

In [1]: `import pandas as pd`

In [2]: `import numpy as np`

In [3]: `ip=pd.read_csv("geoip2-ipv4_csv.csv")`

In [4]: `ip.describe()`

Out[4]:

	geoname_id
count	1.715610e+05
mean	3.498003e+06
std	2.002172e+06
min	4.951800e+04
25%	2.017370e+06
50%	2.921044e+06
75%	6.252001e+06
max	7.909807e+06

In [5]: `ip.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 172754 entries, 0 to 172753
Data columns (total 8 columns):
network                172754 non-null object
geoname_id             171561 non-null float64
continent_code         120141 non-null object
continent_name         171561 non-null object
country_iso_code       171513 non-null object
country_name          171561 non-null object
is_anonymous_proxy     172754 non-null bool
is_satellite_provider  172754 non-null bool
dtypes: bool(2), float64(1), object(5)
memory usage: 8.2+ MB
```

In [7]: `ip.head()`

Out[7]:

	network	geoname_id	continent_code	continent_name	country_iso_cod
0	41.74.160.0/20	49518.0	AF	Africa	RW
1	41.77.160.0/22	49518.0	AF	Africa	RW
2	41.138.80.0/21	49518.0	AF	Africa	RW
3	41.186.0.0/16	49518.0	AF	Africa	RW
4	41.197.0.0/16	49518.0	AF	Africa	RW

In [8]: ip.head(100)

Out[8]:

	network	geoname_id	continent_code	continent_name	country_iso_
0	41.74.160.0/20	49518.0	AF	Africa	RW
1	41.77.160.0/22	49518.0	AF	Africa	RW
2	41.138.80.0/21	49518.0	AF	Africa	RW
3	41.186.0.0/16	49518.0	AF	Africa	RW
4	41.197.0.0/16	49518.0	AF	Africa	RW
5	41.215.248.0/22	49518.0	AF	Africa	RW
6	41.216.96.0/20	49518.0	AF	Africa	RW
7	41.216.112.0/21	49518.0	AF	Africa	RW
8	41.216.120.0/22	49518.0	AF	Africa	RW
9	41.222.244.0/22	49518.0	AF	Africa	RW
10	41.242.140.0/22	49518.0	AF	Africa	RW
11	104.143.19.0/24	49518.0	AF	Africa	RW
12	105.21.96.0/19	49518.0	AF	Africa	RW
13	105.178.0.0/15	49518.0	AF	Africa	RW
14	154.68.64.0/18	49518.0	AF	Africa	RW
15	196.12.140.0/22	49518.0	AF	Africa	RW
16	196.12.144.0/22	49518.0	AF	Africa	RW
17	196.44.240.0/20	49518.0	AF	Africa	RW
18	196.49.7.0/24	49518.0	AF	Africa	RW
19	196.223.12.0/24	49518.0	AF	Africa	RW
20	197.157.128.0/18	49518.0	AF	Africa	RW
21	197.157.212.0/22	49518.0	AF	Africa	RW
22	197.234.244.0/22	49518.0	AF	Africa	RW
23	197.243.0.0/17	49518.0	AF	Africa	RW
24	41.78.72.0/22	51537.0	AF	Africa	SO
25	41.79.196.0/22	51537.0	AF	Africa	SO
26	41.223.108.0/22	51537.0	AF	Africa	SO
27	80.78.20.233/32	51537.0	AF	Africa	SO
28	80.78.20.234/31	51537.0	AF	Africa	SO
29	80.78.20.236/30	51537.0	AF	Africa	SO
...	...	...	...	...	...

70	131.117.160.0/21	69543.0	AS	Asia	YE
71	155.254.201.0/24	69543.0	AS	Asia	YE
72	185.11.8.0/22	69543.0	AS	Asia	YE
73	185.71.132.0/22	69543.0	AS	Asia	YE
74	185.80.44.0/22	69543.0	AS	Asia	YE
75	185.80.140.0/22	69543.0	AS	Asia	YE
76	195.94.0.0/19	69543.0	AS	Asia	YE
77	198.69.12.0/23	69543.0	AS	Asia	YE
78	205.160.110.0/23	69543.0	AS	Asia	YE
79	5.1.104.0/21	99237.0	AS	Asia	IQ
80	5.8.240.0/21	99237.0	AS	Asia	IQ
81	5.10.224.0/21	99237.0	AS	Asia	IQ
82	5.42.192.0/19	99237.0	AS	Asia	IQ
83	5.104.72.0/21	99237.0	AS	Asia	IQ
84	5.149.96.0/20	99237.0	AS	Asia	IQ
85	31.24.200.0/22	99237.0	AS	Asia	IQ
86	31.25.136.0/21	99237.0	AS	Asia	IQ
87	31.177.39.0/24	99237.0	AS	Asia	IQ
88	37.17.128.0/19	99237.0	AS	Asia	IQ
89	37.77.48.0/21	99237.0	AS	Asia	IQ
90	37.77.64.0/20	99237.0	AS	Asia	IQ
91	37.98.224.0/21	99237.0	AS	Asia	IQ
92	37.205.112.0/21	99237.0	AS	Asia	IQ
93	37.236.0.0/14	99237.0	AS	Asia	IQ
94	46.30.224.0/21	99237.0	AS	Asia	IQ
95	46.31.72.0/21	99237.0	AS	Asia	IQ
96	46.243.16.0/21	99237.0	AS	Asia	IQ
97	46.253.128.0/20	99237.0	AS	Asia	IQ
98	62.145.104.0/22	99237.0	AS	Asia	IQ
99	62.201.192.0/18	99237.0	AS	Asia	IQ

100 rows x 8 columns



In [9]: `ip.tail()`

Out[9]:

--	--	--	--	--	--

	network	geoname_id	continent_code	continent_name	count
172749	217.194.157.32/27	NaN	NaN	NaN	NaN
172750	217.194.157.64/26	NaN	NaN	NaN	NaN
172751	217.194.157.128/25	NaN	NaN	NaN	NaN
172752	217.194.158.0/23	NaN	NaN	NaN	NaN
172753	217.194.222.0/25	NaN	NaN	NaN	NaN



In [10]: `import matplotlib.pyplot as plt`

In [13]: `plt.show()`

In [18]: `ip`

In [18]:

In [18]:

In [21]: `ip.hist(edgecolor='black', linewidth=1.5)  
fig=plt.gcf()  
plt.show()`

KeyError Traceback (most recent call |  
ast)

/srv/conda/lib/python3.6/site-packages/numpy/lib/histograms.py in \_unsigned\_subtract(a, b)

```

    279     try:
--> 280         dt = signed_to_unsigned[dt.type]
    281     except KeyError:
```

KeyError: <class 'numpy.bool\_'>

During handling of the above exception, another exception occurred:

TypeError Traceback (most recent call |  
ast)

```

<ipython-input-21-72e866792314> in <module>()
----> 1 ip.hist(edgecolor='black', linewidth=1.5)
      2 fig=plt.gcf()
      3 plt.show()
```

/srv/conda/lib/python3.6/site-packages/pandas/plotting/\_core.py in hist  
\_frame(data, column, by, grid, xlabelsize, xrot, ylabelsize, yrot, ax,  
sharex, sharey, figsize, layout, bins, \*\*kwds)

```

    2135     for i, col in enumerate(_try_sort(data.columns)):
    2136         ax = _axes[i]
-> 2137         ax.hist(data[col].dropna().values, bins=bins, **kwds)
    2138         ax.set_title(col)
    2139         ax.grid(grid)
```

```

/srv/conda/lib/python3.6/site-packages/matplotlib/_init_.py in inner
(ax, *args, **kwargs)
1715 warnings.warn(msg % (label_namer, func._na
me_),
1716 RuntimeWarning, stacklevel=2)
-> 1717 return func(ax, *args, **kwargs)
1718 pre_doc = inner._doc__
1719 if pre_doc is None:

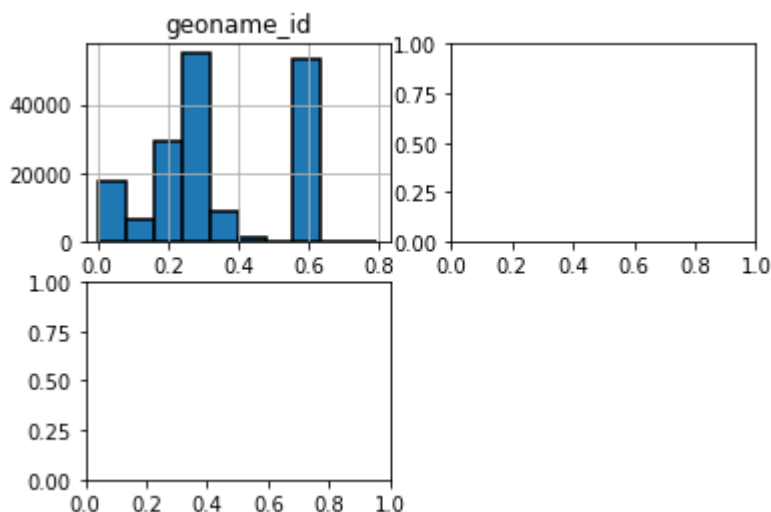
/srv/conda/lib/python3.6/site-packages/matplotlib/axes/_axes.py in hist
(***failed resolving arguments***)
6163 # this will automatically overwrite bins,
6164 # so that each histogram uses the same bins
-> 6165 m, bins = np.histogram(x[i], bins, weights=w[i], **
hist_kwargs)
6166 m = m.astype(float) # causes problems later if i
t's an int
6167 if mlast is None:

/srv/conda/lib/python3.6/site-packages/numpy/lib/histograms.py in histo
gram(a, bins, range, normed, weights, density)
730
731 # Pre-compute histogram scaling factor
--> 732 norm = n_equal_bins / _unsigned_subtract(last_edge, fir
st_edge)
733
734 # We iterate over blocks here for two reasons: the firs
t is that for

/srv/conda/lib/python3.6/site-packages/numpy/lib/histograms.py in _unsi
gned_subtract(a, b)
280 dt = signed_to_unsigned[dt.type]
281 except KeyError:
--> 282 return np.subtract(a, b, dtype=dt)
283 else:
284 # we know the inputs are integers, and we are deliberat
ely casting

TypeError: numpy boolean subtract, the `^` operator, is deprecated, use
the bitwise_xor, the `^` operator, or the logical_xor function instead.

```



```
In [25]: ip.plot(kind="scatter", x="geoname_id", y="network")
```

```
-----  
-----  
ValueError                                Traceback (most recent call  
ast)  
<ipython-input-25-e9f966f3eeed> in <module>()  
----> 1 ip.plot(kind="scatter", x="geoname_id", y="network")  
  
/srv/conda/lib/python3.6/site-packages/pandas/plotting/_core.py in _ca  
ll__(self, x, y, kind, ax, subplots, sharex, sharey, layout, figsize, u  
se_index, title, grid, legend, style, logx, logy, loglog, xticks, ytic  
s, xlim, ylim, rot, fontsize, colormap, table, yerr, xerr, secondary_y,  
sort_columns, **kwds)  
    2625                                     fontsize=fontsize, colormap=colormap,  
table=table,  
    2626                                     yerr=yerr, xerr=xerr, secondary_y=sec  
ondary_y,  
-> 2627                                     sort_columns=sort_columns, **kwds)  
    2628     __call__.__doc__ = plot_frame.__doc__  
    2629  
  
/srv/conda/lib/python3.6/site-packages/pandas/plotting/_core.py in plot  
_frame(data, x, y, kind, ax, subplots, sharex, sharey, layout, figsize,  
use_index, title, grid, legend, style, logx, logy, loglog, xticks, ytic  
ks, xlim, ylim, rot, fontsize, colormap, table, yerr, xerr, secondary_  
y, sort_columns, **kwds)  
    1867                                     yerr=yerr, xerr=xerr,  
    1868                                     secondary_y=secondary_y, sort_columns=sort_col  
umns,  
-> 1869                                     **kwds)  
    1870
```