

Explorando Base de dados da Seguradora

Importando Bibliotecas ou Packages

In [1]:



```
import pandas as pd
import os
```

Atalhos

- Comando jupyter notebook inicializa o servidor.
- A cria uma célula acima.
- B cria uma célula abaixo.
- D apaga uma célula.
- M muda uma célula para markdown.
- Y muda uma célula para code.
- D,D apaga uma célula.
- C cópia uma célula.

Importando os dados csv

Definindo current_path

In [2]:

```
current_path = os.getcwd()
#print(current_path)
#dados/aluguel.csv
dados_seguradora = pd.read_csv(os.path.join(current_path, 'dados', 'aluguel.csv'), sep=';')
dados_seguradora
```

Out[2]:

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condominio	IPTU	Valor
0	Apartamento	Centro	1	0	0	15	800.0	390.0	20.0	53
1	Apartamento	Higienópolis	1	0	0	48	800.0	230.0	0.0	16
2	Apartamento	Cachambi	2	0	0	50	1300.0	301.0	17.0	26
3	Apartamento	Grajaú	2	1	0	70	1500.0	642.0	74.0	21
4	Apartamento	Lins de Vasconcelos	3	1	1	90	1500.0	455.0	14.0	16
...
19826	Quitinete	Glória	1	0	0	10	400.0	107.0	10.0	40
19827	Quitinete	Flamengo	1	0	0	23	900.0	605.0	0.0	39
19828	Quitinete	Centro	1	0	0	24	1100.0	323.0	0.0	45
19829	Quitinete	Copacabana	1	0	0	22	1500.0	286.0	200.0	68
19830	Quitinete	Centro	0	0	0	27	800.0	350.0	25.0	29

19831 rows × 11 columns



Importando json

In [3]:

```
file_dados_path_json = os.getcwd() + '/dados' + '/aluguel.json'
json = open(file_dados_path_json, 'r', encoding='utf-8')
print(json.read())
```

```
[{"Tipo": "Quitinete", "Bairro": "Copacabana", "Quartos": 1, "Vagas": 0, "Suites": 0, "Area": 40, "Valor": 1700.0, "Condominio": 500.0, "IPTU": 60.0}, {"Tipo": "Casa", "Bairro": "Jardim Bot\u00e2nico", "Quartos": 2, "Vagas": 0, "Suites": 1, "Area": 100, "Valor": 7000.0, "Condominio": null, "IPTU": null}, {"Tipo": "Conjunto Comercial/Sala", "Bairro": "Barra da Tijuca", "Quartos": 0, "Vagas": 4, "Suites": 0, "Area": 150, "Valor": 5200.0, "Condominio": 4020.0, "IPTU": 1111.0}, {"Tipo": "Apartamento", "Bairro": "Centro", "Quartos": 1, "Vagas": 0, "Suites": 0, "Area": 15, "Valor": 800.0, "Condominio": 390.0, "IPTU": 20.0}, {"Tipo": "Apartamento", "Bairro": "Higien\u00f3polis", "Quartos": 1, "Vagas": 0, "Suites": 0, "Area": 48, "Valor": 800.0, "Condominio": 230.0, "IPTU": null}, {"Tipo": "Apartamento", "Bairro": "Vista Alegre", "Quartos": 3, "Vagas": 1, "Suites": 0, "Area": 70, "Valor": 1200.0, "Condominio": null, "IPTU": null}, {"Tipo": "Apartamento", "Bairro": "Cachambi", "Quartos": 2, "Vagas": 0, "Suites": 0, "Area": 50, "Valor": 1300.0, "Condominio": 301.0, "IPTU": 17.0}, {"Tipo": "Casa de Condom\u00ednio", "Bairro": "Barra da Tijuca", "Quartos": 5, "Vagas": 4, "Suites": 5, "Area": 750, "Valor": 22000.0, "Condominio": null, "IPTU": null}, {"Tipo": "Casa de Condom\u00ednio", "Bairro": "Ramos", "Quartos": 2, "Vagas": 2, "Suites": 0, "Area": 65, "Valor": 1000.0, "Condominio": null, "IPTU": null}, {"Tipo": "Conjunto Comercial/Sala", "Bairro": "Centro", "Quartos": 0, "Vagas": 3, "Suites": 0, "Area": 695, "Valor": 35000.0, "Condominio": 19193.0, "IPTU": 3030.0}]
```

In [4]:

```
df_json = pd.read_json(file_dados_path_json)
df_json
```

Out[4]:

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condominio	IPTU
0	Quitinete	Copacabana	1	0	0	40	1700	500.0	60.0
1	Casa	Jardim Botânico	2	0	1	100	7000	NaN	NaN
2	Conjunto Comercial/Sala	Barra da Tijuca	0	4	0	150	5200	4020.0	1111.0
3	Apartamento	Centro	1	0	0	15	800	390.0	20.0
4	Apartamento	Higienópolis	1	0	0	48	800	230.0	NaN
5	Apartamento	Vista Alegre	3	1	0	70	1200	NaN	NaN
6	Apartamento	Cachambi	2	0	0	50	1300	301.0	17.0
7	Casa de Condomínio	Barra da Tijuca	5	4	5	750	22000	NaN	NaN
8	Casa de Condomínio	Ramos	2	2	0	65	1000	NaN	NaN
9	Conjunto Comercial/Sala	Centro	0	3	0	695	35000	19193.0	3030.0

Importando txt

In [5]:

```
file_dados_path_txt = os.getcwd() + '/dados' + '/aluguel.txt'
txt = open(file_dados_path_txt, 'r', encoding='utf-8')
print(txt.read())
```

Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condominio	IP
TU								
Quitinete	Copacabana	1	0	0	40	1700	50	
0	60							
Casa	Jardim Botânico	2	0	1	100	7000		
Conjunto Comercial/Sala	Barra da Tijuca	0	4	0	150	5200	4020.0	1111.0
00								
Apartamento	Centro	1	0	0	15	800	390.0	20.0
Apartamento	Higienópolis	1	0	0	48	800	230.0	NaN
0								
Apartamento	Vista Alegre	3	1	0	70	1200	NaN	NaN
Apartamento	Cachambi	2	0	0	50	1300	301.0	17.0
1	17							
Casa de Condomínio	Barra da Tijuca	5	4	5	750	22000	NaN	NaN
000								
Casa de Condomínio	Ramos	2	2	0	65	1000	NaN	NaN
Conjunto Comercial/Sala	Centro	0	3	0	695	35000	19193.0	3030.0
193	3030							

In [6]:

```
df_txt = pd.read_table(file_dados_path_txt)
df_txt
```

Out[6]:

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condominio	IPTU
0	Quitinete	Copacabana	1	0	0	40	1700	500.0	60.0
1	Casa	Jardim Botânico	2	0	1	100	7000	NaN	NaN
2	Conjunto Comercial/Sala	Barra da Tijuca	0	4	0	150	5200	4020.0	1111.0
3	Apartamento	Centro	1	0	0	15	800	390.0	20.0
4	Apartamento	Higienópolis	1	0	0	48	800	230.0	NaN
5	Apartamento	Vista Alegre	3	1	0	70	1200	NaN	NaN
6	Apartamento	Cachambi	2	0	0	50	1300	301.0	17.0
7	Casa de Condomínio	Barra da Tijuca	5	4	5	750	22000	NaN	NaN
8	Casa de Condomínio	Ramos	2	2	0	65	1000	NaN	NaN
9	Conjunto Comercial/Sala	Centro	0	3	0	695	35000	19193.0	3030.0

Importando Excel

In [7]:

```
file_dados_path_xlsx = os.getcwd() + '/dados' + '/aluguel.xlsx'
df_xlsx = pd.read_excel(file_dados_path_xlsx)
df_xlsx
```

Out[7]:

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condominio	IPTU
0	Quitinete	Copacabana	1	0	0	40	1700	500.0	60.0
1	Casa	Jardim Botânico	2	0	1	100	7000	NaN	NaN
2	Conjunto Comercial/Sala	Barra da Tijuca	0	4	0	150	5200	4020.0	1111.0
3	Apartamento	Centro	1	0	0	15	800	390.0	20.0
4	Apartamento	Higienópolis	1	0	0	48	800	230.0	NaN
5	Apartamento	Vista Alegre	3	1	0	70	1200	NaN	NaN
6	Apartamento	Cachambi	2	0	0	50	1300	301.0	17.0
7	Casa de Condomínio	Barra da Tijuca	5	4	5	750	22000	NaN	NaN
8	Casa de Condomínio	Ramos	2	2	0	65	1000	NaN	NaN
9	Conjunto Comercial/Sala	Centro	0	3	0	695	35000	19193.0	3030.0

Importando HTML

In [8]:

```
file_dados_path_html = os.getcwd() + '/dados' + '/dados_html_1.html'
df_html = pd.read_html(file_dados_path_html)
df_html[0]
```

Out[8]:

	FAIXA ETÁRIA	UNAFISCO SAÚDE	SOFT II	SOFT PARTICIPATIVO	UNIQUE
0	00 A 18		R\$218,99	R\$192,60	R\$334,48
1	19 A 23		R\$263,63	R\$207,67	R\$380,06
2	24 A 28		R\$322,44	R\$224,26	R\$464,39
3	29 A 33		R\$402,05	R\$258,63	R\$566,68
4	34 A 38		R\$455,17	R\$306,37	R\$664,61
5	39 A 43		R\$489,29	R\$417,93	R\$742,72
6	44 A 48		R\$582,65	R\$495,24	R\$869,47
7	49 A 53		R\$718,86	R\$562,64	R\$1.046,29
8	54 A 58		R\$908,54	R\$721,00	R\$1.355,74
9	59 OU MAIS		R\$1.244,50	R\$1.151,62	R\$2.003,53

In [9]:

```
file_dados_path_html_link = 'https://unafiscosaude.org.br/site/comparativo-de-planos/'
df_html_link = pd.read_html(file_dados_path_html_link)
df_html_link[0]
```

Out[9]:

	NOME COMERCIAL PLANO >>	SOFT II	SOFT PARTICIPATIVO	UNIQUE	PREMIUM II	
0	Registro ANS	Nº. 471839149	Nº. 478132175	Nº. 478131177	Nº. 468235121	Nº. 48
1	Situação do Plano ANS	Ativo	Ativo	Ativo	Ativo	
2	Tipo de Contratação	Coletivo por adesão	Coletivo por adesão	Coletivo por adesão	Coletivo por adesão	Col
3	Segmentação Assistencial	"Ambulatorial + Hospitalar com Obstetrícia	"Ambulatorial + Hospitalar com Obstetrícia	"Ambulatorial + Hospitalar com Obstetrícia + O...	"Ambulatorial + Hospitalar com Obstetrícia + O...	"Ambu Hospit Obs
4	Odontologia	Não há	Não há	Sim (Rol ANS)	Sim (Rol ANS) + extra ROL	Sim (f + O
5	Formação do preço:	Preestabelecido	Preestabelecido	Preestabelecido	Preestabelecido	Preesta
6	Vínculo do Beneficiário:	Sem Vínculo Empregatício	Sem Vínculo Empregatício	Sem Vínculo Empregatício	Sem Vínculo Empregatício	Sen Emp
7	Padrão de acomodação	Apartamento (Acomodação Individual)	Apartamento (Acomodação Individual)	Apartamento (Acomodação Individual)	Apartamento (Acomodação Individual)	Apa (Acoi Ir
8	Coparticipação	Não há	30% em consultas, exames, terapias e atendimen...	30% em consultas, exames, terapias e atendimen...	Não há	
9	Cobertura de procedimentos	Limitada ao Rol ANS	Limitada ao Rol ANS	Limitada ao Rol ANS	Rol + cerca de 380 procedimentos extra cobertu...	Limitac
10	Serviços e coberturas adicionais:	Não há	Não há	Não há	Assistência/internação domiciliar (Home Care...	- Acess odonto forma
11	Livre Escolha:	Não há	Não há	Reembolso conforme valores – Referencial de Ho...	Reembolso conforme valores – Referencial de Ho...	Re (, Refer
12	Rede de atendimento:	básica conforme opções disponibilizadas por in...	básica conforme opções disponibilizadas por in...	básica + rede adicional conforme opções dispon...	básica + rede adicional (direta e indireta) + ...	básic (indireta

In [10]:

```
df_html_link[1]
```

Out[10]:

	FAIXA ETÁRIA	SOFT II	UNIQUE	SOFT PARTICIPATIVO	PREMIUM II	PLATINUM
0	00 a 18	R\$ 251,41	R\$ 361,42	R\$ 208,11	R\$ 549,96	R\$ 435,62
1	19 a 23	R\$ 302,68	R\$ 410,67	R\$ 224,39	R\$ 624,93	R\$ 494,98
2	24 a 28	R\$ 370,20	R\$ 501,79	R\$ 242,32	R\$ 763,58	R\$ 604,81
3	29 a 33	R\$ 461,60	R\$ 612,32	R\$ 279,46	R\$ 931,75	R\$ 738,03
4	34 a 38	R\$ 522,59	R\$ 718,14	R\$ 331,05	R\$ 1.024,93	R\$ 865,57
5	39 a 43	R\$ 561,75	R\$ 802,54	R\$ 451,59	R\$ 1.145,38	R\$ 967,29
6	44 a 48	R\$ 668,94	R\$ 939,50	R\$ 535,13	R\$ 1.340,82	R\$ 1.132,37
7	49 a 53	R\$ 825,34	R\$ 1.130,56	R\$ 607,96	R\$ 1.613,50	R\$ 1.362,65
8	54 a 58	R\$ 1.043,10	R\$ 1.464,94	R\$ 779,07	R\$ 2.090,74	R\$ 1.765,67
9	59 ou mais	R\$ 1.428,83	R\$ 2.164,90	R\$ 1.244,38	R\$ 3.261,12	R\$ 2.609,33

In [11]:

```
file_dados_path_html_link_fr = 'https://www.federalreserve.gov/releases/h3/current/default.'
df_html_fr = pd.read_html(file_dados_path_html_link_fr)
len(df_html_fr)
```

Out[11]:

3

In [12]:



df_html_fr[0]

Out[12]:

	Date	Reserve balances required				Reserve balances maintained 4	
	Date	Reserve balance requirements 1	Top of penalty-free band 2	Bottom of penalty-free band 3	Total	Balances maintained to satisfy reserve balance requirements 5	Balances maintained that exceed the top of the penalty-free band 6
0	Month 7	Month 7	Month 7	Month 7	Month 7	Month 7	Month 7
1	Aug. 2019	134639	148114	121167	1520876	147555	1373321
2	Sept. 2019	134707	148188	121227	1439771	147498	1292273
3	Oct. 2019	134998	148509	121489	1481513	148050	1333462
4	Nov. 2019	140705	154787	126625	1529341	154256	1375085
5	Dec. 2019	138984	152893	125076	1630090	152562	1477528
6	Jan. 2020	144816	159309	130326	1645384	158765	1486619
7	Feb. 2020	138370	152218	124524	1656989	151612	1505377
8	Mar. 2020	113139	124461	101817	2045148	124147	1921000
9	Apr. 2020	0	0	0	2953586	0	2953586
10	May 2020	0	0	0	3217635	0	3217635
11	June 2020	0	0	0	3043556	0	3043556
12	July 2020	0	0	0	2718529	0	2718529
13	Aug. 2020	0	0	0	2799719	0	2799719
14	NaN	NaN	NaN	NaN	NaN	NaN	NaN
15	Two weeks ending 8	Two weeks ending 8	Two weeks ending 8	Two weeks ending 8	Two weeks ending 8	Two weeks ending 8	Two weeks ending 8
16	July 15, 2020	0	0	0	2795785	0	2795785
17	July 29, 2020	0	0	0	2623526	0	2623526
18	Aug. 12, 2020	0	0	0	2751124	0	2751124

Date	Reserve balances required				Reserve balances maintained 4	
	Reserve balance requirements 1	Top of penalty-free band 2	Bottom of penalty-free band 3	Total	Balances maintained to satisfy reserve balance requirements 5	Balances maintained that exceed the top of the penalty-free band 6
19 Aug. 26, 2020	0	0	0	2821740	0	2821740
20 Sept. 9, 2020	0	0	0	2854690	0	2854690

In [13]:



df_html_fr[1]

Out[13]:

	Date	Reserves			Vault cash 2			Monetary base	
	Date	Total 1	Required	Total	Used to satisfy required reserves 3	Surplus 4	Total 5	Total balances maintained 6	Currency in circulation
0	Month 7	Month 7	Month 7	Month 7	Month 7	Month 7	Month 7	Month 7	Month 7
1	Aug. 2019	1586844	200608	78451	65969	12483	3271378	1520876	1750502
2	Sept. 2019	1504805	199741	76550	65035	11516	3202668	1439771	1762898
3	Oct. 2019	1547136	200621	76660	65623	11037	3252807	1481513	1771294
4	Nov. 2019	1595222	206586	76921	65881	11040	3315551	1529341	1786209
5	Dec. 2019	1698344	207239	80168	68255	11914	3426464	1630090	1796375
6	Jan. 2020	1715219	214652	81216	69835	11381	3442569	1645384	1797185
7	Feb. 2020	1726943	208324	80735	69954	10781	3454474	1656989	1797485
8	Mar. 2020	2099369	167360	77114	54222	22892	3883155	2045148	1838007
9	Apr. 2020	2953586	0	88104	0	88104	4844940	2953586	1891355
10	May 2020	3217635	0	101345	0	101345	5149527	3217635	1931891
11	June 2020	3043556	0	104201	0	104201	5001978	3043556	1958422
12	July 2020	2718529	0	98392	0	98392	4700365	2718529	1981836
13	Aug. 2020	2799719	0	95327	0	95327	4807394	2799719	2007675
14	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
15	Two weeks ending 8	Two weeks ending 8	Two weeks ending 8	Two weeks ending 8	Two weeks ending 8	Two weeks ending 8	Two weeks ending 8	Two weeks ending 8	Two weeks ending 8
16	July 15, 2020	2795785	0	99459	0	99459	4772522	2795785	1976737
17	July 29, 2020	2623526	0	97225	0	97225	4609143	2623526	1985617
18	Aug. 12, 2020	2751124	0	95638	0	95638	4750981	2751124	1999857

Date		Reserves		Vault cash 2			Monetary base		
	Date	Total 1	Required	Total	Used to satisfy required reserves 3	Surplus 4	Total 5	Total balances maintained 6	Currency in circulation
19	Aug. 26, 2020	2821740	0	95066	0	95066	4830999	2821740	2009260
20	Sept. 9, 2020	2854690	0	95313	0	95313	4876690	2854690	2022000

In [14]:



df_html_fr[2]

Out[14]:

	Date	Borrowings from the Federal Reserve					Nonborrowed reserves 2
	Date	Total	Primary	Secondary	Seasonal	Other credit extensions 1	Nonborrowed reserves 2
0	Month 3	Month 3	Month 3	Month 3	Month 3	Month 3	Month 3
1	Aug. 2019	117	16	0	101	0	1586727
2	Sept. 2019	101	15	0	86	0	1504704
3	Oct. 2019	63	4	0	59	0	1547073
4	Nov. 2019	26	4	0	22	0	1595196
5	Dec. 2019	21	10	0	11	0	1698323
6	Jan. 2020	6	5	0	1	0	1715213
7	Feb. 2020	3	3	0	0	0	1726939
8	Mar. 2020	40554	19538	0	0	21015	2058816
9	Apr. 2020	124461	38299	0	0	86162	2829125
10	May 2020	111845	22180	0	2	89663	3105791
11	June 2020	97671	8456	0	11	89204	2945885
12	July 2020	92943	4670	0	26	88248	2625585
13	Aug. 2020	83126	2851	0	38	80237	2716593
14	NaN	NaN	NaN	NaN	NaN	NaN	NaN
15	Two weeks ending 4	Two weeks ending 4	Two weeks ending 4	Two weeks ending 4	Two weeks ending 4	Two weeks ending 4	Two weeks ending 4
16	July 15, 2020	94326	5055	0	21	89251	2701459
17	July 29, 2020	92428	4376	0	31	88021	2531099
18	Aug. 12, 2020	85636	3111	0	37	82488	2665489
19	Aug. 26, 2020	82030	2651	0	41	79339	2739710
20	Sept. 9, 2020	80170	2786	0	34	77350	2774520

In [15]:



```
df_html_fr[2].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 7 columns):
 #   Column                                     Non-Null Count  Dtype  Dtype2
---  --
0   (Date, Date)                             20 non-null    object
1   (Borrowings from the Federal Reserve, Total)  20 non-null    object
2   (Borrowings from the Federal Reserve, Primary)  20 non-null    object
3   (Borrowings from the Federal Reserve, Secondary)  20 non-null    object
4   (Borrowings from the Federal Reserve, Seasonal)  20 non-null    object
5   (Borrowings from the Federal Reserve, Other credit extensions 1)  20 non-null    object
6   (Nonborrowed reserves 2, Nonborrowed reserves 2)  20 non-null    object
dtypes: object(7)
memory usage: 1.3+ KB
```

Relatório de Análise I

In [16]:



```
dados_seguradora.head()
```

Out[16]:

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condominio	IPU	Valor m2	
0	Apartamento	Centro	1	0	0	15	800.0	390.0	20.0	53.33	/
1	Apartamento	Higienópolis	1	0	0	48	800.0	230.0	0.0	16.67	/
2	Apartamento	Cachambi	2	0	0	50	1300.0	301.0	17.0	26.00	/
3	Apartamento	Grajaú	2	1	0	70	1500.0	642.0	74.0	21.43	/
4	Apartamento	Lins de Vasconcelos	3	1	1	90	1500.0	455.0	14.0	16.67	/

Informações Gerais sobre a Base de Dados

In [17]:

```
dados_seguradora.dtypes
```

Out[17]:

```
Tipo          object
Bairro        object
Quartos      int64
Vagas        int64
Suites        int64
Area          int64
Valor         float64
Condominio    float64
IPTU          float64
Valor m2      float64
Tipo Agregado object
dtype: object
```

Tranformando em DataFrame

In [18]:

```
pd.DataFrame(dados_seguradora.dtypes)
```

Out[18]:

	0
Tipo	object
Bairro	object
Quartos	int64
Vagas	int64
Suites	int64
Area	int64
Valor	float64
Condominio	float64
IPTU	float64
Valor m2	float64
Tipo Agregado	object

In [19]:



```
tipos_de_dados = pd.DataFrame(dados_seguradora.dtypes, columns=['Tipos de Dados'])
tipos_de_dados
```

Out[19]:

Tipos de Dados	
Tipo	object
Bairro	object
Quartos	int64
Vagas	int64
Suites	int64
Area	int64
Valor	float64
Condominio	float64
IPTU	float64
Valor m2	float64
Tipo Agregado	object

In [20]:



```
tipos_de_dados.columns.name = 'Variáveis'
tipos_de_dados
```

Out[20]:

Variáveis	Tipos de Dados
Tipo	object
Bairro	object
Quartos	int64
Vagas	int64
Suites	int64
Area	int64
Valor	float64
Condominio	float64
IPTU	float64
Valor m2	float64
Tipo Agregado	object

In [21]:



```
dados_seguradora.shape
```

Out[21]:

```
(19831, 11)
```

In [22]:



```
imoveis = dados_seguradora.shape[0]  
variaveis = dados_seguradora.shape[1]
```

In [23]:



```
print("imóveis: {}\nvariáveis: {}".format(imoveis,variaveis))
```

```
imóveis: 19831  
variáveis: 11
```

In [24]:



```
print('A base de dados apresenta {} registros de imóveis, e {} variáveis'.format(imoveis,va
```

```
A base de dados apresenta 19831 registros de imóveis, e 11 variáveis
```