

Análise e Manipulação de Dados com Python

com Pandas e Matplotlib

Humberto da Silva Neto

Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo
Campus Vitória

23 de Novembro de 2018



Agenda

1 Introdução

- Introdução ao Pandas

2 Correção dos dados

3 Padronização e Normalização de dados

4 Binning

5 Correlação

6 Visualização dos dados

7 Extras

- Waffle Charts
- Word Clouds



Github Repository

Repositório

Py-Pandas-Minicourse

IPython Notebooks

- Pandas Basics
- Pandas Plot
- Waffle Charts
- Word Clouds



Mas antes de tudo ...

1 Quem aqui sabe programar?



Mas antes de tudo ...

- 1 Quem aqui sabe programar?**
- 2 Quem aqui sabe programar **em Python**?**



Mas antes de tudo ...

- 1 Quem aqui sabe programar?
- 2 Quem aqui sabe programar **em Python?**
- 3 Já usou alguma vez a biblioteca Matplotlib?



Mas antes de tudo ...

- 1 Quem aqui sabe programar?**
- 2 Quem aqui sabe programar **em Python**?**
- 3 Já usou alguma vez a biblioteca Matplotlib?**
- 4 E Pandas?**



Mas antes de tudo ...

- 1** Quem aqui sabe programar?
- 2** Quem aqui sabe programar **em Python?**
- 3** Já usou alguma vez a biblioteca Matplotlib?
- 4** E Pandas?

Não se preocupe!

A medida que fizermos os exercícios eu tentarei ajudar o máximo que puder.

Sinta-se livre para pedir ajuda aos colegas também. **Conversem e discutam soluções!**



Introdução



Análise de dados



Dennis Junk diz que análise de dados é :

"toda maneira de decompor os dados, avaliar as tendências ao longo do tempo e comparar um setor ou medida com outro. Também pode incluir as várias maneiras pelas quais os dados são visualizados para tornar as tendências e os relacionamentos intuitivos em um piscar de olhos."



Análise de dados



Dennis Junk diz que análise de dados é :

"toda maneira de decompor os dados, avaliar as tendências ao longo do tempo e comparar um setor ou medida com outro. Também pode incluir as várias maneiras pelas quais os dados são visualizados para tornar as tendências e os relacionamentos intuitivos em um piscar de olhos."



Em outras palavras, análise de dados é

Um processo de inspeção, limpeza, transformação e modelagem de dados com o objetivo de **descobrir informações úteis**.



Dados estruturados x não-estruturados

Structured Data

Size	#bedrooms	...	Price (1000\$)
2104	3		400
1600	3		330
2400	3		369
:	:		:
3000	4		540

User Age	Ad Id	...	Click
41	93242		1
80	93287		0
18	87312		1
:	:		:
27	71244		1

Unstructured Data



Audio



Image

Four scores and seven years ago...

Text

Fonte: Andrew Ng

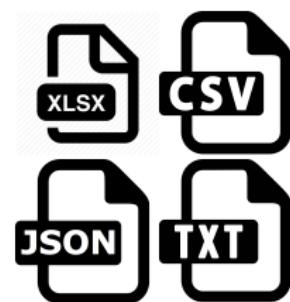
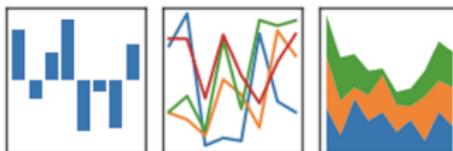


O que é Pandas?

Pandas é uma biblioteca de código aberto escrita para a linguagem de programação Python para manipulação e análise de dados. Em particular, oferece estruturas de dados e operações para manipular tabelas numéricas e séries temporais.

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Estrutura de dados

As estruturas de Dados fornecidas pelos Pandas são de dois tipos distintos:

- 1 **Pandas Series:** é um array de uma dimensão. Você pode considerar um Series também como uma coluna de uma tabela. Exemplo:

INDEX

A	3
B	-5
C	7
D	4

- 2 **Pandas DataFrame:** é uma estrutura de dados rotulados em 2-D com colunas de tipo potencialmente diferente. Exemplo:

INDEX

	País	Capital	População
1	Bélgica	Bruxelas	123465
2	Índia	Nova <u>Delhi</u>	456789
3	Brasil	Brasília	987654



Fontes: Pandas, Paulo Vasconcellos e Daksh

Comandos básicos

- 1 Ler e salvar arquivos:

Data Formate	Read	Save
csv	pd.read_csv()	df.to_csv()
json	pd.read_json()	df.to_json()
excel	pd.read_excel()	df.to_excel()
hdf	pd.read_hdf()	df.to_hdf()
sql	pd.read_sql()	df.to_sql()
...

Obs.: Lendo arquivos de Excel:

```
» xlsx = pd.ExcelFile('seu_arquivo_excel.xlsx')
» df = pd.read_excel(xlsx, 'Planilha 1')
```



2 Visualizando o DataFrame:

Comando	Descrição
<code>df.head()</code>	Exibe as primeiras cinco linhas
<code>df.tail()</code>	Exibe as últimas cinco linhas
<code>df.index</code>	Exibe o index da tabela
<code>df.columns</code>	Exibe as categorias da tabela
<code>df.x.unique()</code>	Exibe os elementos únicos de uma coluna x

3 Selecionando linhas e colunas

Comando	Descrição
<code>df['c1']</code>	Extrai uma coluna específica c1
<code>df[['c1', 'c2']]</code>	Extrai as colunas c1 e c2
<code>df.loc['row name']</code>	Extrai uma linha com base no seu nome
<code>df.iloc[row index]</code>	Extrai uma linha com base no seu index
<code>df.drop(i, index=0)</code>	Remove a linha i
<code>df.drop(c1, index=1)</code>	Remove a coluna c1



4 Resumos dos dados do DataFrame:

Comando	Descrição
<code>df.sum()</code>	Soma dos valores de um DataFrame
<code>df.min()</code>	Menor valor de um DataFrame
<code>df.max()</code>	Maior valor de um DataFrame
<code>df.idxmin()</code>	Index do menor valor
<code>df.idxmax()</code>	Index do maior valor
<code>df.mean()</code>	Média dos valores
<code>df.median()</code>	Mediana dos valores
<code>df.dtypes</code>	Tipos de cada coluna (int64, object, etc.)
<code>df.info()</code>	Informações (dtype, valores não nulos, etc.)
<code>df.describe()</code>	Resumo estatístico (quartis, mediana, etc.)



LET'S
START!



Google Colaboratory - Parte I

pandas-basic.ipynb

File Edit View Insert Runtime Tools Help

CODE TEXT CELL CELL

COMMENT SHARE CONNECT EDITING

Minicurso: Análise e Manipulação de Dados com Python: Parte 1

Minicurso: Análise e Manipulação de Dados com Python
Instrutor: Humberto da Silva Neto

Aluno:

Tabela de conteúdos:

Parte 1:

1. [Introdução ao Pandas](#)
2. [Correção dos dados](#)
 - o [Identificando e lidando com valores ausentes](#)
 - o [Corrigindo os tipos dos dados](#)
3. [Padronização de dados](#)
4. [Normalização de dados](#)
5. [Binning](#)
6. [Correlação](#)

Parte 2:

7. Preparando os dados
8. Visualização de dados usando Matplotlib
 - o Gráficos de Linha



Correção dos dados



1985 Auto Imports Database

Source Information

- Creator/Donor: Jeffrey C. Schlimmer
- Date: 19 May 1987
- Tabela

Description

This data set consists of three types of entities: (a) the specification of an auto in terms of various characteristics, (b) its assigned insurance risk rating, (c) its normalized losses in use as compared to other cars.

Missing Attribute Values: (denoted by "?")

#2 – 41 #6 – 2 #19 – 4 #20 – 4 #22 – 2 #23 – 2 #26 – 4



Primeiros passos

Importando a tabela:

- 1 Importe o arquivo .csv



Primeiros passos

Importando a tabela:

- 1 Importe o arquivo .csv
- 2 Importe o arquivo .csv **corretamente** (*sem cabeçalho*)



Primeiros passos

Importando a tabela:

- 1 Importe o arquivo .csv
- 2 Importe o arquivo .csv **corretamente** (*sem cabeçalho*)
- 3 Adicione o cabeçalho indicado



Primeiros passos

Importando a tabela:

- 1 Importe o arquivo .csv
- 2 Importe o arquivo .csv **corretamente** (*sem cabeçalho*)
- 3 Adicione o cabeçalho indicado

```
df.columns = headers
```



Primeiros passos

Importando a tabela:

- 1 Importe o arquivo .csv
- 2 Importe o arquivo .csv **corretamente** (*sem cabeçalho*)
- 3 Adicione o cabeçalho indicado

```
df.columns = headers
```

- 4 Verifique:



Primeiros passos

Importando a tabela:

- 1 Importe o arquivo .csv
- 2 Importe o arquivo .csv **corretamente** (*sem cabeçalho*)
- 3 Adicione o cabeçalho indicado

```
df.columns = headers
```

- 4 Verifique:
 - 1 As informações básicas do DataFrame



Primeiros passos

Importando a tabela:

- 1 Importe o arquivo .csv
- 2 Importe o arquivo .csv **corretamente** (*sem cabeçalho*)
- 3 Adicione o cabeçalho indicado

```
df.columns = headers
```

- 4 Verifique:
 - 1 As informações básicas do DataFrame

```
df.info()
```



Primeiros passos

Importando a tabela:

- 1 Importe o arquivo .csv
- 2 Importe o arquivo .csv **corretamente** (*sem cabeçalho*)
- 3 Adicione o cabeçalho indicado

```
df.columns = headers
```

- 4 Verifique:
 - 1 As informações básicas do DataFrame

```
df.info()
```

- 2 Os dados estatísticos do DataFrame



Primeiros passos

Importando a tabela:

- 1 Importe o arquivo .csv
- 2 Importe o arquivo .csv **corretamente** (*sem cabeçalho*)
- 3 Adicione o cabeçalho indicado

```
df.columns = headers
```

- 4 Verifique:

- 1 As informações básicas do DataFrame

```
df.info()
```

- 2 Os dados estatísticos do DataFrame

```
df.describe()
```



Valores ausentes: "?" → NaN

O marcador padrão do Pandas para dados ausentes é **NaN**. Contudo, a tabela importada utilizava o termo "?" para essa situação.



Valores ausentes: "?" → NaN

O marcador padrão do Pandas para dados ausentes é **NaN**. Contudo, a tabela importada utilizava o termo "?" para essa situação.

Perguntas:

- 1 Qual o problema?



Valores ausentes: "?" → NaN

O marcador padrão do Pandas para dados ausentes é **NaN**. Contudo, a tabela importada utilizava o termo "?" para essa situação.

Perguntas:

- 1 Qual o problema? O DataFrame vai entender "?" como um objeto (string) qualquer. Por causa disso, os métodos intrínsecos do Pandas não podem ser usados para resolvê-los.



Valores ausentes: "?" → NaN

O marcador padrão do Pandas para dados ausentes é **NaN**. Contudo, a tabela importada utilizava o termo "?" para essa situação.

Perguntas:

- 1 Qual o problema? O DataFrame vai entender "?" como um objeto (string) qualquer. Por causa disso, os métodos intrínsecos do Pandas não podem ser usados para resolvê-los.
- 2 Como resolver?



Valores ausentes: "?" → NaN

O marcador padrão do Pandas para dados ausentes é **NaN**. Contudo, a tabela importada utilizava o termo "?" para essa situação.

Perguntas:

- 1 Qual o problema? O DataFrame vai entender "?" como um objeto (string) qualquer. Por causa disso, os métodos intrínsecos do Pandas não podem ser usados para resolvê-los.
- 2 Como resolver? Iremos substituir todos as ocorrências de "?" para NaN



Valores ausentes: "?" → NaN

O marcador padrão do Pandas para dados ausentes é **NaN**. Contudo, a tabela importada utilizava o termo "?" para essa situação.

Perguntas:

- 1 Qual o problema? O DataFrame vai entender "?" como um objeto (string) qualquer. Por causa disso, os métodos intrínsecos do Pandas não podem ser usados para resolvê-los.
- 2 Como resolver? Iremos substituir todos as ocorrências de "?" para NaN

```
df.replace("?", np.nan, inplace=True)
```



Valores ausentes: "?" → NaN

O marcador padrão do Pandas para dados ausentes é **NaN**. Contudo, a tabela importada utilizava o termo "?" para essa situação.

Perguntas:

- 1 Qual o problema? O DataFrame vai entender "?" como um objeto (string) qualquer. Por causa disso, os métodos intrínsecos do Pandas não podem ser usados para resolvê-los.
- 2 Como resolver? Iremos substituir todos as ocorrências de "?" para NaN

```
df.replace("?", np.nan, inplace=True)
```

- 3 Como encontrar esses elementos?



Valores ausentes: "?" → NaN

O marcador padrão do Pandas para dados ausentes é **NaN**. Contudo, a tabela importada utilizava o termo "?" para essa situação.

Perguntas:

- 1 Qual o problema? O DataFrame vai entender "?" como um objeto (string) qualquer. Por causa disso, os métodos intrínsecos do Pandas não podem ser usados para resolvê-los.
- 2 Como resolver? Iremos substituir todos as ocorrências de "?" para NaN

```
df.replace("?", np.nan, inplace=True)
```

- 3 Como encontrar esses elementos?

```
df.isnull()
```



Lidando com os valores ausentes

Perguntas:

- 1 Como lidar com esses dados?



Lidando com os valores ausentes

Perguntas:

1 Como lidar com esses dados?

1 Descartar dados

- Descartar a linha inteira
- Descartar a coluna inteira



Lidando com os valores ausentes

Perguntas:

1 Como lidar com esses dados?

1 Descartar dados

- Descartar a linha inteira
- Descartar a coluna inteira

2 Substituir dados

- Substituir pela média
- Substituir pela frequência
- Substituir baseando-se em outras funções

2 Qual dos métodos devemos aplicar para as seguintes colunas? Por que?

bore	horsepower	normalized-losses	num-of-doors
peak-rpm	price	stroke	
Descartar linhas	Frequência	Média	



Lidando com os valores ausentes

Perguntas:

1 Como lidar com esses dados?

1 Descartar dados

- Descartar a linha inteira
- Descartar a coluna inteira

2 Substituir dados

- Substituir pela média
- Substituir pela frequência
- Substituir baseando-se em outras funções

2 Qual dos métodos devemos aplicar para as seguintes colunas? Por que?

bore	horsepower	normalized-losses	num-of-doors
peak-rpm	price	stroke	
Descartar linhas	Frequência	Média	



Lidando com os valores ausentes

Perguntas:

1 Como lidar com esses dados?

1 Descartar dados

- Descartar a linha inteira
- Descartar a coluna inteira

2 Substituir dados

- Substituir pela média
- Substituir pela frequência
- Substituir baseando-se em outras funções

2 Qual dos métodos devemos aplicar para as seguintes colunas? Por que?

bore	horsepower	normalized-losses	num-of-doors
peak-rpm	price	stroke	
Descartar linhas	Frequência	Média	



Lidando com os valores ausentes

Perguntas:

- ## 1 Como lidar com esses dados?

1 Descartar dados

- Descartar a linha inteira
 - Descartar a coluna inteira

2 Substituir dados

- Substituir pela média
 - Substituir pela frequência
 - Substituir baseando-se em outras funções

- 2 Qual dos métodos devemos aplicar para as seguintes colunas? Por que?

bore	horsepower	normalized-losses	num-of-doors
peak-rpm	price	stroke	
Descartar linhas	Frequência	Média	



Corrigindo o formato dos dados

Perguntas:

- 1 Como verificar o tipo dos dados?



Corrigindo o formato dos dados

Perguntas:

- 1 Como verificar o tipo dos dados?

```
df.dtypes
```



Corrigindo o formato dos dados

Perguntas:

- 1 Como verificar o tipo dos dados?

```
df.dtypes
```

- 2 Por que esse problema aconteceu?



Corrigindo o formato dos dados

Perguntas:

- 1 Como verificar o tipo dos dados?

```
df.dtypes
```

- 2 Por que esse problema aconteceu? Por algum motivo havia elementos nessa linha que eram uma string. Ex.: "?"



Corrigindo o formato dos dados

Perguntas:

- 1 Como verificar o tipo dos dados?

```
df.dtypes
```

- 2 Por que esse problema aconteceu? Por algum motivo havia elementos nessa linha que eram uma string. Ex.: "?"
- 3 Qual tipo as colunas abaixo deveriam ter?

bore	horsepower	normalized-losses
peak-rpm	price	stroke
STRING	INT	FLOAT



Corrigindo o formato dos dados

Perguntas:

- 1 Como verificar o tipo dos dados?

```
df.dtypes
```

- 2 Por que esse problema aconteceu? Por algum motivo havia elementos nessa linha que eram uma string. Ex.: "?"
- 3 Qual tipo as colunas abaixo deveriam ter?

bore	horsepower	normalized-losses
peak-rpm	price	stroke
STRING	INT	FLOAT



Corrigindo o formato dos dados

Perguntas:

- 1 Como verificar o tipo dos dados?

```
df.dtypes
```

- 2 Por que esse problema aconteceu? Por algum motivo havia elementos nessa linha que eram uma string. Ex.: "?"
- 3 Qual tipo as colunas abaixo deveriam ter?

bore	horsepower	normalized-losses
peak-rpm	price	stroke
STRING	INT	FLOAT



Corrigindo o formato dos dados

Perguntas:

- 1 Como verificar o tipo dos dados?

```
df.dtypes
```

- 2 Por que esse problema aconteceu? Por algum motivo havia elementos nessa linha que eram uma string. Ex.: "?"
- 3 Qual tipo as colunas abaixo deveriam ter?

bore	horsepower	normalized-losses
peak-rpm	price	stroke
STRING	INT	FLOAT



Padronização e Normalização de dados



Padronização de dados

Perguntas:

- 1 O que é padronização?



Padronização de dados

Perguntas:

- 1 O que é padronização? Padronização é o processo de transformar dados em um formato comum que permite ao pesquisador fazer uma comparação significativa.



Padronização de dados

Perguntas:

- 1 O que é padronização? Padronização é o processo de transformar dados em um formato comum que permite ao pesquisador fazer uma comparação significativa.

Exemplo:

- 1 milhas por galão → L/100km



Padronização de dados

Perguntas:

- 1 O que é padronização? Padronização é o processo de transformar dados em um formato comum que permite ao pesquisador fazer uma comparação significativa.

Exemplo:

- 1 milhas por galão → L/100km

Questão:

- 1 Crie as colunas "city-L/100km" e "highway-L/100km" com os valores convertidos de mpg para L/100km das colunas "city-mpg" e "highway-mpg", respectivamente.



Normalização de dados

Perguntas:

- 1 O que é normalização?



Normalização de dados

Perguntas:

- 1 O que é normalização? Normalização é o processo de transformar valores de diversas variáveis em um intervalo similar. Normalizações típicas incluem escalar a variável de modo que :
 - a média da variável seja 0;
 - a variância da variável seja 1;
 - os valores da variável variem de 0 a 1.



Normalização de dados

Perguntas:

- 1 O que é normalização? Normalização é o processo de transformar valores de diversas variáveis em um intervalo similar. Normalizações típicas incluem escalar a variável de modo que :
- a média da variável seja 0;
 - a variância da variável seja 1;
 - os valores da variável variem de 0 a 1.

Questão:

- 1 Normalize as colunas "length", "width" e "height" entre 0 e 1.



Binning



Binning

Perguntas:

- 1 O que é *binning*?



Binning

Perguntas:

- 1 O que é *binning*? *Binning* é um processo de transformação de variáveis numéricas contínuas em classes discretas e categóricas, para análises agrupadas.



Binning

Perguntas:

- 1 O que é *binning*? *Binning* é um processo de transformação de variáveis numéricas contínuas em classes discretas e categóricas, para análises agrupadas.

Exemplo:

- 1 Em nosso conjunto de dados, "horsepower" é uma variável de valor real variando de 48 a 288, possui 57 valores únicos. Que tal rearranjá-los em três classes para simplificar a análise?



Correlação



Correlação

Perguntas:

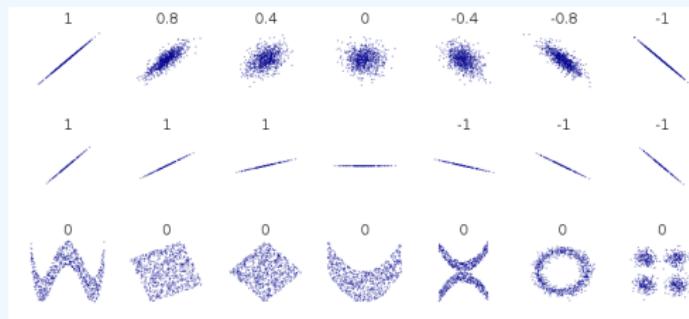
- 1 O que é correlação?



Correlação

Perguntas:

- O que é correlação? Na estatística o coeficiente de correlação de Pearson (r), que também é chamado de coeficiente de correlação produto-momento, mede a relação que existe entre duas variáveis dentro de uma mesma escala métrica.



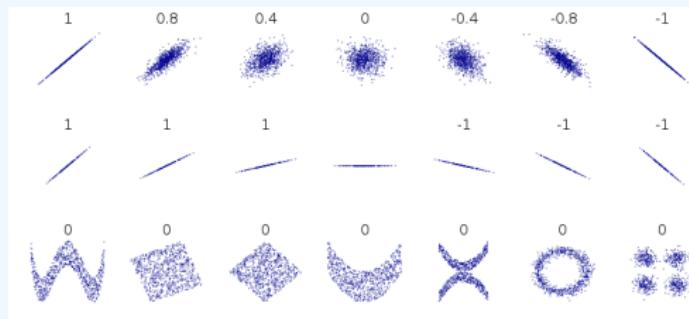
Fonte: [Wikipedia](#)



Correlação

Perguntas:

- O que é correlação? Na estatística o coeficiente de correlação de Pearson (r), que também é chamado de coeficiente de correlação produto-momento, mede a relação que existe entre duas variáveis dentro de uma mesma escala métrica.



Fonte: [Wikipedia](#)

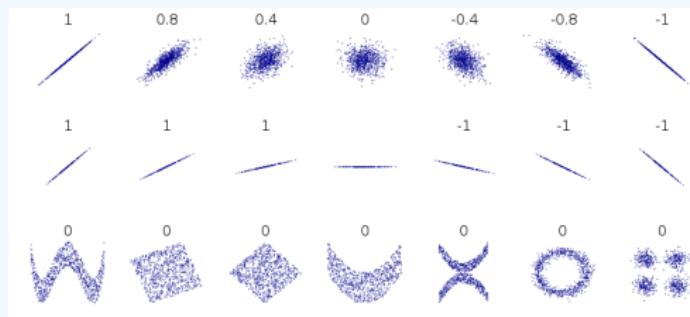
- Como verificar a correlação entre as variáveis de um DataFrame?



Correlação

Perguntas:

- O que é correlação? Na estatística o coeficiente de correlação de Pearson (r), que também é chamado de coeficiente de correlação produto-momento, mede a relação que existe entre duas variáveis dentro de uma mesma escala métrica.



Fonte: [Wikipedia](#)

- Como verificar a correlação entre as variáveis de um DataFrame?

```
df.corr()
```



Gráfico de dispersão + Regressão Linear

Figure: Correlação positiva

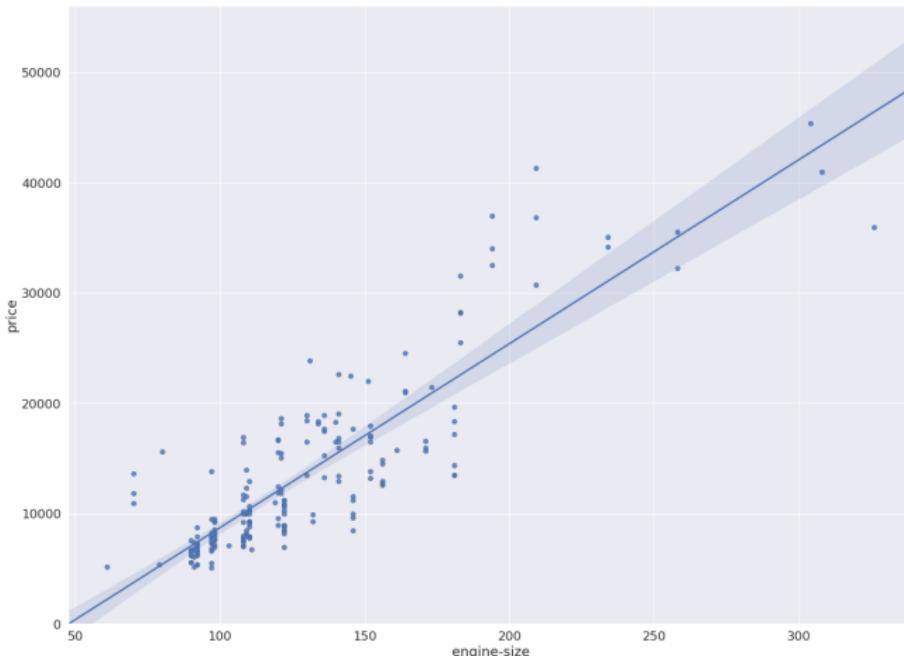


Gráfico de dispersão + Regressão Linear

Figure: Correlação negativa

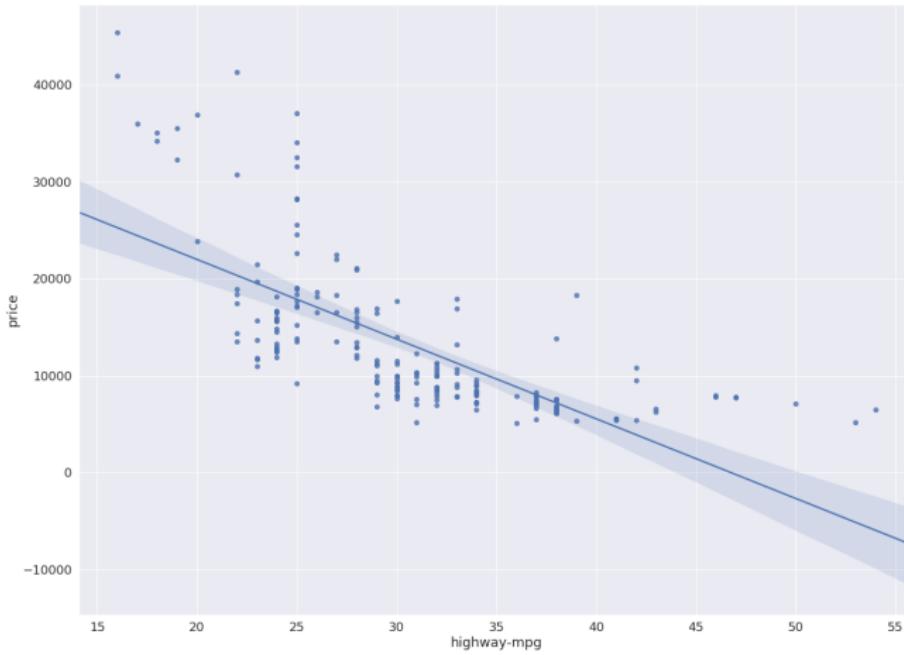


Gráfico de dispersão + Regressão Linear

Figure: Correlação próxima de zero

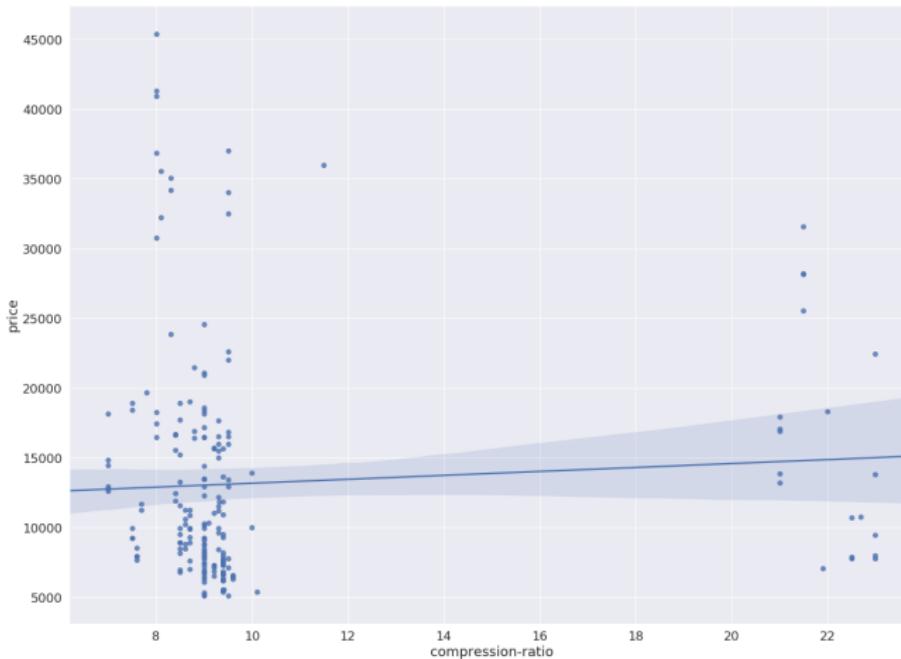
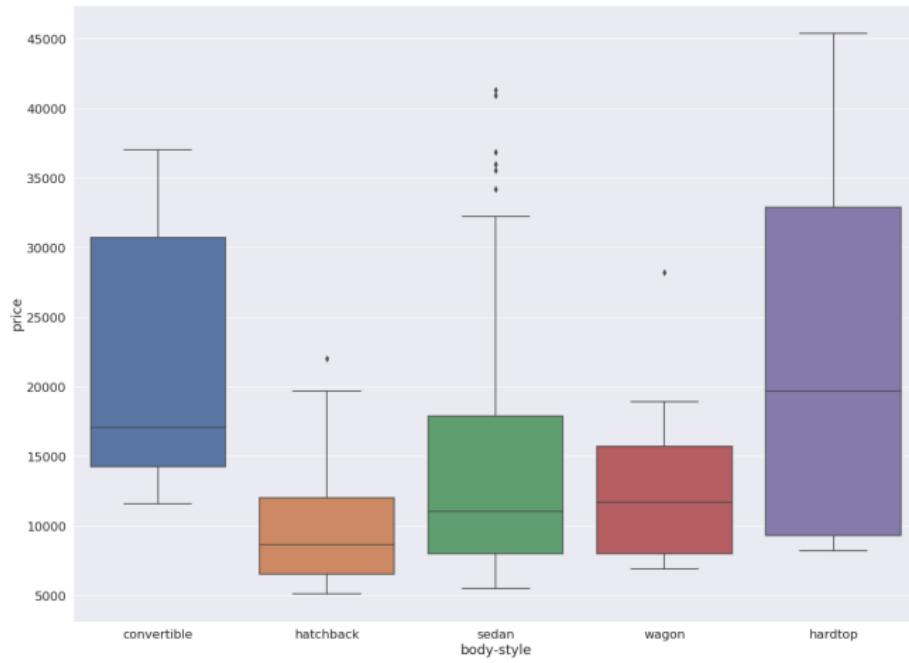


Diagrama de caixa



Visualização dos dados



The Dataset: Immigration to Canada from 1980 to 2013

Canada.xlsx (read-only) - LibreOffice Calc

File Edit View Insert Format Styles Sheet Data Tools Window Help

A1:U20 Classification



**United Nations
Population Division
Department of Economic and Social Affairs**

International Migration Flows to and from Selected Countries: The 2015 Revision

POP/DB/MIG/FlowRev.2015
December 2015 - Copyright © 2015 by United Nations. All rights reserved
Suggested citation: United Nations, Department of Economic and Social Affairs, Population Division (2015). International Migration Flows to and from Selected Countries: The 2015 Revision. (United Nations database, POP/DB/MIG/FlowRev.2015).

Reporting country: Canada
Criterion: Citizenship

Classification	Origin/Destination	Area	Major area	Region	Development region								
						DEV	DevName	1980	1981	1982	1983	1984	1985
21	Type	Coverage	OdName	AREA	AreaName	REG	RegName	DEV	DevName	1980	1981	1982	1983
22	Immigrants	F-foreigners	Afghanistan	935	Asia	5501	Southern Asia	1902	Developing regions	16	39	39	47
23	Immigrants	F-foreigners	Albania	908	Europe	925	Southern Europe	1901	Developed regions	1	0	0	0
24	Immigrants	F-foreigners	Algeria	901	Africa	911	North Africa	1902	Developing regions	71	90	63	44
25	Immigrants	F-foreigners	American Samoa	909	Oceania	957	Polynesia	1902	Developing regions	0	1	0	0
26	Immigrants	F-foreigners	Andorra	908	Europe	925	Southern Europe	1901	Developed regions	0	0	0	0
27	Immigrants	F-foreigners	Angola	903	Africa	911	Middle Africa	1902	Developing regions	1	3	6	6
28	Immigrants	F-foreigners	Antigua and Barbuda	904	Latin America and the Caribbean	915	Caribbean	1902	Developing regions	0	0	42	50
29	Immigrants	F-foreigners	Argentina	902	Latin America and the Caribbean	914	Latin America	1902	Developing regions	364	426	650	241
30	Immigrants	F-foreigners	Armenia	905	Asia	922	Western Asia	1902	Developing regions	0	0	0	0
31	Immigrants	F-foreigners	Australia	909	Oceania	927	Australia and New Zealand	1901	Developed regions	702	639	484	317
32	Immigrants	F-foreigners	Austria	908	Europe	926	Western Europe	1901	Developed regions	234	238	201	117
33	Immigrants	F-foreigners	Azerbaijan	935	Asia	922	Western Asia	1902	Developing regions	0	0	0	0
34	Immigrants	F-foreigners	Bahrain	902	Latin America and the Caribbean	914	Latin America	1902	Developing regions	20	25	35	14
35	Immigrants	F-foreigners	Bahrain	935	Asia	922	Western Asia	1902	Developing regions	0	0	0	0
36	Immigrants	F-foreigners	Bangladesh	935	Asia	5501	Southern Asia	1902	Developing regions	83	84	86	81
37	Immigrants	F-foreigners	Barbados	904	Latin America and the Caribbean	915	Caribbean	1902	Developing regions	372	376	299	244
38	Immigrants	F-foreigners	Belarus	908	Europe	923	Eastern Europe	1901	Developed regions	0	0	0	0
39	Immigrants	F-foreigners	Belgium	908	Europe	926	Western Europe	1901	Developed regions	511	543	519	297
40	Immigrants	F-foreigners	Belize	904	Latin America and the Caribbean	916	Central America	1902	Developing regions	16	27	13	23

Regions by Citizenship Canada by Citizenship [2] Canada by Citizenship [2] PageStyle_Canada by Citizenship

Sheet 2 of 3 20 rows, 21 columns selected Average: ; Sum: 0 100%

Aplicando correções

Importando a tabela:

- 1 Importe a tabela
- 2 Remover as colunas: ['AREA', 'REG', 'DEV', 'Type', 'Coverage']



Aplicando correções

Importando a tabela:

- 1 Importe a tabela
- 2 Remover as colunas: ['AREA', 'REG', 'DEV', 'Type', 'Coverage']

```
df.drop(columns, axis=1, inplace=True)
```



Aplicando correções

Importando a tabela:

- 1 Importe a tabela
- 2 Remover as colunas: `['AREA', 'REG', 'DEV', 'Type', 'Coverage']`

```
df.drop(columns, axis=1, inplace=True)
```

- 3 Renomeie as colunas para melhor compreensão



Aplicando correções

Importando a tabela:

1 Importe a tabela

2 Remover as colunas: ['AREA', 'REG', 'DEV', 'Type', 'Coverage']

```
df.drop(columns, axis=1, inplace=True)
```

3 Renomeie as colunas para melhor compreensão

```
df.rename(columns = {OldName : NewName},  
          inplace=True)
```



Aplicando correções

Importando a tabela:

1 Importe a tabela

2 Remover as colunas: ['AREA', 'REG', 'DEV', 'Type', 'Coverage']

```
df.drop(columns, axis=1, inplace=True)
```

3 Renomeie as colunas para melhor compreensão

```
df.rename(columns = {OldName : NewName},  
          inplace=True)
```

4 Crie uma coluna que represente a soma do número de imigrantes de 1980 a 2013



Aplicando correções

Importando a tabela:

1 Importe a tabela

2 Remover as colunas: ['AREA', 'REG', 'DEV', 'Type', 'Coverage']

```
df.drop(columns, axis=1, inplace=True)
```

3 Renomeie as colunas para melhor compreensão

```
df.rename(columns = {OldName : NewName},  
          inplace=True)
```

4 Crie uma coluna que represente a soma do número de imigrantes de 1980 a 2013

```
df['Total'] = df.sum(axis=1)
```

5 Verifique se há a ocorrência de valores ausentes



Indexação e Seleção

Selecionando colunas:

- 1 `df.column_name` → [retorna Serie]
- 2 `df["coluna"]` → [retorna Serie]
- 3 `df[["coluna 1", "coluna 2"]]` → [retorna DataFrame]

Selecionando linhas:

- 1 `df.loc[label]` → [por rótulos]
- 2 `df.iloc[index]` → [por índices]



Gráfico de linhas

Gráfico de linhas – Haiti

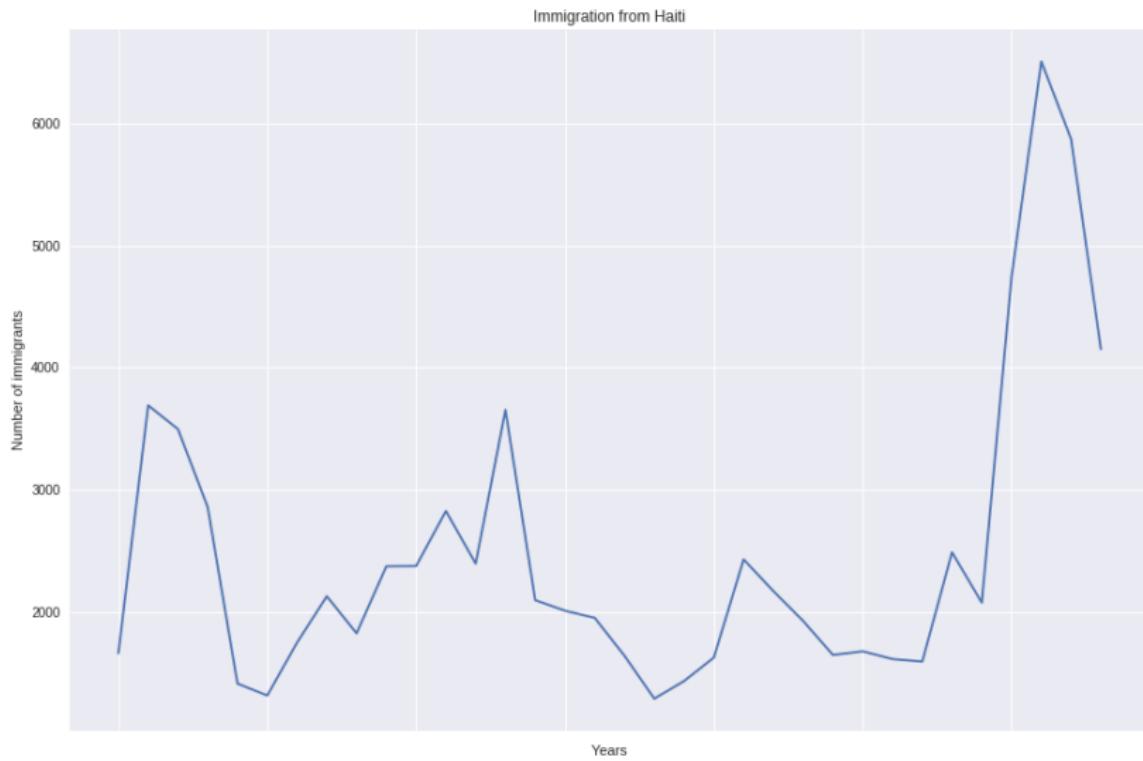


Gráfico de linhas

Gráfico de linhas – China e Índia

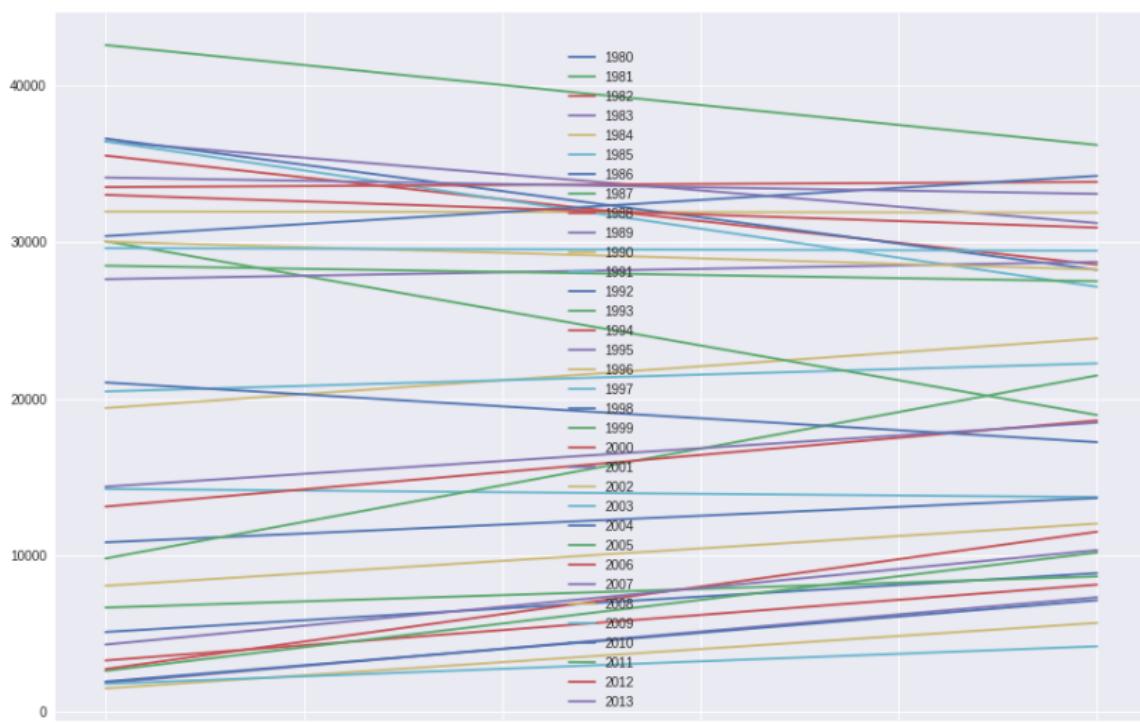


Gráfico de linhas – China e Índia

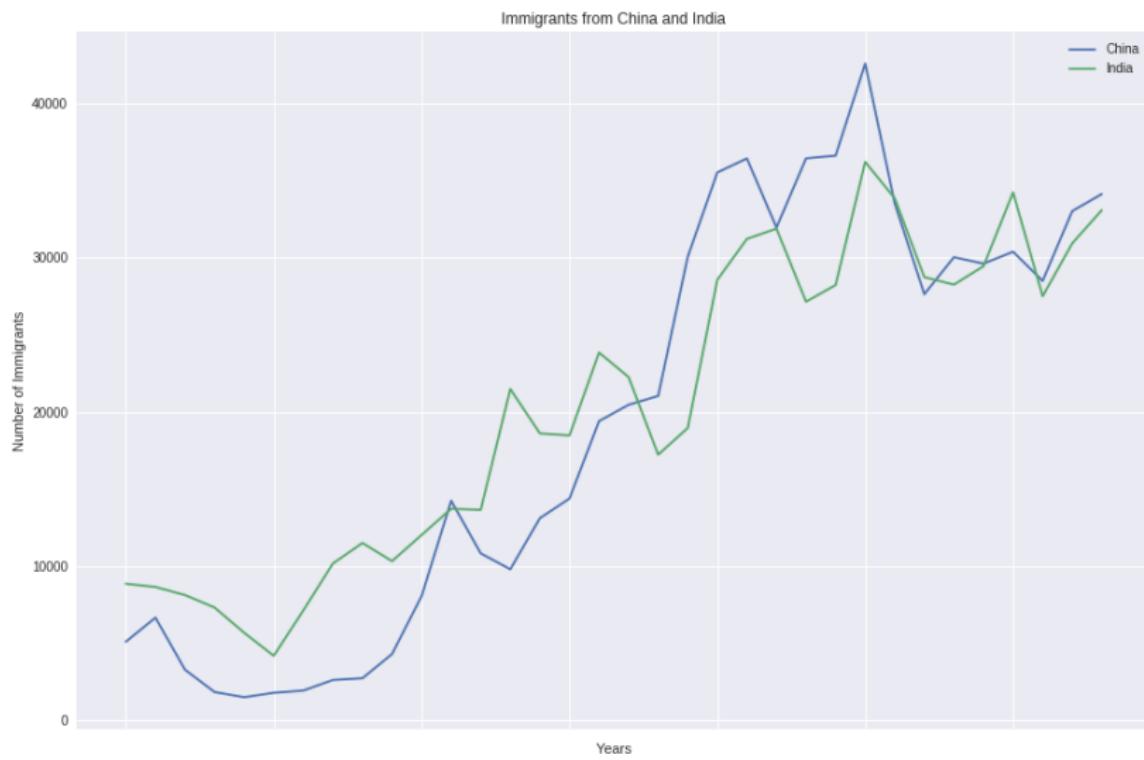


Gráfico de linhas

Gráfico de linhas – Top 5

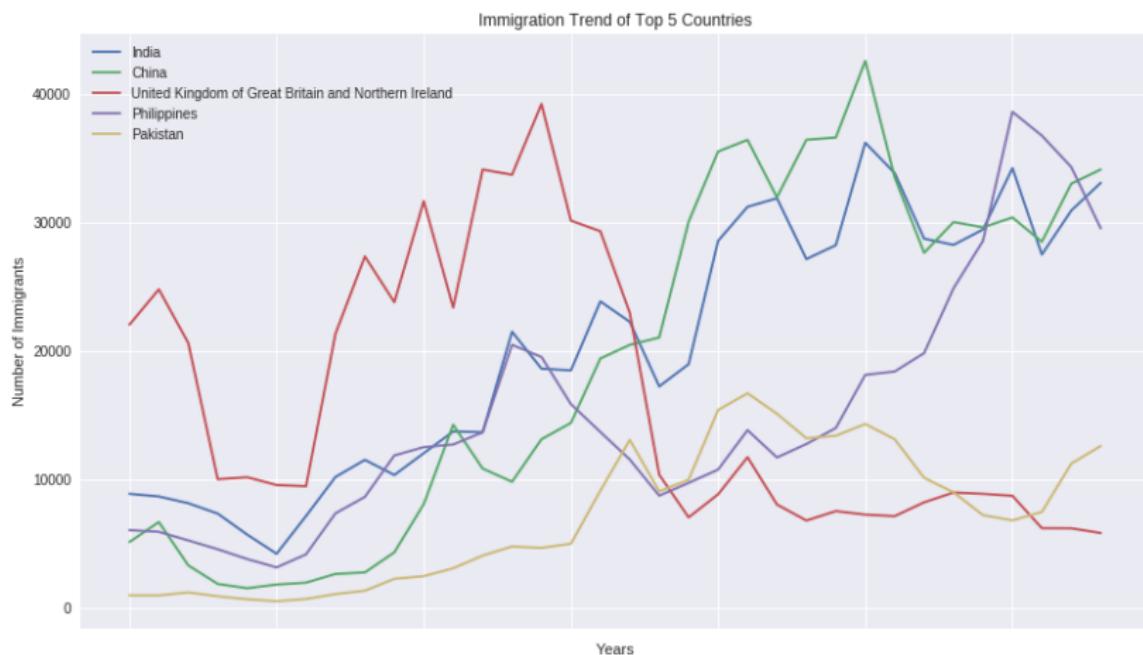


Gráfico de área

Gráfico de área - Top 5

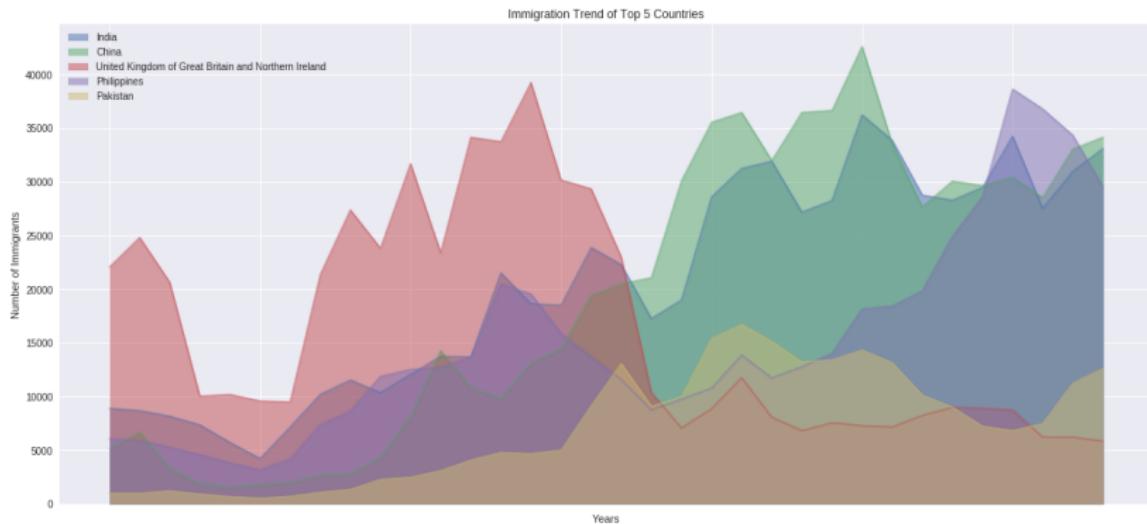
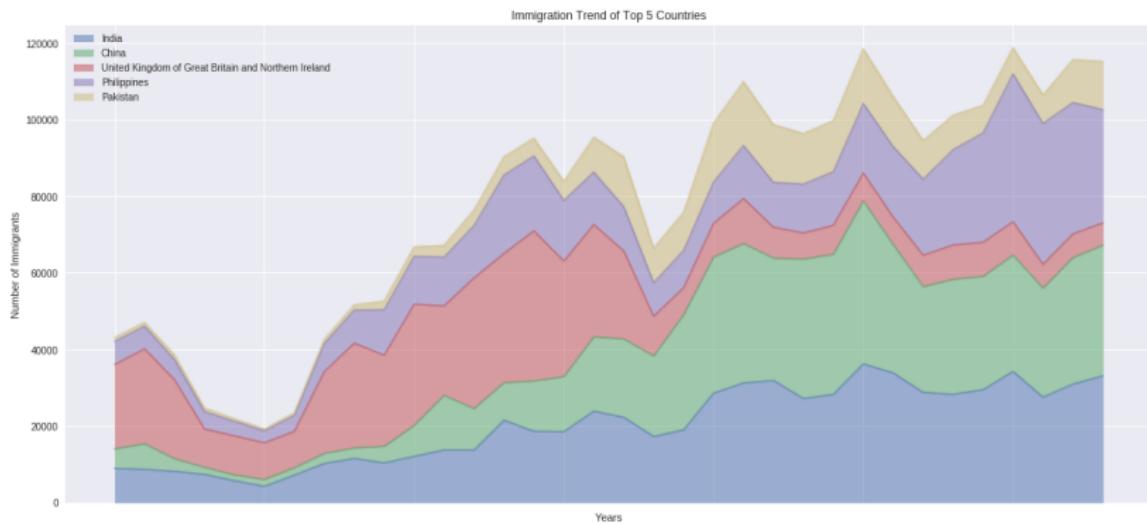


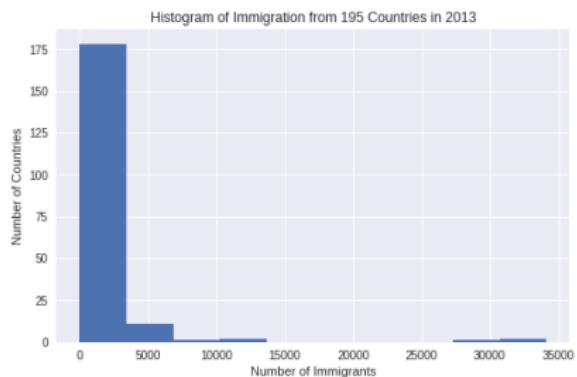
Gráfico de área

Gráfico de área - Top 5



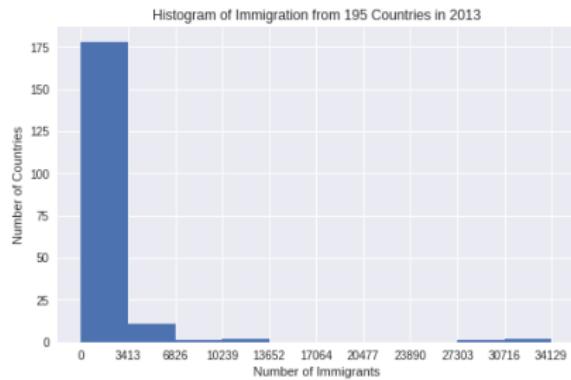
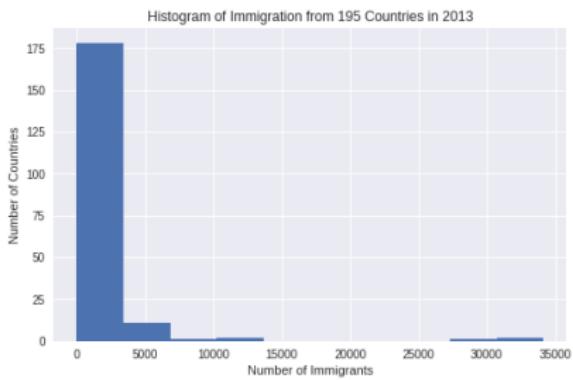
Histogramas

Histogramas – 2013



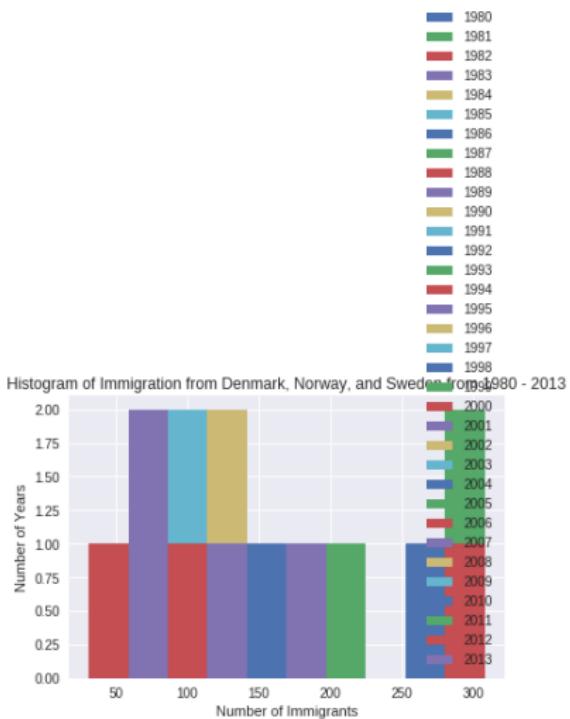
Histogramas

Histogramas – 2013



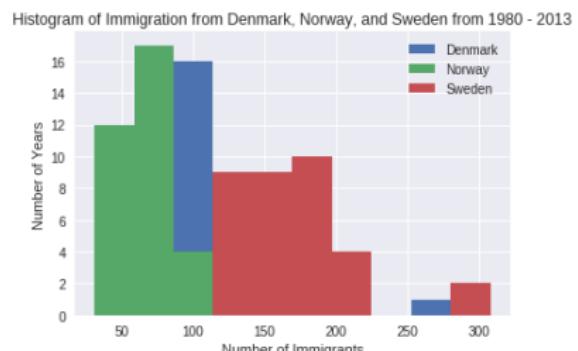
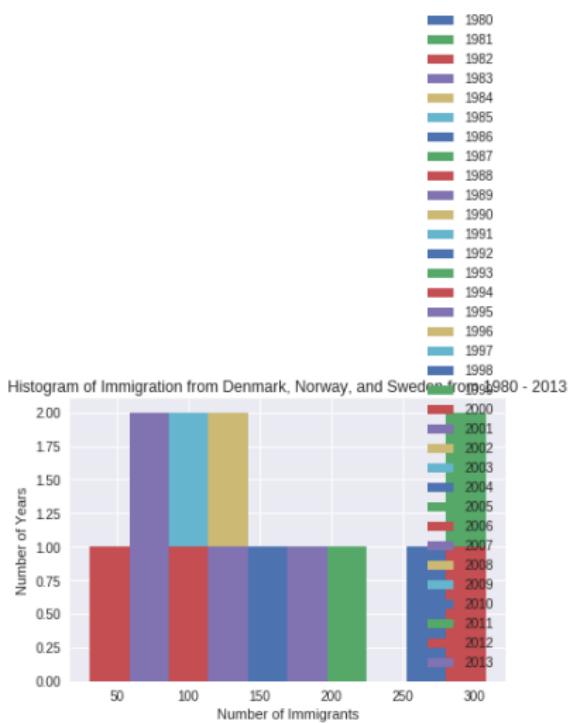
Histogramas

Histogramas – Dinamarca, Noruega, Suécia



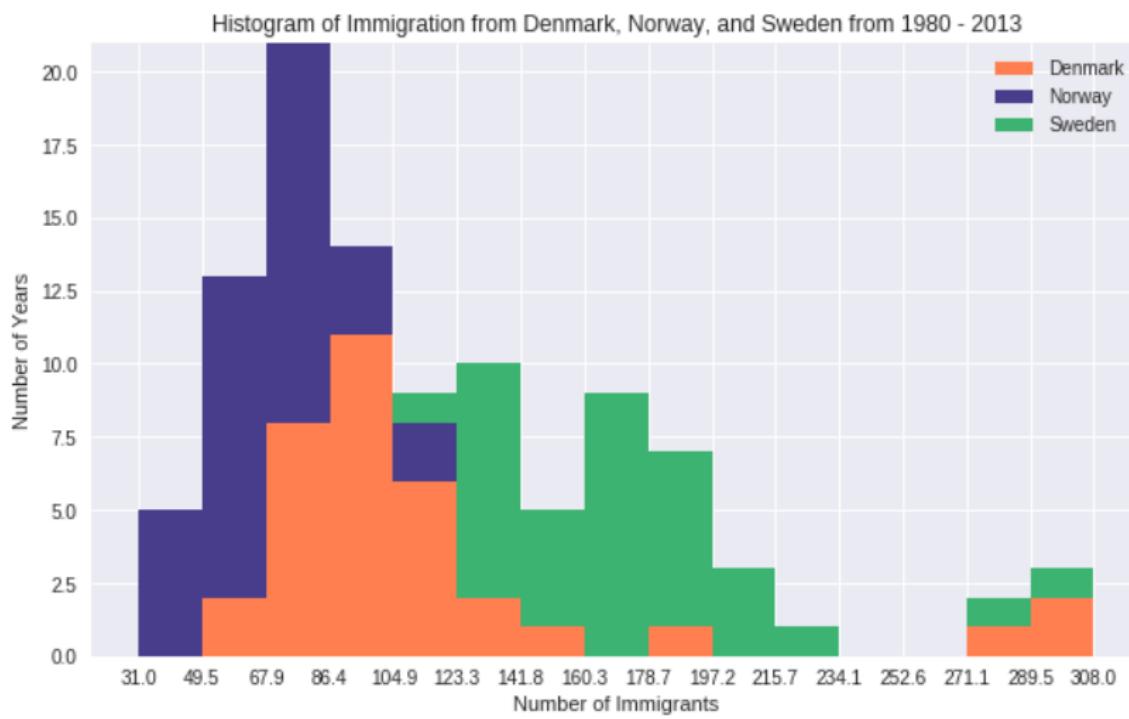
Histogramas

Histogramas – Dinamarca, Noruega, Suécia



Histogramas

Histogramas – Dinamarca, Noruega, Suécia



Barras Verticais – Islândia

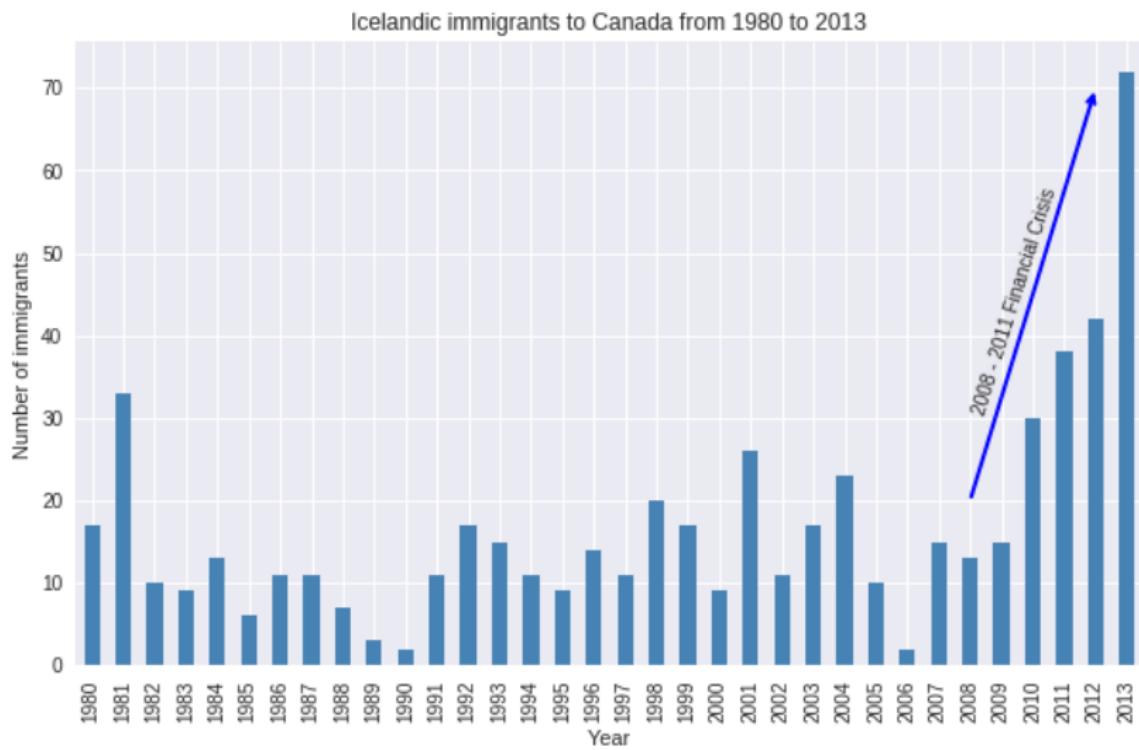


Gráfico de barras

Barra Horizontais – Top 15

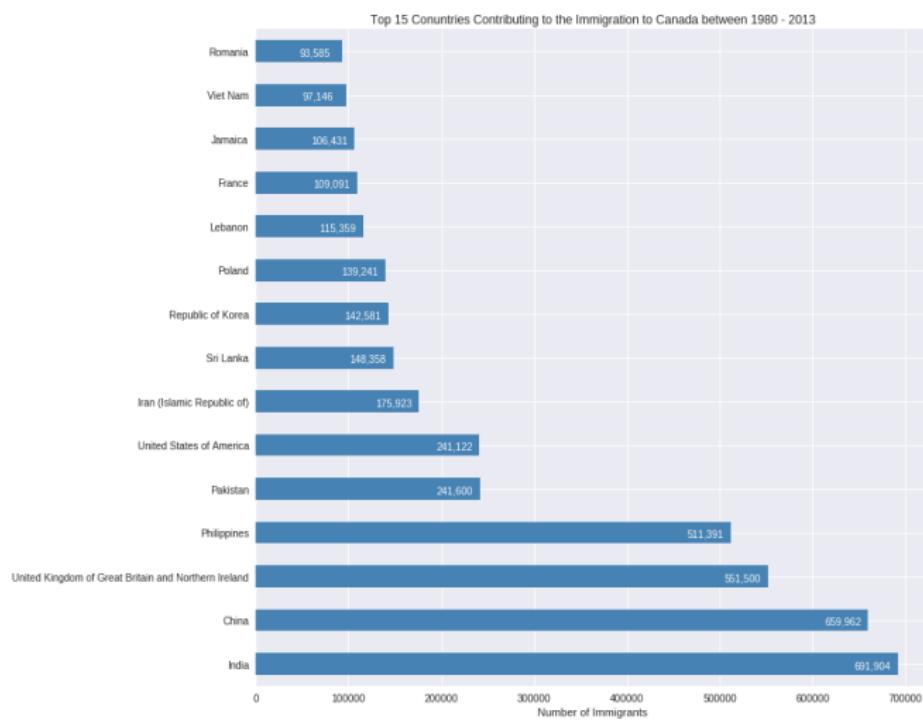


Gráfico de pizza

Gráfico de pizza – Continentes

Total

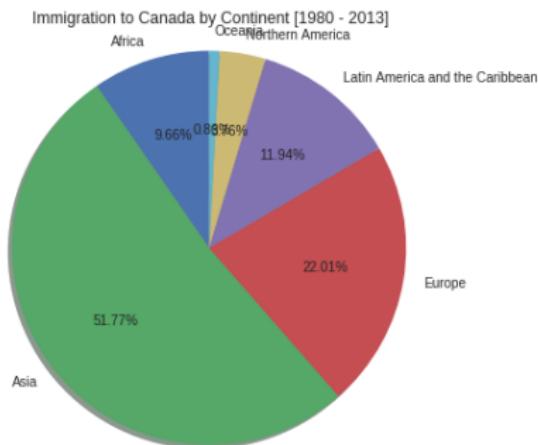


Gráfico de pizza

Gráfico de pizza – Continentes

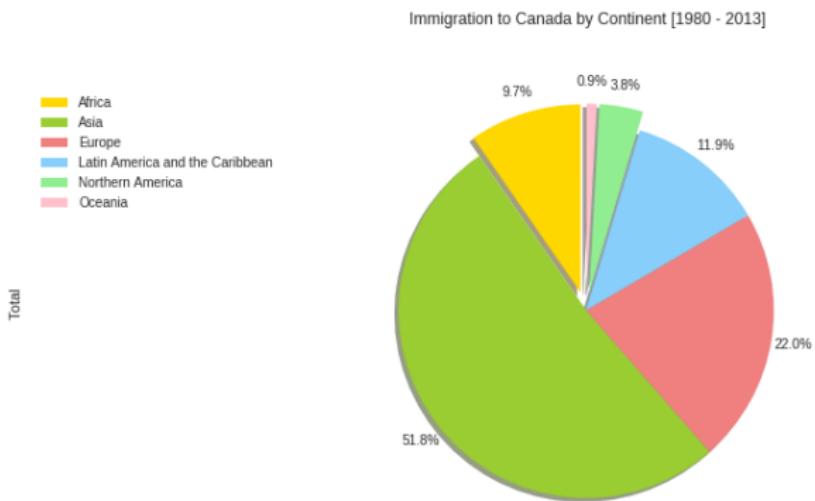


Diagrama de caixa – 80s, 90s e 00s

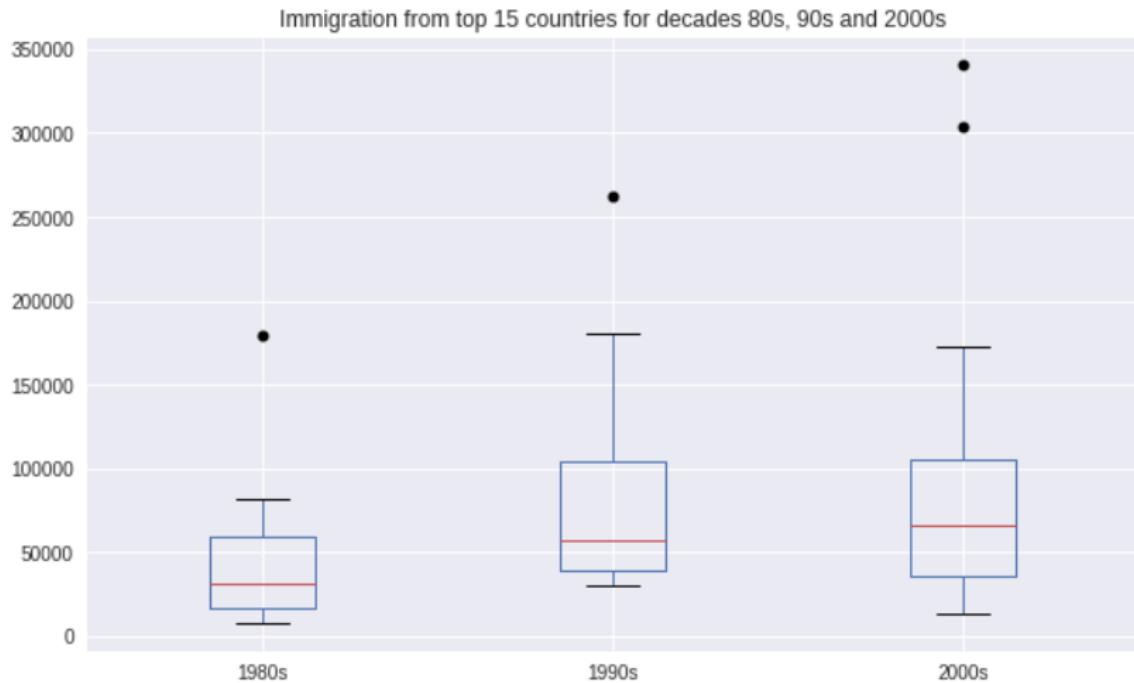
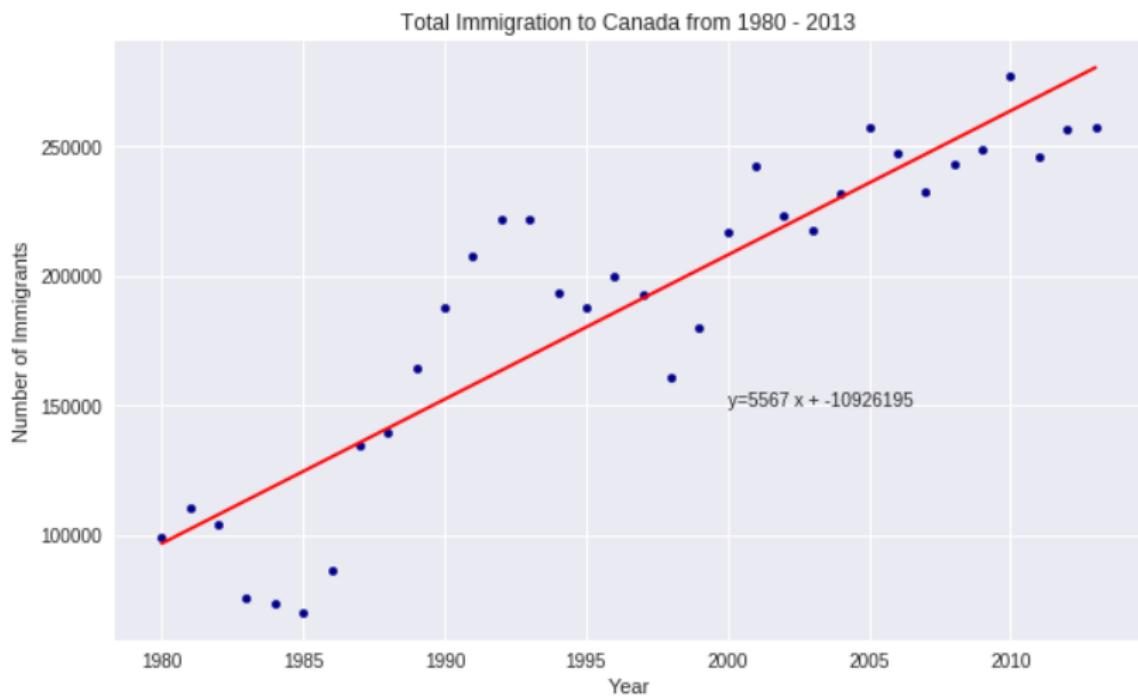


Gráfico de dispersão

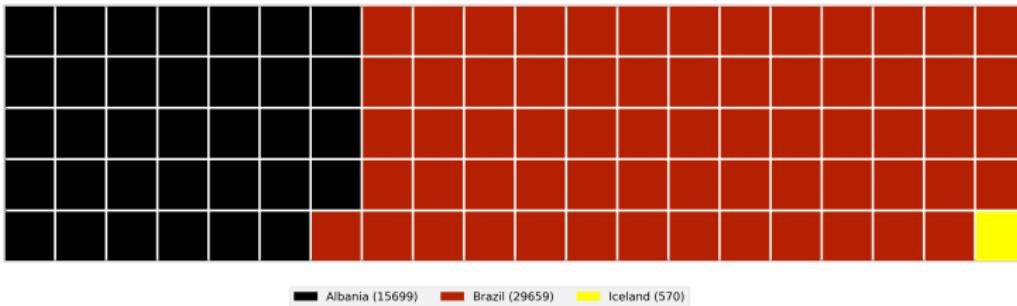
Gráfico de dispersão – Total



Extras



Waffle Charts



Word Clouds



Cheat Sheet

Python For Data Science Cheat Sheet

Pandas Basics

Learn Python for Data Science interactively at www.DataCamp.com



Pandas

The Pandas library is built on NumPy and provides easy-to-use data structures and data analysis tools for the Python programming language.



Use the following import convention:

```
>>> import pandas as pd
```

Pandas Data Structures

Series

A one-dimensional labeled array capable of holding any data type



```
>>> s = pd.Series([1, -5, 7, 4], index=['a', 'b', 'c', 'd'])
```

DataFrame

Columns	Country	Capital	Population
Index	Bulgaria	Brussels	1110000
	India	New Delhi	1303171835
	Brazil	Brasilia	207847528

```
>>> data = {'Country': ['Belgium', 'India', 'Brazil'],
   'Capital': ['Brussels', 'New Delhi', 'Brasilia'],
   'Population': [1110000, 1303171835, 207847528]}

>>> df = pd.DataFrame(data,
   columns=['Country', 'Capital', 'Population'])
```

I/O

Read and Write to CSV

```
>>> pd.read_csv('file.csv', header=None, nrows=5)
>>> df.to_csv('myDataFrame.csv')
```

Read and Write to Excel

```
>>> pd.read_excel('file.xlsx')
>>> pd.to_excel('dir/myDataFrame.xlsx', sheet_name='Sheet1')

Read multiple sheets from the same file
>>> xls = pd.ExcelFile('file.xlsx')
>>> df = pd.read_excel(xls, 'Sheet1')
```

Asking For Help

```
>>> help(pd.Series.loc)
```

Also see NumPy Arrays

Selection

Getting

```
>>> a['b']
->
>>> df[1]
   Country    Capital  Population
1  India      New Delhi  1303171835
2  Brazil     Brasilia  207847528
```

Get one element

Get subset of a DataFrame

Selecting, Boolean Indexing & Setting

By Position

```
>>> df.iat[0,0]
'Belgium'
>>> df.iat[0,0]
'Belgium'
>>> df.iat[0,0]
'Belgium'
```

Select single value by row & column

By Label

```
>>> df.loc[0, ['Country']]
   Country
0  Belgium
>>> df.at[0, ['Country']]
   Country
0  Belgium
```

Select single value by row & column labels

By Label/Position

```
>>> df.ix[2]
   Country    Brazil
   Capital  Brasilia
   Population 207847528
```

Select single row of subset of rows

```
>>> df.ix[:, 'Capital']
0  Brasilia
1  New Delhi
2  Brasilia
```

Select a single column of subset of columns

```
>>> df.ix[1, 'Capital']
'New Delhi'
```

Select rows and columns

```
>>> df.ix[1, 'Capital']
'New Delhi'

Boolean indexing
```

Series s where value is not =>
s where value is <-1 or >2

```
>>> s[(s > 1) | (s < -1) | (s > 2)]
>>> af[df['Population']>12000000000]
```

Use filter to adjust DataFrame

Setting

```
>>> a['a'] = 6
```

Set Index a of Series a to 6

Dropping

```
>>> a.drop(['a', 'c'])
Drop values from rows (axis=0)
>>> df.drop('Country', axis=1)
Drop values from columns(axis=1)
```

(axis=0)

(axis=1)

Sort & Rank

```
>>> df.sort_index()
>>> df.sort_values(by='Country')
>>> df.rank()
```

Sort by labels along an axis
Sort by the values along an axis
Assign ranks to entries

Retrieving Series/DataFrame Information

Basic Information

```
>>> df.shape
(3, 3)
>>> df.index
Describe index
>>> df.columns
Describe DataFrame columns
>>> df.info()
Info on DataFrame
>>> df.count()
Number of non-NaN values
```

Summary

```
>>> df.sum()
   (rows,columns)
   (rowsums)
   (columnsums)
   (min,max)
   (df.min(), df.max())
   (df.describe())
   (df.mean())
   (df.median())
```

Sum of values
Cumulative sum of values
Minimum/maximum values
Minimum/Maximum index value
Summary statistics
Mean of values
Median of values

Applying Functions

```
>>> f = lambda x: x**2
>>> df.apply(f)
Apply function
>>> df.applymap(f)
Apply function element-wise
```

Data Alignment

Internal Data Alignment

NaN values are introduced in the indices that don't overlap:

```
>>> a3 = pd.Series([7, -2, 3], index=['a', 'c', 'd'])
>>> a + a3
0    10.0
b    NaN
c    5.0
d    7.0
```

Arithmetic Operations with Fill Methods

You can also do the internal data alignment yourself with the help of the fill methods:

```
>>> a.add(s3, fill_value=0)
   10.0
   -9.0
   5.0
   7.0
>>> a.sub(s3, fill_value=2)
   8.0
   5.0
   3.0
   5.0
>>> a.div(s3, fill_value=4)
   2.5
   0.5
   1.25
   1.75
```

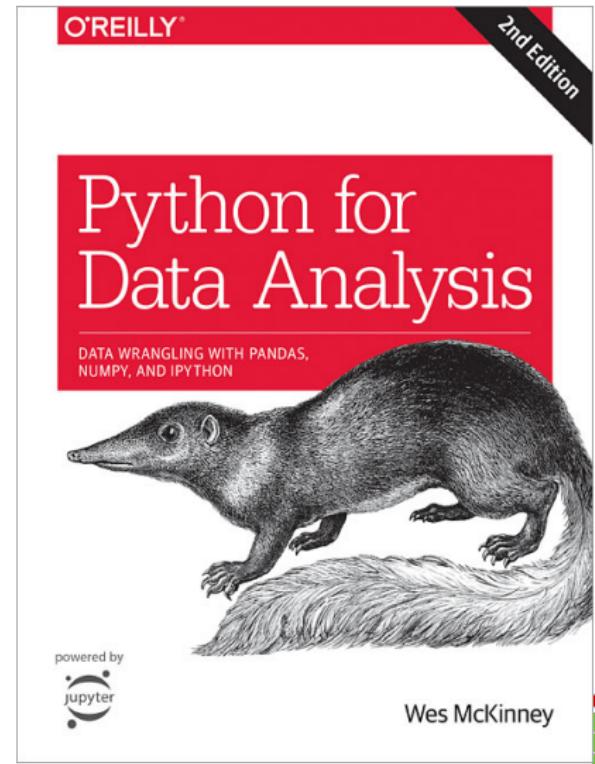


Referências



M Medium

stack overflow



Contato

HUMBERTO DA SILVA NETO

E-mail *humberto.nt4@gmail.com*

Telefone +55 27 99857-1584

