In [1]:

```
%load_ext watermark
%watermark
```

```
2019-05-17T17:18:04+02:00

CPython 3.6.5
IPython 6.4.0

compiler   : GCC 7.2.0
system     : Linux
release    : 5.0.13-arch1-1-ARCH
machine    : x86_64
processor  :
CPU cores  : 4
interpreter: 64bit
```

VISUALIZACIÓN DE DATOS AVANZADA

En este capitulo se trataran los métodos para poder personalizar las gráficas de Matplotlib y como cargar Estilos ya predefinidos. Además se realizaran ejemplos con otras librerías interesantes para la visualización de datos:

- IPyWidgets.
- Cartopy.
- Seaborn.
- BokehJS

Carga de Datos y Preparacion de DataSet

Como en el apartado de visualización básica de datos utilizaremos el Boston Housing Dataset. Recopilado en 1976 y publicado en Berkeley (https://www.law.berkeley.edu/files/Hedonic.PDF)

In [2]:

```
import pandas as pd
df = pd.read_csv("boston_dataset.csv")

#renombramos las variables
df = df.rename(columns={
    "TOWN":"CIUDAD",
    "CRIM":"INDICE_CRIMEN",
    "ZN":"PCT_ZONA_RESIDENCIAL",
    "INDUS":"PCT_ZONA_INDUSTRIAL",
    "CHAS":"RIO_CHARLES",
    "NOX":"OXIDO_NITROSO_PPM",
    "RM":"N_HABITACIONES_MEDIO",
    "AGE":"PCT_CASAS_40S",
    "DIS_EMPLEO":"DISTANCIA_CENTRO_EMPLEO",
    "RAD":"DIS_AUTOPISTAS",
    "TAX":"CARGA_FISCAL",
    "PTRATIO":"RATIO_PROFESORES",
    "B":"PCT_NEGRA",
    "MEDV":"VALOR_MEDIANO",
    "LSTAT":"PCT_CLASE_BAJA"
})

df.head()
```

Out[2]:

	CIUDAD	LON	LAT	VALOR_MEDIANO	INDICE_CRIMEN	PCT_ZONA_RESIDENCIAL	PCT_ZONA_INDUSTRIAL
0	Nahant	-70.955	42.2550	24.0	0.00632	18.0	2.31
1	Swampscott	-70.950	42.2875	21.6	0.02731	0.0	7.07
2	Swampscott	-70.936	42.2830	34.7	0.02729	0.0	7.07
3	Marblehead	-70.928	42.2930	33.4	0.03237	0.0	2.18
4	Marblehead	-70.922	42.2980	36.2	0.06905	0.0	2.18

Personalización de Gráficos

Mediante los metodos de la librería **PyPlot** de **MatplotLib** veremos como especifciar titulos para los gráficos y como personalizar la forma de los punteros en gráficos, su tamaño y su color.

```
In [3]:
```

```
%matplotlib notebook
```

```
In [4]:
```

```
import matplotlib.pyplot as plt
```

```
In [5]:
```

```
#Cambiamos el punto con marker, color y el tamanio tan solo llamando a los parametros
df.plot.scatter(x="N_HABITACIONES_MEDIO", y="VALOR_MEDIANO", marker="*", color="pink", figsize=(8,8))

plt.title("Relacion entre el numero de habitaciones y el valor de las viviendas")

plt.xlabel("Numero medio de habitaciones")

plt.ylabel("Valor mediano de las viviendas ($1000s)")
```

```
Out[5]:

Text(0,0.5,'Valor mediano de las viviendas ($1000s)')
```

Damos un tamanio para las figuras por defecto en las librerias

In [6]:

```
import matplotlib as mpl

mpl.rcParams['figure.figsize'] = (8,8)
```

In [7]:

```
df.plot.scatter(x="N_HABITACIONES_MEDIO", y="VALOR_MEDIANO", marker="*", color="pink")

plt.title("Relacion entre el numero de habitaciones y el valor de las viviendas")

plt.xlabel("Numero medio de habitaciones")

plt.ylabel("Valor mediano de las viviendas ($1000s)")
```

Out[7]:

```
Text(0,0.5,'Valor mediano de las viviendas ($1000s)')
```

Establecer estilos en Matplotlib

Por defecto Matplotlib tiene un estilo definido, un aspecto muy característico y facilmente reconocible. Pero también permite personalizar los estilos de gráficas de una forma muy sencilla, utilizando hojas de estilos predefinidas y que vienen incluidas con Matplotlib

```
In [8]:
```

```
#mostramos la lista disponible de estilos en pyplot.
plt.style.available
```

```
Out[8]:
```

```
['fivethirtyeight',
 'bmh',
 'seaborn-talk',
 'seaborn-notebook',
 'seaborn',
 'seaborn-whitegrid',
 'ggplot',
 'tableau-colorblind10',
 'dark_background',
 'seaborn-deep',
 '_classic_test',
 'seaborn-colorblind',
 'seaborn-darkgrid',
 'seaborn-poster',
 'grayscale',
 'seaborn-dark',
 'seaborn-paper',
 'fast',
 'seaborn-pastel',
 'seaborn-white',
 'seaborn-ticks',
 'classic',
 'seaborn-bright',
 'Solarize_Light2',
 'seaborn-muted',
 'seaborn-dark-palette']
```

Podemos encontrar mas estilos aqui ("https://matplotlib.org/gallery/style_sheets/style_sheets_reference.html")

```
In [9]:
```

```
plt.style.use("fivethirtyeight")
```

```
In [10]:
```

```
df.plot.scatter(x="N_HABITACIONES_MEDIO", y="VALOR_MEDIANO")

plt.title("Relacion entre el numero de habitaciones y el valor de las viviendas")

plt.xlabel("Numero medio de habitaciones")

plt.ylabel("Valor mediano de las viviendas ($1000s)")
```

Out[10]:

```
Text(0,0.5,'Valor mediano de las viviendas ($1000s)')
```

IPyWidgets

IpyWidgets (https://ipywidgets.readthedocs.io/en/stable/) es una librería que nos permite importar widgets FrontEnd para poder interactuar con las gráficos. Podemos invocarlo con interact.

In [11]:

```
from ipywidgets import interact
```

In [12]:

```
#creamos la funcion grafico varible para comparar la columna 1 seleccionable desde el ComboBox con el Valor Media
no
@interact(col1=df.columns.tolist())
def grafico_variable(col1):
    df.plot.scatter(x=col1, y="VALOR_MEDIANO")
    plt.title("{} vs VALOR_MEDIANO".format(col1))
```

In [13]:

```
#Indicamos a matplotlib que estamos trabajando con notebook para que se reescale mejor.
%matplotlib notebook
```

In [14]:

```
df.plot.scatter(x="LON", y="LAT")
```

Out[14]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1a5ee6ea90>
```

Cartopy

Cartopy (https://scitools.org.uk/cartopy/docs/latest/) es una librería diseñada para procesar datos geoespaciales en orden para "plottear" mapas y poder realizar analisis de datos.

In [49]:

```
import cartopy.crs as ccrs

from cartopy.io import img_tiles
```

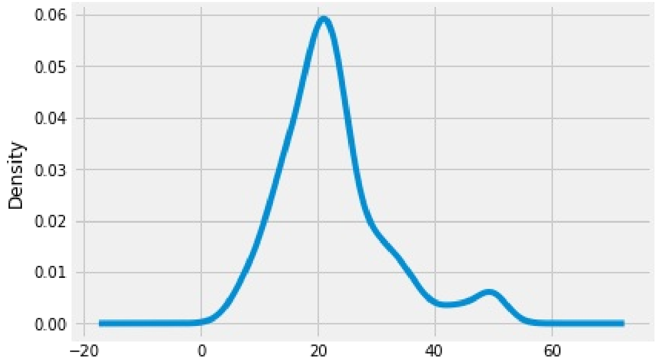
```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-49-fca1b84c8aa4> in <module>()
----> 1 import cartopy.crs as ccrs
      2 
      3 from cartopy.io import img_tiles

ModuleNotFoundError: No module named 'cartopy'
```

In [45]:

```
df.VALOR_MEDIANO.plot.kde()
```

Out[45]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1a4cea3c88>
```



In [46]:

```
primer_quintil = df.VALOR_MEDIANO.quantile(0.2)
primer_quintil
```

Out[46]:

```
15.3
```

In [47]:

```
cuarto_quintil = df.VALOR_MEDIANO.quantile(0.8)
cuarto_quintil
```

Out[47]:

```
28.2
```

In [48]:

```
imagery = img_tiles.GoogleTiles()


ax = plt.axes(projection=imagery.crs)

limites_mapa = (-71.38 ,-70.77,42.03 , 42.47)

ax.set_extent(limites_mapa)

ax.add_image(imagery, 10)

df_primer_qt = df[df.VALOR_MEDIANO<primer_quintil]

df_tercer_qt = df[df.VALOR_MEDIANO>cuarto_quintil]


plt.plot(df_primer_qt.LON, df_primer_qt.LAT, transform=ccrs.Geodetic(), marker=".",
         markersize=10, color="red", linewidth=0, alpha=0.5)

plt.plot(df_tercer_qt.LON, df_primer_qt.LAT, transform=ccrs.Geodetic(), marker=".",
         markersize=10, color="green", linewidth=0, alpha=0.5)

plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-48-e458cdb2207d> in <module>()
----> 1 imagery = img_tiles.GoogleTiles()
      2 
      3 
      4 ax = plt.axes(projection=imagery.crs)
      5 

NameError: name 'img_tiles' is not defined
```

Seaborn

Basada en matplotlib, se usa para hacer más atractivos los gráficos e información estadística en Python. Su objetivo es darle una mayor relevancia a las visualizaciones, dentro de las tareas de exploración e interpretación de los datos.

```
In [24]:
```

```
import seaborn as sns
```

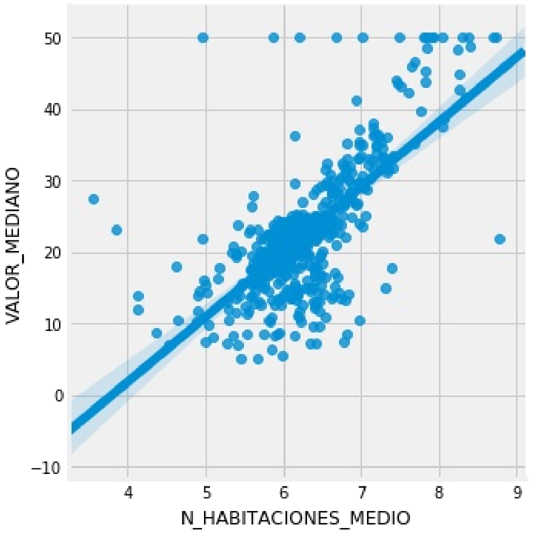
```
In [25]:
```

```
#especificamos a matplotlib para incluir los gráficos en el notebook
%matplotlib inline
```

```
In [26]:
```

```
sns.lmplot(x="N_HABITACIONES_MEDIO", y="VALOR_MEDIANO", data=df)
```

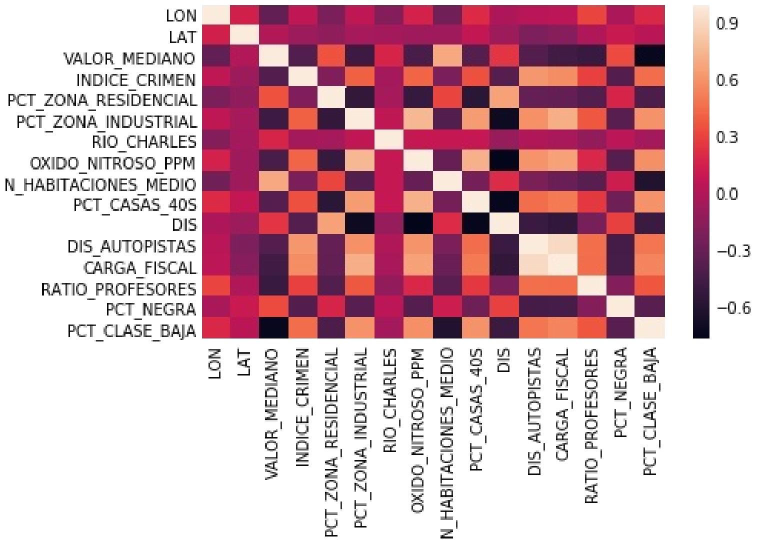
```
Out[26]:

<seaborn.axisgrid.FacetGrid at 0x7f1a51676d68>
```



```
In [27]:
```

```
sns.heatmap(df.corr())
```

```
Out[27]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f1a923bfe80>
```



BokehJS

BokehJS (https://bokeh.pydata.org/en/latest/docs/dev_guide/bokehjs.html) es una librería que nos permitirá realizar graficas pensadas para mostrar gráficos en un navegador. Bokeh es una librería para visualizaciones interactivas diseñada para funcionar en los navegadores web modernos. Su objetivo es proporcionar una construcción elegante y concisa de gráficos modernos al estilo de D3.js, y para ampliar esta capacidad con la interactividad y buen rendimiento sobre grandes volúmenes de datos. Bokeh puede ayudar a cualquier persona a crear en forma rápida y sencilla gráficos interactivos, dashboards y aplicaciones de datos

In [28]:

```
import bokeh.plotting as bk

bk.output_notebook()
```

(https://bokeh.pydata.org) BokehJS 0.12.16 successfully loaded.

In [29]:

```
df.INDICE_CRIMEN
```

Out[29]:

```
0       0.00632
1       0.02731
2       0.02729
3       0.03237
4       0.06905
5       0.02985
6       0.08829
7       0.14455
8       0.21124
9       0.17004
10      0.22489
11      0.11747
12      0.09378
13      0.62976
14      0.63796
15      0.62739
16      1.05393
17      0.78420
18      0.80271
19      0.72580
20      1.25179
21      0.85204
22      1.23247
23      0.98843
24      0.75026
25      0.84054
26      0.67191
27      0.95577
28      0.77299
29      1.00245
         ...
476     4.87141
477    15.02340
478    10.23300
479    14.33370
480     5.82401
481     5.70818
482     5.73116
483     2.81838
484     2.37857
485     3.67367
486     5.69175
487     4.83567
488     0.15086
489     0.18337
490     0.20746
491     0.10574
492     0.11132
493     0.17331
494     0.27957
495     0.17899
496     0.28960
497     0.26838
498     0.23912
499     0.17783
500     0.22438
501     0.06263
502     0.04527
503     0.06076
504     0.10959
505     0.04741
Name: INDICE_CRIMEN, Length: 506, dtype: float64
```

In [30]:

```
df["CRIMEN_QUINTIL"] = pd.qcut(df.INDICE_CRIMEN, 5)
```

In [31]:

```
df.CRIMEN_QUINTIL.cat.categories
```

Out[31]:

```
IntervalIndex([(0.00532, 0.0642], (0.0642, 0.15], (0.15, 0.55], (0.55, 5.581], (5.581, 88.976]],
              closed='right',
              dtype='interval[float64]')
```

In [32]:

```
from bokeh.palettes import brewer

colors = brewer["Spectral"][len(df.CRIMEN_QUINTIL.unique())]
colors
```

Out[32]:

```
['#2b83ba', '#abdda4', '#ffffbf', '#fdae61', '#d7191c']
```

In [33]:

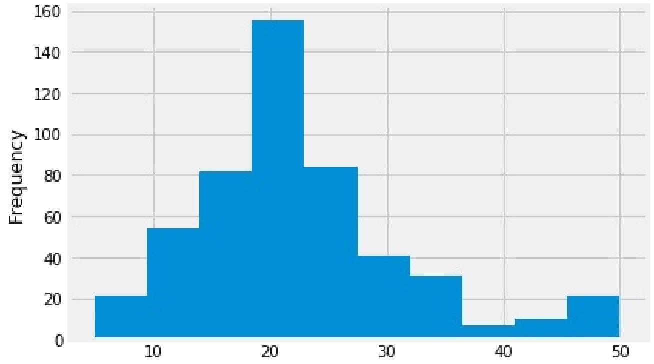
```
p = bk.figure(
plot_width=600,
    plot_height=600,
    title="Habitaciones vs Valor vivienda vs crimen"
)

for i, quintil in enumerate(df.CRIMEN_QUINTIL.cat.categories):
    df_q = df[df.CRIMEN_QUINTIL==quintil]
    p.scatter(df_q.N_HABITACIONES_MEDIO, df_q.VALOR_MEDIANO, color=colors[i],
             legend="({}-{})".format(quintil.left, quintil.right)
             )

bk.show(p);
```

In [34]:

```
df.VALOR_MEDIANO.plot.hist()
```

Out[34]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1a4df7dbe0>
```



In [35]:

```
import numpy as np

hist, edges = np.histogram(df.VALOR_MEDIANO, bins=20)
```

In [36]:

```
hist
```

Out[36]:

```
array([ 9, 12, 18, 36, 41, 41, 84, 71, 72, 12, 23, 18, 17, 14,  6,  1,  5,
        5,  2, 19])
```

In [37]:

```
edges
```

Out[37]:

```
array([ 5.  ,  7.25,  9.5 , 11.75, 14.  , 16.25, 18.5 , 20.75, 23.  ,
       25.25, 27.5 , 29.75, 32.  , 34.25, 36.5 , 38.75, 41.  , 43.25,
       45.5 , 47.75, 50.  ])
```

In [38]:

```
p1 = bk.figure(title="Histograma valor viviendas", tools="save,hover", background_fill_color="#E8DDCB")



p1.quad(top=hist,bottom=0, left=edges[:-1], right=edges[1:], fill_color="#026560")

bk.show(p1)
```

In [41]:

```
from altair import Chart, Color, Scale
```

In [43]:

```
chart = Chart(df)

scale = Scale(range=['#996666', '#b34d4d', '#cc3333','#e61919','#ff0000'])


chart.mark_point().encode(
x="N_HABITACIONES_MEDIO",
    y="VALOR_MEDIANO",
    color=Color("CRIMEN_QUINTIL", scale=scale)

)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/opt/anaconda3/lib/python3.6/site-packages/altair/vegalite/v3/api.py in to_dict(self, *args, **kwarg
s)
    357         copy = self.copy()
    358         original_data = getattr(copy, 'data', Undefined)
--> 359         copy.data = _prepare_data(original_data, context)
    360 
    361         if original_data is not Undefined:

/opt/anaconda3/lib/python3.6/site-packages/altair/vegalite/v3/api.py in _prepare_data(data, context)
     90     # consolidate inline data to top-level datasets
     91     if data_transformers.consolidate_datasets:
---> 92         data = _consolidate_data(data, context)
     93 
     94     # if data is still not a recognized type, then return

/opt/anaconda3/lib/python3.6/site-packages/altair/vegalite/v3/api.py in _consolidate_data(data, cont
ext)
     57 
     58     if values is not Undefined:
---> 59         name = _dataset_name(values)
     60         data = core.NamedData(name=name, **kwds)
     61         context.setdefault('datasets', {})[name] = values

/opt/anaconda3/lib/python3.6/site-packages/altair/vegalite/v3/api.py in _dataset_name(values)
     33     if isinstance(values, core.InlineDataset):
     34         values = values.to_dict()
---> 35     values_json = json.dumps(values, sort_keys=True)
     36     hsh = hashlib.md5(values_json.encode()).hexdigest()
     37     return 'data-' + hsh

/opt/anaconda3/lib/python3.6/json/__init__.py in dumps(obj, skipkeys, ensure_ascii, check_circular,
allow_nan, cls, indent, separators, default, sort_keys, **kw)
    236         check_circular=check_circular, allow_nan=allow_nan, indent=indent,
    237         separators=separators, default=default, sort_keys=sort_keys,
--> 238         **kw).encode(obj)
    239 
    240 

/opt/anaconda3/lib/python3.6/json/encoder.py in encode(self, o)
    197         # exceptions aren't as detailed.  The list call should be roughly
    198         # equivalent to the PySequence_Fast that ''.join() would do.
--> 199         chunks = self.iterencode(o, _one_shot=True)
    200         if not isinstance(chunks, (list, tuple)):
    201             chunks = list(chunks)

/opt/anaconda3/lib/python3.6/json/encoder.py in iterencode(self, o, _one_shot)
    255                 self.key_separator, self.item_separator, self.sort_keys,
    256                 self.skipkeys, _one_shot)
--> 257         return _iterencode(o, 0)
    258 
    259 def _make_iterencode(markers, _default, _encoder, _indent, _floatstr,

/opt/anaconda3/lib/python3.6/json/encoder.py in default(self, o)
    178         """
    179         raise TypeError("Object of type '%s' is not JSON serializable" %
--> 180                         o.__class__.__name__)
    181 
    182     def encode(self, o):

TypeError: Object of type 'Interval' is not JSON serializable
```

Out[43]:

```
Chart({
  data:                    CIUDAD      LON      LAT  VALOR_MEDIANO  INDICE_CRIMEN  \
  0              Nahant -70.9550  42.2550           24.0        0.00632   
  1          Swampscott -70.9500  42.2875           21.6        0.02731   
  2          Swampscott -70.9360  42.2830           34.7        0.02729   
  3          Marblehead -70.9280  42.2930           33.4        0.03237   
  4          Marblehead -70.9220  42.2980           36.2        0.06905   
  5          Marblehead -70.9165  42.3040           28.7        0.02985   
  6               Salem -70.9360  42.2970           22.9        0.08829   
  7               Salem -70.9375  42.3100           22.1        0.14455   
  8               Salem -70.9330  42.3120           16.5        0.21124   
  9               Salem -70.9290  42.3160           18.9        0.17004   
  10              Salem -70.9350  42.3160           15.0        0.22489   
  11              Salem -70.9440  42.3170           18.9        0.11747   
  12              Salem -70.9510  42.3060           21.7        0.09378   
  13               Lynn -70.9645  42.2920           20.4        0.62976   
  14               Lynn -70.9720  42.2870           18.2        0.63796   
  15               Lynn -70.9765  42.2940           19.9        0.62739   
  16               Lynn -70.9870  42.2985           23.1        1.05393   
```

```
17                  Lynn -70.9780  42.2850          17.5        0.78420
18                  Lynn -70.9925  42.2825          20.2        0.80271
19                  Lynn -70.9880  42.2776          18.2        0.72580
20                  Lynn -70.9835  42.2770          13.6        1.25179
21                  Lynn -70.9820  42.2810          19.6        0.85204
22                  Lynn -70.9775  42.2790          15.2        1.23247
23                  Lynn -70.9730  42.2790          14.5        0.98843
24                  Lynn -70.9693  42.2816          15.6        0.75026
25                  Lynn -70.9640  42.2840          13.9        0.84054
26                  Lynn -70.9597  42.2870          16.6        0.67191
27                  Lynn -70.9597  42.2825          14.8        0.95577
28                  Lynn -70.9570  42.2800          18.4        0.77299
29                  Lynn -70.9510  42.2780          21.0        1.00245
..                   ...      ...      ...           ...            ...
476  Boston Forest Hills -71.0565  42.1880          16.7        4.87141
477  Boston Forest Hills -71.0528  42.1920          12.0       15.02340
478  Boston Forest Hills -71.0558  42.1913          14.6       10.23300
479  Boston Forest Hills -71.0670  42.1945          21.4       14.33370
480  Boston West Roxbury -71.1008  42.1740          23.0        5.82401
481  Boston West Roxbury -71.0950  42.1730          23.7        5.70818
482  Boston West Roxbury -71.0900  42.1665          25.0        5.73116
483  Boston West Roxbury -71.0975  42.1608          21.8        2.81838
484     Boston Hyde Park -71.0804  42.1540          20.6        2.37857
485     Boston Hyde Park -71.0750  42.1455          21.2        3.67367
486     Boston Hyde Park -71.0715  42.1550          19.1        5.69175
487     Boston Hyde Park -71.0650  42.1610          20.6        4.83567
488              Chelsea -71.0189  42.2344          15.2        0.15086
489              Chelsea -71.0228  42.2335           7.0        0.18337
490              Chelsea -71.0245  42.2368           8.1        0.20746
491              Chelsea -71.0160  42.2382          13.6        0.10574
492              Chelsea -71.0297  42.2447          20.1        0.11132
493               Revere -71.0125  42.2462          21.8        0.17331
494               Revere -71.0125  42.2500          24.5        0.27957
495               Revere -71.0105  42.2547          23.1        0.17899
496               Revere -71.0010  42.2525          19.7        0.28960
497               Revere -70.9947  42.2496          18.3        0.26838
498               Revere -71.0050  42.2455          21.2        0.23912
499               Revere -70.9985  42.2430          17.5        0.17783
500               Revere -70.9920  42.2380          16.8        0.22438
501             Winthrop -70.9860  42.2312          22.4        0.06263
502             Winthrop -70.9910  42.2275          20.6        0.04527
503             Winthrop -70.9948  42.2260          23.9        0.06076
504             Winthrop -70.9875  42.2240          22.0        0.10959
505             Winthrop -70.9825  42.2210          19.0        0.04741

     PCT_ZONA_RESIDENCIAL  PCT_ZONA_INDUSTRIAL  RIO_CHARLES  \
0                    18.0                 2.31            0
1                     0.0                 7.07            0
2                     0.0                 7.07            0
3                     0.0                 2.18            0
4                     0.0                 2.18            0
5                     0.0                 2.18            0
6                    12.5                 7.87            0
7                    12.5                 7.87            0
8                    12.5                 7.87            0
9                    12.5                 7.87            0
10                   12.5                 7.87            0
11                   12.5                 7.87            0
12                   12.5                 7.87            0
13                    0.0                 8.14            0
14                    0.0                 8.14            0
15                    0.0                 8.14            0
16                    0.0                 8.14            0
17                    0.0                 8.14            0
18                    0.0                 8.14            0
19                    0.0                 8.14            0
20                    0.0                 8.14            0
21                    0.0                 8.14            0
22                    0.0                 8.14            0
23                    0.0                 8.14            0
24                    0.0                 8.14            0
25                    0.0                 8.14            0
26                    0.0                 8.14            0
27                    0.0                 8.14            0
28                    0.0                 8.14            0
29                    0.0                 8.14            0
..                    ...                  ...          ...
476                   0.0                18.10            0
477                   0.0                18.10            0
478                   0.0                18.10            0
479                   0.0                18.10            0
480                   0.0                18.10            0
481                   0.0                18.10            0
```

```
482                   0.0                18.10            0
483                   0.0                18.10            0
484                   0.0                18.10            0
485                   0.0                18.10            0
486                   0.0                18.10            0
487                   0.0                18.10            0
488                   0.0                27.74            0
489                   0.0                27.74            0
490                   0.0                27.74            0
491                   0.0                27.74            0
492                   0.0                27.74            0
493                   0.0                 9.69            0
494                   0.0                 9.69            0
495                   0.0                 9.69            0
496                   0.0                 9.69            0
497                   0.0                 9.69            0
498                   0.0                 9.69            0
499                   0.0                 9.69            0
500                   0.0                 9.69            0
501                   0.0                11.93            0
502                   0.0                11.93            0
503                   0.0                11.93            0
504                   0.0                11.93            0
505                   0.0                11.93            0

     OXIDO_NITROSO_PPM  N_HABITACIONES_MEDIO  PCT_CASAS_40S     DIS  \
0                0.538                 6.575           65.2  4.0900   
1                0.469                 6.421           78.9  4.9671   
2                0.469                 7.185           61.1  4.9671   
3                0.458                 6.998           45.8  6.0622   
4                0.458                 7.147           54.2  6.0622   
5                0.458                 6.430           58.7  6.0622   
6                0.524                 6.012           66.6  5.5605   
7                0.524                 6.172           96.1  5.9505   
8                0.524                 5.631          100.0  6.0821   
9                0.524                 6.004           85.9  6.5921   
10               0.524                 6.377           94.3  6.3467   
11               0.524                 6.009           82.9  6.2267   
12               0.524                 5.889           39.0  5.4509   
13               0.538                 5.949           61.8  4.7075   
14               0.538                 6.096           84.5  4.4619   
15               0.538                 5.834           56.5  4.4986   
16               0.538                 5.935           29.3  4.4986   
17               0.538                 5.990           81.7  4.2579   
18               0.538                 5.456           36.6  3.7965   
19               0.538                 5.727           69.5  3.7965   
20               0.538                 5.570           98.1  3.7979   
21               0.538                 5.965           89.2  4.0123   
22               0.538                 6.142           91.7  3.9769   
23               0.538                 5.813          100.0  4.0952   
24               0.538                 5.924           94.1  4.3996   
25               0.538                 5.599           85.7  4.4546   
26               0.538                 5.813           90.3  4.6820   
27               0.538                 6.047           88.8  4.4534   
28               0.538                 6.495           94.4  4.4547   
29               0.538                 6.674           87.3  4.2390   
..                 ...                   ...            ...     ...   
476              0.614                 6.484           93.6  2.3053   
477              0.614                 5.304           97.3  2.1007   
478              0.614                 6.185           96.7  2.1705   
479              0.614                 6.229           88.0  1.9512   
480              0.532                 6.242           64.7  3.4242   
481              0.532                 6.750           74.9  3.3317   
482              0.532                 7.061           77.0  3.4106   
483              0.532                 5.762           40.3  4.0983   
484              0.583                 5.871           41.9  3.7240   
485              0.583                 6.312           51.9  3.9917   
486              0.583                 6.114           79.8  3.5459   
487              0.583                 5.905           53.2  3.1523   
488              0.609                 5.454           92.7  1.8209   
489              0.609                 5.414           98.3  1.7554   
490              0.609                 5.093           98.0  1.8226   
491              0.609                 5.983           98.8  1.8681   
492              0.609                 5.983           83.5  2.1099   
493              0.585                 5.707           54.0  2.3817   
494              0.585                 5.926           42.6  2.3817   
495              0.585                 5.670           28.8  2.7986   
496              0.585                 5.390           72.9  2.7986   
497              0.585                 5.794           70.6  2.8927   
498              0.585                 6.019           65.3  2.4091   
499              0.585                 5.569           73.5  2.3999   
500              0.585                 6.027           79.7  2.4982   
501              0.573                 6.593           69.1  2.4786   
```

```
502              0.573                 6.120           76.7  2.2875
503              0.573                 6.976           91.0  2.1675
504              0.573                 6.794           89.3  2.3889
505              0.573                 6.030           80.8  2.5050

     DIS_AUTOPISTAS  CARGA_FISCAL  RATIO_PROFESORES  PCT_NEGRA  \
0                 1           296              15.3     396.90
1                 2           242              17.8     396.90
2                 2           242              17.8     392.83
3                 3           222              18.7     394.63
4                 3           222              18.7     396.90
5                 3           222              18.7     394.12
6                 5           311              15.2     395.60
7                 5           311              15.2     396.90
8                 5           311              15.2     386.63
9                 5           311              15.2     386.71
10                5           311              15.2     392.52
11                5           311              15.2     396.90
12                5           311              15.2     390.50
13                4           307              21.0     396.90
14                4           307              21.0     380.02
15                4           307              21.0     395.62
16                4           307              21.0     386.85
17                4           307              21.0     386.75
18                4           307              21.0     288.99
19                4           307              21.0     390.95
20                4           307              21.0     376.57
21                4           307              21.0     392.53
22                4           307              21.0     396.90
23                4           307              21.0     394.54
24                4           307              21.0     394.33
25                4           307              21.0     303.42
26                4           307              21.0     376.88
27                4           307              21.0     306.38
28                4           307              21.0     387.94
29                4           307              21.0     380.23
..              ...           ...               ...        ...
476              24           666              20.2     396.21
477              24           666              20.2     349.48
478              24           666              20.2     379.70
479              24           666              20.2     383.32
480              24           666              20.2     396.90
481              24           666              20.2     393.07
482              24           666              20.2     395.28
483              24           666              20.2     392.92
484              24           666              20.2     370.73
485              24           666              20.2     388.62
486              24           666              20.2     392.68
487              24           666              20.2     388.22
488               4           711              20.1     395.09
489               4           711              20.1     344.05
490               4           711              20.1     318.43
491               4           711              20.1     390.11
492               4           711              20.1     396.90
493               6           391              19.2     396.90
494               6           391              19.2     396.90
495               6           391              19.2     393.29
496               6           391              19.2     396.90
497               6           391              19.2     396.90
498               6           391              19.2     396.90
499               6           391              19.2     395.77
500               6           391              19.2     396.90
501               1           273              21.0     391.99
502               1           273              21.0     396.90
503               1           273              21.0     396.90
504               1           273              21.0     393.45
505               1           273              21.0     396.90

     PCT_CLASE_BAJA     CRIMEN_QUINTIL
0              4.98  (0.00532, 0.0642]
1              9.14  (0.00532, 0.0642]
2              4.03  (0.00532, 0.0642]
3              2.94  (0.00532, 0.0642]
4              5.33     (0.0642, 0.15]
5              5.21  (0.00532, 0.0642]
6             12.43     (0.0642, 0.15]
7             19.15     (0.0642, 0.15]
8             29.93       (0.15, 0.55]
9             17.10       (0.15, 0.55]
10            20.45       (0.15, 0.55]
11            13.27     (0.0642, 0.15]
12            15.71     (0.0642, 0.15]
13             8.26      (0.55, 5.581]
  14            10.26      (0.55, 5.581]
  15             8.47      (0.55, 5.581]
  16             6.58      (0.55, 5.581]
  17            14.67      (0.55, 5.581]
  18            11.69      (0.55, 5.581]
  19            11.28      (0.55, 5.581]
  20            21.02      (0.55, 5.581]
  21            13.83      (0.55, 5.581]
  22            18.72      (0.55, 5.581]
  23            19.88      (0.55, 5.581]
  24            16.30      (0.55, 5.581]
  25            16.51      (0.55, 5.581]
  26            14.81      (0.55, 5.581]
  27            17.28      (0.55, 5.581]
  28            12.80      (0.55, 5.581]
  29            11.98      (0.55, 5.581]
  ..              ...                ...
  476           18.68      (0.55, 5.581]
  477           24.91    (5.581, 88.976]
  478           18.03    (5.581, 88.976]
  479           13.11    (5.581, 88.976]
  480           10.74    (5.581, 88.976]
  481            7.74    (5.581, 88.976]
  482            7.01    (5.581, 88.976]
  483           10.42      (0.55, 5.581]
  484           13.34      (0.55, 5.581]
  485           10.58      (0.55, 5.581]
  486           14.98    (5.581, 88.976]
  487           11.45      (0.55, 5.581]
  488           18.06       (0.15, 0.55]
  489           23.97       (0.15, 0.55]
  490           29.68       (0.15, 0.55]
  491           18.07     (0.0642, 0.15]
  492           13.35     (0.0642, 0.15]
  493           12.01       (0.15, 0.55]
  494           13.59       (0.15, 0.55]
  495           17.60       (0.15, 0.55]
  496           21.14       (0.15, 0.55]
  497           14.10       (0.15, 0.55]
  498           12.92       (0.15, 0.55]
  499           15.10       (0.15, 0.55]
  500           14.33       (0.15, 0.55]
  501            9.67  (0.00532, 0.0642]
  502            9.08  (0.00532, 0.0642]
  503            5.64  (0.00532, 0.0642]
  504            6.48     (0.0642, 0.15]
  505            7.88  (0.00532, 0.0642]

  [506 rows x 18 columns],
  encoding: FacetedEncoding({
    color: Color({
      scale: Scale({
        range: ['#996666', '#b34d4d', '#cc3333', '#e61919', '#ff0000']
      }),
      shorthand: 'CRIMEN_QUINTIL'
    }),
    x: X({
      shorthand: 'N_HABITACIONES_MEDIO'
    }),
    y: Y({
      shorthand: 'VALOR_MEDIANO'
    })
  }),
  mark: 'point'
})
```