Relatório de Análise VIII

Identificando e Removendo Outliers

In [26]:
```
# importando Pandas
# importando Matplot lib
# configurando o tamanho da representação visual via figsize()
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
plt.rc('figure', figsize = (14, 6))
```

In [27]:
```
# importando a base de dados
dados = pd.read_csv('data/aluguel_residencial.csv', sep = ';')
```

In [28]:
```
# exibindo os primeiros dez itens
dados.head(10)
```

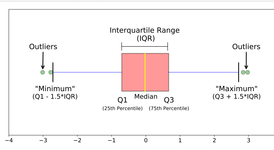
Out[28]:

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condomínio	IPTU	Valor m2	Tipo Agregado
0	Quitinete	Copacabana	1	0	0	40	1700.0	500.0	60.0	42.50	Apartamento
1	Casa	Jardim Botânico	2	0	1	100	7000.0	0.0	0.0	70.00	Casa
2	Apartamento	Centro	1	0	0	15	800.0	390.0	20.0	53.33	Apartamento
3	Apartamento	Higienópolis	1	0	0	48	800.0	230.0	0.0	16.67	Apartamento
4	Apartamento	Cachambi	2	0	0	50	1300.0	301.0	17.0	26.00	Apartamento
5	Casa de Condomínio	Barra da Tijuca	5	4	5	750	22000.0	0.0	0.0	29.33	Casa
6	Casa de Condomínio	Ramos	2	2	0	65	1000.0	0.0	0.0	15.38	Casa
7	Apartamento	Grajaú	2	1	0	70	1500.0	642.0	74.0	21.43	Apartamento
8	Apartamento	Lins de Vasconcelos	3	1	1	90	1500.0	455.0	14.0	16.67	Apartamento
9	Apartamento	Copacabana	1	0	1	40	2000.0	561.0	50.0	50.00	Apartamento

Obs.: *Teremos uma representação gráfica que vai ajudar a compreender a técnica para remoção de outliers. Utilizando Box-plot, que possui a seguinte configuração: temos uma mediana, em que dividimos os dados em 50%, para a direita e esquerda, igualmente. Teremos o Q1, que se refere ao primeiro quartil e parte em 25% e 75%, já o Q3 parte os dados em 75% e 25%. A diferente entre Q3 e Q1 gera o intervalo interquartílico, isto é, as estatísticas que geraremos para realizar o corte de outliers.*

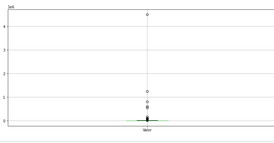
In [29]:
```
# importando a representação grafica do box-plot
from IPython.display import Image
Image(filename='data/Boxplot.png')
```

Out[29]:



In [30]:
```
# gerando nosso box-plot
dados.boxplot(['Valor'])
```

Out[30]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x17a2fd51340>
```



In [31]:
```
# criando uma visualização clara dos dados ao realizar uma seleção em nosso dataframe
dados[dados['Valor'] >= 500000]
```

Out[31]:

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condomínio	IPTU	Valor m2	Tipo Agregado
7629	Apartamento	Barra da Tijuca	1	1	0	65	600000.0	980.0	120.0	9230.77	Apartamento
10636	Casa de Condomínio	Freguesia (Jacarepaguá)	4	2	3	163	600000.0	900.0	0.0	4907.98	Casa
12661	Apartamento	Freguesia (Jacarepaguá)	2	2	1	150	550000.0	850.0	150.0	3666.67	Apartamento
13846	Apartamento	Recreio dos Bandeirantes	3	2	1	167	1250000.0	1186.0	320.0	7485.03	Apartamento
15520	Apartamento	Botafogo	4	1	1	300	4500000.0	1100.0	0.0	15000.00	Apartamento

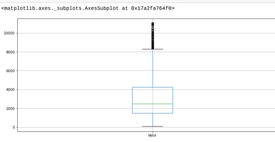
In [32]:
```
# para facilitar a digitação, criamos uma series que chamaremos de 'Valor'
valor = dados['Valor']
```

In [33]:
```
# calculando o Q1, primeiro quartil
# calculando o Q2, segundo quartil
# calculando o IQR, intervalo interquartílico
# calculando os limites superior e inferior
Q1 = valor.quantile(.25)
Q3 = valor.quantile(.75)
IQR = Q3 - Q1
limite_inferior = Q1 - 1.5 * IQR
limite_superior = Q3 + 1.5 * IQR
```

In [34]:
```
# criando uma seleção dos dados que estão apenas dentro desses dois limites
selecao = (valor >= limite_inferior) & (valor <= limite_superior)
dados_new = dados[selecao]
```

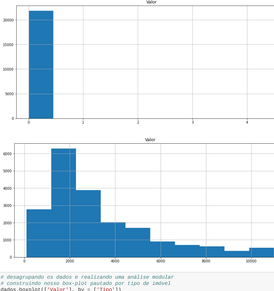
In [35]:
```
# gerando novamente o box-plot
dados_new.boxplot(['Valor'])
```

Out[35]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x17a2a764f60>
```



In [36]:
```
# gerando o histograma, uma distribuição de frequências dos dados
# podemos comparar as frequências dos DataFrame 'dados' e 'dados_new'
dados.hist(['Valor'])
dados_new.hist(['Valor'])
```

Out[36]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001?A2FA53280>]],
      dtype=object)
```



In [37]:
```
# desagrupando os dados e realizando uma análise modular
# construindo nosso box-plot pastado por tipo de imóvel
dados.boxplot(['Valor'], by = ['Tipo'])
```

Out[37]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x17a3040b070>
```

In [38]:
```
# criando 'grupo_tipo' com base na variável 'Valor' por Tipo
grupo_tipo = dados.groupby('Tipo')['Valor']
```

In [39]:
```
# verificando o tipo da variável
type(grupo_tipo)
```

Out[39]:
```
pandas.core.groupby.generic.SeriesGroupBy
```

In [40]:
```
# propriedade groups cria um dicionário,
# cuja a chave será o Tipo, e os índices de localização.
grupo_tipo.groups
```

Out[40]:
```
{'Apartamento': Int64Index([    2,     3,     4,     7,     8,     9,    11,    13,    14,
                15,
                ...
                21813, 21814, 21816, 21817, 21818, 21819, 21821, 21823, 21824,
                21825],
               dtype='int64', length=18780),
 'Casa': Int64Index([    1,    22,    54,    57,    96,   100,   144,   160,   180,
                230,
                ...
                21582, 21606, 21614, 21667, 21672, 21699, 21756, 21781, 21793,
                21804],
               dtype='int64', length=965),
 'Casa de Condomínio': Int64Index([    5,     6,    12,    16,    42,    58,   166,   168,
    183,
                207,
                ...
                21706, 21711, 21715, 21752, 21763, 21764, 21782, 21791, 21803,
                21820],
               dtype='int64', length=996),
 'Casa de Vila': Int64Index([   83,   212,   228,   303,   332,   697,   822,   844,   918,
                1012,
                ...
                21184, 21189, 21253, 21325, 21353, 21366, 21588, 21635, 21716,
                21762],
               dtype='int64', length=249),
 'Quitinete': Int64Index([    0,    10,    28,    71,    78,    86,   101,   120,   146,
                174,
                ...
                21304, 21410, 21441, 21656, 21682, 21687, 21728, 21748, 21815,
                21822],
               dtype='int64', length=836)}
```

In [41]:
```
# calculando o Q1, primeiro quartil
# calculando o Q2, segundo quartil
# calculando o IQR, intervalo interquartílico
# calculando os limites superior e inferior
Q1 = grupo_tipo.quantile(.25)
Q3 = grupo_tipo.quantile(.75)
IQR = Q3 - Q1
limite_inferior = Q1 - 1.5 * IQR
limite_superior = Q3 + 1.5 * IQR
```

In [42]:
```
# visualizando Q1
# temos uma Series por tipo de imóvel
Q1
```

Out[42]:
```
Tipo
Apartamento           1700.0
Casa                  1100.0
Casa de Condomínio    4000.0
Casa de Vila           750.0
Quitinete              900.0
Name: Valor, dtype: float64
```

In [43]:
```
# visualizando Q3
# temos uma Series por tipo de imóvel
Q3
```

Out[43]:
```
Tipo
Apartamento           5000.0
Casa                  9000.0
Casa de Condomínio    15250.0
Casa de Vila           1800.0
Quitinete              2450.0
Name: Valor, dtype: float64
```

In [44]:
```
# visualizando o 'limite_inferior'
# temos uma Series por tipo de imóvel
limite_inferior
```

Out[44]:
```
Tipo
Apartamento          -3250.0
Casa                 -11950.0
Casa de Condomínio   -12875.0
Casa de Vila           -825.0
Quitinete             -0.0
Name: Valor, dtype: float64
```

In [45]:
```
# visualizando o 'limite_superior'
# temos uma Series por tipo de imóvel
limite_superior
```

Out[45]:
```
Tipo
Apartamento           9950.0
Casa                  22850.0
Casa de Condomínio    32125.0
Casa de Vila           3375.0
Quitinete              3400.0
Name: Valor, dtype: float64
```

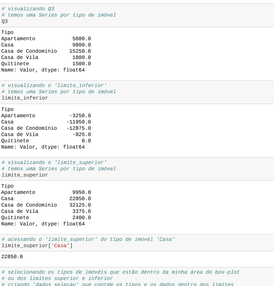
In [46]:
```
# acessando o 'limite_superior' do tipo de imovél 'Casa'
limite_superior['Casa']
```

Out[46]:
```
22850.0
```

In [47]:
```
# selecionando os tipos de imoveis que estão dentro da minha área do box-plot
# ou dos limites superior e inferior
# criando 'dados_selecao' que contem os tipos e os dados dentro dos limites
# criando um novo DataFrame que faz a concatenação de todos tipos de imoveis
dados_new = pd.DataFrame()
for tipo in grupo_tipo.groups.keys():
    eh_tipo = dados['Tipo'] == tipo
    eh_dentro_limite = (dados['Valor'] >= limite_inferior[tipo]) & (dados['Valor'] <= limite
_superior[tipo])
    selecao = eh_tipo & eh_dentro_limite
    dados_selecao = dados[selecao]
    dados_new = pd.concat([dados_new, dados_selecao])
```

In [48]:
```
# visualizando nosso box-plot após as limpezas dos dados
dados_new.boxplot(['Valor'], by = ['Tipo'])
```

Out[48]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x17a3040bf10>
```



In [49]:
```
# exportando nossos dados sem outliers
dados_new.to_csv('data/aluguel_residencial_sem_outliers.csv', sep = ';', index = False)
```