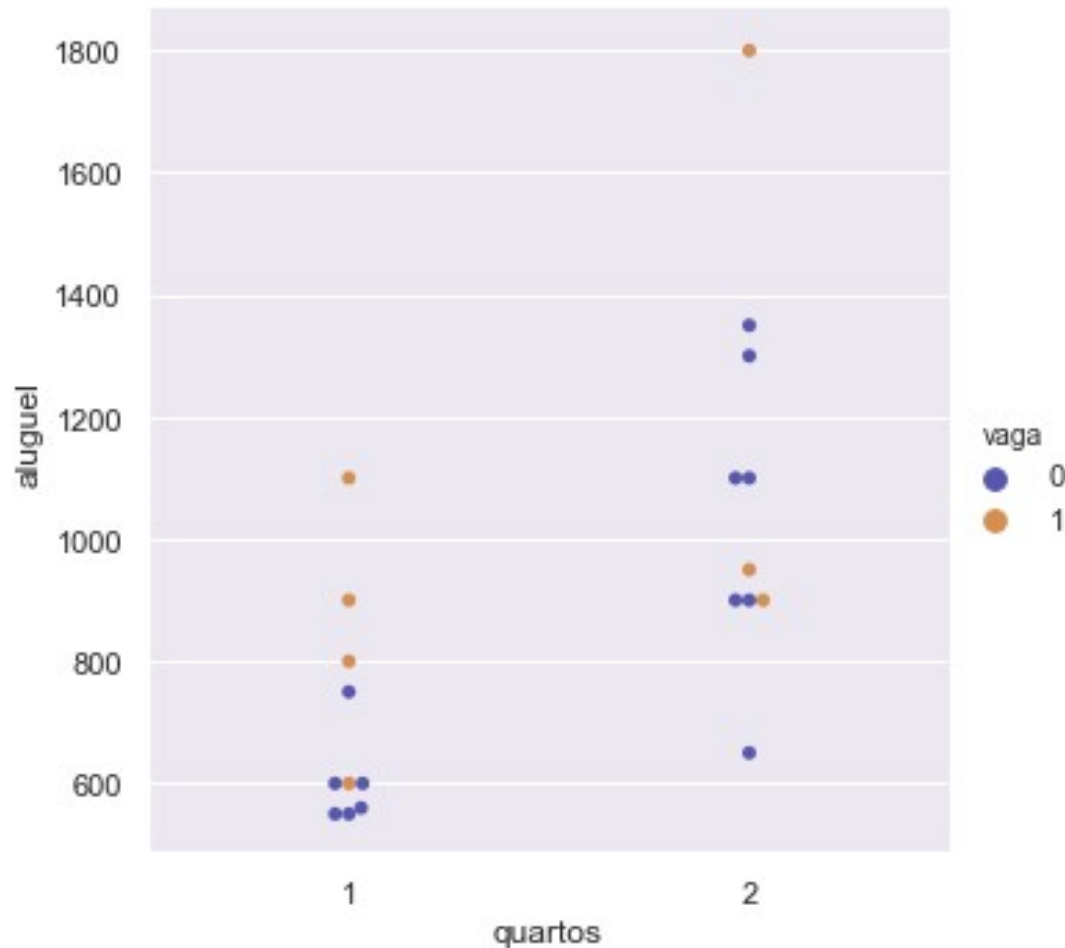
```
sns.jointplot(x="area", y="aluguel", data=df);
```



Distribuição

Plotagem de instâncias por categoria:

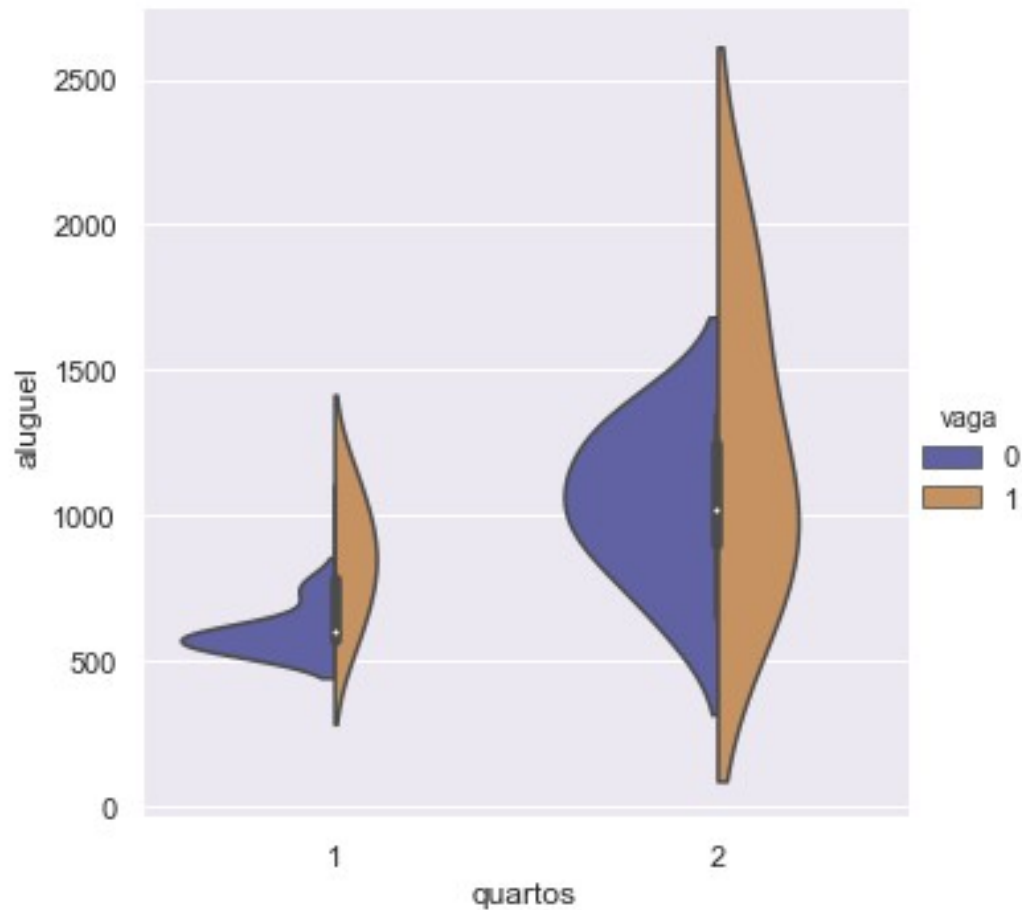
```
sns.catplot(x="quartos", y="aluguel", hue="vaga",  
            kind="swarm", data=df);
```



Distribuição

Violin plot

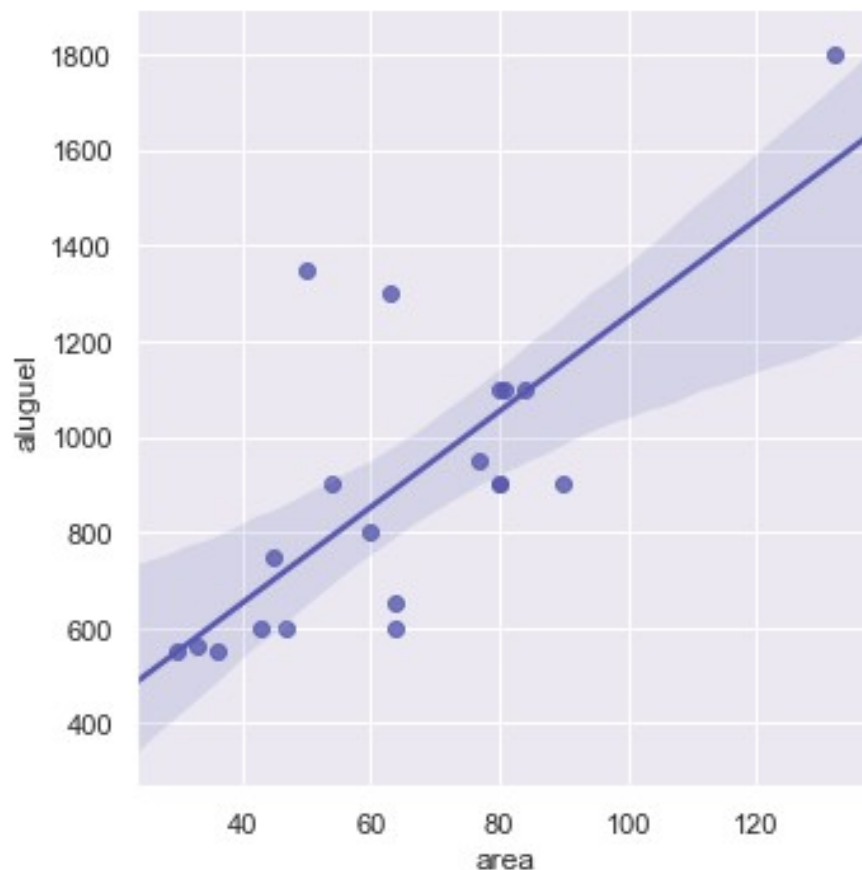
```
sns.catplot(x="quartos", y="aluguel", hue="vaga",  
            kind="violin", split = True, data=df);
```



Correlação

Os recursos do Seaborn para análise de correlação também são muito poderosos. Por exemplo, scatterplot com reta de regressão para estimar o relacionamento linear entre as variáveis:

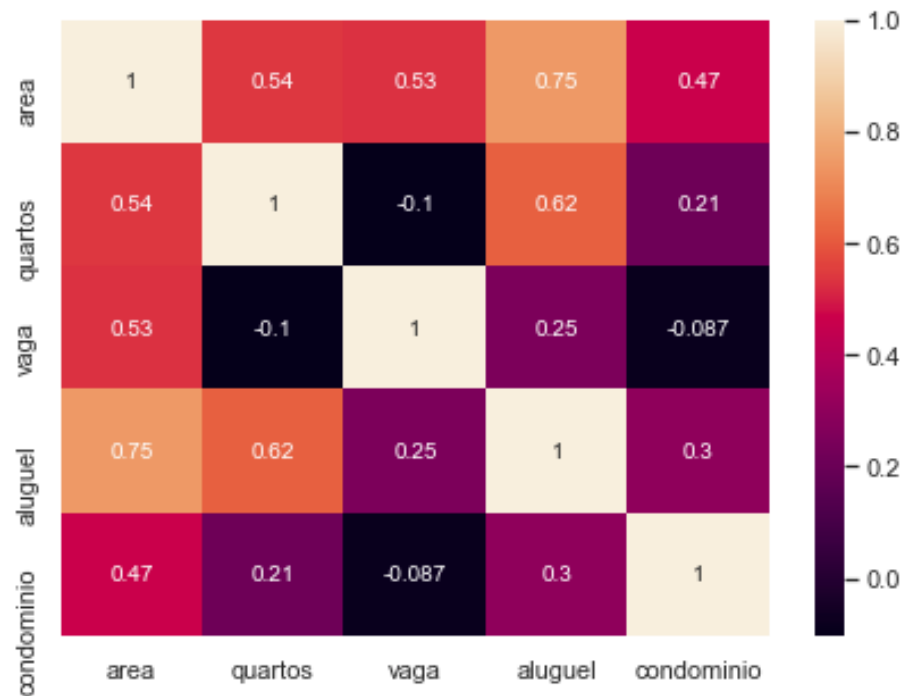
```
sns.lmplot(x="area", y="aluguel", data=df);
```



Correlação - Heatmap

	area	quartos	vaga	aluguel	condominio
area	1.000000	0.542466	0.533035	0.748196	0.466627
quartos	0.542466	1.000000	-0.104828	0.619797	0.214173
vaga	0.533035	-0.104828	1.000000	0.251974	-0.087415
aluguel	0.748196	0.619797	0.251974	1.000000	0.302494
condominio	0.466627	0.214173	-0.087415	0.302494	1.000000

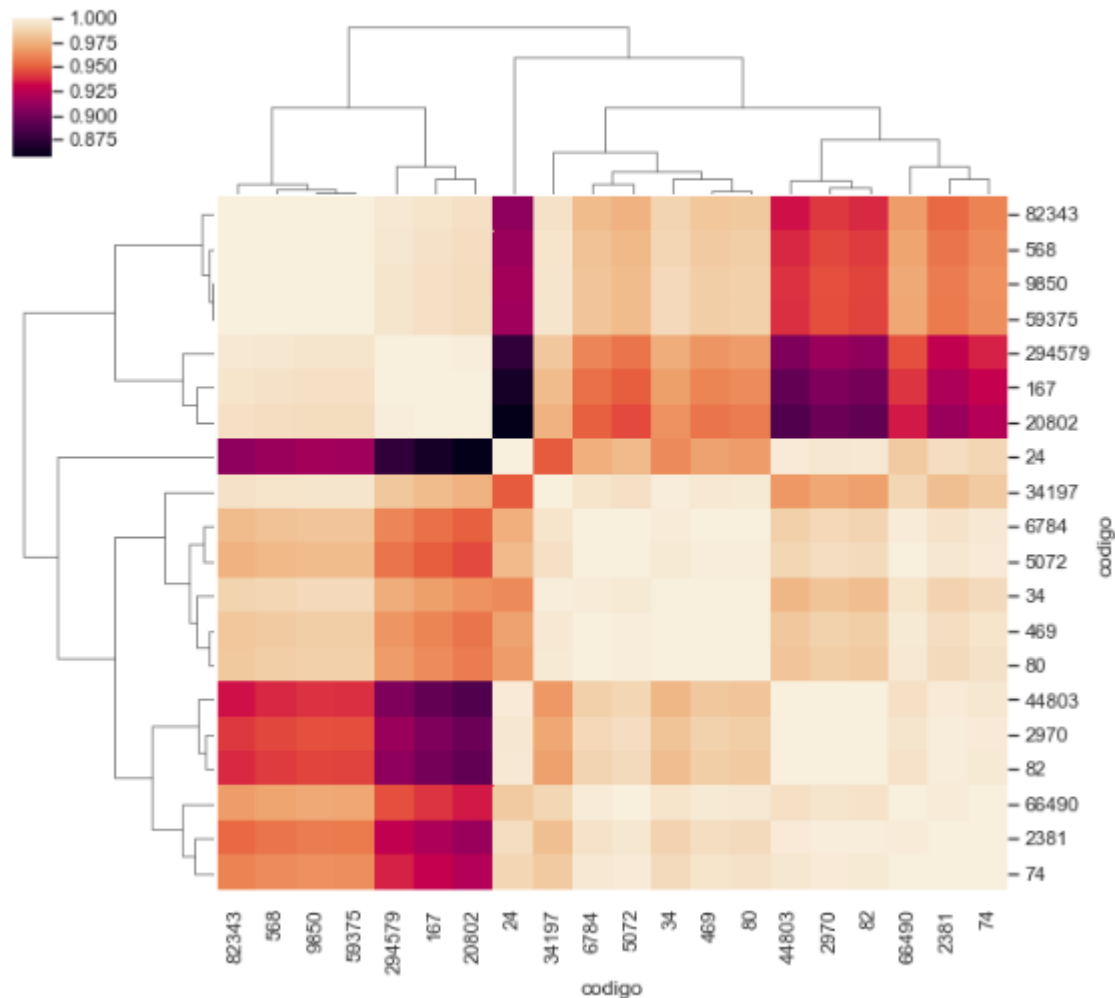
```
fig, ax = plt.subplots(figsize=(7, 5))  
hm = sns.heatmap(df_corr, ax = ax, annot=True)
```



Correlação - Heatmap/dendograma

O Seaborn também permite a construção de dendogramas para identificação de clusters entre as observações. No exemplo abaixo buscamos identificar agrupamentos entre as ofertas de apartamentos:

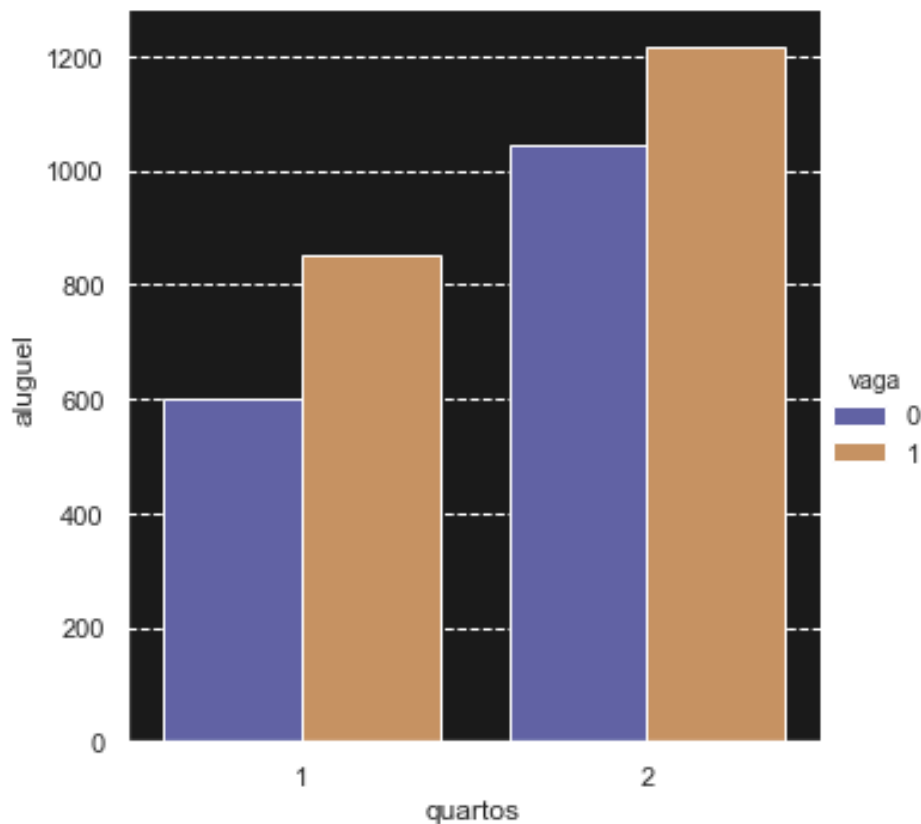
```
cm = sns.clustermap(df_corr_aps, figsize=(10,9),  
                    col_cluster=True, annot=False)
```



Ajustando estilos

O Seaborn simplifica o ajuste de estilos aplicados aos gráficos gerados. Estilos de cores dos elementos, fontes e tipos de linhas podem ser alterados com o método `set_style`:

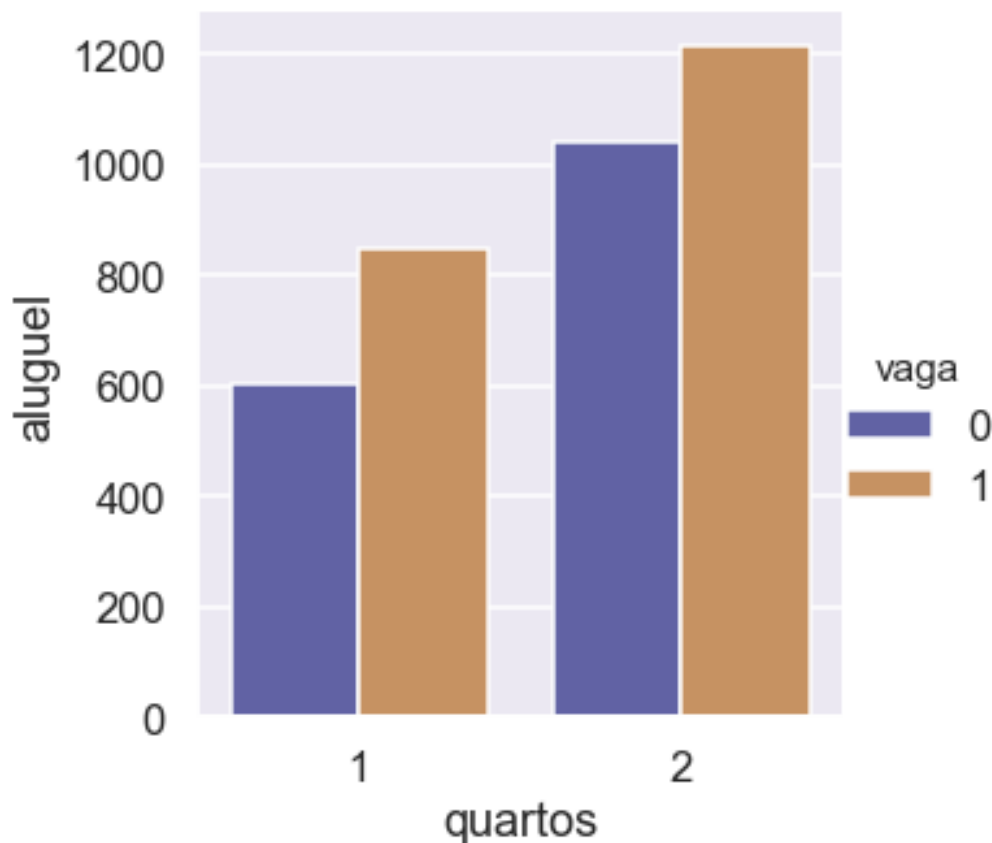
```
sns.set_style("dark",  
              {"axes.facecolor": ".1",  
               'axes.grid': True, 'grid.linestyle': '--'})  
cp = sns.catplot(x="quartos", y="aluguel", hue="vaga",  
                 kind="bar", ci=None, data=df)
```



Ajustando estilos

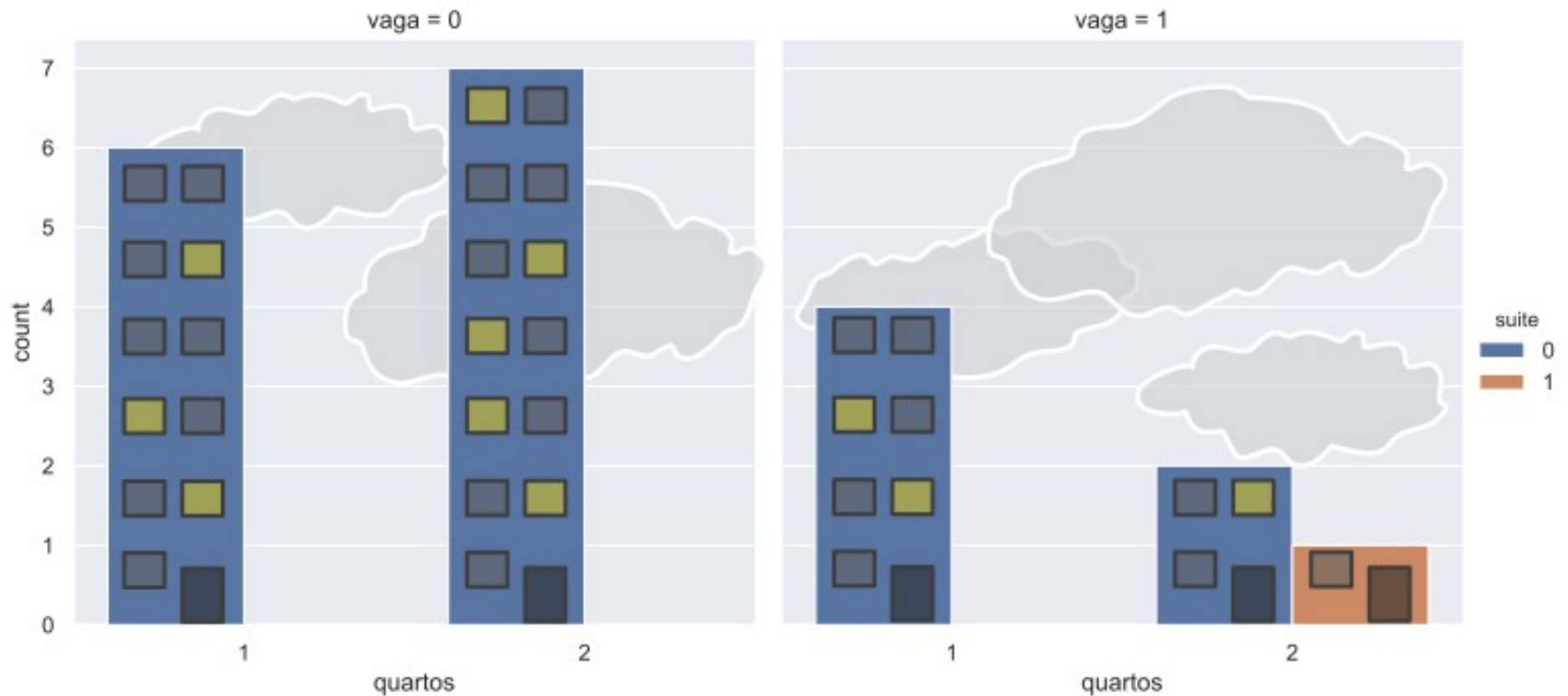
Para alterar fontes de forma a produzir gráficos adequados para cada tipo de contexto, é possível usar o método `set_context`. As opções possíveis são “notebook”, “paper”, “talk”, “poster”

```
sns.set()  
sns.set_context("talk")  
cp = sns.catplot(x="quartos", y="aluguel",  
                 hue="vaga", kind="bar", ci=None, data=df)
```



Infográficos

Para criar infográficos, você pode editar um gráfico salvo como SVG usando seu editor vetorial favorito.



Exercícios!

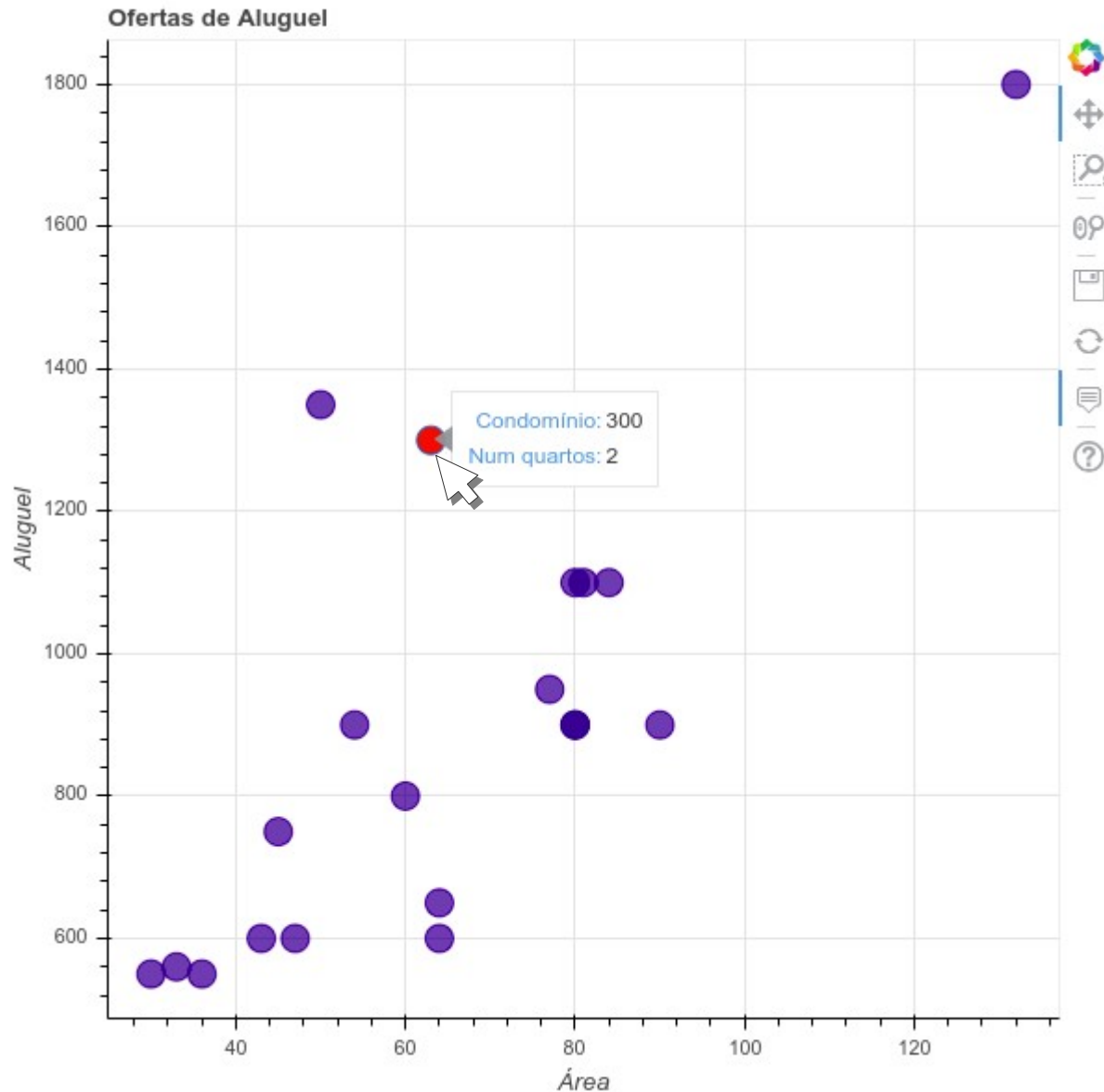
- Revise o conteúdo do notebook:
 - 06b-Visualização – Seaborn.ipynb

Bokeh - Interatividade

- Visualizações interativas
- Dashboards



Bokeh - Interatividade



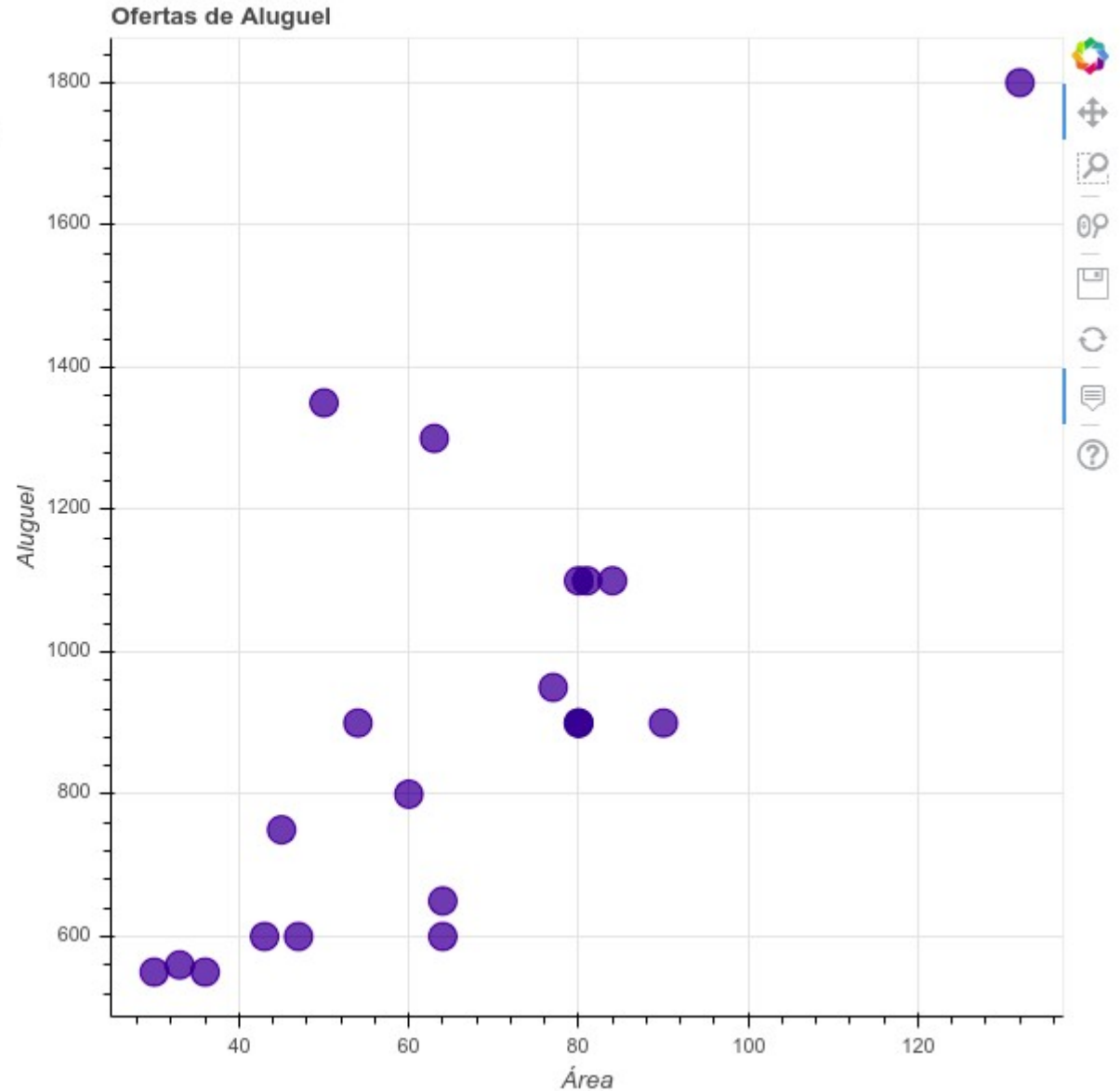
Bokkeh - Dashboard

show(app)

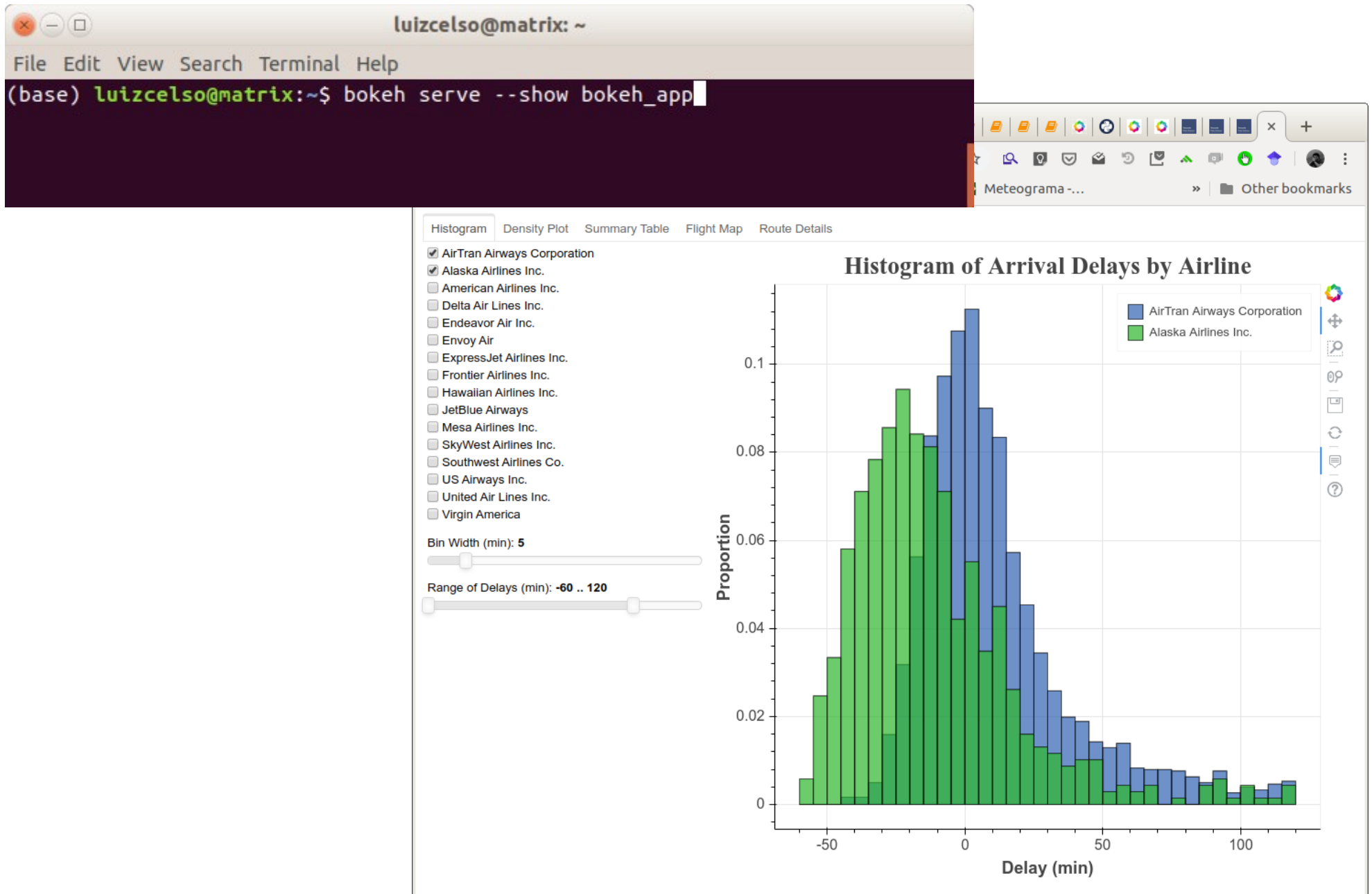
☒ 1

☒ 2

Faixa de aluguel: 100 .. 2000



Bokeh - Dashboard



Exercícios!

- Revise o conteúdo dos notebooks:
 - 06a-Visualização – Matplotlib.ipynb
 - 06b-Visualização – Seaborn.ipynb
 - 06c-Visualização – Interatividade.ipynb
 - 06d-Visualização – Interatividade-Controles.ipynb